Architettura degli Elaboratori

Reti Sequenziali, Flip-Flop e Banco dei Registri

Barbara Masucci
UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Punto della situazione

- Abbiamo visto che il processore MIPS dispone di 32 registri interni da 32 bit ciascuno
- Come sono realizzati questi registri?

- Obiettivo di oggi
  - Studio di alcuni circuiti sequenziali (componenti con memoria)
    - Il latch, componente di base di tutti gli elementi di memoria che studieremo in seguito (memorizza un bit)
    - Il flip-flop, ottenuto dal latch mediante l'aggiunta di un segnale di clock (memorizza un bit)
    - Il registro, corrispondente a un array di 32 flip-flop con lo stesso segnale di clock (memorizza 32 bit)
    - Il banco dei registri, costituito dai 32 registri interni al MIPS

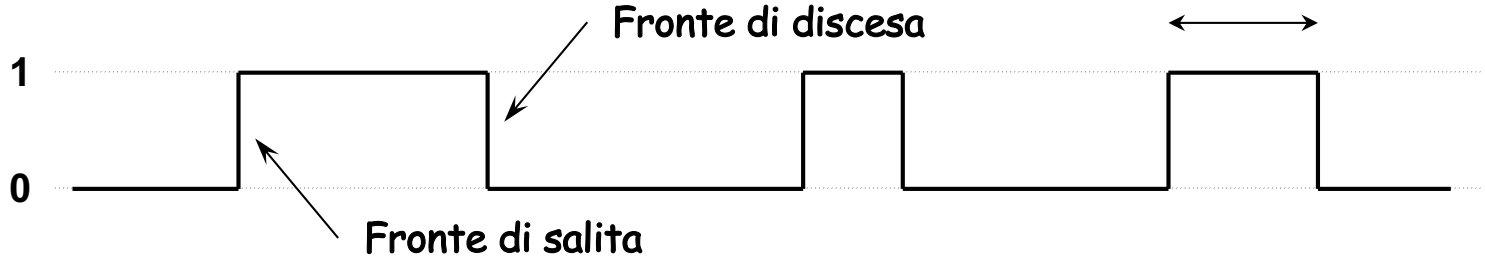
Circuiti sequenziali

- Un circuito combinatorio è
  - Una rete di porte logiche il cui output non dipende dagli input precedenti, ma solo dall'input corrente
  - Si tratta quindi di un circuito senza memoria
  - Appartiene a questa categoria l'ALU

- Un circuito sequenziale è
  - Una rete di porte logiche il cui output dipende non solo dall'input corrente, ma anche da input precedenti
  - Si tratta quindi di un circuito con memoria
  - Appartengono a questa categoria i registri, la memoria dati e la memoria istruzioni

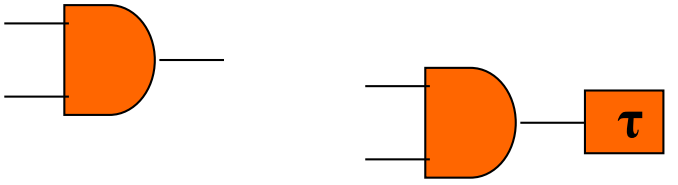
La componente tempo

- Nelle reti sequenziali, piuttosto che parlare di variabili binarie, parleremo di forme d'onda temporali binarie, cioè tensioni che
  - Ad ogni istante possono assumere uno dei due valori 0 e 1
  - Cambiano valore istantaneamente
  - Vengono rappresentate mediante diagrammi temporali



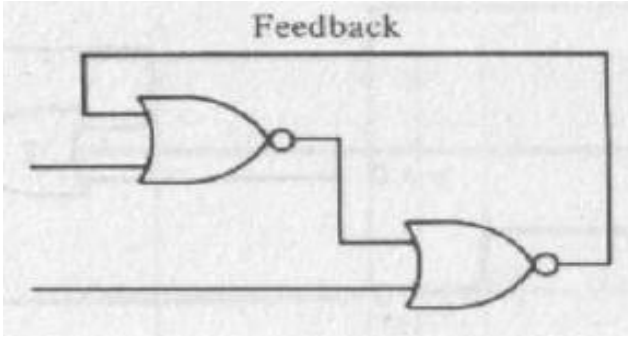
Ritardo di propagazione

- Nelle reti combinatorie abbiamo assunto che le varie porte realizzassero una trasmissione istantanea
- Adesso invece dobbiamo considerare il ritardo di propagazione (τ)
  - Ad esempio, mettiamo sul diagramma temporale gli input A,B di una porta AND e il suo output C=AB
    - Se A e B assumono valore 1 all'istante zero, C assumerà valore 1 all'istante τ
    - Se A e B assumono valore 0 all'istante t_1, C assumerà valore 0 all'istante $t_1 + \tau$
- Possiamo vedere una porta con ritardo come combinazione di
  - Una porta ideale, priva di ritardo
  - Un elemento di ritardo puro τ



