Appunti sull'interconnessione tra registri

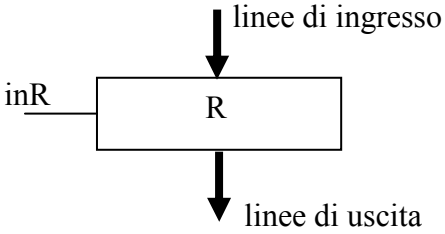
a cura di A. Massini

revisione di Daniele Gorla

Registri

Una cella di memoria in grado di contenere tutti i *k* 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 *k* celle di memoria elementari, ognuna contenente un bit di informazione. In queste dispense (salvo ove diversamente specificato) assumeremo che le celle di memoria elementari siano ottenute tramite flip-flop (FF) di tipo SR, ma tutta la trattazione non dipende da questa scelta.

Schematicamente, rappresenteremo un registro come segue:



La freccia in grassetto indica un insieme di *k* 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 *k* celle elementari da un bit che compongono il registro.

Interconnessione tra registri

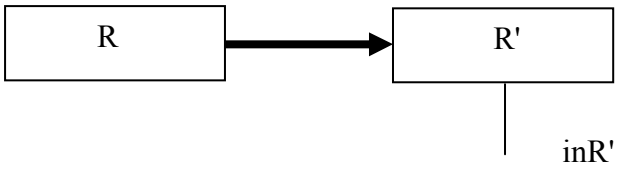
Il trasferimento di informazioni tra registri viene realizzato tramite *reti di interconnessione*; tali reti permettono di portare l'informazione nei moduli di elaborazione o di memorizzazione. In realtà è più appropriato parlare di duplicazione, piuttosto che di trasferimento, in quanto il contenuto 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.

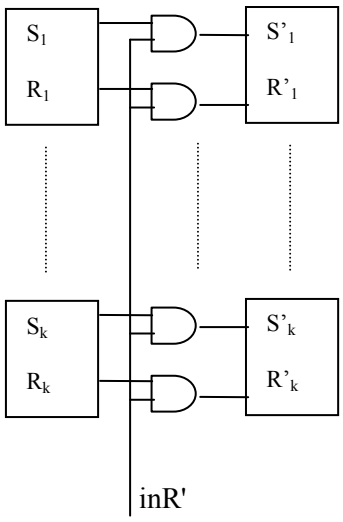
	destinazione prefissata	**destinazione variabile**
sorgente prefissata	punto-punto (porte logiche o buffer tri-state)	1-m con decodificatore
sorgente variabile	Multiplexer	mesh e bus

Sorgente e destinazione prefissata: interconnessione punto a punto

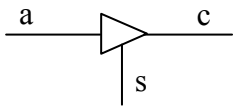
L'interconnessione punto a punto consente di trasferire il contenuto di un registro sorgente prefissato R in un registro destinazione prefissato R'.



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 é un interruttore elettronico e viene schematizzato nel modo seguente:



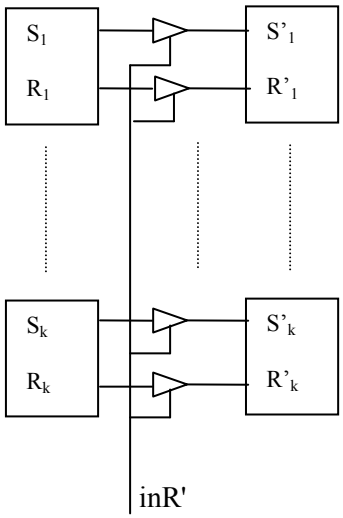
Quando il segnale di controllo s del buffer è:

- 0 l'impedenza fra ingresso e uscita del buffer é molto alta e l'interruttore è aperto: é 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:
    - il valore di c è 0 se a è 0
    - il valore di c è 1 se a è 1.

