UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC

ALEX BEVILAQUA

BRUNO CESCA

MATEUS CALZA

# UNIDADE LÓGICA ARITMÉTICA COM FPGA

DESCRIÇÃO DE HARDWARE


JOAÇABA

2020

## Objetivo

Utilizar a linguagem de descrição de hardware VHDL para desenvolver um circuito que implemente uma unidade lógica aritmética - ULA, com as operações de adição, subtração, AND e XOR.

## Codificação

```
PACKAGE const IS
  CONSTANT n : INTEGER := 8; -- bits
  CONSTANT m : INTEGER := 2 ** n; -- total de combinacoes
  CONSTANT p : INTEGER := n + 1; -- bits com sinalizacao
  CONSTANT q : INTEGER := 2 ** p; -- total de combinacoes com sinalizacao
END const;
```

```
LIBRARY ieee;
USE work.const.ALL;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;
USE ieee.numeric_std.ALL;
```

```
ENTITY alu IS
  PORT (
    a : IN std_logic_vector(n - 1 DOWNTO 0);
    b : IN std_logic_vector(n - 1 DOWNTO 0);

    mode : IN INTEGER RANGE 0 TO 3; -- 0 (add), 1 (sub), 2 (and), 3 (xor)

    result : OUT std_logic_vector(n - 1 DOWNTO 0);
    carryBorrow : OUT BIT
  );
END alu;
```

```vhdl
ARCHITECTURE vhdl OF alu IS
  SIGNAL signedBits : std_logic_vector (p DOWNTO 0);
BEGIN
  PROCESS (a, b, mode)
    VARIABLE intResult : INTEGER;
    VARIABLE intA : INTEGER;
    VARIABLE intB : INTEGER;

  BEGIN
    IF mode = 0 THEN
      intA := to_integer(unsigned(a));
      intB := to_integer(unsigned(b));
      intResult := intA + intB;
      IF intResult < m THEN
        carryBorrow <= '0';
        result <= a + b;
      ELSE
        carryBorrow <= '1';
        result <= "00000000";
      END IF;
      -- result <= a - b;
    ELSIF mode = 1 THEN
      intA := to_integer(unsigned(a));
      intB := to_integer(unsigned(b));
      intResult := intA - intB;
      IF intResult >= 0 THEN
        carryBorrow <= '0';
        result <= a - b;
      ELSE
        carryborrow <= '1';
        result <= "00000000";
      END IF;
    ELSIF mode = 2 THEN
      result <= a AND b;
      carryBorrow <= '0';
    ELSIF mode = 3 THEN
      result <= a XOR b;
      carryBorrow <= '0';
    ELSE
      result <= "00000000";
      carryBorrow <= '0';
    END IF;
  END PROCESS;
END vhdl;
```

