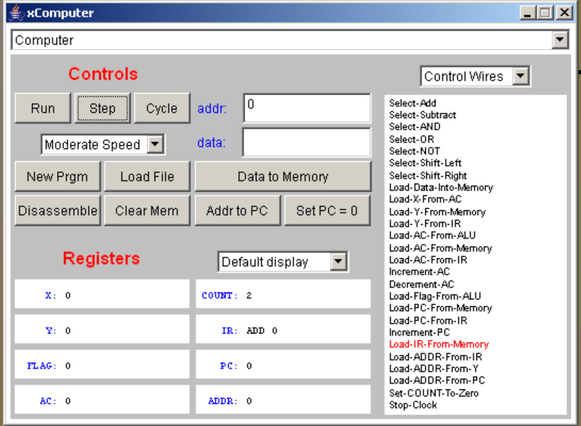
7- Estudio de un S.E.T.I. básico II (Hardware)

BIBLIOGRAFÍA

[Eck 95]

David J. Eck,
"The Most Complex Machine",
A. K. Peters.

german.bordel@ehu.es
http://gtts.ehu.es/German



xComputer

Computer

Controls

Run | Step | Cycle | addr: 0

Moderate Speed | data:

New Prgm | Load File | Data to Memory

Disassemble | Clear Mem | Addr to PC | Set PC = 0

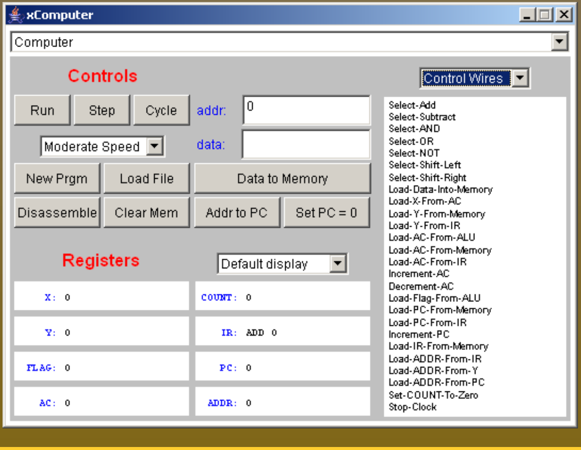
Control Wires

```
Select-Add
Select-Subtract
Select-AND
Select-OR
Select-NOT
Select-Shift-Left
Select-Shift-Right
Load-Data-Into-Memory
Load-X-From-AC
Load-Y-From-Memory
Load-Y-From-IR
Load-AC-From-ALU
Load-AC-From-Memory
Load-AC-From-IR
Increment-AC
Decrement-AC
Load-Flag-From-ALU
Load-PC-From-Memory
Load-PC-From-IR
Increment-PC
Load-IR-From-Memory
Load-ADDR-From-IR
Load-ADDR-From-Y
Load-ADDR-From-PC
Set-COUNT-To-Zero
Stop-Clock
```

Registers

Default display

```
X: 0          COUNT: 2
Y: 0          IR: ADD 0
FLAG: 0       PC: 0
AC: 0         ADDR: 0
```



xComputer

Computer

Controls

Run | Step | Cycle | addr: 0

Moderate Speed | data:

New Prgm | Load File | Data to Memory

Disassemble | Clear Mem | Addr to PC | Set PC = 0

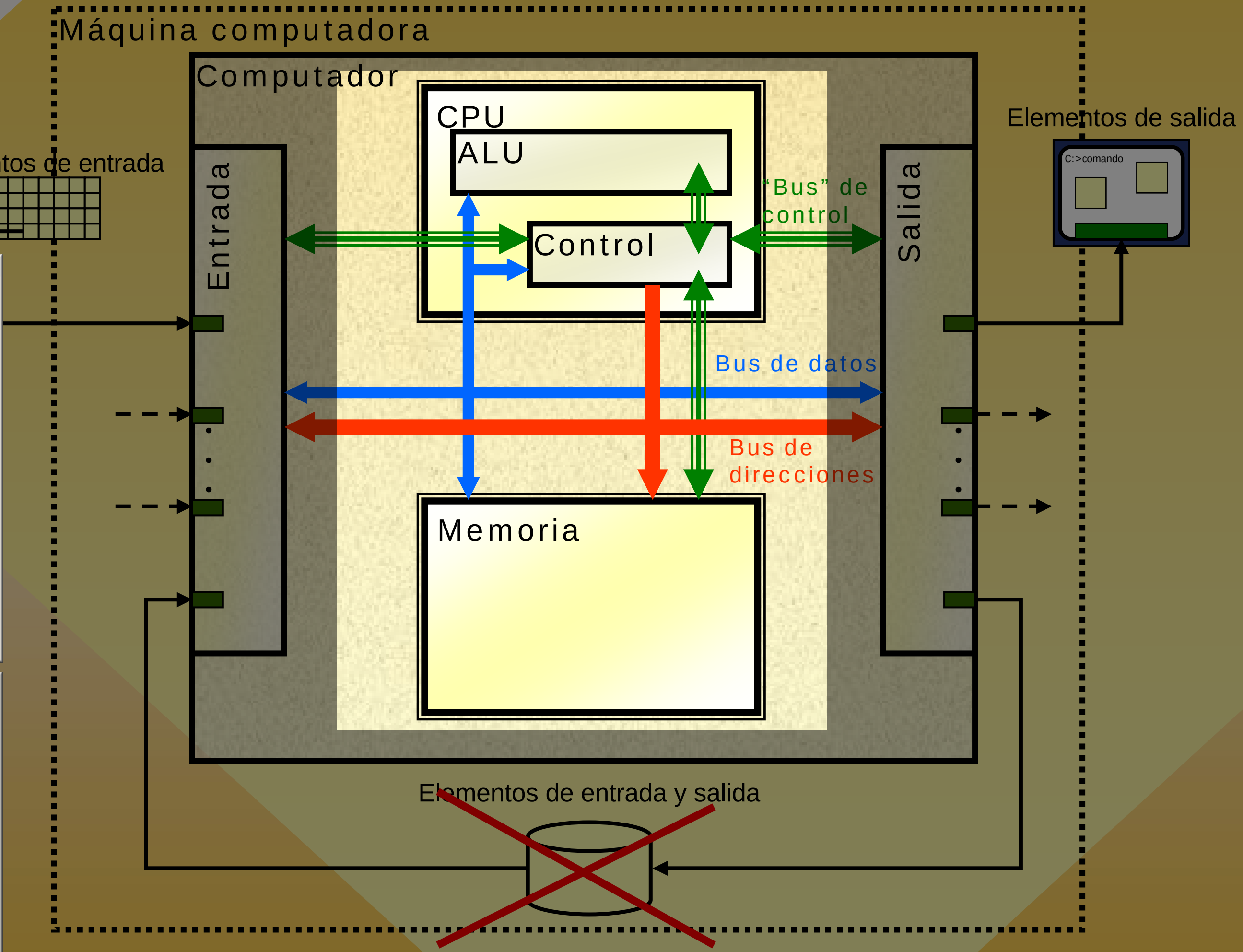
Control Wires

```
Select-Add
Select-Subtract
Select-AND
Select-OR
Select-NOT
Select-Shift-Left
Select-Shift-Right
Load-Data-Into-Memory
Load-X-From-AC
Load-Y-From-Memory
Load-Y-From-IR
Load-AC-From-ALU
Load-AC-From-Memory
Load-AC-From-IR
Increment-AC
Decrement-AC
Load-Flag-From-ALU
Load-PC-From-Memory
Load-PC-From-IR
Increment-PC
Load-IR-From-Memory
Load-ADDR-From-IR
Load-ADDR-From-Y
Load-ADDR-From-PC
Set-COUNT-To-Zero
Stop-Clock
```

Registers

Default display

```
X: 0          COUNT: 0
Y: 0          IR: ADD 0
FLAG: 0       PC: 0
AC: 0         ADDR: 0
```



When a silicon wafer is heated to about 1200 degrees Celsius in an atmosphere of water vapour or oxygen a skin of silicon dioxide forms on the surface. This skin is a most effective seal against the ingress of moisture at room temperatures and has made possible the method of manufacture of planar transistors which is described below.

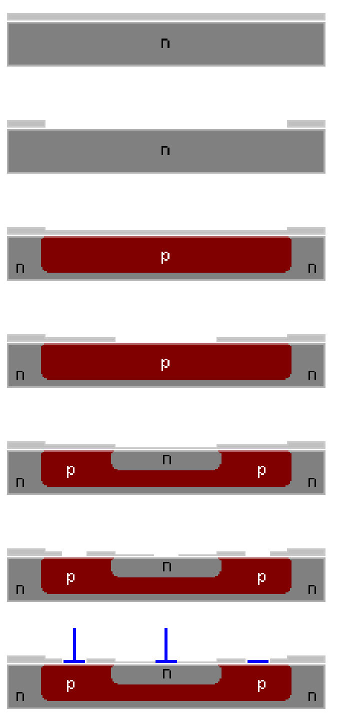
A crystal of n-type silicon, about 1 inch in diameter, is cut into slices about 0.008 inches thick. The slices are lapped and etched to approximately 0.003 iches thickness and, if required, an epitaxial layer can be formed on one surface. The slices are now heated in an oxidising atmosphere to acquire a protective coating of silicon dioxide. At this stage each slice has a sectional view similar to that shown in (a) below. Each slice yields ultimately up to 1,000 transistors and the next stage is to mark off the individual transistors. This is achieved by a photo-lithographic process: each slice is coated in a dark room with a photo-sensitive material (known as photo-resist) and is then exposed to ultra-violet light via a mask containing an array of apertures corresponding to the base areas of the 1,000 transistors. The slice is now developed to remove the photo-resist from these regions thus exposing the silicon dioxide coating. Next the slice is treated with an etch which removes the silicon dioxide from the exposed regions. The remainder of the photo-resist is now dissolved: the cross-section of the slice now appears as in (b) below which shows a gap in the layer of silicon dioxide defining the base area for a single transistor.

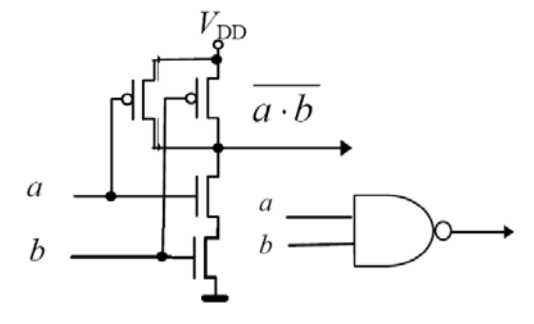
The slice is now exposed at a high temperature to a boron-rich atmosphere. The silicon dioxide coating protects the slice against diffusion of boron except at the exposed areas and here boron diffuses isotropically, i.e. horizontally under the protective coating as well as vertically into the crystal, thus forming a p-type base region. Other more precise ways of forming such a region have been developed, for example by ion implantation. This involves a sharply defined bombardment of the substrate by a beam from an ion gun which enables the active base area to be closely controlled in area and shape, a process which can be compared with precision etching. The slice is now returned to the oxidising atmosphere and a coating of silicon dioxide is formed over the base areas (and the rest of the slice) to give a cross-section similar to that shown in (c) below.

