

EA773 - Laboratório de Circuitos Lógicos

Relatório Roteiro 3

Autor: Mateus Henrique Silva Araújo

RA: 184940

Data: 06/10/2022

1. Projeto 1: Registrador síncrono de 4 bits

1.1. Escopo:

Projeto de um circuito sequencial que implemente um registrador síncrono de 4 bits sensível a borda de subida, utilizando apenas flip-flops tipo D, portas AND, OR e NOT. O registrador deve possuir funcionalidades de carregamento e limpeza síncronas.

1.2. Especificação de alto-nível:

Entradas:

$x(t) \in \{0,1,2,3, \dots, 15\};$

$LD(t) \in \{0,1\};$

$CLR(t) \in \{0,1\}.$

Saídas:

$z(t) \in \{0,1,2,3, \dots, 15\}.$

Estado:

$s(t) \in \{0,1,2,3, \dots, 15\};$

$s(0) = 0.$

Função de transição de estado e saída:

$$s(t+1) = \begin{cases} s(t) & , \text{se } LD(t) = 0 \text{ e } CLR(t) = 0 \\ x(t) & , \text{se } LD(t) = 1 \text{ e } CLR(t) = 0 \\ 0 & , \text{se } CLR(t) = 1 \end{cases}$$

$z(t) = s(t).$

1.3. Especificação binária:

Entradas:

$\underline{x} = (x_3, x_2, x_1, x_0)$ com $x_i \in \{0,1\}$ e $i = 0, \dots, 3;$

$LD(t) \in \{0,1\};$

$CLR(t) \in \{0,1\}.$

Saídas:

$\underline{z}(t) = (z_3, z_2, z_1, z_0)$ com $z_i \in \{0,1\}$ e $i = 0, \dots, 3.$

Estado:

$\underline{s} = (s_3, s_2, s_1, s_0)$ com $s_i \in \{0,1\}$ e $i = 0, \dots, 3$;
 $\underline{s}(0) = 0000$.

Função de transição de estado e saída:

Como cada bit do registrador é independente dos demais, com o intuito de tanto simplificar a minimização do circuito quanto organizar o esquemático do projeto final, criou-se um módulo que implementa um registrador de um bit, com carga e limpeza síncronas. Dessa forma, o circuito completo seria composto por 4 desses módulos, cada um recebendo os sinais de carga e limpeza fornecido ao registrador, bem como o valor de carga respectivo a cada bit. Para tanto, a tabela de transição de estados e excitação foi construída considerando as entradas LD (sinal de carga paralela), CLR (sinal de limpeza), x_i (valor a ser carregado) e q_i (estado atual do flip-flop). Nela também estão dispostos q_i^* (estado posterior que o flip-flop deve assumir) e D_i (valor que deve estar na entrada D do flip-flop para gerar a transição desejada).

Tabela de transição de estados e excitação

LD	CLR	x_i	q_i	q_i^*	D_i
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Como $z(t) = s(t)$, a função de saída do circuito será $\underline{z} = \underline{q}$.

1.4. Minimizações:

D_i :	$x'_i q'_i$	$x'_i q_i$	$x_i q_i$	$x_i q'_i$
$LD' CLR'$	0	1	1	0
$LD' CLR$	0	0	0	0
$LD CLR$	0	0	0	0
$LD CLR'$	0	0	1	1

$$D_i = LD' \cdot CLR' \cdot q_i + LD \cdot CLR' \cdot x_i = CLR' \cdot (LD' \cdot q_i + LD \cdot x_i)$$

*Figura 2:
Diagrama
esquemático
do circuito
sequencial
que
implementa
um
registrador
de 4 bits.
Este circuito
faz uso do
módulo
“registrador
_1_bit”
apresentado
acima.*

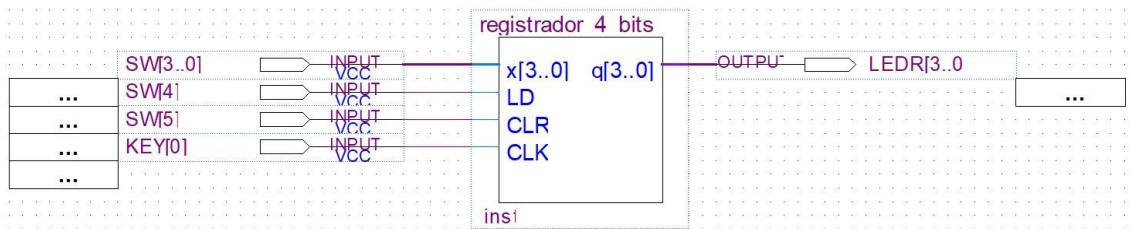


Figura 3: Diagrama esquemático do módulo de teste do registrador de 4 bits. A associação de pinos para entradas e saídas aplicadas nesse circuito seguiu o proposto no roteiro da atividade. Esse módulo foi usado para o teste do circuito do registrador na placa, tendo sido efetuados testes semelhantes ao das simulações feitas.

1.6. Simulações:

Como todos os circuitos desse projeto estão submetidos a clocks manuais, somente simulações funcionais foram efetuadas. O módulo do registrador de um bit passou por uma simulação a qual considerou todas as combinações possíveis das entradas e cujo resultado está apresentado na figura 4. Já o circuito do registrador de 4 bits foi avaliado por meio de uma simulação que considerou diferentes combinações de entrada, que foram escolhidas com o intuito de demonstrar todas as funcionalidades requisitadas no roteiro da atividade. O resultado de tal simulação está disposto na figura 5.

