

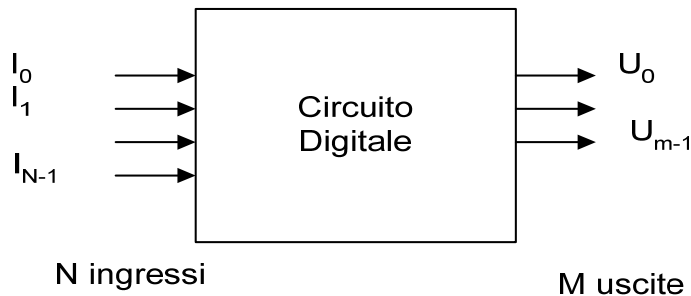
I circuiti dei calcolatori, le memorie, i bus. I fondamenti della rappresentazione dell'informazione e della sua trasmissione ed elaborazione.

Dispensina per gli studenti di Ingegneria Gestionale. A.A. 2007-2008
Di P. Foglia

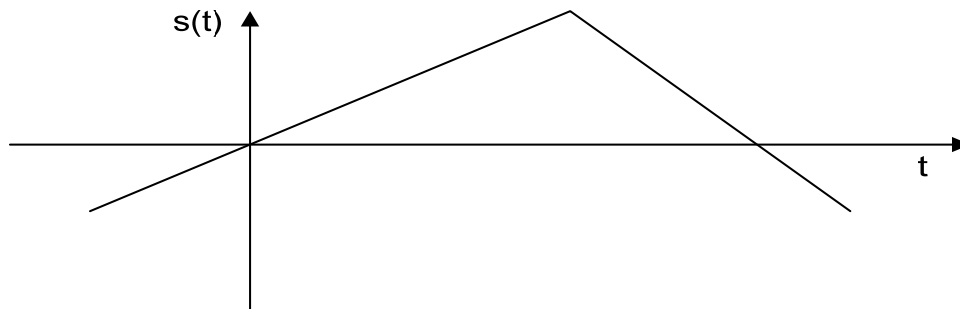
1.0 le nozioni di base

I circuiti digitali

I circuiti che costituiscono il calcolatore sono circuiti digitali, ossia circuiti che ricevono in ingresso segnali digitali e forniscono in uscita segnali digitali.



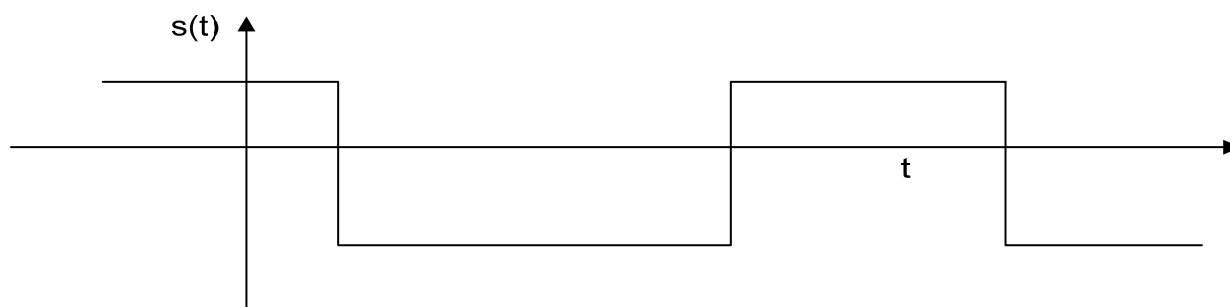
Circuito Digitale: il circuito riceve dei segnali in ingresso e produce delle uscite. Sia gli ingressi che le uscite sono segnali digitali, ossia sono insiemi di segnali binari.



Segnale

Un segnale è una grandezza fisica variabile nel tempo, che trasporta informazione associata alle sue variazioni.

Un segnale digitale è un insieme di segnali binari, ossia un insieme di segnali che possono assumere solo due valori.



Segnale digitale ad una cifra

I segnali in ingresso ed in uscita al calcolatore sono tensioni; in particolare sono tensioni binarie: possono assumere solo due valori, a cui, logicamente, si attribuisce il significato logico di 1 o 0.

Ad esempio, esistono calcolatori per cui la tensione di 0V rappresenta il valore 0, la tensione di 2.5 V rappresenta il valore 1.

Proprietà delle variabili binarie, codifica e rappresentazione, elaborazione non numerica, trasmissione.

Avendo N variabili binarie, ossia variabili che possono assumere solo i valori di 0 ed 1, e considerando come unica la enupla dei valori delle variabili, l'insieme dei valori assumibili da tali variabili vale 2^n (dal calcolo combinatorio).

Esempio: avendo due variabili binarie I_0 ed I_1 , i possibili valori assumibili dalle variabili sono

| I_0 | I_1 |
|-------|-------|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

E dunque la coppia di variabili può assumere 4 distinti valori.

