

Unità aritmetico-logica e registri espandibili

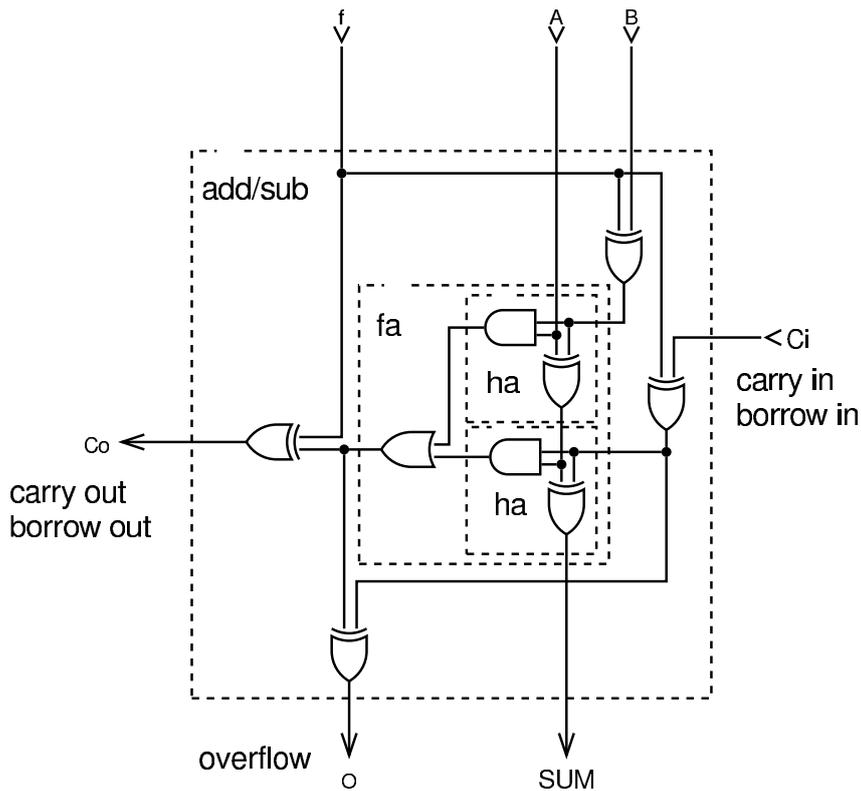
Unità aritmetico-logica a bit singolo	1779
Registri	1786

In questa sezione viene presentato uno studio alternativo per la costruzione di una ALU da un solo bit, allineabile in parallelo per ottenere ranghi superiori, assieme a dei registri, anche questi a singolo bit, ma espandibili. Per comprendere il funzionamento e l'uso dei registri, è comunque necessario leggere la descrizione relativa ai flip-flop che appare a partire dalla sezione [u101](#).

Il file sorgente di questo studio dovrebbe essere disponibile presso [allegati/circuiti-logici/modular-alu-0050.v](#).

Unità aritmetico-logica a bit singolo

Figura u100.1. Modulo **as**: per $f=0$, somma con riporto; per $f=1$ sottrazione con prestito; se il riporto (o il prestito) in ingresso è diverso da quello in uscita, si verifica lo straripamento.



$f=0$

viene calcolata la somma ($A + B$) e generato eventualmente il riporto; C_i e C_o funzionano in qualità di riporti (in ingresso e in uscita).

$f=1$

viene calcolata la sottrazione ($A - B$) e generato eventualmente la richiesta di prestito; C_i e C_o funzionano in qualità di richiesta di prestito (borrow).

Figura u100.2. Modulo **logic**: unità logica, comandata dall'ingresso f che seleziona il tipo di operazione logica. Nella didascalia interna dei valori assegnabili all'ingresso f , il simbolo «+» va inteso come operatore OR.

