



Politecnico di Milano
Corsi di Laurea in Ingegneria Matematica
e Ingegneria Fisica
Dipartimento di Elettronica ed Informazione

Logica e codifica binaria dell'informazione

Anno Accademico **2002 –2003**

Indice

- Logica proposizionale
- Operatori Logici AND, OR, NOT,...
- Rappresentazione binaria, complemento a 2
- Codifica dei numeri reali in virgola fissa e mobile
- Codifica dei caratteri

Logica proposizionale

- Formalizzazione del linguaggio naturale
- Alfabeto costituito da:
 - Lettere proposizionali
 - Simboli: AND, OR, NOT, OGNI, APPARTENENZA,...

AND	\wedge
OR	\vee
NOT	\neg
OGNI	\forall
APPARTENENZA	\in
ESISTENZA	\exists

- Simboli ausiliari (parentesi,...)

Tavole di verità

(per determinare il valore di verità di una formula composta)

X	Y	$X \wedge Y$	$X \vee Y$	$X \Rightarrow Y$	$X \Leftrightarrow Y$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

Insiemi completi

- Le porte logiche AND, OR, NOT costituiscono un insieme funzionalmente completo, cioè sono in grado di realizzare qualsiasi funzione.
- Altri esempi di insiemi funzionalmente completi sono:
 - NAND
 - NOR

Tecnologie Hardware

- Rappresentazione dell'informazione (caratteri, testo, immagini, suoni, video,...) in forma binaria
- Basata sull'astrazione di catene di un valore elementare che può essere solo "VERO" o "FALSO" (oppure "0" o "1") = bit.
- Un dispositivo HW numerico (digitale) rappresenta i bit tramite diversi possibili "valori elettrici".
- Un calcolatore è costituito da un insieme di circuiti che consentono memorizzare, trasferire ed elaborare i bit.

Reti Logiche

- Una rete logica è costituita dalla combinazione di circuiti logici.
- L'output di una rete logica è sempre un valore logico (0/1).
- Esistono reti logiche
 1. *Combinatorie*: l'uscita dipende solo dagli ingressi
 2. *Sequenziali*: l'uscita dipende dagli ingressi e dallo stato del sistema.

Codifica degli interi

- **Unsigned int** il cui valore si calcola **come** (Es. caso 16 bit)

$$x = \sum_{i=0}^{15} x_i \times 2^i$$

Nota: il bit di destra è il meno significativo (LSB), quello a sinistra è il più significativo (MSB)

- Si può usare il primo bit per rappresentare il segno (rappresentazione in ampiezza e segno). Poco usata.

There are 10 types of
people on earth:
who understand binary
and who don't...

Codifica degli interi (2)

- Rappresentazione in “**complemento a 2**”
 - Primo bit a sinistra è *INDICATIVO* del segno
 - Numeri positivi: come in “ampiezza e segno”
 - Numeri negativi: algoritmo (vedi dopo...)
- Vantaggi:
 1. una sola rappresentazione dello zero,
 2. struttura ciclica,
 3. somme tra numeri di segno opposto si ottengono operando meccanicamente bit per bit senza preoccuparsi del segno (ignoro riporti a sx)

CPL2: conversioni dei numeri negativi

- **Da DECIM. a CPL2**
 - Scrivo il numero POSITIVO in binario (*mettere uno 0 davanti per ricordarsi del bit di segno*)
 - Complemento tutti i bit
 - Sommo 1
- **Da CPL2 a DECIM.**
 - Complemento tutti i bit
 - Decodifico da binario a decimale
 - Sommo 1
 - Cambio il segno

Osservazioni

(sul complemento a 2)

- Minimo numero rappresentabile è $10000\dots000000$, che risulta maggiore di una unità rispetto al max numero rappresentabile
- $-1 = 11111\dots1_2$
- Su 16 bit: $MAX = 2^{15} - 1$; $min = -2^{15}$
- Su 8 bit: $MAX = 2^7 - 1$; $min = -2^7$

Altre rappresentazioni

- **Esadecimale**: risulta comodo usare rappresentazione che considera pacchetti di 4 bit. Ciascun pacchetto risulta codificabile con intero 0..15.
 - Per i numeri tra 10 e 15 si usano le cifre **ESADECIMALI A, B, C, D, E, F.**
- Es.** $0100101100100110_2 = 4B26_{16}$
- Esiste anche la rappresentazione ottale.

