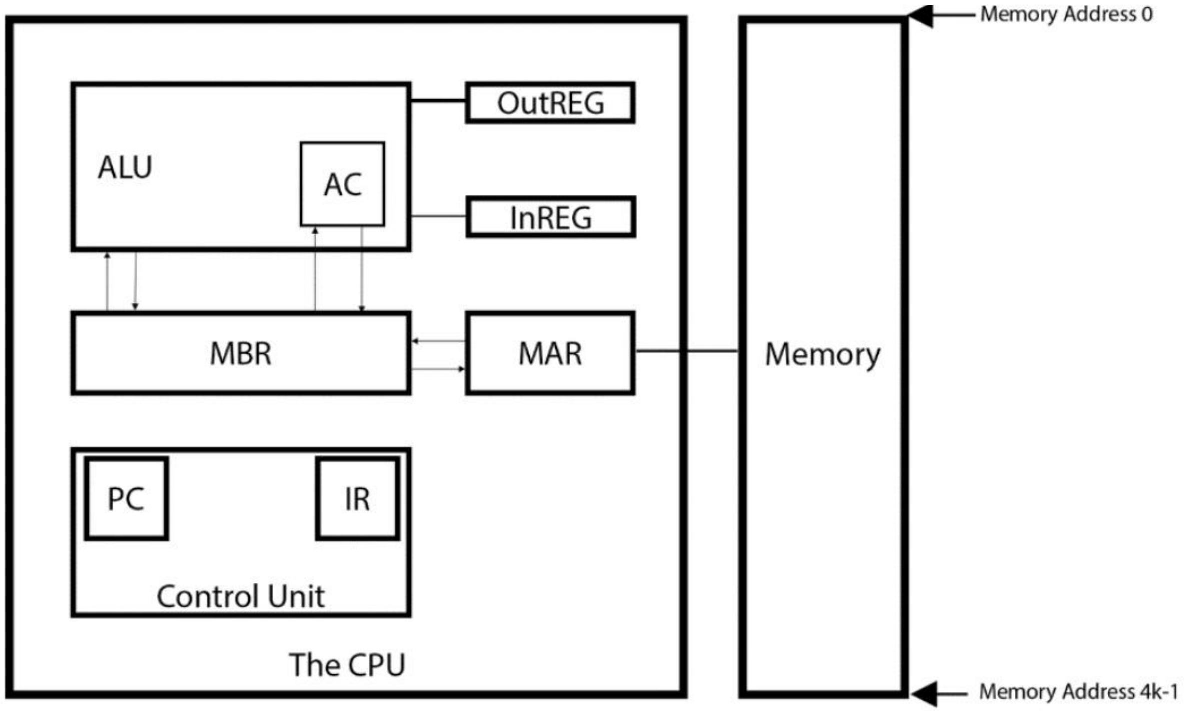
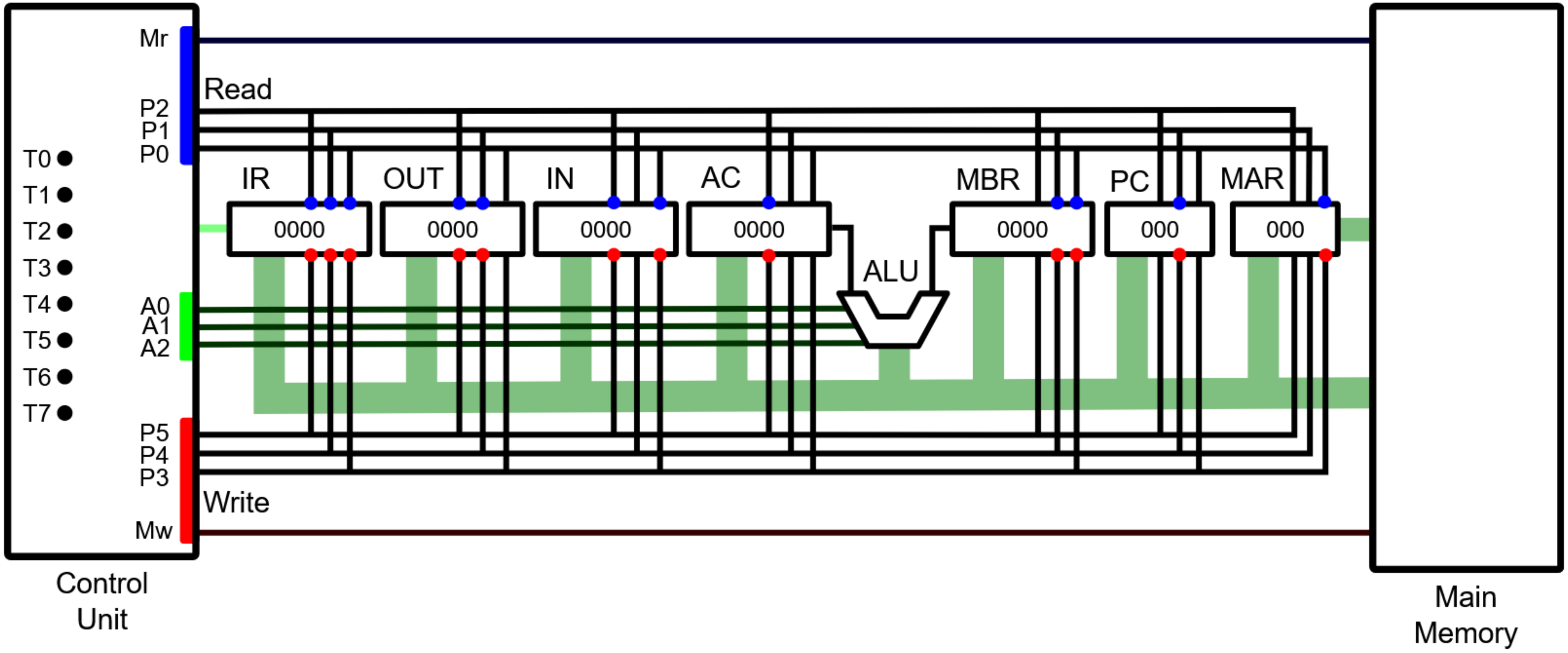
