

Esercizio II-DLX

Calcolatori Elettronici L-A

Esercizio n. 1

Con riferimento al DLX con Datapath sequenziale visto durante il corso si indichi un possibile formato e il diagramma degli stati ottimizzato delle istruzioni

LWC Ri, (Rj), Rz

SWC (Ri), Rj, Rz

L'istruzione LWC Ri,(Rj),Rz nel caso in cui il registro Rz sia zero, prevede la lettura nel registro Ri della word memorizzato all'indirizzo di memoria specificato da Rj. Se Rz non è zero l'istruzione LWC non deve effettuare alcun accesso alla memoria.

L'istruzione SWC (Ri),Rj,Rz nel caso in cui il registro Rz sia zero, prevede la scrittura all'indirizzo di memoria specificato da Ri della word memorizzata nel registro Rj. Se Rz non è zero l'istruzione SWC non deve effettuare alcun trasferimento verso la memoria.

Esercizio 1

LWC Ri, (Rj),Rz

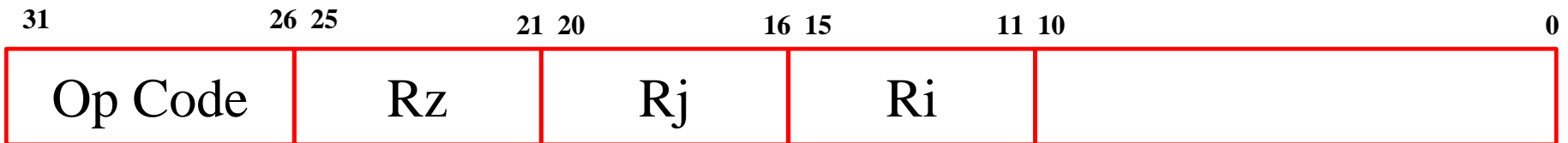
L'istruzione **LWC Ri, (Rj), Rz**

se $Rz = 0$ deve caricare in Ri il contenuto della memoria indirizzata da Rj .

Se $Rz \neq 0$ non deve svolgere alcuna operazione.

Il formato delle istruzioni DLX è composto da 32 bit.

Il codice operativo è codificato con 6 bit e i registri implicati nell'operazione (Ri, Rj, Rz) sono codificati con 5 bit (32 possibili registri). Il formato dell'istruzione è di tipo R.

