

Architettura degli Elaboratori

Tutorato

a cura di Manuela Flores

16/12/2022 ore 9 – 11:30



Barbara Masucci – Classe: Resto_1

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

Multiple 1-4

1)

Si consideri l'implementazione del processore MIPS a singolo ciclo, come studiato. Durante l'esecuzione dell'istruzione **sw \$s1, 24(\$s2)** quale sarà il **primo** ingresso alla ALU?

- A. Il contenuto del registro \$s1
- B. 24
- C. Il contenuto del registro \$s2
- D. Nessuna delle risposte precedenti

2)

Il pipelining:

- A. Riduce il numero di istruzioni eseguibili nell'unità di tempo
- B. Aumenta il numero di istruzioni eseguibili nell'unità di tempo
- C. Riduce il tempo per eseguire una singola istruzione
- D. Nessuna delle risposte precedenti

3)

Se in un flip-flop D temporizzato, il segnale di Clock è 1, allora

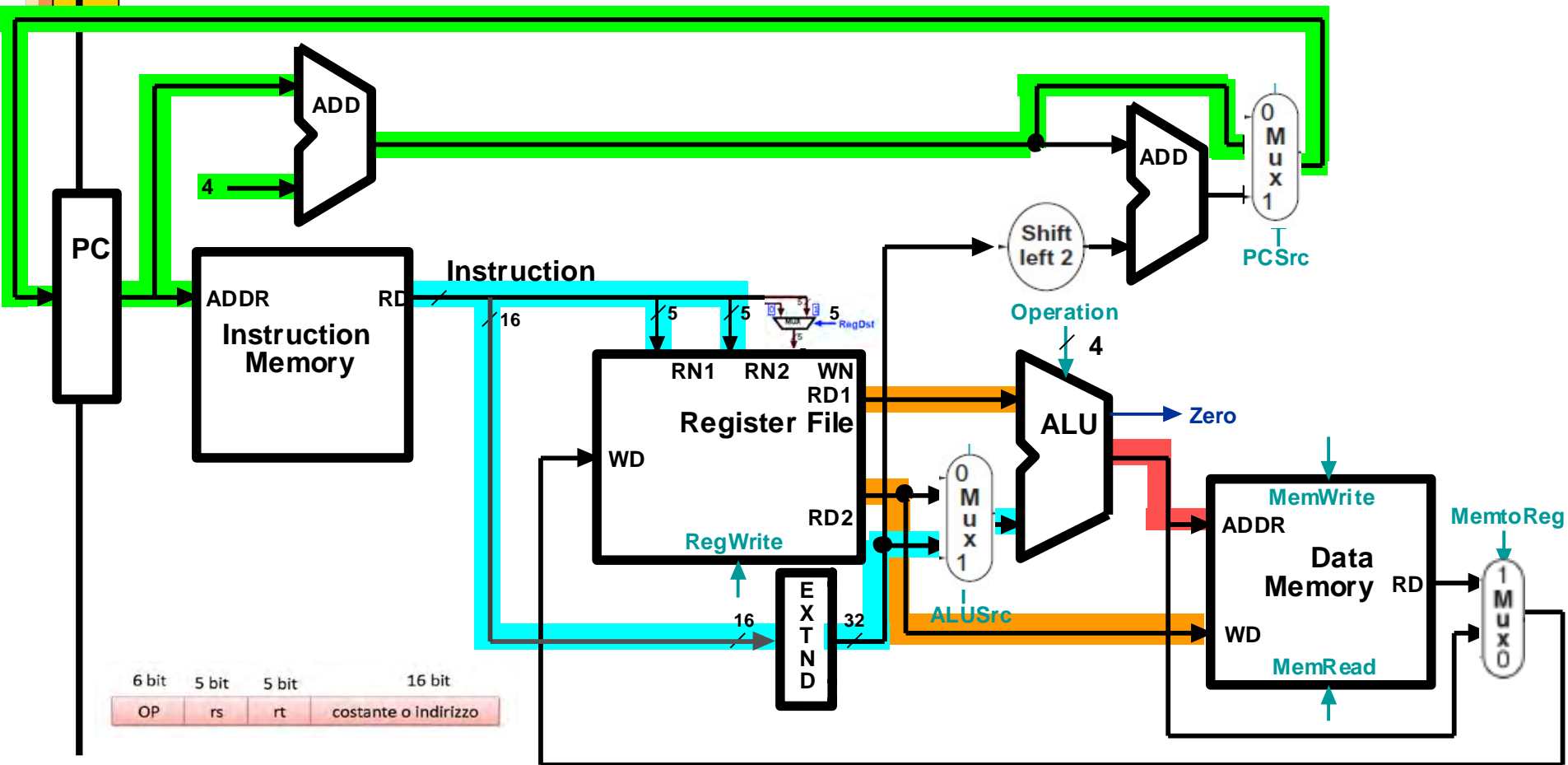
- A. Il flip-flop è aperto e l'uscita Q sarà uguale a D
- B. Il flip-flop è aperto e l'uscita Q sarà uguale a 1
- C. Il flip-flop è chiuso e l'uscita Q resta invariata
- D. Nessuna delle risposte precedenti

4)

In una memoria cache a mappatura diretta si ha una *hit* (successo), in seguito alla richiesta di un indirizzo, se:

- A. L'indirizzo era stato già richiesto in precedenza (dall'accensione della macchina)
- B. Nella linea indicata dal campo indice dell'indirizzo, il bit di validità è posto a 1 oppure il campo tag è uguale al campo tag dell'indirizzo
- C. Nella linea indicata dal campo indice dell'indirizzo, il bit di validità è posto a 1 e il campo tag è diverso dal campo tag dell'indirizzo
- D. Nessuna delle risposte precedenti

Implementazione a Ciclo Singolo



Datapath dell' istruzione sw

`sw rt, offset(rs)`

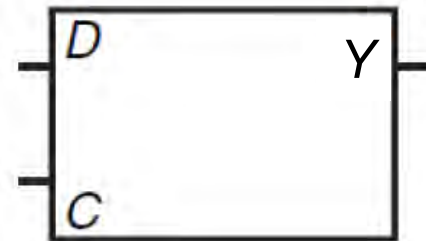
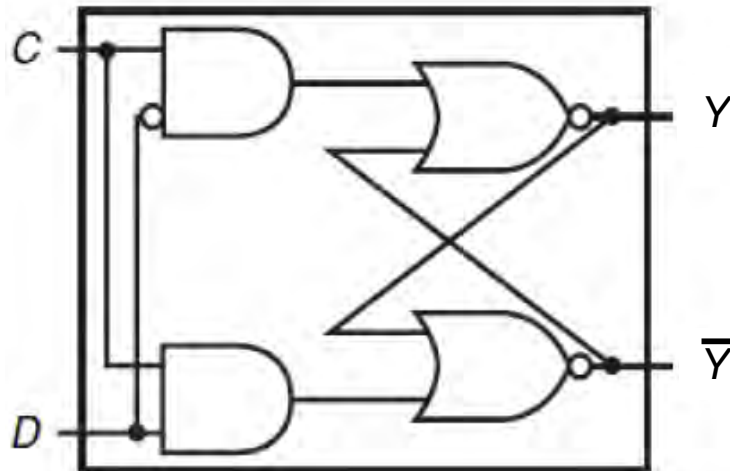
Osservazioni

- L'implementazione con pipeline aumenta il **numero di istruzioni contemporaneamente in esecuzione**
- Quindi, la pipeline
 - Aumenta il **throughput**
(numero di istruzioni eseguite nell'unità di tempo)
 - Non diminuisce la **latenza**
(tempo di esecuzione della singola istruzione), che resta sempre la stessa



Flip-Flop D

- Quando il Clock è 1, cioè il suo fronte è **alto**, il circuito è **aperto** e l'output Y diventa uguale all'input D (il circuito lascia passare il dato D)
- Quando il Clock è 0, cioè il suo fronte è **basso**, il circuito è **chiuso** e l'output Y mantiene il valore che aveva l'ultima volta che il circuito era aperto (il circuito non lascia passare il dato D)



Lezione 23 pag. 20

Migrazione delle informazioni

- Consideriamo **due livelli adiacenti** della gerarchia
 - Livello superiore
 - Livello inferiore
- **Blocco o linea**
 - La più piccola quantità di informazione che può essere trasferita tra due livelli adiacenti nella gerarchia
 - **Analogia con la biblioteca:** blocco = libro
- **Hit (successo)**
 - Se il dato richiesto dal processore è contenuto in uno dei blocchi presenti nel livello superiore
 - **Analogia con la biblioteca:** l'informazione cercata dallo studente è presente in uno dei libri a disposizione sulla scrivania

Lezione 23 pag. 21

Migrazione delle informazioni

➤ Miss (fallimento)

- Se il dato richiesto dal processore non è contenuto in uno dei blocchi presenti nel livello superiore
- In tal caso bisogna **accedere al livello inferiore** della gerarchia per recuperare il blocco contenente il dato richiesto e **copiarlo nel livello superiore**

➤ Analogia con la biblioteca

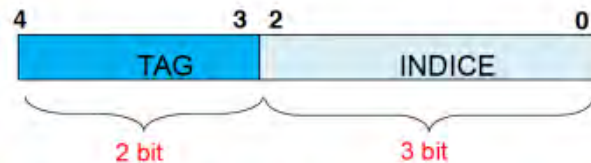
- L'informazione cercata dallo studente non è presente in nessuno dei libri sulla scrivania
- In tal caso lo studente deve **cercare tra gli scaffali della biblioteca** per recuperare il libro contenente l'informazione di cui ha bisogno
- Se lo studente trova il libro, **lo porta sulla scrivania**



Lezione 23 pag. 37

Hit!

Se ricevo un indirizzo dalla memoria



Avrò una **hit** (dato presente nella cache) se, nella linea di cache INDICE

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})

Il tag coincide con **TAG** e $V=1$



Multiple 5-8

5)

Nel processore a singolo ciclo studiato, l'unità di controllo e il banco dei registri sono rispettivamente:

- A. un circuito combinatorio e un circuito sequenziale
- B. un circuito sequenziale e un circuito combinatorio
- C. entrambi circuiti combinatori
- D. Nessuna delle risposte precedenti.

6)

Si consideri l'implementazione studiatà del processore con pipeline.

Cosa fa l'istruzione `lw $s1, 24($s2)` nello stadio MEM?

- A. Esegue la somma del contenuto del registro `$s2` con 24
- B. Scrive nel registro `$s1` il dato prelevato dalla memoria
- C. Preleva l'istruzione dalla memoria istruzioni
- D. Nessuna delle risposte precedenti

7)

La lettura di un registro nel banco dei registri (*register file*) avviene tramite

- A. un multiplexer che ha il numero del registro come segnale di controllo
- B. un decoder che ha in input il dato da scrivere
- C. un decoder che ha in input il numero del registro
- D. Nessuna delle risposte precedenti

8)

Nel processore a singolo ciclo (come studiato) il segnale di controllo `MemRead`:

- A. è 1 per tutte le istruzioni e permette la lettura dalla memoria istruzioni
- B. è 1 per l'istruzione `sw` e permette la lettura dalla memoria dati
- C. non esiste
- D. Nessuna delle precedenti

