

正確學會

改訂新版 ディジタル回路と Verilog HDL

Verilog 的 16 堂課

第 14 章

除法器電路設計

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

本章重點

- 14.1 除法電路
- 14.2 延伸學習

14.1 除法電路

- 除法的順序是做多次減法與移位來求出商數與餘數，其運算可推展如下：
 1. 如果「被除數<除數」則不做減法，「被除數 \geq 除數」用被除數對除數做減法
 2. 被除數對除數做減法，結果如果是負（有借位）再對除數做加法，如果是'0'或者正（沒有借位）就不進行加法，直接做下一個位元
 3. 用被除數對除數做減法，如果結果是負的（有借位）就同時對除數做加法，以及對下一個位元做減法運算

$$\begin{array}{r}
 10111100 \\
 -) 1011 \\
 \hline
 \text{沒有借位} \quad 0000 \longrightarrow \oplus \longrightarrow "1"
 \end{array}$$

$$\begin{array}{r}
 0001 \\
 -) 1011 \\
 \hline
 \text{有借位} \quad 0110 \longrightarrow \ominus \longrightarrow "0" \\
 +) 1011 \text{ 加回} \\
 \hline
 0001
 \end{array}$$

$$\begin{array}{r}
 0011 \\
 -) 1011 \\
 \hline
 \text{有借位} \quad 1000 \longrightarrow \ominus \longrightarrow "0" \\
 +) 1011 \text{ 加回} \\
 \hline
 0011
 \end{array}$$

$$\begin{array}{r}
 0110 \\
 -) 1011 \\
 \hline
 \text{有借位} \quad 1011 \longrightarrow \ominus \longrightarrow "0" \\
 +) 1011 \text{ 加回} \\
 \hline
 0110
 \end{array}$$

$$\begin{array}{r}
 1100 \\
 -) 1011 \\
 \hline
 \text{沒有借位} \quad \underbrace{0001} \longrightarrow \oplus \longrightarrow "1" \\
 \text{剩餘數}
 \end{array}$$

▲ 圖 14.1 用加回法做除法的步驟

- 減法的結果如果是負的，在做下一個位元的減法時把除數做加法即可，稱為減放法
- 下頁圖商數在 *1 為 MSB, *5 為 LSB 所以為「10001」，餘數則為 *5 運算結果的「0001」。不過減放法在遇到商數的 LSB 為 '0' 的時候，因為餘數仍舊是負的，所以需要把除數加回來

$$*1 \quad \begin{array}{r} 010111100 \\ -) 1011 \\ \hline \text{沒有借位} \end{array} \quad \begin{array}{r} 00000 \\ \longrightarrow \oplus \longrightarrow "1" \end{array}$$

$$\begin{array}{r} 010011101 \\ -) 1011 \\ \hline \text{有借位} \end{array} \quad \begin{array}{r} 11110 \\ \longrightarrow \ominus \longrightarrow "0" \end{array}$$

$$*2 \quad \begin{array}{r} 00001 \\ -) 1011 \\ \hline \text{有借位} \end{array} \quad \begin{array}{r} 10110 \\ \longrightarrow \ominus \longrightarrow "0" \end{array}$$

$$\begin{array}{r} 11101 \\ +) 1011 \\ \hline \text{有借位} \end{array} \quad \begin{array}{r} 01000 \\ \longrightarrow \oplus \longrightarrow "1" \end{array}$$

$$*3 \quad \begin{array}{r} 01101 \\ +) 1011 \\ \hline \text{沒有借位} \end{array} \quad \begin{array}{r} 11000 \\ \longrightarrow \ominus \longrightarrow "0" \end{array}$$

$$\begin{array}{r} 10001 \\ -) 1011 \\ \hline \text{沒有借位} \end{array} \quad \begin{array}{r} 00110 \\ \longrightarrow \oplus \longrightarrow "1" \end{array}$$

$$*4 \quad \begin{array}{r} 10000 \\ +) 1011 \\ \hline \text{沒有借位} \end{array} \quad \begin{array}{r} 11011 \\ \longrightarrow \ominus \longrightarrow "0" \end{array}$$

$$\begin{array}{r} 01100 \\ -) 1011 \\ \hline \text{沒有借位} \end{array} \quad \begin{array}{r} 00001 \\ \longrightarrow \oplus \longrightarrow "1" \end{array}$$

$$*5 \quad \begin{array}{r} 10110 \\ +) 1011 \\ \hline \text{有借位} \end{array} \quad \begin{array}{r} 00001 \\ \hline \text{剩餘數} \end{array} \quad \rightarrow \oplus \longrightarrow "1"$$

