

ESTRUCTURA DE COMPUTADORES

Ejercicios Tema 1: El procesador

1. Modifica la unidad de control vista en clase para dar soporte a la siguiente instrucción:

```
addi rt, rs, inm    # rt = rs + inm
```

Para realizar el ejercicio, completa la tabla de verdad de la unidad de control:

				Banco Registros	ALU	Mem. Datos		Multiplexores Configuración Ruta de Datos			
Instrucción	Form	Código Op.	Función	EReg	OpALU	LMem	EMem	MxPC	MxALU	MxDst	MxER
add rd, rs, rt	R	000000	100000	1	010	0	0	0	0	1	0
sub rd, rs, rt	R	000000	100010	1	110	0	0	0	0	1	0
and rd, rs, rt	R	000000	100100	1	000	0	0	0	0	1	0
or rd, rs, rt	R	000000	100101	1	001	0	0	0	0	1	0
lw rt, desp(rs)	I	100011		1	010	1	0	0	1	0	1
sw rt, desp(rs)	I	101011		0	010	0	1	0	1	X	X
beq rs, rs, etiq	I	000100		0	110	0	0	Z	0	X	X
addi rt, rs, inm											

2. Modifica la ruta de datos y unidad de control vista en clase para dar soporte a la siguiente instrucción:

```
ori rt, rs, inm    # rt = rs v inm (extendiendo a 0s)
```

3. Codifica las siguientes instrucciones:

```
add $t2, $t1, $t0  
li $t0, 0x10002000
```

Puedes hacer uso del simulador PCSpim (o del QtSpim) para realizar el ejercicio. Ten en cuenta que las pseudo-instrucciones se desdobl原因 en un conjunto de instrucciones antes de su codificación. Justifica la respuesta (indicando los diferentes campos de cada instrucción: COp, Rs, Rt, ...)

4. Analiza la operación que realizan las instrucciones codificadas de la siguiente forma a partir de la dirección 0x00400000.

Dirección	Contenido
0x00400000	0x3c081000
0x00400004	0x8d090000
0x00400008	0x8d0a0004
0x0040000C	0x012a5820
0x00400010	0xad0b0008

Para realizar este ejercicio debemos tener presente la codificación de las instrucciones MIPS.

- ## 5. Modifica la ruta de datos para dar soporte a la instrucción

```
lui rt, inm      # rt31..16 = inm; rt15..0 = 0
```

Para ello se decide modificar el recurso “Extensión de signo” con el fin de que tenga una funcionalidad adicional. El nuevo recurso se denomina ahora `Ext Signo/Shifter` y se configura con una nueva señal de control denominada `shift`. Cuando la señal `shift` vale 0 el recurso realiza la extensión de signo de la entrada. Cuando la señal `shift` vale 1, el recurso genera una salida de 32 bits a partir de la entrada de 16 bits, donde los 16 bits de la entrada aparecen en los 16 bits de mayor peso en la salida y los 16 bits de menor peso de la salida tienen el valor 0.

Indica los cambios a realizar tanto en la ruta de datos que se muestra a continuación, así como los cambios a realizar en la unidad de control.

[illegible]

