

EE109: Introduction to Embedded Systems

Fall 2020 - Final Exam

11/21/20, 2PM – 3:40PM + 20 min to upload

[Complete all the information in the box below.]

Name: **SOLUTIONS**

Student ID: _____

Email: _____@usc.edu

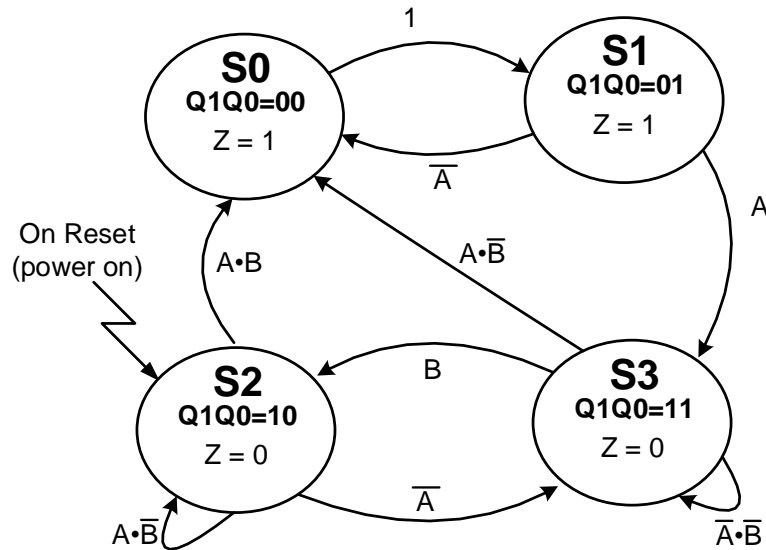
Lecture section (Circle One):

Annavaaram	Redekopp	Weber	Weber
9 a.m.	11 a.m.	12:30 p.m.	2 p.m.

Calculators ARE allowed.

Page	Ques.	Your score	Max score	Recommended Time
1			0	0 min.
2	1		14	10 min.
3	2		15	15 min.
4	3		10	10 min.
5	4		12	15 min.
6	5		10	15 min.
7	6		12	20 min.
8	7		10	15 min.
	Total		83	100 min.
		Scan/Upload		20 min.

1. **State Machines I (14 pts.):** Consider the completed state diagram shown below to answer the questions below.



- a.) Complete the state transition table by filling in the next state columns and the output column in the table below.

Current State		Next State				Output
		A B = 0 0	A B = 0 1	A B = 1 0	A B = 1 1	
State	Q1 Q0	State*	State*	State*	State*	Z
S0	0 0	S_1_	S_1_	S_1_	S_1_	1_
S1	0 1	S_0_	S_0_	S_3_	S_3_	1_
S2	1 0	S_3_	S_3_	S_2_	S_0_	0_
S3	1 1	S_3_	S_2_	S_0_	S_2_	0_

- b.) To implement the reset condition, what should be connected to the following flip flop inputs?

- i.) The **SET** of the Q0 flip-flop should be connected to:

- RESET
- not RESET
- 0 (GND)**
- 1 (Vdd)

- ii.) The **CLR** of the Q0 flip-flop should be connected to:

- RESET**
- not RESET
- 0 (GND)
- 1 (Vdd)

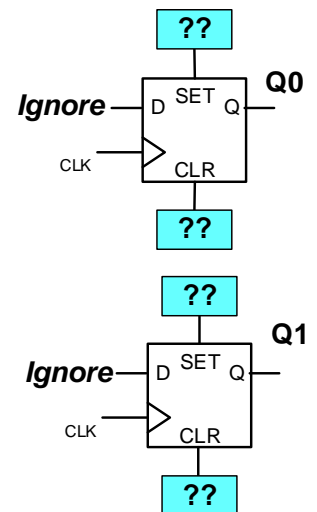
- iii.) The **SET** of the Q1 flip-flop should be connected to:

- RESET**
- not RESET
- 0 (GND)
- 1 (Vdd)

- iv.) The **CLR** of the Q1 flip-flop should be connected to:

- RESET
- not RESET
- 0 (GND)**
- 1 (Vdd)

