Esercizi svolti e da svolgere
sugli argomenti trattati nella lezione 25

Esercizi svolti

Es. 1. Si progetti in dettaglio il circuito che, dati quattro registri sorgente S_i e quattro registri destinazione D_i (per i = 1,…,4), consenta i seguenti trasferimenti in parallelo:

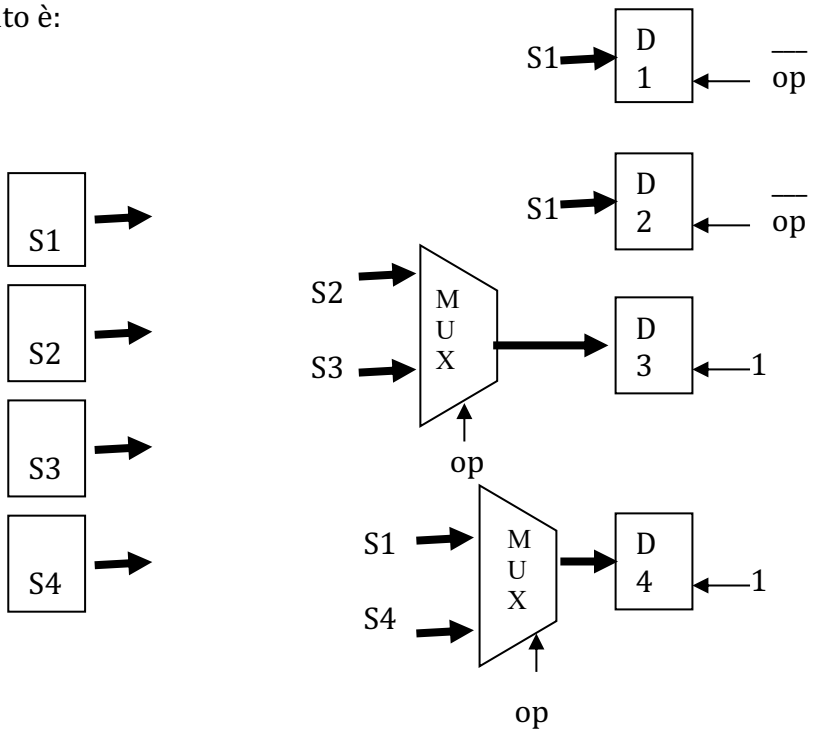
(a) $S_1 \rightarrow D_1, D_2, D_4$ e $S_2 \rightarrow D_3$

(b) $S_3 \rightarrow D_3$ e $S_4 \rightarrow D_4$

Identificando D_1 con D_3 e D_2 con D_4 (cioè il circuito modificato ha come registri destinazione solo D_1 e D_2), i trasferimenti richiesti sarebbero ancora leciti? Perché?

SOLUZIONE:

Anzitutto, osserviamo che D1 e D2 ricevono solo da S1, mentre D3 e D4 ricevono da S2 o S3 e da S1 o S4, rispettivamente. Pertanto, si può fare una connessione punto-a-punto tra S1 e D1 e tra S1 e D2; invece, D3 e D4 richiederanno un MUX 2-a-1 per poter selezionare la loro entrata. In base all'operazione richiesta (come specificato dal bit *op*), la rete di controllo attiverà o meno i registri D1 e D2 per la scrittura (op = 0 attiva i registri, op = 1 li disattiva; quindi in_D1 e in_D2 saranno la negazione di op). I registri D3 e D4 sono invece sempre abilitati alla scrittura (in_D3 e in_D4 settati a 1); quello che serve è un'opportuna selezione del registro mittente per ognuno dei due registri (cioè, un opportuno bit di controllo per i due MUX): quando op = 0, deve selezionare S2 per D3 e S1 per D4; quando op = 1, deve selezionare S3 per D3 e S4 per D4. Quindi, per entrambe i MUX il segnale di controllo è il bit op.
Pertanto, il circuito è:



Nella rete con uniche destinazioni i registri D1 e D2, il trasferimento (b) sarebbe ancora lecito mentre il trasferimento (a) no: infatti, nel primo caso non ci sono conflitti poiché registri

