総

学習指導要領（2）- 知・技 - ア
学習指導要領（3）- 知・技 - ア
学習指導要領（2）- 思・判・表 - ア
学習指導要領（3）- 思・判・表 - ア
学習内容（2）- ア メディアとコミュニケーション
学習内容（3）- ア コンピュータの仕組みと処理

情報 IV

4ビットの2進数の加減算を行う回路を設計する手順を考える。ただし、$A+B$ は A と B の論理和（OR）を表し、$A \cdot B$ は A と B の論理積（AND）を表す。また、$\overline{A}$ は A の否定（NOT）を表す。$\overline{A+B}$ は A と B の論理和の結果を否定した否定論理和（NOR）であり、$\overline{A \cdot B}$ は A と B の論理積の結果を否定した否定論理積（NAND）である。

(ア) 次の文章の空欄 (89)(90)(91)(92) ～ (101)(102)(103)(104)(105)(106) にあてはまる数字を解答欄にマークしなさい。

ある自然数に数値を加算してもとの自然数を1桁増やすことを考えよう。このとき加算する最小の数値を、ある基数を指定して、その基数の補数と呼ぶ。例えば、2桁の10進数 46_{10} の10（基数）の補数は、加算して3桁になる数なので 54_{10} となる。n ビットの2進数の場合も同様に、元の数を 2^n にするために補う数として、2の補数を求めることができる。4ビットの2進数を $A_3A_2A_1A_0$ とすると、

$$A_3A_2A_1A_0 + Y_3Y_2Y_1Y_0 \ = \ 10000_2$$

となる $Y_3Y_2Y_1Y_0$ は、$A_3A_2A_1A_0$ の2の補数となる。具体的な手順は次のようになる。

手順1　もとの2進数の各ビットを反転（1であれば0に、0であれば1に変換）する。
手順2　反転して得られた2進数に1を加算する。

この2の補数を用いて整数を表現する方法を考える。4ビットの2進数を次のように正の整数、ゼロ、負の整数と対応させることにしよう。

10進数	2進数	10進数	2進数
−8	1000	0	0000
−7	1001	1	0001
−6	1010	2	0010
−5	1011	3	0011
−4	1100	4	0100
−3	1101	5	0101
−2	1110	6	0110
−1	1111	7	0111

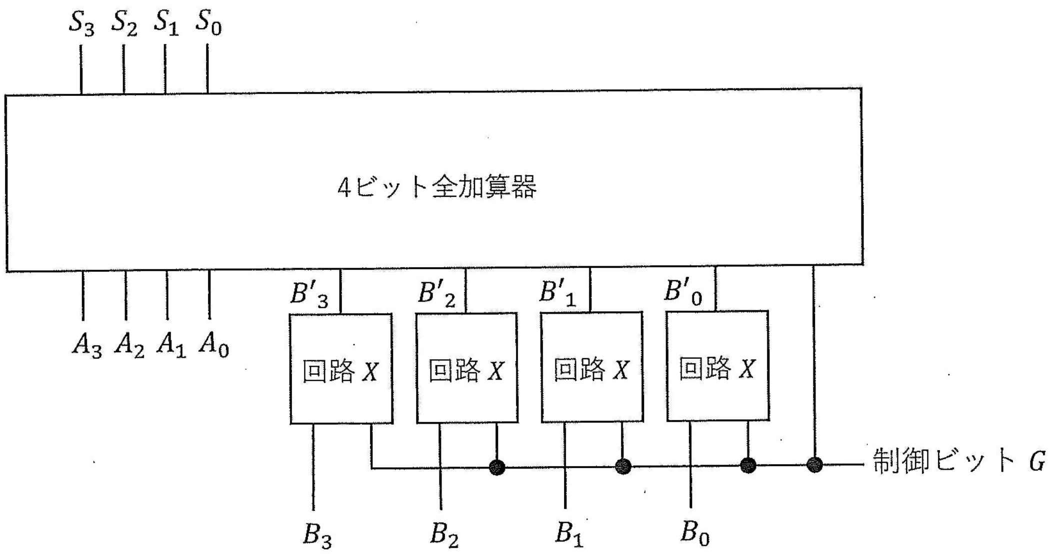
このとき、10進数 6_{10} に対応する2進数 0110_2 の2の補数は (89)(90)(91)(92) $_2$ であり、10進数の (93)(94) $_{10}$ に相当する。このような正負の数値の割り当て方法が2の補数を用いた整数表現である。ただし、4ビットの2進数 0000_2 の2の補数は 0000_2 とする。同様に2の補数を用いた16ビットの整数表現を用いると、(95)(96)(97)(98)(99)(100) $_{10}$ ～ (101)(102)(103)(104)(105)(106) $_{10}$ までの整数を扱うことができる。

