Compito di Architettura dei Calcolatori - A.A. 2011-12
Prova di esame del 13 febbraio 2013

Istruzioni: Spiegare chiaramente TUTTE le assunzioni che vengono effettuate per chiarire eventuali punti che si ritengono ambigui o non specificati.

1) [10 punti]
Disegnare e discutere gli schemi dell'Unità di Controllo di una CPU nei seguenti tre casi di struttura della Control Word:

- 1 solo campo *next*-CW nella Control Word
- 2 campi *next*-CW nella Control Word
- struttura variabile della Control Word

SVOLGIMENTO:

Si veda il cap.17 del libro di testo e dei lucidi presentati a lezione, in particolare i lucidi dalle pagine 15 a 18 comprese.

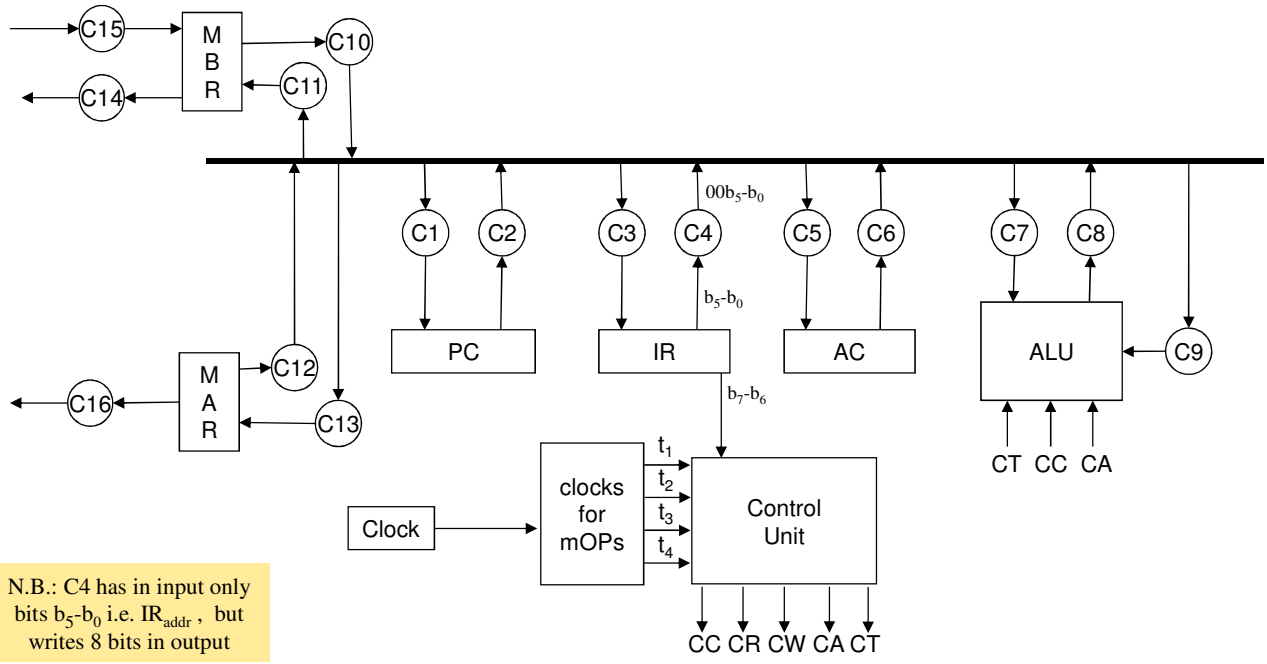
2) [10 punti]
Scrivere un programma nel linguaggio assembly di VCPU presentato a lezione mediante l'emulatore ENIAC per calcolare, dati due valori iniziali A e B, i primi N≥3 termini della successione di Fibonacci con questi valori iniziali. P.es.:, se A=1, B=3 e N=6 il programma calcola 1 3 4 7 11 18. I valori A, B, e N vengono forniti al programma in tre celle di memoria predefinite. Ogni numero della successione va messo in una cella di memoria distinta. La serie ottenuta va memorizzata in celle di memoria consecutive.

