



МИР электроники

К. ФРИКЕ

Вводный курс цифровой электроники

Перевод с немецкого
под редакцией и с дополнением
В.Я. Кремлева

*Рекомендовано Московским
Институтом Электронной Техники
(МИЭТ) в качестве учебного пособия
для студентов, специализирующихся
в области проектирования цифровых
интегральных схем*

ТЕХНОСФЕРА

Москва

2003

К.Фрике

Вводный курс цифровой электроники

Москва:

Техносфера, 2003. — 432с. ISBN 5-94836-015-6

Книга дает научно обоснованное введение в цифровую технику, предоставляя в полной мере ее основы, вплоть до конструирования и программирования. Читатель получает знания, которые делают возможным понимание большинства цифровых технических схем.

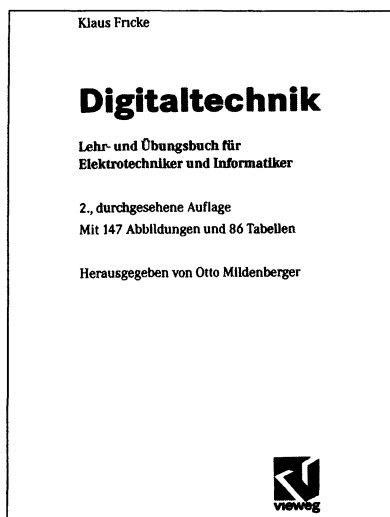
Особенно детально рассматривается синтез логических схем. На примерах обсуждаются наиболее часто применяемые стандартные схемы — мультиплексоры и преобразователи кода. Подробно анализируются основы арифметики с фиксированной запятой и аппаратная реализация арифметических блоков.

Для различных блоков ЗУ приводятся типовые временные диаграммы.

В качестве введения в проектирование структур компьютеров представлены драйверы с программным управлением.

Подробно описана элементная база цифровых устройств и рассмотрены ключевые проблемы их энергетики. Дано четкое введение в микропроцессорную технику.

Учебное пособие будет полезно инженерам-практикам, активно использующим элементную базу цифровой техники, программирование микроконтроллеров и ПЛИС.



© 2001 Friedr. Vieweg & Sohn
Verlagsgesellschaft mbH,
Braunschweig/Wiesbaden
Der Verlag Vieweg ist ein Unternehmen
der Fachverlagsgruppe Bertelsmann Springer
© 2003, ЗАО «РИЦ «Техносфера»
перевод на русский язык,
оригинал-макет, оформление

ISBN 5-94836-015-6

ISBN 3-528-13861-0 (нем.)

Содержание

Глава 1.	
Введение	16
Глава 2.	
Кодирование и системы счисления	19
2.1. Коды	19
2.2. Двоичный код	20
2.3. Арифметические операции с фиксированной запятой в двоичной системе	22
2.3.1. Целочисленное сложение в двоичной системе	22
2.3.2. Сложение чисел с фиксированной запятой.....	23
2.3.3. Представление с помощью обратного кода	23
2.3.4. Представление с помощью двойного дополнения (точное дополнение, two's complement)	24
2.3.5. Вычитание при представлении с помощью точного дополнения	25
2.3.6. Переполнение числового диапазона	26
2.3.7. Умножение	27
2.3.8. Деление	28
2.4. Шестнадцатеричный код	29
2.5. Восьмеричный код	30
2.6. Код Грея	30
2.7. Двоично-десятичный код	31
2.8. Алфавитно-цифровые коды	32
2.9. Упражнения	32
Глава 3.	
Переключательная алгебра	34
3.1. Переключательная переменная и переключательная функция	34
3.2. Двухразрядные переключательные функции	36
3.3. Вычислительные правила	38
3.4. Каноническая дизъюнктивная нормальная форма (KDNF).....	40
3.5. Каноническая конъюнктивная нормальная форма (KKNF) ..	41
3.6. Представление функций с помощью KKNF и KDNF.	42
3.7. Минимизация с помощью переключательной алгебры.....	44

3.8.	Способ упрощенной записи	45
3.9.	Символическое обозначение логических элементов	46
3.9.1.	Основы структуры символов	46
3.9.2.	Индексация зависимости	47
3.9.3.	Зависимость вида И (G)	47
3.9.4.	Зависимость по типу ИЛИ (V)	48
3.9.5.	Зависимость по типу EXOR (N).	49
3.9.6.	Образующая соединение зависимость (Z)	49
3.9.7.	Зависимость с передачей (X).....	49
3.10.	Упражнения.....	50
Глава 4.		
Поведение логических вентилей		52
4.1.	Положительная и отрицательная логика	53
4.2.	Определение времени переключения.....	54
4.3.	Передаточная характеристика, запас по помехоустойчи- вости	56
4.4.	Вентили	58
4.4.1.	Вентили с открытым коллектором (open collector) ...	60
4.4.2.	Вентиль с тремя состояниями	61
4.5.	Упражнения.....	62
Глава 5.		
Схемотехника		64
5.1.	КМОП	64
5.1.1.	Нагрузочная способность	66
5.1.2.	Основные схемы NAND и NOR	67
5.1.3.	Передаточный вентиль.....	68
5.1.4.	Элемент с тремя состояниями	69
5.1.5.	Специфические свойства КМОП.....	69
5.2.	ТТЛ	71
5.2.1.	Нагрузка выходов	72
5.3.	Эмиттерно-связанная логика.....	73
5.4.	Интегральная инжекционная логика (I^2L)	74
5.5.	Рассеиваемая мощность и характеристики переключе- ния транзисторных переключателей.....	76
5.6.	Упражнения.....	78

Дополнение.

Элементная база цифровых устройств. Энергетика обработки цифровой информации	79
Д.1. Введение	79

Д.2. Энергетика и скорость производства цифровой информации	79
Д.2.1. Классификация вентилей как энергопотребителей....	81
Д.2.2. Общие вопросы энергетики цифровых вентилей	84
Д.2.3. Скорость производства информации	93
Д.3. Организация и энергетика цифровых схем без отношения.....	105
Д.3.1. Общий принцип организации схем и схемотехнические разновидности	106
Д.3.2. Энергетика производства информации.....	114
Д.3.3. Мощностные характеристики.....	116
Д.4. Схемотехнические и энергетические характеристики логических схем с отношением	118
Д.4.1. Классификация базовых вентилей с отношением.....	119
Д.4.2. Основные схемотехнические и структурно-топологические типы	121
Д.4.3. Особенности обработки информации и энергопотребления.....	128
Д.4.4. Особенности энергопотребления.....	130
Д.5. Схемотехника логических вентилей на токовых ключах..	132
Д.5.1. Общий принцип схемотехнической организации и классификация	133
Д.5.2. Интегральная инжекционная логика.....	136
Д.5.3. Быстродействие и энергетика токовых ключей.....	145
Д.6. Принципы организации частично диссипативных схем ...	149
Д.6.1. Принцип адиабатического переключения	150
Д.6.2. Статические CMOS вентили с коллапсирующим импульсным питанием.	153
Д.6.3. Вентили со ступенчатой перезарядкой выходной емкости	157
Д.7. Вентили с нетрадиционной организацией энергопитания	161
Д.7.1. Питание ионизирующим излучением	161
Д.7.2. Схемотехнические и структурно-топологические решения базовых логических вентилей, питающихся излучением	165
Д.8. Заключение	172
Глава 6.	
Логические схемы	173
6.1. Минимизация с помощью диаграмм Карно-Вейча	173
6.1.1. Минимизация KDNF	173

6.1.2.	Минимизация нормальной ККНФ.....	177
6.1.3.	Диаграммы Карно-Вейча для 2, 3, 4, 5, 6 входных переменных	178
6.1.4.	Неполностью заданные функции.....	178
6.2.	Способ Квина-Мак-Класки	180
6.3.	Другие направления оптимизации.....	184
6.3.1.	Преобразование логической схемы И/ИЛИ в схему НЕ-И ..	185
6.3.2.	Преобразование логической схемы ИЛИ/И в логическую схему НЕ-ИЛИ	186
6.4.	Воздействие времени задержки на логические схемы.....	187
6.4.1.	Отрицательное воздействие на структуру.....	187
6.4.2.	Отрицательное воздействие на функционирование... ..	189
6.4.3.	Классификация отрицательных воздействий	190
6.5.	Упражнения.....	190

Глава 7.

Асинхронные триггеры	193
7.1. Принципиальные особенности структуры триггеров.	194
7.2. Анализ асинхронных триггеров.....	194
7.3. Систематический анализ.	196
7.4. Анализ с учетом задержки вентилей	198
7.5. Элементы ЗУ	201
7.5.1. RS -триггер	201
7.5.2. RS -триггер с тактовым входом	201
7.5.3. D -триггер	204
7.5.4. D-триггер с управлением по переднему фронту им- пульса.....	205
7.5.5. Управление передним и задним фронтом	210
7.5.6. J-K-триггер	211
7.5.7. T-триггер.....	212
7.5.8. Пример	213
7.5.9. Общее о триггерах.	213
7.6. Упражнения.....	216

Глава 8.

Синхронные драйверы.....	219
8.1. Синтез драйверов (пример 1)	219
8.1.1. Постановка задачи	219
8.1.2. Формирование диаграммы состояний.....	220
8.1.3. Структура схемы управления электромеханическими устройствами	221
8.1.4. Формирование таблицы последовательности состояний	222

8.1.5.	Формирование таблицы запуска и таблицы выходных сигналов	222
8.2.	Синтез драйверов (пример 2)	225
8.2.1.	Постановка задачи	225
8.2.2.	Составление диаграммы состояний	225
8.2.3.	Составление таблицы последовательности состояний	226
8.2.4.	Разработка логической схемы для запускающих сигналов	226
8.2.5.	Полная таблица состояний	227
8.2.6.	Полная диаграмма состояний	228
8.2.7.	Временные характеристики драйверов	229
8.3.	Упражнения	231
Глава 9.		
Мультиплексоры и преобразователи кода		234
9.1.	Мультиплексор	234
9.1.1.	Реализация функций мультиплексора	235
9.2.	Преобразователь кода	238
9.2.1.	Преобразователь двоично-десятичного кода в десятичный код 7442	239
9.2.2.	Демльтиплексор	240
9.2.3.	Генерирование наборов функций	241
9.3.	Аналоговые мультиплексоры и демльтиплексоры	243
9.4.	Упражнения	244
Глава 10.		
Цифровые счетчики		247
10.1.	Асинхронный счетчик	247
10.1.1.	Двоичный счетчик по модулю 8	247
10.1.2.	Счетчик по модулю 6	248
10.1.3.	Асинхронный обратный счетчик	249
10.1.4.	Временные характеристики асинхронных счетчиков	250
10.2.	Синхронные счетчики	251
10.2.1.	4-битовый двоичный счетчик	252
10.2.2.	Счетчик по модулю 6 с использованием кода Грея ...	254
10.2.3.	Синхронный 4-битовый реверсивный двоичный счетчик 74191	256
10.3.	Упражнения	257
Глава 11.		
Сдвиговые регистры		258
11.1.	Временные характеристики сдвиговых регистров	259

11.1.1. Сдвиговый регистр 74194	261
11.2. Сдвиговый регистр с обратной связью	262
11.2.1. Счетчик Мебиуса, счетчик Джонсона.....	264
11.2.2. Псевдослучайные последовательности	266
11.3. Упражнения.....	268

Глава 12.

Арифметические устройства	269
12.1. Полный сумматор.....	269
12.2. Последовательный сумматор	269
12.3. Сумматор с последовательным переносом (ripple-carry-adder) ..	270
12.4. Сумматор с параллельным переносом.....	271
12.4.1. Каскадирование сумматоров с параллельным переносом.....	274
12.4.2. Сравнение сумматоров	275
12.5. Арифметико-логические вычислительные устройства (ALU, АЛУ).....	276
12.5.1. Примеры операций	280
12.6. Компараторы.....	283
12.6.1. 2-битовый компаратор	284
12.6.2. Каскадируемые компараторы.....	286
12.7. Упражнения.....	286

Глава 13.

Цифровые ЗУ	288
13.1. Принципиальная структурная схема ЗУ.....	289
13.2. ROM.....	289
13.3. PROM.....	291
13.4. EPROM.....	292
13.5. EEPROM	294
13.6. EAROM	295
13.7. NOVRAM.....	295
13.8. RAM.....	296
13.8.1. Статическое RAM	296
13.8.2. Пример работы RAM	297
13.9. Динамическое RAM	301
13.10. Квасистатическое DRAM	306
13.11. ЗУ «пожарная цепочка»	307
13.11.1. Примеры FIFO	307
13.12. Каскадирование ЗУ	308
13.13. Увеличение длины слов.....	309
13.14. Увеличение емкости ЗУ	309

13.14.1. Полное декодирование.....	311
13.14.2. Частичное декодирование.....	312
13.14.3. Линейное декодирование	314
13.15. Упражнения.....	315

Глава 14.

Программируемые логические блоки.....	318
14.1. Семейства ASIC	319
14.2. Программируемые логические ИС (PLD).....	322
14.2.1. Типы PLD	323
14.3. ROM, EPROM, EEPROM	324
14.4. PLA.....	326
14.5. PAL.....	332
14.6. GAL	334
14.7. Программирование логических блоков PLD	337
14.7.1. Тестирование.....	339
14.8. Программируемые полем вентильные матрицы (FPGA) ..	339
14.8.1. Структура FPGA	340
14.8.2. Конфигурируемые логические блоки (CLB).....	342
14.8.3. IO-блоки.....	342
14.8.4. Соединительные линии.....	343
14.8.5. Программирование FPGA	344
14.9. EPLD.....	345
14.9.1. Пример EPLD:CY7C343.....	345
14.10. Gate-Arrays.....	345
14.10.1. Структура канализированных вентильных матриц ..	346
14.11. ASIC со стандартными ячейками.....	350
14.12. ASIC на основе полностью заказного проектирования	351
14.13. Упражнения.....	351

Глава 15.

Принципы построения микропроцессоров	353
15.1. Кооперирующиеся управляющие схемы	353
15.2. Компьютер фон-Неймана.....	354
15.3. Операционные блоки	355
15.3.1. Архитектура операционных блоков	355
15.3.2. Система шин	357
15.4. Управляющие блоки	358
15.5. Микропрограммирование	360
15.6. Упражнения.....	362

Глава 16.**Структура, система команд и работа**

микропроцессора	363
16.1. Структура микропроцессора.....	363
16.2. Выводы 8085А	366
16.3. Блоки ЗУ и периферийные блоки	369
16.4. Периферийные блоки	370
16.5. Параллельный интерфейс микропроцессора 8255.....	371
16.6. Процесс выполнения команд	372
16.6.1. Цикл выборки операционного кода (OF).....	374
16.6.2. Цикл считывания памяти (MR, memory read)	375
16.6.3. Цикл записи в ЗУ (MW, memory write).....	376
16.6.4. Пример выполнения команды	377
16.7. Управление с помощью прерывания (interrupt)	378
16.8. Ассемблерное программирование	380
16.9. Набор команд	380
16.10. Команда перехода	381
16.10.1. Адресация типа «Register»	381
16.10.2. Адресация типа «Register Indirect»	382
16.10.3. Адресация типа «Immediate»	383
16.10.4. Адресация типа «Direct»	383
16.11. Арифметические команды	384
16.12. Логические операции	386
16.13. Установка и стирание флагов переноса (carry flag).....	387
16.14. Команды сдвига.....	387
16.15. Разветвление программ	387
16.15.1. Пример перехода	388
16.16. Команды для подпрограммы.....	389
16.17. Команды для управления процессором	392
16.18. Ассемблерные команды	394
16.19. Примеры программ	397
16.19.1. Программирование параллельного интерфейса 8255 .	397
16.19.2. Передача данных в подпрограмму	398
16.20. Упражнения.....	400
А. Приложение	401
А.1. Индикация функций	401
Решения задач	405
Литература	427
Список технических терминов	428

ГЛАВА I

ВВЕДЕНИЕ

В последние годы значение цифровой техники все более и более возрастает. Причина этого заключается в значительных преимуществах цифровой техники, при создании очень сложных систем. Это достигается путем представления сигнала двумя значениями, которые могут обрабатываться логическими вентилями с сильно нелинейными передаточными характеристиками без сбоев, накопления и дальнейшего распространения искажений сигнала. Благодаря такому представлению сигналов удалось создать полупроводниковую технологию, позволяющую реализовать до 10^8 элементов на одном кристалле.

Поскольку целью систем цифровой техники является обработка сигналов, следует несколько подробнее рассмотреть понятие «сигнал». Сигналы служат для переноса информации. Они описываются такими физическими величинами, как напряжение, ток, давление, сила и т. д. Амплитуды таких величин зависят от времени. Передаваемая информация заключается в изменяющихся амплитудных значениях. Пусть, например, измеряется зависящий от времени уровень жидкости F в резервуаре. На рис. 1-1а показана зависимость уровня жидкости от времени. Если датчик выдает электрический сигнал, напряжение которого пропорционально величине заполнения, получаем временную зависимость напряжения U_s , показанную на рис. 1-1б. Этот сигнал имеет непрерывную величину, то есть в измеряемом диапазоне могут появиться все значения амплитуды.

Системы, которые могут обрабатывать непрерывные по величине сигналы, называются аналоговыми системами.

Дискретные по величине сигналы, которые также называют цифровыми сигналами, могут в отличие от этого принимать только определенные дискретные значения в фиксированные моменты времени. Примером цифрового сигнала является сигнал, который с помощью двух различных уровней напряжения показывает, закрыта или открыта дверь. Подобные сигналы могут непосредственно обрабатываться цифровыми системами.

В том случае, если сигналы с непрерывной величиной должны передаваться цифровыми системами, эти сигналы должны быть пред-

варительно квантизируются. В соответствии с данным процессом, с определенными временными точками (точками считывания) соотносена амплитуда сигнала дискретной амплитудной ступени. Так сделано на рис. 1.1 с для вышеупомянутого примера, в котором показан датчик уровня заполнения. При этом получаем напряжение с дискретными величинами U_q . При квантизировании необходимо принимать во внимание ошибку округления.

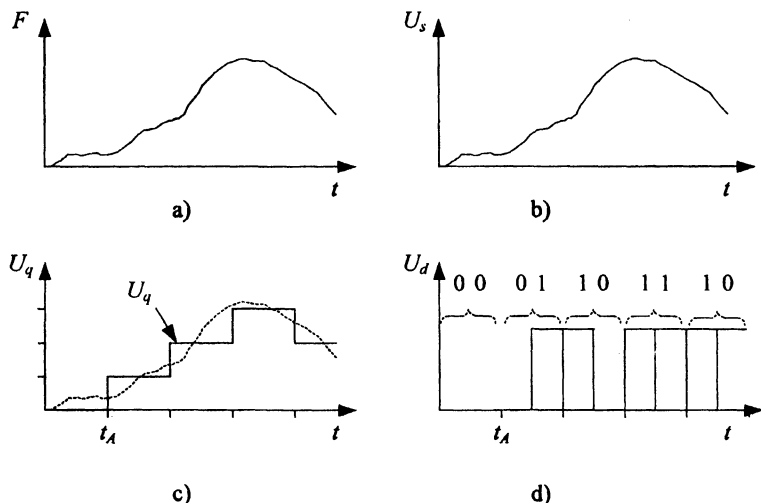


Рис. 1.1. Пример «оцифровывания» сигналов.

- Зависимость от времени t уровня заполнения резервуара F .
- Зависимость от времени выходного напряжения датчика U_s .
- Квантизированная зависимость от времени напряжения U_q при 4-х амплитудных ступенях.
- Приведение амплитуд в соответствие с записью величин в виде кода 00, 01, 10 и 11.

При цифровой передаче сигнал вначале должен быть переведен в цифровую форму. В этом случае амплитуда будет представляться с помощью последовательности цифр. Каждая цифра представляет собой сигнал с дискретным значением. На рис. 1.1 d приведен пример кодирования с помощью двух следующих друг за другом цифр. Амплитудная ступень 0 представлена двумя цифрами. Амплитуды 1, 2 и 3 становятся сочетаниями 01, 10, 11. Следует различать также дискретные во времени сигналы и непрерывные во времени сигналы. Дискретные во времени сигналы могут изменять свою амплитуду только в определенные моменты времени, в то время как непрерывные во времени сигналы могут изменять свою амплитуду

в произвольные моменты времени. Цифровые системы могут быть дискретными во времени, в этом случае их называют синхронными. Синхронизация осуществляется с помощью тактового сигнала.

Благодаря ограничению в использовании конечного числа амплитудных ступеней, цифровая система обладает высокой помехозащищенностью. Подвергнувшиеся воздействию помех цифровые сигналы могут быть однозначно приведены к первоначальным дискретным амплитудным значениям. Но, чтобы не возникла ошибка, помеха не должна превышать половину расстояния между двумя амплитудными ступенями.

Цифровые системы имеют ряд преимуществ перед аналоговыми системами:

- При использовании цифровых сигналов не происходит воспроизведения их искажений, благодаря чему появляется возможность реализации систем любой степени сложности, например, микропроцессоров. Это свойство цифровых систем определяет их превосходство и при передаче на большие расстояния.
- Цифровые системы сравнительно легко проектировать, поскольку способ их описания представляющий собой булеву алгебру — аппарат очень удобный для автоматизации. Сегодня разработка сложных цифровых систем автоматизирована посредством применения высокопроизводительных алгоритмов.
- Цифровые системы можно относительно просто тестировать.

Недостаток цифровых систем:

- Цифровые системы являются более медленными системами, чем аналоговые. Поэтому в области высоких частот доминирует аналоговая техника.

ГЛАВА 2

КОДИРОВАНИЕ И СИСТЕМЫ СЧИСЛЕНИЯ

2.1. Коды

В цифровой технике коды используются для того, чтобы оптимально представить сигнал для какого-либо случая применения. Код отображает символы одного множества через символы второго множества. При этом должна существовать возможность декодирования, чтобы из кодированных символов можно было получить исходные символы.

Известным примером кода является код Морзе. Определение кода производится с помощью таблицы соответствия. Для кода Морзе подобное соответствие отображено в табл. 2.1. Этот код является обратимым, поскольку из буквы можно получить символ Морзе, а из него можно вновь образовать букву. Но это справедливо только для текста, который написан строчными буквами, поскольку код Морзе не делает различия между прописными и строчными буквами. Строго говоря, из кода Морзе нельзя восстановить путем декодирования текст, записанный с помощью строчных и прописных букв.

Таблица 2.1. Код Морзе.

Алфавит	Код Морзе	Алфавит	Код Морзе	Алфавит	Код Морзе
a	· -	j	· - - -	s	· · ·
b	· - - -	k	- · -	t	-
c	- · · ·	l	· - · ·	u	· · -
d	- · ·	m	- -	v	· · · -
e	·	n	- ·	w	· - -
f	· · · ·	o	- - - -	x	- · · ·
g	- - ·	p	· - - ·	y	- - - -
h	· · ·	q	- - - -	z	- · · ·
I	· ·	r	· - ·		

Для каждого применения имеется более или менее подходящий код. Так для проведения операций над числами в компьютере рационально применять другой код, чем для передачи чисел по линии связи. В данной главе исследуются различия между отдельными кодами и даны указания по их специфическому применению.

Комбинацию нескольких символов кода называют словом (word). В последующем мы ограничимся технически важным случаем, в котором все слова одного кода имеют одинаковую длину n . В коде Морзе этот случай не имеет места. Если в код входит множество символов N , то N^n различных слов могут иметь длину n . Если используются все N^n возможных слов одного кода, то в этом случае говорят о минимальном коде. Если используют менее чем N^n слов, то его называют избыточным кодом. Ниже можно найти описание наиболее употребительных кодов, полное описание можно получить из [8].

2.2. Двоичный код

Двоичный код является важнейшим кодом в цифровых системах, поскольку он является универсальным. Благодаря ограничению, в соответствии с которым применяются только символы 1 и 0, становится возможной обработка сигналов с помощью схемных элементов, работающих как переключатели. Двоичный код позволяет также пользоваться арифметикой, аналогичной арифметике десятичных систем. Двоичную систему счисления можно рассматривать как кодирование десятичной системы. Двоичное число состоит из слова, которое образовано символом $C_i \in \{0, 1\}$. Символы C_i одного слова называют в цифровой технике битами. Слово z в двоичном представлении формируется путем последовательного присоединения отдельных битов, как это показано ниже:

$$z_B = c_{n-1}c_{n-2} \cdots c_1c_0, c_{-1}c_{-2} \cdots c_{-m+2}c_{-m+1}c_{-m} \quad (2.1)$$

Двоичное число имеет n разрядов перед запятой и m разрядов после запятой. Отдельным битам присвоены, в соответствии с их позицией i в слове, весовые коэффициенты 2^i . На основе этого можно рассчитать эквивалентное десятичное число z_{10} :

$$\begin{aligned} z_D = g(z_B) = \\ = c_{n-1}2^{n-1} + c_{n-2}2^{n-2} + \dots + c_12^1 + c_02^0 + c_{-1}2^{-1} + \dots + c_{-m}2^{-m} \end{aligned} \quad (2.2)$$

Рассмотрим в качестве примера двоичное число $10110,001_B$, которое как двоичное число отмечено символом B (binary). Оно интерпретируется как:

$$g(z_2) = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = z_{10} = 22,375_{10}$$

Двоичный (или дуальный) код обозначается как взвешенный код, поскольку стоящие дальше влево биты обладают более высокими весовыми коэффициентами. Уравнение (2.2) можно рассматривать как правило, в соответствии с которым производится преобразование двоичных чисел в десятичные числа.

Преобразование десятичных чисел в двоичные числа является более сложным. Оно может быть описано различными алгоритмами для целочисленной и дробной частей. В приведенном выше примере с числом $22,375_{10}$ алгоритм должен быть представлен следующим образом:

- Вначале формируется целочисленная часть двоичного числа. Для этого целочисленная часть десятичного числа последовательно делится на 2 и записывается остаток, пока не будет получен 0.

22 : 2 = 1	остаток 0	$\uparrow$ целочисленная часть двоич- ного числа
11 : 2 = 5	остаток 1	
5 : 2 = 2	остаток 1	
2 : 2 = 1	остаток 0	
1 : 2 = 0	остаток 1	

Соответствующее числу 22_{10} двоичное число представляет собой 10110_2 .

- Второй шаг заключается в преобразовании дробной части десятичного числа в дробную часть двоичного числа. Вначале дробная часть десятичного числа умножается на 2. Целочисленная часть отделяется, она образует разряды двоичного числа с наименьшими значениями.

Процесс повторяется, как это показано ниже.

0,375 · 2 = 0,75	+0	$\downarrow$ дробная часть двоичного числа
0,75 · 2 = 0,5	+1	
0,5 · 2 = 0	+1	

В этом примере мы видим, что остаток равен 0. Но не обязательно так всегда бывает. В нормальном случае дробная часть эквивалентного двоичного числа имеет бесконечно большое количество разрядов. В этом случае необходимо удовольствоваться определенным числом разрядов после запятой и ограничить этим точность. В нашем случае $0,375_{10}$ точно соответствует $0,011_2$.

На основе целочисленной и дробной частей получаем искомое двоичное число $10110,011_2$.

2.3. Арифметические операции с фиксированной запятой в двоичной системе

В данной главе описываются арифметические операции с числами с фиксированной запятой. Арифметические операции с фиксированной запятой означают, что в них запятая всегда стоит на фиксированном месте. При этом место, на котором стоит запятая, ориентируется на позицию в ЗУ, на которой находится число. В этом случае нет необходимости реализовать запятую в аппаратуре компьютера. Она существует только в голове программиста. Мы ограничиваемся постоянной длиной слова n , как это имеет место в компьютерах. На основе этого можно обсудить проблему переполнения допустимой области.

2.3.1. Целочисленное сложение в двоичной системе

Целочисленное сложение двух чисел A и B производится в двоичной системе точно так же, как и в десятичной системе — по разрядам. Как и там, в каждом разряде должны быть просуммированы обе двоичных цифры a_n и b_n и перенос из предыдущего разряда C_{n-1} .

При сложении возникают (табл. 2.2) новая сумма S_n и новый перенос C_n .

В этой таблице дискретной линией разделены входные и выходные величины. Например:

$$\begin{array}{r}
 01111110 \\
 + 00110101 \\
 \hline
 = 10110011 \\
 \text{перенос } 1111100
 \end{array}$$

Необходимо следить, чтобы в приведенном выше примере суммировались два числа длиной по 8 бит и чтобы итог тоже имел длину 8 бит, чтобы не было переполнения допустимой зоны.

Таблица 2.2. Сложение в двоичной системе со слагаемыми a_n, b_n и переносом из предыдущего разряда c_{n-1} . Сумма равна S_n , новый перенос C_n .

a_n	b_n	c_{n-1}	c_n	s_n
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

2.3.2. Сложение чисел с фиксированной запятой

В этом случае, когда суммируются два числа с фиксированной запятой, важным моментом является то, чтобы аналогично обычной процедуре в десятичной системе обе запятые стояли друг над другом. Так, при сложении двух чисел длиной в 8 бит запятая у обоих чисел должна стоять, например, на третьем месте.

Например:

$$\begin{array}{r}
 01100,010 \\
 + 00110,111 \\
 \hline
 = 1100,110 \\
 \text{перенос } 10011,001
 \end{array}$$

2.3.3. Представление с помощью обратного кода

Для того, чтобы иметь малые затраты на аппаратную часть (hardware) компьютера, были предприняты усилия по сведению к одному алгоритму вычитания и сложения. Этого можно добиться, если применять двоичные цифры в их дополняющей форме. Различают единичное дополнение (обратный код, поразрядное дополнение) и двойное дополнение (точное дополнение).

Поразрядное дополнение формируется путем замены всех нулей на единицы и обратно. Следовательно, поразрядным дополнением (one's complement) 0001 является 1110. Ниже поразрядное дополнение двоичного числа A обозначено $\neg A$. Очевидно, что при представлении n -битового слова имеем:

$$\neg A + A = 2^n - 1 \quad (2.3)$$

Например, при представлении 8-битового слова имеем:

$$10110011 + 01001100 = 11111111 = 2^8 - 1$$

Можно так преобразовать уравнение (2.3), чтобы получить формулу для расчета поразрядного дополнения:

$$\neg A = 2^n - 1 - A \quad (2.4)$$

2.3.4. Представление с помощью двойного дополнения (точное дополнение, two's complement)

Точное дополнение A_{K2} образуется из поразрядного дополнения $\neg A$ путем прибавления 1:

$$A_{K2} = \neg A + 1 \quad (2.5)$$

Следовательно, с учетом (2.4) будет справедливо:

$$A_{K2} = 2^n - A \quad (2.6)$$

Мы видим, что в данном представлении содержится « $-A$ », благодаря чему оно удобно для проведения вычитания. Учтем также, что 2^n в двоичном представлении имеет $n + 1$ разрядов. Ниже пример точного дополнения для 10101100:

$$A_{K2} = \neg A + 1 = 01010011 + 1 = 01010100$$

Представление 4-битовых двоичных слов в круговой форме, приведенное на рис. 2.1, позволяет показать числовой диапазон. Соответствующее данным значениям наибольшее представляемое положительное число равно 7_D , соответствующее данным значениям наибольшее отрицательное число равно -8_D . Следовательно, числовой диапазон построен несимметрично, поскольку отрицательное число занимает больше места, чем положительное. Наибольшее и наименьшее представляемые числа можно выразить как:

$$z_{max} = 2^{n-1} \quad (2.7)$$

$$z_{min} = -2^{n-1} \quad (2.8)$$

На рис. 2.1 можно видеть, что малые числа, сформированные на основе точного дополнения, содержат много ведущих единиц, когда они отрицательны, и содержат много ведущих нулей, когда они

положительны. Соответствующие данным значениям большие числа, сформированные на основе точного дополнения, содержат далеко влево отстоящий нуль, когда они отрицательны, и далеко влево отстоящий нуль, когда они положительны. Их собственным точным дополнением является число 1000_B (-8_D). Важно установить, что при представлении на основе точного дополнения имеем только один 0. Это облегчает запрос, равен ли результат 0. Напротив, при представлении на основе поразрядного дополнения имеют место двоичное число 0000_B , соответствующее $+0_D$ и двоичное число 1111_B , соответствующее -0_D .

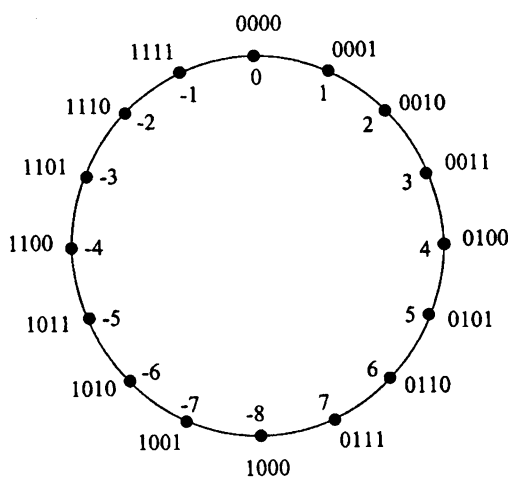


Рис. 2.1. Представление 4-битовых слов с помощью 4-битового дополнения.

2.3.5. Вычитание при представлении с помощью точного дополнения

Пусть должны были вычтены одно из другого два положительных двоичных числа A и B . При условии применения точного дополнения в соответствии с уравнением (2.6) вычитание можно провести следующим образом:

$$A - B = A - B + B_{K2} - B_{K2} = A - B + B_{K2} - (2^n - B) \quad (2.9)$$

Раскрытие скобок в правой части уравнения дает:

$$A - B = A + B_{K2} - 2^n \quad (2.10)$$

Что означает вычитание 2^n ? Поясним это на примере операции вычитания $7 - 3 = 4$ в 4-битовой двоичной системе. Сумма двоичного

эквивалента числа 7 и дополнение двоичного эквивалента числа 3 равняется:

$$\begin{array}{r}
 + \quad 0111 \quad 7_{10} \\
 \quad 1101 \quad -3_{10} \\
 \hline
 = 10100 \\
 \hline
 - 10000 \\
 = 0100 \quad 4_{10}
 \end{array}$$

Вычитание числа 10000_B , проведенное в соответствии с уравнением (2.10), дает правильный результат 0100_B . Это может произойти в 4-битовом компьютере просто потому, что высший результат игнорируется. Итак, при проведении вычитания с помощью точного дополнения нет необходимости учитывать высший перенос c_4 . Но необходимо соблюдать осторожность в связи с переполнением числового диапазона. Исследуем это ниже.

2.3.6. Переполнение числового диапазона

Исходя из вышесказанного возникает необходимость рассмотрения проблемы переполнения числового диапазона (overflow) в связи с представлением на основе точного дополнения. Переполнение числового диапазона может происходить только в двух случаях. А именно, когда суммируются два положительных числа либо суммируются два отрицательных числа. В связи с этим рассмотрим несколько примеров, относящихся к 4-битовому представлению.

- Пример переполнения числового диапазона при сложении двух положительных чисел:

$$\begin{array}{r}
 + \quad 0101 \quad 5_{10} \\
 \quad 0101 \quad 5_{10} \\
 \hline
 = 1010 \quad -6_{10}
 \end{array}$$

Очевидно, что результат является неправильным. Ошибка возникает за счет переноса 3-го разряда на место 4-го разряда, что приводит к симуляции отрицательного числа. Этот перенос c_3 в представлении, использующем n бит, обычно обозначается как c_{n-1} . Перенос c_4 (в общем случае c_n) из разряда 4 в разряд 5 называется *Carry* (*Cy*). В этом примере данный перенос не имеет места.

- Пример переполнения численного диапазона при сложении отрицательных чисел:

$$\begin{array}{r}
 + \quad 1011 \quad -5_{10} \\
 \quad 1011 \quad -5_{10} \\
 \hline
 = (1)0110 \quad 6_{10}
 \end{array}$$

В этом примере также появляется неправильный результат. Имел место не перенос c_{n-1} из разряда 3 в разряд 4, а перенос c_n из разряда 4 в разряд 5.

- Для сравнения проведем сложение двух отрицательных чисел без переполнения числового диапазона:

$$\begin{array}{r}
 + \quad 1111 \quad -1_{10} \\
 \quad 1101 \quad -3_{10} \\
 \hline
 = (1)1100 \quad -4_{10}
 \end{array}$$

Имели место переносы c_n и c_{n-1} .

Сведем эти результаты вместе с другими, здесь не показанными случаями, в таблицу. На основе результатов, представленных в табл. 2.3, для двух двоичных чисел A и B , которые лежат в числовом диапазоне, определяемом n -битовым представлением на базе точного дополнения, можно установить перенос переполнения при сложении.

Таблица 2.3. Перенос переполнения при сложении в случае n -битового представления на основе точного дополнения.

	Правильный результат	Перенос переполнения
$A + B$	$c_n = 0, c_{n-1} = 0$	$c_n = 0, c_{n-1} = 1$
$A - B$	$c_n = c_{n-1}$	невозможен
$-A - B$	$c_n = 1, c_{n-1} = 1$	$c_n = 1, c_{n-1} = 0$

Следовательно, правильный результат имеет место тогда, когда $c_n = c_{n-1}$, неправильный результат — когда $c_n \neq c_{n-1}$.

2.3.7. Умножение

Умножение выполняется так же, как и для десятичной системы. Рассмотрим пример умножения на основе двоичной системы для чисел

после запятой. Так в соответствии с верхним примером имеем:

$$10, 10 \cdot 101, 1 = 1101, 110$$

Деление чисел с точным дополнением также можно свести к умножению и сложению [20].

2.4. Шестнадцатеричный код

На практике наряду с двоичным кодом внедрился шестнадцатеричный код, поскольку он обеспечивает лучшее обозрение длинных двоичных чисел. Шестнадцать шестнадцатеричных цифр определены в табл. 2.4. Шестнадцатеричные цифры больше девяти представлены буквами *A–F*. Для преобразования двоичных в шестнадцатеричные числа объединяют по четыре цифры двоичного числа, которые интерпретируются как шестнадцатеричный разряд. Благодаря этому шестнадцатеричное число занимает только четверть разрядов, занимаемых двоичным числом одинаковой величины.

Например:

$$\begin{array}{c|c|c} 0110 & 1100 & 1111 \\ \hline G & C & F \end{array}$$

Итак, справедливо выражение $0110011001111_2 = GCF_{16}$.

Таблица 2.4. Шестнадцатеричные числа.

десятичные	двоичные	шестнадцатеричные	десятичные	двоичные	шестнадцатеричные
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

В качестве обозначения шестнадцатеричного числа используется индекс *H*. Преобразование шестнадцатеричного числа в десятичное число и обратно проще всего производить через соответствующее двоичное число. Также возможно производить преобразование с помощью алгоритма, как при преобразовании двоичного числа в

десятичное число. Обратное преобразование производится аналогично представленному уравнением (2.2).

2.5. Восьмеричный код

Восьмеричный код применяется аналогично шестнадцатеричному, только объединяются лишь по 3 разряда двоичного числа. Для восьмеричного кода применяются цифры десятичного кода от 0 до 7, он обозначается индексом 0.

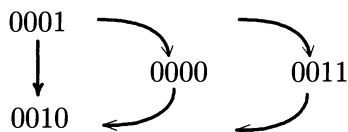
Например:

$$\begin{array}{c|c|c|c} 1 & 1 & 0 & 1 \\ \hline 6 & 5 & 4 & 3 \end{array}$$

Следовательно, справедливо $11010100011_2 = 6543_8$.

2.6. Код Грея

Часто в цифровой технике требуется для числового кода схема кодирования, в соответствии с которой при переходе от одного числа к следующему изменялась бы только одна цифра. Выполнение этого условия необходимо, когда вследствие технических неточностей момент переключения не может быть точно выдержан. При переключении двух цифр из-за этого могут возникнуть неправильные коммутационные операции. В качестве примера подобной ошибки рассмотрим переключение от 1_{10} к 2_{10} в двоичном коде:



При этом переключении изменяются биты 0 и 1 при одновременном переключении непосредственно достигается новое число. Если вначале изменяется бит 0, то появляется число 0000, и только когда изменяется бит 1, получаем правильное число 0010. Если же сначала изменяется бит 1 и потом изменяется бит 0, то в промежутке возникает число 0011. Коды Грея позволяют избежать этой очень серьезной ошибки за счет того, что при переходе от одного кодового слова к следующему изменяется только один разряд. В табл. 2.5 представлен 4-разрядный код Грея. В дополнение к вышесказанному показанный код имеет свойство, которое заключается в его цикличности, так как также и при переходе от высшего числа (15D)

низшему числу изменяется только один разряд. Циклические коды Грея могут быть сконструированы для всех прямых длин периодов.

Таблица 2.5. Пример 4-разрядного кода Грея.

Десятичные (числа)	Код Грея	Десятичные (числа)	Код Грея
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

2.7. Двоично – десятичный код

Если мы хотим подвести к десятичным цифрам некоторые отметки, то для этого подойдет код, в котором отдельным десятичным цифрам приданы двоично-кодированные кодовые слова. Этот код обозначается как двоично-десятичный код (BCD-код). Подходящая возможность реализации заключается в том, чтобы представить через соответствующую каждой из них 4-разрядную двоичную цифру. Поскольку отдельные разряды имеют веса 8, 4, 2 и 1, данный код называют кодом типа 8-4-2-1. Существует также возможность построить двоично-десятичный код 2, 4, 2, 1 (Aiken-Code). К другим двоично-десятичным кодам относятся код с избытком три (3-Excess-Code) и двоично-десятичный код Грея (BCD-Gray-Code) [5].

Таблица 2.6. Двоично-десятичный код.

Десятичная цифра	Код типа 8-4-2-1	Десятичная цифра	Код типа 8-4-2-1
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

2.8. Алфавитно – цифровые коды

Существует большое количество кодов, осуществляющих представление алфавитно-цифровых символов с помощью двоичных цифр. Известным примером подобных символов является код ASCII (American Standard Code for Information Interchange), который содержит также ряд управляющих символов.

Таблица 2.7. Код ASCII (ч.1, управляющие символы)

ASCII		ASCII	
00		10	
01		11	
02		12	
03		13	
04		14	
05		15	
06		16	
07		17	
08		18	
09		19	
0A		1A	
0B		1B	
0C		1C	
0D		1D	
0E		1E	
0F		1F	

2.9. Упражнения

Задача 2.1. Преобразуйте следующие двоичные числа в десятичные числа:

- а) 1110, 101
- б) 10011, 1101

Задача 2.2. Преобразуйте следующие десятичные числа в двоичные числа:

- а) 22, 125
- б) 45, 33

Задача 2.3. Рассчитайте приведенные ниже примеры с использованием точных дополнений при длине слова в 6 бит. Укажите, имеет ли место переполнение числового диапазона.

а) 010101 – 001010

б) –010111 – 011011

Задача 2.4. Рассчитайте в двоичной системе:

а) 110101 · 010101

б) 1101110 : 110

Задача 2.5. Разработайте циклический код Грея с длиной периода 6.

Таблица 2.8. Код ASCII (ч.2).

ASCII		ASCII		ASCII		ASCII		ASCII		ASCII	
20	SP	30	0	40	@	50	P	60	'	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	M	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E		6E	n	7E	"
2F	/	3F	?	4F	O	5F	-	6F	o	7F	DEL

ГЛАВА 3

ПЕРЕКЛЮЧАТЕЛЬНАЯ АЛГЕБРА

Цифровая техника имеет то преимущество перед аналоговой техникой, что она основывается на относительно простой, но при этом мощной теории, на Булевой алгебре, называемой также переключательной алгеброй. В данной главе представлены теоретические основы цифровой техники. Булева алгебра может быть применена для решения почти всех возникающих при разработке цифровых схем проблем при условии выполнения технологических предпосылок, которые рассматриваются в главе 4.

3.1. Переключательная переменная и переключательная функция

В цифровой технике применяют специальные переменные и функции. Под Булевой переменной понимают переменную, которая может принимать только значения 0 и 1. С Булевыми переменными могут быть образованы функции. Функцию:

$$y = f(x_1, x_2, x_3, \dots x_n) \quad \text{при } x_i, y \in \{0, 1\} \quad (3.1)$$

называют n -разрядной переключательной или двоичной функцией. Область значений подобной функции также определяется двоичной системой счисления с элементами 0 и 1. Функции могут быть определены таблицами, в которых величины функций связываются с возможными 2^n комбинациями n входных переменных. Эти таблицы называются таблицами истинности.

Очень простая функция, которая связывает входную переменную x с выходной переменной y , представлена в табл. 3.1. Можно видеть, что все переключательные функции могут быть определены таблицей, в которой содержатся все входные переменные, поскольку учитывать необходимо только два элемента 0 и 1.

Определенную табл. 3.1 переключательную функцию $y = f(x)$ именуют «отрицание», «дополнение» или НЕТ (NOT). В последующем она обозначается оператором « $\neg$ » и читается « y равен не x »:

$$y = \neg x. \quad (3.2)$$

Таблица 3.1. Таблица истинности инвертора:

x	y
0	1
1	0

«Отрицание» является одноразрядной переключательной функцией, поскольку она обладает только одним входным символом. Схемный элемент «инвертор» обозначается с помощью схемного обозначения, представленного на рис. 3.1:

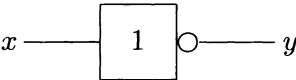


Рис. 3.1. Схемное обозначение инвертора.

Имеются ли еще одноразрядные переключательные функции? Путем систематических проб находим их общим числом 4, все они сведены в табл. 3.2. Других видов одноразрядной переключательной функции y не существует. Переключательные функции $y = 0$ и $y = 1$ вырабатывают постоянные, которые не зависят от входа. Поэтому существенной для переключательной алгебры является только одноразрядная двоичная функция $y = \neg x$.

Таблица 3.2. Одноразрядные двоичные функции.

Таблица истинности	Функции	Схемное обозначение	Наименование						
<table><tr><td>x</td><td>y</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td></tr></table>	x	y	0	0	1	0	$y = 0$		
x	y								
0	0								
1	0								
<table><tr><td>x</td><td>y</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	y	0	0	1	1	$y = x$		
x	y								
0	0								
1	1								
<table><tr><td>x</td><td>y</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	y	0	1	1	0	$y = \neg x$		НЕТ, «дополнение», «отрицание», «инверсия»
x	y								
0	1								
1	0								
<table><tr><td>x</td><td>y</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	x	y	0	1	1	1	$y = 1$		
x	y								
0	1								
1	1								

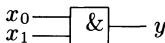
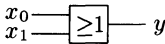
3.2. Двухразрядные переключательные функции.

В принципе можно образовывать произвольные двоичные функции многих входных переменных. Но оказалось практичным вначале рассматривать только функции с одной или двумя входными переменными, а функции с большим количеством входных переменных сводить к ним.

Двоичную функцию с входными переменными x_0 и x_1 также можно определить с помощью таблицы. Комбинацию входных переменных x_0 и x_1 можно представить в виде вектора $X = [x_0, x_1]$. Двум входным переменным соответствуют 4 возможных входных вектора X , которые часто индицируются через их десятичные эквиваленты. Так x_2 означает, что $x_1 = 1$ и $x_0 = 0$, или, выражаясь по-другому, что $x_2 = [x_0, x_1] = [0, 1]$.

Наряду с «отрицанием» технически возможными являются основные логические элементы И или ИЛИ, которые определяются табл. 3.3. Также обозначают И как AND или «конъюнкция»; ИЛИ как OR или «дизъюнкция».

Таблица 3.3. Основные логические элементы И и ИЛИ.

Таблица истинности	Функции	Схемное обозначение	Наименование															
<table><tr><th>x_1</th><th>x_0</th><th>y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x_1	x_0	y	0	0	0	0	1	0	1	0	0	1	1	1	$y = x_0 \wedge x_1$		И, AND, «КОНЪЮНКЦИЯ»
x_1	x_0	y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
<table><tr><th>x_1</th><th>x_0</th><th>y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1	x_0	y	0	0	0	0	1	1	1	0	1	1	1	0	$y = x_0 \vee x_1$		ИЛИ, OR, «ДИЗЪЮНКЦИЯ»
x_1	x_0	y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Возникает вопрос о других возможных двухразрядных двоичных функциях. Чтобы системно ответить на этот вопрос, можно осуществить перестановку значений входных переменных y , которые можно получить из четырех возможных входных векторов. В общем случае функция $y(x_1, x_0)$ может быть определена таблицей истинности (табл. 3.4).

На основе этой таблицы можно сделать заключение о возможности образования $2^4 = 16$ различных двоичных функций с двумя

входными переменными. Все возможные двухразрядные двоичные функции приведены в табл. 3.6. Представление двоичных функций выполнено в соответствии со стандартом DIN [43].

Таблица 3.4. Таблица истинности для двухразрядной двоичной функции.

x_1	x_0	y
0	0	$y(0, 0)$
0	1	$y(0, 1)$
1	0	$y(1, 0)$
1	1	$y(1, 1)$

Технически важными функциями NAND, NOR, «эквивалентность» и EXOR (также «исключительное ИЛИ», дизъюнкция) соответствуют собственным логическим символам. На практике они часто реализуются на основе специальных схем.

В этой таблице показано, что отдельные функции могут быть представлены только с помощью логических элементов AND, OR и NOT. Поэтому каждая логическая функция может быть представлена с помощью этих трех логических элементов. Все двоичные функции также могут быть представлены с помощью функции NOR, точно также и с помощью NAND. Поэтому эти функции называют совершенными.

Доказательство эквивалентности можно привести путем размещения таблиц истинности. Таким образом, например, может быть доказана эквивалентность для логического элемента EXOR на основе AND, OR или NOT:

$$x_0 \leftrightarrow x_1 = (\neg x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1) \quad (3.3)$$

В табл. 3.5 вначале оцениваются оба выражения в скобках. Затем формируется логическое OR, соответствующее обоим выражениям в скобках, и вписывается в пустую колонку. Так как последняя и предпоследняя колонки совпадают, уравнение доказано, поскольку в последней колонке стоит определение функции «исключительное ИЛИ» (то есть EXOR).

Таблица 3.5. Доказательство на основе таблицы истинности.

x_1	x_0	$\neg x_0 \wedge x_1$	$x_0 \wedge \neg x_1$	$(\neg x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1)$	$x_0 \leftrightarrow x_1$
0	0	0	0	0	0
0	1	0	1	1	1
1	0	1	0	1	1
1	1	0	0	0	1

Таблица 3.6. Двухразрядные двоичные функции: таблица истинности, представление на основе (AND, NOT, OR), логический символ и название функции.

Таблица истинности	Функции	Схемное обозначение	Наименование										
<table><tr><td>x_0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>x_1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	x_0	1	0	1	0	x_1	1	1	0	0			
x_0	1	0	1	0									
x_1	1	1	0	0									
<table><tr><td>y</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	y	0	0	0	0	$y = 0$		нуль					
y	0	0	0	0									
<table><tr><td>y</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	y	0	0	0	1	$y = \neg(x_0 \vee x_1)$ $y = (x_0 \bar{\vee} x_1)$	x_0 — <table><tr><td>≥ 1</td></tr></table> — y x_1 —	≥ 1	NOR				
y	0	0	0	1									
≥ 1													
<table><tr><td>y</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	y	0	0	1	0	$y = x_0 \wedge \neg x_1$		запрет					
y	0	0	1	0									
<table><tr><td>y</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table>	y	0	0	1	1	$y = \neg x_1$		дополнение					
y	0	0	1	1									
<table><tr><td>y</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	y	0	1	0	0	$y = \neg x_0 \wedge x_1$		запрет					
y	0	1	0	0									
<table><tr><td>y</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>	y	0	1	0	1	$y = \neg x_0$		дополнение					
y	0	1	0	1									
<table><tr><td>y</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	y	0	1	1	0	$y = (\neg x_0 \wedge x_1) \vee$ $\vee (x_0 \wedge \neg x_1)$ $y = (x_0 \leftrightarrow x_1)$	x_0 — <table><tr><td>$= 1$</td></tr></table> — y x_1 —	$= 1$	EXOR				
y	0	1	1	0									
$= 1$													
<table><tr><td>y</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	y	0	1	1	1	$y = \neg(x_0 \wedge x_1)$ $y = (x_0 \bar{\wedge} x_1)$	x_0 — <table><tr><td>$\&$</td></tr></table> — y x_1 —	$\&$	NAND				
y	0	1	1	1									
$\&$													
<table><tr><td>y</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	y	1	0	0	0	$y = x_0 \wedge x_1$	x_0 — <table><tr><td>$\&$</td></tr></table> — y x_1 —	$\&$	AND				
y	1	0	0	0									
$\&$													
<table><tr><td>y</td><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>	y	1	0	0	1	$y = (x_0 \wedge x_1) \vee$ $\vee (\neg x_0 \wedge \neg x_1)$ $y = (x_0 \leftrightarrow x_1)$	x_0 — <table><tr><td>$=$</td></tr></table> — y x_1 —	$=$	эквивалентность				
y	1	0	0	1									
$=$													
<table><tr><td>y</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	y	1	0	1	0	$y = x_0$		идентичность					
y	1	0	1	0									
<table><tr><td>y</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	y	1	0	1	1	$y = x_0 \vee \neg x_1$		импликация					
y	1	0	1	1									
<table><tr><td>y</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	y	1	1	0	0	$y = x_1$		идентичность					
y	1	1	0	0									
<table><tr><td>y</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	y	1	1	0	1	$y = x_1 \vee \neg x_0$		импликация					
y	1	1	0	1									
<table><tr><td>y</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	y	1	1	1	0	$y = x_0 \vee x_1$	x_0 — <table><tr><td>≥ 1</td></tr></table> — y x_1 —	≥ 1	OR				
y	1	1	1	0									
≥ 1													
<table><tr><td>y</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	y	1	1	1	1	$y = 1$		единица					
y	1	1	1	1									

3.3. Вычислительные правила

Важными для упрощения сложных функций являются вычислительные правила булевой алгебры. Для доказательства отдельных законов могут быть использованы таблицы истинности.

Вычислительные правила булевой алгебры перечислены ниже.
Переместительный (коммутативный) закон:

$$x_0 \wedge x_1 = x_1 \wedge x_0 \quad (3.4)$$

$$x_0 \vee x_1 = x_1 \vee x_0 \quad (3.5)$$

Сочетательный (ассоциативный) закон:

$$(x_0 \wedge x_1) \wedge x_2 = x_0 \wedge (x_1 \wedge x_2) \quad (3.6)$$

$$(x_0 \vee x_1) \vee x_2 = x_0 \vee (x_1 \vee x_2) \quad (3.7)$$

Распределительный закон (закон дистрибутивности):

$$x_0 \wedge (x_1 \vee x_2) = (x_0 \wedge x_1) \vee (x_0 \wedge x_2) \quad (3.8)$$

$$x_0 \vee (x_1 \wedge x_2) = (x_0 \vee x_1) \wedge (x_0 \vee x_2) \quad (3.9)$$

Закон поглощения (закон абсорбции):

$$x_0 \wedge (x_0 \vee x_1) = x_0 \quad (3.10)$$

$$x_0 \vee (x_0 \wedge x_1) = x_0 \quad (3.11)$$

Существование нейтральных элементов:

$$x_0 \wedge 1 = x_0 \quad (3.12)$$

$$x_0 \vee 0 = x_0 \quad (3.13)$$

Существование дополнительных элементов:

$$x_0 \wedge \neg x_0 = 0 \quad (3.14)$$

$$x_0 \vee \neg x_0 = 1 \quad (3.15)$$

Теорема Де-Моргана:

$$x_0 \wedge x_1 = \neg(\neg x_0 \vee \neg x_1) \quad (3.16)$$

$$x_0 \vee x_1 = \neg(\neg x_0 \wedge \neg x_1) \quad (3.17)$$

На основе симметрии законов можно сделать следующие выводы.

Если справедлив один закон, то справедлив и закон, который получают путем взаимной замены AND и OR и постоянных 0 и 1. Полученный таким образом закон называют дуальным законом. Как, например, законы (3.16) и (3.17) дуальны по отношению друг к другу. Аналогично обозначают функцию F' , которая получилась из

функции F путем замены AND на OR и 0 на 1, как функцию, дуальную по отношению к F .

Важно также установить, что NAND и NOR не являются ассоциативными. И так, справедливо:

$$(x_0 \bar{\wedge} x_1) \bar{\wedge} x_2 \neq x_0 \bar{\wedge} (x_1 \bar{\wedge} x_2) \quad (3.18)$$

$$(x_0 \bar{\vee} x_1) \bar{\vee} x_2 \neq x_0 \bar{\vee} (x_1 \bar{\vee} x_2) \quad (3.19)$$

3.4. Каноническая дизъюнктивная нормальная форма (KDNF).

Любую двоичную функцию можно представить с использованием только логических элементов AND, OR или NOT. Это может быть выполнено на основе систематического подхода, как в примере с функциями, приведенными в табл. 3.7. Можно действовать двумя различными способами. Мы начнем с канонической нормальной дизъюнктивной формы (KDNF).

Таблица 3.7. Таблица истинности для примера с KDNF.

x_2	x_1	x_0	Десятичный	y
0	0	0	0	1
0	0	1	1	0
0	1	0	2	1
0	1	1	3	1
1	0	0	4	0
1	0	1	5	1
1	1	0	6	1
1	1	1	7	0

Для этого рассмотрим сначала входные векторы x_i , для которых функция $y = f(x)$ принимает значение 1. Следовательно, для этих входных векторов справедливо $f(x_i) = 1$. В нашем случае это x_0 , x_2 , x_3 , x_5 и x_6 . Проведем для каждого подобного входного вектора операцию конъюнкции (логического умножения, И) с элементом x_i , который как раз для этого входного вектора принимает значение 1. Для x_5 было бы:

$$m_5 = x_2 \wedge \neg x_1 \wedge x_0 \quad (3.20)$$

m_5 называют также «минтермом». Минтермы содержат всегда все входные переменные, поэтому их называют полной конъюнкцией.

В случае минтерма, входные переменные могут быть инвертированными либо не инвертированными, в зависимости от того, что представляет собой переменная 1 или 0. В этом примере входные минтермы имеют вид:

$$m_0 = \neg x_2 \wedge \neg x_1 \wedge \neg x_0 \quad (3.21)$$

$$m_2 = \neg x_2 \wedge x_1 \wedge \neg x_0 \quad (3.22)$$

$$m_3 = \neg x_2 \wedge x_1 \wedge x_0 \quad (3.23)$$

$$m_6 = x_2 \wedge x_1 \wedge \neg x_0 \quad (3.24)$$

Следовательно, при определенном варианте входных переменных минтерм имеет значение 1.

Вся функция может быть представлена на основе дизъюнкции (логическое сложение) минтермов. Функция получает значение 1, когда, по крайней мере, один из минтермов равен 1. Этот способ представления называется «канонической дизъюнктивной нормальной формой» (KDNF). В нашем случае функция может быть представлена следующим образом:

$$y = (\neg x_2 \wedge \neg x_1 \wedge \neg x_0) \vee (\neg x_2 \wedge x_1 \wedge \neg x_0) \vee (\neg x_2 \wedge x_1 \wedge x_0) \vee (x_2 \wedge \neg x_1 \wedge x_0) \vee (x_2 \wedge x_1 \wedge \neg x_0) \quad (3.25)$$

3.5. Каноническая конъюнктивная нормальная форма (KKNF)

В качестве альтернативы для представления функции могут быть применены входные векторы x_i , при которых функция принимает значение 0, то есть когда будет справедливо равенство $f(x_i) = 0$. Для функции показанной в табл. 3.7, такими векторами будут x_1 , x_4 и x_7 .

Сформируем так называемые макстермы. Это дизъюнкции, которые равны 0, когда приложен соответствующий входной вектор x_i :

$$M_1 = x_2 \vee x_1 \vee \neg x_0 \quad (3.26)$$

$$M_4 = \neg x_2 \vee x_1 \vee x_0 \quad (3.27)$$

$$M_7 = \neg x_2 \vee \neg x_1 \vee \neg x_0 \quad (3.28)$$

Итак, входные переменные, которые в входном векторе равны 1, выступают в макстерме инвертированы. Входные переменные, которые в входном векторе равны 0, появляются в макстерме в неинвертированном виде. Таким образом, макстерм равен 0 только для $x_2 = 0$, $x_1 = 0$ и $x_0 = 0$.

Вся функция может быть представлена теперь на основе конъюнкции макстермов, так как значение функции только тогда равно 0, когда, по крайней мере, один из макстермов равен 0. Форма представления, называемая как «каноническая конъюнктивная нормальная форма» (KKNF), представлена в следующем примере:

$$y = (x_2 \vee x_1 \vee \neg x_0) \wedge (\neg x_2 \vee x_1 \vee x_0) \wedge (\neg x_2 \vee \neg x_1 \vee \neg x_0) \quad (3.29)$$

3.6. Представление функций с помощью KKNF и KDNF.

На практике часто возникает вопрос, как от конкретной проблемы перейти к необходимым для ее решения переключательным функциям. В связи с этим рассмотрим в качестве примера функцию «четность» (parity) f_p , представленную в табл. 3.8. Должна быть реализована схема с тремя входами, которая на выходе y выдает 1 тогда, когда четное число входных сигналов равно 1. В качестве первой операции установим таблицу истинности для функции $y = f_p(x_2, x_1, x_0)$. Затем рассмотрим все комбинации входных сигналов, для которых входной сигнал должен быть равен 1. В данном конкретном случае этому соответствуют варианты комбинаций входных сигналов, в которых содержатся две 1 или ни одной. Этим исчерпываются все возможные случаи. В табл. 3.8 даны дополнительно все десятичные эквиваленты входных векторов.

Таблица 3.8. Таблица истинности для приведенной в качестве примера функции «четность» $y = f_p(x_2, x_1, x_0)$.

x_2	x_1	x_0	Десятичные эквиваленты	y
0	0	0	0	1
0	0	1	1	0
0	1	0	2	0
0	1	1	3	1
1	0	0	4	0
1	0	1	5	1
1	1	0	6	1
1	1	1	7	0

Затем формируем KDNF. Нам потребуются минтермы m_i , соответствующие входным векторам с десятичными эквивалентами 6,

5, 3, 0. Эти минтермы связываются через логическое ИЛИ. В этом примере KDNF имеет вид:

$$y = (x_2 \wedge x_1 \wedge \neg x_0) \vee (x_2 \wedge \neg x_1 \wedge x_0) \vee (\neg x_2 \wedge x_1 \wedge x_0) \vee (\neg x_2 \wedge \neg x_1 \wedge \neg x_0) \quad (3.30)$$

Соответствующая логическая схема содержит 4 вентили И, соединенные с четырехходовым вентилям ИЛИ.

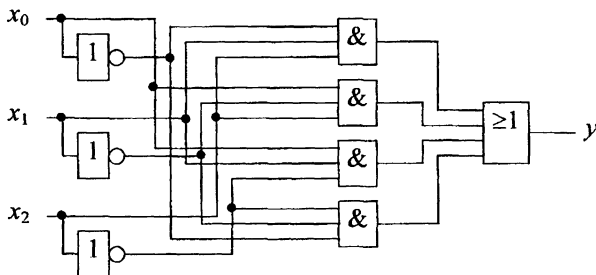


Рис. 3.2. Логическая схема реализации KDNF, соответствующей функции «четность».

KKNF образуют макстермы с десятичными эквивалентами 1, 2, 4, 7. Их связывают логические вентили И. Соответствующая KKNF имеет следующий вид:

$$y = (x_2 \vee x_1 \vee \neg x_0) \wedge (x_2 \vee \neg x_1 \vee x_0) \wedge (\neg x_2 \vee x_1 \vee x_0) \wedge (\neg x_2 \vee \neg x_1 \vee \neg x_0) \quad (3.31)$$

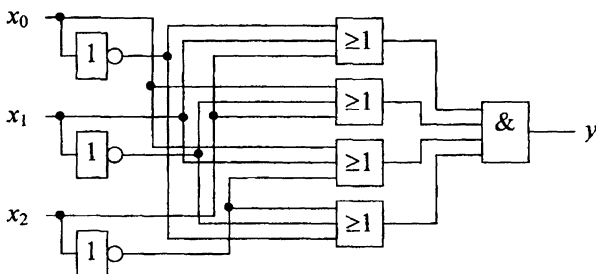


Рис. 3.3. Логическая схема реализации KKNF, соответствующей функции «четность».

KKNF и KDNF являются равнозначными формами представления функции. Но зачастую они имеют различную сложность, так как число минтермов обуславливается числом входных векторов, при которых функция принимает значение 1, в то время как число

макстермов определяется числом входных векторов, для которых функция принимает значение 0. В представленном случае KKNF и KDNF одинаковы по затратам на их формирование.

Для работы с нормальными формами важным является обобщение законов Де-Моргана, так называемое, «правило Шеннона», которое гласит:

Для любой булевой функции

$$y = f(x_0, x_1, \dots, x_n, \wedge, \vee, \leftrightarrow, \leftrightarrow, 1, 0)$$

существует инвертированная функция

$$\neg y = f(\neg x_0, \neg x_1, \dots, \neg x_n, \vee, \wedge, \leftrightarrow, \leftrightarrow, 0, 1)$$

Это означает, что все переменные должны быть проинвертированы, а все операции должны быть заменены на дуальные им. Например, пусть дана функция:

$$y = (x_2 \vee x_1 \vee \neg x_0) \wedge (x_2 \vee \neg x_1 \vee x_0).$$

Тогда имеем инвертированную функцию следующего вида:

$$\neg y = (\neg x_2 \wedge \neg x_1 \wedge x_0) \vee (\neg x_2 \wedge x_1 \wedge \neg x_0).$$

В соответствии с этим правилом KKNF можно сформировать и тогда, когда KDNF определяет инверсную функцию. Для приведенного выше примера получаем:

$$\begin{aligned} \neg y &= (\neg x_2 \neg \wedge x_1 \wedge x_0) \vee (\neg x_2 \wedge x_1 \wedge \neg x_0) \vee \\ &\vee (x_2 \wedge \neg x_1 \wedge \neg x_0) \vee (x_2 \wedge x_1 \wedge x_0) \end{aligned} \quad (3.32)$$

Затем применяет правило Шеннона и напрямую получает KKNF:

$$\begin{aligned} y &= (x_2 \vee x_1 \vee \neg x_0) \wedge (x_2 \vee \neg x_1 \vee x_0) \wedge \\ &\wedge (\neg x_2 \vee x_1 \vee x_0) \wedge (\neg x_2 \vee \neg x_1 \vee \neg x_0) \end{aligned} \quad (3.33)$$

3.7. Минимизация с помощью переключательной алгебры

Нормальные формы KKNF и KDNF подходят главным образом для составления булевых уравнений. Но с точки зрения затрат по количеству вентилях эти формы не идеальны. Для упрощения очень

хорошо подходит уравнение тождества, которое может быть выведено следующим образом:

$$\begin{aligned}(x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1) &= x_0 \wedge (x_1 \vee \neg x_1) = \\ &= x_0 \wedge 1 \\ &= x_0\end{aligned}$$

Следовательно, справедливо:

$$(x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1) = x_0 \quad (3.34)$$

В соответствии с правилами дуализма:

$$(x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) = x_0 \quad (3.35)$$

Пример. Пусть необходимо минимизировать следующую функцию:

$$\begin{aligned}y &= (x_0 \wedge \neg x_1 \wedge x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge \neg x_2 \wedge x_3) \vee \\ &\vee (\neg x_0 \wedge x_1 \wedge x_2 \wedge x_3) \vee (\neg x_0 \wedge x_1 \wedge \neg x_2 \wedge x_3)\end{aligned}$$

Мы видим, что могут быть объединены, например, термы 1 и 2, 2 и 3, а также 4 и 5. Сначала объединяют два первых терма, сохранив второй, так как он еще понадобится для объединения с третьим термом:

$$\begin{aligned}y &= (x_0 \wedge x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge \neg x_2 \wedge x_3) \vee \\ &\vee (\neg x_0 \wedge x_1 \wedge x_2 \wedge x_3) \vee (\neg x_0 \wedge x_1 \wedge \neg x_2 \wedge x_3)\end{aligned}$$

$$y = (x_0 \wedge x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge x_3) \vee (\neg x_0 \wedge x_1 \wedge x_3)$$

Могут быть объединены также оба последних терма:

$$y = (x_0 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_3)$$

Это представление является минимальным. Для реализации требуются два вентиля AND и один вентиль OR. Графический метод минимизации представлен в главе 6.

3.8. Способ упрощенной записи

Сложные функции не всегда легко прочесть:

$$f(x_3, x_2, x_1, x_0) = (\neg x_0 \wedge \neg x_2) \vee (x_2 \wedge x_3) \vee (x_0 \wedge \neg x_1 \wedge x_2) \vee (x_1 \wedge x_3) \quad (3.36)$$

Для упрощения записи, который можно опустить символ конъюнкции и скобки. При этом принимается условие, что сначала формируется конъюнкция, а в заключении дизъюнкция. На основании этого можно записать уравнение (3.36) следующим образом:

$$f(x_3, x_2, x_1, x_0) = \neg x_0 \neg x_2 \vee x_2 x_3 \vee x_0 \neg x_1 x_2 \vee x_1 x_3 \quad (3.37)$$

3.9. Символическое обозначение логических элементов

Применяемые символы логических элементов цифровой техники, приведенные в данной книге, соответствуют системе DIN 40900. В соответствии с этой системой предписаний, вначале закругленные символы логических элементов были заменены на новые прямоугольные, так как существовала уверенность в том, что закругленное схемное обозначение не может быть нарисовано с использованием компьютера. Между тем в предписание была введена индексация зависимостей, которая позволяла считывать со структурной схемы поведение цифровых схем. Ниже дано короткое введение в применяемую систематику. В отдельных главах применяемые символы будут поясняться при их появлении. В приложении будет приведена объединенная таблица.

3.9.1. Основы структуры символов

Символы имеют обрамление, в котором в верхней части находится знак, показывающий основную функцию схемы (рис. 3.4). В рассмотренных выше символах ими были знаки $\&$, ≥ 1 , $= 1$, 1 . В приложении можно найти все возможные символы, сведенные в таблицу.

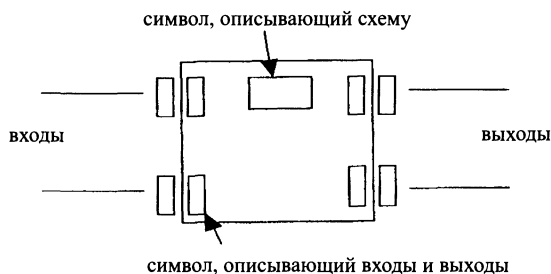


Рис. 3.4. Обобщенная структура символа логического элемента.

Как правило, входы находятся слева, выходы размещены справа. При отклонении от правил следует обозначать направление сигналов стрелками.

На рис. 3.4 обозначены также места, в которые с помощью дополнительных обозначений могут быть введены более точные данные о входах и выходах. Внутри обрамления благодаря этому можно записать замечания о внутреннем логическом состоянии схемы. Вне обрамления стоят такие обозначения как инверсионная петля логических состояний, инверсионный треугольник для уровней (уровни рассматриваются в следующей главе 4), или замечания о виде сигналов. Таблицы возможных символов приведены в приложении.

Если схема содержит общий управляющий блок, то это обозначается как на рис. 3.5 а. Общий выходной блок обозначается двумя двойными линиями, как представлено на рис. 3.5 б.

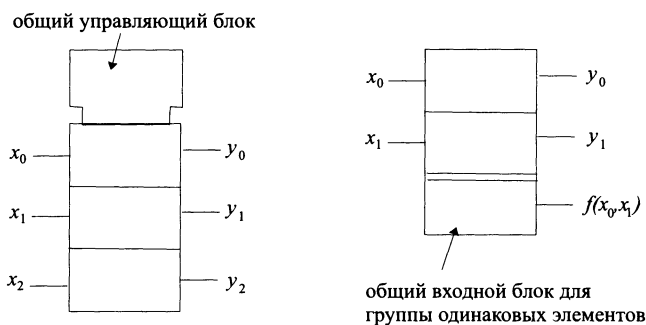


Рис. 3.5. Обобщенная структура символических обозначений логических элементов.

3.9.2. Индексация зависимости

При индексации зависимости с помощью одной буквы описывается влияние одного из входов (или выходов) на другие входы или выходы. Эта буква определяет влияние более строго. За буквой следует идентификационное число. То же самое число находим и у входов и выходов, на которые оказывается влияние. Точнее все это разъясняется на следующих примерах.

3.9.3. Зависимость вида И (G)

Буква G, представленная на одном из входов, может означать зависимость, соответствующая логической функции И. На рис. 3.6 на входе x для более точного описания проставлено обозначение G1. Поскольку вход x_0 обозначен через 1, то он оказывается связанным

с входом x_1 , логической функцией И. Вход x_2 обозначен через $\neg 1$. Исходя из этого, он оказывается связанным функцией И с входом x_1 в случае выполнения над последним операции НЕ. Индекс относится к внутренним состояниям. Возможные инверсионные петли будут обсуждены позже. Они определяют внешнее поведение.

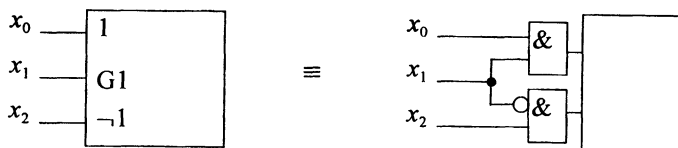


Рис. 3.6. Зависимость по типу И (G).

Как показано на рис. 3.7, индексация зависимости может быть применена и для выходов.

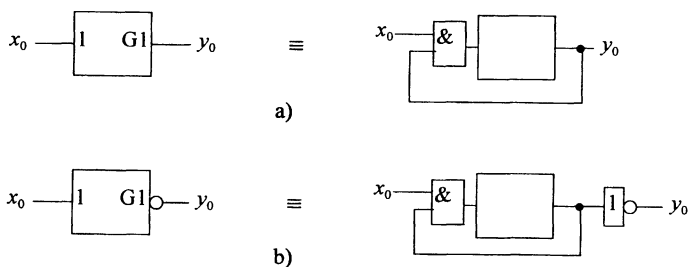


Рис. 3.7. Зависимость по типу И (G), применяемая на выходе. а) без инверсии; б) с инверсией выхода.

Если два входа имеют одинаковые обозначения (рис. 3.8), то эти входы связаны посредством логического ИЛИ.

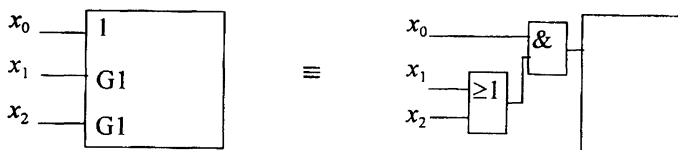


Рис. 3.8. Зависимость по типу И (G) для двух входов, которые обозначены с помощью G1.

3.9.4. Зависимость по типу ИЛИ (V)

Если отмеченный индексом V_n вход или выход имеет внутреннее состояние 1, то в этом случае все входы и выходы, обозначенные числом n , имеют значение 1. Если обозначенный через V_n входной

или выходной сигнал имеет значение 0, то в этом случае сигналы на подвергающихся влиянию входах и выходах имеют нормально определенные значения. Два примера приведены на рис. 3.9.

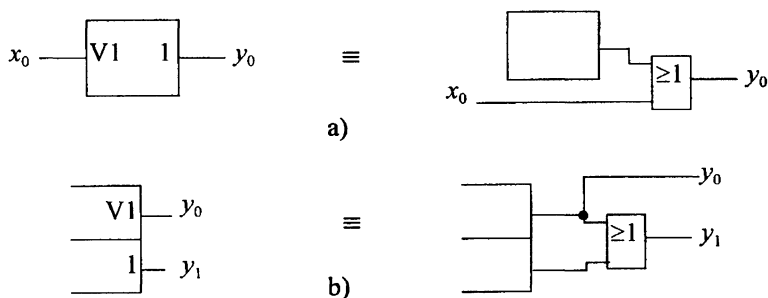


Рис. 3.9. Зависимость по типу ИЛИ (V).

3.9.5. Зависимость по типу EXOR (N).

Обозначенные через N_n входы и выходы состоят с подвергаемыми их влиянию входами и выходами в отношениях, определяемых функцией EXOR. Если на помеченном N_n входе или выходе 1, то на обозначенных через n входах и выходах сигнал инвертируется. В прочих случаях они не подвергаются влиянию.

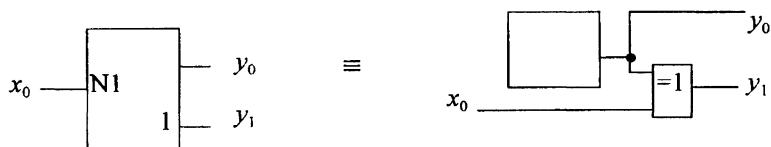


Рис. 3.10. Зависимость по типу EXOR (N).

3.9.6. Образующая соединение зависимость (Z)

Вход или выход, индексированный буквами Z_n , должен оказаться связанным со всеми входами и выходами, обозначенными через n (рис. 3.11).

3.9.7. Зависимость с передачей (X)

Если вход или выход, индексированный буквами X_n , имеет значение 1, то все входы и выходы, обозначенные через n , оказываются связанными двунаправленно (рис. 3.12). В других случаях обозначенные через n входы и выходы изолированы друг от друга.

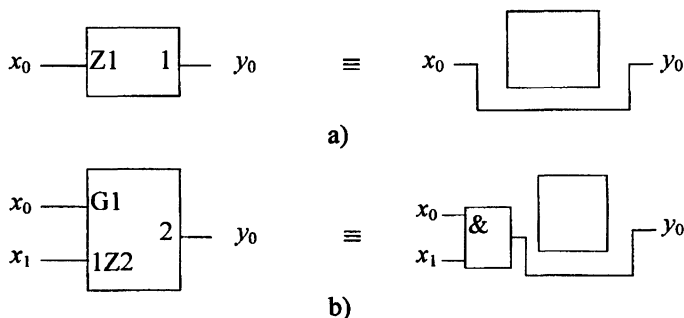


Рис. 3.11. Образующая соединение зависимость (Z).

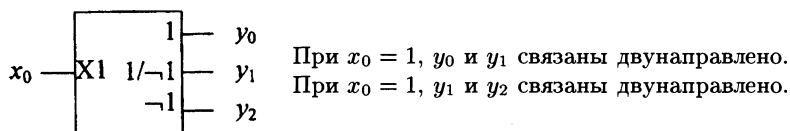


Рис. 3.12. Зависимость с передачей (X).

Дальнейшие зависимости (C, S, R, EN, M, A) описываются в соответствующих главах в приложении.

3.10. Упражнения

Задача 3.1. Докажите обобщенные законы (3.10) и (3.11) с помощью таблицы истинности.

Задача 3.2. Минимизируйте следующую функцию с помощью Булевой алгебры:

$$\begin{aligned}
 y = & (x_0 \wedge x_1 \wedge x_2 \wedge \neg x_3) \vee (x_0 \wedge x_1 \wedge x_2 \wedge x_3) \vee \\
 & \vee (\neg x_0 \wedge \neg x_1 \wedge x_2 \wedge x_3) \vee (\neg x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3) \vee \\
 & \vee (x_0 \wedge \neg x_1 \wedge x_2 \wedge x_3) \vee (x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3)
 \end{aligned}$$

Задача 3.3. Сформируйте KKNF и KDNF для системы с входными переменными a , b и c , у которой на входах S_1 и S_0 выдается сумма трех входных переменных $a + b + c$. При этом S_1 должен иметь значение 2, а S_0 значение 1.

Задача 3.4. Могут ли два следующих уравнения при условии упрощения привести к результату, при котором будет сформирована двухступенчатая логическая схема?

а) KDNF из уравнения (3.30).

б) KKNF из уравнения (3.31).

Задача 3.5. Докажите:

а) $a \leftrightarrow \neg b = \neg(a \leftrightarrow b)$

б) Если справедливо $f = a \leftrightarrow b \leftrightarrow c$, то справедливо и $\neg f = \neg a \leftrightarrow \neg b \leftrightarrow \neg c$.

ГЛАВА 4

ПОВЕДЕНИЕ ЛОГИЧЕСКИХ ВЕНТИЛЕЙ

В данной главе реальные рабочие характеристики логических вентилей будут рассмотрены настолько глубоко, насколько это необходимо для понимания цифровых схем. В главе 5 эта тема рассматривается более полно.

Вначале будет рассмотрен вопрос, в какой мере широко можно применять двоичные системы в качестве моделей реальных систем. Для этого будет использован инвертор. На рис. 4.1 а показаны типичные двоичные сигналы, появляющиеся на входе и выходе инвертора в реальной системе в типовом случае. На рис. 4.1 б показан сигнал $x(t)$, являющийся идеализацией входного сигнала $U_e(t)$, на рис. 4.1 с показан сигнал $y(t)$, являющийся идеализацией выходного сигнала.

Как показывает рисунок, реальная система отличается от идеализированной системы в следующем:

- Реальная система характеризуется непрерывным сигналом. Технические системы по природе своей имеют допуски и на них накладываются помехи, создаваемые такими статистическими процессами, как шумы. Вследствие этого невозможна генерация сигналов, которые принимали бы строго два амплитудных значения.
- Переходы от значения 0 к значению 1 происходят в реальных системах плавно. Фронты импульса описываются временем нарастания и временем спада.
- Выходной сигнал реагирует на входной сигнал с задержкой во времени. Этот и приведенный в последнем пункте эффекты являются следствием конечной скорости реакции реальных компонентов.

Система на основе цифровой техники выполняется таким образом, чтобы она работала как система с дискретными значениями сигналов до тех пор, пока фактический сигнал изменяется внутри заданных амплитудных и временных границ.

- Амплитуды. Вначале проектирование цифровых систем должно производиться с помощью методов аналоговой техники, чтобы убедиться, что сигнал находится в заданных амплитудных пределах. Если это имеет место, то 0 и 1 будут безусловно различаться, и с системой можно будет работать, используя описанные в главе 3 методы цифровой техники.
- Время задержки. В случае пренебрежения временем задержки сигнала в вентилях функционирование системы происходит с ошибками. С помощью соответствующего построения схемы необходимо избежать влияния времени задержки сигналов на работу схемы. Правила проектирования, позволяющие достигнуть этого, изложены в последующих главах.

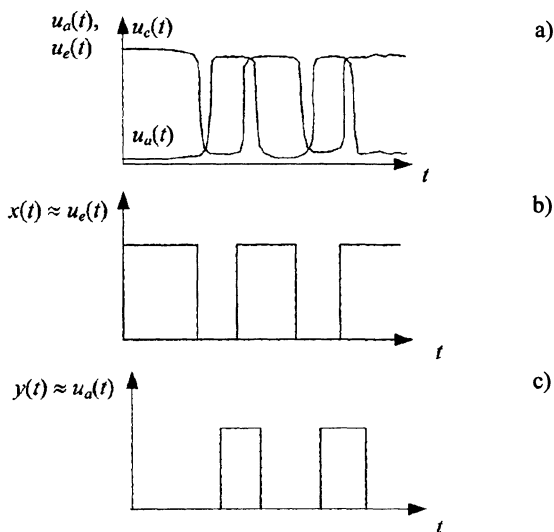


Рис. 4.1. а) Реальная цифровая система с входным сигналом $U_e(t)$ и выходным сигналом $U_d(t)$; б) идеализированный входной сигнал $x(t)$; в) идеализированный выходной сигнал $y(t)$.

4.1. Положительная и отрицательная логика

В цифровой технике работают с переключателями, которые могут выдавать только два различных уровня напряжения. Высокий уровень напряжения выражается через H(=High), низкий через L(=Low). До этого момента в Булевой алгебре использовали символы 0 и 1. оба символа соотносятся в технике с двумя уровнями напряжения.

В электрической цифровой схеме высокое напряжение, например, может соответствовать 1, а низкое напряжение — 0. Возможны и обратные соотношения. Тогда говорят об отрицательной логике.

Таблица 4.1. Соответствие уровней напряжения логическим состояниям.

Напряжение	Уровень	Логические состояния	
		положительная логика	отрицательная логика
$\approx 5\text{ В}$	H	1	0
$\approx 0\text{ В}$	L	0	1

В структурных схемах вместо логических символов можно применять обозначения уровней напряжения. Пример показан на рис. 4.2. Обозначения с использованием уровней можно узнать по пометкам в виде треугольников вместо пометок в виде инверсионных кружков. Как только в структурной схеме появляется треугольник, речь идет о представлении с помощью уровней. Исходя из вышесказанного, в случае положительной логики замена треугольников на инверсионные кружки дает привычное изображение на основе логической величины. В свою очередь при работе с отрицательной логикой все входы и выходы, не имеющие треугольников, можно снабдить инверсионными кружками.

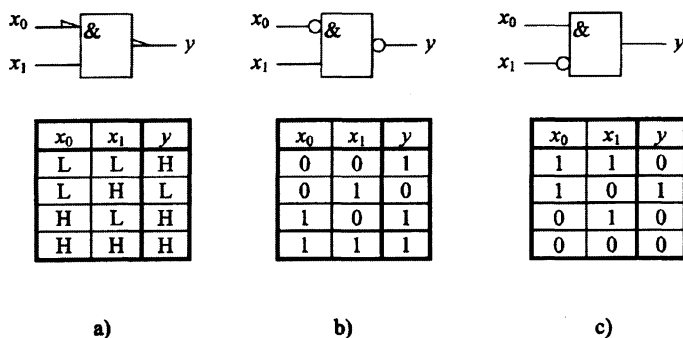


Рис. 4.2. Логические символы и таблицы истинности для: а) изображения с помощью уровней; б) положительная логика; в) отрицательная логика.

4.2. Определение времени переключения

Электронные переключатели реагируют на смену входных сигналов с задержкой. Кроме того, время перехода от низкого (*low*) уровня к высокому (*high*) уровню (или обратно) не может быть сделано

произвольно коротким. Время нарастания t_{tLH} (*transition time low – high*) и время спада t_{tHL} (*transition High – low*) определяются интервалами, границы которых соответствуют 10% и 90% максимальной амплитуды напряжения.

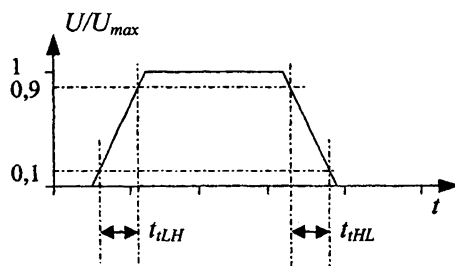


Рис. 4.3. Определение времени нарастания t_{tLH} и времени спада t_{tHL} .

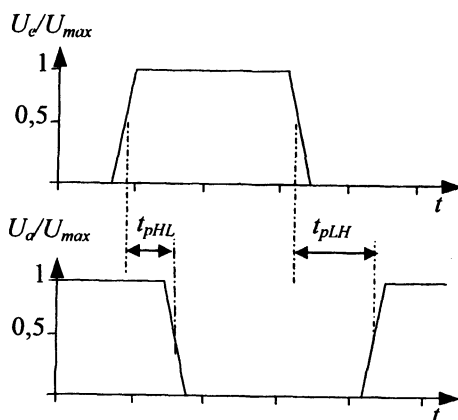


Рис. 4.4. Определение величин t_{pLH} и t_{pHL} .

Время задержки распространения при переходе от уровня *low* к уровню *high* t_{pLH} (*propagation delay time low – high*) и время задержки распространения при переходе от уровня *high* к *low* t_{pHL} (*propagation delay high – low*) определяются как интервалы времени между моментом времени соответствующим уровню напряжения на входе, равному 50% максимального напряжения на входе и моментом достижения такого же уровня напряжения на выходе. Время задержки распространения сигнала в логическом венти́ле представляет собой среднее значение упомянутых выше интервалов времени:

$$t_p = (t_{pHL} + t_{pLH})/2 \quad (4.1)$$

Теперь следует определить отрезки времени, характеризующие тактовый сигнал. В цифровой технике тактовые сигналы применяются для синхронизации различных процессов. Здесь отрезок времени, в течение которого тактовый сигнал находится на высоком уровне напряжения, именуется t_p . Тактовый период определяется как T_p . Часто применяют тактовую частоту $f_p = 1/T_p$.

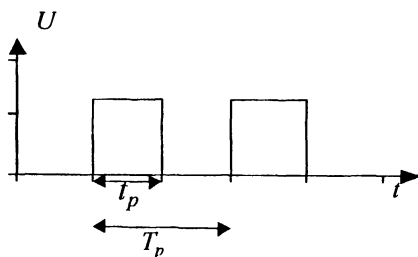


Рис. 4.5. Идеальный тактовый сигнал с длительностью импульсов t_p и периодом повторения импульсов T_p .

4.3. Передаточная характеристика, запас по помехоустойчивости

Передаточная характеристика отражает амплитудную характеристику цифрового вентиля и представляет собой зависимость выходного напряжения от входного. Как правило, она приводится только для инвертора, поскольку характеристики других вентиляей могут быть сведены к ней. На рис. 4.6 приведены идеализированная и реальная передаточные характеристики инвертора. Идеализированная характеристика резко изменяется при входном напряжении $U_e = U_s$, переходя с высокого выходного уровня U_H на низкий выходной уровень U_L . Реальная характеристика обладает переходом с плавным наклоном. Реальные характеристики вентиляей зависят от температуры и имеют разброс, определяемый условиями изготовления.

Посредством передаточной характеристики вводятся границы, в пределах которых сигнал можно рассматривать как H либо как L . На рис. 4.7 приведена типичная передаточная характеристика вентиля. На этот же рисунок вынесены граничные характеристики, задаваемые спецификацией инвертора. Характеристика реального инвертора лежит между граничными. Как можно увидеть из рисунка, выходной сигнал U_a лежит между U_{aHmin} и U_{eLmax} и V_{DD} ¹,

¹ V_{DD} — напряжение питания инвертора.

в то время как входной сигнал U_e лежит между $0B$ и U_{eLmax} . Эти границы задаются для всех вентилях, используемых в схеме.

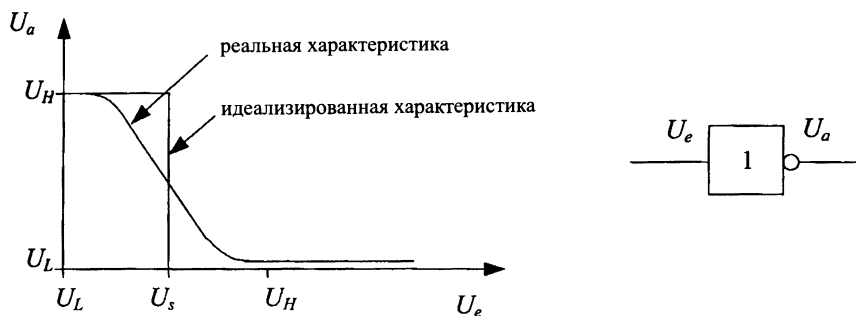


Рис. 4.6. Реальная и идеализированная передаточные характеристики инвертора.

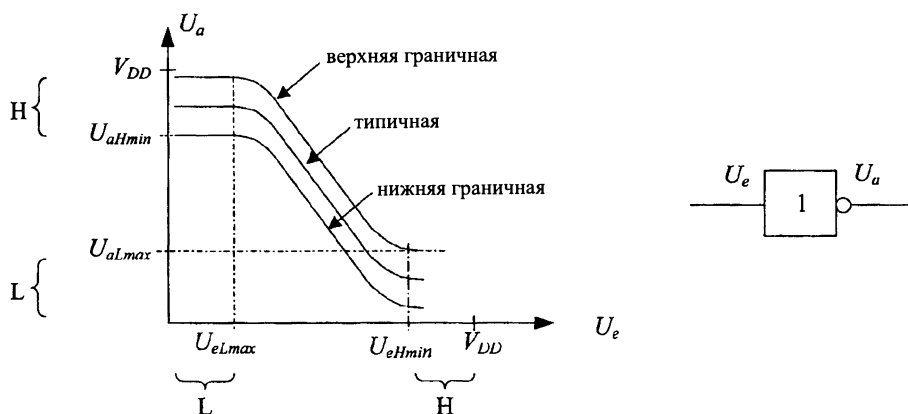


Рис. 4.7. Передаточные характеристики инвертора

На рис. 4.8 приведены определенные на основе вышесказанного области возможных значений для выходного напряжения какого-либо вентиля U_a и для входного напряжения следующего вентиля U_e . При этом становится ясно, что передаточная характеристика должна выглядеть так, чтобы границы лежали в следующем порядке: область, в которой сигнал на входе второго вентиля признается высоким (*high*), должна перекрывать область, в которой выходной сигнал первого вентиля может лежать в наихудшем случае. Только в этом случае сигнал с выхода первого вентиля всегда будет правильно распознан на входе второго вентиля, и не будет иметь места сбой в работе схемы и дальнейшее распространение ошибки. Это наблюдение имеет фундаментальное значение в цифровой технике. Если

выбрать таким соотношение входных и выходных уровней для всех вентилях, то можно будет строить сколько угодно сложные схемы, не заботясь об амплитудных условиях. Но при этом необходимо следить за тем, чтобы к одному вентилю было подсоединено только максимально допустимое количество вентилях, так как нагрузка на большее количество вентилях на выходе может изменить уровень выходного сигнала.

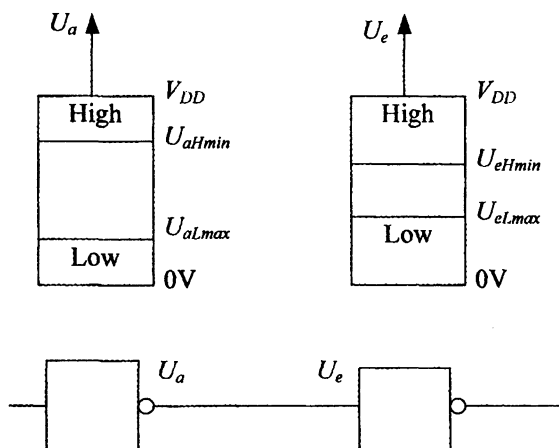


Рис. 4.8. Границы входного и выходного сигналов для двух следующих друг за другом инверторов.

По рис. 4.8 можно также определить величины запаса помехоустойчивости. Запас помехоустойчивости для высокого (*high*) уровня U_{nH} и запас помехоустойчивости для низкого (*low*) уровня U_{nL} определяются как разности уровней напряжения на выходе и на следующем входе:

$$\text{Высокий уровень (high):} \quad U_{nH} = U_{aHmin} - U_{eHmin}$$

$$\text{Низкий уровень (low):} \quad U_{nL} = U_{eLmax} - U_{aLmax}$$

Следовательно, величины запаса по помехоустойчивости (интервалы помехоустойчивости) представляют собой «интервалы безопасности». Для того, чтобы импульсы помех не вызывали сбоев в работе схем, эти интервалы должны возможно большими.

4.4. Вентили

В цифровой технике используются транзисторы, работающие в режиме переключения. То есть используются переключатели, которые

управляются одним сигналом. На рис. 4.9 приведены два символа управляемых переключателей. Левый закрывается при $x = H$, правый при $x = L$.

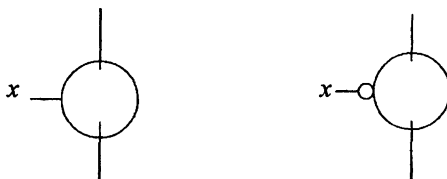


Рис. 4.9. Символы для переключателей. Слева включение происходит при $x = H$. Справа включение происходит при $x = L$.

Как правило, логические вентили реализуются на комплиментарных¹ полевых или биполярных транзисторах и за счет этого имеют малые токи и мощность рассеяния в стационарных состояниях (рис. 4.10). Всегда, когда один из переключателей открыт, а другой закрыт и наоборот.

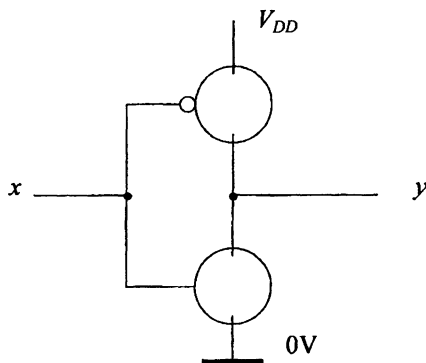


Рис. 4.10. Комплиментарный инвертор.

Если $x = H$, то нижний переключатель открыт, и выход y подсоединен к 0, и, следовательно, лежит на L . Если $x = L$, то выход y замкнут с напряжением V_{DD} , и, следовательно, лежит на H . Почти все вентили имеют, как этот инвертор, подобную схему, называемую также схемой типа *totem pole* (пушпульный каскад). Для специальных применений могут быть предложены и другие варианты схем, как это показано в дальнейшем.

¹Комплиментарными называются полевые транзисторы с каналами n и p -типа проводниками и биполярные n - p -и p - n - p транзисторы.

4.4.1. Вентили с открытым коллектором (open collector)

В этом варианте схемы вентиль состоит только из одного переключателя, как это показано в отдельных, пунктирных прямоугольниках на рис. 4.11. Один вывод переключателя через резистор R_0 подсоединяется к положительному напряжению питания V_{CC} . Этот вариант вентиля широко используется в биполярных схемах. К общему резистору R_0 может быть подключено большее число выходов переключателей.

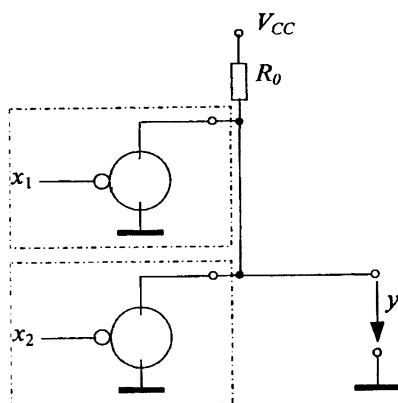


Рис. 4.11. Два переключателя с выходами на основе открытых коллекторов, образующих при соединении логический вентиль.

При положительной логике (высокий уровень напряжения $H = 1$) получаем связь между выходами по типу **И**. В этом случае, для всех x_i необходимо иметь равенство $x_i = 1$, чтобы все переключатели были открыты, а выход перешел на высокий уровень напряжения ($=\text{High}$).

Таблица 4.2. Рабочие характеристики схемы с открытыми коллекторами (рис. 4.11) при использовании положительной и отрицательной логики.

Уровни напряжения		
x_2	x_1	y
L	L	L
L	H	L
H	L	L
H	H	H

Положительная логика логическое И		
x_2	x_1	y
0	0	0
0	1	0
1	0	0
1	1	1

Отрицательная логика логическое ИЛИ		
x_2	x_1	y
1	1	1
1	0	1
0	1	1
0	0	0

Данную схему называют **wired-or** или **wired-and** («проводное ИЛИ» или «проводное И»). Она используется для экономии переключате-

лей, особенно в тех случаях, когда требуются переключатели с многими входами. Пример показан на рис. 4.12. На схемных обозначениях вентилей выход с открытым коллектором, соответствующий рис. 4.11, обозначен подчеркнутыми ромбами. Аналогично биполярным схемам в схемах на полевых транзисторах возможен и выход с открытым стоком, но он употребляется редко.

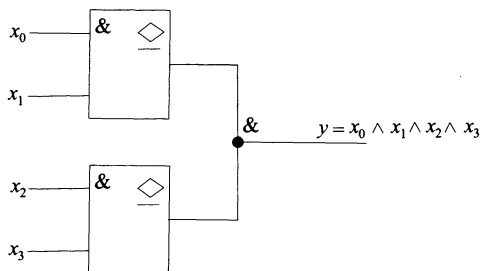


Рис. 4.12. Схемные обозначения для двух И — вентилей с выходами на основе открытых коллекторов.

4.4.2. Вентиль с тремя состояниями

В тех случаях, когда из соображений экономии используют один кабель для взаимной передачи информации между многими передатчиками, часто применяют системы с шинами. Для того, чтобы выходы нескольких блоков подсоединить к одной шине, необходимо обеспечить отключение от шины неактивных блоков, т. е. сделать их выход высокоомными. Этому добиваются с помощью особой схемы, которую именуют «схемой с тремя сторонами», или (*three-state*, сокращенно TS). Если несколько выходов схем «с тремя состояниями» работают совместно с одной шиной, то в каждый момент времени может быть разрешен («enable») только один выход, другие должны оставаться в отключенном (высокоомном) состоянии.

На рис. 4.13 показана схема, в которой оба выходных переключателя могут быть одновременно переведены в высокоомное состояние с помощью enable — сигнала (то есть «разрешающего» сигнала) E . Логический символ представлен с индикацией зависимости, которая позднее будет объяснена подробнее. Сокращение E_n со стоящей за ней цифрой 1 означает, что выход, помеченный цифрой 1, управляется входом E_n . При наличии нескольких выходов все, помеченные цифрой 1, управляются enable — входом. Треугольник означает выход с тремя состояниями.

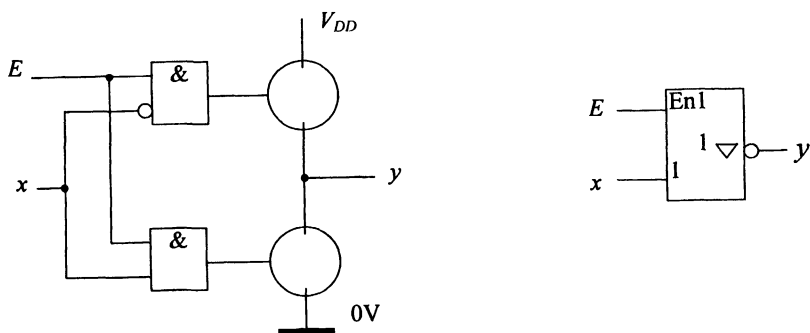


Рис. 4.13. Схема с тремя состояниями. Слева: принципиальная схема с дающим разрешение (enable) E и входным сигналом x . Справа: логический символ.

На рис. 4.14 в качестве примера показаны три двунаправленных интерфейса, работающих с одной шиной, к которой может быть подключено большое количество подобных интерфейсов. Интерфейс n может быть включен с помощью $E_n = 1$ на передачу. Но при этом необходимо обеспечить отключение остальных интерфейсов. Принимать каждый интерфейс может независимо от других. Таким образом, величина потенциала в шине устанавливается одним передатчиком.

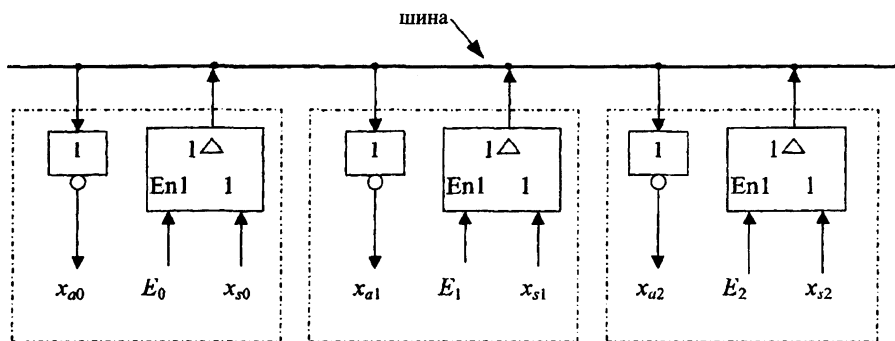
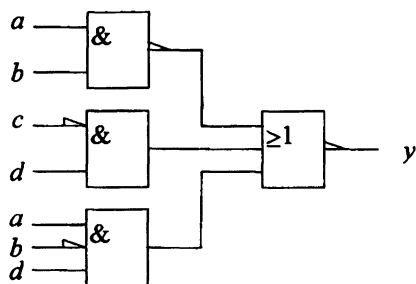


Рис. 4.14. Три двунаправленных интерфейса с выходами типа TS , которые сообщаются через одну шину.

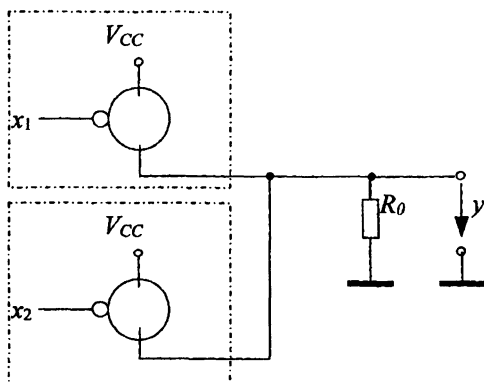
4.5. Упражнения

Задача 4.1.

- Упростите представленную ниже схему.
- Представьте символические обозначения логических элементов, соответствующие положительной и отрицательной логике.

**Задача 4.2.**

Для следующих схем дайте диаграмму уровней и таблицы истинности для положительной и отрицательной логики аналогично табл. 4.2.



ГЛАВА 5

СХЕМОТЕХНИКА

В цифровых логических переключательных схемах транзисторы используются как переключатели. Их задача заключается в том, чтобы замыкать или размыкать цепь для протекания тока. Поэтому в идеальном случае они должны переключаться из положения короткого замыкания во включенном состоянии, в положение с бесконечно большим сопротивлением в выключенном состоянии. Кроме того, в соответствии с рис. 4.6 они должны при определенном пороговом напряжении U_s осуществлять резкое переключение. Однако реальные транзисторы выполняют эти задачи неполно. В следующих параграфах обсуждаются общеупотребительные технологии логических переключательных схем и их свойства.

5.1. КМОП

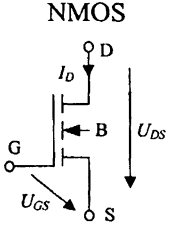
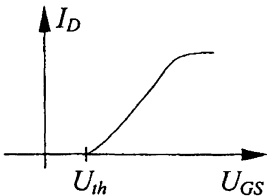
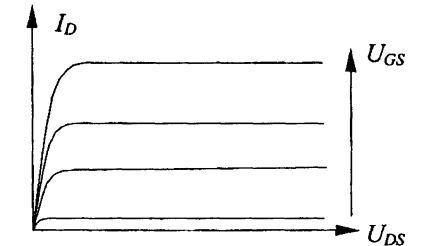
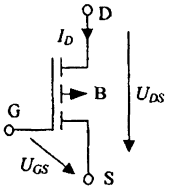
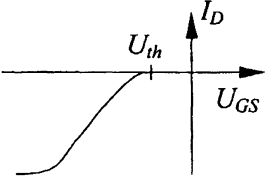
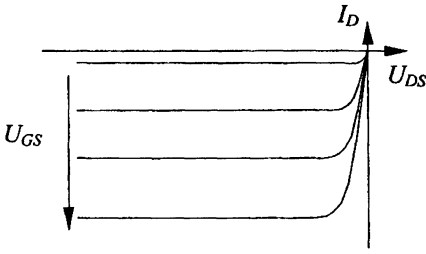
Наиболее часто применяемой технологией логических переключательных схем является КМОП-технология (КМОП = комплементарная структура металл–окисел–полупроводник).

Применяемые полевые транзисторы (ПТ) имеют то преимущество, что, их затвор изолирован оксидом, и поэтому в статическом состоянии ток на входе не протекает. В табл. 5.1 представлены схемы переключателей, переходные и выходные характеристики n -канального и p -канального МОП-ПТ обогащенного типа. МОП-ПТ с каналами n и p типа называют также N-МОП и P-МОП транзисторами.

На переходных характеристиках показано пороговое напряжение U_{th} , при котором транзистор начинает проводить (th — threshold, порог). U_{th} у N-МОП транзисторов положительно, а у P-МОП транзисторов отрицательно.

Как видно из переходных характеристик, N-МОП транзистор включен при положительных напряжениях затвор-исток U_{gs} более высоких, чем U_{th} . P-МОП транзистор включается при напряжениях затвор-исток U_{gs} , более отрицательных, чем пороговое напряжение U_{th} . Можно, однако, видеть что, характеристика в области перехода от выключенного к включенному состоянию, является нелинейной.

Таблица 5.1. Характеристики N-МОП и P-МОП транзисторов.

Тип транзистора	Переходная характеристика	Выходная характеристика
NMOS 		
		

Ток стока N-МОП транзисторов положителен, в то время как у P-МОП транзисторов он отрицателен. Поэтому транзисторы соединяют в схему, объединяя стоки обоих транзисторов так, как это показано на рис. 5.1. Затворы обоих транзисторов соединяют между собой, так что $U_{gs(NMOS)} = U_e$ и $U_{gs(PMOS)} = U_e - V_{DD}$. За счет правильного выбора U_{th} и V_{DD} обеспечивается, стационарные состояния, при которых всегда один из транзисторов выключен, а другой — включен. Полученная схема работает как инвертор, так как при $U_e = 0$ В N-МОП транзистор выключен, а P-МОП транзистор проводит. Отсюда $U_a = V_{DD}$. Напротив, при $U_e = V_{DD}$ P-МОП транзистор выключен и N-МОП транзистор проводит, так что $U_a = 0$ В. Поэтому структура электронной схемы соответствует инвертору на рис. 4.10.

Схема также применяется как цифровой усилитель. В этом случае, ее обозначают как буфер. Кроме того, она образует основу для базовых цифровых КМОП-вентилей NAND (НЕ-И) и NOR (НЕ-ИЛИ). Для оценки качества вентиля важной является передаточная характеристика $U_a = f(U_c)$. Как можно видеть из рис. 5.2, передаточная характеристика КМОП-вентилей почти идеальна, поскольку она резко изменяется от одного состояния сигнала к другому.

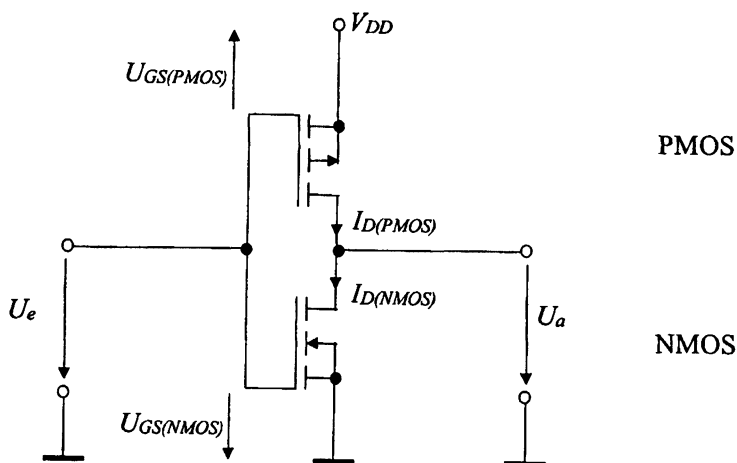


Рис. 5.1. КМОП-инвертор.

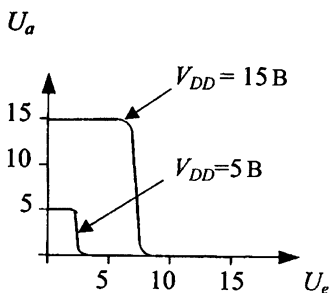


Рис. 5.2. Передаточные характеристики КМОП-инверторов при напряжениях питания 5 В и 15 В.

5.1.1. Нагрузочная способность

Под нагрузочной способностью (*fan-out*) понимают коэффициент разветвления на выходе логического элемента, то есть количество вентилях, подсоединяемых к выходу. Как правило, к выходу одного вентиля подключается несколько входов других вентилях. К стандартному КМОП-вентилю может быть подключено большое число (например, 50) входов стандартных вентилях, что является следствием чисто емкостного характера входа КМОП вентиля. При описании говорят, что КМОП-вентиль имеет коэффициент нагрузки по выходу, или *fan-out*, например 50. Но при такой емкостной нагрузке выхода как будет показано ниже повышается время переключения.

5.1.2. Основные схемы NAND и NOR

Основные КМОП-схемы образуются из инвертора путем подсоединения к N-МОП и P-МОП-транзистором еще одному транзистору последовательно или параллельно. Таким образом, получают схему NAND или NOR (рис. 5.3). В схеме NOR, представленной на рис. 5.3, выходной сигнал y соответствует уровню L всякий раз, когда один из входов соответствует уровню H , поскольку тогда, по крайней мере, один из n -канальных ПТ проводит заперт и, по крайней мере один из p -канальных ПТ. Напротив, в схеме NAND y находится на уровне L только тогда, когда оба входа находятся на уровне H . В этом случае n -канальные ПТ проводят, а p -канальные ПТ находятся в запертом состоянии.

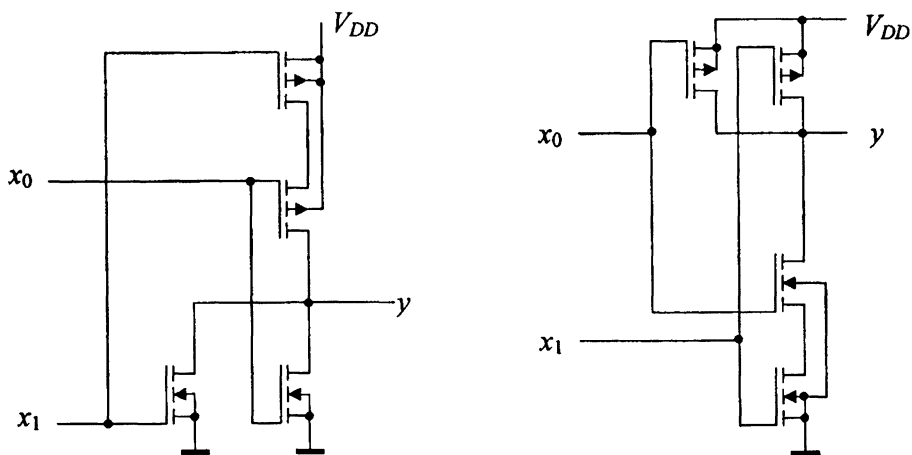


Рис. 5.3. Основные КМОП-вентили: слева — NOR, справа — NAND.

