Методичка по ЛАБЕ 5

(Счетные элементы (СЭ) или элементы счетчиков)

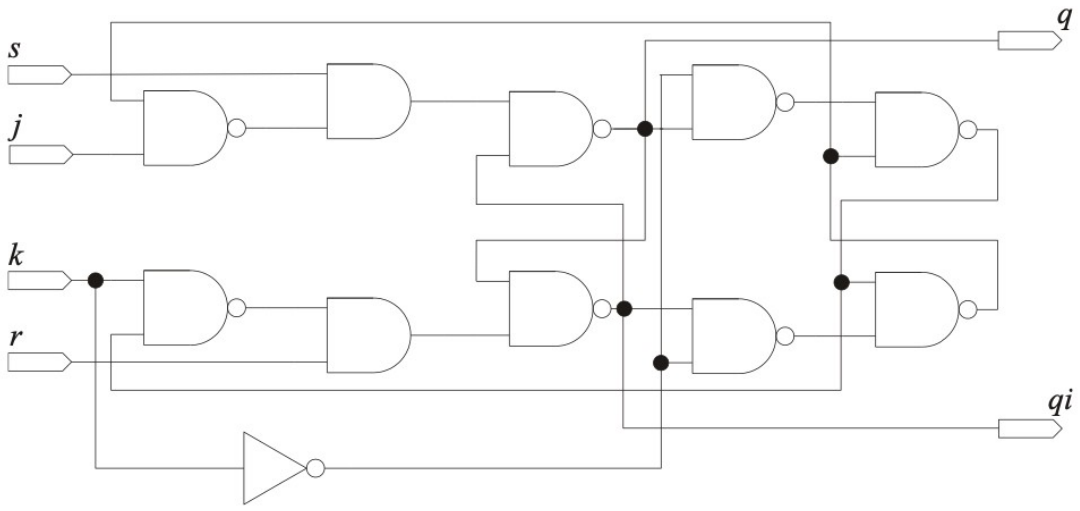


Рис. 10.16. Схема JK-RS-триггера

Таблица истинности jk-триггера практически совпадает с таблицей истинности синхронного RS-триггера. Для того чтобы исключить запрещённое состояние, его схема изменена таким образом, что при подаче двух единиц jk-триггер превращается в счётный T-триггер.

J=1 установка 1, K=1 установка 0, J=K=0 хранение, J=K=1 переключение в счетный режим.

R=0 установка в 0 (инверсный вход), S=0 установка в 1 (тоже инверсный вход).
R=S=1 – хз как это назвать, типа режим при котором входы j и k начинают работать(?),
R=S=0 – запрещенная комбинация. В методичке это названо защелкой.

Текущее состояние						Последующее состояние		Название режима
~S	~R	C	J	K	Qt	Q(t+dt)	~Q(t+dt)	
1	1	0,1,┌	X	X	Q	Q	~Q	Хранение инф-ии
		└	0	0	Q	Q	~Q	Хранение инф-ии
		└	1	0	X	1	0	Установка в "1"
		└	0	1	X	0	1	Установка в "0"
		└	1	1	Q	~Q	Q	Счет по модулю 2 (деление частоты вх. имп. на 2)
0	1		X	X	X	X	1	0	Установка в "1"
1	0		X	X	X	X	0	1	Установка в "0"
0	0		X	X	X	X	1	1	Неопред. сост-е

(не обращаем внимание на C в таблице, это тактовый вход, в нашем случае у нас его нет)

Так же важно что с помощью r и s можно устанавливать 0 и 1, и с помощью j и k можно устанавливать 0 и 1, однако r и s считаются более приоритетными.

Расписывать сильно не буду, ориентируйтесь на таблицу выше

Часть 1:

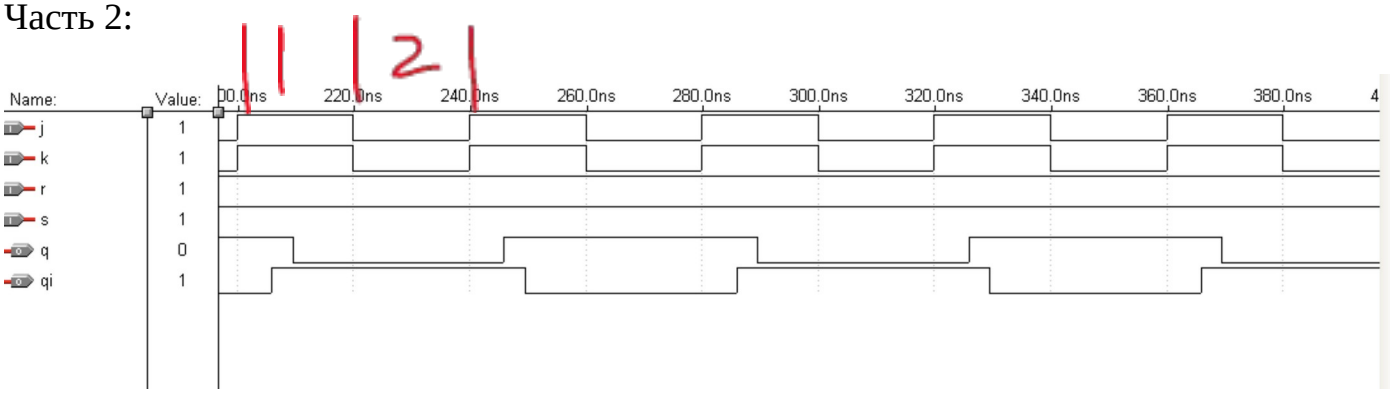
1) С помощью R и S устанавливаем 1 на прямой выход
2) Храним его (j=k=0)
3) Устанавливаем 0 (k=1)
4) Храним
5) Устанавливаем 1 (j=1)
6) Храним

Можно заметить что принцип работы такой же как у RS триггера

Не считая первый промежуток R и S входы ставим всегда = 1 (см табличку)

В начале нельзя ставить r=s=0 потому что неопределенное поведение, если ставить r=s=1 все ломается. Я поставила именно r=1 чисто от балды, можно и наоборот.