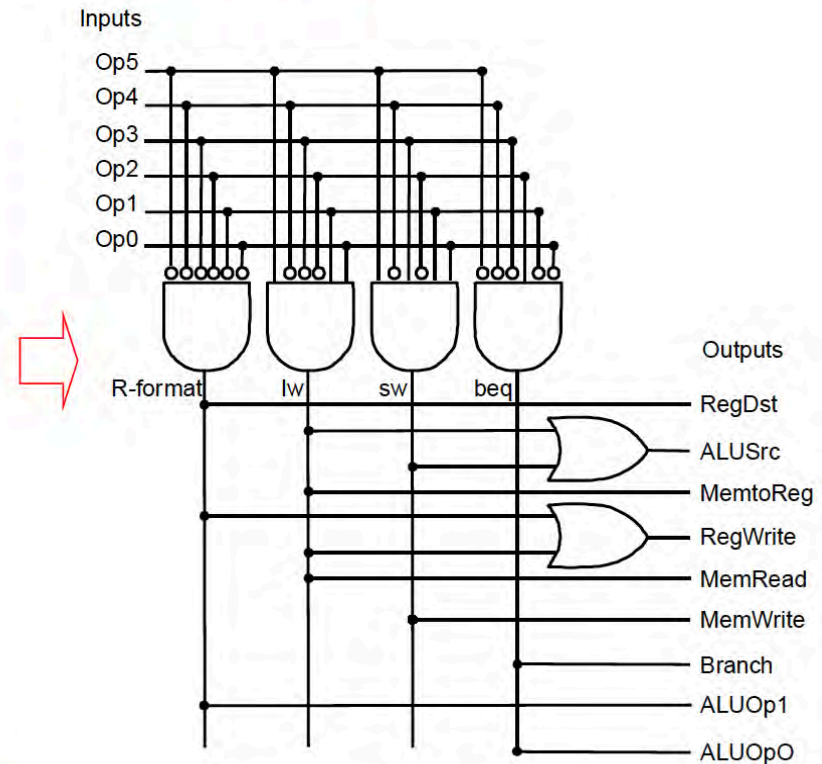
Unità di controllo

Implementazione

Tabella di verità dell'unità di Controllo

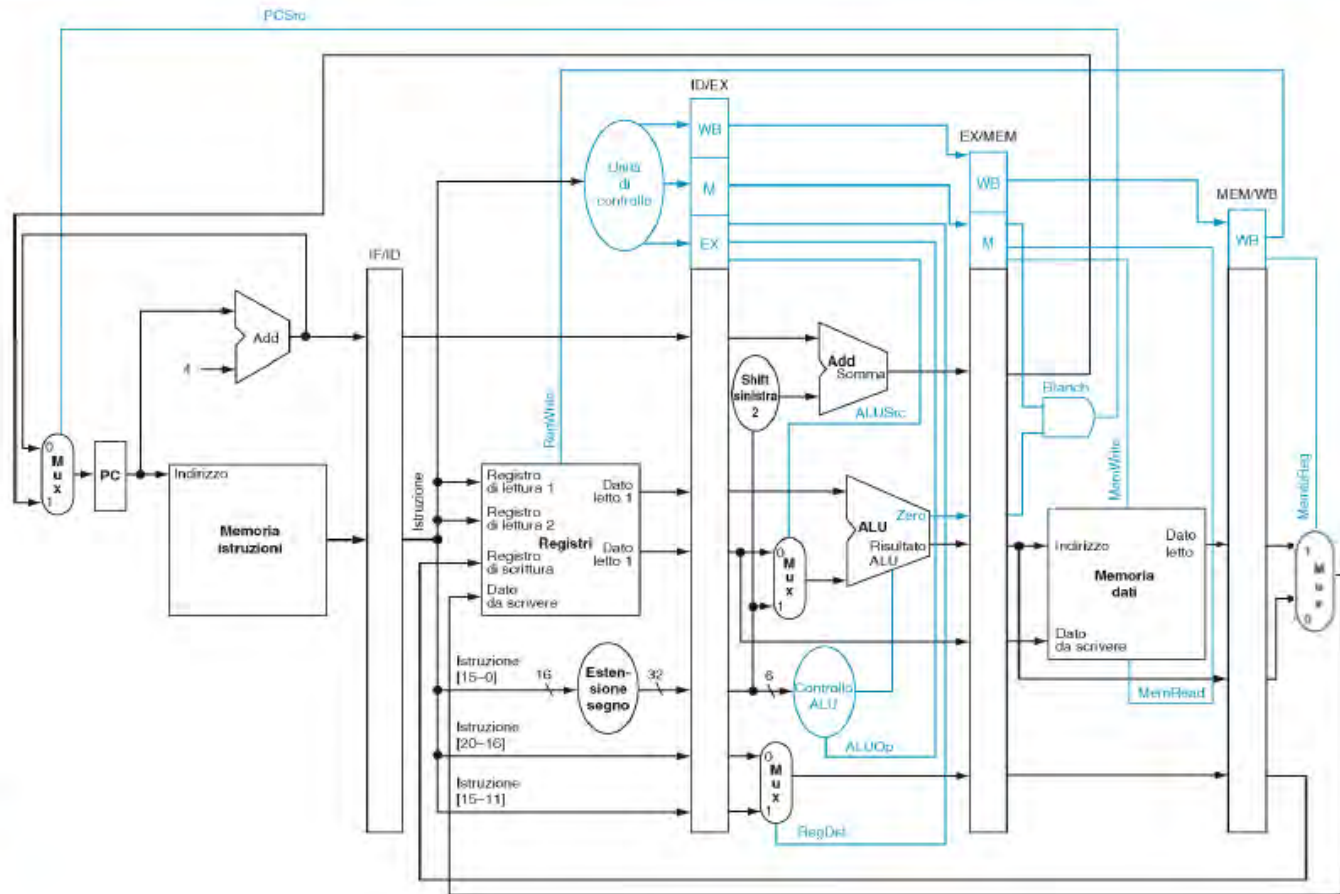
	Segnale	form.			
		R	lw	sw	beq
Inputs = Opcode	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	x	x
	ALUSrc	0	1	1	0
	MemtoReg	0	1	x	x
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Rete Combinatoria realizzabile tramite PLA



Lezione 22 pag. 3

Unità di elaborazione con pipeline e controllo

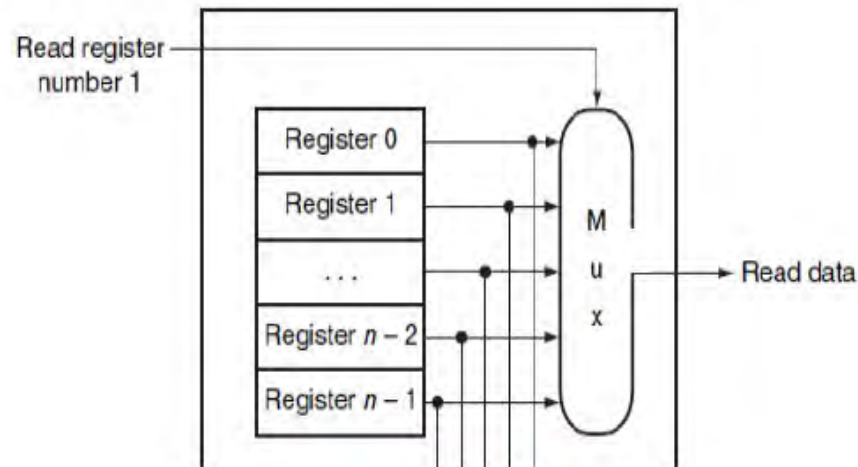


Lezione 16 pag. 28

Leggere in un registro

➤ Realizzazione:

- Usa un MUX $2^5:1$ in cui il register number è usato per ottenere i 5 segnali di selezione del MUX
- L'output del MUX è il contenuto del registro indicato



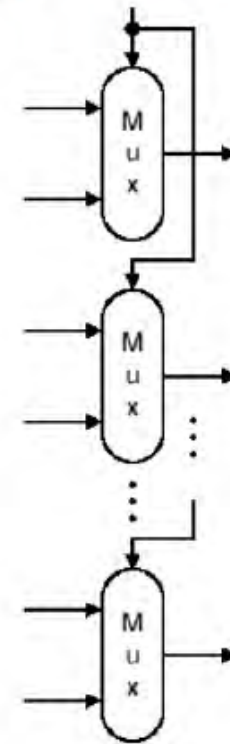
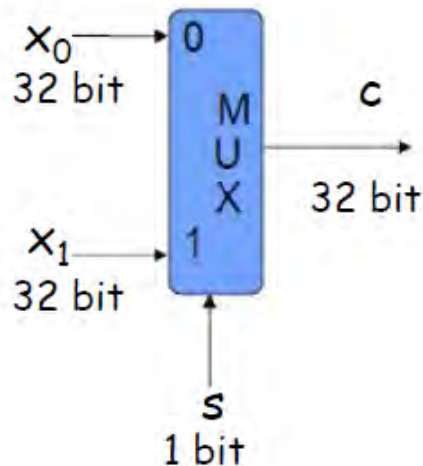
Multiplexer $2^5:1$



Lezione 9 pag. 15

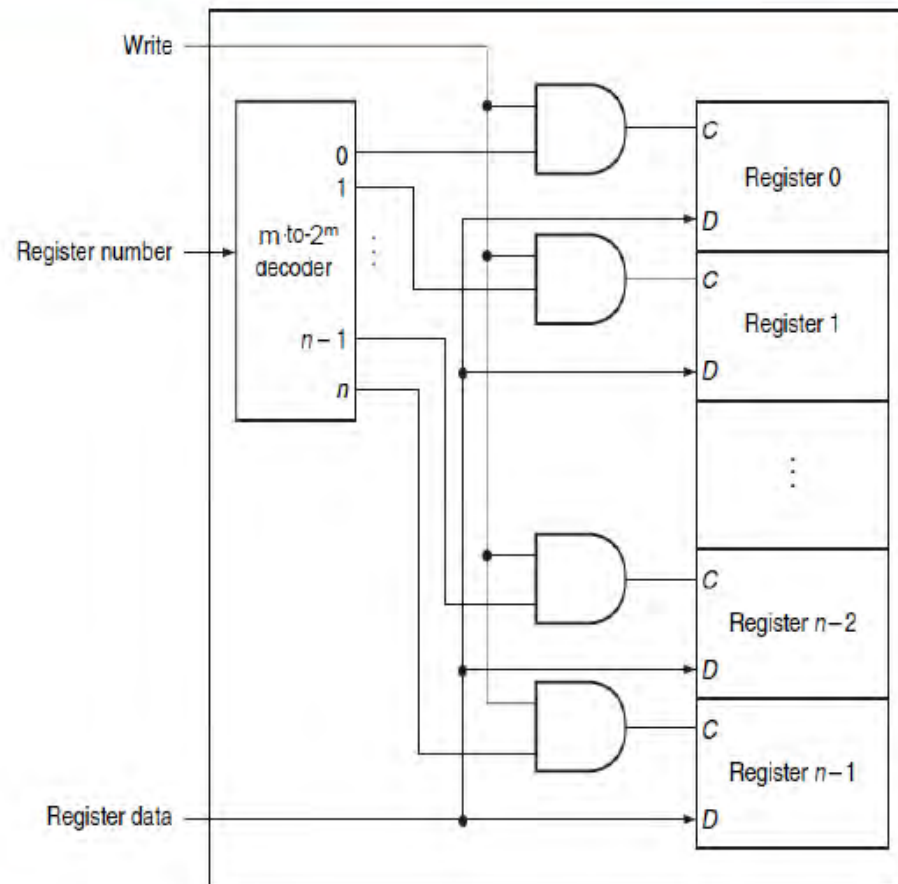
Multiplexer 2:1 a 32 bit

- Costruito con 32 multiplexer 2:1 con un segnale di controllo distribuito ai vari multiplexer



