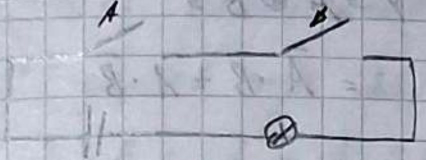
Логические функции.

И (AND)

1)



2)

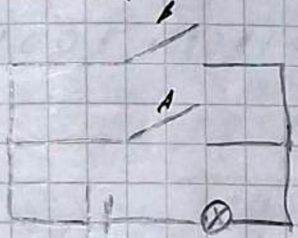
3)

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

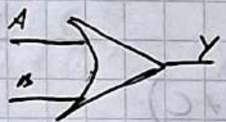
4) $Y = A \cdot B.$

ИЛИ (OR)

1)



2)



3)

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

4) $Y = A + B.$

NE (NOT)

2)

3)

A	Y
0	1
1	0.

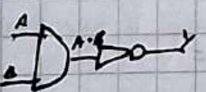
4) $Y = \overline{A}$

NI (NAND)

negacija

2)

или



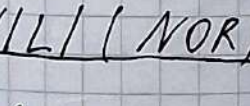
3)

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

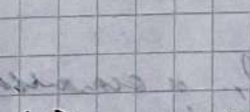
4) $Y = \overline{A \cdot B}$

NILI (NOR)

2)



или

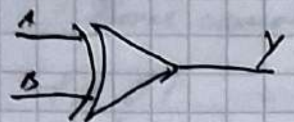


3)

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0.

4) $Y = \overline{A + B}$

XOR

2) 

3)

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0.

4) $Y = A \oplus B =$

$= \bar{A} \cdot B + A \cdot \bar{B}$

NXOR - как $\overline{XOR}$

Boolean algebra.

Имя	AND	OR.
Identitet	$1 \cdot A = A$	$0 + A = A$
Nul-pravilo	$0 \cdot A = 0$	$1 + A = 1.$
Idenpotencija	$A \cdot A = A$	$A + A = A$
Inverzni	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1.$
Komutativnost	$A \cdot B = B \cdot A$	$A + B = B + A$
Asociativnost	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$
Distributivnost	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C.$
Apsorpcija	$A \cdot (A + B) = A$	$A + A \cdot B = A.$
Demorganov	$\overline{A \cdot B} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$

Пример:

$$(A+B) \cdot (A+C) = \underbrace{A \cdot A}_{A} + A \cdot C + A \cdot B + B \cdot C =$$

$$= \underbrace{A + A \cdot C}_{A \text{(apsorpcija)}} + A \cdot B + B \cdot C =$$

$$= \underline{A + A \cdot B} + B \cdot C = A + B \cdot C.$$

(!) Можно запоминать только AND, а остальное менять только $0 \rightleftarrows 1$ и знаки + и · местами.

Sum of Products и Products of sums

Порой чтобы вычислить в условии, нужно написать их формулу, и для этого есть методы SOP и POS.

Допустим, мы имеем:

A	B	C	F (результат)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

~~SOP~~

POS (maxterm)

Вычисляем по нулям
(product of sums)

$$\boxed{m \begin{cases} 0 \Leftrightarrow A \\ 1 \Leftrightarrow \bar{A} \end{cases}}$$

1)

	A	B	C	F
m_0	0	0	0	0
m_1	0	0	1	0
m_2	0	1	0	1
m_3	0	1	1	0
m_4	1	0	0	1
m_5	1	0	1	1
m_6	1	1	0	1
m_7	1	1	1	1

2) $F(A,B,C) = M_0 + M_1 + M_3$ - velika slova

3) Преобразуем

$$F(A,B,C) = \underbrace{(A+B+C)}_{M_0} \cdot \underbrace{(A+B+\bar{C})}_{M_1} \cdot (A+\bar{B}+\bar{C})$$

maxterm — сумма всех входов функции или инверсии

$(A+B+C)$ и $(A+B+\bar{C})$... - maxterms

→ В этой функции 3 maxterma

POS — который член maxterm

SOP. (minterm)

Вычисляем по единицам.
(sum of product).

