



Laboratório 1: Análise e síntese de funções combinacionais de uma única saída.

Bernardo Hoffmann da Silva
Marcos Vinicius Pereira Veloso

4.1

a.

Note-se que, para a obtenção da tabela verdade, basta associarmos o valor 1 de saída aos números que constam na sequência de Fibonacci em tal conjunto delimitado pela entrada, isto é, os valores dados pelo conjunto (1, 2, 3, 5, 8, 13). Com isso, pode-se montar a Tabela Verdade apresentada na Tabela 1.

Tabela 1: Tabela Verdade do problema 4.1.

#	x3	x2	x1	x0	f
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

b.

A partir da tabela verdade (Tabela 1), podemos construir o mapa de Karnaugh reproduzido na Tabela 2, para então poder fazer as devidas simplificações lógicas.

Tabela 2: Representação do Mapa de Karnaugh.

X_3X_2 X_1X_0	00	01	11	10
00	0	0	0	1
01	1	1	1	0
11	1	0	0	0
10	1	0	0	0

Posto isso, note-se que a função lógica de saída simplificada é dada por

$$F = \bar{x}_3\bar{x}_2x_1 + \bar{x}_3\bar{x}_1x_0 + x_2\bar{x}_1x_0 + x_3\bar{x}_2\bar{x}_1\bar{x}_0$$

Desse modo, uma possível representação do diagrama lógico é dado na Figura (1)

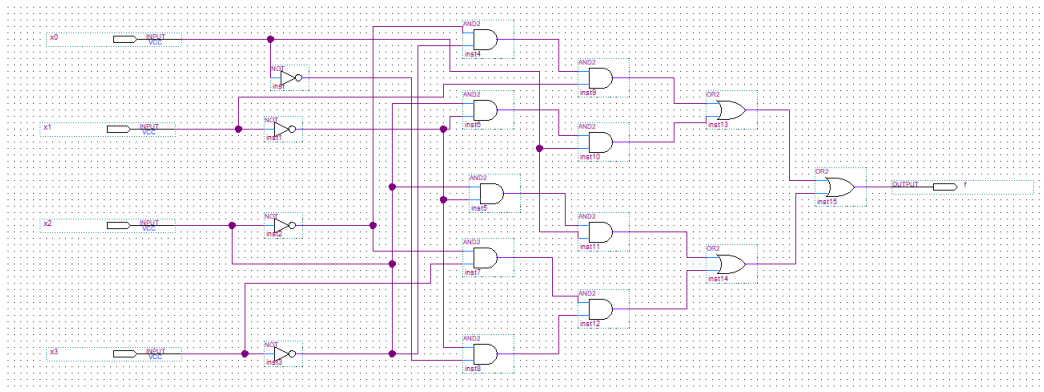


Figura 1: Representação do diagrama lógico no programa computacional proposto.

c.

Posto o circuito montado na aplicação, pode-se então compilar o resultado em forma de diagrama de temporização. Com efeito, o resultado da simulação é representado na Figura 2. Note-se que os resultados obtidos de tal diagrama coincidem com os resultados da Tabela verdade denotada no item 4.1.a.

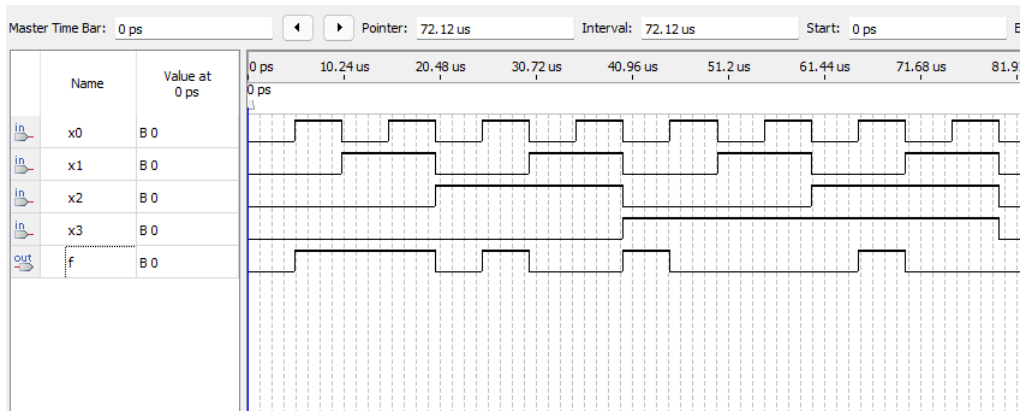


Figura 2: Diagrama de temporização obtido através da simulação do diagrama lógico dado na Figura (1).