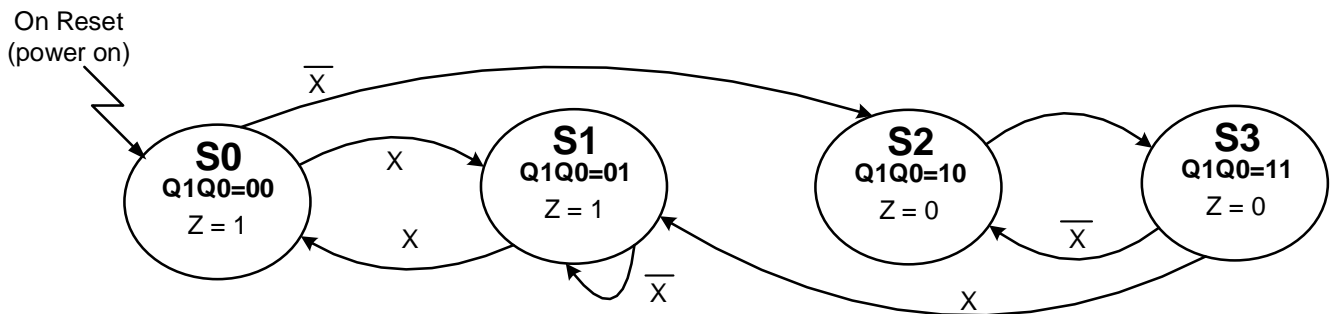
RESET
—



2. **State Machines II (15 pts).** Given the state transition table below, answer the following questions:

Current State		Next State				Output
State	Q1Q0	State*	X=0 Q1*Q0*	State*	X=1 Q1*Q0*	Z
S0	0 0	S2	1 0	S1	0 1	1
S1	0 1	S1	0 1	S0	0 0	1
S2	1 0	S3	1 1	S3	1 1	0
S3	1 1	S2	1 0	S1	0 1	0

- a.) Use the state table above to complete the corresponding state diagram (**fill in/draw all the correct state transitions** and be sure to label them correctly based on the table). Combine transition arrows where appropriate. **Fill in the Z output values** for each state.



- b.) Find a minimal, SOP equation for D1 (input to the Q1 flip-flop) and a minimal equation (SOP or POS) for Z. Show work and put your final answer in the blanks below.

