

3.9 Unità aritmetiche e logiche

Al Paragrafo 2.3 si è accennato all'aritmetica binaria. Si può fare un'osservazione, apparentemente ovvia ma di fondamentali conseguenze: le tabelline delle operazioni aritmetiche contengono i due simboli 0 e 1, esattamente come due sono i simboli

nell'algebra delle reti. Ne deriva che le tabelline aritmetiche possono essere interpretate come le tabelle di verità delle funzioni logiche corrispondenti alle operazioni aritmetiche.

3.9.1 Semisommatore

Si consideri la somma di due bit. La tabella riportata a sinistra di Figura 3.46 rappresenta, a un tempo, sia la tabellina aritmetica della somma S_i e del relativo riporto R_i , sia la tabella di verità delle due funzioni logiche $S_i(A_i, B_i)$ e $R_i(A_i, B_i)$. Conseguentemente, se si costruisce la rete che ha come uscite S_i e R_i , si costruisce la rete che effettua la somma aritmetica di due bit e ne calcola il riporto. La rete in questione prende il nome di *Semisommatore* e viene indicata con HA (da *Half Adder*).

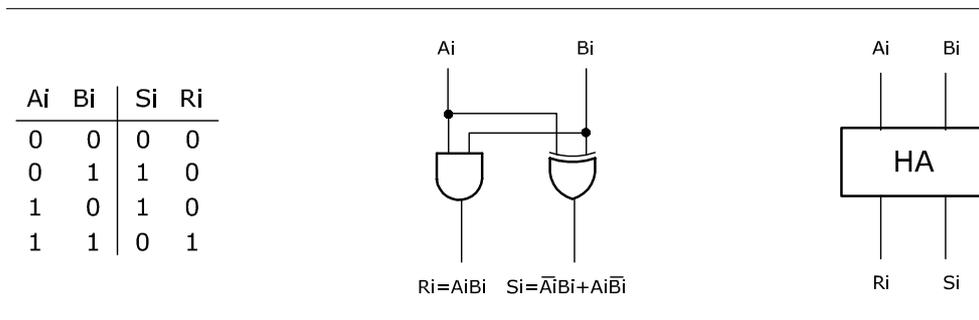


Figura 3.46 A sinistra viene riportata la tabellina aritmetica della somma e del riporto di due bit. La tabella in questione può essere interpretata come la tabella di verità delle due funzioni logiche S_i e R_i . Al centro viene data la rete corrispondente. A destra viene data una schematizzazione della rete come blocco funzionale. Il blocco è stato indicato come HA (da *Half Adder*), per *Semisommatore*.

3.9.2 Somma di due numeri interi

Vogliamo ora costruire la rete che effettua la somma di due numeri interi, rappresentati in forma binaria attraverso le due parole di n bit $A = [A_{n-1} \dots A_0]$ e $B = [B_{n-1} \dots B_0]$. Si indichi con $S = [S_{n-1} \dots S_0]$ il risultato della somma.

In Figura 3.47 viene riportato un sommatore di due numeri di n bit costruito modellando il procedimento di somma con “carta e matita”, come illustrato al Paragrafo 2.3. La rete somma a partire dai bit meno significativi procedendo verso sinistra e tenendo conto del riporto. La cella elementare del sommatore di Figura 3.47 viene detta *Sommatore completo* e indicata con FA (da *Full Adder*). A differenza del semisommatore, questa cella somma due bit e un eventuale riporto. La parola $[S_{n-1} \dots S_1 S_0]$ costituisce la somma di $[A_{n-1} \dots A_1 A_0]$ con $[B_{n-1} \dots B_1 B_0]$. Si noti che è necessario porre $R_{-1} = 0$.

Il tempo impiegato dalla rete di Figura 3.47 per calcolare la somma dipende dalla lunghezza della parola e dalla particolare coppia di numeri sommati. Infatti la

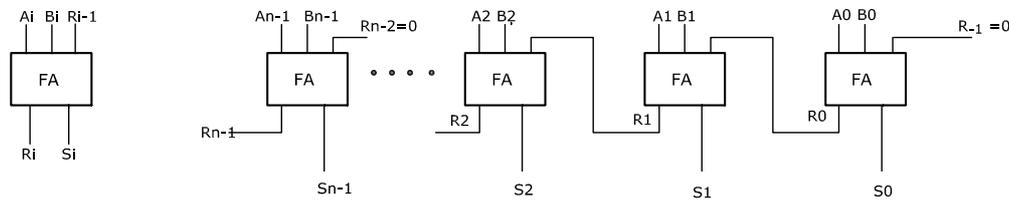


Figura 3.47 Schema di un sommatore di parole di n bit, detto sommatore di *ripple*, costruito impiegando la cella detta *Sommatore completo* (FA).

generica cella produce una uscita stabile solo dopo che è diventato stabile il riporto in ingresso. Nel peggiore dei casi il riporto può propagarsi dal bit meno significativo a quello più significativo.

