Circuiti Combinatori

Componenti e Applicazioni

Corso di Informatica

IIS Fermi Sacconi Ceci

27 novembre 2025

Indice

Cosa sono i Circuiti Combinatori?

Definizione

Un **circuito combinatorio** è un sistema di porte logiche dove:

- Le uscite dipendono **solo** dagli ingressi attuali
- Non c'è memoria dello stato precedente
- Il comportamento è descritto da funzioni logiche
- Sono circuiti **senza memoria**

Caratteristiche:

- n ingressi, m uscite
- Funzione: $F : 2^n \rightarrow 2^m$
- Non considerano il tempo di propagazione

Analisi e Sintesi

Analisi:

- Dato il circuito
- Determinare la funzione
- Ricavare tabella di verità

Sintesi:

- Data la funzione desiderata
- Progettare il circuito
- Ottimizzare il numero di porte

Forme di specifica:

1. Descrizione verbale
2. Tabella di verità
3. Espressione analitica
4. Schema logico

Decodificatore (Decoder)

Definizione

Un decodificatore attiva **una sola uscita**, selezionandola in base alla combinazione degli ingressi.

Caratteristiche:

- n ingressi di selezione
- 2^n uscite
- Una sola uscita attiva alla volta
- Spesso presente un ingresso di abilitazione (enable)

Notazione:

- Ingressi: $a_0, a_1, ..., a_{n-1}$
- Uscite: $y_0, y_1, ..., y_{2^n-1}$
- Enable: en

Decodificatore 2-to-4

Tabella di verità:

en	a_1	a_0	y_3	y_2	y_1	y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Equazioni logiche:

$$y_0 = en \cdot \overline{a_1} \cdot \overline{a_0}$$

$$y_1 = en \cdot \overline{a_1} \cdot a_0$$

$$y_2 = en \cdot a_1 \cdot \overline{a_0}$$

$$y_3 = en \cdot a_1 \cdot a_0$$

Applicazioni dei Decodificatori

- **Selezione di memoria**: indirizzamento di celle
- **Display a 7 segmenti**: conversione binario-decimale
- **Demultiplexing**: instradamento di dati
- **Generazione di funzioni logiche**: ogni uscita è un mintermine

Esempio pratico:

- Decodificatore 3-to-8 per selezionare 1 tra 8 dispositivi
- Usato nei sistemi di memoria e I/O

Demultiplatore (Demux)

Definizione

Un demultiplatore **instrada** un singolo ingresso dati verso una delle 2^n uscite, selezionata dagli ingressi di selezione.

Differenza con il decodificatore:

- Decodificatore: uscita attiva (1) o inattiva (0)
- Demultiplatore: uscita = valore dell'ingresso dati

Componenti:

- n ingressi di selezione
- 1 ingresso dati (d)
- 2^n uscite
- Ingresso di abilitazione (opzionale)

Demultiplatore 1-to-4

Tabella di verità:

en	a_1	a_0	d	y_3	y_2	y_1	y_0
0	X	X	d	0	0	0	0
1	0	0	d	0	0	0	d
1	0	1	d	0	0	d	0
1	1	0	d	0	d	0	0
1	1	1	d	d	0	0	0

Nota: L'uscita selezionata riporta il valore di d

Multiplatore (Mux)

Definizione

Un multiplatore **seleziona** uno tra 2^n ingressi e lo riproduce nell'uscita, in base agli ingressi di selezione.

Caratteristiche:

- 2^n ingressi dati $(d_0, d_1, ..., d_{2^n-1})$
- n ingressi di selezione
- 1 uscita (y)
- Spesso un ingresso di abilitazione

Analogia:

- Funziona come un **selettore rotativo**
- Commutatore controllato digitalmente

Multiplatore 4-to-1

Tabella di verità:

en	a_1	a_0	d_3	d_2	d_1	d_0	y
0	X	X	d_3	d_2	d_1	d_0	0
1	0	0	d_3	d_2	d_1	d_0	d_0
1	0	1	d_3	d_2	d_1	d_0	d_1
1	1	0	d_3	d_2	d_1	d_0	d_2
1	1	1	d_3	d_2	d_1	d_0	d_3

Equazione:

$$y = en \cdot (\overline{a_1} \cdot \overline{a_0} \cdot d_0 + \overline{a_1} \cdot a_0 \cdot d_1 + a_1 \cdot \overline{a_0} \cdot d_2 + a_1 \cdot a_0 \cdot d_3)$$

Applicazioni dei Multiplatori

- **Selezione di dati**: scegliere tra diverse sorgenti
- **Implementazione di funzioni logiche**: ogni mux può implementare qualsiasi funzione booleana
- **Trasmissione dati**: multiplexing di canali
- **Operazioni condizionali**: selezione basata su condizioni

Esempio: Un mux 4-to-1 può implementare qualsiasi funzione a 2 variabili:

- Variabili di selezione = variabili della funzione
- Ingressi dati = valori della tabella di verità

Multiplatori e Demultiplatori in Parallelo

Necessità

Per gestire dati a più bit (bus), servono più mux/demux in parallelo.

