

Ayudantía IV

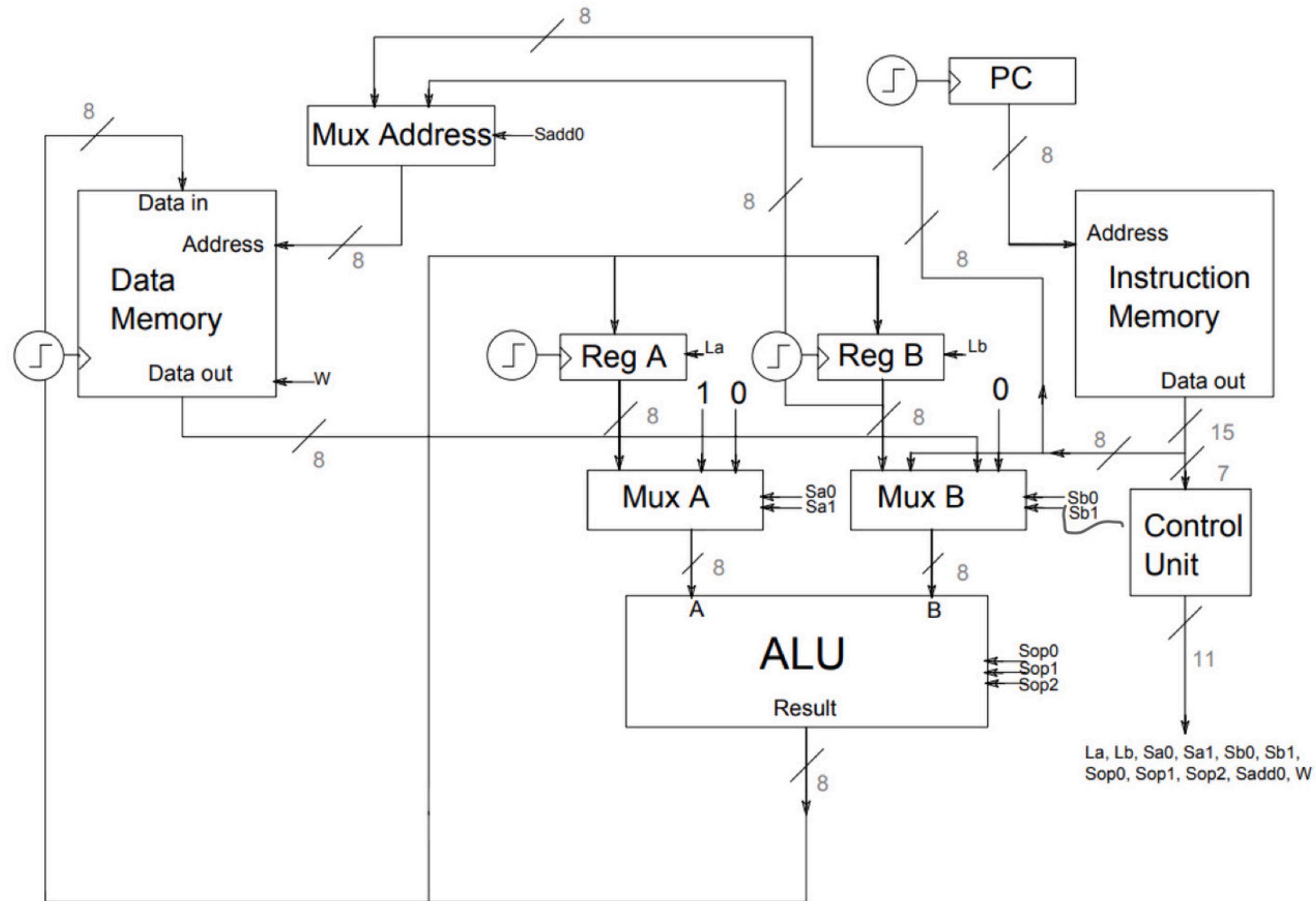
IIC2343

Programabilidad

Daniela Ríos



Diagrama Computador Básico



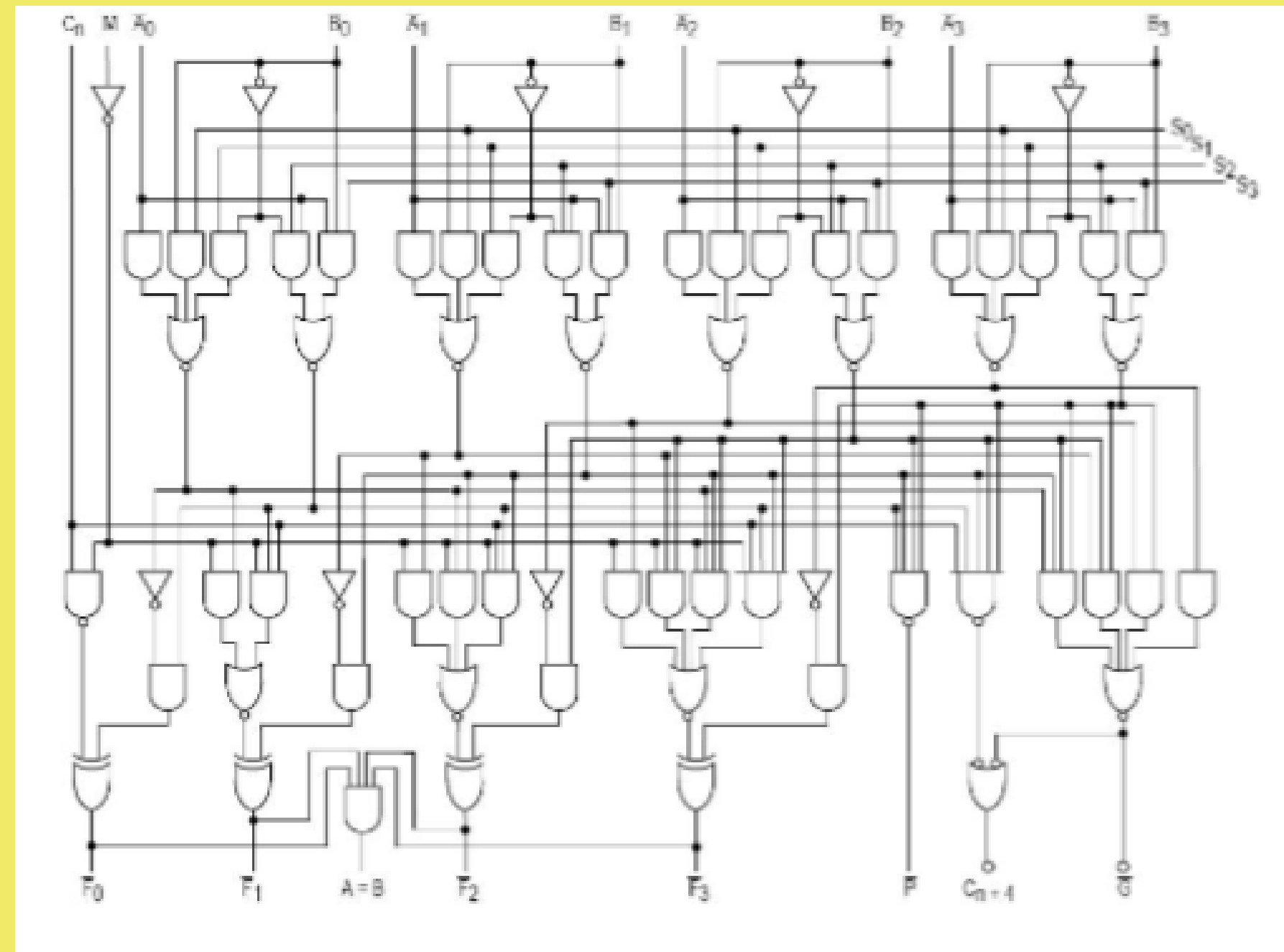
viendolo por
primera vez



