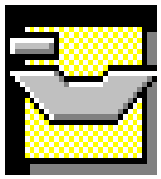


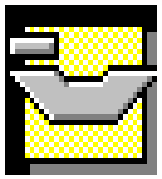
COMPUTERS ORGANIZATION

2ND YEAR COMPUTE SCIENCE MANAGEMENT ENGINEERING

UNIT 1 - INTRODUCTION

JOSÉ GARCÍA RODRÍGUEZ
JOSÉ ANTONIO SERRA PÉREZ

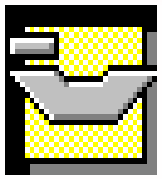
departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Description

Definitions (I)

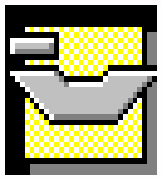
- A computer is:
  - A complex electronic system, made up of a huge number of basic electronic devices
  - A complex hierarchy system



Description

Definitions (II)

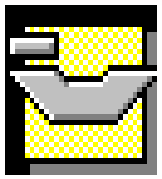
- We can define:
  - ***structure*** as the way in which components are related ones to each others
  - ***function*** as the operation done by each component as part of the structure



Description

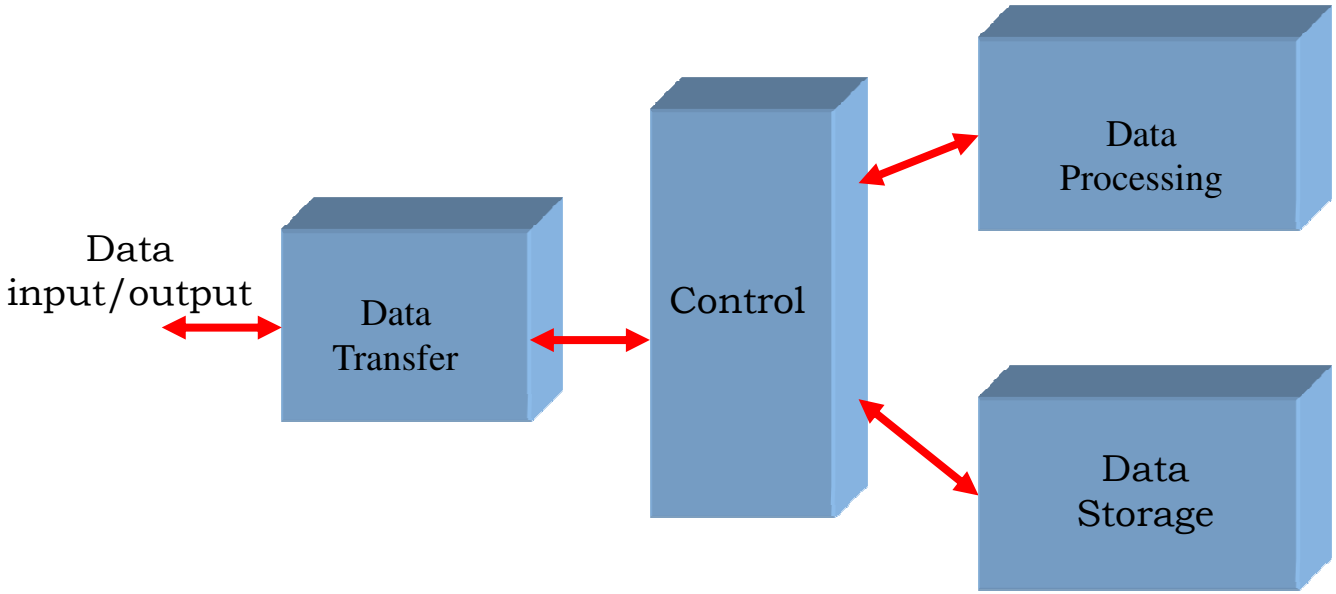
Functions (I)

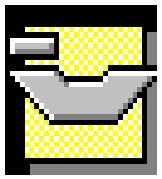
- Computer Basic Functions:
  - Data processing
  - Data storage
  - Data transfer
  - Control



Blocks

Functions (II)

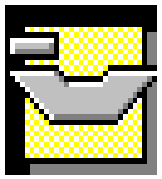




Description

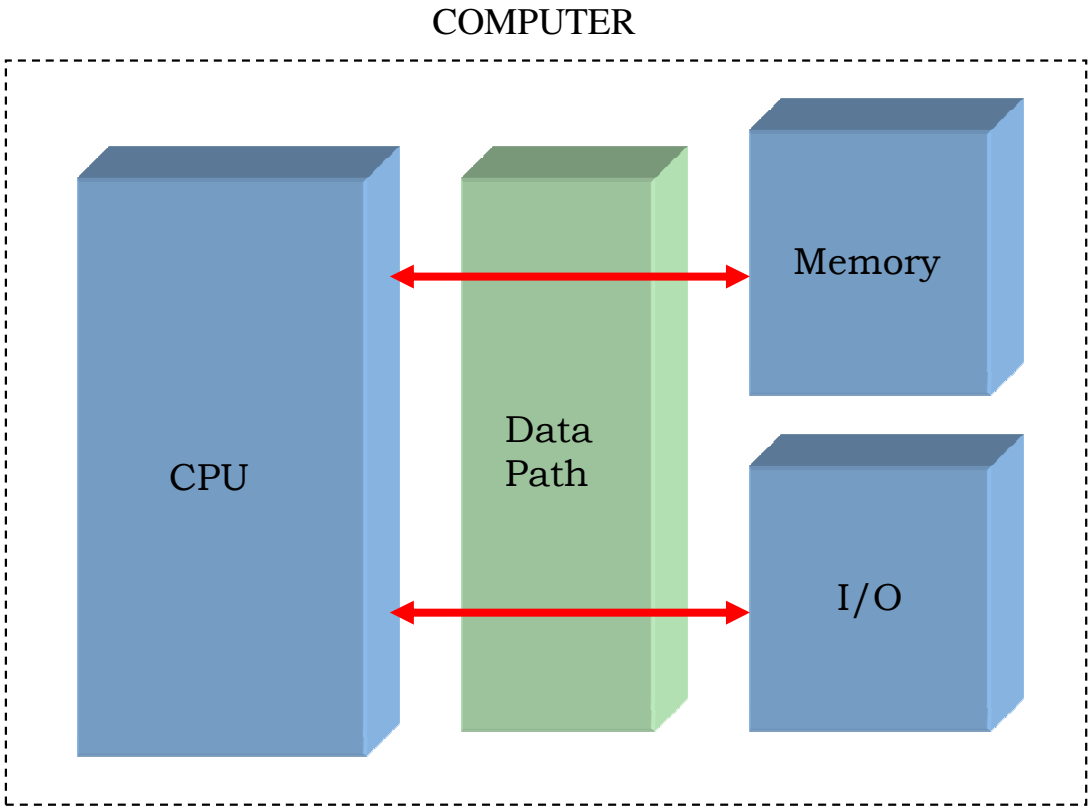
Structures (I)