$$\boxed{m \begin{cases} 0 \Leftrightarrow \bar{A} \\ 1 \Leftrightarrow A \end{cases}}$$

1)

	A	B	C	F
m_0	0	0	0	0
m_1	0	0	1	0
m_2	0	1	0	1
m_3	0	1	1	0
m_4	1	0	0	1
m_5	1	0	1	1
m_6	1	1	0	1
m_7	1	1	1	1

2) $F(A,B,C) = m_2 + m_4 + m_5 + m_6 + m_7$

3) Преобразуем.

$$F(A,B,C) = \overset{m_2}{\bar{A}\cdot B\cdot \bar{C}} + \overset{m_4}{A\cdot \bar{B}\cdot \bar{C}} + \overset{m_5}{A\cdot \bar{B}\cdot C} + \overset{m_6}{A\cdot B\cdot \bar{C}} + \overset{m_7}{A\cdot B\cdot C}$$

minterm - произведение всех входов функции или инверсий.

$(\bar{A}\cdot B\cdot \bar{C})$ и $(A\cdot \bar{B}\cdot \bar{C})$ - minterms

→ В этой функции 5 minterma

SOP - который член minterm.

Стандартная форма (Standardna Forma)

Форма, без сокращений

$$F(A,B,C)_{SOP} = \boxed{\bar{A}\cdot B\cdot \bar{C} + A\cdot \bar{B}\cdot \bar{C} + A\cdot \bar{B}\cdot \bar{\bar{C}} + A\cdot B\cdot \bar{C} + A\cdot B\cdot C}$$

Минимальная форма (Minimalna Forma)

(Stand. F → min. F.)

Форма, после сокращения, используя разл. законы (стр. 30)

$$F(A,B,C)_{SOP} = \bar{A}\cdot B\cdot \bar{C} + A\cdot \bar{B}\cdot \bar{C} + A\cdot \bar{B}\cdot \bar{\bar{C}} + A\cdot B\cdot \bar{C} + ABC =$$

$$= \bar{A}\cdot B\cdot \bar{C} + A\cdot \bar{B}\,\cancel{(C\cdot \bar{C})} + A\cdot B\cdot \cancel{(C\cdot \bar{C})} =$$

$$= \bar{A}\cdot B\cdot \bar{C} + A\cdot \bar{B} + A\cdot B = \bar{A}\cdot B\cdot \bar{C} + A\,\cancel{(B\cdot \bar{B})} =$$

$$= \bar{A}\cdot B\cdot \bar{C} + A \overset{distrib.}{=} (A+\bar{A})\cdot (A\cdot B\cdot \bar{C}) =$$

$$= \boxed{A\cdot B\cdot C}$$ minimalna forma.

(min. F.) → (Stand. F)

Нам нужно из minimal формы вернуть её в исходное состояние.

$$Y = \underset{m_1}{A} + \underset{m_2}{B\bar{C}}$$

1) Переменные: 3: A, B и C.

2) m_1: A ✓, B ×, C × ; m_2: A ×, B ✓, C ✓

3) По SOP:

$$Y = A + B\bar{C} = A\cdot \overset{B\cdot\bar{B}}{1}\cdot \underset{C\cdot\bar{C}}{1} + \overset{A\cdot\bar{A}}{1}\cdot B\cdot \bar{C}$$

(!) minterms и maxterms мы считаем только в stand. forme.

$$Y = A\cdot (B\cdot \bar{B})(C\cdot \bar{C}) + (A\cdot \bar{A})\cdot B\cdot \bar{C} =$$

$$= (A\cdot B + A\cdot \bar{B})(C\cdot \bar{C}) + AB\bar{C} + \bar{A}B\bar{C}$$

$$= ABC + \underbrace{AB\bar{C}}_{повтор} + A\bar{B}C + A\bar{B}\bar{C} + \underbrace{\cancel{AB\bar{C}}}_{повтор} + \bar{A}B\bar{C}$$

4) Убираем повторы.

$$= ABC + \underbrace{AB\bar{C}}_{остаётся} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C}$$ <=> Standardna forma. (5 minterms)

(!) Также и с POS, но вместо ~~умножение~~ умножение сложение, и вместо 1 мы пишем 0.

$$Y = A + B\bar{C} = (A+0+0)\cdot(0+B+\bar{C})$$

(!) Чтобы узнать максимальное количество возможных maxterm/minterm'ов, то используем формулу:

$\boxed{2^n = mint/maxt}$, где n - кол-во переменных.