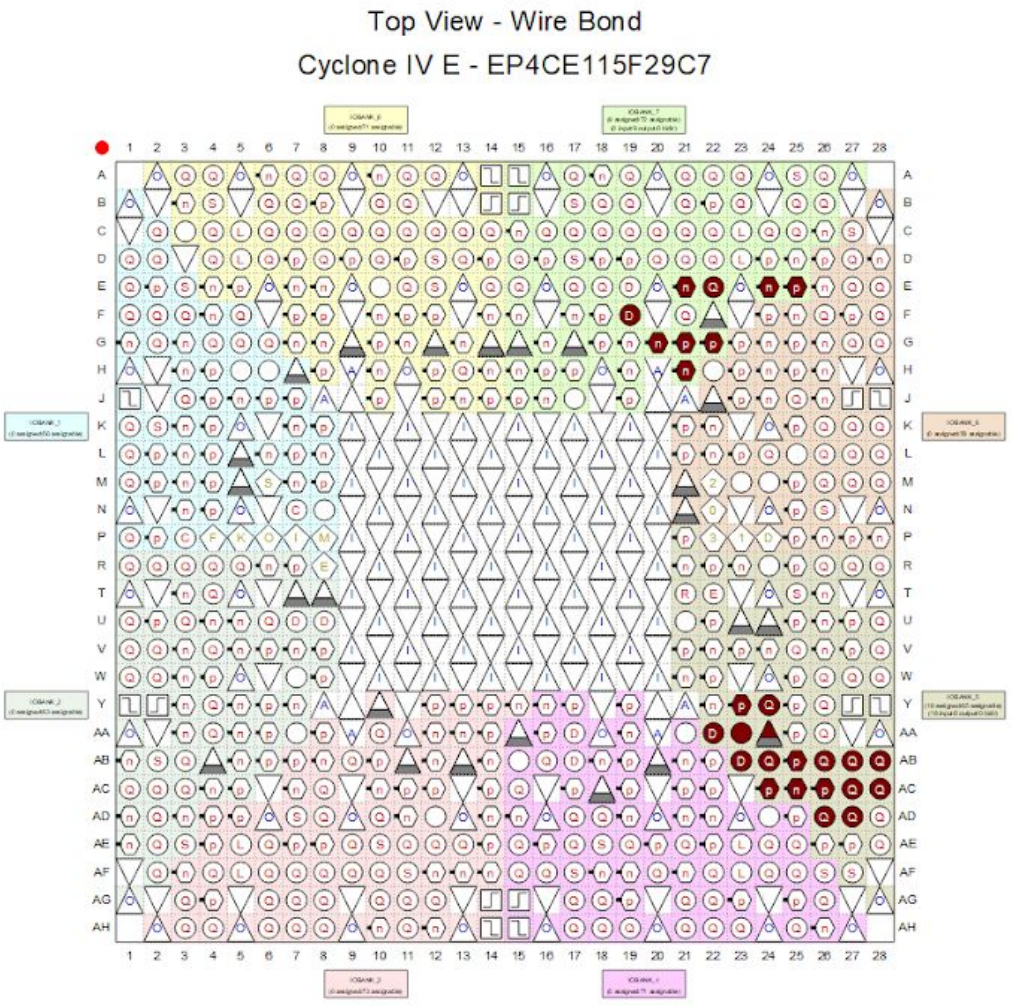
## Teste de mesa programado

```vhdl
LIBRARY ieee;
USE work.const.ALL;
USE ieee.std_logic_1164.ALL;

ENTITY alu_testbench IS
END alu_testbench;

ARCHITECTURE vhdl OF alu_testbench IS
  COMPONENT alu
    PORT (
      a : IN std_logic_vector(n - 1 DOWNTO 0);
      b : IN std_logic_vector(n - 1 DOWNTO 0);
      mode : IN INTEGER RANGE 0 TO 3;
      result : OUT std_logic_vector(n - 1 DOWNTO 0);
      carryBorrow : OUT BIT
    );
  END COMPONENT;

  FOR alu_0 : alu USE ENTITY work.alu;

  SIGNAL a : std_logic_vector(n - 1 DOWNTO 0);
  SIGNAL b : std_logic_vector(n - 1 DOWNTO 0);
  SIGNAL mode : INTEGER RANGE 0 TO 3;
  SIGNAL result : std_logic_vector(n - 1 DOWNTO 0);
  SIGNAL carryBorrow : BIT;

BEGIN
  alu_0 : alu PORT MAP(a => a, b => b, mode => mode, result => result, carryBorrow
=> carryBorrow);
  PROCESS
    TYPE pattern_type IS RECORD
      a : std_logic_vector(n - 1 DOWNTO 0);
      b : std_logic_vector(n - 1 DOWNTO 0);
      mode : INTEGER RANGE 0 TO 3;
      result : std_logic_vector(n - 1 DOWNTO 0);
      carryBorrow : BIT;
    END RECORD;
    TYPE pattern_array IS ARRAY (NATURAL RANGE <>) OF pattern_type;
    CONSTANT patterns : pattern_array :=
    (
    ("00011011", "00000111", 0, "00100010", '0'),
    ("00000111", "00000011", 1, "00000100", '0'),
    ("00000111", "00011011", 2, "00000011", '0'),
    ("00000001", "00000111", 3, "00000110", '0'),
    ("10000000", "10000000", 0, "00000000", '1')
    );
  BEGIN
    FOR i IN patterns'RANGE LOOP
      a <= patterns(i).a;
      b <= patterns(i).b;
      mode <= patterns(i).mode;
      WAIT FOR 1 ns;

      ASSERT result = patterns(i).result AND carryBorrow = patterns(i).carryBorrow
      REPORT "Ocorreu um problema! Onde A = "
        & to_hstring(a)
        & ", B = "
        & to_hstring(b)
        & ", MODE = "
        & INTEGER'image(mode) SEVERITY error;

      ASSERT result = patterns(i).result
      REPORT "Resultado errado = "
        & to_hstring(result)
        & ", resultado esperado = "
        & to_hstring(patterns(i).result) SEVERITY error;

      ASSERT carryBorrow = patterns(i).carryBorrow
      REPORT "Carry/borrow errado = "
        & BIT'image(carryBorrow)
        & ", carry/borrow esperado = "
        & BIT'image(patterns(i).carryBorrow) SEVERITY error;
    END LOOP;
    ASSERT false REPORT "Fim do teste" SEVERITY note;

    WAIT;
  END PROCESS;
END vhdl;
```

