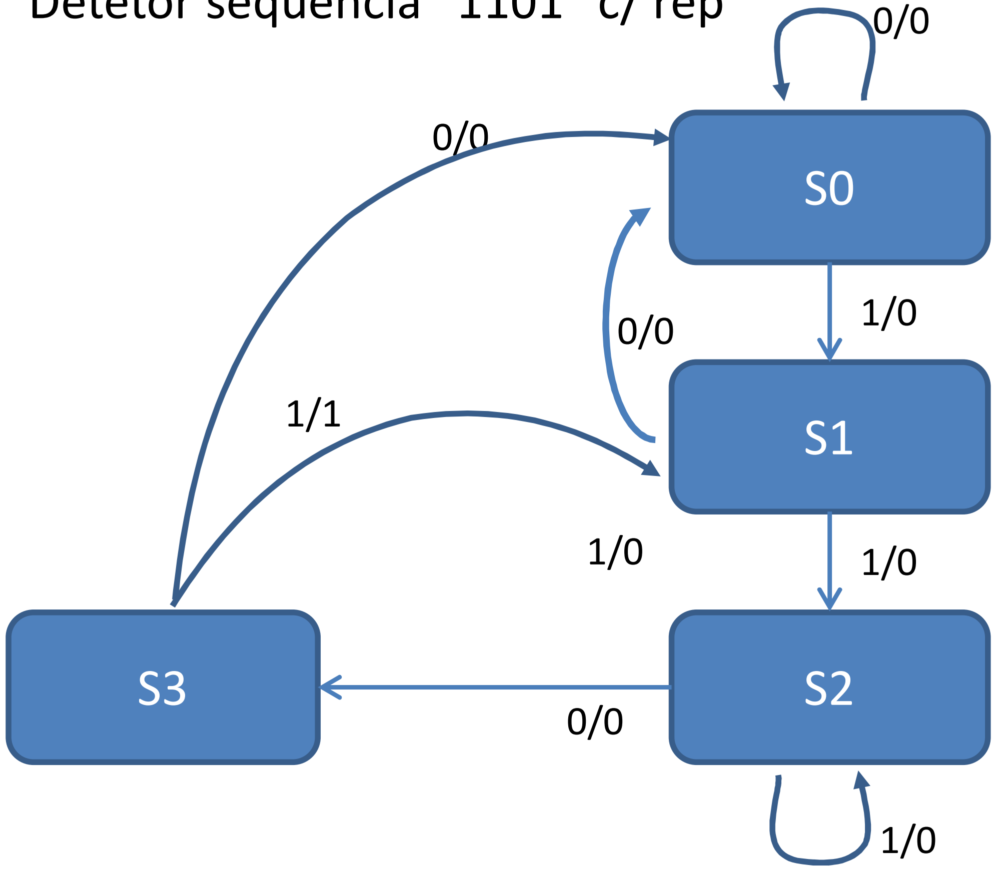
Guião 11_2021
Problema 1

Detetor sequencia “1101” c/ rep



S	Q1	Q0	x	S*	D1	D0	y
S0	0	0	0	S0	0	0	0
S0	0	0	1	S1	0	1	0
S1	0	1	0	S0	0	0	0
S1	0	1	1	S2	1	0	0
S2	1	0	0	S3	1	1	0
S2	1	0	1	S2	1	0	0
S3	1	1	0	S0	0	0	0
S3	1	1	1	S1	0	1	1

$$D1 = Q_1\overline{Q_0} + X\,(Q_1 \oplus Q_0)$$

D1=Q1*	Q1Q0			
x	00	01	11	10
0				1
1		1		1

$$D0 = Q_1(\overline{X \oplus Q_0}) + X(\overline{Q_1 \oplus Q_0})$$

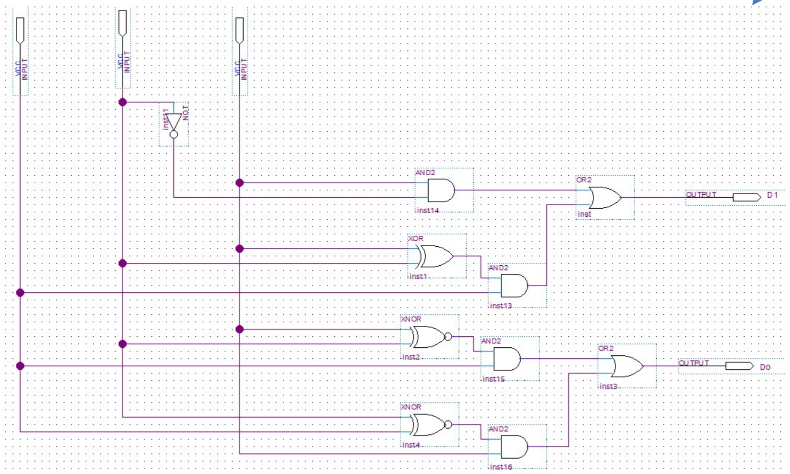
D0=Q0*	Q1Q0			
X	00	01	11	10
0				1
1	1		1	

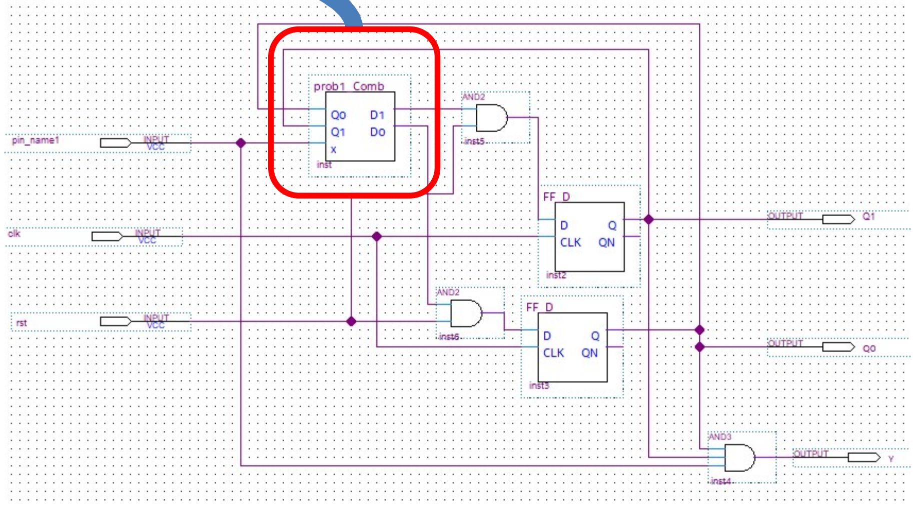
$$Y = Q_1Q_0X$$

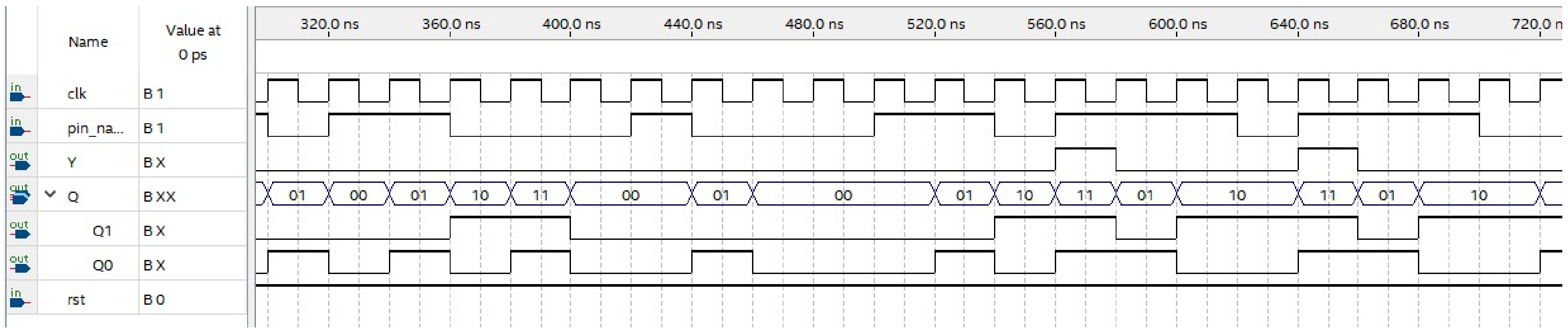
Design a sequence detector, according to Mealy’s model, whose output, y, is '1' whenever input sequence “1101” occurs. Overlapped sequences are allowed. An example is given below, where the x input values are received one at each clock cycle (read them from left to right):

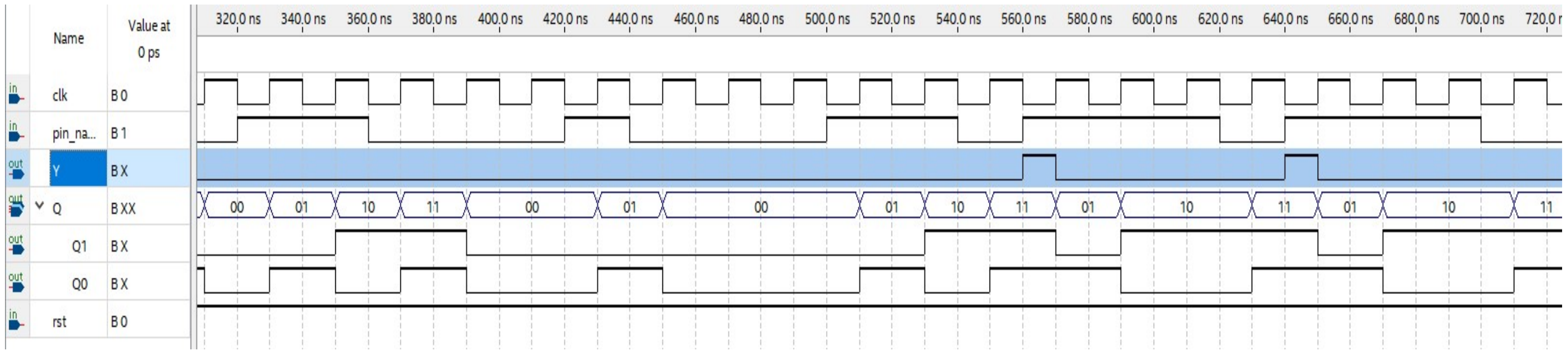
x 0 1 0 1 1 1 0 1 1 0 1 0 1 1
x 0 1 0 1 1 1 0 1 1 0 1 0 1 1
y 0 0 0 0 0 0 0 1 0 0 1 0 0 0

