



**UNIVERSITÀ
DEGLI STUDI
DI TRIESTE**

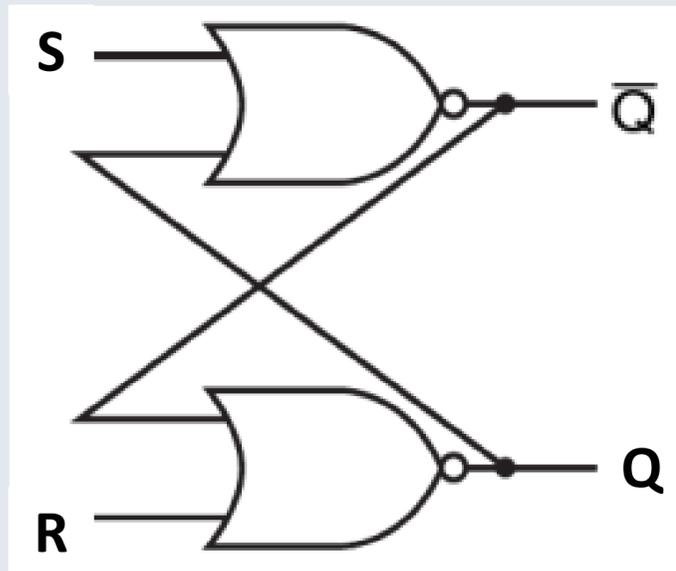
Circuiti in logica sequenziale

Prof.ssa Giulia Cisotto

giulia.cisotto@units.it

Trieste, 09 aprile 2025

S-R Latch è un circuito composto da 2 porte NOR concatenate. **S = Set** e **R = Reset**.



| Input | | Stato Interno | Output | |
|-------|---|---------------|--------|-----------|
| S | R | Old Q | Q | \bar{Q} |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1/0 | 0 | 1 |
| 1 | 0 | 1/0 | 1 | 0 |
| 1 | 1 | 1/0 | 0 | 0 |

S = 0 e R = 0
Configurazione di riposo

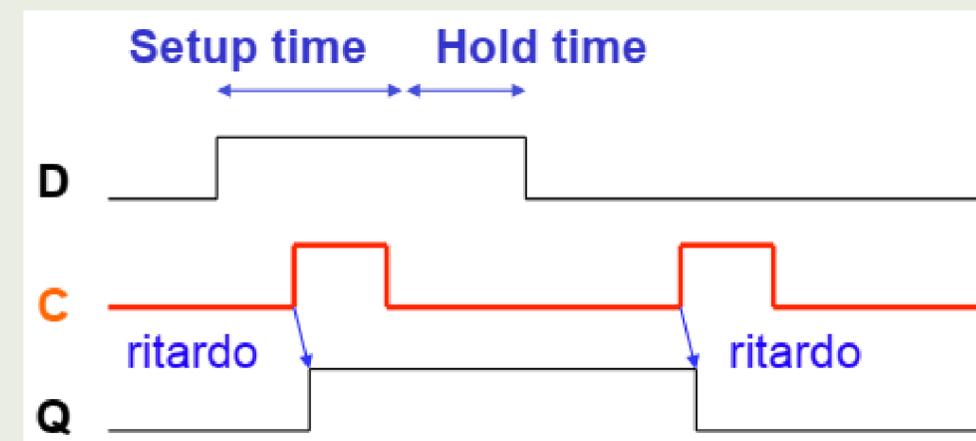
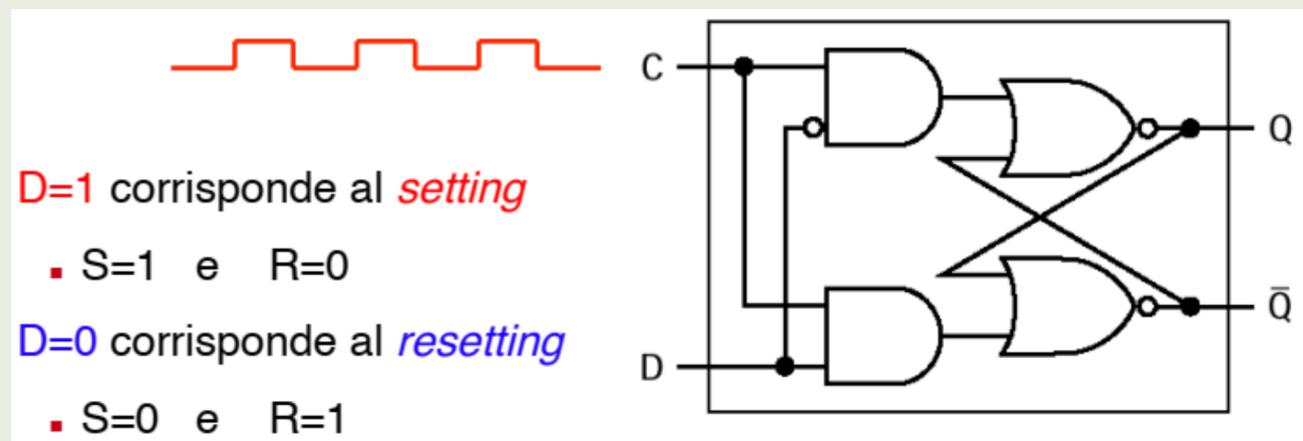
| Input | | Stato Interno | Output | |
|-------|---|---------------|--------|-----------|
| S | R | Old Q | Q | \bar{Q} |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1/0 | 0 | 1 |
| 1 | 0 | 1/0 | 1 | 0 |
| 1 | 1 | 1/0 | 0 | 0 |

