MicroMem

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 11.11.2020 10:06:47
7.// Design Name:
8.// Module Name: tb_MicroMem
9.// Project Name:
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revisión:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module tb_MicroMem;
24.
25.reg Clk;
            26.reg Rst;
    27.
28.MicroMem uut(
 29. .Clk(Clk),
 30. .Rst(Rst)
31. )
   32.
33.initial
 34.begin
35.Rst=1;
36.Clk=0;
37.
38.#2 Rst=0;
39.#10 $finish;
 40.end
 41.
42.always
 43. #1 Clk = !Clk;
44.endmodule
```

45.

Microprocesador_8bits

```
1.  `timescale 1ns / 1ps
2.  //////////////////////////////////////////////////////////////////////////////////
3.  // Company:
4.  // Engineer:
5.  //
6.  // Create Date: 10.11.2020 19:44:53
7.  // Design Name:
8.  // Module Name: Microprocesador_RISC_8bits
9.  // Project Name:
10. // Target Devices:
11. // Tool Versions:
12. // Description:
13. //
14. // Dependencies:
15. //
16. // Revision:
17. // Revision 0.01 - File Created
18. // Additional Comments:
19. //
20. //////////////////////////////////////////////////////////////////////////////////
21.

22.

23.

24.module Microprocesador_RISC_8bits(
 25. input       Clk,
26.  input       Rst,
 27.  input  [8:0] Instruction,
 28. input  [7:0] DataIn_Bus,
 29. output [7:0] Address_Instruction_Bus,
 30. output [7:0] DataOut_Bus,
 31. output [7:0] Address_Data_Bus,
  32.output      L_E
 33. );
```

```
34.////////////////////////////////////////////////////////////////////////////////////////////////////35
.////////////////
36.wire [7:0]W_Num;
37.wire[2:0] W_Sel;
38.wire [5:0] W_Sel_R;
39.wire  W_Sel_RLE;
40.wire [7:0] W_Dato_N;
41.wire [7:0] W_RX;
42.wire [7:0] W_RY;
43.wire [2:0] W_Ope;
44.wire [7:0] W_Resultado;
45.wire [2:0] W_Flags;
46.wire [1:0] W_Sel_Sali;
47.////////////////////////////////////////////////////////////////////////////////////////////////////48
.///////////
49.Sel_Datos Selector (
  50. .DataIn_Bus(DataIn_Bus),
 51  .Resultado(W_Resultado),
52.    .Ry(W_RY),
 53.   .Address_Instruction_Bus(Address_Instruction_Bus),
 54  .Num(W_Num),
 55.  .Sel(W_Sel),
 56.  .Dato_N(W_Dato_N)
57.);
58.
59.////////////////////////////////////////////////////////////////////////////////////////////////////60
.////////////////
61.ALU operaciones (
  62. .Ope(W_Ope),
  63. .Ry(W_RX),
 64.  .Rx(W_RY),
 65.  .Resultado(W_Resultado),
```

```
 66.  .Flags(W_Flags)
67.);
68.////////////////////////////////////////////////////////////////////////////////////////////////////69
.////////////////
70.Conca_Num Inmediato (
 71  .Instruction(Instruction),
 72. .Num(W_Num)
73.);
74////////////////////////////////////////////////////////////////////////////////////////////////////75
.///////////////
76.Banco_R Registros (
 77. .Sel_R(W_Sel_R), // Tres bits para cada uno,Sel_R[2:0] para Rx y Sel_R[5:3] para Ry
 78.  .Dato_N(W_Dato_N),// Dato de entrada que sera almacenado
 79.  .Rx(W_RX),
 80.   .Ry(W_RY),
 81. .Sel_RLE(W_Sel_RLE), // Con 0 va a leer y con 1 a escribir
 82.   .Rst(Rst),
 83.   .Clk(Clk)
84.);
85.////////////////////////////////////////////////////////////////////////////////////////////////////86
.///////////////
87.Deco_Ins Instrucciones (
 88.  .Rx(W_RY),
 89.  .Instruction(Instruction),
 90.  .Flags(W_Flags),
 91.  .Sel_RLE(W_Sel_RLE), //Salida que dice si se va a leer o guardar
92.    .Sel_R(W_Sel_R),// Selecciona el registro a utilizarse
 93 .Sel_Sali(W_Sel_Sali), //Señal que controla la salida
 94.  .Ope(W_Ope), //Operación de la ALU
 95.  .Address_Instruction_Bus(Address_Instruction_Bus),
 96. .Sel(W_Sel), //Sel de datos
 97.  .Clk(Clk),
```

