3

学習のツボ

～ 合格のための学習をしよう ～

試験に合格するためには、いわゆる学問として一通り学習するだけでは駄目です。出題の傾向や特徴を知っておかなければなりません。

最低限知っておくべき事項や、合格のための学習のツボを押さえましょう。

合格のための学習をしよう！

前章で、いかに多くの過去問題が流用されているのかを示しました。本章では、**必須の知識**と、**学習を進める上で知っておくべきツボ**を紹介します。

世の中に出回っている基本情報技術者試験用のテキストには、受験学習の**役に立たない**のではないかと思えるものがあります。その点を明らかにして、合格のための学習について考えてみましょう。

・問題あるテキスト － どこが出題されたのかが分からない！

どの分野や項目が過去に出題された実績があるのか、今後出題される可能性があるのか、といったことをまったく示さないテキストや、過去に一度も出題されていない分野や項目に多くのページをさいているテキストがあります。

限られた時間の学習で合格を目指すには、試験に出題された実績がなく、これから出題される可能性も低い分野や項目に時間をかけてはいけません。

出題されやすい分野・項目に重点をおいて学習しましょう。

・問題あるテキスト － 試験で使われない用語で解説されている！

過去に出題された問題文中で何度も利用されている専門用語とは、まったく異なる用語を用いて解説するテキストがあります。たとえば、最初から最後まで“バイナリツリー”という語句で解説するテキスト。ちなみに、平成６年度秋期以降に 10 回以上出題されていますが、すべて“バイナリツリー”ではなく“２分木”です。

読者のためではなく、著者の自己満足のために、試験範囲を独りよがりに解説するテキストを用いて学習するのでは、合格から遠ざかりかねません。

試験問題で使われる語句を優先して覚えるべきです。

・問題あるテキスト － どこまで深く学習すればよいのか分からない！

テキストに次のような解説があるとします。

> FTP（file transfer protocol）は、インターネットにおいてファイル転送に用いられるプロトコル（通信規約）である。

さて、FTP という語句を覚えればいいのでしょうか、それとも FTP が file transfer protocol の略であることまでを覚える必要があるのでしょうか？

この疑問に対する答えは、過去問題を調べれば得られますし、過去に出題されていない分野や項目や用語に関しても、ある程度は類推できます。

だからといって、みなさん一人一人が、過去問題を調べ尽くす努力をする必要はありません。なぜならば、**徹底的な過去問題の調査結果をもとにして、本書と付属ディスクが執筆されている**のですから。

DISK

本章では、必須の知識と学習のツボを解説するとともに、過去に出題された問題を具体例として紹介します。付属ディスクには、それらの一覧表も用意されています。問題番号をクリックするだけで閲覧できます。

10の整数乗を表す接頭語を覚えよう

キロ（k）、メガ（M）、マイクロ（μ）など、10 の整数乗倍を表す接頭語は、常識かつ必須の知識です。以下の表に示すものは**すべて覚える**必要があります。

単位	記号	値	日本語
テラ	T	10^{12}	一兆
ギガ	G	10^{9}	十億
メガ	M	10^{6}	百万
キロ	k	10^{3}	千
ミリ	m	10^{-3}	千分の一
マイクロ	μ	10^{-6}	百万分の一
ナノ	n	10^{-9}	十億分の一
ピコ	p	10^{-12}	一兆分の一

平成 10 年度までは、この知識をそのまま問う問題が出題されていました。

■ 平成10年度（1998年度）秋期 午前 問4

時間の単位として、ナノ（n）秒、ピコ（p）秒、マイクロ（μ）秒、ミリ（m）秒などが用いられる。この n，p，μ，m は、10 の整数乗倍を表す接頭語である。これらを小さい順に並べたものはどれか。

ア　n，p，μ，m
イ　n，μ，p，m
ウ　p，n，μ，m
エ　p，μ，n，m

■ 平成7年度（1995年度）春期 午前 問5

1 ピコ秒に等しい値はどれか。

ア　1 ナノ秒の 1,000 倍
イ　1 マイクロ秒の 100 万分の 1
ウ　2 のマイナス 12 乗秒
エ　10 のマイナス 10 乗秒
オ　10 のマイナス 11 乗秒

