

Appunti sull'interconnessione tra registri

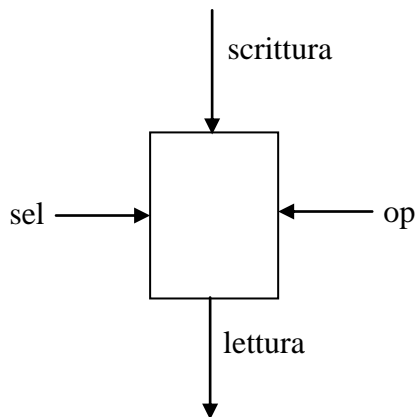
a cura di A. Massini

Cella di memoria

Una cella elementare di memoria è una rete sequenziale in grado di conservare nel tempo un solo bit di informazione.

Le tre proprietà fondamentali che caratterizzano una cella sono:

1. **conservare** nel tempo l'informazione (un singolo bit);
2. consentire di **leggere**, cioè duplicare, il valore memorizzato su una opportuna linea di uscita;
3. consentire di **scrivere** nella cella un valore arbitrario presente sulla linea di scrittura.



La linea operazione **op** specifica l'operazione che si vuole effettuare sulla cella:

op=1 lettura

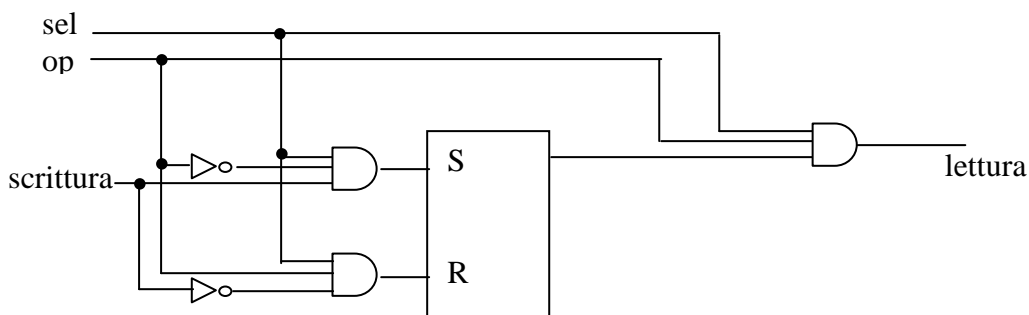
op=0 scrittura

La linea selezione **sel** indica se cella è stata selezionata per effettuare una operazione di lettura o scrittura:

sel=1 cella selezionata

sel=0 cella non selezionata.

Una cella di memoria può essere realizzata con un latch di tipo SR: le linee di ingresso *r* ed *s* permettono la scrittura, mentre la lettura è possibile tramite l'uscita del latch.

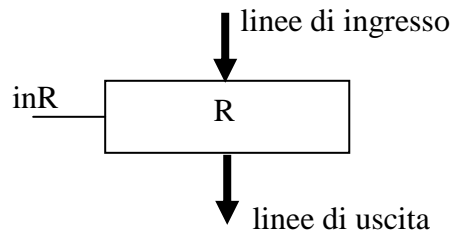


E' possibile realizzare celle di memoria con altri tipi di latch anche se non sempre il compito è banale; il problema deriva dal fatto che i segnali di ingresso (funzioni di eccitazione) sono sempre presenti. Per convincersi della difficoltà si provi a progettare una cella di memoria usando FF di

tipo D, tenendo conto che quando il segnale d vale 0 il latch non conserva il valore precedente ma assume quello del segnale di ingresso, cioè 0.

Registri

Una cella di memoria in grado di contenere tutti gli n bit una parola (una parola va considerata come un'unità indivisibile di informazione e tipicamente può essere composta di 8, 16, 32 o 64 bit) è detta registro ed è composta da n celle elementari.



La freccia in grassetto indica un insieme di linee, una per ogni cella elementare (flip-flop) componente il registro; quindi nel disegno a fianco è rappresentato un registro ad ingresso ed uscita paralleli.

La linea di selezione inR serve per selezionare tutte le n celle elementari da un bit che compongono il registro.

Interconnessione tra registri

Il trasferimento di informazioni tra registri viene realizzato tramite **reti di interconnessione**; tali reti permettono di portare l'informazione nei moduli di elaborazione o di memorizzazione.

In realtà è più appropriato parlare di duplicazione, piuttosto che di trasferimento, in quanto il contenuto del registro sorgente rimane immutato e una sua copia compare nel registro destinazione.

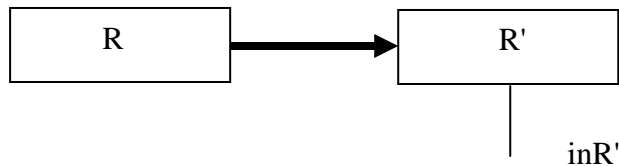
Distinguiamo quattro modalità di trasferimento tra registri, che si ottengono nei diversi casi in cui i registri sorgente e destinazione siano prefissati o variabili.

