Compito di Architettura dei Calcolatori - A.A. 2009-10
Prova di esame del 16 febbraio 2011

COGNOME: NOME: MATRICOLA:

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

1) [10 punti] Descrivere e spiegare, anche mediante l'uso di diagrammi, le tre tecniche per la gestione dell'Input/Output da parte della CPU.

SVOLGIMENTO:

Si veda il capitolo 7 del libro di testo e dei lucidi presentati a lezione.

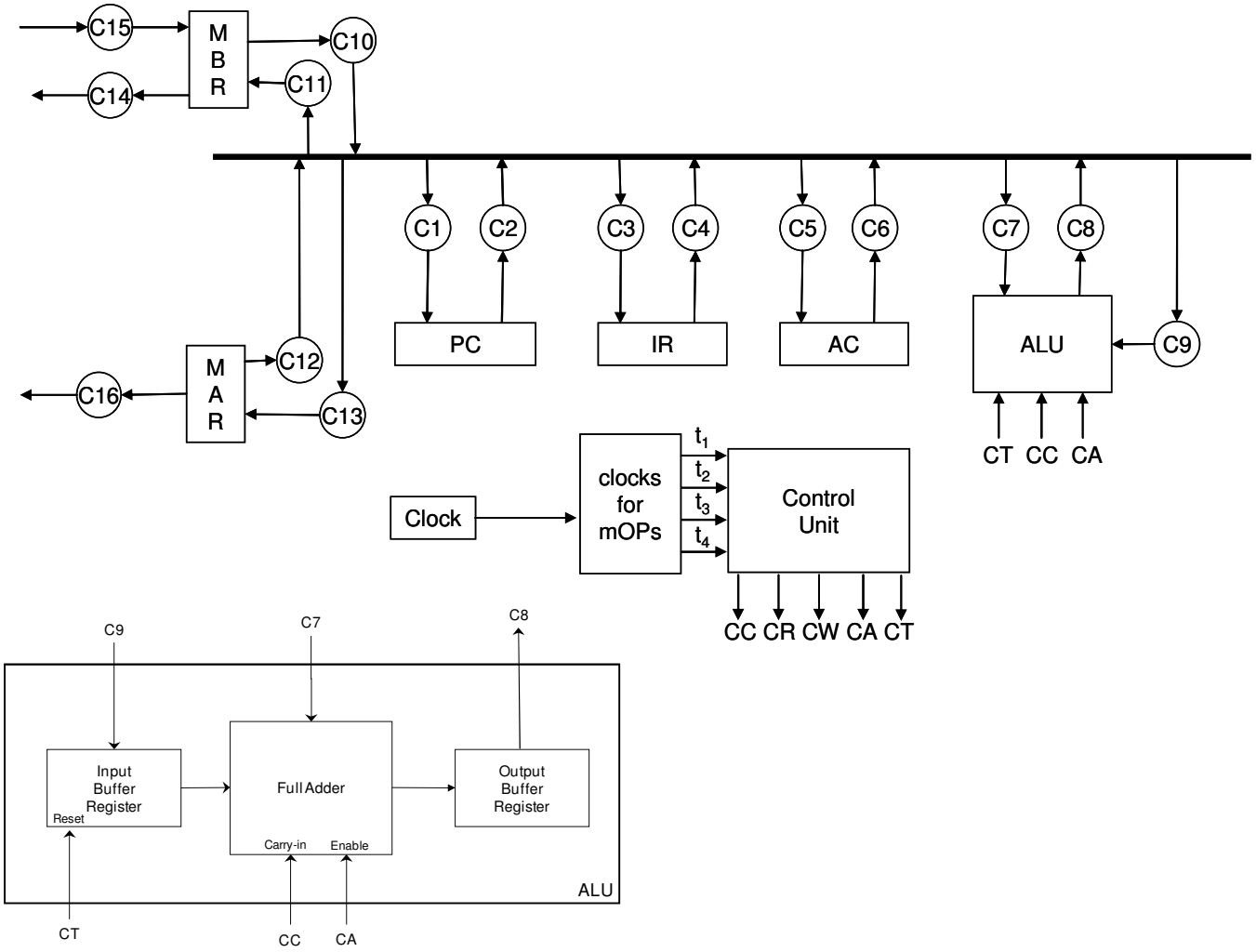
2) [10 punti] Scrivere un programma nel linguaggio assembly di VCPU presentato a lezione mediante l'emulatore ENIAC per calcolare, dati due valori iniziali A e B, i primi N termini della successione di Fibonacci con questi valori iniziali. P.es.:, se A=0, B=2 e N=6 il programma calcola 2 4 6 10 16 26. 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 di termini da calcolare
14   STORE DX      ; si scrive in DX il numero di termini ancora da calcolare. Non è indispensabile
;                    usare DX, si potrebbe manipolare direttamente il valore nella cella 3, ma si
;                    preferisce non alterare i dati di ingresso al programma
15   SUB 0         ; si attivano i flag aritmetici senza alterare il valore dell'accumulatore
16   JZ 25         ; non ci sono più termini da calcolare e si finisce
; 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   JMP 16        ; l'operazione di salto non altera il valore dei flag aritmetici
25   HLT
```

