

SSC0108 – Prática em Sistemas Digitais

CPU01 – Contador e Memória

Nome	N.º USP
Thiago Cardoso	11796594
Guilherme Pacheco	11797091
Pedro Martelletto	11795641

Obs 1: Utilize este arquivo como relatório de entrega, inserindo as informações a partir da próxima página.

Obs 2: Este relatório é em grupo, deverá ser convertido em **PDF** e entregue via Moodle.

Obs 3: Não serão aceitos outros formatos.

Atividades

1. Utilizando os modelos de circuitos no LogiSIM, conecte o contador de 8 bits à memória. Utilize subcircuitos para facilitar a montagem. Apresente resultados de simulação usando o osciloscópio digital. Observe as seguintes exigências de montagem:
 - a. Para este relatório não se deve enviar arquivos do LogiSIM, apenas PDF.
 - b. O resultado da contagem deve de ser apresentado nos LEDs da máscara DE0-CV do LogiSIM, utilizando os LEDs (L0-L7).
 - c. O resultado da memória deve de ser apresentado nos displays de 7 segmentos (HEX5-HEX2).
 - d. A chave S9 deverá ser utilizada como clock do circuito.
 - e. A chave S0 deverá ser usada como Master CLR do circuito.
 - f. Utilize capturas para provar que está funcionando.
 - g. Mostre simulações de contagem com o osciloscópio digital.

Atividade Relatório :

A cada borda de subida do clock, temos o acréscimo em uma unidade no contador e o deslocamento em uma posição no ponteiro da memória ROM.

Usando a Memória 1 :