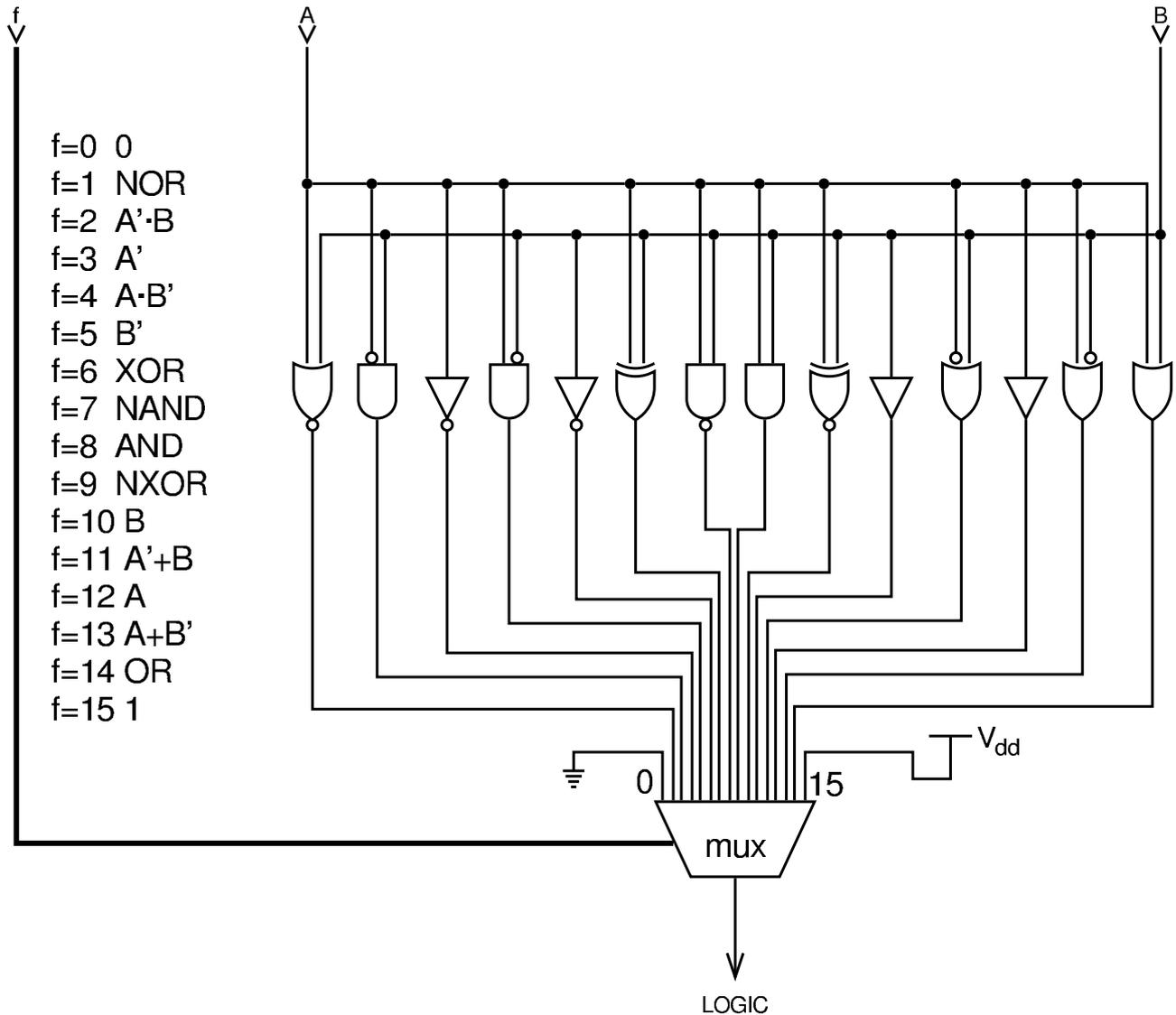


Figura u100.3. Modulo **ltgt**: confronto. Data la necessità di realizzare una ALU espandibile, il confronto avviene a livello di bit, usando il riporto in ingresso in caso di uguaglianza. L'esito viene trasmesso attraverso il riporto in uscita. Il modulo è composto da due parti; nella figura, nell'ordine, appaiono i moduli: **lt** ($A < B$), **gt** ($A > B$), **ltgt**. Il modulo complessivo **ltgt** esegue il primo o il secondo confronto, in base al valore contenuto nell'ingresso f . Il controllo di uguaglianza non viene fatto, perché per ottenerlo è sufficiente sfruttare l'unità logica con l'operatore NXOR.

