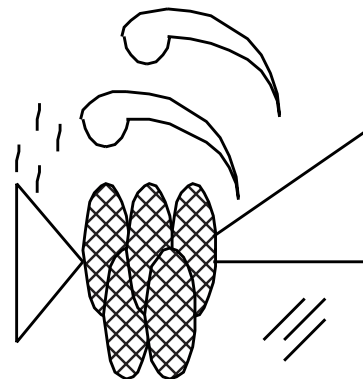


Digital Logic Design

Final Examination

Problems	Points
1. _____	4
2. _____	9
3. _____	10
4. _____	12
Total _____	35



Was the exam fair ?

yes

☐

no

☐

Problem 1 4 points

Table 1 contains three values of a function $x(i)$ in different radix (base) representations.

- 1.1 For $i=1,2,3$, express $x(i)$ in decimal (base ten) representation, and express both, $x(i)$ and $-x(i)$ in the **eight-bit base-two** representation which uses the **two's complement** notation for negative numbers. Show your computation on the opposite page, and enter the results into Table 1.

3

Table 1: Number representations

function $x(i)$		$x(i)$'s decimal representation	eight-bit base-two representation of	
i	$x(i)$		$x(i)$	$-x(i)$
1	-30_{10}	-30_{10}	11100010_2	00011110_2
2	134_8	92_{10}	01011100_2	10100100_2
3	$-3D_{16}$	-61_{10}	11000011_2	00111101_2

Hint: Direct conversion from octal and hexadecimal to binary representation is easier, and should be applied. Students are advised to avoid an indirect, e.g. octal $\rightarrow$ decimal $\rightarrow$ binary conversion. No partial credit will be given for a correct conversion from an erroneous decimal representation.

- 1.2 Using the **eight-bit base-two** representation and the **two's complement** notation for negative numbers, show the calculation of the difference $x(2)-x(1)$ when only an adder circuit is available.

1

Hint: This part of the problem is considered independent of the results of calculation performed in part 1.1. No partial credit will be earned for a correct calculation procedure applied to erroneous binary representations of $x(2)$ and $-x(1)$ - whether taken from Table 1, or calculated independently.

$$\begin{array}{r}
 01011100 \\
 +00011110 \\
 \hline
 01111010_2 = 122_{10}
 \end{array}$$

Result of calculations under 1.2 to be graded: 01111010_2

Problem 2 9 points

Given is a logical function F_1 in the decimal sum-of-minterms representation (2-1) and (2-2).

$$F_1(A,B,C,D) = \Sigma(1, 3, 5, 7, 9, 15) \quad (2-1)$$

$$d(A,B,C,D) = \Sigma(4, 6, 12, 13) \quad (2-2)$$

- 1 2.1 Prepare the Karnaugh map representation of the function F_1 in the space provided for it in Figure 2-1(a)

:

		CD			
		00	01	11	10
AB	00		1	1	
	01	d	1	1	d
	11	d	d	1	
	10		1		

(a)

$$F_1 = \bar{A} \bar{D} + B \bar{D} + \bar{C} \bar{D}$$

(b)

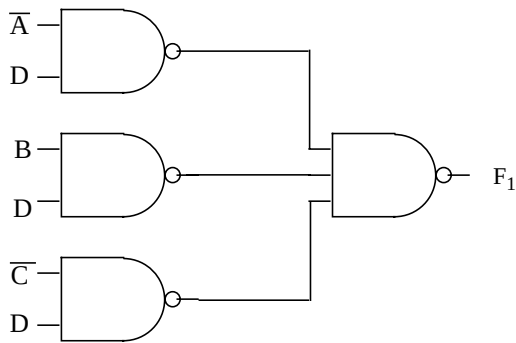
$$\bar{F}_1 = (A + \bar{D})(\bar{B} + \bar{D})(C + \bar{D})$$

(c)

