

# Lez. 6 – Progetto Controllore - Datapath

## Laboratorio di Architettura degli Elaboratori

Stefano Centomo

17-18 dicembre 2020

- 1 Unità funzionali di un datapath
- 2 Modellazione di una FSMD in SIS
- 3 Esercizi

- 1 Unità funzionali di un datapath
- 2 Modellazione di una FSMD in SIS
- 3 Esercizi

Per includere un componente all'interno di un modello blif usare le seguenti istruzioni:

```
.subckt nomeComp paramFormale=paramAttuale ...
```

```
.search nomeFileComp.blif
```

.latch <input> <output> <type> <control> <init-val>

- input è l'ingresso del latch.
- output l'uscita del latch.
- type può essere fe, re, ah, al, as, che corrispondono a “falling edge,” “rising edge,” “active high,” “active low,” or “asynchronous.”
- control è il segnale di clock per il latch. Può essere un clock del modello, l'uscita di una qualsiasi funzione del modello, o la parola “NIL” per nessun clock interno. Ciò significa che il registro utilizza il clock generale del circuito in cui è inserito.
- init-val è lo stato iniziale del latch, che può essere 0, 1, 2, 3. “2” indica “don't care” e “3” è “unknown.” o non specificato.

# Registro a 1 bit

```
.model REGISTRO  
.inputs A  
.outputs O  
.latch A 0 re NIL 0  
.end
```

# Registro a 4 bit

```
.model REGISTRO4
.inputs A3 A2 A1 A0
.outputs O3 O2 O1 O0
.subckt REGISTRO A=A3 O=O3
.subckt REGISTRO A=A2 O=O2
.subckt REGISTRO A=A1 O=O1
.subckt REGISTRO A=A0 O=O0
.search registro.blif
.end
```

```
.model SOMMATORE
.inputs A B CIN
.outputs O COUT
.names A B K
10 1
01 1
.names K CIN O
10 1
01 1
.names A B CIN COUT
11- 1
1-1 1
-11 1
.end
```



# Sommatore a 2 bit

```
.model SOMMATORE2
.inputs A1 A0 B1 B0 CIN
.outputs O1 O0 COUT
.subckt SOMMATORE A=A0 B=B0 CIN=CIN O=O0 COUT=C0
.subckt SOMMATORE A=A1 B=B1 CIN=C0 O=O1 COUT=COUT
.search sommatore.blif
.end
```

# Multiplexer a 4 ingressi 1 bit ciascuno

```
.model MUX1
.inputs S1 S0 i3 i2 i1 i0
.outputs out
.names S1 S0 i3 i2 i1 i0 out
001--- 1
01-1-- 1
10--1- 1
11---1 1
.end
```

# Multiplexer a 4 ingressi 2 bit ciascuno

```
.model MUX2
.inputs X1 X0 a1 a0 b1 b0 c1 c0 d1 d0
.outputs o1 o0
.subckt MUX1 S1=X1 S0=X0 i3=a1 i2=b1 i1=c1 i0=d1 out=o1
.subckt MUX1 S1=X1 S0=X0 i3=a0 i2=b0 i1=c0 i0=d0 out=o0
.search mux1.blif
.end
```

# Multiplexer a 2 ingressi da 3 bit ciascuno

```
.model MUX3
.inputs A2 A1 A0 B2 B1 B0 S
.outputs O2 O1 O0
.names S A2 B2 O2
11- 1
0-1 1
.names S A1 B1 O1
11- 1
0-1 1
.names S A0 B0 O0
11- 1
0-1 1
.end
```

# Demultiplexer a 1 bit e 4 uscite

```
.model DEMUX
.inputs S1 S0 IN
.outputs X Y Z W
.names S1 S0 IN X
001 1
.names S1 S0 IN Y
011 1
.names S1 S0 IN Z
101 1
.names S1 S0 IN W
111 1
.end
```

# Comparatore a 4 bit

```
.model UGUALE4
.inputs A3 A2 A1 A0 B3 B2 B1 B0
.outputs 0
.subckt xnor A=A3 B=B3 X=X3
.subckt xnor A=A2 B=B2 X=X2
.subckt xnor A=A1 B=B1 X=X1
.subckt xnor A=A0 B=B0 X=X0
.names X3 X2 X1 X0 0
1111 1
.search xnor.blif
.end
```

## “Maggiore” a 6 bit

```
.model 6_gt
.inputs A5 A4 A3 A2 A1 A0 B5 B4 B3 B2 B1 B0
.outputs AgtB
.subckt xor A=A5 B=B5 X=X5
.subckt xor A=A4 B=B4 X=X4
.subckt xor A=A3 B=B3 X=X3
.subckt xor A=A2 B=B2 X=X2
.subckt xor A=A1 B=B1 X=X1
.subckt xor A=A0 B=B0 X=X0
.names A5 A4 A3 A2 A1 A0 X5 X4 X3 X2 X1 X0 AgtB
1-----1----- 1
-1----01---- 1
--1---001--- 1
---1--0001-- 1
----1-00001- 1
-----1000001 1
.search xor.blif
```

