

Calcolatori Elettronici L A

Prova scritta del 14/03/'03

Esercizio N. 1 (11 punti)

1.1) Data la seguente sequenza di istruzioni eseguita su un processore DLX dotato di pipeline:

LW Rx, alfa(Ry)

ADD Rz, Rk, Rx

Motivando in modo chiaro la risposta, dire se è possibile progettare una Forwarding Unit capace di evitare l'alea di dato ed eventualmente disegnarne lo schema. (5 punti)

1.2) Data la seguente sequenza di istruzioni eseguita su un processore DLX dotato di pipeline:

ADD Rz, Rk, Rx

SUB Ri, Rj, Rz

Motivando in modo chiaro la risposta, dire se è possibile progettare una Forwarding Unit capace di evitare l'alea di dato ed eventualmente disegnarne lo schema. (6 punti)

(Per le risposte alle domande 1.1 e 1.2 si può eventualmente usare l'apposito schema della pipeline-DLX fornito con gli appunti delle lezioni. Per ciascuna risposta di questo esercizio considerare solo le ipotesi esplicitate nella rispettiva domanda)

Esercizio N. 2 (14 punti)

Un sistema a microprocessore basato su 8088 a 5 MHz dispone di 16KB di Eprom (1 chip agli indirizzi alti) e 128KB di Ram (2 chip contigui a partire dall'indirizzo 0h). Il sistema è dotato di una interfaccia parallela 8255 attiva in ricezione (modo 1 input) sulla porta A e gestita ad interrupt.

2.1) disegnare lo schema del sistema e progettare il circuito per la generazione del segnale di READY nell'ipotesi che le periferiche di I/O richiedano 1 Twait e le memorie non richiedano alcun Twait. (3 punti)

2.2) scrivere le espressioni di selezione semplificate delle memorie e dei dispositivi di I/O. (3 punti)

2.3) Disegnare lo schema del circuito necessario a far sì che il software possa leggere, ad un certo indirizzo di I/O, su un "bit" del bus dati se sia stato ricevuto il carattere 0Fh dalla porta A dell'interfaccia parallela. Progettare tale circuito in modo che resti traccia di un carattere 0Fh anche se in seguito sono stati letti caratteri differenti. Tale bit deve quindi essere resettato solo all'avvio del sistema o con una eventuale scrittura comandata dal software allo stesso indirizzo di I/O a cui viene letto il bit. (8 punti)

Esercizio N. 3 (8 punti)

3.1) Dire di quali registri ha bisogno un generico Stack implementato nella memoria centrale di un sistema a microprocessore e dire a cosa servono. (4 punti)

3.2) Descrivere le ipotesi relative alla "trama" ed alla frequenza di campionamento sulle quali si basa la ricezione di dati di un'interfaccia di comunicazione seriale asincrona RS232. (4 punti)

CONSEGNARE ANCHE IL TESTO ASSIEME ALL'ELABORATO

Durante il compito non si può uscire dopo le ore 11:00

Consegnando un elaborato per la correzione decade la validità del voto precedente

Correzione e registrazione il 28/03/03, ore 9:30 Via Rasi

I risultati, appena disponibili, saranno pubblicati in Segreteria e sul sito

<http://www.ingce.unibo.it> → selez. il proprio C.d.L. → piano degli studi → Calcolatori Elettronici L A

SOLUZIONE del compito di Calcolatori Elettronici L A del 14/03/03

1.1) Nell'implementazione in pipeline, l'istruzione ALU "ADD" ha bisogno del dato contenuto in Rx quando entra nella fase EX. Tuttavia questa fase dell'istruzione "ADD" coincide con la fase MEM dell'istruzione precedente "LW" e, fintanto che quest'ultima non è terminata, il dato non è ancora stato letto dalla memoria e quindi non è ancora presente nel DLX. Si viene quindi a creare una situazione tale per cui la "ADD" ha bisogno di un dato quando ancora questo non è fisicamente presente sul microprocessore e quindi non è possibile attivare nessun tipo di forwarding all'interno del DLX che eviti di inserire uno stallo della pipeline.

