



ІІТМО

Элементы цифровых и аналого-цифровых устройств

Николаев Николай Анатольевич

2023

Комбинационные логические элементы



- Мультиплексоры
- Демультимплексоры
- Шифраторы (кодеры)
- Дешифраторы (декодеры)

Последовательные логические элементы (элементы с памятью)

- Триггеры
- Регистры
- Счетчики

Комбинационные логические схемы – это логические устройства, в которых состояние зависит только от текущего состояния их входов. Такие схемы строятся на базе исключительно логических элементов и не требуют элементов памяти.



К комбинационным логическим элементам относятся:

- Мультиплексоры
- Демультимплексоры
- Шифраторы (кодеры)
- Дешифраторы (декодеры)

Список использованных источников

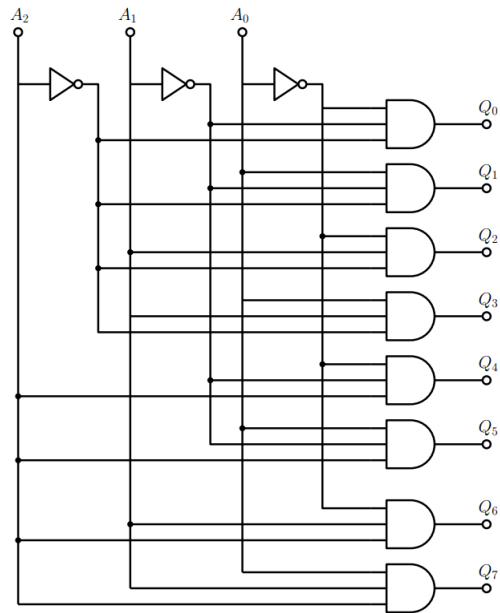
1. Sarma M. S. Introduction to electrical engineering. – New York : Oxford University Press, 2001. – С. 715-716.
2. Tokheim R. L. Digital Electronics: Principles and Applications, 8th Edition. – McGraw-Hill, Inc., 2014.
3. Kleitz W. Digital Electronics: A practical approach with VHDL. – Prentice Hall, 2011.
4. Harris S., Harris D. Digital design and computer architecture: arm edition. – Morgan Kaufmann, 2015.
5. Paul Scherz, Simon Monk. Practical Electronics for Inventors, Fourth Edition. - McGraw-Hill, Inc., 2016.



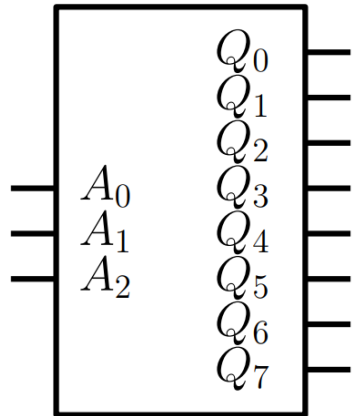
Дешифратор (декодер)

Дешифратор (декодер) – комбинационное логическое устройство, которое декодирует (преобразует) n – битную комбинацию (двоичный код) в m выходов ($m \leq 2^n$).

Блок схема для декодирования трехбитного сигнала приведена на рисунке. Схема имеет три входа и восемь выходов – по одному для каждой комбинации входных сигналов.



Дешифратор (декодер)



| A_2 | A_1 | A_0 | Q_7 | Q_6 | Q_5 | Q_4 | Q_3 | Q_2 | Q_1 | Q_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



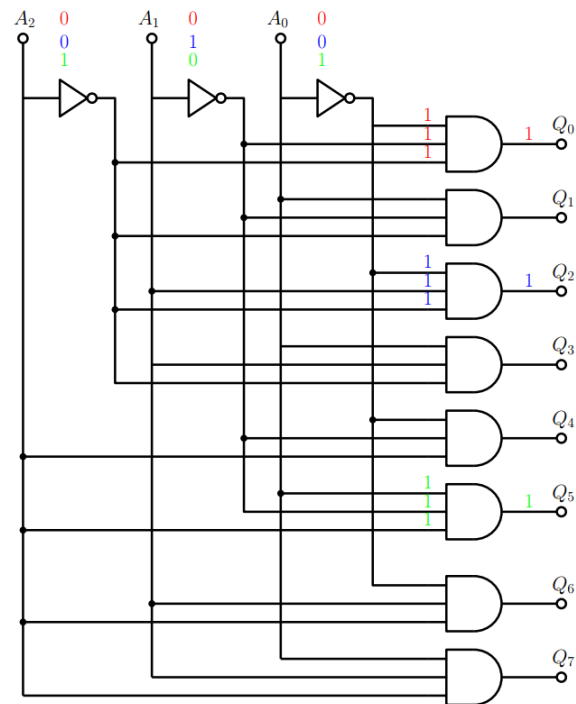
Дешифратор (декодер)

На рисунке приведен процесс преобразования трех двоичных комбинаций входного сигнала в соответствующий выходной сигнал

$000 \rightarrow Q_0$

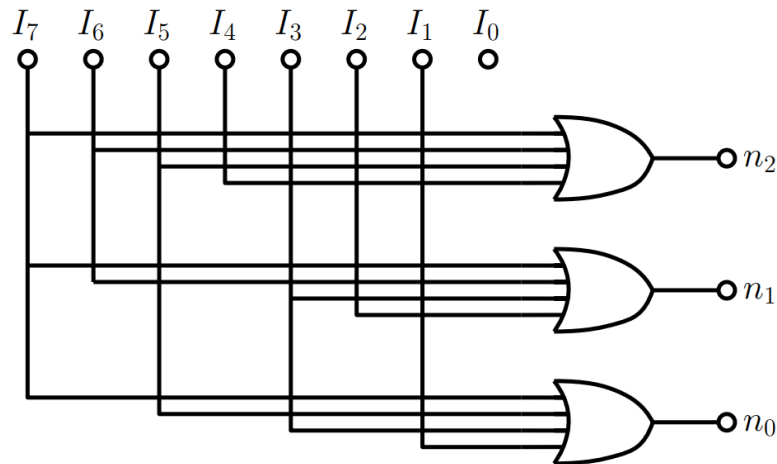
$010 \rightarrow Q_2$

$101 \rightarrow Q_5$

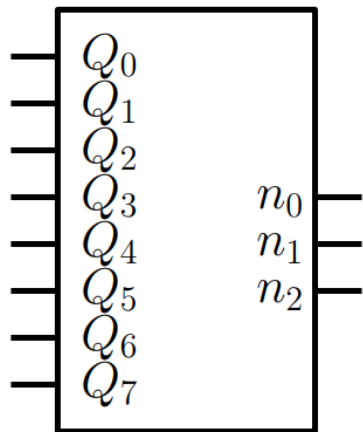


Шифратор (кодер)

Кодирование - это процесс формирования закодированного представления набора входных данных, и он является обратной операцией декодирования. Шифратор (кодер) - это комбинационное устройство, которое генерирует n -разрядный двоичный код, однозначно идентифицирующий один из m активированных входов ($0 \leq m \leq 2^n - 1$). На рисунке показана структурная схема 8-элементного 3-разрядного шифратора.



Шифратор (кодер)



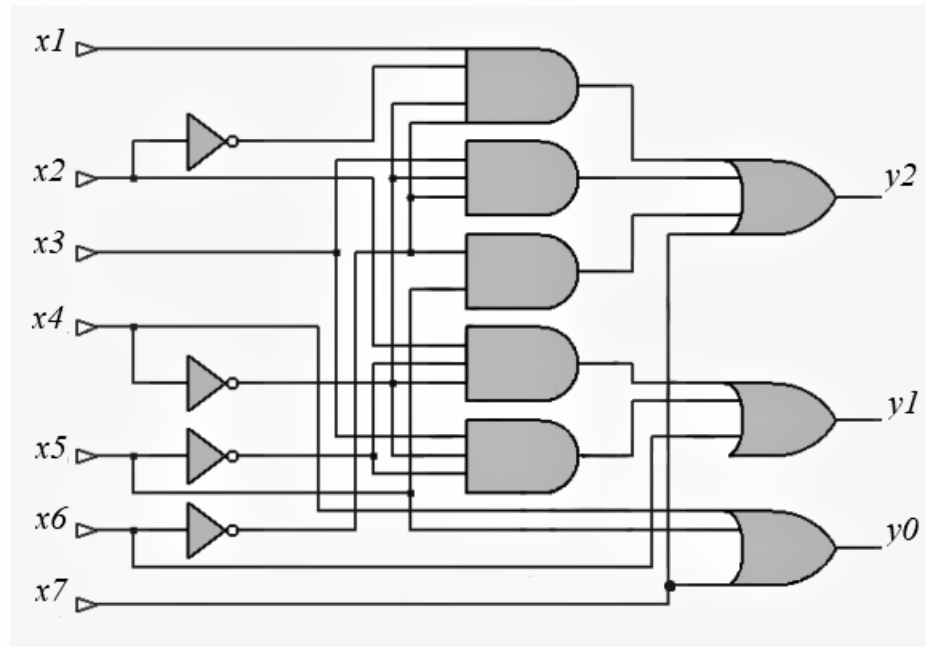
| I_7 | I_6 | I_5 | I_4 | I_3 | I_2 | I_1 | I_0 | n_2 | n_1 | n_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |



Шифратор (кодер)

На рисунке приведена схема приоритетного шифратора 8-3.