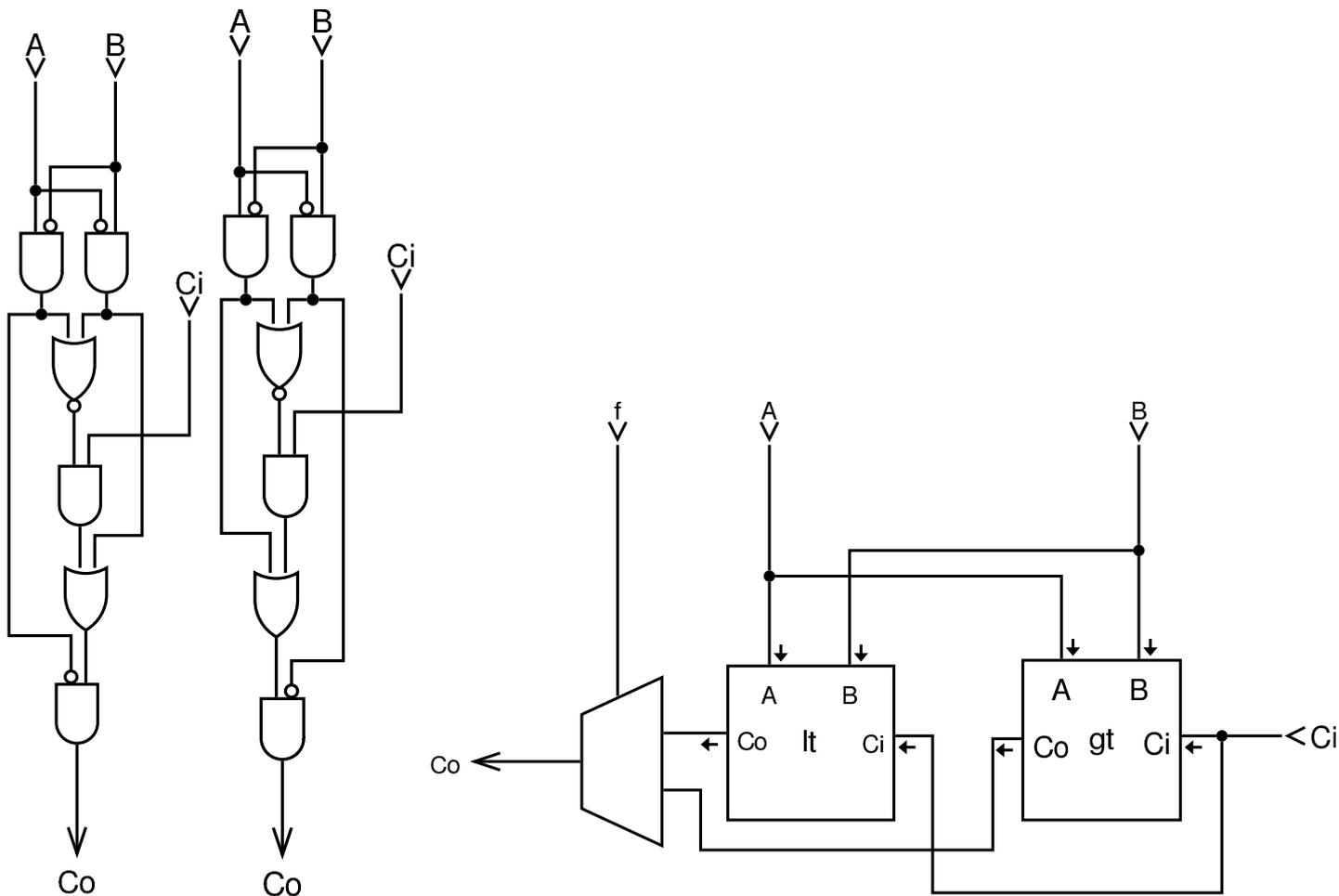


Figura u100.4. Moduli **shl** e **shr**: scorrimento a sinistra e a destra. Lo scorrimento avviene necessariamente utilizzando il riporto; per la precisione, per lo scorrimento a sinistra si usa il riporto «normale», mentre per lo scorrimento a destra si usa un riporto apposito, da sinistra a destra, indicato con le sigle **Di**, **Do**. Il riporto a destra non tiene conto del segno, perché in questa fase si considera un solo bit. L'ingresso **f** serve, in entrambi i casi, a selezionare l'operando sul quale intervenire: **A** o **B**.