Guião 11_2021
Problema 1









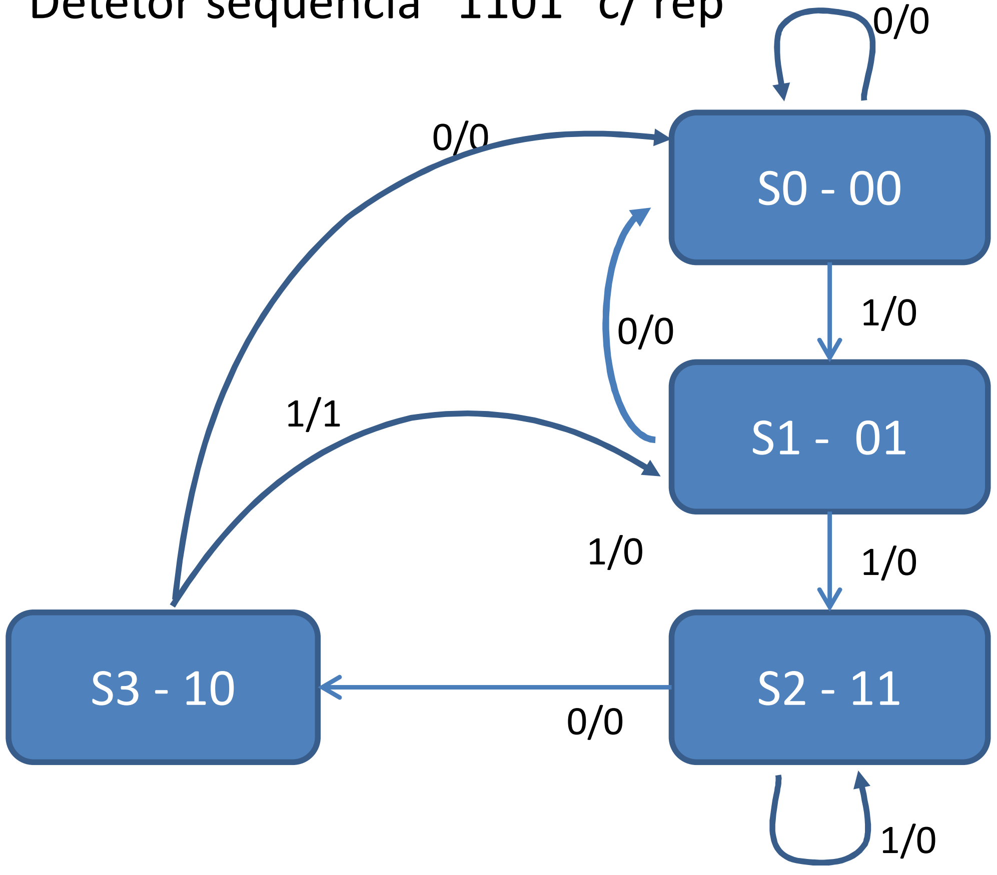
[Paper and pencil + Quartus Prime]. Design a sequence detector, according to Mealy's model, whose output, `y`, is '1' whenever input sequence "1101" occurs. Overlapped sequences are allowed. An example is given below, where the `x` input values are received one at each clock cycle (read them from left to right):

x	0	1	0	1	1	1	0	1	1	0	1	0	1	1
y	0	0	0	0	0	0	0	1	0	0	1	0	0	0

Create a new project named "SeqDet1101" in *Quartus Prime* software. Create a new file for a logic diagram called "SeqDet1101.bdf" to implement detector based on logic gates and D flip-flops (use `dff` component in Quartus library). Perform functional simulation and check whether the detector works correctly for the input sequence given above.

Guião 11_2021
Problema 1

Detetor sequencia “1101” c/ rep



Codificação Gray dos Estados

S	Q1	Q0	x	S*	D1	D0	y
S0	0	0	0	S0	0	0	0
S0	0	0	1	S1	0	1	0
S1	0	1	0	S0	0	0	0
S1	0	1	1	S2	1	1	0
S2	1	1	0	S3	1	0	0
S2	1	1	1	S2	1	1	0
S3	1	0	0	S0	0	0	0
S3	1	0	1	S1	0	1	1

$$D1 = Q_1 Q_0 + X\,Q_0$$

D1=Q1*	Q1Q0			
X	00	01	11	10
0			1	
1		1	1	

$$D0 = X$$

D0=Q0*	Q1Q0			
X	00	01	11	10
0				
1	1	1	1	1

$$Y = Q_1 Q_0' X$$

Design a sequence detector, according to Mealy’s model, whose output, y, is '1' whenever input sequence “1101” occurs. Overlapped sequences are allowed. An example is given below, where the x input values are received one at each clock cycle (read them from left to right):

```
x 0 1 0 1 1 1 0 1 1 0 1 0 1 1
y 0 0 0 0 0 0 0 1 0 0 1 0 0 0
```

. *[Paper and pencil + Quartus Prime]*. Design a sequential circuit that detects 5-bit long input sequences which start with ”11” and contain exactly 3 “1”s. The circuit should work in such a way that once two initial “1”s are detected, the sequence is parsed to the end (with or without success), i.e. the following sequence can only start after three more bits are received. An example is given below, where the `x` input values are received one at each clock cycle (read them from left to right):

x	1	0	0	1	1	0	1	1	0	0	1	1	0	1	0
y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Create a new project named “SeqDet3in5” in *Quartus Prime* software. Create a new file for a logic diagram called “SeqDet3in5.bdf” to implement detector based on logic gates and D flip-flops (use `dff` component in Quartus library). Perform functional simulation and check whether the detector works correctly for the input sequence given above.

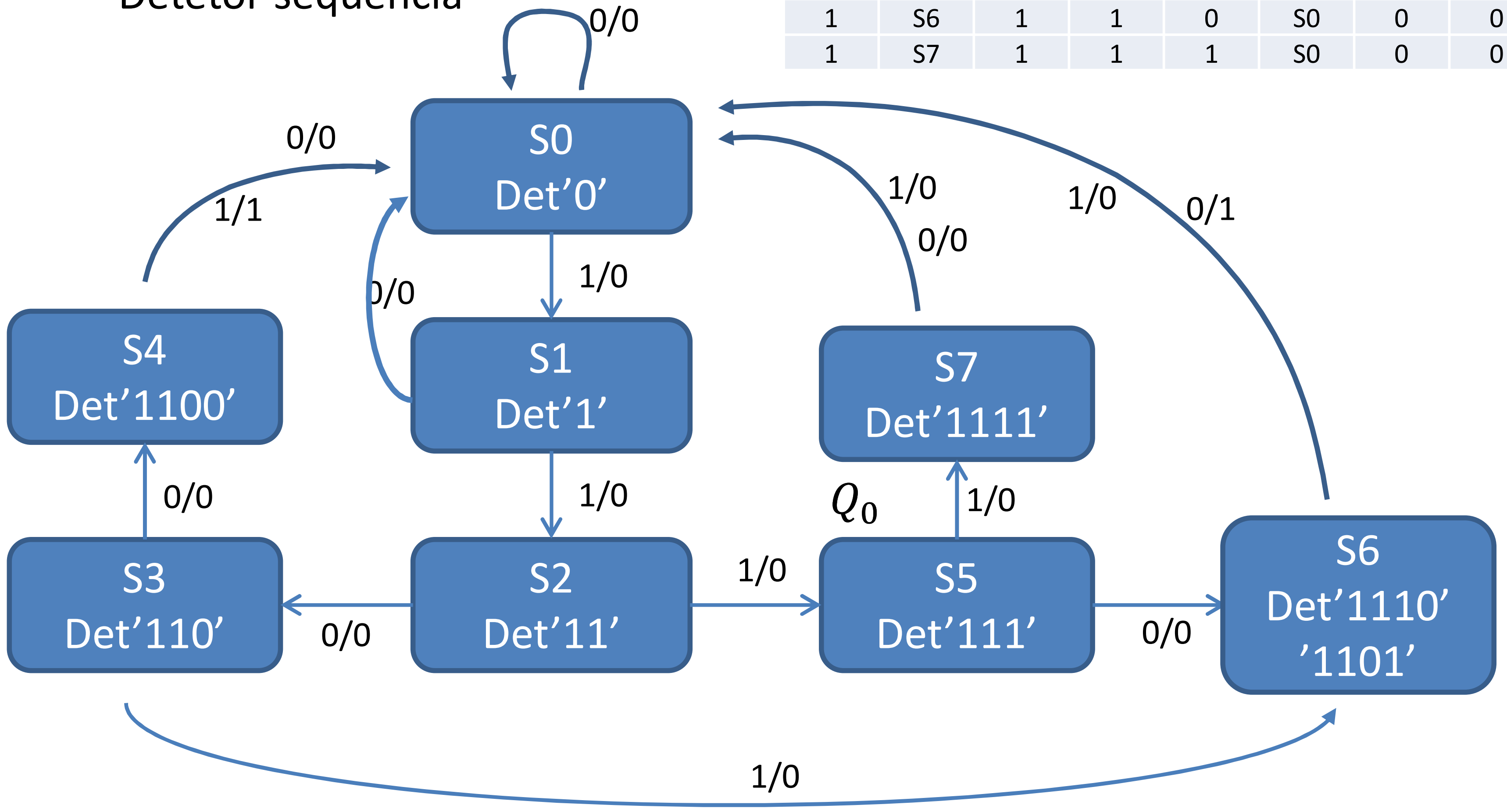
Guião 11_2021
Problema 2

[Paper and pencil + Quartus Prime]. Design a sequential circuit that detects 5-bit long input sequences which start with ”11” and contain exactly 3 “1”s. The circuit should work in such a way that once two initial “1”s are detected, the sequence is parsed to the end (with or without success), i.e. the following sequence can only start after three more bits are received. An example is given below, where the x input values are received one at each clock cycle (read them from left to right):

```
x 1 0 0 1 1 0 1 1 0 0 1 1 0 1 0
y 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

x		Q2	Q1	Q0		D2	D1	D0	y
0	S0	0	0	0	S0	0	0	0	0
0	S1	0	0	1	S0	0	0	0	0
0	S2	0	1	0	S3	0	1	1	0
0	S3	0	1	1	S4	1	0	0	0
0	S4	1	0	0	S0	0	0	0	0
0	S5	1	0	1	S6	1	1	0	0
0	S6	1	1	0	S0	0	0	0	1
0	S7	1	1	1	S0	0	0	0	0
1	S0	0	0	0	S1	0	0	1	0
1	S1	0	0	1	S2	0	1	0	0
1	S2	0	1	0	S5	1	0	1	0
1	S3	0	1	1	S6	1	1	0	0
1	S4	1	0	0	S0	0	0	0	1
1	S5	1	0	1	S7	1	1	1	0
1	S6	1	1	0	S0	0	0	0	0
1	S7	1	1	1	S0	0	0	0	0

Detetor sequência



Guião 11_2021
Problema 2

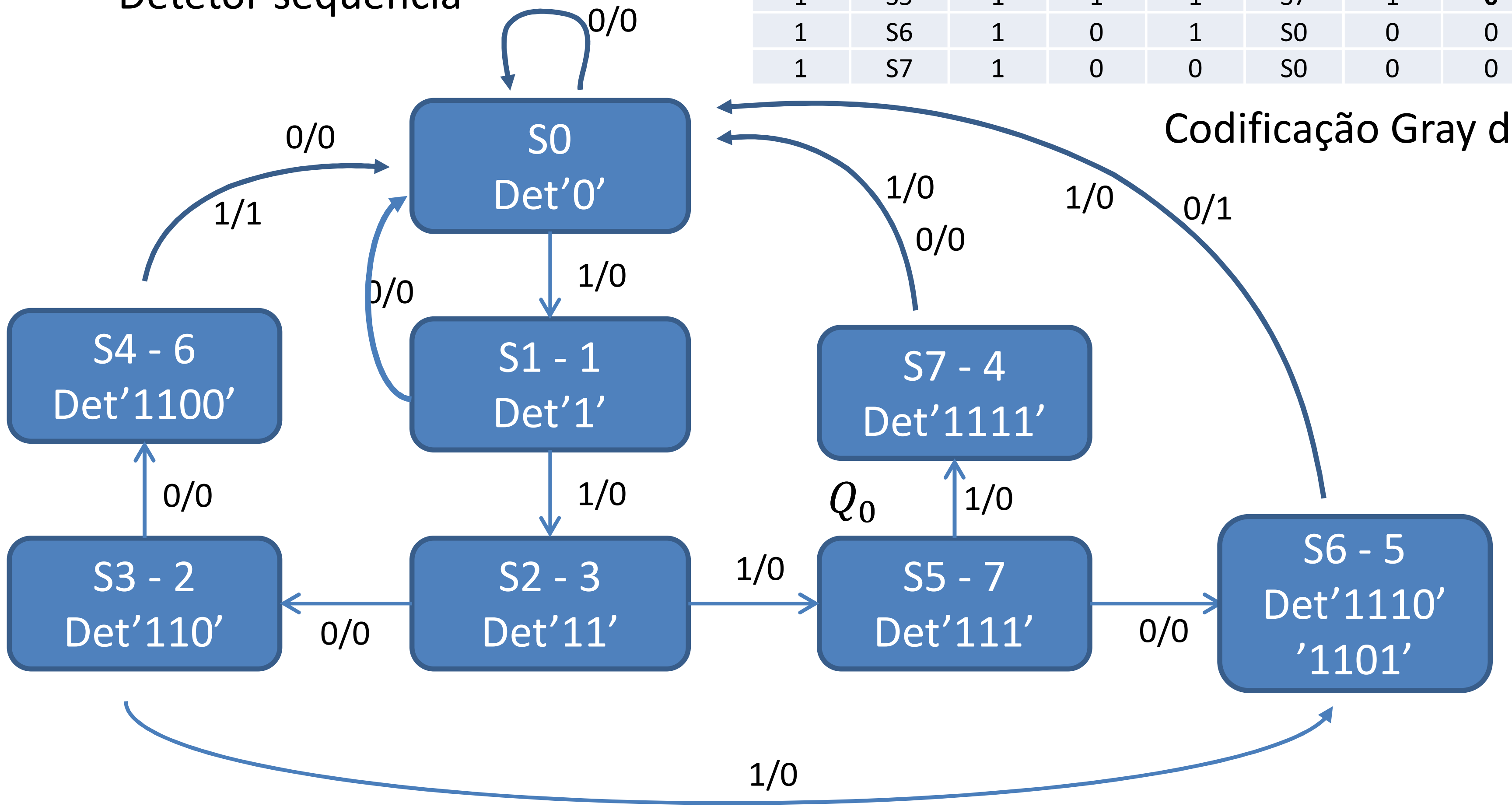
[Paper and pencil + Quartus Prime]. Design a sequential circuit that detects 5-bit long input sequences which start with "11" and contain exactly 3 "1"s. The circuit should work in such a way that once two initial "1"s are detected, the sequence is parsed to the end (with or without success), i.e. the following sequence can only start after three more bits are received. An example is given below, where the x input values are received one at each clock cycle (read them from left to right):

