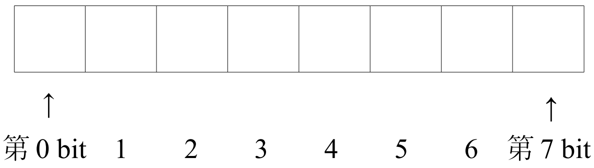
コンピュータⅠ
（第４章 演習問題）

提出日　　：2009年7月26日
所　属　　：情報工学科1年次
学籍番号：095739K
氏　名　　：富銘　孔太

演習問題

4.1　コンピュータのデータ表現に関する次の説明を読んで、設問中の(　)に入れるべき適当な数値を解答群の中から選べ。なお、解答は重複して選んでも良い。



〔データ表現に関する説明〕

（1）データはすべて 8 ビットで表現する。

（2）数値は 8 ビットの 2 進数で表現する。負数は 2 の補数で表現する。

（3）設問中、算術データとは、第 0 ビットを符号とする 2 進数とする。
　　また論理データとは、8 ビットのビットパターンを符号なしの 2 進数とみなしたものとする。
　　シフト演算もこれに準ずる。

（4）論理右シフト演算を行ったとき、空いたビット位置にはすべて 0 が入るが、算術右シフトの場合は、符号ビットと同じものが入る。

（5）算術シフトでは、符号ビットは元のまま不変であるが、論理左シフトでは 8 ビットすべてがシフトの対象となる。いずれの場合も空いたビット位置には 0 が入る。

（6）図 1 は 1 ビット算術右シフトを行った例であり、図 2 は 1 ビット算術左シフトを行った例である。

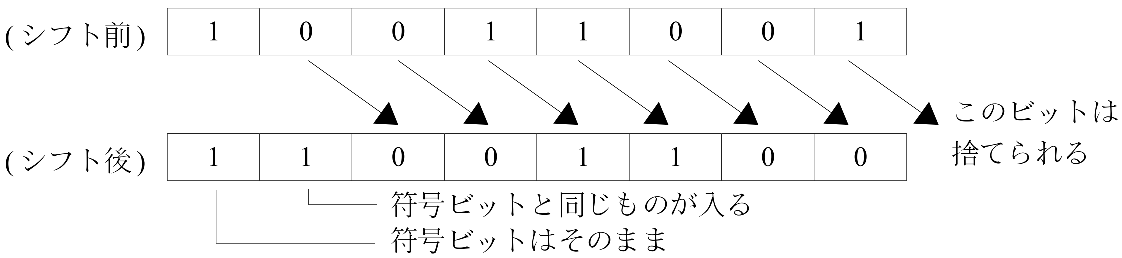


図１　算術右シフト

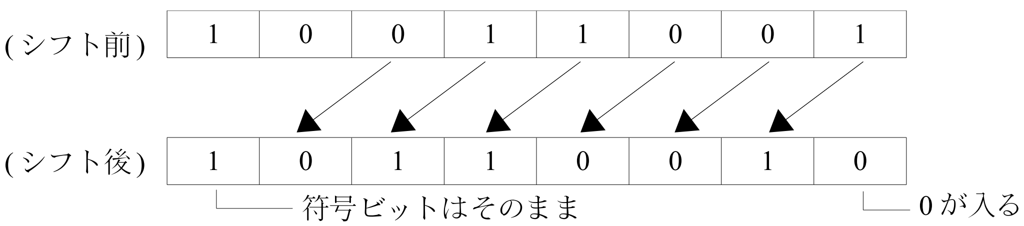


図２　算術左シフト

〔設問１〕

このコンピュータでは、算術データは（　a　）の範囲の値を、論理データは（　b　）の範囲の値を表現できる。

〔設問２〕

10進数の -5 を2進数で表現すると（　c　）となる。これは、論理データとしてみた場合、10進数で表現すると（　d　）である。

〔設問３〕

10進数の -5 を表現する2進数を左へ3ビットシフトした結果を2進数で表現すると、算術シフトでは（　e　）、論理シフトでは（　f　）となる。

また、算術シフトでは、あふれがなければ（　g　）倍することに等しい。

〔設問４〕

一般に正数データを左へ n ビットシフトすることは（　h　）倍にすることに等しく、右へ n ビットシフトすることは（　i　）倍することに等しい。ただし、右へ n ビットシフトした結果、あふれ出たビットの中に1が1つでもあると、その分は（　j　）となる。

