

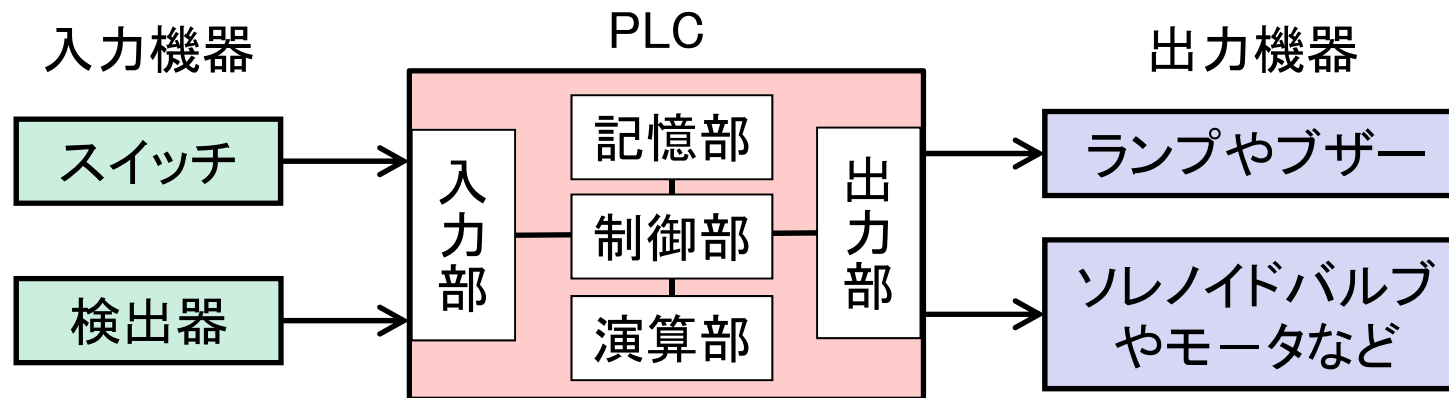
シーケンス制御

- **シーケンス制御**

あらかじめ定められた順序に従って制御の各段階を逐次進めていく制御

- **プログラマブルロジックコントローラ: PLC (シーケンサ)**

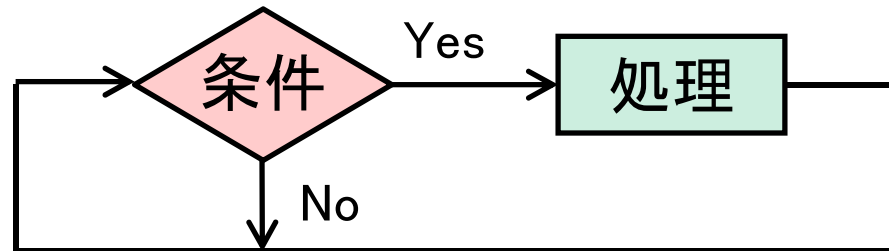
入出力部を介して各種装置を制御するものであり、プログラマブルな命令を記憶するためのメモリを内蔵した電子装置



条件制御と順序制御

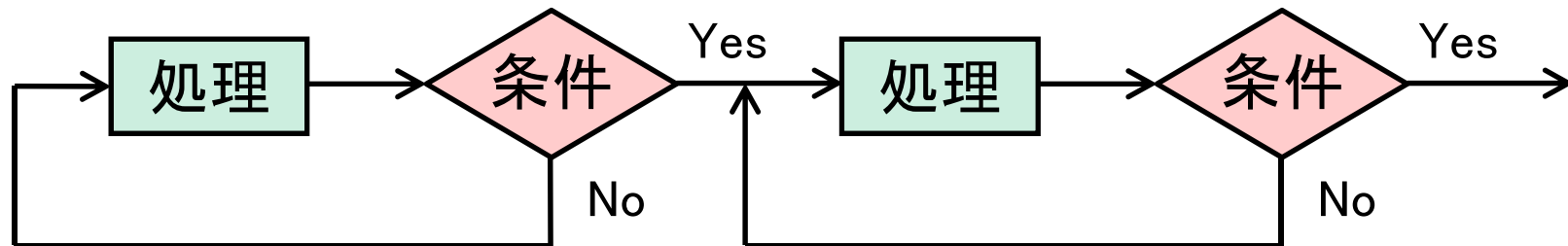
● 条件制御

条件制御における条件は、ある処理をするための条件
(処理条件)



● 順序制御

順序制御における条件とは、次の処理に移るための条件
(移行条件)

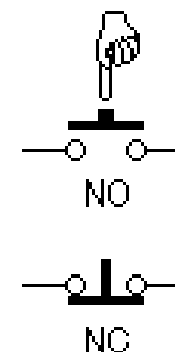


スイッチに関する用語

● NO, NC

Normally Open: スイッチ操作しない状態で, 接点が開

Normally Close: スイッチ操作しない状態で, 接点が開



● Make and Break (Make: 閉, Break: 開)

メイク接点 : OFFで開, ONで閉となる接点 (a接点)

ブレーク接点 : OFFで閉, ONで開となる接点 (b接点)

● BBM, MBB

Break-Before-Make

Make-Before-Break

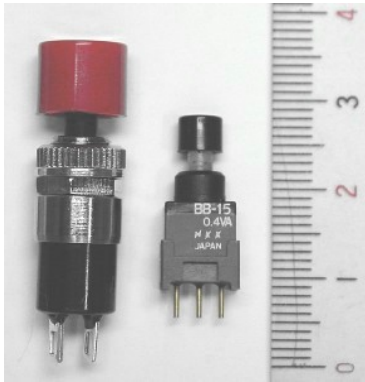


● モーメンタリ, オルタネイト

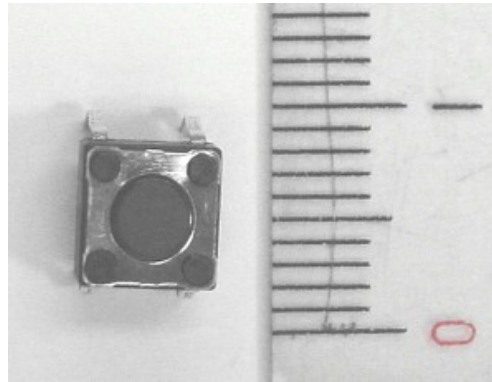
モーメンタリ : 手を離せば, 初期の状態へ復元

オルタネイト : 押すたびに状態が切り替わり, 保持

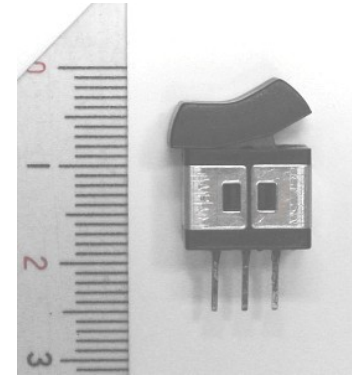
スイッチの例



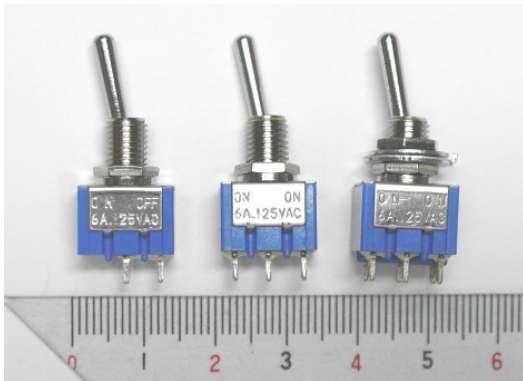
プッシュ・スイッチ
(押しボタン・スイッチ)



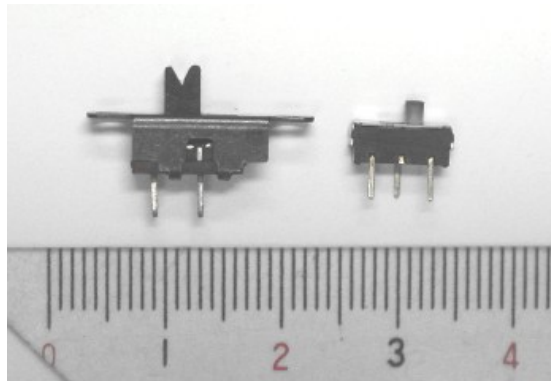
タクトイル・スイッチ
(タクト・スイッチ)



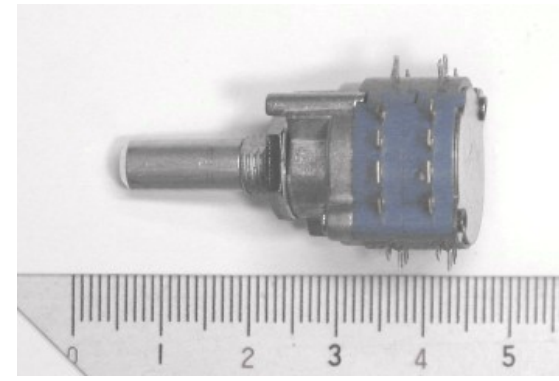
ロッカ・スイッチ
(シーソー・スイッチ)