```
x 1 0 0 1 1 0 1 1 0 0 1 1 0 1 0
y 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

x		Q2	Q1	Q0		D2	D1	D0	y
0	S0	0	0	0	S0	0	0	0	0
0	S1	0	0	1	S0	0	0	0	0
0	S2	0	1	1	S3	0	1	0	0
0	S3	0	1	0	S4	1	1	0	0
0	S4	1	1	0	S0	0	0	0	0
0	S5	1	1	1	S6	1	0	1	0
0	S6	1	0	1	S0	0	0	0	1
0	S7	1	0	0	S0	0	0	0	0
1	S0	0	0	0	S1	0	0	1	0
1	S1	0	0	1	S2	0	1	1	0
1	S2	0	1	1	S5	1	1	1	0
1	S3	0	1	0	S6	1	0	1	0
1	S4	1	1	0	S0	0	0	0	1
1	S5	1	1	1	S7	1	**0**	0	0
1	S6	1	0	1	S0	0	0	0	0
1	S7	1	0	0	S0	0	0	0	0

Codificação Gray dos Estados

Detetor sequencia



Guião 11_2021
Problema 2

D2 = Q2+

xQ2

Q1Q0	00	01	11	10
00				
01				
11		1		1
10	1			1

x Q2' Q1 + Q1 Q0' Q2' + Q1 Q0 (X xor Q2)

D0 = Q0+

xQ2

Q1Q0	00	01	11	10
00				1
01				1
11		1		1
10				1

xQ2' + Q1 Q0 (X xor Q2)

D1 = Q1+

xQ2

Q1Q0	00	01	11	10
00				
01				1
11	1		1	1
10	1			

x' Q2' Q1 + Q1 Q0 X + X Q2' Q0

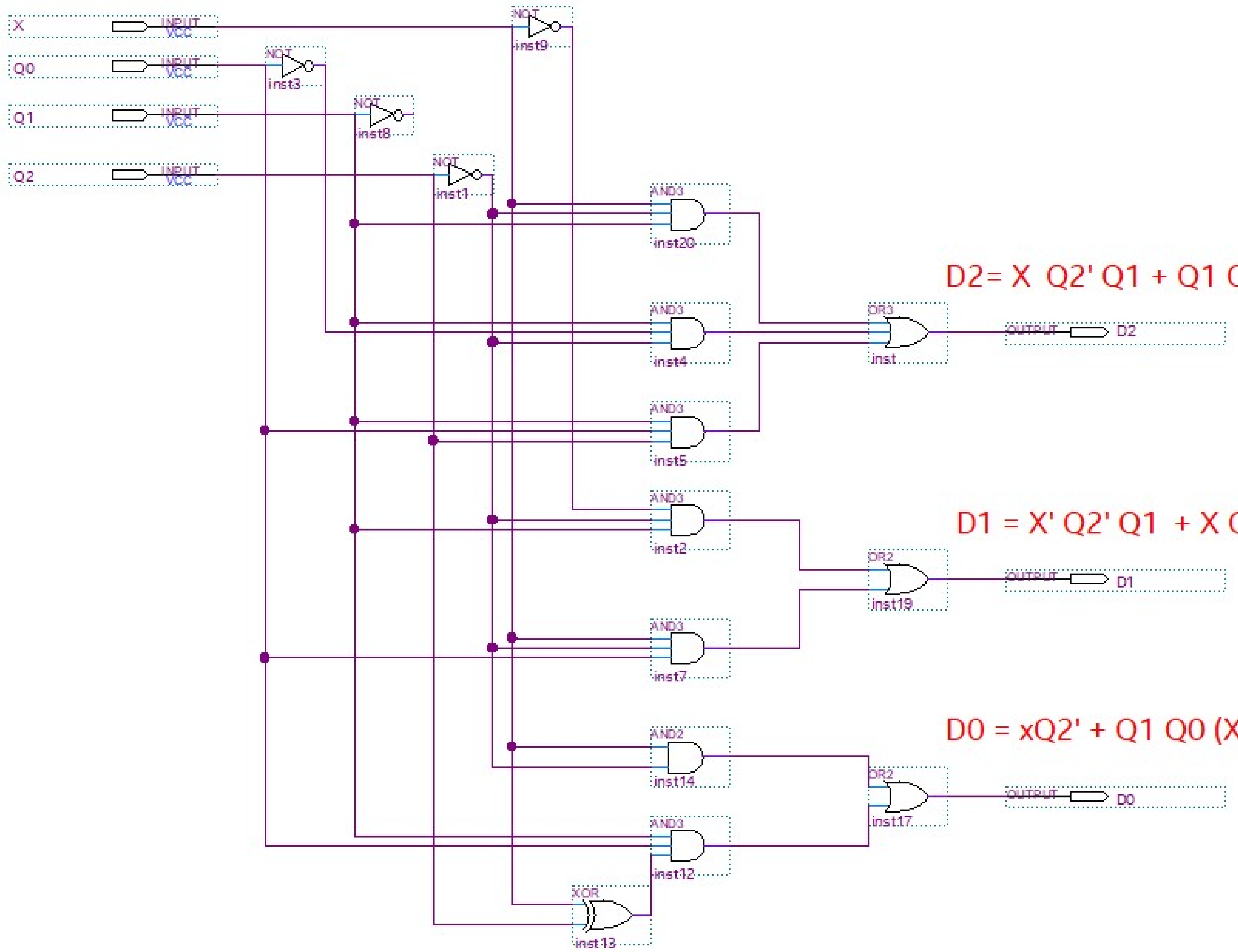
y

xQ2

Q1Q0	00	01	11	10
00				1
01		1		1
11				1
10			1	1

x Q2' + Q1 Q0' X + X' Q2 Q1' Q0

Guião 11_2021
Problema 2



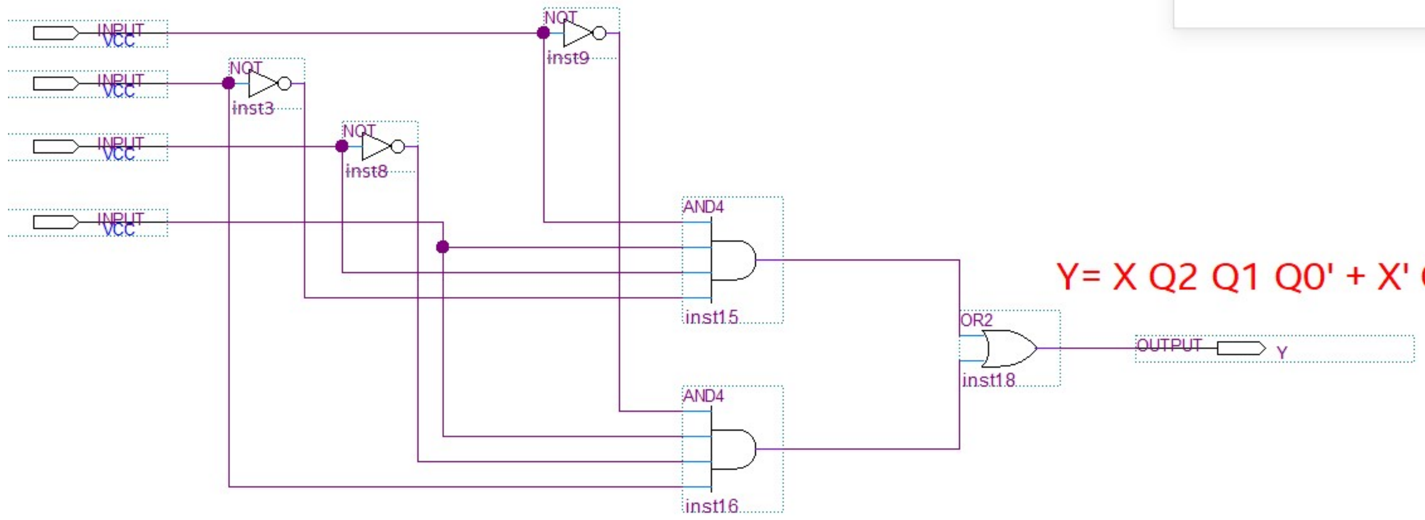
D2= X Q2' Q1 + Q1 Q0' Q2' + Q2 Q1 Q0

D1 = X' Q2' Q1 + X Q2' Q0

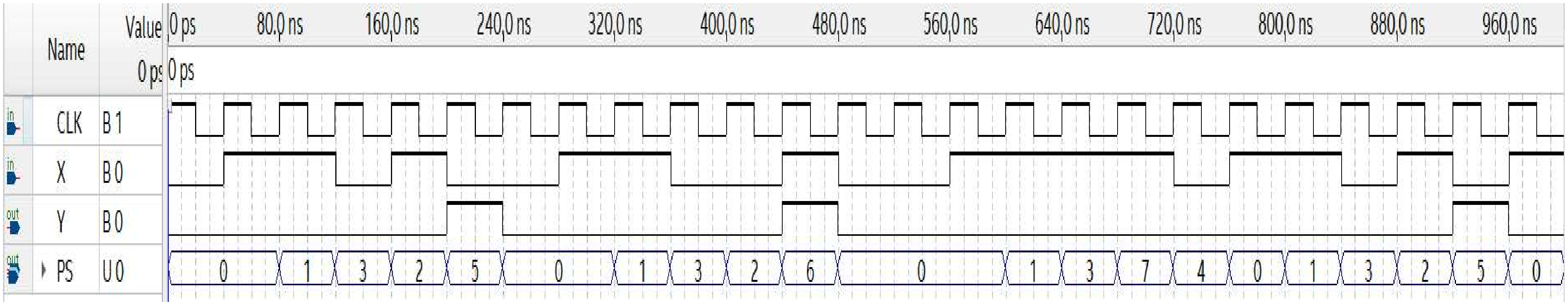
D0 = xQ2' + Q1 Q0 (X xor Q2)