Si indichino con τ_R e con τ_S i tempi di commutazione che la cella FA richiede per calcolare rispettivamente il riporto e la somma. Nel caso peggiore il riporto deve propagarsi attraverso tutte le celle. Si ha dunque un tempo di commutazione di caso peggiore pari a $\Delta_R = n\tau_R$, per il riporto R_{n-1} , e pari a $\Delta_S = (n - 1)\tau_R + \tau_S$, per il calcolo della somma.

3.9.3 Sommatore completo

Per la realizzazione del sommatore completo si può seguire il seguente ragionamento.¹¹

- La somma di tre bit A_i, B_i, R_{i-1} dà risultato 1 solo se è dispari il numero di bit a 1, cioè se è 1 l'OR esclusivo dei tre bit. In altri termini:

$$S_i = A_i \oplus B_i \oplus R_{i-1} = (A_i \oplus B_i) \oplus R_{i-1}$$

- Il riporto vale 1 quando: (a) la somma di A_i e B_i dà direttamente riporto; oppure (b) quando vale 1 la somma di A_i e B_i , con riporto in ingresso pure a 1. Questa descrizione a parole viene riformulata come:

$$R_i = A_i B_i + (A_i \oplus B_i) R_{i-1}$$

Dalle precedenti relazioni, tenuto conto che le due uscite del semisommatore rappresentano l'AND e lo XOR dei due ingressi, si ricava facilmente che il sommatore completo può essere costruito impiegando due semisommatori come in Figura 3.48. Vogliamo ora calcolare il tempo di commutazione di caso peggiore per il sommatore di Figura 3.47. Si indichi con τ il tempo di commutazione di una generica porta

¹¹Ovviamente, la rete che realizza il sommatore completo può essere anche ottenuta partendo dalla tabella di verità di S_i e R_i di Figura 3.48. Si veda il Paragrafo 3.9.5.

e si assuma che tutte le porte commutino in un tempo τ . Il tempo richiesto a un semisommatore per calcolare somma o riporto risulta pari a τ . Conseguentemente, per il sommatore completo si ha: $\tau_R = 3\tau$ e $\tau_S = 2\tau$. Dunque, per un sommatore di n bit:

$$\Delta_S = 3\tau(n - 1) + 2\tau = (3n - 3 + 2)\tau = (3n - 1)\tau$$

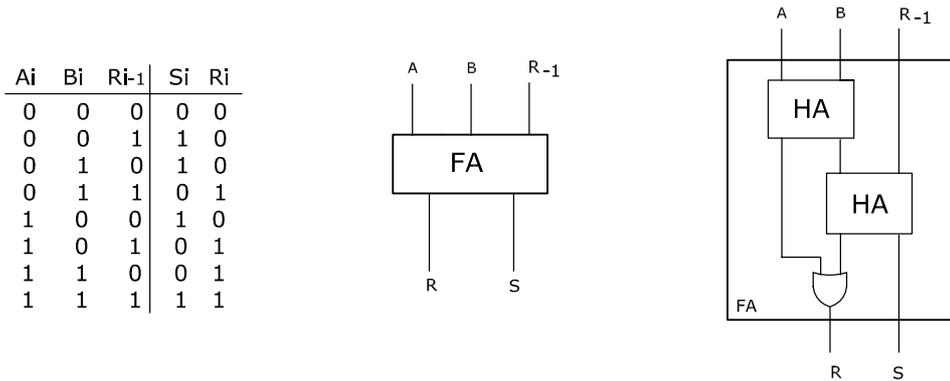


Figura 3.48 A sinistra la tabella di verità per il sommatore completo. Al centro la schematizzazione come blocco funzionale. A destra il sommatore completo costruito con due semisommatori.

