

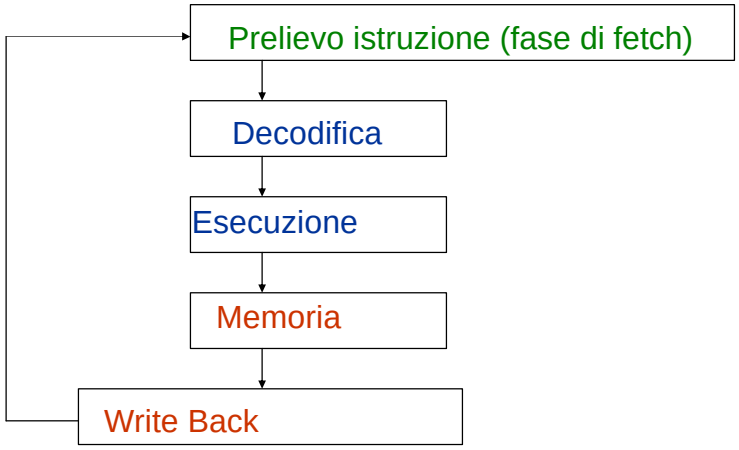

Una CPU multi-ciclo

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it

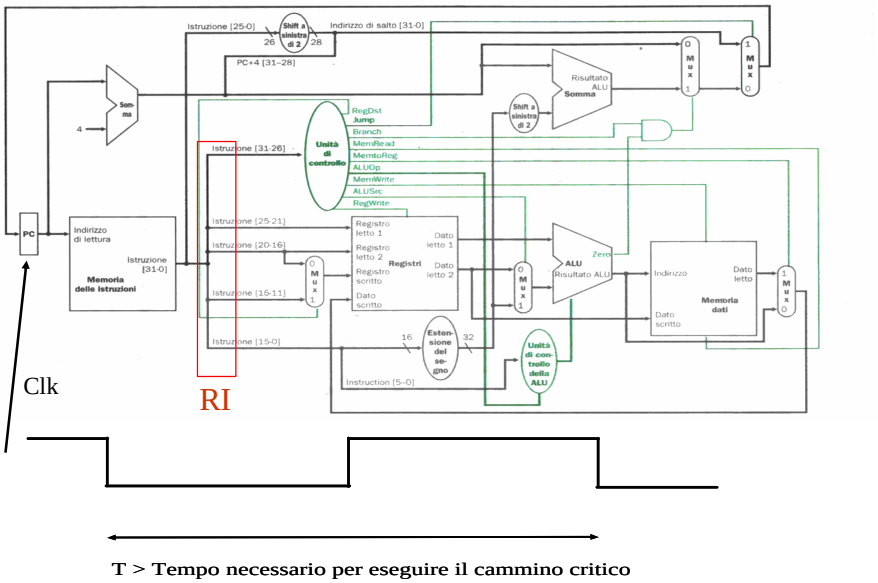
Università degli Studi di Milano


Sommario

Ciclo di esecuzione di un'istruzione



CPU +UC a Ciclo singolo



T > Tempo necessario per eseguire il cammino critico

Problemi

•Duplicazione della Memoria e triplicazione della ALU.
Tuttavia le unità funzionali sono utilizzate in fasi diverse del ciclo di esecuzione di un'istruzione (e.g. Memoria istruzioni in fase di fetch, Memoria dati in fase di Lettura/scrittura).

•Durata uguale per istruzioni che richiedono tempi molto diversi.
Il clock deve essere impostato secondo il cammino critico.

Esecuzione in un singolo ciclo di clock

Assumiamo: memoria (2ns), ALU e sommatori (2ns), lettura/scrittura registri (1ns), decodifica (2ns), nessun ritardo, tempi trascurabili per gli altri elementi della CPU, componenti indipendenti possono lavorare in parallelo.