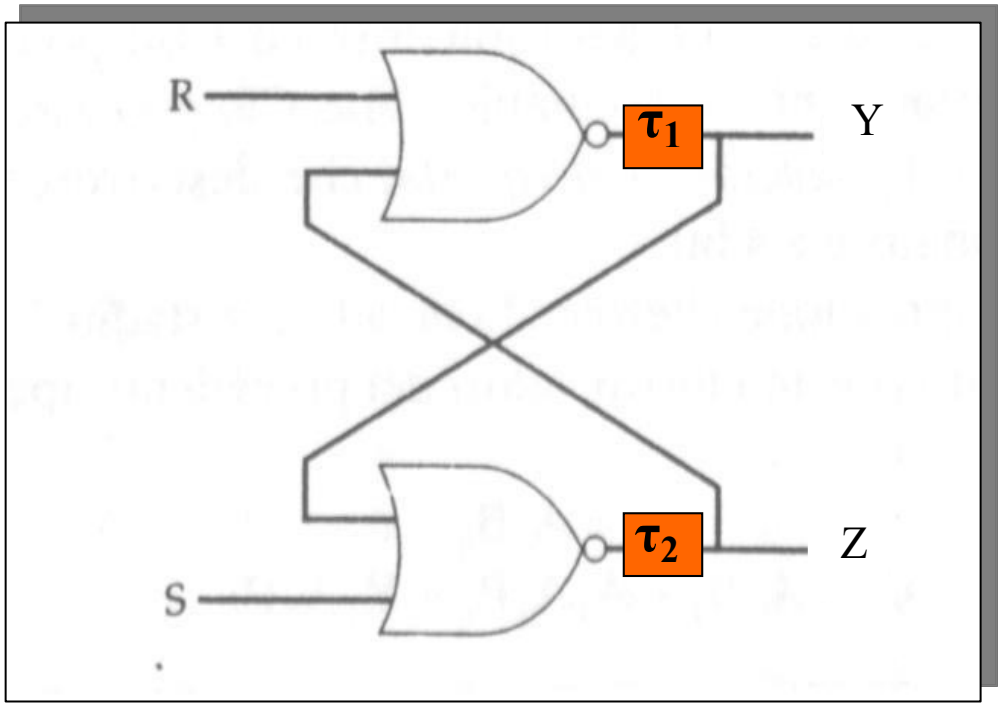
Latch S-R

- Per le reti combinatore abbiamo fornito dei circuiti di base (porte AND, OR, NOT)
- Vogliamo fare lo stesso per le reti sequenziali
- Vediamo quindi come è fatto il più piccolo circuito sequenziale (latch S-R)
  - Si tratta del componente di base di tutti gli elementi di memoria che saranno studiati in seguito
  - Iniziamo aggiungendo ad una semplice rete combinatoria, composta da due porte NOR interconnesse, una connessione di feedback
  - Poi ridisegnamo la rete risultante, mettendo in evidenza il fatto che ciascuna porta ha un ritardo



Latch S-R

- La rete risultante ha due ingressi (R ed S) e due uscite (Y e Z)



- L'espressione booleana in uscita per Y è la seguente:

$$Y = \overline{R+Z} = \overline{(R+\overline{(S+Y)})} = \overline{R}(S+Y)$$

la variabile Y dipende da sé stessa!

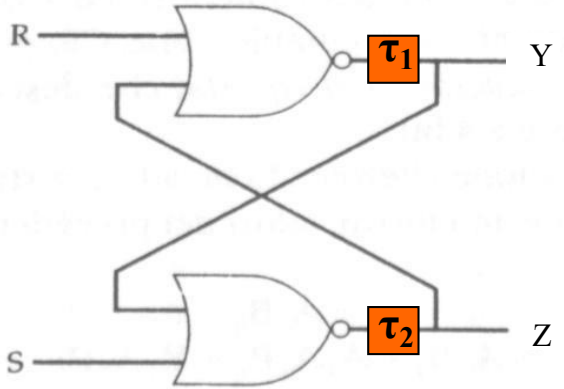
Latch S-R

- Tuttavia nell'espressione

$$Y = \overline{R}(S+Y)$$

la Y al primo membro e quella al secondo membro rappresentano due forme d'onda in istanti differenti

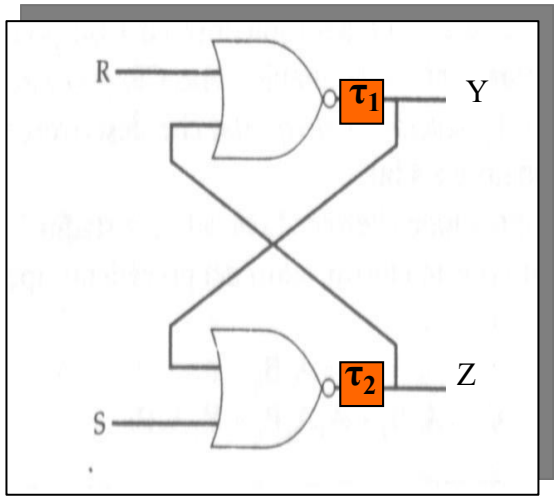
  - A causa dei ritardi τ_1 e τ_2 di propagazione delle porte



- Quindi, per le reti sequenziali, la descrizione attraverso espressioni logiche risulta inadeguata

Latch S-R

- Analizziamo il comportamento del latch quando R ed S sono contemporaneamente 0
  - Supponiamo che all'istante zero siano R=S=0, Y=1, Z=0
  - All'istante τ_1, Y resta 1 perché $Y=\overline{R+Z} = 1$
  - Inoltre all'istante $\tau_1+\tau_2$, Z resta 0 perché $Z=\overline{Y+S} = 0$

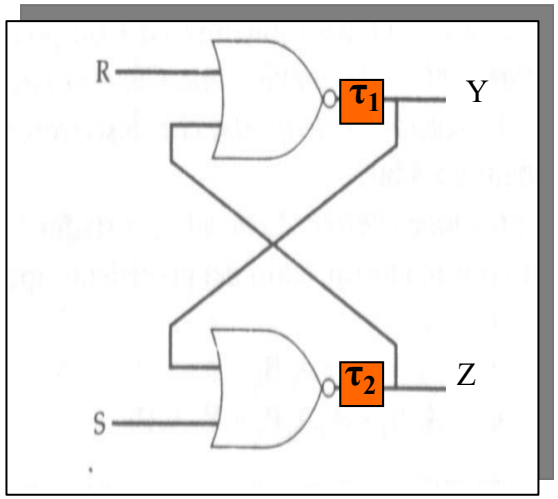


Quindi la situazione S=R=0 non modifica gli output Y=1 e Z=0 del latch

Stabilità:
il latch resta nello stesso stato che aveva prima

Latch S-R

- Analizziamo il comportamento del latch quando R ed S sono contemporaneamente 0
  - Analogamente, supponiamo che all'istante zero siano R=S=0, Y=0, Z=1
  - All'istante τ_1, Y resta 0 perché $Y=\overline{R+Z} = 0$
  - Inoltre all'istante $\tau_1+\tau_2$, Z resta 1 perché $Z=\overline{Y+S} = 1$

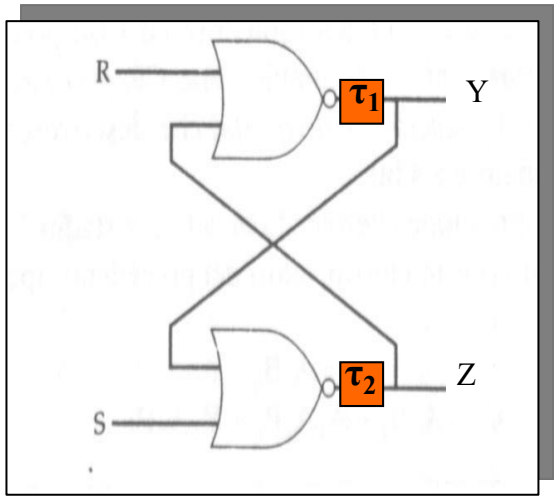


Quindi la situazione S=R=0 non modifica gli output Y=0 e Z=1 del latch

Stabilità:
il latch resta nello stesso stato che aveva prima

Latch S-R

- Analizziamo il comportamento del latch quando R ed S non sono contemporaneamente 0
  - Supponiamo che all'istante zero siano R=S=0, Y=1, Z=0
  - Se R diventa 1 all'istante t_1, all'istante $t_1+\tau_1$, Y diventa 0 perché $Y=\overline{R+Z} = \overline{1+0} = 0$ (CAMBIO DI STATO)
  - Inoltre all'istante $t_1+\tau_1+\tau_2$, Z diventa 1 perché $Z=\overline{Y+S} = \overline{0+0} = 1$

