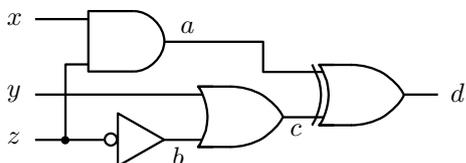


I 次の論理回路について、信号線  $a, b, c, d$  の値が、入力  $x, y, z$  の値に応じて、どのように変わるかを真理値表に書き込み下さい。(10 点)



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

II 右の真理値表で定義される論理関数  $f(x, y, z)$  を表す論理式を求め、その論理式に相当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。(15 点)

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

学籍番号

氏名

(裏面に続く)

この定期試験の採点結果とこの科目の総合成績は、来週末までに、Email で [txxxxxx@mail.ryukoku.ac.jp](mailto:txxxxxx@mail.ryukoku.ac.jp) 宛にお送りします。この連絡が不要な人は右の「連絡不要」を丸で囲んでください。

連絡不要

III 論理式  $F = \bar{z} \cdot (\bar{x} + y) + x \cdot (\bar{y} + \overline{(\bar{y} + z)})$  について次の問いに答えなさい。(25 点)

(1) 論理式  $F$  を変形し、論理和標準形になったところまで変形をやめなさい。

(2) (1) の結果をさらに変形し、主論理和標準形にしなさい。

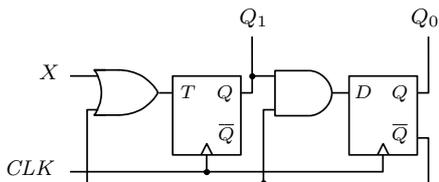
(3) 右の真理表が、論理式  $F$  のものとなるように空欄を埋めなさい。

$x$	$y$	$z$	$F$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

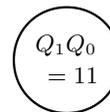
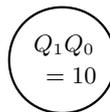
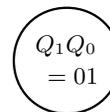
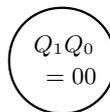
(4) 論理式  $F$  のカルノー図を書きなさい。

(5) (4) のカルノー図を利用して、 $F$  を簡単化した論理式をかきなさい。

IV  $CLK$  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)



