CAMINO DE DATOS

(DATAPATH)

CAMINO DE DATOS

(DATAPATH)

MEMORIA DE INSTRUCCIONES

MEMORIA DE DATOS

ARCHIVO DE REGISTROS

UNIDAD ARITMETICO LOGICA (ALU)

CONTADOR DE PROGRAMA

MULTIPLEXORES

BUSES

DATAPATH

MEMORIA DE INSTRUCCIONES

TENDENCIAS DE LA FILOSOFIA RISC

1.- HACER MENOS COSAS POR INSTRUCCION

2.-HACER INSTRUCCIONES DE LONGITUD FIJA

A.- PROCESADORES LOAD-STORE

B.- MEMORIA DE INSTRUCCIONES ARREGLOS DE VECTORES DE 8 BIT

C.- MEMORIA DE DATOS ARREGLOS DE VECTORES DE 8 BIT

DATAPATH

MEMORIA DE INSTRUCCIONES

CONCEPTOS PRELIMINARES BÁSICOS

BIT

NIBBLE

BYTE

HALFWORD

WORD

DOUBLEWORD

QUADWORD

DATAPATH

MEMORIA DE INSTRUCCIONES

LA MEMORIA DE INSTRUCCIONES ESTA ALMACENADA EN ARREGLOS DE BYTE

Para el modelo MIPS32 las instrucciones son de 32 bit

PC	Byte de n-ésima instrucción	
00000	BYTE 0	Primera Instrucción
00001	BYTE 1	
00010	BYTE 2	
00011	BYTE 3	
00100	BYTE 0	Segunda Instrucción
00101	BYTE 1	
00110		
00111		
01000		
01001		

INST0 <= (BYTE0)&(BYTE1)&(BYTE2)&(BYTE3)

EL CONTADOR DE PROGRAMA, DE AHORA EN ADELANTE “PC” DEBERÀ INCREMENTARSE EN 4 CADA VEZ QUE SE LEA UNA INSTRUCCIÓN

UNA INSTRUCCIÓN O UN DATO

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

Para el modelo MIPS32 las instrucciones son de 32 bit

Para leer la siguiente instrucción el contador de programa se incrementa en 4

PC	Byte de n-ésima instrucción	
00000		Primera Instrucción
00001		
00010		
00011		
00100		Segunda Instrucción
00101		
00110		
00111		
01000		
01001		

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

Para el modelo MIPS32 las instrucciones son de 32 bit

PC	Byte de n-ésima instrucción	
00000		Primera Instrucción
00001		
00010		
00011		
00100		Segunda Instrucción
00101		
00110		
00111		
01000		
01001		

Para leer la siguiente instrucción el contador de programa se incrementa en 4

Implementar un módulo de lectura.

Por cada valor del PC el módulo deberá contar con un incremento interno para leer cuatro localidades de memoria

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

Para el modelo MIPS32 las instrucciones son de 32 bit

PC	Byte de n-ésima instrucción	
00000	Byte 0	Primera Instrucción
00001	Byte 1	
00010	Byte 2	
00011	Byte 3	
00100	Byte 0	Segunda Instrucción
00101	Byte 1	
00110	Byte 2	
00111	Byte 3	
01000		
01001		

Instruction Word 0			
Half Word A		Half Word B	
Byte 0	Byte 1	Byte 2	Byte 3

Big – endian format

INST0 <= (BYTE0)&(BYTE1)&(BYTE2)&(BYTE3)

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

3CM12: continuaremos el lunes 06 de dic 2021

Para el modelo MIPS32 las instrucciones son de 32 bit

PC	Byte de n-ésima instrucción	
00000	Byte 0	Primera Instrucción
00001	Byte 1	
00010	Byte 2	
00011	Byte 3	
00100	Byte 0	Segunda Instrucción
00101	Byte 1	
00110	Byte 2	
00111	Byte 3	
01000		
01001		

Instruction Word 0			
Half Word B		Half Word A	
Byte 3	Byte 2	Byte 1	Byte 0
31 to 24	23 to 16	15 to 8	7 to 0

Little – endian format

RE32INST <= signalData(PC + 3)&signalData(PC + 2)&

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

3CM12: continuaremos el lunes 06 de dic 2021

Para el modelo MIPS32 las instrucciones son de 32 bit

PC	Byte de n-ésima instrucción	
00000	Byte 0	Primera Instrucción
00001	Byte 1	
00010	Byte 2	
00011	Byte 3	
00100	Byte 0	Segunda Instrucción
00101	Byte 1	
00110	Byte 2	
00111	Byte 3	
01000		
01001		

Instruction Word 0			
Half Word B		Half Word A	
Byte 3	Byte 2	Byte 1	Byte 0

31 to 24 23 to 16 15 to 8 7 to 0

Little – endian format

signalData(PC) <= INST0(7 downto 0)

signalData(PC + 1) <= INST0(15 downto 8)

signalData(PC + 2) <= INST0(23 downto 16)

signalData(PC + 3) <= INST0(31 downto 24)

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

REFERENCIAS

ARM7EJ-S (rEV 1). Technical Reference Manual. Copyright © 2001 ARM Limited. All rights reserved. ARM DDI 0214B DDI0214.boom.pdf, page 2-4, section 2.3.2

Intel. 64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf, página 1-4 Vol. 2A
section 1.3.1

Patterson and Hennessy. 5th Edition. Appendix A. Page A-43

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

TIPO R

OPCODE	rs	rt	rd	shamt	funct
31 downto 26	25 downto 21	20 downto 16	15 downto 11	10 downto 6	5 downto 0
6bit	5bit	5bit	5bit	5bit	6bit

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

FORMATO INSTRUCCIONES MIPS

OPCODE	rs	rt	rd	shamt	funct
31 downto 26	25 downto 21	20 downto 16	15 downto 11	10 downto 6	5 downto 0
6bit	5bit	5bit	5bit	5bit	6bit

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

TIPO I

(Inmediato)

OPCODE	rs	rt	Immediate
31 downto 26	25 downto 21	20 downto 16	15 downto 0

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

TIPO J

(Jump)

OPCODE	Tarjet
31 downto 26	25 downto 0

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

Ciclo de Instrucción

Etapas de una Instrucción (Segmentación de una Instrucción)

Captar instrucción (Fetch Instruction, **FI**).- Leer la supuesta siguiente instrucción instrucción en el buffer de instrucción.

Decodificar instrucción (Decode instrucction, **DI**).- Determinar el código de operación y los campos de operandos.

Calcular operandos (Calculate Operands, **CO**).- Calcula la dirección efectiva de cada operando fuente. Pa ello se usa alguno de los modos de direccionamientos vistos previamente.

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

Ciclo de Instrucción

Etapas de una Instrucción (Segmentación de una Instrucción)

Captar operandos (Fetch Operands, **FO**).- Cargar en Buffer de ALU, los operandos.