Lezione 16 pag. 30

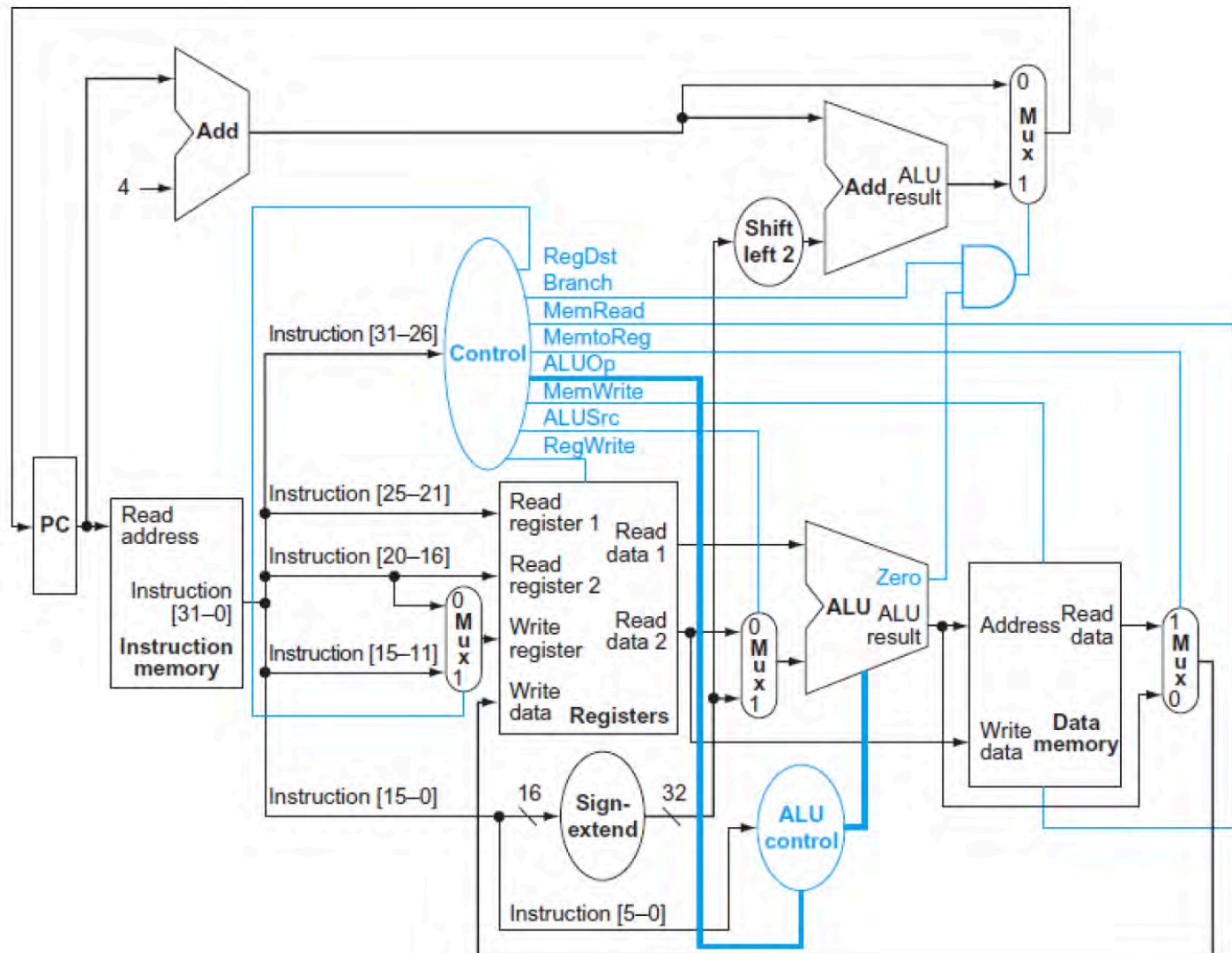
Scrivere in un registro



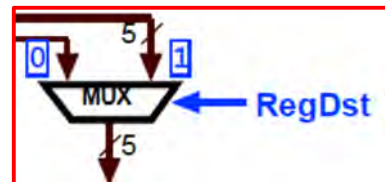
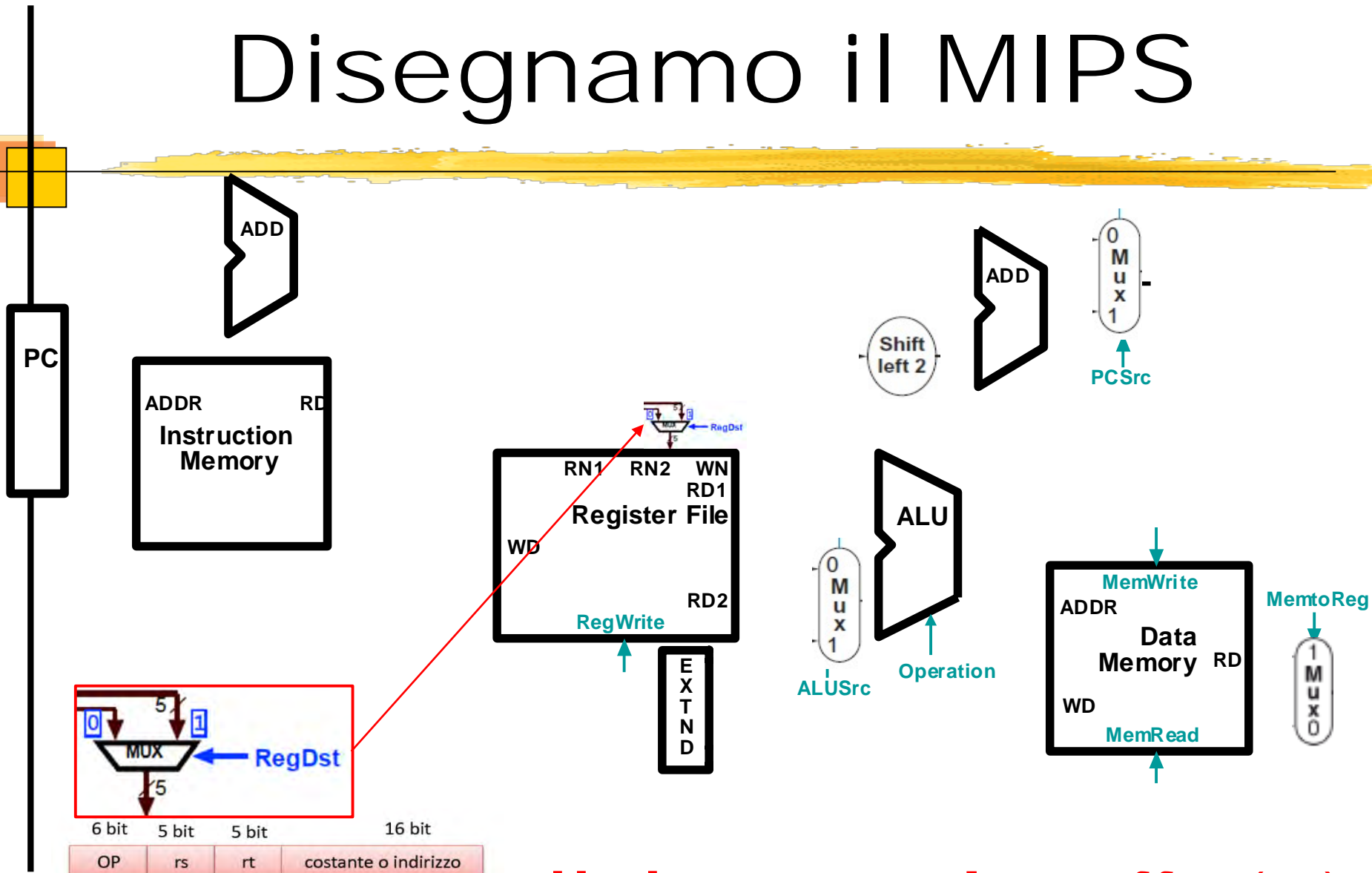
$$m=5$$
$$n=2^m=32$$



Implementazione a Ciclo Singolo



Disegniamo il MIPS



6 bit	5 bit	5 bit	16 bit		
OP	rs	rt	costante o indirizzo		

6 bit	5 bit	5 bit	5 bit	5 bit	6 bit
OP	rs	rt	rd	shamt	funct

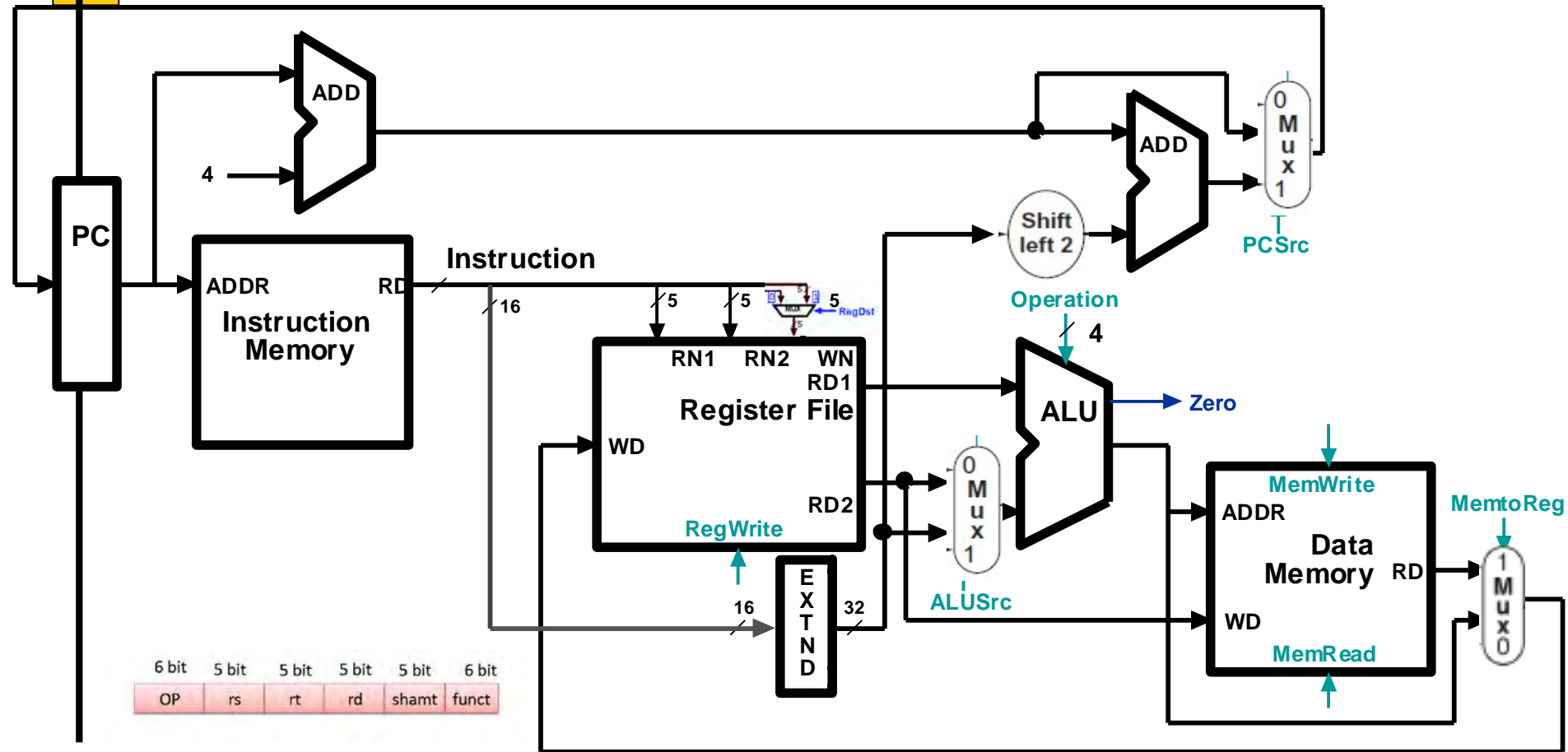
add rd,rs,rt

beq rs,rt,offset

lw rt,offset(rs)

sw rt,offset(rs)

