

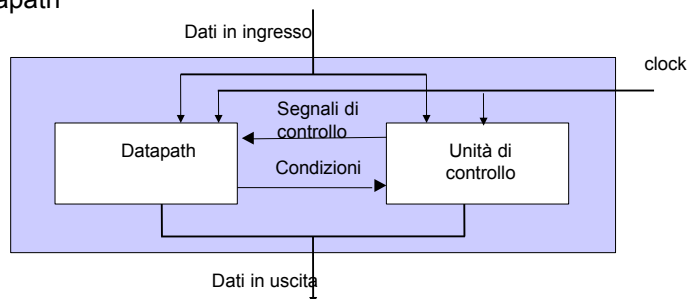
# Progettazione di un sistema digitale

Calcolatori Elettronici

## Progettazione di un sistema digitale

Può essere vista come la progettazione di un sistema formato da due componenti:

- Il datapath, che esegue le azioni richieste al sistema
- L'unità di controllo che, sulla base dei dati d'ingresso e delle condizioni prodotte dal datapath, genera i comandi per il datapath



## Progettazione del datapath

La progettazione del datapath può essere realizzata utilizzando dei componenti di libreria

Componenti tipici utilizzati per il datapath:

- Multiplexer
- Decoder
- Comparatori
- Alu
- Registri

## Progettazione del datapath

La progettare il datapath richiede, partendo dalle specifiche del sistema ovvero da ciò che il sistema deve realizzare, la realizzazione di uno schematico che definisce

- quali sono i componenti necessari;
- come sono tra loro collegati;
- quali sono le condizioni e i risultati prodotti;
- quali sono i segnali di controllo che devono essere prodotti dall'unità di controllo;

Nella progettazione del datapath è necessario tenere conto di eventuali vincoli di progetto quali ad esempio:

- Massima latenza
- Massima area
- Massima potenza

## Progettazione dell'unità di Controllo

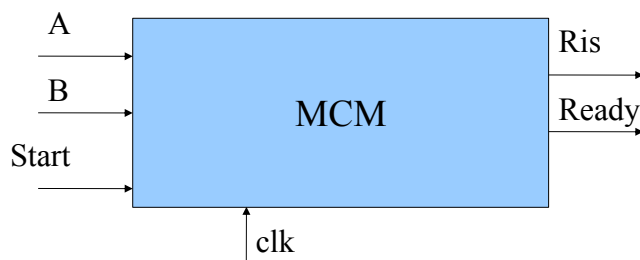
Progettare l'unità di controllo equivale a progettare una Macchina a stati finiti (FSM)

Individuati gli stati e i segnali di controllo per il datapath, la progettazione dell'unità di controllo può essere realizzata utilizzando i metodi di sintesi delle reti sequenziali sincrone

## Esempio: Minimo comune multiplo

Progettare un sistema digitale che dati due ingressi  $A > 0$  e  $B > 0$ , quando il segnale start diventa 1 calcola il minimo comune multiplo tra A e B.

Quando il risultato è disponibile, viene posto a 1 il segnale Ready



## Esempio: Minimo comune multiplo

Descriviamo mediante una specifica eseguibile il comportamento richiesto e selezioniamo i componenti che servono

```
while (true)
{ Ready='1';
  do
  while (start!='1');
  ma=A; mb=B; Ready='0';
  while (ma!=mb)
  if (ma<mb)
    ma=ma+A;
  else
    mb=mb+B;
  Ris=ma;
}
```

## Esempio: Minimo comune multiplo

Per realizzare il datapath servono:

Registri per memorizzare ma, mb e Ris

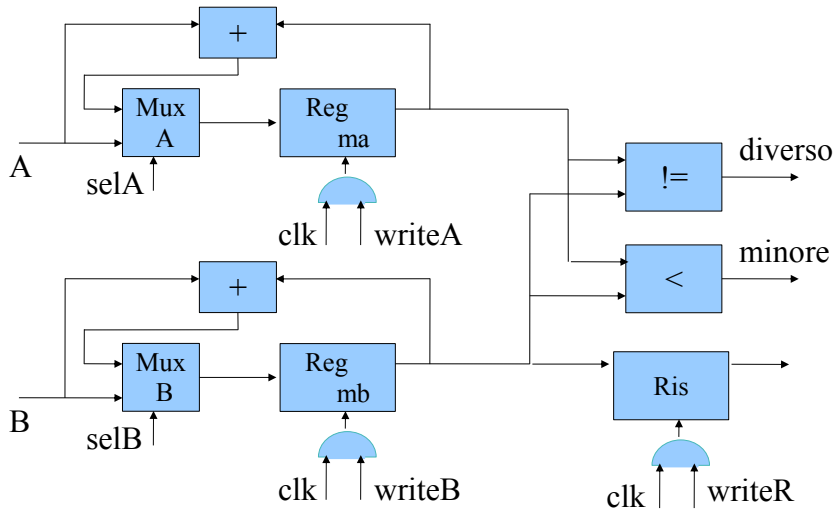
Comparatori per la condizione  $(A \neq B)$  e per  $(A < B)$