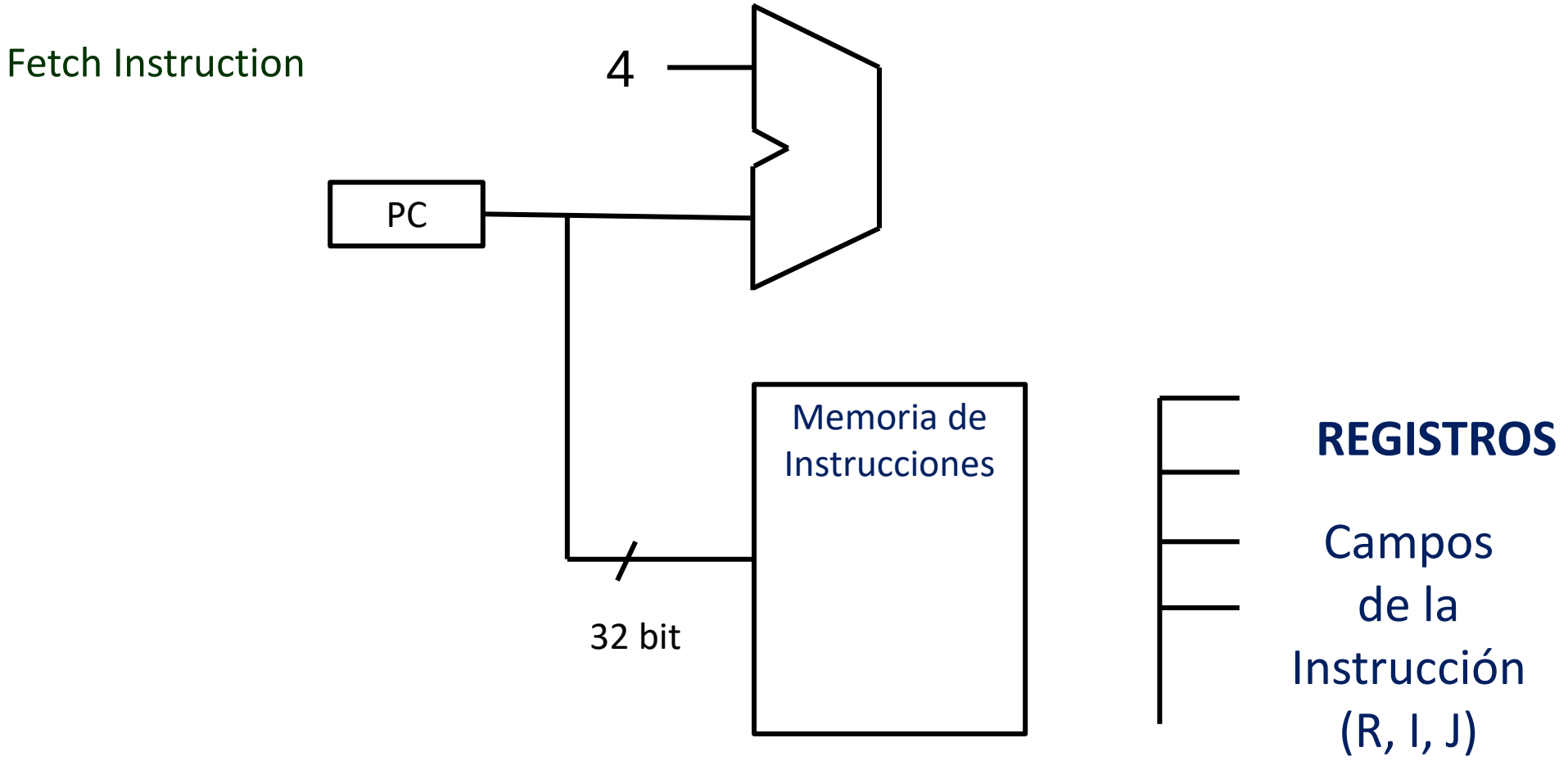
Ejecutar Instrucción (Execute Instrucction, **EI**).- Realizar la operación indicada y almacenar el resultado, si lo hay en la posición del operando destino.