Nella seguente tabella sono riportati i nomi delle reti che realizzano questi quattro tipi di interconnessione.

	destinazione prefissata	destinazione variabile
sorgente prefissata	punto-punto: porte logiche o buffer tri-state	1-m con decodificatore
sorgente variabile	multiplexer	mesh e bus

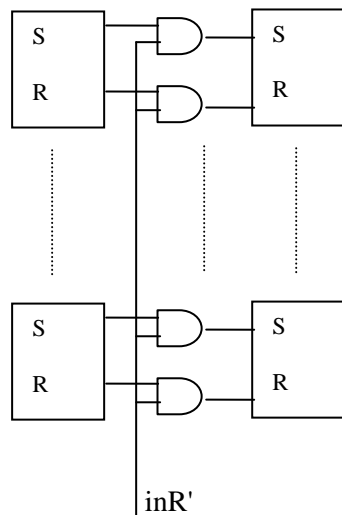
Sorgente e destinazione prefissata: interconnessione punto a punto

L'interconnessione punto a punto consente di trasferire il contenuto di un registro sorgente prefissato R in un registro destinazione prefissato R'.



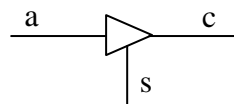
Ogni volta che si deve eseguire un trasferimento da R a R' la linea di selezione inR' va posta ad 1.

Nello schema seguente si vede che il segnale di controllo inR' agisce su porte logiche ed abilita il trasferimento.



In alternativa alle porte logiche si possono usare buffer tri-state.

Un **buffer tri-state** è un interruttore elettronico e viene schematizzato nel modo seguente:



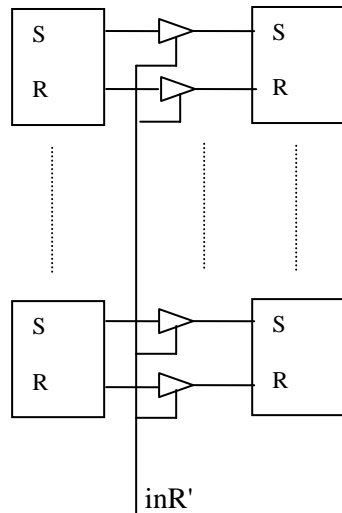
Quando il segnale di controllo s del buffer è:

- 0 l'impedenza fra ingresso e uscita del buffer è molto alta e l'interruttore è aperto: è come se il collegamento fra sorgente e destinazione fosse "tagliato";
- 1 l'impedenza è trascurabile, per cui a e c sono direttamente collegati e l'informazione presente sulla sorgente viene trasferita a destinazione:
 - o il valore di c è 0 se a è 0
 - o il valore di c è 1 se a è 1.

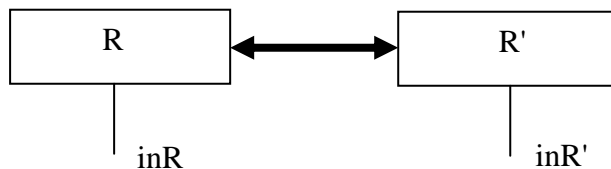
Il dispositivo può quindi assumere tre stati, da cui il nome:

- circuito aperto: $s = 0$
- circuito chiuso e uscita 0: se $s=1$ e $c=0$
- circuito chiuso e uscita 1: se $s=1$ e $c=1$.

Nello schema seguente il segnale inR' (che ha il ruolo del segnale indicato prima con s) viene utilizzato per controllare tutti i flip-flop componenti il registro.

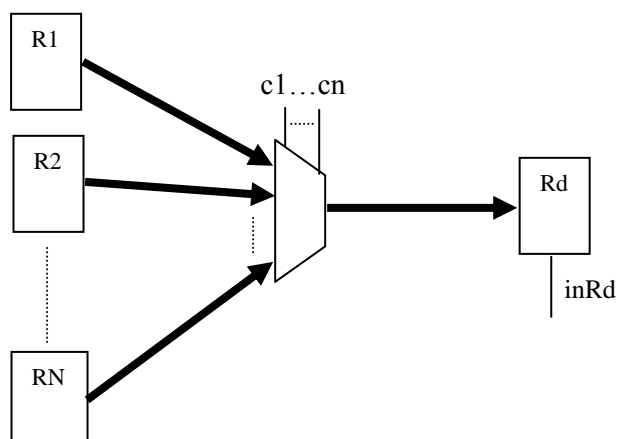


E' possibile realizzare una rete che permetta il trasferimento in modo bidirezionale tra R e R' , naturalmente dotando anche R di una linea di selezione.