В отличие от обычного шифратора, приоритетный шифратор при наличии на входах нескольких возбужденных линий (запросов) будет на выходе выдавать сигнал (двоичный код), соответствующих старшему входу.



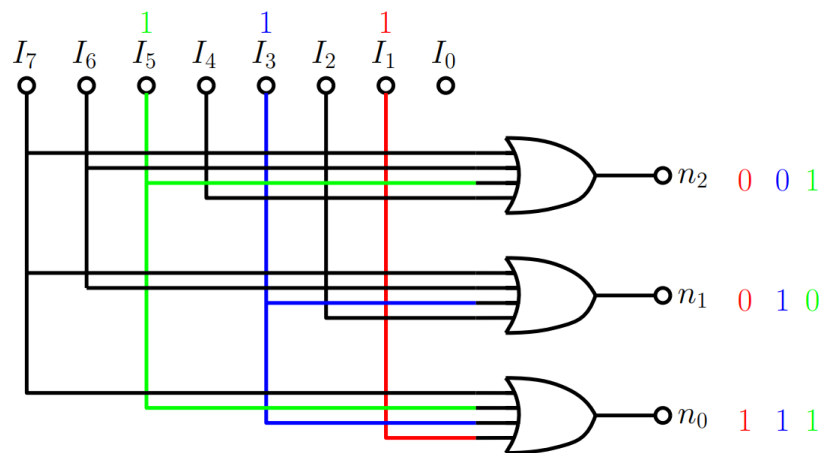
Шифратор (кодер)

На рисунке приведены
процессы преобразования
трех входных сигналов в
соответствующие выходные
двоичные комбинации

$I_1 \rightarrow 001$

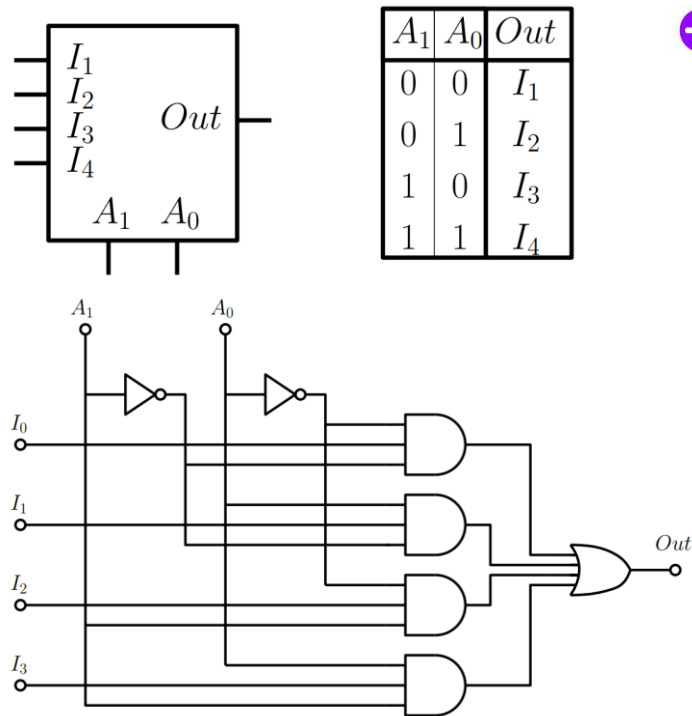
$I_3 \rightarrow 010$

$I_5 \rightarrow 101$

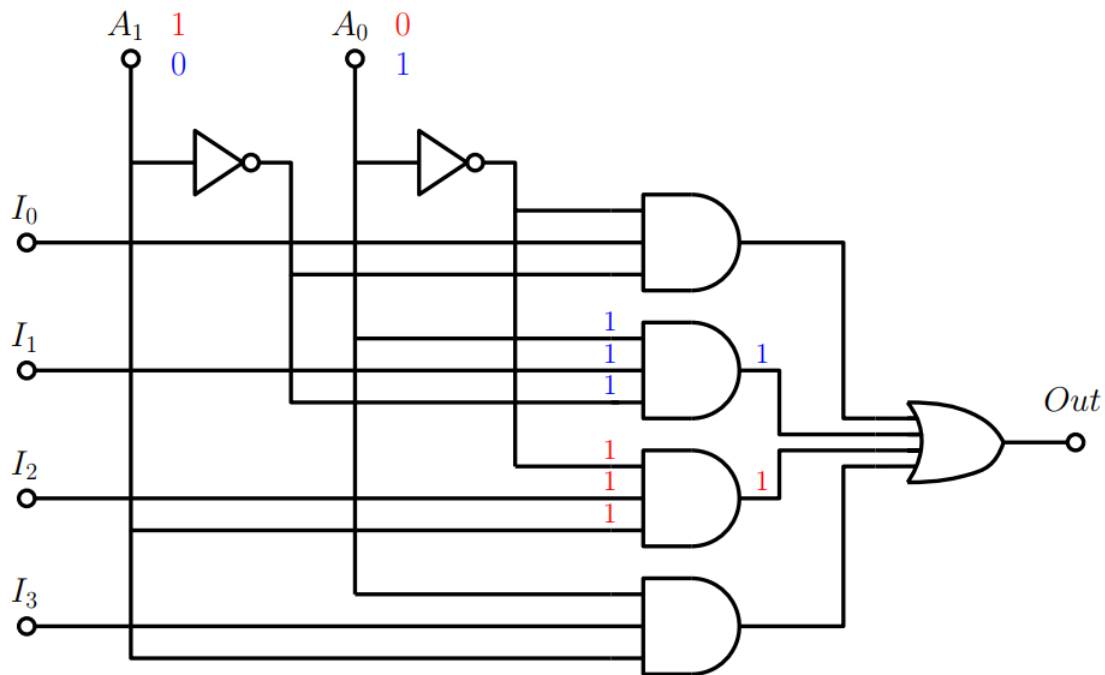


Мультиплексор

Мультиплексор - это селектор данных. Мультиплексор - это комбинационная схема, которая выбирает один из нескольких возможных входных сигналов и направляет этот сигнал на один выходной терминал. Выбор конкретного входного сигнала управляется набором переменных выбора. Мультиплексор с n входными переменными выбора обычно может выбрать один из 2^n входных сигналов.

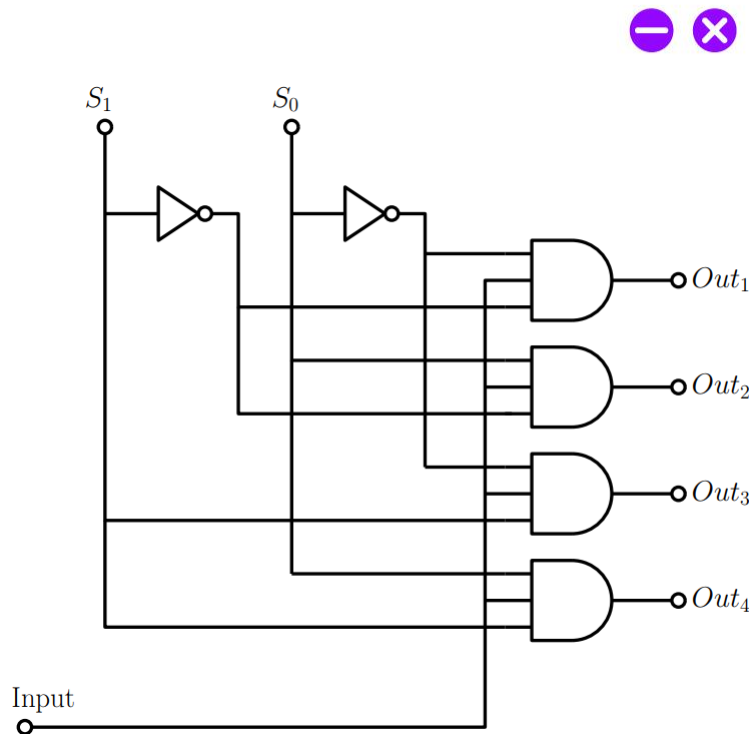


Мультиплексор



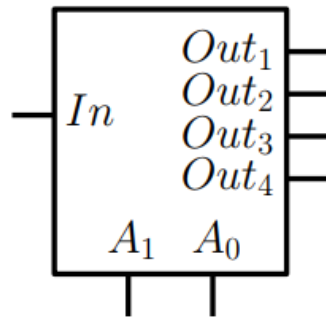
Демультимплексор

Демультимплексор (или распределитель данных) является противоположностью мультиплексора. Он принимает один входных данных и направляет его на один из нескольких возможных выходов. Чтобы выбрать выходной сигнал (Out_1 , Out_2 , Out_3 или Out_4), на который вы хотите отправить входной сигнал, необходимо подать соответствующую логическую комбинацию на входы (S_0 , S_1).