(イ) 次の文章の空欄 $\boxed{(107)}$ にあてはまるもっとも適したものを下の選択肢から選び、その番号を解答欄にマークしなさい。

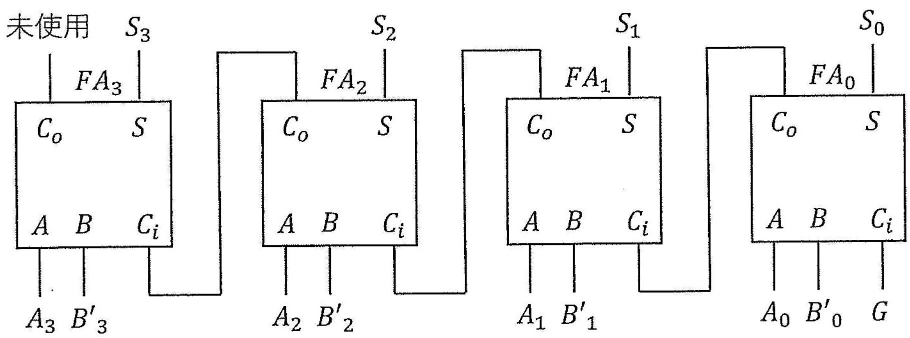
2 の補数を用いた 4 ビットの整数表現を用いると、次のように加算器を用いて 4 ビットの 2 進数の減算を行うことが可能となる。ただし、計算結果が −8〜7 の範囲に収まる整数を対象とするものとする。

$$
\begin{aligned}
A_3A_2A_1A_0 - B_3B_2B_1B_0 &= A_3A_2A_1A_0 + (-B_3B_2B_1B_0) \\
&= A_3A_2A_1A_0 + (B_3B_2B_1B_0\text{の 2 の補数})
\end{aligned}
$$

4 ビットの 2 進数演算 $A_3A_2A_1A_0 \pm B_3B_2B_1B_0 = S_3S_2S_1S_0$ を対象に、次図の上のような回路構成を考えよう。多くの CPU では、図のような回路で実際に加減算を行う。4 ビット全加算器回路は、次図の下のように 4 つの全加算器から構成される。各全加算器（FA）は A、B、および下位ビットからの桁上がり C_i を入力とし、加算した結果を和 S および桁上がり C_o として出力する。



4 ビット加減算器の全体構成



4 ビット全加算器の内部構成

上図の回路では、制御ビット $G = 0$ の場合に加算 $A_3A_2A_1A_0 + B_3B_2B_1B_0 = S_3S_2S_1S_0$ が実行され、$G = 1$ の場合に減算 $A_3A_2A_1A_0 - B_3B_2B_1B_0 = S_3S_2S_1S_0$ が実行されることとする。入力 B_k と G か

ら B_k' を出力する回路を X とすると、回路 X の真理値表は次のようになる。

B_k	G	B_k'
0	0	0
0	1	1
1	0	1
1	1	0

真理値表より、B_k' を求める論理式は次のようになる。

$$B_k' = \boxed{\text{(107)}} \tag{1}$$

【$\boxed{\text{(107)}}$ の選択肢】

(1) $\overline{B_k} \cdot \overline{G}$　(2) $\overline{B_k} \cdot G$　(3) $B_k \cdot \overline{G}$　(4) $B_k \cdot G$
(5) $\overline{B_k} \cdot \overline{G} + \overline{B_k} \cdot G$　(6) $\overline{B_k} \cdot \overline{G} + B_k \cdot \overline{G}$　(7) $\overline{B_k} \cdot \overline{G} + B_k \cdot G$　(8) $\overline{B_k} \cdot G + B_k \cdot \overline{G}$
(9) $\overline{B_k} \cdot G + B_k \cdot G$　(0) $B_k \cdot \overline{G} + B_k \cdot G$

(ウ) 次の文章の空欄 $\boxed{\text{(108)}}\boxed{\text{(109)}}$ にあてはまる数字を解答欄にマークしなさい。

回路 X をNAND（否定論理積）回路のみから構成することを考える。式（1）に対して、復元の法則とド・モルガンの定理を順に適用すると、式（2）のように変形できる。

$$B_k' = \overline{\overline{\overline{B_k} \cdot G} \cdot \overline{B_k \cdot \overline{G}}} \tag{2}$$

式（2）はさらに変形が可能であり、1つの回路 X を構成するNAND回路の数は、最小で $\boxed{\text{(108)}}\boxed{\text{(109)}}$ 個となる。論理式の変形には、次に示す論理演算の諸定理を用いてよい。

公理	恒等の法則
$1 + A = 1$ $0 \cdot A = 0$	$0 + A = A$ $1 \cdot A = A$
同一の法則	**補元の法則**
$A + A = A$ $A \cdot A = A$	$A + \overline{A} = 1$ $A \cdot \overline{A} = 0$
交換の法則	**結合の法則**
$A + B = B + A$ $A \cdot B = B \cdot A$	$A + (B + C) = (A + B) + C$ $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
分配の法則	**吸収の法則**
$A \cdot (B + C) = A \cdot B + A \cdot C$ $(A + B) \cdot (A + C) = A + B \cdot C$	$A \cdot (A + B) = A$ $A + A \cdot B = A$ $A + \overline{A} \cdot B = A + B$ $\overline{A} + A \cdot B = \overline{A} + B$
復元の法則	**ド・モルガンの定理**
$\overline{\overline{A}} = A$	$\overline{A \cdot B} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A} \cdot \overline{B}$