

[illegible]

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	0
1	1	1	0	0
1	1	1	1	0

0	0	0	0	0	0
0	1	1	0	0	0
0	0	1	0	0	0
0	0	0	0	1	1
0	0	0	1	1	0
1	0	0	0	0	1
0	0	0	1	1	0
0	0	1	1	1	0

1.4. Minimizações:

$z_6:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	1	1	0	0
x'_3x_2	0	0	1	0
x_3x_2	1	0	0	0
$x_3x'_2$	0	0	0	0

$$z_6 = x'_3x'_2x'_1 + x_3x_2x'_1x'_0 + x'_3x_2x_1x_0$$

$z_5:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	0	1	1	1
x'_3x_2	0	0	1	0
x_3x_2	0	1	0	0
$x_3x'_2$	0	0	0	0

$$z_5 = x_3x_2x'_1x'_0 + x'_3x'_2x_0 + x'_3x_1x_0 + x'_3x'_2x_1$$

$z_4:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	0	1	1	0
x'_3x_2	1	1	1	0
x_3x_2	0	0	0	0
$x_3x'_2$	0	1	0	0

$$z_4 = x'_3x_2x'_1 + x'_2x'_1x_0 + x'_3x_0$$

$z_3:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	0	1	0	0
x'_3x_2	1	0	1	0
x_3x_2	0	0	1	0
$x_3x'_2$	0	1	0	1

$$z_3 = x'_3x_2x'_1x'_0 + x_3x'_2x_1x'_0 + x'_2x'_1x_0 + x_2x_1x_0$$

$z_2:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	0	0	0	1
x'_3x_2	0	0	0	0
x_3x_2	1	0	1	1
$x_3x'_2$	0	0	0	0

$$z_2 = x'_3x'_2x_1x'_0 + x_3x_2x_1 + x_3x_2x'_0$$

$z_1:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	0	0	0	0
x'_3x_2	0	1	0	1
x_3x_2	1	0	1	1
$x_3x'_2$	0	0	1	0

$$z_1 = x'_3x_2x'_1x'_0 + x_3x_2x'_0 + x_3x_1x_0 + x_2x_1x'_0$$

$z_0:$	$x'_1x'_0$	x'_1x_0	x_1x_0	$x_1x'_0$
$x'_3x'_2$	0	1	0	0
x'_3x_2	1	0	0	0
x_3x_2	0	1	0	0
$x_3x'_2$	0	0	1	0

$$z_0 = x'_3x_2x'_1x'_0 + x'_3x'_2x'_1x_0 + x_3x_2x'_1x_0 + x_3x'_2x_1x_0$$

1.5. Esquemático do circuito:

Seguindo as equações de minimização mostradas acima, o circuito do decodificador para display de 7 segmentos foi montado e seu diagrama esquemático está disposto na figura 1. É válido ressaltar que, com o intuito de deixar o circuito final o mais semelhante possível ao apresentado no roteiro da atividade, a saída z teve seu nome alterado para *seg*.

