Fabio Fassetti

Reti Logiche e Calcolatori

Lezione 1

Introduzione al corso

Il corso mira a fornire solide basi in merito al funzionamento del livello logico-digitale, mediante l'acquisizione di capacità di analisi e di sintesi delle reti logiche combinatorie e sequenziali, all'organizzazione ed al funzionamento della macchina calcolatore, mediante l'acquisizione delle tecniche di microprogrammazione, ed alla conoscenza del linguaggio di programmazione assembly.

Il corso è anche su Facebook

Gruppo su Reti Logiche e Calcolatori

Contatti

Fabio Fassetti

Contatti: Cubo 41C - III piano, fabio.fassetti@unical.it

Ricevimento

- in via telematica, sulla piattaforma TEAMS
- durante le lezioni previste dall'orario
- martedì pomeriggio dalle 15:00 alle 17:00
- su appuntamento
- quando mi trovate on line :)

Introduzione al corso

Conoscenze preliminari

- Numeri binari, complemento a 2, codifica di numeri reali
- Rudimenti di logica proposizionale

Materiale didattico

- Fabrizio Lucci, Linda Pagli - Reti Logiche e Calcolatore - Bollati Boringhieri Editore
- Manuale di programmazione Intel
- Appunti di lezione

Introduzione al corso

Metodi di valutazione

- L'esame consiste in una prova scritta e in una prova orale obbligatoria.
- È prevista la possibilità di conservare lo scritto per una sessione, ossia è possibile sostenere la prova orale o nella stessa sessione della prova scritta o nella sessione successiva.

Algebra Booleana

Algebra Booleana

Rappresenta lo strumento matematico alla base dell'analisi e della sintesi di circuiti logici / reti logiche

- Costanti: 0, 1
- Variabili: x, y, z, … con valore in {0,1}
- Funzioni: $z = f(x1, ..., xn) => \{0,1\}^n -> \{0,1\}$
- Rappresentazioni: Tabella di verità, Espressione algebrica

Algebra Booleana

x	$z = f1(x)$
0	1
1	0

NOT

$$f1(x) = \overline{x}$$

$x1$	$x2$	$z = f2(x1,x2)$
0	0	0
0	1	1
1	0	1
1	1	1

OR

$$f2(x1,x2) = x1 + x2$$

$x1$	$x2$	$z = f3(x1,x2)$
0	0	0
0	1	0
1	0	0
1	1	1

AND

$$f3(x1,x2) = x1 \cdot x2$$

Algebra Booleana

Proprietà.

Ogni funzione booleana ammette un'espressione algebrica costruita utilizzando gli operatori NOT, AND, OR e le parentesi.

$$f(x1,x2,x3) = x1 \cdot \overline{x2} + x3$$

Priorità degli operatori

$\bar{\ }$, $\cdot$, $+$

Algebra Booleana - Proprietà

0. $\overline{\overline{x}} = x$

OR

1. $x + 0 = x$
2. $x + 1 = 1$
3. $x + \overline{x} = x$
4. $x + x = 1$
5. $x + y = y + x$ *[commutativa]*
6. $x + (y + z) = (x + y) + z$ *[associativa]*
7. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ *[distributiva]*

Algebra Booleana

Proprietà duale

Data una proprietà P, si definisce duale di P la proprietà P’ ottenuta da P sostituendo

- AND con OR e viceversa
- 0 con 1 e viceversa

Principio di dualità

Se una proprietà P è valida nell’algebra booleana, allora è anche valida la sua duale.

Algebra Booleana - Proprietà

0. $\overline{\overline{x}} = x$

OR

1. $x + 0 = x$
2. $x + 1 = 1$
3. $x + \overline{x} = 1$
4. $x + x = x$
5. $x + y = y + x$ *[commutativa]*
6. $x + (y + z) = (x + y) + z$ *[associativa]*
7. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ *[distributiva]*

AND

1. $x \cdot 1 = x$
2. $x \cdot 0 = 0$
3. $x \cdot \overline{x} = 0$
4. $x \cdot x = x$
5. $x \cdot y = y \cdot x$ *[commutativa]*
6. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ *[associativa]*
7. $x + (y \cdot z) = (x + y) \cdot (x + z)$ *[distributiva]*

Algebra Booleana - Teorema di De Morgan

$$\overline{x1 + x2} = \overline{x1} \cdot \overline{x2}$$

$$\overline{x1 \cdot x2} = \overline{x1} + \overline{x2}$$

Forme Canoniche

Ci consentono di passare dalle tabelle di verità alle espressioni algebriche.

x1	x2	x3	z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Forme Canoniche

Ci consentono di passare dalle tabelle di verità alle espressioni algebriche.

x1	x2	x3	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Forme Canoniche

MINTERMINE

funzione logica che vale 1 in corrispondenza di una e una sola configurazione degli ingressi e 0 altrimenti

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6
0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0
1	0	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	0

Forme Canoniche

MINTERMINE

- definizione
  - funzione logica che vale 1 in corrispondenza di una e una sola configurazione degli ingressi e 0 altrimenti
- espressione algebrica
  - AND di tutte le variabili prese dirette se valgono 1 nella configurazione per cui il mintermine vale 1, negate se valgono 0