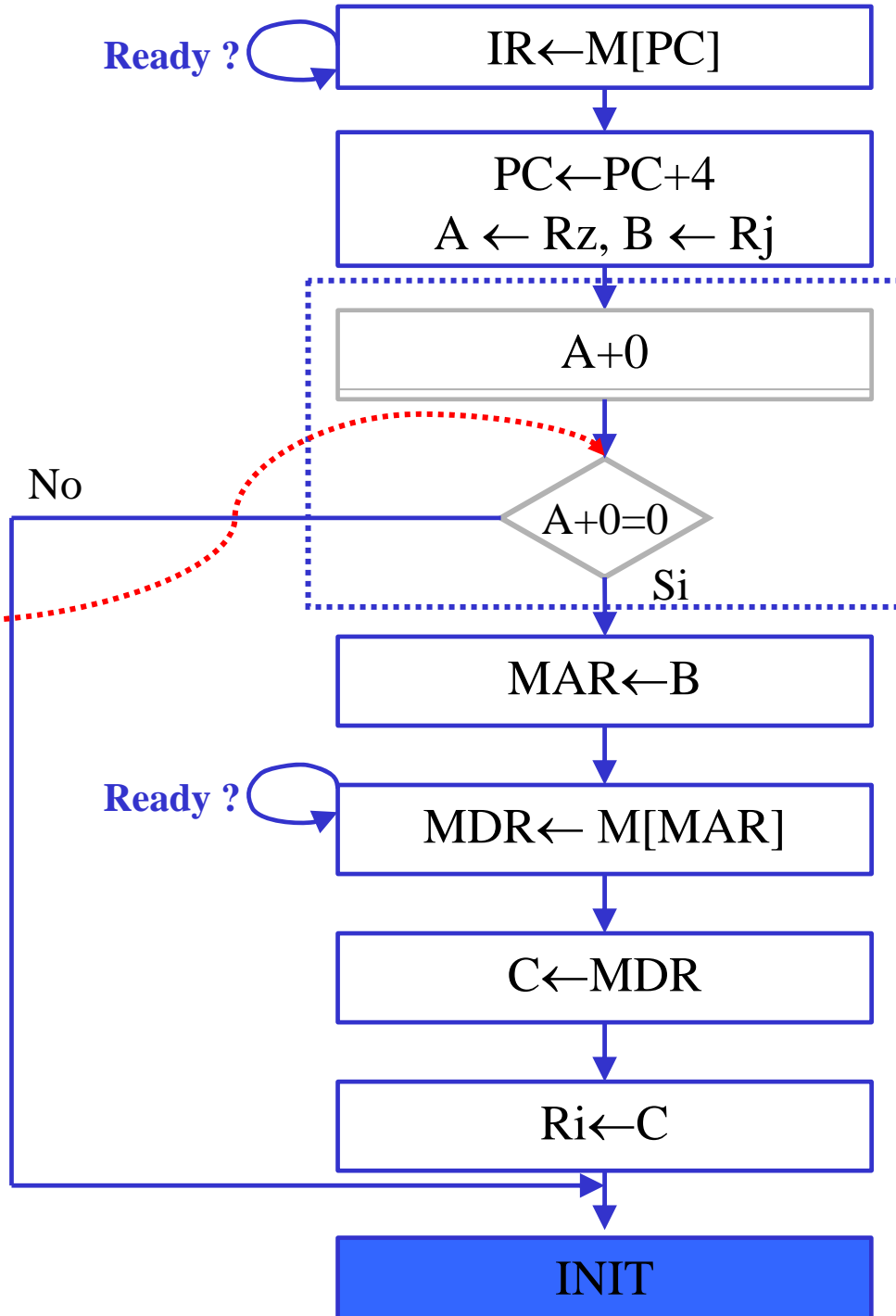


Il diagramma degli stati di questa istruzione è riportato nel lucido successivo.

Diagramma degli stati dell'istruzione

LWC Ri, (Rj),Rz

Il test della condizione $A=0$ consiste nell'analisi del flag di Zero (ZF) della ALU dopo l'operazione $A+0$.



Esercizio 1

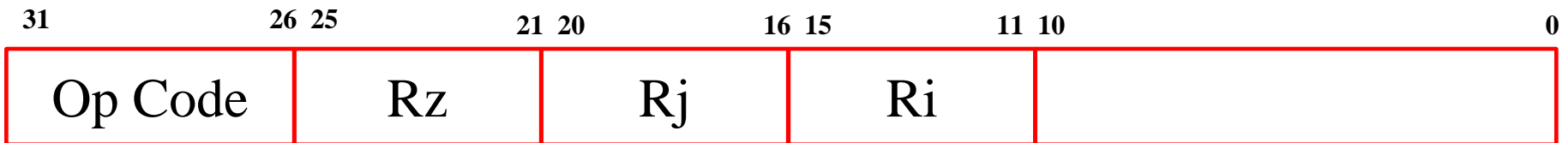
SWC (Ri), Rj,Rz

L'istruzione **SWC (Ri), Rj, Rz**

se $Rz = 0$ deve caricare all'indirizzo di memoria specificato da Ri il contenuto del registro Rj .
Se $Rz \neq 0$ non deve svolgere alcuna operazione.

Il formato delle istruzioni DLX è composto da 32 bit.

