

Nom:

Cognoms :

1. (4 punts) Donat el camí de dades d'un RISC-V single-cycle estàndard (figura 1) per a la instrucció “blt” (not taken), **quin és el valor dels senyals de control?** Utilitzeu 0, 1 o * per als senyals i N/A si el senyal donat no és proporcionat per la lògica de control; per a l'ALU, suposem que 0 es refereix a l'operació d'addició i 1 a la resta (sub). Per al selector d'immediats, utilitzeu I per a tipus immediat, S per a tipus store, B per a tipus branch, J per a tipus de salt i U per al tipus upper-immediate. **(Raoneu breument la resposta)**

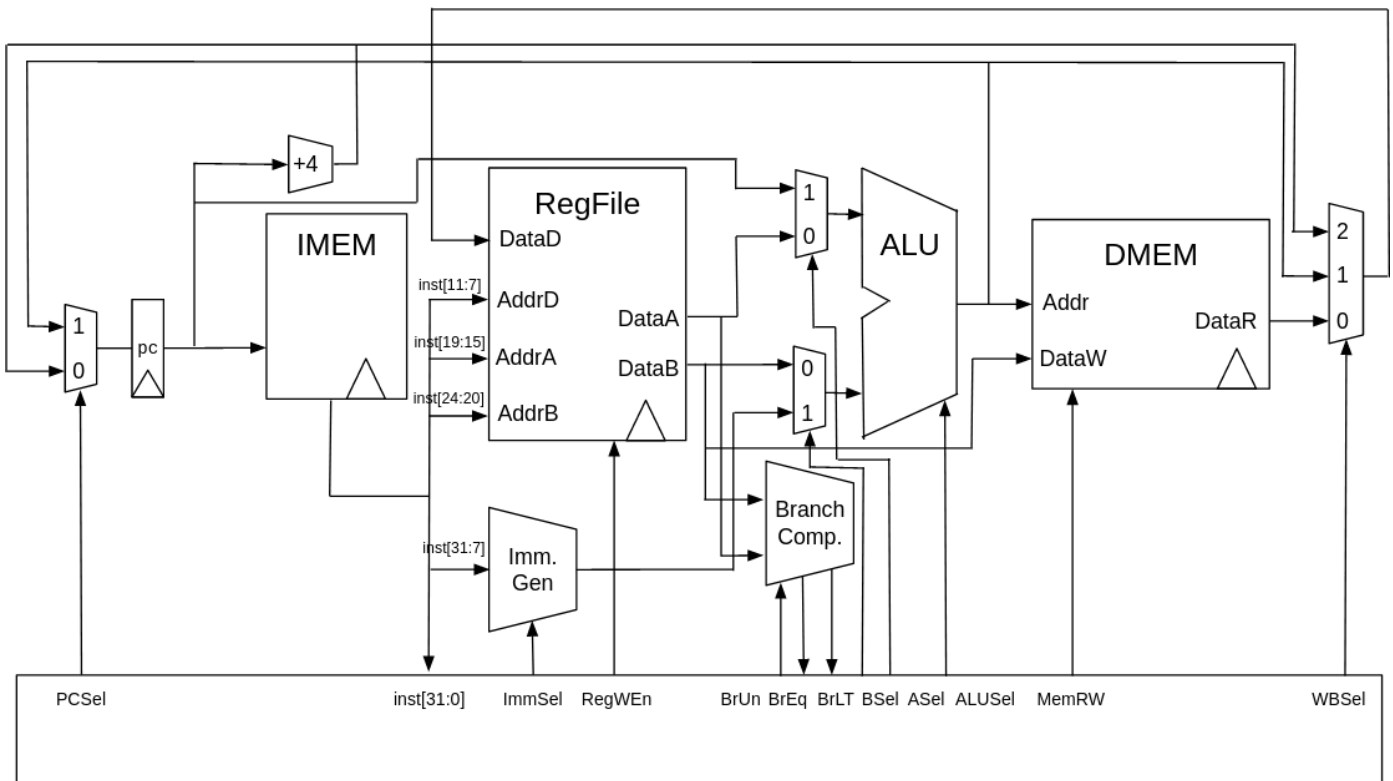


Fig.1

Instrucció **blt** (not taken)

- a) PCSel: **0**
- b) Inst[31:0]: **N/A**
- c) ImmSel: **B**
- d) RegWEn: **0**
- e) BrUn: **0 o 1**
- f) BrEq: **N/A**



Nom:
Cognoms :

g) BrLt: **N/A**

h) BSel: **1**

i) ASel: **1**

j) ALUSel: **0**

k) MemRW: **0**

l) WBSel: *****

Nom:

Cognoms :

2. (5 punts) La figura 2 mostra el camí de dades (datapath) d'un processador RISC-V amb 6 etapes de pipeline. És similar al processador de 5 etapes estudiat durant el curs però hem dividit l'etapa de memòria en dues etapes diferents. Suposem que **no** hi ha **forwarding**, **no** hi ha **branch prediction** i que només podem fer **una operació** amb un mateix **registre** per **cicle**.

