

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 $i = 1, \dots, 4$), consenta i seguenti trasferimenti in parallelo:

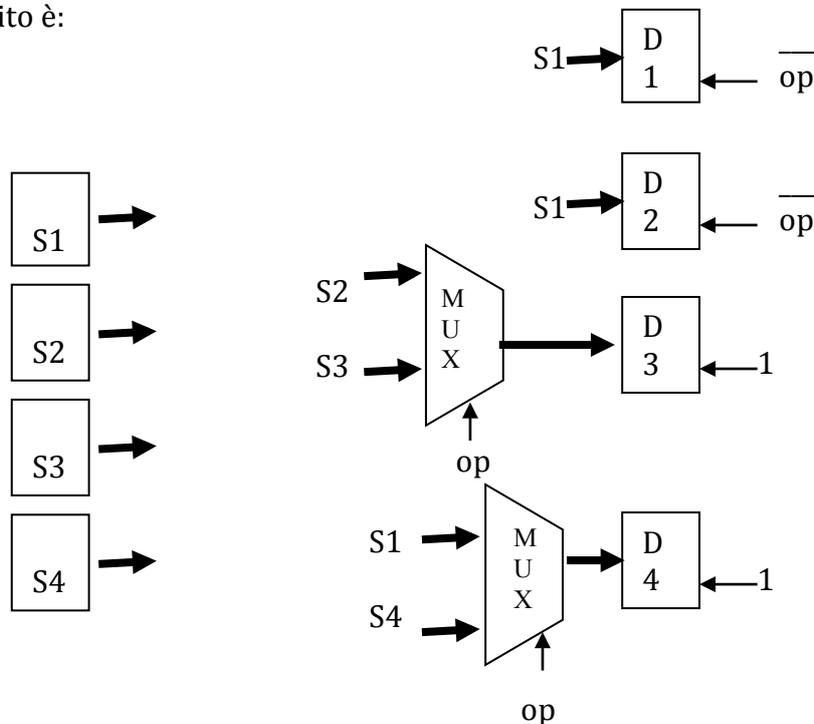
(a) $S_1 \rightarrow D_1, D_2, D_4$ e $S_2 \rightarrow D_3$

(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 D_1 e D_2 ricevono solo da S_1 , mentre D_3 e D_4 ricevono da S_2 o S_3 e da S_1 o S_4 , rispettivamente. Pertanto, si può fare una connessione punto-a-punto tra S_1 e D_1 e tra S_1 e D_2 ; invece, D_3 e D_4 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 D_1 e D_2 per la scrittura ($op = 0$ attiva i registri, $op = 1$ li disattiva; quindi in_D_1 e in_D_2 saranno la negazione di op). I registri D_3 e D_4 sono invece sempre abilitati alla scrittura (in_D_3 e in_D_4 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 S_2 per D_3 e S_1 per D_4 ; quando $op = 1$, deve selezionare S_3 per D_3 e S_4 per D_4 . Quindi, per entrambe i MUX il segnale di controllo è il bit op . Pertanto, il circuito è:



Nella rete con uniche destinazioni i registri D_1 e D_2 , il trasferimento (b) sarebbe ancora lecito mentre il trasferimento (a) no: infatti, nel primo caso non ci sono conflitti poiché registri