トグル・スイッチ



スライド・スイッチ



ロータリ・スイッチ

基本的な論理演算

～ ブール代数 ～

- **否定** (NOT) 論理式 $Y = \bar{A}$



真理値表

A	Y
0	1
1	0

- **論理積** (AND) 論理式 $Y = A \cdot B$



ビットのリセットや抽出

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

- **論理和** (OR) 論理式 $Y = A + B$

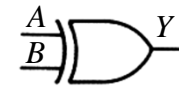


ビットのセット

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

- **排他的論理和** (XOR, EXOR) 論理式 $Y = A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$

$$Y = A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$$

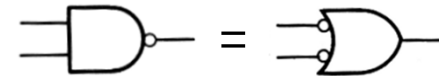


ビットの反転

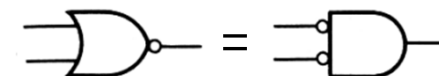
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

- **ド・モルガンの定理**

$$Y = \overline{A \cdot B} = \bar{A} + \bar{B} \quad (\text{NAND})$$



$$Y = \overline{\bar{A} + \bar{B}} = \bar{\bar{A}} \cdot \bar{\bar{B}} \quad (\text{NOR})$$



ブール代数

$$A \cdot 1 = A$$

$$A + 1 = 1$$

単位元

$$A \cdot 0 = 0$$

$$A + 0 = A$$

零元

$$A \cdot \bar{A} = 0$$

$$A + \bar{A} = 1$$

補元

$$A \cdot B = B \cdot A$$

$$A + B = B + A$$

交換律

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

$$A + (B + C) = (A + B) + C$$

結合律

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C) \quad A + (B \cdot C) = (A + B) \cdot (A + C)$$

分配律

$$A \cdot (A + B) = A$$

$$A + (A \cdot B) = A$$

吸収律

$$A \cdot A = A$$

$$A + A = A$$

べき等律

$$\overline{(\bar{A})} = A$$

二重否定

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

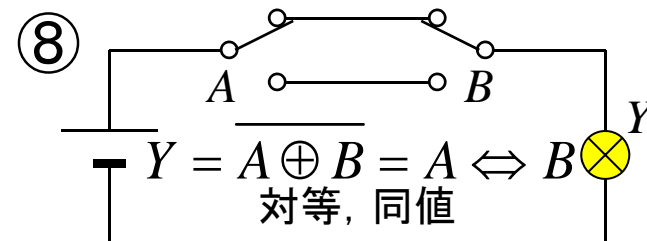
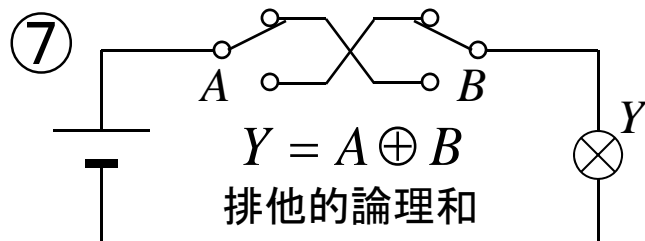
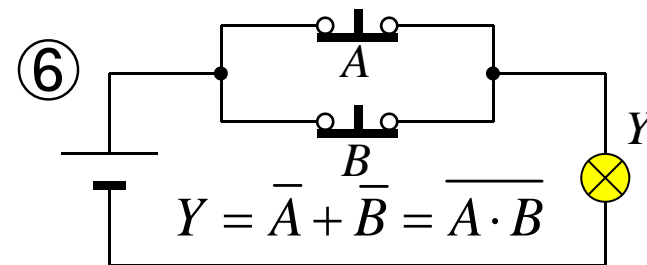
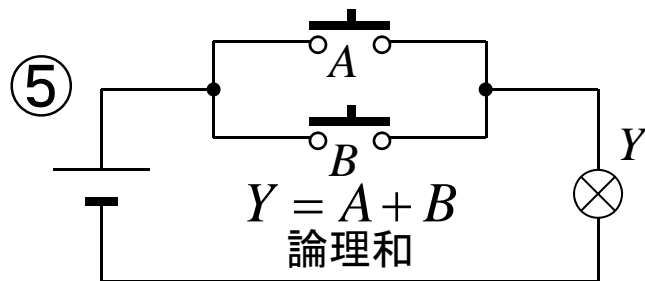
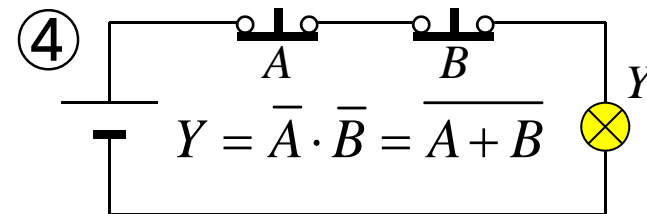
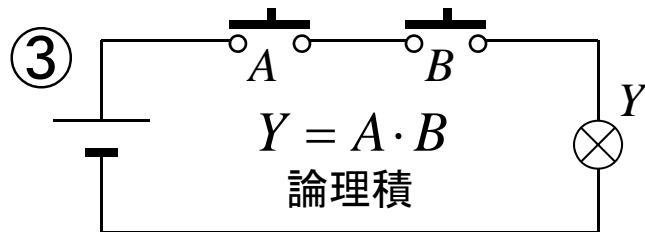
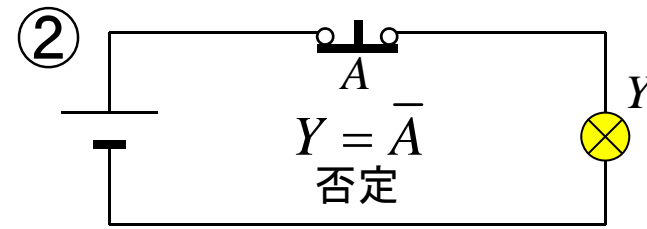
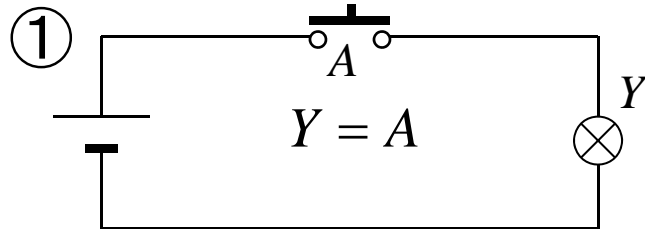
$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

ド・モルガンの定理

定数0,1を含む論理関数の恒等式は0と1, ANDとORを同時に入れ替えても成立する

双対定理

スイッチ回路と論理演算

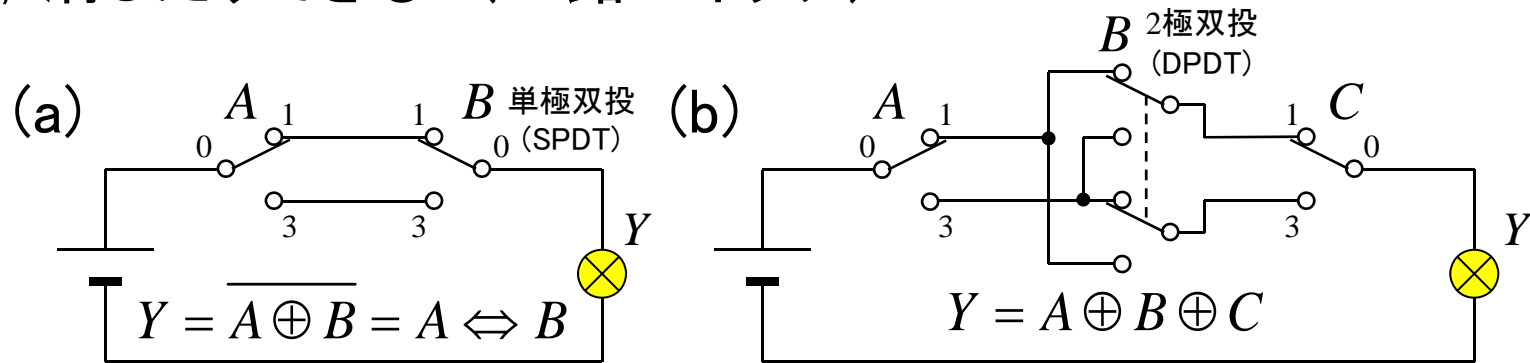


押しボタン・スイッチを押したときA, Bが1. トグル・スイッチは上が1, 下が0.

