The University of Alabama in Huntsville
ECE Department
EE 202 – 02
Fall 2010
Final Exam Solution

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

D	Q(t+1)
0	0
1	1

T	Q(t+1)
0	Q(t)
1	Q'(t)

1. (1 point) A major trend in digital design methodology is the use of a _hardware description language_ to describe and simulate the functionality of a digital circuit.
2. (1 point) _Propagation delay_ is the average transition delay time for a signal to propagate from input to output.
3. (1 point) A _parity bit_ is an extra bit included with a binary message to make the number of 1's either odd or even.
4. (1 point) A multiplexer can be constructed with three-state gates: high, low and _high impedance_.
5. (1 point) A register capable of transferring the binary information held in each cell to its neighboring cell, in a selected direction, is called a _shift register_.
6. (5 points Convert (1022101_3) to decimal:

$$1022101_3 = 1 \times 3^6 + 0 \times 3^5 + 2 \times 3^4 + 2 \times 3^3 + 1 \times 3^2 + 0 \times 3^1 + 1 \times 3^0$$
$$= 1 \times 729 + 0 \times 243 + 2 \times 81 + 2 \times 27 + 1 \times 9 + 0 \times 3 + 1 \times 1$$
$$= 729 + 0 + 162 + 54 + 9 + 0 + 1 = 955_{10}$$

7. (10 points) Convert decimal +18 and +25 to binary, using the signed-2's-complement representation and enough digits to accommodate the numbers. Then perform the binary equivalent of (+18) + (-25). Convert the answer back to decimal and verify that it is correct.

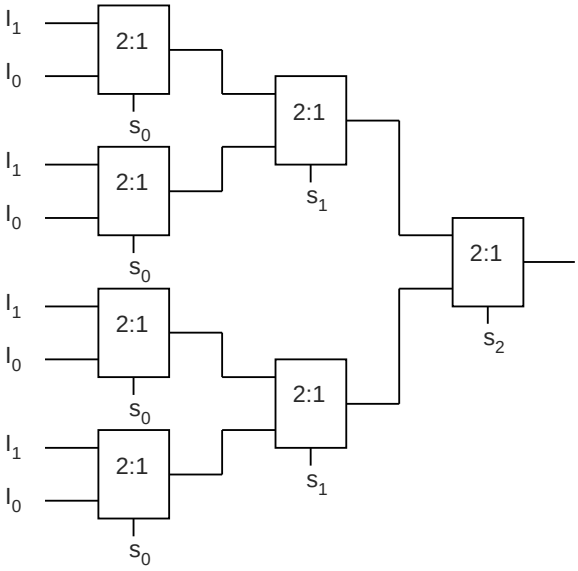
2	18		2	25		2	7	
2	9	0	2	12	1	2	3	1
2	4	1	2	6	0	2	1	1
2	2	0	2	3	0	2	0	1
2	1	0	2	1	1			
2	0	1		0	1			

+18 = 001 0010 +25 = 001 1001 -25 = -32 + 7 -25 = 10 0111

+18 = 001 0010 18 - 25 = -7
+(-25)= 110 0111

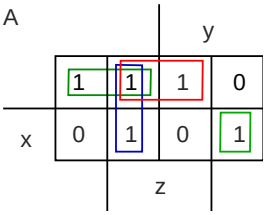
$$111\ 1001 = 1 \times -2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= -64 + 32 + 16 + 8 + 0 + 0 + 1$$
$$= -7 \checkmark$$

8. (10 points) Construct an 8 × 1 multiplexer with as many 2 × 1 multiplexers and any additional logic that you might need. Use block diagrams for the components.

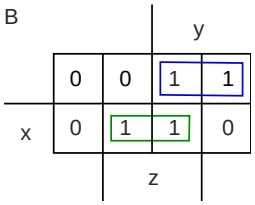


9. (15 points) Design a circuit that implements the following truth table. You do not have to draw a circuit diagram.

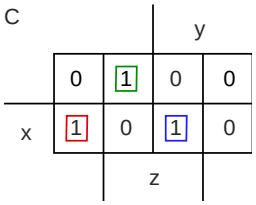
x	y	z		A	B	C
0	0	0		1	0	0
1	0	1		1	1	0
0	1	0		0	1	0
1	1	1		0	1	1
1	0	0		0	0	1
0	0	1		1	0	1
1	1	0		1	0	0
0	1	1		1	1	0



A = x'y' + x'z + y'z + xyz'



B = xz + x'y



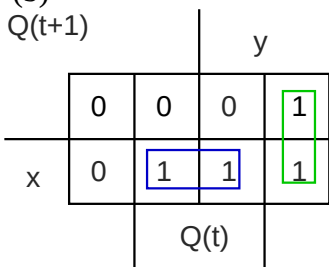
C = xy'z' + x'y'z + xyz

10. (15 points) An XY flip-flop has four operations, clear to 0, complement, no change, and set to 1, when inputs are X and Y are 00, 01, 10, and 11, respectively.
    (a) Tabulate the characteristic table.
    (b) Derive the characteristic equation.
    (c) Tabulate the excitation table.

