

Dr. Nicola Nicolici **COE/EE2DI4 Midterm Test #2** *Nov 21, 2005*

Instructions: This examination paper includes 12 pages and 20 multiple-choice questions starting on page 3. You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of your invigilator. The answers for all the questions must be indicated by filling the corresponding circle on the optical scanning (OMR) examination sheet. **This OMR examination sheet is the only page to be handed in.** The instructions for completing the OMR examination sheet are provided on page 2. Read and follow these instructions with care! There is one mark for each question. Answer all questions. There is no penalty for guessing. This is a closed book exam. No reference material of any kind is permitted. No calculators of any kind are permitted. **Time allowed is 50 minutes.**

Note: A' and \bar{A} are used interchangeably.

OMR examination instructions

Multiple choice questions (numbered 1 to 20) – indicate your answer by filling the corresponding circle on the OMR answer sheet

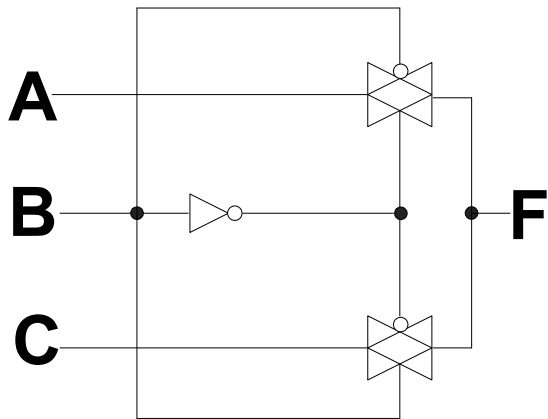


Figure 1 - CMOS circuit for question 1.

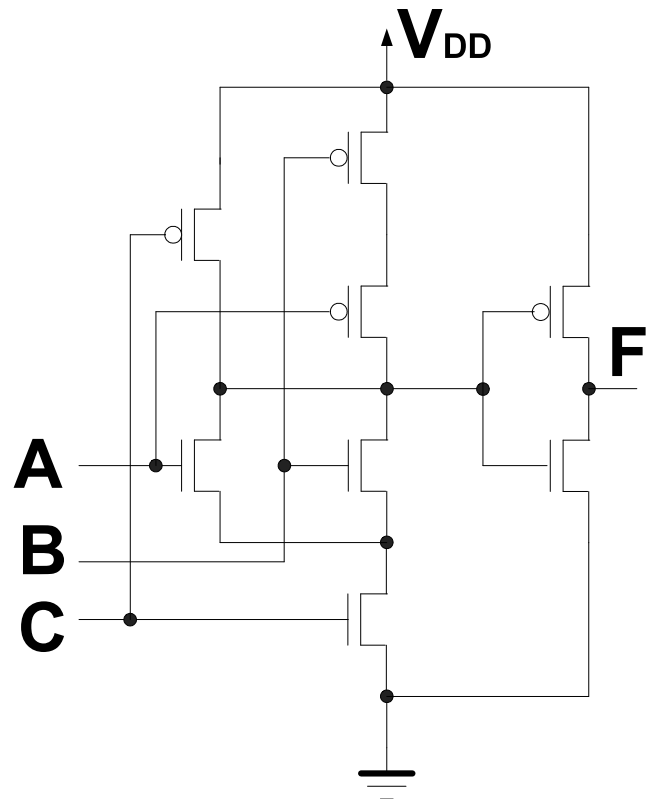


Figure 2 - CMOS circuit for question 2.