Istruzione	Memoria istruzioni	Lettura registri Decodifica	Operazione ALU	Memoria dati	Write back	Totale
Tipo R	2	2	2	0	1	7ns
lw	2	2	2	2	1	9ns
sw	2	2	2	2	0	8ns
beq	2	2	2	0	0	6ns
j	2	2	0	0	0	4ns

•La durata del ciclo di clock deve essere pari al percorso più lungo (cammino critico).
Percorso più lungo dovuto ad istruzione di caricamento (lw)

Valutazione della prestazione della CPU a singolo ciclo

Dipende dal programma.

	lw	sw	beq	j	R	fp (add)	fp (mul)	Durata Clock (max)	Durata media
Durata	9ns	8ns	6ns	4ns	7ns	12ns	20ns		
Caso I	24%	12%	18%	2%	44%			9ns	7.36ns
Caso II	31%	21%	5%	2%	27%	7%	7%	20ns	8.98ns

In ogni caso, un'implementazione a clock singolo porta ad uno spreco di tempo notevole.

Come gestire istruzioni di durata diversa?

Quando è efficiente l'implementazione a singolo ciclo?

Clock di durata variabile è una soluzione?

Non viene risolto il problema della duplicazione delle unità funzionali.

Sommario

I problemi della UC a singolo ciclo di clock

Principi ispiratori di una CPU multi-ciclo. Le fasi di fetch e decodifica.

Esecuzione multi-ciclo delle istruzioni R

Esecuzione multi-ciclo delle istruzioni lw/sw.

Esecuzione multi-ciclo delle istruzioni di salto.

Analisi della CPU multi-ciclo.

Caratteristiche CPU multi-ciclo

Spezza l'istruzione in più passi, dove ciascun passo impiega lo stesso tempo.

Il clock non sincronizza più l'intera istruzione ma solamente il singolo passo.

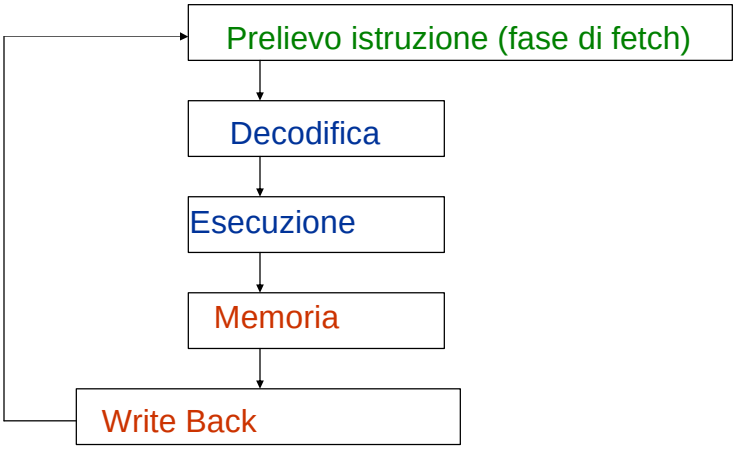
Le istruzioni possono essere eseguite in un numero diverso di cicli di clock.

Consente di riutilizzare le unità funzionali (in cicli di clock diversi).

Richiede l'aggiunta di HW addizionali (registri di memoria temporanea). Questi devono memorizzare lo stato delle unità funzionali, cioè l'informazione che può servire ai passi successivi e che rischia di essere sovrascritta dal riuso dell'unità funzionale.

L'unità di controllo diventa una FSM.

Ciclo di esecuzione di un'istruzione



Le istruzioni richiederanno da 3 a 5 cicli di clock

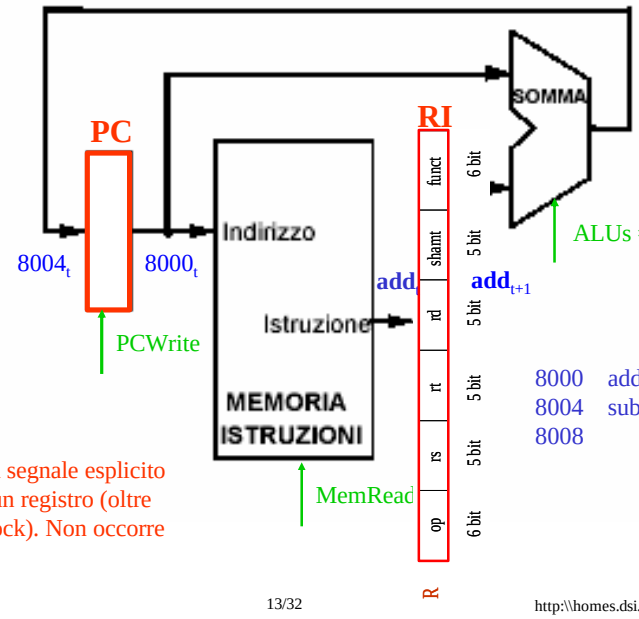
Valutazione della prestazione della CPU multi-ciclo

