

Compito di Architettura dei Calcolatori - A.A. 2006-07
 Prova di esame del 26 febbraio 2008

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] Elencare tutte le principali modalità conosciute per l'indirizzamento degli operandi nelle istruzioni e per ognuna di esse disegnare il diagramma per il calcolo dell'indirizzo effettivo dell'operando.

SVOLGIMENTO:

Si veda il cap.10 del libro di testo e dei lucidi presentati a lezione, in particolare i lucidi alle pagine 3, 4, 6, 8, 11, 14, 16.

2) [10 punti] Scrivere un programma assembly per calcolare, dati due numeri A e B, i primi N termini della successione di Fibonacci. A, B, e N devono essere variabili del programma. P.es., se A=0, B=1 e N=6 il programma calcola 0 1 1 2 3 5. Ogni numero della successione va messo in una cella di memoria distinta. La serie ottenuta va memorizzata in celle di memoria consecutive.

Descrivere chiaramente TUTTE le assunzioni fatte per chiarire aspetti non specificati o che si ritengono ambigui. Specificare se si usa l'assembly di VCPU presentato a lezione mediante l'emulatore ENIAC o si usa un altro linguaggio assembly (in tal caso indicare quale).

SVOLGIMENTO:

Si riporta lo svolgimento con riferimento all'emulatore ENIAC.

```

0   JMP 5           ; salta alla cella d'inizio del programma
; I DATI del programma, memorizzati in celle che sono usate come variabili di ingresso al programma
1   0              ; contiene F0 e poi uno dei due valori precedenti per il calcolo di F(N)
2   1              ; contiene F1 e poi uno dei due valori precedenti per il calcolo di F(N)
3   6              ; contiene N, contatore dei valori da calcolare
4   30             ; contiene l'indirizzo della cella a partire da cui si scrivono in uscita i valori
                    ; calcolati

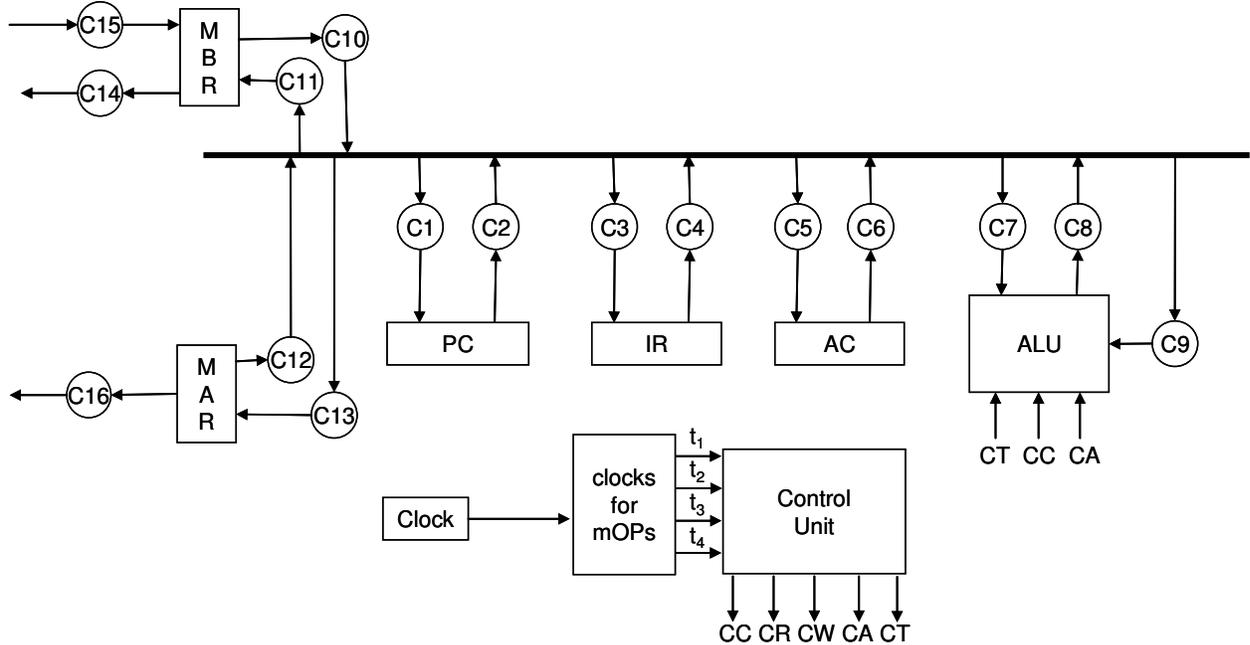
; IL PROGRAMMA
; si inizializza DX che viene usato per indirizzare la cella su cui scrivere il valore corrente
5   LOAD @4
6   STORE DX
; se N vale 0 si esce subito (si assume che N non possa essere negativo)
7   LOAD @3
8   SUB 0          ; si sottrae 0 solo per settare i flag della ALU
9   JZ 34
; si produce F0 in uscita
10  LOAD @1
11  STORE @DX
; si decrementa N e se adesso vale 0 si esce subito
12  DEC @3
13  JZ 34
; si produce F1 in uscita
14  LOAD @2
15  INC DX
16  STORE @DX
; si decrementa N e se adesso vale 0 si esce subito
17  DEC @3
18  JZ 34
; se il programma è arrivato qua ha scritto almeno F0 e F1 (quindi N valeva all'inizio almeno 3)
; questa è la parte generica in cui si calcola F(N) come somma di F(N-1) e F(N-2)
; il calcolo viene per semplicità del programma effettuato con due insiemi di istruzioni
; il primo insieme di istruzioni calcola F(N) usando la cella 1 come F(N-2) e la cella 2 come F(N-1)
; e di conseguenza salva F(N) nella cella 1 perché quello che c'era là non serve più
19  LOAD @1
20  ADD @2
21  STORE @1
; si produce in uscita FN
22  INC DX
23  STORE @DX
; si decrementa N e se adesso vale 0 il calcolo è terminato
24  DEC @3
25  JZ 34
; il secondo insieme di istruzioni calcola F(N) usando la cella 1 come F(N-1) e la cella 2 come F(N-2)
; di conseguenza salva F(N) nella cella 2 perché quello che c'era là non serve più