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

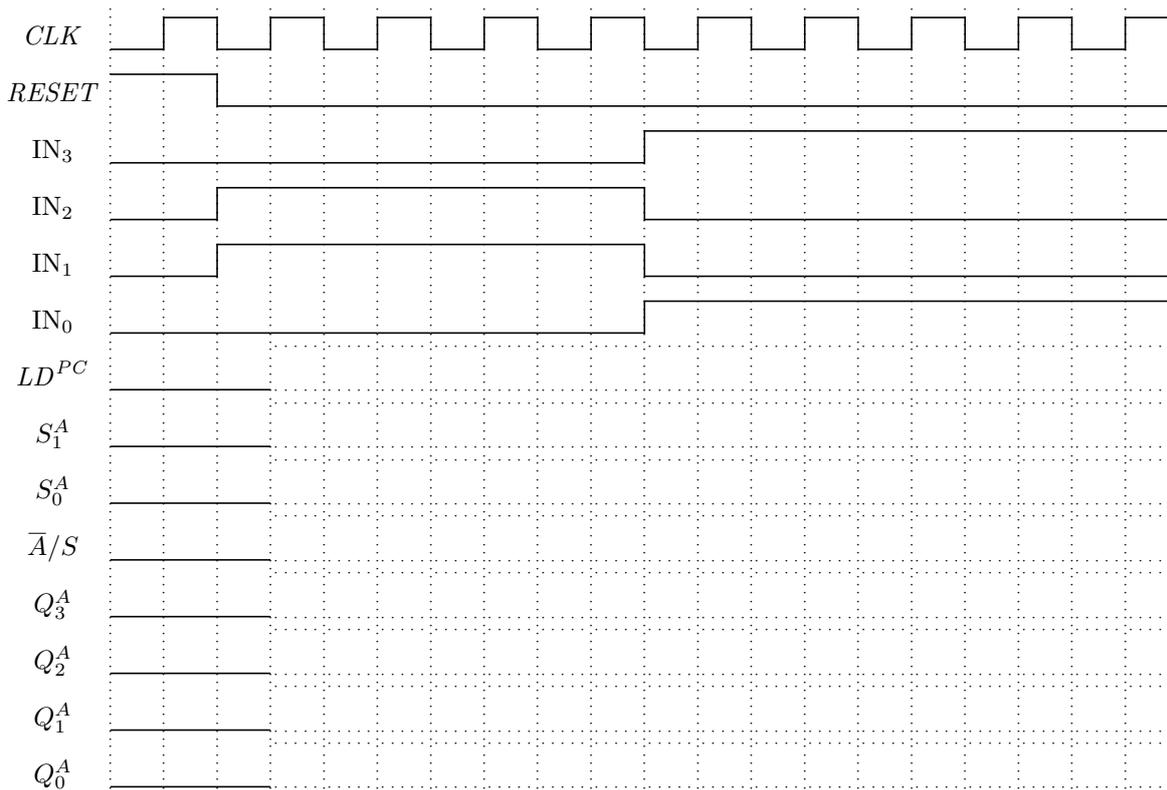


V 右の状態遷移表に基づいて状態遷移を行う順序論理回路を2つのD-フリップフロップと論理ゲートで実現する回路図を書きなさい。入力  $X$ 、クロック信号  $CLK$ 、出力  $Q_1$ 、 $Q_0$  の信号線をそれぞれ回路図に明示すること。(15 点)

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

