

Calcolatori Elettronici (12AGA)

Esame del 2.7.2024

Traccia di soluzioni per parte 2

Domanda #11

Si descriva il principio di località dei riferimenti riportando qualche esempio di comportamento che lo dimostri e spiegando le ragioni che lo supportano.

Si riporti anche in quali contesti architettureali viene sfruttato il principio di località dei riferimenti.

Note

Il testo richiedeva

- Di enunciare il principio di località dei riferimenti
- Di riportare qualche esempio di comportamento che lo dimostri, spiegando le ragioni che lo supportano
- Di elencare in quali contesti architetturali viene sfruttato (ad esempio, cache e memoria virtuale).

Il principio si applica a qualsiasi accesso in memoria, per dati o istruzioni.

Domanda #13

Si consideri un processore connesso a una memoria da 1 Kbyte e dotato di una cache direct mapped composta da 8 linee da 32 byte ciascuna.

Si assuma che la cache sia inizialmente vuota e che il processore esegua una serie di accessi in memoria in cui genera i seguenti indirizzi:

0001001010, 0111001100, 1000110100, 1100000000, 0101100110,
0001001000, 1010000010, 0010100010, 0111001000, 1100011000,
0001000000, 0000110100, 1010010010, 1001000000, 0011100000

Per ogni accesso, si indichi la linea in cui si trova eventualmente il blocco di memoria referenziato e se l'accesso in cache determina un hit o un miss.

Si specifichi il numero del blocco memorizzato in ciascuna linea della cache al termine della sequenza.

Sequenza di accessi

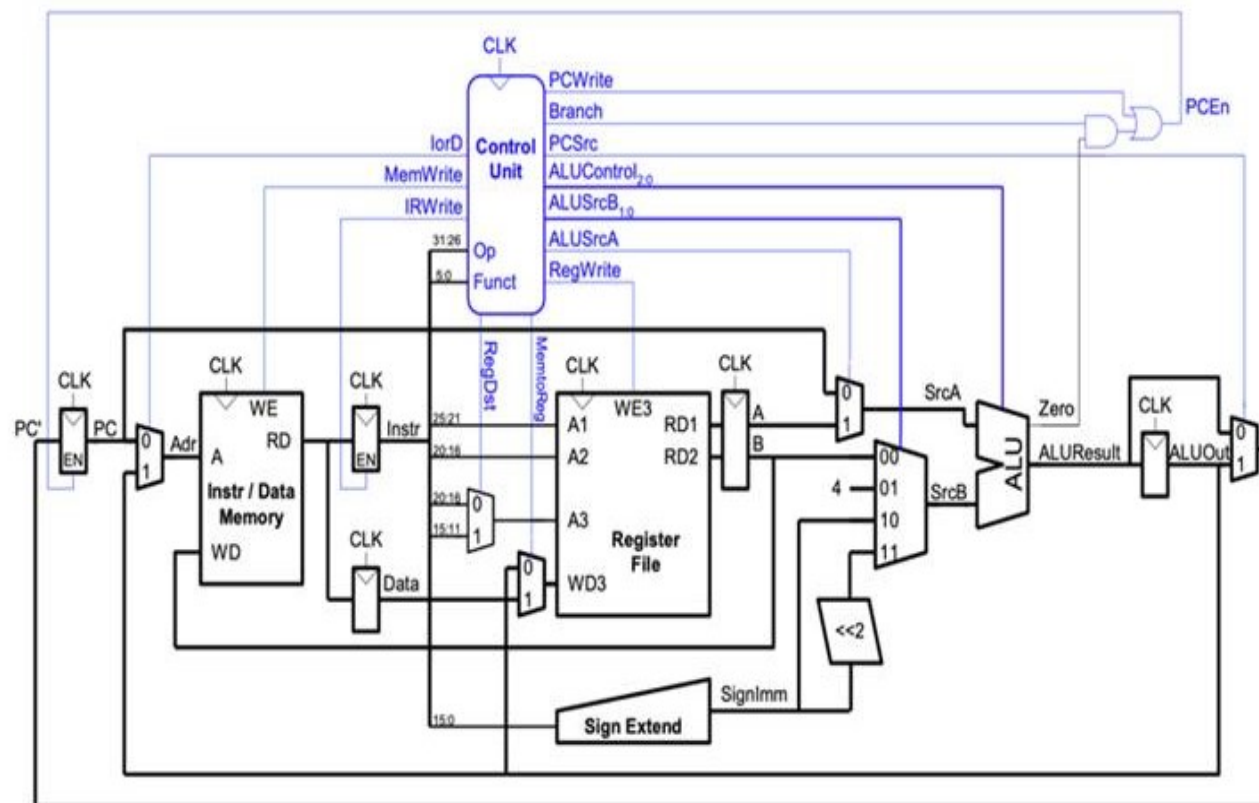
	linea	Blocco	
00 010 01010	2	00010	M
01 110 01100	6	01110	M
10 001 10100	1	10 001	M
11 000 00000	0	11 000	M
01 011 00110	3	01011	M
00 010 01000	2	00010	H
10 100 00010	4	10100	M
00 101 00010	5	00101	M
01 110 01000	6	01110	H
11 000 11000	0	11000	H
00 010 00000	2	00010	H
00 001 10100	1	00001	M
10 100 10010	4	10100	H
10 010 00000	2	10010	M
00 111 00000	7	00111	M

Contenuto finale della cache

Linea	Numero del blocco
0	11000 – 24
1	00001 – 1
2	10010 – 18
3	01011 – 11
4	10100 – 20
5	00101 – 5
6	01110 – 14
7	00111 - 7

Domanda #14

Utilizzando la tabella riportata, si elenchino le micro-operazioni eseguite da un processore MIPS durante la fase di esecuzione (ignorando il fetch) dell'istruzione `sw $s2, 100($s1)`.



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

Domanda #12

	PCWrite	Branch	PCSrc	ALUControl	ALUSrcB	ALUSrcA	RegWrite	MemtoReg	RegDst	IRWrite	MemWrite	IorD
1	0	0					0	0			0	0
2	0	0		010	10	1	0	0			0	0
3	0	0					0	0			1	1

cc1: lettura da RegFile

cc2: esecuzione somma tramite ALU

cc3: scrittura in memoria

Domanda #12

Indirizzo	Blocco (riportarlo anche in forma binaria)	Numero di linea	H/M
0100 0000 0011 0011	0100 0000 0011 = $1024+2+1$	3	M
1010 1000 1000 0011	1010 1000 1000 = $2048+512+128+8$	8	M
0000 0000 0101 0100	0000 0000 0101 = $4+1$	5	H
0000 1000 1001 1000	0000 1000 1001 = $128+8+1$	9	M
0000 0100 0111 1010	0000 0100 0111 = $64+4+2+1$	7	M
0000 0100 0010 0101	0000 0100 0010 = $128+2=130$	2	M
0101 0000 1111 0110	0101 0000 1111 = $1024+256+8+4+2+1$	15	M
0000 1000 1001 1111	0000 1000 1001 = $128+8+1$	9	H
0000 0000 1000 1100	0000 0000 1000 = 16	8	M
0000 0011 0011 0100	0000 0011 0011 = $32+16++2+1$	19	M
0000 0011 0011 0110	0000 0011 0011 = $32+16++2+1$	19	H
1010 1000 1000 0110	1010 1000 1000 = $2048+512+128+8$	8	M