```

```

26 LOAD @2
27 ADD @1
28 STORE @2
; si produce in uscita FN
29 INC DX
30 STORE @DX
; si decrementa N e se adesso vale 0 il calcolo è terminato
31 DEC @3
32 JZ 34
; altrimenti si ricomincia il ciclo
33 JMP 19
; termine del programma
34 HLT
    
```

3) [10 punti] Dato lo schema della semplicissima CPU (VS0) sotto disegnato nella versione a singolo bus disegnare un nuovo schema (utilizzando sempre una struttura interna della CPU basata su singolo bus) per poter eseguire le istruzioni STORE S D e STORE S (D). Esse memorizzano il valore contenuto in un registro S, rispettivamente, nel registro D o nella cella di memoria il cui indirizzo è contenuto nel registro D. Sia S che D sono uno dei registri di un banco che contiene 8 possibili registri. Descrivere inoltre, con riferimento a tale nuovo schema, il flusso dei dati tra le varie componenti della CPU per l'esecuzione completa di tali due istruzioni e raggruppandolo in micro-operazioni.

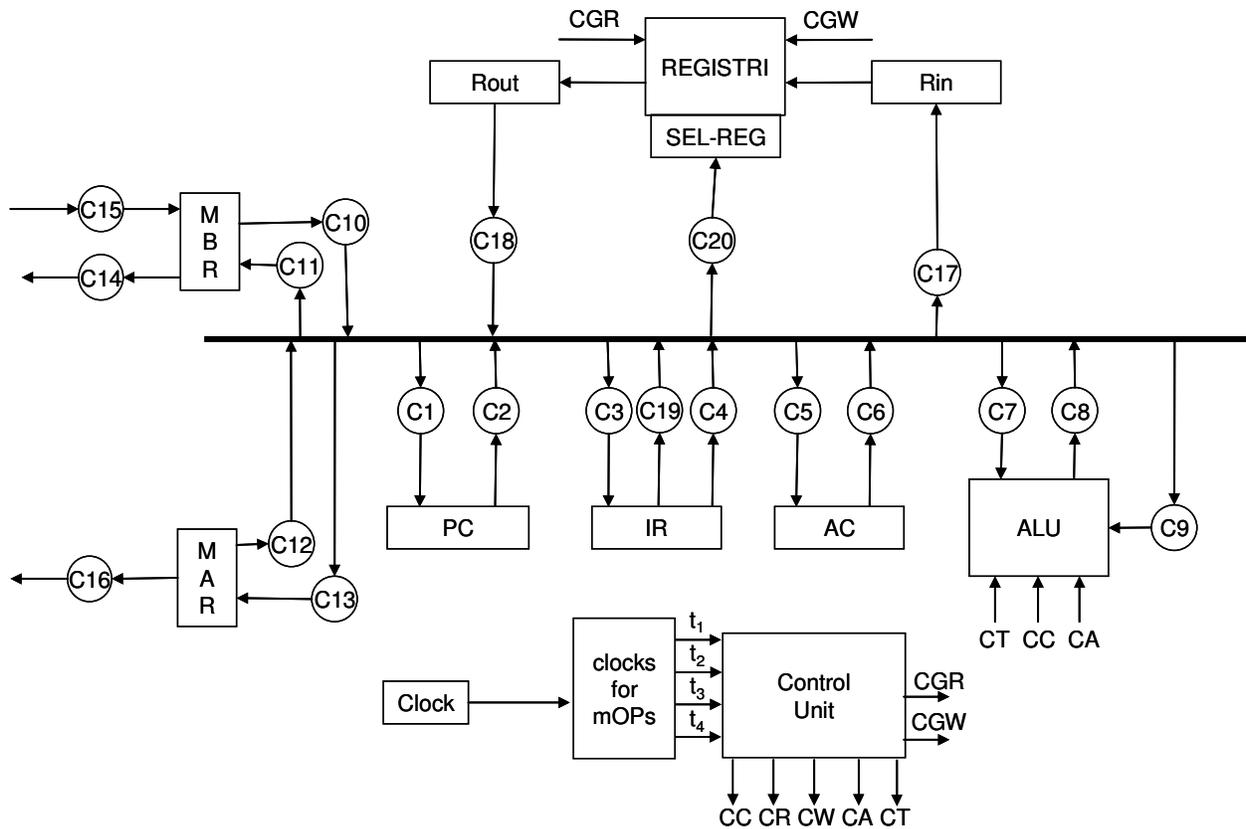


SVOLGIMENTO:

Bisogna ovviamente aggiungere allo schema 8 registri che vengono acceduti come se fossero una memoria. Serve quindi un registro di selezione (SEL-REG nel nuovo schema sotto riportato) cui si fa arrivare il valore del registro desiderato tra gli otto, come se fosse il Memory Address Register per l'accesso alla RAM. Il valore del registro desiderato è un numero a 3 bit che viene preso da IR e viene fatto arrivare a SEL-REG dal bus mediante il nuovo segnale di controllo C20. Servono inoltre due nuovi segnali di controllo per attivare la scrittura (CGW) sul e la lettura (CGR) dal registro che viene selezionato mediante il valore in SEL-REG.