Часть 2:



Делаем j=k=1, счетный режим

Храним

Каждый импульс на прямом выходе q значение меняется с 1 на 0, следующий импульс с 0 на 1, и тп. Это называется счетный режим

Если дать j=k=1 сплошняком, то есть без промежутков (2), то счет не пойдет так как у нас нет тактового импулься и изменения импульсов не будет, импульс будет сплошной

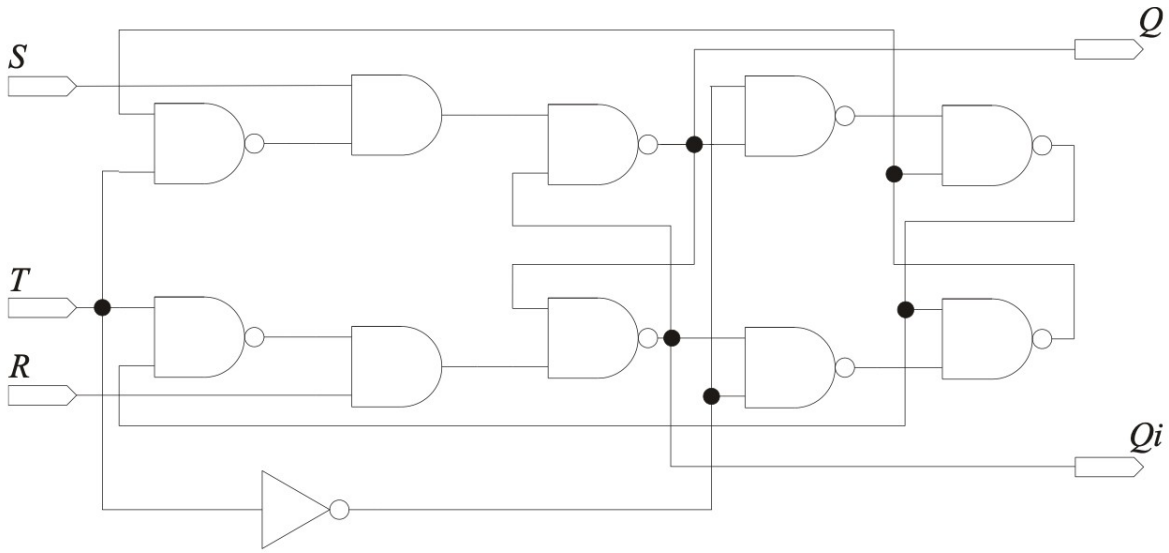
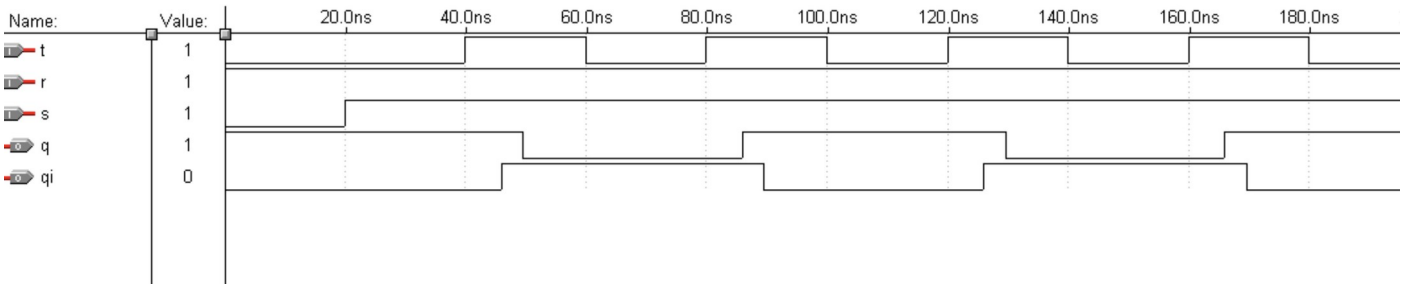


