

Compito di Architettura dei Calcolatori - A.A. 2006-07
Prova di esame del 30 gennaio 2007

COGNOME:

NOME:

MATRICOLA:

Istruzioni: Spiegare con precisione e chiarezza TUTTE le assunzioni che vengono effettuate per risolvere eventuali punti che si ritengono ambigui o non specificati. Tempo assegnato per lo svolgimento: 90 minuti.

1) [10 punti] Disegnare e discutere gli schemi dell'Unità di Controllo di una CPU in funzione della struttura della Control Word (1 o 2 campi next-CW, o struttura variabile).

SVOLGIMENTO:

Si veda il capitolo 15 del libro di testo e dei lucidi presentati a lezione, in particolare i lucidi: 15, 16, 17, 18.

2) [10 punti] Sia data una sequenza di valori positivi, tutti diversi fra loro, rappresentata tramite un vettore di byte già fornito. Scrivere un programma assembly in grado di trovare il secondo valore più piccolo della sequenza ed il suo indice nel vettore (denotando con 1 l'indice del primo elemento della sequenza). Ad esempio, sulla sequenza (4,5,3,2,1) il programma deve trovare come valore 2 e come indice 4, perché il secondo valore più piccolo è 2 e si trova al quarto posto. Al termine dell'esecuzione del programma, il valore trovato deve essere nel registro CL e l'indice trovato deve essere nel registro DL.

SVOLGIMENTO:

```
ORG 100h                ; standard, per poter compilare ed eseguire un file COM
jmp start               ; salta al codice

;
; I dati del programma
;

vettore db 10, 9, 3, 4, 8, 7, 1, 2, 11
                ; La sequenza di valori positivi da analizzare.
                ; La soluzione di questa istanza è CL=2, DL=8

vettore_lungh dw 9      ; La lunghezza del vettore

sec_min db -1           ; Rappresenta il secondo valore più piccolo
                ; della sequenza in input. '-1' indica che la
                ; variabile non è stata inizializzata.

prim_min db -1          ; Rappresenta il valore più piccolo della
                ; sequenza in input. '-1' indica che la
                ; variabile non è stata inizializzata.

sec_indice db 0          ; Indice del secondo valore più piccolo.
prim_indice db 0         ; Indice del primo valore più piccolo.

;
; Il codice del programma
;

; Il problema può essere risolto cercando prima il minimo,
; e successivamente effettuando la ricerca del minimo

start:
    mov cx, vettore_lungh    ; Leggo il vettore in tutta la sua lunghezza
prima_analisi:
    mov bx, cx               ; Sfrutto il contatore per indirizzare gli elementi del vettore
    mov al, vettore[bx-1]    ; Accedo agli elementi a partire dall'ultimo
    cmp prim_min, -1         ; Controllo che la variabile non sia stata inizializzata
    jne min_inizializzato_1
    mov prim_min, al          ; Se non è inizializzata, allora la inizializzo
                                ; con il primo elemento.
    mov prim_indice, bl       ; Inizializzo anche l'indice
    jmp end_loop_1           ; Passo al valore successivo nel vettore
min_inizializzato_1:
    cmp al, prim_min          ; Se la variabile è stata già inizializzata
    jnb end_loop_1           ; allora verifico se è più grande del valore
    mov prim_min, al          ; minimo finora trovato.
    mov prim_indice, bl
end_loop_1:
    loop prima_analisi

; Ora cerco il secondo valore più piccolo
```

```

    mov cx, vettore_lungh
seconda_analisi:
    mov bx, cx
    mov al, vettore[bx-1]
    cmp prim_min, al
    je end_loop_2

    cmp sec_min, -1
    jne min_inizializzato_2
    mov sec_min, al

    mov sec_indice, bl
    jmp end_loop_2
min_inizializzato_2:
    cmp al, sec_min
    jnb end_loop_2
    mov sec_min, al
    mov sec_indice, bl
end_loop_2:
    loop seconda_analisi

; Inserisco i risultati in CL e DL, come richiesto dal testo dell'esame

    cmp sec_min, -1
    jne sec_min_inizializzato
    mov cl, prim_min
    mov dl, prim_indice
    jmp end
sec_min_inizializzato:
    mov cl, sec_min
    mov dl, sec_indice
end:
    ret