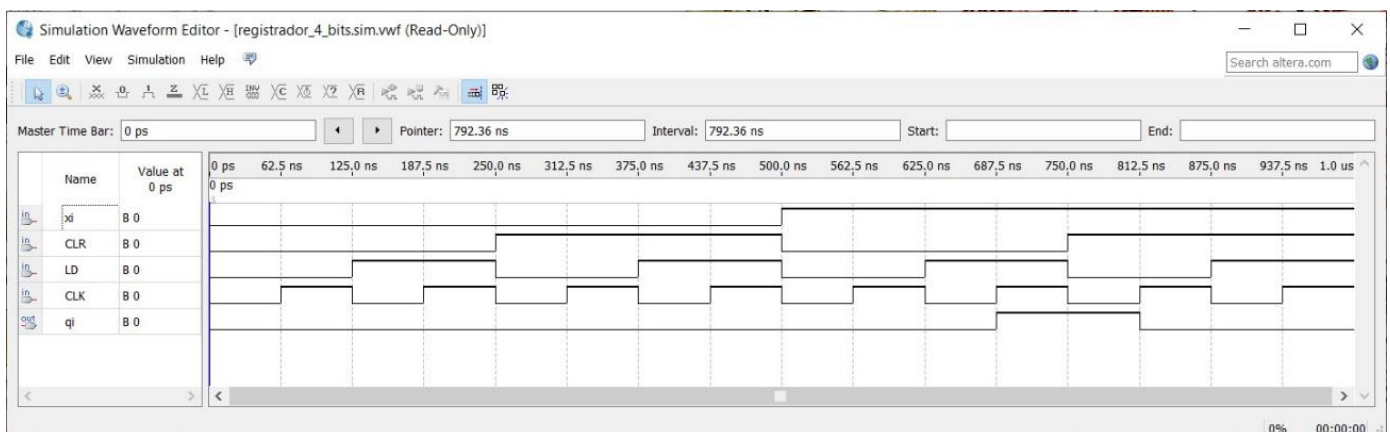


Figura 4: Simulação funcional do módulo “registrador_1_bit”. Nela foram consideradas todas as combinações possíveis das entradas. É válido destacar o intervalo entre 625 ns e 875 ns, uma vez que nele nota-se a capacidade do circuito de efetuar a carga (quando LD=1) e a limpeza (quando CLR=1) síncrona (ou seja, somente na subida de CLK).

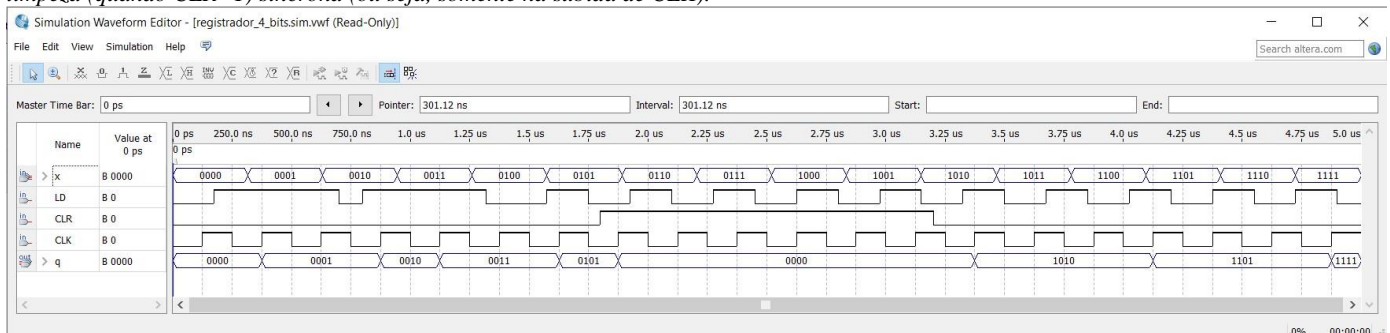


Figura 5: Simulação funcional do circuito do registrador de 4 bits. Nela foram usadas diferentes combinações das entradas de modo a demonstrar todas as funcionalidades desejadas. Durante toda o intervalo de tempo da simulação, nota-se a capacidade do carregamento síncrono do circuito (que ocorre somente se LD=1, CLR=0 e há uma borda de subida do clock). No intervalo entre 1,75 µs e 3,25 µs nota-se não só a capacidade de limpeza síncrona do circuito (que ocorre quando CLR=1 e há uma borda ativa do clock), mas também a preferência da limpeza ante a carga (especificada na função de transição de estado e saída).

2. Projeto 2: Contador síncrono módulo 16

2.1. Escopo:

Projeto de um circuito sequencial que implemente um contador síncrono de módulo 16 sensível a borda de subida, utilizando apenas flip-flops JK, bem como portas AND, OR e NOT. O contador deve possuir funcionalidades de habilitação (enable), carga paralela (load) e limpeza (clear) síncronas, assim como emissão de sinal de término de contagem (terminal count).