単極双投 (SPDT)

階段灯スイッチ回路

(a) 1階と2階のどちらのスイッチを操作してもランプを点けたり、消したりできる（三路スイッチ）



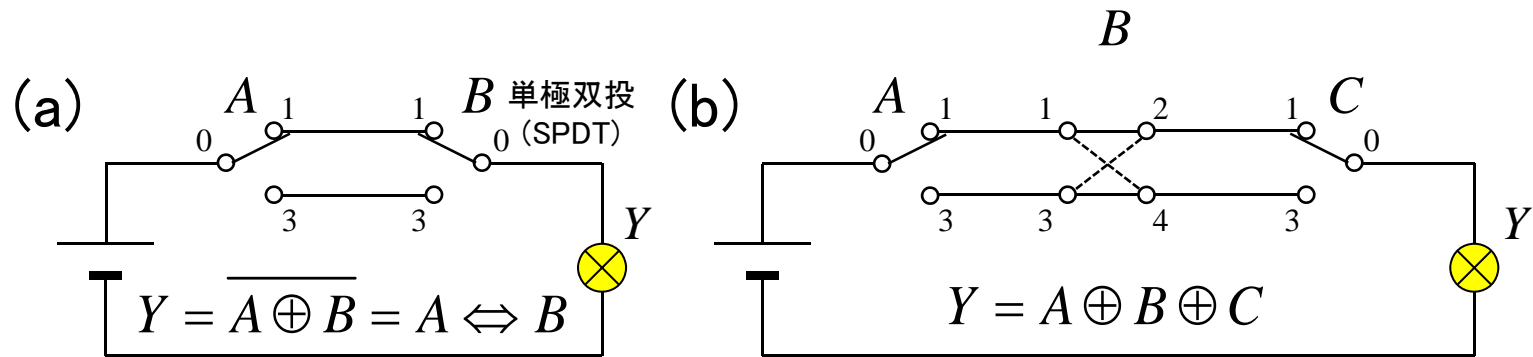
(b) 3個のスイッチのどのスイッチでもランプを点けたり、消したりできる

スイッチBは、上下のスイッチが同時に動作する必要がある（四路スイッチ）

4個以上のときは、3個の時の真ん中のスイッチ回路を必要な数だけ接続する

階段灯スイッチ回路

(a) 1階と2階のどちらのスイッチを操作してもランプを点けたり、消したりできる（三路スイッチ）



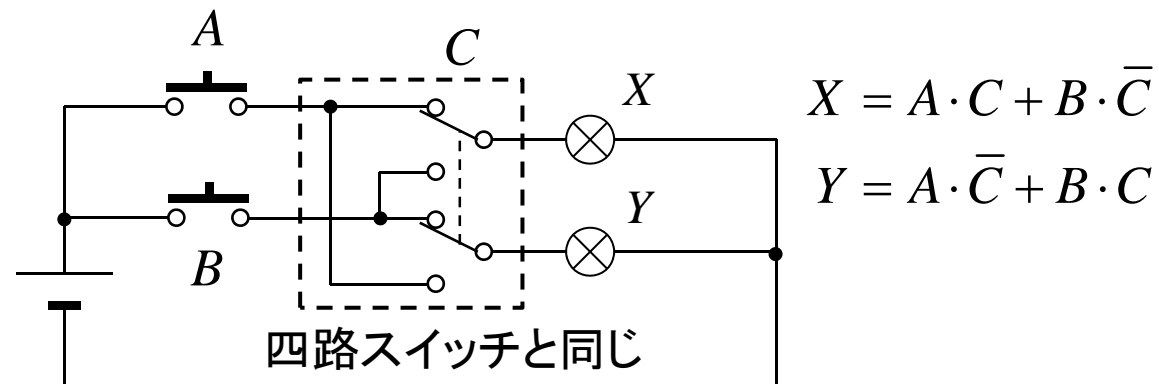
埋込三路スイッチ



埋込四路スイッチ

信号入れ替え回路

2つの信号線をスイッチで入れ替える



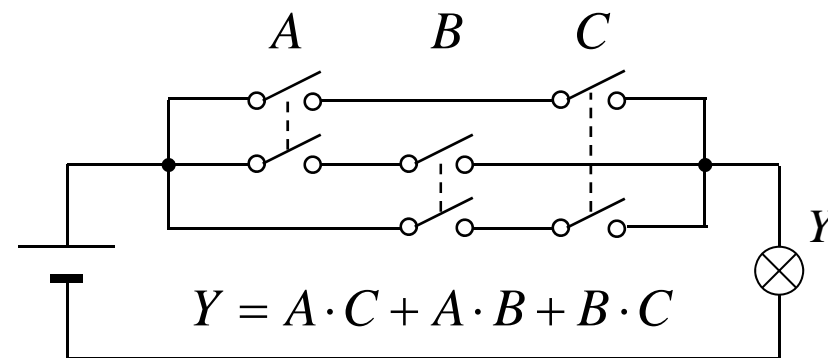
点線で結ばれたスイッチCは、同時に動作する必要がある

上のスイッチを押すと上のランプが点灯し、下のスイッチを押すと下のランプが点灯する状態と、

上のスイッチを押すと下のランプが点灯し、下のスイッチを押すと上のランプが点灯する状態とを入れ替える

多数決回路

3個のスイッチのうち、いずれか2個がONになったときランプが点灯する



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

積和形式の論理式

A B C	Y
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

真理値表

点線で結ばれたスイッチは、同時に動作する必要がある

5個のスイッチのうちいずれか3個がONになったときランプが点灯する回路を同じように作ろうとすると、回路が大きくなってしまふ

$$Y = ABC + ABD + ABE + ACD + ACE + BCD + BCE + ADE + BDE + CDE$$

論理回路

● 組合わせ回路

出力が現在の入力のみで決まる回路

算術演算(加算器, 乗算器, 比較器など)

エンコーダ(符号器), デコーダ(復号器)

マルチプレクサ(選択回路), デマルチプレクサ

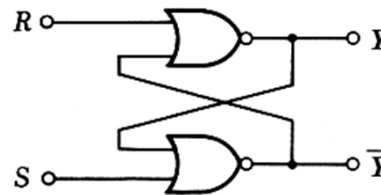
● 順序回路

出力が現在の入力だけでなく,
過去の入力にも依存する回路

➡ **記憶** が必要

フリップフロップ

カウンタ, レジスタ



入力		出力
R	S	Y
0	0	変化なし
0	1	1
1	0	0
1	1	(禁止)

RSフリップフロップの例

論理関数の簡略化

● カルノー図

$$f = A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + \overline{A + C}$$

C \ AB		A				グレイコード
		00	01	11	10	
C	0	1	1	0	0	
	1	0	1	0	1	
		$\bar{A} \cdot \bar{C}$	$\bar{A} \cdot B$	B	$A \cdot \bar{B} \cdot C$	

$$f = A \cdot \bar{B} \cdot C + \bar{A} \cdot B + \bar{A} \cdot \bar{C}$$

A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

真理値表

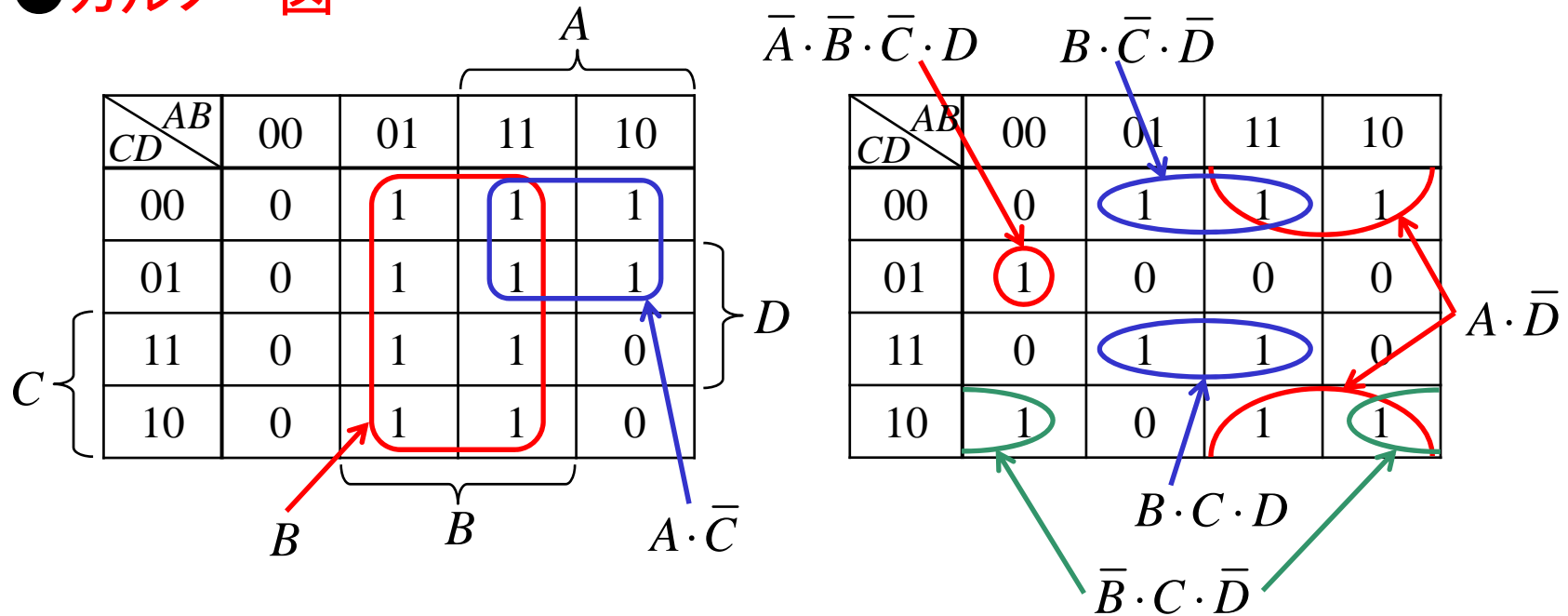
グレイコード(グレイ符号)