```
 98.   .Rst(Rst)
99.);
100.////////////////////////////////////////////////////////////////////////////////////////////////////10
1.//////////////////
102.Control_Sali Salida (
 103.  .Ry(W_RY),
  104. .Rx(W_RX),
 105  .Num(W_Num),
 106.  .Sel_Sali(W_Sel_Sali),
 107.  .Address_Data_Bus(Address_Data_Bus),
  108. .DataOut_Bus(DataOut_Bus),
 109.  .L_E(L_E),
 110.   .Clk(Clk),
 111.  .Rst(Rst)
112.);
113.endmodule
```

ALU

```
1.  `timescale 1ns / 1ps
2.  //////////////////////////////////////////////////////////////////////////////////
3.  // Company:
4.  // Engineer:
5.  //
6.  // Create Date: 10.11.2020 19:30:55
7.  // Design Name:
8.  // Module Name: ALU
9.  // Project Name:
10. // Target Devices:
11. // Tool Versions:
12. // Description:
13. //
14. // Dependencies:
15. //
16. // Revision:
17. // Revision 0.01 - File Created
18. // Additional Comments:
19. //
20. //////////////////////////////////////////////////////////////////////////////////
21.
```

```
22.
23.module ALU(
 24.input [2:0] Ope,
 25.  input [7:0] Ry,
 26.  input [7:0] Rx,
 27. output [7:0] Resultado,
 28. output [2:0] Flags
 29.  );
 30.  reg [8:0] R0;
 31.
 32. always@*
  33.  begin
   34. case (Ope)
   35.  3'b000: begin R0<=Ry+Rx;end
   36.  3'b001: begin R0<=Ry-Rx; end
   37.  3'b010: begin R0<=Ry<<Rx; end
   38.   3'b011: begin R0<=Ry>>Rx;end
   39.   3'b100: begin R0<=~Rx; end
   40. 3'b101: begin R0<=Ry&Rx;end
   41.  3'b110: begin R0<=Ry|Rx;end
   42.  3'b111: begin R0<=Ry^Rx;end
   43.  endcase
  44.  end
   45. assign Resultado= R0[7:0]; //Los 7 bits menos significativos son el resultado de la operacion
   46. //Bit más siginificativo es el de acarreo
  47.  //Banderas
   48. assign Flags[0]=&(~R0);//Bit cero
  49.   assign Flags[1]= R0[8]; //Bit de acarreo C
  50.  assign Flags[2]=R0[7]; //N:Negativo
   51.
   52.
```

```
53.
54.endmodule
```

Sel_Datos

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company: Universidad Autonoma de Zacatecas
4.// Engineer: Roboticos
5.//
6.// Create Date: 18.10.2020 15:48:20
7.// Design Name: Sel_Datos
8.// Module Name: Sel_Datos
9.// Project Name: Microprocesador_RISC_8bits
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Sel_Datos(
24.     input [7:0] DataIn_Bus,
25.     input [7:0] Resultado,
26.     input [7:0] Ry,
27.     input [7:0] Address_Instruction_Bus,
28.     input [7:0] Num,
29.     input [2:0] Sel,
30.     output reg [7:0] Dato_N
31.     );
32.
33.
 34.
35.     always@*
36.     begin
 36.     case (Sel)
37.     3'b000 :begin Dato_N<=Resultado;end
38.     3'b001 :begin  Dato_N<=DataIn_Bus;end
39.     3'b010 :begin  Dato_N<=Num;end
 40.     3'b011 :begin  Dato_N<=Address_Instruction_Bus;end
```

```
41.       3'b100 :begin  Dato_N<=Ry;end
 42.      default: begin  Dato_N<=0;end
43.       endcase
44.
45.       end
46.
47.endmodule
```

