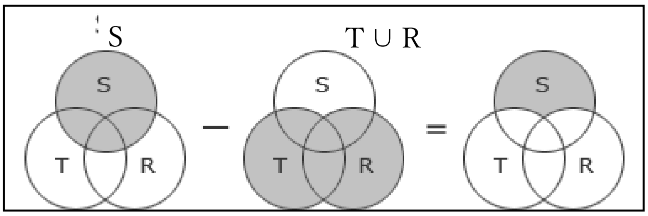
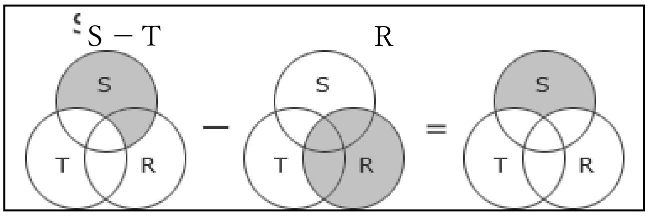
Ｃｈａｐｔｅｒ３　コンピュータの回路を知る

３－１　論理回路とベン図〔解答・解説〕

問１　ア
〔解説〕まず問題文の集合Ｓ－(Ｔ∪Ｒ)は次のような集合になる。



正解のアの集合は



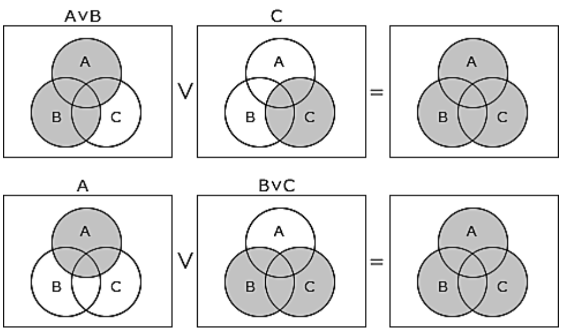
問題文中の集合と一致する。

問２　エ
〔解説〕論理演算の演算則の一つに「結合の法則」がある。

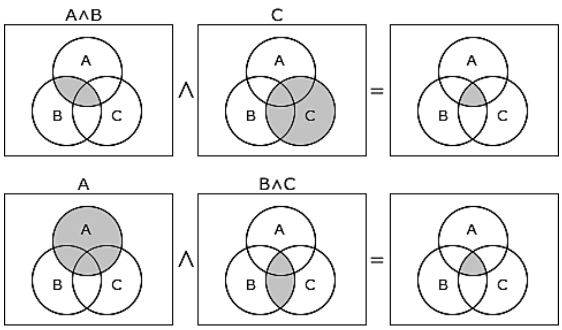
(Ａ・Ｂ)・Ｃ＝Ａ・(Ｂ・Ｃ)
(Ａ＋Ｂ)＋Ｃ＝Ａ＋(Ｂ＋Ｃ)

排他的論理和(⊕)も論理和演算の一種なので、論理和の場合と同様に結合の法則が成立するため、3つの演算記号のすべての場合で結合の法則が成立することになる。
論理演算は集合演算と同様の性質を持っているので、上記の「結合の法則」をベン図を用いて表すと次のようになる。

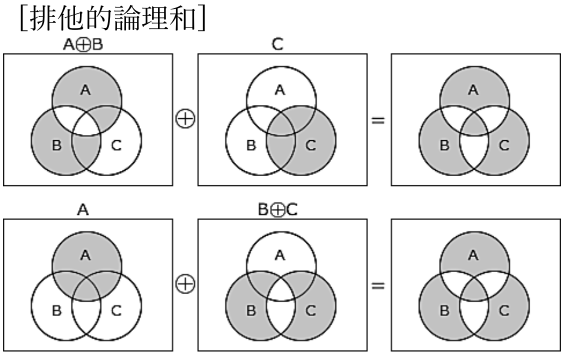
[論理和]



[論理積]



[排他的論理和]