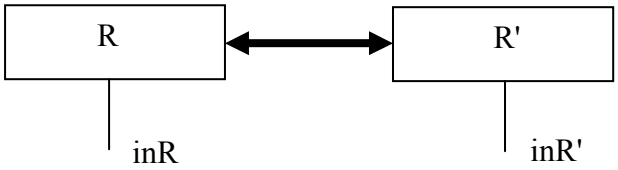
Il dispositivo può quindi assumere tre stati, da cui il nome:

- 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 buffer posti tra i flip-flop dei due registri.



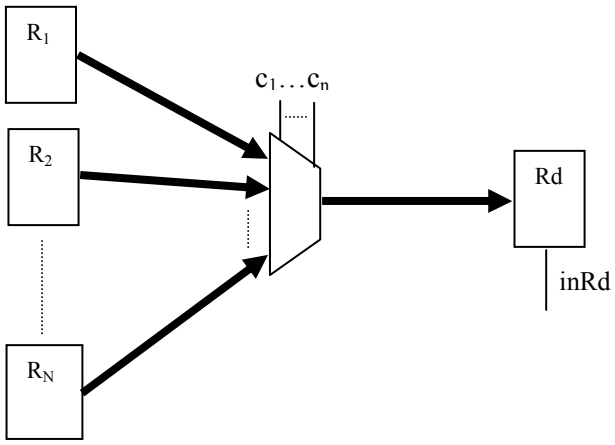
E' possibile realizzare una rete che permetta il trasferimento in modo bidirezionale tra R e R', naturalmente dotando anche R di una linea di selezione.



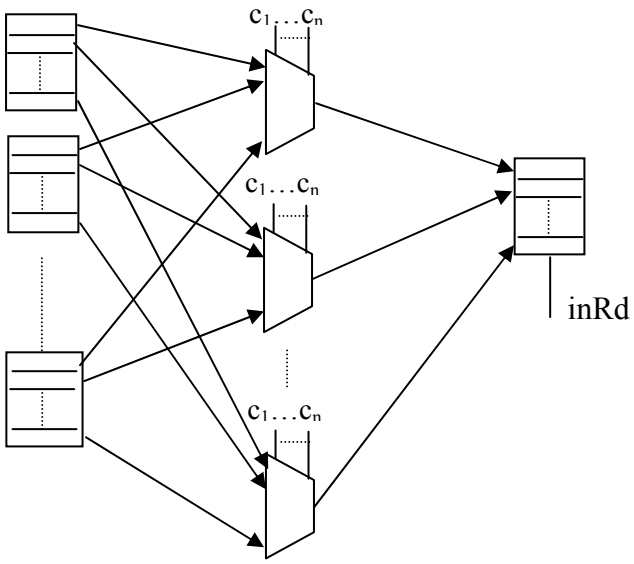
Si lascia per esercizio al lettore il completamento con tutti i dettagli.

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 Rd è prefissato.



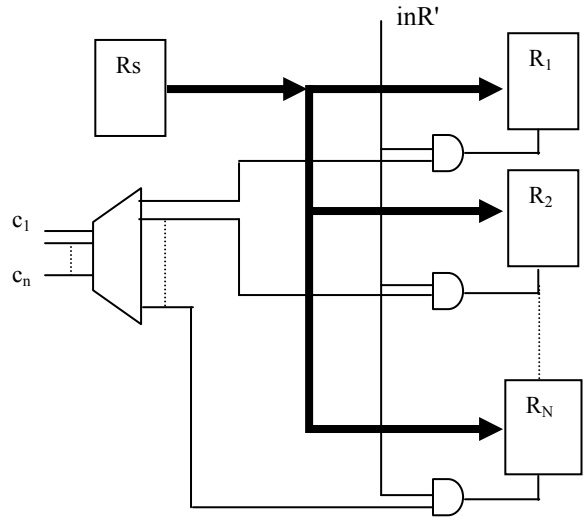
I segnali di selezione del multiplexer sono c_1, …, c_n (dove n è la parte intera superiore di log N) e forniscono la codifica binaria dell'indice i del registro R_i il cui contenuto deve essere copiato in Rd. Il multiplexer nella figura sopra, con ingressi ed uscita in grassetto, indica un insieme di *k* multiplexer ognuno utilizzato per uno dei K flip-flop (cella di memoria elementare) componenti i registri:



Il primo FF di ogni registro sorgente è connesso con il primo multiplexer, l'uscita del multiplexer va al primo FF di Rd; il secondo FF di ogni registro sorgente è connesso con il secondo multiplexer, l'uscita del multiplexer va al secondo FF di Rd; e così via fino all'ultimo FF. Le linee di selezione c_1, …, 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 FF (la sequenza binaria c_1 c_2…c_n fornisce l'indice i del registro sorgente da selezionare).

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

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

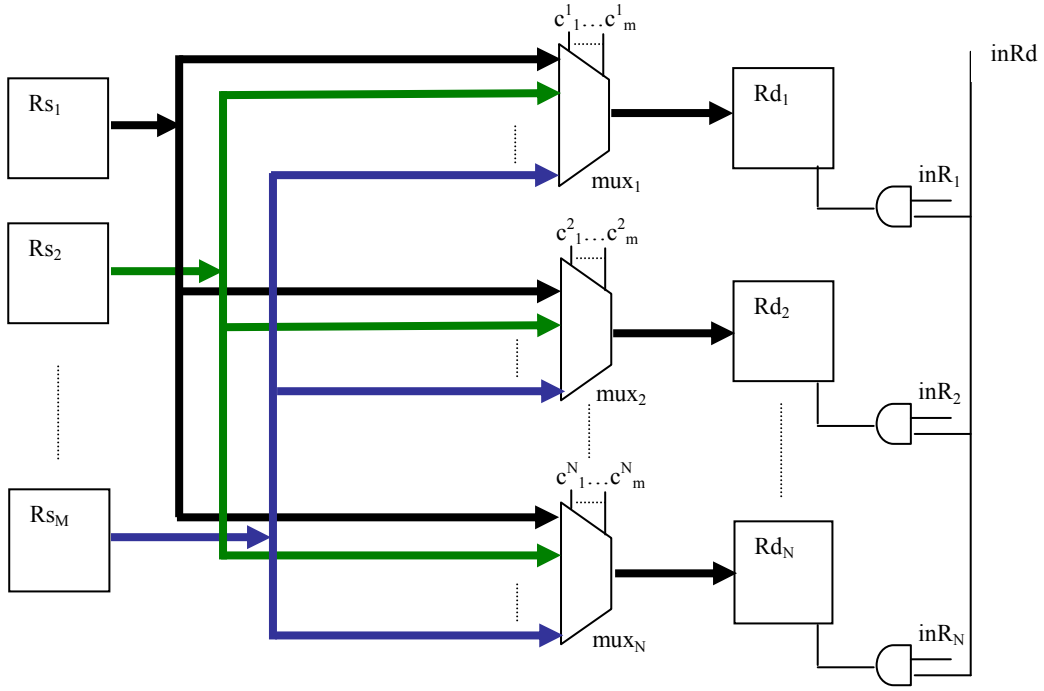


La stessa linea di controllo inR' serve a controllare tutti gli N registri destinazione: se il segnale su tale linea vale 1 vuol dire che è abilitata l'operazione di scrittura su uno dei registri destinazione. Il contenuto del registro sorgente Rs viene copiato nel registro R_j, con j codificato dalle n linee di selezione c_1...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

