

## Calcolatori Elettronici L A

### Prova scritta del 15/12/'03

#### Esercizio N. 1 (12 punti)

Si vuole estendere il set di istruzioni del DLX *sequenziale* aggiungendo le istruzioni PUSH e POP per la gestione a “stack” di un’area memoria. Non esistendo nel DLX risorse hardware specifiche per la gestione dello stack, oltre a spostare dati tra i registri e lo stack, le istruzioni devono consentire di specificare il registro generale scelto come puntatore alla “cima” (stack pointer) e di aggiornarlo automaticamente sotto le ipotesi: 1) che, al termine dell’istruzione di accesso allo stack, esso punti alla prima cella libera dello stack. 2) che lo stack si riempia per indirizzi di memoria crescenti.

In particolare **PUSH [Ry], Rx** deve scrivere Rx all’indirizzo di memoria puntato da Ry e aggiornare Ry, mentre **POP Rx, [Ry]** deve aggiornare Ry e scrivere su Rx il contenuto della memoria puntata da Ry.

1.1) Si mostri, ad esempio, come potrebbe essere codificata l’istruzione “PUSH [R27], R8” utilizzando il formato che si ritiene più opportuno tra quelli delle istruzioni del DLX. (2 punti)

1.2) Attenendosi alla codifica proposta nella risposta 1.1), con riferimento al datapath *sequenziale* del DLX visto a lezione, si disegnino i diagrammi degli stati il più possibile ottimizzati che controllano l’esecuzione delle due nuove istruzioni. (8 punti)

1.3) Si calcoli il CPI delle due nuove istruzioni, considerato che ogni accesso in memoria necessita in totale di 2 Tck (1 + 1Tw). (2 punti)

#### Esercizio N. 2 (12 punti)

Un sistema a microprocessore basato su 8088 a 5 MHz dispone di 16KB di Eprom (1 chip visibile dall’indirizzo FFFFh a scendere) e 256KB di Ram (1 chip a partire da 00000h). Il sistema trasmette dati su una linea seriale RS232 attraverso un’interfaccia 8251 gestita ad interrupt. Progettando un’opportuna rete logica, fare in modo che tutti e soli i dati in trasmissione (non le parole di comando e nemmeno quelle di stato) siano inviati all’8251 con l’inversione dei pesi binari (cioè “BD0” sia il bit di maggior peso “D7” dell’8251, e così via fino a “BD7” che deve corrispondere a “D0” dell’8251). L’operazione di inversione dei pesi deve risultare trasparente al  $\mu P$ , ossia esso non deve eseguire alcuna routine né istruzione per compierla.

2.1) Nell’ipotesi che nessun dispositivo richieda periodi di wait, disegnare lo schema di progetto con la rete per l’inversione dei pesi, esplicitando le espressioni dei suoi segnali di attivazione. (7 punti)

2.2) Scrivere le espressioni di selezione semplificate delle memorie e dei dispositivi di I/O. (2 punti)

2.3) Supponendo che l’8251 sia stato programmato con parità pari e 2 stop-bit, tenendo conto dell’inversione dei pesi, disegnare la forma d’onda in funzione del tempo della trama generata dall’8251 sull’uscita seriale Tx, in conseguenza all’istruzione “OUT BASE8251, 3Fh”. (3 punti)

#### Esercizio N. 3 (9 punti)

3.1) Descrivere in breve il principio di Ahmdal e la sua utilità nel campo della progettazione dei microprocessori. (3 punti)

3.2) Spiegare cosa misura il parametro “throughput” di un processore con pipeline. (2 punti)

3.3) Dire su quali elementi della pipeline agisce la HDU e in quale modo li condiziona per introdurre uno stallo. (4 punti)

### SCRIVERE IN MODO LEGGIBILE CONSEGNARE ANCHE IL TESTO

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 19/12/03, ore 9:00 Via Rasi**