S = 1 e R = 0
Configurazione di SET

| Input | | Stato Interno | Output | |
|-------|---|---------------|--------|-----------|
| S | R | Old Q | Q | \bar{Q} |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1/0 | 0 | 1 |
| 1 | 0 | 1/0 | 1 | 0 |
| 1 | 1 | 1/0 | 0 | 0 |

S = 0 e R = 1
Configurazione di RESET

D Latch è un circuito composto da 1 S-R latch, due porte AND e il clock. **C = clock** e **D = Data**.



METODOLOGIE DI TIMING

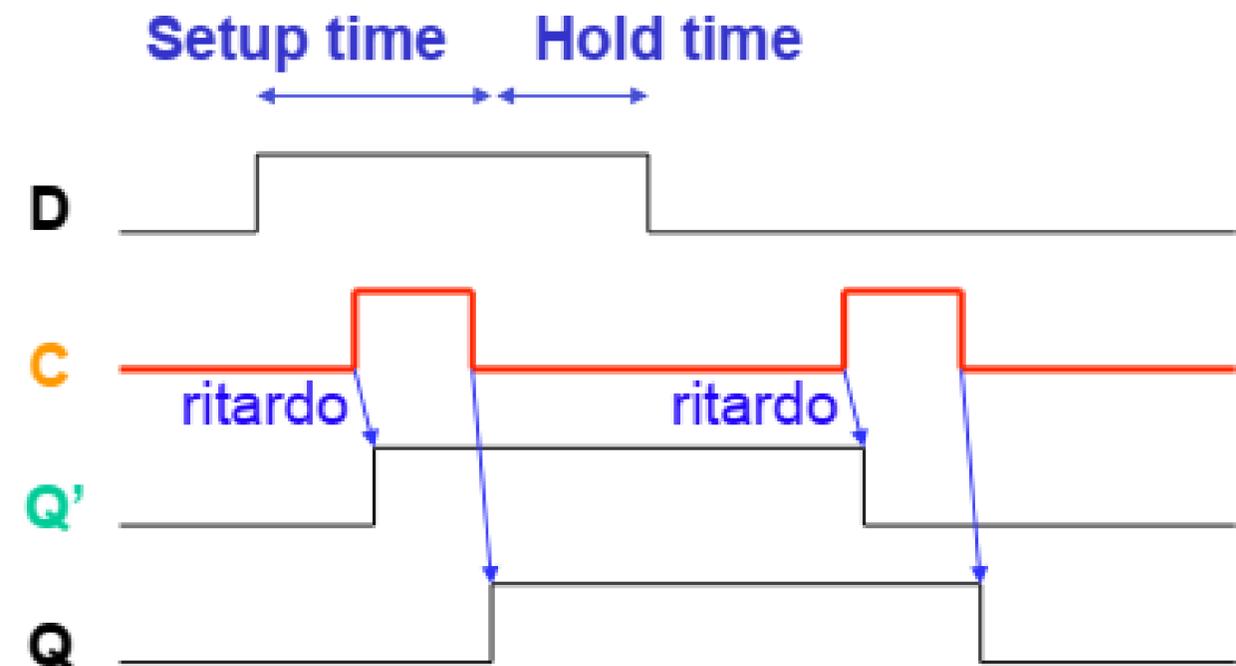
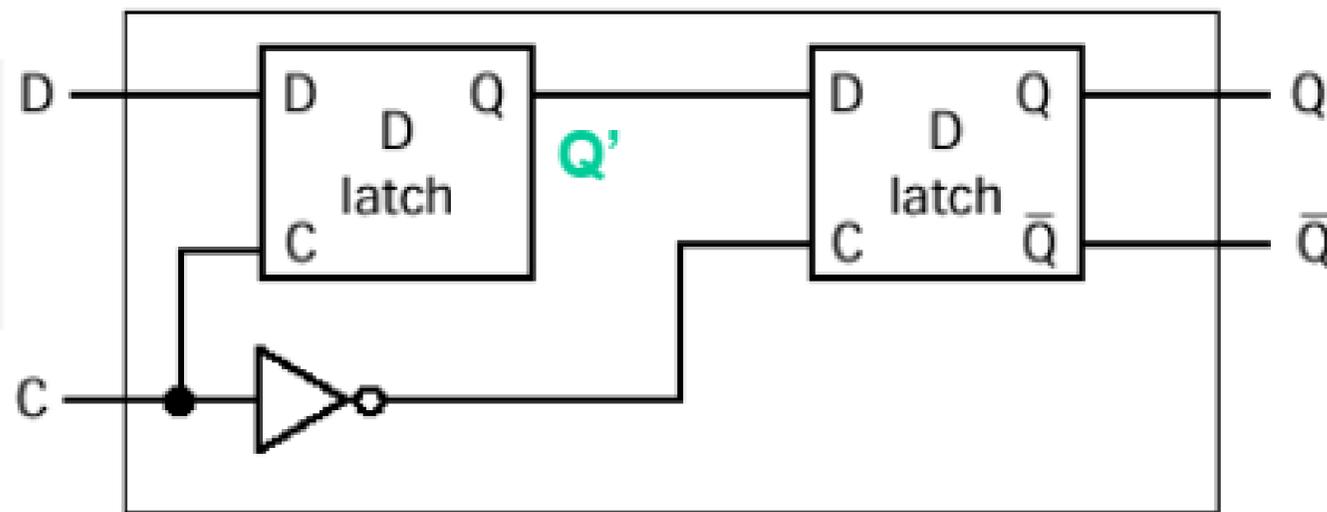
- Fino ad ora abbiamo visto una metodologia di timing detta **level-triggered** che avviene sul livello alto (o basso) del clock
- Esiste una metodologia di timing chiamata **edge-triggered** che avviene sul fronte di salita (o di discesa) del clock
 - la memorizzazione avviene istantaneamente
 - l'eventuale segnale di ritorno «sporco» non fa in tempo ad arrivare a causa dell'istantaneità della memorizzazione

