



12 Pages

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
3. The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. **Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.**

SURNAME: Davies

FORENAME(S): Adam

[illegible]

(Gweler yr ochr arall am y fersiwn Cymraeg)

Have more



3.) Please See Print off labled :
Question - 3

4a.) Please See Print off labled :
Question - 4a

~~4b~~
4b.) Please See Print off labled :
Question - 4b.

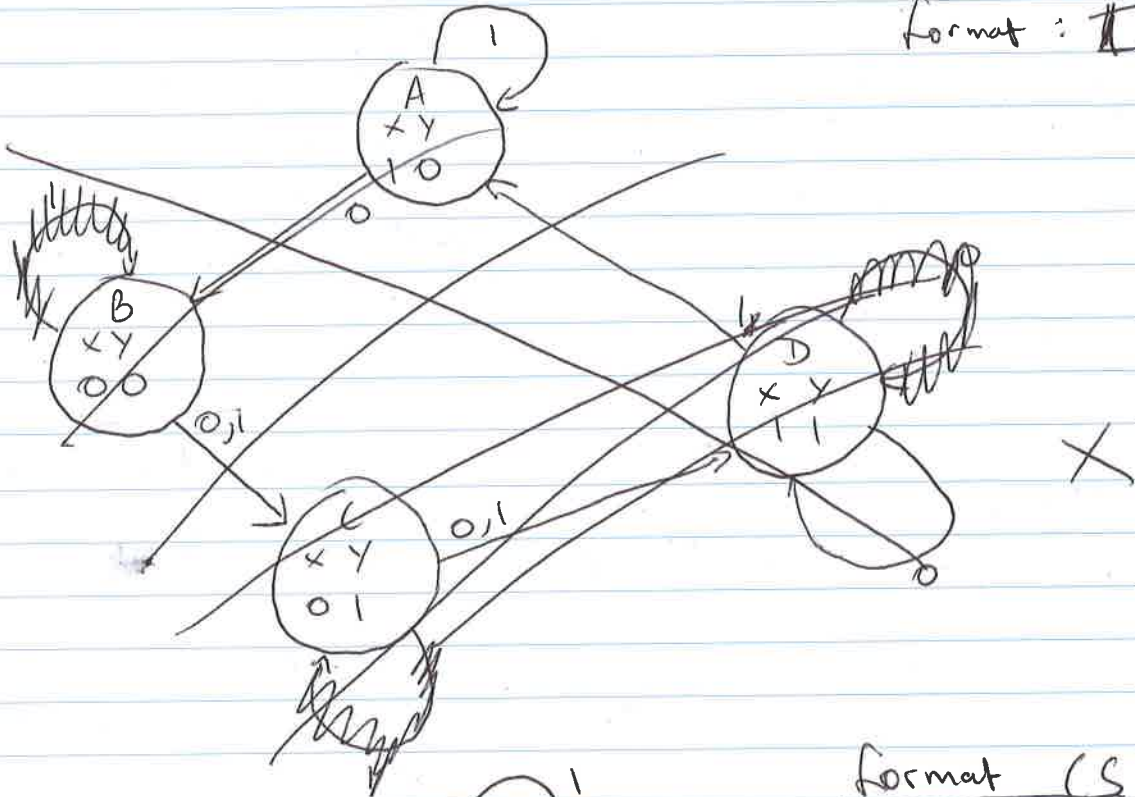
Q1a)

