

(1) 言葉の確認(次の言葉の意味を再度確認し、簡単に説明しなさい。)

ビット: 情報の最小単位、「2つのうちのどちらか」を表現することができる情報

ニブル: 4ビットのこと

バイト: 8ビットのこと

(2) 空白に当てはまる数値/式を書きなさい。

ノイマン型のコンピュータは情報をビットの組合せで表現します。1ビットで表現できる情報は2種類です。2ビットでは(4)種類の情報を表現できます。逆に、16種類の情報を表現するためには(4)ビット必要になります。一般に、nビットで表現できる情報の種類は次の式で表すことができます。

表現できる情報の種類 = (2^n)

(3) 次の表を暗記しなさい。

2進数	10進数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

1. 次の表を暗記しなさい。

2. 次の計算をしなさい。

	2進数	16進数	10進数	
	0000	0	0	1) 0101 1110 ₂
	0001	1	1	+ 0101 0001 ₂
	0010	2	2	<u>1010 1111</u> ₂
	0011	3	3	2) 0111 1111 ₂
	0100	4	4	+ 0000 0001 ₂
	0101	5	5	<u>1000 0000</u> ₂
	0110	6	6	3) 0110 1111 ₂
	0111	7	7	+ 0110 0101 ₂
	1000	8	8	<u>1101 0100</u> ₂
	1001	9	9	4) 1111 1111 ₂
	1010	A	10	- 0110 1101 ₂
	1011	B	11	<u>1001 0010</u> ₂
	1100	C	12	5) 1000 0000 ₂
	1101	D	13	- 0000 0001 ₂
	1110	E	14	<u>0111 1111</u> ₂
	1111	F	15	6) 0011 1100 ₂
				- 0010 1101 ₂
				<u>0000 1111</u> ₂

3. 次の表の空欄を埋めなさい。

	2進数(8桁で)	16進数(2桁で)	10進数
1)	0001 0000	10	16
2)	0011 0010	32	50
3)	0110 0100	64	100
4)	0111 1111	7F	127
5)	1000 0010	82	130
6)	0001 1100	1C	28
7)	0011 1000	38	56
8)	1110 0000	E0	224
9)	0001 1111	1F	31
10)	1010 1010	AA	170

1. 2の補数表現を用いた8ビット符号付き2進数で表現しなさい。

$$1) \quad 30 =_{10} \quad \mathbf{0001 \ 1110}_2$$

$$2) \quad -30 =_{10} \quad \mathbf{1110 \ 0010}_2$$

$$2) \quad 100 =_{10} \quad \mathbf{0110 \ 0100}_2$$

$$2) \quad -100 =_{10} \quad \mathbf{1001 \ 1100}_2$$

$$3) \quad 55 =_{10} \quad \mathbf{0011 \ 0111}_2$$

$$2) \quad -55 =_{10} \quad \mathbf{1100 \ 1001}_2$$

$$4) \quad 80 =_{10} \quad \mathbf{0101 \ 0000}_2$$

$$2) \quad -80 =_{10} \quad \mathbf{1011 \ 0000}_2$$

$$5) \quad 127 =_{10} \quad \mathbf{0111 \ 1111}_2$$

$$2) \quad -127 =_{10} \quad \mathbf{1000 \ 0001}_2$$

2. 次の表で、2進数は2の補数表現を用いた符号付き2進数です。表の空欄を埋めなさい。

$$1) \quad \mathbf{0011 \ 1100}_2$$

$$\mathbf{60}_{10}$$

$$+ \mathbf{0010 \ 1101}_2$$

10進数で同じ計算は

$$+$$

$$\mathbf{45}_{10}$$

$$\mathbf{0110 \ 1001}_2$$

$$\mathbf{105}_{10}$$

$$2) \quad \mathbf{0110 \ 0100}_2$$

$$\mathbf{100}_{10}$$

$$+ \mathbf{1000 \ 0001}_2$$

10進数で同じ計算は

$$+$$

$$\mathbf{-127}_{10}$$

$$\mathbf{1110 \ 0101}_2$$

$$\mathbf{-27}_{10}$$

$$3) \quad \mathbf{1110 \ 0100}_2$$

$$\mathbf{-28}_{10}$$

$$+ \mathbf{0100 \ 0001}_2$$

10進数で同じ計算は

$$+$$

$$\mathbf{65}_{10}$$

$$\mathbf{0010 \ 0101}_2$$

$$\mathbf{37}_{10}$$

$$4) \quad \mathbf{1110 \ 0100}_2$$

$$\mathbf{-28}_{10}$$

$$+ \mathbf{1100 \ 0001}_2$$

10進数で同じ計算は

$$+$$

$$\mathbf{-63}_{10}$$

$$\mathbf{1010 \ 0101}_2$$

$$\mathbf{-91}_{10}$$

$$5) \quad \mathbf{0011 \ 0010}_2$$

$$\mathbf{50}_{10}$$

$$- \mathbf{1100 \ 1111}_2$$

10進数で同じ計算は

$$-$$

$$\mathbf{-49}_{10}$$

$$\mathbf{0110 \ 0011}_2$$

$$\mathbf{99}_{10}$$

$$6) \quad \mathbf{1110 \ 0100}_2$$

$$\mathbf{-28}_{10}$$

$$- \mathbf{0001 \ 1001}_2$$

10進数で同じ計算は

$$-$$

$$\mathbf{25}_{10}$$

$$\mathbf{1100 \ 1011}_2$$

$$\mathbf{-53}_{10}$$

1. 2進数から10進数に変換しなさい。

1) $0001.0101_2 =$	1.3125_{10}
2) $1101.0101_2 =$	13.3125_{10}
3) $110.10001_2 =$	6.53125_{10}
4) $101.10101_2 =$	5.65625_{10}
5) $001.11111_2 =$	1.96875_{10}

