

第5章 pp. 63-82

デジタル回路 演習問題 06

6.01 4 bit Encoder 回路 ECD4()を設計せよ。

```
define ECD4( ) { input A[ ]; output B[ ];  
                A[1]A[3]OR(1)->B[0];A[2]A[3]OR(2)->B[1]; }
```

6.02 2 bit Decoder回路 DECD2()を設計せよ。

```
define DECD2( ) { input B[ ]; output A[ ];  
                B[0]inv(1)->BB[0]; B[1]inv(2)->BB[1];  
                BB[0] B[1]AND(0)->A[0] ;B[0]BB[1]AND(1)->A[1] ;  
                BB[0] B[1]AND(2)->A[2] ;B[0] B[1]AND(3)->A[3] ; }
```

6.03 7 segment LED回路の segment(a)を出力する回路 7SegA()を設計せよ。

```
define 7SegA( ) { input B[ ]; output a ;  
                B[2]inv(1)->BB[2]; B[0]inv(2)->BB[2];B[2] B[0]AND(1)->G[1] ;  
                BB[2]BB[0]AND(2)->G[2]; B[3]B[1]C[1]C[2]OR4(1)->a ; }
```

6.04 3入力Multiplexer回路 MLTX3()回路を設計せよ。

```
define MLTX3( ) { input A[ ]; output F;  
                S[0]inv(1)->BS[0]; S[1]inv(2)->BS[1];  
                A[0]BS[1]BS[0]AND3(1)->C[1];A[1]BS[1] S[0]AND3(2)->C[2];  
                A[2] S[1] S[0]AND3(3)->C[3]; C[1]C[2]C[3]OR3(1)->F; }
```

6.05 8 bit 一致判定回路 Match8()を設計せよ。そのDCDL codeを示せ。

6.06 1 bit比較判定回路 Comp()を設計せよ。

```
define Comp( ) { input A,B; output C[ ];  
                [A]inv(1)->[BA];[B]inv(2)->[BB];  
                [A][BB]AND(1)->C[1]; [B][BA]AND(2)->C[2]; }
```

6.07 けた借り型の 1 bit 減算回路 HS()を設計せよ。

```
define HS( ) { input X,Y; output K,Z;  
                [X]inv(1)->[BX]; [BX][Y]AND(1)->[K]; [X][Y]EXOR(1)->[Z]; }
```

6.08 1 bit の Dynamic Shift Register 回路 DSR()を設計せよ。

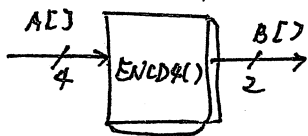
```
define DSR( ) { input Vin,Write,Read ; output Vout;  
                [Vin][Write]swN(1)->[VA]; CA(VA,GND);  
                [VA]inv(1)->[VB]; CB(VB,GND);[VB][Read]swN(2)->[Vout]; }
```

7段のDynamic Shift Register回路 DSR7()を設計せよ。

Vin(t) がどの様に伝搬し、Vout(t)として出力されるかを説明せよ。

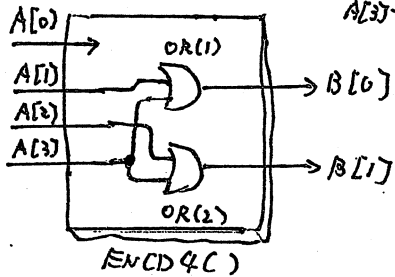
6.01 4bit Encoder 回路 ENCD4C) の設計

(P.81 練習問題 5-1)



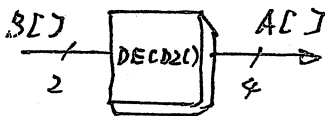
| A[3] | A[2] | A[1] | A[0] | B[1] | B[0] |
|------|------|------|------|------|------|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

$B[0]$ が 1 になるのは、 $A[1]=1$ 又は $A[3]=1$
 $B[0] = A[1] + A[3]$ となる。
 $B[1]$ が 1 になるのは、 $A[2]=1$ 又は $A[3]=1$
 $B[1] = A[2] + A[3]$
 $A[0]$ の入力信号は 無視してよい!!