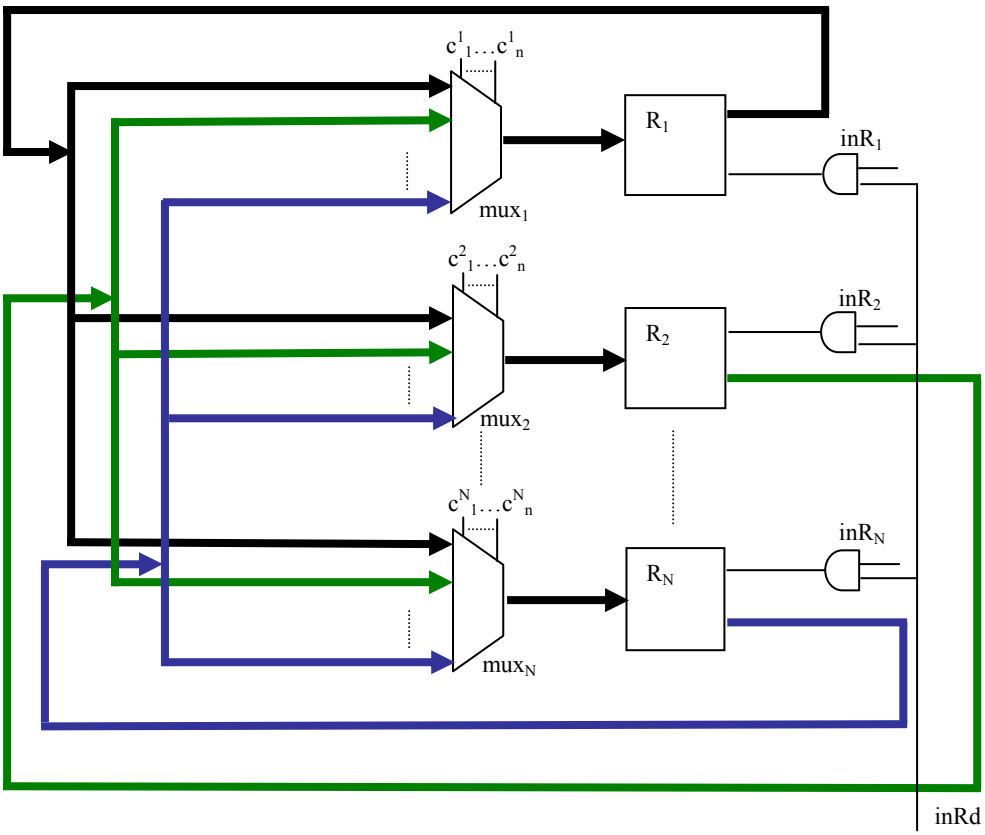
1 - Interconnessione di tipo mesh

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* FF), analogamente a quanto mostrato per il caso di sorgente variabile e destinazione prefissata**.** Il multiplexer *i* è controllato dalle linee di controllo $c^i_1 \ldots c^i_m$ (dove m è la parte intera superiore di log 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 $inRd_h$, posto in AND con un segnale globale inRd che abilita i trasferimenti.

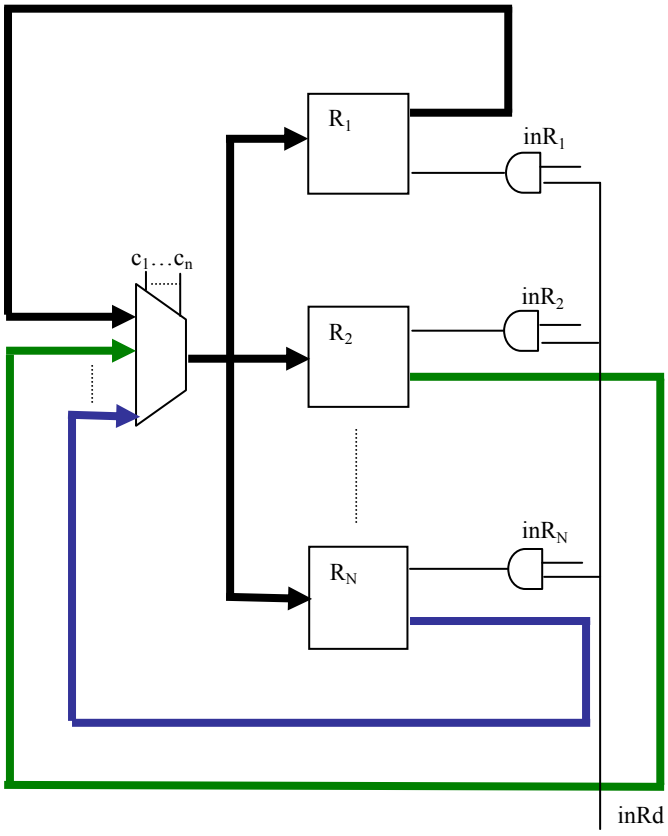
Se si rimuove la distinzione tra registri sorgente e registri destinazione, la rete di interconnessione si presenta come nello schema seguente.



