

# Calcolatori Elettronici (12AGA)

Esame del 20.7.2021

Correzione

# Domanda #11

Si consideri un processore connesso ad una memoria da 128kB e dotato di una cache direct mapped da 8 linee, ciascuna da 32 byte.

Assumendo che inizialmente le 8 linee contengano i primi 8 blocchi di memoria (quindi la linea 0 contiene il blocco 0, la linea 1 il blocco 1, e così via), si determini quali dei seguenti 12 accessi in memoria da parte del processore provocano un hit, e quali un miss, scrivendo H o M nella colonna di destra della corrispondente riga nella tabella.

# Domanda #11

Indirizzo	Blocco	Linea acceduta	H/M
0 0000 0000 0011 0011			
1 0000 0000 0001 1000			
0 0000 0010 1000 1110			
0 0000 0000 1011 1110			
0 0000 0000 1001 1111			
1 0000 0000 0011 0011			
0 0000 0101 0001 0011			
0 0000 1010 0101 0100			
0 0000 0011 0011 0100			
0 0100 0000 0011 0110			
0 0000 1000 1001 1000			
1 0000 0000 0011 0111			

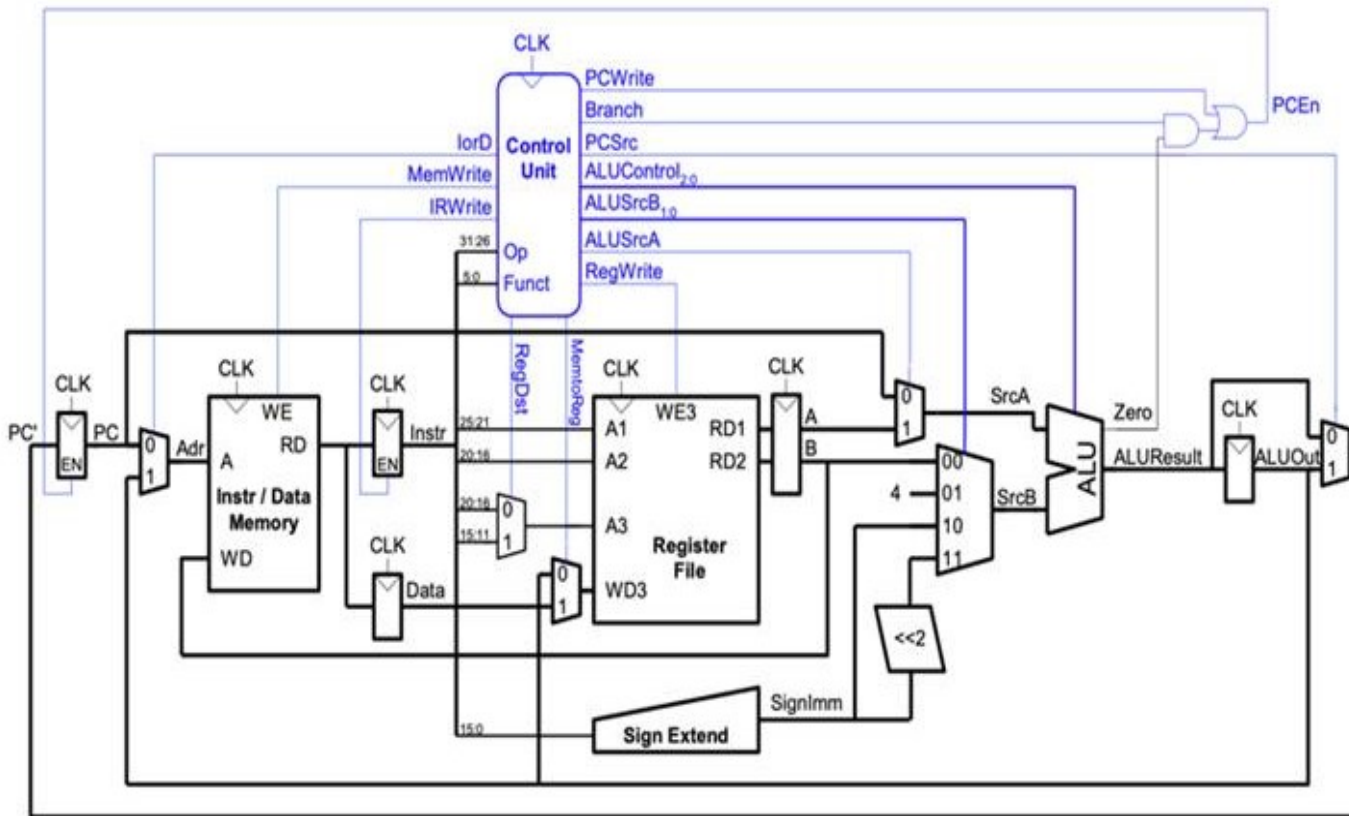
# Domanda #11

Indirizzo	Blocco	Linea acceduta	H/M
0 0000 0000 0011 0011	1	1	H
1 0000 0000 0001 1000	2048	0	M
0 0000 0010 1000 1110	20	4	M
0 0000 0000 1011 1110	5	5	H
0 0000 0000 1001 1111	205	4	M
1 0000 0000 0011 0011	2049	1	M
0 0000 0101 0001 0011	40	0	M
0 0000 1010 0101 0100	80	2	M
0 0000 0011 0011 0100	28	1	M
0 0100 0000 0011 0110	513	1	M
0 0000 1000 1001 1000	68	4	M
1 0000 0000 0011 0111	2049	1	M

## Domanda #12

Utilizzando la tabella riportata, si elenchino le micro-operazioni eseguite da un processore MIPS durante la fase di esecuzione (ignorando il fetch) dell'istruzione `addi $s1, $s2, 40`.