4.2

a.

Considere os pontos de potenciais x , y e z denotados na Figura (3).

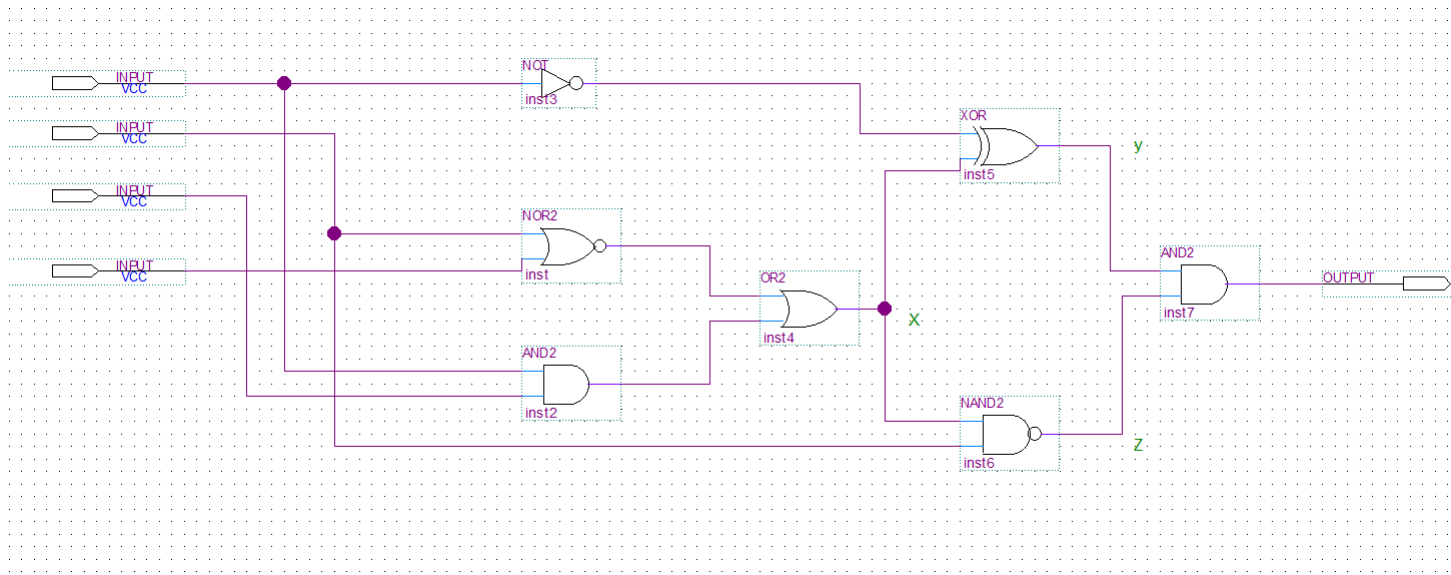


Figura 3: Representação do circuito não simplificado no programa.

Nota-se que, com os pontos denotados na figura, a expressão lógica é tal que $F = zy$, em que z, y são obtidos quando sabido o valor de x . Para o valor lógico de x , tem-se então

$$x = \overline{b + d} + ac = \overline{bd} + ac \quad (1)$$

Por conseguinte, os valores de z e y são dados por

$$z = \overline{bx} = \overline{b(\overline{bd} + ac)} = \overline{abc} = \overline{a} + \overline{b} + \overline{c} \quad (2)$$

$$\begin{aligned}
 y &= \bar{a}\bar{x} + ax = \bar{a}(\overline{\bar{b}\bar{d} + ac}) + a(\bar{b}\bar{d} + ac) \\
 &= \bar{a}b + \bar{a}d + a\bar{b}\bar{d} + ac
 \end{aligned}
 \tag{3}$$

Em que aplicando continuamente o Teorema de De Morgan e utilizando as simplificações necessárias, chegamos no valor especificado de y . Por fim, o valor da função F é então dada por

$$\begin{aligned}
 F &= zy = (\bar{a} + \bar{b} + \bar{c})(\bar{a}b + \bar{a}d + a\bar{b}\bar{d} + ac) \\
 &= \bar{a}b + \bar{a}d + \bar{a}\bar{b}\bar{d} + a\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{c}d + a\bar{b}\bar{c}\bar{d} \\
 F &= \bar{a}b + \bar{a}d + a\bar{b}\bar{d} + a\bar{b}\bar{c}
 \end{aligned}
 \tag{4}$$

b.