Например, при 3 переменных A, B, C, макс. возможно:

$2^3 = 8$ возможных min/max термов.

КАРНАУГХОВА ТАБЛИЦА.
КАРТА КАРНО
Karnaughova karta.

Карта карно (Karnaugh map) – графический способ представления булевых функций с целью их удобной и наглядной ручной минимизации.

1) Так мы имеем таблицу. — minterm

	~~A~~	B	C	F
m_0	0	0	0	1
m_1	0	0	1	0
m_2	0	1	0	1
m_3	0	1	1	0
m_4	1	0	0	1
m_5	1	0	1	1
m_6	1	1	0	1
m_7	1	1	1	1

2) Перерисуем её в карту Карно

A \ BC	00	01	11	10
0	1_0	$_1$	$_3$	1_2
1	1_4	1_5	1_7	1_6
11				
10				

tražimo implikante.

3) Находим импликанты (implikante):

Implikanti je svaki brojeva u tabele, koji imaju jednu grupu

(!) Импликанты – любые числа в таблице, объединённые в одну группу.

Каждый импликант состоит из 1; 2; 4; 6 ....

В таблице: $\boxed{1}$: 6 $\boxed{2}$: 7 $\boxed{4}$: 2.
импликанта:

Simple Implicants

Простые импликанты: Импликанты, которые группируют, и группа может состоять из ~~части~~ частей других (но не целиком).

	00	01	11	10
0	1	1		
1		1	1	

⇓

A \ BC	00	01	11	10
0	1			1
1	1	1	1	1

→ 2 простых импликанта.

не может быть, принадлежит к 4-клет. имп.

ПОДСКАЗКА:

Простые импликанты относительно друг друга ~~рассчитыва~~ рассчитываем.

Эссенциальные импликанты мы считаем относительно всех.

Esencial implikants

Эссенциальные импликанты: такие, как и простые, но всегда должен быть хоть 1 свободный член,

	00	01	11	10
0	1	1		
1		1	1	

Таким образом мы считаем единичные импликанты:

A \ BC	00	01	11	10
0	1			1
1	1	1	1	1

II

I

$$F = \mathrm{I} + \mathrm{II} = A + \cancel{B}\bar{C}$$

Далее мы смотрим, что у нас меняется

Например:

A \ BC	00	01	11	10
0	1	1		1
1	1		1	

мы возьмём

A \ BC	00
0	1
1	1

, где BC не меняется, а A ме

и взаимно уничтожаются, поэтому пишем $\bar{B}\bar{C}$

Don't care — это означает, что мы можем ставить всё на свой выбор

		1	1
1	1	X	X

=>

		1	1
1	1		

или

		1	1
1	1	1	1

Будут разные решения, но нам всё равно (don't care)

(!) Лучше ставить единицы.

maxterm

MAxterm

Maxterm'ы мы находим также, как и minterm, но наоборот.
Так вместо $0 = \bar{A}$, мы пишем $0 = A$
И считаем мы нули.

A \ BC	00	01	11	10
0	1	0	0	1
1	1	1	1	1

$= A + \bar{C}$ (вместо умножения плюс)

