

# デジタル回路 ヴェット (5)

第4章 p. 45-62

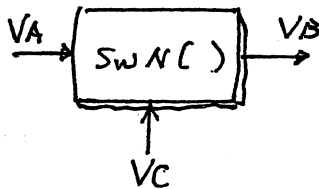
\*\*\*\*\*  
デジタル回路 演習問題 05  
\*\*\*\*\*

- 5.01 N-type Switch 回路  $swN()$  を定義せよ。  
NMOS( $\quad$ ) のtransistor 1個で構成した場合の問題点を説明せよ。
- 5.02 P-type Switch 回路  $swP()$  を定義せよ。  
PMOS( $\quad$ ) のtransistor 1個で構成した場合の問題点を説明せよ。
- 5.03 CMOS Switch 回路  $CswN()$  と  $CswP()$  について説明せよ。
- 5.04 BCD Encoder 回路  $BCDencd()$  を設計せよ。  
ただし、入力信号  $A[ ]$  が all zero の時、出力信号  $B[ ]=(1111)$  とし、  
入力信号  $A[ ]$  のうち、2つ以上の信号線が high の時は  $B[ ]=(1110)$  を出力する。  
最後に BCD Decoder 回路  $BCDdecdec()$  の設計手順について説明せよ。
- 5.05 Multiplexer 回路  $MULTP()$  を設計せよ。
- 5.06 Demultiplexer 回路  $DEMULTP()$  を設計せよ。
- 5.07 n-bit の一致回路  $CMPN()$  を設計せよ。
- 5.08 わり算回路  $WARU()$  回路を設計せよ。  
2つの入力信号ベクトルを n-bit の  $A[ ]$  と  $B[ ]$  とし、  
わり算の計算結果を  $A[ ]/B[ ] = C[ ], D[ ]$  とする。  
有効桁数を  $kMax$  とする。

\*\*\*\*\*

デジタル回路  
PC. 31  
第4章  
デジタル

5.01 Switch N 回路. N-type (NMOS) switch 回路 swN() 回路の性質

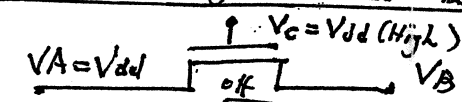
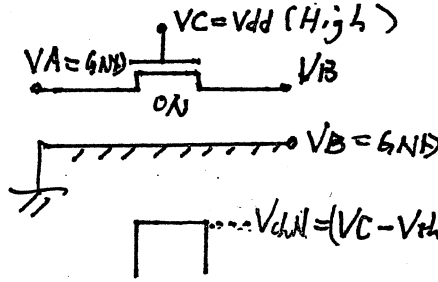


制御信号  $V_C = \text{Low}$  の時、Switch が off.  
 $V_A$  の信号は  $V_B$  に伝達されない。  
 (Switch = ON ( $V_C = \text{High}$ ) にすると、  
 $V_A$  の信号が  $V_B$  に伝達される。)

しかし、swN() 回路を NMOS Transistor 1 個で構成すると、問題あり!

$V_A = \text{GND}$  の時は  $V_B = \text{GND}$  にする

しかし、 $V_A = V_{dd}$  の時は、 $V_B = V_{dd} - V_{thN}$  とする

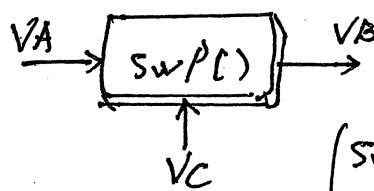


$V_{ch} = V_{gate} - V_{thN}$

完全に  $V_B$  は  $V_{dd}$  に近づかず!!  
 $V_{thN}$  の問題!!