ALU (Arithmetic Logic Unit)

¿Qué sabemos de la ALU?

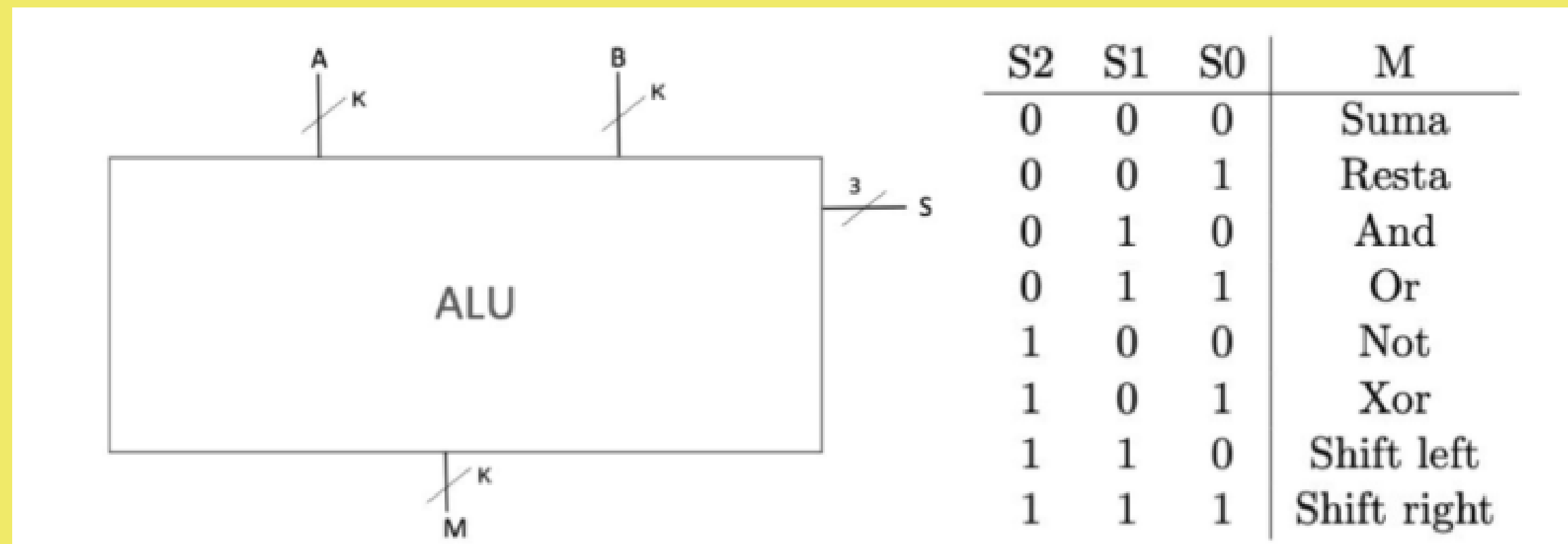
En resumen, es un circuito digital muy feo que es capaz de hacer diferentes operaciones aritméticas



