

Problema 1:

- Sea un procesador con el siguiente conjunto de instrucciones:

Instrucción		Formato de la instrucción (XXXX-No usado)
Mnemónico	Significado	
ADD <i>rs, rf</i>	$rs \leftarrow ADD(rs, rf)$	00 <i>rs</i> <i>rf</i> XXXX 00
ADD <i>rs, #cte</i>	$rs \leftarrow ADD(rs, cte)$	00 <i>rs</i> <i>cte</i> 01
ADD <i>rs, (rf)</i>	$rs \leftarrow ADD(rs, M[rf])$	00 <i>rs</i> <i>rf</i> XXXX 10
ADD <i>rs, dir</i>	$rs \leftarrow ADD(rs, M[PC+1+dir])$	00 <i>rs</i> <i>dir</i> 11
JZ <i>desp</i>	Si $Z=1$ $PC \leftarrow ADD(PC+1, desp)$	10 <i>desp</i>

- Las características del procesador son las siguientes:
 - Tamaño de palabra de 16 bits; Banco de registros de 16 registros; *cte*, *desp* y *dir* están representados en C2; los 2 últimos bits de las instrucciones ADD indican el modo de direccionamiento del segundo operando; PC se incrementa en 1
- Dibuja el datapath del procesador.
- Diseña la unidad de control suponiendo un procesador monociclo
- Añade al diseño la siguiente instrucción:

$LAC\ rd \quad rd \leftarrow M[AC] \quad 11||xxxxxxx||rd$
 (AC es un registro de propósito específico)

Problema 2:

- Diseñar la unidad de datos y de control (versión monociclo) de una máquina que ejecuta las siguientes instrucciones:

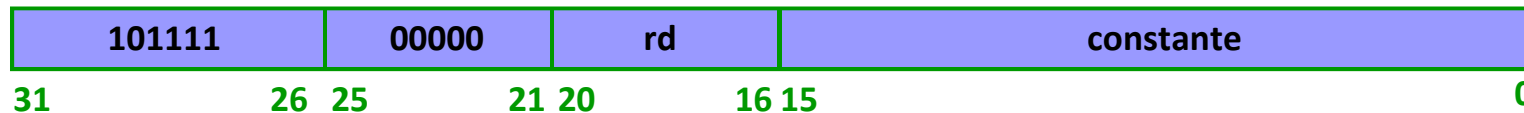
Mnemónico	Descripción
LOAD X	Transfiere el contenido de la posición X al acumulador: $AC \leftarrow MEM(X)$
STORE X	Transfiere el contenido del acumulador a la posición X: $MEM(X) \leftarrow AC$
ADD X	Suma al acumulador el contenido de la posición X: $AC \leftarrow AC + MEM(X)$
AND X	AND lógica del acumulador con el contenido de la posición X: $AC \leftarrow AC \wedge MEM(X)$
JUMP X	Salto incondicional a la dirección X: $PC \leftarrow X$
JUMPZ X	Salta a la dirección X si el resultado fue cero: $si Z \Rightarrow PC \leftarrow X$
COMP	Complemento lógico del acumulador: $AC \leftarrow \overline{AC}$
RSHIFT	Desplazamiento a la derecha del acumulador: $AC \leftarrow 0, AC[15 : 1]$

- El tamaño de palabra del procesador es de 16 bits y las memorias son de tamaño 8K x16
- La ALU realiza las siguientes operaciones: suma y AND lógica de dos operandos, desplazamiento a la derecha y complemento lógico del operando que entra por el puerto B.

Problema 3:

- Modifica la unidad de datos y de control del procesador MIPS para añadir la siguiente instrucción:

Instrucción LUI: LUI rd, constante : $rd \leftarrow \text{constante}, 0..0$



Problema 4:

Supongamos que tenemos el procesador del problema 1. Dados los contenidos de memoria principal que se muestran en la tabla (en hexadecimal) y suponiendo que PC contiene 0045_{16} y que el contenido de todos los registros es 0, mostrar como evolucionan los contenidos de los elementos de memoria (traza) en la tabla vacía

dir.	contenido
...	...
29_{16}	0033_{16}
$2A_{16}$	0042_{16}
$2B_{16}$	0008_{16}
...	...
41_{16}	0023_{16}
42_{16}	$000A_{16}$
43_{16}	$0C01_{16}$
44_{16}	0001_{16}
45_{16}	1793_{16}

dir.	contenido
46_{16}	0542_{16}
47_{16}	$0BD5_{16}$
48_{16}	0480_{16}
49_{16}	0405_{16}
$4A_{16}$	8002_{16}
$4B_{16}$	0801_{16}
$4C_{16}$	$0C01_{16}$
$4D_{16}$	1001_{16}
$4E_{16}$	1401_{16}
...	...

Dirección comienzo	mnemónico ensamblador	Operando/s fuente	operando destino	valor a escribir