Структура реального КМОП-вентилей показана на рис. 5.4. Схема может быть расчленена на четыре части:

1. Входная схема защиты должна препятствовать разрушению вентиля статическим электричеством. Верхние диоды предназначены для напряжений, которые превышают рабочее напряжение, и они включены в проводящем направлении. Нижние диоды предназначены для напряжений, которые являются отрицательными.
2. Входной буфер уменьшает сдвиг входного уровня последовательно включенных ПТ (особенно у вентилей с более чем с двумя входами).

3. Собственно логический вентиль, генерирующий логическую функцию.
4. Выходной драйвер, улучшающий передаточную характеристику, уменьшающий обратное воздействие выхода на вход и повышающий нагрузочный ток.

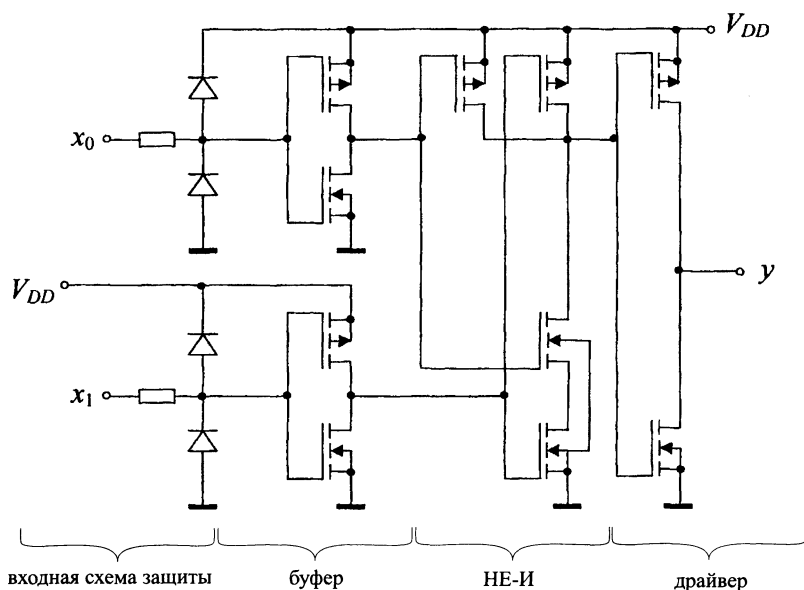


Рис. 5.4. КМОП-вентиль НЕ-ИЛИ (HEF4001).

5.1.3. Передаточный вентиль

Показанная на рис. 5.5 схема известна как «передаточный вентиль» (*transmission gate*). Речь идет об аналоговом переключателе с многосторонним применением. Инвертор, состоящий из T_3 и T_4 , формирует сигналы для собственно передаточного вентиля, состоящего из T_1 и T_2 . Если на вход S инвертора подан сигнал H , то к T_2 приложен высокий потенциал, а к T_1 потенциал L . Поскольку T_1 и T_2 симметричны относительно стока и истока, оба транзистора включены и передаточный вентиль проводит. И наоборот, при подаче низкого (*low*) уровня на вход инвертора T_1 и T_2 выключены и передаточный вентиль закрыт. Благодаря применению n - и p -канальных ПТ схема является симметричной.

Передаточный вентиль может быть применен, например, для присоединения к шине в качестве переключателя с тремя состояниями.

ми TS . Он используется также для более эффективной реализации логических вентилях [12].

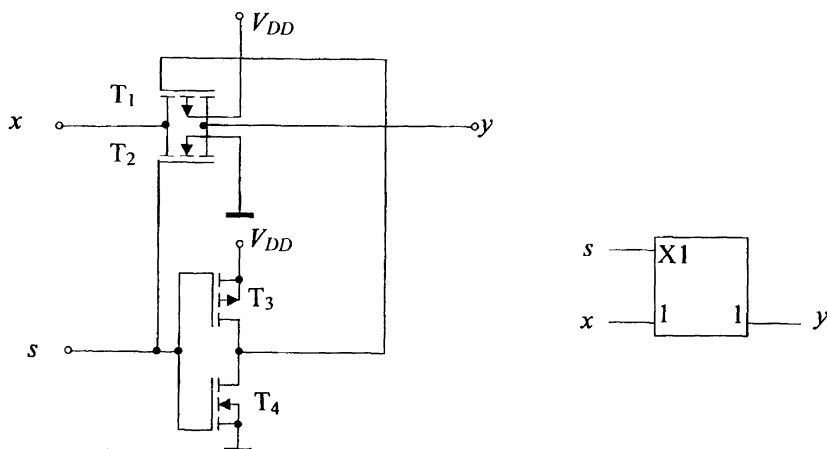


Рис. 5.5. Передаточный КМОП-вентиль и логический символ.

На логическом символе управляющий вход отмечен знаком $x1$, который указывает на то, все входы и выходы, обозначенные цифрой 1, связаны между собой, когда на обозначенном через $x1$ входе имеет место 1.

5.1.4. Элемент с тремя состояниями

Элемент с тремя состояниями в КМОП-варианте может быть выполнен, например, с двумя дополнительными транзисторами, которые в зависимости от состояния входа или проводят или находятся в высокоомном состоянии. Подобная схема показана на рис. 5.6.

5.1.5. Специфические свойства КМОП

- Неиспользуемые выходы должны быть соединены с корпусом (то есть с землей), V_{DD} или с используемым входом, иначе потенциал на них будет неопределенным.
- Эффект «защелкивания» (*latch-up*) может привести к перегрузке схемы. При этом в КМОП-инверторе включается паразитный тиристор. Этот эффект возникает при больших токах и при высокой температуре окружающей среды.
- КМОП-схемы очень чувствительны, несмотря на входную схему защиты, к статическому электричеству.

- КМОП-вентили могут быть, в отличие от ТТЛ-вентилей, соединены параллельно как со стороны входа, так и со стороны выхода. Поскольку при возрастающей температуре ток стока падает, при параллельном включении на выходе транзистор с более высоким током нагрузки имеет тенденцию к уменьшению тока нагрузки. Вследствие этого, КМОП вентили при параллельном включении становятся термически стабильными.

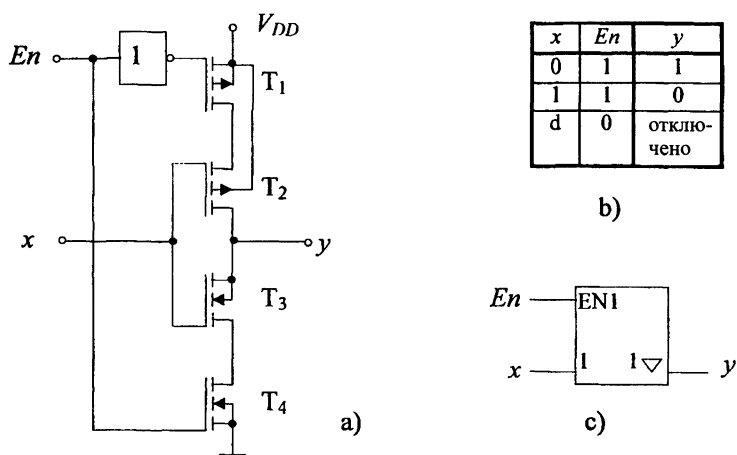


Рис. 5.6. Вентиль с тремя состояниями на основе КМОП. а) схема б) таблица истинности в) логический символ

- КМОП блоки имеют очень большой коэффициент разветвления по выходу, поскольку входы вентиля очень высокоомны. При больших коэффициентах разветвления (fan-out) время нарастания и время спада импульса сильно возрастает, как это будет показано ниже.
- Фронты импульсов, используемых для управления КМОП-вентильями, должны иметь возможно большую крутизну. При медленном переключении выходные транзисторы слишком долго находятся в проводящем состоянии, что вызывает тепловые проблемы. Кроме того, КМОП схемы в момент переключения очень чувствительны к помехам, что может привести к ошибочным переключениям.

5.2. ТТЛ

Ранее наиболее распространенным способом реализации логических вентилях была транзисторно-транзисторная логика (ТТЛ) (рис. 5.7). Ее функции основываются на применении многоэмиттерного транзистора T_1 на входе. Если на всех входах напряжения близки к напряжению питания (H), то коллектор входного транзистора работает как эмиттер, и транзистор работает в инверсном активном режиме. Включается следующий транзистор T_2 (рис. 5.7), вследствие чего коллектор на выходе соответствует уровню L . Для того, чтобы входной ток был малым, коэффициент усиления инверсионного тока транзистора T_1 должен быть близким к 1. Поэтому концентрация легирующей примеси в коллекторе должна быть близка к концентрации примеси в базе. Если на одном из входов напряжение соответствует уровню L , то входной транзистор T_1 работает в активном нормальном режиме (пропускает ток). Напряжение коллектор–эмиттер опускается до минимального остаточного напряжения, а следующий транзистор T_2 запирается. Выходное напряжение соответствует уровню H . Скорость переключения может быть еще улучшена, если между базой и коллектором включить диод Шоттки таким образом, чтобы ограничить протекание тока в диоде база–коллектор транзистора. При этом заряд в базе остается малым и длительной перезарядки при переходе от прямого режима к инверсному режиму можно избежать. В типовое обозначение ТТЛ-вентиля с диодом Шоттки вводится буква S . Транзисторы на логической схеме обозначаются S -образным значком.

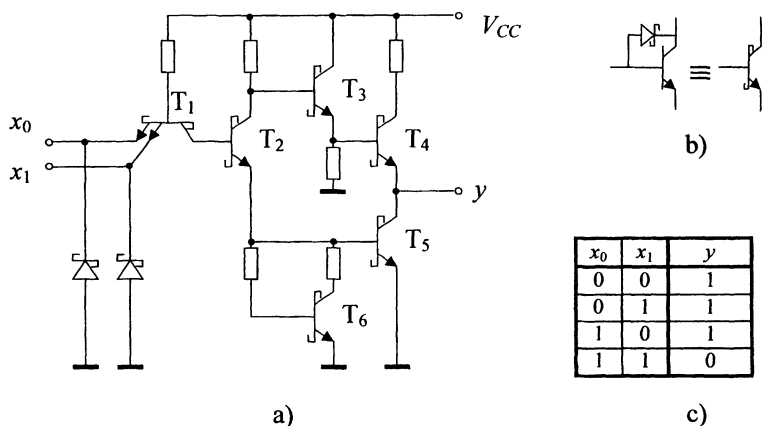


Рис. 5.7. а) ТТЛ-вентиль НЕ-И (74S00); в) маркировка транзисторов с диодами Шоттки; с) таблица истинности.

ТТЛ-вентиль стремя состояниями (*TS*) показан на рис. 5.8, он имеет разрешающий выход *En*, с помощью которого выход можно сделать высокоомным (отключенным).

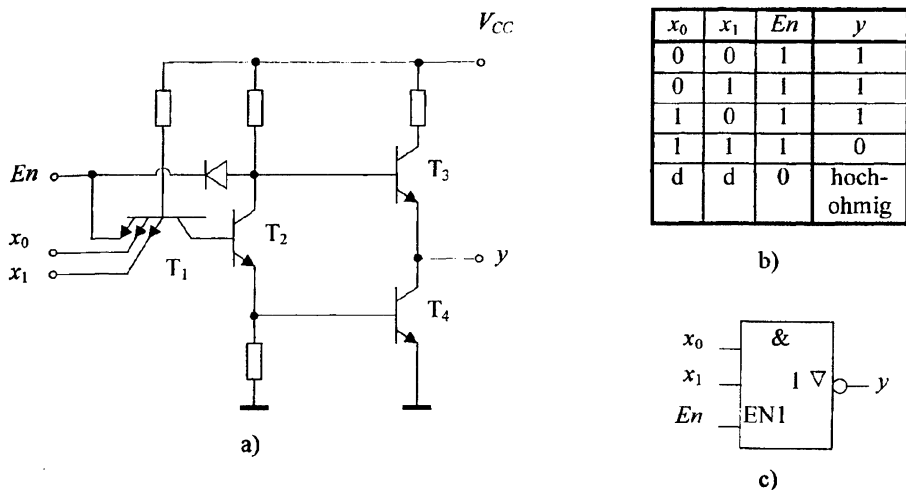


Рис. 5.8. а) принцип построения ТТЛ-вентилей типа *tri-state* (NAND); б) таблица истинности для положительной логики; в) логический символ 1-высокоомный.

Если на входе *En* имеет место низкое напряжение *L*, то верхний выходной транзистор T_3 заперт. Поскольку разрешающий вход *En* подключен к эмиттеру транзистора T_1 , то этот транзистор открывается. Вследствие этого T_2 запирается, и на эмиттере резисторе T_2 нет падения напряжения, поэтому T_4 тоже заперт. Так как оба выходных транзистора T_3 и T_4 заперты, выход вентиль находится в высокоомном состоянии.

Если на входе *En* напряжение соответствует уровню *H*, то соответствующий эмиттер и диод ток не пропускают. Схема работает как нормальный вентиль НЕ-И.

5.2.1. Нагрузка выходов

К выходу ТТЛ вентиль может быть подключено только ограниченное количество входов следующих вентилей. У ТТЛ нагрузочный ток выходного каскада ограничен. Стандартные ТТЛ-элементы имеют жестко ограниченные максимальные нагрузочные токи, указанные в табл. 5.2.

Отсюда следует, что к одному стандартному ТТЛ-вентилю может быть подключено до 10 стандартных ТТЛ вентилей. Коэффи-

циент разветвления стандартной ТТЛ-серии равен 10, но при этом существует и другой коэффициент разветвления, который имеет место при смешанном использовании различных серий. Синонимами понятия «коэффициент разветвления» (*fan-out*) являются понятия «коэффициент запуска» (*drive-factor*) «коэффициент нагрузки» (*load-factor*).

Таблица 5.2. Максимальные выходные и минимальные входные токи стандартных ТТЛ-элементов.

	Максимальный нагрузочный ток	Минимальный входной ток
Низкий уровень	16 мА	1,6 мА
Высокий уровень	0,4 мА	0,04 мА

Для всех ТТЛ-серий справедливо:

- Напряжение питания $V_{cc} = 5 \text{ В}$
- Почти одинаковые входные и выходные уровни (ТТЛ-уровни) для всех серий:

$$\left. \begin{aligned} U_{OLmax} &= 0,4 \text{ В} \\ U_{OHmin} &= 2,4 \text{ В} \end{aligned} \right\} \text{ выходные уровни}$$

$$\left. \begin{aligned} U_{iLmax} &= 0,8 \text{ В} \\ U_{iLmin} &= 2,0 \text{ В} \end{aligned} \right\} \text{ входные уровни}$$

- Совместимость между выводами одинаковых типов ИС, принадлежащих к различным сериям
- Разомкнутые входы соответствуют логическому уровню (H)
- Невозможность параллельного включения выхода.

5.3. Эмиттерно-связанная логика

В эмиттерно-связанной логике (*ECL*, *emitter-coupled logic*) используются дифференциальные усилители, в которых транзисторы не вводят в насыщение (рис. 5.9). Благодаря этому данные схемы имеют повышенное быстродействие.

Во входном дифференциальном усилителе схемы напряжения входных сигналов x_0 и x_1 сравниваются с опорным сигналом. Если x_0 и x_1 имеют значение L , то транзисторы T_1 и T_2 заперты, а транзистор

T_3 открыт. Если, наоборот, x_0 и x_1 имеют значение H , то T_1 или T_2 открыты, а T_3 заперт. Выходной сигнал $\neg y$ имеет значение H . Следовательно, схема выполняет функцию вентиля ИЛИ-НЕ. Порог переключения может быть задан напряжением на базе T_3 , установленным с помощью делителя напряжения.

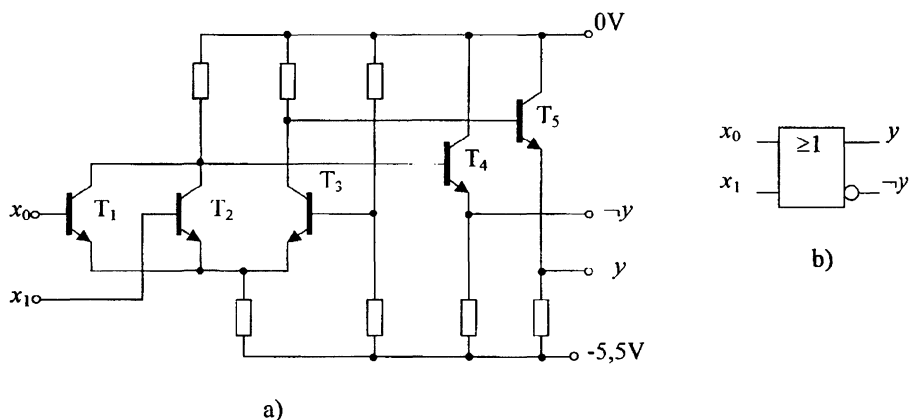


Рис. 5.9. НЕ-ИЛИ-вентиль на основе ELC: а) схема; в) логический символ для положительной логики.

Можно отметить следующие свойства ELC-вентилей:

- ELC вентили быстрее ТТЛ вентилей
- В стационарных состояниях они потребляют большую мощность, а при высоких частотах меньшую мощность, чем КМОП и ТТЛ;
- Одинаковые значения рассеиваемой мощности при низком и высоком уровнях;
- Высокий коэффициент разветвления по выходу
- Низкий запас помехоустойчивости.

5.4. Интегральная инжекционная логика (I^2L)

Интегральная инжекционная логика (I^2L) особенно хорошо подходит для интеграции, так как для нее требуется очень малая площадь поверхности кристалла.

На рис. 5.10 показан типичный инвертор. Транзистор T_1 работает как источник постоянного I_o тока. Если входной сигнал x находится на уровне (H), весь ток втекает в базу T_2 , который становится

открытым. Выходы y_1 и y_2 в этом случае находятся на уровне (L). Если вход соответствует уровню (L), то ток I_o течет в предыдущий вентиль, а выходные сигналы соответствуют уровню (H).

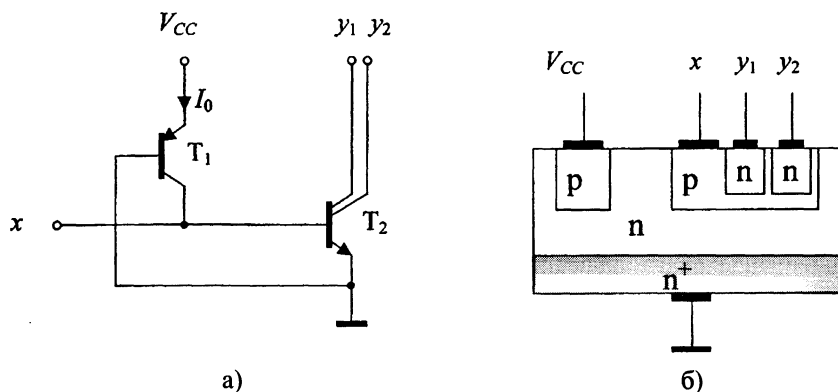


Рис. 5.10. а) схема I^2L -инвертора; б) физическая структура

Логические схемы I^2L типа могут работать с очень малыми рабочими напряжениями, лежащими ниже 1 В. В этом случае запас по помехоустойчивости очень мал. На рис. 5.11 показан ИЛИ-НЕ вентиль, выполненный по I^2L технологии. Оба открытых коллектора могут быть использованы для реализации других логических функций.

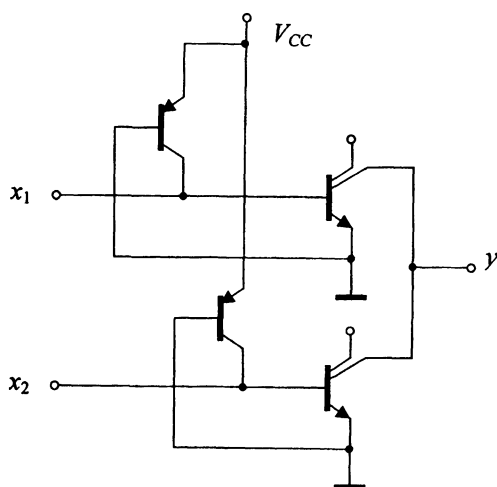


Рис. 5.11. Вентиль ИЛИ-НЕ, выполненный по I^2L технологии.

5.5. Рассеиваемая мощность и характеристики переключения транзисторных переключателей

Ниже исследуются режимы переключения КМОП вентилях, к выходу которого подключена КМОП нагрузка. Для этого используется модель транзисторного переключателя, представленная на рис. 5.12. В этой модели транзистор представлен сопротивлениями R_{on} и R_{off} , в зависимости от того, включен он или выключен. Проводники и входная схема, подключенные к выходу, представлены емкостью C_i .

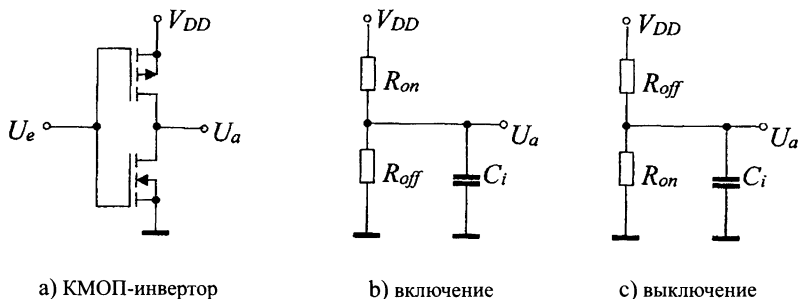


Рис. 5.12. а) КМОП-инвертор; в) модель для процесса включения; с) модель для процесса выключения.

Конденсатор C_i состоит из входных емкостей следующих вентилях, емкостей проводников и из выходной емкости C_{DS} инвертора. В случае использования биполярной схемотехники необходимо учитывать входное сопротивление следующих вентилях.

Следует учитывать, что сопротивление в положении «включено» R_{on} при заданной длине затвора транзистора зависит от ширины затвора, поскольку ток стока пропорционален отношению ширины затвора к длине затвора. Поэтому транзисторы малой площади имеют высокое сопротивление в состоянии «включено». Если решить дифференциальное уравнение, описывающее изменение выходного напряжения во времени, то получим выражения для переключения от H к L :

$$U_a = V_{DD} \left(1 - e^{-t/R_{on}C_i} \right) \quad (5.1)$$

И для переключения от H к L :

$$U_a = V_{DD} e^{-t/R_n C_i} \quad (5.2)$$

Постоянная временная этой функции представляет собой аппроксимацию времени переключения вентиля:

$$t_S \approx R_{on} C_i \quad (5.3)$$

Из этого выражения следует — чтобы получить малое время переключения, сопротивление транзистора во включенном состоянии и выходная емкость должны быть минимизированы.

Рассчитав на основе уравнений (5.1) и (5.2) среднюю рассеивающую мощность P при периодическом переключении с частотой f , и добавив статическую мощность потерь ($V_{DD}/R_{on} + R_{off}$), получим:

$$P = V_{DD}^2 \left(\frac{1}{R_{on} + R_{off}} + f C_i \right) \quad (5.4)$$

Из приведенного можно сделать следующие выводы:

- Для быстродействующих схем необходимы низкие сопротивления во включенном состоянии и, следовательно, транзисторы с большой шириной затвора W (если применяются ПТ).
- Быстродействующие схемы требуют, чтобы емкости проводников были малы, поэтому в быстродействующих схемах к выходу можно подключать только малое число вентилях.
- При увеличении скорости переключения возрастает мощность потерь.
- У быстродействующих и высокоинтегрированных схем напряжение питания должно быть уменьшено.

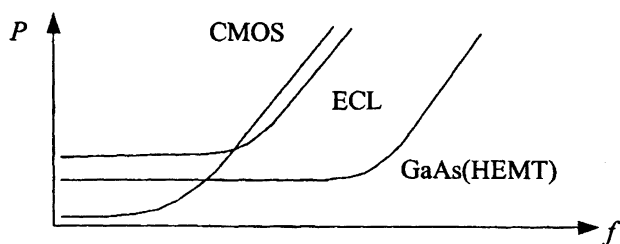


Рис. 5.13. Схематическая зависимость мощности от частоты переключений.

На рис. 5.13 показана зависимость потребления мощности от частоты при использовании для реализации логики на различных технологиях. Наряду с кремниевыми КМОП-технологией и ЭСЛ-технологией представлены результаты, полученные для технологии, на базе сложного полупроводника-арсенида галлия (GaAs). Применяемые в этой технологии специальные полевые транзисторы представляют собой транзисторы с высокой проводимостью электронов

(high-electron-mobility transistor НЕМТ). Как видно из рисунка, в соответствии с уравнением (5.4) имеются статическая составляющая мощности и пропорциональная частоте динамически составляющая. На низких частотах предпочтительней является КМОП-технология, на высоких частотах лучшие результаты дает GaAs-технология.

5.6. Упражнения

Задача 5.1.

- а) Сконструируйте КМОП-НЕ-И вентиль с тремя входами.
- б) Сконструируйте КМОП-НЕ-ИЛИ вентиль с тремя входами.

Задача 5.2. Сформируйте таблицу истинности и символическое обозначение логического элемента на рис. 5.8, приняв за основу отрицательную логику.

Задача 5.3. Определите, какая логическая функция $y = f(x_1, x_0)$ реализуется представленным вентилем, если используется положительная логика.

ДОПОЛНЕНИЕ

ЭЛЕМЕНТНАЯ БАЗА ЦИФРОВЫХ УСТРОЙСТВ ЭНЕРГЕТИКА ОБРАБОТКИ ЦИФРОВОЙ ИНФОРМАЦИИ

Д.1. Введение

Цифровая техника обрабатывает, хранит и отображает информацию, необходимую для решения самых разнообразных проблем. Все перечисленное производится с помощью электронных схем-полупроводниковых интегральных схем. Неизбежной платой за производство новой информации являются затраты энергии. От того, как организованы интегральные схемы зависит, как быстро, точно и какими энергозатратами обеспечивается производство информации.

Высокие скорости и низкие энергетические затраты — это две главных цели развития элементной базы цифровой техники. Эти цели достигаются за счет совершенствования системных решений цифровых устройств, принципов и алгоритмов их функционирования. Другая группа средств достижения этих целей включает схемотехнические, структурные и топологические решения базовых логических вентилей и элементарных переключателей.

Целью данного раздела является изложение основных идей и технических решений, являющихся теоретических основой создания цифровой техники на схемотехническом и структурно-топологическом уровнях.

Д.2. Энергетика и скорость производства цифровой информации

Энергетические затраты на производство информации в цифровых системах и скорость ее производства тесно взаимосвязаны. За высокие скорости приходится расплачиваться высокими затратами энергии. И наоборот, сэкономить энергию можно за счет уменьшения скорости.

Электронные схемы логических вентилей — базовые элементы цифровых интегральных устройств с точки зрения энергетики, представляют собой совокупность электронных компонентов, организованных в единый ансамбль электронных цепей для передачи, накопления и диссипации электрической энергии, потребляемой с внешних источников. В совокупность электронных компонентов обязательно входят переключатели, управляемые входными электрическими сигналами. Эти переключатели формируют электрические цепи совместно или с другими переключателями, или с пассивными компонентами, с помощью которых двоичная цифровая информация представляется в виде уровней напряжений (или токов) за счет потоков энергии между полосами источников питания. При смене входной информации переключатели изменяют конфигурации цепей и соответственно потоков энергии в них. При этом происходит накопление энергии в одних компонентах и ее диссипация в других.

Количество энергии, необходимое для предоставления информации и скорости накопления и диссипации энергии, определяющие скоростные характеристики производства информации, определяются:

- принципами действия элементарных переключателей и других компонентов вентилей;
- видами их соединения в электронных схемах вентилей и способам подключения к источникам энергии;
- способами воплощения компонентов и соединений между ними и источниками энергии в интегральной структуре.

Эти факторы совместно определяют как количество энергии так и скорость ее потребления, необходимые для достижения определенной скорости обработки информации. Они тесно взаимосвязаны и трудно однозначно четко выделить их индивидуальные вклады.

Тем не менее, если приоритет отдать энергетике производства информации, то логично вначале рассмотреть факторы, в первую очередь, определяющие необходимое количество энергии. С этой целью рассмотрим логические вентили как энергопотребители.

Какое количество энергии необходимо для производства информации и достаточно для этого производства с заданным быстродействием — мало ответить на этот вопрос. Надо еще разобраться, как и какими средствами этого достигнуть.

Для этого в данном разделе излагаются принципы организации электронных схем логических вентилей.



Воплощение схемотехнических принципов в конкретные базовые элементы цифровых схем начинается с подбора элементарных переключателей.

Этот выбор определяется прежде всего требованиями по быстродействию, которые предъявляются к цифровому устройству. Скорость производства информации на уровне базового логического вентиля в основном определяется принципом действия элементарного переключателя. Поэтому в данном разделе рассмотрены основные типы переключателей, как элементы цифровых вентилях, главным образом определяющие скорость производства цифровой информации.

И, наконец, электронная схема должна быть реализована в виде интегральной структуры. От того, каким образом это будет выполнено, зависит скорость производства информации и цена, которую за это надо будет заплатить. Изготовление интегральных цифровых устройств дорогостоящее дело. Поэтому требования к параметрам физической структуры, к геометрическим (топологическим) размерам элементов не должны быть чрезмерно жесткими. Они должны соответствовать достигнутому уровню развития технологии. Желательно решить задачу без усложнения технологии по возможности только за счет соответствующей организации конфигурации схемы соединения компонентов и подбора вида и параметров электронных ключей. То есть схемотехническим путем. Особенности структур базовых элементов, определяющие энергетику и скоростные качества базовых вентилях будут изложены в последующих разделах дополнения.

Д.2.1. Классификация вентилях как энергопотребителей

Рассматриваемая классификация в качестве классификационных признаков использует свойства логических вентилях, характеризующих их энергетические характеристики.

Первым классификационным признаком в соответствии со сказанным выше целесообразно использовать вид источников питания, из которого цифровые схемы потребляют энергию, необходимую для производства информации. По этому признаку все виды цифровых схем можно сгруппировать в три класса.

К первому классу отнесены вентилях, питающиеся от традиционных источников электроэнергии искусственного происхождения: вторичные источники стабилизированного напряжения, батарейки, аккумуляторы и другие. Эти источники используются для подавляющего большинства цифровых схем.

Ко второму классу вентиляей — энергопотребителей относятся вентили, которые для питания используют ионизирующее излучение. Широко известны логические вентили, питающиеся от света искусственных источников. Существуют также логические схемы работоспособные от солнечного света.

Они отнесены ко второму классу. К данному классу относятся также и вентили, способные использовать энергию других источников энергии. В качестве таких источников используется фоновое радиоизлучение, сейсмическая энергия вибрации стен зданий, перепады атмосферного давления и другие. Вентили данного класса принципиально отличаются от вентиляей предыдущего класса наличием встроенных преобразователей энергии.

Известны также логические вентили, способные функционировать как от искусственных традиционных источников, так и от естественных источников энергии. Например, такие вентили могут питаться энергией солнечного света, а в его отсутствие — от традиционного аккумулятора. Вентили данного третьего класса принципиально отличаются от двух предыдущих наличием специальных средств, обеспечивающих изменения режима электропитания.

Традиционные источники питания для передачи энергии требуют, по крайней мере, двух шин: собственно шин питания и общей шины. Шины подключаются к каждому вентилю цифрового устройства и занимают значительную часть площади кристалла. Это обстоятельство во многом определяет топологию интегральной схемы и затрудняет внутрисхемные соединения. Большая длина шин и большая площадь шин также отрицательно сказываются на надежности устройств и плотности их упаковки. Энергетическое снабжение с использованием традиционных источников является типичным централизованным с присущими ему недостатками.

Любой одиночный дефект, приводящий к короткому замыканию шин питания, приводит к катастрофическому отказу устройства в целом.

Протекание токов по шинам питания обуславливает термализацию части поставляемой источником энергии. Неизбежные потери энергии при транспорте от источника к вентилю делает принципиально невозможным создание полностью адиабатических схем, то есть, схем совершенно не диссипирующих энергию.

Вторым классификационным признаком рассматриваемой классификации логично использовать характеристику вентиляей-потребителей, отражающую степень диссипации энергии в процессе об-

работки и хранения информации. Долгое время существовали логические элементы только полностью диссипирующие потребляемую от источников питания энергию. Электрическая энергия, поставляемая как искусственными, так и естественными источниками электропитания в процессе производства информации диссипировалась полностью, превращаясь в тепло, выделяемое на резистивных элементах конструкции цифровых устройств. Это порождало большие проблемы теплоотвода. Сравнительно недавно появились новые виды вентилях и логических схем, принципиально отличающиеся от упомянутых выше. Появились логические схемы лишь частично диссипирующие энергию, потребляемую от источника в процессе производства информации. В цифровых схемах данного вида энергия диссипируется лишь часть энергии поставляемой источником, другая же часть — обратно возвращается в источник питания и используется повторно. Схемы данного вида имеют качественные отличия от широко используемых традиционных схем, обуславливающие их несомненные достоинства в экономичности.

Изложенное выше иллюстрируется верхним фрагментом классификационной диаграммы, приведенной на рис. Д.2.1.

В нижнем ряду классификационной диаграммы представлены виды схмотехнических воплощений логических вентилях. Схмотехнические виды отличаются способами представления двоичной информации и способами ее обработки. Именно от того, как представляется и обрабатывается информация, зависит энергетика логических вентилях. Полностью диссипирующие энергию вентилях представлены тремя видами: так называемыми схемами с отношением, схемами без отношения и схемами типа переключателей тока. Эти виды схмотехнических решений потребляют энергию, как правило, от традиционных источников, реже — от излучения. В процессе производства информации эти виды вентилях полностью диссипируют потребляемую энергию.

В схемах с отношением при производстве информации уровень логического «0» формируется за счет падения напряжения на резисторе. При этом транспорт энергии от источника энергопитания сопровождается ее диссипацией.

В схемах без отношения уровень логической «1» формируется путем накопления энергии на паразитной выходной емкости. В процессе транспорта энергии от источника энергопитания к накопителю имеет место ее диссипация на резистивных элементах. Однако, при некоторых уровнях потери энергии можно значительно уменьшить.

Упомянутые три вида схмотехнических решений с их разновидностями и энергетическими особенностями будут подробно рассмотрены в разделах Д.3 и Д.4. В нижнем ряду классификационной диаграммы размещены также несколько видов частично диссипирующих схем вентиляей. Описание принципов их функционирования и конкретных схмотехнических решений посвящен раздел Д.6, здесь же отметим, что эти виды вентиляей также питаются от традиционных источников энергии, как правило, аккумуляторов.

Что же касается вентиляей, потребляющих энергию от комбинированных источников энергии (класс III на классификационной диаграмме), то они находятся в ранней стадии развития. Имеются лишь их отдельные схмотехнические решения, но уже очевидно, что эти могут быть как полностью так и частично диссипирующие. Вентиляей данного класса описаны в разделе Д.6.

Приведенная классификация имеет не вполне законченный вид и не отображает в полной мере все многообразие логических вентиляей. В пределах каждого из видов схмотехнической организации имеются схмотехнические разновидности, реализуемые в различных структурных и приборных вариантах на различных технологических платформах. Все эти варианты интегрального воплощения конечно же определяют как количество энергии так и мощность, необходимые для производства информации. Однако, учет всех схмотехнических разновидностей и вариантов воплощения в классификационной диаграмме на рис. Д.2.1. сделал бы ее слишком сложной и запутанной. Поэтому исключительно по соображениям удобства в соответствующих разделах дополнения приведены отдельные классификационные диаграммы для схем с отношением, схем без отношения и токовых переключателей. Их можно рассматривать как фрагменты общей классификации.

Д.2.2. Общие вопросы энергетики цифровых вентиляей

В работе любой логической схемы можно выделить четыре различных с точки зрения энергетики промежутка времени. На рис. Д.2.2 приведена эпюра изменения напряжения на выходе вентиля в зависимости от времени, на которой отменены эти временные интервалы. На протяжении промежутков t_0 и t_1 вентиль находится в неактивных, так называемых, стационарных состояниях В интервалах времени t_{10} и t_{01} вентиля находится в процессе перехода из одного состояния в другое и наоборот. Именно в эти промежутки времени и происходит собственно производство информации. В стационар-

ных режимах информационные состояния, представленные уравнениями напряжений на выходе, не изменяются. Каждое из четырех состояний логических вентилях обеспечивается энергетически различным образом. Соответственно и различны затраты энергии на производство единицы информации.



Рис. Д.2.1. Классификация цифровых вентилях.

Способы подачи энергии, конкретные механизмы ее использования и расходования различны для различных типов вентилях на протяжении выделенных выше временных интервалов процесса производства информации.

Для того чтобы систематически проанализировать энергетику производства цифровой информации во всем ее многообразии выше была проведена классификация логических вентилях как потребителей электроэнергии.

Способы подачи энергии в вентили определяются прежде всего видом источника питания. Механизмы использования и расходования зависят от схмотехнической организации и видами используемых компонентов вентилях. Все это и определяет классификаци-

онные признаки, которые помогут рассмотреть энергетику производства информации последовательно и систематически. Но вполне целесообразно рассмотреть общие для всех потенциальных логических вентиляей вопросы энергетики производства информации посредством электронных схем.

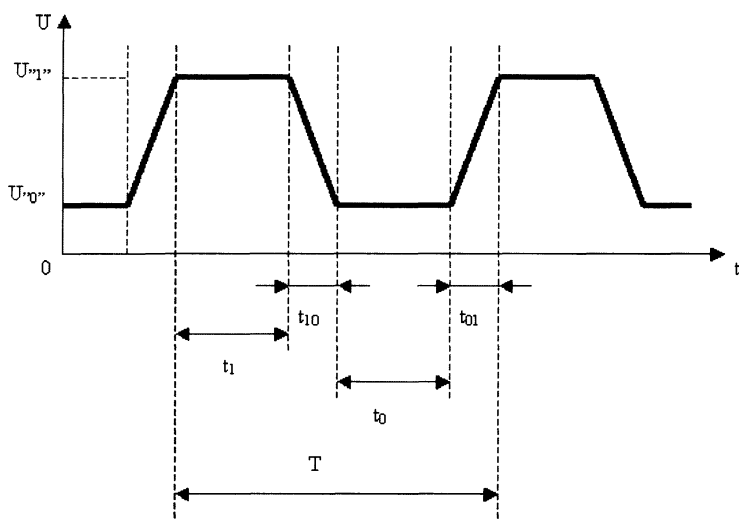


Рис. Д.2.2. Эпюра напряжения на выходе цифрового вентиля.

В общем случае средняя энергия W_{cp}^1 , необходимая для производства и хранения одного бита информации, может быть определена выражением:

$$W_{cp}^1 = 0,5 (W_{cp}^1 + W_{cm}^0 + W_{\partial}^{1-0} + W_{\partial}^{0-1}), \quad (\text{Д.2.1})$$

где W_{cp}^1 — энергия, необходимая для представления логической «1» или энергия сохранения стационарного состояния вентиля, когда на его выходе напряжение равно $U_{«1»}$;

W_{cm}^0 — энергия, необходимая для представления логического «0» $U_{«0»}$;

W_{∂}^{1-0} — энергия, необходимая для производства логического «0», или энергия, обеспечивающая отпирание вентиля;

W_{∂}^{0-1} — энергия, необходимая для производства логической «1», или энергия, обеспечивающая запираение вентиля.

Энергия W_{cm}^1 , обеспечивающая представление логической «1», может быть накоплена и хранится на конденсаторе $C_{\text{вых}}$, входящем в состав электронной схемы логического вентиля без отношения (выходная емкость) и тогда эта энергия будет определять величину $U_{«1»}$

согласно выражению:

$$U_{«1»} = \sqrt{\frac{2W_{cm}^1}{C_{вых}}}. \quad (Д.2.2)$$

В вентильях другого типа (вентили с отношением) сохранение данного стационарного состояния может быть обеспечено при протекании тока, то есть при транспорте энергии от источника через выход вентиля.

Энергия W_{cm}^1 , обеспечивающая представление логического «0» может быть накоплена и храниться на конденсаторе $C_{вх}$, входящем в состав электронной схемы (входная емкость).

В вентильях другого типа сохранение логического «0» обеспечивается за счет транспорта энергии от источника энергоснабжения через открытый переключатель.

В обоих случаях энергия необходима для представления информационного состояния логического вентиля, но механизмы ее использования принципиально различны. Для вентиля первого типа характерным является представление информации хранящейся энергией. Для вентиля второго типа представление информации сопряжено с транспортом энергии.

Транспорт электрической энергии согласно закону Джоуля-Ленца сопровождается ее превращением в тепло. И поэтому даже в стационарных состояниях от источника энергоснабжения потребляется энергия. Количества энергии W_{cm}^1 и W_{cm}^0 для вентиля с отношением зависят от промежутков времени t_1 и t_2 их пребывания в стационарных состояниях, когда информация не производится.

При производстве информации в вентильях первого типа происходит накопление энергии W_{∂}^{10} на конденсаторе $C_{вх}$ и энергии W_{∂}^{01} на конденсаторе $C_{вых}$. Накопление энергии происходит путем ее транспорта от источника энергопитания по электрическим цепям с паразитными резисторами и, следовательно, сопровождается джоулевыми потерями.

Стационарные компоненты W_{cm}^1 и W_{cm}^0 энергии в большинстве типов вентилях определяется не только необходимыми величинами напряжений $U_{«1»}$ и $U_{«0»}$, но также и различными паразитными эффектами в электронных схемах. К таким паразитным эффектам, в значительной мере определяющим энергетику логических схем, прежде всего необходимо отнести диссипацию энергии на резистивностях переключателей и соединений и из-за утечки тока через закрытые переключатели.

Важной энергетической характеристикой любого логического вентиля является скорость $P(t)$ изменения энергии во времени:

$$P(t) = \frac{dW}{dt}. \quad (\text{Д.2.3})$$

С учетом различных мгновенных значений мощности на различных временных интервалах производства информации общее выражение для энергии (Д.2.1) можно записать в следующем виде:

$$W_{cp}^1 = 0,5 \left(\int_0^{t_1} P_{cm}^1(t) dt + \int_{t_1}^{t_{10}} P_{\partial}^{10}(t) dt + \right. \\ \left. + \int_{t_1+t_{10}}^{t_1+t_{10}+t_0} P_{cm}^0(t) dt + \int_{t_1+t_{10}+t_0}^T P_{\partial}^{01}(t) dt \right), \quad (\text{Д.2.4})$$

где: $P_{cm}^0(t)$ — мощность в открытом состоянии вентиля или мощность представления логического «0»;

$P_{cm}^1(t)$ — мощность в закрытом состоянии вентиля или мощность предоставления логической «1»;

$P_{\partial}^{10}(t)$ — мощность производства логического «0»;

$P_{cm}^{01}(t)$ — мощность производства логической «1».

Стационарные составляющие мощности P_{cm}^0 и P_{cm}^1 определяет скорости расходов энергии, источником питания при представлении и хранении информации в виде уровней $V_{\ast 0}$ и $V_{\ast 1}$ соответственно. Динамические составляющие P_{∂}^{10} и P_{∂}^{01} определяют скорости расхода энергии при производстве вентилями новой информации под воздействием входного сигнала.

Пределы интегрирования в выражении для энергии единичного переключения поясняются эпюрой выходного напряжения, приведенной на рис. Д.2.2. Выражение (Д.2.4) приближенно определяет среднее значение энергии единичного переключения логического вентиля, работающего при периодически изменяющемся входном сигнале.

Общее количество энергии, потребляемой вентилем за время работы t при частоте f периодически изменяющегося входного сигнала определяется выражением

$$W_{\Sigma} = W_{cp}^1 \cdot f \cdot t. \quad (\text{Д.2.5})$$

Эта энергия полностью диссипируется в вентилях традиционной микросхемотехники и порождает проблемы теплоотвода. На

рис. Д.2.3 приведена временная зависимость плотности мощности микропроцессоров, иллюстрирующая серьезность проблемы теплоотвода в цифровых устройствах. Эффективный теплоотвод необходим для термостабилизации цифровых устройств, поскольку повышение температуры является основным дестабилизирующим фактором при производстве информации с высокими скоростями. Поэтому цифровые системы характеризуются величиной предельно допустимой мощности $P_{\text{доп}}$, которая характеризует минимально возможную скорость отвода тепла от кристалла интегральной схемы.

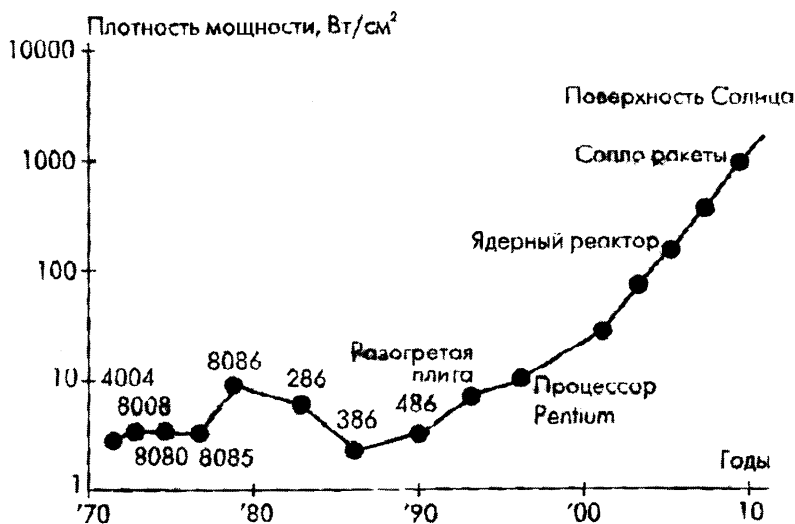


Рис. Д.2.3. Временная зависимость плотности мощности микропроцессоров.

Величина $P_{\text{доп}}$ определяется конструкцией корпуса интегральной схемы, условиями и конструктивными средствами теплоотвода.

Механизм производства информации посредством электронных схем заключается в формировании различных электрических цепей в различные временные интервалы. Электрические цепи обязательно включают резисторы. Переключатели поочередно подключают выход вентиля через эти цепи то к шине питания, то к общей шине. По этим цепям протекают токи и, согласно закону Джоуля-Ленца, превращается в тепло электрическая энергия, определяемая выражением:

$$W = I^2 R. \quad (\text{Д.2.6})$$

Этот процесс диссипации энергии, поставляемой источником питания неизбежно происходит во всех типах вентилях, представленных в классификационной диаграмме на рис. Д.2.1. Количество диссипированной и термализованной энергии тем не менее существенно различно для разных типов вентилях. Почему термализация энергии неизбежна в цифровых системах, построенных на электронных схемах из традиционных управляемых переключателях, традиционных электронных компонентах (резисторах, конденсаторах, диодах и т.п.). Ответ прост. Электронные компоненты являются «неидеальными». Их шунтируют паразитные сопротивления утечки. Паразитные сопротивления включаются и последовательно с «идеальными» элементами. Сопротивления утечки включены последовательно и/или параллельно конденсаторам, диодам и, конечно, переключателям (транзисторам). Сопротивления утечки имеют большую величину (порядка $10^9 - 10^{12}$ Ом), но, тем не менее, они в значительной мере определяют энергетику цифровых устройств обработки информации, выполненных в виде ультра БИС.

Действительно, даже при $E_n = 1$ В ток утечки одного вентиля составляет величину I_i ($10^{-9} - 10^{-12}$) А, а в устройствах, выполненных в виде ультра БИС, ток утечки будет равен NI_i и при $N = 10^8$ составляет ($10^{-1} - 10^4$) А. Параллельные паразитные сопротивления утечки шунтируют электронно-дырочные переходы и диэлектрические области переключателей. Физические механизмы, обуславливающие эти утечки тока определяются туннельным эффектом, термоэлектрической эмиссией, генерацией носителей в области пространственного заряда и другими физическими эффектами.

Для различных переключателей-транзисторов эти эффекты будут рассмотрены в последующих разделах дополнения. Не вдаваясь пока в детали физических механизмов токов утечки, отметим, что их роль в энергетике производства информации постоянно возрастает при уменьшении геометрических размеров переключателей как топологических, так и структурных.

Параллельные паразитные сопротивления утечек электронно-дырочных переходов определяются в значительной мере поверхностными эффектами и плохо контролируруемыми технологическими процессами очистки поверхности. Величины сопротивлений утечки определяются достигнутым технологическим уровнем. Их можно уменьшить, но полностью устранить невозможно.

Последовательные паразитные сопротивления представляют собой сопротивления рабочих полупроводниковых областей компонен-

тов, контактов к ним и внутрисхемных соединений. При субмикронных размерах элементов и контактных окон к ним сопротивления становятся недопустимо большими.

Уменьшать же сопротивление этих областей путем увеличения уровня их легирования нельзя, поскольку практически предельные уровни концентрации примесей уже достигнуты (при больших концентрациях наступает вырождение полупроводника, приводящее к снижению подвижности носителей и ухудшению характеристик транзистора).

Количество потребляемой энергии и скорость ее потребления (мощность) от источников питания являются главными энергетическими характеристиками логических схем.

Общепринятым основным параметром, характеризующим энергетiku логического вентиля является энергия единичного переключения W_0 , приближенно определяемая как средняя энергия производства единицы информации приближенно задаваемая выражением:

$$W_0 = P\tau_{\text{ср}},$$

где: P — мощность вентиля; $\tau_{\text{ср}}$ — среднее время задержки переключения вентиля.

Энергия единичного переключения является величиной постоянной для вентиля определенного схемотехнического типа, конструкции, физической структуры и технологии. Величина W_0 определяется только параметрами вентиля (величиной удельных значений паразитных емкостей и резисторов, геометрическими размерами и т.п.) и не зависит от режимов его работы.

Скорость потребления энергии от источника питания (мощность P) логического вентиля может варьироваться в широких пределах, но увеличение мощности неизбежно приводит к уменьшению величины $\tau_{\text{ср}}$ и наоборот, таким образом, эта величина энергии единичного переключения остается постоянной, то есть:

$$W_0 = \text{const.}$$

Характеристика является объективной характеристикой совершенства логического вентиля. И поэтому широко используется для сравнения технологий.

В общем случае в традиционных интегральных цифровых устройствах с полной диссипацией энергии вся энергия, потребляемая от источников питания рассеивается на резистивных элементах электронной схемы, превращаясь в тепло.

Для них справедливо равенство выражения

$$Q = \sum_{i=1}^N W_{cpi}^1 f_i t, \quad (\text{Д.2.7})$$

где Q — количество тепла, выделяемого кристаллом интегрального цифрового устройства за время t ;

W_{cpi}^1 — средняя энергия переключения i -того вентиля;

f_i — частота переключения i -того вентиля.

В подавляющем большинстве известных в настоящее время логических схем энергия расходуется полностью, что порождает одну из важнейших проблем — обеспечение теплоотвода. Стабилизация температуры электронных устройств требует использования эффективных конструктивных средств. Это усложняет аппаратуру. Увеличивается ее вес и габариты, уменьшается эксплуатационная надежность. Для современной электроники, таким образом, проблема уменьшения энергии переключения вентиля становится проблемой номер один. Острая необходимость решения этой проблемы стимулирует поиск новых более энергетически эффективных решений и на уровне элементарных логических вентилях. Одним из возможных путей повышения энергетической эффективности является разработка схемотехнических решений вентилях, в которых энергия, поставляемая источником питания, расходуется в процессе функционирования лишь частично превращаясь в тепло.

Идея создания логических вентилях и, соответственно, цифровых устройств на их основе базируется на возвращении части энергии, уже использованной для производства информации, обратно в источник энергоснабжения. Для подобных энергетически реверсивных вентилях термализуется лишь часть энергии, а возвращаемая энергия определяется выражением:

$$W^R = (1 - v) \sum_{i=1}^N W_{cpi}^1 f_i t, \quad (\text{Д.2.8})$$

Для полностью диссипирующих вентилях $v = 1$, для адиабатических вентилях $v = 0$. На полностью адиабатические вентилях практически не реализуемы, поэтому можно говорить об асимптотически адиабатических вентилях, для которых $v \rightarrow 0$.

Основные идеи и подходы создания реверсивной логики будут изложены в последующих разделах.

Д.2.3. Скорость производства информации

Скорость производства информации на уровне логических вентилей определяется их схемотехнической организацией, топологией и, главным образом, быстродействием элементарных переключателей-транзисторов.

Быстродействие транзисторов, характеризующиеся временами переключения из открытого состояния в закрытое и наоборот, определяется принципом действия, физической структурой и топологическими размерами. В данном разделе будут рассмотрены принципы функционирования основных типов переключателей и их принципиальные особенности.

На протяжении сорока пяти лет переключатели совершенствовались. Этот процесс носит эволюционный характер — прежде всего уменьшаются размеры классических транзисторов и совершенствуются их физическая структура. Одновременно с этим процессом ведутся интенсивные поиски новых принципов функционирования, которые обеспечили бы более высокое быстродействие. В настоящее время известно большое число транзисторов различных типов, разобраться с которыми удобно с помощью классификационной диаграммы, приведенной на рис. Д.2.4.

Первым классификационным признаком транзисторов как переключателей для цифровых схем являются принцип действия.

В транзисторах скоростные и усилительные свойства обеспечиваются специфическими физическими процессами, явлениями и эффектами, протекающими в монокристаллических и поликристаллических полупроводниковых материалах. В результате взаимодействия этих процессов и их определенной последовательности оказывается возможным управлять величиной тока между токопроводящими (управляемыми) электродами, посредством напряжения и/или тока на входном электроде. Причем малые величины входных воздействий позволяют управлять токами между токопроводящими электродами в широком диапазоне.

Для обеспечения высокого быстродействия элементарных переключателей необходимо чтобы транзисторы были способны коммутировать возможно большие токи при возможно меньших управляющих напряжениях.

Количественно это свойство транзистора как переключателя-усилителя определяется параметром, называемым крутизной и пре-

образования:

$$K = \frac{\partial I_{\text{вых}}}{\partial U_{\text{вх}}}.$$

Видами физических процессов определяются также такие важные характеристики переключателя как сопротивления между токопроводящими электродами в открытом $R_{\text{от}}$ и закрытом $R_{\text{зак}}$ состояниях.

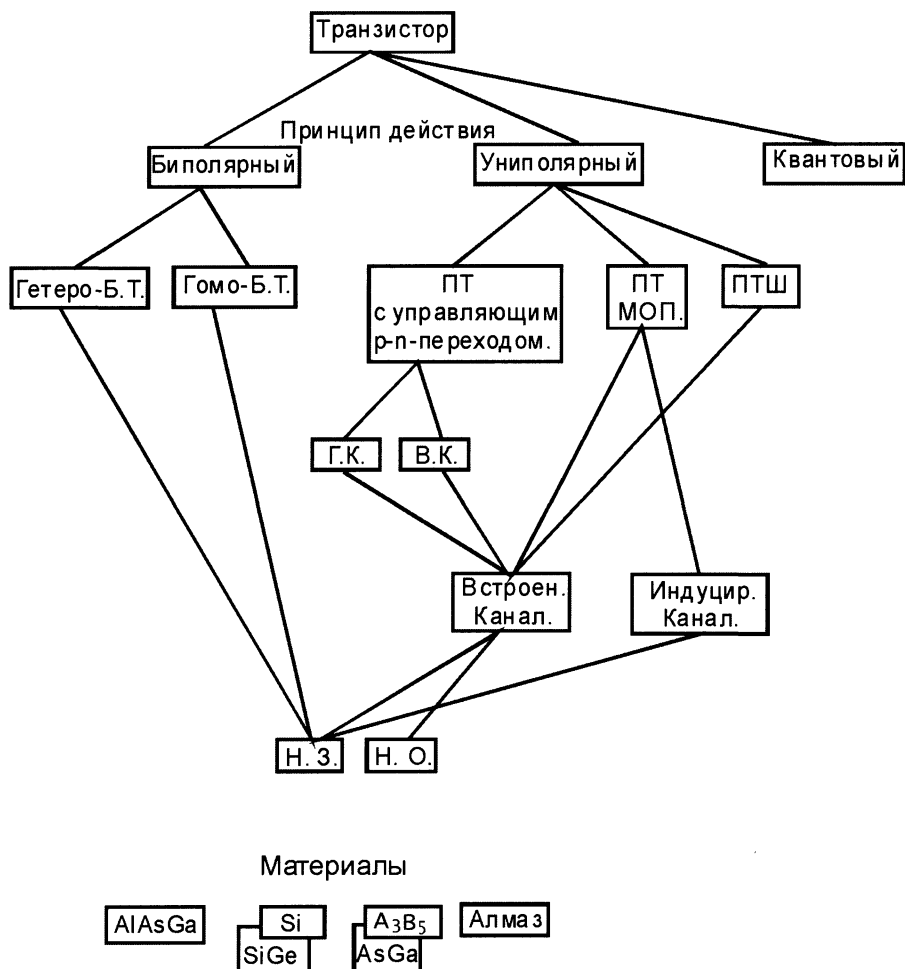


Рис. Д.2.4. Классификационная диаграмма транзисторов как переключателей.

Физические процессы определяют также сколько энергии необходимо затратить на отпирание и запираание переключателя и за какое время это можно осуществить. Скоростные качества транзи-

сторков определяются характерными рабочими частотами. По принципу действия транзисторы подразделяются на два класса биполярные и униполярные.

В биполярных транзисторах под воздействием входных сигналов протекают процессы инжекции неосновных носителей заряда, их рекомбинация с основными носителями в процессе дрейфа и диффузии, коллектирование носителей заряда.

Эти процессы обуславливает перемещение и накопление электронов и дырок в структуре биполярного транзистора, токи и потенциалы на внешних токопроводящих электродах транзисторов в зависимости от внешнего воздействия на управляющем электроде.

Особенностями биполярного транзистора как переключателя, вытекающими из его принципа действия, являются высокая величина крутизны преобразования, малые величины сопротивления в открытом состоянии и высокие значения сопротивления в закрытом состоянии.

Крутизна преобразования определяется следующим выражением:

$$K_{BT} = \frac{\partial I_K}{\partial U_{БЭ}},$$

где I_K — ток коллектора, $U_{БЭ}$ — напряжение между базой и эмиттером.

Высокая величина крутизны преобразования обусловлена процессом инжекции неосновных зарядов, их дрейфом и диффузией, обуславливающих близкие к единице значения коэффициента передачи тока и экспоненциальной зависимостью тока от входного напряжения. Малые величины сопротивления между токопроводящими электродами эмиттера и коллектора в открытом состоянии обусловлены, так называемым, режимом насыщения. В этом режиме сопротивление идеального биполярного транзистора равно нулю. В реальном же транзисторе это сопротивление определяется паразитным сопротивлением области коллектора. Поэтому открытый биполярный транзистор способен пропускать между электродами эмиттера и коллектора токи больших величин без существенного падения напряжения между ними. Такова позитивная сторона режима насыщения, специфического режима работы биполярного транзистора. Негативная сторона режима насыщения заключается в том, что в этом режиме происходит накопление избыточного заряда неосновных носителей. Этот избыточный заряд увеличивает время закрывания транзистора на величину времени его рассасывания.

Другими факторами, определяющими быстродействие биполярных транзисторов являются паразитные емкости.

В униполярных транзисторах, в отличие от биполярных, принцип действия базируется на управлении входным воздействием (полем) потоком носителей одного знака или электронов, или дырок. Униполярные транзисторы называются полевыми, причем последнее название используется чаще.

Ток в полевых транзисторах, поскольку они являются униполярными, переносится только основными носителями и паразитный эффект накопления неосновных носителей в них отсутствует. За исключением полевых транзисторов с управляющим p - n переходом. Это первая особенность полевых транзисторов как переключателей, вытекающая из физического принципа действия. Быстродействие полевых транзисторов определяется сопротивлением канала, прямо пропорциональным его длине и паразитными емкостями. От длины канала зависит также и величина крутизны преобразования, которая в данном случае определяется выражением:

$$K_{ПТ} = \frac{\partial I_c}{\partial U_{зн}},$$

I_c — ток стока, ток между токопроводящими электродами стока и истока, $U_{зн}$ — напряжение на управляющем электроде (затворе).

Вторым классификационным признаком можно принять структуру транзистора. По данному принципу биполярные транзисторы подразделяются на гомо и гетеро структурные. Особенностью биполярных транзисторов, вытекающей из физической структуры, является необходимость их изоляции друг от друга при использовании в составе интегральной схемы. Средства изоляции (обратносмещенный p - n переход, диэлектрические области и комбинации p - n переходов и диэлектриков) являются вспомогательными элементами конструкции, которые не определяют выполнение функций ключа, но требуют дополнительных затрат площади и вносят паразитные емкости, от которых зависит их быстродействие. Известны и широко используются в интегральных схемах цифровых устройств три типа изоляции: изоляция обратносмещенным p - n переходом, полная диэлектрическая и комбинированная (часть структуры изолируется диэлектриком, другая — p - n переходом).

Полевые транзисторы реализуются в трех структурных типах. Наиболее широко распространенным типом является транзистор со структурой «металл-окисел полупроводник» (МОП-транзистор). МОП-тран-

зистор имеет две структурные разновидности: со встроенным каналом и индуцированным каналом.

Особенностью полевых транзисторов данного типа и обоих структурных разновидностей является их самоизоляция от подложки и, следовательно, друг от друга при использовании в составе интегральных схем цифровых устройств. Это свойство позволяет тратить для размещения переключателей меньшую площадь поверхности кристалла по сравнению с биполярным транзистором.

Второй тип полевого транзистора — полевой транзистор с управляющим $p-n$ переходом в структурном отношении занимает промежуточное положение между биполярным транзистором и полевым МОП-транзистором. Данный тип полевого транзистора имеет две структурные разновидности. Он может быть выполнен с горизонтальным или вертикальным встроенными каналами.

Третьим классификационным признаком транзисторов — переключателей является вид вольт-амперных характеристик (ВАХ). ВАХ бывает двух принципиально различных типов: нормально закрытого и нормально открытого. Транзисторы различных принципов действия и различных структурных видов, которые закрыты при напряжении на управляющем электроде, равном нулю, являются нормально закрытыми (НЗ). Типичные ВАХ транзисторов НЗ типа приведены на рис. Д.2.5 и Д.2.6.

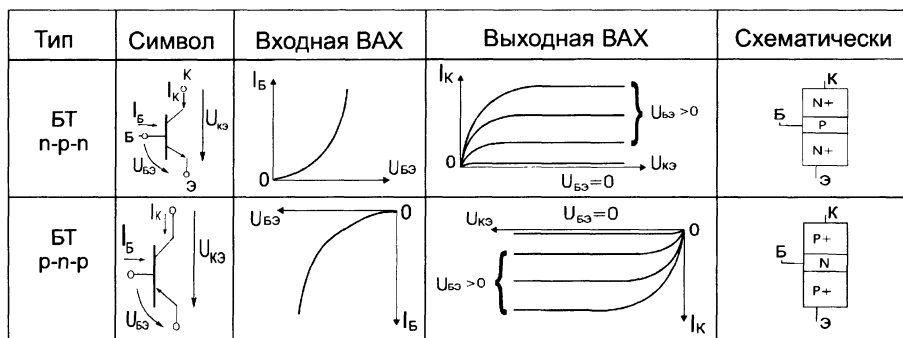


Рис. Д.2.5. ВАХ биполярных транзисторов.

К НЗ транзисторам относятся биполярные транзисторы, полевые транзисторы с управляющим $p-n$ переходом, МОП транзисторы с индуцированным каналом и полевые транзисторы Шоттки. К транзисторам нормально открытого типа (НО) относятся полевые транзисторы со встроенным каналом. Типичные ВАХ транзисторов НО типа приведены на рис. Д.2.6.

Следствием принципиальных различий в НЗ и НО ВАХ является то, что транзисторы как переключатели могут быть разделены на приборы, управляемые током и приборы, управляемые напряжением. Это обстоятельство является очень важным с точки зрения энергетики производства информации. Дело в том, что отпирание переключателя НЗ типа неизбежно сопровождается потреблением тока и, следовательно, энергии по входной цепи (см. рис Д.2.5 и Д.2.6). В подавляющем большинстве переключателей эта энергия термализуется по мере потребления. Другое дело в переключателях НО типа. В этих приборах запираение производится путем подачи на управляющий электрод (затвор полевого транзистора) напряжения. В любом из известных типов НО элементарных переключателей — МОП транзисторов, ПТШ и полевом транзисторе с управляющим p - n -переходом ток по входной цепи не потребляется. Следовательно, не потребляется и энергия от источника питания.

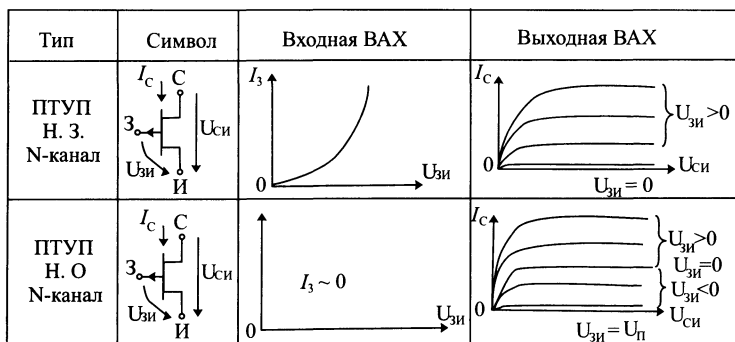


Рис. Д.2.6. ВАХ полевых транзисторов с управляющим переходом (ПТУП).

Энергия накапливается на входной емкости (емкости затвористок) транзистора. Важно подчеркнуть, не рассеивается по мере отпирания, как это имеет место в биполярных транзисторах, а накапливается. Это обстоятельство создает предпосылки для ее повторного использования при производстве информации (см. раздел Д.6).

Повышение скорости обработки цифровой информации, прежде всего, обязано совершенствованию структур.

Прогресс в области микроэлектроники в целом и в области цифровой техники в частности на протяжении более пятидесяти лет обусловлен в основном за счет совершенствования транзисторов на базе развития технологии.

Совершенствование транзисторов ведется по следующим стратегическим направлениям. Первое — уменьшение геометрических

размеров классических типов транзисторов традиционных конструкций. Уже более тридцати лет уменьшение топологических и структурных размеров ведется путем простого масштабирования.

То есть одновременно с уменьшением длины канала полевого транзистора, размеров контактных окон и электродов к рабочим областям затвора, истока и стока уменьшается и толщина подзатворного диэлектрика и глубины залегания p - n переходов (или толщины областей).

В биполярном транзисторе одновременно с уменьшением размеров эмиттера, базы, коллектора и электродов к ним уменьшаются и глубины залегания p - n переходов эмиттер-база и коллектор-база.

Второе стратегическое направление — создание полной диэлектрической изоляции с целью уменьшения паразитных емкостей структур переключателей и разработка новых конструктивно-топологических и структурных решений в рамках классических принципов действия.

Третье стратегическое направление — разработка новых переключателей на квантово-механических принципах функционирования на основе гетероструктур с нанометровыми размерами в рамках полупроводниковой технологии.

И, наконец, четвертое направление — исследование возможностей использования в качестве переключателей молекул, полимерных и других материалов и структур, создаваемых на базе нанотехнологий.

Наиболее распространенной типовой конструкцией МОП-транзистора является LDD -структура (*Lightly Doped Drain*), приведенная на рис. Д.2.7. Параметры транзисторной структуры при различных проектных нормах приведены в табл. Д.2.1. Под проектной нормой обычно подразумевают минимальный топологический размер (контактного окна, длины канала или другого характерного элемента). Особенностью конструкции является наличие мелких слабелегированных областей n -типа, которые удлиняют области истока и стока в сторону канала. Концентрацию легирующей примеси в этих областях (фосфор и бор) и режимы ее разгонки выбирают таким образом, чтобы получить плавный p - n переход. Обычно концентрация примеси составляет от $4 \cdot 10^{18}$ до $8 \cdot 10^{18} \text{ см}^{-3}$. В n^+ -областях она достигает $(5 \cdot 10^{19} - 1 \cdot 10^{20} \text{ см}^{-3})$. Наличие слабелегированных областей снижает напряженность электрического поля в канале на границе со стоком и уменьшает энергию горячих электронов, которые вызывают деградацию параметров транзистора. Слабелеги-

рованные *LDD*-области также повышают напряжение прокола, инжекционного и лавинного пробоя транзистора и уменьшают эффект модуляции длины канала.

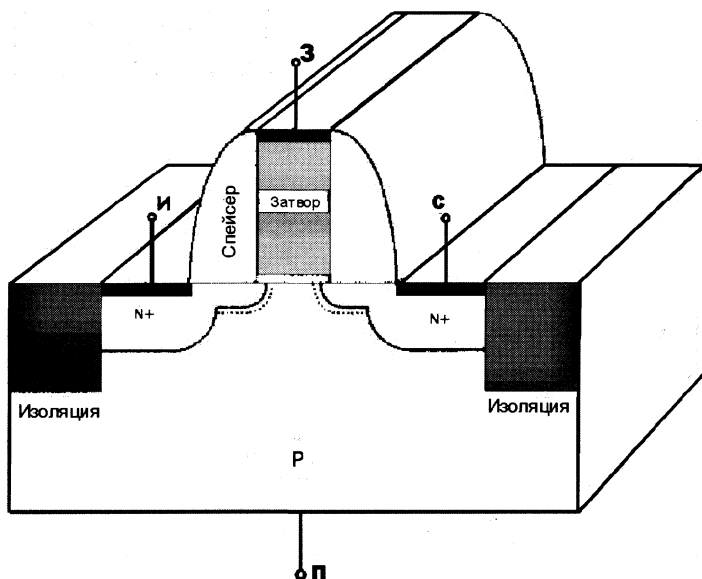


Рис. Д.2.7. Типовая структура МОП-транзистора.

Толщина поликремниевого затвора составляет порядка 300 нм. Для обеспечения малых величин емкостей транзистора выбирают слаболегированную подложку, а для обеспечения необходимого порогового напряжения и снижения напряжения прокола применяют дополнительное легирование канала примесью того же типа, что и в подложке.

Таблица Д.2.1.

Проектные нормы, нм	250	180	130	100	70
Толщина окисла, нм	4 – 5	3 – 4	2 – 3	1,5 – 2	< 1,5
Глубина <i>p-n</i> переходов, нм	50 – 100	36 – 72	26 – 52	20 – 40	15 – 30
Ток утечки, нА/мкм	1	1	3	3	10
Мощность/кристалл, Вт	70	93	121	120	

Контакты к областям истока, стока и к поликремниевому затвору выполняют с промежуточным формированием слоя TiSi_2 или CoSi_2 толщиной порядка 40 нм, что обеспечивает удельное сопротивление около 5 Ом на квадрат. Изоляция между поликремниевым

затвором и контактами к истоку и стоку выполняется в виде спейсера из S_3N_4 . Концентрация примеси в приповерхностной области, где индуцируется канал, составляет $5 \cdot 10^{17} — 1 \cdot 10^{18} \text{ см}^{-3}$. Увеличение концентрации примеси свыше этого значения, необходимое для транзисторов с длиной канала менее 100 нм, ведет к появлению туннелирования электронов через p - n переходы истока и стока.

Толщина окисла для транзисторов с длиной канала 0,1 мкм составляет 1,5–2 нм. Между толщиной окисла t_{OX} и длиной канала L транзисторов данной структуры существует эмпирическая зависимость $L = 45 \cdot t_{OX}$.

Для формирования изолирующих карманов МОП-транзисторов разного типа проводимости используют фосфор и бор. Изоляцию между карманами выполняют обычно мелкими канавками, стенки которых окисляют, а внутренность заполняют поликремнием. Эта технология изоляции стала доминирующей в транзисторах, выполненных по 0,25-микронной технологии и пришла на смену изоляции локальным окислением кремния (*LOCOS*).

Данная типовая структура транзистора обеспечила снижение длины затвора от 10 мкм в 70-х годах до 0,06 мкм в настоящее время путем масштабирования. Однако переход проектных норм через границу 130 нм в рамках данной типовой структуры транзистора наталкивается на физические ограничения. Уменьшение толщины окисла приводит к росту туннельного тока утечки затвора, уменьшение глубины залегания p - n переходов — к росту последовательного сопротивления областей транзистора. Уменьшение глубины p - n переходов до 10 нм приводит к увеличению сопротивления слоев истока и стока до 10 кОм/квадрат, что ограничивает коммутирующую способность транзистора по току. При уменьшении длины канала и порогового напряжения растёт подпороговый ток.

МОП-транзисторы обладают свойством самоизоляции от подложки и нет принципиальной необходимости в специальных конструктивных элементах для их изоляции.

Однако для повышения быстродействия путем минимизации барьерных емкостей изоляция нужна. Идея полной диэлектрической изоляции появилась на заре развития микроэлектроники, но большие технологические проблемы не позволяли ее реализовать. Работы, тем не менее, продолжались. Наконец, в конце девяностых годов появилась промышленная технология полной диэлектрической изоляции. На базе этой технологии стало развиваться второе стратегическое направление совершенствования транзисторов и цифровых устройств на их основе.

МОП-транзисторы, изготовленные по технологии «кремний на изоляторе» (КНИ) обеспечивают более высокие скорости обработки цифровой информации, поскольку наличие толстого окисла вместо кремния под областями истока и стока существенно уменьшает величины емкостей на подложку. Типовая структура транзистора приведена на рис. Д.2.8. КНИ-структуры отличаются высокой радиационной стойкостью и повышенной надежностью при высоких температурах. Короткоканальные эффекты в КНИ-приборах могут быть подавлены простым уменьшением толщины кремниевого слоя. Транзисторы с малыми утечками получаются на пленках кремния толщиной порядка 10 нм.

Основным недостатком данной структуры транзистора является то, что он имеет увеличенный подпороговый ток вследствие эффекта плавающей подложки. Этот недостаток осложняет понижение потребляемой мощности в выключенном состоянии транзисторного ключа.

Для КНИ-структур используют три способа изоляции: локальное окисление кремния (*LOCOS*), изоляция мелкими канавками (*STI*) и меза-изоляция.

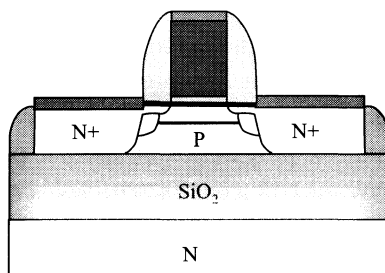


Рис. Д.2.8. КНИ структура транзистора.

КНИ-структура, приведенная на рис. Д.2.8, имеет длину канала 0,28 мкм и ширину 9,1 мкм. Толщина слоя изолирующего окисла составляет 390 нм, толщина слоя кремния на окисле равна 190 нм. Подзатворный окисел имеет толщину 4,7 нм, толщина поликремниевого затвора — 300 нм. Кармашки, легированные бором для *n*-канальных транзисторов и фосфором — для *p*-канальных, необходимы для предохранения от смыкания истока и стока и подавления эффекта снижения порогового напряжения с уменьшением длины канала. На области истока, стока и затвора имеется слой силицида TiSi_2 толщиной 50 нм для уменьшения сопротивления контактов. Контакты к областям транзистора выполнены из вольфрама. Изо-

ления между транзисторами на кристалле выполняется из окисла кремния.

Одной из проблем изготовления транзисторов на тонких пленках кремния является высокое последовательное сопротивление областей истока и стока. Для его уменьшения используют самосовмещенный силицидный процесс, использующий силицид титана или кобальта.

На рис. Д.2.9. приведен разрез физической структуры еще одного варианта транзистора со структурой типа КНИ.

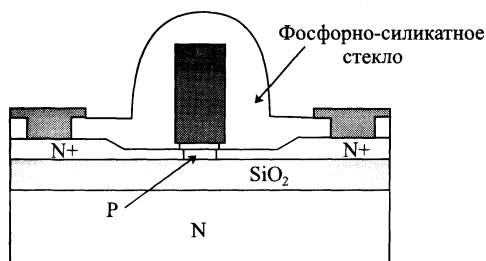


Рис. Д.2.9. МОП-транзистор с полной диэлектрической изоляцией.

Транзистор имеет длину канала 40 нм, изготовлен по технологии КНИ на экстремально тонком слое кремния (было изготовлено три варианта транзисторов с толщиной кремния 4, 11 и 18 нм).

Подзатворный окисел имеет толщину 4,7 нм.

Слой исходного кремния на изоляторе имел удельное сопротивление 3 Ом·см. Толщина слоя заглубленного окисла составляла 100 нм и он был выращен на подложке n -типа с удельным сопротивлением 0,02 Ом·см. Пленка кремния для формирования областей истока и стока имела толщину 80 нм. В ней селективным травлением была получена область толщиной от 4 до 18 нм, в которой впоследствии была сформирована область, в которой индуцируется канал транзистора. Таким образом, при тонком слое кремния для подзатворной области толщины области истока и стока оказались достаточно большими, что обеспечило их низкое омическое сопротивление.

В рамках второго стратегического направления — совершенствования транзисторов на базе структур кремний на изоляторе появились идеи создания трехмерных конструкций.

Для уменьшения длины канала МОП-транзисторов с традиционной планарной структурой правила масштабирования диктуют необходимость уменьшать глубину залегания p - n переходов и толщину подзатворного окисла. Однако для транзисторов с длиной ка-

нала менее 100 нм толщина подзатворного диэлектрика становится настолько тонкой, что ток туннелирования через него становится недопустимо большим и определяющим мощность, потребляемую СБИС в стационарных режимах. Мелкие p - n переходы приводят к большому сопротивлению контактных областей. Для предотвращения смыкания областей истока и стока неизбежно применение высоколегированного ($> 10^{18} \text{ см}^{-3}$) стопора. Однако стопор снижает пропускную способность транзистора по току и увеличивает токи в подпороговой области.

Таким образом метод уменьшения размеров путем масштабирования типовой планарной конструкции МОП транзистора исчерпал свои возможности. Для совершенствования транзистора был выбран путь создания трехмерных конструкций с двойным или окольцовывающим затвором. В трехмерных конструкциях затвор с двух (или со всех) сторон охватывает область канала. Это позволяет эффективнее управлять потенциальным барьером между истоком и стоком и существенно ослабить большинство короткоканальных эффектов в транзисторах с проектными нормами менее 50 нм. Уменьшается также емкость p - n переходов, увеличивается радиационная стойкость.

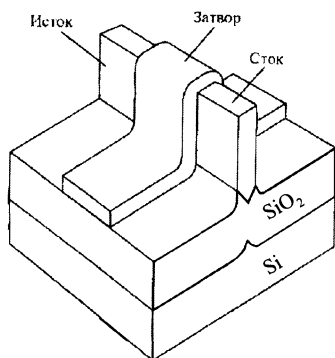


Рис. Д.2.10. МОП-транзистор с двойным затвором.

Принцип действия транзистора *DELTA* с двойным затвором иллюстрируется рис. Д.2.10. На толстом окисла создается островок кремния в форме бруска, который служит каналом транзистора. Затвор охватывает область канала с трех сторон. Это обеспечивает большую крутизну и малые токи утечки в подпороговой области. Канал транзистора получается сильно обедненным.

На рис. Д.2.11. приведена иллюстрация другого варианта развития идеи трехмерных транзисторных структур.

В транзисторах с окольцовывающим затвором канал ориентирован перпендикулярно поверхности кристалла, и затвор со всех сторон окружает канал (*Surrounding Gate Transistor, SGT*). Такая структура обеспечивает независимость длины канала от разрешающей способности литографического оборудования.

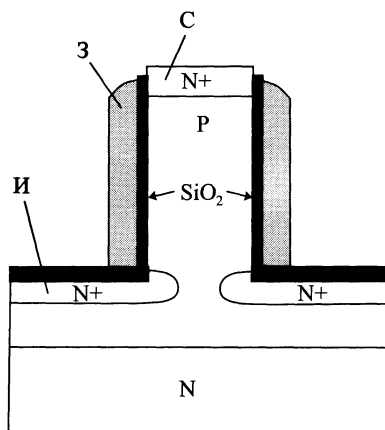


Рис. Д.2.11. МОП-транзистор с вертикальным каналом.

Такая структура имеет минимальные подпороговые токи и большую передаточную проводимость. Вертикальное расположение канала обеспечивает высокую степень интеграции.

Интегральные биполярные транзисторные структуры современных СБИС в подавляющем большинстве случаев относятся к планарно-эпитаксиальным с боковой диэлектрической изоляцией.

На рис. Д.2.12 приведен разрез физической структуры биполярного транзистора с так называемой щелевой изоляцией, которая является разновидностью боковой диэлектрической изоляции. В данной структуре толщина коллекторной области составляет порядка 1 мкм., глубина залегания перехода база-коллектор колеблется в пределах 0,4–0,5 мкм. И глубина залегания эмиттерного перехода — несколько десятых долей мкм.

Д.3. Организация и энергетика цифровых схем без отношения

Представление информации в виде уровней напряжений «0», «1» и ее обработка в цифровых логических схемах требует затрат энергии. В различных электронных схемах эти уровни формируются по

разному, соответственно и различные энергетические затраты на обработку и хранение цифровой информации.

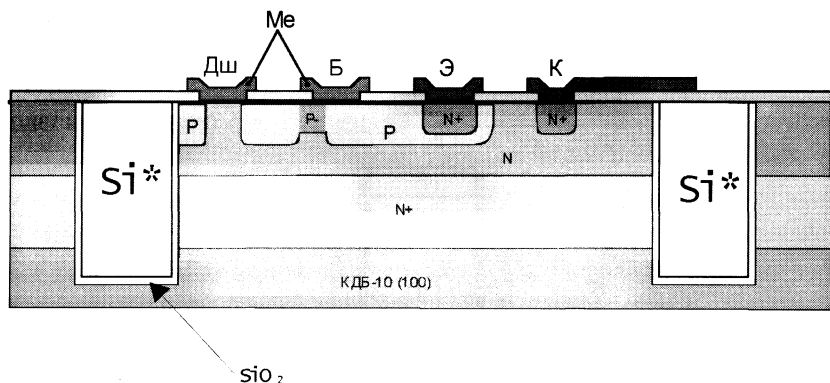


Рис. Д2.12. Разрез структуры биполярного транзистора.

Принцип организации логических схем без отношений является наиболее выгодным с энергетической точки зрения для цифровых устройств с традиционными источниками питания и стал доминирующим для создания цифровых устройств в виде сверхбольших интегральных схем с количеством элементов на кристалле порядка 10^6 и более.

Д.3.1. Общий принцип организации схем и схемотехнические разновидности

Принцип организации схем без отношения иллюстрируется обобщенной схемой инвертора, приведенной на рис. Д.3.1.

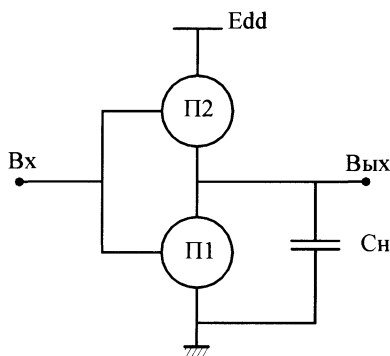


Рис. Д.3.1. Обобщенная схема инвертора без отношения.

По определению схемами без отношения принято считать схемы логических вентилях, в которых напряжение логического нуля $U_{«0»}$ не зависит от соотношения внутренних (паразитных) сопротивлений переключателей П1 и П2. Напряжение $U_{«0»}$ определяется напряжением общей шины, поскольку когда открыт переключатель П1, переключатель П2 закрыт и токи в схеме не протекают (если пренебречь токами утечки).

Общим принципом формирования напряжения логической единицы $U_{«1»}$ логических вентилях данного класса заключается в формировании напряжения на нагрузочной емкости C_n при ее заряде током, протекающим через открытую цепь заряда. Формирование уровня $U_{«0»}$ происходит при разряде нагрузочной емкости, током, протекающим через открытую цепь разряда. На рис. Д.3.2 цепь заряда включает сопротивление R_2 переключателя П2, а цепь разряда — сопротивление R_1 переключателя П1.

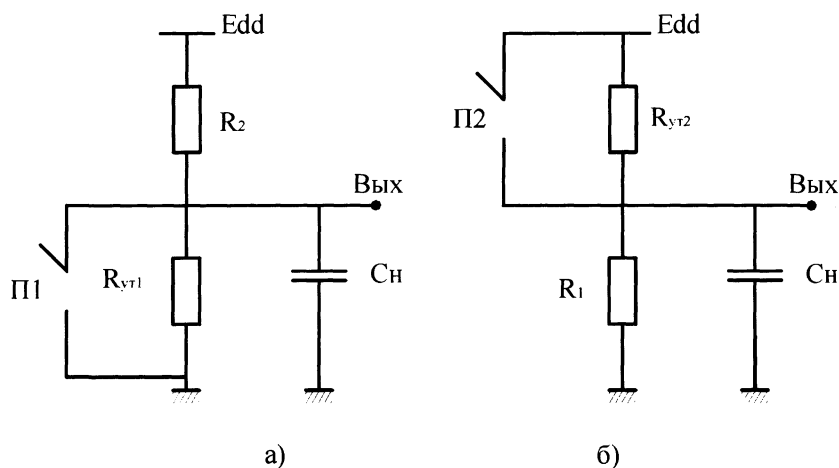


Рис. Д.3.2. Цепи заряда а) и разряда б) нагрузочной емкости C_n в схеме без отношения.

В качестве переключателей П1 и П2 используются транзисторы с взаимодополняющим типом проводимости или полевые транзисторы с каналами п и р типа проводимости или биполярные транзисторы п-р-п или р-п-р типа проводимости. Это позволяет объединить их управляющие электроды и обеспечить при этом в стационарных состояниях открытое состояние одного переключателя и закрытое состояние другого.

Следствием этого является разомкнутое состояние цепи между шиной питания и общей шиной в обоих стационарных состояниях и

отсутствие токов. Для представления информации на выходе в виде напряжений $U_{«0»}$ и $U_{«1»}$ поэтому в принципе не требуется затрат энергии. Но поскольку переключатели неидеальны, параллельно с их токопроводящими электродами имеются паразитные сопротивления утечек R_{ym1} и R_{ym2} и, строго говоря, в стационарных состояниях протекают токи утечек и потребуется энергия. Физические механизмы, определяющие величины токов различны для различных типов переключателей и будут рассмотрены ниже при рассмотрении схемотехнических решений. Отметим только, что с уменьшением размеров переключателей ток и утечки начинают все больше и больше определять энергетику вентилях комплементарной логики.

Другой особенностью схем без отношения, определяющей их преимущества, является максимальный логический перепад

$$\Delta U = U_{«1»} - U_{«0»} = E_{dd}$$

Максимальный логический перепад обуславливает высокую помехоустойчивость цифровых устройств на основе вентилях без отношения.

Схемотехнические разновидности схем без отношения. Общий принцип организации логических вентилях без отношения был воплощен во множестве конкретных схемотехнических решений, использующих практически все типы известных транзисторов.

Вначале были изобретены вентилях, использующие кремниевые комплементарные полевые транзисторы MOS типа. При малой степени интеграции и особенно для быстродействующих применений CMOS схемы проигрывали в конкурентной борьбе другим типам логических вентилях (вентилям с отношением и токовыми переключателями). Причины этого заключались в том, что вопросы энергетики не были определяющими, а технология изготовления была сложной и дорогой.

По мере роста степени интеграции ситуация менялась. При достижении степени интеграции порядка 10^4 вентилях на кристалле и после преодоления серьезных технологических трудностей вентилях типа CMOS нашли широкое применение в цифровой технике благодаря именно их основному достоинству, отмеченному выше.

Эволюция элементной базы микроэлектроники закономерно привела к широкому использованию в СБИС наиболее энергетически экономичного схемотехнического решения. При степени интеграции порядка 10^6 вентилях на кристалле энергетические параметры стали первостепенными, и поэтому CMOS вентилях стали основной

элементной базой цифровых устройств. CMOS логика представляет собой блестящее схемотехническое решение, что нельзя сказать о его структурно-топологических воплощениях. Эти вентили имеют сложную структуру (см. рис. Д.3.3) и обладают сравнительно невысокой плотностью компоновки. При одинаковых нормах проектирования плотность компоновки CMOS схем в четыре раза уступает логике с отношением и в три раза И²Л.

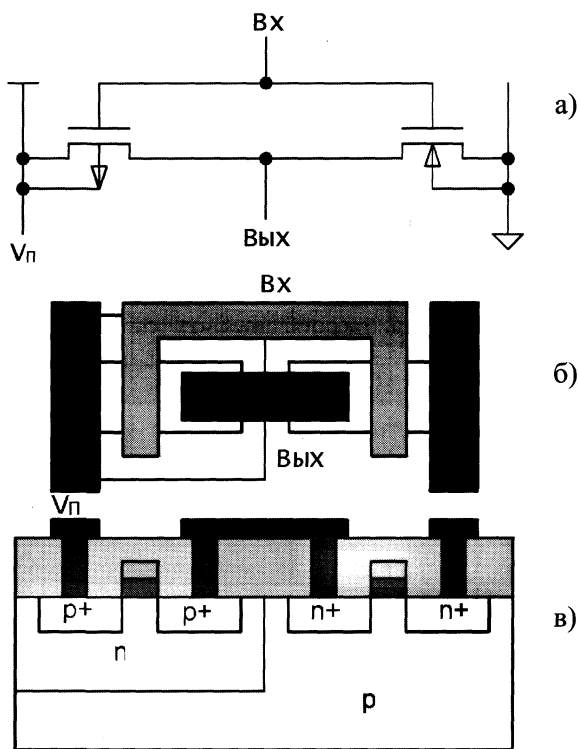


Рис. Д.3.3. Электрическая схема (а), топология (б) и разрез структуры (в) CMOS-инвертора.

При уменьшении размеров в область глубокого субмикрона в традиционных конструкциях CMOS элементов сильнее проявляются паразитные эффекты, нарушающие их нормальное функционирование. Поэтому традиционные структурно-топологические решения входят в противоречие с одним из основных законов микроэлектроники — при уменьшении размеров улучшаются все параметры приборов. Поэтому велись и продолжаются поиски более совершенных структурных решений вентилях без отношений. Как показы-

вает морфологический анализ, в рамках общего схемотехнического принципа построения базового логического элемента на комплиментарных транзисторах возможны и иные схемотехнические решения, подобные классическому CMOS.

Множество этих модификаций содержит различные комбинации известных в настоящее время транзисторов и средств изоляции. Однако, аналогичным CMOS по своему основному достоинству — отсутствие токов в стационарных состояниях является лишь часть схем. К ним относятся только схемы, построенные на нормально закрытых транзисторах, управляемых напряжением и не потребляющих тока по входной цепи.

Таблица Д.3.1.

П1 П2		Si, SOI, Si/Ge				GaAs, GaAlAs/GaAs				X
		n-MOS	n-MES	n-JFET	BT n-p-n	n-MOS	n-MES	n-MOD	n-DJS	?
		1	2	3	4	5	6	7	8	
1	Si	p-MOS	CMOS							
2		p-MES	CMES							
3		p-JFET		CJFET						
4		BT n-p-n			CBL					
5	Ga-As	p-MOS				CMOS GaAs				
6		p-MES					CMES			
7		p-MOD						CMOD		
8		p-DJS							CDJS	
X		?								?

Некоторые интересные модификации приведены в морфологической табл. Д.3.1. В морфологическом ящике (табл. Д.3.1) все конкретные электрические схемы, соответствуют обобщенной структурной схеме, приведенной на рис. Д.3.1, и обладают свойствами классических CMOS вентилях. Морфологический метод гарантированно обеспечивает полноту перебора всех возможных комбинаций известных транзисторов, которые можно использовать в качестве переключателей П1 и П2. При формировании горизонтальной оси морфологического ящика (ось П1 переключательных элементов) использованы транзисторы, в которых ток обусловлен переносом электронов. Эти транзисторы выбраны для целей морфологического анализа ввиду их очевидного преимущества по быстродействию перед р-канальными транзисторами.

Горизонтальную ось содержит в позициях 1–8 нормально-закрытые быстродействующие транзисторы. В данной же работе мы ограничиваемся набором: кремниевых n-MOS, n-JFET, n-MES, n-p-n транзисторов и арсенидгаллиевых n-MOS, n-MES и n-MOD транзисторов. На вертикальной оси морфологического ящика размещены и р-канальные аналоги перечисленных выше транзисторов.

В табл. Д.3.1 каждая из позиций от 1.1 до 8.8 представляет собой схему инвертора, полученную в результате подстановки схемотехнического обозначения переключателей П1 и П2 в обобщенную эквивалентную схему, приведенную на рис. Д.3.1.

Морфологическая таблица содержит все варианты комплиментарных вентилях, построенных на множестве различных типов транзисторов, использованных при морфологическом анализе. Диагональные позиции морфологического ящика содержат классический вариант вентиля без отношения — CMOS (см. рис. Д.3.3) и его приборные разновидности, получившие название CMOS-подобные (CMOS-like) вентили: CMOS-комплиментарная логика на кремниевых и арсенидгаллиевых транзисторах с затвором Шоттки, CJFET комплиментарная логика на кремниевых полевых транзисторах с управляющим p-n переходом; CBL-комплиментарная логика на биполярных транзисторах; CMOD-комплиментарная логика на гетероструктурных транзисторах с двумерным электронным газом; CDJS-комплиментарная цифровая логика на квантовых приборах с эффектом «Кулоновой блокады».

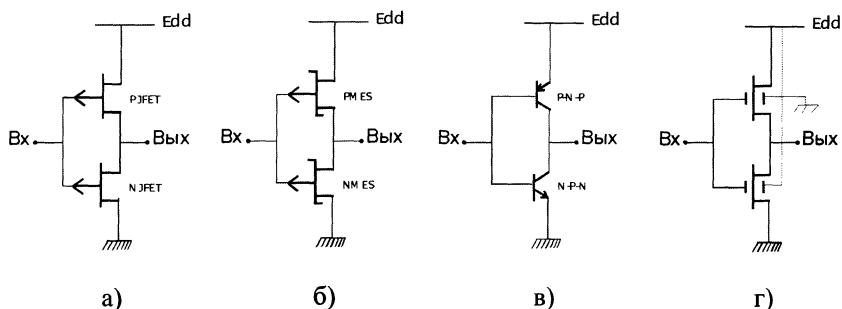


Рис. Д.3.4. Разновидности CMOS-Like вентилях : а) – CJFET; б) – CMES; в) – CBL; г) – CDJS;

Перечисленные выше варианты CMOS-like вентилях являются очевидными разновидностями классического вентиля без отношения, их электрические схемы приведены на рис. Д.3.4. Кроме них возможны и другие варианты сочетаний комплиментарных тран-

зисторов различных типов, расположенные вне диагонали табл. Д.3.1. Исследование таких сочетаний возможно, если транзисторы будут совместимы технологически и совместимы по уровням напряжений и токов. Некоторые из подобных сочетаний технологически не целесообразны в рамках ограничений транзисторной планарной кремниевой технологии, но могут оказаться эффективными, например, при реализации по технологии «кремний на диэлектрике» или в виде «трехмерных» структур. Каждая из разновидностей вентиля CMOS-like имеет несколько структурно-технологических вариантов воплощения. Например, вентиль CJFET может быть воплощен в виде традиционной планарной структуры, включающей транзисторы с горизонтальными или вертикальными каналами. Но такой вариант воплощения CJFET будет иметь низкую плотность компоновки и низкое быстродействие. С использованием структуры типа SOI (кремний на изоляторе) вентиль CJFET типа становится компактным, по плотности компоновки не уступающим классическому CMOS вентилю, а по технологичности и быстродействию превосходящим его. Структуры SOI варианта CJFET логики иллюстрируются схематическими разрезами и топологией приведенными на рис. Д.3.5.

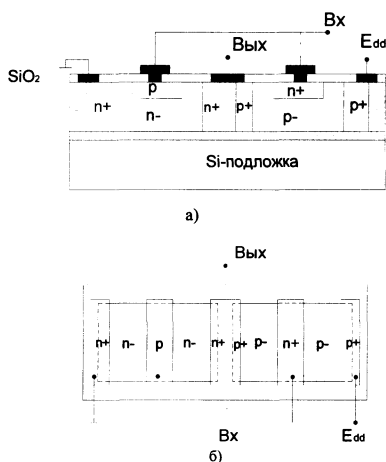


Рис. Д.3.5. Структура SOI CJFET: а) разрез структуры; б) топология.

Очевидным достоинством данного варианта вентиля CJFET по сравнению с классическим CMOS вентилем является технологическая простота. В структуре JFET-транзистора отсутствует один важный конструктивный элемент, без которого невозможен MOS транзистор с изолированным затвором. Этим конструктивным эле-

ментом является тонкий подзатворный диэлектрик, формирование которого является самой прецизионной и сложной технологической операцией во всем технологическом маршруте изготовления. Толщина подзатворного диэлектрика составляет величину порядка единиц нанометров (до 3 атомных слоев). На рис. Д.3.6. приведен другой вариант CJFET вентиля, в котором используются разновидности JFET транзисторов, так называемые, SIT-транзисторы (транзисторы со статической индукцией). Достоинством данного варианта воплощения CJFET логики является, то что длина канала в используемых в ней транзисторах не определяется разрешающей способностью фотолитографии и, следовательно, транзисторы с меньшей длиной канала, а поэтому и с большим быстродействием могут быть реализованы с помощью более дешевого фотолитографического оборудования. Недостатком данного варианта являются плохая технологичность по другим операциям технологического процесса.

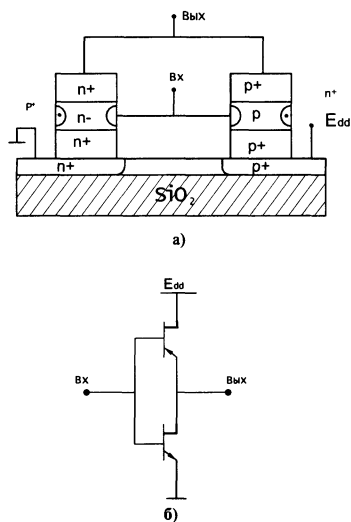


Рис. Д.3.6. Комплементарный вентиль на SIT-транзисторах: а – схематический разрез структуры; б – схемное изображение.

На рис. Д.3.7 приведена иллюстрация еще одного варианта реализации SOI CJFET вентиля, в котором как и в предыдущем варианте используются транзисторы с вертикальным каналом, длина которого не зависит от разрешающей возможности литографии. Но в отличие от него транзисторы имеют планарную конструкцию, что гораздо проще в технологической реализации.

Д.3.2. Энергетика производства информации

Анализ энергетики удобно провести на примере базового логического вентиля, описанного в разделе Д.3.1.

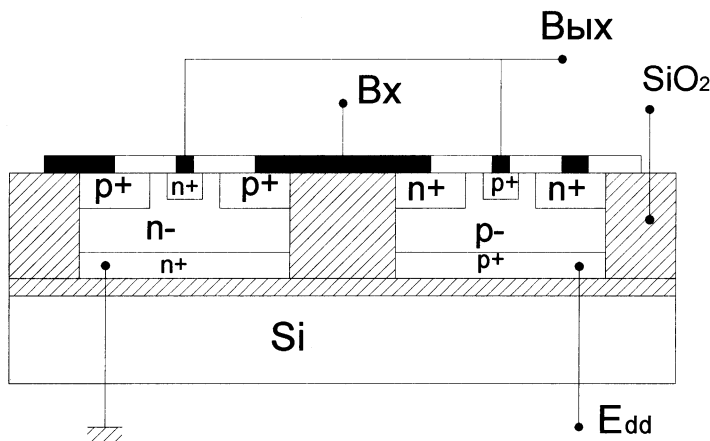


Рис. Д.3.7. Схематический разрез комплементарного вентиля SOI CJFET на транзисторах с вертикальным каналом.

Когда на ходе инвертора имеет место напряжение логического «0», то p-канальный MOS транзистор открыт и от источника питания протекает ток I_{dd} , заряжающий нагрузочную емкость C_H . В этом переходном процессе (отрезок времени t_{10} на рис. Д.2.2) инвертор описывается эквивалентной схемой 3.2.а, из которой видно, что ток заряда протекает через сопротивление R_2 открытого p-канального транзистора. При этом от источника питания потребляется энергия со скоростью, определяемой выражением:

$$P_{dd}^{01} = E_{dd} \cdot I_{dd} \quad (\text{Д.3.1})$$

Величина P_{dd}^{01} определяет скорость потребления энергии W^{0-1} при производстве логической «1». По истечении промежутка времени t_{10} нагрузочная емкость зарядится до напряжения E_{dd} . Это время приближенно определяется выражением:

$$t_{10} = (2 \div 3) R_2 \cdot C_H \quad (\text{Д.3.2})$$

За это время в нагрузочной емкости будет запасена энергия, равная W_c , определяемая выражением:

$$W_c = \frac{1}{2} C_H \cdot E_{dd}^2 \quad (\text{Д.3.3})$$

при заряде нагрузочной емкости средняя величина тока будет определяться выражением:

Следовательно, за время заряда нагрузочной емкости от источника питания была отобрана энергия, определяемая формулой:

$$W^{0-1} = \int_{t_1+t_{10}}^{t_{01}} P_{\partial}^{01}(t) dt \approx E_{dd} \cdot \frac{C_n \cdot E_{dd}}{t_{01}} \cdot t_{01} \approx C_H \cdot E_{dd}^2 \quad (\text{Д.3.4})$$

Из закона сохранения энергии таким образом вытекает, что при производстве информации из-за протекания через R_2 тока при заряде C_H в тепло превратилась энергия, определяемая выражением:

$$W^{0-1} - W_c = \frac{1}{2} C_H \cdot E_{dd}^2 \quad (\text{Д.3.5})$$

Важно отметить, что величина термализованной энергии не зависит от величины сопротивления открытого р-канального транзистора.

В стационарном состоянии, соответствующем логической «1», на выходе (промежуток времени t_1) выходная емкость разряжается через сопротивление утечки R_{ym1} п-канального MOS-транзистора и подзаряжается через сопротивление R_2 открытого р-канального транзистора. Поскольку $R_2 \ll R_{ym1}$, то изменение выходного напряжения не происходит, оно остается практически равным E_{dd} . В данном стационарном состоянии, строго говоря, от источника питания потребляется энергия, поскольку через сопротивление утечки R_{ym1} протекает ток I_{ym} .

Этот ток имеет малую величину порядка (1–2) нА. Но при большом количестве элементов на кристалле общая мощность определяется выражением:

$$P_{cm}^1 = \sum_1^N I_{ym}^1 \cdot E_{dd} \quad (\text{Д.3.6})$$

где: N — количество элементов на кристалле может составить величину, которой нельзя пренебречь.

При изменении напряжения на выходе схемы закрывается р-канальный MOS транзистор, открывается п-канальный MOS транзистор и происходит разряд нагрузочной емкости через сопротивление канала R_1 п-канального MOS транзистора. В процессе разряда емкости запасенная в ней энергия, определяемая выражением (Д.3.3) полностью термализуется на сопротивлении R_1 и других паразитных резистивностях схемы. Эквивалентная схема вентиля в данном

стационарном состоянии приведена на рис. Д.3.2.б. Из схемы видно, что нагрузочная емкость находится в разряженном состоянии поскольку $R_1 \ll R_{ym2}$. При этом имеет место потребление энергии за счет тока утечки через закрытый р-канальный MOS транзистор.

Д.3.3. Мощностные характеристики

Важнейшим параметром, характеризующим энергетику цифровых систем, является скорость потребления энергии от источника питания. Для всех проходных вентилях, за исключением логических вентилях без отношения статического типа этот параметр (мощность) определяется выражением:

$$P = P_{\partial} + P_{ск.m} + P_{ym} + P_{ст} \quad (Д.3.7)$$

В этом выражении содержится четыре компоненты, соответствующие различным интервалам времени при работе вентиля при периодическом входном сигнале.

Динамическая компонента P_{∂} определяется следующим выражением:

$$P_{\partial} = \alpha \cdot C_H \cdot E_{dd}^2 f \quad (Д.3.8)$$

где f — частота входного сигнала;

α — фактор переключательной активности.

Компонента $P_{ск.m}$ обусловлена протеканием, так называемого, сквозного тока $I_{ск}$ во время переходных процессов при переключении вентиля. В течение коротких промежутков времени, когда открыты оба транзистора, между шиной питания и общей шиной имеет место гальваническая цепь, по которой протекает сквозной ток. Этот ток обуславливает потребление энергии от источника питания со скоростью, определяемой выражением:

$$P_{ск.m} = I_{ск} \cdot E_{dd}. \quad (Д.3.9)$$

При линейном изменении входного напряжения, когда

$$U_{ex}(t) = \frac{E_{dd}}{\tau} \cdot t \quad (Д.3.10)$$

среднее значение $I_{ск}$ определяется выражением:

$$I_{ск}^{сред} = \frac{\beta}{12 \cdot E_{dd}} \cdot (E_{dd} - 2V_t)^3 \cdot \frac{\tau}{t}, \quad (Д.3.11)$$

где $\beta = \beta_n W_n = \beta_p W_p$, $V_t = V_{tn} = |V_{tp}|$ — пороговое напряжение транзисторов.

Соответственно, рассматриваемая компонента мощности будет определяться выражением:

$$\rho_{\text{ск.т}} = \frac{\beta}{2} (E_{dd} - 2V_t)^3 \frac{\tau}{t}.$$

Компонента мощности P_{ym} обусловлена двумя типами токов утечки: током утечки обратносмещенного перехода сток-подложка и подпороговыми токами через канал закрытого транзистора. Ток утечки обратносмещенного перехода определяется выражением:

$$I_{ym} = I_s \left[\exp \left(\frac{U}{m\varphi_T} \right) - 1 \right], \quad (\text{Д.3.12})$$

где I_s — ток насыщения р-п перехода, m — коэффициент неидеальности, $\varphi_T = \frac{kT}{q}$. Удельное значение тока I_s составляет величину $(1 \div 5) \cdot 10^{-12} \text{ А} \cdot \text{см}^{-2}$ (при $t = 25^\circ \text{ С}$).

Так называемые подпороговые токи I_{cu} протекают между истоком и стоком транзистора при напряжениях между затвором и стоком меньших пороговых. Величина подпороговых токов определяется выражением:

$$I_{nn} = k \left\{ \exp \left[\frac{(U_{зс} - V_t)}{n\varphi_T} \right] \right\} \left(1 - \exp \left[\frac{U_{cu}}{\varphi_T} \right] \right) \quad (\text{Д.3.13})$$

где k — эмпирический коэффициент, величина которого определяется технологией; $n = 1 + \frac{\Omega t_{cx}}{D}$, где t_{cx} — толщина подзатворного диэлектрика, D — ширина обедненной области канала, Ω — отношение диэлектрических постоянных кремния и двуокиси кремния.

Статическая компонента P_{cm} мощности отсутствует в классических КМОП схемах логических вентилях статического типа. Однако, существуют разновидности КМОП вентилях, в которых имеет место потребление энергии от источника питания даже в стационарных состояниях. К таким разновидностям относится, так называемая псевдоNMOS логика и логика на NMOS проходных транзисторах.

Из всех компонентов мощности, входящих в выражение (Д.3.7), наибольшей является динамическая компонента в случае, когда величина фактора переключательной активности α приближается к единице. Величина α зависит от типа цифрового устройства и системной организации СБИС, в которой используются КМОП вентили. Есть такие типы систем, в которых одновременно в процессе переключения находится большинство вентилях. К таким системам

относятся, например, микропроцессорные системы. Для обеих систем $\alpha \approx 1$. И, наоборот, в запоминающих устройствах одновременно в процессе переключения находится лишь небольшая ($\sim 10 \div 100$) элементов из нескольких миллионов. Для этих систем $\alpha \rightarrow 0$.

Как видно из выражения (Д.3.8), с увеличением частоты мощность увеличивается линейно, поэтому схемы без отношения теряют свое основное преимущество при достаточно высоких частотах даже по сравнению с ЭСЛ и ТТЛ (см. раздел 5.5, рис. 5.13). В цифровых схемах, которые большую часть времени находятся в ждущем режиме $\alpha \rightarrow 0$ и, следовательно, динамическая составляющая уменьшается, но при большой степени интеграции ($N = 10^6 - 10^7$) начинает превалировать компонента $P_{ум}$. Эта ситуация усугубляется при уменьшении размеров транзисторов.

Динамическая компонента имеет сильную (квадратичную) зависимость от напряжения питания. Поэтому радикального уменьшения мощности можно добиться путем уменьшения напряжения питания.

Д.4. Схемотехнические и энергетические характеристики логических схем с отношением

По определению логическими схемами с отношением считают схемы, в которых напряжение $U_{«0»}$ зависит от соотношения сопротивлений переключающего и нагрузочного элементов в открытом состоянии.

В схемах с отношением уровень логического $U_{«0»}$ формируется за счет падения напряжения, создаваемого рабочим током на сопротивлении электронного переключателя в открытом состоянии (рис. Д.4.1). При этом отношение сопротивлений переключателя $R_{отк}^{\Pi}$ к нагрузочному сопротивлению R_n должно быть возможно меньшим. Величина $U_{«0»} \rightarrow 0$ при $R_{отк}^{\Pi}/R_n \rightarrow 0$. Уровень логической единицы $U_{«1»}$ формируется за счет заряда нагрузочной емкости через сопротивление R_n или за счет падения напряжения, создаваемого входящим током нагрузки $I_{вх}$ на сопротивление нагрузки $R_{вх}$. Первый случай реализуется в вентилях, в которых используются переключатели, управляемые напряжениями. В таких переключателях входной ток $I_{вх} = 0$. К таким переключателям относят полевые транзисторы с изолированным затвором и индуцируемым каналом (MOS-транзисторы) и полевые транзисторы с управляемым переходом.

дом (р-п переходом или диодом Шоттки) ПТУП нормально открытого типа.

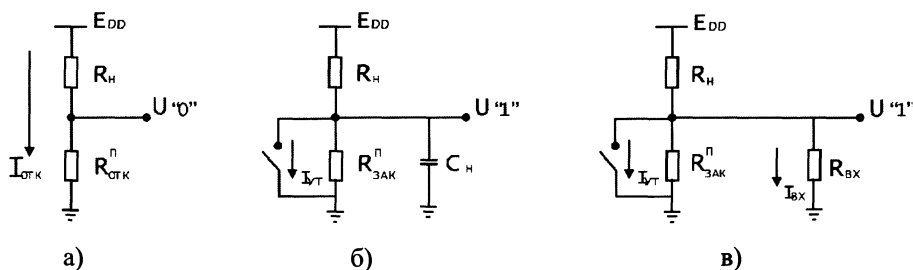


Рис. Д.4.1. Резистивно-ключевые схемы вентиля с отношением.

Второй случай реализуется в вентилях, в которых используется переключатели, управляемые током. К таким переключателям относят биполярные транзисторы, полевые транзисторы с управляющим переходом нормально закрытого типа.

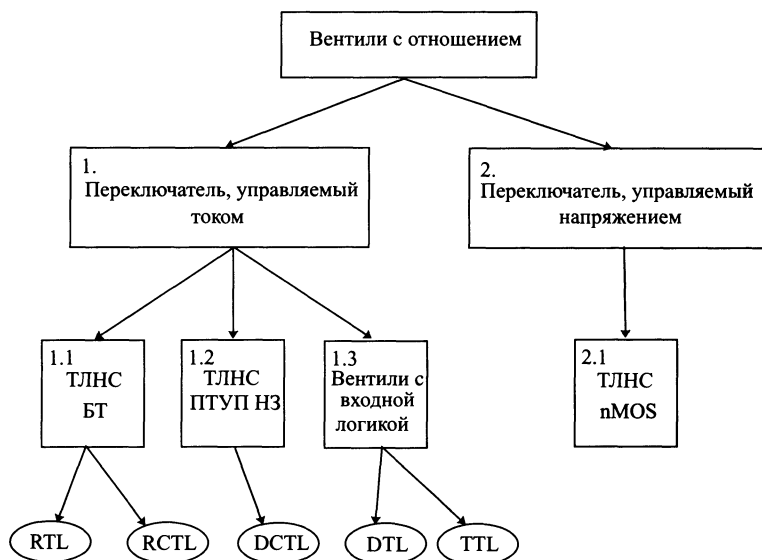


Рис. Д.4.2. Классификационная диаграмма вентилей с отношением.

Д.4.1. Классификация базовых вентилей с отношением

В качестве первого классификационного признака вентилей с отношением в соответствии со сказанным выше принимается вид переключающего элемента. По этому признаку сформированы два

класса базовых вентилях с отношением на классификационной диаграмме, приведенной на рис. Д.4.2.

В пределах первого класса можно выделить три вида логических схем, отличающихся переключательными элементами и конфигурацией схем.

Два из трех видов этих вентилях (виды 1.1 и 1.2) относятся к схемам с непосредственными связями и с источником питания одной полярности. Структурная схема базовых элементов с непосредственными связями приведена на рис. Д.4.3. Необходимым условием реализации таких связей в логических схемах является одинаковая полярность входного и выходного напряжений переключательного элемента.

