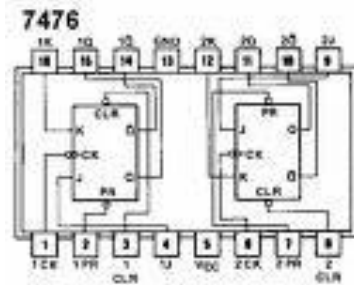


# FLIP-FLOP JK





## Objetivos:

- Adquirir conhecimentos em dispositivos de lógica programável;
- Estudo do flip-flop JK;
- Estudo do circuito contador.

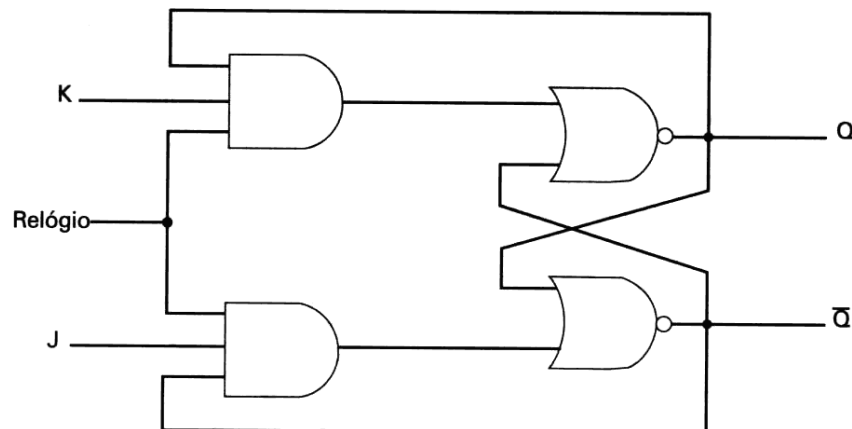


Para a implementação de circuitos contadores assíncronos é necessário a compreensão dos flip-flops JK.

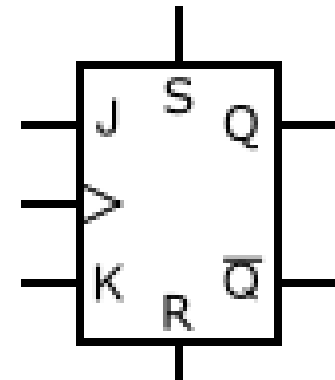
Tabela verdade flip-flop JK

J	K	$Q_{\text{Futuro}}$	Comentário
0	0	$Q_{\text{Anterior}}$	Memória
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q_{\text{Anterior}}}$	Inverte

Circuito flip-flop JK

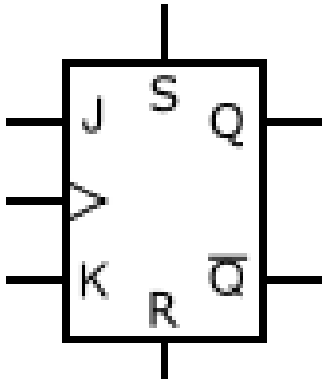


Representação simplificada

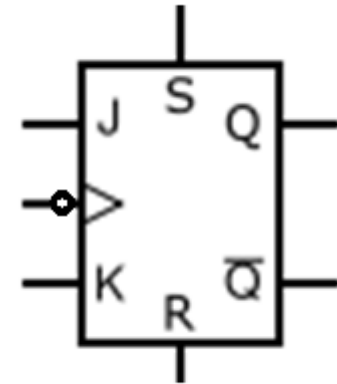


Características do sinal de ativação do clock:

Flip-flop de borda de subida



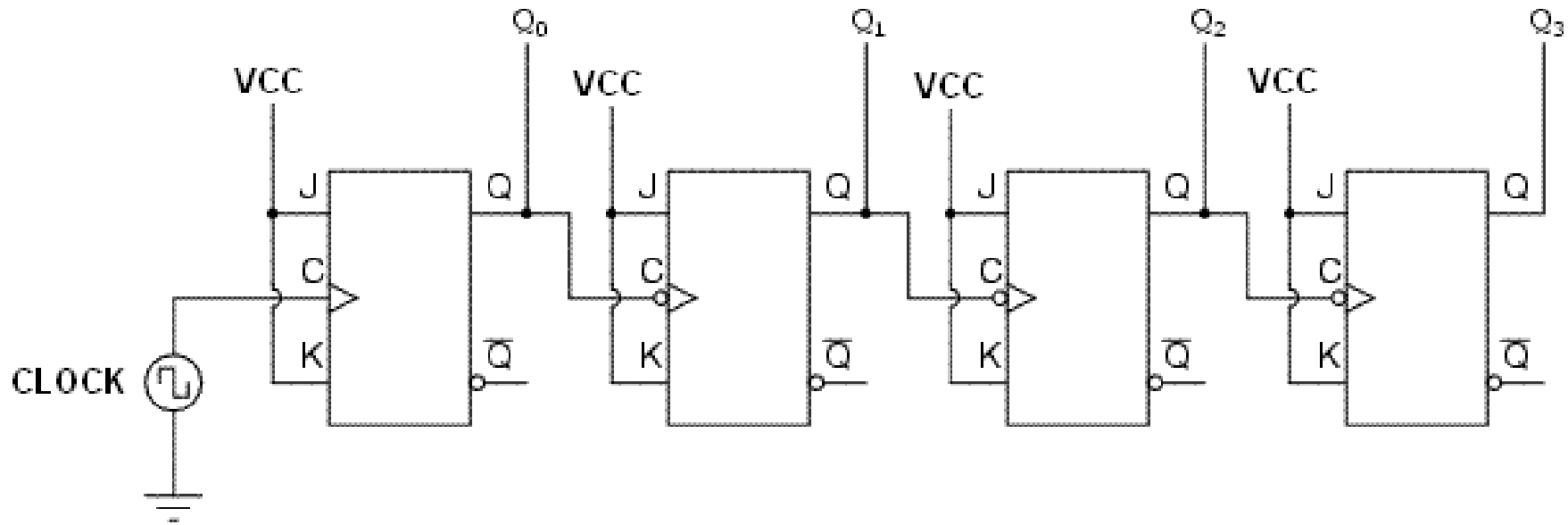
Flip-flop de borda de descida



O Tipo de borda indica qual é o momento de ativação (atualização) do componente:

- Subida - de zero para um;
- Descida - de um para zero.

