

6

PUSH Ra

- 1) $MAR \leftarrow SP; AC \leftarrow Ra$ ($R_{SP}, W_{MAR}, TA, W_{AC}$)
- 2) $DATMEM(MAR) \leftarrow AC; SP \leftarrow SP - 1$ (R_{AC}, W_{MEM}, D_{SP})

7) ~~4)~~ ST, LD con post-increment.

Unidad de datos:

- Opción 1: añadir operación $INCB$ a ALU ($out = B + 1$)
 Opción 2: añadir operación IAC (inc. AC) a acumulador

ST R_b, R_a

- 1) $AC \leftarrow R_b$ (TB, W_{AC})
- 2) $MAR \leftarrow AC; AC \leftarrow R_b + 1$ ($W_{MAR}, R_{AC}, W_{AC}, INCB$)
- 3) $R_b \leftarrow AC; AC \leftarrow R_a$ (R_{AC}, W_{REG})
- 4) $DATMEM(MAR) \leftarrow AC$

Necesitamos escribir R_b

- I) $W_{REGA} + W_{REGB}$
- II) $ST R_b, R_a \rightarrow ? \rightarrow ST R_a, R_b$

7

ST Rd, Rf

Rbase → seleccionados en sb: iv [210]

LD Rd, Rf

* Suponemos no podemos cambiar cod. p. ni format. instrucción.

Necesitamos: 1) poder incrementar Rbase

2) guardar nuevo valor.

Solución: 1) operación INCB en ALU

2) una señal de escritura en banco de registros: WREGS.

ST Rd, Rf.

1) $AC \leftarrow R_d$ (TB, WAC)

2) $MAR \leftarrow AC$; $AC \leftarrow R_d$ (RAC, WMAR, TH, WAC)

3) $DATMEM(MAR) \leftarrow AC$; $AC \leftarrow R_d + 1$ (WMEM, RAC,

4) $R_d \leftarrow AC$ (RAC, WREGS) INCB, WAC)

LD Rd, Rf

1) $AC \leftarrow R_f$ (TB, WAC)

2) $MAR \leftarrow AC$, (RAC, WMAR)

3) $R_f \leftarrow DATMEM(MAR)$, $AC \leftarrow R_f + 1$ (RMEM, WREG, INCB, WAC)

4) $R_f \leftarrow AC$ (RAC, WREGS)



R0 LR1
R3 R2

Estategia : restar R1 de R0 hasta que $RC < 0$.

Cociente : nº de restas hasta (sin contar) $RC < 0$.

Resto : valor final de $RC + R1$.

```
LDI R2, 4      { Inicialización
LOOP SUB R0, R1 }
BRLO END      { Bucle principal
SUBI R2, $FF   ; ADDI R2, 1 : R2 = R2 - (-1)
JMP LOOP
END MOV R3, RC  }
ADD R3, R1     { Finalización
STOP
```

EDC-T1 Tema 3

9

```
LD R6, $34      ; Puntero a lista
LDI R1, 10      ; contador
LDI R4, 0       ; acumulador
} Inicialización

LOOP LD R2, R6   ; R2: registro auxiliar
ADD R4, R2      ;
SUBI R1, 1      ;
SUBI R6, $FF    ;
BRZS END
JMP LOOP

END STOP        ; resultado en R4
} Finalización

} Bucle principal
```

(10)

Micro-operaciones BRXX div. (original)

1) ~~IFZ~~ :

$$I_7=0, I_8=0, Z=1$$

$$I_7=0, I_8=1, C=1$$

$$I_7=1, I_8=0, V=1$$

$$I_7=1, I_8=1, N \oplus V=1$$

: AC ← div

(imm, TB, W_{AC})

2) (solo si se cumple)

$$PC \leftarrow AC \quad (R_{AC}, W_{PC})$$

Micro-operaciones modificadas.

$I_{10}=0 \rightarrow$ igual que antes.

$I_{10}=1 \rightarrow$ igual pero condición se cumple cuando los bits de estado tienen el valor opuesto:

$$Z=0, C=0, V=0, N \oplus V=0$$

Cambiamos la condición en función de I_{10} :

$$Z=1 \rightarrow Z \oplus I_{10} = 1 \quad \left. \begin{array}{l} Z=1, I_{10}=0 \\ Z=0, I_{10}=1 \end{array} \right\}$$

$$C=1 \rightarrow C \oplus I_{10} = 1$$

$$V=1 \rightarrow V \oplus I_{10} = 1$$

$$N \oplus V=1 \rightarrow N \oplus V \oplus I_{10} = 1$$

11

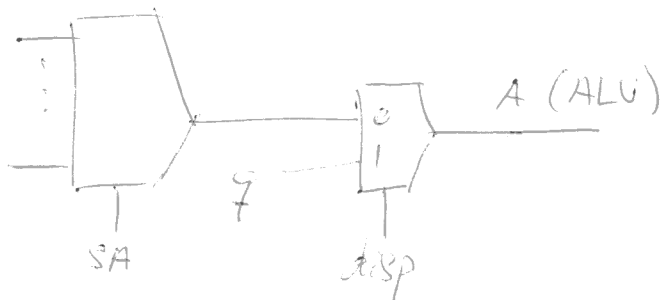
$$7 = I_{7..3}$$

$$0 \leq 7 < 32$$

Tenemos que hacer $MAR \leftarrow R_{base} + 7$.

1) En ST, LD, el registro base está en $sb = I_{2..0}$.
 Usamos el mismo registro base y modificamos la entrada a de la ALU para poder sumar un desplazamiento. Además, definimos un nuevo formato de instrucción.