Escribir operando (Write Operand, **WO**).- Almacenar el resultado en memoria

DATAPATH

MEMORIA DE INSTRUCCIONES

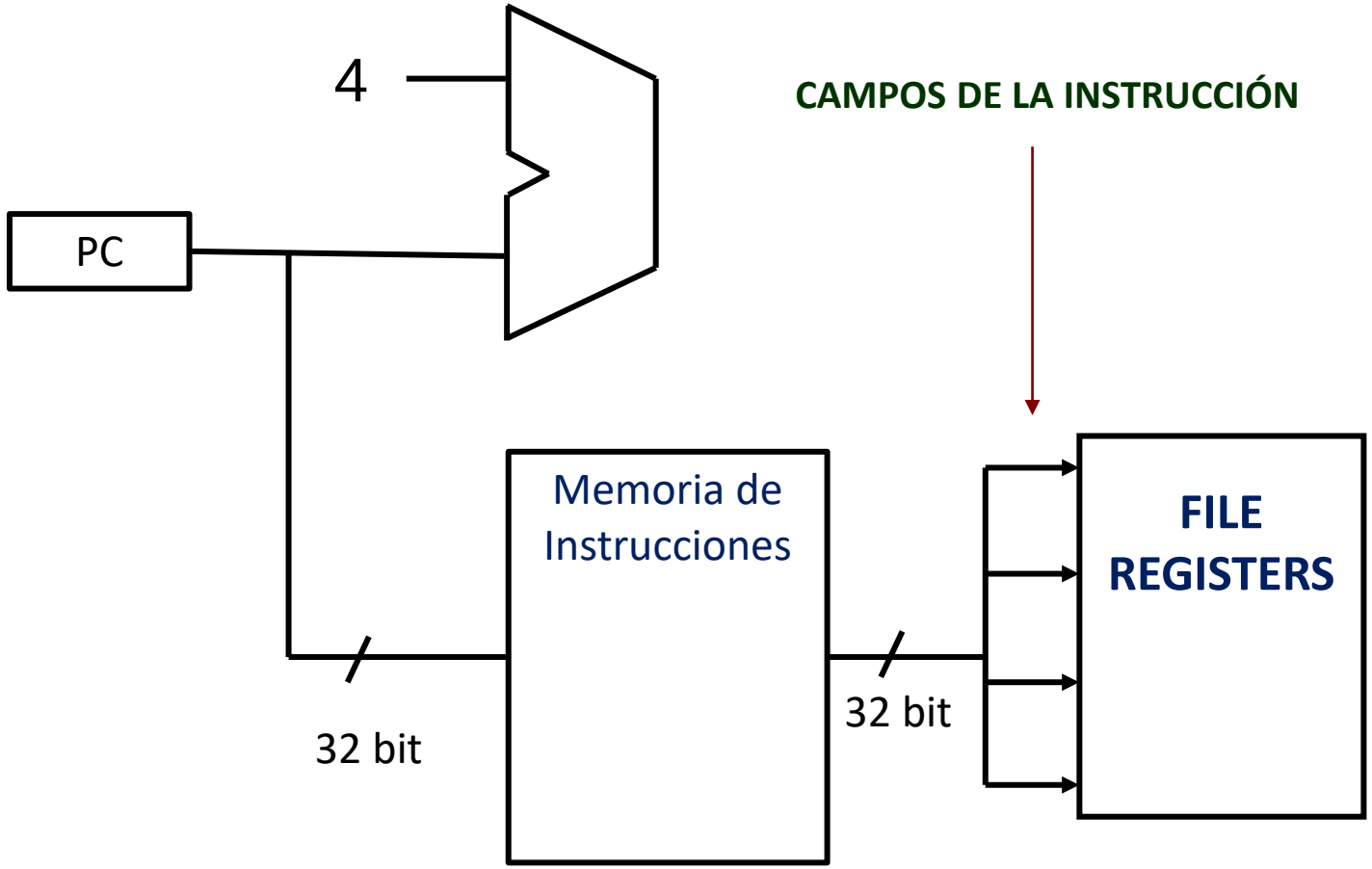
MEMORIA DE INSTRUCCIONES DE 8 BYTE



DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE



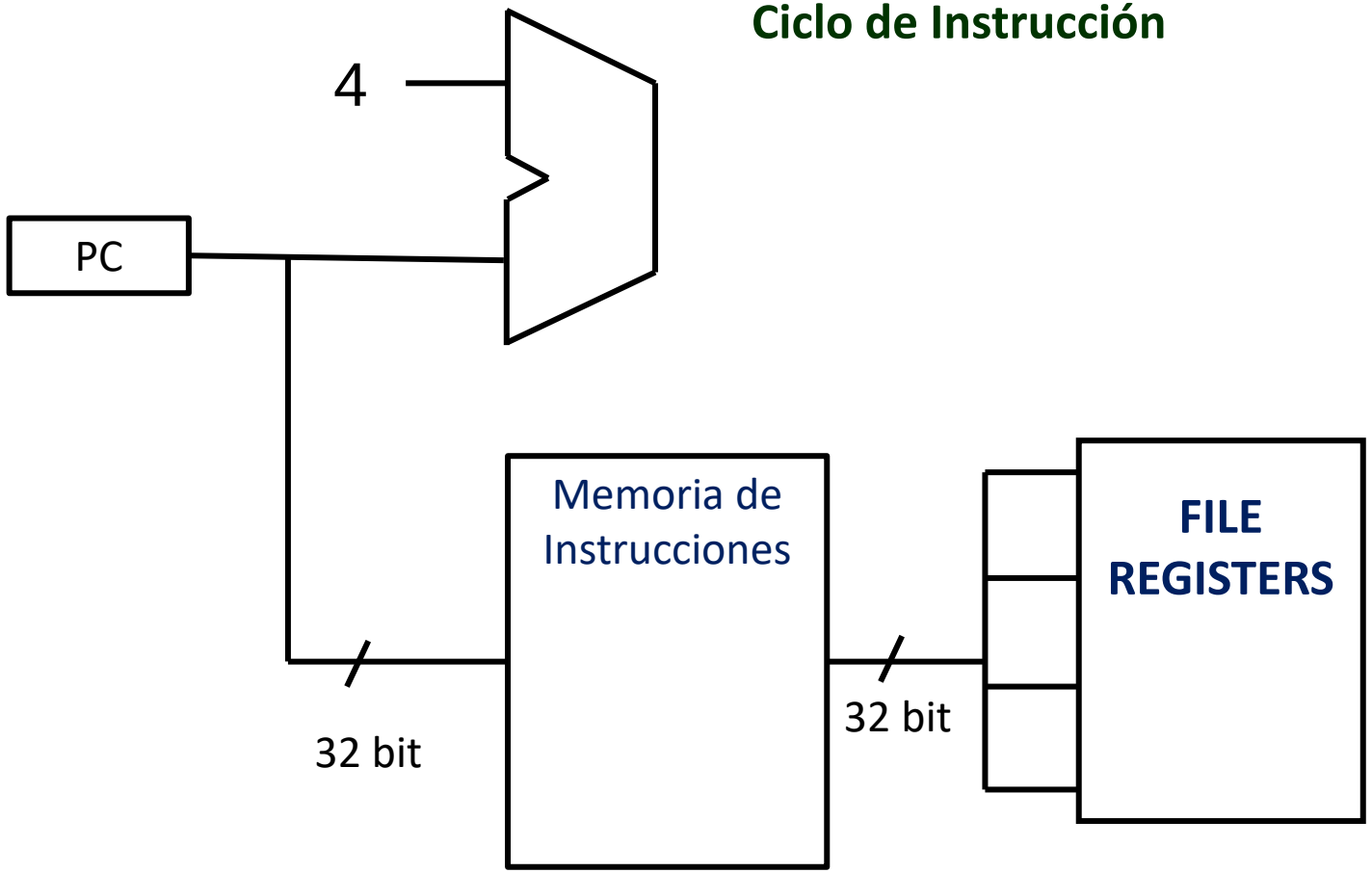
Fetch Instruction

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

Ciclo de Instrucción



