



Ayudantía I1

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

Andrés González

Sebastián Ramos



01

02

03

04

05

06



Temas

- Lógica Digital
- Representaciones numéricas
- Hardware sin manejo de memoria
- Hardware con manejo de memoria

01

02

03

04

05

06





01

02

03

04

05

06



Pregunta 1

Indique la base β en la cual la siguiente ecuación es correcta:

$$7_{\beta} + 8_{\beta} = 13_{\beta}$$

Representación posicional

$$\sum_{k=0}^{n-1} s_k \times b^k$$





Pregunta 1

$$13_{\beta} = \beta^1 * 1 + \beta^0 * 3 = \beta + 3$$

$$\beta^0 * 7 + \beta^0 * 8 = 7 + 8 = 15$$

entonces, $7_{\beta} + 8_{\beta} = 13_{\beta}$:

$$\beta + 3 = 15,$$

$$\beta = 12$$

01

02

03

04

05

06



01

02

03

04

05

06





01

02

03

04

05

06



Pregunta 2

Luego del vergonzoso planteamiento de Martín Lasarte ante Ecuador y Colombia, la selección chilena se enfrenta a una nueva fecha triple de la cual depende su clasificación al mundial y la continuidad de “Machete”. Se pueden presentar las siguientes situaciones:

I) Si Chile le gana a Perú y Venezuela, entonces clasifica al mundial.

II) Si Chile le gana a Paraguay, Martín Lasarte se queda, en otro caso se va.

III) Si Chile pierde con Perú o Venezuela, no clasifica al mundial.

Sea H para ganarle a Perú, S para ganarle a Paraguay, Q para ganarle a Venezuela, C para clasificar y L para se queda “Machete”:

- a) Cree la tabla de verdad correspondiente
- b) Cree la fórmula lógica para las salidas C y L
- c) A partir de la fórmula lógica crea el circuito lógico





Pregunta 2

a)

I) Si Chile le gana a Perú y Venezuela, entonces clasifica al mundial.

H: Gana a Perú

S: Gana a Paraguay

Q: Gana a Venezuela

C: Clasifica

L: Lasarte se queda

H	S	Q	C	L
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1	1	
1	1	0		
1	1	1	1	

01

02

03

04

05

06



01

02

03

04

05

06





01

02

03

04

05

06



01

Pregunta 2

a)

II) Si Chile le gana a Paraguay, Martín Lasarte se queda, en otro caso se va.

H: Gana a Perú

S: Gana a Paraguay

Q: Gana a Venezuela

C: Clasifica

L: Lasarte se queda

H	S	Q	C	L
0	0	0		0
0	0	1		0
0	1	0		1
0	1	1		1
1	0	0		0
1	0	1	1	0
1	1	0		1
1	1	1	1	1

02

03

04

05

06





Pregunta 2

a)

III) Si Chile pierde con Perú o Venezuela, no clasifica al mundial.

H: Gana a Perú

S: Gana a Paraguay

Q: Gana a Venezuela

C: Clasifica

L: Lasarte se queda

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1



Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$C = (H \wedge \neg S \wedge Q) \vee (H \wedge S \wedge Q)$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$C = (H \wedge \neg S \wedge Q) \vee (H \wedge S \wedge Q)$$

Reordenando,

$$C = (H \wedge Q \wedge \neg S) \vee (H \wedge Q \wedge S)$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$C = (H \wedge \neg S \wedge Q) \vee (H \wedge S \wedge Q)$$

Reordenando,

$$C = (H \wedge Q \wedge \neg S) \vee (H \wedge Q \wedge S)$$

$$\text{Sea } M = H \wedge Q$$

$$C = (M \wedge \neg S) \vee (M \wedge S) = M$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$C = H \wedge Q$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L = (\neg H \wedge S \wedge \neg Q) \vee (\neg H \wedge S \wedge Q) \vee (H \wedge S \wedge \neg Q) \vee (H \wedge S \wedge Q)$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L = (\neg H \wedge S \wedge \neg Q) \vee (\neg H \wedge S \wedge Q) \vee (H \wedge S \wedge \neg Q) \vee (H \wedge S \wedge Q)$$

Sea,

$$L1 = (\neg H \wedge S \wedge \neg Q) \vee (\neg H \wedge S \wedge Q)$$

$$L2 = (H \wedge S \wedge \neg Q) \vee (H \wedge S \wedge Q)$$

Tal que $L = L1 \vee L2$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L1 = (\neg H \wedge S \wedge \neg Q) \vee (\neg H \wedge S \wedge Q)$$

$$\text{Sea } M = \neg H \wedge S$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L1 = (\neg H \wedge S \wedge \neg Q) \vee (\neg H \wedge S \wedge Q)$$

$$\text{Sea } M = \neg H \wedge S$$

Tenemos

$$L1 = (M \wedge \neg Q) \vee (M \wedge Q) = \neg H \wedge S$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L2 = (H \wedge S \wedge \neg Q) \vee (H \wedge S \wedge Q)$$

$$\text{Sea } M = H \wedge S$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L2 = (H \wedge S \wedge \neg Q) \vee (H \wedge S \wedge Q)$$

$$\text{Sea } M = H \wedge S$$

Tenemos

$$L2 = (M \wedge \neg Q) \vee (M \wedge Q) = H \wedge S$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

Sabemos que $L = L1 \vee L2$

y que,

$$L1 = \neg H \wedge S = S \wedge \neg H$$

$$L2 = H \wedge S = S \wedge H$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

Sabemos que $L = L1 \vee L2$

y que,

$$L1 = \neg H \wedge S = S \wedge \neg H$$

$$L2 = H \wedge S = S \wedge H$$

Entonces,

$$L = (S \wedge \neg H) \vee (S \wedge H)$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

$$L = (S \wedge \neg H) \vee (S \wedge H)$$

Por el axioma,

$$L = S$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

b) Cree la fórmula lógica para las salidas C y L

De esa forma,

$$C = H \wedge Q$$

$$L = S$$

$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

Pregunta 2

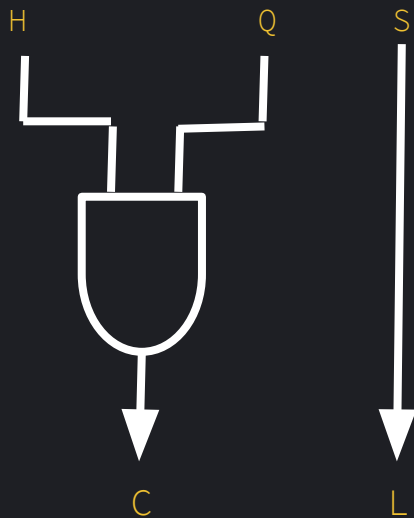
$$(A \wedge \neg B) \vee (A \wedge B) \equiv A$$

c) A partir de la fórmula lógica crea el circuito lógico

$$C = H \wedge Q$$

$$L = S$$

H	S	Q	C	L
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1





Pregunta 3 (I1-2016-1)

Diseñe un circuito que multiplique por 4 un número natural de 8 bits. Asuma que no ocurrirá overflow.



