



Architettura degli Elaboratori

Informatica per la Comunicazione Digitale

Università degli Studi di Milano

Lezione 10:

## Banco dei registri e datapath (parte B)

Marco Tarini

26



### Varianti e integrazioni dei register file

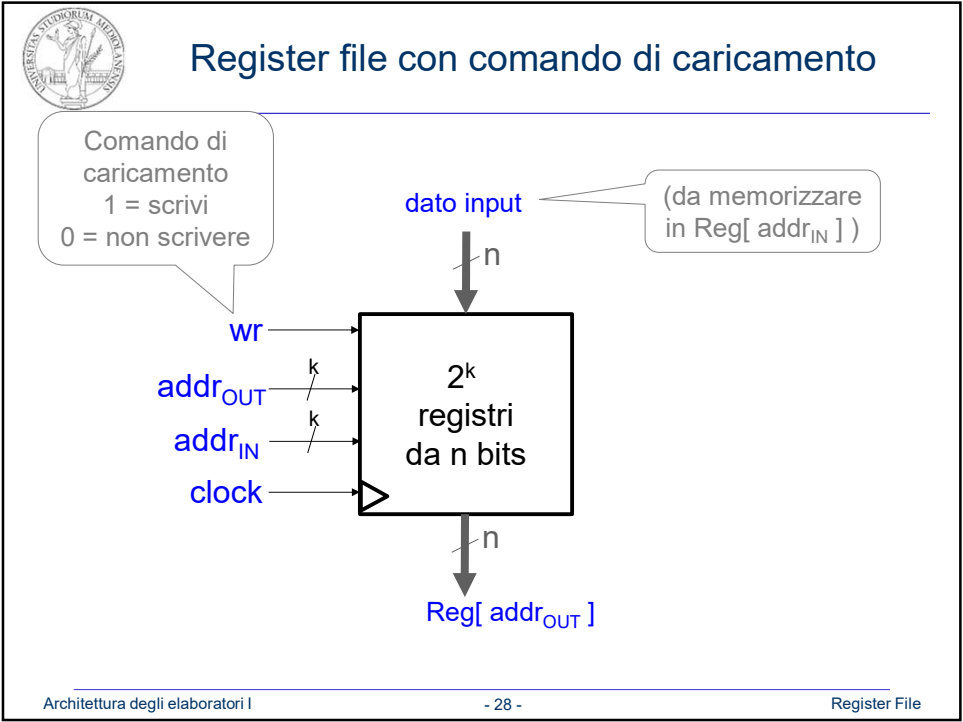
- Posso scegliere il numero di «bus», di entrata e di uscita di un Register File
  - ▶ nell'esempio visto l'ultima volta, il Register File è dotato di:
    - 2 bus di uscita  
(da cui leggere il contenuto di 2 registri dati, contemporaneamente!)
    - 1 di entrata  
(da cui immettere il contenuto da memorizzare in 1 registro dato)
  - ▶ Altre scelte sono possibili: per esempio, posso avere 2 canali di entrata e 3 di uscita (ogni canale = dato e address).
- Bit ulteriori di comando possono che essere aggiunti al register file:
  - ▶ **comando di caricamento** (write o WR), per abilitare/disabilitare la memorizzazione.  
Quando WR = 0 non viene memorizzato alcun valore in alcun registro (quindi, gli input data\_in e address\_in vengono ignorati)
  - ▶ **comando di azzeramento** (reset): per azzerare uno o anche tutti i registri contemporaneamente