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

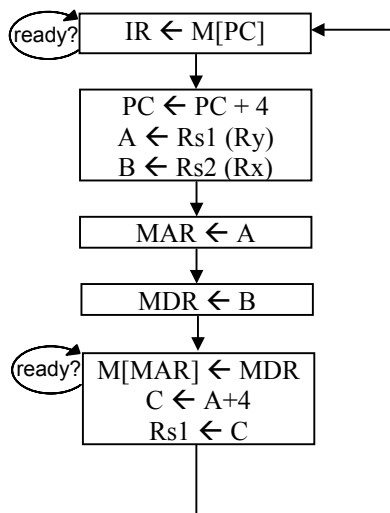
<http://www.ing2.unibo.it> → Didattica → Lauree Triennali (selez. il proprio C.d.L.) → piano degli studi → Calcolatori Elettronici L A

## SOLUZIONE del compito di Calcolatori Elettronici L A del 15/12/03

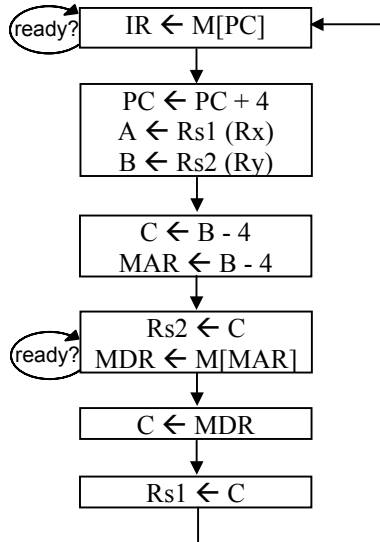
1.1) (2 punti) L'unico formato che non si adatta alla codifica di queste istruzioni è il formato "J"; quindi si possono utilizzare indifferentemente i formati:

	OpCode (6 bit)	Rs1=Ry (5bit)	Rs2=Rx (5bit)	Immediato (16bit)	
I	PUSH	27	8	inutilizzati	
	OpCode (6 bit)	Rs1=Ry (5bit)	Rs2=Rx (5bit)	Rd (5bit)	estensione al codice (11bit)
R	PUSH	27	8	inutilizzati	inutilizzati

1.2) (8 punti) **PUSH [Ry], Rx**



**POP Rx, [Ry]**



1.3) (2 punti) L'istruzione PUSH ha CPI=7 mentre POP ha CPI=8.

2.1) (7 punti) Vedere schema nella pagina seguente.

2.2) (2 punti) Espressioni di selezione semplificate

–memorie:

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

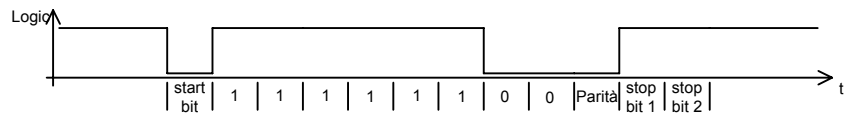
$$CSR0 = !BA19 \cdot !IO/M\#$$

–periferiche di I/O:

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

$$CS8251 = !BA15 \cdot IO/M\#$$

2.3) (3 punti) Trama seriale RS232:

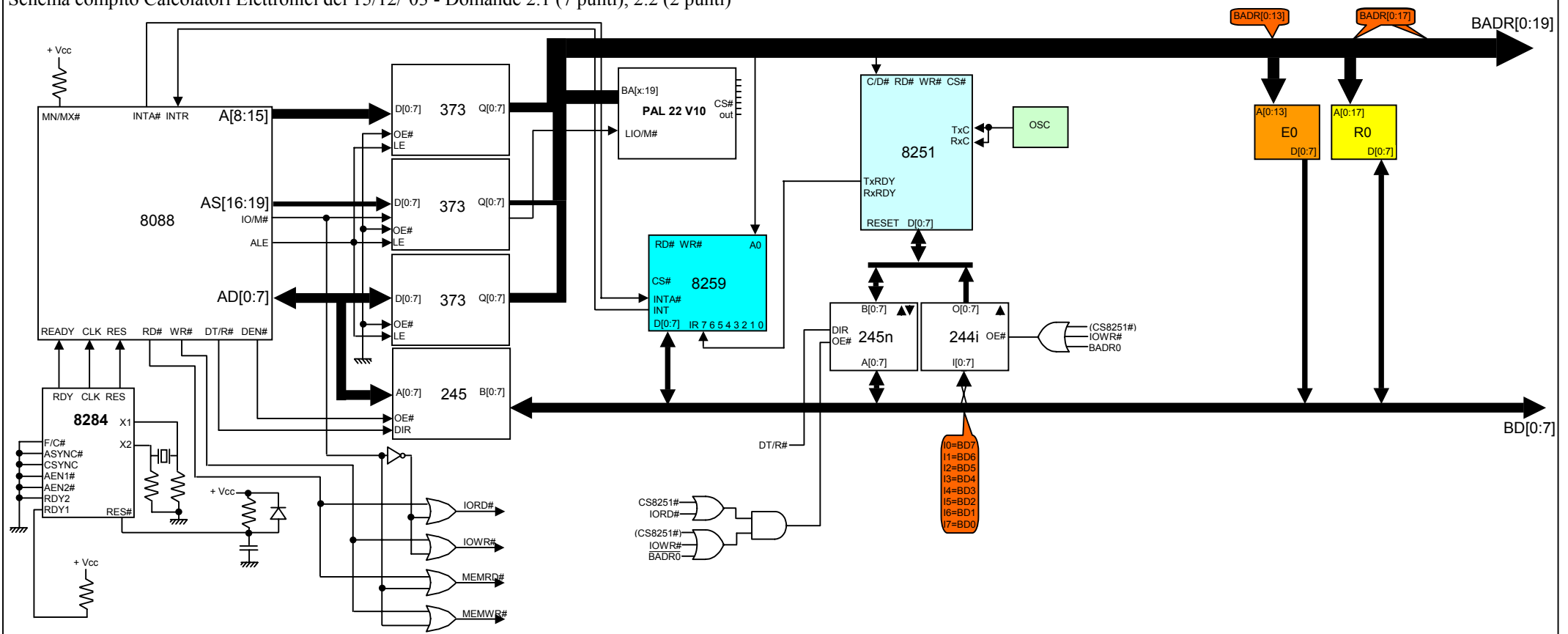


3.1) (3 punti) La legge di Ahmdal sostiene che il miglioramento delle prestazioni globali di un sistema è tanto maggiore quanto più si riesce a rendere rapido quel tipo di sottoprocesso che si esegue con maggior frequenza. Dati sperimentali confermano che –mediamente– in un programma software, il solo 10% del codice occupa la CPU per il 90% del tempo necessario ad eseguire tutto il programma. Il principio di Ahmdal suggerisce il miglior criterio per ottimizzare la CPU, affinché questo avvenga in modo mirato sulle parti hardware impegnate a svolgere quel 10% del codice.

3.2) (2 punti) Il Throughput esprime la frequenza con cui la pipeline conclude l'esecuzione delle istruzioni e corrisponde quindi al "numero di istruzioni completate al secondo". Si definisce genericamente  $\text{Throughput} = 1/T_p$  (dove  $T_p$  è il periodo di clock del  $\mu P$ ). Per i processori con pipeline vale  $\text{Throughput}(p) = N_{\text{stadi}}/T_A$  ( $T_A$  corrisponde al tempo di permanenza delle istruzioni in pipeline, cioè la loro latenza)

3.3) (4 punti) Per introdurre uno stallo la Hazard Detection Unit inibisce il WE# del PC impedendogli l'incremento a PC+4, inibisce il WE# del pipeline register IF/ID mantenendo i vecchi valori di PC e della istruzione in IR e agisce in ID sulle linee di controllo, disattivandole (stato NOP, NO-Operation) affinché nessuna operazione sia eseguita negli stadi a valle EX, MEM e WB.