2.2. Especificação de alto-nível:

Entradas:

$$x(t) \in \{0,1,2,3, \dots, 15\};$$

$$E(t) \in \{0,1\};$$

$$LD(t) \in \{0,1\};$$

$$CLR(t) \in \{0,1\}.$$

Saídas:

$$z(t) \in \{0,1,2,3, \dots, 15\}.$$

$$TC(t) \in \{0,1\}.$$

Estado:

$$s(t) \in \{0,1,2,3, \dots, 15\};$$

$$s(0) = 0.$$

Função de transição de estado e saída:

$$s(t+1) = \begin{cases} s(t) & , \text{se } E(t) = 0 \\ (s(t) + 1) \bmod 16 & , \text{se } E(t) = 1, LD(t) = 0 \text{ e } CLR(t) = 0 \\ x(t) & , \text{se } E(t) = 1, LD(t) = 1 \text{ e } CLR(t) = 0 \\ 0 & , \text{se } E(t) = 1 \text{ e } CLR(t) = 1 \end{cases}$$
$$z(t) = s(t).$$

2.3. Especificação binária:

Entradas:

$$\underline{x} = (x_3, x_2, x_1, x_0) \text{ com } x_i \in \{0,1\} \text{ e } i = 0, \dots, 3;$$

$$E(t) \in \{0,1\};$$

$$LD(t) \in \{0,1\};$$

$$CLR(t) \in \{0,1\}.$$

Saídas:

$$\underline{z}(t) = (z_3, z_2, z_1, z_0) \text{ com } z_i \in \{0,1\} \text{ e } i = 0, \dots, 3.$$

$$TC(t) \in \{0,1\}.$$

Estado:

$$\underline{s} = (s_3, s_2, s_1, s_0) \text{ com } s_i \in \{0,1\} \text{ e } i = 0, \dots, 3;$$

$$\underline{s}(0) = 0000.$$

Função de transição de estado e saída:

Como o circuito sequencial a ser construído apresenta 8 bits de entrada e mais 4 bits provenientes da realimentação do sistema, uma tabela de transição de estados e excitação completa teria 12 colunas de entradas e, portanto, 2^{12} linhas, contendo todas as combinações possíveis dessas entradas. Isto faz com que a tanto a construção manual dessa tabela quanto a minimização das expressões lógicas provenientes dela seja impraticável.

Para contornar tal empecilho, o circuito projetado foi dividido em duas partes: uma parte que fornece o sinal de realimentação adequado para cada flip-flop (o qual depende do bit que este elemento de memória representa) e uma parte que seleciona qual sinal deve ser fornecido aos flip-flops de acordo com a combinação das entradas E , LD e CLR fornecidas ao contador. Dessa forma, para implementar a parte do sistema que fornece a realimentação adequada a cada flip-flop, usamos a tabela de transição de estados e excitação de um contador de módulo 16 padrão. Nela estão apresentados os estados atuais de cada flip-flop (q_3, q_2, q_1, q_0), os estados futuros que devem ser assumidos ($q_3^*, q_2^*, q_1^*, q_0^*$) e as entradas J e K que devem ser fornecidas para provocar a transição desejada.

Tabela de transição de estados e excitação de um contador módulo 16

q_3	q_2	q_1	q_0	q_3^*	q_2^*	q_1^*	q_0^*	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	1	0	x	0	x	0	x	1	x
0	0	0	1	0	0	1	0	0	x	0	x	1	x	x	1
0	0	1	0	0	0	1	1	0	x	0	x	x	0	1	x
0	0	1	1	0	1	0	0	0	x	1	x	x	1	x	1
0	1	0	0	0	1	0	1	0	x	x	0	0	x	1	x
0	1	0	1	0	1	1	0	0	x	x	0	1	x	x	1
0	1	1	0	0	1	1	1	1	x	x	0	x	0	1	x
0	1	1	1	1	0	0	0	x	0	x	1	x	1	x	1
1	0	0	0	1	0	0	1	x	0	0	x	0	x	1	x
1	0	0	1	1	0	1	0	x	0	0	x	1	x	x	1
1	0	1	0	1	0	1	1	x	0	0	x	x	0	1	x
1	0	1	1	1	1	0	0	x	0	1	x	x	1	x	1
1	1	0	0	1	1	0	1	x	0	x	0	0	x	1	x
1	1	0	1	1	1	1	0	x	0	x	0	1	x	x	1
1	1	1	0	1	1	1	1	x	0	x	0	x	0	1	x
1	1	1	1	0	0	0	0	x	1	x	1	x	1	x	1

Esta tabela será utilizada para definir a entrada de realimentação (R_i) a ser fornecida ao circuito combinacional que seleciona a entrada de cada flip-flop. Esse circuito atua, basicamente, como um multiplexador, sendo os sinais E , LD e CLR suas entradas de seleção. Para definirmos as demais entradas a serem selecionadas, precisamos analisar o comportamento desejado mediante cada possibilidade de seleção:

Tabela verdade do seletor de entrada

CLR	LD	E	J_i	K_i
0	0	0	0	0

Minimização das funções do circuito seletor										
J_i :	$LD'E'$	$LD'E$	$LD E$	$LD E'$		K_i :	$LD'E'$	$LD'E$	$LD E$	$LD E'$
CLR'	0	R_i	x_i	0		CLR'	0	R_i	x_i'	0
CLR	0	0	0	0		CLR	0	1	1	0
$J_i = CLR' \cdot LD' \cdot E \cdot R_i + CLR' \cdot LD \cdot E \cdot x_i$ $J_i = CLR' \cdot E \cdot (LD \cdot x_i + LD' \cdot R_i)$						$K_i = CLR \cdot E + CLR' \cdot LD' \cdot E \cdot R_i + CLR' \cdot LD \cdot E \cdot x_i'$ $K_i = E (CLR + LD' \cdot R_i + LD \cdot x_i')$				

2.5. Esquemático do circuito:

Usando das funções minimizadas acima, o circuito seletor de entrada foi implementado e encapsulado no módulo “transmite_entrada”. O diagrama esquemático desse circuito está apresentado na figura 6. Dessa forma, para implementar o circuito do contador com as funcionalidades desejadas, bastou configurar as entradas de realimentação para cada flip-flop, de acordo com as funções minimizadas na seção anterior, e ligá-la, juntamente com as demais entradas do circuito, no módulo “transmite_entrada” correspondente, assim como mostrado no diagrama final do circuito (figura 7). Além disso, com intuito de testar o circuito implementado na FPGA, o próprio contador foi encapsulado em um módulo de teste, cujo esquemático é mostrado na figura 8, o qual foi carregado na placa e passou pelos mesmos testes feitos nas simulações.

