

**CORSO DI ARCHITETTURA DEGLI ELABORATORI A-K M-Z
PROGRAMMAZIONE ASSEMBLY: PROGETTO**

20 DICEMBRE 2007

CONSEGNA Giovedì 7 febbraio 2008

Lo studente programmi nel linguaggio Assembly 8088 presentato a lezione un micro 8088 usato per controllare un sistema automatico di ispezione ottica.

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

Il sistema verifica le parti meccaniche prodotte da una catena automatica, mentre scorrono su un rullo di trasporto. La verifica avviene tramite una telecamera e un software per l'elaborazione delle immagini. Se l'oggetto analizzato non corrisponde nelle dimensioni ad uno riportato nel database esso viene scartato azionando un relé. Il sistema è schematizzato in Figura 1. Il blocco 8088 incorpora anche la logica di controllo delle varie periferiche. I segnali di controllo sono indicati in tratteggio.

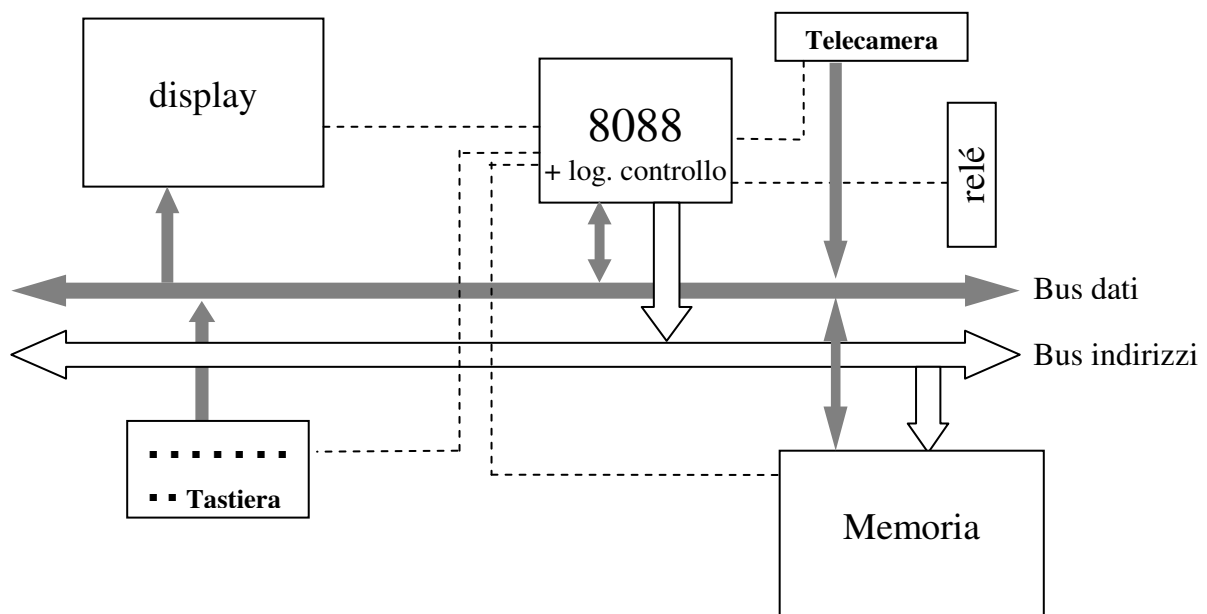


Figura 1: Descrizione del sistema.

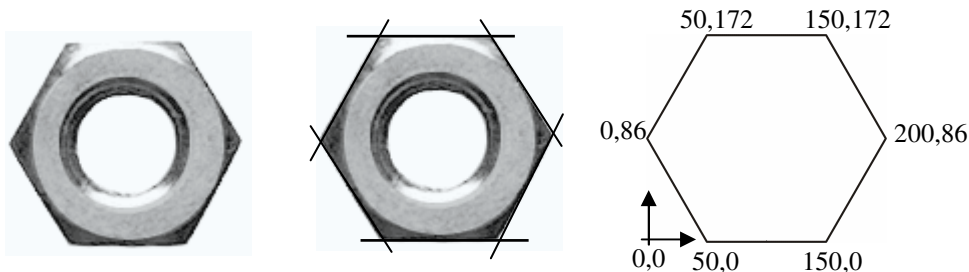
Il sistema è composto da una consolle di dialogo con l'operatore (tastiera e display), una memoria principale, una periferica di input (telecamera) e una di output (relé).

Compito dello studente è esclusivamente la programmazione in Assembly del micro processore.

Funzionamento della telecamera

La telecamera incorpora un software per il trattamento delle immagini. In particolare la telecamera restituisce in uscita la descrizione dell'oggetto osservato come insieme delle coordinate dei suoi punti estremi.