■ 平成6年度（1994年度）秋期 午前 問12

10 の整数乗倍を表す接頭語の記号 G（ギガ）、k（キロ）、M（メガ）、T（テラ）の四つについて、その大小関係を正しく表しているものはどれか。

ア　G < k < M < T
イ　k < G < T < M
ウ　k < M < G < T
エ　M < G < k < T
オ　M < T < G < k

平成 11 年度以降は、このような問題は出題されていませんが、10 の整数乗倍を表す接頭語の知識なしには解けない問題が数多く出題されています。

ここでは、その一例をあげます。

■ 平成13年度(2001年度)春期 午前 問20

50 MIPS の処理装置がある。この処理装置の平均命令実行時間は幾らか。

ア　20 ナノ秒　　イ　50 ナノ秒
ウ　2 マイクロ秒　　エ　5 マイクロ秒

▷ MIPS は頻出分野ですから、第４章（p.124）で詳しく解説します。

■ 平成8年度(1996年度)春期 午前 問20

毎秒 300k バイトのデータを伝送効率 30％の LAN を使用して送る。この LAN に必要な伝送速度は、最低何Mビット／秒か。

ア　1.0　　イ　2.4　　ウ　8.0　　エ　10　　オ　24

▷伝送効率（通信回線の性能）は頻出分野ですから、第４章（p.164）で詳しく解説します。

■ 平成14年度(2002年度)秋期 午前 問28

画面の大きさが横 640 ドット、縦 480 ドットで、256 色が同時に表示できるパソコンのモニタの画面全体を使って、30 フレーム／秒のカラー動画を再生表示させる。このとき、1 分間に表示される画像データの量（バイト）として、最も近いものはどれか。ここで、データは圧縮しないものとする。

ア　300 k　　イ　1 M　　ウ　550 M　　エ　133 G

▷画像データの容量は頻出分野ですから、第４章（p.142）で詳しく解説します。

■ 平成13年度(2001年度)春期 午前 問55

A 社の受注システムのサーバでは、120G バイトのハードディスクを使っている。このハードディスクの 10％を占めるデータベースを毎週バックアップする場合、バックアップの媒体として、最も適切なものはどれか。ここで、バックアップソフトは、圧縮率 50％でバックアップできるものとする。

ア　3.5 インチ MO　　イ　CD-R
ウ　DAT　　エ　ZIP

ビットと基数を完全に理解しておこう

コンピュータ内部では文字や数値データはビットで表現され、その解釈には２進数や16進数が利用されます。ビットや基数について完璧に理解しておきましょう。

■ コンピュータは2進数

ビット（*bit* = *binary digit*）は、コンピュータ内部で表現できる情報の最小単位です。ON なのか／ OFF なのか、あるのか／ないのか、といった二つの状態のどちらであるかを、２進数の１けたである０と１とで表します。

$\boxed{0}$　$\boxed{1}$

ビット

なお、一般に８ビットのことを１**バイト**（*byte*）といいます。

■ 私たちの日常生活は10進数

私たちの日常生活では、普通は 10 進数を用いて数値を表現します。自分の生年月日を２進数や 16 進数で即座に言える人は、あまりいないでしょう。

■ 基数

10 進数は 10 を**基数**とする数であり、２進数は２を基数とする数です。**基数**について、まずは 10 進数を例に考えていきます。

・10 進数

10 進数では、以下に示す 10 種類の数字を利用して数を表現します。

$\boxed{0}$　$\boxed{1}$　$\boxed{2}$　$\boxed{3}$　$\boxed{4}$　$\boxed{5}$　$\boxed{6}$　$\boxed{7}$　$\boxed{8}$　$\boxed{9}$

これらを使い切ったら、けたが繰り上がって 10 となります。２けたの数は、10 から始まって 99 までです。その次は、さらに繰り上がった 100 です。

このことから、次のことが分かります。

１けた	…	０から９まで 10 種類の数を表す。	$\boxed{0}$ ～ $\boxed{9}$
～２けた	…	０から 99 まで 100 種類の数を表す。	$\boxed{0}\boxed{0}$ ～ $\boxed{9}\boxed{9}$
～３けた	…	０から 999 まで 100 種類の数を表す。	$\boxed{0}\boxed{0}\boxed{0}$ ～ $\boxed{9}\boxed{9}\boxed{9}$