Guião 11_2021

Problema 2



Y= X Q2 Q1 Q0' + X' Q2 Q1' Q0

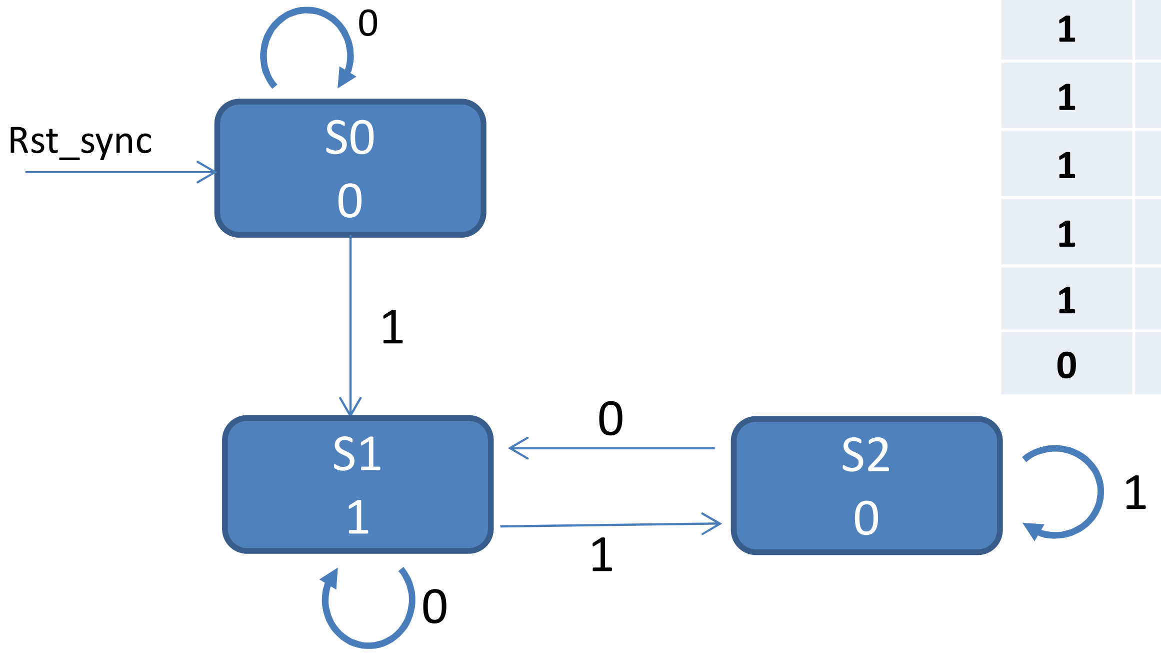


Guião 11_2021
Problema 3

Design and implement a synchronous sequential circuit, according to Moore's model, that performs the arithmetic negation of a two's complement number of arbitrary length, that enters the circuit starting with its least significant bit. Admit that the circuit has a synchronous active-low reset input. An example is given below, where the x input values are received one at each clock cycle (read them from left to right):

x 0 1 0 1
y 0 1 1 0

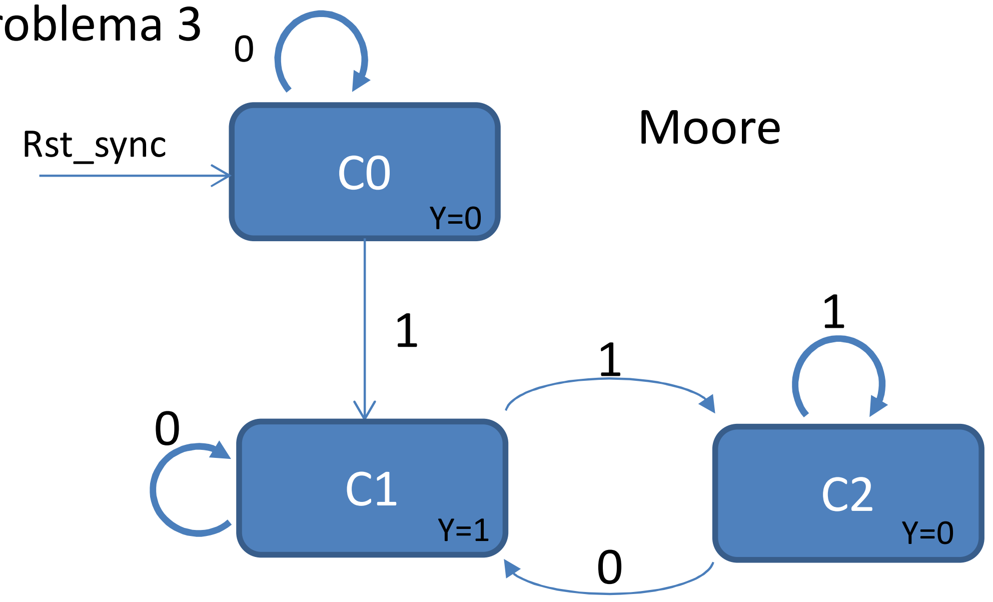
The inserted number, represented in two's complement is 1010 (-6) and its calculated arithmetic negation is 0110 (6)

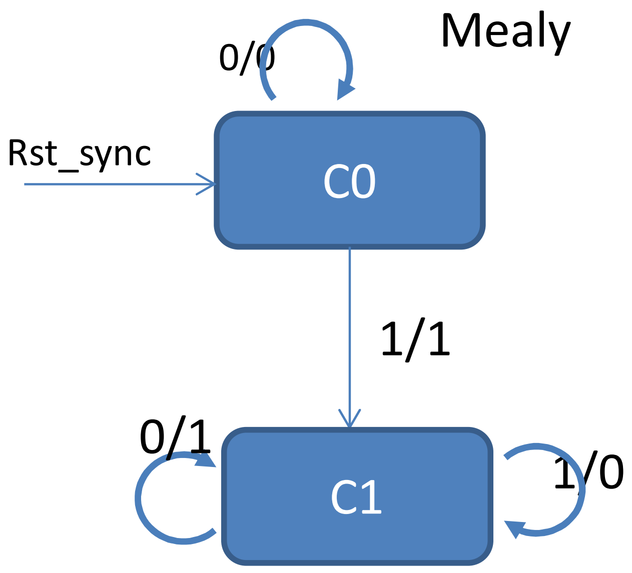


rs_sync	S	Q1	Q0	In	S*	D1	D0	y
1	S0	0	0	0	S0	0	0	0
1	S0	0	0	1	S1	0	1	0
1	S1	0	1	0	S1	0	1	0
1	S1	0	1	1	S2	1	0	1
1	S2	1	0	0	S1	0	1	0
1	S2	1	0	1	S2	1	0	1
1	S3	1	1	0	S0	x	x	x
1	S3	1	1	1	S0	x	x	x
0	x	x	x	x	S0	0	0	0

Guião 11_2021

Problema 3





		X=0		X=1		
Q1	**Q0**	**D1**	**D0**	**D1**	**D0**	**out**
0	0	0	0	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	0
1	1	*	*	*	*	*

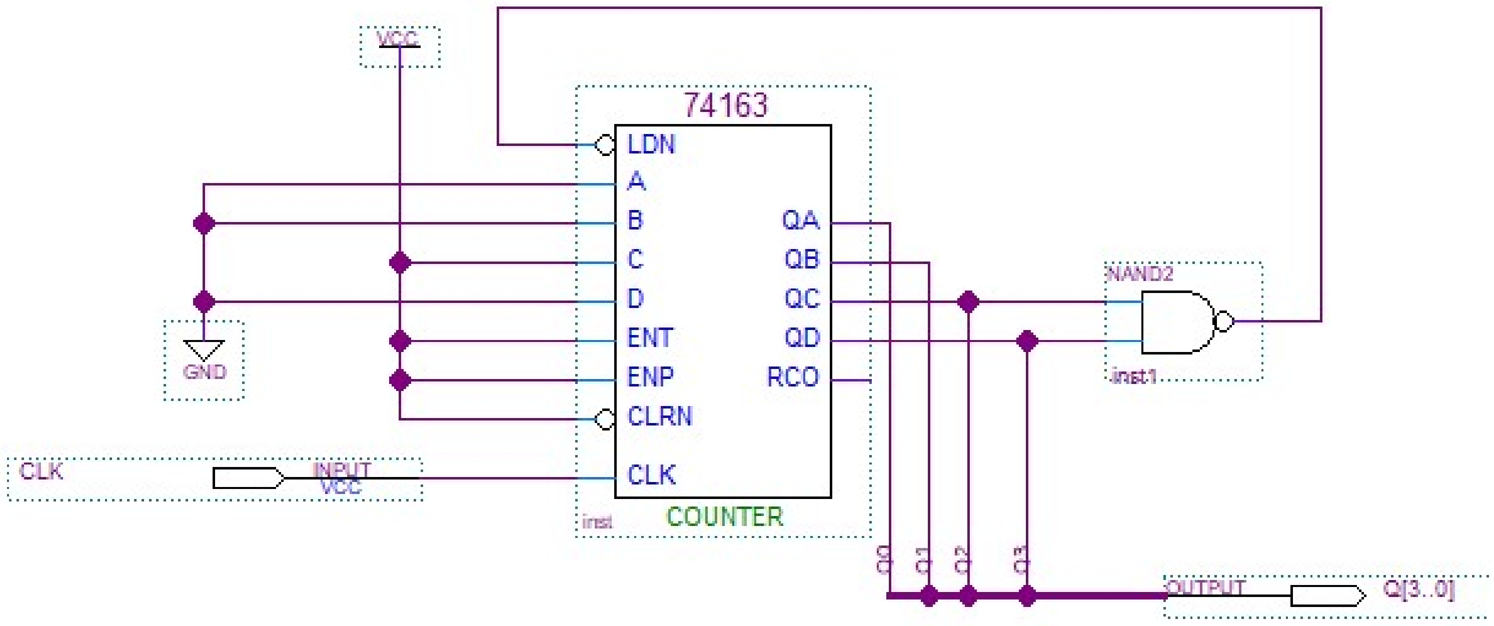
Guião 11_2021

Problema 4 Gray counter

Nest state truth table

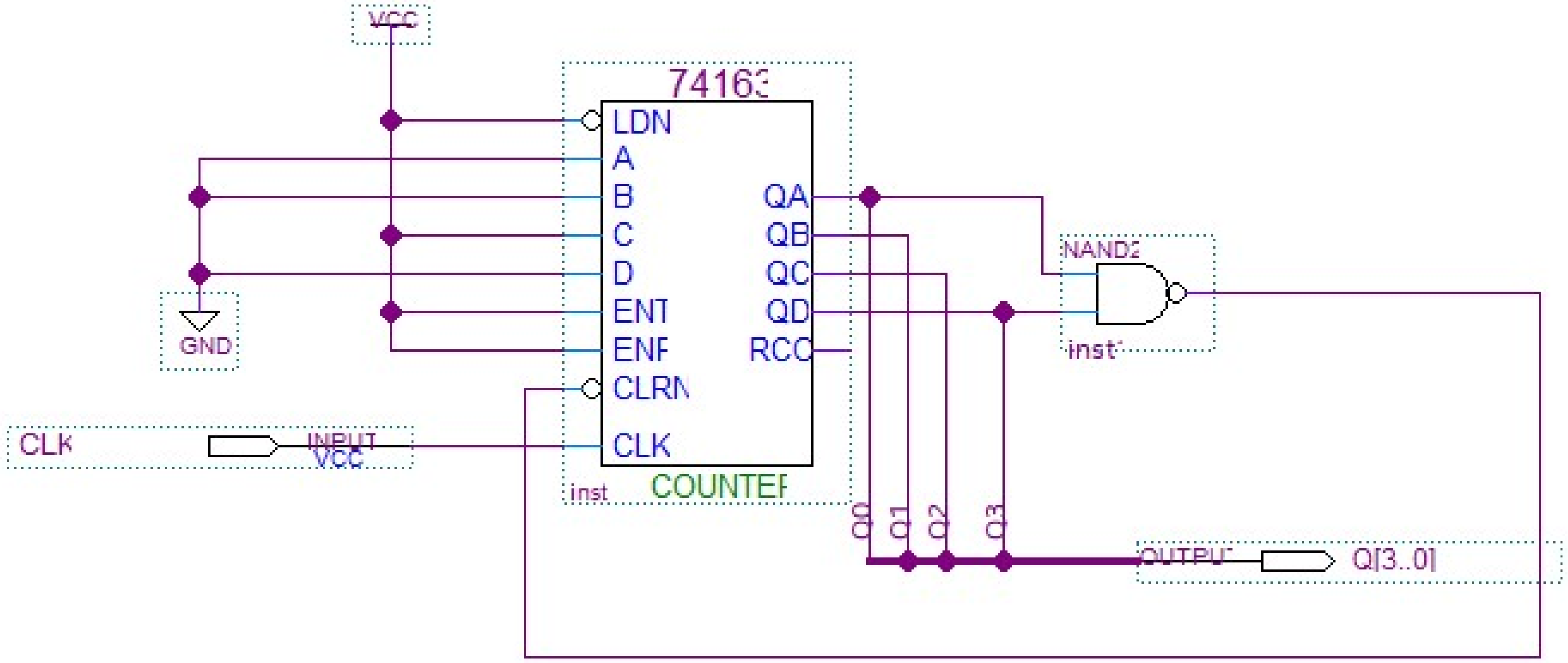
Q2	Q1	Q0	Q2+	Q1+	Q0+
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Guião 11_2021
Problema 5