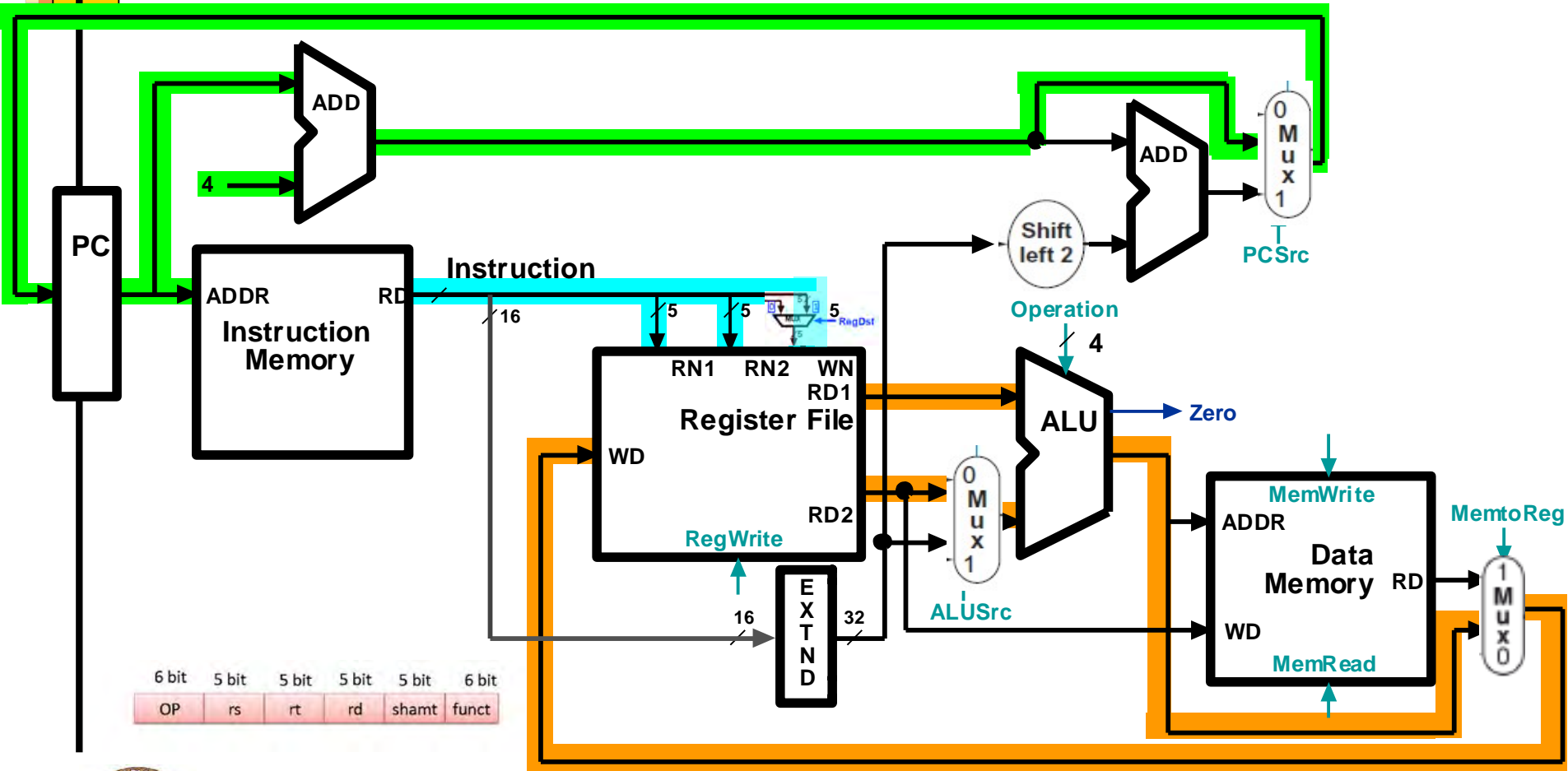
Implementazione a Ciclo Singolo



add rd,rs,rt



Implementazione a Ciclo Singolo

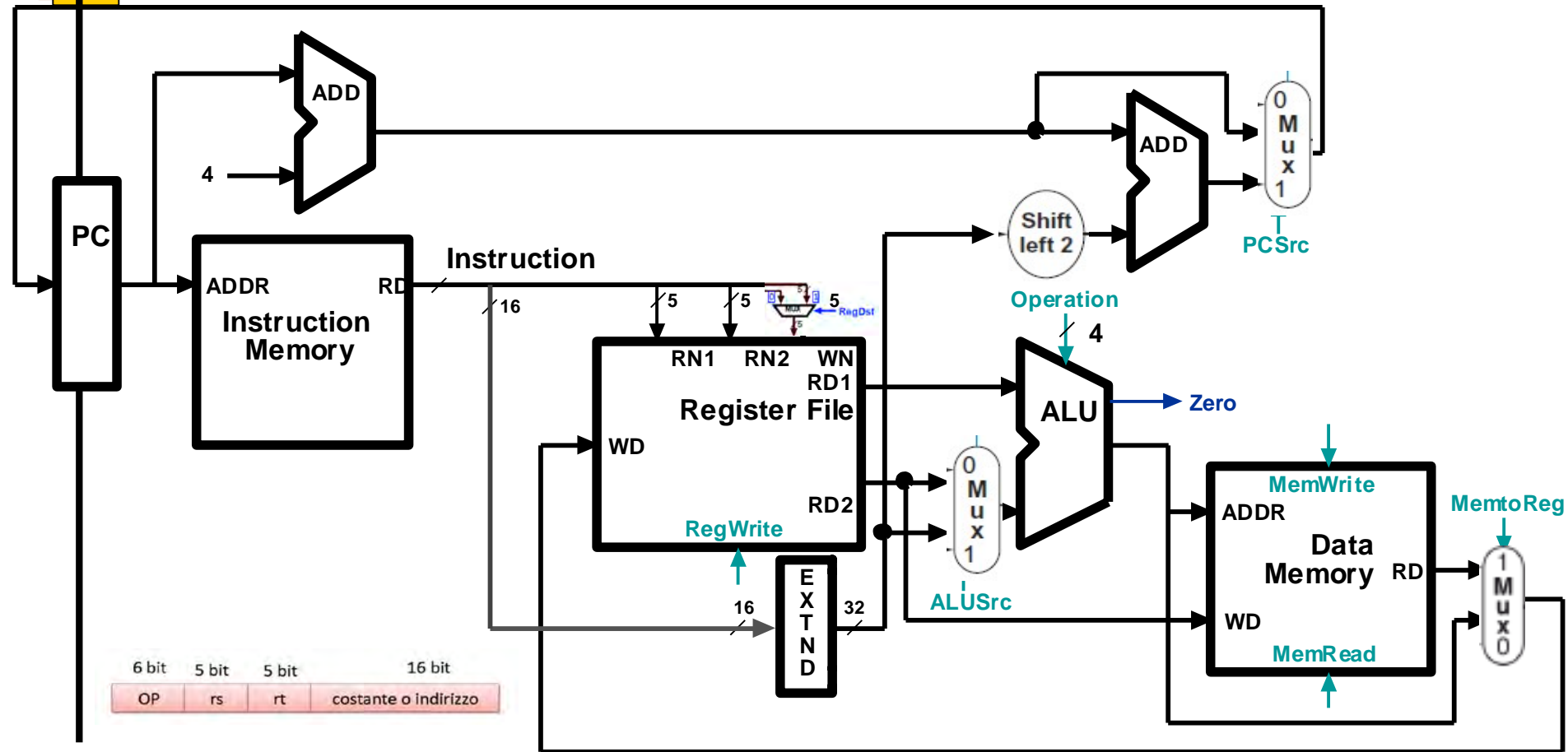


Datapath dell' istruzione add

add rd,rs,rt



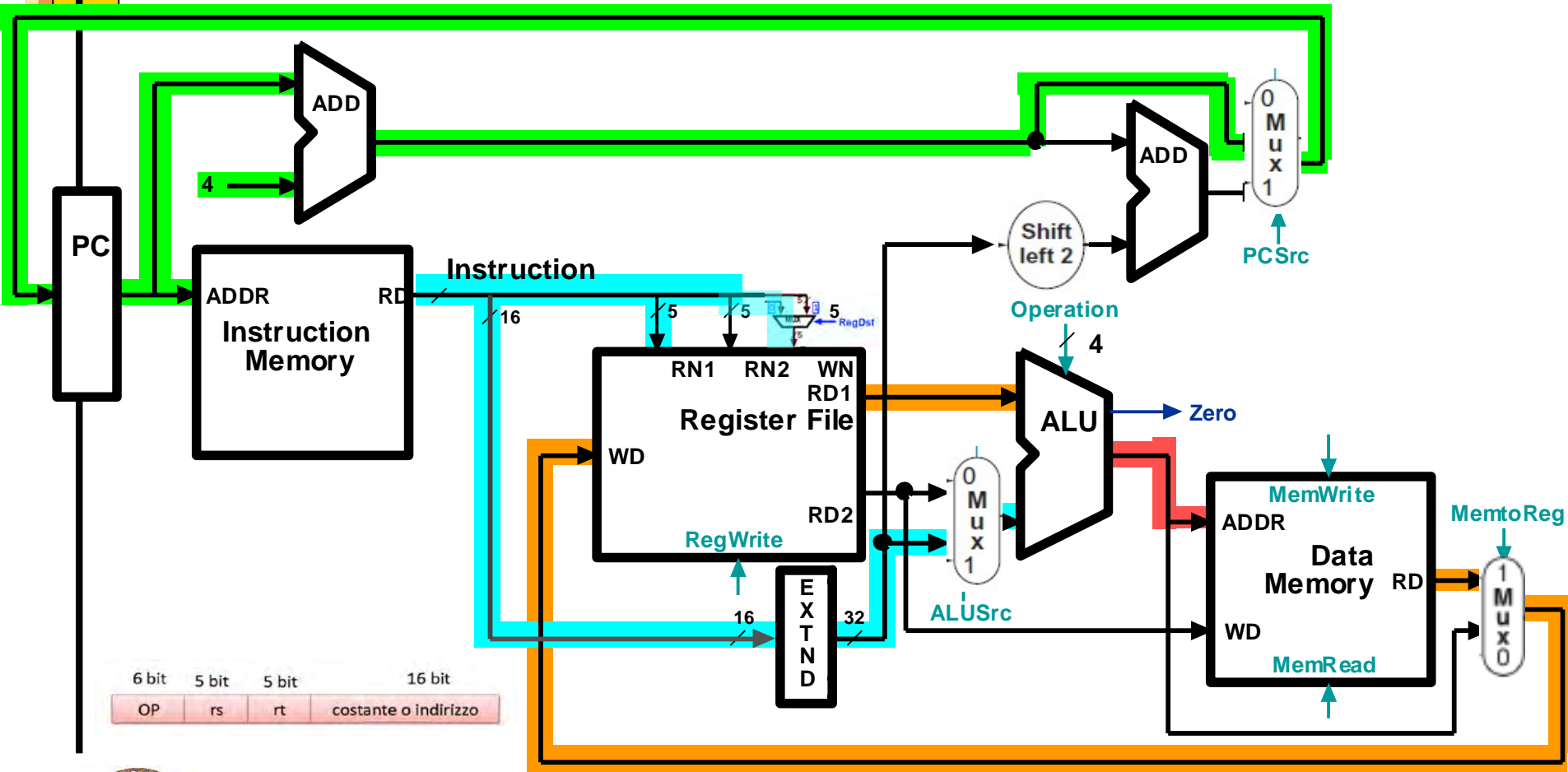
Implementazione a Ciclo Singolo



lw rt,offset(rs)



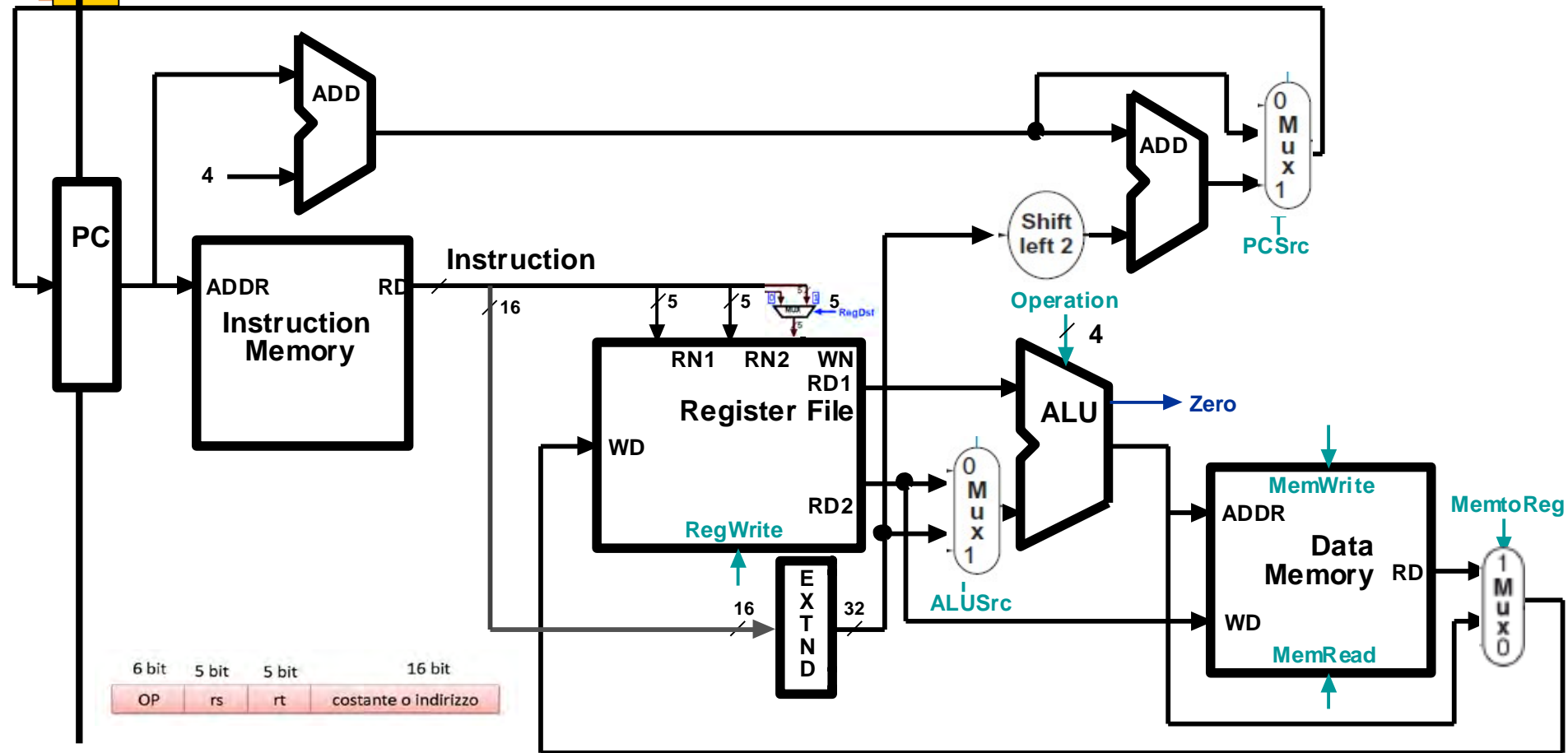
Implementazione a Ciclo Singolo



Datapath dell' istruzione lw

lw rt,offset(rs)