10 進数の各けたは、下のけたから順に 10^0, 10^1, 10^2, … と、10 のべき乗の重みをもちます。したがって、たとえば 1234 は、次のように解釈できます。

$$1234 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

※ 10^0 は１です（2^0 でも 8^0 でも、とにかく０乗の値は１です）。

10 進数の性質をまとめましょう：

> 10 種類の数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 を用いて数を表現する。
> d けたあれば０から $10^d - 1$ までの 10^d 種類の数を表せる。
> 下から第 n けた目は 10^{n-1} の重みをもつ。

・2 進数

2 進数では、以下に示す 2 種類の数字を利用して数を表現します。

$\boxed{0}$　$\boxed{1}$

これらを使い終わると、けたが繰り上がって 10 となり、さらにその次の数は 11 となります。これで 2 けたを使い切りますので、その次は 100 です。

このことから、次のことが分かります。

1 けた	…	0 から 1 まで 2 種類の数を表す。	$\boxed{0}$ 〜 $\boxed{1}$
〜2 けた	…	0 から 11 まで 4 種類の数を表す。	$\boxed{0\,0}$ 〜 $\boxed{1\,1}$
〜3 けた	…	0 から 111 まで 8 種類の数を表す。	$\boxed{0\,0\,0}$ 〜 $\boxed{1\,1\,1}$

2 進数の各けたは、下のけたから順に $2^0, 2^1, 2^2, \cdots$ と、2 のべき乗の重みをもちます。したがって、たとえば 1010 は、次のように解釈できます。

$$1010 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

2 進数の性質をまとめましょう：

> 2 種類の数字 0, 1 を用いて数を表現する。
> d けたあれば 0 から $2^d - 1$ までの 2^d 種類の数を表せる。
> 下から第 n けた目は 2^{n-1} の重みをもつ。

・r 進数

10 進数と 2 進数について考えてきましたので、基数に関する一般的な性質が分かったと思います。一般に、r 進数の性質は次のようになります。

> r 種類の数字を用いて数を表現する。
> d けたあれば 0 から $r^d - 1$ までの r^d 種類の数を表せる。
> 下から第 n けた目は r^{n-1} の重みをもつ。

■ 基数の表記

ただの 10 では、何進数の 10 なのか分かりません。そこで、値を ( ) で囲んで、その後に基数を下付き文字で付ける表記法を使うことがあります。たとえば、10 進数の 10 は $(10)_{10}$、2 進数の 10 は $(10)_2$ と表記します。

なお、基数が 10 を超える場合、10 種類以上の数字が必要となりますので、9 に続く数字としてアルファベットを A から順に使います。

たとえば 16 進数では、次に示す 16 種類の数字を利用します。先頭から順に、10 進数の 0 〜 15 に対応します（表記は小文字でも構いません）。

$\boxed{0}$ $\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{4}$ $\boxed{5}$ $\boxed{6}$ $\boxed{7}$ $\boxed{8}$ $\boxed{9}$ $\boxed{A}$ $\boxed{B}$ $\boxed{C}$ $\boxed{D}$ $\boxed{E}$ $\boxed{F}$

基数変換できるようになっておこう

コンピュータの内部で使われる２進数は、比較的小さな数でも、けた数が膨れ上がります。たとえば、10 進数で４けたの 1963 は、２進数では 11110101011 であり、実に 11 けたとなります。そのため、私たち人間がコンピュータに指示をするプログラムなどでは、２進数と親和性の高い 16 進数や８進数をよく用います。

そのため、ある基数で表現された数値を、異なる基数へと変換する方法は、必ず身に付けておかねばなりません。

■ r進数から10進数への変換

r 進数の数値の各けたが、下から順に r^0, r^1, r^2, …の重みをもつことを利用して、次のように行います。

> **r 進数 → 10 進数の変換法**
> 各けたに対して下から順に r^0, r^1, r^2, … をかけた値の総和を求める。

いくつかの例を示します。

例　**$(123)_8$ を 10 進数に変換する。**

$$\begin{aligned}(123)_8 &= 1\times 8^2 + 2\times 8^1 + 3\times 8^0\\ &= 1\times 64 + 2\times 8 + 3\times 1\\ &= 83\end{aligned}$$