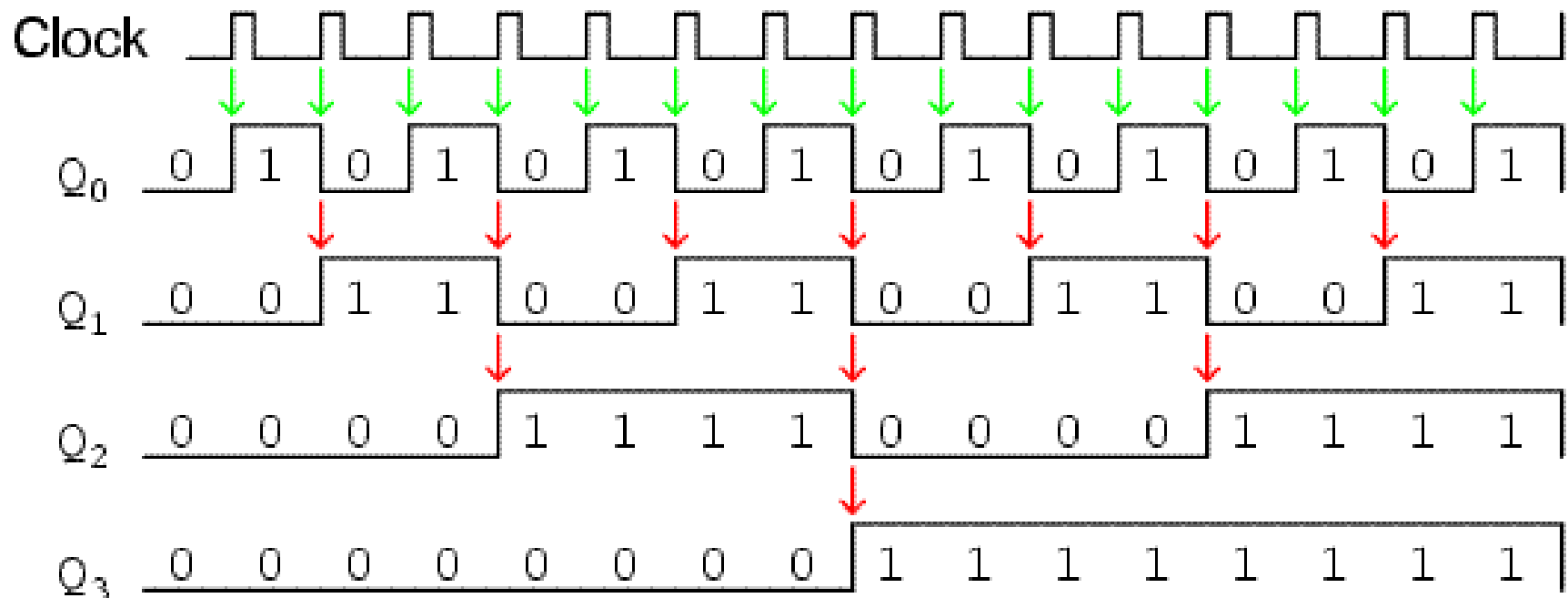
Contador assíncrono de 4 bits:



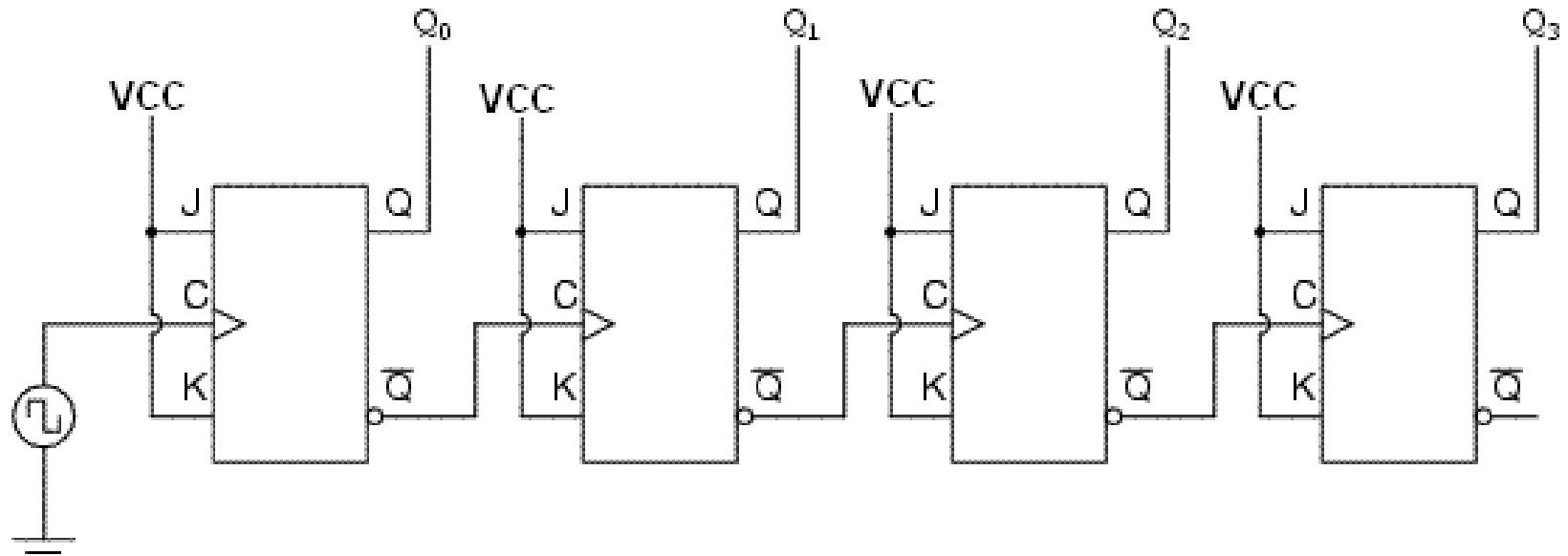
Inicialmente, supõem-se as saídas zeradas. Aplica-se um pulso de clock no primeiro flip-flop, cuja mudança de estado na saída ocorrerá na subida do clock. O flip-flop seguinte mudará o nível lógico na saída sempre que ocorrer a mudança (descida do clock) de nível lógico no flip-flop anterior. O diagrama de tempo abaixo ilustra melhor a sequência de funcionamento do contador.

