Параллельные регистры

Лабораторная работа 4: Параллельные регистры

Цель работы: изучить виды и состав регистров; овладеть методами синтеза регистровых структур на базе современных типов триггеров и регистров; приобрести навыки в создании, отладке и экспериментальном исследовании регистровых структур в среде САПР БИС «Ковчег 3.04».

4.1. Теоретические сведения

4.2.1. Определения, классификация и основные понятия

Регистр — операционный узел ЭВМ, представляющий собой регулярную совокупность элементов памяти и комбинационных схем, предназначенный для выполнения микроопераций приёма, хранения, преобразования и выдачи цифрового кода, а также простейших поразрядных операций, операций над цифровым кодом и выработки осведомительного сигнала о состоянии регистра. Регистры являются распространённым типом последовательностных узлов в современных цифровых устройствах. В качестве элементов памяти в регистрах используются либо статические элементы, например триггеры различного типа, либо динамические элементы, например МДП-, ПЗС-структуры и т.п.

Комбинационные схемы строятся на основе какого-либо логического базиса. Если регистры выполняются в виде интегральной схемы, то элементы памяти и комбинационные схемы выполняются по единой технологии (ТТЛ, ТТЛШ, КМОП, ЭСЛ, $И^2Л$ и др.).

Приём — занесение новых данных в регистр (иногда называется вводом, записью).

Данные — это числа, команды, управляющие коды и т.п. Разновидностью приёма является операция ***установки начального состояния***, выполняемая с помощью одного управляющего сигнала.

Хранение — режим работы регистра, при котором данные не изменяются. В отличие от оперативных, постоянных и долговременных запоминающих устройств, в регистрах осуществляется кратковременное хранение (запоминание) данных, например на период выполнения одного или нескольких тактов (циклов) работы всего устройства.

Преобразование цифрового кода — одна из микроопераций (операций) вида:

- сдвиг данных на один или несколько разрядов;
- преобразование параллельного кода в последовательный и обратно;
- организация задержки во времени;
- выполнение поразрядных операций НЕ, И, ИЛИ, сумма по модулю 2, эквивалентность и др.;
- преобразование кодов (например прямого кода числа со знаком в дополнительный код и т.д.);
- умножение и деление двоичного полинома на двоичный полином;
- генерация различных кодов и т.д.

Схемы конкретных регистров могут выполнять лишь некоторые из указанных операций. В общем случае под преобразованием понимаются не только перечисленные операции, но и управление процессом с помощью последовательности сигналов в результате выполнения соответствующих операций над кодами.

Выдача цифрового кода — вывод данных из регистра для передачи в другие узлы цифрового устройства (иногда называется операцией считывания или просто считыванием).

Разрядность *n* регистра, как правило, совпадает (или определяется) с разрядностью обрабатываемого цифрового кода.

Состояние регистра — конкретное значение цифрового кода, зафиксированного в *n* элементах памяти регистра.

Осведомительным называется сигнал, вырабатывающий соотношения типа «меньше», «равно», «больше» над состоянием регистра; сигнал проверки знака числа, на нуль и др.

Регистры классифицируются по различным признакам [30].

По способам приёма и выдачи данных регистры подразделяются на параллельные (статические или регистры памяти), последовательные, параллельно-последовательные, последовательно-параллельные, комбинированные (универсальные, многофункциональные).

В параллельных регистрах приём и выдача кодов производятся по всем разрядам одновременно, их основная функция — хранение слова. В них же могут осуществляться и поразрядные операции над словами.

В последовательных регистрах слова принимаются и выдаются последовательно разряд за разрядом, их называют сдвигающими (сдвиговыми), поскольку тактирующие сигналы перемещают слово в разрядной сетке.

По направлению передач или сдвига данных регистры подразделяют на:

- однонаправленные, когда сдвиг данных осуществляется только влево или только вправо, сдвиг данных от старших разрядов к младшим называется правым сдвигом или сдвигом вправо; сдвиг данных от младших разрядов к старшим называется левым сдвигом или сдвигом влево и реверсивным, когда сдвиг данных может осуществляться в любую сторону (в зависимости от управляющего сигнала).

В параллельно-последовательных регистрах приём данных осуществляется в параллельном коде, а выдача данных — в последовательном коде. В последовательно-параллельных регистрах приём данных осуществляется в последовательном коде, а выдача — в параллельном коде. В комбинированных регистрах приём и выдача данных осуществляются с различными сочетаниями перечисленных выше способов.

По количеству цепей передачи одного разряда данных регистры подразделяют на однофазные и парафазные.

Под передачей данных понимаются приём, выдача и сдвиг кода.

В однофазных регистрах передача осуществляется только по одной цепи в каждом разряде (или в прямом, или в обратном коде). В парафазных регистрах передача осуществляется по двум цепям в каждом разряде (и в прямом, и в обратном коде).

При сравнительной оценке однофазных и парафазных регистров необходимо отметить, что однофазные являются более эффективными при выполнении на ИС, так как они содержат в два раза меньше входов и выходов данных, чем парафазные регистры. Это позволяет сократить число выводов ИС, что особенно важно при проектировании на БИС.

По используемой системе синхронизации регистры подразделяют на однотактные и многотактные.

Однотактные регистры управляются одной последовательностью синхронизирующих сигналов, а многотактные — несколькими. Однотактные регистры строятся на триггерах, имеющих в своей структуре две элементарные запоминающие ячейки, синхронизируемых перепадом (фронтом, спадом), или на так называемых триггерах с внутренней задержкой срабатывания. Типичный пример таких триггеров — универсальные *D*- и *JK*-триггеры, *MS*-структуры с квазиоднофазной синхронизацией. Многотактные регистры строятся на триггерах, имеющих в своей структуре одну элементарную запоминающую ячейку, синхронизируемых потенциальным сигналом; на динамических МДП-инверторах, на структурах, способных хранить данные в различных физических средах (ПЗС-структуры, магнитные домены и т.п.).

По типу запоминающих элементов, используемых в регистрах для хранения данных, регистры подразделяют на статические и динамические.

Статические регистры строятся на триггерах, выполненных на логических элементах, и сохраняют данные до тех пор, пока действует источник питающего напряжения. Статические регистры имеют ограничение по максимальной частоте синхросигналов.

Динамические регистры строятся на динамических МДП-структурах, в которых данные сохраняются в виде заряда на конденсаторах затвор-исток МДП-транзисторов, в твёрдом теле (ПЗС-структуры) и т.п. Динамические регистры имеют ограничение и по максимальной, и по минимальной частоте синхросигналов. В данном учебном пособии рассматриваются только статические регистры.

По числу источников, от которых поступают данные, регистры подразделяют на регистры с одним направлением данных и регистры с многими направлениями данных.

По влиянию входных данных на выход регистры подразделяют на прозрачные и непрозрачные.

Прозрачными называются регистры, обладающие свойством при активном уровне синхросигнала адекватно, в соответствии с типом используемых триггеров отслеживать на выходе все переключения входных данных. Прозрачными являются регистры хранения на триггерах, имеющих в своей структуре одну элементарную запоминающую ячейку, синхронизируемых уровнем сигнала (например на триггерах-защёлках). Прозрачные регистры не должны применяться в режиме сдвига, поскольку пока действует активный синхросигнал «сдвиг», передача данных из разряда в разряд будет проходить безостановочно и триггеры могут переключаться многократно, тогда как требуется сдвиг на один разряд.

Непрозрачными называются регистры, обладающие свойством при любом уровне синхросигнала не передавать на выход переключения входных данных. Свойство непрозрачности — очень важное свойство, именно оно позволяет использовать режим сдвига при однофазной синхронизации.

По схемотехнике выходных цепей регистры подразделяют на регистры со стандартным (на два состояния) выходом (ТТЛ, КМДП, ЭСЛ и т.п.), регистры, выходные цепи которых выполнены по схеме с открытым коллектором, эмиттером или стоком (ОК, ОЭ, ОС), регистры, выходные цепи которых выполнены с тремя состояниями выхода.

Схемотехника выполнения выходных цепей регистров существенно влияет на структуру схем, обеспечивающих обмен данными между регистровыми узлами цифровых устройств.

Приведём ряд примеров различного функционального назначения регистров: регистр хранения, регистр сдвига, регистр-аккумулятор, регистр общего назначения, регистр данных, регистр адреса, регистр состояния, регистры — порты ввода и вывода, регистр-шлюз, регистр — генератор кодовых последовательностей, регистры-счётчики, регистры последовательного приближения, регистровая память типа «очередь», регистровая память типа «стек» и т.д.

Важнейшими параметрами регистров являются разрядность, статические параметры и характеристики, динамические (временные и частотные) параметры.

Наращивание разрядности регистров достигается добавлением необходимого числа соответствующих триггеров, тактовые входы которых присоединяют к сигналам синхронизации, а информационные входы и выходы триггеров используют по назначению.

Статические параметры и характеристики регистра — это уровни «0» и «1», пороговое входное напряжение $U_{\text{пор.вх}}$, входные и выходные токи ($I^0_{\text{вх}}$, $I^1_{\text{вх}}$, $I^0_{\text{вых}}$, $I^1_{\text{вых}}$), коэффициент разветвления по выходу $K_{\text{раз}}$ и т.д., т.е. обычные статические параметры и характеристики ИС.

Динамические параметры определяются для конкретного типа регистра, например минимальная длительность тактового сигнала, время опережения фронта информационного входа относительно фронта тактового сигнала и др.

4.2.2. Организация цепей приёма

Организация и структура цепей приёма данных в параллельных регистрах определяются следующими факторами:

- формой представления входных данных (однофазная, парафазная);
- типом триггера, применяемого в регистре;
- системой синхронизации и управления;
- специальными требованиями, такими как маскирование части вводимого слова, наличие общих цепей сброса и установки, выполнение поразрядных логических операций над вводимым словом и словом, хранящимся в регистре.

Рассмотрим несколько примеров формирования цепей приёма данных в параллельных регистрах.

Пример 4.1. Пусть входные данные x_i представлены в однофазной форме, в качестве триггера в регистре используется асинхронный *RS*-триггер в базисе И-НЕ. Определить структуру регистра.

Характеристическая таблица асинхронного *RS*-триггера в базисе И-НЕ представлена в разделе 3 в табл. 3.5. Функционирование цепей приёма в этом случае приведено в табл. 4.1, где индекс *i* характеризует произвольный разряд параллельного регистра, Q_i^t — содержимое триггера до приёма новых данных, Q_i^{t+1} — содержимое триггера после приёма новых данных, R_i^* и S_i^* — функции возбуждения информационных входов триггера, используемого при построении регистра.

Таблица 4.1. Функционирование цепей приёма в параллельных регистрах (к примеру 4.1)

Номер набора	x_i	Q_i^t	Q_{i+1}^t	R_i^*	S_i^*
0	0	0	0	×	1
1	0	1	0	0	1
2	1	0	1	1	0
3	1	1	1	1	×

На рис. 4.1, *а* представлена эталонная карта Карно. Выражения для R_i^* и S_i^* получим из карт Карно (рис. 4.1, *б* и *в* соответственно):

$$R_i^* = x_i\,, \tag{4.1}$$

$$S_i^* = \overline{x_i}\,. \tag{4.2}$$

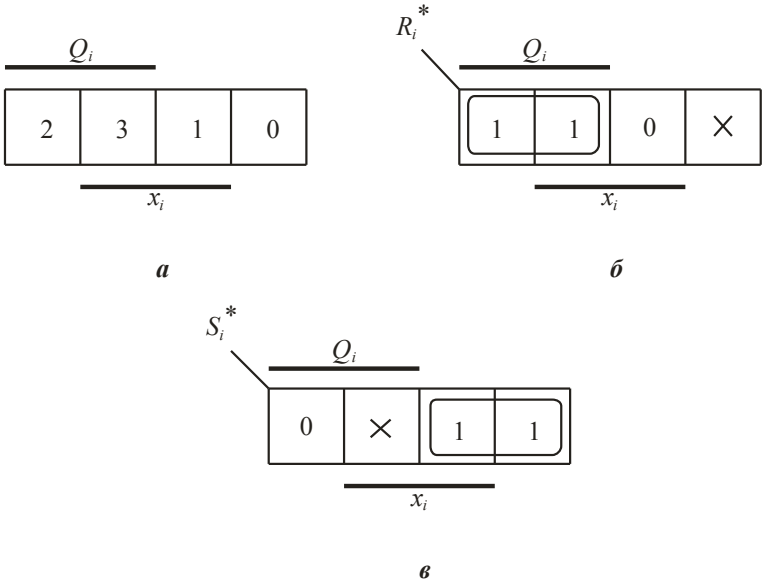
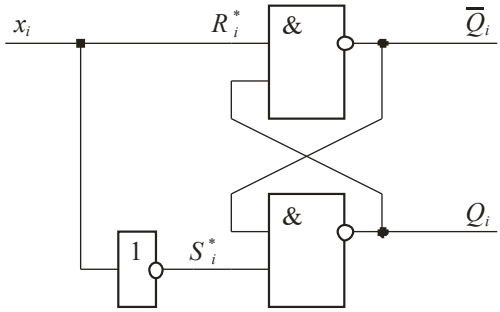


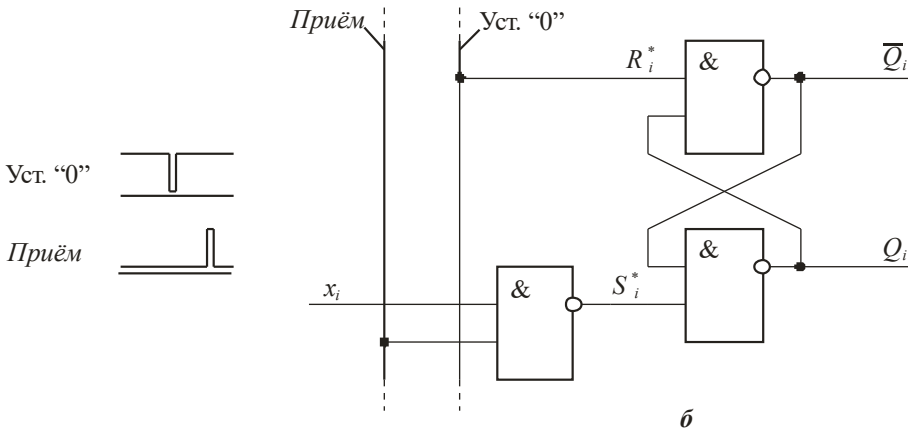
Рис. 4.1. Карты Карно: *а* — эталонная; *б* — для R_i^* ; *в* — для S_i^*

Схема, соответствующая выражениям (4.1) и (4.2), представлена на рис. 4.2, *а*. Следует обратить внимание, что если входные данные заданы в парафазной форме, то инвертор *D*1.1 не потребуется. Полученная схема не имеет режима хранения, поэтому применяется либо для формирования сигналов с крутыми фронтами, либо в качестве буферного усилителя. В последнем случае достаточно было

бы двух инверторов. Режим хранения можно обеспечить, если реализовать схему, представленную на рис. 4.2, *б*. Сначала подаётся сигнал «Уст. 0», а затем сигнал «Приём». Если $x_i = 0$, то после прихода сигнала «Приём» триггер остается в нулевом состоянии; если $x_i = 1$, то он установится в единичное состояние.