3) [10 punti] Dato lo schema della semplicissima CPU (VS0) sotto disegnato nella versione a singolo bus disegnare il nuovo schema (utilizzando sempre una struttura interna della CPU con singolo bus) per poter eseguire l'istruzione DIS_LOAD R. Essa carica l'accumulatore AC con il valore contenuto nella cella di memoria di indirizzo pari alla somma del valore contenuto nel registro R e del valore contenuto nell'accumulatore AC, dove R specifica uno dei registri di un banco che contiene quattro possibili registri. Si ricordi che la ALU nella versione a singolo bus ha la struttura sotto riportata. Descrivere inoltre, con riferimento a tale nuovo schema, il flusso dei dati

tra le varie componenti della CPU per l'esecuzione completa di tale nuova istruzione e indicarne il raggruppamento in micro-operazioni.

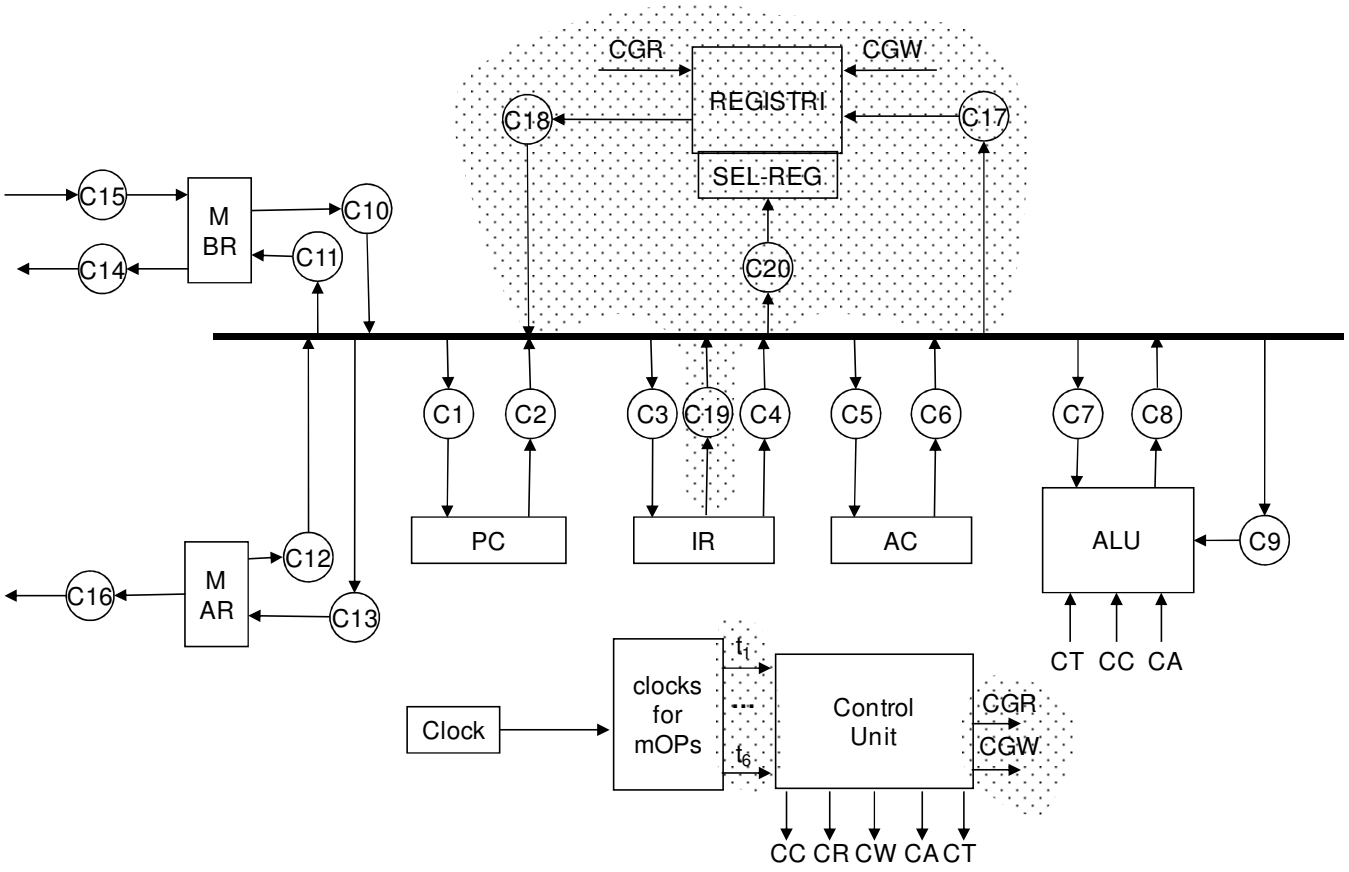


SVOLGIMENTO:

Il nuovo schema della CPU, con evidenziate le parti aggiunte, è disegnato più sotto.

Sono stati aggiunti allo schema 4 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 i quattro, come se fosse il Memory Address Register per l'accesso alla RAM. Il valore del registro desiderato è un numero a 2 bit che viene preso da IR e viene fatto arrivare a SEL-REG dal bus mediante il nuovo segnale di controllo C20. Servono inoltre un nuovo segnale di controllo per attivare la lettura (CGR) dal registro che viene selezionato mediante il valore in SEL-REG.