Após os 16 pulsos de clock, o contador irá reiniciar a contagem.

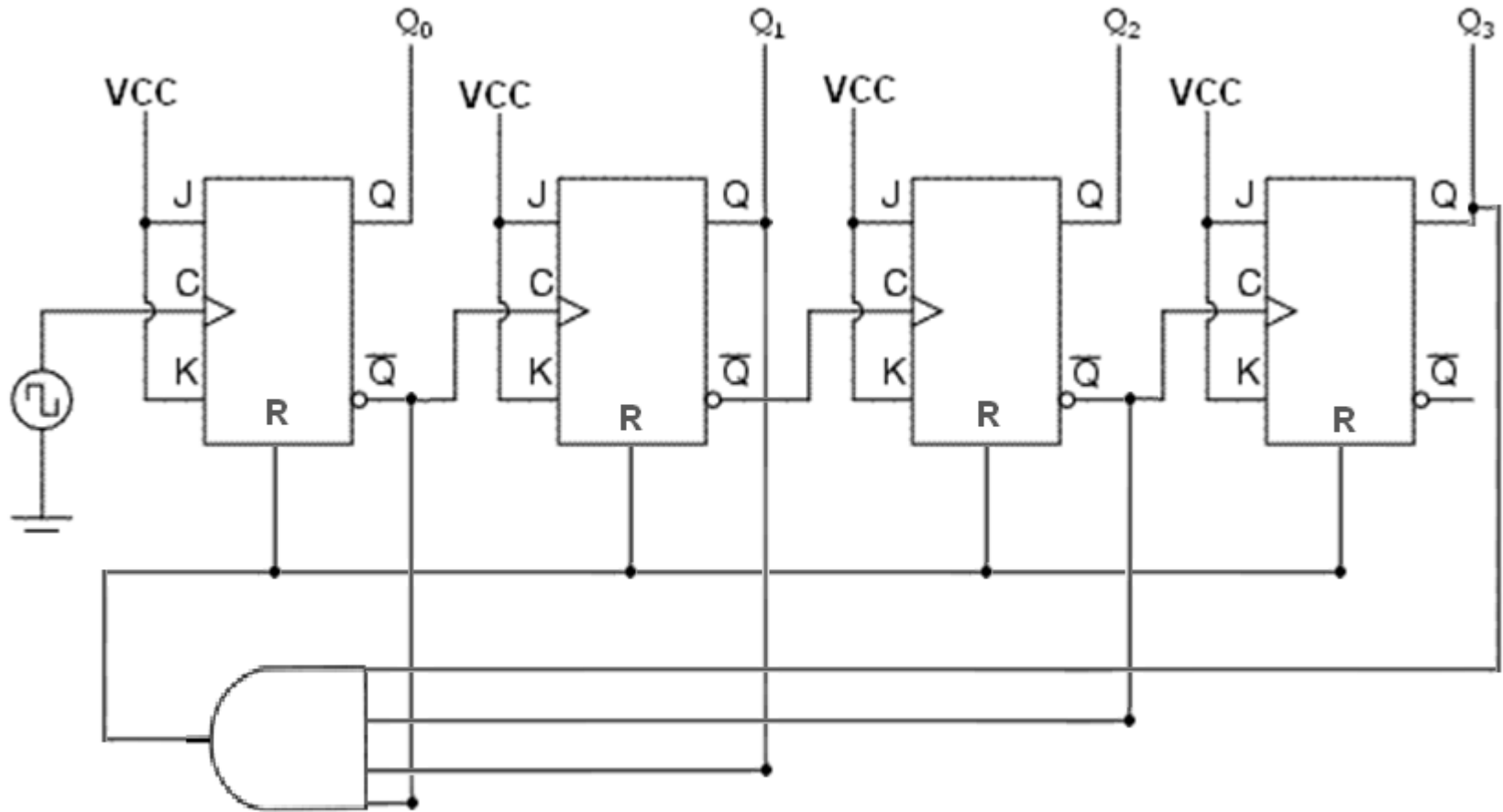
Forma de onda do contador



