

析別論理演算

ビットパターンXとFFの論理演算

X=10110110について考える。

- ① 論理和はXのすべてのビットが1となる。
- ② 論理積はXのビットは変化しない。
- ③ 排他的論理和はXのすべてのビットは反転する。

10110110	10110110	10110110
<u>OR11111111</u>	<u>AND11111111</u>	<u>EOR11111111</u>
11111111	10110110	01001001

ビットパターンXと00の論理演算

X=10110110について考える。

- ① 論理和はXのビットは変化しない。
- ② 論理積はXのすべてのビットが0になる。
- ③ 排他的論理和はXのビットは変化しない。

10110110	10110110	10110110
<u>OR00000000</u>	<u>AND00000000</u>	<u>EOR00000000</u>
10110110	00000000	10110110

ビットパターンX同士の論理演算

X=10110110について考える。

- ① 論理和はXのビットは変化しない。
- ② 論理積はXのビットは変化しない。
- ③ 排他的論理和はXのすべてのビットが0になる。

10110110	10110110	10110110
<u>OR10110110</u>	<u>AND10110110</u>	<u>EOR10110110</u>
10110110	10110110	00000000

ビットパターンと80の論理演算

$X=10110110$ について考える。

- ① 論理和は、MSB、 X のビットは変化しない。
- ② 論理積は、MSBのビットは変化せず、他のビットは0になる。
- ③ 排他的論理和は、
MSBのビットは反転し、他のビットは変化しない。

10110110	10110110	10110110
<u>OR10000000</u>	<u>AND10000000</u>	<u>EOR10000000</u>
10110110	10000000	00110110

マスクングとは

- ① マスクングは、マスクを使って、
あるビットパターンの一部を抽出する方法である。
- ② 対象となるビットパターンとマスクとの間で、
ビット単位の論理演算を行う。
- ③ 8ビットの任意のビットパターンと
 $(03)_{16}$ の論理積を求めると、
任意のビットパターンの
下位2ビットのビットパターンを抽出できる。

マスキングの具体例

- ① ビットパターン10110011の
最下位の2ビットを抽出する。

$$\begin{array}{r} 10110011 \\ \underline{AND 00000011} \\ 00000011 \end{array}$$

- ② マスクとして、
上位6ビットはすべて0、下位2ビットは11を使用する。
- ③ 桁別論理積の演算結果として、下位2ビットが抽出さ

ビットパターンの分割法

ビットパターン10110100から
順次2ビットずつ抽出して、
8ビットのビットパターンを4個形成する場合の
手続きは次の手順で行う。

- ① 10110100と00000011の論理積を求める。

$$10110100 \wedge 00000011 = \underline{00000000}$$

00000000を得る。

② 10110100を右に2ビットシフトする。

10110100→00101101

00101101を得る。

③ 00101101と00000011の論理積を求める。

00101101 \wedge 00000011 = 00000001

00000001を得る。

④ 00101101を右に2ビットシフトする。

00101101 → 00001011

00001011を得る。

⑤ 00001011と00000011の論理積を求める。

00001011 \wedge 00000011 = 00000011

00000010を得る。

⑥ 00001011を右に2ビットシフトする。

00001011 → 00000010

00000010を得る。

⑦ 00000010と00000011の論理積を求める。

00000010 \wedge 00000011 = 00000010

00000010を得る。

⑧ 次の4個のビットパターンを得る。

① 00000000

② 00000001

③ 00000011

④ 00000010

ビットパターンの合成法

4個のビットパターンから
最下位の2ビットを抽出して、
8ビットのビットパターン1個を
合成する手順は次の通りである。

① 00000010を2ビット左にシフトする。

00000010→00001000

00001000を得る。

② 00001000と00000011の論理和を求める。

$$00001000 + 00000011 = 0000\underline{1011}$$

00001011を得る。

③ 00001011を2ビット左にシフトする。

$$00001011 \rightarrow 00101100$$

00101100を得る。

④ 00101100と00000001の論理和を求める。

$$00101100 + 00000001 = 00\underline{101101}$$

00101101を得る。

⑤ 00101101を2ビット左にシフトする。

$$00101101 \rightarrow 10110100$$

10110100を得る。

⑥ 10110100と00000000の論理和を求める。

$$10110100 + 00000000 = \underline{10110100}$$

10110100を得る。

⑦ 合成結果は、10110100となる。