*Elettronica *

*Digitale*



*Elettronica *

*Digitale*



Aritmetica binaria

L'aritmetica binaria utilizza due simboli, lo 0 e l' 1, per rappresentare i numeri in forma binaria.

0 ed 1 vengono denominati 'bit' (binary digit=cifra binaria).

Come per l'aritmetica decimale, anche quella binaria costituisce un sistema di rappresentazione numerico posizionale, in cui ciascuna cifra assume un valore opportuno, dipendentemente dalla posizione occupata e dal peso relativo ad essa.

Il peso, relativo ad una certa posizione, é l'esponente a cui occorre elevare la base 2, per ottenere il valore della posizione stessa.

Come, in un numero a base decimale, le unità occupano, dalla destra alla sinistra del numero, il primo posto, le decine il secondo, le centinaia il terzo, e così via, i bits in un numero a base binaria occupano analoghe posizioni (anche se non con la stessa denominazione).



Per convertire un numero decimale in uno binario, occorre dividerlo per la base 2, tante volte finchè non é più possibile la divisione, riportando a sinistra i quozienti parziali, a destra i resti (che sono 0 oppure 1), quindi leggendo il numero in forma binaria, dal basso verso l'alto.



Per la conversione da binario in decimale, occorre moltiplicare ciascuna cifra, a cominciare da quella più a destra (cifra meno significativa=least significant bit=LSB), per la potenza di base 2 elevata al peso relativo a quella posizione (iniziando dal peso 0 e da destra), fino all'ultima cifra a sinistra (cifra più significativa=most significant bit=MSB), quindi sommando tutti i prodotti effettuati.

Per sapere quante cifre binarie occorrono per rappresentare un numero in base decimale, occorre considerare l'esponente della base 2 che permette di raggiungere il numero voluto; per es., per codificare 64, occorrono 7 cifre binarie, perchè 2 elevato a 7 é uguale a 128, per cui 64 rientra in tale numero ( non 6 cifre binarie, come si potrebbe pensare, perchè la numerazione parte da 0, non da 1).

Un metodo semplice di conversione é quello di considerare i valori posizionali con i rispettivi pesi e posizionare tanti 1 o 0, in modo da avere complessivamente il numero, in forma decimale, voluto.

Le operazioni dell'aritmetica binaria

Le operazioni fondamentali includono l'addizione, la sottrazione, la moltiplicazione, la divisione, il complemento ad uno, il complemento a due.

Le regole sono le seguenti:

addizione: 0+0=0 ; 0+1=1 ; 1+0=1 ; 1+1=0 con riporto di 1;

sottrazione: 0-0=0 ; 0-1= non si può, di presta una 'duina' dalla cifra binaria precedente ; 1-0=1 ; 1-1=0;

moltiplicazione: 0x0=0 ; 0x1=0 ; 1x0=0 ; 1x1=1;

divisione: 0:0=0 ; 0:1=0 ; 1:0=non é possibile ; 1:1=1;

complemento ad uno: 0 cambia in 1 ; 1 cambia in 0 (per tutte le cifre del numero binario);

complemento a due: si addiziona 1 alla cifra meno significativa del complemento ad uno del numero binario.

Si noti che, con la complementazione a due, una sottrazione può essere eseguita come una addizione.







La logica digitale riguarda la verifica della verità o falsità delle proposizioni componenti una proposizione complessa.

Si costituisce attraverso tre proposizioni logiche fondamentali: la congiunzione, la disgiunzione, la negazione.

La congiunzione logica tra due o più proposizioni risulta vera, se tutte le proposizioni di partenza sono vere contemporaneamente.

La disgiunzione logica, invece, risulta vera, se almeno una delle proposizioni di partenza é vera.

La negazione logica indica l'opposto di ciò che si asserisce.

Nel campo delle applicazioni elettroniche, la logica proposizionale si traduce in funzioni logiche di commutazione ed in circuiti logici combinatori composti da porte logiche.

Si può chiarire meglio il concetto di funzione logica ricorrendo all'analogia elettrica.

Le funzioni logiche corrispondenti alla congiunzione, disgiunzione e negazione sono le funzioni 'AND', 'OR', 'NOT'.