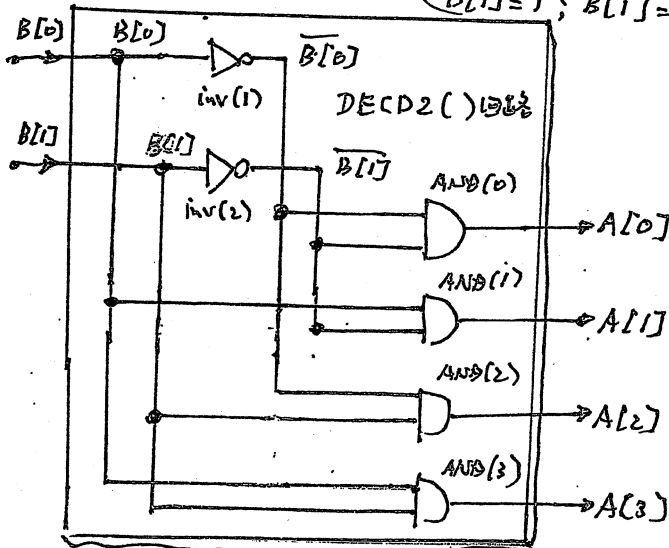


この場合、A[3] の信号端子は複数個 High
 になる可能性がある。ゆえに A[3] の信号は 4 本のうち
 1 本だけ High になるように 補強回路 が必要。

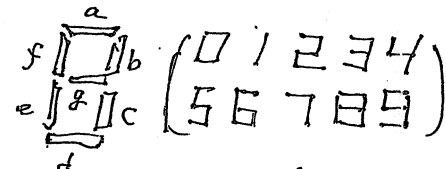
6.02 2 bit Decoder 回路 DECD2C) の設計 (P.81 練習問題 5-2)



$B[0]=B[1]=0$ の時のみ $A[0]=1$; $A[0] = \overline{B[0]} \cdot \overline{B[1]}$
 $B[0]=1; B[1]=0$ の時のみ $A[1]=1$; $A[1] = B[0] \cdot \overline{B[1]}$
 $B[0]=0; B[1]=1$ の時のみ $A[2]=1$; $A[2] = \overline{B[0]} \cdot B[1]$
 $B[0]=1; B[1]=1$ の時のみ $A[3]=1$; $A[3] = B[0] \cdot B[1]$



| 10進 | B[1] | B[0] | A[3] | A[2] | A[1] | A[0] |
|-----|------|------|------|------|------|------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 | 1 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 1 | 0 | 0 | 0 |



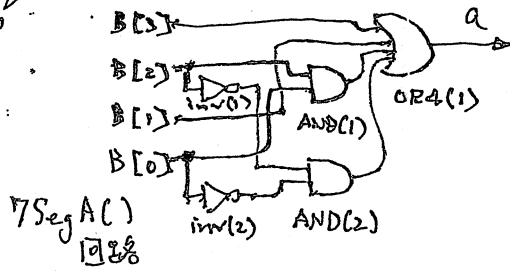
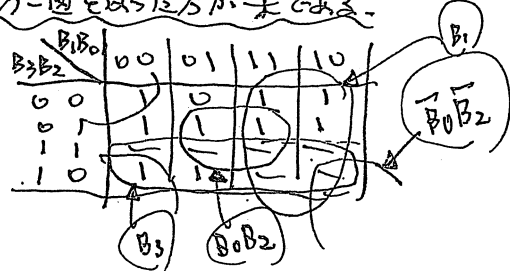
6.03 7 Segment LED 回路の設計.

$a=0$ の場合のみ $B[1]=0$ と $B[0]=0$ の場合のみ
 $(B[3]=0; B[2]=0; B[1] \neq B[0])$ の場合

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

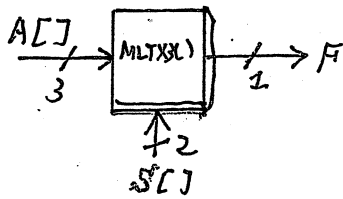
$a = \overline{B_3} \overline{B_2} \overline{B_1} \overline{B_0} + \overline{B_3} \overline{B_2} B_1 \overline{B_0} = \overline{B_3} \overline{B_2} (\overline{B_1} \overline{B_0} + B_1 \overline{B_0}) = \overline{B_3} \overline{B_2} (\overline{B_0}) + \overline{B_3} \overline{B_2} B_1$
 $a = B_3 + B_1 + (\overline{B_2} \overline{B_0}) \cdot (B_2 \overline{B_0}) = B_3 + B_1 + (B_2 + \overline{B_0}) \cdot (\overline{B_2} + B_0)$
 $a = B_3 + B_1 + \overline{B_0} \overline{B_2} + B_2 \overline{B_0}$ となる。