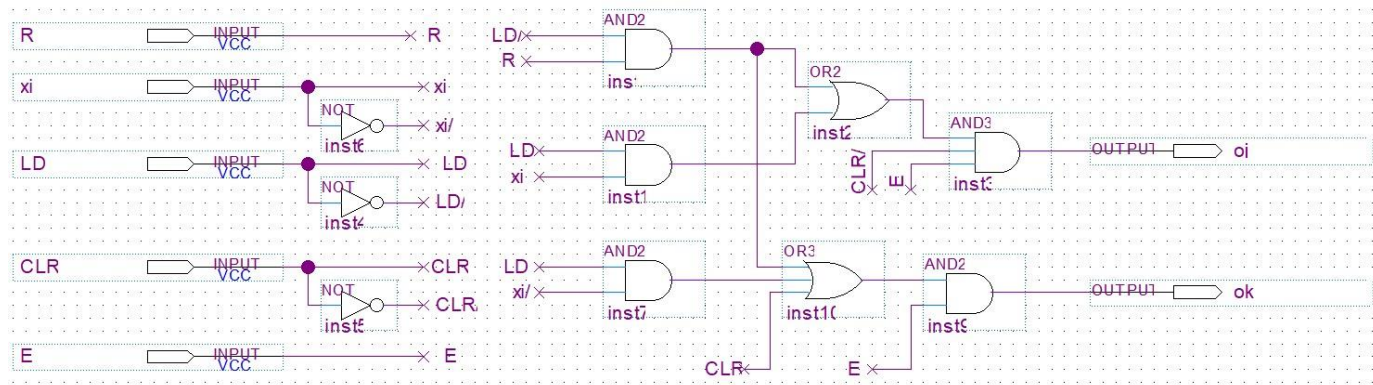


Figura 6: Diagrama esquemático do módulo “transmite_entrada”, que implementa o circuito seletor das entradas J_i e K_i do flip-flop i no circuito do contador. Este circuito foi implementado usando as equações de minimização apresentadas na seção anterior.

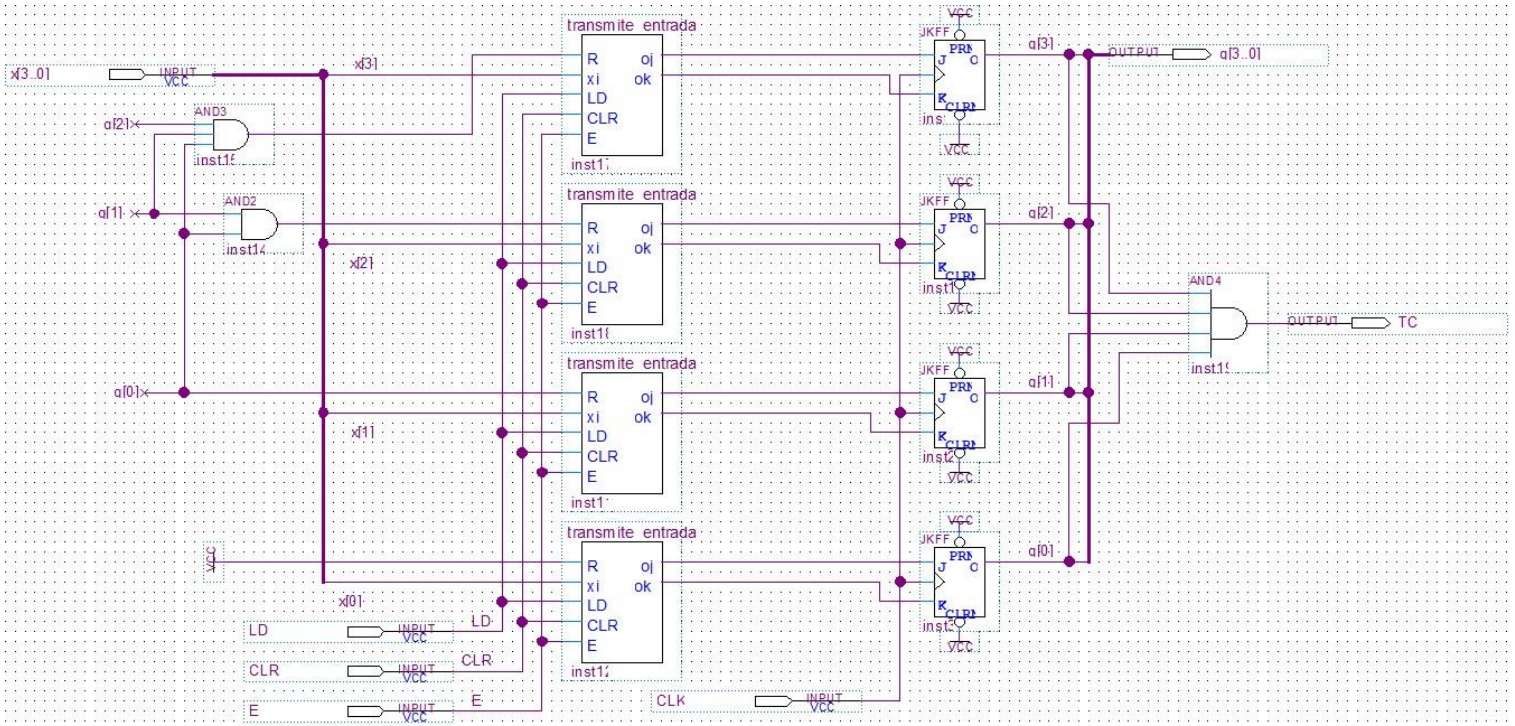


Figura 7: Diagrama esquemático do circuito final do contador de módulo 16. Assim como descrito anteriormente, ele consiste basicamente em uma parte que seleciona o sinal adequado a ser fornecido aos flip-flops de acordo com as entradas de load, clear e enable recebidas (componentes do módulo “transmite_entrada”), e uma parte que calcula o sinal de realimentação próprio a ser usado em cada caso de seleção.