Sommatori per realizzare  $ma = ma + A$  e  $mb = mb + B$

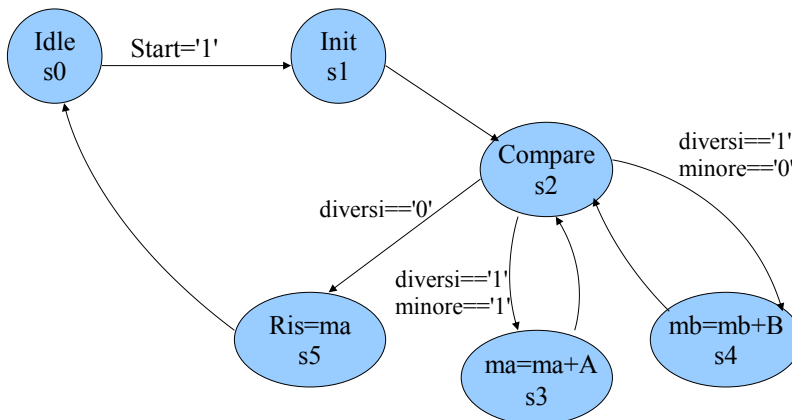
Multiplexer per selezionare l'ingresso dei registri ma (A o  $ma + A$ ) mediante SelA e mb (B o  $mb + B$ ) mediante SelB

Porte AND per combinare il clock con i segnale di abilitazione della scrittura in ma (WriteA), in mb (writeB) e in Ris (WriteR)

## Datapath Minimo comune multiplo



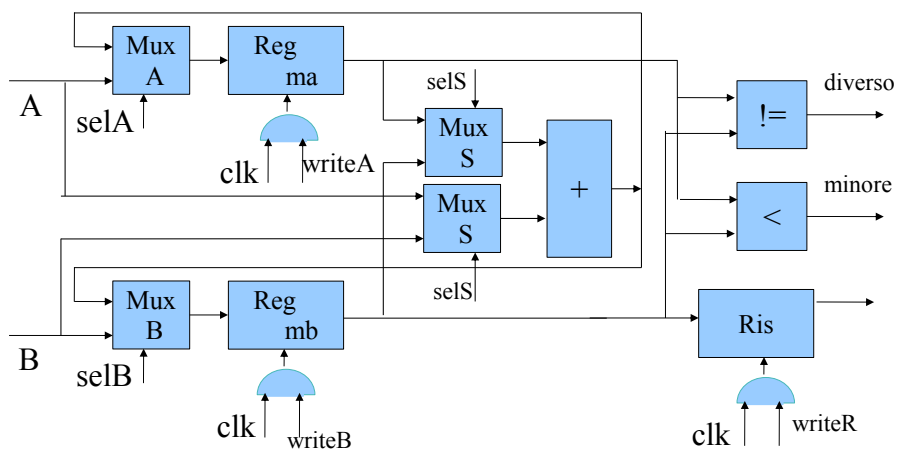
## FSM( macchina di moore): mcm



## FSM: uscite

	S0	S1	S2	S3	S4	S5
SelA	-	0	1	1	1	1
SelB	-	0	1	1	1	1
WriteA	0	1	0	1	0	0
WriteB	0	1	0	0	1	0
WriteR	0	0	0	0	0	1
Ready	1	0	0	0	0	0

