

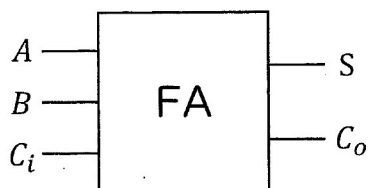
情報Ⅳ

複数ビット同士の加算を行う回路を設計する手順を考える。次の文章の空欄 (65) から (68) および (69) (70)、(77) (78) には適切な数字を、空欄 (71) (72) から (75) (76) にはもっとも適したものを選択肢から選び、解答欄にマークしなさい。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、 \bar{A} は A の否定 (NOT) を表す。

学習指導要領 (3) - 知・技 - ア
学習指導要領 (3) - 思・判・表 - ア

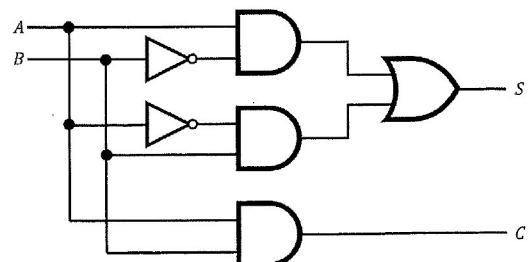
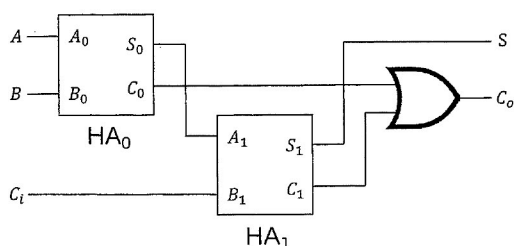
学習内容 (3) - ア コンピュータの仕組みと処理

(ア) 次図 (左) に示す回路記号は 1 ビットの全加算器回路 (FA) であり、その動作は図 (右) の真理値表のようになる。全加算器回路は 1 ビットの入力 A 、 B 、および下位ビットの加算によって生じた桁上がり C_i を入力とし、算術加算した結果を和 S および桁上がり C_o として出力する。



| A | B | C_i | S | C_o |
|-----|-----|-------|-----|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

全加算器回路は次図 (左) に示すように、2 個の半加算器回路 (HA) と 1 個の OR 回路で構成することができる。個々の半加算器回路は 1 ビットの入力 A 、 B を算術加算した結果を、和 S 、桁上がり C として出力する回路であり、基本論理回路 (論理ゲート) である OR 回路、AND 回路、NOT 回路を用いて、次図 (右) のように構成することができる。

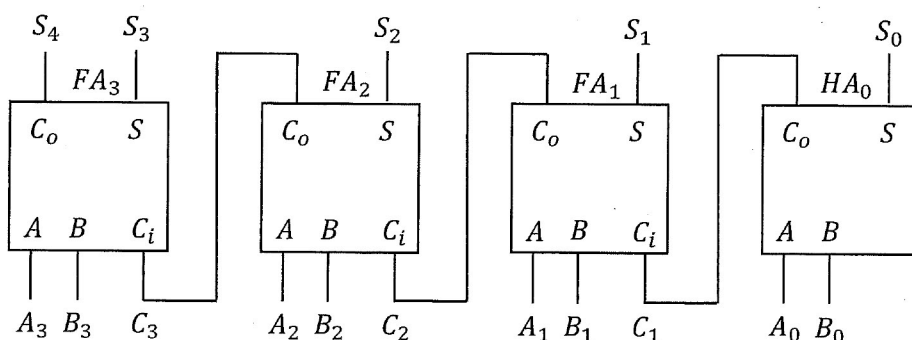


論理回路の動作速度は、入力から出力に至る回路に存在する AND 回路と OR 回路の個数（段数）に大きく影響される。半加算器回路では、入力 A または B から出力 S に至る段数は $\boxed{(65)}$ 、出力 C までの段数は $\boxed{(66)}$ である。一方、全加算器回路の場合は、出力 S までの段数は $\boxed{(67)}$ 、出力 C までの段数は $\boxed{(68)}$ である。ただし、NOT 回路は段数に含めない。

学習指導要領 (3) - 知・技 - ア
 学習指導要領 (3) - 知・技 - ウ
 学習指導要領 (3) - 思・判・表 - ア
 学習指導要領 (3) - 思・判・表 - ウ
 学習内容 (3) - ア コンピュータの仕組みと処理
 学習内容 (3) - ウ モデル化とシミュレーション

(イ) 3 個の全加算器回路 (FA) と 1 個の半加算器回路 (HA) を並べることで、下図のような 4 ビットの並列加算回路を実現できる。

$$A_3A_2A_1A_0 + B_3B_2B_1B_0 = S_4S_3S_2S_1S_0$$



上記の回路構成の場合、 A_0 の入力から最終桁の S_4 が出力されるまでの段数は $\boxed{(69)} \boxed{(70)}$ となり、桁数が増えたとこの段数の増加が大きな問題となってくる。そこで、この段数を減らす方法を考えてみよう。