01

02

03

04

05

06



01

02

03

04

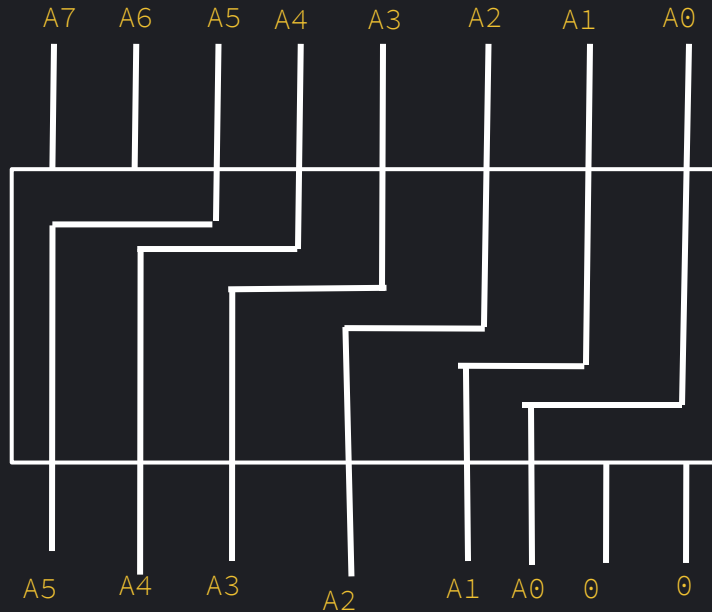
05

06



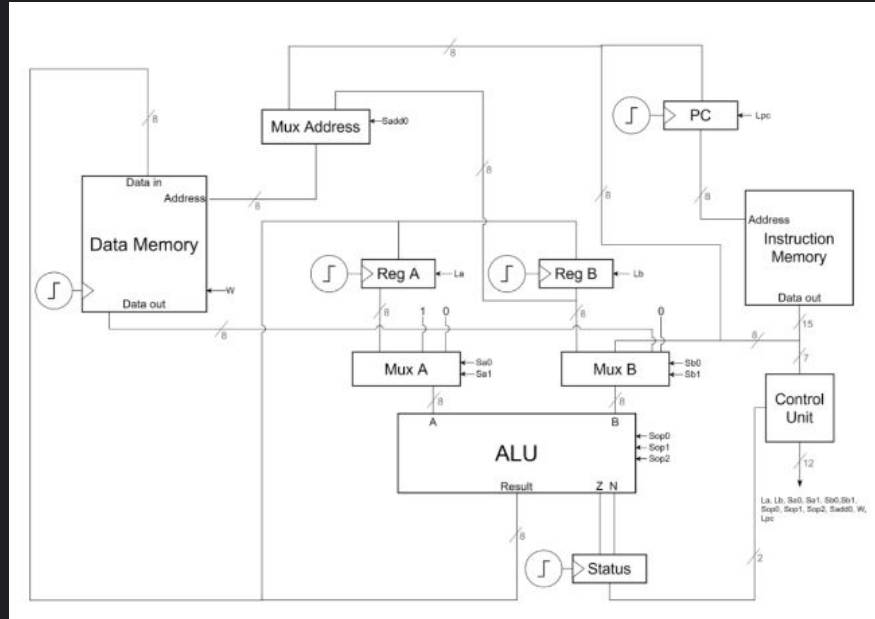
Pregunta 3 (I1-2016-1)

Diseñe un circuito que multiplique por 4 un número natural de 8 bits. Asuma que no ocurrirá overflow.



Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.





01

02

03

04

05

06



Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

Agregar **registros extendidos** de 16 bits independientes a una **ALU extendida** que pueda recibir a 4 registros (2 de 8 bits y 2 de 16 bits).

01

02

03

04

05

06

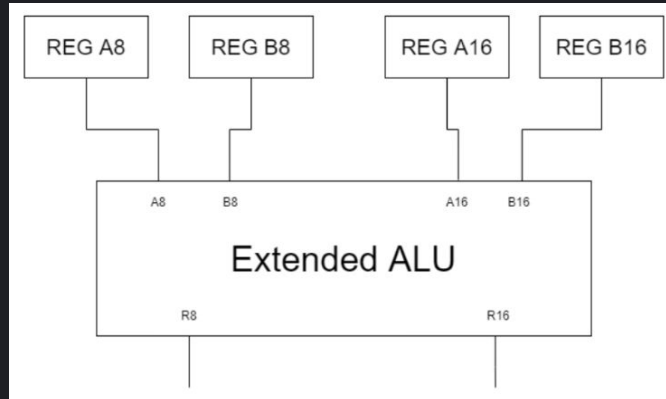


Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

Agregar **registros extendidos** de 16 bits independientes a una **ALU extendida** que pueda recibir a 4 registros (2 de 8 bits y 2 de 16 bits).





01

02

03

04

05

06



Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

Además se debe modificar el **almacenamiento en memoria** para que este pueda soportar 16 bits. Esto se puede hacer de muchas maneras distintas. Una puede ser añadiendo un multiplexor para que la memoria sepa cuando deba manejar 8 bits o 16 bits, o bien se pueden definir 2 ciclos, para que se lea/escriba una palabra en cada ciclo y así completar los 16 bits.