1. The logic function $F(A,B,C)$ implemented by the complementary metal-oxide semiconductor (CMOS) circuit shown in Figure 1:

1. $AB' + B'C$
2. $AB' + BC$
3. $AB + B'C$
4. $AB + BC$
5. none of the above

2. The logic function $F(A,B,C)$ implemented by the CMOS circuit shown in Figure 2 is:

1. $AB + C$
2. $AC + BC$
3. $A'B' + C'$
4. $A' + B'C'$
5. none of the above

3. The function implemented using three 2-input LUTs as shown in Figure 3 is:

1. $F(A,B,C) = \sum m(0,2,3,4,6,7)$
2. $F(A,B,C) = \sum m(0,2,4,6,7)$
3. $F(A,B,C) = \sum m(0,2,6,7)$
4. $F(A,B,C) = \sum m(0,2,7)$
5. none of the above

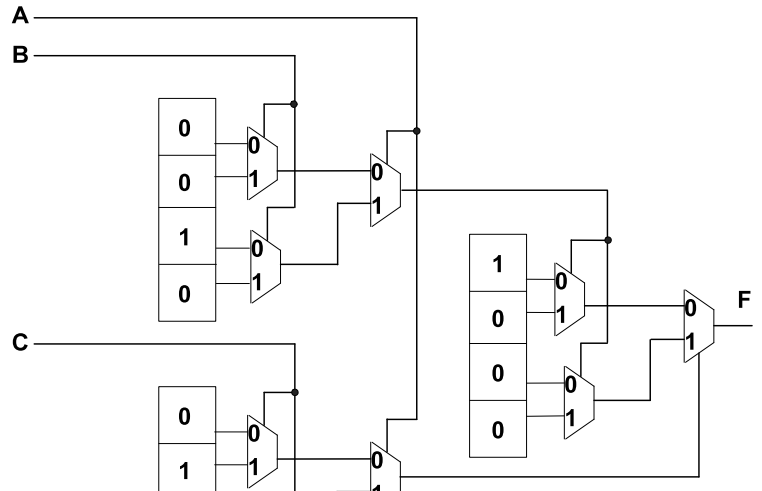
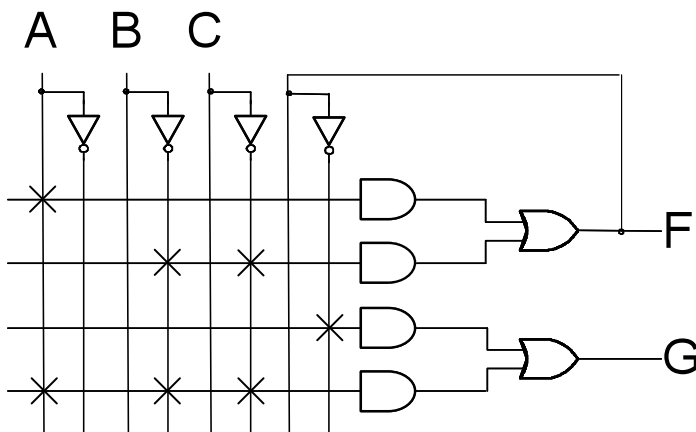


Figure 3 – Circuit for question 3.

4. The programmable array logic (PAL) implementation of function $G(A,B,C)$ shown in Figure 4 is:

1. $G(A,B,C) = \sum m(1,2,3,4,5,6)$
2. $G(A,B,C) = \sum m(1,2,3,4,5)$
3. $G(A,B,C) = \sum m(1,2,3,4)$
4. $G(A,B,C) = \sum m(1,2,3)$
5. none of the above

Figure 4 – Circuit for question 4. An X indicates that the wires are connected, i.e., the vertical signal is an input to the corresponding horizontal AND gate.



5. Consider a read-only memory (ROM) with 6 address (input) lines and 12 data (output) lines. The total capacity of this ROM is:

1. 32 bytes
2. 64 bytes
3. 72 bytes
4. 96 bytes
5. 128 bytes

6. Consider that you have available in your component library a ROM with 9 address lines and 8 data lines. We can label it as the *ROM_9_8* unit. Let's assume that in your application you need a ROM with 10 address lines and 16 data lines and we will label it as the *ROM_10_16* unit. Assuming that you have sufficient logic gates at your disposal for putting together several smaller ROMs into a larger one, what is the minimum number of *ROM_9_8* units that need to be used for implementing a *ROM_10_16* unit?