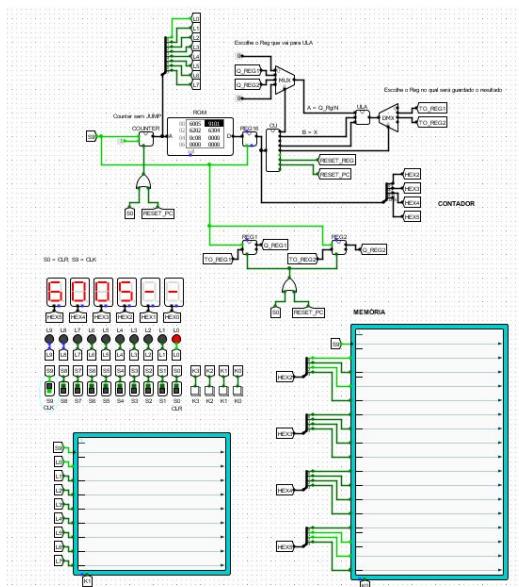


Fig1. 6005 é equivalente a 01100000000010

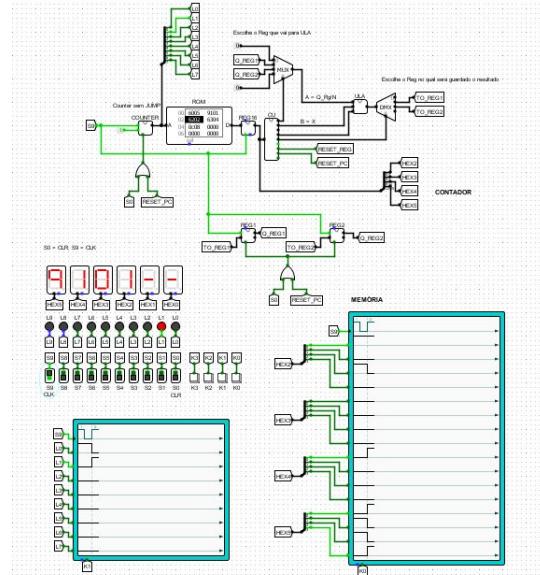


Fig2. 9101 é equivalente a 100100000000001

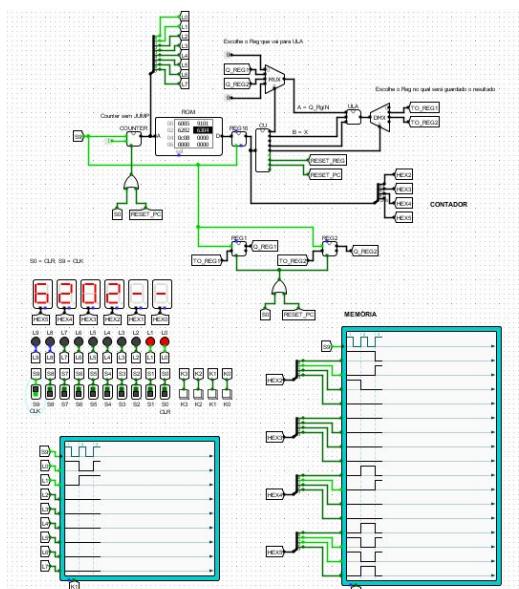


Fig3. 6202 é equivalente a 0110001000000010

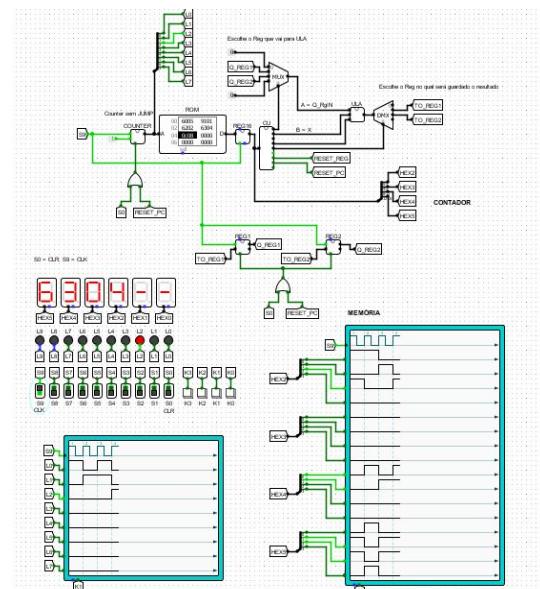


Fig4. 6304 é equivalente a 0110001100000100

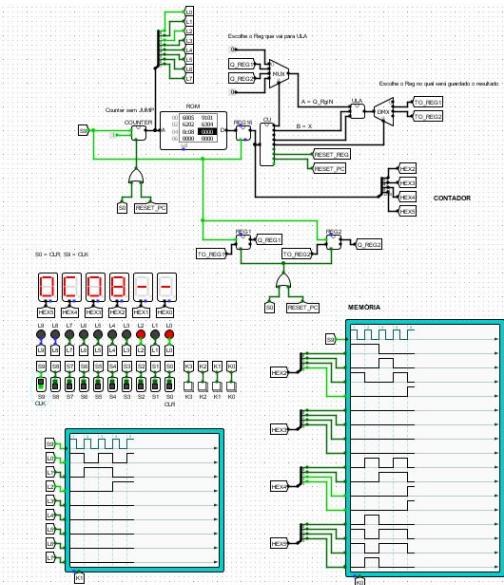


Fig5. 0c08 é equivalente a 0000110000001000

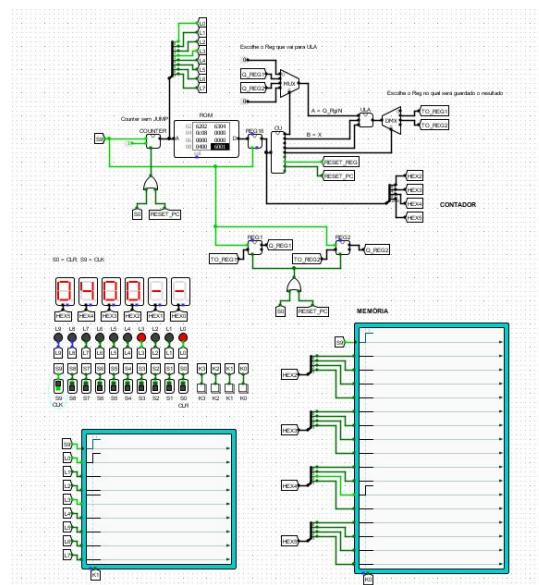


Fig6. 0400 é equivalente a 0000010000000000

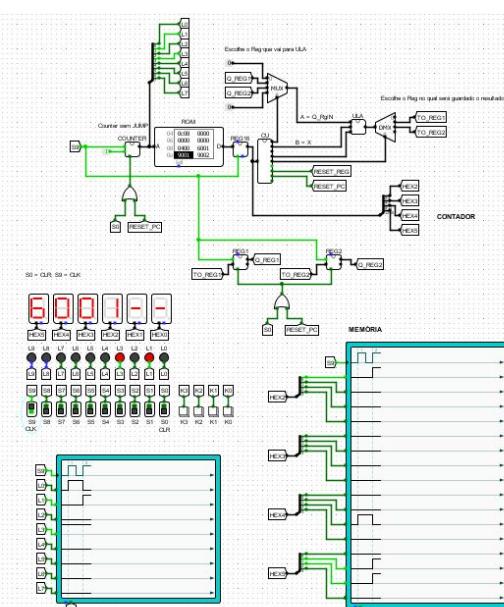


Fig7. 6001 é equivalente a 0110000000000001

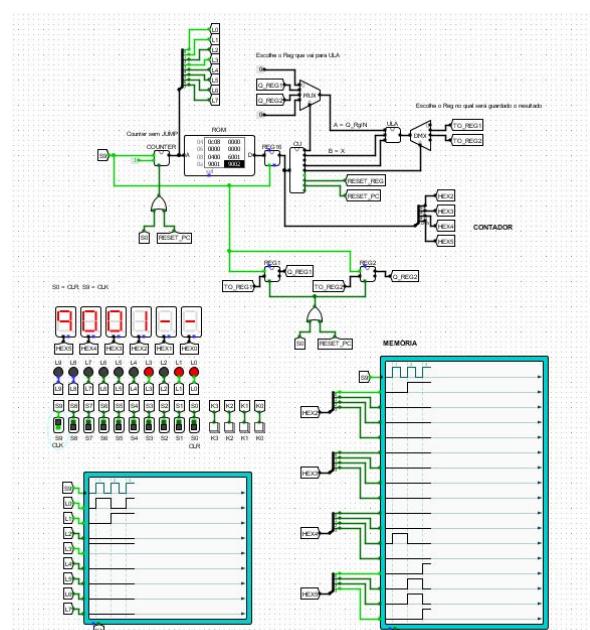


Fig8. 9001 é equivalente a 1001000000000001

Obs : Para fins de praticidade, faremos oito testes em cada memória. Os outros valores seguirão a mesma lógica mostrada.

Usando a memória 2 :