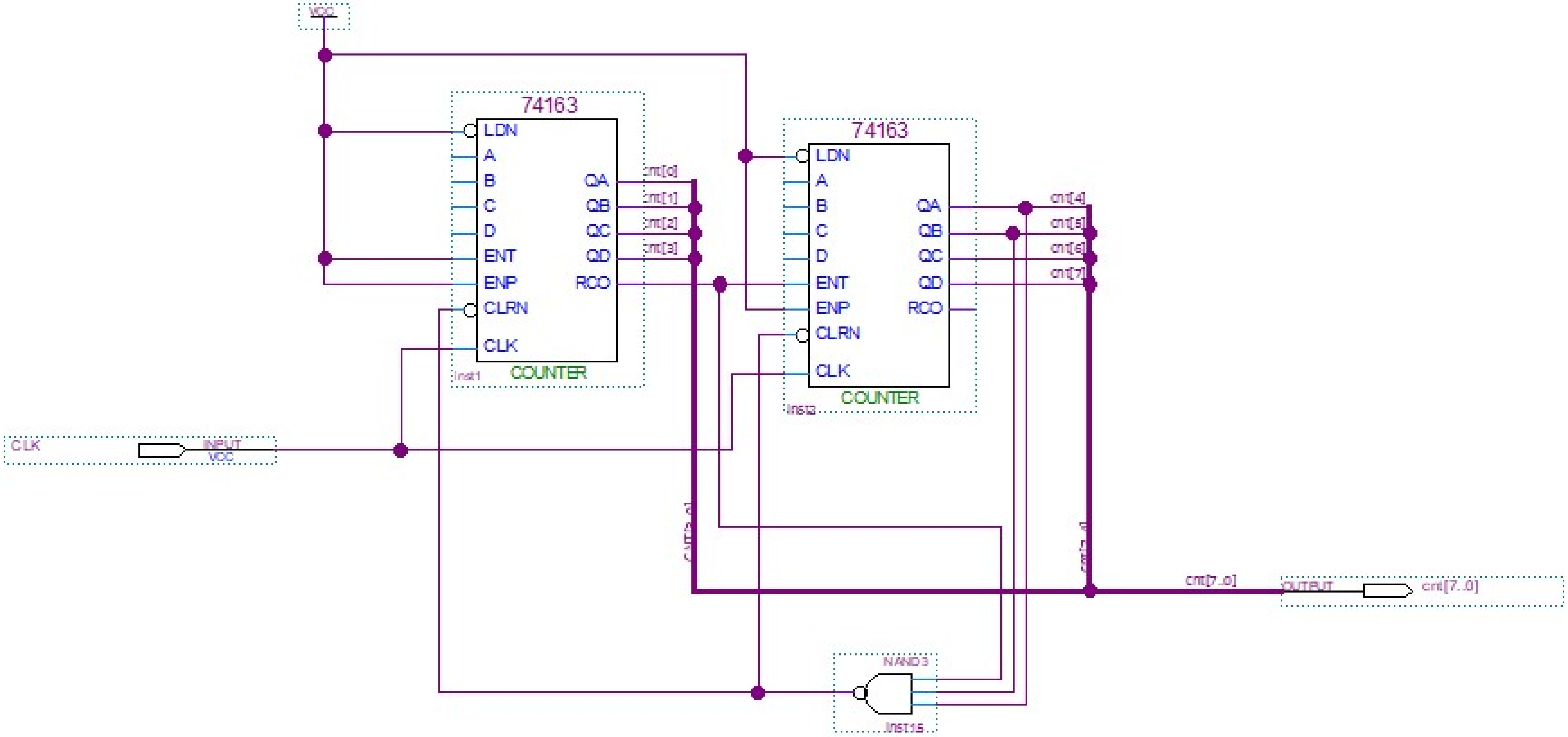


Guião 11_2021

Problema 6

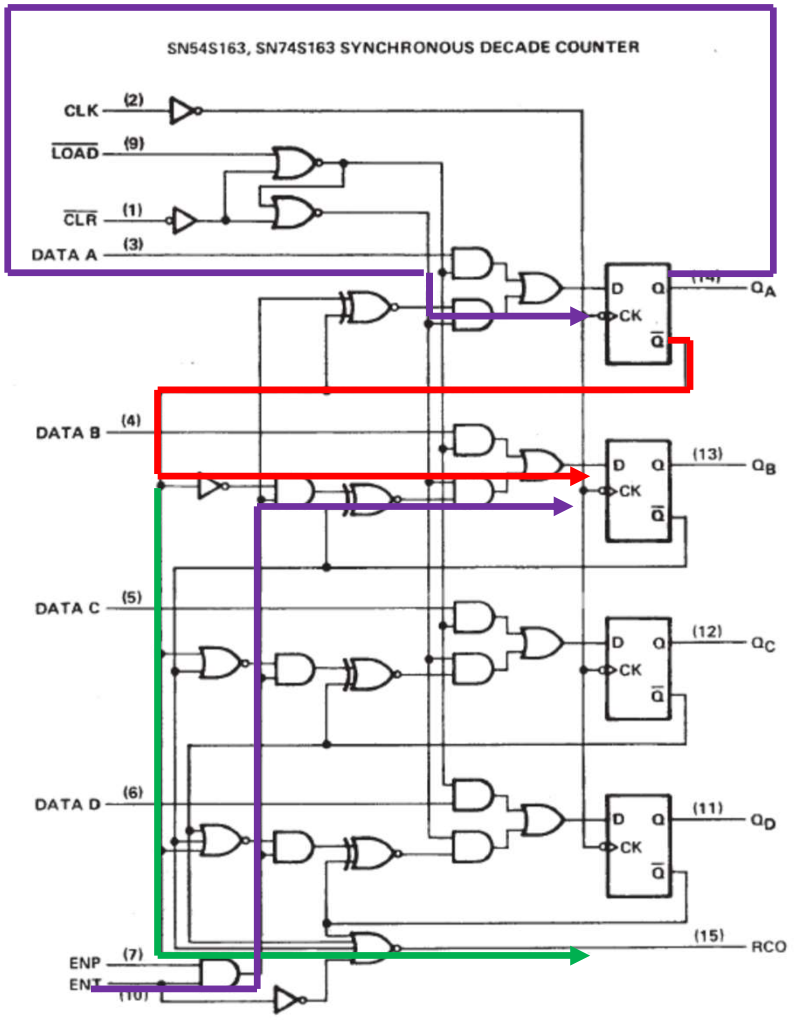


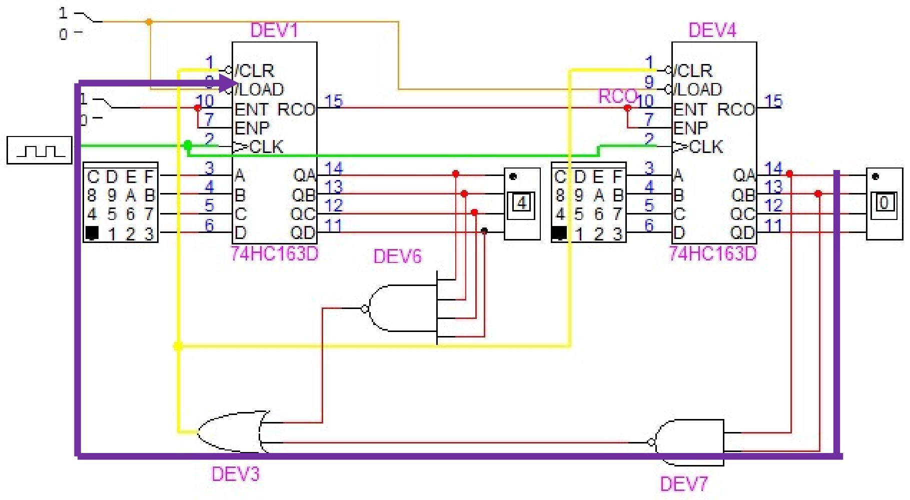
Guião 11_2021
Problema 7 counter mod 64

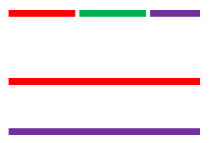


Com base nos componentes 74x163 crie um contador módulo 64. Determine, justificando a máxima frequência de funcionamento do circuito tendo em conta as seguintes especificações temporais:

a. flip-flops que compõem o contador: t_{setup}=10 ns, t_{hold}=4 ns, t_{pHL}=20 ns, t_{pLH}=15 ns;
b. tempo de atraso de uma porta lógica elementar (se usada): t_{porta} = 5 ns.

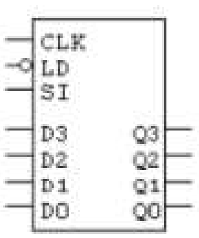


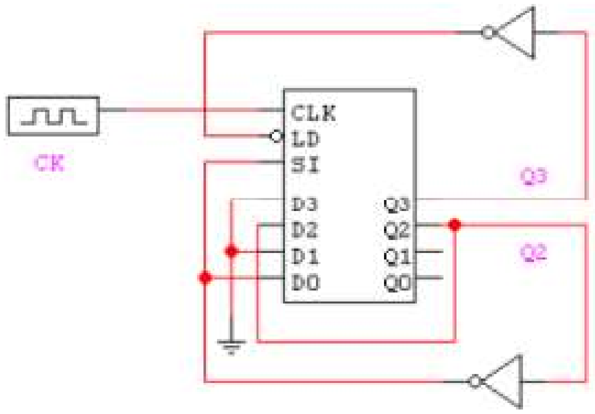




Guião 11_2021
Problema 8

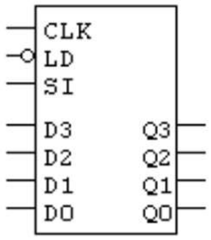
The circuit in Fig. 1a) is a 4-bit shift register, which shifts left (i.e. Q0 → Q3) and has a synchronous active-low load input. Design this circuit with logic gates and D flip-flops.
Create a new project named "CounterShiftReg" in *Quartus Prime* software. Create a new file for a logic diagram called "ShiftReg4.bdf" to implement the shift register. Perform functional simulation. In Fig. 1b) the circuit is used as a special counter. Determine the state diagram of the counter and check your conclusions with simulation in Quartus Prime.