In conseguenza di ciò, dovendo rappresentare un alfabeto di simboli tramite variabili binarie, (un alfabeto è un insieme di simboli, proprio come le lettere dell'alfabeto), se n è il numero di simboli dell'alfabeto, il numero di variabili binarie necessarie per rappresentare tali simboli è

$$N_{bit} = \lceil \log_2 n \rceil \quad \text{dove } \lceil \rceil \text{ è l'operatore intero superiore (1)}$$

Ciascuna variabile binaria prende il nome di bit (da binary digit, ossia cifra binaria).

La corrispondenza fra i simboli dell'alfabeto ed i valori delle variabili binarie che li rappresentano si esprime tramite una tabella (tabella di corrispondenza). Tale corrispondenza prende il nome di codifica, e permette di associare ad ogni simbolo dell'alfabeto un insieme di valori delle variabili binarie che li rappresentano.

Esempio:

si hanno i simboli A, B, C. Si vuole individuare una codifica per tali simboli:

Poiché i simboli sono 3, occorrono 2 bit, ossia 2 variabili binarie. Infatti, nella formula (I), il logaritmo in base due viene 1. qualcosa, e dunque l'intero superiore viene 2. D'altra parte, se si assume che il numero di cifre sia pari ad uno, il numero di simboli rappresentabili verrebbe 2 (2 elevato alla uno) e dunque sarebbe insufficiente per rappresentare 3 simboli, mentre se assumo che il numero di cifre sia pari a due, posso rappresentare 4 (2 elevato al quadrato) simboli.

A questo punto, occorre generare la tabella di corrispondenza. Si scrivono le 4 possibili configurazioni che le due variabili binarie possono assumere, e tre di esse si associano ai tre simboli. Per la quarta configurazione, l'associazione è ND (non definita).

| b1 | b0 | Simbolo |
|----|----|---------|
| 0 | 0 | A |
| 0 | 1 | B |
| 1 | 0 | C |
| 1 | 1 | N.D. |

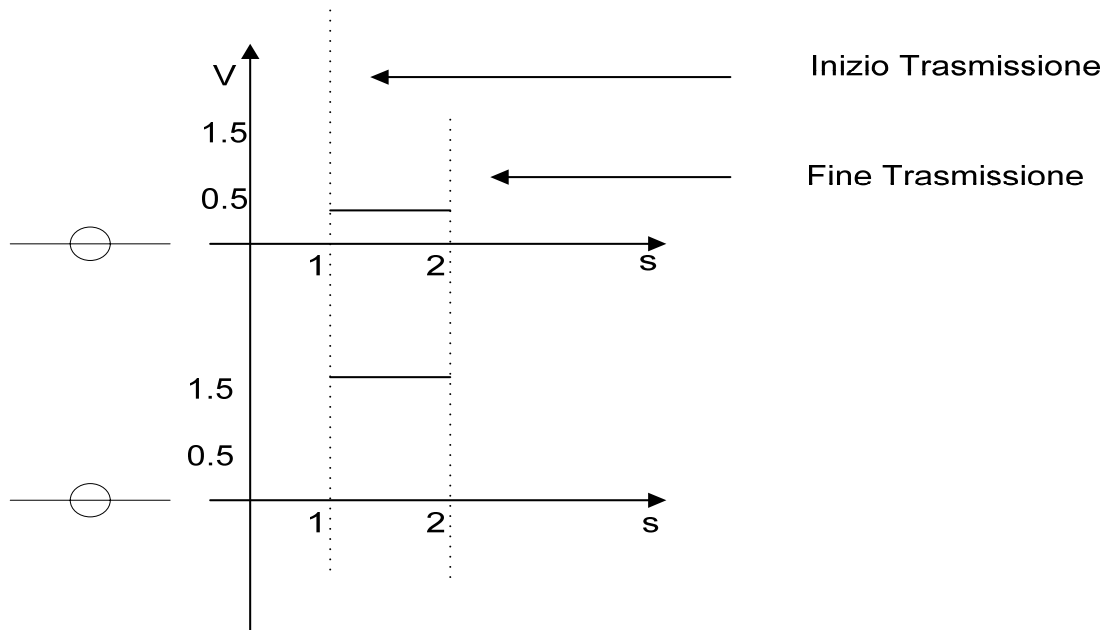
Tabelle di corrispondenza

In tal modo, dovendo effettuare la trasmissione o la elaborazione dei simboli o di parole composte da quei simboli, si può effettuare la trasmissione o la elaborazione delle variabili binarie che li rappresentano.