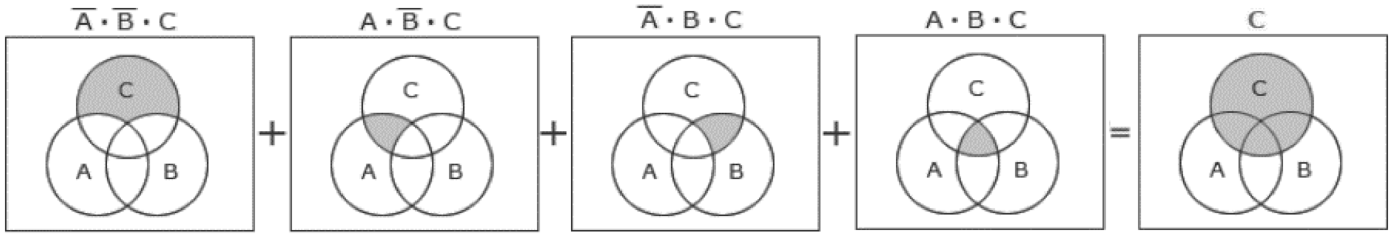


問３　ウ
〔解説〕２つのビット列を　１００１　として

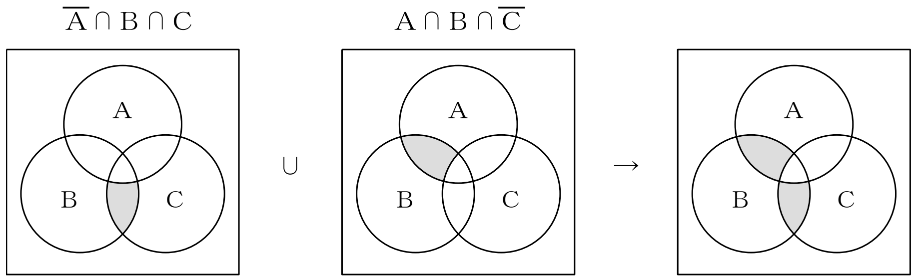
- a　AND　b ……　１００１ AND １００１ ＝ １００１
- a　OR　 b ……　１００１ OR　１００１ ＝ １００１
- a　XOR　b ……　１００１ XOR １００１ ＝ ００００

よって、ウが正解である。

問４　エ
〔解説〕各式をベン図化して考えてみる。



問５　ウ
〔解説〕 $(\overline{A}\cap B\cap C)\cup(A\cap B\cap\overline{C})$をベン図で表してみると下図のようになるので，正解はウである。



問６　イ
〔解説〕否定論理積(ＮＡＮＤ)は、２つの入力がともに１の場合にだけ結果が０、その他の場合は１となる演算である。
まずX＝０，Ｙ＝０のときに演算結果が０になるかを検証
［ア］　((０ ＮＡＮＤ ０)ＮＡＮＤ ０)ＮＡＮＤ ０
＝(１ ＮＡＮＤ ０) ＮＡＮＤ ０
＝１ ＮＡＮＤ ０
＝１（結果が０ではないので誤り）

［イ］　(０ ＮＡＮＤ ０)ＮＡＮＤ(０ ＮＡＮＤ ０)
＝１ ＮＡＮＤ １
＝０（結果が０なので正しい可能性がある）

［ウ］　(０ ＮＡＮＤ ０)ＮＡＮＤ(０ ＮＡＮＤ ０)
＝１ ＮＡＮＤ １
＝０（結果が０なので正しい可能性がある）

［エ］　０ ＮＡＮＤ(０ ＮＡＮＤ(０ ＮＡＮＤ ０))
＝０ ＮＡＮＤ（０ ＮＡＮＤ １）
＝０ ＮＡＮＤ １
＝１（結果が０ではないので誤り）

次に正しい可能性のある「イ」と「ウ」について、X＝１，Ｙ＝０の時に演算結果が１になるか検証
［イ］　（１ ＮＡＮＤ １）ＮＡＮＤ（０ ＮＡＮＤ ０）
＝０ ＮＡＮＤ １
＝１

