

Nom:

Cognoms :

RISC-V (2 punts)

1. Considereu el tradicional RISC-V de 5 etapes de pipeline (F, D, EX, MEM i WB) estudiat durant el curs (Figura 1). Considereu que els registres del pipeline estan situats on les línies discontinües.

Per a aquesta qüestió, considereu el següent:

- No hi ha forwarding
- El branch predictor SEMPRE prediu que el branc no saltarà (not taken). Si s'equivoca, buida el pipeline.
- No podem llegir i escriure el mateix registre o adreça de memòria en el mateix cicle de rellotge
- No hi ha cap altre optimització implementada al processador.

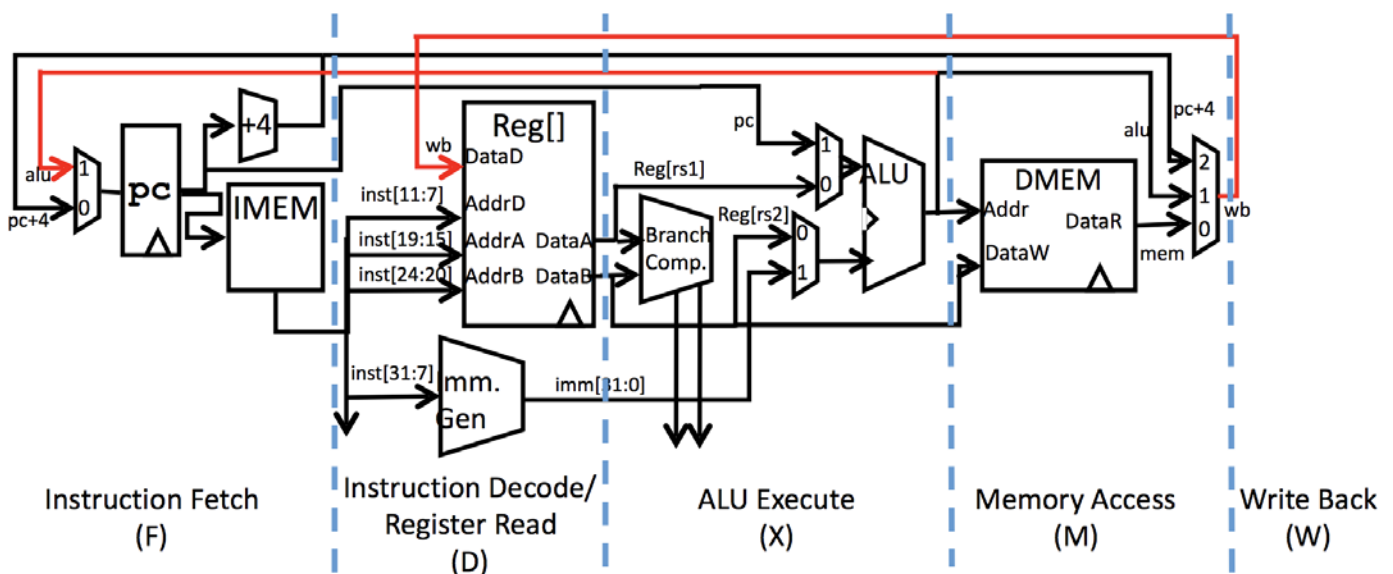


Fig.1: Esquema simplificat del nostre processador

- a) Ompliu una taula com la de la següent pàgina demostrant com seria l'execució del següent programa (la primera instrucció ja està feta a la taula): (1.5 punts)

Codi: Registres amb nomenclatura RIPS (o MIPS)

```
ori s1, x0, 0xF

andi s2, x0, 0

beq s1, s2, exit

lw s1, 0xc(s0)

xor s1, s1, s2

exit:

lw s1, 0xc(s0)
```

Nom:
Cognoms :

[illegible]

(l'execució del codi pot requerir més o menys cicles dels tabulats)

En el qüestionari haureu de respondre preguntes com:

- Nombre de cicles necessaris per executar el programa **16 cicles**
- Nombre de noops introduïts **6 noops**

b) Considereu que els màxims retards en cada etapa son:

1. F: 200 ns
2. D: 150 ns
3. EX: 100 ns
4. MEM: 300 ns
5. WB: 250 ns

- i. Quin és el temps mínim per executar una instrucció? **1500 ns** (0.25 punts)
- ii. Quin és el throughput en el millor cas? Dir d'una altra manera, quin és el temps mínim, en el millor cas, que ha de passar per obtenir el resultat d'una instrucció? **1/300 ns** (0.25 punts)



Nom:

Cognoms :

CACHÉ (3 punts)

2. Considereu que tenim una sola caché de dades (L1) amb les següents característiques:

- Mida de 4 KiB
- Blocs de 16 B
- Mapejat directe

La següent part de codi s'executa en un processador amb adreces de 32 bits, on cada int té una mida de 4 B:

```
#define SIZE 8192 // 213
int ARRAY [SIZE]; // note: extra aligned: ((int) ARRAY) %64 == 0
int main() {
    ARRAY [0] = ARRAY[4] + ARRAY[8]; // BEFORE LOOP 1

    for (int i=0; i < SIZE -16; i += 4) { // LOOP 1
        ARRAY [i] += ARRAY[i+4] + ARRAY[i+8] + ARRAY[i+12];
    }

    for (int i=SIZE -1; i >= 0; i -= 32) { // LOOP 2
        ARRAY [i] += 10;
    }
}
```

a) Calculeu el nombre de bits emprats per al TAG, index i offset (0.3 punts):