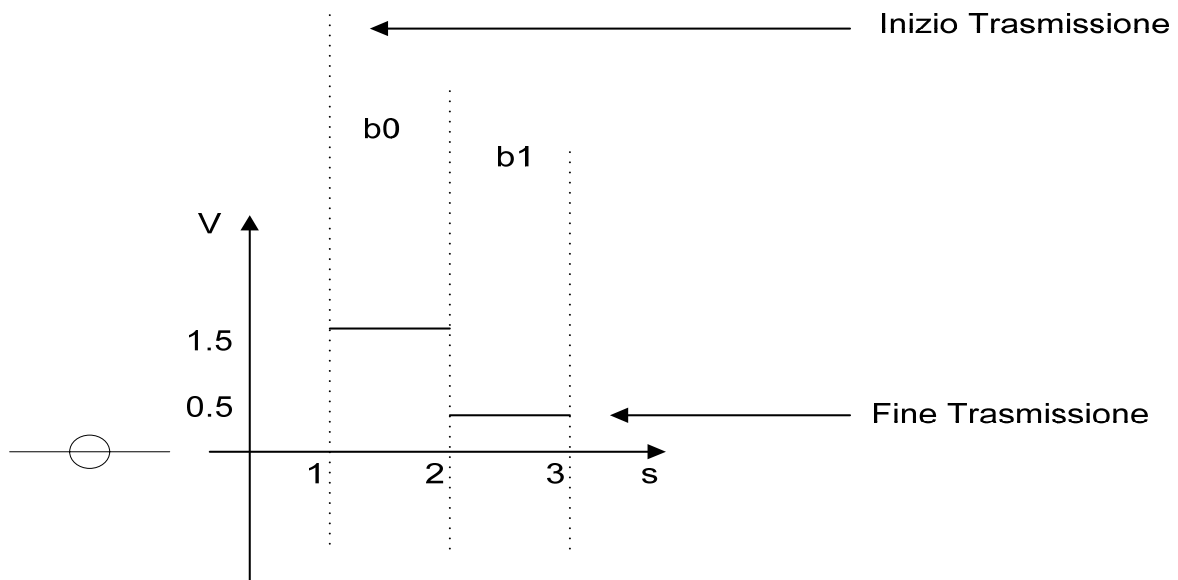
Ad esempio, dovendo trasmettere in modo "digitale" il carattere B della codifica precedente, si possono utilizzare due fili, uno rappresentante il bit b1, l'altro il bit b0, e trasmettere sul filo rappresentante b1 la tensione corrispondente al valore di 0, sull'altro la tensione corrispondente al valore 1 (trasmissione parallela). La comunicazione fra computer e stampante, con il cavo parallelo, avviene secondo tale modalità.

In alternativa, si può utilizzare un unico filo, e dire che il primo valore di tensione che trasmetto è il valore di b0, il secondo di b1. Imponendo che la trasmissione di un valore duri un intervallo costante, sono in grado di trasmettere e ricevere i vari simboli (trasmissione seriale). Un modem è un dispositivo che trasmette con tale modalità.

Esempio: Trasmissione seriale e parallela del carattere B. La tensione rappresentante lo 0 vale 0.5 V, la tensione rappresentante 1 vale 1.5 V, l'intervallo di trasmissione vale 1s (1 secondo)



Trasmissione Parallela



Trasmissione Seriale

Poiché il calcolatore è in grado di elaborare informazione binaria, il calcolatore diventa in grado di elaborare anche l'informazione associata all'alfabeto di simboli ed alle parole generate con tale alfabeto (e dunque di effettuare elaborazione non numerica).

I circuiti di base dei calcolatori

I circuiti di base per la realizzazione dei calcolatori, sono i fili, le reti combinatorie, le memorie elementari ed in più generale le reti sequenziali.

2.0 i fili

i fili rappresentano i collegamenti fra i vari elementi del computer. Possono essere materialmente dei fili o delle connessioni su di un circuito stampato.

Sui fili viaggiano delle tensioni binarie, ossia delle tensioni che possono assumere solo due valori, a cui faremo corrispondere, per convenzione, i valori logici di 0 e di 1.

I fili vengono rappresentati o singolarmente o come dei fasci.

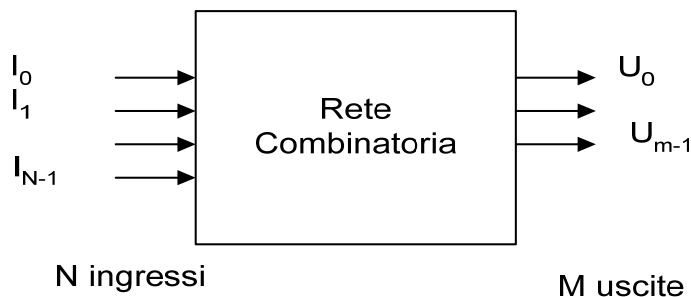
Ai fili è inoltre associata una direzione, con riferimento al cammino di un segnale, ed un filo può essere monodirezionale o bidirezionale



3.0 le reti combinatorie

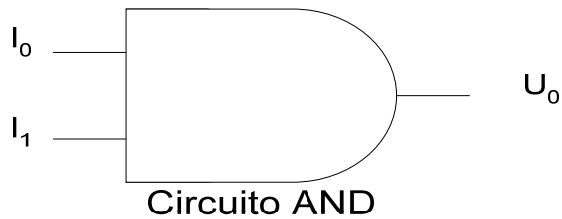
Una rete combinatoria è un circuito elettrico dotato di N ingressi ed M uscite (chiamate piedini), e caratterizzato da una legge di funzionamento che associa ad ogni valore degli ingressi un valore per le uscite (ossia $U=f(I)$). Tale legge di funzionamento si può descrivere tramite tavole di verità, dal momento che l'insieme dei valori assumibili per gli ingressi è finito (pari a 2^n).