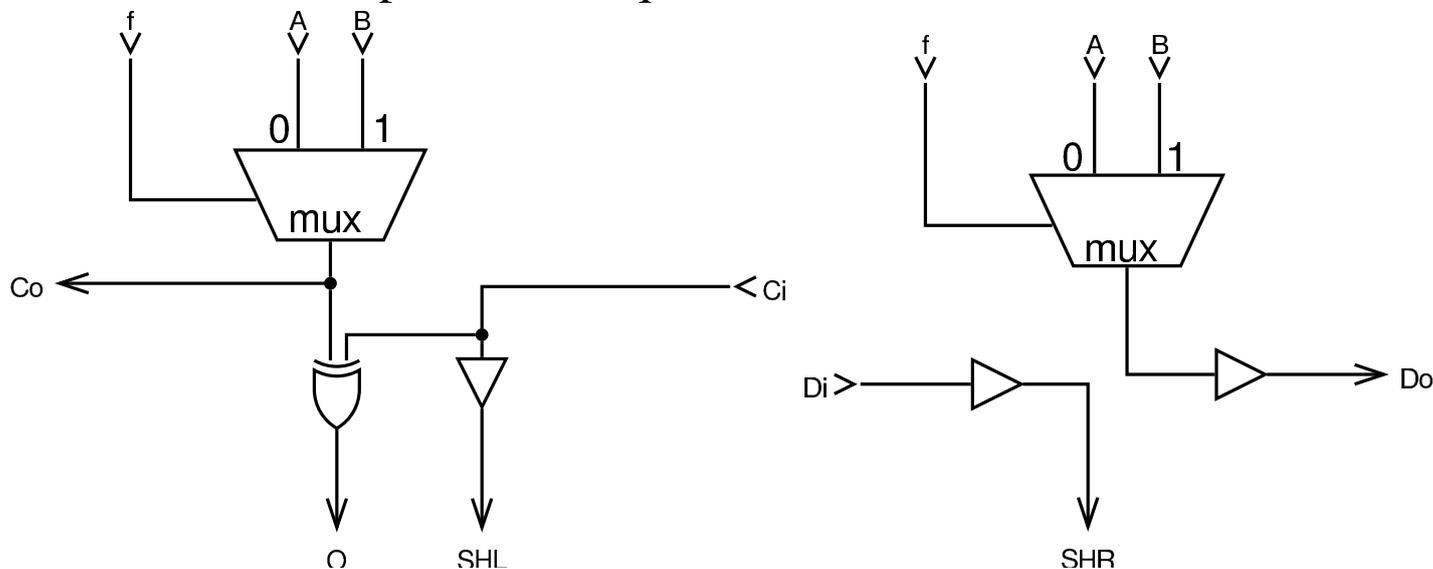


Figura u100.5. Modulo **z**: Zero. Determina se il valore è pari a zero e se lo sono anche i moduli precedenti (attraverso il riporto). Se fino a questa cifra i valori sono a zero, attiva il riporto in uscita. L'ingresso f consente di selezionare quale operando verificare.

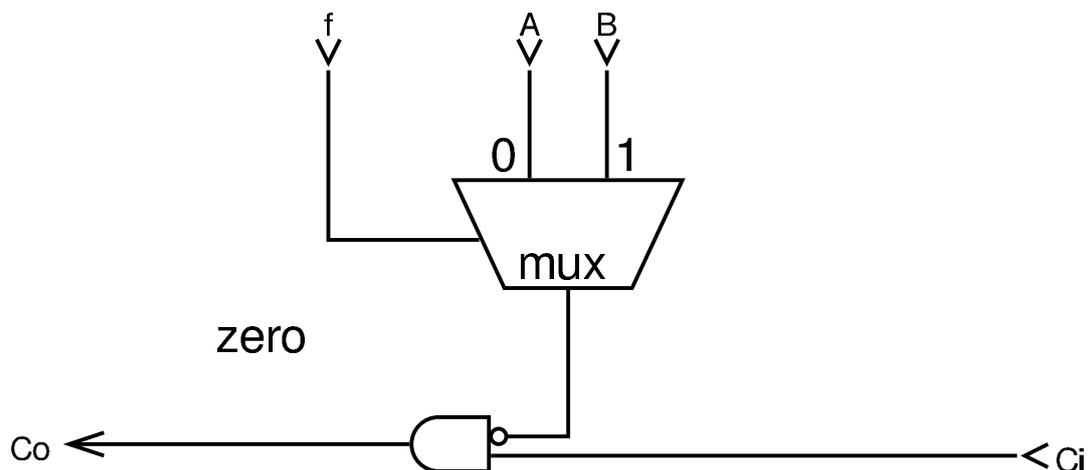


Figura u100.6. Modulo **alu**: ALU completa dei vari componenti aritmetici, logici e di scorrimento. Quando l'ingresso f ha il bit più significativo pari a zero, contiene la funzione dell'unità logica, altrimenti contiene le funzioni per gli altri moduli. Gli altri moduli, quando contengono a loro volta l'ingresso f , questo lo traggono da f_3 di quello principale, ovvero dal quarto bit.

