Taller pre-entrega 2

IIC2343

Formalidades

º La entrega 1 es hoy a las **20 horas**.

º Feedback recibido de parte de los estudiantes

º Serán 4 entregas. El porcentaje de la E1 mantiene su porcentaje original.
Nota_proyecto: 0.15 * E1 + 0.25 * E2 + 0.25 * E3 + 0.35 E4 - 0.25 min(E2,E3)

º Se publicará los enunciados de la E2 y E3 la próxima semana, esto con el objetivo de que puedan organizar mejor sus tiempos. Ambas entregas tendrán un plazo de dos semanas, favor comenzar a trabajar lo más pronto posible.

Contenidos

º Una introducción a lo que será la próxima entrega

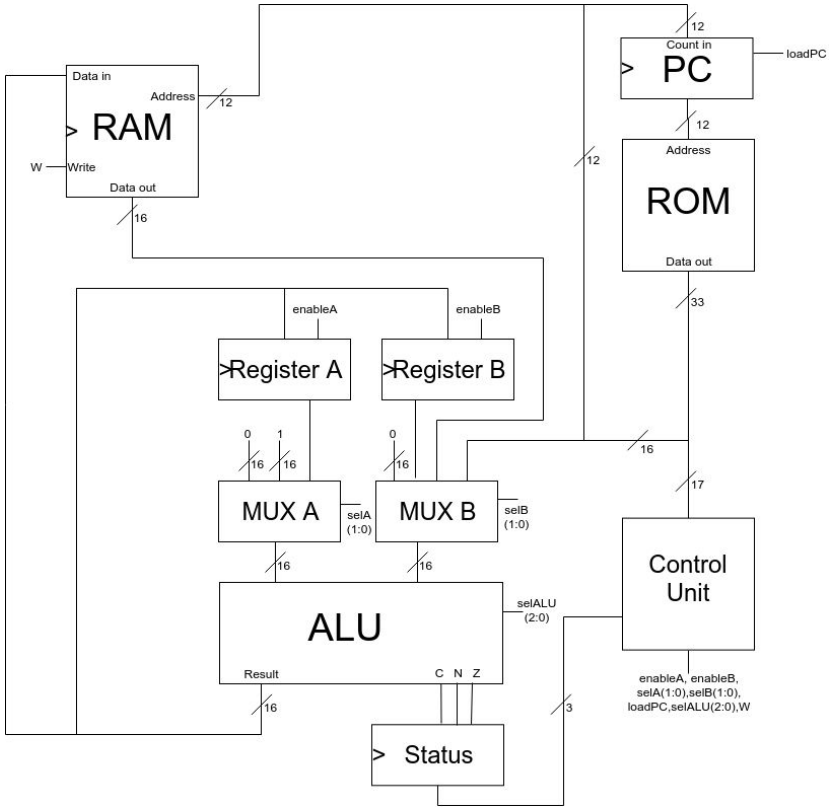
º Descripción paso a paso de la construcción de un computador

º Componentes adicionales: ROM

º Atender dudas entrega 1

Una introducción a lo que será la próxima entrega

º Para esta entrega tendrán que diseñar y armar su propio computador básico de 16 bits, el cual utilizarán posteriormente para ejecutar programas hechos en lenguaje assembly