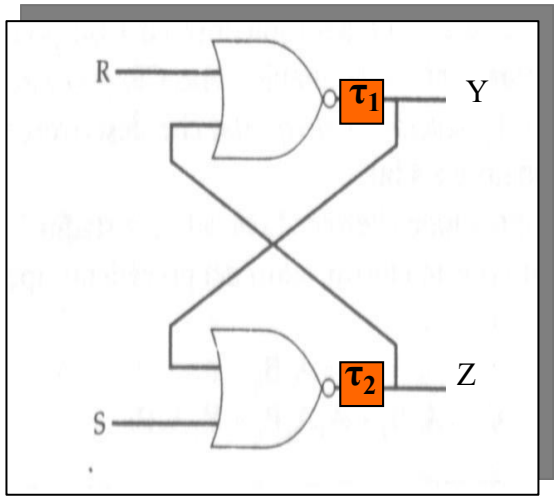


Quindi l'output del latch è passato da (Y=1, Z=0) a (Y=0, Z=1) a causa dell'azione di R

Passaggio di Y da 1 a 0

Latch S-R

- Che succede se il segnale R cambia nuovamente dopo aver provocato il cambio di stato (Y=0, Z=1)?
  - Supponiamo che R diventi 0 all'istante $t_2 > t_1+\tau_1+\tau_2$
  - Poiché gli ingressi alla prima porta NOR sono R=0 e Z=1, all'istante $t_2+\tau_1$ l'uscita Y=0 viene mantenuta
  - Poiché gli ingressi alla seconda porta NOR sono Y=0 e S=0, all'istante $t_2+\tau_1+\tau_2$, l'uscita Z=1 viene mantenuta

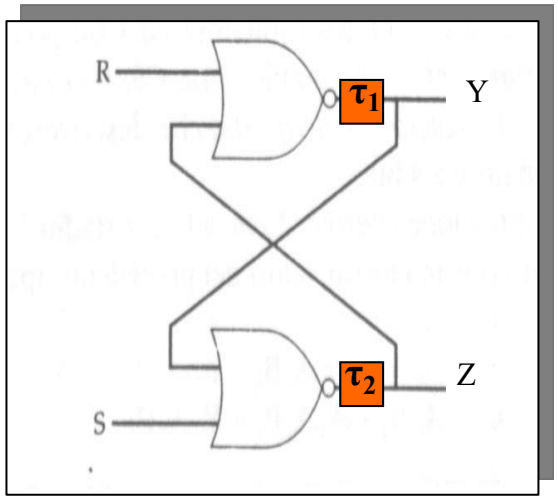


Non c'è un ulteriore cambio di stato!

Al di fuori dell'intervallo $[t_1,t_2]$
il latch è in uno stato stabile e $Z=\overline{Y}$

Latch S-R

- Analogamente,
  - Supponiamo che all'istante zero siano R=S=0, Y=0, Z=1
  - Se S diventa 1 all'istante t_1, all'istante $t_1+\tau_2$, Z diventa 0 perché $Z=\overline{Y+S} = \overline{0+1} = 0$
  - Inoltre all'istante $t_1+\tau_1+\tau_2$, Y diventa 1 perché $Y=\overline{R+Z} = \overline{0+0} = 1$ (CAMBIO DI STATO)

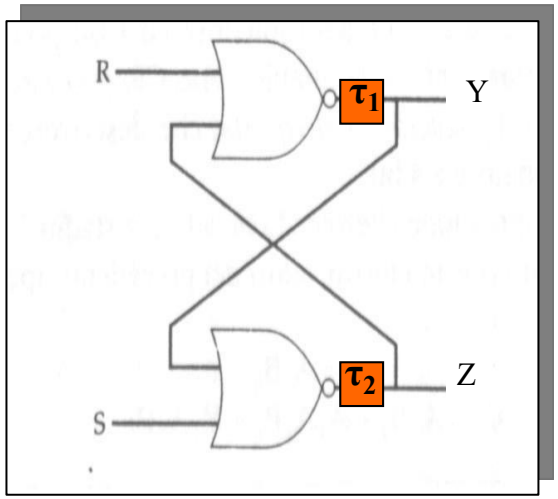


Quindi l'output del latch è passato da (Y=0, Z=1) a (Y=1, Z=0) a causa dell'azione di S

Passaggio di Y da 0 a 1

Latch S-R

- Che succede se il segnale S cambia nuovamente dopo aver provocato il cambio di stato (Y=1, Z=0)?
  - Supponiamo che S diventi 0 all'istante $t_2 > t_1+\tau_1+\tau_2$
  - Poiché gli ingressi alla seconda porta NOR sono Y=1 e S=0, all'istante $t_2+\tau_2$ l'uscita Z=0 viene mantenuta
  - Poiché gli ingressi alla prima porta NOR sono R=0 e Z=0, all'istante $t_2+\tau_2+\tau_1$, l'uscita Y=1 viene mantenuta

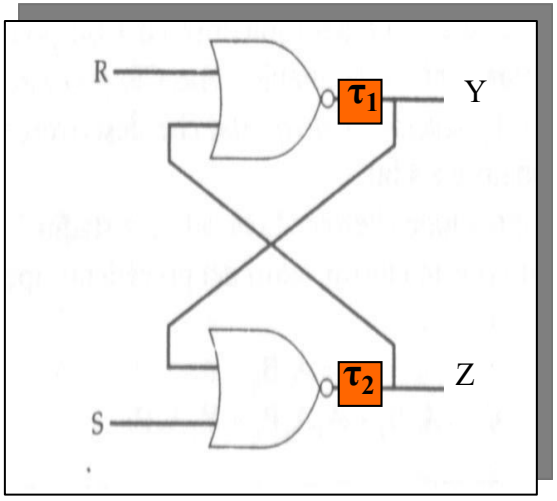


Non c'è un ulteriore cambio di stato!