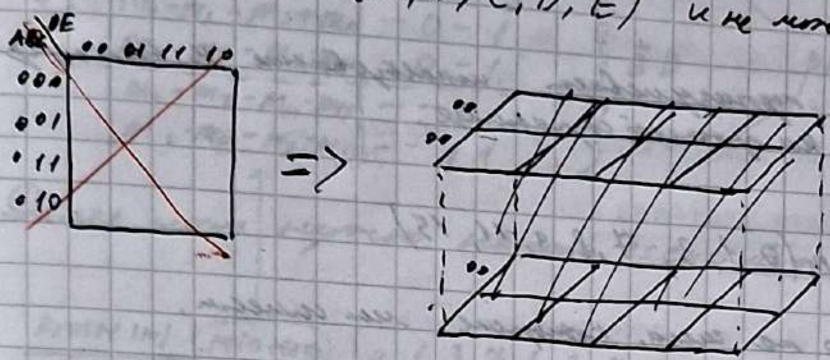
A \ BC	00	01	11	10
0	0	0		
1			0	0

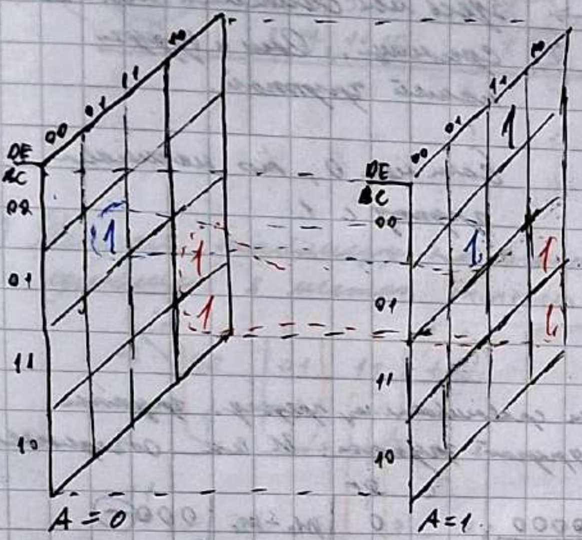
$= (A + B)^{\mathrm{I}} \cdot (\bar{A} + \bar{B})^{\mathrm{II}}$

(!) Maxterm пишем с большой буквы M ($M_1, M_2, M_3, \dots, M_n$).

K'map с 5 переменными.

1) Мы имеем $F(A,B,C,D,E)$ и не можем уместить в таблицу





Мы накладываем одну таблицу на другую для расчёта.

Мы берём одну переменную и его разделяем таблицу на 2 таблицы: с $A=0$ и с $A=1$.

Синий: у нас в одной таблице 1 расположен в 01:01, и в другой. Это 1 импликант

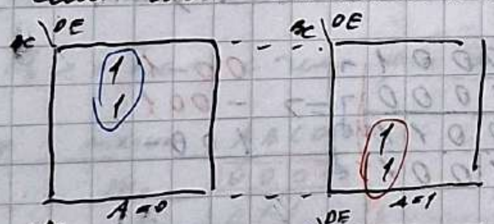
$\bar{B}E$ (A нет)

Красный: у нас 2 в одной, и 2 в другой на одном месте, а значит это 1 импликант

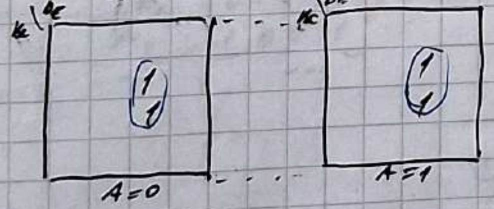
$\Rightarrow D\bar{E}B$

Чёрный: у нас только в одной таблице, а значит $01/\boxed{11}$ $A=1$ $= A\overline{BC}DE$

① Мы берём вместе как 1 импликант из 2 таблиц, только если они НА ОДНОЙ ПОЗИЦИИ.



$\Rightarrow$ тут 2 импликанта, а не один.



$\Rightarrow$ тут 1 импликант из четырёх единиц.

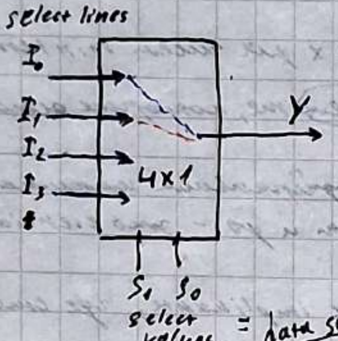
Мультиплексеры.

Kombinaciona mreža - logička mreža čiji izlaz zavisi samo od ulaza i ne zavisi od ...

Imamo: MUX, DeMUX, koder, dekoder.

Multiplexer = MUX = data selector.

Multiplexer - механизм выбора чего-либо по точкам входа.



= data selector.