NMOS の場合、gate 電圧が  $V_{dd}$  に近づくと、Channel 電位  $V_{ch} : (V_{dd} - V_{thN})$  とする

5.02 Switch P 回路. P-type (PMOS) switch 回路 swP() 回路の性質

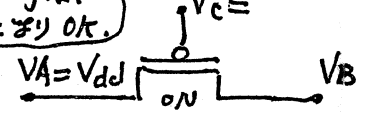
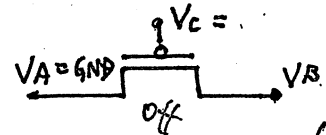


N-type Switch 回路とは逆に、  
 制御信号  $V_C = \text{High}$  の時、Switch が off.  
 $V_A$  の信号は  $V_B$  に伝達されない。  
 (Switch = OFF = GND ( $V_C = \text{Low}$ ) にすると、  
 $V_A$  の信号が  $V_B$  に伝達される。)

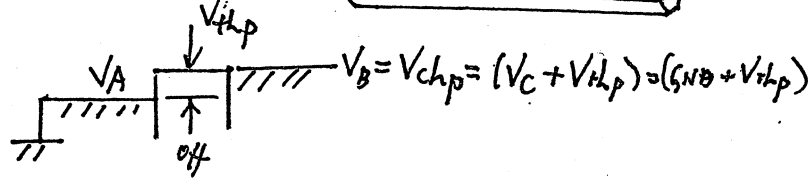
しかし、swP() 回路を PMOS Transistor 1 個で構成すると、問題あり

$V_A = \text{GND}$  の時は、 $V_B = \text{GND} + V_{thP}$  とする!!

$V_A = V_{dd}$  の時は、  
 $V_B = V_{dd}$  とする OK.

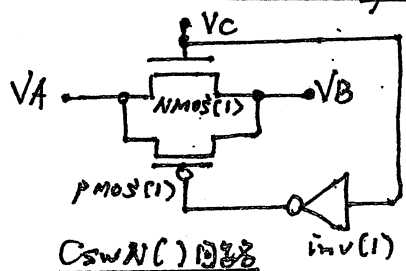


$V_{chp} = V_{gate} + V_{thP}$

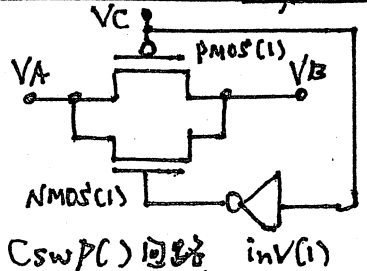


ON ...  $V_{chp} = \text{GND} + V_{thP}$

5.03 さて CMOS switch N-type 回路 CswN() と P-type 回路 CswP() を使う。

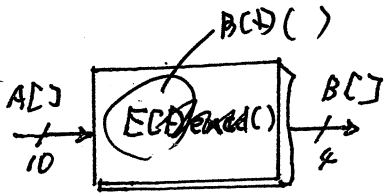


CswN() 回路 inv(1)



CswP() 回路 inv(1)

「デジタル回路の世界」  
 (第4章 pp. 319~323)  
 参考のこと。

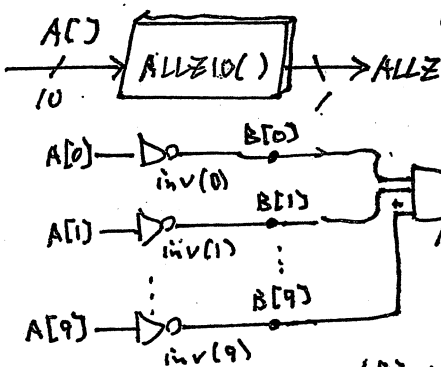


(A[10] の信号は 10 本あるが、そのうちの 1 本だけが High になる。2 本以上は High になる事は無い。)  
 (しかし、例外もある... A[10] の入力が高レベル (Low) の場合は、BCD = (1111)<sub>2</sub> を出力する事になる。

A[10] の位の意味	BCD の値
A[0] が High の時 0 を意味する	0000
A[1] が High の時 1 を意味する	0001
...	...
A[8] が High の時 8 を意味する	1000
A[9] が High の時 9 を意味する	1001

また、A[10] が 2 以上 High になる時は、BCD = (1110)<sub>2</sub> を出力する事になる...