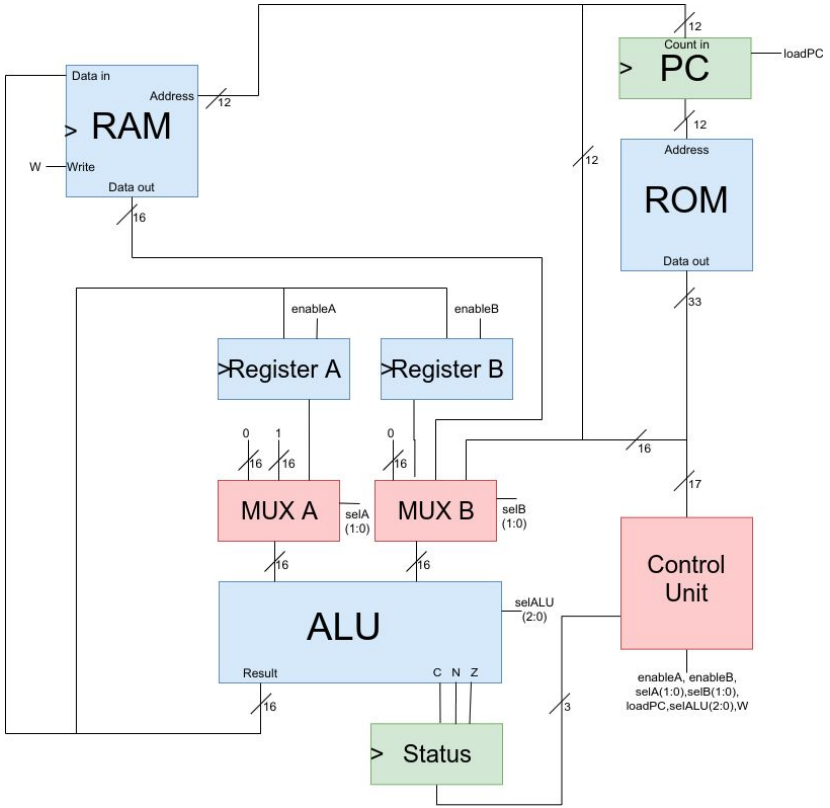
¡ No se asusten !

Una introducción a lo que será la próxima entrega

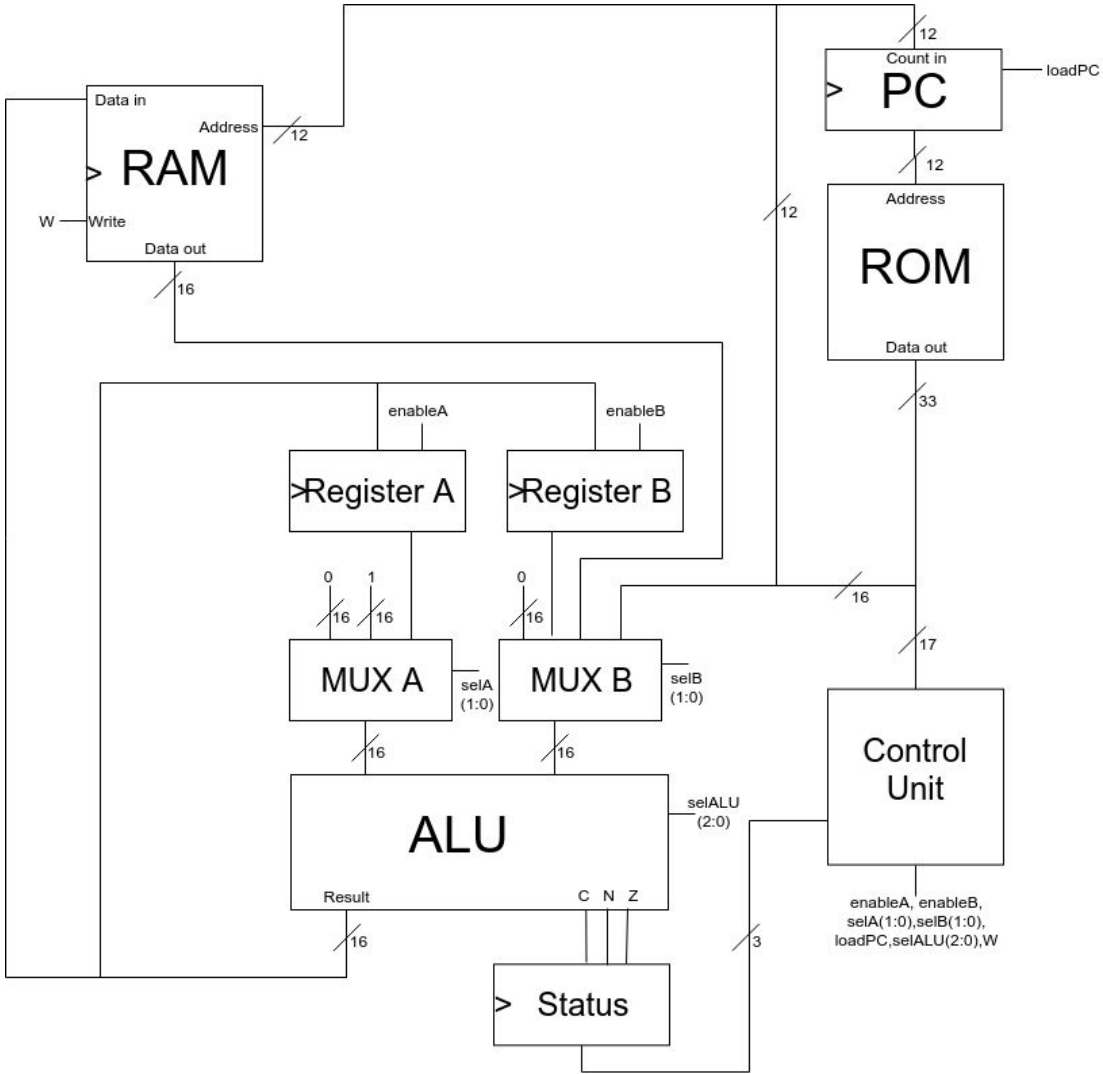
º Los componentes azules ya estarán a su disposición (a excepción de la ALU si es que no hicieron la entrega 1).