Демультимплексор

Алгоритм работы демультимплексора описывается таблицей истинности.



| A_1 | A_0 | Output |
|-------|-------|--------------|
| 0 | 0 | $Out_1 = In$ |
| 0 | 1 | $Out_2 = In$ |
| 1 | 0 | $Out_3 = In$ |
| 1 | 1 | $Out_4 = In$ |

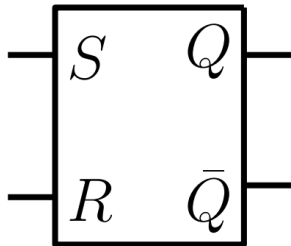
В последовательных логических элементах, в отличие от элементов комбинационной логики, присутствуют элементы памяти, таким образом, выходной сигнал элемента зависит не только от текущего состояния (сигналов на его входах), но и от предыдущих состояниях, информация о которых хранится в элементе памяти.

К последовательным логическим элементам относятся:

- Триггеры
- Регистры
- Счетчики

RS триггер

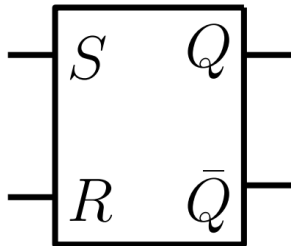
На слайде приведено условное обозначение RS-триггера и соответствующая таблица истинности



| <i>Input</i> | | <i>Output</i> | |
|--------------|--------|---------------|-----------------|
| In_1 | In_2 | Q_{n+1} | \bar{Q}_{n+1} |
| 0 | 0 | Q_n | \bar{Q}_n |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | Not allowed | |

RS триггер

На слайде приведено условное обозначение RS-триггера и соответствующая таблица истинности



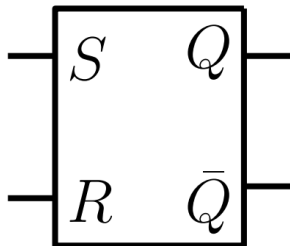
| <i>Input</i> | | <i>Output</i> | |
|--------------|--------|---------------|-----------------|
| In_1 | In_2 | Q_{n+1} | \bar{Q}_{n+1} |
| 0 | 0 | Q_n | \bar{Q}_n |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | Not allowed | |

Можно ли реализовать RS-триггер на базе блоков/элементов комбинационной логики?



RS триггер

На слайде приведено условное обозначение RS-триггера и соответствующая таблица истинности



| <i>Input</i> | | <i>Output</i> | |
|--------------|--------|---------------|-----------------|
| In_1 | In_2 | Q_{n+1} | \bar{Q}_{n+1} |
| 0 | 0 | Q_n | \bar{Q}_n |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | Not allowed | |

Можно ли реализовать RS-триггер на базе блоков/элементов комбинационной логики?

Конечно, можно!!!

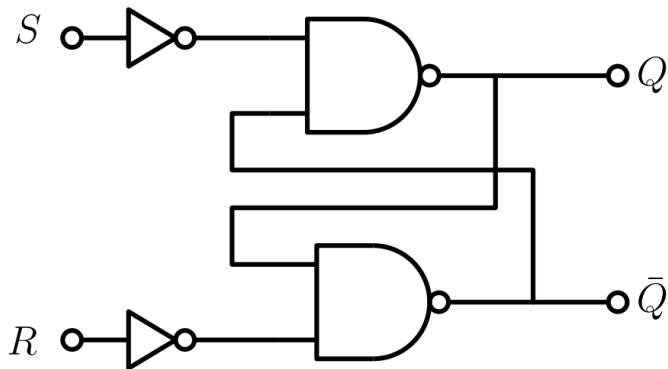


RS триггер

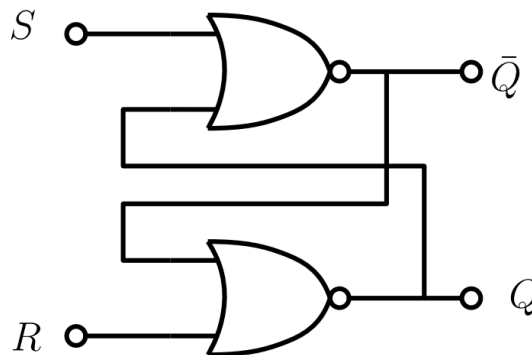
RS триггер может быть реализован на базе двух базовых типов логики – на базе элементов И-НЕ и ИЛИ-НЕ



Cross-NAND SR flip-flop

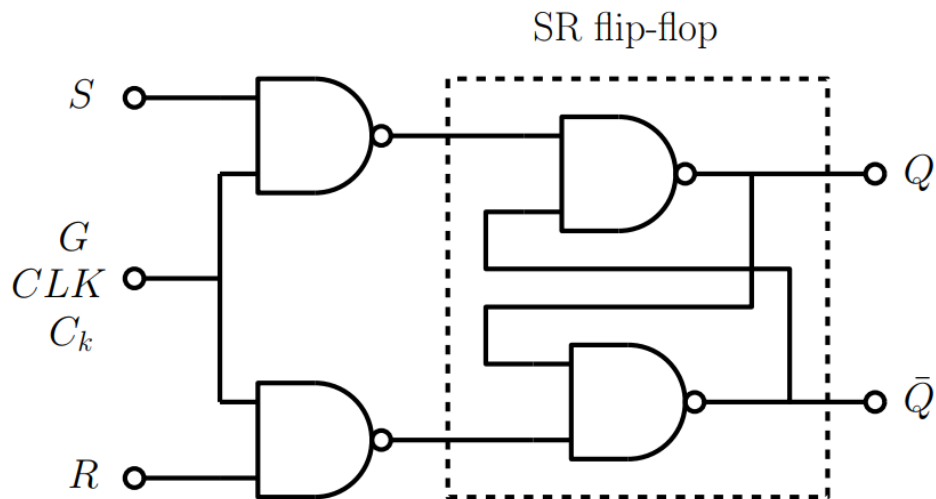


Cross-NOR SR flip-flop



Синхронный RS триггер

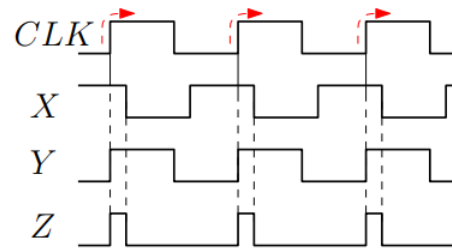
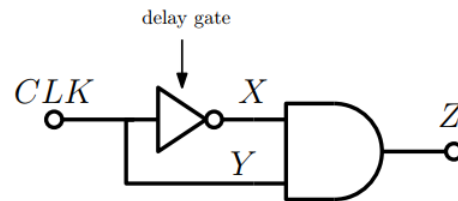
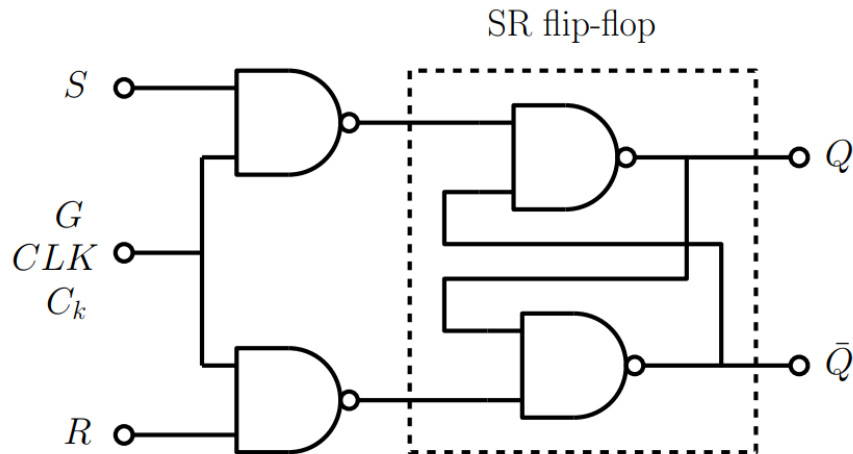
Триггер SR, запускаемый по уровню - это синхронный триггер, означающий включение или отключение входов S и R с помощью управляющего импульса.



| C_k | S | R | Q | \bar{Q} | Mode |
|-------|---|---|---|-----------|-----------------|
| 0 | 0 | 0 | Q | \bar{Q} | Hold |
| 0 | 0 | 1 | Q | \bar{Q} | Hold SR-inputs |
| 0 | 1 | 0 | Q | \bar{Q} | Hold disabled |
| 0 | 1 | 1 | Q | \bar{Q} | Hold |
| 1 | 0 | 0 | 0 | 0 | Hold |
| 1 | 0 | 1 | 0 | 1 | RESET SR-inputs |
| 1 | 1 | 0 | 1 | 0 | SET enabled |
| 1 | 1 | 1 | Q | \bar{Q} | Inderterminate |