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*. Invece, la selezione del registro che va copiato in R_i viene effettuata tramite le linee di controllo a mux_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).

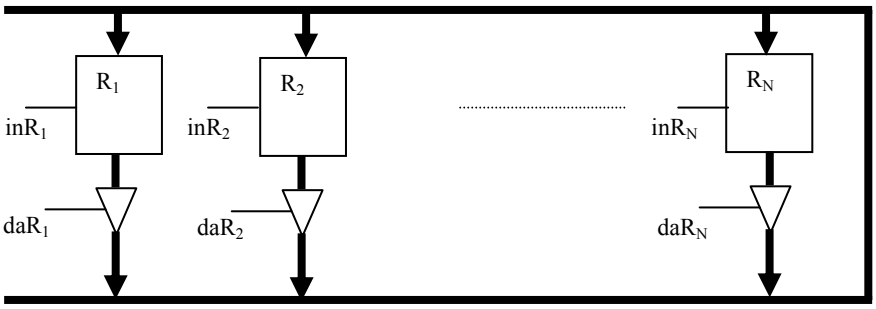


Il contenuto del registro sorgente R_i viene trasferito all'uscita del multiplexer se le linee di controllo $c_1 \ldots c_n$ codificano l'indice *i*. Il contenuto del registro sorgente viene trasferito nel registro destinazione R_j se le linee di controllo inR_j e inRd sono entrambe 1. L'impostazione 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. Il prezzo da pagare è che così si perde la possibilità di effettuare più trasferimenti in parallelo, cosa che invece permetteva lo schema precedente (con un multiplexer per ogni registro destinazione).

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 (dove *k* è il numero di FF 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 tramite un buffer tri-state. Per effettuare il trasferimento da R_i ad R_j è sufficiente attivare la linea di controllo daR_i dell'*i*-esimo buffer tri-state e la linea di selezione inR_j del *j*-esimo registro.

Ovviamente, il bus non permette 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:

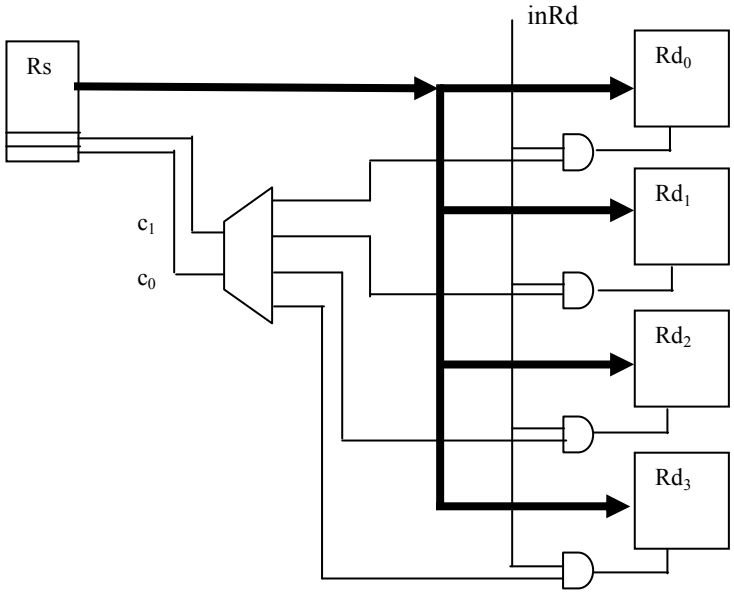
- stabilire quali sono i collegamenti necessari alla realizzazione dei trasferimenti richiesti
- progettare i circuiti che permettono di attivare correttamente le linee di controllo secondo le specifiche date