La tabella di verità associa ad ognuno dei possibili valori degli ingressi i valori delle uscite.



Esempio di reti combinatorie:

Il circuito logico (porta) AND



Possibili valori degli ingressi

La tabella di verità del circuito AND

| I_0 | I_1 | U_0 |
|-------|-------|-------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Valori delle uscite

La porta and può avere in generale n variabili di ingresso. In tale porta l'uscita vale 1 se e solo se tutti i valori delle variabili di ingresso valgono uno. Tale porta può essere considerata un riconoscitore di uni in ingresso.

Il circuito logico (porta) NOT

E' una rete ad un ingresso e ad una uscita. Prende anche il nome di porta NOT. L'uscita è ottenuta applicando la funzione logica di "negazione" all'ingresso.

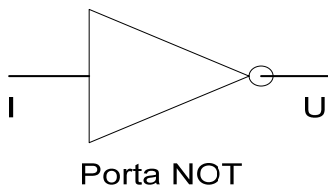


Tabella di verità della porta not

| I | U |
|---|---|
| 0 | 1 |
| 1 | 0 |

Il circuito logico (porta) OR

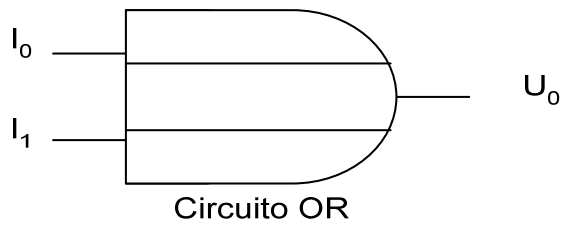


Tabella di verità del circuito OR

| I_0 | I_1 | U_0 |
|-------|-------|-------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

La porta and può avere in generale n variabili di ingresso. In tale porta l'uscita vale 1 se almeno uno dei valori delle variabili di ingresso vale uno.

La maschera o riconoscitore di ingressi

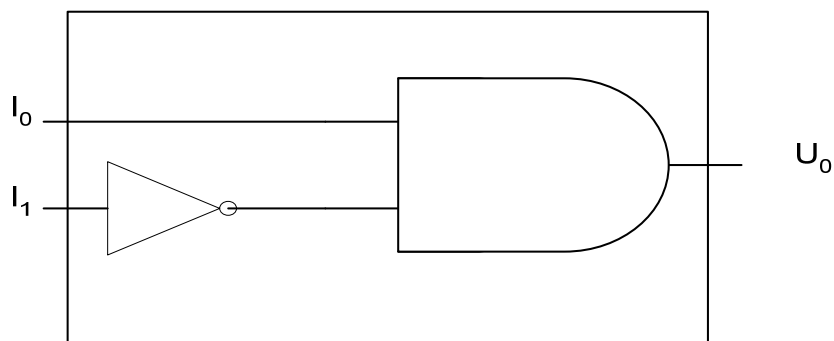
E' una rete combinatoria ad N ingressi in cui l'uscita vale uno per una sola configurazione dei valori di ingresso. Si ottiene tramite una porta AND ad N ingressi e delle porte NOT, negando, prima della AND, gli ingressi che devono essere riconosciuti a 0.

Esempio:

Tabella di verità di una maschera che riconosce 10

| I_0 | I_1 | U_0 |
|-------|-------|-------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Implementazione della maschera precedente:

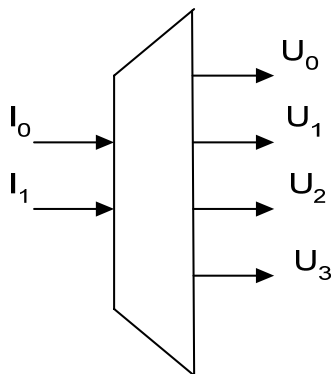


Maschera 10

In generale, combinando porte AND, OR e NOT, è possibile realizzare una qualunque rete combinatoria

Il decodificatore

Un altro esempio di circuito combinatorio è il decodificatore, dotato di N ingressi e 2^n uscite, tale che per ognuna delle configurazioni di ingresso viene abilitata una ed una sola configurazione di uscita (ossia un solo piedino di uscita ha il valore uno, tutti gli altri sono a zero).