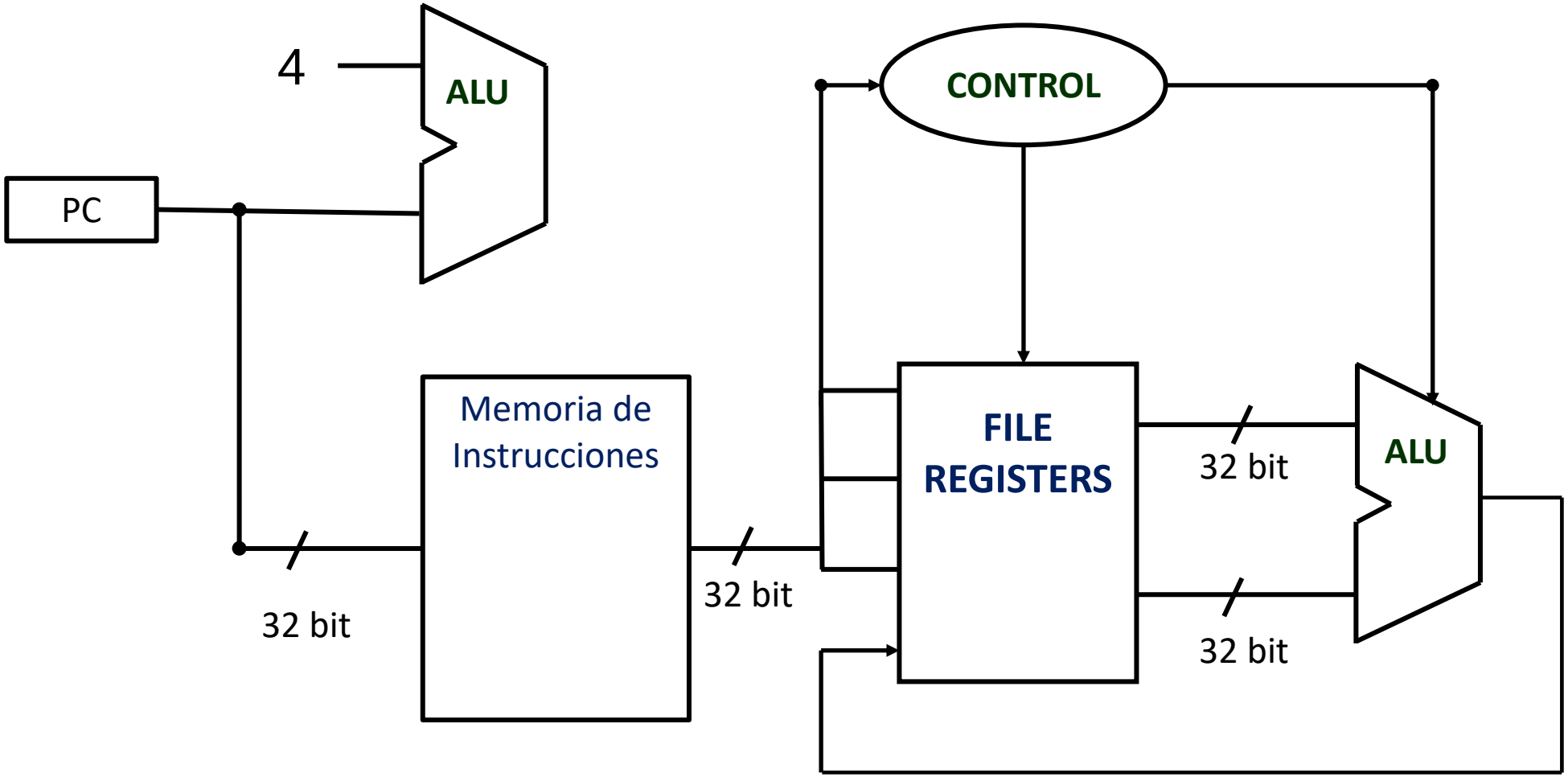
Fetch Instruction

Un archivo de registro es una colección de registros en los que se puede leer o escribir cualquier registro especificando el número del registro en el archivo. Además, obsérvese que todo se considera formando parte del procesador

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE



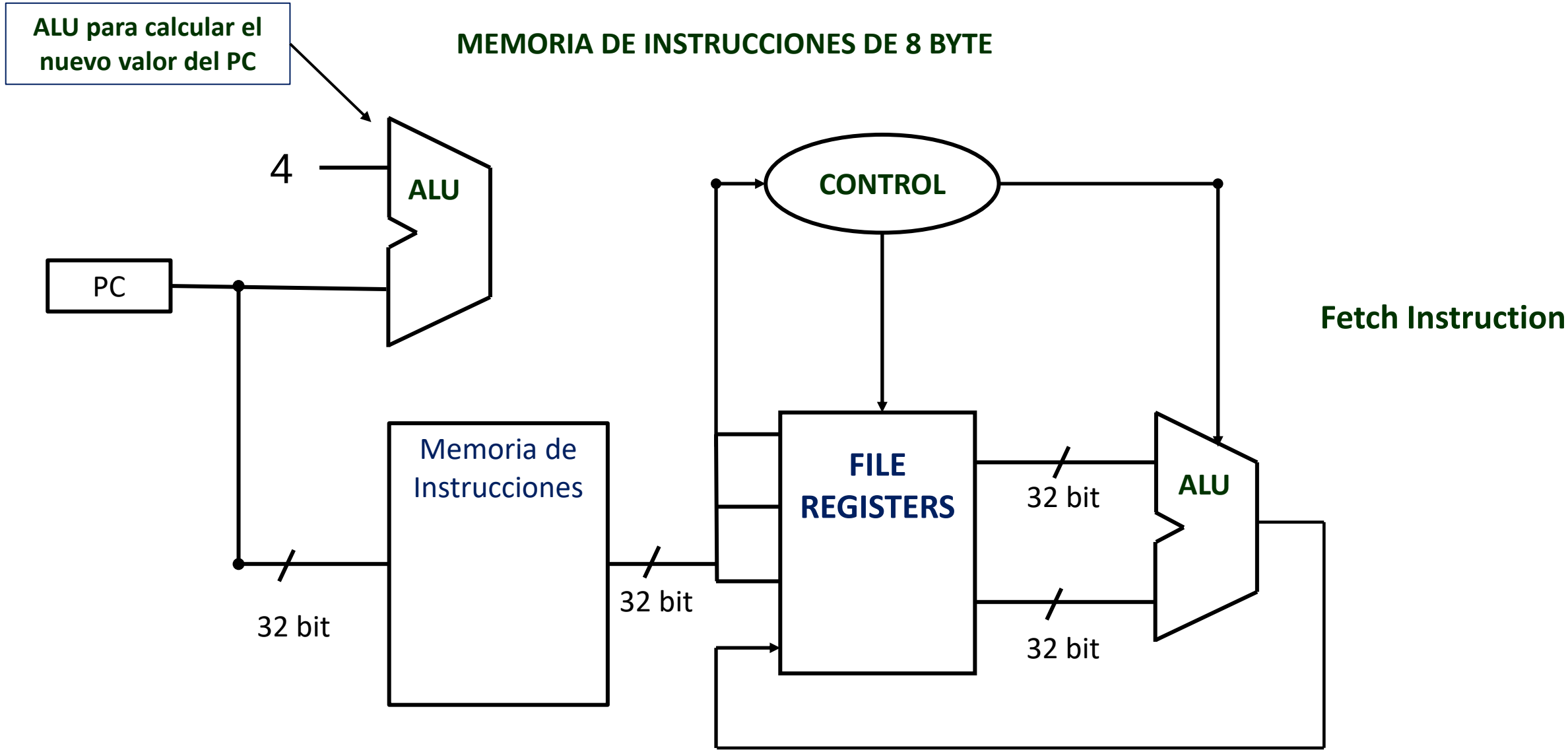
Instrucción "add"

Fetch Instruction

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

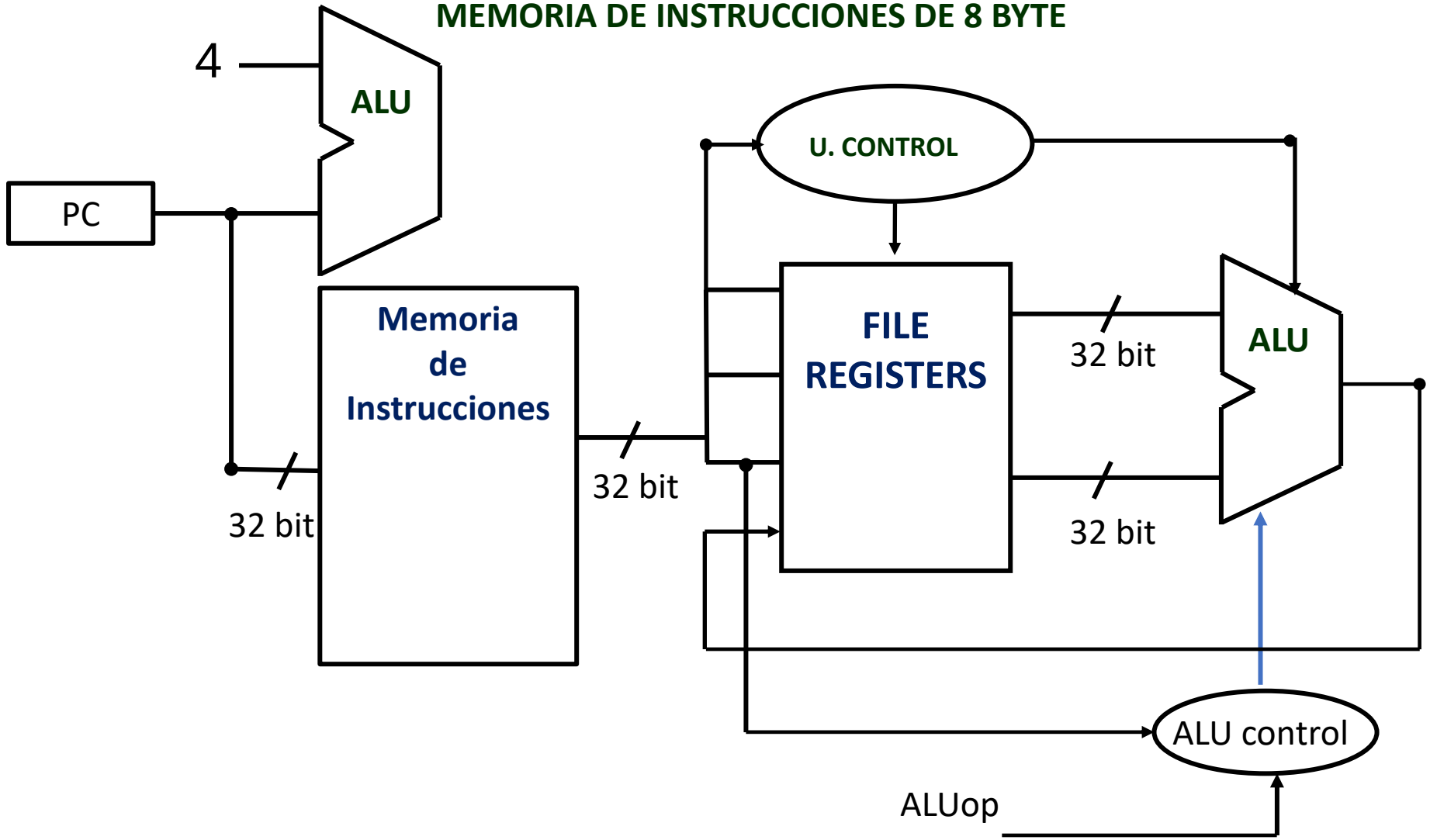


Fetch Instruction

DATAPATH

MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

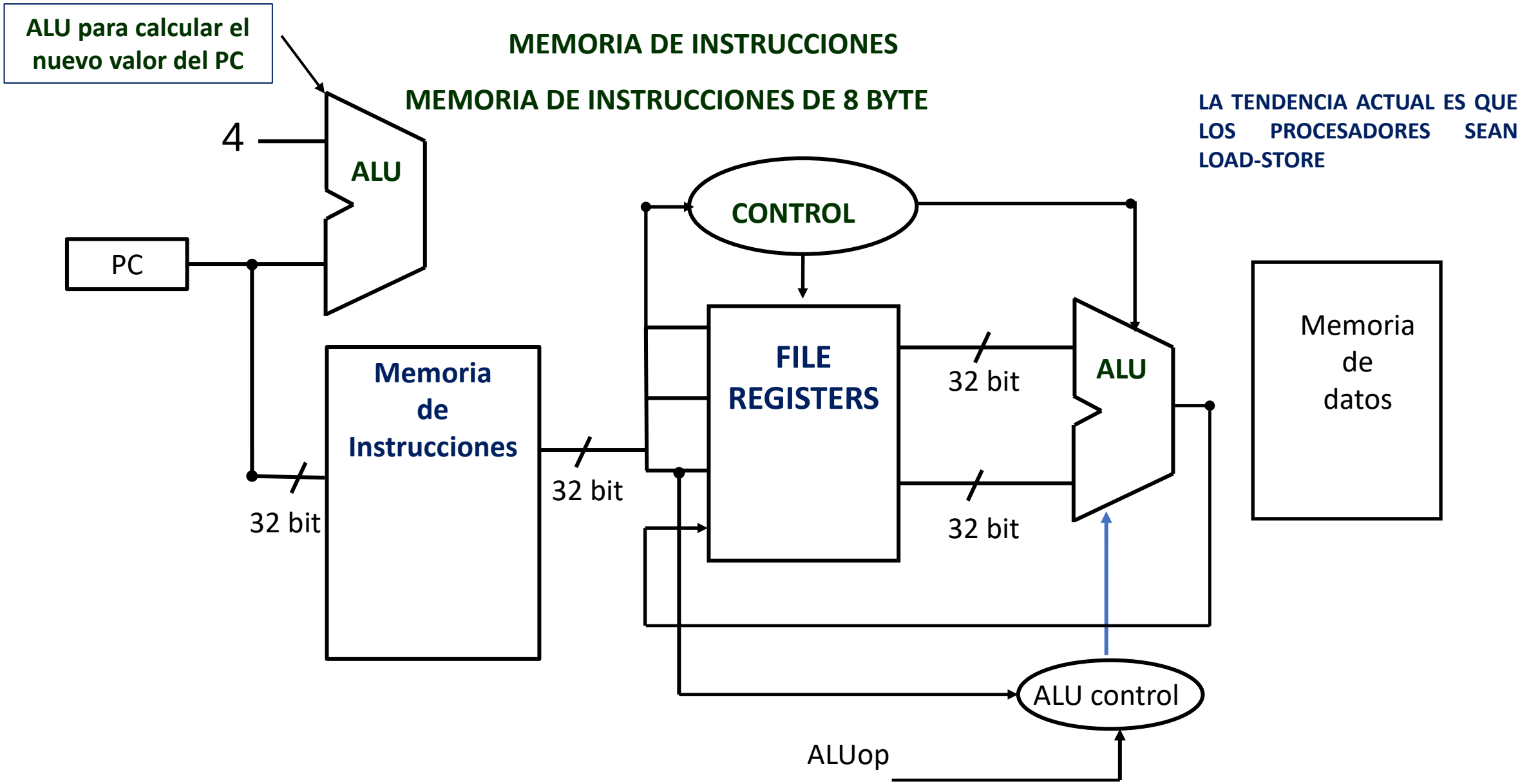


DATAPATH

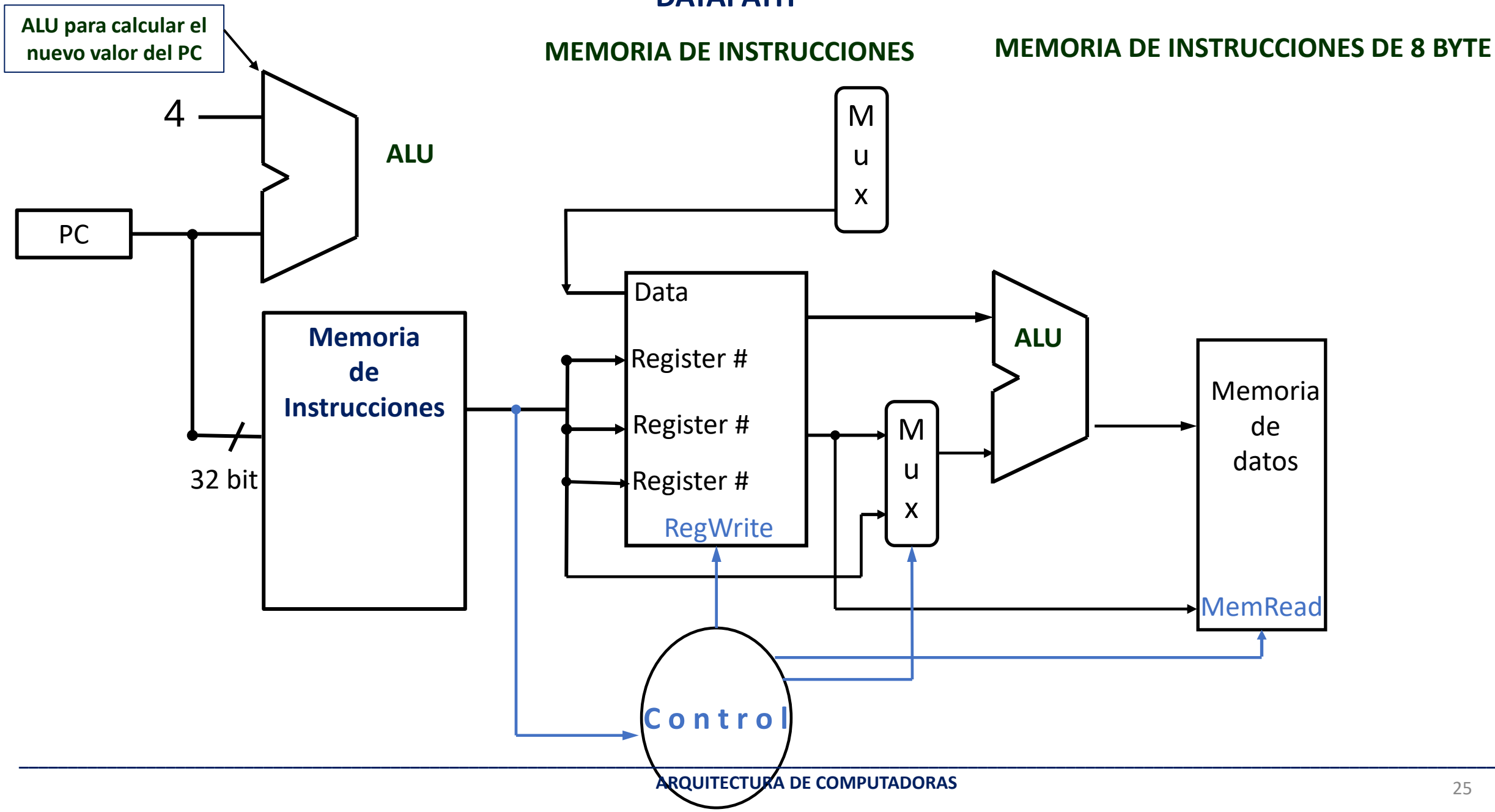
MEMORIA DE INSTRUCCIONES

MEMORIA DE INSTRUCCIONES DE 8 BYTE

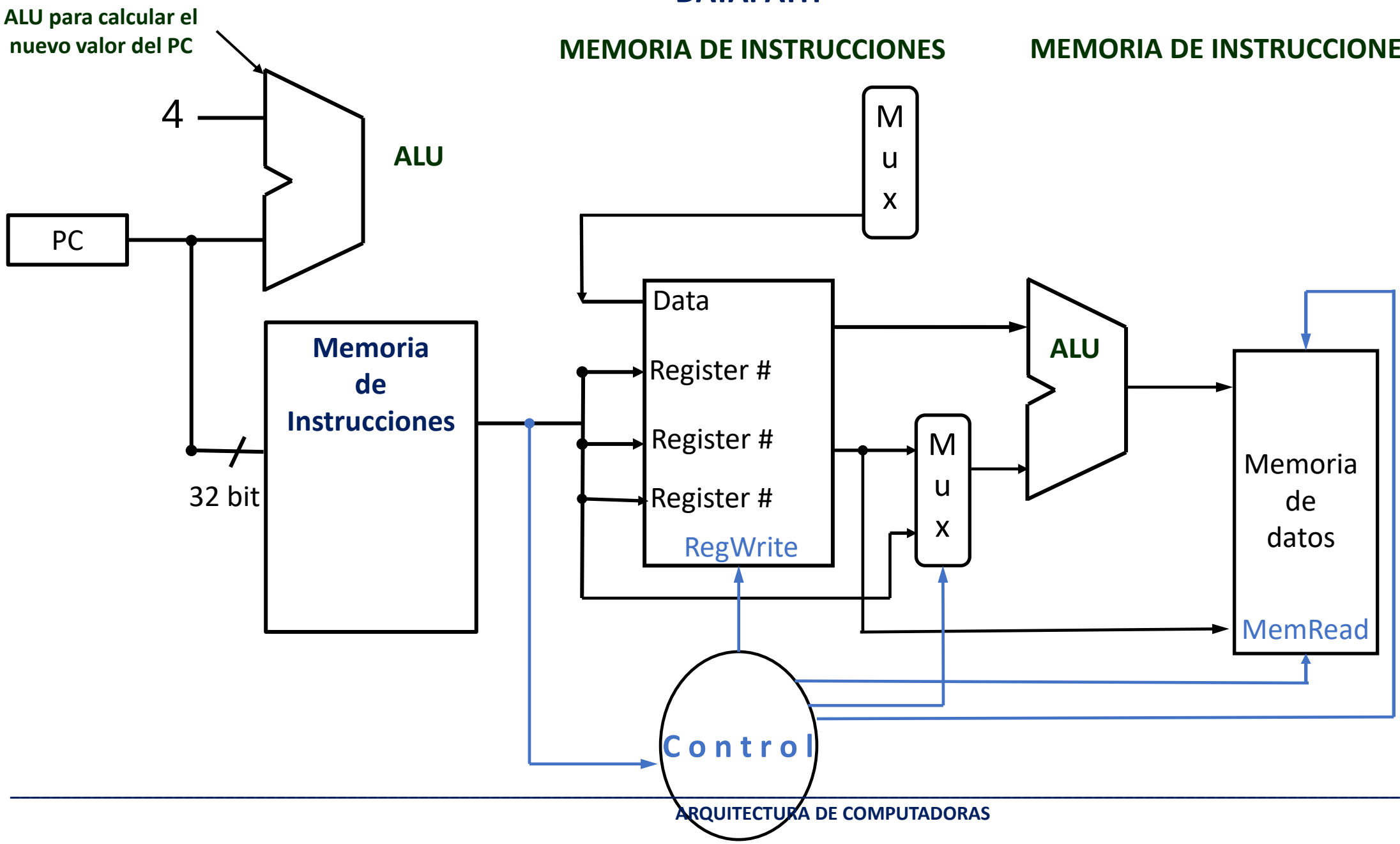
LA TENDENCIA ACTUAL ES QUE LOS PROCESADORES SEAN LOAD-STORE



DATAPATH


ALU para calcular el nuevo valor del PC
MEMORIA DE INSTRUCCIONES
MEMORIA DE INSTRUCCIONES DE 8 BYTE
4
ALU
M
u
x
PC
Memoria
de
Instrucciones
32 bit
Data
Register #
Register #
Register #
RegWrite
ALU
M
u
x
Memoria
de
datos
MemRead
Control

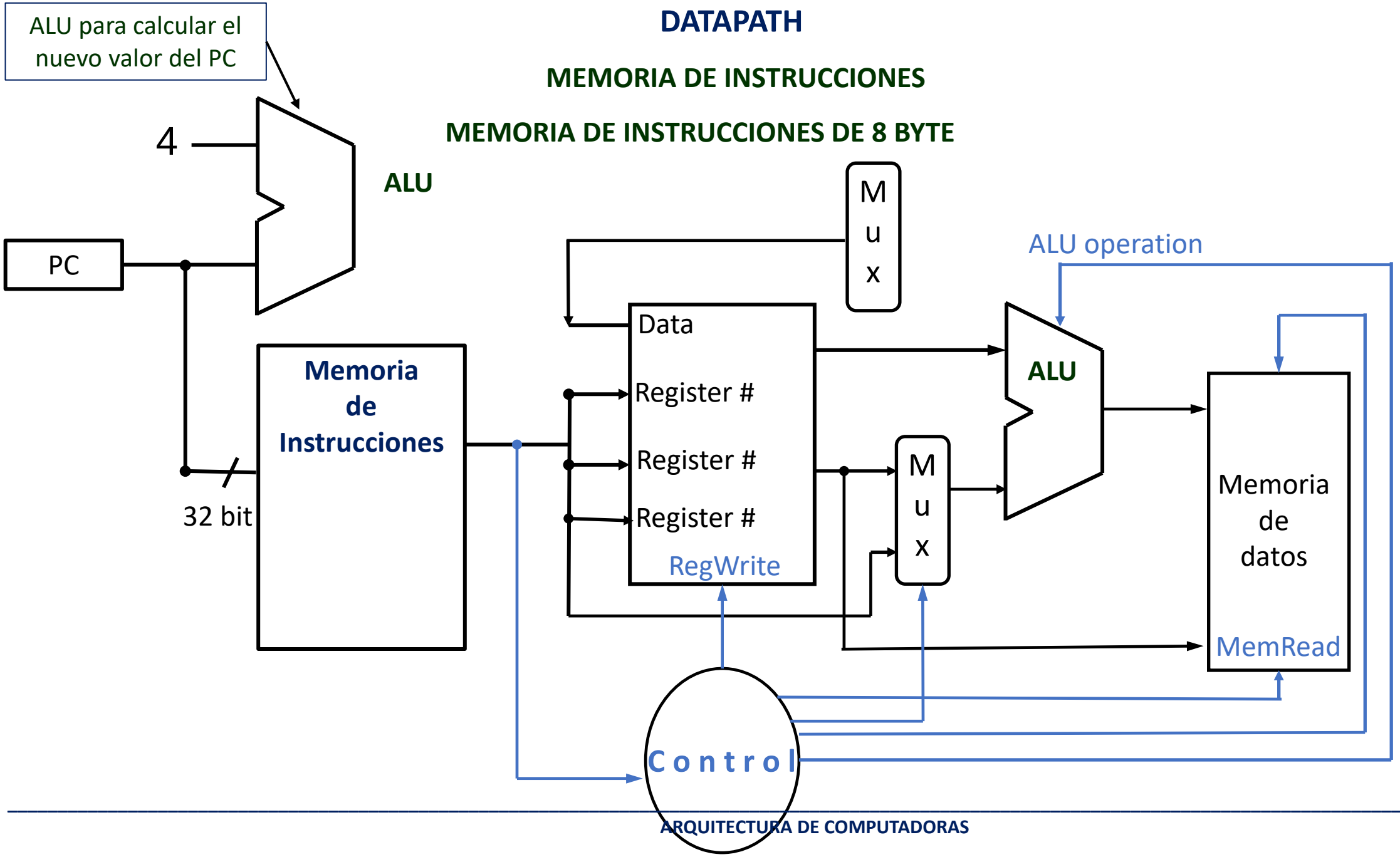
DATAPATH


ALU para calcular el nuevo valor del PC
MEMORIA DE INSTRUCCIONES
MEMORIA DE INSTRUCCIONES DE 8 BYTE
4
ALU
PC
M u x
Data
Register #
Register #
Register #
RegWrite
Memoria de Instrucciones
32 bit
ALU
M u x
Memoria de datos
MemRead
Control

DATAPATH

MEMORIA DE INSTRUCCIONES

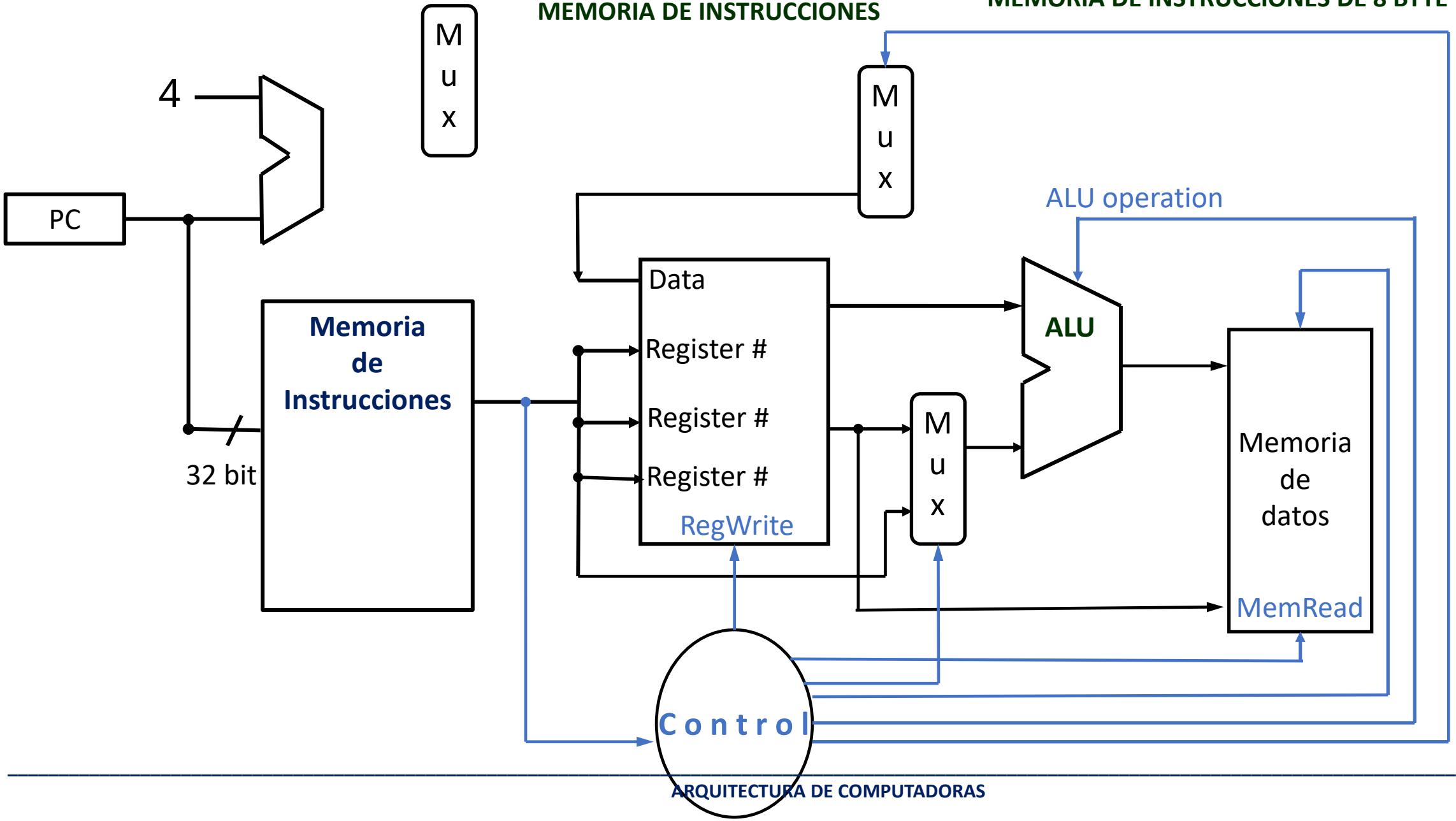
MEMORIA DE INSTRUCCIONES DE 8 BYTE

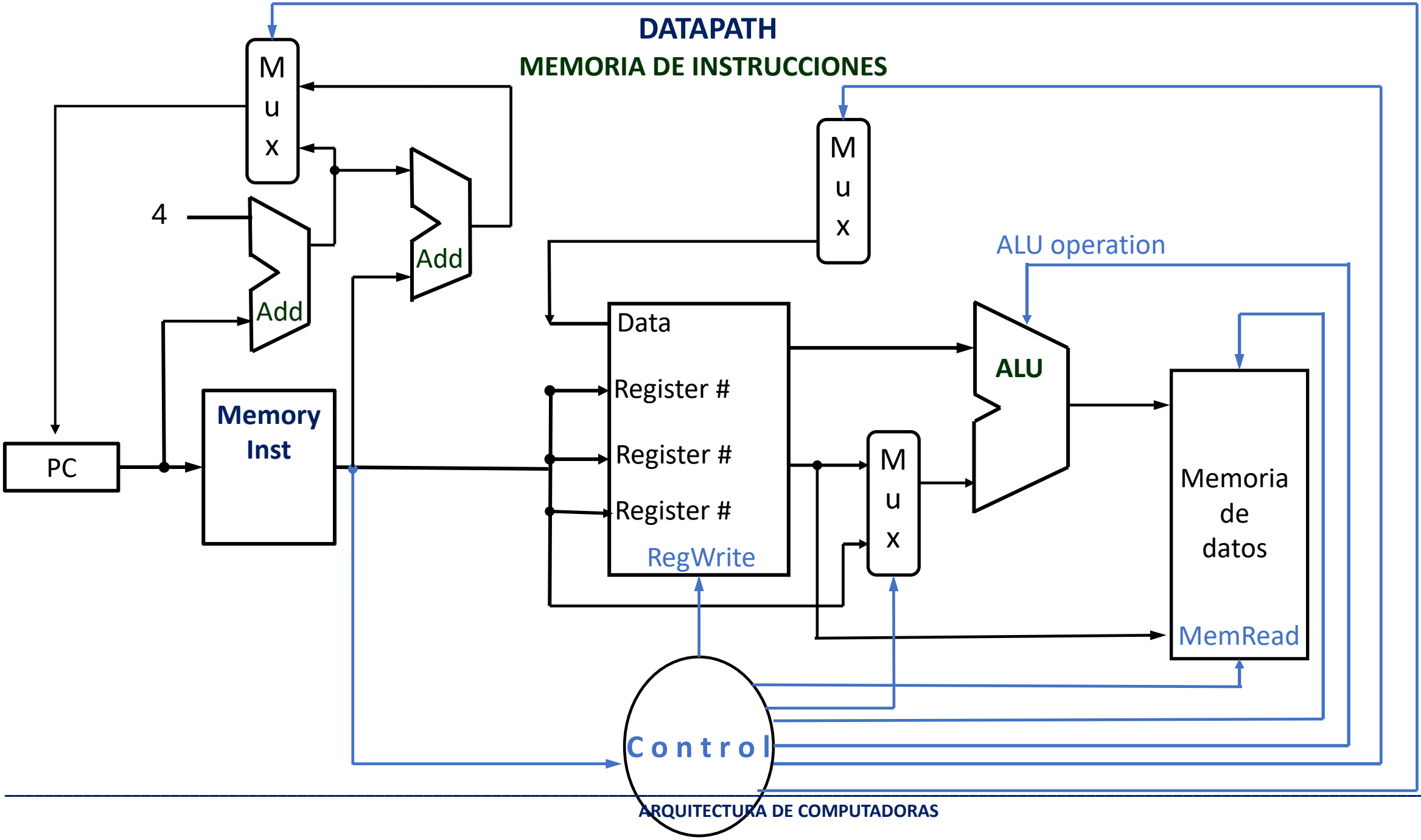


DATAPATH

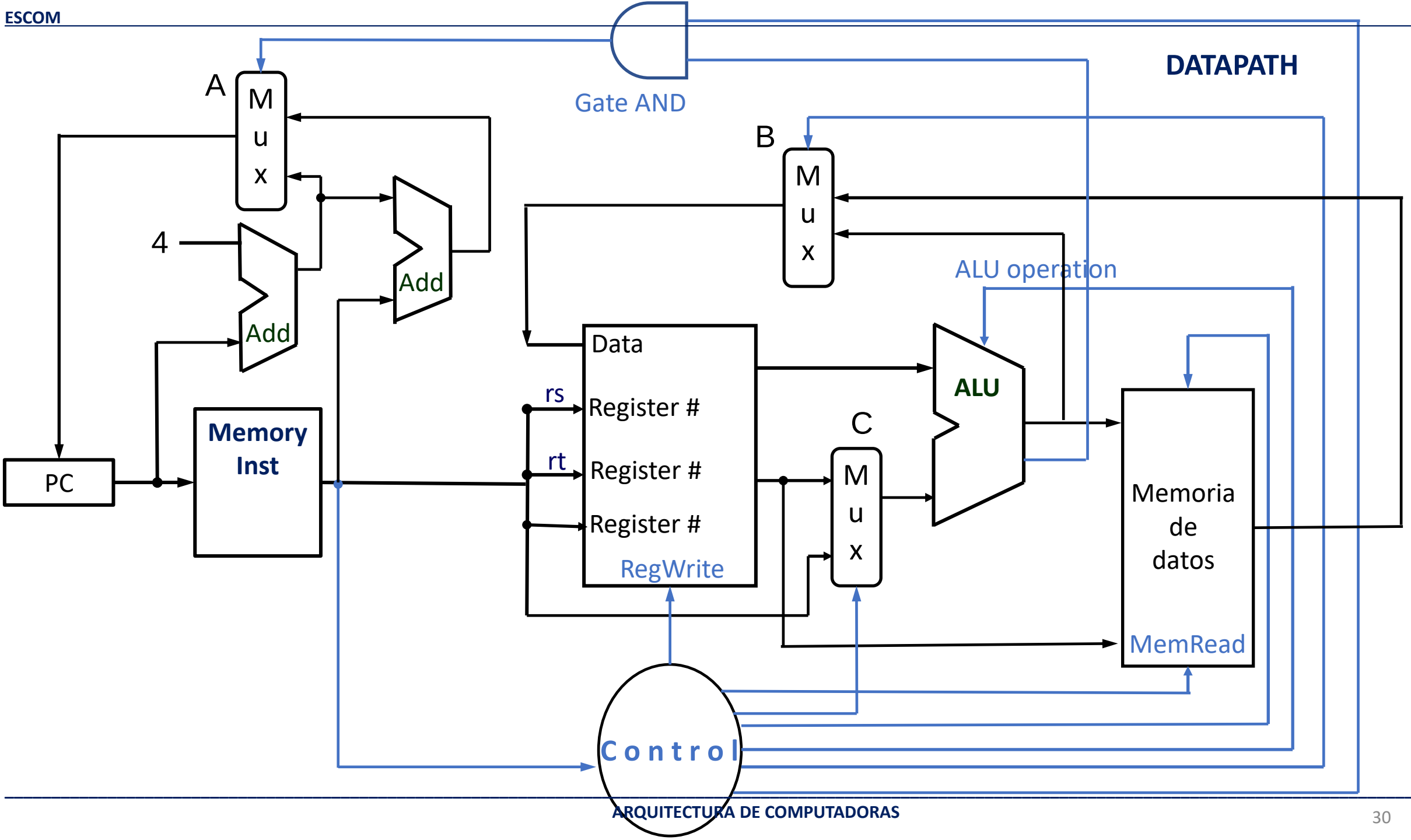
MEMORIA DE INSTRUCCIONES

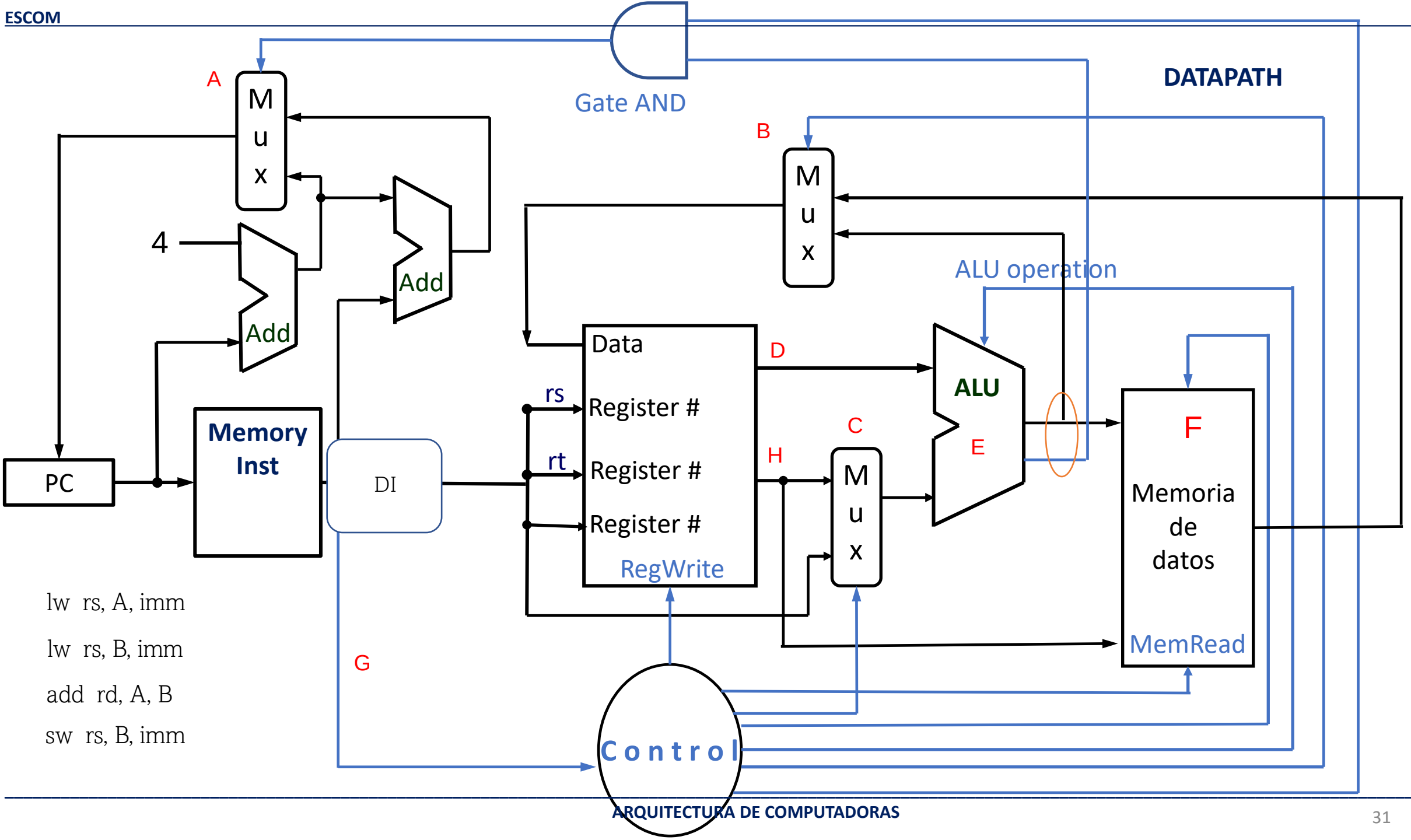
MEMORIA DE INSTRUCCIONES DE 8 BYTE




DATAPATH
MEMORIA DE INSTRUCCIONES
M
u
x
4
Add
Add
M
u
x
ALU operation
Data
Register #
Register #
Register #
RegWrite
ALU
Memory
Inst
PC
M
u
x
Memoria
de
datos
MemRead
Control

DATAPATH


Gate AND
A
M
u
x
B
M
u
x
4
Add
Add
Add
ALU operation
Data
rs
Register #
rt
Register #
Register #
RegWrite
ALU
C
M
u
x
Memory
Inst
PC
Memoria
de
datos
MemRead
Control


DATAPATH
Gate AND
A
M
u
x
B
M
u
x
4
Add
Add
Add
ALU operation
Data
Register #
Register #
Register #
RegWrite
rs
rt
D
H
C
M
u
x
ALU
E
F
Memoria
de
datos
MemRead
Memory
Inst
PC
DI
G
Control
lw rs, A, imm
lw rs, B, imm
add rd, A, B
sw rs, B, imm

DATAPATH

MEMORIA DE INSTRUCCIONES

TIPO R

OPCODE	rs	rt	rd	shamt	funct
31 downto 26	25 downto 21	20 downto 16	15 downto 11	10 downto 6	5 downto 0
000000 **Para el caso de las instrucciones aritméticas**	R-Fuente	R-Fuente	R-Destino	Solo para desplazamientos	Código de operación específica

Formato de Instrucciones Aritmético-Lógicas

El campo “shamt” es usado solo en operaciones de desplazamiento

DATAPATH

MEMORIA DE INSTRUCCIONES

	OPCODE	rs	rt	offset (address)
TIPO I	31 downto 26	25 downto 21	20 downto 16	15 downto 0
(Inmediato)	100011 ó 101011	00000	00001 Pagina 83	offset

Formato de Instrucciones Load o Store

Load: opcode = (35 base 10), (100011 base 2)

Store: opcode = (43 base 10), (101011 base 2)