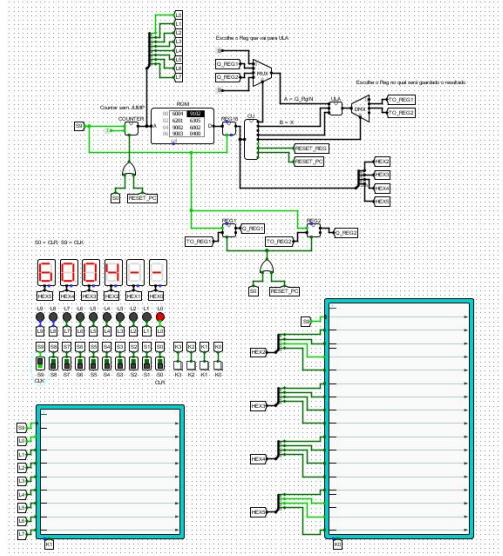


Fig9. 6004 é equivalente a 01100000000000100

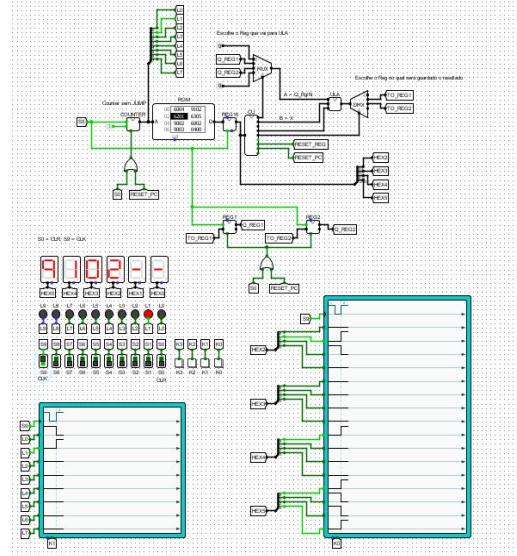


Fig10. 9102 é equivalente a 1001000100000010

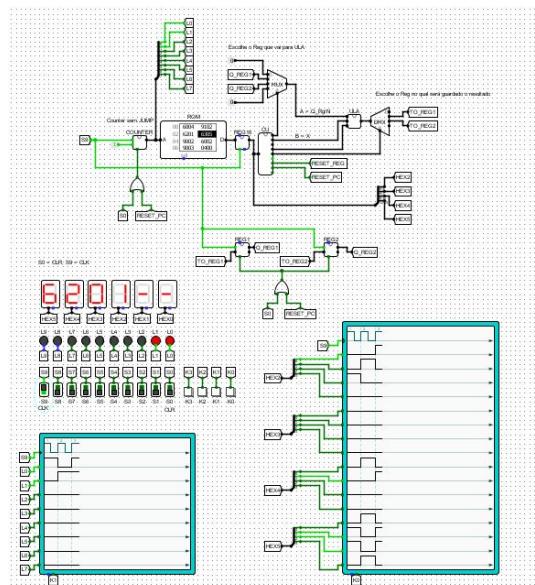


Fig11. 6201 é equivalente a 0110001000000001

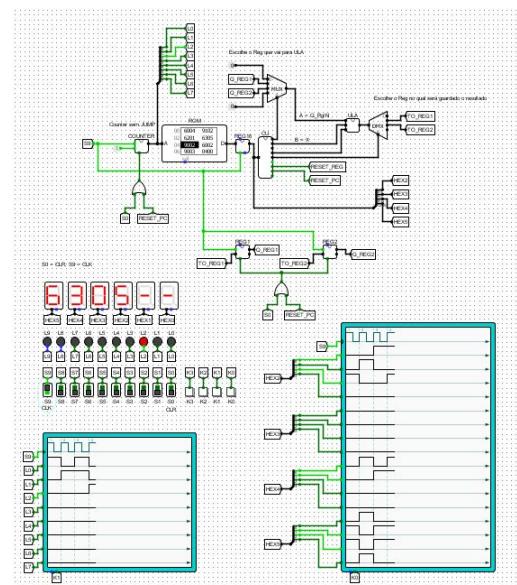


Fig12. 6305 é equivalente a 01100011000000101

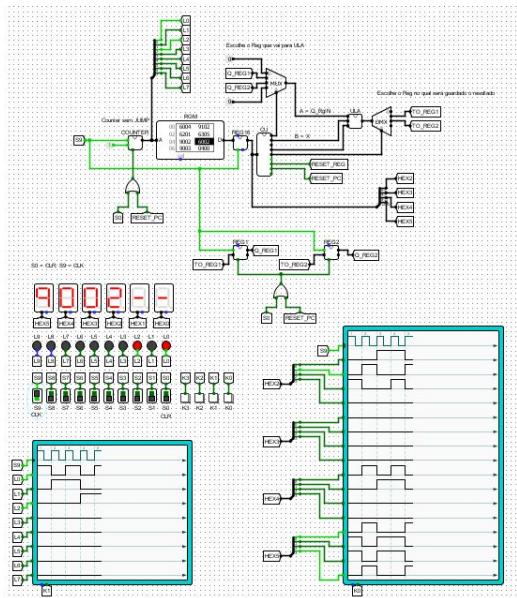


Fig13. 9002 é equivalente a 1001000000000010

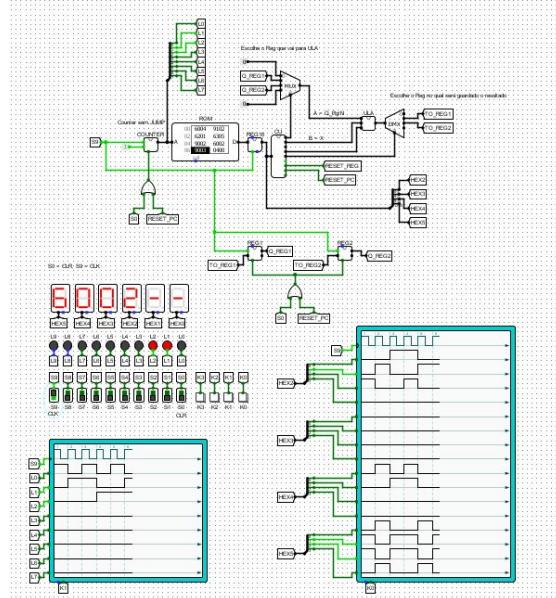


Fig14. 6002 é equivalente a 0110000000000010

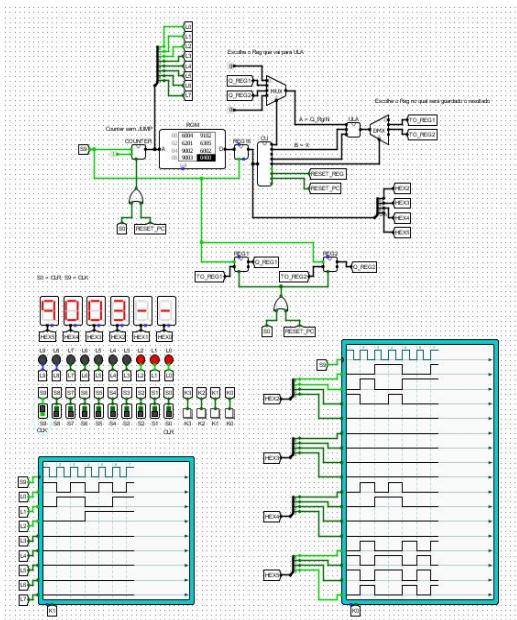


Fig15. 9003 é equivalente a 1001000000000011

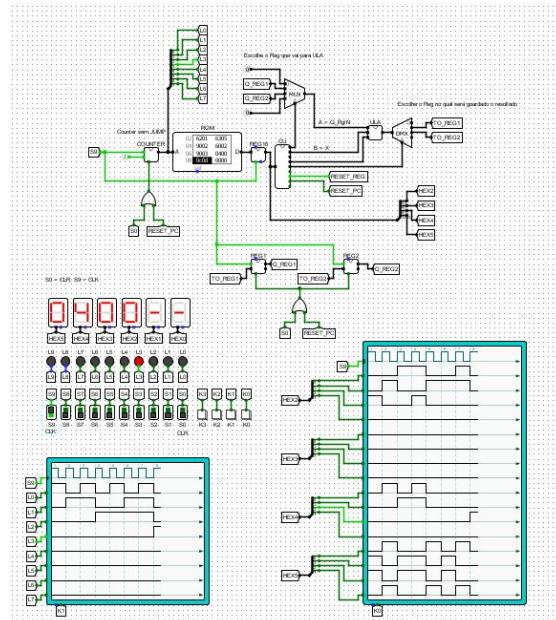
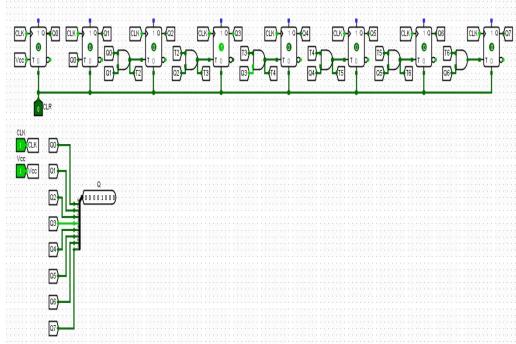


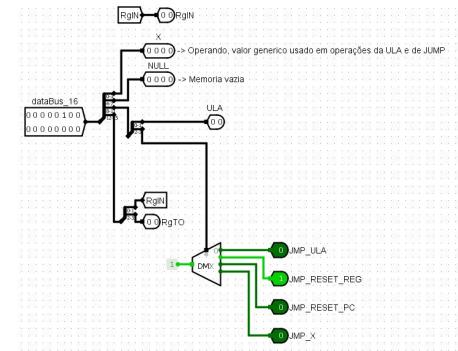
Fig16. 0400 é equivalente a 0000010000000000

Componentes usados no circuito:

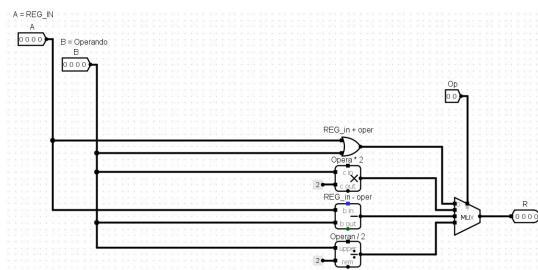
Contador :



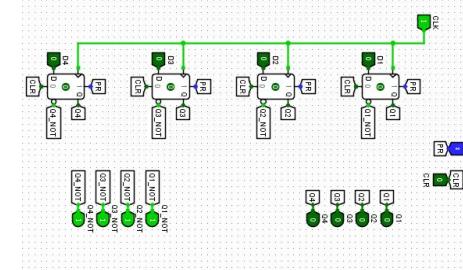
Unidade de Controle:



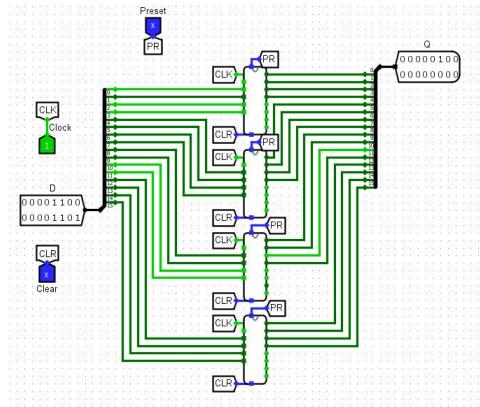
Unidade Lógica Aritmética:



Registradores de 4 Bits:



Registrador 16 Bits:



Atividades passadas em aula:

a) Atividade 1 :

Simular os Flip-Flops tipo D e tipo T no LogiSIM utilizando LEDs nas saídas. Faça em um arquivo em Branco.