## “Minore-uguale” a 6 bit

```
.model 6_le
.inputs C5 C4 C3 C2 C1 C0 D5 D4 D3 D2 D1 D0
.outputs CleD
.subckt 6_gt A5=C5 A4=C4 A3=C3 A2=C2 A1=C1 A0=C0 ...
          B5=D5 B4=D4 B3=D3 B2=D2 B1=D1 B0=D0 AgtB=z
.names z CleD
0 1
.search 6_gt.blif
.end
```



# Generazione di un valore costante a 0 (di un bit)

```
.model zero1  
.outputs uscita  
.names uscita  
.end
```

# Generazione di un valore costante a 1 (di un bit)

```
.model uno1  
.outputs uscita  
.names uscita  
1  
.end
```

# Generazione del valore costante dieci (su 4 bit)

```
.model dieci4  
.outputs 03 02 01 00  
.subckt uno1 uscita=03  
.subckt zero1 uscita=02  
.subckt uno1 uscita=01  
.subckt zero1 uscita=00  
.end
```

# Outline

- 1 Unità funzionali di un datapath
- 2 Modellazione di una FSMD in SIS**
- 3 Esercizi

# Modellazione di una FSMD in SIS

- 1 Modellare la FSM del Controllore mediante la tabella delle transizioni e assegnare la codifica agli stati con `state_assign` `jedi`
- 2 Salvare il Controllore ottenuto dopo la codifica degli stati in un file diverso da quello originale usando il comando `write_blif nomefile`
- 3 Modellare il Datapath come una interconnessione di componenti funzionali quali sommatori, moltiplicatori, multiplexer, registri, ecc.
- 4 Inglobare il Controllore e il Datapath in un unico file blif come se fossero due componenti, collegando opportunamente i segnali di comunicazione. E' importante specificare con la direttiva `.search` il file del Controllore ottenuto dopo la codifica degli stati e non quello scritto a mano.

## Esempio: Semaforo con Priorità Temporizzato

Si consideri il classico esempio di un semaforo posto all'incrocio tra una strada principale (direttrice Nord-Sud, NS) e una secondaria (direttrice Est-Ovest, EO). Il semaforo può assumere solo i colori verde e rosso ed è dotato di sensori che rilevano la presenza di traffico. Il circuito che controlla il semaforo ha 2 segnali di ingresso forniti dai sensori (TRAFFICONS e TRAFFICOEO) e due di uscita (LUCENS e LUCEEO) con il seguente significato:

- TRAFFICONS[1]: segnala la presenza di traffico lungo la direttrice NS assumendo il valore 1
- TRAFFICOEO[1]: segnala la presenza di traffico lungo la direttrice EO assumendo il valore 1
- LUCENS[1]: deve essere posto a 1 per accendere la luce verde sulla strada NS. Se viene posto a 0 si accende la luce rossa.
- LUCEEO[1]: deve essere posto a 1 per accendere la luce verde sulla strada EO. Se viene posto a 0 si accende la luce rossa.

## Esempio: Semaforo con Priorità Temporizzato

Per evitare incidenti, il circuito di controllo deve garantire che le luci sulle strade NS e EO siano sempre accese in opposizione. Il circuito assegna priorità alla strada NS e commuta dal verde al rosso su NS solo se  $\text{TRAFFICONS}=0$  e  $\text{TRAFFICOEO}=1$ , in caso contrario mantiene il verde su NS. In assenza di traffico sia su NS che su EO il semaforo non modifica la configurazione raggiunta. Non appena giunge traffico su NS, indipendentemente da cosa succede su EO, il semaforo assegna la luce verde a NS e la luce rossa a EO.

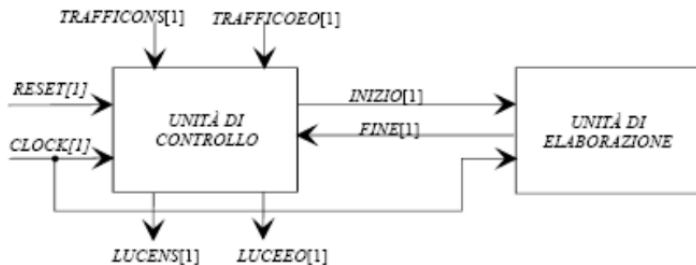
## Esempio: Semaforo con Priorità Temporizzato

Per evitare incidenti, il circuito di controllo deve garantire che le luci sulle strade NS e EO siano sempre accese in opposizione. Il circuito assegna priorità alla strada NS e commuta dal verde al rosso su NS solo se  $\text{TRAFFICONS}=0$  e  $\text{TRAFFICOEO}=1$ , in caso contrario mantiene il verde su NS. In assenza di traffico sia su NS che su EO il semaforo non modifica la configurazione raggiunta. Non appena giunge traffico su NS, indipendentemente da cosa succede su EO, il semaforo assegna la luce verde a NS e la luce rossa a EO.