Todos os códigos versionados podem ser encontrados em:

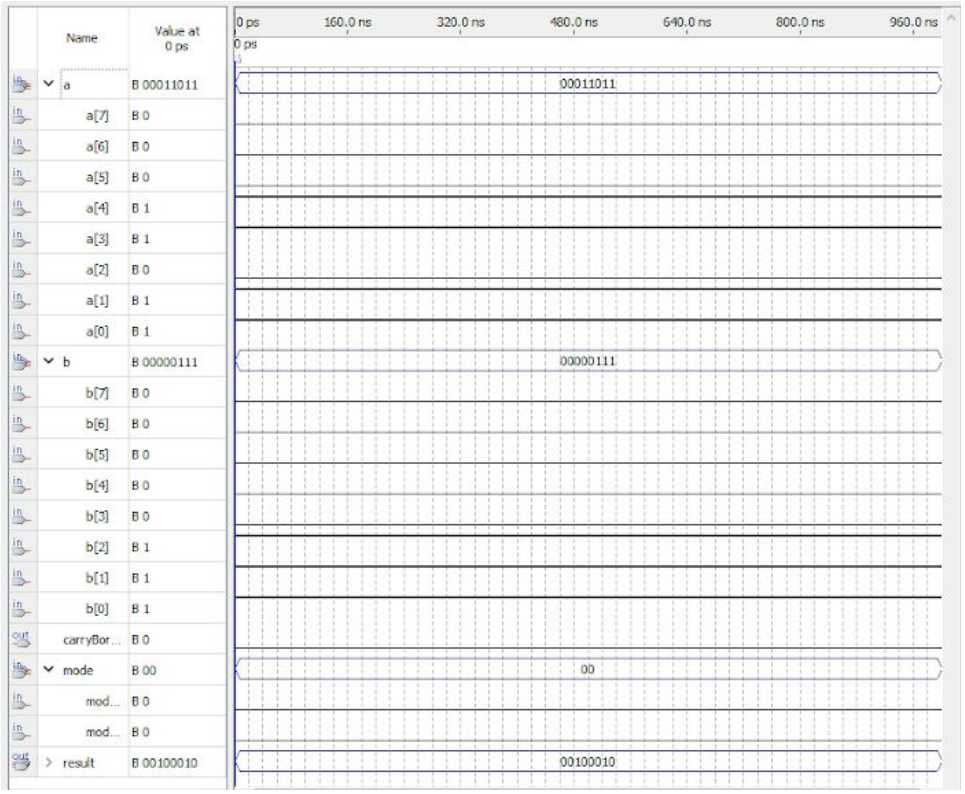
https://github.com/mateuscalza/fpga-alu

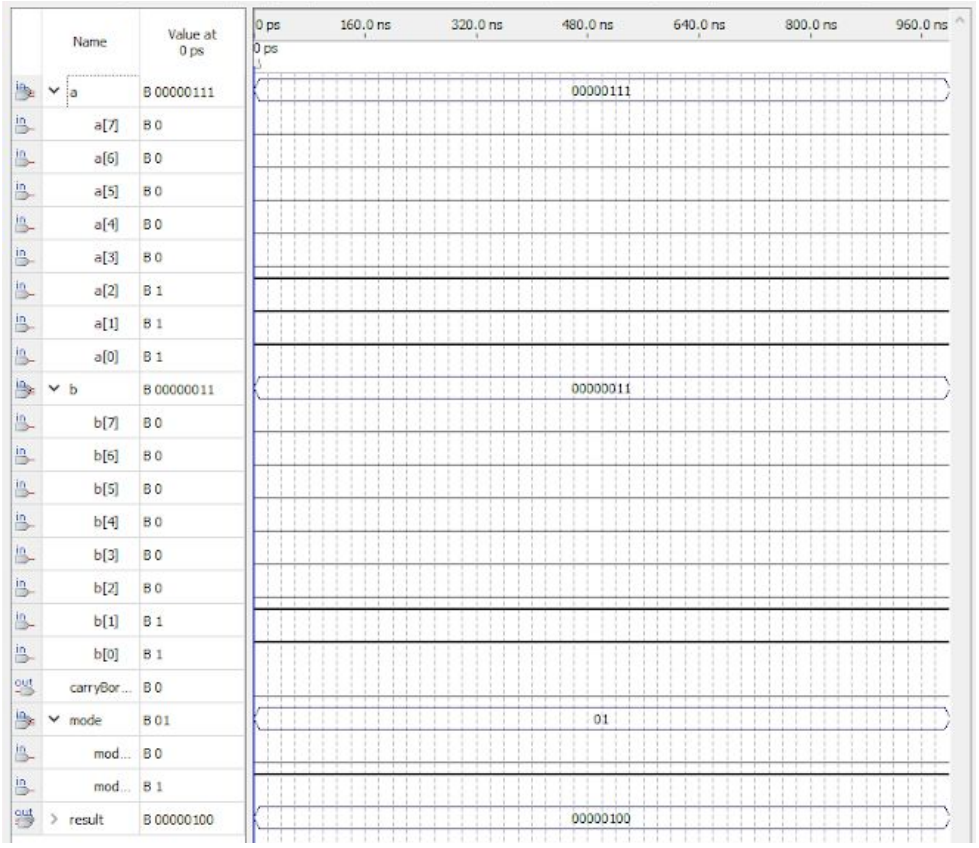
# Pinout

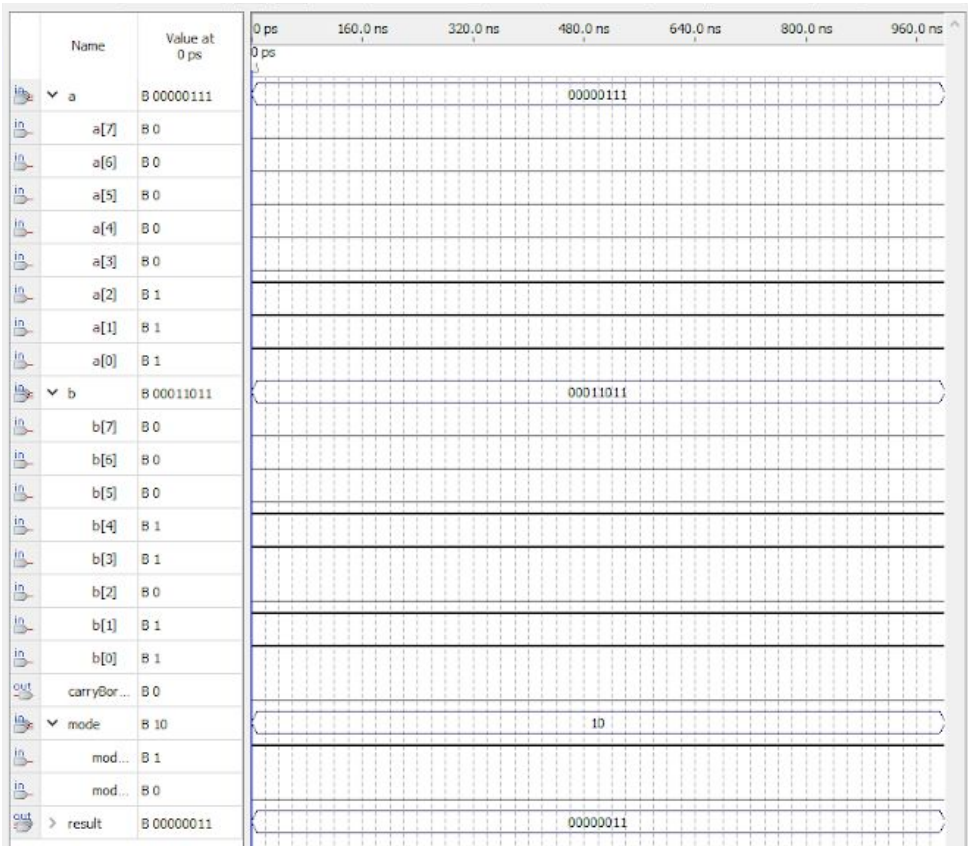


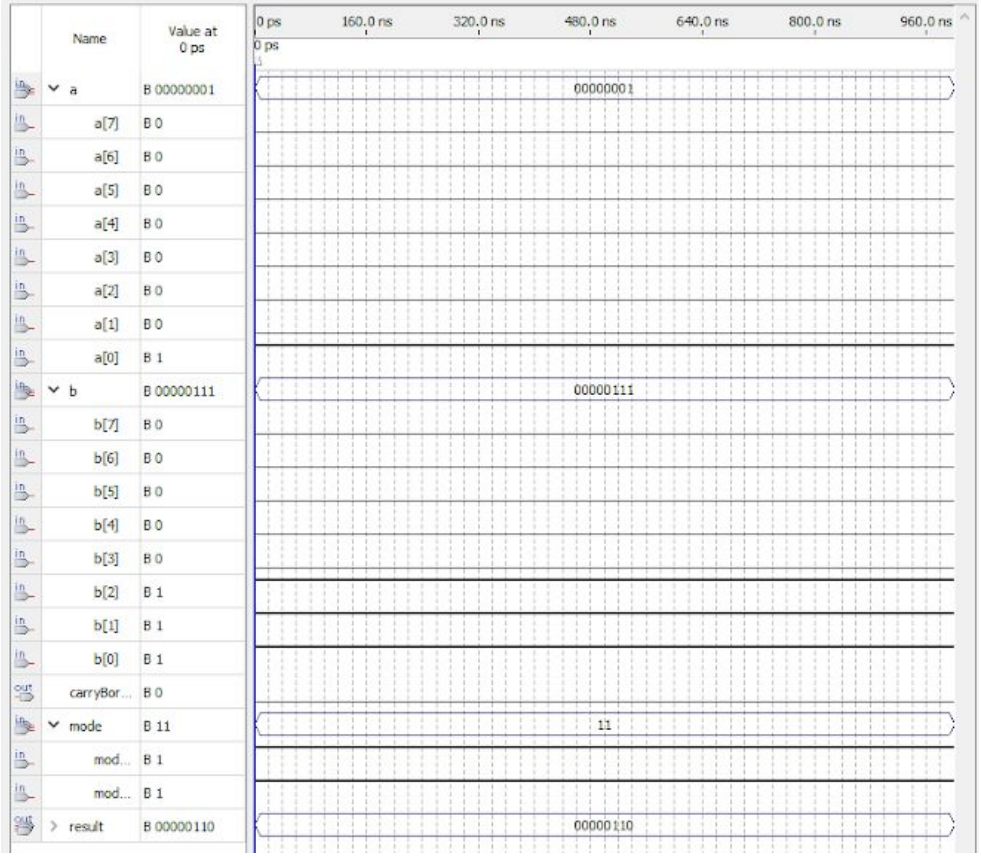

| Node Name | Direction | Location | I/O Bank | VREF Group | Fitter Location | I/O Standard | Reserved | Current Strength | Slew Rate | Differential Pair |
|---|---|---|---|---|---|---|---|---|---|---|
| a[7] | Input | PIN_Y23 | 5 | B5_N2 | PIN_N8 | 2.5 V (default) | | 8mA (default) | | |
| a[6] | Input | PIN_Y24 | 5 | B5_N2 | PIN_L4 | 2.5 V (default) | | 8mA (default) | | |
| a[5] | Input | PIN_AA22 | 5 | B5_N2 | PIN_R7 | 2.5 V (default) | | 8mA (default) | | |
