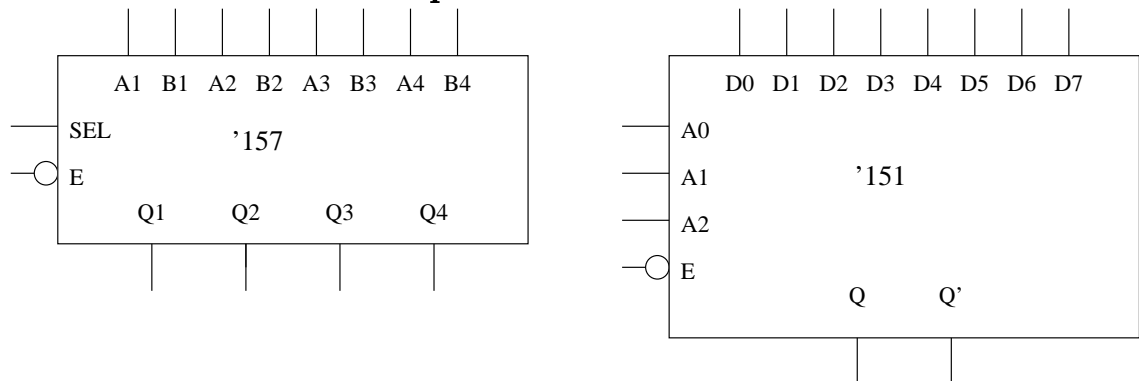


Multiplexer

- Il **multiplexer** è un circuito che seleziona il dato presente su uno fra N ingressi e lo trasmette all'uscita; è l'equivalente elettronico di un interruttore meccanico a N posizioni. Come primo esempio vediamo (a sinistra) il '157 che contiene 4 sezioni identiche del tipo *da 2 a 1*:



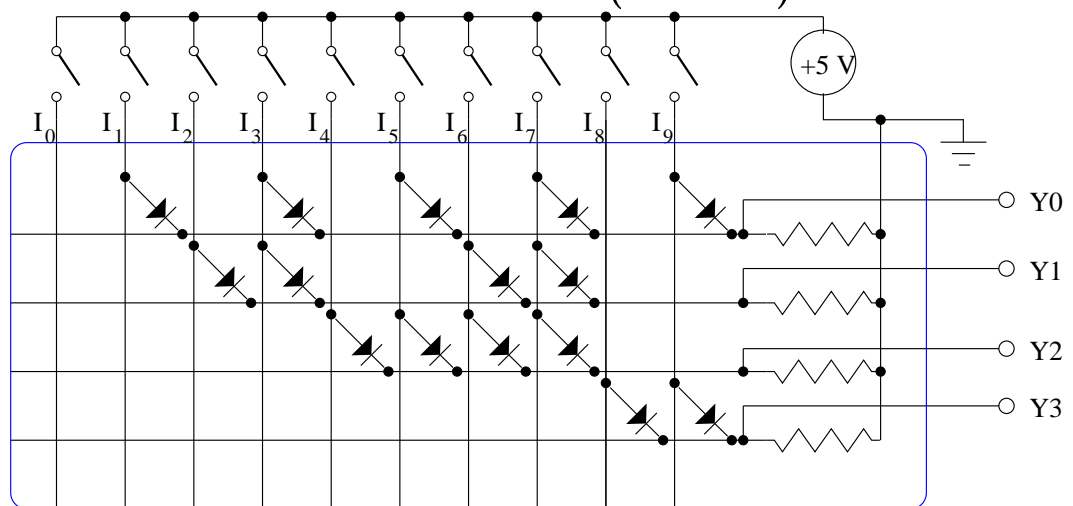
quando l'ingresso di controllo SEL (*SElect*) è LOW, gli ingressi A_i vengono passati alle rispettive uscite Q_i ; quando SEL è HIGH vengono passati gli ingressi B_i ; infine, se l'ingresso di abilitazione \bar{E} è HIGH, tutte le uscite diventano LOW, secondo la tabella di verità:

Ingressi				Uscite	(X = qualsiasi)
\bar{E}	SEL	A_i	B_i	Q_i	
H	X	X	X	L	
L	L	L	X	L	
L	L	H	X	H	
L	H	X	L	L	
L	H	X	H	H	

- Questo tipo di circuito può essere usato per commutare simultaneamente 4 linee mediante un singolo interruttore (elettronico o meccanico) collegato all'ingresso *SEL*, con il vantaggio di ridurre i cablaggi e di migliorare l'immunità al rumore.
- Un altro multiplexer di uso frequente è il '151, del tipo *da 8 a 1*, con uscita diretta Q e negata \bar{Q} ; l'ingresso selezionato dai 3 bit di indirizzo ($A_0 A_1 A_2$) viene passato all'uscita Q ; se l'ingresso di abilitazione \bar{E} è HIGH, l'uscita Q diventa LOW.
- È sempre possibile collegare in parallelo due o più multiplexer per aumentare il numero di ingressi; ad es. con quattro bit di indirizzo, due '151 e un paio di porte logiche aggiuntive si può costruire un multiplexer del tipo *da 16 a 1*.