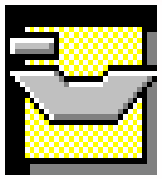
- Computer Basic Structures:
  - Central Processing Unit
  - Main Memory
  - I/O Unit
  - Data path



Blocks

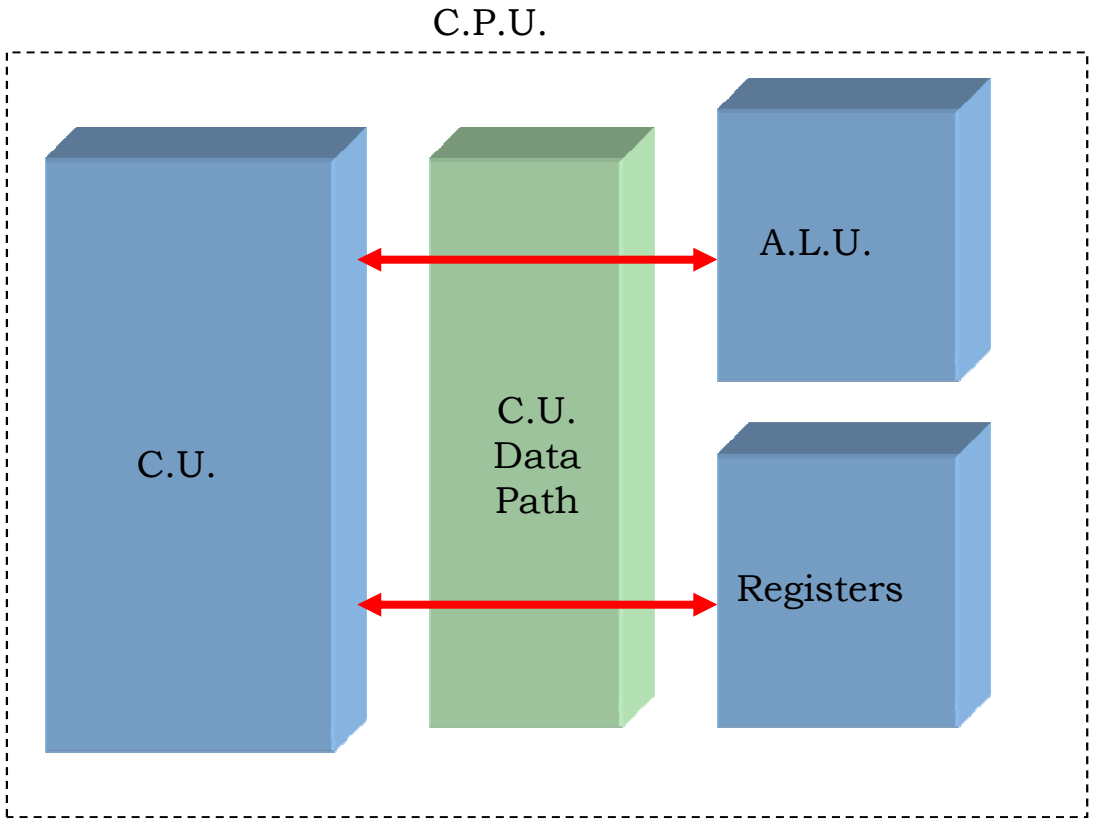
Structures (II)

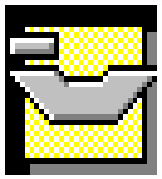




Blocks

Structures (III)

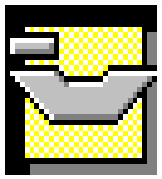




Premises

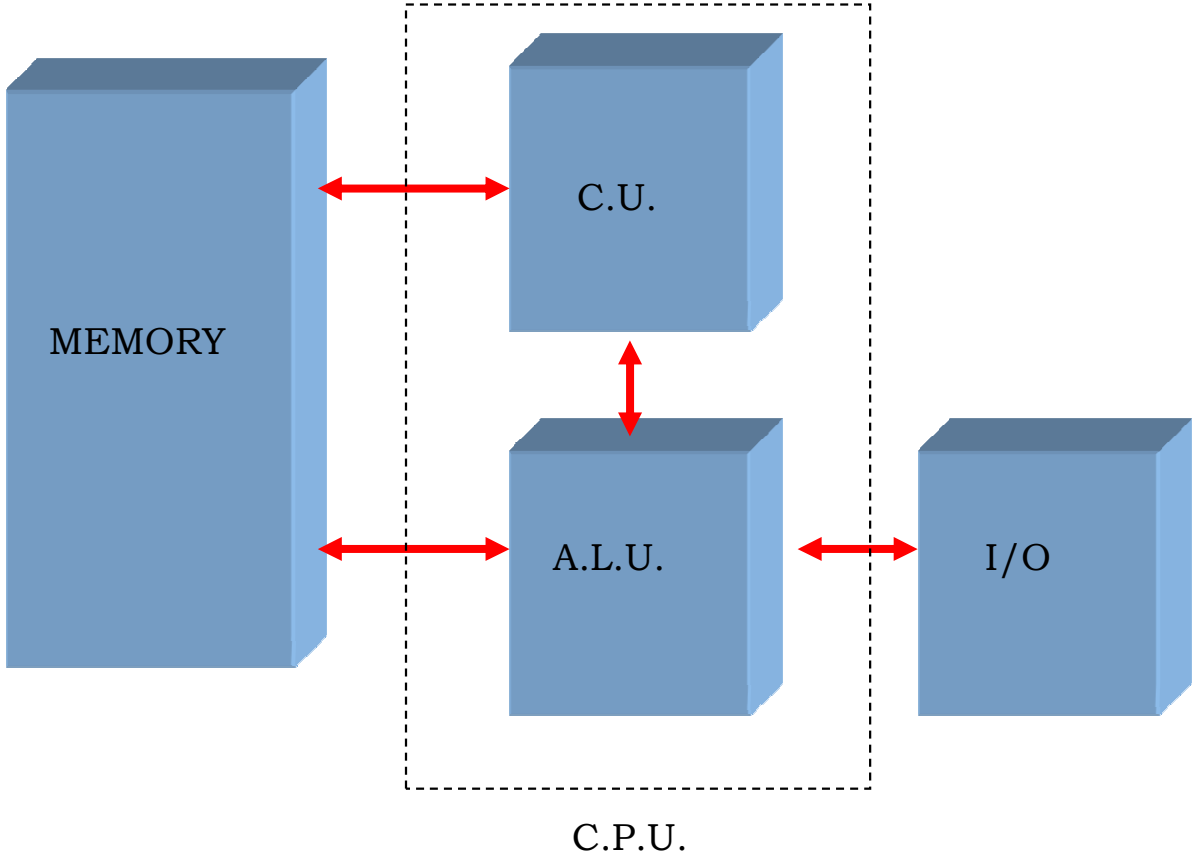
von Neumann Machine (I)