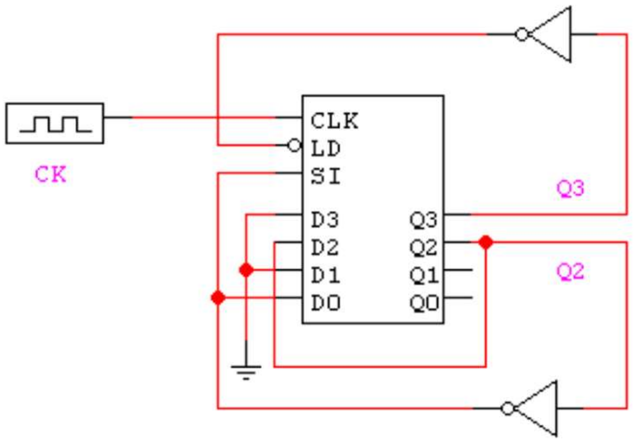


							LOAD			
	Q3	Q2	Q1	Q0	LD=Q3	SI=Q2'	D3=0	D2=Q2	D1=0	D0=Q2'
0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0	1
3	0	0	1	1	0	1	0	0	0	1
7	0	1	1	1	0	0	0	1	0	0
14	1	1	1	0	1	0	0	1	0	0
4	0	1	0	0	0	0	0	1	0	0
8	1	0	0	0	1	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0	1

Guião 11_2021
Problema 8



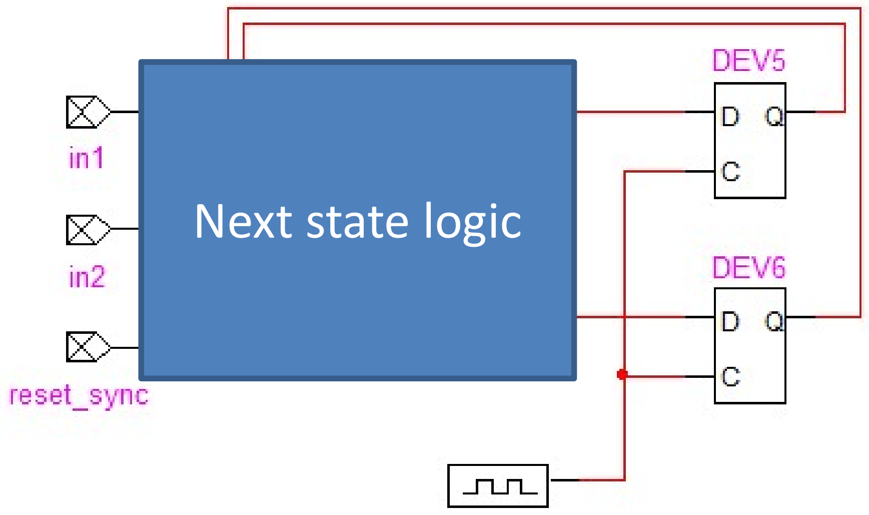
a)

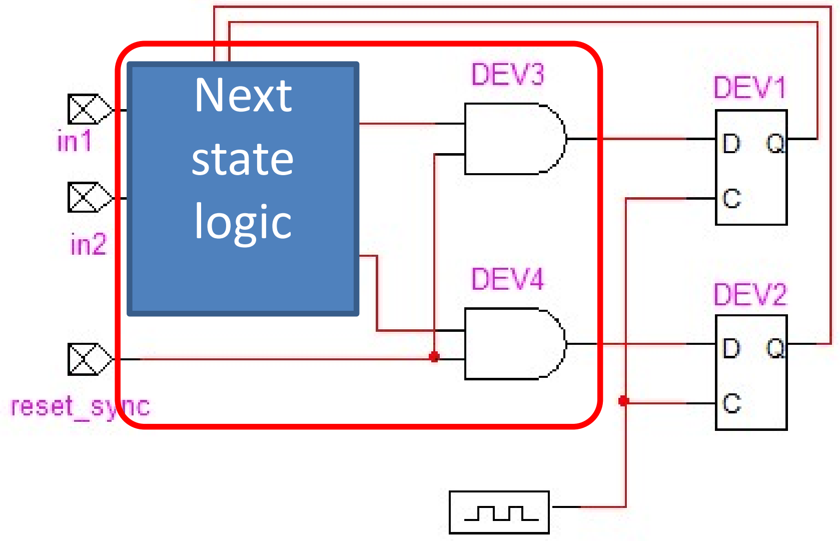


b)

Q3	Q2	Q1	Q0	D3	D2/Q2	D1	D0/Q2'	Si/ Q2'	LD/Q3'
0	0	0	0	0	0	0	1	1	1
0	0	0	1	0	0	0	1	1	1
0	0	1	1	0	0	0	1	1	1
0	1	1	1	0	1	0	0	0	1
1	1	1	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	1	0
0	0	1	1	0	0	0	1	1	1

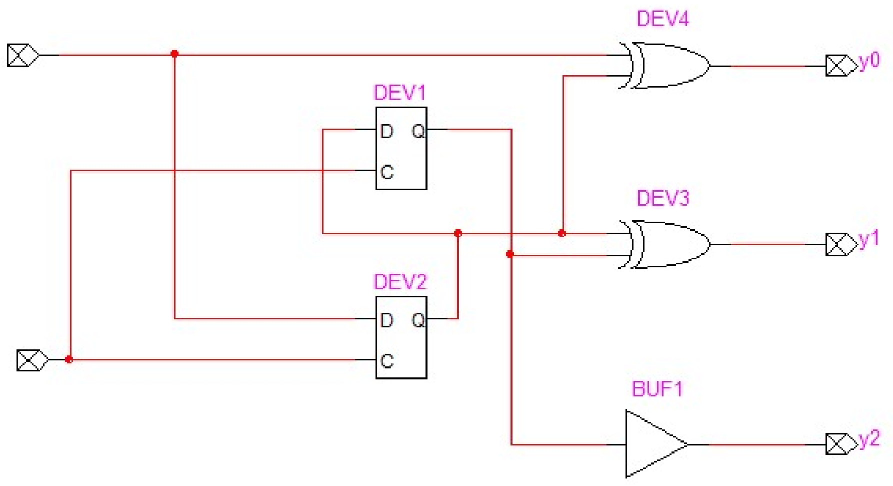
Guião 8
Problema 4

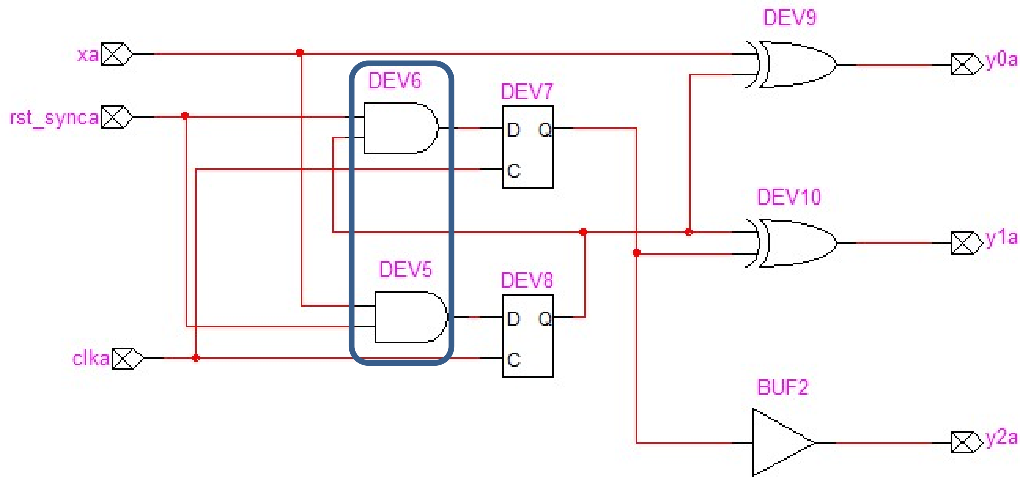




Guião 8

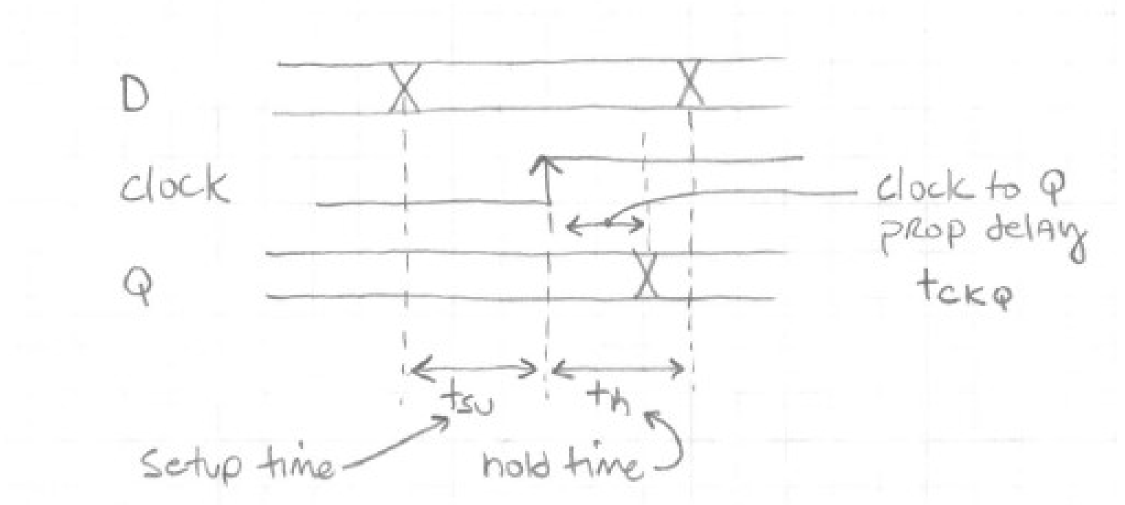
Problema 4





http://web.engr.oregonstate.edu/~traylor/ece 474/lecture_verilog/beamer/tsu_and_th

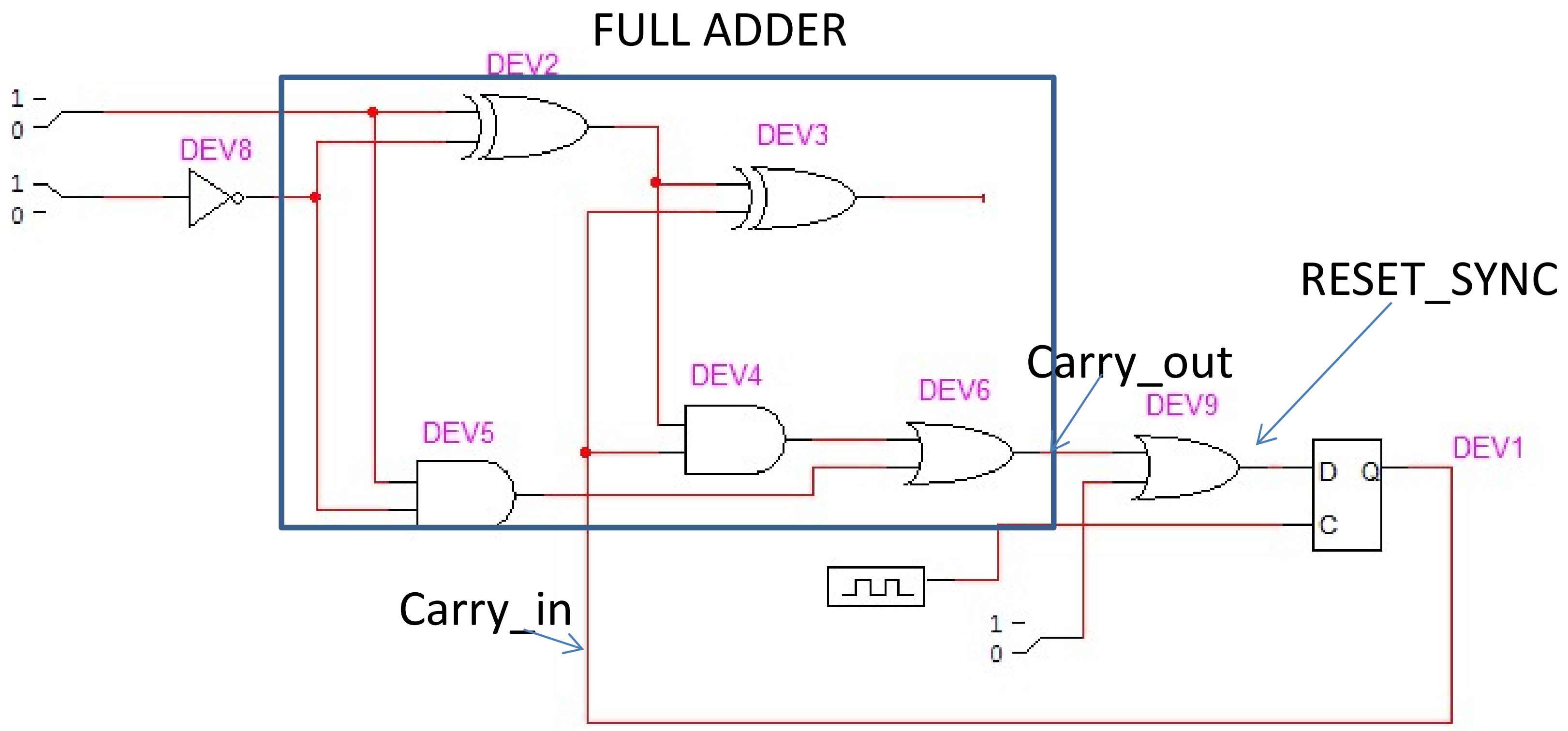
- Considering D-type edge-triggered, Flip Flops (FF's)
- Just before and just after the clock edge, there is a critical time region where the D input must not change.