隣接した2つのセルは1変数の値しか異なる

隣接する2のべき乗個のセルが1ならば簡略化可能
カルノー図の上下・左右は繋がっている

論理関数の簡略化

● カルノー図

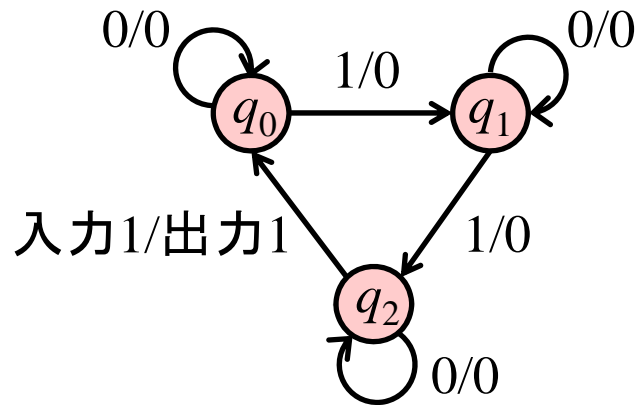


複数の取り方がある場合もある

● クワイン・マクラスキー法

順序回路

● 状態遷移図



時刻 i における

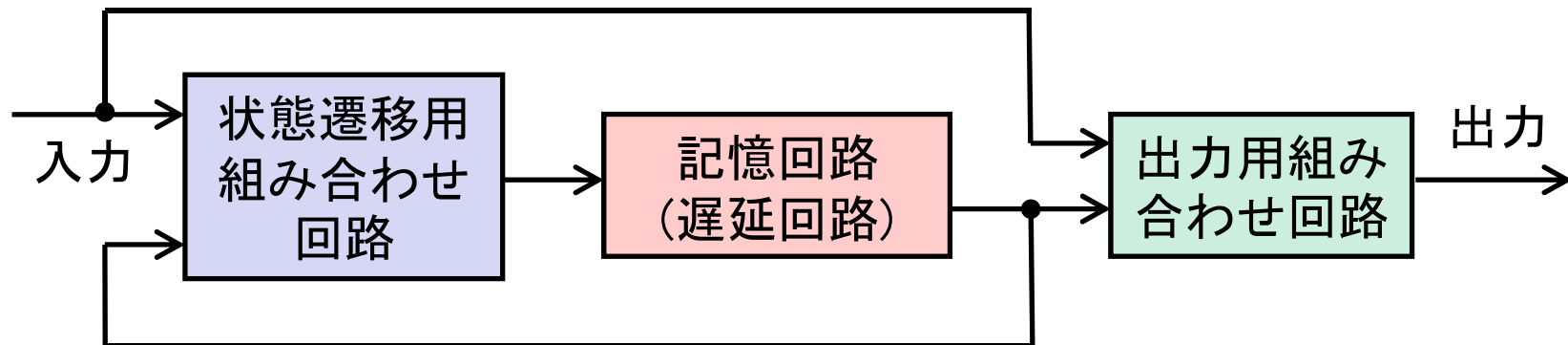
入力 x_i , 出力 y_i , 状態 q_i

$q_{i+1} = \omega(x_i, q_i)$ 状態遷移関数

$y_i = \delta(x_i, q_i)$ 出力関数

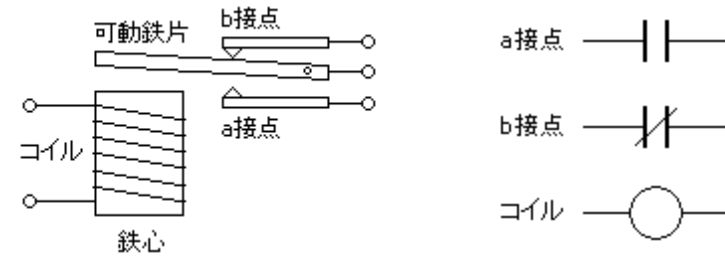
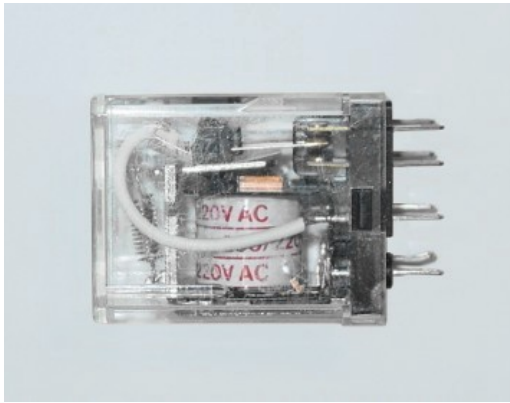
状態遷移表と出力表でも表すことができる

● 順序回路の構成法の例



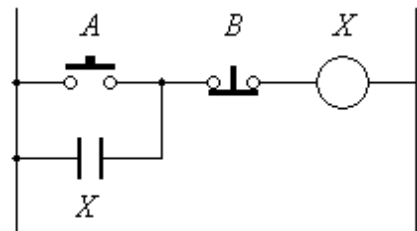
自己保持回路

●電磁リレー(電磁継電器)