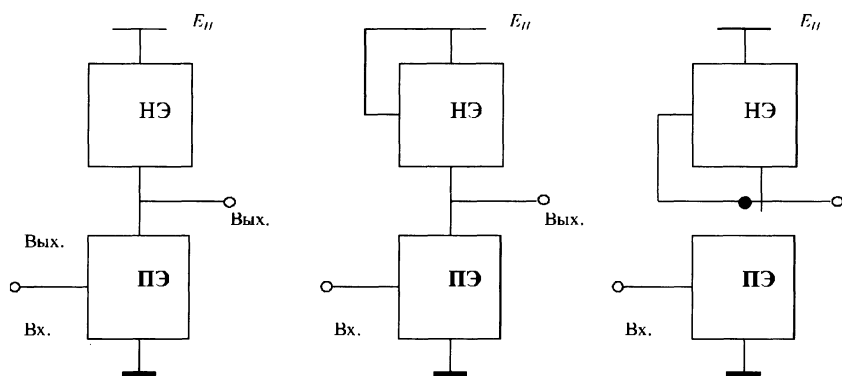


Рис. Д.4.3. Структурные схемы базовых логических элементов с непосредственными связями.

Различия в свойствах базовых элементов, построенных на различных полупроводниковых приборах нормально закрытого типа (БТ, ПТУП, МОП и др.), обуславливаются различиями в характере зависимостей их входных и выходных токов от соответствующих напряжений. Необходимым условием пригодности прибора для построения инвертора в соответствии с рис. Д.4.3 является нормально закрытый вид его выходных ВАХ. В морфологической табл. Д.4.1 сведены некоторые разновидности базовых элементов транзисторной логики с непосредственными связями. Одинаковая полярность и одинаковый порядок величин входных и выходных напряжений позволяют использовать для построения схем с непосредственными связями и одним источником питания БТ и различные виды нормально закрытых ПТ. Третий вид 1.3 относится к так называемым схемам с входной логикой, обобщенная структурная схема которых приведена на рис. Д.4.4. Так как входная цепь МОП-транзистора не

потребляет тока (управление напряжением), то использование транзистора в схемах с непосредственными связями дает дополнительные схемотехнические преимущества по сравнению с другими приборами. В частности, поэтому схемы с непосредственными связями на МДП-транзисторах с индуцированным каналом нашли наиболее широкое применение в интегральных микросхемах.

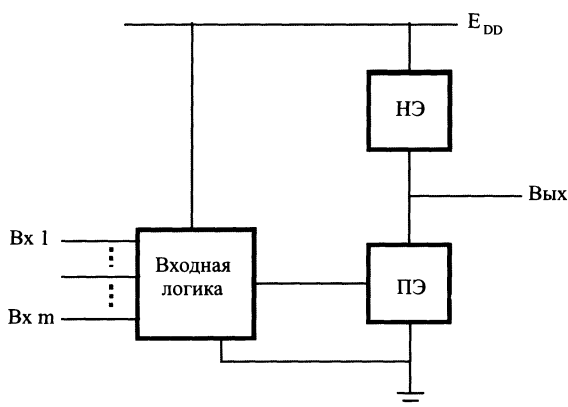


Рис. Д.4.4. Обобщенная структурная схема базового логического элемента с входной логикой.

Д.4.2. Основные схемотехнические и структурно-топологические типы

Идеальной с точки зрения схемотехнической организации является цифровая схема, в которой переключательные элементы смежных каскадов соединены непосредственно, без дополнительных элементов. Такой подход к схемотехнической организации и технологическая совместимость нагрузочного и переключательного элементов обуславливает простоту конструкции базового логического вентиля и высокую плотность компоновки.

Базовые элементы на биполярных транзисторах. Основой таких схем является инвертор, приведенный в табл. Д.4.1 (позиция 1.1). Типичное соединение базовых инверторов показано на рис. Д.4.5. В большинстве цифровых интегральных микросхемах используют параллельное соединение БТ в отличие от схем на ПТУП, ПТШ и МОП-транзисторах, в которых осуществляют также последовательное и последовательно-параллельное соединения. Параллельное соединение переключательных БТ представляет собой схему ИЛИ-НЕ.

Таблица Д.4.1.

НЭ ПЭ	1 R	2 D	3 ПТШ НЗ	4 ПТУП НЗ	5 МОП НЗ	6 МОП НО	7 ПТУП НО	8 ПТШ НО
1 БТ	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
2 ПТУП	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8
3 ПТШ	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8
4 МДП	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8

Работа схем на БТ основана на том, что остаточное напряжение между коллектором и эмиттером в режиме насыщения БТ настолько мало, что его недостаточно для открывания нагрузочных транзисторов, базы которых подсоединены к коллектору этого транзистора. Кроме того, в режиме отсечки коллекторный ток БТ практически равен нулю. Эти два свойства БТ делают возможной работу в ключевом режиме, когда насыщенный управляющий транзистор закрывает нагрузочные транзисторы и, наоборот, когда управляющий транзистор в режиме отсечки открывает нагрузочные БТ. Большое различие в значениях коллекторных токов в открытом и закрытом состояниях оказывается достаточным для создания на нагрузке необходимого перепада логических уровней. Напряжение U_0 выбирают равным остаточному напряжению транзистора в режиме насыщения $U_{кзн}$, а напряжение U_1 выбирают равным напряжению между базой и эмиттером БТ, соответствующему режиму насыщения $U_{эбн}$. В схеме, приведенной на рис. Д.4.5, значения E_{DD} и R_n выбирают из условий неискаженной передачи информации по логической цепи из последовательно соединенных инверторов. Поскольку

БТ является прибором, управляемым током, то выражения, определяющие E_{DD} и R_n , имеют вид:

$$E_{DD} - (n \cdot I_6 + I_{\kappa}^0) \cdot R_n \geq U_1,$$

$$E_{DD} - I_{\kappa}^1 \cdot R_n \leq U_0,$$

где I_6 , I_{κ}^1 — ток базы и коллектора насыщенного транзистора; I_{κ}^0 — ток коллектора в режиме отсечки.

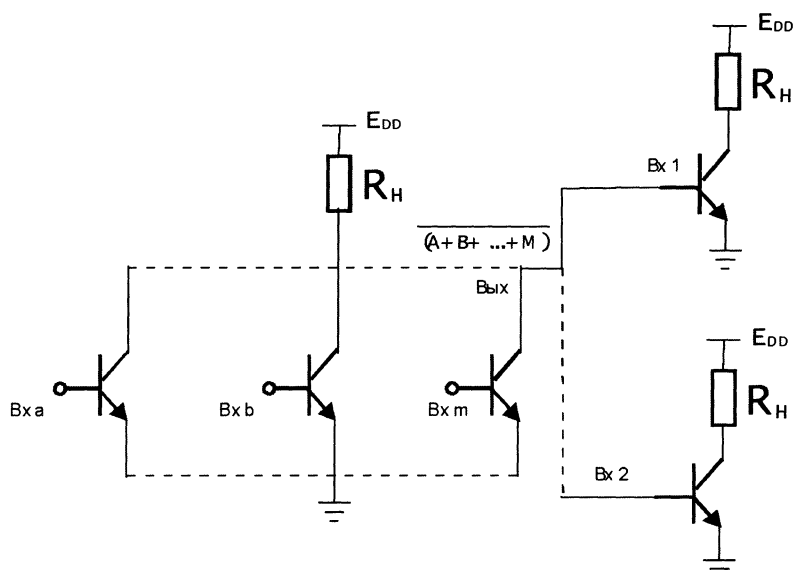


Рис. Д.4.5. Логическая схема ИЛИ-НЕ с непосредственными связями на биполярных транзисторах.

Схема ИЛИ-НЕ, показанная на рис. Д.4.5, работает следующим образом. При наличии хотя бы на одном из m входов напряжения U_1 соответствующий переключательный транзистор входит в режим насыщения и напряжение на выходе схемы уменьшается до $U_0 = U_{\text{кэн}}$. Если на всех входах устанавливается напряжение U_0 , то ток источника питания через нагрузочный резистор поступает на базы нагрузочных транзисторов, открывая их. Неравномерное распределение тока между базами транзисторов, подключенных к одному выходу инвертора, и создает главную трудность реализации данного класса схем. Входные характеристики транзисторов имеют разброс, обусловленный разбросом технологических параметров, различиями в режимах работы и неодинаковостью температур транзисторов.

С ростом температуры входная характеристика сдвигается влево со скоростью $(1-2) \text{ мВ}/^\circ\text{С}$. Разброс технологических параметров (удельного поверхностного сопротивления, толщины базы, сопротивления контакта к базе и т.д.) обуславливает сдвиг характеристик.

Базовые элементы на МОП-транзисторах с одинаковым типом проводимости каналов. Если используется в качестве нагрузочного и переключательного элементов одинаковые МОП-транзисторы с индуцированным каналом, то возможны две схемотехнические конфигурации инвертора (см. рис. Д.4.6 а), б)). Недостатками первого варианта инвертора являются: малый логический перепад и сравнительно большая площадь. Первый недостаток заключается в том, что логический перепад зависит от порогового напряжения и определяется выражением:

$$\Delta U = E_{dd} - V_t - U_{\text{«0»}}.$$

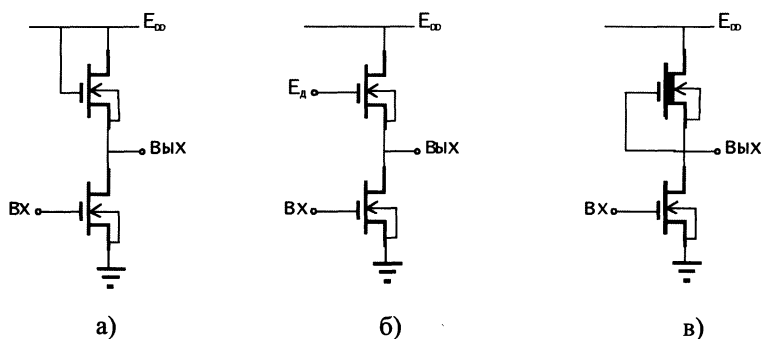


Рис. Д.4.6. Инверторы на транзисторах с каналами одинакового типа проводимости: а) – с индуцированными каналами; б) – с дополнительным источником смещения; в) – с объединенной нагрузкой;

Объясняется это тем, что нагрузочный транзистор работает в режиме истокового повторителя и закрывается при заряде нагрузочной емкости, когда напряжение между его истоком и затвором уменьшается до пороговой величины V_t . При этом напряжение на входе достигает величины $U_{\text{«1»}} = E_{DD} - V_t$. Этот недостаток устраняется при подключении затвора нагрузочного транзистора к дополнительному источнику напряжения E_d . Если $E_d \geq E_{DD} + V_t$, то $\Delta U = E_{DD}$. Однако введение дополнительного источника питания с дополнительной шиной уменьшает плотность компоновки и представляет известные неудобства. Второй недостаток вытекает из необходимости превышения сопротивления канала нагрузочного транзистора сопротивлению канала переключательного трпнзистора. Если

канал переключательного транзистора представляется возможным выполнить с минимальными топологическими размерами, то длина канала нагрузочного транзистора должна быть большей, для обеспечения малой величины U_{0*} . Рассмотренные конструкции обладают наилучшей технологичностью, однако отмеченные выше недостатки оказались настолько серьезными, что пришлось искать более совершенные схемы. На рис. Д.4.6 в) приведена схемная конфигурация, которая обеспечивает наибольшую плотность компоновки, максимальный логический перепад и хорошую технологичность. Используя в качестве нагрузочного элемента транзистор со встроенным каналом, можно при минимальной площади обеспечить условия, при которых сопротивления канала открытого переключательного транзистора будет много меньше сопротивления канала нагрузочного элемента. Данная схемотехническая конфигурация воплощается в компактную физическую структуру и топологию (см. рис. Д.4.7 а), б)). При высокой плотности компоновки данная конструкция обеспечивает максимальный логический перепад и высокое быстродействие.

Базовые вентили на полевых транзисторах с управляющим переходом. Аналогично строятся вентили на кремниевых и арсенид-галлиевых n -канальных ПТУП и ПТШ. (позиции 2.7, 2.8 и 3.7, 3.8 в морфологической табл. Д.4.1). Варианты этих вентилей, реализованные в виде структур с полной диэлектрической изоляцией (например, типа SOI), обладают минимальными паразитными емкостями, способны коммутировать большие токи и, следовательно, являются потенциально высокоскоростными. Практически максимальное быстродействие достигнуто в, так называемой, логике с непосредственными связями (DCFL), в которой используется нормально-закрытые арсенид-галлиевые ПТШ в качестве переключательных элементов и нормально-открытые — в качестве нагрузочных. В схемах DCFL могут быть использованы так же НЕМТ-транзисторы и другие, быстродействующего типа транзисторы, если они допускают нормально-закрытые и нормально-открытые варианты выполнения.

Условия неискаженной передачи информации в данных схемах имеют вид:

$$\begin{aligned} U_{вых}^0 &= E_n - (nI_3^1 + I_c^0)R_n > U_1, \\ U_{вых}^1 &= E_n - R_n(I_c^1 + nI_3^0) < U_0, \end{aligned}$$

где I_3^1, I_3^0 — токи, протекающие через затвор в открытом состоя-

нии при $U_{zu} = U_1$, $U_{cu} = U_0$ и закрытом при $U_{zu} = U_0$, $U_{cu} = U_1$ состояниях ПТ; I_c^0 , I_c^1 — токи, протекающие между стоком и истоком в открытом состоянии при $U_{zu} = U_0$, $U_{cu} = U_1$ состояниях ПТ. В наихудшем случае

$$\Delta U = R_n(I_{\text{вблх}}^1 - I_{\text{вблх}}^0 - n/B * I_{\text{бх}}^1).$$

где B — коэффициент усиления ПТ по току.

Из этого выражения следует, что для получения $\Delta U > 0$ необходимо, чтобы коэффициент разветвления по входу n был меньше коэффициента усиления по току B .

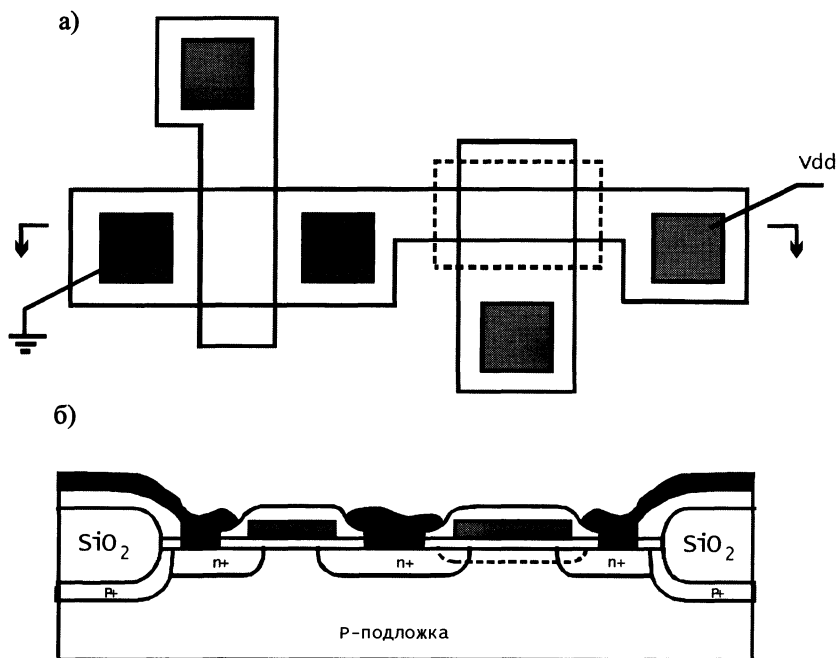


Рис. Д.4.7. Инвертор на транзисторах с каналами одинакового типа проводимости (рис. Д.4.6 в): а) топология; б) разрез физической структуры.

Токи, протекающие через ПТУП и ПТШ, зависят от напряжений U_0 и U_1 . При выборе значения U_1 необходимо учитывать зависимость коэффициента усиления по току от тока затвора, т.к. она имеет экспоненциальный вид и значение B резко изменяется при изменении U_{zu} . Особенностью передаточной характеристики базового элемента схем с непосредственными связями на ПТУП и ПТШ является зависимость $U_{\text{вблх}}^0$ от числа элементов нагрузки. Поскольку

ПТУП и ПТШ являются приборами, потребляющими ток по входной цепи, то с увеличением числа затворов ПЭ нагрузки, подключаемых к выходу закрытого инвертора, его входное напряжение уменьшается. Если в отсутствие нагрузки ($n = 0$) $U_{вых}^0 = E_n$, то при $n \neq 0$ $U_{вых}^0 < E_n$. Напряжение U_0 зависит от остаточного напряжения между стоком и истоком открытого ПТ: $U_0 = R_n I_c^1$, где R_k — сопротивление канала ПТУП и ПТШ.

Остановимся подробнее на факторах, определяющих быстродействие базовых элементов рассмотренного типа. Так же как и БТ, в нормально-закрытых ПТУП, работающих при прямых смещениях на p - n -переходе затвор-исток, происходит накопление избыточного заряда неосновных носителей в затворной, истоковой и канальной областях. Поэтому быстродействие схем с непосредственными связями на ПТУП также зависит от времени рассасывания неосновных носителей, которое и определяет в основном и время формирования положительного фронта выходного импульса при переключении инвертора при закрывающимся ПЭ. Кроме того, параметром, характеризующим длительность переходных процессов в схемах с непосредственными связями на ПТУП, является зарядка паразитной выходной емкости инвертора при формировании отрицательного фронта выходного импульса. Разрядка выходной емкости происходит через сопротивление канала открытого ПТУП и зависит в первом приближении от постоянной времени, равной $R_k C_n$. Следует отметить, что составляющая времени задержки переключения, обусловленная накоплением избыточного заряда неосновных носителей, больше других составляющих при значениях рабочих токов, соответствующих высокому уровню инжекции p - n -перехода затвор-исток ПТУП. Однако с уменьшением уровня токов влияние эффектов накопления на время задержки переключения уменьшается. При малых уровнях токов, соответствующих низкому уровню инжекции p - n -перехода затвор-исток ПТУП, их влияние незначительно по сравнению с влиянием времени зарядки-разрядки паразитных емкостей. В инверторах, использующих ПТШ в качестве ПЭ, эффекты, связанные с накоплением неосновных носителей, отсутствуют, поскольку ПТШ является униполярным прибором. Время задержки переключения инверторов такого типа зависит от времени пролета носителей заряда через канал и времени зарядки и разрядки паразитных емкостей инвертора.

Базовые логические элементы ИЛИ-НЕ и И-НЕ на ПТУП и ПТШ строятся путем их последовательного и параллельного соединения

аналогично тому, как строятся схемы на МДП-транзисторах одного типа электропроводности. На рис. Д.4.8 приведена схема комплексного логического вентиля m И— n ИЛИ-НЕ. Для того чтобы при последовательном соединении m ПТ в схеме И напряжение U_0 не превысило допустимого уровня, необходимо ширину канала ПЭ увеличить в m раз. Схема работает следующим образом. Когда на входы одной из n последовательных цепей НЭ подается напряжение U_1 , транзисторы этой цепи открыты и между шиной питания и общей шиной протекает ток, создающий падение напряжения на сопротивлении канала НЭ: $U_{\text{вых}}^1 < U_0$. При этом все ПЭ нагрузочных схем оказываются закрытыми (нагрузочные схемы на рис. Д.4.8 не показаны). Если хотя бы на одном из входов последовательно соединенных ПЭ имеет место напряжение U_0 , то соответствующий ПТ оказывается закрытым. Напряжение на выходе логического вентиля равно E_D при отсутствии нагрузки на выходе.

Напряжение на выходе схемы при наличии на входе нагрузки, например, в виде p затворов аналогичных ПЭ:

$$U_{\text{вых}} = E_{DD} - (pI_s^1 + nI_c^0)R_n,$$

где I_s' — ток затвора в открытом состоянии ПТ, I_c^0 — ток стока в закрытом состоянии, R_n — сопротивление канала открытого ПТ.

Одной из особенностей схемотехники ИМС является то, что технологическая совместимость является важнейшим критерием выбора элементов для реализации той или иной схемы инвертора из приведенных в табл. Д.4.1. Под *технологической совместимостью* понимают возможность изготовления различных схемных элементов в едином технологическом процессе. Предпочтение отдается схемам, содержащим однотипные элементы. Примерами таких схем могут служить инверторы на полевых транзисторах (позиции 2.4, 3.3, 4.5). Наилучшей технологичностью и универсальностью обладают инверторы, в которых нагрузочным и переключательным элементами являются n-МОП-транзисторы с индуцированным и встроенным каналами.

Д.4.3. Особенности обработки информации и энергопотребления

Рассмотренные в предыдущем разделе схемотехнические и структурно-топологические типы базовых логических вентилях с отношением отличаются простотой конструкции и малооперационностью технологии изготовления. Простейшие конструкции обеспечивают

высокую плотность компоновки при реализации цифровых устройств. В этом заключается их бесспорное преимущество по сравнению со схемами без отношения.

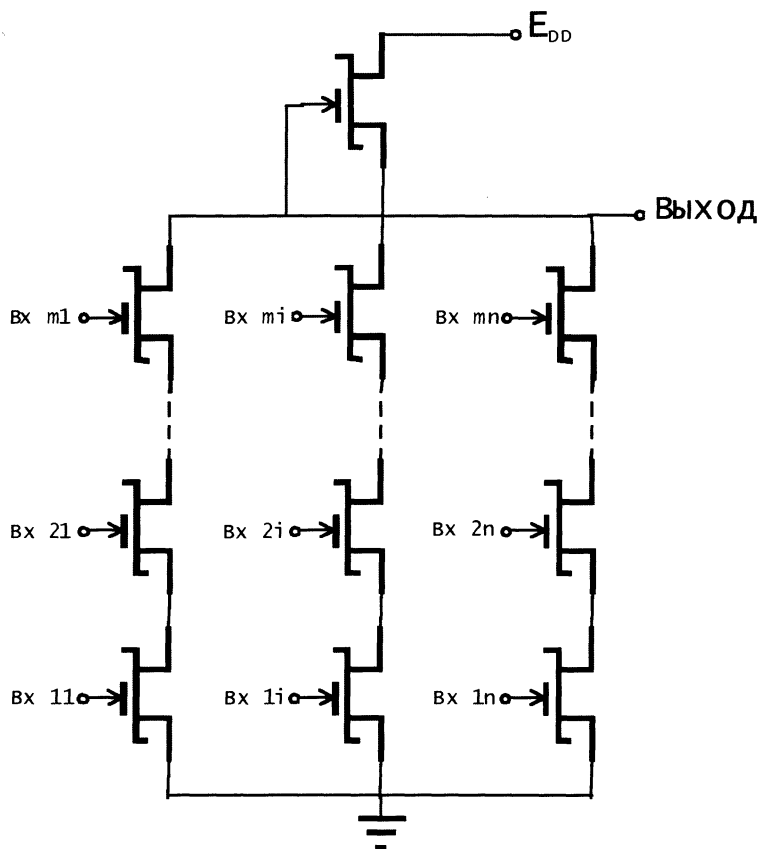


Рис. Д.4.8. Логика с непосредственными связями на полевых транзисторах Шоттки. (DCFL)

Другим важным достоинством является высокое быстродействие и, соответственно, высокая скорость производства информации. Это достоинство так же является следствием простоты конструкции, которое обеспечивает минимальные значения паразитных емкостей транзисторных структур и соединений, и высокие значения коммутируемых ими токов. Сочетание малых значений емкостей и больших токов, заряжающих и разряжающих эти емкости в процессе производства информации, позволяет достичь рекордно малых значений задержек переключения. Малые значения паразитных емкостей достигаются за счет использования полной диэлектрической

изоляции базовых вентилях. Высокие значения токов при сравнительно малых площадях рабочих областей планарных структур и сравнительно небольших напряжениях питания обуславливает униполярный (электронный) механизм проводимости элементарных переключателей в открытом состоянии. В данном типе базовых логических вентилях носителями являются электроны, в отличие от КМОП-элементов, где ток обусловлен как электронами так и дырками, подвижность которых в несколько раз ниже чем у электронов.

Д.4.4. Особенности энергопотребления

Особенности энергопотребления логических схем с отношением, вытекающие из способа производства информации, рассмотрим на примере вентиля типа DCFL. Данный тип базовых элементов является типичным представителем схем с непосредственными связями на приборах, управляемых током.

Общее выражение для скорости потребления энергии от источника питания имеет вид:

$$P = P_{ct}^{cp} + P_d + P_{ym}. \quad (Д.4.1)$$

Среднее значение статической компоненты мощности P_{ct}^{cp} определяется выражением:

$$P_{ct}^{cp} = \eta P_{c0} + (1 - \eta) P_{c1} \quad (Д.4.2)$$

где η — часть времени, в течении которого выход вентиля находится в состоянии «0»; P_{c0} — мощность в режиме открытого переключательного элемента; P_{c1} — мощность в режиме закрытого переключательного элемента;

Когда переключательный элемент инвертора или, по крайней мере, один из переключательных элементов в схеме «ИЛИ-НЕ» DCFL-типа открыт, имеет место гальваническая связь между шиной питания и общей шиной, по которой протекает ток. Таким образом, для сохранения логического состояния необходимо потребление энергии от источника питания.

Когда все переключательные элементы в схеме «ИЛИ-НЕ» DCFL-типа закрыты, прямая гальваническая цепь между шиной питания и общей шиной отсутствует, но имеет место цепь, соединяющая шину питания с общей шиной через диод Шоттки нагрузки, подключенный к выходу. По этой цепи протекает ток, формирующий уровень логической «1». Таким образом для сохранения данного логического состояния необходим ток и, следовательно, расходование энергии



от источника питания. Величина тока определяется напряжением питания, сопротивлениями каналов нагрузочного и переключательного транзисторов и вольт-амперной характеристикой диода Шоттки. С точки зрения минимизации количества потребляемой энергии, напряжение питания должно быть возможно меньшим, а сопротивление канала нагрузочного элемента возможно большим. Однако, при этом быстродействие вентиля, определяемое временем заряда паразитной нагрузочной емкости C_n , будет минимальным.

Поэтому упомянутые параметры выбираются на основе компромисса между скоростью производства информации и энергопотреблением.

Мгновенное значение динамической компоненты $P_d(t)$ в выражении (Д.4.1) определяется выражением:

$$P_d(t) = \frac{dE}{dt} = i(t) \cdot V_{dd}, \quad (\text{Д.4.3})$$

где dE и i — мгновенные значения энергии и тока, протекающего от источника питания в переходном режиме при смене логического состояния на выходе вентиля. Мгновенное значение тока определяются выражением:

$$i(t) = C_n \cdot \frac{dV_{out}}{dt}, \quad (\text{Д.4.4})$$

На основании (Д.4.3) и (Д.4.4) для энергии, потребляемой от источника питания при заряде нагрузочной емкости, можно записать:

$$\begin{aligned} E_{0 \rightarrow 1} &= \int_0^T p(t) dt = V_{dd} \int_0^T i(t) dt = V_{dd} \int_{V_0}^{V_1} C_n dV_{out} = \\ &= V_{dd} \cdot C_n \cdot (V_1 - V_0). \end{aligned} \quad (\text{Д.4.5})$$

В отличие от КМОП уровень логической «1» $V_1 \neq V_{dd}$, а уровень логического «0» $V_0 \neq 0$. Практически для DCFL вентиля $\Delta U \leq 0,7 \text{ В}$. Динамическая компонента ищется с учетом сказанного выше и определяется соотношением:

$$P_d = \alpha_{0 \rightarrow 1} \cdot V_{dd} \cdot C_n \cdot (V_1 - V_0) \cdot f, \quad (\text{Д.4.6})$$

где $\alpha_{0 \rightarrow 1}$ — фактор переключательной активности.

И, наконец, третья компонента мощности P_{yT} определяется формулой:

$$P_{ym} = V_{dd} \cdot I_{ym}, \quad (\text{Д.4.7})$$

где $I_{ут}$ — ток обусловленный утечкой через закрытый переключа-
тельный транзистор.

Основной особенностью энергопотребления логических вентилях с отношением является постоянное потребление энергии, необходи-
мой как для производства информации, так и для ее сохранения в
виде уровней напряжения логических «0» и «1». Для вентилях с от-
ношением, построенных на приборах, управляемых напряжением,
процесс производства новой информации во время переходных про-
цессов при переключении и поддержке информации в стационарных
режимах во многом отличаются. Формирование напряжения $U_{«1»}$ на
выходе вентиля в них происходит путем заряда нагрузочной емко-
сти C_n , подключенной к выходу, током, протекающим от источника
тока через нагрузочный элемент, за время τ_{01} , определяемое выраже-
нием:

$$\tau_{01} = 2 \cdot R_n \cdot C_n, \quad (Д.4.8)$$

где R_n — сопротивление нагрузочного элемента.

Для поддержания уровня $U_{«1»}$, не требуется протекания тока и,
следовательно, в данном стационарном состоянии ток не протека-
ет (за исключением тока утечки нагрузочной емкости) и источник
питания не рассеивает энергии.

При уменьшении норм проектирования и, соответственно, раз-
меров рабочих областей логических вентилях, уменьшаются вели-
чины паразитных емкостей и уменьшается энергия единичного пе-
реключения ($P \times \tau$). Поэтому схема типа DCFL может оказаться
более эффективной (например, при больших частотах), чем КМОП-
логика.

Д.5. Схемотехника логических вентилях на токовых ключах

Принцип действия вентилях данного типа заключается в переключе-
нии тока генератора постоянного тока между двумя цепями, име-
ющими общий узел, подключенный к генератору тока. В зависимости
от состояния входа ток неизменной величины, поставляемый гене-
ратором тока, может протекать в общую шину по одной из двух
возможных цепей, формируя уровни логического нуля и логической
единицы.

При этом уровень логического нуля $U_{«0»}$ может формироваться
двумя способами.

Первый способ заключается в том, что уровень $U_{«0»}$ формируется на выходном электроде, подключенном к цепи, соединенной с общей шиной, когда ток в цепи отсутствует. Этот способ реализуется в эмиттерно-связанной логике или в истоково-связанной логике, называемой также логикой на переключателях тока. Уровень логической единицы $U_{«1»}$ в подобных вентилях формируется за счет падения напряжения на нагрузочном резисторе в цепи, соединенной с общей шиной, когда ток генератора тока переключен в эту цепь. При этом в другой цепи ток отсутствует.

Второй способ формирования уровней $U_{«0»}$ и $U_{«1»}$, реализованный в вентилях с инжекционным питанием, также заключается в переключении тока генератора постоянного тока из одной цепи в другую. Упомянутые цепи имеют два общих узла. Одним общим узлом является общая шина («земля»), другим — узел подключения генератора неизменного тока, который является входом вентиля.

Д.5.1. Общий принцип схемотехнической организации и классификация

Организация логических схем на токовых ключах базируется на использовании генератора тока, величина которого неизменна для всех режимов работы схем. Ток генератора тока, оставаясь неизменным по величине, переключается, по крайней мере, одним переключателем в две разные цепи. По какой цепи протекает ток, определяется входным сигналом, управляющим переключателем.

Представление логической «1» при этом осуществляется за счет создания падения напряжения на линейных или нелинейных элементах схемы одной цепи. Уровень логического «0» определяется протеканием этого тока по другой цепи.

Этот принцип может быть реализован в двух типах схем, приведенных на рис. Д.5.1 и рис. Д.5.2. Схема первого типа представляет собой дифференциальный усилитель и содержит два переключателя П1 и П2, которые управляются двумя разными входными сигналами. При соответствующем входном сигнале, когда П1 находится в проводящем состоянии, ток генератора тока 1 протекает по цепи 1 и создает на резисторе падение напряжения и на выходе имеет место напряжение $U_{«1»}$. При этом П2 разомкнут и ток в цепи 2 отсутствует и, следовательно, напряжение на выходе 2 равно напряжению общей шины. При противоположных входах П1 — закрыт, П2 — открыт, напряжения на выходах имеют противоположные значения.

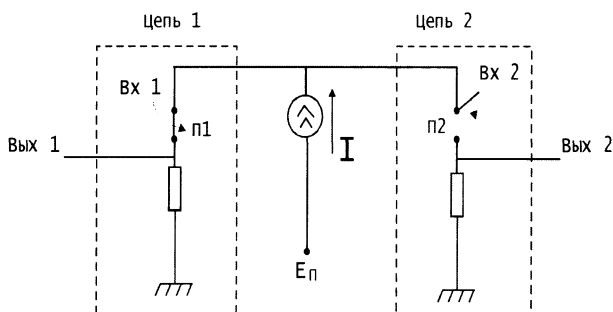


Рис. Д.5.1. Эквивалентная схема токового ключа на основе дифференциального усилителя.

В схемах второго типа ток генератора I переключается переключателем П1, управляемым входным сигналом Вх1. При открывающем входном сигнале ток I , протекая по резистору, создает на выходе 1 напряжение $U_{«0»}$. При разомкнутом П1 ток I втекает в цепь 2, а именно, во входную цепь переключателя П2, и открывает П2. Если входная цепь переключателя П2 представляет собой диод, то на вых1 будет формироваться напряжение $U_{«1»}$ в соответствии с видом вольт-амперной характеристикой и величиной тока I .

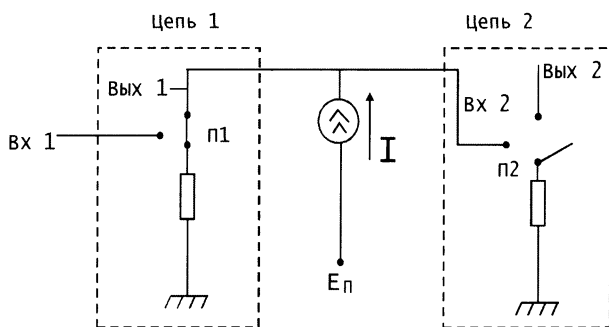


Рис. Д.5.2. Эквивалентная схема токового ключа с инжекционным питанием.

В соответствии со способами формирования логических уровней, описанными выше, логические схемы на токовых ключах можно сгруппировать в два класса (см. рис. Д.5.3). Вентили первого класса строятся на основе схемы дифференциального усилителя, вентили второго класса известны под названием схем с инжекционным питанием. Обобщенные структурные схемы этих классов приведены на рис. Д.5.4 и рис. Д.5.5.

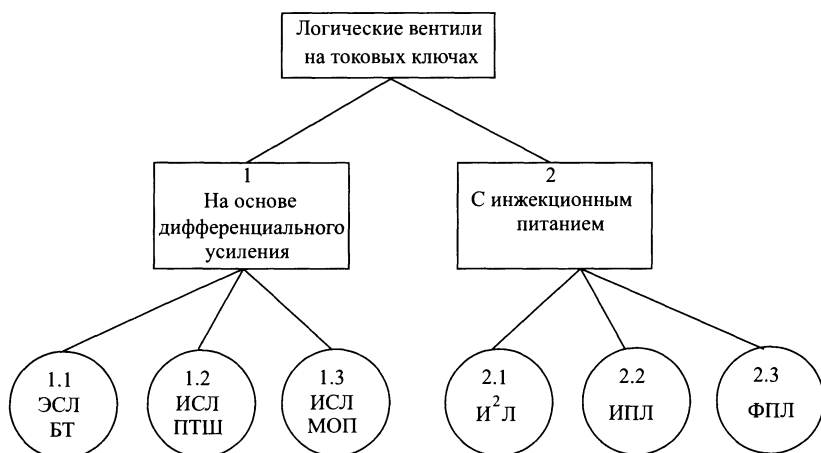


Рис. Д.5.3. Классификация вентилях на токовых ключах.

В первом классе логических вентилях наиболее широко распространенным типом являются логические схемы эмиттерно-связанной логики (ЭСЛ), описанные в разделе 5.3. В этот же класс входят схемотехнические решения логических схем стоково-связанной логики. Отличия последней от ЭСЛ заключается в использовании в качестве ПЭ1 и ПЭ2 полевых транзисторов.

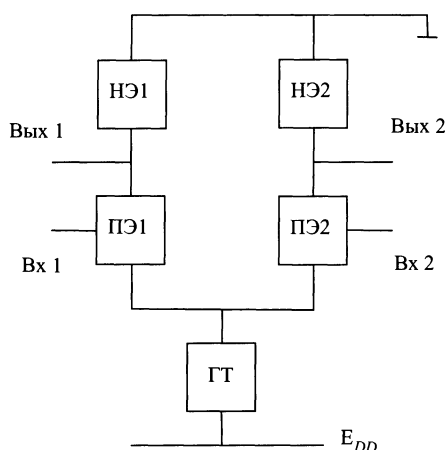


Рис. Д.5.4. Обобщенная структурная схема токового ключа на основе дифференциального усилителя.

В схемах ЭСЛ (см. позиции 1.2 и 1.3 на классификационной диаграмме рис. Д.5.3) могут быть использованы или полевые транзисторы с затвором Шоттки или МОП-транзисторы.

Д.5.2. Интегральная инжекционная логика

В разделе 5.4 приведены электрическая схема и разрез физической структуры одного из возможных вариантов И²Л. В настоящее время известно множество схемотехнических и структурно-топологических видов вентиляй данного класса, обладающих важными для СБИС энергетическими особенностями.

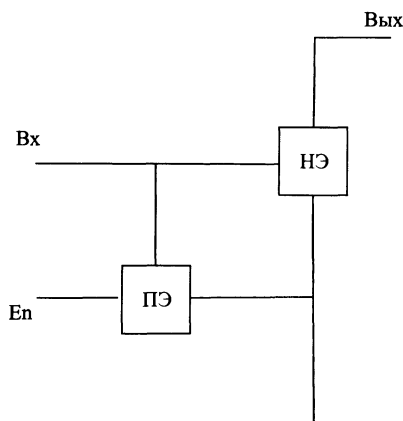


Рис. Д.5.5. Обобщенная структурная схема токового ключа с инжекционным питанием.

На рис. Д.5.6 приведена морфологическая таблица известных схем вентилей с инжекционным питанием.

В позиции 1.1 таблицы расположена электрическая схема классического варианта И²Л инвертора. Эта электрическая схема может быть воплощена в интегральную структуру, отличающуюся высокой плотностью компоновки (в 3–4 раза превышающую плотность компоновки КМОП вентиля), низкой энергией переключения и сравнительно простой технологией изготовления. Остановимся более подробно на особенностях конструкции инжекционной логики, чтобы вскрыть заложенное в физической структуре основное противоречие, не позволяющее повысить энергию переключения и быстродействие, и уменьшение потребляемой мощности без ухудшения остальных параметров, главным из которых является технологичность.

Сравнительно низкое быстродействие классической инжекционной логики является ее принципиальным недостатком, неизбежно вытекающим из физической структуры. Дело в том, что использование инверсного включения переключательного транзистора с вертикальной структурой обуславливает значительное накопление неосновных носителей заряда в эмиттерной области, в отличие от

нормального включения, при котором неосновные носители заряда накапливаются преимущественно в коллекторе, по сравнению с чем накопление носителей в эмиттере ничтожно мало.

НЭ \ ПЭ	1 БТ р-п-р	2 ПТУП р-канал	3 ПТШ р-канал
1 БТ п-р-п	1.1 	1.2 	1.3
2 ПТУП п-канал	2.1 	2.2 	2.3
3 ПТШ п-канал	3.1 	3.2 	3.3

Рис. Д.5.6. Морфологическая таблица вентилях с инжекционным питанием.

Поэтому ограничить накопление носителей заряда в структуре и, следовательно, повысить быстродействие и уменьшить энергию переключения так, как это делается, например, в транзисторно-транзисторной логике (см. раздел 5.2), не представляется возможным. Использование известных средств повышения быстродействия транзисторно-транзисторной логики — легирование структуры золотом и шунтирование коллекторного перехода диодом Шоттки — в классической инжекционной логике невозможно по следующим причинам. Уменьшение времени жизни неосновных носителей заряда посредством легирования золотом структуры инжекционной логики, состоящей из совмещенной пары биполярных транзисторов дополняющего типа проводимости, приводит к уменьшению и без того малого коэффициента усиления горизонтального нагрузочного транзистора. Шунтирование же коллекторного перехода диодом Шоттки уменьшает накопление избыточного заряда носителей в структуре на ничтожно малую величину.

Таким образом, основное техническое противоречие классической инжекционной структуры заложено в конструкции переключательного элемента — инверсно включенного биполярного транзистора. С одной стороны, для увеличения быстродействия желательно ограничить ток через эмиттерный переход и связанное с ним накопление неосновных носителей заряда, но при этом ухудшаются нагрузочные свойства переключательного элемента, и в конечном счете нарушается его работоспособность, так как необходимым условием работы инжекционного ключа является условие насыщения переключательного транзистора.

Устранение рассмотренного выше противоречия является необходимым условием решения задачи о повышении быстродействия, уменьшения энергии переключения инжекционной логики без усложнения технологии изготовления, увеличения рассеиваемой мощности и уменьшения плотности упаковки.

Одним из возможных путей устранения основного противоречия является отказ от биполярной транзисторной структуры переключательного элемента и использования в этой роли другого полупроводникового прибора, технологически совместимого с биполярным транзистором цепи питания.

Однако этот путь повышения быстродействия предполагает отказ от одного из наиболее значительных достижений инжекционной логики-простоты технологии.

Вторым серьезным недостатком классического варианта инжекционной логики является многоколлекторная организация выхода логических элементов. Для исключения эффекта разброса входных характеристик нагрузочных транзисторов выход инжекционного логического элемента выполняется в виде нескольких коллекторов, количество которых определяется количеством входов нагрузки. При этом каждый коллектор соединяется металлизированным проводником только с одним входом. Такая организация связей между логическими элементами значительно усложняет металлизацию внутрисхемных соединений и затрудняет выполнение интегральной схемы с однослойной металлизацией. В данном случае имеется противоречие между плотностью упаковки и нагрузочной способностью логического элемента. Повысить плотность компоновки инжекционного логического вентиля можно за счет сокращения количества коллекторов, но при этом уменьшается нагрузочная способность логического элемента. Данное противоречие обусловлено особенностями работы переключательного элемента, а именно свойствами входной характеристики биполярного транзистора.

Как известно, физическая структура инжекционной логики достаточно проста, но требует использования сравнительно высокоомной эпитаксиальной пленки. Это требование диктуется инверсным включением биполярного транзистора. Поэтому дальнейшее упрощение структуры, в частности использование безэпитаксиальной монокристаллической подложки, не представляется возможным. В данном случае имеет место противоречие между технологичностью и усилительными свойствами переключающего элемента.

Подводя итог сказанному выше, можно констатировать, что основные противоречия, не позволяющие совершенствовать классическую инжекционную логику в части уменьшения энергии переключения, повышения плотности компоновки и степени интеграции, тесно связанной с технологичностью структуры, содержатся в принципе действия и конструкции переключающего элемента — инверсно включенного биполярного транзистора.

Отсюда следует вывод: радикального улучшения параметров классического варианта инжекционной логики можно добиться только отказавшись от биполярной транзисторной структуры, использующейся в качестве переключающего элемента.

В позициях 2.1 морфологической таблицы (рис. Д.5.6) размещена электрическая схема инжекционно-полевой логики (ИПЛ), в которой в качестве переключающего элемента использован транзистор с управляющим переходом (ПТУП).

Структура ИПЛ элемента представлена на рис. Д.5.7. Базовый элемент содержит нормально закрытый ПТУП, затвор которого является входом, а сток — выходом инвертора. В структуре, показанной на рис. Д.5.7 генератор тока выполнен в виде биполярного транзистора.

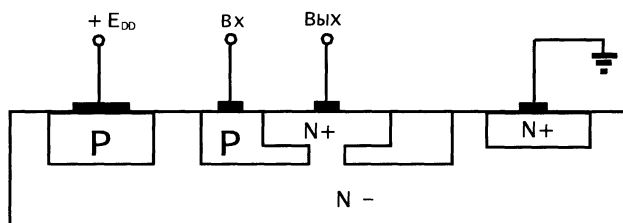


Рис. Д.5.7. Структура ИПЛ-элемента.

Важной особенностью физической структуры элемента ИПЛ является возможность формирования рабочих областей в монокристаллической подложке без эпитаксиального слоя. Такая конструктив-

ная особенность обеспечивает ряд преимуществ этого элемента по сравнению с известными биполярными функционально — интегрированными элементами — технологичность, высокий процент выхода годных. Наличие диэлектрического маскирующего участка, под которым формируется канал полевого транзистора с управляющим p - n -переходом приводит к возможности создания переключательного транзистора, работающего на полевом эффекте. Выходы элемента ИПЛ могут быть выполнены в виде нескольких независимых стоковых областей аналогично И²Л вентилю.

С точки зрения простоты технологии изготовления, оптимальной является структура переключательного элемента в виде конструкции с вертикальным каналом, а генератора тока в виде горизонтального биполярного транзистора. Такая конструкция обеспечивает функциональную интеграцию и технологическую совместимость биполярного и полевого транзисторов.

Элемент ИПЛ работает следующим образом. Переключение базы и эмиттера p - n - p -транзистора соответственно к «земле» и «плюсу» цепи питания и подачи на последний напряжения, равного напряжению открывания эмиттерного p - n -перехода ($0,3 \pm 0,6$) В, обеспечивает протекание тока питания между эмиттером и коллектором p - n - p -транзистора. P - n - p -транзистор включен по схеме с общей базой и является генератором постоянного тока, величина которого практически не зависит от напряжения на коллекторе. Ток питания представляет собой ток неосновных носителей заряда, перемещающихся под действием дрейфа и диффузии в базовой области p - n - p -транзистора, являющейся одновременно областью истока полевого транзистора. Неосновные носители заряда коллектируются p - n -переходом затвор-исток. В зависимости от потенциала, приложенного к входному электроду, возможны две ситуации. При подаче на затвор потенциала логического нуля U^0 ($0 \div 0,1$) В, ток питания через цепь источника сигнала замыкается на «землю». В данном режиме канал НЗПТУП перекрыт областью пространственного заряда (ОПЗ) p - n -перехода затвор-исток. При этом между входом элемента и «землей» имеет место высокое сопротивление, препятствующее протеканию тока нагрузки. Отсутствие гальванической связи между входом и «землей» приводит к накоплению неосновных носителей заряда в области затвора, которые наводят потенциал, примерно равный напряжению источника питания ($0,3 \div 0,6$) В, соответствующий напряжению логической единицы U^1 . Когда p - n -переход затвор-исток смещается в прямом направлении, сопроти-

вление канала резко уменьшается вследствие уменьшения слоя объемного заряда в канале. Переключательный НЗПТУП открывается. Как можно видеть из качественного рассмотрения основных режимов работы, НЗПТУП в элементе ИПЛ работает в режиме, не характерном для работы полевых транзисторов в традиционных схемах. Напряжение между затвором и истоком, при котором происходит перекрывание канала ОПЗ, обычно имеет отрицательную полярность и составляет несколько вольт. Для элементов ИПЛ это напряжение имеет положительную полярность и составляет несколько долей вольта.

Дальнейшее увеличение плотности компоновки ИПЛ элементов достигается следующим образом.

Расщепление затвора, изменение пропорций канала и совмещение затворов дают семейство модификации инжекционно-полевого элемента, иллюстрируемого рис. Д.5.8. Далее каждая из конструкций рассматривается подробно.

Целью модификаций исходной структуры рис. Д.5.8 а) являлось повышение плотности упаковки интегральных логических элементов. Эта цель достигнута посредством выполнения ПЭ в виде ПТ с несколькими затворами, совмещенными с коллекторными областями БТ, и подключенными к входным электродам, и стоковой областью, расположенной между затворными областями и подключенной к выходному электроду. Схематическое изображение структуры логического элемента приведено на рис. Д.5.8 б). Принцип действия ФИЭ второго уровня интеграции поясняется рис. Д.5.9.

При выборе удельного сопротивления подложки I и расстояния между областями 4' и 4 таким образом, чтобы толщина ОПЗ перехода затвора при подаче на вход логического «0» была бы больше расстояния между областями 4' и 4, логический элемент будет выполнять логическую функцию И-НЕ. То есть предложенное техническое решение позволяет значительно расширить функциональные возможности ИПЛ. На одном полупроводниковом кристалле можно реализовать как вентиль ИЛИ-НЕ, так и вентиль И-НЕ только путем изменения расстояния между затворными элементами.

Рассмотренная выше конструкция может быть реализована в БИС простейшей архитектуры на безэпитаксиальной подложке. Однако, этим не исчерпывается все многообразие возможностей использования функционально-интегрированных элементов (ФИЭ) рассматриваемого типа. Например, архитектура СБИС с полной диэлектрической изоляцией, использующая диэлектрическую подлож-

ку, открывает широкие возможности для реализации новых конструктивных решений ФИЭ, обладающих высокой плотностью компоновки при малой энергии переключения и высоким быстродействием.

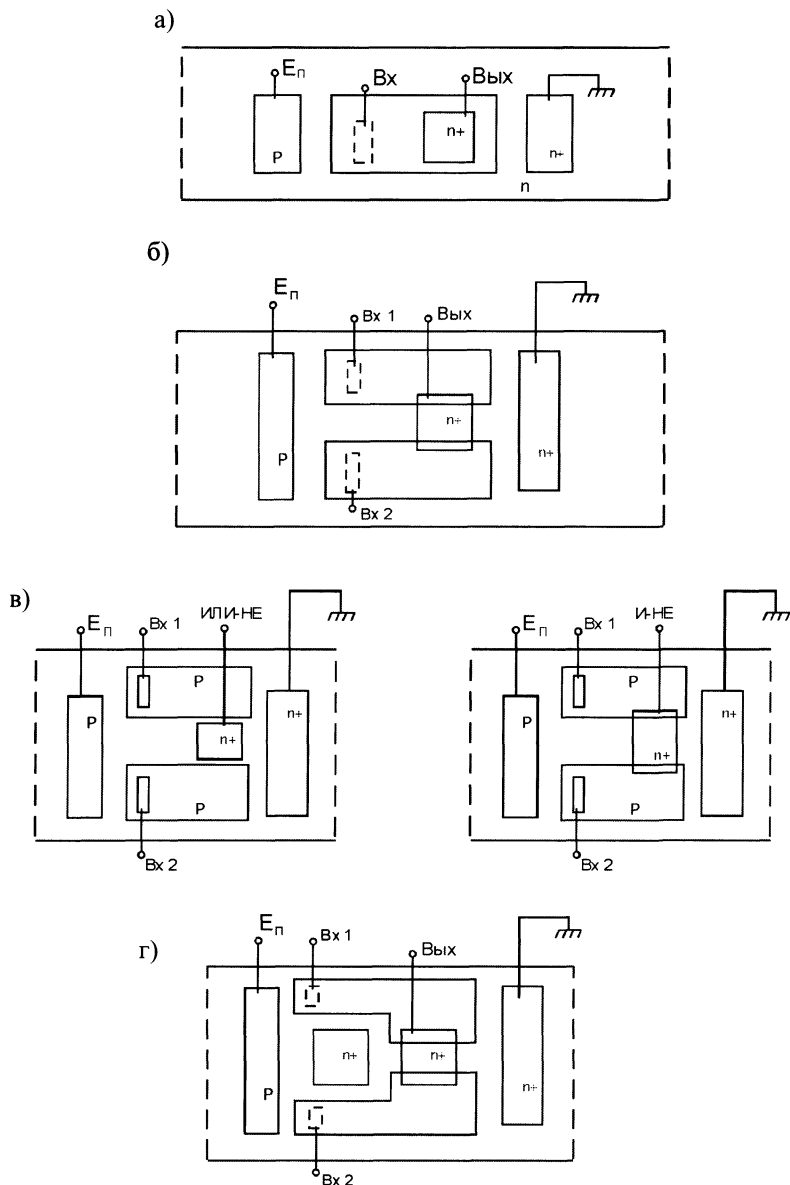


Рис. Д.5.8. Модификации ФИЭ второго уровня интеграции.

В качестве иллюстрации сказанному проанализируем модификацию конструкции базового элемента ИПЛ и преимущества, выте-

кающие из использования для его реализации диэлектрической изоляции.

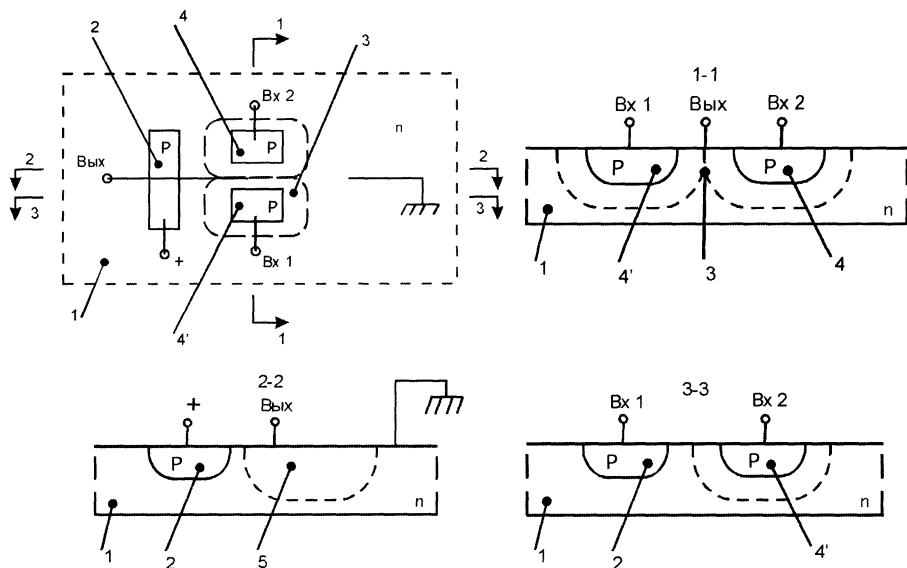


Рис. Д.5.9. Структурная схема двухвходового логического вентиля ИЛИ-НЕ: 1 — исток; 2 — инжектор; 3 — сток; 4, 4' — затворы; 5 — проекция обедненной области.

На рис. Д.5.10 приведено одно из возможных конструктивных решений ИПЛ элемента — двухвходового логического элемента ИЛИ-НЕ.

Здесь области истока и стока подключены соответственно к общей шине и выходу логического элемента. Входной сигнал подается на затворы элемента, расположенные между стоком и истоком на расстоянии, не превышающем удвоенной ширины ОПЗ p - n -перехода затвор-исток в равновесном состоянии (границы области объемного заряда показаны пунктирной линией).

Существенным достоинством этой конструкции является малая площадь p - n -переходов при сравнительно больших топологических размерах.

Энергия единичного переключения определяется выражением:

$$p \cdot \tau = \frac{\Delta U (C_{zu} + C_{zc})}{\alpha I_n}, \quad (\text{Д.5.1})$$

где ΔU — логический перепад; C_{zu} , C_{zc} — барьерные емкости p - n -переходов затвор-исток и затвор-сток; α — коэффициент передачи тока инжектора; I_n — ток источника питания.

Если допустить, что удельные значения барьерных емкостей р-п переходов затвор-исток $C_{зи}^{yd}$ и затвор-сток $C_{зс}^{yd}$ равны, то

$$(p \cdot \tau)_{nm} = \frac{\Delta U \cdot C_{зи}^{yd}}{\alpha I_n} \cdot 2S_{зс}. \quad (\text{Д.5.2})$$

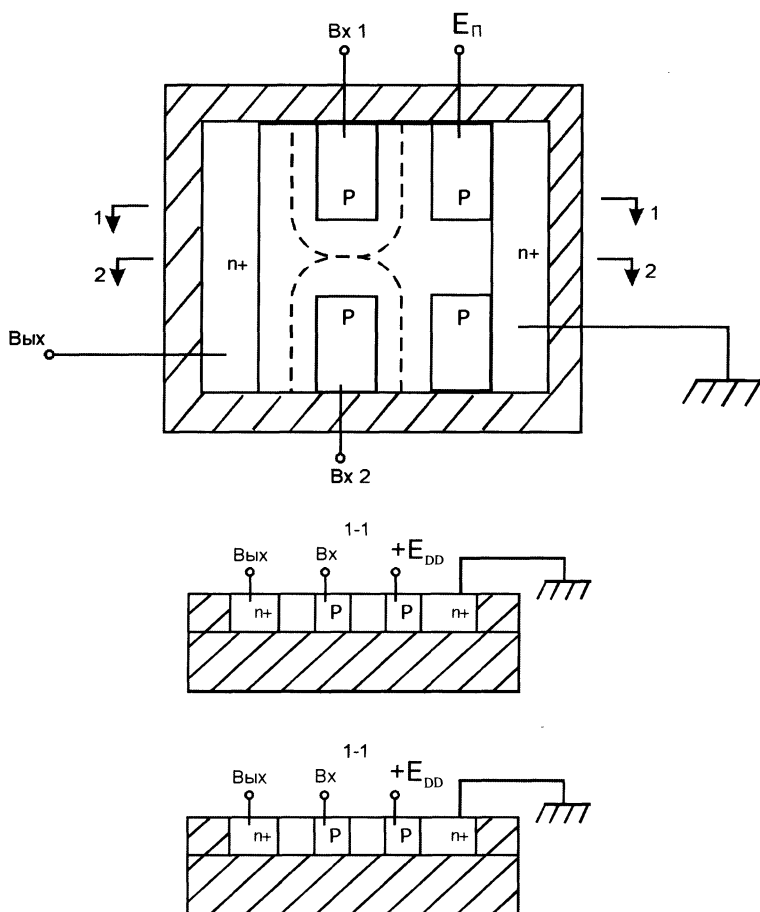


Рис. Д.5.10. ФИЭ второго уровня интеграции на диэлектрической подложке.

Нагрузочная способность ИПЛ элемента определяется выражением:

$$n = \frac{U_{«0»}}{R_{\kappa} j_{n_{zu}} S_{зи}} \left(\exp \frac{U_{«1»}}{\varphi_T} - 1 \right), \quad (\text{Д.5.3})$$

где: $U_{«0»}$, $U_{«1»}$ — напряжения логического «0» и логической «1», R_{κ} — сопротивление канала полевого транзистора в открытом состоянии,

$j_{нэс}$ — удельное значение начального тока перехода затвор-сток, ϕ_m — температурный потенциал.

Как видно из формул, основные характеристики ИПЛ-элемента в значительной мере определяются величинами площадей p - n -переходов. Поэтому является весьма актуальным применение средств и методов, обеспечивающих уменьшение размеров приборов. Кроме чисто количественного выигрыша в параметрах приборов данная конструкция позволяет реализовать новые качественные возможности. Например, появляется возможность обеспечить полную структурно-технологическую совместимость ИПЛ схем с КМОП схемами. Это делает целесообразным построение высокоэффективных комбинированных СБИС с интерфейсными схемами на КМОП элементах и внутренними схемами на ИПЛ элементах. Такое сочетание различного рода элементов целесообразно с точки зрения повышения плотности компоновки, уменьшения рассеиваемой мощности и повышения быстродействия.

Использование структур с диэлектрической изоляцией дает возможность реализовать симметричную, с взаимозаменяемыми истоком и стоком, полностью изолированную структуру полевого транзистора. Это обстоятельство обеспечивает использование всего арсенала схемотехнических цифровых средств для построения СБИС.

Недостатком ИПЛ элемента является сравнительно низкое быстродействие, обусловленное зарядом и разрядом диффузионной и барьерной емкостей p - n -переходов малыми токами (порядка 10^{-9} А).

Дальнейшего повышения быстродействия интегральных логических ИПЛ типа можно достичь посредством выполнения затворов переключательного элемента и коллекторов биполярного нагрузочного транзистора в виде неинжектирующих выпрямляющих контактов. Предложенное конструктивное решение иллюстрируется рис. Д.5.11.

В этих конструкциях затворные области полевого переключательного элемента выполнены в виде неинжектирующих выпрямляющих контактов, способных только коллектировать неосновные носители заряда, инжектированные эмиттерами горизонтального нагрузочного БТ.

Д.5.3. Быстродействие и энергетика токовых ключей.

В вентилях типа переключателей тока от источника питания потребляется постоянный ток I , неизменный для всех стационарных состояний и переходных процессов. Поэтому скорость потребления энергии в них является величиной постоянной.

В вентилях с инжекционным питанием уровень $U_{«0»}$ определяется падением напряжения, создаваемым током источника питания на сопротивлении коллектора биполярного транзистора и определяется выражением:

$$U_{«0»} = \alpha I_s \cdot \left\{ \exp \left[\frac{E_{DD}}{m\varphi_T} \right] - 1 \right\} \cdot R_k, \quad (\text{Д.5.4})$$

где I_s — начальный ток перехода инжектор–база p – n – p -транзистора, R_k — сопротивление коллектора n – p – n -транзистора, α — коэффициент передачи тока p – n – p -транзистора.

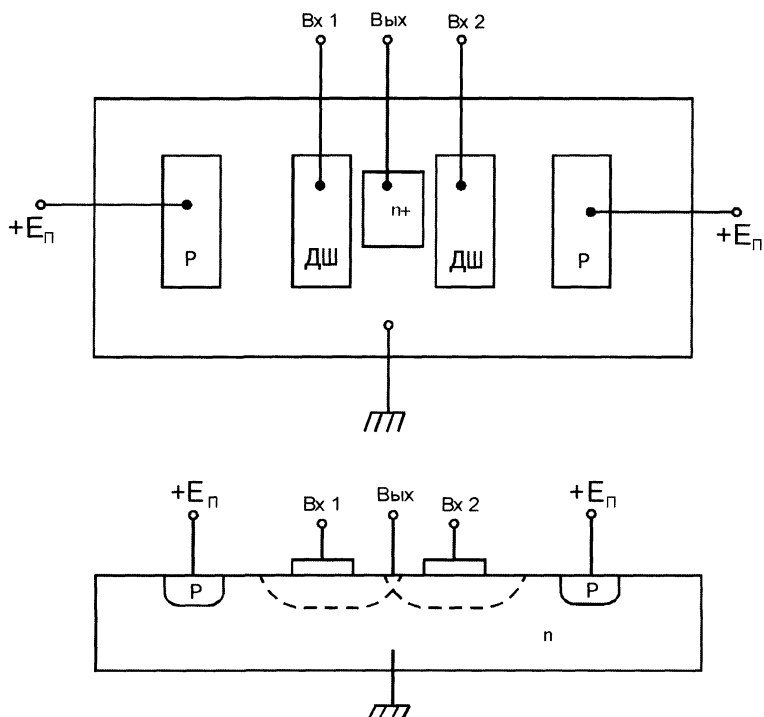


Рис. Д.5.11. Двухвходовые ИПЛ-вентили с затвором Шоттки.

Уровень $U_{«0»}$ в стационарном состоянии поддерживается открытым состоянием переключателя n – p – n -транзистора предыдущего каскада, которое обеспечивается потреблением тока питания, втекающего в его базу.

Следовательно, для предоставления информации в данном стационарном режиме требуется потребление энергии от источника со

скоростью, определяемой выражением:

$$P_{cm}^0 = E_{DD} \cdot I_s \left\{ \exp \left[\frac{E_{DD}}{m\varphi_T} \right] \right\}. \quad (\text{Д.5.5})$$

Нетрудно показать, что для поддержания уровня $U_{\text{«1»}}$ и представления информации в другом стационарном состоянии требуется такая же мощность $P_{cm}^1 = P_{cm}^0$. В обоих стационарных состояниях энергия, поставляемая источником питания, полностью термализуется в вентилях на паразитных резистивностях (входном сопротивлении, сопротивлении коллектора n - p - n -транзистора в открытом состоянии, соединениях и т.п.)

Во время переходных процессов при открывании и закрывании вентиля с инжекционным питанием, когда и происходит собственно производство информации, ток и напряжение питания остаются постоянными. Поэтому остается постоянной и мощность, то есть

$$P_{cm}^1 = P_{cm}^0 = P_{\text{дин}}^1 = P_{\text{дин}}^0 = E_n \cdot I_n.$$

В вентилях ЭСЛ типа от источника питания потребляется неизменный ток I . В зависимости от входного сигнала этот ток в стационарных состояниях протекает через один из транзисторов дифференциальной пары и создает падение напряжения на резисторе, подключенном к коллектору этого транзистора. Таким образом формируется и поддерживается напряжение $U_{\text{«1»}}$ на одном из выходов ЭСЛ вентиля. При этом от источника питания потребляется энергия со скоростью:

$$P_{cm}^1 = E_n \cdot I.$$

В другом стационарном состоянии при смене входного воздействия на противоположный постоянный ток I протекает через другой транзистор дифференциальной пары, а в коллекторной цепи первого транзистора ток отсутствует и на выходе, подключенном к его коллектору, имеет место напряжение $U_{\text{«0»}}$. Но от источника питания продолжает потребляться ток той же самой величины и, следовательно, потребляется такое же количество энергии. И в этом стационарном состоянии энергия потребляется с той же скоростью $P_{cm}^1 = P_{cm}^0$.

Во время переходных процессов при производстве информации токи через транзисторы дифференциальной пары в сумме продолжают составлять постоянную величину и, следовательно, скорость потребления энергии от источника питания остается постоянной.

Для ЭСЛ вентиля вся энергия, поставляемая источником питания, полностью рассеивается на нагрузочных резисторах и паразитных резистивных схемах.

Таким образом, для вентилях типа переключателей тока является характерным при низких частотах независимость мощности от частоты переключения, то есть от скорости производства информации.

Абсолютные значения мощности определяются величинами напряжения тока питания. С точки зрения экономии энергии напряжение питания должно быть возможно меньшей величины. Но уменьшение величины напряжения питания влечет за собой уменьшение величины тока, что в свою очередь увеличивает время зарядки паразитных емкостей и, в конечном счете, увеличение времен задержек переключения вентиля и уменьшение предельной рабочей частоты. Таким образом, экономия энергии входит в противоречие со скоростью производства информации в цифровых вентилях данного типа. Кроме этого имеются и другие факторы, ограничивающие уменьшение величины мощности.

Например, в вентилях типа И²Л при уменьшении напряжения питания всего на 60 мВ ток питания уменьшается в десять раз. Предоставляются таким образом широкие возможности варьирования мощности и затрат энергии при производстве информации. Однако, уменьшение тока коллектора биполярного $n-p-n$ -транзистора, который используется в качестве переключателя в И²Л вентиле, вызывает, начиная с некоторого достаточно малого тока, падение коэффициента усиления по току. Этот эффект влечет за собой уменьшение нагрузочной способности вентиля и, в конечном счете, нарушение его работоспособности. Тем не менее, вентили И²Л-типа способны функционировать при напряжениях питания (0,7–0,4) В и в широком диапазоне токов (10^{-3} – 10^{-7}) А. В этом заключается их замечательные энергетические особенности.

С точки зрения энергетики ЭСЛ вентили являются самыми затратными потому, что они предназначены для максимальной скорости производства информации. С этой целью величина напряжения питания, параметры компонентов схемы (транзисторов и резисторов) выбираются таким образом, чтобы обеспечить максимальное значение токов при отсутствии насыщения транзисторов-переключателей. Для достижения максимального быстродействия используется и дорогостоящие технологические методы минимизации величин паразитных емкостей. При высоких значениях токов и малых значениях паразитных емкостей достигаются малые зна-

чения задержек переключения (десятки пикосекунд) и максимально высокие частоты при производстве информации. Большие энергетические (порядка мВт/вентиль) в этом типе вентиля является платой за высокое быстродействие. Сложная технология, сравнительно малая плотность компоновки также могут быть отнесены к затратам, обеспечивающим быстродействие любой ценой. Принципиальным недостатком такого подхода к достижению экстремально высокой скорости производства информации является возникающее противоречие между энергетикой и степенью интеграции. Поскольку вся энергия, потребляемая ЭСЛ вентилями, термолизуется в кристалле интегрального цифрового устройства, а предельно допустимая величина мощности, отводимой $P_{\text{доп}}$ от кристалла, составляет величину (30–50) Вт в зависимости от способа охлаждения, то допустимая степень интеграции

$$N = \frac{P_{\text{доп}}}{P_{\text{вент}}}$$

будет весьма невысокой. Таким образом, принятый при создании ЭСЛ схем подход оказался не перспективным с точки зрения универсальной элементной базы для создания цифровых систем в виде СБИС и тем более ультра-БИС.

Примером противоположного подхода является изложенная выше энергетика вентилях И²Л типа. Этот подход кратко можно сформулировать так — минимизация потребляемой энергии любой ценой, даже ценой снижения скорости производства информации. Этот подход также имеет право на жизнь. Вентили И²Л типа могут быть эффективно использованы, например, в «системах на кристалле» с автономным питанием.

Этими двумя экстремальными точками зрения энергетики и скорости производства информации подходами в создании базовых вентилях цифровых устройств не ограничиваются все их возможные варианты. Имеется множество компромиссных решений, занимающих промежуточное положение между И²Л и ЭСЛ. К ним прежде всего необходимо отнести КМОП-логику с ее многочисленными модификациями.

Д.6. Принципы организации частично диссипативных схем

При производстве и хранении информации посредством электронных логических схем, описанных выше, вся потребляемая от источ-

ников питания мощность рассеивается в окружающей среде. Поскольку информация представляется в виде напряжений и такое представление в схемах известных конфигураций связано с протеканием токов, то превращение электрической энергии в тепло неизбежно при каждом акте производства информации. При увеличении степени интеграции и увеличении рабочей частоты такое построение цифровых схем становится неприемлемым из-за обострившейся проблемы теплоотвода. Уменьшение рассеяния энергии актуально также и с точки зрения увеличения времени работы автономных вычислительных и коммуникационных средств, работающих от источников электропитания ограниченной емкости.

Поэтому появились и развиваются новые подходы к организации электропитания электронных схем. Эти подходы принципиально отличаются тем, что имеют целью не простую экономию расходуемой энергии, а ее многократное использование при производстве информации. Многократность использования энергии предусматривает ее возврат обратно в источник питания после производства информации. Известные подходы к решению проблемы энергетики производства информации базируются на принципе адиабатического переключения, который рассматривается ниже. Далее этот подход иллюстрируется несколькими конкретными схемотехническими воплощениями.

Д.6.1. Принцип адиабатического переключения

Адиабатическое переключение или бездиссипативное (вернее, квази-бездиссипативное) переключение включает две стадии.

На первой стадии в процессе формирования $U_{«1»}$, осуществляется заряд емкости током источника питания через резистор (сопротивление канала открытого МОП транзистора) без Джоулевых потерь энергии. На второй — в процессе формирования $U_{«0»}$ — возврат энергии, накопленной на конденсаторе, обратно в источник.

На второй стадии адиабатического переключения или квазибездиссипативного переключения энергия, накопленная на конденсаторе, должна быть возвращена в источник питания. Возврат или реверс энергии может быть осуществлен при следующих необходимых условиях.

Во-первых, переключательный транзистор (n-канальный МОП транзистор) должен быть закрыт для того чтобы исключить протекание тока на землю и термализацию энергии. Во-вторых, источник питания должен быть способен не только отдавать энергию, но и принимать возвращаемую энергию для повторного использования.

Если эти условия выполняются, то оказывается практически возможным реализация логических схем, обладающих способностью квазибездиссипативного производства информации.

Для пояснения принципа адиабатического переключения обычно используется простейшая модель цепи заряда нагрузочной емкости КМОП вентиля, приведенной на рис. Д.6.1. Для реализации адиабатического заряда, то есть заряда емкости через резистор без рассеяния Джоулева тепла, эта цепь должна иметь не постоянный источник питания, а источник с напряжением, зависит от времени. Напряжение источника питания должно быть функцией времени, например, линейной.

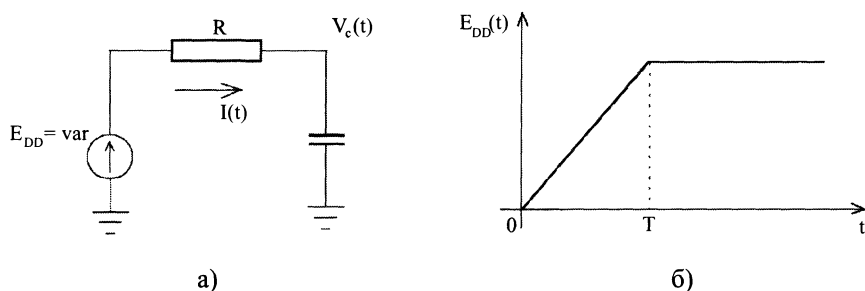


Рис. Д.6.1. Модель цепи заряда частично диссипирующего ключа: а) эквивалентная схема; б) временная зависимость напряжения источника питания.

Если в начальный момент времени $t = 0$ нагрузочная емкость разряжена, то напряжение на ней в различные моменты времени будет определяться выражением:

$$V_c(t) = \frac{1}{C} \int_0^t I(\Theta) d\Theta \simeq \frac{1}{C} \bar{I}(t) \cdot t, \quad (\text{Д.6.1})$$

где $\bar{I}(t)$ — среднее значение тока в интервале времени от 0 до t .

Выражение для среднего значения тока можно записать в виде:

$$\bar{I}(t) = \frac{CV_c(t)}{t}.$$

Тогда энергия, рассеиваемая на сопротивлении R за время от 0 до T будет определяться формулой:

$$W_{\text{дисс.}} = R \int_0^T I^2(\Theta) d\Theta \simeq 2 \cdot R \bar{I}^2(T) T = \frac{RC}{T} \cdot C \cdot E_{DD}^2. \quad (\text{Д.6.2})$$

Из данного выражения видно, что при медленном нарастании напряжения питания $E_{DD}(t)$ по сравнению с собственной постоянной времени RC цепи заряда емкости, когда

$$T \gg RC, \quad (\text{Д.6.3})$$

значение диссипирующей энергии стремится к нулю:

$$W_{\text{diss.}} \rightarrow 0$$

На основании записанного выше делается фундаментальный вывод: диссипацию энергии можно асимптотически свести к практически пренебрежимо малой величине.

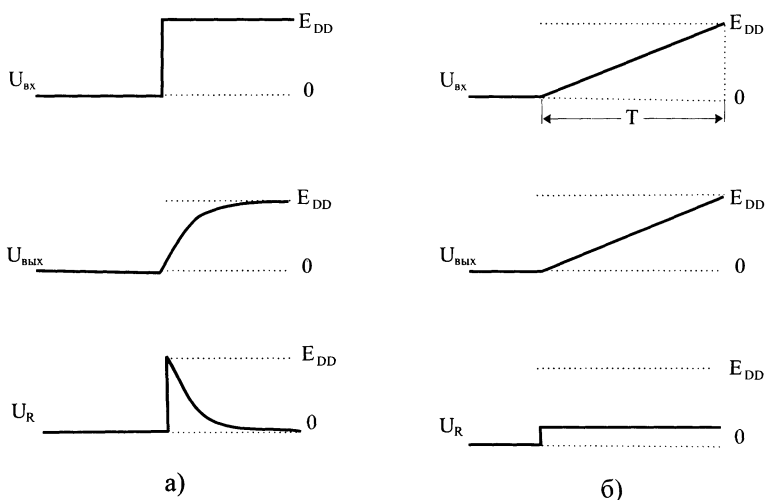


Рис. Д.6.2. Иллюстрация энергетических потерь при заряде конденсатора от импульсного источника питания: а) с импульсами с малой длительностью фронта; б) с большой длительностью фронта ($T \gg RC$).

На рис. Д.6.2 приведены временные эпюры входного, выходного напряжений и падения напряжения на сопротивлении открытого ключа, иллюстрирующие адиабатический заряд емкости при формировании $U_{«1»}$.

Рассмотрим подробнее какой ценой можно достичь этого.

Во-первых, необходимо использовать источник с напряжением, зависящим от времени. Использование таких источников сопряжено с определенными трудностями.

Во-вторых, время нарастания напряжения питания должно быть больше постоянной RC цепи заряда. Увеличивая время нарастания

напряжения заведомо ухудшает скорость производства информации. В этом противоречии между энергетическими затратами и скоростью производства информации и заключается принципиальный недостаток рассматриваемого подхода. В рамках данного метода предложено множество конкретных типов цифровых схем с частично (не полной) диссипацией энергии, которые рассматриваются ниже.

Д.6.2. Статические CMOS вентили с коллапсирующим импульсным питанием.

Энергетическую реверсивность традиционного статического CMOS-вентилля можно обеспечить путем создания условий для возвращения энергии в источник питания. Для этого необходимо использовать импульсный источник питания, создающий условия для поочередного протекания тока в как в направлении нагрузочной емкости, так и в направлении источника питания.

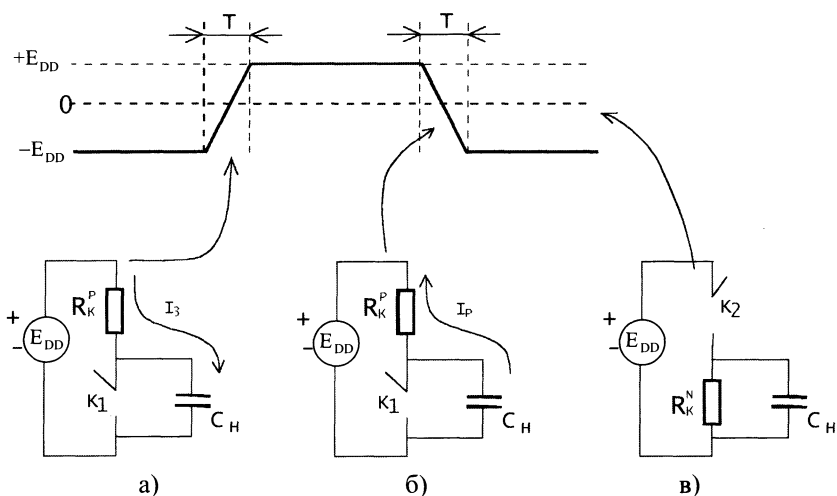


Рис. Д.6.3. Эквивалентные схемы квазибездиссипативного КМОП ключа: а) режим производства информации; б) режим реверса энергии; в) режим смены логического состояния входа.

На рис. Д.6.3 приведены резистивно-ключевые модели CMOS-вентилля, поясняющие его работу от источника энергопитания с управляемым напряжением. В реверсном режиме вентиль работает следующим образом. При входном напряжении, равном нулю, открывается р-канальный MOS-транзистор и при медленном нарастании напряжения питания током I_3 происходит заряд нагрузочной емкости в течении промежутка времени $0,5T \gg R_K^p \cdot C_H$. При транс-

порте энергии от источника питания при этом джоулевы потери на R_k^p пренебрежимо малы. При постоянном напряжении, равном $+E_{dd}$ информация в виде уровня $U_{«1»} = E_{DD}$ сохраняется на входе схемы рис. Д.6.3 а). При спаде напряжения питания рис. Д.6.3 б) нагрузочная емкость разряжается током I_p , протекающим к источнику питания. Энергия возвращается в источник питания. При равенстве нулю напряжения на конденсаторе состояние выхода инвертора может быть изменено на противоположное: р-канальный МОП транзистор закрывается (K2 — разомкнут), а п-канальный канальный МОП транзистор открывается (K1 — замкнут). Если состояние выхода инвертора изменяется в момент времени $0,5T$, когда $E_{DD} = 0$, то диссипация энергии не происходит, поскольку отсутствуют сквозные токи через транзисторы. Это условие является основным для бездиссипативной смены логического состояния выхода. После смены состояния выхода происходит заряд нагрузочной емкости до напряжения $U_{«0»}$.

Рассмотрим условие бездиссипативного производства информации в логической цепи. Для этой цели используем цепочку последовательно соединенных инверторов КМОП типа (рис. Д.6.4). Для выполнения условия бездиссипативной смены логических состояний схемы необходимо обеспечить следующее. Во-первых, необходимо для каждого i -го инвертора использовать собственный управляемый источник питания E_{DD} . Во-вторых, импульсы питающих напряжений должны быть коллапсирующими. То есть, импульс каждого i -го вентиля должен быть как бы вложенным в импульс $(i-1)$ -ого вентиля (см. рис. Д.6.4 б)). Эти условия обеспечивают передачу информации без диссипации энергии. Действительно, в режиме, когда происходит формирование уровня $U_{«1»}$ на выходе $(i-1)$ -ого инвертора, и, следовательно, изменяется логическое состояние на входе i -ого инвертора, напряжения питания i -ого инвертора равно нулю и сквозные токи отсутствуют. Напряжение питания i -ого инвертора начинает плавно увеличиваться только спустя промежуток времени t_0 , достаточный для завершения всех переходных процессов. Таким образом, в режиме производства информации (этот режим называют режимом вычислений) на выходах четных инверторов устанавливаются уровни $U_{«1»}$, на выходах нечетных — уровни $U_{«0»}$. Результат — уровень напряжения на выходе n -ого инвертора может быть запомнен, например, с помощью триггерной схемы. После этого осуществляется режим стирания информации. В этом режиме энергия, затраченная на производство информации, возвращается в источник питания. Реверс энергии производится в обратном поряд-

ке. Вначале возвращается в источник питания E_{DD} энергия, накопленная в нагрузочной емкости n -ого инвертора. При этом состояние его входа не изменяется. Это обеспечивается коллапсирующими импульсами питания. В последнюю очередь энергия извлекается из нагрузочной емкости первого инвертора при уменьшении напряжения питания E_{DD} . На этом цикл заканчивается и состояние входа первого инвертора может принимать другое значение.

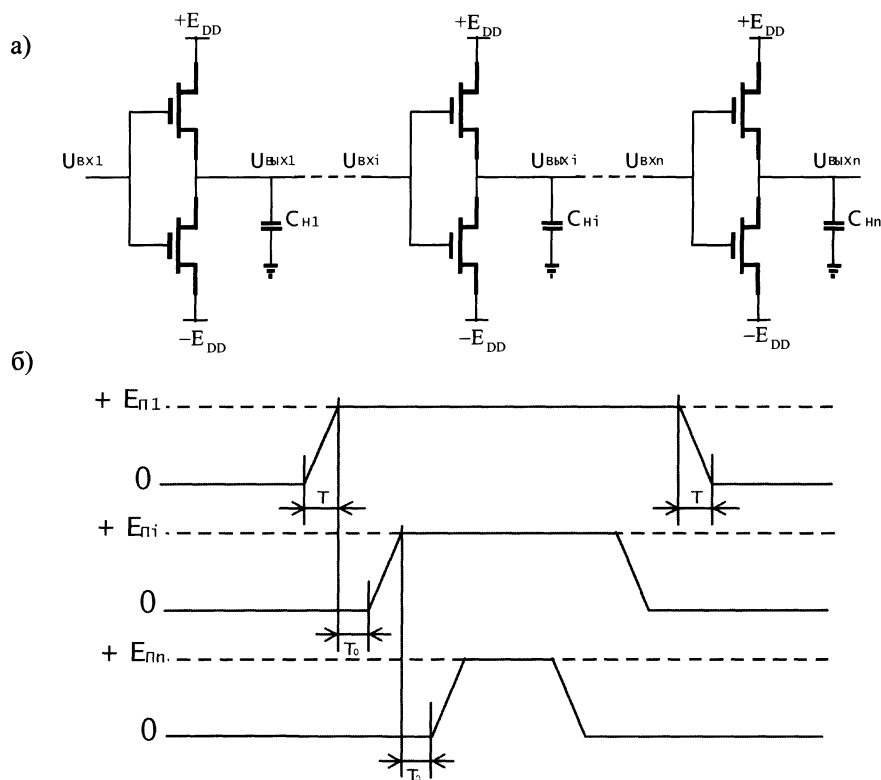


Рис. Д.6.4. а) логическая цепь инверторов; б) коллапсирующие импульсы питания.

Для иллюстрации практического применения изложенного выше принципа рассмотрим организацию комбинационных логических субблоков с реверсом энергии. Для обеспечения возможности реверса энергии логический субблок должен быть организован в соответствии со следующим правилом. Субблок любой сложности должен быть топологически упорядочен. Проще всего это сделать расположив входящие в него элементарные логические вентили в виде матрицы $(m \times n)$, приведенной на рис. Д.6.5. При этом каждый

i -ый ($i = 1, 2, \dots, n$) столбец матрицы формируется из логических вентилях, входы которых соединены с выходами вентилях предыдущих столбцов ($1, 2, \dots, n$) столбцов. Все вентили одного столбца подключаются к одной i -ой шине питания. Входы вентилях первого столбца подключаются ко входам логического субблока, а выходы вентилях последнего столбца — ко входам дополнительных элементов памяти. Формирование строк матрицы может быть осуществлено произвольно. Количество строк определяется количеством вентилях в первом столбце. Элементы матрицы, образующие последний столбец ($n + 1$), расположены только в строках, где расположены вентили, выходы которых подключены к выходам субблока.

В упорядоченном таким образом субблоке выполняются все условия адиабатического переключения, если для питания каждого столбца используется индивидуальный источник питания, а все импульсы являются коллапсирующими (см. рис. Д.6.5).

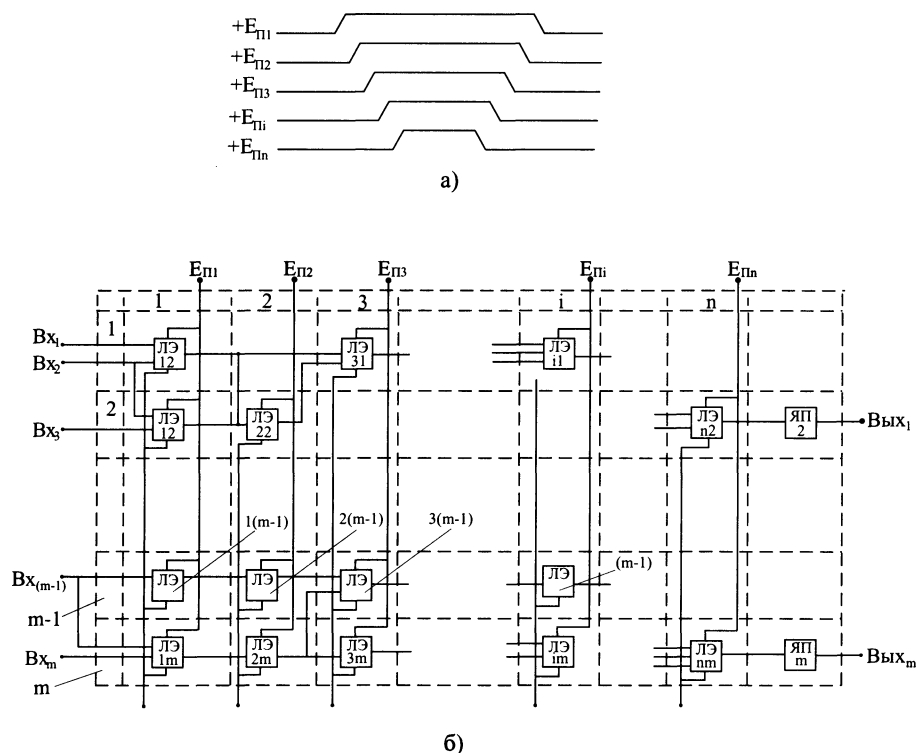


Рис. Д.6.5. Организация логических субблоков с реверсом энергии: а) коллапсирующие импульсы; б) матрица субблоков.

Д.6.3. Вентили со ступенчатой перезарядкой выходной емкости

Идея данного варианта асимптотически бездиссипативного производства информации основана на ступенчатой зарядке нагрузочной выходной емкости посредством поочередного подключения ее к нескольким (N) промежуточным источникам постоянного напряжения. Напряжения промежуточных источников ступенчато увеличивается от 0 до E_n с шагом E_{DD}/N .

Разрядка нагрузочной емкости производится в источники также ступенчато, путем ее поочередного подключения к тем же промежуточным источникам, но в обратном порядке. Преимущества способа вытекают из того, что на каждой ступени от источника энергопитания потребляется малое количество энергии W_c , а в результате удастся уменьшить диссипацию энергии в целом. Как было показано выше, ключом к достижению адиабатического переключения является уменьшение приращения напряжения при заряде емкости. В идеальном случае для достижения минимальной диссипации для данного времени заряда можно достичь при постоянном токе заряда. Это требует создания источников энергопитания с линейно изменяющимся напряжением питания. Создание таких источников на одном кристалле с логическими схемами представляет определенные трудности, поскольку в их используются индуктивности с высокой добротностью. Ступенчатая зарядка является, по существу, альтернативным способом генерации аппроксимативно линейного напряжения. его достоинство заключается в том, что этот способ не требует использования индуктивностей.

На рис Д.6.6 приведена электрическая схема, иллюстрирующая идею способа. Существует способ выходной нагрузочной емкости C_H . Последовательное подключение источников постоянного напряжения $E_{n1} = \frac{E_n}{N}$, $E_{n2} = \frac{2E_n}{N}$, $E_{nN} = E_n$ в соответствующей электрической схеме может быть осуществлено посредством электронных ключей $1, 2, \dots, N$.

В качестве электронных ключей могут быть использованы как р-канальные, так и n-канальные MOS транзисторы. Схема работает следующим образом. Вначале открывается первый транзистор, все остальные транзисторы при этом закрыты и нагрузочная емкость заряжается до напряжения E_{n1} . Через интервал времени Δt , превышающий удвоенную величину постоянной времени цепи заряда, закрывается первый транзистор и открывается второй транзистор. Остальные транзисторы остаются закрытыми. Нагрузочная емкость зарядится еще на $\frac{E_n}{N}$ и напряжение на нее достигнет вели-

чины $\frac{2E_n}{N}$. В результате N последовательных подключений источников $E_{n1}, E_{n2}, \dots, E_N$ за время $\Delta t \cdot N$ напряжение на нагрузочной емкости достигнет $E_{\text{п}}$.

Для каждой ступени зарядки нагрузочной емкости диссипация энергии будет определяться формулой:

$$W_{cm} = q\bar{U} = C_n \cdot \frac{E_n}{N} \cdot \frac{E_n}{2N} = \frac{1}{2} C_n \cdot \frac{E_n^2}{N^2}. \quad (\text{Д.6.4})$$

Полная энергия W^3 , диссипирующая в процессе зарядки нагрузочной емкости за N ступеней, будет определяться выражением:

$$W^3 = N \cdot W_{cm} = \frac{1}{2} C_n \cdot \frac{E_n^2}{N}. \quad (\text{Д.6.5})$$

Сравнение выражения (Д.6.5) с выражением (Д.6.4) показывает, что диссипация при пошаговой зарядке меньше для обычного CMOS вентиль в N раз.

Как видно из выражения (Д.6.5) при $N \rightarrow \infty$ $W^3 \rightarrow 0$, т.е. в электрической схеме на рис. Д.6.6 достигается асимптотически бездиссипативное производство информации, конкретно, формирование уровня $U_{\text{«1»}}$.

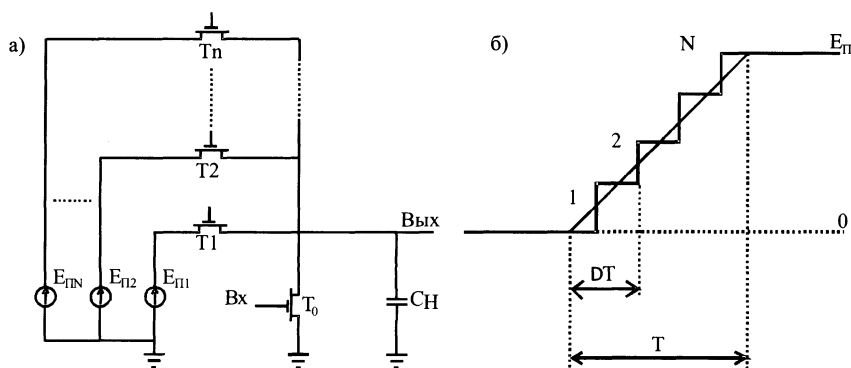


Рис. Д.6.6. Способ ступенчатой зарядки выходной емкости: а) электрическая схема; б) эпюра выходного напряжения.

При формировании уровня $U_{\text{«0»}}$ производится ступенчатая разрядка нагрузочной емкости. Открывается $(N - 1)$ транзистор и к нагрузочной емкости подключается источник питания с напряжением $\frac{(N-1)}{N} \cdot E_n$, которое в данный момент времени меньше напряжения на ней. Поэтому часть заряда, накопленного на емкости, равная:

$$\Delta q = C_n \cdot \frac{1}{N} \cdot E_n, \quad (\text{Д.6.6})$$

возвращается в источник, следовательно, возвращается в источник и соответствующая часть энергии

$$\Delta W^p = \frac{1}{2} C_n \cdot \left(\frac{E_n}{N} \right)^2. \quad (\text{Д.6.7})$$

Далее открывается $(N - 2)$ транзистор и при закрытых остальных транзисторах очередная порция энергии, определяемая выражением (Д.6.7), возвращается в $(N - 2)$ источник. Таким образом, открывая через промежутки времени Δt транзисторы с меньшими потерями, можно вернуть в источник почти всю энергию, накопленную на емкости C_n . Только часть энергии, определяемая выражением (Д.6.7), термализуется при открытом транзисторе, подключенном к общей шине.

Таким образом, приближенно можно считать, что диссипация энергии при одном цикле производства информации составит величину, определяемую формулой:

$$W = C_n \cdot \frac{E_n^2}{N}. \quad (\text{Д.6.8})$$

Существенным недостатком реверсивной логической схемы, приведенной на рис. Д.6.7, является большое количество источников питания и, соответственно, большое количество шин, занимающих значительную площадь кристалла. Этот недостаток можно устранить, используя вместо источников напряжения конденсаторы большой емкости — «танки». Модифицированная схема представлена на рис. Д.6.7. Если конденсаторы зарядить до напряжений, равных напряжениям источников $E_{n1}, E_{n2}, \dots, E_N$ (рис. Д.6.6) и выбрать емкость «танков» $C_m \ll C_n$, то последовательно открывая $T_1, \dots, T_N$, можно пошаговым образом зарядить C_n до напряжения E_n . На каждом шаге зарядки, согласно модели, приведенной на рис. Д.6.8, напряжение на C_n будет увеличиваться на величину, определяемую выражением:

$$\Delta U_{\text{вых}} \approx \frac{E_n \cdot C_m}{N \cdot (C_m + C_n)}. \quad (\text{Д.6.9})$$

Однако, практически N не может быть произвольно большим, поскольку каждая ступень зарядки требует включения одного из транзисторов ($T_1, T_2, \dots, T_N$) и на это требуется время.

Время включения транзисторов определяется постоянной времени:

$$\tau_i = R_i \cdot C_i, \quad (\text{Д.6.10})$$

где R_i и C_i — сопротивление канала и входная емкость транзистора. Время заряда T таким образом будет определяться выражением:

$$T = R_i \cdot C_n \cdot N. \quad (\text{Д.6.11})$$

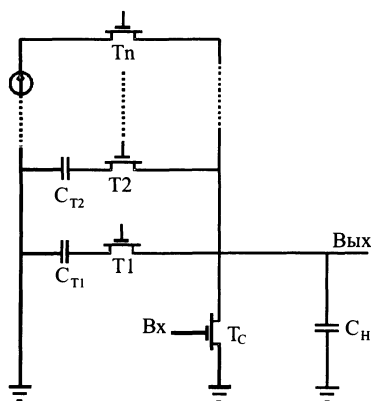


Рис. Д.6.7. Схема, реализующая ступенчатую зарядку нагрузочной емкости.

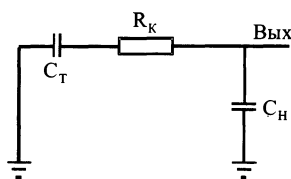


Рис. Д.6.8. Модель ступенчатой зарядки.

Сопротивления каналов R_i транзисторов прямо пропорционально ширине затвора, поэтому для уменьшения времени заряда C_n на каждой ступени необходимо увеличивать ширину затворов. Но при этом будет возрастать входная емкость C_i и возрастет время, необходимое для отпираания транзисторов. Поэтому при заданном T существует оптимальное число ступеней N_{onm} , определяемое выражением:

$$N_{onm} = \sqrt[3]{\frac{T}{4m\tau_{cp}}}, \quad (\text{Д.6.12})$$

где $m = 3$, T — заданное время зарядки, τ_{cp} — среднее значение постоянной времени входной цепи транзисторов.

Величина τ_{cp} определяется выражением:

$$\tau_{cp} = \frac{1}{2N} \cdot \left(\sum_{i=1}^N \tau_i + \sum_{i=0}^{N-1} \tau_i \right). \quad (\text{Д.6.13})$$

Общее количество энергии, диссипированной при разряде и заряде C_n , будет определяться по формуле:

$$W = W^3 + W^p = \left(\frac{1}{N} + 2N^2 m \frac{\tau_{cp}}{T} \right) C_n E_n^2. \quad (\text{Д.6.14})$$

При оптимальном количестве ступеней N_{onm} (Д.6.12) значение количества энергии, диссипированной при заряде и разряде C_n , будет определяться выражением:

$$W_{onm} = \frac{3}{2} \sqrt[3]{\frac{4m\tau_{cp}}{T}} \cdot C_n E_n^2. \quad (\text{Д.6.15})$$

Приведенные примеры не исчерпывают частично диссипативные цифровые схемы. Известны, например, варианты динамических схем и другие. Общим недостатком известных адиабатических логических схем является низкое быстродействие и сложная организация. Первый недостаток носит фундаментальный характер. Второй — вытекает из необходимости использования источника переменного напряжения.

Д.7. Вентили с нетрадиционной организацией энергопитания

Идея использования фундаментального свойства полупроводников — способности непосредственного преобразования энергии ионизирующего излучения в электрическую энергию для питания цифровых схем была впервые реализована в интегральной инжекционной логике. Однако, такой способ организации питания по ряду технических причин оказался не эффективным и не нашел широкого практического применения.

Вторая попытка была предпринята в работе. В цифровых схемах инжекционно — полевой логики впервые удалось обеспечить работу от солнечного света, так сказать дарового источника энергии.

Но и эта логика по ряду причин, которые будут обсуждаться ниже, также не получила пока широкого развития. Предполагалось также использовать для питания цифровых интегральных схем и радиоактивное излучение и другие нетрадиционные источники энергопитания, но и они не находили применения в цифровых схемах.

Однако с развитием технологии «умной пыли» — автономных коллективных систем интеллектуальных датчиков интерес к нетрадиционным способам энергоснабжения возрастает. Появились сообщения об использовании для энергопитания интегральных схем фонового светового и радиоизлучения, вибрации стен зданий, перепадов атмосферного давления и других даровых источников энергии. Более того, одной из основных тенденций развития элементной базы цифровых СБИС является постоянное из года в год снижение напряжения питания. Напряжение питания асимптотически приближается к напряжению фото-ЭДС ($\approx 0,4\text{ В}$), которое может быть получено с помощью солнечной батареи. Эти факты дают основание полагать, что идеи нетрадиционной организации энергопитания будут востребованы и поэтому достойны рассмотрения.

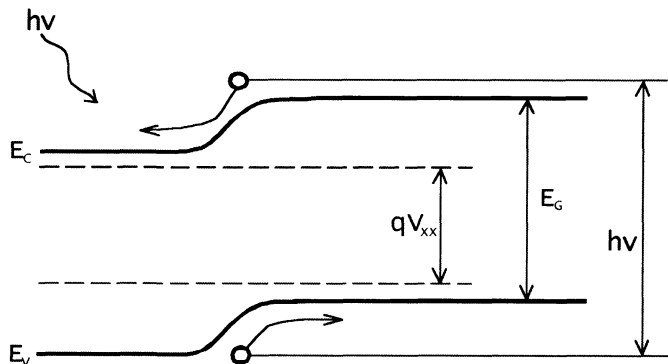
Д.7.1. Питание ионизирующим излучением

Для реализации простейшего логического элемента необходимо и достаточно трех элементов: переключательного прибора, генератора тока и источника энергии.

В схемах с нетрадиционной цепью питания источником энергии являются некие внешние по отношению к ним естественные или искусственные источники. При этом каждый вентиль должен в своем составе иметь преобразователь энергии, который может быть совмещен конструктивно с генератором тока и с элементами переключательного прибора.

Широко известно, что освещенный диод (p - n -переход или диод Шоттки) является хорошим преобразователем энергии и генератором тока. В обычном p - n -переходе с шириной запрещенной зоны E_g при его освещении каждый фотон с энергией, большей E_g , дает в выходную мощность вклад, равный E_g . Остальная часть энергии фотона термолизуется. Для определения эффективности преобразования (или к.п.д.) рассмотрим диаграмму энергетических зон освещаемого p - n -перехода (рис. Д.7.1 а)). Соответствующая эквивалентная цепь показана на рис. Д.7.1 б), где параллельно переходу введен источник постоянного тока I_L , описывающий возбуждение неравновесных носителей излучением.

а)



б)

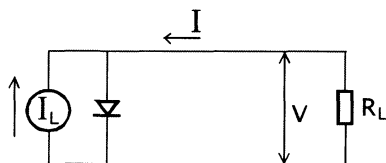


Рис. Д.7.1. Энергетическая диаграмма p - n -перехода при освещении (а) и идеализированная эквивалентная схема солнечного элемента (б). (R_L -нагрузочное сопротивление).

Вольт-амперная характеристика такого прибора определяется выражениями

$$I = I_s \left(e^{\frac{qV}{kt}} - 1 \right) - I_L, \quad (\text{Д.7.1})$$

$$J_s = I_s/S = qn_i / \left(\frac{1}{N_A} \sqrt{\frac{D_n}{\tau_n}} + \frac{1}{N_D} \sqrt{\frac{D_p}{\tau_p}} \right) e^{-E_g/kT}, \quad (\text{Д.7.2})$$

где: S — площадь, I_s — удельное значение тока насыщения, N_A — концентрация акцепторной примеси, N_D — концентрация донорной примеси, n_i — собственная концентрация носителей заряда, D_n — коэффициент диффузии электронов, D_p — коэффициент диффузии дырок, τ_n — время жизни электронов, τ_p — время жизни дырок.

График вольт-амперной характеристики приведен на рис. Д.7.2. Поскольку вольт-амперная характеристика проходит через четвертый квадрант, это означает, что прибор служит источником энергии. При соответствующем подборе нагрузочного сопротивления R_w вырабатываемая энергия может достигать 80% произведения $I_{кз} \cdot V_{xx}$ ($I_{кз}$ — ток короткого замыкания, V_{xx} — напряжение холостого хода элементов).

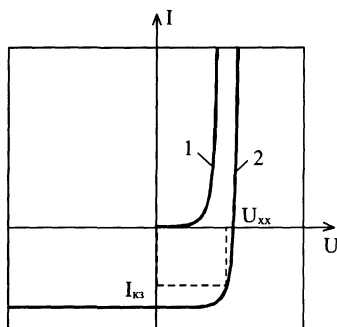


Рис. Д.7.2. Вольт-амперная характеристика p - n -перехода: 1 — неосвещенного; 2 — освещенного.

Из уравнения (Д.7.1) получаем, что напряжение холостого хода при $R_L = \infty$ равно

$$V_{XX} = \frac{kT}{q} \ln \left(\frac{I_L}{I_s} + 1 \right) \approx \frac{kT}{q} \ln \left(\frac{I_L}{I_s} \right). \quad (\text{Д.7.3})$$

Следовательно, при заданном токе I_L напряжения холостого хода возрастает логарифмически при уменьшении тока насыщения I_s . Выходная мощность равна

$$P = IV = I_s V \left(e^{qV/kT} - 1 \right) - I_L V \quad (\text{Д.7.4})$$

Максимальная выходная мощность определяется выражением

$$P_m = I_m V_m \simeq I_L \left[V_{XX} - \varphi_T \ln \left(1 + \frac{1}{\varphi_T} \right) - \varphi_T \right] = I_L (E_m/q), \quad (\text{Д.7.5})$$

где

$$E_m \equiv q \left[V_{XX} - \varphi_T \ln \left(1 + \frac{1}{\varphi_T} \right) - \varphi_T \right] \quad (\text{Д.7.6})$$

Величина E_m соответствует максимальной энергии, которая выделяется на нагрузке при поглощении одного фотона и при оптимальном согласовании элемента с внешней цепью.

Минимальное значение I_s для Si при 300 К составляет приблизительно 10^{-15} А/см².

Идеальная эффективность преобразования равна отношению максимальной выходной мощности к мощности падающего излучения P_0 и может быть определена графически из рисунка:

$$\eta = \frac{P_m}{P_0} = \frac{I_L (E_m/q)}{P_0} = \left[V_m^2 I_s (q/kT) e^{qV_m/kT} \right] / P_0 \quad (\text{Д.7.7})$$

Максимальная эффективность оказывается равной 30% и достигается при $E_g = 1,35$ эВ, если использовать параметры материала, характерные для полупроводников типа $A^{III}B^V$.

Кривая зависимости эффективности преобразования от ширины запрещенной зоны имеет широкий диапазон, в пределах которого она слабо зависит от E_g . Поэтому все полупроводники, которые имеют ширину запрещенной зоны от 1 до 2 эВ пригодны для создания интегральных цифровых схем с питанием от излучения.

Значение мощности падающего излучения определяется мощностью источника света, состоянием окружающей среды, взаимным расположением приемника (поверхности интегральной схемы) и источника (искусственного или естественного).

При использовании солнечного света характерными значениями интенсивности излучения являются интенсивность в свободном пространстве, равная 1353 Вт/м², на поверхности Земли — 925 Вт/м² (когда Солнце стоит в зените), на поверхности Земли — 691 Вт/м² (при угле 60° к горизонту).

Для эффективного использования света для непосредственного питания интегральных цифровых схем можно использовать оптически сконцентрированное излучение. Солнечный свет можно сфокусировать до 844 кВт/см² (интенсивность 1000 Солнц).

Из изложенного выше ясно, что $p-n$ -переход (или диод Шоттки), являясь преобразователем энергии излучения, может служить генератором тока в цепях питания цифровых вентилях. При этом такой источник питания может обеспечить приемлемую скорость поставки энергии для производства информации.

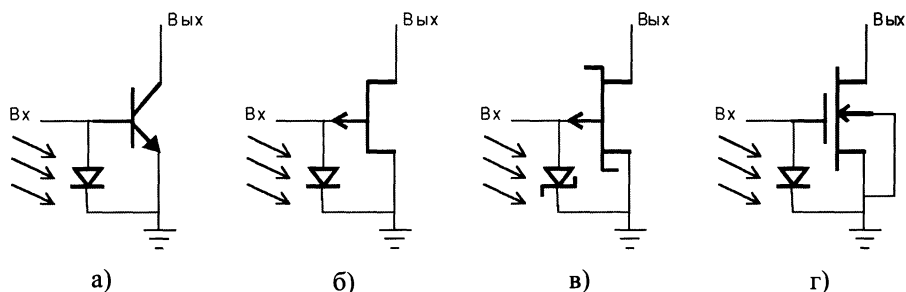


Рис. Д.7.3. Электрические схемы инверторов с питанием от излучения: а) И²Л типа, б) ИПЛ типа, в) ИПС вариант; г) п-МОП типа.

Д.7.2. Схемотехнические и структурно-топологические решения базовых логических вентилях, питающихся излучением

На рис. Д.7.3 приведены электрические схемы простейших логических вентилях с нетрадиционной организацией цепи питания. В отличие от классических вариантов И²Л в этих вентилях функцию генератора тока выполняет не $p-n-p$ -биполярный транзистор, а облучаемый диод. Все эти вентили имеют общий принцип действия, который заключается в следующем. Ионизирующее излучение, например, солнечный свет, через поверхность интегральной схемы проникает в объем полупроводника и поглощается в нем. При этом квантами света генерируются электронно-дырочные пары. Часть носителей заряда, сгенерированных на расстоянии, не превышающем диффузионную длину, достигает $p-n$ -переходов (эмиттер-база и коллектор-база для БТ, и затвор-сток и затвор-исток для ПТУП). В электронно-дырочных переходах происходит их разделение и подхватываемые полем дырки переходят в p -область, а электроны остаются в n -области. Вместе с ростом концентрации носителей заряда возрастает создаваемое ими электрическое поле, направленное против встроенного поля перехода. Возникшее поле будет препятствовать переходу дырок в p -область. Вместе с тем по мере возрастания этого поля возникнет обратное движение дырок в n -область, а элек-

тронов — в p -область. Как известно, генерированные светом дырки являются неосновными для n -области, а в p -области неосновными носителями заряда являются электроны. Следовательно, встроенное поле перехода способствует перемещению неосновных носителей в противоположные области, а возникшее поле способствует перемещению основных носителей.

Состояние равновесия наступит тогда, когда потоки носителей заряда, проходящие через переход в обоих направлениях, не станут равными. В это время между электродами p - n -перехода устанавливается некоторая разность потенциалов, называемая фото-ЭДС. Таким образом, p - n -переход непосредственно преобразует энергию света в электрическую энергию.

И²Л с питанием от излучения. Приведенная на рис. Д.7.3 а) схема инвертора воплощается в интегральную структуру на рис. Д.7.4. Особенностью интегрального воплощения данного варианта логической схемы является совмещение диода — преобразователя энергии излучения в электрическую энергию с p - n -переходом 1 эмиттер-база переключающего биполярного транзистора n - p - n -типа. Такое исполнение цепи питания не требует дополнительной площади на поверхности кристалла и позволяет достигнуть высокой плотности компоновки еще и потому, что на поверхности нет традиционных шин питания и общей шины (роль которой выполняет n + подложка). Простота конструкции обеспечивает и малооперационную технологию изготовления интегральных схем данного типа. Как видно из рис. Д.7.4, в структуре имеются также p - n -переходы 2, которые могут в принципе выполнять функции диодов — преобразователей энергии. Однако их вклад в ток питания незначителен из-за высоких уровней легирования коллекторов и базы. Сгенерированные квантами света электронно-дырочные пары в областях базы и коллектора в большинстве своем не достигают области объемного заряда p - n -перехода, где происходит их разделение и не вносят вклада в фототок, поскольку эта диффузионная область является нереконпрессированной, и носители заряда имеют в ней малую диффузионную длину.

Основной вклад в фототок дает нижняя область p - n -перехода, находящаяся в некомпенизированной эпитаксиальной пленке. При этом в верхней диффузионной области поглощается значительное количество квантов света.

В интегральной структуре используется, так называемый, инверсный транзистор (эмиттером служит подложка 3, а коллекто-

ром — верхние $n+$ диффузионные области). Инверсный транзистор имеет существенные недостатки: коэффициент усиления составляет несколько единиц, его быстродействие сравнительно невелико из-за эффекта накопления избыточного заряда в базовой и эмиттерной областях в режиме насыщения. Первый недостаток определяет особенности структуры и схемотехники логических схем. Для логических схем данного класса является характерным использование многоколлекторных структур (см. рис. Д.7.4) и минимальная нагрузочная способность. Каждый коллектор-выход соединяется только с одной базой-входом последующего каскада. Типичная схемная конфигурация приведена на рис. Д.7.5. В схемах широко используется простейший логический элемент «монтажное И». Второй недостаток И²Л схем — сравнительно низкое быстродействие (время задержки переключения составляет сотни наносекунд).

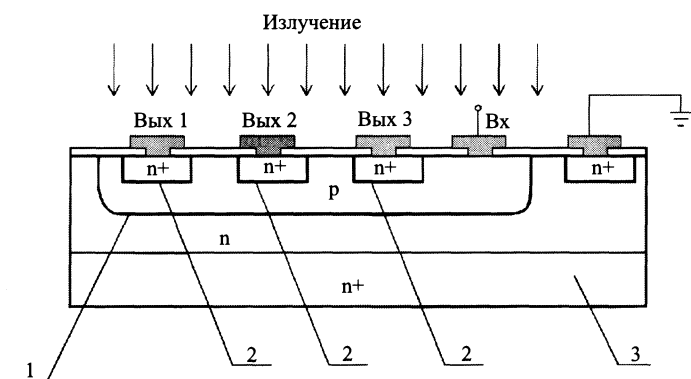


Рис. Д.7.4. Разрез структуры многоколлекторного И²Л-элемента: 1 — переход база-эмиттер; 2 — коллектор; 3 — эмиттер.

ИПЛ вентили с питанием от излучения. Электрическая схема ИПЛ инвертора (рис. Д.7.3 б)) содержит нормально закрытый ПТУП, к истоку и затвору которого подключен p - n -переход, преобразующий энергию излучения в электрический ток (рис. Д.7.6 а)). Роль такого преобразователя может выполнять переход затвор-исток полевого транзистора. В отличие от большинства известных схем в ИПЛ схемах переключатели — полевые транзисторы работают в режиме прямого смещения переход затвор-исток. При нулевом смещении на затворе канал полевого транзистора должен быть перекрыт обедненными областями перехода затвор-исток, обусловленными контактной разностью потенциалов между затвором и истоком. При положительном смещении на затворе ширина обеднен-

ных областей уменьшается, что вызывает уменьшение сопротивления канала. При этом имеет место инжекция неосновных носителей заряда из области затвора в исток.

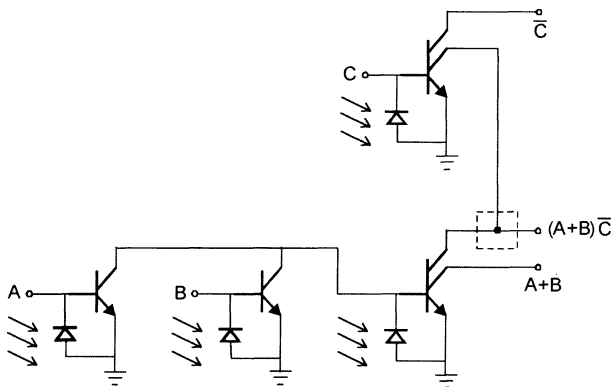


Рис. Д.7.5. Типичная схемная конфигурация И²Л с питанием от света.