- The region just before the clock edge is called *setup time* (t_{su})
- The region just after the clock edge is called *hold time* (t_h)

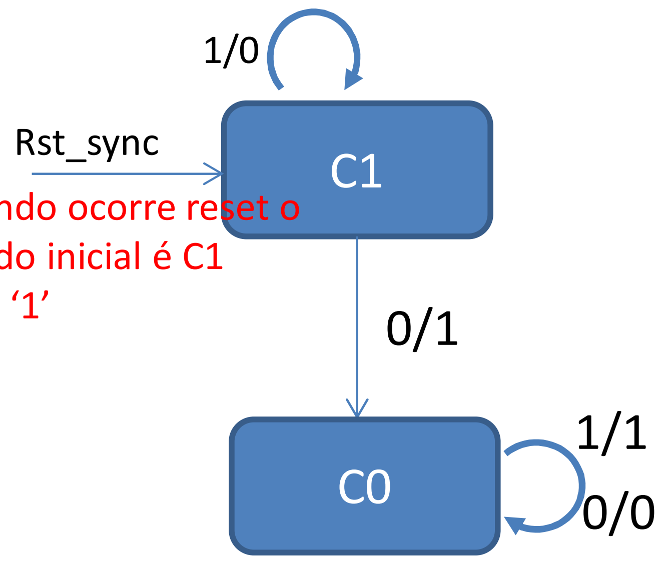
$$T_{clk} > T_{setup} + \max(T_{pHL}, T_{pLH}) + T_p$$
$$T_{hold} < \max(T_{pHL}, T_{pLH}) + T_p$$

Guião 8
Problema 7



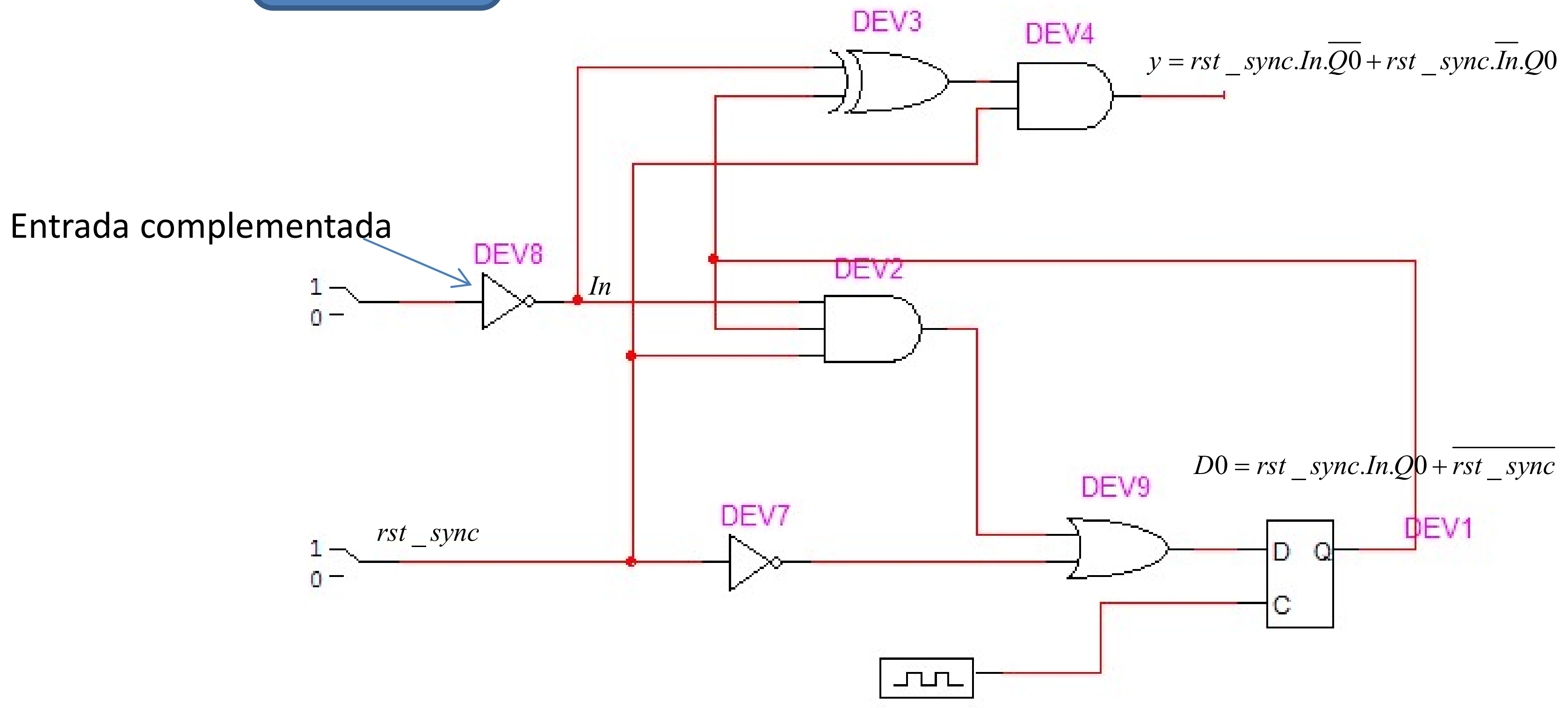
Guião 8
Problema 7

C1 estado carry out = ‘1’
C0 estado carry out = ‘0’



Quando ocorre reset o estado inicial é C1
D0 = ‘1’

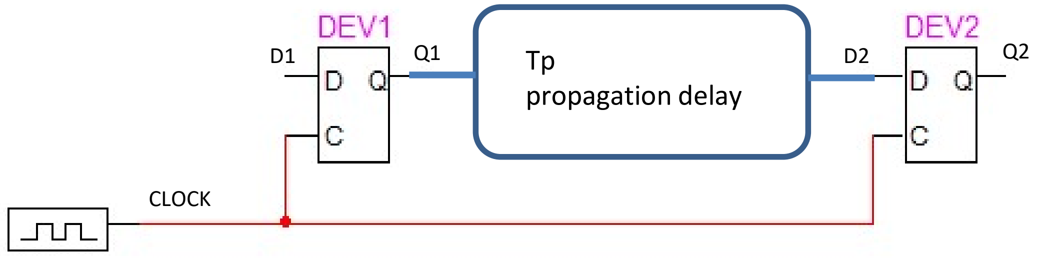
rs_sync	In	Q0	D0	y
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0
0	*	*	1	0

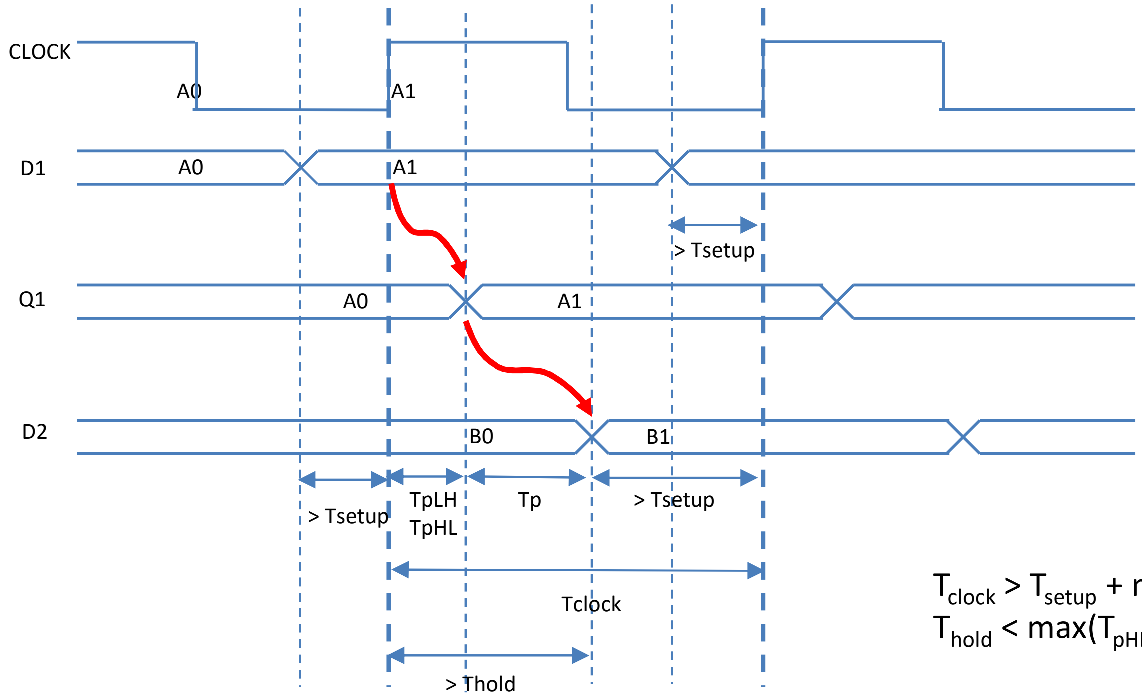


$$y = rst\_sync.In.\overline{Q0} + rst\_sync.\overline{In}.Q0$$

$$D0 = rst\_sync.In.Q0 + \overline{rst\_sync}$$

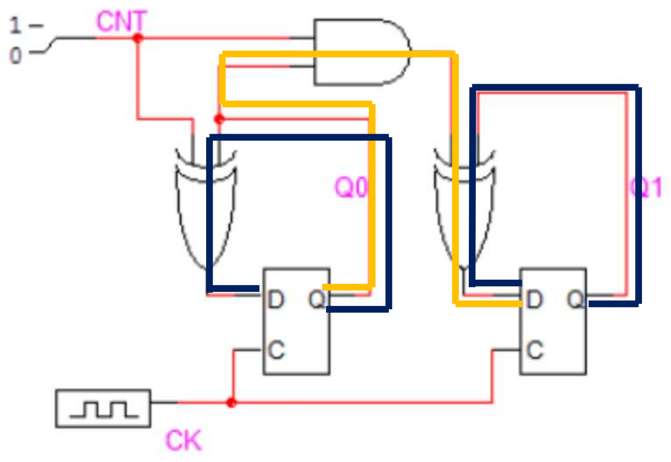
Análise temporal de um circuito síncrono





$$T_{clock} > T_{setup} + \max(T_{pHL},T_{pLH}) + T_p$$
$$T_{hold} < \max(T_{pHL},T_{pLH}) + T_p$$

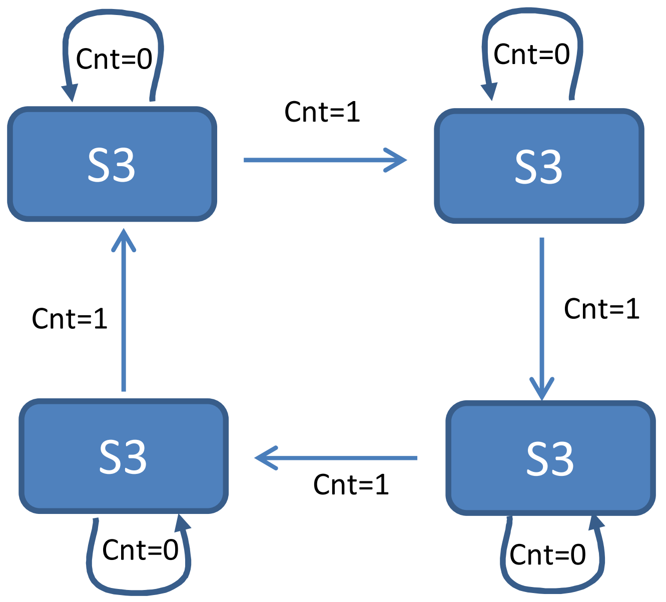
O problema das limitações temporais de um circuito digital resumem-se à análise da propagação dos sinais entre saídas e entradas, do mesmo ou de outro de Flip Flop, consoante se trata de um circuito com *feedback* ou sem *feedback*.