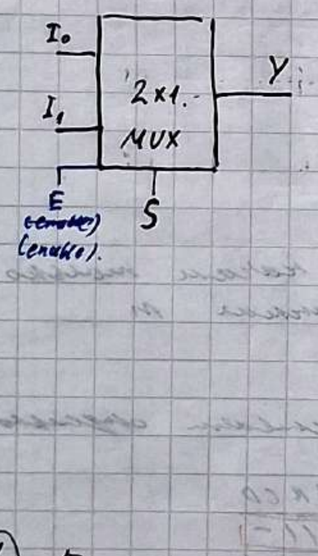
=>

S_0	S_1	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

MUX $\underbrace{4}_{input}\times\underbrace{1}_{output}$

(!) В стандартном MUX $n = 2^{m}$ — n: select lines, m: select values

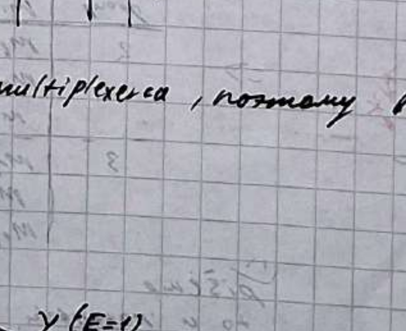
MUX 2x1:



E	S	Y
0	x	0
1	0	I_0
1	1	I_1

E - включение/выключение схемы. Если схема выключена, то остальные переменные X (don't care)

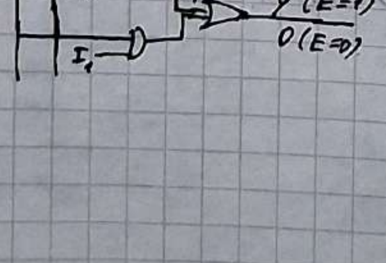
Схема:



Формула:

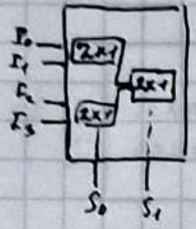
$$Y = E\bar{S}I_0 + ESI_1$$

(!) E - отвечает за on/off multiplexerа, поэтому в конце просто приписываем к схеме.



Синтезирование ~~мульт~~ мультиплексора $(4x1) \Rightarrow (2x1)$

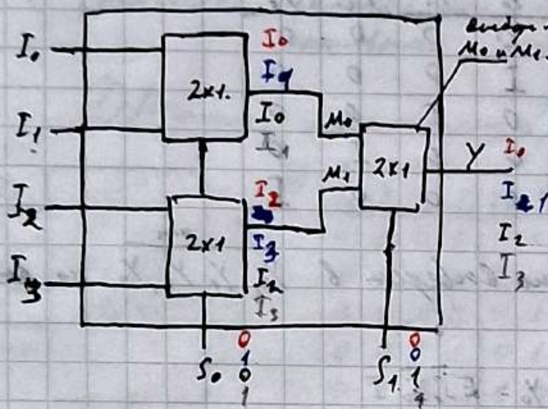
менее схематично.



, где S_0 - выбор I_0 или $I_1 = M_0$
I_2 или $I_3 = M_1$

S_1 выбор M_0 или M_1 (между мультиплексорами)

расширенная версия:



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

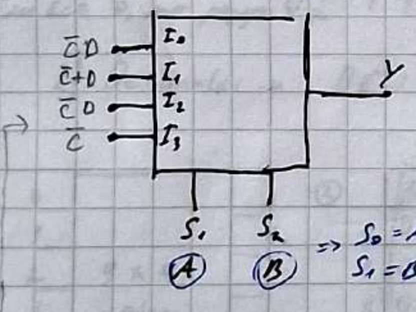
(!) Также и с $8x1 \Rightarrow 2x1$. и т.д.

MUX и булевая функция.

1) $F(A,B,C,D) = \Sigma m(1,4,5,7,9,12,13)$

2) Сколько переменных, столько и входов $ABCD \Rightarrow 4x1$. MUX

3) Записать MUX и выбрать 2 переменные (лучше первые 2 AB)



$\Rightarrow S_0 = A$
$S_1 = B$

Осталось найти I_0, I_1, I_2, I_3

3) Рисуем таблицу по Карно (стр. 37) совмещая с S_0 и S_1 (A и B).