	$I_{15..11}$	$I_{10..7}$	$I_{7..3}$	$I_{2..0}$
D	Cód. op.	registro base	7	n. base STD/LDD

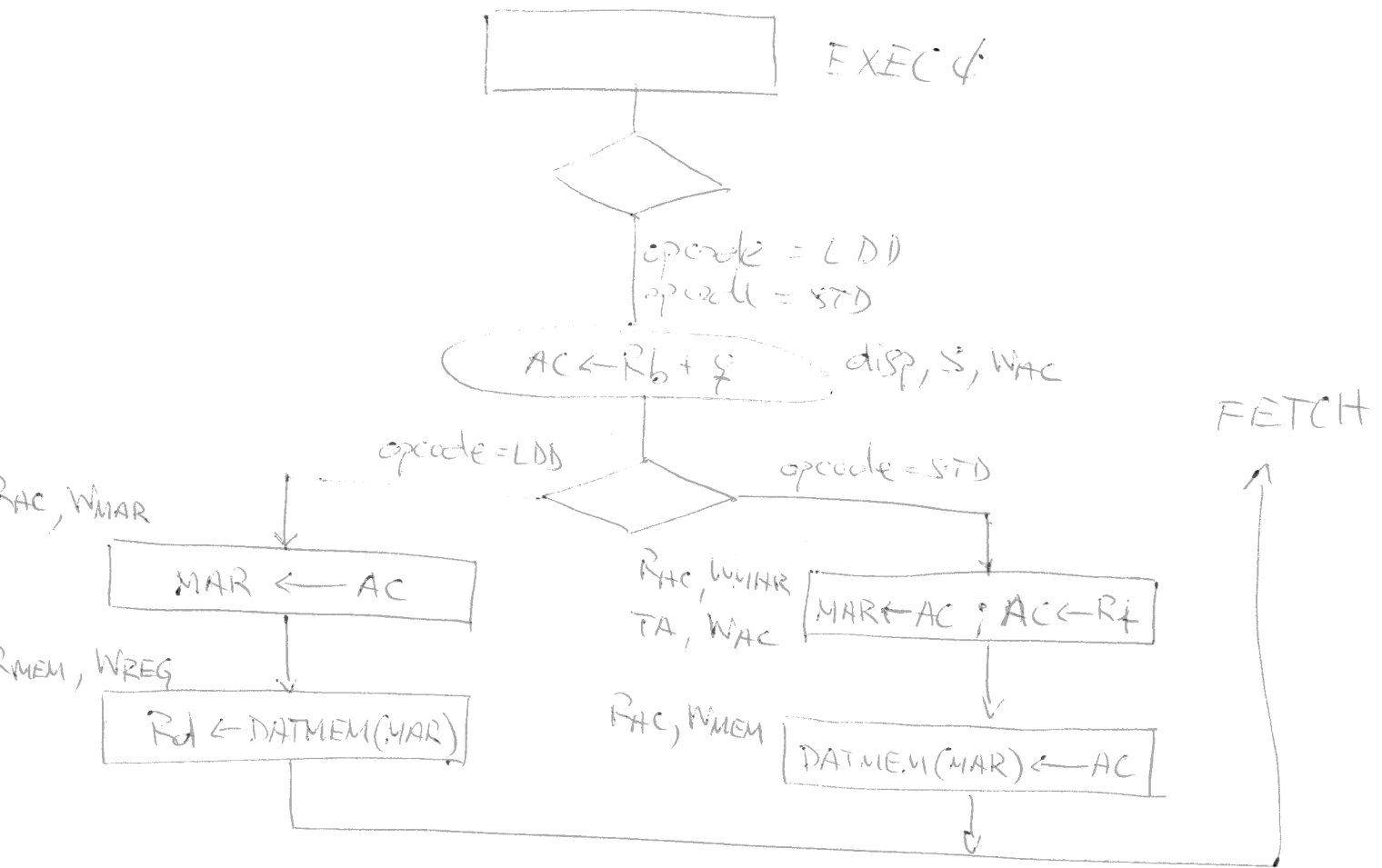


LDD $R_d, R_b + 7$

- 1) $AC \leftarrow R_b + 7$ ($ctrlsp, S, W_{AC}$)
- 2) $MAR \leftarrow AC$ (R_{AC}, W_{MAR}) \rightarrow como LD
- 3) $R_d \leftarrow DATMEM(MAR)$ (R_{MEM}, W_{REG}) \uparrow

STD $R_b + \delta, R_d$

- 1) $AC \leftarrow R_b + \delta$ (disp, S, W_{AC})
- 2) $MAR \leftarrow AC ; AC \leftarrow R_d$ (R_{AC}, W_{MAR}, TA, W_{AC})
- 3) $DATMEM(MAR) \leftarrow AC$ (R_{AC}, W_{MEM})



Pregunta:

* Utilidad de LDD y STD si ya tenemos LD, LDS, ST, STS.

* ¿Qué pasaría si la memoria de datos > 256 pal?

(12)

d) Hecho en pb. 10.

(13) La diferencia es que hay que almacenar el estado de la ALU en SREG. Ejemplo: ADD

ADD Rd, Rf

$$1) AC \leftarrow R_d + R_f; \quad \underbrace{SREG \leftarrow \{C, N, Z, V\}}_{CS3} \quad (\text{WREG})$$

(S, WAC)

$$2) R_d \leftarrow AC \quad (RAC, WREG)$$

(14) Se añade un acarreo de entrada a la ALU (Cin)

ADC Rd, Rf

$$1) AC \leftarrow R_d + R_f + C; \quad SREG \leftarrow \{C, N, Z, V\}$$

(S, Cin = C, WAC, WSR)

$$2) R_d \leftarrow AC \quad (RAC, WREG)$$