Conca_Num

```
1. `timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 10.11.2020 23:00:35
7.// Design Name:
8.// Module Name: Conca_Num
9.// Project Name:
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Conca_Num
24.#(parameter n = 8 )
25.(
26.input          [n:0]   Instruction,
          27.output reg [n-1:0] Num
          28.);
          29.reg [n-1:0]VD;
30. always@*
          31.begin
32.
 33.VD = 8'b0
  34.Num <= {VD[7:3],Instruction[2:0]};
   35.
   36.end
```

```
37.endmodule
```

Deco_Ins

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 10.11.2020 19:26:28
7.// Design Name:
8.// Module Name: Deco_Ins
9.// Project Name:
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Deco_Ins(
24.input [7:0] Rx, //Valor que viene del banco de registros
25.input [8:0] Instruction,
26.input [2:0] Flags,
27.output Sel_RLE, //Salida que dice si se va a leer o guardar
28.output [5:0] Sel_R,// Selecciona el registro a utilizarse
29.output [1:0] Sel_Sali, //Señal que controla la salida
30.output [2:0] Ope, //Operación de la ALU
31.output [7:0] Address_Instruction_Bus,
32.output [2:0] Sel, //Sel de datos
33.input Clk,
34.input Rst
35.);
    36.
37.wire [3:0]CondJ; //Cable que une al Jum y al Decodificador
38.reg [2:0] NFlags; //Registro para almacenar las banderas despues de una math
 39.
40.Decodificador c_Decodificador(
41..Instruction(Instruction),
42..Cond(CondJ),
```

```
43..Sel_RLE(Sel_RLE),
44..Sel_R(Sel_R),
 45..Sel_Sali(Sel_Sali),
46..Ope(Ope),
  47..Sel(Sel)
48.
  49.
50.Jump c_Jump(
  51..Flags(NFlags),
   52..Rx(Rx),
   53..Cond(CondJ),
   54..Clk(Clk),
   55..Rst(Rst),
   56..Address_Instruction_Bus(Address_Instruction_Bus)
  57.);
  58.
  59.
   60.always @(posedge Clk, posedge Rst) begin
     61.if(Rst)
       62.begin
   63.NFlags<=0;
 64.end
65.else
  66.if (Instruction[8:6]==3'b110)
    67.NFlags<=Flags;
     68.          end
69.endmodule
```

Decodificador

```
1.  `timescale 1ns / 1ps
2.  //////////////////////////////////////////////////////////////////////////////////
3.  // Company:
4.  // Engineer:
5.  //
6.  // Create Date: 10.11.2020 19:27:56
7.  // Design Name:
8.  // Module Name: Decodificador
9.  // Project Name:
10. // Target Devices:
11. // Tool Versions:
12. // Description:
13. //
14. // Dependencies:
15. //
16. // Revision:
17. // Revision 0.01 - File Created
```