ALU (Arithmetic Logic Unit)

¿Qué sabemos de la ALU?

En resumen, es un circuito digital muy feo que es capaz de hacer diferentes operaciones aritméticas. Nosotros usaremos la abstracción

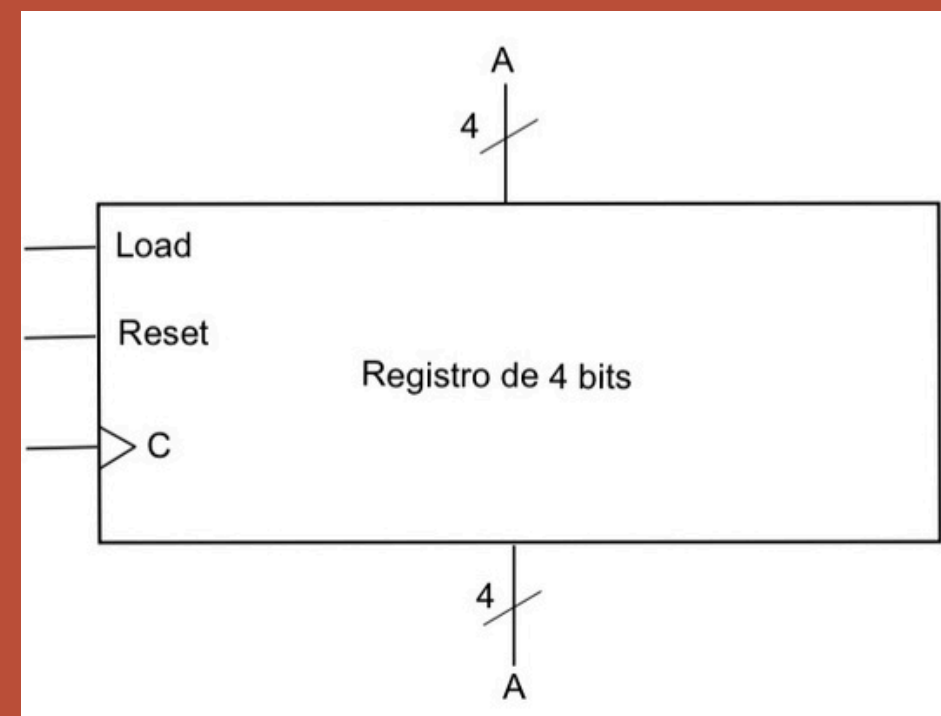
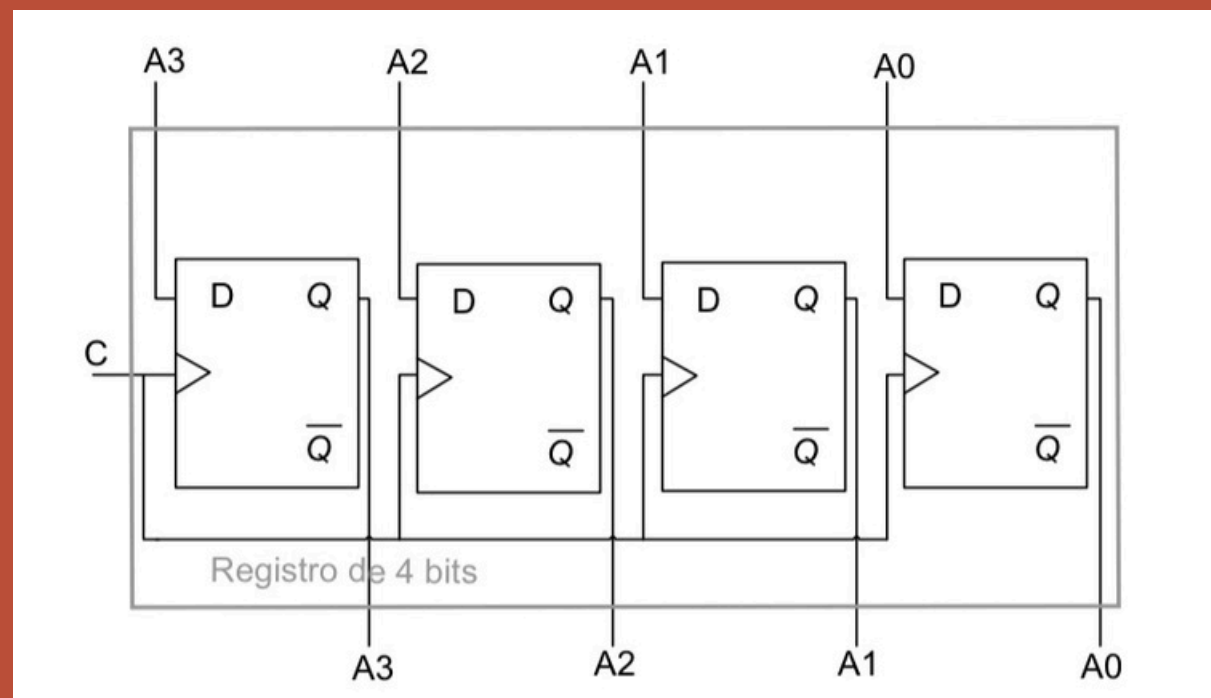