| a[4] | Input | PIN_AA23 | 5 | B5_N2 | PIN_K7 | 2.5 V (default) | | 8mA (default) | | |
| a[3] | Input | PIN_AA24 | 5 | B5_N2 | PIN_L8 | 2.5 V (default) | | 8mA (default) | | |
| a[2] | Input | PIN_AB23 | 5 | B5_N2 | PIN_R1 | 2.5 V (default) | | 8mA (default) | | |
| a[1] | Input | PIN_AB24 | 5 | B5_N2 | PIN_L6 | 2.5 V (default) | | 8mA (default) | | |
| a[0] | Input | PIN_AC24 | 5 | B5_N2 | PIN_P1 | 2.5 V (default) | | 8mA (default) | | |
| b[7] | Input | PIN_AB26 | 5 | B5_N1 | PIN_L7 | 2.5 V (default) | | 8mA (default) | | |
| b[6] | Input | PIN_AD26 | 5 | B5_N2 | PIN_P2 | 2.5 V (default) | | 8mA (default) | | |
| b[5] | Input | PIN_AC26 | 5 | B5_N2 | PIN_M5 | 2.5 V (default) | | 8mA (default) | | |
| b[4] | Input | PIN_AB27 | 5 | B5_N1 | PIN_J7 | 2.5 V (default) | | 8mA (default) | | |
| b[3] | Input | PIN_AD27 | 5 | B5_N2 | PIN_M3 | 2.5 V (default) | | 8mA (default) | | |
| b[2] | Input | PIN_AC27 | 5 | B5_N2 | PIN_J6 | 2.5 V (default) | | 8mA (default) | | |
| b[1] | Input | PIN_AC28 | 5 | B5_N2 | PIN_L3 | 2.5 V (default) | | 8mA (default) | | |
| b[0] | Input | PIN_AB28 | 5 | B5_N1 | PIN_R2 | 2.5 V (default) | | 8mA (default) | | |
| carryBorrow | Output | PIN_F19 | 7 | B7_N0 | PIN_L2 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| mode[1] | Input | PIN_AB25 | 5 | B5_N1 | PIN_K8 | 2.5 V (default) | | 8mA (default) | | |
| mode[0] | Input | PIN_AC25 | 5 | B5_N2 | PIN_U3 | 2.5 V (default) | | 8mA (default) | | |
| result[7] | Output | PIN_G21 | 7 | B7_N1 | PIN_L1 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[6] | Output | PIN_G22 | 7 | B7_N2 | PIN_J5 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[5] | Output | PIN_G20 | 7 | B7_N1 | PIN_M1 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[4] | Output | PIN_H21 | 7 | B7_N2 | PIN_M7 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[3] | Output | PIN_E24 | 7 | B7_N1 | PIN_N3 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[2] | Output | PIN_E25 | 7 | B7_N1 | PIN_M8 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[1] | Output | PIN_E22 | 7 | B7_N0 | PIN_N4 | 2.5 V (default) | | 8mA (default) | 2 (default) | |
| result[0] | Output | PIN_E21 | 7 | B7_N0 | PIN_M2 | 2.5 V (default) | | 8mA (default) | 2 (default) | |