(a)

X	Y	Q(t)	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(b)



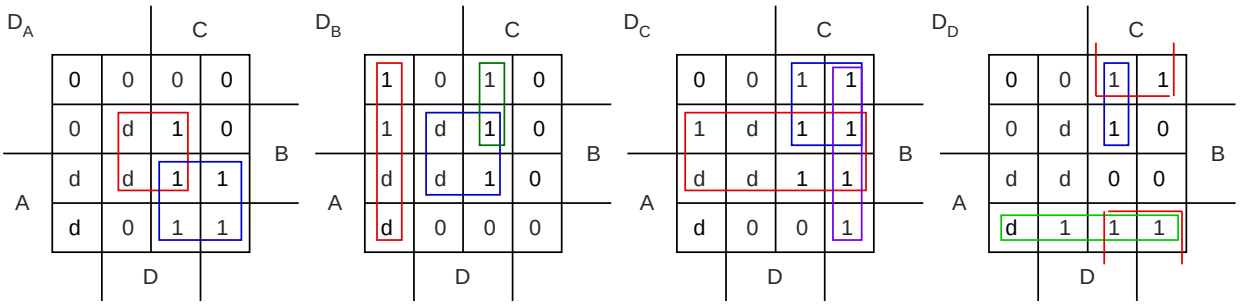
Q(t+1) = XQ(t) + YQ'(t)

(c)

Q(t)	Q(t+1)	X	Y
0	0	d	0
0	1	d	1
1	0	0	d
1	0	1	d

11. (20 points) Design a 4-bit counter which counts in the sequence 0000, 0100, 0110, 0010, 0011, 0111, 1111, 1110, 1010, 1011, 1001, 0001, 0000 using clocked D flip-flops. You do not have to draw the circuit diagram. Is the counter self-correcting if it comes up in an unused state?

Current State	Next State	D_A	D_B	D_C	D_D
0000	0100	0	1	0	0
0100	0110	0	1	1	0
0110	0010	0	0	1	0
0010	0011	0	0	1	1
0011	0111	0	1	1	1
0111	1111	1	1	1	1
1111	1110	1	1	1	0
1110	1010	1	0	1	0
1010	1011	1	0	1	1
1011	1001	1	0	0	1
1001	0001	0	0	0	1
0001	0000	0	0	0	0



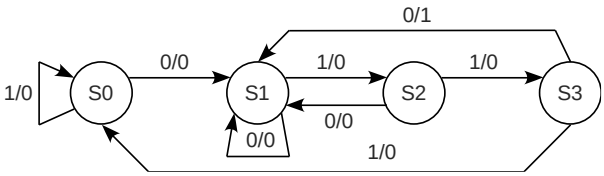
$D_A = AC + CD$ $D_B = C'D' + BD + A'CD$ $D_C = B + A'C + CD'$ $D_D = AB' + B'C + A'CD$

Unused States	D_A	D_B	D_C	D_D	Next States
0101	1	1	1	0	1110, correcting
1000	0	1	0	1	0101, 1110, correcting
1100	0	1	1	0	0110, correcting
1101	1	1	1	0	1110, correcting

12. (20 points) Design a Mealy sequential circuit that has one input and one output. This circuit has an output of 1 whenever its input string has the string 0110 in sequence and otherwise has an output of 0. Two sequences can overlap. Use JK flip-flops. You do not have to draw the circuit diagram.

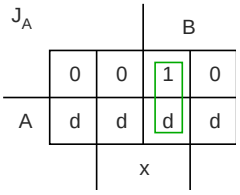
Input: 0110 1101 1101 0111 1110 0011 0101 0111 11
Output: 0001 0010 0000 0000 0000 0000 1000 0000 00

S0: no valid part of the sequence yet
S1: received 0
S2: received 01
S3: received 011

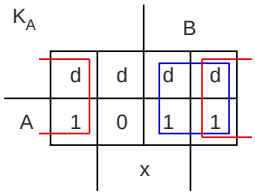


Current State	Input	Next State	Output
S0	0	S1	0
S0	1	S0	0
S1	0	S1	0
S1	1	S2	0
S2	0	S1	0
S2	1	S3	0
S3	0	S1	1
S3	1	S0	0

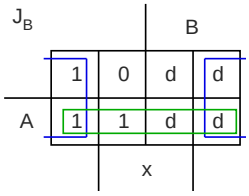
Current State	Input	Next State	Output	J_A	K_A	J_B	K_B
00	0	01	0	0	d	1	d
00	1	00	0	0	d	0	d
01	0	01	0	0	d	d	0
01	1	10	0	1	d	d	1
10	0	01	0	d	1	1	d
10	1	11	0	d	0	1	d
11	0	01	1	d	1	d	0
11	1	00	0	d	1	d	1



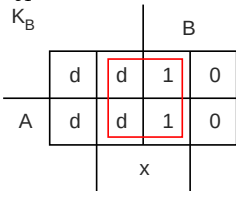
$J_A = Bx$



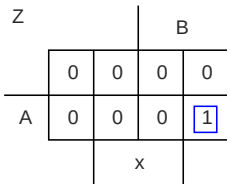
$K_A = B + x'$



$J_B = x' + A$



$K_B = x$



$Z = ABx'$