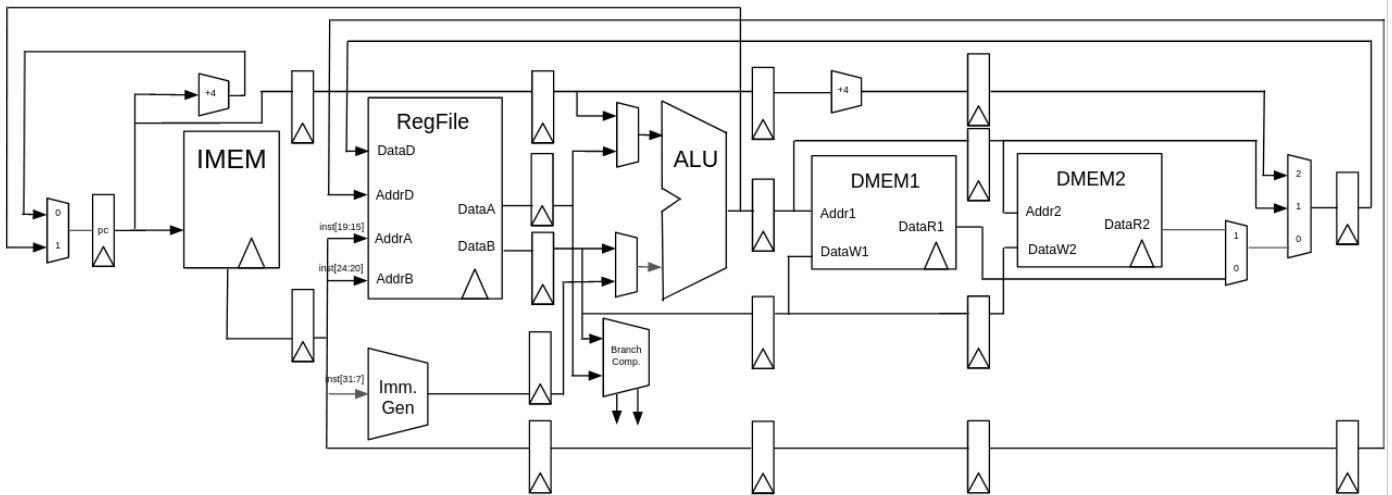


Fig.2.

Contesteu les següents preguntes:

- a) (0.5 punts) Per quin motiu voldríem dividir l'etapa de memòria en dues etapes diferents?

Degut a que deu ser l'etapa més lenta, això fa que tot el pipeline hagi de funcionar al seu ritme. Al dividir-la és possible que augmentem la latència de la instrucció però augmentarem la freq. De rellotge, això vol dir que resoldrem més instruccions per unitat de temps.

- b) (0.5 punts) Millorem l'IPC? Millorem el CPI ?

- 1) Si no tenim dependències de dades CPI hauria de seguir tendint a 1
- 2) Afegir una nova etapa de memòria en un processador sense mesures d'optimització (forwarding, branch prediction, etc.) fa que calgui esperar un cicle més per tenir els resultats a WB, per tant potencialment el CPI empitjorarà quan hi hagi hazards de dades. L'IPC és l'invers del CPI, per tant empitjora en la mateixa mesura.



Nom:

Cognoms :

- c) (1 punt) Explica els avantatges i desavantatges d'un processador segmentat respecte un d'un sol cycle. O dit d'una altra forma, quins avantatges i desavantatges té el datapath del problema 2 si el comparem amb el datapath del problema 1 ?

el segmentat té un temps de cycle menor, però té hazards

Volem executar el següent codi:

1. bne x0, t0, next
2. addi t1, t0, 1
3. lb s0, 0(t1)
4. shw s0, 4(t0)

Quins hazards podem trobar a cada línia de codi i a què son deguts? (Marqueu la opció correcta, si marqueu la incorrecta resta el valor de la pregunta a la vostra nota)

d) Línies 1/2 (0.33 punts)

- i. Estructural
- ii. Cap
- iii. **Control**
- iv. Dades

e) Línies 2/3 (0.33 punts)

- i. Estructural
- ii. Cap



Nom:

Cognoms :

iii. Control

iv. Dades

f) Línies 3/4 (0.33 punts)

i. Estructural

ii. Cap

iii. Control

iv. Dades

g) Quin és el nombre mínim de NOOPs que necessitem afegir per executar el codi correctament? (2 punts)

Resposta nº NOOPs: **10** (si no ompliu la taula no la considerarem vàlida encara que sigui correcte)

	Temps →																						
Instrucció	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Nom:

Cognoms :

3. (1 punt) El processador RISC-V single-cycle de la següent figura (figura 3) suporta la ISA RV32I.

