

# 付 録

市販の PLC は様々な命令・機能を有しているが、ここでは敢えて接点とリレーだけでデジタル回路のテキスト等<sup>(1)~(4)</sup>にある回路のラダー図を作成する。当然 PLC の内部ではこんな事はしていない。以下のラダー図の設計には ON=1, OFF=0 として、基本的なデジタル回路の設計法を利用している。論理式をラダー図に変換するとき b 接点は利用できるため 1 変数の否定は使えるが、標準では NOT 命令 (INV 命令) がないため論理式の否定はできない。したがって、1 変数についてのみの否定となるように変形するか、一時変数にしてから否定をとる必要がある。また、基本的な最小積和形的设计で得られた結果より命令数が少なくなる論理式もありうるが、そのままにしてある。

以下のラダー図は次のような PLC の性質を利用しており、プログラムの順序を入れ替えると機能しないことがある。また、リレー回路では正しく機能しない。

- ・ スキャンサイクルの最初にスイッチ、センサ等の状態が一括入力され、内部メモリに記憶される。接点条件はこの内部メモリの状態を使用する。
- ・ 内部リレー(メモリ)はプログラムの実行にしたがい逐次変化する。
- ・ 出力の状態は一旦内部メモリに記憶され、スキャンサイクルの最後に出力が一括確定される。

以下、入出力信号の記号は原則として次の通り。それ以外の場合は適宜明記する。

A, B, X : 組み合わせ回路等の入力

多ビットの時は A<sub>0</sub>, A<sub>1</sub>, …とする。A<sub>0</sub>が LSB

Y : 組み合わせ回路等の出力

S : セレクタ入力, 加算出力

C : キャリ入出力

Q : フリップフロップやレジスタの出力

T, D, J, K : フリップフロップの入力

CLK : クロックパルス, シフトパルス

CLR : クリア

PRS : プリセット

M : 一時メモリ

## 1. 組み合わせ回路

以下, A の否定 (NOT) は  $\bar{A}$  と表す。

### ● 論理積 AND

$$Y = A \cdot B \quad (\text{以下, 記号} \cdot \text{は省略する})$$

定数 0 は  $Y = A\bar{A} = 0$  として得ることができる。

### ● 論理和 OR

$$Y = A + B$$

優先順位は NOT > AND > OR とする。

### ● 排他的論理和 XOR (EXOR)

$$Y = A \oplus B = A\bar{B} + \bar{A}B = (\bar{A} + \bar{B})(A + B)$$

2 進数 1 桁の桁上りを除いた加算。

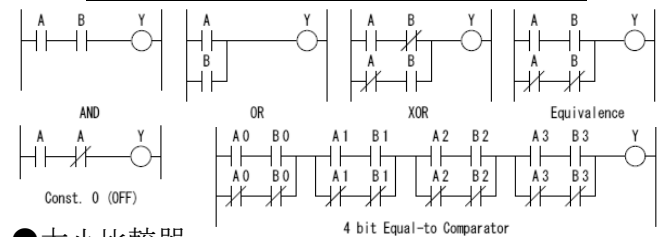
### ● 一致, 対等 EQV

排他的論理和 XOR の否定。

$$Y = A \leftrightarrow B = AB + \bar{A}\bar{B} = (\bar{A} + B)(A + \bar{B})$$

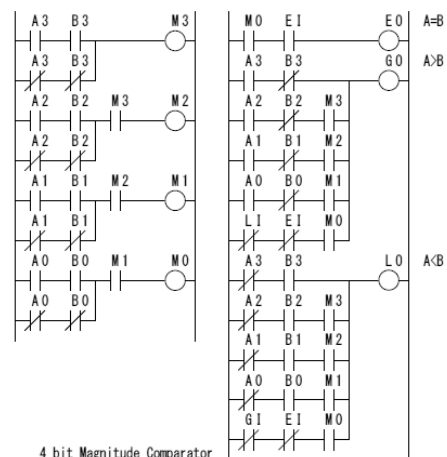
多ビットの場合は AND でつなげて一致検出器を構成する。

A	B	AND	OR	XOR	EQV
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	1	0	1



### ● 大小比較器

4 ビットのデータの比較器を示す。A=B のとき E<sub>0</sub> が ON, A>B のとき G<sub>0</sub> が ON, A<B のとき L<sub>0</sub> が ON になる。E<sub>1</sub>, G<sub>1</sub>, L<sub>1</sub> は下位 4 ビットの比較結果であり、使用しない場合は E<sub>1</sub> を ON, G<sub>1</sub> と L<sub>1</sub> を OFF にすればよい。具体的には定数 0 (OFF) を用いるか, E<sub>1</sub> の a 接点は短絡し, E<sub>1</sub> の b 接点およびその論理積は削除する。



### ●加算器

全加算器(FA)は A と B と下位桁からのキャリ(桁上がり)入力  $C_1$  の 3 ビットの加算を行う。加算出力を S, キャリ出力を  $C_0$  とすると,

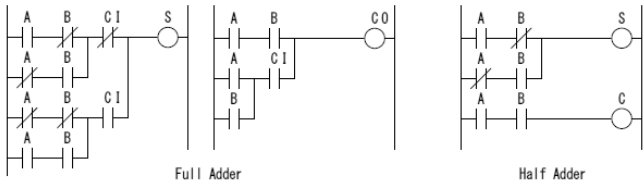
$$S = A \oplus B \oplus C_1 = \overline{A}\overline{B}C_1 + \overline{A}B\overline{C_1} + A\overline{B}\overline{C_1} + ABC_1$$

$$= (\overline{A}\overline{B} + \overline{A}B)\overline{C_1} + (A\overline{B} + AB)C_1$$

$$C_0 = AB + (A \oplus B)C_1 = AB + AC_1 + BC_1$$

$$= AB + (A + B)C_1$$

また, キャリ加算を行わない半加算器(HA)を示す。なお, 桁上先見加算器(CLA)については省略する。



### ●減算器

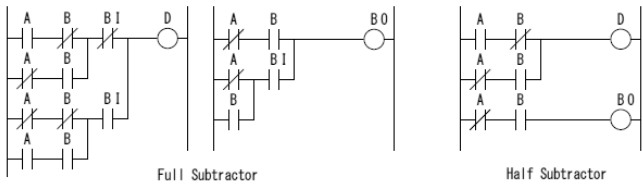
各 1 ビットの被減数を A, 減数を B, 下位桁のボロー(桁借り)を  $B_1$ , 差を D, 上位桁からのボローを  $B_0$  とすると, 2 の補数を用いない全減算器(FS)は次のようになる。

$$D = A \oplus B \oplus B_1 = \overline{A}\overline{B}\overline{B_1} + \overline{A}B\overline{B_1} + A\overline{B}\overline{B_1} + AB\overline{B_1}$$

$$= (\overline{A}\overline{B} + \overline{A}B)\overline{B_1} + (A\overline{B} + AB)B_1$$

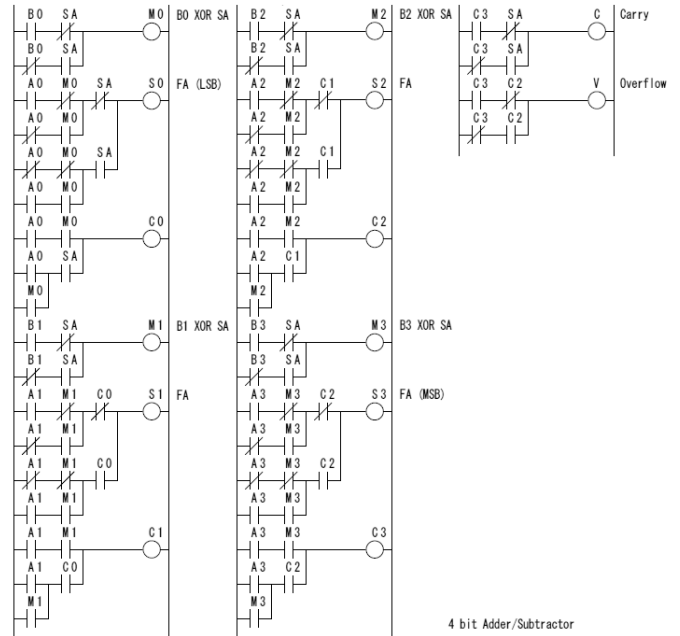
$$B_0 = \overline{A}B + \overline{A}B_1 + BB_1 = \overline{A}B + (\overline{A} + B)B_1$$

また, 下位桁のボローのない半減算器(HS)を示す。



### ●加減算器

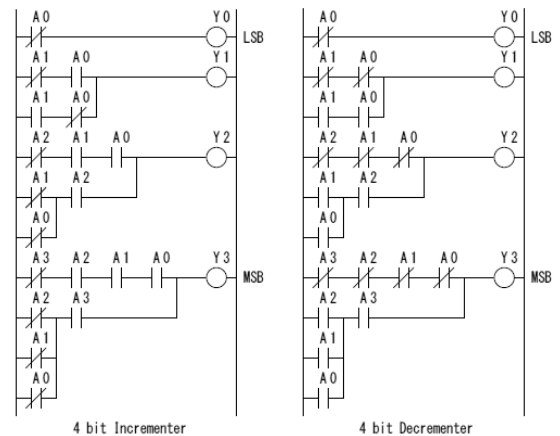
減算を 2 の補数を用いて行う。2 の補数は否定をとることで得られる 1 の補数に 1 を加えることによって求める。加算と減算に共用可能な 4 ビット加減算器を示す。  $S_A$  が ON のときは減算, OFF のときは加算を行う。最下位ビットへのキャリ入力へこの  $S_A$  を入力すると, 加算のとき 0 が, 減算のとき 1 が加えられることになる。B あるいはその否定  $\overline{B}$  は  $S_A$  により選択される。この部分は  $S_A$  と B の XOR を取っていることと等しい。なお, この回路には 2 の補数を用いた加減算の結果が範囲外の値になったとき ON するオーバーフロー V を設けてある。また, キャリ C は符号無し 2 進数の演算結果であり, キャリまたはボローがあるとき  $C=1$  となる。なお, 多段接続には  $C_3$  を用いる。



### ●インクリメンタ・デクリメンタ

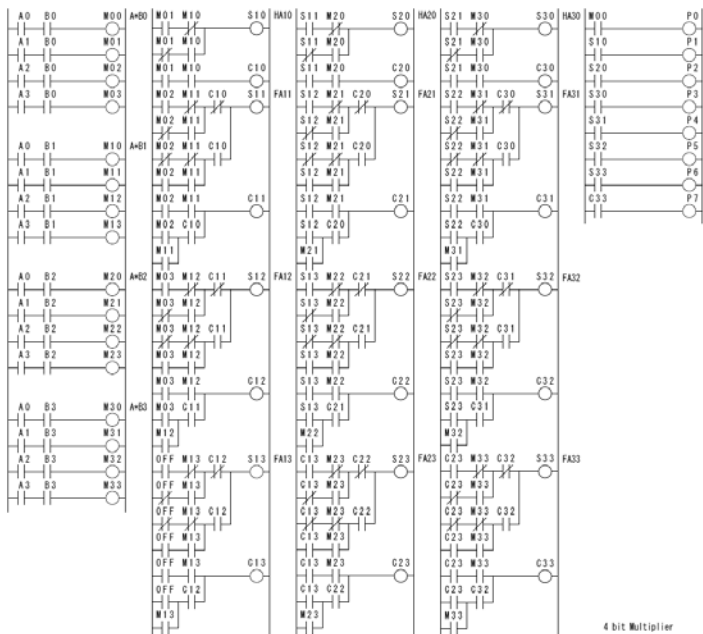
4 ビットの入力 ( $A_0 \sim A_3$ ) を 1 だけ増やす回路と 1 だけ減らす回路を加算器や減算器を使用せずに論理式により構成した例を示す。