S_1S_0 AB \ CD	00	01	11	10
00		1		
01	1	1	1	
11	1	1		
10		1		

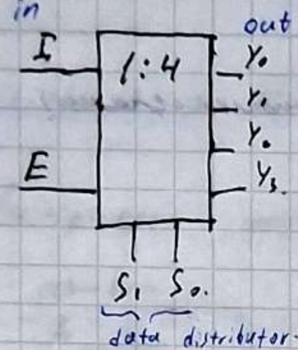
$I_0 = \bar{C}D$
$I_1 = \bar{C}\bar{D} + \bar{C}D$

S_1	S_0	Y
0	0	$I_0 = \bar{C}D$
0	1	$I_1 = \bar{C}\bar{D} + \bar{C}D = \bar{C}+D$.
1	0	$I_2 = \bar{C}D$
1	1	$I_3 = \bar{C}$

т.е. берём во внимание только ряд, который мы смотрим в S_1 и S_0.

DEMUX - (demultiplexer) - демультиплексоры.

Выполняет функцию, обратную мультиплексору. Коммутирует один входной канал на несколько выходных.



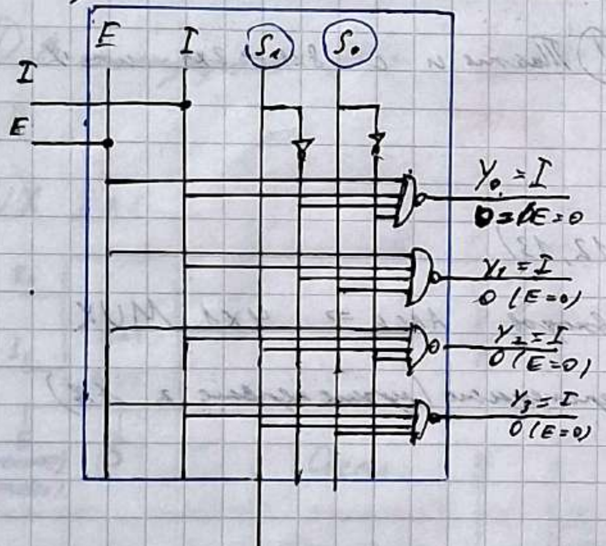
$n = 2^{m}$ - ddt distributor

output

Пример с таблицей:

	E	S_1	S_0	Y_0	Y_1	Y_2	Y_3
isklučen. —	0	x	x	0	0	0	0
	1	0	0	I	0	0	0
	1	0	1	0	I	0	0
	1	1	0	0	0	I	0
	1	1	1	0	0	0	I

Следовательно, что выберем в I, то выведем в Y_0, Y_1, Y_2, Y_3 по выбору через S_1 и S_0 data distributor.



$Y_0 = E\bar{S_1}\bar{S_0}I$

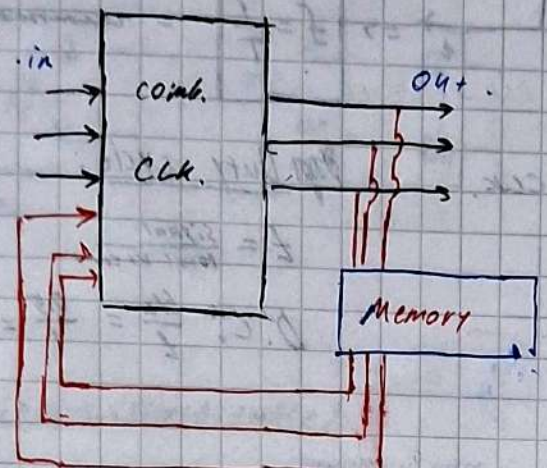
$Y_1 = E\bar{S_1}S_0 I$

$Y_2 = ES_1\bar{S_0}I$

$Y_3 = ES_1S_0I$.

Sekvencijalna mreža

Sekvencijalna mreža - это та же сеть, как и kombinaciona (стр. 42), но с большим отличием - мы имеем память.