# Domanda #12



Funzione svolta	ALU Control <sub>2:0</sub>
100000 (add)	010 (Add)
100010 (sub)	110 (Sub)
100100 (and)	000 (And)
100101 (or)	001 (Or)
101010 (slt)	111 (SLT)

# Domanda #12

[illegible]

# Domanda #13

Con riferimento al meccanismo del DMA

1. Si descrivano le funzionalità offerte dal DMA Controller
2. Si evidenzino i vantaggi offerti dal DMA rispetto ad altri meccanismi di trasferimento dati da/verso periferiche, quali l'I/O programmato o l'interrupt
3. Si descrivano le connessioni tra il DMA controller e il resto del sistema
4. Si illustrino i passaggi attraverso i quali avviene un trasferimento in DMA, partendo dalla fase di programmazione da parte della CPU.



## Domanda #13.1

Il DMA Controller permette di gestire il trasferimento di dati da una periferica alla memoria del sistema (o viceversa) con un intervento ridotto da parte della CPU.

## Domanda #13.2

Rispetto all'I/O programmato il DMA permette di eseguire il trasferimento senza che la CPU debba verificare per ciascun trasferimento lo stato della periferica ed eseguire direttamente il trasferimento.

Rispetto all'Interrupt il vantaggio sta nel fatto che la CPU deve intervenire solo nella fase di programmazione del DMA Controller.

## Domanda #13.3

Il DMA Controller è connesso al bus di sistema come una normale interfaccia di periferica.

È inoltre connesso alla CPU attraverso i segnali di DMAReq e DMAAck.

Comunica inoltre direttamente con le periferiche da esso gestite.

## Domanda #13.4

- *Programmazione:*  
la CPU carica nei registri IOAR e DC l'indirizzo dell'area di memoria e il numero di parole da trasferire; la CPU informa inoltre il DMA controller della direzione del trasferimento (memoria → periferica o viceversa) e definisce le modalità di trasferimento

# Domanda #13.4

- *Traferimento:*
  - il DMA Controller riceve una richiesta di trasferimento da parte di una periferica
  - il DMA Controller invia un segnale di DMA Request alla CPU
  - quando la CPU giunge a un punto di rilevamento del segnale di DMA Request, rilascia il bus e attiva il segnale di DMA Acknowledge
  - il DMA Controller inizia il trasferimento; dopo il trasferimento di ciascuna parola, IOAR e DC vengono aggiornati
  - il DMA Controller può sospendere temporaneamente il trasferimento (ad esempio perché la periferica non ha più dati da trasferire) disattivando DMA Request; la CPU disattiva DMA Acknowledge, e riprende il controllo del bus
  - quando DC giunge a zero, il trasferimento termina
  - il DMA Controller invia un Interrupt alla CPU.

# Domanda #14

Un sommatore può essere realizzato utilizzando varie architetture, tra le quali

- Sommatore seriale
- Sommatore di tipo Ripple Carry Adder
- Sommatore di tipo Carry Lookahead Adder.

Per ciascuna di esse:

1. Si elenchino i componenti hardware richiesti, specificando quanti componenti sono necessari per realizzare un sommatore su  $n$  bit e descrivendo i collegamenti tra i vari componenti
2. Se ne descriva il funzionamento
3. Si illustrino vantaggi e svantaggi.

# Domanda #14.1

- Sommatore seriale
  - 1 full-adder e 1 DFF: l'ingresso del FF è connesso al Cout del FA, l'uscita al Cin
- Sommatore di tipo Ripple Carry Adder
  - n FA connessi in cascata: il Cout dell'uno entra nel Cin del successivo
- Sommatore di tipo Carry Lookahead Adder
  - n FA modificati connessi ad una logica di generazione del carry che alimenta tutti i Cin.

# Domanda #14.2

- Sommatore seriale
  - Si resetta il FF
  - Ad ogni periodo di clock si inviano al FA due bit di peso corrispondente, uno per ciascun operando. L'eventuale Cout viene memorizzato dal FF.
- Sommatore di tipo Ripple Carry Adder
  - Ciascun FA riceve due bit di peso corrispondente, uno per ciascun operando, oltre al carry proveniente dal FA precedente.
- Sommatore di tipo Carry Lookahead Adder
  - Ciascun FA riceve due bit di peso corrispondente, uno per ciascun operando e produce i bit g e p, che vanno alla logica di generazione del carry
  - La logica produce i bit di carry per ciascun FA
  - Ciascun FA produce il bit risultato.



# Domanda #14.3

- Sommatore seriale
  - Hardware di costo minimo
  - Richiede  $n$  periodi di clock per produrre il risultato completo
- Sommatore di tipo Ripple Carry Adder
  - Hardware di costo linearmente proporzionale a  $n$
  - Ritene tempo pari a  $n\Delta$  per produrre il risultato completo, essendo  $\Delta$  il ritardo del singolo FA
- Sommatore di tipo Carry Lookahead Adder
  - Hardware di costo esponenzialmente proporzionale a  $n$
  - Ritene tempo pari a  $3d$  per produrre il risultato completo, essendo  $d$  il ritardo di un circuito a due livelli (FA o logica di generazione del carry).