LW Rx, alfa(Ry)	IF	ID	EX	MEM	WB
ADD Rz, Rk, Rx		IF	ID	EX	MEM

1.2) In questo caso il registro in comune alle due istruzioni è Rz e la seconda istruzione "SUB" ne ha bisogno "al più tardi" in EX, leggendolo dal Pipeline Register ID/EX per eseguire la sottrazione Rj-Rz e scriverla nel PR EX/MEM. Contemporaneamente viene eseguito MEM dell'istruzione "ADD" dove in pratica il dato viene solo passato al PR MEM/WB. In questo caso è possibile evitare lo stallo di "SUB" in EX poiché in quel momento il dato è presente sul DLX. Faremo in modo che "SUB" possa ricevere Rz dal PR EX/MEM portando un ramo di forward nei mux in ingresso all'ALU e programmando la Forwarding Unit in modo che commuti il mux sul forwarding solo quando necessario. Nel nostro caso si noti che, se la FU non commutasse uno dei due ingressi dell'ALU sul ramo di forwarding, questa riceverebbe normalmente il dato dal PR ID/EX che contiene il valore di Rz sbagliato perché fetchato da RF quando non ancora aggiornato dal WB dell'istruzione "ADD".

ADD Rz, Rk, Rx	IF	ID	EX	MEM	WB
SUB Ri, Rj, Rz		IF	ID	EX	MEM

Si veda più avanti lo schema con il circuito di forwarding.

2.1) Vedere schema per il circuito del ready e la connessione delle periferiche e delle memorie al bus di sistema.

2.2) Espressioni di selezione semplificate memorie:

$$CSE0 = BA19 \cdot !IO/M\#$$

$$CSR0 = !BA19 \cdot !BA16 \cdot !IO/M\#$$

$$CSR1 = !BA19 \cdot BA16 \cdot !IO/M\#$$

periferiche di I/O:

$$CS8259 = BA15 \cdot IO/M\#$$

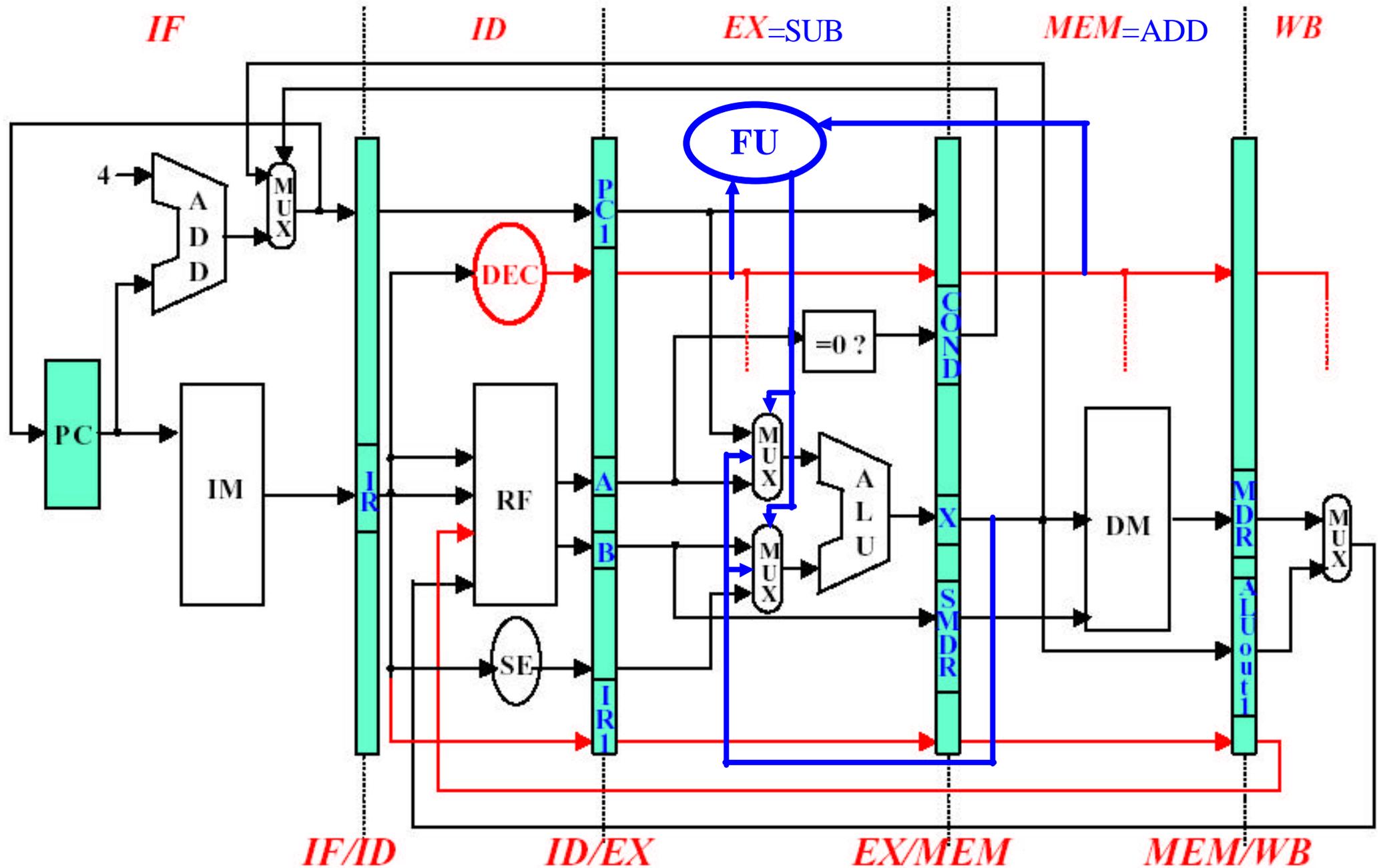
$$CS8255 = !BA15 \cdot !BA14 \cdot IO/M\#$$