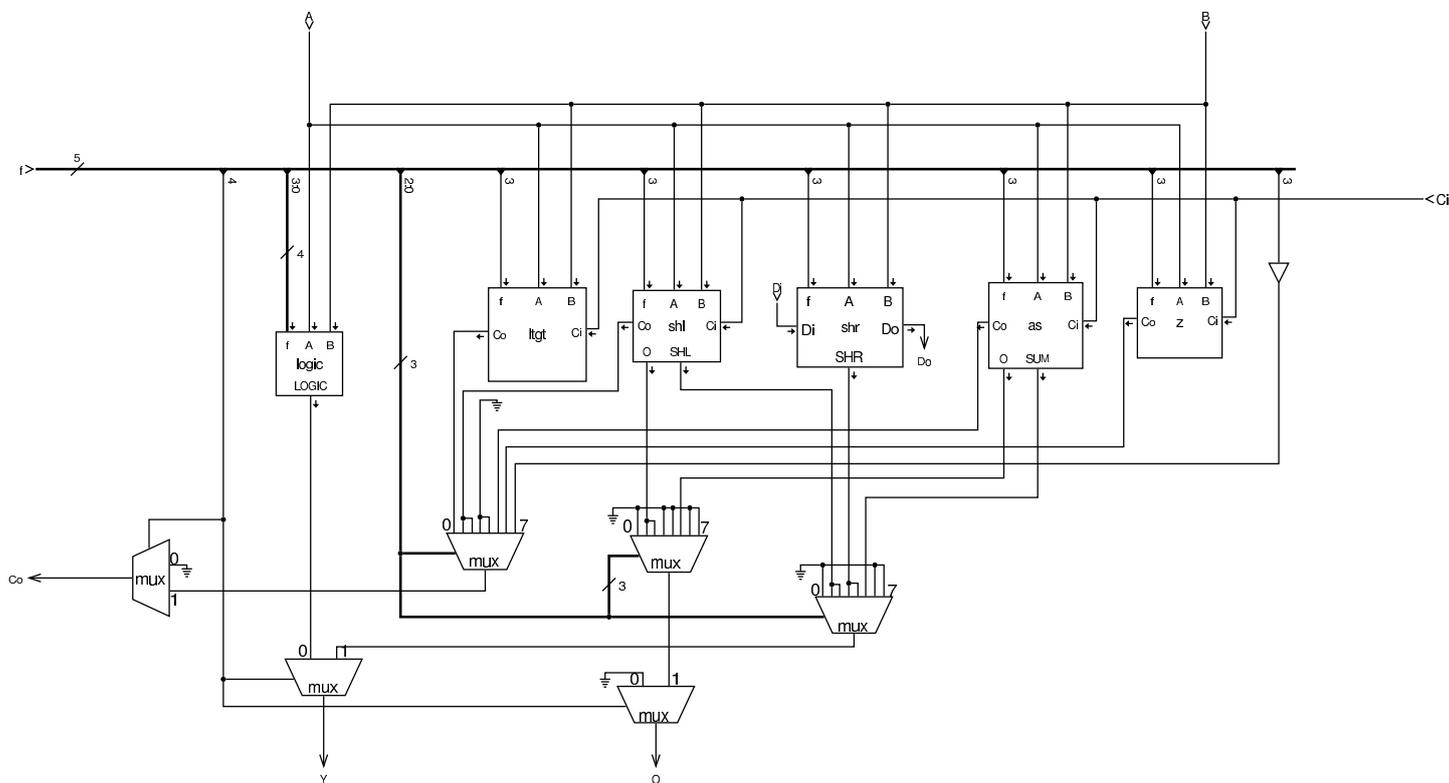