Le proposizioni di partenza sono rappresentate da variabili di ingresso (indipendenti), quella globale da variabili d'uscita (dipendenti).

Gli esempi che seguono sono relativi a funzioni a due variabili d'ingresso e ad una variabile d'uscita.

Il concetto di logica digitale combinatoria si riferisce ad analisi logiche in cui il tempo non ha influenza sulla commutazione delle funzioni, per cui le uscite dei circuiti combinatori non dipendono dai valori assunti precedentemente, ma variano unicamente in presenza di nuove combinazioni degli ingressi.







Altre funzioni possono essere derivate da quelle fondamentali: 'NAND' - 'NOR' - 'EXOR' - 'EXNOR'.

La funzione 'NAND' é costituita dalla funzione 'AND' seguita da una 'NOT'.

La funzione 'NOR' é costituita dalla funzione 'OR' seguita da una 'NOT'.

La funzione 'EXOR' é costituita dalla funzione 'OR', in cui la combinazione degli ingressi entrambi alti fornisce un'uscita bassa.

La funzione 'EXNOR' é costituita dalla funzione 'EXOR' seguita da una 'NOT'.





Algebra di Boole

Proprietà e teoremi

Proprietà della somma

x1+x2=x2+x1 Proprietà commutativa

x1+x2+x3=x1+(x2+x3)=(x1+x2)+x3 Proprietà distributiva

x+0=x

x+1=1

x1·x2+x1·x3=x1·(x2+x3) Proprietà distributiva della somma rispetto al prodotto

Proprietà del prodotto

x1·x2=x2·x1 Proprietà commutativa

x1·x2·x3=x1·(x2·x3)=(x1·x2)·x3 Proprietà distributiva

x·0=0

x·1=x

x1·(x2+x3)=x1·x2+x1·x3 Proprietà distributiva del prodotto rispetto alla somma

Proprietà della negazione

0=1 ; 1=0 ; 0=0; 1=1

x=x

x+x=1

x·x=0

Teoremi di idempotenza

x+x+x=x

x·x·x=x

x1+(x1·x2)=x1+x2 (1) (1) 0+(1·0)=0 (2) 0·(1+0)=0

1+(0·0)=1 1·(0+0)=0

x1·(x1+x2)=x1·x2 (2) 0+(1·1)=1 0·(1+1)=0

1+(0·1)=1 1·(0+1)=1

Teoremi di De Morgan

I° Teorema

x1+x2=x1·x2

II° Teorema

x1·x2=x1+x2

Algebra di Boole

Metodi di minimizzazione delle reti logiche combinatorie

Metodo delle forme canoniche

Data una funzione logica ad n variabili, essa è rappresentabile mediante una tavola di verità contenente un numero di righe pari alle combinazioni di valori delle n variabili (2n).

Considerando una riga della tavola di verità di una data funzione logica, si indica con il termine “mintermine”, il prodotto tra le variabili di ingresso, della riga, che assumono il valore “1” ed i complementi delle variabili che assumono valore “0”.

Si indica, invece, con “maxtermine”, la somma tra le variabili di ingresso di valore “0” ed i complementi delle variabili che assumono valore “1”.

Esempio:

X=A·B M1=A+B

M2=A+B

M3=A+B

m4=A·B



I forma canonica

Una funzione logica può essere ridotta, considerando la somma di tutti i mintermini che determinano un livello “1”, in uscita, per la funzione stessa.

II forma canonica

Una funzione logica può essere ridotta, considerando il prodotto di tutti i maxtermini che determinano un livello “0”, in uscita, per la funzione stessa.

Metodo delle mappe di Karnaugh

Data la tavola di verità di una determinata funzione logica e stabilita la prima forma canonica della funzione, si può rappresentare il circuito logico relativo a tale funzione, semplificato, costruendo la relativa mappa di karnaugh ed individuando eventuali raggruppamenti di caselle adiacenti.

Per caselle adiacenti si intendono quelle che differiscono per uno ed uno solo stato delle variabili di ingresso.

Una mappa di Karnaugh è un insieme di caselle, il cui numero è determinato da 2n, con n uguale al numero di variabili di ingresso.