Step ① A[10] の入力が高レベル 0 の時、1 を出力する回路 ALLZ10() を作る。



```
define ALLZ10() { input A[10]; output ALLZ10;
    for (k=0 to 9) A[k] inv(k) -> B[k];
    B[0] AND10(1) -> ALLZ10 }
```

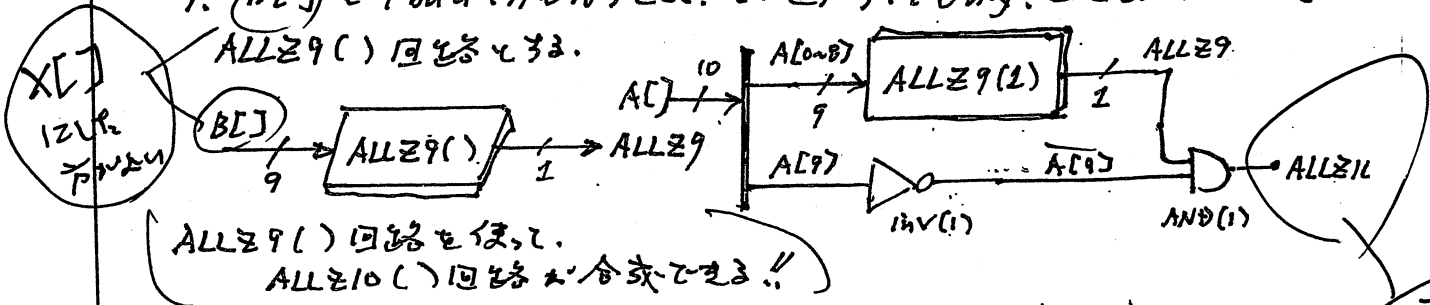
(同様に ALLZ9() も合成できる)

(AND10() 回路は、入力端子が 10 個ある AND 回路のことになる。)

A[10] が高レベル 0 なら、BCD は高レベル 1 になる!

Step ② A[10] の入力のうち、1 つだけ 1 を出力している時のみ、1 を出力する回路 ONEA() を作る。

今、BCD を 9 bit のバケツ信号として、その値が高レベル 0 の時、1 を出力する回路 E ALLZ9() 回路を作る。



(ALLZ9() 回路を使う。 ALLZ10() 回路を合成できる!!)

次に、A[k] = 1 となる (k=0, 1, 2, ..., 9)。A[k]=1 の時、残り 9 個の信号が高レベル 0 である時、ONEA() 回路は 1 を出力する。すなわち、

$$ONEA = \sum_{k=0}^9 A[k] \{ \overline{\{A[10] - A[k]\}} ALLZ9(k) \}$$

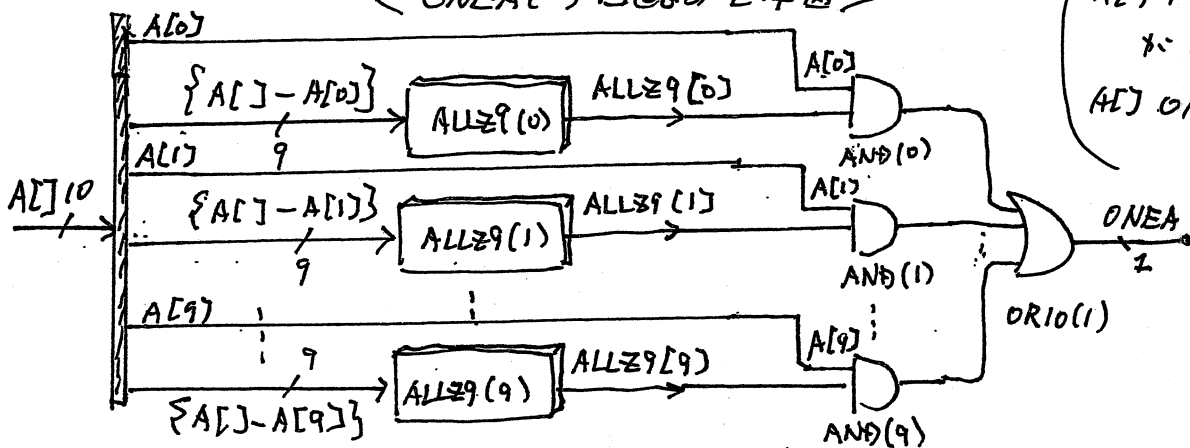
ここで、{A[10] - A[k]} とは、10 本の信号線 A[10] から A[k] を取り除いた残り 9 本の信号線の束 (バケツ) を意味する。9 本ずつ、ALLZ9() 回路の入力信号とすることができる。その出力は、ALLZ9[k] とする (for k=0 to 9)。そして次に、ALLZ9[k] と A[k] と AND(k) 回路を通し、最後に、OR10() 回路を通過させる。

