



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2343 - Arquitectura de computadores  
2do semestre del 2015

# Informe Entrega 3

## Grupo 1

### Felipe Aguayo

Implementó Control Unit, colaboró en el debug del Assembler y probó ejemplos.

### Carlos García

Colaboró en la implementación de MuxIN, MuxDataIN, PCAdder, SP, MuxS y MuxPC.

### Nicolás Gebauer

Hizo conexiones en Basys 3; implementó Control Unit, MuxIN, MuxDataIN, PCAdder, SP, MuxS y MuxPC; trabajó en el debug del Assembler; y probó ejemplos.

### Andrés Matte

Programó el Assembler y trabajó en el correspondiente debug.

### Dasen Razmilic

Programó las funciones pedidas para los arreglos de enteros.

# Estructura de Instrucciones de la CPU

A continuación se presenta la especificación de la estructura de las instrucciones de la CPU, es decir, la función de cada uno de los 33 bits de una instrucción:

## Salida Instruction Memory:

- **Literal - Opcode**  
00000000000000000000000000000000
- **Dirección de instrucción**  
00000000000000000000000000000000

## Salida Control Unit:

El opcode es analizado por el Control Unit, el cual genera las señales de control en un bus de la siguiente manera:

- Lpc: **Load PC**  
000000000000000000000000
- La: **Load A**  
000000000000000000000000
- Lb: **Load B**  
000000000000000000000000
- Sa0 y Sa1: **Selector Mux A**  
000000000000000000000000
- Sb0 y Sb1: **Selector Mux B**  
000000000000000000000000
- Sop0, Sop1, Sop2: **Selector ALU**  
000000000000000000000000
- Sadd0 y Sadd1: **Selector Mux Address**  
000000000000000000000000
- Sdin0: **Selector Mux DataIn**  
000000000000000000000000
- Spc0: **Selector Mux PC**  
000000000000000000000000
- W: **Write RAM**  
000000000000000000000000
- IncSp: **Incremento SP**  
000000000000000000000000
- DecSp: **Decremento SP**  
000000000000000000000000

Luego simplemente conectamos cada parte del bus, es decir, cada señal de control, con el sistema que controla.

## **Tabla de Instrucciones**

La Tabla de instrucciones es la que se puede apreciar en el Cuadro 1: Instrucciones soportadas por la CPU-Parte 1 y en el Cuadro 2: Instrucciones soportadas por la CPU-Parte 2 .

## **Assembler**

Para utilizar el assembler se debe arrastrar el archivo de prueba que contiene el código assembly al archivo Assembler.exe . Este último generará un archivo output.txt que tendrá el código en lenguaje vhdl que se debe copiar y pegar en el proyecto.

## **Funciones**

Los algoritmos solicitados para solucionar los arreglos de enteros se podrán encontrar en la carpeta Algoritmos Entrega/AlgoritmosE3

Cuadro 1: Instrucciones soportadas por la CPU - Parte 1

Instrucción	Operadores	Opcode	Instrucción	Operadores	
MOV	A, B	0000000	OR	A, B	0100111
	B, A	0000001		B, A	0101000
	A, Lit	0000010		A, Lit	0101001
	B, Lit	0000011		B, Lit	0101010
	A, Dir	0000100		A, Dir	0101011
	B, Dir	0000101		B, Dir	0101100
	Dir, A	0000110		Dir	0101101
	Dir, B	0000111		A, DirB	0101110
	A, DirB	0001000		B, DirB	0101111
	B, DirB	0001001	XOR	A, B	0110000
ADD	DirB, A	0001010		B, A	0110001
	DirB, Lit	0001011		A, Lit	0110010
	A, B	0001100		B, Lit	0110011
	B, A	0001101		A, Dir	0110100
	A, Lit	0001110		B, Dir	0110101
	B, Lit	0001111		Dir	0110110
	A, Dir	0010000		A, DirB	0110111
	B, Dir	0010001		B, DirB	0111000
	Dir	0010010	NOT	A	0111001
	A, DirB	0010011		B, A	0111010
SUB	B, DirB	0010100		Dir, A	0111011
	A, B	0010101		DirB, A	0111100
	B, A	0010110	SHL	A	0111101
	A, Lit	0010111		B, A	0111110
	B, Lit	0011000		Dir, A	0111111
	A, Dir	0011001		DirB, A	1000000
	B, Dir	0011010	SHR	A	1000001
	Dir	0011011		B, A	1000010
	A, DirB	0011100		Dir, A	1000011
	B, DirB	0011101		DirB, A	1000100
AND	A, B	0011110	INC	A	1000101
	B, A	0011111		B	1000110
	A, Lit	0100000		Dir	1000111
	B, Lit	0100001		DirB	1001000
	A, Dir	0100010	DEC	A	1001001
	B, Dir	0100011			
	Dir	0100100	CMP	A, B	1001010
	A, DirB	0100101		A, Lit	1001011
	B, DirB	0100110		A, Dir	1001100
				A, DirB	1001101

Cuadro 2: Instrucciones soportadas por la CPU - Parte 2

Instrucción	Operadores	Opcode
PUSH	A	1001110
	B	1001111
POP (1)		1100000
POP (2)	A	1010000
	B	1010001
CALL	Dir	1010010
RET (1)		1100001
RET (2)		1010011