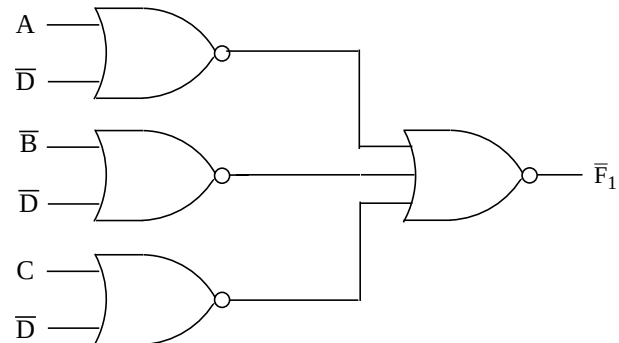
Figure 2-1 Representation forms of the function F_1 . (a)Karnaugh map. (b)Minimum number of literals SOP representation of F_1 . (c)Minimum number of literals POS representation of $\bar{F}_1$.

- 2 2.2 Apply the Karnaugh map minimization method to derive the Minimum number of literals sum-of-products (SOP) representation of the function F_1 . Enter the derived algebraic representation in the space reserved for Figure 2-1(b).
- 2 2.3 Apply the Karnaugh map minimization method to derive the Minimum number of literals product-of-sums (POS) representation of the function $\bar{F}_1$. Enter the derived algebraic representation in the space reserved for Figure 2-1(c).
- 2 2.4 In the space reserved for Figure 2-2(a), prepare a logic circuit diagram of the two-level NAND-NAND form of implementation of the derived minimum number of literals SOP expression of the function F_1 .

- 2.4 In the space reserved for Figure 2-2(b), prepare a logic circuit diagram of the two-level NOR-NOR form of implementation of the derived minimum number of literals POS expression of the function $\bar{F}_1$.



(a)



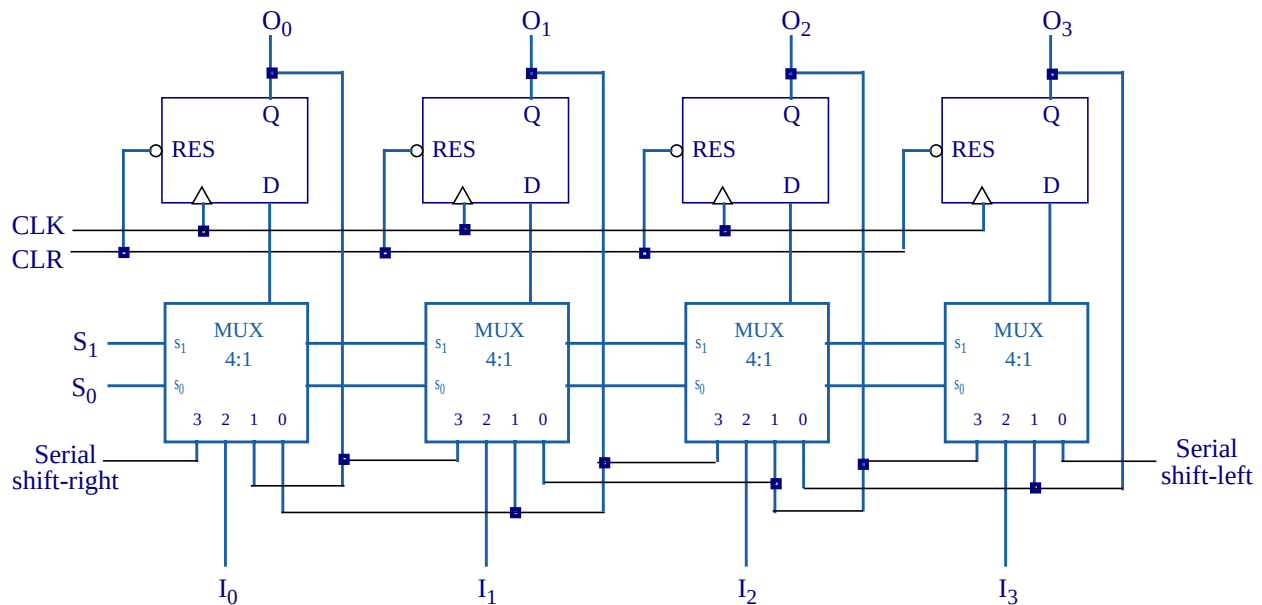
(b)

Figure 2-2 Two-level implementation of the minimum number of literals expressions of the functions F_1 and $\bar{F}_1$. (a) NAND-NAND implementation of F_1 . (b) NOR-NOR implementation of $\bar{F}_1$.