Codifica dei razionali

- Rappresentati con precisione finita (reali trasformati in razionali).
- Due tipi diversi di rappresentazione:
 - In virgola fissa (fixed-point)
 - In virgola mobile (floating-point)

Parte frazionaria

- $P_{10} = 0 + d_{-1} * 10^{-1} + d_{-2} * 10^{-2} + \dots$
- $P_2 = 0 + b_{-1} * 2^{-1} + b_{-2} * 2^{-2} + \dots$
- La conversione da binario a decimale è come quella vista per i numeri naturali, ma con le potenze della base negative.
- Per la conversione da decimale a binario l'algoritmo di conversione è "duale" rispetto a quello per la parte intera. In questo caso si opera per moltiplicazioni successive (per la base 2) della sola parte frazionaria. Le cifre (bit) ottenute sono quelle dei riporti sulle unità intere.

Virgola fissa

- Analoga alla rappresentazione dei numeri interi.
- Viene stabilita a priori la posizione del punto decimale: le cifre si dividono in “parte intera” e “parte decimale”.

Esempio: 16 bit, unsigned, 3 bit per la parte intera.

$$x = \sum_{i=-13}^2 x_i \times 2^i$$

Si ottiene moltiplicando l'intero senza la virgola per 2^{-13}

Virgola mobile, singola precisione (IEEE 754)

- Il numero rappresentato deve essere espresso nel formato $1, m \times 2^p$
- Il numero viene memorizzato su 32 bit (partendo dal meno significativo) :
 - Bit 31: segno
 - Bit 23-30: bit di esponente
 - Bit 22-0: mantissa in rappresentazione binaria



Virgola mobile IEEE 754 (2)

- Nel campo esponente si codifica il valore $p+2^{(n_e-1)}-1$, dove n_e è il numero di bit usati per l'esponente
 - In questo modo non devo usare una notazione modulo e segno o CPL2
- Nel campo mantissa si rappresenta SOLO la parte frazionaria della mantissa (0,m)
 - L' "1" è sottinteso: risparmio 1 bit
- Per la doppia precisione: 11 bit per l'esponente e 52 per la mantissa.

Osservazioni

- Sia la rappresentazione in virgola fissa che quella mobile sono “granulari” (sottoinsieme dei razionali).
- **Virgola fissa:** uniformemente distribuiti tra min e MAX (passo = 2^{-d}).
- **Virgola mobile:** distribuzione non uniforme (più radi per valori elevati dell'esponente e più fitti per valori piccoli).

Codifica dei caratteri

- Perché i caratteri alfanumerici possano essere rappresentati all'interno di un calcolatore è necessario fornirne una rappresentazione binaria.
- Esistono diverse convenzioni.
“Problemi” per gestire i set estesi (localizzazione)
 - ASCII
 - Unicode
 - [Altre: BCD, EBCDIC,...]

ASCII

- American Standard Code for Information Interchange. Sviluppato da ANSI.
- Ogni carattere (inglese) è rappresentato da un numero a 7 bit (0-127).
- Usato dai sistemi UNIX, DOS(-based).
- ***Extended ASCII***: 8 bit. 128 char aggiuntivi per caratteri non-inglesi.

Unicode

*“Unicode provides a unique number for every character,
no matter what the platform,
no matter what the program,
no matter what the language”.*

- Comprende set di caratteri di cui alcuni in codifica multibyte che coprono quasi tutte le lingue scritte del mondo.
- Usato dai sistemi Windows NT/2000.
- Esistono funzioni standard ANSI per leggere testo Unicode.