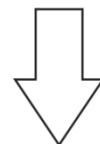
▲ 圖 14.2 用減放法做除法的步驟

$$\begin{array}{r} 00011 \\ -) 1011 \\ \hline \text{有借位} \end{array} \quad \begin{array}{r} 11000 \\ \longrightarrow \ominus \longrightarrow "0" \end{array}$$

$$\begin{array}{r} 1011 \\ +) 1011 \\ \hline \end{array} \quad \begin{array}{r} 00011 \\ \hline \text{剩餘數} \end{array}$$

▲ 圖 14.3 減放法當商數的 LSB 為 '0' 的時候

加法/減法	進位/借位	結果	商數	下個位元的運算
加法	無	(\ominus)	0	加法
加法	有	(\oplus)	1	減法
減法	無	(\oplus)	1	減法
減法	有	(\ominus)	0	加法



加法/減法	進位/借位	結果	商數	下個位元的運算
0	0	0	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	0



加法：'0'

減法：'1'



(\ominus) : '0'

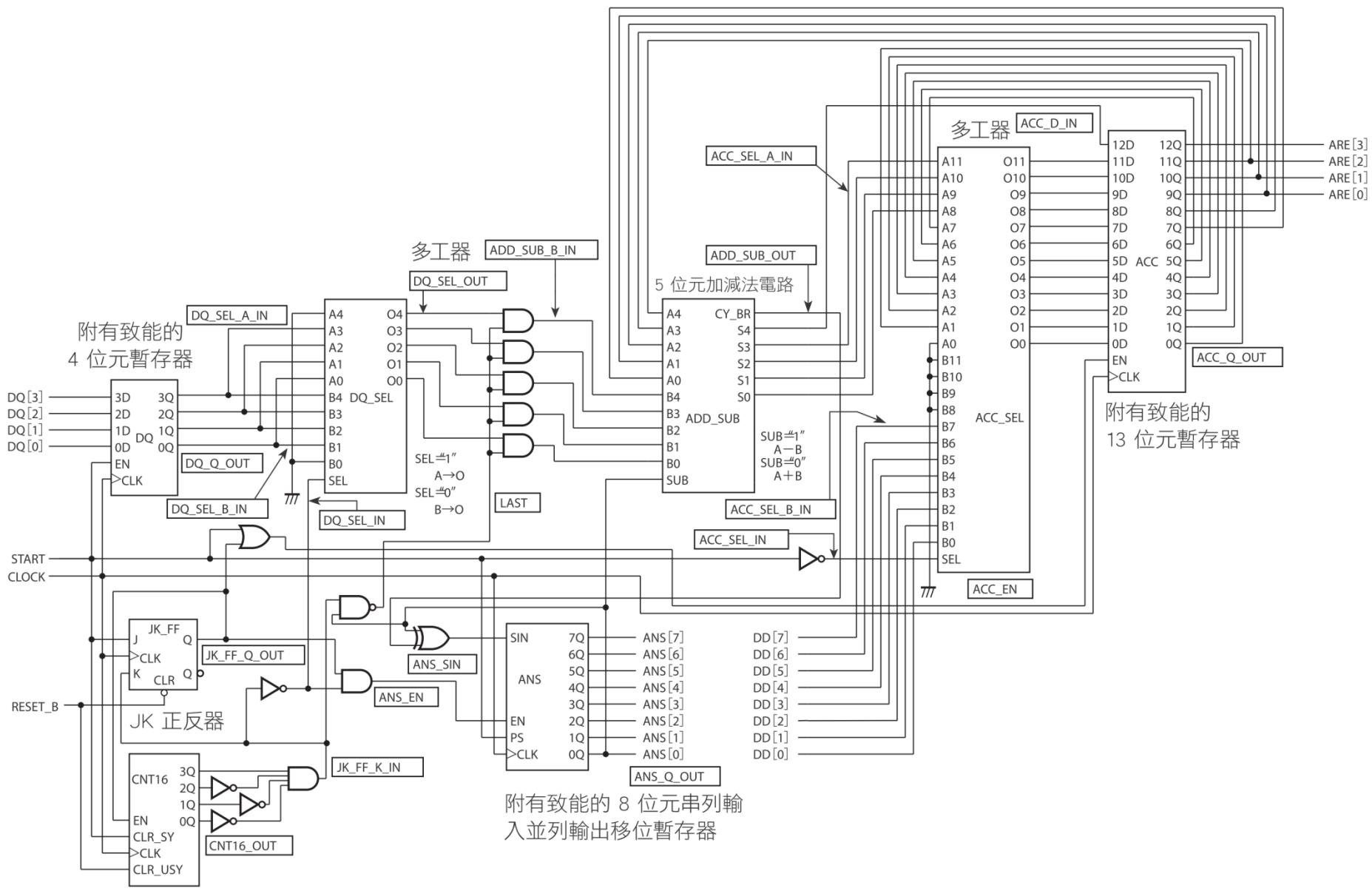
(\oplus) : '1'



▲表 14.1 用減放法來做除法的步驟

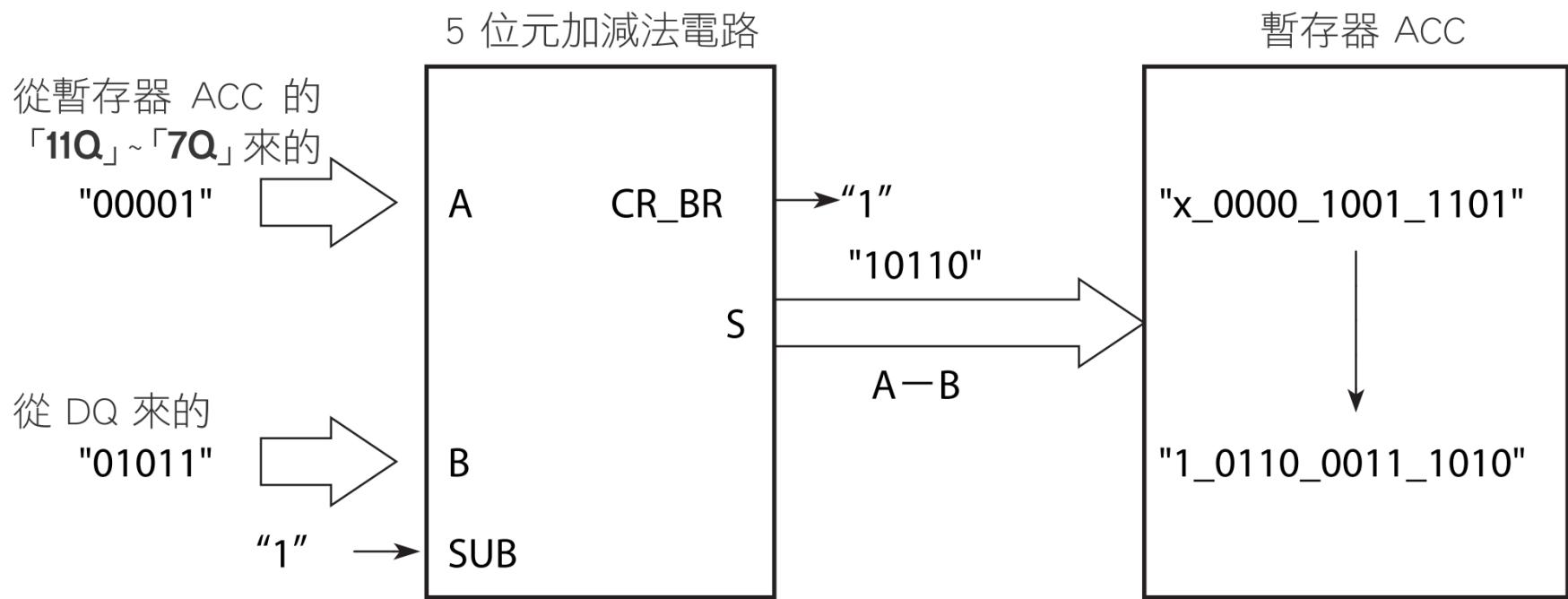
14.1.1 電路構成

- 運算最開始的時候先紀錄被除數，從被除數對除數邊做移位邊相減，最後要記錄餘數所需的暫存器稱做「ACC」
- 放置除數的暫存器稱做「DQ」
- 根據從被除數對除數做相減的結果求出來的商數做依序紀錄的移位暫存器稱為「ANS」。

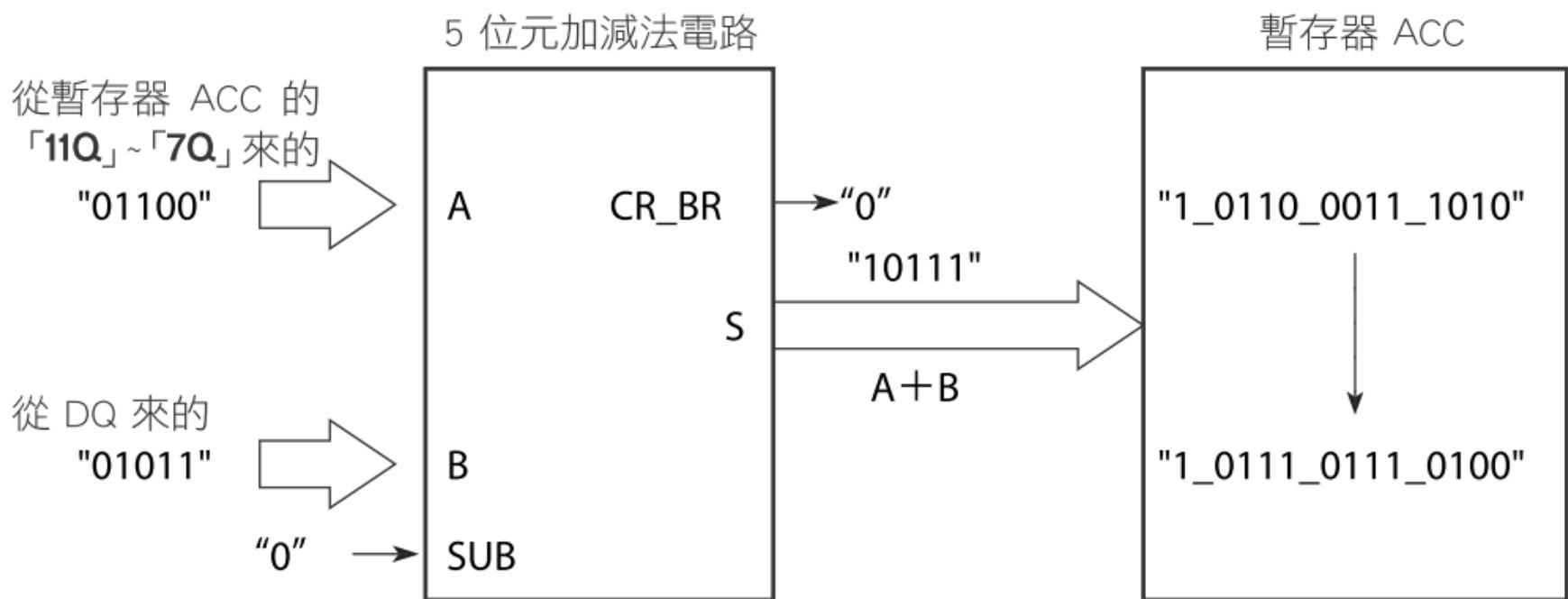


▲ 圖 14.4 8 位元 ÷ 4 位元除法電路

- 注意暫存器 ACC的移位機能
- 可用來實現「每 1 位元做移位且與除數做相減或相加」
 - 「11Q」 → 「A4」 → 「S4」 → 「12D」
 - 「10Q」 → 「A3」 → 「S3」 → 「11D」
 - ..
 - ..
 - 「1Q」 → 「2D」
 - 「0Q」 → 「1D」