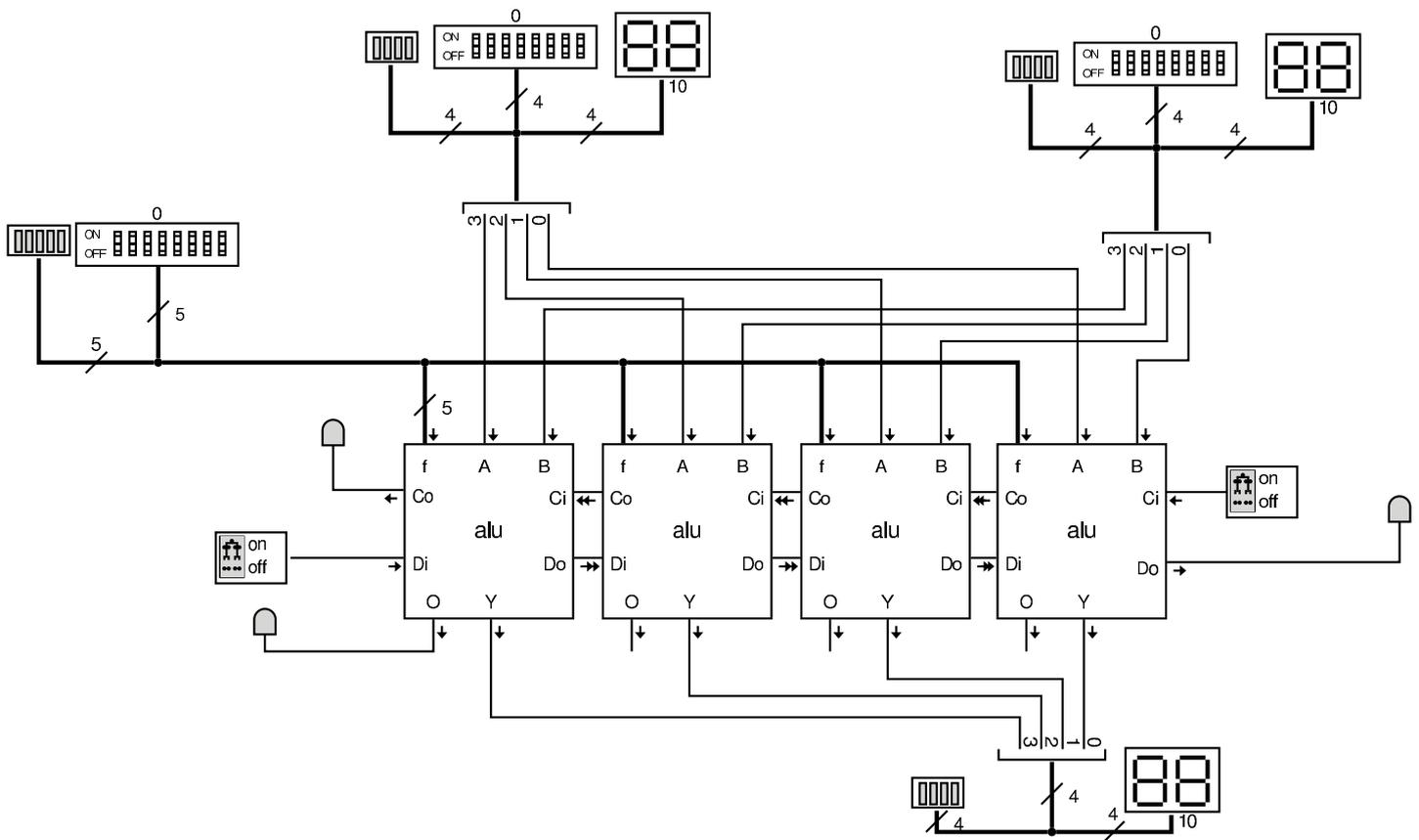