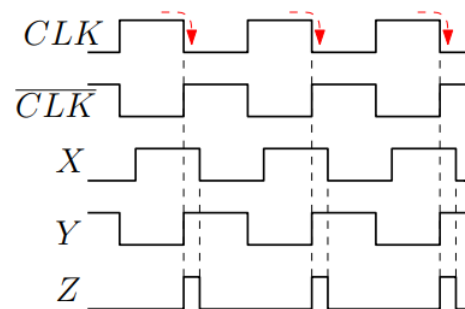
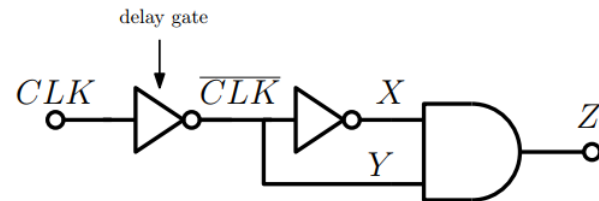
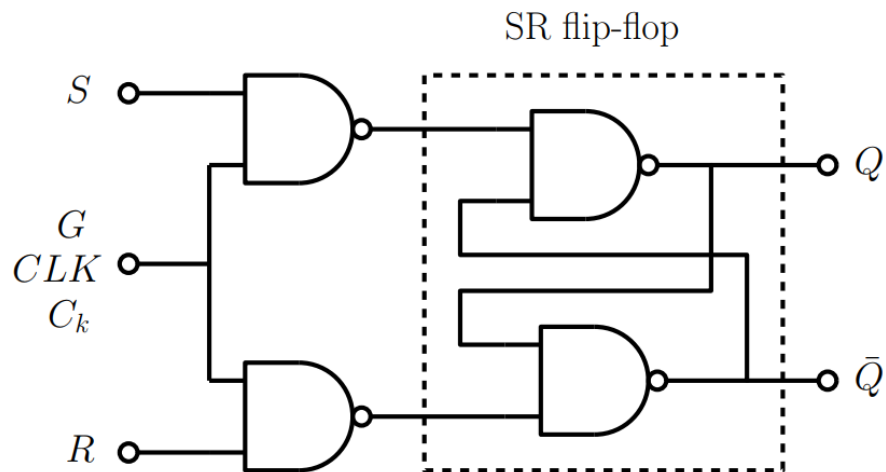
Синхронный RS триггер

Триггер, запускаемый по фронту, выполняет выборку входных данных только во время положительного или отрицательного тактового фронта (\uparrow = положительный фронт, \downarrow = отрицательный фронт).



Positive edge-triggered

Синхронный RS триггер

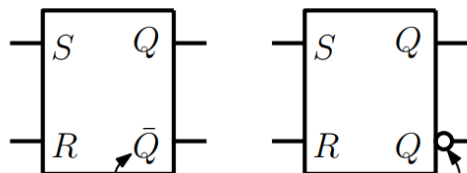


Negative edge-triggered

Обозначения RS-триггеров

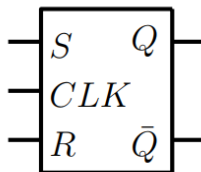


Simple SRFF

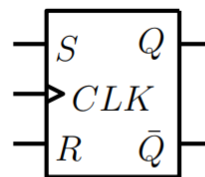


alternative
representation

level triggered

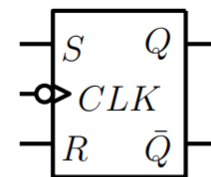


edge-triggered



positive edge-
triggered input

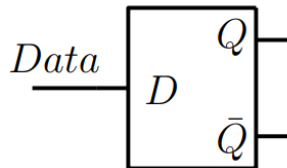
edge-triggered



negative edge-
triggered input

D триггер

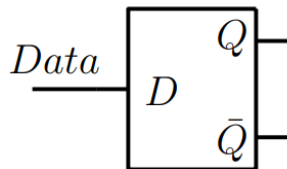
D триггер (защелка или элемент задержки). Триггер D-типа представляет собой устройством с одним входом. По сути, это триггер SR, где S заменяется на D, а R заменяется на D (инверсное D).



| D | Q | \bar{Q} | Mode |
|---|---|-----------|-------|
| 0 | 0 | 1 | Reset |
| 1 | 1 | 0 | Set |

D триггер

D триггер (защелка или элемент задержки). Триггер D-типа представляет собой устройством с одним входом. По сути, это триггер SR, где S заменяется на D, а R заменяется на D (инверсное D).

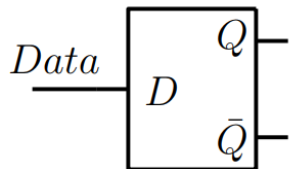


| D | Q | \bar{Q} | Mode |
|---|---|-----------|-------|
| 0 | 0 | 1 | Reset |
| 1 | 1 | 0 | Set |

Как реализовать D – триггер?

D триггер

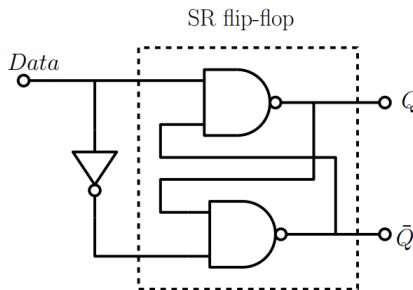
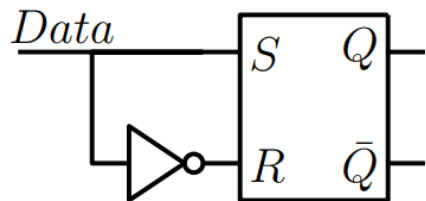
D триггер (защелка или элемент задержки). Триггер D-типа представляет собой устройством с одним входом. По сути, это триггер SR, где S заменяется на D, а R заменяется на D (инверсное D).



| D | Q | \bar{Q} | Mode |
|---|---|-----------|-------|
| 0 | 0 | 1 | Reset |
| 1 | 1 | 0 | Set |

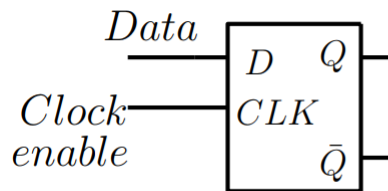
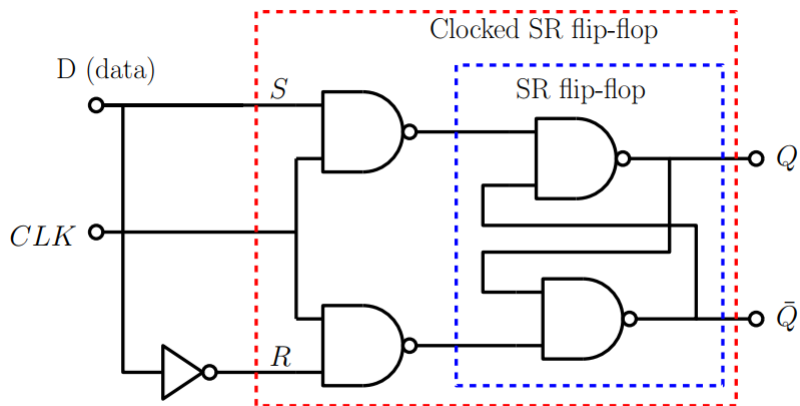
Как реализовать D – триггер?

Конечно, мы можем реализовать D-триггер на базе RS триггера!



D - триггер

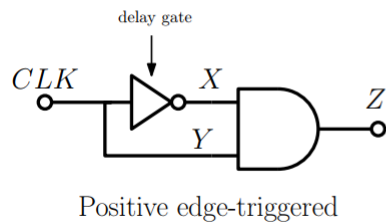
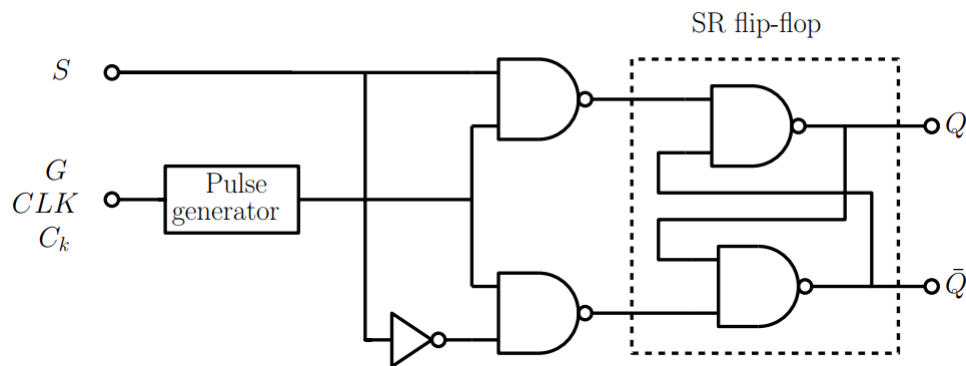
Можно реализовать синхронный D-триггер



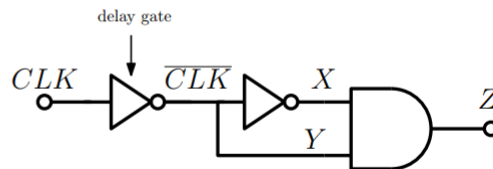
| Input | | Output | | |
|-------|---|--------|-----------|-------|
| CLK | D | Q | \bar{Q} | Mode |
| 0 | X | Q | \bar{Q} | Hold |
| 1 | 0 | 0 | 1 | Reset |
| 1 | 1 | 1 | 0 | Set |