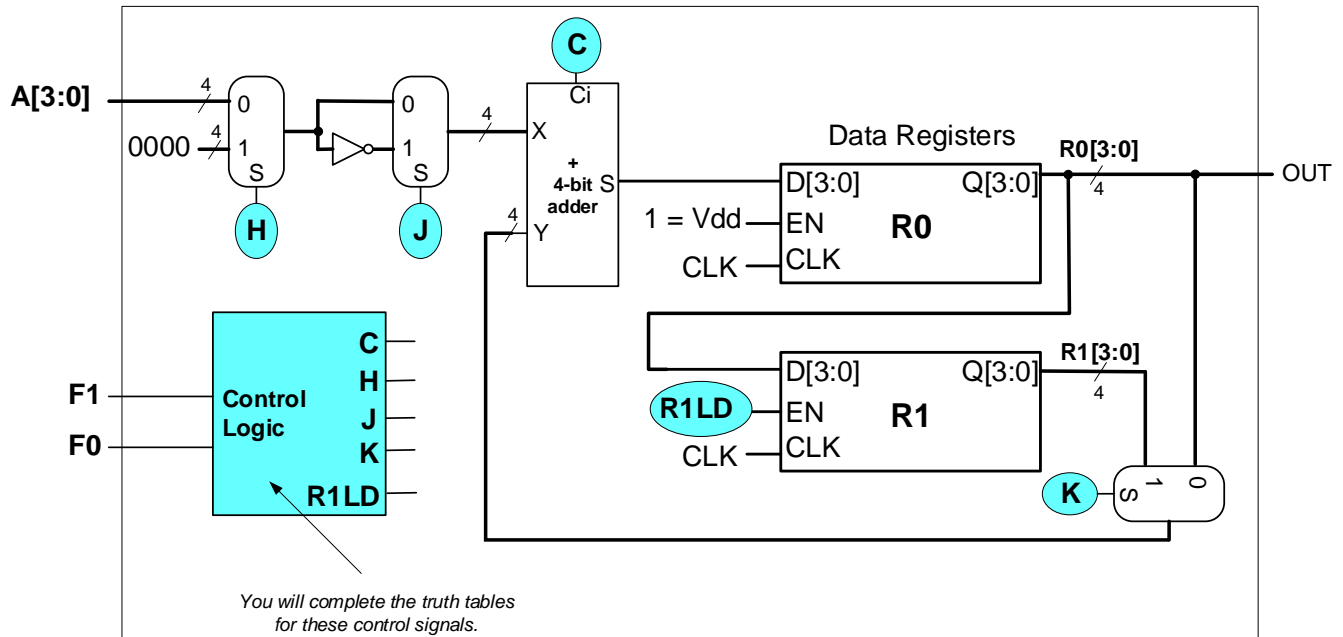
D₁ Map

Q ₁ Q ₀	X	
	0	1
00	1	0
01	0	0
11	1	0
10	1	1

D₁ = Q1Q0' + X'Q1 + X'Q0'

Z = Q1'

3. **Datapath Design I (10 pts.):** Consider the datapath below with the accompanying table showing the correspondence of **desired operations** to the function select bits, F[1:0]. Complete the table for the control signals: H, J, C, K, and R1LD to achieve the desired operations. All input and output numbers are 2's complement numbers.



Complete the table for the values of H, J, C, K and R1LD. Use d for don't care where appropriate.

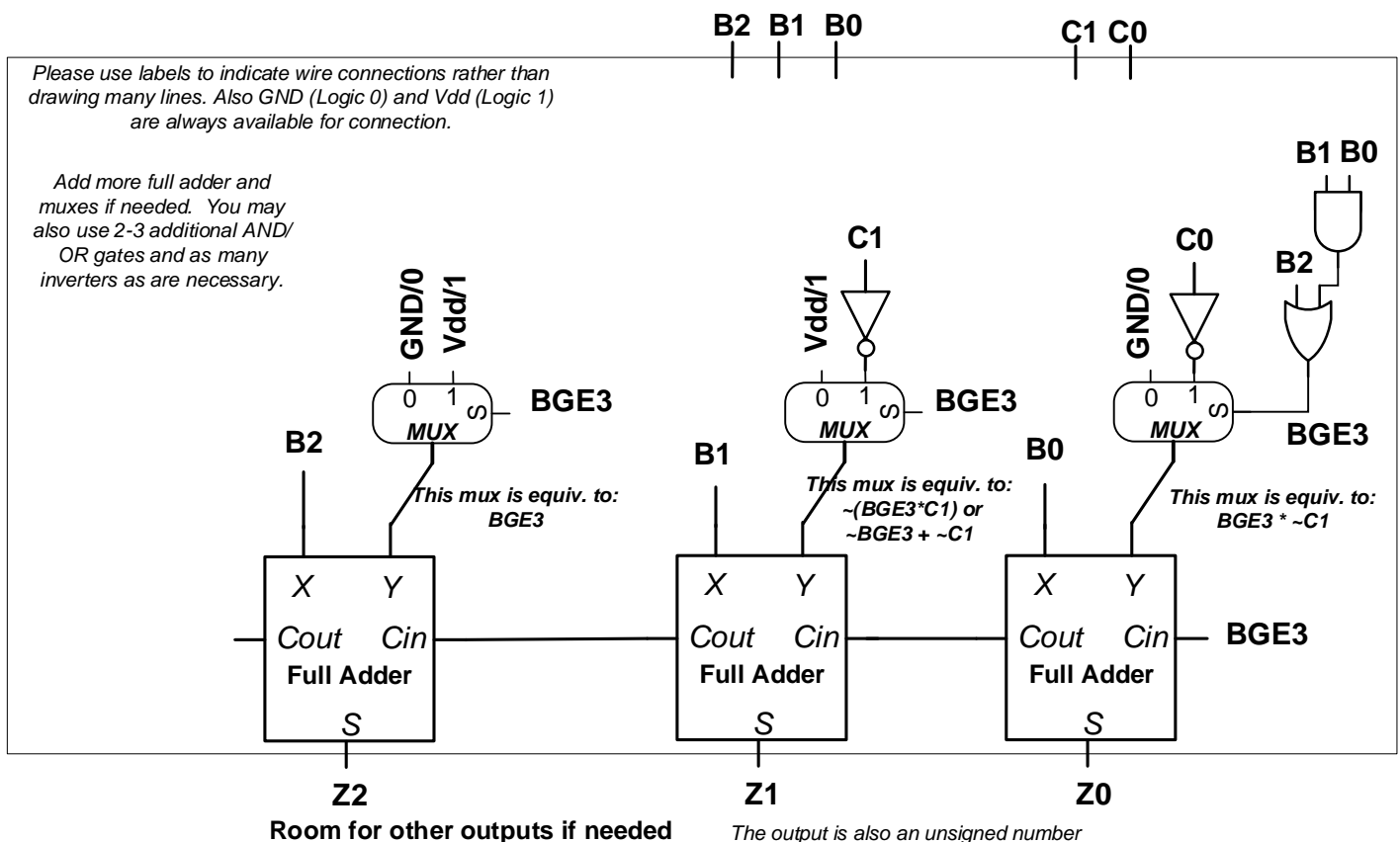
Desired Operation	F1	F0	H	J	C	K	R1LD
R0 = R0 - A R1 (no change)	0	0	0	1	1	0	0
R0 = R0 + 1 R1 (no change)	0	1	1	0	1	0	0
R0 = R1 - 1 R1 (no change)	1	0	1	1	0	1	0
Swap R0 and R1 values	1	1	1	0	0	1	1