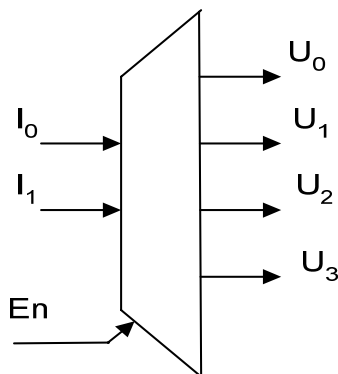
Il decodificatore

Tabella di verità del decodificatore

| I_0 | I_1 | U_0 | U_1 | U_2 | U_3 |
|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

Il decodificatore con abilitazione

E' simile alla rete precedente, ma con un ingresso in più (enable o EN), tale che, per $EN=1$ la rete si comporta come il decodificatore del passo precedente. Per $EN=0$ le uscite valgono sempre 0. Da qui il significato di EN: abilita o meno il funzionamento del decodificatore.



Il decodificatore con abilitazione

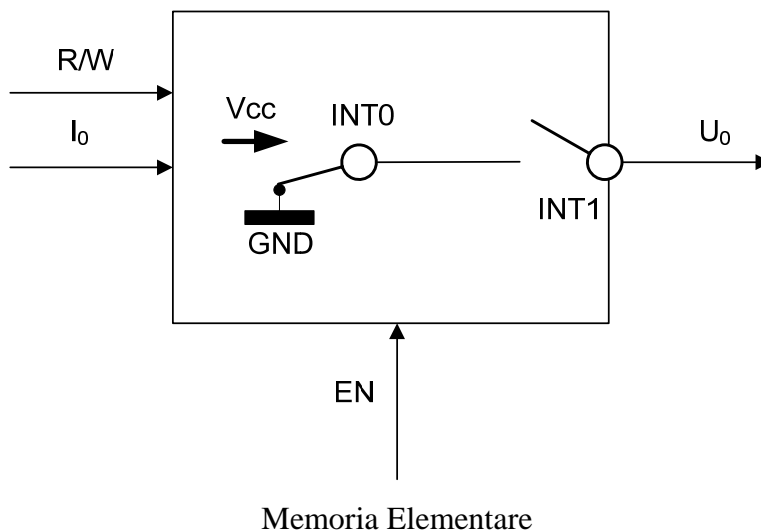
Tabella di verità del decodificatore con abilitazione

| EN | I ₁ | U ₀ | U ₁ | U ₂ | U ₃ |
|----|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

4.0 le memorie elementari e le reti sequenziali asincrone

La memoria elementare è un circuito con un piedino di ingresso, un piedino di uscita, due piedini di controllo (Read/Write e Enable) ed uno stato interno rappresentato da un bit di informazione (il contenuto della memoria).

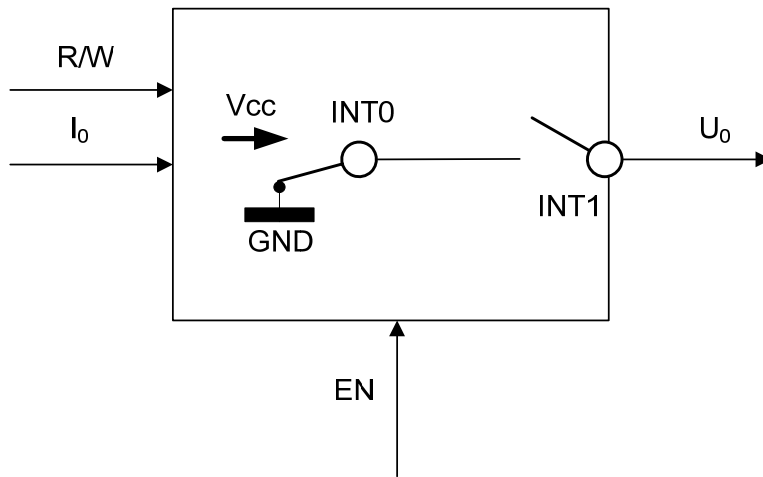
Si può pensare che esistano due interruttori all'interno del circuito (INT0 ed INT1). Il primo (INT0) può essere posto o alla tensione di 0 (o GND o massa), rappresentante lo stato logico 0, o alla tensione di VCC, rappresentante lo stato logico 1. Il secondo consente di collegare la tensione di VCC o GND (il base alla posizione di INT0) all'uscita. Il piedino R/W indica se occorre fare una operazione di lettura o scrittura, il piedino EN serve per abilitare il circuito.