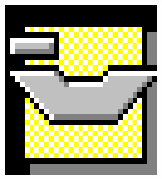
- It is based on three concepts:
  - Only one read/write memory.
  - Access to memory indicating its address.
  - Continuous program execution.



Blocks

von Neumann Machine (II)

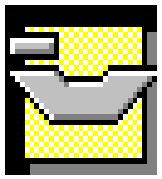




Description

MaNoTaS (I)

- Resources
  - ALU capable of do A/L operations.
  - 64Kbytes memory.
  - Reduced Instructions Set.
  - Four addressing modes.
  - A register bank.
  - State register (Z,C,O,I)



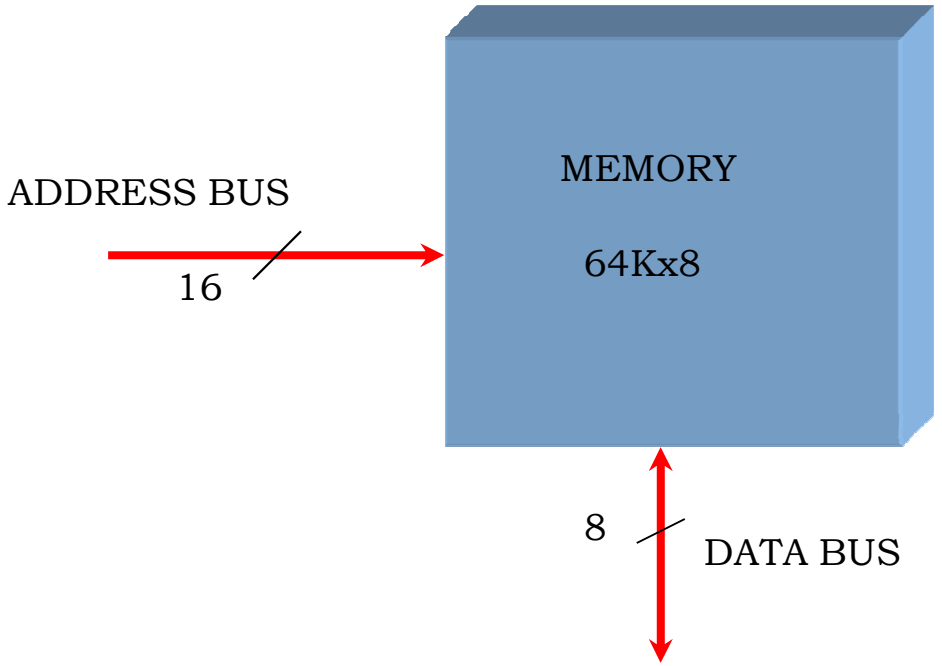
Memory

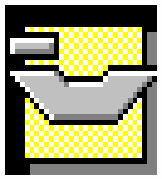
MaNoTaS. (II)

- Capacity 64Kbytes
  - 16 lines for address
  - 8 lines for data
- Two paths for the information
  - Bidirectional for data
  - Unidirectional for instructions

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE

MaNoTaS. (III)

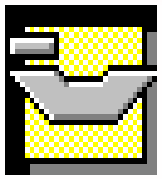




Addressing Modes

MaNoTaS. (IV)

- MaNoTaS has 4 addressing modes
  - Immediate.
  - Direct to memory.
  - Direct to register.
  - Indirect to register.



Addressing Modes

MaNoTaS. (V)

- ## Immediate.

  The data is specified after the operation code. It consists of 2 bytes, one used for the operation code and the other for the data.

- ## Direct to memory.

  The operand is defined using the memory address that store it. This mode consists of 3 bytes, one used by the operation code and the two others used for the address.

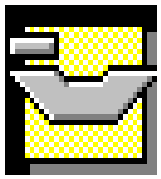
- ## Direct to register.

  The operand is stored in the specified register. This mode consists of 1 byte.

- ## Indirect to register.

  The operand is stored at the address given by the specified register. This mode consists of 1 byte

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE

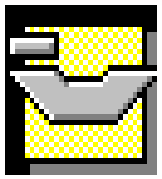


Instructions Set

MaNoTaS. (VI)

Transfer	Arithmetic	Logic	Control	I/O
LDA addr	ADD r1	ANA r1	JMP addr	IN #n
STA addr	ADI data	ANI data	JZ addr	OUT #n
LDAX	INR r1	ORA r1	JO addr	
STAX	DER r1	ORI data	JC addr	
LFA	SUB r1	XRA r1	CALL addr	
SFA	SUI data	XRI data	RET	
MOV r1,r2	CMP r1	CMA	INT #n	
MVI $data_8$,r1	CPI data		IRET	
MVIL lbl_name,r1			CLI	
MVIH lbl_name,r1			STI	
PUSH r1			NOP	
POP r1				
PUSHF				
POPF				

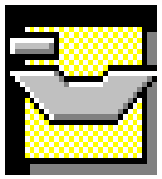
departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Instructions Description