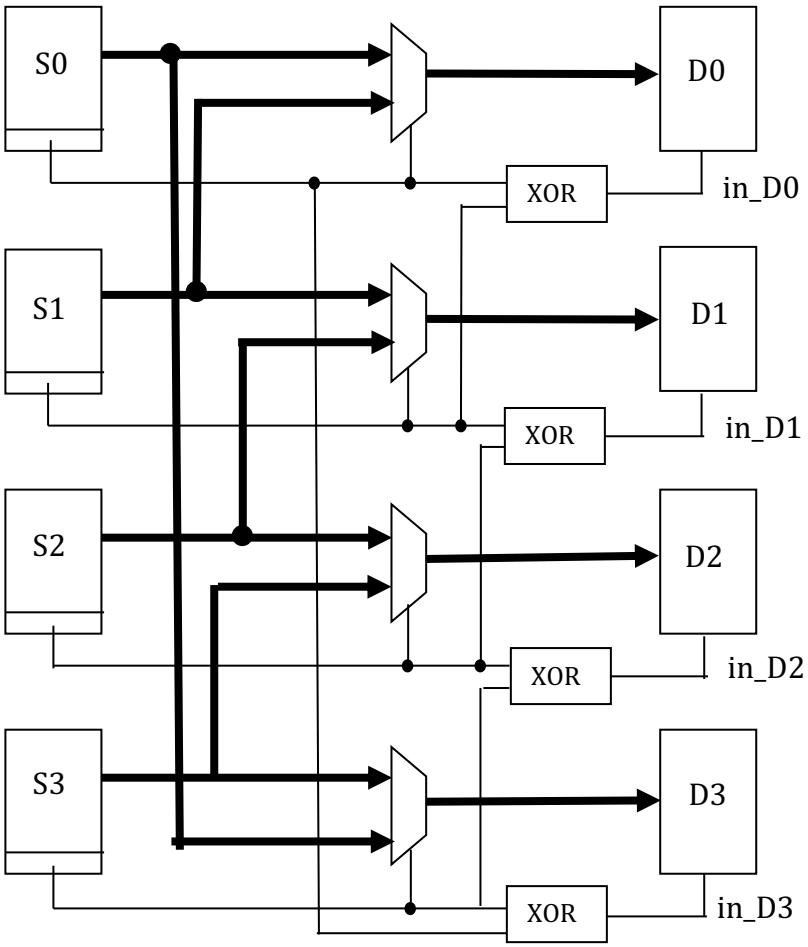
sorgente e destinazione sono disgiunti, mentre nel secondo caso c'è conflitto sul registro D1 che dovrebbe ricevere in input contemporaneamente il dato proveniente da S1 e da S2.

Es. 2. Si progetti una rete di interconnessione tra 4 registri sorgente S0, S1, S2, S3 e 4 registri destinazione D0, D1, D2, D3 nella quale siano possibili i seguenti trasferimenti paralleli: se Si è pari allora Si viene copiato in Di; altrimenti $S_{(i+1) \bmod 4}$ viene copiato in Di. Si abbia che il registro Di è abilitato alla scrittura se e solo se Si + $S_{(i+1) \bmod 4}$ è dispari. Specificare il valore di **tutti** i segnali di controllo presenti nello schema e disegnare lo schema. Quale struttura può essere utilizzata se non si richiede che i trasferimenti avvengano in parallelo?