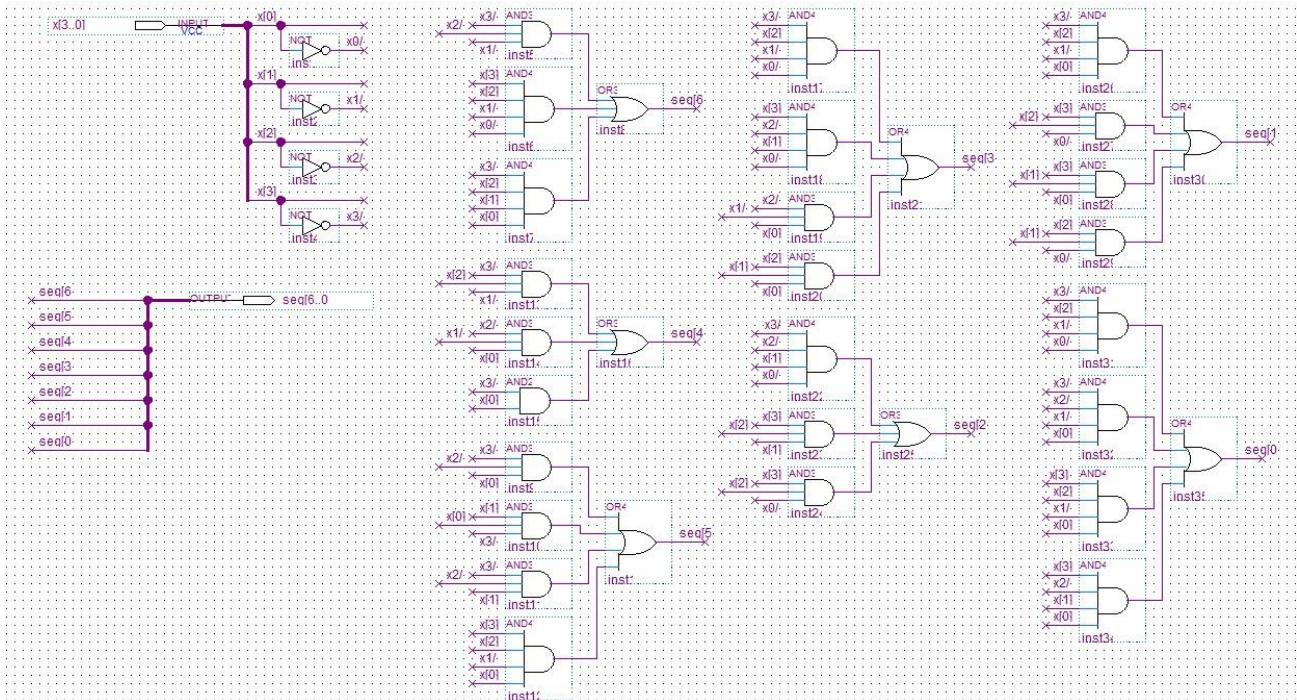


Figura 1: Diagrama esquemático do circuito do decodificador para display de 7 segmentos. Esse circuito será utilizado no circuito final dessa atividade, por isso ele foi encapsulado em um módulo e nomeado como “decodificador-7seg”.

1.6. Simulações:

Como todos os circuitos desse projeto estão submetidos a clocks manuais, somente simulações funcionais foram efetuadas. O circuito do decodificador passou por uma simulação na qual todas as combinações possíveis de entrada foram consideradas. O resultado dela está disposto na figura 2.

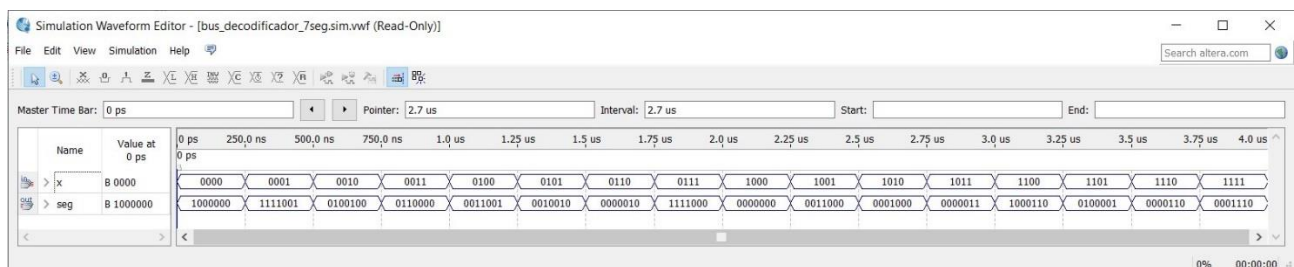


Figura 2: Simulação funcional do circuito do decodificador para display de sete segmentos. Nela é possível notar que o circuito funciona de acordo com o previsto pela tabela verdade apresentada anteriormente, ou seja, a saída do circuito é a representação no display da conversão para hexadecimal da entrada binária.

2. Projeto 2: Módulo “bus_tristate”

2.1. Escopo:

Projeto de um circuito combinacional que implemente um barramento de 4 bit, controlado por meio de buffers tri-state ligado a um único sinal de controle. Este barramento deve apresentar o valor de sua entrada caso o sinal de controle esteja ativo, ficando em estado de alta impedância, caso contrário.

2.2. Especificação de alto-nível:

Entrada:

$\underline{in} = (in_3, in_2, in_1, in_0)$, com $in_k \in \{0,1\}$ e $k = 0, \dots, 3$;

$e \in \{0,1\}$ (entrada de controle).

Saída:

$\underline{out} = (out_3, out_2, out_1, out_0)$, com $out_k \in \{0,1\}$ e $k = 0, \dots, 3$.