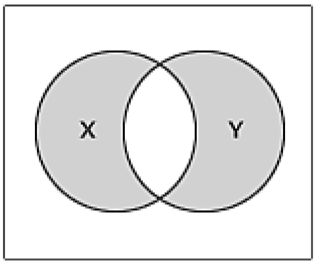
［ウ］　（１ ＮＡＮＤ ０）ＮＡＮＤ（１ ＮＡＮＤ ０）
＝１ ＮＡＮＤ １
＝０　（結果が１ではないので誤り）

残った「イ」が答えとして適切。

問 7　ア
〔解説〕相補演算とは、集合演算によって得られる結果が互いにもう一方の演算の補集合となっている関係、すなわち X AND Y と NOT (X AND Y)のような関係になっているものを指す。
排他的論理和(XOR)は、2 つの入力値が異なれば真、同じであれば偽を返す論理演算で、

排他的論理和の真理値表とベン図

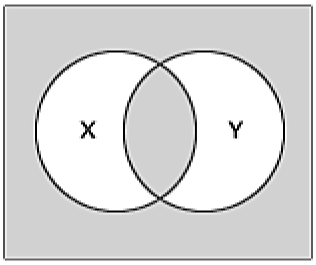
X	Y	結果
0	0	0
0	1	1
1	0	1
1	1	0



排他的論理和の相補演算になるのは、XOR の補集合(XOR のベン図の白い部分)が結果として得られる演算なので、答えとして適切なのは「等価演算」ということになる。

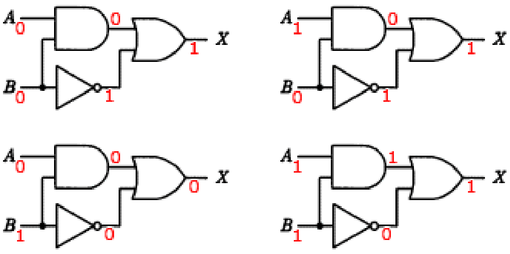
等価演算の真理値表とベン図

X	Y	結果
0	0	1
0	1	0
1	0	0
1	1	1



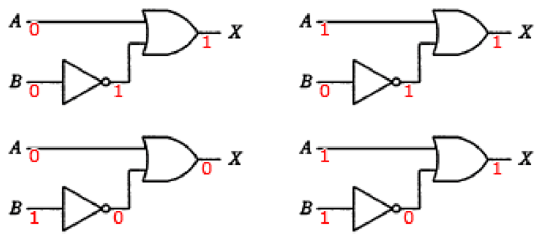
3－2　論理回路と基本回路〔解答・解説〕

問 1　イ
〔解説〕設問の論理回路に(A=0, B=0), (A=1, B=0), (A=0, B=1), (A=1, B=1)の 4 つの値を入力すると X には左の値が出力される。右は正解に入力した場合。



入力A	入力B	結果X
0	0	1
1	0	1
0	1	0
1	1	1

イ



入力A	入力B	結果X
0	0	1
1	0	1
0	1	0
1	1	1

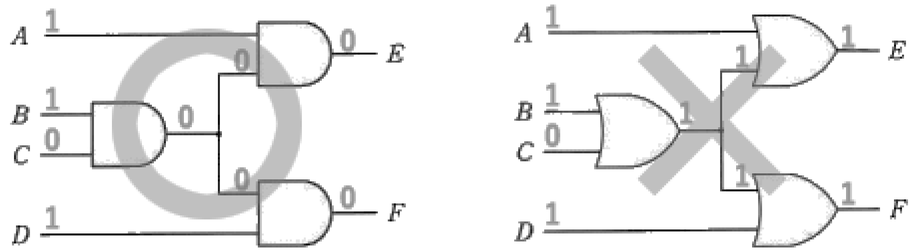
問 2　ウ
〔解説〕X＝0, Y＝0 のとき, X OR (X□Y) ＝1 より, X□Y＝1
X＝0, Y＝1 のとき, X OR (X□Y) ＝1 より, X□Y＝1
X＝1, Y＝0 のとき, X AND (X□Y) ＝0 より, X□Y＝0
X＝1, Y＝1 のとき, X AND (X□Y) ＝1 より, X□Y＝1

