

・8ビット列の下位4ビットが変化しない操作はどれか

ア. 16進表記0Fのビット列との排他的論理和をとる

イ. 16進表記0Fのビット列との否定論理積をとる。 解法: 適当な8ビット列と16進表記0Fを2進数に変換したものをそれぞれ論理演算する。

ウ. 16進表記0Fのビット列との論理積をとる。

エ. 16進表記0Fのビット列との論理和をとる。

適当な8ビット列として「10101010」を使う 《なんでもお水

(16進数を2進数に変換)

0F → (0)₁₆ (15)₁₆ → 10進数で「15」これを2進数に変換する 2) 15 ... 1
 $\times 16$ (1111)₂なので、8ビット列では「00001111」 2) 7 ... 1
 $0 + 15$ となる。 2) 3 ... 1
1

変換が終了したら、適当に作った8ビット列と変換した2進数で「論理演算」

ア. 排他的論理和 (XOR)

10101010 ← 適当に作ったビット列

XOR 00001111 ← 16進数 → 2進数変換したもの。

10100101

↑ 1行ずつ論理演算

問題は下位4ビットが、16進数を2進数にしたものと同一である必要があるので、

アは ×

イ. 否定論理積 (NAND)

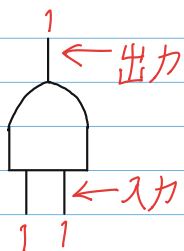
10101010
NAND 00001111
11110101 ← 問題と異なる。

ウ. 論理積 (AND)

10101010
AND 00001111
00001010 ← 問題と同じ下位4ビットが同一なので!
答え: ウ

論理回路

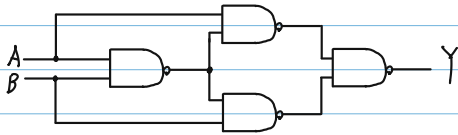
論理演算を行う回路のこと。



	MIL記号	
論理積 AND		半円に2つの直角。
論理和 OR		3つの角がすべて尖っている。
否定 NOT		三角形と丸。記号の先端に付いている小さな丸は否定を表す。
否定論理積 NAND		〃
否定論理和 NOR		〃
排他的論理和 XOR		〃

論理回路を解くコツ

例: 図の論理回路と等価であるものをA~Eで選べ



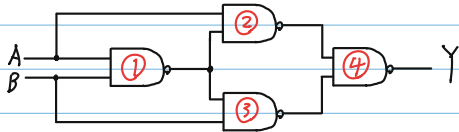
P. $A \oplus B \rightarrow Y$

I. $A \oplus B \rightarrow Y$

ウ. $A \oplus B \rightarrow Y$

E. $A \oplus B \rightarrow Y$

このような問題を解くときは演算結果のベン図を書く。



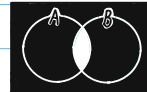
A =

B =

色付きのところが1、白のところが0

①の演算結果

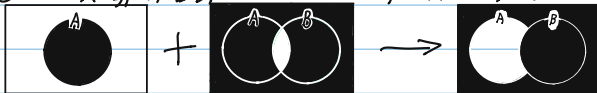
AとBの入力を否定論理積 (NAND) するので



となる。

②の演算結果

Aと①の演算結果を合せて NAND するので

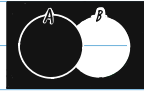


← NANDは1と1で0を出力するので
この結果となる。

③の演算結果

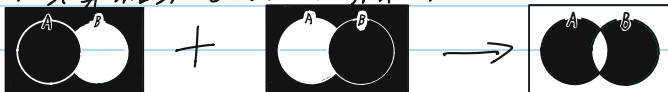
②と同じ要領で解く

②では入力かB



④の演算結果

②の演算結果と③の演算結果を足すと...?



となる。

この結果は

A	B	
0	0	0
0	1	1
1	0	1
1	1	0

である。これは XOR であるので、答えは ウ。

加算回路

論理回路を組み合わせて足し算をする回路を加算回路と呼ぶ
加算回路には半加算回路と全加算回路がある。

半加算回路

2進数の足し算において、下の位からの桁上りを考慮しない加算回路
桁がくり上がったら、それ以上計算できない。←入力が2つしかないため。

半加算回路での計算パターン

0	1	0	1
+ 0	+ 0	+ 1	+ 1
0	1	1	1 0

2桁目に入れたので、これ以上計算できない。

この式にX、Y、C、Sを代入して、真理値表を作ると

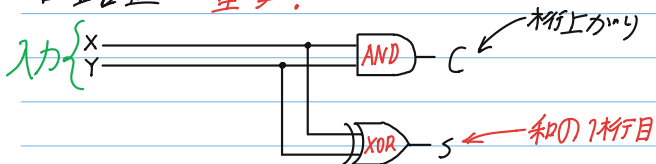
X + Y C S	入力		出力	
	X	Y	C	S
	0	0	0	0
	0	1	0	1
	1	0	0	1
	1	1	1	0

となる

このときのXとYの入力におけるCとSの結果を見ると

CはAND (論理積) の結果 SはXOR (排他的論理和) となっている

回路図 重要!