```
18. // Additional Comments:
19. //
20. //////////////////////////////////////////////////////////////////////////////////
21.
22.
23. module Decodificador(
24.     input [8:0] Instruction,
25.     output reg [3:0] Cond,
26.     output reg Sel_RLE, //Señal que indica si se va a leer o escribir
27.     output reg [5:0] Sel_R, //Señal para seleccionar los registros
28.     output reg [1:0] Sel_Sali, //Señal que controla el modulo Control_Sali
29.     output reg [2:0] Ope, //Señal que indica que operacón
30.     output reg [2:0] Sel //Selecciona el dato en el modulo Sel_Dato
31.     );
32.
33.     always @(Instruction) begin
34.         case (Instruction [8:6])
35.          //Load Rx,#Num
36.             3'b001: begin
37.                     Ope<=0; Sel_R<= {3'b000,Instruction[5:3]}; Sel_RLE<=1;
    Sel_Sali<=2'b00; Sel<=3'b010; Cond<=4'b0001;   end
38.          //Load Rx,[Ry]
39.             3'b010: begin
40.                     Ope<=0; Sel_R<={Instruction[2:0],Instruction[5:3]}; Sel_RLE<=1;
    Sel_Sali<=2'b01; Sel<= 3'b001;Cond<=4'b0001;  end
41.          //Store #Num
42.             3'b011: begin
43.                     Ope<=0; Sel_R<= {3'b000, Instruction[5:3]}; Sel_RLE<=0;
    Sel_Sali<=2'b10; Sel<=3'b010; Cond<=4'b0001; end
44.          //Store [Rx],Ry
45.             3'b100: begin
46.                     Ope<=0; Sel_R<={Instruction[2:0], Instruction[5:3]}; Sel_RLE<=0;
    Sel_Sali<=2'b11; Sel<=3'b100;Cond<=4'b0001;  end
47.          //Move Rx,Ry
48.             3'b101: begin
49.                      Ope<=0; Sel_R<={Instruction[2:0], Instruction[5:3]}; Sel_RLE<=1;
    Sel_Sali<=2'b00; Sel<=3'b100;Cond<=4'b0001; end
50.          //Math
51.             3'b110: begin
52.                      Ope<=Instruction[2:0]; Sel_R<={Instruction[5:3],3'b000}; Sel_RLE<=1;
    Sel_Sali<=2'b00; Sel<=3'b000;Cond<=4'b0001;  end
53.          //Jump
54.             3'b111: begin
55.                 if (Instruction[2:0]==3'b001)  // jump sin condicion y guardar pc en R7
56.                     begin
57.                      Ope<=0; Sel_R<={Instruction[5:3], 3'b111}; Sel_RLE<=1;
    Sel_Sali<=2'b00; Sel<=3'b011;Cond<={1'b1, Instruction[2:0]};  end
58.
```

```
            else
            begin Ope<=0; Sel_R<={Instruction[5:3], 3'b000}; Sel_RLE<=0;
    Sel_Sali<=2'b00; Sel<=3'b011;Cond<={1'b1, Instruction[2:0]};  end
            end
         //Nop
           3'b000: begin
                   Ope<=0; Sel_R<=0; Sel_RLE<=0; Sel_Sali<=2'b00;
    Sel<=3'b111;Cond<=4'b0001; end

           default: begin Ope<=0; Sel_R<=0; Sel_RLE<=0; Sel_Sali<=2'b00;
    Sel<=3'b111;Cond<=4'b0001; end

        endcase



    end

endmodule
```

Jump

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 10.11.2020 19:29:03
// Design Name:
// Module Name: Jump
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Jump(
    input [2:0] Flags,
```

```
25.  input [7:0] Rx,
26.  input [3:0] Cond,
27.  output [7:0] Address_Instruction_Bus,
28.  input Clk,
29.  input Rst
30.  );
31.
32.  reg [7:0]pc; //Guarda el Address_Instruction_Bus
33.
34.      always@(posedge Clk,posedge Rst)
35.       begin
36.        if(Rst)
37.             pc<=0;
38.             else
39.                 case(Cond)
40.                     4'b1000: pc <= Rx;
41.                     4'b1001: pc <= Rx;
42.
43.                    4'b1010:
44.                     begin
45.                        if(Flags[0])
46.                        pc <= Rx;
47.                            else
48.                         pc = pc+1'b1;
49.                        end
50.                    4'b1011:
51.                     begin
52.                        if(~Flags[0])
53.                      pc <= Rx;
54.                            else
55.                       pc= pc+1'b1;
56.                        end
57.                     4'b1100:
58.                     begin
59.                        if(Flags[1])
60.                        pc <= Rx;
61.                            else
62.                       pc = pc+1'b1;
63.                         end
64.                     4'b1101:
65.                     begin
66.                        if(~Flags[1])
67.                        pc <= Rx;
68.                            else
69.                       pc = pc+1'b1;
70.                         end
71.                     4'b1110:
72.                     begin
```

```
                    if(Flags[2])
                    pc <= Rx;
                        else
                     pc= pc+1'b1;
                       end
                  4'b1111:
                  begin
                     if(~Flags[2])
                     pc<= Rx;
                        else
                      pc= pc+1'b1;
                       end
                   4'b0000:
                        begin pc= pc; end   //Codificación
                        default: pc<=pc+1'b1;
                         endcase
          end
         assign Address_Instruction_Bus=pc;

endmodule
```