The emitter areas are now defined by a similar process of masking, photo-lithography, exposure to ultra-violet light, etching, etc., and the silicon dioxide is removed from the emitter areas to give a cross-section such as that shown in (d) below. The slice is now heated whilst exposed to an atmosphere rich in phosphorus. This forms an n-type emitter region by diffusion and the exposed area is again sealed by heating the slice in an oxidising atmosphere to form a layer of silicon dioxide. See (e) below.

Holes are now made in the silicon dioxide coating as shown in (f) to permit ohmic contacts to be made to the base and emitter areas, the position of the holes being again determined by a mask. Contacts are then made to the transistors by a process of evaporation: the slice is placed in a vacuum chamber in which aluminium is evaporated, e.g. from a hot filament. This results in a deposition of a thin coating of aluminium over the entire face of the slice. Finally the aluminium is removed from the areas in which it is not required by a masking and selective etching operation. The slice is now divided up into individual transistors and connections are made to the base and emitter regions of each transistor as shown in (g) below. The base area of each transistor is sometimes of approximately annular shape surrounding a circular emitter area but in power transistors both base and emitter areas may be in the form of parallel strips.

Planar transistors lend themselves well to mass production. Planar technology revolutionised silicon transistor manufacture in the 1960s. The transistors are particularly robust and the protection of the silicon dioxide coating is such that even without sealing in cans the transistors will operate well under boiling water! Leakage currents are very low and the transistors can be designed to work at frequencies well over 1 GHz. In 1963 the process also made possible for the first time mass production of f.e.t.s although the pronciple of this type of transistor had been described by Shockley 11 years earlier.

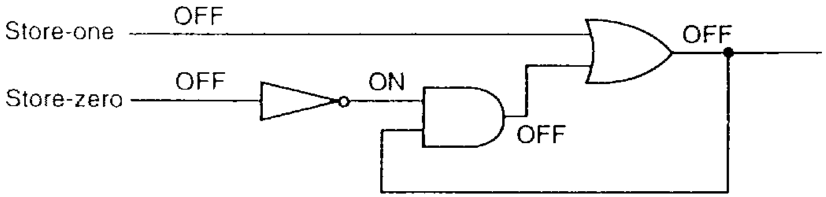


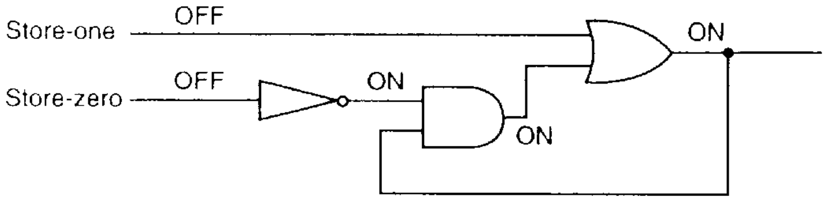


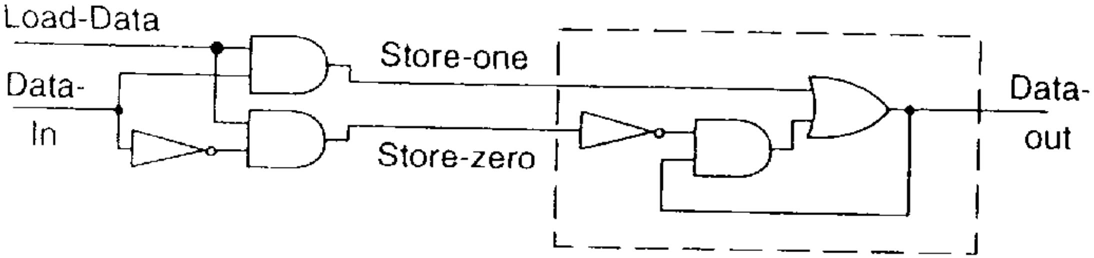
$$\overline{A}=\overline{AA}$$

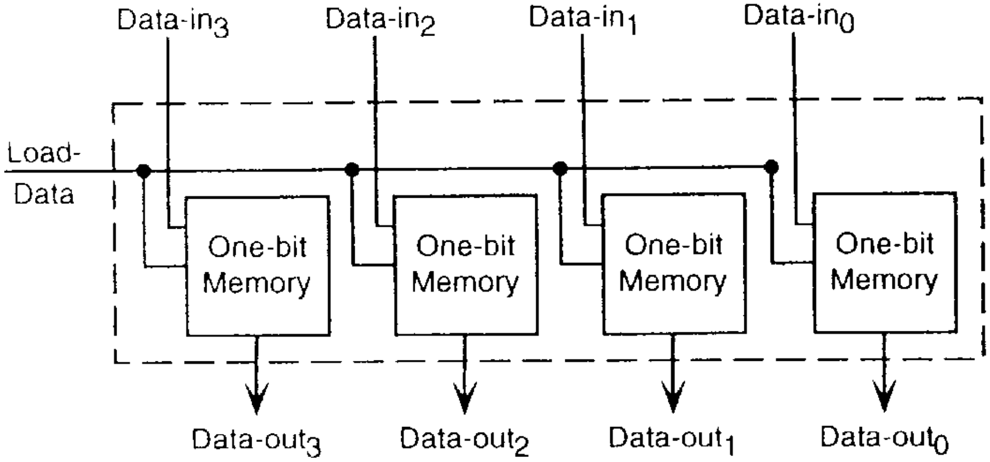
$$AB = \overline{(\overline{AB})}$$

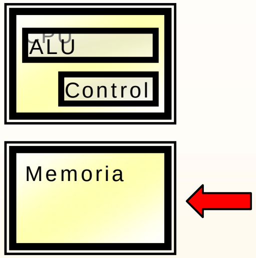
$$A+B= \overline{(\overline{AA})(\overline{BB})}$$


Store-one
OFF
Store-zero
OFF
ON
OFF
OFF


Store-one
OFF
Store-zero
OFF
ON
ON
ON


Load-Data
Data-
In
Store-one
Store-zero
Data-
out


Data-in$_3$
Data-in$_2$
Data-in$_1$
Data-in$_0$
Load-
Data
One-bit
Memory
One-bit
Memory
One-bit
Memory
One-bit
Memory
Data-out$_3$
Data-out$_2$
Data-out$_1$
Data-out$_0$