3.9.4 Somma con calcolo anticipato del riporto

È possibile ridurre i tempi di calcolo della somma con la tecnica del calcolo anticipato del riporto. La tecnica si basa sulla struttura algebrica di S_i e R_i .

Facendo riferimento alla Figura 3.49 e posti $p_i = A_i \oplus B_i$ e $g_i = A_i B_i$, si ha:

$$S_i = A_i \oplus B_i \oplus R_{i-1} = (A_i \oplus B_i) \oplus R_{i-1} = p_i \oplus R_{i-1}$$

$$R_i = A_i B_i + (A_i \oplus B_i)R_{i-1} = g_i + p_i R_{i-1}$$

Dunque:

$$R_0 = g_0 + p_0 R_{-1}$$

$$R_1 = g_1 + p_1 R_0 = g_1 + p_1 g_0 + p_1 p_0 R_{-1}$$

$$R_2 = g_2 + p_2 R_1 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 R_{-1}$$

$$R_3 = g_3 + p_3 R_2 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 R_{-1}$$

Nelle precedenti espressioni i termini p_i e g_i , detti rispettivamente *funzione propagata* e *funzione generata*, sono calcolati in un tempo τ pari alla commutazione di una sola porta. Gli R_i sono dunque calcolati in un tempo pari alla commutazione di tre porte. Di conseguenza, il calcolo di S richiede un tempo $\Delta_S = 4\tau$. La rete che effettua

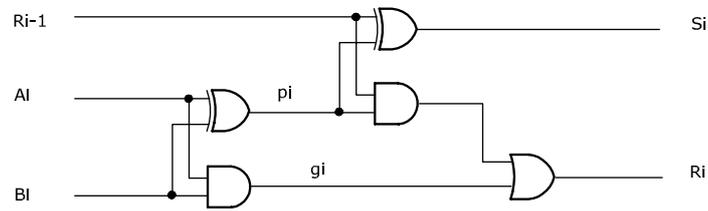


Figura 3.49 Funzione generata (g_i) e funzione propagata (p_i) per il sommatore completo.

il calcolo dei riporti viene detta *look-ahead carry generator*. In Figura 3.50 viene mostrato lo schema di un sommatore di parole di 4 bit con calcolo anticipato del riporto.

Considerando l'espressione di R_3 e posto: $G = g_3 + p_3g_2 + p_3p_2g_1 + p_3p_2p_1g_0$ e $P = p_3p_2p_1p_0$, R_3 si riscrive come:

$$R_3 = G + PR_{-1}$$

È quindi possibile impiegare in modo iterativo il calcolo anticipato del riporto, costruendo reti a più livelli. Per esempio, si supponga di volere sommare parole di 16 bit. Il sommatore può essere costruito impiegando quattro sommatori da 4 bit, come quello di Figura 3.50, e un ulteriore *look-ahead carry generator*, come in Figura 3.51.

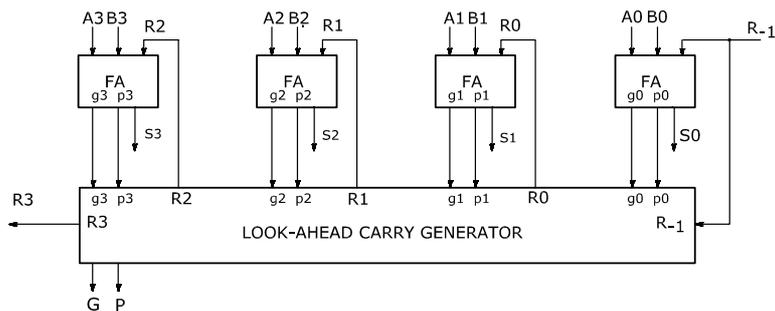


Figura 3.50 Somma di parole di 4 bit con calcolo anticipato del riporto.