1. 16
2. 12
3. 8
4. 4
5. 2

7.

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY question7 IS
    PORT(w : IN STD_LOGIC_VECTOR(0 TO 3);
         y : OUT STD_LOGIC_VECTOR(0 TO 2));
END question7;

ARCHITECTURE Behavior OF question7 IS
BEGIN
    PROCESS (w)
    BEGIN
        IF w(3) = '1' THEN y <= "001";
        ELSIF w(2) = '1' THEN y <= "011";
        ELSIF w(1) = '1' THEN y <= "111";
        ELSE y <= "000";
        END IF;
    END PROCESS;
END Behavior;

```

In the above VHDL description, the assignment to the output signal $y(2)$ is equivalent to:

1. $y(2) \leq (\text{not } w(3)) \text{ and } (\text{not } w(2)) \text{ and } (\text{not } w(1));$
2. $y(2) \leq (\text{not } w(3)) \text{ or } (\text{not } w(2)) \text{ or } (\text{not } w(1));$
3. $y(2) \leq w(3) \text{ and } w(2) \text{ and } w(1);$
4. $y(2) \leq w(3) \text{ or } w(2) \text{ or } w(1);$
5. none of the above

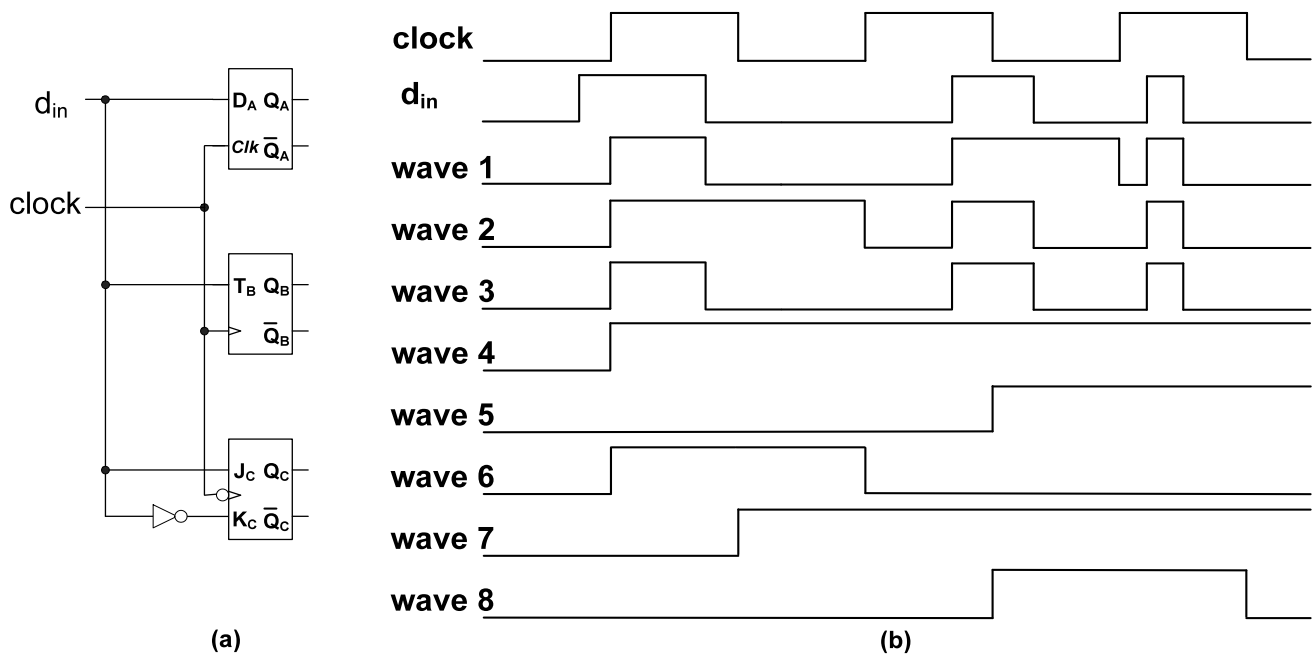


Figure 5 - Sequential (or storage) elements and signal waveforms for questions 8, 9 and 10. Note, the initial value of all the storage elements is zero and they are assumed to be ideal, i.e., the propagation delays and setup/hold times are considered to be zero.