Com o intuito de comparar os valores lógicos obtidos do circuito original e do circuito simplificado, dado o circuito do enunciado, pode-se compilar seu diagrama temporal, tal como segue na Figura (4).

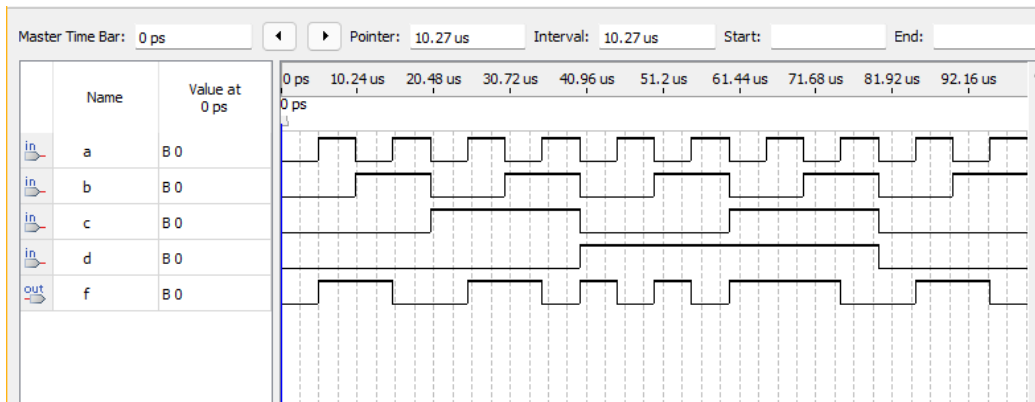


Figura 4: Diagrama de temporização do circuito não simplificado do item 4.2.

c.

Com efeito, o circuito simplificado pode ser montado a partir da Relação (4):

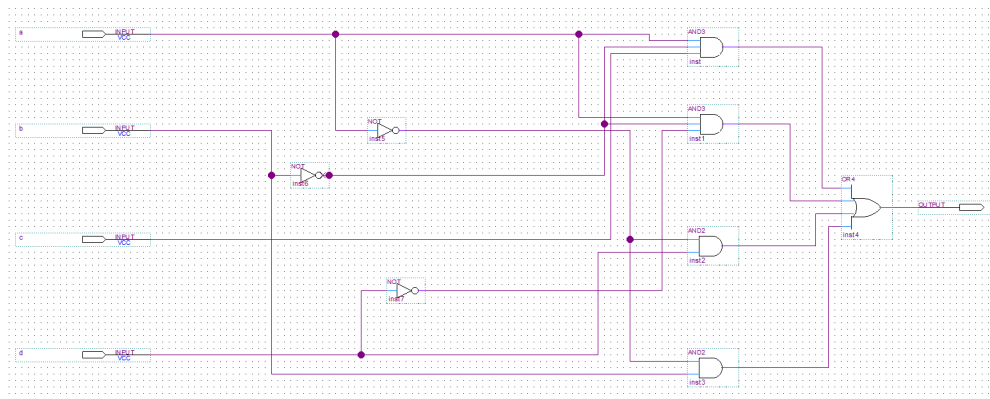


Figura 5: Circuito simplificado do item 4.2.

d.

Por conseguinte, dado o circuito da Figura (5), o diagrama de temporização pode, de modo análogo, ser simulado. Seu resultado segue na Figura (6).

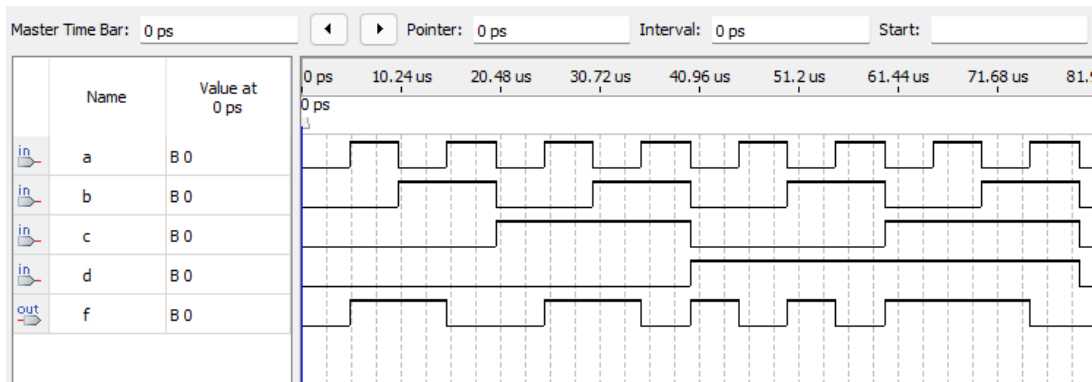


Figura 6: Diagrama de temporização do circuito simplificado do item 4.2.