La figura 2 chiarisce il funzionamento della telecamera: cattura immagine ed elaborazione.



OUTPUT della telecamera: 50 0 0 86 50 172 150 172 200 86 150 0

Figura 2: Funzionamento della telecamera.

L'oggetto osservato è quindi tradotto in un'insieme di punti (i suoi vertici) descritti con le coordinate cartesiane nella forma X,Y . L'output della telecamera è quindi la descrizione numerica dell'oggetto come insieme delle coordinate X,Y partendo dal punto più vicino all'origine $0,0$ procedendo in senso orario.

La finestra di osservazione della telecamera è dal punto $0,0$ al punto $255\ 255$.

Funzionamento del sistema

Il sistema funziona nel seguente modo:

- 1) Attende che la telecamera rilevi un oggetto;
- 2) Riceve dalla telecamera la descrizione numerica dell'oggetto;
- 3) Visualizza sul display l'oggetto;
- 4) Cerca nel database l'oggetto;
- 5) Decide se scartare l'oggetto;

Emulazioni

Non avendo a disposizione un sistema reale, in particolare le periferiche di IO, si introducono le seguenti emulazioni:

- 1) Tastiera e display sono emulati dalla consolle dell'elaboratore usato.
- 2) L'evento asincrono di lettura di un'immagine da parte della telecamera è emulato con la pressione del tasto associato alla lettera x .
- 3) La descrizione numerica dell'immagine letta è riportata nel file *immagine.txt*. Un esempio di tale file è la riga: `0 0 0 42 143 40 140 1`
- 4) L'azionamento del meccanismo con cui scartare i pezzi difettosi tramite relé è emulato dalla scrittura sul file *scartati.txt* della codifica numerica dell'immagine dell'oggetto scartato.

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

Punto 1

Si disegni il diagramma degli stati del sistema. Il diagramma deve indicare anche gli stati d'errore.

Punto 2

Si implementi la procedura

output IN_TELECAMERA (void)

che resta in attesa di leggere un'immagine e restituisce tramite lo stack la descrizione numerica dell'immagine letta. *(Si ricorda che l'evento asincrono dell'arrivo di un oggetto è emulato dalla pressione del tasto x e che la lettura di un'immagine è emulata attraverso il file di testo immagine.txt).*

In caso di errore ad aprire o chiudere il file la procedura restituisce (sempre tramite lo stack) il valore -1. In caso di lettura di lettera diversa da x la procedura rimane in attesa.

Si implementi la procedura

stato OUT_SCARTA (oggetto_da_scartare)

che invocata emula l'azione di scartare scrivendo sul file di testo *scartati.txt* la codifica numerica dell'oggetto scartato. Il file è aperto prima della scrittura e chiuso dopo. La funzione restituisce il valore 0 in caso di successo oppure -1 in caso di errore ad aprire o chiudere il file *scartati.txt*.

Punto 3

Si implementi la funzione

intero DISTANZA (punto1, punto2)

che date le coordinate cartesiane di due punti calcola la loro distanza esprimendola con un numero intero. Il valore calcolato viene restituito.

(Si noti che al Punto 4 questa funzione è usata per decidere se scartare o meno un oggetto. Tale decisione si basa su una disuguaglianza. Si valuti quindi la possibilità di introdurre approssimazioni, nel calcolo della distanza, che ne semplifichino l'implementazione algoritmica. Si motivino le eventuali scelte fatte)

Punto 4

Si scriva un programma assembly 8088 che realizza il sistema di ispezione ottica.

In particolare si implementi un blocco di INIZIALIZZAZIONE che aperto il file *data_base.txt* ne carica l'intero contenuto in memoria, e chiude il file.

Il file *data_base.txt* è il seguente:

```
50 0 0 86 50 172 150 172 200 86 150 0
0 0 0 120 120 120 120 0
0 0 0 40 140 40 140 0
0 0 50 144 100 0
```

e rappresenta le forme degli oggetti che fanno da riferimento per la telecamera.

(Il sistema in pratica confronta l'immagine acquisita dalla telecamera con tali oggetti per capire se il pezzo prodotto è esatto o se va scartato poiché difettoso)

Eseguita l'inizializzazione il programma esegue ciclicamente il seguente algoritmo:

- 1) invoca la procedura IN_TELECAMERA
- 2) confronta l'immagine letta col database caricato in fase di inizializzazione
- 3) decide se scartare l'oggetto e in caso di scarto invoca la procedura OUT_SCARTA

Il programma gestisce gli stati di errore ritornati dalle funzioni OUT_SCARTA e IN_TELECAMERA producendo adeguati messaggi di errore, e gli stati di successo con altrettanti messaggi a video. Il display varia quindi a seconda dello stato del sistema.

