2006年度 ハードウェア構成法 中間試験解説

林崎 弘成

平成 19 年 6 月 15 日

1　問題 1: $A - B - 1$

1.1　問題

$A = [A_3, A_2, A_1, A_0]$, $B = [B_3, B_2, B_1, B_0]$ を 4 bit の整数とする。
$A - B - 1$ を計算する回路を示してください。

1.2　入出力範囲とビット数

入力は $-8 \leq A \leq +7$, $-8 \leq B \leq +7$。

出力は $-16 \leq A - B - 1 \leq +14$ となります ($A = -8, B = +7$ の時に $A - B - 1 = -16$, $A = +7, B = -8$ の時に $A - B - 1 = +14$)。
よって、出力は 5 bit の整数 となります。

以降、出力を $C = [C_4, C_3, C_2, C_1, C_0]$ と表記します。

1.3　解法 1

$-B = \overline{B} + 1$ であるので、

$$\begin{aligned} A - B - 1 &= A + (-B) - 1 \\ &= A + (\overline{B} + 1) - 1 \\ &= A + \overline{B} \end{aligned} \tag{1}$$

となります。

ここで、結果は 5 bit なので、普通の正数加算器に入れる場合にはまず 5 bit に符号拡張してから $A + \overline{B}$ の計算を行う必要があります。筆算の形で表現すると以下のようになります。

$$\begin{array}{cccccc} & A_3 & A_3 & A_2 & A_1 & A_0 \\ + & \overline{B_3} & \overline{B_3} & \overline{B_2} & \overline{B_1} & \overline{B_0} \\ \hline & C_4 & C_3 & C_2 & C_1 & C_0 \end{array}$$

1.4　誤答例

ここで、以下のように 4 bit のまま計算すると、結果の最上位ビット (C_4) がずれてしまいます。

$$\begin{array}{cccccc} & & A_3 & A_2 & A_1 & A_0 \\ + & & \overline{B_3} & \overline{B_2} & \overline{B_1} & \overline{B_0} \\ \hline & C_4 & C_3 & C_2 & C_1 & C_0 \end{array}$$

例えば、$A=[0000], B=[0000]$ の場合、正答は $C=A-B-1=0-0-1=-1=[11111]$ になりますが、上のような筆算をすると

$$
\begin{array}{cccccc}
 & & 0 & 0 & 0 & 0 \\
+ & & 1 & 1 & 1 & 1 \\
\hline
 & 0 & 1 & 1 & 1 & 1
\end{array}
$$

となって、$C=[01111]=15$ となってしまいます。

1.5 解法 2

$A+\overline{B}$ を計算する点では解法 1 と同じですが、その計算を工夫することで少し回路がすっきりしたものになります。

まず、整数 (符号付き) A の最上位ビットを反転させて、$A'=[\overline{A_3}, A_2, A_1, A_0]$ とします。この A' を正数 (符号無し) として解釈すると、$A'=A+8$ となります。

A	A'
1000 = -8	0000 = 0
1001 = -7	0001 = 1
1010 = -6	0010 = 2
1011 = -5	0011 = 3
1100 = -4	0100 = 4
1101 = -3	0101 = 5
1110 = -2	0110 = 6
1111 = -1	0111 = 7
0000 = 0	1000 = 8
0001 = +1	1001 = 9
0010 = +2	1010 = 10
0011 = +3	1011 = 11
0100 = +4	1100 = 12
0101 = +5	1101 = 13
0110 = +6	1110 = 14
0111 = +7	1111 = 15

同様に、$\overline{B}$ の最上位ビットを反転させて $\overline{B}'=[B_3, \overline{B_2}, \overline{B_1}, \overline{B_0}]$ とすると、$\overline{B}'=\overline{B}+8$ となります。

$A', \overline{B}'$ は共に 4 bit 正数 となるので、これをそのまま正数加算器で足し合わせます。

$$
\begin{array}{cccccc}
 & & \overline{A_3} & A_2 & A_1 & A_0 \\
+ & & B_3 & \overline{B_2} & \overline{B_1} & \overline{B_0} \\
\hline
 & C'_4 & C'_3 & C'_2 & C'_1 & C'_0
\end{array}
$$

こうして出た結果は、

$$
\begin{aligned}
C' &= [C'_4, C'_3, C'_2, C'_1, C'_0] \\
&= A' + \overline{B}' \\
&= (A+8) + (\overline{B}+8) \\
&= A + \overline{B} + 16 \qquad (2)
\end{aligned}
$$

となっているので、なので、$A+\overline{B}$ を出すには $C'-16$ とする必要があります (以下では、$-16=(10000)_2$ を足しています)。

$$\begin{array}{rccccc} & C_4' & C_3' & C_2' & C_1' & C_0' \\ + & 1 & 0 & 0 & 0 & 0 \\ \hline & C_4 & C_3 & C_2 & C_1 & C_0 \end{array}$$

まとめると、下のような筆算になります。

$$\begin{array}{rccccc} & & \overline{A_3} & A_2 & A_1 & A_0 \\ & & B_3 & \overline{B_2} & \overline{B_1} & \overline{B_0} \\ + & 1 & 0 & 0 & 0 & 0 \\ \hline & C_4 & C_3 & C_2 & C_1 & C_0 \end{array}$$

この計算は、$A' + \overline{B}'$ を 4 bit の加算器で計算して、その最上位ビットを反転させるとできます。

2 問題 2: $C \times C$ の上位 2 bit

2.1 問題

$C = [C_5, C_4, C_3, C_2, C_1, C_0]$ を 6 bit の整数とする。
$C \times C$ の最上位の 2 bit を求める回路を示してください。

2.2 入出力範囲とビット数

入力は $-32 \le C \le +31$。
出力は $0 \le C \times C \le +1024$ となります ($C = -32$ の時に $C \times C = +1024$)。
よって、出力は 11 bit の 正数 となります。
以降、$C \times C$ を $P = [P_{10}, P_9, P_8, P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0]$ と表記します。求めるべきはこの最上位 2 bit、P_{10}, P_9 です。

2.3 解法 1

各 bit が 1 になる値の範囲を計算して、それを論理圧縮する方法です。
P_{10} は $2^{10} = 1024$ の位なので、このビットが 1 になるのは、$P = C \times C \ge 1024$ つまり $|C| \ge 32$ の時です。入力範囲内でこの条件を満たすのは $C = -32 = [100000]$ の時だけなので、

$$P_{10} = C_5 \cdot \overline{C_4} \cdot \overline{C_3} \cdot \overline{C_2} \cdot \overline{C_1} \cdot \overline{C_0} \tag{3}$$

となります。
P_{10} は $2^{10} = 512$ の位なので、このビットが 1 になるのは、$1024 > P \ge 512$ ($P_{10} = 0$ の場合) または $2048 > P \ge 1536$ ($P_{10} = 1$ の場合) の時です。$P = C \times C$ は最大で 1024 なので、一番目の場合だけを考えると、$32 > |C| \ge \sqrt{512} = 22.6...$ つまり $-31 \le C \le -23$ または $23 \le C \le 31$ となります。
P_9 が 1 になる入力ビットパターンを列挙すると、以下のようになります。

C	C_5	C_4	C_3	C_2	C_1	C_0
-31	1	0	0	0	0	1
-30	1	0	0	0	1	0
-29	1	0	0	0	1	1
-28	1	0	0	1	0	0
-27	1	0	0	1	0	1
-26	1	0	0	1	1	0
-25	1	0	0	1	1	1
-24	1	0	1	0	0	0
-23	1	0	1	0	0	1
+23	0	1	0	1	1	1
+24	0	1	1	0	0	0
+25	0	1	1	0	0	1
+26	0	1	1	0	1	0
+27	0	1	1	0	1	1
+28	0	1	1	1	0	0
+29	0	1	1	1	0	1
+30	0	1	1	1	1	0
+31	0	1	1	1	1	1

軽く論理圧縮すると、以下のようになります。$-$ は Don't Care を表します。

C	C_5	C_4	C_3	C_2	C_1	C_0
-31	1	0	0	0	0	1
-30〜-29	1	0	0	0	1	-
-28〜-25	1	0	0	1	-	-
-24〜-23	1	0	1	0	0	-
$+23$	0	1	0	1	1	1
$+24$〜$+31$	0	1	1	-	-	-

さらに論理圧縮して、

C_5	C_4	C_3	C_2	C_1	C_0
1	0	0	-	-	1
1	0	0	-	1	-
1	0	0	1	-	-
1	0	1	0	0	-
0	1	-	1	1	1
0	1	1	-	-	-

式に落とすと、例えば以下のようになる。

$$P_9 = C_5 \cdot \overline{C_4} \cdot \{\overline{C_3} \cdot (C_2 + C_1 + C_0) + C_3 \cdot \overline{C_2} \cdot \overline{C_1}\} \tag{4}$$
$$+ \overline{C_5} \cdot C_4 \cdot \{C_3 + (C_2 \cdot C_1 \cdot C_0)\} \tag{5}$$