D - триггер

Реализация синхронных D - триггеров

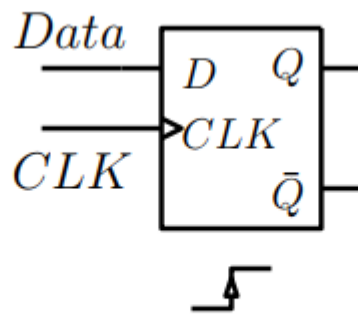
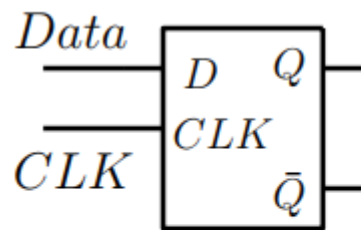


Positive edge-triggered

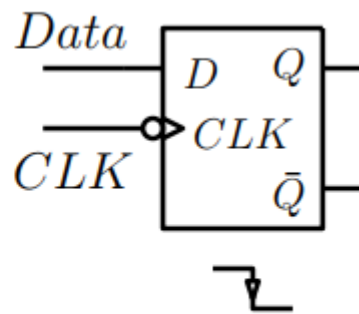


Negative edge-triggered

Обозначения D триггера



positive edge-
triggered input

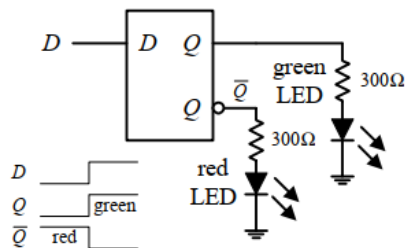


negative edge-
triggered input

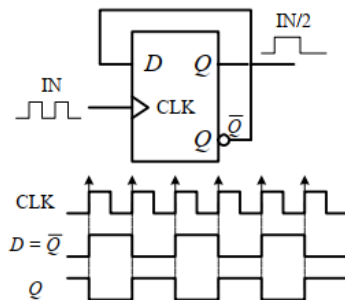
Применение D триггера



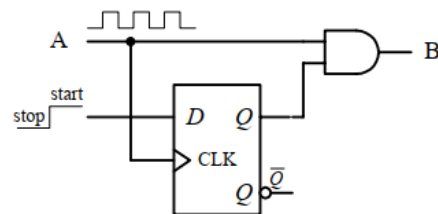
Stop-Go indicator



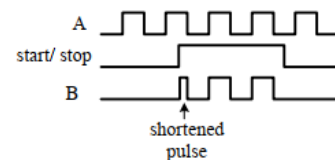
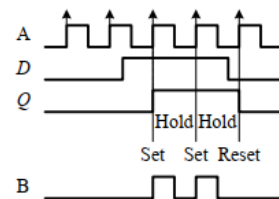
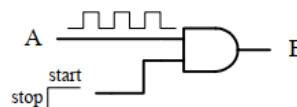
Divide-by-two counter



Synchronizer



What would happen if D flip-flop wasn't in place



Триггер JK напоминает триггер SR, где J действует как S, а K действует как R. Однако, в отличие от триггера SR, который имеет неопределенный режим, когда $S = 1$, $R = 1$, триггер JK имеет переключаемый режим, когда $J = 1$, $K = 1$.

Переключение означает, что выходы Q и \bar{Q} переключаются в свои противоположные состояния на каждом активном тактовом фронте.

Как и D триггер мы можем создать JK триггер на основе RS триггера.

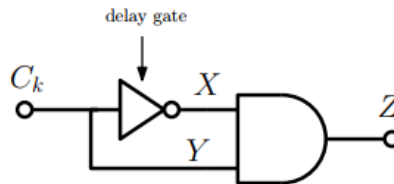
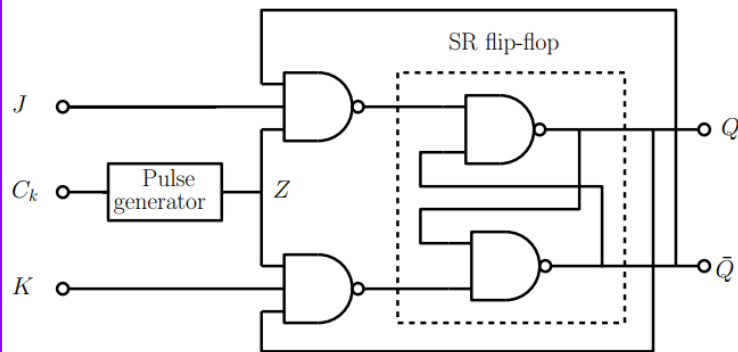
Чтобы создать триггер JK, изменяется внутренняя логическая схема триггера RS, в нее включаются две линии обратной связи с перекрестной связью между выходом и входом.

Однако эта модификация означает, что триггер JK не может быть только синхронным.



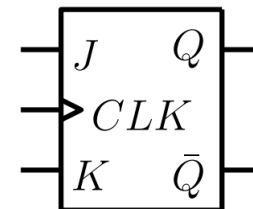
JK триггер

Синхронизация по переднему фронту



| C_k | J | K | Q | \bar{Q} | Mode |
|-------|---|---|---|-----------|--------|
| 0 | X | X | Q | \bar{Q} | Hold |
| 1 | X | X | Q | \bar{Q} | Hold |
| ↓ | X | X | Q | \bar{Q} | Hold |
| ↑ | 0 | 0 | Q | \bar{Q} | Hold |
| ↑ | 0 | 1 | 0 | 1 | Reset |
| ↑ | 1 | 0 | 1 | 0 | Set |
| ↑ | 1 | 1 | Q | Q | Toggle |

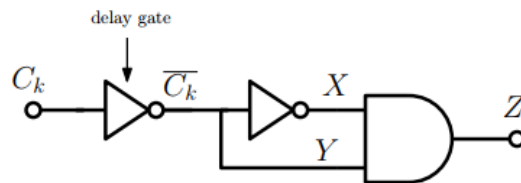
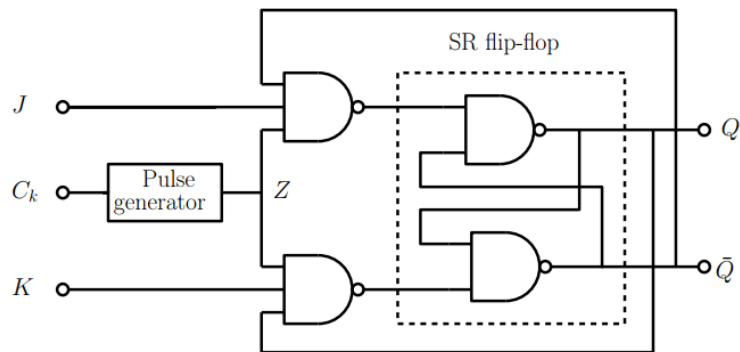
edge-triggered



positive edge-triggered input

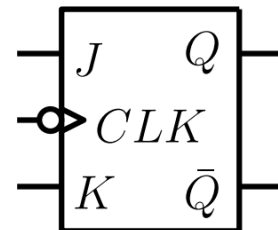
JK триггер

Синхронизация по заднему фронту



| C _k | J | K | Q | Q̄ | Mode |
|----------------|---|---|---|----|--------|
| 0 | X | X | Q | Q̄ | Hold |
| 1 | X | X | Q | Q̄ | Hold |
| ↑ | X | X | Q | Q̄ | Hold |
| ↓ | 0 | 0 | Q | Q̄ | Hold |
| ↓ | 0 | 1 | 0 | 1 | Reset |
| ↓ | 1 | 0 | 1 | 0 | Set |
| ↓ | 1 | 1 | Q | Q | Toggle |

edge-triggered



negative edge-triggered input

Регистр сдвига – это набор триггеров, соединенных таким образом, чтобы хранящиеся в триггерах биты сдвигались влево или вправо, тем самым упрощая умножение или деление данных.



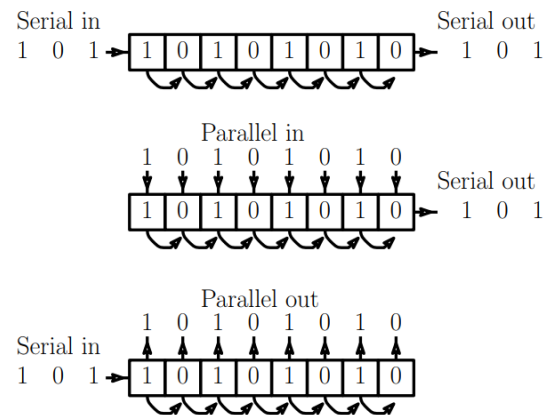
Слова данных, проходящие через цифровую систему, часто должны временно удерживаться, копироваться и сдвигаться по битам влево или вправо. Устройством, которое может быть использовано для таких приложений, является сдвиговый регистр.