2. 10進数から8ビット2進数に変換しなさい。

1) $10.5_{10} =$	10101000_2	小数点
2) $9.625_{10} =$	10011010_2	小数点
3) $13.25_{10} =$	11010100_2	小数点
4) $7.09375_{10} =$	11100011_2	小数点
5) $3.28125_{10} =$	01101001_2	小数点

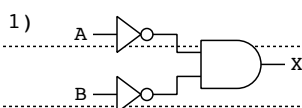
3. 次の文字の文字コードを答えなさい。(ASCII文字コード表から)

1) 文字「A」	41_{16}
2) 文字「Z」	$5A_{16}$
3) 文字「a」	61_{16}
4) 文字「l」	31_{16}
5) 文字「#」	23_{16}

4. 次の文字コードの文字を答えなさい。(ASCII文字コード表から)

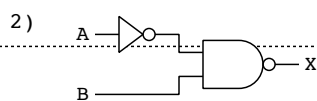
1) 33_{16}	3
2) $4F_{16}$	O
3) 21_{16}	!
4) $7A_{16}$	Z

1. 次の論理回路の論理式と真理値表を答えなさい。



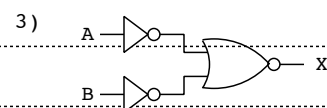
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

$$X = \overline{A} \cdot B$$



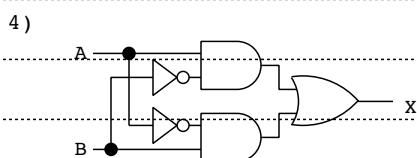
A	B	X
0	0	1
0	1	0
1	0	1
1	1	1

$$X = \overline{A} \cdot \overline{B}$$



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

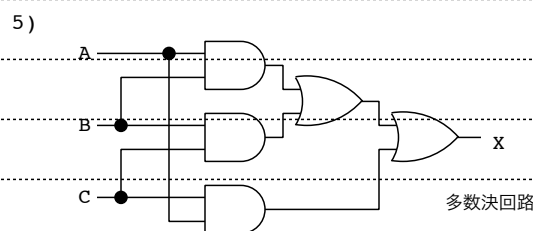
$$X = \overline{A} + \overline{B}$$



XORと同じ

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

$$X = A \cdot \overline{B} + \overline{A} \cdot B$$



多数決回路

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

$$X = A \cdot B + B \cdot C + C \cdot A$$

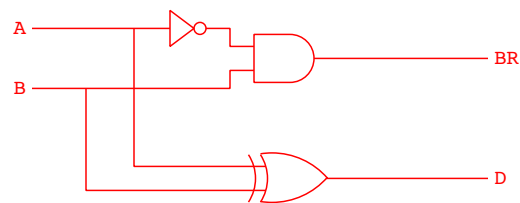
1. 引き算器(減算器)を設計しなさい。

1) 半減算器(HS:Half Subtractor) (A-Bを計算する回路)

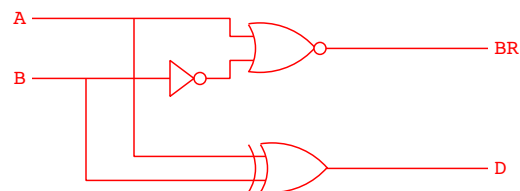
真理値表

A	B	BR	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

回路図



別解:



論理式

$$BR = \overline{A} \cdot B$$

別解: $A + B$

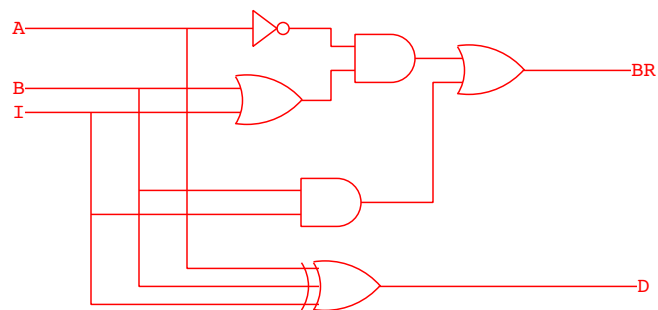
$$D = A \oplus B$$

2) 全減算器(FS:Full Subtractor) (A-B-Iを計算する回路)

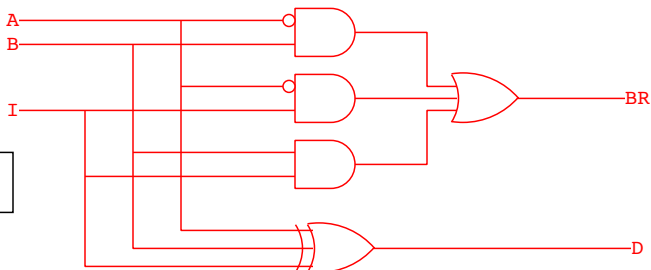
真理値表

A	B	I	BR	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

回路図



別解:



論理式

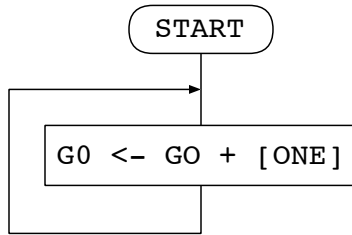
$$BR = \overline{A} \cdot (B + I) + B \cdot I$$

別解: $\overline{A} \cdot B + \overline{A} \cdot I + B \cdot I$

$$D = A \oplus B \oplus I$$

次のプログラムを実行して、ランプの変化を観察しなさい。

フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	30 04	LOOP	ADD	G0, ONE
02	A0 00		JMP	LOOP
04	01	ONE	DC	1

