

**CORSO DI ARCHITETTURA DEGLI ELABORATORI
PROGRAMMAZIONE ASSEMBLY: PROGETTO**

1 GIUGNO 2009

STUDENTI ANNO IN-CORSO – Consegna: Lunedì 29 Giugno 2009
STUDENTI ANNO PRECEDENTE – Consegna: Lunedì 13 Luglio 2009

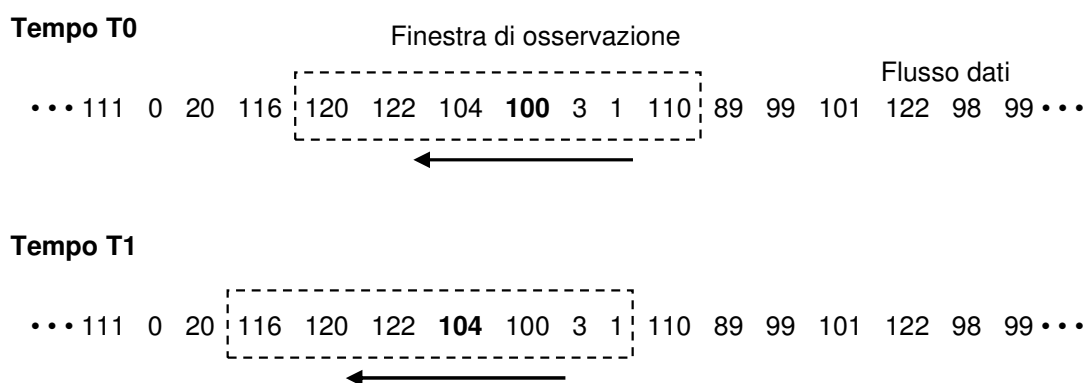
Lo studente programmi nel linguaggio Assembly 8088 un micro 8088 che realizza il sistema sotto descritto. Si usi la versione di assembly e l'assemblatore allegato al libro "Andrew S. Tanenbaum. *Architettura dei Calcolatori. Un Approccio Strutturato*. Pearson Prentice Hall, 5 ed., 2006.", e scaricabile dall'indirizzo internet <ftp://ftp.cs.vu.nl/pub/evert>

Descrizione del sistema

Un micro 8088, 1MB di memoria e un'interfaccia di I/O sono usati per realizzare sistema di elaborazione di segnali video digitali. Il segnale in ingresso, scomposto in fotogrammi ("frame"), è rappresentato da una sequenza finita di pixel la cui descrizione è un numero (in notazione decimale).

Il sistema filtra il segnale in ingresso pesando ogni pixel con quelli circostanti, contenuti all'interno di una finestra di osservazione. La finestra di osservazione è spostata per scandire l'intero fotogramma.

La Figura sottostante rappresenta il sistema in due successivi istanti di esecuzione:



Nell'esempio di sopra la finestra di osservazione ha dimensione sette. Contiene quindi sette numeri (pixel). Al tempo T0 il numero 100, collocato al centro della finestra di osservazione, sarà pesato usando i tre numeri alla sua destra (3, 1 e 110) e quelli alla sua sinistra (104, 122 e 120).

Al successivo tempo T1 sarà invece il numero 104 a trovarsi al centro della finestra, e quindi pesato usando i numeri adiacenti a destra (100, 3 e 1) e a sinistra (122, 120 e 116).

Le operazioni di filtraggio svolte sulla finestra sono tre:

- 1) calcolo del massimo – l’elaborazione restituisce il valore massimo contenuto nella finestra;
- 2) calcolo del minimo – l’elaborazione restituisce il valore minimo contenuto nella finestra;
- 3) calcolo del mediano – l’operazione di media all’interno della finestra è eseguita riordinando i numeri in maniera crescente e considerando il numero al centro della finestra come valore medio, o meglio mediano. La Figura sottostante illustra il calcolo del mediano:

	Finestra di osservazione						
Valori letti	120	122	104	100	3	1	110
Finestra ordinata	122	120	110	104	100	3	1
Valore mediano							104

Si noti che il file in ingresso (la descrizione numerica di un fotogramma) ha dimensione finita, ad esempio 480.000 pixel. Pertanto, a regime la finestra di osservazione avrà dimensione fissa (negli esempi di sopra la dimensione è 7). Invece all’inizio della scansione del fotogramma ci sarà una fase di creazione della finestra. In particolare scandendo il primo pixel la finestra avrà dimensione 1, scandendo il secondo pixel avrà dimensione 2, al terzo pixel dimensione 3, fino a raggiungere la dimensione pre-fissata (7, negli esempi di sopra).