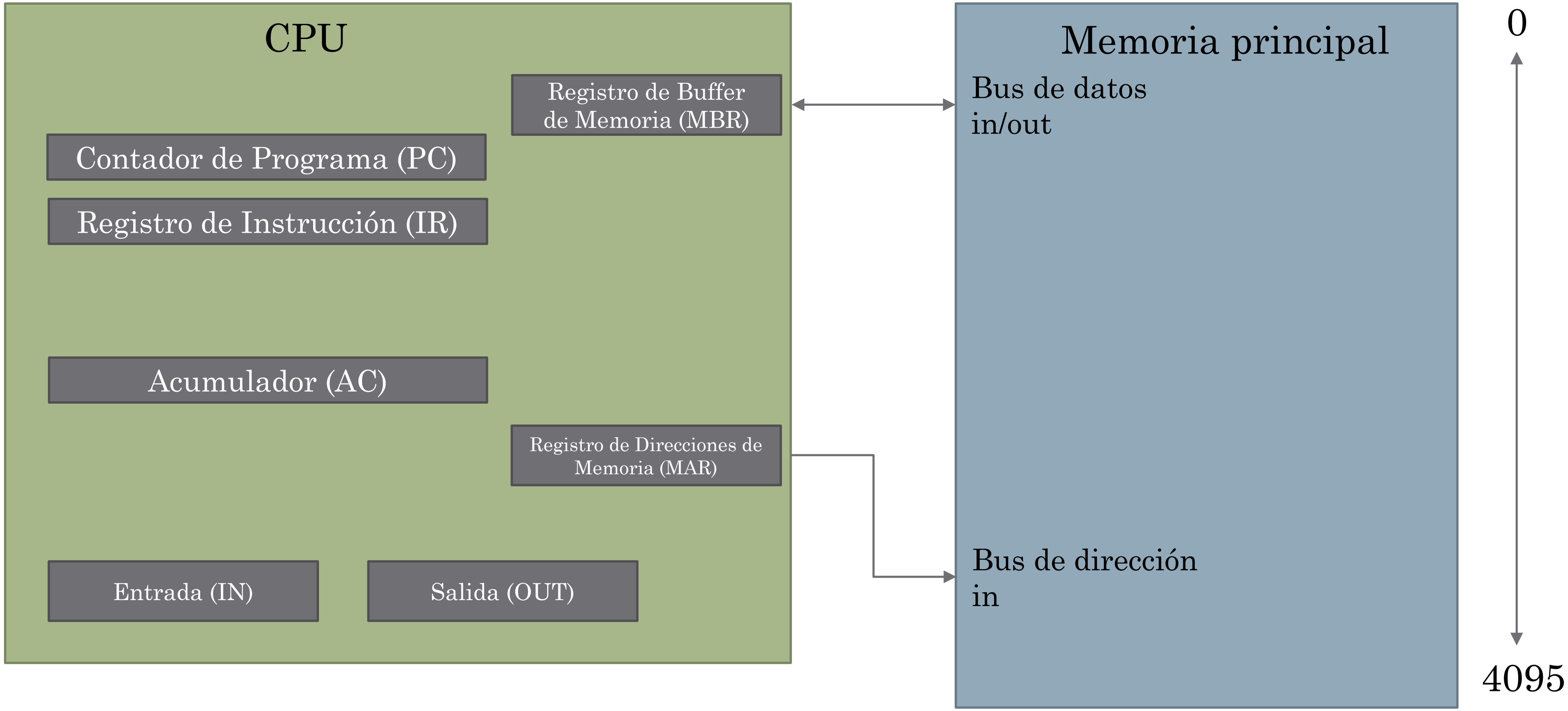
Simulador MARIE