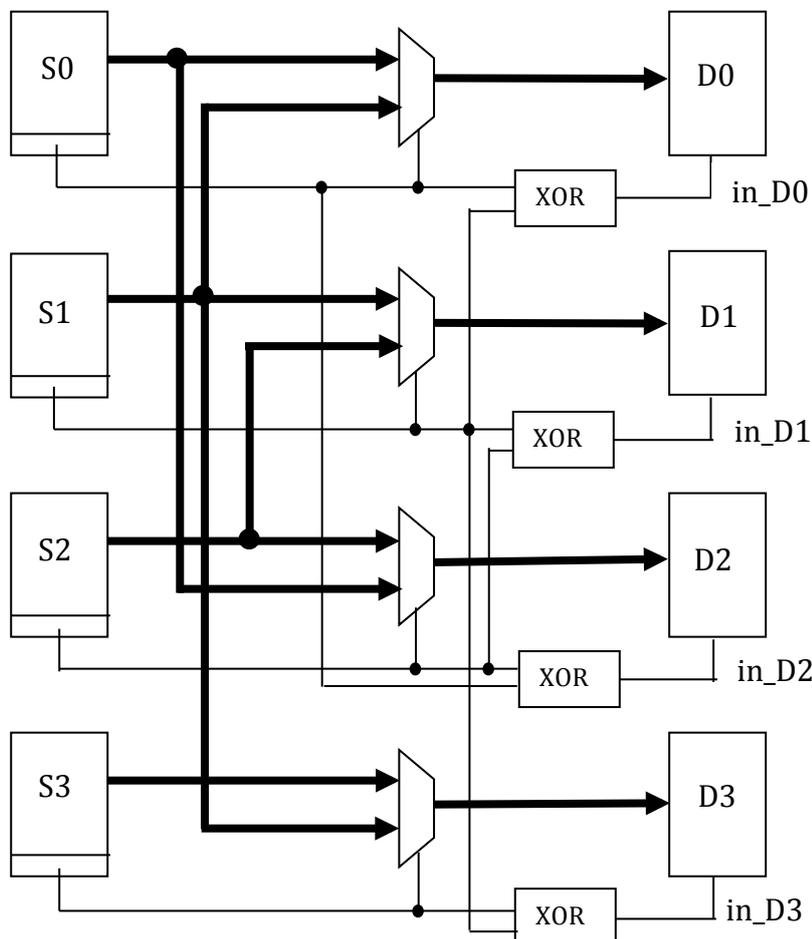
sorgente e destinazione sono disgiunti, mentre nel secondo caso c'è conflitto sul registro D1 che dovrebbe ricevere in input contemporaneamente il dato proveniente da S1 e da S2.

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 S_i è pari allora S_i viene copiato in D_i ; altrimenti $S_{(i+1) \bmod 3}$ viene copiato in D_i . Si abbia che il registro D_i è abilitato alla scrittura se e solo se $S_i + S_{(i+1) \bmod 3}$ 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 S_i e $S_{(i+1) \bmod 3}$ sulla base di parità del registro S_i , 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 S_i è pari, altrimenti è dispari. Il valore del bit meno significativo viene quindi usato come segnale di controllo per il multiplexer. Il segnale in_D_i 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 S_i e $S_{(i+1) \bmod 3}$. 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+S0** è dispari, riceve **S2** se S2 è pari, **S0** se S2 è dispari
- D3** è abilitato se **S3+S1** è dispari, riceve **S3** se S3 è pari, **S1** se S3 è dispari