а



б

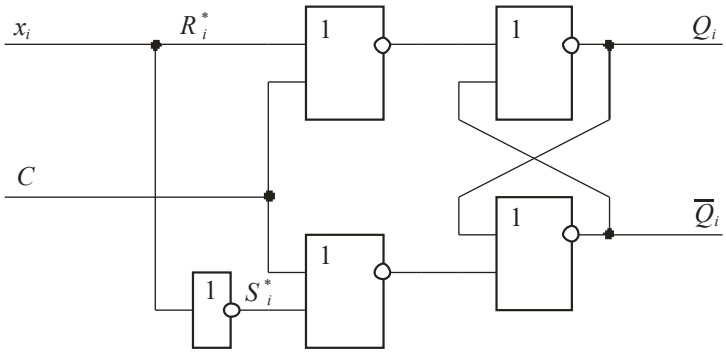


Рис. 4.2. Организация цепей приёма: *а* — для асинхронного *RS*-триггера; *б* — то же с двухтактным управлением; *в* — для синхронного *RS*-триггера

Перечислим недостатки схемы, показанной на рис. 4.2, *б*: схема требует для приёма данных двух тактов (недостаток существенный), сигналы управления имеют противоположные активные уровни (недостаток несущественный).

Приём данных за один такт можно осуществить, если использовать синхронный *RS*-триггер. Схема одного разряда регистра в этом случае имеет вид, представленный на рис. 4.2, *в*, на котором синхронный *RS*-триггер реализован в базисе ИЛИ-НЕ. Входные данные x_i записываются в триггер активным уровнем «0» на *C*-входе и сохраняются при действии пассивного уровня «1» на *C*-входе.

Пример 4.2. Пусть входные данные x_i представлены в однофазной форме, а в качестве триггеров в регистре используются *D*-триггеры, тактируемые потенциалом «1». Характеристическая таблица *D*-триггера представлена в табл. 1.26 (она совпадает с таблицей *D*-триггера, тактируемого перепадом). Функционирование цепей приёма в этом случае приведено в табл. 4.2. Из рис. 4.3 следует

$$D_i^* = x_i. \tag{4.3}$$

Таблица 4.2. Функционирование цепей приёма в параллельных регистрах (к примеру 4.2)

Номер набора	x_i	Q_i^t	Q_{i+1}^t	D_i^*
0	0	0	0	0
1	0	1	0	0
2	1	0	1	1
3	1	1	1	1

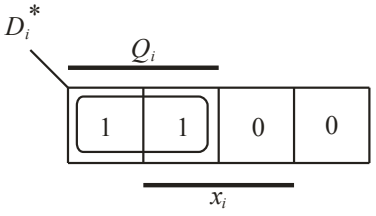


Рис. 4.3. Карта Карно для D_i^*

На рис. 4.4, *а* приведена схема 4-разрядного регистра. На схеме входные данные обозначаются индексированными буквами D_i (*data)*, причём 2^i — двоичный вес соответствующего разряда регистра. Иногда индекс в явном виде отражает двоичный вес: *D*1, *D*2, *D*4, *D*8 и т.д. Если регистр построен на триггерах-защёлках, то его называют «регистр-защёлка».

Условным изображением регистра, представленным на рис. 4.4, *б*, пользуются тогда, когда на схеме необходимо показать каждый вход и выход данных. Если же данные рассматриваются как единое понятие — ***шина данных***, то используют обозначение, показанное на рис. 4.4, *в*.

Простота организации цепей приёма для *D*-триггеров обеспечила их широкое использование для построения параллельных регистров. Организацию цепей приёма для *JK*-триггеров определите самостоятельно.

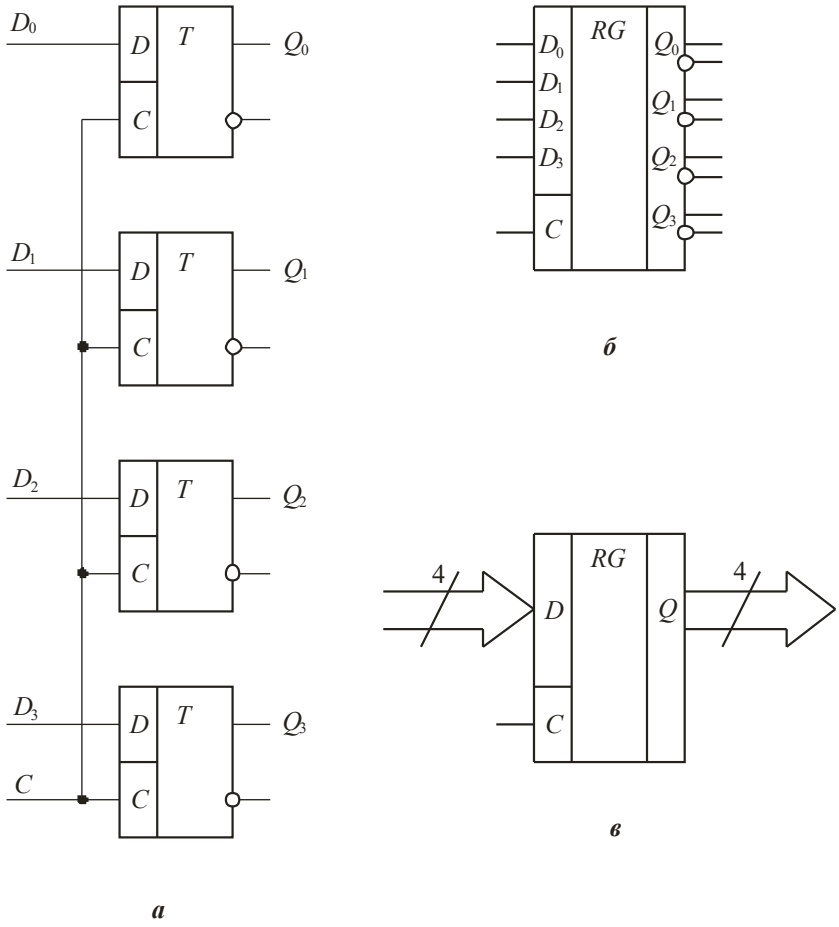


Рис. 4.4. Регистр на *D*-триггерах: *а* — схема; *б*, *в* — условные обозначения

4.2.3. Организация цепей приёма с маскированием

Маскированием называется операция приёма данных в часть разрядов регистра, в другую часть автоматически заносятся нули (или единицы). ***Маска*** — любой набор битов, обеспечивающий требуемое маскирование.

Определим структуру цепей приёма с маскированием при следующей формулировке задачи: каждый разряд регистра имеет информационный вход x_i и вход маскирования m_i, причём при $m_i = 0$ в соответствующий разряд записывается «0», а при $m_i = 1$ записывается x_i.

Пример 4.3. Пусть входные данные представлены в однофазной форме, в качестве триггеров в регистре используются *D*-триггеры. Функционирование цепей приёма в этом случае приведено в табл. 4.3. Из рис. 4.5, *б* следует:

$$D_i^* = m_i x_i \,. \tag{4.4}$$

На рис. 4.6 приведена схема одного разряда регистра на *D*-триггерах с маскированием входных данных.

Пример 4.4. То же на *JK*-триггерах.

Функционирование цепей приёма в этом случае приведено в табл. 4.4. Характеристическая таблица *JK*-триггера представлена в табл. 1.20. Из рис. 4.7, *а*,*б* следует:

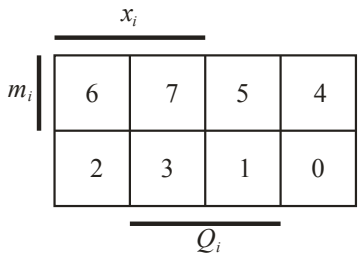
$$J_i^* = m_i x_i \,, \tag{4.5}$$

$$K_i^* = \overline{m_i} + \overline{x_i} = \overline{m_i x_i} \,. \tag{4.6}$$

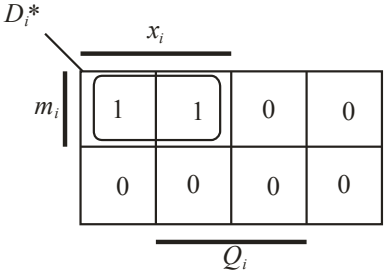
На рис. 4.8 приведена соответствующая схема.

Таблица 4.3. Функционирование цепей приёма в параллельных регистрах с маскированием входных данных (к примеру 4.3)

Номер набора	m_i	x_i	Q_i^t	Q_i^{t+1} D_i^*
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1



а



б

Рис. 4.5. Карты Карно: *а* — эталонная; *б* — для D_i^*

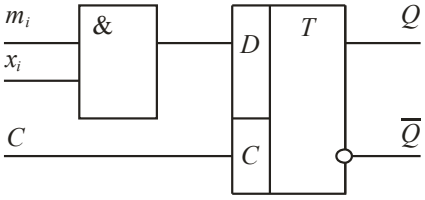
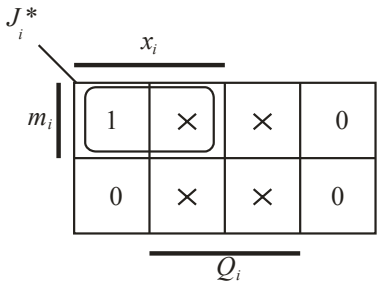


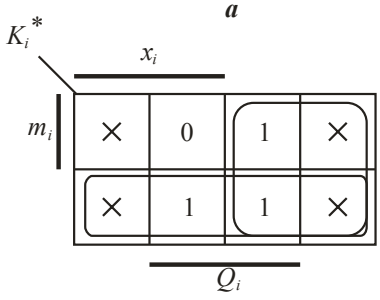
Рис. 4.6. Схема цепи приёма с маскированием для одного разряда регистра на *D*-триггерах

Таблица 4.4. Функционирование цепей приёма в параллельных регистрах с маскированием входных данных (к примеру 4.4)

Номер набора	m_i	x_i	Q_i^t	Q_i^{t+1}	J_i^*	K_i^*
0	0	0	0	0	0	×
1	0	0	1	0	×	1
2	0	1	0	0	0	×
3	0	1	1	0	×	1
4	1	0	0	0	0	×
5	1	0	1	0	×	1
6	1	1	0	1	1	×
7	1	1	1	1	×	0



а



б

Рис. 4.7. Карты Карно: *а* — для J_i^*; *б* — для K_i^*

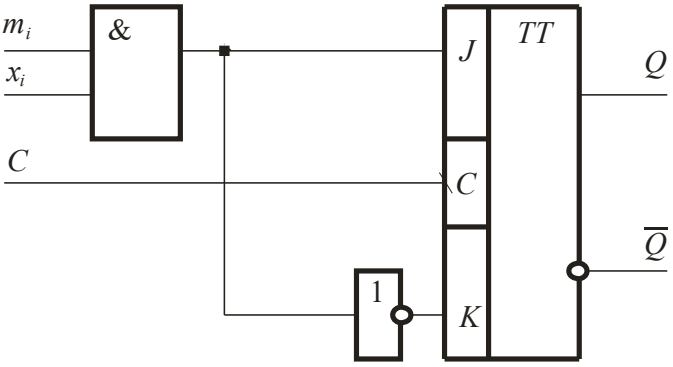


Рис. 4.8. Схема цепи приёма с маскированием для одного разряда регистра на *JK*-триггерах

4.2.4. Формирование в регистре общих цепей сброса и установки

Все регистры, кроме информационных и тактовых входов, в подавляющем большинстве случаев дополняются общими для всех разрядов входами сброса (*R*) и установки (*S*). Эти входы устанавливают соответствующие состояния всех разрядов регистра независимо от сигналов, поступающих на информационные и тактовые входы во время действия сигналов *R* и *S*.

Следовательно, сигналы *R* и *S* имеют максимальный приоритет, часто *R*- и *S*-входы называют ***асинхронными***. По окончании асинхронного сигнала установленное им состояние сохраняется вплоть до очередного перепада или уровня синхросигнала *C*.

Важно подчеркнуть, что асинхронными входами *R* и *S* можно снабдить только непрозрачные триггеры. Прозрачный *D*-триггер, например, не сможет сохранить установленное *R*-входом состояние, если *R*-сигнал окончился во время действия активного уровня синхросигнала, поскольку из-за прозрачности на её выходе тут же установится уровень *D*-входа.

Организацию асинхронных *R*- и *S*-входов рассмотрим для синхронного *RS*-триггера, выполненного в базисе ИЛИ-НЕ. Характеристическое уравнение для него имеет вид (вывод его сделайте самостоятельно)

$$Q^{t+1} = \overline{S^t}\,\overline{C^{t+1}} + R^t Q^t + C^{t+1} Q^t . \qquad (4.7)$$

Из уравнения (4.7) следует, что $Q^{t+1} = 0$ (установку в 0) можно обеспечить в двух случаях: 1) $Q^t = 0$ и $C = 1$; 2) $Q^t = 0$ и $S = 1$. $Q^t = 0$ обеспечивается добавлением *R*-входа к *RS*-триггеру, образованному элементами *D*1.3 и *D*1.4, а $C = S = 1$ обеспечивается добавлением третьего входа к элементу *D*1.2. Соединенные вместе добавочные входы образуют общий для схемы *R*-вход. На рис. 4.9 он показан жирной линией.