Registros

- Combinación de unidades de almacenamiento de 1 bit por ejemplo, Flip-Flop D.
- El tamaño de bits que puede almacenar un registro dependerá del número de unidades de almacenamiento de 1 bit combinadas.



Los vimos la semana pasada



Señales de control:
Load: Carga el valor de A
Reset: limpia el registro
C : Clock



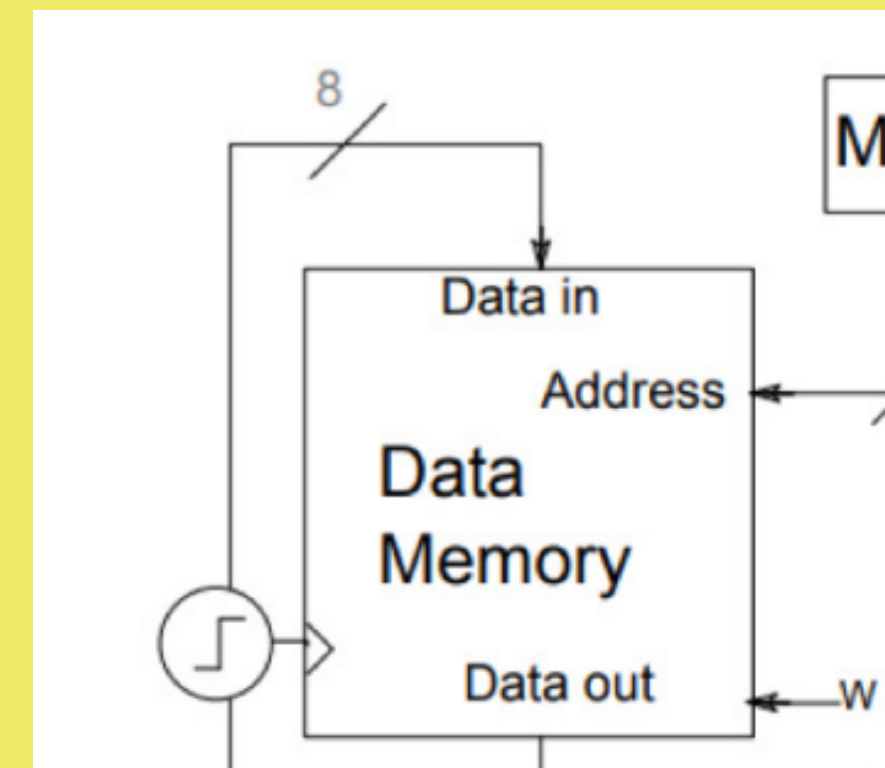
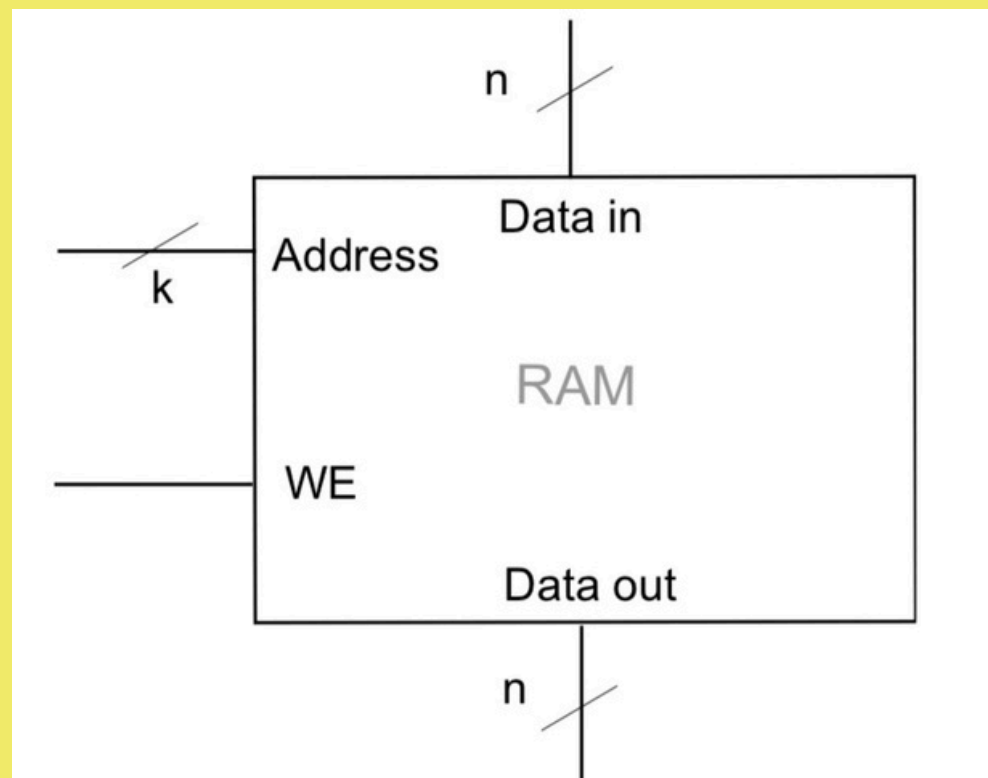
Data Memory