Note-se que, como esperado, o resultado de saída é o mesmo que o obtido do circuito não simplificado, como constado na Figura (4).

4.3

a.

Note-se que, com os valores representados pelo diagrama temporizado, pode-se montar uma Tabela Verdade da função de saída, tal como segue abaixo.

Tabela 3: Tabela verdade do montado a partir de dados do enunciado.

#	c	b	a	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Com tal tabela, pode-se montar então o diagrama de Karnaugh de tal tabela, como segue.

Tabela 4: Mapa de Karnaugh montado a partir da Tabela Verdade denotada.

a	c b	00	01	11	10
0		0	0	1	0
1		1	1	1	0

Note-se então que, a partir do diagrama, uma expressão lógica simplificada do circuito é dada por:

$$F = a\bar{c} + bc \quad (5)$$

b.

Para a implementação do circuito com o uso exclusivo de portas NAND, uma estratégia de resolução consiste em transformar a função de (5) em termos de produto e complementar usando o Teorema de De Morgan. Assim, chega-se em

$$F = \overline{\overline{a\bar{c} + bc}} = \overline{\overline{a\bar{c}} \cdot \overline{bc}} \quad (6)$$

Com efeito, pode-se então montar o diagrama esquemático do circuito da expressão de (6) como segue:

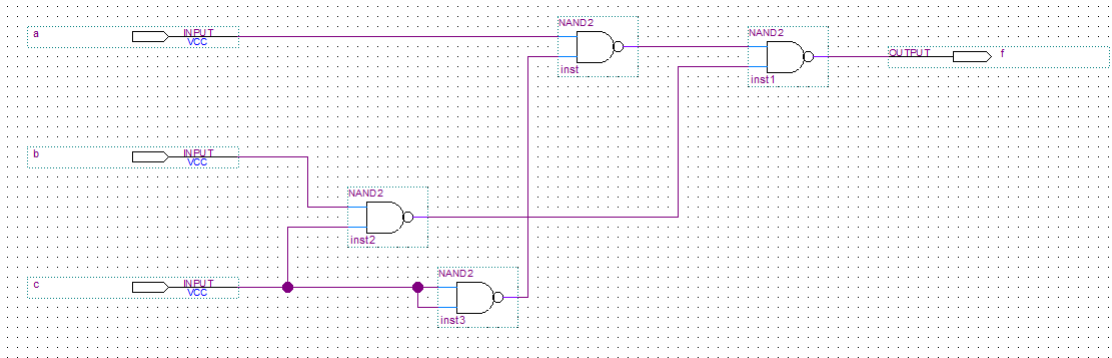


Figura 7: Circuito do enunciado elaborado a partir do uso de portas NAND de duas entradas.

Por conseguinte, com o circuito montado, pode-se obter também a simulação pelo diagrama de temporização.

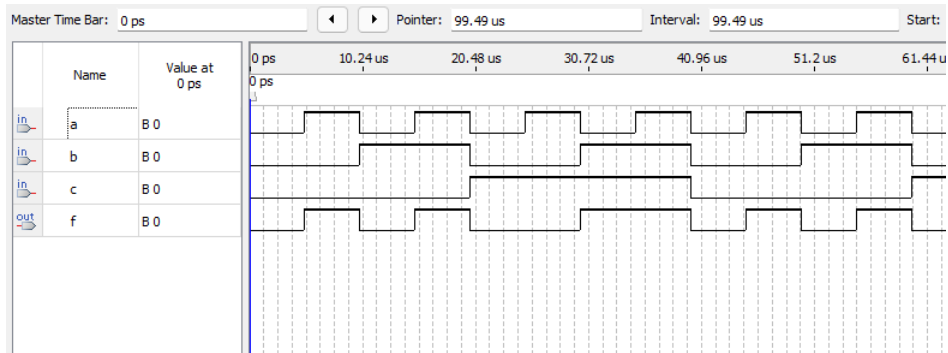


Figura 8: Diagrama de temporização do circuito simplificado da Figura (7).

c.