カク-網を使った方が楽である。



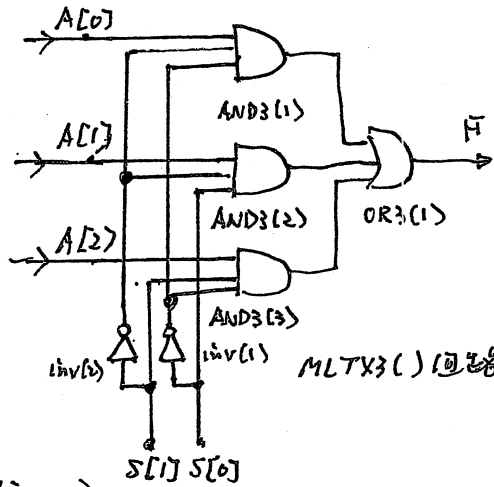
7Seg AC) 回路

6.04 3入力 Multiplexer 回路 MLTX3()



| S[1] | S[0] | F |
|------|------|------|
| 0 | 0 | A[0] |
| 0 | 1 | A[1] |
| 1 | 0 | A[2] |
| 1 | 1 | 0 |

$$F = \bar{S}_1 \bar{S}_0 A_0 + \bar{S}_1 S_0 A_1 + S_1 \bar{S}_0 A_2$$



6.05 8ビット一致判定回路 MatchB() の設計

A[] = B[] の時のみ、F=0 とする回路とす。

<EXORC() 回路>

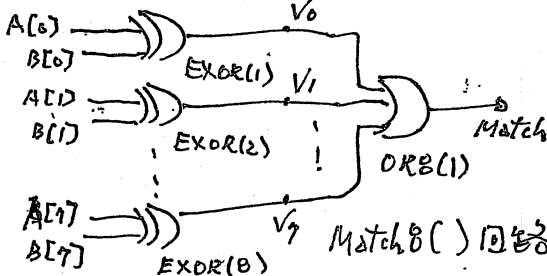
| A | B | EXOR |
|---|---|------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

EXOR() 回路は、

A=B の時のみ F=0 とす。

(OR() 回路は、7ビットの信号を 0 の時のみ、出力が 0 とおきか)

この EXOR() 回路を 8個使ひ、その出力 8本を ORB() 回路に入力すれば良い!! (DCOL code)



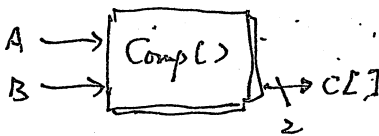
```

define MatchB() {
    input A[], B[];
    output Match;

    for k=0 to 7 {
        A[k] B[k] EXOR(k) -> V[k];
        V[k] ORB(1) -> Match;
    }
}
    
```

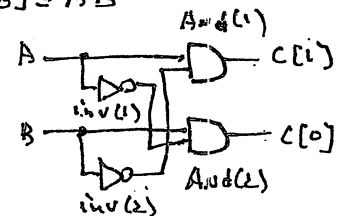
6.06 2ビット比較判定回路 Comp() の設計

$$C[1] = A\bar{B} \quad C[0] = \bar{A}B$$



| A | B | C[1] | C[0] |
|---|---|------|------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

A=B=0
A<B
A>B
A=B=1



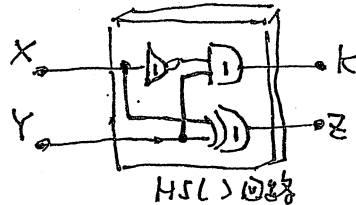
6.07 1けた借り型 1ビット減算回路 HS() の設計 (HS=half Subtractor)

Z = X - Y を計算し、X=0, Y=± の時 Z=1 とす。k=1 とす。

| X | Y | Z | k |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

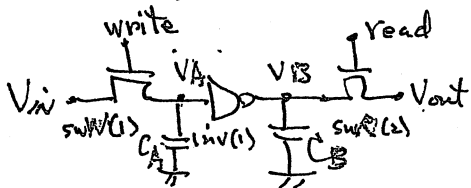
$$Z = \bar{X}Y + X\bar{Y}$$

$$k = \bar{X}Y$$



6.06 Dynamic Shift Register 回路 DSR(1) の設計

1 bit Register 回路とは 1 bit の (1 桁の) 情報を登録 (registering) する回路。
× 1 bit 回路のことではない!