4. **Datapath Design II (12 pts.):** Complete the combinational design below which takes in a 3-bit **unsigned** number, **B (B[2:0])** and 2-bit **unsigned** number **C (C[1:0])**. Your circuit should produce an **unsigned** output, **Z** (of appropriate number of bits) according to the description:

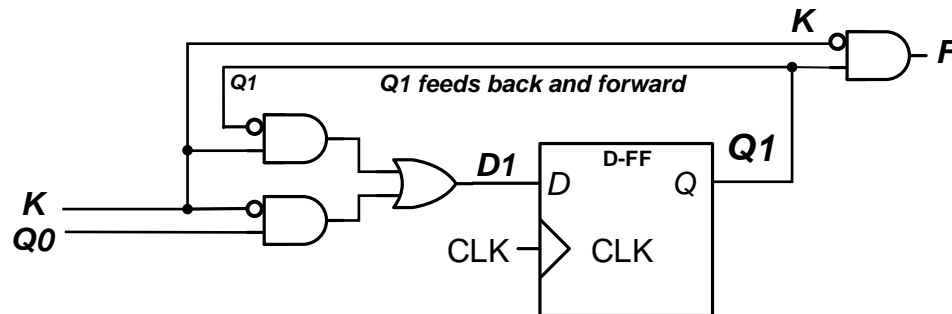
if ($B \geq 3$) then $Z = B - C$
 else $Z = B + 2_{10}$

You may draw additional full adders and/or 2-to-1 muxes if needed. You may use as many inverters as needed and may also use 5 additional simple AND/OR/NAND/NOR/XOR/XNOR gates.

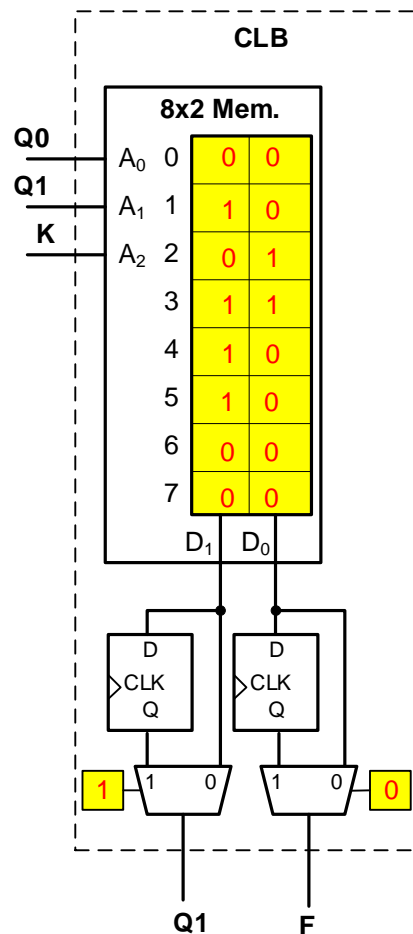
- a.) What is the decimal range of values that may be output for Z: 0 to 7
- b.) How many bits are required to represent that range: 3
- c.) Complete the logic for the signal BGE3 (i.e. the condition $B \geq 3$) in the box on the right.
- d.) Complete the rest of the circuit. Please label inputs with signal names rather than drawing long lines/wires.