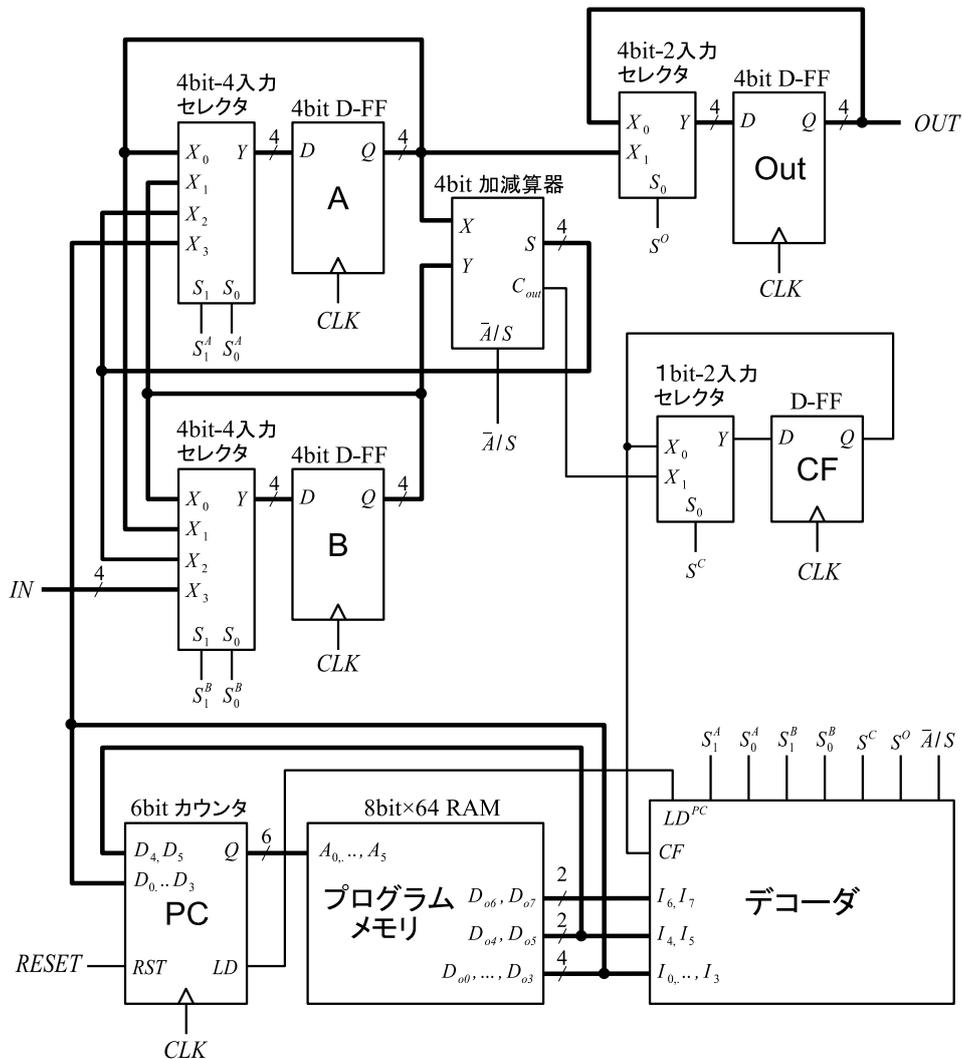
VI 下のタイミングチャートは、クロック信号の立ち上がり同期して、第13回配布資料の4bit CPUの状態がどのように変化していくかを記したものである。このタイミングチャートを完成しなさい。ただし、タイミングチャート中の $Q_3^A Q_2^A Q_1^A Q_0^A$ は、Aレジスタ(4bit D-フリップフロップ)の出力 $Q_3 Q_2 Q_1 Q_0$ とする。また、プログラムメモリ中に記憶された機械語プログラムは右の通りとする。(15点)