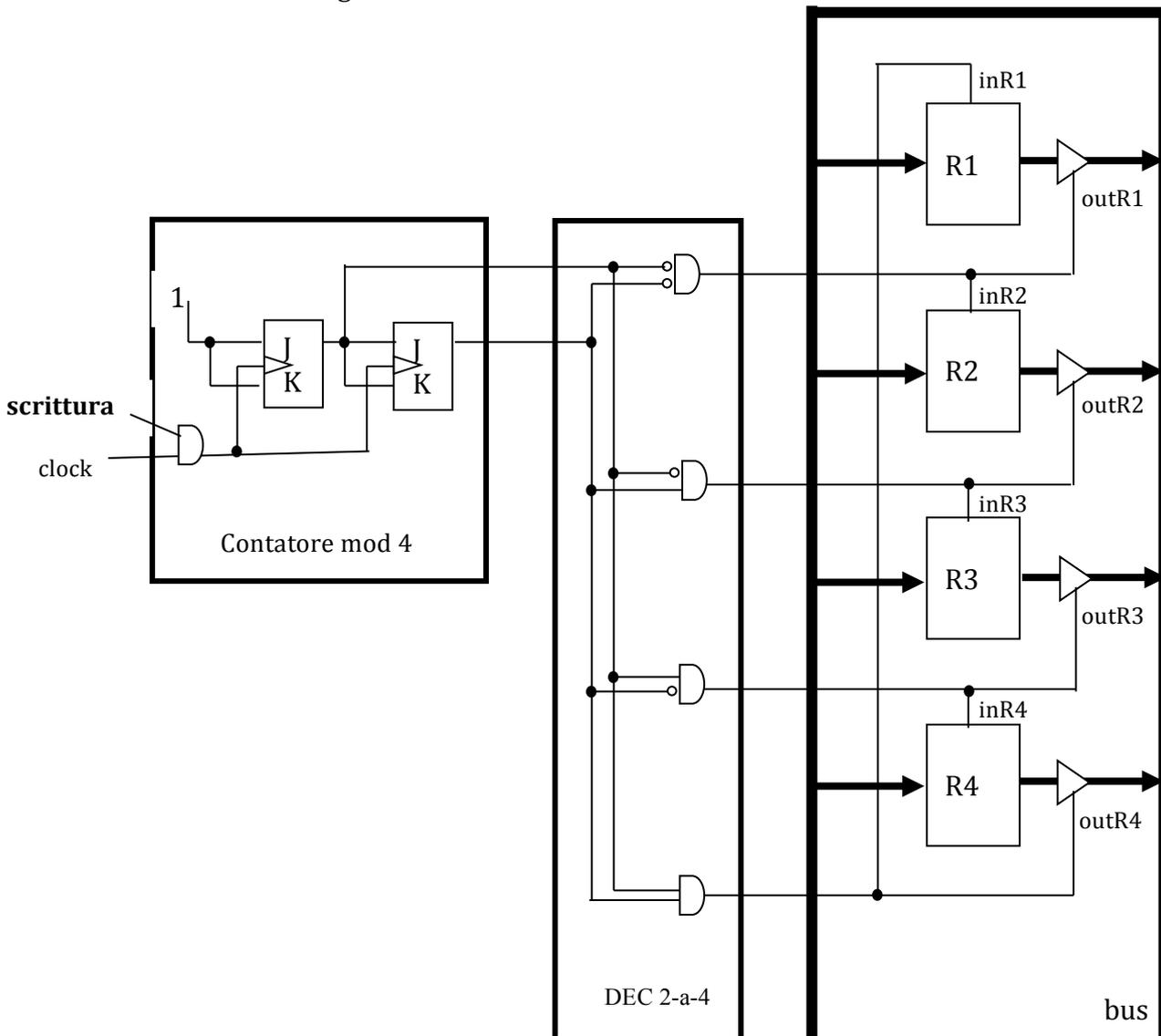


Senza trasferimenti in parallelo si può usare un bus.

Es. 3. I registri R1, R2, R3 e R4 sono connessi ad un bus. Se il segnale **scrittura** 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 fino al dettaglio di porte logiche e flip-flop (N.B. i registri non vanno dettagliati).

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 **scrittura**=1 bisogna avviare i trasferimenti e quindi temporizzare le operazioni eseguendo in sequenza R1 --> R2 poi R2 --> R3 poi R3 --> R4 e infine R4 --> R1. Per ottenere tale temporizzazione, il segnale scrittura fa partire un contatore mod 4, i cui valori vengono decodificati tramite un decodificatore 2-a-4. Le uscite del decodificatore, opportunamente usate, forniscono i valori di abilitazione alla scrittura inR_i e i segnali di controllo dei buffer tri-state outR_i che permettono il passaggio sul bus del valore sul registro R_i.



Es. 4. Si hanno a disposizione due registri sorgente S0 e S1 da 16 bit che contengono reali memorizzati in rappresentazione a virgola mobile normalizzata: il primo bit rappresenta il segno, i successivi 12 bit rappresentano la mantissa e gli ultimi 3 bit l'esponente in complemento a due. Si dispone anche di due registri destinazione da 12 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 $D_{(i+1) \text{ MOD } 2}$

(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	...	b12	b13	...	b15
----	----	-----	-----	-----	-----	-----

dove b0 è il bit di segno, b1/.../b12 sono i bit della mantissa e b13/b14/b15 i bit dell'esponente.