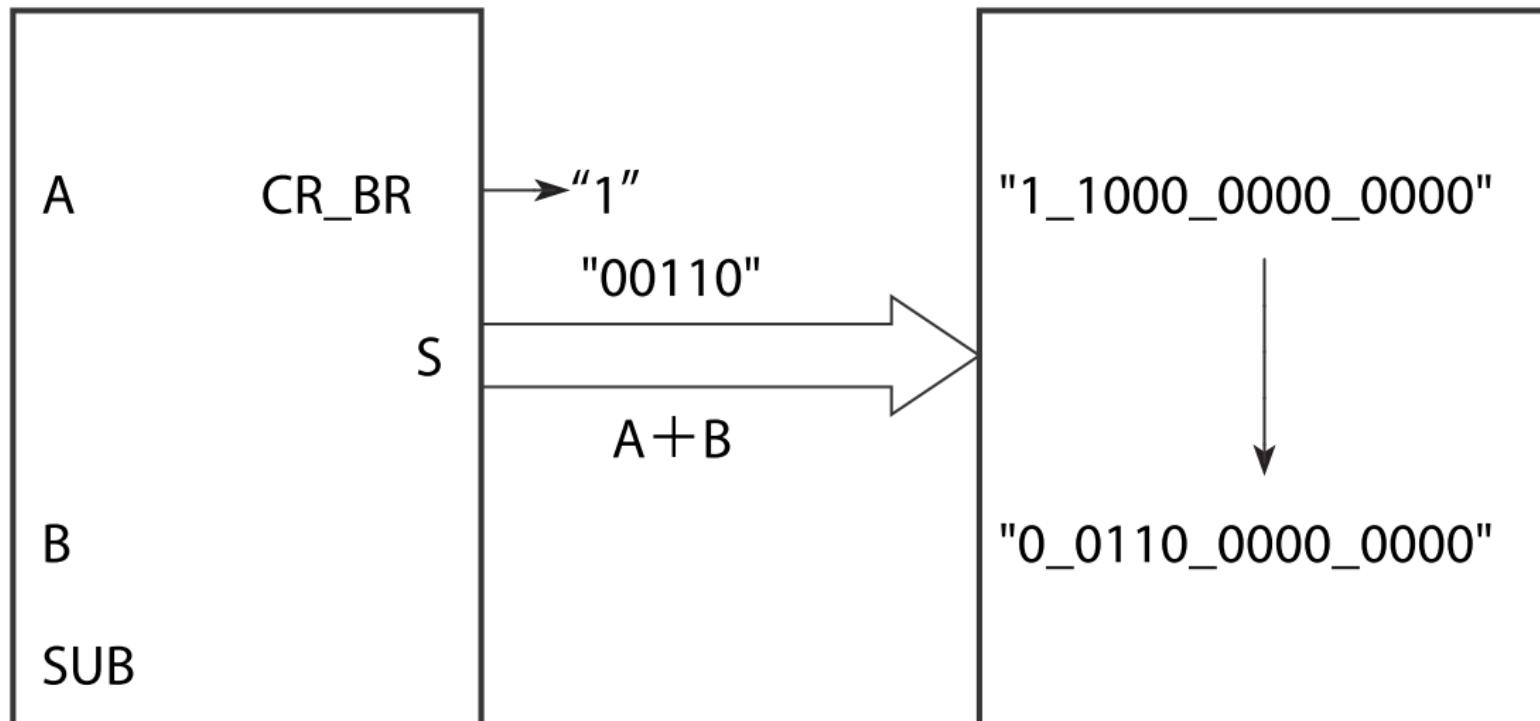
▲ 圖 14.7 為求出 MSB 的商數用 ACC 與 DQ 做相減



▲ 圖 14.8 為求出第 6 位元的商數用 ACC 與 DQ 做相加

5 位元加減法電路

暫存器 ACC



▲ 圖 14.9 為了求餘數做 DQ 與 ACC 的相加