(1) PCを表示したままステップモードで実行する。PCはどのように変化しましたか？

00 -> 02 -> 00 -> 02 の順に変化した。

(ADD命令と、JMP命令を交互に実行している様子が分かった。)

正解は教えた範囲では
分からないので、
想像で答える。

(2) PCを表示したまま通常実行する。PCはどのように変化していると考えられますか？(停止にはSTOPボタンを使用)

D2, D1, D0 が点灯した。D0 が点灯したことから、命令の実行途中でPCは1番地ずつ増加していると想像できる。

D2が点灯したことからJMP命令実行時に、一旦、PCが04になっていることが想像できる。

以上を総合するとPCは、00 -> 01 -> 02 -> 03 -> 04 -> 00 の順に変化していると考えられる。

(3) G0を表示したままステップモードで実行する。G0はどのように変化しましたか？

00 -> 01 -> 01 -> 02 -> 02 -> 03 -> 03 の順に変化した。

(1命令実行する毎に止まっているのでADD命令を実行するのは2回に1回だけ)

(4) PCを表示したまま(ADD命令を実行する毎に止まるように)ブレークモードで実行する。

PCはどのようにになりましたか？

02 のまま変化しないように見える。

(ADDを実行後に止まるので、PCが次のJMP命令を実行する直前の02になった状態で止まる。)

(5) G0を表示したまま(ADD命令を実行する毎に止まるように)ブレークモードで実行する。

G0はどのようにになりましたか？

00 -> 01 -> 02 -> 03 -> 04 -> 05 -> 06 の順に変化した。

(ADD命令を実行する度に止まるので)

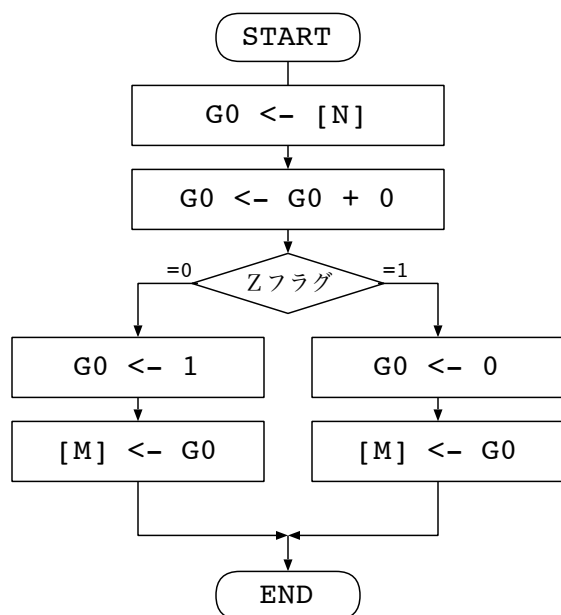
(6) G0を表示したまま(ADD命令を実行する毎に止まるように)ブレークモードで実行する。

各フラグが「1になる条件」は何ですか？

	1になる条件
Cフラグ	FFHに1加えた時と、00Hから1引いた時(符号無し2進数と考えたときのオーバーフロー)
Zフラグ	FFHに1加えた時と、01Hから1引いた時(計算結果が0になったとき)
Sフラグ	計算結果のD7が1のとき(符号付き2進数と考えたときのマイナス)

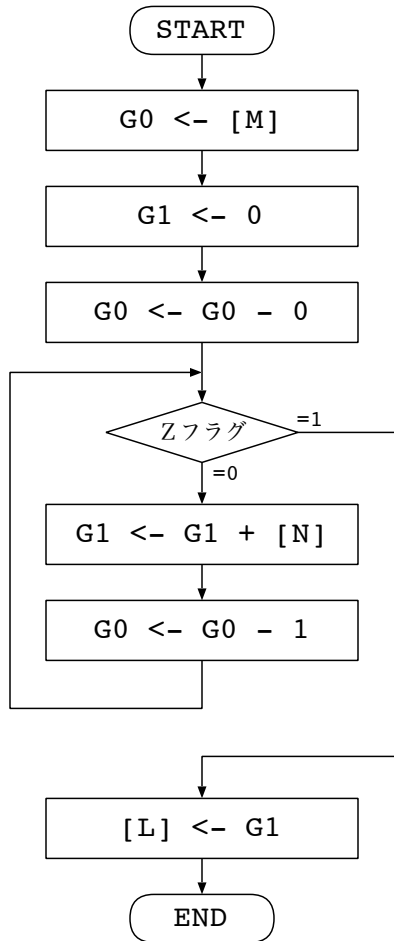
(1) N番地の値がゼロならM番地にゼロを、そうでなければM番地に1を格納するプログラム

フローチャート

[illegible]

(2) 前のプログラムをN番地、または、M番地のデータがゼロの場合も正常に動作するように改良しなさい。

フローチャート



ポイント

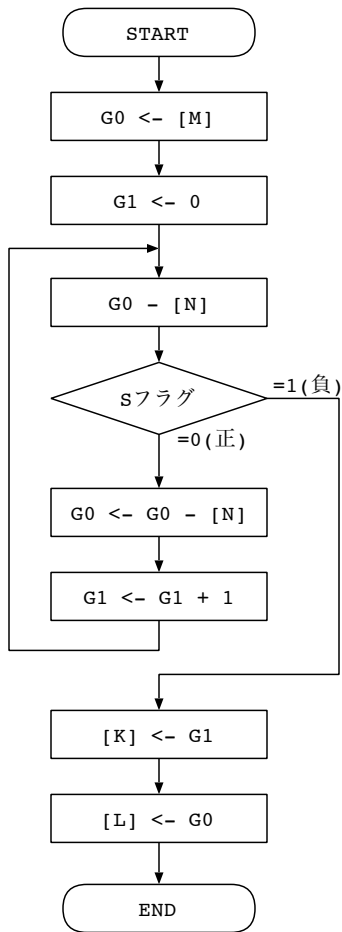
[M]がゼロの場合にそなえ、条件判断を最初にする。
フラグの値が確実に変化するようにSUB命令を使用している。