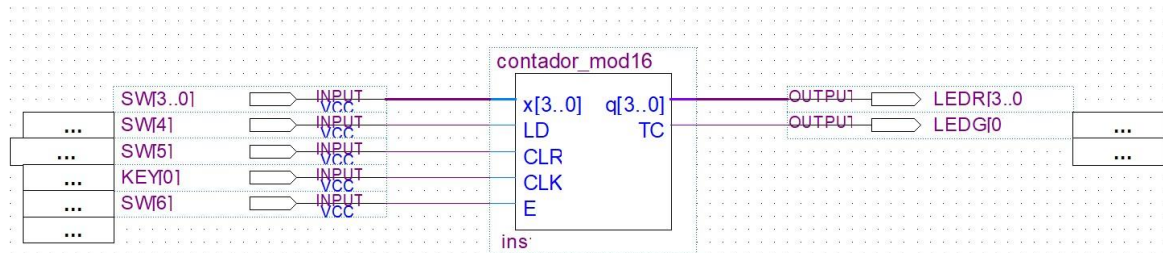


Figura 8: Diagrama esquemático do módulo de teste do contador de módulo 16. A associação de pinos para entradas e saídas aplicadas nesse circuito seguiu o proposto no roteiro da atividade. Esse módulo foi usado para o teste do circuito do registrador na placa, tendo sido efetuados testes semelhantes ao das simulações feitas. Esse módulo também foi usado no terceiro projeto desse roteiro.

2.6. Simulações:

Como todos os circuitos desse projeto estão submetidos a clocks manuais, somente simulações funcionais foram efetuadas. O módulo “transmite_entrada” passou por uma simulação a qual considerou todas as combinações possíveis das entradas e cujo resultado está apresentado na figura 9. Já o circuito do contador módulo 16 foi avaliado por meio de simulações que consideraram diferentes combinações de entrada, escolhidas com o intuito de demonstrar todas as funcionalidades requisitadas no roteiro da atividade. Os resultados delas estão dispostos nas figuras 10 a 13.

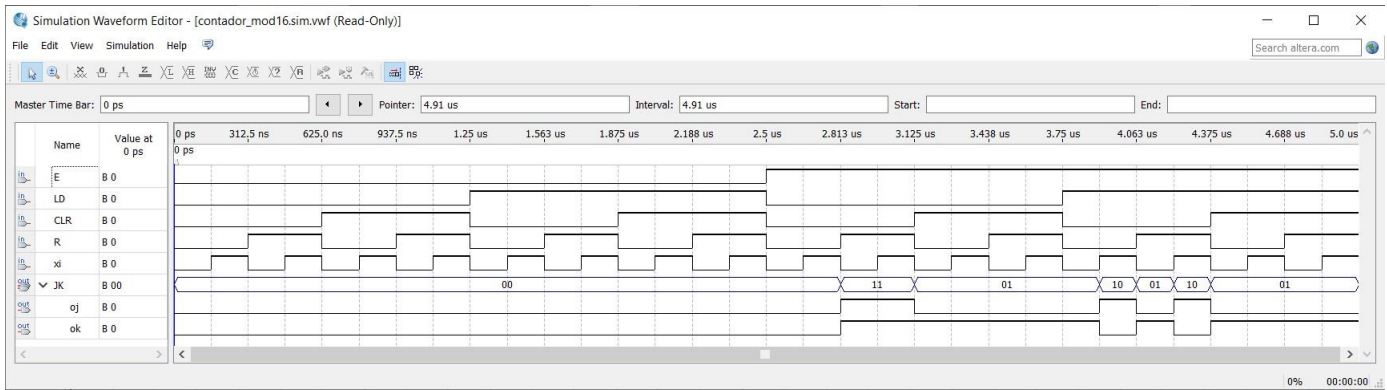


Figura 9: Simulação funcional do módulo “transmite_entrada”. Nela foram consideradas todas as combinações possíveis das entradas do módulo. Nota-se que as saídas obtidas são as mesmas que as previstas nas tabelas verdade usadas.