Banco_R

```
timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: Universidad Autonoma de Zacatecas
// Engineer: Roboticos
//
// Create Date: 18.10.2020 15:48:20
// Design Name: Sel_Datos
// Module Name: Sel_Datos
// Project Name: Microprocesador_RISC_8bits
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Banco_R(

```

```
25.     input [5:0] Sel_R, // Tres bits para cada uno,Sel_R[2:0] para Rx y Sel_R[5:3] para
    Ry
26.     input [7:0] Dato_N,// Dato de entrada que sera almacenado
27.     output[7:0] Rx,
28.     output[7:0] Ry,
29.     input       Sel_RLE, // Con 0 va a leer y con 1 a escribir
30.     input Rst,
31.     input Clk
32.     );
33.
34.     reg [7:0] Registro [7:0]; //arreglo, registro de 8 bits y nos dice que son 8 registros
35.
36.     always @(posedge Clk, posedge Rst) begin
37.         if(Rst)
38.             begin
39.                 Registro[0]<=0;
40.                 Registro[1]<=0;
41.                 Registro[2]<=0;
42.                 Registro[3]<=0;
43.                 Registro[4]<=0;
44.                 Registro[5]<=0;
45.                 Registro[6]<=0;
46.                 Registro[7]<=0;
47.             end
48.         else
49.             if (Sel_RLE)
50.                 Registro[Sel_R[2:0]]<=Dato_N;//se esta escribe lo que tenga Dato_N en
    Rx
51.
52.         end
53.
54.         assign Ry=Registro[Sel_R[5:3]];
55.         assign Rx=Registro[Sel_R[2:0]];
56. endmodule
```

Control_Sali

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 10.11.2020 22:52:42
7.// Design Name:
8.// Module Name: Control_Sali
9.// Project Name:
10.// Target Devices:
```

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module Control_Sali(
    input [7:0] Ry,
    input [7:0] Rx,
    input [7:0] Num,
    input Clk,
    input Rst,
    input [1:0] Sel_Sali,
    output reg [7:0] Address_Data_Bus,
    output reg [7:0] DataOut_Bus,
    output reg L_E
    );

    always @(posedge Clk, posedge Rst)
    begin
    if (Rst)
    begin
    Address_Data_Bus<=0;
    DataOut_Bus<=0;
    L_E<=0;
    end
    else
    case (Sel_Sali)
    2'b00: begin //Nop
     Address_Data_Bus<=0;
     DataOut_Bus<=0;
     L_E<=0; end
    2'b01: begin// Load [Ry], Rx
    Address_Data_Bus<=Ry;
     DataOut_Bus<=0;
      L_E<=0; //Lectura
      end
   2'b10: begin //Store [Rx],#Num
    Address_Data_Bus<=Rx;
     DataOut_Bus<=Num;
     L_E<=1; //Escritura
      end
    2'b11: begin //Store [Rx],Ry
```

```
            60.      Address_Data_Bus<=Rx;
            61.DataOut_Bus<=Ry;
           62.L_E<=1; //Escritura
            63.end
           64.default: begin
         65.Address_Data_Bus<=0;
        66.DataOut_Bus<=0;
       67.L_E<=0;//Lectura
        68.end
        69.endcase
    70.end
    71.always @(Rx,Ry,Num,Sel_Sali) begin
    72.if (Sel_Sali==2'b01)
    73.begin
    74.   DataOut_Bus<=0;
   75.Address_Data_Bus<=Ry;
 76.end
77.end
78.endmoodule
```