〔解　答〕

a	エ	b	ア	c	ウ	d	カ	e	ア
f	ア	g	ウ	h	ア	i	カ	j	ア

〔説　明〕

（設問１）2進数7ビットでは、　$2^7=128$　個の情報を表すことが出来る。よって、最上位ビットが1の場合と0の場合を考えると、算術データの範囲は、−128 ～ 127 となる。

2進数8ビットでは、　$2^8=256$　個の情報を表すことが出来る。
よって、論理データの範囲は、0 ～ 255 となる。

（設問２）10進数 −5 を2進数で表すと $(11111011)_2$ 。論理データとして10進数にすると、

$1 \cdot 2^7+1 \cdot 2^6+1 \cdot 2^5+1 \cdot 2^4+1 \cdot 2^3+1 \cdot 2^1+1 \cdot 2^0=251$　となる。

（設問３）算術シフトでは符号ビットを考慮するので、　$(11111011)_2 \rightarrow (11011000)_2$ 。
論理シフトでは8ビット全てをシフトするので、　$(11011000)_2$　となる。
あふれがなければ、3ビットシフトは　$2^3=8$　倍となる。

（設問４）算術シフトでは、あふれがなければ左に n ビットシフトは　2^n　倍。
右に n ビットシフトは、　$1/2^n$　倍になる。

あふれたビットに1が1つでもあれば、その分は 切り捨て となる。

4.2　論理演算に関する次の設問に答えよ。

〔設問〕

入力変数 A, B に対する出力関数 F が次の真理値表に示すものであるとき、(1)〜(5)の各出力関数を得るのに適する論理式 a〜e, およびその名称 f〜j を解答群の中から選べ。

なお、論理式中の “·” は論理積を, “＋” は論理和を, “ $\overline{X}$ ”は x の否定を表すものとする。

入力変数 / 出力関数	A B	0011 0101	論理式	名　称
(1)	F	0001	a	f
(2)	F	1000	b	g
(3)	F	1110	c	h
(4)	F	0110	d	i
(5)	F	1001	e	j

〔解　答〕

a	イ	b	オ	c	ア	d	ウ	e	エ
f	イ	g	エ	h	ア	i	オ	j	ウ

〔説　明〕

（1）　$(0011)_2$　AND　$(0101)_2$　＝　$(0001)_2$

よって、論理式は、　$F = A \cdot B$　となり名称は、論理積(AND 演算)である。

（2）　$(0011)_2$　OR　$(0101)_2$　＝　$(0111)_2$　となりこれの否定は　$(1000)_2$

よって、論理式は、　$F = \overline{A + B}$　となり名称は、否定論理和(NOR 演算)である。

（3）　$(0011)_2$　AND　$(0101)_2$　＝　$(0001)_2$　となり、これの否定は　$(1110)_2$

よって、論理式は、　$F = \overline{A \cdot B}$　となり名称は、否定論理積(NAND 演算)である。

（4）　$(0011)_2$　AND　$(\overline{0101})_2$　OR　$(\overline{0011})_2$　AND　$(0101)_2$　＝　$(0110)_2$

よって、論理式は、　$F = A \cdot \overline{B} + \overline{A} \cdot B$　となり名称は、排他的論理和(EOR 演算)である。

（5）　$(0011)_2$　AND　$(0101)_2$　OR　$(\overline{0011})_2$　AND　$(\overline{0101})_2$　＝　$(1001)_2$

よって、論理式は、　$F = A \cdot B + \overline{A} \cdot \overline{B}$　となり名称は、一致演算(投下演算)である。

4.3　次の記述による前提を与えたとき、その結論として正しいものを解答郡の中から選べ。

（a）命題 A, B がある。A ならば B である。
（b）集合 A, B, C がある。A は B を含み、B は C を含まない。
（c）論理値 A, B があり、A は真、B は偽である。