All’interno delle caselle, ciascuna delle quali è indicata dalle relative variabili di ingresso, si scrivono i valori di uscita “1” dei mintermini, in corrispondenza delle relative variabili di ingresso.

La funzione di commutazione è costituita dalla somma di tanti mintermini, quanti sono i raggruppamenti di stati “1” che si possono avere tra caselle adiacenti.

Ciascun mintermine è costituito dal prodotto logico delle variabili di ingresso, che rimangono al valore “1”, nel passaggio da una casella a quella adiacente del raggruppamento e dei negati delle variabili di ingresso, che rimangono al valore “0”.

Rappresentazione di porte logiche mediante altre porte

AND, OR, NOT, EXOR, EXNOR con porte NAND

AND, OR, NOT, EXOR, EXNOR con porte NOR

Circuiti AOI (AND-OR-NOT)

Applicazioni di reti logiche combinatorie

Comparatore binario

Semiaddizionatore (half-adder)

Addizionatore completo

Multiplexer

Demultiplexer

Encoder

Decoder

Generatore di parità

Rivelatore di parità

Famiglie di integrazione

SSI

MSI

LSI

VLSI

Tecnologie costruttive

RTL, DTL

TTL

CMOS

ECL

Parametri porte logiche

Livelli logici

Ritardo di propagazione

Potenza assorbita

Fan-in, fan-out

Immunità al rumore

Interfacciamento

Elettronica

Digitale

*Logica Sequenziale*

I circuiti logici sequenziali sono caratterizzati da uscite che commutano dipendentemente dai valori assunti in precedenza dalle stesse, cioè, si potrebbe dire che tali uscite ricordano i valori precedenti, fintantochè non interviene un evento esterno che le faccia commutare.

Per comprendere tale funzionamento, si analizza un circuito logico sequenziale, elementare per eccellenza, il quale si può dire costituisca un elemento di memorizzazione di un singolo bit.

Si tratta del LATCH S-R.

E' costituito da due porte NOR collegate come in figura, in cui viene effettuata una reazione incrociata tra le uscite e gli ingressi, che porta ad ottenere risultati diversi da una comune porta logica combinatoria.

In tale circuito si devono considerare i ritardi di propagazione delle porte logiche utilizzate, ritardi che, combinati con la retroazione tra le uscite e gli ingressi, determinano il funzionamento particolare dei circuiti sequenziali, rispetto a quelli combinatori.

L'effetto del ritardo temporale viene riportato globalmente nel blocco in figura.



Nello studio che segue, si analizza una schematizzaione a blocchi di un circuito sequenziale, in cui si tiene conto del blocco combinatorio di partenza e della rete di reazione, rappresentata da un blocco di ritardo temporale.



Si considerano due variabili di ingresso x1 e x2, due variabili di uscita z1 e z2, due variabili interne v1 e v2,due variabili di uscita interne y1 e y2 (funzioni interne), che tengono conto dello stato interno del circuito sequenziale.

Normalmente, nell'analisi combinatoria, z é funzione solo di x; in questo contesto z é funzione di x e di t (tempo).

Per risolvere una tale situazione, occorrerebbe ricorrere a metodi analitici piuttosto complessi, ma si può affrontare il problema utilizzando metodi di graficizzazione delle funzioni, come di seguito esposto.

Si può, in effetti, considerare che y é funzione del tempo, tramite la variabile v, per cui y é funzione di v.

In definitiva si ha che z é funzione di x ed y, che, a sua volta é funzione di v.

Ricapitolando, si ha:

z=f(x) per le funzioni combinatorie;

z=f(x,t) per le funzioni sequenziali;

y=f(t) funzione interna del tempo;

y=f(v) funzione interna delle variabili interne;

z=f(x,y) funzione di uscita rispetto agli ingressi ed alle funzioni interne.

In base al circuito riportato sopra, si può ricavare un sistema di equazioni per le uscite z e la variabile interna v:



Si definisce uno stato stabile, quando v=y.

Quando v risulta diverso da y, si ha uno stato di instabilità.

Uno stato di instabilità tende sempre alla stabilità.