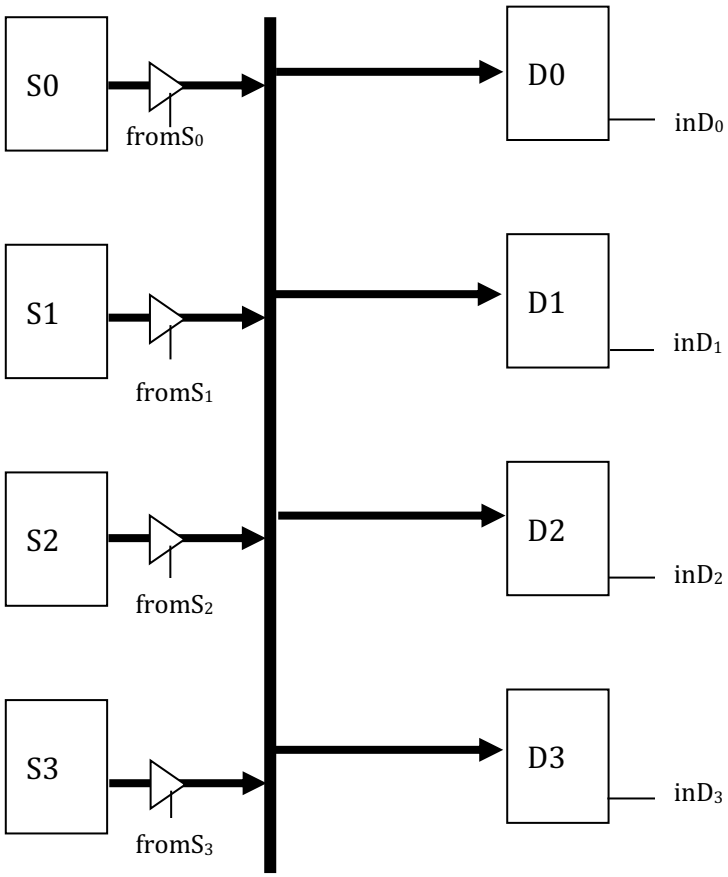
SOLUZIONE:

La rete di interconnessione richiesta è una rete molti a molti. Si deve avere un multiplexer per ogni registro destinazione che selezioni una delle due possibili entrate tra Si e $S_{(i+1) \bmod 4}$ sulla base di parità del registro Si, utilizzando quindi un solo segnale di controllo. Per verificare se il contenuto di un registro è pari basta controllare il suo bit meno significativo: se il bit è zero allora il contenuto di Si è pari, altrimenti è dispari. Il valore del bit meno significativo viene quindi usato come segnale di controllo per il multiplexer. Il segnale in_Di si ottiene in modo molto semplice osservando che la somma tra due numeri è dispari se e solo se esattamente uno dei due addendi è dispari ed utilizzando quindi una porta XOR le cui entrate sono i bit meno significativi di Si e $S_{(i+1) \bmod 4}$. I trasferimenti da realizzare sono:

D0 è abilitato se **S0+S1** è dispari**,** riceve **S0** se S0 è pari**, S1** se S0 è dispari
D1 è abilitato se **S1+S2** è dispari**,** riceve **S1** se S1 è pari**, S2** se S1 è dispari
D2 è abilitato se **S2+S3** è dispari**,** riceve **S2** se S2 è pari**, S3** se S2 è dispari
D3 è abilitato se **S3+S0** è dispari**,** riceve **S3** se S3 è pari**, S0** se S3 è dispari



Senza trasferimenti in parallelo si può usare un bus. Bisogna però assicurare l'accesso esclusivo al bus, quindi dobbiamo selezionare quale trasferimento si vuole effettuare (tramite 2 segnali di input, chiamati *a1* e *a0*, con 00 che abilita i primo trasferimento, 01 il secondo, 10 il terzo, e 11 il quarto) o sequenzializzare i trasferimenti (usando un contatore modulo 4, che produce i bit *a1* e *a0*). In entrambe i casi, l'interconnessione è la seguente:



dove (denotiamo con *li* il LSB di Si)

