Calcolatori Elettronici (12AGA)

Esame del 1.7.2020

Traccia per la correzione parte II

Domanda #1

Si descrivano le funzionalità offerte dal DMA Controller e si illustrino le modalità e i passaggi attraverso i quali avviene un trasferimento in DMA, partendo dalla fase di programmazione.

Traccia per la risposta

- L'uso del DMAC si articola in due fasi:
  - *Programmazione*:
    la CPU scrive l'indirizzo della zona di memoria e il numero di parole da trasferire nei registri IOAR e DC, rispettivamente; la CPU fornisce inoltre al DMAC (in un ulteriore registro di controllo) l'informazione sulla direzione del trasferimento (memoria $\rightarrow$ dispositivo o viceversa) e la modalità di trasferimento (a blocchi, cycle stealing o transparent DMA)
  - *Trasferimento*:
    il DMAC esegue il trasferimento attraverso i seguenti passi :
    - Il DMA Controller riceve una richiesta di trasferimento dal dispositivo periferico
    - Il DMA Controller attiva il segnale DMA Request verso la CPU
    - Quando la CPU riceve il DMA Request, rilascia il bus e attiva il segnale DMA Acknowledge
    - Il DMA Controller inizia il trasferimento; al termine del trasferimento di ciascuna parola, I registri IOAR e DC sono aggiornati
    - Il DMA Controller può sospendere temporaneamente il trasferimento (ad esempio perchè il dispositivo periferico non è pronto per trasferire un nuovo dato) disabilitando DMA Request; la CPU disabilita DMA Acknowledge, e riprende il controllo del bus
    - Quando DC diventa zero, il trasferimento è concluso
    - Il DMA Controller invia una richiesta di interruzione alla CPU.
- Si vedano I lucidi per la descrizione dei modi di trasferimento
  - Burst mode
  - Cycle stealing
  - Transparent DMA

Domanda #2

Si descrivano le differenti soluzioni di arbitraggio di un bus elencandone vantaggi e svantaggi.

Traccia per la risposta

L'arbitraggio di un bus può essere eseguito con tecniche appartenenti a due gruppi

- Arbitraggio distribuito
  - Esempio: bus SCSI – flessibilità, tolleranza ai guasti, velocità e costo medi
- Arbitraggio centralizzato
  - Daisy chaining – massima semplicità HW – lento – non flessibile – non tollerante ai guasti
  - Polling – maggiore complessità dell'arbitro e maggiore numero di linee – più flessibile – più tollerante ai guasti
  - Richieste indipendenti – massima complessità dell'arbitro e massimo numero di linee – massima flessibile –tollerante ai guasti

Si vedano i lucidi per le descrizioni delle varie tecniche

Domanda #3

Si illustri un esempio di articolazione in stadi di un’architettura pipeline (spiegando la funzione di ciascuno stadio) e si descrivano le principali cause di stallo e le principali contromisure software.

Traccia per la risposta

Una possibile architettura a pipeline può essere organizzata in 4 stadi

- Fetch – caricamento istruzione da memoria e aggiornamento PC
- Decode – decodifica e caricamento operandi
- Execute – operazione aritmetico/logica
- Write – scrittura risultato

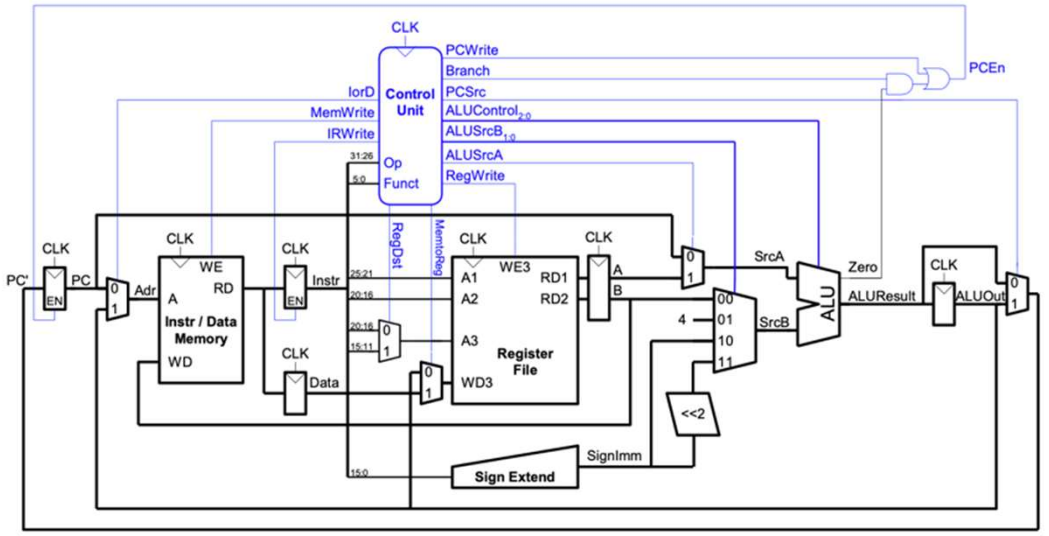
Cause di stallo

- Impossibilità di uno stadio di completare l'operazione in un singolo cc (ad esempio nel caso di operazioni aritmetiche o accesso in memoria)
- Dipendenze di dato
- Salti

Negli ultimi due casi il compilatore può evitare lo stallo inserendo delle istruzioni NOP.

Domanda #4

Utilizzando la tabella riportata, si elenchino le micro-operazioni eseguite da un processore MIPS durante la fase di fetch e l'esecuzione dell'istruzione lw $s1, 12($s2) che assegna al registro destinazione $s1 il valore contenuto all'indirizzo di memoria ($s2 + 12)



Traccia per la soluzione

PCWrite	Branch	PCSrc	ALUControl	ALUSrcB	ALUSrcA	RegWrite	MemtoReg	RegDst	IRWrite	MemWrite	IorD
0	0	-	-	-	-	0	-	-	0	0	0
1	-	0	ADD (010)	01	0	0	-	-	1	0	0
0	0	-	-	-	-	0	-	-	0	0	-
0	0	-	ADD (010)	10	1	0	-	-	0	0	-
0	0	-	-	-	-	0	-	-	0	0	1
0	0	-	-	-	-	0	-	-	0	0	-
0	0	-	-	-	-	1	1	0	0	0	-

Si vedano i lucidi per il dettaglio sulle operazioni eseguite ad ogni cc.