JMP命令はフラグを変化しないので、SUBとJZの間にJMPを実行してもよい。

[illegible]

(1) M番地のデータをN番地のデータで割り、商をK番地、余りをL番地に格納するプログラムを作りなさい。

フローチャート

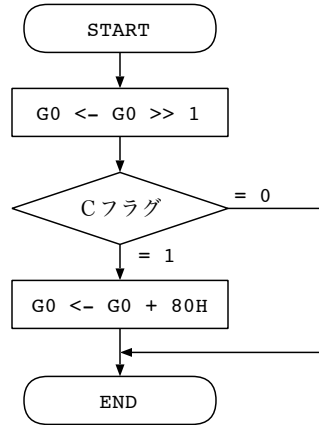


G0:割られる数
G1:引いた回数

アドレス	機械語	ラベル	オペレーション	オペランド
00	10 15		LD	G0,M
02	14 13		LD	G1,ZERO
04	50 16	LOOP	CMP	G0,N
06	AC 0E		JM	OWARI
08	40 16		SUB	G0,N
0A	34 14		ADD	G1,ONE
0C	A0 04		JMP	LOOP
0E	24 17	OWARI	ST	G1,K
10	20 18		ST	G0,L
12	FF		HALT	
13	00	ZERO	DC	0
14	01	ONE	DC	1
15	64	M	DC	100
16	0A	N	DC	10
17	00	K	DS	1
18	00	L	DS	1
19				

(1) G0 の値を 1 ビット右回転するプログラムを作りなさい。

フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	93	L0	SHRL	G0
01	A8 05		JC	L1
03	A0 07		JMP	L2
05	30 0A	L1	ADD	G0, BIT
07	FF	L2	HALT	
08	A0 00		JMP	L0
0A	80	BIT	DC	80H

(2) M番地のデータの7倍をL番地に格納するプログラムを作りなさい。

フローチャート

解答例 1

$$\begin{aligned}
 L &= M \times 7 \\
 &= M \times 4 + M \times 2 + M
 \end{aligned}$$

アドレス	機械語	ラベル	オペレーション	オペランド
00	10 0D		LD	G0, M
02	91		SHLL	G0
03	20 0F		ST	G0, TMP
05	91		SHLL	G0
06	30 0F		ADD	G0, TMP
08	30 0D		ADD	G0, M
0A	20 0E		ST	G0, L
0C	FF		HALT	
0D	05	M	DC	5
0E	00	L	DS	1
0F	00	TMP	DS	1

フローチャート

解答例 2

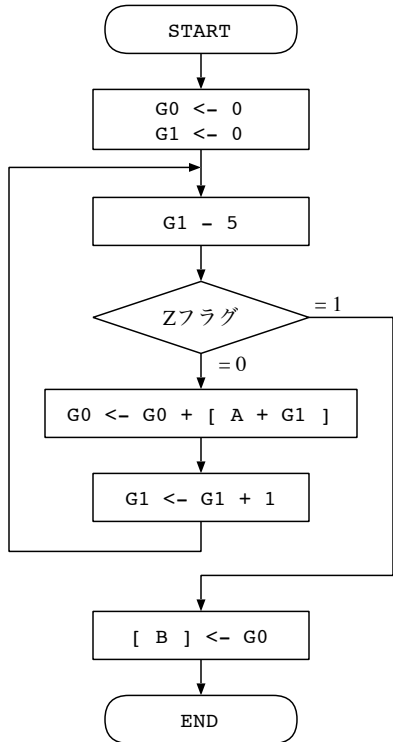
$$\begin{aligned}
 L &= M \times 7 \\
 &= M \times 8 - M
 \end{aligned}$$

アドレス	機械語	ラベル	オペレーション	オペランド
00	10 0A		LD	G0, M
02	91		SHLL	G0
03	91		SHLL	G0
04	91		SHLL	G0
05	40 0A		SUB	G0, M
07	20 0B		ST	G0, L
09	FF		HALT	
0A	05	M	DC	5
0B	00	L	DS	1

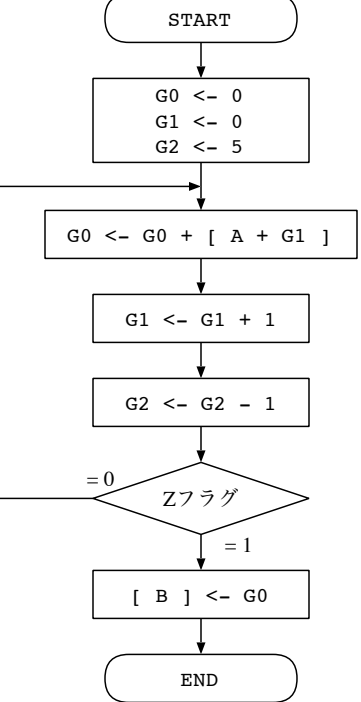
(1) A番地からの5バイトのデータの和をB番地に求めるプログラムを作りなさい。