Рис. 10.17. Схема TRS-триггера

Короче то же самое что и до этого, только jk выходы склеены в t



См часть 2 в прошлой схеме

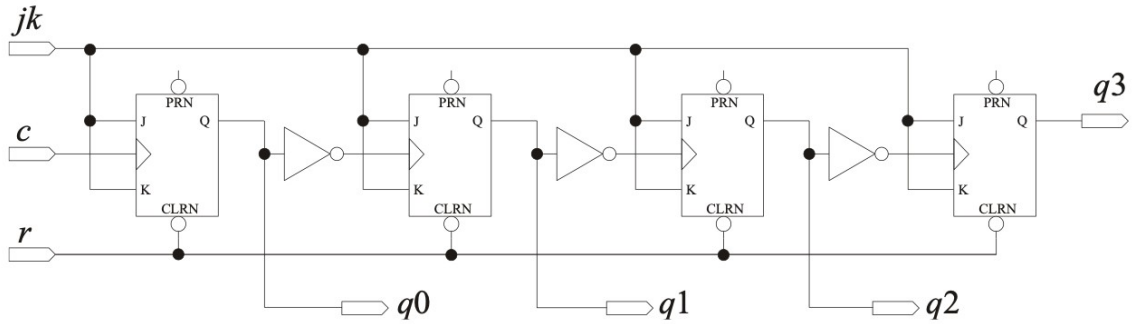


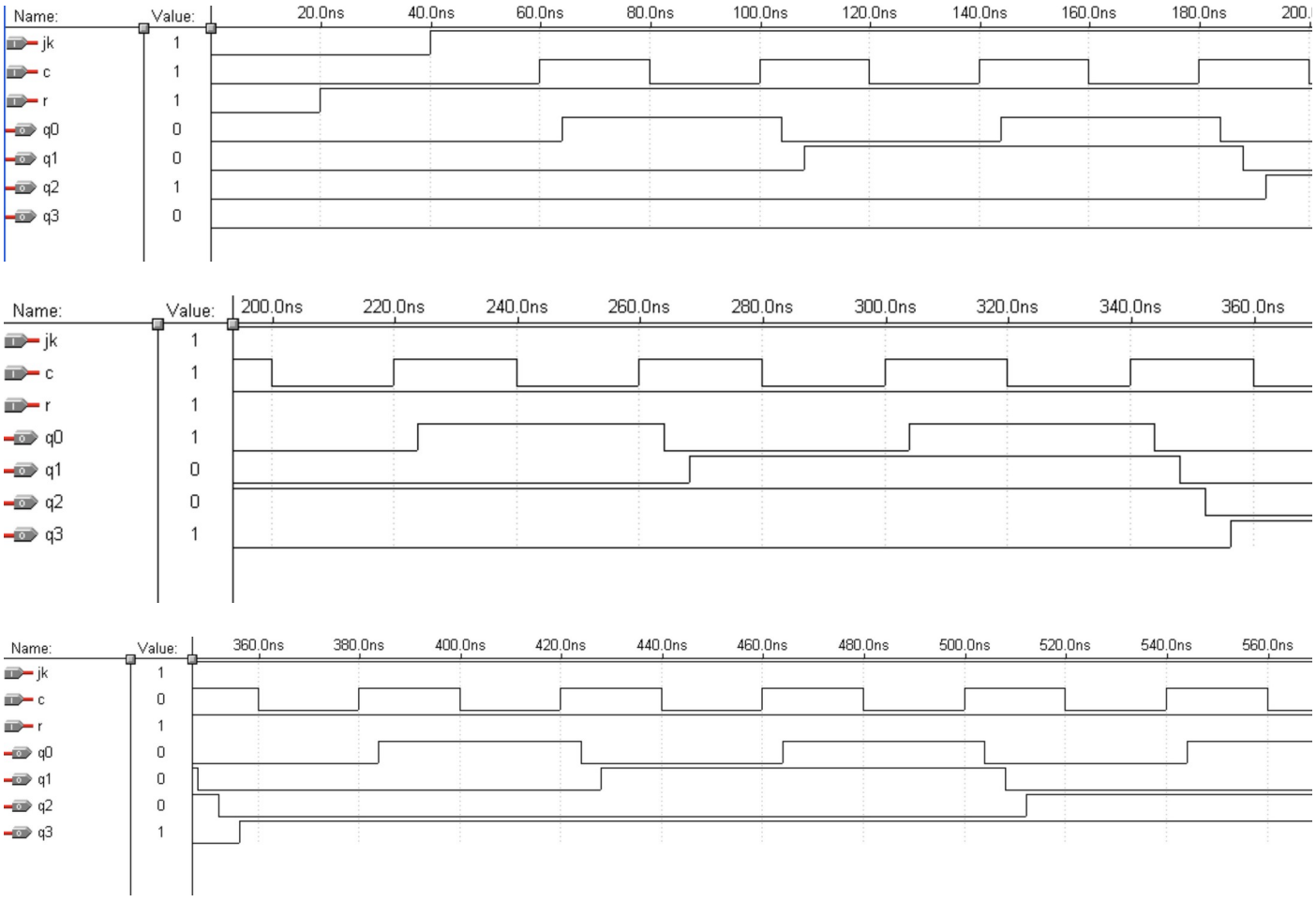
Рис. 10.18. Схема четырехразрядного счетчика