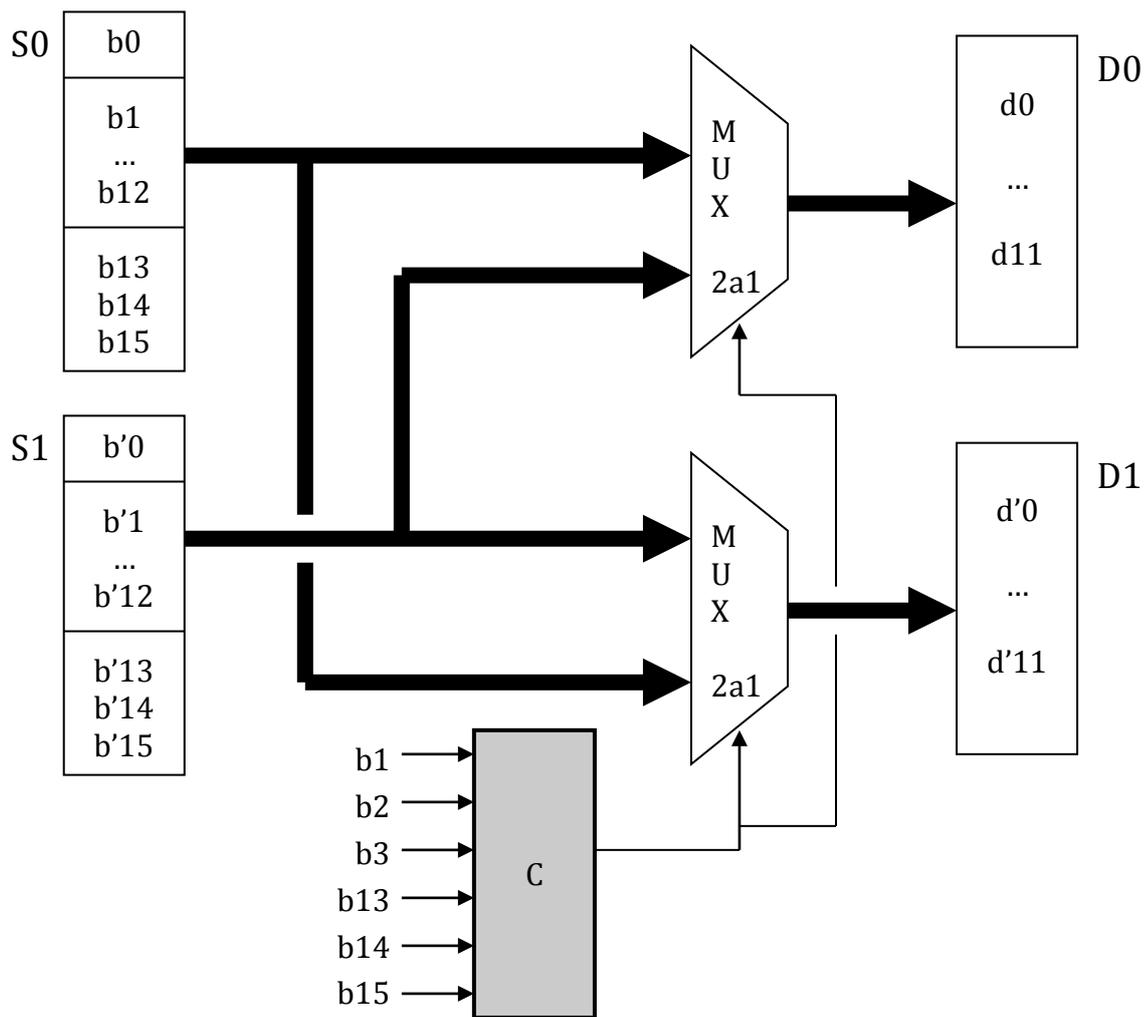
Per definire il circuito di controllo, basta osservare che

- se $b_{13} = 1$ oppure $b_{13} = b_{14} = b_{15} = 0$ allora il numero ha parte intera uguale a 0 (che quindi è pari)
- se $b_{13} b_{14} b_{15} = 001$ allora la parte intera è b1 (che quindi è pari sse b1 è 0)
- se $b_{13} b_{14} b_{15} = 010$ allora la parte intera è b1 b2 (che quindi è pari sse b2 è 0)
- se $b_{13} b_{14} b_{15} = 011$ allora la parte intera è b1 b2 b3 (che quindi è pari sse b3 è 0)

Pertanto il primo trasferimento verrà effettuato se e solo se $X = 1$, dove X è definito come

$$X = b_{13} + \underline{b_{13} b_{14} b_{15}} + \underline{b_1 b_{13} b_{14} b_{15}} + \underline{b_2 b_{13} b_{14} b_{15}} + \underline{b_3 b_{13} b_{14} b_{15}}$$

Infatti, X vale 1 se e solo se uno dei quattro casi sopra elencati si verifica. Sia C il circuito che prende in input b1/b2/b3/b13/b14/b15 e dà in output 1 sse $X = 1$. La rete di interconnessione richiesta è pertanto



dove la linea spessa contiene solo i 12 bit delle mantisse. Inoltre i multiplexer selezionano l'input più in alto se il segnale di controllo vale 1, l'input più in basso altrimenti.

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, \dots, S_4 da due bit e 6 registri destinazione D_1, \dots, 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:

$$a) S_1 \rightarrow D_3, D_4$$

$$b) S_2 \rightarrow D_1, D_2$$

$$c) S_3 \rightarrow D_5$$

$$d) S_4 \rightarrow D_6$$

dove con $S_i \rightarrow D_k$ 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 $S_i \rightarrow D_k$ 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