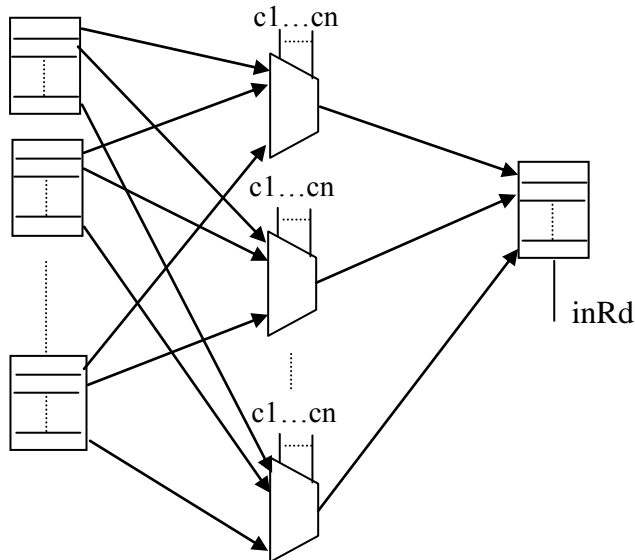
Sorgente variabile e destinazione prefissata: interconnessione di tipo multiplexer

Il registro sorgente può essere un qualsiasi registro R_i di un insieme di N registri, il registro destinazione R_d è prefissato.



I segnali di selezione del multiplexer sono $n = \log N$: c_1, \dots, c_n e forniscono la codifica binaria dell'indice i del registro R_i il cui contenuto deve essere copiato in R_d .

Il multiplexer nella figura sopra, con ingressi ed uscita in grassetto, indica un insieme di K multiplexer ognuno utilizzato per uno dei K flip-flop (cella di memoria elementare) componenti i registri.

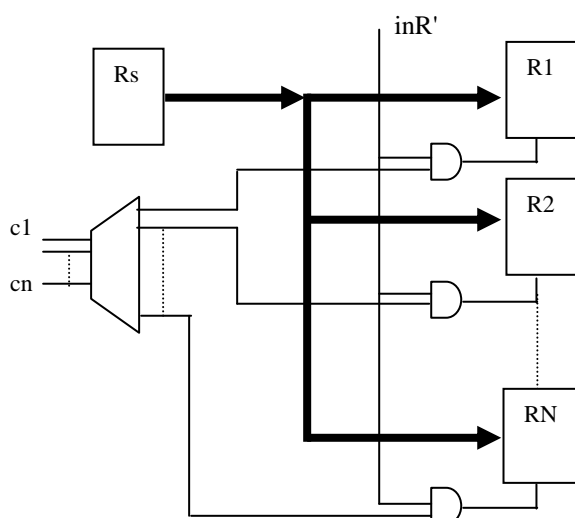


Il **primo** flip-flop di ogni registro sorgente è connesso con il **primo** multiplexer, l'uscita del multiplexer va al **primo** flip-flop di R_d ; il **secondo** flip-flop di ogni registro sorgente è connesso con il **secondo** multiplexer, l'uscita del multiplexer va al **secondo** flip-flop di R_d , e così via fino all'ultimo flip-flop.

Le linee di selezione c_1, \dots, c_n portano lo stesso valore a tutti i multiplexer poiché servono a specificare un singolo registro sorgente di cui ogni multiplexer seleziona uno specifico flip-flop (la sequenza binaria $c_1c_2\dots c_n$ fornisce l'indice i del registro sorgente da selezionare).

Sorgente prefissata e destinazione variabile: interconnessione 1-m con decodificatore

Il registro sorgente R_s è prefissato, mentre il registro destinazione R_d può essere un qualsiasi registro R_i di un insieme di N registri, che viene selezionato utilizzando un decodificatore.



La stessa linea di controllo inR' serve a controllare tutti gli N registri destinazione: se il segnale su tale linea vale 1 vuol dire che è abilitata l'operazione di scrittura su uno dei registri destinazione.