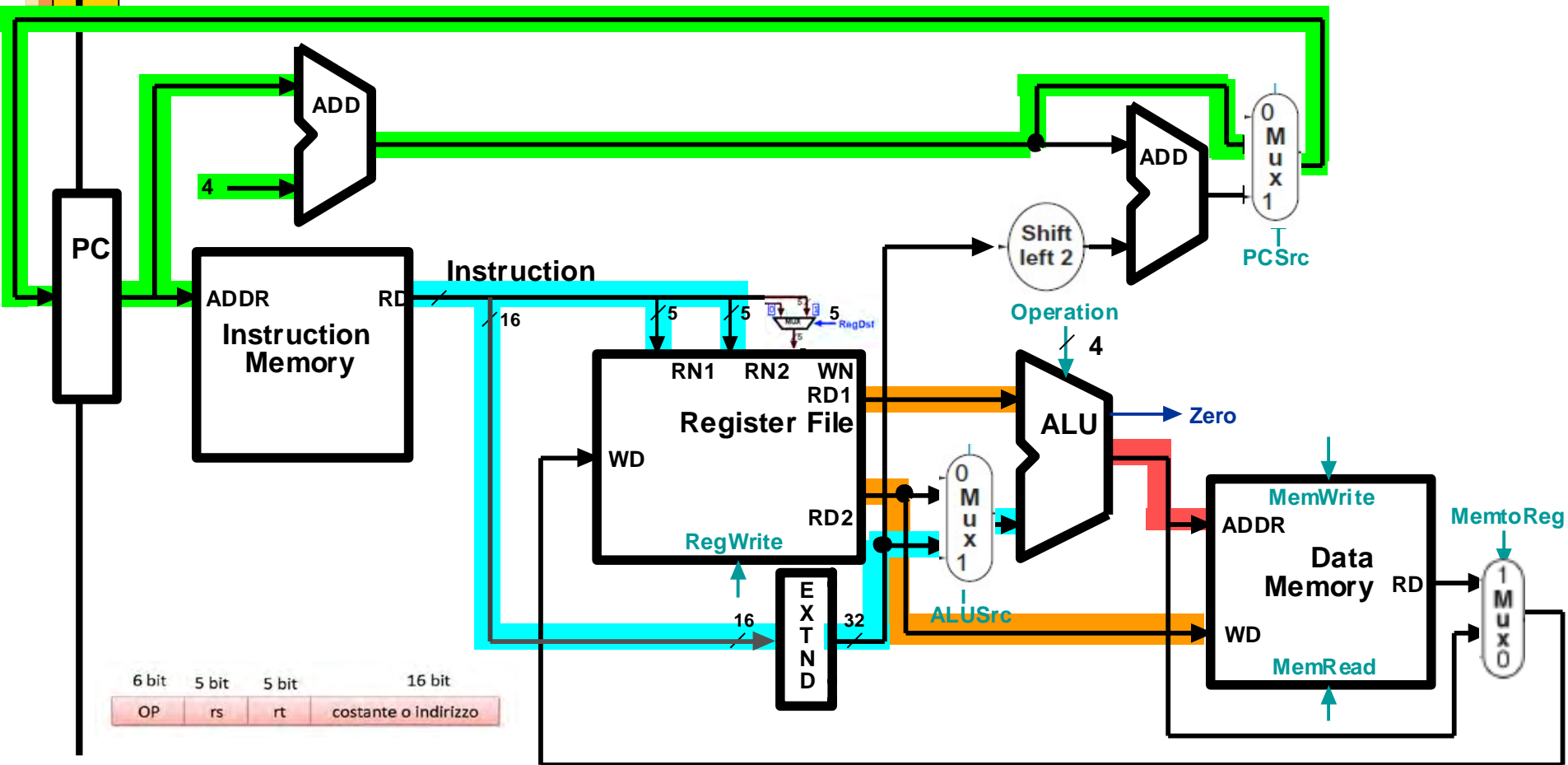
Implementazione a Ciclo Singolo



`sw rt, offset(rs)`



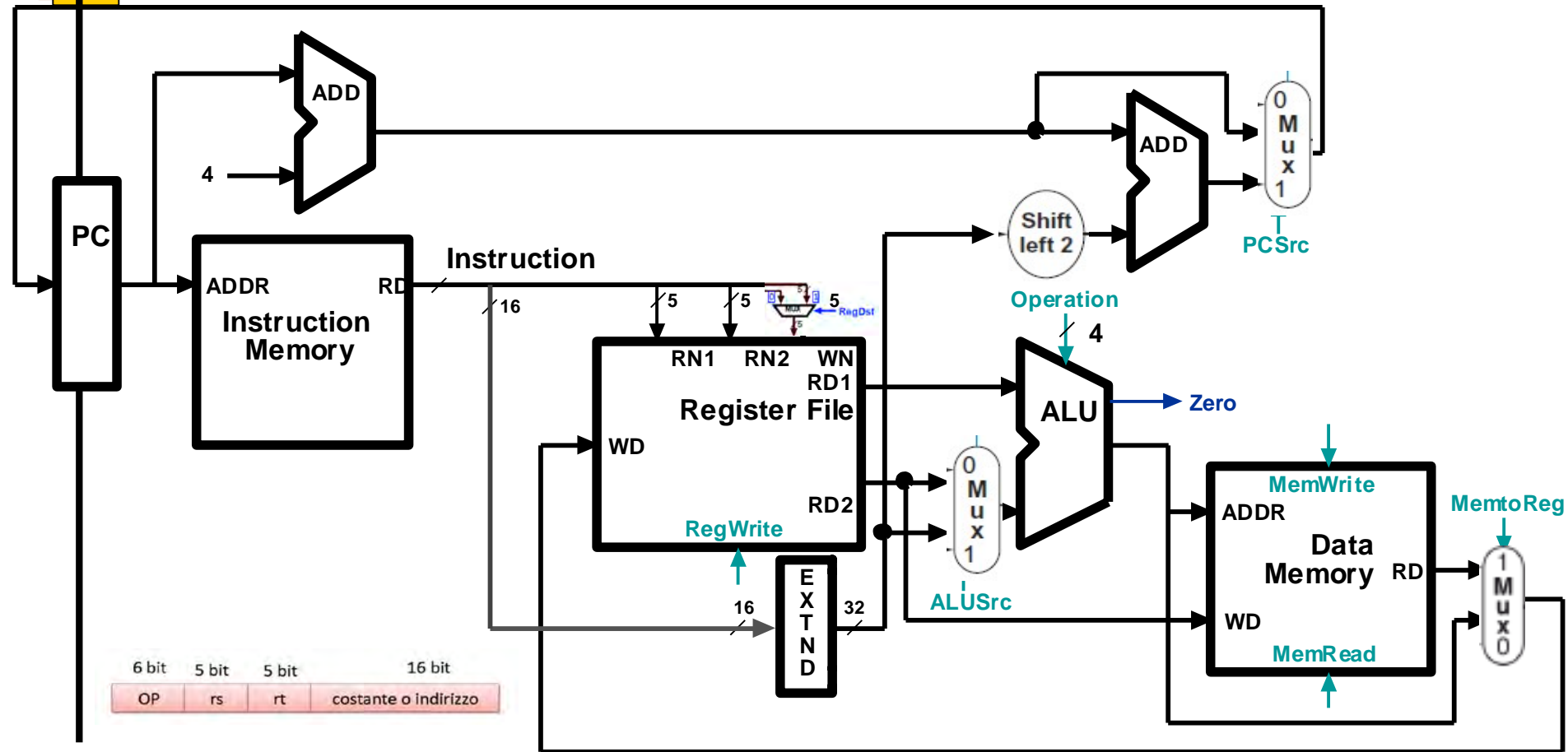
Implementazione a Ciclo Singolo



Datapath dell' istruzione sw

sw rt,offset(rs)