Si noti che nello schema di VS0 presentato a lezione IR 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 la nuova istruzione specificata. Nel nuovo schema il nuovo segnale di controllo C19 mette sul bus il valore dei 2 bit b_1b_0 di IR corrispondenti al registro usato per la nuova istruzione. (Il segnale C4 rimane inalterato e continua ad essere usato per mettere sul bus il valore dei 6 bit della parte indirizzo di IR).



Con le modifiche precedentemente descritte, riportate nel nuovo schema qua sopra, le micro-operazioni sono le seguenti (si noti quindi che il registro R specificato nelle istruzioni è un parametro generico che di volta in volta indicano quale dei quattro registri viene selezionato).

Si faccia attenzione che l'esecuzione di questa nuova istruzione richiede due tempi di clock in più che sono stati aggiunti allo schema (cioè il circuito di generazione dei segnali di clock per la Control Unit adesso genera 6 segnali invece di 4).

	DIS_LOAD *R*	
t1	SEL-REG ← IR_R	C19 C20
t2	ALU ← REGISTRI	CGR C18 C9
t3	ALU ← AC+ALU	C6 C7 CA
t4	MAR ← ALU	C8 C13
t5	MBR ← memory	C16 CR C15
t6	AC ← MBR	C10 C5

Si noti che nello schema sono stati aggiunti anche i nuovi segnale di controllo C17 e CGW, il primo che mette il bus in comunicazione con il banco dei 4 nuovi registri e il secondo che abilita la scrittura sul banco stesso, segnali che sono entrambi indispensabili per caricare valori nei 4 nuovi registri, anche se non vengono usati per l'esecuzione di questa istruzione.