La Data Memory es una Memoria de escritura-lectura (Ram). Tiene tres buses de entrada:

Address : bus de direccionamiento, indica palabra que se quiere seleccionar (k)

Data in : dato que se guardará en la posición seleccionada (k)

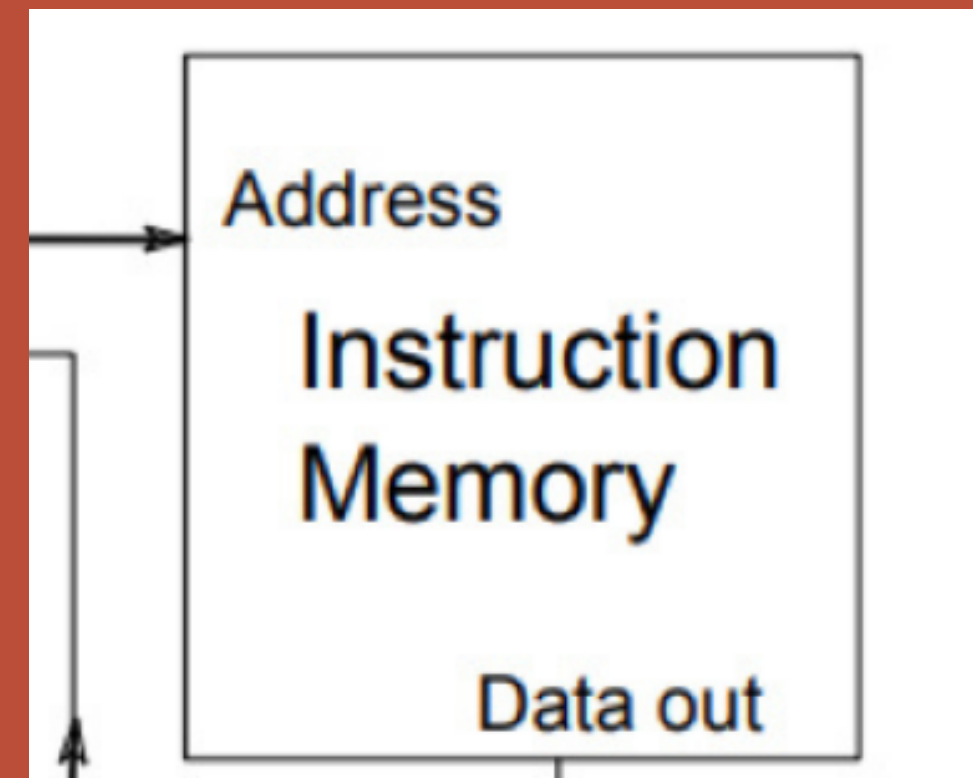
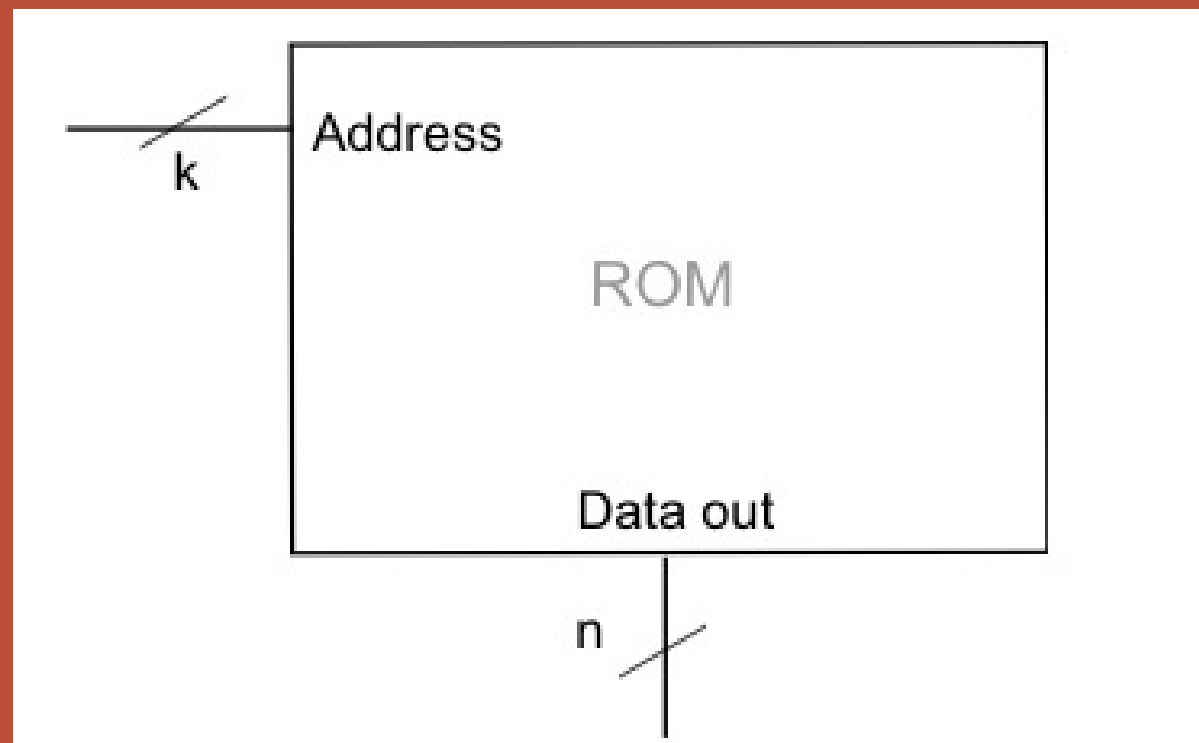
WE : Enable, indica si se escribe en la memoria