Прямое смещение на переходе затвор-исток полевого транзистора в ИПЛ инверторе создается при освещении p - n -перехода — преобразователя энергии. Также как и в инжекционных схемах, питаемых излучением, каждый из ИПЛ инверторов снабжен источником тока — преобразующим p - n -переходом. Распределенный характер источника питания, функционально интегрированного с переключающими транзисторами, способствует повышению эффективности преобразования энергии излучения, а также повышению надежности ИПЛ схем.

Диод — преобразователь можно рассматривать как эквивалентную нагрузку переключающего полевого транзистора. Известно, что при освещении p - n -перехода его вольт-амперная характеристика (рис. Д.7.6 б)) смещается по оси токов на величину, равную фототоку J_F

$$J_G = J_{G0} \left[\exp \left(\frac{U_{GS}}{m\phi_T} \right) - 1 \right] - J_F, \quad (\text{Д.7.8})$$

где J_G — ток затвора, J_{G0} — начальный ток перехода затвор-исток, U_{GS} — смещение на переходе затвор-исток, $m\phi_T$ — эффективный температурный потенциал ($m = 1 \dots 2$).

Сопоставление вольт-амперных характеристик освещенного p - n -перехода и полевого транзистора (рис. Д.7.6) позволяет определить логические уровни и область работоспособности ИПЛ схемы. Как видно из рис. Д.7.6 величина минимального фототока, обес-

печивающая работоспособность ИПЛ инвертора, определяется начальным током стока при нулевом смещении на затворе

$$J_{F \min} \simeq J_{D0} = J_D|_{U_{GS}=0}. \quad (\text{Д.7.9})$$

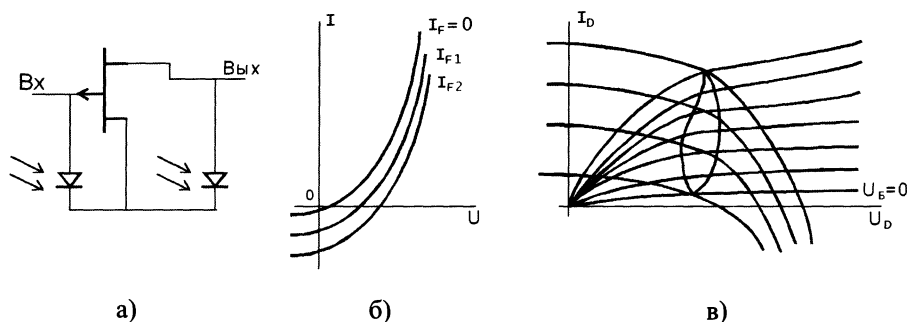


Рис. Д.7.6. ИПЛ элемент с питанием от света: а) электрическая схема; б) ВАХ освещенного p - n -перехода; в) ВАХ ПТУП.

С увеличением фототока логический перепад возрастает и достигает максимума, а затем начинает уменьшаться. Уменьшение логического перепада при больших значениях фототока объясняется увеличением крутизны нагрузочной характеристики. Анализ нагрузочной и выходной вольт-амперной характеристик ИПЛ инвертора показывает, что логические перепады не превышают напряжения холостого хода на освещенном p - n -переходе и составляют десятки и сотни милливольт.

Конструктивно интегральный ИПЛ инвертор выполняется в виде ПТУП с вертикальным каналом. Роль преобразователя энергии выполняет переход затвор-исток (рис. Д.7.7 а)). Описанную структуру можно создать, например, путем диффузии бора и фосфора в монокристаллическую подложку n -типа. В отличие от инжекционных схем, питаемых излучением, для ИПЛ схем необходима не низкоомная, а высокоомная подложка. Удельное сопротивление подложки определяется шириной канала полевого транзистора. Действительно, из условия перекрытия канала с шириной $2a$ при нулевом смещении на затворе для модели транзистора с резкими p - n переходами получаем следующее выражение для расчета концентрации доноров в подложке:

$$N_D = \frac{2\varepsilon\varepsilon_0\varphi_k}{qa^2}, \quad (\text{Д.7.10})$$

где q — заряд электрона, ε — диэлектрическая проницаемость полу-

проводника, ϕ_k — контактная разность потенциалов, равная

$$\varphi_k = \varphi_T \ln \frac{N_A N_D}{n_i^2}. \quad (\text{Д.7.11})$$

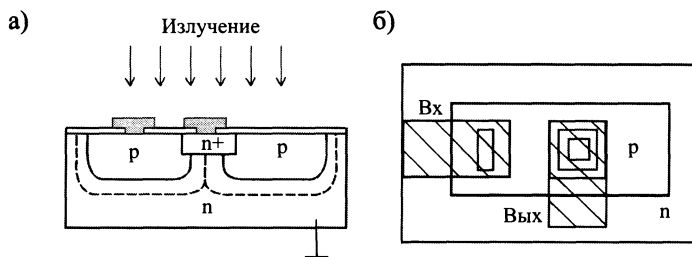


Рис. Д.7.7. ИПЛ-инвертор с питанием от излучения: а) разрез физической структуры, б) топология.

Оценка величины N_D , приведенная по формулам (Д.7.10), (Д.7.11) для прибора с шириной канала 1 мкм и концентрацией акцепторов в области затвора 10^{17} см^{-3} дает значение 10^{15} см^{-3} , что соответствует удельному сопротивлению подложки 4,5 Ом·см.

Сравнение фотоэлектрических характеристик p - n -перехода преобразователя энергии для инжекционных полевых структур показывает, что и в последних осуществляется более эффективное преобразование энергии излучения в фототок. Известно, что КПД преобразования определяется суммой квантовых эффективностей p - и n -областей перехода

$$\eta = \eta_n + \eta_p. \quad (\text{Д.7.12})$$

При прочих равных условиях квантовая эффективность n -области определяется количеством генерированных светом дырок, достигших p - n -перехода

$$\eta_n \simeq \int_{x_j}^{\infty} \exp(-\alpha x) \exp\left(-\frac{x - x_j}{L_P}\right) dx = \frac{\exp(-\alpha x_j)}{\alpha + L_P^{-1}}, \quad (\text{Д.7.13})$$

где α — коэффициент поглощения света, x_j — глубина залегания p - n -перехода, L_P — диффузионная длина дырок в подложке.

Как следует из выражений (Д.7.12) и (Д.7.13), большая часть диффузионной длины L_P в ИПЛ структурах по сравнению с инжекционными обеспечивает более эффективное преобразование энергии излучения.

ИПЛ элементы с питанием от излучения могут быть выполнены также, как функционально-интегрированные структуры на многозатворных ПТШ и ПТУП (см. раздел Д.5.2).

Вентили с питанием от излучения на МОП транзисторах.

На рис. Д.7.3 г) приведена одна из принципиально возможных электрических схем такого типа.

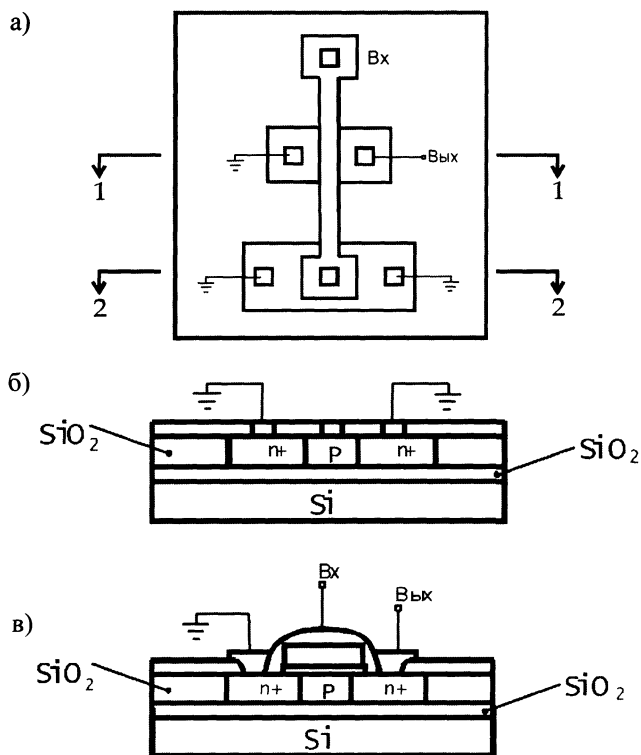


Рис. Д.7.8. Инвертор с питанием от излучения на МОП-транзисторах: а) топология, б) разрез диода-преобразователя энергии (1–1), в) разрез переключающего n-МОП транзистора (2–2).

В отличие от И²Л и ИПЛ данная электрическая схема может быть воплощена в интегральную структуру с полной диэлектрической изоляцией без совмещения диода с транзистором, приведенную на рис. Д.7.8. Структурв может быть выполнена по технологии с полной диэлектрической изоляцией типа SOI. Особенностью работы данного вентиля является необходимость выполнения условия:

$$U_{\Phi} \geq V_{\text{п}}, \quad (\text{Д.7.14})$$

где U_{Φ} — фото-ЭДС преобразование энергии, $V_{\text{п}}$ — пороговое напряжение n-МОП транзистора.

Полная диэлектрическая изоляция позволяет в данном случае реализовать на одном кристалле как последовательный цепи пере-

ключательных транзисторов, так и параллельные, а также смешанные последовательно — параллельные. Достоинством данной конструкции является также возможность подсветки структур снизу через данный слой изолирующего слоя двуокиси кремния. Другим достоинством является их технологическая совместимость с классическими КМОП системами и схемами ИПЛ типа. В случае невыполнения условия (Д.7.14) в цепи питания необходимо использовать несколько последовательно соединенных диодов-преобразователей.

Достоинством всех рассмотренных схемотехнических и структурно-топологических решений цифровых схем с питанием от излучения являются:

- отсутствие традиционных аккумуляторов и батарей, которое позволяет уменьшить вес и габариты цифрового устройства;
- наивысшая плотность компоновки;
- повышенная надежность, поскольку из конструкции исключаются длинные, понизывающие весь кристалл шины питания и общая шина.

Цифровые логические схемы с питанием от излучения идеально подходят для создания цифровых устройств с автономным питанием.

Д.8. Заключение

В 1958 г. были изобретены интегральные схемы, благодаря которым электроника стала микроэлектроникой. Что дали эти достаточно простые инженерные решения общеизвестно.

На протяжении более чем сорокалетней истории развития микроэлектроники ведется постоянное совершенствование элементной базы. Микроэлектроника плавно трансформируется в наноэлектронику. Этот процесс носит эволюционный характер — прежде всего уменьшаются размеры классических транзисторов, совершенствуется их физическая структура. Одновременно с этим процессом ведутся интенсивные поиски новых приборных принципов функционирования, схемотехнических и структурно-топологических решений элементной базы, которые обеспечили бы более высокое быстродействие и лучшую энергетику.

ГЛАВА 6

ЛОГИЧЕСКИЕ СХЕМЫ

Логическая схема представляет собой функциональный узел, который выдает выходную величину, зависящую только от значений входных переменных в данный момент времени. Он описывается переключательной функцией. Ниже представлены методы, позволяющие проводить минимизацию переключательных функций графическим способом или с помощью таблиц. Минимизированная функция KDNF-типа соответствует дизъюнктивной нормальной форме, минимизированная функция KKNF — конъюнктивной нормальной форме (KNF).

6.1. Минимизация с помощью диаграмм Карно–Вейча

6.1.1. Минимизация KDNF

Метод минимизации логической схемы с помощью диаграмм Карно–Вейча хорошо подходит для проектирования подобных схем «вручную». Для пояснения этого метода используется приведенный ниже пример. Подлежащая минимизации переключательная функция определена в табл. 6.1.

Таблица 6.1. Пример переключательной функции

	x_3	x_2	x_1	x_0	y
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0

	x_3	x_2	x_1	x_0	y
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Для минимизации применяются диаграммы, в которых каждое поле точно соответствует дизъюнкции входных переменных, следовательно, соответствует одному минтерму. Эти диаграммы именуются диаграм-

мами Карно-Вейча (KV-диаграмма). На рис. 6.1 показаны две KV-диаграммы, в которых поля обозначены через минитермы или, соответственно, через значения функций-комбинаций входных переменных. Диаграмма сконструирована таким образом, чтобы при переходе от одного поля к другому изменялась только одна переменная.

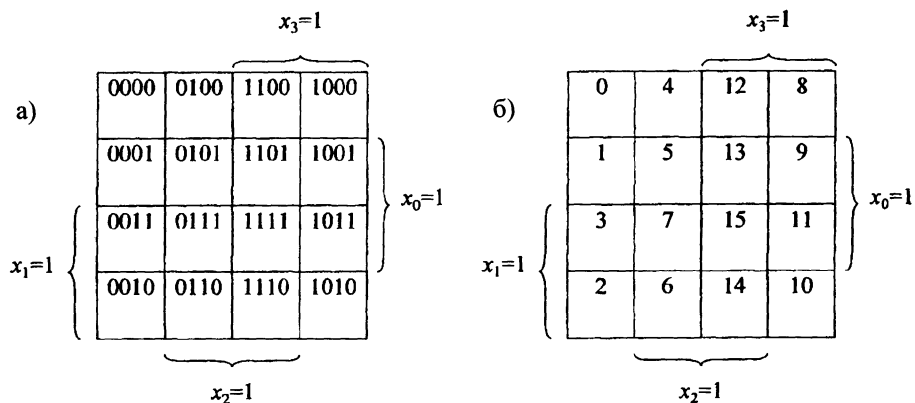


Рис. 6.1. Диаграммы Карно-Вейча для 4 входных переменных: а) с двоичным обозначением полей; б) с обозначением через минитермы.

В данной диаграмме для минимизации KDNF маркируются минитермы переключательной функции. Данный пример приводим в результате к диаграмме:

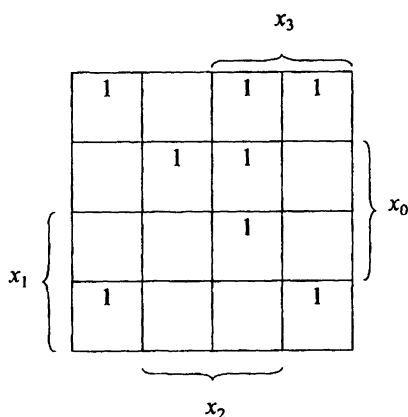


Рис. 6.2. Диаграмма Карно-Вейча с минитермами функции, показанной на рис. 6.1.

Теперь соседние поля, которые отличаются только одной переменной, могут быть соединены в соответствии с переместительным

законом (уравнение (3.34)):

$$(x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1) = x_0 \quad (6.1)$$

На основе этого могут быть образованы весьма большие области полей с 1. Но приемлемы только прилегающие друг к другу области с 1, 2, 4, 8 и т.д. полями. Эти поля описываются конъюнкцией входных переменных, которые называются импликантами. При этом мысленно соединяют левую сторону полей с правой стороной, точно также как и верхнюю сторону с нижней. Импликант, состоящий из 4 входных переменных, состоит из одного поля (в случае функции с четырьмя переменными). Если импликант имеет на одну переменную меньше, то при этом число полей удваивается. Поэтому для минимизации затрат на вентили формируются максимально возможно большие поля.

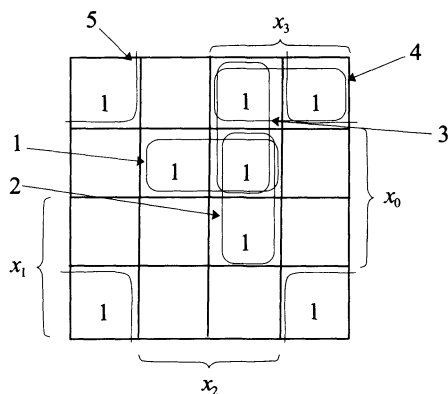


Рис. 6.3. Диаграмм Карно-Вейча с минтермами функции из табл. 6.1

Находим для области 1 импликант I_1 :

$$\text{область } I_1 = x_0 \neg x_1 x_2$$

Убеждаемся, что никакой другой из импликантов полностью не перекрывает I_1 . Импликант функции I_1 называют первичным импликантом, поскольку не имеется какого-либо другого импликанта I_X , который бы полностью перекрывал I_1 . Импликанты дизъюнктивной формы (DNF) называют термами логического произведения (product term).

Приведенные в примере импликанты промаркированы цифрами от 1 до 5. Иные первичные импликанты подобрать нельзя. Для других маркированных первичных импликантов можно с помощью

переменных на краю диаграммы определить конъюнкции, которые однозначно образуют следующие области:

$$\text{область 2: } I_2 = x_0 x_2 x_3$$

$$\text{область 3: } I_3 = \neg x_1 x_2 x_3$$

$$\text{область 4: } I_4 = \neg x_0 \neg x_1 x_3$$

$$\text{область 5: } I_5 = \neg x_0 \neg x_2$$

В диаграмме для четырех входных переменных область из четырех полей соответствует импликанту с двумя переменными, как это имеет место для импликанта I_5 . Этот импликант лежит в четырех углах диаграммы, которая рассматривается как связанная.

Различают:

- основные, первичные импликанты P_K :

Первичный импликант является основным первичным импликантом в том случае, если он не перекрывает дизъюнкции всех других первичных импликантов. Следовательно, основные импликанты соответствуют 1, которую они в одиночку покрывают. Основные первичные импликанты в каждом случае представляются в минимизированной форме DNF.

- абсолютно элиминируемые первичные импликанты P_A :

Первичный импликант элиминируем (то есть устраним) в том случае, когда он полностью перекрывается основным первичным импликантом. Он является избыточным.

- относительно элиминируемые первичные импликанты P_R :

Все остальные первичные импликанты называются относительно элиминируемыми первичными импликантами. Выборка относительно элиминируемых первичных импликантов может быть взята из минимизированной формы DNF.

Например, имеются множества:

$$P_K = \{I_1, I_2, I_5\}$$

$$P_A = \emptyset$$

$$P_R = \{I_3, I_4\}$$

Минимизированная переключательная функция составляется из основных первичных импликантов а выборка из относительно элиминируемых первичных импликантов таким образом, чтобы все минитермы были покрыты. Следовательно, упрощенная функция будет

справедлива тогда, когда устраняются относительно элиминируемые первичные импликанты 4:

$$f(x_3, x_2, x_1, x_0) = x_0 \neg x_1 x_2 \vee x_0 x_2 x_3 \vee \neg x_1 x_2 x_3 \vee \neg x_0 \neg x_2 \quad (6.2)$$

И тогда, когда элиминируются первичные импликанты 3:

$$f(x_3, x_2, x_1, x_0) = x_0 \neg x_1 x_2 \vee x_0 x_2 x_3 \vee \neg x_0 \neg x_1 x_3 \vee \neg x_0 \neg x_2 \quad (6.3)$$

6.1.2. Минимизация нормальной KKNF

Метод минимизации KKNF основывается на использовании макстермов. На местах единиц следует рассматривать нули. В аналогичном приведенному выше примере в диаграмму вносятся макстермы.

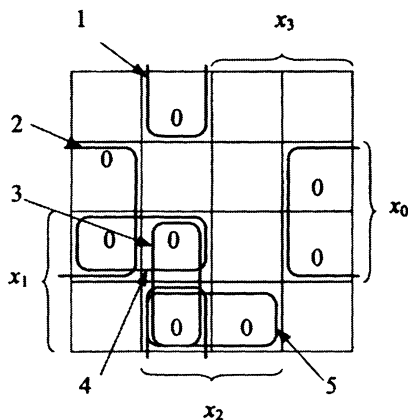


Рис. 6.4. Диаграмма Карно-Вейча с макстермами для функции из табл. 6.1.

Действуя по тем же правилам, что и при определении DNF, отметим возможно наибольшие области полей с 0. Отмеченные на рис. 6.4 области представляют собой первичные импликанты конъюнктивной нормальной формы (KNF). Они представлены дизъюнкциями входных переменных, которые вне данных областей выдают значения функции, равные 1:

$$\text{область 1: } I_1 = x_0 \vee \neg x_2 \vee x_3$$

$$\text{область 2: } I_2 = \neg x_0 \vee x_2$$

$$\text{область 3: } I_3 = \neg x_1 \vee \neg x_2 \vee x_3$$

$$\text{область 4: } I_4 = \neg x_0 \vee \neg x_2 \vee x_3$$

$$\text{область 5: } I_5 = x_0 \vee \neg x_1 \vee \neg x_2$$

Следовательно, в данном примере получаем множества:

$$P_K = \{I_1, I_2, I_5\}$$

$$P_A = \emptyset$$

$$P_R = \{I_3, I_4\}$$

Минимальную форму можно получить путем применения основных первичных импликантов и импликанта I_3 :

$$\begin{aligned} f(x_3, x_2, x_1, x_0) &= \\ &= (x_0 \vee \neg x_2 \vee x_3)(\neg x_0 \vee x_2)(\neg x_1 \vee \neg x_2 \vee x_3)(x_0 \vee \neg x_1 \vee \neg x_2) \end{aligned} \quad (6.4)$$

Вторую из возможных минимальных форм KNF можно вывести с помощью применения основных первичных импликантов и импликанта I_4 :

$$\begin{aligned} f(x_3, x_2, x_1, x_0) &= \\ &= (x_0 \vee \neg x_2 \vee x_3)(\neg x_0 \vee x_2)(\neg x_0 \vee \neg x_1 \vee x_3)(x_0 \vee \neg x_1 \vee \neg x_2) \end{aligned} \quad (6.5)$$

6.1.3. Диаграммы Карно–Вейча для 2, 3, 4, 5, 6 входных переменных

Здесь вы можете найти порядок составления различных диаграмм Вейча с внесенными десятичными эквивалентами. Диаграммы Карно–Вейча с более чем пятью переменными применяются очень редко, поскольку они не наглядны.

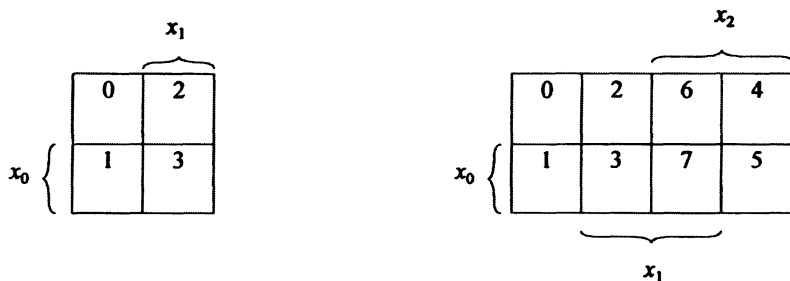


Рис. 6.5. Диаграммы Карно–Вейча для 2 и 3 входных переменных.

6.1.4. Неполностью заданные функции.

Иногда функция задается неполностью. В таком случае некоторые значения функций могут быть выбраны произвольно. Они маркируются в диаграмме Карно буквой d (don't care). Эти don't care

минтермы могут быть использованы для минимизации функции. В следующем примере (рис. 6.8) приведена функция, заданная ее диаграммой Карно.

	0	4	12	8	24	28	20	16	
	1	5	13	9	25	29	21	17	
	3	7	15	11	27	31	23	19	
	2	6	14	10	26	30	22	18	

Рис. 6.6. Диаграмма Карно-Вейча для 5 входных переменных.

	0	8	24	16	48	56	40	32	
	1	9	25	17	49	57	41	33	
	3	11	27	19	51	59	43	35	
	2	10	26	18	50	58	42	34	
	6	14	30	22	54	62	46	38	
	7	15	31	23	55	63	47	39	
	5	13	29	21	53	61	45	37	
	4	12	28	20	52	60	44	36	

Рис. 6.7. Диаграмма Карно-Вейча для 6 переменных.

		x_2	
		0	1
x_0		d	d
		1	0
		x_1	

Рис. 6.8. Пример неполностью заданной функции.

Теперь первичные импликанты при условии включения полей вида d, могут быть выделены так, чтобы можно было обрабатывать максимально возможные области. При этом полям вида d могут быть предписаны значения 0 или 1.

		x_2			
		0	1	d	d
x_0		1	d	1	0
		x_1			

Рис. 6.9. Первичные импликанты для примера, показанного на рис. 6.7.

Отсюда для минимизированной формы получаем:

$$f(x_3, x_2, x_1, x_0) = x_0 \neg x_2 \vee x_1 \quad (6.6)$$

Без применения термов вида don't care (то есть с $d = 0$) получили бы следующую минимизированную форму:

$$f(x_3, x_2, x_1, x_0) = x_0 \neg x_1 \neg x_2 \vee \neg x_0 x_1 \neg x_2 \vee x_0 x_1 x_2 \quad (6.7)$$

Следовательно, с помощью термов вида don't care функцию можно представить более просто.

6.2. Способ Квина–Мак–Класки

К способам минимизации логических схем, которые пригодны для компьютерной реализации, относится способ Квина–Мак–Класки.

В основе его лежат таблицы, процесс обработки которых соответствует уравнению (3.34):

$$(x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1) = x_0. \quad (6.8)$$

Функция представлена с использованием минтермов, выполненных на основе двоичного эквивалента. Для выступающей в минтерме переменной установлено обозначение 1, для переменной с отрицанием — обозначение 0 и для не появляющейся переменной обозначение $(-)$.

Например:

$$x_3 \neg x_2 x_0 \quad \text{записывается как: } 10-1$$

Данный способ представим ниже с использованием примера, приведенного в табл. 6.1. Минтермы переключательной функции внесены в таблицу (табл. 6.2), в которой они собраны в группы с одинаковым числом 1-элементов. Столбцы содержат: десятичный эквивалент и группу (то есть число единичных элементов двоичного эквивалента).

Таблица 6.2. Упорядочение минтермов по группам с равным числом 1-элементов.

Десятичные числа	x_3	x_2	x_1	x_0	Группа
0	0	0	0	0	0
2	0	0	1	0	1
8	1	0	0	0	1
5	0	1	0	1	2
10	1	0	1	0	2
12	1	1	0	0	2
13	1	1	0	1	3
15	1	1	1	1	4

Далее в табл. 6.3 в отдельные строчки собраны термы следующих друг за другом групп, отличающихся одним разрядом. Эта таблица является результатом применения уравнения (6.8). Разряд, в котором элементы различаются, помечен чертой $(-)$. Для формирования десятичного эквивалента внесены десятичные числа минтермов, из которых составлен новый терм.

В данном примере 0 и 1 могут быть объединены, поскольку они различаются только разрядом x_1 . Все термы, которые позволяют их объединить, промаркированы в табл. 6.2 знаком (поскольку, например, минтермы 0 и 1 сплавлены вместе, они маркируются

в табл. 6.2 с помощью одного знака). Не маркированные первичные термы представляют собой первичные импликанты, они появляются в минимизированной переключательной функции (в данной примере это еще не имело места).

Таблица 6.3. Объединение минтермов в группы с одинаковым числом 1-элементов (вариант 1).

Десятичные числа	x_3	x_2	x_1	x_0	Группа
0, 2	0	0	–	0	0
0, 8	–	0	0	0	0
2, 10	–	0	1	0	1
8, 10	1	0	–	0	1
8, 12	1	–	0	0	1
5, 13	–	1	0	1	2
12, 13	1	1	0	–	2
13, 15	1	1	–	1	3

При формировании табл. 6.4 вновь использован данный способ. Снова производится объединение элементов следующих друг за другом групп, приведенных в табл. 6.3. Вновь объединяются термы, которые различаются только на один двоичный разряд.

Если в двоичном эквиваленте мы имеем несколько одинаковых термов, то в этом случае все термы, кроме одного, вычеркиваются.

Обработка с помощью данного способа продолжается до тех пор, пока можно объединять вместе какие-либо термы. Не отмеченные галочкой термы представляют собой первичные импликанты. Следовательно, к первичным импликантам относятся:

8, 12

5, 13

12, 13

13, 15

0, 2, 8, 10

Таблица 6.4. Объединение минтермов в группу с равным числом 1-элементов (2-й вариант); 3-вычеркивание (1 строка).

Десятичные числа	x_3	x_2	x_1	x_0	Группа
0, 2, 8, 10	–	0	–	0	0
0, 8, 2, 10	–	0	–	0	0

Теперь следует провести классификацию первичных импликантов, разделив их на основные первичные импликанты, абсолютно

элиминируемые первичные импликаты и относительно элиминируемые первичные импликаты. Это достигается с помощью следующей таблицы, которую можно назвать таблицей первичных импликант. На ординате отложены минтермы переключательной функции, вдоль абциссы — первичные импликаты. Те минтермы, которые содержатся в первичном импликанте, отмечены значком $\times$.

В том случае, когда в столбце находится только один значок $\times$, соответствующий ему первичный импликант является основным первичным импликантом. Охваченные им минтермы отмечены значком в кружке $\otimes$. В данном примере минтермы 0, 2 и 10 охвачены только основным первичным импликантом 0, 2, 8, 10, появившимся в минимизированной DNF. Охваченные им минтермы 0, 2, 8 и 10 отмечены, в том числе и в других строках, значком $\otimes$.

Таблица 6.5. Приведенная в качестве примера таблица с первичными импликантами

	0	2	5	8	10	12	13	15
8, 12				$\times$		$\times$		
5, 13			$\times$				$\times$	
12, 13						$\times$	$\times$	
13, 15							$\times$	$\times$
0, 2, 8, 10	$\times$	$\times$		$\times$	$\times$			

Импликаты 5, 13 и 13, 15 также являются основными первичными импликантами, поскольку только они охватывают по одному минтерму 5 и 15. Охваченные минтермы 5, 3 и 15 отмечены значком ($\otimes$).

Таблица 6.6. Приведенная в качестве примера таблица с первичными импликантами, минтермы в которой отмечены значком $\otimes$.

	0	2	5	8	10	12	13	15
8, 12				$\otimes$		$\times$		
5, 13			$\otimes$				$\otimes$	
12, 13						$\times$	$\otimes$	
13, 15							$\otimes$	$\otimes$
0, 2, 8, 10	$\otimes$	$\otimes$		$\otimes$	$\otimes$			

Из оставшихся первичных импликантов, которые являются относительно элиминируемыми первичными импликантами, подбирается минимальное число, позволяющее охватить остающиеся минтермы. Затем на их основе формируется совместно с основными первичными импликантами минимальная форма переключательной

функции. Например, для остающегося минтерма 12 могут быть выбраны первичные импликанты 8, 12 или 12, 13.

Таблица 6.7. Установление связи между импликантами.

Десятичные числа	x_3	x_2	x_1	x_0	Импликант
8, 12	1	–	0	0	$x_3 \neg x_1 \neg x_0$
5, 13	–	1	0	1	$x_2 \neg x_1 x_0$
12, 13	1	1	0	–	$x_3 x_2 \neg x_1$
13, 15	1	1	–	1	$x_3 x_2 x_0$
0, 2, 8, 10	–	0	–	0	$\neg x_2 \neg x_0$

Итак, при применении первичных импликантов 12, 13 можно получить:

$$f(x_3, x_2, x_1, x_0) = x_2 \neg x_1 x_0 \vee x_3 x_2 x_0 \vee x_3 x_2 \neg x_1 \vee \neg x_2 \neg x_0 \quad (6.9)$$

или, если применить импликанты 8, 12:

$$f(x_3, x_2, x_1, x_0) = x_2 \neg x_1 x_0 \vee x_3 x_2 x_0 \vee x_3 \neg x_1 \neg x_0 \vee \neg x_2 \neg x_0 \quad (6.10)$$

Эти уравнения идентичны минимизированным формам, найденным с помощью диаграммы Карно-Вейча.

6.3. Другие направления оптимизации

Логическая схема, описанная с помощью нормальных форм KDNF и KKNF либо с помощью минимизированных форм DNF и KNF, может быть реализована напрямую в виде двухступенчатой управляющей схемы. Следует учитывать, что двухступенчатая схема имеет удвоенное время задержки, если мы пренебрегли задержкой инвертора или если в нашем распоряжении имеются инвертированные входные переменные.

Но при реализации необходимо, как правило, соблюдать и другие ограничения:

- Часто управляющая схема должна быть построена на основе вентилях одного типа, например, NOR или NAND;
- Часто задается максимальная величина времени задержки, так что рассматриваться могут только двухступенчатые управляющие схемы;
- Совместно должны минимизироваться большое число функций;

- Как правило, задается максимальное число термов логического произведения в программируемых узлах.

На некоторые из этих особенностей будет указано в последующем при реализации логических схем.

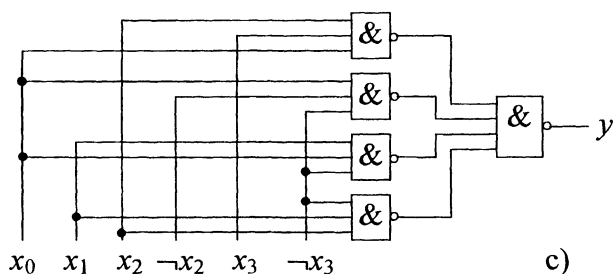
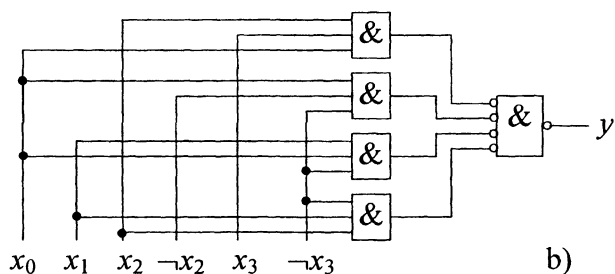
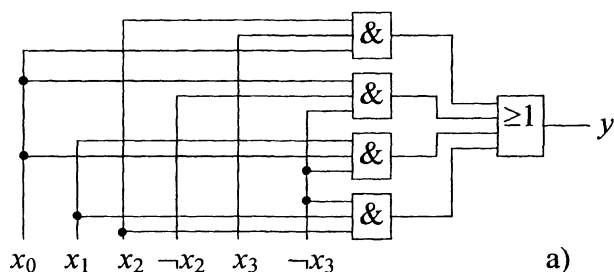


Рис. 6.10. а) Логическая схема на основе DNF; б) преобразование вентиля ИЛИ; в) перемещение инверсионных кружков.

6.3.1. Преобразование логической схемы И/ИЛИ в схему НЕ-И

Пусть показанная на рис. 6.10 а логическая схема, которая может быть получена из DNF, должна быть преобразована в логическую схему, состоящую только из вентилях НЕ-И. В соответствии с правилом

Моргана вначале преобразуем вентиль ИЛИ в вентиль И (рис. 6.10 б). Затем сдвигаем инверсионные кружки с входа этого И-вентиля на выходы И-вентилей, после чего получаем логическую схему, состоящую только из вентилях НЕ-И (рис. 6.10 с).

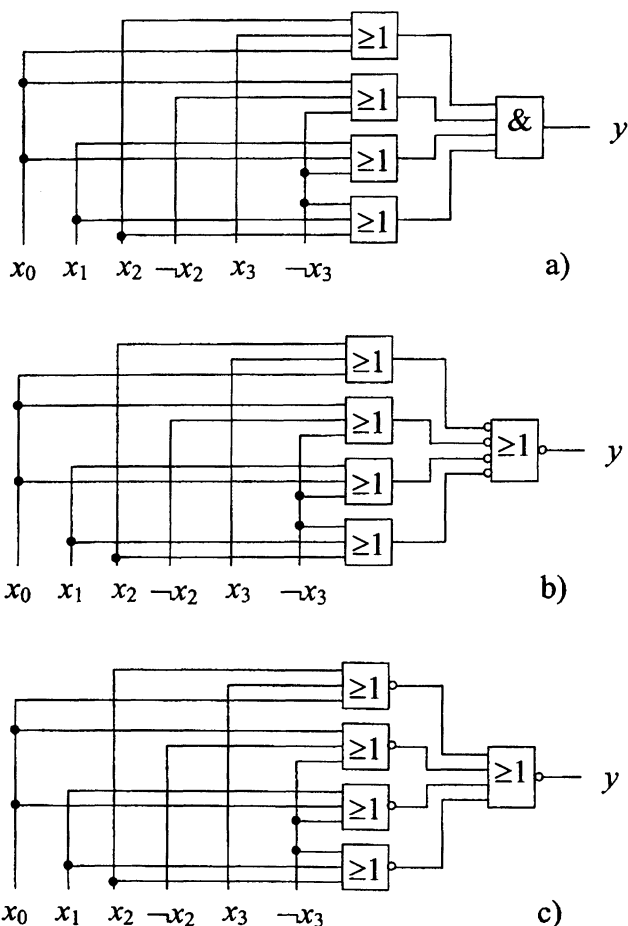


Рис. 6.11. а) логическая схема на основе KNF; б) преобразование вентиля И; с) перемещение инверсионных кружков.

6.3.2. Преобразование логической схемы ИЛИ/И в логическую схему НЕ-ИЛИ

При преобразовании логической схемы ИЛИ/И в логическую схему, состоящую только из вентилях НЕ-ИЛИ, действуют аналогично вышеизложенному. На рис. 6.11 показано, что за счет преобразования И-вентиля на выходе (рис. 6.11 б) и перемещения инверсионных

кружков (рис. 6.11 с) возникает логическая схема, состоящая только из вентилей НЕ-ИЛИ.

6.4. Воздействие времени задержки на логические схемы

6.4.1. Отрицательное воздействие на структуру

До сих пор принималось, что «время задержки» равно нулю. Это означает, что выходные сигналы появляются немедленно, без задержки. На практике это предположение является слишком оптимистичным. Если принять, что время задержки вентиля конечно, на выходе логической схемы могут появляться периодические ложные сигналы. Данный эффект называют «структурным риском». На рис. 6.12 показан вентиль, реализующий функцию:

$$y = x_1 x_0 \vee x_2 \neg x_0 \quad (6.11)$$

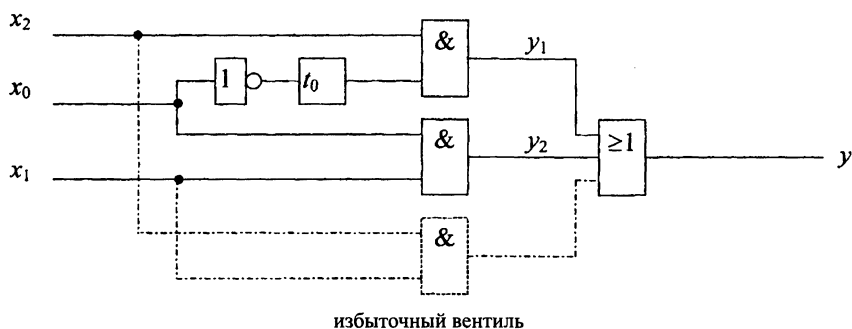


Рис. 6.12. Логическая схема со структурным риском.

Время задержки сигнала в инверторе равно t_0 . Если величины задержки вентилях И равны между собой, нет необходимости учитывать их при рассмотрении времени задержки в И- или ИЛИ-вентильях.

Временные зависимости сигналов $x_0(t)$, $y_1(t)$, $y_2(t)$ и $y(t)$ показаны на рис. 6.13. Во временной зависимости выходного сигнала $y(t)$ можно видеть провал с длительностью t_0 , который возникает из-за временной задержки в инверторе. В идеальной схеме он бы не появился.

При рассмотрении процесса в диаграмме Карно-Вейча (рис. 6.14) устанавливаем, что имеется на лицо переход между двумя первичными импликантами. Коррекцию ошибки можно повести с помощью

вентиля, реализующегося терм x_1x_2 . Теперь логическая схема будет описываться следующей функцией:

$$y = x_1x_0 \vee x_2\bar{x}_0 \vee x_1x_2 \quad (6.12)$$

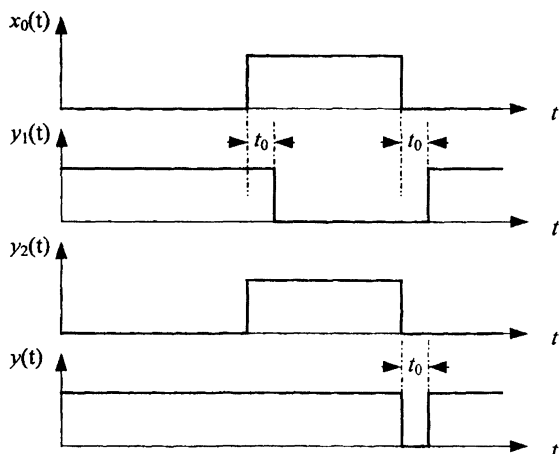


Рис. 6.13. Временная зависимость сигналов при переключении сигнала x_0 в схеме, показанной на рис. 6.12 ($x_1 = x_2 = 1$).

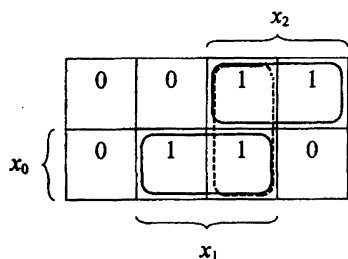


Рис. 6.14. Диаграмма Карно-Вейча для логической схемы, показанной на рис. 6.12; добавлен корректирующий вентиль (его действие отмечено штриховой линией).

Проблема возникает тогда, когда в DNF стоят два импликанта, из которых один импликант представляет собой переменную в форме с отрицанием, а другой импликант не имеет отрицательной формы, и при этом значения импликантов равны. Это имеет место в уравнении (6.11) при $x_1 = x_2 = 1$. Выход из затруднительного положения можно обеспечить путем введения импликанта, перекрывающего место соединения обоих импликантов.

6.4.2. Отрицательное воздействие на функционирование

«Функциональный риск» появляется, например, тогда, когда две входных переменных изменяются, а выходное состояние логической схемы должно оставаться в значении 1. Разъясним этот случай на примере, заданном с помощью диаграммы Вейча (рис. 6.15).

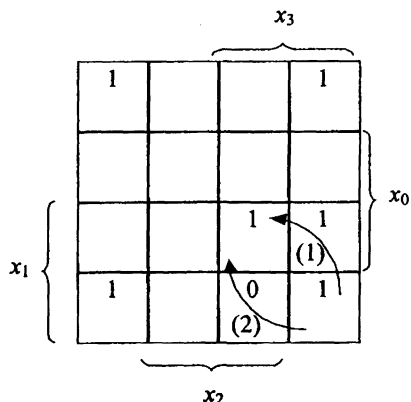


Рис. 6.15. Диаграмма Вейча для логической схемы, соответствующая случаю «функционального риска». Указаны оба возможных пути переключения.

При переходе от $(x_3, x_2, x_1, x_0) = (1, 0, 1, 0)$ к $(x_3, x_2, x_1, x_0) = (1, 1, 1, 1)$ переключение может произойти, в зависимости от величин времени задержки вентилей в соответствии с двумя вариантами. Если вначале проявляется действие x_0 , на выходе продолжительное время остается 1, что является правильным (путь 1 на рис. 6.15). Если вначале проявляется действие x_2 , возникает срыв (путь 2).

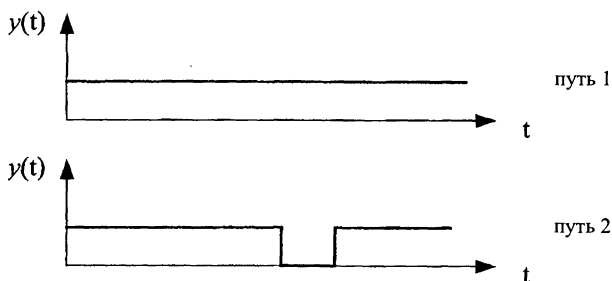


Рис. 6.16. Временные зависимости выходных сигналов логической схемы, соответствующей рис. 6.16, для двух возможных путей переключения.

6.4.3. Классификация отрицательных воздействий

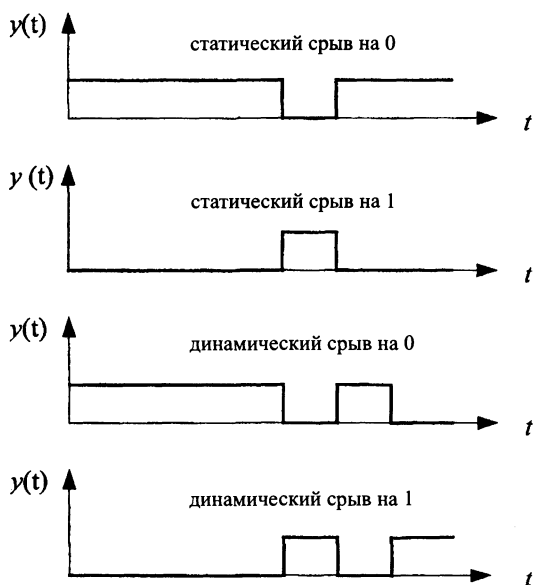


Рис. 6.17. Классификация отрицательных воздействий (рисков).

6.5. Упражнения

Задача 6.1. Пусть булева функция $f(x_3, x_2, x_1, x_0)$ задана таблицей истинности (табл. 6.8).

- Внесите значения функции в диаграмму Карно;
- Определите все первичные импликанты нормальной формы KDNF функции f ;
- Укажите основные первичные импликанты, абсолютно элиминируемые первичные импликанты и относительно элиминируемые первичные импликанты;
- Определите минимальную дизъюнктивную нормальную форму f ;
- Получите KDNF с помощью метода Квина-Мак-Класки.

Задача 6.2. Пусть не полностью заданная булева функцию определена через минтермы и макстермы. Не заданные значения относятся к виду *don't care*. Функция $f(x_4, x_3, x_2, x_1, x_0)$ имеет минтермы (x_4 : MSB, x_0 : LSB):

$$m_0, m_2, m_4, m_7, m_{16}, m_{21}, m_{24}, m_{25}, m_{28},$$

и макстермы:

$$M_1, M_9, M_{11}, M_{13}, M_{15}, M_{18}, M_{19}, M_{26}, M_{27}, M_{30}, M_{31}.$$



а) Нарисуйте диаграмму Карно и внесите в нее минтермы и макстермы.

б) Определите наиболее простые дизъюнктивные и наиболее простые конъюнктивные нормальные формы. При этом поля *don't care* должны быть использованы оптимально.

Таблица 6.8.

x_3	x_2	x_1	x_0	$f(x_3, x_2, x_1, x_0)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Задача 6.3. Через их минтермы m_i заданы три переключательные функции (x_3 : MSB, x_0 : LSB):

$$f_1(x_3, x_2, x_1, x_0) = m_0, m_4, m_5$$

$$f_2(x_3, x_2, x_1, x_0) = m_4, m_5, m_7$$

$$f_3(x_3, x_2, x_1, x_0) = m_3, m_5, m_7, m_{11}, m_{15}.$$

а) Задайте для каждой функции отдельно минимальную DNF, формируя при этом диаграмму Карно для каждой функции.

б) Покажите на основе трех диаграмм Карно, что три функции имеют общие термы и задайте наиболее простую логическую функцию, в которой общие термы реализуются только один раз.

в) Изобразите оптимальную логическую схему.

Задача 6.4. На рисунке показана цифровая схема, в которой может иметь место «структурный риск». Время задержки одного вентиля (И, ИЛИ, НЕ-ИЛИ) всегда равно t_0 .

- а) Задайте булеву функцию $y = f(x_3, x_2, x_1, x_0)$.
- б) Внесите функцию в диаграмму Карнр-Вейча.
- в) Промаркируйте на диаграмме Карно-Вейча позиции, для которых возможен риск срыва.
- г) Предложите схему с аналогичной функцией, в которой не проявляется «структурный риск».

ГЛАВА 7

АСИНХРОННЫЕ ТРИГГЕРЫ

Схему асинхронного триггера можно представить как схему, полученную из логической схемы, у которой, по крайней мере, один из выходов соединен со входом. В дальнейшем эта логическая схема обозначается как *SN1*. Триггеры называют также последовательностными схемами или конечными автоматами. Поведение триггера зависит как от значений входных переменных в данный момент времени, так и от входных переменных x_i , в предыдущие моменты времени. Поэтому он может хранить информацию. Хранящаяся информация называется параметрами состояния, здесь они обозначаются через z_i .

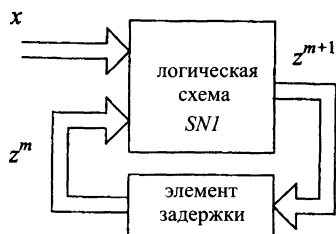


Рис. 7.1. Асинхронный триггер: логическая схема с обратной связью, использующая входной вектор X и вектор обратной связи Z , соответствующий моментам времени t и $t + 1$.

Для развязки входов и выходов асинхронных триггеров требуется введение элемента задержки в цепь обратной связи. Триггеры, в которых тактовый сигнал управляет развязанными буферными накопителями в цепи обратной связи, называют синхронными триггерами. Вследствие задержки между входом и выходом рациональным является рассмотрение параметров состояния в два различных момента времени обозначенных индексами t и $t + 1$. Рассматриваться должны только входные сигналы x_i , которые изменяют свои значения в дискретные моменты времени. Интервал между двумя изменениями входного сигнала должен быть настолько большим, чтобы в промежутке на всех соединительных линиях установились фиксированные значения сигналов. Это называют «работой в основном режиме».

7.1. Принципиальные особенности структуры триггеров.

В триггер всегда входит логическая схема $SN1$, которая имеет цепь обратной связи с элементом, вносящим задержку. Но триггер имеет также выходы, сигналы на которых могут быть выявлены двумя различными способами во второй логической схеме $SN2$ (рис. 7.2):

- В автомате Мура (Moore's Automaton) входные переменные y вычисляются только на основе параметров состояния Z^m
- В случае же автомата Мили (Mealy's Automaton), напротив, в составе логической схемы $SN2$ применяются не только параметры состояния Z^m , но также и входные переменные x , служащие входными величинами.

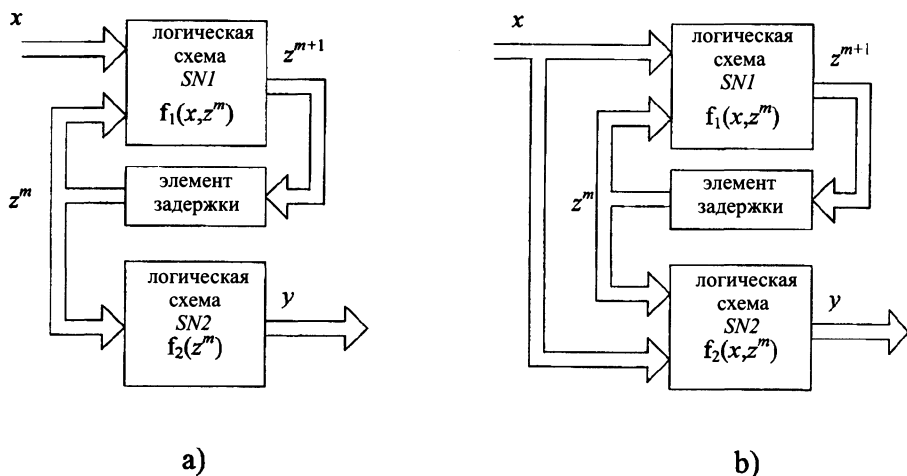


Рис. 7.2. а) Автомат Мура; б) Автомат Мили.

7.2. Анализ асинхронных триггеров.

В качестве примера проведен анализ триггера НЕ-ИЛИ (NOR). Он представляет собой идеальную логическую схему с обратной связью (рис. 7.3). Сокращения S и R , которыми обозначаются входные сигналы, означают «установка» (set) и «возврат» (reset). Здесь один выход обозначен через Q_1 , часто также обозначаемый как Q . Вторым выход Q_2 может быть также обозначен как инвертирующий выход $\neg Q$. Но при этом второму выходу инвертирующая функция придается не всегда, как это будет показано ниже.

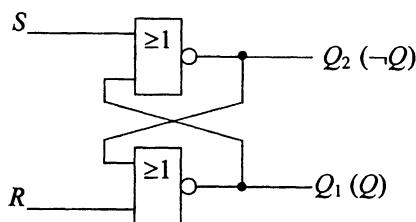


Рис. 7.3. Триггер NOR (в скобках — другие обычно применяемые обозначения выходов).

Разобраться в работе данной схемы можно на основе опыта, приобретенного ранее при анализе логических схем.

1. Начнем со случая $S = 1, R = 0$. В этом случае сигнал на выходе верхнего NOR-вентилia имеет значение $Q_2 = 0$. Сигналы на обоих входах нижнего NOR-вентилia имеют значения 0, так что $Q_1 = 1$. Установка триггера произведена. Внесем результат в таблицу истинности табл. 7.1, где показаны две возможные формы представления таблицы истинности.
2. В противоположном случае, когда $S = 0, R = 1$, вследствие симметрии устанавливаются значения на выходах $Q_1 = 0$ и $Q_2 = 1$. Триггер установлен в исходное состояние.
3. Теперь рассмотрим вариант, когда $S = 0, R = 0$. В этом случае поведение триггера будет определяться предыдущим состоянием.

Если имеет место выходной сигнал $Q_1 = 1$, то входной сигнал верхнего вентилia равен 1 и сохраняется $Q_2 = 0$. Сохраняется также $Q_1 = 1$, так как сигналы на обоих входах этого вентилia имеют значение 0. Данное состояние стабильно и поэтому удерживается.

Напротив, если на выходе имеем $Q_2 = 1$, из соображений симметрии получим, что удерживаются $Q_1 = 0$ и $Q_2 = 1$. Поэтому в табл. 7.1 вносится информация о том, что предыдущее состояние запоминается ($Q^m = Q^{m+1}$). Указанные в табл. 7.1 формы представления различаются значениями, которые имеют выходные сигналы Q и $\neg Q$ в моменты времени t и $t + 1$.

4. В соответствии с оставшимся вариантом $S = 1$ и $R = 1$. В этом случае оба выхода устанавливаются на 0. Этот вариант исключается, так как выходы не будут взаимно инверсными.

Таблица 7.1. Две формы таблицы истинности вентиля НЕ-ИЛИ на RS -триггерах.

S	R	Q^{m+1}	$\neg Q^{m+1}$
0	0	Q^m	$\neg Q^m$
0	1	0	1
1	0	1	0
1	1	—	—

S	R	Q^m	Q^{m+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	—
1	1	1	—

7.3. Систематический анализ.

Систематический анализ может быть проведен на основе составления булевых функций f_1 и f_2 (см. рис. 7.2) для логических схем $SN1$ и $SN2$. Перечертим структурную схему в соответствии с рис. 7.4 и введем суммарное время задержки t_1 .

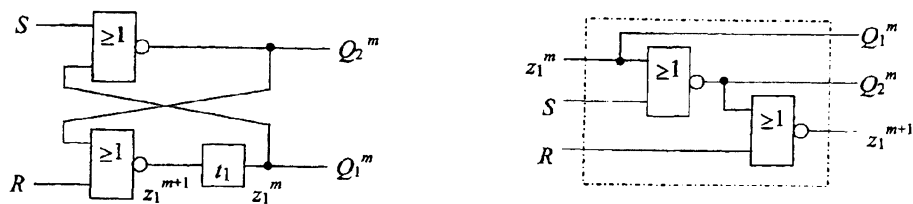


Рис. 7.4. а) Логическая схема с обратной связью (НЕ-ИЛИ вентиль на основе RS -триггера).

С помощью передаточной функции выходной сигнал идеальной логической схемы $SN1$ описывается как функция входных величин R , S , z_1^{m+1} :

$$z_1^{m+1} = \neg(\neg(S \vee z_1^m) \vee R) = \neg RS \vee \neg R z_1^m \quad (7.1)$$

Выходные функции описывают поведение логической схемы $SN2$, которая частично совпадает с логической схемой $SN1$, поскольку применен верхний вентиль НЕ-ИЛИ:

$$Q_1^m = z_1^m \quad (7.2)$$

$$Q_2^m = \neg(S \vee z_1^m) = \neg S \neg z_1^m \quad (7.3)$$

На основании этих уравнений могут быть составлены табл. 7.1. Но уравнения состояний могут быть также внесены в, так называемую, таблицу последовательности состояний (табл. 7.2).

Таблица 7.2. Диаграмма последовательности состояний в форме диаграммы Карно-Вейча.

		z_1^{m+1}				Q_1^m, Q_2^m			
z_1^m		$\neg S \neg R$	$S \neg R$	SR	$\neg SR$	$\neg S \neg R$	$S \neg R$	SR	$\neg SR$
0	⊙	⊙	1	⊙	⊙	01	00	00	01
1	⊙	⊙	⊙	0	0	10	10	10	10

В первую диаграмму вносится новое состояние Z_1^{m+1} . Эта величина влияет на изменение ее значения на выходе за счет обратной связи с входом. Во вторую диаграмму внесены входные величины, в данном случае Q_1^m и Q_2^m .

На следующем шаге в таблице последовательности состояний стабильные состояния обозначаются кружками. Они характеризуются уравнением $Z_1^m = Z_1^{m+1}$. В этих случаях система после установившегося выходного состояния удерживается в стабильном состоянии. В качестве примера можно рассмотреть входную комбинацию $R = 0$, $S = 1$ при одновременном соблюдении равенства $Z_1^{m+1} = 1$.

Другой формой отображения является диаграмма состояний на рис. 7.5. В диаграмме состояний внутренние состояния, в данном случае Z_1^{m+1} , обозначены кружками. Возможные переходы отмечены стрелками. Необходимые для них условия, определяемые входными переменными, отмечены на стрелках. Наклонной чертой отделены значения входных переменных. Из данной диаграммы, например, можно увидеть, что возможен переход от $Z_1^m = 0$ к $Z_1^m = 1$ при $R = 0$, $S = 1$, что так называемый, рефлексивный переход имеет место при $Z_1^m = 0$, при $R = S = 1$; состояние $Z_1^m = 1$ является рефлексивным для $\neg R$ независимо от S .

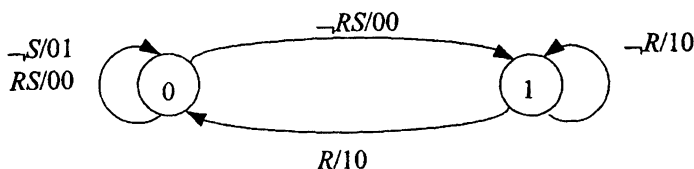


Рис. 7.5. Диаграмма состояний; в кружках стоят значения Z_1^m , после косой черты $Q_1 Q_2$.

Представленный в этом параграфе систематический анализ дает те же результаты, что и анализ в параграфе 7.2, проведенный на основе простых наблюдений.

7.4. Анализ с учетом задержки вентилей

Ниже будет показано, что вышеприведенный анализ слишком упрощен, поскольку он не полностью учитывает задержку сигнала в вентиллях. Этот анализ не отражает некоторые из возникающих проблем. Продемонстрируем на основе НЕ-ИЛИ триггера, как можно проанализировать работу схемы, в которой оба NOR-вентилля имеют конечное время задержки (рис. 7.6).

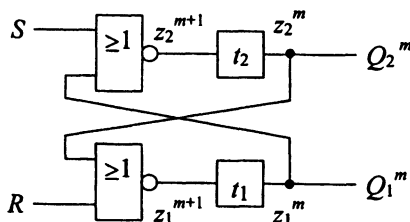


Рис. 7.6. Логическая схема с обратной связью (*RS*-NOR-триггер), учитывающая время задержки вентилей t_1 и t_2 .

Можно видеть, что теперь имеются две переменных, характеризующие состояния. Для них можно определить по рис. 7.6 передаточные функции:

$$z_1^{m+1} = \neg(R \vee z_2^m) = \neg R \neg z_2^m \quad (7.4)$$

$$z_2^{m+1} = \neg(S \vee z_1^m) = \neg S \neg z_1^m \quad (7.5)$$

и выходные функции:

$$Q_1^m = z_1^m \quad (7.6)$$

$$Q_2^m = z_2^m \quad (7.7)$$

Из уравнений состояния вновь можно получить KV-диаграмму. Сигналы Z_1^{m+1} , Z_2^{m+1} задаются как функции величин на входе R , S , Z_1^m , Z_2^m (табл. 7.3). Стабильные состояния отмечены кружками.

В тех случаях, когда $Z_1^{m+1} \neq Z_1^m$ или $Z_2^{m+1} \neq Z_2^m$, следует переход к другой комбинации входных сигналов. Эти нестабильные входные состояния выделены подчеркиванием.

Если для одной комбинации входных переменных подчеркнуты обе переменные, характеризующие состояния, то изменяются обе

величины. При этом говорят о двухкомпонентном переходе. В этом случае решающим моментом является соотношение задержек. Переход с более короткой задержкой вентиля первым воздействует на выход и вентиля это определяет следующее состояние.

Таблица 7.3. Таблица последовательностей состояний в форме KV-диаграммы.

$Z_2^{m+1} Z_1^{m+1}$				
$Z_2^m Z_1^m$	$\neg R \neg S$	$\neg RS$	RS	$R \neg S$
00	<u>11</u>	<u>01</u>	<u>00</u>	<u>10</u>
01	<u>01</u>	<u>01</u>	<u>00</u>	<u>00</u>
11	<u>00</u>	<u>00</u>	<u>00</u>	<u>00</u>
10	<u>10</u>	<u>00</u>	<u>00</u>	<u>10</u>

Теперь в соответствии с табл. 7.3 составим диаграмму состояний. Поясним это на примере. Допустим, что триггер с входными переменными $R = 1$, $S = 1$ находится в стабильном состоянии $Z_2^m Z_1^m = 00$. В табл. 7.3 находим в первой строке стабильное состояние $Z_2^{m+1} Z_1^{m+1} = 00$. Переключим входы в состояние $R = 0$, $S = 0$. Состояния, которые могут установиться, следует искать в первом столбце таблицы. Поскольку характеризующие состояние переменные $Z_2^m Z_1^m = 00$ вначале остаются без изменения, мы должны провести считывание в первой строке таблицы под новой комбинацией входных переменных новых переменных, характеризующих состояния. Находим $Z_2^{m+1} Z_1^{m+1} = 11$. Это означает, что обе, характеризующие состояние, переменные имеют тенденцию к изменению. При этом возможны три варианта:

1. Если время задержки у первого вентиля меньше ($t_1 < t_2$), то триггер переходит в стабильное состояние $Z_2^{m+1} Z_1^{m+1} = 01$.
2. Если $t_1 > t_2$, то триггер переходит в стабильное состояние $Z_2^{m+1} Z_1^{m+1} = 10$.
3. Если $t_1 = t_2$, то переход переходит в направлении $Z_2^{m+1} Z_1^{m+1} = 11$. Опять имеет место двухкомпонентный переход. Следовательно, триггер опять переключится обратно к $Z_2^{m+1} Z_1^{m+1} = 00$, после чего процесс будет повторяться периодически.

Таким образом, мы видим, что поведение триггеров зависит от времени задержки вентилей. Рассматриваемый процесс называют «гонка» или «гасе», причем в зависимости от того, различны или одинаковы конечные состояния, различают критичные и некритичные гонки. В рассматриваемом случае «гонка», переходящая при переключении от $RS = 11$ к $RS = 00$, представляет собой критичный

вариант, так как при этом триггер может вести себя тремя различными способами.

Вышесказанное поясняется на рис. 7.7 а. Переходы состояний при смене $= 11$ (в состоянии $Z_2^m Z_1^m = 00$) на $RS = 00$ дают конечный результат в соответствии с временами задержки вентилях. Это единственный случай, когда в RS -NOR-триггере имеет место «критичная гонка». Ее можно избежать, если исключить состояния 11.

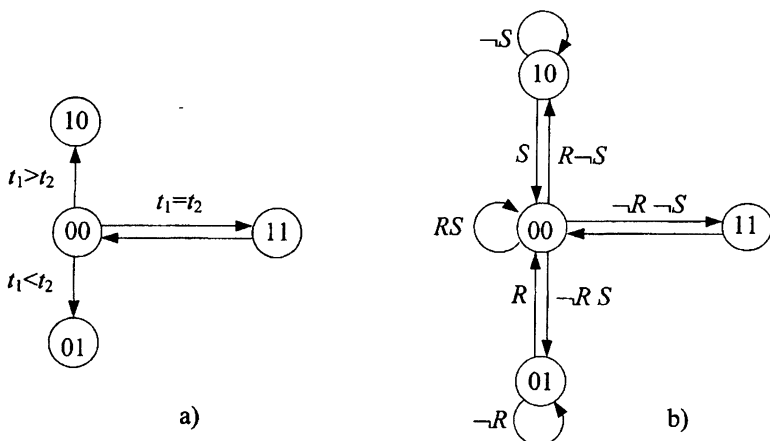


Рис. 7.7. а) Переходы состояний RS -NOR-триггера для случая замены $RS = 11$ (в состоянии $Z_2^m Z_1^m = 00$) на $RS = 00$, разделенные по направлениям в соответствии с временами задержек вентилях. б) Диаграмма состояний для всех переходов.

На рис. 7.7 б показана диаграмма состояний со всеми возможными переходами. Устанавливаем, что состояние 00 лежит в центре и поэтому его необходимо проходить при каждом переходе. Если запретить входную комбинацию $RS = 11$, то в этом случае триггер не сможет стабильно оставаться в состоянии 00 и тогда это состояние будет быстро проходить. Больше «критичных гонок» не имеется, поскольку приходят только однокомпонентные переходы.

Если триггер находится в состоянии 10, он стабилен при $S = 0$. В этом случае триггер возвращается в исходное состояние. Если затем изменить состояния входов на $SR = 10$, то триггер перейдет через состояние 00 в состояние 01, не оставаясь в состоянии 00.

Просуммировав выше сказанное, можно установить, что:

1. Состояние $R = S = 1$ у RS -NOR-триггера приводит к появлению некомплементарных выходных сигналов.
2. Если за запрещенным состоянием $RS = 11$ следует состояние $RS = 00$, то в результате в соответствии с рис. 7.7 может

иметь место поведение трех видов: колебания между 00 и 11, стабильность состояний 10 и 01. Это единственная «критичная гонка», которая может произойти в триггере. Ее можно избежать, если за запрещенным состоянием сразу не следует состояние $RS = 00$.

3. В асинхронных схемах возникают проблемы, если переходные функции не реализуются без рисков. Тогда на входах логической схемы $SN1$ могут появиться короткие помехи, приводящие к ложному или непредсказуемому поведению триггеров.
4. Двухкомпонентных переходов переменных, характеризующих состояние, следует по возможности избегать, поскольку существует возможность появления «гонки».

7.5. Элементы ЗУ

В последнем параграфе было показано, что применение асинхронных триггеров проблематично. Поэтому применяются только те схемы, поведение которых хорошо известно. К ним относятся триггеры (flip-flop, сокращенно FF), которые кратко описаны в данном параграфе. Все схемы можно реализовать в интегральном виде.

7.5.1. RS -триггер

RS -триггер может быть сконструирован из NAND- или NOR-вентилей. RS -триггер с NAND-вентильями можно также рассматривать как работающий с отрицательной логикой.

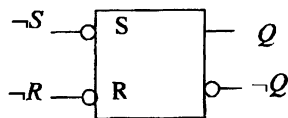
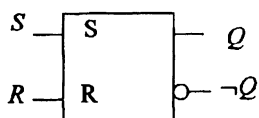
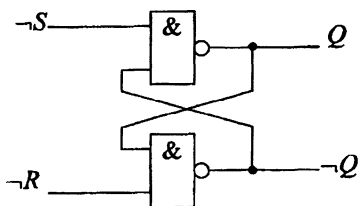
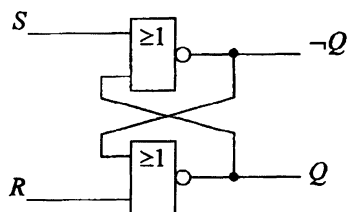
Проблематичным для RS -триггеров является то, что появление даже коротких импульсов помехи на входах R и S ведет к ошибочным установке и возврату в исходное состояние триггера. Поэтому при работе с RS -триггерами, управляемыми тактовым уровнем, применяют дополнительно тактовый сигнал, ограничивающий время, в течение которого входы активны.

7.5.2. RS -триггер с тактовым входом

RS -триггер с тактовым входом обозначают также как RS -защелку (RS-latch) или триггер в функции «промежуточное ЗУ» (рис. 7.9). Хотя этот триггер составлен из NAND-вентилей, он работает с положительной логикой. Передаточную функцию для управляемого состояниями RS -триггера можно определить из структурной схемы 7.9.

$$\begin{aligned} Q^{m+1} &= \neg(\neg(SC)(\neg(\neg(RC)Q^m))) = SC \vee (\neg(RC)Q^m) = \\ &= SC \vee RQ^m \vee \neg CQ^m \end{aligned} \quad (7.8)$$

Передаточная функция свидетельствует о том, что триггер устанавливается в следующее состояние, когда $S \wedge C = 1$ или когда предыдущим состоянием было $Q^m = 1$ и $\neg R = 1$, или $\neg C = 1$. Из передаточной функции можно составить таблицу состояний.



S	R	Q^{m+1}	$\neg Q^{m+1}$
0	0	Q^m	$\neg Q^m$
1	0	1	0
0	1	0	1
1	1	—	—

$\neg S$	$\neg R$	Q^{m+1}	$\neg Q^{m+1}$
0	0	—	—
1	0	0	1
0	1	1	0
1	1	Q^m	$\neg Q^m$

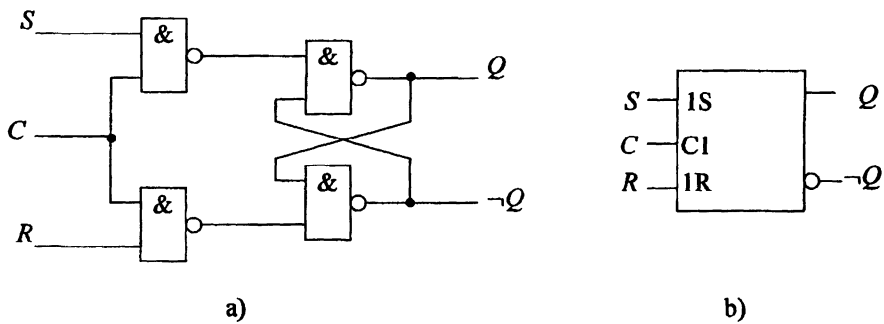
$$Q^{m+1} = \neg(R \vee \neg(S \vee Q^m)) = \neg RS \vee \neg RQ^m \quad Q^{m+1} = \neg(\neg S \neg(\neg RQ^m)) = S \vee \neg RQ^m$$

Рис. 7.8. RS -триггер с NOR- или NAND-вентильми. Сверху вниз: структурная схема, логический символ, таблица истинности и передаточная функция.

Из рис. 7.9с следует, что в этом триггере нельзя избежать запрещенного состояния. Теперь, правда, можно избежать помех на входах S и R в интервале времени, в котором тактовый сигнал $C = 0$. Триггер может быть запущен только при положительном тактовом импульсе. Этот вид управления называют управлением уровнем тактового импульса или управлением состоянием.

На рис. 7.10 показаны временные характеристики RS -триггера. Можно видеть, что триггер в течение интервала времени, в течение которого тактовый сигнал находится на уровне 1, пропускает все входные сигналы. Таким образом импульсы помехи могут произвести установку триггера и его возврат, даже при наличии фикс-

сированных значений входных величин. На рис. 7.10 указаны такие величины времени задержки сигналов при переходах от высокого уровня к низкому уровню и наоборот. Все рассматриваемые в дальнейшем триггеры имеют тактовый вход.



c)

S	R	C	Q^{m+1}	$\neg Q^{m+1}$
0	0	1	Q^m	$\neg Q^m$
1	0	1	1	0
0	1	1	0	1
1	1	1		
d	d	0	Q^m	$\neg Q^m$

Рис. 7.9. RS-триггер с тактовым входом. а) Структурная схема; б) Логический символ; в) Таблица истинности.

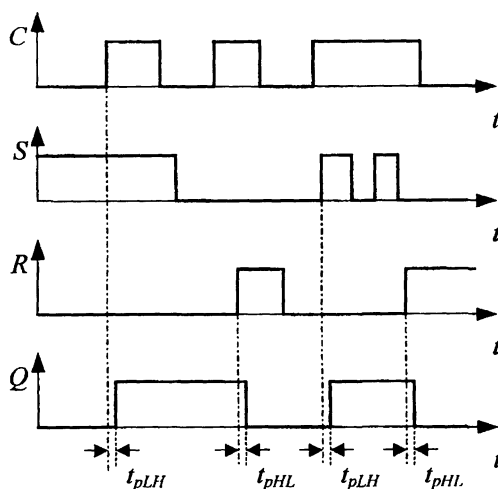


Рис. 7.10. Типичные характеристики управляемого тактовым уровнем RS-триггера; C — тактовый сигнал.

7.5.3. D-триггер

Как правило, для того, чтобы обойти недостаток RS -триггера, заключающийся в наличии запрещенного входного состояния, применяют другие триггеры. Важнейшим является D -триггер (рис. 7.11). D -триггер образован RS -триггером с управлением тактовым уровнем, в котором для нового входа (D) установлено, что $D = S = \neg R$. Благодаря этому исчезает запрещенное состояние. Передаточную функцию можно получить, введя в уравнение (7.8) равенство $D = S = \neg R$. Получаем:

$$Q^{m+1} = DC \vee \neg CQ^m \quad (7.9)$$

Передаточная функция D -триггера говорит о том, что при $C = 1$ вход для ввода данных переключается, а при $C = 0$ запоминается старое состояние.

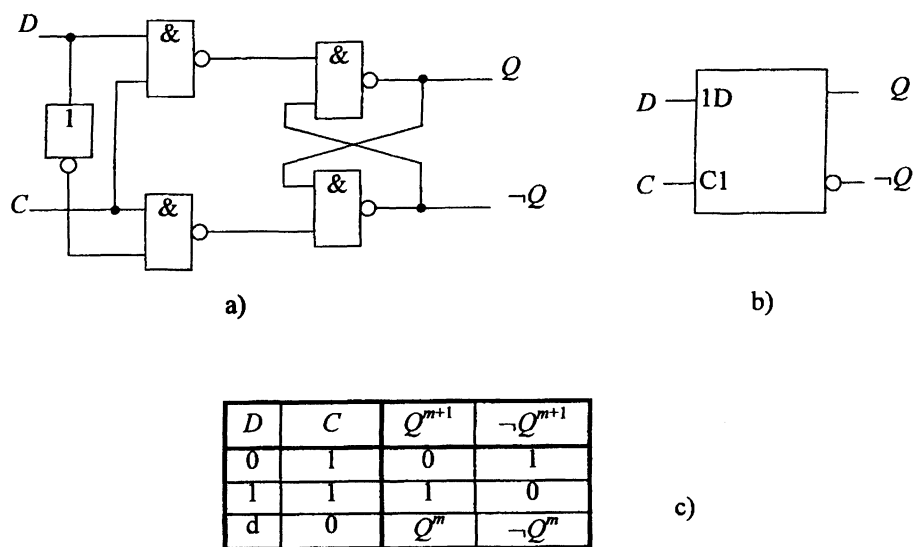


Рис. 7.11. D -триггер. а) структурная схема, б) логический символ, в) таблица истинности.

На рис. 7.12 показаны типичные временные характеристики D -триггеров. Мы видим, что во время высокого уровня тактового сигнала D -триггер воспроизводит изменение входных сигналов. В этом состоянии D -триггер представляет собой элемент задержки (D от английского delay). Подобная задержка затрудняет передачу сигналов особенно в сдвиговых регистрах. Там применяют триггеры с предварительными накопителями, в которых вновь запомнен-

ное состояние проявляется только при появлении на входе отрицательного фронта тактового импульса.

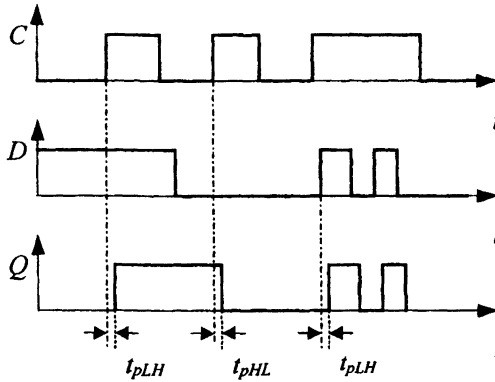


Рис. 7.12. Типичные временные характеристики управляемых тактовым уровнем D-триггеров.

Решение о том, какая информация запоминается в безбуферном D-триггере, выносится в конце фазы High (т.е. в конце импульса с высоким уровнем) в тактовом интервале t_w (рис. 7.13). Выходной сигнал будет определенным в том случае, если информационный сигнал является постоянным некоторое время до и после заднего фронта тактового импульса. Эти интервалы времени называют временем установки (setup) t_s и временем удержания (hold) t_h . Если же входной сигнал в данные отрезки времени не является постоянным, запомненная величина не является определенной. Выходной сигнал может измениться в течение интервала переброса t_k . В безбуферном триггере этот интервал перекрывается с действующим интервалом.

7.5.4. D-триггер с управлением по переднему фронту импульса

Для того, чтобы обойти недостаток управляемых состоянием D-триггеров, который заключается во влиянии помех и изменений входного сигнала на записанную информацию во время тактового импульса, применяют D-триггеры с управлением по фронтам. В идеальном случае у управляемого фронтом импульса триггера входной сигнал должен оставаться неизменным только в течение интервала времени, соответствующего фронту тактового импульса. У этих триггеров не имеет никакого значения длина фазы тактирования. Разумеется, фронт тактового импульса должен иметь наименьшую крутизну.

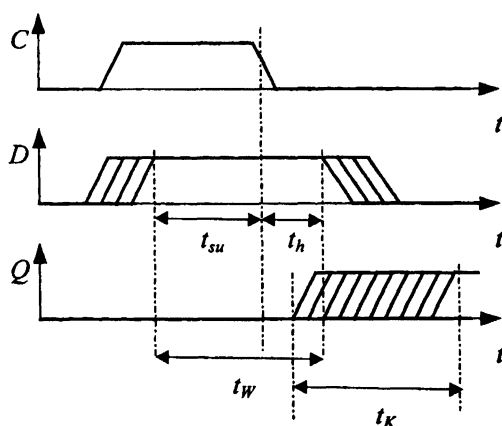


Рис. 7.13. Определение времени установки (setup) t_s и времени удержания t_h , действующего интервала t_w и интервалов переброса t_k

Соответствующая указанным требованиям схема показана на рис. 7.14. На ней приведен управляемый передним фронтом D-триггер, взятый из ТТЛ-версии интегральной схемы 7474. С целью проведения анализа разделим его в указанных местах. Тогда он будет свободен от воздействия обратных связей.

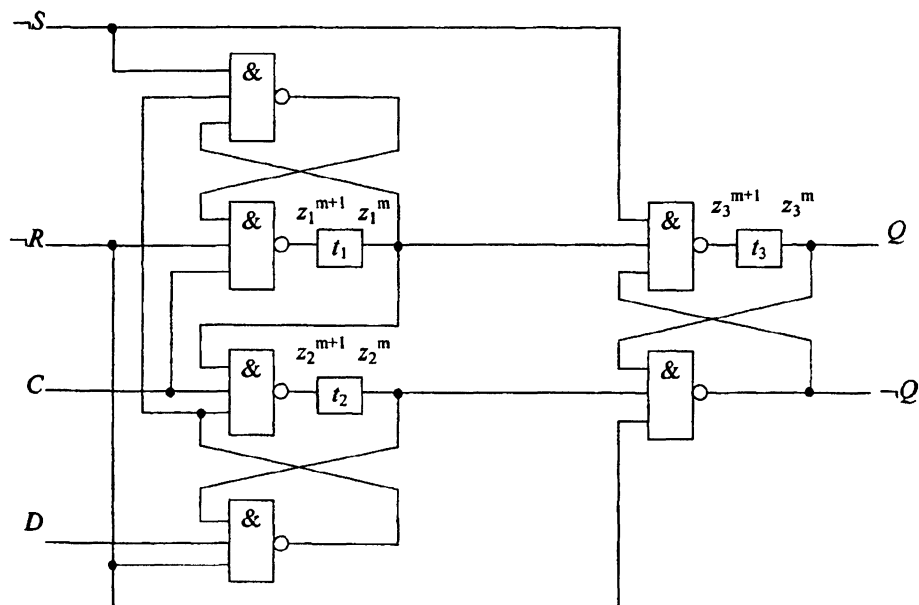


Рис. 7.14. Структурная схема D-триггера с управлением по переднему фронту (7474). В ветви схемы введены элементы задержки.

Для простоты анализа установим входы $\neq R$ и $\neg S$ на 1. Через эти входы триггер независимо от тактового импульса может быть установлен в противоположное или возвращен в исходное состояние. По рис. 7.14 можно определить следующие переходные функции:

$$z_1^{m+1} = \neg (C (\neg (z_1^{m+1} \neg (D z_2^m)))) = \neg C \vee z_1^m \neg D \vee z_1^m \neg z_2^m \quad (7.10)$$

$$z_1^{m+1} = \neg (\neg (D z_2^m) C z_1^m) = \neg C \vee z_2^m D \vee z_1^m \quad (7.11)$$

$$z_3^{m+1} = \neg (z_1^m \neg (z_2^m z_3^m)) = \neg z_1^m \vee z_2^m z_3^m \quad (7.12)$$

Уравнения состояния свободны от схемных рисков, поскольку не одно из уравнений не содержит переменной в форме с отрицанием и в форме без отрицания. Выходная функция имеет вид:

$$Q = z_3^m \quad (7.13)$$

Таблица 7.4. Таблица последовательности состояний управляемого по переднему фронту D-триггера 7474 для Z_1^{m+1} , Z_2^{m+1} , Z_3^{m+1} .

$Z_1^m Z_2^m Z_3^m$	$\neg C \neg D$	$\neg C D$	$C D$	$C \neg D$
000	111	111	011	011
001	111	111	011	011
010	111	111	011	011
011	111	111	011	011
101	110	110	100	100
101	110	110	100	100
110	110	110	010	100
111	111	111	011	101

На основе переходных уравнений (7.10)–(7.12) можно сформировать таблицу последовательностей состояний (табл. 7.4). Таблица последовательностей состояний показывает, что при нормальной работе может быть только 6 состояний. В то же время состояния 000 и 001 могут наступить только при включении на короткое время. Они не могут быть следствием какого-либо другого состояния. Стабильные состояния отмечены сплошной линией.

Теперь с помощью диаграммы состояний, показанной на рис. 7.15, исследуем поведение управляемого передним фронтом D-триггера. При этом состояния 001 и 000 не учитываются, поскольку они наступают только при включении. Поскольку рассматривается бистабильная схема, то когда тактовый сигнал $C = 0$, могут суще-

становать оба стабильных состояния:

Установлено: $Q = z_3^m = 1$
 $z_1^m z_2^m z_3^m = 111$

или

Сброшено: $Q = z_3^m = 0$
 $z_1^m z_2^m z_3^m = 110$

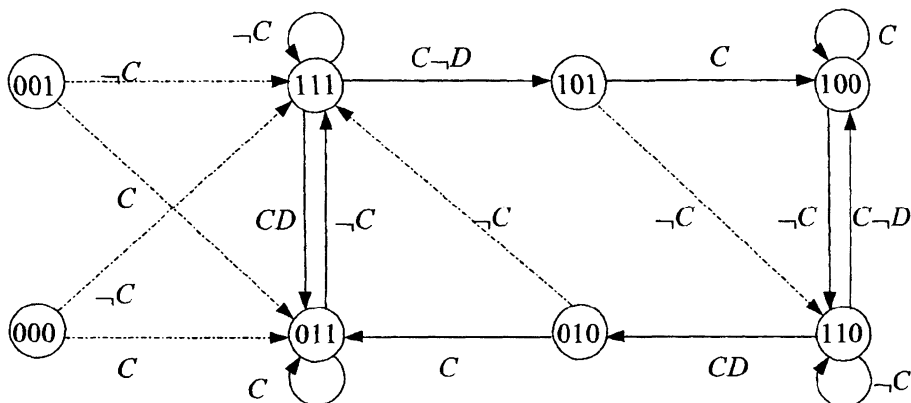


Рис. 7.15. Диаграмма состояний D-триггера 7474. Отмеченные штриховыми линиями переходы не требуются для первоначального понимания. При нормальной работе по ним переходы не происходят.

Для анализа примем, например, что триггер установлен в исходное состояние и что тактовый сигнал $C = 0$. Тогда триггер будет в состоянии 110. При нарастающем фронте на входе тактового сигнала C происходит следующее:

1. Если $D = 1$, тогда триггер переходит в состояние 010 и затем без воздействия D в состояние 011. Поскольку вход D не оказывает влияния, несущественно, изменился ли на нем сигнал к этому моменту времени или нет. Это означает, что входной сигнал D играет роль только во время нарастающего фронта тактового импульса. При отрицательном фронте тактового сигнала триггер переходит в состояние 111. Теперь, поскольку справедливо $Q = Z_3^m$, выход находится в состоянии 1. Триггер установлен. Выходной сигнал изменяет свое значение только после окончания действующего интервала.
2. Если $D = 0$, происходит переход в состояние 100. На выходе остается 0. В этом состоянии вход D заперт, так как остается независимым от D . При отрицательном фронте тактового сигнала происходит переход в состояние 110. Триггер остается в исходном состоянии.

Аналогичный процесс происходит тогда, когда в состоянии 111 появляется нарастающий фронт тактового сигнала.

Следовательно, оба триггера, образуемые вентилями 1–4, задача, заключающаяся в запираании входа, как только тактовый импульс принимает значение 1. Поэтому триггер реагирует только при нарастающем фронте на входе D (рис. 7.16).

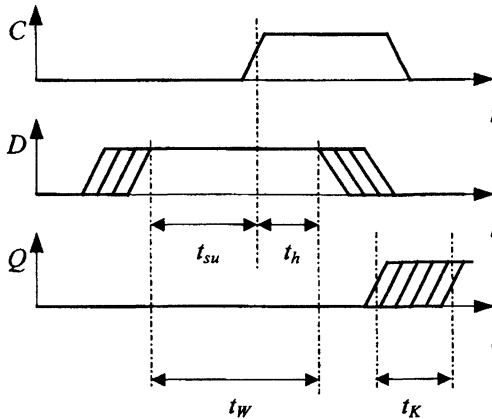


Рис. 7.16. Временные диаграммы управляемого передним фронтом D-триггера.

На временной диаграмме 7.16 показано, что выходной сигнал изменяет свое значение только после действующего интервала (состоящего из лежащего перед тактовым фронтом интервала t_{su} и лежащего после него t_h), как это уже было установлено при обсуждении диаграммы состояний.

Действующий интервал короче, чем у управляемого тактовым уровнем D-триггера, и лежит около нарастающего фронта. Отрезок времени, в течение которого выходной сигнал изменяет свое значение, называют интервалом переброса. Это поведение триггера, отличающееся отсутствием наложения действующего интервала, возникает как следствие буферизации. Поэтому этот триггер называют буферизированный, управляемый передним фронтом D-триггер.

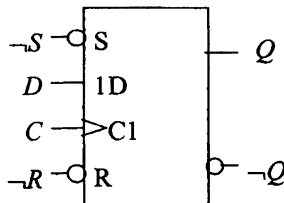


Рис. 7.17. Логический символ управляемого передним фронтом D-триггера 7474 с асинхронной установкой и возвратом.

Таблица 7.5. Таблица истинности для управляемого передним фронтом D-триггера 7474 с асинхронными установками и возвратом (состояние неопределенно). Нарастающий фронт тактового сигнала обозначен $\uparrow$.

D	C	$\neg S$	$\neg R$	Q^{m+1}
d	d	0	1	1
d	d	1	0	0
d	d	0	0	*
1	$\uparrow$	1	1	1
0	$\uparrow$	1	1	0
d	0	1	1	Q^m
d	1	1	1	Q^m

В таблице истинности управляемого передним фронтом D-триггера (табл. 7.5) нарастающие тактовые фронты обозначены стрелками. В логическом символе (рис. 7.17) управление передним фронтом представлено треугольником на тактовом входе.

7.5.5. Управление передним и задним фронтом

В тех случаях, когда в состоящей из D-триггеров цепочке сдвигового регистра данные должны передаваться дальше, как в «пожарной цепочке» ПЗС, вход и выход применяемых триггеров никогда не должны соединяться непосредственно. Поэтому, для этого случая применения подходят только такие типы триггеров, у которых достаточно далеко друг от друга лежат действующий интервал и интервал переброса. Для этой цели могут быть применены, например, упомянутые выше управляемые передним фронтом D-триггеры буферизированного типа. До тех пор, пока не произойдет фазовый сдвиг тактовых импульсов (clock-skew), информация будет правильно передаваться дальше, так как действующий интервал и интервал переброса не перекрываются. Но поскольку оба интервала разнесены на малое расстояние, при фазовых сдвигах тактового сигнала, которые всегда происходят в реальных схемах, могут иметь место сбои в передаче информации.

Для данной цели более предпочтительными являются триггеры с управлением двумя фронтами, которые передают информацию с выхода только с появлением падающего фронта тактового импульса на выходе (рис. 7.19). Триггеры этого вида называют master-slave-триггерами. Можно считать, что D-триггер с управлением двумя фронтами составлен из двух D-триггеров с управлением передним

фронтом, у которых второй триггер запускается инверсией тактового сигнала, соответствующего спадающему фронту.

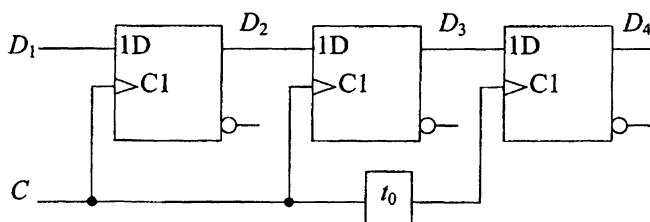


Рис. 7.18. Сдвиговый регистр с управляемыми передним фронтом D-триггерами.

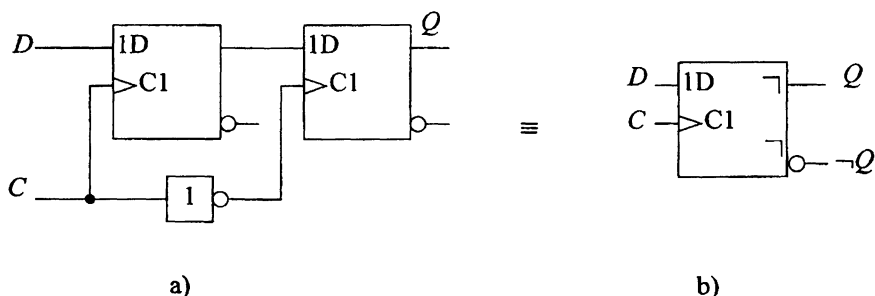


Рис. 7.19. D-триггер с управлением двумя фронтами: а) принцип; б) логический символ.

7.5.6. J-K-триггер

JK-триггер может быть образован на основе RS-триггера типа master-slave путем подачи обратной связи с выходов Q и $\neg Q$ на входы R и S при высоком уровне тактового сигнала в соответствии со следующими формулами:

$$S = CJ - Q^m \quad (7.14)$$

$$R = CKQ^m \quad (7.15)$$

Из формул следует, что установку триггера можно производить только после его возвращения в исходное состояние. Поскольку R и S не могут быть одновременно равны 1, так как справедливо либо $Q = 1$ либо $\neg Q = 1$.

Переходную функцию JK-триггер можно получить на основе переходной функции RS-триггера при условии применения уравнений (7.14) и (7.15):

$$Q^{m+1} = S \vee \neg RQ^m \quad (7.16)$$

$$Q^{m+1} = CJ\bar{Q}^m \vee \neg(CKQ^m)Q^m \quad (7.17)$$

$$Q^{m+1} = CJQ^m \vee \neg KQ^m \vee \neg CQ^m \quad (7.18)$$

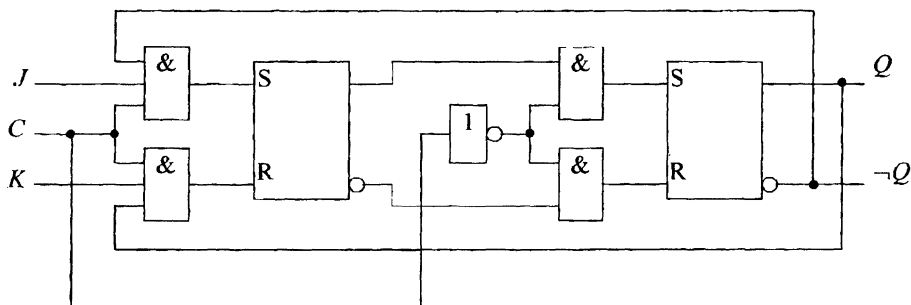
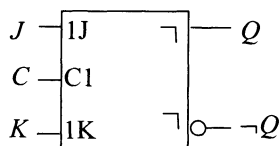


Рис. 7.20. JK-триггер.

Из переходной функции получаем таблицу последовательности состояний, представленную на рис. 7.21.

C	J	K	Q^{m+1}
1	0	0	Q^m
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}^m$
0	d	d	Q^m

а)



б)

Рис. 7.21. а) Таблица истинности; б) логический символ JK-триггера.

Следовательно, JK-триггер ведет себя как RS-master-slave-триггер, пока сигналы J и K не будут одновременно равны 1. Но если $J = K = 1$, выходной сигнал изменяется при каждом тактовом импульсе. Это упрощает конструирование делителей частоты и цифровых счетчиков на основе JK-триггеров.

7.5.7. Т-триггер.

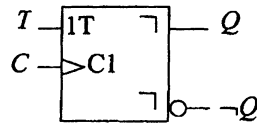
Т-триггер (рис. 7.22) образуется из JK-триггера, введением нового входа T , связанного с обоими входами JK-триггера: $T = J = K = 1$. По-английски его называют toggle-flip-flop.

7.5.8. Пример

Типичным примером интегрального триггера является D-триггер 74175. Эта микросхема содержит 4 одинаковых управляемых передним фронтом D-триггера, синхронизируемых от одного источника тактовых сигналов. Все триггеры подключены к одному и тому же входу возврата в исходное положение R .

T	C	Q^{n+1}
0	0	Q^n
0	$\uparrow$	Q^n
1	$\uparrow$	$\neg Q^n$

а)



б)

Рис. 7.22. а) Таблица истинности и б) логический символ управляемого передним фронтом Т-триггера.

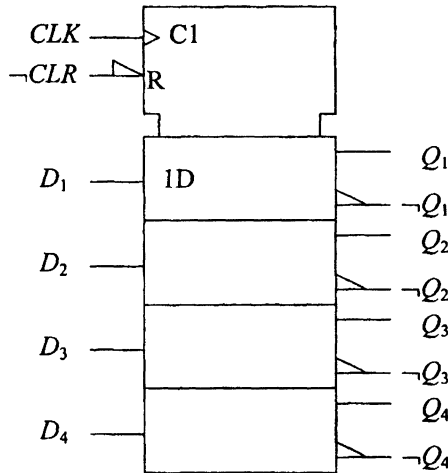


Рис. 7.23. Логический символ микросхемы 74175, содержащей 4 D-триггера с управлением передним фронтом.

7.5.9. Общее о триггерах.

В табл. 7.6 собраны наиболее важные виды триггеров. Отметим, что некоторые из триггеров не применяются. Так, например, Т-триггер без тактового входа не стабилен, а D-триггер вырождается до простого сквозного соединения.

Все указанные в табл. 7.6 триггеры могут иметь дополнительные асинхронные входы установки и возврата в исходное состояние.

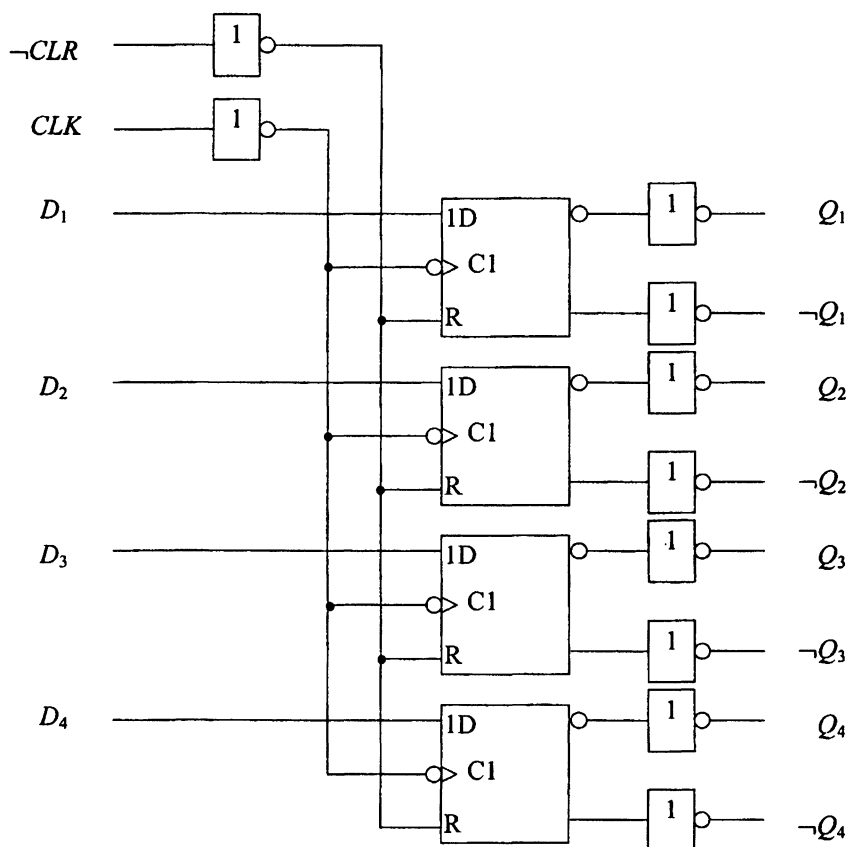


Рис. 7.24. Структурная схема микросхемы 74175, содержащей 4 D-триггера с управлением передним фронтом.

В табл. 7.7 показано положение интервала переброса и действующего интервала у различных типов триггеров относительно положения тактового фронта.

- При отсутствии управления тактовым сигналом действующий интервал и интервал переброса длительное время активны. Это имеет смысл только для RS-триггера.
- При тактированном управлении состоянием положение действующего интервала строго установлено. Он привязан к высокому, или скорее к отрицательному, тактовому уровню. Интервал переброса перекрывается с действующим интервалом.
- При управлении двумя состояниями положение действующего интервала дополнительно контролируется тактовым сиг-

налом. Действующий интервал и интервал переброса следуют вплотную друг за другом.

Таблица 7.6. Таблица важнейших триггеров.

	без тактового управления	управление состоянием	управление с помощью двух состояний	управление одним фронтом	управление двумя фронтами
RS					
D	задержка				
JK	нестабильный	нестабильный			
T	нестабильный	нестабильный			

- При управлении двумя фронтами действующий интервал и интервал переброса прилегают соответственно к положительному и отрицательному фронтам тактового сигнала. Важным моментом при управлении двумя фронтами является возможность смещения действующего интервала и интервала переброса за счет выбора коэффициента заполнения тактовых импульсов, то есть отношения длительности высокого уровня к длительности низкого уровня. Действующий интервал и интервал переброса не перекрываются.
- буферизация означает, что действующий интервал и интервал переброса не перекрываются. Как правило, расстояние соста-

влияет приблизительно длину задержки вентиля. Часто буферизация связана с управлением одним фронтом, как это было показано в главе 7.5.4 для D-триггера, выполненного с помощью ТТЛ-техники.

Таблица 7.7. Таблица с важнейшими триггерами.

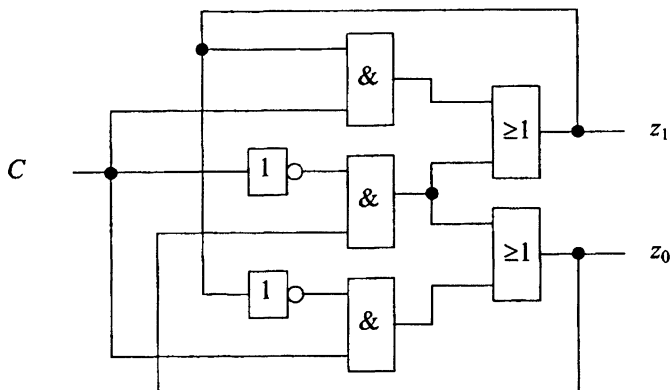
	без тактового управления	управление состоянием	управление с помощью двух состояний	управление одним фронтом	управление двумя фронтами

7.6. Упражнения

Задача 7.1. На приведенном рисунке показан асинхронный драйвер с двумя цепями обратной связи.

Проанализируем схему.

- Сформируйте таблицу состояний и внесите в нее все стабильные состояния.
- Нарисуйте диаграмму состояний.
- Какие проблемы могут возникнуть в схеме? Предложите улучшение схемы, которые помогут устранить эти проблемы.
- Опишите функции схемы.



Задача 7.2. Необходимо спроектировать асинхронную схему с тактовым входом C , сигнальным входом E и выходом Q .

При выполнении условия, согласно которому при нарастающем фронте тактового сигнала входной сигнал E имеет значение 1, схема должна выдавать выходной импульс одинаковой длительности с тактовым импульсом. Обстоятельства, связанные с этим моментом, поясняются на проведенном ниже рисунке.

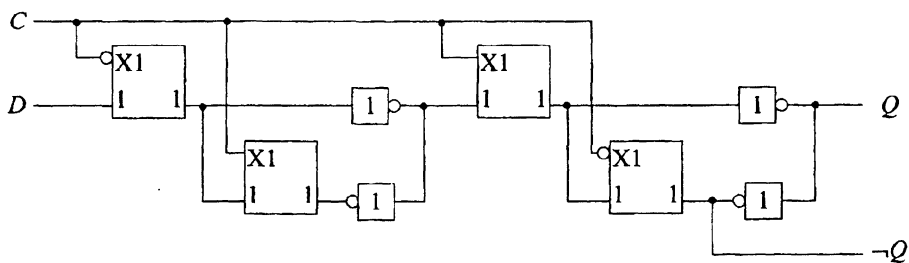
а) Составьте диаграмму состояний для реализации этой функций. Вам требуется для этого 3 состояния. Четвертое (избыточное) состояние должно быть при $C = 1$ стабильным. При $C = 0$ эта управляющая схема (драйвер) должна переходить в нормальный режим.

б) Сформируйте таблицу состояний для этой схемы. Закодируйте состояние так, чтобы происходили в основном однокомпонентные переходы. Попытайтесь одну из величин, характеризующих состояние, использовать непосредственно как выходной сигнал.

в) Разработайте схему, свободную от рисков.

Задача 7.3. Переделайте временную диаграмму, показанную на рисунке 7.12, для управляемого передним фронтом D-триггера.

Задача 7.4. На рисунке показана обычная реализация управляемого фронтом тактового импульса D-триггера на основе КМОП-технологии. Ниже проанализируем этот триггер.



В этой схеме пары проходных вентилях могут быть преобразованы в И-ИЛИ-схему в соответствии со следующим примером (направление прохождения сигнала только слева направо).

Действуйте при анализе следующим образом:

а) Определите, сколько транзисторов нужно для схемы, если исходить из того, что тактовый сигнал также должен быть инвертирован.

б) Начертите общую схему, учитывая заданные преобразования. Отметьте места, в которых схему следует разделить, чтобы она была свободна от обратной связи.

- в) Задайте переходную функцию (функции) и выходную функцию (функции).
- г) Сформируйте таблицу состояний.
- д) Начертите диаграмму состояний. На базе диаграммы состояний поясните функции схемы.

ГЛАВА 8

СИНХРОННЫЕ ДРАЙВЕРЫ

Синхронный драйвер (управляющая схема, запускающая схема, возбудитель, автомат) может быть образован из асинхронной логической схемы, путем встраивания в цепь обратной связи буферных запоминающих блоков, управляемые тактовым сигналом CLK.

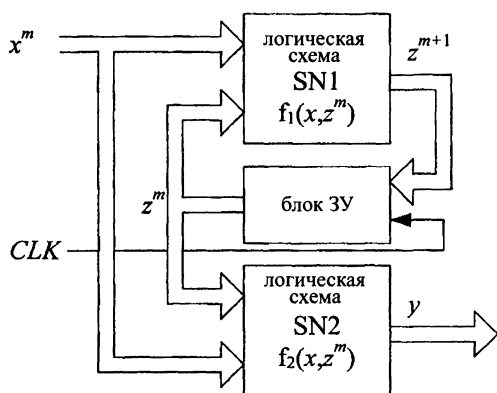


Рис. 8.1. Синхронный драйвер Мили (Mealy) с тактовым входом CLK.

Преимущество синхронного драйвера заключается в том, что выходной сигнал логической схемы влияет на поведение драйвера только в состоянии переходного процесса. Поэтому здесь не является существенным риск ложного срабатывания. Кроме того, не могут возникнуть «гонки» (races) при переключении, так как вход и выход логической схемы развязаны благодаря введению запоминающих элементов.

8.1. Синтез драйверов (пример I)

Рассмотрим пример разработки синхронного драйвера.

8.1.1. Постановка задачи

Пусть необходимо спроектировать управляющую схему (драйвер) с четырьмя состояниями, включающую и выключающую три электро-

механических устройства («машины»). Драйвер должен быть сконструирован на основе JK-триггеров. Поведение должно зависеть входного сигнала τ . При $\tau^m = 0$ должны периодически пробегаться 4 состояния, электромеханические устройства должны включаться в четыре возможных состояния в соответствии с табл. 8.1.

При величине входного сигнала $\tau^m = 1$ драйвер должен перейти в состояние 1. Драйвер должен оставаться в этом состоянии до тех пор, пока $\tau = 1$. Электромеханические устройства должны выключаться возможно быстрее (τ^m является сигналом аварийного выключателя).

Таблица 8.1. Управление электромеханическими устройствами («машинами») M_1, M_2, M_3 в четырех состояниях ($\tau^m = 0$).

	Машина 1 M_1	Машина 2 M_2	Машина 3 M_3
Состояние 1	включено	включено	включено
Состояние 2	выключено	включено	включено
Состояние 3	выключено	включено	выключено
Состояние 4	включено	включено	выключено

8.1.2. Формирование диаграммы состояний

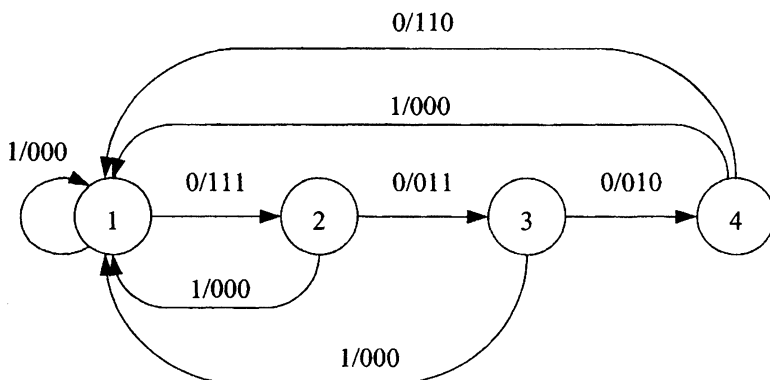


Рис. 8.2. Диаграмма состояний при управлении электромеханическими устройствами (индексация: $\tau^m / M_1 M_2 M_3$).

Теперь можно построить диаграмму состояний, исходя из состояния 1. При $\tau^m = 0$ драйвер пробегает все состояния по порядку. В соответствии с таблицей значения трех выходных сигналов отделяется наклонной чертой от τ^m . Если $\tau^m = 1$, логическая схема

переходит в состояние 1 и остается в нем пока $\tau^m = 1$. Три выходных сигнала остаются в состояниях 000.

Как следует из диаграммы состояний:

1. Речь идет о автомате Мили (Mealy), поскольку величины выходных переменных зависят от величины входной переменной τ^m . В диаграмме состояний это четко показано с помощью двух различных путей для значений τ^m 0 и 1, которые с различными выходными величинами ведут из состояния 4 в состояние 1.
2. В состоянии 1 имеет место так называемое рефлексивное состояние. Это состояние удерживается пока $\tau^m = 1$.

8.1.3. Структура схемы управления электромеханическими устройствами

Структура принципиальной схемы машинного управления представляет собой автомат Мили с входным сигналом τ^m и с выходным сигналом M шириной в 3 бита (рис. 8.3). Поскольку необходимо прогнать всего 4 состояния, можно обойтись двумя JK-триггерами.

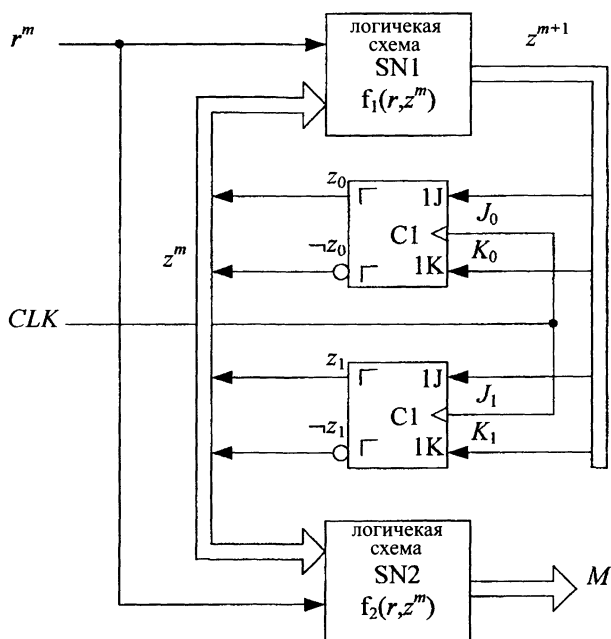


Рис. 8.3. Структура проектируемого драйвера.

Теперь необходимо установить соответствие между значениями логической переменной, хранящимися в JK-триггерах, и 4 состояниями. Можно выбрать произвольную последовательность кодовых слов. В частности нет необходимости быть привязанным к однокомпонентным переходам, так как благодаря применению триггеров в цепи обратной связи двухкомпонентные переходы проблемы не представляют. Сигналы на входах триггеров считываются только после их стабилизации тактовым сигналом. Но, зачастую, кодирование состояний с помощью однокомпонентных переходов ведет к более простой схеме.

Выбираем для кодирования хранящиеся в обоих JK-триггерах значения z_i , так как это показано в табл. 8.2. При определенных обстоятельствах какое-либо другое кодирование может привести к более простой схеме.

Таблица 8.2. Кодирование состояний.

состояние	z_1	z_2
1	0	0
2	0	1
3	1	1
4	1	0

Для реализации схемы необходимо спроектировать драйверы SN1 и SN2. С целью получения более простой схемы могут быть также использованы инвертирующие выходы триггеров.

8.1.4. Формирование таблицы последовательности состояний

Таблицу последовательности состояний можно получить путем считывания данных из диаграммы состояний, показанной на рис. 8.2. Действовать можно в соответствии с порядком, который был применен для асинхронных схем. У синхронных драйверов отдельные состояния, имеющие индексы m и $m + 1$, различаются на один тактовый период. В таблицу (табл. 8.3) внесены выходные сигналы M_1 , M_2 и M_3 , предназначенные для трех электромеханических устройств и последовательных состояний z^{n+1} , z_0^{m+1} , зависящие от входных величин τ^m , z^m и z_0^m .

8.1.5. Формирование таблицы запуска и таблицы выходных сигналов

Теперь следует составить уравнения для входных сигналов JK-триггеров, а именно для J_1 , K_1 , J_2 и K_2 . Для этого было бы полезно отме-



нить те значения J и K , которые при данном значении переменной состояния z^m необходимы для получения желаемого состояния последовательности z^{n+1} .

Таблица 8.3. Таблица состояний при управлении электромеханическими устройствами

τ^m	z_1^m	z_0^m	z_1^{m+1}	z_0^{m+1}	M_1	M_2	M_3
0	0	0	0	1	1	1	1
0	0	1	1	1	0	1	1
0	1	0	0	0	1	1	0
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Если, например, должно оставаться состояние $z^m = 0$, этого можно достичь с помощью установки $J = 0$ при произвольном K . Пусть $K = 1$ означает «возврат в исходное состояние» и $K = 0$ «запоминание». Оба значения ведут к получению $z^m = 0$. Аналогичные размышления приводят к другим величинам, которые всегда содержат степень свободы.

Таблица 8.4. Запуск JK-триггера в зависимости от старого и нового состояний.

z^m	z^{m+1}	J	K	описание
0	0	0	d	запоминание или возврат в исходное состояние
0	1	1	d	смена или установка
1	0	1	d	смена или установка в исходное состояние
1	1	d	0	запоминание или возврат в исходное состояние

Значение из табл. 8.4 вносятся в КВ-диаграмму, которая может быть разработана на основе табл. 8.3 последовательности состояний.