- ・インクリメンタ  $Y_0 = \overline{A_0}$   
 $Y_1 = \overline{A_1}A_0 + A_1\overline{A_0}$   
 $Y_2 = \overline{A_2}A_1A_0 + A_2(\overline{A_1} + \overline{A_0})$   
 $Y_3 = \overline{A_3}A_2A_1A_0 + A_3(\overline{A_2} + \overline{A_1} + \overline{A_0})$
- ・デクリメンタ  $Y_0 = \overline{A_0}$   
 $Y_1 = \overline{A_1}\overline{A_0} + A_1A_0$   
 $Y_2 = \overline{A_2}\overline{A_1}\overline{A_0} + A_2(A_1 + A_0)$   
 $Y_3 = \overline{A_3}\overline{A_2}\overline{A_1}\overline{A_0} + A_3(A_2 + A_1 + A_0)$



### ●乗算器

符号無し 4 ビット乗算器 ( $P=A \times B$ ) を加算器 4 個を直列接続したリプルキャリアダー(RCA) 3 組で構成した例<sup>(3)</sup>を示す。なお, 図中の OFF は定数 0 である。また, PLC は並列処理でないので桁上先見加算器(CLA)は使用していない。

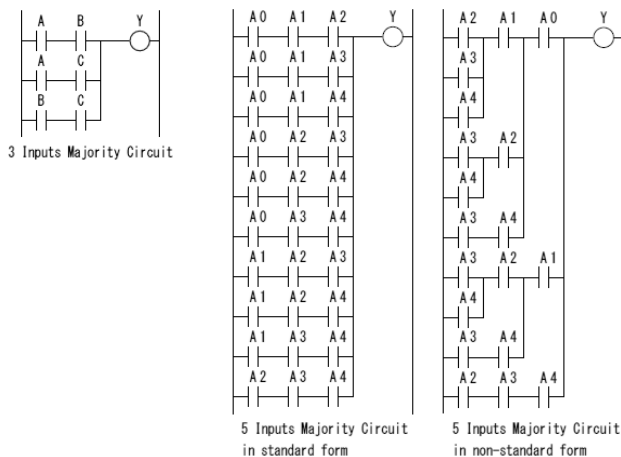


●多数決回路

A, B, C の 3 入力のうち 2 つ以上が ON のとき出力 Y が ON になる多数決回路を示す.

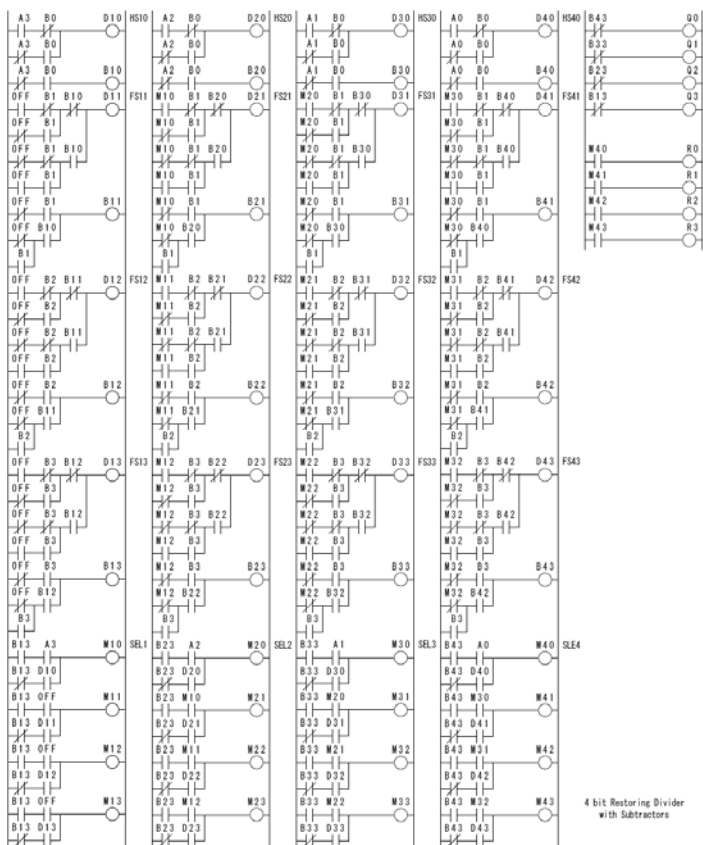
$$Y = AB + AC + BC + ABC = AB + AC + BC$$

入力の数  $n$  が奇数の場合, 項の数はまとめないと  $nC_{(n+1)/2}$  と増えてしまう. 積項の数は 5 個の入力 ( $A_0 \sim A_4$ ) の場合は 10 で, 7 個の入力 ( $A_0 \sim A_6$ ) の場合は 35 である. 5 入力の場合の標準形と編集した多数決回路を示す.



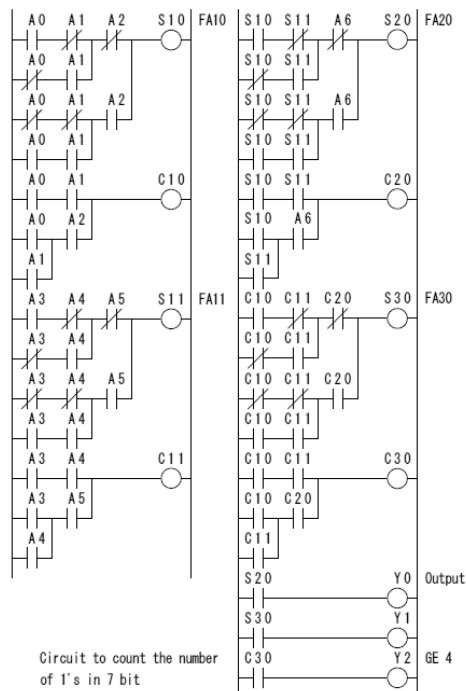
●除算器

符号無し 4 ビット除算 ( $A/B$  の商  $Q$ , 剰余  $R$ ) を回復法 (引き戻し法) で構成した除算器の例<sup>(3)</sup>を示す. 減算器 4 個を直列接続して 1 組とし, 4 組用いる. 被除数の上位 3 ビットに定数 0 (図中の OFF) を補い, 上位から順に減算しボローにより商を決定する. そのボローにより下位の演算に減算結果を用いるか減算する前の値を用いるかを選択する.



加算器を多段に接続して全てを加算することにより ON の数を数えてもよいが, 複雑になってしまう. 例えば, 7 個の入力 ( $A_0 \sim A_6$ ) のうち ON の数を数えて 2 進数で  $Y_0 \sim Y_2$  に出力する回路には図のようなものがある. 4 以上ならば過半数であるので,  $Y_2$  が ON ならば過半数である.

なお, 入力の数 が 偶数の場合は引き分けがあるが, 例えば半数より多いときに出力 Y を ON にするなどすればよい.

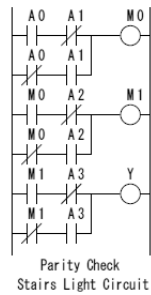


●パリティチェック, 階段灯回路

4ビットの入力(A<sub>0</sub>~A<sub>3</sub>)のうち1(ON)の個数が偶数か奇数かを検査するパリティチェック回路を示す. 排他的論理和を用いて出力Yを

$$Y = A_0 \oplus A_1 \oplus A_2 \oplus A_3$$

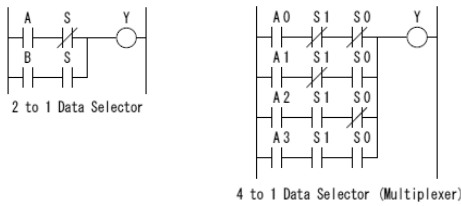
とすると, 入力のうち1の個数が奇数の時に出力が1になる. 図の回路ではこれを積和形式に展開せず一時メモリ M<sub>i</sub>を用いて多段接続しており, ビット数が増えても同様に拡張すればよい.



また, この回路は, 入力にトグルスイッチを用いると階段灯の回路のように, どのスイッチを操作しても出力を反転させることができる.

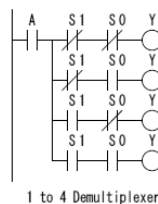
●選択器(セレクタ, マルチプレクサ)

A, BのどちらかをSにより選んでYに出力する回路, およびA<sub>0</sub>~A<sub>3</sub>の中から1つをS<sub>0</sub>, S<sub>1</sub>による2進コードにより選んでYに出力する回路を示す.



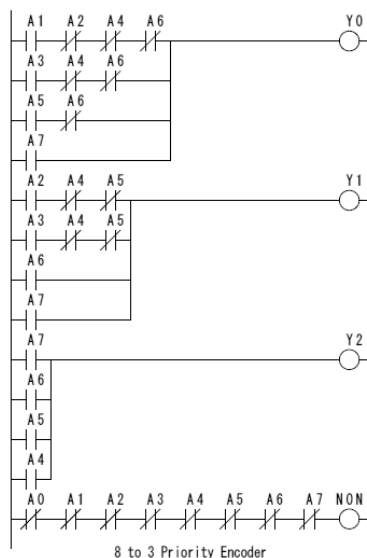
●分配器(デマルチプレクサ)

入力AをS<sub>0</sub>, S<sub>1</sub>で選ばれたY<sub>0</sub>~Y<sub>3</sub>のうちの1つに出力する回路を示す.



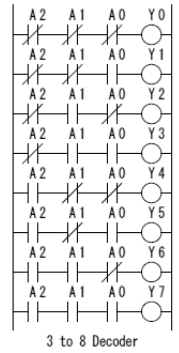
●符号器(エンコーダ)

ONしているA<sub>0</sub>~A<sub>7</sub>の入力に対してY<sub>0</sub>~Y<sub>2</sub>に2進コードを出力する回路を示す. 優先順位はA<sub>7</sub>>A<sub>6</sub>>...>A<sub>0</sub>で, 入力が全てOFFのときはN<sub>ON</sub>がONになる.



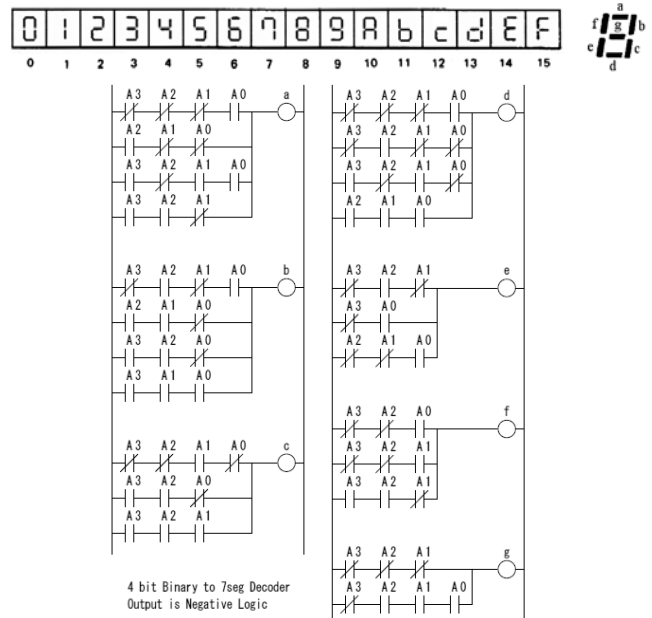
●復号器(デコーダ)

A<sub>0</sub>~A<sub>2</sub>による2進コードに対応するY<sub>0</sub>~Y<sub>7</sub>のうち1つをONにする回路を示す.



●7セグメント表示デコーダ

A<sub>0</sub>~A<sub>3</sub>による2進コードにより7セグメント表示器に対応する出力a~gをOFFにする回路を示す. 表示対応表を下に示す(M54405やDM9370<sup>(4)</sup>と異なり小文字の'c'としてある. 大文字の'C'とするにはどうすればよいかは各自考えてみよう). なお, TTL ICの設計を流用しているため出力は負論理だが, b接点を使用できるので問題ない.



●グレイコード(交番2進符号)

グレイコードは前後に隣接する符号間のハミング距離が必ず1, すなわち1ビットしか異ならないという特徴をもつ. 表に4ビットグレイコードを示す.

nビットの2進数をB, グレイコードをGとすると, 相互の変換は次のようになる.

