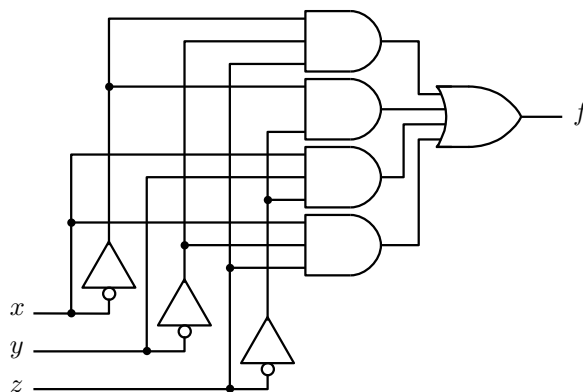


I 次の論理回路の真理値表を書きなさい。(10 点)



II 次の真理値表で定義される論理関数 $f(w, x, y, z)$ について考える。

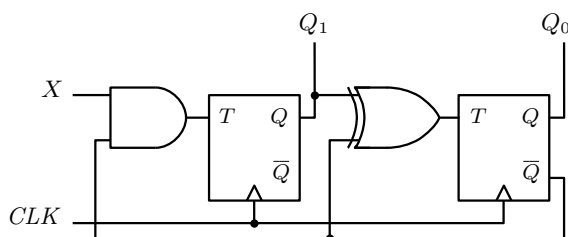
w	x	y	z	$f(w, x, y, z)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1

w	x	y	z	$f(w, x, y, z)$
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

(1) この論理関数のカルノー図を書きなさい。(10 点)

(2) (1) のカルノー図を利用して、この論理関数を表す簡単な論理式を求めなさい。(10 点)

III CLK に同期して動作する次の順序回路の状態遷移表と状態遷移図を完成しなさい。(20 点)



IV リセット可能な 3 bit カウンタの回路図を完成しなさい。非同期式カウンタとしても同期式カウンタとしてもよい。この回路は CLK に同期して次の状態遷移表に基づいた状態遷移を行う。(20 点)

現在の状態			入力	次の状態		
Q_2	Q_1	Q_0	\overline{RST}	Q_2	Q_1	Q_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	0	1	1
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	0	1	1	0	1
1	0	1	0	0	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	1	0	0	0

V 解答用紙の表は、RESET = 0、IN = 0101 と固定した場合、第 13 回配布資料の 4bit CPU の状態がどのように変化していくかをクロック信号の立ち上がりで区切られる時刻ごとに記したものである。この表の空欄を適当なビットパターンで埋めなさい。ただし、プログラムメモリ中に記憶された機械語プログラは右の通りとし、解答欄の表の、PC、A、B、CF、Out の欄は、各レジスタの Q の値を意味するものとする。(30 点)

番地	ビットパターン	命令
0	0001 1000	IN B
1	0110 1001	CPY A, 9
2	0100 0100	ADD A, B
3	1000 0111	JPC 7
4	0100 0101	SUB A, B
5	1000 1000	JPC 8
6	0000 0010	OUT A
7	0010 1000	SWP A, B
8	0000 0010	OUT A
9	1100 0111	JMP 7
10	0000 0000	NOP
⋮	⋮	⋮

計算機システム I ・ 定期試験 ・ 解答用紙

I

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0

x	y	z	f
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

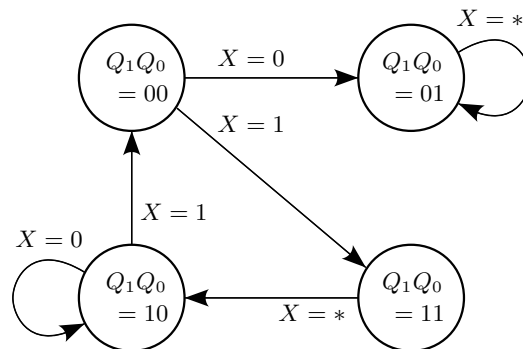
II (1)

$w x$	$y z$			
	00	01	11	10
00	0	0	1	0
01	1	0	1	0
11	1	0	0	1
10	1	0	0	1

(2) $f(w, x, y, z) = \bar{w} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z} + w \cdot \bar{z}$

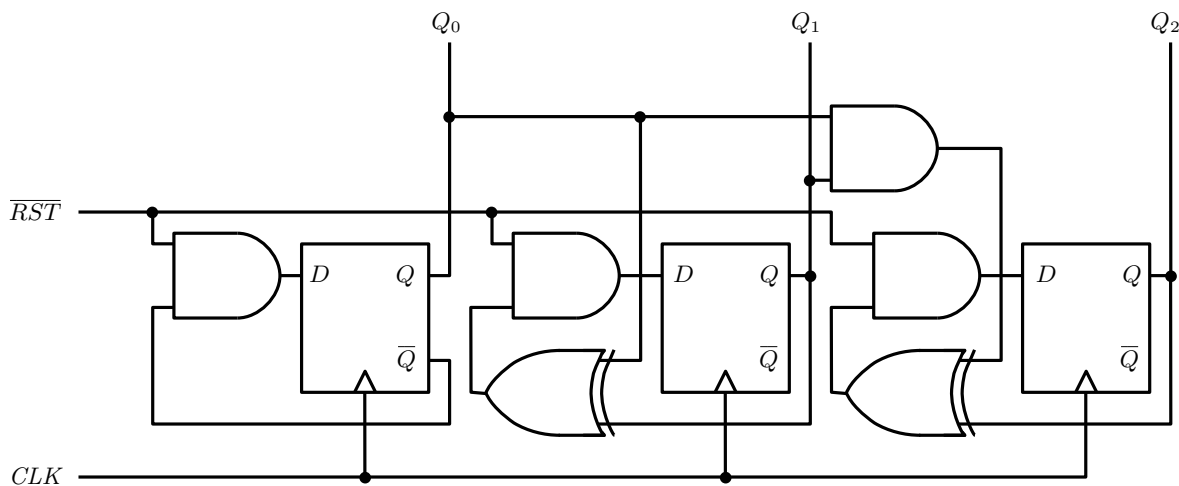
III

現在の状態		入力 X	次の状態	
Q_1	Q_0		Q_1	Q_0
0	0	0	0	1
0	0	1	1	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0



計算機システム I ・ 定期試験 ・ 解答用紙

IV



V

時刻	PC	LD^{PC}	S_1^A	S_0^A	S_1^B	S_0^B	S^C	S^O	\bar{A}/S	A	B	CF	Out
0	000000	0	0	0	1	1	0	0	0	0100	1010	1	1111
1	000001	0	1	1	0	0	0	0	1	0100	0101	1	1111
2	000010	0	1	0	0	0	1	0	0	1001	0101	1	1111
3	000011	0	0	0	0	0	0	0	1	1110	0101	0	1111
4	000100	0	1	0	0	0	1	0	1	1110	0101	0	1111
5	000101	1	0	0	0	0	0	0	0	1001	0101	1	1111
6	001000	0	0	0	0	0	0	1	0	1001	0101	1	1111
7	001001	1	0	0	0	0	0	0	1	1001	0101	1	1001
8	000111	0	0	1	0	1	0	0	0	1001	0101	1	1001
9	001000	0	0	0	0	0	0	1	0	0101	1001	1	1001
10	001001	1	0	0	0	0	0	0	1	0101	1001	1	0101