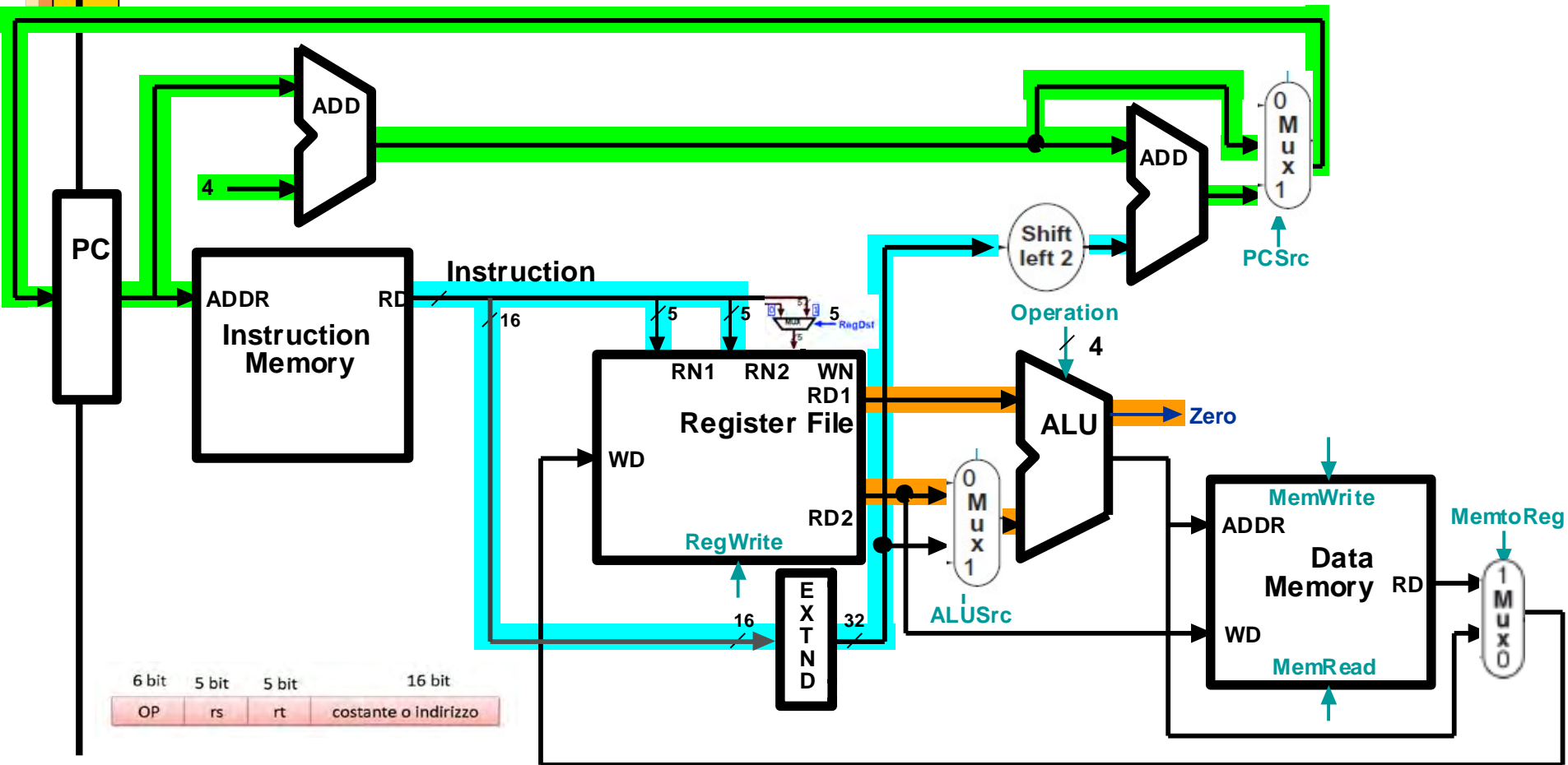
Implementazione a Ciclo Singolo



beq rs,rt,offset



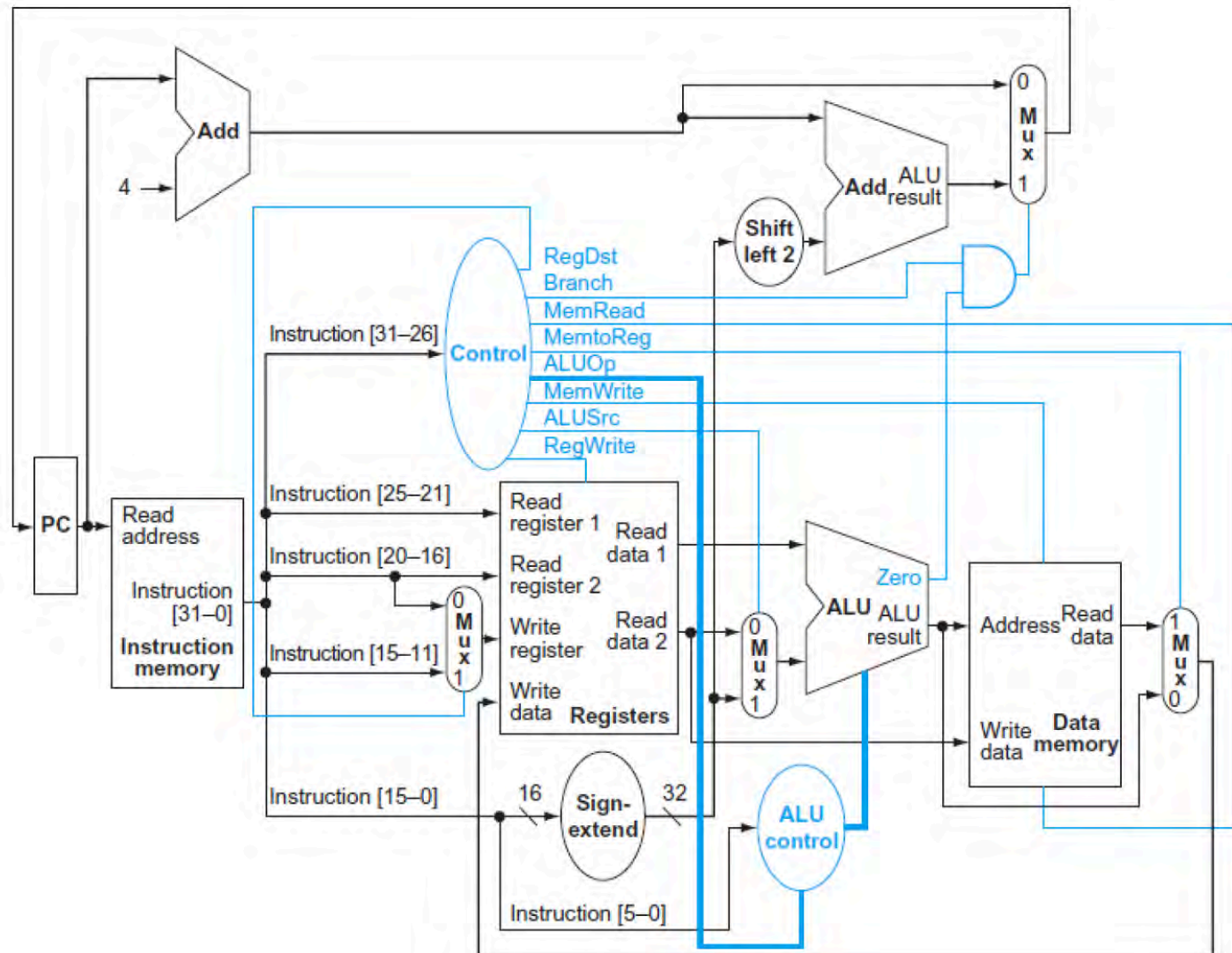
Implementazione a Ciclo Singolo



Datapath dell' istruzione `beq`

`beq rs,rt,offset`

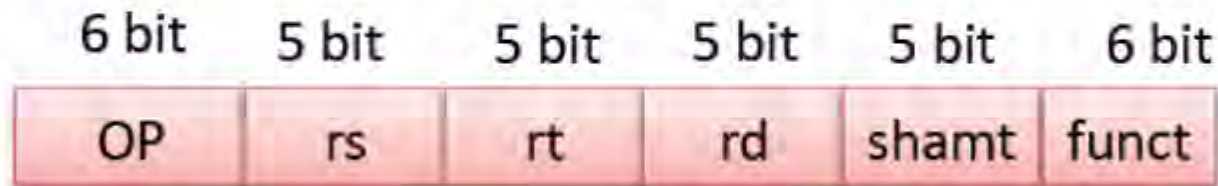
Multiplexer nel MIPS



Multiplexer nel MIPS

E' stato aggiunto anche un **multiplexer** per selezionare il **registro di scrittura** (target)

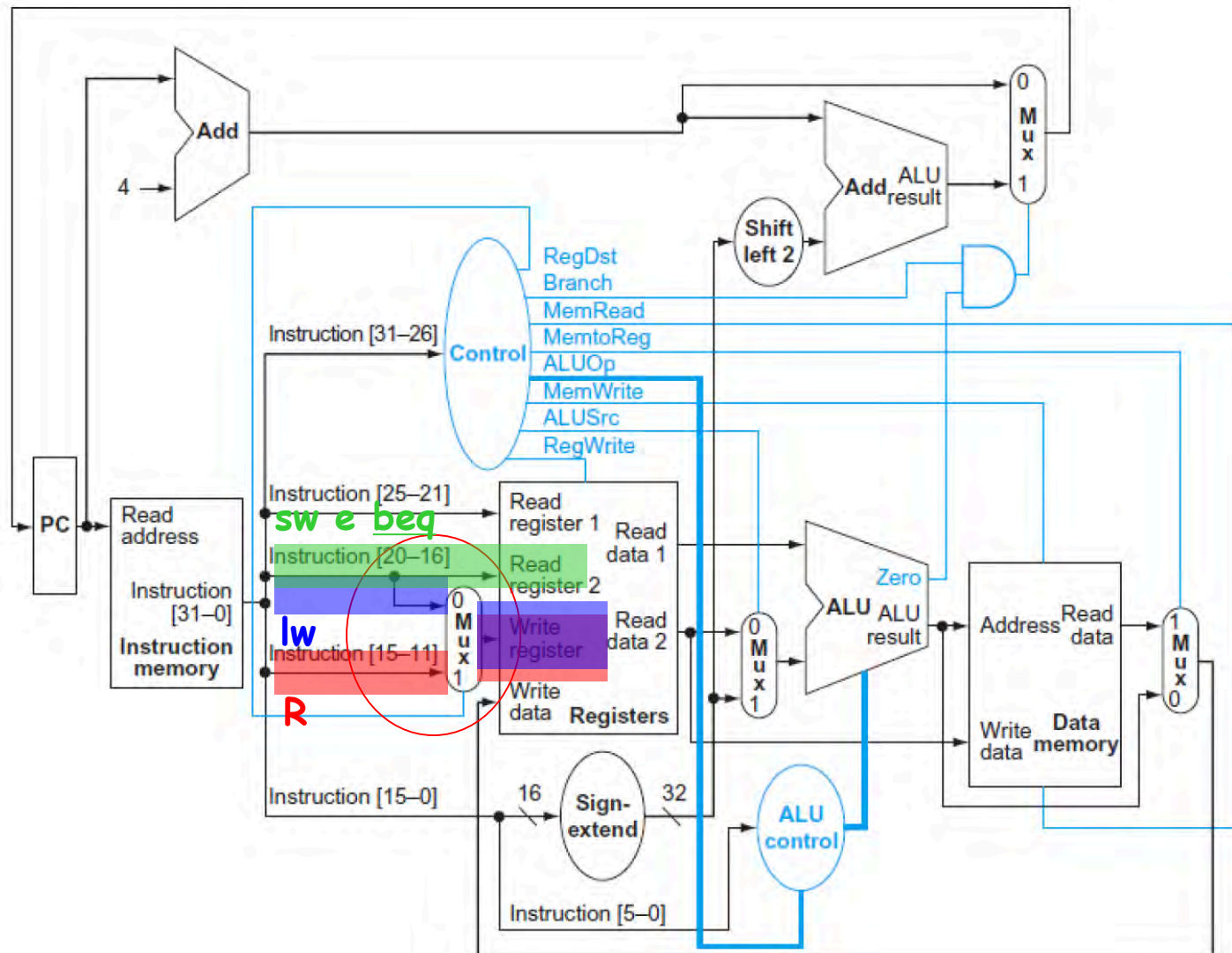
- La necessità sorge dal fatto che il registro target si trova in **posizioni diverse** a seconda del tipo dell'istruzione
 - Posizione dei bit da 15 a 11 per istruzioni di **tipo R**



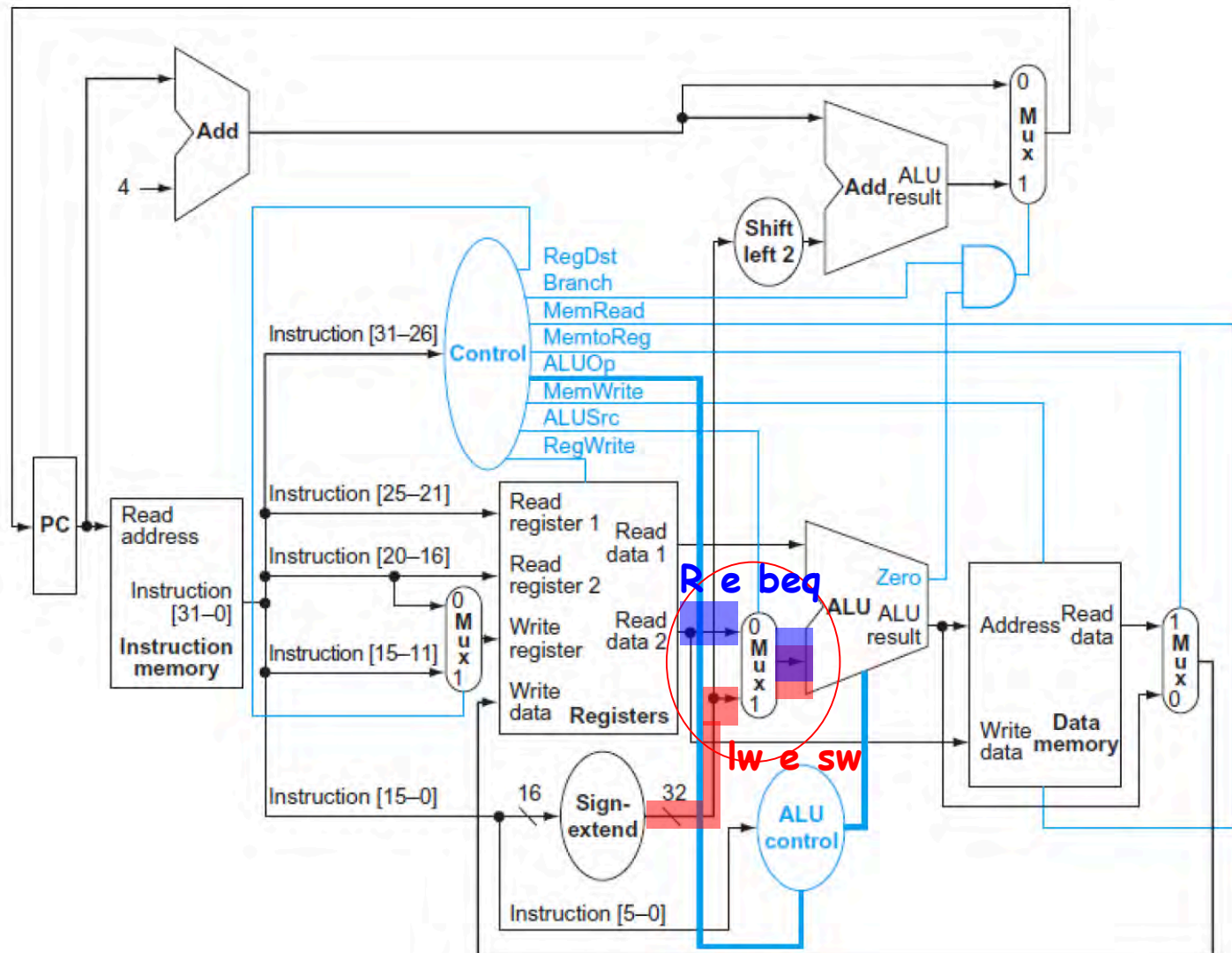
- Posizione dei bit da 20 a 16 per istruzioni **lw, sw** **registro target**