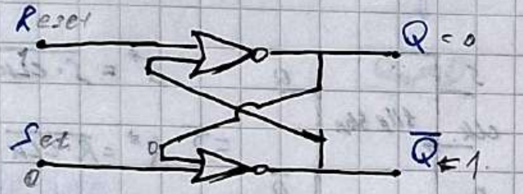
Например, counter:

memory

0 + 1 = 1
1 + 1 = 2
2 + 1 = 3
3 + 1 = 4
4 + 1 = 5
...

memory - это и ~~есть~~ есть flip flop.

SR Latch. - базовый storage элемент, который сохраняет 1 бит. → NILI → NI



Случай 1: S=0; R=1, то Q=0, а потом $\bar{Q}$=1.
↓ память
~~проект 2~~: S=0; R=0; Q=0; $\bar{Q}$=1

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

reset, reset

not used, not save.

Случай 2: S=1; R=0; $\bar{Q}$=0, то Q=1.
↓ память
S=0; R=0; Q=1. $\bar{Q}$=0

Случай 3. S=1; R=1; Q=0; $\bar{Q}$=1.
✗ S≠1 и R≠1.
S=0; R=0; Q=0; $\bar{Q}$=1.

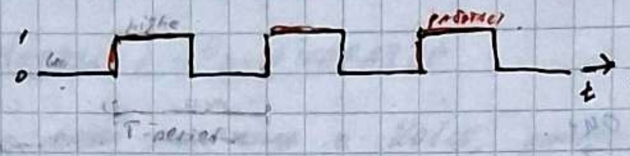
S	R	Q	$\bar{Q}$
0	0	memory.	
0	1	0	1
1	0	1	0.
1	1.	not used	

Для NI цепи точно так же, но немного наоборот Q и $\bar{Q}$.

S	R	Q	$\bar{Q}$
0	0	~~memory~~ not used	
0	1	1	0
1	0	0	1
1	1	~~not used~~ memory.	

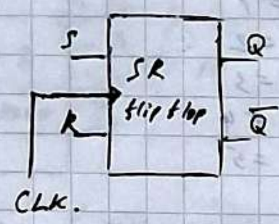
Clock.

flip flop работает с определённой тактовой частотой (Clock или Signal)



=> $\boxed{f = \frac{1}{T}}$ — частота. f - frekvencija [Hz], T = vreme [s]

Поэтому к схеме приписываем CLK.



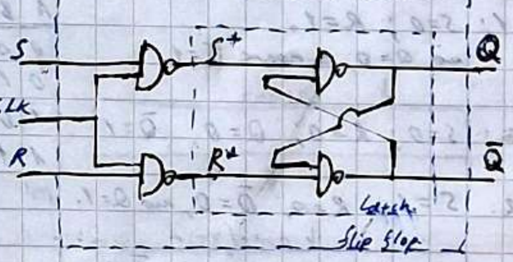
Duty cycle:

$t = \frac{signal}{total\ vreme}$

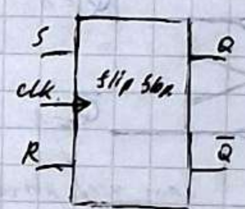
$D.C.: \frac{t/2}{t} = \frac{0,5}{1} = 0,5 = 50\%$.

Разница flip/flop и Latch, SR flip flop

flip flop зависит от Clock, а latch - это просто схема для ~~работоспособности~~ работоспособности. Latch - это часть flip flop'а.



=>



$S^* = \overline{S \cdot CLK} = \bar{S} + \overline{CLK}$

=> $R^* = \overline{R \cdot CLK} = \bar{R} + \overline{CLK}$

Из этого мы можем сделать таблицу.

CLK	S	R	Q	$\bar{Q}$
0	x	x	memory.	
1	0	0	memory.	
1	0	1	0	1
1	1	0	1	0
1	1	1	not used.	

Tablica istinosti (truth table)

CLK	S	R	Q_{n+1} (next state)
0	x	x	Q_n ← (present state)
1	0	0	Q_n ← (memory)
1	0	1	0
1	1	0	1.
1	1	1	not used.

=> characteristic table.