・2進数(バイナリ)からグレイコードへの変換

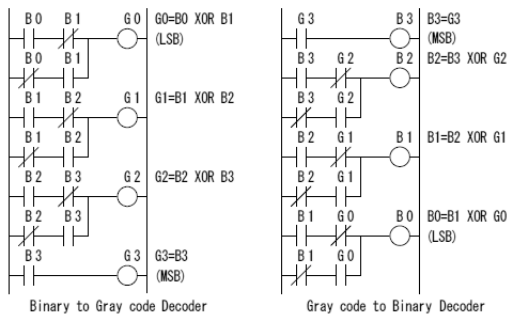
$$G_i = B_i \oplus B_{i+1} \quad (i=0, \dots, n-2)$$

$$G_{n-1} = B_{n-1} \quad (\text{MSB})$$

・グレイコードから2進数への変換

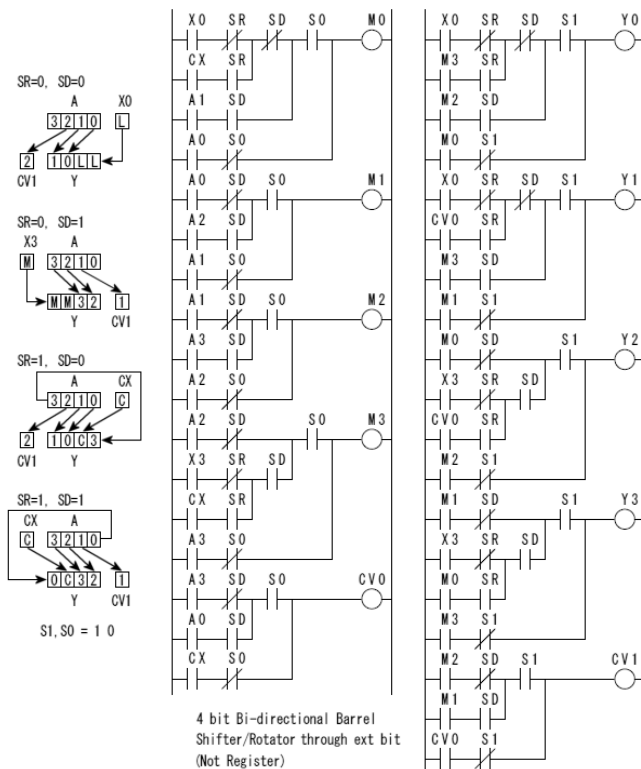
$$B_{n-1} = G_{n-1} \quad (\text{MSB})$$

$$B_i = B_{i+1} \oplus G_i \quad (i = n-2, \dots, 0)$$



●シフト・ローテータ

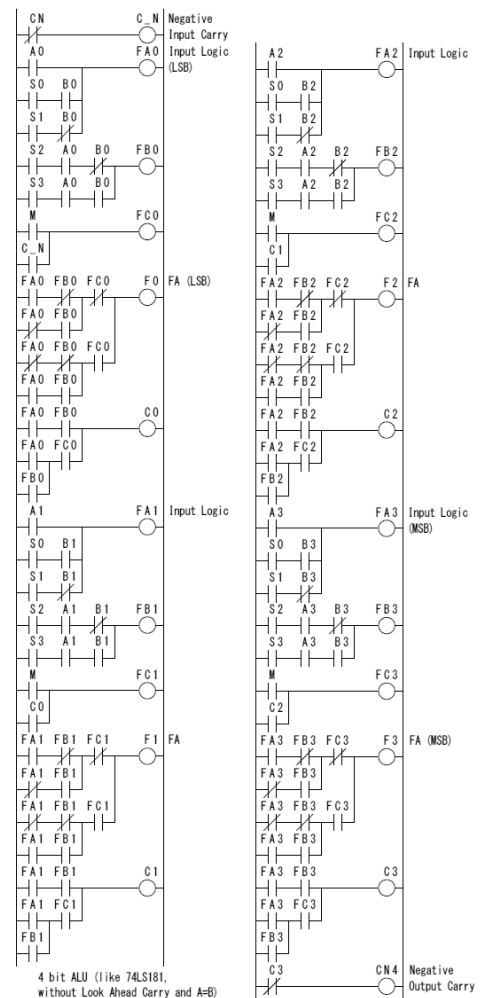
$2^i$  ビットシフトしたデータと元のデータを  $S_i$  で選択する回路を多段に接続することによりバレルシフトを構成することができる。4ビット双方向シフト/ローテータを示す。入力  $A_0 \sim A_3$  を  $S_0, S_1$  による2進数で指定されたビット数分シフトまたはローテートして  $Y_0 \sim Y_3$  に出力する。 $S_R$  が OFF のときは空いたビットが  $X_0$  または  $X_3$  になるシフトであり、 $S_R$  が ON のときは外部ビット  $C_X$  を含めて溢れたビットが空いたビットに入る5ビットのローテートになる。 $C_{V1}$  は最後に溢れたビットである。 $S_D$  が ON のとき MSB から LSB 方向に、 $S_D$  が OFF のとき LSB から MSB 方向にシフトまたはローテートする。



●算術・論理演算器 (ALU)

算術演算器の出力と論理演算器の出力をセクタで選択すれば算術・論理演算器 (ALU) が構成できる。また、全加算器 (FA) への入力論理を変更することにより ALU を構成することもできる。

74181<sup>(4)</sup> に似た演算機能を持つ4ビット ALU を後者の方法で構成した例を示す。PLC は並列処理でないので、ここでも桁上先見加算器 (CLA) は使用していない。ALU への4ビット2進数の入力を  $A$  と  $B$ 、キャリ入力を  $C_n$  とし、ALU の4ビットの出力を  $F$ 、74181 のキャリ出力  $C_{n+4}$  に対応するキャリを  $C_{n4}$  とする。選択信号  $S_0 \sim S_3$  および  $M$  に対する正論理の演算機能選択表を示す。なお、キャリは論理が逆なので注意すること。これらの演算機能を実現するために使用した FA の入力 ( $F_A, F_B$  およびキャリ  $F_C$ ) の論理関数を以下に示す。論理演算のキャリ出力は無意味であり 74181 との互換性はない。また、先見キャリ  $G, P$  および  $A=B$  の出力は省略した。その他の ALU の例をこの付録末の補足 1 に示す。



$$F_A = A + S_0B + S_1\bar{B}$$

$$F_B = S_2\bar{A}\bar{B} + S_3AB$$

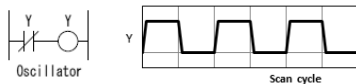
$$F_C = M + C$$

$S_3 \sim S_0$	M=1 (Logic)	M=0 (Arithmetic) $C_n=1$ (no carry)	M=0 (Arithmetic) $C_n=0$ (with carry)
0000	$\bar{A}$	A	A plus 1
0001	$A + B$	A + B	(A + B) plus 1
0010	$\bar{A}\bar{B}$	A + $\bar{B}$	(A + $\bar{B}$ ) plus 1
0011	0	minus 1	0
0100	$\bar{A}B$	A plus $\bar{A}\bar{B}$	A plus $\bar{A}\bar{B}$ plus 1
0101	$\bar{B}$	(A + B) plus $\bar{A}\bar{B}$	(A + B) plus $\bar{A}\bar{B}$ plus 1
0110	$A \oplus B$	A minus B minus 1	A minus B
0111	$\bar{A}\bar{B}$	$\bar{A}\bar{B}$ minus 1	$\bar{A}\bar{B}$
1000	$\bar{A} + B$	A plus AB	A plus AB plus 1
1001	$A \oplus \bar{B}$	A plus B	A plus B plus 1
1010	B	(A + $\bar{B}$ ) plus AB	(A + $\bar{B}$ ) plus AB plus 1
1011	AB	AB minus 1	AB
1100	1	A plus A (shift left)	A plus A plus 1
1101	$A + \bar{B}$	(A + B) plus A	(A + B) plus A plus 1
1110	$A + B$	(A + $\bar{B}$ ) plus A	(A + $\bar{B}$ ) plus A plus 1
1111	A	A minus 1	A

### ●発振器

出力の否定を入力にフィードバックすると、1 スキャンの間 ON, 次の1 スキャンの間 OFF を繰り返す発振器となる。ラダー図とタイミングチャートを示す。

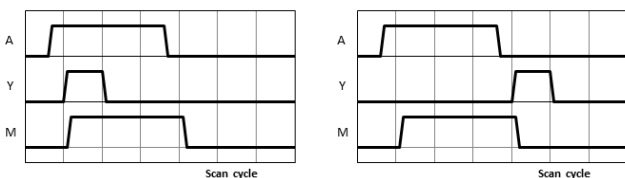
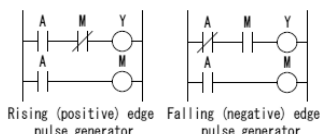
スキャンタイムが一定ならば、後に示すカウンタと組み合わせてタイマが構成できる。



### ●立上がり, 立下がりパルス出力

入力 A の立上がり時(OFF から ON への変化時)または立下がり時(ON から OFF への変化時)に1 スキャンのみ ON するパルスを Y に出力する。PLC によっては微分と称しているものもある。

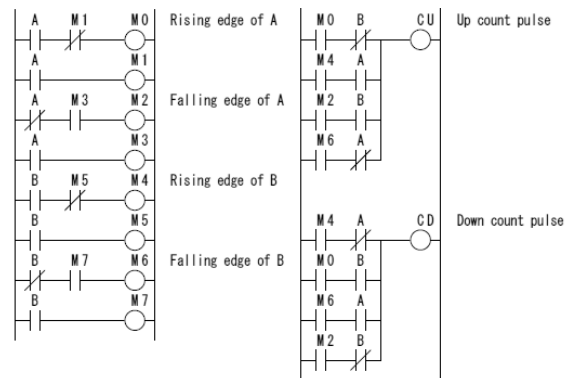
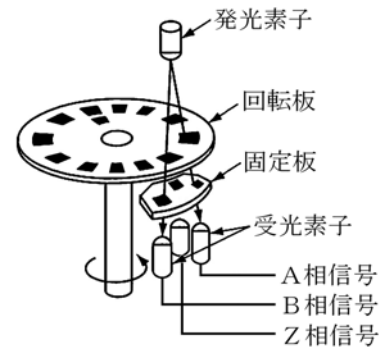
ラダー図とタイミングチャートを示す。なお、プログラムの順序を入れ替えると機能しない。



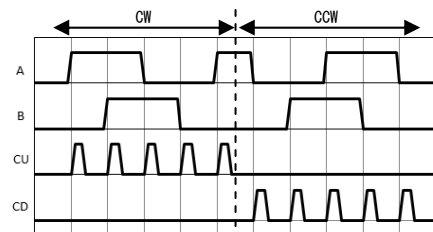
後述の新入力優先回路のようにスイッチが長押しされると困る場合や、カウンタのクロック入力のようにスイッチを押した時、あるいは押して放した時に動作する回路に用いる。

### ●ロータリエンコーダの方向弁別回路

回転角度を検出するためのインクリメンタル型光学式ロータリエンコーダは、90° 位相差を持つ A 相, B 相のパルス信号を発生する。図の4 通倍回路は、A 相, B 相信号の立上がりおよび立下がりパルスを利用することにより回転方向を弁別し、タイミングチャートに示すようにアップカウントパルス  $C_U$  とダウンカウントパルス  $C_D$  を出力する。この出力  $C_U$  と  $C_D$  を後述のアップダウンカウンタのクロック入力に接続して計数することができる。1 回転に1 パルス出力する Z 相信号は基準位置としてカウンタのクリア入力に接続したり、多回転の計数に使用したりできる。



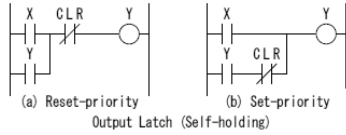
Quadrature Detector for Incremental Rotary Encoder