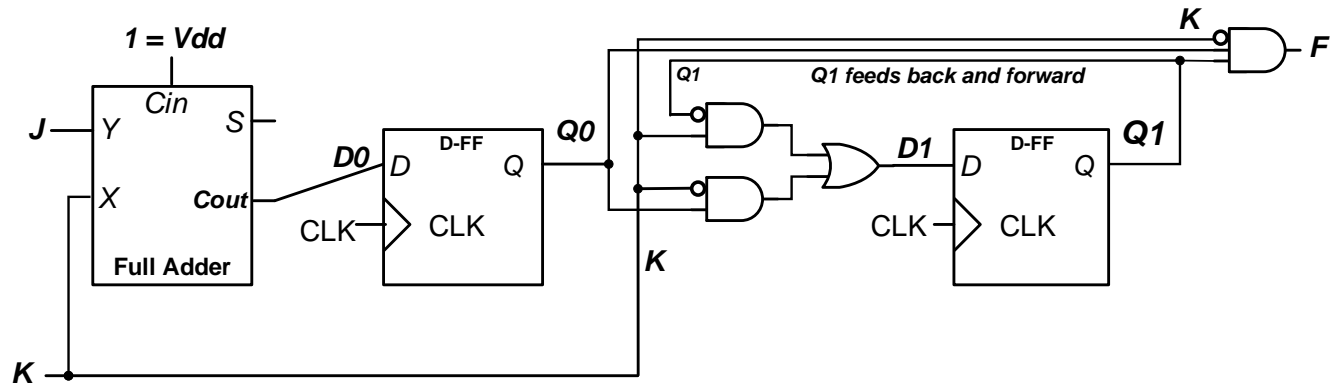
5. **FPGAs/Memories Design (10 pts.):** Given the desired hardware design below complete the following question.



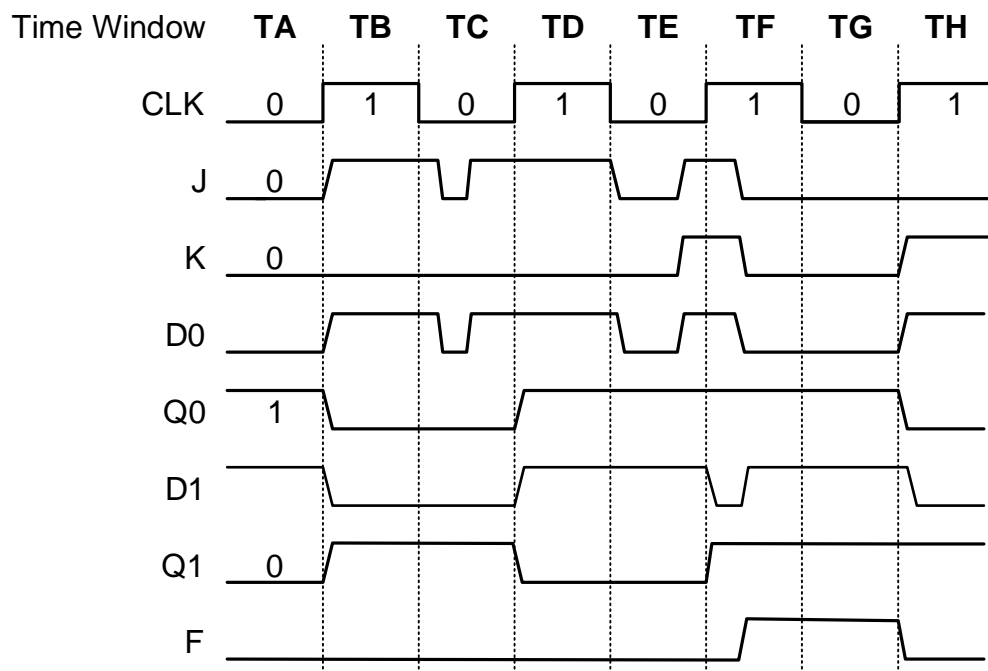
Show how to implement the design above using **one** 3-input, 2-output CLB by determining the contents of the 8x2 memory and the mux selects. Place a dash ('-') in any memory cell (bit place) that is a don't care.