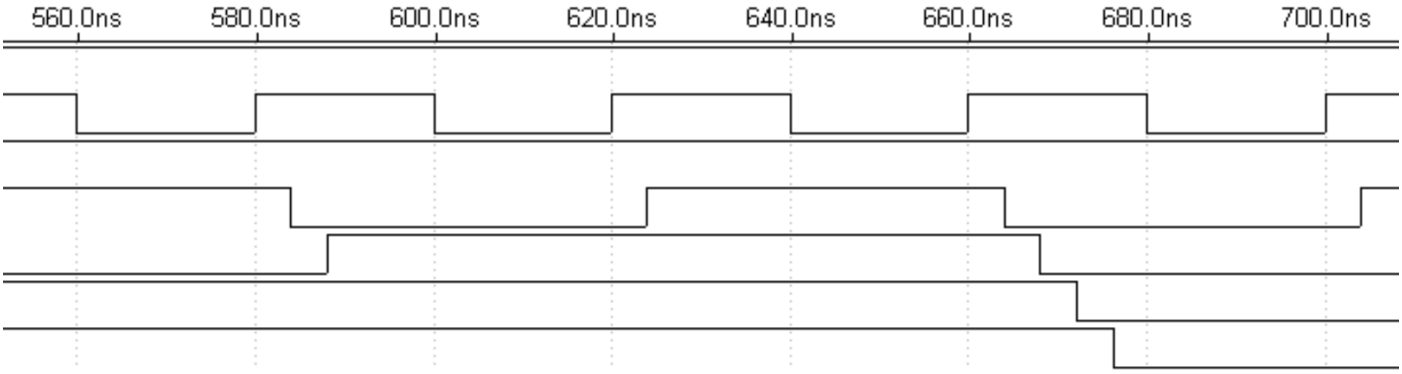
Очевидно из названия – считает до 4

R – то же самое что и в d триггере – ресет (инверсный)

C – тактовый импульс

Jk – данные

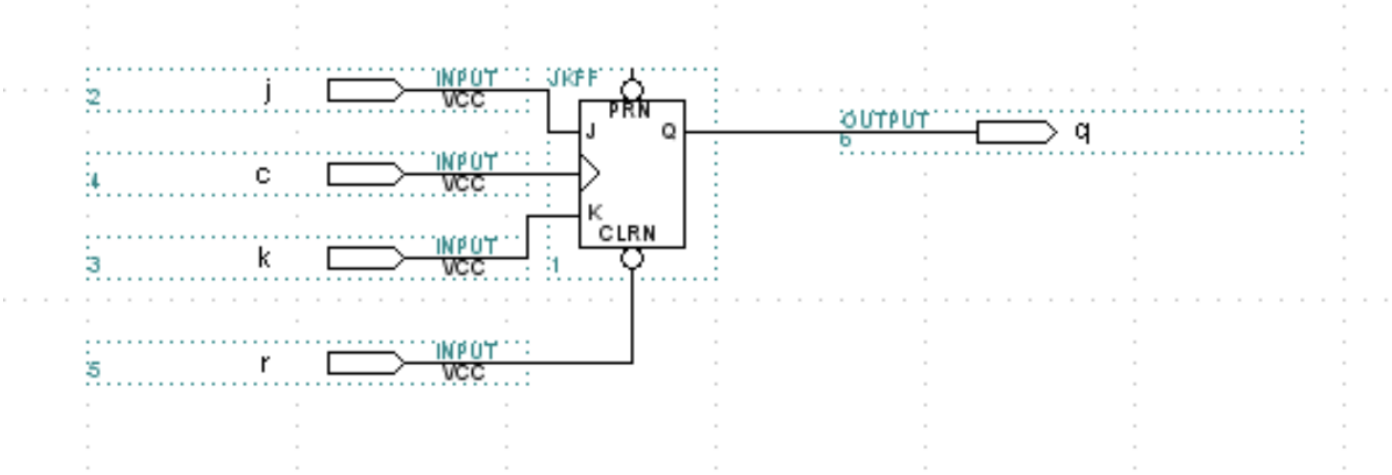




Выход q0 меняется каждый 1 тактовый сигнал, q1 каждые два сигнала, q2 каждые 3 сигнала, q3 каждые 4 сигнала

Таким образом он типа считает до 4

Jkff:



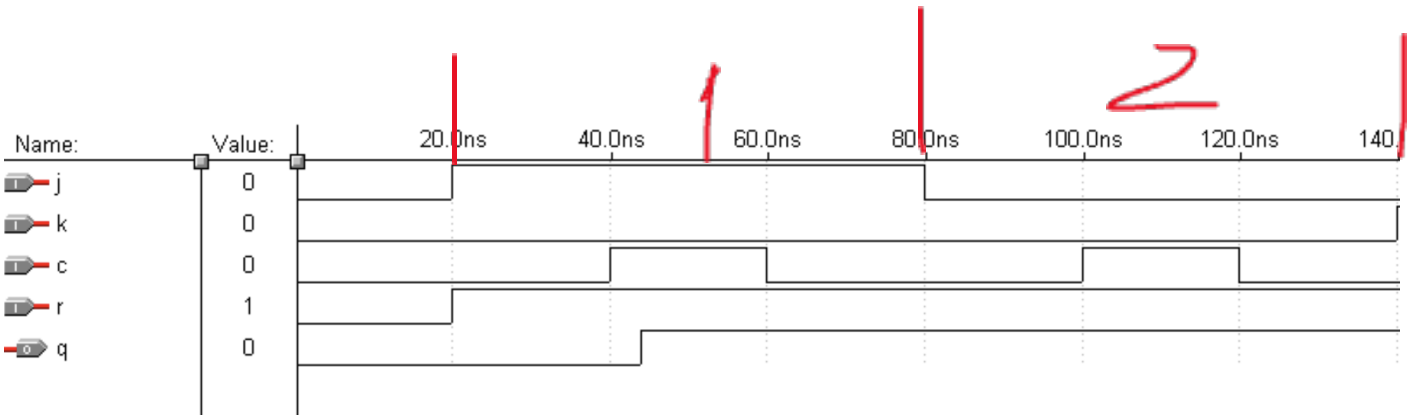
По сути то же самое что и первая схема, но:

1) Без входа s который вместе с r образуют защелку
2) С тактовым входом C, то есть теперь значения снимаются с каждым тактовым сигналом, а не с изменением сигнала на самих входах j и k как до этого

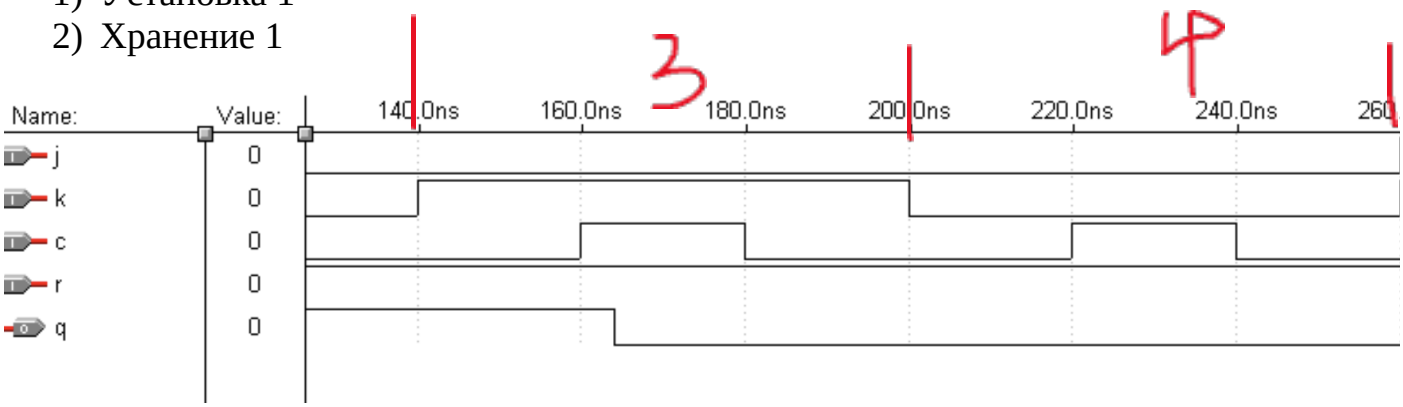
R – ресет, делаем все то же что и в прошлой схеме

C – тактовый вход

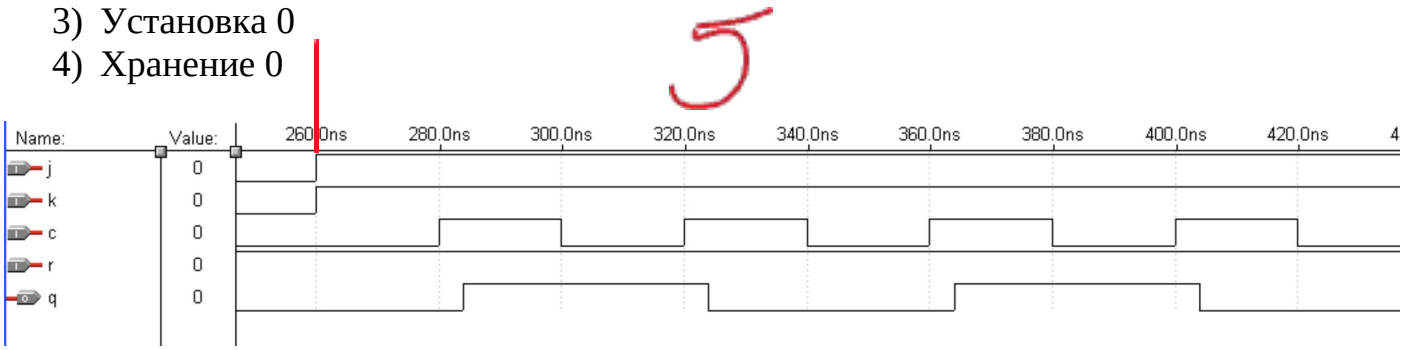
J=1 – установка 1, k=1 – установка 0, j=k=0 – хранение, j=k=1 -



1) Установка 1
2) Хранение 1

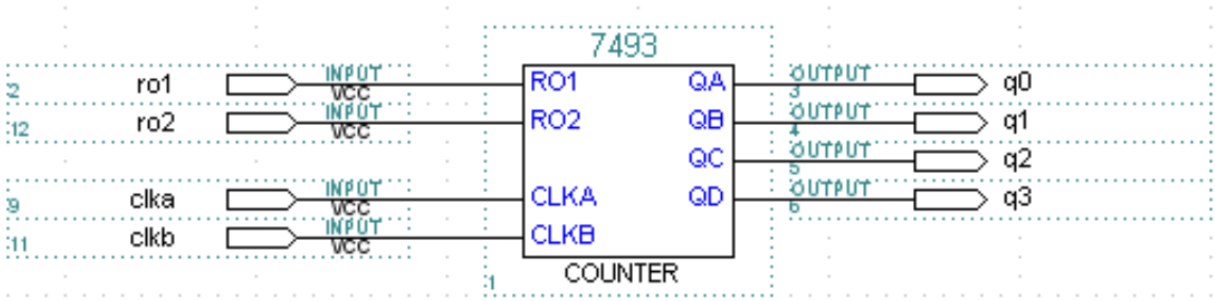


3) Установка 0
4) Хранение 0



5) Режим счетчика (т-триггер)