## 2. 順序回路

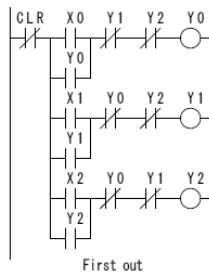
### ●自己保持回路

順序回路を構成するためには記憶回路が必要である。基本的な記憶回路として自己保持回路がある。(a)はリセット優先, (b)はセット優先の自己保持回路である。

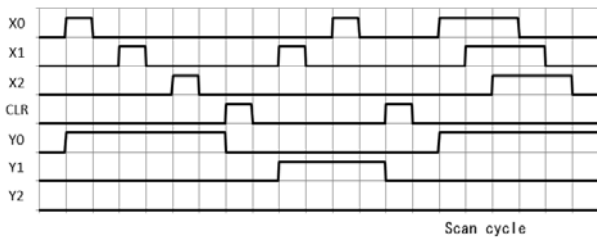


### ●並列優先回路(先行優先回路)

リセット優先自己保持回路を利用したインターロック回路による3入力( $X_0 \sim X_2$ )の並列優先回路を示す。これは早押しクイズのように最も先に ON にされたスイッチに対応する出力を ON にし, 後から別のスイッチを押しても出力は変わらない回路である。

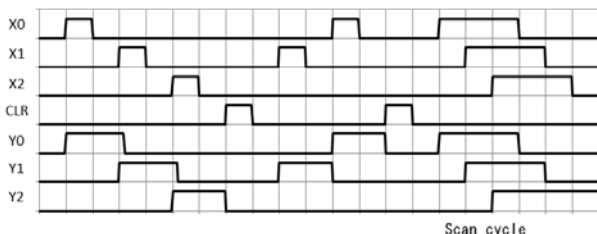
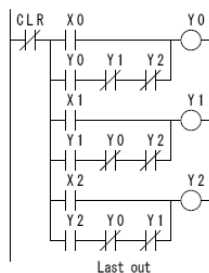


なお, 図の回路ではリセット用の入力 CLR も設けてある。タイミングチャートを下に示す。



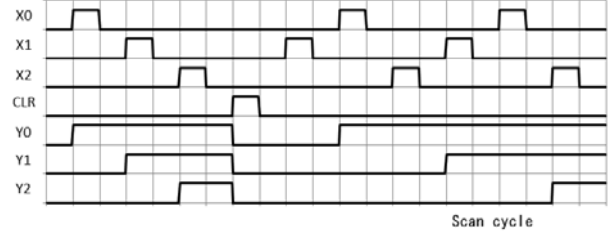
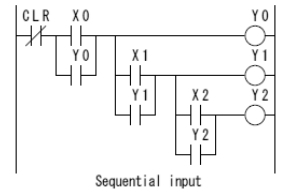
### ●新入力優先回路(後押し優先回路)

図の回路はセット優先自己保持回路を応用した3入力( $X_0 \sim X_2$ )の新入力優先回路で, 後から ON にされたスイッチが優先される回路である。ただし, 複数のスイッチを ON にすると, それに対応する出力が全て ON になってしまい, 期待する動作をしないので, ON 状態が重ならないように操作するか, スwitchの立上がりパルスを使用する必要がある(この付録末の補足2参照)。



### ●直列優先回路

左側の入力側制御母線に近い回路の優先が最も高く, 左側のスイッチから順に ON しないと動作しない回路である。



以下の順序回路はクロック入力 CLK にしたがって動作する。基本的にクロックはパルスであり, 1 スキャン以上 ON だと発振することがある。以降, 現在の状態を  $Q_n$ , 次の状態を  $Q_{n+1}$  と表すことにする。

### ●RS フリップフロップ(RS-FF)

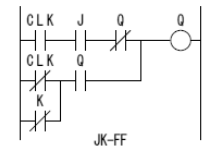
禁止入力があり PLC は並列処理でないのたすき掛けのフィードバックはそのままではうまく動作せず余分のメモリを必要とするため, ここでは省略する。自己保持回路で代用することができる。

S	R	$Q_{n+1}$	動作
0	0	$Q_n$	保持
0	1	0	リセット
1	0	1	セット
1	1	—	禁止

### ●JK フリップフロップ(JK-FF)

RS-FF と違い, 2つの入力を同時に 1 にすることができる。その場合には出力が反転する。

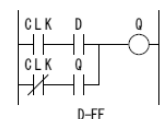
J	K	$Q_{n+1}$	動作
0	0	$Q_n$	保持
0	1	0	リセット
1	0	1	セット
1	1	$\bar{Q}_n$	反転



### ●D フリップフロップ(D-FF)

入力 D のデータを取り込む。SET, RESET 命令があれば, それとほぼ同じ。JK-FF で  $J = D$ ,  $K = \bar{D}$  とすればよいが, 図のように簡略化できる。

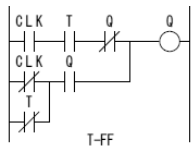
D	$Q_{n+1}$	動作
0	0	リセット
1	1	セット



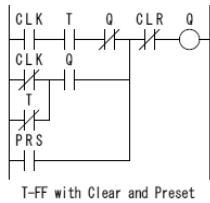
●T フリップフロップ (T-FF)

トリガ入力 T が ON のときクロックが入るたびに出力が反転するトグル動作をする。JK-FF で  $J=K=T$  とする。

T	$Q_{n+1}$	動作
0	$Q_n$	保持
1	$\bar{Q}_n$	反転

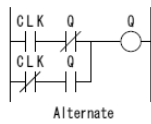


初期化のためのクリア CLR およびプリセット PRS 入力付き T-FF の例を示す。他の FF でも同様に構成することができる。



●オルタネイト

クロックが入るたびに出力が反転する。T-FF で  $T=1$  (ON) とするとオルタネイトになる。



●パルスを用いないオルタネイト

入力 X が ON するたびに出力が反転する回路を下に示す。入力はパルスでなくてもよいが、立上がり、立下がりパルス出力回路と同じく、確実に動作させるためには少なくとも1スキャンサイクルの間は ON を保持しなければならない。タイミングチャートと真理値表を示す。

$M_{n+1}$  の計算に  $Q_n$  ではなく  $Q_{n+1}$  が使われているが、問題ないことが真理値表から確認できる。(\*印と+印部分)

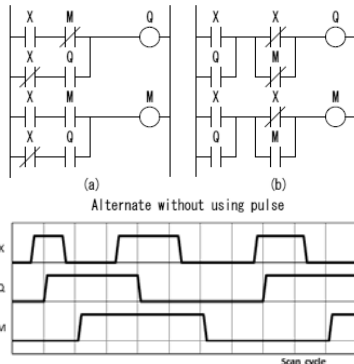
なお、

$$(X + Q) \cdot (\bar{X} + \bar{M}) = X \cdot \bar{M} + \bar{X} \cdot Q$$

$$(X + Q) \cdot (\bar{X} + M) = X \cdot M + \bar{X} \cdot \bar{Q}$$

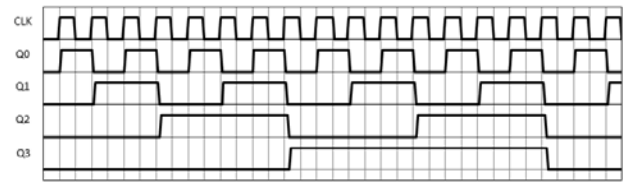
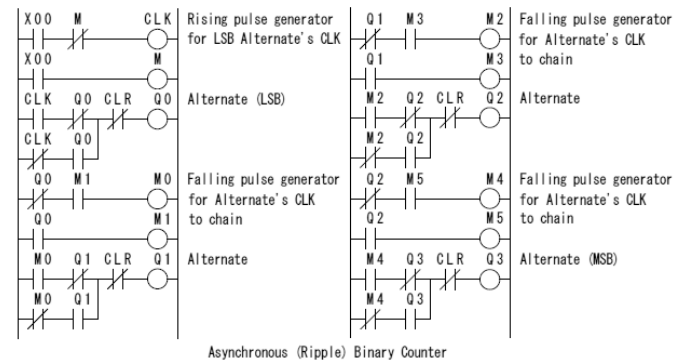
であるから、図(a)と(b)の機能は同等である。

X	$Q_n, M_n$	$Q_{n+1}, M_{n+1}$
0	0 0	0 0
1	0 0	1 0*
1	1 0	1 0+
0	1 0	1 1
0	1 1	1 1
1	1 1	0 1*
1	0 1	0 1+
0	0 1	0 0



●非同期式カウンタ

例として4ビットの $2^4$ 進リプルカウンタ回路を示す。入力 X00 の立上がりパルスを LSB のオルタネイトのクロックへの入力とする。そのオルタネイトの出力の立下がりパルスを次のオルタネイトのクロックへの入力とする。このように MSB まで順次接続することにより、 $2^n$  進アップカウンタが構成できる。なお、立下がりパルスでなく立上がりパルスを使うとダウンカウンタとなる。以降、 $2^n$  進カウンタをバイナリカウンタと呼ぶ。



●10 進アップダウンカウンタ

アップカウンタクロック入力  $C_U$  とダウンカウンタクロック入力  $C_D$  を持つ 10 進アップダウンカウンタの構成例を示す。N 進アップカウンタの構成法の 1 つである  $N-1$  を検出して次のクロックが入ったときにカウンタをリセットする方法を用いる。順序回路の構成方法に従って状態遷移回路を構成し、各 FF の入力に接続する。カウンタは T-FF で構成した。図中の  $M_1 \sim M_3$  が状態遷移用の各 FF の入力である。アップカウンタの場合、ビット n の T-FF への入力は、

$$T_n = Q_0 Q_1 \cdots Q_{n-1} = Q_{n-1} T_{n-1} \quad (n \geq 1), \quad T_0 = 1$$

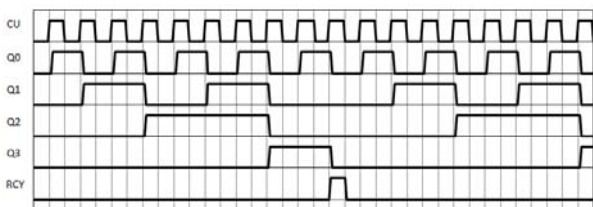
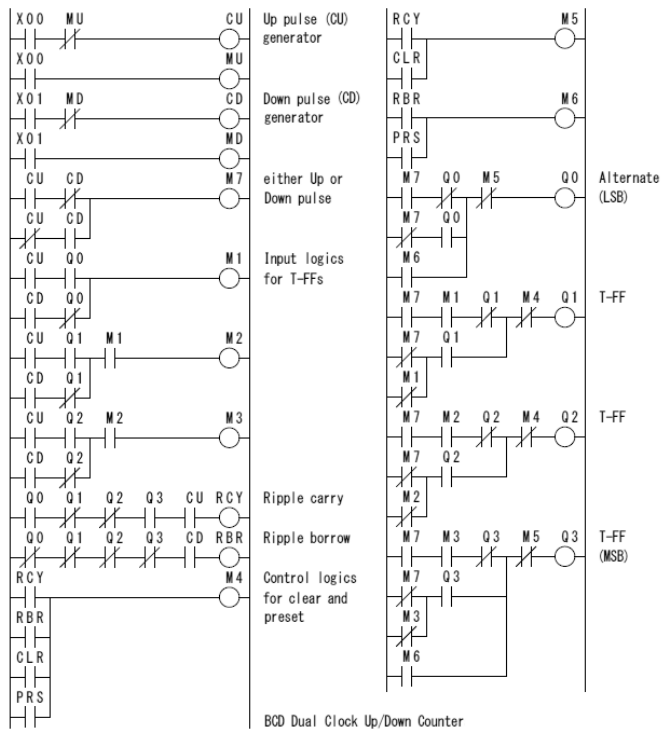
であり、ダウンカウンタの場合は、

$$T_n = \bar{Q}_0 \bar{Q}_1 \cdots \bar{Q}_{n-1} = \bar{Q}_{n-1} T_{n-1} \quad (n \geq 1), \quad T_0 = 1$$