Для функций запуска JK-триггеров, которые описывают логическую схему SN1, считываем из диаграммы Карно–Вейга. (табл. 8.5):

$$J_0 = \neg r^m \neg z_1^m \quad (8.1)$$

$$J_1 = \neg r^m z_0^m \quad (8.2)$$

$$K_0 = r^m \vee z_1^m = \neg(\neg r^m \neg z_1^m) = \neg J_0 \quad (8.3)$$

$$K_1 = r^m \vee \neg z_0^m = \neg(\neg r^m z_0^m) = \neg J_1 \quad (8.4)$$

Таблица 8.5. КВ-диаграмма для функций запуска JK-триггеров и выходные функции.

		r^m		r^m	
J_1K_1	J_0K_0	J_1K_1	J_0K_0	$M_1M_2M_3$	$M_1M_2M_3$
0d	1d	0d	0d	111	000
1d	d0	0d	d1	011	000
d0	d1	d1	d1	010	000
d1	0d	d1	0d	110	000

Из КВ-диаграммы можно такое получить и выходные функции, которые реализованы в логической функции SN2:

$$M_1 = \neg r^m \neg z_0^m \quad (8.5)$$

$$M_2 = \neg r^m \quad (8.6)$$

$$M_3 = \neg r^m \neg z_1^m = J_0 \quad (8.7)$$

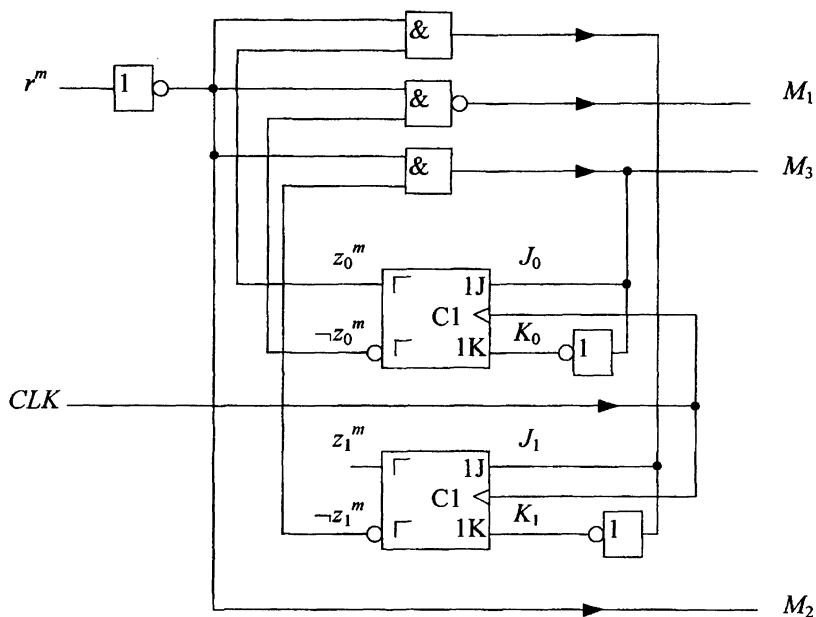


Рис. 8.4. Схема управления электромеханическими устройствами.

Мы видим, что уравнение для J_0 и M_3 идентичны. Поэтому требуется только лишь реализовать соответствующий И-вентиль. Так-

же целесообразным является использование упрощений $K_1 = \neg J_1$ и $K_0 = \neg J_0$.

8.2. Синтез драйверов (пример 2)

8.2.1. Постановка задачи

Пусть необходимо разработать схему управления светодиодами, по которой циклически проходят сигналы красный-красный и желтый-зеленый-желтый-красный. Процесс переключения должен управляться тактовым сигналом. Схема должна быть выполнена на D-триггерах в соответствии с рис. 8.5. Схема управляет светодиодами напрямую. По этой причине предусмотрены три D-триггера, хотя для реализации четырех состояний хватило бы двух D-триггеров. Речь поэтому идет о драйвере Мура, в котором логическая схема SN2 упростилась до уровня сквозных соединений. Схема не имеет других входов, кроме тактового входа CLK.

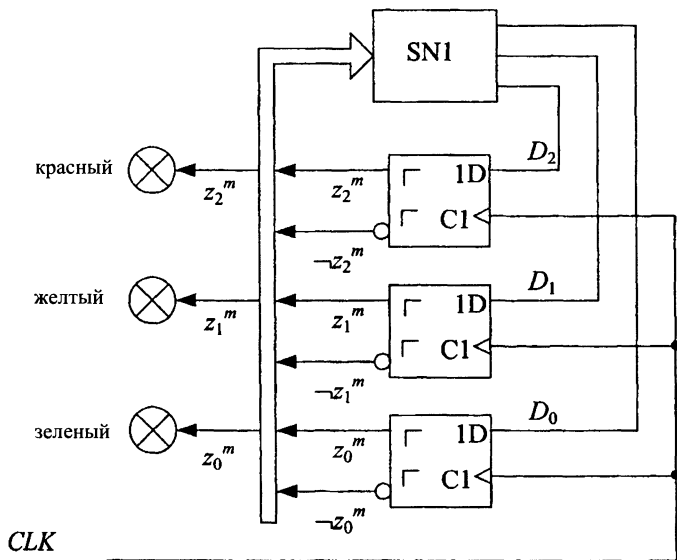


Рис. 8.5. Структура драйвера.

8.2.2. Составление диаграммы состояний

Диаграмма состояний схемы управления светофором составляется очень просто, поскольку переходы между состояниями происходят при каждом такте и входная переменная не является условием для

подобного перехода. Как показано на рис. 8.6, диаграмма имеет круговой характер. Проходятся только 4 из 8 возможных состояний.

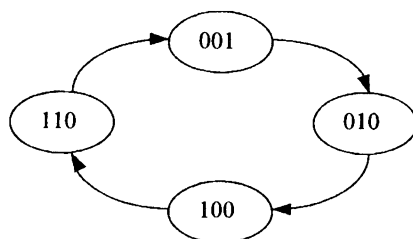


Рис. 8.6. Диаграмма состояний для схемы управления светофором (в кружках z_2^m , z_1^m , z_0^m).

8.2.3. Составление таблицы последовательности состояний

Таблица последовательности состояний (табл. 8.6) выведена из диаграммы состояний. Внесены только 4 состояния, подлежащие прохождению за цикл. Четыре неиспользуемых состояния в настоящий момент не учитываются. Но необходимо убедиться, что драйвер после включения, при котором он может войти в какое-либо произвольное состояние, через несколько тактов перейдет в нормальный цикл. Позже это должно быть проконтролировано.

Таблица 8.6. Таблица последовательности состояний для схемы управления светофором.

z_2^m	z_1^m	z_0^m	z_2^{m+1}	z_1^{m+1}	z_0^{m+1}
1	0	0	1	1	0
1	1	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0

8.2.4. Разработка логической схемы для запускающих сигналов

Для разработки логической схемы, выдающей запускающие сигналы трех D-триггеров, разрабатываются на основе таблицы последовательности состояний три КВ-диаграммы. Это просто, поскольку D-триггер значение, которое присутствует на D-входе, запоминаем как следующее состояние: $D_i = z_i^{m+1}$. Значения для пока неиспользуемых состояний устанавливается произвольно (d).

Из рис. 8.7 выводим следующие уравнения для запускающих сигналов:

$$D_2 = z_2^{m+1} = \neg z_1^m \neg z_0^m \vee \neg z_2^m \neg z_0^m \quad (8.8)$$

$$D_1 = z_1^{m+1} = \neg z_1^m \quad (8.9)$$

$$D_0 = z_0^{m+1} = z_1^m z_2^m \quad (8.10)$$

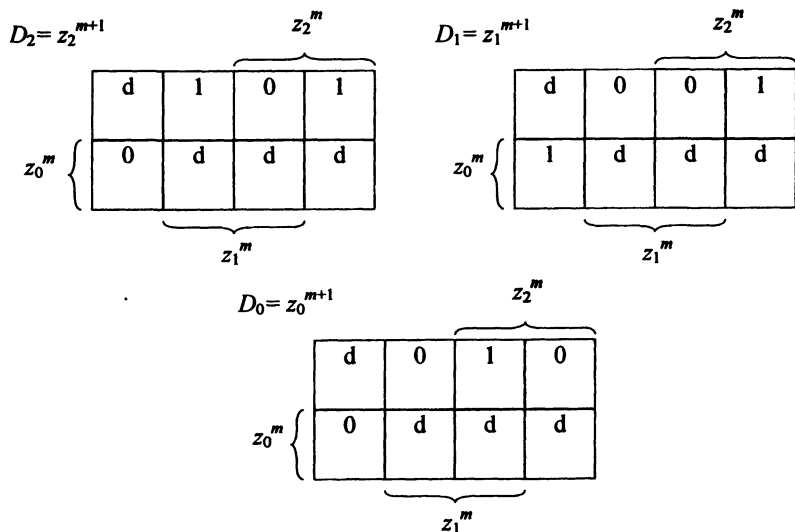


Рис. 8.7. КВ-диаграмма для минимизации логической схемы на основе D-триггеров.

8.2.5. Полная таблица состояний

Теперь на основе уравнений управляющих сигналов могут быть установлены последовательности для пока неиспользованных состояний 000, 011, 101 и 111. Полная таблица состояний (табл. 8.7) составлена при условии использования этих состояний.

Таблица 8.7. Полная таблица последовательности состояний для схемы управления светофором.

z_2^m	z_1^m	z_0^m	z_2^{m+1}	z_1^{m+1}	z_0^{m+1}
0	0	0	1	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	0	0	1

8.2.6. Полная диаграмма состояний

Как следует из полной диаграммы состояний (рис. 8.8), все состояния, в которые схема может войти при включении, в конце концов, вводят в цикл. Для этого необходимо максимально 2 такта.

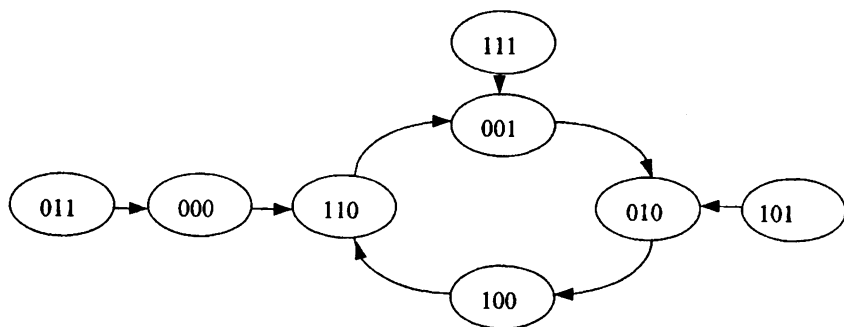


Рис. 8.8. Диаграмма состояний для управления светофором с использованием всех состояний.

Следует обратить внимание на то, что какой-либо другой выбор термов типа don't care в КВ-диаграмме (рис. 8.7) привел бы к другой диаграмме состояний. Но круг из состояний 001, 010, 100 и 110 всегда будет в наличии.

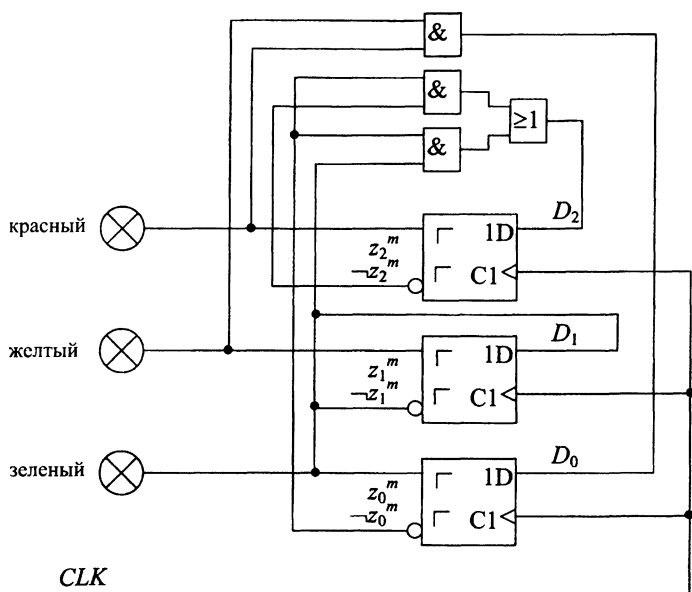


Рис. 8.9. Схема управления светофором

Может случиться, что состояния, которые не принадлежат к круговому циклу драйвера, не войдут автоматически после нескольких тактов в этот круг. Тогда может случиться, что драйвер при включении не войдет в желаемый круговой цикл, а «зависнет» в другом цикле. Для того, чтобы этого избежать, необходимо по другому установить термы don't care.

8.2.7. Временные характеристики драйверов

Для того, чтобы драйвер (рис. 8.10) функционировал так, как это было рассчитано в предыдущем параграфе необходимо выдержать несколько временных условий. Теперь эти условия необходимо исследовать более подробно.

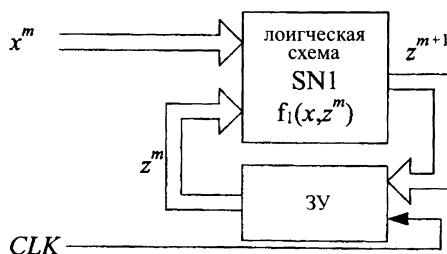


Рис. 8.10. Синхронный драйвер

Для этого изобразим действующий интервал и интервал опрокидывания, привязав их к тактовому сигналу CLK (рис. 8.11).

На рисунке показано перекрытие t_{krit} действующего интервала и интервала опрокидывания. Если в драйвере тактовых сигналов (clock skew) происходит расширение действующего интервала и интервала опрокидывания, то возможны два варианта. В случае управляемых одним фронтом D-триггеров, это может привести к перекрытию действующего интервала и интервала опрокидывания. При применении триггеров с управлением по двум фронтам интервалы — действующий и опрокидывания — не перекрываются, так что t_{krit} становится отрицательным.

На рис. 8.11 показаны также выходные сигналы триггеров z^m . Они стабильны вне интервалов опрокидывания. В интервалы опрокидывания они постоянно изменяются. Выходные сигналы x^m стабильны в тоже время, что и выходные сигналы триггеров.

Теперь можно рассмотреть выходные сигналы логической схемы SN1. Для этого кратко проанализируем общие характеристики логической схемы.

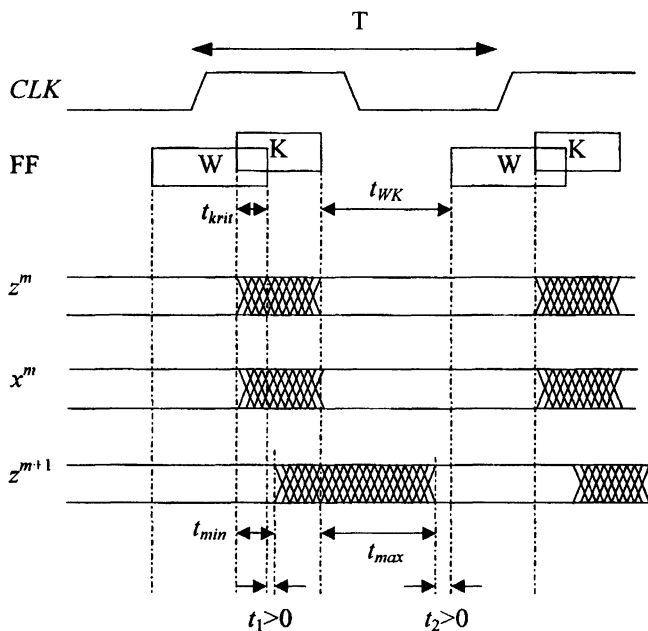


Рис. 8.11. Временные характеристики драйвера.

- Когда входные величины логической схемы изменяются, выходной сигнал определенное время t_{min} останется неизменным. Это время t_{min} является временем запаздывания («мертвое время»). Оно обусловлено временем задержки вентилях. Кроме того линии связи между блоками ЗУ и логической схемой также имеют определенное время задержки.
- Затем выходные величины начинают изменяться. Через определенное время t_{max} все переходные процессы закончились. После этого выходной сигнал стабилен.

Итак, характеризующие состояния переменные начинают изменяться самое раннее вслед за истечением времени t_{min} после начала интервала опрокидывания. После времени t_{max} , прошедшего от окончания интервала опрокидывания, выходные сигналы z^{m+1} драйвера стабильны (рис. 8.11).

По этому рисунку можно определить условия функционирования драйвера. Существенным условием функционирования драйвера является то, что входные переменные триггеров во время действующего интервала должны быть стабильны.

- Поэтому характеризующие состояние переменные z^{m+1} могут изменяться только после окончания действующего интервала. Время t_1 должно быть больше 0.

$$t_1 = t_{min} - t_{krit} > 0 \quad (8.11)$$

- При отсутствии смещения тактового сигнала действующий интервал и интервал опрокидывания должны максимально перекрываться вокруг минимального времени задержки t_{min} . При большом смещении тактового сигнала это условие можно выполнить только с триггерами, управляемыми двумя фронтами. С помощью выбора коэффициента заполнения тактовых импульсов эти интервалы времени могут варьироваться в широких пределах.
- После окончания интервала опрокидывания логическая схема должна вычислить новые входные переменные для блоков ЗУ. Этот процесс должен быть окончен, включая все переходные явления, к моменту начала следующего действующего интервала.

Поэтому второе условие реализации обратной связи гласит:

$$t_2 = t_{WK} - t_{max} > 0 \quad (8.12)$$

Эти условия можно понять более подробно, если учесть различия между величинами времени задержки сигналов в логических схемах SN1 и SN2.

Теперь рассмотрим условия для выходных функций y в логической схеме SN1. Если время задержки этой логической схемы равно времени задержки SN1, выходные сигналы y будут действительны в то же время, что и z^{m+1} . Это обстоятельство может быть использовано для того, чтобы выходной сигнал y подать в буферные блоки ЗУ, чтобы выходные величины были готовы к стабильному использованию синхронно по отношению к x^m и z^m (рис. 8.12).

8.3. Упражнения

Задача 8.1. Разработайте синхронную схему с состояниями A, B, C, D. Она должна зависеть от входных сигналов $R (= \text{reset})$ и $V (= \text{Vprvarts, forvard, вперед})$ в следующих сочетаниях:

При $R = 0 \ V = 1$ пробегание цикла A, B, C, D, A, B;

При $R = 0$ $V = 0$ прохождение цикла D, C, B, A, DC;

При $R = 1$ переход в состояние A не зависит от V .

- Начертите диаграмму состояний.
- Сформулируйте переходную таблицу.
- Реализуйте схему с двумя D – триггерами.

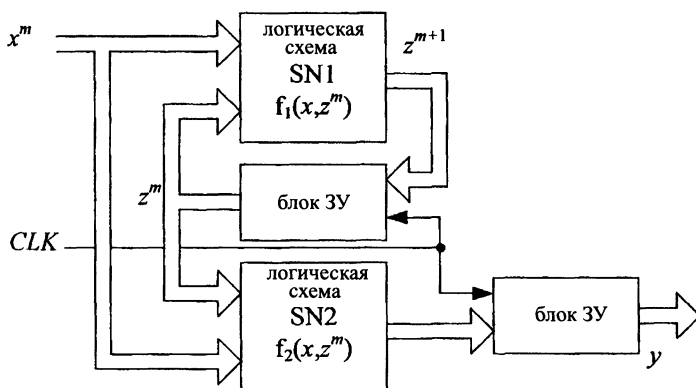


Рис. 8.12. Синхронный драйвер Мили с буферными блоками ЗУ на выходе

Задача 8.2. Разработайте представленную в главе 8.1 схему управления электротехническими устройствами, используя вместо JK-триггеров:

- RS-триггеры;
- D-триггеры.

Сравните затраты на реализации трех видов схем.

Задача 8.3. Разработайте синхронный драйвер Мура, который представляет собой реализацию парковочного автомата, выдающего квитанции на парковку стоимостью в 1,5 евро.

Монеты могут быть вброшены в любой последовательности. Если достигнута или превышена сумма в 1,5 евро, должна быть выдана квитанция на парковку и, при необходимости, выплачена сдача.

Паркованный автомат содержит прибор для контроля монет, который принимает только монеты в 50 центов и в 1 евро. После каждого тактового сигнала на выходе прибора для контроля монет воспроизводится в соответствии с представленной ниже таблицей истинности то, что было вброшено.

Исключается, чтобы на приборе для контроля монет воспроизводилось $M = (1, 1)$ и чтобы в течение тактового периода вбрасывалось более одной монеты. Фальшивые монеты автоматически возвращаются.



Вброс	Выход прибора для контро- ля монет $M = (x_1, x_0)$
Никакой или фальшивая монета	00
Монета в 50 центов	01
Монета в 1 евро	10

Квитанция на парковку выдается с выходным сигналом $S = 1$, одновременно механически запирается прием монет. В других случаях вброс возможен. С сигналом $R = 1$ возвращается монета в 50 центов

а) Выдайте диаграмму состояний и соответствующую ей таблицу последовательности состояний.

б) Определите переходные функции и выходные функции.

ГЛАВА 9

МУЛЬТИПЛЕКСОРЫ И ПРЕОБРАЗОВАТЕЛИ КОДА

В этой главе представлены два стандартных схемных элемента: мультиплексор и преобразователь кода. Эти схемные элементы пригодны для реализации булевых функций или для сведения в пучок многих информационных каналов для передачи по одной линии.

9.1. Мультиплексор

Мультиплексор представляет собой схемный элемент, который подключает один из n цифровых входов к одному выходу. Вход выбирается из группы посредством селективных (адресных) входов.

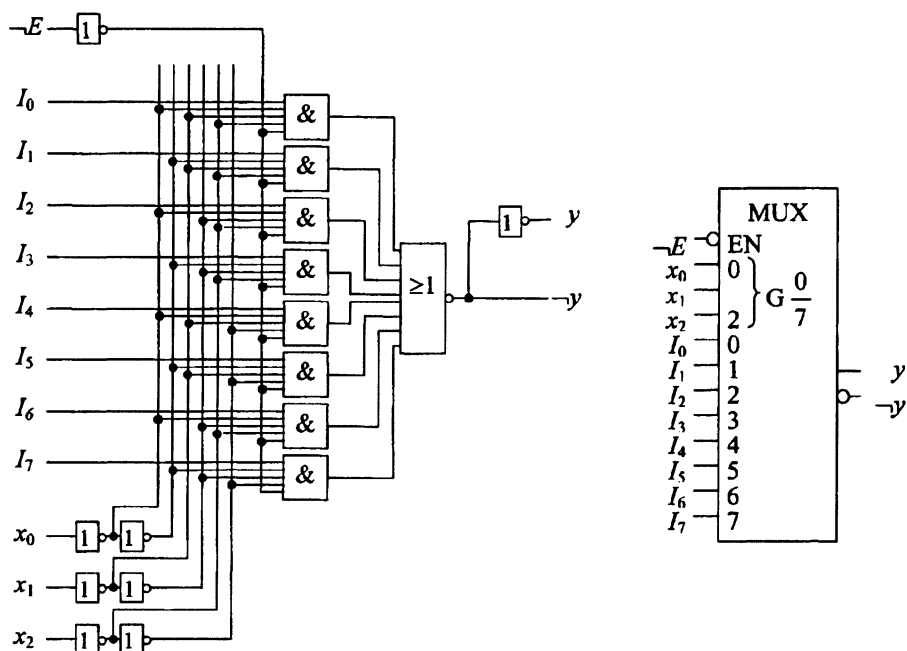


Рис. 9.1. 8:1-мультиплексор 74151 с логическим символом.

В качестве примера на рис. 9.1 показан схемный элемент 74151. Этот мультиплексор обозначается как мультиплексор 8:1, так как с его помощью 8 различных входов I_i могут быть на выбор подсоединены к одному выходу y . В КМОП-версии данный схемный элемент реализуется с помощью проходных вентилях (transmission gates), с помощью селективных входов x_2, x_1, x_0 выбирается один I_i вход. После стабилизации адресных сигналов и сигналов данных выбранный вход может быть подключен с помощью активирующего сигнала $\neg E$ (enable, разрешающий). Выход y остается в состоянии 0, пока $\neg E = 1$. При $\neg E = 0$ выбранный выход подключается.

Данный схемный элемент выполняет следующую логическую функцию:

$$y = \neg E (\neg x_2 \neg x_1 \neg x_0 I_0 \vee \neg x_2 \neg x_1 x_0 I_1 \vee \neg x_2 x_1 \neg x_0 I_2 \vee \neg x_2 x_1 x_0 I_3 \vee x_2 \neg x_1 \neg x_0 I_4 \vee x_2 \neg x_1 x_0 I_5 \vee x_2 x_1 \neg x_0 I_6 \vee x_2 x_1 x_0 I_7) \quad (9.1)$$

Показанный на рис. 9.1 логический символ 8:1-мультиплексора 74151 обозначен надписью MUX. Функция мультиплексора описывается зависимостью $U(G)$ селективных входов x_i и входов данных I_i . Селективные входы пронумерованы от 0 для x_0 до 2 для x_2 .

Таблица 9.1. Таблица истинности 8:1-мультиплексора 74151 (x является произвольной величиной $\in \{0, 1\}$)

$\neg E$	x_2	x_1	x_0	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	y
1	d	d	d	d	d	d	d	d	d	d	d	0
0	0	0	0	d	d	d	d	d	d	d	x	x
0	0	0	1	d	d	d	d	d	d	x	d	x
0	0	1	0	d	d	d	d	d	x	d	d	x
0	0	1	1	d	d	d	d	x	d	d	d	x
0	1	0	0	d	d	d	x	d	d	d	d	x
0	1	0	1	d	d	x	d	d	d	d	d	x
0	1	1	0	d	x	d	d	d	d	d	d	x
0	1	1	1	x	d	d	d	d	d	d	d	x

9.1.1. Реализация функций мультиплексора

Мультиплексор можно использовать для реализации логических функций. Покажем это на примере. Подлежащая реализации логическая функция задана диаграммой Карно, показанной на рис. 9.2. Используется мультиплексор 8:1.

Мультиплексор 8:1 имеет три селективных входа, на которые подаются три из четырех переменных. Для выбора этих трех переменных имеются четыре возможности.

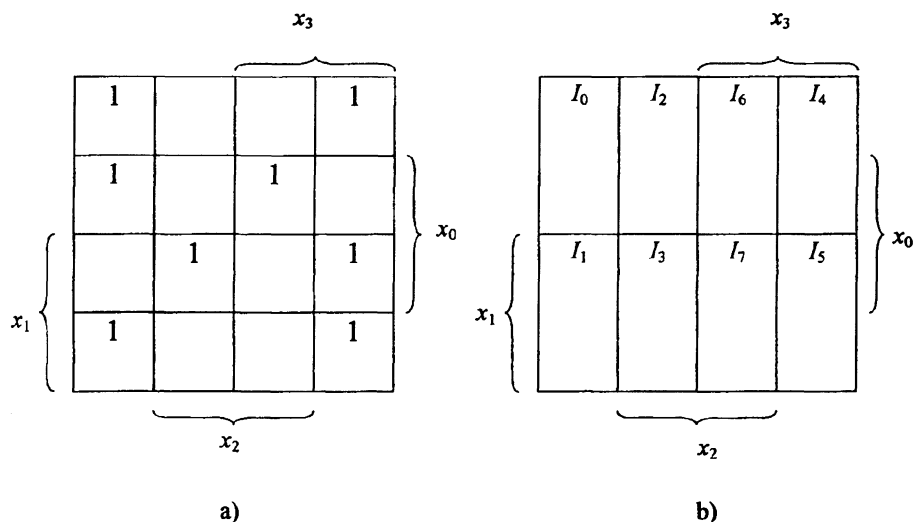


Рис. 9.2. а) Диаграмма Карно–Вейга для функции, приведенной в примере; б) определение входных полей.

Не используемая в данный момент переменная так подается на входы данных I_i , чтобы на выходе мультиплексора появилось заданное значение функции. Теперь поясним прохождение этого процесса.

Сначала установим, какие переменные должны использоваться на селективных входах. Допустим выбраны x_3, x_2, x_1 . С этими тремя переменными на селективных входах в КВ-диаграмме соотносятся поля с 2-мя переменными. На рис. 9.2 заданы поля, которые должны быть приведены в соответствие с одним из входных векторов. При нумерации полей следует учитывать вес селективных входов: x_3 имеет вес 2^2 , x_2 имеет вес 2^1 , x_1 имеет вес 2^0 .

Затем на входы данных мультиплексора должны быть поданы соответствующие остаточные функции. Если какая-либо из них не содержит 1, то на соответствующем входе данных должен быть 0. При наличии двух единиц в одном поле на вход данных подается 1. В случае, когда в поле только одна 1, оно принимает позицию, соответствующую 1.

Например, на рис. 9.2 б можно прочесть, что на I_1 должно быть подано $\neg x_0$, так как поле I_0 имеет только одну 1 в позиции, которая не покрыта граничным обозначением x_0 .

Альтернативно для реализации данной функции может быть применен мультиплексор типа 16:1. В этом случае на его входы должны быть поданы только лишь 0 и 1. Но относительно аппаратных затрат этот вариант не имеет преимуществ перед мультиплексором типа 8:1.

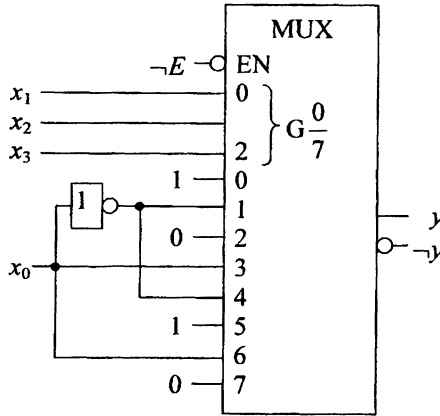


Рис. 9.3. Включение мультиплексора для реализации функции, определенной на рис. 9.2.

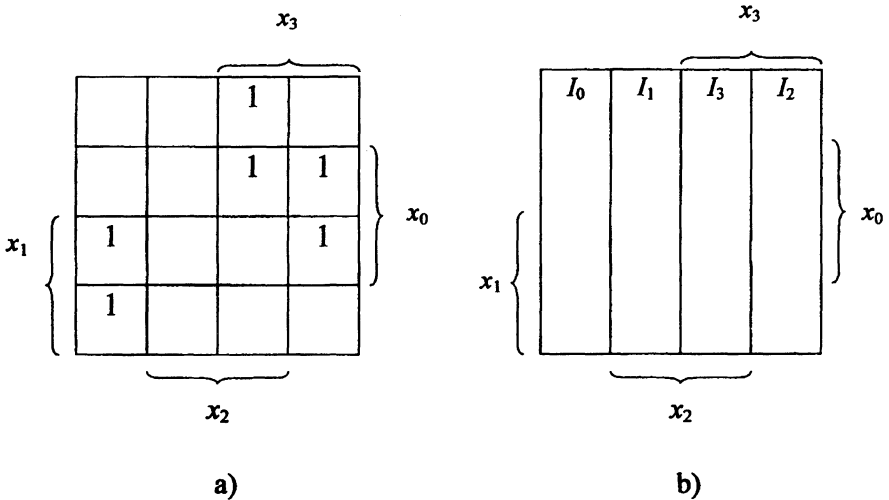


Рис. 9.4. а) Диаграмма Карно-Вейга для функции, взятой в качестве примера; б) Определение входных полей.

В случае применения мультиплексора типа 4:1 на оба селективные входы подаются две переменные, а на четыре входа данных

подаются по одной функции DNF (или KNF), образованные двумя другими переменными. Пример показан на рис. 9.4. Если подать на селективные входы мультиплексора x_3 и x_2 , подача сигналов на входы данных будет особенно простым.

В этом случае можно обойтись одним 4:1-мультиплексором, без дополнительных вентилях (рис. 9.5). Если к селективным входам подвести x_1 и x_6 , то в этом варианте к входам данных необходимо подключить дополнительные вентиля.

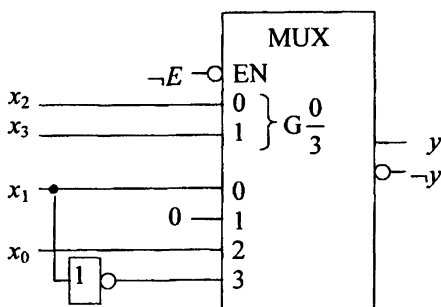


Рис. 9.5. Включение мультиплексора для реализации функции, определенной на рис. 9.4.

9.2. Преобразователь кода

Преобразователь кода представляет собой схему, которая преобразует слова, соответствующие коду 1, подаваемые на m входов, в слово из другого кода, кода 2.

В этом случае кодовое слово на выходе имеет длину n бит. В логическом символе, приведенном на рис 9.6, оба кода указаны в надписи.

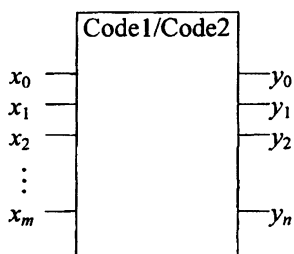


Рис. 9.6. Логический символ преобразователя кода

Преобразователь кода применяется в следующих областях:

- для трансформации, например, двоично-десятичного кода в шестнадцатичный код;

- для генерации наборов функций;
- в качестве демультиплексеров. Как разъясняется ниже, демультиплексер является противоположностью мультиплексера. Он служит для того, чтобы направить данные с одного информационного канала многим адресатам.

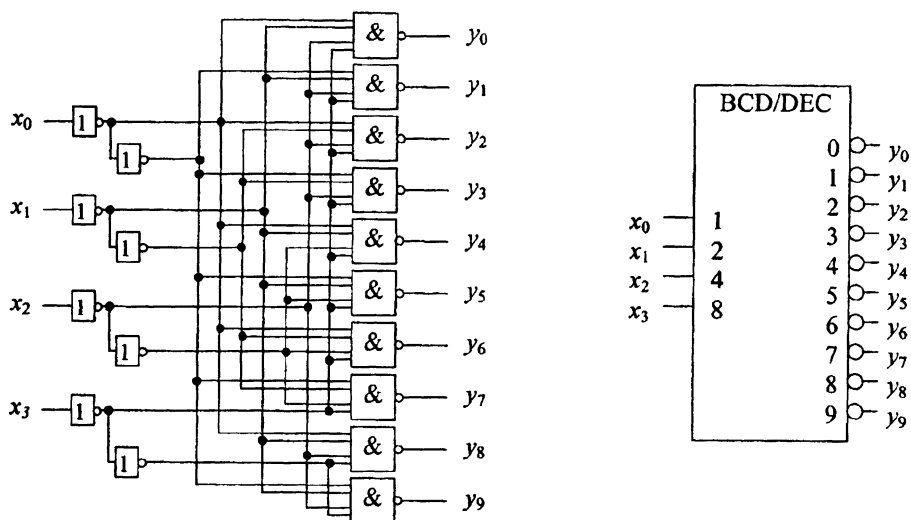


Рис. 9.7. Структурная схема и логический символ преобразователя 7442 двоично-десятичного кода в десятичный код

9.2.1. Преобразователь двоично-десятичного кода в десятичный код 7442

В качестве примера на рис. 9.7 представлен преобразователь кода 7442. Он преобразует двоично-десятичный код в код 1 из 10. Код 1 из 10 является кодом, слова которого имеют свойство, заключающиеся в том, что все разряды, кроме одного, имеют значение 1. Преобразователь кода имеет 4 входа и 10 выходов. В исходном состоянии выходы находятся в состоянии 1 и в случае выборки переключатся на 0.

Каждый выходной сигнал реализует соответствующие макстермы.

$$y_i = M_i \quad i = 0, 1, \dots, 9 \quad (9.2)$$

Функцию можно интерпретировать также и как, реализующую на каждом выходе соответствующий инвертированный минтерм:

$$y_6 = M_6 = x_3 \vee \neg x_2 \vee \neg x_1 \vee x_0 = \neg (\neg x_3 x_2 x_1 \neg x_0) \quad (9.3)$$

В логическом символе преобразователя кода 7442 указаны оба кода, между которыми происходит трансформация. В данном случае это BCD/DEC, что означает из двоично-десятичного кода в десятичный код. Разряды двоично-десятичного кода (BCD) даны внутри обрамления символа слева. На правой стороне указаны разряды для выходов.

Таблица 9.2. Таблица истинности преобразователя 7442 двоично-десятичного кода в десятичный код

десятичный	x_3	x_2	x_1	x_0	y_9	y_8	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	1	1	0	1
2	0	0	1	0	1	1	1	1	1	1	1	0	1	1
3	0	0	1	1	1	1	1	1	1	1	0	1	1	1
4	0	1	0	0	1	1	1	1	1	0	1	1	1	1
5	0	1	0	1	1	1	1	1	0	1	1	1	1	1
6	0	1	1	0	1	1	1	0	1	1	1	1	1	1
7	0	1	1	1	1	1	0	1	1	1	1	1	1	1
8	1	0	0	0	1	0	1	1	1	1	1	1	1	1
9	1	0	0	1	0	1	1	1	1	1	1	1	1	1
10	1	0	1	0	1	1	1	1	1	1	1	1	1	1
11	1	0	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	0	0	1	1	1	1	1	1	1	1	1	1
13	1	1	0	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	0	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1

9.2.2. Демультимплексор

Преобразователи кода, которые преобразуют двоичный код в код 1 из n , могут быть использованы в качестве демультимплексоров. Демультимплексор выполняет задачу, обратную задаче мультиплексора. Демультимплексор коммутирует входной сигнал E к выходам, выбираемым с помощью адресных шин.

В качестве примера используем для этой цели преобразователь двоично-десятичного кода в десятичный код. Для этого имеющий наибольшую значимость вход использован как вход данных E (рис. 9.8).

Входы x_2 , x_1 и x_0 становятся адресными входами демультимплексора. Они выбирают выход. В качестве выходов применяются только электроды от 0 до 7. Функцию определяет таблица истинности

(табл. 9.2). А именно, имеющий наибольшее значение вход x_3 определяет, имеет ли выбранный с помощью входов x_2 , x_1 и x_0 выход состояние 0 или 1. Для этой цели можно было бы также использовать двоичное число, полученное преобразователем восьмеричного кода с 8-ю выходами.

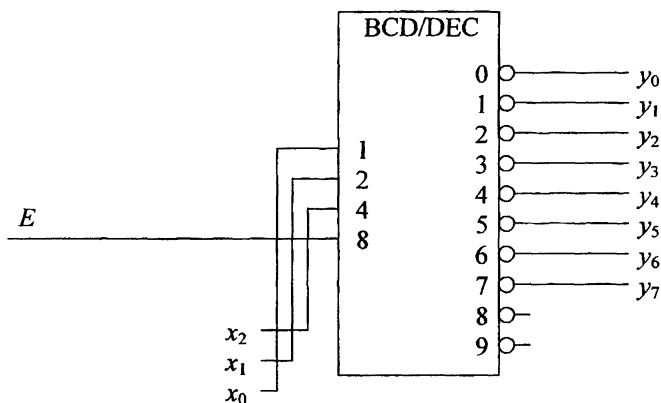


Рис. 9.8. Применение преобразователя десятичного кода в качестве демультиплексора.

Мультиплексор и демультиплексор могут совместно образовывать линию передачи данных, позволяющую передавать n параллельных потоков данных по одному проводу. Принцип этого показан на рис. 9.9. На адресные входы мультиплексора и демультиплексора периодически подаются адреса от 0 до 7. Благодаря этому каждый входной сигнал I_i переносится в одну восьмую интервала времени на выход y_i . Таким образом каждой линии с помощью данной системы выделен отрезок временного интервала. Метод называется временным мультиплексированием (time division multiple access, TDMA, коллективный доступ с временным разделением).

9.2.3. Генерирование наборов функций

С помощью преобразователя кода, который трансформирует в код типа 1 из n , может осуществляться генерация набора функций. В качестве примера рассмотрим реализацию четырех булевых функций y_3 , y_2 , y_1 и y_0 с тремя входами x_2 , x_1 и x_0 . Они приведены в табл. 9.5. В данном случае применен преобразователь двоичного кода в восьмеричный код, реализуемый схемой 74138, поскольку этот преобразователь кода имеет 3 входа и 8 выходов. На основании таблицы истинности можно сказать, что состояние выходов соответствуют

инвертированным минтермам. Альтернативно состояния выходов можно интерпретировать как макстермы функций.

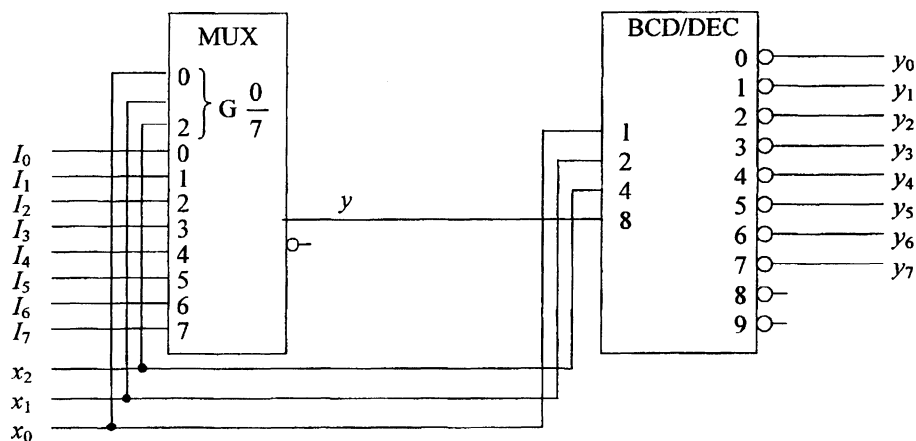


Рис. 9.9. Принцип построения линии передачи данных с мультиплексором и демультиплексором.

Таблица 9.3. Таблица истинности функций y_3, y_2, y_1, y_0

десятичный	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1
2	0	1	0	1	0	0	0
3	0	1	1	1	1	1	1
4	1	0	0	0	1	0	0
5	1	0	1	0	1	0	1
6	1	1	0	1	1	0	1
7	1	1	1	1	0	1	0

В зависимости от того, интерпретируются ли выходы как инвертированные минтермы или как макстермы, возможны два варианта реализации. В первом случае будет образована DNF, во втором случае KNF.

В случае, если исходить из минтермов, необходимо действовать в следующем порядке:

выход, для которого должно иметь место значение функции равное 1, необходимо подключить к NAND-вентилю (рис. 9.10). На основе инверсии выходов схемы 74138 и инверсии значения функции с помощью NAND образуется логическая функция ODER, необходимая для формирования DNF.

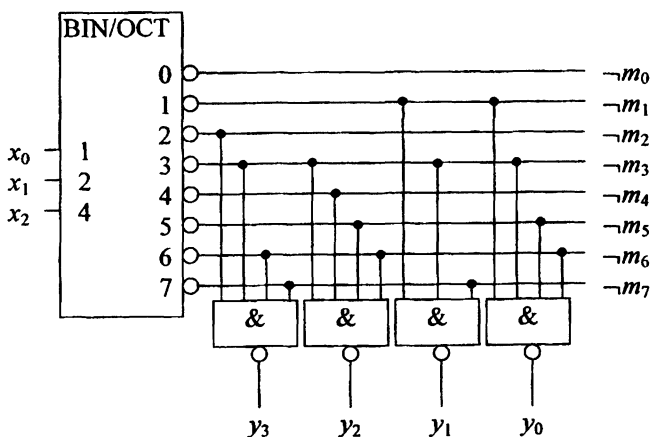


Рис. 9.10. Реализация DNF набора с помощью преобразователя кода

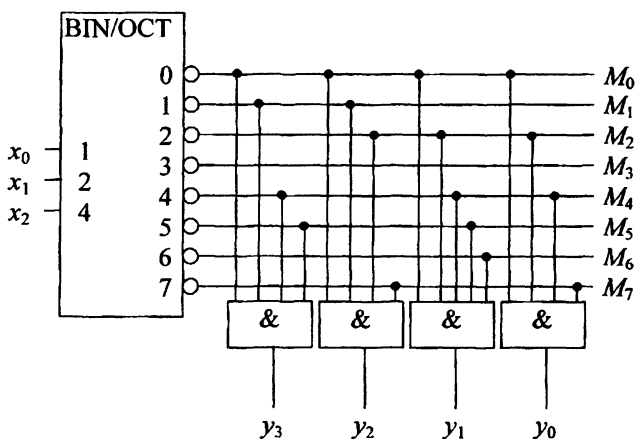


Рис. 9.11. Реализация KNF набора функций с помощью преобразователя кода.

Для образования KNF выходы интерпретируются как макстермы. Таким образом, мы подключаем выходы, входящие в состав входных векторов, которые должны иметь значение функции 0, к И-вентилю (рис. 9.11), поскольку макстермы в KNF соединены в соответствии логической функцией И.

9.3. Аналоговые мультиплексоры и демультиплексоры

При необходимости мультиплексирования аналоговых сигналов для коммутации могут быть использованы проходные вентили (trans-

mission-gates). Так как проходные вентили имеют инвертированный управляющий вход, мультиплексор должен иметь инвертирующие выходы. Схема мультиплексора показана на рис. 9.12. Поскольку она может быть использована потоков информации в обоих направлениях, ее можно использовать как мультиплексор, и как демультиплексор для аналоговых и цифровых сигналов.

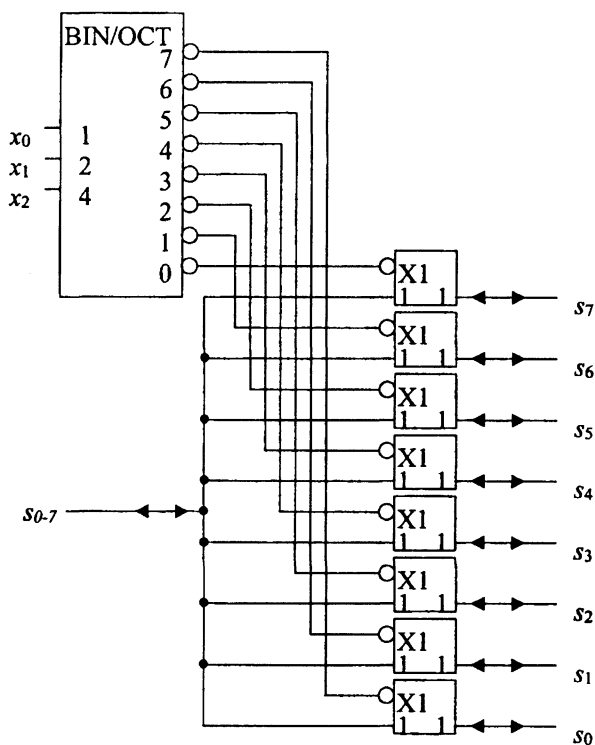


Рис. 9.12. Аналоговый мультиплексор (направление прохождения сигналов справа налево) и демультиплексор (направление прохождения сигналов слева направо)

9.4. Упражнения

Задача 9.1. Переключательная функция $f(a_2, a_1, a_0)$, заданная таблицей (рис. 9.13), должна быть реализована с помощью приведенного мультиплексора. Укажите схему подключения к входам I_i и x_i .

Задача 9.2. Пусть необходимо реализовать логическую схему для функций $F_0(a_1, a_2, a_3)$ и $F_1(a_1, a_2, a_3)$ с помощью мультиплексора или с помощью преобразователя кода. Функции определяются табл. 9.4

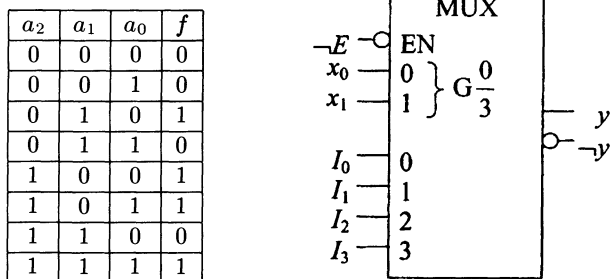


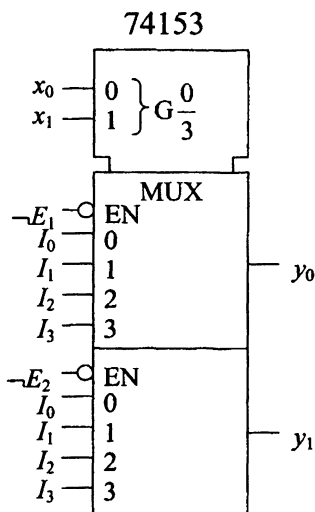
Рис. 9.13.

а) Примените преобразователь кода 74138 (см. стр. 241) для реализации функций.

Таблица 9.4.

a_2	a_1	a_0	F_0	F_1
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

б) Реализуйте эти функции с помощью схемного модуля 74153, который содержит 2 мультиплексора 4:1. Логический символ 74153 представлен справа.



Задача 9.3. Спроектируйте преобразователь кода для трансформации восьмеричного кода в код Грея в соответствии со следующими таблицами.

Таблица 9.5. Таблица истинности функций y_2, y_1, y_0

Вход $x_2 x_1 x_0$	Выход $y_2 y_1 y_0$
000	000
001	001
010	011
011	010

Вход $x_2 x_1 x_0$	Выход $y_2 y_1 y_0$
100	110
101	111
110	101
111	100

ГЛАВА 10

ЦИФРОВЫЕ СЧЕТЧИКИ

Цифровые счетчики представляют собой асинхронные или синхронные драйверы, которые состоят, как правило, из соединенных цепочкой триггеров. Содержание регистров интерпретируется как счетное состояние счетчика.

10.1. Асинхронный счетчик

Асинхронные счетчики представляют собой асинхронные драйверы, для которых не требуются тактовые сигналы. Входной сигнал с последовательностью подлежащих счету импульсов подводится прямо к тактовому входу первого триггера. Тактовые входы следующих триггеров подключены к выходам предыдущих триггеров. В последующем в качестве примеров будут представлены две простые схемы.

10.1.1. Двоичный счетчик по модулю 8

Двоичный счетчик по модулю 8 можно построить из JK-триггеров с управлением по отрицательному фронту, как это показано на рис. 10.1. J- и K-входы JK-триггеров установлены в состояние 1. При каждом отрицательном фронте на входе состояние на выходе первого триггера изменяется. Точно также ведут себя выходы следующих триггеров.

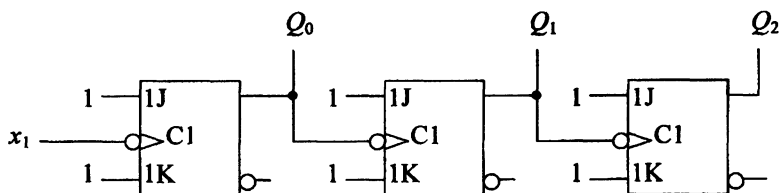


Рис. 10.1. Двоичный счетчик по модулю 8, состоящий из трех JK-триггеров.

Результатом работы этой схемы является временная диаграмма импульсов, показанная на рис. 10.2. После результата счета 111 счетчик опять возвращается к 000. Его называют счетчиком по модулю 8, так как он может периодически показывать 8 различных результатов счета.

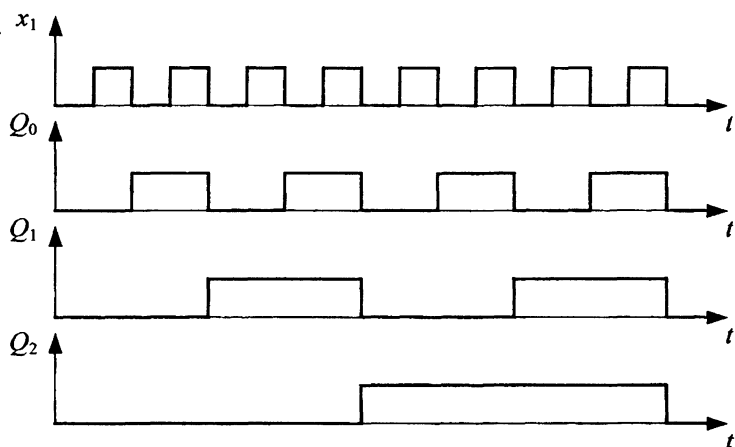


Рис. 10.2. Временная диаграмма двоичного счетчика по модулю 8, показанного на рис. 10.1.

Схема может быть использована также как делитель частоты. Как можно видеть на рис. 10.2, частота выходного сигнала каждой ступени равна половине частоты предыдущей ступени.

10.1.2. Счетчик по модулю 6

Асинхронный счетчик по модулю 6 можно получить путем расширения счетчика по модулю 8. Для этого необходимы JK-триггеры со входом установки в исходное состояние R. Триггеры устанавливаются в исходное состояние тогда, когда достигнут результат счета 6(100).

Считывание производится с помощью подключения вентиля И к выходам Q_1 и Q_2 (рис. 10.3). В счетчике по модулю 6 не должен выдаваться результат счета 110. Но как показано на временной диаграмме импульсов на рис. 10.4, это имеет место во время возврата в исходное состояние. Следовательно, возникает короткий импульс-помеха, который при некоторых применениях не может быть допустим.

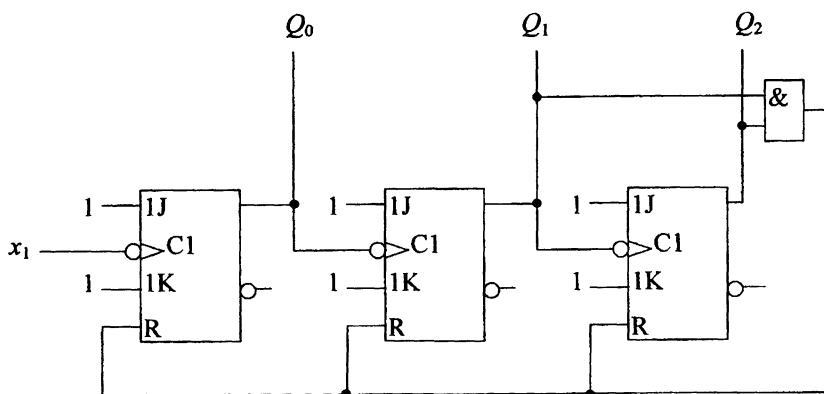


Рис. 10.3. Двоичный счетчик по модулю 6, состоящий из трех JK-триггеров

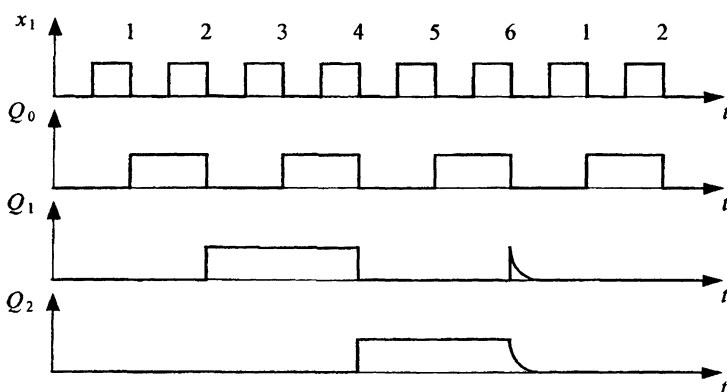


Рис. 10.4. Временная диаграмма двоичного счетчика по модулю 6, показанного на рис. 10.3

10.1.3. Асинхронный обратный счетчик

Если асинхронный счетчик должен считать в обратном направлении, то ко входам следующих ступеней должны быть подключены не выходы Q_i триггеров, а инвертированные выходы — $\neg Q_i$ (рис. 10.5). Благодаря этому JK-триггеры переключаются всегда положительным фронтом, и мы получим импульсную диаграмму, показанную на рис. 10.6.

При появлении падающего фронта входного сигнала первый триггер переключается, и его выход переходит в состояние Н. Последующие триггеры в идеальном случае переключаются одновременно.

10.1.4. Временные характеристики асинхронных счетчиков

Асинхронные счетчики перестают вести себя идеально при величинах тактового периода T_p , имеющих тот же порядок, что и время задержки t_{pd} триггеров. На рис. 10.7 показаны выходные сигналы триггеров асинхронного счетчика с временем задержки вентиля, которое приблизительно соответствует половине тактового периода. Можно видеть, что между правильными результатами счета лежат дополнительные результаты счета.

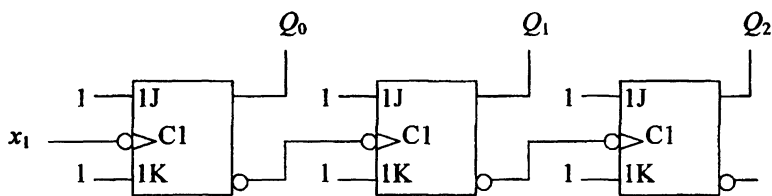


Рис. 10.5. Обратный счетчик по модулю 8, состоящий из трех JK-триггеров

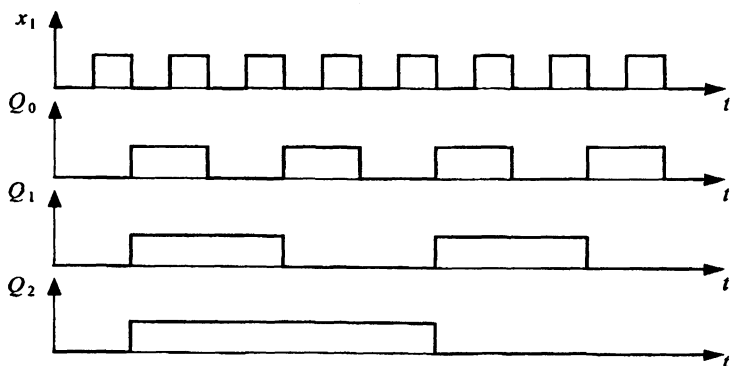


Рис. 10.6. Временная диаграмма обратного счетчика по модулю 8, показанного на рис. 10.5

При несколько больших величинах времени задержки триггеров результат счета 100 больше не наступает. При этом считывание результатов счета также становится невозможным. Максимальная тактовая частота f_{max} асинхронного счетчика с n ступенями, все из которых имеют одинаковое время задержки вентиля t_{pd} , определяется уравнением (10.1). Но реальные счетчики не достигают этих значений.

$$f_{max} = \frac{1}{nt_{pd}} \quad (10.1)$$

Асинхронные счетчики выполнены относительно просто. Но из-за их проблематичности при высоких частотах это преимущество отступает на второй план. Представленные ниже синхронные счетчики позволяют избежать этих проблем с помощью одного такта сдвига процесса переключения в последних ступенях.

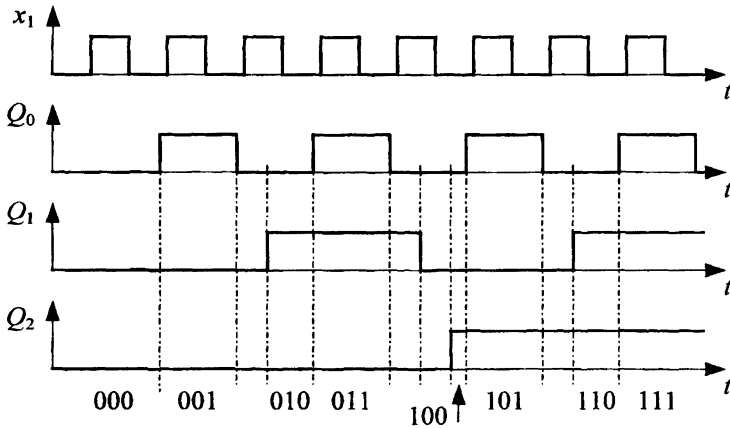


Рис. 10.7. Временная диаграмма показанного на рис. 10.1 двоичного счетчика по модулю 8, имеющего конечное время задержки.

10.2. Синхронные счетчики

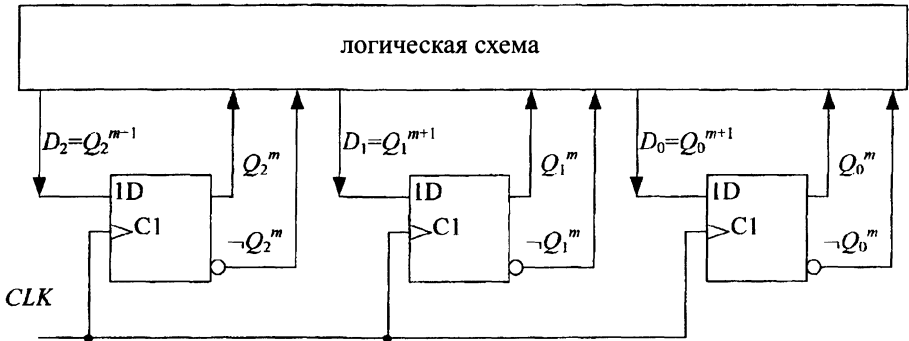


Рис. 10.8. Принцип построения синхронного счетчика

Синхронный счетчик является синхронным драйвером, в котором выполняются сформулированные в главе 8 временные условия. Рис. 10.8 иллюстрирует принцип построения синхронного счетчика с D-триггерами. Могут также быть использованы RS- или JK-триггеры. В синхронных счетчиках каждый регистр переключается почти од-

новременно. В каждый тактовый период запомненные в регистрах состояния воспроизводятся из старых состояний в логической схеме.

Конструирование синхронного счетчика может быть произведено описанными в главе 8 методами. В дальнейшем будут представлены два примера.

10.2.1. 4-битовый двоичный счетчик

Постановка задачи.

Должен быть сконструирован 4-битовый двоичный счетчик, содержащий четыре JK-триггера. Он должен выдавать сигнал переноса C_4 при переключении из положения 1111 в положение 0000. Сначала сформируем таблицу последовательности состояний (табл. 10.1).

Таблица 10.1. Таблица последовательностей состояний 4-битового двоичного счетчика.

Q_3^m	Q_2^m	Q_1^m	Q_0^m	Q_3^{m+1}	Q_2^{m+1}	Q_1^{m+1}	Q_0^{m+1}	Q_3^m	Q_2^m	Q_1^m	Q_0^m	Q_3^{m+1}	Q_2^{m+1}	Q_1^{m+1}	Q_0^{m+1}
0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	0
0	0	1	0	0	0	1	1	1	0	1	0	1	0	1	1
0	0	1	1	0	1	0	0	1	0	1	1	1	1	0	0
0	1	0	0	0	1	0	1	1	1	0	0	1	1	0	1
0	1	0	1	0	1	1	0	1	1	0	1	1	1	1	0
0	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0

Затем на основе таблицы последовательности состояний должны быть разработаны уравнения запуска для четырех JK-триггеров. Для этого вновь используем табл. 8.4, в которой зафиксированы уравнения запуска для обеспечения перехода от одного состояния к следующему состоянию. Получим четыре КВ-диаграммы (рис. 10.9), в которые внесем пары значений функций $J_i K_i$.

Можно сформулировать следующие уравнения запуска JK-триггеров:

$$J_0 = K_0 = 1 \quad (10.2)$$

$$J_1 = K_1 = Q_0^m \quad (10.3)$$

$$J_2 = K_2 = Q_0^m Q_1^m \quad (10.4)$$

$$J_3 = K_3 = Q_0^m Q_1^m Q_2^m \quad (10.5)$$

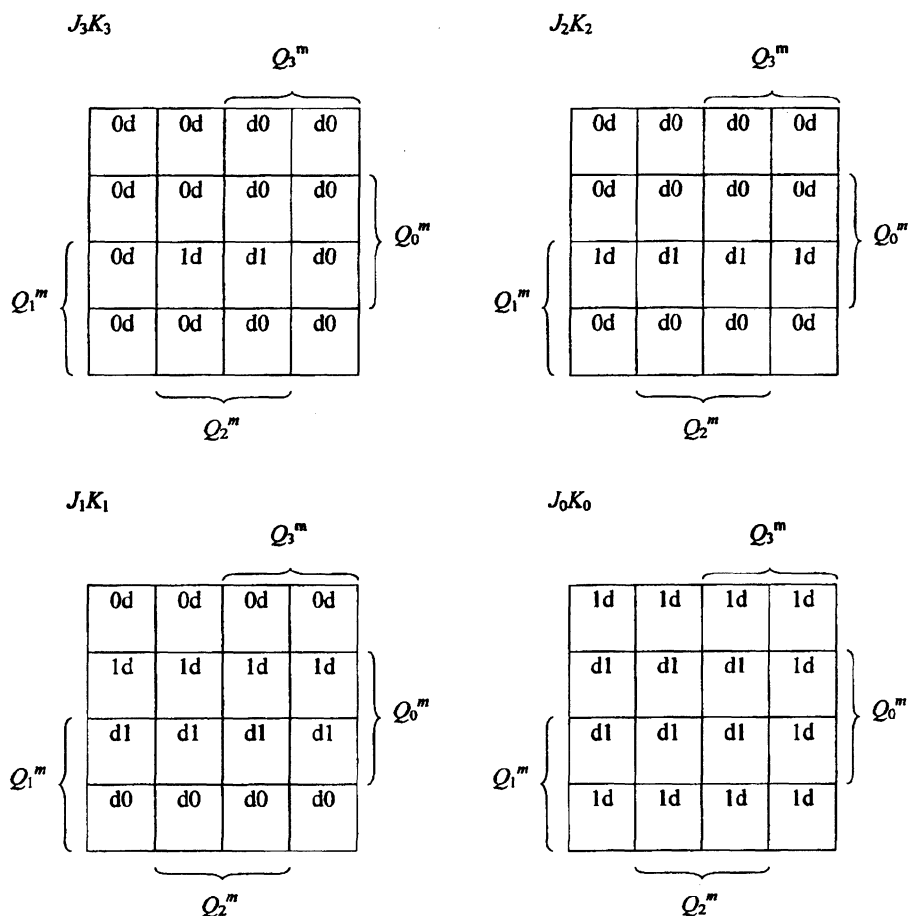


Рис. 10.9. КВ-диаграмма для функций запуска JK-триггеров

Для того, чтобы выявить систематику, уравнения были выведены с помощью КВ-диаграммы. Но эти уравнения можно также записать напрямую, если выявить из таблицы истинности, что триггер изменяет состояние, когда на выходах всех предшествующих триггеров находится 1.

В таблице перенос c_4 не приведен. Он рассчитывается аналогично на основе указанного выше рассуждения по формуле:

$$c_4 = Q_3^m Q_2^m Q_1^m Q_0^m \quad (10.6)$$

Структурная схема полного счетчика представлена на рис. 10.10

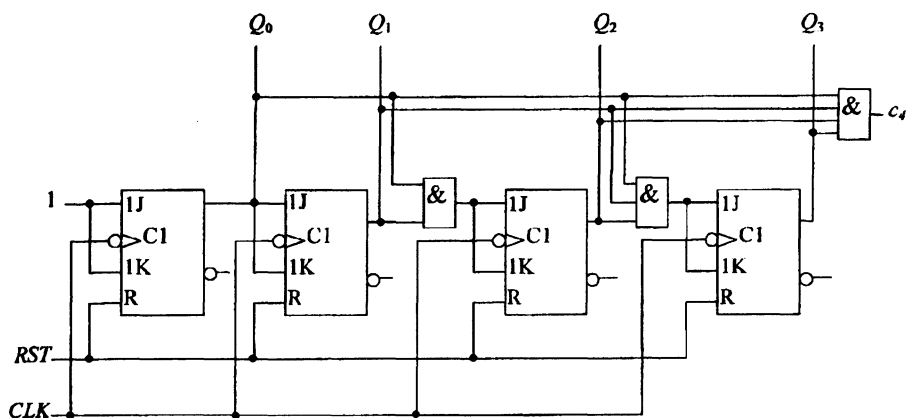


Рис. 10.10. Структурная схема синхронного 4-битового двоичного счетчика

10.2.2. Счетчик по модулю 6 с использованием кода Грея

В качестве примера рассмотрим конструирование счетчика по модулю 6 с использованием кода Грея. Необходимо сосчитать 6 чисел, передаваемых с помощью кода Грея, и при самом высоком результате счета произвести перенос. Используем в счетчике три D-триггера. Начнем с формирования таблицы последовательности состояний (табл. 10.2). Для этого сформируем циклический код Грея с 6 состояниями. При самом высоком результате счета, равном 100, сигнал переноса C_{ii} будет равен 1.

На основе таблицы последовательности состояний можно составить КВ-диаграммы для трех D-триггеров:

Таблица 10.2. Таблица последовательности состояний счетчика по модулю 6 с использованием кода Грея.

Q_2^m	Q_1^m	Q_0^m	Q_2^{m+1}	Q_1^{m+1}	Q_0^{m+1}	C_{ii}
0	0	0	0	0	1	0
0	0	1	0	1	1	0
0	1	0	0	1	0	0
0	1	1	1	1	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	1	0	d	d	d	0
1	1	1	d	d	d	0

Из этого получим функции запуска D-триггеров:

$$D_2 = Q_2^{m+1} = Q_1^m \cap Q_0^m \quad (10.7)$$

$$D_1 = Q_1^{m+1} = Q_0^m \vee Q_1^m \neg Q_2^m \quad (10.8)$$

$$D_0 = Q_0^{m+1} = \neg Q_1^m \neg Q_2^m \quad (10.9)$$

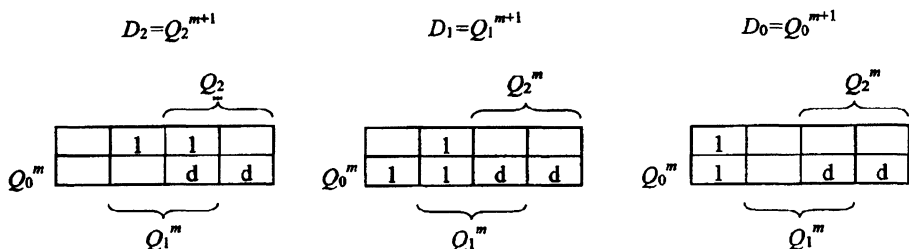


Рис. 10.11. КВ-диаграммы для запуска D-триггеров счетчика по модулю 6 для сигналов с использованием кода Грея

Для функции D_1 термы типа don't care всегда интерпретируются как 1, в то время как все другие термы don't care устанавливаются равным 0. Поэтому при следующем такте счетчик выходит из неиспользуемых состояний в состояние 010. Перенос C_{ii} может быть задан без КВ-диаграммы

$$c_{\bar{u}} = \neg Q_0^m \neg Q_1^m Q_2^m \quad (10.10)$$

Схема счетчика показана на рис. 10.12. Диаграмма состояний схемы с обоими неприменяемыми состояниями, приведенная на рис. 10.13, показывает, что схема из этих состояний и после включения начинает цикл счета в следующем такте.

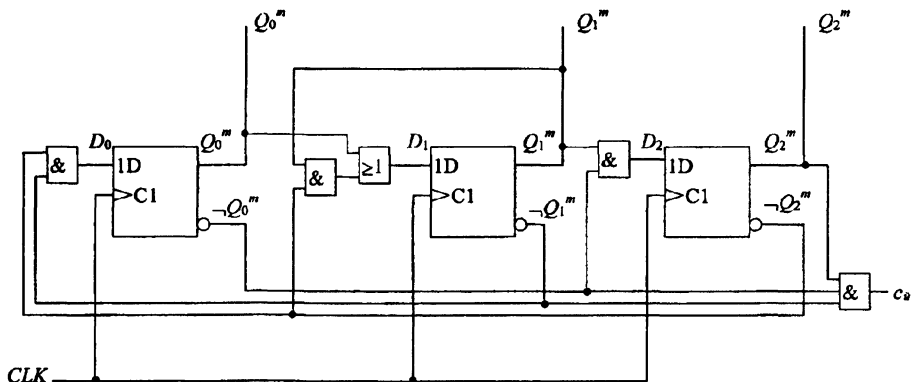
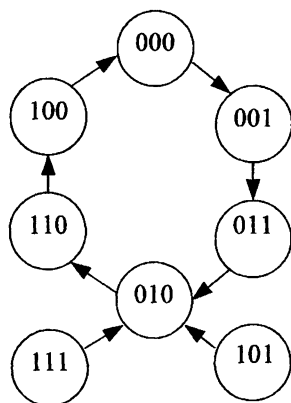


Рис. 10.12. Схема счетчика по модулю 6 с использованием кода Грея

Рис. 10.13. Диаграмма состояний счетчика по модулю 6, использующего код Грея



10.2.3. Синхронный 4-битовый реверсивный двоичный счетчик 74191

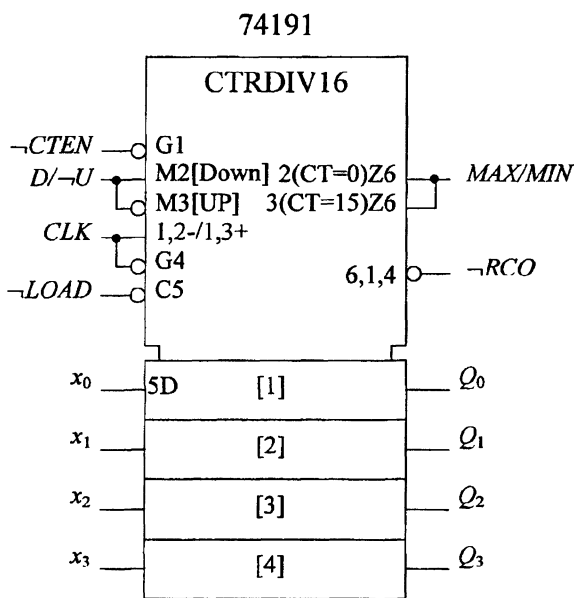


Рис. 10.14. Логический символ 4-битового реверсивного двоичного счетчика 74191

В этом параграфе представлен 4-битовый реверсивный двоичный счетчик (forward-backward counter). Данная схема типична для этого типа счетчиков. Логический символ показан на рис. 10.14.

Обозначение CTRDIV16 (counter dividing by 16) показывает, что счетчик считает до 16. С помощью сигнала $\neg CTEN$ (counter enable) счетчик активируется. С помощью $D/\neg U$ направление счета может быть переключено с прямого на обратное.

С появлением каждого нарастающего фронта тактового сигнала CLK счет продолжается. На логическом символе приведены значки 1, 2– и 1, 3+. Это означает, что тактовый сигнал связан с контактом $\neg CTEN$ (на котором стоит G1) функцией И. Имеется переключение режима работы (зависимость от моды M), в соответствии с которым M2 определено для счета в обратном направлении с входами DI-И. Кроме того, инвертированный тактовый сигнал привязан к контакту $\neg RCO$ через функцию И, что отмечено обозначением G4.

Выход MAX/MIN имеет различные функции при прямом счете (цифра 3) и обратном счете (цифра 2). Итак, вновь имеем зависимость от M. При обратном счете выход MAX/MIN переходит в состояние 1, если состояние счетчика соответствует 0, это обозначается через $CT = 0$.

Соответственно при прямом счете максимальное состояние счетчика обозначается $CT = 15$. Выход MAX/MIN соединен, кроме того, Z-зависимостью (Z6) с выходом MAX/MIN, если одновременно тактовый сигнал $CLK = 0$ (из-за G4) и на контакте — $CTEN = 0$ (из-за G4). Это установлено последовательностью цифр 6, 1, 4 на контакте $\neg RCO$, которая определяет зависимость Z и обе зависимости G. Из этого следует, что $\neg RCO$ является синхронным выходом, в то время как выход MAX/MIN работает асинхронно. Счетчик может быть загружен параллельно через входы $(x_3, x_2, x_1, x_0) = 1000$, счетчик будет подсчитывать 7 ступеней до 1111. После этого он выдает сигнал переноса $\neg RCO = 0$. Если связать этот выход с $\neg LOAD$, счетчик будет загружен до состояния 1000 и цикл начнется снова.

Очевидно, что счетчик может быть использован как делитель частоты на 16, так как он выдает при непрерывном счете через каждые 16 тактовых импульсов сигнал переноса $\neg RCO$.

10.3. Упражнения

Задача 10.1. Сконструируйте асинхронный обратный счетчик, который пробегает последовательность 000, 111, 110, 101, 100, 011, 000 и т.д.

Задача 10.2. Сконструируйте двоичный счетчик, который при $V = 1$ в прямом направлении просчитывает последовательность 000, 001, 010, 011, 100, 000. При $V = 0$ счетчик должен считать ту же последовательность в обратном направлении. Используйте 3 D-триггера.

Задача 10.3. Как должен быть подключен модуль двоичного счетчика 74191, чтобы его можно было использовать как десятичный делитель.

ГЛАВА II

СДВИГОВЫЕ РЕГИСТРЫ

Сдвиговые регистры состоят из цепочки триггеров, в которых передача информации осуществляется как в «пожарной цепочке» ПЗС (bucket-brigade). Они могут быть построены, например, из D-триггеров или из JK-триггеров. На рис. 11.1 показан пример регистра с четырьмя JK-триггерами. Для того, чтобы информация передавалась одновременно по всей цепочке, применяются триггеры с управлением фронтом.

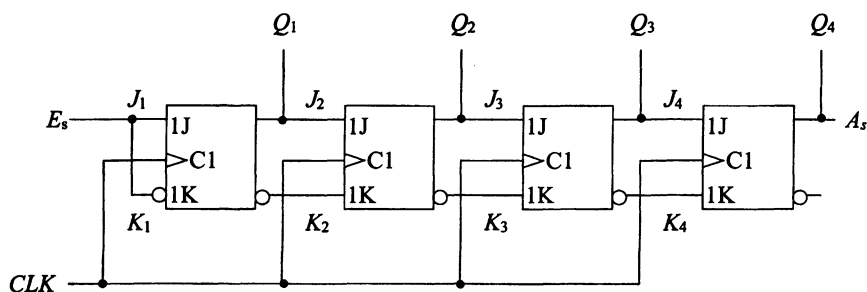


Рис. 11.1. Цепочка сдвигового регистра из четырех JK-триггеров.

Представленный сдвиговый регистр имеет последовательный вход E_s и последовательный выход A_s . Параллельные выходы обозначены как Q_i . Функция этого сдвигового регистра со сдвигом вправо описывается следующими уравнениями:

$$Q_0^{m+1} = E_s^m \quad (11.1)$$

$$Q_i^{m+1} = Q_{i-1}^m \quad \text{при } 0 < i \leq 4 \quad (11.2)$$

$$A_s^m = Q_4^m \quad (11.3)$$

Сдвиговые регистры находят универсальное применение в центральных процессорах (CPU) компьютеров при проведении операций умножения и деления. Они также применяются для последовательно-параллельного и параллельно-последовательного преобразований кодов. Кроме того, они служат как ЗУ на основе «пожарных цепочек» (first-in first-out, FIFO, обратного магазинного типа).

Сдвиговые регистры могут обладать следующими свойствами:

- способностью переключения между сдвигом влево и сдвигом вправо
- наличием параллельных входов для одновременной установки триггеров
- наличием параллельных выходов
- наличием последовательных входов и выходов

11.1. Временные характеристики сдвиговых регистров

В том случае, если применяется триггер с малым расстоянием между действующим интервалом и интервалом переброса, проблематичным становится появление тактового фазового сдвига (clock skew). Подобные регистры образуются, в частности, при использовании буферных триггеров. Фазовый тактовый сдвиг может привести к тому, что при одном тактовом импульсе информация передается на многие ступени либо теряется. Это является следствием перекрытия действующего интервала и интервала переброса из-за влияния тактового фазового сдвига.

Рассмотрим этот случай на примере регистра, состоящего из двух управляемых одним фронтом буферных D-триггеров (рис. 11.2). Второй D-триггер будет запускаться с тактовым фазовым сдвигом t_0 .

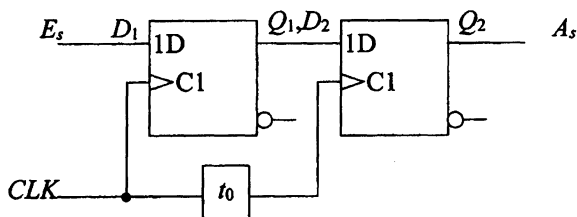


Рис. 11.2. Сдвиговый регистр с D-триггерами управляемыми передним фронтом.

На рис. 11.3 а показаны соотношения без тактового сдвига ($t_0 = 0$). Действующие интервалы и интервалы переброса обоих триггеров занимают одни и те же интервалы времени. Как следует из рисунка, информация правильно передается от одного триггера к следующему.

На рис. 11.3 б тактовый сигнал второго триггера задержан относительно первого триггера на t_0 . Вследствие этого действующий

интервал второго триггера $W2$ сдвигается в интервал переброса первого триггера $K1$ поэтому, то что запоминается во втором триггере, полностью зависит от случая.

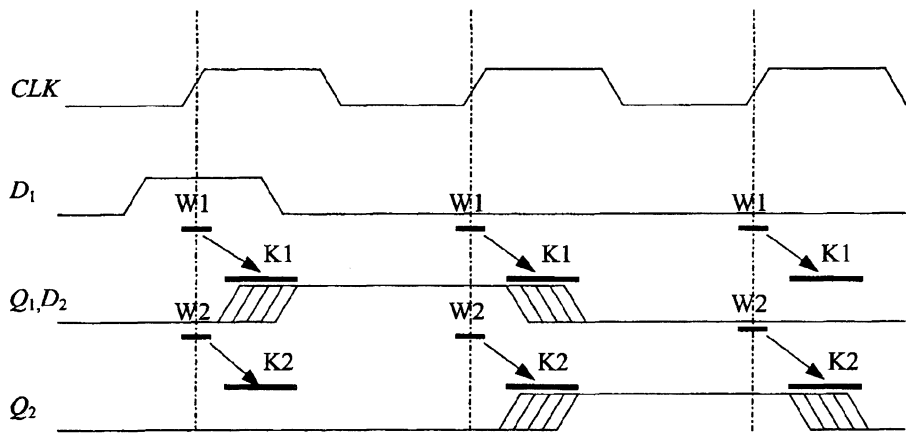


Рис. 11.3 а. Временная диаграмма сдвигового регистра, показанного на рис. 11.2, без тактового сдвига ($t_0 = 0$).

При несколько большем сдвиге t_0 на выходе Q_2 второго триггера появится тот же выходной сигнал, что и на выходе Q_1 первого триггера. Теперь бит «переваливается» без запоминания через второй триггер.

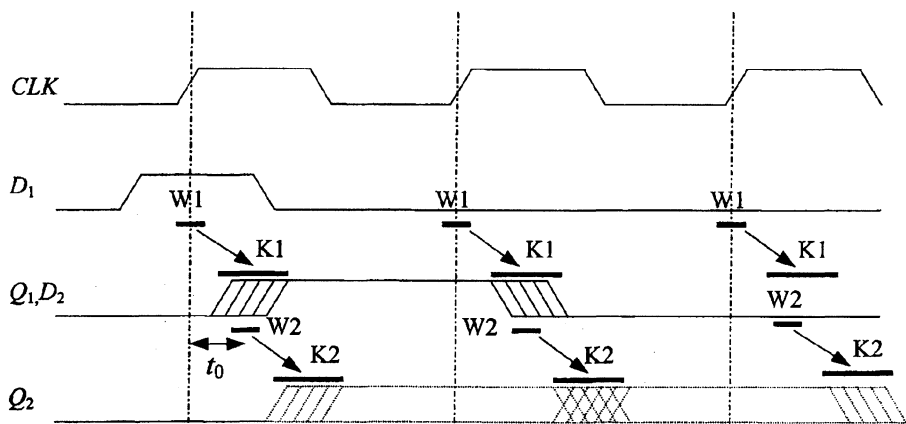


Рис. 11.3 б. Временная диаграмма цепочки сдвигового регистра, показанной на рис. 11.2, при наличии тактового сдвига ($t_0 > 0$).

Если необходимо допустить большой тактовый сдвиг, то часто выбирают триггеры с управлением по двум фронтам. При примене-

нии триггеров с двумя фронтами можно иметь тактовый сдвиг до половины тактового периода.

11.1.1. Сдвиговый регистр 74194

В качестве примера 4-битового сдвигового регистра на рис.11.4 представлен сдвиговый регистр 74194. Он представляет собой управляемый фронтом сдвиговый регистр с многими рабочими состояниями, которые устанавливаются сигналами S_0 и S_1 (табл. 11.1). Рабочие состояния описываются в логическом символе зависимостью от режима.

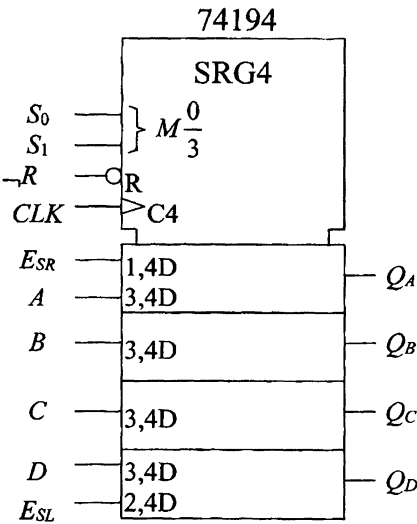


Рис. 11.4. Логический символ 4-битового двунаправленного параллельно загружаемого сдвигового регистра 74194.

Таблица 11.1. Режимы работы сдвигового регистра 74194.

Режим работы	S_0	S_1	CLK	E_{SR}	E_{SL}	A	B	C	D	Q_A	Q_B	Q_C	Q_D
Параллельная загрузка	1	1	↑	d	d	A	B	C	D	A	B	C	D
Сдвиг вправо	0	1	↑	d	1	d	d	d	d	1	Q_A	Q_B	Q_C
			↑	d	0	d	d	d	d	0	Q_A	Q_B	Q_C
Сдвиг влево	1	0	↑	1	d	d	d	d	d	Q_B	Q_C	Q_D	1
			↑	0	d	d	d	d	d	Q_B	Q_C	Q_D	0
Гашение такта	0	0	d	d	d	d	d	d	d	Q_A	Q_B	Q_C	Q_D

Сдвиговый регистр загружается параллельно через входы A , B , C , D . Для $S_0 = 1$ и $S_1 = 1$ в логическом символе использована

цифра 3. С ними устанавливается рабочее состояние «параллельная загрузка». Последовательно снабжены этой цифрой и входы A, B, C, D . Через входы E_{SL} (при сдвиге влево) и E_{SR} (при сдвиге вправо) можно ввести сигнал последовательно. На изображении логического символа сдвиг влево обозначается цифрой 2, поэтому цифра 2 указана у входа E_{SL} . В качестве последовательного выхода могут быть применены Q_A и Q_B , в зависимости от выбора сдвига влево либо сдвига вправо.

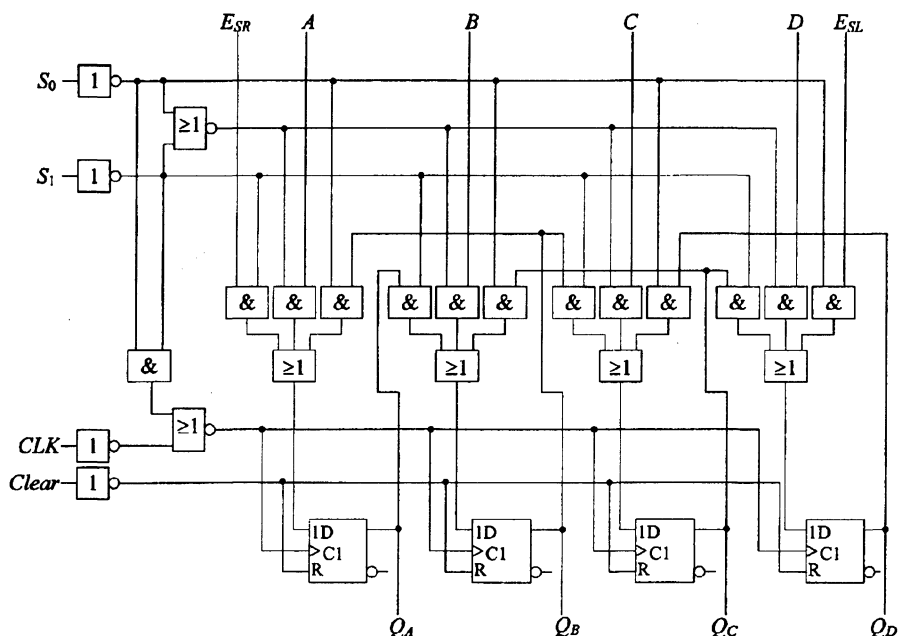


Рис. 11.5. Структурная схема 4-битового двунаправленного параллельно загружаемого сдвигового регистра 74194.

11.2. Сдвиговый регистр с обратной связью

Сдвиговый регистр с обратной связью получают при присоединении отдельных выходов ступеней сдвигового регистра через логическую схему обратно ко входу. На рис. 11.6 показана принципиальная схема данного типа регистра. Функция сдвигового регистра описывается следующими уравнениями:

$$Q_1^{m+1} = f(Q_1^m, Q_2^m, Q_3^m) \quad (11.4)$$

$$Q_2^{m+1} = Q_1^m \quad (11.5)$$

$$Q_3^{m+1} = Q_2^m \quad (11.6)$$

Единственная степень свободы заключается в выборе функции $f(Q_1^m, Q_2^m, Q_3^m)$. Поэтому для каждого состояния возможно появление только двух различных последовательных состояний.

В табл. 11.4 приведена таблица истинности сдвигового регистра с обратной связью и тремя блоками ЗУ, представленного на рис. 11.6. В левом столбце представлено содержание D-триггеров в момент времени t . В момент времени $t + 1$ в первом D-триггере находится значение функции, генерированное логической схемой. Значения Q_1^m и Q_2^m сдвигаются в оба следующих D-триггера 2 и 3.

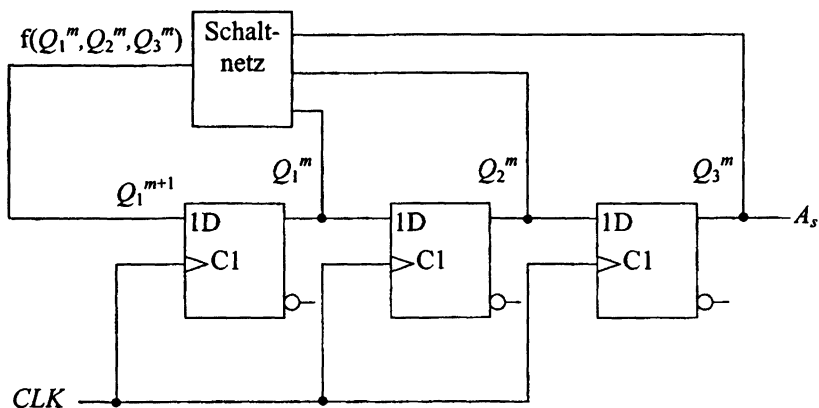


Рис. 11.6. Принципиальная схема сдвигового регистра с обратной связью.

Таблица 11.2. Таблица истинности сдвигового регистра с обратной связью показанного на рис. 11.6.

Q_1^m	Q_2^m	Q_3^m	Q_1^{m+1}	Q_2^{m+1}	Q_3^{m+1}
0	0	0	$f(0, 0, 0)$	0	0
0	0	1	$f(0, 0, 1)$	0	0
0	1	0	$f(0, 1, 0)$	0	1
0	1	1	$f(0, 1, 1)$	0	1
1	0	0	$f(1, 0, 0)$	1	0
1	0	1	$f(1, 0, 1)$	1	0
1	1	0	$f(1, 1, 0)$	1	1
1	1	1	$f(1, 1, 1)$	1	1

Пусть, например, должна генерироваться следующая последовательность содержаний регистра: 000, 100, 101, 001, 000 и т.д. С последовательного выхода A_S можно снять последовательность 000100010001. Следовательно, схема может быть применена как делитель частоты на 4. Альтернативно, различные сочетания содер-

жения регистра можно также интерпретировать как счетные состояния счетчика, который, разумеется, считает с использованием специального кода. На основе этого можно было бы разработать счетчик по модулю 4.

Составим таблицу истинности для реализации этого сдвигового регистра. В этой таблице, показанной на рис. 11.3, учитываются только состояния, проходящие в желательном цикле.

Таблица 11.3. Таблица истинности при генерации последовательности: 000, 100, 010, 001, 000.

Q_1^m	Q_2^m	Q_3^m	Q_1^{m+1}	Q_2^{m+1}	Q_3^{m+1}
0	0	0	1	0	0
1	0	0	0	1	0
0	1	0	0	0	1
0	0	1	0	0	0

И без применения KV -диаграммы из этой таблицы можно считать функцию $f(Q_1^m, Q_2^m, Q_3^m)$.

$$Q_1^{M+1} = f(Q_1^m, Q_2^m, Q_3^m) = \neg Q_1^m \neg Q_2^m \neg Q_3^m \quad (11.7)$$

11.2.1. Счетчик Мебиуса, счетчик Джонсона

Одной из часто применяемых форм сдвигового регистра является счетчик Мебиуса (счетчик Джонсона). В этом счетчике выход через инвертор соединяется с входом. На рис. 11.7 приведен счетчик Джонсона с четырьмя JK-триггерами. Содержание запоминающих ячеек ЗУ задается таблицей истинности, из которой видно, что генерируются числовые последовательности с восемью различными содержаниями счетчика.

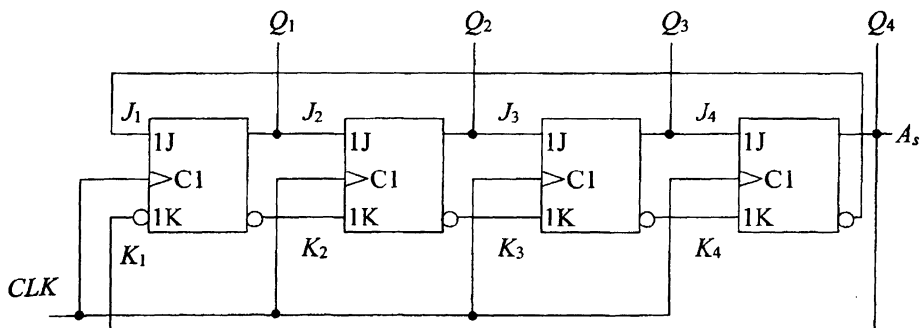


Рис. 11.7. Счетчик Джонсона, состоящий из четырех JK-триггеров.

Логическая схема счетчика Джонсона задается следующей формулой:

$$J_1 = K_1 = f(Q_1^m, Q_2^m, Q_3^m, Q_4^m) = \neg Q_4^m \quad (11.8)$$

При $J_1 = K_1 = 0$ запомненное в триггере значение удерживается, при $J_1 = K_1 = 1$ оно инвертируется. Теперь можно составить таблицу истинности счетчика Джонсона. Констатируем, что имеются два независимых цикла, в зависимости от того, с какого начального состояния стартует счетчик при включении. Но длина обоих циклов одинакова. При желании запустить определенный цикл необходимо предварительно задать соответствующее исходное состояние.

Таблица 11.4. Таблица истинности счетчика Джонсона, показанного на рис. 11.7.

Q_1^m	Q_2^m	Q_3^m	Q_4^m	Q_1^{m+1}	Q_2^{m+1}	Q_3^{m+1}	Q_4^{m+1}	Q_1^m	Q_2^m	Q_3^m	Q_4^m	Q_1^{m+1}	Q_2^{m+1}	Q_3^{m+1}	Q_4^{m+1}
0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	1
1	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0
1	1	0	0	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	0	1	1	1	1	1	0	1	0	1	1	0	1
1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	0
0	1	1	1	0	0	1	1	0	1	1	0	1	0	1	1
0	0	1	1	0	0	0	1	1	0	1	1	0	1	0	1
0	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0

Поведение кольцевого счетчика может быть изображено с помощью диаграммы состояний, приведенной на рис. 11.8.

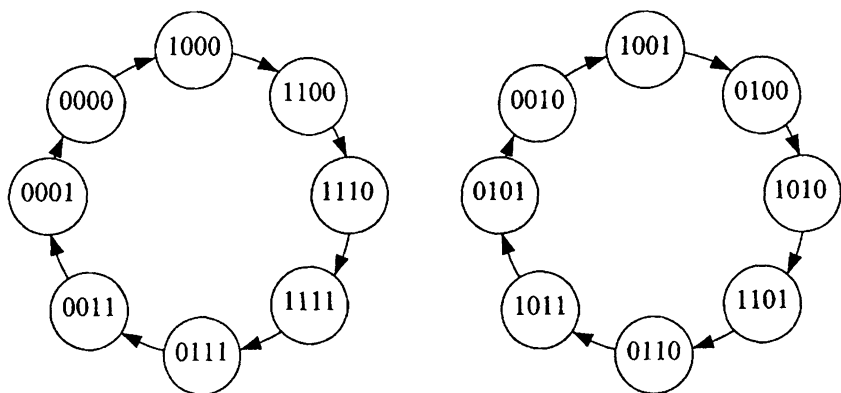


Рис. 11.8. Диаграмма состояний счетчика Джонсона, показанного на рис. 11.7. В кружках стоят $Q_1^m, Q_2^m, Q_3^m, Q_4^m$.

11.2.2. Псевдослучайные последовательности

С помощью сдвигового регистра на последовательном выходе могут генерироваться двоичные числовые последовательности, имеющие такое распределение нулей и единиц, которое почти полностью соответствует случайной двоичной числовой последовательности. Эти числовые последовательности называют псевдослучайными. Псевдослучайные последовательности имеют период и поэтому являются детерминированными. Псевдослучайные последовательности генерируются с помощью обратного подсоединения последовательных выходов цепочки сдвигового регистра через дизъюнктивный вентиль (вентиль с функцией «исключающие ИЛИ», exclusive NOR). На рис. 11.9 приведен пример для $n = 7$.

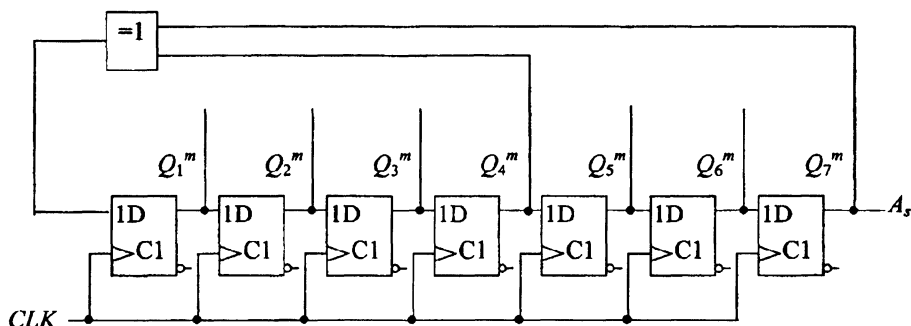


Рис. 11.9. Сдвиговой регистр для генерации псевдослучайной числовой последовательности.

В зависимости от позиций контура обратной связи на последовательном выходе A_s появляются последовательности с различной длиной. Особенно интересны варианты соединений обратной связи, при которых формируются числовые последовательности максимально длинными периодами. Эти числовые последовательности с максимально длинными периодами называют M -последовательностями. Только они имеют псевдослучайные свойства. Период P максимально длинной случайной последовательности, которую можно получить из сдвигового регистра с длиной n -бит, составляет:

$$P = 2^n - 1 \quad (11.9)$$

В процессе генерации псевдослучайных последовательностей цепочка сдвигового регистра содержит все возможные двоичные числа, за исключением 0. Состояние 0 стабильно, но вследствие этого данное состояние не может выступать в качестве начального со-

стояния. Поэтому этого псевдослучайные последовательности обладают свойством, которое заключается в том, что за период 1 появляется более часто, чем 0. Нули и единицы распределены неравномерно. Что является отклонением от идеальной случайной последовательности. Несмотря на это псевдослучайные последовательности хорошо подходят для тестирования информационных каналов. В табл. 11.5 приведены варианты наборов с различным числом контуров обратной связи для количества каскадов $n = 8$. Максимально длинная псевдослучайная последовательность содержит все двоичные числа, соответствующие длине n , за исключением числа 0. Число 0 не должно появляться, так как оно при любом выборе положения контура обратной связи приводит в то же самое состояние.

Таблица 11.5. Количество контуров обратной связи при формировании псевдослучайных последовательностей («-» — нет обратной связи, x — обратная связь).

n	Обратная связь								периоды
	1	2	3	4	5	6	7	8	
2	x	x							3
3	-	x	x						7
4	-	-	x	x					15
5	-	-	x	-	x				31
6	-	-	-	-	x	x			63
7	-	-	-	x	-	-	x		127
8	-	-	-	x	x	x	-	x	255

На рис. 11.10 приведен пример генерации псевдослучайной последовательности при $n = 3$. Сформированная последовательность может быть получена из таблицы последовательности состояний.

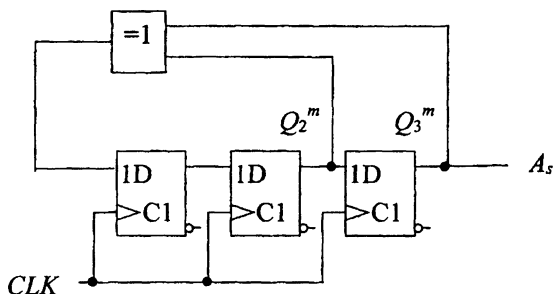


Рис. 11.10. Сдвиговый регистр для генерации максимально длинной псевдослучайной последовательности при $n = 3$.

На последовательном выходе A_s сдвигового регистра, получаем последовательность: 1 1 1 0 0 1 0. Предварительное условие заключается в том, чтобы начальное содержимое сдвигового регистра составляло 1 1 1.

11.3. Упражнения

Задача 11.1. Сконструируйте цепочку сдвигового регистра из D-триггеров, которая выдает на последовательном выходе последовательность:

0 1 0 0 1 1 и т.д.

Сколько D-триггеров вам понадобится?

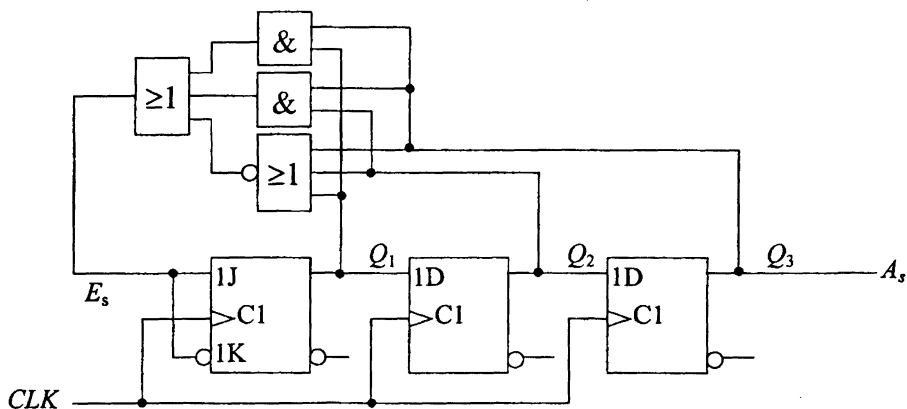
Задача 11.2. Укажите псевдослучайную последовательность, выдаваемую сдвиговым регистром с четырьмя триггерами. Контур обратной связи должны быть проложены таким образом, чтобы последовательность была максимально длинной.

Задача 11.3. Проанализируйте показанный ниже сдвиговый регистр с обратной связью, содержащий JK-триггер и 2 D-триггера.

а) составьте запускающую функцию $E_S = f(Q_1^m, Q_2^m, Q_3^m)$ для первого триггера.

б) сформулируйте следующую из составленной функции таблицу последовательности состояний.

в) начертите диаграмму состояний.



ГЛАВА 12

АРИФМЕТИЧЕСКИЕ УСТРОЙСТВА

12.1. Полный сумматор

В главе 3 уже было приведено определение операции сложения двух двоичных чисел с учетом переноса. Логическая схема, которая производит это сложение, называется полным сумматором. Перенос из предыдущего разряда и оба слагаемых суммируются, после чего выдаются сумма и перенос к следующему разряду. Полный сумматор выполняет переключательные функции для суммирующего выхода F_i и переноса (carry) к следующей ступени C_{i+1} .

$$F_i = \neg c_i \neg x_i y_i \vee \neg c_i x_i \neg y_i \vee c_i \neg x_i \neg y_i \vee c_i x_i y_i = x_i \oplus y_i \oplus c_i \quad (12.1)$$

$$c_{i+1} = x_i y_i \vee c_i (x_i \vee y_i) \quad (12.2)$$

Полному сумматору для выполнения сложения необходимо время, равное утроенному времени задержки вентиля t_p , $t_\Sigma = 3t_p$. Для переноса берутся в расчет два времени задержки: $t_{\bar{u}} = 2t_p$.

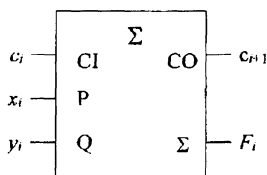


Рис. 12.1. Логический символ полного сумматора, действующего в соответствии с уравнениями (12.1) и (12.2).

12.2. Последовательный сумматор

Если должны быть просуммированы двоичные, например, четырех разрядные числа, то в этом случае слагаемые могут быть подведены к полному сумматору (full adder) с помощью двух сдвиговых регистров. Перенос промежуточно запоминается в ЗУ. После этого результат находится в сдвиговом регистре, соответствующем числу x . Суммирование проводится при каждом тактовом импульсе C .

Время, требуемое для сложения двух m -позиционных двоичных чисел составляет m -кратную величину по отношению ко времени, которое требуется для сложения чисел полному сумматору $t_\Sigma = m3t_p$.

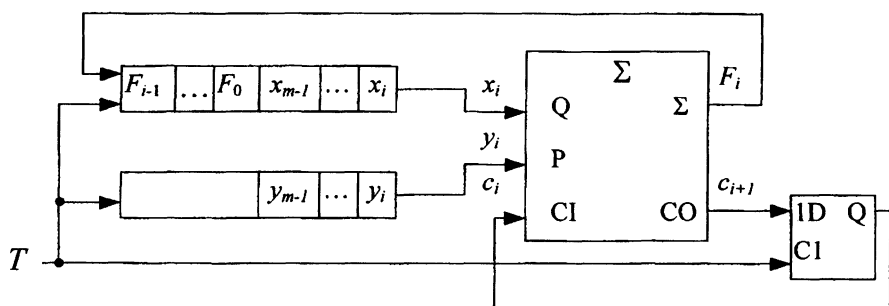


Рис. 12.2. Последовательный сумматор для m разрядов. Осуществляется i -тый шаг.

12.3. Сумматор с последовательным переносом (ripple-carry-adder)

Из m сумматоров можно сконструировать суммирующее устройство для двух m -разрядных двоичных чисел, подсоединив выход переноса к входу переноса следующего полного сумматора (рис. 12.3).

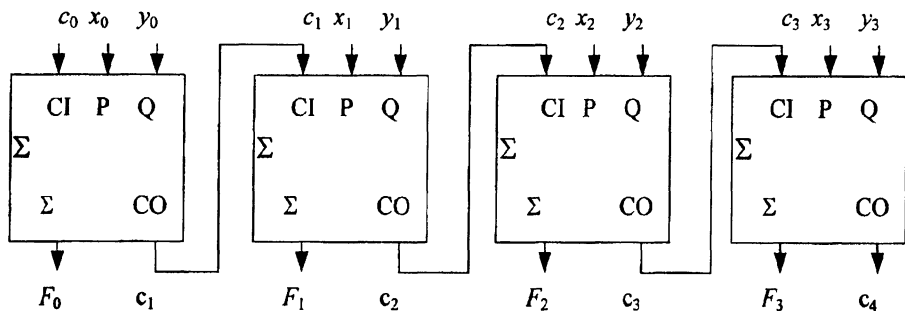


Рис. 12.3. Сумматор с последовательным переносом для 4 бит

Как велико время выполнения суммирования двух m -разрядных чисел? На перенос C_1 требуется только две задержки, поскольку для расчета переноса не нужен инвертор. Итак, суммарное время переноса C_m составляет:

$$t_{\bar{u}} = 2mt_p \quad (12.3)$$

($t_{\bar{u}} \rightarrow \bar{u}$ bergang = перенос)

Последний бит, вычисляемый суммы, который устанавливается, называется самый значимый бит (most significant bit (MSB)). Констатируем, что для операции суммирования требуется умноженное

на $(m - 1)$ время для подсчета переноса плюс время для подсчета самого значимого бита F_{m-1} . MSB суммы F_{m-1} справедливо для времени задержки:

$$t_{\Sigma} = (2(m - 1) + 3) t_p = (2m + 1) t_p \quad (12.4)$$

Это очень большое время выполнения суммирования. Если выбрать для реализации двухкаскадный драйвер (с инверторами), получим оптимальное время суммирования $3t_p$. Но следует учитывать, что таблица истинности при сложении двух 8-позиционных двоичных чисел имеет длину $2^{17} = 131072$ строк (8 бит + 8 бит + 1 бит (C_o) = 17). Реализация подобной логической схемы привела бы к чрезмерным схемным затратам.

12.4. Сумматор с параллельным переносом

Компромисс между схмотехническими затратами и временем задержки представляет сумматор с опережением переноса (carry-look-ahead adder). Он образован схемой, состоящей из упрощенных одно-разрядных сумматоров, которые не содержат схемы для формирования перехода. Разумеется, перенос для отдельных полных сумматоров осуществляется с помощью дополнительной логической схемы. На основе многократного применения уравнения (12.2) для отдельных переносов получаем:

$$c_1 = \underbrace{x_0 y_0}_{g_0} \vee c_0 \underbrace{(x_0 \vee y_0)}_{p_0} = g_0 \vee c_0 p_0 \quad (12.5)$$

$$c_2 = \underbrace{x_1 y_1}_{g_1} \vee c_1 \underbrace{(x_1 \vee y_1)}_{p_1} = g_1 \vee c_1 p_1 = g_1 \vee g_0 p_1 \vee c_0 p_0 p_1 \quad (12.6)$$

$$\begin{aligned} c_3 &= \underbrace{x_2 y_2}_{g_2} \vee c_2 \underbrace{(x_2 \vee y_2)}_{p_2} = g_2 \vee c_2 p_2 = \\ &= g_2 \vee g_1 p_2 \vee g_0 p_1 p_2 \vee c_0 p_0 p_1 p_2 \end{aligned} \quad (12.7)$$

$$\begin{aligned} c_4 &= \underbrace{x_3 y_3}_{g_3} \vee c_3 \underbrace{(x_3 \vee y_3)}_{p_3} = g_3 \vee c_3 p_3 = \\ &= \underbrace{g_3 \vee g_2 p_3 \vee g_1 p_2 p_3 \vee g_0 p_1 p_2 p_3}_G \vee \underbrace{c_0 p_0 p_1 p_2 p_3}_P \end{aligned} \quad (12.8)$$

При этом принято:

$$g_i = x_i y_i \quad (12.9)$$

$$p_i = x_i \vee y_i \quad (12.10)$$

Равенство $g_i = 1$ означает, что в каждом терме C_{i+1} генерируется перенос. В этом случае обе входных переменных соответствующего каскада полного сумматора равны 1 (сравните с уравнением (12.9)). Поэтому g_n называют также «генерирование переноса» (carry generate). В противовес этому p_i воздействует только на один переход, также, если $C_i = 1$, p_i представляет собой 1, когда только одна из входных переменных равна 1 (уравнение (12.10)). P_i называют также «распространение переноса» (carry propagate). На рис. 12.4 показана схема сумматора с опережением переноса. Мы видим, что в начале реализуются функции g_i и p_i . Из них с помощью инвертирующих выражений от 12, 5 до 12, 7 получаем величины от $\neg C_0$ до $\neg C_3$.

Поскольку функции g_n и p_n уже сформулированы, их используют также при расчете сумм F_i . Как можно легко убедиться, справедливо:

$$x_i \leftrightarrow y_i = g_i \leftrightarrow p_i \quad (12.11)$$

Отсюда в соответствии с уравнением (12.1):

$$F_i = x_i \leftrightarrow x_i \leftrightarrow c_i = g_i \leftrightarrow p_i \leftrightarrow c_i \quad (12.12)$$

В схеме используют инвертированную выходную функцию $\neg F_i$, которую получают путем инвертирования C_i .

$$\neg F_i = g_i \leftrightarrow p_i \leftrightarrow \neg c_i \quad (12.13)$$

Поэтому в схеме, приведенной на рис. 12.4 применены еще два вентиля «исключающее ИЛИ», предназначенные для формирования выходных функций. Кроме того, в данном схемном узле генерируется функция $\neg P$ и $\neg G$, из которых затем в соответствии с уравнением (12.8) можно получить перенос C_4 .

Как видно из рисунка, при формировании сумм F_i сигнал проходит 4 вентиля, следовательно:

$$t_\Sigma = 4t_p \quad (12.14)$$

Общее время задержки при расчете $\neg P$ и $\neg G$ составляет:

$$t_p = t_g = 3t_p \quad (12.15)$$

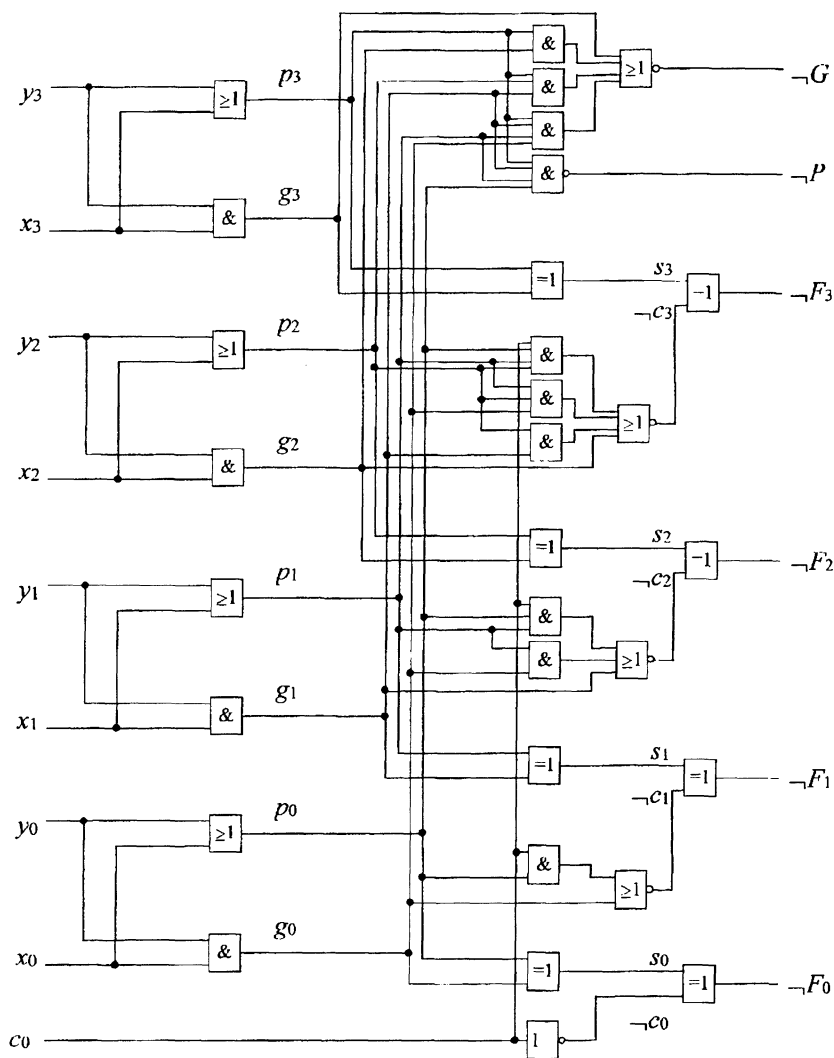


Рис. 12.4. Сумматор с логической схемой для генерации переносов $\neg C_i$ в соответствии с принципом «опараллельного переноса» (carry-look-ahead).