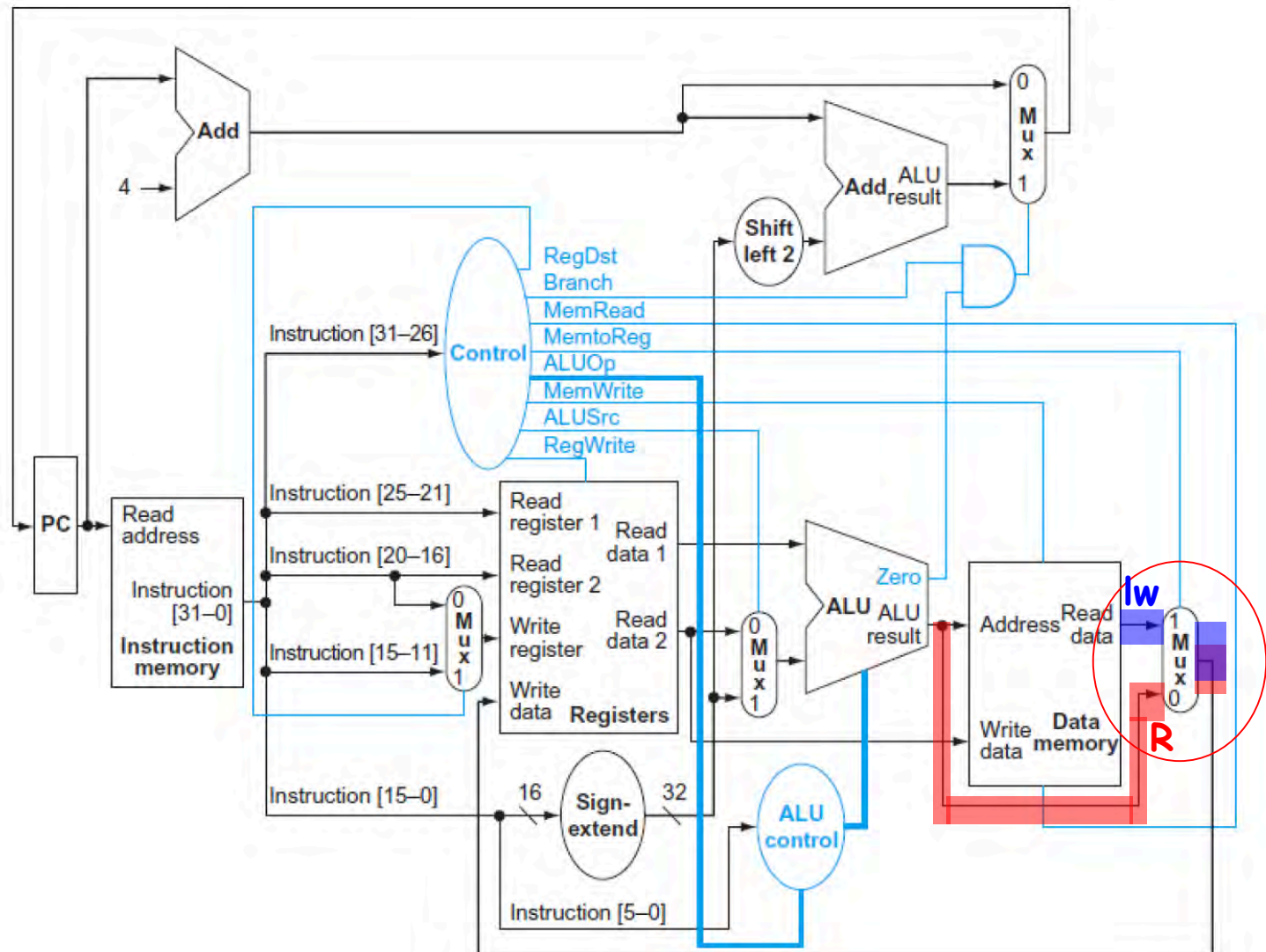
Multiplexer nel MIPS



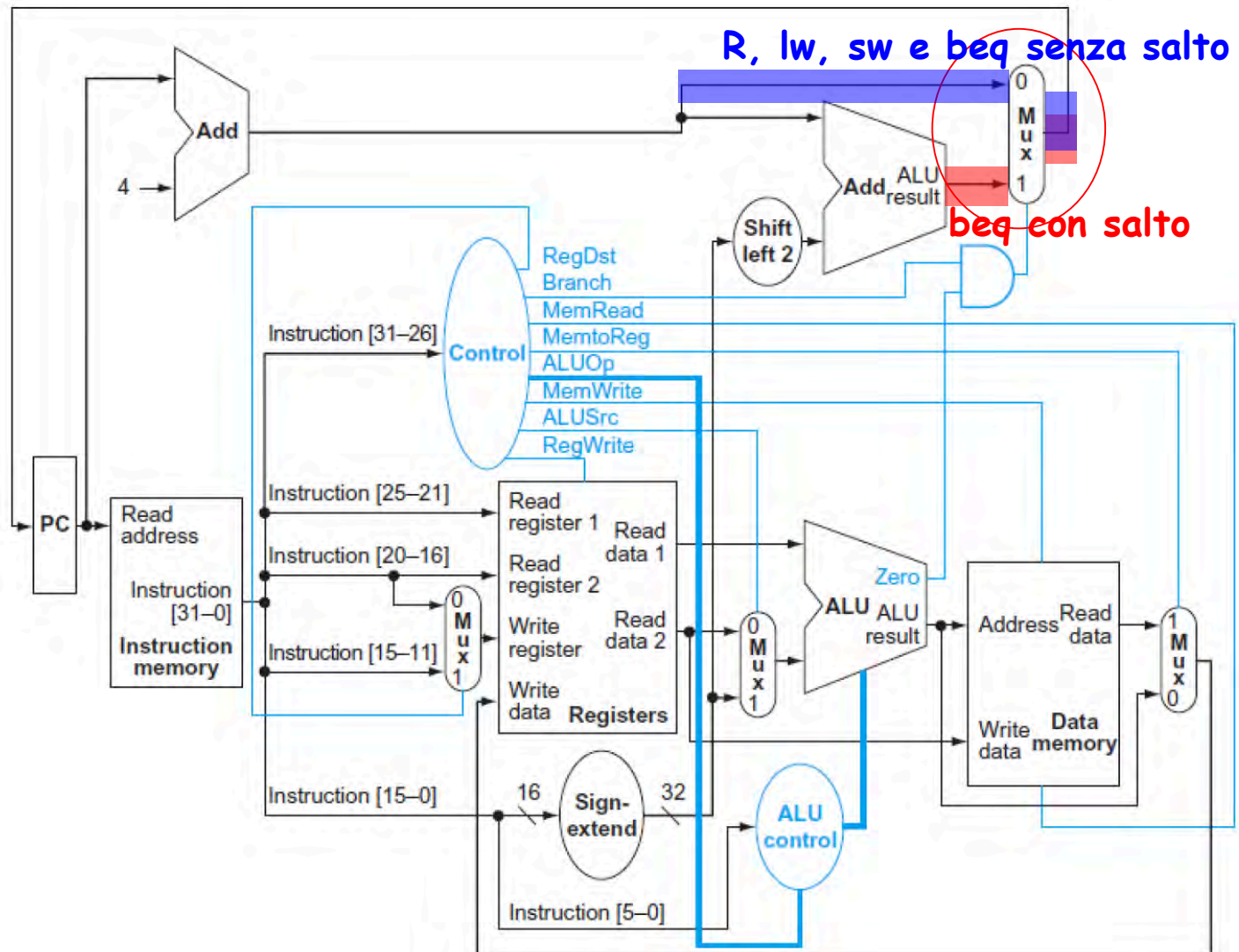
Multiplexer nel MIPS



Multiplexer nel MIPS



Multiplexer nel MIPS



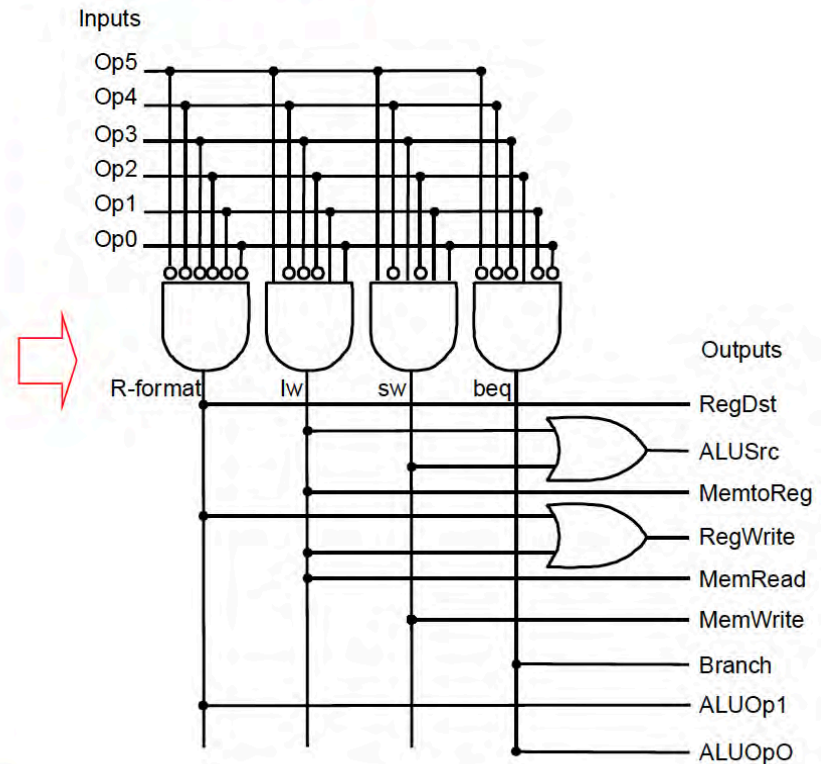
Unità di controllo

Implementazione

Tabella di verità dell'unità di Controllo

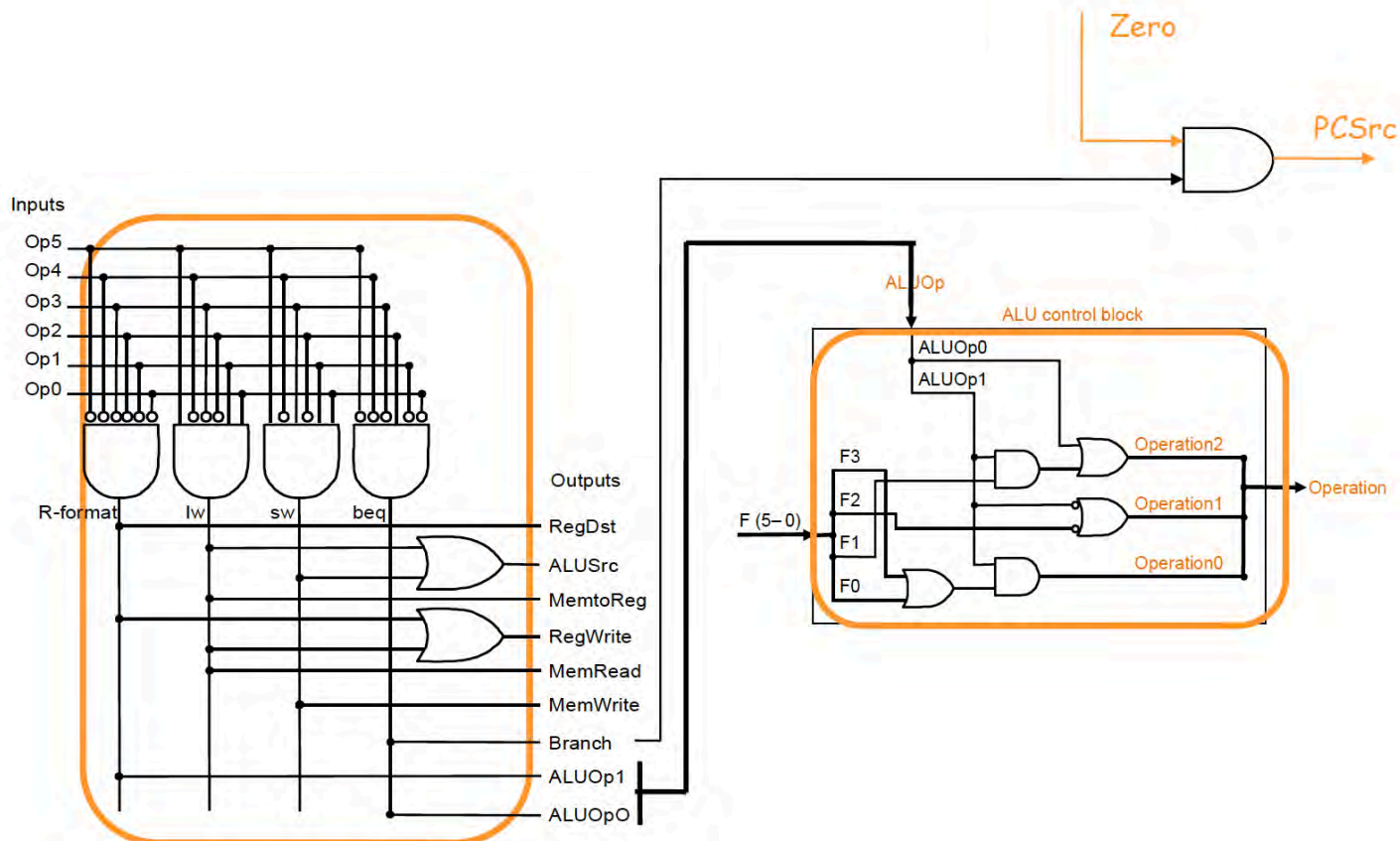
	Segnale	form.			
		R	lw	sw	beq
Inputs = Opcode	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	x	x
	ALUSrc	0	1	1	0
	MemtoReg	0	1	x	x
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Rete Combinatoria realizzabile tramite PLA