Função:

$$out_k = \begin{cases} in_k & , \text{se } e = 1 \\ Z & , \text{caso contrário} \end{cases} \quad , \text{ com } k = 0, \dots, 3 \text{ e } Z \text{ sendo o estado de alta impedância.}$$

2.3. Especificação binária:

Como o circuito é muito simples e o estado de alta impedância não apresenta representação binária adequada, a especificação binária acaba sendo a mesma que a especificação de alto nível.

2.4. Minimizações:

A única forma de implementar o circuito que descreve a função apresentada na especificação é por meio do uso de buffers tri-state. Como cada bit do a barramento é independente dos demais, basta ligar cada um deles a um buffer desse tipo e associar o sinal de controle dele ao sinal de enable do barramento. Dessa forma, nenhuma minimização será necessária.

2.5. Esquemático do circuito:

Seguindo o apresentado acima, o circuito do módulo foi implementado e seu esquemático está disposto na figura 3.

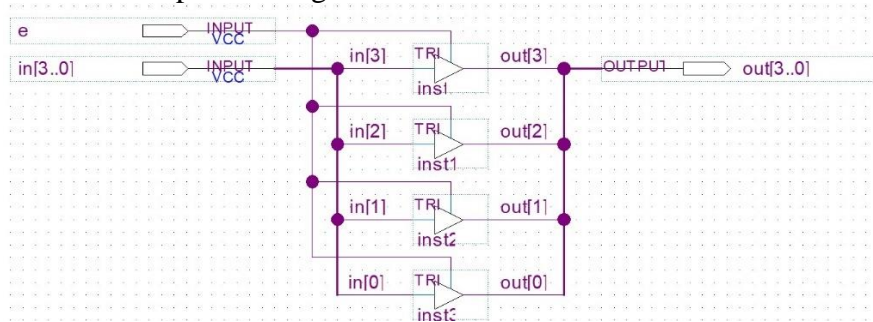


Figura 3: Diagrama esquemático do módulo “bus_tristate”. Este módulo também será utilizado no circuito final desta atividade.

2.6. Simulações:

O circuito deste módulo passou por uma simulação funcional simples, a qual buscou comprovar a capacidade dele de descrever o comportamento de três estados desejado. O resultado dessa simulação está disposto na figura 4.

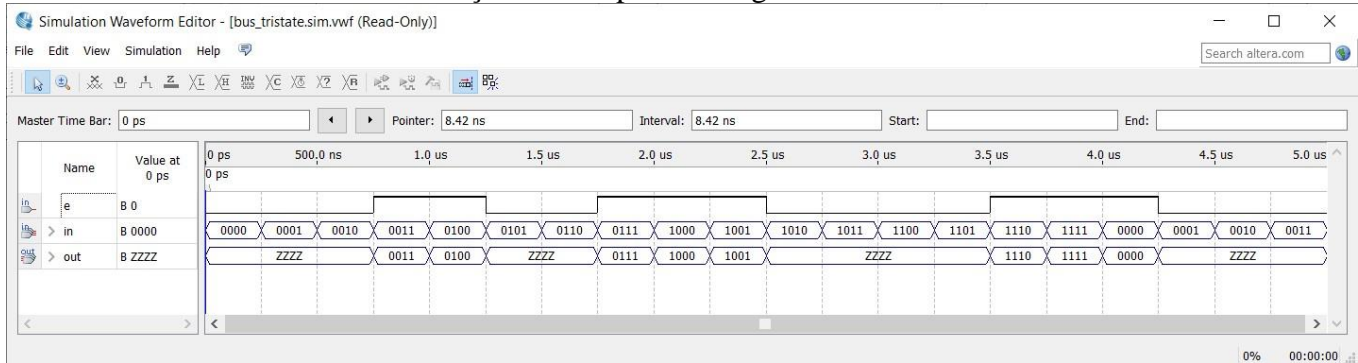


Figura 4: Simulação funcional do módulo “bus_tristate”. Nela é possível notar que o barramento disponibiliza os sinais de sua entrada somente se o valor da entrada de controle é 1, ficando em alta impedância (Z) caso ele seja igual a zero. Tal comportamento é exatamente o descrito pela especificação do circuito

3. Projeto 1: ULA Estendida

3.1. Escopo:

Projeto de um circuito sequencial capaz de implementar uma unidade lógico-aritmética com capacidade tanto de armazenar seus operandos e resultados em registradores, quanto de exibi-los em displays de 7 segmentos. Esse circuito utilizará não só dos dois módulos desenvolvidos nessa atividade, mas também dos módulos “ULA” e “registrador_4_bits” construídos em atividades anteriores.

