電気学第一（計算機の基礎：論理回路）

担当：一色　剛（南３－５１７：内２８４２）

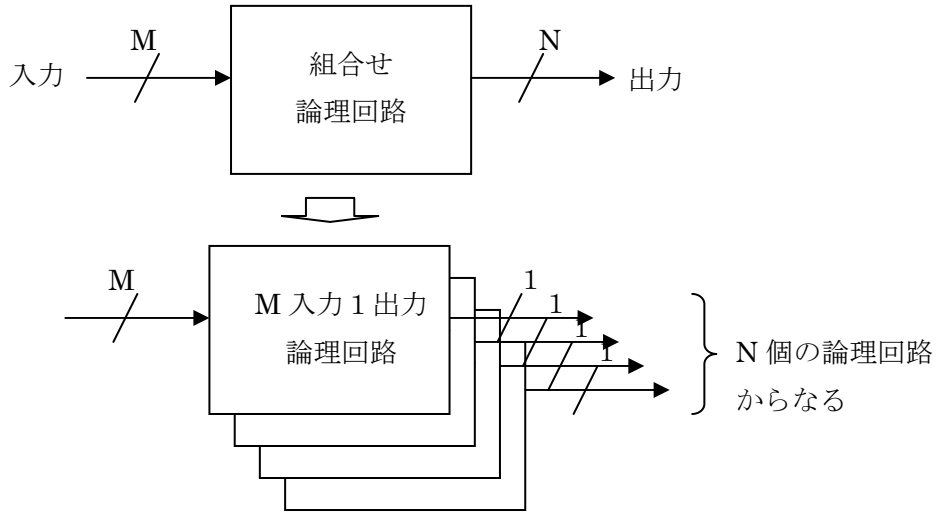
１．論理変数（２値変数）:「０」または「１」の値を取る.

２．組合せ論理回路

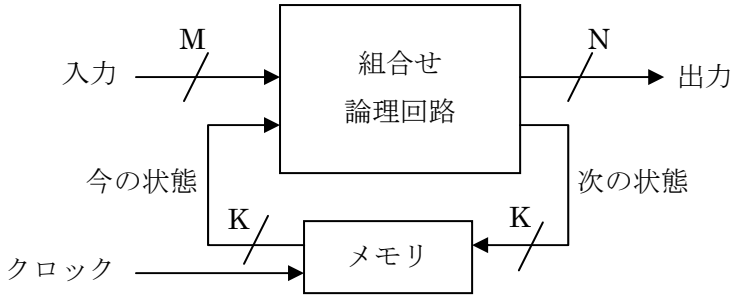
(ア) 入力：M個の２値変数（Mビット）→　2^M通りの入力パターン

(イ) 出力：N個の２値変数（Nビット）→　2^N通りの出力パターン

(ウ) 入出力の写像の定義：論理関数　$\{0,1\}^M \rightarrow \{0,1\}^N$

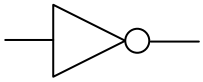


３．順序論理回路：組合せ回路にメモリ素子（状態を記憶）を追加した論理回路



４．基本論理関数及び論理回路

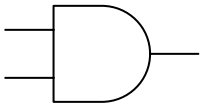
(ア) NOT：$f(A) = \neg A = \overline{A}$　（論理否定）