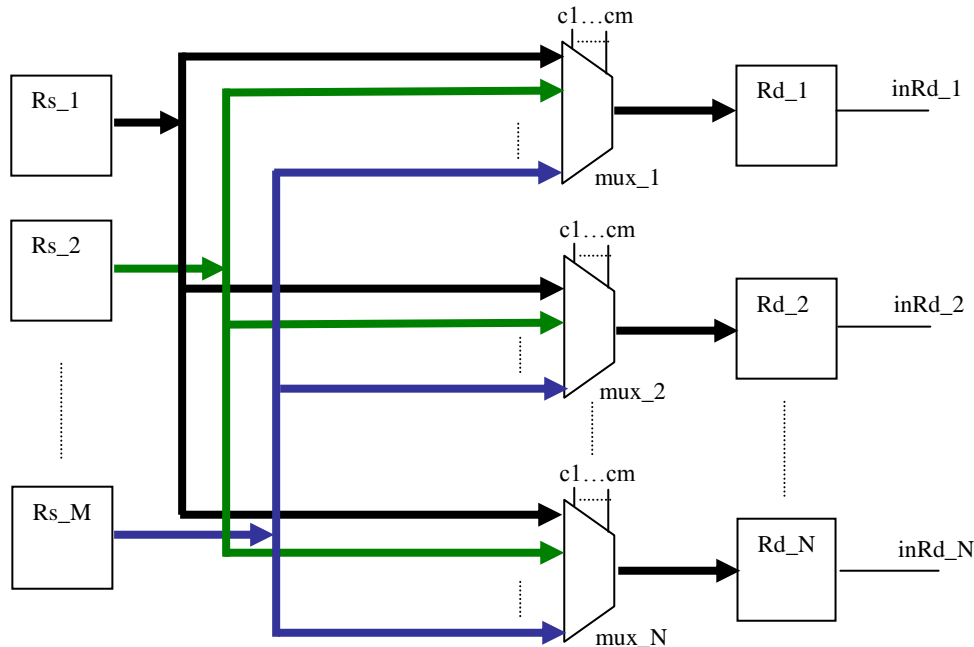
Il contenuto del registro sorgente R_s viene copiato nel registro R_j , con j codificato dalle n linee di selezione $c_1\dots c_n$ in ingresso al decodificatore, che rendono uguale ad 1 solo la j -ima uscita.

Le linee in grassetto indicano che c' è una linea per ogni FF (cella elementare componente il registro).

Sorgente variabile e destinazione variabile

1 - Interconnessione di tipo mesh

Il caso più complesso è quello in cui si richiede l'interconnessione tra M registri sorgente e N registri destinazione.



Per realizzare la rete occorrono N multiplexer, mux_i , ognuno per un registro destinazione.

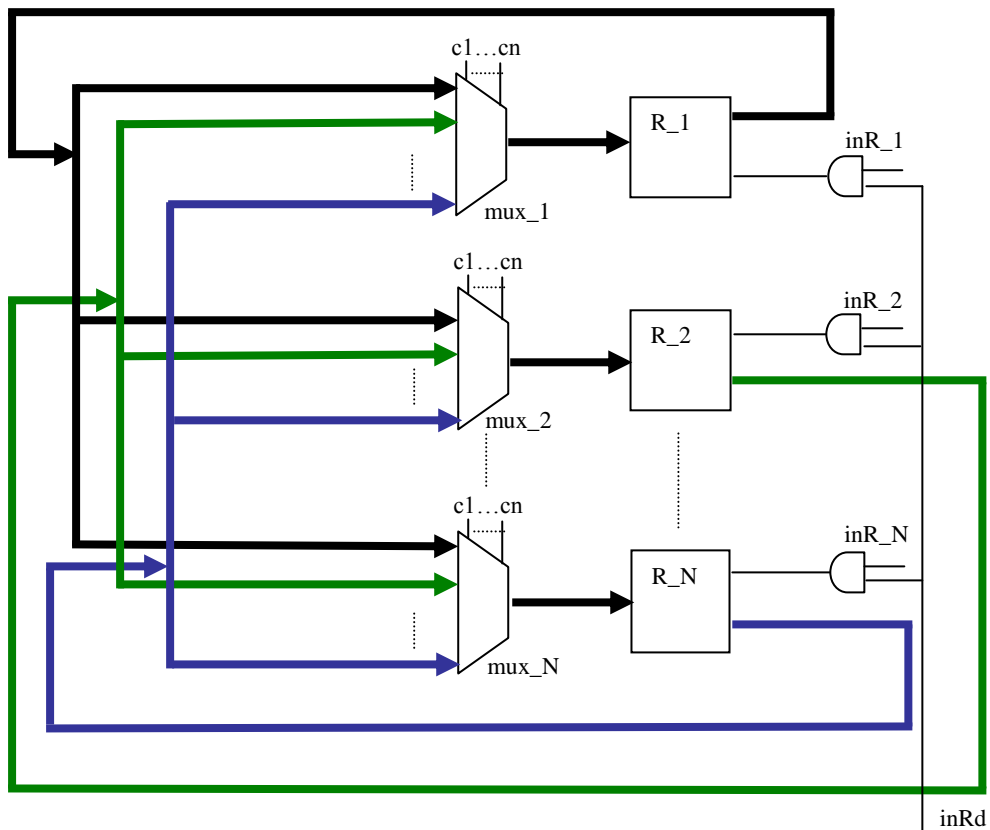
In realtà ognuno di questi multiplexer rappresenta K multiplexer, uno per ognuna delle K celle elementari che compongono i registri (tutti i registri sorgente e destinazione sono costituiti da K Flip-Flop), analogamente a quanto mostrato per il caso di sorgente variabile e destinazione prefissata.

Tutti i multiplexer sono controllati da $m = \log M$ linee di controllo, c_1, \dots, c_m , che permettono di selezionare uno degli M registri sorgente Rs_j fornendo l'indice j del registro sorgente il cui contenuto deve essere trasferito.