Instruction Memory

La Instruction Memory es una Memoria de lectura (Rom) en la cual se almacenan las instrucciones a ejecutar por nuestro computador. Tiene tres buses de entrada:

Address : bus de direccionamiento, indica palabra que se quiere leer (k)

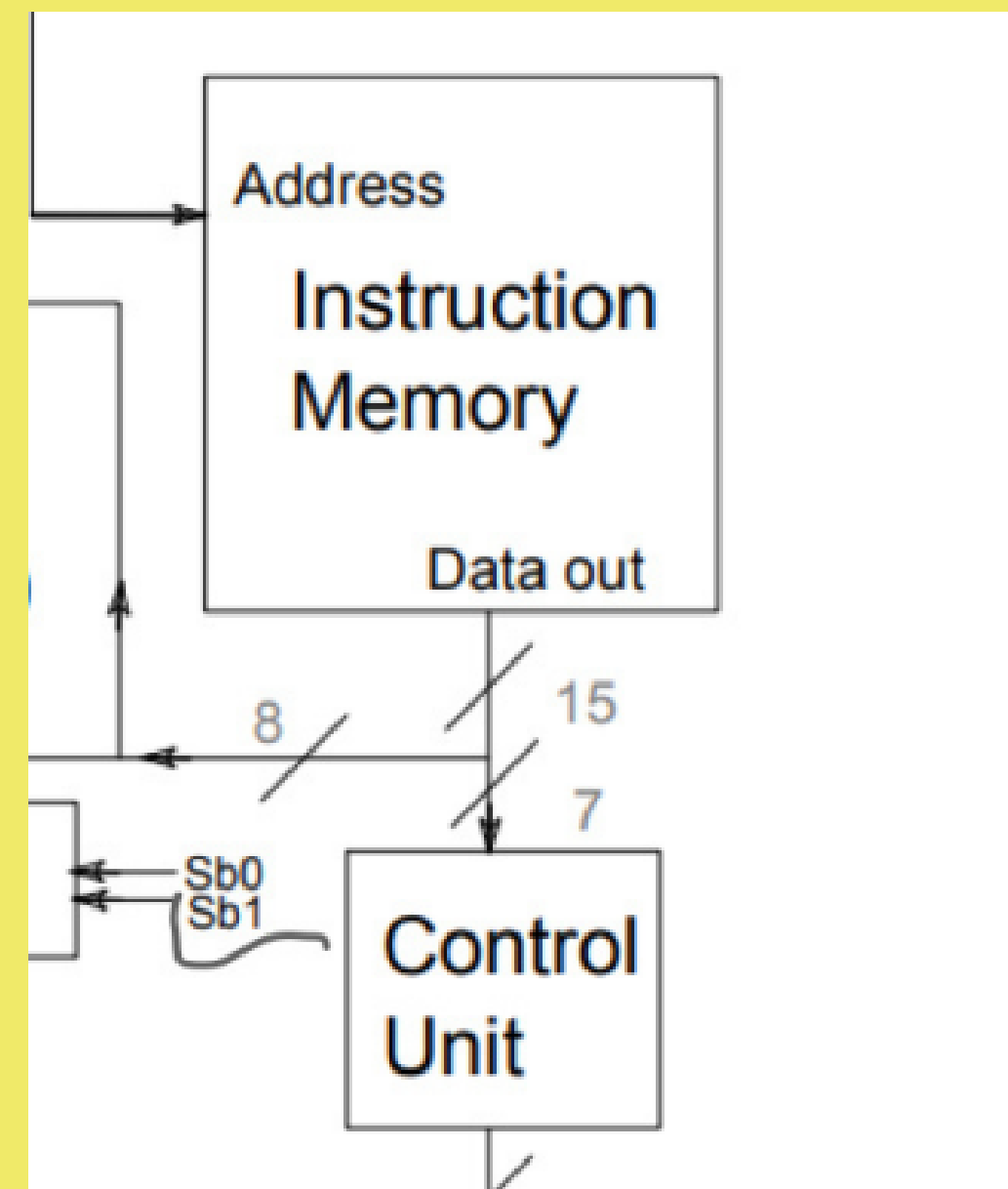


