Systems I
Fall 2012
Mid-term exam solutions

1. Question: (20 points) Prove that NOR is a *universal operator*. That is, prove that any logic function is equivalent to some expression that uses only the NOR operator. You may demonstrate equivalence of NOR with other operators using Boolean algebra or (labeled) circuit diagrams.

   Answer: Since a DNF expression can be formed for any function, and since DNF employs only AND, OR, and NOT, we infer that this trio of operations is, together, universal. If we can calculate these three using only NOR, then NOR is also universal.

   (a) NOT:

   $$\bar{x} = \overline{x + x} \quad \text{(self disjunction)}$$

   (b) OR:

   $$\begin{aligned} x + y &= \overline{\overline{x + y}} && \text{(double negation)} \\ &= \overline{(\overline{x + y}) + (\overline{x + y})} && \text{(self disjunction)} \end{aligned}$$

   (c) AND:

   $$\begin{aligned} xy &= \overline{\overline{xy}} && \text{(double negation)} \\ &= \overline{\bar{x} + \bar{y}} && \text{(DeMorgan's)} \\ &= \overline{(\overline{x + x}) + (\overline{y + y})} && \text{(self disjunction)} \end{aligned}$$

   Discussion: *To be added later...*

2. QUESTION: (20 points) Consider the logic function described by the following truth table:

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

(a) Write this function in *disjunctive normal form*.

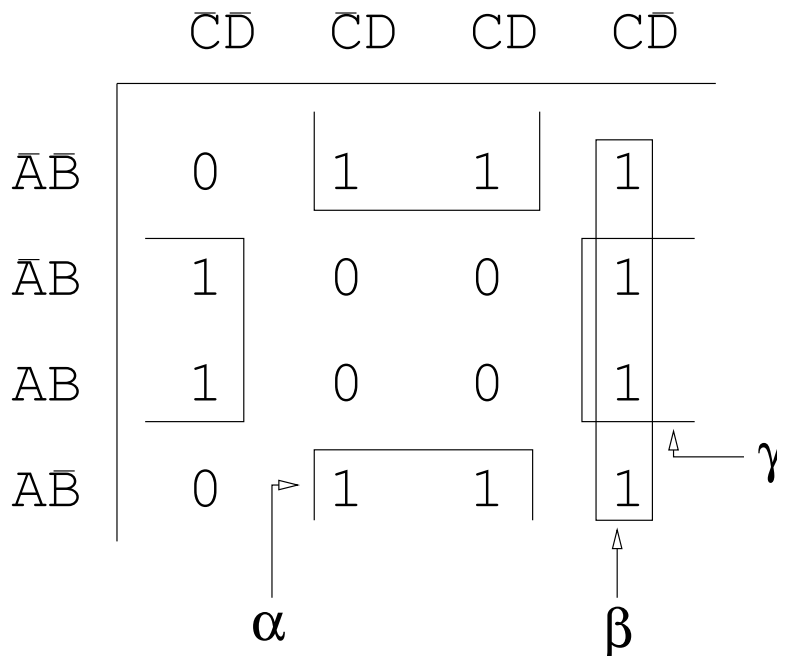
(b) Simplify this expression by using a *Karnaugh map*.

ANSWER:

(a) Disjunctive Normal Form:

$$\begin{aligned} Y = \quad & \bar{A}\bar{B}\bar{C}D + \\ & \bar{A}\bar{B}C\bar{D} + \\ & \bar{A}\bar{B}CD + \\ & \bar{A}B\bar{C}\bar{D} + \\ & \bar{A}BC\bar{D} + \\ & A\bar{B}\bar{C}D + \\ & A\bar{B}C\bar{D} + \\ & A\bar{B}CD + \\ & AB\bar{C}\bar{D} + \\ & ABC\bar{D} + \end{aligned}$$

(b) Karnaugh map:



$$Y = \alpha + \beta + \gamma$$
$$\bar{B}D + C\bar{D} + B\bar{D}$$

Discussion: *To be added later...*

3. QUESTION: (30 points) Create a circuit that calculates $a > b$, where a and b are 4-bit, two's-complement values. This circuit should have a 1-bit output that is 1 when $a > b$, and 0 otherwise. It should calculate the correct answer for all possible inputs. For any high-level components that you use (e.g., 1-bit full adders), you must **show their inner structure** by providing either a circuit diagram or logic functions.