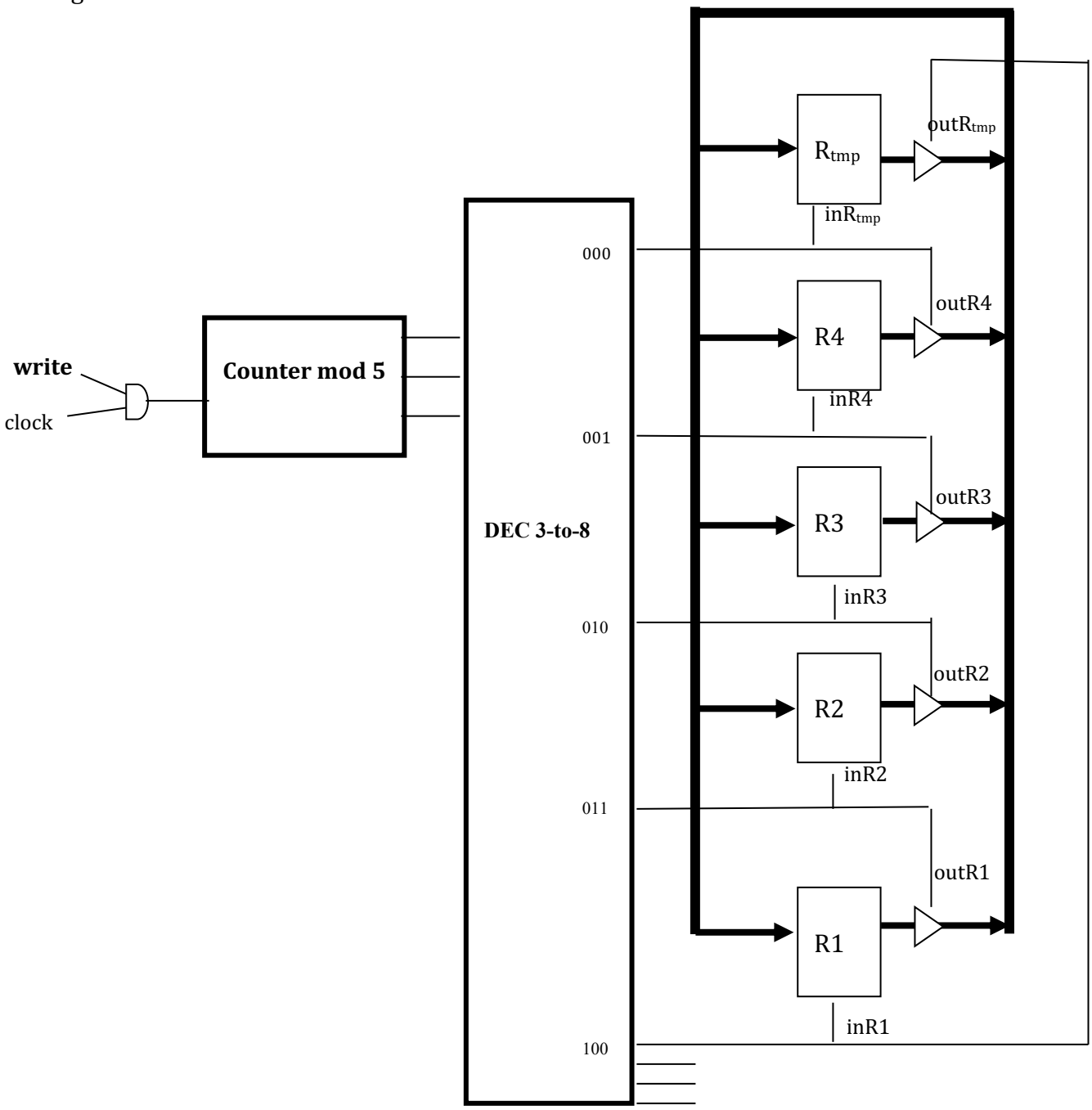
$$\text{inD0} = \overline{a1}\ \overline{a0}(l0 \oplus l1) \qquad \text{fromS0} = \overline{a1}\ \overline{a0}\ \overline{l0} + a1\ a0\ l3$$
$$\text{inD1} = \overline{a1}\ a0(l1 \oplus l2) \qquad \text{fromS1} = \overline{a1}\ a0\ \overline{l1} + \overline{a1}\ \overline{a0}\ l0$$
$$\text{inD2} = a1\ \overline{a0}(l2 \oplus l3) \qquad \text{fromS2} = a1\ \overline{a0}\ \overline{l2} + \overline{a1}\ a0\ l1$$
$$\text{inD3} = a1\ a0(l3 \oplus l0) \qquad \text{fromS3} = a1\ a0\ \overline{l3} + a1\ \overline{a0}\ l2$$

