



## Arquitectura de computadores

### Pràctica 2

6 de març de 2019

Disseny de la unitat Aritmètica i Lògica (ALU)

i

Unitat de control de l'ALU

Antoni Escobet

# PRÀCTICA 2 – Disseny de la Unitat Aritmètica i Lògica (ALU) i la unitat de control

## 1. Objectius

En aquesta practica es pretén:

- Repassar els conceptes relacionats amb la unitat aritmètica i lògica del processador MIPS vist a la classe de teoria.
- Dissenyar una ALU de 32 bits per nombres sencers.
- Dissenyar la unitat de control per les instruccions bàsiques.
- Ampliar els coneixements sobre el llenguatge VHDL i l'edició esquemàtica de circuits amb QUARTUS.

## 2. Material

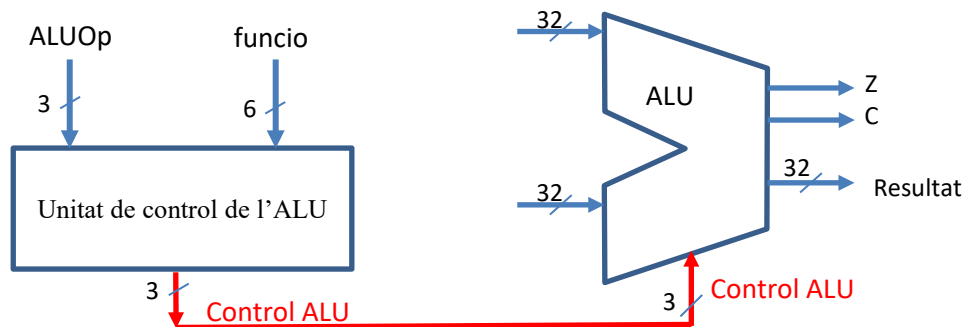
L'únic material necessari per la realització d'aquesta pràctica és el paquet de programari d'ALTERA, el Quartus instal·lat als ordinadors del laboratori, i que us podeu descarregar gratuïtament de la pàgina web d'Altera (<http://www.altera.com>).

La simulació del vostre disseny s'ha de fer amb el simulador "ModelSim-Altera" que proporciona el mateix paquet de programari d'Altera.

Recordeu que la placa sobre la que és pot implementar els vostres dissenys, és la CYCLONE IV E EP4CE22F17C6.

## 3. Problema proposat

En aquesta primera pràctica s'ha de realitzar el disseny de la unitat aritmètica i lògica pel processador que s'ha vist a teoria i el seu bloc de control.



El disseny del processador vist a classe, amb el seu camí de dades i la seva unitat de control permet executar un conjunt reduït d'instruccions relacionades amb la unitat aritmètica i lògica:

add, sub, and, or, slt, beq, bne, ..

És a dir, que pot realitzar sumes i restes de nombres sencers de 32 bits, i les operacions lògiques AND i OR sobre 32 bits. També pot comparar dos operants per saber si un és més gran que l'altre.

Per tal de poder fer els salts condicionals, serà necessari que l'ALU proporcioni a la unitat de control, l'indicador de si el resultat de l'operació val zero (Z) i si ens en portem un en una operació aritmètica (C).

## 4. Descripció de la practica

### 4.1. Disseny de l'ALU

La unitat aritmètica lògica serà asíncrona i tindrà la següent definició d'entrades i sortides:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_signed.all;