問 3　エ

〔解説〕奇数パリティは、データを構成するビット全体の中でビット「1」の数が奇数個になるようにパリティビットを付加する方式。パリティビットを含めた各ビットをすべてXOR演算した結果は必ず「1」となる。

問 4　ウ

〔解説〕[ウを確認]



3－3　基本回路を組み合わせた論理回路〔解答・解説〕

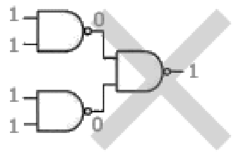
問１　イ

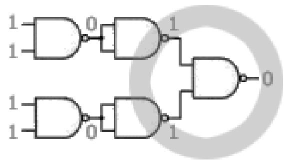
〔解説〕４入力ＮＡＮＤを構成することになっているので、４つの入力のいずれかが"０"であるときに"１"を出力する、または全ての入力が"１"のときに"０"を出力する論理回路になっていればよい。

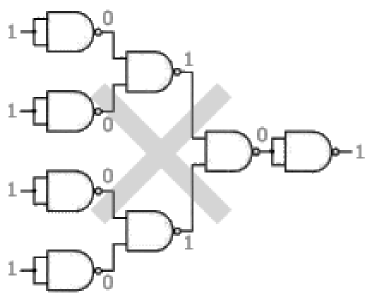
入力1	入力2	入力3	入力4	出力
0	0	0	0	1
0	1	0	0	1
0	0	1	0	1
0	0	0	1	1
0	1	1	0	1
0	1	0	1	1
0	0	1	1	1
0	1	1	1	1
1	0	0	0	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	1
1	1	1	0	1
1	1	0	1	1
1	0	1	1	1
1	1	1	1	0