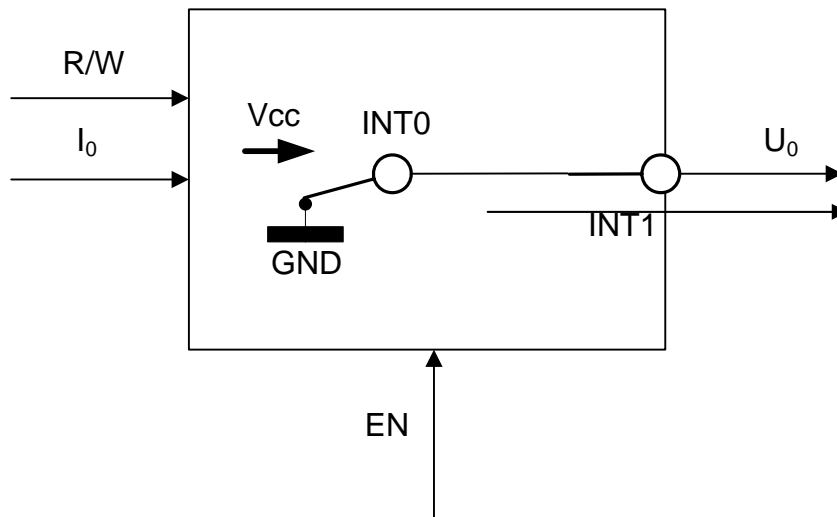
Mentre per un circuito combinatorio il valore dell'uscita è funzione dei valori degli ingressi, in una memoria elementare il comportamento del circuito (legge che permette di determinare il valore dello stato interno e delle uscite) è determinato da una legge più complessa, in quanto dipende dal valore dell'ingresso, dal valore dei piedini di controllo, ma anche dallo stato interno.

Il funzionamento avviene secondo la seguente legge:

- 1) quando il piedino di Enable è a 0, lo stato interno si conserva. Il circuito è "insensibile" al valore del piedino di ingresso e del piedino di Read/Write. L'uscita è "non connessa" (circuito aperto).
- 2) Quando il piedino di Enable vale 1:
 - a. Se Read/Write vale 1 (operazione di lettura), l'uscita del circuito assume il valore della tensione a cui è connesso INT0. Tale tensione rappresenta lo stato interno del circuito, per cui, logicamente, si effettua la lettura dello stato interno..
 - b. Se Read/Write vale 0 (operazione di scrittura), l'interruttore INT0 viene messo a GND o VCC sulla base del valore del piedino I0. Se I0 vale 0, l'interruttore va a GND (ossia 0 logico), se I0 vale 1 INT0 si sposta a VCC (ossia 1 logico). Si può dire che lo stato interno (posizione di INT0) viene aggiornato allo stato del Pin di ingresso. Il piedino di uscita è non connesso..



Funzionamento del circuito quando $EN=0$. Lo stato di R/W e del Piedino di Ingresso è ininfluenza. L'uscita è "aperta"..



Funzionamento del circuito quando $EN=1$ e $R/W=1$ (operazione di lettura). $INT1$ si chiude, di modo che il valore dello stato interno (posizione di $INT0$) finisce in uscita, ossia il contenuto della memoria viene letto.

La differenza fra il circuito combinatorio ed il circuito "memoria elementare", che tecnicamente fa parte della famiglia dei circuiti sequenziali asincroni risiede nello stato interno. In un circuito combinatorio, noti i valori degli ingressi, si è sempre in grado di conoscere l'uscita (determinata dalla tabella di verità).

In un circuito sequenziale, esiste il concetto di stato interno ed è la sequenza con cui sono stati forniti gli ingressi "nel passato" che determina l'uscita (considerando come ingresso anche i piedini di controllo).

Più propriamente l'uscita è funzione dello stato interno. Lo stato interno è funzione degli ingressi e, in generale, dello stato interno precedente (per le reti sequenziali asincrone).

Esempio:

se lo stato interno vale uno (ossia INT0 è a VCC), l'ingresso I0=0, R/W=0, EN=1 genera l'uscita 1 (U0 = 1).

se lo stato interno vale zero (ossia INT0 è a VCC), l'ingresso I0=0, R/W=0, EN=1 genera l'uscita 0 (ossia lo stesso ingresso di prima) genera l'uscita 0 (U0 = 0).

Da notare che il valore dell'ingresso I0 è ininfluenza ai fini del risultato.

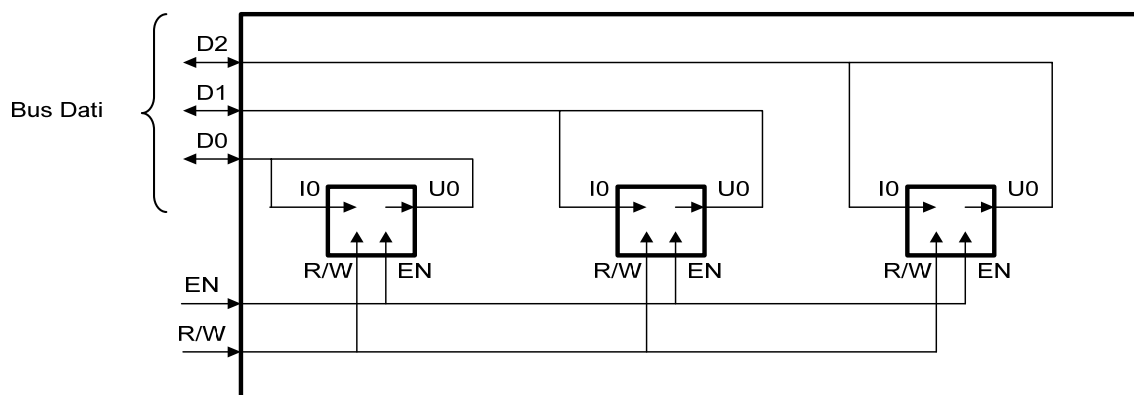
5.0 la memoria principale del computer

Utilizzando tali circuiti è possibile capire come è realizzata la memoria all'interno di un computer.