例　**$(1FD)_{16}$ を 10 進数に変換する。**

$$\begin{aligned}(1FD)_{16} &= 1\times 16^2 + 15\times 16^1 + 13\times 16^0\\ &= 1\times 256 + 15\times 16 + 13\times 1\\ &= 509\end{aligned}$$

例　**$(101)_2$ を 10 進数に変換する。**

$$\begin{aligned}(101)_2 &= 1\times 2^2 + 0\times 2^1 + 1\times 2^0\\ &= 1\times 4 + 0\times 2 + 1\times 1\\ &= 5\end{aligned}$$

■ 10進数からr進数への変換

基数変換の原理を理解するために、10 進数 1963 を 10 進数に（！）変換してみます。

まずは 1963 を 10 で割ります。1963 ÷ 10 の余り３は、1963 の１の位と同じ値です。また、商である 196 は、１の位である３が消え去ることを無視すれば、1963 を右に１けたずらした値です。

すなわち **10 で割ると、右に 1 けたずれるとともに、最下位のけたが弾きだされます。**

引き続き、商である 196 に対して、同じ操作を行います。10 で割った余り 6 は、変換前の数 1963 における下から 2 けための数字であり、商の 19 は、変換前の数を右に 2 けたずらした値ですね。

このように、10 で割って余りを求め、商に対して除算を繰り返します（右図）。ただし、商が 0 になったら終了です。

この過程で求められた余りを**逆順に**並べると、変換後の 10 進数である 1963 が得られます。

10	1963	
10	196	3
10	19	6
10	1	9
	0	1

＊

上記の手順において “10” を “2” に置きかえると『10 進数を 2 進数に変換する』方法となります。というのも、ある数を 2 で割ることは、2 進数で右に 1 けたずらすことと等しいからです。実際に、このことを確かめてみましょう。

例　$(110)_2$　2 で割る→　$(11)_2$　※ 10 進数では 6 ÷ 2 → 3

例　$(1111)_2$　2 で割る→　$(111)_2$　※ 10 進数では 15 ÷ 2 → 7

それでは、10 進数から 2 進数への変換を具体例で示します。

例　**$(57)_{10}$ を 2 進数に変換する。**

57 を 2 で割ると、商は 28 で余りは 1 です。その商である 28 を 2 で割ると、商は 14 で余りは 0 です。この作業を繰り返す過程は右図のようになり、変換後の 2 進数は 111001 となります。

2	57	
2	28	1
2	14	0
2	7	0
2	3	1
2	1	1
	0	1

この手順は、もちろん 2 進数以外に対しても適用できます。一般には、次のようになります。

> **10 進数 → r 進数の変換法**
>
> 商が 0 になるまで整数を r で割って商と余りを求めていき、得られた余りを逆順に並べる。

例　**$(57)_{10}$ を 8 進数に変換する。**

変換の様子は右図のようになります。変換結果は $(71)_8$ です。

8	57	
8	7	1
	0	7

例　**$(57)_{10}$ を 16 進数に変換する。**

変換の様子は右図のようになります。変換結果は $(39)_{16}$ です。

16	57	
16	3	9
	0	3

■ 2進数･8進数･16進数間の基数変換

２進数、８進数、16 進数間の基数変換は、いとも簡単に行えます。

まずは、右に示す表を見てください。これは 10 進数の 0 ～ 15 を、10 進数、２進数、８進数、16 進数で並べたものです。

この表は、以下のことを示しています。

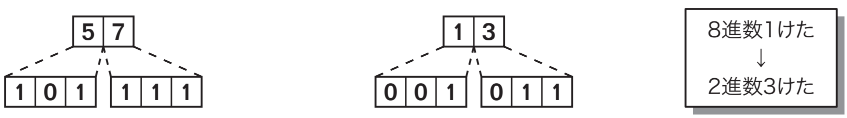
> ８進数１けた 0 ～ 7 は、
> 　２進数の３けた 000 ～ 111 に対応する。
> 16 進数１けた 0 ～ F は、
> 　２進数の４けた 0000 ～ 1111 に対応する。