entity ALU32 is
  Port (
    Op1, Op2 :      in STD_LOGIC_VECTOR (31 downto 0);
    Control_ALU :  in STD_LOGIC_VECTOR (2 downto 0);
    Resultat :     out STD_LOGIC_VECTOR (31 downto 0);
    Z :            out STD_LOGIC;
    C :            out STD_LOGIC)
end ALU32;
```

on **Op1** i **Op2** són els dos operands, **Resultat** és, com el seu nom indica, el resultat de l'operació a realitzar, **Z** és la sortida zero que valdrà 1 si el resultat és zero, **C** és el bit de "carry". I per últim **Control\_ALU** són les línies de control que li indicaran a la unitat aritmètica i lògica el tipus d'operació a realitzar. A la Taula es mostren els codis que indiquen el tipus d'operació a realitzar per l'ALU.

<b>Operació</b>	<b>Funció</b>
0 0 0	AND
0 0 1	OR
0 1 0	Suma
1 1 0	Resta
1 1 1	Activar si menor que

### 4.2. Operacions de l'ALU

#### 4.2.1. AND

Fa l'operació lògica *I* entre els dos operands d'entrada. Modifica la sortida Z depenen del resultat. La sortida C sempre val zero.

$$\text{Resultat} = \text{OP1} \text{ and } \text{OP2}$$

#### 4.2.2. OR

Fa l'operació lògica *O* entre els dos operands d'entrada. Modifica la sortida Z depenen del resultat. La sortida C val zero.

$$\text{Resultat} = \text{OP1} \text{ or } \text{OP2}$$

#### 4.2.3. Suma

Fa l'operació aritmètica *suma* entre els dos operands d'entrada. Modifica les sortides Z i C depenen del resultat.

$$\text{Resultat} = \text{OP1} + \text{OP2}$$

#### 4.2.4. Resta

Fa l'operació aritmètica *resta* entre els dos operands d'entrada. Modifica les sortides Z i C depenen del resultat.

$$\text{Resultat} = \text{OP1} - \text{OP2}$$

#### 4.2.5. Activar si menor que (*set-on-less-than*).

Si  $\text{OP1} < \text{OP2} \rightarrow \text{Resultat} = 1$ , en cas contrari  $\text{Resultat} = 0$ .

Un exemple pot ser: `slt $1,$2,$3` → si ( $\$2 < \$3$ ) llavors  $\$1 = 1$ ; en cas contrari  $\$1 = 0$ ;

Per tant, la instrucció *activar si menor que* (`slt`) posarà el bit menys significatiu del resultat a 0 o 1 depenent del resultat de la comparació (la resta de bits, de l'1 al 31, es deixaran a zero).

### 4.3. Unitat de control de l'ALU

Aquesta unitat de control de l'ALU, ha de generar els tres bits de comandament que té la nostre ALU. Les entrades d'aquesta unitat, seran dos bits generats per la UC que es dissenyarà en una altre pràctica, i els 6 bits del camp de "funció" de la instrucció. Aquest bloc també ha de ser asíncron

Els codis d'operació (CO) i funció (func) són els següents per a les diferents instruccions que executa el processador:

<i>Instrucció</i>	ALUOp	Funció						Operació ControlALU
		F5	F4	F3	F2	F1	F0	
add	10	1	0	0	0	0	0	010
sub	10	1	0	0	0	1	0	110
and	10	1	0	0	1	0	0	000
or	10	1	0	0	1	0	1	001
nop	10	0	0	0	0	0	0	xxx
addi	00	x	x	x	x	x	x	010
lw	00	x	x	x	x	x	x	010
sw	00	x	x	x	x	x	x	010
bne	01	x	x	x	x	x	x	110
slt	11	x	x	x	x	x	x	111

Una part del seu disseny el teniu descrit a les transparències de classe a la descripció del processador monocicle.