Al di fuori dell'intervallo $[t_1,t_2]$,
il latch è in uno stato stabile e $Z=\overline{Y}$

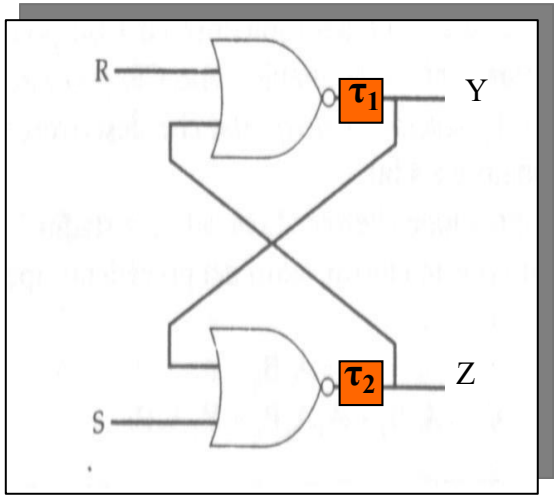
Latch S-R

- Gli output Y e Z del latch soddisfano la condizione $Z=\overline{Y}$
  - Y viene detto uscita non complementata,
  - $\overline{Y}$ viene detto uscita complementata



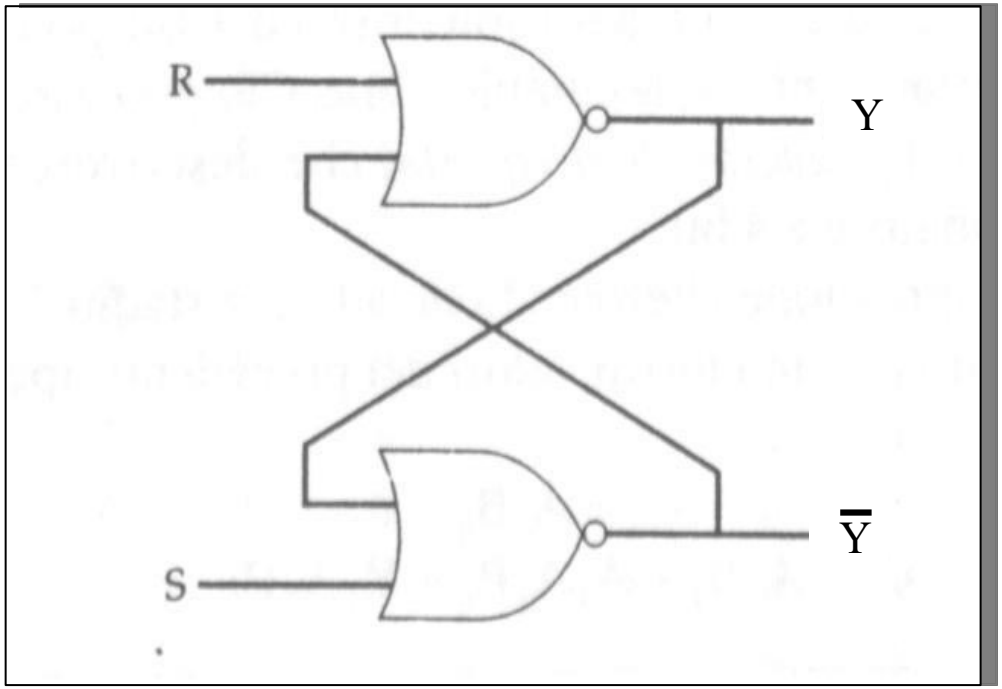
Latch S-R

- Gli input S ed R del latch vengono detti SET e RESET, rispettivamente
  - Poiché quando l'input S è uguale ad 1, esso fa diventare Y=1, esso viene detto SET
  - Poiché quando l'input R è uguale ad 1, esso fa diventare Y=0, esso viene detto RESET



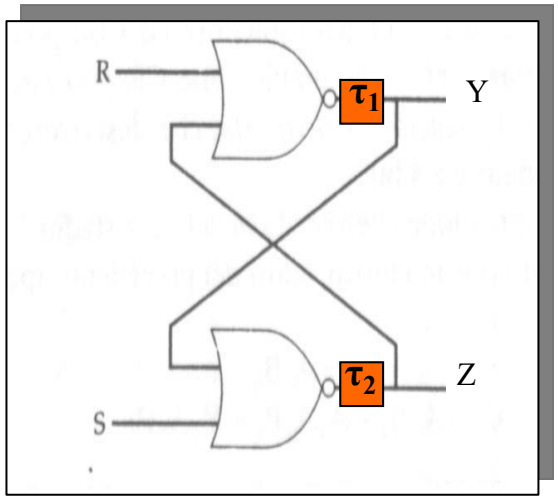
Latch S-R

- Normalmente, i quattro segnali del latch sono quindi
  - Due input: R ed S
  - Due output Y (o 1) e $\overline{Y}$ (o 0)



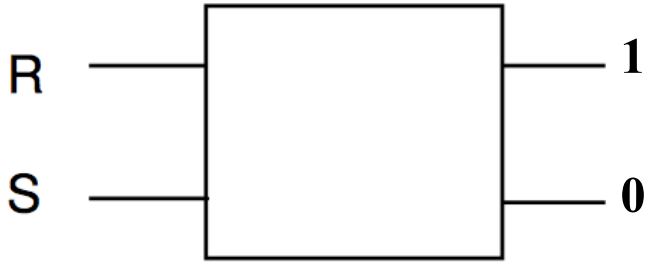
Latch S-R

- Nota: R ed S non possono valere contemporaneamente 1, perchè Y non può assumere due valori diversi
  - Se R=1, Y è forzato ad essere uguale a 0
  - Se S=1, Y è forzato ad essere uguale ad 1
- Questa configurazione viene contrassegnata come illegale, per cui vale sempre la condizione R AND S = 0



Latch S-R

- Quindi, il latch è
  - Nello stato Y=0 se per ultimo si è presentato l'input R=1
  - Nello stato Y=1 se per ultimo si è presentato l'input S=1
- Questo significa che la rete ricorda il (o anche, si aggancia al) suo ingresso
  - Latch = aggancio
- Il simbolo del latch nei circuiti digitali è il seguente



Latch S-R:
Tavola di verità

S	R	Stato attuale	Prossimo stato
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	-	-
1	1	-	-

Stabilità (S=R=0)

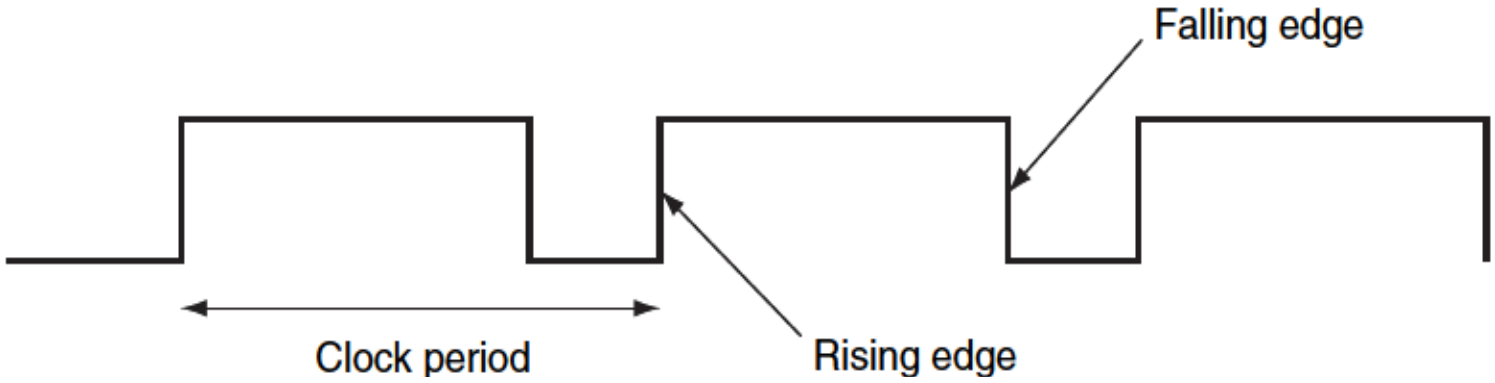
Reset (S=0, R=1)