Регистр сдвига



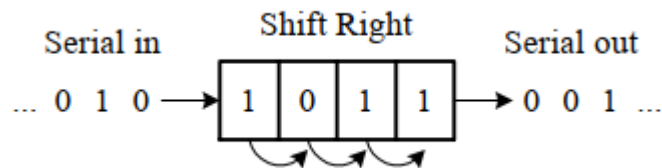
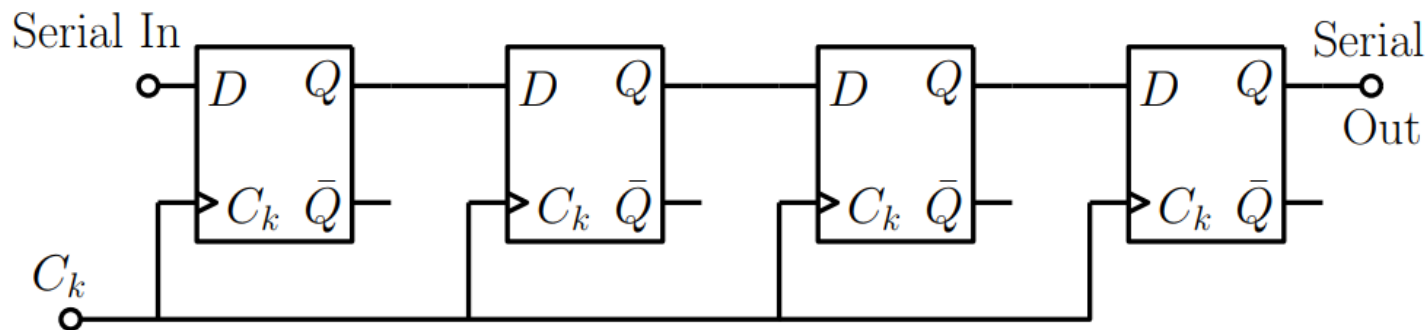
Различают регистры сдвига 4-х типов:

1. *С последовательным входом/параллельным выходом (Serial-in/Parallel-out – SIPO)* – данные в последовательном коде загружаются по одному биту в каждом такте и одновременно доступны в параллельном коде на выходе.
2. *С последовательным входом/последовательным выходом (Serial-in/Serial-out – SISO)* – данные вводятся в регистр сдвига и выводятся из него по одному биту в каждом такте.
3. *С параллельным входом/последовательным выходом (Parallel-in/Serial-out – PISO)* – все биты данных загружаются одновременно, но сдвигаются по одному биту в каждом такте.
4. *С параллельным входом/последовательным выходом (Parallel-in/Serial-out – PIPO)* – все биты данных загружаются одновременно и сдвигаются вместе.

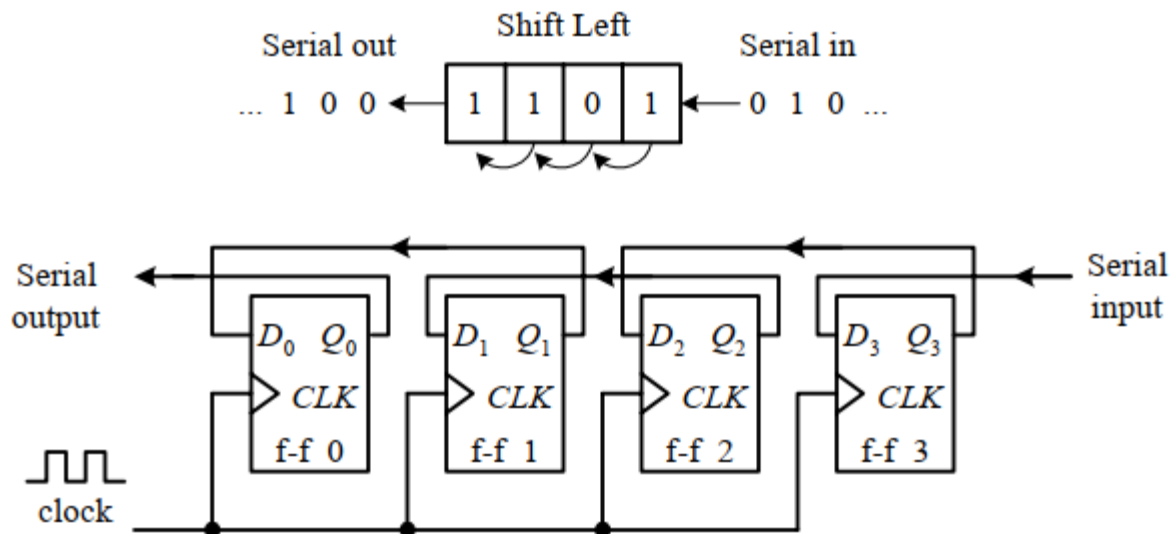


Регистр сдвига

Serial-In/Serial-Out Shifter Registers

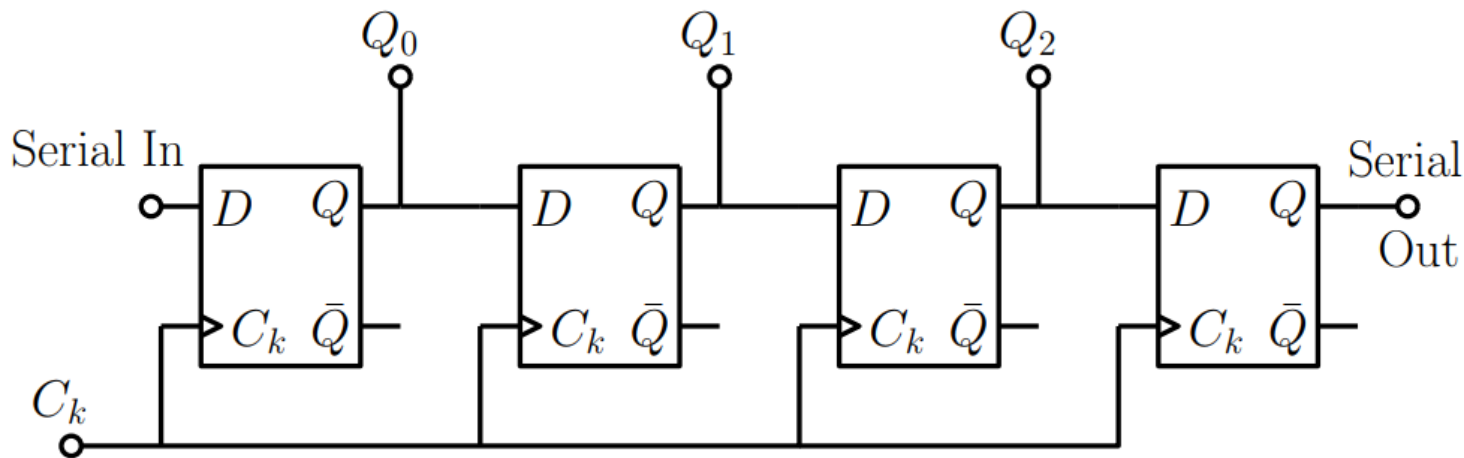


Регистр сдвига

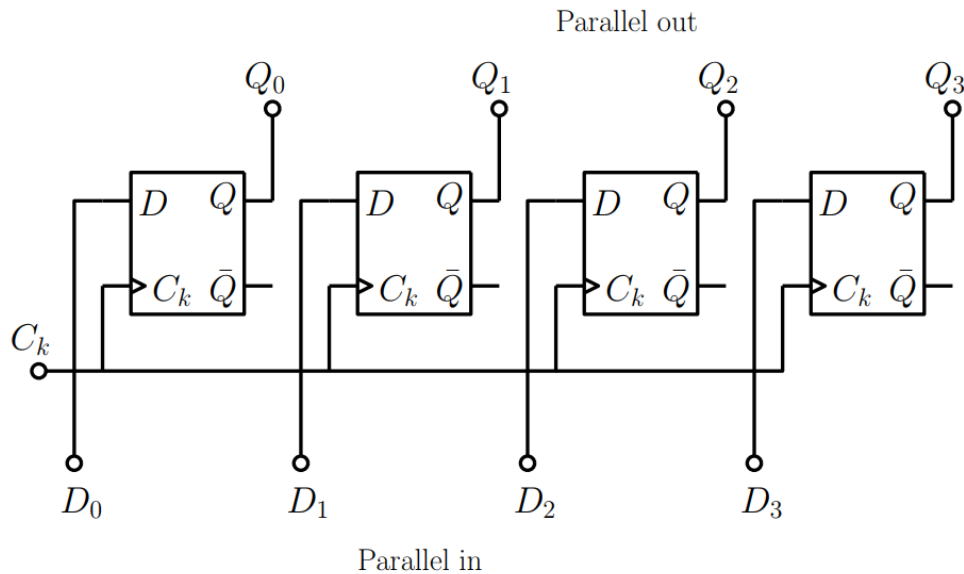


Регистр сдвига

Serial-In/Parallel-Out Shift Registers



Parallel-In/Parallel-Out Shift Registers



Регистр сдвига можно использовать в качестве счетчика, поскольку данные сдвигаются для каждого тактового импульса.



Счетчик - это регистр, который проходит через заданную последовательность состояний при приеме входных импульсов.