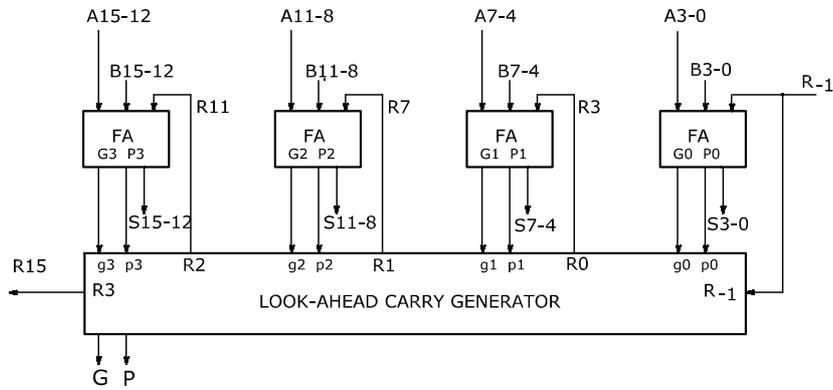


Figura 3.51 Somma di parole di 16 bit. Il sommatore è costruito utilizzando 4 sommatore di 4 bit con calcolo anticipato del riporto. Ciascuno dei sommatore di 4 bit impiega al suo interno esattamente la stessa rete per il calcolo anticipato del riporto.

3.9.5 Una semplice unità aritmetica e logica

In Figura 3.52 vengono riportate le mappe di Karnaugh corrispondenti alla tabella di verità di Figura 3.48. Da esse si ricava:

$$S_i = \overline{A_i} \overline{B_i} R_{i-1} + \overline{A_i} B_i \overline{R_{i-1}} + A_i \overline{B_i} \overline{R_{i-1}} + A_i B_i R_{i-1}$$

$$R_i = A_i R_{i-1} + A_i B_i + B_i R_{i-1}$$

Le precedenti relazioni permettono di tracciare lo schema della rete corrispondente al sommatore completo. Per motivi che saranno chiari più avanti, conviene esprimere S_i in funzione di $\overline{R_i}$. A tale scopo si faccia riferimento alla Figura 3.53. In Figura 3.53a viene riportata la mappa di $\overline{R_i}$; con l'artificio di Figura 3.53b e di Figura 3.53c la mappa di $\overline{R_i}$ viene ricondotta a quella di S_i . Alla mappa di Figura 3.53c corrisponde l'espressione (3.4). La rete corrispondente è riportata in Figura 3.54.

$$S_i = A_i \overline{R_i} + B_i \overline{R_i} + \overline{R_i} R_{i-1} + A_i B_i R_{i-1} \quad (3.4)$$

Si noti che $\overline{R_i}$ è l'uscita di una rete a 2 livelli, mentre per S_i la rete è a 4 livelli.

3.9.6 Esempio di costruzione di un'unità aritmetica

Partiamo dal sommatore appena visto e mostriamo come sia possibile aggiungere della logica in modo da fargli effettuare altre operazioni.

Sappiamo che per la sottrazione basta rappresentare in forma negativa il secondo termine e sommarlo al primo (Paragrafo 2.4):

$$A - B = A + (-B)$$

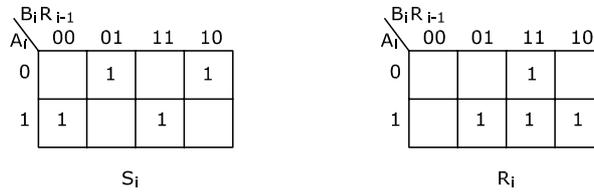


Figura 3.52 Mappe Karnaugh di S_i e R_i .

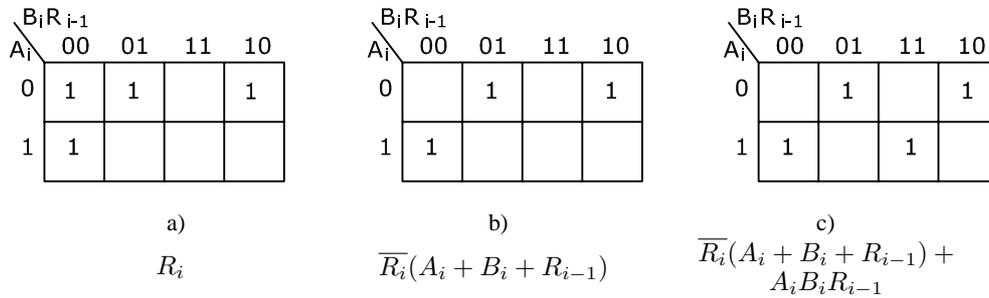


Figura 3.53 Metodo pratico per pervenire a una espressione di S_i in funzione di A_i , B_i , R_i e R_{i-1} .