Set (S=1, R=0)

Input non permessi

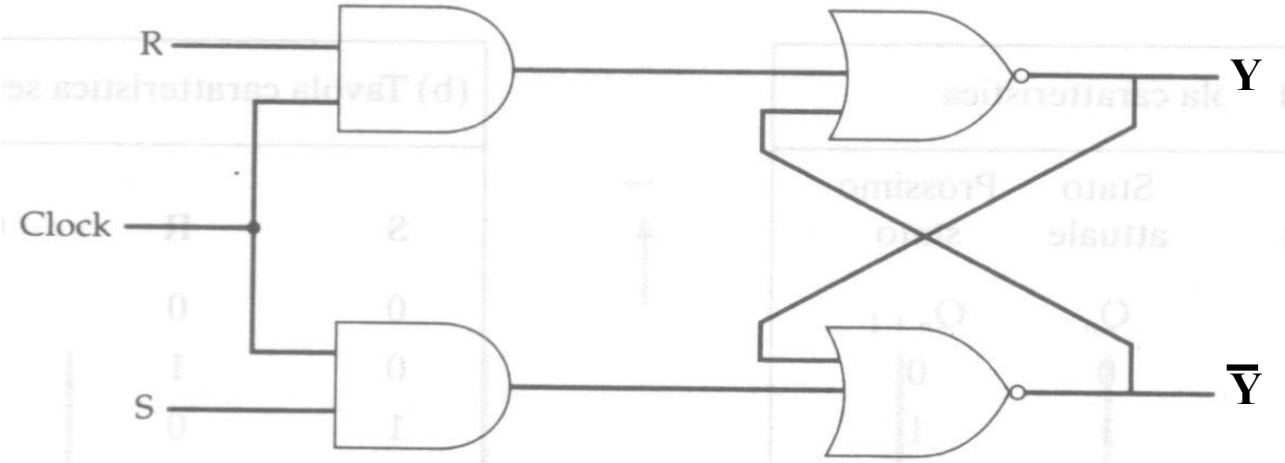
Aggiunta del Clock

- Modifichiamo il latch S-R aggiungendo un nuovo input: il Clock
  - Mandiamo i due segnali di ingresso S ed R in due porte AND il cui secondo ingresso è il Clock
- Il Clock è una forma d'onda che controlla gli istanti in cui un elemento di memoria può cambiare stato
  - E' importante temporizzare gli eventi per evitare che un dato venga contemporaneamente letto e scritto



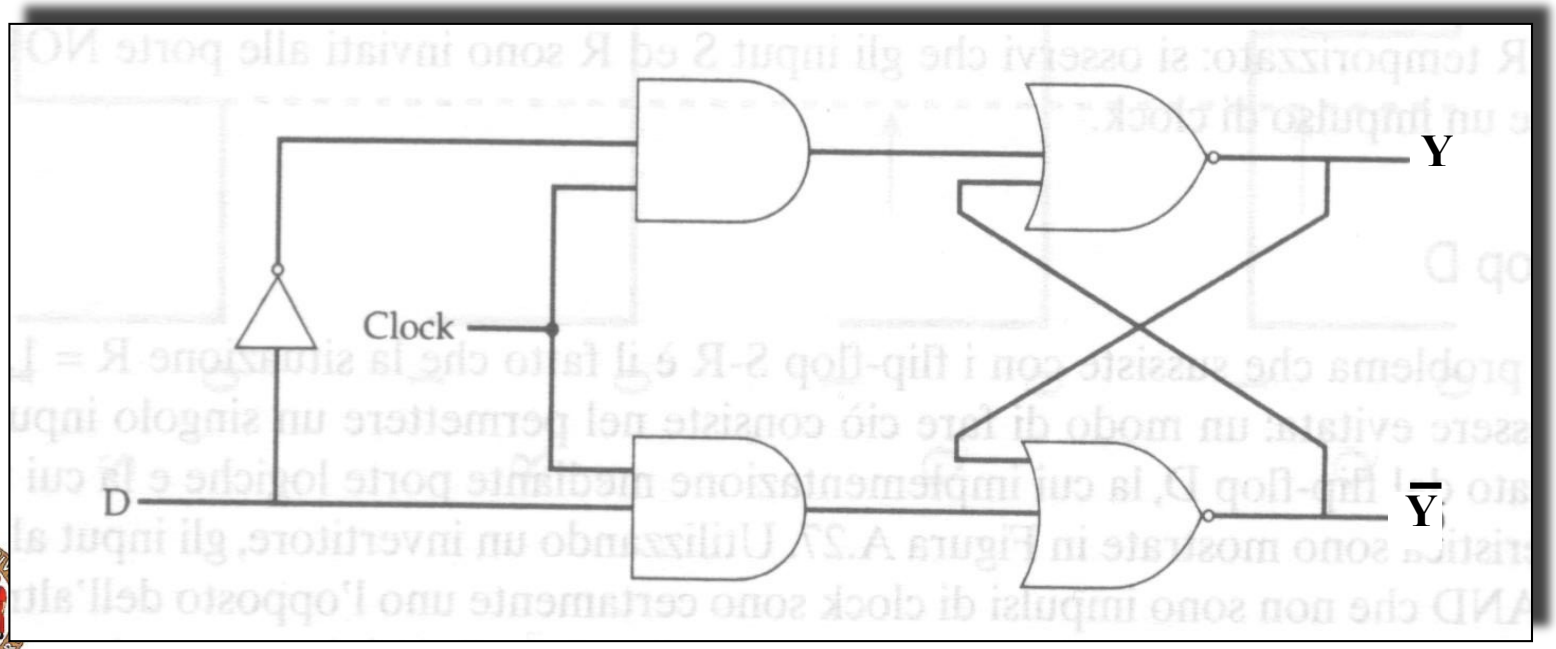
Flip-Flop S-R

- Il latch S-R a cui viene aggiunto un segnale di Clock viene detto Flip-Flop S-R
- I cambiamenti di stato del Flip-Flop S-R sono gestiti dal segnale di Clock in aggiunta ai segnali S ed R
- Il Flip-Flop S-R è sensibile ai cambiamenti di stato quando il fronte d'onda del clock è alto