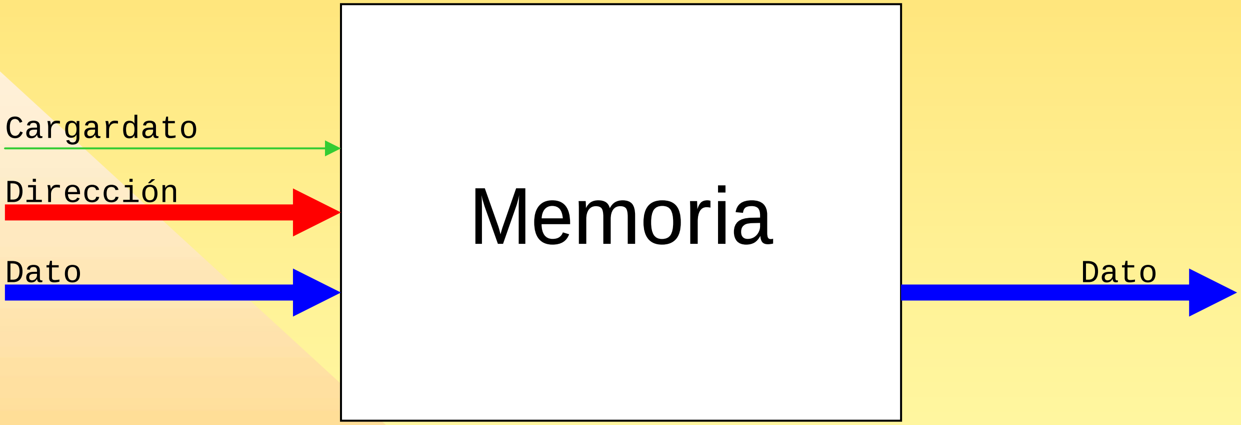


CPU

ALU

Control

Memoria



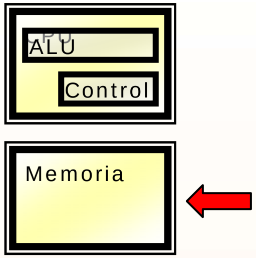
Cargardato

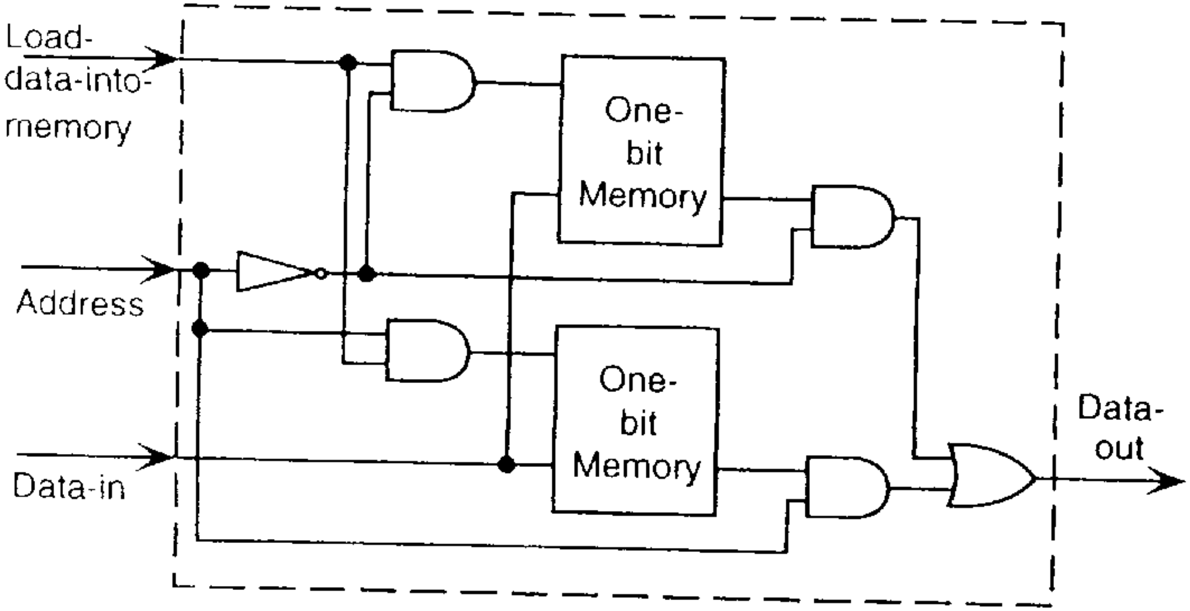
Dirección

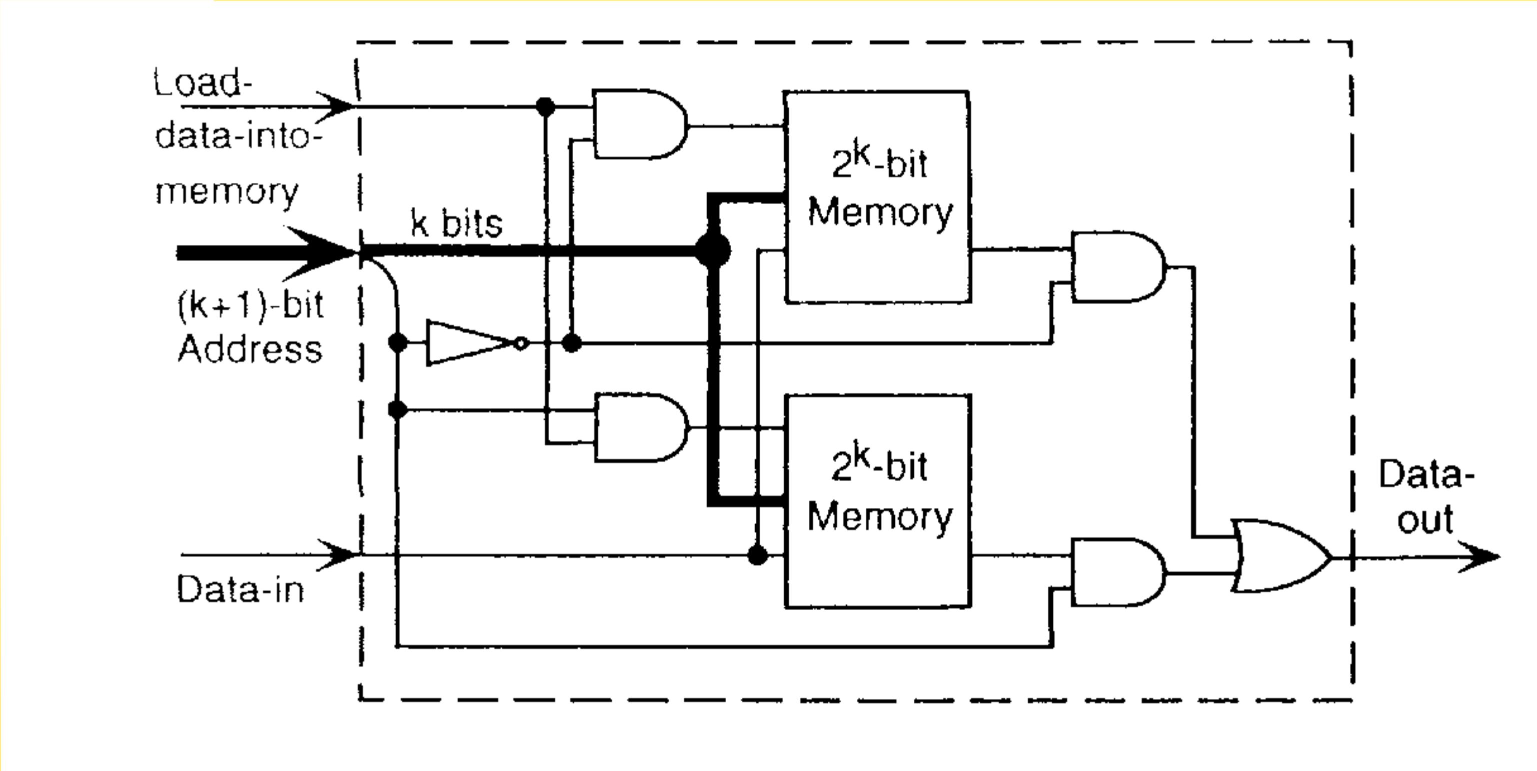
Dato

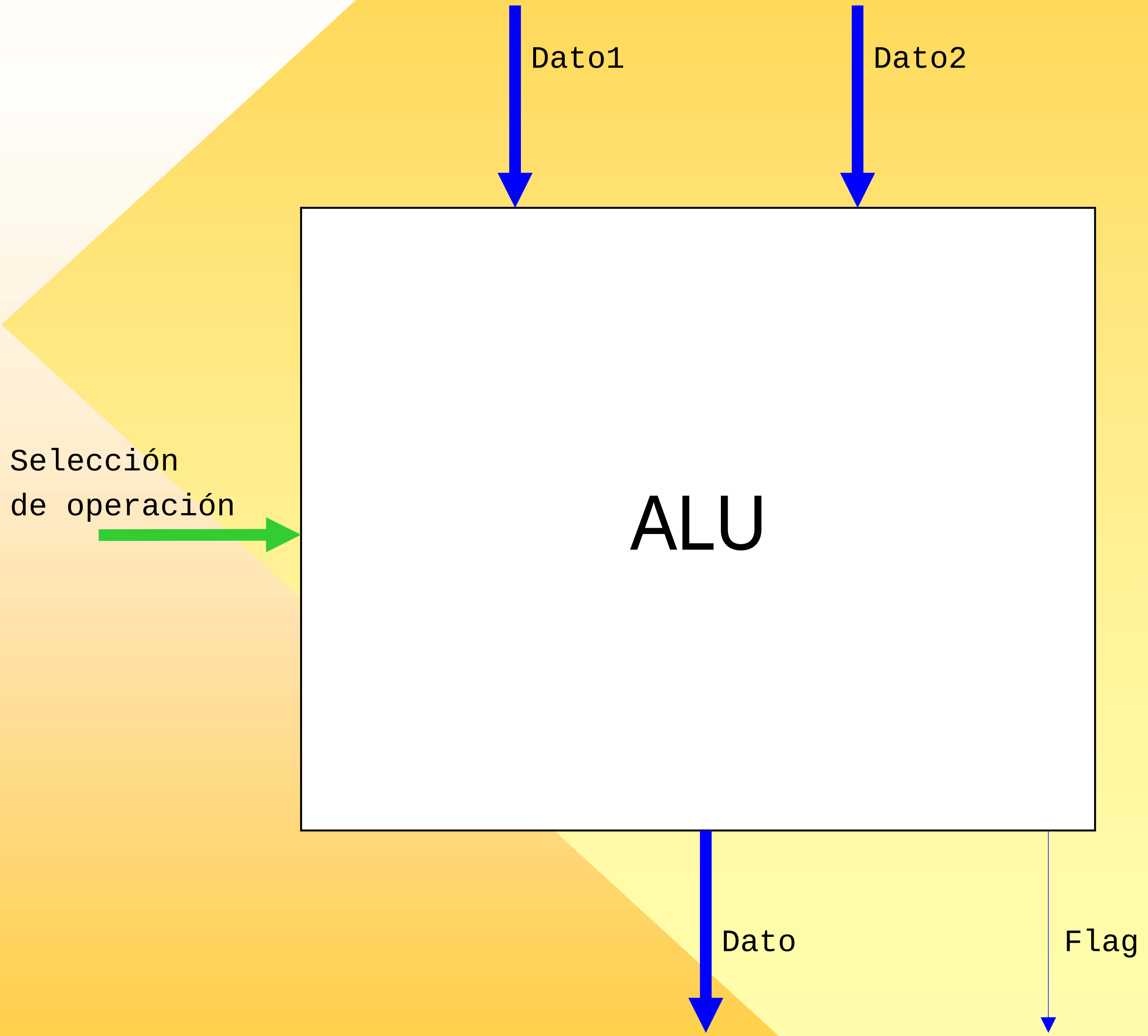
Memoria

Dato

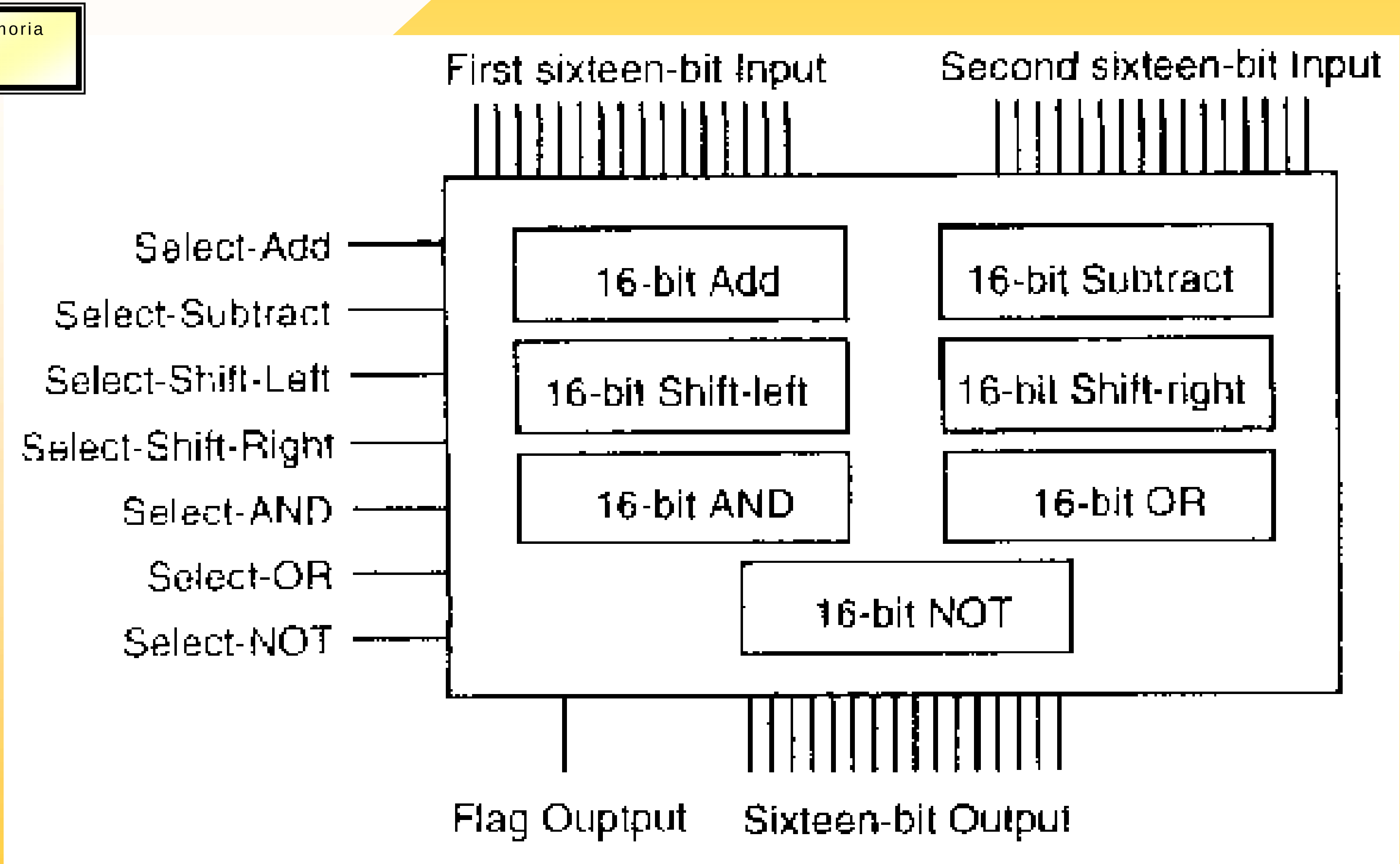

CPU
ALU
Control
Memoria


Load-data-into-memory
Address
Data-in
One-bit Memory
One-bit Memory
Data-out

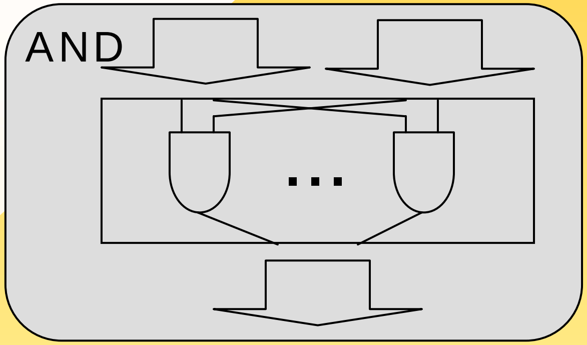

Load-
data-into-
memory
k bits
(k+1)-bit
Address
2k-bit
Memory
2k-bit
Memory
Data-
out
Data-in

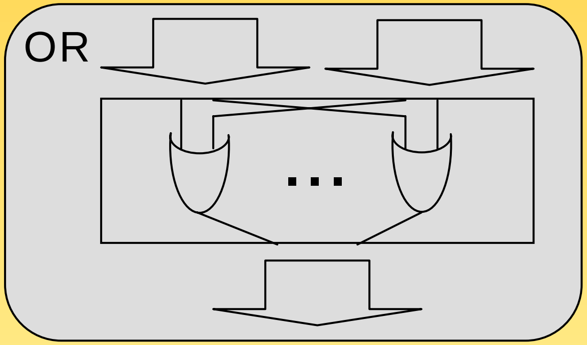

Dato1
Dato2
Selección
de operación
ALU
Dato
Flag

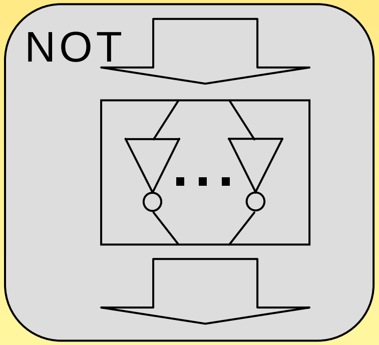
ADD,SUB,AND,OR,NOT,SHL,SHR