01

02

03

04

05

06

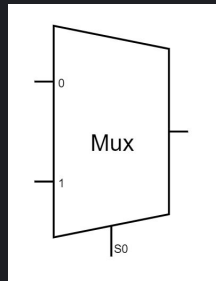


Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

Además se debe modificar el **almacenamiento en memoria** para que este pueda soportar 16 bits. Esto se puede hacer de muchas maneras distintas. Una puede ser añadiendo un multiplexor para que la memoria sepa cuando deba manejar 8 bits o 16 bits, o bien se pueden definir 2 ciclos, para que se lea/escriba una palabra en cada ciclo y así completar los 16 bits.



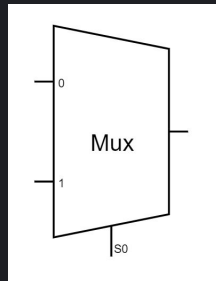
Si no hay un multiplexor, que puede pasar con respecto a los 8 bits?

Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

Además se debe modificar el **almacenamiento en memoria** para que este pueda soportar 16 bits. Esto se puede hacer de muchas maneras distintas. Una puede ser añadiendo un multiplexor para que la memoria sepa cuando deba manejar 8 bits o 16 bits, o bien se pueden definir 2 ciclos, para que se lea/escriba una palabra en cada ciclo y así completar los 16 bits.



Si no hay un multiplexor, que puede pasar con respecto a los 8 bits?

Los carry pueden irse al mundo de 16 bits 🤪



01

02

03

04

05

06



Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

No debemos olvidar que también se debe modificar la **ISA** para esta arquitectura

ADD8, ADD16, SUB8, SUB16, ...

Los **opcodes** correspondientes también deben ser creados.

01

02

03

04

05

06



Pregunta 4

Instrucción	Operandos	Opcode	La	Lb	Sa0	Sb0	Sb1	Sop2	Sop1	Sop0	Operación
MOV	A,B	000000	1	0	1	0	0	0	0	0	A=B
	B,A	000001	0	1	0	1	1	0	0	0	B=A
	A,Lit	000010	1	0	1	0	1	0	0	0	A=Lit
	B,Lit	000011	0	1	1	0	1	0	0	0	B=Lit
ADD	A,B	000100	1	0	0	0	0	0	0	0	A=A+B
	B,A	000101	0	1	0	0	0	0	0	0	B=A+B
	A,Lit	000110	1	0	0	0	1	0	0	0	A=A+Lit
SUB	A,B	000111	1	0	0	0	0	0	0	1	A=A-B
	B,A	001000	0	1	0	0	0	0	0	1	B=A-B
	A,Lit	001001	1	0	0	0	1	0	0	1	A=A-Lit
AND	A,B	001010	1	0	0	0	0	0	1	0	A=A and B
	B,A	001011	0	1	0	0	0	0	1	0	B=A and B
	A,Lit	001100	1	0	0	0	1	0	1	0	A=A and Lit
OR	A,B	001101	1	0	0	0	0	0	1	1	A=A or B
	B,A	001110	0	1	0	0	0	0	1	1	B=A or B
	A,Lit	001111	1	0	0	0	1	0	1	1	A=A or Lit
NOT	A,A	010000	1	0	0	0	0	1	0	0	A=notA
	B,A	010001	0	1	0	0	0	1	0	0	B=notA
XOR	A,A	010010	1	0	0	0	0	1	0	1	A=A xor B
	B,A	010011	0	1	0	0	0	1	0	1	B=A xor B
	A,Lit	010100	1	0	0	0	1	1	0	1	A=A xor Lit
SHL	A,A	010101	1	0	0	0	0	1	1	0	A=shift left A
	B,A	010110	0	1	0	0	0	1	1	0	B=shift left A
SHR	A,A	010111	1	0	0	0	0	1	1	1	A=shift right A
	B,A	011000	0	1	0	0	0	1	1	1	B=shift right A



01

02

03

04

05

06



Pregunta 4

Transforme la arquitectura del computador básico para que esta soporte como tipos de datos nativos números enteros de 8 y 16 bits de manera independiente.

Solución:

Los **opcodes** correspondientes también deben ser creados. Se re-aprovechan los preexistentes para 8 bit, luego se crean nuevos opcodes para ADD16, SUB16, AND16, etc.

El opcode más alto ocupado en el assembly básico es el $011000_2 = 24_{10}$ y el máximo es $111111_2 = 63_{10}$, por lo que se tienen suficientes opcodes desocupados para agregar este duplicado.

Además, es necesario agregar las señales de control apropiadas para manejar 8 o 16 bits, dependiendo del opcode en uso.

01

02

03

04

05

06





01

02

03

04

05

06



01

Pregunta 5

Considera la siguiente versión del computador básico y su correspondiente set de instrucciones (página 3). Queremos agregar un tercer registro, *C*, al computador básico, y las correspondientes instrucciones **ADD A,C** y **MOV C,B**.

- Dibuja y explica de manera clara y precisa los cambios necesarios en el circuito del computador básico: destaca claramente componentes, cables, buses de datos nuevos/modificados y cualquier otro cambio que hagas. La ALU no cambia en cuanto a que sigue teniendo dos inputs, *A* y *B*, y un output, *Result*, y que todos son de 8 bits.
- Especifica las nuevas instrucciones en cuanto a opcodes, señales de control, y operación, similarmente a las otras instrucciones en el set y de manera coherente con tu dibujo en a).





01

02

03

04

05

06



Pregunta 5

Transforme la arquitectura del computador básico para que esta soporte un nuevo registro C y las instrucciones ADD A,C y MOV C,B sin modificar la ALU.

Solución:

Se debe modificar lo que entra a la ALU, con tal de que esta pueda realizar las operaciones.