flip-flop

D FLIP-FLOP

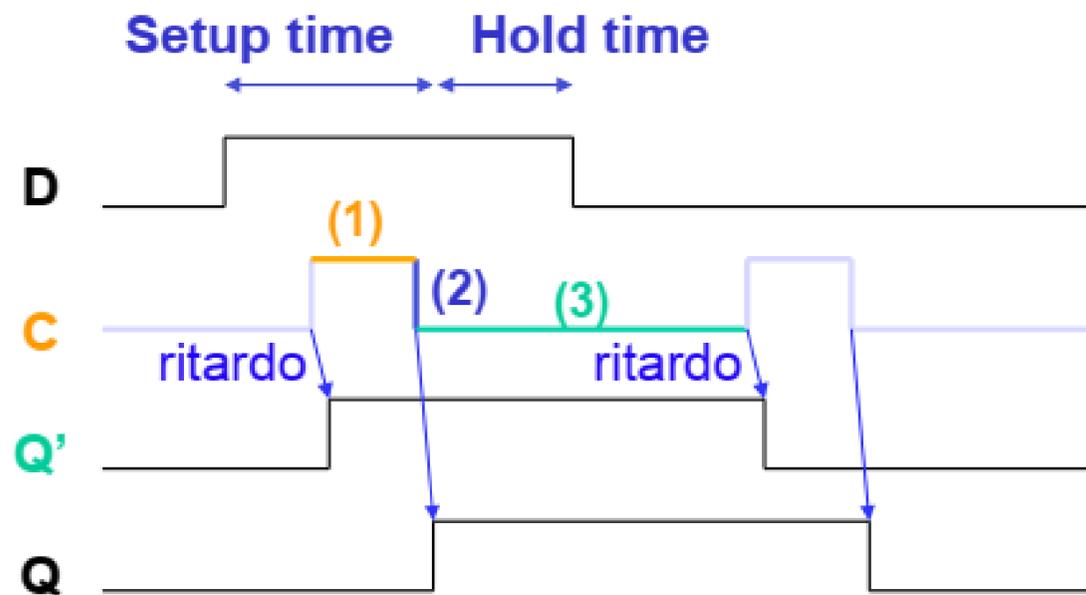
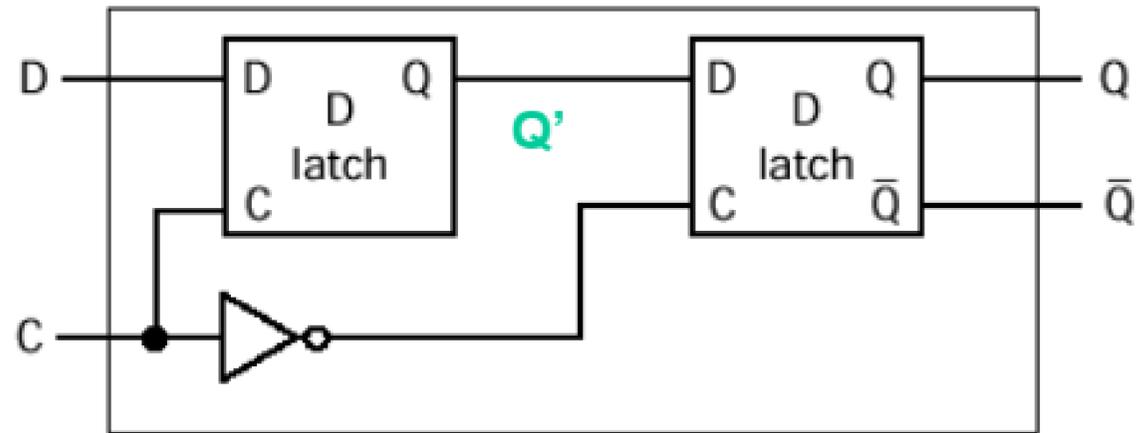
Il D Flip-flop è usabile come input e output durante lo stesso ciclo di clock