Аппаратные (hardware) затраты, необходимые для расчета переносов при сложении двух m -разрядных чисел, можно определить на основе уравнения (12.5)–(12.8) (как и для m больше 4). Получаем для числа вентилей N_g :

$$N_g = \sum_{i=0}^m i \quad (12.16)$$

и для максимального числа входов вентиля N_e :

$$N_e = m \quad (12.17)$$

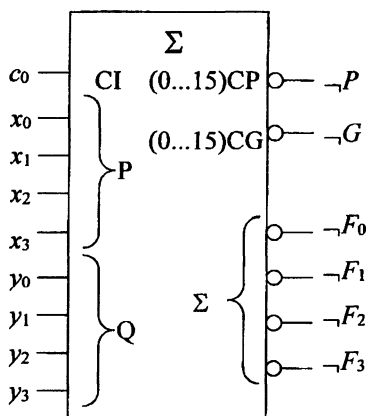


Рис. 12.5. Логический символ сумматора, показанного на рис. 12.4.

12.4.1. Каскадирование сумматоров с параллельным переносом

Так как количество необходимых входов в вентиль и число вентилях при более длинных словах сильно увеличивается, целесообразно формировать в начале блоки из 4-битовых сумматоров с опережением переноса. Этот метод проиллюстрируем в начале на примере сумматора для 16-битовых двоичных чисел, который сформирован из четырех описанных выше 4-битовых сумматоров с параллельным переносом (CLA).

Для 4-битовых сумматоров i ($i = 1 \dots 4$) вспомогательные сигналы $\neg P_i$ и $\neg G_i$ а также C_0 выводятся в соответствии с приведенными ниже уравнениями CLA-генератора. Примененные уравнения соответствуют уравнениям 12.5-12.8. На основе этого генерируются переносы C_4 , C_8 и C_{12} для отдельных блоков. Эта схема имеет обозначение 74182.

$$c_4 = G_0 \vee c_0 P_0 = \neg (\neg G_0 \neg P_0 \vee \neg c_n G_0) \quad (12.18)$$

$$\begin{aligned} c_8 &= G_1 \vee c_4 P_1 = G_1 \vee G_0 P_1 \vee c_0 P_0 P_1 = \\ &= \neg (\neg G_1 \neg P_1 \vee \neg G_0 \neg G_1 P_0 \vee \neg c_n \neg G_0 \neg G_1) \end{aligned} \quad (12.19)$$

$$\begin{aligned}
 c_{12} &= G_2 \vee c_8 P_2 = G_2 \vee G_1 P_2 \vee G_0 P_1 P_2 \vee c_0 P_0 P_1 P_2 = \\
 &= \neg(\neg G_2 \neg P_2 \vee \neg G_1 \neg G_2 \neg P_1 \vee \\
 &\quad \vee \neg G_0 \neg G_1 \neg G_2 \neg P_0 \vee \neg c_n \neg G_0 \neg G_1 \neg G_2)
 \end{aligned} \quad (12.20)$$

$$\begin{aligned}
 c_{16} &= G_3 + c_{12} P_3 = \\
 &= \underbrace{G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3}_G + c_0 \underbrace{P_0 P_1 P_2 P_2}_P
 \end{aligned} \quad (12.21)$$

$$\neg P = \neg P_0 \vee \neg P_1 \vee \neg P_2 \vee \neg P_3 \quad (12.22)$$

$$\begin{aligned}
 \neg G &= \neg G_3 \neg P_3 \vee \neg G_2 \neg G_3 \neg P_2 \vee \neg G_1 \neg G_2 \neg G_3 \neg P_1 \vee \\
 &\quad \vee \neg G_0 \neg G_1 \neg G_2 \neg G_3
 \end{aligned} \quad (12.23)$$

В схеме 74182 C_{16} не генерируется, но вместо него генерируются сигналы $\neg G$ (блок генерации) и $\neg P$ (блок распространения), из которых затем с помощью двух вентилях (в соответствии с уравнением (12.21)) можно сформировать C_{16} .

На рис. 12.6 показана полная схема генератора с опережением переноса, содержащегося в схеме 74182. Входы P_i и G_i и выходы P и G инвертируются, чтобы обеспечить совместимость с сумматором, показанным на рис. 12.4.

Логический символ схемы 74182 можно найти на рис. 12.7. Схему генератора с опережением переноса на 16 бит можно составить из 4 арифметико-логических устройств (АЛУ, ALU) показанных на рис. 12.4, а одной схемы 74182. Схема генератора показана на рис. 12.8.

12.4.2. Сравнение сумматоров

В табл. 12.1 сравниваются 3 сумматора по их величинам времени задержки. Можно видеть, что при более высоких аппаратурных затратах сумматоры CLA-типа обладает большим быстродействием.

Таблица 12.1. Время задержки $/t_p$.

	4 бит	16 бит	64 бит
Последовательный сумматор	12	48	192
Сумматор с последовательным переносом	9	33	129
Сумматор CLA-типа	4	8	12

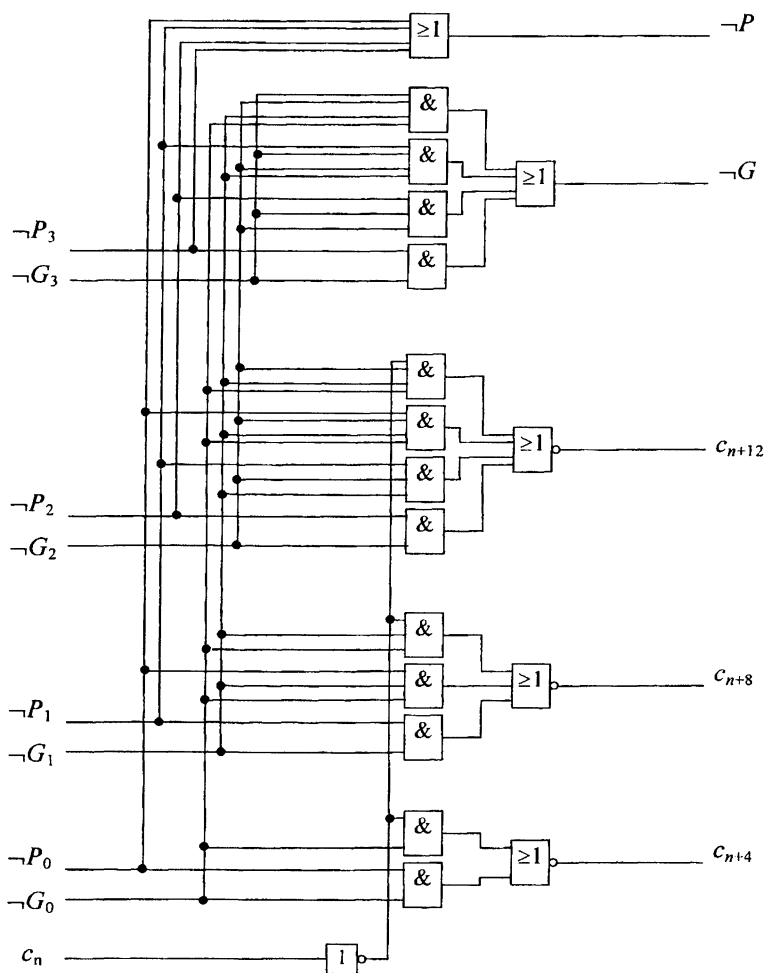


Рис. 12.6. Генератор с параллельным переносом 7418.

12.5. Арифметико-логические вычислительные устройства (ALU, АЛУ)

Арифметико-логические устройства (англ., arithmetic logic unit=ALU) должны иметь возможность наряду со сложением и вычитанием осуществлять логические функции. Как правило, эти схемы содержат сумматор для двух слагаемых. Кроме того, они в состоянии проводить с помощью специальной схемы такие логические операции с двумя операндами, как, например, побитовые И-соединения между двумя операндами.

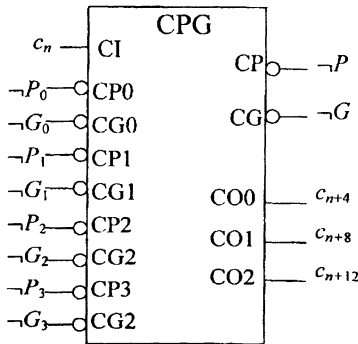


Рис. 12.7. Логический символ генератора с последовательным переносом 74182.

В качестве примера представлено 4-битовое АЛУ 74181. Оно проводит, в зависимости от 4-битового управляющего слова S , различные операции. С помощью выхода M может производиться переключение от логических к арифметическим операциям.

Схему ALU 74181 можно рассматривать как расширение схемы сумматора, приведенной на рис. 12.4. Для выполнения различных операций создаются вместо сигналов p_i и g_i сигналы p'_i и g'_i в соответствии со следующими формулами:

$$g'_i = \neg (\neg x_i \vee s_0 \neg y_i \vee s_1 y_i) \quad (12.24)$$

$$p'_i = \neg (s_3 \neg x_i \neg y_i \vee s_2 \neg x_i y_i) \quad (12.25)$$

Для управляющего слова $S = (1, 0, 0, 1)$ имеем $p'_i = p_i$ и $g'_i = g_i$. Схема формирования показана на рис. 12.9. Она применяется вместо схемы формирования p_i и g_i , приведенной на рис. 12.4.

Схема ALU 74181 показана на рис. 12.10. Мы видим, суммирование может быть реализовано с помощью логической схемы «исключающее ИЛИ» (EXOR) в соответствии с уравнением (12.13). После сокращения получаем t_i :

$$t_i = g'_i \oplus p'_i = \neg (s_3 \neg x_i \neg y_i \vee s_2 \neg x_i y_i \vee \neg s_1 x_i y_i \vee \neg s_0 x_i \neg y_i) \quad (12.26)$$

С помощью других управляющих слов S могут быть образованы другие функции, как это будет показано ниже.

Как обрабатываются переносы C_i ? Сначала отметим, что переносы $\neg C_i$ требуются только для арифметических операций. Поэтому вводят вход M , который для проведения арифметических операций должен быть установлен в состояние 0. При работе с логикой с опережением переноса из уравнений (12.5)–(12.7) получаем с учетом M

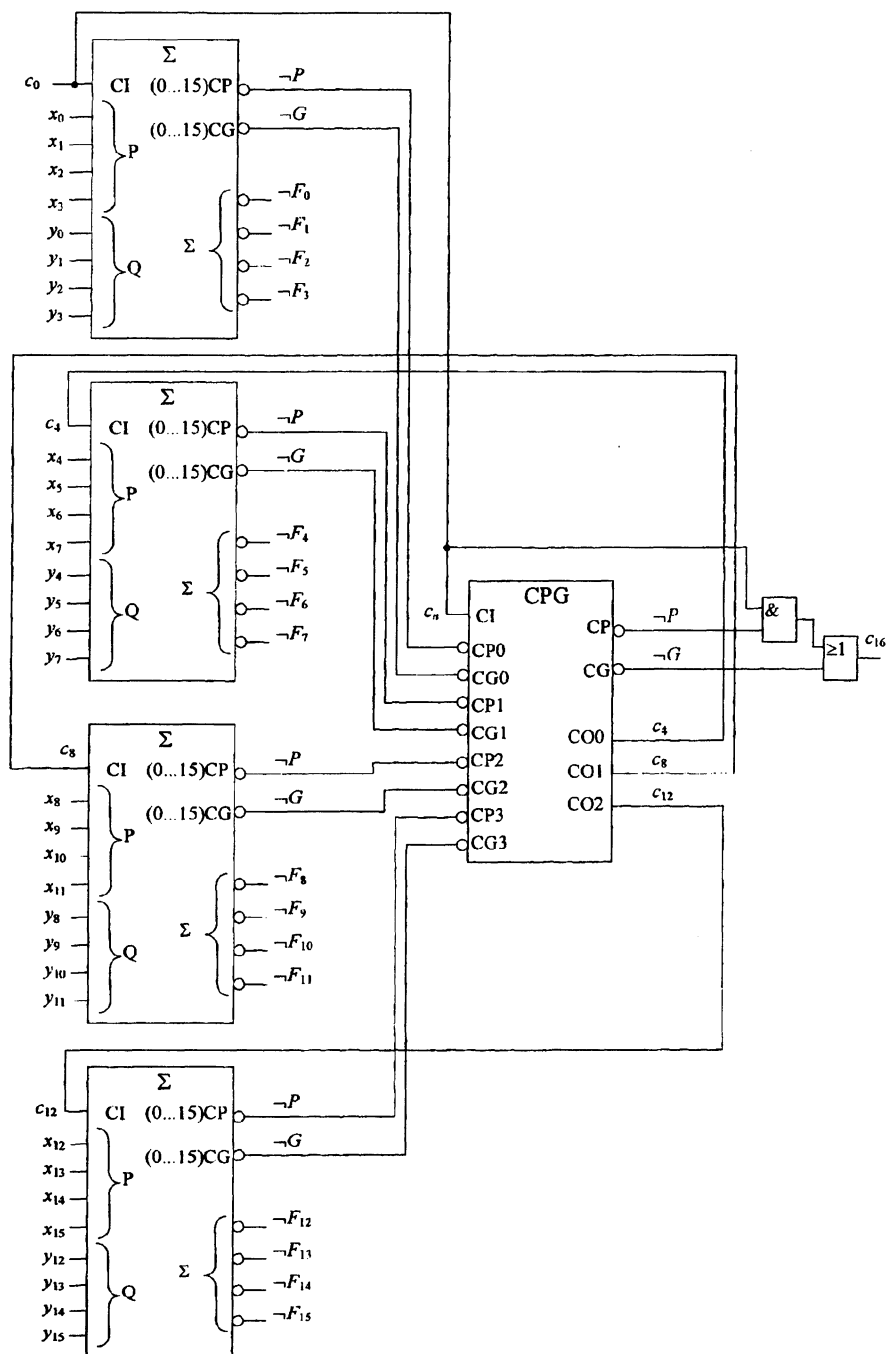


Рис. 12.8. Сумматор типа CLA («с последовательным переносом») на 16 бит, состоящий из 4-битовых сумматоров и одного CLA-генератора.

следующие уравнения для сигналов от U_0 до U_3 . В случае $M = 0$ U_i заменяют сигналы переноса C_i :

$$u_0 = \neg (\neg M c_0) \quad (12.27)$$

$$u_1 = \neg (\neg M g'_0 \vee \neg M p'_0 c_0) \quad (12.28)$$

$$u_2 = \neg (\neg M g'_1 \vee \neg M g'_0 p'_1 \vee \neg M p'_0 p'_1 c_0) \quad (12.29)$$

$$u_3 = \neg (\neg M g'_2 \vee \neg M g'_1 p'_2 \vee \neg M g'_0 p'_0 p'_1 \vee \neg M p'_0 p'_1 p'_2 c_0) \quad (12.30)$$

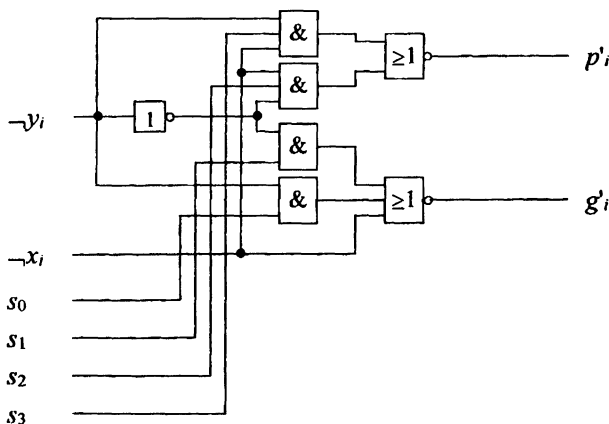


Рис. 12.9. Логическая схема для генерации 16 различных функций.

В случае $M = 0$ (арифметические функции) эти уравнения идентичны инвертированным уравнениям (12.5)–(12.7). Для $M = 1$ все $U_i = 1$.

Как можно видеть из рис.12-4, при $\neg C_i$ с помощью логической функции «исключающее ИЛИ» (EXOR) формируются выходные функции:

$$\neg F_i = u_i \leftrightarrow t_i \quad (12.31)$$

Сигнал переноса и сигналы генерации блока и распространения блока создаются в соответствии с уравнением (12.8):

$$\neg G = g'_3 \vee g'_2 p'_3 \vee g'_1 p'_2 p'_3 \vee g'_0 p'_1 p'_2 p'_3 \quad (12.32)$$

$$\neg P = p'_0 p'_1 p'_2 p'_3 \quad (12.33)$$

$$c_4 = G \vee p'_0 p'_1 p'_2 p'_3 c_0 \quad (12.34)$$

12.5.1. Примеры операций

Сложение

Управляющее слово при проведении сложения имеет вид: $S = (1, 0, 0, 1)$. При этом получаем:

$$g'_i = \neg (\neg x_i \vee \neg y_i) = x_i y_i = g_i \quad (12.35)$$

$$p'_i = \neg (\neg x_i \neg y_i) = x_i \vee y_i = p_i \quad (12.36)$$

$$t_i = g_i \leftrightarrow p_i \quad (12.37)$$

Так как $M = 0$, действительно:

$$u_0 = \neg c_0 \quad (12.38)$$

$$u_1 = \neg (g_0 \vee p_0 c_0) \quad (12.39)$$

$$u_2 = \neg (g_1 \vee g_0 p_1 \vee p_0 p_1 c_0) \quad (12.40)$$

$$u_3 = \neg (g_2 \vee g_1 p_2 \vee g_0 p_0 p_1 \vee p_0 p_1 p_2 c_0) \quad (12.41)$$

Отсюда справедливо $U_i = \neg C_i$, и, как следует из схемы, сумма будет рассчитываться следующим образом:

$$\neg F_i = u_i \leftrightarrow t_i = \neg c_i \leftrightarrow x_i \leftrightarrow y_i = \neg (c_i z \leftrightarrow x_i \leftrightarrow y_i) \quad (12.42)$$

Вычитание (subtraction)

Управляющее слово для вычитания имеет вид: $S = (0, 1, 1, 0)$. При этом определяем для g'_i и p'_i :

$$g'_i = \neg (\neg x_i \vee y_i) = x_i \neg y_i \quad (12.43)$$

$$p'_i = \neg (\neg x_i y_i) = x_i \vee \neg y_i \quad (12.44)$$

Видим, что в противоположность уравнениям (12.35) и (12.36) y_i оказывается инвертированным. Если установить $C_0 = 1$, что вызывает суммирование с 1, получим двойное дополнение (точное дополнение) для y . Отсюда получаем как результат разность $x - y$ при представлении с двойным дополнением.

Отрицание (negation)

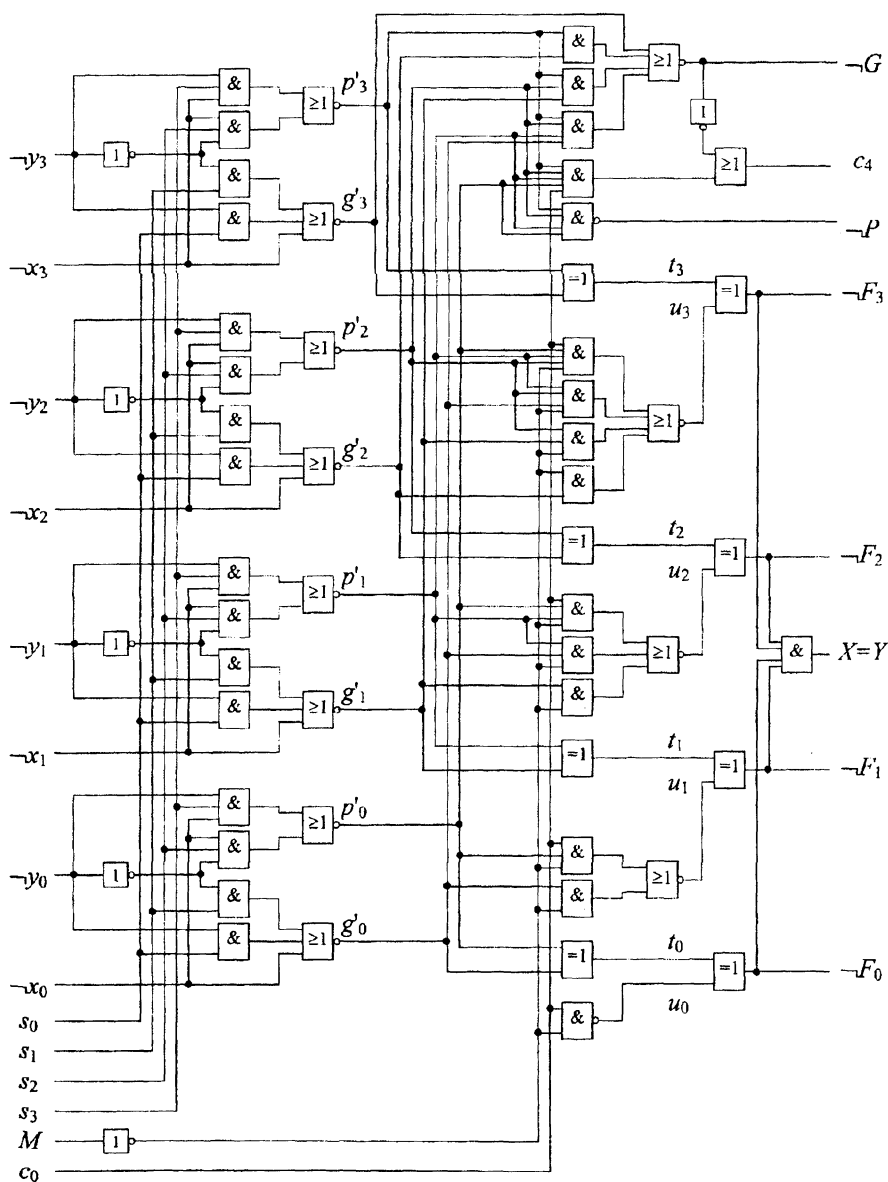
Управляющим словом является $S = (0, 0, 0, 0)$ и $M = 1$.

На основе этого получим в соответствии с уравнением (12.26):

$$t_i = \neg (x_i y_i \vee x_i \neg y_i) = \neg x_i \quad (12.45)$$

Так как $M = 1$, получим $U_i = 1$, и на выходе появятся в соответствии с уравнением (12.31) инвертированные x_i :

$$\neg F_i = u_i \leftrightarrow t_i = \neg t_i = \neg (\neg x_i) \quad (12.46)$$



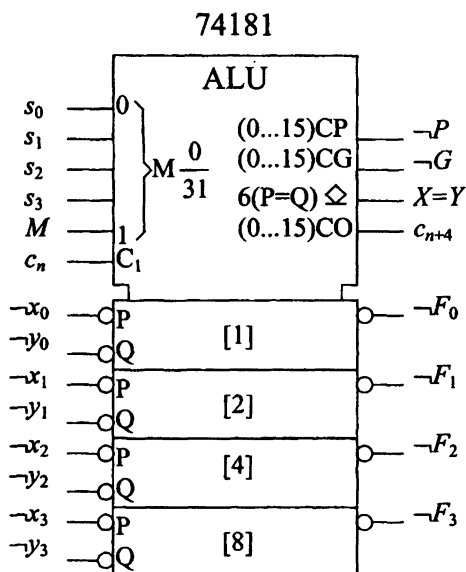


Рис. 12.11. Логический символ ALU 74181.

Дизъюнкция (disjunction)

Управляющим словом является $S = (1, 0, 1, 1)$ и $M = 1$.

На основе этого получим в соответствии с уравнением (12.26):

$$t_i = \neg (\neg x_i \neg y_i) = x_i \vee y_i \quad (12.47)$$

Так как $M = 1$, получим $U_i = 1$, и на выходе появится в соответствии с уравнением (12.31) дизъюнкция x_i и y_i :

$$\neg F_i = u_i \Leftrightarrow t_i = \neg t_i = \neg (x_i \vee y_i) \quad (12.48)$$

Конъюнкция (conjunction)

Управляющим словом является $S = (1, 1, 1, 0)$ и $M = 1$.

На основе этого получим в соответствии с уравнением (12.26):

$$t_i = \neg (\neg x_i \neg y_i \vee x_i y_i \vee x_i \neg y_i) = \neg (\neg x_i \vee \neg y_i) = x_i y_i \quad (12.49)$$

Так как $M = 1$, получим $U_i = 1$, и на выходе появится в соответствии с уравнением (12.31) конъюнкция x_i и y_i :

$$\neg F_i = u_i \Leftrightarrow t_i = \neg t_i = \neg (x_i y_i) \quad (12.50)$$

Могут быть образованы в совокупности 32 различных функции, некоторые из которых имеют очень малое значение. В табл. 12.2 собраны только те функции, которые можно получить с помощью ALU 74181.

Таблица 12.2. Функции ALU 74181.

Выбранные входы				Выходы F_i		
s_3	s_2	s_1	s_0	$M = 1$ логические функции	$M = 0$ арифметические функции	
					$c_0 = 0$	$c_0 = 1$
0	0	0	0	$\neg x$	$x - 1$	x
0	0	0	1	$\neg(xy)$	$(xy) - 1$	xy
0	0	1	0	$\neg x \vee y$	$(x \neg y) - 1$	$x \neg y$
0	0	1	1	1	-1	0
0	1	0	0	$\neg(x \vee y)$	$x + (x \vee \neg y)$	$x + (x \vee \neg y) + 1$
0	1	0	1	$\neg y$	$xy + (x \vee \neg y)$	$xy + (x \vee \neg y) + 1$
0	1	1	0	$\neg(x \nleftrightarrow y)$	$x - y - 1$	$x - y$
0	1	1	1	$x \vee \neg y$	$x \vee \neg y$	$x \vee \neg y + 1$
1	0	0	0	$\neg xy$	$x + (x \vee y)$	$x + (x \vee y) + 1$
1	0	0	1	$x \nleftrightarrow y$	$x + y$	$x + y + 1$
1	0	1	0	y	$(x \neg y) + (x \vee y)$	$(x \neg y) + (x \vee y) + 1$
1	0	1	1	$x \vee y$	$x \vee y$	$x \vee y + 1$
1	1	0	0	0	$x + x$	$x + x + 1$
1	1	0	1	$x \neg y$	$xy + x$	$xy + x + 1$
1	1	1	0	xy	$x \neg y + x$	$x \neg y + x + 1$
1	1	1	1	x	x	$x + 1$

Когда требуется большая длина слов, могут быть совместно соединены с помощью генератора с опережением переноса 74182 по четыре схемы. Возможно дальнейшее каскадирование.

12.6. Компараторы

Как правило, компараторы сравнивают два слова одинаковой длины, показывая при этом, какое число больше. Компараторы вводят, например, в компьютеры, чтобы тестировать условия перехода.

Реализация компараторов, как правило, требует очень больших схемотехнических затрат, которые также, как и для сумматоров, увеличиваются в сильной степени при возрастании числа разрядов при заданном времени задержки. Поэтому при большой длине слов используются каскадируемые компараторы.

12.6.1. 2-битовый компаратор

Пусть в качестве примера должен быть разработан компаратор для двух 2-битовых двоичных чисел x и y , который проверял бы равенство ($x = y$) и имел бы по одному выходу $x > y$ и $x < y$. Для этого сначала составим таблицу истинности (табл. 12.3).

С помощью минимизации получаем:

$$A_{x < y} = \neg x_1 y_1 \vee \neg x_1 x_0 y_0 \vee \neg x_0 y_1 y_0 \vee \neg x_0 y_1 y_0 \quad (12.51)$$

$$A_{x > y} = x_1 \neg y_1 \vee x_0 \neg y_1 \neg y_0 \vee x_1 x_0 \neg y_1 \quad (12.52)$$

Выходной сигнал $A_{x=y}$ можно определить исходя из того факта, что он равен 1 тогда, когда оба других выходных сигнала равны 0:

$$A_{x=y} = \neg A_{x < y} \neg A_{x > y} = \neg (A_{x < y} \vee A_{x > y}) \quad (12.53)$$

Благодаря этому время задержки для этого входа увеличивается на время задержки одного вентиля. На рис. 12.12 показана схема этого компаратора.

Таблица 12.3. Таблица истинности для 2-битового компаратора.

y_1	y_0	x_1	x_0	$x = y$	$x < y$	$x > y$
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

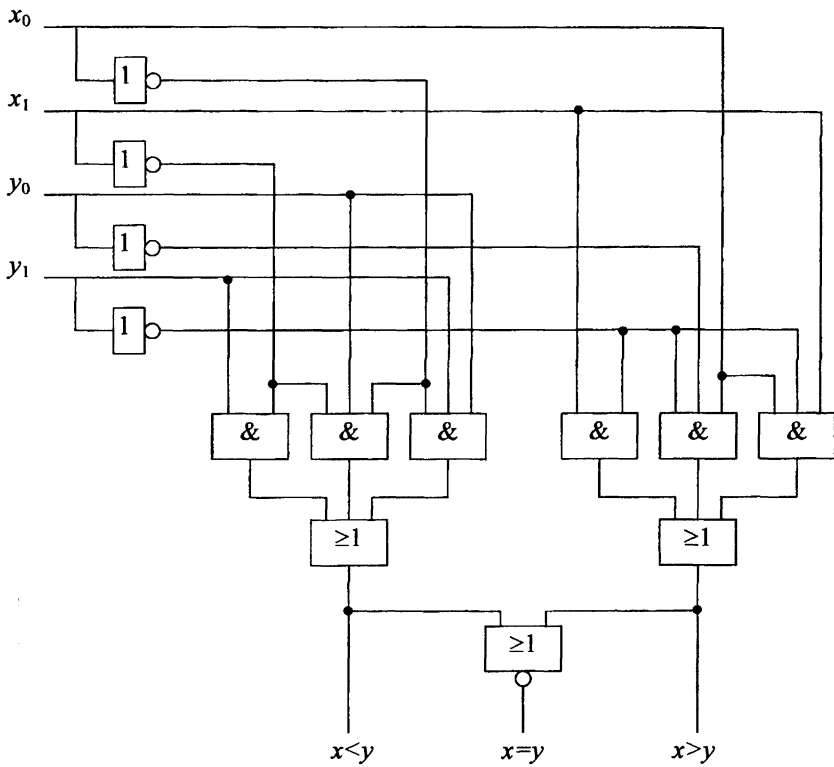


Рис. 12.12. Структурная схема 2-битового компаратора.

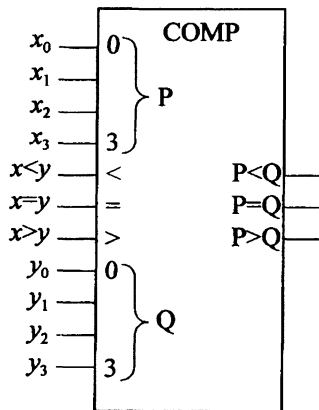


Рис. 12.13. Логический символ 4-битового компаратора 7485.

12.6.2. Каскадируемые компараторы

На рис. 12.13 показан логический символ 4-битового компаратора 7485, который имеет выходы для равенства ($x = y$), для $x > y$ и для $x < y$. Дополнительно данная схема имеет три входа, на которые передаются переносы аналогичных схем, значимость разрядов которых ниже.

На основе трех подобных схем можно построить 12-битовый компаратор, который представлен на рис. 12.14. У схемы с наименьшей значимостью разряда на вход переноса, соответствующий равенству подается 1, в то время как на входы «больше» и «меньше» подается 0. Обе схемы с большей значимостью принимают на входы выходные сигналы предыдущих схем. Следует учитывать, что сигнал переноса «гофрируется». Поэтому время выполнения растет линейно при увеличении количества схем.

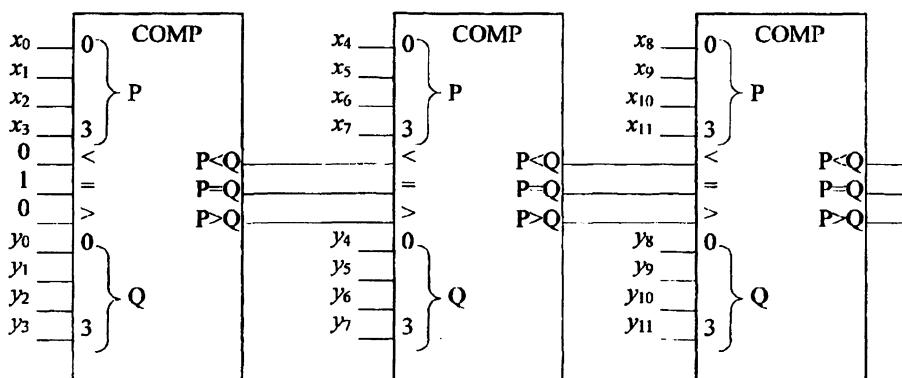


Рис. 12.14. Схема 12-битового компаратора с тремя схема 7485.

12.7. Упражнения

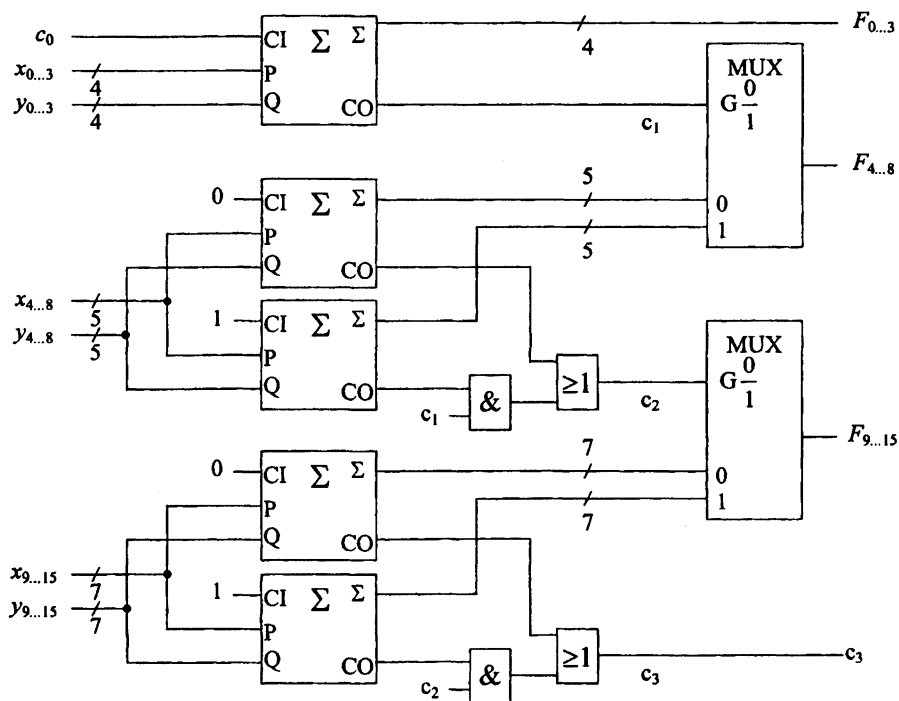
Задача 12.1. Как можно применить 74181 как компаратор для двух 4-битовых слов?

Задача 12.2. Выведите функцию 74181 на основе содержания введенных уравнений для управляющего слова $M = 1$ и $S = 0110$.

Задача 12.3. Альтернативой сумматора с опережением переноса является сумматор с селекцией переноса. На рисунке показан вариант для 16 бит. Пять показанных на структурной схеме сумматоров являются сумматорами с переносом пульсаций с шириной 4, 5 и 7 бит в 1-й, 2-й и 3-й ступени (каскаде).

Объясните функцию схемы.

Задайте величины задержки выходных сигналов как величины, кратные времени задержки вентиля t_p (время задержки мультиплексора $2t_p$).



ГЛАВА 13

ЦИФРОВЫЕ ЗУ

Блоки ЗУ служат для запоминания больших массивов данных. Они становятся важнейшей составляющей частью цифровых компьютеров. Различают следующие блоки ЗУ и способы их организации:

Полупроводниковые ЗУ и ЗУ массового применения

Полупроводниковые ЗУ реализуются на полупроводниковом чипе. Массовые ЗУ имеют высокую плотность ячеек памяти, следовательно, они могут запоминать большое количество данных в малом объеме. Примерами ЗУ массового применения являются жесткие диски, CD-ROM и магнитные ленты. Здесь они не рассматриваются.

Последовательный и произвольный доступ

Последовательный доступ означает, что данные вводятся и считываются только через один порт. Под этим понимаются ЗУ типа «пожарная цепочка», которые работают как сдвиговые регистры. Большей частью они организованы по принципу FIFO (first in, first out). Произвольный доступ говорит о том, что каждая запоминающая ячейка доступна независимо от других.

ЗУ с адресацией по местоположению и ЗУ с адресацией по содержанию (ассоциативное ЗУ)

ЗУ с адресацией по местоположению имеют адрес, делающий доступной любую ячейку ЗУ. При адресации по содержанию ЗУ информацию отыскивают по ассоциации с частью самой информации. Например, в списке поставок номер заказа может быть использован для поиска информации о данном товаре.

Энергозависимое и энергонезависимое ЗУ

Энергозависимые ЗУ теряют информацию при выключении напряжения питания, тогда как энергонезависимые ЗУ сохраняют ее.

Постоянные ЗУ и ЗУ с оперативной записью и считыванием

Постоянные ЗУ однажды программируются и с этого момента могут только читаться. Они энергонезависимы. В ЗУ с оперативной записью и считыванием можно многократно проводить запись и считывание.

ЗУ с побитовой и пословной организацией

При побитовой организации ЗУ каждый бит доступен в отдельности. При пословной организации всегда одновременно считывается или записывается целое слово.

Информационная емкость ЗУ определяется как произведение количества слов ЗУ на их длину (разрядность).

13.1. Принципиальная структурная схема ЗУ

Схемы ЗУ подключаются как правило к шинной системе. Благодаря этому большое число различных ЗУ может быть подключено параллельно. Система шин должна давать возможность записывать в определенную ячейку ЗУ или считывать из нее. При этом различают:

Адресные шины

Адресная шина подключается к каждому узлу ЗУ адресов, под которыми могут храниться данные или проводиться их поиск.

Управляющие шины

Управляющая шина содержит все линии, необходимые для управления схемным узлом. К ним относится линия, обеспечивающая выбор узла и содержащая вывод для выбора чипа (chip select, CS). Поскольку все узлы подключены к одной и той же адресной шине, то с помощью этой линии производится выбор соответствующего узла. С помощью линии считывание-запись можно осуществлять переключение между чтением и записью (read-write, RD-W).

Шины данных

Шина данных подключается ко всем схемным узлам. Чтобы в линиях не создавались конфликты, выходы блоков ЗУ, влияющие на шину данных, должны быть выходами типа tristate («с тремя состояниями»). Их делают свободными для включения с помощью линии output enable (OE, «разрешающий выход»), входящей в управляющую шину.

13.2. ROM

ROM является сокращением для read only memory (ЗУ только с чтением, постоянное ЗУ, ПЗУ).

ROM является схемой ЗУ, содержание данных в котором определяется изготовителем с помощью маски. Поэтому содержание данных фиксировано и их можно только читать. Запомненные данные являются энергозависимыми.

ROM-схемы различаются числом битов, которые запомнены. Обычно используются ЗУ с длинами слов 1, 4, 8 и 16 бит.

В ROM, как и в большинстве других цифровых ЗУ, отдельные ячейки ЗУ расположены в виде матрицы (рис. 13.1). Ячейки ЗУ лежат в точках пересечений линий. К ним обращаются, подав напряжения 1 на строчную линию и на линию столбцов. Строчную линию называют числовой шиной (word line), линию столбцов называют шиной данных (data line). Преимущество данного устройства заключается в экономии шин. Для n^2 ячеек требуется $2n$ шин против n^2 в линейных устройствах.

Распределение числовых шин (word line) по адресам от A_0 до A_2 осуществляется строчным декодером. Столбцевой декодер осуществляет выбор шин данных. Поскольку шина данных наряду с выбором столбца осуществляет вывод накопленной информации к выходу, между декодером и матрицей ЗУ включается усилитель считывания. На рисунке показано ЗУ емкостью 8×8 .

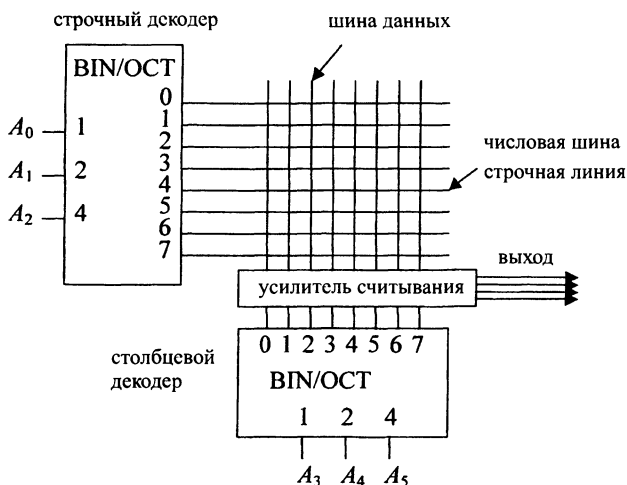


Рис. 13.1. Принципиальная структурная схема ROM с емкостью 8×8 .

Реализация ROM с емкостью 4×4 бит на основе КМОП технологии показана на рис. 13.2. Матрица ЗУ состоит из 16 n -канальных МОП ПТ. Если в ячейке ЗУ должна быть запомнена величина Н (то есть High), то контакта к стоку не должно быть. Технологически это может быть достигнуто с помощью одной единственной маски, которая, в зависимости от того, должен быть транзистор подсоединен или нет, создает соединение с транзистором на основе проводящей дорожки либо нет.

Сопротивлением нагрузки для шин данных служит p -канальный МОП ПТ. Столбцевой декодер все время подключает с помощью проходного транзистора к выходу одну из шин данных.

Если с помощью строчного декодера выбирается какая-либо строка, переводя при этом соответствующий выход строчного декодера в состояние Н, то в этом случае шины данных, с которыми контактируют МОП ПТ, подтягиваются к низкому уровню напряжения L . В прочих случаях они остаются на V_{DD} . К выходу подключается только шина данных, проходной транзистор которой включен декодером данных.

Можно видеть, что для ячейки ЗУ необходим только один транзистор, что обуславливает высокую плотность ячеек в ЗУ.

Логический символ ROM с емкостью $1K \times 8$ бит показан на рис. 13.3. Наряду с адресными входами данный схемный узел имеет вход выбора чипов ($\neg CS$) и вход разрешения выхода ($\neg OE$). Вход $\neg CS$ служит для выбора ROM, когда к одной шине подключено большое количество ROM. Если в дополнении к этому вход $\neg OE$ находится в состоянии L , выход будет низкоомным.

В символе ROM используется адресная зависимость обозначенная буквой A . Фигурная скобка охватывает адресные входы. Дробь после A дает в числителе самый низкий и в знаменателе самый высокий адреса ЗУ. На входах машинного слова шириной 8 бит опять приведена буква A в знак того, что выходы всегда подсоединены к выбранной адресом ячейки. Выходы выполнены по типу tristate («с тремя состояниями»).

13.3. PROM

PROM соответствует по структуре ROM, но с тем отличием, что она может программироваться пользователем. PROM также выполнена в виде матрицы, со столбцевым и строчным декодерами для адресного декодирования. Возможная реализация может быть разработана на основе рис. 13.2. Стоки транзисторов в ячейках ЗУ PROM могут быть законтрактивированы вместо соединения проводящими дорожками соединением плавкими перемычками (fusible link). Это соединение показано на рис. 13.4. В том случае, если должно быть запомнено состояние Н, плавкая перемычка при программировании расплавляется током. Как правило, для этого необходимо специальное программирующее устройство. Плавкая перемычка подобна плавкому предохранителю. Записанная информация является

энергонезависимой. Однажды расплавленная плавкая перемычка не может быть вновь восстановлена. Поэтому подобные схемы обозначаются как OTP-ROM (OTP — one time programmable), то есть как ROM с однократным программированием.

Логический символ PROM подобен ROM (рис. 13.3).

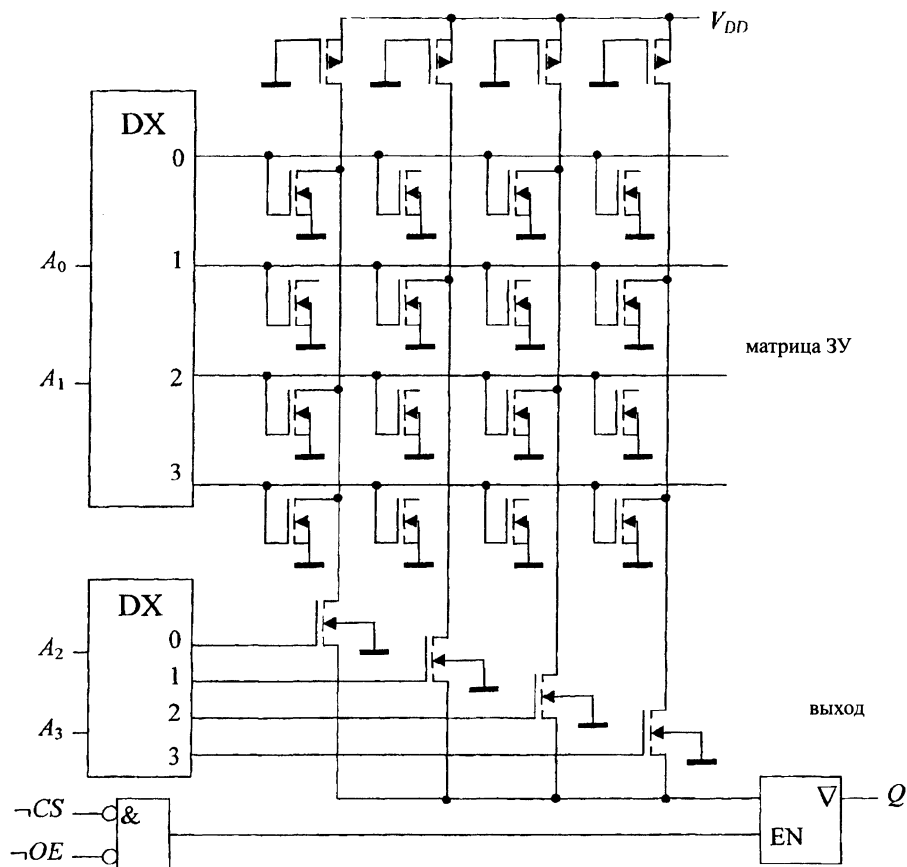


Рис. 13.2. Структура ROM с емкостью 4×4 бит.

13.4. EPROM

Через EPROM обозначают erasable programmable ROM, стираемое программируемое ПЗУ, СППЗУ. EPROM по своей структуре соответствует ROM или PROM только на месте плавких перемычек или программируемых с помощью маски соединений находятся «стираемые» запоминающие элементы.

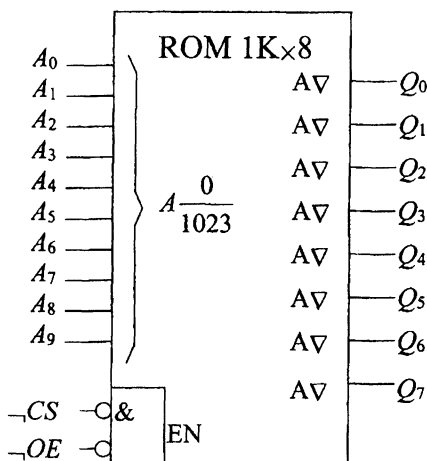


Рис. 13.3. ROM с емкостью $1K \times 8$.

В качестве таких запоминающих элементов используются МОП ПТ с плавающим затвором (рис. 13.5). Эти МОП ПТ являются нормально-закрытыми транзисторами с дополнительным затвором, который не имеет внешнего вывода и называется плавающим затвором. Изначально этот затвор имеет нулевой потенциал. С помощью наведения заряда на плавающем затворе может быть осуществлена запись информации в ячейки.

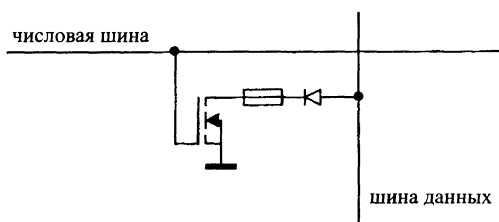


Рис. 13.4. Ячейка ЗУ типа PROM.

При отсутствии заряда на плавающем затворе транзистор функционирует как обычный n -канальный транзистор нормально-закрытого типа. Достаточно большое положительное напряжение на затворе включает транзистор. Это состояние ведет к уровню L на шине данных в том случае, если данная ячейка ЗУ выбрана числовой шиной.

В том случае, если должен быть запомнен уровень H, на плавающем затворе должен быть накоплен отрицательный заряд. Теперь

транзистор всегда заперт и при выборе ячейки с помощью числовой шины шин данных остается на уровне Н. Отрицательный заряд на плавающем затворе создается за счет туннелирования электронов через оксид. При повышенном напряжении между стоком и подложкой поле между затворным электродом и каналом достигает таких высоких значений и за счет лавинного пробоя появляются «горячие» электроны. Определенное количество электронов при этом может туннелировать через оксид. На плавающем затворе таким образом накапливается отрицательный заряд, который запирает транзистор. Облучение УФ светом в течение приблизительно 20 минут, ионизирует изолирующий материал, лежащий между затвором и истоком, благодаря чему заряд рассасывается. При этом информация стирается. Благодаря хорошим диэлектрическим свойствам оксида накопленный заряд остается стабильным в течение многих лет.

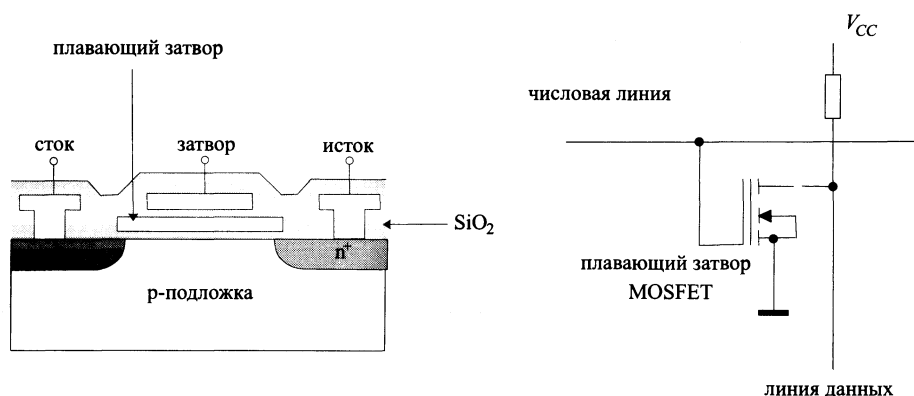


Рис. 13.5. МОП ПТ с плавающим затвором, структура и электрическая схема.

Для программирования между истоком и стоком подается относительно высокое напряжение.

Логический символ EPROM идентичен ROM (рис. 13.3).

13.5. EEPROM

Через аббревиатуру EEPROM обозначают электрически стираемое программируемое ПЗУ, ЭСППЗУ (electrically erasable programmable ROM). Эта схема ЗУ электрически записывается и электрически стирается.

Единичная ячейка ЗУ также как и в EPROM, построена на одном МОП ПТ с плавающим затвором. Однако оксид между пла-

вающим затвором и каналом имеет меньшую толщину. Благодаря этому оказывается возможным, при повышенном напряжении между затвором и каналом, перемещать электроны с затвора в канал и обратно. Это происходит за счет эффекта туннелирования Фаулера-Нордхайма.

EEPROM со специальными ячейками ЗУ иногда называют flash-EEPROM («мигающее» EEPROM). Они стираются не только полностью, но и поблочно.

Коммерчески доступные ROM, PROM, EPROM, EEPROM часто совместимы по выводам корпуса так, что возможно применение в фазе разработки PROM или EPROM, которые при переходе к серийной продукции можно заменить на ROM или на PROM.

Логический символ EEPROM идентичен логическому символу ROM (рис. 13.3).

13.6. EAROM

Через EAROM обозначают электрически программируемое ПЗУ, ЭППЗУ (electrically alterable ROM). По своему поведению EAROM похоже на EEPROM.

К различиям между EAROM и EEPROM. Распространены два различных способа обозначений, которые частично противоречат друг другу:

Часто схемы ЗУ с более высокой емкостью обозначают как EEPROM, в то время как узлы с меньшей емкостью обозначают как EAPROM

Но иногда через EEPROM, и в особенности через flash-EEPROM, обозначают схемы ЗУ, которые стираются целиком или поблочно. Под EAROM понимают тогда только ЗУ, стираемые по битам либо по байтам.

13.7. NOVRAM

NOVRAM (not volatile RAM) представляет собой энергонезависимые RAM (random access memory). Подобные ЗУ с произвольным доступом (то есть random access memory) создается из записывающего-считывающего ЗУ, не сохраняющего информацию при выключении электропитания. При выключении системы информационное содержимое сохраняется в течение приблизительно 10 мс в EEPROM. Для этого в каждой ячейке ЗУ содержится ячейка RAM и ячейка

EEPROM. Таким образом объединяются преимущества RAM, а именно быстрые чтение и запись в произвольной ячейки, и преимущество EEPROM, а именно независимость от источника энергоснабжения.

Таблица 13.1. Обзор энергонезависимых ЗУ.

Обозначение	Программирование		Стирание
ROM (read only memory)	Маска	Однократное	Невозможно
PROM (программируемое ROM) программируемое полем ROM, однократное PROM (OTPROM)	Эл.	Однократное	Невозможно
EPROM (erasable ROM)	Эл.	Многократное	УФ облучение (20 минут), все содержимое ЗУ
EEPROM (electrically erasable ROM) flash-EEPROM	Эл.	Многократное	Электрическое, все содержимое ЗУ или по битам (20–100 мс)
EAPROM (electrically alterable ROM)	Эл.	Многократное	Электрическое, по битам (20–100 мс)
NOVRAM (nonvolatile RAM)	Эл.	Многократное	Электрическое, по битам (100 нс)

13.8. RAM

RAM является сокращением для random access memory (то есть ЗУ с произвольной выборкой, ЗУПВ). Под этим понимается схема ЗУ, в которое можно произвольно записывать и из которой можно произвольно считывать информацию. Различают статические RAM и динамические RAM (DRAM). В статических RAM в качестве ячеек памяти применяют триггеры. Схемы с высокой степенью интеграции работают в основном с использованием динамического запоминания информации в конденсаторах, управляемых одним транзистором.

13.8.1. Статическое RAM

Ячейка ЗУ RAM показана на рис. 13.6. Она построена из двух КМОП-инверторов. Обращаться к ячейке ЗУ можно через числовую шину. Для выбора ячейки на числовую шину подается Н. Вследствие этого T_5 и T_6 становятся низкоомными.

В процессе записи можно, подав уровень Н на шину данных DL, записать Н в ЗУ. В таком случае T_3 будет открыт, а T_4 — закрыт.

Правый инвертор выдаст L. После этого будет настроен на H левый инвертор. Точно также можно записать L в ЗУ, подав H на шину данных — DL. При считывании ячейка вновь выбирается с помощью числовой шины. На шинах данных может быть считан запомненный бит.

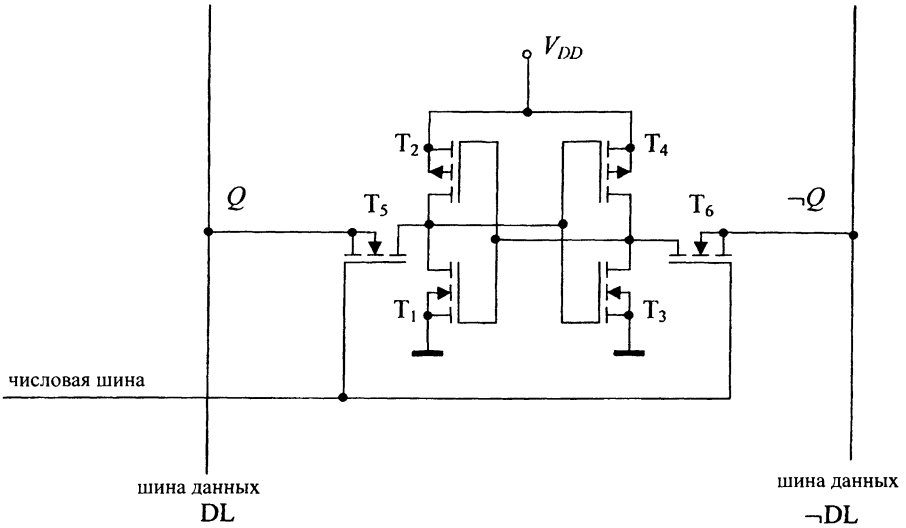


Рис. 13.6. Ячейка ЗУ типа RAM, выполненная с помощью КМОП-технологии.

Для считывания с обеих шин данных используется один симметричный усилитель считывания. Имеется также усилитель записи.

13.8.2. Пример работы RAM

В качестве примера представим типичную схему RAM. Емкость данного ЗУ представляет $2K \times 8$ бит, или 2K байт. Логический символ представлен на рис. 13.7.

Вход $\neg CS$ (chip select, «выбор чипа»), который также называют $\neg CE$ обеспечивает выбор данной схемы, если к одной шине подключено несколько ЗУ. Если $\neg CS = H$, то схема находится в режиме хранения с уменьшенным потреблением тока.

Поскольку эта схема должна быть подключена к шине данных, она имеет выходы типа tristate («с тремя состояниями»), которые могут быть сделаны высокоомными с помощью сигнала $\neg OE = H$.

В соответствии со своей емкостью схема ЗУ имеет 11 адресных входов A_0-A_{10} . При $RD/\neg WR = H$ содержимое ячейки ЗУ может быть считано, при $RD/\neg WR = L$ в нее можно записывать.

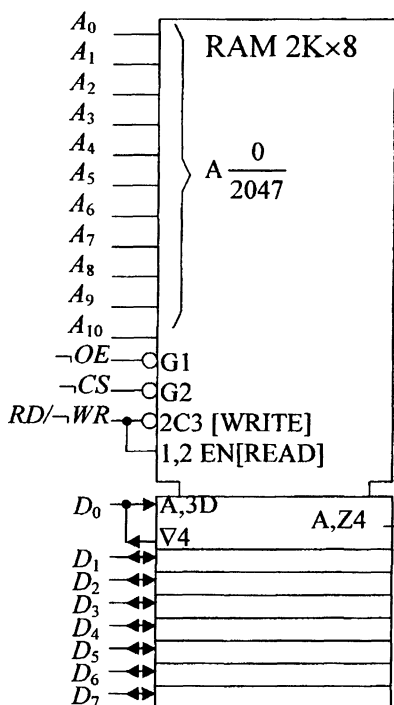


Рис. 13.7. RAM с емкостью ЗУ $2\text{ К} \times 8$ бит.

На временной диаграмме (рис. 13.8) представлен цикл считывания. На протяжении всего цикла считывания должно быть

$$RD/\neg WR = H.$$

Когда значащие адреса подводятся к адресной шине, вначале с помощью $\neg CS$ выбирается схема, затем выход может быть активирован с помощью $\neg OE$. После декодирования адресов в RAM значащие данные находятся на шине данных.

Внесенные во временную диаграмму интервалы времени определяются следующим образом:

t_{RC} — время цикла считывания — read cycle time.

В этот интервал времени может быть проведен полный цикл считывания. Интервал времени является важным, когда проводится много циклов считывания один за другим.

t_{AA} — время выборки адреса — address access time.

Если к адресной шине приложены значащие адреса, то после интервала времени, обозначаемого как время выборки адреса, на шине данных находятся значащие данные.

t_{CO} — минимальное время действия сигнала $\neg OE = L$ до появления данных на шине данных.

t_{OO} — интервал времени, в течение которого данные еще находятся на шине данных после того, как сигнал $\neg OE$ перешел на уровень Н.

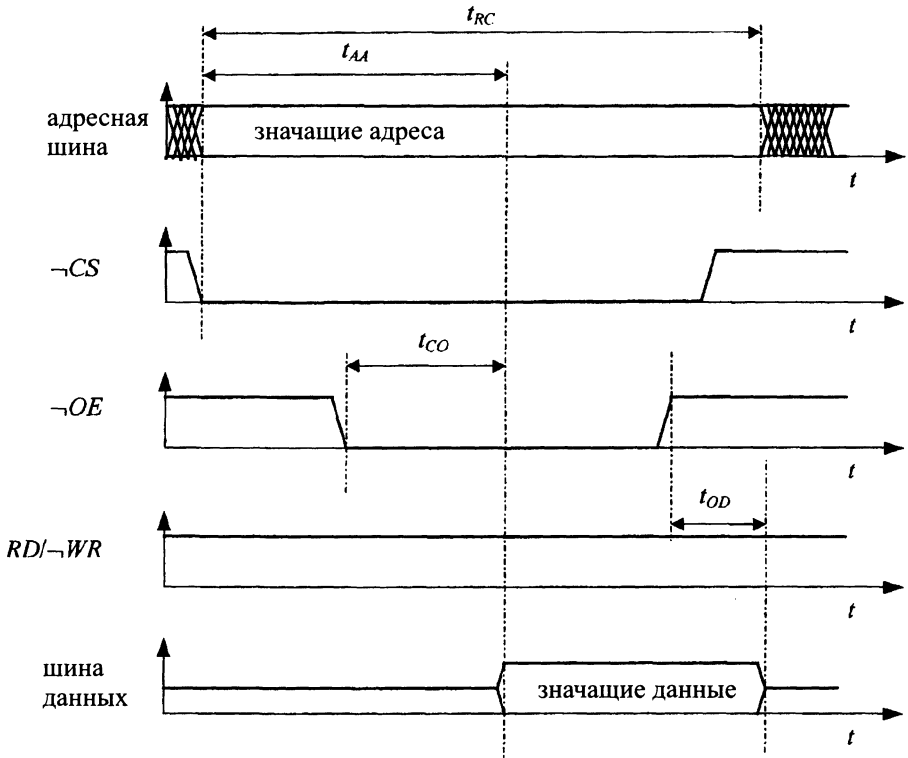


Рис. 13.8. Цикл считывания RAM.

На рис. 13.9 представлена временная диаграмма цикла записи. В одном цикле записи всегда $\neg OE = H$ и передающий блок (как правило, микропроцессор) может выдавать данные на шину данных. Для записи в ячейку ЗУ должны быть справедливы равенства $RD/\neg WR = L$ и $\neg CS = L$. Различают два случая:

1. Ранняя запись (early write). При этом в течение всего цикла записи $RD/\neg WR = L$, а процесс записи запускается отрицательным фронтом сигнала $\neg CS$. Цикл ранней записи изображен на рис. 13.9.
2. Поздняя запись (late write). При этом в течение всего цикла записи $\neg CS = L$, а процесс записи запускается отрицательным

фронтом сигнала $RD/\neg WR = L$. Здесь в противовес циклу ранней записи $RD/\neg WR = L$ и $\neg CS = L$ меняются ролями.

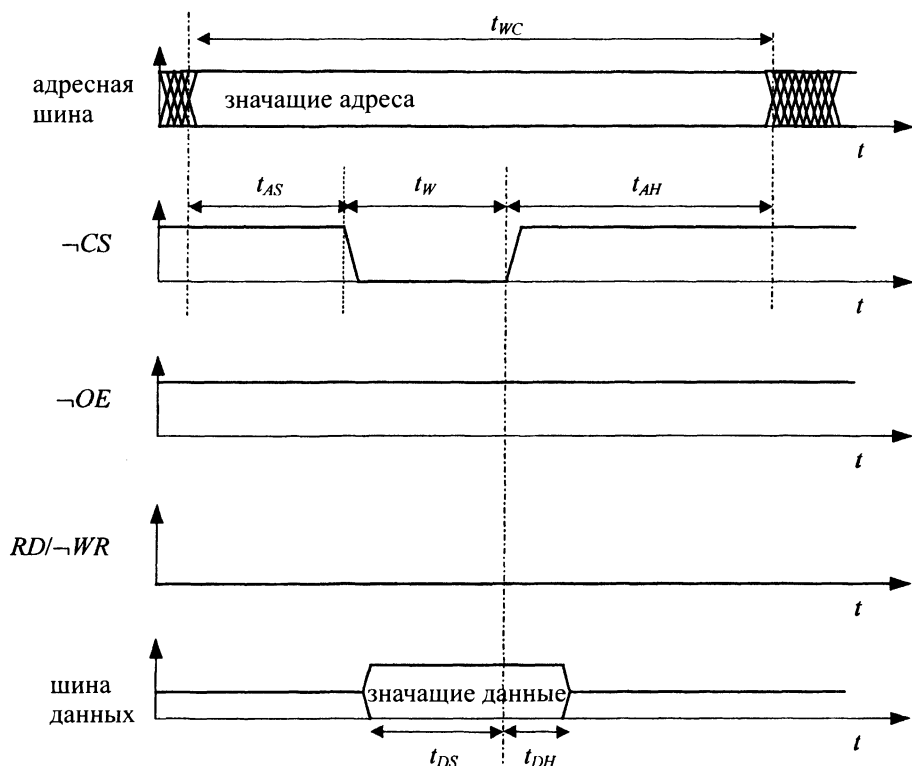


Рис. 13.9. Цикл записи RAM (early write, ранняя запись).

На временной диаграмме зафиксированы следующие временные интервалы:

t_{WC} — время цикла записи — write cycle time. В течение этого интервала времени может быть проведен полный цикл записи.

t_{DS} — соответствует времени установки (setup time).

t_{DH} — соответствует времени удержания (hold time) D-триггера. В течение зафиксированного этими интервалами отрезка времени данные должны стабильно находиться на шине данных.

t_{AS} — время установки адреса (address time set).

t_{AH} — время удержания адреса (address hold time).

t_{AS} — интервал времени, в течение которого адреса перед $\neg CS$ -импульсом с длительностью t_W должны стабильно быть в наличии.

t_{AH} определяет интервал времени, в течение которого адреса должны быть в наличии после импульса $\neg CS$. Оба интервала времени необходимы для декодирования строчных и столбцевых адресов в RAM.

13.9. Динамическое RAM

DRAM (динамическое RAM) представляет собой энергозависимое полупроводниковое ЗУ, в котором информация запоминается в конденсаторах. Благодаря очень простой структуре ячейки ЗУ подобные DRAM имеют очень большую информационную емкость.

Запоминающий DRAM элемент показан на рис. 13.10. Приложение Н к числовой линии позволяет выбрать ячейку ЗУ. Тогда накопленный на конденсаторе заряд может стекать по шине данных. Наличие заряда означает содержимое ЗУ со значением Н, а значению L соответствует отсутствие заряда. Чтение разрушает накопленный заряд, так что после каждого считывания заряд должен быть снова восстановлен (регенерирован).

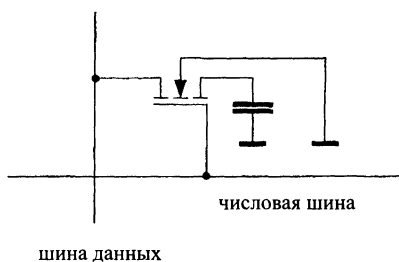


Рис. 13.10. Ячейка динамического RAM.

Динамическое RAM организовано так, что процесс считывания автоматически связан с регенерацией заряда. Если ячейка ЗУ не считывается определенное время, заряд рассасывается, и информация теряется. Поэтому все запомненное содержимое должно периодически регенерироваться с помощью процесса считывания. Поскольку сопротивление МОП ПТ в выключенном состоянии очень велико, то достаточно очень малых емкостей конденсаторов, чтобы поддерживать время разряда в диапазоне миллисекунд.

В качестве примера представлена схема TMS416400. Это DARM имеет емкость $4\text{М} \times 4\text{бит}$. Данные в каждой ячейке ЗУ должны регенерироваться каждые 64 мс. В эту схему строчные и столбцовые адреса должны загружаться через те же самые контактные площадки, чтобы схема была достаточно малой площади. Как показано на принципиальной схеме (рис. 13.11) в TMS416400 для выборки столбцов используется 10 бит и для выборки строк 12 бит. Для введения строчных адресов используется контактная площадка $\neg RAS$ (row address strobe, выборка строчных адресов), а для введения столбцовых адресов контактная площадка $\neg CAS$ (column address strobe, выборка столбцовых адресов). С помощью усилителя записи и считывания вводятся и считываются слова длиной 4 бита.

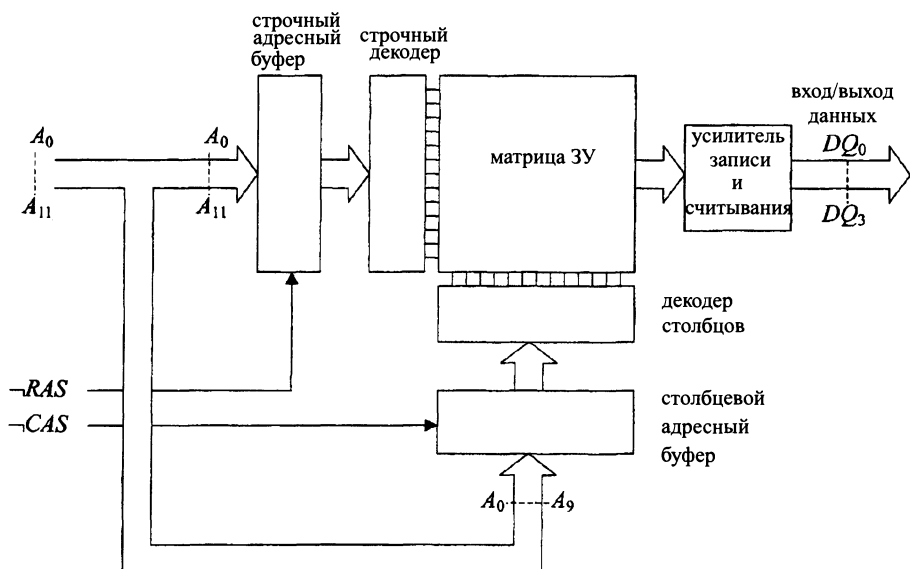


Рис. 13.11. Принципиальная схема DRAM с емкостью $4\text{М} \times 4\text{бит}$ (TMS416400).

Логический символ TMS416400 показан на рис. 13.12.

По индикации зависимости можно видеть, что $\neg RAS$ (с индикацией зависимости C20) использует адресные шины от A_0 до A_{11} (индикация зависимости 21).

Считывание

Временная диаграмма процесса считывания показана на рис. 13.13. Вначале следует подать строчные адреса, которые вводятся спадающим фронтом сигнала $\neg RAS$. Затем подается адрес столбца, который вводится спадающим фронтом сигнала $\neg CAS$. В логическом

символе эти взаимозависимости можно определить на основе обозначений $C20$ и $C21$. С приложением $\neg WR = 1$ начинается время доступа адреса t_{AA} , по окончании которого находится значащие данные. Если при спадающем фронте сигнала $\neg CAS$ (цифра 21) входной сигнал $\neg RAS = 0$ (цифры 23 и 24) и входной сигнал $\neg OE = 0$ (цифра 25). Эта взаимозависимость представлена в нижнем прямоугольнике внутри обрамления символа.

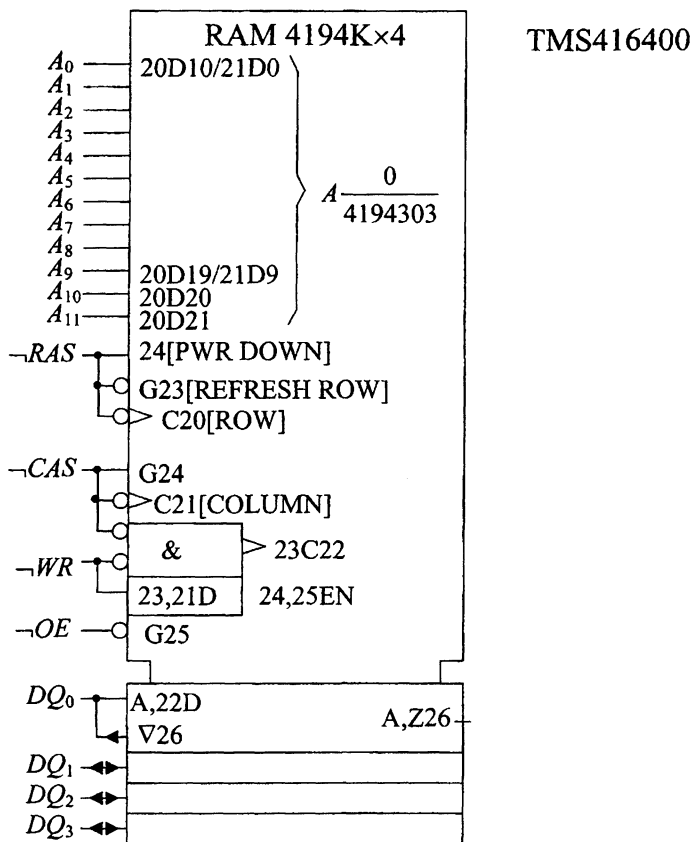


Рис. 13.12. Логический символ TSM416400.

Область ЗУ, которая может быть найдена, по одному и тому же строчному адресу, обозначают как «страницу». Имеется упрощенный способ чтения (англ. page mode), который применяют при желании прочесть повышенное количество данных на одной странице. При этом после введения строчного адреса остается постоянным $\neg RAS = 0$. Для отыскания различных данных на странице соответствующие адреса столбцов многократно варьируются и вво-

дятся спадающим фронтом сигнала $\neg CAS$. Еще одно улучшение в TSM416400 достигается за счет того, что вслед за отрицательным фронтом сигнала $\neg RAS$ уже начинается оценка адресов столбцов, которые уже могут быть введены вскоре после спадающего фронта $\neg RAS$ (после hold time). Связанное с этим повышение скорости обозначают как page mode (то есть «улучшенный способ чтения страницы»). Когда подходит спадающий фронт $\neg CAS$, декодирование адреса столбца уже началось. Время доступа для чтения на одной странице при использовании enhanced page mode составляет t_{CAC} и это время короче t_{AA} .

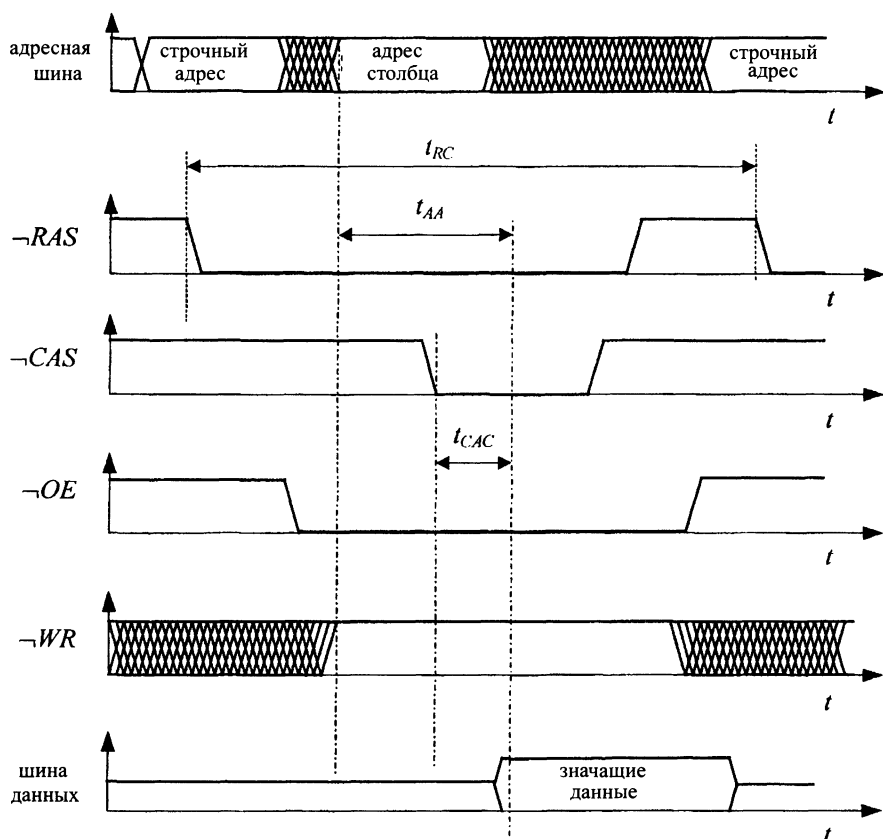


Рис. 13.13. Цикл считывания TSM416400.

Запись

Для записи вначале вводится строчный адрес, который вводится спадающим фронтом $\neg RAS$. Затем накладывается адрес столбца, который вводится спадающим фронтом сигнала $\neg CAS$.

При записи, как показывает цифра 23 индикации зависимости $\neg RAS = L$. Данные во вход данных вводятся тогда, когда на одном из входов $\neg CAS$ или $\neg WR$ имеет место уровень L , а на другом входе появляется спадающий фронт. На рис. 13.14 показан случай, при котором сначала $\neg WR$ переходит в состояние L и потом спадающий фронт $\neg CAS$ вводит данные. Этот случай называется early write. Также как и в триггере, данные должны быть стабильны на входе $\neg CAS$ между временем установки (setup time) t_{DS} перед спадающим фронтом и временем удержания (hold time) t_{DH} после спадающего фронта.

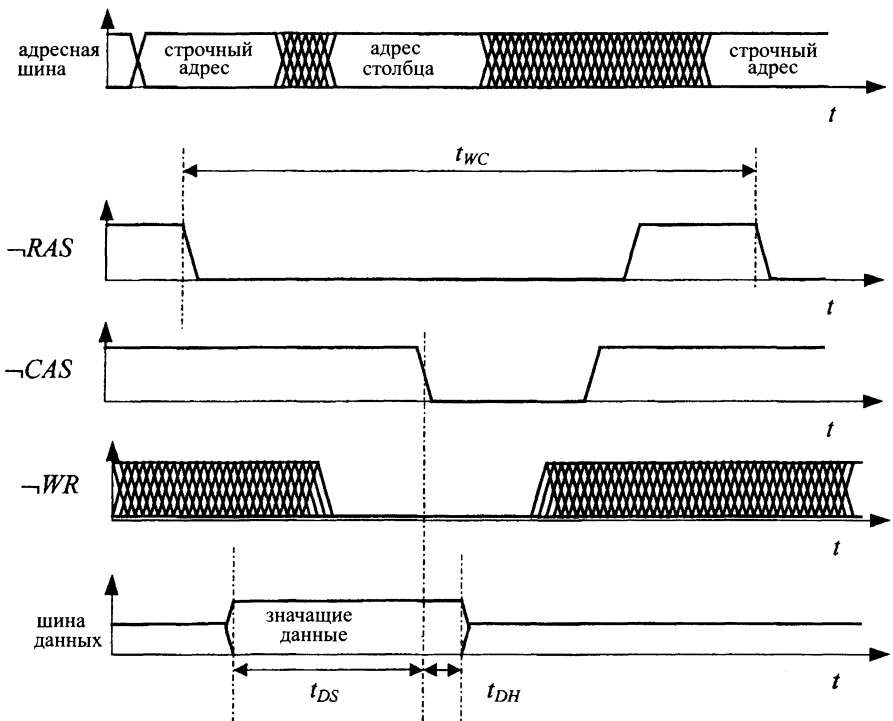


Рис. 13.14. Цикл записи (early write) TSM416400.

Если данные вводятся при падающем фронте на $\neg WR$, то этот процесс называют late write.

Регенерация

Каждые 64 мс каждая ячейка ЗУ должна регенерироваться (восстанавливать информацию), в противном случае данные будут утеряны. Нормальная операция записи или считывания одного бита в строке вновь регенерирует все биты этой ячейки. Поэтому для пол-

ной регенерации достаточно периодически прочитывать все 4096 строки, выбирая адресные линии от A_0 до A_{11} с помощью счетчика. TSM416400 может быть регенерирован в соответствии со следующей методикой:

Регенерация (refresh) только с помощью RAS

На $\neg CAS$ при этом будет оставаться H . Как показано на изображении логического символа, для получения низкоомных выходов $\neg CAS$ и $\neg OE$ должны быть в состоянии L . Отсюда следует, что выход в этом случае остается низкоомным так, что потери мощности в чипе во время регенерации остаются низкими. Для этой операции регенерации используются адреса, генерируемые дополнительным счетчиком. После каждой смены адресов с помощью $\neg RAS$ вводится новый адрес подлежащей регенерации ячейки.

Скрытая регенерация (hidden refresh).

В этом случае процесс регенерации производится одновременно с процессом чтения. Но $\neg CAS$ после окончания процесса считывания остается в состоянии L . Вследствие этого, данные на выходе в течение следующей операции остаются действительными и следующий процесс регенерации не оказывает влияния. Теперь $\neg RAS$ переключается циклически между L и H . Пословные адреса подлежащих регенерации ячеек ЗУ генерируются внутри. Регенерация с помощью CAS перед RAS (CAS before RAS). При этом виде регенерации в начале появляется спадающий фронт на $\neg CAS$, а затем на $\neg RAS$. Благодаря обратной последовательности сигнализируется прохождение цикла регенерации в схеме ЗУ. Игнорируются приложенные извне адреса и применяются адреса подлежащих регенерации строк, взятые из внутреннего счетчика строчных адресов. Для обеспечения последовательности операций $\neg CAS$ остается в состоянии L и $\neg RAS$ циклически переключается между L и H .

Хранение

Схема может быть установлена в режим хранения, в котором она потребляет очень малую мощность.

13.10. Квзистатическое DRAM

Если в DRAM содержатся средства для проведения регенерации и внутренний счетчик для генерации адресов, то его называют квазистатическим DRAM, поскольку внешне подобное ЗУ функционирует как статическое RAM.

13.11. ЗУ «пожарная цепочка»

ЗУ типа «пожарная цепочка» представляют собой цифровые ЗУ, в которых данные могут запоминаться последовательно. На вход данные подаются последовательно, с выхода они могут сниматься в той же последовательности. Подобные ЗУ называют также FIFO (first in first out).

ЗУ типа «пожарная цепочка» применяют в качестве буферов, например, тогда, когда, поток данных должен передаваться в интерфейсе между двумя не синхронизированными по тактированию системами. Если поставляется больше данных, чем считываются, то в этом случае осуществляется промежуточное запоминание данных.

Другой организационной формой является LIFO (last in first out), которое обозначают также как ЗУ магазинного (stack). Структура LIFO аналогична FIFO.

13.11.1. Примеры FIFO

Рассмотрим FIFO с 64 запоминающими ячейками (SN74ACT2226, компания Texas Instruments).

Для того чтобы проводить независимо запись и считывание, в качестве основного элемента ЗУ применено RAM с двумя портами (рис. 13.15). Оно представляет собой ячейку RAM, расширенную за счет второй пары шин. В ЗУ имеются два полностью независимых управляющих порта. На обоих можно проводить считывание одновременно. Разумеется, нельзя одновременно считывать одну и ту же ячейку и записывать в нее. Это возможно только для различных ячеек. Поэтому нужна логика, с помощью которой можно было бы распознавать подобные конфликты.

Для блока FIFO (рис. 13.16) в качестве ЗУ используется RAM с двумя портами, в которой ячейки ЗУ размещены по кругу. Как можно видеть на блок схеме, для считывания и записи применяются отдельные тактовые сигналы ($RdClk$ и $WrClk$), которые не должны быть синхронными. Для записи данных по входу D требуется, чтобы $WrEn = 1$ (write enable), чтобы флаг готовности к вводу (input-ready flag) $InRdy = 1$, чтобы y такта записи $WrClk$ имел место нарастающий фронт. То же самое справедливо для выхода. $RdEn = 1$, $OutRdy = 1$ и нарастающий фронт такта считывания $RdClk$ должны появиться для того, чтобы на выходе Q мог быть считан один бит.

Ячейки ЗУ, в которых в данном RAM с двумя портами могут быть записаны данные, запоминаются в указателе «запись». Он

представляет собой блок полупроводникового ЗУ, в котором хранится адрес ячейки ЗУ, в которой будет осуществлена следующая запись. Это будет адрес, который на 1 ниже того, по которому записывали перед этим. В указателе «считывание» хранятся адреса ячеек ЗУ, из которых будет производиться следующее считывание. После процесса считывания указатель производит снижение на 1. Следовательно, запомненные данные стоят между обоими указателями, как это показано на блок-схеме. Дополнительно имеется flag (индикатор) почти заполненного или почти пустого ЗУ (F/F). На половину заполненное ЗУ показывает индикатор HF .

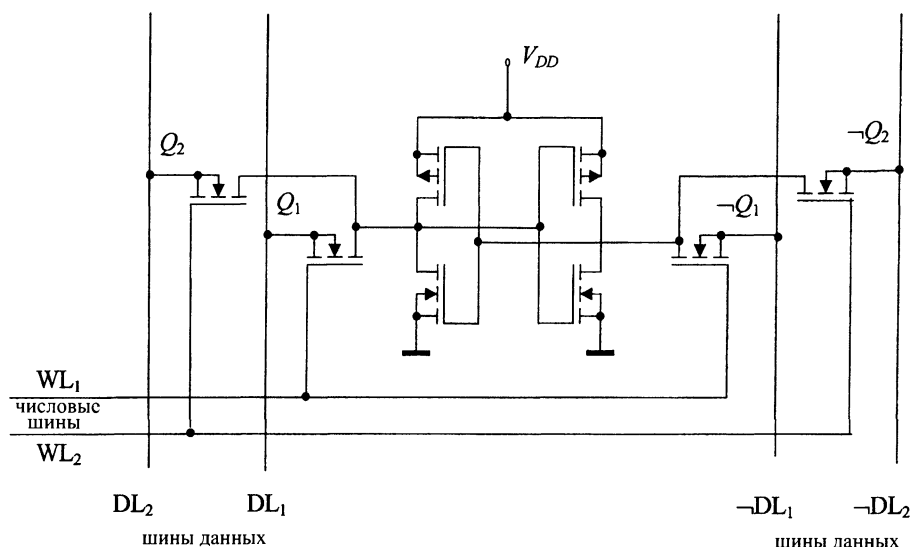


Рис. 13.15. Принципиальная схема RAM-ячейки с двумя входами.

Вход $\neg RST$ (reset) позволяет установить ЗУ в исходное состояние, то есть стирается.

Логический символ FIFO показан на рис. 13.17.

13.12. Каскадирование ЗУ

Во многих случаях появляется необходимость составления ЗУ из многих схем. Это имеет место, когда:

1. Не хватает емкости ЗУ. Следует различать, когда используются слишком короткие слова и когда применяется слишком малое количество ячеек ЗУ.

- Блок ЗУ должен состоять из энергонезависимых *ROM* и энергозависимых ЗУ с записью и считыванием.
- Из-за соображений стоимости необходимо комбинировать быстрые и медленные ЗУ.

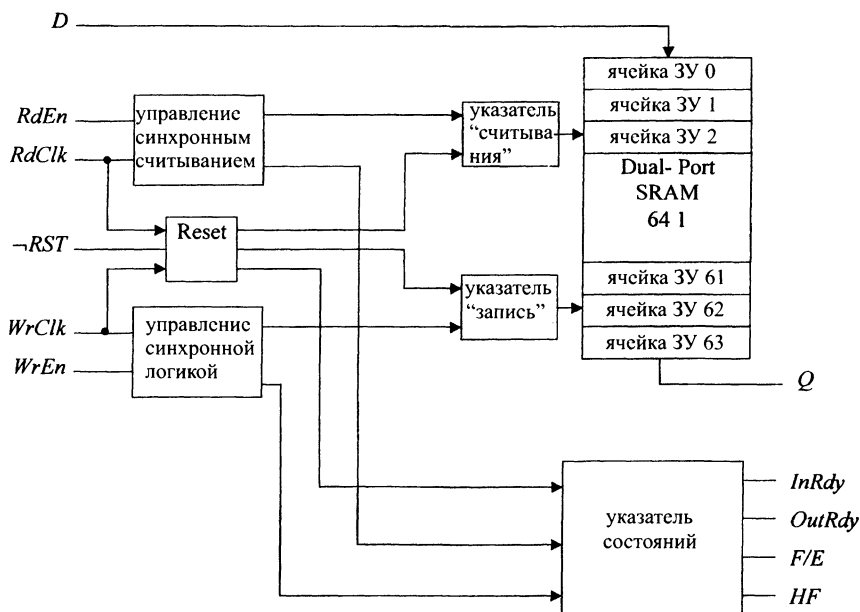


Рис. 13.16. Блочная схема FIFO-ЗУ SN74ACT2226.

13.13. Увеличение длины слов

Пусть, например, в шине данных должно быть реализовано слово с длиной 8 бит, а в нашем распоряжении имеются схемы ЗУ с длиной слова 4 бит. В этом случае их можно соединить между собой так, как это показано на рис. 13.18. К отдельным схемам (ЗУ адресная шина и шина данных) подводятся идентично. Вход и выход данных первой схемы ЗУ подключена для передачи от нулевого до 3-го бита в шину данных, в то время как вход и выход данных другой схемы ЗУ подсоединены для передачи по шине данных от 4-го до 7-го битов.

13.14. Увеличение емкости ЗУ

В тех случаях, когда число ячеек в ЗУ должно быть увеличено, необходимо включать вместе несколько ЗУ меньшей емкости.

Ниже даны несколько примеров для ЗУ с емкостью $8\text{ К} \times 8\text{ бит}$. Для данного ЗУ необходимо объединить 4 отдельные схемы ЗУ с емкостями $2\text{ К} \times 8\text{ бит}$. Это ЗУ должно быть подключено к адресной шине с разрядностью 16 бит. Разрядность шины данных составляет 8 бит. Отдельные схемы ЗУ емкостью $2\text{ К} \times 8\text{ бит}$ имеют 11 адресных портов A_0 – A_{10} .

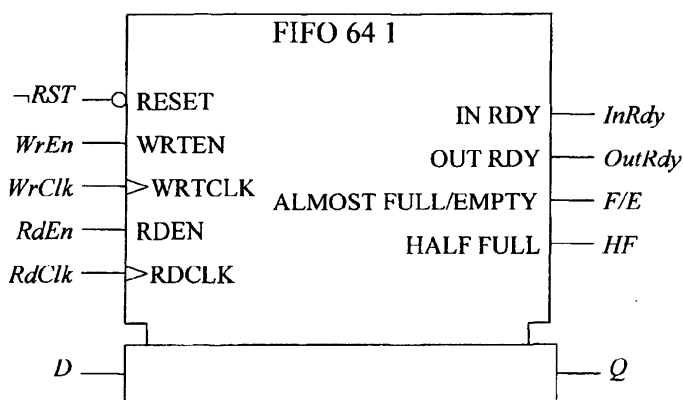


Рис. 13.17. Логический символ FIFO-ЗУ SN74ACT2226.

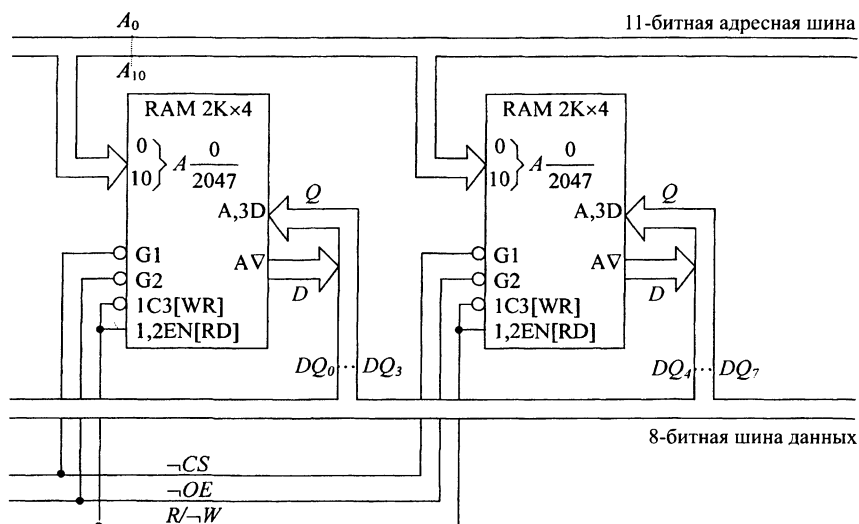


Рис. 13.18. Увеличение длины слов ЗУ до 8 бит

Задача, которую необходимо решить при соединении в один блок нескольких схем ЗУ, заключается в обеспечении такого декодирова-

ния адресных линий $A_{11}-A_{15}$, которое было бы нацелено на выборку ячеек памяти в отдельных схемах ЗУ. Ниже представлено несколько обычно используемых решений этой задачи с тремя *RAM* и одной *ROM*.

13.14.1. Полное декодирование

При полном декодировании используются все адресные шины и каждая ячейка ЗУ имеет только один адрес.

Эти условия можно обеспечить, применяя демультиплексор с 4 выходами, которые подключены ко входам выбора чипа $\neg CS$ четырех схем ЗУ (рис. 13.19). Входы демультиплексора подключаются к адресным шинам A_{11} и A_{12} .

Адресные шины с более высокими номерами от A_{13} до A_{15} подключаются ко входу выбора чипа демультиплексора с помощью вентилей ИЛИ с тем, чтобы при адресах, которые выше *IFFFY* (*H* «шестнадцатиричный») не затрагивалась ни одна из схем.

Адресный план системы (рис. 13.19) показан в табл. 13.2. Адреса ячеек ЗУ представлены на основе шестнадцатиричного способа записи и с помощью двоичного представления. Таблица показывает, что ячейки лежат плотно. Следовательно программист может данные перезаписывать произвольно (но не в *ROM*), не принимая во внимание какие-либо пробелы. Самым высоким адресом является $IFFFH = 4 \times 2048 - 1$, что соответствует емкости ЗУ 8 К.

Таблица 13.2. Адресный план системы полного декодирования, показанной на рис. 13.19. Приведены самые низкие и самые высокие адреса для каждого ЗУ.

Схемный узел	Адрес (шестнадцатиричный)	Адрес (двоичный)															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 (ROM)	0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0 7 F F	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2 (RAM)	0 8 0 0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0 F F F	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
3 (RAM)	1 0 0 0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	1 7 F F	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
4 (RAM)	1 8 0 0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	1 F F F	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

В последнее время декодирование адресов для систем ЗУ часто проводят с помощью логических схем, которые рассматриваются в главе 14.

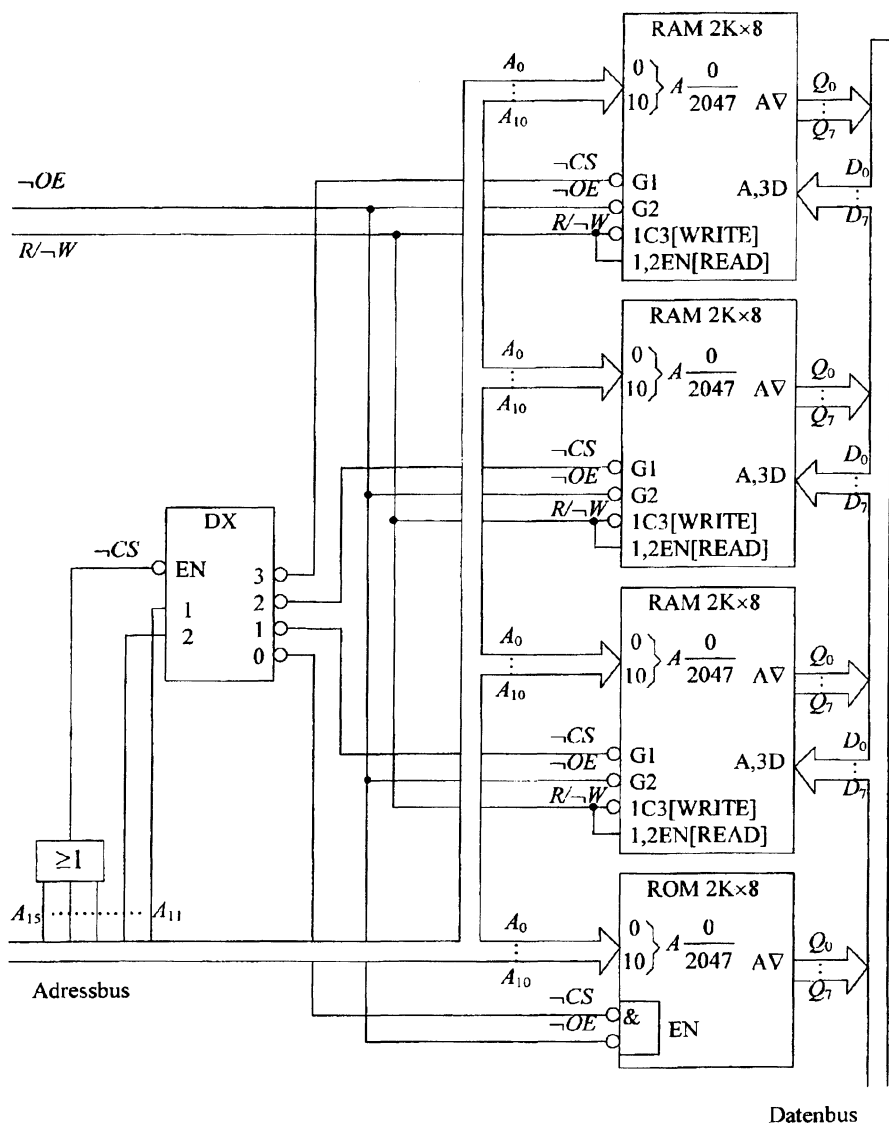


Рис. 13.19. Полное декодирование системы с ЗУ емкостью 8 К.

13.14.2. Частичное декодирование

При частичном декодировании используются не все адресные шины. В системе, показанной на рис. 13.20 для декодирования адресных линий A_{11} и A_{12} применен демультиплексор. Для того, чтобы уменьшить аппаратные затраты, более высокие адресные линии

от A_{13} до A_{15} не декодируются. Находящиеся на этих линиях биты являются, следовательно, don't care.

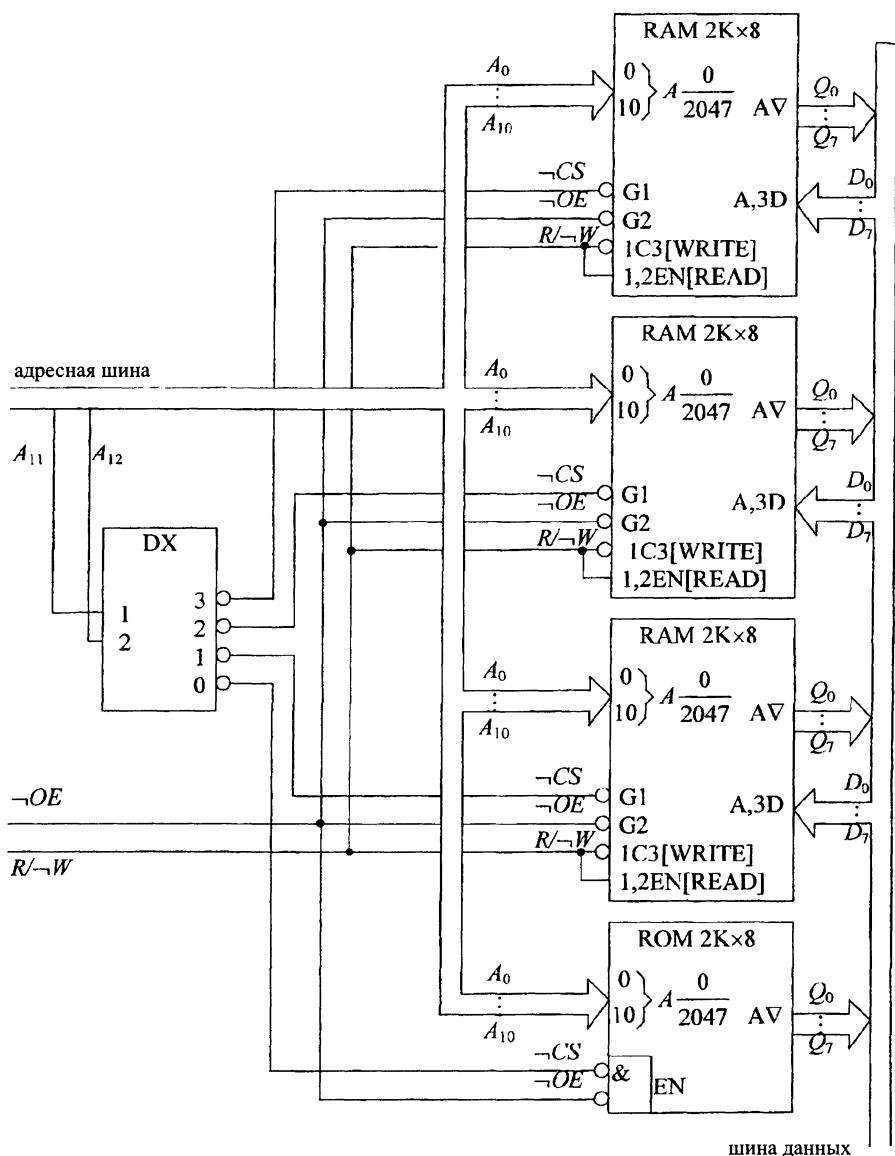


Рис. 13.20. Частичное декодирование системы ЗУ с емкостью 8 К.

Как показывает адресный план, приведенный в табл. 13.3, каждую ячейку ЗУ можно выбрать по 8 адресам, поскольку 3 старших бита являются битами don't care. Но при программировании имеет

смысл установить $A_{15} = A_{14} = A_{13} = 0$. Теперь можно достаточно просто определить шестнадцатичные адреса.

Таблица 13.3. Адресный план системы частичного декодирования, показанной на рис. 13.20. Даны нижние и самые высокие адреса для каждого ЗУ.

Схемный узел	Адрес (шестнадцатичный)	Адресные шины (двоичные)															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 (ROM)	0 0 0 0	d	d	d	0	0	0	0	0	0	0	0	0	0	0	0	0
	E 7 F F	d	d	d	0	0	1	1	1	1	1	1	1	1	1	1	1
2 (RAM)	0 8 0 0	d	d	d	0	1	0	0	0	0	0	0	0	0	0	0	0
	E F F F	d	d	d	0	1	1	1	1	1	1	1	1	1	1	1	1
3 (RAM)	1 0 0 0	d	d	d	1	0	0	0	0	0	0	0	0	0	0	0	0
	F 7 F F	d	d	d	1	0	1	1	1	1	1	1	1	1	1	1	1
4 (RAM)	1 8 0 0	d	d	d	1	1	0	0	0	0	0	0	0	0	0	0	0
	F F F F	d	d	d	1	1	1	1	1	1	1	1	1	1	1	1	1

13.14.3. Линейное декодирование

При линейном декодировании отказываются от декодера. Вместо этого верхние адресные шины от A_{11} до A_{14} подключены непосредственно ко входам выбора чипа CS схем ЗУ. Для того, чтобы обеспечить более простую структуру зоны памяти, на рис. 13.21 использованы схемы с неинвертированным входом выбора чипа GS . Линейное декодирование сильно ограничивает полезную зону памяти. В данном случае могут быть подключены только 5 схем с емкостью $2\text{ К} \times 8\text{ бит} = 16\text{ Кбайт}$, так как в наличии имеются только 5 адресных линий. Поэтому применение линейного декодирования ограничивается системами с малой потребностью в ячейках ЗУ. На приведенном в табл. 13.4 адресном плане можно видеть, что в зоне памяти появились пробелы. В тех случаях, когда подобное ЗУ вводится в микропроцессорную систему, программист должен следить за тем, чтобы пытался перезаписывать данные в эти пробелы. При шестнадцатичном представлении адресного плана предварительно задается $A_{15} = 0$. В противном случае система будет плохо просматриваться. Следует учесть, что при адресе 7800 Н обращение происходит ко всем четырем схемам ЗУ и вследствие этого могут появиться ошибки. Линейное декодирование применяется только там, где к многоразрядной адресной шине подсоединяется только несколько схем ЗУ с малой емкостью.

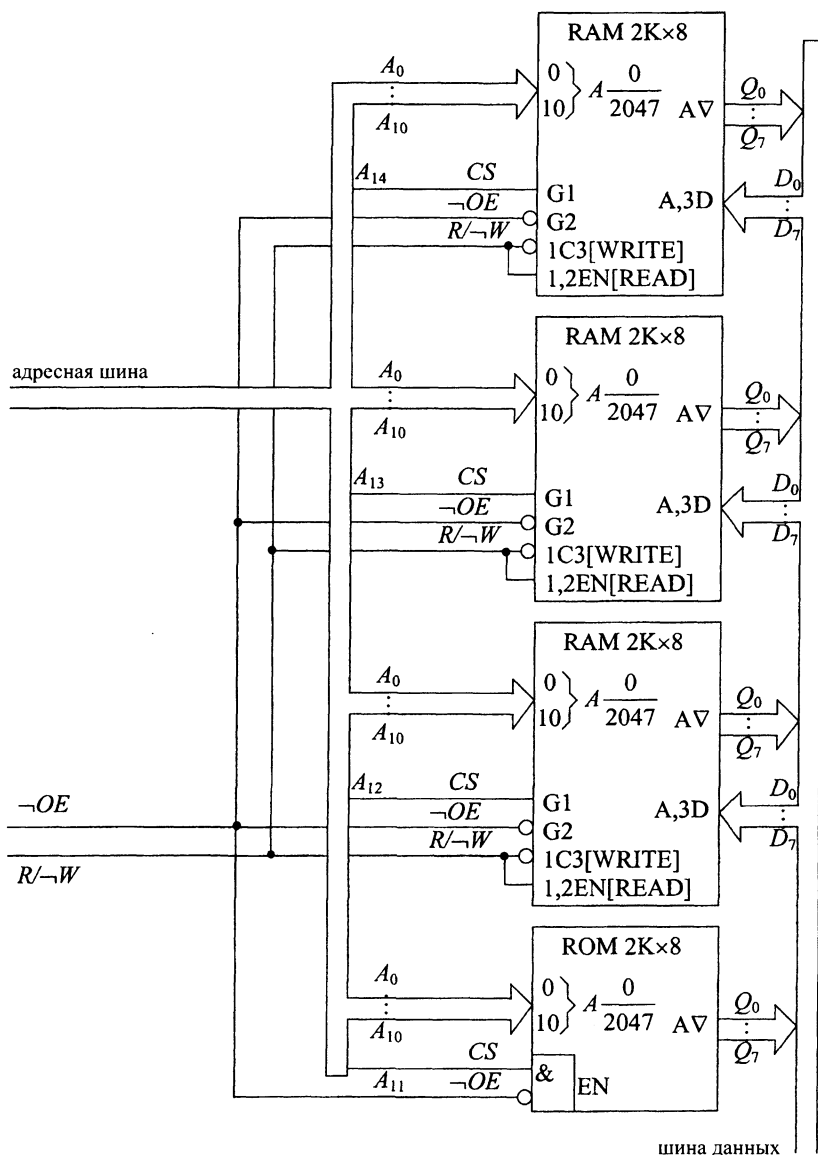


Рис. 13.21. Линейное декодирование системы ЗУ 8К.

13.15. Упражнения

Задача 13.1. Сконструируйте ЗУ для 16-битовой адресной шины и 8-битовой шины данных. Должны быть установлены, начиная с

адреса 0000Н, одна схема ROM с емкостью 4 К байт и схемы RAM с емкостями 2 К, 2 К и 8 К байт.

1. Ячейки ЗУ «полностью декодируются». Определите для каждой схемы ЗУ самый нижний и самый верхний адреса.
2. Декодирование должно проводиться с помощью демультимплексора. К каким адресным шинам должны быть подключены входы декодера?
3. Разработайте логическую схему, которая, будучи подключенной к выходам декодера, будет правильно управлять отдельными схемами ЗУ. Как будут подключаться оставшиеся адресные шины?

Таблица 13.4. Адресный план, показанный на рис. 13.21 системы с линейным декодированием. Приведены самые низкие и самые высокие адреса для каждого ЗУ.

Схемный узел	Адрес (шестнадцатичный)	Адресные шины (двоичные)															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 (ROM)	0 8 0 0	d	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0 F F F	d	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
2 (RAM)	1 0 0 0	d	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	1 7 F F	d	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
3 (RAM)	2 0 0 0	d	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	2 7 F F	d	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
4 (RAM)	4 0 0 0	d	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	4 7 F F	d	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Задача 13.2. В системе УЗ с малым количеством ячеек памяти к адресной шине разрядностью 8 бит должен быть подключен ЗУ с длиной слова 8 бит. Должны быть подключены, начиная с низких адресов, интерфейсы с числом ячеек, соответствующих 2 байтам, 4 байтам и 8 байтам.

Адреса должны декодироваться линейно.

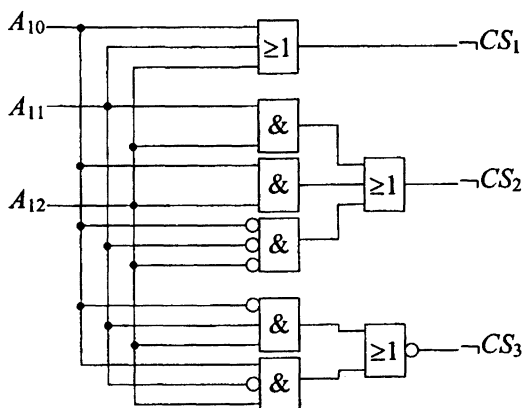
а) Предложите возможный вариант схемы. Как будут подключаться адресные шины?

б) Сформируйте адресный план.

Задача 13.3. Внизу показана схема декодера, предназначенного для микропроцессора с адресной шиной с разрядностью 16 бит. Три

схемы ЗУ, которые управляются этой схемой, имеют инвертирующие входы выбора чипа – $\neg CS_1$ – $\neg CS_2$ и $\neg CS_3$.

- Сформируйте булевы уравнения схемы декодера.
- Создайте на основе этого результирующую адресную схему.
- какую емкость было бы целесообразно задать для подсоединенных схем ЗУ, если по каждому адресу должен запрашиваться один байт?
- О каком виде декодирования идет речь?



ГЛАВА 14

ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ БЛОКИ

Для проектирования драйверов или логических схем предоставляются различные возможности. По экономическим соображениям предпочтение отдается стандартным элементам, которые могут быть изготовлены в больших количествах. В связи с этим возникает вопрос, насколько стандартные элементы соответствуют специальным требованиям отдельных заказчиков. Полупроводниковый рынок предлагает следующие возможности.

Комбинация на одной печатной плате стандартных ИС с низким уровнем интеграции

В данном случае, как правило, на одной печатной плате (PCB = = printed circuit board) соединяются отдельные вентили и ИС с малой степенью интеграции (SSI = small scale integration) и ИС со средней степенью интеграции (MSI = medium scale integration). Этот метод обеспечивает очень высокую гибкость. При этом схемы потребляют очень высокую мощность, так как все вентили должны иметь возбудитель линии. Кроме того, подобные схемы очень дороги в изготовлении. Они подходят прежде всего для малосерийного производства. Возможны также логические узлы из отдельных диодов и транзисторов. Данный метод изготовления сегодня из-за высоких затрат на монтаж и проверку используется только в исключительных случаях.

Специализированное по применению программное обеспечение (software)

К данному направлению относится прежде всего применение микропроцессора. За счет применения программного обеспечения данный подход имеет высокую гибкость, позволяющую подгонять характеристики к соответствующим требованиям.

Специализированная по применению аппаратные средства (hardware)

Гибкость можно обеспечить за счет применения ИС специализированных в соответствии с потребностями заказчика. Подобные ИС

охватываются общим понятием *ASIC* = *application specific integrated circuit*). К ним относятся ИС, которые с помощью физических изменений (а также с помощью конфигурационных программ) могут быть приведены в соответствие с определенными требованиями. *ASIC* охватывает как специально изготовленные ИС (полностью заказное проектирование) со специально разработанной для заказчика логикой, так и стандартные схемные узлы, в которых заказчики могут разрывать соединения током для достижения определенных схемных конфигураций.

14.1. Семейства ASIC

В настоящее время существует большое число различных типов *ASIC*. Грубо *ASIC* можно подразделить следующим образом.