Problem 3 10 points

A partial (incomplete) logic circuit of a universal (bidirectional) shift-register is shown in Figure 3.1(a).

- 8 3.1 Design and enter in Figure 3.1(a) the missing connections such that the completed circuit of Figure 3.1(a) implements the universal shift register specified by the function table shown in Figure 3.1(b).



(a)

Operation code		Register operation
S_1	S_0	
0	0	shift left
0	1	no change
1	0	parallel load
1	1	shift right

(b)

Signal designation	Signal description
O_0	Serial output for shift left
O_3	Serial output for shift right

(c)

Figure 3.1 MUX-based implementation of a Universal shift register. (a)The logic circuit of the universal shift register. (b)Function table of the shift register, showing the operation codes to be implemented by the design. (c)Space for writing in the answer to part 3.2.

- 2 3.2 In the space reserved for Figure 3-1(c), write the (already present) signal designations from Figure 3.1(a) which represent the serial outputs for the shift left and shift right operations of the register.

Problem 4 12 points

Design a *state machine (SM)* which satisfies the following specification,

- the ASM chart of the SM is shown in Figure 4.1(b),
- the SM has two signal inputs A and B,
- the internal state memory of the machine is to be implemented using positive edge triggered D-type flip flops.

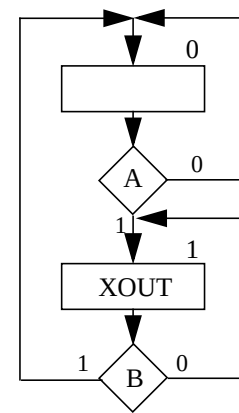
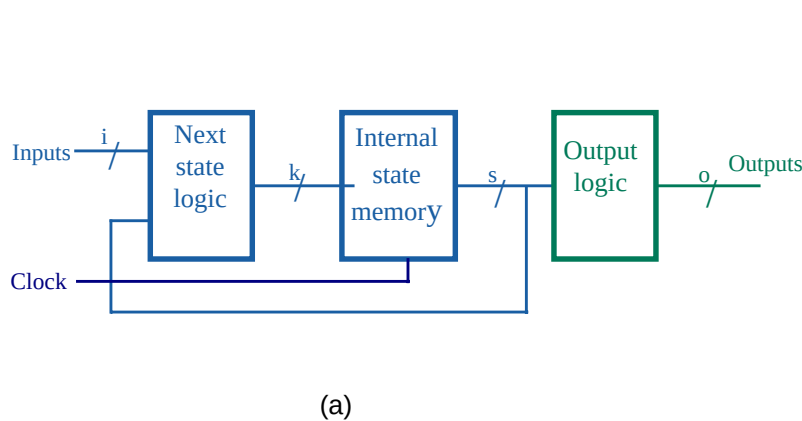


Figure 4.1 Moore-type State Machine. (a)General architecture. (b)ASM chart of a specific SM to be designed.

Show the following steps of the design procedure.

- 2 4.1 In the space reserved in Figure 4.2(a) compose the *state transition table* of the SM using the information from the state diagram.
- 2 4.2 In the space reserved for Figure 4.2(b) fill in the contents of the *D-type flip-flop excitation table*.
- 2 4.3 In the space reserved for Figure 4.2(c) compose the *transition excitation table* of the SM combining the information from the state transition table and the *flip-flop excitation table*.
- 2 4.4 In the space reserved for Figure 4.2(d) fill in the Karnaugh map of the flip-flop input function D whose truth table can be extracted from the *transition excitation table*.
- 2 4.5 Using the Karnaugh map, derive the simplified expression of the flip-flop input function D, and enter it in the space reserved for Figure 4.2(e).

