Architettura degli Elaboratori

Simone Lidonnici

5 maggio 2024

Indice

E

Esercizi

E.1 Esercizi sulla CPU

E.1.1 Divisione in bit delle istruzioni

Nome	Campi						Commenti
Dimensione	6 bit	5 bit	5 bit	5 bit	5 bit	5 bit	Le istruzioni in MIPS sono a 32 bit
Formato R	op	rs	rt	rd	shamt	funct	Formato delle istruzioni aritmetiche
Formato I	op	rs	rt	indirizzo/costante			Formato delle istruzioni di trasferimento dati, di salto condizionato e immediate
Formato J	op	indirizzo di destinazione					Formato delle istruzioni di salto incondizionato

E.1.2 Segnali ALU

Istruzione	ALUOP		Campi funct					Segnali ALU				Operazione
lw e sw	0	0	-	-	-	-	-	0	0	1	0	ADD
beq	-	1	-	-	-	-	-	0	1	1	0	SUB
add	1	-	-	0	0	0	0	0	0	1	0	ADD
sub	1	-	-	0	0	1	0	1	1	1	0	SUB
and	1	-	-	0	1	0	0	0	0	0	0	AND
or	1	-	-	0	1	0	1	0	0	0	1	OR
slt	1	-	-	1	0	1	0	0	1	1	1	SLT

E.1.3 Segnali attivi per ogni istruzione

Istruzione	RegDst	RegWrite	ALUSrc	ALUOP		MemRead	MemWrite	MemToReg	Branch	Jump
Tipo R	1	1	0	1	-	-	0	0	0	0
Tipo I	0	1	1	1	-	-	0	0	0	0
lw	0	1	1	0	0	1	0	1	0	0
sw	-	0	1	0	0	0	1	-	0	0
beq	-	0	0	-	1	-	0	-	1	0
j	-	0	-	-	-	-	0	-	-	1

E.1.4 Numeri dei registri

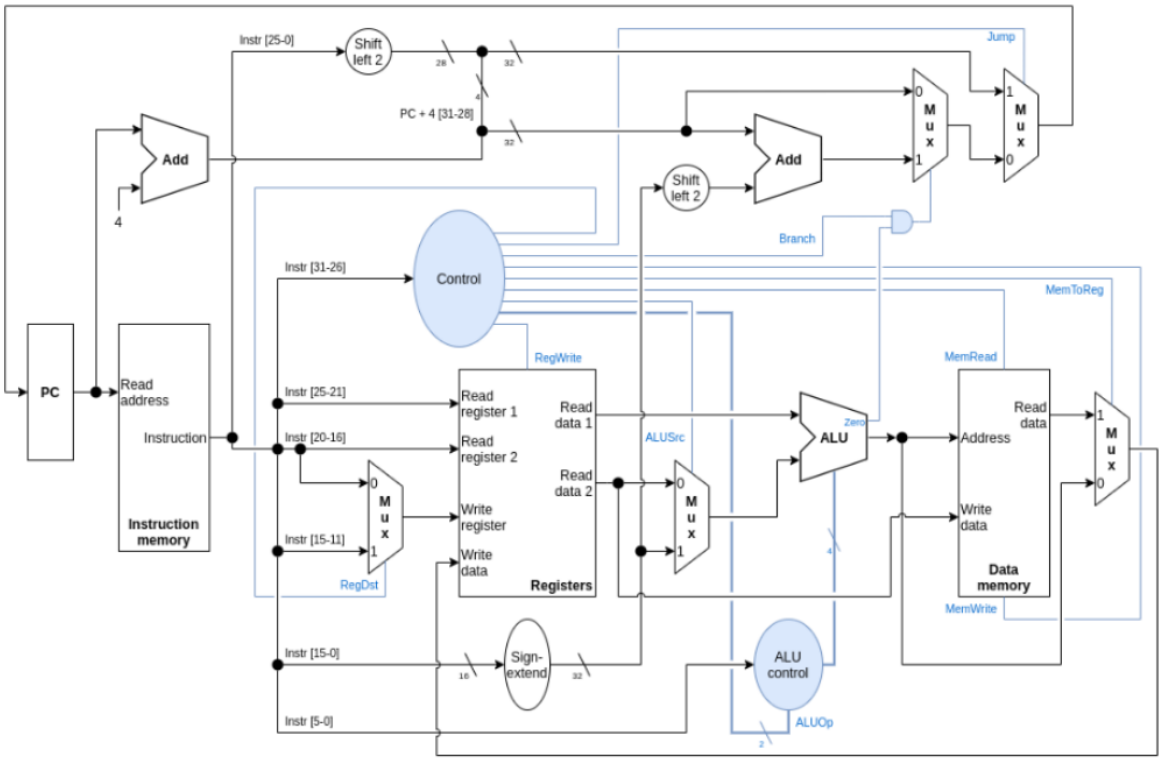
Registri utili:

Registro	Numero
$at	1
$v0	2
$a0	4
$t0	8
$s0	16

Registri totali:

Registro	Numero
$at	1
$v0	2
$v1	3
$a0	4
$a1	5
$a2	6
$a3	7
$t0	8
$t1	9
$t2	10
$t3	11
$t4	12
$t5	13
$t6	14
$t7	15
$s0	16
$s1	17
$s2	18
$s3	19
$s4	20
$s5	21
$s7	22
$s8	23
$t8	24
$t9	25
$k0	26
$kl	27
$gp	28
$sp	29
$fp	30
$ra	31

E.1.5 Circuito completo senza pipeline



E.2 Esercizi sulla pipeline

E.2.1 Dove si applica il forwarding

Dopo→ Prima ↓	lw	Operazione matematica o logica: add, sub, and	Salto condizionato	sw
lw	mem→ exe	mem→ exe	mem→ decode	mem→ mem
Operazione matematica o logica: add, sub, and	exe→ exe	exe→ exe	exe→ decode	exe→ exe
li	exe→ exe	exe→ exe	exe→ decode	exe→ exe
sw	exe→ exe	exe→ exe	exe→ decode	//

E.3 Esercizi sulla Cache

E.3.1 Cache Direct-Mapped

Memoria con l linee(set) e w word per ogni blocco.
Tutte le dimensioni sono da considerarsi in bit tranne la dimensione della cache.

Nome	Formule	Notazione
Dimensione indice di linea	$\log_2(l)$	n
Dimensione offset di word	$\log_2(w \cdot 4)$	m
Dimensione di un blocco	$w \cdot 4 \cdot 8$	DBl
Dimensione del tag	$32 - n - m$	tag
Dimensione di una linea	V.bit(1)+tag+DBl	DLn
Dimensione della cache (in byte)	$l \cdot DLn$	DC

Esempio:
$l = 32$
$w = 16$
$n = \log_2(32) = 5$
$m = \log_2(16 \cdot 4) = 6$
$\text{DBl} = 16 \cdot 4 \cdot 8 = 512$
$\text{tag} = 32 - 5 - 6 = 21$
$\text{DLn} = 1 + 21 + 512 = 534$
$\text{DC} = 32 \cdot 534 = 17088 \text{ bit} = 2134 \text{ byte} \approx 2.1 \text{ KB}$

E.3.2 Cache Set-Associative

Memoria con v vie, l linee(set) e w word per ogni blocco.
Tutte le dimensioni sono da considerarsi in bit tranne la dimensione della cache.

Nome	Formule	Notazione
Dimensione indice di linea	$\log_2(l)$	n
Dimensione offset di word	$\log_2(w \cdot 4)$	m
Dimensione di un blocco	$w \cdot 4 \cdot 8$	DBl
Dimensione del tag	$32 - n - m$	tag
Dimensione di una linea	V.bit(1)+LRU bit(1)+tag+DBl	DLn
Dimensione della cache (in byte)	$v \cdot l \cdot DLn$	DC

Esempio:
$v = 4$
$l = 32$
$w = 16$
$n = \log_2(32) = 5$
$m = \log_2(16 \cdot 4) = 6$
$\text{DBl} = 16 \cdot 4 \cdot 8 = 512$
$\text{tag} = 32 - 5 - 6 = 21$
$\text{DLn} = 1 + 1 + 21 + 512 = 535$
$\text{DC} = 4 \cdot 32 \cdot 535 = 68480 \text{ bit} = 8560 \text{ byte} \approx 8.36 \text{ KB}$

E.3.3 Riempire i campi di una linea

Dato un indirizzo A e una memoria con l linee e w word per blocco.
I risultati devono essere numeri interi e vanno approssimati per difetto. (% indica il resto della divisione)

Nome	Formule	Notazione
Numero del blocco	$\lfloor \frac{A}{4w} \rfloor$	#BL
Indice di linea	$\#\text{BL}\%l$	Id
Tag	$\lfloor \frac{\#\text{Bl}}{l} \rfloor$	tag
Offset del blocco	$A\%(4w)$	OBl
Offset word	$\lfloor \frac{\text{OBl}}{4} \rfloor$	Ow

Esempio:
$w = 8$
$l = 4$
Chiamate alla cache sequenziali, ricordando che il bit di validità all'inizio è 0 e dopo che viene chiamato per la prima volta un indice diventa 1

	128	130	162	40	47	8
#Bl	4	4	5	1	1	0
Id	0	0	1	1	1	0
tag	1	1	1	0	0	0
Hit/Miss	MISS	HIT	MISS	MISS	HIT	MISS

Tipi di MISS:

- **Cold start**: se #Bl non è mai stato caricato
- **Cap**: se anche in una cache fully-associative #Bl darebbe un MISS
- **Conflict**: se il #Bl è stato già caricato ma poi sovrascritto da un'altro, ma non darebbe MISS in una cache fully-associative