# Simulações


Name
Value at 0 ps
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
a
B 00011011
00011011
a[7]
B 0
a[6]
B 0
a[5]
B 0
a[4]
B 1
a[3]
B 1
a[2]
B 0
a[1]
B 1
a[0]
B 1
b
B 00000111
00000111
b[7]
B 0
b[6]
B 0
b[5]
B 0
b[4]
B 0
b[3]
B 0
b[2]
B 1
b[1]
B 1
b[0]
B 1
carryBor...
B 0
mode
B 00
00
mod...
B 0
mod...
B 0
result
B 00100010
00100010


Name
Value at 0 ps
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
a
B 00000111
00000111
a[7]
B 0
a[6]
B 0
a[5]
B 0
a[4]
B 0
a[3]
B 0
a[2]
B 1
a[1]
B 1
a[0]
B 1
b
B 00000011
00000011
b[7]
B 0
b[6]
B 0
b[5]
B 0
b[4]
B 0
b[3]
B 0
b[2]
B 0
b[1]
B 1
b[0]
B 1
carryBor...
B 0
mode
B 01
01
mod...
B 0
mod...
B 1
result
B 00000100
00000100

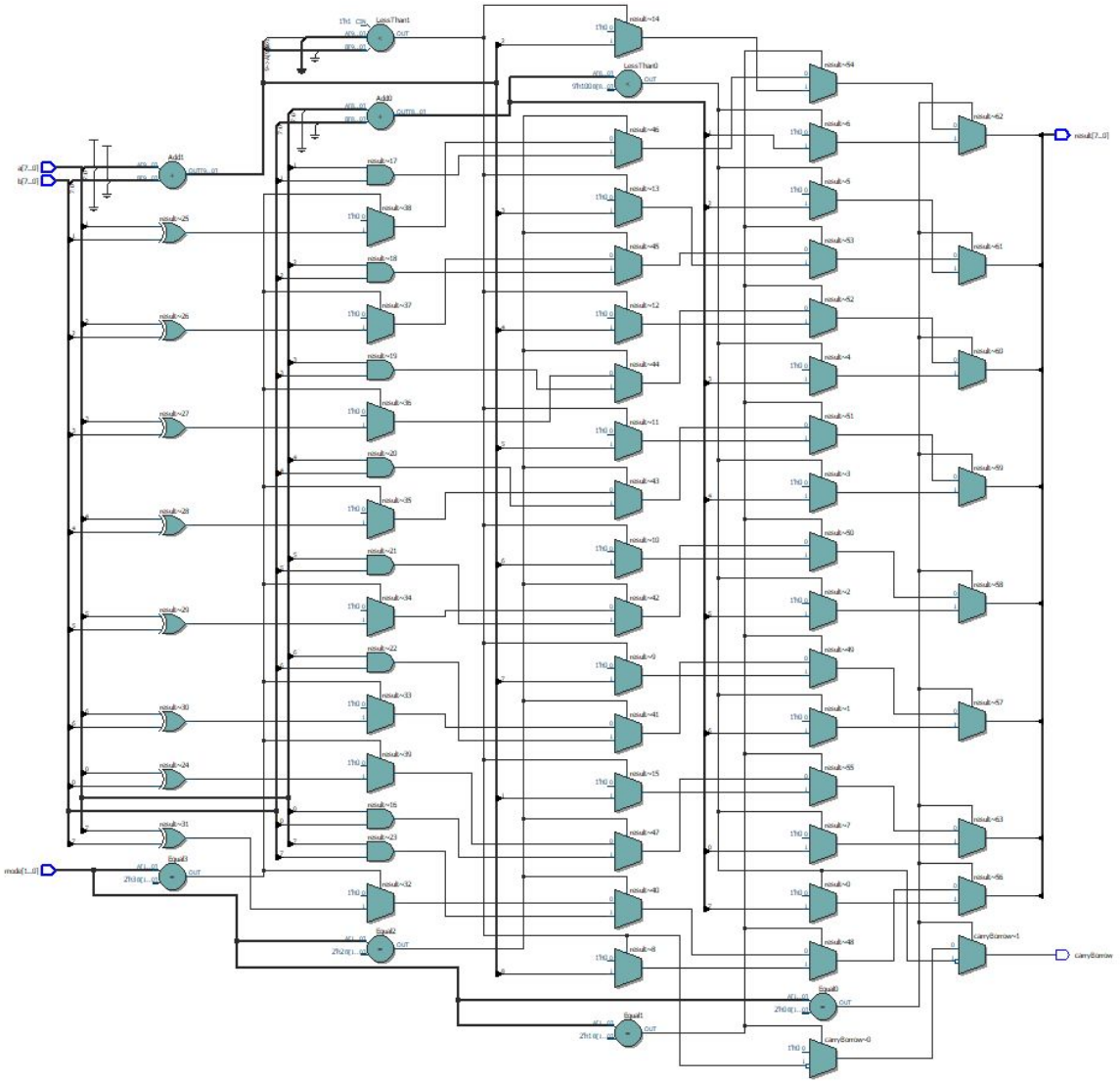

Name
Value at 0 ps
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
a
B 00000111
00000111
a[7]
B 0
a[6]
B 0
a[5]
B 0
a[4]
B 0
a[3]
B 0
a[2]
B 1
a[1]
B 1
a[0]
B 1
b
B 00011011
00011011
b[7]
B 0
b[6]
B 0
b[5]
B 0
b[4]
B 1
b[3]