9本

ALLZ10

OR10

< ONEA( ) 回路の全体図 >



(A[7]より2桁のみ  
は High の位  
A[i] ONEA( )  
は 1 と 2)

ALLZ9( ) 回路を 10 個、AND( ) 回路も 10 個、そして、10 入力 OR 回路 OR10( ) を 1 個使う!!

step ③ λ 入力信号 A[i] と出力信号 B[i] の全体比較表、

λ 入力	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	B[3]	B[2]	B[1]	B[0]
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	0
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	0	0	0	1	0	0
7	0	0	1	0	0	0	0	0	0	0	0	1	1	0
8	0	1	0	0	0	0	0	0	0	0	0	1	0	0
9	1	0	0	0	0	0	0	0	0	0	0	1	0	0
ALLZ10 = 1 の時、											1	1	1	1
ONEA = 0 の時											1	+	1	0

B[3] の回路

$$B[3] = A[9] + A[8] + ALLZ10 + \overline{ONEA}$$

B[2] の回路

$$B[2] = A[4] + A[5] + A[6] + ALLZ10 + \overline{ONEA}$$

B[1] の回路

$$B[1] = A[2] + A[3] + A[6] + A[7] + ALLZ10 + \overline{ONEA}$$

B[0] の回路

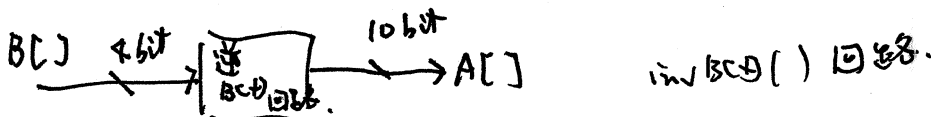
$$B[0] = A[1] + A[3] + A[5] + A[7] + A[9] + ALLZ10$$

(B[3] は 4 入力 OR 回路 OR4(1) を使う。  
B[2] は 5 入力 OR 回路 OR5(1) を使う。  
B[1] は 6 入力 OR 回路 OR6(1) を使う。  
B[0] は 6 入力 OR 回路 OR6(2) を使う。

(OR6( ) 回路は OR6(1) と OR6(2) の 2 個 Copy して使う。  
ONEA 信号は ONEA 信号を 1 個使う。  
を 入力 して、inv( ) 回路を 1 個使う。

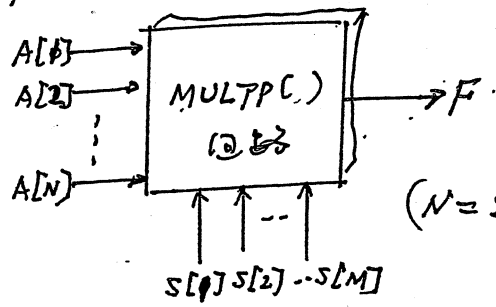
\* B[i] 信号から元の信号 A[i] を復元する回路

たとえば、A[0] は B[i] の信号がすべて 1 であるとき、inv( ) 回路を 4 つ使って B[i] を作り、その各信号 4 本を λ 入力 4 入力 AND 回路 AND4( ) 回路を 1 個使えばいい。同様に、A[1] 作り、inv( ) 回路を 3 つ使う。B[3], B[2], B[1] を作り、この 3 つの信号と B[0] の信号の、合計 4 つの信号を 4 入力 AND 回路に入力して、A[1] 信号が出力される。