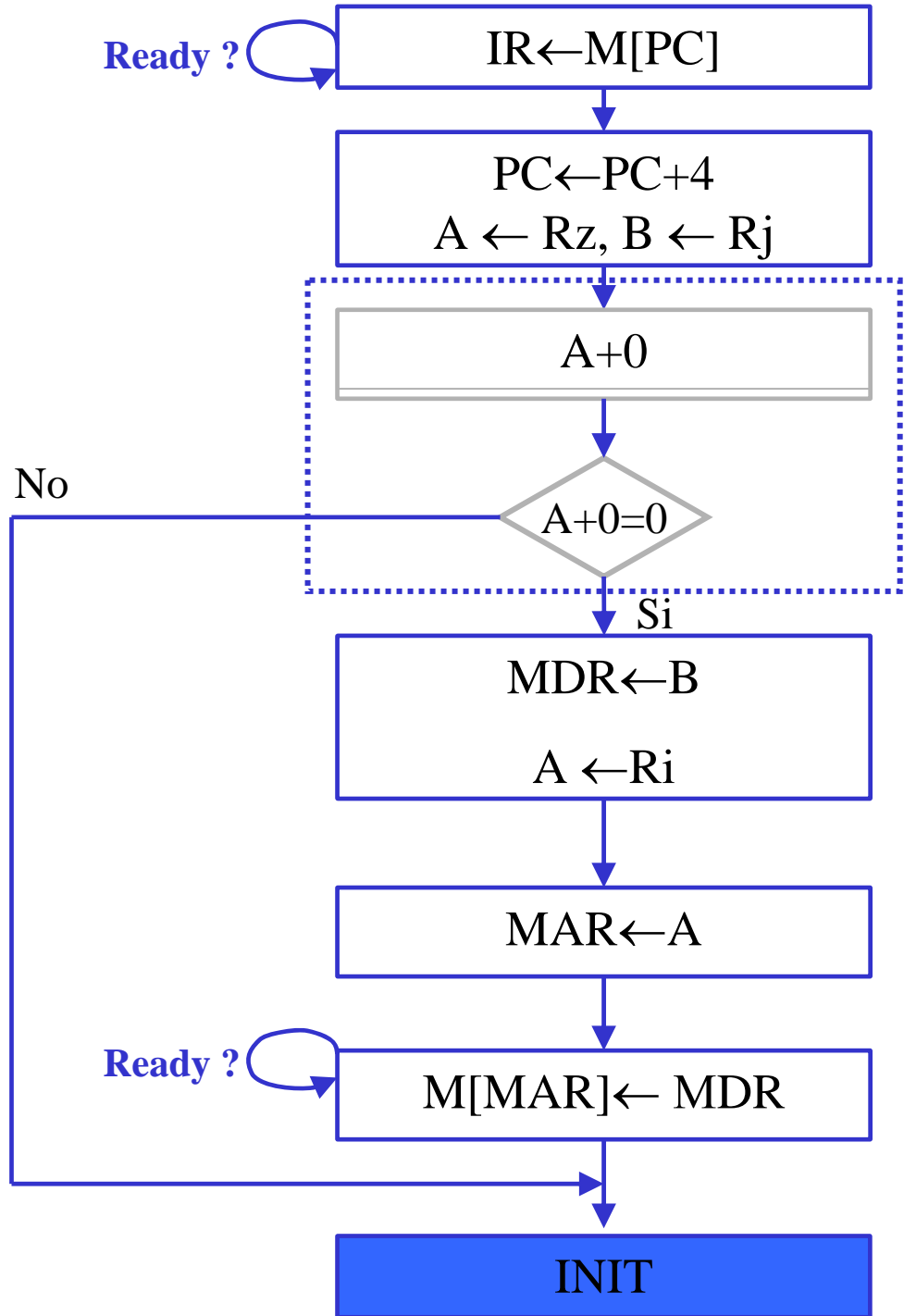
Il codice operativo è codificato con 6 bit e i registri implicati nell'operazione (Ri, Rj, Rz) sono codificati con 5 bit (32 possibili registri). Il formato dell'istruzione è di tipo R.



Il diagramma degli stati di questa istruzione è riportato nel lucido successivo.

Diagramma degli stati dell'istruzione

SWC (Ri), Rj, Rz



Esercizio n. 2

Considerando un sistema basato sul processore DLX non pipelined; si supponga di dovere sostituire tutti gli elementi nulli di un vettore di N elementi di tipo *word* con il valore memorizzato in R4.

2-1) Si scriva il codice assembler DLX non utilizzando istruzioni LWC e SWC

2-2) Si scriva il codice assembler DLX utilizzando anche le nuove istruzioni definite nell'Esercizio n.1.

2-3) Nei due casi: il numero di istruzioni dipende dal numero di elementi nulli nel vettore ?

Si supponga che l'indirizzo iniziale del vettore sia contenuto nel registro R1, l'indirizzo finale in R2 e che gli accessi alla memoria avvengono senza stati di wait.

Esercizio 2-1

Utilizzando unicamente le istruzioni originali del DLX.

R1 indirizzo primo elemento del vettore

R2 indirizzo ultimo elemento del vettore

R4 valore che sostituisce gli elementi nulli del vettore

```
    ADDI    R2,R2,4        ; indirizzo primo elemento vuoto
                                ; in coda al vettore
```

```
CICLO: LW    R5,0(R1)      ; carica nuovo elemento del vettore
```

```
    BNEZ    R5,LOOP       ; se R5 non è zero salta a LOOP
```

```
    SW     0(R1), R4      ; se è zero lo sostituisce con R4
```

```
LOOP: ADDI    R1,R1,4      ; incrementa indice vettore
```

```
    SUB     R6,R2,R1      ; diff. tra indice e ultimo valore
```

```
    BNEZ    R6, CICLO     ; se non è zero ripete il ciclo
```

Esercizio 2-2

Utilizzando l'istruzione SWC definita nell'esercizio 1.

R1 indirizzo primo elemento del vettore

R2 indirizzo ultimo elemento del vettore

R4 valore che sostituisce gli elementi nulli del vettore

ADDI R2,R2,4 ; indirizzo primo elemento vuoto
; in coda al vettore

CICLO: LW R5,0(R1) ; carica nuovo elemento del vettore

SWC (R1),R4,R5 ; se R5=0 scrive R4 all'ind. R1

ADDI R1,R1,4 ; incrementa indice vettore

SUB R6,R2,R1 ; diff. tra indice e ultimo valore

BNEZ R6, CICLO ; se non è zero ripete il ciclo

Esercizio 2-3

Nel caso 2-1 il numero di istruzioni dipende dal numero di elementi nulli del vettore.

Nel caso 2-2 il numero di istruzioni è indipendente dal numero di elementi nulli.