Realizzato ponendo in serie 2 D-latch: il primo viene detto **master** e il secondo **slave**



Nota. Si sfrutta sia il fronte di salita del clock che quello di discesa. Questo permette di **separare le fasi** di acquisizione del nuovo dato ($D \rightarrow Q'$) e di aggiornamento dello stato ($Q' \rightarrow Q$)

D FLIP-FLOP



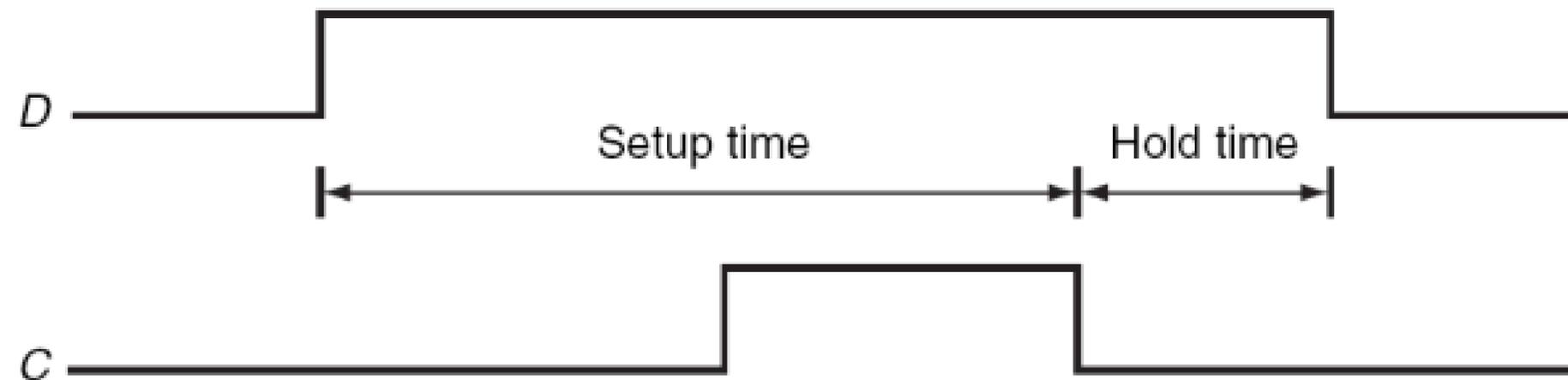
(1) Il **primo latch è aperto** e pronto per memorizzare D. Il valore memorizzato Q' fluisce fuori, ma il **secondo latch è chiuso** \Rightarrow nel circuito combinatorio a valle entra ancora il vecchio valore di Q.

(2) Il segnale del clock scende, e in questo istante il secondo latch viene aperto per memorizzare il valore di Q' .