SVOLGIMENTO:

```
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                  ; contiene il PRIMO valore iniziale A (cioè F_1)
2                  ; contiene il SECONDO valore iniziale B (cioè F_2)
3                  ; contiene il NUMERO di termini da calcolare della serie valore (N)
4                  ; contiene l'indirizzo della cella a partire dalla quale si scrive il RISULTATO
; IL PROGRAMMA
; inizializzazione con il calcolo del primo valore F_3
5    LOAD @4       , si carica nell'accumulatore l'indice della cella in cui scrivere il risultato
6    STORE CX      ; lo si scrive in CX, usato sia per indirizzare la cella in cui scrivere il risultato
;                    corrente F_k sia per accedere alle celle con i valori F_k-1 e F_k-2
7    LOAD @1       ; si carica A (cioè F_1)nell'accumulatore
8    STORE @CX     ; lo si scrive nella prima cella
9    INC CX        ; si incrementa l'indice della cella in cui scrivere il risultato
10   LOAD @2       ; si carica B (cioè F_2) nell'accumulatore
11   STORE @CX     ; lo si scrive nella seconda cella
12   DEC CX        , si decrementa CX che adesso punta al valore F_k-2, cioè F_1, cioè A
13   LOAD @3       ; si carica nell'accumulatore il numero complessivo di termini da calcolare
14   DEC AX        ; il primo termine è stato calcolato
15   DEC AX        ; il secondo termine è stato calcolato
16   STORE DX      ; si scrive in DX il numero di termini ancora da calcolare. Si potrebbe manipolare
;                    direttamente il valore nella cella 3, ma si preferisce non alterare i dati di ingresso
; qui c'è da calcolare ancora almeno un termine, questa è la parte iterata
;  prima si calcola F_k come somma di F_k-2 e F_k-1 e lo si scrive
17   LOAD @CX      ; si carica nell'accumulatore F_k-2
18   INC CX        ; adesso CX punta alla cella che contiene F_k-1
19   ADD @CX       ; che viene sommato a F_k-2 per dare F_k
20   INC CX        ; adesso CX punta alla cella in cui scrivere F_k
21   STORE @CX     ; viene scritto il valore di F_k
22   DEC CX        ; si decrementa CX che adesso punta al valore F_k-1
;  poi si decrementa il numero dei termini ancora da calcolare e si torna al test per iterare o meno