Schema compito Calcolatori Elettronici del 15/12/'03 - Domande 2.1 (7 punti), 2.2 (2 punti)



## Calcolatori Elettronici L A

### Prova scritta del 15/12/'03

#### Esercizio N. 1 (12 punti)

Si vuole estendere il set di istruzioni del DLX *sequenziale* aggiungendo le istruzioni PUSH e POP per la gestione a “stack” di un’area memoria. Non esistendo nel DLX risorse hardware specifiche per la gestione dello stack, oltre a spostare dati tra i registri e lo stack, le istruzioni devono consentire di specificare il registro generale scelto come puntatore alla “cima” (stack pointer) e di aggiornarlo automaticamente sotto le ipotesi: 1) che, al termine dell’istruzione di accesso allo stack, esso punti alla prima cella libera dello stack. 2) che lo stack si riempia per indirizzi di memoria decrescenti.

In particolare **PUSH [Ry], Rx** deve scrivere Rx all’indirizzo di memoria puntato da Ry e aggiornare Ry, mentre **POP Rx, [Ry]** deve aggiornare Ry e scrivere su Rx il contenuto della memoria puntata da Ry.

1.1) Si mostri, ad esempio, come potrebbe essere codificata l’istruzione “PUSH [R27], R8” utilizzando il formato che si ritiene più opportuno tra quelli delle istruzioni del DLX. (2 punti)

1.2) Attenendosi alla codifica proposta nella risposta 1.1), con riferimento al datapath *sequenziale* del DLX visto a lezione, si disegnino i diagrammi degli stati il più possibile ottimizzati che controllano l’esecuzione delle due nuove istruzioni. (8 punti)

1.3) Si calcoli il CPI delle due nuove istruzioni, considerato che ogni accesso in memoria necessita in totale di 2 Tck (1 + 1Tw). (2 punti)

#### Esercizio N. 2 (12 punti)

Un sistema a microprocessore basato su 8088 a 5 MHz dispone di 16KB di Eprom (1 chip visibile dall’indirizzo FFFFh a scendere) e 256KB di Ram (1 chip a partire da 00000h). Il sistema trasmette dati su una linea seriale RS232 attraverso un’interfaccia 8251 gestita ad interrupt. Progettando un’opportuna rete logica, fare in modo che tutti e soli i dati in trasmissione (non le parole di comando e nemmeno quelle di stato) siano inviati all’8251 con l’inversione dei pesi binari (cioè “BD0” sia il bit di maggior peso “D7” dell’8251, e così via fino a “BD7” che deve corrispondere a “D0” dell’8251). L’operazione di inversione dei pesi deve risultare trasparente al  $\mu P$ , ossia esso non deve eseguire alcuna routine né istruzione per compierla.

2.1) Nell’ipotesi che nessun dispositivo richieda periodi di wait, disegnare lo schema di progetto con la rete per l’inversione dei pesi, esplicitando le espressioni dei suoi segnali di attivazione. (7 punti)

2.2) Scrivere le espressioni di selezione semplificate delle memorie e dei dispositivi di I/O. (2 punti)

2.3) Supponendo che l’8251 sia stato programmato con parità dispari e 2 stop-bit, tenendo conto dell’inversione dei pesi, disegnare la forma d’onda in funzione del tempo della trama generata dall’8251 sull’uscita seriale Tx, in conseguenza all’istruzione “OUT BASE8251, 3Fh”. (3 punti)

#### Esercizio N. 3 (9 punti)

3.1) Descrivere in breve il principio di Ahmdal e la sua utilità nel campo della progettazione dei microprocessori. (3 punti)

3.2) Spiegare cosa misura il parametro “throughput” di un processore con pipeline. (2 punti)

3.3) Dire su quali elementi della pipeline agisce la HDU e in quale modo li condiziona per introdurre uno stallo. (4 punti)

### SCRIVERE IN MODO LEGGIBILE CONSEGNARE ANCHE IL TESTO

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 19/12/03, ore 9:00 Via Rasi**