Stesso dicasi per la scansione dell’ultima parte del fotogramma quando la finestra si ridurrà fino ad avere dimensione 1 scandendo l’ultimo pixel.

Emulazioni

Due file di testo emulano le periferiche di IO.

input.txt contiene la descrizione numerica del segnale digitale in ingresso.

output.txt contiene la descrizione numerica del segnale digitale in uscita (i valori mediani calcolati).

Le chiamate di sistema per manipolare file sono mascherate dalle funzioni di libreria messe a disposizione dall’assemblatore (`_READ`, `_WRITE`, `_OPEN`, ...).

NOTA: quanto non specificato in questo testo fa parte delle scelte del progettista, quindi dello studente!!! Si riportino le scelte, commentandole, nella relazione.

SI SVOLGANO I SEGUENTI PUNTI

Assumendo che:

- un pixel è 16bit
- la finestra di osservazione contiene 8 elementi (numerati da 0 a 7 con il valore mediano in posizione 4) – Si indichi con DIM la dimensione della finestra di osservazione.
- la dimensione di un fotogramma è 800x600 pixel – Si indichi con DIM_FOTOGRAMMA la dimensione del fotogramma.

Punto 1

Si usi lo stack per realizzare la finestra di osservazione “*buffer*” riservandone un’apposita area e creando un puntatore alla locazione di partenza..

(Suggerimento, si potrebbe usare BP per individuare l’area dello stack destinata al buffer)

Si crei il vettore “*fotogramma*” collocato nel segmento dati e dimensionato per contenere la descrizione numerica di un fotogramma.

Punto 2

Si implementi le procedure

stato CARICA (fotogramma*)

e

stato SALVA (fotogramma*)

che:

1- ricevono in ingresso il puntatore al vettore *fotogramma*

2- rispettivamente CARICA copia il contenuto di *input.txt* inizializzando il vettore *fotogramma* mentre SALVA copia il contenuto del vettore *fotogramma* nel file *output.txt* (il file *output.txt* deve contenere un solo fotogramma, quindi il vecchio contenuto del file verrà cancellato ad ogni nuova invocazione della procedura)

3- restituiscono lo stato di esecuzione (0 in caso di successo, 1 in caso di errore – ad esempio errori nella apertura, chiusura, lettura del file)

(Si usi AX per ritornare lo stato di esecuzione e lo stack per il passaggio del puntatore al vettore fotogramma)

Punto 3.A - Solo per studenti ANNO IN-CORSO

Si implementi le funzioni

int MINIMO (valore1, valore2,valoreDIM, DIM)

e

int MASSIMO (valore1, valore2,valoreDIM, DIM)

che ricevono in ingresso DIM valori numerici e restituiscono rispettivamente il valore più piccolo e quello più grande.

(Si usi AX per ritornare il risultato e lo stack per il passaggio dei valori in ingresso; DIM è passato in ingresso)

Punto 3.B - Solo per studenti ANNO PRECEDENTE

Si implementi la funzione

mediano MEDIANO (valore1, valore2,valoreDIM, DIM)

che riceve in ingresso DIM valori numerici e dopo averli ordinati restituisce il valore mediano.
(Si usi AX per ritornare il valore mediano e lo stack per il passaggio dei valori in ingresso; DIM è passato in ingresso)

Punto 4

Si implementi la procedura

```
void STAMPA (valore1, valore2, ....valoreDIM, DIM)
```

che stampa a video i DIM valori numerici passati in ingresso.