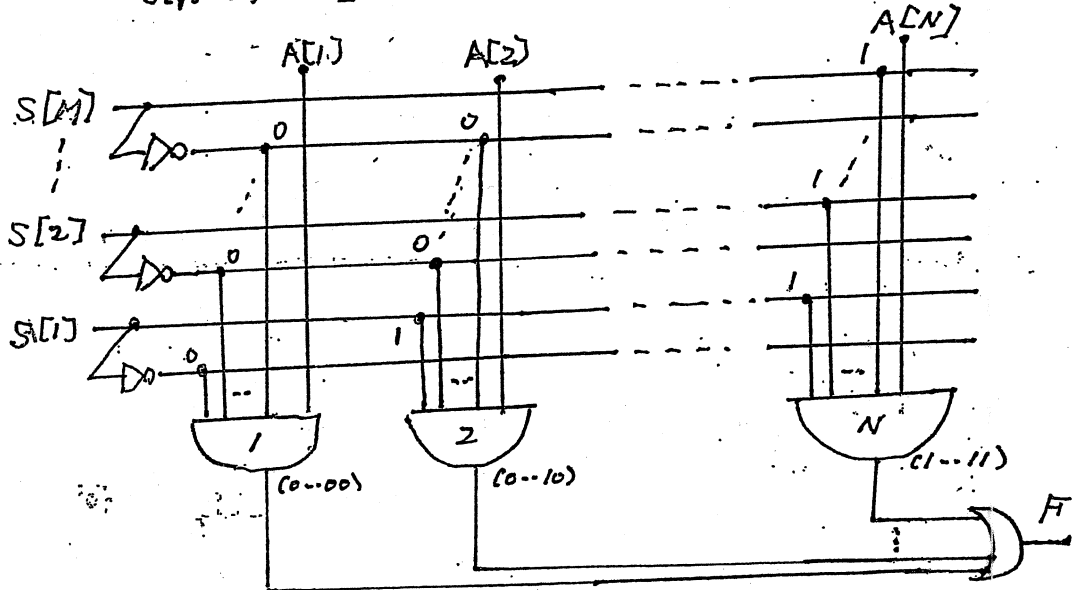


p.65  
図 5-3  
参考

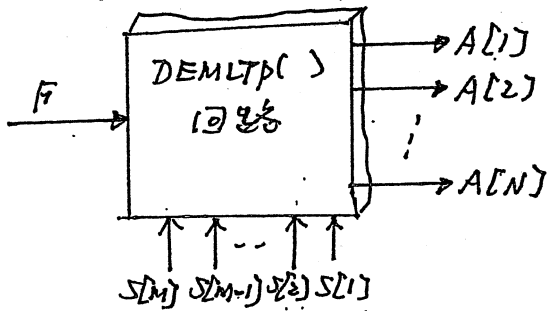
5.05 Multiplexer 回路 MULTP(.) の設計



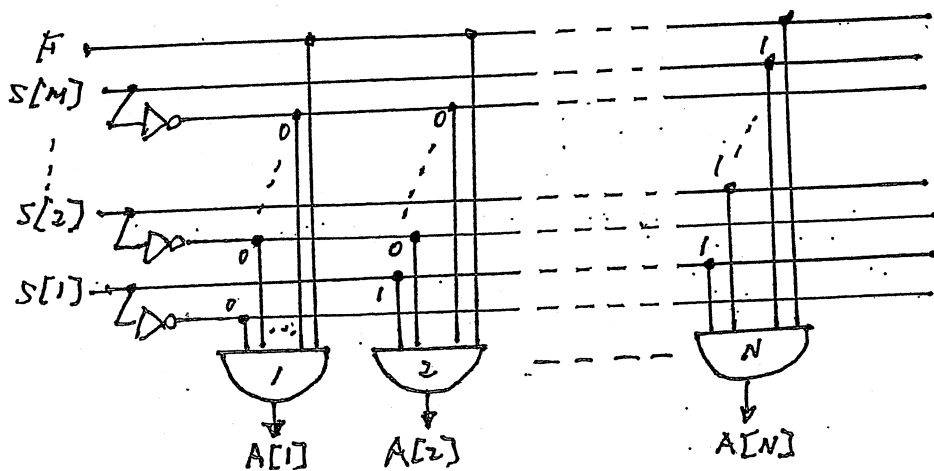
2進法数  $(S[M], S[M-1], \dots, S[2], S[1])_2$   
 が 10進法数  $(i)_{10}$  に対応する  
 時、 $A[i]$  の信号が  $F$  の出力  
 端子に伝達される。