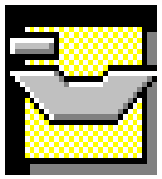
MaNoTaS. (VII)

- **Transfer.**
  - LDA addr A ← M(addr)
  - STA addr M(addr) ← A
  - LDAX A ← M(D-E)
  - STAX M(D-E) ← A
  - MOV r1,r2 r2 ← r1
  - MVI data,r1 r1 ← data



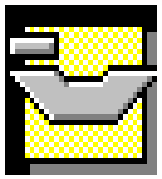
MaNoTaS. (VIII)

- **Arithmetic.**
  - ADD r1 — A ← A+r1
  - ADI data — A ← A + data
  - INR r1 — r1 ← r1 + 1
  - SUB r1 — A ← A – r1
  - SUI data — A ← A – data
  - CMP r1 — A – r1 if A=r1 then FZ=1; if A < r1 ⇒ FC=1
  - CPI data — A – data if A=data then FZ=1; if A < data ⇒FC=1



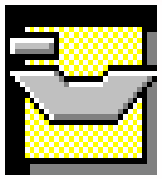
MaNoTaS. (IX)

- **Logic.**
  - ANA r1 — A ← A and r1
  - ANI data — A ← A and data
  - ORA r1 — A ← A or r1
  - ORI data — A ← A or data
  - XRA r1 — A ← A ⊕ r1
  - XRI data — A ← A ⊕ data
  - CMA — A ← C1(A)
- **Flags handle.**
  - LFA — A ← flags
  - SFA — flags ← A



MaNoTaS. (X)

- **Program sequence break.**
  - JMP addr — PC ← addr
  - JZ addr — if FZ=1; PC ← addr
  - JC addr — if FC=1; PC ← addr
  - JO addr — if FO=1; PC ← addr
  - CALL addr — $M(SP) \leftarrow PC_L$; SP ← SP+1
  - — $M(SP) \leftarrow PC_H$; SP ← SP+1 PC ← addr
  - RET — SP ← SP – 1; $PC_H \leftarrow M(SP)$
  - — SP ← SP – 1 $PC_L \leftarrow M(SP)$
  - IRET — SP ← SP – 1; $PC_H \leftarrow M(SP)$;
  - — SP ← SP – 1; $PC_L \leftarrow M(SP)$; I← 1

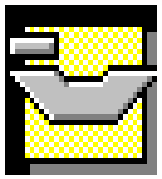


Instructions Encoding

MaNoTaS. (X)

- One byte wide

MOV A, A 44h	MOV B, A 00h	MOV C, A 01h
MOV A, B 40h	MOV B, B 04h	MOV C, B 05h
MOV A, C 41h	MOV B, C 08h	MOV C, C 09h
MOV A, D 42h	MOV B, D 0Ch	MOV C, D 0Dh
MOV A, C 43h	MOV B, E 10h	MOV C, E 11h
MOV D, A 02h	MOV E, A 03h	
MOV D, B 06h	MOV E, B 07h	
MOV D, C 0Ah	MOV E, C 0Bh	
MOV D, D 0Eh	MOV E, D 0Eh	
MOV D, E 12h	MOV E, E 13h	

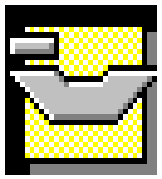


Instructions Encoding

MaNoTaS. (XI)

- One byte wide

LDAX	B0h	STAX	90h		
STI	AAh	LFA	81h		
CLI	ABh	SFA	82h		
ADD A	45h	SUB A	46h	CMP A	47h
ADD B	30h	SUB B	18h	CMP B	1Ch
ADD C	31h	SUB C	19h	CMP C	1Dh
ADD F	32h	SUB D	1Ah	CMP D	1Eh
ADD E	33h	SUB E	1Bh	CMP E	1Fh
INR A	4Bh	DER A	A0h	CMA	80h
INR B	2Ch	DER B	A1h		
INR C	2Dh	DER C	A2h		
INR D	2Eh	DER D	A3h		
INR E	2Fh	DER E	A4h		

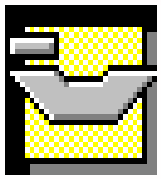


Instructions Encoding

MaNoTaS. (XII)

- One byte wide

ANA A	48h	ORA A	49h	XRA A	4Ah
ANA B	20h	ORA B	24h	XRA B	28h
ANA C	21h	ORA C	25h	XRA C	29h
ANA D	22h	ORA D	26h	XRA D	2Ah
ANA E	23h	ORA E	27h	XRA E	2Bh
PUSH A	55h	POP A	5Ah	PUSHF	50h
PUSH B	56h	POP B	5Bh	POPF	51h
PUSH C	57h	POP C	5Ch	RET	7Bh
PUSH D	58h	POP D	5Dh	IRET	7Ch
PUSH E	59h	POP E	5Eh	NOP	FFh



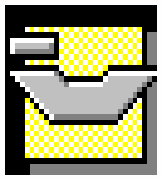
Instructions Encoding

MaNoTaS. (XIII)

- Two bytes wide

MVI data, A	64h data	MVI data, B	60h data
MVI data, C	61h data	MVI data, D	62h data
MVI data, E	63h data	ADI data	35h data
SUI data	36h data	CPI data	37h data
ANI data	68h data		
ORI data	69h data		
XRI data	6Ah data		
INT #n	54h interruptNumber		
IN #n	52h portNumber		
OUT #n	53h portNumber		

departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE

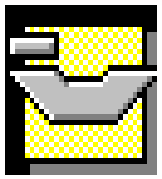


Instructions Descriptions

MaNoTaS. (XIV)

- Three bytes wide

MVI addr, SP	65h	dL	dH
LDA addr	70h	dL	dH
STA addr	71h	dL	dH
JMP addr	74h	dL	dH
JZ addr	72h	dL	dH
JC addr	73h	dL	dH
JO addr	75h	dL	dH
CALL addr	7Ah	dL	dH

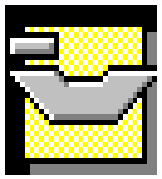


A.L.U.