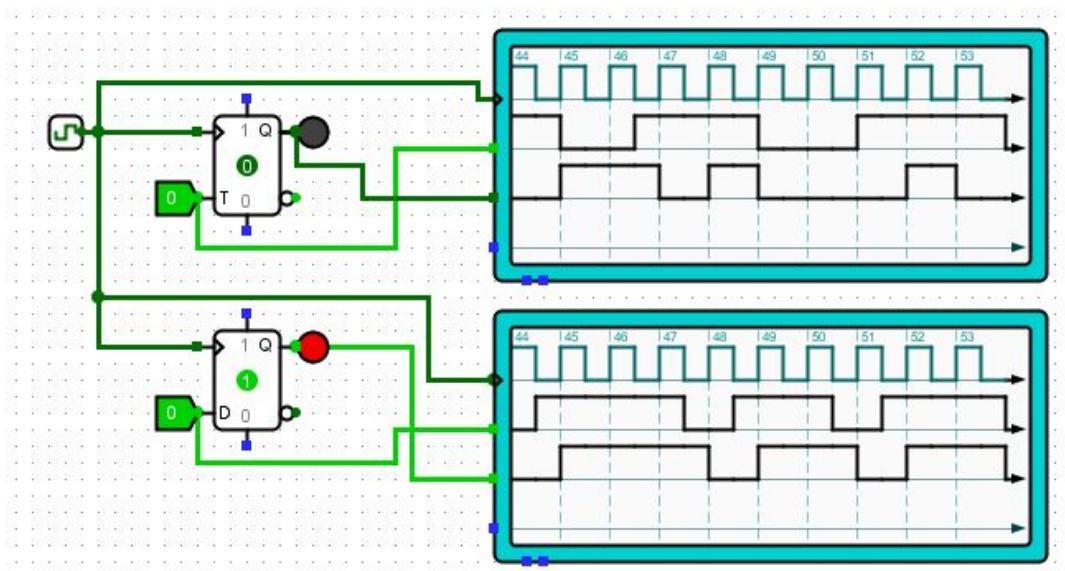


Fig1. Simulação dos flip-flops tipo D e tipo T

Criar as simulações dos Flip-Flops tipo D e tipo T utilizando o arquivo de design da placa DE0-CV fornecido (DE0-CV.circ).

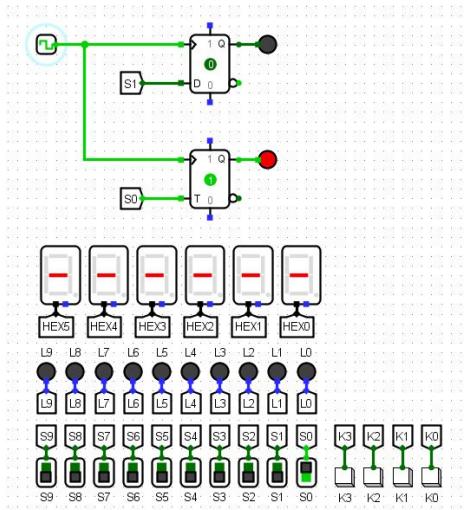


Fig2. Funcionamento dos leds mediante DE0-CV

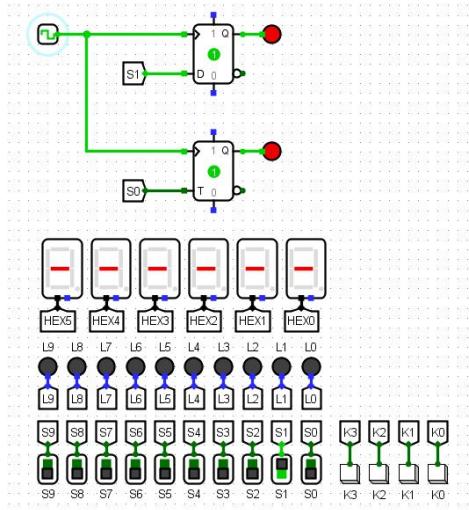


Fig3. Funcionamento dos leds mediante DE0-CV

b) Atividade 2 :

Crie um registrador de carga paralela de 4 bits e simule no LogiSIM. Utilize osciloscópio digital para confrontar os dados.

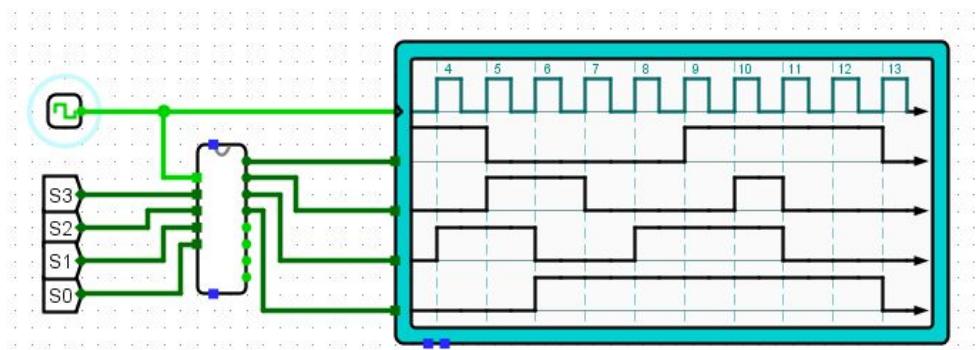


Fig4. Simulação do registrador de 4bits, sendo S_i , $i=\{0,1,2,3\}$ botões da máscara DE0-CV

Utilize a máscara da DE0-CV para simular um registrador de carga paralela colocando seus dados em um display de 7 segmentos.