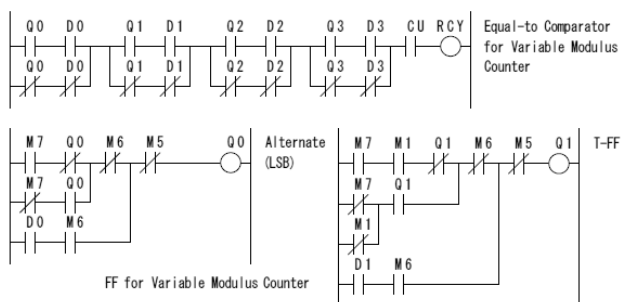
である。これらを  $C_U$  と  $C_D$  で選択すればよい。なお、LSB ( $n=0$ ) は T-FF の入力を 1 としたオルタネイトである。10 進カウンタなので、アップカウンタ時には一致検出器で 9 (2 進数で 1001) を検出し、次の  $C_U$  でリプルキャリ  $R_{CY}$  に出力する。この  $R_{CY}$  を用いて FF をクリアする。ダウンカウント時には



0を検出して、次の  $C_D$  でリプルボロー  $R_{BR}$  を出力し、FFに  $N-1$  をプリセットする。  $R_{CY}$  と  $R_{BR}$  を次の桁のカウンタの  $C_U$  と  $C_D$  に接続することにより多段のカウンタが構成できる。なお、初期化は  $CLR$  または  $PRS$  で行う。アップカウント時のタイミングチャートを示す。



一致検出器を定数でなくスイッチ入力との一致検出器に置き換え、その入力を  $R_{BR}$  で FF にロードする回路にプリセットを変更すると可変カウンタとなる。変更例を下図に示す。なお、  $Q_2$ 、  $Q_3$  は  $Q_1$  と同様に変更する。



### ●同期式カウンタ

PLC ではプログラムが順次実行されるので同期式とはならないが、デジタル回路の設計で得られた FF への入力を用いてカウンタを構成することはできる。付表にいくつかのカウンタの入力例を示す。ただし、表記の簡略化のため  $Q_0, Q_1, \dots$  を  $A, B, \dots$  とし、  $A$  の否定  $\bar{A}$  は小文字で  $a$  と表記している。なお、範囲外からの補正はしていない。

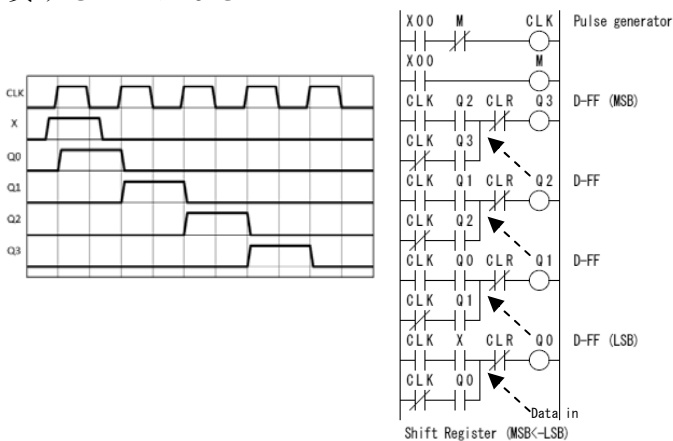
例えば、T-FF で 10 進カウンタを構成するときの T-FF への入力は次のようになる。  $T=1$  の場合はオルタネイトである。これらのラダー図は省略する。

10 進アップカウンタ	10 進ダウンカウンタ
$T_0 = 1$	$T_0 = 1$
$T_1 = \bar{Q}_3 Q_0$	$T_1 = (Q_1 + Q_2 + Q_3) \bar{Q}_0$
$T_2 = Q_1 Q_0$	$T_2 = (Q_2 \bar{Q}_1 + Q_3) \bar{Q}_0$
$T_3 = (Q_2 Q_1 + Q_3) Q_0$	$T_3 = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$

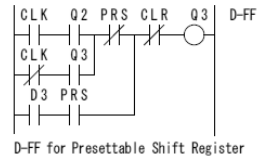
### ●MSB←LSB シフトレジスタ (左シフトレジスタ)

シフトパルスが入ると LSB から MSB 方向にデータがシフトし、新しいデータ  $X$  が LSB に入る 4 ビット左シフトレジスタを D-FF で構成した例を示す。これは直列データを並列データに変換する回路としても利用される。

ラダー図中の破線の矢印で示すように MSB から降順に下位ビットのデータを D-FF に取り込んでいる。逆に LSB から昇順に下位ビットのデータを D-FF に取り込むようにするとデータが筒抜けになってしまう。次の双方向シフトレジスタのように状態遷移回路を構成すればよいが、内部メモリを浪費することになる。



シフトレジスタに値を設定するために各 D-FF に図のように並列プリセットを追加することもできる。並列に値を設定しシフトパルスを加えると Q<sub>3</sub> からセットされた値が直列に出る。これは並列データを直列データに変換する直並列変換器としても利用される。

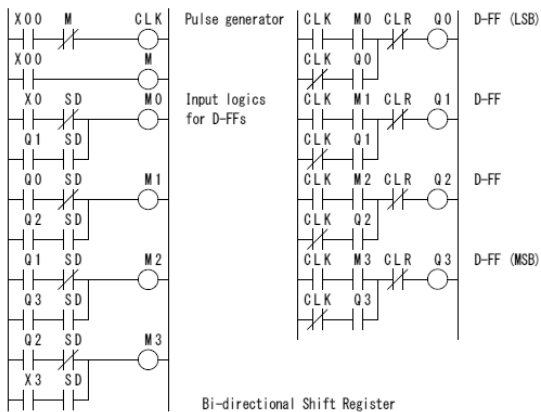


同様に MSB から LSB 方向にシフトする右シフトレジスタを構成することができる。この場合は LSB から昇順に上位ビットのデータを D-FF に取り込むように構成する。

### ● 双方向シフトレジスタ

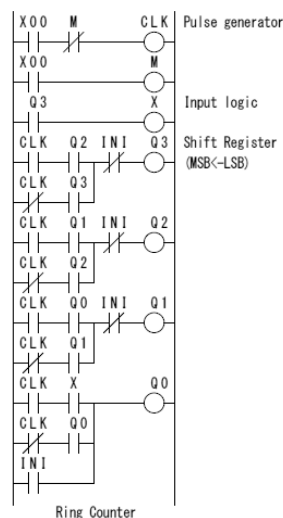
S<sub>D</sub> が ON のときシフトパルスが入ると MSB から LSB 方向に、S<sub>D</sub> が OFF のときシフトパルスが入ると LSB から MSB 方向にシフトする 4 ビット双方向シフトレジスタの例を示す。X<sub>0</sub> と X<sub>3</sub> は新しいデータである。

また、S<sub>D</sub> とシフトパルスではなく左シフトと右シフトのための 2 つのシフトパルス入力を持つ双方向シフトレジスタも 10 進アップダウンカウンタと同様に構成することができる。

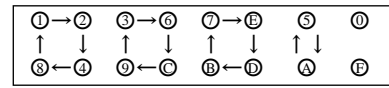


### ● リングカウンタ

シフトレジスタの最終段の出力を初段の入力に接続する。ただ一つの 1 が FF 上を回るようにすると n 個の FF で n まで計数できる。n = 4 で初段への入力を X = Q<sub>3</sub> とした場合のラダー図と状態を 16 進数表記し、入力が 1 のときの状態遷移

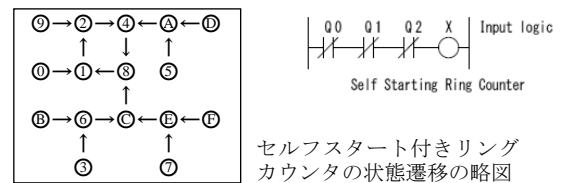


を矢印で表し、入力が 0 のときの保持を省略した状態遷移の略図を示す。状態遷移の略図中の状態 0 と F は単独で安定であり、いくつかの閉ループがある。ラダー図中の INI は Q<sub>0</sub> のみを 1 とし、他を 0、すなわち状態 1 にする初期化入力である。



リングカウンタの状態遷移の略図

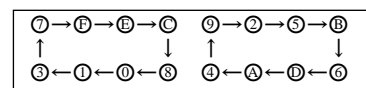
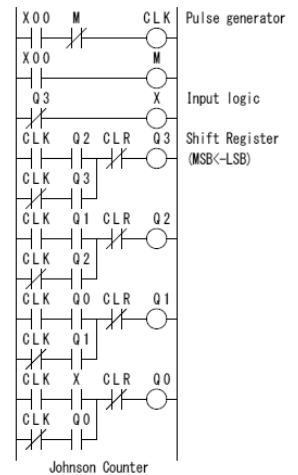
また、CLK が数回入力されたとき、ただ一つの FF のみ 1 にするためのセルフスタート回路として、初段への入力を  $X = \bar{Q}_0\bar{Q}_1\bar{Q}_2$  としたときのラダー図の変更部分と状態遷移の略図を示す。



セルフスタート付きリングカウンタの状態遷移の略図

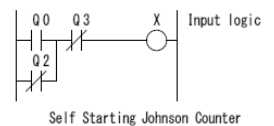
### ● ジョンソンカウンタ

シフトレジスタの最終段の出力の否定を初段の入力に接続する。n 個の FF で 2n まで計数でき、50% デューティのオーバーラップした出力が得られる。n = 4 で初段への入力を  $X = \bar{Q}_3$  とした場合のラダー図と状態遷移の略図を示す。状態遷移図には 2 つの閉ループがある。ラダー図中の CLR は全て 0 (状態 0) にする入力である。

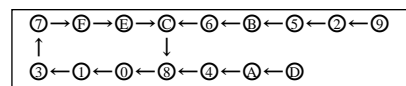


ジョンソンカウンタの状態遷移の略図

また、セルフスタート回路として初段への入力を  $X = (Q_0 + \bar{Q}_2)\bar{Q}_3$  としたときのラダー図の変更部分と状態遷移の略図を示す。



セルフスタート付きジョンソンカウンタの状態遷移の略図



セルフスタート付きジョンソンカウンタの状態遷移の略図

●グレイコードカウンタ

グレイコードを出力するカウンタのための各 FF への入力を一般的な手法で求めると複雑になってしまう。例えば、4ビットのグレイコードカウンタを D-FF で構成した場合の入力は XOR を使用しないと次のようになる。

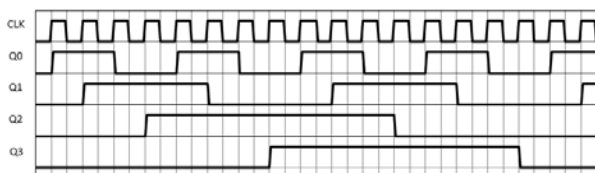
$$D_0 = Q_1(Q_2\bar{Q}_3 + \bar{Q}_2Q_3) + \bar{Q}_1(Q_2Q_3 + \bar{Q}_2\bar{Q}_3)$$

$$D_1 = \bar{Q}_0Q_1 + Q_0(Q_2Q_3 + \bar{Q}_2\bar{Q}_3)$$

$$D_2 = \bar{Q}_0Q_1\bar{Q}_3 + Q_2(Q_0 + \bar{Q}_1)$$

$$D_3 = \bar{Q}_0\bar{Q}_1Q_2 + Q_3(Q_0 + Q_1)$$

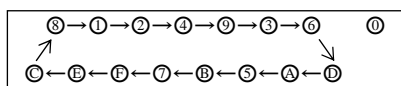
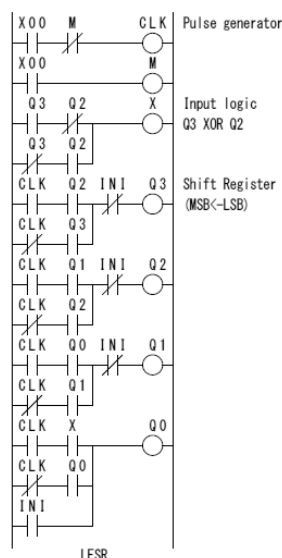
なお、このラダー図は省略するが、タイミングチャートを示す。(p.4 のグレイコード表を参照のこと)



グレイコードカウンタの構成法としては、バイナリカウンタの出力に2進数からグレイコードへの変換器を付ける方法や、一つ余分に FF を使用するが入力回路を単純化できる方法などもある。

●リニアフィードバック シフトレジスタ (LFSR)