Sulla memoria si possono effettuare due operazioni, lettura e scrittura. La memoria è organizzata in parole o word, ossia in sequenze di bit. Ciò vuol dire che la lettura o la scrittura accedono ad una parola e non al singolo bit.

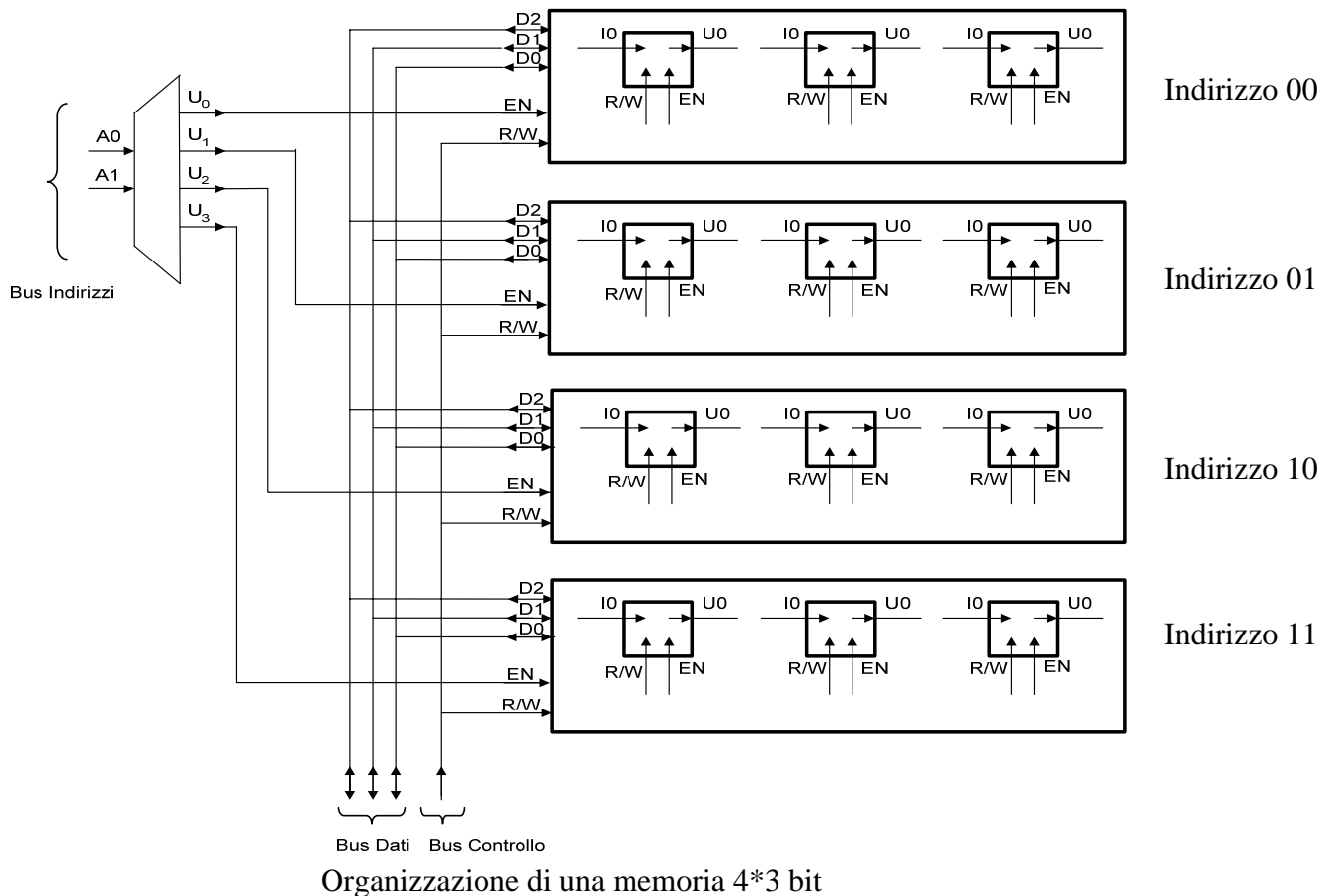
Dimensioni tipiche per la parola sono il byte (8 bit), oppure 16 o 32 bit.