Flip-Flop D

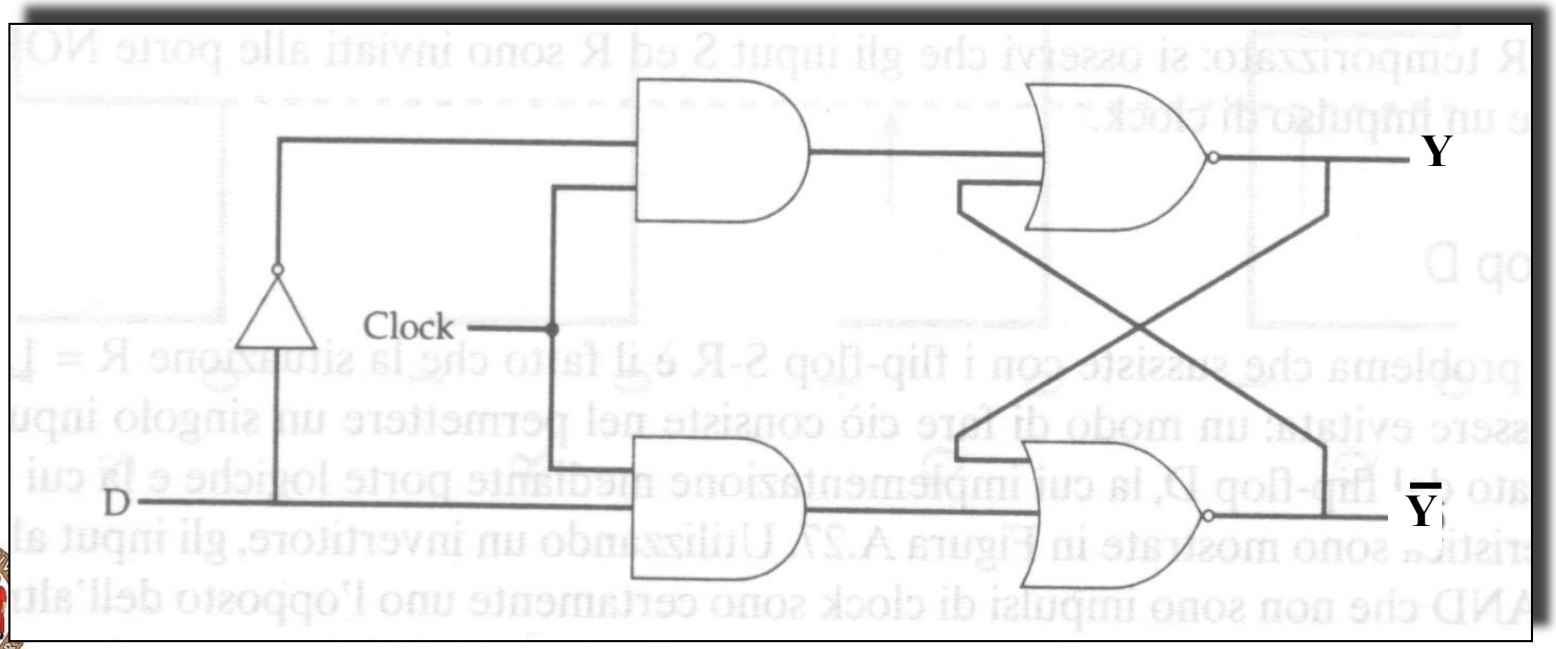
- Vediamo come modificare il Flip-Flop S-R per evitare l'input S=R=1
- Nel Flip-Flop D, c'è un unico input, D, che determina i segnali R ed S (S=D e R=$\overline{D}$)



Flip-Flop D

L'ouput del Flip-Flop D corrisponde al suo input

- Infatti, quando D=1, S=1 e R=0, quindi il prossimo stato del Flip-Flop (output) sarà Y=1
- Analogamente, quando D=0, S=0 e R=1, quindi il prossimo stato del Flip-Flop (output) sarà Y=0



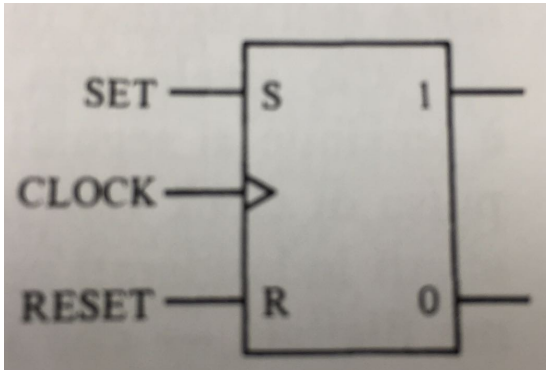
Flip-Flop D:
Tavola di verità

D	S	R	Stato attuale	Prossimo stato
1	1	0	0	1
1	1	0	1	1
0	0	1	0	0
0	0	1	1	0

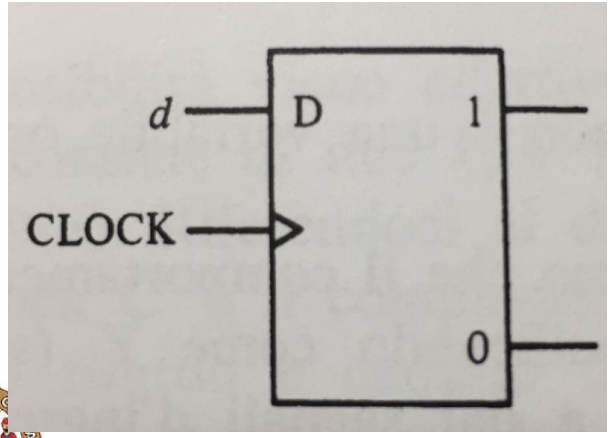
L'ouput (prossimo stato) del Flip-Flop D corrisponde al suo input