Outra forma de implementação de contador assíncrono



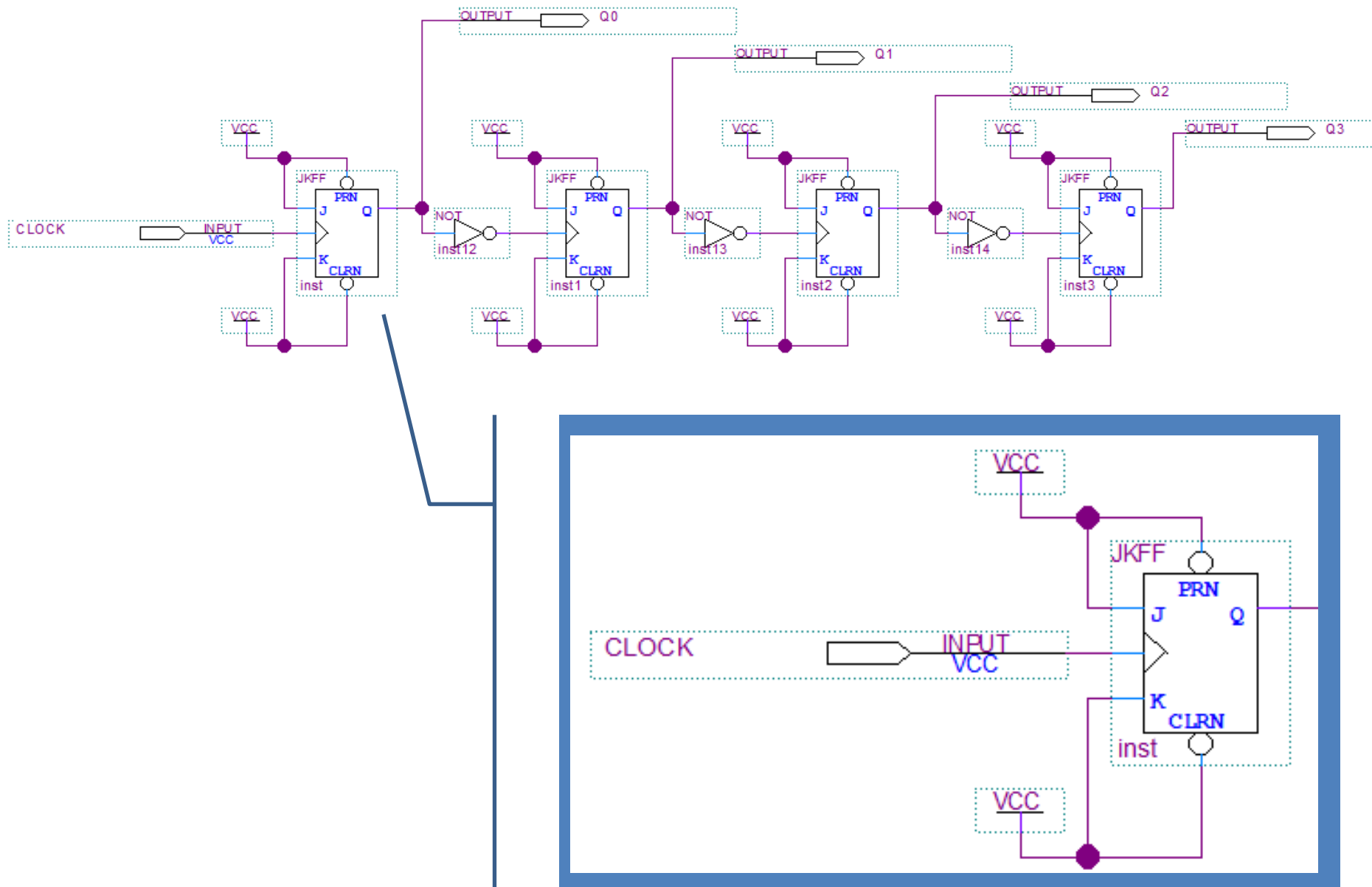
Versão contador de 0 até 9:



Pois quando o contador chegar em  $10_{(10)}$  ou  $1010_{(2)}$  irá ser *resetado*, pois os valores  $1010_{(2)}$  estão para  $Q_3Q_2Q_1Q_0$  que irão ativar os resets dos flip-flops por meio da porta AND.



Circuito a ser implementado no quartus:



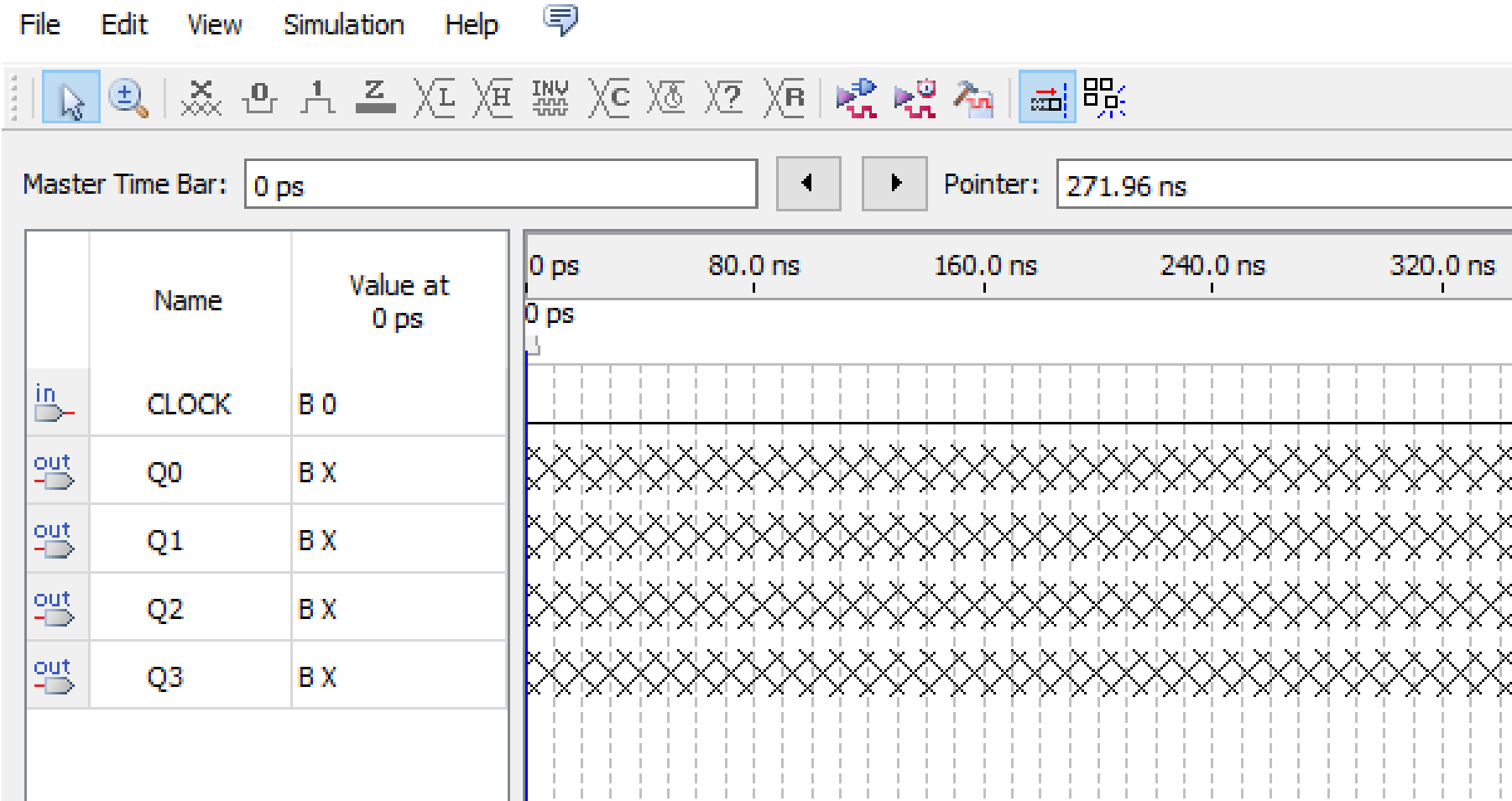
## Tabela dos pinos de entrada (chaves on/off)