番地	ビットパターン	命令
0	0110 1101	CPY A, 13
1	0001 1000	IN B
2	0100 0100	ADD A, B
3	1000 0101	JPC 5
4	0010 1000	SWP A, B
5	0000 0010	OUT A
6	1100 0100	JMP 4
7	0000 0000	NOP
8	0000 0000	NOP
⋮	⋮	⋮

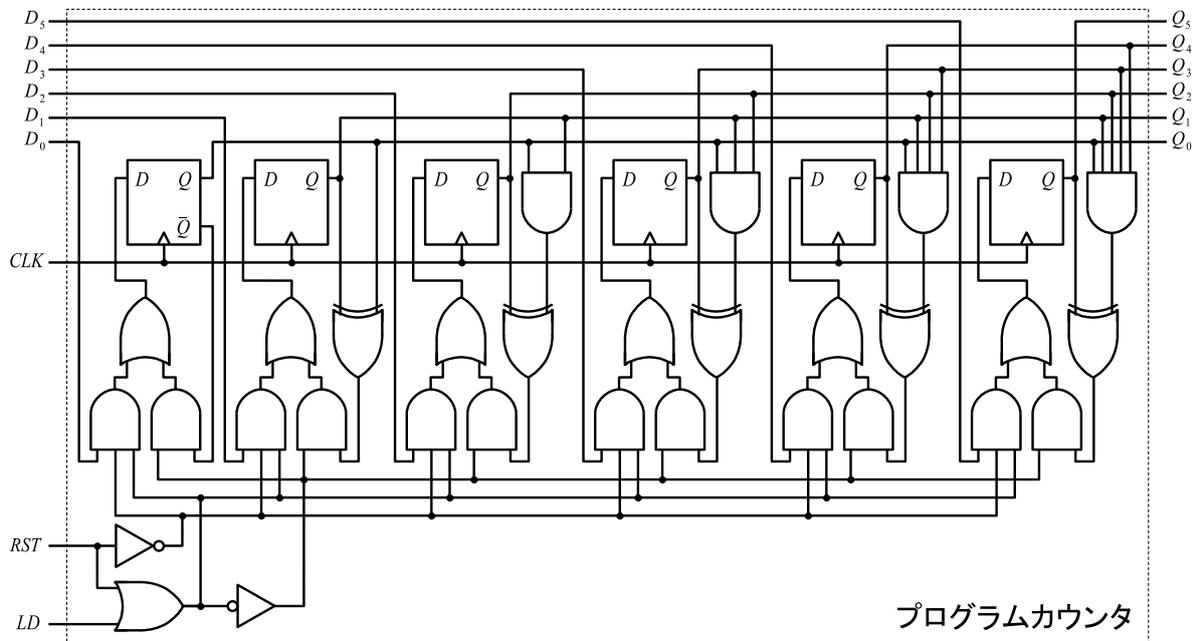


4 bit CPU の機械語命令

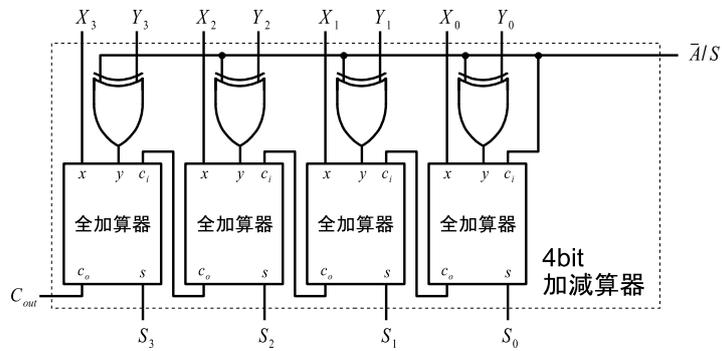
機械語命令のビットパターン								命令	動作
$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$		
0	0	0	0	0	0	0	0	NOP	何もしない
0	0	0	0	0	0	1	0	OUT A	A レジスタの値を OUT へ出力
0	0	0	0	0	1	0	1	CMP A, B	A-B の計算結果を CF へ反映
0	0	0	0	1	0	0	0	CPY B, A	A レジスタ の値を B へコピー
0	0	0	1	0	1	0	0	ADD B, A	A+B の計算結果を B へ書き込み
0	0	0	1	0	1	0	1	XSUB B, A	A-B の計算結果を B へ書き込み
0	0	0	1	1	0	0	0	IN B	IN の値を B レジスタへ書き込み
0	0	1	0	0	0	0	0	CPY A, B	B レジスタの値を A へコピー
0	0	1	0	1	0	0	0	SWP A, B	A レジスタ と B レジスタの値を交換
0	1	0	0	0	1	0	0	ADD A, B	A+B の計算結果を A へ書き込み
0	1	0	0	0	1	0	1	SUB A, B	A-B の計算結果を A へ書き込み
0	1	1	0	data				CPY A, data	定数 data を A レジスタへ書き込み
1	0	address				JPC address	CF=1 ならば address 番地にジャンプ		
1	1	address				JMP address	常に address 番地にジャンプ		



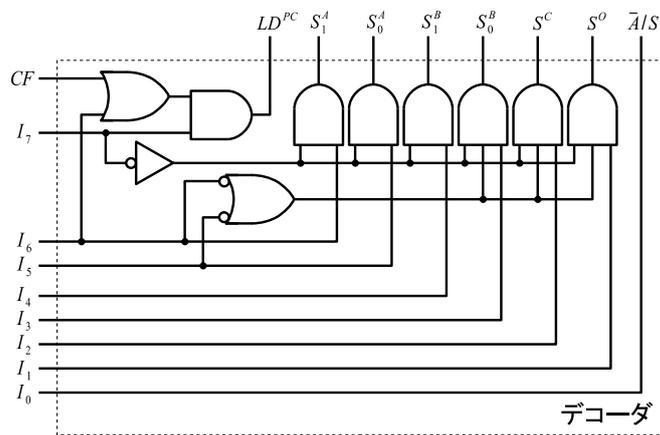
4 bit CPU の回路図



PC (プログラムカウンタ) の回路図



4 bit 加減算器の回路図



デコーダの回路図