Vari tipi di Flip-Flop

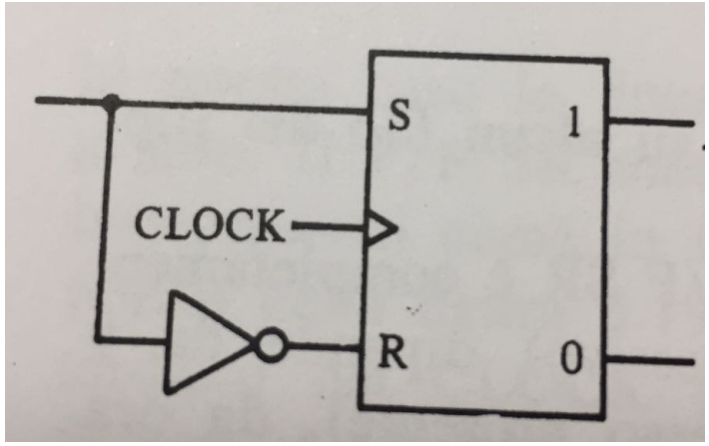
- Il simbolo circuitale del Flip-Flop S-R è



- Il simbolo circuitale del Flip-Flop D è

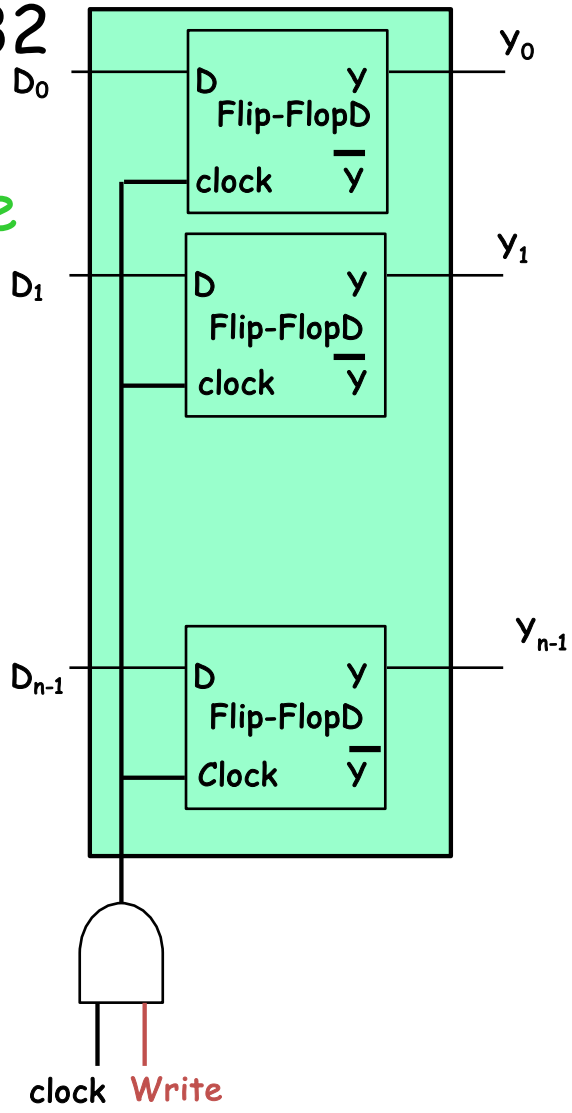


e corrisponde a



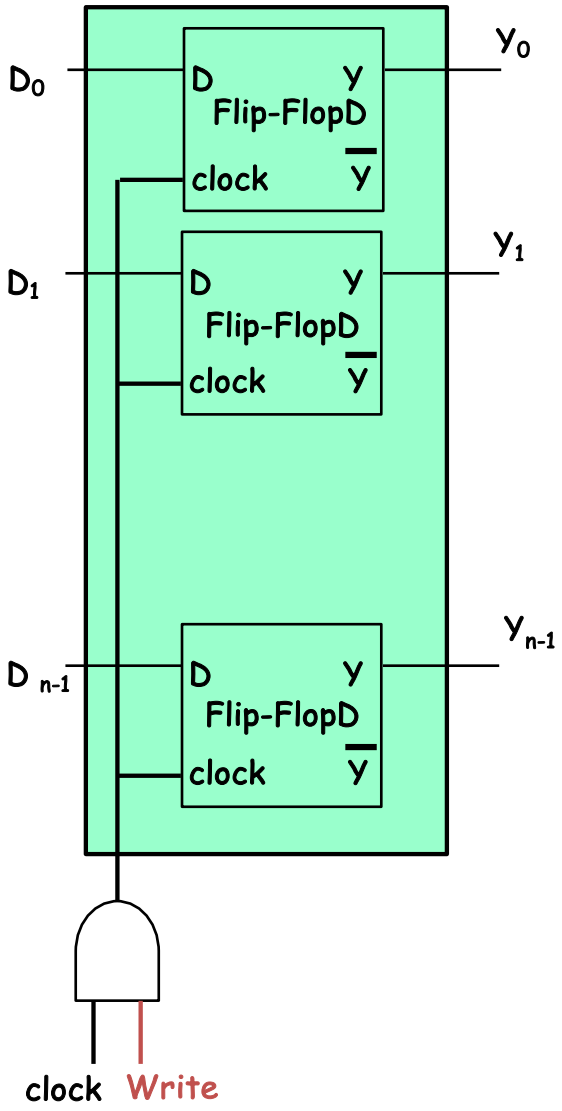
Registri

- Un registro nel MIPS è un array di n=32 Flip-Flop di tipo D connessi tra loro
- In un registro si può leggere o scrivere una stringa di n bit
  - La lettura (READ), non altera lo stato del registro
  - La scrittura (WRITE) invece si
- In aggiunta al segnale di Clock, c'è un segnale di Write
  - Se Write=0, lettura
  - Se Write=1, scrittura



Registri

- Se Clock=Write=1
  - Il clock di tutti i Flip-Flop D è 1 e l'output di ciascuno di essi corrisponde al suo input
    - Viene scritta la parola $D_0,\ldots,D_{n-1}$
- In tutti gli altri casi (Clock AND Write=0)
  - Lo stato di ciascun Flip-Flop D non viene modificato
  - Corrisponde alla lettura della parola $Y_0,\ldots,Y_{n-1}$