2.3.1 注意点と誤答例

-32 のビットパターンを [111111] としている答案がありましたが、[111111] $= -1$ です。

また、$C=-32$ の場合には、$C \times C = +1024 = [10000000000]$ なので、$P_9 = 0$ です。$|C| \geq \sqrt{512}$ だけだと $C=-32$ の時に $P_9 = 1$ となってしまいますので、$C \neq -32$ をきちんと入れることが必要です。[1]

2.4 解法 2

6 bit 整数の乗算器を書いてその上位 2 bit を出力する方法です。

整数乗算は以下のようになります。所々反転していることの解説は省きます (2004 年度の中間試験解説[2]などに解説があるようです)。最後の定数は $0 \times 0 = 0$ となるようにしています。

$$
\begin{array}{cccccccccccc}
 & & & & & & \overline{C_0 \cdot C_5} & C_0 \cdot C_4 & C_0 \cdot C_3 & C_0 \cdot C_2 & C_0 \cdot C_1 & C_0 \cdot C_0 \\
 & & & & & \overline{C_1 \cdot C_5} & C_1 \cdot C_4 & C_1 \cdot C_3 & C_1 \cdot C_2 & C_1 \cdot C_1 & C_1 \cdot C_0 & \\
 & & & & \overline{C_2 \cdot C_5} & C_2 \cdot C_4 & C_2 \cdot C_3 & C_2 \cdot C_2 & C_2 \cdot C_1 & C_2 \cdot C_0 & & \\
 & & & \overline{C_3 \cdot C_5} & C_3 \cdot C_4 & C_3 \cdot C_3 & C_3 \cdot C_2 & C_3 \cdot C_1 & C_3 \cdot C_0 & & & \\
 & & \overline{C_4 \cdot C_5} & C_4 \cdot C_4 & C_4 \cdot C_3 & C_4 \cdot C_2 & C_4 \cdot C_1 & C_4 \cdot C_0 & & & & \\
 & C_5 \cdot C_5 & \overline{C_5 \cdot C_4} & \overline{C_5 \cdot C_3} & \overline{C_5 \cdot C_2} & \overline{C_5 \cdot C_1} & \overline{C_5 \cdot C_0} & & & & & \\
+ & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
 & P_{10} & P_9 & P_8 & P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}
$$

2 回出現する項を整理すると、

$$
\begin{array}{cccccccccccc}
 & \overline{C_4 \cdot C_5} & \overline{C_3 \cdot C_5} & \overline{C_2 \cdot C_5} & \overline{C_1 \cdot C_5} & \overline{C_0 \cdot C_5} & C_0 \cdot C_4 & C_0 \cdot C_3 & C_0 \cdot C_2 & C_0 \cdot C_1 & 0 & C_0 \\
 & C_5 & & C_3 \cdot C_4 & C_2 \cdot C_4 & C_1 \cdot C_4 & C_1 \cdot C_3 & C_1 \cdot C_2 & & C_1 & & \\
 & & & C_4 & & C_2 \cdot C_3 & & C_2 & & & & \\
 & & & & & C_3 & & & & & & \\
+ & & & & & 1 & & & & & & \\
\hline
 & P_{10} & P_9 & P_8 & P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}
$$

あとはこれを足し合わせます。下の桁についてはキャリーだけ計算すれば済みます。

また、少し論理圧縮を利かせることもできます。P_2 の桁から P_3 の桁への繰り上がりがある条件は $C_0 \cdot C_1$ と C_1 が両方 1 の時つまり $(C_0 \cdot C_1) \cdot C_1$ で、これは $C_0 \cdot C_1$ と同値です。次に P_3 の桁から P_4 の桁への繰り上がりがある条件は元々P_3 の桁にあった $C_0 \cdot C_2$ と、P_2 の桁からの繰り上がり $C_0 \cdot C_1$ とが両方 1 の時つまり $(C_0 \cdot C_2) \cdot (C_0 \cdot C_1)$ で、これは $C_0 \cdot C_1 \cdot C_2$ と圧縮できます。これ以上の桁についても同様に論理圧縮していくと、原理的には解法 1 と同じものになるはずです[3]。

---

[1] この誤答をしたうちの一人は私です。

[2] `http://www-hiraki.is.s.u-tokyo.ac.jp/lectures/hw/test2004.pdf`

[3] 私は実際に論理圧縮して確かめてみた訳では*ありません*。かなり大変なことになると思います。