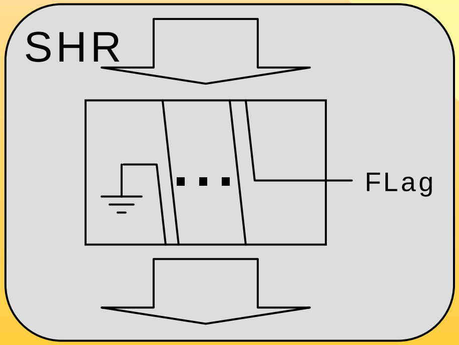


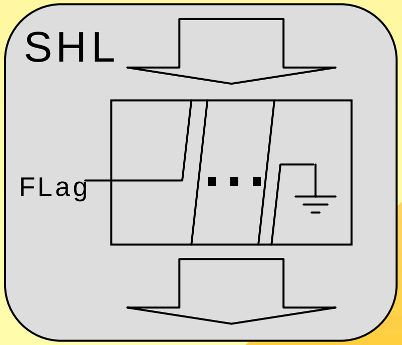
ADD,SUB,AND,OR,NOT,SHL,SHR





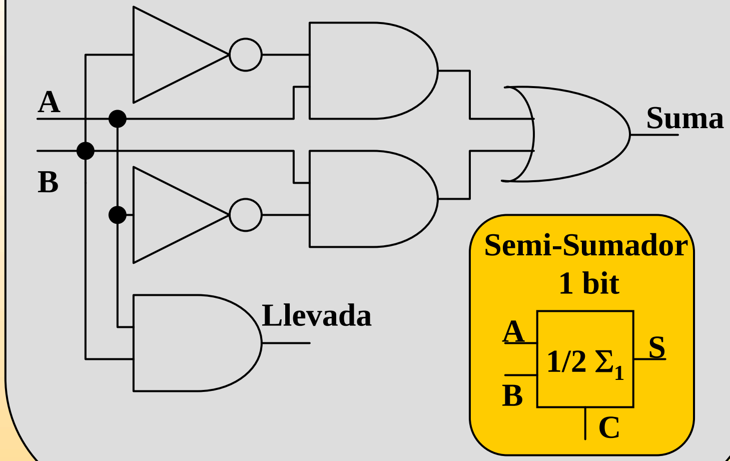


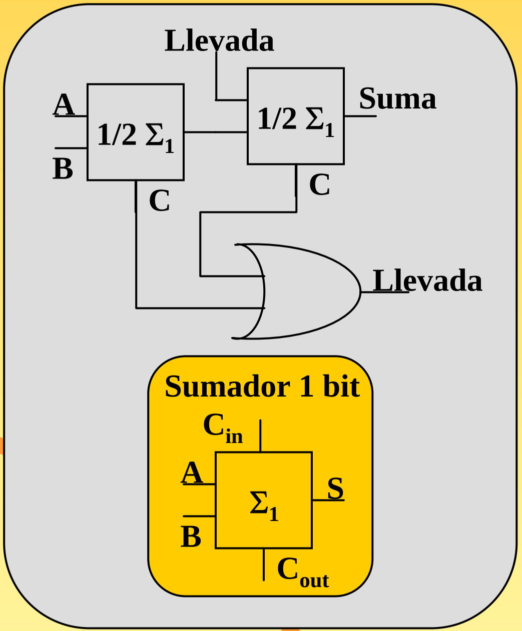


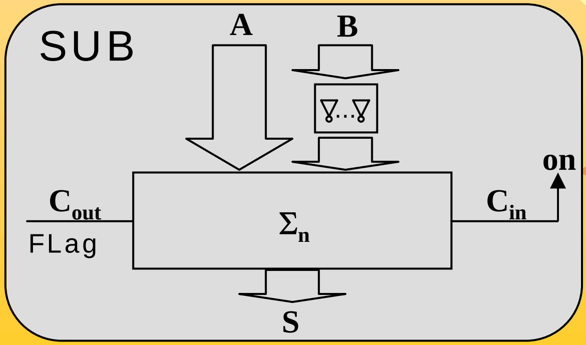


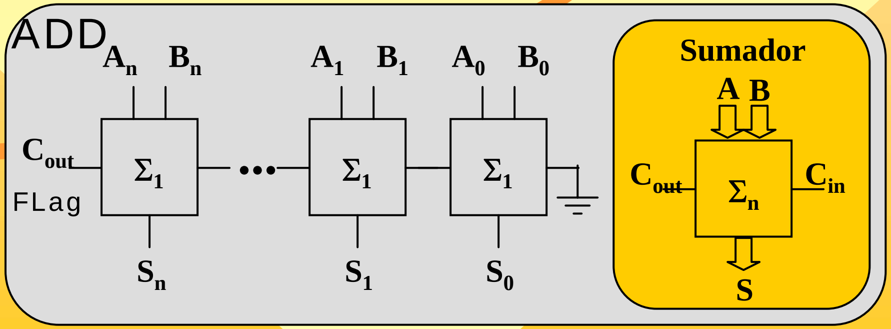
ADD,SUB,AND,OR,NOT,SHL,SHR

A	B	Suma	Llevada
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



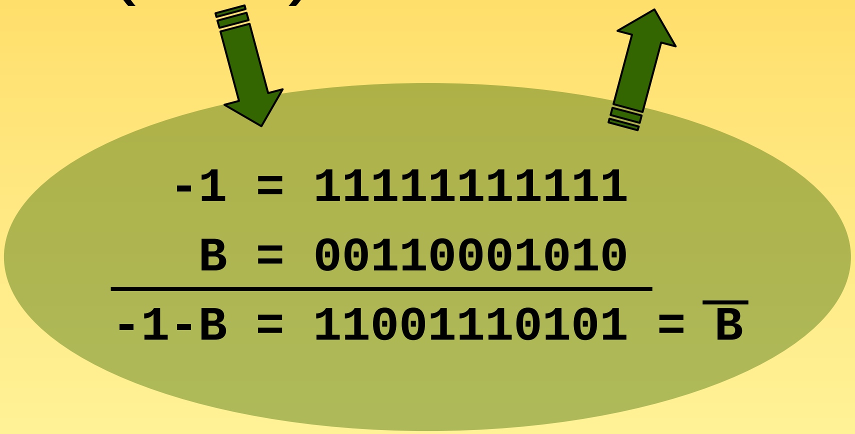






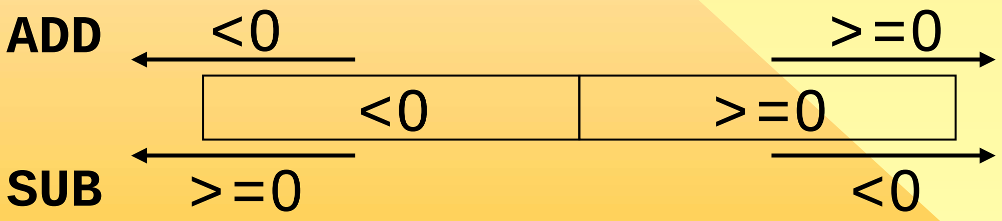
Resta en complemento a 2 basada en suma

$$A-B = A+(-B-1+1) = A+(-1-B)+1 = A+\overline{B}+1$$



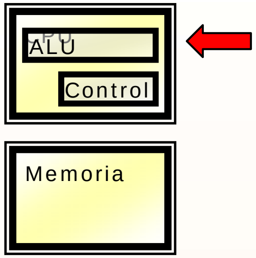
$$\begin{array}{rl} -1 = & 11111111111 \\ B = & 00110001010 \\ \hline -1-B = & 11001110101 = \overline{B} \end{array}$$

