Compito di Architettura dei Calcolatori - A.A. 2009-10
Prova di esame del 17 settembre 2010

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] 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.11 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 nel linguaggio Assembly di Virtual CPU per effettuare la differenza tra due valori positivi di 48 bit memorizzati ognuno in due celle di memoria consecutive e scrivere il risultato in altre celle di memoria consecutive. Assumere che il primo termine (minuendo) sia sempre maggiore del secondo termine (sottraendo) così da essere certi che la differenza è sempre positiva. Commentare il programma scritto spiegandone la logica e le scelte effettuate sia relativamente a come rappresentare i valori a 48 bit mediante le celle di memoria (che si ricorda sono a 24 bit) sia relativamente all'organizzazione dell'operazione di differenza.

SVOLGIMENTO:

Poiché la ALU di vCPU assume che in ingresso vi siano valori rappresentati da una codifica **complementata** a 24 bit, la corretta gestione dell'operazione ritiene di capire quando sia necessario riportare un prestito dalla sottrazione tra le parti meno significative dei valori verso la sottrazione tra le parti più significative. Per simmetria rispetto al caso dell'addizione tra rappresentazioni su codifiche **non complementate**, nel caso della sottrazione è l'assenza del bit di riporto in uscita dalla ALU (cioè CA=0) che indica la necessità di gestire il prestito (laddove nel caso di addizione era il caso CA=1 che indicava la necessità di gestire il riporto). Fa eccezione il caso in cui il sottraendo sia nullo, nel qual caso CA=0 ma non serve gestire il prestito. Infine, la sottrazione tra le parti più significative, anche tenendo conto del prestito, fornisce comunque un valore non negativo, data la specifica del problema, ed è pertanto correttamente rappresentabile con i 24 bit più significativi.

```
0   JMP 10        ; 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 la parte MENO significativa del PRIMO valore (LSB_1)
2                 ; contiene la parte PIU' significativa del PRIMO valore (MSB_1)
3                 ; contiene la parte MENO significativa del SECONDO valore (LSB_2)
4                 ; contiene la parte PIU' significativa del SECONDO valore (MSB_2)
5                 ; contiene la parte MENO significativa del RISULTATO
6                 ; contiene la parte PIU' significativa del RISULTATO
; IL PROGRAMMA
; ci si assicura che il registro usate per gestire l'eventuali prestito siano inizializzato a 0
7   LOAD 0        ; carica il valore 0 nell'accumulatore
8   STORE BX      ;
; produce la differenza tra le parti MENO significative di PRIMO e SECONDO
9   LOAD @1       ; carica LSB_1 nell'accumulatore
10  SUB @3        ; sottrae all'accumulatore LSB_2
11  STORE @5      ; scrive la parte MENO significativa del RISULTATO
; questa STORE non ha alterato il valore del bit CA della SUB che la precede
12  JC 17         ; se CA=1 allora non c'è prestito e si prosegue
13  LOAD @3       ; carica LSB_2 nell'accumulatore
14  SUB 0         ; gli sottrae 0
15  JZ 17         ; se ZE=1 allora LSB_2 era nullo e non c'è prestito e si prosegue
16  DEC BX        ; registra il prestito da scalare
; produce la differenza tra le parti MENO significative di PRIMO e SECONDO
17  LOAD @2       ; carica MSB_1 nell'accumulatore
18  SUB @4        ; sottrae all'accumulatore MSB_2
19  ADD BX        ; scala l'eventuale prestito
20  STORE @6      ; scrive la parte PIU' significativa del RISULTATO
21  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 gestire un segnale di interruzione INT che arriva all'unità di controllo. Assumere di avere un registro SP (*stack pointer*) per la gestione di una zona di memoria con modalità "a pila" ed un registro GI (*gestione interruzioni*) per l'accesso in modalità indiretta alla routine per la gestione delle interruzioni. Descrivere inoltre, con riferimento a tale nuovo schema, sia il flusso dei dati tra le varie componenti della CPU dal momento dell'arrivo del segnale di interruzione al momento in cui si inizia ad eseguire la routine per la gestione delle interruzioni che il microprogramma per la gestione del segnale di interruzione.

SVOLGIMENTO:

Il nuovo schema della CPU, con evidenziate le parti aggiunte, è disegnato più sotto. Il flusso dei dati è sostanzialmente quello descritto, in generale e non con specifico riferimento al nuovo schema della CPU, nei lucidi 34 e 35 del capitolo 12. Prima si salva il valore corrente di PC, all'indirizzo di memoria il cui valore è memorizzato in SP (incrementando successivamente SP). Poi si carica in PC l'indirizzo della prima istruzione della routine di gestione delle interruzioni. Tale indirizzo non è contenuto in GI ma è presente in memoria in una locazione specificata da GI (accesso in modalità indiretta). Il microprogramma per la gestione del segnale di interruzione INT è quindi il seguente:

t1	MAR ← SP	C19, C13
t2	MBR ← PC	C2, C11
t3	(memory) ← MBR	C14, C16, CW
	(SP) + 1	C19, C7, CC, CA, CT
t4	MAR ← GI	C17, C13
t5	SP ← (ALU)	C8, C18
	MBR ← (memory)	C16, C15, CR
t6	PC ← MBR	C22

Successivamente si proseguirà normalmente con la fase di fetch che preleva la prima istruzione di tale routine. L'attivazione del segnale CT al tempo t3 è necessaria per azzerare il buffer interno dell'ingresso della ALU connesso alla porta di controllo C9. Si noti nel diagramma modificato la necessità di generare segnali di clock fino a t7 per le MicroOperazioni,.