6. **Sequential Logic (12 pts.):** Complete the waveform for the operation of the circuit below. You need to know the behavior or logic of the full adder and we cannot answer questions about how that works. **The flip-flops are positive-edge triggered.**



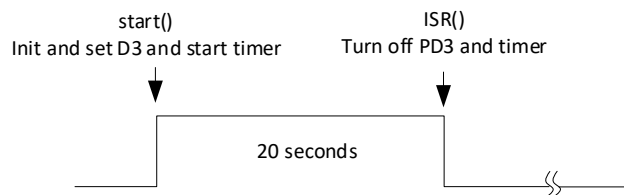
We have added guide/grid lines for D0, Q0, D1, Q1, and F. You must draw the actual waveforms given the circuit above. We have started Q0 and Q1 by giving their initial values.



7. **Arduino Coding and Timers (10 pts):** TIMER1 (16-bit timer) is to be used to generate one pulse of **20 seconds** on Port D, bit 3. When the "start" function is called, initialize bit D3 appropriately and start TIMER1. Some code is already provided. When the 20 seconds have elapsed, PD3 should go to zero and the timer should be stopped (should not run nor generate interrupts). You only need to generate one 20 second pulse and we do not want or need repeated 20-second pulses. **No delay functions are allowed.**

Assuming the prescaler is set to 256, a.) declare any needed global variables, b.) complete start() to initialize bit D3 and select a counter modulus, and c.) complete the ISR which should cause PD3 to be cleared to 0 after 20 second pulse as well as disabling the timer.

Recall the Arduino runs at 16 MHz.



```

/* Declare global variables */
int count = 0;

void start()
{
    /* Initialize D3 and output a 1 */
    DDRD |= (1 << 3);
    PORTD |= (1 << 3);
    /* Timer init code */
    TCCR1B |= (1 << WGM12); // set to appropriate CTC mode
    TIMSK1 |= (1 << OCIE1A); // enable the local interrupt enable
    // count * OCR1A should be 1.25 million (1,250,000).

    OCR1A = _____
    TCCR1B |= (1 << CS12); //turn on timer, prescaler of 256
}
ISR(TIMER1_COMPA_vect)
{ // No delay function may be used.
    count++;
    // can be == or >= or some other equivalent if..else structure
    // count * OCR1A should be 1.25 million (1,250,000).
    if(count == /* answers may vary */ ) {
        PORTD &= ~(1 << 3); // or PIND |= (1 << 3);
        TCCR1B &= ~(1 << CS12);
    }
}

```


Intentionally blank for scratch work. You may detach it but please turn it in with your exam:

Name: _____ Section time: _____

Single-Variable Theorems

(T1)	$X + 0 = X$	(T1')	$X \cdot 1 = X$	(Identities)
(T2)	$X + 1 = 1$	(T2')	$X \cdot 0 = 0$	(Null elements)
(T3)	$X + X = X$	(T3')	$X \cdot X = X$	(Idempotency)
(T4)	$(X')' = X$			(Involution)
(T5)	$X + X' = 1$	(T5')	$X \cdot X' = 0$	(Complement)

Two- and Three-Variable Theorems

(T6)	$X + Y = Y + X$	(T6')	$X \cdot Y = Y \cdot X$	(Commutativity)
(T7)	$(X + Y) + Z = X + (Y + Z)$	(T7')	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	(Associativity)
(T8)	$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$	(T8')	$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$	(Distributivity)
(T9)	$X + X \cdot Y = X$	(T9')	$X \cdot (X + Y) = X$	(Covering)
(T10)	$X \cdot Y + X \cdot Y' = X$	(T10')	$(X + Y) \cdot (X + Y') = X$	(Combining)
(T11)	$X \cdot Y + X' \cdot Z + Y \cdot Z =$ $X \cdot Y + X' \cdot Z$	(T11')	$(X + Y) \cdot (X' + Z) \cdot (Y + Z) =$ $(X + Y) \cdot (X' + Z)$	(Consensus)

DeMorgan's Theorem

$(X \cdot Y)' = X' + Y'$	$(X + Y)' = X' \cdot Y'$	(DeMorgan's)
--------------------------	--------------------------	--------------