```

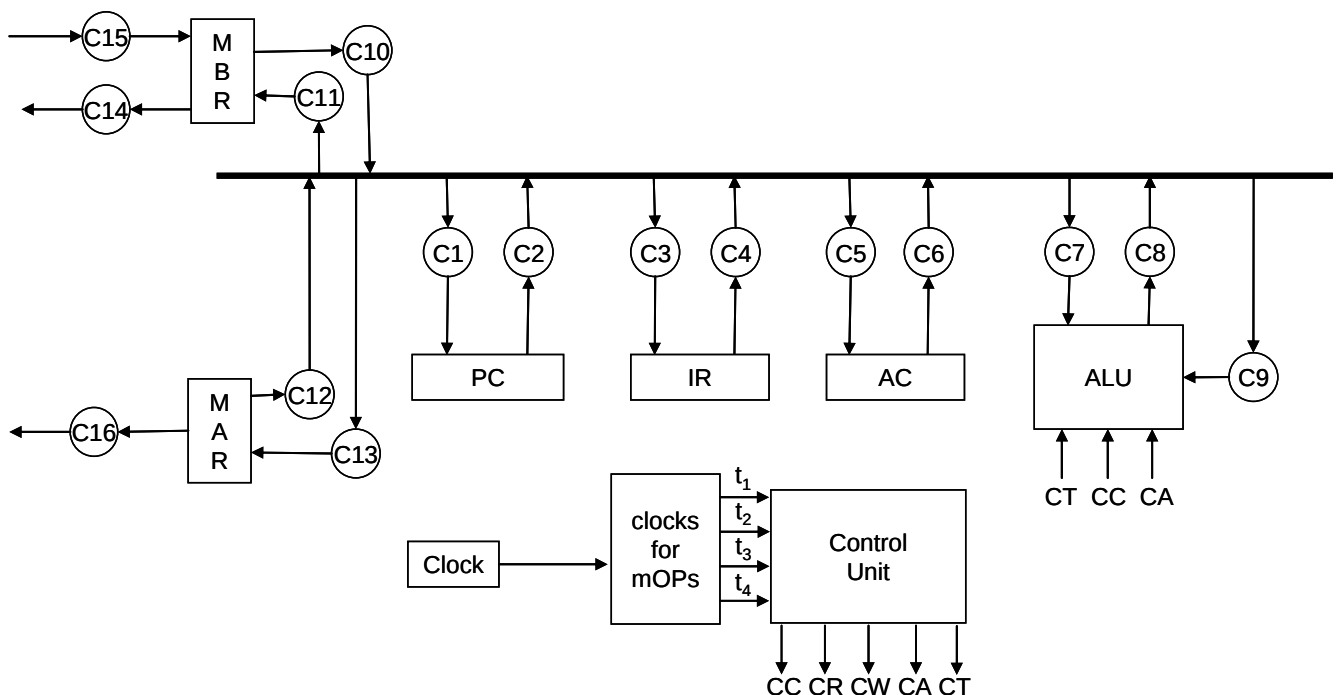
; Sfrutto il contatore per indirizzare gli elementi del vettore
 ; Accedo agli elementi a partire dall'ultimo
 ; Se AL contiene il valore più piccolo, allora lo ignoro

 ; controllo che la variabile non sia stata inizializzata
 ; Se non è inizializzata, allora la inizializzo
 ; con il primo elemento diverso da prim_min.
 ; Inizializzo anche l'indice.
 ; Passo al valore successivo nel vettore.

 ; Se la variabile è stata già inizializzata
 ; allora verifico se è più grande del valore
 ; minimo finora trovato.

3) [10 punti] Dato lo schema della semplicissima CPU (VS0) sotto disegnato nella versione a singolo bus descrivere le micro-operazioni necessarie per l'effettuazione della fase di prelevamento della prossima istruzione (fetch),

- indicando quali sono i segnali di controllo da attivare ad ogni micro-operazione
- spiegando la funzione di ogni segnale di controllo da attivare
- giustificando perché le operazioni elementari sono raggruppate nelle micro-operazioni descritte



SVOLGIMENTO:

Per il punto a. si veda l'Appendice B dei lucidi presentati a lezione, in particolare i lucidi: 39, 40, 41, 42, 43. I punti b. e c. sono conseguenti.