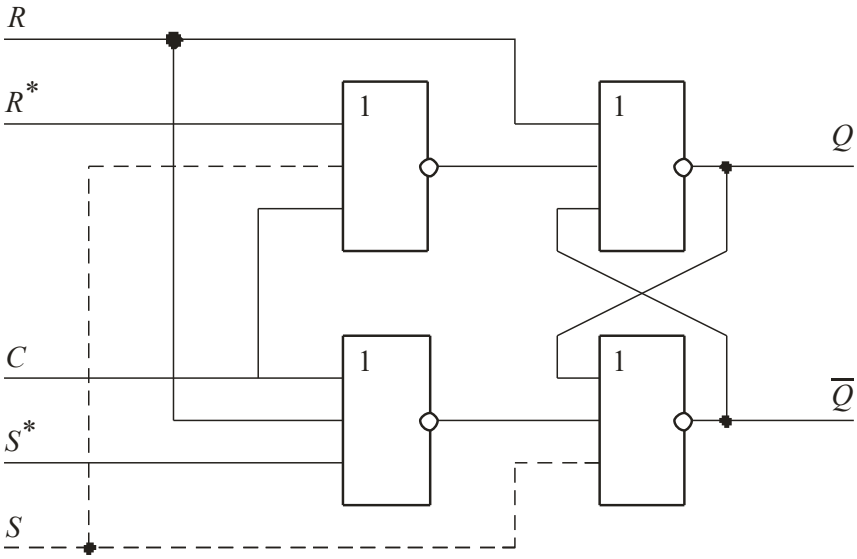


Рис. 4.9. Тактируемый *RS*-триггер в базисе ИЛИ-НЕ с асинхронными входами установки и сброса

Из уравнения (4.7) также следует, что $Q^{t+1}=1$ (установку в 1) можно обеспечить в двух случаях: 1) $Q^t=1$ и $C=1$; 2) $Q^t=1$ и $R=1$. $Q^t=1$ обеспечивается добавлением *S*-входа к *RS*-триггеру, образованному элементами *D*1.3 и *D*1.4, а $C=S=1$ обеспечивается добавлением третьего входа к элементу *D*1.1. Соединенные вместе добавочные входы образуют общий для схемы *S*-вход. На рис. 4.9 он показан штриховой линией. Последний анализ можно было бы и не проводить, а воспользоваться симметрией схемы и входов *R* и *S*.

Следует обратить внимание, что активные уровни на *R*- и *S*-входах — это уровни «1», а на R^*-, S^*- и *C*-входах — уровни «0»!

Характеристическое уравнение для схемы, представленной на рис. 4.10, имеет вид

$$Q^{t+1}=Q_2^{t+1}=\overline{C^{t+1}}Q_1^t+C^{t+1}Q_2^t. \qquad (4.8)$$

Кроме того, для Q_1^{t+1} можно записать

$$Q_1^{t+1}=S^tC^{t+1}+\overline{R^t}Q_1^t+\overline{C^{t+1}}Q_1^t. \qquad (4.9)$$

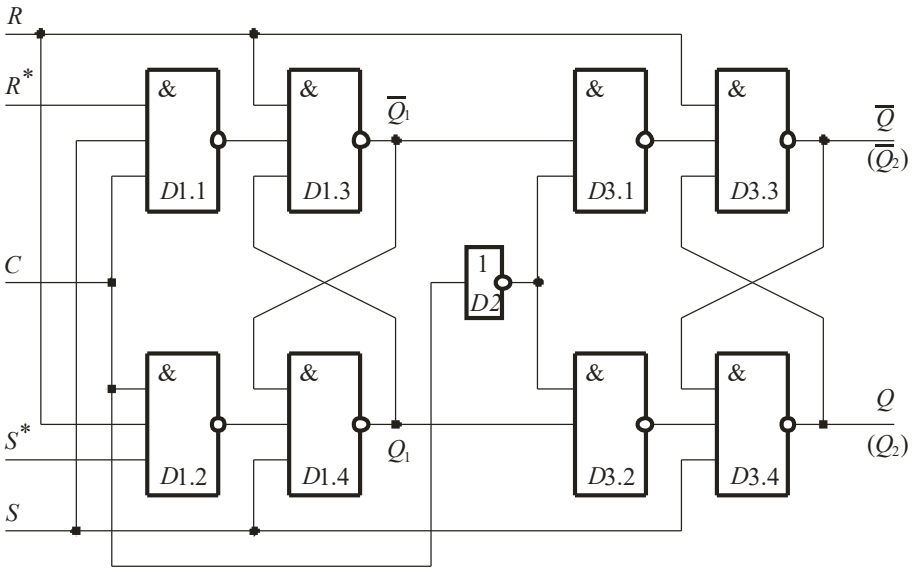


Рис. 4.10. Схема с асинхронными входами установки *S* и сброса *R*

Вывод уравнения (4.8) сделайте самостоятельно.

Из уравнения (4.8) следует, что для обнуления $Q^{t+1} = Q_2^{t+1}$ при любом значении сигнала *C* достаточно обнулить оба элементарных триггера *D*1.3–*D*1.4 и *D*3.3–*D*3.4. Кроме того, из (4.9) следует, что надо обеспечить *SC* = 0 при обнулении элементарного триггера *D*1.3–*D*1.4. Подробный анализ опущен, так как он аналогичен проведенному выше анализу для схемы синхронного триггера, реализованного в базисе ИЛИ-НЕ. Схема с асинхронными входами установки *S* и сброса *R* приведена на рис. 4.10.

В заключение отметим следующее. Из рис. 4.9 и 4.10 видно, что *R*- и *S*-входы даже для отдельных триггеров не являются единичными нагрузками. Это надо учитывать при разработке многоразрядных регистров. Если регистр выполняется в виде ИС, СИС или БИС, то для установочного входа и входа сброса используют на кристалле специальный буферный усилитель, обеспечивающий получение единичной нагрузки по входам *R* и *S* регистра.

4.2.5. Особенности реализации цепей приёма

Многообразие вариантов использования регистров в цифровых схемах приводит в ряде случаев к необходимости разработки специальной реализации цепей приёма. Рассмотрим несколько примеров таких случаев.

1. Если регистр принимает данные от большого числа источников, то входные цепи регистра снабжаются мультиплексором. На рис. 4.11 приведен пример приёма данных от восьми источников.

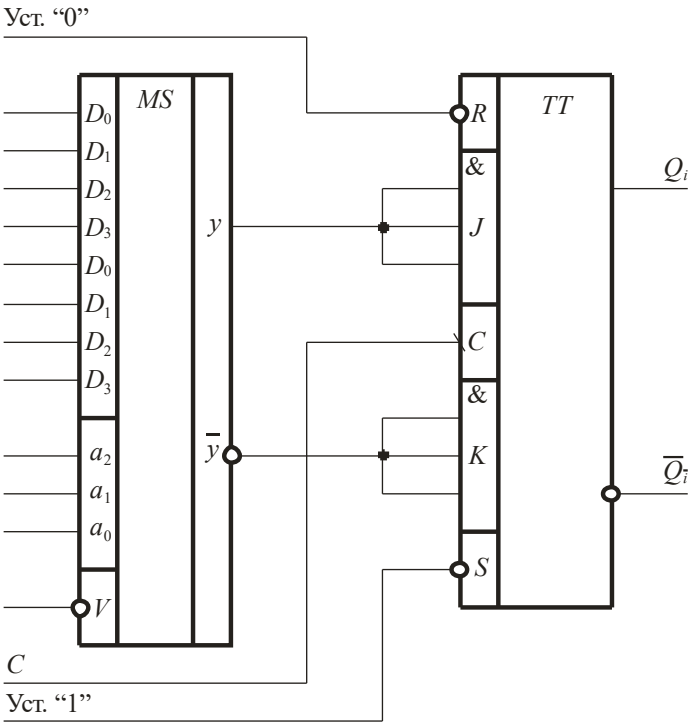


Рис. 4.11. Схема приёма данных в регистр от восьми источников

2. Если регистр выполняется на дискретных триггерах, каждый из которых имеет входы «Уст. 0» и «Уст. 1», то последние можно использовать для ввода данных, а информационные входы использовать для других целей, например для организации цепей сдвига.

3. Некоторые универсальные триггеры снабжаются специальным входом «Запрет записи» или «Разрешение записи». Запись в такой регистр можно осуществить только при наличии разрешающего сигнала на соответствующем входе. В схеме, показанной на рис. 4.11, вход *V* мультиплексора может выполнять функции входа разрешения записи.

4. В некоторых регистрах имеется специальный вход управления полярностью сигнала записи. Так, например, микросхема 564ТМ3 состоит из четырёх одноступенчатых *D*-триггеров типа «защёлка», имеющих общий вход синхронизации *C*, вход управления полярностью *P* и независимые информационные входы *D*0–*D*3. Запись информации осуществляется при определённых сигналах на входе синхронизации *C* и входе управления полярностью *P*. При подаче на вход управления *P* низкого (высокого) уровня напряжения запись данных производится при наличии на входе синхронизации *C* также низкого (высокого) уровня напряжения.

Для целей управления уровнем сигнала записи или его активным перепадом широко используются элементы «сумма по модулю 2»: $y = v\overline{x} + \overline{v}x$, так как при $v = 0$ $y = x$, а при $v = 1$ $y = \overline{x}$.

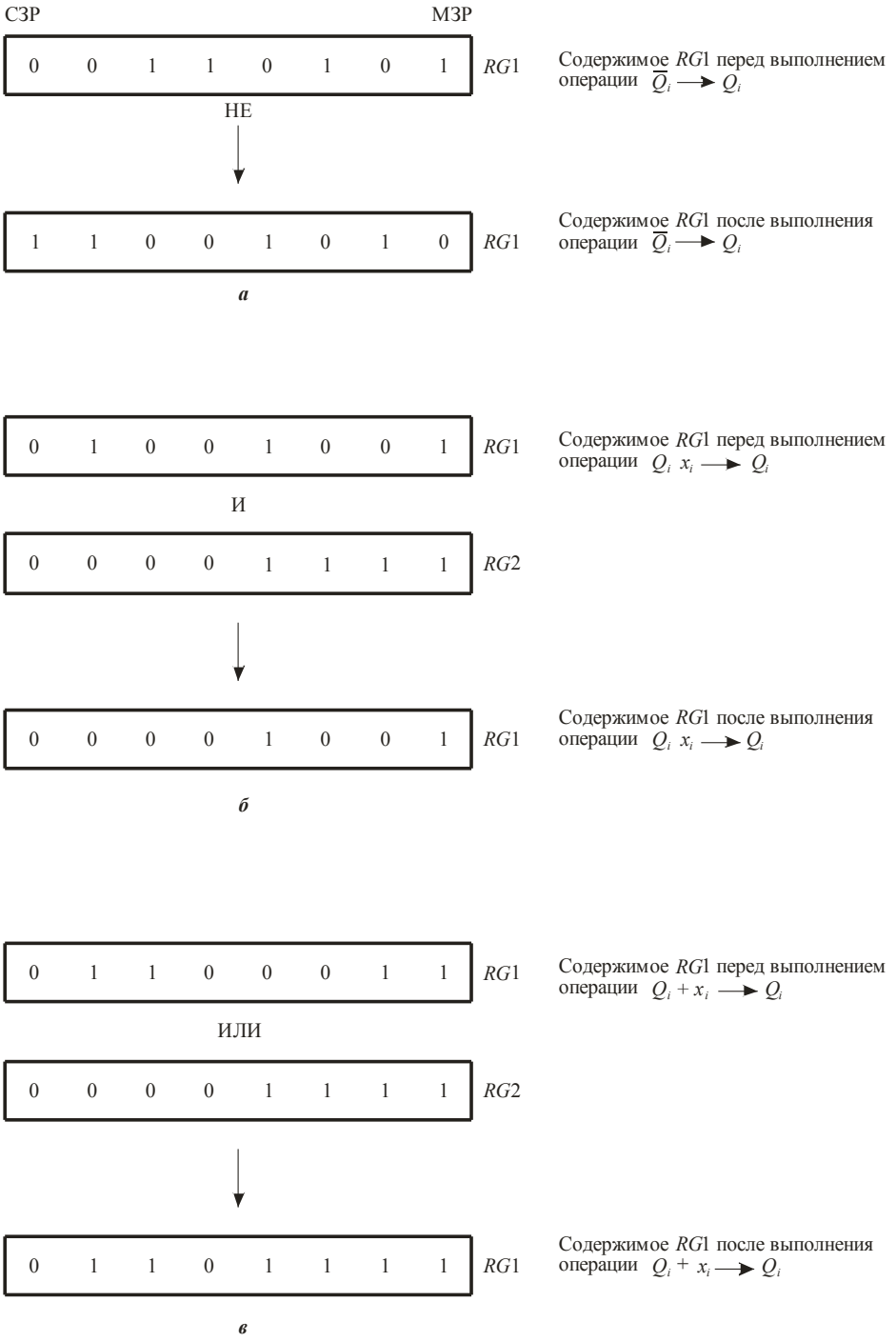


Рис. 4.12. Выполнение поразрядных операций над содержимым *RG*1 и *RG*2:
а — НЕ; *б* — И; *в* — ИЛИ