When up $(CS = 1) = 10$
 When down $(CS = 0) = 10$

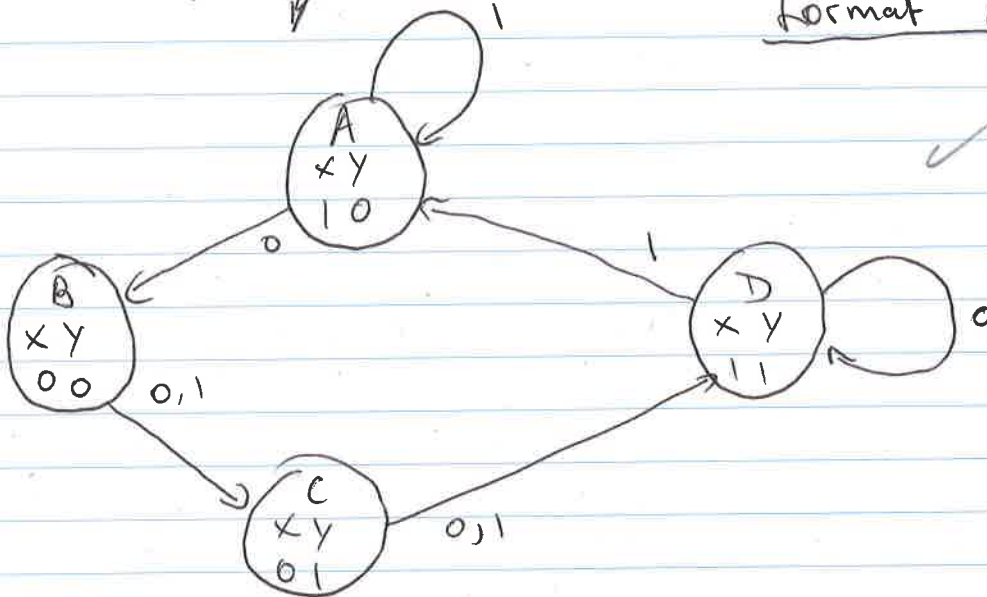
00
 01
 11

hold until back to $(CS = 1) = 10$.

format : ~~CS~~



format CS



4.

Present State		Input
A	B	CS
0	0	0
0	0	1

b.)

Present State			Next State	
X	Y	CS	X	Y
0	0	0	0	1
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

6

* See next page for ~~less~~ more presentable table

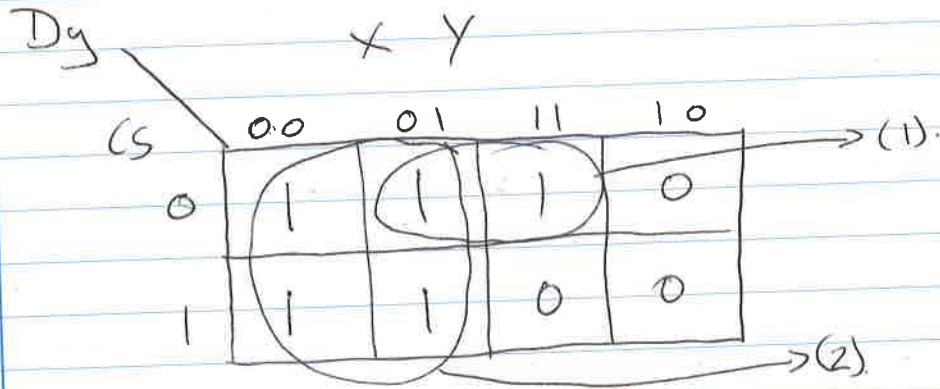
c.) Two flip flops used (D Type).
ci.)

D _x	X Y				
	CS	00	01	11	10
0	0	0	1	1	0
1	0	0	1	1	1

- (1) X . CS ✓
- (2) Y. ✓

$$D_x = X \cdot CS + Y$$

Question No.