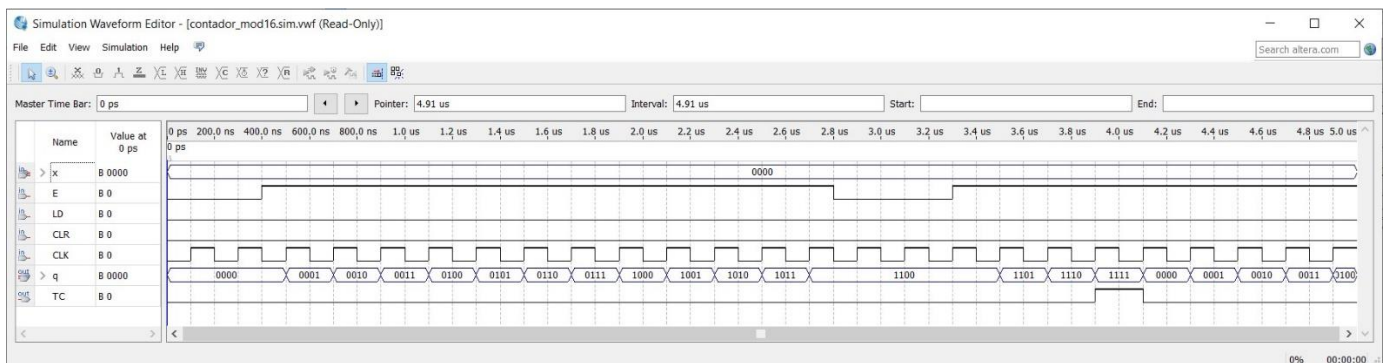


Figura 10: Simulação funcional do circuito do contador de módulo 16. Essa simulação busca demonstrar o comportamento normal do contador, que muda de estado sempre que há uma borda ativa do clock e $E = 1$, pausando em um mesmo estado se $E = 0$. Também é possível notar que o sinal TC funciona corretamente, pois só fica em nível alto quando o contador atinge o estado 15.

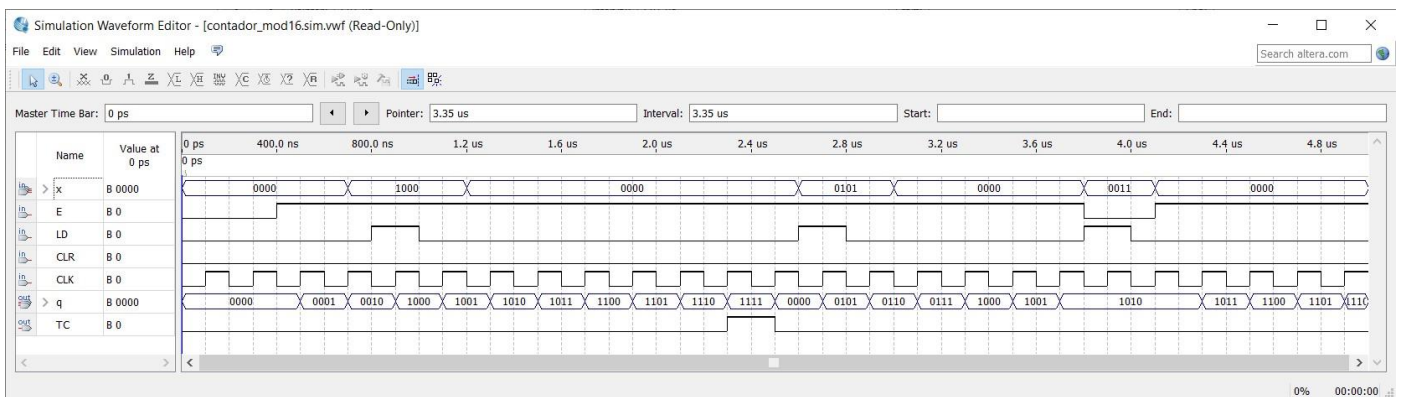


Figura 11: Simulação funcional do circuito do contador de módulo 16. Essa simulação busca demonstrar a funcionalidade de carregamento síncrono do circuito e sua interação com a contagem. Nota-se que a carga só ocorre se $LD = E = 1$ e há uma borda ativa do clock, inserindo o valor em x nos flip-flops corretamente e, em seguida, seguindo a contagem normal. Percebe-se também que o sinal TC continua coerente com o esperado.

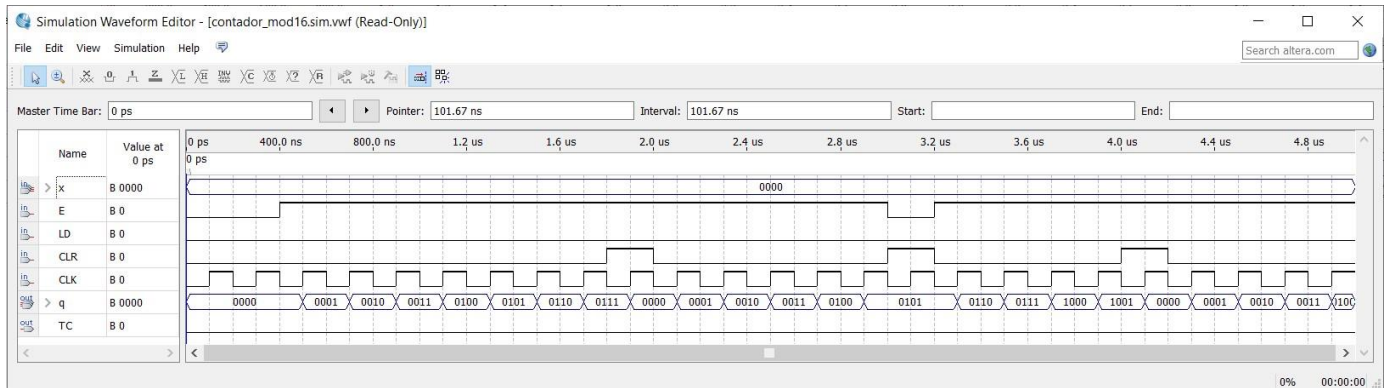


Figura 12: Simulação funcional do circuito do contador de módulo 16. Essa simulação busca demonstrar a funcionalidade de limpeza síncrona do circuito e sua interação com a contagem. Nota-se que ela só ocorre se $CLR = E = 1$ e há uma borda ativa do clock, zerando a contagem atual e, em seguida, seguindo normalmente.