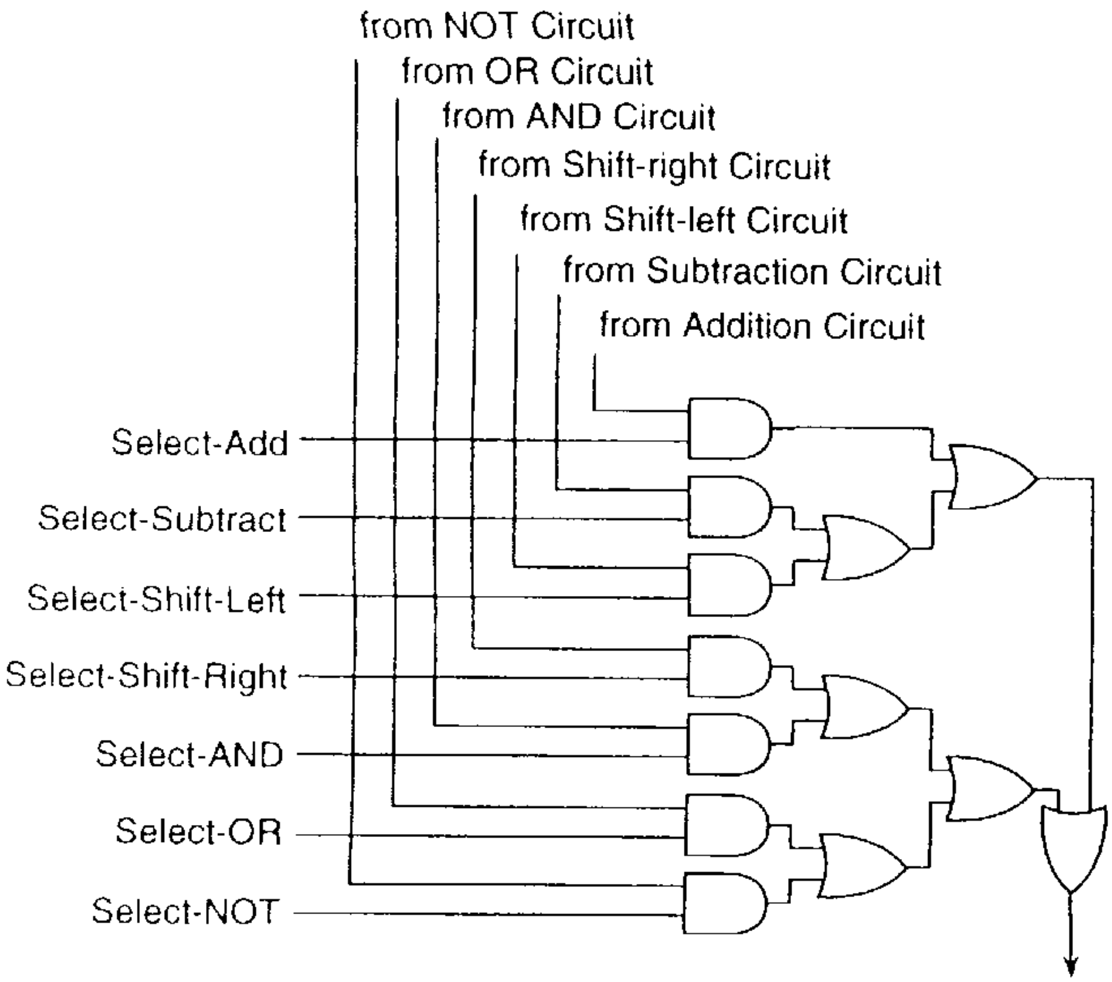
Situaciones de rebose en ADD y SUB

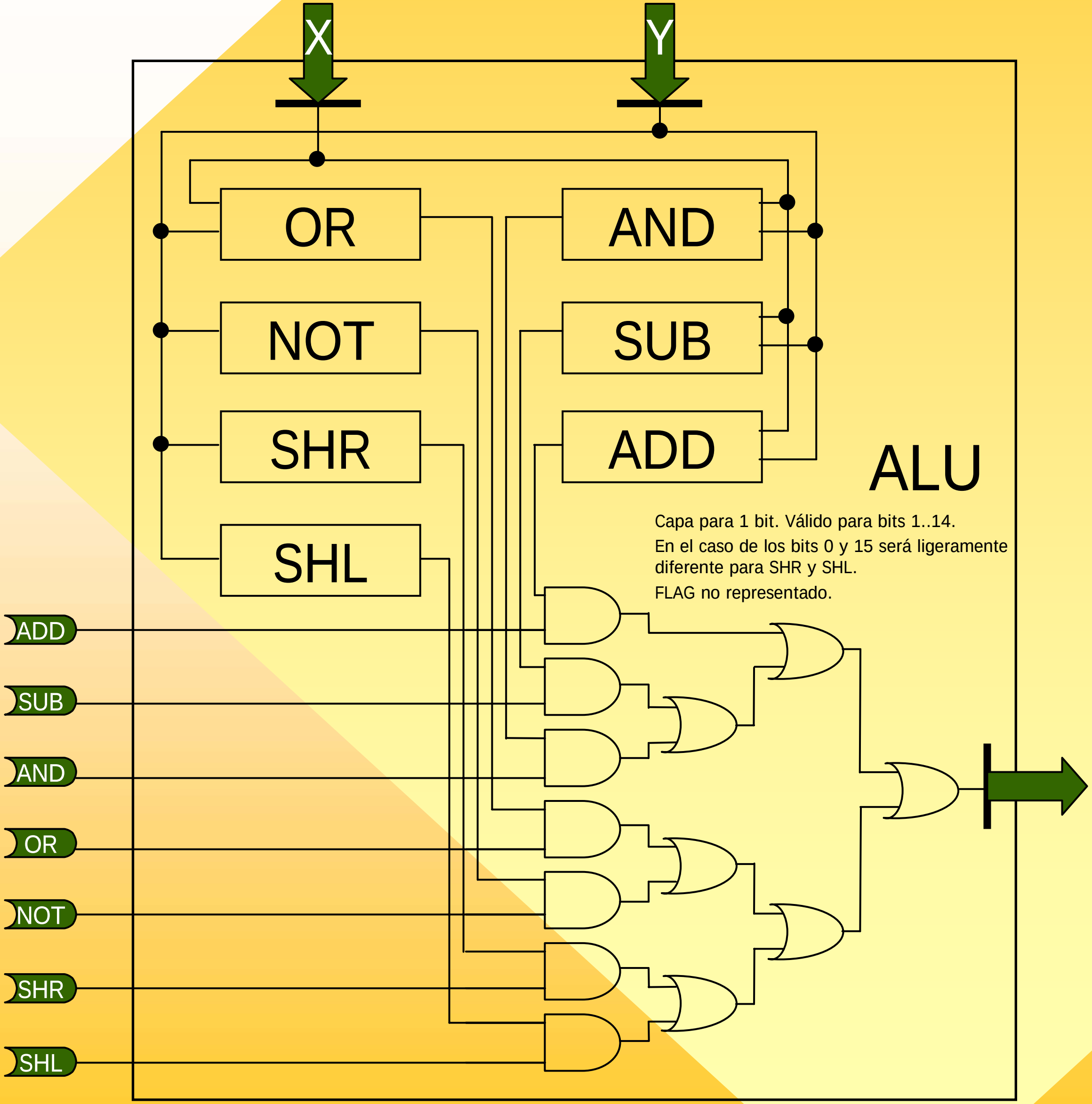


(+)+(+)=(-)   (-)+(-)=(+)

(+)-(-)=(-)   (-)-(+)=(+)


CPU
ALU
Control
Memoria

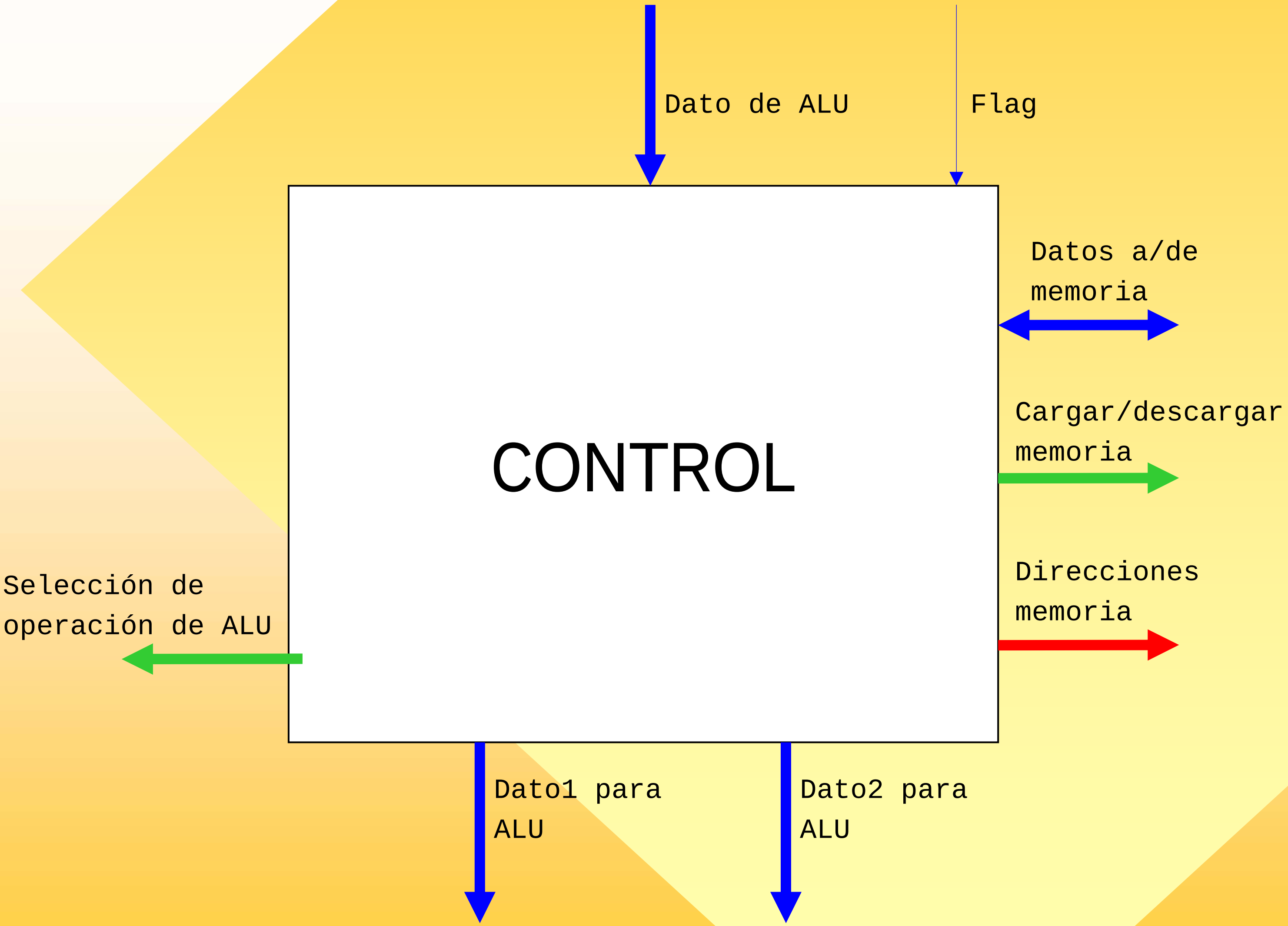

from NOT Circuit
from OR Circuit
from AND Circuit
from Shift-right Circuit
from Shift-left Circuit
from Subtraction Circuit
from Addition Circuit
Select-Add
Select-Subtract
Select-Shift-Left
Select-Shift-Right
Select-AND
Select-OR
Select-NOT