8. Given the **clock** and **d_{in}** waveforms in Figure 5(b) the waveform for **Q_A** from Figure 5(a) is:

1. wave 1 from Figure 5(b)
2. wave 2 from Figure 5(b)
3. wave 3 from Figure 5(b)
4. wave 4 from Figure 5(b)
5. wave 5 from Figure 5(b)

9. Given the **clock** and **d_{in}** waveforms in Figure 5(b) the waveform for **Q_B** from Figure 5(a) is:

1. wave 4 from Figure 5(b)
2. wave 5 from Figure 5(b)
3. wave 6 from Figure 5(b)
4. wave 7 from Figure 5(b)
5. wave 8 from Figure 5(b)

10. Given the **clock** and **d_{in}** waveforms in Figure 5(b) the waveform for **Q_C** from Figure 5(a) is:

1. wave 4 from Figure 5(b)
2. wave 5 from Figure 5(b)
3. wave 6 from Figure 5(b)
4. wave 7 from Figure 5(b)
5. wave 8 from Figure 5(b)

11. In Figure 6, if the current state is $Q_3Q_2Q_1Q_0=1111$, then after the next *positive edge* of the *clock* signal the new state will be:

1. 1111
2. 0000
3. 0001
4. 0010
5. 0100

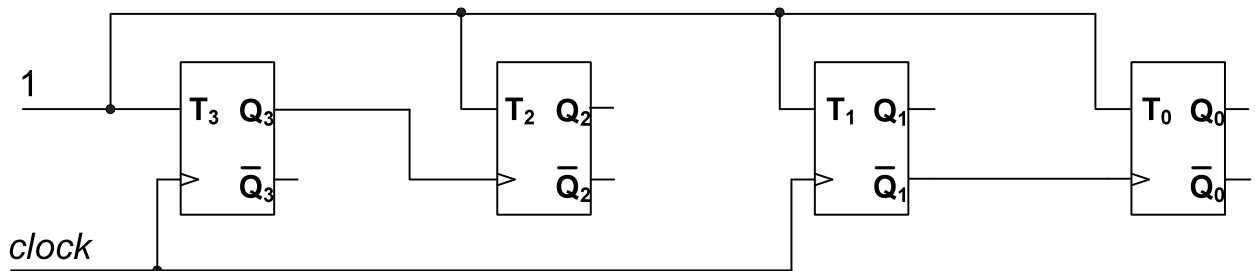


Figure 6 - Circuit for question 11 (note, the flip-flops are *positive edge* triggered).