Il registro destinazione Rd_h , su cui deve essere trasferita (più precisamente copiata) l'informazione, viene abilitato alla scrittura dall'apposito segnale di controllo $inRd_h$.

Se si rimuove la distinzione tra registri sorgente e registri destinazione la rete di interconnessione si presenta come nello schema seguente.

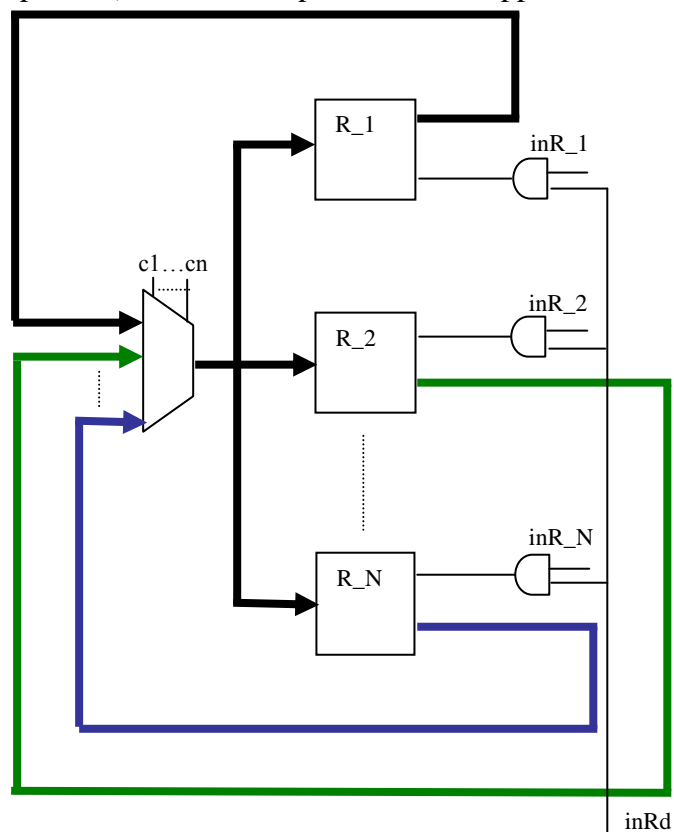
L'abilitazione alla scrittura su un registro si ottiene come AND tra $inRd$, che è il segnale di controllo che abilita alla scrittura su un registro destinazione, e inR_i segnale di controllo che abilita alla scrittura sullo specifico registro di destinazione i ; in particolare il segnale inR_i è uguale a 1 se l'indice del registro destinazione su cui si deve scrivere e il valore dell'indice i del mux sono uguali e ciò si verifica per un solo multiplexer (ad ogni multiplexer è associato un comparatore che esegue questo confronto che non viene mostrato nello schema.; l'uscita del comparatore i fornisce il valore della linea inR_i).



La realizzazione di una rete di tipo mesh non pone problemi di tipo concettuale, ma piuttosto di ordine pratico al crescere del numero di registri; infatti, l'utilizzazione di un numero molto elevato di registri richiederebbe un numero di porte tale da occupare buona parte dello spazio disponibile sul circuito integrato.

Come esempio si pensi di voler realizzare una rete di interconnessione per il trasferimento di informazioni tra 128 registri da 32 bit.

Un secondo tipo di rete mesh. Un modo per ovviare al problema appena descritto è quello di utilizzare un solo multiplexer (ossia un multiplexer che ne rappresenti K , uno per ogni FF).



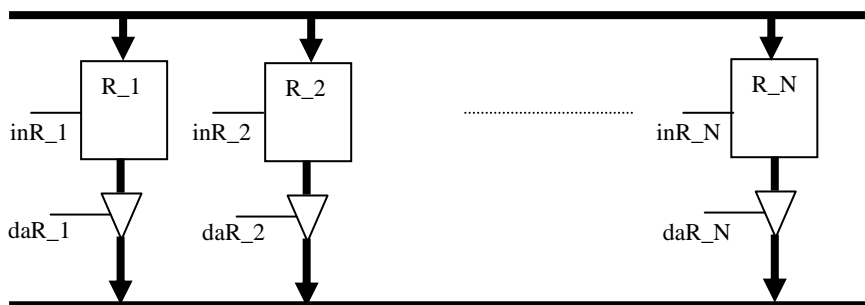
Il contenuto del registro sorgente R_i viene trasferito all'uscita del multiplexer se le linee di controllo $c_1 \dots c_n$ codificano l'indice i .

Il contenuto del registro sorgente viene trasferito nel registro destinazione R_j se le linee di controllo inR_j e $inRd$ sono entrambe 1. L'impostazione a 1 della linea inR_j si può ottenere tramite un decodificatore, impostando al valore j le sue linee di controllo.

Il vantaggio di tale schema rispetto al precedente riguarda il numero di porte. Lo schema che usa un multiplexer per ogni registro destinazione permette di effettuare più trasferimenti in parallelo, a patto, naturalmente, che le coppie tra cui avvengono i trasferimenti siano opportunamente disgiunte.