MaNoTaS. (XV)

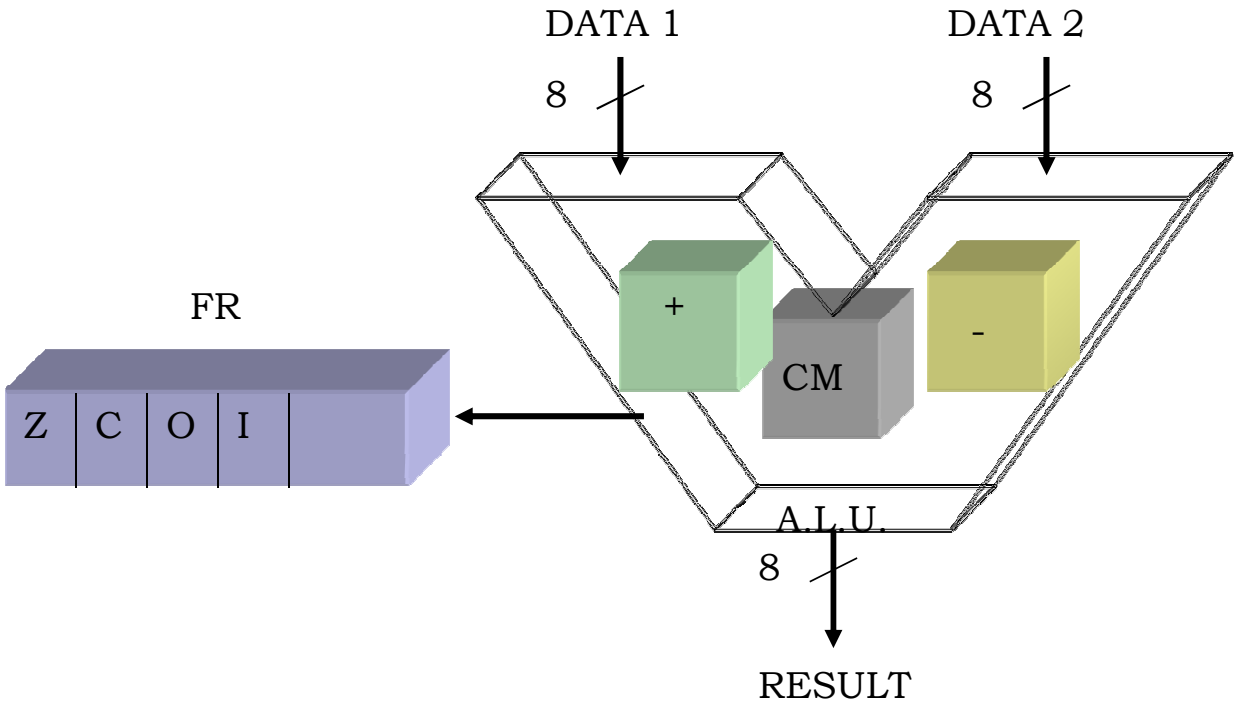
- It performs the following operations
  - Addition, subtraction, comparison
  - And, Or, Xor, Not, Increment and Decrement
- Status register has
  - Zero (Z). It is set to 1 when the last ALU operation is equal to zero.
  - Carry (C). It is set to 1 when the operands' two most significant bits addition causes a carry on.
  - Interruption (I). It is set to 1 to enable interruptions.
  - Overflow (O). It is set to 1 when the last ALU operation is greater than the maximum number allowed for the specified word width.

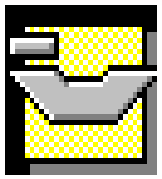
departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



A.L.U.

MaNoTaS. (XVI)

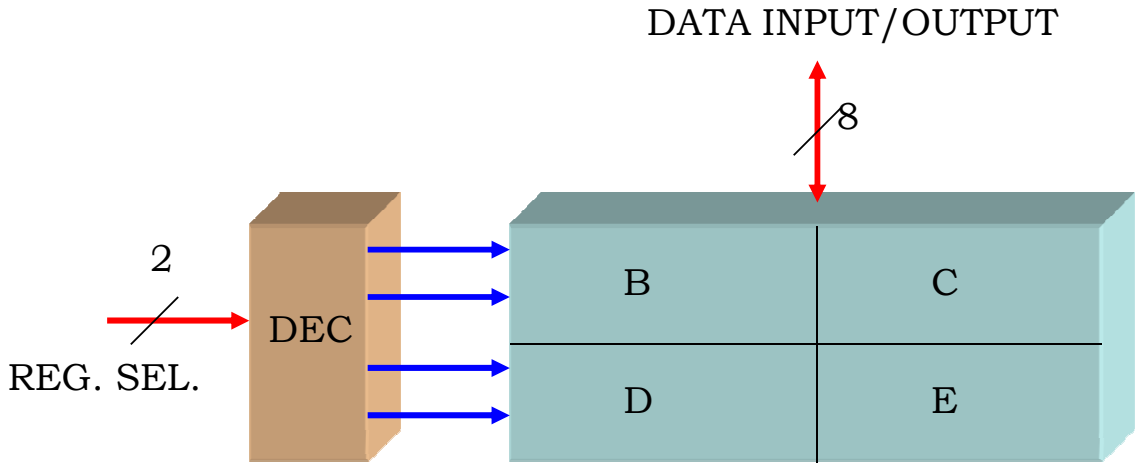


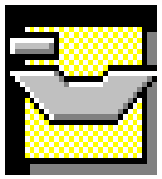


Registers Bank

MaNoTaS. (XVII)

- The instructions definition forces to provide a register bank in MaNoTaS.



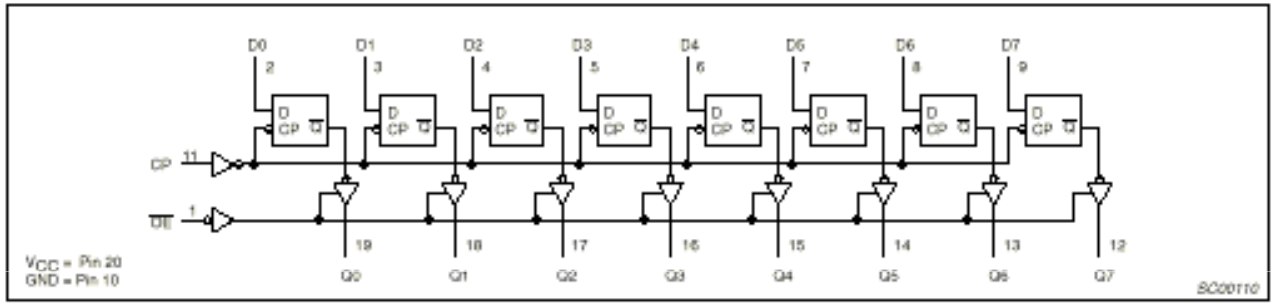


Registers Bank

MaNoTaS. (XVII)