Architettura degli elaboratori I

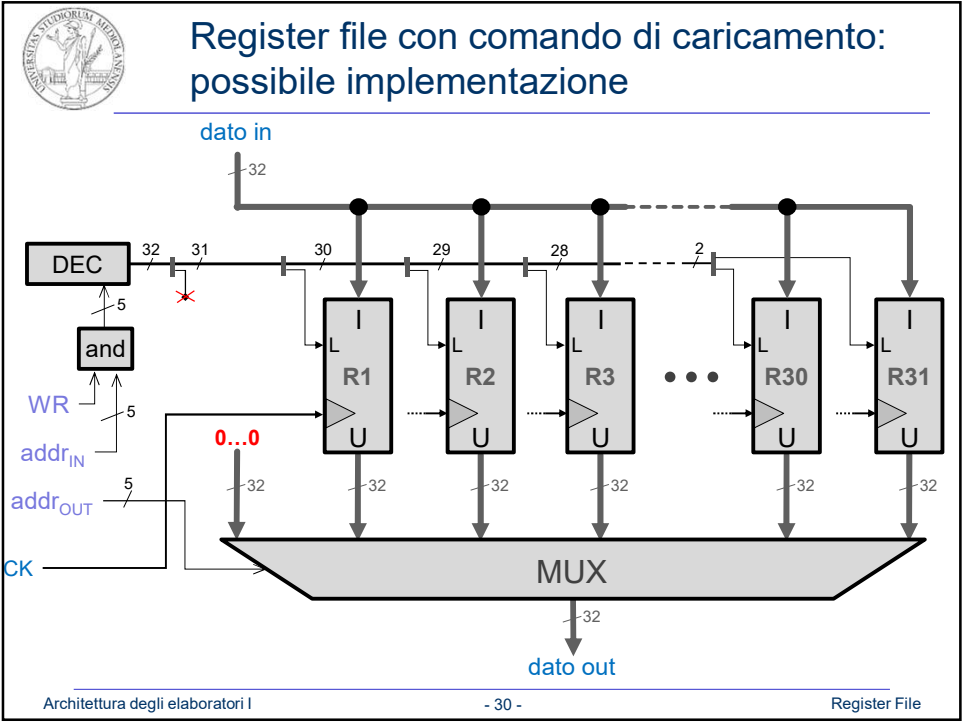
- 27 -

Register File


27



28



30




## Register file con comando di caricamento: possibile implementazione (spiegazione)

- Descrizione a basso livello (implementazione del blocco funzionale):
  - ▶ Quando il comando di caricamento «write» è basso (0), annullo semplicemente tutto l'indirizzo di input, tramite AND (quindi scrivo su \$0)
  - ▶ Anzi, ometto completamente il registro \$0 (risparmio di flip-flops)
  - ▶ Quando è richiesta una scrittura su \$0 ( $WR = 1, addr\_in = 0$ ): nessun registro esistente avrà il comando di caricamento settato
  - ▶ Quando è richiesta una lettura da \$0 ( $WR = 0, addr\_out = 0$ ): mando in output una costante 0..0
  - ▶ Vantaggio: semplicità di implementazione
- Descrizione ad alto livello (come blocco funzionale):
  - ▶ **Il registro \$0 è speciale: memorizza sempre e solo il valore "0"**
  - ▶ Scrivere nel registro \$0 non ha effetto, leggere da \$0 significa ricevere 0
  - ▶ Svantaggio: ho un registro in meno
  - ▶ Piccolo vantaggio collaterale: la ALU non deve mettere a disposizione un'istruzione "Zero" (che restituisca costantemente 0) – basta usare \$0

Architettura degli elaboratori I- 31 -Register File

31



## Domande

- E' possibile, nel data path visto, leggere dallo stesso registro due volte del register file?
  - ▶ E scrivere nello stesso registro in cui si sta leggendo?
  - ▶ Ad esempio, è lecito il comando assembly  
**ADD \$5 \$5 \$5**  
che (se è lecito), ha l'effetto di scrivere nel registro numero 5 la somma del suo valore con se stesso – quindi, il suo doppio?
  - ▶ Per rispondere, devi guardare come è implementato il banco di registri
- Mettiamo di voler copiare il registro \$6 nel registro \$9. Ti viene in mente un comando del nostro linguaggio assembly per ottenere questo effetto, usando solo le operazioni matematiche di base, sapendo che il register file adotta la soluzione vista per implementare il comando di caricamento?

Architettura degli elaboratori I- 32 -Register File

32