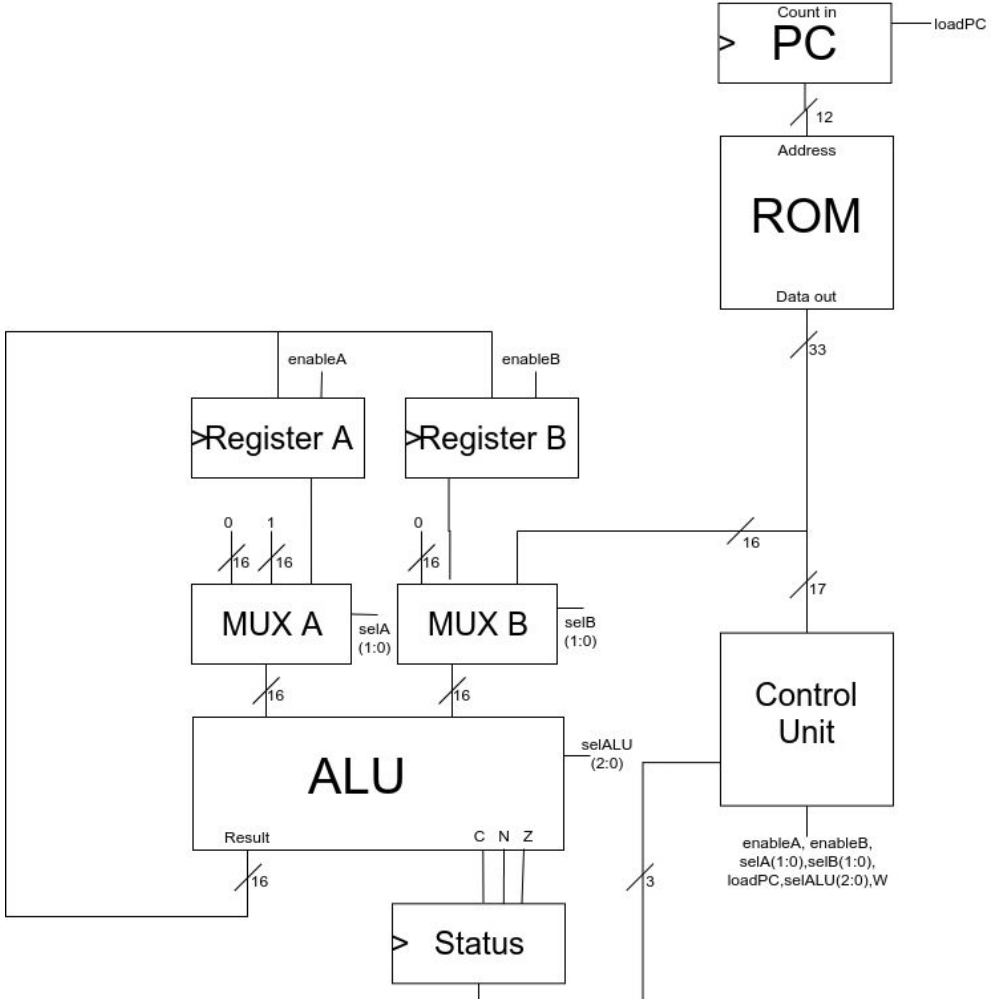
º Los componentes en verde son registros que pueden generar usando de base el componente Reg