フローチャート

解答例 1



解答例 2

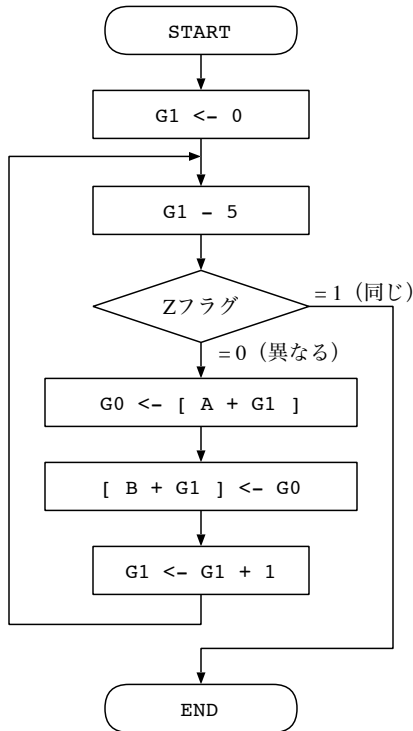


アドレス	機械語	ラベル	オペレーション	オペランド
解答例 1				
00	13 00		LD	G0, #0
02	17 00		LD	G1, #0
04	57 05	LOOP	CMP	G1, #5
06	A4 0E		JZ	STOP
08	31 11		ADD	G0, A, G1
0A	37 01		ADD	G1, #1
0C	A0 04		JMP	LOOP
0E	20 16	STOP	ST	G0, B
10	FF		HALT	
11	02 01 05	A	DC	2,1,5,9,3
14	09 03			
16	00	B	DS	1
17				
解答例 2				
00	13 00		LD	G0, #0
02	17 00		LD	G1, #0
04	1B 05		LD	G2, #5
06	31 13	LOOP	ADD	G0, A, G1
08	37 01		ADD	G1, #1
0A	4B 01		SUB	G2, #1
0C	A4 10		JZ	STOP
0E	A0 06		JMP	LOOP
10	20 18	STOP	ST	G0, B
12	FF		HALT	
13	02 01 05	A	DC	2,1,5,9,3
16	09 03			
18	00	B	DS	1
19				

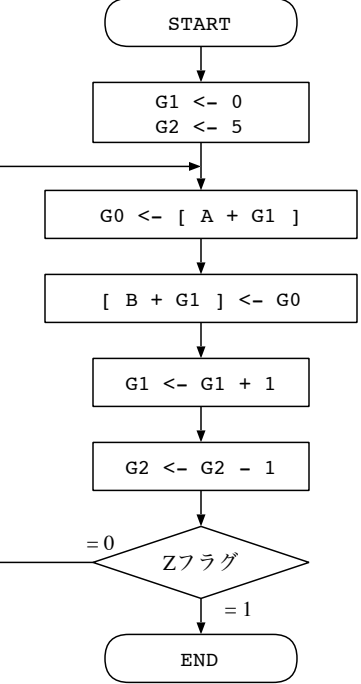
(2) A番地からの5バイトをB番地からの5バイトにコピーするプログラムを作りなさい。

フローチャート

解答例 1



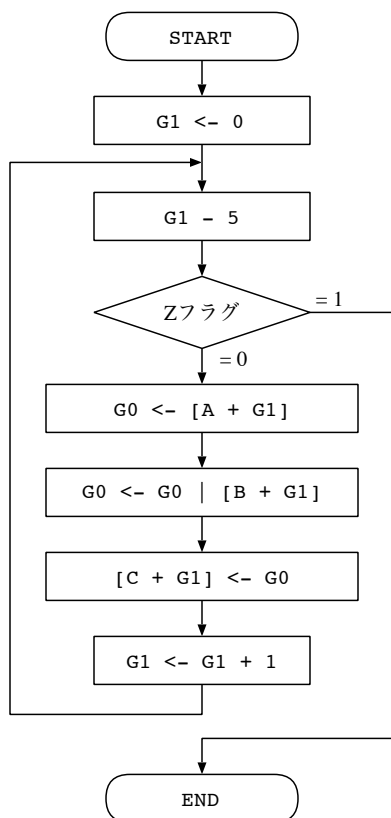
解答例 2



アドレス	機械語	ラベル	オペレーション	オペランド
	解答例 1			
00	17 00		LD	G1,#0
02	57 05	LOOP	CMP	G1,#5
04	A4 0E		JZ	STOP
06	11 0F		LD	G0,A,G1
08	21 14		ST	G0,B,G1
0A	37 01		ADD	G1,#1
0C	A0 02		JMP	LOOP
0E	FF	STOP	HALT	
0F	02 01 05	A	DC	2,1,5,9,3
12	09 03			
14	00 00 00	B	DS	5
17	00 00			
19				
	解答例 2			
00	17 00		LD	G1,#0
02	1B 05		LD	G2,#5
04	11 11	LOOP	LD	G0,A,G1
06	21 16		ST	G0,B,G1
08	37 01		ADD	G1,#1
0A	4B 01		SUB	G2,#1
0C	A4 10		JZ	STOP
0E	A0 04		JMP	LOOP
10	FF	STOP	HALT	
11	02 01 05	A	DC	2,1,5,9,3
14	09 03			
16	00 00 00	B	DS	5
19	00 00			
1B				