Программируемые логические ИС (*programmable logic devices, PLD*)

Класс программируемых логических ИС (рис. 14.1) оказался первым семейством *ASIC*. С точки зрения изготовителей к ним могут быть отнесены стандартные блоки программируемых логических ИС, которые изготавливаются идентичными для всех заказчиков. Благодаря возможности программирования матриц из И- и ИЛИ-вентилей заказчики могут структурировать ИС на своей фирме так, как ему требуется.



Рис. 14.1. Структура *ASIC* различных типов

ASIC на основе полузаказного проектирования

В данном случае рассматриваемая *ASIC*, содержащие расположенные в виде матрицы вентили. Подобные *ASIC* могут быть приведены изготовителем в соответствие с требованиями заказчика с помощью структурирования линий связи. Отдельные вентили тестируются изготовителем и их характеристики полностью известны,

так что имеется в наличии библиотека ячеек, которые могут быть собраны пользователем в законченную схему на основе использования программного обеспечения. Стоимость разработки полужаказных *ASIC* меньше, чем у полностью заказных *ASIC*, но они медленнее. Поскольку они хуже используют поверхность чипа, затраты на чип выше, чем в случае полностью заказных *ASIC*. К этой группе относятся вентильные матрицы (*gate arrays*) и *ASIC* со стандартными ячейками.

***ASIC* на основе полного заказного проектирования.**

В случае данного типа *ASIC* речь идет о разработке, которая специально проводится для данного заказчика. Подобная разработка ничем не отличается от разработки обычной стандартной ИС. Этот подход обеспечивает наибольшую гибкость. Могут быть реализованы любые функции. Возможны даже вкрапления аналоговых частей схемы. Из-за больших затрат на разработку полностью заказное проектирование окупается только при больших объемах изготовления ИС. При этом может быть получена очень высокая плотность интеграции. Скорость обработки сигналов может быть очень высокой, если это требуется.

Из табл. 14.2 следует, что при переходе от программируемых пользователем логических блоков к полностью заказным ИС время, стоимость разработки, функциональная гибкость и минимальные объемы выпуска ИС увеличиваются. В то же время площадь чипа и стоимость изготовления одного чипа уменьшаются. На основе этого получаем различные области применения различных видов *ASIC*. На практике могут, разумеется, появиться отклонения от этих закономерностей.

Некоторые области применения появились только в результате развития *ASIC*. К ним можно отнести многофункциональные наручные часы, компьютерные устройства для чековых карт и портативные персональные компьютеры.

Основными определяющими параметрами при разработке цифровых систем являются:

- Затраты на разработку. Затраты на разработку различных *ASIC* очень сильно отличаются.
- Стоимость одной функции схемного блока. Как правило, при высоких затратах на разработку *ASIC* стоимость изготовления схемного блока минимальна, вследствие чего *ASIC* окупаются только при большом количестве образцов. Сюда относится также дополнительная стоимость корпусов, электропитания, печатной платы и т.д.

- Стоимость складирования и хранения материала.
- Затраты на эксплуатацию и техническое обслуживание.
- Время реализации проекта.
- Персональная производительность при разработке.

Таблица 14.1. Систематизация ASIC по структуре. Отражены основные тенденции в отношении затрат на различные ASIC и потребления мощности в них.

Тип ASIC	Программируемые пользователем логические блоки	Вентильные матрицы	ASIC со стандартными ячейками	Полностью заказные ASIC
фиксированные структуры	матрицы И/ИЛИ, матрицы из логических блоков	логические вентили	цифровые и аналоговые стандартные ячейки	—
программируемые блоки	матрицы, программируемые пережиганием перемычек, без пережигания и с помощью электрического заряда	соединяющие линии	все маски	все маски
формы реализации	PLA, PAL, PROM, EPROM, FPGA	канализированные вентильные матрицы, «море вентилей» (segof gates)	ASIC со стандартными ячейками, ИС с блоками ячеек	—

Для того, чтобы выбрать подходящую технологию для конкретного случая применения, необходимо произвести проверку всех типов ASIC в соответствии с этим пунктом. Мотивы выбора определенных ASIC могут быть очень различными. В общем случае ASIC имеют следующие преимущества:

- Как правило, ASIC охватывает функции многих стандартных элементов, вследствие чего вероятность ошибки схемы падает, она становится надежнее.

- Уменьшаются затраты на разводку на печатной плате. При определенных обстоятельствах могут быть использованы более дешевые печатные платы.
- Скопировать ASIC сложно, поскольку ее функции снаружи просматриваются с трудом. Вследствие этого достижения, полученные при разработке, легче уберечь от конкурентов.

ASIC тестируются легче, чем схемы из большого числа стандартных компонентов, если соответствующие возможности тестирования были учтены при разработке.

Таблица 14.2. Сравнение стандартных компонентов и различных типов ASIC.

Параметр	Стандартные компоненты	Программируемые пользователем ИС	ASIC с полузаказным проектированием	ASIC с полностью заказным проектированием
мощность	средняя–высокая	средняя	высокая	очень высокая
стоимость разработки	низкая	низкая	средняя–высокая	высокая–очень высокая
стоимость масок	—	—	низкая–высокая	высокая
продолжительность разработки	короткая	средняя	средняя	средняя–очень высокая
стоимость одного элемента	низкая	высокая	средняя	низкая
плотность интеграции	высокая	низкая	высокая	очень высокая

14.2. Программируемые логические ИС (PLD)

Программируемые логические ИС (PLD) являются по своей сути стандартными схемными элементами, конфигурацию которых заказчики могут изменять в своих целях. Их электрические характеристики хорошо известны, поэтому можно получить очень достоверные модели для моделирования. Программируемые логические

ИС могут быть разработаны достаточно надежно на основании моделирования. Стоимость одного образца, отнесенная к количеству вентиляей, может оказаться очень малой, так как PLD могут быть изготовлены с высокой степенью интеграции. При этом с помощью небольшого ресурса проектирования можно добиться высокой степени использования вентиляей. Поэтому PLD хорошо подходят для вариантов применения, когда требуется малое число образцов. В число их преимуществ входит и то, что они могут быть скорректированы за короткий срок.

К преимуществам программируемых логических ИС относится также то, что они тестируются изготовителем. Пользователю приходится проверять только топологию.

Ниже программируемые логические ИС обозначаются общим термином «programmable logic devices» (PLD).

Благодаря возможности конфигурации логики пользователем в PLD объединяются преимущества стандартных схемных узлов, характеристики которых оптимально согласованы с потребностями заказчиков.

14.2.1. Типы PLD

Эти типы различаются по способу программирования:

- Пережигаемые перемычки (fuse link), электрические пережигаемые соединения.
- Без пережигаемых перемычек (anti fuse), электрически формируемые соединения.
- 1-битовые RAM-ячейки; триггеры.
- EPROM-ячейки: долговременное хранение заряда, информация может стираться при воздействии ультра фиолетового облучения.
- EEPROM-ячейки: долговременное хранение заряда, информация может стираться электрическим путем.

В PLD используются программируемые И- и ИЛИ-матрицы. В зависимости от структуры можно выделить следующие PLD:

PLA	— программируемые И- и ИЛИ-матрицы;
PAL	— программируемая И-матрица, фиксированная ИЛИ-матрица;

- GAL — такие же, как PLA, но с дополнительными программируемыми выходными схемами;
- EEPROM — программируемая ИЛИ-матрица, фиксированная И-матрица;
- FPGA, LCA — электрически программируемая энергозависимая логическая матрица;
- CPLD, EPLD — электрически программируемая энергонезависимая логическая матрица, информация, стирается ультрафиолетовым излучением или электрически.

Ниже дано упрощенное представление используемых в PLD И- и ИЛИ-вентилей, которые имеют, как правило, большое число входов (рис. 14.2).

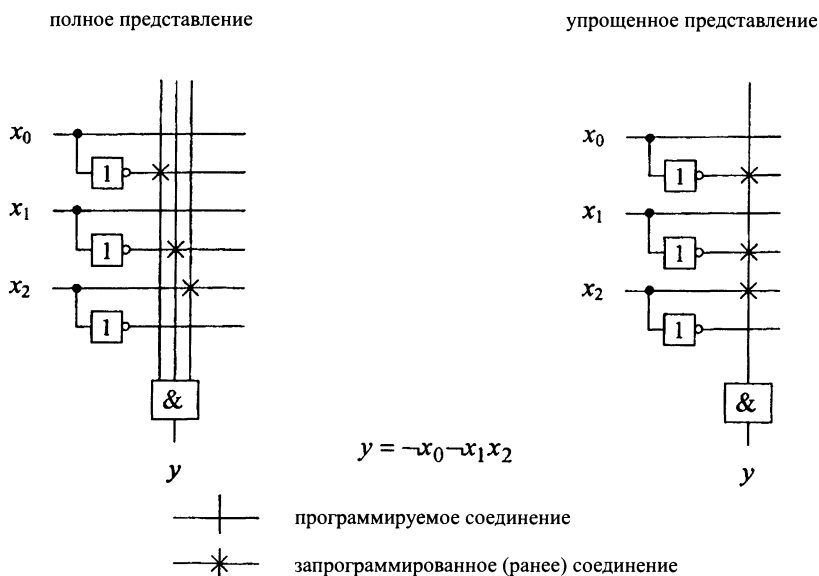


Рис. 14.2. Полное и упрощенное представление И-вентилей в PLD.

14.3. ROM, EPROM, EEPROM

На основе энергонезависимых ЗУ ROM, EPROM, EEPROM и т. д., могут быть выполняться логические схемы. Пусть, например, с помощью ROM с емкостью 16×2 бит должна быть реализована функция с четырьмя входными переменными и двумя выходными переменными. Под каждым из 16 адресов в ROM могут храниться запомнены

значения обеих функций. При этом еще возможно выполнение логической операции. Таким образом ROM, используемая как логическая схема, универсальна. Но оно может оказаться не эффективным, если функция имеет очень мало нулей в таблице истинности.

Применение ROM предпочтительно в следующих случаях:

Когда функция задана таблицей истинности. Тогда функция может быть без дальнейшей обработки введена в ЗУ.

Когда функция нуждается в большом количестве термов логического произведения (булевых логических термов).

Когда очевидно, что схема должна часто изменяться.

Когда в таблице истинности имеется большое число единиц.

ROM можно представить себе как логическую схему, в состав которой входят: фиксированная И-матрица, осуществляющая адресное декодирование, и программируемая масками ИЛИ-матрица (рис. 14.3).

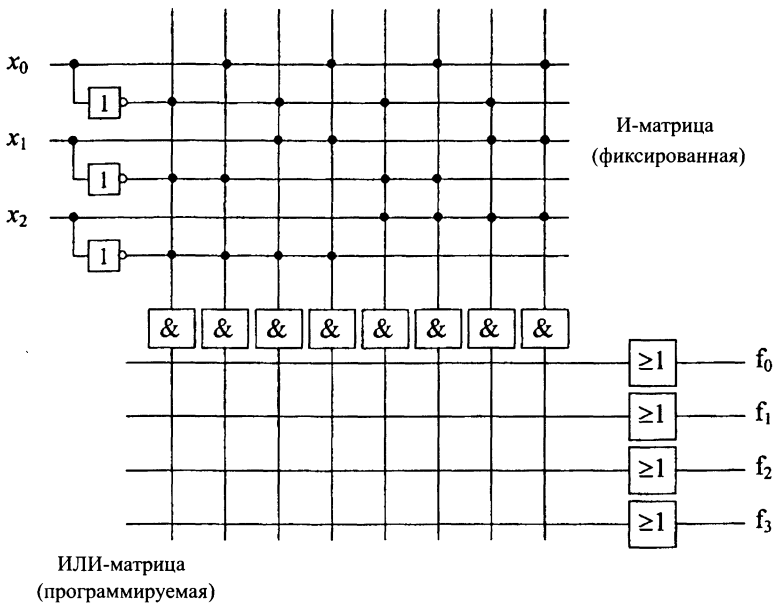


Рис. 14.3. Представление ROM с емкостью 8×4 (32 бит) с И- и ИЛИ-матрицами.

14.4. PLA

Как показано на рис. 14.4, программируемые логические матрицы PLA (programmable logic array) включают программируемую И-матрицу. С помощью PLA можно непосредственно создавать нормальную DNF, причем термы логического произведения (product term) реализуются с помощью И-матрицы, а суммарные термы с помощью ИЛИ-матрицы. При этом число термов логического произведения меньше 2^n . Общие термы логического произведения могут быть поданы на многие логические входы.

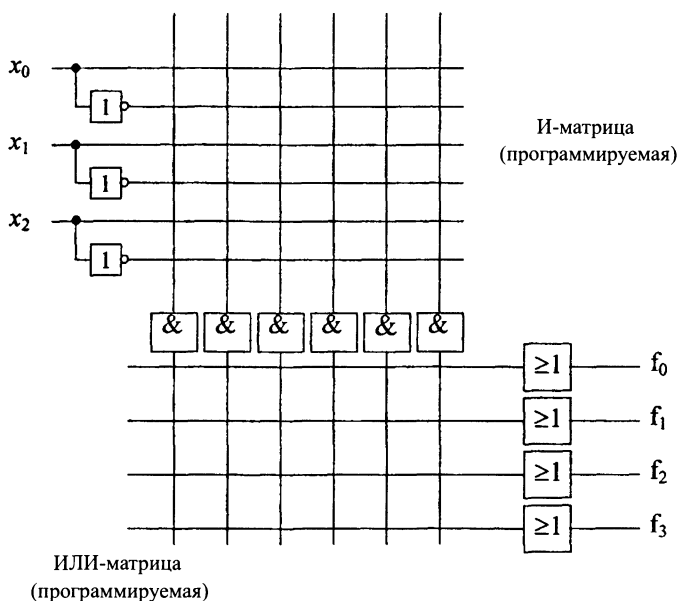


Рис. 14.4. Общая схема PLA.

В общую схему, приведенную на рис. 14.4, могут быть введены некоторые улучшения.

Выходы должны иметь возможность подвергаться логической операции НЕ, так как для многих функций дополнение реализуется с меньшими затратами.

Часто необходимы выходы типа tri-state («с темя состояниями»). В этом случае имеет смысл, чтобы некоторые выходы могли бы быть использованы как входы.

При определенных обстоятельствах выходные сигналы должны подвергаться промежуточному запоминанию. Поэтому многие PLA имеют на выходе триггеры.

С помощью введения инвертированной обратной связи из матрицы ИЛИ могут быть при определенных обстоятельствах сэкономлены многие термы логического произведения.

Обсудим полезность и реализацию улучшений на основе двух следующих схем.

Входящий в PLA вентиль «исключающее ИЛИ» (EXOR), показанный на рис. 14.5, служит для управления полярностью выхода. Если на его втором входе имеет место 0, вентиль работает как буфер, если же второй вход находится в состоянии 1, выход становится инвертированным. Следовательно, всегда можно выбрать между реализацией функции и ее инвертированием, экономя таким образом термы логического произведения.

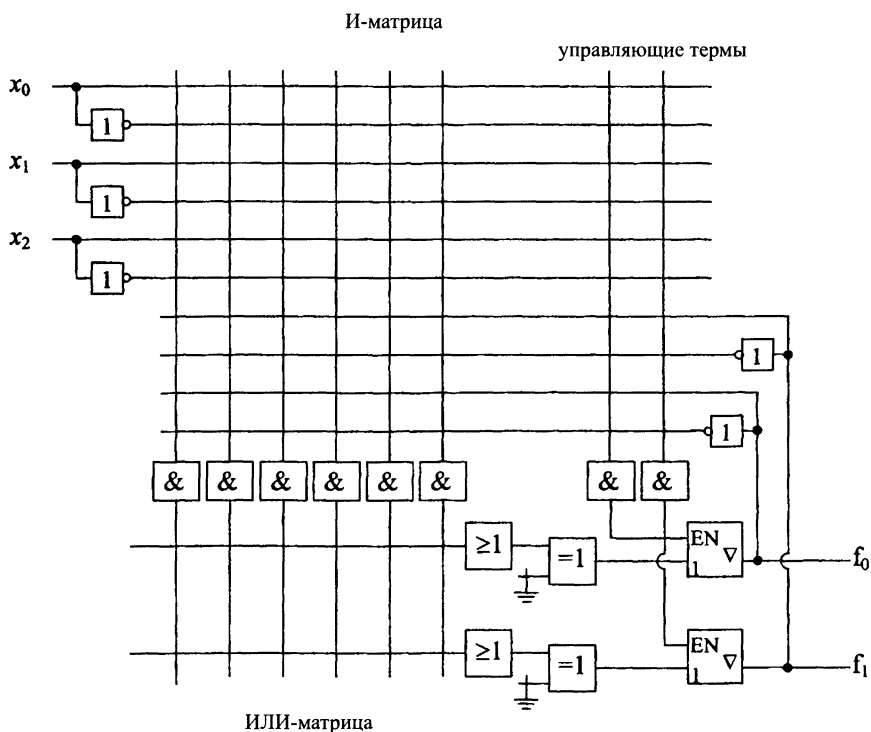


Рис. 14.5. PLA с вентилями EXOR и двунаправленным выходом типа tri-state.

Кроме того, в PLA на рис. 14.5 используются на выходах вентиля типа tri-state. Как альтернативный вариант можно рассматривать PLA, у которых разрешающие (enable) входы управляются дополнительными выходами матрицы ИЛИ, либо применяются дополнительные выводы. Поскольку выходы выполнены как триггеры с вы-

ходами типа tri-state, их можно использовать как двунаправленные интерфейсы. Соответствующие входы также ведут в И-матрицу. Повышенная гибкость обеспечивает лучшее использование PLA.

В PLA на рис. 14.7 имеется инвертированная обратная связь, ведущая из матрицы ИЛИ, которую также называют дополнительной (обращенной) матрицей (complement array). Эта обратная связь способствует экономии термов логического произведения в тех случаях, когда обрабатываются задачи, при решении которых выходы должны принимать определенные значения с рядом исключений.

Пусть, например, должен быть разработан сегментный декодер для представленных с помощью десятично-двоичного кода (BCD) чисел, который при входных значениях более 1000_В должен показывать на дисплее *E* как обозначение ошибок. Определение цифр с помощью соответствующего кода показано на рис. 14.6.

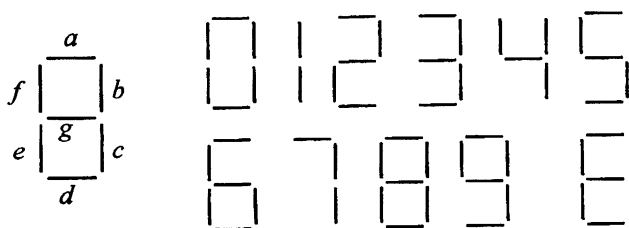


Рис. 14.6. Определение 10 цифр и *E* как обозначения ошибок для случая 7-сегментной индикации.

Для цифр от 0 до 9 требуются 10 термов логического произведения. Они подаются на вход дополнительной матрицы. Дополнительная матрица не реагирует на появление одного из этих термов. Если же не задействован ни один из термов логического произведения, а подана псевдотетрада (двоичные числа между 10 и 15), то в этом случае приходят в действие сегменты, формирующие букву *E*.

На рис. 14.8 показана в упрощенной форме PLA, служащая для реализации управляющей схемы (драйвера). PLA данного вида обозначают как устройства задания последовательности (sequencer). В показанную схему входят:

- Регистр состояний с выходами P_0 , P_1 и P_2 . Выходы регистра подведены к И-матрице.
- Вход $P/\neg E$ может быть запрограммирован так, что он действует либо как вход enable, управляющий tri-state-буфером, либо как вход preset («предварительная установка») для D-триггеров.

- Добавочная матрица может быть использована для того, чтобы устанавливать определенное состояние при включении или при ошибках. Добавочная матрица (complement array) срабатывает тогда, когда ни одно из непосредственно применяемых состояний не наступает при включении.

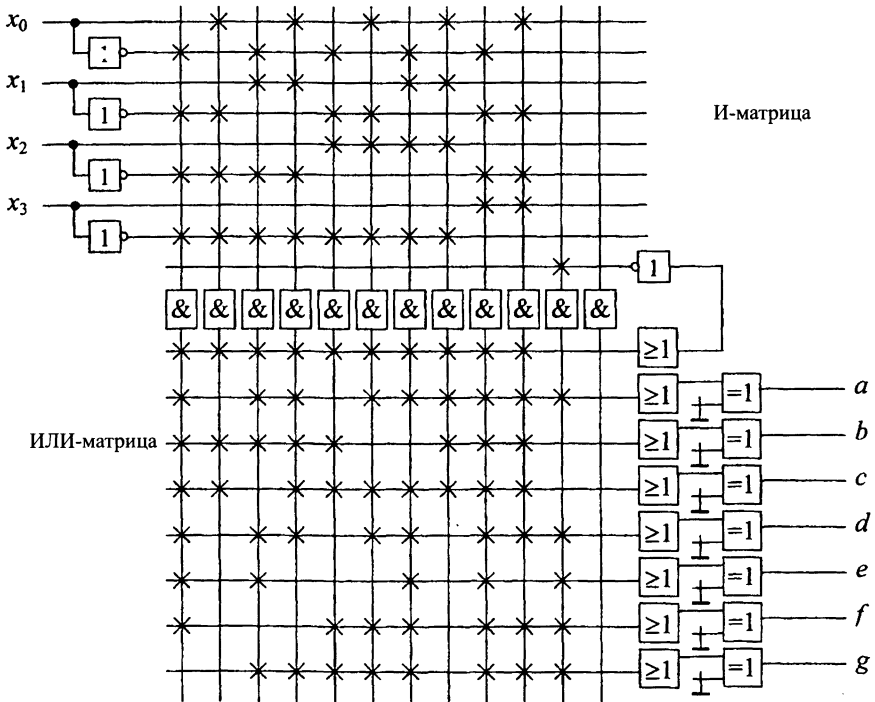


Рис. 14.7. PLA с инвертированной обратной связью, выведенной из матрицы ИЛИ (пример с индикацией семью сегментами).

Пусть, например, на основе показанной на рис. 14.8 PLA, должен быть спроектирован двоичный счетчик по модулю 5, который для входного сигнала $x = x_0 = 0$ считает в прямом направлении и для $x = x_0 = 1$ в обратном направлении. На основе этой информации получаем табл. 14.3.

На основе таблицы состояний может быть разработана таблица запуска (табл. 14.4) для RS-триггеров данной схемы.

Из таблицы запуска может быть выведена KV-диаграмма (рис. 14.9). В этой KV-диаграмме (диаграмме Карно-Вейча) находятся свободные поля, которые принадлежат состояниям, не возникающих в нормальных циклах счета. Из этих состояний, которые могут установиться при включении или при возникновении помехи, счетчик дол-

жен переходить в последовательные состояния $(P_0, P_1, P_2) = (0, 0, 0)$.

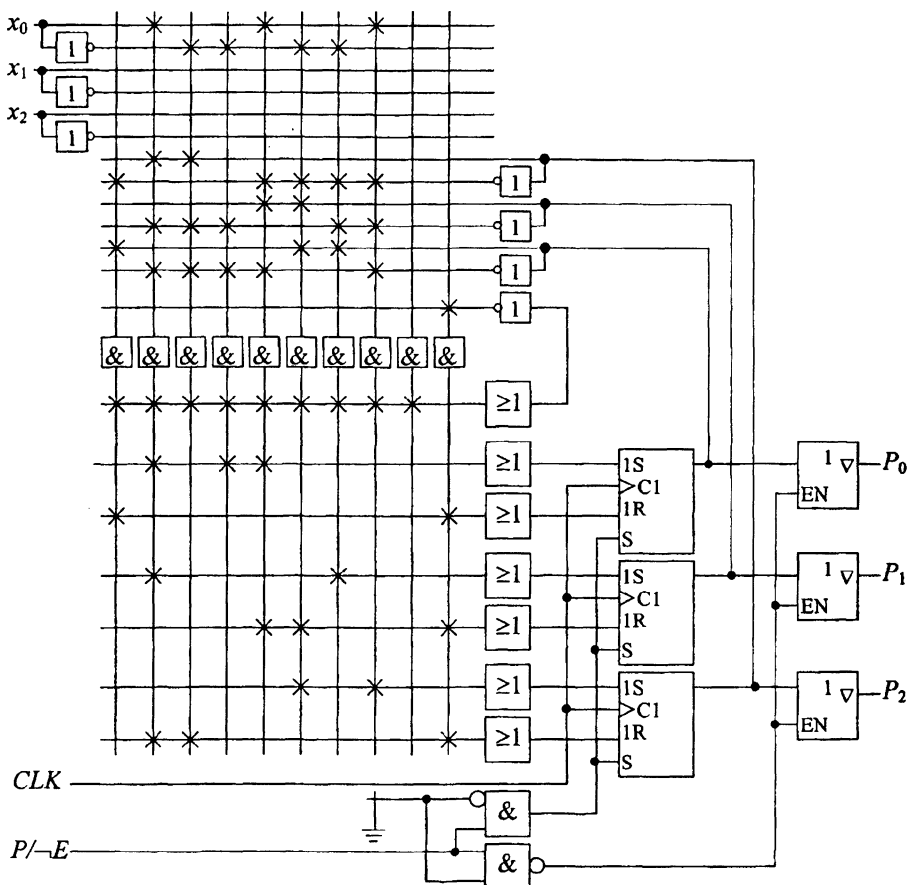


Рис. 14.8. PLA для реализации управляющих схем (устройств задания последовательности, sequencer), программирование для данного примера реверсивный двоичный счетчик для счета по модулю 5.

Таблица 14.3. Таблица последовательности состояний двоичного счетчика по модулю 5.

P_2^m	P_1^m	P_0^m	$x_0 = 0$			$x_0 = 1$		
			P_2^{m+1}	P_1^{m+1}	P_0^{m+1}	P_2^{m+1}	P_1^{m+1}	P_0^{m+1}
0	0	0	0	0	1	1	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	0	1	0
1	0	0	0	0	0	0	1	1

Таблица 14.4. Таблица запуска для RS-триггеров двоичного счетчика по модулю 5.

P_2^m	P_1^m	P_0^m	$x_0 = 0$			$x_0 = 1$		
			R_2S_2	R_1S_1	R_0S_0	R_2S_2	R_1S_1	R_0S_0
0	0	0	d0	d0	01	01	d0	d0
0	0	1	d0	01	10	d0	d0	10
0	1	0	d0	0d	01	d0	10	01
0	1	1	01	10	10	d0	0d	10
1	0	0	10	d0	d0	10	01	01

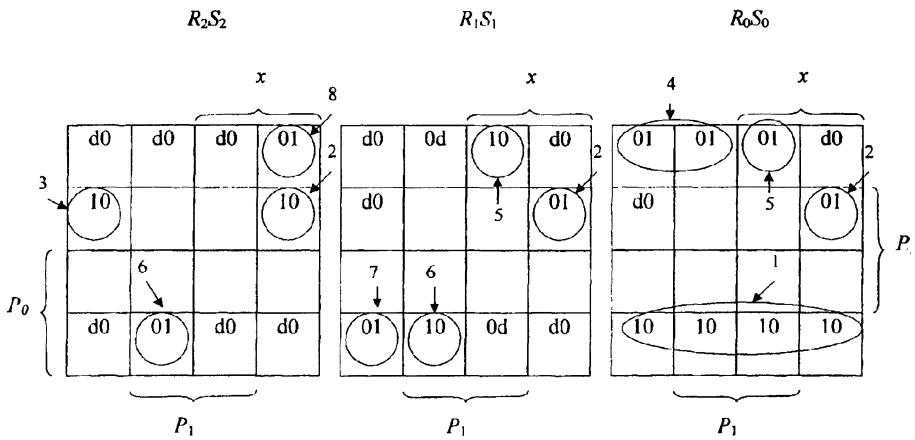


Рис. 14.9. Диаграмма Карно-Вейча для запуска RS-триггеров.

Из диаграммы Карно-Вейча получаем уравнения:

$$R_0 = \underbrace{P_0 \neg P_2}_1 \quad (14.1)$$

$$S_0 = \underbrace{\neg P_0 \neg P_1 P_2 x}_2 \vee \underbrace{\neg P_0 P_1 \neg P_2 x}_5 \vee \underbrace{\neg P_0 \neg P_1 \neg x}_4 \quad (14.2)$$

$$R_1 = \underbrace{\neg P_0 P_1 \neg P_2 x}_5 \vee \underbrace{P_0 P_1 \neg P_2 \neg x}_6 \quad (14.3)$$

$$S_1 = \underbrace{\neg P_0 \neg P_1 P_2 x}_2 \vee \underbrace{P_0 \neg P_1 \neg P_2 \neg x}_7 \quad (14.4)$$

$$R_2 = \underbrace{\neg P_0 \neg P_1 P_2 x}_2 \vee \underbrace{\neg P_0 \neg P_1 P_2 \neg x}_3 \quad (14.5)$$

$$S_2 = \underbrace{P_0 P_1 \neg P_2 \neg x}_6 \vee \underbrace{\neg P_0 \neg P_1 \neg P_2 x}_8 \quad (14.6)$$

Импликанты, маркированные цифрами от 1 до 8 реализуются с помощью первых 8 И-вентилей И-матрицы, как это показано на рис. 14.2. К выходам этих 8 И-вентилей подключена малая комплементарная матрица. При этом данное подключение осуществлено таким образом, чтобы эта комплементарная матрица при состояниях нормального цикла не реализована. Во всех других случаях комплементарная матрица через И-вентиль 10 в И-матрице подает 1 на R- вход RS-триггера и возвращает его в исходное состояние. ИЛИ-матрица программируется в соответствии с уравнениями от (14.1) до (14.6).

Вход $P/\neg E$ программируется таким образом, чтобы он был решающим входом для выходов. Соответствующий пункт программирования часто опускается, что соответствует 1.

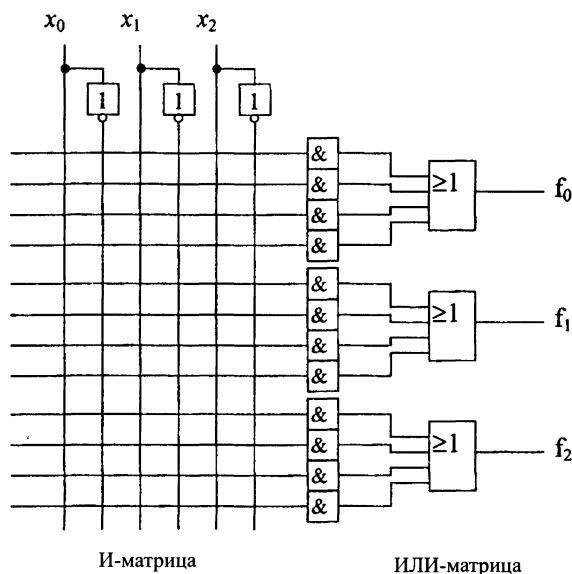


Рис. 14.10. PAL с 3 входами, 3 выходами и с 4 термами логических произведений на каждом из выходов.

14.5. PAL

PAL (Programmable Array Logic, «Программируемая матричная логика»), показанная на рис. 14.10, является упрощенным вариантом PLD). Она имеет только лишь программируемую И-матрицу. ИЛИ-матрица ограничена несколькими (на рис. 14.10 это четыре) термами логического произведения.

С помощью PAL многие функции могут быть реализованы с меньшими аппаратными затратами, чем в случае PLA. Однако структу-

ра PAL не дает возможности пользоваться совместно общими термами логических произведений многих функций, как в случае PLA.

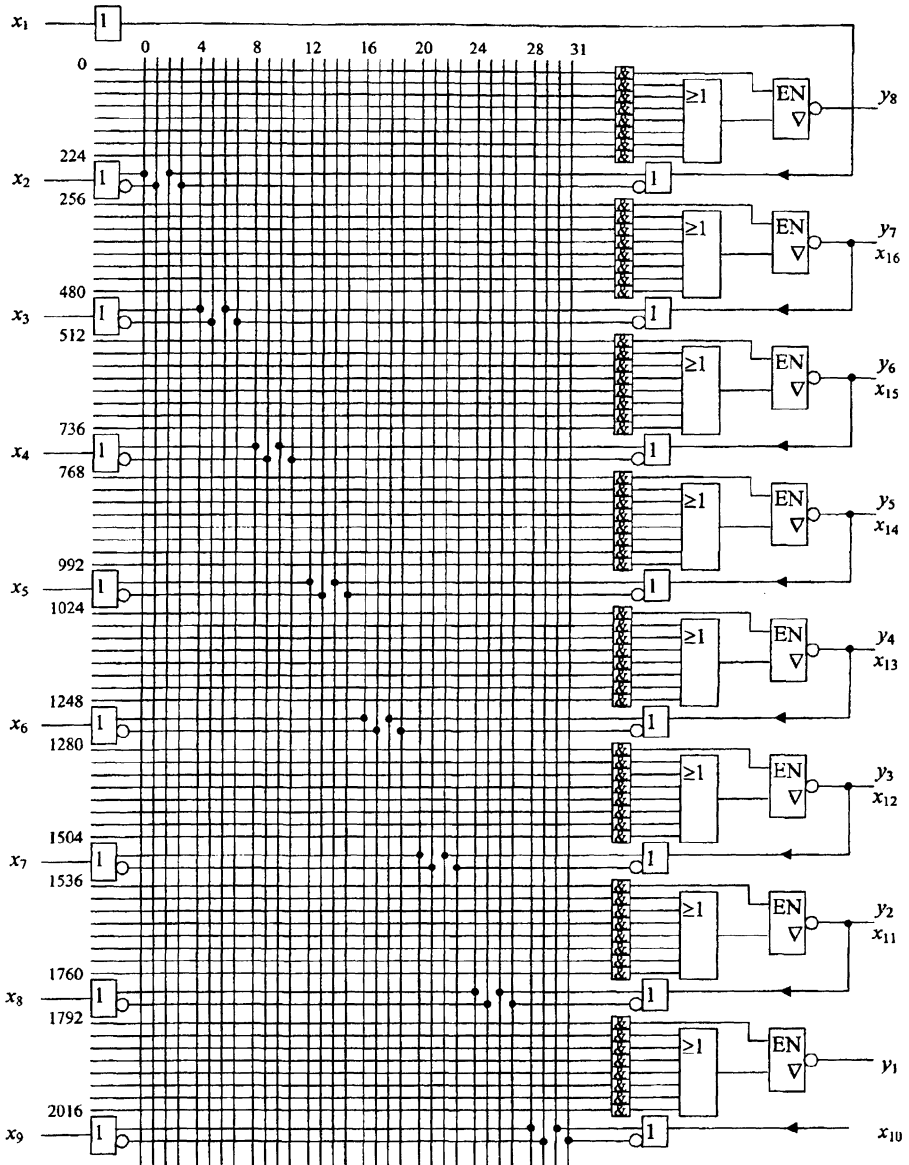


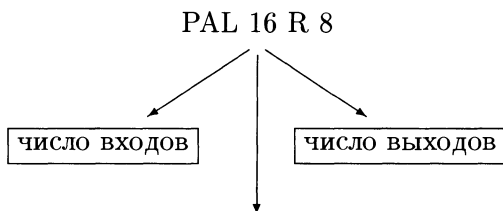
Рис. 14.11. PAL 16 L 8.

В показанной на рис.14.11 PAL 18P8 для управления полярностью используются вентили «исключающее ИЛИ» (EXOR-gate). Выходы типа Tristate («с тремя состояниями») управляются термами

логических произведений (product term). Выходы могут быть использованы и как входы, благодаря чему спектр вариантов внедрения PAL может быть увеличен.

В настоящее время для некоторых PAL реализована возможность целенаправленно придавать термы логических произведений отдельным выходам (product term steering). Это конечно не означает, что термы логических произведений, относящиеся к различным функциям, могут быть использованы совместно.

Для PAL применяется единая схема обозначений:



<i>R</i>	— синхронный выход регистра
<i>Ra</i>	— асинхронный выход регистра
<i>S</i>	— комбинационный выход с управлением термами логического умножения (product term steering)
<i>H</i>	— выход активный, уровень H (high)
<i>L</i>	— выход активный, уровень L (low)
<i>V</i>	— изменяющаяся выдача сигналов
<i>X</i>	— вентиль EXOR и выход регистра
<i>XP</i>	— вентиль EXOR и программируемая выходная полярность
<i>XPR</i>	— вентиль EXOR и программируемая полярность регистра
<i>C</i>	— дополнительный выход (complement output)
<i>P</i>	— выход с программируемой полярностью

14.6. GAL

Наименование GAL (generic array logic, «базовая матричная логика») соответствует улучшенному варианту PAL, в котором на выходах PAL имеются программируемые ячейки (OLMC, output logic macro cell). Эти ячейки могут быть запрограммированы для работы в качестве входа, выхода или выхода с тремя состояниями.

GAL изготавливаются по технологии EECMOS, объединяющей КМОП-процесс с технологией электрически стираемых ЗУ (EEPROM).

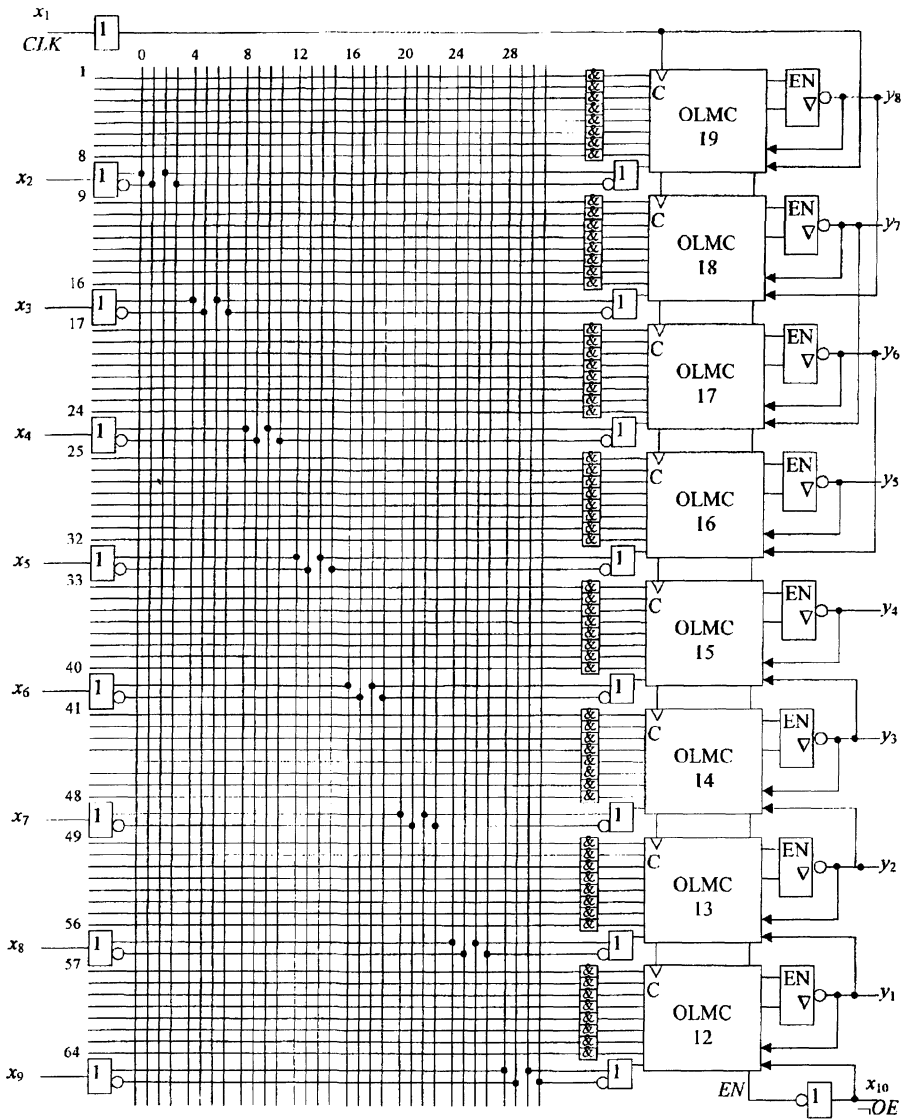


Рис. 14.12. Структура GAL 16V8.

Они имеют поэтому относительно малую рассеиваемую мощность и высокое быстродействие. Данные схемы узлы можно многократно программировать (до 2000 раз) и стирать. К преимуществам GAL относится и то, что программируемость может быть проверена изготовителем. Так как конфигурацию выходов можно изменять, малого числа GAL достаточно, чтобы заменить большой спектр программируемых логических ИС (PLD). Структура GAL 16V8 показана

на на рис. 14.12. В GAL 16V8 входят 8 выходных программируемых макро ячеек (OLMC). В каждой OLMC может быть соединено по схеме ИЛИ 8 термов логического произведения. Вход x_1 может быть использован как тактовый вход CLK , вход x_{10} — как «разрешающий выход» (output enable) вход — OE .

Таблица 14.5. Значения сигналов SYN , ACO и $AC1(n)$.

	Функция OLMC	SYN	ACO	$AC1(n)$	Описание
Простой режим (логическая схема)	Вход	1	0	1	— Выход типа tri-state, всегда высокоомный; — Входной сигнал идет в соседнюю ячейку; — Возможно только для OLMC 12-14 и 17-18.
	Выход	1	0	1	— Выход всегда включен; — Никакой обратной связи; — Имеются в наличии 8 термов логического произведения для логики; — Возможно для всех OLMC.
Комплексный режим (двухнаправленная схема)	Вход/выход типа tri-state	1	1	1	— Освобождение после термина логического произведения; — В наличии 7 термов логического произведения для логики.
Режим с регистрами (схема с регистрами)	Регистр, вход/выход типа tri-state	0	1	1	— Действует CLK (тактовый сигнал); — Освобождение выхода с помощью термина логического произведения; — В наличии 7 термов логического произведения для логики.
	Регистр, вход/выход типа tri-state	0	1	1	

Один из OLMC представлен на рис. 14.13. Некоторые OLMC могут эксплуатироваться двунаправленно, только OMLC 15 и 16 могут функционировать исключительно как выход. Все OLMC имеют цепи обратной связи, ведущие в И-матрицу. Как можно видеть на рис. 14.13, имеется 16 входов, так что с учетом инверсии получаем 32 различных столбца И-матрицы.

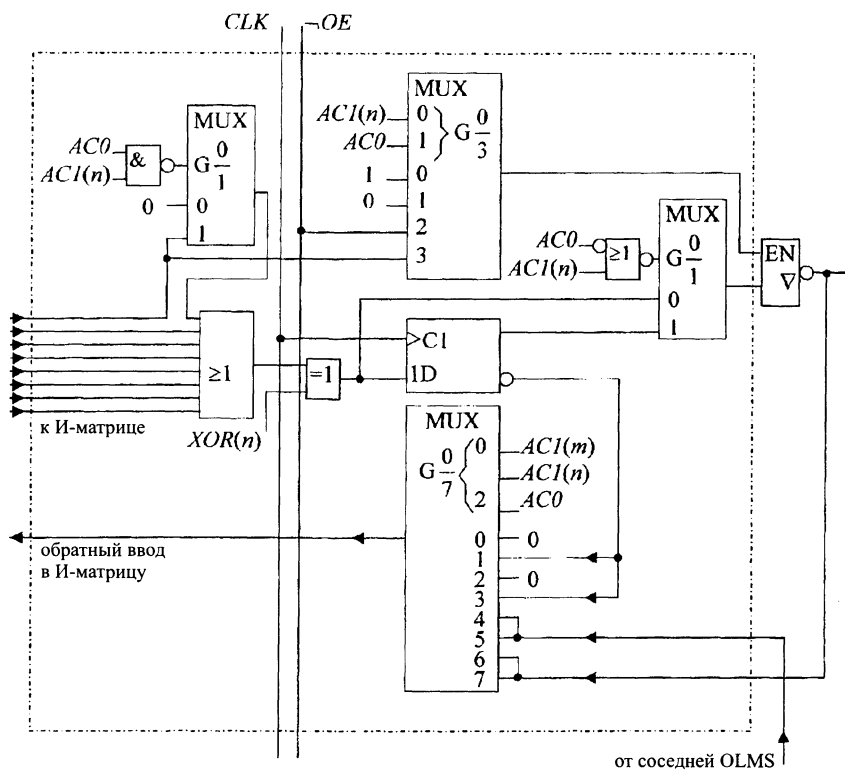


Рис. 14.13. OLMC N_n , входящая в GAL 16V8, соседняя OLMC имеет N_m .

Ячейка OLMC, входящая в GAL 16V8, управляется сигналами $XOR(n)$, SYN , ACO и $ACI(n)$. SYN и ACO оказывают действие на все OLMC, $XOR(n)$ и $ACI(n)$ индивидуально подбираются для каждой OLMC. $XOR(n)$ управляет полярностью выхода. $XOR(n)$ означает активный сигнал LOW .

14.7. Программирование логических блоков PLD

Программирование PLD-блоков проводится на основе массива данных, соответствующих формату JEDEC (Joint Electronic Devices Engineering Council). На рис. 14.14 показан пример, относящийся к

GAL 16V8. Массив данных имеет, в начале каждой строки строчный адрес, который начинается с *L.

В каждой строке находится 32 бита, поскольку GAL 16V8 совместно с 32 входами охватывает 32 столбца в И-матрице, как это можно видеть на рис. 14.12. Значение «1» означает, что соединение в соответствующем месте разорвано, «0» означает, что оно существует.

```
*L0000 111111011011110111011111111111
*L0032 111101011111101111101111011111
*L0064 1111111111010101111110101101111
*L0096 111011110111111110111111111111
*L0128 111111110111110111111111011111
```

Рис. 14.14. Извлечение из JEDEC-файла, соответствующего PLD GAL 16V8.

Строки JEDEC-файла, описывающего GAL 16V8, заполнены следующим образом:

0000-2047	Соединения логической матрицы в соответствии с приведенным выше примером.
2048-2055	$XOR(n)$ -бит для OLMC 12–19.
2056-2119	Электронная сигнатура: 64 бита для собственного применения.
2120-2127	$AC(n)$ -бит для OLMC 12–19.
2128-2191	Освобождение термов логического произведения от $PT0$ до $PT63$.
2192	SYN -бит.
2193	ACO -бит.

Для программирования GAL вводится в режим программирования с помощью подведения определенного напряжения к одному из выводов (здесь вывод 2 = 16,8 В). На 6 выводах (вывод 18, выводы от 3-го до 7-го) набираются строки матрицы ЗУ и в соответствии с тактовым сигналом $SCLK$ (вывод 8) биты, которые лежат на контакте $SDIN$ (вывод 9), передвигаются в сдвиговый регистр GAL, который снабжен электронной защитой от копирования. Когда введен бит защиты, запрограммированные данные не могут быть считаны. Остается возможным только стирание.

Гарантируется работоспособность GAL течение 10 лет. Процесс программирования продолжается в течение нескольких секунд.



14.7.1. Тестирование

При применении управляющих схем (драйверов) важно протестировать, исключены ли состояния, в которые драйвер при нормальной эксплуатации не должен входить. Для этого целесообразна загрузка всех регистров произвольно выбранным значением. Для этого GAL 16V8 имеет схему, с помощью которой регистр может быть загружен. Этот вид работы активируется приложением 15 В на *PRLD* (вывод 11).

Теперь в соответствии с тактовым сигналом *DCLK* (вывод 1) через последовательный вход *SDIN* (вывод 9) данные могут быть продвинуты через сдвиговый регистр. С последовательного выхода *SDOUT* (вывод 12) данные могут быть сняты. Используются только те ячейки регистра, которые могут быть конфигурированы как выход регистра.

14.8. Программируемые полем вентильные матрицы (FPGA)

Программируемые полем вентильные матрицы (FPGA, field programmable gate arrays) представляют собой стандартные логические блоки, конфигурацию которых задает пользователь. Они состоят из многих PLD, взаимодействия между которыми обеспечивается матрицей соединений.

FPGA имеют следующие преимущества:

- При применении специального программного обеспечения FPGA можно легко запрограммировать.
- Не требуется какое-либо складирование схем у заказчика, поскольку стандартные логические блоки легко доступны.
- При заказном же проектировании необходимо учесть время доставки продолжительностью в несколько месяцев, которое пройдет после выдачи заказа изготовителю интегральных схем.
- Тестирование проводится просто.
- FPGA тестируются изготовителем, поэтому пользователю достаточно проведения сокращенной программы испытаний.
- В противоположность дискретной реализации требуется очень малое количество элементов, благодаря чему схема становится надежнее.

- Затраты на разводку проводящих дорожек меньше. Благодаря этому при определенных обстоятельствах могут быть использованы более дешевые печатные платы.
- В FPGA может быть реализована оптимальная архитектура. Поэтому они обладают высоким быстродействием.
- Изменения проводятся легко, так как для этого достаточно только изменения программы.

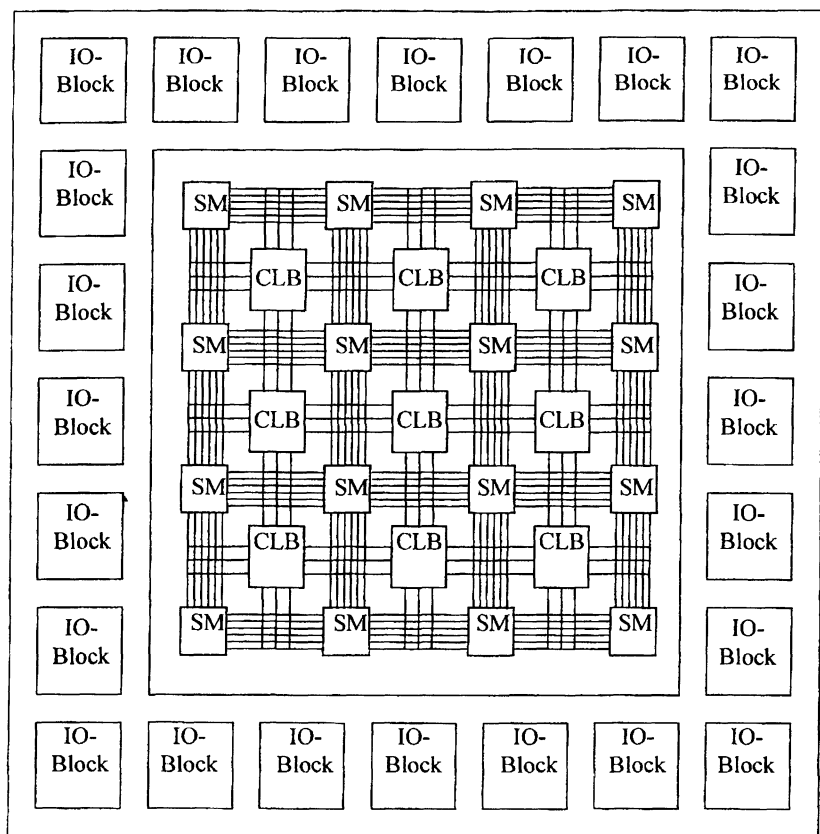


Рис. 14.15.

14.8.1. Структура FPGA

FPGA представляет собой программируемые в соответствии с применением матрицы из логических блоков, выполненных большей частью в форме PLA (программируемых логических матриц). Логиче-

ская конфигурация устанавливается программой, которая хранится в статическом RAM, выполненном в кристалле. В начале работы статическое RAM загружается из ROM. В альтернативном варианте FPGA может быть запрограммирована, например, с помощью пережигаемых перемычек.

Ниже описывается семейство FPGA PC3000, выпускаемое компанией Xilinx ИЗГОТОВИТЕЛЕМ ПОЛУПРОВОДНИКОВЫХ ПРИБОРОВ. Речь идет о семействе logic cell arrays (матриц логических ячеек).

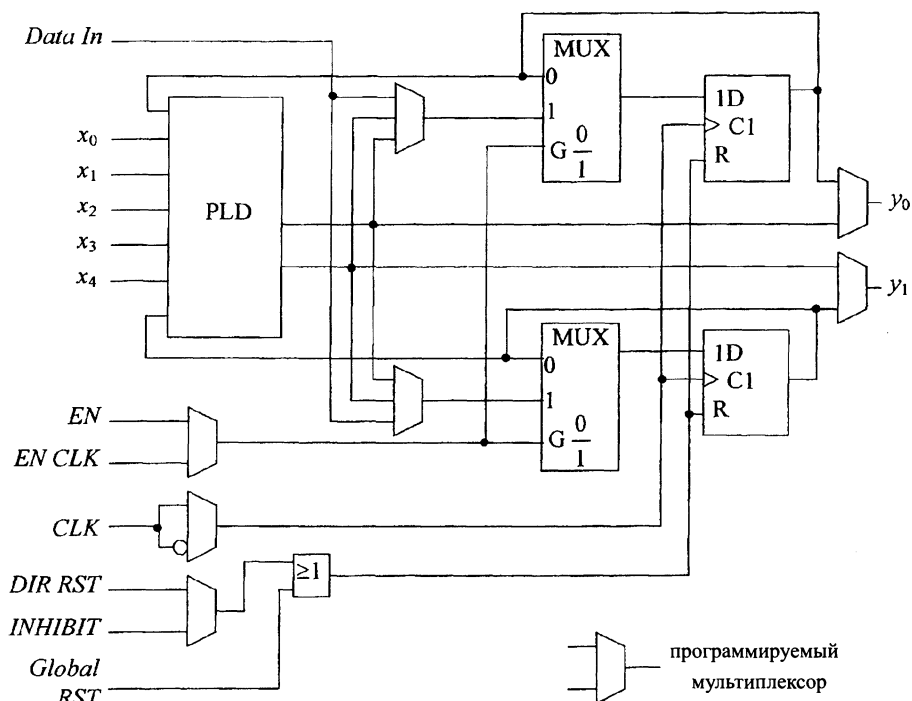


Рис. 14.16. Конфигурируемый логический блок (CLB) FPGA из семейства XC3000 компании Xilinx.

Архитектура FPGA может быть разделена на 3 типа блоков, из которых формируются различные конфигурации:

Логика сведена в образующие различные конфигурации блоки (CLB, configurable logic blocks). CLB соответствуют приблизительно одному PAL и могут быть запрограммированы с помощью конфигурационной программы. CLB размещены в форме матрицы в центральной части FPGA.

Входы и выходы реализованы с помощью блоков input/output (IO-blocks). Их можно включить как вход, выход, выход типа tri-

state или как двунаправленный интерфейс. IO-блоки размещены на краю FPGA рядом с выводами корпуса.

Соединительные линии реализованы в виде программируемой сети, которая подключает друг к другу CLB и IO-блоки. Программируемые переключательные матрицы располагаются между CLB.

14.8.2. Конфигурируемые логические блоки (CLB)

Все CLB идентичны. Они размещаются в виде матриц, соответствующих размерам FPGA. Так, например, модуль XC3090, входящий в семейство XC3000, содержит $12 \times 12 = 144$ CLB. Каждый CLB содержит логическую схему с 5 входами, которые выведены наружу, и с двумя входами, которые могут служить как входы для организации обратной связи с последующими схемами. Два D-триггера могут быть использованы для хранения признаков состояний. Но они могут быть загружены также и через вход *Data In*. Для управления модулем служат входы «прямой возврат в исходное состояние» (direct reset) *DIR DST*, вход тактового сигнала *CLK* и разрешающий (enable) тактовый вход *ENCLK*. Конфигурация логической функции логической схемы хранится в описанном конфигурационном ЗУ. Переключение схемы от D-триггера к драйверу (управляющей схеме) осуществляется с помощью адресации мультиплексора на основе конфигурационной программы.

14.8.3. IO-блоки

Для каждого подсоединяющегося вывода предусмотрен свой IO-блок (блок вход-выход). Например, модуль XC3090 имеет 166 выводов и столько же IO-блоков. В каждый IO-блок входят дополнительные элементы, обеспечивающие следующие возможности:

- Настройка на различные логические уровни (ТТЛ- или КМОП-уровни), которая может программироваться с помощью «подтягивающего» сопротивления (pull-up resistor).
- Защита от перенапряжений с помощью диодов.
- Промежуточное ЗУ для ввода и вывода данных.
- Но возможен также и прямой выход.
- Возможность программирования выхода как выхода типа tri-state. При этом через буфер выход делается активным или высокоомным.

14.8.4. Соединительные линии

Гибкость FPGA обеспечивается в значительной мере за счет программирования разводки. В nFPGA для этого имеется два уровня металлизации. Присутствующие линии могут быть иными способами соединены между собой на основе использования переключательных матриц и «программируемых точек связи» (PIP, programmable interconnect points). Входы CLB и IO-блоков могут быть запрограммированы так, что они будут подключены к окружающим их линиям.

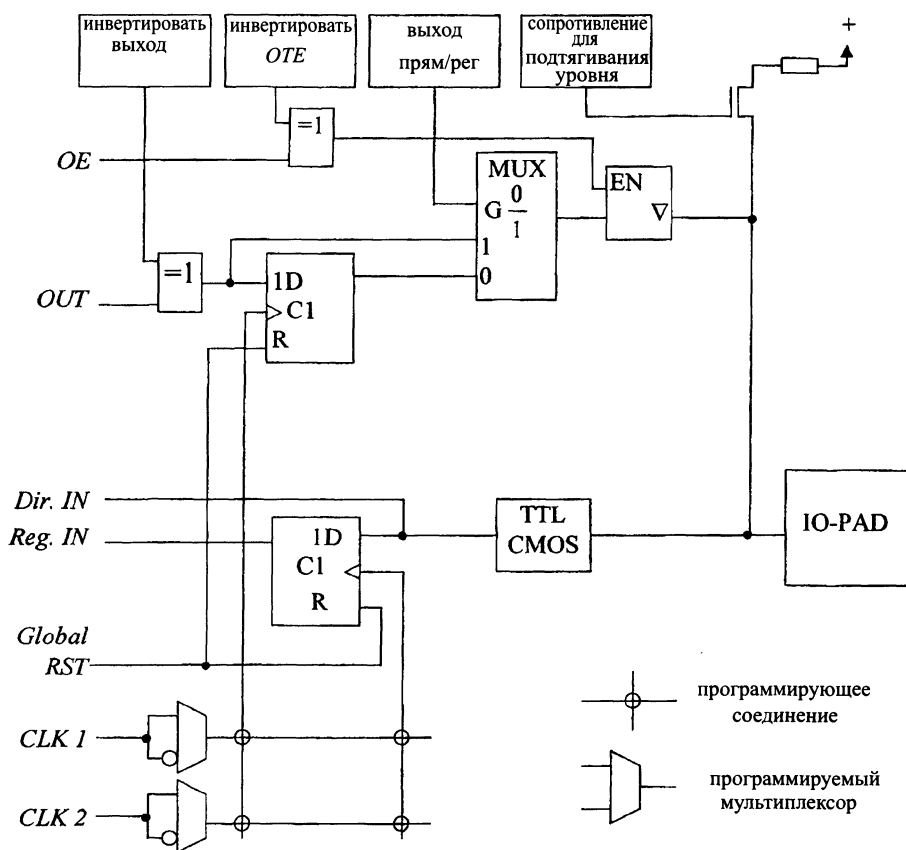


Рис. 14.17. Конфигурируемый блок IO семейства XC3000 компании Xilinx.

Имеются следующие типы соединительных элементов:

- В проходах между CLB размещено по 5 горизонтальных и по 5 вертикальных соединительных линий. Эти линии обозначаются как «связь общего назначения» (general purpose interconnect).

- Для быстрого соединения используется «прямая связь» (direct interconnection).
- Для соединения на большие расстояния используются «длинные линии».

14.8.5. Программирование FPGA

Программирование FPGA как правило начинается с задания схемы:

- либо в схематической форме, на основе графического ввода,
- либо на языке программирования, например, на языке VHDL (VLSI hardware description language) или на языке ABEL.
- В случае FPGA компании Xilinx эти данные конвертируются в список связей базовых модулей, именуемый XNF-файл. Он содержит данные всех вентилях и соединения между ними. Но при этом не учитываются физические данные соединительных линий, например, время задержки.

Список связей (net list) задает основу для перепроверки логической функций схемы.

В заключение на основе XNF-файла может быть проведено распределение вентилях по CLB (implementation, реализация). Одновременно устанавливаются соединительные линии. Этот процесс называют «размещение и маршрутизация» (place and route). При этом учитываются такие данные пользователя, как положение выводов, критические пути и т.д. Данные известной схемы запоминаются в LCA-файле (line control array). На основе этого возможно имитирование схемы с реальными значениями времен задержек.

В LCA-файле генерируется поток битов, содержащий данные о конфигурации. В FPGA данные запоминаются в триггерах, которые в процессе создания конфигурации включаются как длинный сдвиговый регистр. Число конфигурационных битов в зависимости от размеров FPGA варьируется между 12 кбит и 420 кбит. При определенной конфигурации FPGA могут быть включены друг за другом (цепочка типа «ромашка», daisy chain). При этом при загрузке поток битов пробегает вначале одну, потом другую FPGA и затем останавливается в конце цепочки сдвигового регистра второй FPGA. Число конфигурационных битов не зависит от степени использования FPGA.

14.9. EPLD

EPLD («стираемые программируемые логические устройства erasable programmable logic device»), которые также называют CPLD, создаются с помощью УФ-излучения. Они очень хорошо подходят для малых серий и лабораторных образцов.

14.9.1. Пример EPLD:CY7C343

EPLD CY7C343, разработанное компанией Cypress, выполнено по технологии, используемой для КМОП-EPROM.

Максимальная мощность рассеяния корпуса с 44-выводами, составляет 2,5 Вт. Могут быть использованы различные тактовые частоты, начиная с 100 МГц. Архитектура EPLD представлена на рис. 14.18.

Ядром EPLD является центральная матрица соединений, которые связывают между собой группы конструктивно объединенных элементов. В схему входят 28 двунаправленных интерфейсов, которые управляются через ИО-порты, и 8 простых входов. Логика собрана в 4 блока, каждый из которых содержит 16 макроячеек. Макроячейка содержит триггер и логику в форме PLA («программируемой логической матрицы»). Каждая из макроячеек имеет выход, ведущий к центральной матрице соединений. Некоторые из макроячеек имеют связь с ИО-блоками. Другие применяются как «скрытая логика».

14.10. Gate-Arrays

Gate-arrays представляют собой специализированные в соответствии с применением ИС (ASIC's), в составе которых изготовителем предлагается матрица из вентилях с фиксированной геометрией (т.н. master-slaves, пластины с базовыми кристаллами). Изготовитель структурирует в соответствии с требованиями заказчика только металлизированные межсоединения. Предлагаются gate-arrays, содержащие до 25000 вентилях. Для их изготовления используется, как правило, КМОП-технология.

Gate-arrays можно подразделить по видам обрабатываемых сигналов:

- цифровые
 - аналоговые
 - смешанные цифровые и аналоговые
- или по структуре:
- channelled gate-arrays, канализированные вентиляльные матрицы (соединительные проводники проходят по специальным каналам)

- sea-of gates, «море вентиляй» (соединительные проводники проходят по матричным ячейкам).

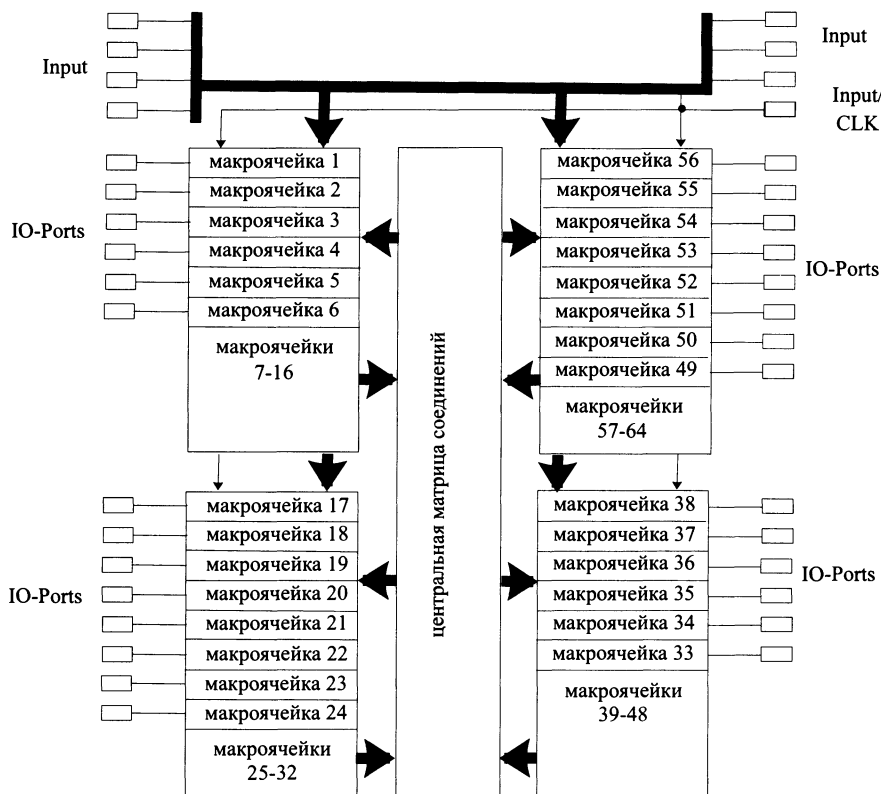


Рис. 14.18. Архитектура EPLD CY7C343 компании Cypress.

14.10.1. Структура канализированных вентиляных матриц

Gate-arrays (рис. 14.19) состоит из матрицы с однородными ячейками, периферийных ячеек и тестовых структур. Между ними расположены каналы для соединительных трасс. Матрица gate-arrays может иметь несколько сотен выводов.

Однородные матричные ячейки содержат по несколько пар p -и n -МОП транзисторов. Вначале, на базовом кристалле эти МОП-ПТ между собой не связаны. В подобном виде у изготовителя хранится запас полупроводниковых пластин с базовыми матричными кристаллами. Затем может быть осуществлено создание конкретной схемы в соответствии со специфическими требованиями заказчика. Для этого могут быть использованы один или несколько уров-

ней межсоединений. Таким образом из отдельных матричных ячеек могут быть сформированы, например, вентиль НЕ-И (NAND) или триггер. Обычно для реализаций требований заказчика используются от 1 до 8 фотошаблонов. Использование для соединений нескольких уровней помогает экономить площадь кристалла и повышать быстродействие.

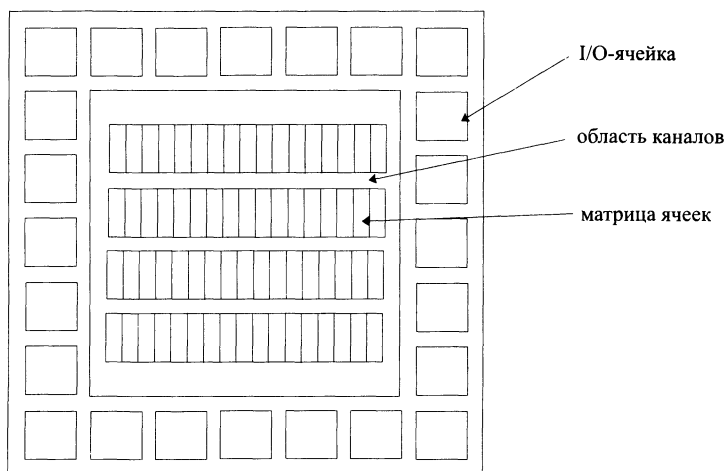


Рис. 14.19. Структура вентильной матрицы.

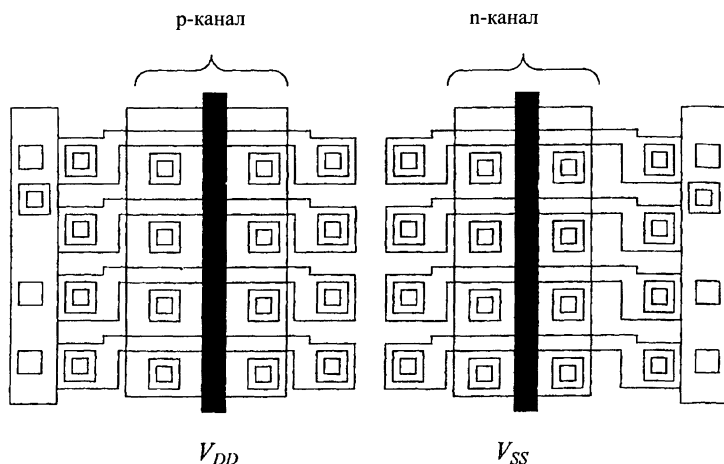


Рис. 14.20. Матричная ячейка вентильной матрицы (gate array). Другие матричные ячейки подключаются выше и ниже.

С использованием дополнительных заказных уровней металлизации могут быть образованы дополнительные библиотечные эле-

менты. Характеристики этих библиотечных элементов измеряются изготовителем и тестируются с высокой точностью. После этого заказчик может вызвать из компьютера символ соответствующей ячейки (например, НЕ-ИЛИ-вентиля) и соединить этот символ с имитационной моделью. Кроме того, запоминается топология металлизированных межсоединений, по которой позднее может быть изготовлен фотошаблон (маска).

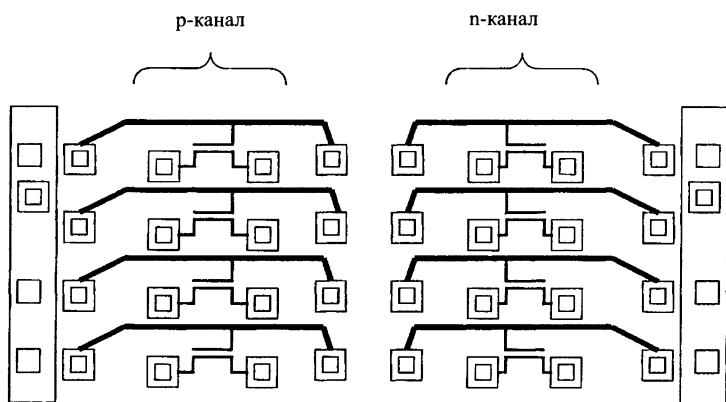


Рис. 14.21. Элементарные соединения в матричной ячейке, показанной на рис. 14.20.

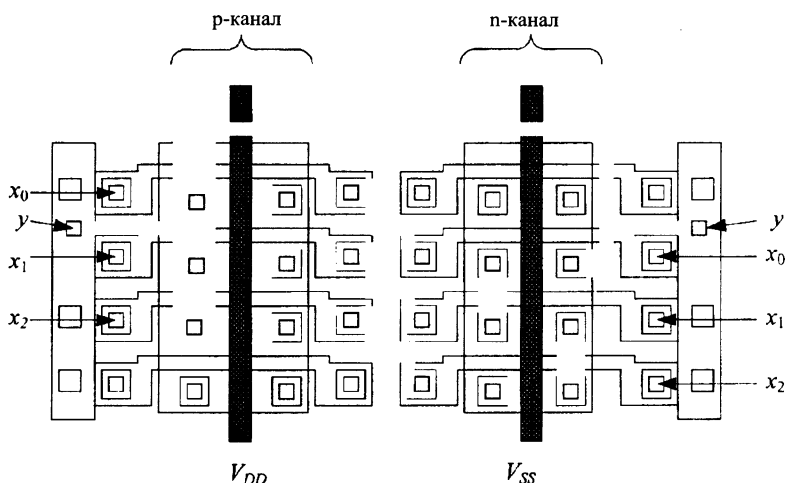


Рис. 14.22. Библиотечная ячейка: вентиль НЕ-ИЛИ с тремя входами x_0 , x_1 , x_2 и выходом y .

На рис. 14.22 представлена в качестве примера библиотечная ячейка, содержащая NAND-вентиль с тремя входами. Плоскость соеди-

нений, обеспечивающая конфигурацию заказной ИС, например инвертор, показана серым цветом. Не этого не потребуется вся ячейка. Поскольку для инвертору нужны только два транзистора, остальные компоненты не используются. Для мощного возбудителя параллельно подключается 2 дополнительных транзистора.

В периферийных ячейках содержатся мощные каскады, подключенные к выводам. На основе специализированной в соответствии с применением разводки периферийные ячейки могут быть включены в качестве входа, выхода или как двунаправленный интерфейс.

В тестовые и специализированные структуры входят, например, «модуль управления процессом» (PCM, process controle module), знаки совмещения, обозначение кристалла, номер версии. В каналах разводки располагаются проводники, связывающих отдельные вентили.

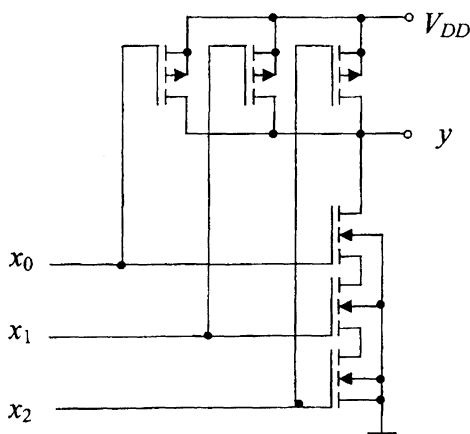


Рис. 14.23. Электрическая схема библиотечной ячейки: НЕ-И-вентиль с тремя входами x_0 , x_1 , x_2 и выходом y .

Комплексные ASIC можно экономично производить только с помощью поддерживаемых компьютерами инструментальных систем проектирования. Проектирование производится, как правило, на рабочих станциях заказчика, или, все в большей степени, на персональных компьютерах. Преимуществом вентильных матриц (gate arrays) является то, что изготовитель имеет готовую библиотеку с протестированными схемами матричных ячеек, в число которых входят, например, отдельные вентили (малый уровень интеграции, SSI), мультиплексоры (средний уровень интеграции, MSI) и малые микропроцессоры (СБИС, LSI). Пользователь может на основе этой библиотеки разрабатывать ИС на основе собственных проектов, по-

лагаясь при этом на то, что за счет применения точной модели ИС (почти) всегда начнет сразу функционировать.

14.11. ASIC со стандартными ячейками

ASIC со стандартными ячейками имеют больше степеней свободы, чем вентиляльные матрицы (gate-arrays).

Они имеют следующие особенности:

- ширина стандартных ячеек произвольна, фиксирована только высота;
- содержание ячеек является произвольным;
- возможны аналоговые функции;
- размеры каналов для разводки определяются требованиями заказчика (то есть являются заказными);
- особые функции: предлагаются ROM и RAM, выполненные по специальной технологии;
- все фотошаблоны являются заказными;
- размеры кристаллов (чипов) являются заказными.

ASIC со стандартными ячейками имеют следующие достоинства и недостатки:

- плотность интеграции выше, чем у gate-arrays, вследствие чего стоимость в пересчете на вентиль меньше;
- стоимость разработки и время разработки больше, чем в случае gate-arrays, вследствие чего ASIC со стандартными ячейками могут стать рентабельными только при производстве партии около 30000 ИС в год;
- изготовитель предоставляет программное обеспечение для тестирования библиотечных ячеек;
- сроки поставки больше, чем в случае gate-arrays.

В ASIC со стандартными ячейками структура библиотечных ячеек определяется специальными требованиями. Поэтому площадь кристалла меньше, чем у gate-arrays. В качестве альтернативного варианта может быть проведена оптимизация быстродействия.

14.12. ASIC на основе полностью заказного проектирования

ASIC на основе полностью заказного проектирования не отличается от обычных стандартных ИС. При проектировании разработчику предоставляется полная свобода. Изготовитель предлагает только лишь инструменты проектирования, которые адаптированы к характеристикам процесса изготовления.

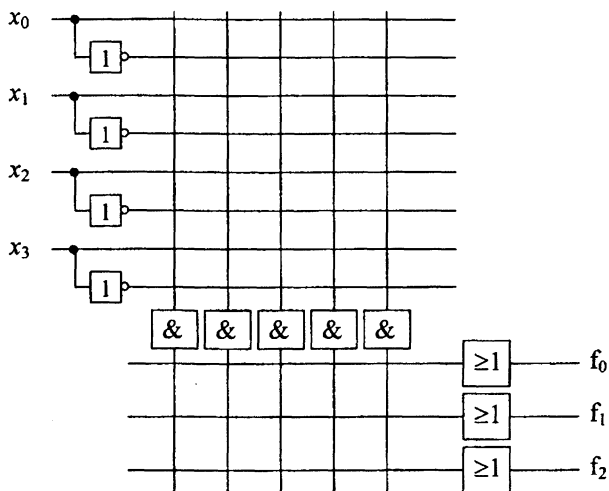
14.13. Упражнения

Задача 14.1. Пусть с помощью одной программируемой логической матрицы (PLA) должны быть реализованы три булевых функции f_0 , f_1 , f_2 . Отметьте в приведенной ниже схеме необходимые соединения точками и обозначьте выходы PLA.

$$f_0(a, b, c, d) = ad \vee \neg a \neg bcd$$

$$f_1(a, b, c, d) = \neg ab \vee \neg cd \neg abcd \vee a \neg bcd$$

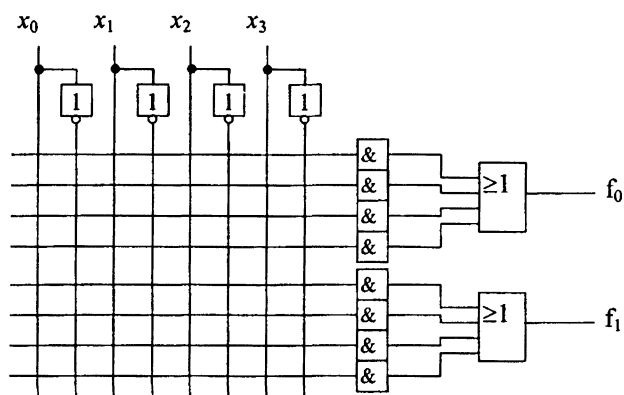
$$f_2(a, b, c, d) = \neg((a \vee b \vee \neg c)(\neg c \vee d)(\neg a \vee c \vee \neg d))$$



Задача 14.2. Пусть с помощью одной PLA должны быть реализованы 2 булевых функции f_0 и f_1 . Обозначьте на схеме необходимые соединения точками и обозначьте выходы.

$$f_0(a, b, c, d) = \neg a \neg b \neg c \neg d \vee ab \neg c \neg d \vee a \neg b \neg c \neg d \vee \neg ab \neg cd \vee \neg a \neg bcd \vee \neg abcd$$

$$f_1(a, b, c, d) = \neg a \neg b \neg c \neg d \vee ab \neg cd \vee \neg a \neg bcd \vee \neg abc \neg d$$



ГЛАВА 15

ПРИНЦИПЫ ПОСТРОЕНИЯ МИКРОПРОЦЕССОРОВ

15.1. Кооперирующиеся управляющие схемы

Цифровые системы с многими внутренними состояниями, определяемыми большим числом входов, могут быть разработаны с помощью описанных выше методов с большим трудом. Трудность заключается в чрезвычайно большом объеме требуемой таблицы последовательностей состояний. Поэтому с помощью обсуждавшихся выше методов проектирования могут быть построены только простые управляющие схемы.

Проблема разработки сложных систем может быть решена путем использования кооперирующихся управляющих схем. При таком подходе сложная схема делится на операционный и управляющий блоки, которые могут быть разработаны отдельно. Подобная структура показана на рис.15.1. Операционному блоку совместно с управляющей шиной S_i может быть придана конфигурация, соответствующая конкретной задаче. Так, например, рассмотренное в главе 12 арифметико-логическое устройство можно считать простым операционным блоком. По шине состояний Z_i информация о полученных результатах передается из операционного блока в управляющий блок.

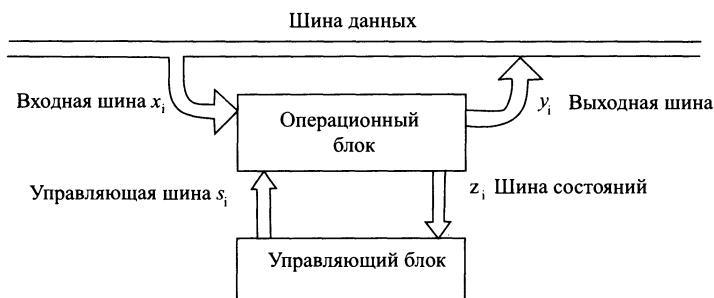


Рис. 15.1. Структура кооперирующейся управляющей системы.

С помощью операционного блока можно, как правило, обрабатывать большое число задач, поскольку операционный блок выпол-

няется максимально гибким. Универсальный операционный блок содержится, к примеру, в компьютере Неймана. Операционные блоки содержат, как правило, арифметико-логические устройства (ALU) и регистры для запоминания переменных. Все проводимые операции должны выполняться с помощью имеющихся в наличии аппаратных средств. Управляющий блок осуществляет координацию операций, проводимых в операционном блоке. Различают управляющие блоки с фиксированной структурой и с микропрограммным управлением. Первые имеют большое быстродействие, вторые являются более гибкими.

15.2. Компьютер фон-Неймана

Концепция компьютера фон-Неймана, на которой основывается работа большинства применяемых сегодня компьютеров, представляет собой развитие изображенной выше кооперирующейся управляющей схемы. В компьютере фон-Неймана используется описанное разделение схемы на управляющий и операционные блоки. Более того, в компьютере фон-Неймана управляющий блок управляется программой, которая определяет последовательность операций. С помощью этого достигается еще большая гибкость, так как, выбрав другую программу, можно легко перейти к обработке другой задачи. Программа вместе с требуемыми операционному блоку данными хранится в ЗУ. Программу, которая исполняется микропроцессором, называют машинной программой. На практике машинная программа формируется путем перевода с языка более высокого уровня (например, с языка PASCAL).

Принцип работы компьютера фон-Неймана представлен на рис. 15.2. Соединение групп блоков осуществляется с помощью трех шин. По шине данных передаются данные и команды, адресная шина передает адреса, по которым данные и команды можно найти в ЗУ. По управляющей шине направляются сигналы, которые, например, переключают ЗУ от чтения (RD) на запись (WR) и обратно. Понятия «чтение» и «запись» каждый раз могут интерпретироваться с точки зрения работы микропроцессора.

К блокам компьютера фон-Неймана относится:

Операционный блок

Операционный блок содержит АЛУ и блок регистров для хранения промежуточных результатов, как это уже описывалось выше. Как правило, его делают насколько возможно более универсальным.

Управляющий блок

Управляющий блок служит для управления компьютером.

Он выполняет машинные программы.

Для этого управляющий блок извлекает команды машинной программы из ЗУ и закладывает их в регистр команд. Он переводит команды в управляющие алгоритмы и управляет циклами программ.

ЗУ

В компьютере фон-Неймана в ЗУ хранятся как данные, так и программы. ЗУ собирается из блоков, выполняемых по различной технологии. Данные хранятся большей частью в RAM, так как они должны постоянно меняться. Управляющая программа или ее части запоминаются в ROM. Кроме того, часть данных и программы могут быть выведены на жесткий диск.

Ввод и вывод

Сюда относятся такие периферийные устройства, как принтеры, мониторы, сети данных и внешние дисководы. Собственно микропроцессор содержит, как правило, управляющий и операционный блоки. Совместно с ЗУ и блоками ввода и вывода микропроцессор образует компьютер или ЭВМ. У так называемых микроконтроллеров на кристалле присутствуют дополнительно ЗУ (RAM и ROM), блок ввода и вывода и, часто, также АЦП.

На основе микропроцессора создан универсальный схемный блок, который с помощью программного обеспечения можно использовать для решения многих задач. Поэтому микропроцессор находит широкое применение. Можно без преувеличения сказать, что микропроцессор является изобретением, которое больше всего повлияло на нашу рабочую повседневность.

15.3. Операционные блоки

15.3.1. Архитектура операционных блоков

Архитектура операционных блоков может быть приспособлена к различным специальным применениям. Компьютер фон-Неймана подходит для решения почти всех задач арифметико-логического типа. Наряду с этим имеются операционные блоки для решения специальных задач. Например, имеются процессоры, оптимизированные для обработки сигналов, т. н. «цифровые сигнальные процессоры» (DSP, digital signal processor). Они подходят, например, для

того, чтобы эффективно проводить преобразование Фурье (FFT, fast Faurier transformation).

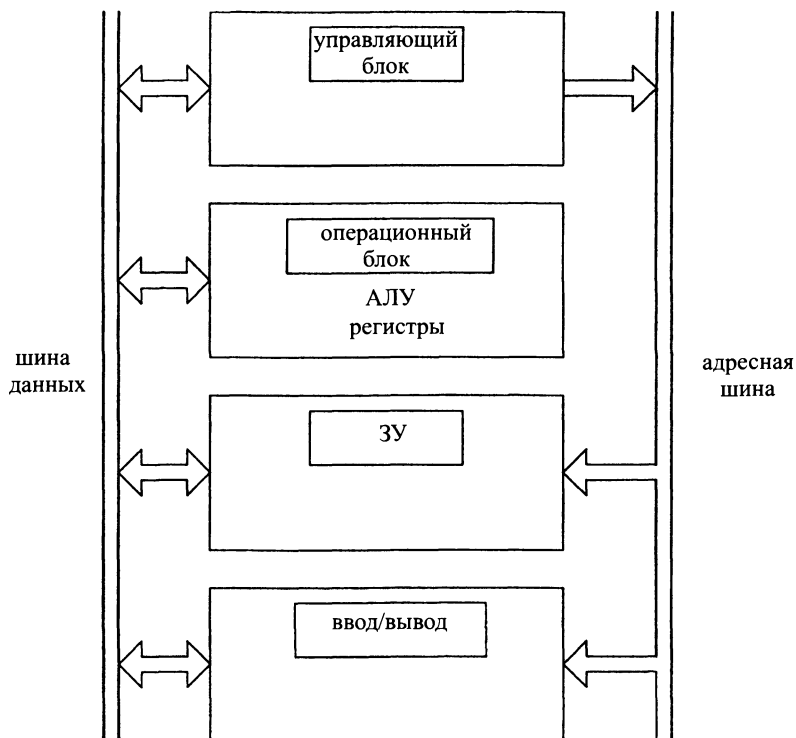


Рис. 15.2. Принципиальная структура компьютера фон-Неймана. Управляющая шина, соединяющая все блоки между собой, не показана.

Типичная структура универсального регистерно-арифметического устройства представлена на рис. 15.3. Оно состоит из двух регистров А и В. Регистр А часто обозначают как «накапливающий регистр» (accumulator). Он принимает результат операции.

С помощью представленного устройства могут проводиться логические операции сложения и вычитания. Для умножения и деления может быть применен сдвиговый регистр со сдвигом вправо и влево. Во флаг-регистре содержится информация о результатах операций. Так определенный бит во флаг-регистре является индикацией того, что результат равен 0, другой бит показывает, состоялся ли перенос (carry). Содержание флаг-регистра может быть для осуществления разветвления программы.

Другими составными частями операционных блоков являются интерфейсы для ввода и вывода данных и блоки регистров для хранения промежуточных результатов.

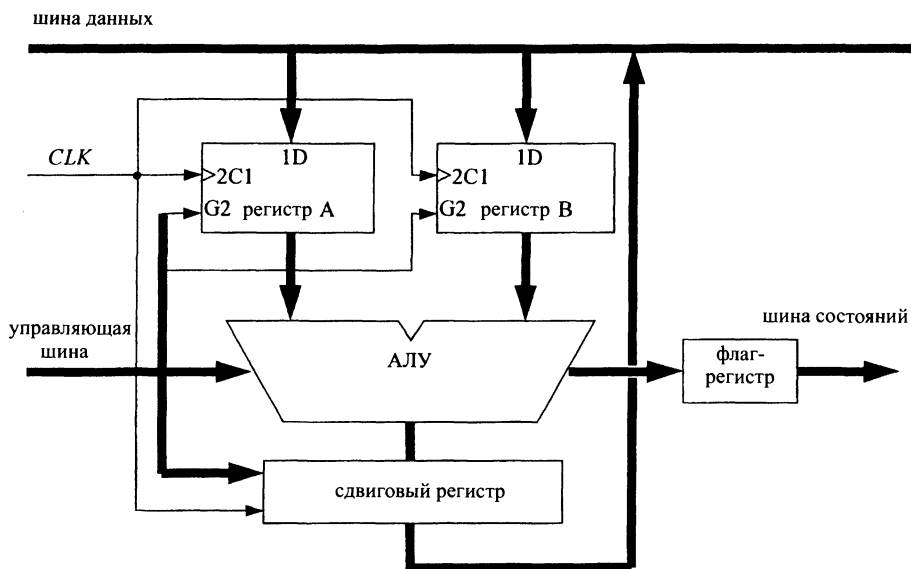


Рис. 15.3. Типичное ориентированное по шинам регистрно-арифметическое устройство.

Константы, которые необходимы операционному блоку для выполнения арифметических операций, могут быть установлены с помощью аппаратных средств либо заданы частью управляющего вектора на шину данных.

15.3.2. Система шин

Работа операционных блоков основывается на применении систем шин. В качестве альтернативного варианта применяются мультиплексоры, которые управляют потоком сигналов.

Технологическими предпосылками применения шин являются:

- применение выходов типа tri-state
- Высокий коэффициент разветвления драйверов шин, малая нагрузка, соответствующая входам вентиляей.
- Управление шинами происходит в соответствии тактовыми сигналами.

Важно, чтобы фазовый сдвиг тактового сигнала оставался в заданных границах, поскольку к шине при наличии разрешающего сигнала «Enable» в любой момент времени может быть

подключен только один выход. Большинство блоков драйверов шин рассчитывается так, чтобы столкновения не имели места. Пусть с помощью имеющих одинаковую длину символов Enable («разрешающий») и disable («запрещающий») осуществляется управление двумя выходами, работающими на одну ширину. В этом случае результирующее время, в течение которого один из выходов остается низкоомным, должно полностью лежать внутри интервала времени, в течение которого другой выход является высокоомным.

- В системах шин все схемные блоки имеют, как правило, одинаковую ширину слов.

Преимущества шинных систем заключаются в том, что им требуется меньшее количество проводников, и в том, что они являются более гибкими, но зато они имеют меньшее быстродействие, чем обычные проводные схемы.

15.4. Управляющие блоки

Различают управляющие блоки с фиксированной структурой и управляющие блоки с микропрограммным управлением. Первые имеют большее быстродействие. Их предпочитают вводить, когда требуется реализация малого числа команд. Это справедливо для компьютеров типа RISC (reduced instruction set computers, «компьютеры с уменьшенным набором команд»).

Управляющие блоки с микропрограммным управлением являются очень гибкими, их можно легко приспособить для различных случаев применения. С другой стороны они являются более медленными, поскольку требуют времени для вывода команды из ЗУ.

Объясним функцию типичного управляющего блока с помощью рис. 15.4. Для каждой команды машинной программы в ROM стартовых адресов содержится стартовый адрес, по которому в ROM микропрограмм можно найти первую микропрограммную команду из последовательности команд, соответствующих указанной команде, заданной в машинном коде. Часть машинной команды, которая определяет выбор регистра операционного блока, прямо передается в операционный блок. В рассматриваемом случае могут быть обработаны 256 машинных команд.

Между ROM стартовых адресов и ROM микропрограмм включен мультиплексор, который позволяет получить три разных адреса для следующей микрокоманды:

- первая микропрограммная команда задается с помощью ROM стартовых адресов;
- с помощью инкрементной переключательной схемы адрес может быть повышен на 1;
- при переходах часть микрокоманды может быть использована как адрес следующей команды. Как правило, возможны также условные переходы. Для этого опрашивается флаг-регистр, в котором находится полученная из операционного блока информация о результате счетной операции.

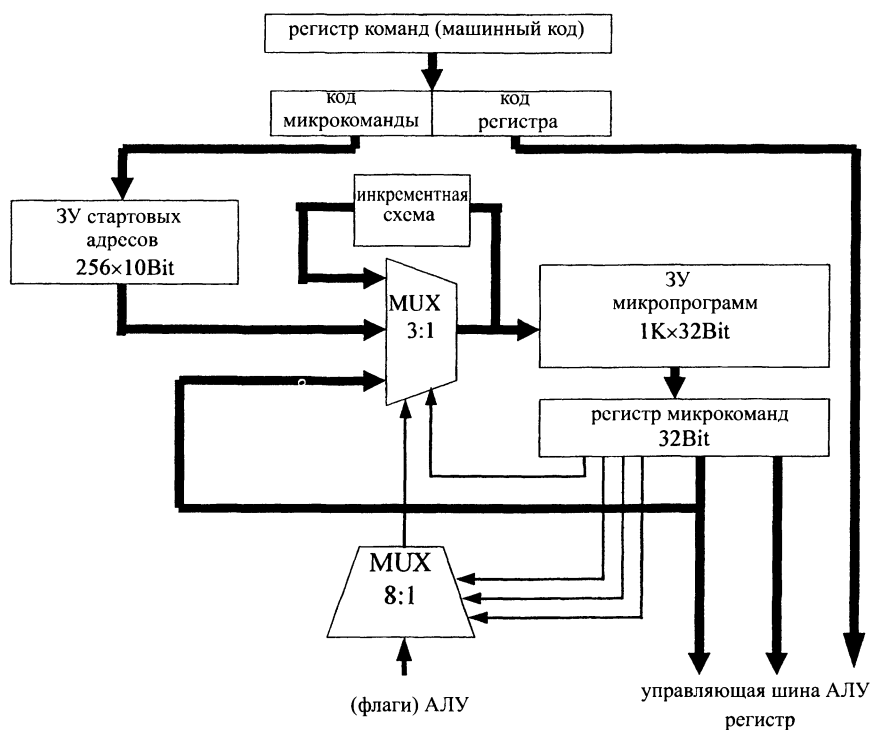


Рис. 15.4. Типичный управляющий блок.

На основе микропрограммного командного слова формируются управляющие команды для АЛУ. Для того, чтобы это можно было провести с малыми затратами и без дальнейшего декодирования, по большей части для определенной управляющей линии АЛУ используется один бит микропрограммной команды.

15.5. Микропрограммирование

В случае микропрограммного управления кооперирующихся управляющих систем должен быть согласован определенный формат для команд. Если командный формат известен, то может быть проведен расчет размеров ЗУ для микропрограмм. Отдельные команды для микропрограмм не обязательно должны иметь одинаковую длину. Должны быть учтены следующие точки зрения:

- поскольку затраты для декодирования должны быть малыми, часто с помощью какого-либо бита реализуют определенную функцию, например, с помощью определенного бита переключают определенный регистр на «считывание»;
- как правило, управляющее слово делится на различные зоны. Так, например, одна из зон может быть использована для управления АЛУ, а другие задают адрес для регистров и выходов (интерфейс);
- часть управляющей шины служит как адрес перехода (jump) при переходах. Часто каждая микрокоманда имеет поле, в котором стоит адрес следующей микрокоманды.

Теперь, руководствуясь показанным на рис. 15.5 операционным блоком, разработаем короткую микрокоманду. Особенностью этого операционного блока является возможность с помощью переменной управления I_m подать часть управляющего слова, а именно константу $A = (A_3, A_2, A_1, A_0)$ на шину данных.

Формат микрокоманды длиной 13 бит, с помощью которой можно производить переход регистра, логические и арифметические операции, может выглядеть для операционного блока на рис. 15.5 так, как показано в табл. 15.1.

Таблица 15.1. Формат микрокоманды для операционного блока, показанного на рис. 15.5.

АЛУ					Считывание		Запись		Адрес/константа			
S_3	S_2	S_1	S_0	M	A	B	A_L	I_M	A_3	A_2	A_1	A_0

- первые пять бит представляют собой управляющее для АЛУ. Так, для проведения сложения, могут, например, стоять $S = (S_3, S_2, S_1, S_0) = (1101)$ и $M = 0$.

- в поле «считывания» означают $A = 1$ или $B = 1$, что в накапливающий регистр (accumulator) A либо в регистр B переводятся данные с шины данных.
- В поле «запись» при $A_L = 1$ результат счета ALU передается на шину, при $I_m = 1$ постоянная $A = (A_3, A_2, A_1, A_0)$ последним словом подается на шину.
- Последнее поле $A = (A_3, A_2, A_1, A_0)$ содержит константы или адреса величин, запомненных во внешних ЗУ, или адрес переходов микропрограммы.

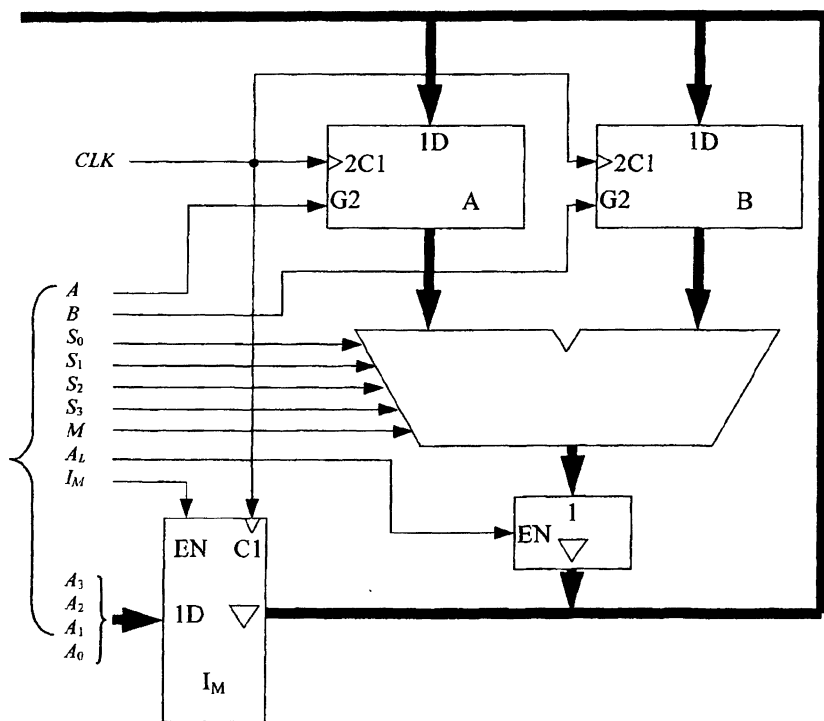


Рис. 15.5. Операционный блок для примера микропрограммы, показанной в табл. 15.2.

На основе этих исходных данных можно следующим образом запрограммировать сложение констант 1000_B и 0011_B :

Для $d = 0$ получаем три строчки микропрограммы:

```
0000010011000
0000001010011
1101010100000
```


В первой строке вследствие $I_m = 1$ константа 1000_B подается на шину данных и вследствие $A = 1$ записывается в накопительном регистре. Во второй строке константа 0011_B загружается в регистр В. В третьей строке проводится сложение ($S = (S_3, S_2, S_1, S_0) = (1101)$ и $M = 0$) и результат запоминается в накопительном регистре ($A = 1$). Кроме того, выход АЛУ подключается к шине данных ($A_L = 1$).

Таблица 15.2. Микропрограмма для указанного выше примера.

АЛУ					Считывание		Запись		Адрес/константа			
S_3	S_2	S_1	S_0	M	A	B	A_L	I_M	A_3	A_2	A_1	A_0
d	d	d	d	d	1	0	0	1	1	0	0	0
d	d	d	d	d	0	1	0	1	0	0	1	1
1	1	0	1	0	1	0	1	0	d	d	d	D

Составляющий микропрограмму программист должен иметь при этом точную информацию об аппаратных средствах. Так, в приведенном примере при последней команде результат сложения содержания регистров А и В вновь записывается в А. Это может происходить только при одинаковых тактовых сигналах, когда либо регистр А управляется двумя фронтами, либо время запаздывания линий в АЛУ достаточно велико. При этом тактовый сдвиг фазового сигнала должен контролироваться достаточно точно.

15.6. Упражнения

Задача 15.1. Каковы достоинства и недостатки совместного хранения данных и команд в том же самом ЗУ в компьютере фон-Неймана?

Задача 15.2. Что произойдет, если в операционном блоке, показанном на рис. 15.5, переменные $A_L = 1$ и $I_n = 1$ будут сдвинуты? Что следует из этого для программирования блоков?

ГЛАВА 16

СТРУКТУРА, СИСТЕМА КОМАНД И РАБОТА МИКРОПРОЦЕССОРА

16.1. Структура микропроцессора

Структура микропроцессора описывается на примере микропроцессора 8085А компании Intel [19, 22]. Микропроцессор 8085А представляет собой 8-битный процессор, что означает, что шина данных имеет ширину 8 бит. Адреса имеют ширину 16 бит, что соответствует адресуемому адресному пространству 64 кБайт. Один байт состоит из 8 бит.

Микропроцессор 8085А требует напряжения питания 5 В. Он имеет интегрированный кварцевый генератор с внешним кварцем. Тактовая частота лежит между 3 и 6 МГц. Кварц работает на удвоенной частоте.

К отдельным компонентам, показанным на рис. 16.1 относятся:

Накапливающий регистр и временный (буферный) регистр

Эти оба регистра соответствуют регистрам А и В операционного блока, который обсуждался в главе 15. Оба имеют ширину 8 бит. Накапливающий регистр (accumulator) ниже обозначается сокращенно через А.

Флаг-регистр

Флаг-регистр выдает информацию о результатах операции АЛУ. Он имеет ширину 8 бит и может показать следующие 6 флагов:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S	Z		AC		P	V	CY

- S Флаг знака (sign-flag), эквивалентен биту 7 результата операции.
- Z Нулевой флаг (zero-flag) устанавливается тогда, когда все биты равны 0.

- АС Флаг вспомогательного переноса (*auxiliary-carry-flag*), является индикацией переноса из бита 3. Он обслуживает арифметические операции в двоично-десятичном коде.
- P Флаг четности (*parity-flag*) показывает, что число единиц в результате является четным.
- СУ Флаг переноса (*carry-flag*) устанавливается при арифметических операциях. Например, он устанавливается при сложении, когда результат больше 8 бит. Если результат меньше, СУ устанавливается на 0.
- V Флаг переполнения (*overflow-flag*) является индикацией переполнения при арифметической операции в соответствии с разделом 2.3.6.

Блок регистров

Он состоит из трех пар регистров (B,C), (D,E), (H,L), которые содержат по 8 бит. К ним можно обращаться как по одиночке (например, за данными), либо как к паре (например, за адресами).

Указатель вершины стека

Указатель вершины стека запоминает адрес последнего байта, который был записан в стеке. Когда в стек записываются новые данные, указатель вершины стека понижает свое значение, при каждой операции в стек записываются по 2 бита. Существуют команды ассемблерного типа, при которых содержание регистровой пары и слово состояния процессора (флаг-регистр и накапливающий регистр) могут быть записаны в стек.

Регистр команд

В командный регистр приходит машинная команда после того, как она выведена из ЗУ. В заключение она декодируется.

Счетчик команд

Счетчик команд содержит адрес следующей подлежащей исполнению команды; после того, как операционный код команды доставлен из ЗУ, счетчик повышает свое значение и указывает на следующую команду.

Декодер команд

В декодере команд из машинной команды выводятся управляющие команды как для отдельных регистров и АЛУ, так и для периферийного управления.

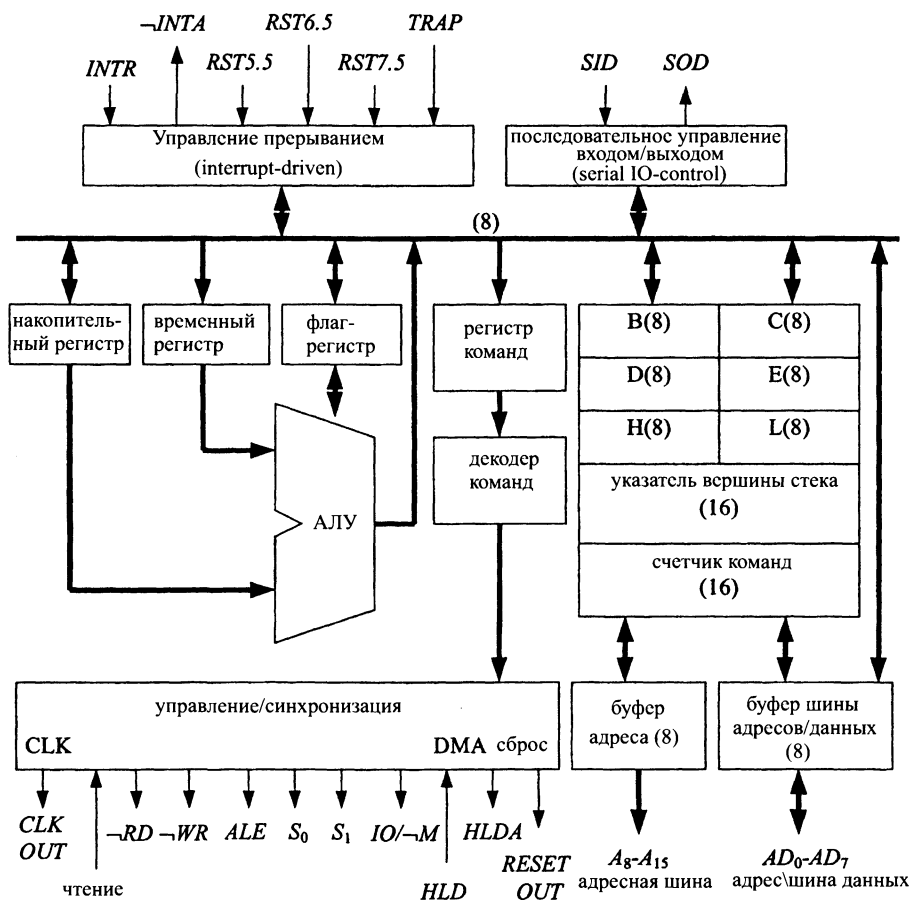


Рис. 16.1. Структура 8085А (ширина регистров показана в скобках). Внутренняя шина имеет ширину 8 бит.

Адресный буфер

В адресном буфере запоминается имеющая более высокие значения часть адреса до тех пор, пока она не считана с соответствующей схемы (ЗУ, схема I/O).

Информационно-адресный буфер

Информационно-адресный буфер запоминает данные, которые должны быть поданы на адресную шину до тех пор, пока соответствующая схема (ЗУ, схема I/O) не прочтает их. Кроме того, через этот буфер выдается младший байт. Это позволяет сэкономить 8 выводов схемы, а внутри требуемая ширина шины составляет только 8 бит.

Управление с помощью прерывания

У микропроцессора 8085А имеется 5 различных возможностей восприятия прерывания. При этом возможно воздействие на вычислительный процесс снаружи, например, чтобы сообщать о вводе через клавиатуру.

Последовательное I/O управление

(управление входом/выходом)

У микропроцессора 8085А имеется простой последовательный интерфейс, осуществляющий обмен данных. Выход SOD означает Serial Output Data («последовательные выходные данные»), вход называется «последовательные входные данные» (Serial Input Data).

16.2. Выводы 8085А

Микропроцессор 8085А поставляется в корпусе с 40 выводами (рис. 16.2):

X_1, X_2	Контакты для кварцевого резонатора.
CLK(OUT)	Задаёт тактовые сигналы 8085А для периферийных блоков.
RESET IN	С его помощью процессор можно установить в исходное состояние. Если вход находится в состоянии L (low), то процессор переходит в основное состояние. При подходе нарастающего фронта программа в позиции 0000 стартует.
RESET OUT	Поставляет сигнал возврата для периферии.
TRAP	} входы для сигналов прерывания
RST7.5	
RST6.5	
RST5.5	
INTR	
$\neg$ INTA	interrupt acknowledge (подтверждение прерывания), запрашивает при прерывании сигнал от адресной шины.
SOD/SID	Последовательные вход и выход данных.
AD_0-AD_7	Входы и выходы типа tri-state. Они являются выходами для младшей части адреса ЗУ и входами или выходами для шины данных.
A_8-A_{15}	Выходы типа tri-state для старшей части адреса ЗУ.
ALE	с помощью сигнала ALE (address latch enable, разблокирование адресного фиксатора) процессор показывает, что шина AD_0-AD_7 передаёт адреса (а не данные).

Адреса при появлении падающего фронта ALE становятся стабильными и могут на основе управления по отрицательному фронту подвергнуться промежуточному хранению во внешнем регистровом узле с фиксацией состояния (latch).

HOLD	HOLD («фиксация состояния») представляет собой один из выходов. После поступления сигнала HOLD процессор делает выходы данных и выходы на адресную шину высокоомными, с тем чтобы внешний процессор (DMA-controller, direct memory access controller, «контроллер прямого доступа в ЗУ») мог бы обмениваться данными с ЗУ.
HLDA	Процессор показывает, что он получил сигнал HOLD.
READY	ЗУ или периферийные схемные узлы сообщают на этот вход, что они готовы (ready) к приему данных. Этот порядок действий применяется только в случае медленных ЗУ. При использовании более быстродействующих ЗУ READY подключается через резистор 3 кОм к напряжению питания V_{cc} .
S_1/S_0	Выходы, которые показывают статус процессора (см. таблицу ниже).
$\neg RD, \neg WR$	$\neg RD$ индицирует с помощью сигнала L (low), что процессор освободил шину данных и ожидает данные от периферийных устройств либо от ЗУ. Данные принимаются процессором при появлении нарастающего фронта $\neg RD$ -сигнала. Появление L (low) на $\neg WR$ является индикацией того, что на шину данных поданы значащие сигналы.
$IO/\neg M$	ЗУ/вход-выход: индицирует обращение к периферийному устройству, а не к ЗУ.

У микропроцессора 8085A напрямую выведены только те 8 линий, которые соответствуют старшим адресам. Младшие 8 адресных линий мультиплексированы с шиной данных. Поэтому младшие адреса должны подвергаться промежуточному хранению в регистре. Микропроцессор 8085A выдает специальный сигнал (сигнал ALE) для управления регистрами.

В распоряжении пользователя имеются специальные периферийные блоки и блоки ЗУ, которые осуществляют соответствующее мультиплексирование. К этим блокам должен быть подведен сигнал ALE.

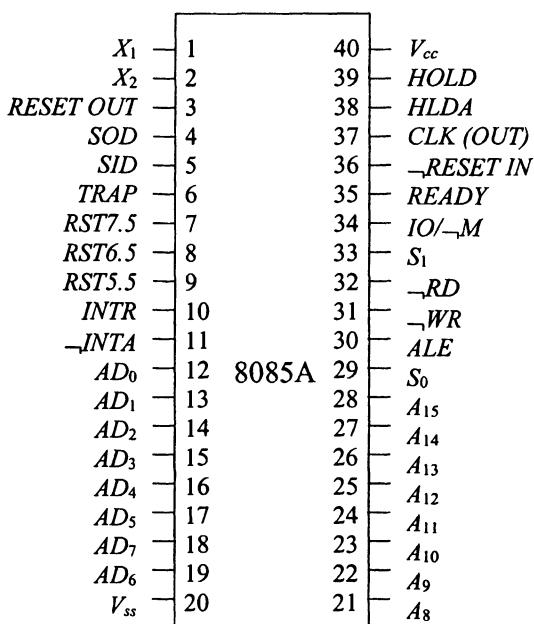


Рис. 16.2. Выводы микропроцессора 8085А.

Таблица 16.1. Значение сигналов S_0 и S_1 .

S_1	S_0	рабочее состояние
0	0	ожидание на основе команды HLT
0	1	запись
1	0	считывание
1	1	получение функционального кода (M1-цикл)

К рабочим состояниям 8085А относятся:

1. Запуск процессора с помощью сигнала RESET.
2. Обработка программы: продолженное получение и выполнение команд программы.
3. Прерывание программы при получении сигнала Interrupt. В заключение выполнения подпрограммы обработки прерывания (interrupt-routine).
4. Прерывание программы сигналом HOLD или командой HLT. Линии шины переключаются в высокоомное состояние, чтобы DMA-controller (direct memory access controller) мог обмени-

ваться данными с ЗУ. Непосредственно после этого программа продолжается.

5. Временная остановка программы сигналом READY до того момента, пока ЗУ не примет либо не выдаст данные.

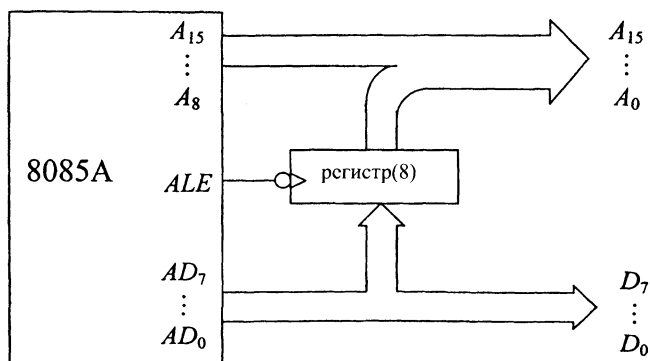


Рис. 16.3. Подключение шины данных и адресной шины.

16.3. Блоки ЗУ и периферийные блоки

К микропроцессору могут быть подключены блоки ЗУ и периферийные блоки. ЗУ предназначаются для сохранения данных и программ и имеют, как правило, емкости в несколько кБайт. Они рассматривались в главе 13. периферийные же блоки имеют емкость ЗУ только в несколько бит. Они используются для связи с внешними блоками.

Различают две возможности подключения блока:

1. Подключение как ЗУ (memory mapped, «с управлением памятью»). Блок выбирается с помощью 16 адресных линий. При этом сигнал $IO/\neg M = 0$. Подобная операция с ЗУ проводится всегда аналогично. Что же касается периферийных блоков, как, например, в случае блока для пользовательского интерфейса, можно действовать подобным образом. Когда периферийные блоки подключаются также, как и блоки ЗУ, нет принципиального различия в том, записывается ли элемент данных в блок ЗУ или в периферийный блок. Поэтому применяются те же команды, которые используются и для записи данных в ЗУ (MOV, MVI, LDA, STA, ... см. ниже).
2. Подключение как периферийный блок («с управлением входом/выходом», IO-mapped). В этом случае к периферийным

блокам обращаются с использованием нижних (или верхних) 8 битов адресной шины и сигнала $IO/\neg M = 1$. Для этого применимы обе команды, IN и OUT. Длина адреса периферийного блока равна каждый раз 8 бит. В микропроцессоре 8085A этот 8-битовый адрес подается при подключении адресного фиксатора (address latch) на младший и старший адресные байты.