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
SW[0]	PIN_V28	Slide Switch[0]	2.5V
SW[1]	PIN_U30	Slide Switch[1]	2.5V
SW[2]	PIN_V21	Slide Switch[2]	2.5V
SW[3]	PIN_C2	Slide Switch[3]	2.5V
SW[4]	PIN_AB30	Slide Switch[4]	2.5V
SW[5]	PIN_U21	Slide Switch[5]	2.5V
SW[6]	PIN_T28	Slide Switch[6]	2.5V
SW[7]	PIN_R30	Slide Switch[7]	2.5V
SW[8]	PIN_P30	Slide Switch[8]	2.5V
SW[9]	PIN_R29	Slide Switch[9]	2.5V
SW[10]	PIN_R26	Slide Switch[10]	2.5V
SW[11]	PIN_N26	Slide Switch[11]	2.5V
SW[12]	PIN_M26	Slide Switch[12]	2.5V
SW[13]	PIN_N25	Slide Switch[13]	2.5V
SW[14]	PIN_J26	Slide Switch[14]	2.5V
SW[15]	PIN_K25	Slide Switch[15]	2.5V
SW[16]	PIN_C30	Slide Switch[16]	2.5V
SW[17]	PIN_H25	Slide Switch[17]	2.5V

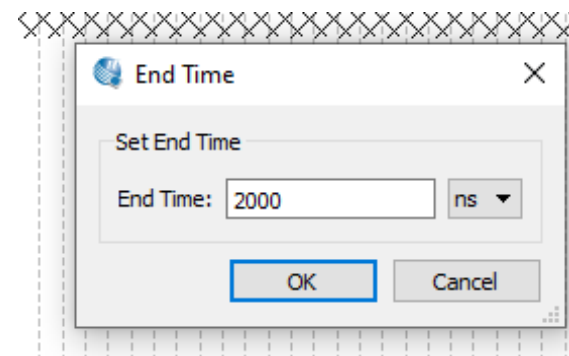
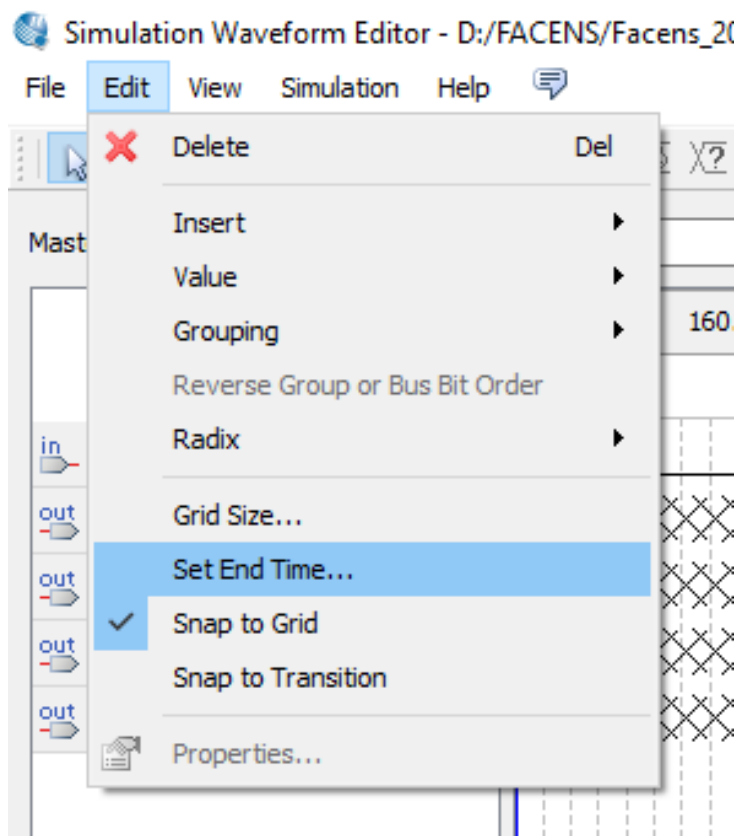
## Tabela dos pinos de saída (Leds)

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
LEDR[0]	PIN_T23	LED Red[0]	2.5V
LEDR[1]	PIN_T24	LED Red[1]	2.5V
LEDR[2]	PIN_V27	LED Red[2]	2.5V
LEDR[3]	PIN_W25	LED Red[3]	2.5V
LEDR[4]	PIN_T21	LED Red[4]	2.5V
LEDR[5]	PIN_T26	LED Red[5]	2.5V
LEDR[6]	PIN_R25	LED Red[6]	2.5V
LEDR[7]	PIN_T27	LED Red[7]	2.5V
LEDR[8]	PIN_P25	LED Red[8]	2.5V
LEDR[9]	PIN_R24	LED Red[9]	2.5V

Insira os sinais para a simulação



Defini a faixa da simulação para até 2000ns



Crie uma forma de onda para o sinal do clock com os parâmetros abaixo:

The screenshot shows a logic simulator interface. On the left, a table lists signals and their values at 0 ps:

	Name	Value at 0 ps
in	CLOCK	B 0
out	Q0	B X
out	Q1	B X
out	Q2	B X
out	Q3	B X

The main workspace shows a timing diagram with a Master Time Bar at 0 ps and a Pointer at 224.28 ns. A 'Clock' dialog box is open, showing the following parameters:

- Base waveform on time period
- Period: 50 ns
- Offset: 100 ns
- Duty cycle (%): 50

The dialog box has 'OK' and 'Cancel' buttons.

- Após a geração do gráfico conectar o KIT da INTEL ao computador e carregar o programa (circuito elaborado) para dentro do FPGA.
- Fazer as devidas configurações dos pinos das entradas e saídas do kit e fazer o teste prático.
- Demonstrar ao professor o funcionamento.

### Relatório 5

- Introdução
- Construção do circuito contador no programa Quartus
  - Definição do circuito no software
- Procedimento experimental executado
- Demonstração com forma de onda na execução do circuito
  - Para modelo de simulação *funcional*
  - Para modelo de simulação *timing*
  - Análise as formas de onda nos dois casos acima descritos
- Conclusão