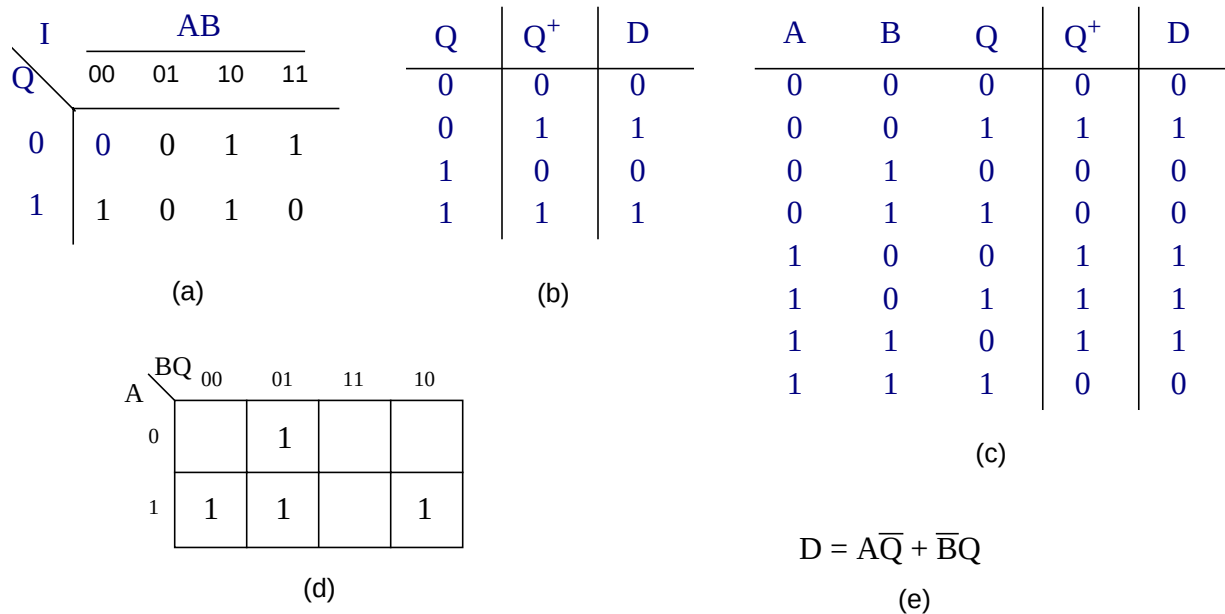


Figure 4.2 State Machine design process. (a) State transition table of the SM. (b) D-type flip-flop excitation table. (c) Transition excitation table of the SM. (d) Karnaugh map of the function D. (e) Simplified logical function D.

4.6 In the space reserved for Figure 4.3 prepare the logic circuit which implements the State Machine whose ASM chart is shown in Figure 4.1(b).

2

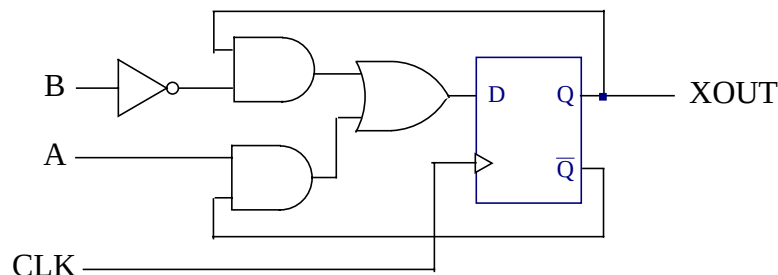


Figure 4.3 Logic circuit of the State Machine whose ASM chart is shown in Figure 4.1(b).

One can observe that with $A=J$ and $B=K$, the sequential circuit in Figure 4.3 implements a JK-type flip-flop. After that observation, one can conclude that the sequential circuit in Figure 4.3 shows how a D-type flip-flop can be converted into a JK-type flip-flop by properly gating the J and K input signals to the input D. The next step is to realize that the same technique can be applied to convert a D-type flip-flop into SR or T-type flip-flops. The final step is to realize that any type of flip-flop can be converted to any of the other three types using the same basic technique for designing SMs.