Mr
Read
P2
P1
P0
T0
T1
T2
T3
T4
T5
T6
T7
IR
0000
OUT
0000
IN
0000
AC
0000
ALU
MBR
0000
PC
000
MAR
000
A0
A1
A2
P5
P4
P3
Write
Mw
Control Unit
Main Memory

Registros


CPU
Registro de Buffer de Memoria (MBR)
Contador de Programa (PC)
Registro de Instrucción (IR)
Acumulador (AC)
Registro de Direcciones de Memoria (MAR)
Entrada (IN)
Salida (OUT)
Memoria principal
Bus de datos in/out
Bus de dirección in
0
4095

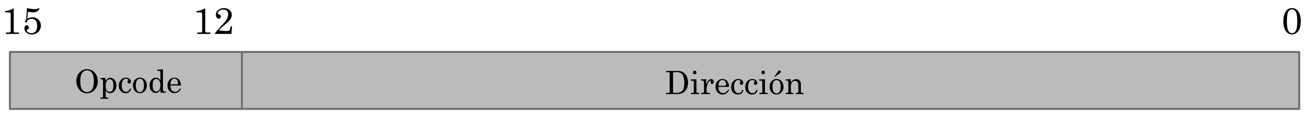
Líneas de control de **Lectura**

Registro	Líneas de control	Código
M[MAR]	Mr	-
MAR	P0	001
PC	P1	010
MBR	P1 P0	011
AC	P2	100
IN	P2 P0	101
OUT	P2 P1	110
IR	P2 P1 P0	111

Líneas de control de **Escritura**

Registro	Líneas de control	Código
M[MAR]	Mw	-
MAR	P3	001
PC	P4	010
MBR	P4 P3	011
AC	P5	100
IN	P5 P3	101
OUT	P5 P4	110
IR	P5 P4 P3	111