$$(1) Y \cdot \overline{C_s}$$

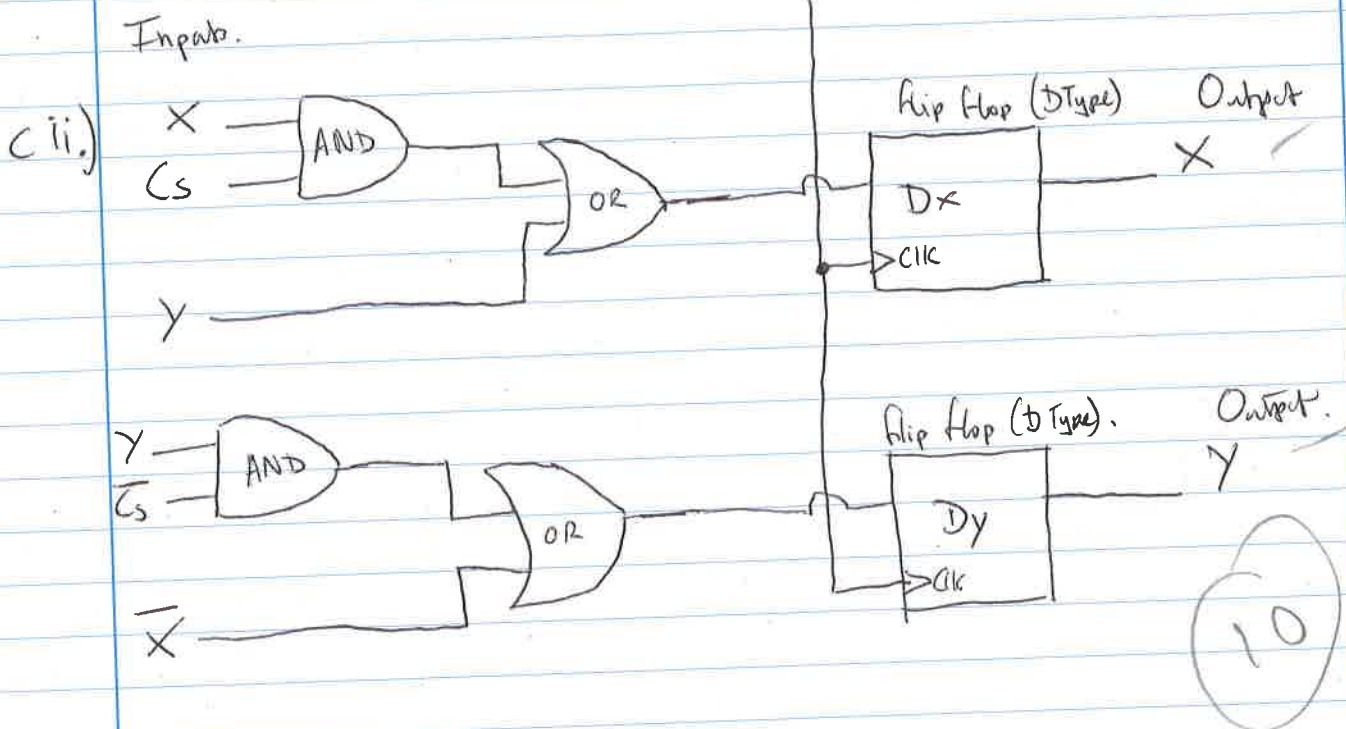
$$(2) \overline{X}$$

$$D_y = Y \cdot \overline{C_s} + \overline{X}$$

Terms =

$$D_x = X \cdot C_s + Y$$

$$D_y = Y \cdot \overline{C_s} + \overline{X}$$



*

~~State~~ ~~Present~~ ~~State~~

b) Continued.