Da tali equazioni, si possono ricavare le seguenti tabelle di uscita, di eccitazione, degli stati, di flusso, nonchè un grafo di flusso, dal quale ricavare il funzionamento del LATCH S-R:





Nelle tabelle, i cerchietti indicano stati stabili, le frecce percorsi da stati instabili a stati stabili.

Dalla tabella di flusso (o dal grafo di flusso) si evince che:

- se S va basso ed R basso, Q mantiene il valore precedente;

- se S va alto ed R basso, Q andrà alto, se era basso, altrimenti rimane alto;

- se S va basso ed R alto, Q andrà basso, se era alto, altrimenti rimane basso;

- se S ed R vanno entrambi alti, l'uscita non é determinabile univocamente.

Tali risultati si possono riassumere in una tabella di verità:

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Q | Q |
| L | L | Q | Q |
| L | H | L | H |
| H | L | H | L |
| H | H | IND | IND |

Il circuito LATCH S-R si può ottenere ugualmente mediante porte NAND, negate agli ingressi (mediante porte NOT), come in figura:



Per eliminare lo stato di indeterminazione, relativo ad ingressi contemporaneamente alti, si considera un circuito analogo, in cui l'ingresso R viene collegato tramite una porta logica NOT all'ingresso S.

In tal modo, S ed R possono essere soltanto 'alto' l'uno e 'basso' l'altro, o viceversa, ma noni contemporaneamente 'alti' o 'bassi'.

Tale configurazione é denominata LATCH D.



La tabella di verità per il latch di tipo D é la seguente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D | S | R | Q | Q |
|  |  |  |  |  |
| L | L | H | L | H |
| H | H | L | H | L |
|  |  |  |  |  |

Poichè tali circuiti sequenziali sono inseriti in circuitazioni più complesse, in cui occorre spesso sincronizzare il funzionamento delle singole unità logiche sequenziali, si considerano latches di tipo S-R o di tipo D con ingressi di abilitazione.



Seguono le tabelle di verità di tali latches:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E | S | R | Q | Q |
| L | X | X | Q | Q |
| H | L | L | Q | Q |
| H | L | H | L | H |
| H | H | L | H | L |
| H | H | H | IND | IND |

Tabella di verità di latch S-R con ENABLE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E | D | S | R | Q | Q |
| L | X | X | X | Q | Q |
|  |  |  |  |  |  |
| H | L | L | H | L | H |
| H | H | H | L | H | L |
|  |  |  |  |  |  |

Tabella di verità di latch D con ENABLE

I latcthes finora analizzati commutano sul pianerottolo del segnale di comando (ENABLE).

Altri tipi di circuiti sequenziali sono comandati da appositi segnali di temporizzazione (CLOCK) e commutano sui fronti di salita dei segnali di comando (POSITIVE EDGE TRIGGERED), o sui fronti di discesa dei segnali di comando (NEGATIVE EDGE TRIGGERED), o alla fine di un impulso completo di clock (MASTER-SLAVE).

Tali circuiti sono denominati FLIP-FLOP.

Le reazioni interne, in tali circuiti, non permettono che le uscite commutino contemporaneamente agli ingressi, in presenza del segnale di clock, per cui le uscite sono ben separate dagli ingressi.



(Una forma d'onda si può dire impulsiva, quando al tendere a zero della sua durata, la sua ampiezza tende all'infinito.)

Di seguito si analizzano i FLIP-FLOP di tipo J-K e di tipo MASTER-SLAVE.

In tali circuiti, si pone in evidenza l'assenza di indeterminazione nel caso gli ingressi siano entrambi alti.

Infatti, in tale caso, l'uscita inverte.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J | K | CK | Q | Q |
| L | L | | | Q | Q |
| L | H | | | L | H |
| H | L | | | H | L |
| H | H | | | INV | INV |

Tabella di verità di FLIP-FLOP di tipo J-K

La struttura del FLIP-FLOP MASTER-SLAVE é composta da due parti, il MASTER (padrone), che inizia il suo funzionamento sul fronte di salita dell'impulso di comando, lo SLAVE (schiavo), che svolge la propria funzione sul fronte di discesa dell'impulso di comando.

La non contemporaneità delle azioni é garantita dalla porta logica NOT, inserita tra il CLOCK e un ingresso delle NAND di abilitazione dello SLAVE.