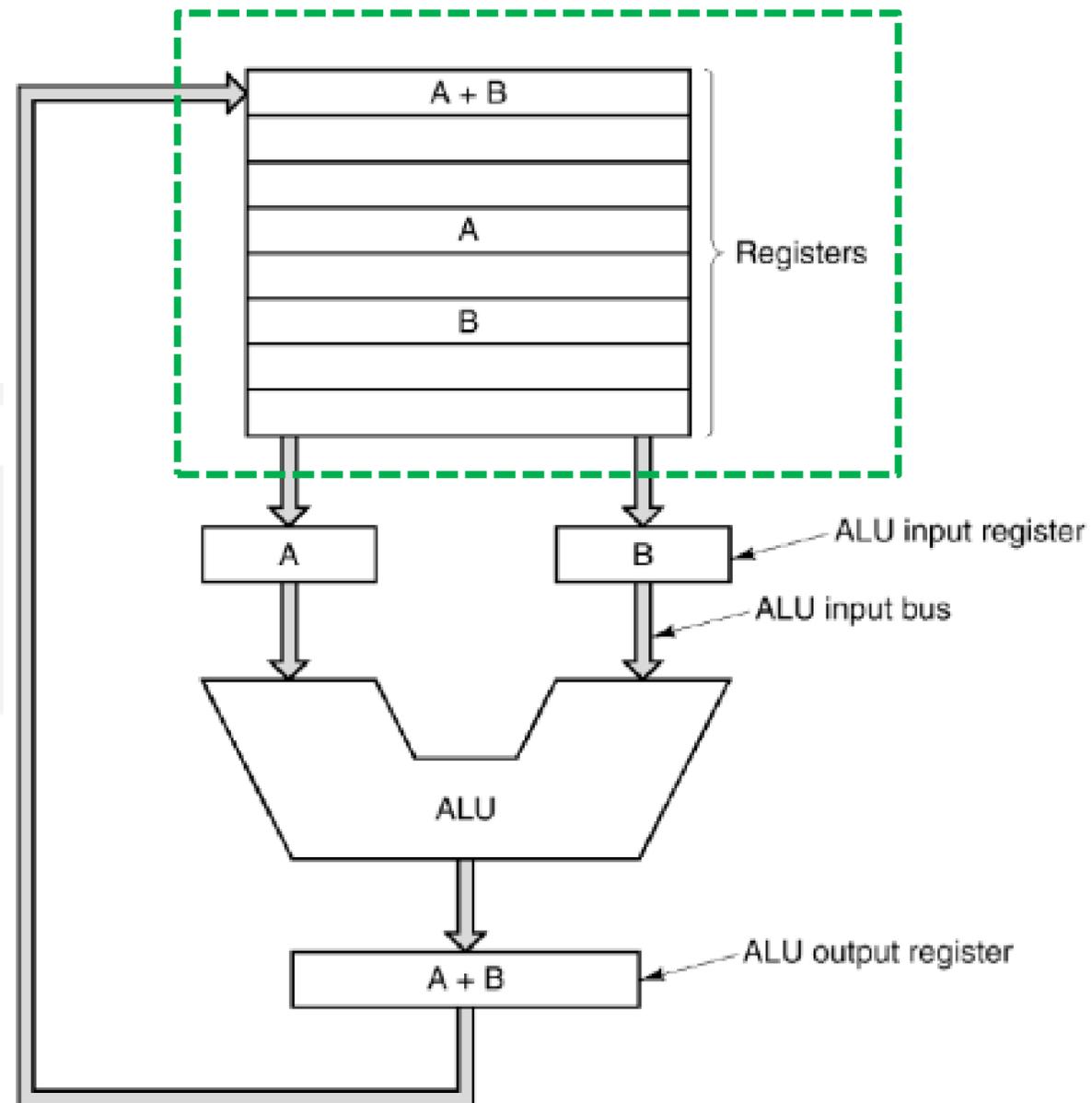
(3) Il **secondo latch è aperto**, memorizza D (Q'), e fa fluire il nuovo valore Q nel circuito a valle. Il **primo latch** è invece **chiuso**, e non memorizza niente.

D FLIP-FLOP

Il segnale D deve essere attivo per un periodo abbastanza lungo:
setup time (prima del clock edge) + hold time (dopo il clock edge)