TAG = 20 bits, Index = 8 bits, Offset = 4 bits

b) Quin és el hit rate en el **LOOP 1**? Considereu que la caché comença inicialment buida quan executem main() (1 punt): **$h = 4/5$**

Abans de loop 1 hem portat els blocs ARRAY[0], ARRAY [4] i ARRAY[8], per tant al començar el loop els tenim a caché. En la primera iteració no tenim ARRAY[12]. Així que fem 4 hits (llegir ARRAY[0], ARRAY[4], ARRAY[8] i **ESCRIURE** ARRAY[0]) i un miss. Això ho repetim durant tot el loop

c) Quin és el hit rate per al **LOOP 2** si no hem fet cap reset a la caché després d'executar LOOP 1? (1 punt): **$h = 9/16$**

Operem de manera similar a l'anterior considerant que ara el loop és decreixent i que tenim diversos blocs guardats a memòria. Recordeu que realitzem una lectura i una escriptura.

d) Considereu que necessitem 100 cicles de rellotge per accedir a memòria; accedir a dades que es troben a la caché ens porta 5 cicles de rellotge. Considereu també que el hit rate de LOOP 1 és del 60% i que el hit rate de LOOP 2 és del 75%. El rellotge del sistema té una freqüència d'1GHz. Quin és el temps mig d'accés a caché per al: (0.7 punts)

i. LOOP 1:

De la teoria $T_{med} = h T_{ac} + (1-h) * T_{amp} = 0.6 * 5 \text{ ns} + 0.4 * 100 \text{ ns} = 43 \text{ ns}$ (no del tot correcte aquest cop però s'accepta).

També hi ha AMAT (Average Memory Access Time) = $T_{ac} + (1-h)*T_{amp} = 45 \text{ ns}$ (41 ns si heu fet look-aside)

ii. LOOP 2:



Nom:

Cognoms :

De la teoria $T_{med} = h T_{ac} + (1-h) * T_{amp} = 0.75 * 5 \text{ ns} + 0.25 * 100 \text{ ns} = 28.75 \text{ ns}$ (no del tot correcte aquest cop però s'accepta).

$AMAT = T_{ac} + (1-h) * T_{amp} = 30 \text{ ns}$

(27.5 ns si heu fet look-aside)



Nom:

Cognoms :

Memòria principal (1 punt)

3. A l'etiqueta d'un mòdul de memòria RAM que hem d'instal·lar a un ordinador trobem la següent informació: DDR2-800 (5-5-5-3). (1 punt)

Handwritten notes: DDR 2 → per cada flanc de rellotge 64 bits = 8B

Freqüència teòrica del BUS?	400 MHz
Ample banda teòric de cada mòdul de M.P.?	6400 MiB/s
Quants cicles de bus es necessiten per fer la transferència de 8 Bytes?	11 cicles <i>t = t_{RCO} + C₂ + 1 cicle = 11 cicles</i>
Quants cicles de bus es necessiten per fer la transferència de 16 Bytes contigus?	11 cicles <i>→ 11 cicles</i>
Quina latència teòrica té (en ns)?	5 * (1/400 MHz) = 12.5 ns

Administració de memòria (2.5 punts)

4. Una matriu de 2048×2048 , amb elements de 16 bits, es normalitzarà de la manera següent: Per a cada columna, es troba l'element més gran i tots els elements de la columna es divideixen pel valor d'aquest element. Assumeix que cada pàgina de la memòria virtual consta de 4KiB i que 1MiB bytes de la memòria principal s'assignen per emmagatzemar dades de matriu durant aquesta computació. Assumeix que es triga 10 ms a carregar una pàgina del disc a la memòria principal quan es produeix un error de pàgina.
- Suposeu que la matriu es processa una columna a la vegada. Quantes falles de pàgina es produiria i quant temps triga a completar el procés de normalització si els elements de la matriu s'emmagatzemen en ordre de columna en la memòria virtual (primer tots els elements de la columna 0, a continuació els de la columna 1, etc)? (0.75 punts)
 - Repetiu la part (a) assumint que els elements s'emmagatzemen en ordre de fila (primer tots els elements de la fila 0, a continuació els de la fila 1, etc)? (0.75 punts)
 - Proposar una manera alternativa de processament de la matriu per reduir el nombre de falles de la pàgina quan la matriu s'emmagatzema a la memòria com en l'apartat (b) (1 punt)

Aquest problema és exactament igual que el problema 3.1 de la col·lecció de problemes. El nombre de columnes i de files és el doble, però el nombre de bits dedicats a cada element de la matriu és la meitat. Així doncs, una fila o una columna completa ocupa 4 KiB, que és la mida de la pàgina. Com ja he dit, l'explicació la teniu a problemes, els resultats actualitzats son:

- $T = 2048 \text{ faltes} * 10 \text{ ms/falta} = 20.48 \text{ s}$ per dur les pàgines de memòria secundària a memòria principal.
- $T = (2048)^2 * 10 \text{ ms/falta} = 83886.08 \text{ s} \Rightarrow 23.3 \text{ hores} !!!$
- $T = 2048 * 2 * 10 \text{ ms/falta} = 40.96 \text{ s}$



Nom:
Cognoms :

Qüestions teòriques (1.5 punts)

El temps per contestar les **qüestions de teoria és limitat (30 min màxim)**. Veureu que teniu un qüestionari amb preguntes teòriques tipus test al campus. N'hi ha 12, l'encert puntua 0.25 punts, l'error resta 0.1 punts. Considereu si val la pena respondre a l'atzar o deixar en blanc la resposta.