(2) A番地とB番地からの5バイトの論理和をC番地からの5バイトに求めるプログラムを作りなさい。

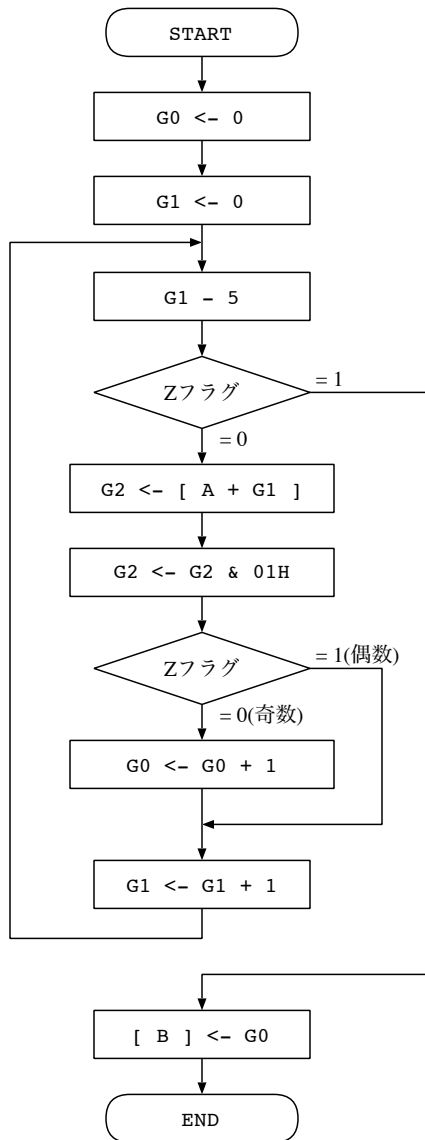
フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	17 00		LD	G1, #0
02	57 05	LOOP	CMP	G1, #5
04	A4 10		JZ	STOP
06	11 11		LD	G0, A, G1
08	71 16		OR	G0, B, G1
0A	21 1B		ST	G0, C, G1
0C	37 01		ADD	G1, #1
0E	A0 02		JMP	LOOP
10	FF	STOP	HALT	
11				
11	12 34	A	DC	12H, 34H
13	56 78		DC	56H, 78H
15	9A		DC	9AH
16				
16	0F 0F	B	DC	0FH, 0FH
18	0F 0F		DC	0FH, 0FH
1A	0F		DC	0FH
1B				
1B	00 00	C	DS	5
1D	00 00			
1F	00			

(1) A番地からの5バイトのデータのなかで、奇数の個数をB番地に求めるプログラムを作りなさい。

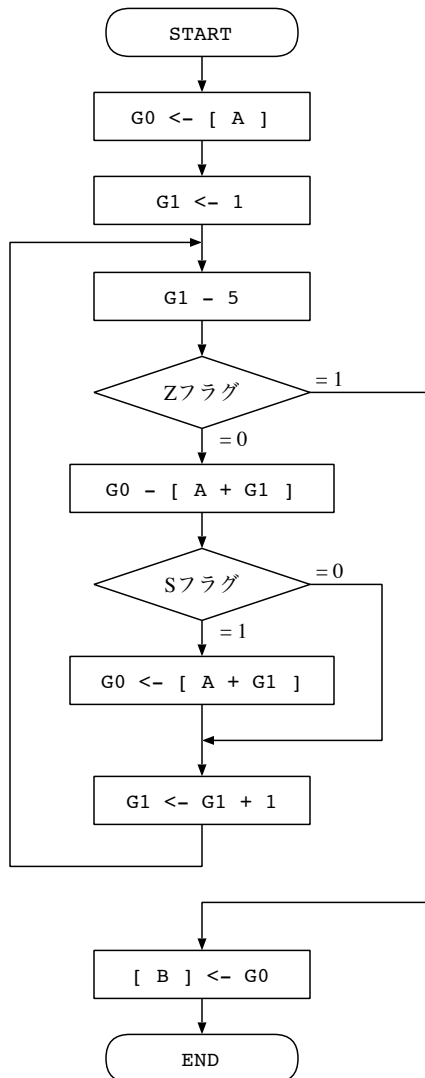
フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	13 00		LD	G0,#0
02	17 00		LD	G1,#0
04	57 05	LOOP	CMP	G1,#5
06	A4 14		JZ	FIN
08	19 17		LD	G2,A,G1
0A	6B 01		AND	G2,#01H
0C	A4 10		JZ	EVEN
0E	33 01		ADD	G0,#1
10	37 01	EVEN	ADD	G1,#1
12	A0 04		JMP	LOOP
14	20 1C	FIN	ST	G0,B
16	FF		HALT	
17				
17	01 02 03	A	DC	1,2,3,4,5
1A	04 05			
1C	00	B	DS	1

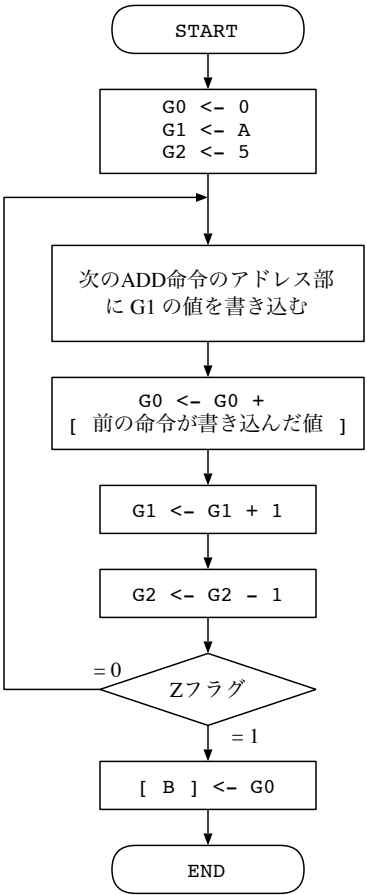
(2) A番地からの5バイトのデータの最大値をB番地に求めるプログラムを作りなさい。

フローチャート

[illegible]