DATAPATH E REGISTER FILE



Un **registro** è costituito da **n flip-flop**

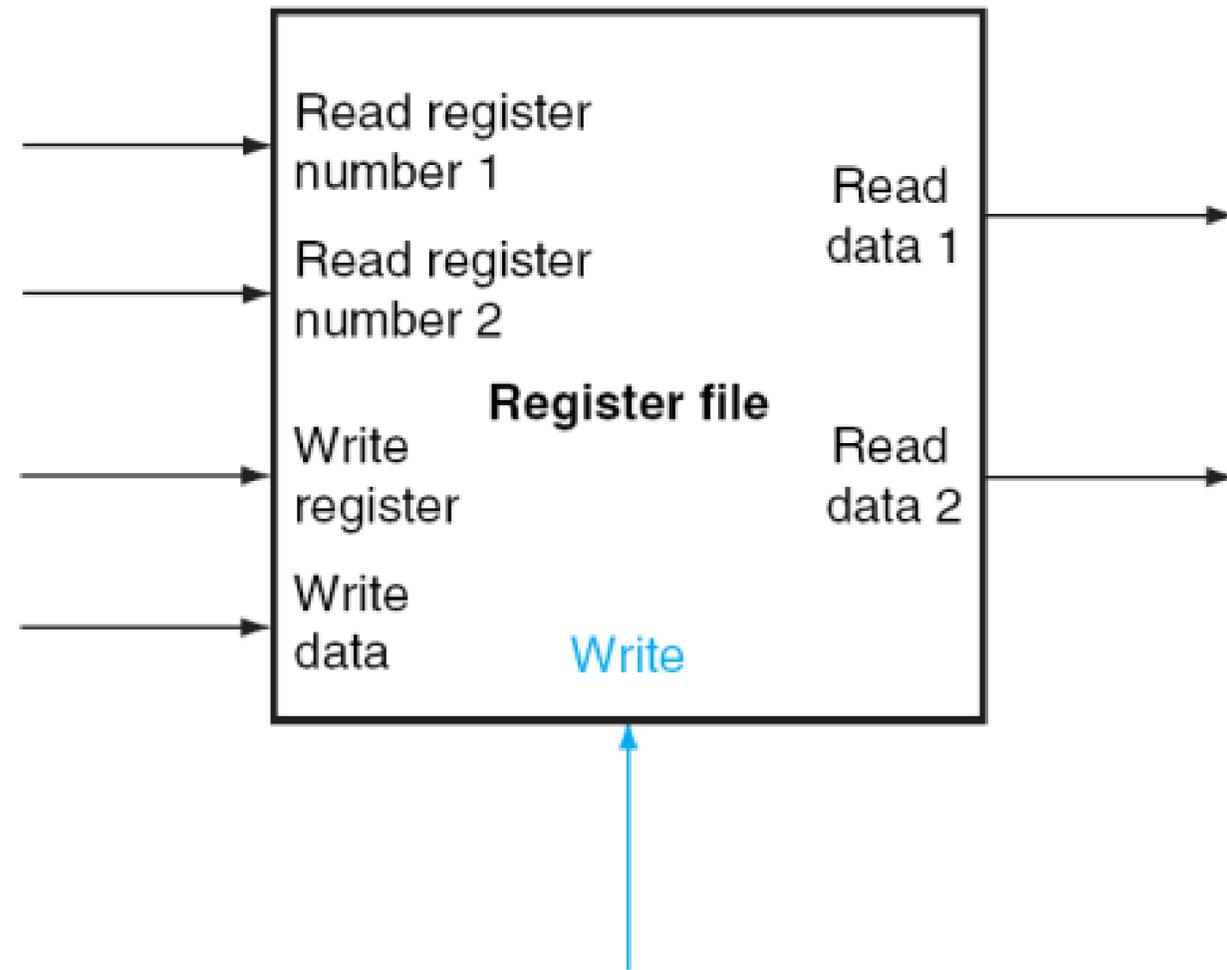
➤ Nel MIPS ogni registro è di 1 word = 4 byte = 32 bit

I registri sono organizzati in un **Register File**

➤ Il Register File del MIPS ha 32 registri (32 x 32 = 1024 flipflop)

➤ Il Register File permette la lettura di 2 registri e la scrittura di 1 registro

REGISTER FILE



REGISTER FILE

Read Reg1 # (5 bit)

n numero del 1o registro da leggere

Read Reg2 # (5 bit)

n numero del 2o registro da leggere

Read data 1 (32 bit)

n valore del 1o registro, letto sulla base di Read Reg1 #

Read data 2 (32 bit)

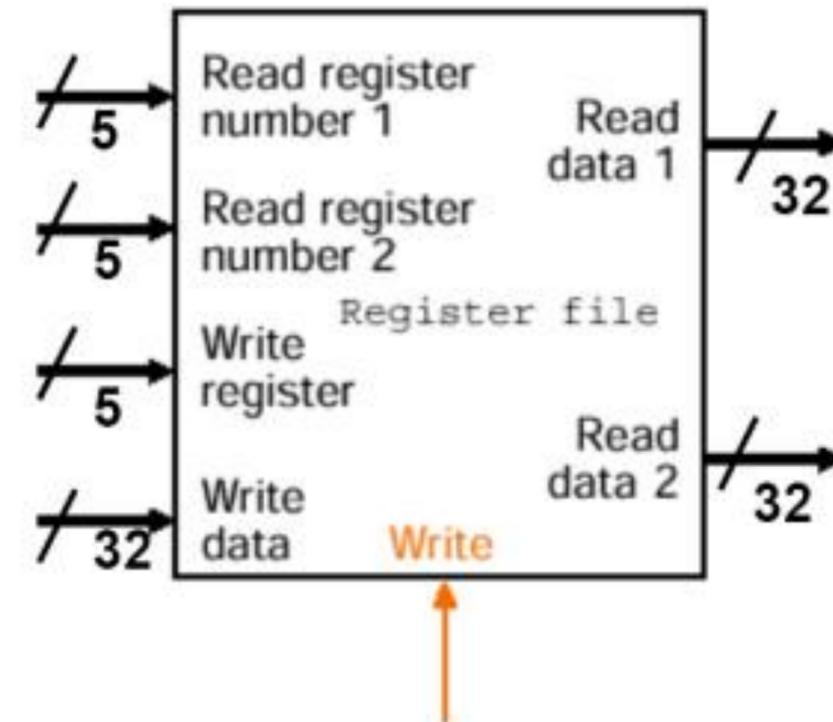
n valore del 2o registro, letto sulla base di Read Reg2 #