Nota-se que ocorre um deslocamento temporal na reação de F quando os bits são trocados. Pode-se considerar esse tempo como um tempo de reação do analisador lógico, já que os bits são trocados em um intervalo de tempo suficientemente pequeno de 100 ns.

4.4

Para uma função (MUX)_{4x1} de quatro bits de entrada e dois bits de sinais de seleção, basta associar a cada par de bits da chave seletora a um valor correspondente da entrada. Fazendo uma associação de x_i ao i -ésimo número formado por s_0s_1 , tem-se que a função saída F pode ser dada por, aplicando o Teorema de Morgan:

$$F = \bar{s}_1\bar{s}_0x_0 + \bar{s}_1s_0x_1 + s_1\bar{s}_0x_2 + s_1s_0x_3$$

$$= \bar{s}_1(\bar{s}_0x_0 + s_0x_1) + s_1(\bar{s}_0x_2 + s_0x_3)$$

$$= \bar{s}_1(\bar{s}_0x_0 + s_0x_1) * s_1(\bar{s}_0x_2 + s_0x_3)$$

$$F = \bar{s}_1(\bar{s}_0x_0 * \bar{s}_0x_1) * s_1(\bar{s}_0x_2 * \bar{s}_0x_3)$$

(7)

(8)

(9)

Com efeito, pode-se então montar o diagrama esquemático a partir da função simplificada (9) e obter o diagrama temporal via programa utilizado, como seguem nas imagens abaixo.

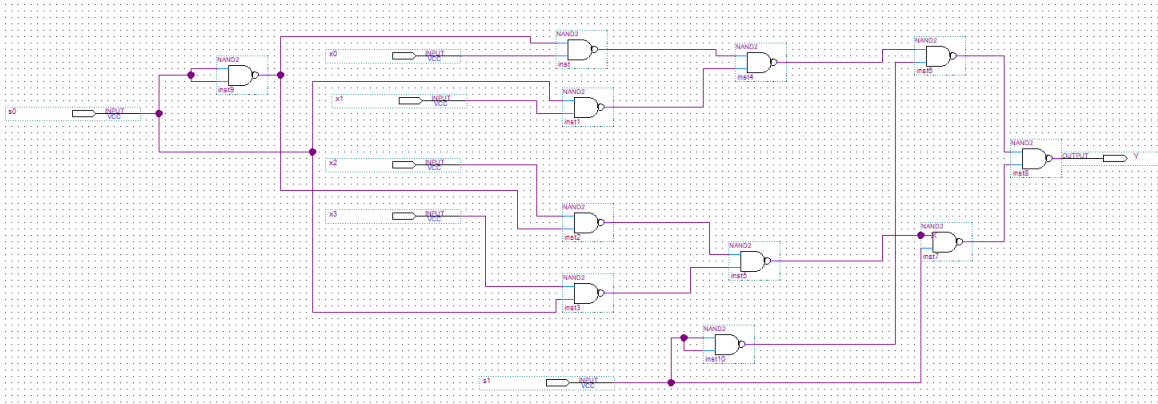


Figura 9: Circuito simplificado com o uso exclusivo de portas NAND de duas entradas.

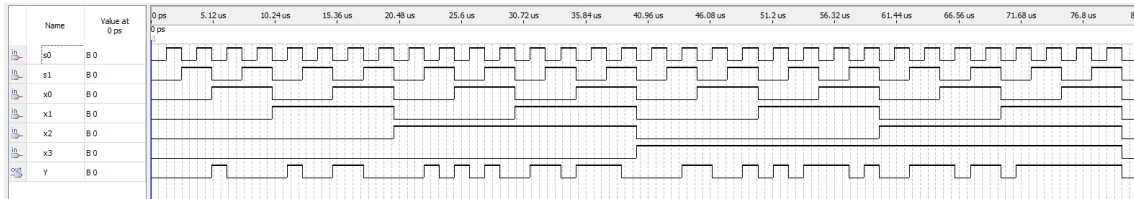


Figura 10: Diagrama de temporização do circuito dado na Figura (9).

4.5

Pode-se montar a seguinte Tabela verdade da função dada:

Tabela 5: Tabela Verdade do problema da Sequência de Fibonacci.