Memoria_RAM

```
1.`timescale 1ns / 1ps
2//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 11.11.2020 09:11:48
7.// Design Name:
8.// Module Name: Memoria_Ram
9// Project Name:
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revisión:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Memoria_RAM
24.#(parameter Ld=256, m=8)(
          25.input L_E,
          26.input [m-1:0] address_data,
```

```
27.input [m-1:0] data_in,
28.output reg [m-1:0] N_dataout
29.);

30.reg [(m-1):0]mem[0:(Ld-1)];
31.
32.initial
33.begin
34.$readmemh("RAM.mem",mem);
35.end
36.
37.always @(address_data,data_in,L_E) begin
38.if(L_E)
39.        begin
40.        mem[address_data]<=data_in;
41.        N_dataout<=mem[address_data];
42.end
43.else
44.N_dataout<=mem[address_data];
45.end
46.endmodule
```

Memoria_ROM

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 11.11.2020 09:11:48
7.// Design Name:
8.// Module Name: Memoria_ROM
9.// Project Name:
.10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17..// Revision 0.01 - File Created
18..// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
```

```
23.module Memoria_ROM
24.#(parameter Ld=256, m=8, n=9)(
        25.input [m-1:0] address,
        26.output reg [n-1:0] N_dout
        27);
28.reg [(n-1):0]mem[0:(Ld-1)];
 29.
30.initial
 31.begin
32 .$readmemb("ROM.mem",mem);
33.end
 34.
35.always @(address) begin
36.N_dout<=mem[address];
37.end
38.  endmodule
```

Multiplicación

001001100
001010010
001011001
001100100
101000101
110001000
101101000
101000010
110011001
101010000
111100011
101000101

División

001001100
001010010
001011001
001100100
101000101
110011000
101101000
101000001
110010001
101001000
111100111
101000101
110011001

Tb_MicroMem

```
1. `timescale 1ns / 1ps
2. //////////////////////////////////////////////////////////////////////////////////
3. // Company:
4. // Engineer:
5. //
6. // Create Date: 11.11.2020 10:06:47
7. // Design Name:
8. // Module Name: tb_MicroMem
9. // Project Name:
10. // Target Devices:
11. // Tool Versions:
12. // Description:
13. //
14. // Dependencies:
15. //
16. // Revision:
17. // Revision 0.01 - File Created
18. // Additional Comments:
19. //
20. //////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module tb_MicroMem;
24.
25.reg Clk;
  26. reg Rst;
   27.
  28. MicroMem uut(
    29. .Clk(Clk),
    30. .Rst(Rst)
    31. );
   32.
 33. initial
    34. begin
    35. Rst=1;
    36. Clk=0;
```

```
37.
38.  #2 Rst=0;
39.  #10 $finish;
40.  end
41.
42. always
43.   #1 Clk = !Clk;
44.endmodule
```

Tb_Microprocesador_8bits

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 10.11.2020 23:13:24
7.// Design Name:
8.// Module Name: Tb_Microprocesador_RISC_8bits
9.// Project Name:
10.// Target Devices:
11// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Tb_Microprocesador_RISC_8bits;
        24.reg Clk;
        25.reg Rst;
        26.reg  [7:0] DataIn_Bus;
        27.reg  [8:0] Instruction;
```

```
28.wire [7:0] Address_Data_Bus;
29.wire [7:0] DataOut_Bus;
30.wire L_E;
31.wire [7:0] Address_Instruction_Bus;



35.Microprocesador_RISC_8bits uut(
36..Clk(Clk),
37..Rst(Rst),
38..Instruction(Instruction),
39..DataIn_Bus(DataIn_Bus),
40..Address_Instruction_Bus(Address_Instruction_Bus),
41..DataOut_Bus(DataOut_Bus),
42..Address_Data_Bus(Address_Data_Bus),
43..L_E(L_E)
44.);

45.initial
46.begin
47.Rst=1;
48.Clk=0;
49.Instruction=0;
50.DataIn_Bus=0;
51.
52.#1 Rst=0; Instruction=9'b001_000_111; //load Rx Num
53.#2 Instruction=9'b001_001_110; //load Rx Num
54.#2 Instruction=9'b010_010_000; // load Rx [Ry]
55.DataIn_Bus=10;
56.#2 Instruction=9'b011_001_010; // store [Rx] Num
57.#2 Instruction=9'b100_010_000; // store [Rx] Ry
58.#2 Instruction=9'b101_011_000; // move Rx Ry
59.#2 Instruction=9'b110_011_001; // Math Rx OP
60.#2 Instruction=9'b111_100_001; // Jump [Rx] Cond
61.#2 Instruction=9'b000_100_001; // Nop
62.
63.end
64.
65.always
66.#1 Clk = !Clk;
67.
68.
69.endmodule
```