Write Reg # (5 bit)

n numero del registro da scrivere

Write data (32 bit)

n valore da scrivere nel registro Write Reg #

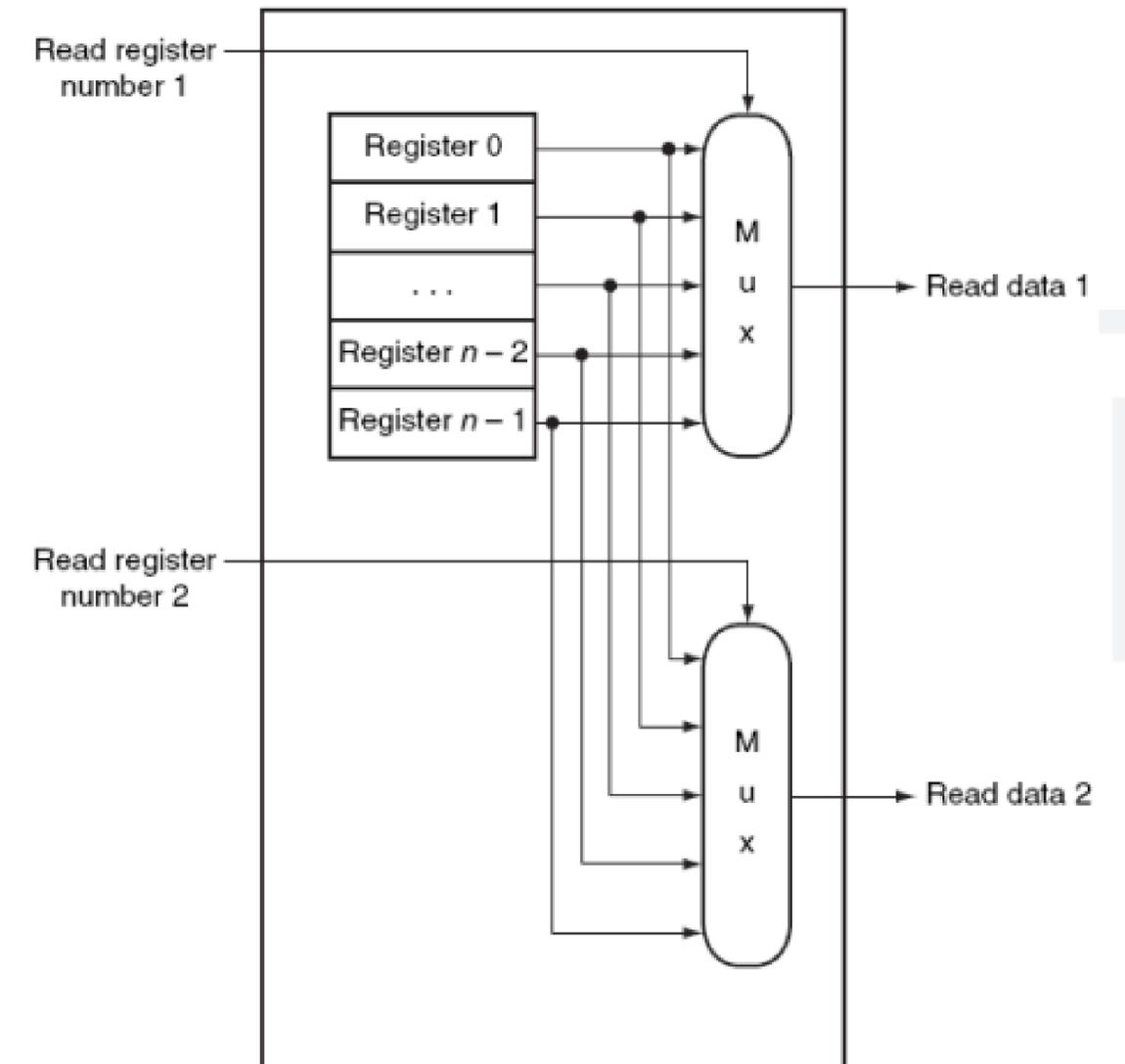


Write

- segnale di controllo messo in AND con il *clock*
- solo se **Write=1**, il valore di **Write data** viene scritto in uno dei registri