01

02

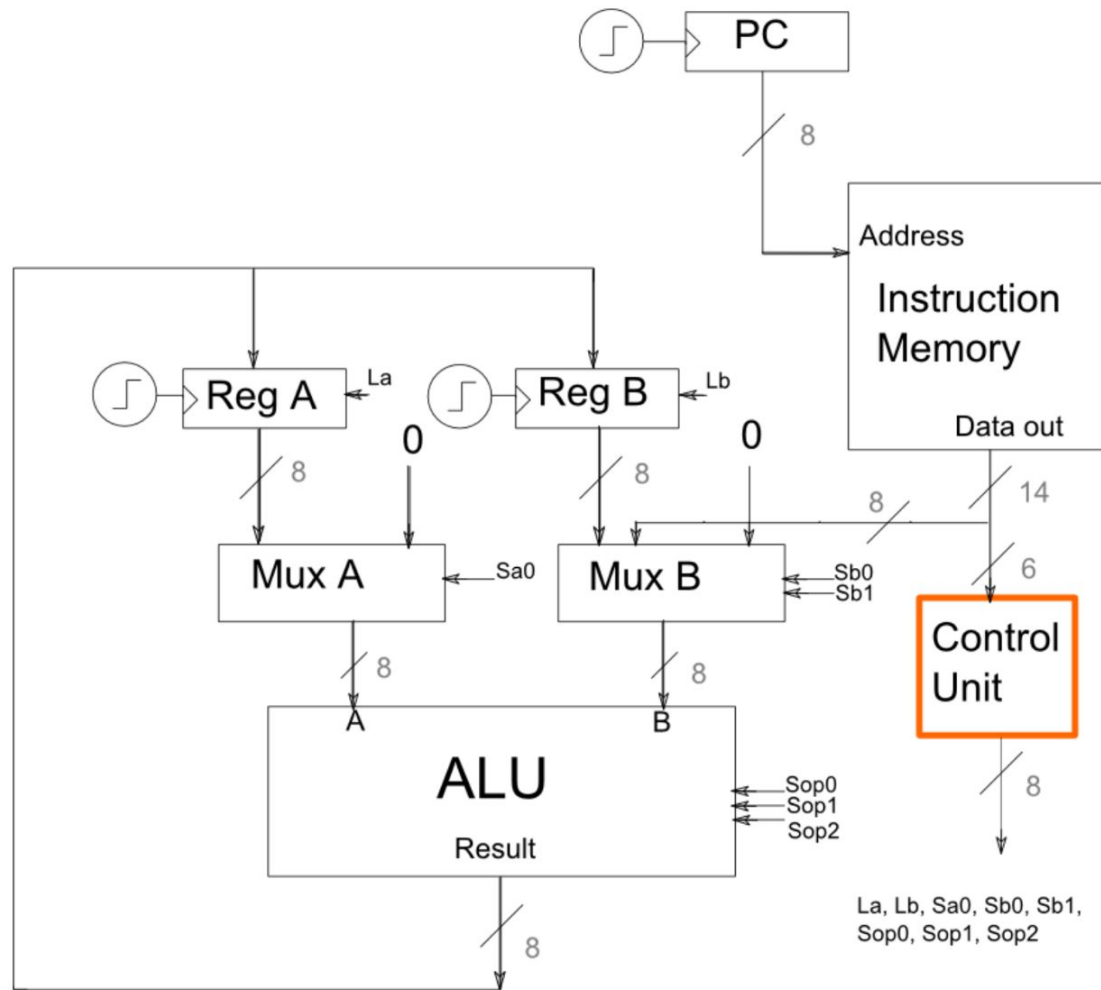
03

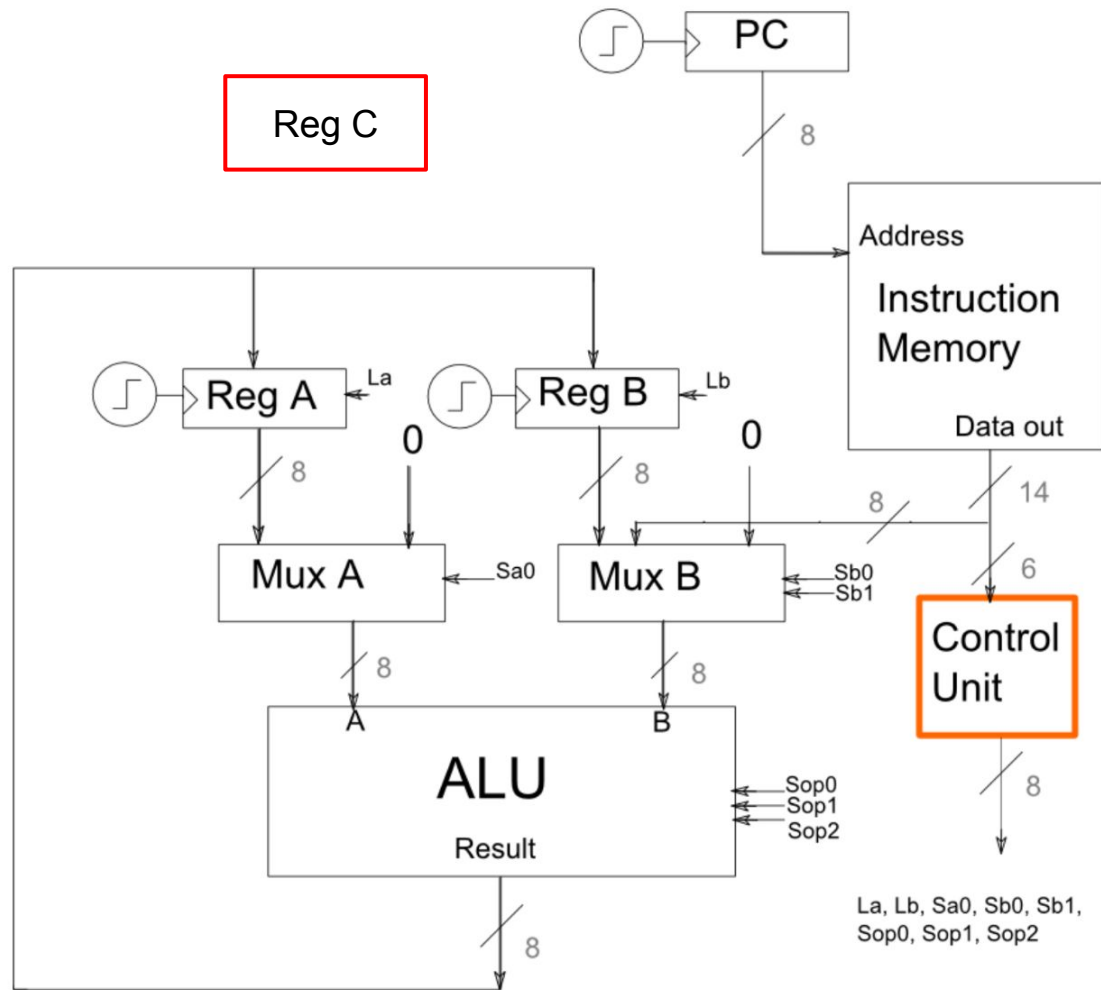
04

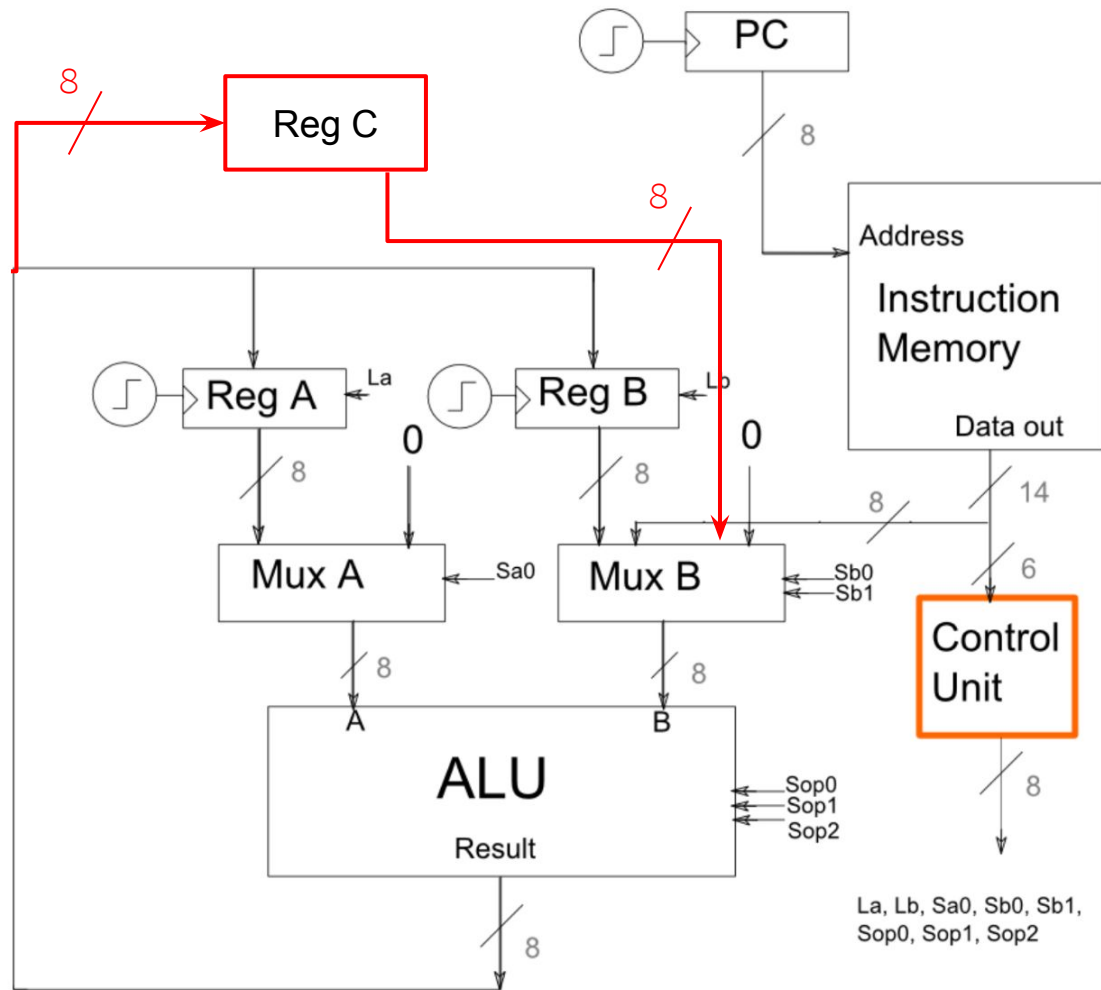
05

06



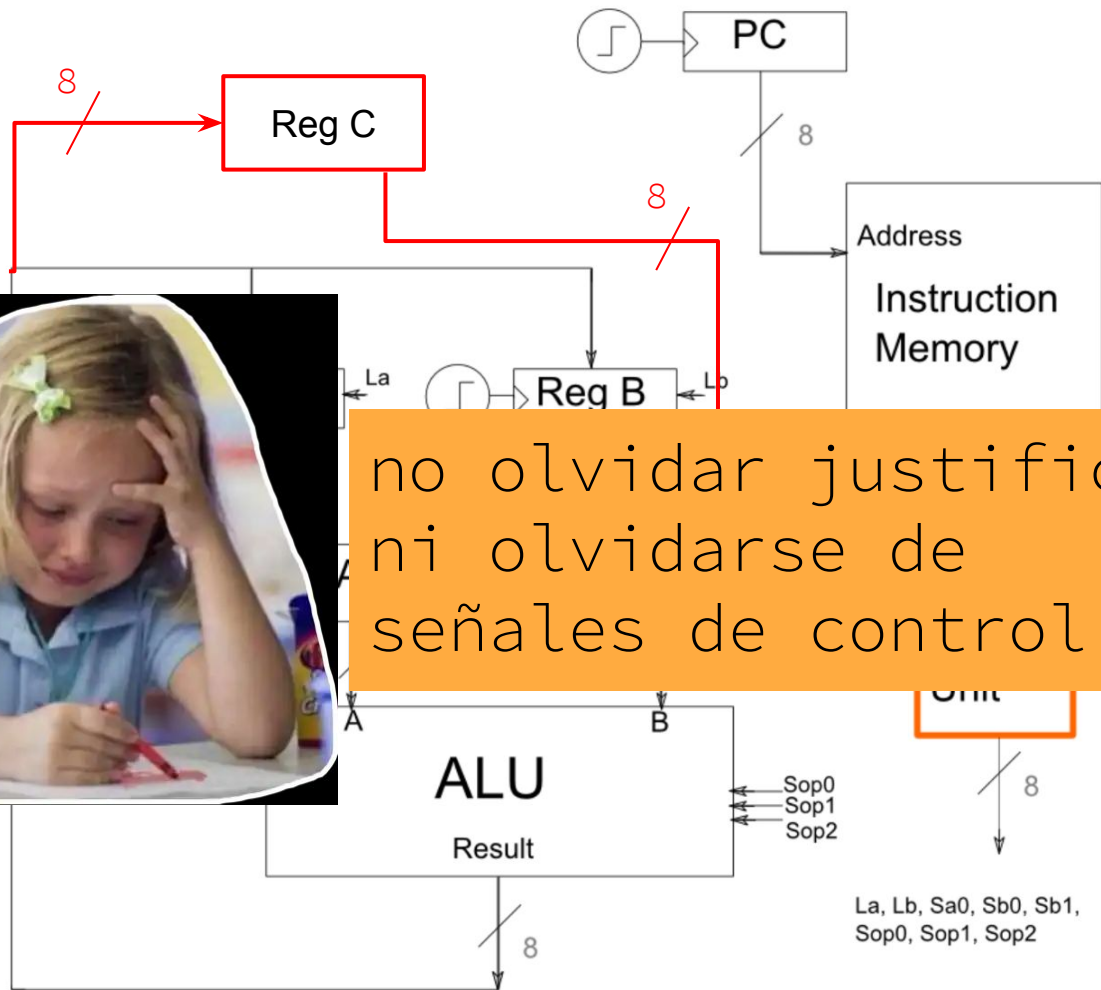


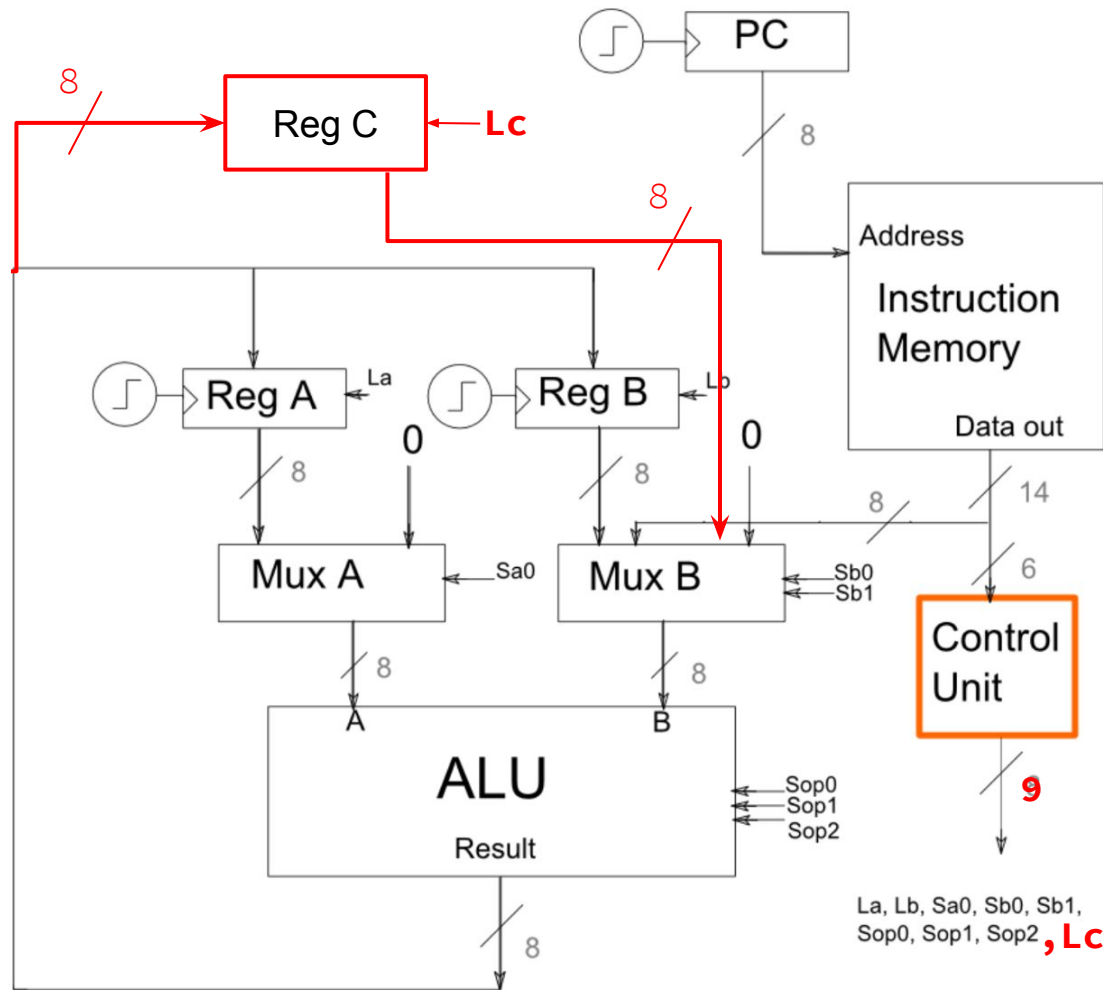






no olvidar justificar
ni olvidarse de
señales de control!!





La, Lb, Sa0, Sb0, Sb1,
Sop0, Sop1, Sop2, **Lc**



01

02

03

04

05

06



Pregunta 5

Transforme la arquitectura del computador básico para que esta soporte un nuevo registro C y las instrucciones ADD A,C y MOV C,B sin modificar la ALU.

Solución:

Primero, se debe conectar el registro al output de la ALU para que pueda almacenar el resultado de esta. También se debe colocar la señal de control Lc con tal de indicar cuando el output va a parar al registro C.

En cualquier momento que agreguemos una señal de control, esta debe verse reflejada en el Control Unit.

Luego, se debe conectar el registro al MUX B con tal que el valor de C pueda entrar a la ALU por el Input B. Esto se puede conectar así nomás porque el MUX B ya tiene una combinación abierta de señales de control para aceptar otro input.





01

02

03

04

05

06



Pregunta 5

Transforme la arquitectura del computador básico para que esta soporte un nuevo registro C y las instrucciones ADD A,C y MOV C,B sin modificar la ALU.

Solución:

Por que solamente nos importa que entre al Input B y no al Input A?

Esto se debe a que la instrucción ADD A,C requiere que el registro C entre por el Input B a la ALU (en el Input A va a ir lo del registro A), mientras que la instrucción MOV C,B necesita que el Input A sea cero y el Input B tenga el valor del registro B.





01

02

03

04

05

06



Pregunta 5

Transforme la arquitectura del computador básico para que esta soporte un nuevo registro C y las instrucciones ADD A,C y MOV C,B sin modificar la ALU.