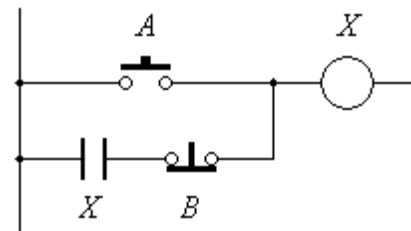


コイルに電流が流れると鉄心が電磁石となつて可動鉄片を引きつけ、それによつて接点を閉じたり開いたりする

●自己保持回路



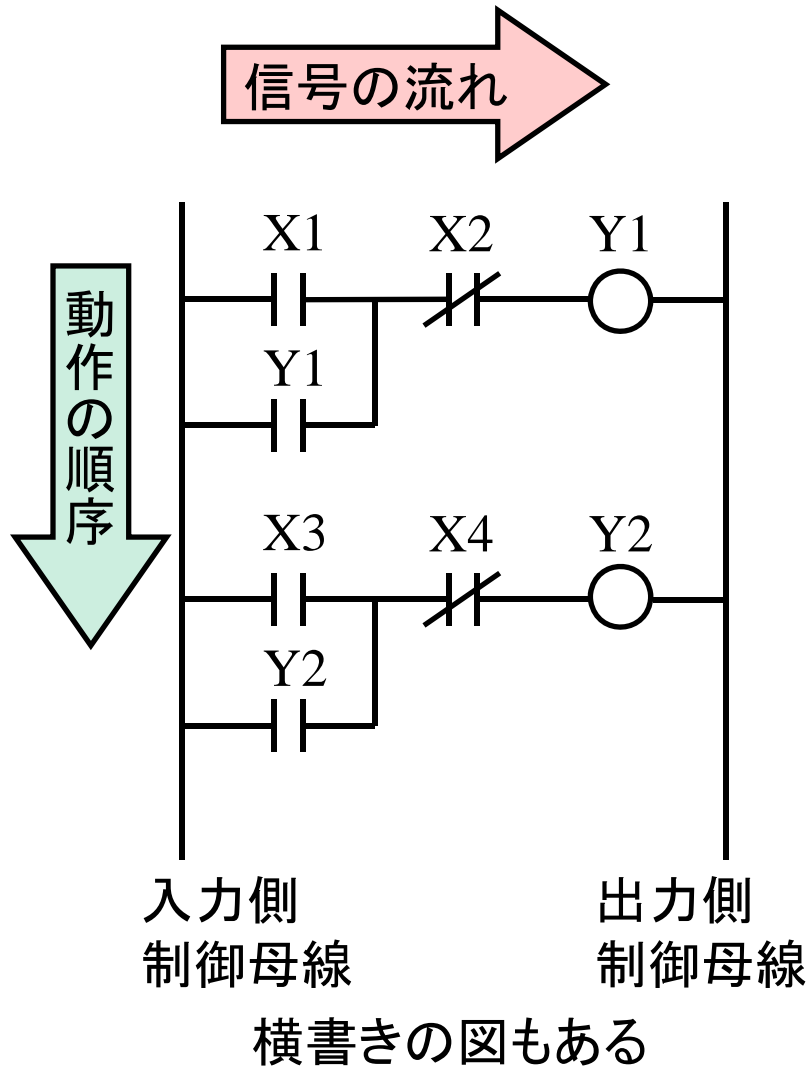
リセット優先



セット優先

➡ 順序回路

ラダー図

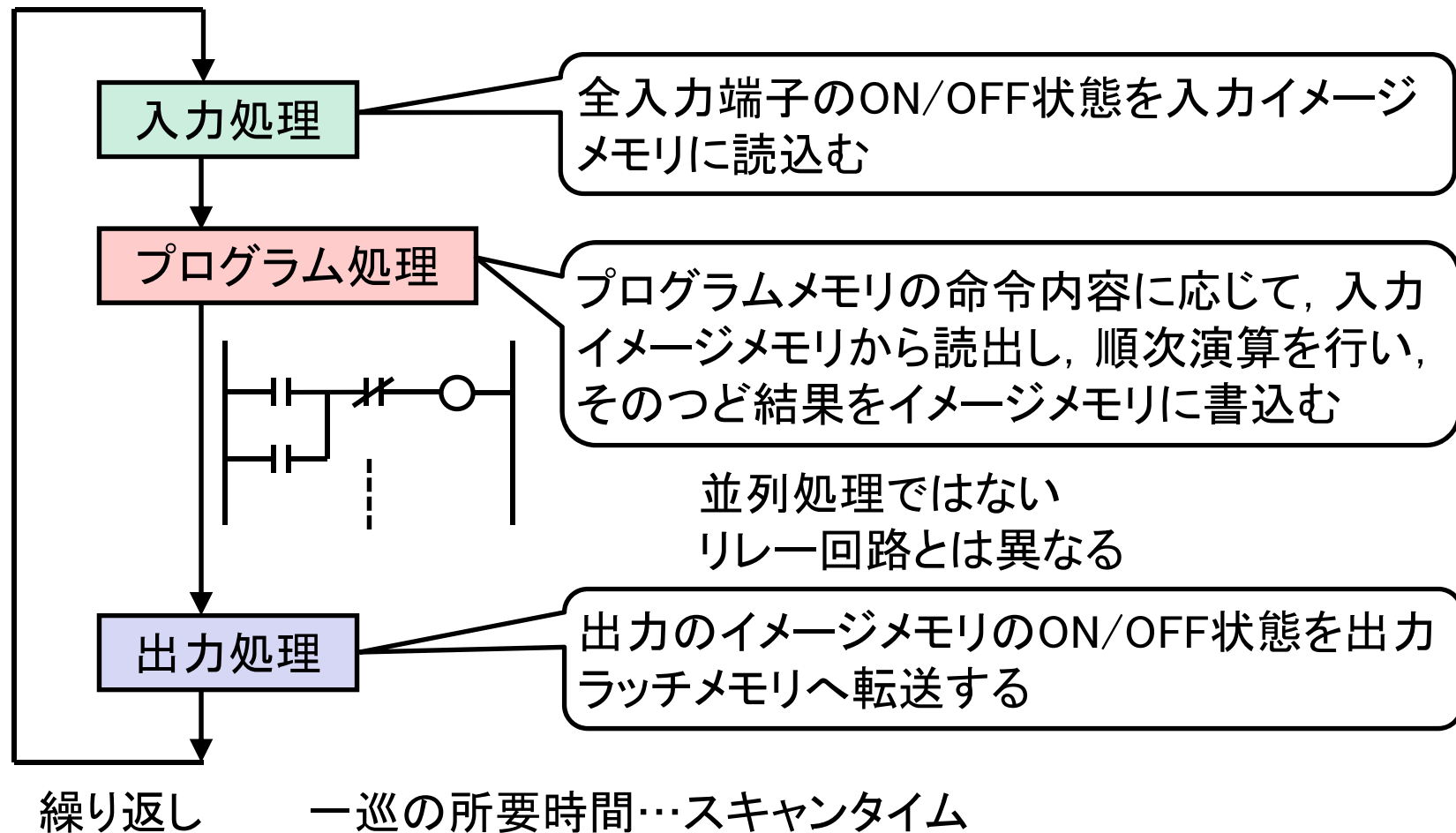


- ・電源は省略
- ・スイッチやリレー接点の区別なし(接点はa接点かb接点)
- ・橋渡しをしない
- ・コイルの右側に接点を書かない
- ・同じ番号のコイル (ダブルコイル) を使用しない

	JIS C 0617	旧JIS	ラダー図
a接点			
b接点			

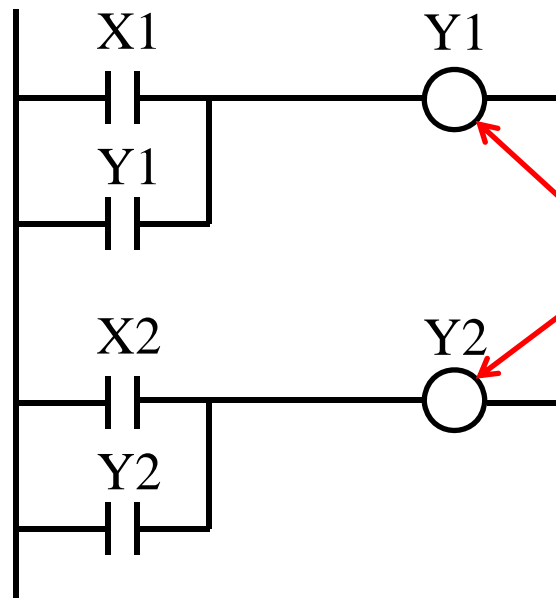
PLCの動作

●一括入出力方式(一括リフレッシュ方式)



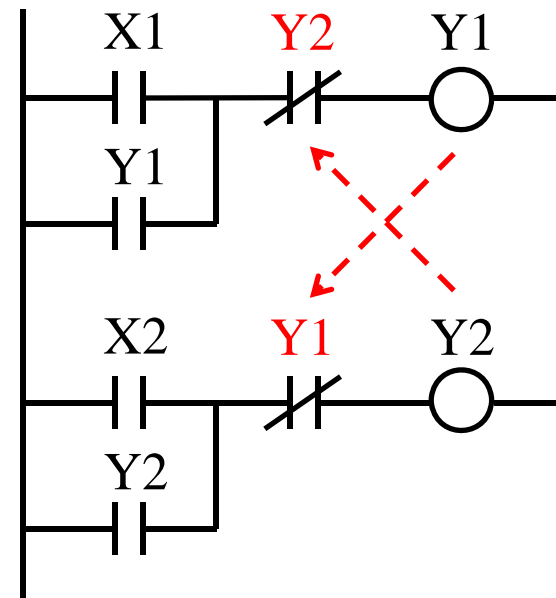
インターロック回路

2つの出力を同時に出力させない…**インターロック**
2つの入力信号のうち、先に動作したほうを優先し、他方の動作を禁止する…**優先回路**



独立な回路

同時に出力可能



インターロック回路

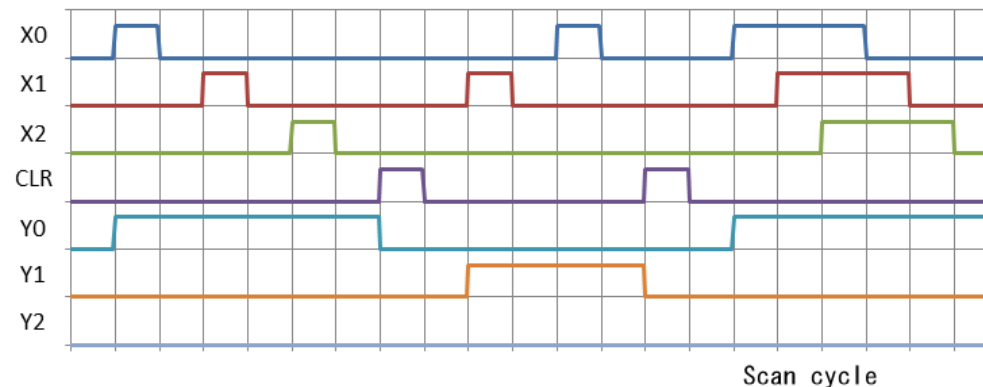
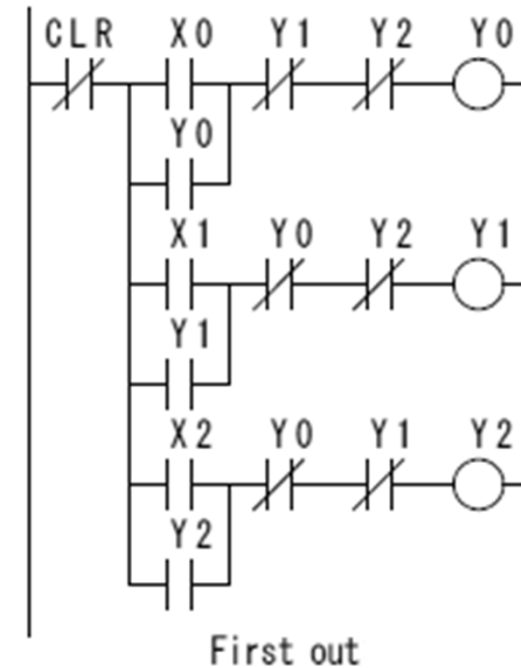
互いに相手の動作を禁止

(リセットは省略してある)

並列優先(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の応用

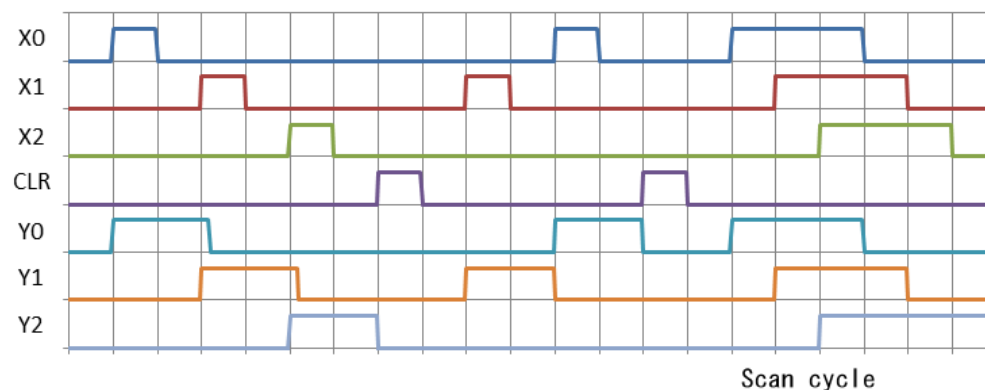
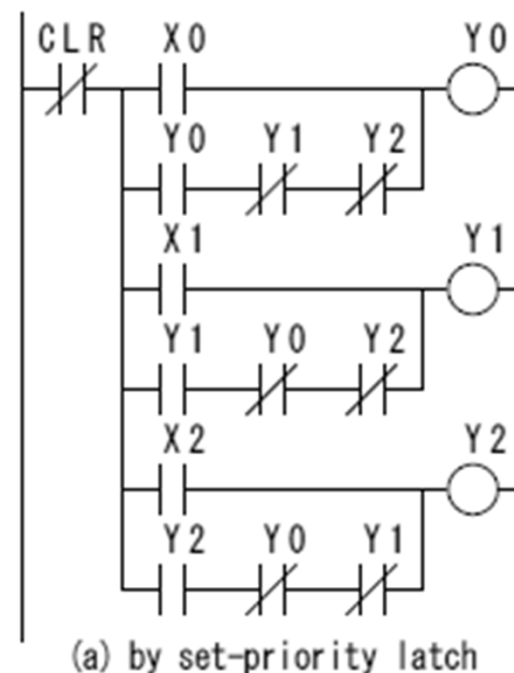
早押しクイズのように、最も先にONにされたスイッチに対応する出力をONにし、後から別のスイッチを押しても出力は変わらない



新入力優先（後押し優先）回路(a)

セット優先自己保持回路を応用した
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

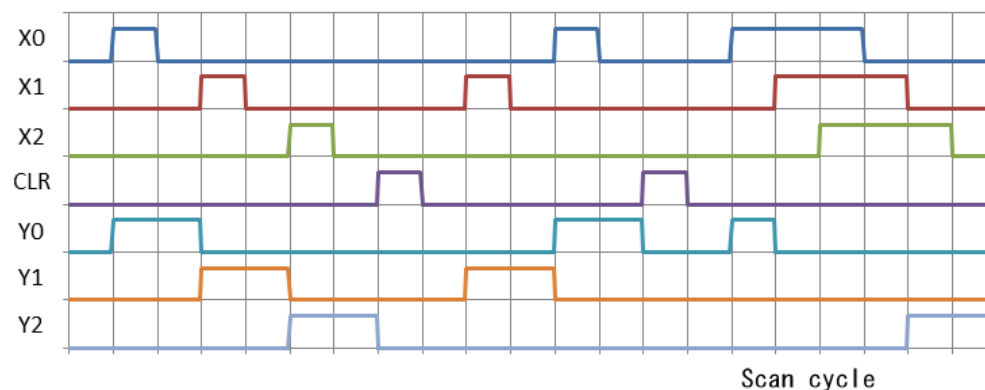
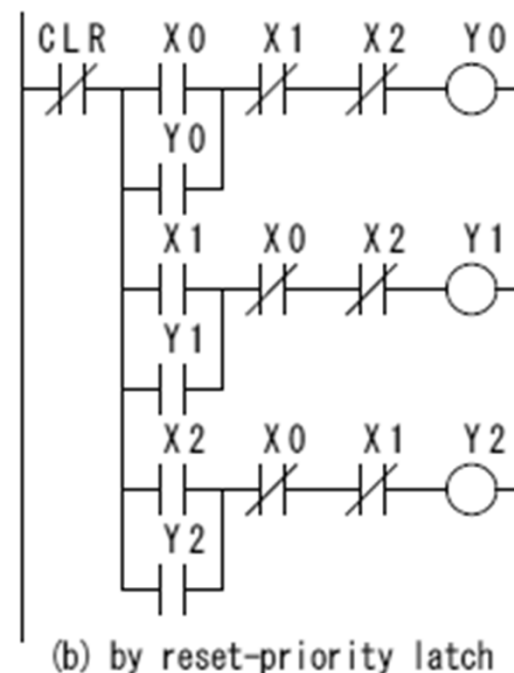
複数の入力がONのとき問題が発生
→立上がりパルス回路を前段に付加
するなどの対策が必要



新入力優先（後押し優先）回路(b)

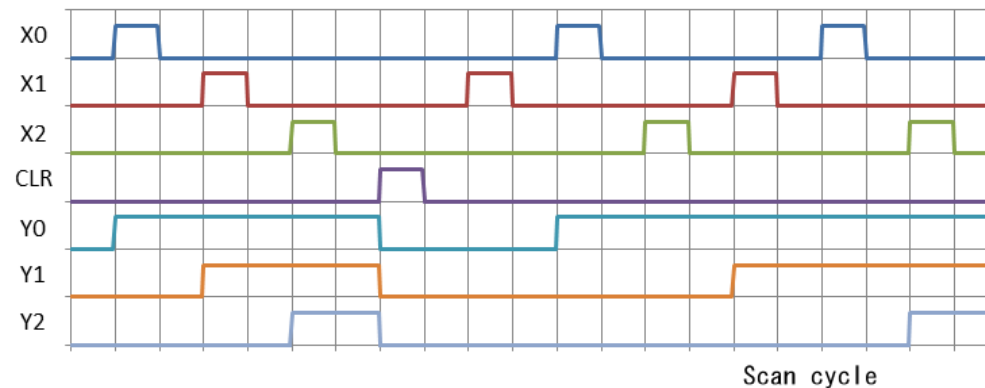
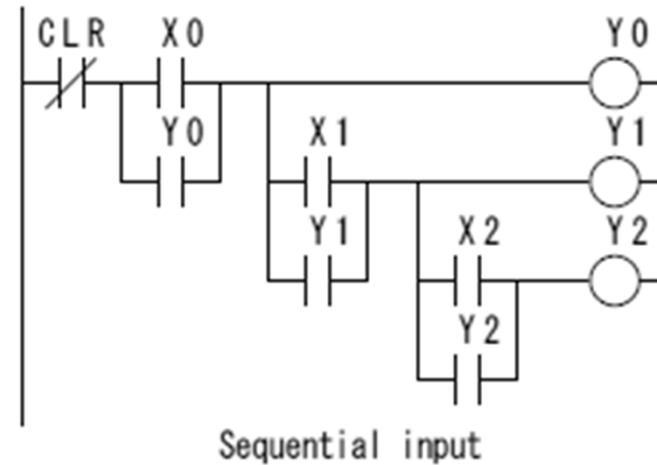
リセット優先自己保持回路でリセットを出力ではなく入力で行う新入力優先回路

複数の入力がONのとき問題が発生
→立上がりパルス回路を前段に付加するなどの対策が必要



直列優先回路

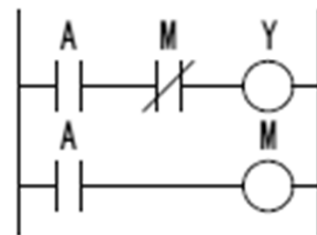
図の左側の入力側制御母線に近い回路の優先が最も高く、左側のスイッチから順にONしないと動作しない回路



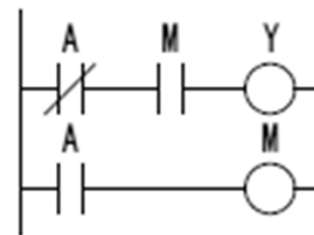
立上がり, 立下がりパルス出力回路

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スキャンサイクルのみ
ONするパルスをYに出力する

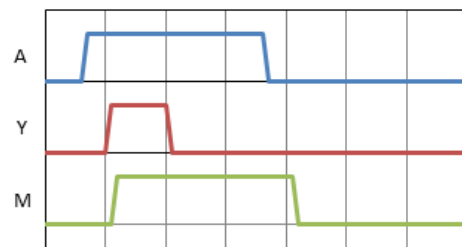
PLCによっては微分と称しているものもある
プログラムの順序を入れ替えると機能しない



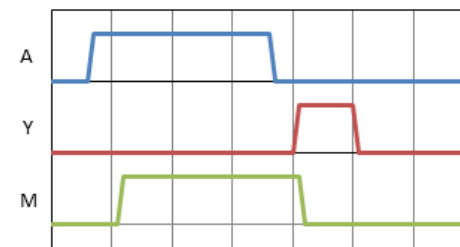
Rising (positive) edge
pulse generator



Falling (negative) edge
pulse generator



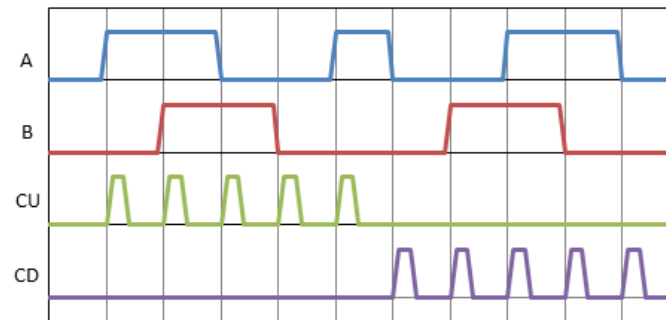
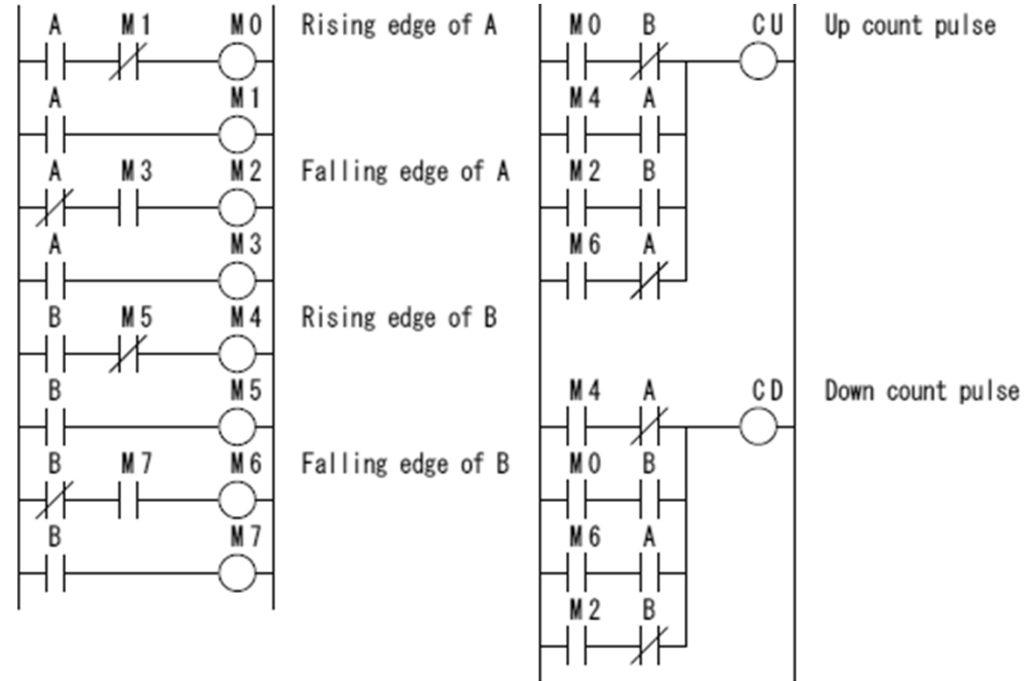
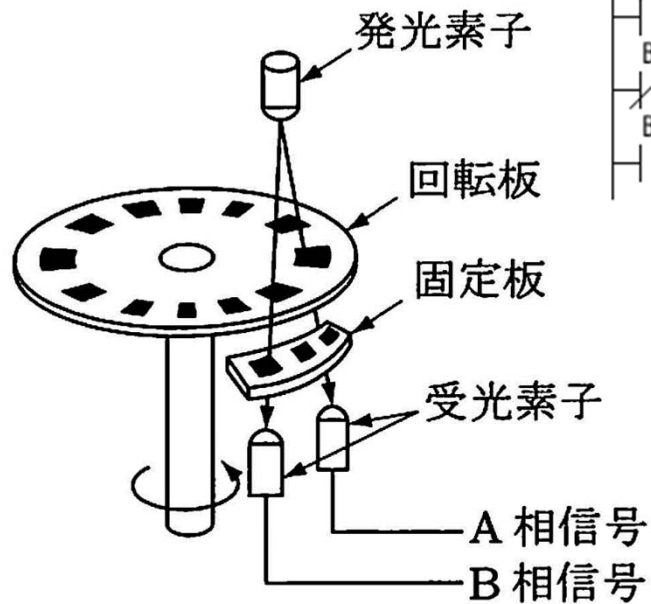
Scan cycle



Scan cycle

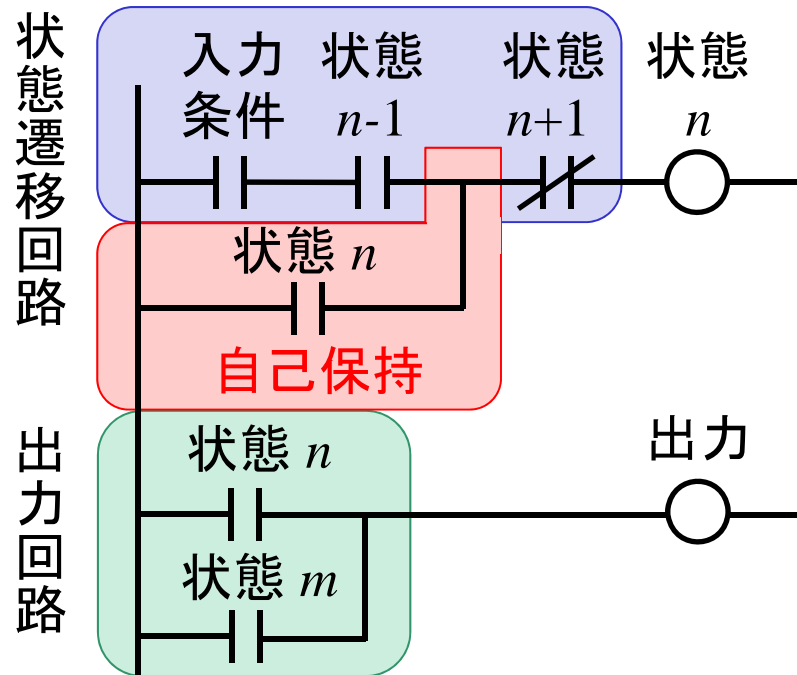
ロータリエンコーダの方向弁別回路

図の4進倍回路は、
A相、B相の立上がり
および立下がりパルス
を利用することにより
回転方向を弁別する



ラダー図の作成

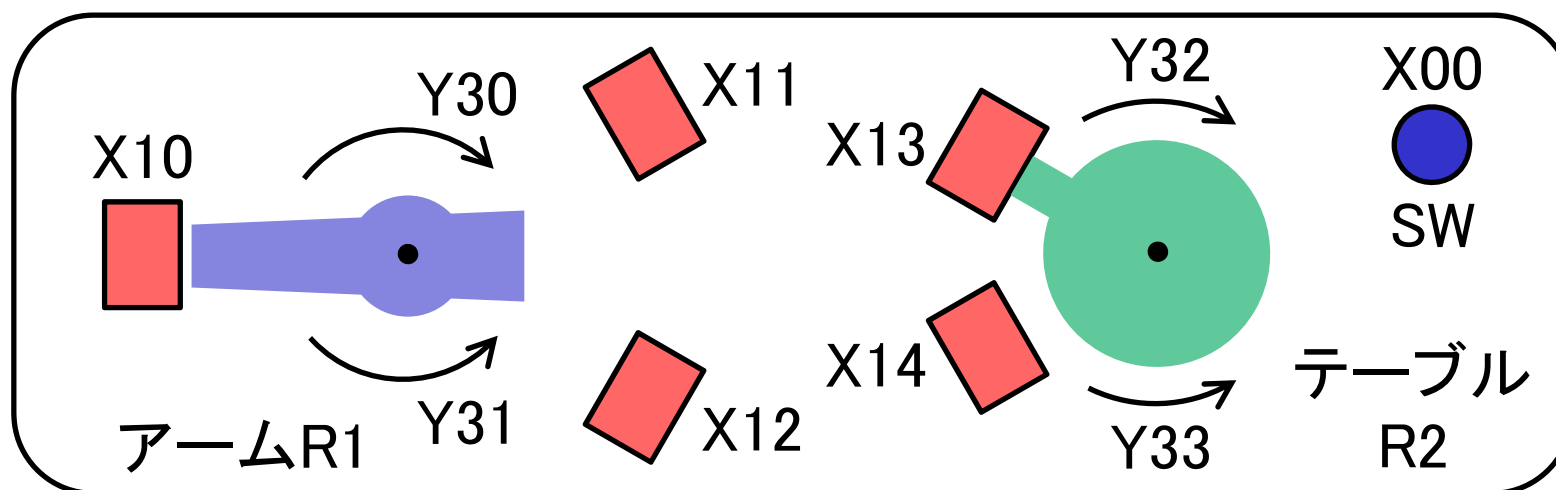
- 状態遷移の考え方をを用いる方法の例
作業行程に分け, それらに状態を割り振る
入力と状態から, 状態遷移回路を作成する
状態(と入力)から, 出力回路を作成する



- ・必要ならばインターロックを入れる
- ・自己保持の結線位置は, 優先順序を考えて変更する
- ・出力がある状態のみで決まる場合などは, 状態を省略できる
- ・複数の条件で出力が決まるときは, 後ろに記述する

例題：制御対象

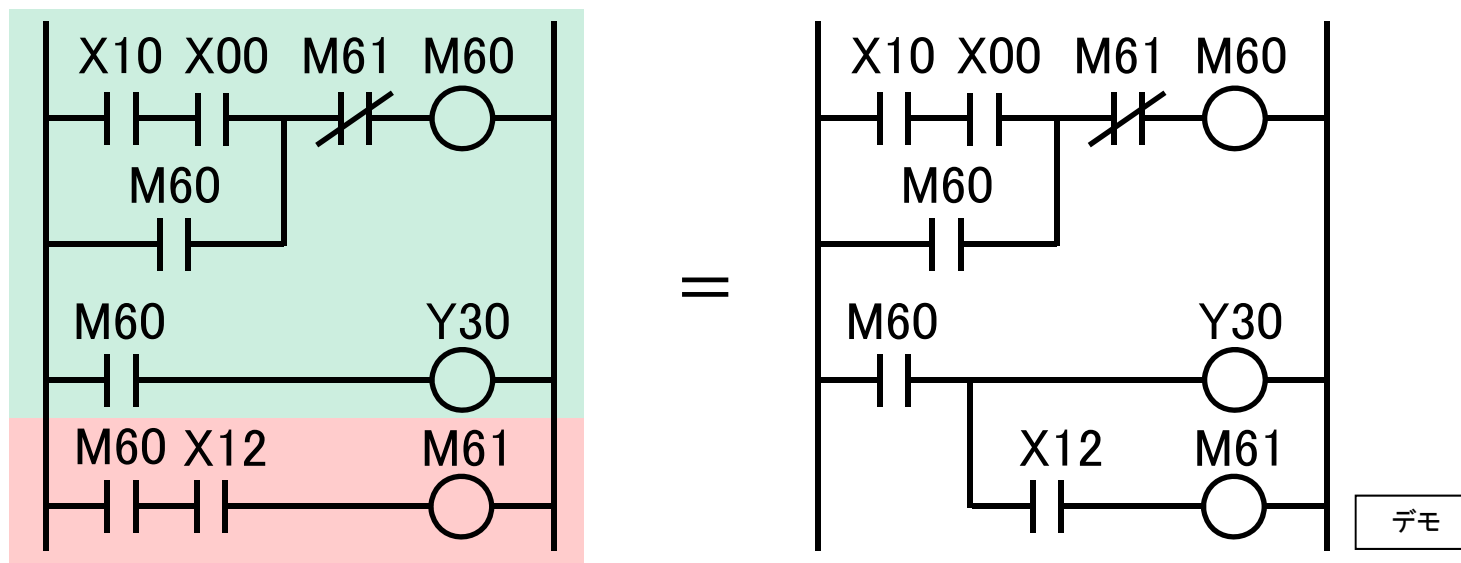
- X00～X02 押しボタンスイッチ
- X03～X05 トグルスイッチ
- X10～X14 センサ
- Y20～Y27 発光ダイオード
- Y30～Y33 アームR1とテーブルR2の回転方向
- M60～ PLCの内部メモリ(内部リレー)