- 74ALS574 Specifications

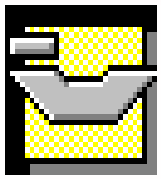
LOGIC DIAGRAM – 74ALS574A



FUNCTION TABLE – 74ALS574A

INPUTS			OUTPUTS REGISTER	INTERNAL	OPERATING MODE
$\overline{OE}$	CP	Dn		Q0 – Q7	
L	↑	l	L	L	Latch and read register
L	↑	h	H	H	
L	↥	X	NC	NC	Hold
H	↥	X	NC	Z	Disable outputs
H	↑	Dn	Dn	Z	

H = High-voltage level
h = High state must be present one setup time before the Low-to-High clock transition
L = Low-voltage level
l = Low state must be present one setup time before the Low-to-High clock transition
NC = No change
X = Don't care
Z = High impedance "off" state
↑ = Low-to-High clock transition
↥ = Not Low-to-High clock transition

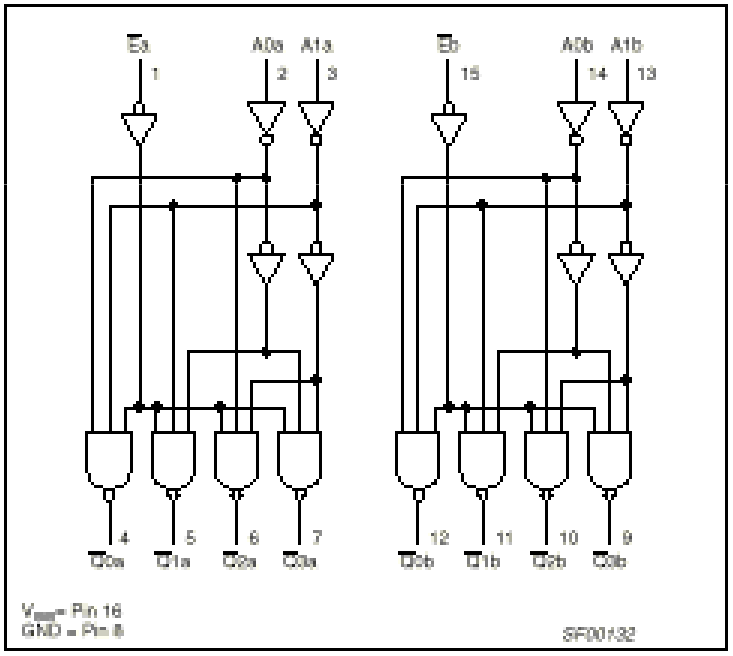


Registers Bank

MaNoTaS. (XVII)

- 74F139 Specifications

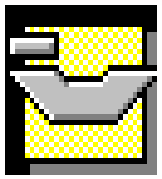
LOGIC DIAGRAM



FUNCTION TABLE

INPUTS			OUTPUTS			
$\overline{E}$	A0	A1	$\overline{O}0$	$\overline{O}1$	$\overline{O}2$	$\overline{O}3$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

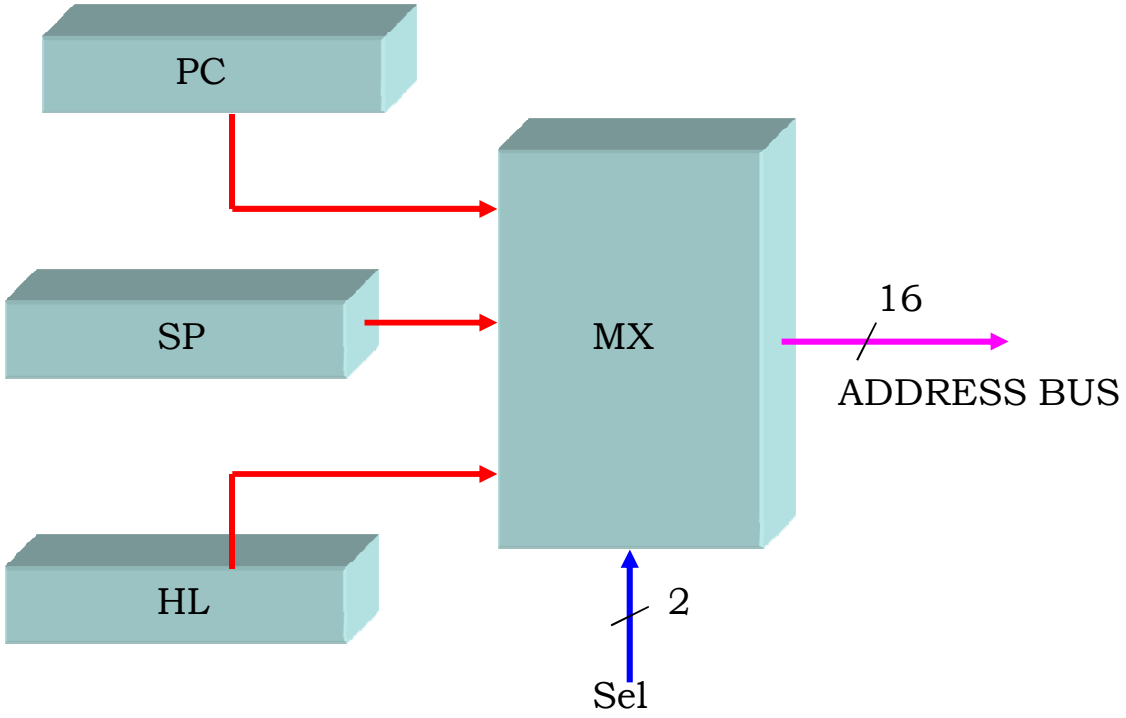
NOTES:
H = High voltage level
L = Low voltage level
X = Don't care

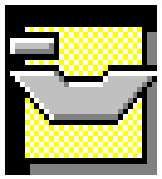


Address Selection

MaNoTaS. (XVIII)

- To select the sources for the address bus, a multiplexor is needed.