Es. 3. I registri R1, R2, R3 e R4 sono connessi ad un bus. Se il segnale **write** è uguale a 1, vengono realizzati in sequenza i seguenti trasferimenti: R1 → R2, R2 → R3, R3 → R4 e R4 → R1. Tenendo conto che utilizzando un bus non è possibile eseguire trasferimenti in parallelo, progettare quanto necessario a produrre **tutti** i segnali di controllo e la loro temporizzazione.

SOLUZIONE:

Il bus non consente trasferimenti paralleli perché in ogni istante solo una informazione può viaggiare su di esso. Per tale motivo, quando l'evento **write**=1, bisogna avviare i trasferimenti

e quindi temporizzate le operazioni eseguendo logicamente R1 --> R2, R2 --> R3, R3 --> R4 e R4 --> R1, in qualche ordine. Nel fare ciò, dobbiamo assicurarci che il contenuto precedente dei registry non sia perso (per esempio, se dapprima spostiamo R1 in R2 e poi R2 in R3, dobbiamo in qualche modo salvare il valore precedente di R2 per poi spostarlo in R3). Perciò, usiamo un registro temporaneo Rtmp ed eseguiamo la seguente sequenza di trasferimenti: R4 → Rtmp (cosicchè il contenuto di R4 è salvato), R3 → R4, R2 → R3, R1 → R2 e infine Rtmp → R1. Per ottenere tale temporizzazione, il segnale scrittura fa partire un contatore mod 5, i cui valori vengono decodificati tramite (le prime 5 uscite di) un decodificatore 3-a-8. Le uscite del decodificatore, opportunamente usate, forniscono i valori di abilitazione alla scrittura inRi e i segnali di controllo dei buffer tri-state outRi che permettono il passaggio sul bus del valore sul registro Ri.



Es. 4. Si hanno a disposizione due registri sorgente S0 e S1 da 16 bit che contengono reali memorizzati in IEEE half-precision (il primo bit rappresenta il segno, i successivi 5 bit rappresentano l'esponente biased e gli ultimi 5 bit la mantissa). Si dispone anche di due registri destinazione da 10 bit D0 e D1. Si vuole realizzare la seguente interconnessione:

- se la parte intera del numero reale memorizzato in S0 è pari, allora trasferisci la mantissa del numero contenuto in Si nel registro Di
- altrimenti trasferisci la mantissa di Si in $D_{(i+1) \bmod 2}$