例題1

●回転動作

アームが原位置(X10)にあるとき、押しボタンスイッチ(X00)を押すと、アームが右回転(Y30)をはじめ、右下(X12)まで来ると停止



(原点復帰が必要. M61を用いずにX12のb接点を用いる方法もある)

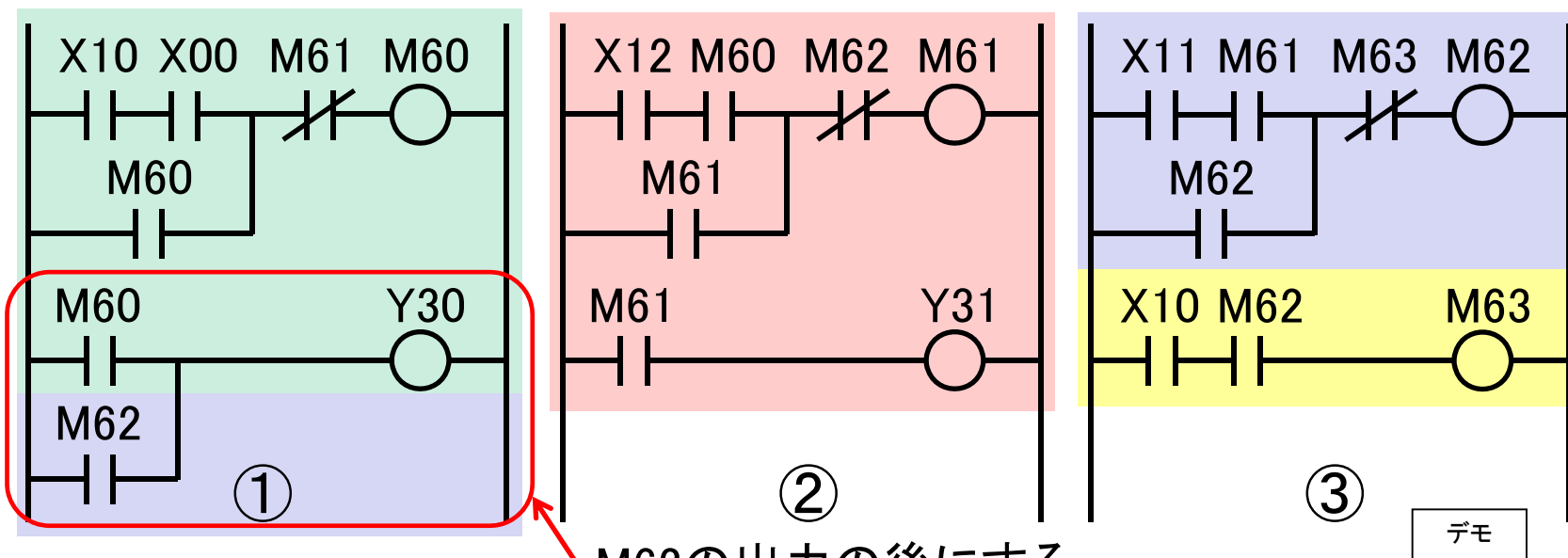
例題2

●往復運動

アームが原位置(X10)にあるとき、押しボタンスイッチ(X00)を押すと、アームが右回転をはじめ、右下(X12)まで来ると停止

アームは直ちに左回転をはじめ、右上(X11)まで来ると停止

アームは直ちに右回転をはじめ、原位置(X10)まで来ると停止



例題3(a)

●同期運動

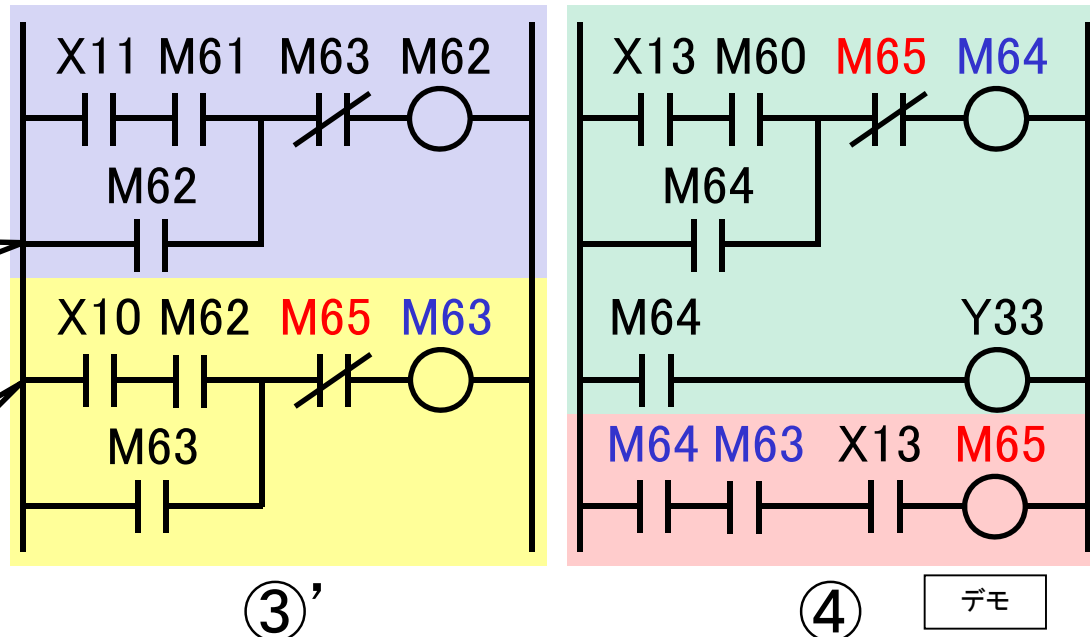
例題2において、アームが動き出すと同時にテーブルも原位置(X13)から左回転をはじめ

アームが原位置(X10)に停止した後も、テーブルは回転を続け、原位置(X13)で停止

例えば、
(a)並進・合流を使うと

例題2の①②と
ここまでは同じ

この部分を変更し、
④を追加



例題3 (b)

●同期運動

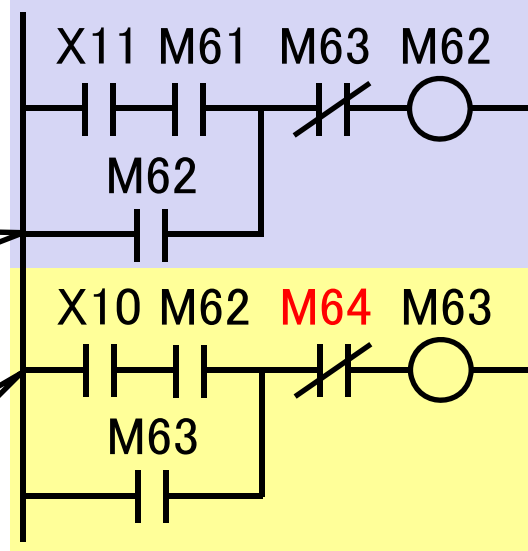
例題2において、アームが動き出すと同時にテーブルも原位置(X13)から左回転をはじめる

アームが原位置(X10)に停止した後も、テーブルは回転を続け、原位置(X13)で停止

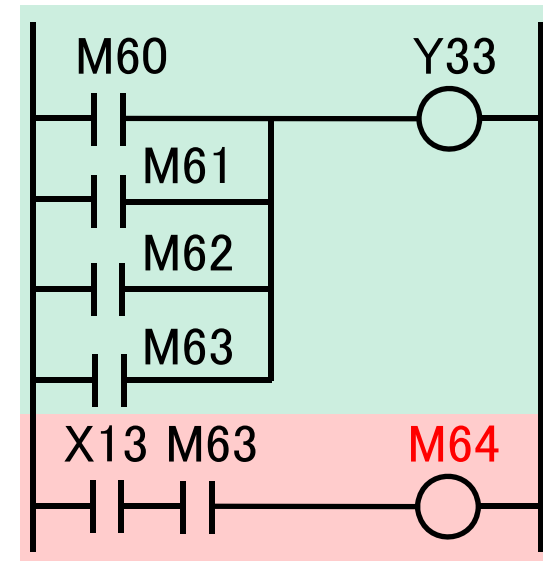
例えば、
(b)2出力を使うと

例題2の①②と
ここまでは同じ

この部分を変更し、
④'を追加



③''



④'

デモ

タイマとカウンタ

●タイマ

タイマの入力コイルをONにすると、時間の計測を開始し、設定された時間に達すると、タイマの出力接点を動作させる

タイマの入力コイルをOFFにすると、リセットされる

…オンディレイタイマ

●カウンタ

カウンタの入力コイルをONした回数を計数して、設定された回数に達すると、カウンタの出力接点を動作させる

カウンタのリセット入力を持つ

●その他

PLCにより、数値データ処理や演算、プログラム制御のできるものもある