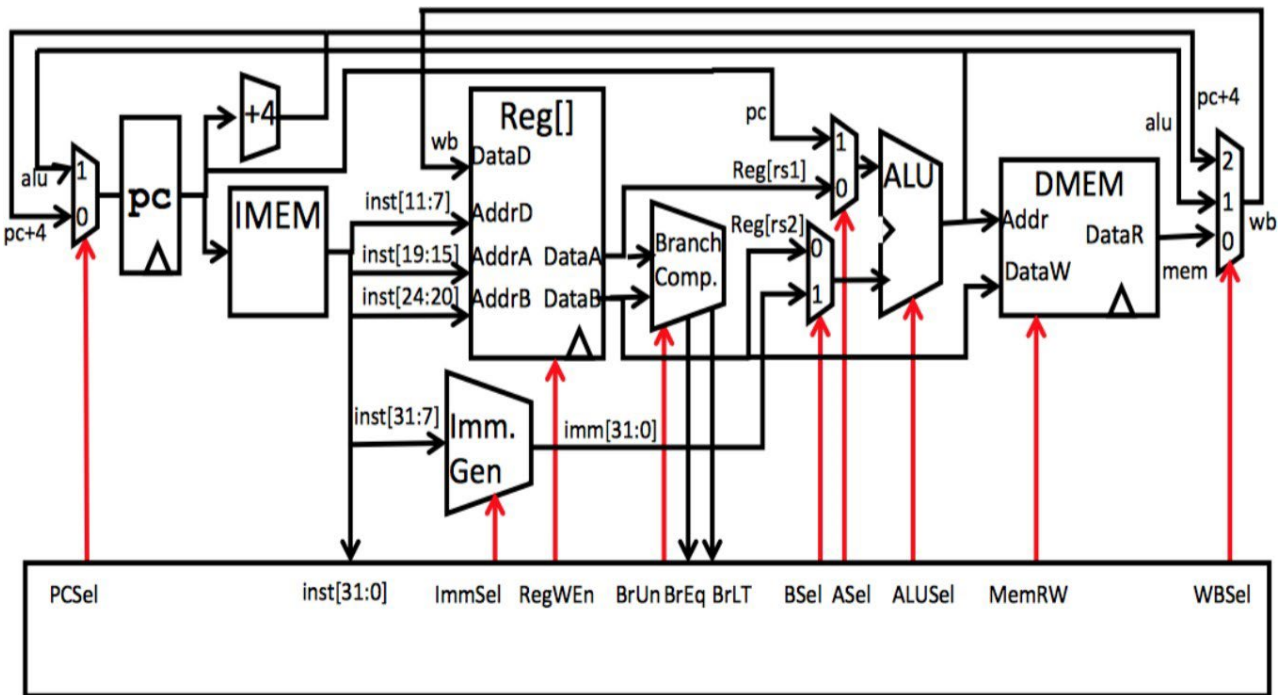


Fig. 3.

Volem ampliar el nostre conjunt d'instruccions per donar suport a algunes instruccions noves. En particular, com el complement a 2 és molt propens a errors, volem implementar la instrucció de negació:

neg rd,rs1, que posa $-R[rs1]$ a $R[rd]$

Quina de les següents afirmació és correcta per a la nova instrucció i per què?:

- a) La nova instrucció es pot implementar sense grans canvis de cablejat en el datapath, només calen canvis en els senyals de control (és a dir, canviar les senyals de control existents per reconèixer la nova instrucció)
- b) La nova instrucció es pot implementar, però necessita canvis en el cablejat en el datapath. Només calen cablejats addicionals i algunes portes lògiques i muxes.
- c) La nova instrucció es pot implementar, però necessita canvis en el cablejat en el datapath i es necessiten unitats aritmètiques addicionals (per exemple, comparadors, sumadors, desplaçaments, etc.).
- d) La nova instrucció no es pot implementar.

Resposta: (recordeu argumentar-la):

(A). Notice **neg** doesn't use all available bits, so we could make **neg rd, rs1** into a special R-type instruction **neg rd, x0, rs1**, such that the instruction does $R[rd] = x0 - R[rs1]$. Notice that subtraction is supported by our default datapath. So we only need to add the new control signal **neg**, which will produce the same ALUSel, ASel, Bsel, ... signals as **sub** does.