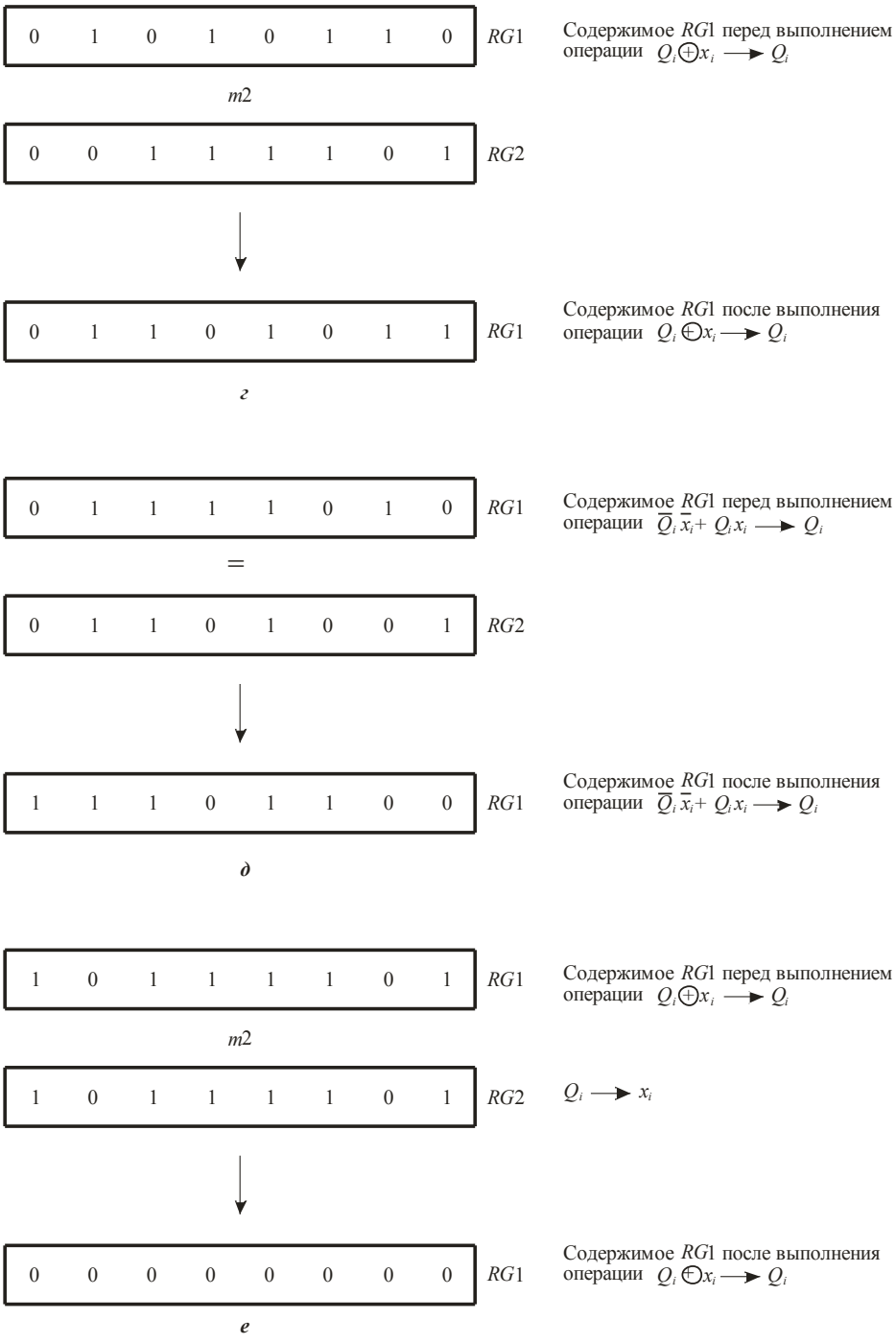


Рис. 4.12. Продолжение. Выполнение поразрядных операций над содержимым *RG*1 и *RG*2: *г* — m2; *д* — равнозначность; *е* — применение операции m2 для очистки *RG*1

4.2.6. Реализация в регистрах поразрядных логических операций

При обработке данных часто требуется выполнение поразрядных логических операций НЕ, И, ИЛИ, исключающее ИЛИ, равнозначности и др. На практике для реализации логических операций служит арифметико-логическое устройство микропроцессора. Логические операции выполняются всегда над содержимым регистра-аккумулятора и каким-то другим словом из другого регистра или из памяти. По окончании логической операции результат загружается в *RG*-аккумулятор. Это означает, что исходное содержимое *RG*-аккумулятора теряется. Рассмотрим в чистом виде организацию цепей приёма регистра, обеспечивающего реализацию указанных выше поразрядных операций. Предполагаем, что операции выполняются над содержимым двух регистров *RG*1 и *RG*2, причём результат сохраняется в *RG*1. Для выполнения любых операций каждый из регистров должен быть выполнен на базе двухступенчатых синхронных триггеров (докажите это положение самостоятельно). Выполнение поразрядных операций над содержимым *RG*1 и *RG*2 иллюстрируется на рис. 4.12, где СЗР — старший значащий разряд; МЗР — младший значащий разряд; Q_i — i-й разряд регистра *RG*1; x_i — i-й разряд регистра *RG*2.

Пример 4.5. Разработать цепи приёма данных x_i для регистра, выполненного на универсальных *D*-триггерах, осуществляющего приём внешних данных при управляющем сигнале $a = 0$ и инвертирование содержимого регистра Q_i при $a = 1$. Операция выполняется при действии активного сигнала на тактовом входе триггера. Функционирование цепей приёма в этом случае приведено в табл. 4.5.

Таблица 4.5. Реализация двух поразрядных операций (к примеру 4.5)

Номер набора	a	x_i	Q_i^t	Q_i^{t+1} D_i^*
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

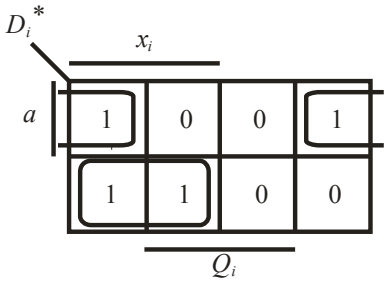


Рис. 4.13. Карта Карно для D_i^*

Из рис. 4.13 следует:

$$D^* = \bar{a}x_i + a\overline{Q_i} \,. \tag{4.10}$$

На рис. 4.14 приведена соответствующая схема. Поразрядная операция НЕ формирует обратный код числа.

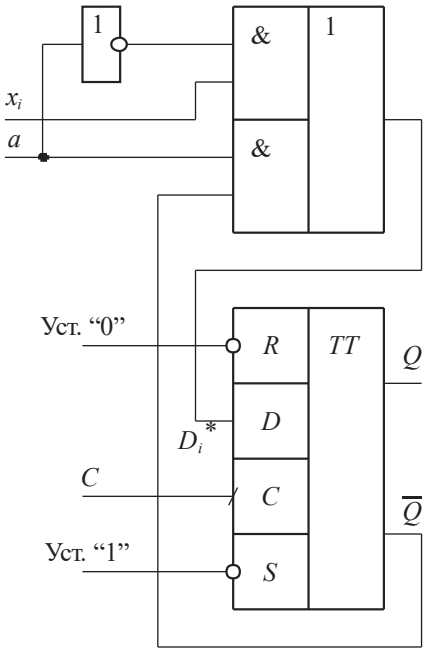


Рис. 4.14. Схема цепей приёма и поразрядного инвертирования

Пример 4.6. Разработать параллельный регистр на *JK*-триггерах, выполняющий две микрооперации:1) $Q_i^{t+1} = x_i^t$ (приём внешних данных в прямом коде); 2) $Q_i^{t+1} = x_i^t Q_i^t$ (поразрядная логическая операция И над входными данными и содержимым регистра).

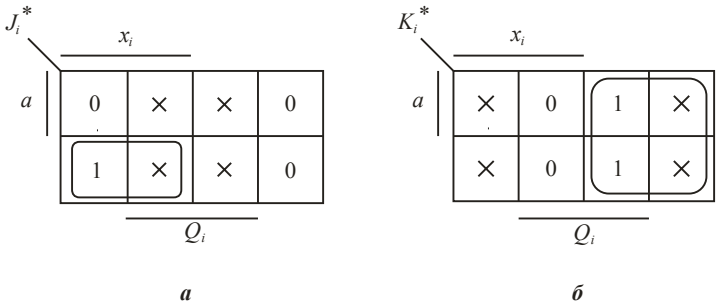
Функционирование такого регистра представлено в табл. 4.6, где a — код микрооперации (пусть при $a = 0$ выполняется первая микрооперация, а при $a = 1$ — вторая). Из рис. 4.15, *а* и *б* следует:

$$J^* = \bar{a}x_i; \tag{4.11}$$

$$K^* = \overline{x_i} \,. \tag{4.12}$$

Таблица 4.6. Реализация двух поразрядных операций (к примеру 4.6)

Номер набора	a	x_i	Q_i^t	Q_i^{t+1}	J_i^*	K_i^*
0	0	0	0	0	0	×
1	0	0	1	0	×	1
2	0	1	0	1	1	×
3	0	1	1	1	×	0
4	1	0	0	0	0	×
5	1	0	1	0	×	1
6	1	1	0	0	0	×
7	1	1	1	1	×	0



а ***б***

Рис. 4.15. Карты Карно: *а* — для J_i^*; *б* — для K_i^*

На рис. 4.16 приведена соответствующая схема. В данной схеме используется *JK*-триггер, активным сигналом которого на входе *C* является переход 01, а на входах *R* и *S* — уровень «1».

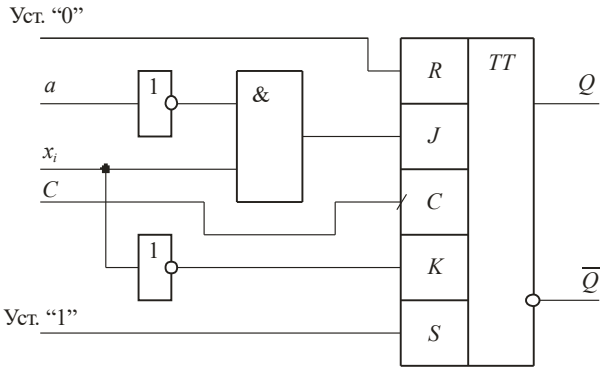


Рис. 4.16. Схема записи и поразрядной операции И

Пример 4.7. Разработать параллельный регистр на *JK*-триггерах, выполняющий две микрооперации: 1) $Q_i^{t+1} = x_i^t$ (приём внешних данных в прямом коде); 2) $Q_i^{t+1} = x_i^t + Q_i^t$ (поразрядная логическая операция ИЛИ над входными данными и содержимым регистра).

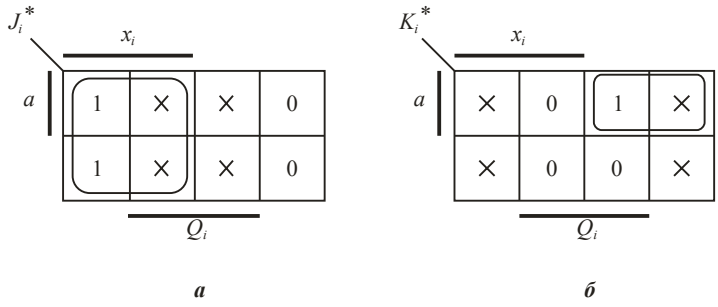
Функционирование такого регистра представлено в табл. 4.7, где a — код микрооперации (пусть при $a = 1$ выполняется первая микрооперация, а при $a = 0$ — вторая). Из рис. 4.17 следует:

$$J^* = x_i; \tag{4.13}$$

$$K^* = a\overline{x_i}. \tag{4.14}$$

Таблица 4.7. Реализация двух поразрядных операций (к примеру 4.7)

Номер набора	a	x_i^t	Q_i^t	Q_i^{t+1}	J_i^*	K_i^*
0	0	0	0	0	0	×
1	0	0	1	1	×	0
2	0	1	0	1	1	×
3	0	1	1	1	×	0
4	1	0	0	0	0	×
5	1	0	1	0	×	1
6	1	1	0	1	1	×
7	1	1	1	1	×	0



а ***б***

Рис. 4.17. Карты Карно: *а* — для J_i^*; *б* — для K_i^*

На рис. 4.18 приведена соответствующая схема. В данной схеме используется *JK*-триггер, активным сигналом которого на входе *C* является переход 10, а на входах *R* и *S* — уровень «0».

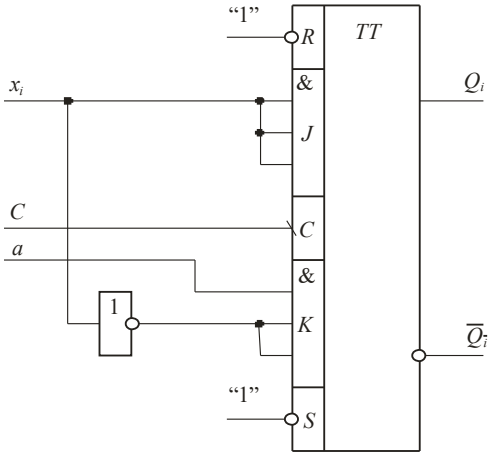


Рис. 4.18. Схема одного разряда параллельного регистра

Пример 4.8. Разработать параллельный регистр на универсальных *D*-триггерах, выполняющий две микрооперации: 1) $Q_i^{t+1} = x_i^t$ (приём внешних данных в прямом коде); 2) $Q_i^{t+1} = x_i^t \oplus Q_i^t$ (поразрядная логическая операция «сумма по модулю 2» над входными данными и содержимым регистра).

Функционирование такого регистра представлено в табл. 4.8, где a — код микрооперации (пусть при $a = 1$ выполняется первая микрооперация, а при $a = 0$ — вторая). Из рис. 4.19 следует:

$$D^* = ax_i + x_i\overline{Q_i} + \overline{a}\overline{x}\ Q_i\,. \tag{4.15}$$

Таблица 4.8. Реализация двух поразрядных операций (к примеру 4.8)

Номер набора	a	x_i	Q_i^t	Q_i^{t+1} D_i^*
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

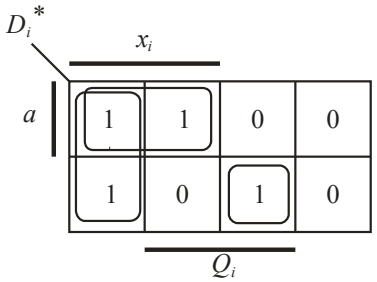


Рис. 4.19. Карта Карно для D_i^*

На рис. 4.20 приведена соответствующая схема.