これらのことを利用して基数変換を行います。

10 進	2 進	8 進	16 進
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

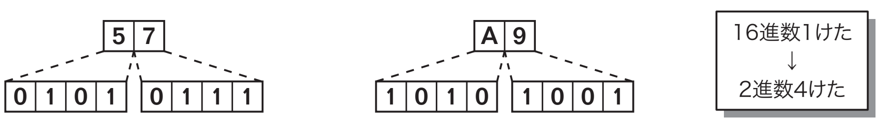
・8 進数 → 2 進数の変換

８進数の各けたを、３けたの２進数に置きかえます。たとえば、$(57)_8$ と $(13)_8$ を２進数に変換すると、それぞれ $(101111)_2$ と $(1011)_2$ になります。



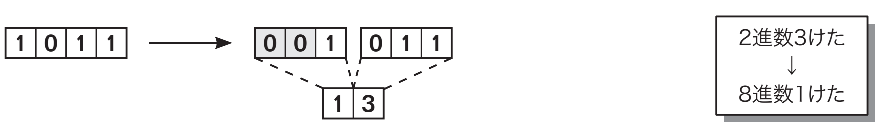
・16 進数 → 2 進数の変換

16 進数の各けたを、４けたの２進数に置きかえます。たとえば、$(57)_{16}$ と $(A9)_{16}$ を２進数に変換すると、それぞれ $(1010111)_2$ と $(10101001)_2$ になります。



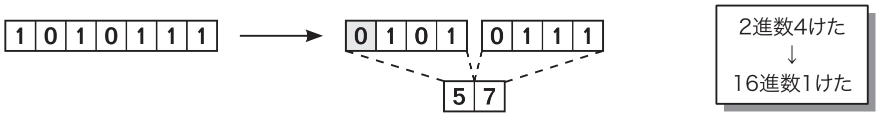
・2 進数 → 8 進数の変換

下から３けたずつに区切り、それぞれの部分を、１けたの８進数に置き換えます。たとえば、$(1011)_2$ は、$(13)_8$ となります（このように、必要に応じて先頭の余白に 0 を追加するとよいでしょう）。



・2 進数 → 16 進数の変換

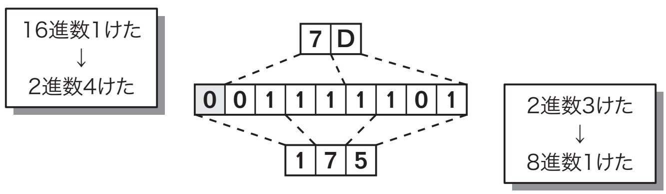
下から 4 けたずつに区切り、それぞれの部分を、1 けたの 16 進数に置き換えます。たとえば、$(1010111)_2$ は、$(57)_{16}$ となります（このように、必要に応じて先頭の余白に 0 を追加するとよいでしょう）。



・16 進数→ 8 進数の変換

『8 進数の 1 けたが、2 進数の 3 けたに対応すること』と『16 進数の 1 けたが、2 進数の 4 けたに対応すること』を組み合わせます。そのため、いったん 2 進数を経由して行うことなります。

$(7D)_{16}$ を 8 進数に変換する手順は下図のようになり、その結果は $(175)_8$ となります。



・8 進数→ 16 進数の変換

16 進数から 8 進数への変換と逆の手順で、いったん 2 進数を経由して行います。

・10 進数→ 4 進数／ 4 進数→ 10 進数の変換

以下に示すように、10 進数から 4 進数への変換が出題されたことがあります。4 進数 1 けたの値 0 〜 3 が、2 進数で 00 〜 11 の 2 けたに対応することを利用すれば、簡単に解けます（4 進数から 10 進数への変換も同様です）。

■ 平成10年度（1998年度）春期 午前 問7

10 進数の 111 を 8 けたの 4 進数として表したものはどれか。

ア　00001233　　イ　00003321　　ウ　00011010　　エ　01101111

単純な基数変換だけで解けるような問題は、最近は出なくなっていますが、その知識が不要となったのではありません。わざわざ問題として出題するほどでもない、当然知っておくべき前提的な知識に〔格下げ〕されたわけです。