Figura u100.7. Esempio di utilizzo in parallelo del modulo **alu**, per ottenere un'elaborazione a quattro bit. Vanno osservate in particolare le connessioni delle linee di riporto **C** e **D**.



Registri

« Attorno alla ALU vengono collocati dei registri, ovvero delle unità di memoria in grado di memorizzare, ognuna, un solo valore. Per i registri si utilizzano flip-flop D, semplici, come nella prima delle figure. I flip-flop sono descritti a partire dalla sezione u101; qui, per comprendere il meccanismo, basti intendere che ricevono l'informazione da memorizzare attraverso l'ingresso **D**, ma solo quando l'ingresso **E** (*enable*) è attivo, quindi mantengono l'informazione disponibile nell'uscita **Q**.

Figura u100.8. Modulo **d**: Flip-flop D semplice, usato per la realizzazione dei registri.

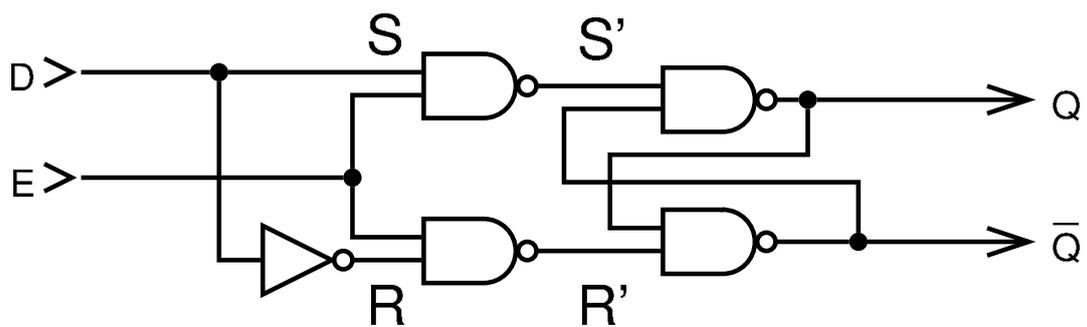


Figura u100.9. Modulo **ralu**: ALU e registri, a un solo bit. Gli ingressi **SRC** e **DST**, determinano, rispettivamente, l'origine e la destinazione del movimento di un'informazione, attraverso il bus. In pratica, **SRC** seleziona cosa mettere nel bus, e **DST** seleziona quale registro deve salvare questa informazione. L'ingresso **ALU** contiene la funzione da far svolgere alla ALU, nel caso sia selezionata l'origine corrispondente al terzo registro, che in pratica è l'uscita della ALU stessa. L'ingresso **E** va attivato quando tutti gli altri ingressi sono stati configurati correttamente, per il tempo necessario al registro di destinazione per recepire l'informazione contenuta nel bus. Va osservato che la ALU esegue le sue operazioni sulla base del contenuto dei primi due registri, e lo fa continuamente; inoltre, il terzo registro non esiste, in quanto è il riferimento usato per individuare l'uscita della ALU, per cui non è possibile scrivere direttamente in questo terzo registro, senza intervenire con una funzione della ALU. Gli ingressi **SRC** e **DST** hanno un bit in più, necessario per controllare altri otto registri esterni, necessari per l'accesso alla memoria.

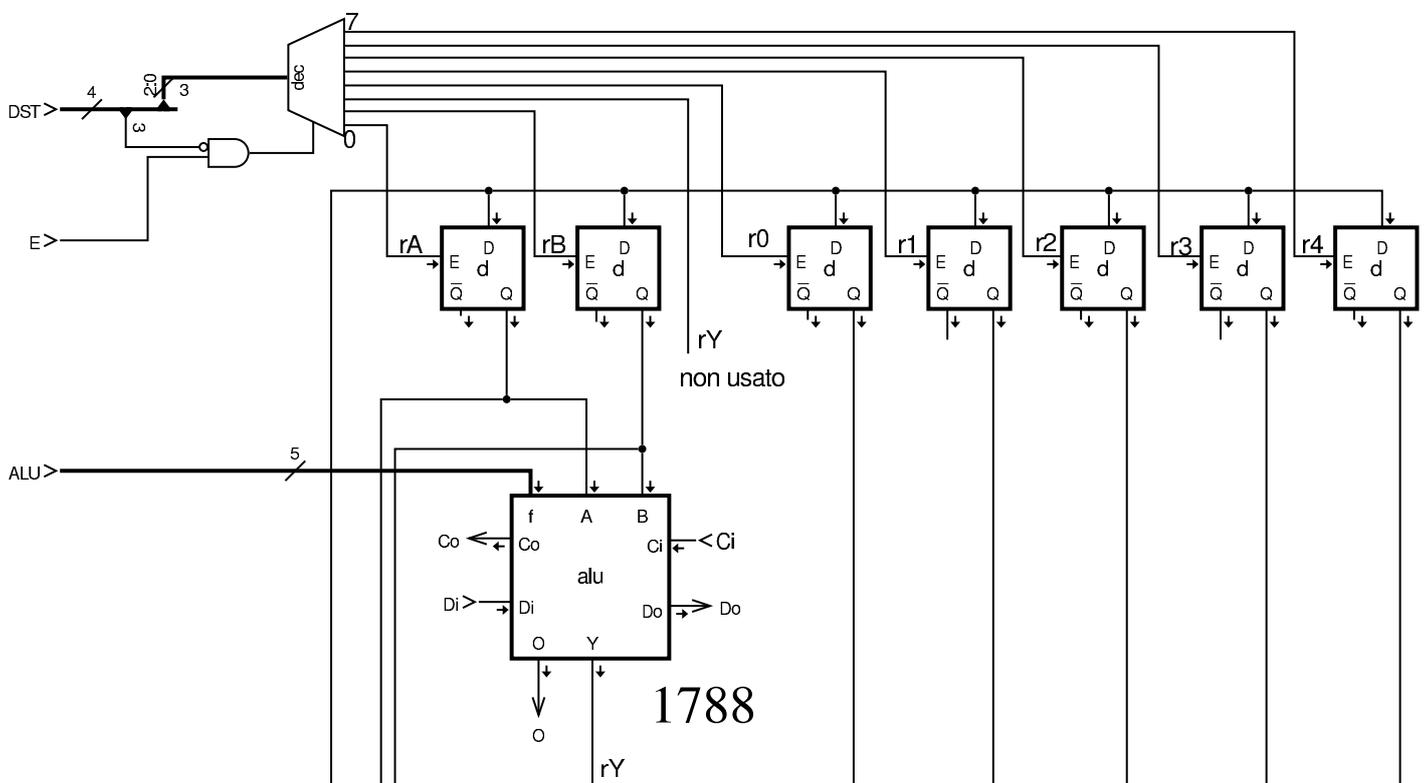


Figura u100.10. Modulo **ralu4**: collocazione in serie di quattro moduli a un solo bit. Lo stesso procedimento si seguirebbe per ranghi superiori.