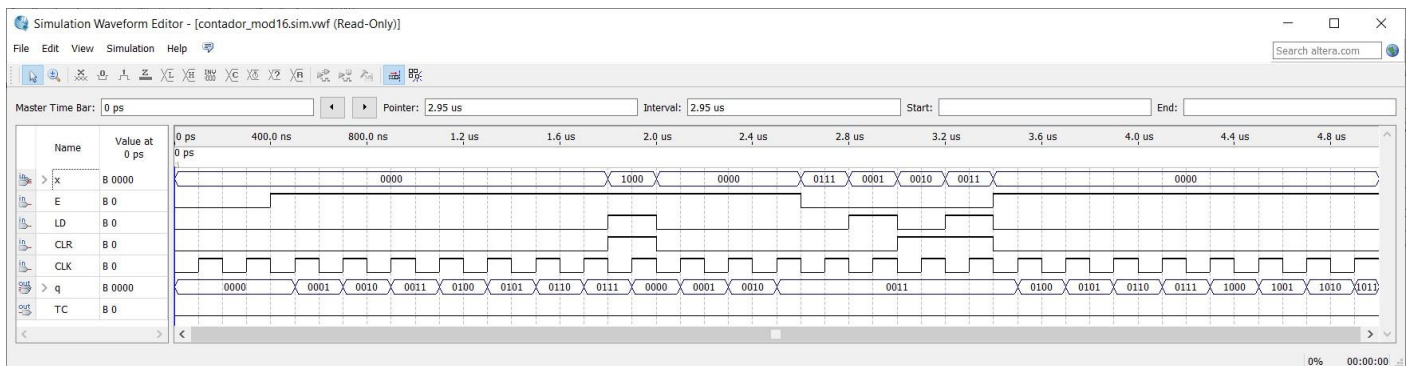


Figura 13: Simulação funcional do circuito do contador de módulo 16. Essa simulação busca demonstrar a interação entre as funcionalidades de carga e limpeza síncrona. Nota-se que a limpeza tem preferência sobre a carga, bem como ambas só acontecem se $E = 1$, assim como descrito no roteiro da atividade.

3. Projeto 3: Contador síncrono módulo 256

3.1. Escopo:

Projeto de um circuito sequencial que implemente um contador síncrono de módulo 256, utilizando, para isso, de dois contadores síncronos de módulo 16. De modo análogo aos contadores de menor ordem, o contador deve possuir funcionalidades de habilitação (enable), carga paralela (load) e limpeza (clear) síncronas, assim como emissão de sinal de término de contagem (terminal count).

3.2. Especificação de alto-nível:

Entradas:

$x(t) \in \{0,1,2,3, \dots, 255\}$;

$E(t) \in \{0,1\}$;

$LD(t) \in \{0,1\}$;

$CLR(t) \in \{0,1\}$.

Saídas:

$z(t) \in \{0,1,2,3, \dots, 255\}$.

$$TC(t) \in \{0,1\}.$$

Estado:

$$s(t) \in \{0,1,2,3, \dots, 255\};$$

$$s(0) = 0.$$

Função de transição de estado e saída:

$$s(t+1) = \begin{cases} s(t) & , \text{se } E(t) = 0 \\ (s(t) + 1) \bmod 256 & , \text{se } E(t) = 1, LD(t) = 0 \text{ e } CLR(t) = 0 \\ x(t) & , \text{se } E(t) = 1, LD(t) = 1 \text{ e } CLR(t) = 0 \\ 0 & , \text{se } E(t) = 1 \text{ e } CLR(t) = 1 \end{cases}$$

$$z(t) = s(t).$$

3.3. Especificação binária:

Entradas:

$$\underline{x} = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) \text{ com } x_i \in \{0,1\} \text{ e } i = 0, \dots, 7;$$

$$E(t) \in \{0,1\};$$

$$LD(t) \in \{0,1\};$$

$$CLR(t) \in \{0,1\}.$$

Saídas:

$$\underline{z}(t) = (z_7, z_6, z_5, z_4, z_3, z_2, z_1, z_0) \text{ com } z_i \in \{0,1\} \text{ e } i = 0, \dots, 7.$$

$$TC(t) \in \{0,1\}.$$

Estado:

$$\underline{s} = (s_7, s_6, s_5, s_4, s_3, s_2, s_1, s_0) \text{ com } s_i \in \{0,1\} \text{ e } i = 0, \dots, 7;$$

$$\underline{s}(0) = 00000000.$$

Função de transição de estado e saída:

Como serão utilizadas duas componentes do módulo “contador_mod16” implementado anteriormente, as funções de transição de estado e de excitação serão as mesmas. Resta apenas estabelecer os sinais que serão enviados para as entradas de cada uma dessas componentes, bem como os sinais de saídas. Para tanto, vamos iniciar definindo um desses contadores de módulo 16 como menos significativo (LS) e o outro como mais significativo (MS). Dessa forma, temos que:

- $\underline{z}(t) = (z_7, z_6, z_5, z_4, z_3, z_2, z_1, z_0)$, sendo $z_i = q_{l_i}$, com $i = 0, \dots, 3$ e q_{l_i} representando as saídas q do contador LS, ou $z_k = q_{m_i}$, com $k = 4, \dots, 7$ e $i = 0, \dots, 3$ e q_{m_i} representando as saídas q do contador MS;
- $TC = TC_m \cdot TC_l$, sendo TC_m o sinal TC de MS e TC_l o sinal TC de LS;
- Para ambos (MS e LS), os sinais LD , CLR e CLK são os mesmos que fornecidos como entrada do contador final;
- Para LS, o sinal E_l (enable desse contador) é o mesmo que o fornecido como entrada do contador final;
- Para MS, o sinal E_m (enable desse contador) deve seguir a tabela verdade abaixo:

Tabela Verdade do sinal E_m				
LD	E	TC_l	CLR	E_m
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