Inoltre, per poter effettuare più velocemente le operazioni da registro a registro, aggiungiamo due buffer, uno per l'ingresso (Rin) ed uno per l'uscita (Rout) dagli otto registri.

Si noti che IR nello schema di VS0 presentato a lezione è un registro a 8 bit di cui 6 (da b_5 a b_0) sono dedicati alla parte indirizzi e 2 (b_7 e b_6) sono dedicati al codice operativo. Tralasciamo le modifiche da effettuare al formato delle istruzioni per aggiungere le due nuove istruzioni specificate. Nel nuovo schema il segnale di controllo C4 mette sul bus il valore dei 3 bit $b_2b_1b_0$ corrispondenti al registro di destinazione per le due nuove istruzioni, mentre il nuovo segnale di controllo C19 mette sul bus il valore dei 3 bit $b_5b_4b_3$ corrispondenti al registro di partenza per le due nuove istruzioni. (Per quanto riguarda la decodifica delle pre-esistenti istruzioni, dovunque vi era il segnale C4 adesso ci dovranno essere entrambi C4 e C19).



Con le modifiche precedentemente descritte, riportate nel nuovo schema qua sopra, le micro-operazioni sono le seguenti (si noti quindi che i registri S e D specificati nelle istruzioni sono due parametri generici che di volta in volta indicano quale degli otto registri viene selezionato)

STORE S D

- t1 SEL-REG ← (IR)_S C19 C20
Rout ← (REGISTRI) CGR
- t2 Rin ← Rout C18 C17
- t3 SEL-REG ← (IR)_D C4 C20
REGISTRI ← Rin CGW

STORE S (D)

- t1 SEL-REG ← (IR)_S C19 C20
Rout ← (REGISTRI) CGR
- t2 MBR ← Rout C18 C11
- t3 SEL-REG ← (IR)_D C4 C20
Rout ← (REGISTRI) CGR
- t4 MAR ← Rout C18 C13
- t5 (memory) ← MBR C16 C14 CW