Q_n	S	R	Q_{n+1}
0	0	0	0 (memory) next state = present state
0	0	1	0
0	1	0	1
0	1	1	x
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	x

CLK=1

=> excitation table.

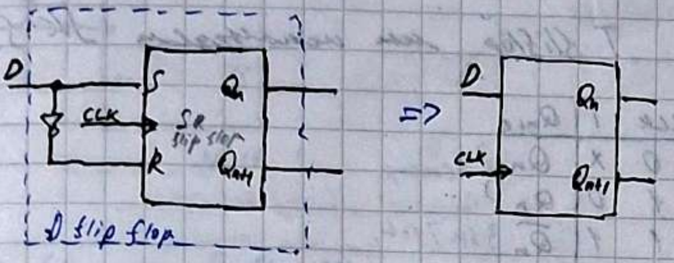
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Q_n \ SR	00	01	11	10
0	0	0	x	1
1	1	0	x	1

=> $Q_{n+1} = I + II = S + Q_n\bar{R}$

D flip flop

Если мы хотим только сохранить значение, то достаточно одной переменной (D), где D-data. D flip flop мы строим из SR flip flop'a.



Т.к. на таблице 1 стр. 49 мы для сохранения используем только S и R.

CLK	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1

Tablica istinosti (truth table)

CLK	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1

=>

characteristic table.

CLK=1

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

⇓

$$\boxed{Q_{n+1} = D}$$

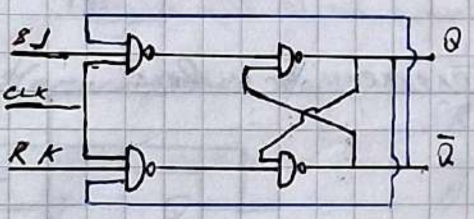
=>

excitation table.

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

JK flip flop

Решает проблему с not used. Он не имеет недопустимый вариант.



CLK	J	K	Q_{n+1}
0	x	x	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}_n$ (toggle)

} как SR flipflop

Tablica istinosti (truth table)

CLK	J	K	Q_{n+1}
0	x	x	Q_n } memory
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}_n$ - toggle.

=>

characteristic table.

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

⇓

Q_n \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

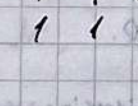
$$Q_{n+1} = I + II = \bar{Q}_n J + Q_n \bar{K}$$

=>

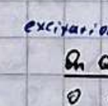
excitation table

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

⇓



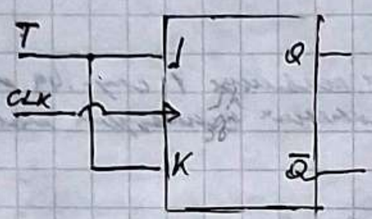
=> $J = Q_{n+1}$ } J



=> $K = \overline{Q_{n+1}}$ } K

T flip flop.

Используется только для переключения (switch.)



Для T fliflop мы используем JK ff

CLK	T	Q_{n+1}	
0	x	Q_n	memory
1	0	Q_n	
1	1	$\overline{Q_n}$	switch.

Tablica istinosti (truth table)

CLK	T	Q_{n+1}
0	x	Q_n
1	0	Q_n
1	1	$\overline{Q_n}$

$\Rightarrow$

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0.

$\Rightarrow$

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow$ $Q_{n+1} = Q_n \oplus T$

$T = Q_n \oplus Q_{n+1}.$

JK flip flop $\Rightarrow$ D flip flop.

① Identifikovati flip flop'ovi

Želeni : D Dostupan JK.

② Napisati karakterističnu tablicu za želeni flip flop.

Mi želemo D:

Q_n	D	Q_{n+1}	J	K
0	0	0	0	x
0	1	1	1	x
1	0	0	x	1
1	1	1	x	0.

← сопоставляем Q_n и Q_{n+1}

③ Pobudna tabela (excitation table) za dostupni flip flop.

Mi imamo JK:

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

④ Написать булевое выражение после сопоставления через Q_n и D для JK

будем значения JK

Q_n \ D	0	1
0		1
1	x	x

$J = D$

Q_n \ D	0	1
0	x	x
1	1	

$K = \overline{D}$