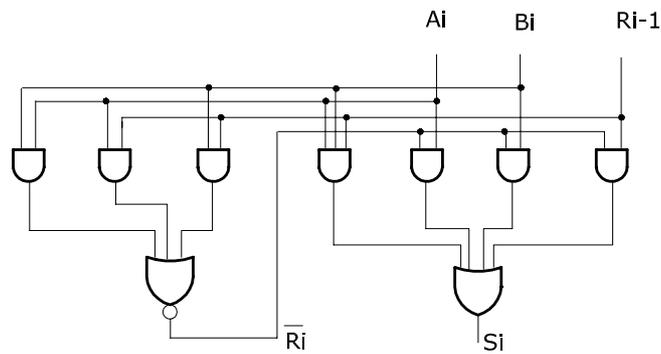


Figura 3.54 Schema per il sommatore completo con uscita S_i in funzione di A_i , B_i , R_i e R_{i-1} . La rete corrisponde – a meno di differenze irrilevanti – a metà del contenuto del componente 7482.

Poiché nella rappresentazione binaria i numeri interi negativi vengono espressi in complemento a 2, basta quel poco di logica che serve per complementare B e sommare 1. Si ottiene così lo schema di principio di Figura 3.55.

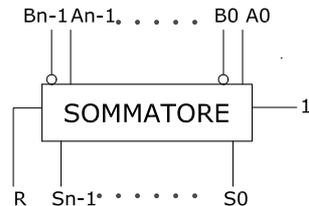


Figura 3.55 Schema della rete che effettua la sottrazione usando un addizzatore.

Invece della rete bloccata di Figura 3.55 conviene prevedere linee di controllo in modo da usare lo stesso sommatore sia per le somme sia per le sottrazioni. Nello schema di Figura 3.56 la linea c_0 seleziona B o il suo complemento, mentre la linea c_1 seleziona come secondo ingresso A oppure 0.

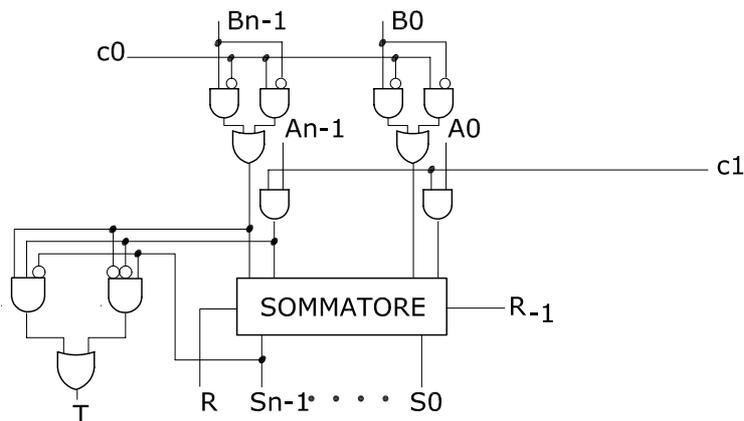


Figura 3.56 Dettaglio della rete che effettua la sottrazione. L'ingresso B può essere presentato al sommatore in forma complementata o no a seconda di c_0 . L'altro ingresso, controllato tramite la linea c_1 , è A oppure 0. Rispetto alla Figura 3.55 è stata aggiunta la logica per il controllo del trabocco di cui si parla più avanti.

Selezionando \overline{B} ($c_0 = 1$) e posto $R_{-1} = 1$, la rete effettua l'operazione:

$$A + \overline{B} + 1$$

ovvero effettua la sottrazione di B da A .

In Tabella 3.4 vengono riportati i risultati in corrispondenza di ogni combinazione di $c_1c_0R_{-1}$. In riferimento alla Tabella 3.4 vale la pena di osservare che:

- per la combinazione $c_1c_0R_{-1} = 010$ la rete esegue un'operazione logica (la complementazione di B);
- per tutte le altre combinazioni la rete effettua operazioni aritmetiche;
- le combinazioni $c_1c_0R_{-1} = 101$ e $c_1c_0R_{-1} = 110$ producono risultati non particolarmente interessanti.

Al Paragrafo 3.9.7 si mostra come aggiungendo altre linee di controllo siano facilmente ottenibili le operazioni logiche AND e OR.