Tb_ALU

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
```

```
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 10.11.2020 19:34:30
7.// Design Name:
8.// Module Name: Tb_ALU
9.// Project Name:
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Tb_ALU;
24. reg[2:0]Ope;
25. reg [7:0]Ry;
26. reg [7:0]Rx;
27. wire [7:0] Resultado;
28. wire[2:0] Flags;
 29.
 30. ALU uut (
  31..Ope(Ope),
 32 .Ry(Ry),
 33. .Rx(Rx),
```

```
34.Resultado(Resultado),
35.  .Flags(Flags));

36.  initial
37.  begin
38.  Ry=8'b00000011;
39.  Rx=8'b00000100;
40.  Ope=0;
41.  #2 Ope=3'b000;
42.  #2 Ope=3'b001;
43.  #2 Ope=3'b100;
44.
45.  end
46.endmodule
```

Tb_Conca_Num

```
1.`timescale 1ns / 1ps
2.//////////////////////////////////////////////////////////////////////////////////
3.// Company:
4.// Engineer:
5.//
6.// Create Date: 10.11.2020 23:08:14
7.// Design Name:
8.// Module Name: Tb_Conca_Num
9.// Project Name:
10.// Target Devices:
11.// Tool Versions:
12.// Description:
13.//
14.// Dependencies:
15.//
16.// Revision:
17.// Revision 0.01 - File Created
18.// Additional Comments:
19.//
20.//////////////////////////////////////////////////////////////////////////////////
21.
22.
```

```
23.module Tb_Conca_Num;
24.
25.      reg [7:0] Instruction;
26.      wire [7:0] Num;
27.   Conca_Num uut(
 28.  .Instruction(Instruction),
 29.  .Num(Num)
30.   );
31.   initial
32.   begin
  33.
34.      Instruction = 0;
  35.
   36.
        37.#2Instruction = 8'b00000001;
  38.
  39.
40.#2Instruction = 8'b00001011;
 41.
 42 .
43.#2Instruction = 8'b11111111;
 44.  end
45.endmodule
```

Tb_Banco_R

```
1.  `timescale 1ns / 1ps
2.  //////////////////////////////////////////////////////////////////////////////////
3.  // Company: Universidad Autonoma de Zacatecas
4.  // Engineer: Roboticos
5.  //
6.  // Create Date: 18.10.2020 15:48:20
7.  // Design Name: Sel_Datos
8.  // Module Name: Sel_Datos
9.  // Project Name: Microprocesador_RISC_8bits
10. // Target Devices:
11. // Tool Versions:
12. // Description:
13. //
14. // Dependencies:
15. //
16. // Revision:
17. // Revision 0.01 - File Created
18. // Additional Comments:
19. //
```