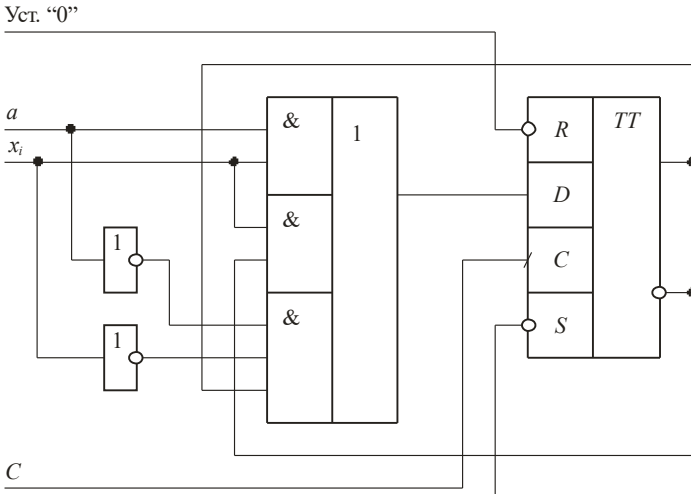


Рис. 4.20. Схема записи и поразрядной операции «сумма по модулю 2»

Если *RG*1 не имеет асинхронного сброса, т.е. нет входа «Уст. 0», то вместо него можно использовать поразрядную операцию «сумма по модулю 2». Для осуществления сброса следует выполнить эту операцию над словом и копией этого слова. Происходящее при этом действие наглядно иллюстрирует рис. 4.12, *е* — результат выполнения операции «сумма по модулю 2» над битом и его копией представляет собой логический 0. Таким образом, при выполнении данной операции над словом и его копией производится сброс или очистка слова.

Пример 4.9. Для реализации поразрядной операции «эквивалентность» (см. рис. 4.12, *д*) достаточно использовать схему, реализующую поразрядную операцию «сумма по модулю 2», а выходное слово снимать не с прямого выхода Q триггера, а с инверсного выхода $\overline{Q}$, так как для двух переменных справедливо соотношение

$$\overline{Q_i}\,\overline{x_i} + Q_i x_i = \overline{Q_i\,\overline{x_i} + \overline{Q_i} x_i} = \overline{Q_i \oplus x_i}\,. \qquad (4.16)$$

Пример 4.10. Используя универсальные *JK*-триггеры, разработать четырёхразрядный параллельный регистр, выполняющий четыре микрооперации:

$$Q_i^{t+1} = x_i\,;$$

$$Q_i^{t+1} = x_i Q_i\,;$$

$$Q_i^{t+1} = x_i + Q_i\,;$$

$$Q_i^{t+1} = x_i \oplus Q_i\,.$$

Так как всего выполняется четыре микрооперации, закодируем их двухразрядным кодом микрооперации $a_1 a_0$ в соответствии с табл. 4.9. Кодирование может быть выполнено и другим способом, в том числе с использованием четырёхразрядного унитарного кода. Функционирование проектируемого регистра приведено в табл. 4.10. Из рис. 4.21 следует:

$$J_i^* = a_1 x_i + \overline{a_0} x_i = (a_1 + \overline{a_0}) x_i = \overline{\overline{a_1} a_0} x_i ;$$

$$K_i^* = a_1 a_0 x_i + \overline{a_1}\,\overline{x_i} . \qquad (4.17)$$

Таблица 4.9. Кодирование четырёх поразрядных микроопераций (к примеру 4.9)

Поразрядные логические операции		
Код операции		Вид операции
a_1	a_0	
0	0	Приём данных
0	1	И
1	0	ИЛИ
1	1	*m*2

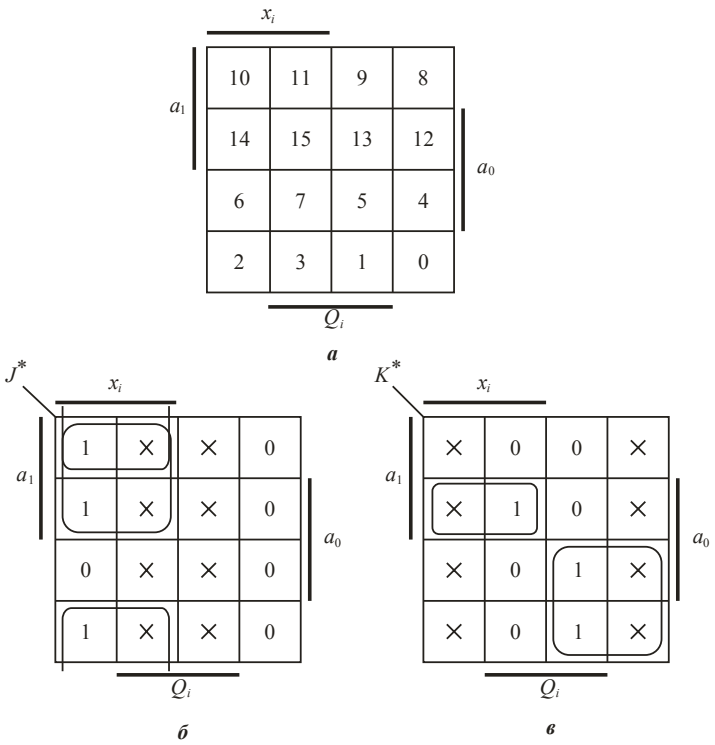


Рис. 4.21. Карты Карно: *а* — эталонная; *б* — для *J**; *в* — для *K**

Таблица 4.10. Реализация четырёх поразрядных операций (к примеру 4.10)

Номер набора	a_1	a_0	x_i	Q_i^t	Q_i^{t+1}	J^*	K^*
0	0	0	0	0	0	0	×
1	0	0	0	1	0	×	1
2	0	0	1	0	1	1	×
3	0	0	1	1	1	×	0
4	0	1	0	0	0	0	×
5	0	1	0	1	0	×	1
6	0	1	1	0	0	0	×
7	0	1	1	1	1	×	0
8	1	0	0	0	0	0	×
9	1	0	0	1	1	×	0
10	1	0	1	0	1	1	×
11	1	0	1	1	1	×	0
12	1	1	0	0	0	0	×
13	1	1	0	1	1	×	0
14	1	1	1	0	1	1	×
15	1	1	1	1	0	×	1

На рис. 4.22 приведена соответствующая схема одного разряда регистра. То, что регистр четырёхразрядный, отражает запись $i = 0...3$. Обратите внимание, что произведение $\overline{a_1}\overline{a_0}x_i$ реализовано логикой на J-входе JK-триггера.

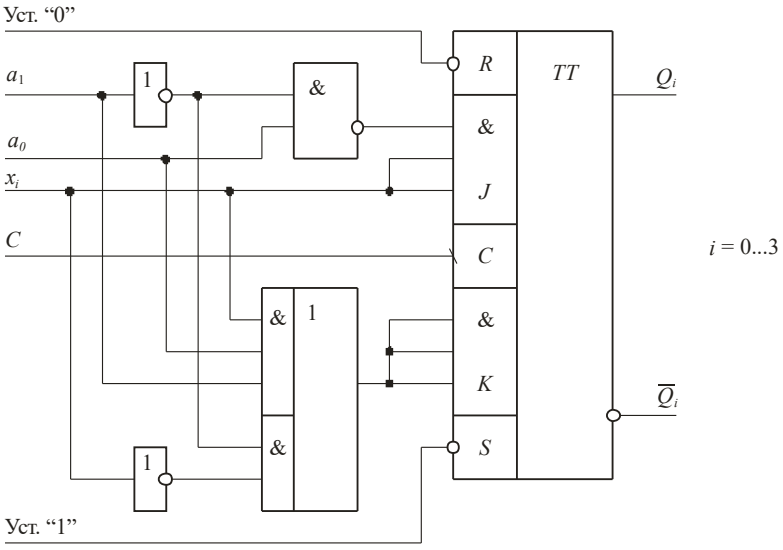


Рис. 4.22. Входные цепи регистра, реализующего четыре операции

4.2.7. Режим хранения

Для организации режима хранения (после ввода данных в регистр) не требуется никаких дополнительных цепей. Режим хранения обеспечивается подачей на тактовые входы всех триггеров, на которых собран регистр, неактивных уровней для одноступенчатых триггеров либо любых уровней для двухступенчатых триггеров, тактируемых переходами (фронтом или спадом). Для триггеров типа К155ТВ1 старой модификации при $C = 1$ реализуется асинхронный режим приёма данных по переходам 10 на *J*- и *K*-входах. Промышленный выпуск этих триггеров прекращен в 1976 г. Режим хранения для этой ИС нужно обеспечивать при $C = 0$. Если регистр имеет вход «Разрешение записи», то этот вход также можно использовать для обеспечения режима хранения.

Если для регистра обеспечен режим хранения, то входные данные не могут изменить его состояние. Выходы же регистра могут быть либо доступны, либо недоступны в зависимости от конкретной реализации цифрового устройства, в котором используются регистры, так как считывание данных с выходов регистра, находящегося в режиме хранения, не изменяет его состояния.

4.2.8. Организация цепей выдачи

Организация и структура цепей выдачи данных в параллельных регистрах определяются следующими факторами:

- формой представления выходных данных (однофазная или парафазная);
- наличием или отсутствием инверсного выхода у триггеров регистра;
- видом выдаваемого кода (прямой или обратный);
- схемотехнической реализацией выходных цепей триггеров регистра (стандартная схема, схема с открытым коллектором (эмиттером, стоком), схема с тремя состояниями выхода);

• наличием управляющего входа «Разрешение выхода»;

• необходимостью выработки осведомительного сигнала о его состоянии или формирования цепей индикации состояния регистра;

• требованием выдачи данных с регистра на многие приёмники;

• требованием маскирования выходных данных.

Рассмотрим несколько примеров формирования цепей выдачи данных в параллельных регистрах. В простейшем случае выходные данные снимаются непосредственно с цепей Q_i и $\overline{Q}_i$ каждого триггера. Ясно, что можно снять данные в прямом и обратном кодах с однофазной или парафазной формой их представления. Недостаток такого решения — нельзя управлять формой представления выходных данных.

Пример 4.11. Разработать выходные цепи регистра, которые под управлением сигнала a выдают данные в прямом коде при $a = 0$ и в обратном коде при $a = 1$. Данные должны выдаваться в однофазной форме. Функционирование выходных цепей в этом случае приведено в табл. 4.11, где Q_i — выход триггера регистра, y_i — выходная цепь, доступная для потребителя. Из рис. 4.23 следует:

$$y_i = \bar{a}Q_i + a\overline{Q_i} = a \oplus Q_i \,. \tag{4.18}$$

Таблица 4.11. Реализация двух поразрядных микроопераций для выходных цепей параллельного регистра (к примеру 4.11)

Номер набора	a	Q_i	y_i
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

На рис. 4.24 приведена соответствующая схема.

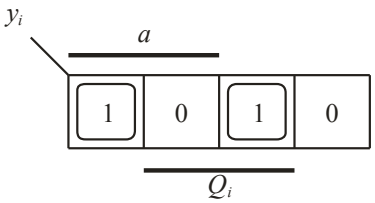


Рис. 4.23. Карта Карно для y_i

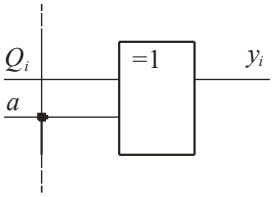


Рис. 4.24. Управляемая выходная цепь регистра

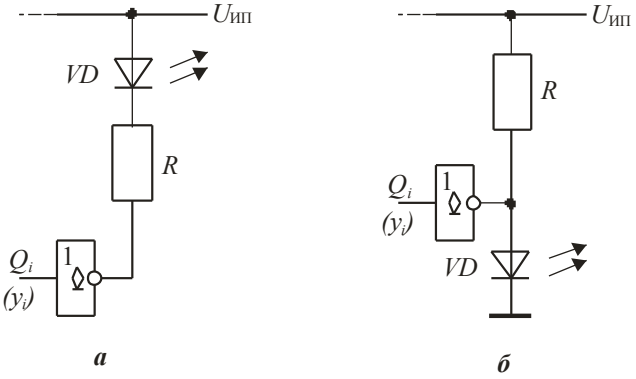


Рис. 4.25. Фрагменты схемы индикации с применением светодиода при включении:
а — от уровня “1”; *б* — от уровня “0”

Отметим также, что управляющий вход «Разрешение выхода» воздействует на выходные буферы регистра. Он устанавливает их в третье состояние или закрывает транзистор в схеме ОК и т.п. Наличие управляющего входа и управляемого буфера на выходе регистра существенно упрощает обмен данными между регистровыми структурам и организацию регистров-файлов.

Цепи индикации состояния регистра могут выполняться с различным уровнем сложности. Простейшие цепи индикации параллельного двоичного кода приведены на рис. 4.25, где предполагается, что инверторы выполнены с открытым коллектором. Для индикации всех состояний многоразрядного регистра применяются знакосинтезирующие индикаторы различных модификаций с соответствующими дешифраторами. Примером сложной системы индикации является так называемая «бегущая строка».

Пример 4.12. Для 4-разрядного параллельного регистра разработать схему, вырабатывающую два осведомительных сигнала: y_1 должен указывать на нулевое состояние регистра, y_2 — на то, что состояние регистра меньше или равно 3 либо больше или равно 12. Для сигналов y_1 и y_2 обеспечить активные уровни «0».

Таблица 4.12. Реализация двух осведомительных сигналов о состоянии 4-разрядного параллельного регистра (к примеру 4.12)

Номер набора	Q_3	Q_2	Q_1	Q_0	y_1	y_2
0	0	0	0	0	0	0
1	0	0	0	1	1	0
2	0	0	1	0	1	0
3	0	0	1	1	1	0
4	0	1	0	0	1	1
5	0	1	0	1	1	1
6	0	1	1	0	1	1
7	0	1	1	1	1	1
8	1	0	0	0	1	1
9	1	0	0	1	1	1
10	1	0	1	0	1	1
11	1	0	1	1	1	1
12	1	1	0	0	1	0
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	1	0

Табл. 4.12 является таблицей истинности для сигналов y_1 и y_2. Для сигнала y_1 удобно записать СКНФ, а затем преобразовать выражение к базису И-НЕ:

$$y_1 = Q_3 + Q_2 + Q_1 + Q_0 = \overline{\overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0}} \,. \qquad (4.19)$$

На рис. 2.26 приведена карта Карно для сигнала y_2, из которой следует:

$$y = Q_3\overline{Q_2} + \overline{Q_3}Q_2 = Q_3 \oplus Q_2 \,. \qquad (4.20)$$

Схема, реализующая осведомительные сигналы y_1 и y_2, приведена на рис. 4.27.