Configurazione:

- Un mux/demux per ogni bit del dato
- Ingressi di selezione **condivisi**
- Ingresso di abilitazione **condiviso**
- Esempio: 4 mux 4-to-1 per dati a 4 bit

Rappresentazione compatta:

- Linee multiple indicate con numero
- Bus rappresentati come linee spesse

Codificatore Binario (Encoder)

Definizione

Un codificatore **traduce** l'ingresso selezionato in un numero binario nelle uscite.

Funzionamento inverso del decodificatore:

- 2^n ingressi
- n uscite
- Una sola linea di ingresso attiva per volta
- Output = codice binario dell'ingresso attivo

Esempio 4-to-2:

w_3	w_2	w_1	w_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Codificatore di Priorità (Priority Encoder)

Differenza

Il codificatore di priorità **ammette più ingressi attivi** simultaneamente, emettendo il codice dell'ingresso con priorità più alta.

Caratteristiche:

- Priorità: ingresso con indice più basso/alto
- Gestisce più ingressi attivi
- Spesso include uscita di validità (z)

Esempio 4-to-2 con priorità (indice basso = alta priorità):

w_3	w_2	w_1	w_0	z	y_1	y_0
0	0	0	0	0	X	X
X	X	X	1	1	0	0
X	X	1	0	1	0	1
X	1	0	0	1	1	0
1	0	0	0	1	1	1

Unità Logiche

Definizione

Un'unità logica esegue **operazioni logiche** (AND, OR, NOT, XOR) su coppie di bit.

Implementazione con multiplatori:

- Ingressi: due operandi (a, b)
- Ingresso di selezione: tipo di operazione (f)
- Uscita: risultato dell'operazione

Esempio unità logica a 4 funzioni:

f_1	f_0	Operazione
0	0	a AND b
0	1	a OR b
1	0	a XOR b
1	1	NOT b

Unità Logiche su Interi

Per operare su numeri a più bit:

- Unità logiche semplici in parallelo
- Una per ogni coppia di bit
- Controllo condiviso (f)
- Operazione bit per bit

Esempio: 4 bit

- 4 unità logiche identiche
- Input: $a_3 a_2 a_1 a_0$ e $b_3 b_2 b_1 b_0$
- Output: $y_3 y_2 y_1 y_0$
- $y_i = f(a_i, b_i)$

Scorrimento (Shift)

Definizione

Lo scorrimento sposta le cifre binarie a sinistra o a destra.

Tipi di scorrimento:

1. **Logico**: inserisce 0 dal lato opposto
2. **Aritmetico**: mantiene il segno (destra)
3. **Circolare**: la cifra espulsa rientra dall'altro lato
4. **Circolare con riporto**: usa il carry

Effetti:

- Scorrimento a sinistra = moltiplicazione per 2
- Scorrimento a destra = divisione per 2

Scorrimento Logico

Scorrimento a sinistra:

- Bit più significativo $\rightarrow$ perso
- 0 inserito a destra
- Esempio: $1011 \rightarrow 0110$

Scorrimento a destra:

- Bit meno significativo $\rightarrow$ perso
- 0 inserito a sinistra
- Esempio: $1011 \rightarrow 0101$

Implementazione:

- Multiplatori a 2 ingressi per ogni bit
- Selezione: direzione dello scorrimento

Scorrimento Aritmetico

Caratteristiche:

- Mantiene il bit di segno (bit più significativo)
- Usato per numeri con rappresentazione in complemento a due
- A sinistra: come lo scorrimento logico (inserisce 0)
- A destra: duplica il bit di segno

Esempi:

- Scorrimento aritmetico a destra: $1011 \rightarrow 1101$
- Scorrimento aritmetico a sinistra: $1011 \rightarrow 0110$

Scorrimento Circolare

- Il bit espulso rientra dall'altro lato
- Nessuna perdita di informazione

Esempio:

- Sinistra: $1011 \rightarrow 0111$
- Destra: $1011 \rightarrow 1101$

Grazie per l'attenzione