5.06 Demultiplexer 回路 DEMLTP(.) の設計

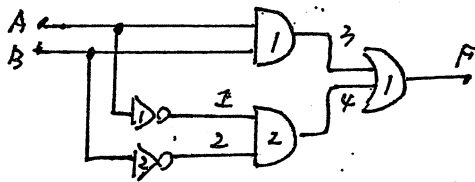
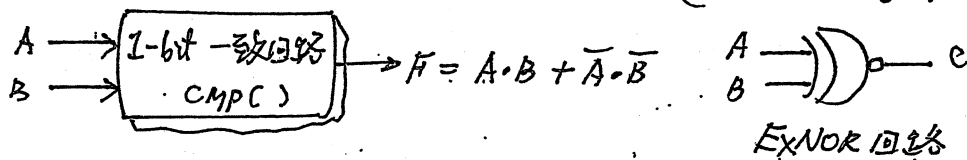


2進法数  $(S[M], \dots, S[2], S[1])_2$   
 が 10進法数  $(i)_{10}$  に対応する  
 時、 $F$  の信号が  $A[i]$  の出力信号  
 端子に出力される。



7/27  
 の  
 11.35  
 分

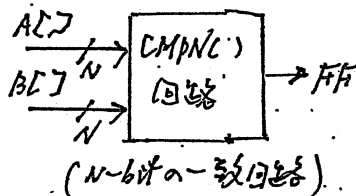
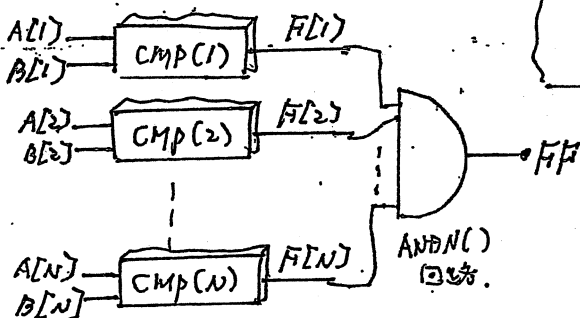
5.07 一致回路 (Exclusive NOR 回路) の設計 (これは普通比較(一致)回路)



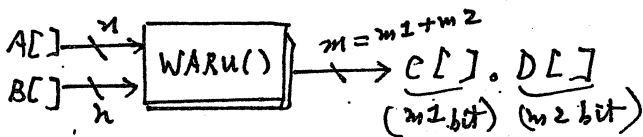
```
define EXNOR(C) {
    input A, B;
    output C;
    [A] inv(1) → [1];
    [B] inv(2) → [2];
    [A][B] AND(1) → [3];
    [1][2] AND(2) → [4];
    [3][4] OR(1) → [F];
}
```

n-bit の一致回路の設計

CMPN( ) 回路



5.08 割り算回路 WARU( ) の設計



<「デジタル回路の世界」pp.350-352参照>

A[] と B[] は n ビットの 2 進法数  
 として、割り算をする。  
 小数にする... 小数点以下の  
 値も D[] として計算したい...

まず、A[] がゼロの場合、B[] もゼロなら 0/0 は NaN = Not a Number  
 A[] がゼロで、B[] がゼロでないなら C[] も D[] もゼロ。

それ以外の場合を考慮。A[] の中には必ず 1 が 1 つはある。  
 B[] の中には必ず 1 が 1 つある場合を考慮。そして 左 B 位 shift をして  
 A[] の最上 A[n-1] に 1 がくるようにする。また B[] の最上にも 1 が来る。  
 B[n-1] が 1 になるように 左 B 位 shift する。A[] は NA 位 左 に shift  
 したとすると、B[] は NB 位 左 に shift したとすると、A[] と B[] の値を  
 step①前にもとに戻すには、(NB - NA) 位 左 に shift すればいい。

Step ① まず、A[] - B[] = R[] を計算する。R[] が正なら { C[0] = 1; A[] = R[] } とする。  
 R[] が負なら { C[0] = 0 } とする。

いすれにせよ、r = 1 として、A[] を 2 倍する。(左) に 1 位 shift する。

Step ② 小数点から右の位 D[r] を次に求める。A[] - B[] = R[] を  
 計算し、R[] が正なら { D[r] = 1; A[] = R[] } とする。  
 R[] が負なら { D[r] = 0 } とする。いすれにせよ、r = r + 1 として、  
 次の位を 1 つ右にずらす。A[r] を 2 倍する。(左) に 1 位 shift する。

Step ③ 次の位が MAX に達したら、Step ② をくりかえす。最後に Step ① を完了!

回路結果