23   DEC DX        ;
24   JNZ 17        ;
25   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) che permette di eseguire le due nuove istruzioni STORE *S D* e STORE *S* (*D*). La prima memorizza il valore contenuto in un registro *S* nel registro

D. La seconda memorizza il valore contenuto in un registro *S* 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 e spiegare inoltre, con riferimento a tale nuovo schema, i micro-programmi per l'esecuzione di ognuna delle due nuove istruzioni. Tralasciare le modifiche da effettuare al formato delle istruzioni per aggiungere le due nuove istruzioni specificate.



SVOLGIMENTO:

(Si veda il nuovo schema a pagina seguente)

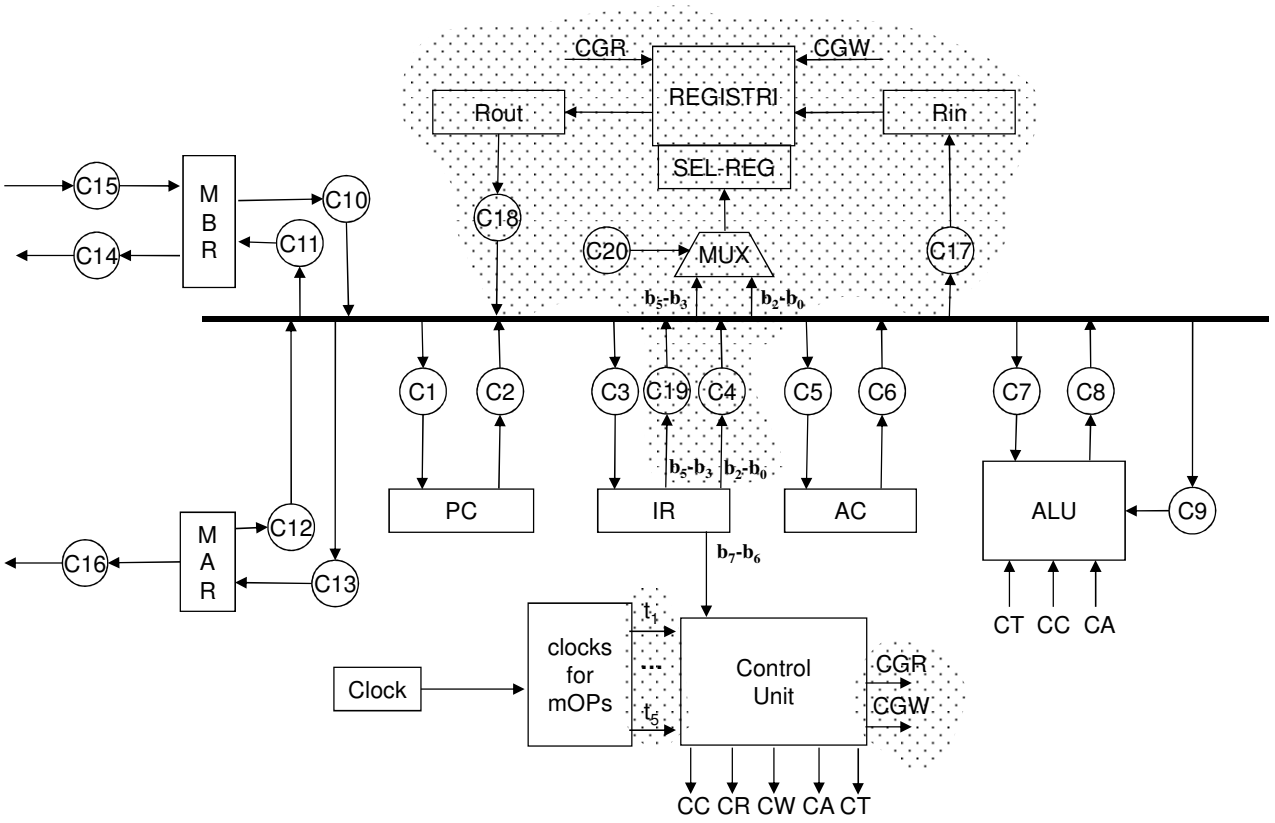
Bisogna ovviamente aggiungere allo schema 8 registri che vengono acceduti come se fossero una memoria. Serve quindi un registro di selezione (SEL-REG nel nuovo schema) cui si fa arrivare il valore del registro desiderato tra gli otto, come se fosse il Memory Address Register per l'accesso alla RAM. Il 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 che comanda un multiplexer avente due ingressi: uno che riceve dal bus i tre bit $b_2b_1b_0$ e l'altro che riceve dal bus i tre bit $b_5b_4b_3$. 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 tre bit $b_2b_1b_0$ (indicati con IR_D) corrispondenti al registro di destinazione per le due nuove istruzioni, mentre il nuovo segnale di controllo C19 mette sul bus il valore dei tre bit $b_5b_4b_3$ (indicati con IR_S) 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). Il multiplexer mette in uscita il primo ingresso (cioè i tre bit $b_2b_1b_0$) quando C20=0, mentre mette in uscita il secondo ingresso (cioè i tre bit $b_5b_4b_3$) quando C20=1.

Si faccia attenzione che l'esecuzione della nuova istruzione STORE *S* (*D*) richiede un tempo di clock in più che è stato aggiunto allo schema (cioè il circuito di generazione dei segnali di clock per la Control Unit adesso genera 5 segnali invece di 4).

Con le modifiche precedentemente descritte, riportate nel nuovo schema qua sotto ed evidenziate con lo sfondo grafico, le micro-operazioni sono le seguenti (si noti 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