Control Unit

Es una componente encargada de decodificar los opcodes e interpretar las señales de control.

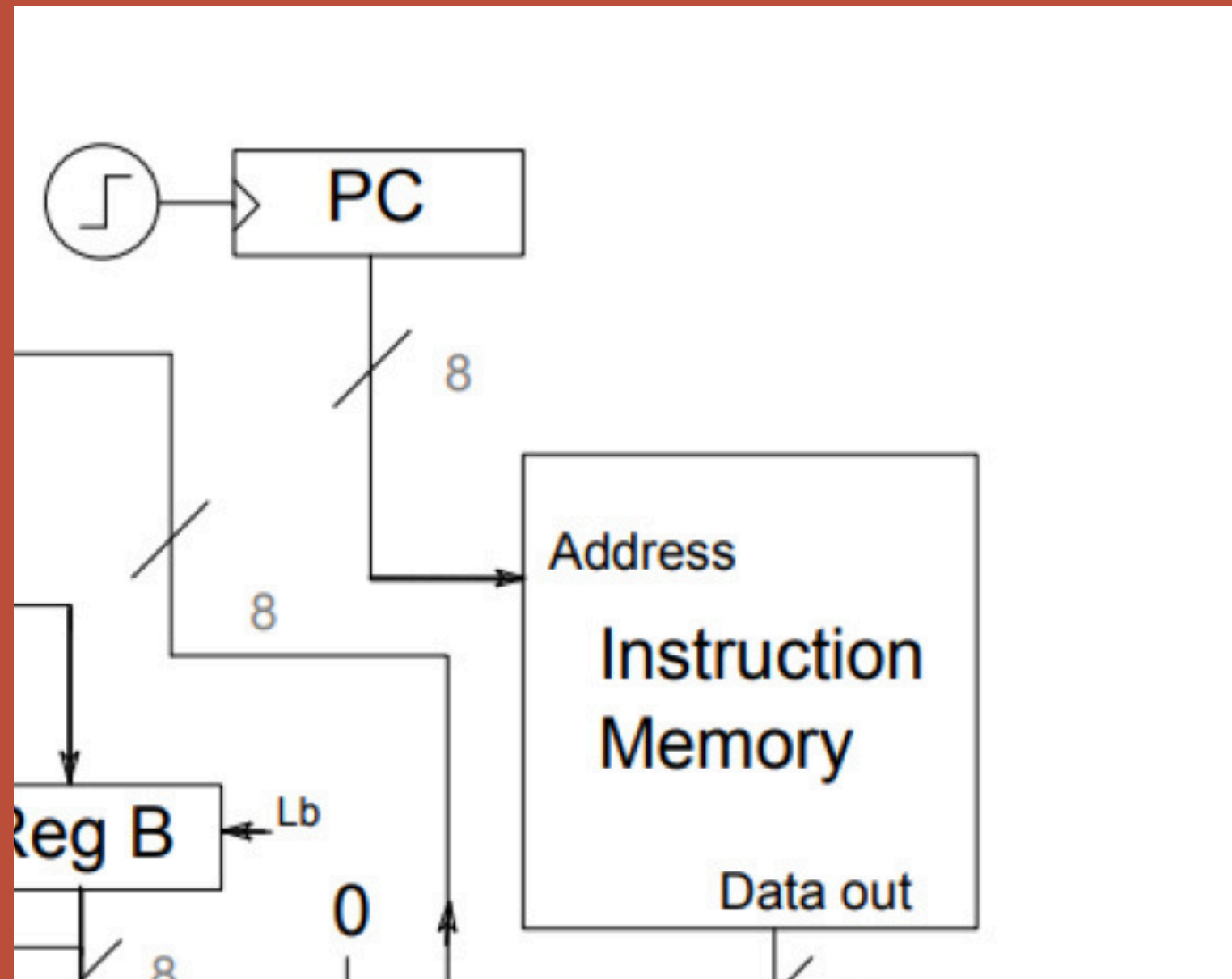
En el caso del computador se puede notar que la salida de la Instruction Memory son 15 bits, donde 7 corresponde a opcode y 8 a un literal