真理値表

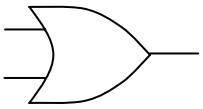
A	$\overline{A}$
0	1
1	0

(イ) AND：$f(A, B) = A \wedge B = A \cdot B = AB$　（論理積）



A	B	A・B
0	0	0
0	1	0
1	0	0
1	1	1

(ウ) OR：$f(A, B) = A \vee B = A + B$　（論理和）



A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

５．論理関数の諸定理

(ア) 吸収則：$A + 1 = 1$

$A \cdot 0 = 0$

$A + 0 = A$

$A \cdot 1 = A$

(イ) 補元則：$A + \overline{A} = 1$

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

(ウ) 結合即：$A+(B+C) = (A+B)+C$

$A(BC) = (AB)\,C$

(エ) 分配則：$A(B+C) = AB + AC$

$AB+C = (A+C)\,(B+C)$

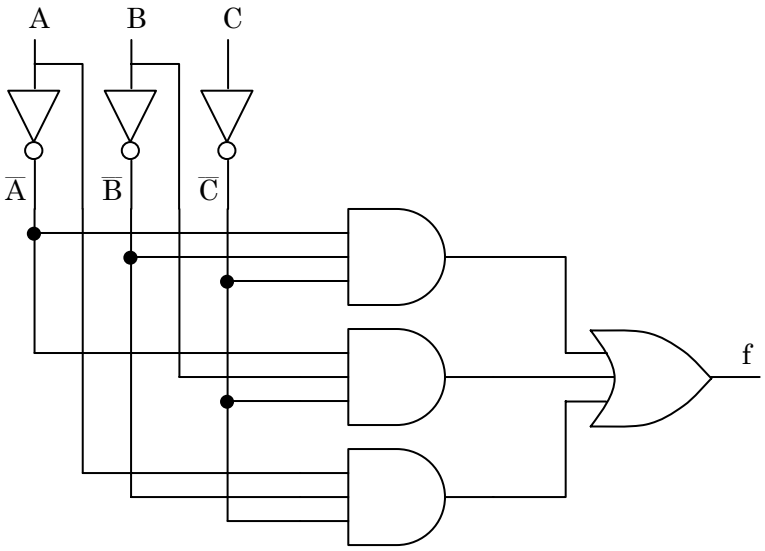
(オ) ド・モルガンの定理：$\overline{(A+B)} = \overline{A}\,\overline{B}$

$\overline{(A\,B)} = \overline{A}+\overline{B}$

６．任意の論理関数の実現方法

A	B	C	f(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

(ア) 論理和標準形：すべての入力変数の肯定形又は否定形の論理積からなる項を論理和で結ぶ．→　$f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C}$

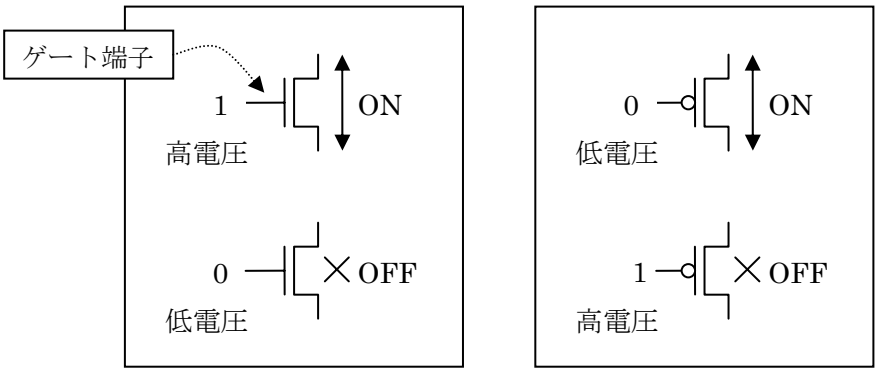


(イ) 論理最適化：論理和標準形から，論理関数の項の数を減らして，論理回路の規模をなるべく小さくする．→

$$
\begin{aligned}
f(A,B,C) &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} \\
&= (\overline{A}\,\overline{B} + \overline{A}\,B + A\,\overline{B})\,\overline{C} && (\overline{C}\text{で因数分解}) \\
&= (\overline{A}\,(\overline{B} + B) + A\,\overline{B})\,\overline{C} && (\overline{A}\text{で部分的に因数分解}) \\
&= (\overline{A} + A\,\overline{B})\,\overline{C} && (\because \overline{B} + B = 1) \\
&= (\overline{A} + \overline{A}\,\overline{B} + A\,\overline{B})\,\overline{C} && (\because \overline{A} = \overline{A} + \overline{A}\,\overline{B}) \\
&= (\overline{A} + (\overline{A} + A)\,\overline{B})\,\overline{C} && (\overline{B}\text{で部分的に因数分解}) \\
&= (\overline{A} + \overline{B})\,\overline{C} && (\because \overline{A} + A = 1)
\end{aligned}
$$