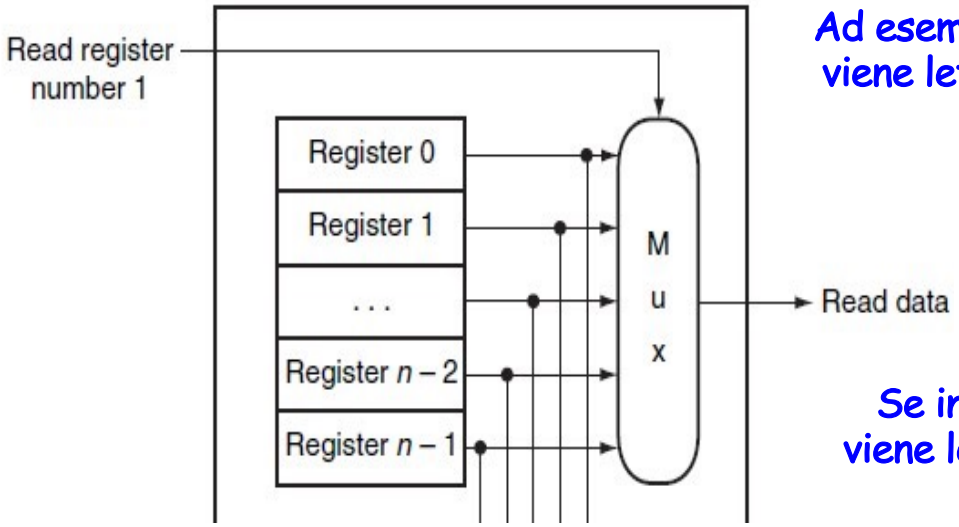
Banco di registri

- Detto anche Register File, costituisce l'insieme dei 32 registri del MIPS

- Ciascun registro è un array di 32 Flip-Flop di tipo D e può essere letto o scritto
  - Per leggere un registro occorre specificare in input il numero del registro (Register number, 5 bit), ottenendo in output il suo contenuto (Read data, 32 bit)
  - Per scrivere un registro occorre specificare in input il numero del registro (Register number, 5 bit) in cui scrivere e il dato da scrivere (Write data, 32 bit)

Leggere in un registro

- Realizzazione (n=32)
  - Usa un MUX 2^5:1 in cui il register number (5 bit) è usato per ottenere i 5 segnali di selezione del MUX
  - Gli input del MUX sono i contenuti dei 32 registri del banco
  - L'output del MUX è il contenuto del registro indicato



Multiplexer 2^5:1

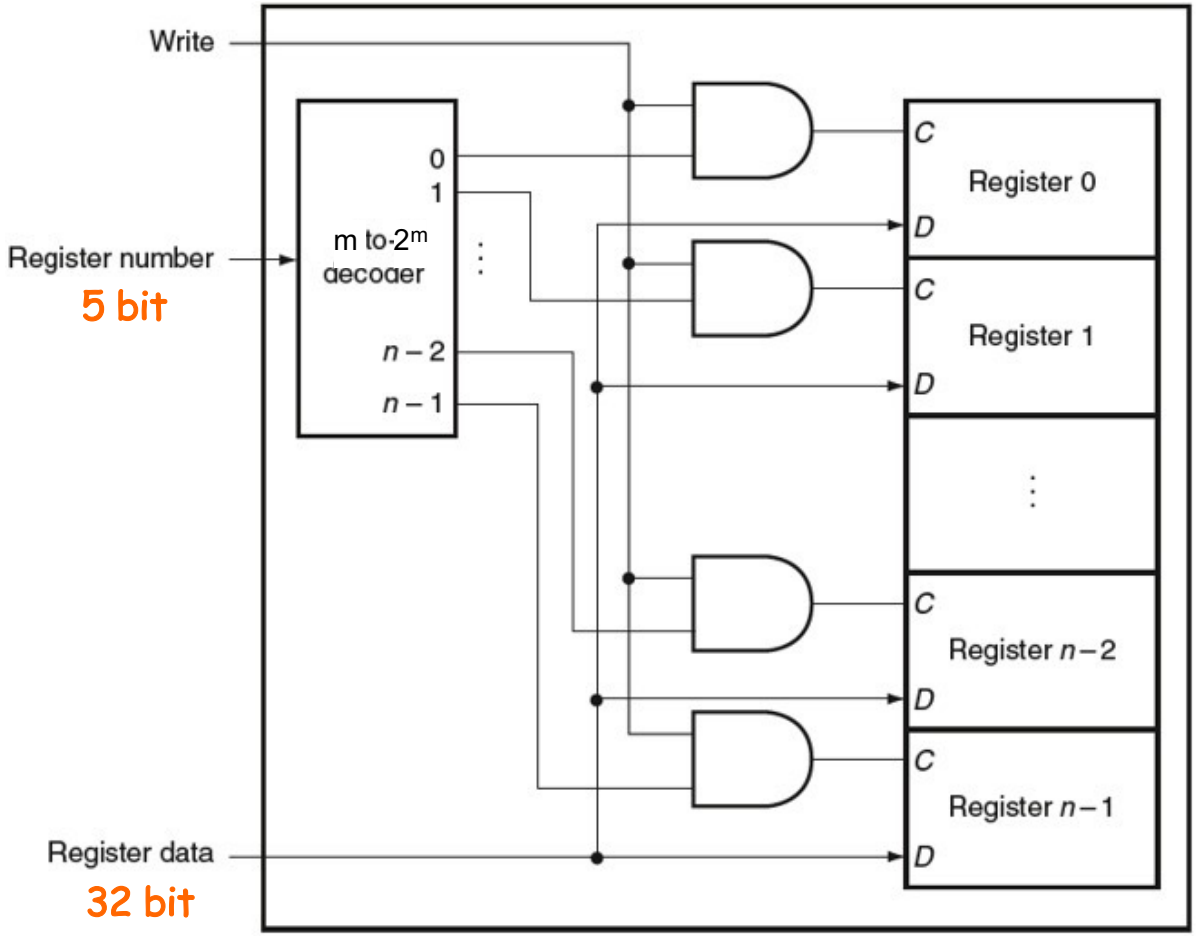
Ad esempio, se il register number è 00000, viene letto il contenuto del registro $zero

Se invece il register number è 11111, viene letto il contenuto del registro $ra

Scrivere in un registro

- Supponiamo di voler scrivere un dato all'interno di un certo registro
  - Ciò può accadere solo quando il segnale di Write è uguale a 1
- Dato in input il numero del registro (5 bit) e il dato da scrivere (32 bit)
  - Viene usato un decodificatore con 5 input (e 32 output) per determinare il registro in cui scrivere
  - L'output del decodificatore va in una porta AND insieme al segnale di Write ed abilita alla scrittura del registro corrispondente
  - I 32 bit del dato da scrivere diventano gli input dei 32 Flip-Flop di tipo D che formano il registro in cui scrivere, il cui stato viene modificato

Scrivere in un registro



Nel MIPS
$m=5$, $n=2^m=32$

Se il register number è 11111, il dato viene scritto nel registro $ra

Memoria

- I registri e il banco dei registri costituiscono i blocchi per piccole memorie all'interno del processore
- Maggiori quantità di memoria sono fornite tramite le SRAM e DRAM
  - Ne parleremo in seguito

Riepilogo e riferimenti

- Diagrammi temporali, latch S-R
  - [P] parr. 5.1, 5.2, 5.3
  - [PH] par. 4.2, Appendice B.7, B.8
- Flip-Flop S-R, D
  - [P] par. 5.5 oppure
  - [PH] par. 4.2, Appendice B.7, B.8
- Banco dei registri
  - [P] par 5.7 oppure [PH] appendice B.8