3.2. Especificação de alto-nível:

Entradas:

$ROW1 \in \{0,1\}$ (entrada que controla o barramento de dados);

$\underline{d} = (d_3, d_2, d_1, d_0)$, com $d_i \in \{0,1\}$ e $i = 0, \dots, 3$ (entrada que insere valores no barramento de dados);

$LD_A, CLR_A, G_A \in \{0,1\}$ (entradas de controle para o registrador A);

$LD_B, CLR_B, G_B \in \{0,1\}$ (entradas de controle para o registrador B);

$LD_{AC}, CLR_{AC}, G_{AC} \in \{0,1\}$ (entradas de controle para o registrador acumulador);

$LD_{FL}, CLR_{FL}, G_{FL} \in \{0,1\}$ (entradas de controle para o registrador de flags);

Saídas:

$\underline{ACM} = (ACM_3, ACM_2, ACM_1, ACM_0)$, com $ACM_i \in \{0,1\}$ e $i = 0, \dots, 3$ (valor presente no registrador acumulador);

$\underline{FLAGS} = (FLAGS_3, FLAGS_2, FLAGS_1, FLAGS_0)$, com $FLAGS_i \in \{0,1\}$ e $i = 0, \dots, 3$ (valor presente no registrador de flags)

$\underline{DISP0} = (DISP0_6, DISP0_5, DISP0_4, DISP0_3, DISP0_2, DISP0_1, DISP0_0)$, com $DISP0_i \in \{0,1\}$ e $i = 0, \dots, 6$ (saída para apresentação no display de 7 segmentos do valor presente no barramento em hexadecimal);

$\underline{DISP1} = (DISP1_6, DISP1_5, DISP1_4, DISP1_3, DISP1_2, DISP1_1, DISP1_0)$, com $DISP1_i \in \{0,1\}$ e $i = 0, \dots, 6$ (saída para apresentação no display de 7 segmentos do valor presente no registrador acumulador em hexadecimal);

$\underline{DISP2} = (DISP2_6, DISP2_5, DISP2_4, DISP2_3, DISP2_2, DISP2_1, DISP2_0)$, com $DISP2_i \in \{0,1\}$ e $i = 0, \dots, 6$ (saída para apresentação no display de 7 segmentos do valor presente no registrador B em hexadecimal);

$\underline{DISP3} = (DISP3_6, DISP3_5, DISP3_4, DISP3_3, DISP3_2, DISP3_1, DISP3_0)$, com $DISP3_i \in \{0,1\}$ e $i = 0, \dots, 6$ (saída para apresentação no display de 7 segmentos do valor presente no registrador A em hexadecimal);

Função:

Como o funcionamento do circuito é complexo, ele será descrito textualmente, ao invés de usando equações, com intuito de manter a clareza e facilitar o entendimento. Dito isto, temos que:

- O sinal $R0W1$ deve controlar se a ULA e os registradores A e B irão receber do barramento os dados da entrada \underline{d} ou o do acumulador AC.
- A entrada \underline{d} atuará tanto como fornecedora de valores para os registradores A e B, quanto para seletora das operações executadas pela ULA (caso permitida, pela entrada $R0W1$, de escrever no barramento).
- As entradas do tipo LD devem controlar o carregamento de dados para o seu registrador correspondente. O valor armazenado do registrador só deve se alterar caso haja um pulso de clock e LD for igual a 1.
- As entradas do tipo G servirão como clocks de cada registrador. O valor armazenado no registrador correspondente só será alterado para o valor que ele recebe em sua entrada caso ocorra um pulso na entrada G e sua entrada LD seja igual a 1.
- As entradas do tipo CLR funcionam como sinais de limpeza síncrona para cada registrador, ou seja, o valor presente no registrador será zerado caso sua entrada CLR seja 1 e ocorra um pulso de clock.
- A ULA interna do circuito receberá como operandos os valores dos registradores A (ligado a entrada \underline{x}) e B (ligado a entrada \underline{y}) e terá sua operação (entrada \underline{f}) selecionada pelo valor disponível no barramento. O resultado da operação ficará disponível (por meio da saída \underline{s}) para ser armazenado no registrador acumulador, enquanto as bandeiras de status (saídas C, V, N e Z da ULA) poderão ser guardadas no registrador de flags. A ULA terá as seguintes operações disponíveis:

Código ($f_3f_2f_1f_0$)	Nome	Operação	Descrição
000x	ADD	$\underline{s} = \underline{x} + \underline{y}$	Adição de x com y
001x	ADDC	$\underline{s} = \underline{x} + \underline{y} + c_i$	Adição de x com y com carry
010x	SUB	$\underline{s} = \underline{x} - \underline{y}$	Subtração y de x
0110	INC	$\underline{s} = \underline{x} + 1$	Incremento unitário de x
0111	DEC	$\underline{s} = \underline{x} - 1$	Decremento unitário de x
1000	NEG	$\underline{s} = -\underline{x}$	Negação aritmética de x em complemento de 2

1001	CMPL	$\underline{s} = \sim \underline{x}$	Complemento bit-a-bit de x
------	------	--------------------------------------	----------------------------

- A saída ACM será ligada na saída do registrador acumulador para poder apresentar diretamente, em binário, o valor armazenado no acumulador da ULA.
- A saída FLAGS será ligada na saída do registrador de flags para poder apresentar seu valor.
- As saídas do tipo DISP serão utilizadas para apresentar os valores dos registradores A, B e AC, bem como o valor disponível no barramento, em displays de 7 segmentos. Para isso, estes sinais primeiro são inseridos em decodificadores, os quais geram sua representação adequada, para depois serem enviados para sua saída respectiva.

3.3. Especificação binária e minimizações:

As entradas e saídas apresentadas na especificação de alto-nível já estão codificadas em binário, portanto não é necessário redefini-las na especificação binária. Quanto à forma binária da função do circuito ULA estendida e as minimizações decorrentes dela, como a descrição de seu funcionamento já é consideravelmente complexa mesmo em linguagem de alto-nível e tal circuito será quase totalmente implementado a partir de módulos já previamente especificados e minimizados, torna-se mais intuitivo implementar o circuito a partir do funcionamento descrito anteriormente, tomando como base o funcionamento binário de cada módulo, do que tentar aglutinar todos esses comportamentos em uma só descrição.

3.4. Esquemático do circuito:

O circuito da ULA estendida foi implementado seguindo as especificações descritas anteriormente, bem como o modelo disponível no roteiro da atividade. Seu diagrama esquemático está disposto na figura 5.