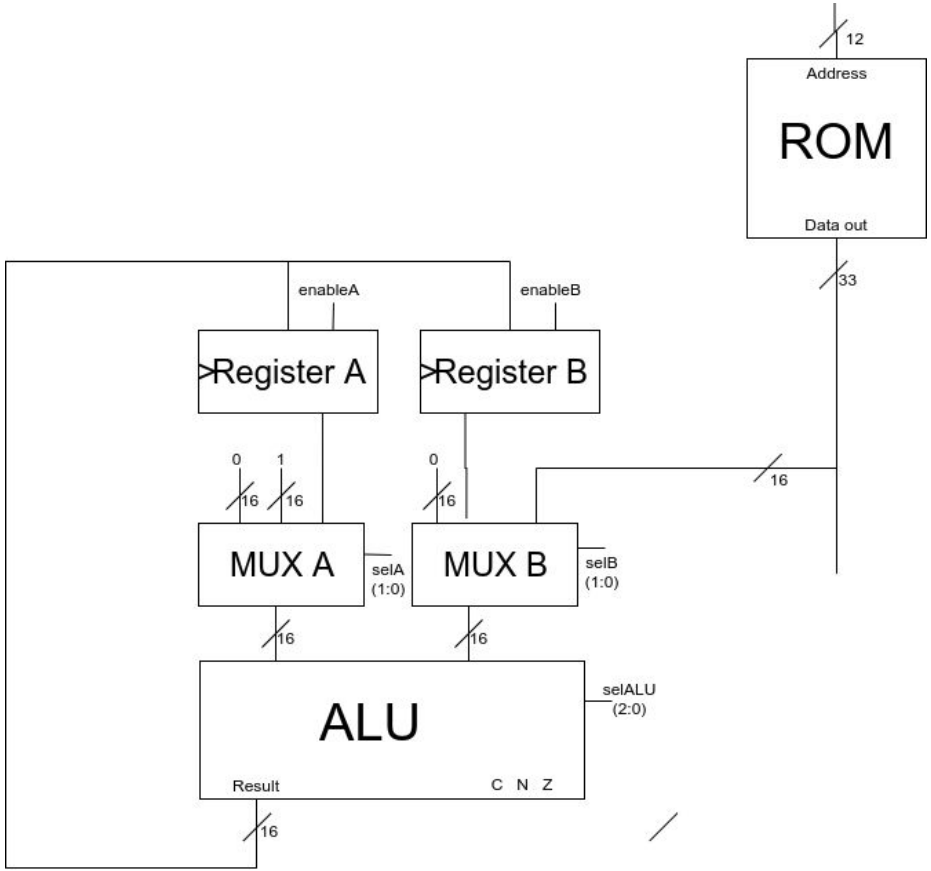
º El mayor desafío de esta entrega es que puedan diseñar su propia palabra de control, junto con su unidad de control