State	Present State		Input CS	Next State		Out	
	X	Y		X	Y	D _x	D _y
B	0	0	0	0	1	0	1
	0	0	1	0	1	0	1
C	0	1	0	1	1	1	1
	0	1	1	1	1	1	1
A	1	0	0	0	0	0	0
	1	0	1	1	0	1	0
D	1	1	0	0	0	0	0
	1	1	1	1	0	1	0

```
1  -----
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:      11:30:18 05/18/2018
6  -- Design Name:
7  -- Module Name:      Question_3 - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity Question_3 is
33     Port ( X : in  STD_LOGIC;
34           CLK : in  STD_LOGIC;
35           Y : out  STD_LOGIC_VECTOR (2 downto 0));
36 end Question_3;
37
38 architecture Behavioral of Question_3 is
39
40     type state_type is (state_1, state_2, state_3, state_4, state_5, state_6,
41 state_7, state_8);
42     signal present_state : state_type := state_1;
43     signal next_state : state_type := state_1;
44
45 begin
46     State : process (X, present_state)
47     begin
48
49         if X = '0' then
50             case present_state is
51                 when state_1 => next_state <= state_1; Y <= "000";
52                 when state_2 => next_state <= state_5; Y <= "101";
53                 when state_3 => next_state <= state_4; Y <= "100";
54                 when state_4 => next_state <= state_6; Y <= "011";
55                 when state_5 => next_state <= state_6; Y <= "011";
```

```
56      when state_6 => next_state <= state_8; Y <= "001";
57      when state_7 => next_state <= state_8; Y <= "001";
58      when state_8 => next_state <= state_2; Y <= "010";
59      when others => NULL;
60      end case;
61
62      else if X = '1' then
63          case present_state is
64              when state_1 => next_state <= state_2; Y <= "010";
65              when state_2 => next_state <= state_3; Y <= "110";
66              when state_3 => next_state <= state_6; Y <= "011";
67              when state_4 => next_state <= state_5; Y <= "101";
68              when state_5 => next_state <= state_3; Y <= "110";
69              when state_6 => next_state <= state_7; Y <= "111";
70              when state_7 => next_state <= state_3; Y <= "110";
71              when state_8 => next_state <= state_1; Y <= "000";
72              when others => NULL;
73          end case;
74      end if;
75  end if;
76  end process State;
77
78  Clocked : process (CLK)
79  begin
80
81      if ((CLK = '1') and (CLK'last_value='0')) then
82          present_state <= next_state;
83      end if;
84  end process Clocked;
85
86
87  end Behavioral;
88
89
```

```
1  -----
2  -- Company:
3  -- Engineer:   Adam Davies (p147908)
4  --
5  -- Create Date:    11:43:34 05/18/2018
6  -- Design Name:
7  -- Module Name:    Question_4a - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19  -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity Question_4a is
33     Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
34           Y : out  STD_LOGIC_VECTOR (7 downto 0));
35 end Question_4a;
36
37 architecture Behavioral of Question_4a is
38
39 begin
40
41     Decoder : process (X)
42     begin
43
44         case X is
45             when "000" => Y <= "11111110";
46             when "001" => Y <= "11111101";
47             when "010" => Y <= "11111011";
48             when "011" => Y <= "11110111";
49             when "100" => Y <= "11101111";
50             when "101" => Y <= "11011111";
51             when "110" => Y <= "10111111";
52             when "111" => Y <= "01111111";
53             when others => NULL;
54         end case;
55     end process Decoder;
56
```



57 end Behavioral;

58

59

```
1  -----
2  -- Company:
3  -- Engineer: Adam Davies (p147908)
4  --
5  -- Create Date:      11:49:22 05/18/2018
6  -- Design Name:
7  -- Module Name:      Question_4b - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19  -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity Question_4b is
33     Port ( X : in  STD_LOGIC;
34           CLK : in  STD_LOGIC;
35           Y : out STD_LOGIC);
36 end Question_4b;
37
38 architecture Behavioral of Question_4b is
39
40     type state_type is (state_a, state_b, state_c, state_d, state_e, state_f,
41 state_g);
42     signal present_state : state_type := state_a;
43     signal next_state : state_type := state_a;
44
45 begin
46     State : process (X, present_state)
47     begin
48
49         if X = '0' then
50             case present_state is
51                 when state_a => next_state <= state_a; Y <= '0'; ✓
52                 when state_b => next_state <= state_c; Y <= '0'; ✓
53                 when state_c => next_state <= state_c; Y <= '0'; ✓
54                 when state_d => next_state <= state_e; Y <= '0'; ✓
55                 when state_e => next_state <= state_a; Y <= '0'; ✓
```

```
56     when state_f => next_state <= state_g; Y <= '0';
57     when state_g => next_state <= state_a; Y <= '1';
58     when others => NULL;
59     end case;
60
61     else if X = '1' then
62         case present_state is
63             when state_a => next_state <= state_b; Y <= '0';
64             when state_b => next_state <= state_c; Y <= '0';
65             when state_c => next_state <= state_d; Y <= '1';
66             when state_d => next_state <= state_f; Y <= '1';
67             when state_e => next_state <= state_a; Y <= '0';
68             when state_f => next_state <= state_f; Y <= '1';
69             when state_g => next_state <= state_a; Y <= '1';
70             when others => NULL;
71         end case;
72     end if;
73 end if;
74 end process State;
75
76 Clocked : process (CLK)
77 begin
78
79     if ((CLK = '1') and (CLK'last_value='0')) then
80         present_state <= next_state;
81     end if;
82 end process Clocked;
83
84
85 end Behavioral;
86
87
```

13