2 – Interconnessione tramite bus

E' possibile realizzare uno schema di collegamento più economico utilizzando buffer tri-state.



L'interconnessione viene realizzata utilizzando un singolo fascio di K linee (ove K è il numero di Flip-Flop componenti i registri) che prende il nome di bus.

Le entrate dei registri sono collegate direttamente al bus, mentre le uscite sono collegate al bus controllate da buffer tri-state.

Per effettuare il trasferimento da R_i ad R_j è sufficiente attivare la linea di controllo (lettura) dell' i -esimo buffer tri-state, daR_i , e la linea di selezione del j -esimo registro, inR_j .

I bus non permettono trasferimenti in parallelo

Utilizzazione di mesh e bus

All'interno di un microprocessore è presente un insieme di poche decine di registri veloci realizzati con tecnologia costosa mentre la memoria è costituita da milioni di registri realizzati con tecnologia più economica.

I registri del microprocessore sono interconnessi tra loro tramite opportune reti mesh, mentre il collegamento tra i registri contenuti nel microprocessore e quelli contenuti nei moduli di memoria è realizzato tramite bus. I bus sono largamente utilizzati all'interno di un'architettura poiché consentono di risolvere con il minimo numero di collegamenti punto-punto il problema dell'interconnessione tra registri.

La progettazione di una rete di interconnessione

I problemi da affrontare durante la progettazione di una rete di interconnessione sono principalmente di due tipi:

- stabilire quali sono i collegamenti necessari alla realizzazione dei trasferimenti richiesti
- progettare i circuiti che permettono di attivare correttamente le linee di controllo secondo le specifiche date

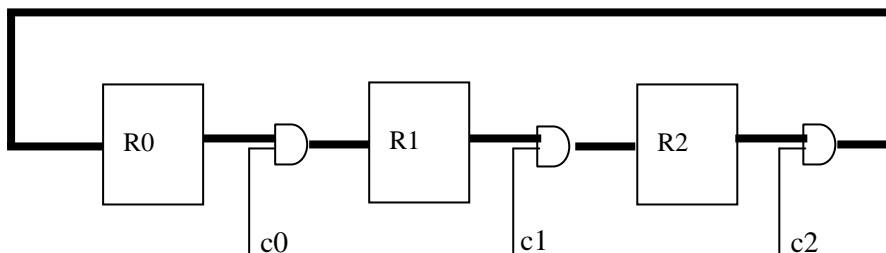
Come affrontare questi problemi viene illustrato con alcuni esempi.

Esempio 1

Si progetti un sistema di trasferimento fra i registri R0, R1, R2 tale che:

- R0 viene portato in R1 se $R0 = R1 + R2$ (ove + indica la somma logica cioè l'OR tra R1 e R2)
- R1 viene portato in R2 se $R0 < R1$
- R2 viene portato in R0 se $R2 > R1$

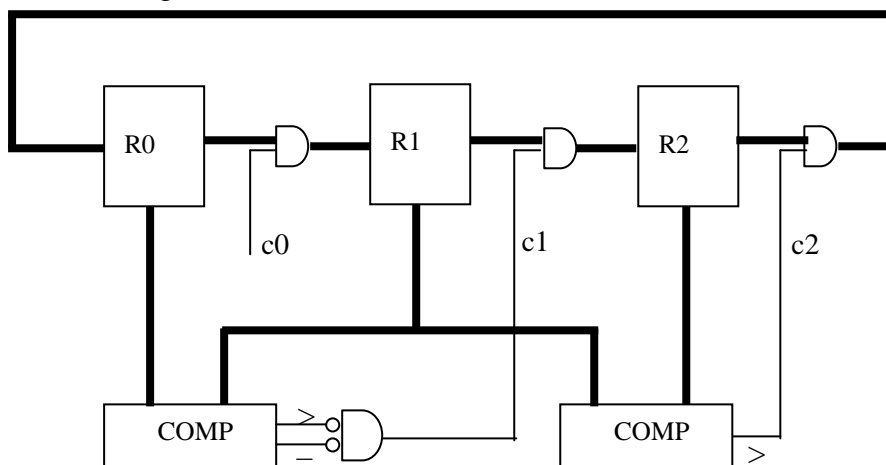
La rete di interconnessione si realizza mettendo insieme tre schemi di trasferimento uno a uno, come mostrato nel disegno seguente:



Alternativamente alle porte logiche AND si possono usare buffer tri-state.

Bisogna poi progettare la parte circuitale che permette di ottenere i valori dei segnali di controllo c0, c1 e c2.

Le condizioni poste per ottenere il trasferimento da R1 a R2 (linea di controllo c1) e da R2 a R0 (linea di controllo c2) permettono di utilizzare le uscite di un comparatore per ottenere c1 e c2, come mostrato nel disegno sotto:



Per progettare la parte combinatoria che permette di ottenere $c0$ seguiamo il metodo di sintesi visto per le reti combinatorie.