７．トランジスタによる論理回路の実現

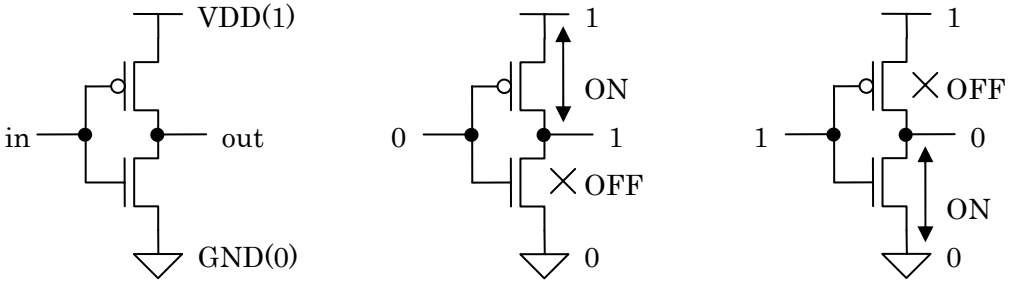
(ア) 2 種類の MOS トランジスタ（スイッチとして動作）



nMOS トランジスタ　　pMOS トランジスタ

互いに逆の動作をする nMOS トランジスタと pMOS トランジスタを両方用いて論理回路を実現する　→　CMOS 論理回路（Complementary MOS）

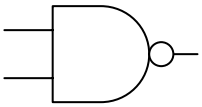
(イ) CMOS 否定（NOT）論理回路



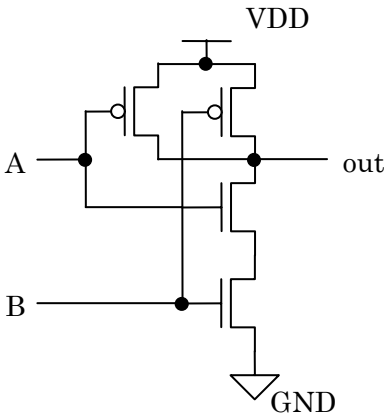
CMOS 否定論理回路　　論理０入力　　論理１入力

８．その他の組合せ論理回路

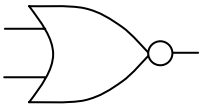
(ア) NAND：f(A, B) = ¬(A∧B) = $\overline{(AB)}$



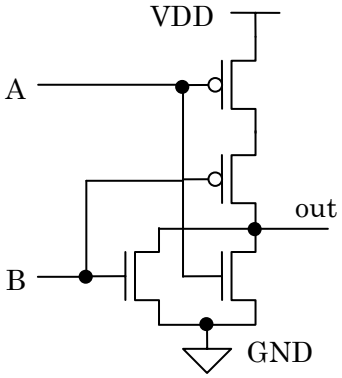
A	B	$\overline{(AB)}$
0	0	1
0	1	1
1	0	1
1	1	0



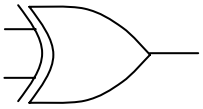
(イ) NOR : $f(A, B) = \neg(A \vee B) = (\overline{A+B})$



A	B	$(\overline{A+B})$
0	0	1
0	1	0
1	0	0
1	1	0



(ウ) EXOR : $f(A, B) = A \oplus B$　（排他的論理和）



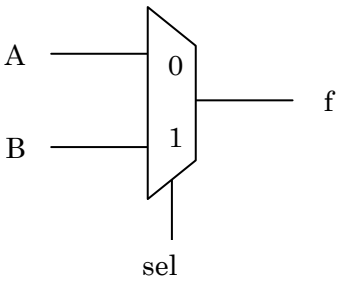
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(エ) マルチプレクサ（選択回路）:

$f(A, B, sel)$ $= A$ (sel = 0 の場合)

$= B$ (sel = 1 の場合)

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

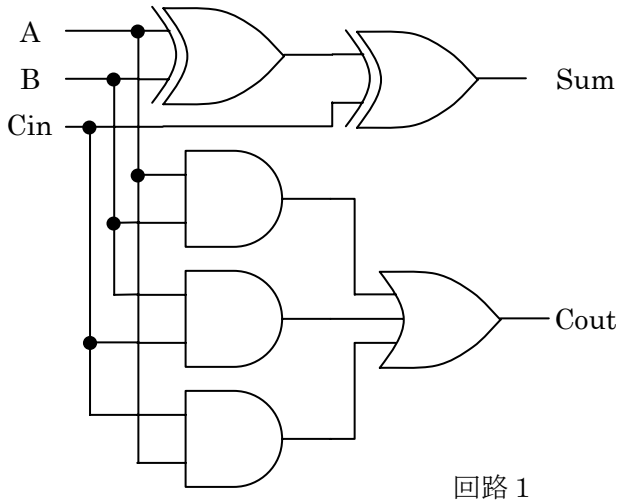


A	B	sel	f(A, B, sel)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

(オ) 1 ビット加算器：

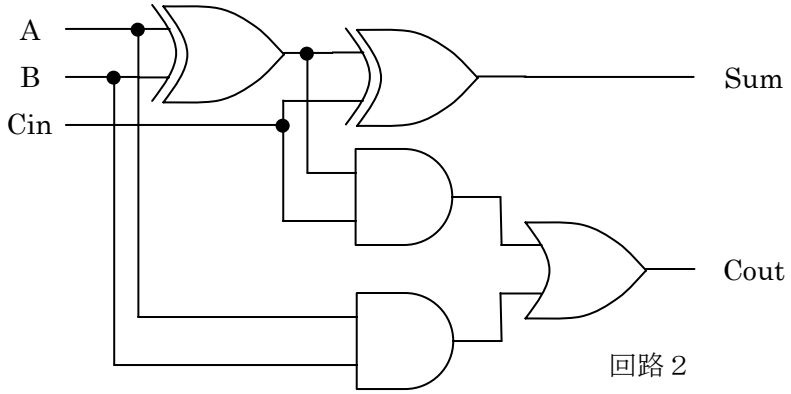
入力：A，B，cin（キャリー入力）

出力：sum，cout（キャリー出力）



回路 1

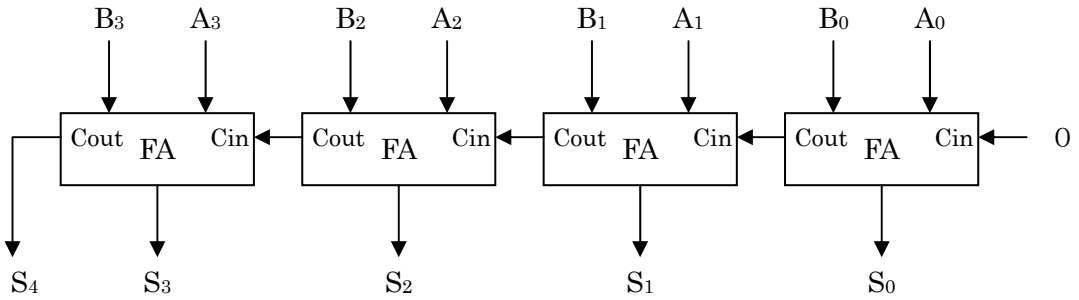
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