#	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Uma forma de associar tal problema a um $(\text{MUX})_{8 \times 1}$, como há 16 entradas na função original, a ideia consiste em associar cada um dos 3 bits da chave seletora (8 entradas) a cada dupla formada dentre as 16 entradas, a saber de associar os três bits iniciais A, B e C a s_2 , s_1 e s_0 , por exemplo, visto que ABC é o mesmo para cada dupla (0,1), (2, 3), ..., (15, 16). Posto isso, temos então que associar a cada uma das 8 entradas do função multiplexadora um valor lógico de modo que, dado uma entrada do tipo (s_2, s_1, s_0, D) , tenhamos a saída lógica correta.

Dada a Tabela verdade note-se que, para a chave $(s_2 s_1 s_0) = (000)$, o valor de saída é zero independente de $D = 0$ ou $D = 1$. Assim, associamos o valor lógico zero para a primeira entrada do multiplexador. De modo análogo, para $(s_2 s_1 s_0) = (001)$, a saída é sempre 1 independente de D . Então associamos o valor lógico 1 para a segunda entrada do multiplexador. Continuando tal raciocínio, podemos associar os valores lógicos, da primeira à última entrada do multiplexador, $(0, 1, D, D, \bar{D}, D, D, 0)$. Cuja representação e simulação segue nas imagens abaixo.

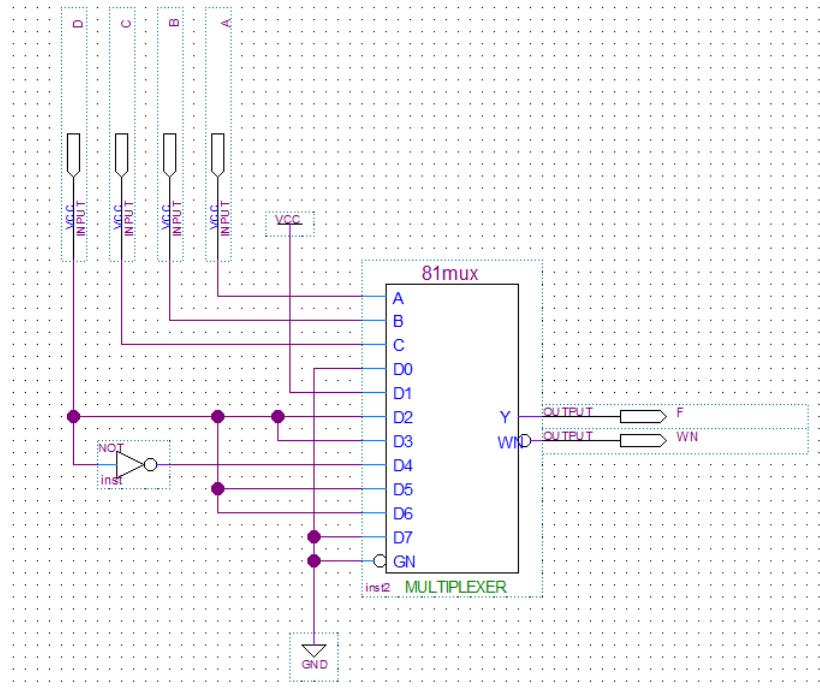


Figura 11: Montagem do circuito com as especificações desejadas do enunciado.

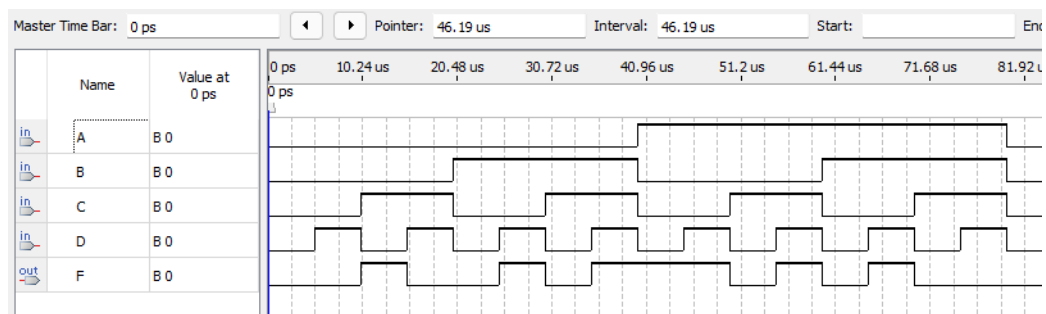


Figura 12: Resultado do diagrama de temporização da função **MUX 8x1** dada na Figura (11).