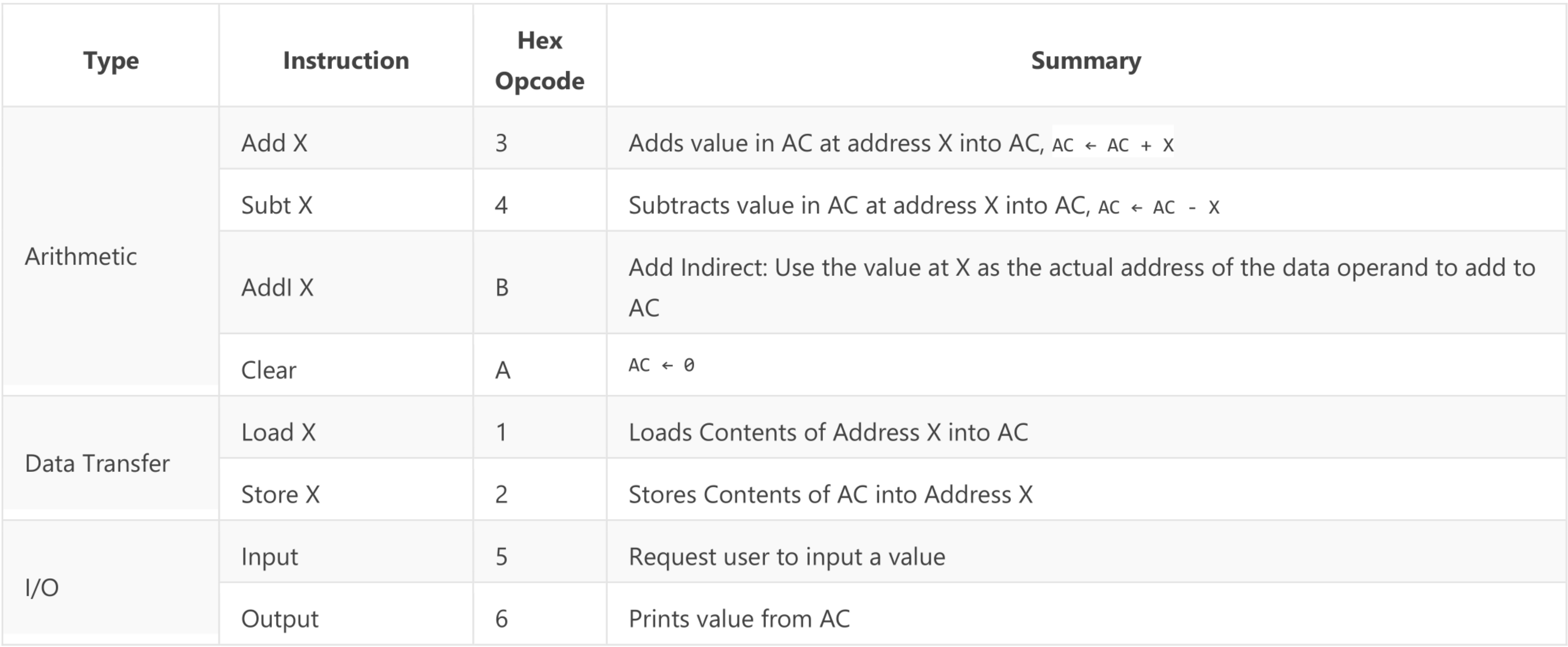
Formato de las instrucciones



Instrucciones

- Aritméticas
- Movimiento de datos
- E/S
- Bifurcaciones
- Subrutinas
- Direccionamiento indirecto

Conjunto de instrucciones



Type	Instruction	Hex Opcode	Summary
Arithmetic	Add X	3	Adds value in AC at address X into AC, `AC ← AC + X`
	Subt X	4	Subtracts value in AC at address X into AC, `AC ← AC - X`
	AddI X	B	Add Indirect: Use the value at X as the actual address of the data operand to add to AC
	Clear	A	`AC ← 0`
Data Transfer	Load X	1	Loads Contents of Address X into AC
	Store X	2	Stores Contents of AC into Address X
I/O	Input	5	Request user to input a value
	Output	6	Prints value from AC

Branch	Jump X	9	Jumps to Address X
	Skipcond (C)	8	Skips the next instruction based on C: if (C) is - 000: Skips if AC < 0 - 400: Skips if AC = 0 - 800: Skips if AC > 0
Subroutine	JnS X	0	Jumps and Store: Stores PC at address X and jumps to X+1
	JumpI X	C	Uses the value at X as the address to jump to
Indirect Addressing	StoreI	E	Stores value in AC at the indirect address. e.g. `StoreI addresspointer` Gets value from addresspointer, stores the AC value into the address
	LoadI	D	Loads value from indirect address into AC e.g. `LoadI addresspointer` Gets address value from addresspointer, loads value at the address into AC
	Halt	7	End the program