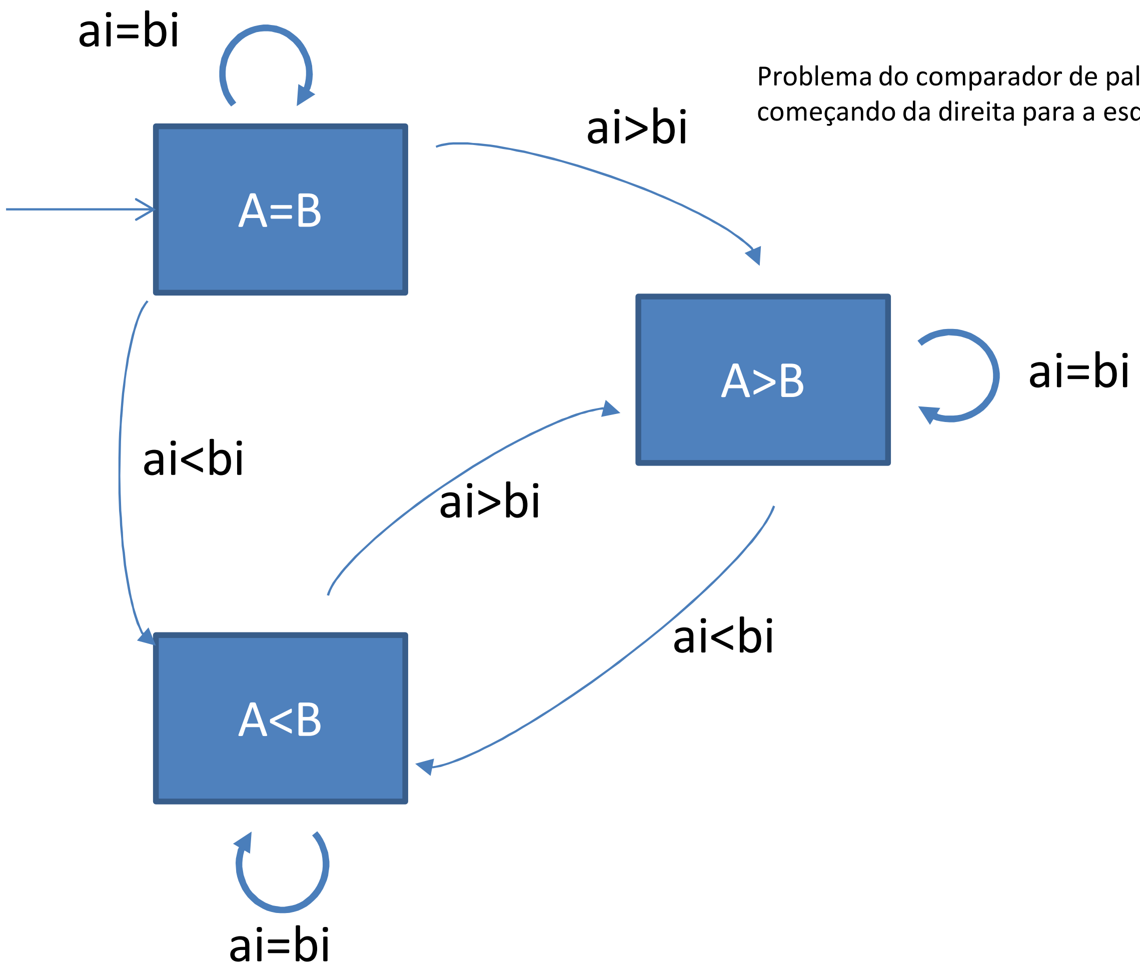


Determine, justificando, atendendo ao esquema lógico, a máxima frequência de funcionamento do circuito sequencial tendo em conta as seguintes especificações temporais: flip-flops: $t_{setup} = 10\,ns, t_{hold} = 4\,ns, t_{pHL} = 20\,ns, t_{pLH} = 15\,ns$; tempo de atraso de cada porta lógica elementar: $t_{porta} = 10\,ns$.

$$T = T_{setup} + 2 * T_{porta} + T_{pHL} \;\; ns$$
$$T = (10 + 2 * 10 + 20) \;\; ns$$
$$T = 50 \;\; ns$$

			CNT=0			CNT=1		
Q1	Q0	S	D1	D0	S*	D1	D0	S*
0	0	0	0	0	0	0	1	1
0	1	1	0	1	1	1	0	2
1	0	2	1	0	2	1	1	3
1	1	3	1	1	3	0	0	0



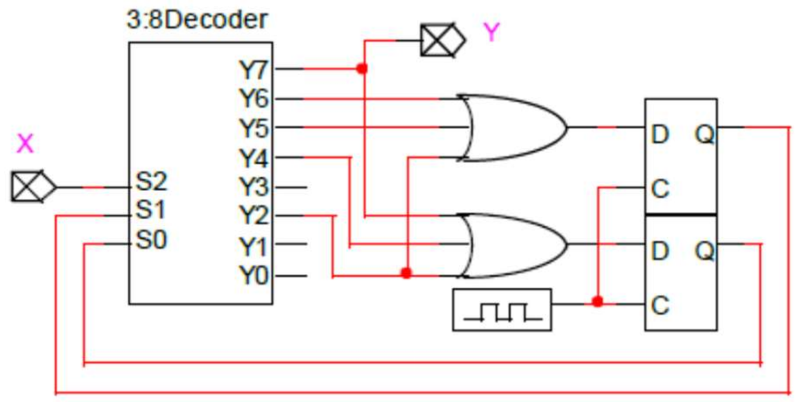


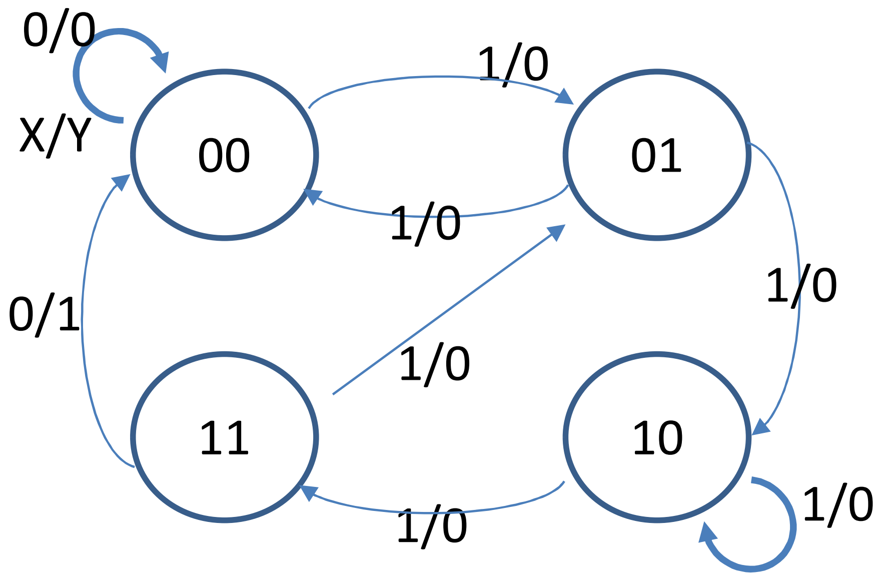
Problema do comparador de palavras de N bits começando da direita para a esquerda

rst	ai	bi	Q0	Q1	D0	D1
0	0	0	0	0	0	0
0	0	0	0	1	0	1
0	0	0	1	0	1	0
0	0	0	1	1	*	*
0	0	1	0	0	0	1
0	0	1	0	1	0	1
0	0	1	1	0	0	1
0	0	1	1	1	0	1
0	1	0	0	0	0	0
0	1	0	0	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	0	0
0	1	1	0	1	0	1
0	1	1	1	0	1	0
0	1	1	1	1	*	*
1	*	*	*	*	0	0

ai,bi Q0,Q1	00	01	11	10
00				
01				1
11	*		*	1
10	1		1	1

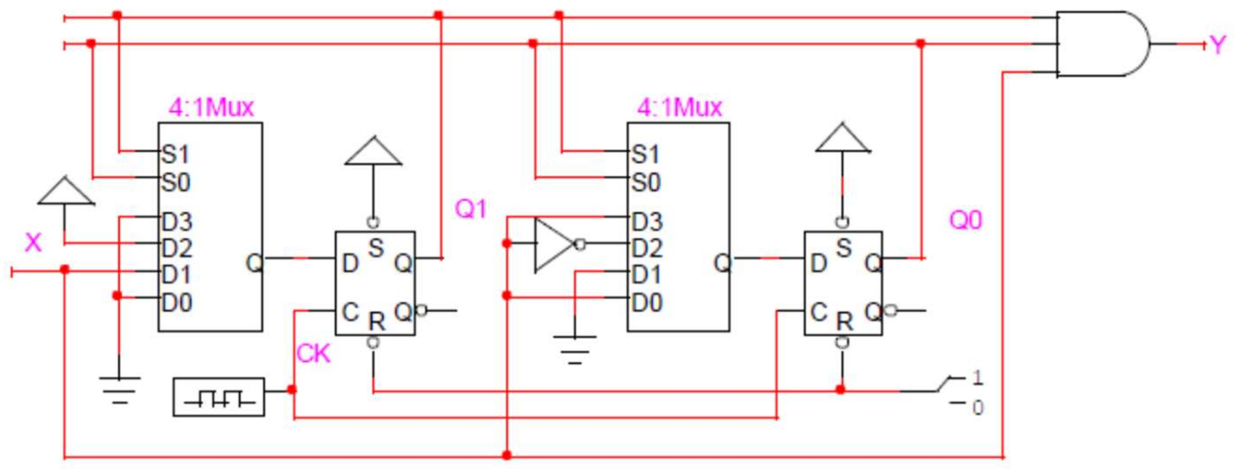
ai,bi Q0,Q1	00	01	11	10
00		1		
01	1	1	1	
11	*	1		
10		1	*	





		X=1		X=0			
Q1	Q0	D1	D0	D1	D0		Y
0	0	0	1	0	0		0
0	1	1	0	0	0		0
1	0	1	0	1	1		0
1	1	0	1	0	0		1

X=	1	0	1	0	1	1	1	0	0	1	1	1	0	1
Y=	0	0	0	0	0	0	0	1	0	0	0	0	1	0


4:1Mux
S1
S0
D3
D2
D1
D0
Q
X
D
S
Q
C
R
Q
Q1
CK
4:1Mux
S1
S0
D3
D2
D1
D0
Q
D
S
Q
C
R
Q
Q0
1
0
Y