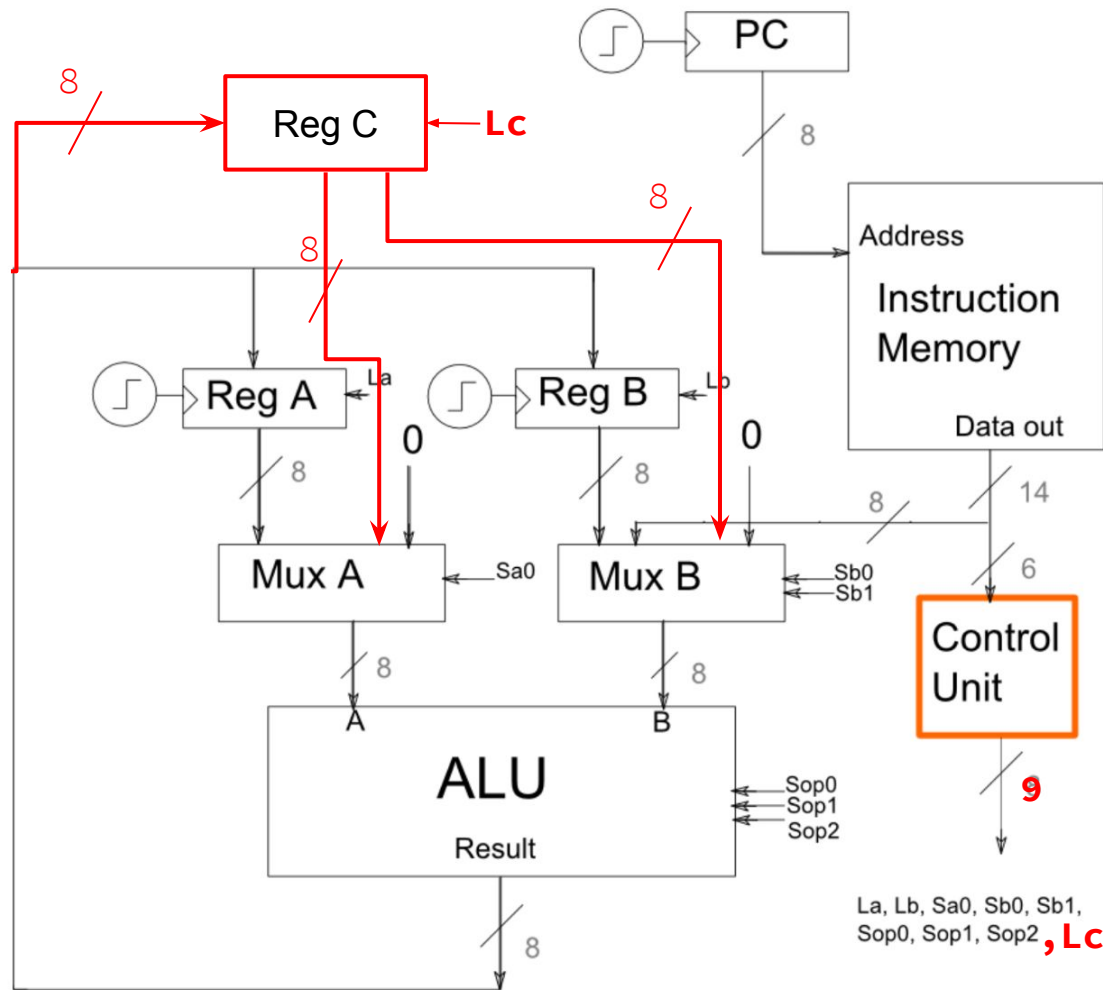
Solución:

Por que solamente nos importa que entre al Input B y no al Input A?

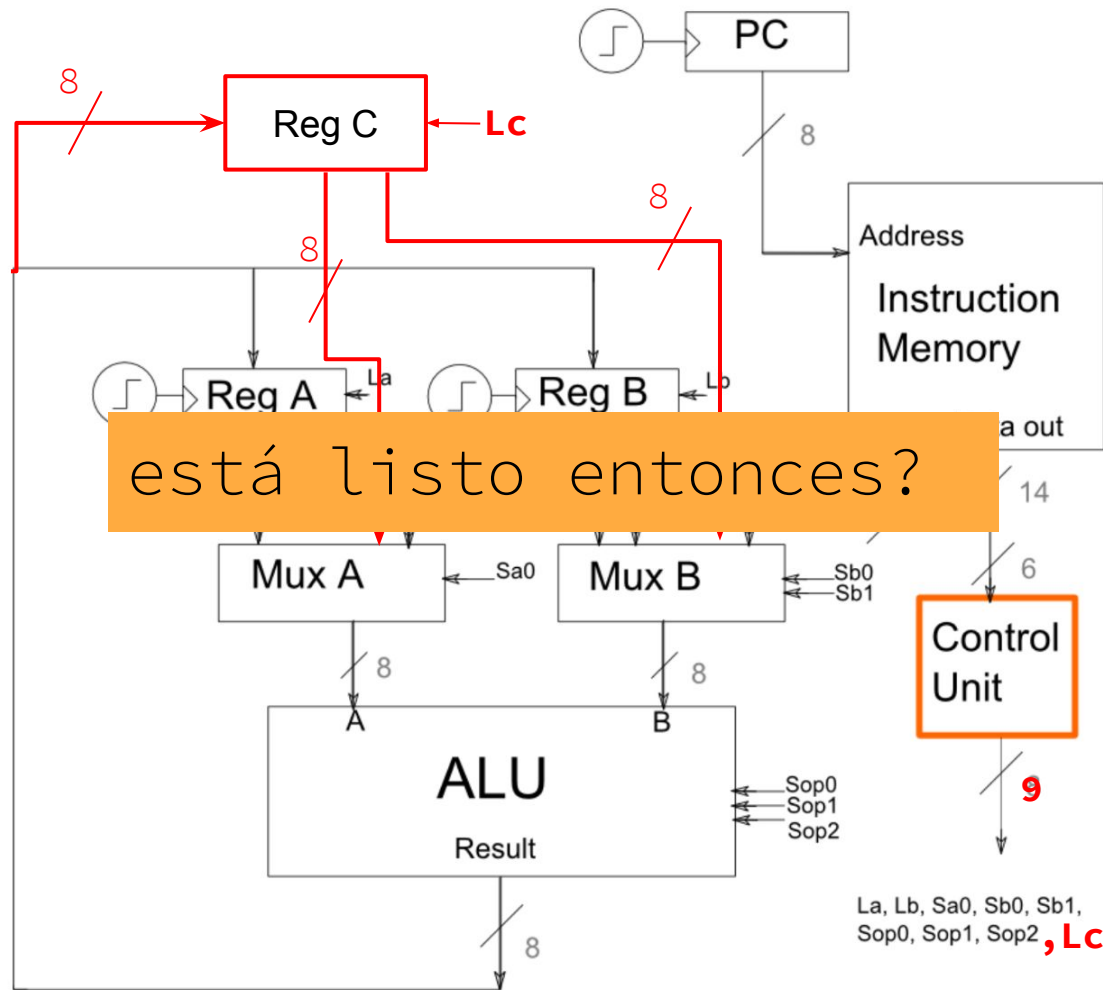
Esto se debe a que la instrucción ADD A,C requiere que el registro C entre por el Input B a la ALU (en el Input A va a ir lo del registro A), mientras que la instrucción MOV C,B necesita que el Input A sea cero y el Input B tenga el valor del registro B.