```
20. //////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Tb_Banco_R;
 24.  reg Rst;
 25. reg Clk;
 26.  reg [5:0] Sel_R;
 27.  reg Sel_RLE;
 28. reg [7:0] Dato_N;
 29. wire [7:0] Rx;
 30.  wire [7:0] Ry;
  31.
 32.  Banco_R uut(
 33.  .Rst(Rst),
 34.  .Clk(Clk),
 35. .Sel_R(Sel_R),
 36.  .Sel_RLE(Sel_RLE),
 37.  .Dato_N(Dato_N),
 38.  .Rx(Rx),
 39.  .Ry(Ry));
  40.
 41.  initial
    42.  begin
       43.  Rst=1;
       44.  Clk=0;
       45.  Sel_R=0;
       46.  Sel_RLE=0;
       47.  Dato_N=0;
        48. #2 Rst=0; Dato_N=8'b00000000; Sel_R=6'b000_000; Sel_RLE=0;
       49.  #2 Dato_N=8'b00000001; Sel_R=6'b000000; Sel_RLE=1;
       50.  #2 Dato_N=8'b00000010; Sel_R=6'b000001; Sel_RLE=1;
```

```
51.  #2 Dato_N=8'b00000011; Sel_R=6'b000010; Sel_RLE=1;
52.  #2 Dato_N=8'b00000100; Sel_R=6'b000011; Sel_RLE=1;
53.  #2 Dato_N=8'b00000101; Sel_R=6'b000100; Sel_RLE=1;
54. #2 Dato_N=8'b00000110; Sel_R=6'b000101; Sel_RLE=1;
55. #2 Dato_N=8'b00000111; Sel_R=6'b000110; Sel_RLE=1;
56. #2 Dato_N=8'b00001000; Sel_R=6'b000111; Sel_RLE=1;
57.
58. #2 Sel_R=6'b001000; Sel_RLE=0;
59. #2 Sel_R=6'b011010; Sel_RLE=0;
60.  #2 Sel_R=6'b101100; Sel_RLE=0;
61.  #2 Sel_R=6'b111110; Sel_RLE=0;
62.
63. end
64.
65. always
66.#1 Clk = !Clk;
67.
68.endmodule
```

Tb_Control_Sali

```
1.  `timescale 1ns / 1ps
2.  //////////////////////////////////////////////////////////////////////////////////
3.  // Company:
4.  // Engineer:
5.  //
6.  // Create Date: 10.11.2020 22:54:27
7.  // Design Name:
8.  // Module Name: Tb_Control_Sali
9.  // Project Name:
10. // Target Devices:
11. // Tool Versions:
12. // Description:
13. //
14. // Dependencies:
15. //
16. // Revision:
```

```
17. // Revision 0.01 - File Created
18. // Additional Comments:
19. //
20. //////////////////////////////////////////////////////////////////////////////////
21.
22.
23.module Tb_Control_Sali;
    24.    reg [7:0] Ry;
    25.    reg [7:0] Rx;
    26.    reg [7:0] Num;
     27.   reg [1:0] Sel_Sali;
    28.    wire [7:0] Address_Data_Bus;
     29.   wire [7:0] DataOut_Bus;
     30.   wire L_E;
     31.   reg Clk;
      32.  reg Rst;
       33.
      34. Control_Sali uut(
      35.  .Ry(Ry),
      36.  .Rx(Rx),
       37. .Num(Num),
      38.  .Sel_Sali(Sel_Sali),
     39.   .Address_Data_Bus(Address_Data_Bus),
     40.   .DataOut_Bus(DataOut_Bus),
     41.   .L_E(L_E),
     42.   .Clk(Clk),
      43.  .Rst(Rst)
        44.             );
       45.
        46.  initial
          47. begin
         48.   Rst=1;
```

```
49. Ry=8'b11011010;
50.  Rx=8'b00110010;
51.  Num=8'b10010111;
52.  Sel_Sali=2'b01;
53.
54. #2
55.  Clk=1;
56.  Sel_Sali=2'b01;
57.
58. #2
59.  Sel_Sali=2'b11;
60.  Clk=1;
61.  #2
62. Sel_Sali=2'b10;
63. Clk=1;
64. #2
65. Sel_Sali=2'b00;
66. Clk=1;
67.  end
68.endmodule
```