-nota: non avendo a disposizione un controllo a matrice del display bisogna riscrivere il display ogni volta che cambia di stato. Non essendo possibile svuotare lo schermo è opportuno inserire molte righe bianche (ad esempio 100) prima di riscrivere lo schermo. -

Il sistema decide se scartare un oggetto verificando che la DISTANZA tra ogni suo punto e il corrispondente dell'immagine del database non sia superiore a 8. Si veda la figura 3 a titolo di esempio. Sono cerchiati i punti in cui l'immagine letta differisce da quella a database. A esempio si dovrà calcolare la distanza tra (0,86) e (8,87). Stesso dicasi per tutti gli altri vertici della figura.

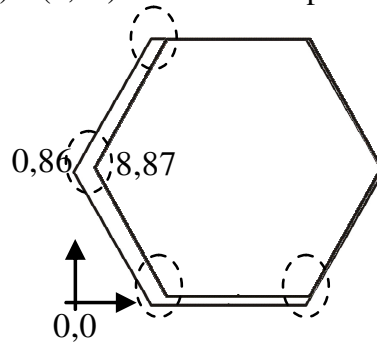


Figura 3: Parametro di scarto.

(Si scarta quindi un oggetto se non soddisfa le due condizioni 1) avere una forma riportata nel database e 2) non discostarsi molto da tale forma)

Punto 5

Fatto il punto 4, si soddisfi la seguente specifica. Il sistema, dopo aver ricevuto un'immagine dalla telecamera la visualizza indicando i vertici dell'immagine col carattere *. L'area di visualizzazione è 32x32 caratteri.

(Si ricorda che un'immagine può al massimo esser grande 256x256 caratteri si deve quindi considerare un fattore di scala per visualizzarla)

Punto 6

Fatto il punto 4, si soddisfi la seguente specifica. Il sistema, dopo aver riconosciuto un'immagine come valida (avendo cioè deciso di non scartare il relativo oggetto) ne calcola l'area e il perimetro. I valori calcolati sono restituiti a video.

Si risolva il seguente punto sotto l'ipotesi semplificativa che l'oggetto rilevato dalla telecamera ha una forma geometrica regolare. *(Quanto detto implica che l'oggetto avrà una delle forme riportate a database ma potrà avere dimensioni diverse)*

Punto 7

Fatti tutti i precedenti punti si suggerisca una possibile implementazione per gestire il caso in cui gli oggetti abbiano forma accettabile (che soddisfa quindi le due condizioni di scarto) ma che si trovino traslati o ruotati. Si risolva questo punto in forma descrittiva, discutendo possibili scelte implementative e riportando solo degli accenni di codice.

Questo punto, se fatto, deve essere parte della relazione.

Si commentino gli eventuali limiti dall'assembly a disposizione.

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

NOTE PER LA CONSEGNA

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.

Modalità di consegna

- Preparare una cartella chiamata “**cognome1_cognome2_cognome3**” (dove **cognome1**, **cognome2**, ... sono i cognomi dei componenti il gruppo di lavoro; evitare spazi nei nomi) contenente le seguenti cartelle/file:
 1. cartella “**progetto**”: contiene i file SORGENTI del progetto
 2. file “**relazione.pdf**”: relazione PDF
 3. file “**sorgente.pdf**”: file PDF contenente i SORGENTI del progetto
 4. file .pdf col testo del progetto (questo file!)
- Comprimere la cartella “**cognome1_cognome2_cognome3**” in “**cognome1_cognome2_cognome3.zip**”
- Spedire il file .zip allegato ad un mail ai seguenti indirizzi (ad entrambi e con un solo mail) entro il termine di consegna:
 1. **biafelice@arces.unibo.it**
 2. **difelice@cs.unibo.it**
- Indicare come **soggetto** del mail: “[**ARCHI**]cognome1_cognome2_cognome3”
- Riportare eventuali commenti nella **relazione**, non nel testo del mail.

Problemi riscontrati durante il precedente appello per la consegna

- Mail inviati separatamente ai due indirizzi: genera confusione rendendo difficile capire chi ha consegnato.
- Mail (forse) spedite ma non arrivate. Alcuni account di posta possono avere malfunzionamenti o limiti nella dimensione della posta in uscita. Si consiglia di mettere in copia alla mail di consegna l'indirizzo di un componente del gruppo per verificarne l'invio.
- Mail spedite senza l'allegato.

Errori di questo tipo possono rendere il progetto non valido vanificando il lavoro fatto.