Note for question 11: Because NOT all the flip-flops' clock inputs share the same *clock* signal, it is considered that the new state value will be observed only when all the flip-flops' outputs have become stable, i.e., after *all* the events triggered by the *positive edge* of the *clock* signal have been completed.

12. The periodic sequence $Q_2Q_1Q_0$ generated by the circuit shown in Figure 7 is:

1. 000, 100, 110, 011, 111
2. 000, 100, 110, 111
3. 000, 110, 011
4. 000, 111
5. none of the above

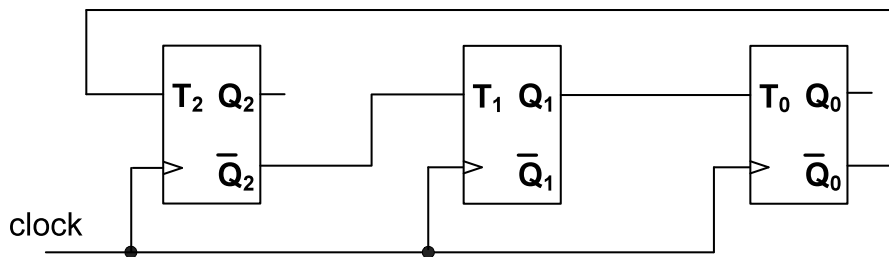


Figure 7 - Circuit for question 12 (it is assumed that after power up the state is 000).

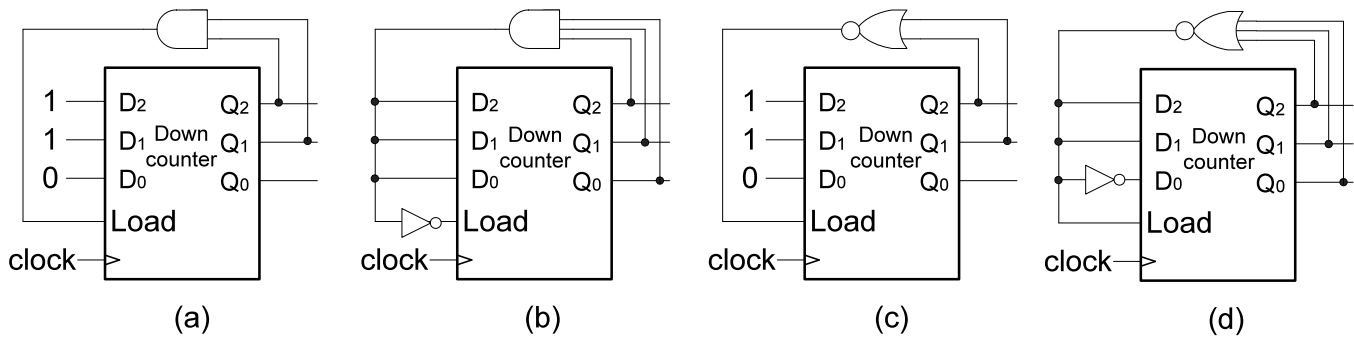


Figure 8 - Circuits for question 13 (it is assumed that after power up the state is 110).

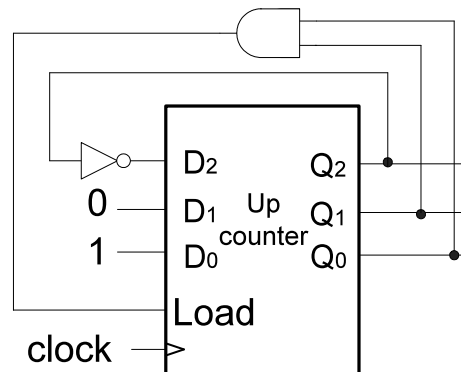
13. Using a synchronous 3-bit *down counter* with parallel load capability (Q_2 and D_2 are the most significant bits) we can generate a counting sequence 6, 5, 4, 3, 2, 1, 0, as shown in:

1. Figure 8(a)
2. Figure 8(b)
3. Figure 8(c)
4. Figure 8(d)
5. none of the above

14. Using a synchronous 3-bit *up counter* with parallel load capability (Q_2 and D_2 are the most significant bits) the circuit shown in Figure 9 generates the following periodic sequence:

1. 1, 2, 3, 4, 5, 6, 7
2. 1, 2, 3, 5, 6, 7