3.4. Minimizações:

$E_m:$	$TC_l'CLR'$	$TC_l'CLR$	$TC_l CLR$	$TC_l CLR'$
$LD'E'$	0	0	0	0
$LD'E$	0	1	1	1
$LD E$	1	1	1	1
$LD E'$	0	0	0	0

$$E_m = E(LD + TC_l + CLR)$$

3.5. Esquemático do circuito:

O circuito do contador de módulo 256 foi implementado seguindo o descrito anteriormente e usando, também, da função minimizada acima. O diagrama esquemático do circuito resultante está apresentado na figura 14. Além disso, com intuito de testar o projeto implementado na FPGA, o próprio contador foi encapsulado em um módulo de teste, cujo esquemático é mostrado na figura 15, o qual foi carregado na placa e passou pelos mesmos testes feitos nas simulações.

capacidade de um único contador de módulo 16. Infelizmente, devido a limitações de software, não é possível apresentar todos os 256 estados em uma única simulação.

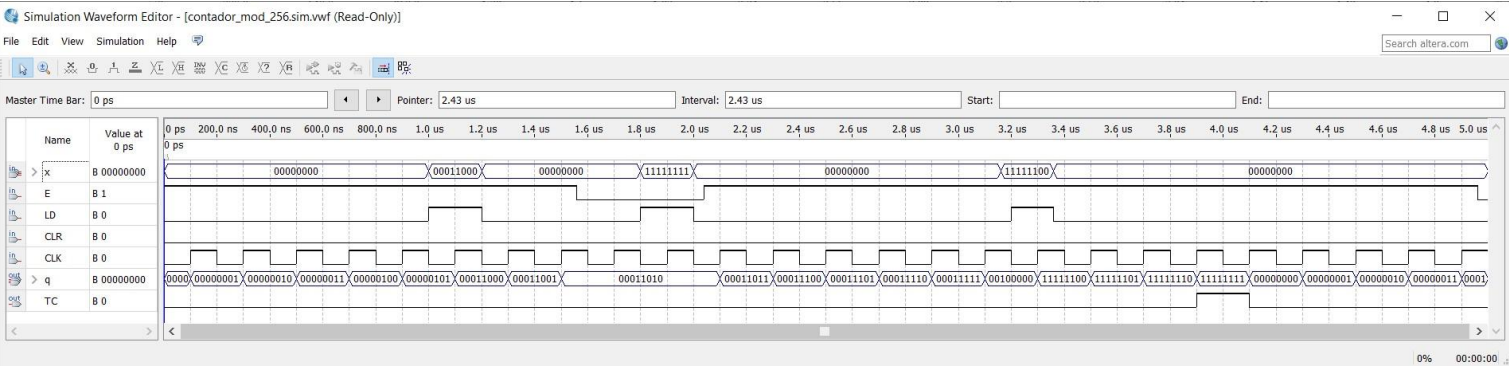


Figura 17: Simulação funcional do circuito do contador de módulo 256. Essa simulação busca mostrar a funcionalidade de carregamento síncrono do contador, o qual insere o valor de x nele mediante a habilitação do sinal de enable e a ocorrência da borda ativa do clock. Além disso, ao observar o intervalo entre 3,8 μ s e 4,2 μ s, nota-se que a saída TC funciona adequadamente.



Figura 18: Simulação funcional do circuito do contador de módulo 256. Essa simulação busca mostrar a funcionalidade de limpeza síncrona do contador, a qual zera a contagem mediante a habilitação do sinal de enable e a ocorrência da borda ativa do clock.

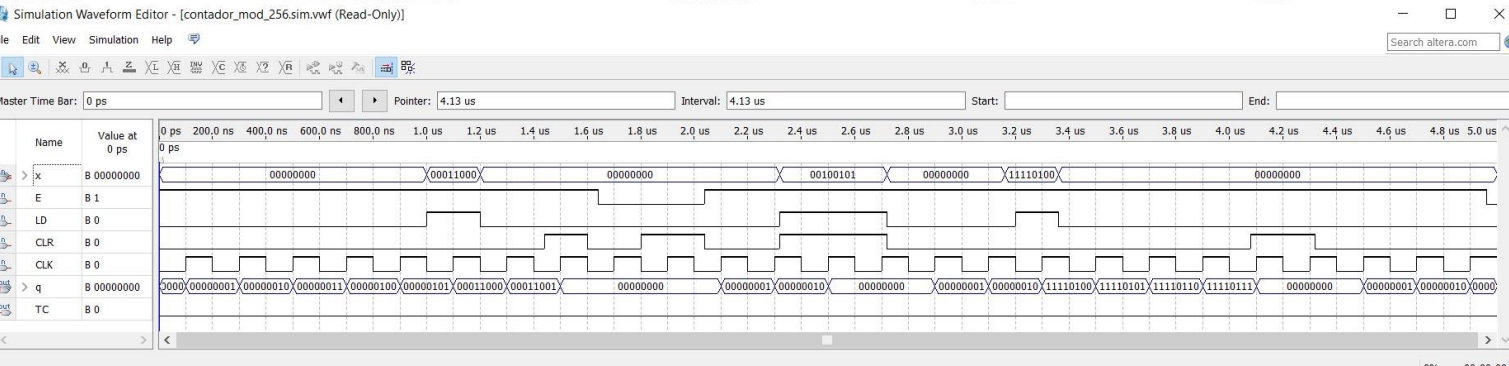


Figura 19: Simulação funcional do circuito do contador de módulo 256. Essa simulação busca demonstrar a interação entre as funcionalidades de carga e limpeza síncrona. Nota-se que a limpeza tem preferência sobre a carga, bem como ambas só acontecem se $E = 1$, assim como descrito no roteiro da atividade.