回路 2

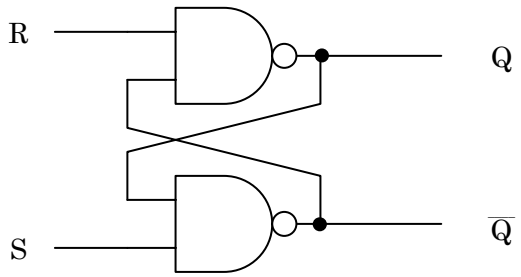
０１０１（５）＋００１１（３）＝１０００（８）

```
    ０１０１（５）
＋　００１１（３）
    １０００（加算結果）
    １１１０（キャリー）
```



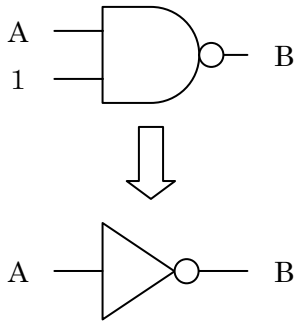
９．メモリ回路

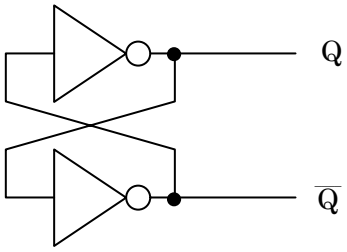
(ア) RS フリップフロップ：



R	S	Q	$\overline{Q}$	状態
0	0	1	1	禁止入力
0	1	1	0	「1」書込み
1	0	0	1	「0」書込み
1	1	Q	$\overline{Q}$	データ保持

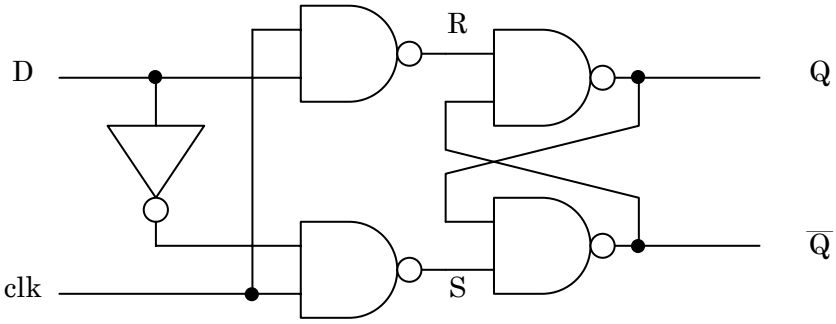
RS フリップフロップ





R=S=1の時のRSフリップフロップの等価回路

(イ) D フリップフロップ：

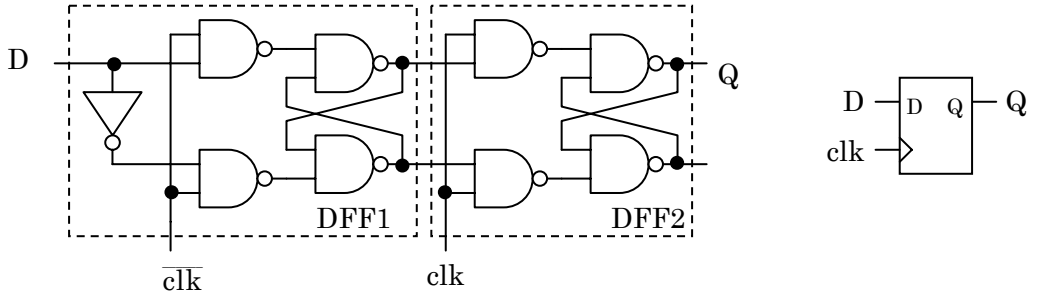


clk＝０：データ保持.

clk＝１：Q=D

clk	D	R	S	Q	$\overline{Q}$	状態
0	0	1	1	Q	$\overline{Q}$	データ保持
0	1	1	1	Q	$\overline{Q}$	データ保持
1	0	1	0	0	1	「0」書込み
1	1	0	1	1	0	「1」書込み

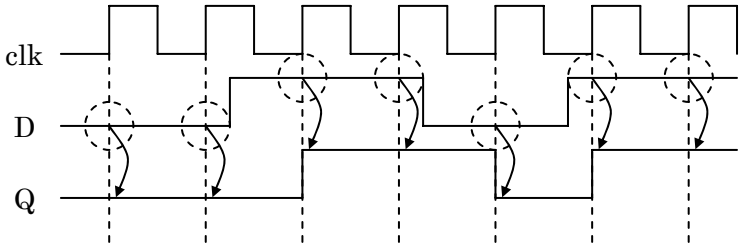
(ウ) マスタスレーブ型（エッジトリガー型）D フリップフロップ：

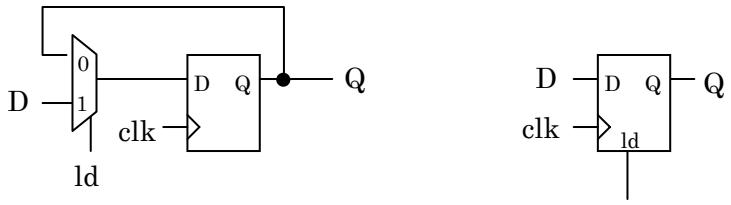


・clk＝０の場合：DFF1 はデータ書込み，DFF2 はデータ保持

・clk＝１の場合：DFF1 はデータ保持，DFF2 はデータ書込み

→　clk が０から１に立ち上がる時（クロックエッジ）の D の値が Q に出力





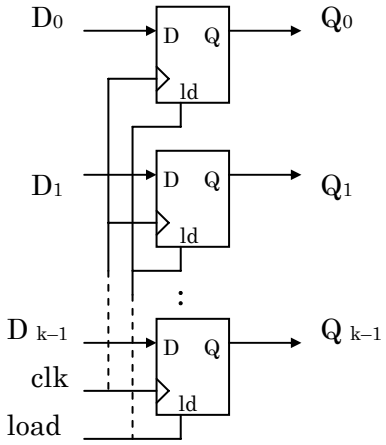
ロード制御付きマスタスレーブ型

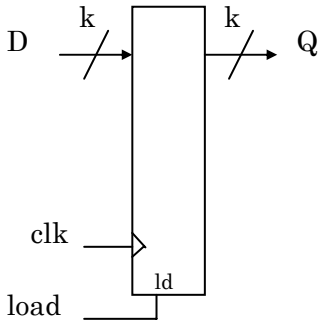
D フリップフロップ

・ld＝1 の場合：D 入力をクロックエッジで取り込む

・ld＝0 の場合：データ保持を続ける

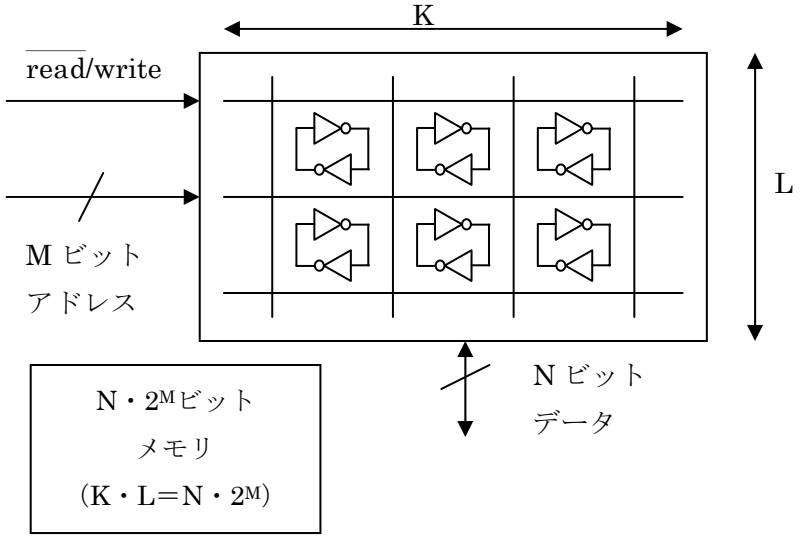
(エ) レジスタ：多ビットデータを一時的に記憶する





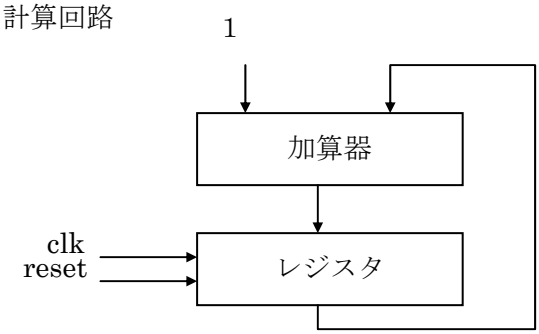
k ビットレジスタ

(オ) メモリ：大量のデータを記憶する（書込み・読出しは１データずつ行う）



１０．　順序回路を用いた計算回路

(ア) カウンタ：



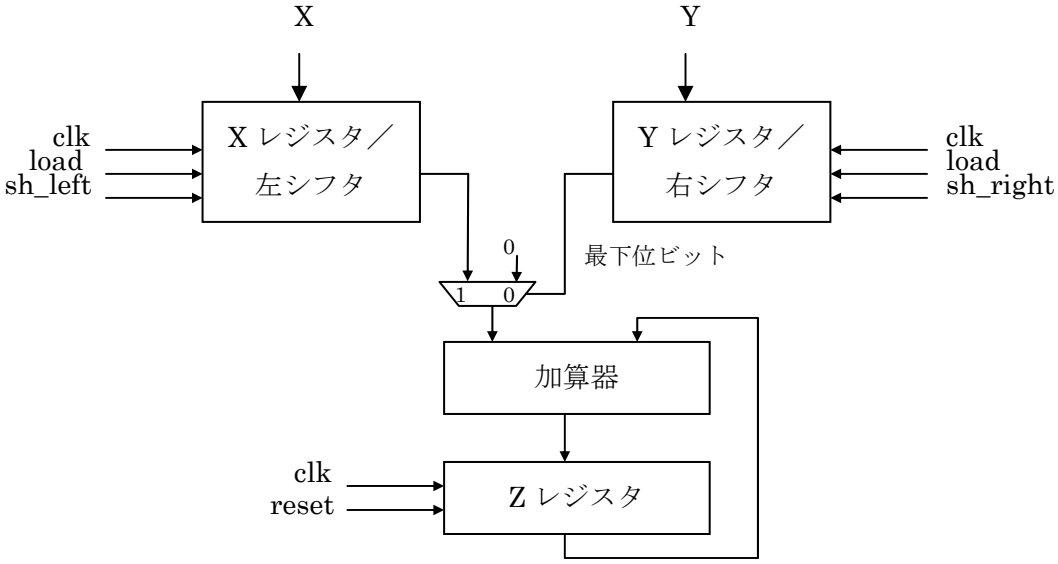
✓　動作：

①　レジスタ値を０にする（reset=1）。

②　その後、clk が 0→1 に変化する度にレジスタの値が１ずつ増える。

(イ) 乗算器：

		X	Y	Z
	０１１１（７）　：X			
×	１１１０（１４）：Y			
	００００	００００１１１	１１１０	００００００００
	０１１１	００００１１１０	０１１１	００００１１１０
	０１１１	００１１１００	００１１	００１０１０１０
	０１１１	００１１１０００	０００１	０１１０００１０
	０１１００００１０（９８）：Z			



✓　動作：

①　Z レジスタ値を０に（reset=1）、X レジスタ・Y レジスタにそれぞれ X 値・Y 値を読み込む(load=1)。（X レジスタの上位ビットは「０」で埋める）

②　その後、Y レジスタの値が「０００００」になるまで、clk が 0→1 に変化する度に以下の動作を繰り返す

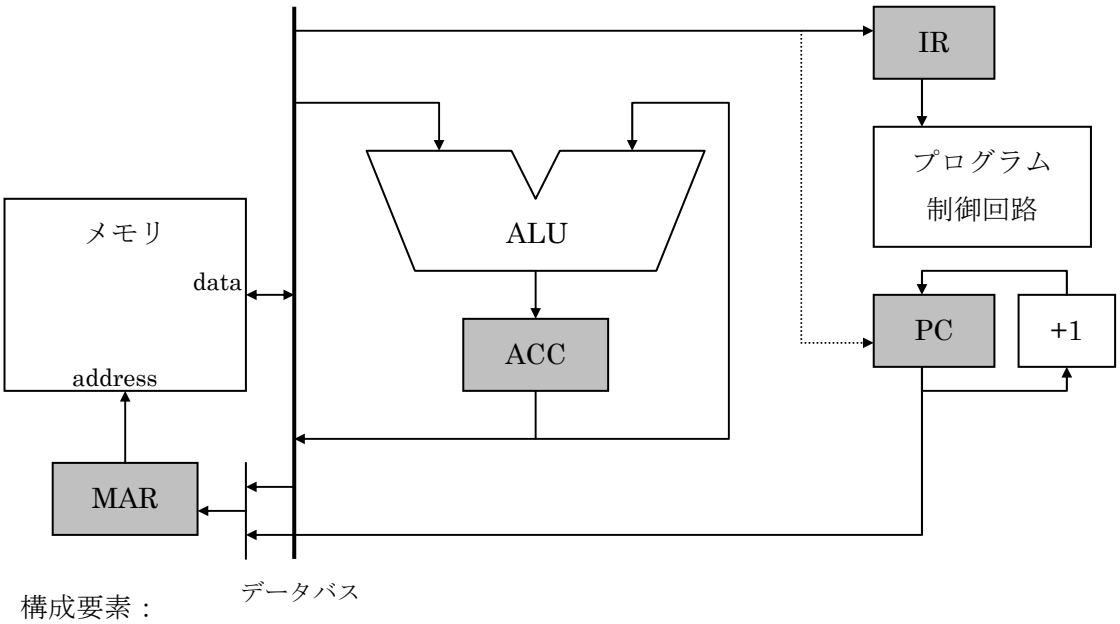
(ア) Y レジスタの最下位ビットが「０」の場合、Z レジスタは変化しない。

(イ) Y レジスタの最下位ビットが「１」の場合、Z レジスタに X レジスタの値が加算される。

(ウ) X レジスタは左に１ビットシフトされる（最下位ビットには「０」が入る）

(エ) Y レジスタは右に１ビットシフトされる（最上位ビットには「０」が入る）

(ウ) マイクロプロセッサ：



構成要素：

① メモリ：プログラム（命令）と演算結果（データ）を保持する（”address”で指定されたメモリ番地に”data”の値を書き込む、又は、読み出す）

② ALU（算術・論理演算ユニット）：1 つ又は 2 つの入力から出力を計算する

③ レジスタ：

1. IR（インストラクションレジスタ）：現在実行する命令を保持する
2. PC（プログラムカウンタ）：現在の命令格納番地を指定する
3. ACC（アキュムレータ）：ALU の出力を蓄える
4. MAR（メモリアドレスレジスタ）：メモリの番地を指定する

④ プログラム制御回路：IR の命令から各回路ブロックへの制御信号を生成する

1. ALU の制御信号：実行する演算を指定（加算・減算・各種論理演算）
2. レジスタの制御信号：値の書き込み許可・不許可を指定
3. メモリの制御信号：書き込み/読み出しを指定
4. データバス制御信号：転送元信号と転送先信号を指定

✓ 動作例:100 番地のデータに 10 を加算して、その結果を 101 番地に書き込む。

1. LOAD 100　（100 番地のデータを読み出し、ACC に格納する）
2. ADD #10　（ACC に 10 を足し、ACC に格納する）
3. STORE 101　（ACC の値を 101 番地に書き込む）