7493:



Такая же штука как Т-триггер, но есть два режима: как т-триггер считать по mod2 (то есть менять значение каждый тактовый сигнал), или по mod8 (то же самое, но меняет значения на трех выходах, на каждый 1 2 и 3 ий сигнал, по сути «умеет считать до 3х»)

ro1 и ro2 – ресет при 1

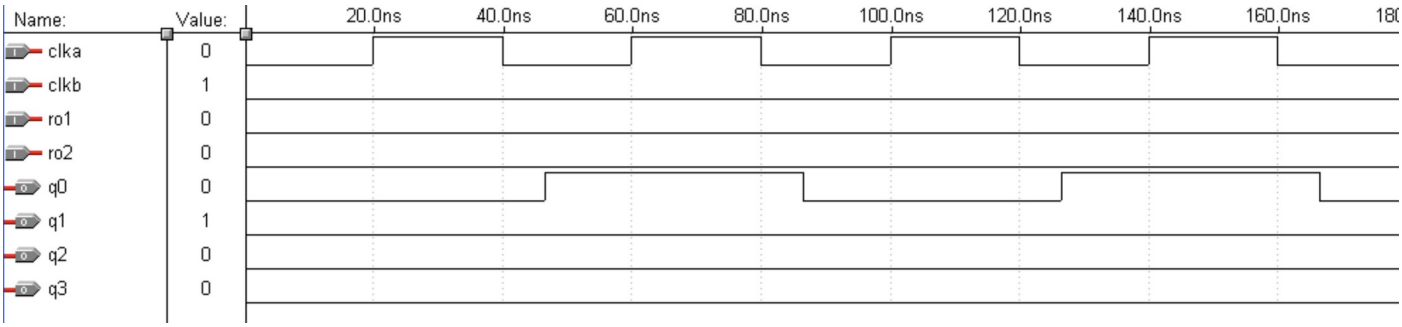
clka – вход для mod2

clkb – вход для mod8

q0 – выход для mod2

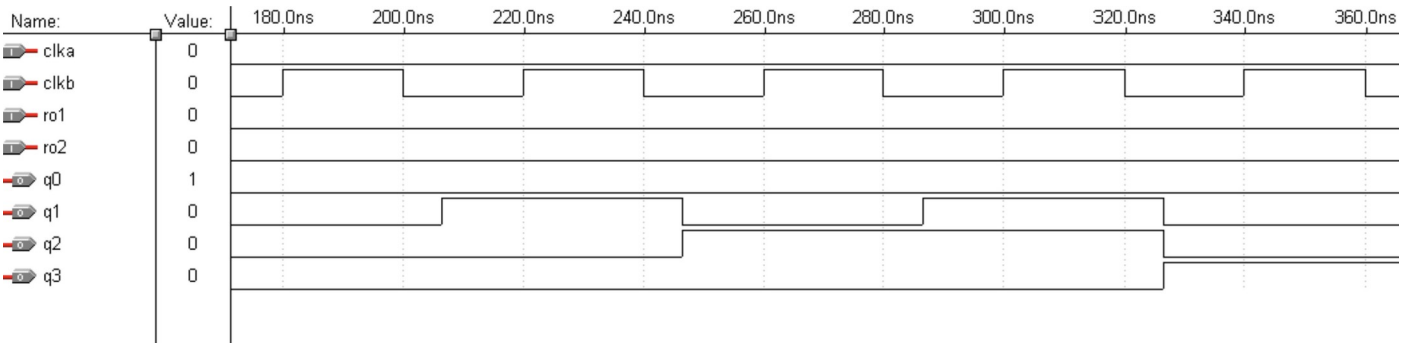
q1, q2, q3 - выход для mod8

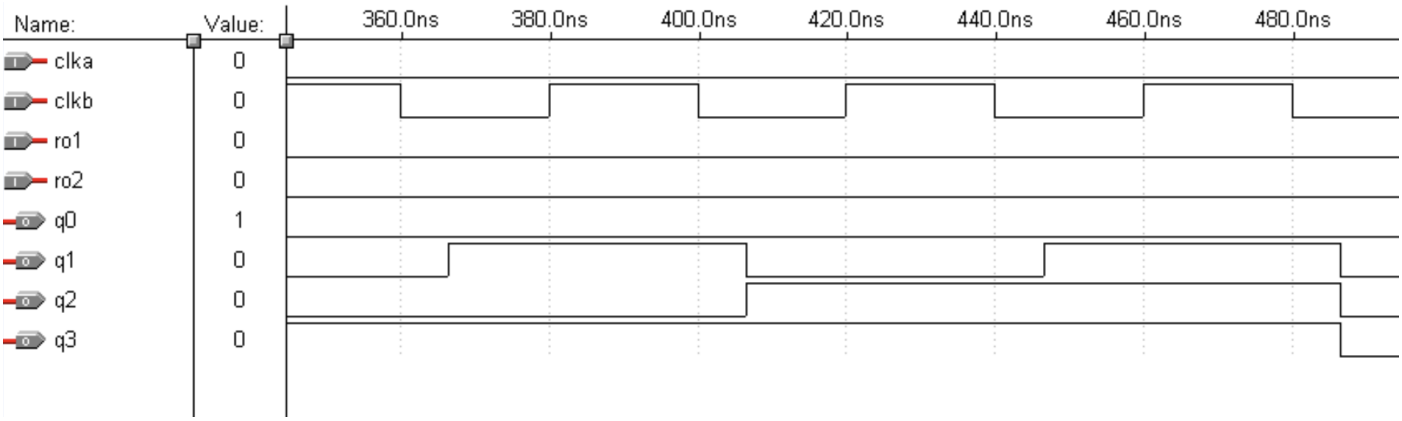
Mod2:



Все то же самое, каждый тактовый сигнал меняет значение

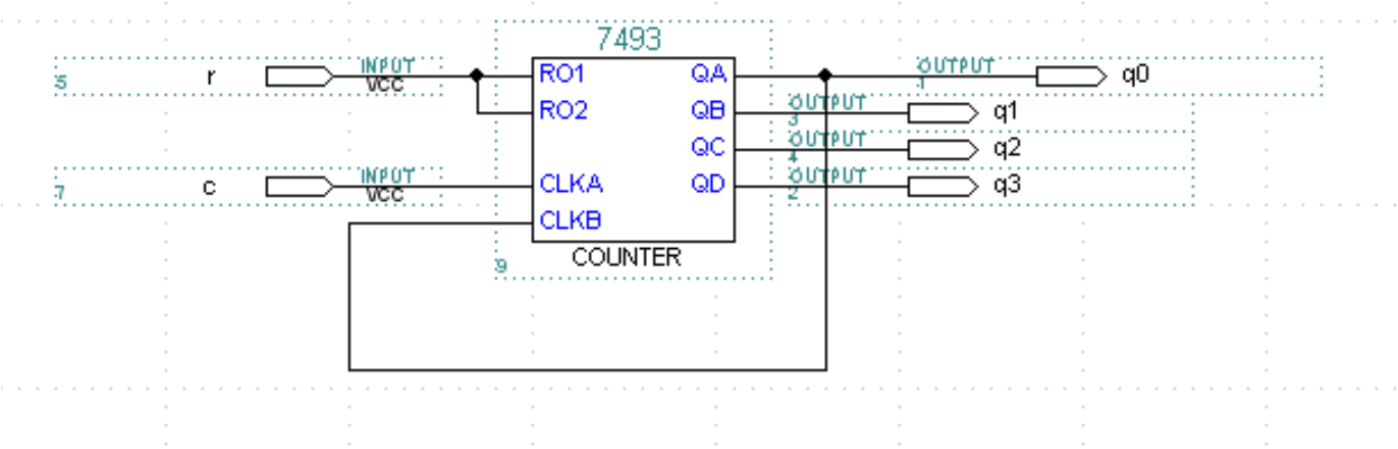
Mod8:





Опять же все то же самое: q0 меняется каждые 1 сигнала, q2 каждые 2, q3 каждые 3 сигнала

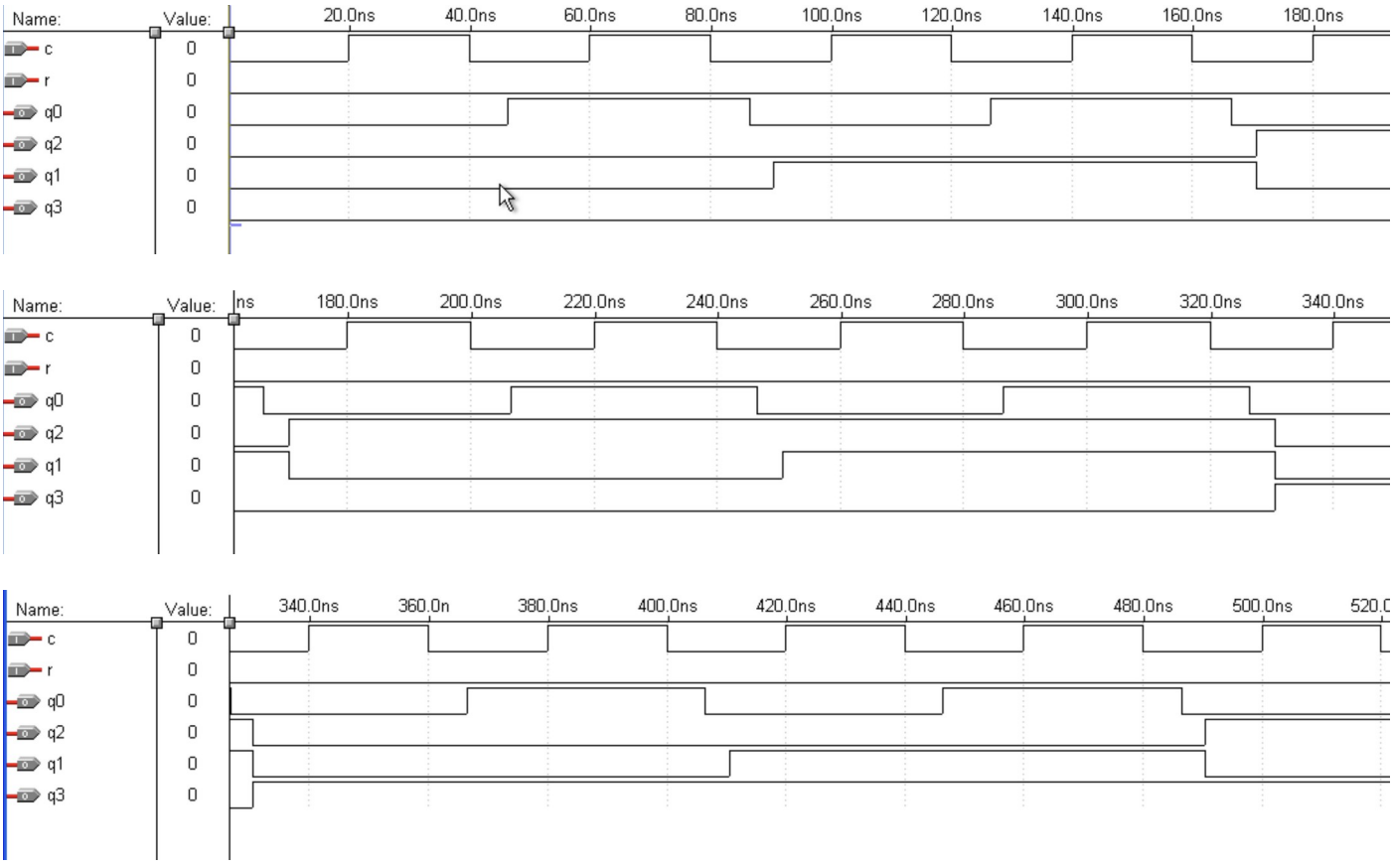
Схема для режима mod16:

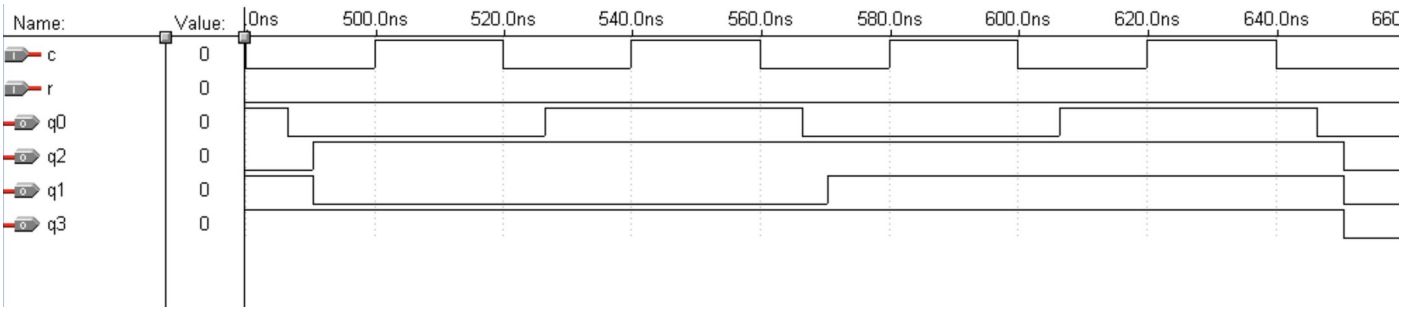


r– ресет при 1

c – вход для mod16

q0, q1, q2, q3 - выход для mod16





Меняются q0 – каждый 1 сигнал, q1 – каждые 2 сигнала, q2 – каждые 3 сигнала, q3 – каждые 4 сигнала

T триггер vhdl:

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY s2_t IS
    PORT
    (
        r   : IN    STD_LOGIC;
        s   : IN    STD_LOGIC;
        t   : IN    STD_LOGIC;
        q   : OUT   STD_LOGIC;
        qi  : OUT   STD_LOGIC
    );
END s2_t;

ARCHITECTURE behav OF s2_t IS
    SIGNAL  qs  : STD_LOGIC;
BEGIN
    PROCESS (t, r, s, qs)
    BEGIN
        IF (r = '0' and s = '0') THEN
            qs <= qs;
        ELSIF (r = '1' and s = '0') THEN
            qs <= '1';
        ELSIF (r = '0' and s = '1') THEN
            qs <= '0';
        ELSE
            IF (t'event and t = '1') THEN
                qs <= not qs;
            END IF;
        END IF;
    END PROCESS;

     q <= qs;
     qi <= not qs;
 END behav;
```

4x разрядный счетчик vhdl:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY s3_4counter IS

    PORT
    (
        jk, c, r : IN STD_LOGIC;
        q0, q1, q2, q3 : OUT STD_LOGIC
    );

END s3_4counter;

ARCHITECTURE behav OF s3_4counter IS
    SIGNAL  qs0, qs1, qs2, qs3  : STD_LOGIC;
BEGIN

    PROCESS (jk, c, qs0)
    BEGIN
        IF (c'event and c = '1') THEN
            qs0 <= not qs0;
        END IF;
    END PROCESS;

    PROCESS (jk, c, qs0, qs1)
    BEGIN
        IF (qs0'event and qs0 = '0') THEN
            qs1 <= not qs1;
        END IF;
    END PROCESS;
```

```
    PROCESS (jk, c, qs1, qs2)
    BEGIN
        IF (qs1'event and qs1 = '0') THEN
            qs2 <= not qs2;
        END IF;
    END PROCESS;

    PROCESS (jk, c, qs2, qs3)
    BEGIN
        IF (qs2'event and qs2 = '0') THEN
            qs3 <= not qs3;
        END IF;
    END PROCESS;

    q0 <= qs0 and r and jk;
    q1 <= qs1 and r and jk;
    q2 <= qs2 and r and jk;
    q3 <= qs3 and r and jk;

END behav;
```

Квартус пошел нафик хихихих сами разберетесь мне впадлу

Спасибо за внимание всем пока