Da notare gli ingressi PR (PRESET=Predisposizione) e CR (CLEAR=Azzeramento).

Qualora il PRESET viene portato a livello alto, l'uscita va comunque alta, qualsiasi sia la configurazione degli ingressi e l'evoluzione del segnale di clock.

Analogamente, ma in modo opposto, funziona il CLEAR, che se posto ad uno, porta l'uscita a zero.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J | K | CK | Q | Q |
| L | L |  | Q | Q |
| L | H |  | L | H |
| H | L |  | H | L |
| H | H |  | INV | INV |

Tabella di verità di FLIP-FLOP di tipo MASTER-SLAVE

Anche per i FLIP-FLOP esiste la configurazione di tipo D. ottenuta collegando una porta logica NOT tra gli ingressi J e K.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D | J | K | CK | Q | Q |
|  |  |  |  |  |  |
| L | L | H |  | L | H |
| H | H | L |  | H | L |
|  |  |  |  |  |  |

Tabella di verità di FLIP-FLOP di tipo D

Connettendo gli ingressi J e K a livello alto, si ottiene la configurazione, cosiddetta, di tipo T.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| J | K | CK | Q | Q |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| H | H |  | INV | INV |

Tabella di verità di FLIP-FLOP di tipo T

Tale configurazione costituisce il blocco fondamentale dei circuiti contatori e divisori di frequenza.

Infatti, ogni due impulsi di clock, si ottiene un impulso del segnale di uscita, per cui tale circuito di tipo T costituisce un divisore per due della frequenza del segnale di clock.

Inoltre, il FLIP-FLOP di tipo T, può considerarsi un circuito contatore, perchè conteggia gli impulsi del segnale di clock, utilizzando il terminale di clock come ingresso.

Collegando più unità di tali circuiti in cascata, si ottengono divisioni o conteggi dipendenti dal numero di unità collegate.

Si ha, in particolare:



n indica il cosiddetto modulo dell'unità di conteggio.

Seguendo l'analisi temporale dei segnali di clock e di uscita, ci si può rendere conto delle affermazioni effettuate.



Il funzionamento di tali circuiti contatori é di tipo asincrono, in quanto la commutazione dei singoli FLIP-FLOP avviene in dipendenza delle unità precedenti.

Infatti, in questo caso, le uscite delle unità precedenti pilotano gli ingressi di clock successivi, per cui la commutazione non è contemporanea su tutti i FLIP-FLOP costituenti il circuito di conteggio.

Tale tipo di collegamento porta, inoltre ad un problema di ritardo di commutazione per le uscite, dovuto ai ritardi delle singole porte logiche costituenti i FLIP-FLOP.

Tali ritardi si sommano, non permettendo un azzeramento simultaneo delle uscite in corrispondenza del modulo previsto per l'unità di conteggio.

Ciò avviene quando la frequenza di clock diviene elevata, per cui il periodo di clock é basso e comparabile con il tempo di ritardo di commutazione (tempo di propagazione) che generalmente é attorno ai 20 ns (nanosecondi).

Si può ovviare a tale inconveniente, portando il segnale di clock contemporaneamente a tutti gli ingressi di clock delle singole unità di conteggio, in modalità sincrona.

I contatori asincroni possono utilmente essere impiegati come divisori di frequenza o come contatori per basse frequenze, mentre quelli sincroni sono impiegati nei conteggi veloci.

Di seguito si analizza un contatore integrato a modulo 2x8, di tipo asincrono,con sigla commerciale 7493.

E' costituito da quattro FLIP-FLOP, di cui il primo a se stante e gli altri tre connessi tra loro internamente, in modo da poter ottenere tre moduli, collegandoli opportunamente.

Utilizzando solo il primo, si ha un modulo 2; utilizzando gli altri tre, si ottiene un modulo 8; utilizzando il primo collegato agli altri tre, si ottiene un modulo 16 (il massimo conteggio possibile).



Dall'analisi del diagramma temporale riportato di seguito si possono visualizzare i comportamenti delle uscite Q e stabilire come vengono effettuati i conteggi su modulo 10 o 16.