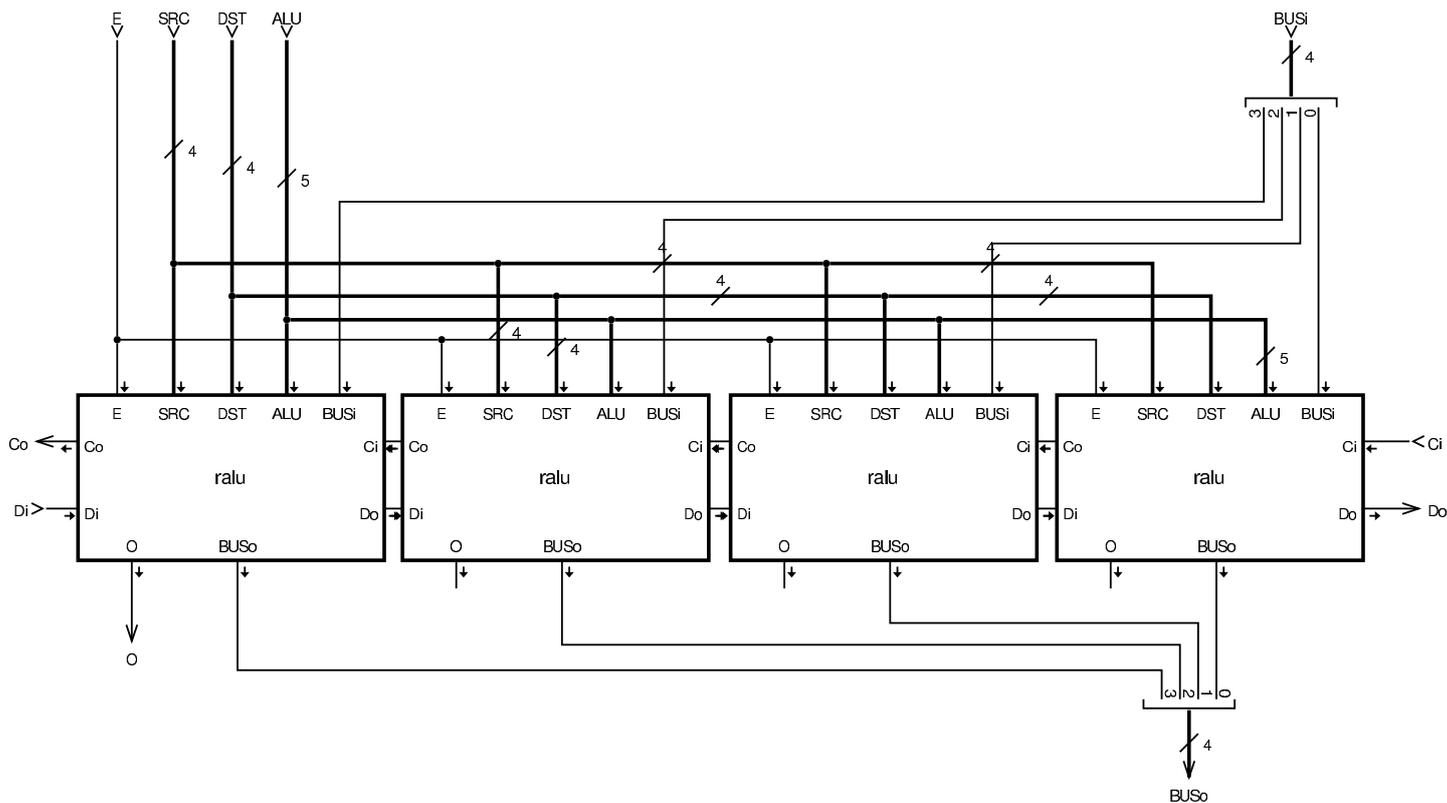


Figura u100.11. Modulo **ralu32**: aggregazione di più moduli più piccoli, per un rango più ampio.

