Departament de Disseny i Programació de Sistemes Electrònics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Arquitectura de computadors

Pràctica 2

Disseny de la unitat Aritmètica i Lògica (ALU)

Antoni Escobet

PRÀCTICA 2 – Disseny de la Unitat Aritmètica i Lògica

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.
- Amplia els coneixements sobre el llenguatge VHDL i l'edició esquemàtica de circuits amb QUARTUS II.

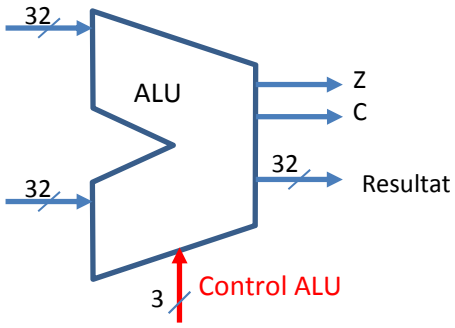
2. Material

L'únic material necessari per la realització d'aquesta pràctica és el paquet de programari d'ALTERA, el Quartus II web edition 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" que proporciona el mateix paquet de programari d'Altera

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.



El disseny del processador monocicle 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

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

La unitat aritmètic lògica tindrà la següent definició d'entrades i sortides:

```
entity ALU32 is
port (   ControlALU: in STD_LOGIC_VECTOR (2 downto 0);
         Op1, Op2: in STD_LOGIC_VECTOR (31 downto 0);
         Res: out STD_LOGIC_VECTOR (31 downto 0);
         Z, C: out STD_LOGIC);
end ALU32
```

on ***OP1*** i ***OP2*** són els dos operands, ***Res*** 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 ***ControlALU*** són les línies de control que li indicaran a la unitat aritmètic i lògica el tipus d'instrucció a realitzar. A la Taula es mostren els codis que indiquen el tipus d'operació a realitzar per l'ALU.

ControlALU	Funció
0 0 0	AND
0 0 1	OR
0 1 0	Suma
1 1 0	Resta
1 1 1	Activar si menor que

4.1. Funcions

4.1.1. 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.

Res = OP1 *and* OP2

4.1.2. OR

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

Res = OP1 *or* OP2

4.1.3. Suma

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

Res = OP1 + OP2

4.1.4. Resta

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

Res = OP1 - OP2

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

Aquesta operació genera 1 si Rs < Rt, i 0 en qualsevol altre cas. Exemple:

slt $1,$2,$3 significa que
if ($2 < $3) then $1 = 1;
else $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).

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. Símbol de l'ALU de 32 bits

Amb el disseny de l'ALU creat a l'apartat anterior creeu un símbol nou. 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".

En tots els apartats, és aconsellable fer un joc de proves el més complert possible.

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 d'ensenyar que funciona correctament a la propera sessió de pràctiques (el dia7 d'abril).

6. Preguntes

a) Heu pogut fer la simulació dels dos dissenys de la mateixa forma? Quines heu utilitzat i per quin motiu?
b) Hi ha un temps mínim per poder fer modificacions als senyals d'entrada?
c) Quin és el temps de resposta dels senyals de sortida?

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