Nota : los opcodes son los códigos de operaciones, se otorgan a cada instrucción para poder decodificarlas



Program Counter

El Program Counter es un contador incremental el cual le indica la dirección de la siguiente instrucción a ejecutar.
(de momento es solo sumar 1 a la instrucción actual ...)



Assembly

¿Qué es?

Es un lenguaje más human-readable, que las señales de control. Esto nos permite entender fácilmente la instrucción que está ejecutando el computador

Ejemplo: MOV A,B

¿Qué hace esta instrucción?



Assembly

Tenemos muchas instrucciones...

Instrucción	L _a	L _b	S _a	S _{b1}	S _{b0}	S ₂	S ₁	S ₀	Operación	Opcode
MOV A, B	1	0	1	0	0	0	0	0	A = B	0000000
MOV B, A	0	1	0	1	0	0	0	0	B = A	0000001
MOV A, Lit	1	0	1	0	1	0	0	0	A = Lit	0000010
MOV B, Lit	0	1	1	0	1	0	0	0	B = Lit	0000011
ADD A, B	1	0	0	0	0	0	0	0	A = A + B	0000100
ADD B, A	0	1	0	0	0	0	0	0	B = A + B	0000101
ADD A, Lit	1	0	0	0	1	0	0	0	A = A + Lit	0000110
ADD B, Lit	0	1	0	0	1	0	0	0	B = A + Lit	0000111
SUB A, B	1	0	0	0	0	0	0	1	A = A – B	0001000
SUB B, A	0	1	0	0	0	0	0	1	B = A – B	0001001
SUB A, Lit	1	0	0	0	1	0	0	1	A = A – Lit	0001010
SUB B, Lit	0	1	0	0	1	0	0	1	B = A – Lit	0001011
AND A, B	1	0	0	0	0	0	1	0	A = A and B	0001100
AND B, A	0	1	0	0	0	0	1	0	B = A and B	0001101
AND A, Lit	1	0	0	0	1	0	1	0	A = A and Lit	0001110
AND B, Lit	0	1	0	0	1	0	1	0	B = A and Lit	0001111

Instrucción	L _a	L _b	S _a	S _{b1}	S _{b0}	S ₂	S ₁	S ₀	Operación	Opcode
OR A, B	1	0	0	0	0	0	1	1	A = A or B	0010000
OR B, A	0	1	0	0	0	0	1	1	B = A or B	0010001
OR A, Lit	1	0	0	0	1	0	1	1	A = A or Lit	0010010
OR B, Lit	0	1	0	0	1	0	1	1	B = A or Lit	0010011
NOT A, A	1	0	0	0	0	1	0	0	A = not A	0010100
NOT B, A	0	1	0	0	0	1	0	0	B = not A	0010101
XOR A, B	1	0	0	0	0	1	0	1	A = A xor B	0010110
XOR B, A	0	1	0	0	0	1	0	1	B = A xor B	0010111
XOR A, Lit	1	0	0	0	1	1	0	1	A = A xor Lit	0011000
XOR B, Lit	0	1	0	0	1	1	0	1	B = A xor Lit	0011001
SHL A, A	1	0	0	0	0	1	1	0	A = SHL A	0011010
SHL B, A	0	1	0	0	0	1	1	0	B = SHL A	0011011
SHR A, A	1	0	0	0	0	1	1	1	A = SHR A	0011100
SHR B, A	0	1	0	0	0	1	1	1	B = SHR A	0011101

Fuente: Clase 5 de la Su :D

Assembly

Direccionamientos

Direccionamiento directo: método de direccionamiento en donde la dirección es un literal.

Ejemplo: MOV(8),A ADD A,(9)

Direccionamiento indirecto :método de direccionamiento en donde la dirección está almacenada en el registro B.

Ejemplo: MOV(B),A ADD A,(B)



Preguntas

Ahora que entendemos mejor los componentes de computador e instrucciones del computador.

Observando el computador...



¿Por qué motivo no está soportada la instrucción:
ADD B, LIT
(en caso general) en el computador básico?

¿Cómo podría implementarse para cualquier literal?



Preguntas

¿Cuál es una instrucción que es soportada por el computador básico pero que no se encuentren en la ISA de este (es decir: que no necesita modificaciones de la arquitectura para implementarla)?

Describa y entregue las señales



Preguntas

¿Cómo se podría implementar en el computador básico la opción de que este avise luego de realizar una operación cuando el resultado es par o impar?



Veamos la
Ayudantía !!!!