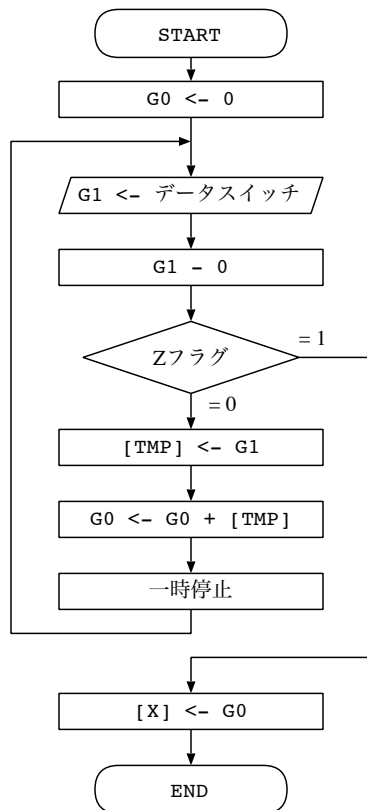
(3) A番地の5バイトデータの合計をB番地に求めるプログラムをインデクスモードなしに作りなさい。

フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	13 00		LD	G0,#0
02	17 15		LD	G1,#A
04	1B 05		LD	G2,#5
06	24 09	LOOP	ST	G1,ADDR
08	30		DC	30H
09	00	ADDR	DS	1
0A	37 01		ADD	G1,#1
0C	4B 01		SUB	G2,#1
0E	A4 12		JZ	STOP
10	A0 06		JMP	LOOP
12	20 1A	STOP	ST	G0,B
14	FF		HALT	
15	02 01 05	A	DC	2,1,5,9,3
19	09 03			
1A	00	B	DS	1
1B				
別の書き方（途中まで）				
00	13 00		LD	G0,#0
02	17 15		LD	G1,#A
04	1B 05		LD	G2,#5
06	24 09	LOOP	ST	G1,X+1
08	30 00	X	ADD	G0,0
0A	37 01		ADD	G1,#1
0C	4B 01		SUB	G2,#1
0E	A4 12		JZ	STOP
...				

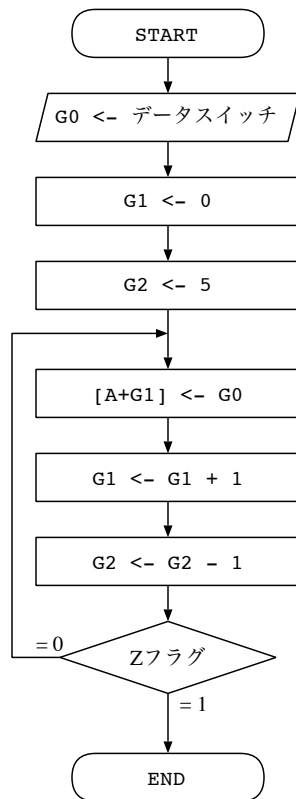
フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	13 00		LD	G0,#0
02	C4 00	LOOP	IN	G1,00H
04	57 00		CMP	G1,#0
06	A4 0F		JZ	FINISH
08	24 12		ST	G1,TMP
0A	30 12		ADD	G0,TMP
0C	FF		HALT	
0D	A0 02		JMP	LOOP
0F				
0F	20 13	FINISH	ST	G0,X
11	FF		HALT	
12				
12	00	TMP	DS	1
13	00	X	DS	1

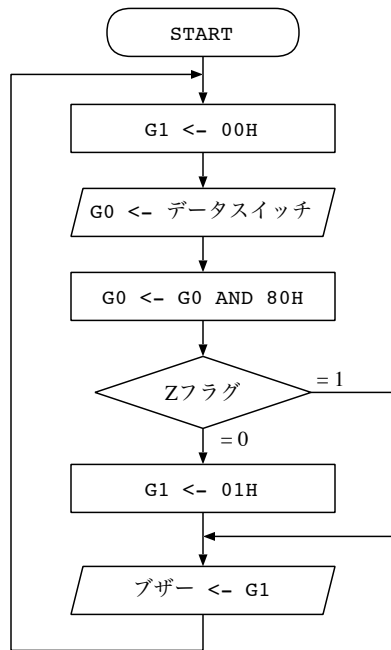
(2) A番地からの5バイトを、データスイッチにセットしてあった値で埋めつくすプログラムを作りなさい。

フローチャート

[illegible]

- (3) データスイッチのビット 7 (D7) をオンにしている間、ブザーを鳴らすプログラムを作りなさい。
(但し、論理演算命令を用いること。)

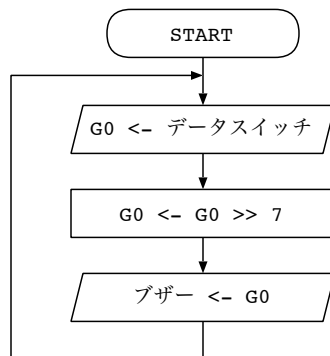
フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	17 00	START	LD	G1,#00H
02	C0 00		IN	G0,00H
04	63 80		AND	G0,#80H
06	A4 0A		JZ	OUTP
08	17 01		LD	G1,#01H
0A	C7 00	OUTP	OUT	G1,00H
0C	A0 00		JMP	START
0E				