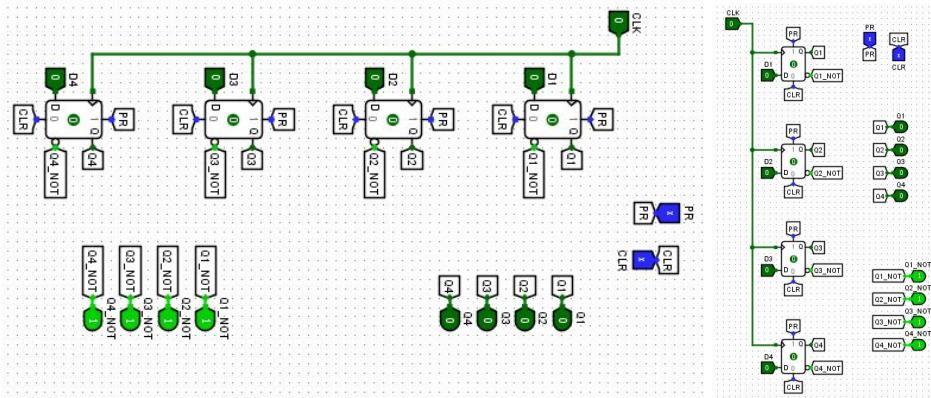


Fig5. Registrador usado nas simulações

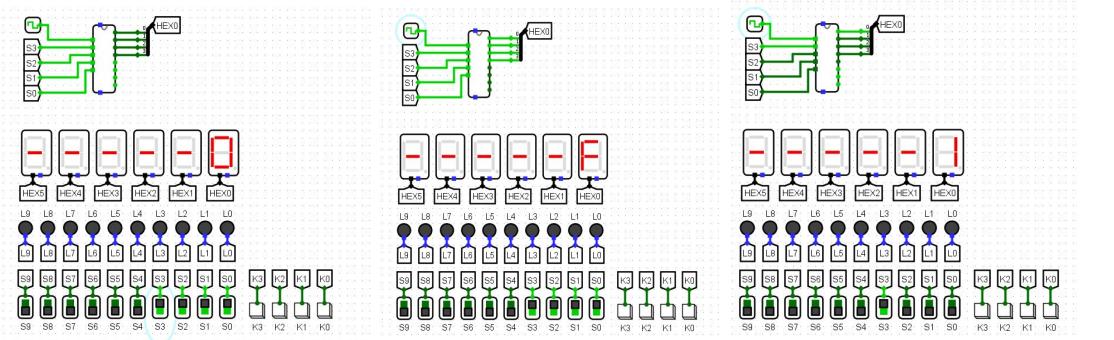


Fig6. Representação no display de 7 segmentos, o nº armazenado é mostrado a cada borda de subida

c) Atividade 3 :

Crie um Contador Síncrono de 8 bits (2^8 estados).

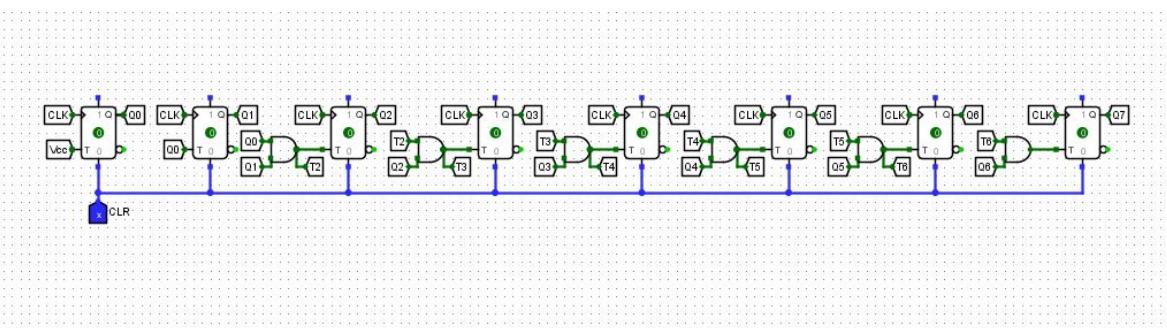


Fig7. Circuito lógico do contador síncrono de 256 estados

Utilize a máscara da DE0-CV para poder testar seu funcionamento.

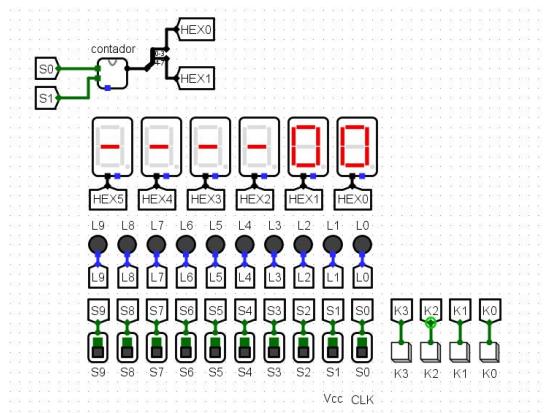


Fig8. Representação do circuito com a máscara $S0 = S1 = 0$

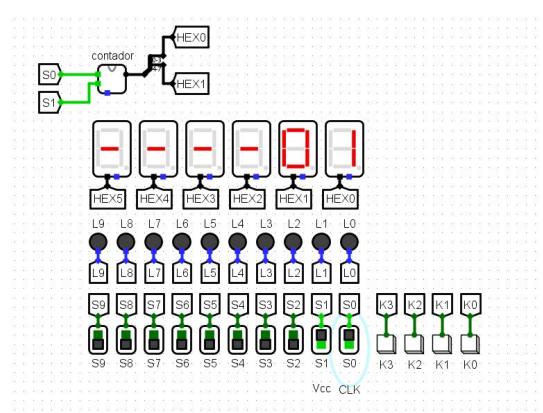


Fig9. Ao ligarmos ambos os botões S0 e S1, a contagem inicia

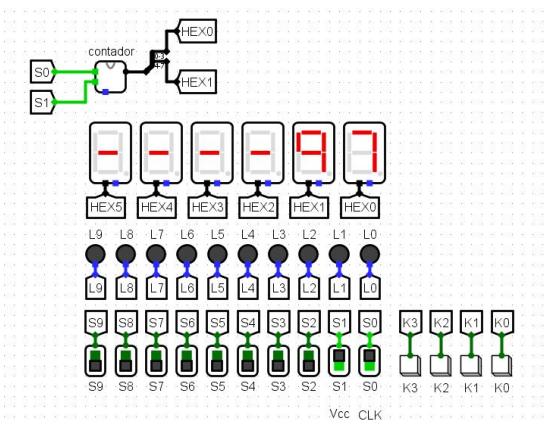


Fig10. Mostra que o contador não se limita a 15 números

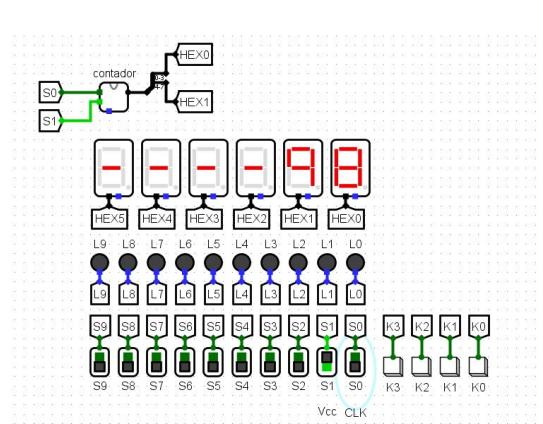


Fig11. Na borda de subida o número é acrescido em um

d) Atividade 4 :

Crie uma ROM com 8 Bits de endereços e 16 Bits de dados.