Que debiéramos hacer para poder realizar la instrucción ADD C,B?

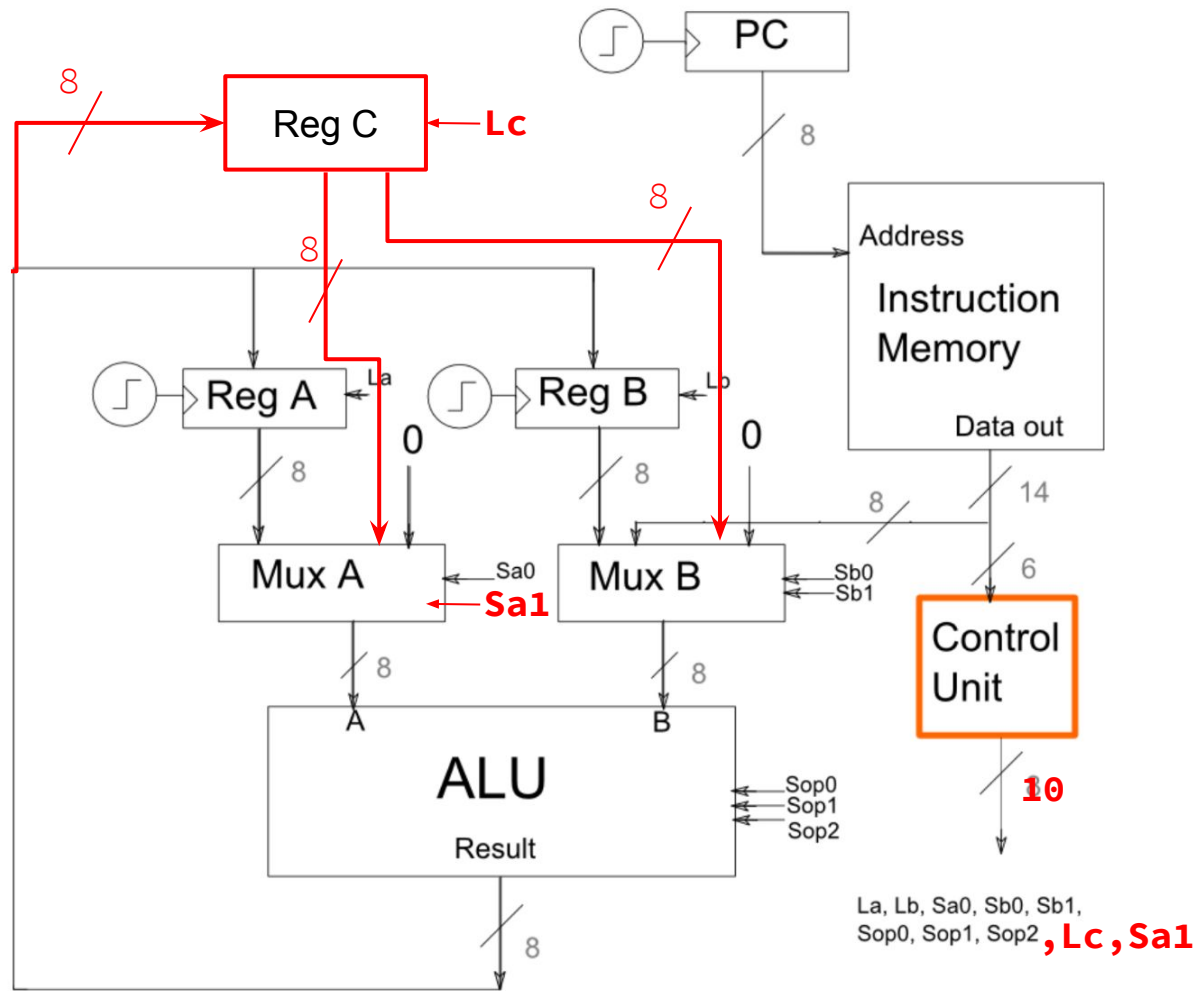




La, Lb, Sa0, Sb0, Sb1,
Sop0, Sop1, Sop2, **Lc**



está listo entonces?





Pregunta 5

Transforme la arquitectura del computador básico para que esta soporte un nuevo registro C y las instrucciones ADD A,C y MOV C,B sin modificar la ALU.

Solución:

	Op	La	Lb	Lc	Sa0	Sb0	Sb1	Sop2, Sop1, Sop0
ADD A,C	011100	1	0	0	0	1	0	0
MOV C,B	011101	0	0	1	1	0	0	0

Pregunta 5

Transforme la arquitectura del computador básico para que esta soporte un nuevo registro C y las instrucciones ADD A,C y MOV C,B sin modificar la ALU.

Solución:

	Op	La	Lb	Lc	Sa0	Sa1	Sb0	Sb1	Sop2, Sop1, Sop0
ADD A,C	011100	1	0	0	0	0	1	0	0
MOV C,B	011101	0	0	1	1	0	0	0	0
ADD C,B	011110	0	0	1	0	1	0	0	0



GL; HF!

$q(\geq \nabla \leq q)$