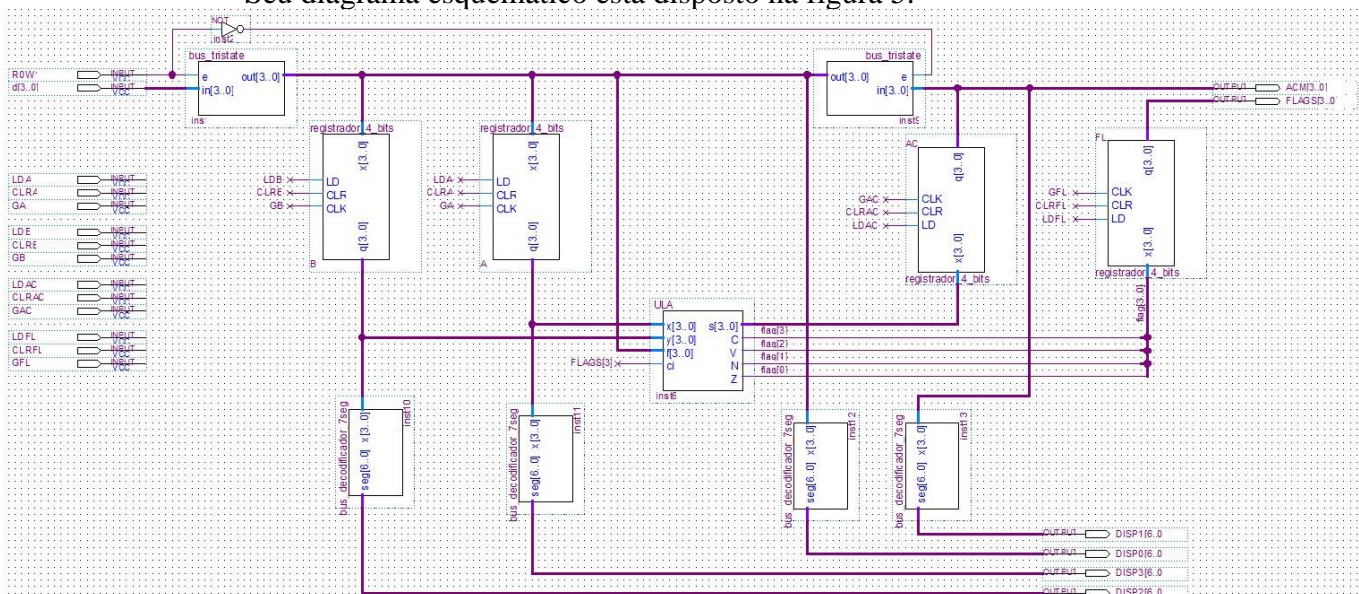


Figura 5: Diagrama esquemático do circuito ULA estendida. Esse circuito implementa a ULA descrita na especificação do projeto, usando, para tal, dos módulos “registrador_4_bits”, “ULA”, “decodificador_7seg” e “bus_tristate” implementados nesta e em outras atividades. Esse circuito também será encapsulado em um módulo, chamado de “bus_ula”, para ser testado na FPGA.

3.5. Simulações:

O circuito da ULA estendida passou por simulações funcionais que testaram cada uma das operações disponíveis com diferentes operandos. Essas simulações estão dispostas nas figuras 6 e 7.



Figura 6: Simulação funcional do circuito ULA estendida. Nela são testadas as operações ADD (intervalo entre 0 μ s e 3 μ s), com os operandos A = 10 e B = 9; ADDC (intervalo entre 3 μ s e 6 μ s), com os operandos A = 3 (injetado a partir do acumulador), B = 5 e com a flag de carry da operação anterior; SUB (intervalo entre 6 μ s e 8 μ s), com os operandos A=4 e B=5 (mantido); e INC (intervalos entre 8 μ s e 10 μ s), com o operando A = -1 (injetado a partir do acumulador). Observando atentamente cada valor de saída, é possível notar que o comportamento obtido é o mesmo do que o desejado na especificação do circuito.