## Datapath MCM con un solo sommatore

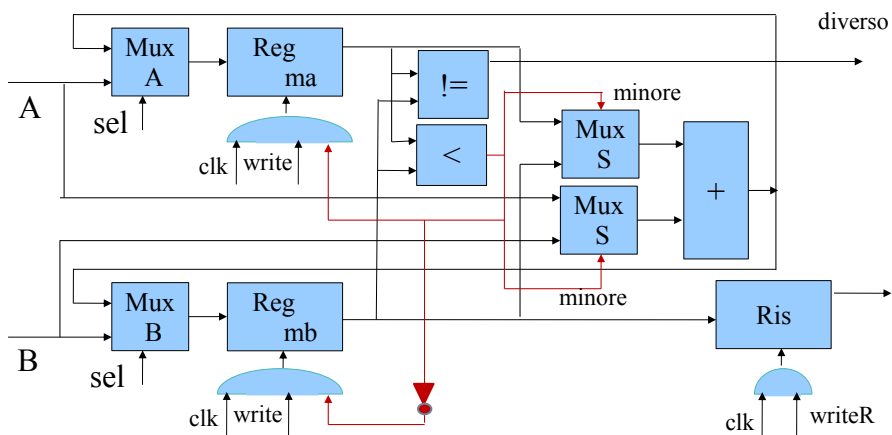


## FSM: uscite con un sommatore

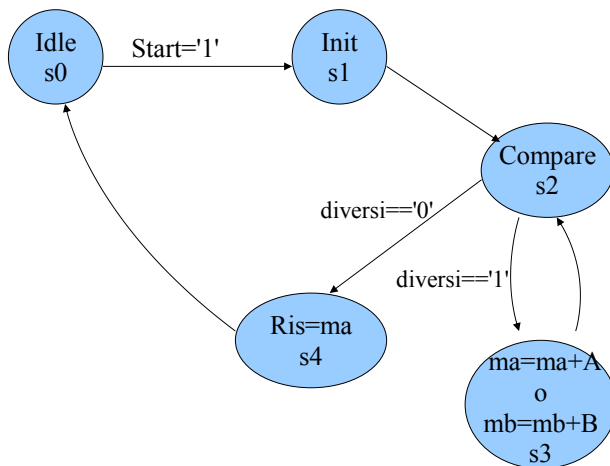
	S0	S1	S2	S3	S4	S5
SelA	-	0	1	1	1	1
SelB	-	0	1	1	1	1
SelS	-	-	-	0	1	-
WriteA	0	1	0	1	0	0
WriteB	0	1	0	0	1	0
WriteR	0	0	0	0	0	1
Ready	1	0	0	0	0	0

Questa soluzione riduce il costo, ma aumenta la latenza poiché viene introdotto un multiplexer nel percorso critico da Reg ma/Reg mb allo stesso registro

## Datapath MCM con un solo sommatore (v2)



## FSM( macchina di moore): mcm v2



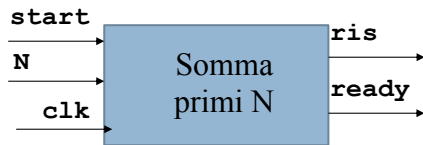
## FSM: uscite con un sommatore (v2)

	S0	S1	S2	S3	S4
Sel	-	0	1	1	1
Write	0	1	0	1	0
WriteR	0	0	0	0	1
Ready	1	0	0	0	0

Questa soluzione riduce il numero degli stati e dei segnali di controllo, ma aumenta ulteriormente la latenza poiché viene introdotto anche un comparatore (oltre aln multiplexer) nel percorso critico da Reg ma/Reg mb allo stesso registro



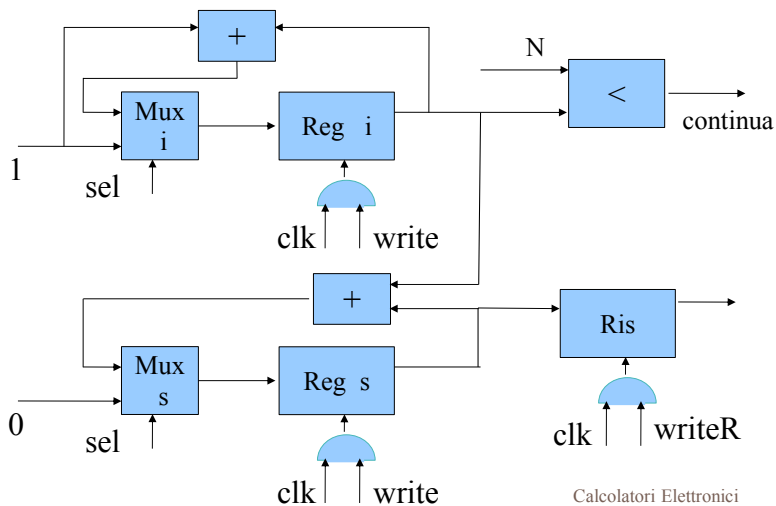
# Somma dei primi N numeri



```
While (1)
{ready=1;
 while(start!=1);
 ready=0;
 i=1; s=0
 while(i<=N)
 {s=s+i; i=i+1;}
 Ris=s;
}
```