3. 1, 2, 3, 6, 7
4. 1, 2, 3, 7
5. 1, 2, 3

Figure 9 – Circuit for question 14 (it is assumed that after power up the state is 001).



15. Which of the circuits shown in Figure 10 (see next page) are equivalent?

1. Figure 10(a) and Figure 10(c)
2. Figure 10(a) and Figure 10(d)
3. Figure 10(b) and Figure 10(c)
4. Figure 10(b) and Figure 10(d)
5. none of the above

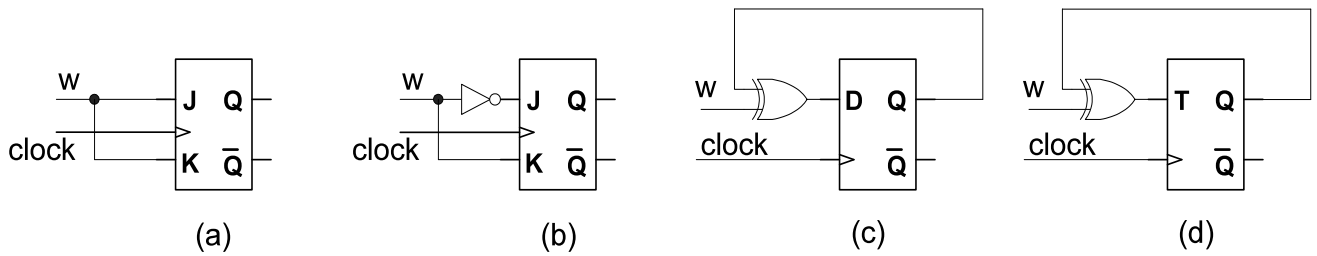
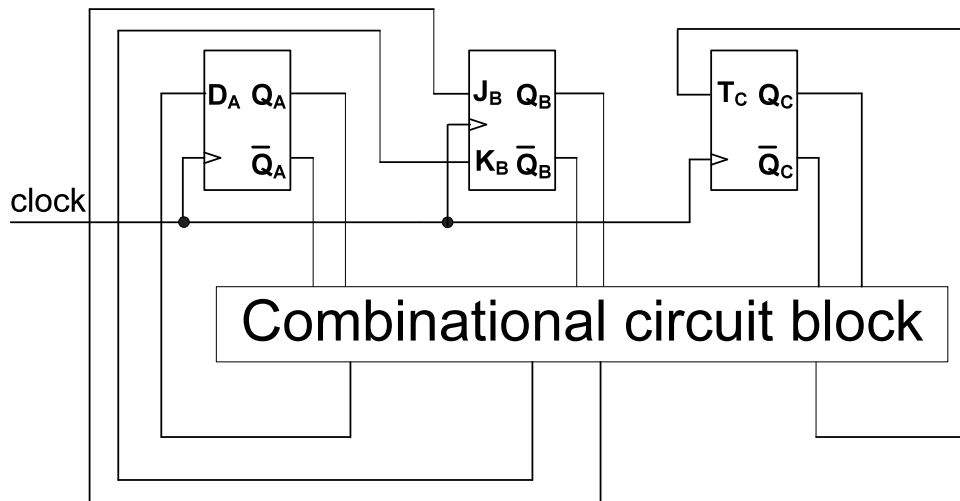


Figure 10 - Circuits for question 15.

Present State	Next State	Use this area for the truth tables of the flip-flops' inputs
000	010	
001	000	
010	011	
011	111	
100	101	
101	001	
110	100	
111	110	

(a)



(b)

Figure 11 - State table and circuit for questions 16, 17 and 18. Note, Q_A is the most significant bit. Tip: fill in the truth tables and use K-maps for logic minimization.

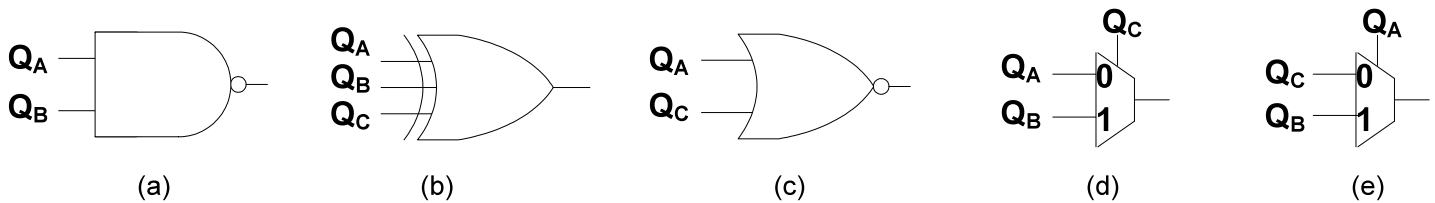


Figure 12 - Circuit implementations for questions 16, 17 and 18.

16. Consider the state transition table of a sequential circuit shown in Figure 11(a) (see previous page). If in our technology library we have only one **D** flip-flop, one **JK** flip-flop and one **T** flip-flop, then a generic implementation is shown