Кроме того, компьютеры, таймеры, частотомеры и различные другие цифровые устройства содержат счетчики для подсчета событий.

По способу формирования внутренних связей счетчики делятся на:

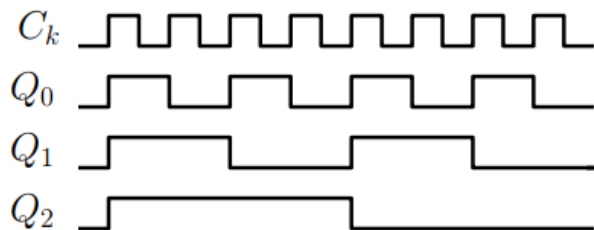
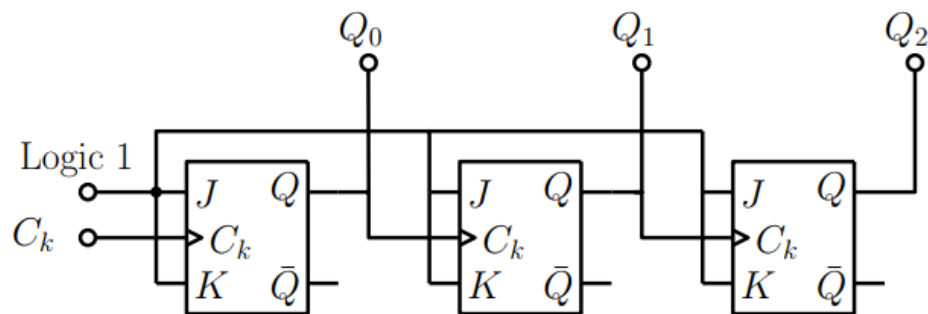
- с последовательным переносом;
- с параллельным переносом;
- с комбинированным переносом;
- кольцевые.

Классификация по модулю счета:

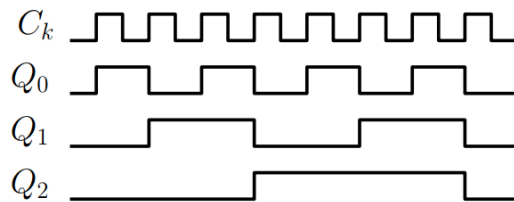
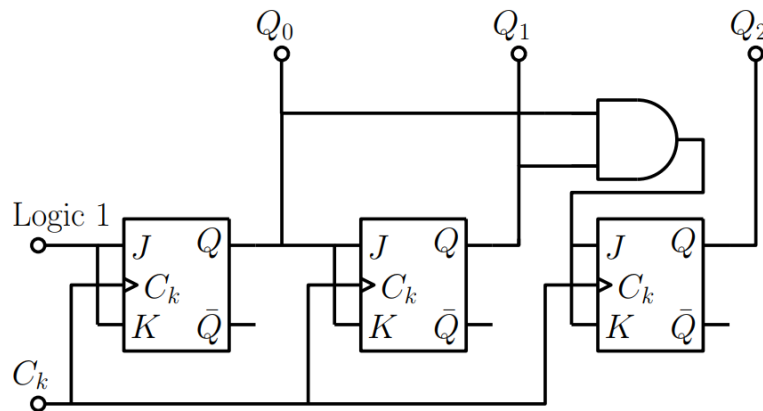
- двоично-десятичные;
- двоичные с произвольным постоянным модулем счета;
- с переменным модулем счета;
- по направлению счета:
- Суммирующие;
- Вычитающие;
- Реверсивные.



Счетчик импульсов (асинхронный счетчик)

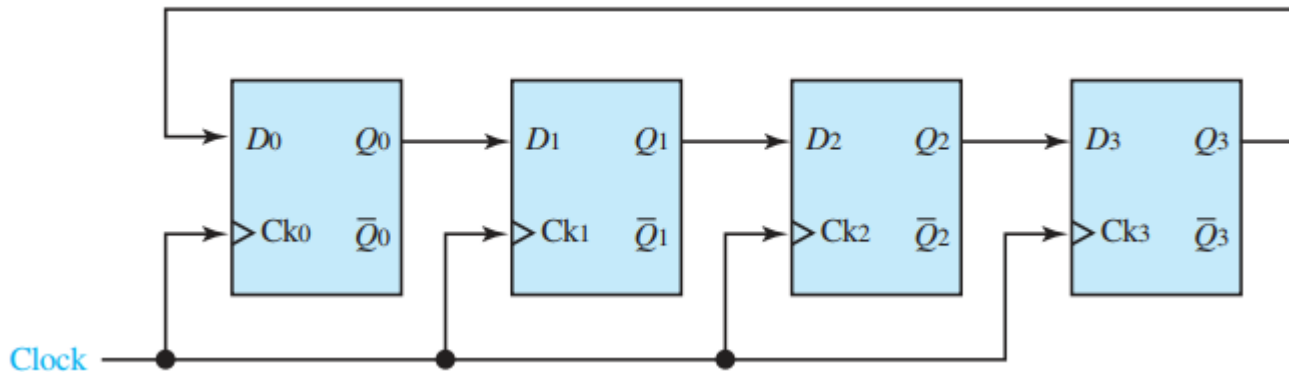


Синхронный счетчик



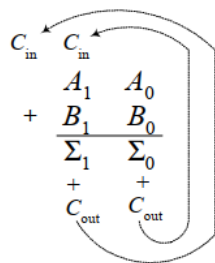
Циклический сдвиговый регистр

Кольцевой счетчик - это уникальный тип сдвигового регистра, который включает обратную связь с выхода последнего триггера на вход первого триггера.



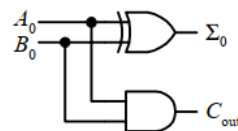
Двоичный сумматор

Mechanics of adding

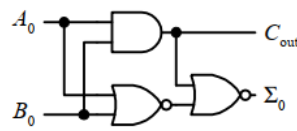


| MSB addition | | | LSB addition | | |
|--------------|-------|----------|--------------|-----------|--|
| A_1 | B_1 | C_{in} | Σ_1 | C_{out} | |
| 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 0 | |
| 0 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 1 | |
| 1 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 0 | 1 | |
| 1 | 1 | 1 | 1 | 1 | |

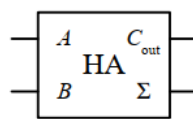
Half-adder



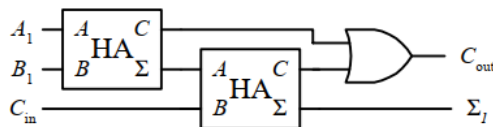
alternative representation



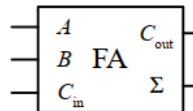
half-adder symbol



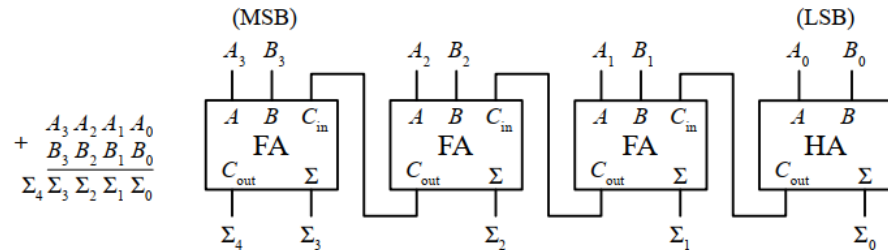
Full-adder



full-adder symbol



4-bit adder

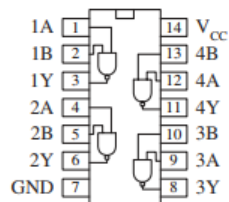


Цифровые логические ИС



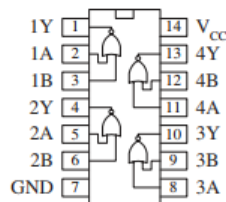
74HC00 Series

Quad 2-input
NAND



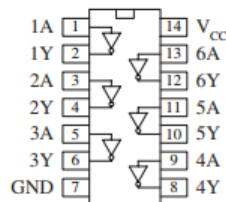
7400, 74LS00,
74F00, 74HC00, etc.

Quad 2-input
NOR



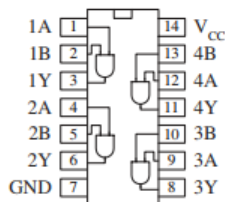
7402, 74LS02,
74F02, 74HC02, etc.

Hex Inverter



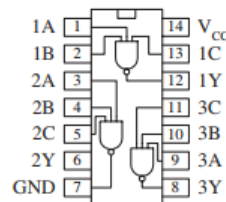
7404, 74LS04,
74F04, 74HC04, etc.

Quad 2-input
AND



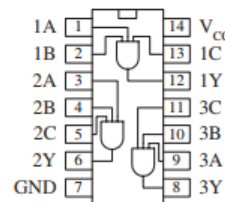
7408, 74LS08,
74F08, 74HC08, etc.

Triple 3-input
NAND



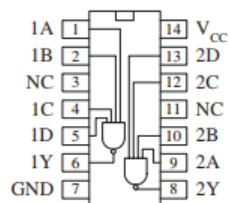
7410, 74LS10,
74F10, 74HC10, etc.