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

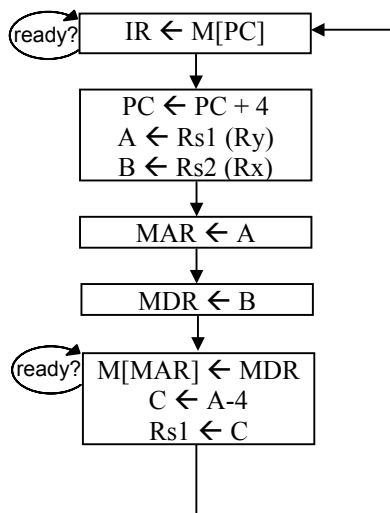
<http://www.ing2.unibo.it> → Didattica → Lauree Triennali (selez. il proprio C.d.L.) → piano degli studi → Calcolatori Elettronici L A

## SOLUZIONE del compito di Calcolatori Elettronici L A del 15/12/03

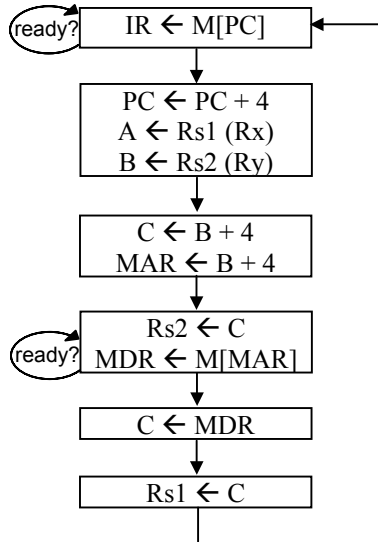
1.1) (2 punti) L'unico formato che non si adatta alla codifica di queste istruzioni è il formato "J"; quindi si possono utilizzare indifferentemente i formati:

	OpCode (6 bit)	Rs1=Ry (5bit)	Rs2=Rx (5bit)	Immediato (16bit)	
I	PUSH	27	8	inutilizzati	
	OpCode (6 bit)	Rs1=Ry (5bit)	Rs2=Rx (5bit)	Rd (5bit)	estensione al codice (11bit)
R	PUSH	27	8	inutilizzati	inutilizzati

1.2) (8 punti) **PUSH [Ry], Rx**



**POP Rx, [Ry]**



1.3) (2 punti) L'istruzione PUSH ha CPI=7 mentre POP ha CPI=8.

2.1) (7 punti) Vedere schema nella pagina seguente.

2.2) (2 punti) Espressioni di selezione semplificate

–memorie:

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

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

–periferiche di I/O:

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

$$CS8251 = !BA15 \cdot IO/M\#$$

2.3) (3 punti) Trama seriale RS232:



3.1) (3 punti) La legge di Ahmdal sostiene che il miglioramento delle prestazioni globali di un sistema è tanto maggiore quanto più si riesce a rendere rapido quel tipo di sottoprocesso che si esegue con maggior frequenza. Dati sperimentali confermano che –mediamente– in un programma software, il solo 10% del codice occupa la CPU per il 90% del tempo necessario ad eseguire tutto il programma. Il principio di Ahmdal suggerisce il miglior criterio per ottimizzare la CPU, affinché questo avvenga in modo mirato sulle parti hardware impegnate a svolgere quel 10% del codice.

3.2) (2 punti) Il Throughput esprime la frequenza con cui la pipeline conclude l'esecuzione delle istruzioni e corrisponde quindi al "numero di istruzioni completate al secondo". Si definisce genericamente  $\text{Throughput} = 1/T_p$  (dove  $T_p$  è il periodo di clock del  $\mu P$ ). Per i processori con pipeline vale  $\text{Throughput}(p) = N_{\text{stadi}}/T_A$  ( $T_A$  corrisponde al tempo di permanenza delle istruzioni in pipeline, cioè la loro latenza)

3.3) (4 punti) Per introdurre uno stallo la Hazard Detection Unit inibisce il WE# del PC impedendogli l'incremento a PC+4, inibisce il WE# del pipeline register IF/ID mantenendo i vecchi valori di PC e della istruzione in IR e agisce in ID sulle linee di controllo, disattivandole (stato NOP, NO-Operation) affinché nessuna operazione sia eseguita negli stadi a valle EX, MEM e WB.

Schema compito Calcolatori Elettronici del 15/12/'03 - Domande 2.1 (7 punti), 2.2 (2 punti)