Siano $x_n \dots x_1$, $y_1 \dots y_n$ e $z_1 \dots z_n$ i bit contenuti in $R0$, $R1$ e $R2$ rispettivamente.

Poiché l'uguaglianza tra $x_n \dots x_1$ e l'OR tra $y_1 \dots y_n$ e $z_1 \dots z_n$ si ottiene se è vera l'uguaglianza bit a bit, stendiamo la tabella di verità che ci permette di trovare la funzione $c0_i$ tale che $c0_i$ è uguale a 1 se $x_i = y_i$ OR z_i . Le prime tre colonne ci permettono di elencare tutte le combinazioni dei bit i -esimi x_i , y_i e z_i , mentre nella quarta colonna mettiamo i valori della funzione $c0_i$.

x_i	y_i	z_i	$c0_i$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Dalla mappa di Karnaugh si ottiene la seguente espressione per $c0_i$:

$$c0_i = x_i y_i z_i + x_i z_i + x_i y_i$$

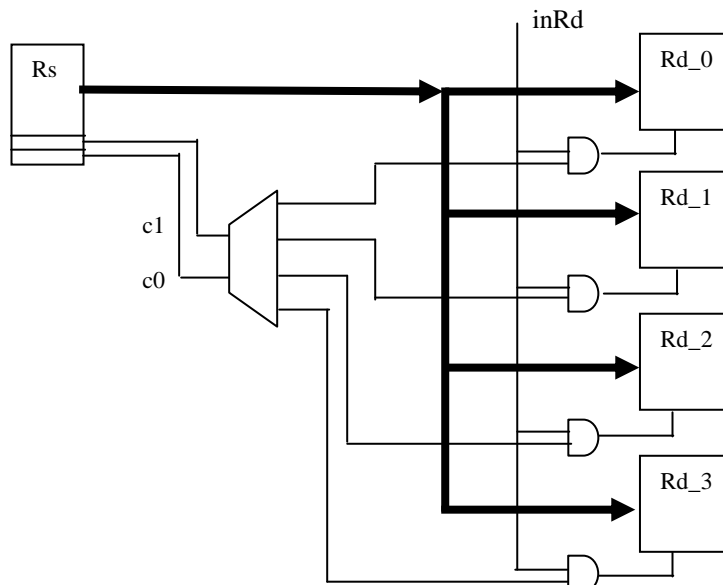
Mettendo in AND tutte le condizioni $c0_i$ ottenute con il metodo appena illustrato si ottiene la condizione c_0 .

Esempio 2

Sia R_s un registro sorgente e siano Rd_0 , Rd_1 , Rd_2 e Rd_3 registri destinazione.

Si progetti la rete di interconnessione tale che quando $inRd$ vale 1 il contenuto di R_s viene trasferito in Rd_j ove j coincide con i due bit meno significativi di R_s . Si mostri il progetto fino al dettaglio di porte logiche.

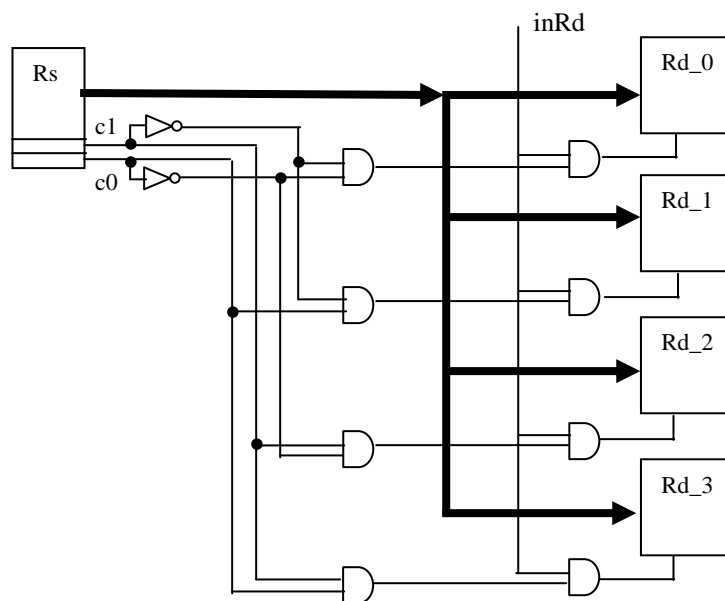
Si tratta di una rete di interconnessione uno a molti tra un registro sorgente e più registri destinazione, il cui schema generale è:



Dovendo dettagliare fino a livello di porte logiche esplicitiamo il decodificatore.
La tabella di verità del decodificatore si presenta nel seguente modo:

c0	c1	r0	r1	r2	r3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Lo schema circuitale è quindi:



Esempio 3

Progettare una rete di interconnessione "multi-molti" (ovvero tra più sorgenti e più destinazioni), che consenta di caricare il contenuto di 2 fra N registri R1...RN su 1 fra M dispositivi di elaborazione E1....EM a due ingressi (la comunicazione avviene fra 2 su N sorgenti ed 1 su M destinazioni)

Disegnare lo schema a blocchi evidenziando tutti i segnali di controllo necessari per:

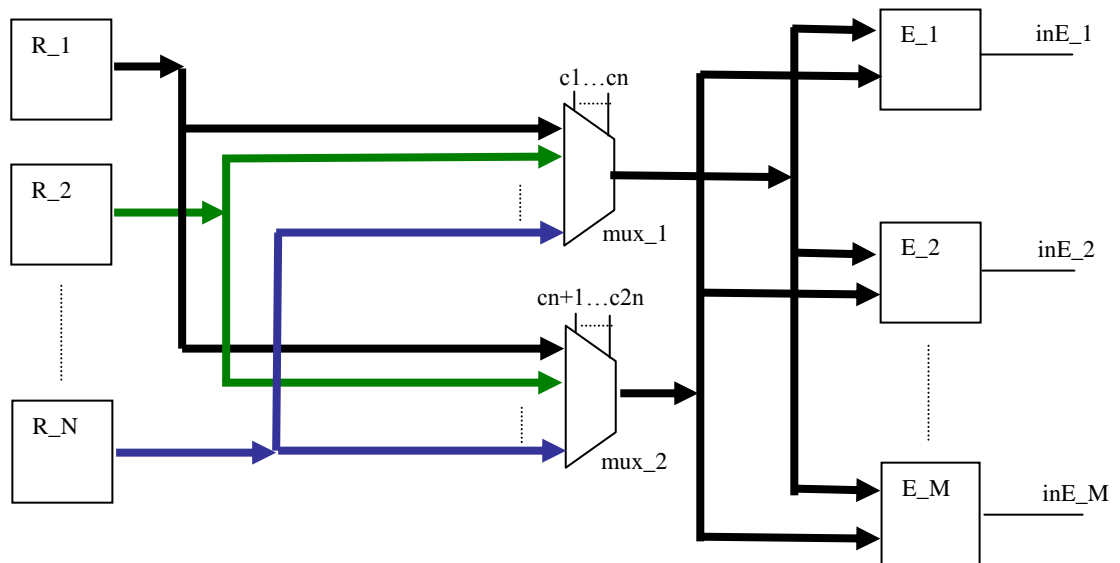
- selezionare 2 fra N registri sorgente, ovvero i due operandi (OP1 e OP2)
- convogliare i 2 operandi in ingresso ad uno fra M dispositivi di elaborazione, Ei.

Disegnare lo schema circuitale (quindi con tutti i dettagli fino al livello di FF , porte logiche, numero e ruolo dei segnali di controllo Ci necessari) per il caso di:

- 3 registri sorgente a due bit (N=3), con FF di tipo JK
- 2 dispositivi destinazione (M=2), di cui uno sia un sommatore aritmetico, e l'altro un circuito logico che esegua l'AND fra i due registri sorgente selezionati.

(Ad esempio, se OP1=R1 e OP2=R3 e Ei=AND, si dovrà eseguire: R1 AND R3)

Lo schema generale si presenta nel seguente modo:



C'è un multiplexer per ognuno degli operandi da inviare alle unità di elaborazione:

- mux1 seleziona il primo operando tra gli N possibili registri sorgente per mezzo dei segnali di controllo sono $c_1 \dots c_n$
- mux2 seleziona il secondo operando, sempre tra gli N possibili registri sorgente, per mezzo dei segnali di controllo sono $c_{n+1} \dots c_{2n}$

e, naturalmente, $n = \log N$.

Ogni multiplexer con fasci di linee in entrata e in uscita rappresenta un insieme di tanti multiplexer, quante sono le linee del fascio, con linee singole in entrata in uscita, attivati dagli stessi segnali di controllo.

Ogni unità di elaborazione è dotata di un segnale di controllo in_{E_j} per abilitare la lettura degli operandi presenti sulle linee di ingresso all'unità.

Lo schema circuitale dettagliato per il caso specifico è dotato di quattro multiplexer (con linee singole):

- mux0 seleziona il bit meno significativo del primo operando
- mux1 seleziona il bit più significativo del primo operando
- mux2 seleziona il bit meno significativo del secondo operando
- mux3 seleziona il bit più significativo del secondo operando

Quindi mux0 e mux1, dovendo selezionare il primo operando, sono controllati dalle linee c_1 e c_2 , mentre mux2 e mux3, selezionano il secondo operando utilizzando le linee di controllo c_3 e c_4 .

L'operazione da eseguire viene scelta ponendo le linee add e and opportunamente a 1.

Notare che c_{in} e c_{out} sono rispettivamente il riporto in entrata e in uscita all'adder.

Lo schema circuitale è il seguente:

