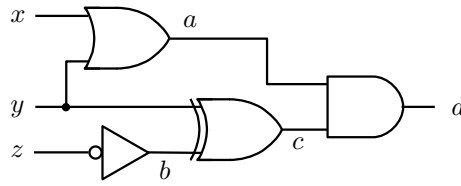


I 次の論理回路について、信号線 a, b, c, d の値が、入力 x, y, z の値に応じて、どのように変わるかを真理値表に書き込みさい。(10 点)



II 右の真理値表で定義される論理関数 $f(x, y, z)$ を考える。

- (1) この論理関数を表す主論理和標準形の論理式を書きなさい。(6 点)
- (2) この論理関数のカルノー図を書きなさい。(10 点)
- (3) (2) 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。(6 点)
- (4) この論理関数 $f(x, y, z)$ を計算する論理回路図をかきなさい。入力 x, y, z 、出力 $f(x, y, z)$ の信号線をそれぞれ回路図に明示すること。(10 点)

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

III 入力 X と出力 Q_1, Q_0 を持ち、クロック信号 CLK の立ち上がりで同期して右の状態遷移表のように状態遷移する順序回路を考える。

- (1) この状態遷移表に相当する状態遷移図を完成しなさい。(10 点)
- (2) 2つの D-フリップフロップと論理ゲートを組み合わせて、このような順序回路の回路図を書きなさい。入力 X 、出力 Q_1, Q_0 、クロック入力 CLK の信号線をそれぞれ回路図に明示すること。(20 点)

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

IV 解答用紙の表は、RESET = 0、IN = 1001 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ムは右の通りとし、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。(28 点)

番地	ビットパターン	命令
0	0110 0101	CPY A, 5
1	0000 1000	CPY B, A
2	0100 0100	ADD A, B
3	0001 1000	IN B
4	0100 0100	ADD A, B
5	1000 0111	JPC 7
6	0010 1000	SWAP A, B
7	0000 0010	OUT A
8	1100 0011	JMP 3
9	0000 0000	NOP
⋮	⋮	⋮

計算機システム I ・ 定期試験 ・ 解答用紙

I

x	y	z	a	b	c	d
0	0	0	0	1	1	0
0	0	1	0	0	0	0
0	1	0	1	1	0	0
0	1	1	1	0	1	1
1	0	0	1	1	1	1
1	0	1	1	0	0	0
1	1	0	1	1	0	0
1	1	1	1	0	1	1

II

(1)

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

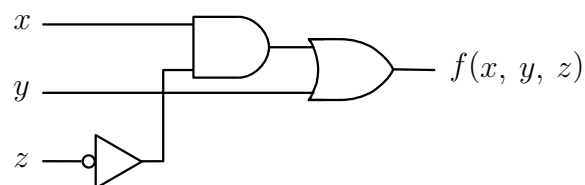
(2)

x	yz			
	00	01	11	10
0	0	0	1	1
1	1	0	1	1

(3)

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

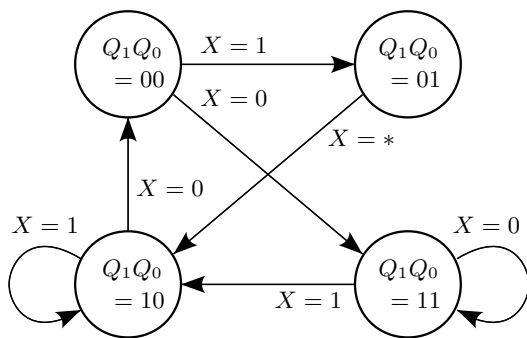
(4)



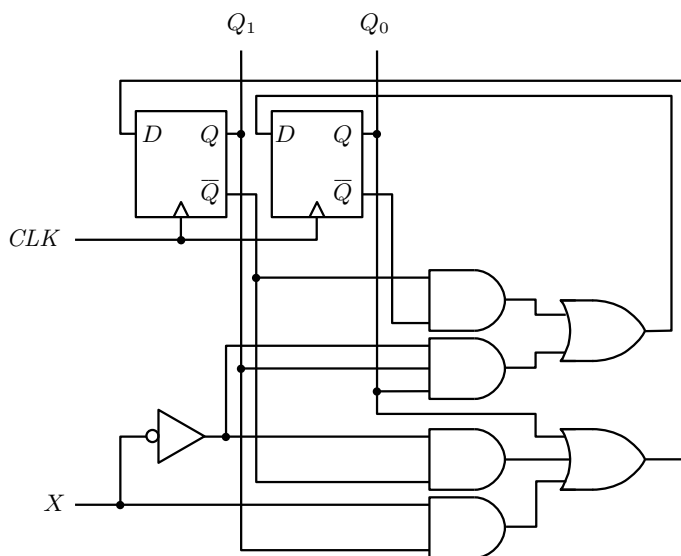
計算機システム I ・ 定期試験 ・ 解答用紙

III

(1)



(2)



IV

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	1	1	0	0	0	0	1	0001	0010	1	0000
1	000001	0	0	0	0	1	0	0	0	0101	0010	1	0000
2	000010	0	1	0	0	0	1	0	0	0101	0101	1	0000
3	000011	0	0	0	1	1	0	0	0	1010	0101	0	0000
4	000100	0	1	0	0	0	1	0	0	1010	1001	0	0000
5	000101	1	0	0	0	0	0	0	1	0011	1001	1	0000
6	000111	0	0	0	0	0	0	1	0	0011	1001	1	0000
7	001000	1	0	0	0	0	0	0	1	0011	1001	1	0011
8	000011	0	0	0	1	1	0	0	0	0011	1001	1	0011