Figure 11(b) (see previous page). Based on the state transition table and circuit topology shown in Figure 11, the logic function, which drives the *input D_A of the leftmost flip-flop from Figure 11(b)* is shown in:

1. Figure 12(a)
2. Figure 12(b)
3. Figure 12(c)
4. Figure 12(d)
5. Figure 12(e)

17. Consider the state transition table of a sequential circuit shown in Figure 11(a) (see previous page). If in our technology library we have only one **D** flip-flop, one **JK** flip-flop and one **T** flip-flop, then a generic implementation is shown Figure 11(b) (see previous page). Based on the state transition table and circuit topology shown in Figure 11, the logic function, which drives the *input J_B of the center flip-flop from Figure 11(b)* is shown in:

1. Figure 12(a)
2. Figure 12(b)
3. Figure 12(c)
4. Figure 12(d)
5. Figure 12(e)

18. Consider the state transition table of a sequential circuit shown in Figure 11(a) (see previous page). If in our technology library we have only one **D** flip-flop, one **JK** flip-flop and one **T** flip-flop, then a generic implementation is shown Figure 11(b) (see previous page). Based on the state transition table and circuit topology shown in Figure 11, the logic function, which drives the *input T_C of the rightmost flip-flop from Figure 11(b)* is shown in:

1. Figure 12(a)
2. Figure 12(b)
3. Figure 12(c)
4. Figure 12(d)
5. Figure 12(e)

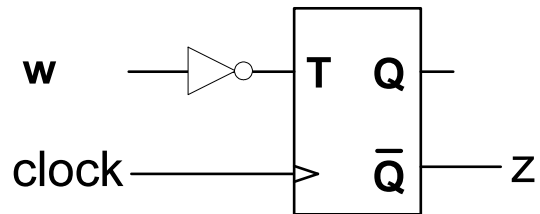


Figure 13 - Circuit for question 19.

19. Consider the sequential circuit shown in Figure 13. Consider the following state assignment: **A** stands for $Q=0$, **B** stands for $Q=1$. The state transition diagram for the circuit from Figure 13 is shown in:

1. Figure 14(a)
2. Figure 14(b)
3. Figure 14(c)
4. Figure 14(d)
5. none of the above

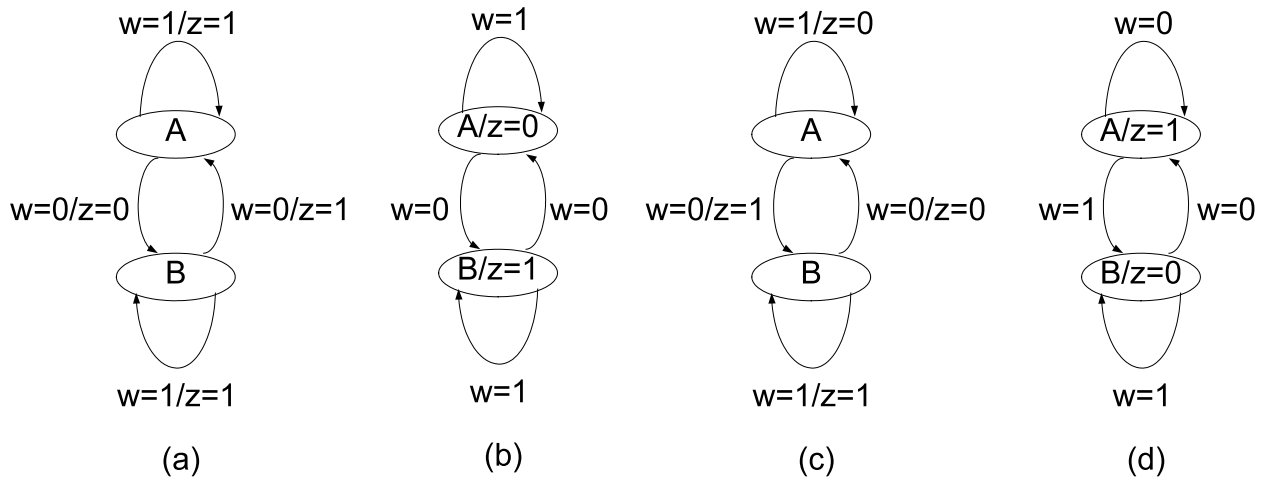


Figure 14 - State transition diagrams for question 19.

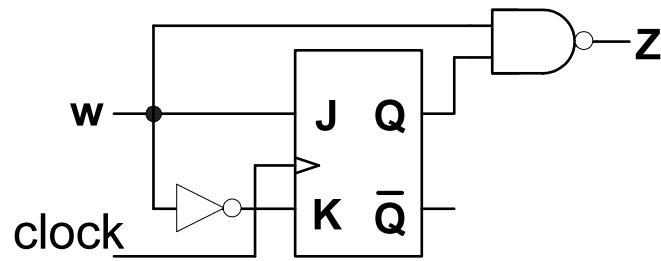


Figure 15 - Circuit for question 20.

20. Consider the sequential circuit shown in Figure 15. Consider the following state assignment: **A** stands for $Q=0$, **B** stands for $Q=1$. The state transition diagram for the circuit from Figure 15 is shown in:

1. Figure 16(a)
2. Figure 16(b)
3. Figure 16(c)
4. Figure 16(d)
5. none of the above

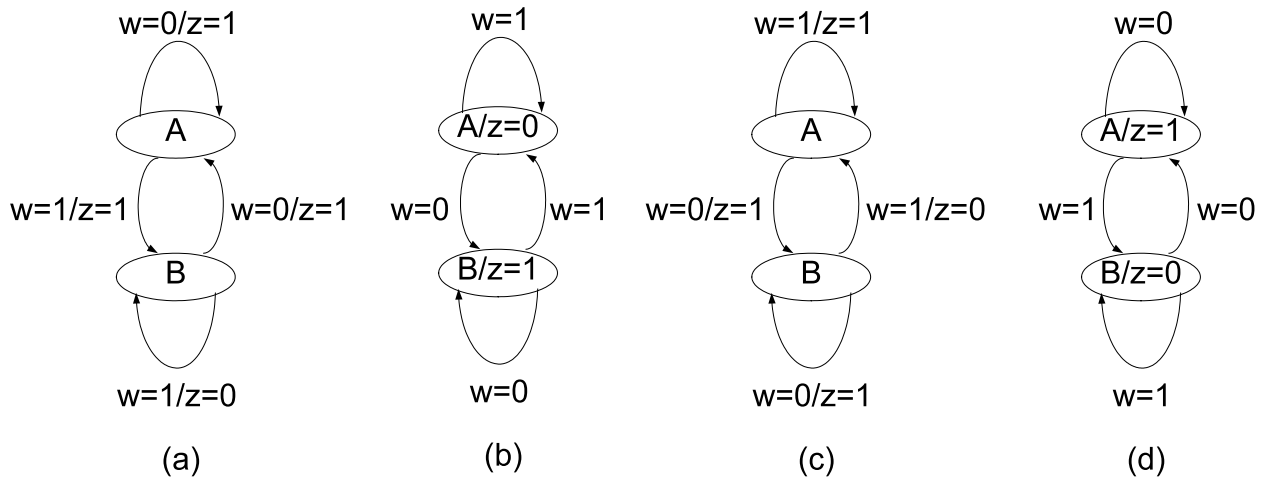


Figure 16 - State transition diagrams for question 20.

- THE END -