

Progetto Assembly 8088

SVD88

di Ferrari Vincenzo e Iadarola Barbara

Il progetto di Assembly di Giugno 2009 è stato sviluppato, scritto e testato su macchine **Intel x86** con sistema **Ubuntu Linux** e sulle seguenti macchine del **Laboratorio Ercolani** :

- **jake.cs.unibo.it** (remoto)
- **nelson.cs.unibo.it** (remoto)
- **melot.cs.unibo.it** (remoto)
- **dancairo.cs.unibo.it** (fisico)

In questo file è possibile trovare la descrizione delle principali funzioni del Sistema di Elaborazione di Segnali Video Digitali SVD88.

All'avvio viene mostrata la seguente schermata:

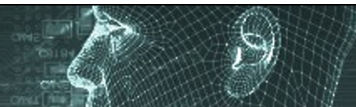
```
File  Modifica  Visualizza  Terminale  Aiuto
*****
* Benvenuto nel sistema di elaborazione di segnali video digitali SVD88 *
*****

*****
* Menu Principale *
*****
* 1. Carica      *
* 2. Filtra      *
* 3. Salva       *
* 4. Automatizza *
* 5. Esci        *
*****

Inserire l'opzione da eseguire: █
```

E' possibile scegliere una delle cinque opzioni presenti nel Menu, altrimenti sarà visualizzato un messaggio di errore con conseguente uscita dal programma.

1. **Carica** : permette di caricare il contenuto del file *input.txt* nella variabile *fotogramma*. Il file viene aperto, letto e inserito all'interno di *fotogramma*: nel caso occorra un errore, il sistema avvisa l'utente con un messaggio e termina l'esecuzione.
2. **Filtra** : calcola, allo scorrimento della finestra di osservazione, i valori mediani. Carica i pixel sullo *stack* partendo dall'ultimo elemento presente nella variabile *fotogramma*, stampando i valori a video prima e dopo il calcolo del mediano; esso avviene ordinando in modo crescente i pixel, spostando alla base dello *stack* quello minore. Il mediano è recuperato scandendo lo *stack* con il *BP* più un indice, in modo da puntare al $DIM/2$ valore nel caso la dimensione della finestra di osservazione sia dispari, o al $DIM/2+1$ valore nel caso essa sia pari. Alla fine della funzione viene aggiornata la variabile *fotogramma* con



- l'inserimento dei valori mediani appena calcolati.
3. **Salva** : restituisce il contenuto della variabile *fotogramma*, all'interno del file *output.txt*. Ogni volta viene creato un nuovo file e sovrascritto a quello precedente: in caso di errore nella creazione, nell'apertura, nella scrittura e nella chiusura del file, l'utente viene avvisato da un messaggio e il programma viene terminato.
 4. **Automatizza** : il sistema esegue in maniera sequenziale i precedenti punti, terminando infine l'esecuzione dello stesso. Dato che la finestra d'osservazione viene aggiornata all'interno del file principale *svd88.s*, viene usata la variabile *varSM* per uscire dalla procedura **Filtra** e ritornare alla funzione **Automatizza**.
 5. **Esci** : termina l'esecuzione del programma.

Condizioni

Per un corretto funzionamento è necessario seguire i seguenti punti :

- non è possibile filtrare il file *input.txt* se prima non è stato caricato tramite la funzione **CARICA**, eseguibile scegliendo la prima opzione del Menu Principale
- non sono ammessi numeri negativi all'interno del file *input.txt*, dato che la finestra dei pixel è stata intesa come un piano cartesiano le cui coordinate possono essere solamente (+x, +y)
- a meno che non lo si carichi nuovamente, non è possibile filtrare più di una volta il file *input.txt*
- i pixel presenti nel file *input.txt* non devono superare la dimensione di 32767 poiché ogni singolo registro non è in grado di rappresentare un valore positivo maggiore di tale limite e infatti verrebbero salvati dei valori negativi all'interno del file *output.txt*

File

Di seguito sono descritti i vari file contenenti il codice sorgente del progetto :

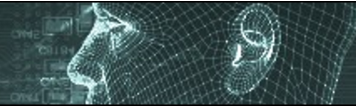
- **svd88.s** : è il file principale al cui interno vengono richiamati tutti gli altri tramite il comando `#include "nomefile.s"`
- **statoCARICA.s** : contiene il codice della funzione **CARICA**
- **statoSALVA.s** : contiene il codice della funzione **SALVA**
- **fzMediano.s** : contiene il codice della funzione **MEDIANO**
- **bufferSTAMPA.s** : contiene il codice della funzione **STAMPA**
- **fotogrammaUpdate.s** : il codice presente nel file permette di aggiornare la variabile *fotogramma* alla fine dell'esecuzione dell'opzione **Filtra** presente nel Menu Principale
- **errori.s** : il file contiene la gestione degli errori che si possono incontrare durante l'esecuzione del programma
- **Menu.s** : tramite questo file viene stampato il Menu Principale

Variabili

In questa sezione sono spiegate le variabili principali del programma :

- **fotogramma** : contiene il contenuto del file *input.txt*, inizialmente, e poi i mediani calcolati
- **DIM** : contiene la dimensione della finestra d'osservazione
- **mediani** : variabile di supporto dove vengono salvati temporaneamente i mediani
- **inputfd** : contiene il file descriptor di *input.txt*
- **outputfd** : contiene il file descriptor di *output.txt*
- **varSM** : è usata per il controllo della quarta opzione del Menu Principale, **Automatizza**





- **DIM_FOTOGRAMMA** : rappresenta la dimensione della variabile *fotogramma*. La grandezza è stata calcolata tenendo conto che all'interno del file *input.txt* è possibile avere al massimo 80*60 righe e per ognuna di esse al massimo 6 cifre (5 per il pixel e una per il ritorno di linea \n): **80*60*6**

Algoritmi

Sono stati sviluppati principalmente due algoritmi all'interno del progetto: l'**algoritmo di ordinamento** della finestra d'osservazione e l'**algoritmo di salvataggio** dei pixel sullo stack.

- **Ordinamento** : per ordinare la finestra d'osservazione vengono utilizzati due cicli annidati del seguente tipo:

```
for (int i = 1; i < (n - 1); i++) {
    for (int j = i + 1; j < n; j++) {
    }
}
```

All'interno del secondo ciclo è presente un controllo :

```
if (v[i] < v[j]) {
    aux = v[i];
    v[i] = v[j];
    v[j] = aux;
}
```

per cui, se l'elemento puntato da *(BP)(SI)* è minore dell'elemento successivo (ovvero quello puntato da *(BP)(DI)*) avviene lo scambio dei due elementi e si incrementano i suddetti indici per scandire l'intera finestra di osservazione.

- **Salvataggio** : per leggere un numero a più cifre è necessario seguire il seguente algoritmo :

$$1.234 = (4 + (3 * 10) + (2 * 100) + (1 * 1000))$$

Il valore viene scandito dalla cifra meno significativa, mentre i restanti vengono moltiplicati per un contatore che verrà moltiplicato per dieci all'interno di un ciclo.