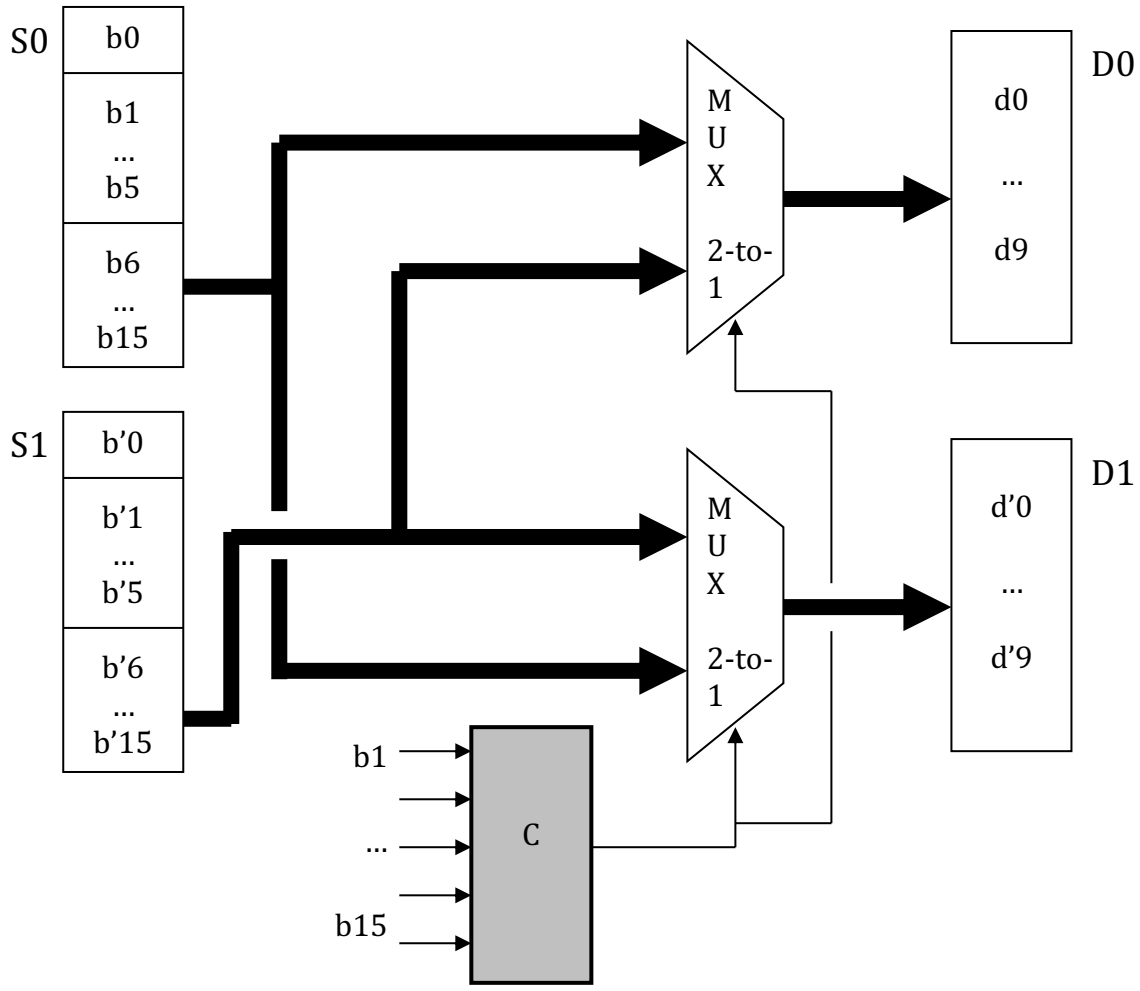
(**NotaBene :** non è richiesto che la mantissa del numero in S0 sia pari, ma che lo sia la parte intera del numero corrispondente. Pertanto è necessario riflettere su come sono collegati la rappresentazione in virgola mobile normalizzata di un numero reale e il numero reale effettivamente codificato in questo modo).

SOLUZIONE:

Diciamo che i registri sorgente siano del tipo

b0	b1 ... b5	b6 ... b15

dove b0 è il bit di segno, b1/…/b5 sono i bit dell'esponente biased e b6/../b15 i bit della mantissa. L'intereconnessione molti-a-molti richiesta è:

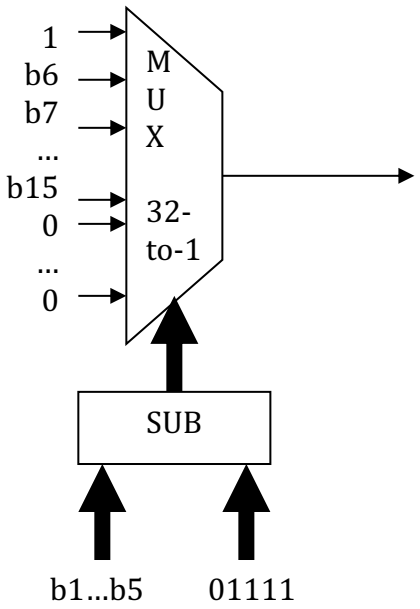


dove la linea spessa contiene solo i 10 bit delle mantisse. Inoltre i multiplexer selezionano l'input più in alto se il segnale di controllo vale 0, l'input più in basso altrimenti.