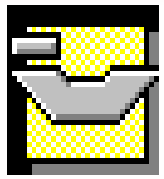


C.U.

MaNoTaS. (XIX)

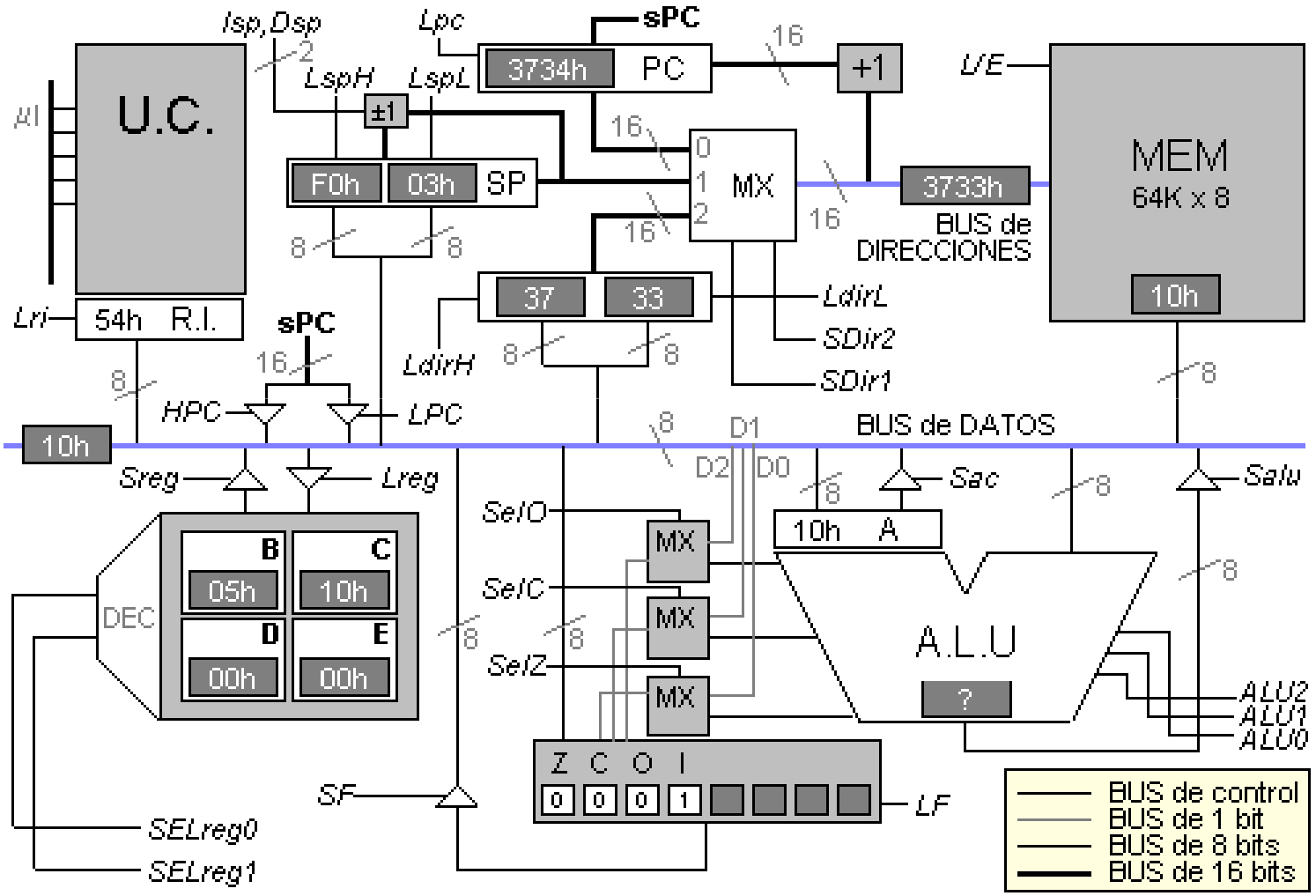
- It does the following operations:
  - It loads the instructions, that are located in the memory, into a register which is named as Instruction Register.
  - It controls a register named as Program Counter, which control what instruction will be executed.
  - It decodes the instruction.
  - It controls the instructions execution and the peripherals communication.

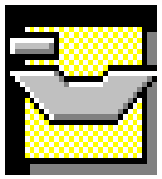
departamento de tecnología informática y computación
UNIVERSIDAD DE ALICANTE