Forme Canoniche

MAXTERMINE

funzione logica che vale 0 in corrispondenza di una e una sola configurazione degli ingressi e 1 altrimenti

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6	s1	s3	s7
0	0	0	1	1	0	0	0	0	1	1	1
0	0	1	0	0	0	0	0	0	0	1	1
0	1	0	1	0	1	0	0	0	1	1	1
0	1	1	0	0	0	0	0	0	1	0	1
1	0	0	1	0	0	1	0	0	1	1	1
1	0	1	1	0	0	0	1	0	1	1	1
1	1	0	1	0	0	0	0	1	1	1	1
1	1	1	0	0	0	0	0	0	1	1	0

Forme Canoniche

MAXTERMINE

- definizione
  - funzione logica che vale 0 in corrispondenza di una e una sola configurazione degli ingressi e 1 altrimenti
- espressione algebrica
  - OR di tutte le variabili prese dirette se valgono 0 nella configurazione per cui il maxtermine vale 0, negate se valgono 1

Forme Canoniche

Ogni funzione f può essere rappresentata mediante un’espressione algebrica in forma canonica

- PRIMA FORMA CANONICA (SOMMA DI PRODOTTI) SP
  - OR di tutti i mintermini associati alle configurazioni in cui f vale 1

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6
0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0
1	0	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	0

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6	p0 + p2 + p4 + p5 + p6
0	0	0	1	1	0	0	0	0	
0	0	1	0	0	0	0	0	0	
0	1	0	1	0	1	0	0	0	
0	1	1	0	0	0	0	0	0	
1	0	0	1	0	0	1	0	0	
1	0	1	1	0	0	0	1	0	
1	1	0	1	0	0	0	0	1	
1	1	1	0	0	0	0	0	0	

$$z = p0 + p2 + p4 + p5 + p6$$

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6	p0 + p2 + p4 + p5 + p6
0	0	0	1	1	0	0	0	0	1
0	0	1	0	0	0	0	0	0	
0	1	0	1	0	1	0	0	0	
0	1	1	0	0	0	0	0	0	
1	0	0	1	0	0	1	0	0	
1	0	1	1	0	0	0	1	0	
1	1	0	1	0	0	0	0	1	
1	1	1	0	0	0	0	0	0	

z = p0 + p2 + p4 + p5 + p6

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6	p0 + p2 + p4 + p5 + p6
0	0	0	1	1	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	
0	1	1	0	0	0	0	0	0	
1	0	0	1	0	0	1	0	0	
1	0	1	1	0	0	0	1	0	
1	1	0	1	0	0	0	0	1	
1	1	1	0	0	0	0	0	0	

$$z = p0 + p2 + p4 + p5 + p6$$

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6	p0 + p2 + p4 + p5 + p6
0	0	0	1	1	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	0	0	
1	0	0	1	0	0	1	0	0	
1	0	1	1	0	0	0	1	0	
1	1	0	1	0	0	0	0	1	
1	1	1	0	0	0	0	0	0	

z = p0 + p2 + p4 + p5 + p6

Forme Canoniche

x1	x2	x3	z	p0	p2	p4	p5	p6	p0 + p2 + p4 + p5 + p6
0	0	0	1	1	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	1
1	0	1	1	0	0	0	1	0	1
1	1	0	1	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0

$$z = p0 + p2 + p4 + p5 + p6$$

Forme Canoniche

Ogni funzione f può essere rappresentata mediante un’espressione algebrica in forma canonica

- SECONDA FORMA CANONICA (PRODOTTO DI SOMME) PS
  - AND di tutti i maxtermini associati alle configurazioni in cui f vale 0

Forme Canoniche

x1	x2	x3	z	s1	s3	s7	s1 · s3 · s7
0	0	0	1	1	1	1	1
0	0	1	0	0	1	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	0	0

$$z = s1 \cdot s3 \cdot s7$$

Insiemi di operatori

Un insieme di operatori si dice funzionalmente completo se ogni funzione può essere rappresentata attraverso mediante i soli operatori dell'insieme.

- $\{+, \cdot, \bar{\ }\}$ è un insieme funzionalmente completo
- $\{\cdot, \bar{\ }\}$ è un insieme funzionalmente completo
- $\{+, \bar{\ }\}$ è un insieme funzionalmente completo
- $\{\cdot, +\}$ non è un insieme funzionalmente completo
    - non è possibile ottenere il *not* attraverso l'uso di *and* e *or*

Operatori NAND e NOR

x1	x2	x1 · x2	x1 | x2
0	0	0	
0	1	0	
1	0	0	
1	1	1	

NAND

x1	x2	x1 + x2	x1 □ x2
0	0	0	
0	1	1	
1	0	1	
1	1	1	

NOR

Operatori NAND e NOR

x1	x2	x1 · x2	x1 | x2
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NAND

x1	x2	x1 + x2	x1 □ x2
0	0	0	
0	1	1	
1	0	1	
1	1	1	

NOR

Operatori NAND e NOR

x1	x2	x1 · x2	x1 | x2
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NAND

x1	x2	x1 + x2	x1 □ x2
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NOR

Non vale la proprietà associativa!

Quesito

Quante funzioni distinte di n variabili possono essere definite?