B 1
b[2]
B 0
b[1]
B 1
b[0]
B 1
carryBor...
B 0
mode
B 10
10
mod...
B 1
mod...
B 0
result
B 00000011
00000011


Name
Value at 0 ps
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
a
B 00000001
00000001
a[7]
B 0
a[6]
B 0
a[5]
B 0
a[4]
B 0
a[3]
B 0
a[2]
B 0
a[1]
B 0
a[0]
B 1
b
B 00000111
00000111
b[7]
B 0
b[6]
B 0
b[5]
B 0
b[4]
B 0
b[3]
B 0
b[2]
B 1
b[1]
B 1
b[0]
B 1
carryBor...
B 0
mode
B 11
11
mod...
B 1
mod...
B 1
result
B 00000110
00000110

## Teste de mesa manual

| A | A binário | B | B binário | Modo | Resultado | Resultado binário | Carry |
|---|---|---|---|---|---|---|---|
| 27 | 11011 | 7 | 111 | 0 - ADD | 34 | 100010 | 0 |
| 7 | 111 | 3 | 11 | 1 - SUB | 4 | 100 | 0 |
| 7 | 111 | 27 | 11011 | 2 - AND | 3 | 11 | 0 |
| 1 | 1 | 7 | 111 | 3 - XOR | 6 | 110 | 0 |
| 128 | 10000000 | 128 | 10000000 | 0 - ADD | 256 | 00000000 | 1 |

## RTL



Arquivo com o RTL completo disponível em:

https://raw.githubusercontent.com/mateuscalza/fpga-alu/master/rtl.pdf