Dipende dal programma.

	lw	sw	beq	j	R	fp (add)	fp (mul)	Durata Clock (max singolo ciclo)	Durata media
Durata	10ns	8ns	6ns	4ns	8ns	12ns	20ns		
Caso I	24%	12%	18%	2%	44%			9ns	8.0ns
Caso II	31%	21%	5%	2%	27%	7%	7%	20ns	9.56ns

A questo confronto va aggiunto che la CPU multi-ciclo consente ancora dei risparmi ulteriori nel tempo di esecuzione.

Circuito della fase di fetch multi-ciclo

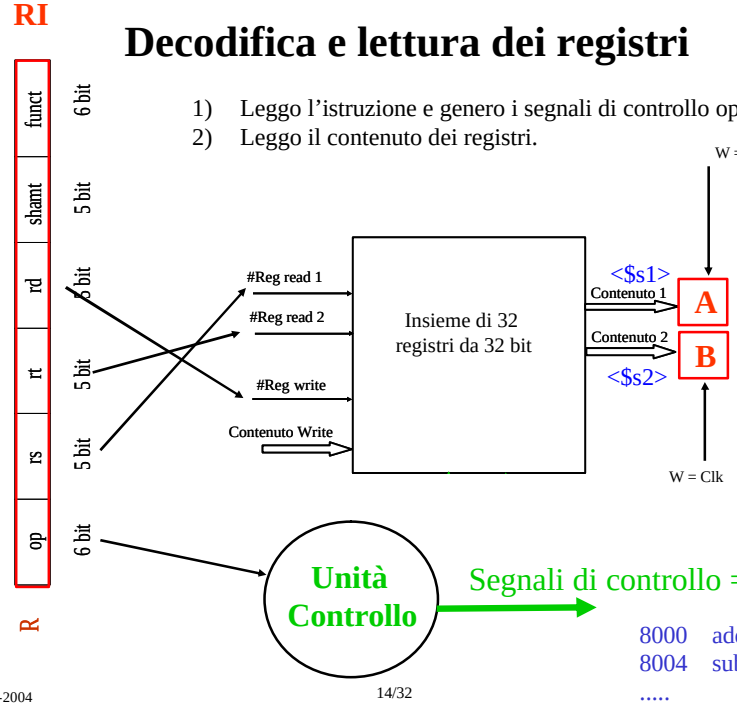


8000 add $s1, $s2, $s3
8004 sub $s4, $s1, $t1
8008

NB: Occorre un segnale esplicito per scrivere in un registro (oltre al segnale di clock). Non occorre per la lettura.

Decodifica e lettura dei registri

1) Leggo l'istruzione e genero i segnali di controllo opportuni.
2) Leggo il contenuto dei registri.



8000 add $s1, $s2, $s3
8004 sub $s4, $s1, $t1
.....

Sommario

I problemi della UC a singolo ciclo di clock

Principi ispiratori di una CPU multi-ciclo. Le fasi di fetch e decodifica.

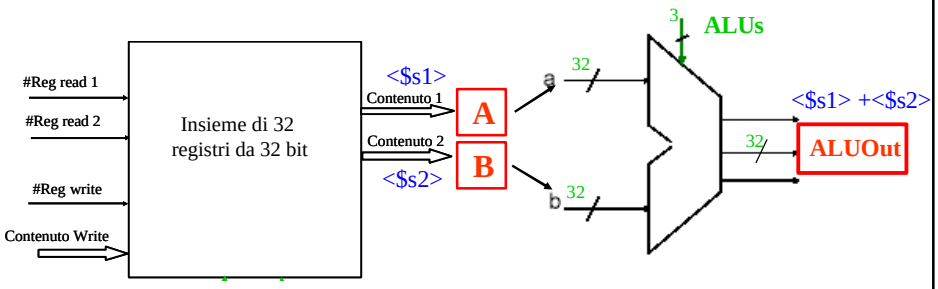
Esecuzione multi-ciclo delle istruzioni R

Esecuzione multi-ciclo delle istruzioni lw/sw.

Esecuzione multi-ciclo delle istruzioni di salto.

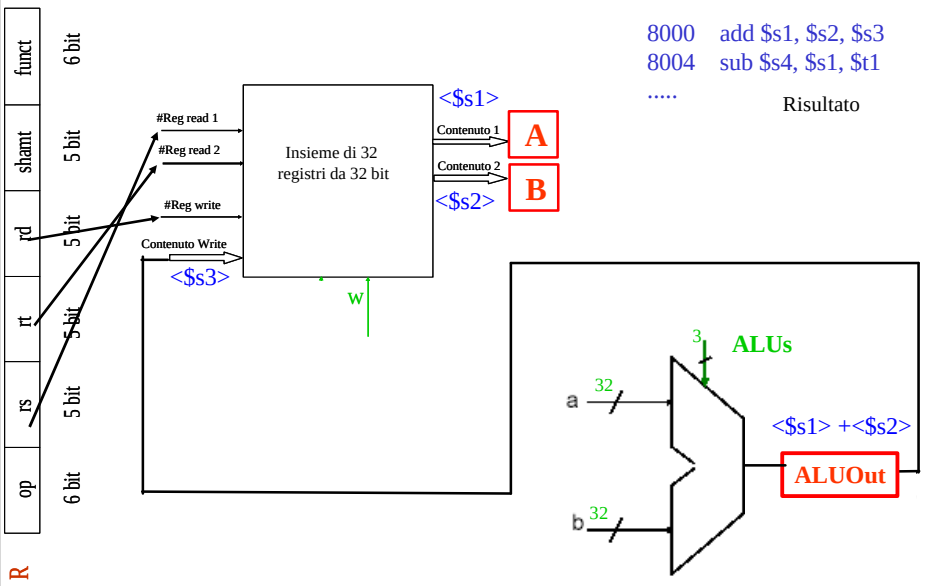
Analisi della CPU multi-ciclo.

Fase di esecuzione (tipo R)



8000 add $s1, $s2, $s3
8004 sub $s4, $s1, $t1
.....

Fase di Write back (tipo R)



Sommario

I problemi della UC a singolo ciclo di clock

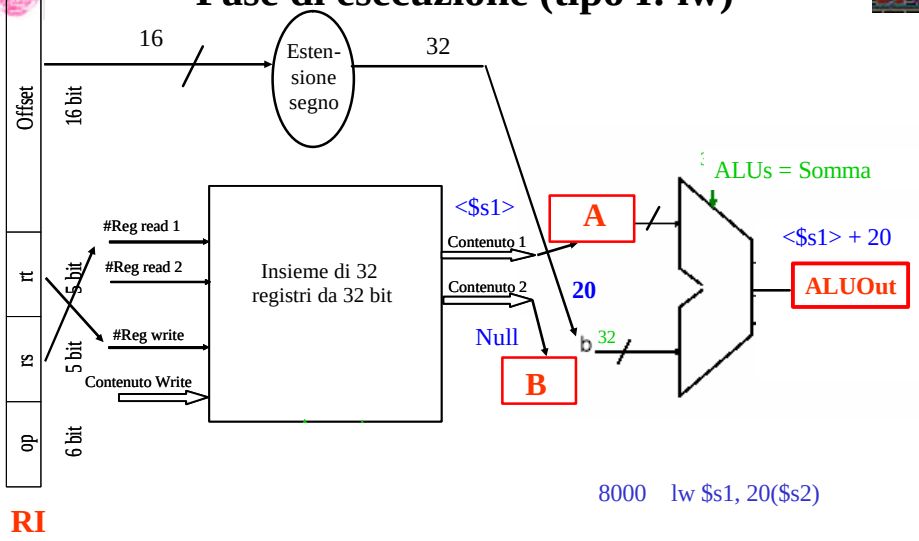
Principi ispiratori di una CPU multi-ciclo. Le fasi di fetch e decodifica.

Esecuzione multi-ciclo delle istruzioni R

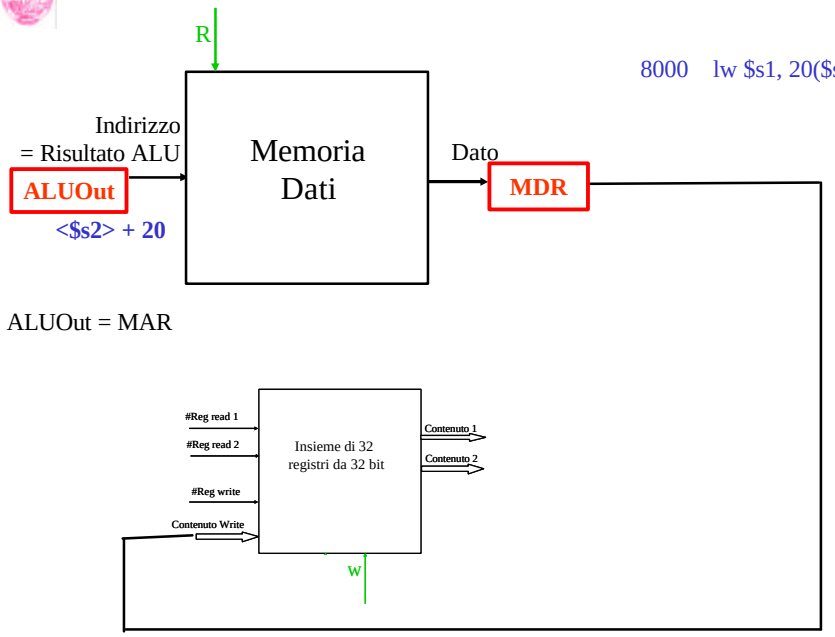
Esecuzione multi-ciclo delle istruzioni lw/sw.

Esecuzione multi-ciclo dei salti ed analisi della CPU multi-ciclo.

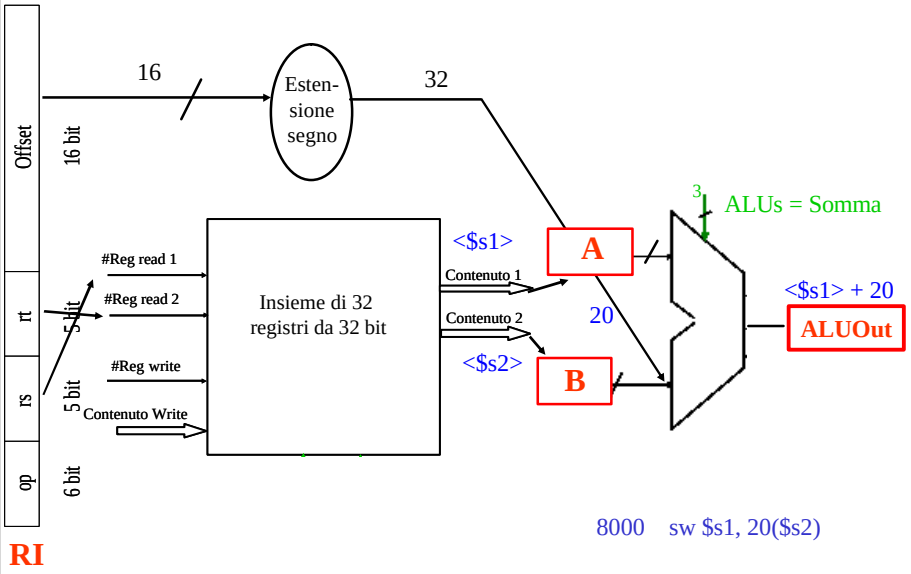
Fase di esecuzione (tipo I: lw)



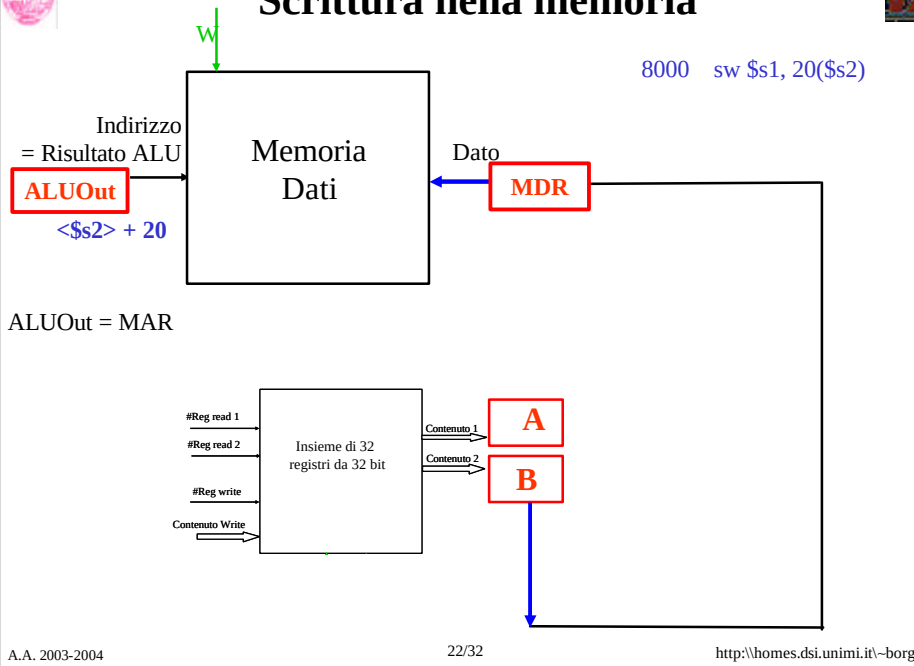
Lettura della memoria + write-back



Fase di esecuzione (tipo I: sw)



Scrittura nella memoria



Sommario

I problemi della UC a singolo ciclo di clock

Principi ispiratori di una CPU multi-ciclo. Le fasi di fetch e decodifica.

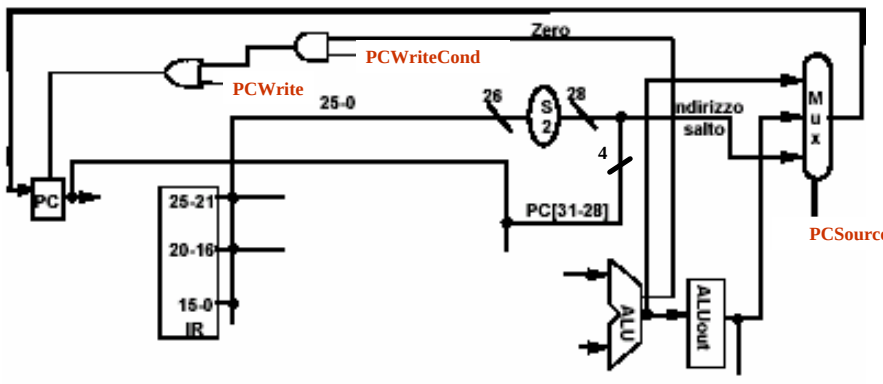
Esecuzione multi-ciclo delle istruzioni R

Esecuzione multi-ciclo delle istruzioni lw/sw.

Esecuzione multi-ciclo delle istruzioni di salto.

Analisi della CPU multi-ciclo.

CPU multi-ciclo: i salti



Mux sceglie tramite PCSource tra:
•Indirizzo ottenuto sommando 4.
•Indirizzo ottenuto dall'ALU (beq).
•Indirizzo ottenuto dal campo offset dell'IR (jump).

PCWrite, e PC Source sono coordinati
PCWriteCond e PCSource sono coordinati.

Sommario

I problemi della UC a singolo ciclo di clock

Principi ispiratori di una CPU multi-ciclo. Le fasi di fetch e decodifica.

Esecuzione multi-ciclo delle istruzioni R

Esecuzione multi-ciclo delle istruzioni lw/sw.

Esecuzione multi-ciclo delle istruzioni di salto.

Analisi della CPU multi-ciclo.

Confronto delle 2 CPU

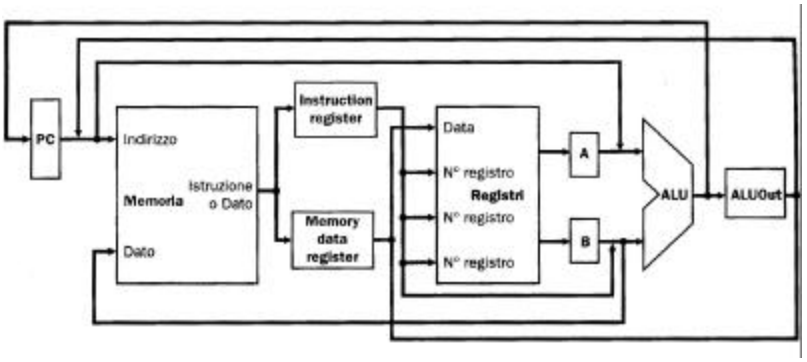
Due Memorie. Si possono compattare a patto di creare 2 registri. 1 registro istruzione (RI) ed un registro dati (MDR).

3 ALU. Si può utilizzare una unica ALU se viene utilizzata in 3 fasi diverse.

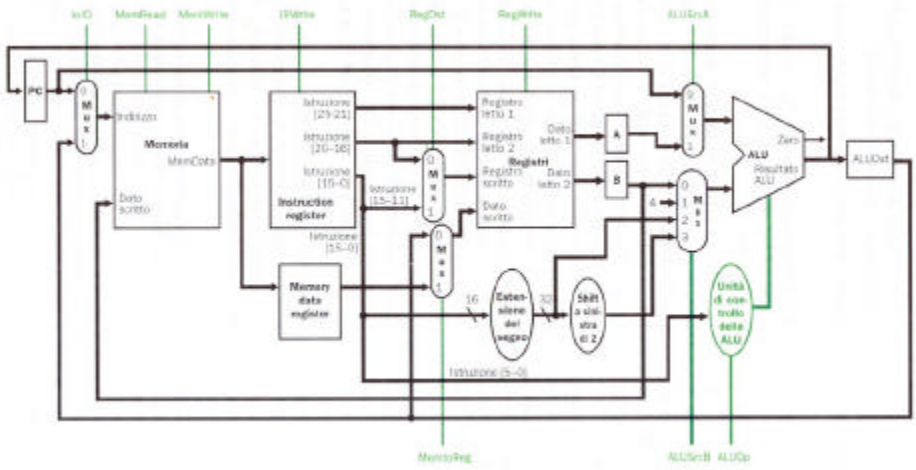
Fase 1: + 4.

Fase 2 Offset + rs.

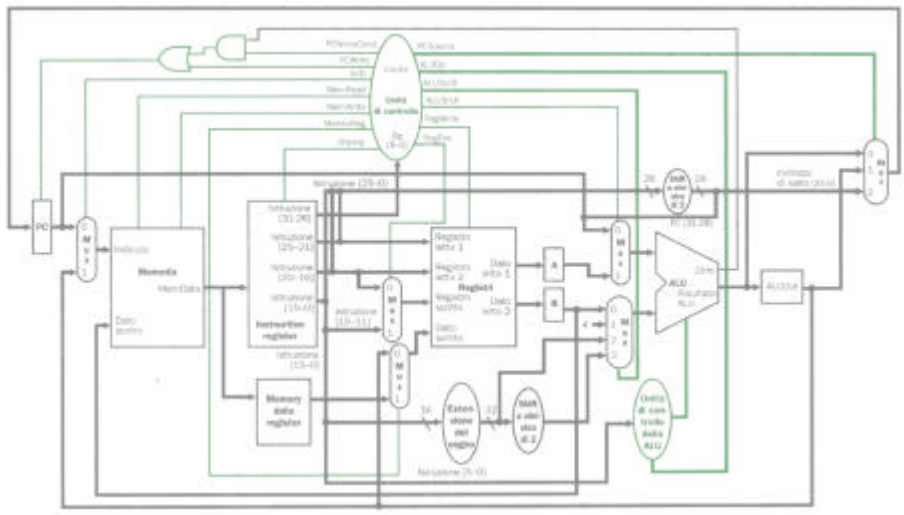
Fase 3. Operazione di tipo R.



CPU multi-ciclo per R, lw/sw, beq



CPU multi-ciclo



Analisi dell'utilizzo dei registri

PC – Sincronizza il ciclo di esecuzione di un'istruzione. Viene aggiornato in fase di fetch (+4). In questo caso contiene l'indirizzo dell'istruzione successiva. Può essere aggiornato anche in fase di decodifica (jump) oppure in fase di esecuzione (beq).
RI – Viene aggiornato alla fine della FF. Mantiene per tutte le fasi, le informazioni sull'istruzione: numero di registri, offset, funct, shmt, opcode. Questa informazione viene utilizzata da:

Istruzioni di tipo beq, jump in fase di Decodifica.
Istruzioni di tipo R e lw, in fase di Decodifica e WriteBack.
Istruzioni di tipo sw, in fase di Decodifica.

A e B – Vengono aggiornati sempre. Il loro valore cambia solo alla fine della DECOD (nelle altre fasi l'uscita del register file non cambia). Mantiene il contenuto dei registri rs e rt. Il contenuto del 2o registro può non essere utilizzato (lw).
ALUOut – Viene aggiornato sempre. Nei primi 3 stadi assume rispettivamente il valore di:

FF (contiene PC + 4).
DECOD (Contiene PC + Offset * 4).
EXEC (Contiene Reg Read 1 + (Reg Read 2 *oppure* Offset * 4).

MDR – Viene aggiornato solo nelle istruzioni lw/sw.
Register File – Viene aggiornato solo nella fase di WriteBack nelle istruzioni R e lw.

Scrittura nei registri della CPU multi-ciclo

Tutti i registri della CPU vengono scritti ad ogni colpo di clock. Non c'è bisogno di sintetizzare nella UC il segnale di scrittura dei registri.

TRANNE CHE PER l'IR ed il PC.

Tutti i registri possono essere riscritti ad ogni passo dell'istruzione, tranne l'IR che deve mantenere il suo valore per tutta la durata dell'esecuzione di un'istruzione ed il PC che viene aggiornato in fase di fetch.

Segnali di controllo

Nome del segnale	Valore	Effetto
ALUSrcA (1 bit)	0	Il primo operando è il valore attuale del PC
	1	Il primo operando proviene dalla prima porta di lettura del Register File
ALUSrcB (2 bit)	00	Il secondo operando proviene dalla seconda porta di lettura del RF
	01	Il secondo operando è la costante + 4
	10	Il secondo operando è l'estensione del segno del campo offset
	11	Il secondo operando proviene dall'estensione del segno e dallo shift a sx di due posizioni, dell'offset
IorD	0	L'indirizzo della memoria proviene dal PC
	1	L'indirizzo della memoria proviene dalla ALU (ALUOut)
PCSource	00	In PC viene scritta l'uscita della ALU (PC+4)
	01	In PC viene scritta il contenuto di ALUOut (indirizzo di una branch)
	10	In PC viene scrittto l'indirizzo di destinazione della jump
PCWrite	0	Nessuno
	1	Viene scritto il registro PC. L'indirizzo scritto in PC è controllato da PCSource
PCWriteCond	0	Nessuno
	1	Il PC viene scritto se anche l'uscita Zero della ALU è affermata

Sommario

I problemi della UC a singolo ciclo di clock

Principi ispiratori di una CPU multi-ciclo. Le fasi di fetch e decodifica.

Esecuzione multi-ciclo delle istruzioni R

Esecuzione multi-ciclo delle istruzioni lw/sw.

Esecuzione multi-ciclo delle istruzioni di salto.

Analisi della CPU multi-ciclo.