Ci resta da definire il circuito di controllo C. Per progettarlo, ricordiamo che un numero è pari se e solo se il suo LBS è 0. Pertanto, il modo più semplice di progettare C è quello di considerare b1…b5 – 01111 (come un numero in complemento a due di 5 bit); se è

- 00000, allora il numero memorizzato è 1,b6…b15, il cui LSB della parte intera è 1;
- 00001, allora il numero memorizzato è 1b6,b7…b15, il cui LSB della parte intera è b6;
- 00010, allora il numero memorizzato è 1b6b7,b8…b15, il cui LSB della parte intera è b7;
- …
- 01001, allora il numero memorizzato è 1b6…b15, il cui LSB della parte intera è b15;
- 01010, allora il numero memorizzato è 1b6…b150 e quindi il suo LSB è 0;
- …
- 01111, allora il numero memorizzato è 1b6…b15000000 e quindi il suo LSB è 0;
- 10000, allora il numero memorizzato è 0,0…01b6…b15 e quindi la sua parte intera è 0;
- …
- 11110, allora il numero memorizzato è 0,01b6…b15 e quindi la sua parte intera è 0;
- 11111, allora il numero memorizzato è 0,1b6…b15 e quindi la sua parte intera è 0.

Quindi, possiamo usare il risultato di tale sottrazione come le 5 linee di controllo di un MUX 32-to-1 che seleziona il LSB della parte intera del numero memorizzato in S0:



Qui le line spesse rappresentano numeri da 5 bit. Tale circuito dà in output 0 se e solo se la parte intera del numero memorizzato in S0 è pari, come richiesto.

Esercizi da svolgere

Es. 1. Dati 3 registri sorgente R0, R1 e R2 e un registro di uscita R' realizzare la rete di interconnessione regolata dai segnali di controllo inR' e op in grado di eseguire i seguenti trasferimenti:
se inR'=0 R' resta inalterato
se inR=1 e op=0 R0 viene copiato in R'
se inR=1 e op=1 viene posto in R' l'OR esclusivo tra R1 e R2

Es. 2. Si supponga di avere 4 registri sorgente $S_1, \ldots, S_4$ da due bit e 6 registri destinazione $D_1, \ldots, D_6$ da tre bit. Il bit più significativo (il terzo) di D_i (che indicheremo con d_i^3) è un indicatore di rilevanza dell'informazione memorizzata nel registro; in particolare, $d_i^3 = 0$ indica che l'informazione non è rilevante e che quindi può essere sovrascritta, mentre $d_i^3 = 1$ indica che l'informazione non può essere cancellata.
Si progetti in dettaglio il circuito che permetta i seguenti trasferimenti:

a) $S_1 \rightarrow D_3, \underline{D_4}$
b) $S_2 \rightarrow D_1, D_2$
c) $S_3 \rightarrow D_5$
d) $S_4 \rightarrow D_6$

dove con $S_i \rightarrow \underline{D_k}$ si intende che tale trasferimento deve dar luogo ad una memorizzazione rilevante in D_k (cioè a seguito del trasferimento si deve avere $d_k^3 = 1$), sempre che d_k^3 non fosse già a 1 (in tal caso il trasferimento $S_i \rightarrow \underline{D_k}$ non deve aver luogo). Inoltre, se il trasferimento $S_i \rightarrow D_k$ (rilevante o meno) avrà luogo, a seguito di esso si avrà $d_k^1 = s_i^1$ e $d_k^2 = s_i^2$ (cioè nei due bit meno significativi di D_k verrà memorizzata l'informazione presente in S_i).
Si assuma infine che inizialmente i registri destinazione contengano tutti i propri bit a 0.

Es. 3. Dati i registri sorgente A e B contenenti valori nella rappresentazione in complemento a 2, il registro destinazione R e due segnali di controllo, c1c0, progettare il circuito tale che:
- se c1c0=(0,0) trasferisce in R il successore di B
- se c1c0=(0,1) trasferisce in R il massimo tra A e B
- se c1c0=(1,0) trasferisce in R il risultato della somma aritmetica tra A e B
- se c1c0=(1,1) trasferisce in R il predecessore di A