全加算回路

半加算回路に更に半加算回路を加えたもので、つまり、半加算では
入力が2つだったのが、全加算回路では3つになる

全加算回路の計算パターン

0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	1
+ 0	+ 1	+ 0	+ 0	+ 1	+ 1	+ 0	+ 1
0	1	1	1	1 0	1 0	1 0	1 1

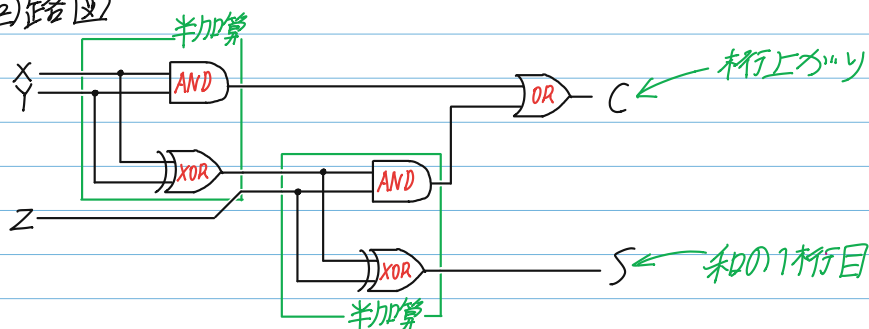
入力3つなので、2通りある。

この式を基に真理値表を作ると、

X Y + Z C S	入力			出力	
	X	Y	Z	C	S
	0	0	0	0	0
	0	0	1	0	1
	0	1	0	0	1
	1	0	0	0	1
	0	1	1	1	0
	1	0	1	1	0
	1	1	0	1	0
	1	1	1	1	1

この真理値表から
2つの半加算回路と1つの論理和
(OR回路)でできてはいることがわかる。

回路図



半加算と全加算の使い分け

半加算回路: ビット列の足し算では、1桁目より下には位はなく、桁上りカッリを考えなくていい

全加算回路: 2桁目以上の位では、下の位からの桁上りカッリを考える必要がある。

論理回路の 其他用語

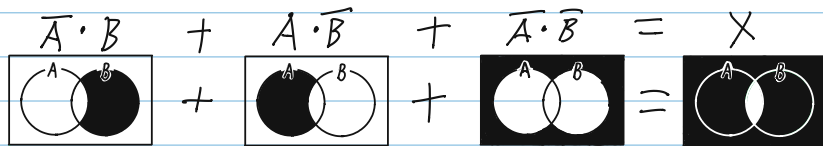
素子: 回路を構成する電子部品、論理演算の回路を構成する素子は論理素子
↳ 別表記でゲートと書かれる場合もある

・論理演算回路

例: 論理式 $X = \bar{A} \cdot B + A \cdot \bar{B} + \bar{A} \cdot \bar{B}$ と同じ結果が得れる論理回路を求めよ

このときの論理積は・論理和は+、 $\bar{A}$ や $\bar{B}$ は否定を指す

計算の優先順位に則り、2解いていく。



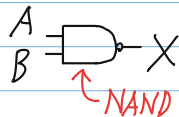
これらを足して

共通していない部分だけが残り
他は土曜ている

計算結果は共通している部分だけでなく、それ以外は1の状態

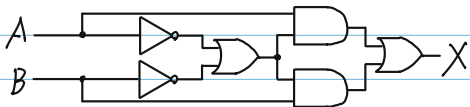
つまり、否定論理積と同じ結果である!

なので求める論理回路は

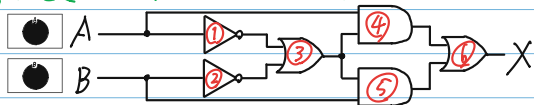


図に示すデジタル回路と等価な論理式はどれか:

↑でどのような式を書けばXと等しくなるのか



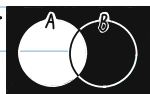
解くときは必ず「ベン図」を描く。



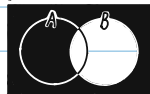
①、②、③の流れ

まず、A、Bの信号は、④、⑤素子に分歧し①、②の素子に入る。

①と②はNOTなので、反転云

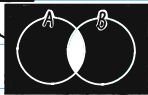


②の信号は



となる。

③はOR回路なので、信号を合わせて



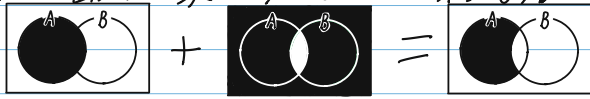
となる

(XOR、NAND、NORは7と7で0になる)

← NANDと等しい

④の流れ

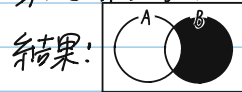
③での出力結果でAの出力を素子の記号からANDさせる。



≪ ANDは1と1のみ1を出力、1と0、0と1、0と0では0となる。

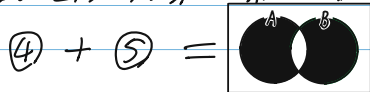
⑤の流れ

④も⑤と同様に3の出力結果を素子の記号からANDさせる。



⑥の流れ

④と⑤の出力結果も素子の記号からORする。



この一連の流れから、Xは $A \cdot B$ と $\bar{A} \cdot B$ を足したもので等しいので、

$$X = A \cdot B + \bar{A} \cdot B \text{ となる。}$$