Data Path

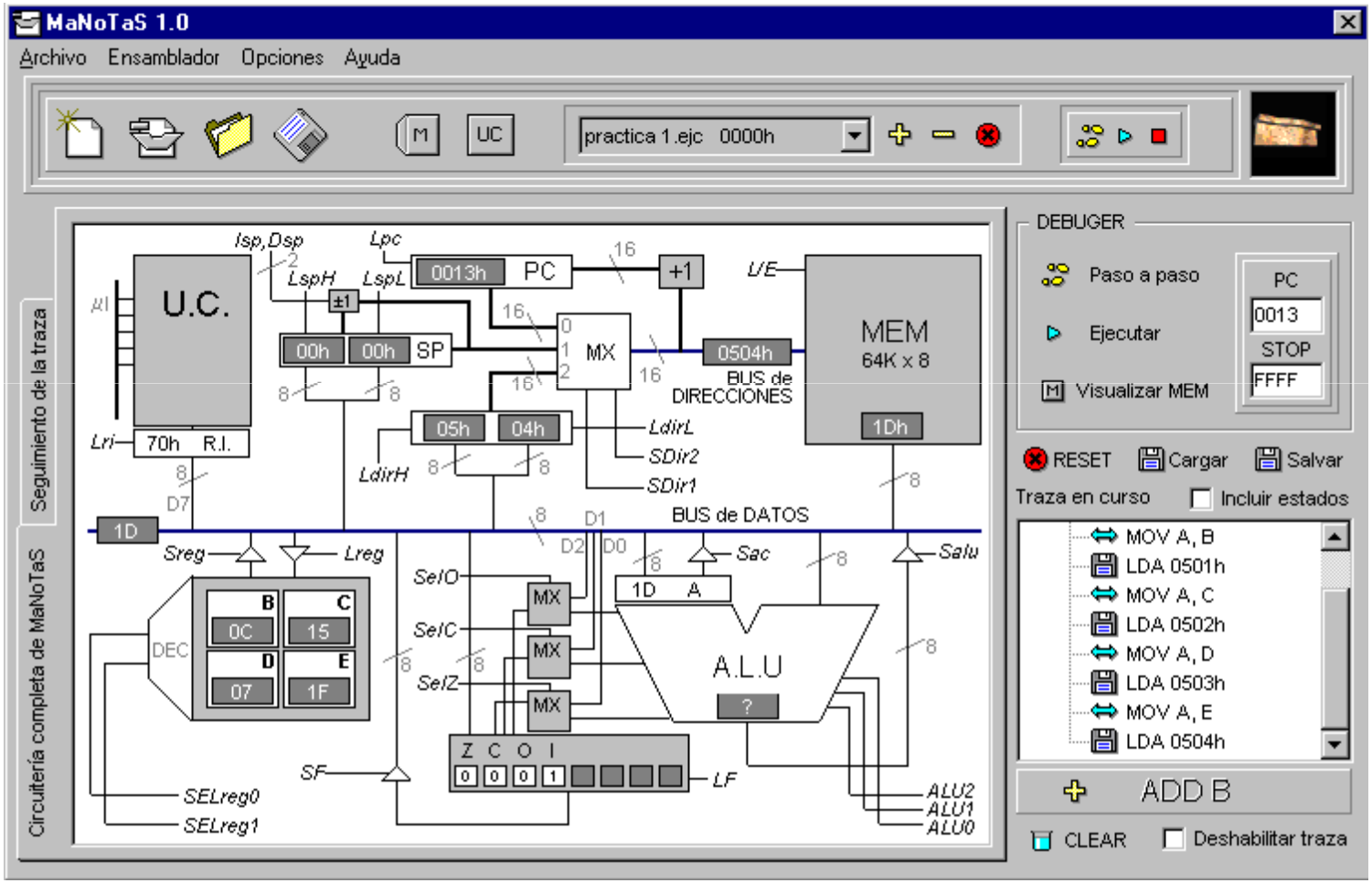
MaNoTaS. (XX)

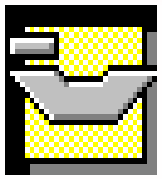




Simulator

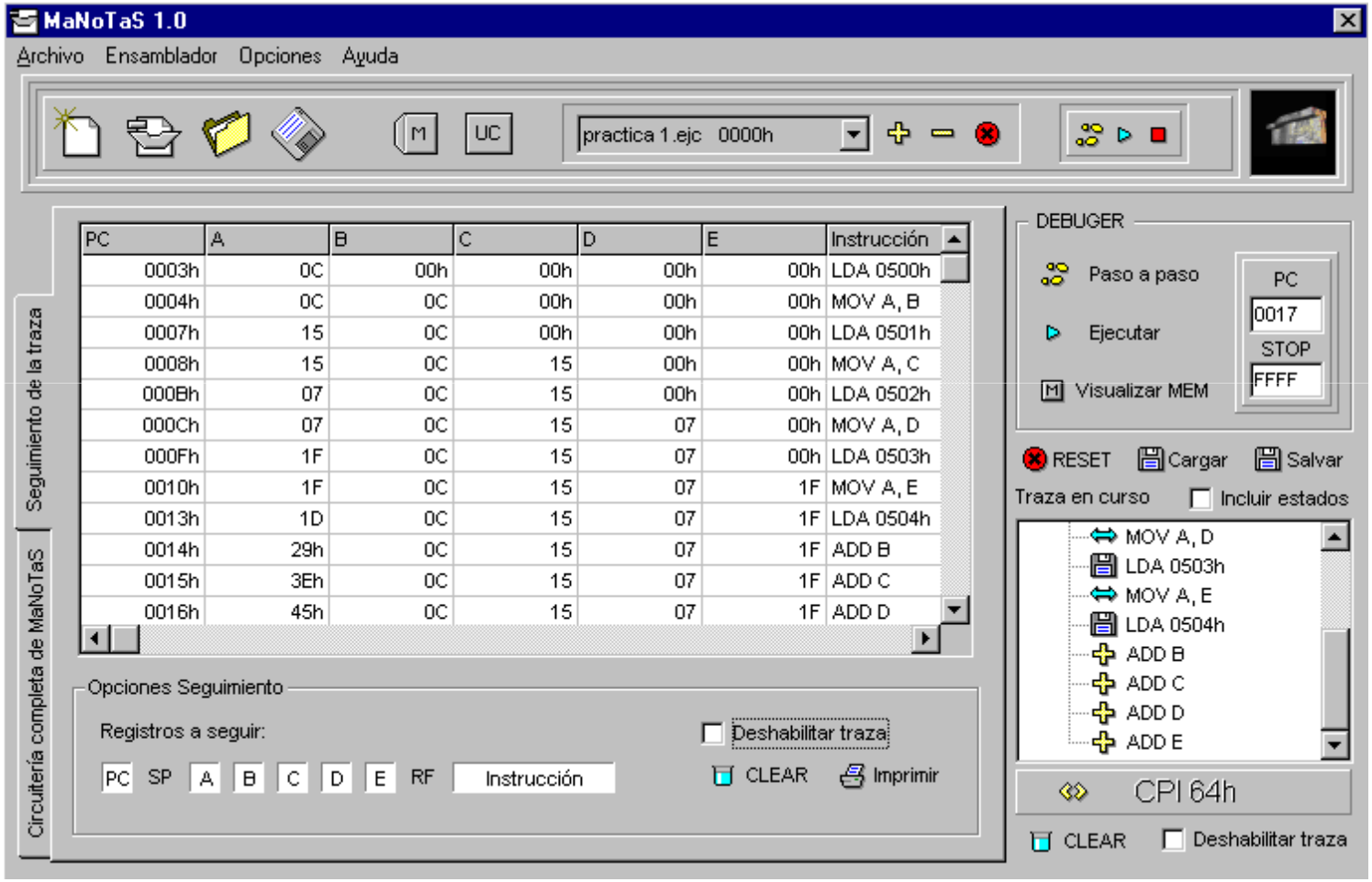
MaNoTaS Simulator. (I)

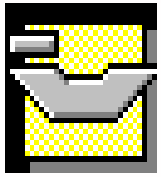




Simulator Trace

MaNoTaS Simulator. (II)





Simulator Editor

MaNoTaS Simulator. (III)