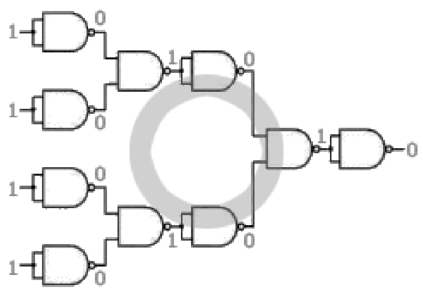
図　4入力NAND回路の真理値表

４つの入力に"１"を設定したときに"０"を出力するか否かによって正しい論理回路図を選別する。



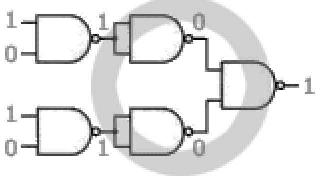


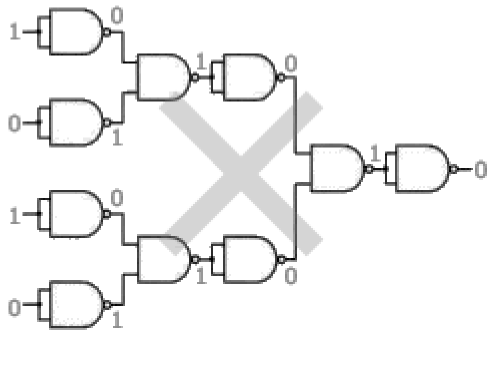




この時点で「ア」と「ウ」は不適切であると判断できる。残った「イ」と「エ」について検証を続け

ここでは４つの入力に"１０１０"を与えた場合の出力を見る。

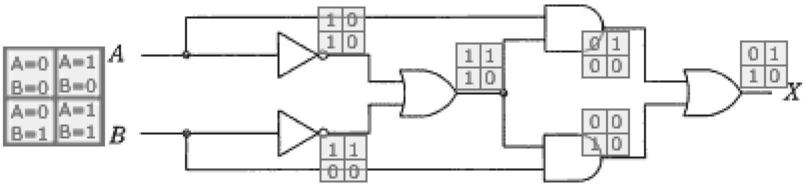




消去法により「イ」が４入力ＮＡＮＤ回路と判断できる。なお「エ」は４入力ＮＯＲ回路。

問２　ウ

〔解説〕まず、問題の回路図を真理値表にする。

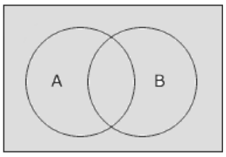


論理回路図の真理値表

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

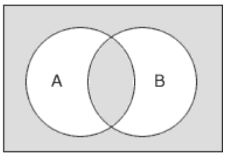
ア

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



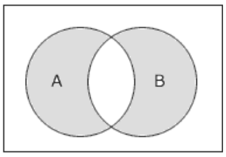
イ

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



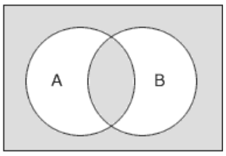
ウ

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



エ

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



問３　ア

〔解説〕入力Ａ、Ｂと出力Ｘ、Ｙの適切な関係は次のようになる。

A	B	X(桁上がり)	Y(和の1桁目)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

表を見ると、２つの入力とＸの関係はＡＮＤの真理値表と一致し、２つの入力とＹの関係はＸＯＲの真理値表と一致している。したがって、Ｘ＝Ａ　ＡＮＤ　Ｂ、Ｙ＝Ａ　ＸＯＲ　Ｂ　となっているアの論理回路図が正解。

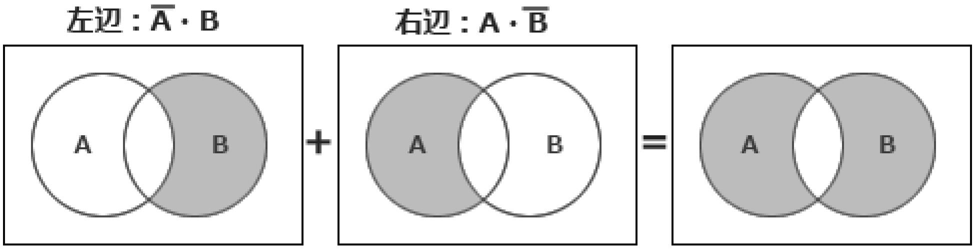
問４　イ
〔解説〕ＡとＢの論理和(ＯＲ)が、次の論理積(ＡＮＤ)への入力。Ｃの否定(ＮＯＴ)が、次の論理積(ＡＮＤ)への入力。したがって、正解は「イ」。

問５　イ
〔解説〕問題及びア，イ，ウ，エの論理回路のＡ，Ｂに（０，０）（１，１）（１，０）（０，１）を入力した場合のＸは次のようになる。

A	B	問題	ア	イ	ウ	エ
0	0	1	1	1	0	1
0	1	0	1	0	0	1
1	0	1	1	1	1	0
1	1	1	0	1	1	1

よって，正解はイである。

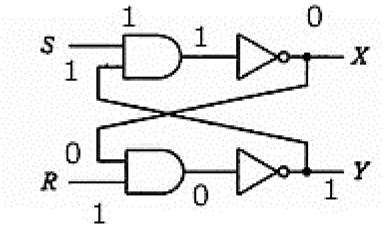
問６　イ
〔解説〕回路図を論理式で表すと「$\overline{\mathrm{A}}$・Ｂ＋Ａ・$\overline{\mathrm{B}}$」。この集合をベン図上に表すと次にようになる。



