

Reti combinatorie

- Rappresentano l'implementazione di funzioni 'pure' (ossia **senza stato**) a livello hardware
- Rete logica con **n** ingressi binari ed **m** uscite binarie
- Ad ogni combinazione di valori di ingresso corrisponde **una ed una sola** combinazione di valori di uscita

Struttura di un elaboratore Architettura degli elaboratori - I Pagina 37

Reti combinatorie (segue)

- La specifica di una funzione logica da implementare mediante rete combinatoria può essere vista come un **programma**
- La progettazione diventa combinazione e complemento di reti già note
- Componenti standard
 - ✓ Confrontatore, commutatore, selezionatore

Struttura di un elaboratore Architettura degli elaboratori - I Pagina 38

Reti combinatorie (segue)

- **Confrontatore**, a due ingressi (x,y) ed una uscita (z)
 - ✓ $z := \text{not } (x = y)$
- **Commutatore**, a due ingressi primari (x,y), un ingresso di controllo (α) ed una uscita (z)
 - ✓ $z := \text{if not } \alpha \text{ then } x \text{ else } y$
- **Selettore**, ad un ingresso primario (x), un ingresso di controllo (α) e due uscite (z_1, z_2)
 - ✓ $\text{if not } \alpha \text{ then } (z_1 := x ; z_2 := 0) \text{ else } (z_1 := 0 ; z_2 := x)$

Struttura di un elaboratore Architettura degli elaboratori - I Pagina 39

Reti combinatorie (segue)

- Operatori aritmetico logici a specifica **diretta**
 - ✓ Addizione, sottrazione, traslazione, rotazione, incremento, decremento, etc.
- Reti aritmetico logiche **multi-funzione**
 - ✓ Eseguono **una** delle operazioni suddette a seconda del valore assunto da un certo numero di variabili di controllo
 - ✓ Si usano per implementare le **ALU** (*arithmetic logic unit*)

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 40

Reti combinatorie (segue)

- Esempio:
 - ✓ m ingressi primari (x_1, \dots, x_m), n ingressi di controllo ($\alpha_1, \dots, \alpha_n$), una uscita (z)

```

z := case  $\alpha_1 \alpha_2 \dots \alpha_n$ 
      when 00...0 =>  $x_1$ 
      when 00...1 =>  $x_2$ 
      ...
      when 11...1 =>  $x_m$ 
    end case
    
```

- ✓ Gli ingressi di controllo sono tipicamente (combinazioni di) bit singoli

Struttura di un elaboratore

Architettura degli elaboratori -1

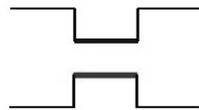
Pagina 41

Reti combinatorie (segue)

- Vediamo alcuni esempi di circuiti:
 - ✓ I segnali sono discretizzati e di solito assumono solo due stati:

0 / FALSO / [0..1] Volt

1 / VERO / [2..5] Volt



- ✓ I circuiti piu' complessi sono realizzati attraverso la combinazione di circuiti semplici (porte logiche)

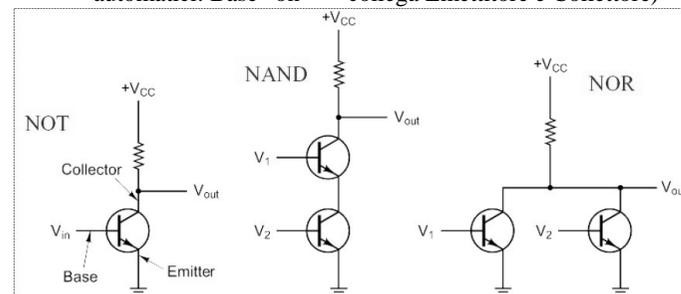
Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 42

Reti combinatorie (segue)

- Porte Logiche:
 - ✓ Sono realizzate tramite transistor (sono in pratica interruttori automatici: Base "on" → collega Elettore e Collettore)



Struttura di un elaboratore

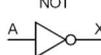
Architettura degli elaboratori -1

Pagina 43

Reti combinatorie (segue)

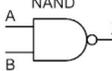
- Le porte logiche di base sono 5:

NOT



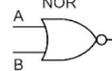
A	X
0	1
1	0

NAND



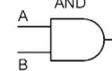
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR



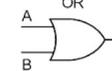
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

AND



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

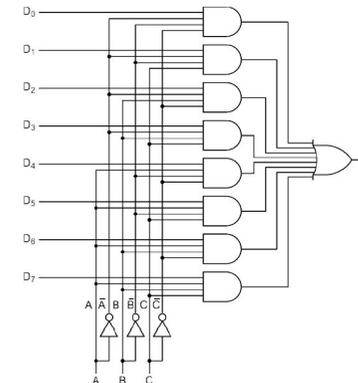
OR



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 44

Reti combinatorie (segue)

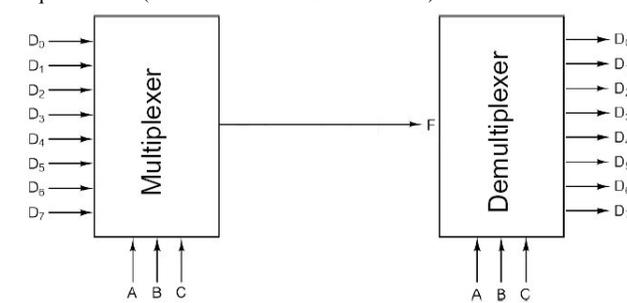


- Multiplexer:**
 - ✓ Risolve il problema di portare verso l'esterno i segnali di un circuito usando solo poche "vie di uscita" (piedini di un chip)
 - ✓ 2ⁿ linee di input (D₀ - D₇)
 - ✓ n linee di controllo (A,B,C)
 - ✓ 1 linea di output (F)

Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 45

Reti combinatorie (segue)

- Demultiplexer:**
 - ✓ E' il circuito inverso del Multiplexer ed è spesso usato in combinazione con quest'ultimo (seleziona comunicazione fra linee)



Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 46

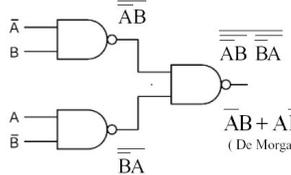
Reti combinatorie (segue)

- Comparatore:**
 - ✓ Compara due ingressi e produce un output che indica la uguaglianza o meno degli ingressi
 - ✓ Esempio di comparatore ad 1 bit: si realizza con porte XOR



XOR (OR Esclusivo)
Simbolo ⊕

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0



Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 47

Reti combinatorie (segue)

- Comparatore a più bit
 - ✓ Comparatori ad 1 bit vengono collegati tramite una porta NOR
 - ✓ L'output vale 1 solo se tutti gli output dei singoli comparatori ad 1 bit valgono 0

Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 48

Reti combinatorie (segue)

- Circuiti Aritmetici: Traslatore (shifter)
 - ✓ Trasla i bit in ingresso (D) di una posizione, a sinistra o a destra a seconda del valore del bit di controllo (C) (nuovo bit impostato a 0)

Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 49

Reti combinatorie (segue)

- Circuiti Aritmetici: Sommatore (Adder)
 - ✓ Sommatore di un bit (half-adder)

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 50

Reti combinatorie (segue)

- Circuiti Aritmetici: Sommatore
 - ✓ L'half-adder non può essere usato per sommare n bit a causa di un possibile riporto:

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

+

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Posso usare l'half-adder

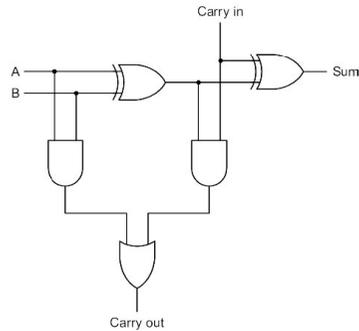
Non posso usare l'half-adder, perché ho un riporto intermedio

Struttura di un elaboratore
Architettura degli elaboratori -1
Pagina 51

Reti combinatorie (segue)

- Circuiti Aritmetici: Sommatore
 - ✓ Full-adder a 1 bit, prevede un possibile riporto

A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Struttura di un elaboratore

Architettura degli elaboratori - 1

Pagina 52

Reti combinatorie (segue)

- Circuiti Aritmetici: Sommatore a n bit
 - ✓ Si collegano n full-adder a 1 bit fra di loro
 - ✓ Il carry out del bit i-esimo deve essere collegato al carry in del bit i+1-esimo ($i=0, \dots, n-2$)
- Circuiti Aritmetici: Arithmetic Logic Unit (ALU)
 - ✓ Circuito capace di eseguire operazioni aritmetiche e logiche elementari su due parole (di n bit) di input A e B (es. AND, OR, complemento, somma aritmetica,...)