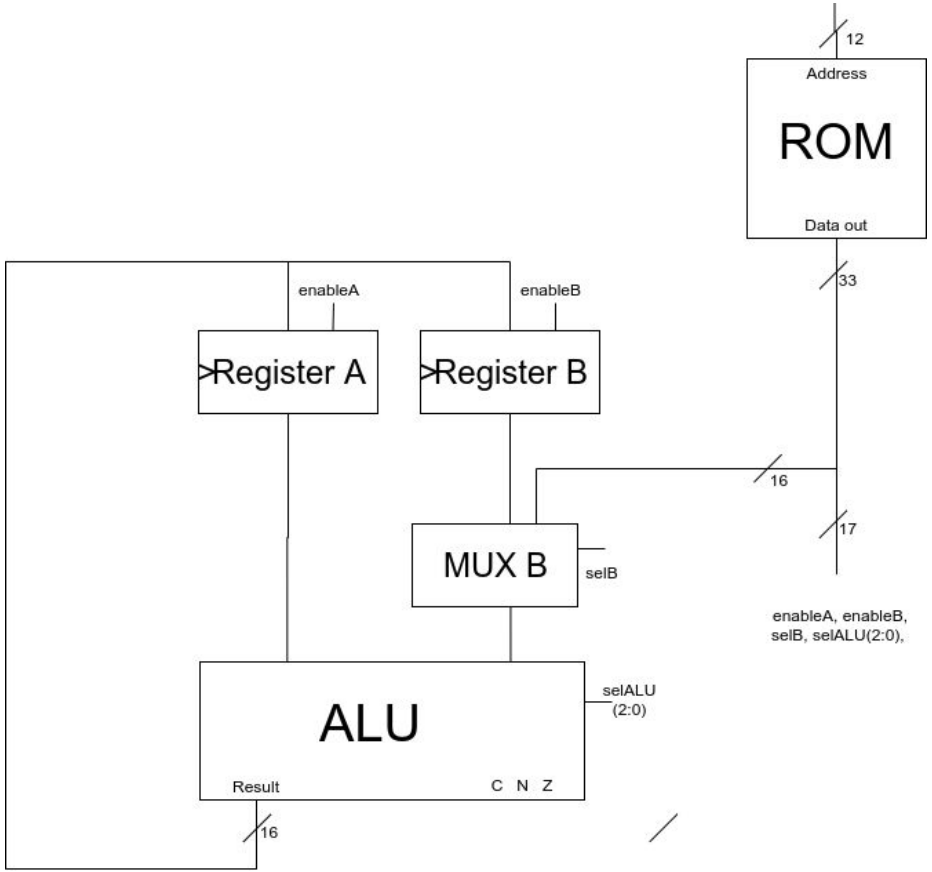
Retrocedamos un poco


12
Count in
PC
loadPC
Data in
Address
RAM
12
W
Write
Data out
16
12
12
Address
ROM
Data out
33
enableA
enableB
Register A
Register B
0
1
0
16
16
16
16
MUX A
selA
(1:0)
MUX B
selB
(1:0)
16
16
17
Control
Unit
ALU
selALU
(2:0)
Result
C N Z
16
3
Status
enableA, enableB,
selA(1:0),selB(1:0),
loadPC,selALU(2:0),W


Count in
PC
loadPC
12
Address
ROM
Data out
33
enableA
enableB
Register A
Register B
0
1
16
16
0
16
16
17
MUX A
selA
(1:0)
MUX B
selB
(1:0)
16
16
Control
Unit
ALU
selALU
(2:0)
Result
C N Z
16
enableA, enableB,
selA(1:0),selB(1:0),
loadPC,selALU(2:0),W
3
Status


12
Address
ROM
Data out
33
enableA
enableB
Register A
Register B
0
1
16
16
0
16
16
MUX A
selA
(1:0)
MUX B
selB
(1:0)
16
16
ALU
selALU
(2:0)
Result
C N Z
16

Tengamos esto como enfoque:



Componentes adicionales: ROM

```
 6  entity ROM is
 7      Port (
 8          address   : in  std_logic_vector(11 downto 0);
 9          dataout   : out std_logic_vector(32 downto 0)
10            );
11  end ROM;
12
13  architecture Behavioral of ROM is
14
15  type memory_array is array (0 to ((2 ** 12) - 1) ) of std_logic_vector
16
17  signal memory : memory_array:= (
18      "00000000000000000000000000000000",
```

Componentes adicionales: ROM - Ejemplo

```
signal memory : memory_array:= (
  -- LITERAL      1              ALUmba
  "00000000000000000000000000000000",
  "00000000000011000000000000000001", -- MOV A,C
  "00000000000000010000000000000010", -- ADD B,A
  "00000000000000000000000000000101", -- ADD A,B
  "00000000000000000000000000111010", -- SHL B
```

1º A es igual a C (12 en decimal)

2º Luego B se le suma A + 1 (sería D)

3º Luego A pasa a ser A+B (C+D= 19, o 25 en decimal)

4º A se le hace SHL, y se guarda en B (32 en hex, 50 en decimal)

Volviendo a la entrega:

º Subido esta la RAM y ROM para que si lo desean puedan comenzar a trabajar en el desarrollo del circuito.

º Pueden pensar en cómo podría ser el diseño de su palabra de control y como será su componente Control Unit

º No olvidar usar las issues para cualquier consulta

Dudas