Come affrontare questi problemi viene illustrato con alcuni esempi.

Esempio 1. Sia Rs un registro sorgente e siano Rd_0, Rd_1, Rd_2 ed Rd_3 quattro registri destinazione. Si progetti la rete di interconnessione tale che quando inRd vale 1 il contenuto di Rs viene trasferito in Rd_j, dove *j* è codificato in binario dai due bit meno significativi di Rs. Si mostri il progetto fino al dettaglio di porte logiche.

Svolgimento

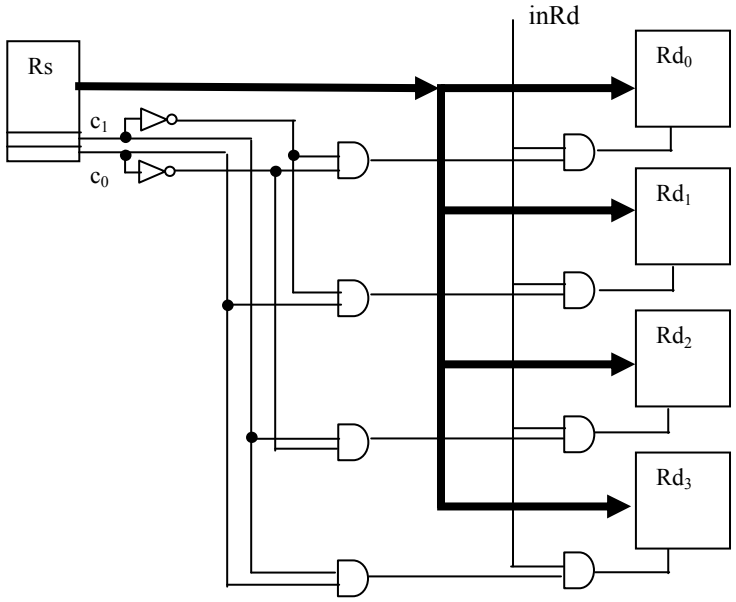
Si tratta di una rete di interconnessione uno a molti tra un registro sorgente e più registri destinazione, il cui schema generale è:



Dovendo dettagliare fino a livello di porte logiche, esplicitiamo il decodificatore. La tabella di verità del decodificatore si presenta nel seguente modo, dove r_0, r_1, r_2 ed r_3 sono le uscite del decodificatore, usate poi per costruire (in AND con il segnale inRd) il segnale di abilitazione dei registri Rd_0, Rd_1, Rd_2 ed Rd_3, rispettivamente:

c_0	c_1	r_0	r_1	r_2	r_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

Lo schema circuitale è quindi:

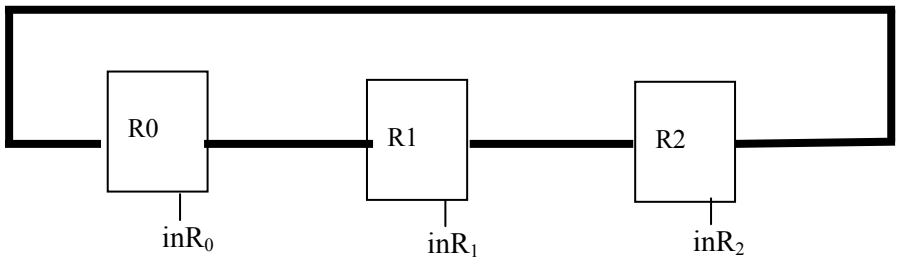


Esempio 2. Si progetti un sistema di trasferimento fra i registri R_0, R_1 ed R_2 tale che:

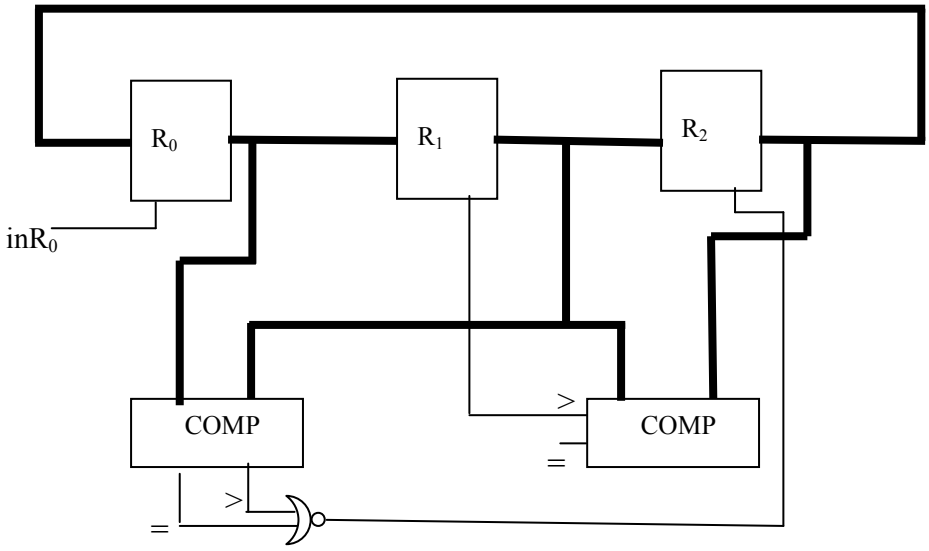
- R_0 viene portato in R_1 se $R_0 = R_1 \mid R_2$ (dove | indica l'OR bit a bit tra R_1 ed R_2)
- R_1 viene portato in R_2 se $R_0 < R_1$
- R_2 viene portato in R_0 se $R_1 > R_2$

Svolgimento

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



Bisogna poi progettate la parte circuitale che permette di ottenere i valori dei segnali di controllo inR_0, inR_1 e inR_2. Le condizioni poste per ottenere il trasferimento da R1 a R2 (linea di controllo inR_2) e da R2 a R0 (linea di controllo inR_0) suggeriscono di utilizzare le uscite di due comparatori, come mostrato nel disegno sotto:



Per progettare la parte combinatoria che permette di ottenere inR_0 seguiamo il metodo di sintesi visto per le reti combinatorie. Siano $x_1 \ldots x_k$, $y_1 \ldots y_k$ e $z_1 \ldots z_k$ i bit contenuti in R_0, R_1 ed R_2 rispettivamente. Ovviamente, l'uguaglianza tra R_0 e l'OR bit a bit tra R_1 ed R_2 si ottiene verificando l'uguaglianza bit a bit; pertanto, stendiamo la tabella di verità che ci permette di trovare la funzione c_i tale che $c_i = 1$ se e solo se $x_i = y_i$ OR z_i:

x_i	y_i	z_i	c_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 c_i:

$c_i = \overline{x_i}\,\overline{y_i}\,\overline{z_i} + x_i z_i + x_i y_i$

Mettendo in AND tutte le condizioni c_i ottenute con il metodo appena illustrato, si ottiene la condizione inR_0.