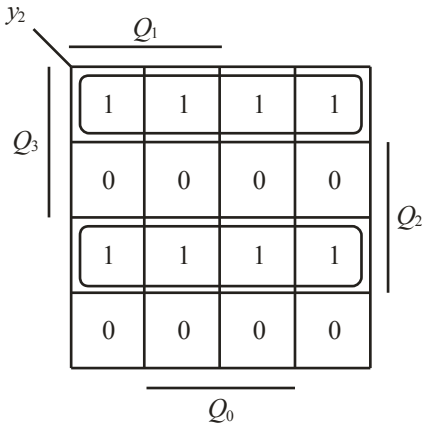


Рис. 4.26. Карта Карно для сигнала y_2

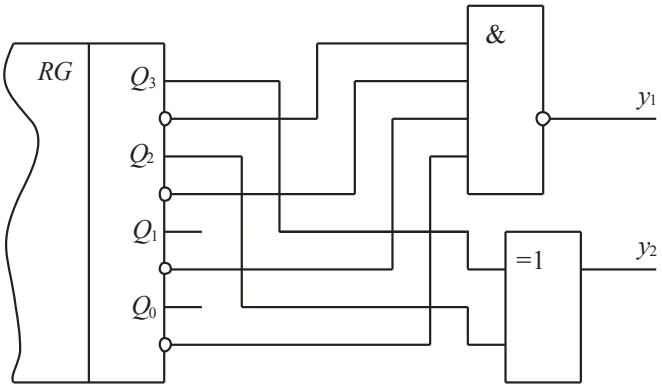


Рис. 4.27. Схема реализации y_1 и y_2

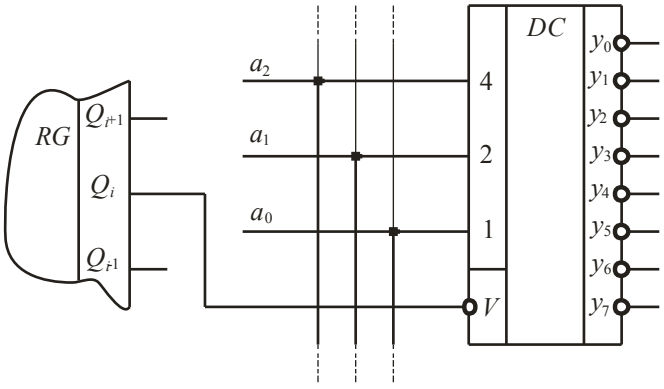


Рис. 4.28. Схема передачи выходных данных
в один из восьми приёмников

Пример 4.13. Разработать схему передачи данных с регистра на один из восьми приёмников. Для решения этой задачи необходимо использовать селектор «1 из 8», в качестве которого можно взять стробируемый дешифратор «1 из 8». Соответствующая схема приведена на рис. 4.28. Поясним её работу. Для произвольного выхода y_j дешифратора можно записать

$$y_j = \overline{\overline{V}\tilde{a}_2\tilde{a}_1\tilde{a}_0}\ , \tag{4.21}$$

где $\tilde{a}_2\tilde{a}_1\tilde{a}_0$ — произвольный адрес приёмника (от 0 до 7).

Пусть подан адрес 5, т.е. $a_2a_1a_0 = 101$, учитывая, что $V = Q_i$, для y_5 можно записать

$$y_5 = \overline{\overline{Q}_i a_2 \overline{a}_1 a_0} = \overline{\overline{Q}_i \cdot 1 \cdot \overline{0} \cdot 1} = Q_i\ . \tag{4.22}$$

На остальных выходах дешифратора при этом адресе будет установлен уровень «1».

Очевидно, что передача данных с регистра на все восемь приёмников одновременно является тривиальной задачей — достаточно соединить параллельно одноименные входные линии всех приёмников и подключить их к соответствующим разрядам регистра.

4.2. Лабораторное задание

Выполнить синтез структур, заданных в индивидуальном задании, построить временные диаграммы работы.

4.2.1. Пример индивидуального задания

1. Используя *D*-триггеры, разработать разряд регистра, выполняющий микрооперации:

- приём данных ($Q_i^{t+1} = x_i^t$);
- логическое «И» ($Q_i^{t+1} = Q_i^t \times x_i^t$);
- «счётный режим» ($Q_i^{t+1} = \overline{Q}_i^t$).

4.2.2. Порядок выполнения работы

Разрабатываемый регистр должен знать, какая операция выполняется в тот или иной момент. Для этого введём дополнительные сигналы *A*1, *A*0, которые однозначно зададут режим работы. Двух сигналов достаточно, чтобы задать четыре операции. В нашем случае операций три, поэтому одна из комбинаций будет не определена. Способ кодирования микроопераций показан в табл. 4.13. В общем случае порядок операций в таблице может быть произвольным.

Таблица 4.13. Кодирование микроопераций

Микрооперация	*A*1	*A*0	Примечание
Хранение данных	0	0	$Q^{t+1} = Q^t$
Логическое «И»	0	1	$Q^{t+1} = X^t \cdot Q^t$
«Счётный режим»	1	0	$Q^{t+1} = \overline{Q^t}$
-	1	1	Не определена

Заполним таблицу переходов одного разряда (табл. 4.14). $A1$, $A0$ — входы, задающие микрооперацию, Q — выход регистра, X — входные данные регистра (все двухместные операции выполняются между X и Q). Левая половина таблицы до двойной черты заполняется простым перебором всех возможных значений. Правая часть заполняется с использованием таблицы микроопераций табл. 4.13. Сначала заполним столбец «Микрооперация», в соответствии с ним заносятся правильные значения в столбец Q^{t+1}, он же будет входом D^*. Там, где операция не определена, ставится знак «×» — безразличное состояние, которое при минимизации функции доопределится до значений «1» или «0».

Таблица 4.14. Таблица переходов одного разряда регистра

Номер набора	$A1$	$A0$	X_i^t	Q_i^t	Q_i^{t+1} D_i^*	Микрооперация
0	0	0	0	0	0	Хранение данных: $Q^{t+1} = Q_i^t$
1	0	0	0	1	1	
2	0	0	1	0	0	
3	0	0	1	1	1	
4	0	1	0	0	0	Лог. «И»: $Q_i^{t+1} = X_i^t \cdot Q_i^t$
5	0	1	0	1	0	
6	0	1	1	0	0	
7	0	1	1	1	1	
8	1	0	0	0	1	«Счётный режим»: $Q_i^{t+1} = Q_i^t$
9	1	0	0	1	0	
10	1	0	1	0	1	
11	1	0	1	1	0	
12	1	1	0	0	×	Не определена
13	1	1	0	1	×	
14	1	1	1	0	×	
15	1	1	1	1	×	

Переносим табл. 4.14 на карту Карно и минимизируем функцию D^* (рис. 4.29). Выбранные покрытия доопределяют неопределённые состояния «×» до логической «1», если «×» попало хотя бы в один контур, до логического «0», если «×» не попал ни в один контур.

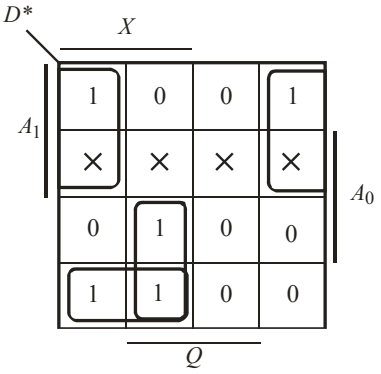


Рис. 4.29. Карты Карно для разряда регистра

В результате минимизации получаем

$$D^* = A_1\overline{Q} + \overline{A_1}XQ + \overline{A_1}\,\overline{A_0}Q = A_1\overline{Q} + \overline{A_1}Q(X + \overline{A_0}).$$

Теперь можно нарисовать схему одного разряда регистра, выполняющего заданные микрооперации (рис. 4.30).

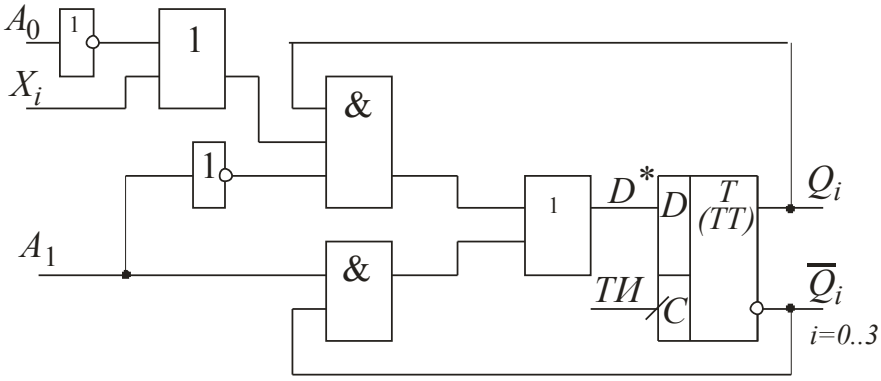


Рис. 4.30. Схема одного разряда регистра

Построим временную диаграмму работы разряда регистра (рис. 4.31).

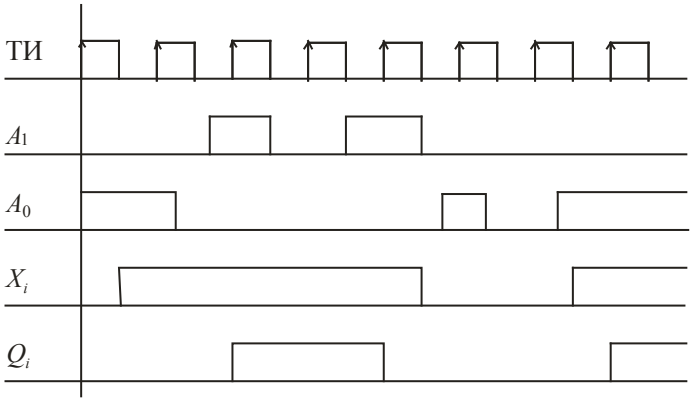


Рис. 4.31. Временная диаграмма работы одного разряда регистра

Теперь перейдём к работе со схемой в САПР БИС «Ковчег 3.04». Создадим новый проект *lab*4 с головной схемой с именем *REGS* аналогично, как это описывалось в предыдущих лабораторных работах.

В открывшемся окне схемного редактора соберём схему одного разряда регистра (см. рис. 4.30) в среде САПР БИС «Ковчег 3.04».

Сначала разместим входы схемы: тактовый импульс, дополнительные сигналы A_0, A_1, которые будут задавать микрооперацию и вход данных *X*. Для этого необходимо выбрать на панели инструментов кнопку «**Порт групповой**» (рис. 4.32, а) для

задания шины входного сигнала *A*, затем кнопку «**Порт одиночный**» (рис. 4.32, б) для установки одиночных сигналов:

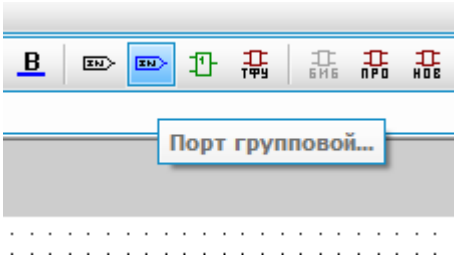


Рис. 4.32, а. Кнопка «**Порт групповой**» на панели инструментов

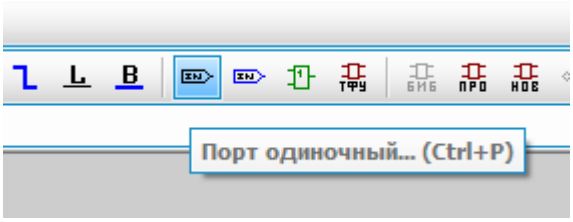


Рис. 4.32, б. Кнопка «**Порт одиночный**» на панели инструментов

Появится окно настроек порта. Необходимо указать имя порта и в случае шины — старший и младший разряды (рис. 4.33, *а*).

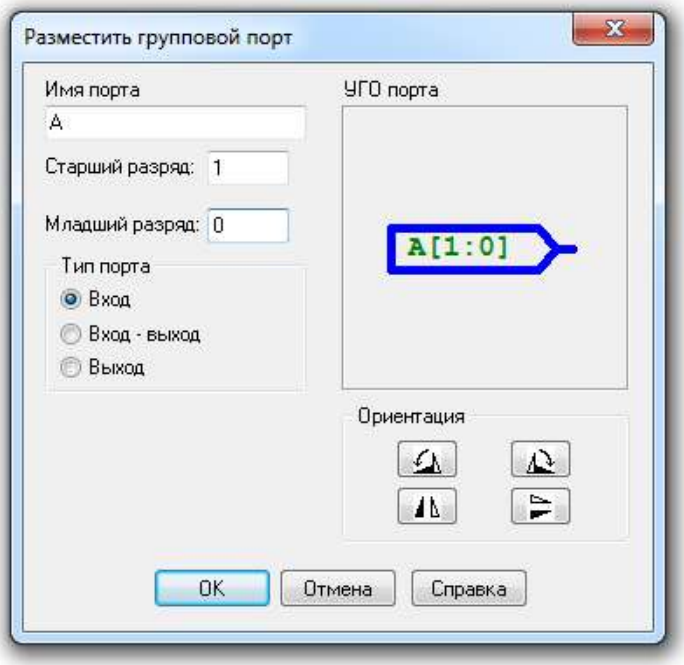


Рис. 4.33, а. Задание параметров группового порта

Для одиночного порта указываются имя порта и, при наличии, разряд (рис. 4.33, *б*).