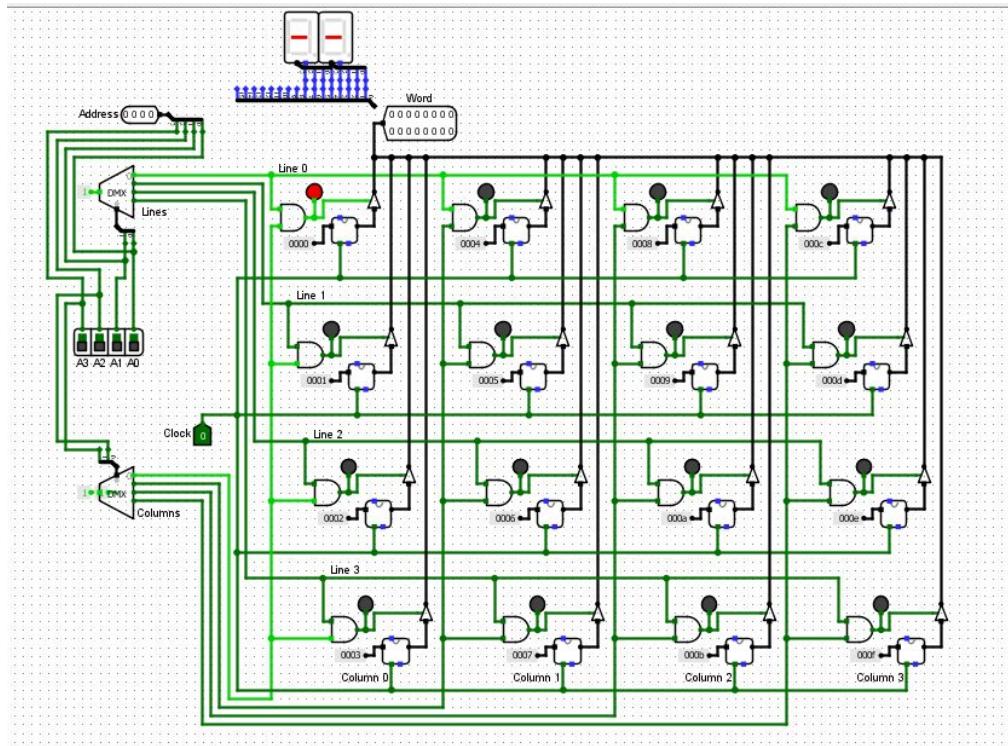


Fig12. Read Only Memory (ROM) 16x16

Conekte o barramento de dados da memória nos displays de 7 segmentos da Máscara DE0-CV.

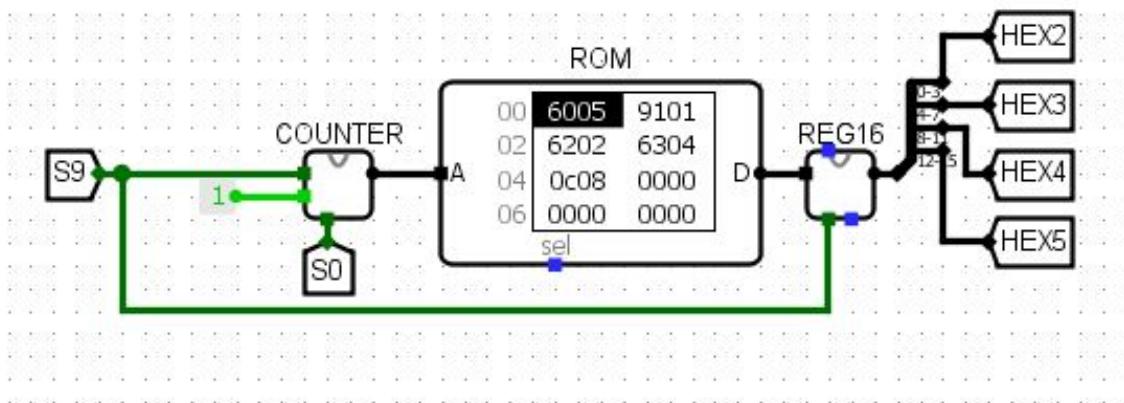


Fig13. Contador ligado a memória ROM e barramento de dados conectado ao display

Preencha o conteúdo da memória de acordo com a Mem 1 e Mem 2 para testar.

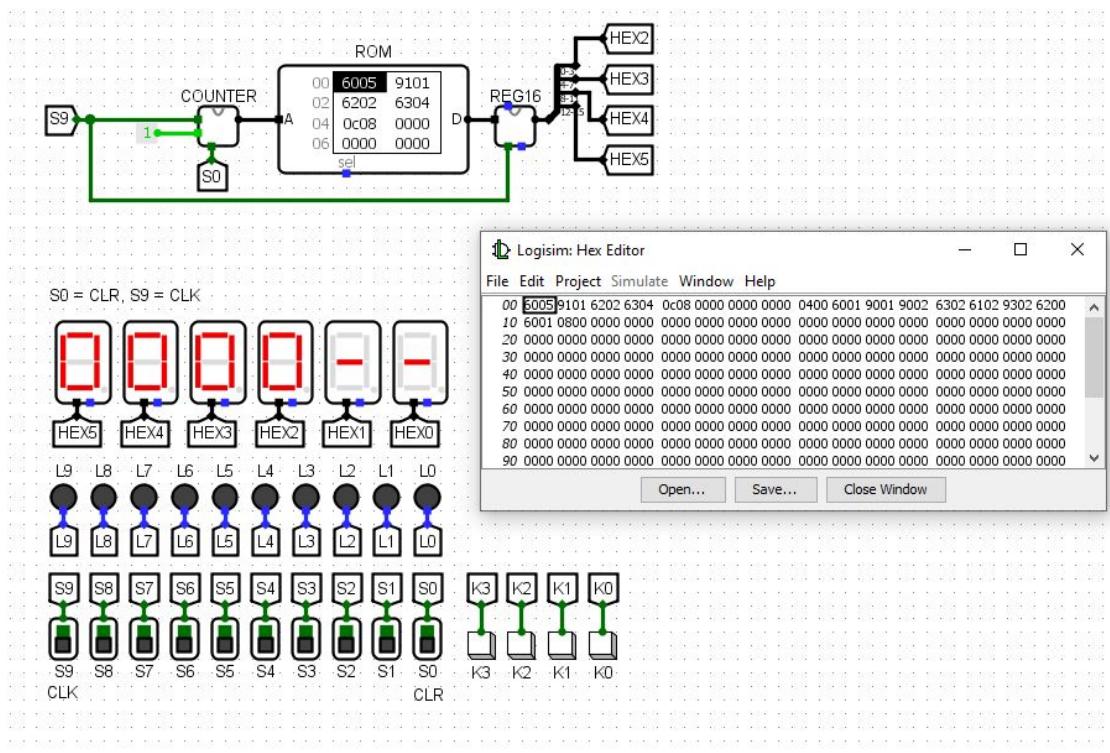


Fig14. Memória ROM preenchida com as informações passadas na Memória 1 para fins de teste

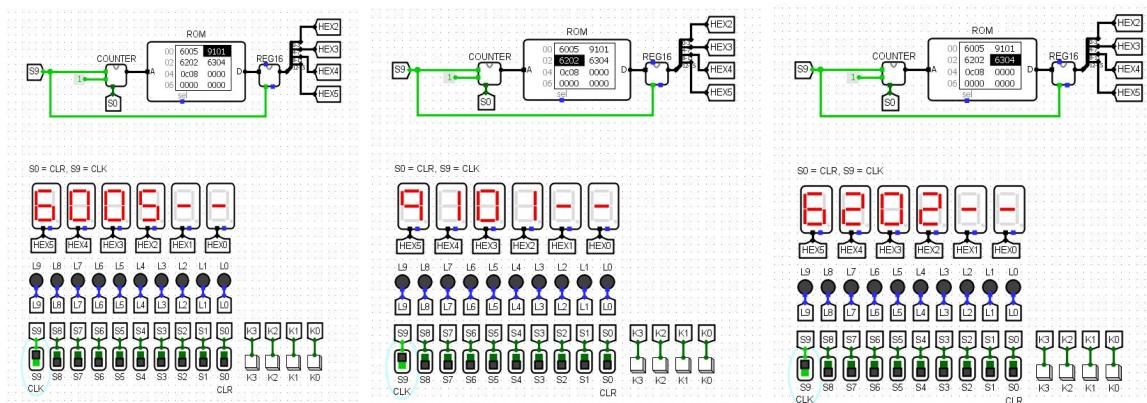


Fig15. Ao acionarmos o clock, imprimimos no display o número contido na ROM, um por um.