Encoder

- Un **encoder** è un circuito che riconosce quale tra N ingressi è nello stato 1 (tutti gli altri ingressi devono essere nello stato 0) e produce in uscita un codice a M bit (dev'essere $2^M \geq N$). Una variante è il **priority encoder**, che produce il codice corrispondente all'ingresso di ordine più alto che si trova nello stato 1.
- Alcune applicazioni dell'encoder sono:
 1. generazione di codici (ad es. BCD o ASCII) mediante una tastiera
 2. stadio finale della conversione Analogico-Digitale parallela (*flash-ADC*)
 3. assegnazione di una priorità alle diversi sorgenti di interruzioni in un calcolatore
- Un esempio concreto è un encoder che produce il codice BCD associato ad una cifra decimale (da 0 a 9):



i diodi possono essere rimpiazzati da transistor multi-emettitore, e basta uno di questi transistor per linea di ingresso.

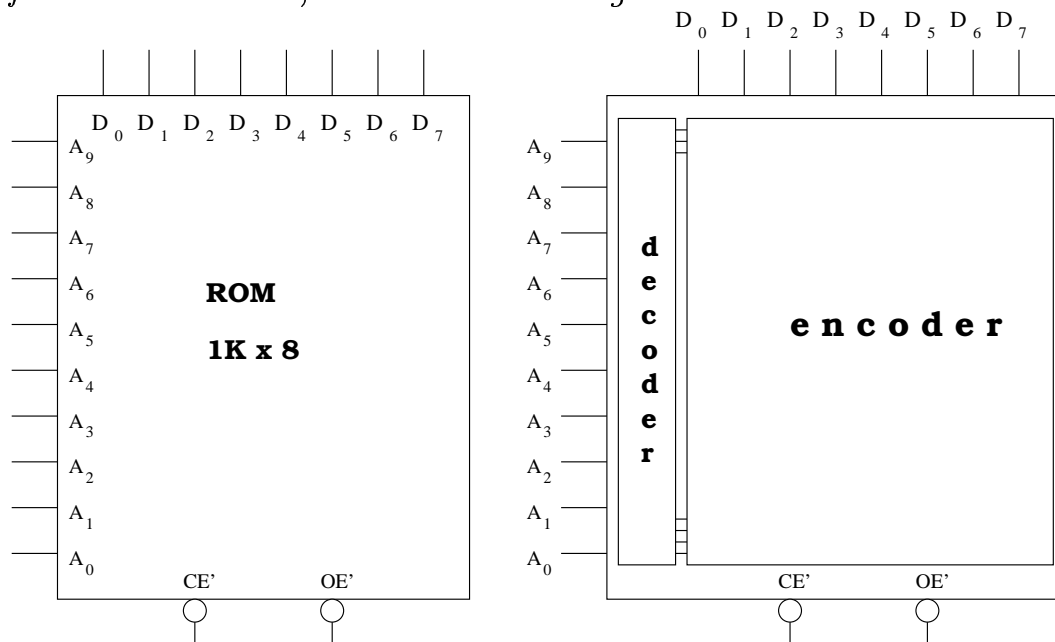
- I circuiti visti fin qui possono essere usati per implementare tabelle di verità arbitrarie: ad esempio un multiplexer a M bit di indirizzo e $N = 2^M$ ingressi può generare direttamente una tabella di verità con N righe, M colonne di ingressi e una colonna di uscita. Con l'aiuto di un invertitore si può realizzare una tabella a $2N$ righe, come richiesto dall'esercizio seguente.

Esercizio

Costruire un circuito la cui uscita Q è 1 se un dato mese in ingresso (da 1 a 12) ha 31 giorni.