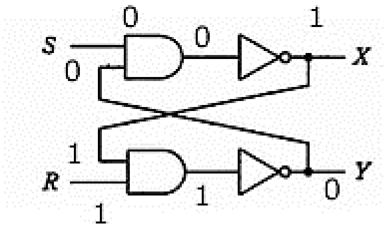
この集合はXOR演算の集合と等しいため「イ」が正解と判断できる。

問 7　ウ

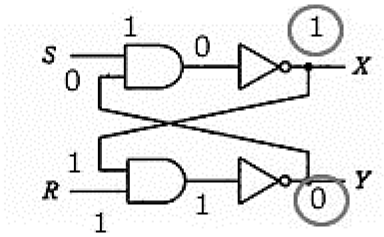
〔解説〕1：初期の状態は次のようになる。



2：S をいったん 0 にすると次のようになる。



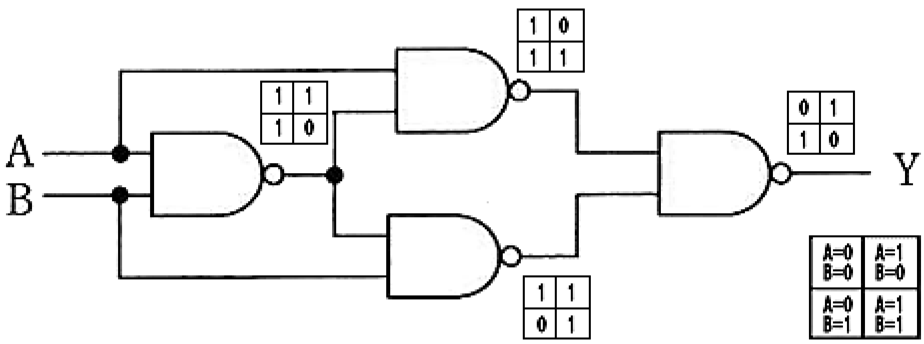
3：S を再び 1 に戻すと次のようになる。



よってウが正解である。

問 8　ウ

〔解説〕それぞれ A と B が 0 又は 1 の値をとった時に、各回路の演算結果である Y の値がどうなるかを考える。(設問の回路記号は NAND 回路)



入力A	入力B	結果Y
0	0	0
1	0	1
0	1	1
1	1	0

結果の真理値表は、XOR 回路の真理値表と一致する。よって正解は「ウ」。

3－4　半加算器と全加算器〔解答・解説〕

問１　エ
〔解説〕ＡＮＤ回路とＸＯＲ回路を一つずつ使って構成するのは，半加算器である。

問２　ア
〔解説〕真理値表より，ＣはＡＮＤ演算の結果，ＳはＸＯＲ演算の結果となっていることがわかる。

問３　ウ
〔解説〕全加算器は、2 進数の加算を行う論理回路で、2 進数の一桁分の加算を下位からの繰り上がりを含めて計算できる。図中の出力 c は Carry out で繰り上がり数、s は Sum で合計を表している。

　考え方としては、入力された３つの値を足して、c と s の出力を求める。この問題の場合、x と z が 1 なので足して 2、2 進数なので１つ上の桁に繰り上がるので c が 1 になり、s は 2－2＝0 で 0 になる。

3－5　ビット操作とマスクパターン〔解答・解説〕

問１　ウ
〔解説〕２進数　１１１１１１００　とＸＯＲ演算を行えばよい。

問２　ア
〔解説〕１６の倍数は１６で割り切れる数であるから１５以下（下位４ビット）は０となる。
（例）１６　→　０００１　００００
　　　３２　→　００１０　００００
　　　４８　→　００１１　００００

問３　ウ
〔解説〕

ア
```
     1010 1010
XOR) 0000 0000
---------------
     1010 1010
```
元のビット列と同じものになる

イ
```
    1010 1010
OR) 0000 0000
--------------
    1010 1010
```
元のビット列と同じものになる

ウ
```
     1010 1010
XOR) 1111 1111
---------------
     0101 0101
```
全ビットを反転したビット列になる

エ
```
    1010 1010
OR) 1111 1111
--------------
    1111 1111
```
全てが１のビット列になり

問４　イ
〔解説〕ａ：ｎの下位４ビットを取り出し，ｘに格納する
　　　　ｂ：次の４ビットを取り出すため，ｎを右に４ビット論理シフトする

問 5　ウ
〔解説〕８ビットのデータの下位７ビットを取り出したいので、最上位ビット以外を「１」としたビット列「０１１１　１１１１」との論理積をとることで下位７ビットを取り出せる。
２進数「０１１１　１１１１」を１６進数で表すと「７Ｆ」なので正解は「ウ」とる。