"rs" es el registro base que es sumado a "offset" para formar la dirección de memoria

DATAPATH

MEMORIA DE INSTRUCCIONES

TIPO I

(Inmediato)

OPCODE	rs	rt	offset (address)
31 downto 26	25 downto 21	20 downto 16	15 downto 0
001000			constant

Formato de Instrucciones aritméticas: constante (Immediate)

addi: opcode = (8 base 10), (001000 base 2)

addi rt, rs, imm

“rs” operando fuente que sumado a “constant” entrega un resultado en el registro “rt”

DATAPATH

MEMORIA DE INSTRUCCIONES

TIPO I

(Inmediato)

OPCODE	rs	rt	offset (address)
31 downto 26	25 downto 21	20 downto 16	15 downto 0
100011 ó 101011			address

Formato de Instrucciones Load o Store

"rs" es el registro base que es sumado a "offset" para formar la dirección de memoria

Para una operación "load" "rt" es el registro destino (nombre del registro de prop. Gen.)

Lo anterior es para el caso de una operación de load

DATAPATH

MEMORIA DE INSTRUCCIONES

TIPO I

(Inmediato)

OPCODE	rs	rt	offset (address)
31 downto 26	25 downto 21	20 downto 16	15 downto 0
100011 ó 101011			address

Formato de Instrucciones Load o Store

“rs” es el registro base que es sumado a “offset” para formar la dirección de memoria

Para una operación “store” “rt” es el registro fuente cuyo valor será almacenado en memoria

DATAPATH

MEMORIA DE INSTRUCCIONES

TIPO I

(Inmediato)

OPCODE	rs	rt	address
31 downto 26	25 downto 21	20 downto 16	15 downto 0
000100			address

Formato de Instrucciones Branch

“rs” y “rt” son registros fuente que serán comparados para hacer una obtener un true o false

El campo de 16 bit (del bit 15 al bit 0) es el campo “address”. Primero se extiende de 16 a 32 bit para obtener el próximo valor de PC