Spieghiamo ora il ruolo delle tre porte di sinistra in Figura 3.56. Se si effettua la somma di due numeri relativi rappresentati con parole di n bit possono capitare questi due risultati straordinari:

- si sommano due numeri positivi e il risultato è negativo;
- si sommano due numeri negativi e il risultato è positivo.

c_1	c_0	R_{-1}	Risultato	Commento
0	0	0	$S = 0 + B = B$	Selezione di B
0	0	1	$S = 0 + \overline{B} + 1 = B + 1$	Incremento di B
0	1	0	$S = 0 + \overline{B} = \overline{B}$	Complementazione di B
0	1	1	$S = 0 + \overline{B} + 1 = -B$	Cambio segno di B
1	0	0	$S = A + B$	Somma $A + B$
1	0	1	$S = A + \overline{B} + 1$	
1	1	0	$S = A + \overline{B} = A - B - 1$	
1	1	1	$S = A + \overline{B} + 1 = A - B$	Differenza $A - B$

Tabella 3.4 Operazioni effettuate dalla rete di Figura 3.56 a seconda degli ingressi di controllo c_1 , c_0 e R_{-1} .

È facile convincersi che questo accade quando il risultato dell'operazione supera la capacità della parola, ovvero supera il massimo numero rappresentabile con quella dimensione di parola. Nel caso di parole di 8 bit si ha *trabocco (overflow)* se: (1) la somma di due numeri positivi risulta maggiore di 127 (massimo numero rappresentabile su 7 bit); (2) la somma di due numeri negativi risulta inferiore a -128 . L'avvenuto trabocco si controlla verificando l'identità dei segni degli addendi e confrontandola con il segno risultante della somma. A tale scopo bastano due porte AND a tre ingressi, collegate come in Figura 3.56 a sinistra.

Per quanto semplice la rete di Figura 3.56 rappresenta una unità aritmetica e logica (ALU). In Figura 3.57, a sinistra, ne viene data una schematizzazione sintetica.

A conclusione di questa sezione conviene riassumere alcuni risultati che abbiamo ottenuto. Siamo partiti da una rete combinatoria per la somma di due numeri in forma binaria. Con l'aggiunta di poca logica abbiamo ottenuto una rete che svolge

l'operazione di sottrazione, un'operazione logica e fornisce indicazione sulla correttezza del risultato. Ovviamente, nella costruzione di una effettiva ALU, si incontrano maggiori difficoltà. Anzitutto non è detto che si riesca a fare un'ALU combinatoria. Se l'operazione è complessa, è possibile che occorra effettuare sequenze di operazioni attraverso reti sequenziali. È il caso tipico delle operazioni in virgola mobile e della relativa unità deputata alla loro esecuzione, denominata FPU (*Floating Point Unit*).¹²

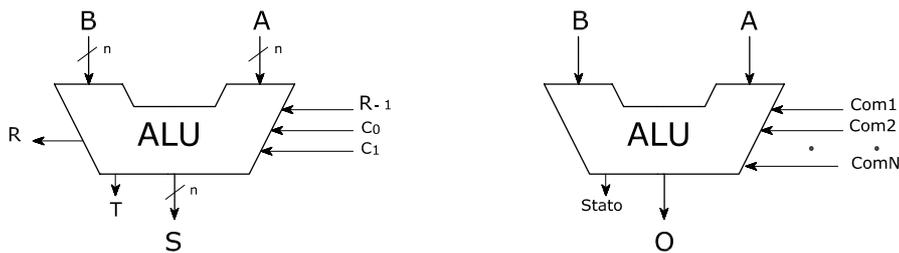


Figura 3.57 A sinistra viene riportata la schematizzazione della rete di Figura 3.56. A destra viene data la schematizzazione di una generica ALU: Com1, Com2, ..., ComN sono le linee di controllo che selezionano le varie operazioni. Qui le linee di comando sono esplicitate. Da un punto di vista realizzativo conviene portare alla ALU i comandi in forma codificata e prevedere sull'ALU un decodificatore per selezionare le singole linee.

3.9.7 Aggiunta di operazioni logiche

Si ponga ora il problema di estendere l'ALU di Figura 3.55 in modo da incorporarvi anche le operazioni logiche di AND e OR tra i due ingressi A e B .