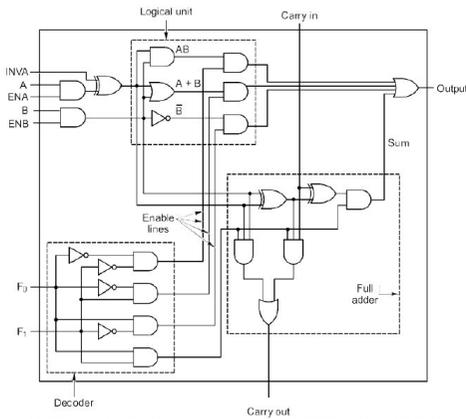
Struttura di un elaboratore

Architettura degli elaboratori - 1

Pagina 53

Reti combinatorie (segue)

- ALU ad 1bit che realizza 4 operazioni (selezionate da F_0 e F_1)



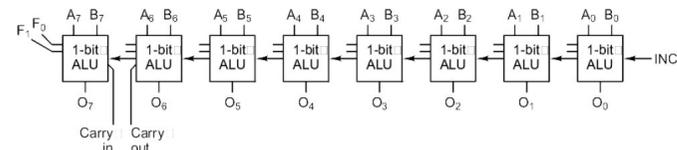
Struttura di un elaboratore

Architettura degli elaboratori - 1

Pagina 54

Reti combinatorie (segue)

- ALU a n bit
 - ✓ Si ottiene concatenando n ALU ad 1 bit
 - ✓ F_0 e F_1 collegati a tutte le ALU
 - ✓ Riporto intermedio propagato da una ALU alla successiva
 - ✓ INC (corrispondente al carry in della ALU "0") permette di sommare 1 al risultato in caso di addizione



Struttura di un elaboratore

Architettura degli elaboratori - 1

Pagina 55

Reti sequenziali

- Rappresentano l'implementazione di funzioni con stato, ossia di **automi a stati finiti**
- Strumenti di espressione (specifica) della **parte controllo** e della **parte operativa** di unità di elaborazione

Struttura di un elaboratore

Architettura degli elaboratori - I

Pagina 56

Reti sequenziali (segue)

- Un automa a stati finiti è una macchina caratterizzata da:
 - ✓ **n** variabili logiche di **ingresso** (2^n stati di ingresso X_i)
 - ✓ **m** variabili logiche di **uscita** (2^m stati di uscita Z_i)
 - ✓ **k** variabili logiche di **stato interno** (2^k stati interni S_i)
 - ✓ una **funzione σ di transizione dello stato interno**
 - $\sigma : X \times S \rightarrow S$
 - ✓ una **funzione ω delle uscite**
 - $\omega : X \times S \rightarrow Z$

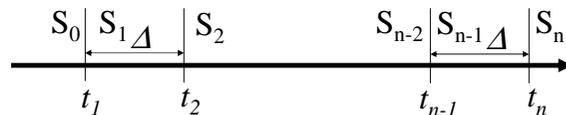
Struttura di un elaboratore

Architettura degli elaboratori - I

Pagina 57

Reti sequenziali (segue)

- In un modello **ideale** di rete sequenziale di tipo **sincrono**, le variazioni di stato ($S_i \rightarrow S_{i+1}$) avvengono in corrispondenza degli istanti di una sequenza temporale **discreta** ($t_1, t_2, \dots, t_n, \dots$) di intervallo **costante** ($\Delta = t_n - t_{n-1}$)



Struttura di un elaboratore

Architettura degli elaboratori - I

Pagina 58