(1 bit dynamic Register 回路)

Write 信号が High にすると、

V_{in} の値を記憶させる。

$V_A = V_{in}$ とする。

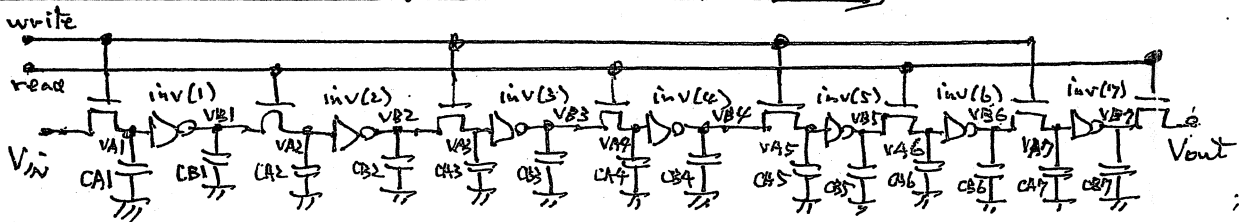
$V_B = \overline{V_{in}}$ とする。その値が

容量 C_A と C_B に保存される。(準備して!)

$Q = CV$
が 1 桁情報

読み出した時は Read 信号を high にする

7 段の Dynamic Shift Register 回路 DSR7(1) の設計



Step ① 今、read 信号が High にすると、 V_{B7} の値が V_{out} に出力される。

同時に、 $V_{out} = V_{B7}$; $V_{A6} = V_{B5}$; $V_{A4} = V_{B3}$; $V_{A2} = V_{B1}$; とする。

Step ② 次に write 信号が High にすると (この時は read 信号は Low!!)

V_{in} の値が V_{A1} に入る。 $V_{A1} = V_{in}$; $V_{A3} = V_{B2}$; $V_{A5} = V_{B4}$; $V_{A7} = V_{B6}$; とする。

書き込み信号 write (t) が $t_1, t_3, t_5, \dots, t_{2k+1}$ に High にするとする。
読み出し信号 read (t) が $t_2, t_4, t_6, \dots, t_{2k}$ に High にするとする (k=0, 1, 2, ...)

$V_{in}(t_{2k+1})$ の値が 1 桁 (shift) して、いよいよ、 $V_{out}(t)$ として出力される。

この 7 段の DSR7(1) 回路では、 $V_{A1}(t_{2k+1}) = V_{in}(t_{2k+1})$

$t = t_{2k+1}$ の時、write = 1 で、 $V_{B1}(t_{2k+1}) = \overline{V_{A1}(t_{2k+1})} = \overline{V_{in}(t_{2k+1})}$

$t = t_{2k+2}$ の時、read 信号が high にする。

$$\left(\begin{aligned} V_{A2}(t_{2k+2}) &= V_{B1}(t_{2k+1}) = \overline{V_{in}(t_{2k+1})} \text{ とする。} \\ V_{B2}(t_{2k+2}) &= \overline{V_{A2}(t_{2k+2})} = V_{in}(t_{2k+1}) \text{ とする。} \end{aligned} \right)$$

次に $t = t_{2k+3}$ の時、write 信号が high とする。

$$V_{A3}(t_{2k+3}) = V_{B2}(t_{2k+2}) = V_{in}(t_{2k+1})$$

$$V_{B3}(t_{2k+3}) = \overline{V_{A3}(t_{2k+3})} = \overline{V_{in}(t_{2k+1})} \text{ とする。}$$

1 桁 1 cycle ごと

(read = high にする) 毎に、write = high にすると $V_{A[2m+1]}$ の信号が $V_{A[2m+3]}$ に shift する。

(入力信号 data $V_{in}(t_{2k+1})$ が 1 桁、shift して行くことにする!!)

$V_{in}(t)$ の値が 1 cycle ごと、 V_{A1} から V_{A3} , V_{A3} から V_{A5} , V_{A5} から V_{A7} に移る。