Triple 3-input
AND



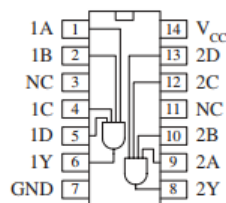
7411, 74LS11,
74F11, 74HC11, etc.

Dual 4-input
NAND



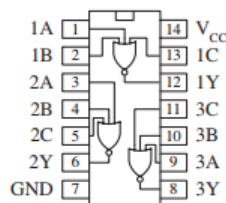
7420, 74LS20,
74F20, 74HC20, etc.

Dual 4-input
AND



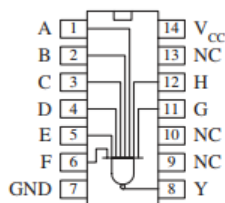
7421, 74LS21,
74F21, 74HC21, etc.

Triple 3-input
NOR



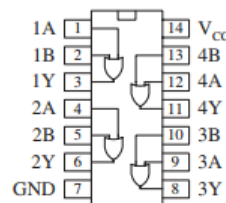
7427, 74LS27,
74F27, 74HC27, etc.

8-input NAND



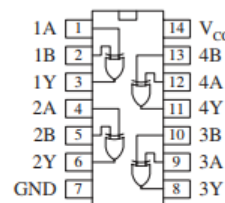
7430, 74LS30,
74F30, 74HC30, etc.

Quad 2-input
OR



7432, 74LS32,
74F32, 74HC32, etc.

Quad 2-input
XOR

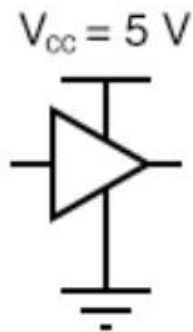
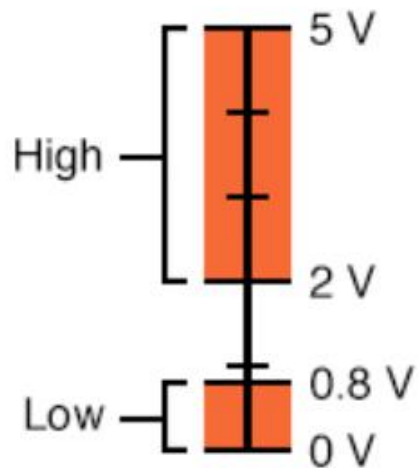


7486, 74LS86,
74F86, 74HC86, etc.

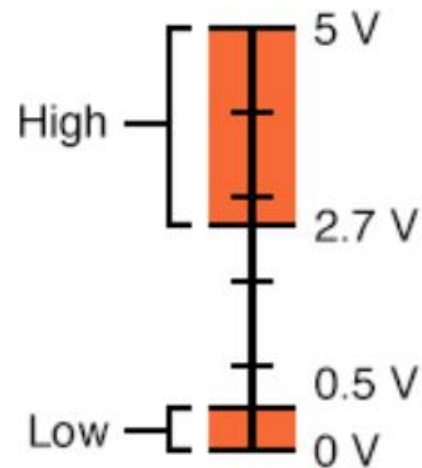
Уровни логических сигналов



Acceptable TTL Gate
Input Signal Levels

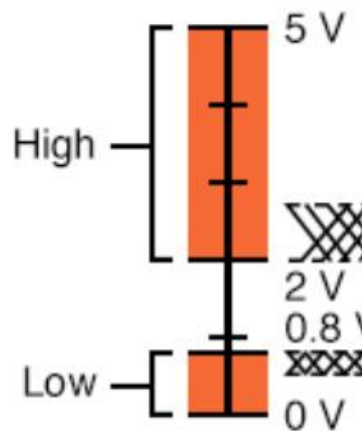


Acceptable TTL Gate
Output Signal Levels

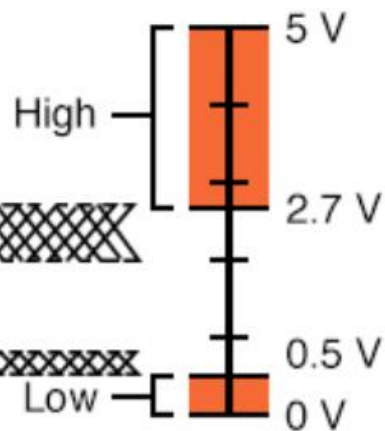


Уровни логических сигналов

Acceptable TTL Gate
Input Signal Levels



Acceptable TTL Gate
Output Signal Levels

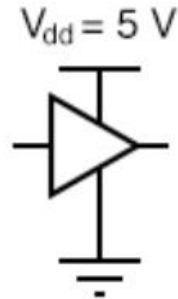
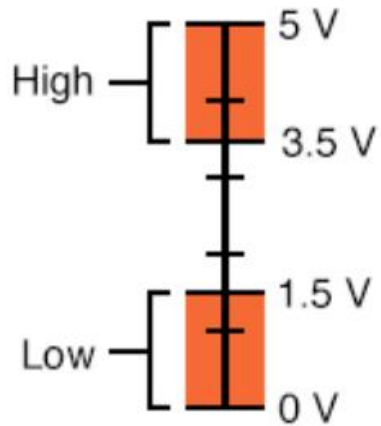


high-level noise margin

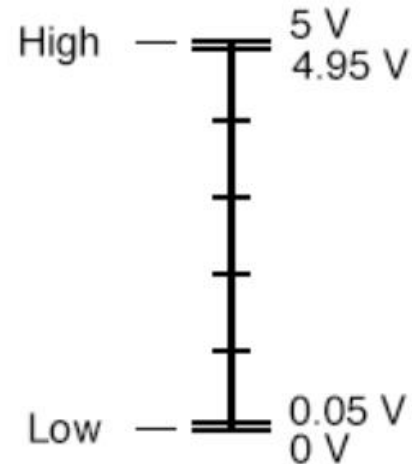
low-level noise margin



Acceptable CMOS Gate
Input Signal Levels

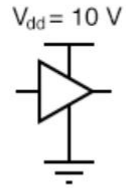
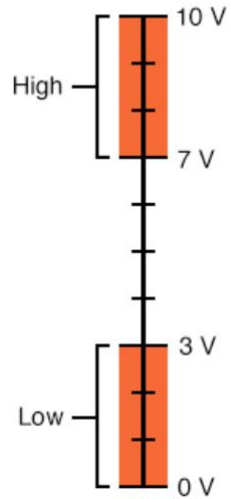


Acceptable CMOS Gate
Output Signal Levels

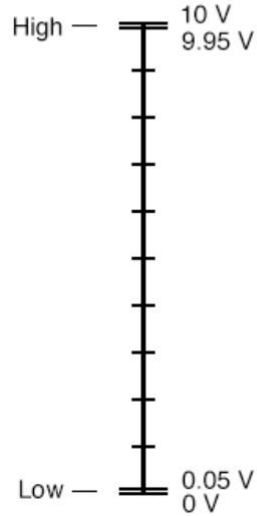




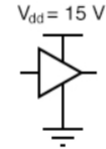
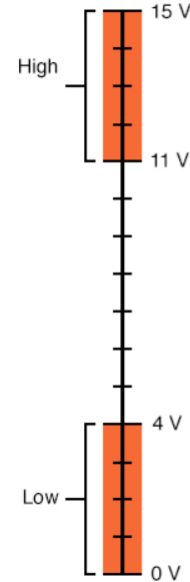
Acceptable CMOS Gate
Input Signal Levels



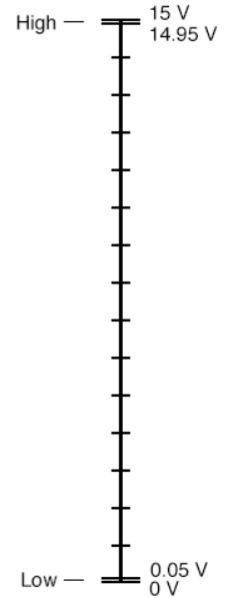
Acceptable CMOS Gate
Output Signal Levels



Acceptable CMOS Gate
Input Signal Levels



Acceptable CMOS Gate
Output Signal Levels



Список использованных источников

1. Sarma M. S. Introduction to electrical engineering. – New York : Oxford University Press, 2001. – С. 715-716.
2. Tokheim R. L. Digital Electronics: Principles and Applications, 8th Edition. – McGraw-Hill, Inc., 2014.
3. Kleitz W. Digital Electronics: A practical approach with VHDL. – Prentice Hall, 2011.
4. Harris S., Harris D. Digital design and computer architecture: arm edition. – Morgan Kaufmann, 2015.
5. Paul Scherz, Simon Monk. Practical Electronics for Inventors, Fourth Edition. - McGraw-Hill, Inc., 2016.

<https://circuitverse.org/simulator>

<https://learn.circuitverse.org/docs/seq-msi/counters.html>

**Спасибо
за внимание!**

it^{'s}**MO** *re than a*
UNIVERSITY

Nikolay.A.Nikolaev@gmail.com