シフトレジスタの出力の XOR をとって初段の入力に接続する。n 個の FF で最大  $2^n - 1$  まで計数できる。n=4 で初段への入力を  $X = Q_2 \oplus Q_3$  とした場合のラダー図と状態遷移の略図を示す。状態遷移の略図には単独の安定状態 0 とそれ以外の 15 個の状態からなる 1 つの閉ループがある。



LFSR の状態遷移の略図

●ドラムシーケンサ

カウンタの各ビットの出力が得られるためドラムシーケンサのような使い方もできる。

一列に並べた 5 個の LED を表のように順次点灯させる例を取り扱う。この例では状態の数は 10 なので、前述の 10 進カウンタを用いる。4ビットのカウンタの出力  $Q_0 \sim Q_3$  を  $Y_0 \sim Y_4$  の 5 個の LED へ変換するデコーダは次のように構成することができる。ラダー図はデコーダ部分のみを示す。10 進カウンタの後にこのデコーダを接続する。

No.	$Q_3 \sim Q_0$	$Y_4 \sim Y_0$
0	0000	00000
1	0001	00001
2	0010	00010
3	0011	00100
4	0100	01000
5	0101	10001
6	0110	01010
7	0111	00100
8	1000	01010
9	1001	10001

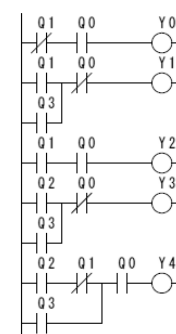
$$Y_0 = \bar{Q}_1Q_0$$

$$Y_1 = (Q_1 + Q_3)\bar{Q}_0$$

$$Y_2 = Q_1Q_0$$

$$Y_3 = (Q_2 + Q_3)\bar{Q}_0$$

$$Y_4 = (Q_2\bar{Q}_1 + Q_3)Q_0$$



Seq. Decoder for BCD Counter

また、状態 1 つにつき 1 ビットを割り当てた 10 ビットのリングカウンタを用いることもできる。その出力  $Q_0 \sim Q_9$  を  $Y_0 \sim Y_4$  の 5 個の LED へ変換するデコーダは次のように構成することができる。カウンタのビット数は増えるが、デコーダの論理関数はわかりやすい。

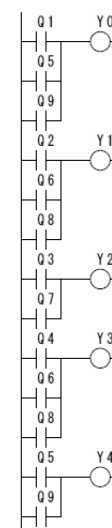
$$Y_0 = Q_1 + Q_5 + Q_9$$

$$Y_1 = Q_2 + Q_6 + Q_8$$

$$Y_2 = Q_3 + Q_7$$

$$Y_3 = Q_4 + Q_6 + Q_8$$

$$Y_4 = Q_5 + Q_9$$



Seq. Decoder for Ring Counter

●その他

正確なクロックパルスが得られれば、それをカウンタに入力することによりタイマを構成することができる。タイマの各ビットの値と指定された値を比較して自己保持すればオンディレイタイマが構成される。同様にオフディレイタイマ、フリッカ、ワンショットなどの回路も構成することができる。(詳しくは参考書等<sup>(5)~(7)</sup>を参照のこと)

付表 同期式カウンタの構成例(イリーガルステートを補正するセルフスタートは考慮していない)

		Synchronous Counter (A,B,C,D,E,F,G = Q0,Q1,Q2,Q3,Q4,Q5,Q6)									
Modulo	Input	Up Counter					Down Counter				
		D	T	J	K	D	T	J	K		
2	10	a	1	1	1	a	1	1	1		
3	10	ba	b	b	1	B	A+B	B	1		
	11	A	A+B	A	1	ba	a	a	1		
4	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
5	10	ca	c	c	1	Ba+C	A+B+C	B+C	1		
	11	bA+Ba	A	A	A	C+BA	Ba+C	C	a		
	12	BA	C+BA	BA	1	cba	ba	ba	1		
6	10	a	1	1	1	a	1	1	1		
	11	cbA+Ba	cA	cA	A	Ca+BA	Ba+Ca	Ca	a		
	12	Ca+BA	BA+CA	BA	A	cba+CA	ba	ba	a		
7	10	ca+ba	c+b	c+b	1	Ba+Ca	A+B+C	B+C	1		
	11	bA+cBa	A+CB	A	A+C	ba+BA	a	a	a		
	12	Cb+BA	BA+CB	BA	B	cba+CA+CB	ba	ba	ba		
8	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
	12	cBA+Cb+Ca	BA	BA	BA	cba+CA+CB	ba	ba	ba		
9	10	da	d	d	1	Ba+Ca+D	A+B+C+D	B+C+D	1		
	11	bA+Ba	A	A	A	Cba+D+BA	Ba+Ca+D	Ca+D	a		
	12	cBA+Cb+Ca	BA	BA	BA	D+CA+CB	Cba+D	D	ba		
	13	CBA	D+CBA	CBA	1	dcba	cba	cba	1		
10	10	a	1	1	1	a	1	1	1		
	11	dbA+Ba	dA	dA	A	Cba+Da+BA	Ba+Ca+Da	Ca+Da	a		
	12	cBA+Cb+Ca	BA	BA	BA	Da+CA+CB	Cba+Da	Da	ba		
	13	Da+CBA	DA+CBA	CBA	A	dcba+DA	cba	cba	a		
11	10	da+ba	d+b	d+b	1	Ba+Ca+Da	A+B+C+D	B+C+D	1		
	11	bA+dBa	A+DB	A	A+D	ba+BA	a	a	a		
	12	cBA+dCb+Ca	BA	BA	BA	Dba+CA+CB	Cba+Dba	Dba	ba		
	13	Db+CBA	DB+CBA	CBA	B	dcba+DA+DB	cba	cba	ba		
12	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
	12	dcBA+Cb+Ca	dBA	dBA	BA	Dba+CA+CB	Cba+Dba	Dba	ba		
	13	Db+Da+CBA	CBA+DBA	CBA	BA	dcba+DA+DB	cba	cba	ba		
13	10	da+ca	d+c	d+c	1	Ba+Ca+Da	A+B+C+D	B+C+D	1		
	11	bA+Ba	A	A	A	Cba+Dba+BA	Ba+Ca+Da	Ca+Da	a		
	12	cBA+dCa+dCb or cBA+dCa+CbA or cBA+dCb+Cba	BA+DC	BA	D+BA	cba+CA+CB	ba	ba	ba		
	13	Dc+CBA	DC+CBA	CBA	C	dcba+DA+DB+DC	cba	cba	cba		
14	10	a	1	1	1	a	1	1	1		
	11	Ba+dbA+cbA	dA+cA	dA+cA	A	Cba+Dba+BA	Ba+Ca+Da	Ca+Da	a		
	12	cBA+dCb+Ca	BA+DCA	BA	BA+DA	cba+CA+CB	ba	ba	ba		
	13	Dc+Da+CBA	CBA+DCA	CBA	CA	dcba+DA+DB+DC	cba	cba	cba		
15	10	da+ca+ba	d+c+b	d+c+b	1	Ba+Ca+Da	A+B+C+D	B+C+D	1		
	11	bA+dBa+cBa	A+DCB	A	A+DC	ba+BA	a	a	a		
	12	cBA+Cb+dCa	BA+DCB	BA	BA+DB	cba+CA+CB	ba	ba	ba		
	13	Dc+Db+CBA	CBA+DCB	CBA	CB	dcba+DA+DB+DC	cba	cba	cba		
16	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
	12	cBA+Cb+Ca	BA	BA	BA	cba+CA+CB	ba	ba	ba		
	13	dCBA+Dc+Db+Da	CBA	CBA	CBA	dcba+DA+DB+DC	cba	cba	cba		
20	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
	12	ecBA+Cb+Ca	eBA	eBA	BA	Dcba+Eba+CA+CB	Cba+Dba+Eba	Dba+Eba	ba		
	13	dCBA+Dc+Db+Da	CBA	CBA	CBA	Eba+DA+DB+DC	Dcba+Eba	Eba	cba		
50	10	a	1	1	1	a	1	1	1		
	11	Ba+fbA+ebA	fA+eA	fA+eA	A	Cba+Dba+Eba+Fba+BA	Ba+Ca+Da+Ea+Fa	Ca+Da+Ea+Fa	a		
	12	cBA+Cb+Ca	BA	BA	BA	Dcba+Ecba+Fcba+CA+CB	Cba+Dba+Eba+Fba	Dba+Eba+Fba	ba		
	13	dCBA+Dc+Db+Da	CBA	CBA	CBA	Edcba+Fdcba+DA+DB+DC	Dcba+Ecba+Fcba	Ecba+Fcba	cba		
	14	Ea+eDCBA+fEb+fEc+fEd 他	FEA+DCBA	DCBA	FA+DCBA	edcba+EA+EB+EC+ED	dcba	dcba	dcba		
15	Fe+Fa+EDCBA	FEA+EDCBA	EDCBA	EA	fedcba+FA+FB+FC+FD+FE	edcba	edcba	edcba			
60	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
	12	Cb+Ca+fcBA+ecBA+dcBA	fBA+eBA+dBA	fBA+eBA+dBA	BA	Dcba+Ecba+Fcba+CA+CB	Cba+Dba+Eba+Fba	Dba+Eba+Fba	ba		
	13	dCBA+Db+Da+fDc+eDc	CBA+FEDBA	CBA	CBA+FEBA	dcba+DA+DB+DC	cba	cba	cba		
	14	eDCBA+Ed+fEc+Eb+Ea	DCBA+FEDBA	DCBA	DCBA+FDDBA	edcba+EA+EB+EC+ED	dcba	dcba	dcba		
	15	Fe+Fd+Fb+Fa+EDCBA	EDCBA+FEDBA	EDCBA	EDBA	fedcba+FA+FB+FC+FD+FE	edcba	edcba	edcba		
100	10	a	1	1	1	a	1	1	1		
	11	bA+Ba	A	A	A	ba+BA	a	a	a		
	12	Cb+Ca+gcBA+fcBA	gBA+fBA	gBA+fBA	BA	Dcba+Ecba+Fcba+Gcba+CA+CB	Cba+Dba+Eba+Fba+Gba	Dba+Eba+Fba+Gba	ba		
	13	dCBA+Dc+Db+Da	CBA	CBA	CBA	Edcba+Fdcba+Gdcba+DA+DB+DC	Dcba+Ecba+Fcba+Gcba	Ecba+Fcba+Gcba	cba		
	14	eDCBA+Ed+fEc+Eb+Ea	DCBA	DCBA	DCBA	Fedcba+Gedcba+EA+EB+EC+ED	edcba	edcba	edcba		
	15	Fb+Fa+fEDCBA+gFc+gFd+gFe 他	GFBA+EDCBA	EDCBA	GBA+EDCBA	fedcba+FA+FB+FC+FD+FE	edcba	edcba	edcba		
16	Gf+Gb+Ga+FEDCBA	GFBA+FEDCBA	FEDCBA	FBA	gfedcba+GA+GB+GC+GD+GE+GF	fedcba	fedcba	fedcba			

補足1. 4ビット ALU の例

ALU への2進数の入力を A と B, キャリ入力を  $C_n$  とし, ALU の出力を F, キャリ出力  $C_{n+4}$  に対応するキャリを  $C_3$ , オーバーフローを V とする. S を ALU の演算機能選択信号とする. 演算機能を実現するための FA への入力論理は種々あるが, 使用した FA への入力  $F_A, F_B$  とキャリ入力  $F_C$  の論理関数を示す. なお, ボローはキャリの否定であり, 論理演算のキャリ出力は無意味であり互換性はない.

・74LS382<sup>(4)</sup>の ALU 部に似た演算機能を持つ4ビット ALU の例を示す.

$$F_A = S_{A0}A + S_{A1}\bar{A} + S_{A2}B + S_{A3}\bar{B}$$

$$= (S_1 + S_2)A + \bar{S}_2\bar{S}_1S_0\bar{A} + (S_2\bar{S}_1S_0 + S_2S_1\bar{S}_0)\bar{B}$$

$$F_B = S_{B0}A + S_{B1}\bar{A} + S_{B2}B + S_{B3}\bar{B}$$

$$= S_2S_0\bar{A} + (\bar{S}_2S_0 + S_2\bar{S}_1\bar{S}_0)B + S_1\bar{S}_0\bar{B}$$

$$F_C = S_C C = (\bar{S}_2S_0 + \bar{S}_2S_1)C$$

$S_2 \sim S_0$	$C_n=1$	$C_n=0$	$F_A \oplus F_B$	A/L
000	CLEAR	CLEAR	$0 \oplus 0$	L
001	B minus A	B minus A minus 1	$\bar{A} \oplus B$	A
010	A minus B	A minus B minus 1	$A \oplus \bar{B}$	A
011	A plus B plus 1	A plus B	$A \oplus B$	A
100	$A \oplus B$	$A \oplus B$	$A \oplus B$	L
101	$A + B$	$A + B$	$(A + \bar{B}) \oplus \bar{A}$	L
110	$AB$	$AB$	$(A + \bar{B}) \oplus \bar{B}$	L
111	PRESET	PRESET	$A \oplus \bar{A}$	L

・74LS681 のシフタ, レジスタ等を除いた ALU 部に似た演算機能を持つ4ビット ALU の例を示す.

$$F_A = S_{A0}A + S_{A1}\bar{A} + S_{A2}B + S_{A3}\bar{B}$$

$$= (\bar{M}\bar{S}_2S_1 + S_1\bar{S}_0 + M\bar{S}_2\bar{S}_1S_0)A$$

$$+ (\bar{M}\bar{S}_2\bar{S}_1S_0 + MS_2\bar{S}_1\bar{S}_0 + \bar{M}S_2S_1S_0)\bar{A}$$

$$+ MS_2\bar{S}_1S_0B + (\bar{M}\bar{S}_2\bar{S}_1\bar{S}_0 + MS_1S_0)\bar{B}$$

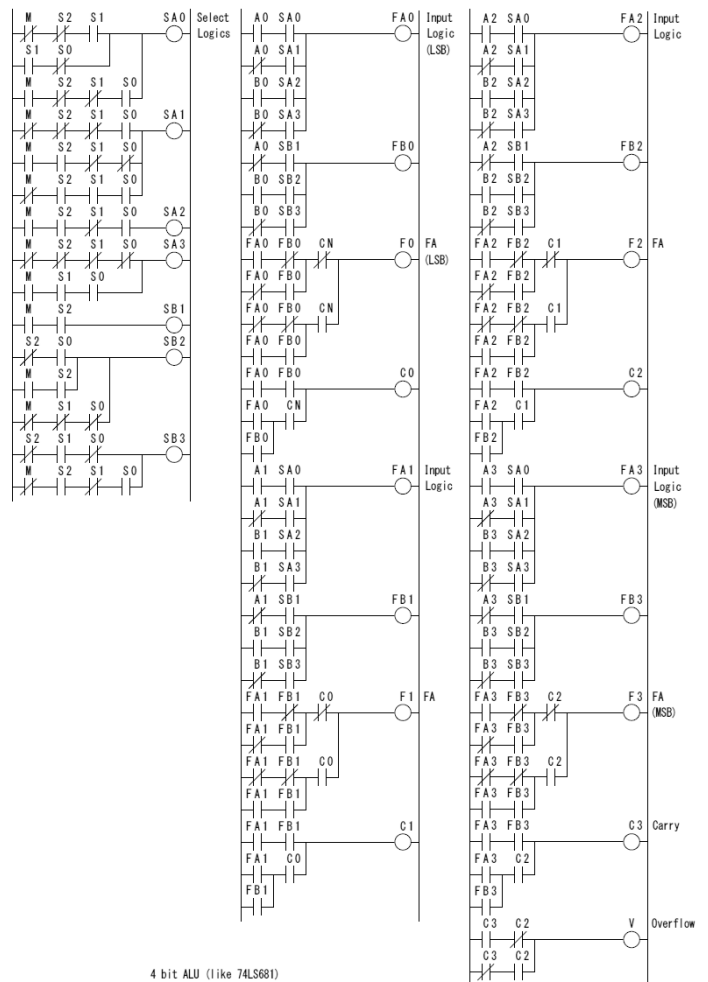
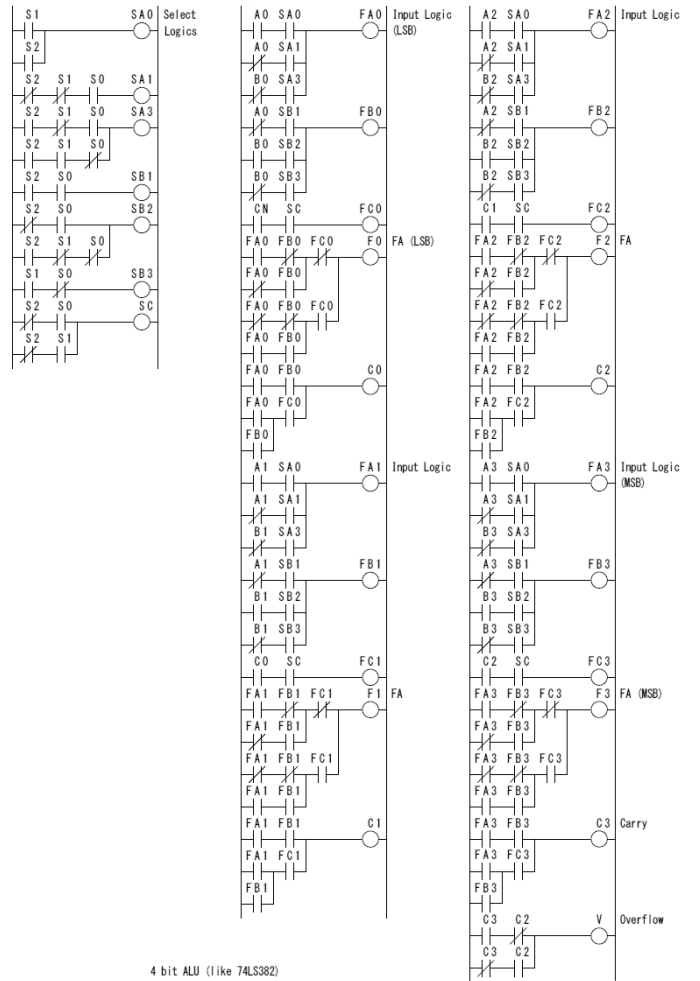
$$F_B = S_{B0}A + S_{B1}\bar{A} + S_{B2}B + S_{B3}\bar{B}$$

$$= MS_2\bar{A} + (\bar{S}_2S_0 + MS_2 + \bar{M}\bar{S}_1\bar{S}_0)B$$

$$+ (\bar{S}_2S_1\bar{S}_0 + \bar{M}S_2\bar{S}_1S_0)\bar{B}$$

$$F_C = C$$

$S_2 \sim S_0$	M=1 (Logic) $C_n=0$	M=0 (Arithmetic) $C_n=0$ (no carry)	M=0 (Arithmetic) $C_n=1$ (with carry)
000	CLEAR	PRESET	CLEAR
001	$A \oplus B$	B minus A minus 1	B minus A
010	$A \oplus \bar{B}$	A minus B minus 1	A minus B
011	PRESET	A plus B	A plus B plus 1
100	$AB$	B	B plus 1
101	$A + B$	$\bar{B}$	$\bar{B}$ plus 1
110	$\bar{A}\bar{B}$	A	A plus 1
111	$A + B$	$\bar{A}$	$\bar{A}$ plus 1



・ M. Morris Mano によるシフト付き ALU<sup>(2)</sup>に基づく 4 ビット ALU の例を示す. ALU への 2 進数の入力を A と B, キャリ入力を C<sub>in</sub>, ALU の演算機能選択信号を S<sub>0</sub>~S<sub>2</sub> とする. この ALU の出力 F をシフタの入力に接続する. シフタの機能選択信号を H<sub>0</sub>~H<sub>2</sub>, シフタの出力を Y とする. また, 状態フラグをキャリ C (ボローは C の否定), オーバーフロー V, ゼロ Z, 符号 S とする. 下の表に機能選択信号に対する演算機能と状態フラグを示す. 表中の状態フラグの\*印は演算結果により変化することを示し, -印は変化しないことを示す.

ALU を構成する FA への入力論理関数を示す.

$$F_A = S_{A0}A + S_{A1}\bar{A} + S_{A2}B + S_{A3}\bar{B}$$

$$= A + S_2\bar{S}_0\bar{S}_1B + S_2S_1\bar{S}_0\bar{B}$$

$$F_B = S_{B0}A + S_{B1}\bar{A} + S_{B2}B + S_{B3}\bar{B}$$

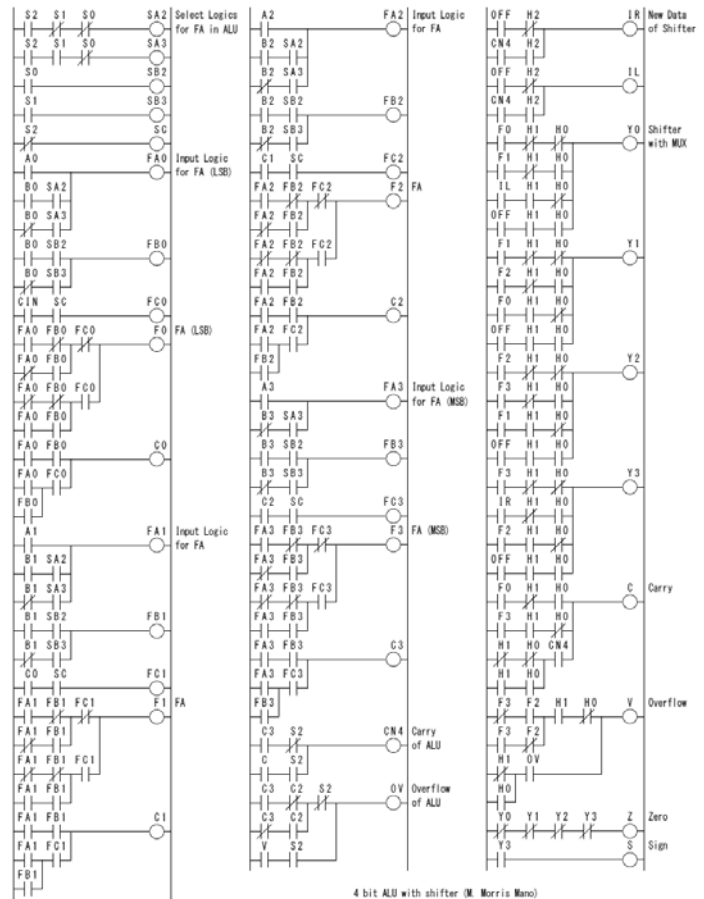
$$= S_0B + S_1\bar{B}$$

$$F_C = S_C C = \bar{S}_2 C$$

1 ビット双方向シフタは H<sub>0</sub>, H<sub>1</sub> を選択信号とするマルチプレクサ (MUX) で構成されており, H<sub>2</sub> でシフト後の空いたビットに 0 が入るかキャリ C が入るかを選択する. いずれにせよ溢れたビットは C に入る.