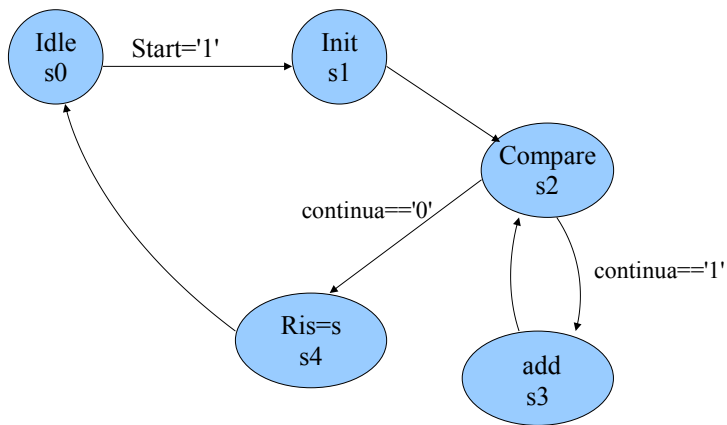
Calcolatori Elettronici

## Somma dei primi N numeri Datapath



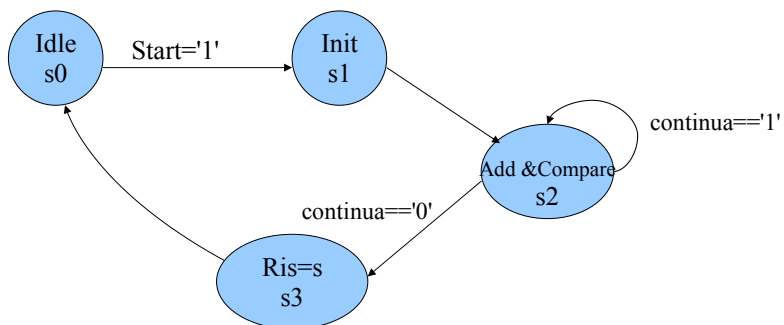
Calcolatori Elettronici

## Somma primi N numeri: FSM (versione 1)



Calcolatori Elettronici

## Somma primi N numeri: FSM (versione 2)

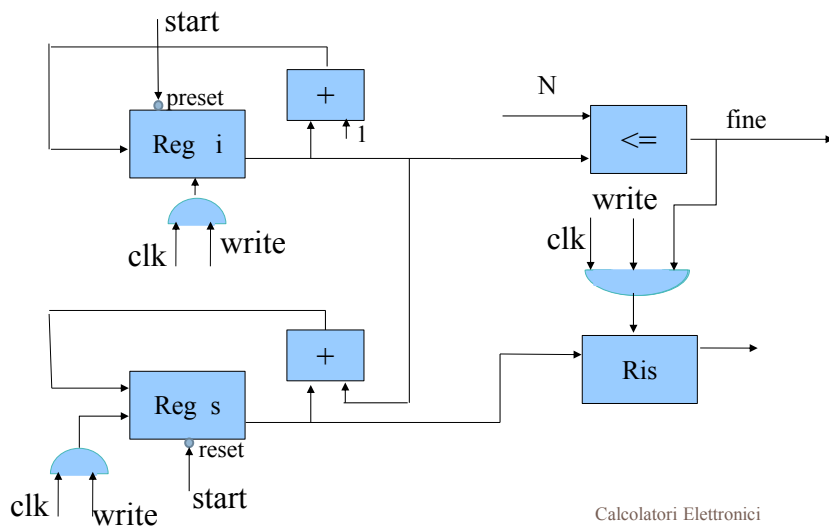


Calcolatori Elettronici

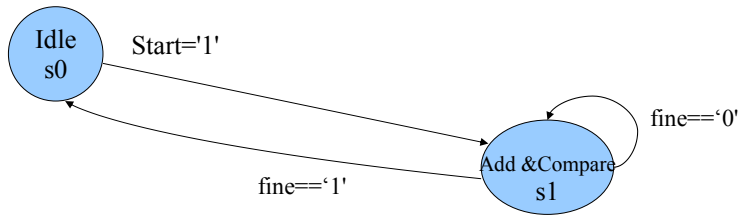
# Somma primi N numeri segnali di controllo( FSMvers.2):

	S0 Idle	S1 Init	S2 Add& Comp	S3 Ris
Sel	-	0	1	1
Write	0	1	1	0
WriteR	0	0	0	1
Ready	1	0	0	0

# Somma dei primi N numeri Datapath (versione 2)



## Somma primi N numeri: FSM (versione 3)



Calcolatori Elettronici

## Somma primi N numeri segnali di controllo( FSMvers.3):

	S0 Idle	S1 Add & Comp
Write	0	1
Ready	1	0