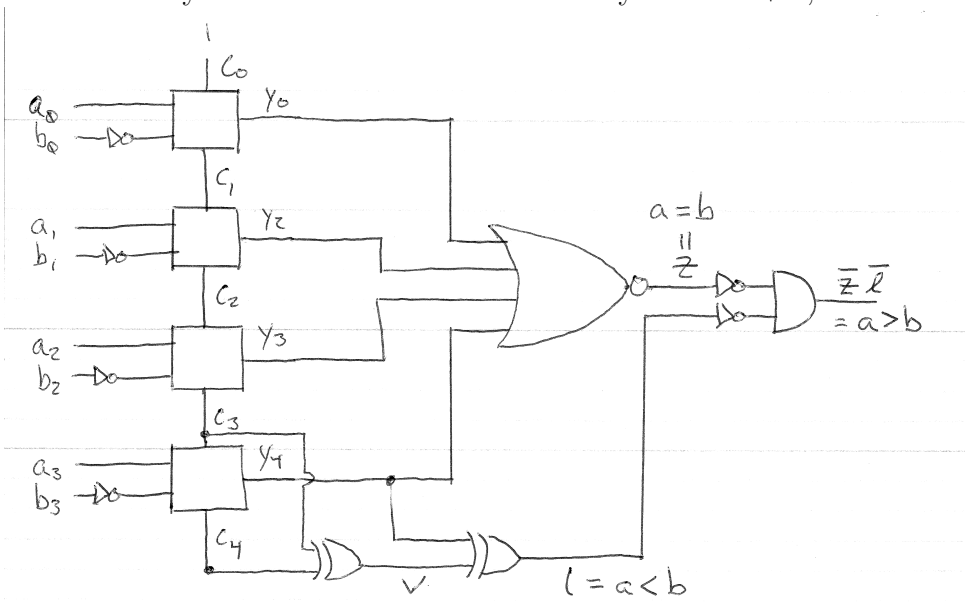
   ANSWER: A *1-bit full adder* is a device that accepts 3 inputs (c_I, a, and b), and emits 2 outputs (c_O and r). Define those outputs as:

   - $r = c_I \oplus a \oplus b$
   - $c_O = c_I a + c_I b + ab$

   Tautologically, $a > b$ is *true* if and only if:

   - $a < b$ is *false*, AND
   - $a = b$ is *false*.

   Consider a 4-bit ripple-carry subtracter, shown below. First, we have previously developed (in class) that $a < b$ can be determined by $l = y_3 \oplus v$, where v indicates overflow by computing $c_3 \oplus c_4$. This determination is stable in the presence of overflow.
   Second, $a = b$ if and only if $y = 0$, which implies that all bits of y are 0. Thus, $a = b$ corresponds to $z = \overline{y_3 + y_2 + y_1 + y_0}$.
   We can finally conclude that $a > b$ if and only if $\bar{l}\bar{z} = \overline{l + z}$, as shown:



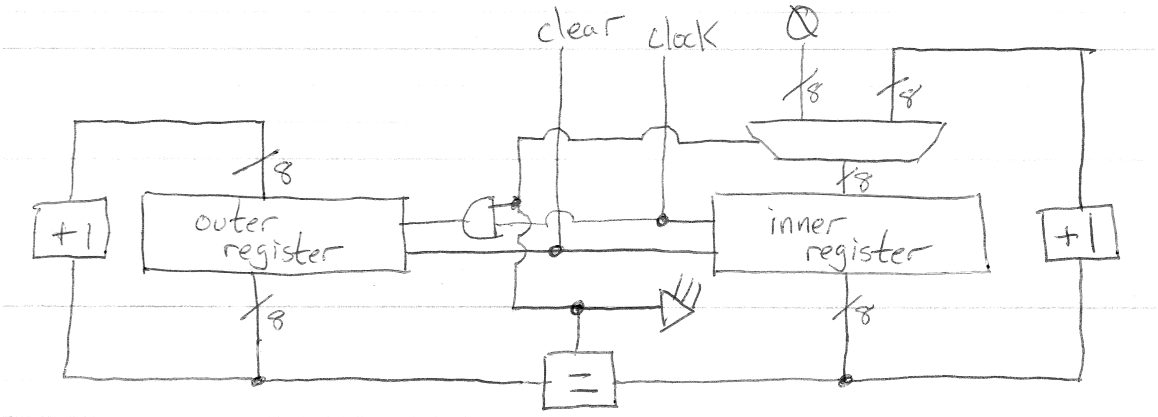
   DISCUSSION: *To be added later...*

4. QUESTION: (30 points) **Draw a circuit** that emits the following repeating sequence:

$$\underbrace{1}_{1}, \underbrace{0,1}_{2}, \underbrace{0,0,1}_{3}, \underbrace{0,0,0,1}_{4}, \underbrace{0,0,0,0,1}_{5}, \ldots, \underbrace{0,\ldots,0,1}_{256}, \underbrace{1}_{1}, \ldots$$

Assume that you have incoming CLOCK and CLEAR lines. You may use basic memory devices such as *flip-flops* (which may be aggregated into *registers*); you may also use other high-level components such as adders, multiplexers, demultiplexers, etc.

ANSWER: This circuit is constructed around **two** 8-bit registers. The *outer register* contains the number of zeros to print in this portion of the sequence—this *subsequence*. The *inner register* counts the position in the current subsequence. While the values of the two registers differ, a 0 is emitted, and the inner register increments while the outer remains unchanged. When the values of the two registers do match, a 1 is emitted, the inner is reset to 0, and the outer is incremented.



DISCUSSION: *To be added later...*