

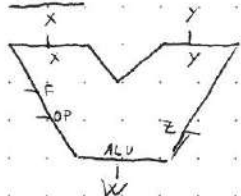
Unitat de Process General (8)

Estructura

REG FILE = Component que és un conjunt de Registres.

"Busca dades per ser operades" → Genera el càlcul → Resultat el porta al Banc de Registres."

ALU



"F" i "OP" són bits de Control

2 bits categoria 3 bits de funcionalitat

"Z" detecta si operació resulta '0'.

La ALU calcula tot però després es selecciona el resultat.

Segueix hi ha algo que es calcula en la ALU (Z) tot i que només es deixa passar.

SHA (Desplacador Aritmètic) = Mou X i Y d'un com en mou i cap a on.

5 bits mags per de Y → $\oplus [0, 15]$ = Esquerra (Multi)
→ $\ominus [-16, -1]$ = Dreta (Div)

La magnitud de Y és el nombre de desplaçant.

Com a màxim es fa desplaçant 2^0 quan és negatiu Y (en a2).

Això és pq en negatiu no pots dir fer '0' desplaçant, màxim pots dir "-1".

Banc de Registre

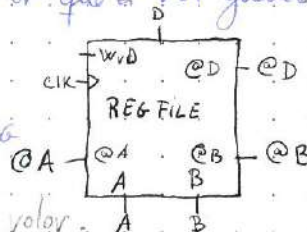
Port de lectura @A i @B. @S'encarrega per valor de A (3 bits = 8 posicions).

Port d'escriptura @D (Em quin registre); D (Valor que es vol guardar).

'W_RD' Verifica si s'ha d'escriure o no.

És el Enable d'un DEC-3-8 del @D; D va a tots els REG.

Si W_RD = '0' tots els REG (que tenen Ld) mantindran el valor.



Registres de 16 bits.

UPG

W_RD MAI pot ser 'x'.

RD-In: Bus dades que entren. 1^a MUX: Discrimina si guardar dades gen. per ALU o que "ve" de fora (RD-in).

WR-Out: Dades que surten.

2^a MUX: Permetre fer operacions amb nombre constants (N).

33 b de Control a total

⚠ Només hi ha 4 bit de Condicions (UPG → UC) i en la Z.

"ADD R3, R1, R5" → Nombres a operació
→ On es guarda el resultat

Com que és limitant les paraules de Control, pot ser que (per exemple) es fiqui W_RD abans que fer el càlcul.

R = Registre

Si la constant (N) no importa 'xxxx' → a càlcul

Algoritmo a graf

if ($R2 \leq R3$) $R3 = R1 - 1$; (IN)



CMPLTEU -R2, R3

SUBI R3, R1, 1

⚠ Se $R2 \leq R3$ estiver:
- ALU: '1'
- Z: '0'

I com que temos em capte Z,
ho hem. de fer inversa.

if ($R3 > R5$) $R2 = R2 + R2$;
else $R3 = R3 - 5$;

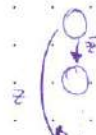
(Z)

ADDI R3, R3, -5 } Em aquil cas
SUBI R3, R3, 5 } sem equivalentes

Nota: em
ordem

SHAI R2, R2, 1 ← Com que a posicao, mais esquerda.

if ($R1 < 15 > == 1$) $R2 = R2 + 1$; (N)



ANDI -R1, 0x8000

#8000 = 1000 0000 0000 0000

ADDI R2, R2, 1



MOVEI R3, 0x0001

SHAI R0, R3, 15

AND -R1, R0

ADD R2, R2, 1