Esempio 3. Progettare una rete di interconnessione "molti-molti", che consenta di caricare il contenuto di 2 fra N registri R_1...R_N su 1 fra M dispositivi di elaborazione E_1....E_M 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:

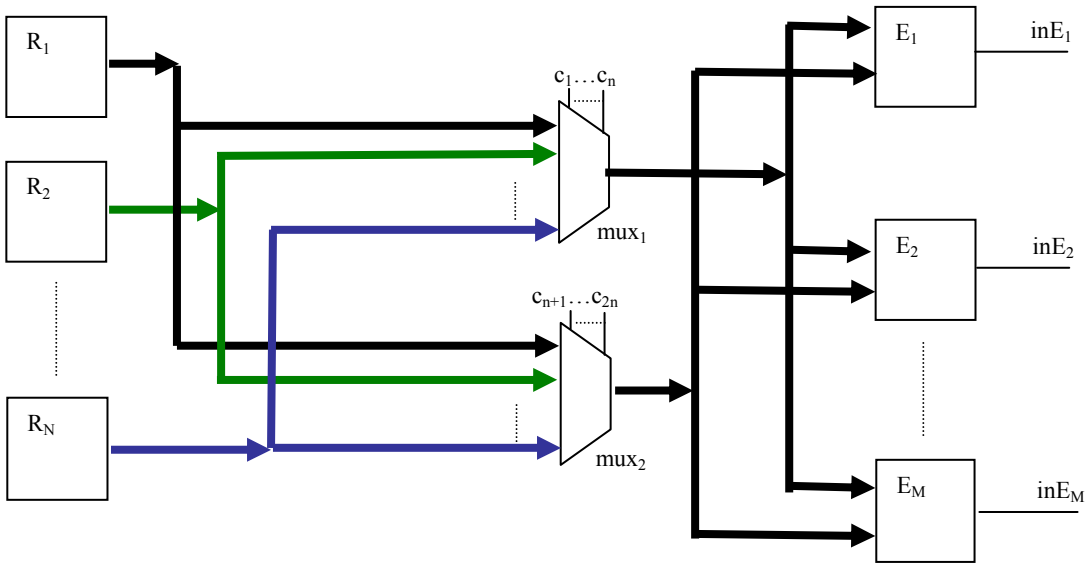
- selezionare 2 fra gli N registri sorgente (ovvero i due operandi)
- convogliare i 2 operandi in ingresso ad uno fra gli M dispositivi di elaborazione.

Disegnare poi lo schema circuitale (quindi con tutti i dettagli fino al livello di FF, porte logiche, numero e ruolo dei segnali di controllo 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 bit a bit fra i due registri sorgente selezionati.

Soluzione

Lo schema generale si presenta nel seguente modo:



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

- mux_1 seleziona il primo operando tra gli N possibili registri sorgente per mezzo dei segnali di controllo c_1….c_n;
- mux_2 seleziona il secondo operando, sempre tra gli N possibili registri sorgente, per mezzo dei segnali di controllo c_{n+1}…c_{2n}.

dove, naturalmente, n è la parte intera superiore di 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 inE_j per abilitare la lettura degli operandi presenti sulle linee di ingresso all'unità. *j*.

Lo schema circuitale dettagliato per il caso specifico è dotato di quattro multiplexer (con linee singole):
- mux_0 seleziona il bit meno significativo del primo operando
- mux_1 seleziona il bit più significativo del primo operando
- mux_2 seleziona il bit meno significativo del secondo operando
- mux_3 seleziona il bit più significativo del secondo operando

Quindi mux_0 e mux_1, dovendo selezionare il primo operando, sono entrambi controllati dalle stesse linee c_1 e c_2, mentre mux_2 e mux_3, selezionando il secondo operando, utilizzando entrambi le stesse linee di controllo c_3 e c_4. L'operazione da eseguire viene scelta ponendo le linee add (corrispondente ad inE_1) e and (corrispondente ad inE_2) opportunamente a 1. Infine, rip è usata per segnalare l'eventuale riporto in uscita dall'addizionatore.

Lo schema circuitale risultante è il seguente: