

Esecuzione di istruzioni (S. 5.2)

- Le categorie di istruzioni rappresentative per i processori RISC sono
 - Caricamento (Load)
 - Aritmetiche e Logiche (Add, And, Or, etc.)
 - Immagazzinamento (Store)
- Per tutte si usa uno schema di operazioni a **cinque** passi (o azioni), ciascun passo si effettua con uno stadio hardware. Per le **istruzioni di caricamento**, es. **Load R5, X(R7)**, indirizzamento con indice e spiazzamento, si ha
 1. Prelievo dell'istruzione dalla memoria, e incremento del **PC**
 2. Decodifica dell'istruzione, e lettura del contenuto di **R7**
 3. Calcolo dell'indirizzo effettivo **X+ [R7]**
 4. Lettura dell'operando dalla memoria all'indirizzo effettivo
 5. Scrittura dell'operando nel registro **R5**

Prof. Tramontana

7

Istruzioni aritmetiche e logiche

- In questa categoria di istruzioni non si accede a operandi in memoria
- Una tipica operazione è **Add R3, R4, R5**, essa richiede i seguenti passi (schema a cinque passi con un passo privo di azioni)
 1. Prelievo istruzione da memoria, e incremento **PC**
 2. Decodifica dell'istruzione, e lettura di **R4** e **R5**
 3. Calcolo somma **[R4]+[R5]**
 4. Nessuna azione
 5. Scrittura del risultato nel registro **R3**
- Nel caso della variante **Add R3, R4, #1000**, il passo 2 legge solo **R4** e il passo 3 calcola la somma **[R4] + 1000**

Passi per **Load R5, X(R7)**

1. Prelievo istruz e increm **PC**
2. Decodifica istruz e lettura **R7**
3. Calcolo indirizzo **X+ [R7]**
4. Lettura da memoria
5. Scrittura nel registro **R5**

Prof. Tramontana

8

Istruzioni di immagazzinamento

- L'istruzione **Store R6, X(R8)** scrive il contenuto di **R6** nella locazione di memoria **X+ [R8]**, è realizzata in cinque passi
 1. Prelievo istruzione da memoria, e incremento **PC**
 2. Decodifica dell'istruzione, e lettura di **R6** e **R8**
 3. Calcolo indirizzo effettivo **X+ [R8]**
 4. Immagazzinamento di **R6** in memoria all'indirizzo effettivo
 5. Nessuna azione

Prof. Tramontana

9

Schema di esecuzione di istruzioni

- Si può notare uno schema generale, che vale anche per altre istruzioni e per altri modi di indirizzamento (es. ponendo X a 0 quando si ha il modo indiretto)

Load R5, X(R7)	Add R3, R4, R5	Store R6, X(R8)
1. Prelievo istruz e increm PC	1. Prelievo istruz e increm PC	1. Prelievo istruz e increm PC
2. Decod istruz e lettura R7	2. Decod istruz e lettura R4 e R5	2. Decod istruz e lettura R6 e R8
3. Calcolo indirizzo X+ [R7]	3. Calcolo somma [R4]+[R5]	3. Calcolo indirizzo X+ [R8]
4. Lettura da memoria	4. Nessuna azione	4. Scrittura memoria
5. Scrittura nel registro R5	5. Scrittura risultato in R3	5. Nessuna azione

Passo Azione

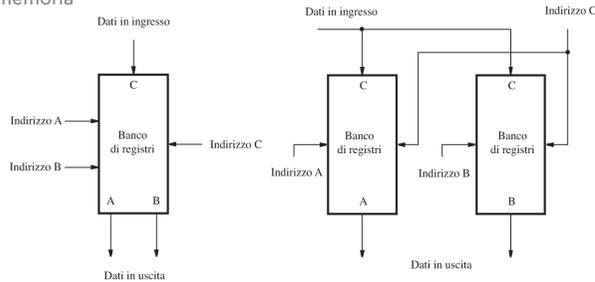
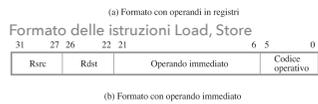
- 1 Preleva un'istruzione e incrementa il contatore del programma
- 2 Decodifica l'istruzione e leggi registri dal banco di registri
- 3 Esegui un'operazione dell'ALU
- 4 Leggi o scrivi dati in memoria se l'istruzione coinvolge un operando in memoria
- 5 Scrivi il risultato nel registro di destinazione, se necessario

Prof. Tramontana

10

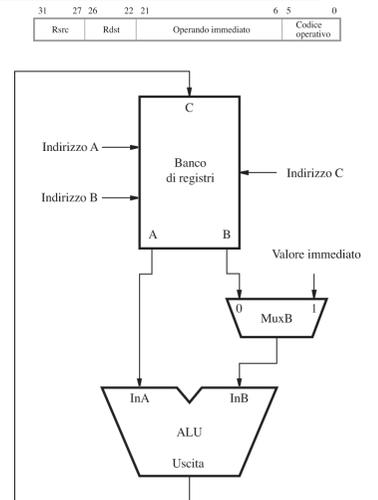
Hardware (S. 5.3): Banco di registri

- ▶ Il banco dei registri è un'unità di memoria **interna al processore**, dotata di circuiti di accesso per selezione i registri e l'operazione di lettura o scrittura. La selezione dei registri avviene tramite il **collegamento degli ingressi** (Indirizzo A, B) ai **campi del registro IR** (5 bit)
- ▶ Gli ingressi **Indirizzo A** e **Indirizzo B** selezionano **simultaneamente** due registri e il loro valore è fornito nelle uscite **A** e **B**. L'ingresso **Indirizzo C** seleziona un registro per il dato di ingresso **C**
- ▶ La soluzione con un unico banco duplica i circuiti di accesso e percorsi dei dati, l'altra soluzione (a destra) duplica le locazioni di memoria



Unità Aritmetica-Logica (ALU)

- ▶ L'ALU è collegata al banco dei registri
- ▶ L'uscita **A** del banco è collegata all'ingresso **InA** dell'ALU
- ▶ Si usa un multiplexore (MuxB) su uno degli ingressi (InB) per selezionare un operando sorgente immediato
- ▶ L'uscita dell'ALU è collegata all'ingresso dati C del banco dei registri per poter scrivere i risultati nel registro destinazione



Prof. Tramontana

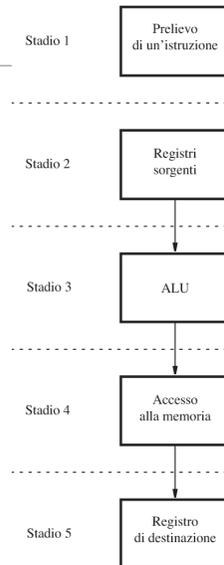
Struttura a cinque stadi

- ▶ Schema di esecuzione a stadi

1. Preleva istruzione e la pone in **IR**, incrementa **PC**. L'informazione in **IR** genera segnali di controllo per tutti i passi
2. Decodifica istruzione e legge registri dal banco di registri
3. Esegue un'operazione dell'ALU
4. Legge o scrive dati in memoria se l'istruzione ha operando in memoria
5. Scrive il risultato nel registro destinazione, se necessario

- ▶ Ogni stadio è completato in un periodo di clock

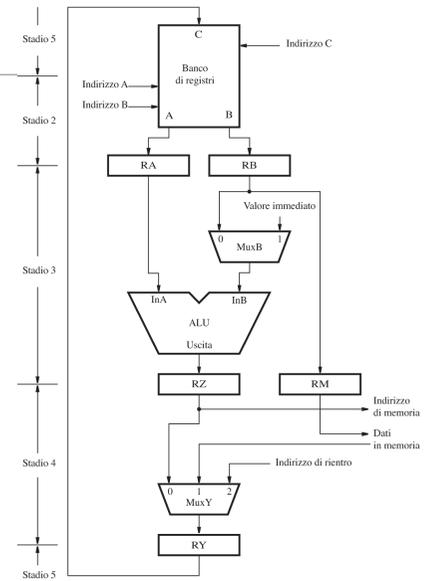
Prof. Tramontana



Percorso dati (datapath)

- ▶ I registri interstadi (inter-stage), **RA, RB, RZ, RM, RY**, tengono i risultati prodotti da uno stadio, sono usati come ingressi per il prossimo stadio durante il ciclo di clock successivo
- ▶ Stadi
 2. Legge dal banco di registri e scrive i valori letti dai registri su **RA** e **RB**
 3. Esegue un'operazione dell'ALU sui valori dati agli ingressi, InA collegato a **RA** e InB collegato a **RB** o Valore immediato, pone risultato in **RZ**
 4. Legge o scrive in memoria se necessario
 5. Scrive risultato in un registro se necessario

Prof. Tramontana



Percorso dati (datapath)

Load R5, X(R7)

2. Leggi R7 dal banco di registri e scrivilo in RA
3. Esegui somma in ALU con input RA e valore immediato X, poni risultato in RZ
4. Leggi la locazione di memoria all'indirizzo tenuto in RZ e poni il valore letto in RY
5. Scrivi RY in R5

Store R6, X(R8)

2. Leggi R8 e R6 dal banco di registri e scrivilo in RA e RB, rispettivamente
3. Esegui somma in ALU con input RA e valore X, poni risultato in RZ, inoltre poni RB in RM
4. Scrivi nella locazione di memoria con indirizzo RZ il valore RM

- Le istruzioni di chiamata a sottoprogramma salvano l'indirizzo di rientro, a questo fine, il MuxY presenta un ingresso ulteriore, collegato al generatore di indirizzi

Prof. Tramontana