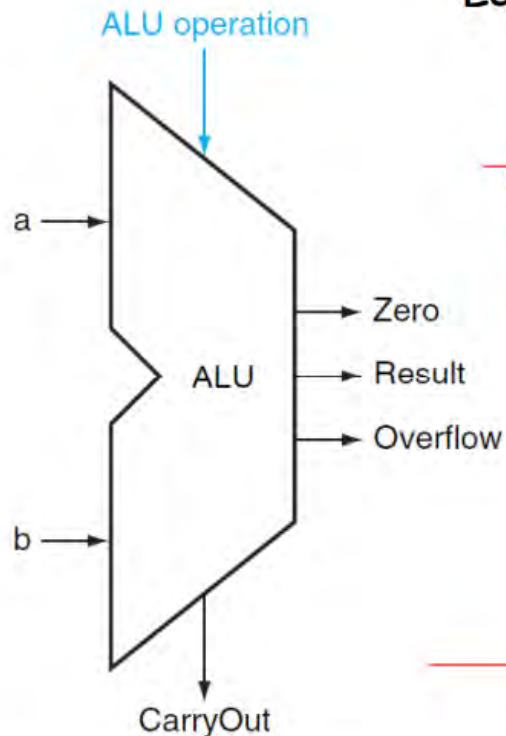


Unità di controllo principale e dell'ALU

Rete combinatoria definita da una coppia di tabelle di verità
Controllo Principale e Controllo ALU



La ALU del MIPS



La ALU prevede le 6 seguenti combinazioni dei suoi **4 segnali di controllo**

~~Ainvert (1 bit), Bnegate (1 bit), Operation (2bit)~~

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

Ignoriamo l'istruzione NOR (e il segnale Ainvert) perché non fa parte del set di istruzioni considerate



Hazard: propagazione e stallo

addi \$s0,\$0,9
addi \$s1,\$0,5
add \$t0,\$s0,\$s1
addi \$s2,\$0,40
sw \$t0,8(\$s2)

IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

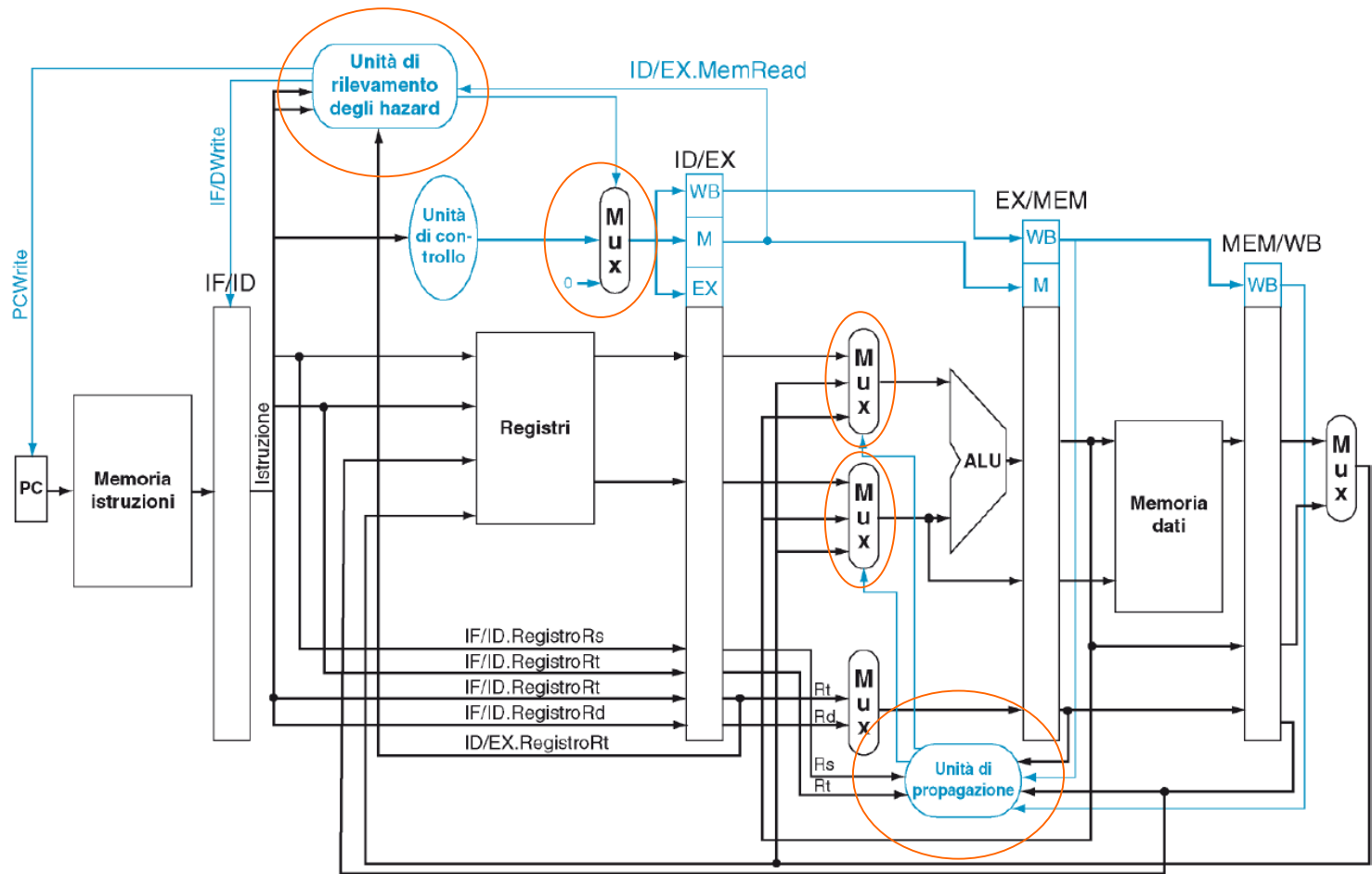
lw \$t0,0(\$s0)
add \$s1,\$s0,\$t0

IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB



MIPS con Pipeline

Con l'aggiunta di Forwarding Unit e Hazard Detection Unit

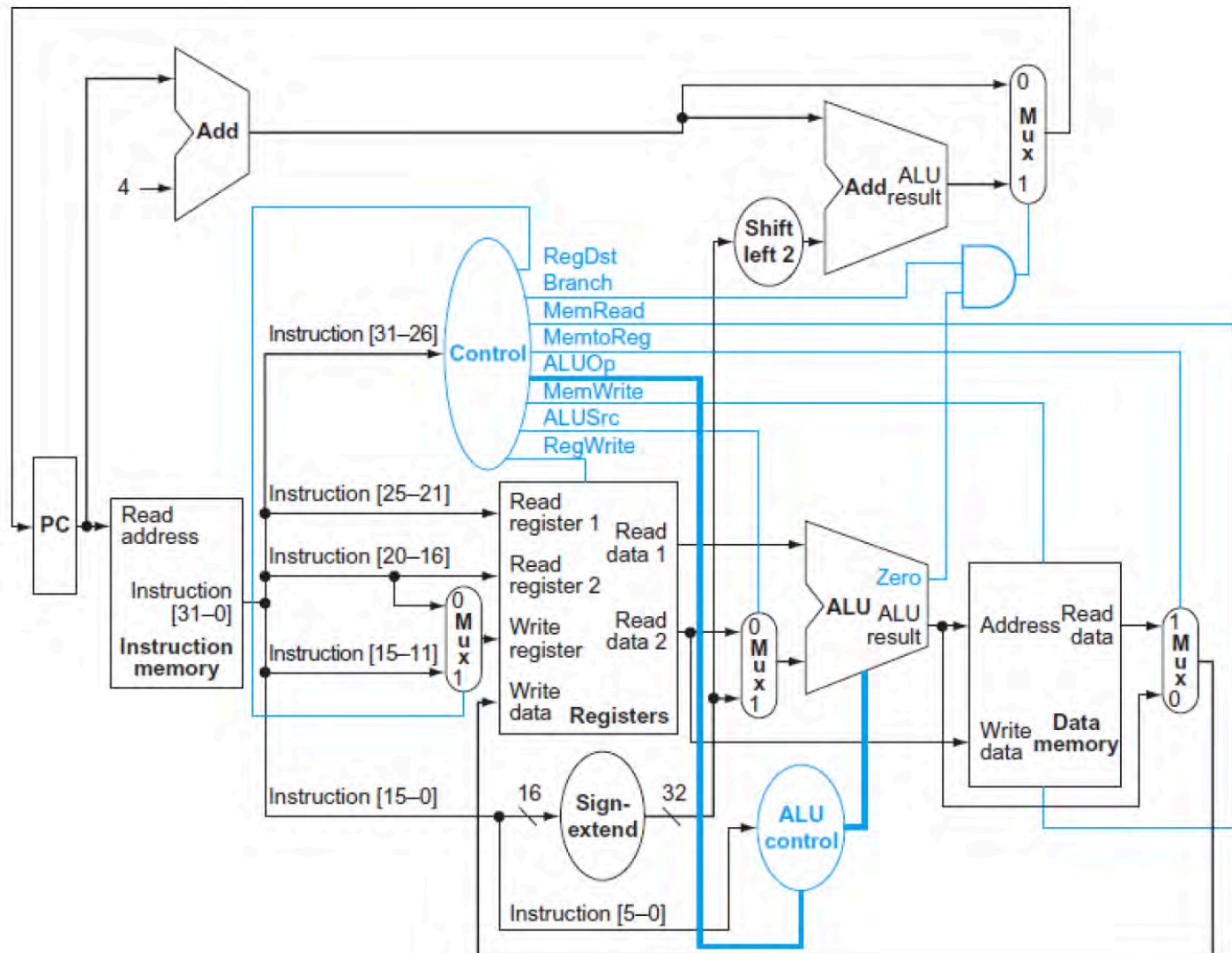


Domanda sul Datapath

- a) Si consideri lo schema implementativo studiato del processore a singolo ciclo MIPS. Si disegni **soltanto** la parte del processore necessaria ad eseguire le istruzioni di formato **R** e l'istruzione **beq**. L'inserimento di blocchi funzionali **non necessari** all'esecuzione delle istruzioni indicate comporterà un aumento dei costi e una riduzione della votazione di questo esercizio. È necessario inserire anche l'**unità di controllo**.
- b) Si consideri il processore disegnato al punto a) mentre esegue l'istruzione **add \$s0, \$s1, \$s2**, dove i registri \$s0, \$s1 e \$s2 contengono rispettivamente i valori 2020, 1 e -7. Indicare i valori di seguito richiesti, ciascuno col **corretto numero di bit** con cui compaiono. Si ricordi che i numeri dei registri \$s0, \$s1 e \$s2 sono rispettivamente 16, 17 e 18 e che il campo `funct` di **add** è 20_{esa}.
- b1) le stringhe binarie in ingresso alle porte di lettura del banco dei registri.
 - b2) le stringhe binarie in ingresso rispettivamente all'unità di controllo principale e all'unità di controllo della ALU.
 - b3) il dato da scrivere nel banco dei registri e il numero del registro in cui scrivere.
- c) Si supponga adesso di voler aggiungere al processore disegnato al punto a) la possibilità di eseguire anche l'istruzione **bne**. Quali modifiche dovranno essere apportate (ai blocchi funzionali dell'unità di elaborazione, all'unità di controllo, ai segnali di controllo, ...)?



Implementazione a Ciclo Singolo



Riepilogo

- Approfondimento Datapath e Hazard
- Prossime lezioni:
 - 19 dicembre 2022 ore 9:00 - 11:30, stesso Team
 - 10 gennaio 2023 ore 16:00 - 19:00, stesso Team