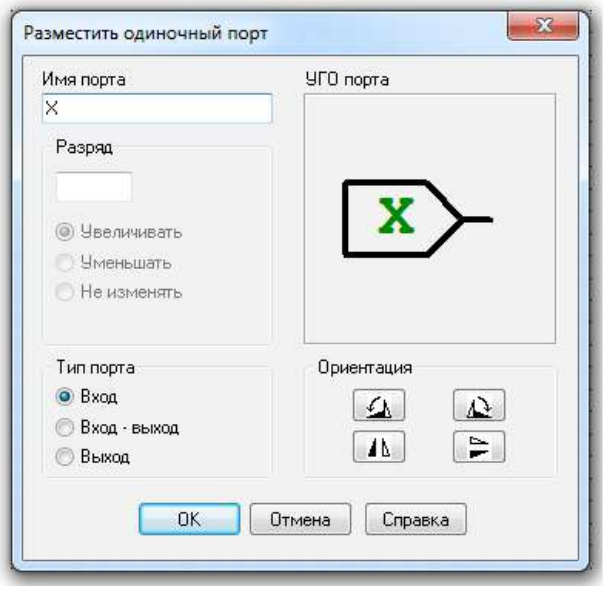


Рис. 4.33, б. Задание параметров одиночного порта

Разместив порты в рабочем поле, нужно правильно оформить и вывести сигналы шины. Для этого необходимо установить метку шины, выбрав инструмент «**Метка шины**» на панели инструментов (рис. 4.34, *а*), в появившемся окне задать имя метки и разряды (рис. 4.34, *б*), и разместить её на шине (рис. 4.34, *в*):

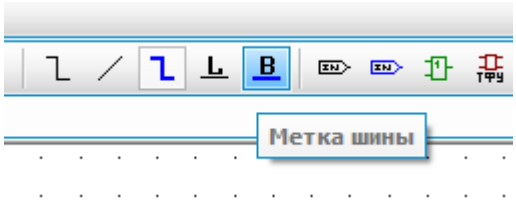


Рис. 4.34, а. Кнопка «**Метка шины**» на панели инструментов

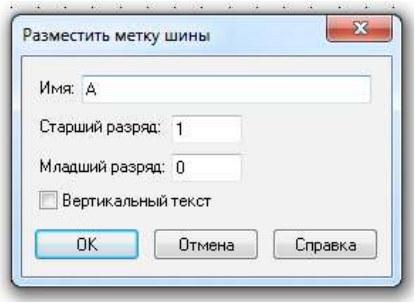


Рис. 4.34, б. Задание параметров — метки шины

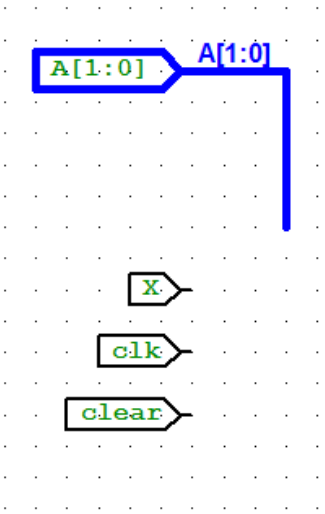


Рис. 4.34, в. Размещение метки шины

Для использования сигналов в схеме необходимо вывести конкретные цепи шины *A*. Для этого необходимо выбрать инструмент «**Ортогональная связь**» на панели инструментов, вывести цепи с шины (рис. 4.35, а), и задать метки связи с помощью соответствующей команды (рис. 4.35, б, в).

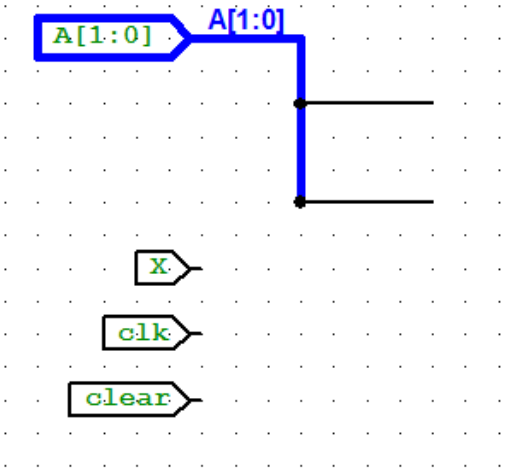


Рис. 4.35, а. Вывод сигналов с шины

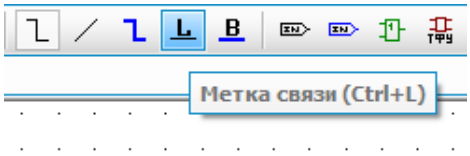


Рис. 4.35, б. Кнопка «**Метка связи**» на панели инструментов

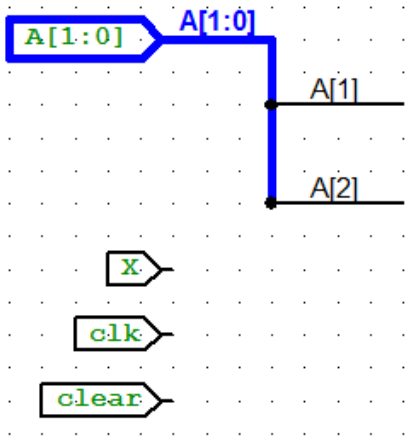


Рис. 4.35, в. Размещение метки связи

Все входные и выходные сигналы схемы должны иметь периферийные ячейки. Вызовите библиотеку ячеек, нажав кнопку «**Ячейка**» на панели инструментов (рис. 4.36, *а*), и в появившемся окне «**Выбор ячейки**» перейдите к группе «**Периферийные ячейки — входные**», в списке ячеек выберите ячейку цифрового входа «*DP*» (рис. 4.36, *б*) (вы можете более подробно изучить функциональность ячейки нажав кнопку «**Описание**»). После этого разместите данные ячейки на входные цепи схемы. Аналогично рассмотренному выше случаю задайте метки на выходах ячеек входа, что позволит не проводить непосредственной связи между элементами схемы, а объединить их через метки связи (рис. 4.36, *в*).

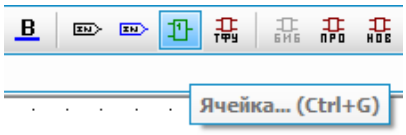


Рис. 4.36, а. Кнопка «**Ячейка**» на панели инструментов

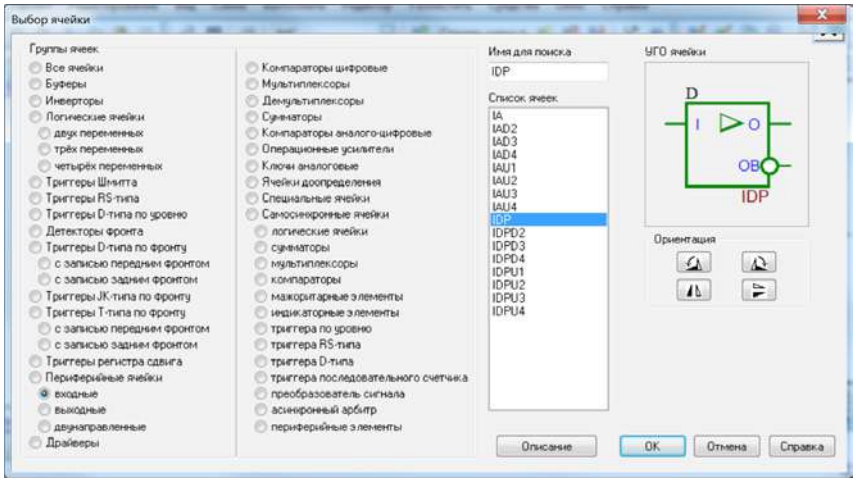


Рис. 4.36, б. Выбор ячейки IDP

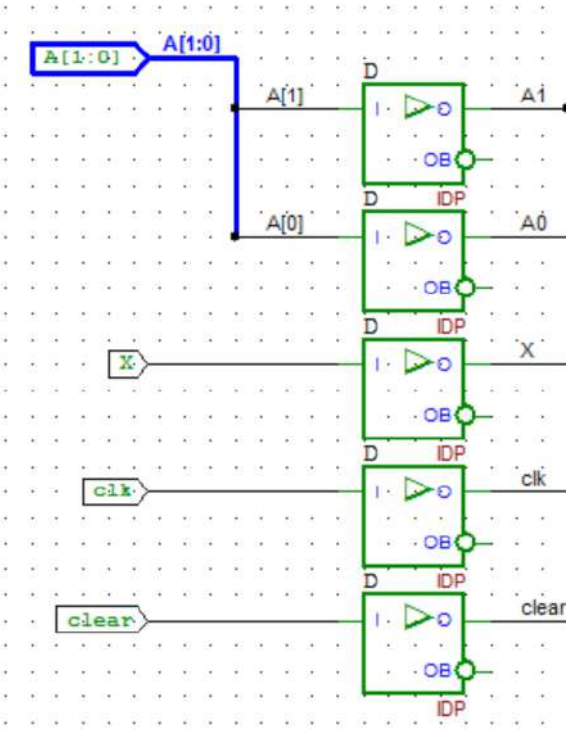


Рис. 4.36, в. Размещение периферийных ячеек

Аналогичным образом сформируем выходы схемы *Out*, *nOut*. Для этого при выборе инструмента «**Порт одиночный**» необходимо указать, что размещается выходной порт. После этого установите выходные периферийные ячейки из списка «**Периферийные ячейки — выходные — OA**» (рис. 4.37, *а*, *б*).

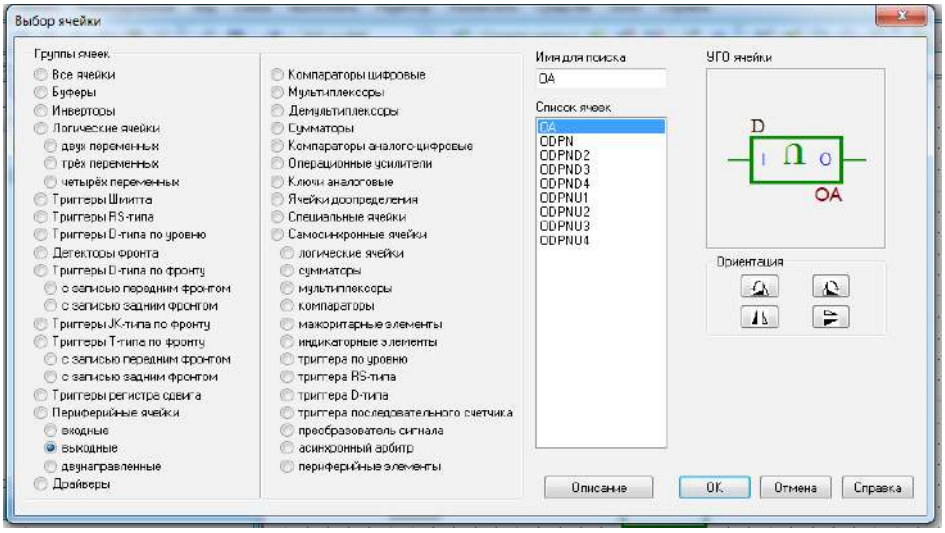


Рис. 4.37, а. Выходная ячейка

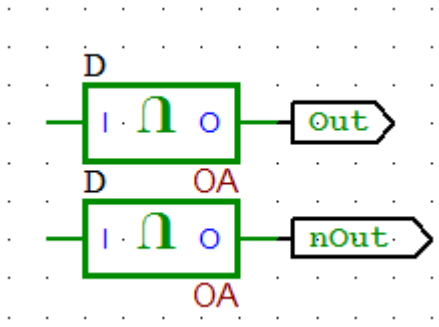


Рис. 4.37, б. Размещение выходных ячеек

Теперь можно приступить к построению устройства в соответствии с синтезированной схемой (рис. 4.30). Выбор необходимых ячеек можно осуществлять либо через инструмент «**Ячейки**» на панели инструментов, либо во вкладке «**Базовые ячейки**» в области панелей быстрого доступа (рис. 4.38).

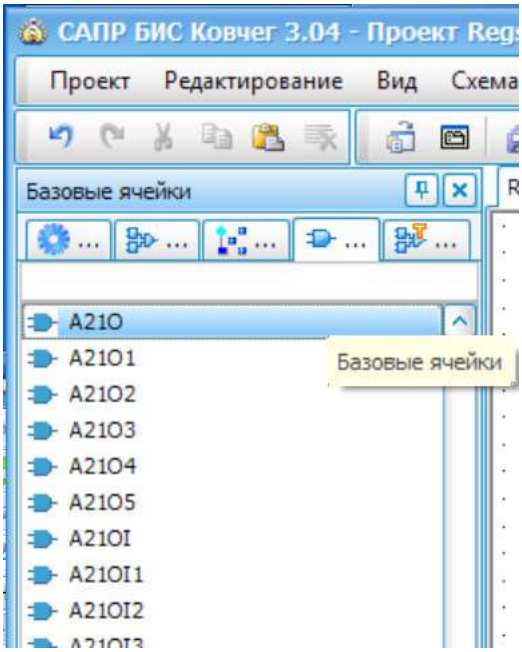


Рис. 4.38. Панель «**Базовые ячейки**»

Для реализации схемы необходимы ячейки *OR*2 (ячейка операции «ИЛИ» с двумя входами, *AND*2, *AND*3 (ячейки операции «И», с соответствующим количеством входов).

Соединим ячейки. Сделать это можно непосредственным соединением входов и выходов ячеек с помощью команды «**Связь ортогональная**», а также при помощи инструмента «**Метка связи**» (рис. 4.39):

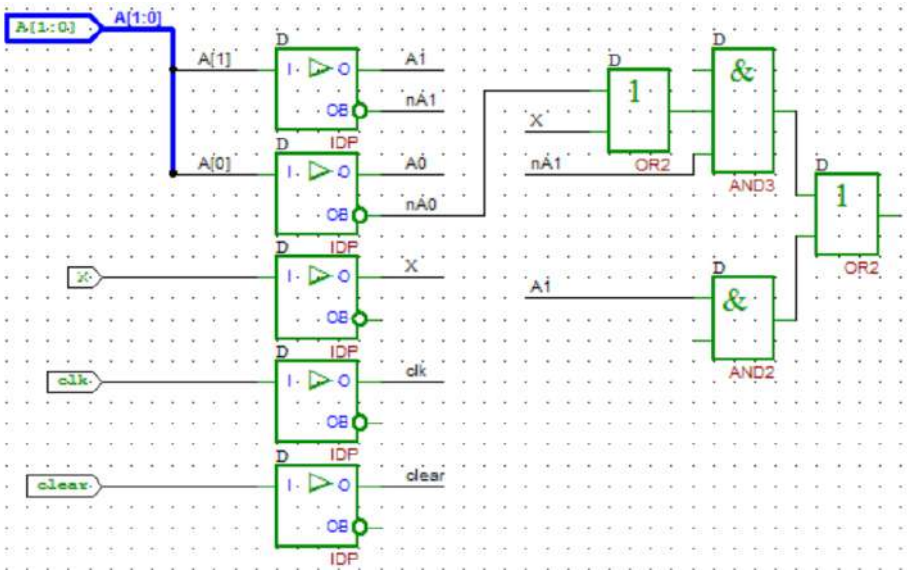


Рис. 4.39. Пример использования меток связи для построения схемы

Теперь разместим основную ячейку данного устройства — *D*-триггер. Его можно найти, открыв инструмент «**Ячейка**» в группе «**Триггеры *D*-типа по фронту — с записью передним фронтом**». Нас интересует исполнение «*FDC*», имеющее вход «*CLR*», предназначенный для сброса триггера в нулевое значение (рис. 4.40).