```
entity ControlALU is
Port (   funcio :           in STD_LOGIC_VECTOR (5 downto 0);
        ALUOp :           in STD_LOGIC_VECTOR (1 downto 0);
        Control_ALU :     out STD_LOGIC_VECTOR (2 downto 0)
); end ControlALU;
```

## 5. Realització pràctica

### 5.1. Disseny de l'ALU de 32 bits

Realitzeu el disseny corresponent a l'ALU de 32 bits explicat i verifiqueu el seu correcte funcionament amb el simulador.

### 5.2. Disseny de la unitat de control de l'ALU de 32 bits

Realitzeu el disseny corresponent a la unitat de control de l'ALU de 32 bits explicat i verifiqueu el seu correcte funcionament amb el simulador.

### 5.3. Símbol de l'ALU de 32 bits i de la unitat de control de l'ALU (opcional)

Amb el disseny de l'ALU creat a l'apartat anterior creeu uns símbols nous. Amb l'editor d'esquemàtics del "Quartus" dissenyeu un esquema que utilitzi aquest nou component (només l'heu de connectar a unes entrades i sortides) i verifiqueu el seu correcte funcionament amb el "ModelSim". El simulador "ModelSim-Altera" només permet la simulació de codi descrit en un llenguatge de descripció de maquinari. Per aquest motiu, si es vol simular l'esquemàtic realitzat, primer s'haurà de transformar a un llenguatge d'aquest tipus. Quartus permet fer aquesta transformació mitjançant la comanda "Create HDL Desing File From Current File" dins de l'opció "Create/Update" del menú "File", que fa que s'obri una finestra que ens deixa triar si volem transformar-ho a VHDL o a Verilog HDL (Pot ser qualsevol dels dos, però us aconsello el VHDL). Aquesta acció genera un codi escrit en Verilog o VHDL, que serà el que hem de posar com a "Top-Level Entity", per tal de poder fer la simulació. A partir d'aquí, ja podem fer la simulació normal. Feu el mateix amb la unitat de control de l'ALU.

En tots els apartats, és aconsellable fer un joc de proves el **més complet** possible. Com ha mínim hi ha d'haver quatre proves per a cadascuna de les operacions i que els resultats mostrin com s'activen els dos bits d'estat (Z i C).

**Presentació de la practica:** Aquesta practica està pensada per que la comenceu a fer en un sessió de laboratori. Al realitzar-se totalment sobre programari que el podeu aconseguir fàcilment, la resta de la practica l'haureu d'acabar vosaltres en altres hores.

S'ha de demostrar que funciona correctament a la propera sessió de pràctiques (el dia 20 de març).

## 6. Preguntes

- Hi ha un temps mínim per poder fer modificacions als senyals d'entrada?
- Quin és el temps de resposta dels senyals de sortida?

S'ha de fer una memòria amb els codis del programa de VHDL comentats i unes pantalles amb els resultats de la simulació. També s'han de respondre les dues qüestions plantejades a l'apartat 6.

Les pràctiques s'han de realitzar pels dos integrants del grup i la nota serà per tant la mateixa per a tots dos.

La nota de cada pràctica s'avalua segons els tres punts següents:

- Treball previ (si n'hi ha)
- Funcionament correcte.
- Documentació.

Per poder realitzar la pràctica en el temps previst és necessari que l'alumne realitzi el treball previ, quan així ho estableixi la pràctica. A l'inici de cada sessió de laboratori el professor recollirà aquest treball previ per a la seva avaluació (el que implica que l'alumne ha de tenir una còpia d'aquest treball per utilitzar-lo durant la sessió). Aquesta part valdrà el 20% de la nota.

És imprescindible per aprovar el Laboratori haver realitzat totes les pràctiques assignades. En el cas de suspendre una pràctica l'alumne tindrà una setmana per fer les correccions pertinents.

La durada de cada pràctica serà d'una sessió tret que s'indiqui el contrari. La memòria de la pràctica es lliurarà a l'Atenea com a màxim el dijous de la setmana que s'ha presentat. A efectes de càlcul de la nota mitjana final, una setmana de demora significarà que la nota es dividirà per 2, si la demora és més gran la nota serà de zero punts. El fet de superar les dues setmanes de demora no exclou el fet d'haver de lliurar la pràctica, ja que cal lliurar obligatòriament totes les pràctiques.

Algunes de les pràctiques contenen un apartat opcional de caràcter voluntari que donarà l'oportunitat a l'alumne d'augmentar la nota obtinguda en els apartats obligatoris de la pràctica, sempre que aquesta sigui superior a 5.

## NORMES GENERALS PER A LA REDACCIÓ DE LA DOCUMENTACIÓ

1. La documentació ha de ser concisa i clara.
2. Seguir el guió de la pràctica però sense copiar l'enunciat, excepte de forma esquemàtica.
3. Per les pràctiques en el laboratori d'ordinadors, recordeu que tots els arxius generats s'esborren en finalitzar la sessió. Es recomana crear una carpeta, per exemple a l'escriptori, per emmagatzemar aquests arxius i copiar en finalitzar la sessió a un disc USB. S'obtindrà còpia impresa només dels fitxers font: esquemes i codi VHDL.
4. Si s'inclouen diagrames temporals han de tenir la mateixa escala de temps i estar sincronitzats entre si.

## NORMES D'ÚS DEL LABORATORI

1. Preparar la pràctica abans d'anar al laboratori, segons les indicacions de cada pràctica.
2. Urbanitat: neteja, nivell de soroll baix, etc.
3. El lloc de treball serà fix i es faran servir els aparells correctament, guardant a l'armari en finalitzar el seu ús.
4. Forma part de la pràctica descobrir els possibles errors en el material utilitzat, així com en els components i circuits integrats emprats en la realització de la mateixa.
5. Per si es penja l'ordinador, és recomanable fer còpia de seguretat de tant en tant en un disc USB.
6. No introduir programes per evitar virus informàtics ni navegar per Internet sense autorització expressa del professor.
7. Està prohibit menjar, beure, fumar i jugar ( amb l'ordinador o sense ) al laboratori.