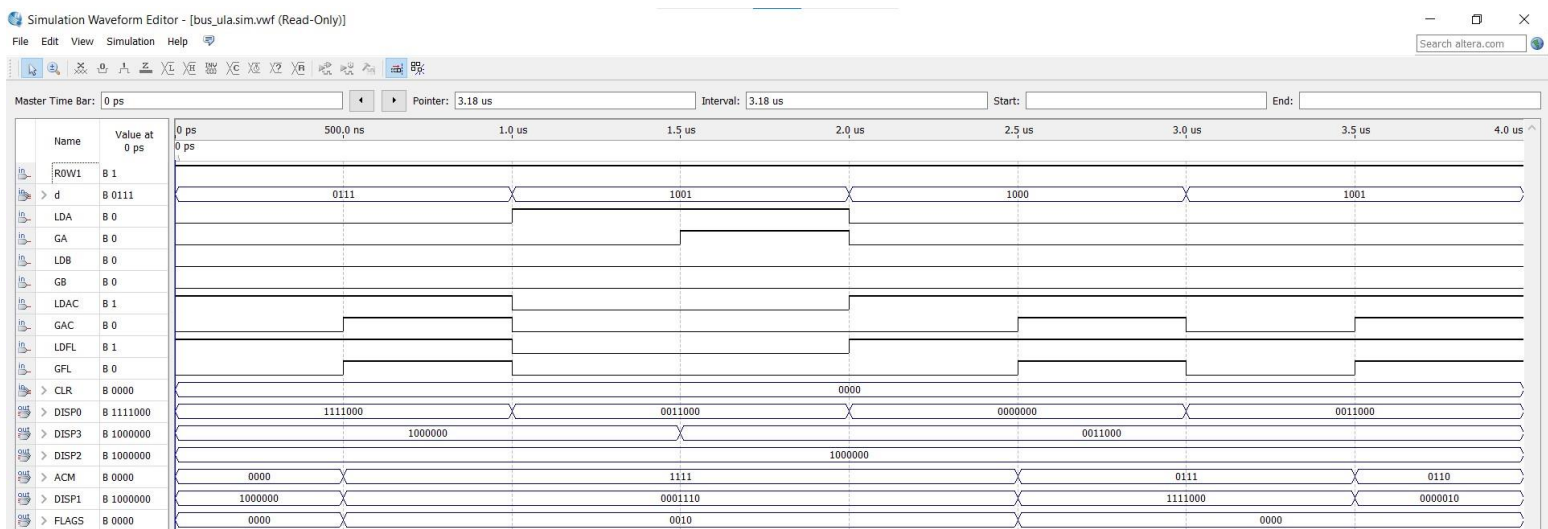


Figura 7: Simulação funcional do circuito ULA estendida. Nela são testadas as operações DEC (intervalo entre 0 μ s a 1 μ s), com o operando A = 0; NEG (intervalo entre 1 μ s e 3 μ s), com o operando A = 9; e CMPL (intervalo entre 3 μ s e 4 μ s), com o operando A = 9 novamente. Observando atentamente cada valor de saída, é possível notar que o comportamento obtido é o mesmo do que o desejado na especificação do circuito.

4. Projeto 4: Circuito de teste da ULA

Este projeto é a parte final da atividade, consistindo, basicamente, em encapsular o circuito anterior em um módulo para testá-lo na FPGA, usando da associação de entradas e pinos apresentada no roteiro. Como se trata do mesmo circuito de ULA estendida, é desnecessário repetir seu escopo, especificações e simulações. Dessa forma, na figura 8 é apresentada o diagrama final do circuito de teste da ULA (circuito “bus_ula_teste”). Esse módulo foi testado com os mesmos casos apresentados nas simulações anteriores.

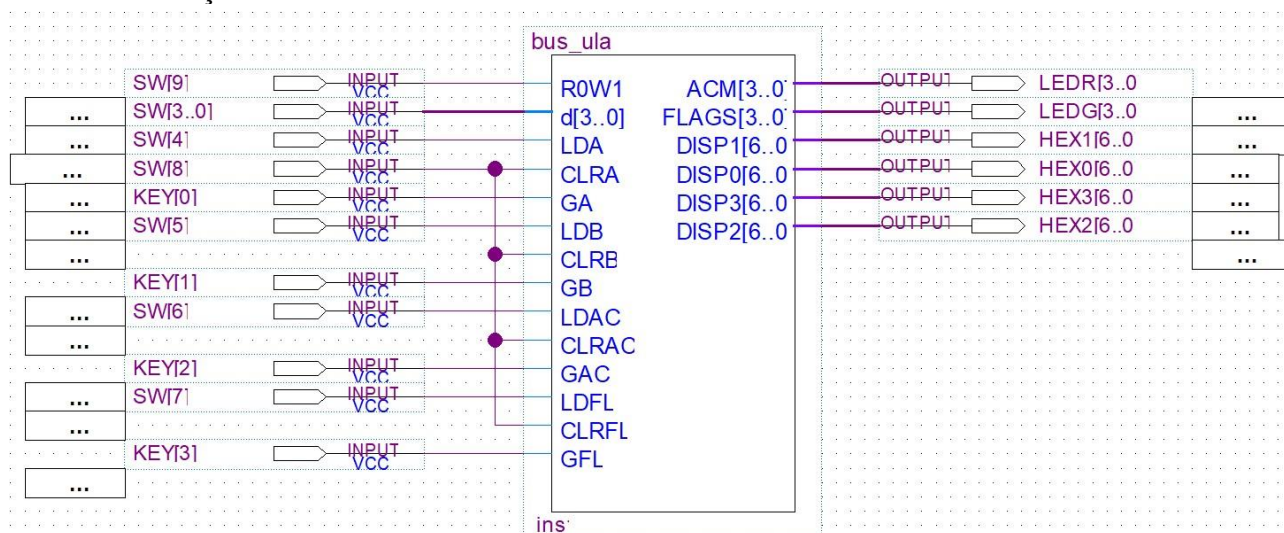


Figura 8: Diagrama esquemático do circuito de teste da ULA estendida. Como dito anteriormente, a ULA estendida foi encapsulada no módulo denominado de “bus_ula” e a associação de entradas e pinos para teste foi adotada de acordo com o apresentado no roteiro da atividade. Os testes aplicados na placa de desenvolvimento para esse circuito foram idênticos aos simulados para a ULA estendida e provocaram resultados análogos.