論理演算を身につけよう

論理演算に関する問題は、毎回必ず出題されます。日常生活で利用する四則演算と同様に、ただ頭で理解するのではなく身につけておかねばなりません。

■ 論理演算

論理演算は、日本語の〔かつ〕や〔または〕などに相当し、真（1）または偽（0）の入出力を行います。

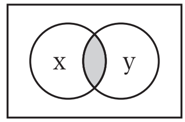
ここでは、各演算について説明するとともに、その**真理値表**と**ベン図**を示します。

・論理積（AND）

二つの入力が両方とも 1 であれば 1 を出力し、そうでなければ 0 を出力します。

x AND y、$x \cdot y$、$x \wedge y$ などと表記します。

	0	1
0	0	0
1	0	1

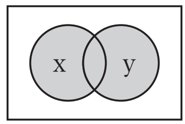


・論理和（OR）

二つの入力のいずれか一方でも 1 であれば 1 を出力し、そうでなければ 0 を出力します。

x OR y、$x + y$、$x \vee y$ などと表記します。

	0	1
0	0	1
1	1	1

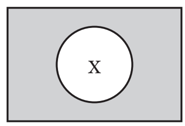


・否定（NOT）

入力が 0 であれば 1 を出力し、1 であれば 0 を出力します。

NOT x、$\overline{x}$、$\neg x$ などと表記します。

0	1
1	0



・排他的論理和（EOR ／ XOR）

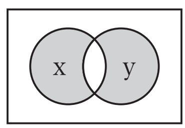
二つの入力のいずれか一方のみが 1 であれば 1 を出力し、そうでなければ 0 を出力します。

『排他的』の英語は“*exclusive*”です。

x EOR y、x XOR y、$x \oplus y$、$x \veebar y$ などと表記します。

なお、$x \oplus y = \overline{x} \cdot y + x \cdot \overline{y}$ であることを覚えておきましょう。

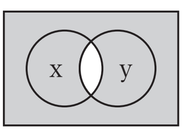
	0	1
0	0	1
1	1	0



・否定論理積（NAND）

論理積の否定です。

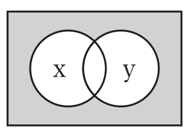
	0	1
0	1	1
1	1	0



・否定論理和（NOR）

論理和の否定です。

	0	1
0	1	0
1	0	0



■ 基本法則

論理演算に関する基本法則を示します。いずれも、左ページの知識で理解できるものです（実際に 0 や 1 を代入すれば確認できます）。

論理演算の計算をスピードアップさせるためにも、これらの法則を何とか頭で理解する程度ではなく、パッと出てくるようにしておきましょう。

なお、四則演算と同様に、以下の規則があります。

- （ ）があれば、その中の演算を先に行う。
- 論理否定が優先し、それに次いで論理積が優先する。
- 同じ優先度の演算は左から順に行う。

・同一則

$x \cdot x = x$　　x どうしの論理積は x となる。

$x + x = x$　　x どうしの論理和は x となる。

・交換則

$x \cdot y = y \cdot x$　　被演算数の順序を変えても結果は変わらない。

$x + y = y + x$　　〃

・結合則

$x \cdot (y \cdot z) = (x \cdot y) \cdot z$　　連続する演算の順序を変えても結果は変わらない。

$x + (y + z) = (x + y) + z$　　〃

・分配則

$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$　　論理積は論理和に対して分配的である。

$x + (y \cdot z) = (x + y) \cdot (x + z)$　　論理和は論理積に対して分配的である。

・吸収則

$x \cdot (x + y) = x$　　y が真でも偽でも、y との論理和との論理積は x となる。

$x + (x \cdot y) = x$　　〃　　y との論理積との論理和は x となる。

・復元則

$\overline{\overline{x}} = x$　　否定の否定は、元に戻る。

・ド・モルガンの法則

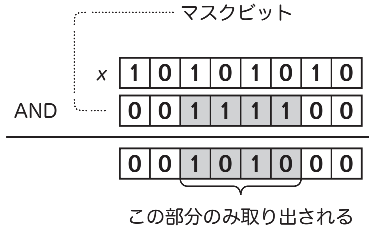
$\overline{x + y} = \overline{x} \cdot \overline{y}$　　“x または y” の否定は、“x でもなく、y でもない”。