Ovviamente, nei semisommatori sono presenti porte AND e OR; in particolare ogni uscita S_i rappresenta l'uscita di una porta OR. Se si vogliono usare queste porte per effettuare l'OR dei due termini occorre annullare gli effetti dei riporti. Si faccia per esempio riferimento al sommatore di Figura 3.54 e si supponga di modificarlo come in Figura 3.58, dove sono state aggiunte due linee (di controllo) c_3 e c_2 .

Si considerino questi casi.

- Se $c_3 = 0$ e $c_2 = 1$, allora il valore di R_i calcolato dalla rete di Figura 3.58 non risulta alterato rispetto a quello calcolato dalla rete di Figura 3.54; dunque se alla Tabella 3.4 si aggiungono due colonne per c_3 e c_2 , la tabella risultante, su tutte le linee in cui si ha 01 per la coppia $c_3 c_2$, è identica alla Tabella 3.4.

¹²Le unità aritmetiche costituiscono da sole un ampio campo di studio e su di esse esiste una vasta letteratura, si veda ad per esempio [Omo94]. L'interesse di questo libro è volto alla realizzazione della logica di controllo dei calcolatori.

c_3	c_2	c_1	c_0	R_{-1}	Risultato	Commento
0	1	0	0	0	$S = 0 + B = B$	Selezione di B
0	1	0	0	1	$S = 0 + B + 1 = B + 1$	Incremento di B
0	1	0	1	0	$S = 0 + \overline{B} = \overline{B}$	Complementazione di B
0	1	0	1	1	$S = 0 + \overline{B} + 1 = -B$	Cambio segno di B
0	1	1	0	0	$S = A + B$	Somma $A+B$
0	1	1	1	1	$S = A + \overline{B} + 1 = A - B$	Differenza $A-B$
0	0	1	0	0	$S = A \text{ OR } B$	Somma logica di A e B
1	-	1	0	-	$S = A \text{ AND } B$	Prodotto logico di A e B

Tabella 3.5 Principali operazioni effettuate dalla ALU estesa.

- Se $c_3 = 1$, allora $\overline{R_i} = 0$ per ogni i ; dunque il contributo a S_i delle tre porte a due ingressi in cui entra $\overline{R_i}$ risulta nullo, mentre la porta a tre ingressi fornisce $A_i \cdot B_i \cdot 1$. In conclusione la rete di Figura 3.58, con $c_3 = 1$, effettua il calcolo $S_i = A_i \text{ AND } B_i$.
- Se $c_3 = 0$, $c_2 = 0$ e $R_{-1} = 0$, allora $\overline{R_i} = 1$ per ogni i . Conseguentemente la rete calcola $S_i = 1 \cdot 0 + A_i \cdot 1 + B_i \cdot 1 + A_i \cdot B_i \cdot 0 = A_i + B_i$.

In Tabella 3.5 viene riportato il dettaglio delle operazioni di rilievo. La rete effettua altre operazioni oltre a quelle riportate in Tabella 3.5. Per esempio, con la configurazione di controllo "1-11-" la rete calcola $A \text{ AND } \overline{B}$.

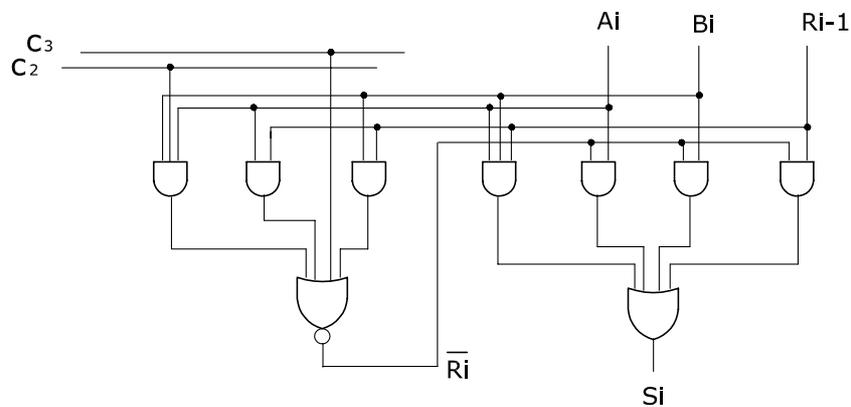


Figura 3.58 Disattivazione degli effetti dei riporti nel sommatore completo attraverso l'aggiunta di due linee di controllo che rendono disponibile l'AND e l'OR dei due ingressi