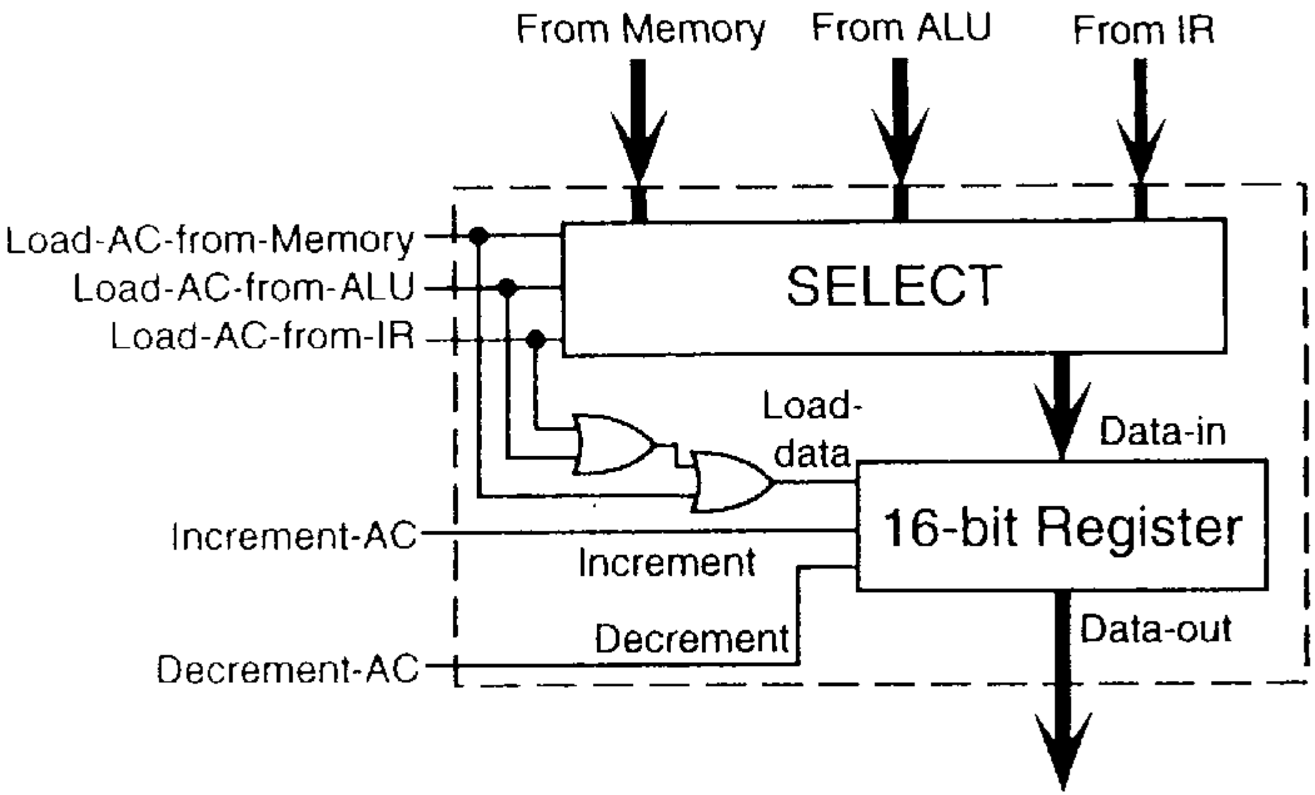


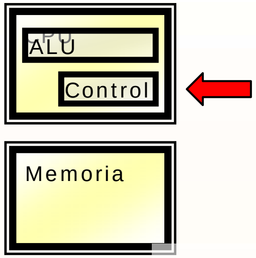
Capa para 1 bit. Válido para bits 1..14.

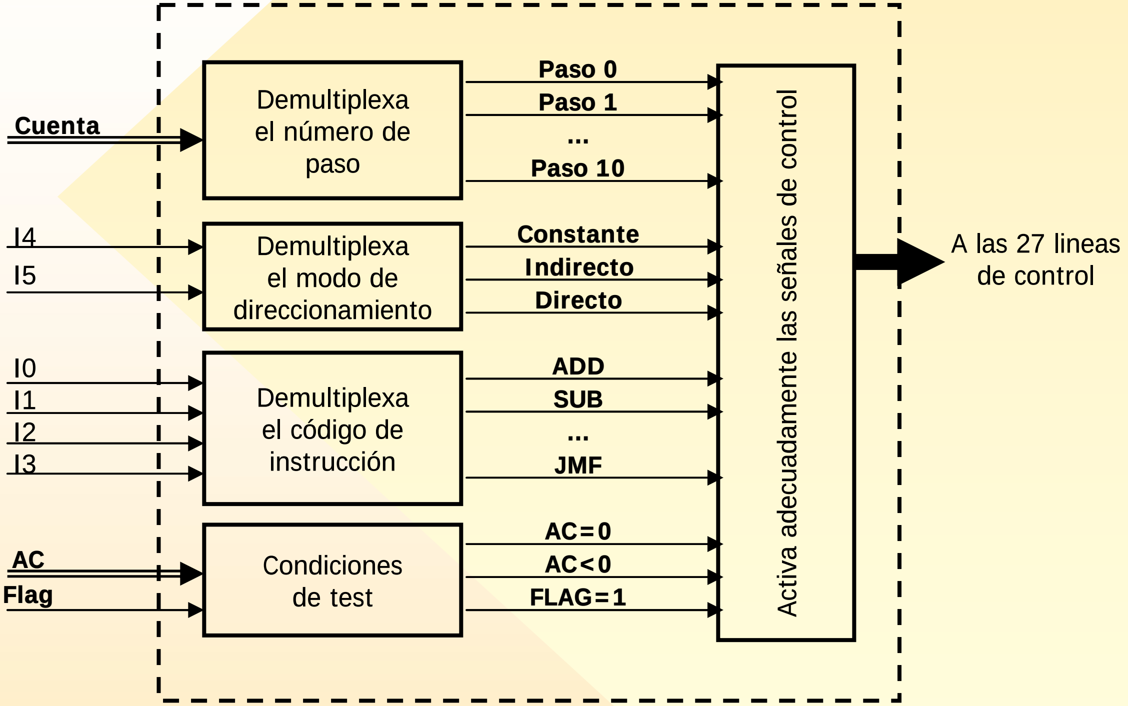
En el caso de los bits 0 y 15 será ligeramente diferente para SHR y SHL.

FLAG no representado.

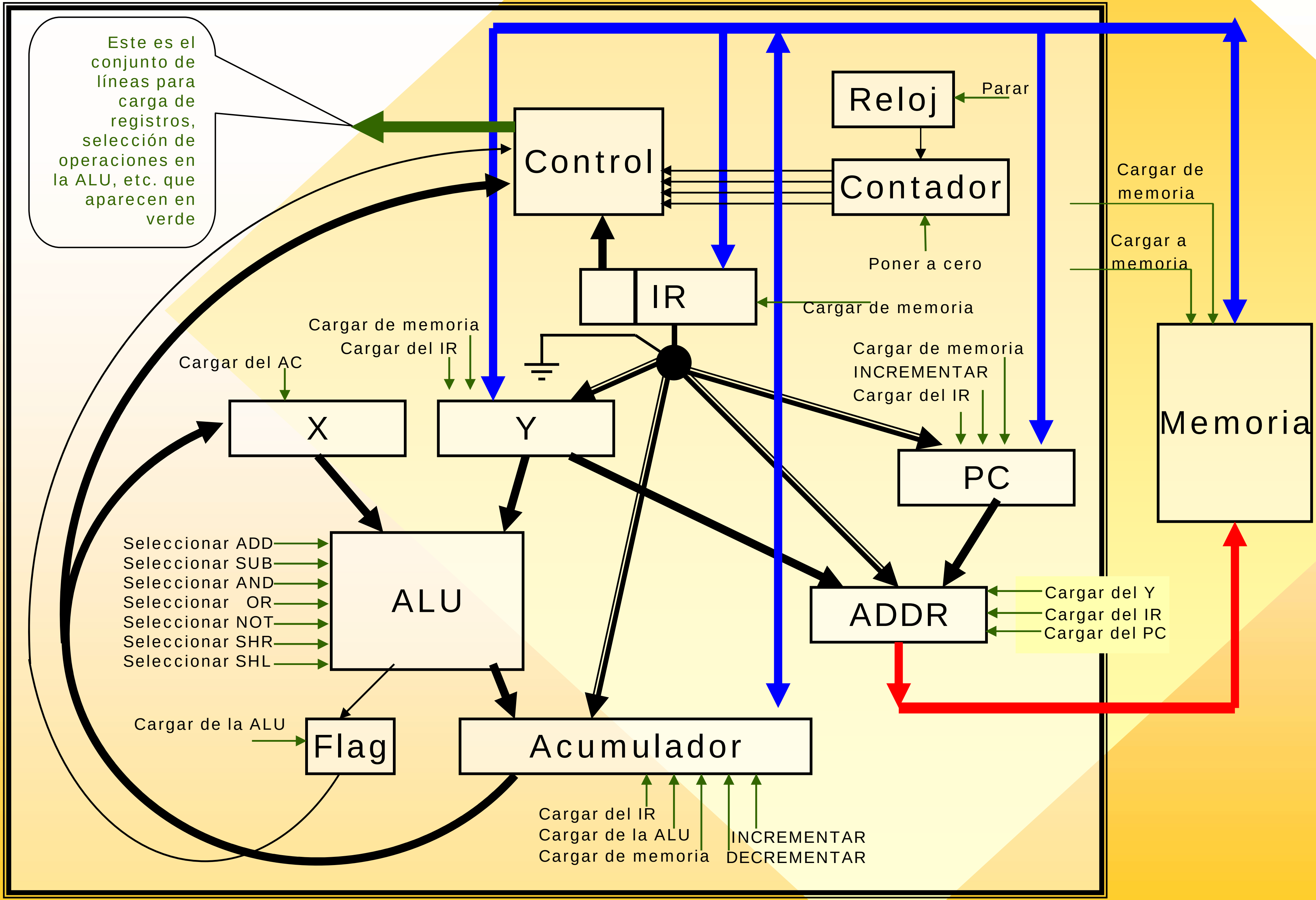

Dato de ALU
Flag
Datos a/de memoria
Cargar/descargar memoria
CONTROL
Direcciones memoria
Selección de operación de ALU
Dato1 para ALU
Dato2 para ALU