$$CSBIT = !BA15 \cdot BA14 \cdot IO/M\#$$

2.3) Per riconoscere il carattere 0Fh bisogna predisporre un campionamento del bus dati durante la lettura della porta A dell'8255. Per triggerare il campionamento sul clock di "FFBIT" devono essere attivi contemporaneamente il chip select dell'8255, il segnale di lettura da I/O e l'indirizzo deve essere quello della porta A (BADR0=0, BADR1=0). Il più "stretto" tra questi segnali è "IORD#": quando esso si disattiva e va alto, il flip-flop campiona. Tale flip-flop camperà "1" se la rete costituita dai 3 AND collegata al bus dati avrà rivelato la configurazione "00001111b" corrispondente a "0Fh". Grazie all'OR posto sull'ingresso "D" del flip-flop e collegato all'uscita "Q", l'1 sarà mantenuto anche quando il flip-flop campionasse mentre sul bus dati è presente un carattere diverso da "0Fh". Il driver in uscita al flip-flop abilita la connessione al bus dati per la lettura del bit, mentre sull'ingresso "CL#" insistono i due segnali di reset previsti.

3.1) I registri si chiamano in genere "Base Pointer" e "Stack Pointer". Essi servono rispettivamente a definire l'inizio dell'area di memoria occupata dallo dello stack e la "cima" dello stack a cui accedere per aggiungere (Push) o prelevare (Pop) dati. Il "Base Pointer" contiene un valore costante fintanto che lo stack è valido, mentre lo "Stack Pointer" viene aggiornato ad ogni accesso allo stack.

3.2) Il segnale "dati" in ingresso ad una interfaccia seriale RS232 è alto quando non transitano dati. Esso è campionato ad una frequenza elevata (ad es. 64 volte più alta del reciproco del tempo di bit). Non appena viene rilevato un livello 0 (start bit) il campionamento successivo avviene dopo Tbit/2, in modo da posizionarsi a metà della durata di ogni bit. I campionamenti successivi avverranno poi con frequenza pari al reciproco del Tbit fino alla fine della trama. E' necessario che trasmettente e ricevente abbiano la stessa frequenza di clock (a meno di un errore di tolleranza) in modo che sia garantito l'allineamento durante la singola trama, che comunque sarà lunga non più di 8 data bit + start bit + parity bit + stop bit.



Schema compito Calcolatori Elettronici del 14/03/'03 - Domande 2.1 e 2.3