OBS: Optamos por não mostrar os testes na memória dois, dado que o procedimento é o mesmo realizado na memória de número um.

f) Atividade 6 :

Faça as adaptações necessárias para que os resultados da ULA sejam armazenados no registrador.

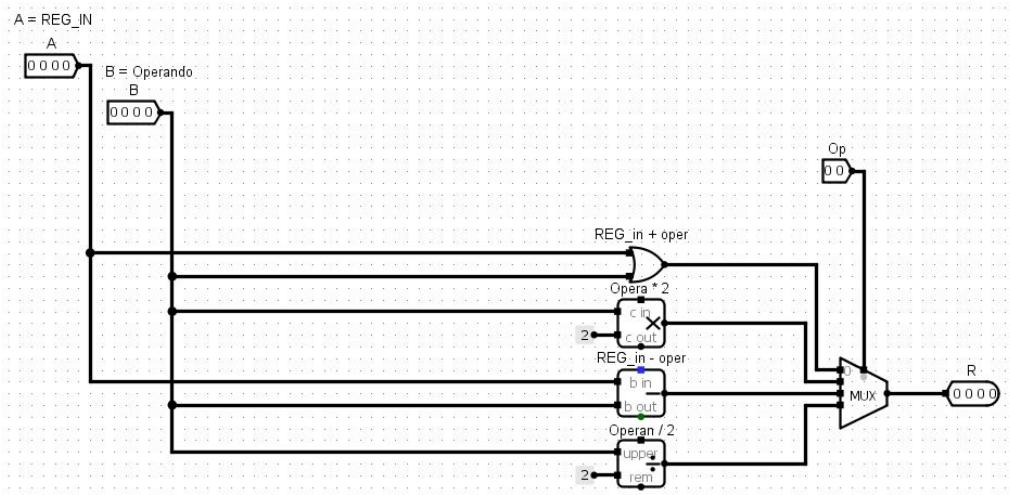


Fig16. Unidade Lógica Aritmética (ULA)

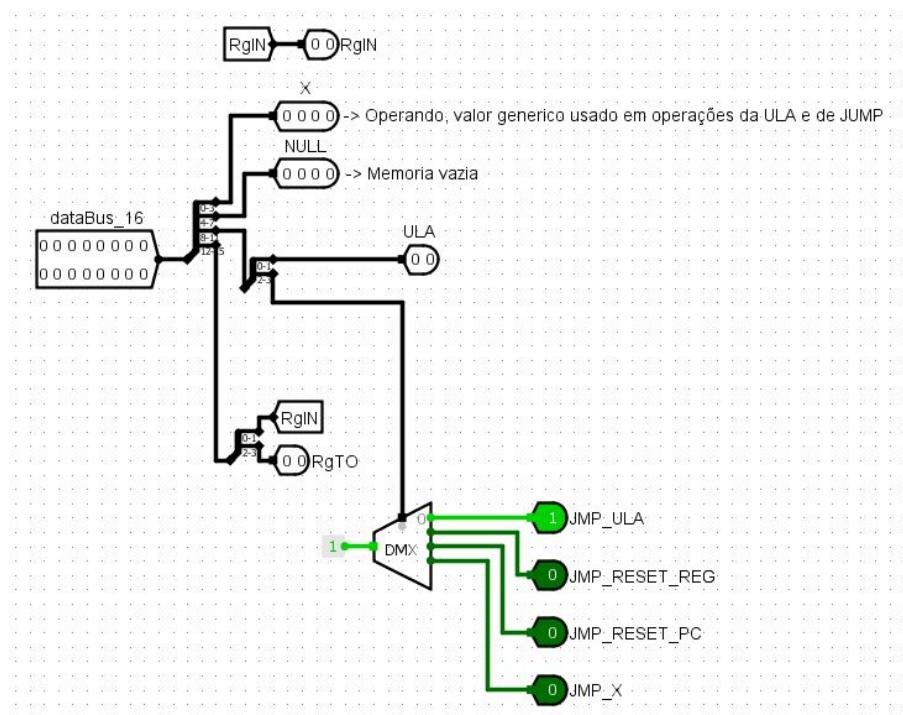


Fig17. Unidade de Controle (CU)

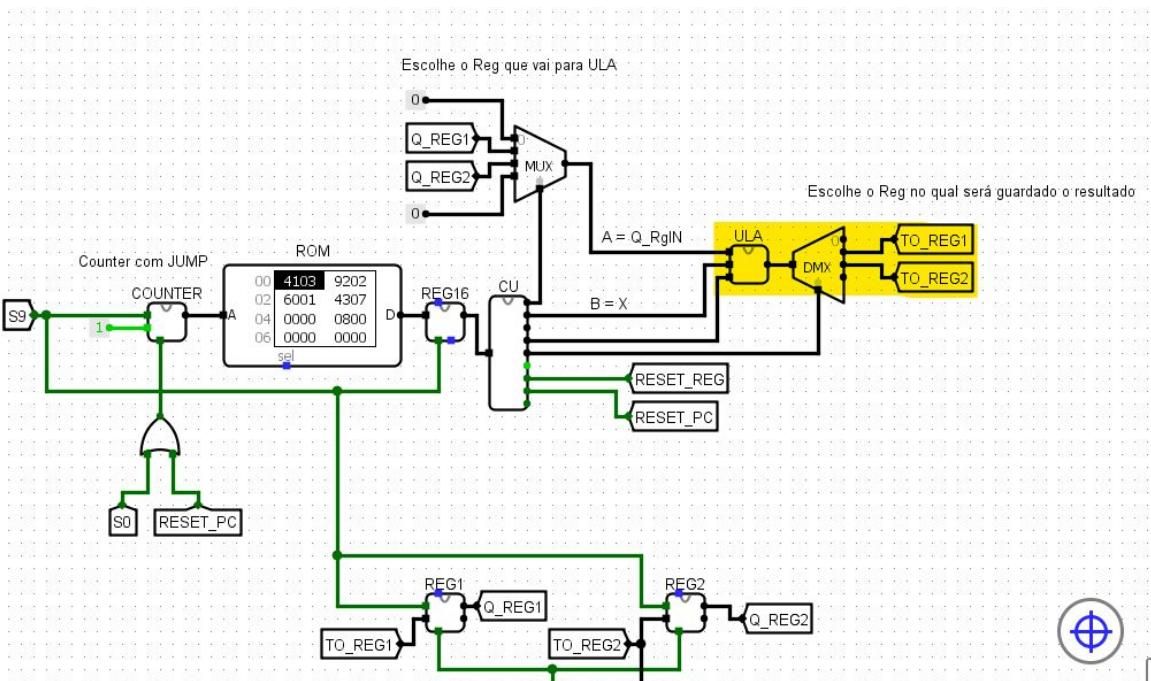


Fig18. O resultado obtido na ULA é movido para os registradores, por um demultiplexador e pela CU.