Una singola parola di n bit si ottiene tramite n memorie elementari viste nel punto precedente. I segnali di controllo R/W ed EN delle memorie elementari sono cortocircuitati (vedi figura). Inoltre, i segnali di ingresso e di uscita della singola memoria elementare possono essere cortocircuitati, poiché, analizzando il funzionamento, solo o l'uno o l'altro sono funzionanti (connessi con lo stato interno). Così, la parola di memoria può essere vista come un circuito che dispone di 2 piedini di controllo (R/W ed EN) e di n (dove n è la dimensione della parola) piedini di ingresso/uscita, ossia bidirezionali, che rappresentano i dati (pin D0-D3 in figura). La figura mostra il montaggio di una riga di memoria a 3 bit.



Organizzazione di una riga di memoria con parola di 3 bit

Tale struttura viene replicata M volte, dove M è la dimensione della memoria. Si dice che la memoria contiene M locazioni da N bit. I piedini dei dati, come pure il segnale di R/W, vengono cortocircuitati. Il segnale di EN(able) invece, per ciascun modulo, viene collegato ad un decodificatore. In tal modo, se $M=2^k$, con k fili si può selezionare una singola parola di memoria. Tali K fili costituiscono "l'indirizzo" della locazione di memoria.



La tabella di verità del decodificatore, affinché le parole della memoria abbiano gli indirizzi in figura, è la seguente:

| A0 | A1 | U0 | U1 | U2 | U3 |
|----|----|----|----|----|----|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

6.0 la composizione del bus ed i cicli di lettura e scrittura

Considerando il bus, nel modello di Van Newman, questo è allora costituito da tre gruppi di fili:

- 1) il bus dati, bidirezionale, che serve per la trasmissione dei dati da leggere e da scrivere fra CPU e dispositivo.
- 2) Il bus di controllo, monodirezionale, con cui la CPU trasmette almeno i comandi di lettura o scrittura ai dispositivi.
- 3) Il bus degli indirizzi, monodirezionale, con cui la CPU seleziona i dispositivi.

Ciò vale tanto per la connessione con le memorie, quanto per la connessione con l'insieme dei dispositivi di I/O.

La CPU, per dialogare con i vari dispositivi, dovrà preparare i segnali sul bus con un ordine determinato che varia in base all'operazione che intende effettuare:

Nel caso di lettura dovrà:

- 1) preparare gli indirizzi
- 2) mettere R/W a 1
- 3) attendere (perché i circuiti devono "elaborare" i segnali) che sul bus dei dati arrivino i dati

Nel caso di scrittura dovrà:

- 4) preparare gli indirizzi
 - 5) preparare i dati
 - 6) mettere R/W a 0
- dopo un certo intervallo, i dati saranno scritti in memoria.

Gli intervalli sono specificati nei manuali dei costruttori.

7.0 Esercizi:

1) Si vogliono codificare tutte le parole contenenti i caratteri A, B, D, E. Individuare una codifica binaria per tali caratteri e rappresentarla in tabella. Di quanti bit c'è bisogno per codificare ciascun carattere?

C'è bisogno di 2 bit, poiché ho 4 valori.

La codifica è una qualsiasi tabella che faccia corrispondere ai 4 valori dei due bit le due lettere.

2) Per la codifica del punto precedente, mostrare, con lo schema parallelo e con lo schema seriale, la trasmissione delle parole ABBA e poi la trasmissione delle parole DDBE

3) Determinare la dimensione del bus dati, del bus degli indirizzi, la tabella di verità del decodificatore ed effettuare il progetto per una memoria 4x4 bit.

4) Progettare una memoria (ossia determinare la dimensione del bus dati, del bus degli indirizzi, la tabella di verità del decodificatore e mostrare lo schema di progetto) destinata a contenere le codifiche dei caratteri A, F, G (si suppone che l'alfabeto, ossia l'insieme dei simboli, sia costituito solo da questi tre caratteri). La memoria deve contenere fisicamente i tre caratteri A, F, G.

5) Progettare una memoria destinata a contenere 4 caratteri tratti dall'alfabeto U, V, Z. Mostrare il contenuto della memoria quando contiene i caratteri UUVV.

6) Progettare una memoria (ossia determinare la dimensione del bus dati, del bus degli indirizzi, la tabella di verità del decodificatore e mostrare lo schema di progetto) destinata a contenere 8 parole costituite dai caratteri A, F, G (si suppone che l'alfabeto, ossia l'insieme dei simboli, sia costituito solo da questi tre caratteri). Le parole sono lunghe 4 caratteri.

7) Ad un bus con 3 bit di indirizzi e due di dati è connessa una RAM 8×2 bit, il cui contenuto è inizialmente nullo (tutti 0). Mostrare come si modifica il contenuto della RAM quando il processore esegue le seguenti operazioni:

- a) il processore legge il contenuto della locazione 000, vi somma 1 e lo trasferisce in 010
- b) il processore scrive 11 in 100
- c) il processore scrive 01 in 111