From Memory
From ALU
From IR
Load-AC-from-Memory
Load-AC-from-ALU
Load-AC-from-IR
SELECT
Load-data
Data-in
16-bit Register
Increment-AC
Increment
Decrement
Decrement-AC
Data-out

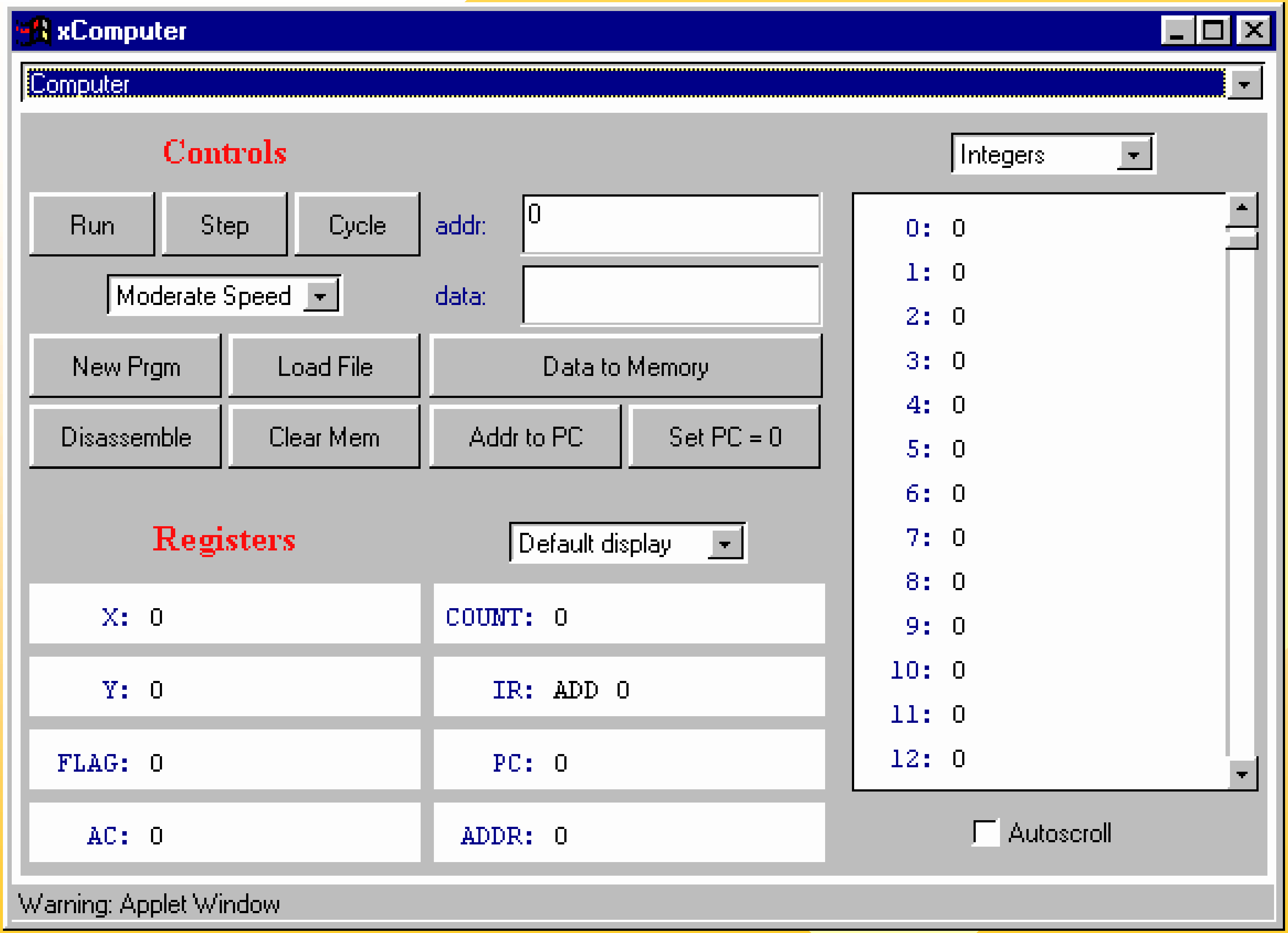

CPU
ALU
Control
Memoria


Cuenta
Demultiplexa el número de paso
Paso 0
Paso 1
...
Paso 10
I4
I5
Demultiplexa el modo de direccionamiento
Constante
Indirecto
Directo
I0
I1
I2
I3
Demultiplexa el código de instrucción
ADD
SUB
...
JMF
AC
Flag
Condiciones de test
AC=0
AC<0
FLAG=1
Activa adecuadamente las señales de control
A las 27 lineas de control

1. Cargar ADDR del PC
2. Cargar IR de memoria
3. Incrementar PC
4. Parar el reloj
5. Cargar ADDR del IR
6. Cargar AC de memoria
7. Cargar memoria del AC
8. Incrementar AC
9. Decrementar AC
10. Cargar X del AC
11. Cargar Y de memoria
12. Seleccionar ADD
13. Seleccionar SUB
14. Seleccionar AND
15. Seleccionar OR
16. Seleccionar NOT
17. Seleccionar SHR
18. Seleccionar SHL
19. Cargar AC de la ALU
20. Cargar Flag
21. Cargar PC del IR
22. Cargar AC del IR
23. Cargar ADDR del Y
24. Cargar PC de memoria
25. Cargar Y de IR
26. Almacenar en memoria
27. Poner a cero el contador



Este es el conjunto de líneas para carga de registros, selección de operaciones en la ALU, etc. que aparecen en verde



Computer

Integers

addr: 0

data:

Moderate Speed

Default display

```
   X: 0          COUNT: 0
   Y: 0             IR: ADD 0
FLAG: 0             PC: 0
  AC: 0           ADDR: 0
```

```
 0: 0
 1: 0
 2: 0
 3: 0
 4: 0
 5: 0
 6: 0
 7: 0
 8: 0
 9: 0
10: 0
11: 0
12: 0
```

□ Autoscroll

Warning: Applet Window

FETCH

S01 1. Cargar ADDR desde el PC

S02 2. Cargar IR desde memoria

S03 3. Incrementar PC

HLT

S04 4. Parar el reloj

LOD

S05 4. Cargar ADDR desde el IR

S06 5. Cargar AC desde memoria

S27 6. Poner COUNT a cero

STO

S05 4. Cargar ADDR desde el IR

S07 5. Cargar AC en memoria

S27 6. Poner COUNT a cero

INC

S08 4. Incrementar AC

S27 5. Poner COUNT a cero

Dec

S09 4. Decrementar AC

S27 5. Poner COUNT a cero

german.bordel@ehu.es
http://gtts.ehu.es/German

INS = ADD, SUB

S05 4. Cargar ADDR desde el IR

S10 S11 5. Cargar X desde el AC y Cargar Y desde memoria

S12/13 S19 S20 6. Seleccionar INS, Cargar AC desde la ALU, Cargar Flag desde la ALU

S12/13 7. Seleccionar INS.

S27 8. Poner COUNT a cero

INS = AND, OR

S05 4. Cargar ADDR desde el IR

S10 S11 5. Cargar X desde el AC y Cargar Y desde memoria

S14/15 S19 6. Seleccionar INS, Cargar AC desde la ALU

S14/15 7. Seleccionar INS.

S27 8. Poner COUNT a cero

INS = SHR, SHL

S10 4. Cargar X desde el AC

S17/18 S19 S20 5. Seleccionar INS, Cargar AC desde la ALU, Cargar Flag desde la ALU

S17/18 6. Seleccionar INS.

S27 7. Poner COUNT a cero

NOT

S10 4. Cargar X desde el AC

S16 S19 5. Seleccionar NOT, Cargar AC desde la ALU,

S16 6. Seleccionar NOT.

S27 7. Poner COUNT a cero

JMP

S21 4. Cargar PC desde el IR

S27 5. Poner COUNT a cero

JMZ

S21 4. Si AC=0 Cargar PC desde el IR

S27 5. Poner COUNT a cero

JMN

S21 4. Si AC<0 Cargar PC desde el IR

S27 5. Poner COUNT a cero

JMF

S21 4. Si FLAG=1 Cargar PC desde el IR

S27 5. Poner COUNT a cero

LOD-C

S22 4. Cargar AC desde IR

S27 5. Poner COUNT a cero

LOD

S05 4. Cargar ADDR desde el IR

S06 5. Cargar AC desde memoria

S27 6. Poner COUNT a cero

LOD-I

S05 4. Cargar ADDR desde el IR

S11 5. Cargar Y desde memoria

S23 6. Cargar ADDR desde el Y

S06 7. Cargar AC desde memoria

S27 8. Poner COUNT a cero

STO

S05 4. Cargar ADDR desde el IR

S07 5. Cargar memoria desde AC

S27 6. Poner COUNT a cero

STO-I

S05 4. Cargar ADDR desde el IR

S11 5. Cargar Y desde memoria

S23 6. Cargar ADDR desde el Y

S07 7. Cargar memoria desde AC

S27 8. Poner COUNT a cero

ADD-C (INS = ADD), SUB-C (INS = SUB)

4. Cargar X desde el AC y Cargar Y desde IR
5. Seleccionar INS, Cargar AC desde la ALU, Cargar Flag desde la ALU
6. Seleccionar INS.
7. Poner COUNT a cero

INS = ADD, SUB

S05 4. Cargar ADDR desde el IR
5. Cargar X desde el AC y Cargar Y desde memoria
6. Seleccionar INS, Cargar AC desde la ALU, Cargar Flag desde la ALU
7. Seleccionar INS.
8. Poner COUNT a cero

ADD-I (INS = ADD), SUB-I (INS = SUB)