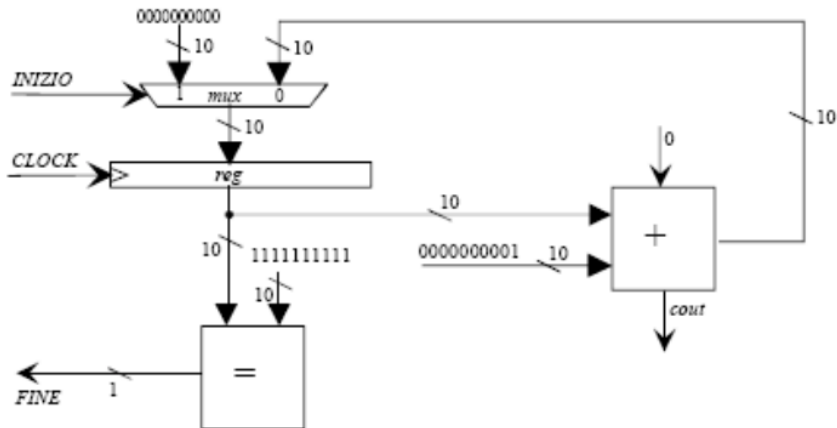
**Il controllore non commuta mai un semaforo da verde a rosso senza aver atteso 1024 cicli di clock.**



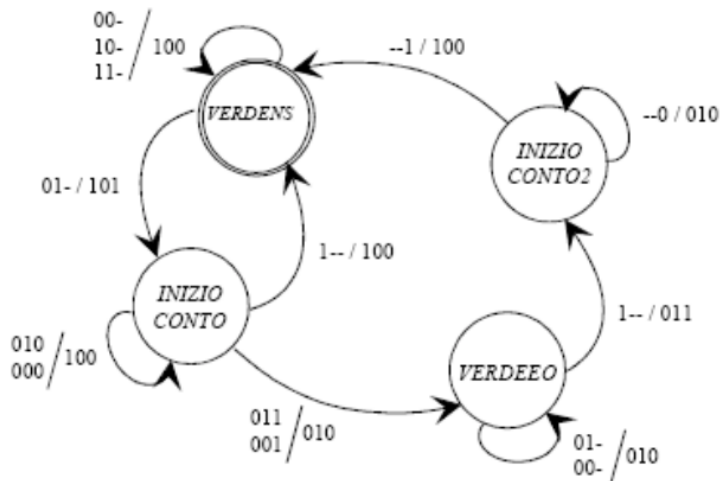
# Esempio: Semaforo con Priorità Temporizzato



# Esempio: Semaforo con Priorità Temporizzato



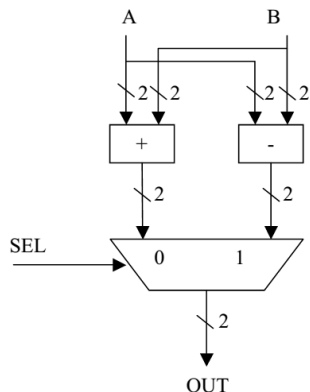
# Esempio: Semaforo con Priorità Temporizzato



- 1 Unità funzionali di un datapath
- 2 Modellazione di una FSM in SIS
- 3 Esercizi

# Esercizio 1

Realizzare in formato blif il seguente datapath. A e B sono due numeri positivi da 2 bit. Se  $SEL=0$  allora OUT fornisce il valore della somma tra A e B; se  $SEL=1$  allora OUT fornisce il valore della differenza tra A e B.



Realizzare in formato `.blif` la FSMD del circuito che controlla l'apertura di una cassaforte. Gli ingressi del circuito sono:

- `APRI[1]`: vale 1 quando viene premuto il bottone invio per aprire la cassaforte. In tal caso il circuito deve controllare che il valore di `NUM` sia uguale a "0110" per aprire la cassaforte.
- `NUM[4]`: è il codice a 4 bit della combinazione.

Le uscite del circuito sono:

- `APERTA[1]`: viene impostato a 1 quando viene premuto il bottone invio e `NUM=0110`. Viene impostato a 0 quando viene premuto il bottone chiudi per richiudere la cassaforte oppure dopo che sono trascorsi 4 cicli di clock da quando la cassaforte è stata aperta.