(Si usi lo stack per il passaggio dei valori in ingresso; DIM è passato in ingresso)

Punto 5

Fatti i punti 1, 2, 3 e 4, si scriva un programma che facendo uso delle funzioni e procedure sviluppate nei quesiti precedenti implementi un sistema per elaborazione di segnali video digitali che legge il fotogramma contenuto in *input.txt* e, dopo averlo copiato in *fotogramma*, ne calcola i valori min/max o mediani da salvare in *output.txt*

Si usi un'interfaccia di dialogo per chiedere all'utente il tipo di elaborazione da eseguire: minimo, massimo o mediano. (Gli studenti ANNO IN-CORSO calcoleranno minimo e massimo, mentre gli studenti ANNO PRECEDENTE calcoleranno il mediano)

Si assuma che il file in ingresso è scandito a regime, si ignori quindi la creazione e distruzione della finestra di osservazione. La finestra avrà quindi sempre dimensione DIM e la scansione del file *input.txt* partirà dal valore di posizione DIM/2.

(Il primo pixel elaborato sarà quindi quello in posizione DIM/2 e si troverà centrato nella finestra di osservazione. All'inizio la finestra di osservazione conterrà i primi DIM pixel del file *input.txt*)

Punto 6

Fatti i punti 4 e 5, si faccia uso della procedura sviluppata al punto 4 per stampare il contenuto della finestra di osservazione a video prima e dopo l'invocazione di MINIMO/MASSIMO/MEDIANO().
(Si noti che dopo la prima invocazione di STAMPA i DIM valori saranno già sullo stack, così come prima della seconda invocazione di STAMPA)

Punto 7.A - Solo per studenti ANNO IN-CORSO

Si svolga il Punto 3.B

Punto 7.B - Solo per studenti ANNO PRECEDENTE

Fatto il Punto 5 se ne modifichi il codice aggiungendo la creazione e distruzione della finestra di osservazione. Ci saranno pertanto due transitori, uno iniziale in cui la finestra crescerà dalla dimensione 1 a DIM, e uno finale in cui decrescerà da DIM a 1.

I punti 1-2-3-4 sono necessari per valutare positivamente l'elaborato. I punti 5-6-7 concorrono al raggiungimento del massimo dei voti.

NOTE SULLA VALUTAZIONE DEL COMPITO

La valutazione del progetto dipende strettamente dai seguenti vincoli:

- Il progetto deve essere consegnato entro le 24.00 della data indicata nella prima pagina.
- Saranno attribuiti dei punti ai seguenti aspetti generali:
 - 1) **Presenza di commenti.** Il codice assembly deve essere commentato in modo da agevolare la comprensione. Ogni funzione deve essere preceduta da opportuni commenti che ne chiariscono la funzionalità e gli argomenti. **I progetti privi di commenti NON saranno valutati.**
 - 2) **Codice funzionante.** Il progetto deve essere assemblabile e funzionante. Indicare nella relazione quali sono i Punti di questo testo svolti e quale sistema operativo si è usato.
 - 3) **Codice ottimizzato.** Il codice disordinato è difficile da verificare. Saranno valutate positivamente scelte progettuali che ottimizzino il codice. L'ottimizzazione deve riguardare anche i commenti: chiari e concisi.
 - 4) **Codice strutturato.** Il codice deve essere opportunamente strutturato facendo uso di chiamate a funzioni.
- **Relazione**, di max. 4 pagine per facilitare la lettura del codice. **Riportare nella relazione le scelte progettuali.**
- **Lavoro di gruppo.** Il progetto deve essere svolto da gruppi di lavoro, composti da almeno due persone e al massimo tre, in maniera collaborativa da tutti i componenti del gruppo. **Ogni componente del gruppo deve conoscere le scelte progettuali e i dettagli implementativi.**
- **Confronto tra gruppi.** E' utile il confronto tra gruppi. Tuttavia, dovendo ogni gruppo fare le sue scelte progettuali, è fondamentale evitare scelte di massa. Saranno valutati negativamente progetti simili tra loro o con simili scelte progettuali.

MODALITA' DI CONSEGNA

Si legga quanto riportato nella pagina: <http://iafelice.web.cs.unibo.it/archi/>