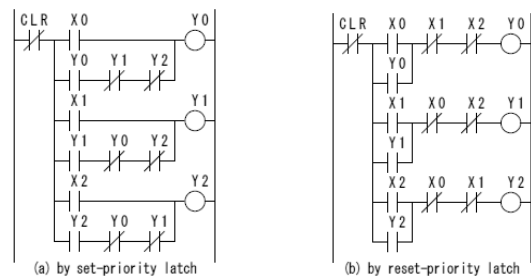
S <sub>2</sub> ~S <sub>0</sub>	C <sub>in</sub> =0	C <sub>in</sub> =1	CVZS
000	A, C←0	A plus 1	* * * *
001	A plus B	A plus B plus 1	* * * *
010	A minus B minus 1	A minus B	* * * *
011	A minus 1	A, C←1	* * * *
100	A + B	A + B	- - * *
101	A ⊕ B	A ⊕ B	- - * *
110	AB	AB	- - * *
111	$\bar{A}$	$\bar{A}$	- - * *

H <sub>2</sub> ~H <sub>0</sub>	Shift function	CVZS
000	No shift	- - * *
001	Shift-right, MSB←0	* - * *
010	Shift-left, LSB←0	* * * *
011	CLEAR	- - * *
100	No shift	- - * *
101	Circulate-right with C (Rotate-right)	* - * *
110	Circulate-left with C (Rotate-left)	* * * *
111	CLEAR	- - * *

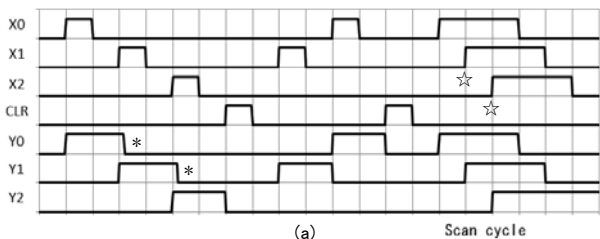


## 補足 2. 新入力優先回路 (後押し優先回路)

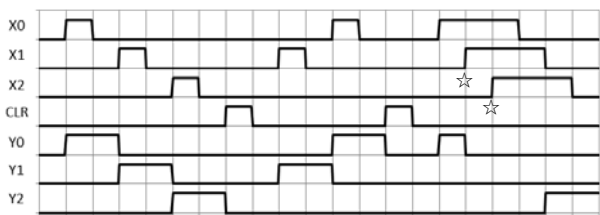
付録の本文中ではセット優先自己保持回路を応用した新入力優先回路の例を示したが, リセット優先自己保持回路でリセットを出力ではなく入力で行うと新入力優先となる. 図(a)はセット優先自己保持回路を用いた 3 入力の回路 (前出と同じ), 図(b)はリセット優先自己保持回路を用いた回路である.



しかし, これらの回路では複数の入力があるとき問題が発生する. タイミングチャートの☆印に示すように, 1 つの入力が ON のままで別の入力が OFF から ON に上がった場合, 図(a)の回路では ON になっている入力に対応する複数の出力が ON になってしまい, 図(b)の回路では全ての出力が OFF になってしまう. また, \*印のように図(a)の回路ではプログラム順序によっては出力 Y の立下りが 1 スキャンサイクル遅れる.

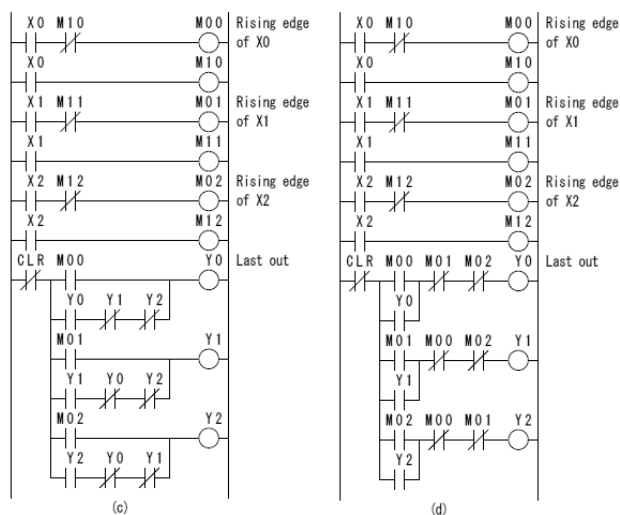


(a) Scan cycle



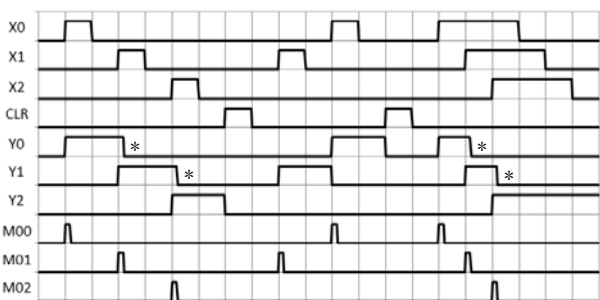
(b) Scan cycle

ある入力が ON のままで別の入力が立上がった場合にも後から立上がった入力に対する出力のみを ON にする方法の 1 つに、立上がりパルス回路を前段に付加する方法がある。図(c), (d)にその回路を示す。ただし、1 スキャンの間で同時に複数の入力が立上がる場合には問題は残る。

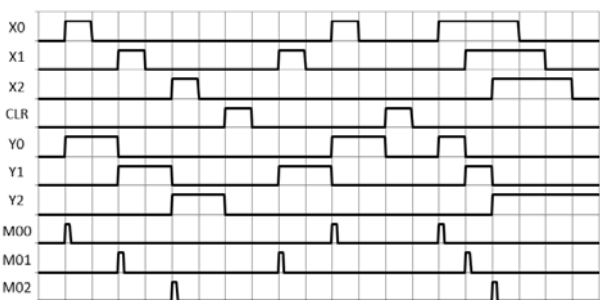


(c)

(d)



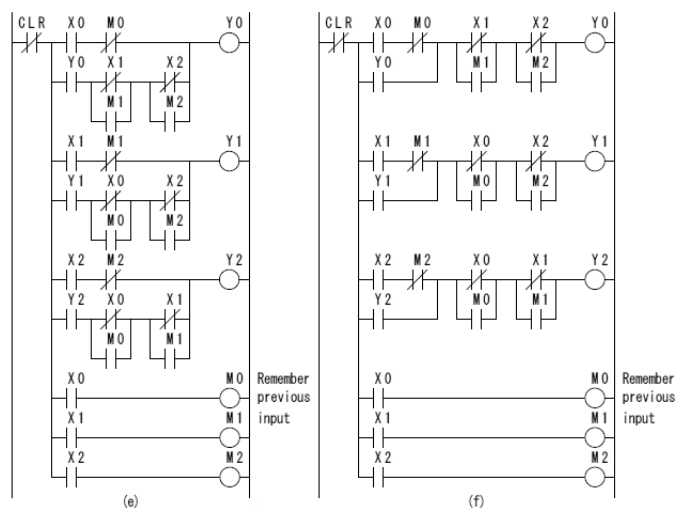
(c) Scan cycle



(d) Scan cycle

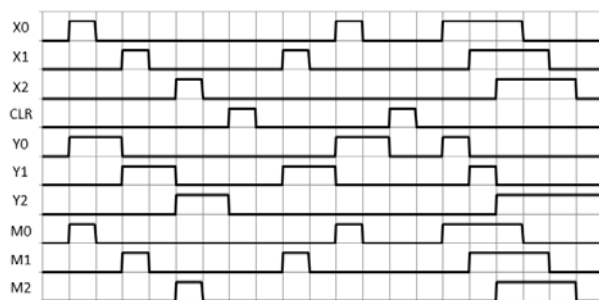
この立上がりパルス回路では 1 スキャン前の入力を  $M_{10} \sim M_{12}$  に記憶しておいて、それと現在の入力  $X_0 \sim X_2$  から立上がりを検出して  $M_{00} \sim M_{02}$  に 1 スキャンのみ ON するパルスを発生させている。入力の記憶を後にして、立上がりパルスのための補助リレー  $M_{00} \sim M_{02}$  を使用しないようにすると、図(e), (f)のようにすることができる。

現在の状態を  $Q_n = (M_n, Y_n)$ , 次の状態を  $Q_{n+1} = (M_{n+1}, Y_{n+1})$ , 入力を  $X$  とすると,  $M_{n+1} = X$  であり, 出力は  $Y_{n+1}$  である。ただし,  $X, M_n, Y_n, M_{n+1}, Y_{n+1}$  の各要素の数は 3 である。  $Y_{n+1}$  のある要素番号のコイル (例えば図中の  $Y_0$ ) の駆動は  $Y_n$  の他の要素番号の接点 (図中の  $Y_1, Y_2$ ) に依存しないためプログラムの順序によらない。図(e), (f) に対する簡略化した状態遷移表 (出力表) を示す。なお, 表中の  $x$  は don't care を表す。網掛けした複数の入力が同時に立上がる場合に注意すること。

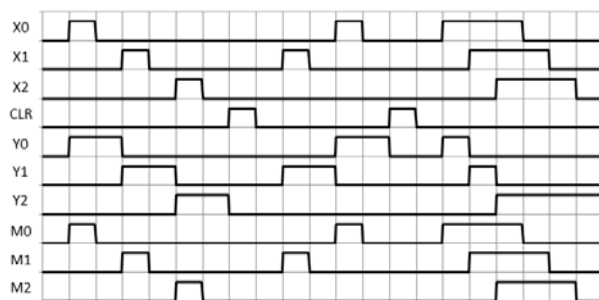


(e)

(f)



(e) Scan cycle



(f) Scan cycle

図(e), (f) の簡略化した状態遷移表

現在の状態 $Q_n$			次の状態 $Q_{n+1}$ 一部 出力 $Y_{n+1}$		現在の状態 $Q_n$			次の状態 $Q_{n+1}$ 一部 出力 $Y_{n+1}$	
$M_n$	$Y_n$	X	(e) $Y_{n+1}$	(f) $Y_{n+1}$	$M_n$	$Y_n$	X	(e) $Y_{n+1}$	(f) $Y_{n+1}$
000	$Y_n$	000	$Y_n$	$Y_n$	100	$Y_n$	x00	$Y_n$	$Y_n$
000	xxx	001	001	001	100	xxx	x01	001	001
000	xxx	010	010	010	100	xxx	x10	010	010
000	xxx	011	011	000	100	xxx	x11	011	000
000	xxx	100	100	100	101	$Y_n$	x0x	$Y_n$	$Y_n$
000	xxx	101	101	000	101	xxx	x1x	010	010
000	xxx	110	110	000	110	$Y_n$	xx0	$Y_n$	$Y_n$
000	xxx	111	111	000	110	xxx	xx1	001	001
001	$Y_n$	00x	$Y_n$	$Y_n$	111	$Y_n$	xxx	$Y_n$	$Y_n$
001	xxx	01x	010	010	次の状態 $Q_{n+1}=(M_{n+1}, Y_{n+1})$ $M_{n+1}=X$ $Y_{n+1}$ が $Y_n$ となっている所は 保持 x は don't care				
001	xxx	10x	100	100					
001	xxx	11x	110	000					
010	$Y_n$	0x0	$Y_n$	$Y_n$					
010	xxx	0x1	001	001					
010	xxx	1x0	100	100					
010	xxx	1x1	101	000					
011	$Y_n$	0xx	$Y_n$	$Y_n$					
011	xxx	1xx	100	100					

注意事項

この付録のいずれの場合も入力にスイッチを用いた時のチャタリングについては考慮していない。

参考書

- 注：新しい版では削除されていることがある。
- “デジタル回路の考え方(改訂2版)”，清水，曾和，オーム社，1991
  - “Digital Logic and Computer Design”，M. Morris Mano, Prentice-Hall, 1979 (日本語訳：“コンピュータの論理設計”，奥川，井上，共立出版，1983)
  - “Digital Design and Computer Architecture”，David Money Harris, Sarah L. Harris, Morgan Kaufmann Pub., 2007 (日本語訳：“デジタル回路設計とコンピュータアーキテクチャ”，鈴木 他，翔泳社，2009)
  - “最新 TTL IC 規格表”，CQ 出版，1979, 1983, 1988 年版
  - ラダー式プログラマブルコントローラ KOSTAC SR-20 技術資料，光洋電子工業，1985
  - 三菱マイクロシーケンサ MELSEC 命令解説書およびテキスト，三菱電機，1984, 1988
  - シャープ プログラマブル コントローラ・ニューサテライト JW プログラム集，SHARP, 2001

ここでは示していないが，ON となる出力を必ず 1 個以下にすることもできる。ただし， $Y_{n+1}$  のある要素番号のコイルの駆動が  $Y_n$  の他の要素番号の接点に陽に依存する場合はラダー図の作成に注意を要する。

補足 3. 7セグメント表示デコーダ

この付録の本文中では小文字の 'c' とした 7セグメント表示デコーダの回路を示したが，ここでは大文字の 'C' とする回路を示す。出力 a, f, g の論理関数が異なっている。なお，この回路の出力も負論理であり，項をまとめていない。