Преимущества, получаемые при подключении периферийного устройства в качестве ЗУ (memory-mapping, управление памятью):

- В нашем распоряжении имеется большое число команд для осуществления передачи данных.

Недостатки подключения периферийных блоков в качестве ЗУ (memory-mapping):

- Количество адресуемых ячеек ЗУ уменьшается. Для того, чтобы различать периферийные блоки и блоки ЗУ, как правило, используется A_{15} . могут быть подключены ЗУ с общей емкостью только 32 К.
- Команды становятся длиннее, так как они должны содержать 16-битовые адреса.

16.4. Периферийные блоки

Периферийные блоки представляют собой программируемые интерфейсы компьютера. Различают параллельные и последовательные интерфейсы. Типичным для большинства интерфейсов является то, что они двунаправлены. Они имеют, как правило, несколько командных регистров, в которых запоминаются особенности процедуры передачи. Например, в одном из регистров хранятся направление и вид сигналов, используемые при обмене данными. Эти командные регистры загружаются процессором. Далее данные должны быть запомнены в промежуточном ЗУ. Это необходимо, поскольку процессор и периферия в редких случаях работают синхронно.

На рис. 16.4 представлен двунаправленный интерфейс, который, конечно, сильно упрощен. Изображен только один канал i интерфейса с шириной 8 бит. Интерфейс подключен к внешней шине данных и управляющей шине микропроцессора. Поэтому входу данных D_i соответствуют вход и выход типа tri-state. Выход также выполнен с помощью вентиля типа tri-state, чтобы он мог работать на двунаправленную шину.

При выдаче данных интерфейс прежде всего должен быть установлен на выдачу. Эта информация запоминается в одном из непоказанных здесь командных регистров. Командный регистр загружается через шину данных. Установив $\neg WR = 0$, можно загрузить один бит в выходной триггер. Поскольку правый tri-state-буфер включен на основе сигнала enable («разрешить»), результат виден на внешней шине.

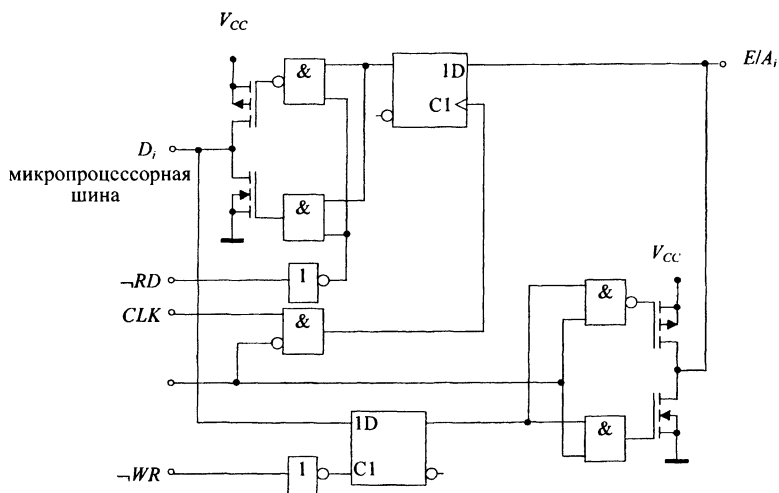


Рис. 16.4. Принцип построения двунаправленного интерфейса. Изображение канала i параллельного интерфейса.

При введении данных устанавливается $\neg RD = 0$. Затем с каждым тактом, имеющаяся на внешней шине, информация загружается в верхний входной триггер. При $\neg RD = 0$ бит может быть выдан на шину микропроцессора.

Взаимодействие с внешней шиной представлено здесь упрощенно. Как правило, имеется протокол, на основе которого можно выяснить, имеются ли новые данные на шине, когда интерфейс установлен на прием, либо выяснить, считываются ли данные, когда интерфейс включен на передачу. Если данные принимаются, можно просигнализировать о приеме в процессор с помощью, например, сигнала прерывания.

16.5. Параллельный интерфейс микропроцессора 8255

В качестве примера блока интерфейса рассмотрим параллельный интерфейс микропроцессора 8255. В него входят три параллельных

интерфейса с портами А, В и С, каждый шириной 8 бит. С помощью двух адресных линий A_1 и A_0 выбирается один из четырех регистров интерфейса. Такими регистрами могут быть как регистры данных портов А, В и С, так и командный регистр. Командный регистр содержит информацию как о том, действует ли интерфейс как вход или как выход, так и об особенностях способов передачи данных.

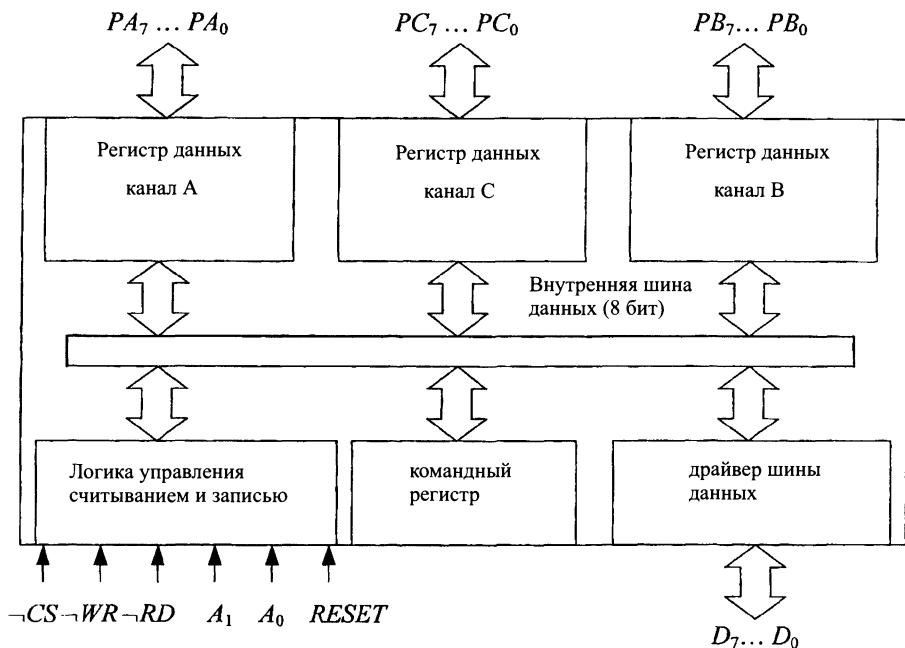


Рис. 16.5. Параллельный интерфейс микропроцессора 8255.

Линии данных от D_0 до D_7 подключены к шине данных, они являются высокоомными до тех пор, пока блок не выбран с помощью сигнала $\neg CS$. К выводам от A_7 до A_0 , от C_7 до C_0 и от B_7 до B_0 можно подключить переключатели, светодиоды, датчики, принтеры др. устройства.

Регистры выбираются линиями данных A_0 и A_1 в соответствии с табл. 16.2.

16.6. Процесс выполнения команд

В нормальном рабочем состоянии процессор реализует непрерывно повторяющуюся последовательность шагов. Постоянно из ЗУ выда-

ются программы, которые затем обрабатываются. В командном ЗУ команды приводятся в порядок в соответствии с последовательностью программы. Счетчик команд (PC, program counter) в процессоре содержит адрес команды, которая будет выполняться следующей.

Таблица 16.2.

A_1	A_0	Регистр
0	0	регистр данных, порт А
0	1	регистр данных, порт В
1	0	регистр данных, порт С
1	1	может быть записано в командный регистр

Команда имеет фиксированный формат, она снабжена специальным кодом и, когда необходимо, адресом. В качестве примера используем ассемблерную программу STA OF12H. Она служит для запоминания содержания накапливающего регистра (accumulator) в ячейке ЗУ OF12H. STA представляет собой так называемое мнемоническое изображение; можно легко заметить, что оно выведено из английского описательного обозначения «store accumulator». Компилирующая ассемблерная программа переводит эту команду на машинный язык, в двоичное представление, которое может быть интерпретировано процессором. В машинном языке заменены друг на друга младшие и старшие байты адреса.

Таблица 16.3. Противопоставление ассемблерного, шестнадцатеричного и двоичного представлений команды STA DF12H. Команда состоит из 3 байтов.

Ассемблерный способ записи	STA	OF12H
Шестнадцатеричный способ записи	32	OF 12
Двоичный способ записи	0011 0010	0001 0010 0000 1111

Для выполнения команды содержание счетчика команд передается на адресную шину и команда из ЗУ загружается в декодер команд. Эта фаза называется «выборка операционного кода» (OF, operation code fetch). В этом машинном цикле команда также декодируется. После этого становится известным, из скольких байтов состоит команда и сколько машинных циклов необходимо, чтобы ее выполнить. В предложенном примере это два дальнейших байта, задающих адрес ЗУ, в котором должно быть перезаписано содержание накопительного регистра. В конце этого цикла счетчик команд повышает свое значение на 1. Теперь он указывает на младшие байты адреса (в данном примере OF H).

В следующем цикле из ЗУ получают младший байт адреса. Этот цикл называется «считывание ЗУ» (MR, memory read). Для этого процессор выдает адрес счетчика команд на адресную шину. Счетчик программ вновь повышает свое значение на 1. В следующем цикле, также являющимся MR-циклом, считывается старший байт адреса.

В последнем цикле содержание накапливающего регистра записывается в ячейке ЗУ, адрес которой был считан перед этим. Этот цикл называется «запись в память» (MW, memory write).

16.6.1. Цикл выборки операционного кода (OF)

Цикл выборки операционного кода является единственным циклом, который имеет 4 такта. Причина этого заключается в том, что после выдачи операционного кода из ЗУ он должен быть сперва декодирован, чтобы решить, какая операция должна быть проведена следующей. Временная диаграмма OF-цикла команды STA OF12H показана на рис. 16.6.

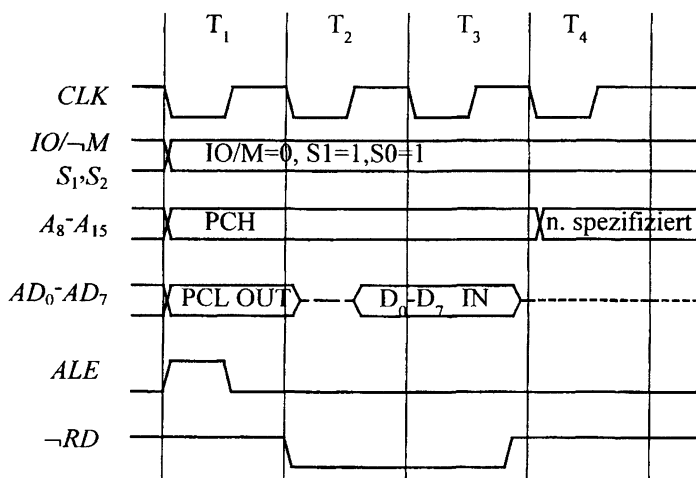


Рис. 16.6. Временная диаграмма цикла выборки операционного кода в случае команды STA OF12H.

Такт 1

В начале цикла посылаются управляющий сигнал и сигнал приоритета. В этом случае $IO/\bar{M} = 0$, $S_1 = 1$, $S_0 = 0$, $INTA = 0$. Содержание счетчика команд выдается на адресную шину. Старшая часть адреса поступает на PCH, младшая часть адреса переводится при появлении сигнала ALE из шины данных в регистр промежуточного ЗУ.

Такт 2

К началу второго такта сигнал считывания $\neg RD$ устанавливается на L (low). ЗУ отвечает на это содержанием ячейки ЗУ, адрес которой выслан в такте 1. После интервала, соответствующего времени обращения к ЗУ, операционный код находится на шине данных.

Такт 3

Данные подаются на шину данных. Сигнал считывания $\neg RD$ вновь устанавливается на H (high). Поскольку протекает OF-цикл, автоматически обрабатывается следующий такт. Значение счетчика программ возрастает на 1.

Такт 4

В течение этого такта декодируется операционный код. Решается, будут ли обрабатываться дальнейшие такты, как это имеет место при команде DCX.

16.6.2. Цикл считывания памяти (MR, memory read)

Цикл считывания памяти (MR, memory read) представлен здесь в виде двух протекающих друг за другом циклов, причем один из циклов содержит «состояние ожидания» (wait state) T_w . Такт T_w происходит тогда, когда на входе *READY* состояние *READY* = 0 показывает, что ЗУ с более высоким временем доступа еще выдается значащие данные на шину данных. В случае быстрых ЗУ, которые могут ответить в течение одного такта, сигнал *READY*, как правило, устанавливается на H. Временная диаграмма этого цикла изображена на рис. 16.7.

Такт 1

В начале цикла посылаются управляющий сигнал и сигнал приоритета. Процессор накладывает адрес счетчика команд на адресную шину. Сигнал ALE вызывает загрузку младшей части адреса в регистр промежуточного ЗУ. Программный счетчик повышает свое значение на 1.

Такт 2

К началу второго такта сигнал считывания $\neg RD$ устанавливается на L (low). ЗУ отвечает на это содержанием ячейки ЗУ, адрес которой выслан в такте 1. После интервала, соответствующего времени выборки, имеются значащие данные. Поэтому сигнал *READY* находится в состоянии H (high).

Такт 3

Данные вводятся на шину данных и сигнал считывания $\neg RD$ вновь устанавливается на H (high). Во втором из показанных MR-циклов

сигнал **READY** во втором такте еще на L (low), и процессор поэтому начинает такт ожидания T_w до тех пор, пока сигнал **READY** не переходит в состояние H (high), показывая этим, что данные являются значащими.

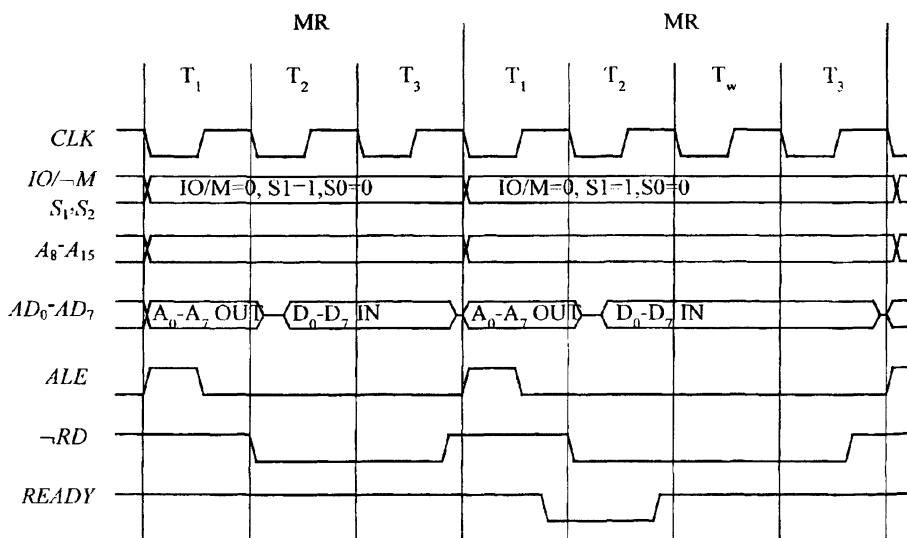


Рис. 16.7. Временная диаграмма цикла считывания ЗУ. Один цикл с другим без состояния ожидания.

16.6.3. Цикл записи в ЗУ (MW, memory write)

На рис. 16.8 представлены циклы memory-write, содержащие и не содержащие интервал, соответствующий состоянию ожидания.

Такт 1

В начале цикла посылаются сигналы и управляющий. Процессор выдает на адресную шину адрес ячейки ЗУ, в которую должна произойти запись.

Такт 2

Если данные находятся на шине данных, сигнал записи $\neg WR$ устанавливается на L (low).

Такт 3

После этого ЗУ вводит данные и отвечает значением H (high) сигнала **READY**, если он готов. Сигнал записи $\neg WR$ устанавливается на H, если сигнал **READY** имеется в наличии в последнем такте; в противном случае выставляется состояние ожидания (wait state).

Аналогичный порядок действий справедлив и цикла IO-write («запись по входу/выходу»), только при этом $IO/\neg M = 0$.

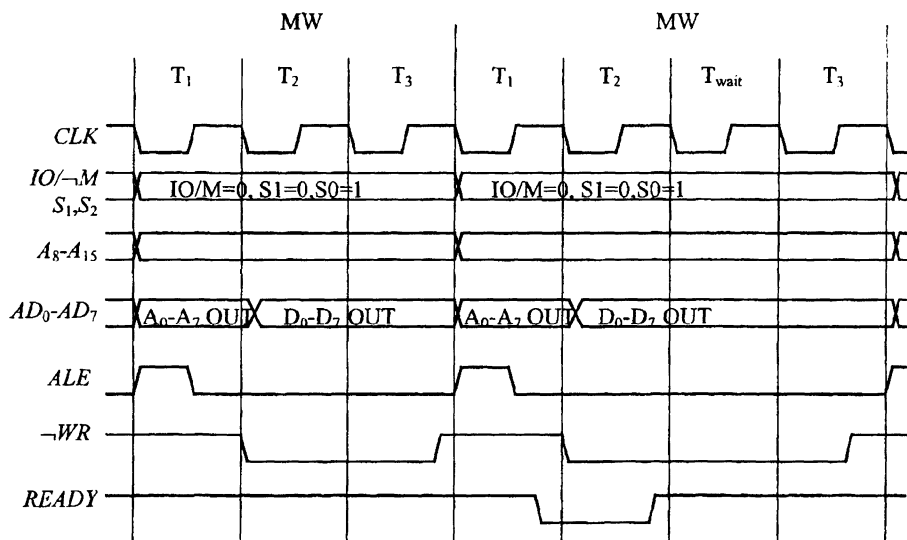


Рис. 16.8. Временная диаграмма цикла memory-write с наличием и без наличия состояния ожидания (wait state). Протекание процесса действительно и для цикла IO-write.

16.6.4. Пример выполнения команды

В качестве примера демонстрируется команда STA IB02H, 3 байта которой записываются в ЗУ по следующим адресам:

Адрес	Содержание	Комментарий
1001	32	STA
1002	02	младший байт адреса
1003	1d	старший байт адреса

Процессор перезаписывает адреса и данные всегда в следующей последовательности: младший байт — старший байт.

На рис. 16.9 показано, что в первом цикле получаем команду STA, во втором цикле младшую часть адреса (02) и в третьем цикле старшую часть адреса (1B). В каждом из трех циклов программный счетчик увеличивает свое значение на 1, от 1001 до 1003. В 4-ом цикле адрес 1B02 подается на адресную шину и содержание накапливающего регистра записывается в адресованную таким образом ячейку ЗУ.

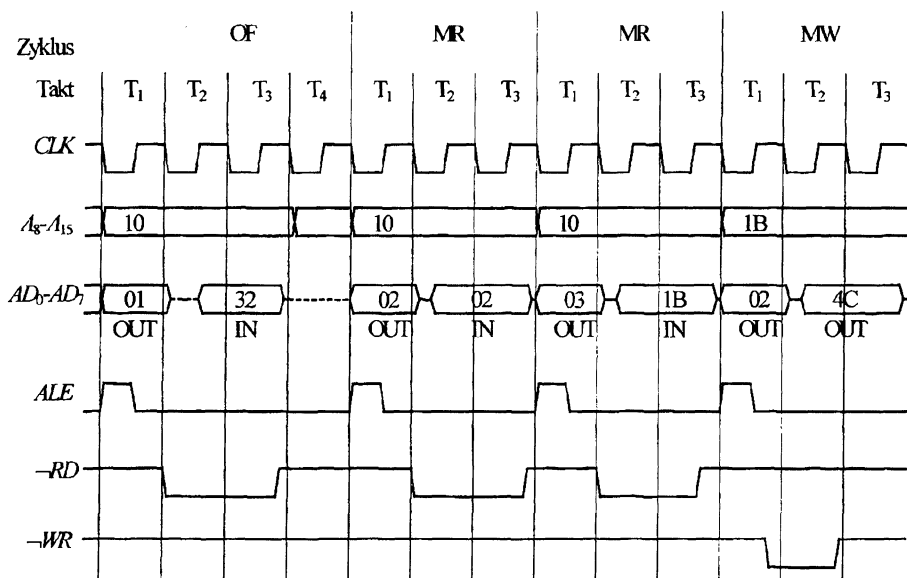


Рис. 16.9. Временная диаграмма команды STA IB02H (содержание из A: 4C).

16.7. Управление с помощью прерывания (interrupt)

Прерывание представляет собой возможность останавливать выполняющуюся программу извне. Это может потребоваться, например, тогда, когда с помощью клавиатуры введена какая-либо команда, или тогда, когда должна быть проведена аварийная остановка управляемого микропроцессором станка. Если на линию прерывания (interrupt line) подан сигнал, выполнение программы прерывается и вместо нее выполняется подпрограмма обработки прерывания (ISR, interrupt-service-routine), которая должна решить имеющиеся задачи. В заключение выполнение программы продолжается.

Сигналы interrupt являются активными High-сигналами (H) и могут формироваться на основе тактовых сигналов процессора. При выполнении каждой подобной команды проверяется, поступил ли сигнал interrupt извне. Это происходит во время предпоследнего такта команды. В любом случае команда выполняется до конца.

При любых прерываниях сначала должно быть сохранено стековое ЗУ (stau). Потом адреса программы обработки прерывания (ISR) загружаются в программный счетчик. В заключение програм-

ма обработки прерывания выполняется. В конце ISR с помощью команды RET содержание счетчика команд выносится из стекового ЗУ и программа продолжается с прежнего места.

Таблица 16.4. Свойства сигналов прерывания у 8085А.

Прерывание	Запуск	Способность	Способность	Адрес	Приоритет
<i>TRAP</i>	фронт состояние	нет	нет	0024	1
<i>RST7.5</i>	фронт	да	да	002С	2
<i>RST6.5</i>	состояние	да	да	003С	3
<i>RST5.5</i>	состояние	да	да	0034	4

Различают следующие моменты:

- блокирование прерывания: с помощью команды EI (enable interrupt, «разрешение прерывания») и команды DI (disable interrupt, «запрещение прерывания») все прерывания, вплоть до TRAP, могут быть включены либо выключены (maskable interrupt, «маскируемое прерывание»). Прерывания, которые могут быть заблокированы, блокируются с помощью INTE-триггера. На него могут воздействовать сигнал возврата в исходное состояние (reset), прерывание и команда DI (disable interrupt).
- маскирование прерывания: с помощью команды SIM (set interrupt mask, «установка маски прерывания») маскируемые прерывания могут быть селективно включены и выключены.

Когда в одно и то же время накладывается несколько прерываний, обрабатывается то, которое имеет самый низкий приоритет. После установления прерывания начинает выполняться программа обработки прерывания (ISR), которая должна быть записана программистом под заданным в табл. 16.4 адресом.

Сигнал прерывания INTR взаимодействует с внешним контроллером, который может обработать большое количество прерываний. Контроллер прерываний при появлении прерывания выдает адрес подлежащей обработке программе прерываний на адресную шину. Для управления взаимодействием используются линии INTR–INTA.

Сигнал reset обрабатывает протекающую программу (также и обработку прерывания). В заключение программа выполняется с адреса 0000.

16.8. Ассемблерное программирование

Программа ассемблерного типа является программой, которая символические обозначения команд (мнемонические обозначения) переносит в двоичный машинный код. По существу, ассемблерная программа является переводчиком. Представленную на ассемблерном языке программу называют source code («исходная программа»), выполняемую микропроцессором программу называют object code («объектная программа»). Процесс перевода называют трансляцией с ассемблерного языка. Перевод может также производиться «от руки». Для учебных целей предлагаются специальные моделирующие программы. Одна из возможных версий представлена в [42].

Язык ассемблерного типа называют также языком программирования, в котором команды представлены в мнемонической форме. Они специфичны для микропроцессоров.

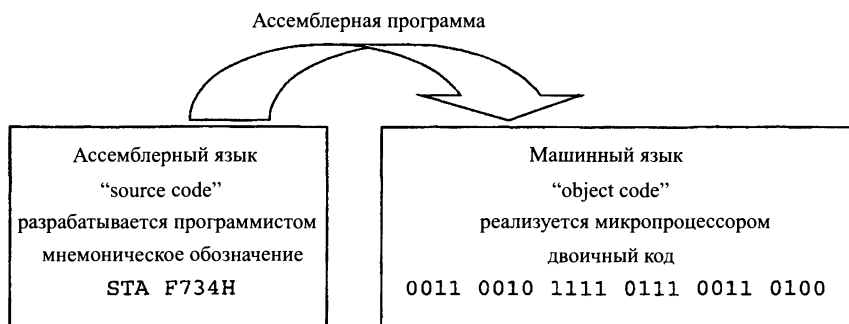


Рис. 16.10. Соотношение ассемблерного языка и машинного языка.

16.9. Набор команд

Ниже обсуждается набор команд микропроцессора 8085А. В качестве примеров подобных команд выбраны несколько экземпляров, относящихся к различным их видам. При обсуждении набора команд использованы следующие сокращения:

addr	16-битовый адрес
data	8-битовый элемент данных
data 16	16-битовое число
byte n	n-ый байт команды
port	8-битовый адрес периферийного блока
r, r1, r2	один из регистров A, B, C, D, E, H, L

gr	одна из пар регистров (старший байт, младший байт) BC (сокращенно B), DE (сокращенно D), HL (сокращенно H)
rh, rl	старший (младший) регистр одной пары
PC	16-битовый счетчик команд. PCH и PCL являются старшим и младшим байтами PC
SP	16-битовый указатель вершины стека
()	содержание регистра или ячейки ЗУ, которое стоит в скобках
(())	содержание ячейки ЗУ с адресом, который стоит в паре регистров, указанной в скобках

16.10. Команда перехода

Команды перехода обеспечивают сдвиг данных. Для этого процессору должны быть известны источник и цель сдвига, которые передаются процессору с использованием различных средств. Для микропроцессора 8085A известны следующие виды адресации «Register» («регистр»), «Register Indirect» («регистр непрямого типа»), «Direct» («прямого типа»), «Immediate» («немедленная»). Ниже описываются команды перехода, при этом учитываются применяемые виды адресации.

16.10.1. Адресация типа «Register»

Так называется тип адресации, при котором операция основывается на регистрах. Переход от одного регистра микропроцессора к другому может происходить при команде MOV. Название команды имеет в основе английское понятие move (пересылать данные). Так MOV A, B означает, что содержание регистра B загружается в накапливающий регистр A. Содержание B сохраняется. Команды MOV состоят из одного байта.

Таблица 16.5. Команды перехода с адресацией типа «Register». B = байты, Z = циклы, T = такты.

Команда	Операнд	Воздействие	B	Z	T
MOV	r1, r2	(r1) $\leftarrow$ (r2)	1	1	4
SPHL		(SP) $\leftarrow$ (H)(L)	1	1	6
XCHG		(H) $\leftrightarrow$ (D), (L) $\leftrightarrow$ (E)	1	1	4
PCHL		(PCH) $\leftarrow$ (H), (PCL) $\leftarrow$ (L)	1	1	6

SPHL означает «MOV HL to SP» («сдвинуть HL к SP»); команда действует так, чтобы содержание HL-регистра записывалось в стек.

XCHG означает «Exchange Hand L with D and E» («перестановка H и L с одной стороны и D и E с другой стороны»).

PSHL значит «Move H and L to PC» («сдвинуть H и L к PC»). Формально это команда перехода для регистров. Но на следующем шаге происходит переход к адресу, который стоит в регистровой паре HL.

16.10.2. Адресация типа «Register Indirect»

В командах с адресацией типа «Register Indirect» называется регистровая пара, в которой стоит адрес, под которым можно найти операнд. Это изображено с помощью способа записи с двойными скобками.

Таблица 16.6. Команды перехода с адресацией типа «Register Indirect». В = байты, Z = циклы, T = такты.

Команда	Операнд	Воздействие	В	Z	T
MOV	М, r	$((H)(L)) \leftarrow (r)$	1	2	7
MOV	r, М	$(r) \leftarrow ((H)(L))$	1	2	7
LDAX	rp (B, D)	$(A) \leftarrow ((rp))$	1	2	7
STAX	rp (B, D)	$((rp)) \leftarrow (A)$	1	2	7
XTHL		$(L) \leftrightarrow ((SP)), (H) \leftrightarrow ((SP)+1)$	1	5	16

Команда MOV М, А означает, что в H и L стоит адрес ячейки ЗУ, в которой записано содержание регистра r. Итак, в этом случае вместо обозначения регистра применено обозначение М (memory).

Команда также может быть применена с обратной последовательностью: MOV r, М. В соответствии с этой командой содержание ячейки ЗУ, адрес которой стоит в HL, в регистр r.

LDAX rp означает «Load accumulator indirect (indexed)» («загрузочный накапливающий регистр косвенного действия, с индексацией»). Содержание ячейки ЗУ, адрес которой стоит в rp, загружается в накапливающий регистр. Здесь rp может соответствовать одна из пар BC или DE.

STAX rp «store accumulator indirect (indexed)» («накапливающий регистр с хранением, косвенного действия, с индексацией»). rp здесь соответствует одной из регистровых пар BD или DE.

XTHL означает «exchange stack with H and L» («замена содержания стека с H и L»). Команда вызывает обмен содержания двух старших битов стека с регистровой парой HL. В регистр L приходит старший байт, в регистр H приходит следующий по старшинству байт, адрес

которого выше на 1. При этом стек адресуется косвенно с помощью указателя вершины стека.

16.10.3. Адресация типа «Immediate»

Команда содержит данные в самом байте 2. Команда **MVI C, 8 FH** загружает регистр C с константой 8 FH. В команду входят 2 байта. В данном представлении 2-ой байт, который содержит подлежащую загрузке константу, добавляется сзади. «Immediate» и «Register Indirect» скомбинированы в команде **MVI M, data**. Байт 2 загружается в ЗУ, адрес которого расположен в паре регистров HL.

Таблица 16.7. Передача команд с адресацией типа «Immediate». В = байты, Z = циклы, T = такты.

Команда	Операнд	Воздействие	В	Z	T
MVI	r, data	$(r) \leftarrow (\text{byte } 2)$	1	2	7
MVI	M, data	$((H)(L)) \leftarrow (\text{byte } 2)$	1	2	7
LXI	rp, data16	$(rh) \leftarrow (\text{byte } 3), (rl) \leftarrow (\text{byte } 2)$	1	2	7

16.10.4. Адресация типа «Direct»

Байт 2 и байт 3 команды при адресации типа «Direct» содержат адрес, под которым можно найти операнд.

LHLD addr означает «Load H and L direct». Пара регистров HL загружается содержанием ячейки ЗУ с заданным адресом. К 5 требуемым циклам относятся OF, MR (чтение младшего байта адреса), MR (чтение старшего байта адреса), MR (чтение младшего байта элемента данных), MR (чтение старшего байта элемента данных). Итак, переносятся 2 байта.

SHLD addr означает «Store H and L direct», «записывание в H и L, прямое».

IN port означает «Input» (вход). Данные, которые выдаются на определенный через port вывод для подключения периферии, загружаются в накапливающий регистр. Port представляет собой 8-битовый адрес. В противоположность переходу с «управляемой памятью» ЗУ (memory-mapped) требуются только 3 цикла: OF, MR, MW.

OUT port соответствует «Output», то есть «выходу». Запомненные в накапливающем регистре данные выдаются через периферийный блок, адрес которого задан элементом программы port.

Команды IN и OUT приводят к установлению $IO/\overline{M} = 1$. При других командах табл. 16.8 будет установлено $IO/\overline{M} = 0$.

Таблица 16.8. Команда перехода с адресацией типа «Direct». В = байты, Z = циклы, T = такты.

Команда	Операнд	Воздействие	В	Z	T
LDA	addr	$(A) \leftarrow ((\text{byte } 3)(\text{byte } 2))$	3	4	13
STA	addr	$((\text{byte } 3)(\text{byte } 2)) \leftarrow (A)$	3	4	13
LHLD	addr	$(L) \leftarrow ((\text{byte } 3)(\text{byte } 2)),$ $(H) \leftarrow ((\text{byte } 3)(\text{byte } 2)+1)$	3	5	16
SHLD	addr	$((\text{byte } 3)(\text{byte } 2)) \leftarrow (L),$ $((\text{byte } 3)(\text{byte } 2)+1) \leftarrow (H)$	3	5	16
IN	port	$(A) \leftarrow (\text{data})$	2	3	10
OUT	port	$(\text{data}) \leftarrow (A)$	2	3	10

16.11. Арифметические команды

Арифметические команды отличаются от команд перехода тем, что они воздействуют на флаг-регистр. В приведенных в приложении таблицах флаги, которые устанавливаются в результате операции, перемаркированы значком X. При этом 1 означает, что флаг установлен, 0 — что убран.

В нашем распоряжении имеются команды сложения и вычитания, соответствующие типам адресации «Register», «Register indirect» и «Immediate». Кроме того, учитывается либо не учитывается стоящий перед операцией в флаг-регистре был переноса (carry-bit). В качестве примера приведены адресуемые по типу «Register indirect» команды сложения, которые учитывают carry-bit (ADC M) либо игнорирующего (ADD M):

ADD M Воздействие: $(A) \leftarrow ((H)(L))$

ADC M Воздействие: $(A) \leftarrow ((H)(L)) + (Cy)$

Команда CPI представляет собой тестовое вычитание (A) – Byte 2 и влияет на содержание накапливающего регистра не оказывает. Например, последовательных команд:

MVI A, 0FH

CPI 0FFH

действует так, что во флаг-регистре стоят следующие значения:

$S = 0, Z = 0, AC = 1, P = 0, Cy = 1.$

Далее в накапливающем регистре стоит 0FH.

Для циклов счета, для обращения с массивами, для банков данных и т. д. существуют специальные команды, с помощью которых

содержание регистра или ЗУ может быть увеличено или уменьшено. Они собраны в табл. 16.9. На carry-bit (бит переноса) влияние не сказывается, но есть возможность контролировать с помощью нулевого бита окончание цикла счета.

Таблица 16.9. 8-битовые команды на увеличение либо на уменьшение.

Команда	Операнд	Воздействие	B	Z	T	S	Z	A C	P	C Y
INR	r	$(r) \leftarrow (r) + 1$	1	1	4	x	x	x	x	
INR	M	$((H)(L)) \leftarrow ((H)(L)) + 1$	1	3	10	x	x	x	x	
DCR	r	$(r) \leftarrow (r) - 1$	1	1	4	x	x	x	x	
DCR	M	$((H)(L)) \leftarrow ((H)(L)) - 1$	1	3	10	x	x	x	x	

В нашем распоряжении имеются специальные команды для арифметических операций (табл. 16.10). При этом учитывается, что команда DAD «add register pair to H and L» воздействует только на бит переноса Carry-bit Cy. Две других команды не ведут к установлению каких-либо флагов.

Таблица 16.10. Арифметические 16-битовые команды с типом адресации «Register».

Команда	Операнд	Воздействие	B	Z	T	S	Z	A C	P	C Y
DAD	rp	$((H)(L)) \leftarrow (H)(L) + (rh)(rl)$	1	3	10					x
INX	rp	$(rh)(rl) \leftarrow (rh)(rl) + 1$	1	1	6					
DCX	rp	$(rh)(rl) \leftarrow (rh)(rl) - 1$	1	1	6					

Для арифметических операций в двоично-десятичном коде имеется команда, которая после сложения чисел в двоично-десятичном коде корректирует перенос (табл. 16.11). Если из-за сложения образуются «псевдотемрады» (то есть шестнадцатиричные числа больше 9) результат с помощью добавления 6 коррелируется.

Таблица 16.11. Команда для арифметических операций в двоично-десятичном коде.

Команда	Операнд	Воздействие	B	Z	T	S	Z	A C	P	C Y
DAA		Коррекция числа в двоично-десятичном коде	1	1	4	x	x	x	x	x

Действие команды DAA является следующим:

- 1) Если 4 последних разряда содержания накапливающего регистра выше, чем 9H, или AC = 1, к содержанию накапливающего регистра добавляется 6H.
- 2) Если первые 4 разряда содержания накапливающего регистра выше, чем 9H, или если Cy = 1, к 4 первым разрядам содержания накапливающего регистра добавляются 6H.

В следующем примере показано, что после сложения двух чисел в двоично-десятичном коде проводится двухшаговая коррекция.

Пример

Сложение двух чисел в двоично-десятичном коде (BCD):

00101000	28 ₁₀
+00110111	37 ₁₀
01011111	1111 ₂ > 9, отсюда коррекция с помощью добавления 110 ₂ = 6 ₁₀
+00000110	+6 ₁₀
01100101	65 ₁₀

В основе этого примера лежит следующий ассемблерный код:

```

MVI A, 28H ; 00101000BCD или 2810 загрузить в накапливающий
                ; регистр
ADI 37H ; 00110111BCD или 3710 к A, тогда (A) = 5FH
DAA ; (A) = 01100101BCD или 6510

```

16.12. Логические операции

Логические операции проводятся поразрядно. В нашем распоряжении находятся AND, OR и EXCLUSIVE OR. Кроме того, содержание накапливающего регистра может быть усложнено. Логические операции воздействуют на флаги. При некоторых операциях флаги устанавливаются независимо от результата. Команды для выполнения логических операций приведены в приложении.

Типичным случаем применения является «маскирование» битов в ЗУ. Пусть в этом случае необходимо перепроверить, равен ли 0 бит 4 накапливающего регистра. В этом случае должен быть установлен флаг Z. Для этого достаточно команды «AND immediate» («И немедленно»).

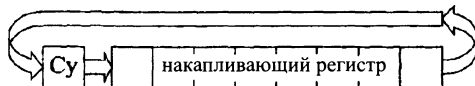
16.13. Установка и стирание флагов переноса (carry flag)

С помощью команд CMC (complement carry, «дополняющий перенос») и STC (set carry, «установочный перенос») можно воздействовать на флаг переноса (carry flag).

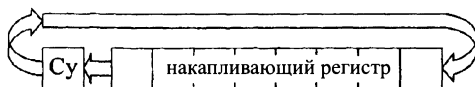
16.14. Команды сдвига

Различают левый и правый сдвиг, при этом carry-bit привлекается, либо нет. На следующих диаграммах показан принцип действия четырех возможных команд сдвига.

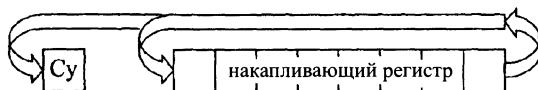
Поворот направо с переносом



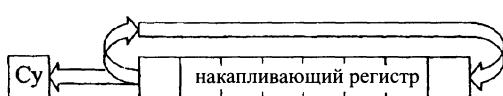
Поворот налево с переносом



Поворот направо



Поворот налево



16.15. Разветвление программ

Команды перехода изменяют нормальную последовательность команд, которая задается возрастанием значения счетчика команд. Ассемблерные программы вызывают на переход (jump) к конкретному адресу в ЗУ. Команда JMP всегда выполняется, как и команда PCHL, которая обсуждалась выше. Напротив, команды JNZ, JZ,

JNC, JC и т. д. выполняемая только тогда, когда выполнены заданные условия. Возможные условия, касающиеся флагов, перечислены в табл. 16.12. Обусловленный переход требует еще одного цикла.

Таблица 16.12. Команды перехода.

Команда	Операнд	Воздействие, если условие выполнено	Условие	B	Z	T
JMP	addr	$(PC) \leftarrow (byte3)(byte2)$	нет	3	3	10
JNZ	addr	$(PC) \leftarrow (byte3)(byte2)$	Z = 0	3	2/3	7/10
JZ	addr		Z = 1			
JNC	addr		CY = 0			
JC	addr		CY = 1			
JPO	addr		P = 0			
JPE	addr		P = 1			
JP	addr		S = 0			
JM	addr		S = 1			

16.15.1. Пример перехода

В следующем примере рассмотрим переходы в последовательности команд, состоящей из LXI и MOV. Вначале адрес цепи перехода неизвестен.

```
JMP          ; переход к цели перехода
LXI H, 0E00H ; загрузка пары регистров HL из 0E00H
MOV V, A     ; перезапись накопительного регистра в 0E00H
....        ; цель перехода
```

Чтобы установить адрес цели перехода, абсолютное положение программы в ЗУ должно быть фиксировано. Кроме того, должно быть отсчитано число байтов (команд и операндов). В примере, приведенном ниже, программа откладывается до адреса 1000H. Из этого получаем цель перехода 1007 H.

Адрес	Содержание	Коментарий
1000	C3	JMP
1001	07	Adress Low-Byte
1002	10	Adress High-Byte
1003	21	LXI
1004	C0	Operand Low-Byte
1005	0E	Operand High-Byte
1006	77	MOV M,A.
1007		Sprungziel (Adresse 1007)

Для программиста задача подсчета байтов между переходом и целью перехода может оказаться чреватой ошибками. Ассемблерная программа разрешает эту задачу автоматически. Для этого задают символические адреса переходов.

В последующих ассемблерных программах используют метку (label) LOOP, которая представляет собой символический адрес

```

ANA    A        ; тестирование накапливающего регистра
JNZ    LOOP     ; переход, если накапливающий регистр
                    ; равен 0
CMA                    ; дополнение накапливающего регистра
LOOP    ....

```

16.16. Команды для подпрограммы

Подпрограммы вызываются командой CALL («вызов»). При выдаче команды CALL осуществляются следующие шаги:

1. При вызове подпрограммы командой CALL вначале запоминается адрес обратного перехода путем размещения в стеке сначала старшего байта счетчика команд, а затем младших байтов. Ячейка ЗУ в стеке определяется в процессе вычитания 1 из адреса, на который показывает указатель вершины стека. Затем младший байт счетчика команд размещается в ячейке ЗУ с адресом (SP)-2.
2. Затем показания указателя вершины стека уменьшаются на 2. В том случае, если выбрано обычное представление, при котором ячейки ЗУ с более старшими адресами вносятся вниз, стек растет вверх. Указатель вершины стека после команды CALL показывает на самую старшую занятую позицию стека (top of stack = TOS).
3. Счетчик команд загружается адресом, имеющимся в операнде. Теперь на этом месте будет стоять следующая подлежащая выполнению команда.

При обратном переходе в соответствии с командой RET (return) производится:

1. Содержание ячейки ЗУ, на которую показывает указатель вершины стека, записываемый в младшем байте счетчика команд.

Содержание ячейки с адресом выше на 1 записывается в старшем байте PC ($PC = \text{program counter} = \text{instruction counter} = \text{счетчик команд}$).

2. Показание указателя вершины стека возрастает на 2.

Таблица 16.13. Команды перехода.

Команда	Операнд	Воздействие, если условие выполнено	Условие	B	Z	T
CALL	addr	$((SP) - 1) \leftarrow (PCH)$ $((SP) - 2) \leftarrow (PCH)$ $(SP) \leftarrow (SP) - 2$ $(PC) \leftarrow (\text{byte3})(\text{byte2})$	нет	3	5	18
CNZ	addr	$((SP) - 1) \leftarrow (PCH)$ $((SP) - 2) \leftarrow (PCL)$ $(SP) \leftarrow (SP) - 2$ $(PC) \leftarrow (\text{byte3})(\text{byte2})$	$Z = 0$	3	2/5	9/18
CZ	addr		$Z = 1$			
CNC	addr		$CY = 0$			
CC	addr		$CY = 1$			
CPO	addr		$P = 0$			
CPE	addr		$P = 1$			
CP	addr		$S = 0$			
CM	addr		$S = 1$			
RET		$(PCL) \leftarrow ((SP))$ $(PCH) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 2$	нет	1	3	10
RNZ		$(PCL) \leftarrow ((SP))$ $(PCH) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 2$	$Z = 0$	1	1/3	6/12
RZ			$Z = 1$			
RNC			$CY = 0$			
RC			$CY = 1$			
RPO			$P = 0$			
RPE			$P = 1$			
RP			$S = 0$			
RM			$S = 1$			
RST n $n \in \{0..7\}$		$((SP) - 1) \leftarrow (PCH)$ $((SP) - 2) \leftarrow (PCL)$ $(SP) \leftarrow (SP) - 2$ $(PC) \leftarrow n \times 8$	нет	1	3	12

Вызовы подпрограммы не влияют на флаги. Дальнейшее выполнение подпрограммы продолжается так, как это можно видеть в таблице.

Команда RST (restart, «перезапуск») вызывает после запоминания РС новый старт программы по определенному адресу. Стартовый адрес является результатом умножения двоичного эквивалента операнда на 8.

Например, для RST $3_{10} = 11_2$ стартовым адресом является

0000000000011000.

С помощью команд PUSH и POP содержание регистров может быть размещено в ЗУ магазинного типа (то есть стеке) и выведено вновь, чтобы избежать его разрушения при операциях подпрограммы. С помощью команд PUSH PSW и POP PSW можно сохранить (содержимое накапливающего регистра и флаг-регистра = processor status word, PSW, слева состояния процессора). Команда XTHL производит взаимную замену пары регистров HL и TOS (top of stack, вершина стека).

Указатель вершины стека может быть загружен в регистрную пару HL с помощью следующих команд

LXI H, 0000H

DAD SP

Указатель вершины стека должен быть инициализирован. Как правило, его устанавливают по самому старшему адресу зоны RAM+1. Поскольку он перед первой записью уменьшает свое значение на 1, он будет при этом показывать на старший адрес. Инициализация может быть осуществлена с помощью команды LXISP.

Таблица 16.14. Команды для стека.

Команда	Операнд	Воздействие, если условие выполнено	B	Z	T
PUSH	rp (rp $\neq$ SP)	$((SP) - 1) \leftarrow (rh)$ $((SP) - 2) \leftarrow (rl)$ $(SP) \leftarrow (SP) - 2$	1	3	12
PUSH	PSW	$((SP) - 1) \leftarrow (A)$ $((SP) - 2) \leftarrow (????????)$ $(SP) \leftarrow (SP) - 2$	1	3	12
POP	rp (rp $\neq$ SP)	$(rl) \leftarrow ((SP))$ $(rh) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 2$	1	3	10
POP	PSW	$(????????) \leftarrow ((SP))$ $(A) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 2$	1	3	10

В табл. 16.15 показана подпрограмма, в которой запоминаются содержания всех регистров. Это необходимо только тогда, когда они используются и в подпрограмме. Без запоминания в стеке было бы утеряно содержание, которое еще может потребоваться в вызывающей программе.

Таблица 16.15. Структура подпрограммы.

Адрес	Машин. код	Метка	Операц. код	Операнд	Комментарии
1000	31 01 18		LXI	SP, 1801H	Инициализация стека
1003	01 EC 0F		LXI	B, 0FECH	Загрузка в BC
1006	11 D1 80		LXI	H, 1CFEH	Вызов подпрограммы
1009	21 FE 1C		LXI	H, 1CFEH	
100C	CD 00 12		CALL	UTPR	
...	...		...	...	...
1200	C5	UTPR	PUSH	B	Запоминание BC
1201	D5		PUSH	D	
1202	E5		PUSH	H	
1203	F5		PUSH	PSW	Запоминание (A) и флагов
...	...		...	...	Команды подпрограммы
120F	F1		POP	PSW	Получить накопительный регистр и флаги
1210	E1		POP	H	Получить назад HL
1211	D1		POP	D	
1212	C1		POP	B	
1213	C9		RET		Обратный переход

В табл. 16.16 показано содержание стека во время выполнения подпрограммы в табл. 16.15. В этом примере принято, что в накапливающем регистре стоит EOH и что флаг-регистр содержит 08H.

16.17. Команды для управления процессором

Команда HLT (half, «останов») приводит процессор в состояние ожидания. Он выходит из этого состояния только при воздействии команд прерывания RESET или TRAP. Другие прерывания должны быть сначала окончены, чтобы исключить воздействие на состояние останова (halt).

NOP означает no operation (НОП, «пустая команда»). Команда служит как «держатель места» (place holder) для команд, которые должны быть использованы позднее, и как «потребитель времени».

EI (разрешение прерывания) и **DI** (запрещение прерывания) устанавливают и запирают **INTE**-триггер. Если **INTE**-триггер заперт, все запираемые прерывания блокированы.

Таблица 16.16. Стек, соответствующий примеру, показанному в табл. 16.15. Показано состояние, имеющее место во время выполнения подпрограммы (**TOS** = top of stack)

ЗУ		Комментарий
Адрес	Содержание	
17F7	08	PSW (FLAG) ←TOS при PC = 1204
17F8	E0	PSW (A)
17F9	FE	L
17FA	1C	H
17FB	D1	E
17FC	80	D
17FD	EC	C
17FE	0F	B
17FF	0F	PCL
1800	10	PCH
1801	FE	←TOS перед вызовом подпрограммы

С помощью команд **RIM** (read interrupt mask, «считывание маски прерываний») и **SIM** (set interrupt mask, «установка маски прерываний») можно считать и протестировать регистр прерываний. Значение соответствующих битов различно в зависимости от того, считывается или загружается регистр прерываний. Загрузка регистра прерывания (**SIM**):

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SOD	SOE	x	R7.5	MSE	M7.5	M6.5	M5.5

Бит 0: блокирование прерывания RST5.5

Бит 1: блокирование прерывания RST6.5

Бит 2: блокирование прерывания RST7.5

Бит 3: для записи содержания накапливающего регистра бит 3 должен быть равен 1

- Бит 4: возврат в исходное состояние (reset) триггера RST7.5 (RST7.5-триггер запоминает поступающие прерывания, также и тогда, когда прерывание заблокировано. Содержание RST7.5-триггер можно стереть с помощью бита 4 или возврата в исходное состояние).
- Бит 5: не определен
- Биты 6,7: управление связью с SOD.

Считывание регистра прерывания (RIM):

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SID	17.5	16.5	15.5	IE	M7.5	M6.5	M5.5

- Бит 1: показывает маску для прерывания RST5.5 (0 = разрешено)
- Бит 2: показывает маску для прерывания RST6.5 (0 = разрешено)
- Бит 3: показывает состояние INTE-триггера (1 = освобождено)
- Бит 4: 15.5 стоит на очереди
- Бит 5: 16.5 стоит на очереди
- Бит 6: 17.5 стоит на очереди

16.18. Ассемблерные команды

Различают ассемблерные команды и команды на ассемблерном языке.

- Команды на ассемблерном языке (напр., LDA 09FEN) устанавливают исходную программу (source-code)
- Ассемблерные команды служат для перевода исходной программы (source code) в объектную программу (object code). Они сообщают о том, с какого адреса программа записывается в ЗУ, и дают указания по размещению символических переменных по ячейкам ЗУ. Например, команда ORG 1000H говорит о том, что следующие команды должны быть размещены начиная с ячейки ЗУ 1000H.

Ниже обсуждаются команды на ассемблерном языке, соответствующие различным группам. Ассемблерная программа осуществляет перевод исходной программы (source code) в объектную программу, но наряду с этим решает и другие задачи:

1. В общем случае машинная программа привязана к определенным ячейкам в ЗУ. Например, при переходах всегда должны быть заданы адреса переходов. Но эти адреса зависят от положения в программе.

Чтобы упростить работу программиста, ассемблерная программа в состоянии проводить расчет конкретных адресов (физических адресов) на основе символических адресов перехода. Сверх того в ассемблер должно быть сообщено, где программа должна находиться в ЗУ. Это реализуется командой:

OR G addr

2. Ассемблеру должно быть сообщено, где команда оканчивается, потому что там оканчивается перевод. Для этого служит **END**.
3. Точно также могут быть обработаны символические имена «name», относящиеся к константам ассемблера. При этом каждой константе присваивается 8-битовое двоичное число.

Имя	Оператор	Операнд	Воздействие
имя	DB	Data	Определяет и инициализирует 8-битовую переменную
имя	DW	Data 16	Определяет и инициализирует 16-битовую переменную.
имя	DS	n	Резервирует n байт

4. Ассемблером игнорируются комментарии, которые должны начинаться точкой с запятой.
5. Ассемблерная программа производит проверку на синтаксические ошибки.
6. Благодаря определению макросов (macro, макрокоманда) затраты на программирование могут быть уменьшены.

При ассемблировании используется много проходов (pass). Сначала переводятся команды. В течение дальнейших проходов распределяются символические адреса и цели переходов. Перевод команд в машинный код производится с помощью списков, специфических для данного процессора.

Таблица 16.17. Пример применения ассемблерных операторов.

Адрес	Машин. код	Метка	Операц. код (op code)	Операнд	Комментарий
		K1	EQU	0A1H	; наложение константы
		PRT	EQU	01H	; наложение адреса порта
			ORG	1000H	; определение зоны команд
1000	3A A1			A, K1	; переход в МК
1002	C3 0C 10		JMP	МК	; добавление K1
1005	C6 A1		AD1	K1	; перезапись в VAR1
1007	32 00 13		STA	VAR1	; выдача в порт
100A	D3 01		OUT	PRT	; загрузка в VAR2
100C	3A 01 13	МК	LDA	VAR2	; зона переменных
			ORG	1300H	
1300		VAR1	DS	1	; 1 байт для VAR
1301	FE	VAR2	DB	IFEH	; инициализация переменных

В примере используются ассемблерные операторы ORG, EQU, DS, DB.

С помощью оператора ORG определяется положение зоны в ЗУ, которая содержит команды (от адреса 1000H) и данные (от адреса 1000H). Ассемблер присваивает затем командам и меткам перехода соответствующие адреса.

Здесь метка МК определяется из ассемблерной программы. Ассемблер вводит в этом случае адрес перехода 1000H. Адрес перезаписывается в последовательность из младших и старших битов.

Оператор EQU применяется для определения констант K1 и PRT. Когда в тексте встречаются эти константы, ассемблер вставляет соответствующие значения. Это служит обеспечению наглядности программы.

Оператор DS вначале резервирует ячейку ЗУ и присваивает ей адрес и символическое имя. В данном случае имеем имя VAR1 и адрес 1300. Теперь программист может обращаться с VAR1 как с адресом, это означает с использованием команд, в которых применяется тип адресации «Direct» (например, LDA addr. STA addr.). При использовании DB, также как и при использовании DS, резервируется ячейка ЗУ и фиксируется символическое имя. В случае

ДВ дополнительно производимая инициализация. В данном случае в ячейку ЗУ 1301 записывается значение FE. Оно может быть перезаписано с помощью программы. В примере необходимая инициализация портов не проводилась.

16.19. Примеры программ

16.19.1. Программирование параллельного интерфейса 8255

Параллельный интерфейс 8255 должен быть запрограммирован перед вводом и выводом данных. Для этого каналы A, B CLow и CHigh должны быть запрограммированы на «считывание данных» и на «пересылку данных».

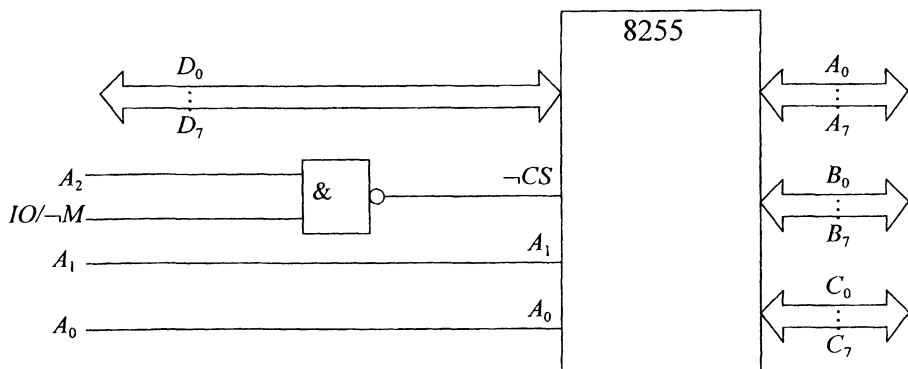


Рис. 16.11. Подключение параллельного интерфейса 8255 (показаны только важнейшие разъемы).

В рассматриваемом случае должно быть принято, что интерфейс должен быть подключен так, как это показано на рис. 16.11. Итак, адреса интерфейса лежат в диапазоне от 0004H до 007H. В соответствии с рис. 16.11 выбирается командный регистр с $A_0 = A_1 = 1$. Он имеет адрес 0007H. Адреса регистров данных указаны в табл. 16.18.

Таблица 16.18. Адреса регистров параллельного интерфейса 8255. Вывода в соответствии с рис. 16.11.

Адрес	Регистр
0004H	Регистр данных А-порта
0005H	Регистр данных В-порта
0006H	Регистр данных С-порта
0007h	Командный регистр

Командный регистр для режима работы 0 выстроен следующим образом:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	A	CH	0	B	CL

Если какой-либо канал должен быть подключен к входу, должен быть установлен соответствующий бит $CL = C_{Low}$, $CH = C_{High}$. В режиме работы 0 данные хранятся в регистре данных пока они не будут перезаписаны на новые данные.

Таблица 16.19. Пример программы для ввода и выдачи данных для параллельного интерфейса 8255.

Адрес	Машин. код	Метка	Операц. код (op code)	Операнд	Комментарий
			ORG	1000H	; зона команд
1000	3E 90		MVI	A, 90H	; управляющий байт A вход, BС выход
1002	B3 70		OUT	07H	; запоминание управляющего бита
1004	BI 04		IN	04H	; считывание, канал A
1006	B3 05		OUT END	05H	; выдача, канал B

16.19.2. Передача данных в подпрограмму

Передача данных из вызываемой программы в подпрограмму происходит, как правило, через регистры. Если их не хватает, данные после вызова подпрограммы могут быть отложены.

В примере, показанном в табл. 16.20, в подпрограмму передается число 1F88H. Для этого с помощью команды **XTHL** приносится содержание РС из стека. Затем при **INXH** пропускается команда **JMP**, так что в паре регистров HL стоит адрес младшего байта отыскиваемого элемента данных. Теперь он может быть загружен с использованием индикации. При следующей команде **INXH** осуществляется дальнейшее просчитывание к адресу старшего байта элемента данных, и он загружается с индикацией. С помощью такого порядка действий может быть обработан весь список параметров.

Другие примеры программ и подробное введение в программирование микропроцессора 8085 с помощью ассемблера можно найти в литературе, позиции [18] и [16].

Таблица 16.20. Программный пример передачи данных в подпрограмму [16, 18].

Адрес	Машин. код	Метка	Операц. код (op code)	Операнд	Комментарий
1000	CD 00 12		ORG	1000H	; зона команд
1003	C3 09 10		CALL	PROG	; вызов подпрограммы
1006	88 1F		JMP	LIST	; пропуск (skip) константы
1008			DW	1F88H	; подлежащая передаче константа
		LIST	...	...	;
			...	...	;
			...	...	;
1200	E3	PROG	XTHL		; PC в HL
1201	E5		PUSH	H	; PC в стек ("на стек")
1202	D5		PUSH	D	; DE в стек
1203	F5		PUSH	PSW	; PSW в стек
1204	23		INX	H	; пропуск команды 1205OMXJMP
1205	23		INX	H	-"-
1206	23		INX	H	-"-
1207	5E		MOV	E, M	; данные, соответствующие младшему байту, в E
1208	23		INX	H	; переход к старшему байту
1209	5E		MOV	D, M	; данные, соответствующие старшему байту в B
			...	...	; подпрограмма
			...	...	; подпрограмма
1210	F1		POP	PSW	; PSW из стека ("со стека")
1211	D1		POP	D	; DE из стека
1212	E1		POP	H	; PC из стека
1213 1214	E3 C9		XTHL RET END		; HL и PC, взаимная замена

16.20. Упражнения

Задача 16.1.

- а) Перечислите виды адресации, применяемые в микропроцессоре с 8085.
- б) Объясните различные времена выполнения адресаций в микропроцессор 8085.
- в) Что происходит при прерывании?
- г) Объясните понятия «стек» и «указатель вершины стека».
- д) Какие шаги реализуются командами CALL и RET при вызове подпрограммы.

Задача 16.2. Напишите программу, обеспечивающую взаимную замену двух чисел. Числа должны находиться в ячейках ЗУ, адреса которых стоят в парах регистров DE и HL. Сколько тактов требуется для выполнения вашей программы? Как долго будет выполняться ваша программа, если тактовая частота процессора равна 2 МГц?

Задача 16.3. Запишите сегмент программы, который сдвигает блок переменной длины (n байт < 255) с одной позиции в ЗУ на другую. Зоны ЗУ не должны перекрываться. Количество подлежащих сдвигу байтов должно стоять в регистре В. В регистровой паре HL должен быть загружен первый адрес подлежащего сдвигу блока в регистровую пару DE первый адрес нового блока.

Задача 16.4. Напишите подпрограмму, которая подсчитывает количество единиц в накапливающем регистре. В заключение количество единиц должно быть передано в накапливающий регистр, соответствующий главной программе.

А. Приложение

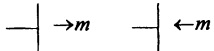
А.1. Индикация функций

Символ	Описание
&	И-вентиль
≥ 1	ИЛИ-вентиль
=1	EXOR-вентиль
=	вентиль эквивалентности
2k	четное число входных сигналов должно иметь значение 1
2k+1	нечетное число входных сигналов должно иметь значение 1
1	вход должен быть в состоянии 1
< или >	выход задающего каскада, сигнал ориентирован в направлении протекания потока сигналов
X/Y	преобразователь кода
MUX	мультиплексер
DMUX или DX	демультиплексер
Σ	сумматор
P-Q	вычитатель
CPG	carry-lock-ahead-generator (генератор с опережением переноса)
π	устройство умножения
COMP	устройство сравнения, компаратор
ALU	арифметико-логическое устройство
SRGm	сдвиговый регистр на m бит
CTRm	счетчик на бит, длина цикла 2^m
CTRDIVm	счетчик с длиной цикла m
RCTRm	асинхронный счетчик с длиной цикла 2^m
ROM	ЗУ только со считыванием (read-only-memory)
RAM	ЗУ с записью и считыванием
FIFO	ЗУ типа first-in-first out (первый записывается — первый считывается)

В следующей таблице собраны логические символы, указываемые вне обрамления, символа:

	Символ	Функция
1		Логическая инверсия входа (внешний 0 порождает внутреннюю 1)
2		Логическая инверсия выхода (внутренний 0 порождает внешнюю 1)
3		Вход, активен при L, эквивалентен N1 в положительной логике
4		Вход, активен при L, протекание потока сигналов справа налево
5		Выход, активен при L, эквивалентен N2 в положительной логике
6		Протекание потока сигналов справа налево
7		Двухнаправленный поток сигналов
8		Динамический вход: активен при появлении положительного фронта
9		Динамический вход: активен при появлении отрицательного фронта
10		Нелогический вход
11		Аналоговый вход при представлении цифровым символом
12		Внутреннее соединение
13		Инвертирующее внутреннее соединение
14		Внутреннее соединение: активно при появлении положительного фронта

Логические символы следующей таблицы лежат внутри обрамления символа. С помощью этого делаются высказывания о внутреннем логическом состоянии схемы.

Символ	Функция
	Буферный выход: изменение только при достижении первоначального состояния входа
	Вход с гистерезисом
	Выход с открытым коллектором $n-p-n$ -транзистора
	Выход с открытым эмиттером $p-n-p$ -транзистора
	Выход типа tri-state
	Выход разрешающий
J, K, R, S, D, T	Входы триггеров: обычное значение букв
	Входы, определяющие сдвиг вправо или сдвиг влево в сдвиговом регистре, $m \in N$, $m = 1$ как правило не указывается
	Двоичный входной вектор со значениями от 0 до n . n является второй степенью значения старшего бита.
	Устанавливающий входной сигнал, заданное значение загружается, если вход активен
	Выход переходит в состояние «?», если регистр принимает заданное значение
	Выход с постоянным значением
	Группа сигналов, образующих единый логический вход
	Внутреннее соединение: активно при положительном фронте

В этой таблице собраны указания на типы индикации зависимости.

Функция	Символ	Вход в состоянии «1»	Вход в состоянии «0»	Стр.
Адрес	A	Выбирается адрес	Адрес не выбирается	164
Контроль	C	активирован	без изменений	82
Разрешение (enable)	EN	активирован	входы не действуют, выходы типа tri-state являются высокоомными, выходы с открытыми коллекторами отключены другие выходы в состоянии 0	38,39
«И»	G	«И» совместно с другими входами	вынуждается -	27
Режим работы (mode)	M	Режим выбирается	Режим не выбирается	135
Отрицание (negation)	N	Состояние отрицания	Нет влияния	29
Возврат в исходное состояние (reset)	R	Триггер устанавливается в исходное состояние	Нет влияния	80
Установка (set)	S	Установка триггера	Нет влияния	80
«ИЛИ»	V	Вынужденная «1»	«ИЛИ» с другими входами	28
Передача	X	Создается двунаправленная связь	Связь разорвана	30
Связь	Z	Вынужденная	Вынужденный 0	29

Решения задач

Решение задачи 2.1.

а)

$$g(1110, 101_2) = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 14,625_{10}$$

б)

$$g(10011, 1101_2) = 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-4} = 19,8125_{10}$$

Решение задачи 2.2.

а)

$33 : 2 = 16$	остаток 1	$\uparrow$ целочисленная часть двоич- ного числа
$16 : 2 = 8$	остаток 0	
$8 : 2 = 4$	остаток 0	
$4 : 2 = 2$	остаток 0	
$2 : 2 = 1$	остаток 0	
$1 : 2 = 0$	остаток 1	

$0,125 \cdot 2 = 0,25$	$+0$	$\downarrow$ дробная часть двоичного числа
$0,25 \cdot 2 = 0,5$	$+0$	
$0,5 \cdot 2 = 0$	$+1$	

Отсюда $33,125_{10} = 100001,001_2$.

б)

$45 : 2 = 22$	остаток 1	$\uparrow$ целочисленная часть двоич- ного числа
$22 : 2 = 11$	остаток 0	
$11 : 2 = 5$	остаток 1	
$5 : 2 = 2$	остаток 1	
$2 : 2 = 1$	остаток 0	
$1 : 2 = 0$	остаток 1	

$$\begin{array}{rcl}
 0,33 \cdot 2 = 0,66 & +0 & \\
 0,66 \cdot 2 = 0,32 & +1 & \left. \begin{array}{l} \text{дробная часть} \\ \text{двоичного числа} \end{array} \right\} \\
 0,32 \cdot 2 = 0,64 & +0 & \\
 0,64 \cdot 2 = 0,28 & +1 &
 \end{array}$$

Теперь двоичное число известно до четвертого знака после запятой. Отсюда $45,33_{10} \approx 101101,0101_2$.

Решение задачи 2.3.

а) Точным дополнением для 001010 является 110110

$$\begin{array}{rcccccccc}
 & & & 0 & 1 & 0 & 1 & 0 & 1 & 21_{10} \\
 + & & & 1 & 1 & 0 & 1 & 1 & 0 & -10_{10} \\
 \hline
 \text{перенос} & 1 & 1 & 0 & 1 & 0 & 0 & & & \\
 = & (1) & 0 & 0 & 1 & 0 & 1 & 1 & & 11_{10}
 \end{array}$$

Имели место переносы c_5 и c_6 , поэтому результат верен.

б) Точным дополнением 010111 является 101001, точным дополнением 011011 является 100101.

$$\begin{array}{rcccccccc}
 & & & 1 & 0 & 1 & 0 & 0 & 1 & -23_{10} \\
 + & & & 1 & 0 & 0 & 1 & 0 & 1 & -27_{10} \\
 \hline
 \text{перенос} & 1 & 0 & 0 & 0 & 0 & 1 & & & \\
 = & (1) & 0 & 0 & 1 & 1 & 1 & 0 & & 14_{10}
 \end{array}$$

Поэтому здесь справедливо $c_5 = 0$ и $c_6 = 1$, отсюда результат неверен.

Решение задачи 2.4.

а) $110101 \cdot 010101 = 010001011001$

б) $1101110 : 110 = 10010, \overline{01}$.

Решение задачи 2.5. Например: 000, 001, 011, 010, 110, 100.

Решение задачи 3.1. Доказательство с помощью таблицы истинности.



Уравнение 3.10

x_1	x_0	$x_0 \vee x_1$	$x_0 \wedge (x_0 \vee x_1)$	x_0
0	0	0	0	0
0	1	1	1	1
1	0	1	0	0
1	1	1	1	1

Уравнение 3.11

x_1	x_0	$x_0 \wedge x_1$	$x_0 \vee (x_0 \wedge x_1)$	x_0
0	0	0	0	0
0	1	0	1	1
1	0	0	0	0
1	1	1	1	1

Решение задачи 3.2.

$$\begin{aligned}
 y &= (x_0 \wedge x_1 \wedge x_2 \wedge \neg x_3) \vee (x_0 \wedge x_1 \wedge x_2 \wedge x_3) \vee (\neg x_0 \wedge \neg x_1 \wedge x_2 \wedge x_3) \\
 &\vee (\neg x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_0 \wedge \neg x_1 \wedge x_2 \wedge x_3) \vee \\
 &\vee (x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3) = \\
 &= (x_0 \wedge x_1 \wedge x_2) \vee (\neg x_0 \wedge \neg x_1 \wedge x_3) \vee (x_0 \wedge \neg x_1 \wedge x_3) = \\
 &= (x_0 \wedge x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3)
 \end{aligned}$$

Последнее уравнение является искомым минимальным представлением.

Решение задачи 3.3. Формирование таблицы истинности

a	0	0	0	0	1	1	1	1
b	0	0	1	1	0	0	1	1
c	0	1	0	1	0	1	0	1
s_1	0	0	0	1	0	1	1	1
s_0	0	1	1	0	1	0	0	1

KDNF (каноническая дизъюнктивная нормальная форма) для s_0 :

$$s_0 = (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c)$$

KDNF (каноническая дизъюнктивная нормальная форма) для s_1 :

$$s_1 = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (a \wedge b \wedge c)$$

KKNF (каноническая дизъюнктивная нормальная форма) для s_0 :

$$s_0 = (a \vee b \vee c) \wedge (a \vee \neg b \vee c) \wedge (a \vee b \vee \neg c) \wedge (\neg a \vee \neg b \vee c)$$

ККНФ (каноническая дизъюнктивная нормальная форма) для s_1 :

$$s_1 = (a \vee b \vee c) \wedge (a \vee b \vee \neg c) \wedge (a \vee \neg b \vee c) \wedge (\neg a \vee b \vee c)$$

Решение задачи 3.4.

$$s_1 = (b \wedge c) \vee (a \wedge c) \vee (a \wedge b) \text{ oder } s_1 = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$$

Нельзя дальше упрощать уравнения для s_0 , поскольку все термы различаются не менее, чем на 2 переменных.

Решение задачи 3.5.

а) $a \leftrightarrow \neg b = ab \vee \neg a \neg b = \neg a \leftrightarrow \neg b = \neg (a \leftrightarrow b)$

б) $f = a \leftrightarrow b \leftrightarrow c \Rightarrow \neg f = \neg (a \leftrightarrow b) \leftrightarrow c = \neg a \leftrightarrow b \leftrightarrow c$

Повторим с b и c :

$$\neg f = \neg a \leftrightarrow \neg b \leftrightarrow \neg c$$

Решение задачи 3.6.

а)

$$y_1 = x_1 x_2 x_3 \vee \neg x_2 x_3$$

$$= x_1 x_2 x_3 \vee \neg x_2 x_3 x_1 \vee \neg x_2 x_3 \text{ (закон абсорбции)}$$

$$= x_1 x_3 \vee \neg x_2 x_3 \text{ (объединение термов 1 и 2 в соответствии с 3.34)}$$

$$= x_3 (x_1 \vee \neg x_2) \text{ (закон дистрибутивности)}$$

б)

$$y_2 = \neg x_1 \neg x_2 \neg x_3 \vee \neg x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee$$

$$\vee x_1 \neg x_2 \neg x_3 \vee x_1 x_2 \neg x_3 \vee \neg x_1 x_2 \neg x_3 =$$

$$= \neg x_2 \neg x_3 \vee x_2 x_3 \vee x_2 \neg x_3 \text{ (объединение термов 1 и 4, 2 и 3, 5 и 6)}$$

$$= \neg x_2 \neg x_3 \vee x_2 x_3 \vee x_2 \neg x_3 \text{ (закон дистрибутивности)}$$

$$= \neg x_3 \vee x_2 \text{ (объединение термов 1 и 4, 2 и 3, 5 и 6)}$$

в)

$$y_3 = \neg x_1 x_2 \neg x_3 \vee \neg (x_1 x_2) \vee x_1 \neg x_2 \neg x_3 \vee \neg x_1 \neg x_2 x_3 x_4$$

$$= \neg x_1 x_2 \neg x_3 \vee \neg x_1 \neg x_2 \vee x_1 \neg x_2 \neg x_3 \vee \neg x_1 \neg x_2 x_3 x_4 \text{ (де-Морган)}$$

$$= \neg x_1 x_2 \neg x_3 \vee \neg x_1 \neg x_2 \vee x_1 \neg x_2 \neg x_3$$

(терм 4 из-за терма 2 опускается)

$$= \neg x_1 x_2 \neg x_3 \vee \neg x_1 \neg x_2 \vee x_1 \neg x_2 \neg x_3 \vee \neg x_1 \neg x_2 \neg x_3 \vee \neg x_1 \neg x_2 \neg x_3$$

(закон абсорбции)

$$= \neg x_1 \neg x_3 \vee \neg x_1 \neg x_2 \vee \neg x_2 \neg x_3 \text{ (объединение термов 1 и 4, 3 и 6)}$$

г)

$$\begin{aligned}
 y_4 &= \neg (\neg (\neg x_1 \neg x_2 \neg x_4) \neg (\neg x_1 \vee \neg x_2 \neg x_3)) \\
 &= \neg x_1 \neg x_2 \neg x_4 \vee \neg x_1 \vee x_2 \vee \neg x_3 \text{ (де-Морган)} \\
 &= \neg x_1 \vee \neg x_2 \vee \neg x_3 \text{ (закон абсорбции)}
 \end{aligned}$$

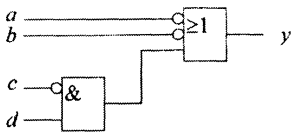
д)

$$\begin{aligned}
 y_4 &= \neg (\neg x_1 x_2 \neg x_3 \vee \neg (x_1 \vee x_2 \vee x_3)) (x_1 \vee \neg x_2) \\
 &= \neg (\neg x_1 x_2 \neg x_3 \vee \neg x_1 \neg x_2 \neg x_3) (x_1 \vee \neg x_2) \text{ (де-Морган)} \\
 &= \neg (\neg x_1 \neg x_3) (x_1 \vee \neg x_2) \text{ (уравнение 3.34)} \\
 &= (x_1 \vee x_3) (x_1 \vee \neg x_2) \geq \text{ (де-Морган)} \\
 &= x_1 \vee x_3 \neg x_2 \text{ (закон дистрибутивности)}
 \end{aligned}$$

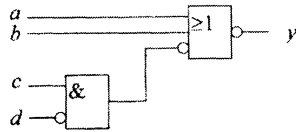
Решение задачи 4.1.

$$a) y = \neg (ab) \vee \neg cd \vee a \neg bd = \neg a \vee \neg b \vee \neg cd \vee a \neg bd = \neg a \vee \neg b \vee \neg cd$$

б) Положительная логика



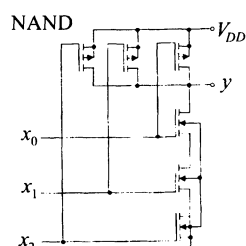
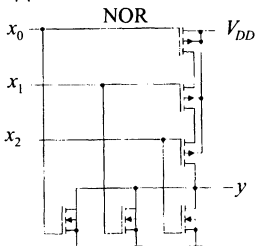
Отрицательная логика

**Решение задачи 4.2.**

Уровень напряжения		
x_2	x_1	y
L	L	H
L	H	H
H	L	H
H	H	L

Положительная логика, функция NAND		
x_2	x_1	y
0	0	1
0	1	1
1	0	1
1	1	0

Отрицательная логика, функция NOR		
x_2	x_1	y
1	1	0
1	0	0
0	1	0
0	0	1

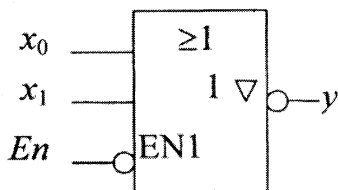
Решение задачи 5.1.

Решение задачи 5.2.

а)

x_0	x_1	E_n	y
1	1	0	0
1	0	0	0
0	1	0	0
0	0	0	1
d	d	1	ВЫСОКООМНЫЙ

б)

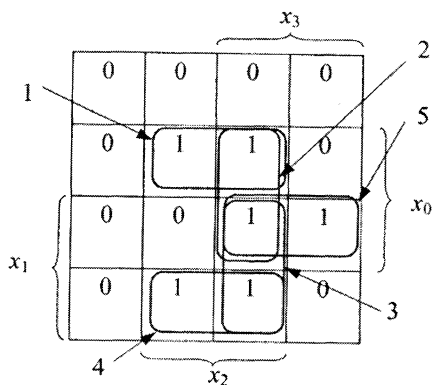


Решение задачи 5.3. Рассматривается вентиль эквивалентности:

$$s = \neg(x_0x_1); \quad y = \neg s \vee \neg x_0 \neg x_1 = x_0x_1 \vee \neg x_0 \neg x_1 = x_0 \leftrightarrow x_1$$

Решение задачи 6.1.

а)



б) Импликанты: 1) $x_0x_2\neg x_1$; 2) $x_0x_2x_3$; 3) $x_1x_2x_3$; 4) $\neg x_0x_1x_2$;

5) $x_1x_0x_3$

в) Первичные импликанты ядра: 1, 4, 5.

Абсолютно элиминируемый первичный импликант: 2,3.

Относительно элиминируемый первичный импликант: $\emptyset$

- г) Имеется только одно решение: $f = x_0x_2 \neg x_1 \vee \neg x_0x_1x_2 \vee x_1x_0x_3$
 д) Решение методом Куина-Мак-Класки

Десятичные	x_3	x_2	x_1	x_0	Группа	
5	0	1	0	1	2	✓
6	0	1	1	0		✓
11	1	0	1	1	3	✓
13	1	1	0	1		✓
14	1	1	1	0		✓
13	1	1	1	1		✓
15	1	1	1	1	4	✓

Объединение термов для получения второй таблицы

Десятичные	x_3	x_2	x_1	x_0	Группа
5, 13	—	1	0	1	2
6, 14	—	1	1	0	
11, 15	1	—	1	1	3
13, 15	1	1	—	1	
14, 15	1	1	1	—	

Дальнейшее объединение невозможно, откуда занесение в таблицу первичных импликантов:

	5	6	11	13	14	15
5, 13	⊗			⊗		
6, 14		⊗			⊗	
11, 15			⊗			⊗
13, 15				×		×
14, 15					×	×

Входящее ядро первичные импликанты 5, 13; 6, 14 и 11, 5 покрывают все минимермы. Поэтому минимальная форма состоит только из входящих в ядро первичных импликантов: $f = x_0x_2 \neg x_1 \vee \neg x_0x_1x_2 \vee x_1x_0x_3$.

Решение задачи 6.2.

- а) Решение для минимальной дизъюнктивной нормальной формы:

$$y = \neg x_3x_2 \wedge \neg x_1x_4 \wedge \neg x_0 \neg x_4$$

				x_3		x_4			
x_1	1	1	d	d	1	1	d	1	
	0	d	0	0	1	d	1	1	
	d	1	0	0	0	0	d	0	
	1	d	d	d	0	0	d	0	
				x_2				x_0	

б) Решение для минимальной конъюнктивной нормальной формы

$$y = (\neg x_1 \vee \neg x_4) (\neg x_3 \vee x_4) \wedge (\neg x_0 \vee x_1 \vee x_4)$$

		x_3				x_4					
x_1		1	1	d	d	1	1	d	1	x_0	
		0	d	0	0	1	d	1	1		
		d	1	0	0	0	0	d	0		
		1	d	d	d	0	0	d	0		
		x_2				x_2					

Решение задачи 6.3.

а)

		x_3				x_3		x_3	
x_1		1	1						
			1						
		x_2		x_2		x_2		x_2	

		x_3				x_3			
x_1			1						
			1						
			1						
		x_2		x_2		x_2		x_2	

		x_3				x_3		x_3	
x_1									
		x_2		x_2		x_2		x_2	

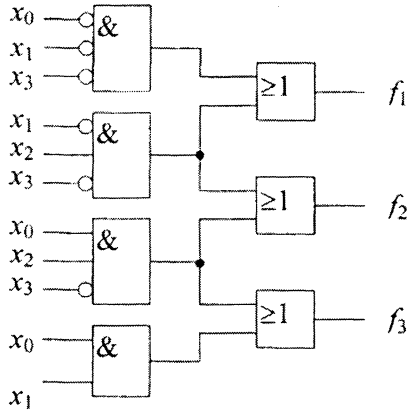
Дизъюнктивная нормальная форма для единичных функций:

$$f_1 = \neg x_0 \neg x_1 \neg x_3 \vee \neg x_1 x_2 \neg x_3;$$

$$f_2 = \neg x_1 x_2 \neg x_3 \vee \neg x_0 x_2 \neg x_3; f_3 = x_0 x_2 \neg x_3 \vee \neg x_0 x_1$$

б) Как f_1 и f_2 , так и f_2 и f_3 , имеют совместный терм, он реализуется только однажды.

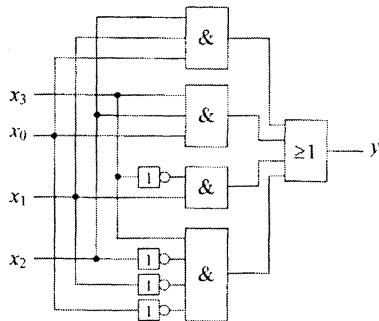
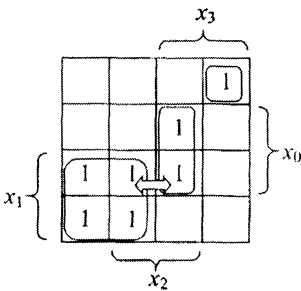
в) В затраты входят 7 вентиля с 17 входами в общей сложности



Решение задачи 6.4.

а) $y = x_0 x_2 x_3 \vee x_1 \neg x_3 \vee \neg x_0 \neg x_1 \neg x_2 x_3$

б)



в) Опасность сбоя (hazard) маркируется значком $\Leftrightarrow$.

Решение задачи 7.1.

а) Условия перехода считываем из структурной схемы:

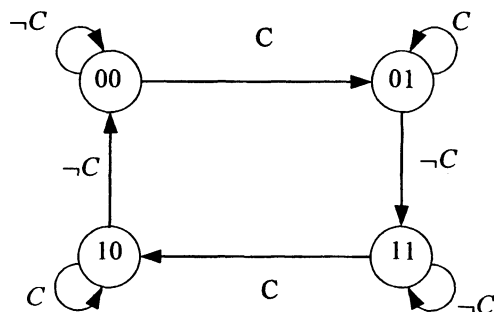
$$z_1^{m+1} = \neg C z_0^m \vee C z_1^m = \neg C z_0^m \vee C z_1^m \vee z_0^m z_1^m$$

$$z_0^{m+1} = \neg C z_0^m \vee C \neg z_1^m$$

Таблица состояний: $z_1^{m+1} z_0^{m+1}$

		00		01		
		11	01	11	10	
z_1^m	{	11	10	00	10	
		00	10			
				z_0^m		
		C				

б) Диаграмма состояний:

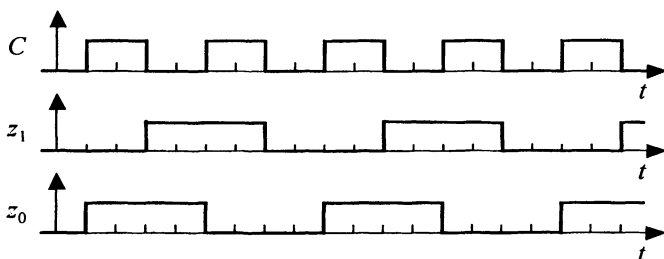


в) Реализация без опасности сбоя с помощью добавления двух избыточных термов:

$$z_1^{m+1} = \neg C z_0^m \vee C z_1^m = \neg C z_0^m \vee z_0^m z_1^m$$

$$z_0^{m+1} = \neg C z_0^m \vee C \neg z_1^m = \neg C \neg z_0^m \vee z_0^m \neg z_1^m$$

г) Из диаграммы состояний получаем временную диаграмму:

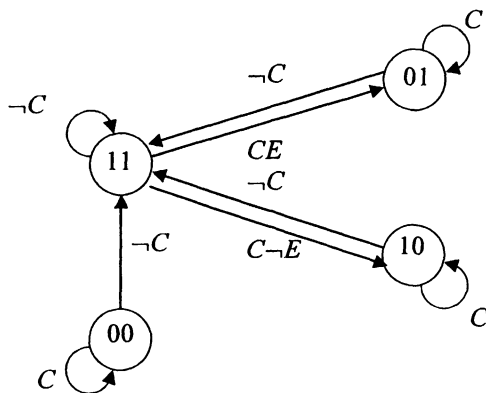


Анализируемая управляющая схема в интервал времени, определяемый двумя импульсами тактового сигнала, проходит циклы 01, 11, 10, 00. Вследствие этого с обоих выходов Z_1 и Z_2 можно снять сигнал с половинной частотой.

Решение задачи 7.2.

а) Диаграмма состояний (см. рисунок внизу) может быть сконструирована так, чтобы состояние $Z_1^n Z_0^m = 00$ соответствовало состоянию покоя, при котором ожидается фронт тактового импульса. Z_0^m может быть использовано как выходной сигнал. Когда подходит нарастающий фронт тактового сигнала, управляющая схема независимо от E переходит в состояние 01 или 10 и остается в нем до тех пор, пока тактовый сигнал имеет значение «1». Состояние 11 является «избыточным».

б) диаграмма состояний



в) таблица последовательности состояний

$z_1^{m+1} z_0^{m+1}$		C			
z_0^m		11	11	00	00
		11	11	01	01
		11	11	01	10
		11	11	10	10
		E			

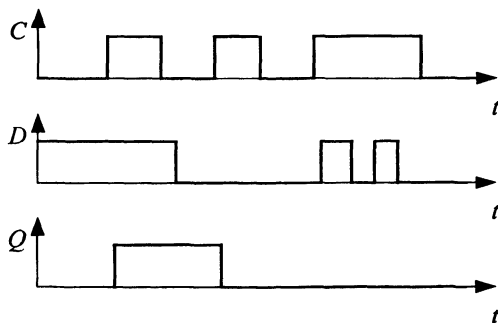
Схема проектируется на основе считывания передаточной функции из диаграммы Карно-Вейча. Выполнение функций должно быть свободно от опасности сбоя, так как эта схема является асинхронной.

Решение на основе дизъюнктивной нормальной функции:

$$z_1^{m+1} = (\neg C \vee z_1^m)(\neg C \vee \neg z_0^m \vee \neg E) = \neg(C \neg z_1^m \vee CEz_0^m)$$

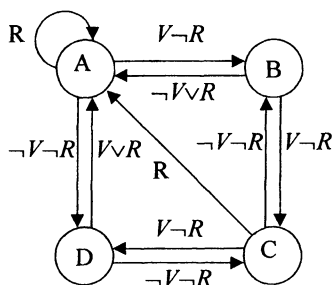
$$z_0^{m+1} = (\neg C \vee z_0^m)(\neg C \vee \neg z_1^m \vee E) = \neg(C \neg z_0^m \vee C \neg Ez_1^m)$$

Решение задачи 7.3.



Решение задачи 8.1.

а)



б)

симв.	$z_1^{m+1} z_0^{m+1}$	$\neg R \neg V$	$R \neg V$	RV	$\neg RV$
A	00	10	00	00	01
B	01	00	00	00	11
C	11	01	00	00	10
D	10	11	00	00	00

в)

$$D_1 = z_1^{m+1} = \neg z_0^m \neg V \neg R \vee z_0^m V \neg R$$

$$D_0 = z_0^{m+1} = \neg z_1^m \neg V \neg R \vee z_1^m V \neg R$$

Решение задачи 8.2.

Запуск RS-триггера зависит от старого и нового содержания.

z^m	z^{m+1}	S	R	Описание
0	0	0	d	Запоминание и возврат в неопр. состояние
0	1	1	0	Установка
1	0	0	1	Возврат в исходное состояние
1	1	d	0	Запоминание или установка

Таблица последовательности состояний 8-5 должна быть изменена соответствующим образом. Выходные функции остаются неизменными.

		r^m		
$S_1 R_1$	$S_0 R_0$	$S_1 R_1$	$S_0 R_0$	
0d	10	0d	0d	$\left. \begin{array}{c} \left. \begin{array}{c} z_0^m \\ z_1^m \end{array} \right\} \right\}$
10	d0	0d	01	
d0	01	01	01	
01	0d	01	0d	

r^m				
D_1	D_0	D_1	D_0	
0	1	0	0	$\left. \begin{array}{c} \left. \begin{array}{c} z_0^m \\ z_1^m \end{array} \right\} \right\}$
1	1	0	0	
1	0	0	0	
0	0	0	0	

Функции выборки RS-триггеров, которые описывают логическую схему SN, считывают из диаграммы Карно-Вейча:

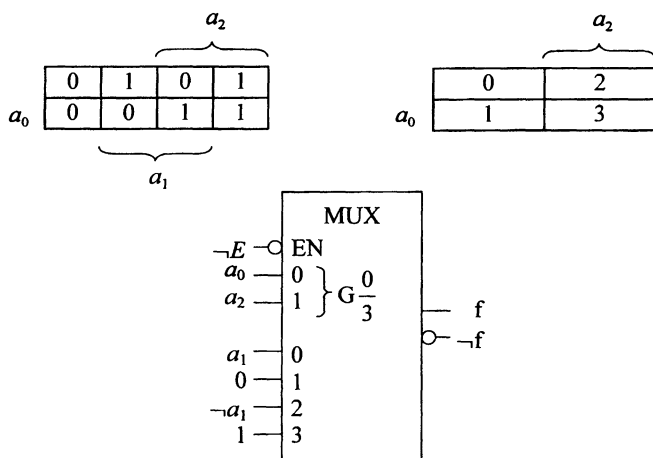
$$\begin{aligned} S_0 &= J_0 = \neg r^m \neg z_1^m; & S_1 &= J_1 = \neg r^m \neg z_0^m \\ R_0 &= K_0 = r^m \vee z_1^m = \neg(\neg r^m \neg z_1^m) = \neg S_0; \\ R_1 &= K_1 = r^m \vee \neg z_0^m = \neg(\neg r^m z_0^m) \end{aligned}$$

Итак, функции выборки для входов S_0, S_1, R_0, R_1 идентичны со случаем реализации JK-триггеров. Для функции запуска D -триггеров получаем:

$$D_0 = J_0 = \neg r^m \neg z_1^m; \quad D_1 = J_1 = \neg r^m z_0^m$$

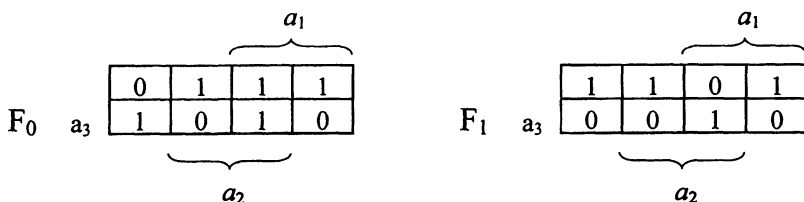
Итак, при реализации на основе D -триггеров получаем самую простую схему. В иных случаях может быть по-другому. Функции запуска для выходов (SN2) при всех реализациях одинаковы.

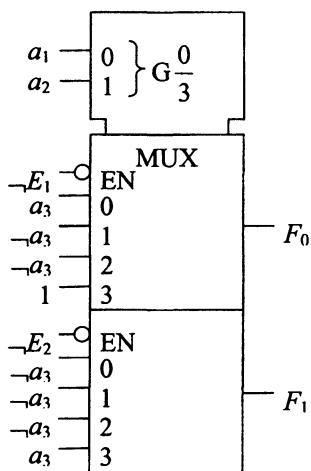
Решение задачи 9.1. Решение для случая, когда на входы мультиплексора поданы a_0 с весом 2^0 и a_2 с весом 2^1 . Возможны и другие решения.



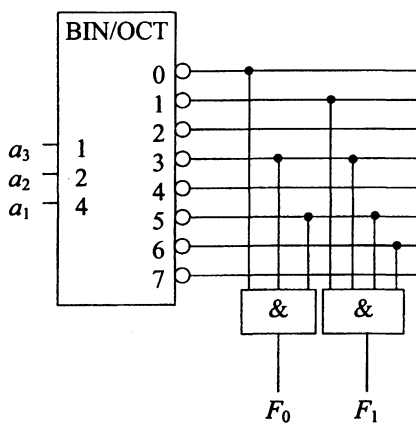
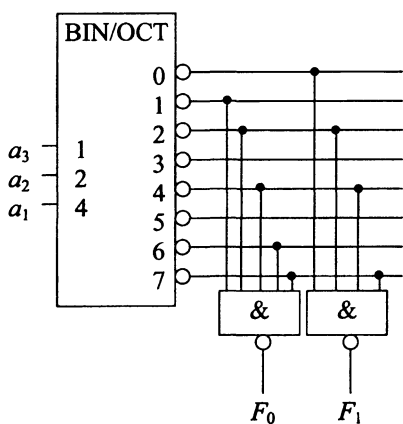
Решение задачи 9.2.

а) реализация на основе мультиплексоров.

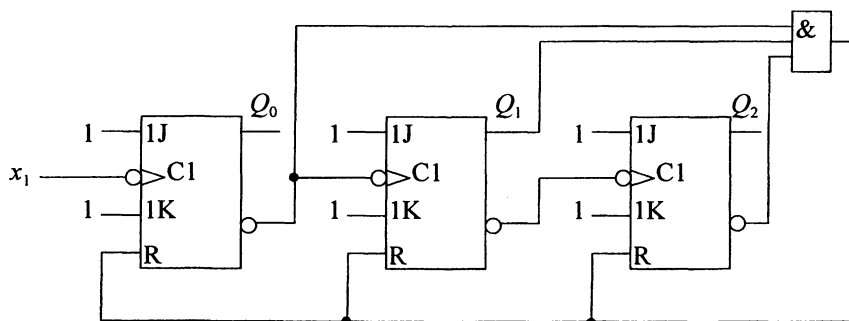




б) две различных реализации с помощью декодера:



Решение задачи 10.1.





Решение задачи 10.2.

Сначала должна быть разработана таблица последовательности состояний на основе заданного кода:

z_2^m	z_1^m	z_0^m	$V = 1$			$V = 0$		
			z_2^{m+1}	z_1^{m+1}	z_0^{m+1}	z_2^{m+1}	z_1^{m+1}	z_0^{m+1}
0	0	0	0	0	1	1	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	0	1	0
1	0	0	0	0	0	0	1	1

Затем формируем диаграмму Карно-Вейча для функций запуска трех D -триггеров:

$$D_2 = z_2^{m+1}$$

 V

$$D_1 = z_1^{m+1}$$

 V

z_1^m	1	0	0	0
	0	d	d	0
	0	d	d	1
	0	d	d	0
		z_2^m		

z_1^m	0	1	0	0
	0	d	d	1
	1	d	d	0
	0	d	d	1
		z_2^m		

$$D_0 = z_0^{m+1}$$

 V

z_1^m	0	1	0	1
	0	d	d	0
	0	d	d	0
	1	d	d	1
		z_2^m		

Уравнения запуска:

$$D_0 = z_0^{m+1} = (z_2^m \neg V) \vee (z_1^m \neg z_0^m) \vee (V \neg z_2^m \neg z_0^m)$$

$$D_1 = z_1^{m+1} = (z_2^m \neg V) \vee (z_0^m \neg z_1^m V) \vee (z_1^m z_0^m \neg V) \vee (z_1^m \neg z_0^m V)$$

$$D_2 = z_2^{m+1} = (\neg z_2^m \neg z_1^m \neg z_0^m \neg V) \vee (z_1^m z_0^m V)$$

Решение задачи 11.1. Вначале формируется таблица последовательности состояний. Начинают с внесения желаемой последовательности сверху вниз в столбцы Q_3^m . Это гарантирует выдвижение последовательности из последовательного выхода. Затем могут быть заполнены столбцы Q_1^m и Q_2^m , внося элементы из столбца Q_3^m по диагонали налево вверх. Из этого получаем автоматически последовательные состояния Q_1^{m+1} , Q_2^{m+1} и Q_3^{m+1} . Не требующиеся состояния 111 и 000 могут быть вначале произвольными.

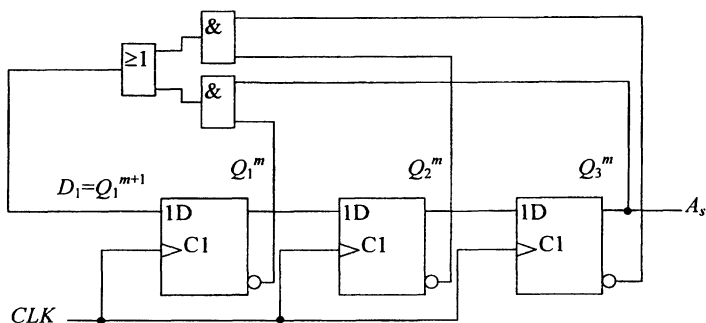
Q_1^m	Q_2^m	Q_3^m	Q_1^{m+1}	Q_2^{m+1}	Q_3^{m+1}
0	1	0	0	0	1
0	0	1	1	0	0
1	0	0	1	1	0
1	1	0	0	1	1
0	1	1	1	0	1
1	0	1	0	1	0
1	1	1	0	1	1
0	0	0	1	0	0

Затем может быть составлена диаграмма Карно-Вейча для входа первого триггера.

$$Q_1^{m+1} = D_1$$

	Q_3^m			
	d		1	1
Q_1^m	1		d	
	Q_2^m			

$$D_1 = Q_3^m \neg Q_1^m \vee \neg Q_3^m \neg Q_2^m$$



Решение задачи 12.1. Модуль 74181 может быть использован как компаратор, если он включен как вычитатель. Для этого мы должны иметь $S = (0, 1, 1, 0)$, $M = 0$ и $c_0 = 1$. Как легко можно установить, теперь для переноса C_4 и выходного сигнала $A_{x=y}$ при разности $x - y$ будет справедлива:

$x = y$	$c_4 = 1, A_{x=y} = 1$
$x > y$	$c_4 = 1, A_{x=y} = 0$
$x < y$	$c_4 = 0, A_{x=y} = 0$

Решение задачи 12.2. Для $S = 0110$ и $M = 1$ получаем в соответствии с уравнением (12.26):

$$t_i = \neg(\neg x_i y_i \vee x_i \neg y_i) = \neg(x_i \leftrightarrow y_i).$$

Поскольку $M = 1$, имеем для всех $u_i = 1$ и получаем в соответствии с уравнением (12.31) функцию:

$$\neg F_i = \neg(x_i \leftrightarrow y_i).$$

Решение задачи 13.1.

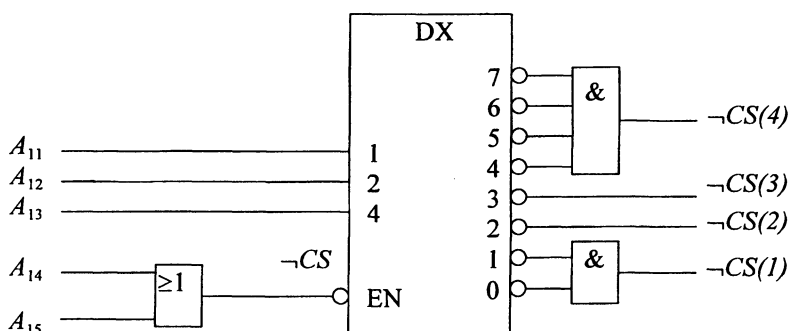
а) Сначала должен быть составлен адресный план. Для этого зоны памяти блоков ЗУ должны быть без пробелов выставлены в ряд одна за другой:

	(Hex)	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
1	0000	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
(4K)	0FFF	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1
2	1000	0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0
(2K)	17FF	0 0 0 1	0 1 1 1	1 1 1 1	1 1 1 1
3	1800	0 0 0 1	1 0 0 0	0 0 0 0	0 0 0 0
(2K)	1FFF	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1
4	2000	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0
(8K)	3FFF	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1

б) В адресном плане зоны адресных линий, которые подходят к ЗУ как входы, отмечены прямоугольниками из штриховых линий.

Декодер должен содержать в качестве старшей входной линии старшую адресную линию, в которой один бит изменен. Это A_{13} . С помощью этого обеспечивается возможность выбора старших ячеек ЗУ. В качестве младшей входной линии в декодере должна использоваться адресная линия, которая не подходит к самому маленькому ЗУ. Это A_{11} .

в) Суммарная зона ЗУ с емкостью 16 К байт разделяется декодером на 8 блоков по 2 К байт. Поэтому для подсоединения к блоку ЗУ с емкостью 8 К байт 4 выхода декодера должны быть связана логической функцией «ИЛИ». Из-за двойной инверсии должен быть использован «И»-вентиль. Результатом этого является следующая схема соединений:

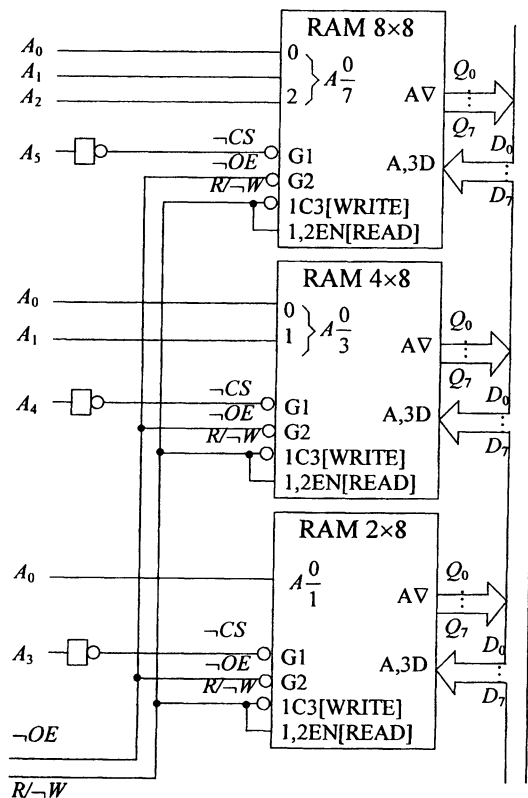


Решение задачи 13.2.

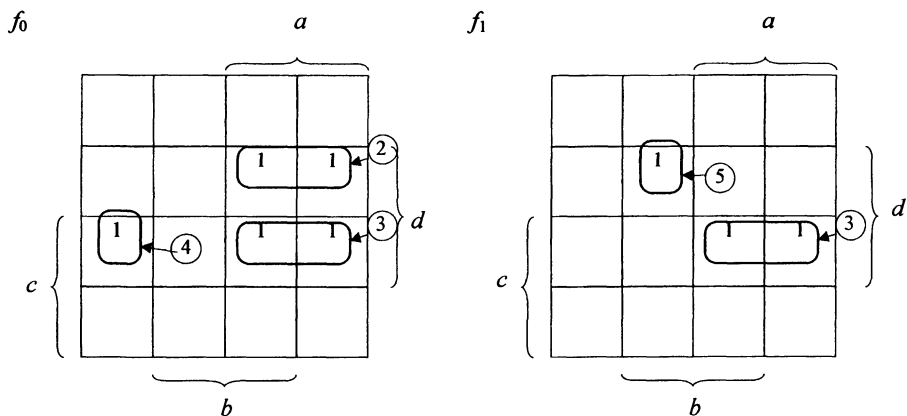
а) самый большой блок ЗУ использует адресные линии от A_0 до A_2 . Исходя из этого в нашем распоряжении для выбора блоков могут быть использованы адресные линии от A_3 до A_7 (см. рисунок на следующей стр.).

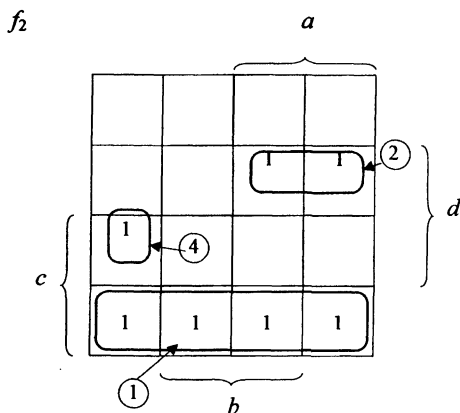
б) Адресный план см. справа (адреса вне данных зон ведут к ошибкам!).

Блок	Адрес (шестнадцатеричный)	Адресные линии (двоичные)							
		7	6	5	4	3	2	1	0
1 (2 бита)	0 8	0	0	0	0	1	0	0	0
	0 9	0	0	0	0	1	0	0	1
2 (4 бита)	1 0	0	0	0	1	0	0	0	0
	1 3	0	0	0	1	0	0	1	1
3 (8 бит)	2 0	0	0	1	0	0	0	0	0
	2 7	0	0	1	0	0	1	1	1



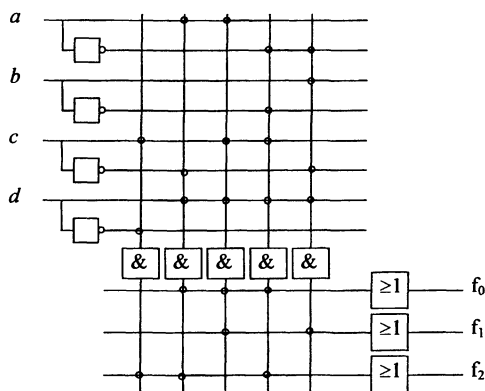
Решение задачи 14.1. Три булевых функции вносятся в 3 диаграммы Карно-Вейча. Затем проводится оптимизирование, направленное на то, чтобы образовалось до 5 термов логического произведения. КВ-диаграммы трех функций имеют вид:





На основе этого получаем 5 термов логического произведения:

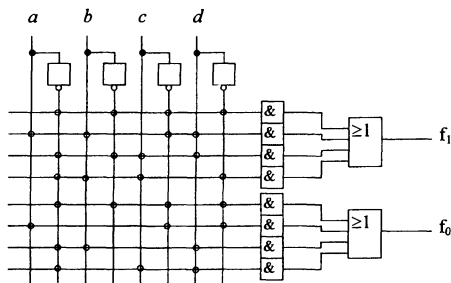
$$P_1 = c_1 \neg d, P_2 = ad \neg c, P_3 = adc, P_4 = \neg a \neg b \neg cd, P_5 = \neg ab \neg cd.$$



Решение задачи 14.2. Функция f_0 формируется таким образом, чтобы ее можно было реализовать с помощью 4 термов логического произведения. Для f_1 это уже имеет место:

$$f_0(a, b, c, d) = \neg a \neg b \neg c \neg d \vee a \neg c \neg d \vee \neg abd \vee \neg acd,$$

$$f_1(a, b, c, d) = \neg a \neg b \neg c \neg d \vee ab \neg cd \vee \neg a \neg bcd \vee \neg abc \neg d$$



Решение задачи 15.1. Последовательно будут проходиться состояния $Z0, Z0, Z1, Z1, Z2, Z2, Z0, Z1, Z0$.

Решение задачи 16.1. Преимуществом совместного запоминания данных и команд в том же самом ЗУ является возможность гибкого разделения зоны памяти. Благодаря этому необходимо, как правило, меньше ЗУ. Недостатком совместного содержания в ЗУ данных и программ является то, что команды и данные должны транспортироваться по одной и той же шине данных. Эта последовательная эксплуатация замедляет работу процессора. Поэтому в качестве альтернативы в некоторых процессорах применяют отдельные ЗУ для данных и команд (т.н. «Гарвардская архитектура»).

Решение задачи 16.2. Если имеем для управляющих переменных $A_L = 1, I_m = 1$, два выхода типа tri-state, работающие на шину данных, будут активны. При определенных обстоятельствах это может иметь следствием разрушение модуля. Поэтому программист микропрограмм должен обладать точными знаниями аппаратных средств, чтобы не вызвать необратимых повреждений блока.

Решение задачи 17.1.

```
MOV B,M    ; (B) ← ((H)(L))
LDAX D     ; (A) ← ((D)(E))
MOV M,A    ; ((H)(L) ← (A))
MOV A,B    ; (A) ← (B)
STAX D     ; ((D)(E)) ← (A)
```

При тактовой частоте 2 МГц получаем время выполнения:

$$32 \cdot 0,5 \text{ мкс} = 16 \text{ мкс}.$$

Решение задачи 17.2. Пусть, например, 6 байт из адреса 1000H сдвигаются в 1200H.

```
MVI    B, 6H      ; сдвиг 6 байт
LXI     H, 1000H   ; определение источника
LXI     D, 1200H   ; определение цели
LOOP    MOV    A,M    ; (A) ← ((H)(L))
        STAX   D      ; ((D)(E)) ← (A)
        INX    D      ; (D)(E) ← (D)(E)+1
        INX    H      ; (H)(L) ← (H)(L)+1
```



```

DCR    B           ; (B) ← (B)-1
JNZ    LOOP

```

Решение задачи 17.3. Подпрограмма «Quer» может выглядеть следующим образом:

```

QUER:  PUSH B       ; сохранение B
        MVI B, 08H   ; число шагов
        MVI C, 00H   ; число единиц
ANF:    RAR          ; сдвиг младшего бита (LSB) на перенос
        JNC MK       ; пропуск, если carry=0
        INR C        ; повышение значения счетчика на 1,
                     ; если carry=1
MK:     DCR B        ; снижение значения счетчика
        JNC ANF      ; при необходимости проверка
                     ; следующего бита
        MOV A, C     ; сумма цифр числа после A
        POP B        ; получение B
        RET          ; обратный переход

```

Литература

- [1] Алексеенко А.Г. Основы микрохемотехники. М. Физматиздат, 2002 г.
- [2] Преснухин Л.Н., Воробьев И.В., Шишкевич А.А. Расчет элементов цифровых устройств: Высшая школа, 1991 г.
- [3] Н.Н. Букреев, Б.М. Максуров, В.И. Горячев. Микроэлектронные схемы цифровых устройств, Сов. радио, 1973 г., Высшая школа, 1987 г.
- [4] Дж. Уейкерли. Проектирование цифровых устройств, Постмаркет, М., 2002 г.
- [5] Г.Г. Казеннов, В.Я. Кремлев, Полупроводниковые интегральные микросхемы, Высшая школа, 1997 г.
- [6] А. Н. Кармазинский. Синтез принципиальных схем цифровых элементов на МДП-транзисторах, Радио и связь, 1983 г.
- [7] М. Шур. Современные приборы на арсениде галлия, Мир, 1991 г.
- [8] Ю. Пожела. Физика быстродействующих транзисторов, Вильнюс, Молкас, 1989 г.
- [9] Н.А. Аваев, В.Н. Дулин, Ю.Е. Наумов. Большие интегральные схемы с инжекционным питанием, Советское радио, 1977 г.
- [10] В.Я. Кремлев, В.И. Мошкин. Функционально-интегрированные элементы БИС. Электронная промышленность, 1976 г.
- [11] С.М. Зи. Физика полупроводниковых приборов, м. Энергия, 1973 г.
- [12] Е.П. Угрюмов. Цифровая схемотехника. СПб.: БВХ-Петербург, 2002 г.

Список технических терминов

2-позиционные двоичные функции

7-сегментный декодер

ABEL

Индикация зависимости

Суммирование (addition)

Суммирование чисел с фиксированной запятой

Целочисленное суммирование

Адресный буфер

Адресная шина

Адресный декодер

Адресация «Direct»

Адресация «Immediate»

Адресация «Register Indirect»

Адресация «Register»

Время доступа адреса

Накапливающий регистр (accumulator)

ALU (АЛУ)

Амплитудное условие

Аналоговая система

Аналоговый демультиплексор

Аналоговый мультиплексор

Аналоговый переключатель

Характеристика выборки

Таблица выборки

Antifuse (программирование без пережигаемых перемычек)

Заявки на книги присылайте по адресу:
125319 Москва, а/я 594
Издательство «Техносфера»
e-mail: knigi@technosphaera.ru
sales@technosphaera.ru
факс: (095) 956 33 46

В заявке обязательно указывайте
свой почтовый адрес!

Подробная информация о книгах на сайтах
<http://www.technosphaera.ru>
<http://www.benran.ru>

К. Фрике
Вводный курс цифровой электроники

Компьютерная верстка — С.А. Кулешов
Дизайн книжных серий — С.Ю. Биричев
Ответственный за выпуск — Л.Ф. Соловейчик

Формат 70 x 100/16. Печать офсетная.
Гарнитура Computer modern LaTeX.
Печ.л. 27. Тираж 5000 экз. Зак. № 8505.
Бумага офсет №1, плотность 65 г/м².

Издательство «Техносфера»
Москва, ул. Тверская, дом 10 строение 3

Отпечатано в ППП «Типография «Наука»
Академиздатцентра «Наука» РАН,
121099 Москва, Шубинский пер., 6