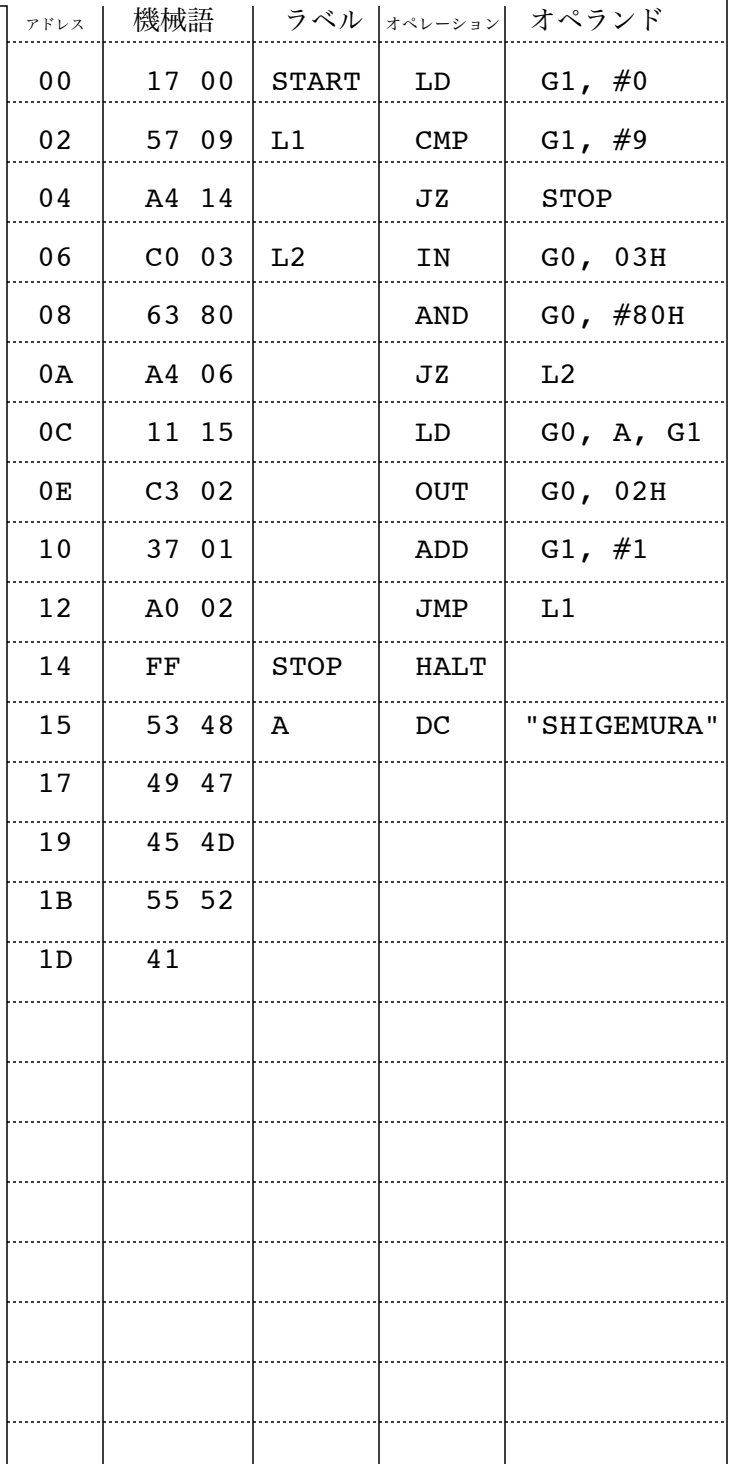
- (4) データスイッチのビット 7 (D7) をオンにしている間、ブザーを鳴らすプログラムを作りなさい。
(但し、シフト命令を用いること。)

フローチャート



アドレス	機械語	ラベル	オペレーション	オペランド
00	C0 00	START	IN	G0,00H
02	93		SHRL	G0
03	93		SHRL	G0
04	93		SHRL	G0
05	93		SHRL	G0
06	93		SHRL	G0
07	93		SHRL	G0
08	93		SHRL	G0
09	C3 00		OUT	G0,00H
0B	A0 00		JMP	START
0D				

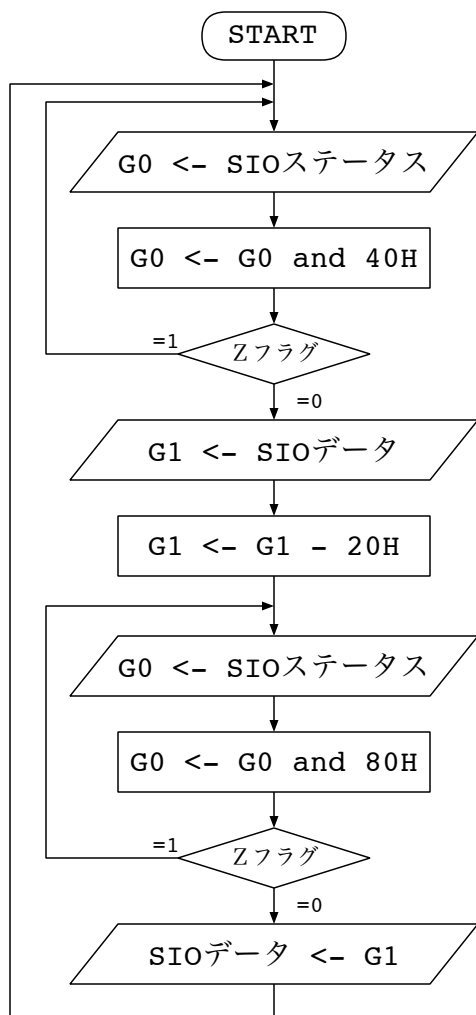
フローチャート



- (1) IN 命令、OUT 命令の使いかたは分かったか？
- (2) AND 命令を使用したビットのテスト(0/1の判定)は理解できたか？
AND 命令でフラグが変化する。
命令表を見て、フラグが変化する命令、変化しない命令の区別を確認！！
- (3) 文字コードの決め方は分かったか？

(2) アルファベット小文字を大文字に変換し表示するプログラム(第1版)を作りなさい。

フローチャート

[illegible]