LETTURA DAL REGISTER FILE

- Utilizza 2 segnali che indicano i registri da leggere (Read Reg1, Read Reg2)
- Utilizza 2 multiplexer: ognuno con 32 ingressi e un segnale di controllo
- Il register file fornisce sempre in output una coppia di registri



SCRITTURA NEL REGISTER FILE

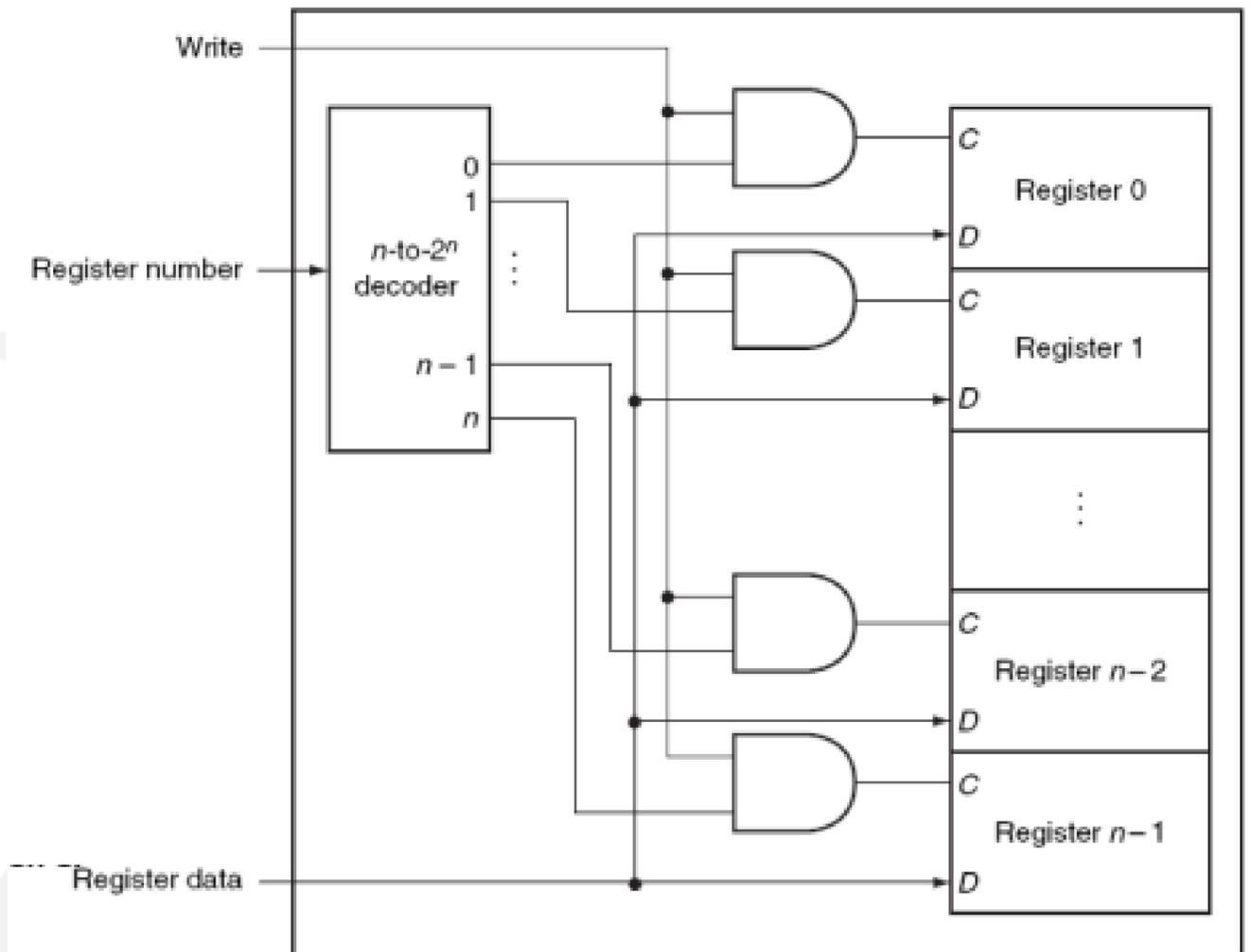
Utilizza 3 segnali:

- Il registro da scrivere (Register Number)
- Il valore da scrivere (Register Data)
- Il segnale di controllo (Write)

Utilizza un decoder che decodifica il numero del registro da scrivere (Write Register)

Il segnale Write (già in AND con il clock) è in AND con l'output del decoder

Se Write non è affermato nessun valore sarà scritto nel registro



Materiale per la lezione

- Appendice B Patterson & Hennessy