〔解　答〕

a	エ	b	ウ	c	エ

〔説　明〕

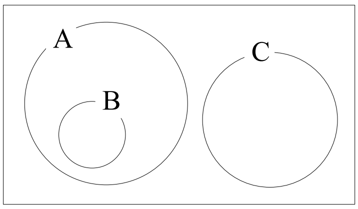
（a）ＡならばＢ（Ａ → Ｂ）の真理値表は次の様になる。

Ａ	Ｂ	Ａ → Ｂ
Ｔ	Ｔ	Ｔ
Ｔ	Ｆ	Ｆ
Ｆ	Ｔ	Ｔ
Ｆ	Ｆ	Ｔ

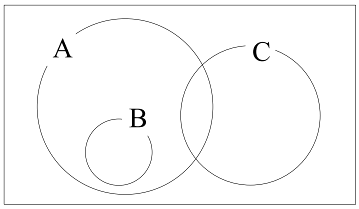
よって、結論は「ＢでなければＡでない」となる。

（b）「 ＡはＢを含み、ＢはＣを含まない」のベン図を書いてみる。

パターン１



パターン２



以上の２パターンが考えられる。
よって、「ＡはＣを含むとも含まないとも決められない」となる。

（c）Ａが真、Ｂが偽のとき。

ア ： A *or* B は、真なので間違い。

イ ： A *and* B は、偽となるので間違い。

ウ ： $\overline{A}$ は、偽なので間違い。

エ ： A *eor* B は、真である。

よって、エが答えとなる。

4.4　ビット演算に関する次の記述中の（　）に入れるべき適切な字句を解答郡の中から選べ。
　8ビット(16進数で表示すると2桁) からなるコードXがアキュムレータにあるとする。
次の(1)～(5) は、アキュムレータ上のXにビットごとの論理演算をほどこして、他のコードを生成する処理である。

（1）Xの最上位ビット(1番左側のビット) を0とするには、（　a　）。
　他の7ビットは変化させないものとする。
（2）Xの最上位ビットを1とするには、（　b　）。他の7ビットは変化させないものとする。
（3）Xの下位4ビットを変化させず、上位4ビットをすべて0とするには、（　c　）。
（4）Xの上位4ビットを変化させず、下位4ビットをすべて1とするには、（　d　）。
（5）Xの全ビットを反転させるには、（　e　）。

〔解　答〕

a	ウ	b	カ	c	ア	d	イ	e	ケ

〔説　明〕

（1）最上位ビットだけを0にするには，$(7F)_{16}$ とANDを取れば良い。

$(1001\,1110)_2$　AND　$(0111\,1111)$　=　$(0001\,1110)_2$

AND演算なので、0のところは0に、1のところは元の値がそのまま出力される。

（2）最上位ビットを1にするには，$(80)_{16}$ とORを取れば良い。

$(0110\,1101)$۲　OR　$(1000\,0000)_2$　=　$(1110\,1101)_2$

最上位ビットに1を足してあげれば良いので、OR演算をする。

（3）下位4ビットを変化させず、上位4ビットを0にするには，$(0F)_{16}$ とANDを取れば良い。

$(1001\,1110)_2$　AND　$(0000\,1111)_2$　=　$(0000\,1110)_2$

(1)と同様な考え方で、0にしたい部分に0でAND演算をする。

（4）上位4ビットを変化させず、下位4ビットを1にするには，$(0F)_{16}$ とORを取れば良い。

(٠١١٠ 1101)$_2$　OR　$(0000\,1111)_2$　=　$(0110\,1111)_2$

(2)と同様な考え方で、1にしたい部分に1でOR演算をする。

（5）全ビットを反転させるには，$(FF)_{16}$ とEORを取れば良い。

$(1001\,1110)_2$　EOR　$(1111\,1111)_2$　=　$(0110\,0001)_2$

EORは、1・0と0・1のとき真で、0・0と1・1のとき偽なので、全ビット1でEOR演算をすることで、全ビット反転させることが出来る。

考　察

問題 4.4 について、特定のビットだけを 0 にしたいときは AND 演算、特定のビットを 1 にしたいときは OR 演算、ビット反転させたいときは EOR 演算を行えば目的にあったコードを生成できるということがわかった。

感　想

今回の課題の内容は、高校の授業でもやったところなので、比較的簡単に解くことが出来ました。でも、算術シフトや論理シフトのところを少し忘れていたので、最初の問題は少々苦戦しました。

参考文献

- コンピュータ・アーキテクチャ入門　　大薮多可志(著)