$\overline{x \cdot y} = \overline{x} + \overline{y}$　　“x かつ y” の否定は、“x ではない、または y ではない”。

■ 論理積とマスク

ビットの並びから必要な部分のみを抽出するのは、論理積を使うと簡単にできます。

たとえば、“10101010” である x から、3 ビット目〜 6 ビット目を取り出して、それ以外のビットは 0 にすることを考えます。

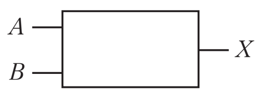


右図に示すように、取り出したい部分を 1 にして、それ以外を 0 にしたビット列を用意します。このようなビット列を**マスクビット**と呼びます。

x とマスクビットとの論理積をとれば、目的の部分がそのまま残って、それ以外のビットが 0 となります。

■ 平成15年度(2003年度)秋期 午前 問17

二つの入力と一つの出力をもつ論理回路で、二つの入力 A、B がともに 1 のときだけ、出力 X が 0 になるものはどれか。



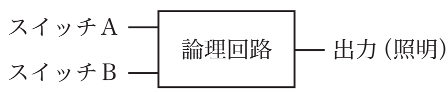
ア　AND 回路　　イ　NAND 回路
ウ　OR 回路　　エ　XOR 回路

■ 平成13年度(2001年度)春期 午前 問18

次の条件を 1 素子で満足する論理回路はどれか。

〔条件〕

階段の上下にあるスイッチ A、B で、一つの照明を点灯、消灯する。すなわち、一方のスイッチの状態にかかわらず、他方のスイッチで照明を点灯、消灯できる。



ア　AND　　イ　NOT　　ウ　OR　　エ　XOR

■ 平成15年度(2003年度)秋期 午前 問6

最上位をパリティビットとする 8 ビット符号において、パリティビット以外の下位 7 ビットを得るためのビット演算はどれか。

ア　16 進数 0F との AND をとる。
イ　16 進数 0F との OR をとる。
ウ　16 進数 7F との AND をとる。
エ　16 進数 FF との XOR（排他的論理和）をとる。

■ 平成15年度(2003年度)秋期 午前 問8

論理式$\overline{Z} = X \cdot \overline{Y} + \overline{X} \cdot Y$の真理値表はどれか。ここで、・は論理積、＋は論理和、$\overline{A}$はAの否定を表す。

ア

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

イ

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

ウ

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

エ

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

▷ p.86 で解説したように、$x \oplus y = \overline{x} \cdot y + x \cdot \overline{y}$を覚えておけば、あっという間に解けます。

■ 平成15年度(2003年度)春期 午前 問7

論理式$\mathrm{A} \vee (\overline{\mathrm{A}} \wedge \mathrm{B})$と等価なものはどれか。ここで、∧は論理積、∨は論理和、$\overline{\mathrm{X}}$はXの否定を表す。

ア　$\mathrm{A} \wedge \mathrm{B}$　　イ　$\mathrm{A} \vee \mathrm{B}$　　ウ　$\mathrm{A} \wedge \overline{\mathrm{B}}$　　エ　$\mathrm{A} \vee \overline{\mathrm{B}}$

■ 平成14年度(2002年度)春期 午前 問8

ビット数が等しい任意のビット列aとbに対して、等式$a = b$と同じことを表すものはどれか。ここで、AND、OR、XORはそれぞれ、ビットごとの論理積、論理和、排他的論理和を表す。

ア　a AND b = 00…0　　イ　a OR b = 11…1
ウ　a XOR b = 00…0　　エ　a XOR b = 11…1

■ 平成13年度(2001年度)秋期 午前 問10

4ビットの2進数で表現された数が二つある。これらのビットごとの論理積は0010であり、ビットごとの論理和は1011となる。二つの数の和はどれか。

ア　1100　　イ　1101　　ウ　1110　　エ　1111

■ 平成13年度(2001年度)春期 午前 問10

ビット列$x = 1100$と$y = 1010$から、1011を得る演算はどれか。ここで、AND、OR、$\overline{\mathrm{Z}}$は、それぞれビットごとの論理積、論理和、Zの否定を表す。

ア　x AND $\overline{y}$　　イ　$\overline{x}$ AND y　　ウ　x OR $\overline{y}$　　エ　$\overline{x}$ OR y