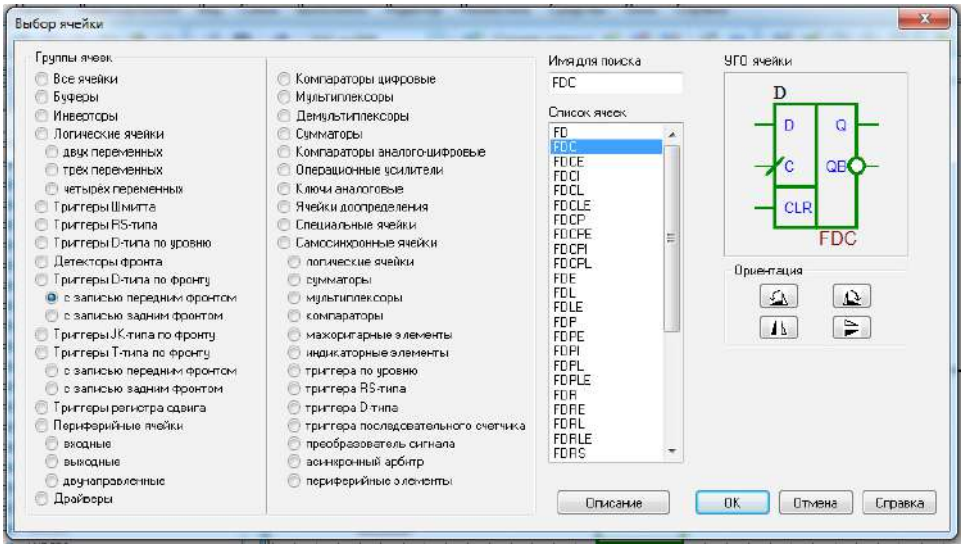


Рис. 4.40. Выбор ячейки FDC

Разместим триггер на схеме и проведём оставшиеся цепи (рис. 4.41).

Последним этапом разработки схемы пронумеруем УГО. Для этого перейдите в меню «**Редактор**» в строке меню и выберите команду «**Автоматически нумеровать**

УГО» (рис. 4.42). После этого в появившемся диалоговом окне дайте согласие на выполняемую операцию, после чего вы увидите, что каждому УГО был присвоен свой номер.

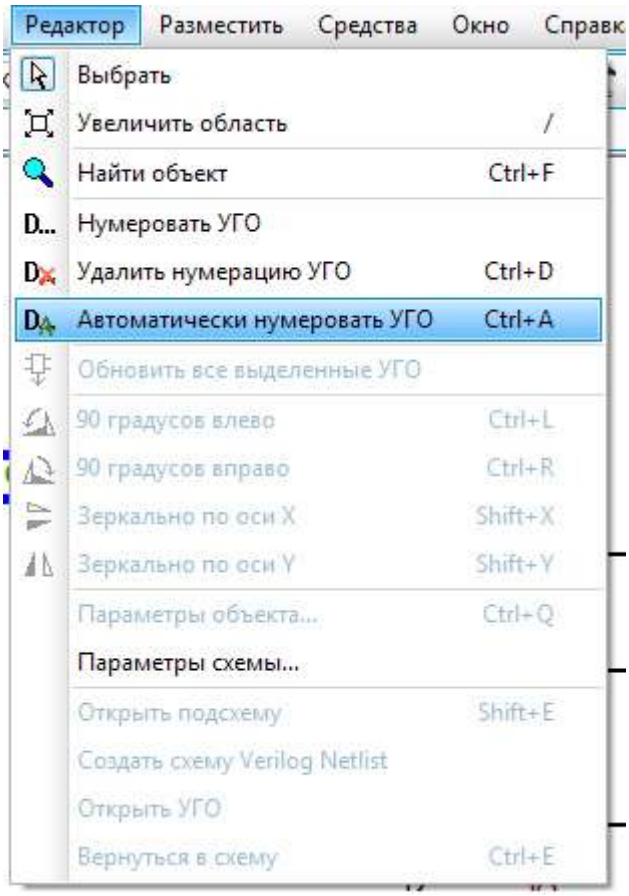


Рис. 4.42. Команда «Автоматическая нумерация УГО»

После этого необходимо выполнить проверку схемы на отсутствие ошибок. Для этого нажмите кнопку с зеленым треугольником «**Трансляция схемы**» на панели инструментов (рис. 4.43, *а*). После чего будет выведено окно, содержащее краткую информацию о выполнении операции (рис. 4.43, *б*). Подробная информация находится в консоли внизу окна программы.

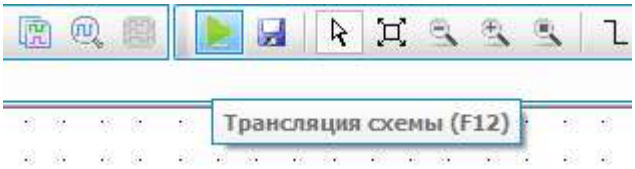


Рис. 4.43, а. Кнопка «**Трансляция схемы**» на панели инструментов

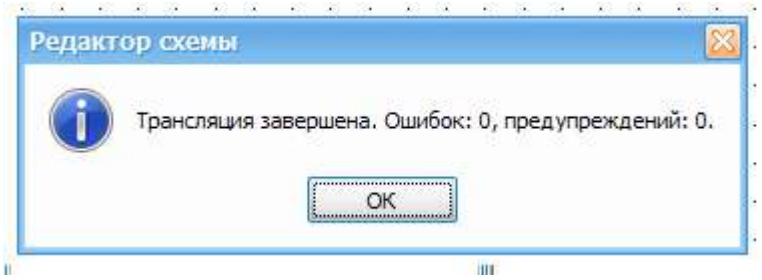


Рис. 4.43, б. Результат выполнения операции трансляции схемы

Перейдём к подготовке моделирования схемы. В первую очередь опишем группы цепей, которые будут отображены в результатах моделирования устройства. Для этого необходимо перейти в подсистему «**Контрольные точки**» в области панелей быстрого доступа (рис. 4.44). Откроется редактор с автоматически сгенерированным описанием. Замените его следующим:


```
1.  ТИ, Сброс, Операция, ВходДанные, Выход;
2.
3.  ТИ       : clk;
4.  Сброс    : clear;
5.  Операция : A[1..0];
6.  ВходДанные  : X;
7.  Выход    : OUT, nOUT;
```

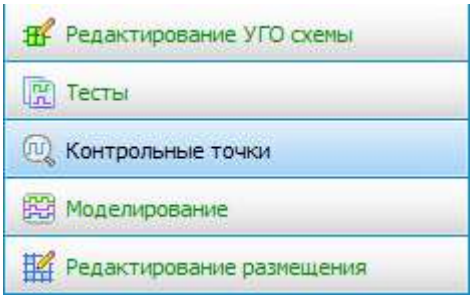


Рис. 4.44. Подсистема «**Контрольные точки**» на панели быстрого доступа

Контрольные точки также требуют трансляции. Для этого нужно нажать эмблему зелёной стрелки на панели инструментов или в верхней части панелей быстрого доступа, которая соответствует команде «**Трансляция контрольных точек**». Далее также будет выведено сообщение о результатах выполнения трансляции.

Для моделирования схемы и построения временных диаграмм необходимо описать входные воздействия для разработанной схемы. Для этого нажмите панель «**Тесты**» в области панелей быстрого доступа. Удалите автоматически сгенерированный текст и опишите тестовые воздействия в соответствии с вашим теоретически построенным временным диаграммам. В нашем случае диаграмма (рис. 4.45) будет проверяться следующим тестовым примером:

```
Сброс, ТестСчРежим = 4, ТестИ, ТестХранение = 2, ТестСчРежим = 1, ТестХра-
нение = 2;

Сброс:
clear = 0,1:2,0;
clk = +:4;
X = 0;
A[1..0]D = 0;

ТестСчРежим:
A[1..0]D = 2;
clk = +;
X = 0;
clear = 0;

ТестИ:
A[1..0]D = 1;
clk = +:2;
X = 1,0;
clear = 0;

ТестХранение:
A[1..0]D = 0;
clk = +;
X = 0;
clear = 0;
```


Транслируйте тесты и переходите к подсистеме «**Моделирование**». При первом запуске программа должна предложить настроить характеристики тактового импульса (сигнал *clk*). Задайте смещение 25%, а длительность — 50%.

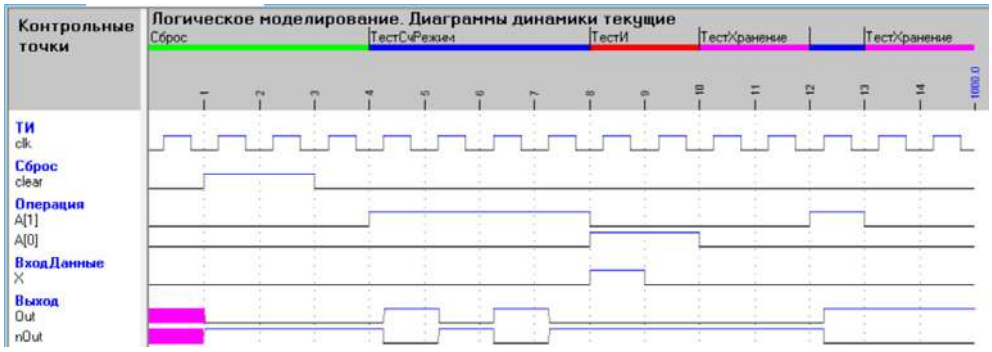


Рис. 4.45. Временная диаграмма

После чего будет построена временная диаграмма, которую необходимо сравнить с теоретической (рис. 4.45), чтобы убедиться, что схема функционирует верно. При несовпадении временных диаграмм с рассчитанными необходимо найти ошибки синтеза или ошибки в схеме и исправить их, выполнить компиляцию и моделирование проекта снова и убедиться в правильности новых результатов.

4.3. Перечень индивидуальных заданий

1. Используя триггеры, разработать 4-разрядный регистр, выполняющий несколько микроопераций. Тип триггера и перечень микроопераций указаны в индивидуальном задании.

3. Используя средства САПР «Ковчег 3.04» промоделировать работу регистра, сравнить временные диаграммы с расчётными.

Вариант 1
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «И»;
- лог. «ИЛИ»;
- «хранение».

Вариант 2
Тип триггера: *D*. Выполняемые микрооперации:
- «приём данных»;
- лог. «И»;
- лог. «=».

***Вариант* 3**
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «И»;
- лог. «*XOR*».

***Вариант* 4**
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «ИЛИ»;
- лог. «1».

***Вариант* 5**
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «И»;
- «0».

***Вариант* 6**
Тип триггера: *JK*. Выполняемые микрооперации:
- приём данных;
- лог. «ИЛИ»;
- лог. «*XOR*».

***Вариант* 7**
Тип триггера: *JK*. Выполняемые микрооперации:
- приём данных;
- лог. «*XOR*»;
- лог. «0».

***Вариант* 8**
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «И»;
- лог. «ИЛИ».

***Вариант* 9**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- лог. «1»;
- лог. «=».

***Вариант* 10**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- лог. «ИЛИ»;
- «приём данных».

***Вариант* 11.**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- лог. «=»;
- «0».

***Вариант* 12**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «ИЛИ»;
- лог. «=»;
- $Q^{t+1} = \bar{Q}^t + x$.

***Вариант* 13**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «ИЛИ»;
- лог. «*XOR*»;
- лог. «0».

***Вариант* 14**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- лог. «=»;
- «счётный режим»;
- лог. «1».

***Вариант* 15**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «*XOR*»;
- лог. «ИЛИ»;
- лог. «1».

***Вариант* 16**
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «И»;
- лог. «0»;
- лог. «1».

***Вариант* 17**
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «ИЛИ»;
- «счётный режим».

***Вариант* 18**
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «ИЛИ»;
- лог. «*XOR*»;
- «приём данных».

***Вариант* 19**
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «1»;
- лог. «И»;
- «счётный режим».

***Вариант* 20**
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «*XOR*»;
- приём данных;
- «счётный режим»;
- лог. «1».

Вариант 21
Тип триггера: *D*. Выполняемые микрооперации:
- приём данных;
- лог. «*XOR*»;
- лог. «ИЛИ».

***Вариант* 22**
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «=»;
- приём данных;
- лог. «ИЛИ».

***Вариант* 23**
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «И»;
- лог. «ИЛИ»;
- «счётный режим»;
- лог. «1».

***Вариант* 24**
Тип триггера: *D*. Выполняемые микрооперации:
- лог. «И»;
- лог. «=»;
- приём данных;
- лог. «0».

***Вариант* 25**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «ИЛИ»;
- лог. «И»;
- «счётный режим».

***Вариант* 26**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- лог. «ИЛИ»;
- «хранение».

***Вариант* 27**
Тип триггера: *JK*. Выполняемые микрооперации:
- приём данных;
- лог. «*XOR*»;
- лог. «0»;
- лог. «1»;

***Вариант* 28**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «*XOR*»;
- «1»;
- «Хранение»;

***Вариант* 29**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- лог. «ИЛИ»;
- лог. «*XOR*»;

***Вариант* 30**
Тип триггера: *JK*. Выполняемые микрооперации:
- лог. «И»;
- «хранение»;
- «счётный режим»;
- лог. «0».