S05 4. Cargar ADDR desde el IR
5. Cargar Y desde memoria
6. Cargar ADDR desde el Y
7. Cargar X desde el AC y Cargar Y desde memoria
8. Seleccionar INS, Cargar AC desde la ALU, Cargar Flag desde la ALU
9. Seleccionar INS.
10. Poner COUNT a cero

AND-C (INS = AND), OR-C (INS = OR)

4. Cargar X desde el AC y Cargar Y desde IR
5. Seleccionar INS, Cargar AC desde la ALU
6. Seleccionar INS.
7. Poner COUNT a cero

INS = AND, OR

S05 4. Cargar ADDR desde el IR
5. Cargar X desde el AC y Cargar Y desde memoria
6. Seleccionar INS, Cargar AC desde la ALU
7. Seleccionar INS.
8. Poner COUNT a cero

AND-I (INS = AND), OR-I (INS = OR)

S05 4. Cargar ADDR desde el IR
5. Cargar Y desde memoria
6. Cargar ADDR desde el Y
7. Cargar X desde el AC y Cargar Y desde memoria
8. Seleccionar INS, Cargar AC desde la ALU
9. Seleccionar INS.
10. Poner COUNT a cero

JMP

4. Cargar PC desde el IR
5. Poner COUNT a cero

JMP-I

S05 4. Cargar ADDR desde el IR
5. Cargar PC desde memoria
6. Poner COUNT a cero

JMZ

4. Si AC=0 Cargar PC desde el IR
5. Poner COUNT a cero

JMZ-I

S05 4. Cargar ADDR desde el IR
5. Si AC=0 Cargar PC desde memoria
6. Poner COUNT a cero

JMN

4. Si AC<0 Cargar PC desde el IR
5. Poner COUNT a cero

JMN -I

S05 4. Cargar ADDR desde el IR
5. Si AC<0 Cargar PC desde memoria
6. Poner COUNT a cero

JMF

4. Si FLAG=1 Cargar PC desde el IR
5. Poner COUNT a cero

JMF -I

S05 4. Cargar ADDR desde el IR
5. Si FLAG=1 Cargar PC desde memoria
6. Poner COUNT a cero

0	2406	Lod 6
1	0007	Add 7
2	x	
3	x	
4	x	
5	x	
6	0055	
7	0011	

FETCH

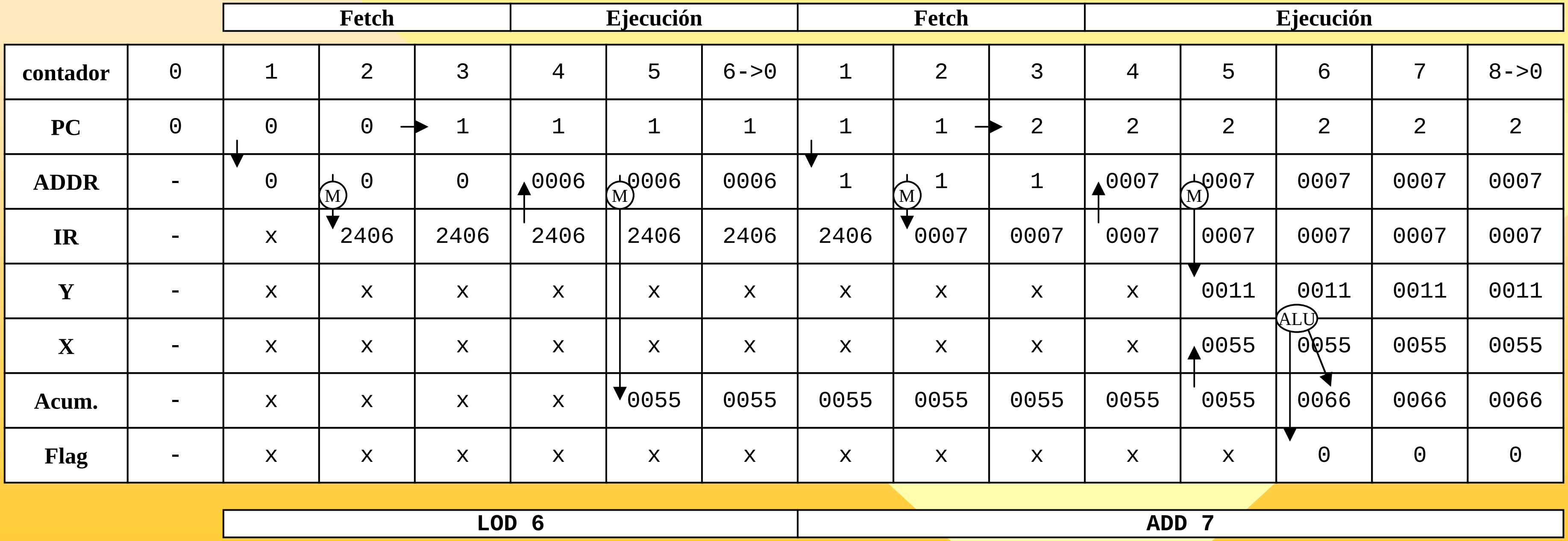
1. Cargar ADDR desde el PC
2. Cargar IR desde memoria
3. Incrementar PC

LOD

4. Cargar ADDR desde el IR
5. Cargar AC desde memoria
6. Poner COUNT a cero

INS = ADD,SUB

4. Cargar ADDR desde el IR
5. Cargar X desde el AC y Cargar Y desde memoria
6. Seleccionar INS, Cargar AC desde la ALU, Cargar Flag desde la ALU
7. Seleccionar INS.
8. Poner COUNT a cero



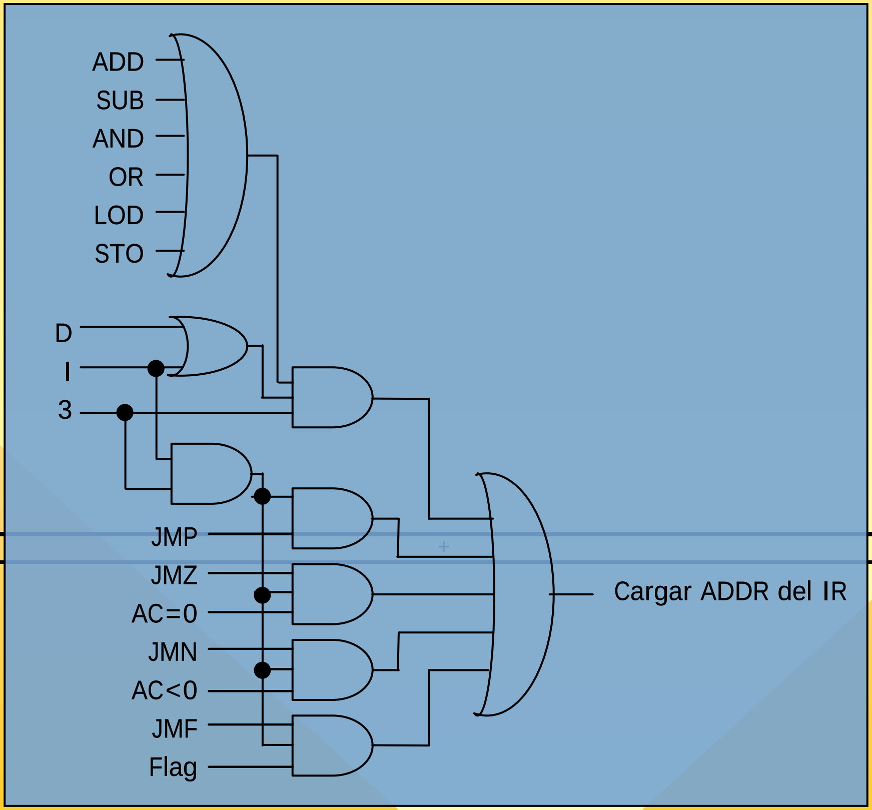
	Fetch			Ejecución			Fetch			Ejecución				
contador	0	1	2	3	4	5	6->0	1	2	3	4	5	6	7	8->0
PC	0	0	0	1	1	1	1	1	1	2	2	2	2	2	2
ADDR	-	0	0	0	0006	0006	0006	1	1	1	0007	0007	0007	0007	0007
IR	-	x	2406	2406	2406	2406	2406	2406	0007	0007	0007	0007	0007	0007	0007
Y	-	x	x	x	x	x	x	x	x	x	x	0011	0011	0011	0011
X	-	x	x	x	x	x	x	x	x	x	x	0055	0055	0055	0055
Acum.	-	x	x	x	x	0055	0055	0055	0055	0055	0055	0055	0066	0066	0066
Flag	-	x	x	x	x	x	x	x	x	x	x	x	0	0	0
		LOD 6						ADD 7							

Esquema

Cuestiones/ejercicios

1. Con nuestra máquina virtual disponemos de 1024 posiciones de memoria. ¿qué modificación deberíamos introducir para disponer de más espacio direccionable?.
2. Con nuestra máquina virtual podemos manejar datos de 16 bits. ¿qué deberíamos hacer para manejar datos más grandes?. Pensar en soluciones software y hardware.

	Contador											Dir.				Instr																	Cond		
	0	1	2	3	4	5	6	7	8	9		00	01	11		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111				
												D	C	I		ADD	SUB	AND	OR	NOT	SHL	SHR	INC	DEC	LOD	STO	HLT	JMP	JMZ	JMN	JMF		AC=0	AC<0	Flag
1.Cargar ADDR del PC	x																																		
2.Cargar IR de memoria		x																																	
3.Incrementar PC			x																																
4.Parar el reloj				x																															
5.Cargar ADDR del IR				x								x		x		x	x	x	x						x	x									
				x										x														x							
				x										x															x				x		
				x										x																x				x	
				x										x																	x				x
6.Cargar AC de memoria																																			
7.Cargar memoria del AC																																			
8.Incrementar AC																																			
9.Decrementar AC																																			
10.Cargar X del AC																																			
11.Cargar Y de memoria																																			
12.Seleccionar ADD											x				x																	x			
13.Seleccionar SUB																																			
14.Seleccionar AND																																			
15.Seleccionar OR																																			
16.Seleccionar NOT																																			
17.Seleccionar SHR																																			
18.Seleccionar SHL																																			
19.Cargar AC de la ALU																																			
20.Cargar Flag																																			
21.Cargar PC del IR																																			
22.Cargar AC del IR																																			
23.Cargar ADDR del Y																																			
24.Cargar PC de memoria																																			
25.Cargar Y de IR																																			
26.Almacenar en memoria																																			
27.Poner a cero el contador																																			
					+								+										+											+	


ADD
SUB
AND
OR
LOD
STO
D
I
3
JMP
JMZ
AC=0
JMN
AC<0
JMF
Flag
Cargar ADDR del IR