Realize simulações utilizando o osciloscópio na saída da ULA, entrada do registrador e saída do registrador.

Como testes, utilizaremos as seguintes instruções:

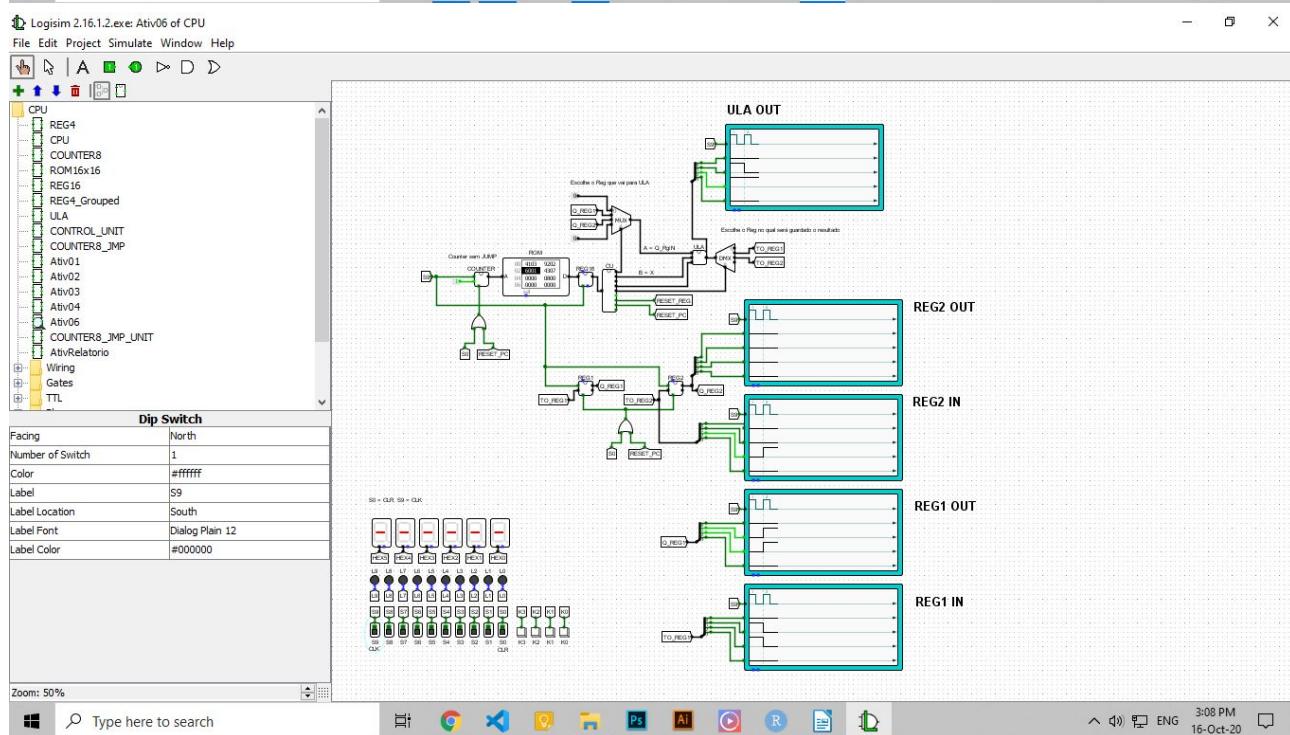
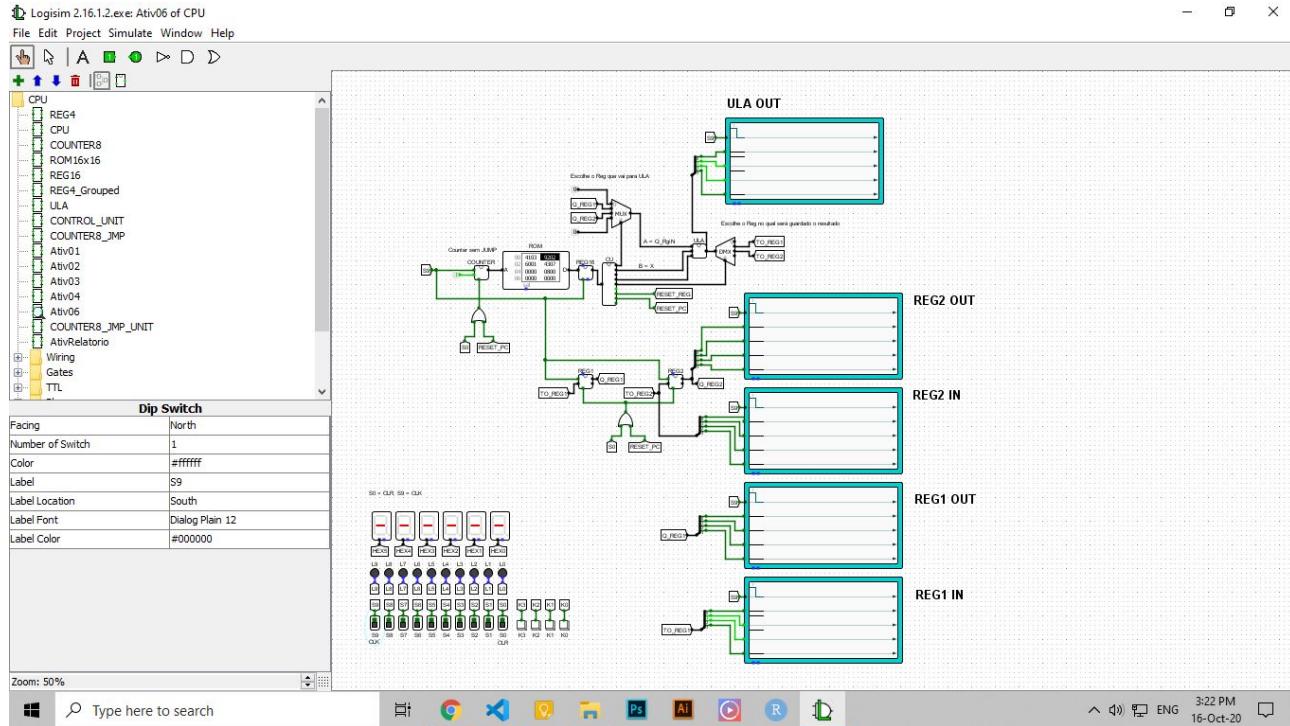
```
# REG1 <- 3 * 2 = 6
0100000100000011 = 0x4103 (HEX)
# REG2 <- REG1 - 2 = 4
1001001000000010 = 0x9202 (HEX)
# REG1 <- REG2 + 1 = 5
0110000000000001 = 0x6001 (HEX)
# REG1 <- 7 / 2 = 3
0100001100000111 = 0x4307 (HEX)
```

Ou seja, esperamos que os resultados dos registradores sejam:

Nº de instruções executadas:	REG1	REG2
1	6	-
2	6	4
3	5	4
4	3	4

É precisamente isso que observamos:

1^a e 2^a Iteração: Observe que REG1 = 6 = 0110 e, depois, REG2 = 4 = 0100



3^a e 4^a Iteração: Observe que REG1 = 5 = 0101 e, depois, REG1 = 3 = 0011