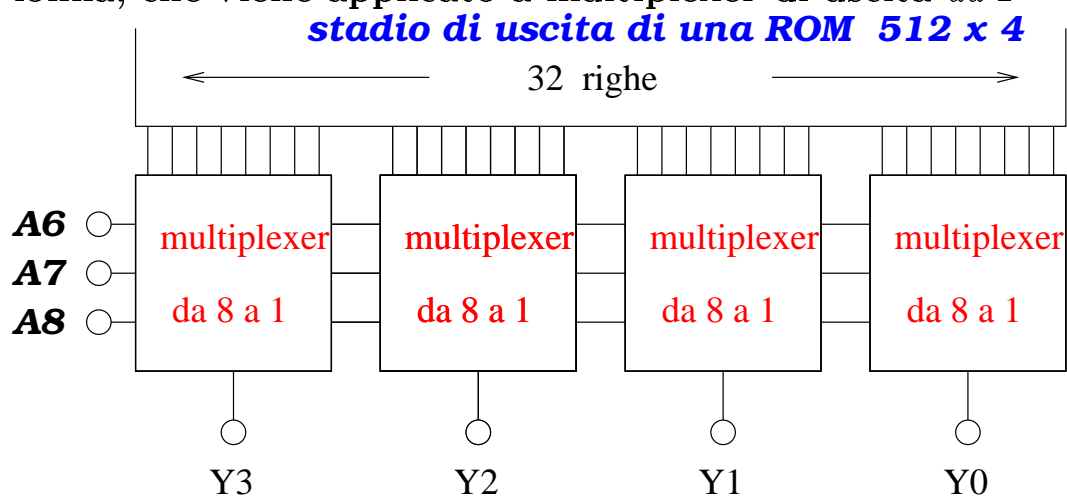
Memorie a sola lettura (ROM)

- Quando si tratta di costruire un circuito combinatorio corrispondente a una tabella di verità complicata con molti ingressi, i metodi visti finora diventano sempre meno pratici all'aumentare del numero di porte elementari necessarie. Fortunatamente esistono metodi più generali, basati su circuiti integrati che permettono di *programmare* i loro collegamenti interni: memorie a sola lettura (*ROM*, ovvero Read Only Memory) e altri circuiti denominati *PAL* e *PLA*. Consideriamo per ora circuiti puramente combinatori, cioè programmati una volta per sempre.
- La **memoria a sola lettura** accetta come ingresso un indirizzo a M bit e per ogni stato degli ingressi fornisce un predeterminato valore degli N bit di uscita. In figura è rappresentata una ROM del tipo $1K \times 8$, dove $1K = 1024 = 2^{10}$ indica la *profondità* di 10 bit, e 8 indica la *larghezza di 8 bit* in uscita.



- La **ROM** può essere vista come la combinazione di un *decoder* dell'indirizzo in ingresso, in cui una soltanto delle 2^M linee di uscita viene attivata volta per volta, seguita da un *encoder* con 2^M ingressi e N uscite; la tabella di verità è immagazzinata nell'encoder (ad es. per mezzo di diodi). La ROM in figura potrebbe essere usata per realizzare un moltiplicatore $4 \text{ bit} \times 4 \text{ bit}$ (utilizzando 8 dei 10 bit d'ingresso).

- Ci sono molte varietà di ROM con diversi metodi di programmazione:
 1. ROM programmate dal costruttore (a livello di maschera di litografia)
 2. PROM programmabili una sola volta dall'utente (mediante opportuni segnali elettrici)
 3. EPROM programmabili più volte dall'utente, con l'informazione immagazzinata sotto forma di carica, isolata mediante SiO_2 e cancellabile mediante esposizione a raggi UV
 4. EEPROM simili alle EPROM ma cancellabili mediante applicazione di una tensione di $\simeq 10$ V
- le ROM e PROM a transistor bipolari hanno tempi di accesso di 25-50 ns, mentre le memorie in tecnologia NMOS o CMOS sono un pò meno veloci; le EPROM e le EEPROM esistono solo in tecnologia MOS.
- Quando il numero di ingressi M aumenta, il decoder di ingresso richiede un numero elevato di porte; per ridurre il numero di componenti si adotta l'*indirizzamento bidimensionale*: le $2^M \times N$ celle di memoria vengono organizzate in 2^k righe per $2^{M-k} \times N$ colonne; una parte dell'indirizzo ($A_0 A_1 \dots A_{k-1}$) viene usata per generare un indirizzo di riga, mentre la parte restante dell'indirizzo ($A_k \dots A_{M-1}$) genera un indirizzo di colonna, che viene applicato a multiplexer di uscita da 2^{M-k} a 1.



In questo modo viene ridotta la complessità del decoder di ingresso aggiungendo i multiplexer di uscita, con un notevole risparmio in numero di componenti.

- Alcune applicazioni delle ROM sono:
 1. tavole di funzioni trigonometriche, logaritmi, etc. (più veloci di un algoritmo software)
 2. generazione di forme d'onda
 3. conversione tra due codici diversi
 4. programmi di controllo (ad es. in un calcolatore tascabile)
 5. logica combinatoria in genere con M ingressi ed N uscite, con M ed N grandi