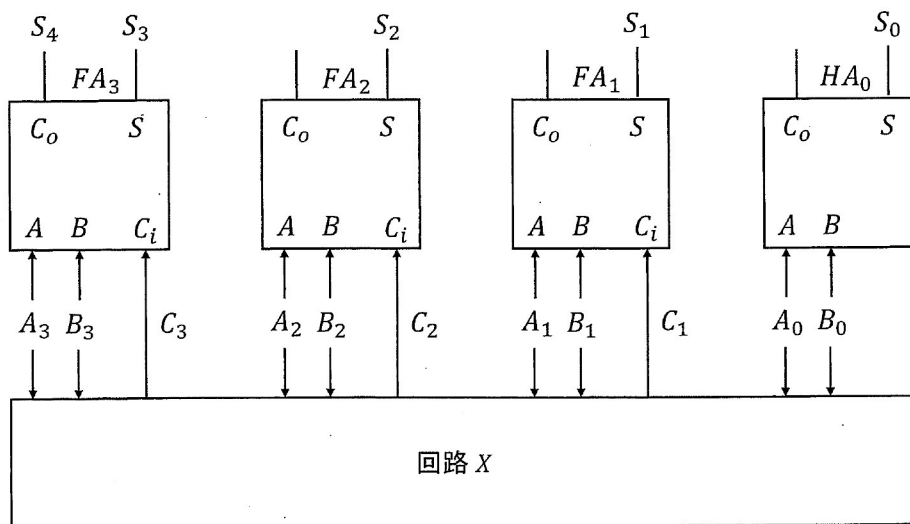
最下位ビットの加算による桁上がり C_1 およびそれに続く桁の桁上がり C_{k+1} は、真理値表から論理式を導出し、論理演算における分配の法則や吸収の法則を用いて変形することで、次のような式で表現できる。

$$C_1 = \boxed{(71)} \boxed{(72)} \quad (1)$$

$$C_{k+1} = \boxed{(73)} \boxed{(74)} + C_k \cdot (\boxed{(75)} \boxed{(76)}) \quad (2)$$

| 分配の法則 | 吸収の法則 |
|---|-------------------------------------|
| $A \cdot (B + C) = A \cdot B + A \cdot C$ | $A \cdot (A + B) = A$ |
| $(A + B) \cdot (A + C) = A + B \cdot C$ | $A + A \cdot B = A$ |
| | $A + \bar{A} \cdot B = A + B$ |
| | $\bar{A} + A \cdot B = \bar{A} + B$ |

式 (1)(2) を展開し、次図のような 4 ビット 並列加算回路を構成しよう。



ここで回路 X は、 $A_3A_2A_1A_0$ と $B_3B_2B_1B_0$ を入力することで、 C_1, C_2, C_3 が出力される回路である。桁数が増えると回路そのものは複雑になるが、回路 X の入力から出力までの段数は高々 $\boxed{(77)} \boxed{(78)}$ となる。ただし、AND 回路、OR 回路ともに、入力数 (2 入力、3 入力、...、 n 入力) に関わらず段数を 1 と数える。

$\boxed{(71)} \boxed{(72)} \sim \boxed{(75)} \boxed{(76)}$ の選択肢】

- | | | | |
|------------------------------|---|---|--|
| (11) $A_0 \cdot B_0$ | (12) $\overline{A_0} \cdot B_0$ | (13) $A_0 \cdot \overline{B_0}$ | (14) $\overline{A_0} \cdot \overline{B_0}$ |
| (15) $A_1 \cdot B_1$ | (16) $\overline{A_1} \cdot B_1$ | (17) $A_1 \cdot \overline{B_1}$ | (18) $\overline{A_1} \cdot \overline{B_1}$ |
| (19) $A_k \cdot B_k$ | (20) $\overline{A_k} \cdot B_k$ | (21) $A_k \cdot \overline{B_k}$ | (22) $\overline{A_k} \cdot \overline{B_k}$ |
| (23) $A_{k+1} \cdot B_{k+1}$ | (24) $\overline{A_{k+1}} \cdot B_{k+1}$ | (25) $A_{k+1} \cdot \overline{B_{k+1}}$ | (26) $\overline{A_{k+1}} \cdot \overline{B_{k+1}}$ |
| (27) $A_0 + B_0$ | (28) $\overline{A_0} + B_0$ | (29) $A_0 + \overline{B_0}$ | (30) $\overline{A_0} + \overline{B_0}$ |
| (31) $A_k + B_k$ | (32) $\overline{A_k} + B_k$ | (33) $A_k + \overline{B_k}$ | (34) $\overline{A_k} + \overline{B_k}$ |
| (35) $A_{k+1} + B_{k+1}$ | (36) $\overline{A_{k+1}} + B_{k+1}$ | (37) $A_{k+1} + \overline{B_{k+1}}$ | (38) $\overline{A_{k+1}} + \overline{B_{k+1}}$ |