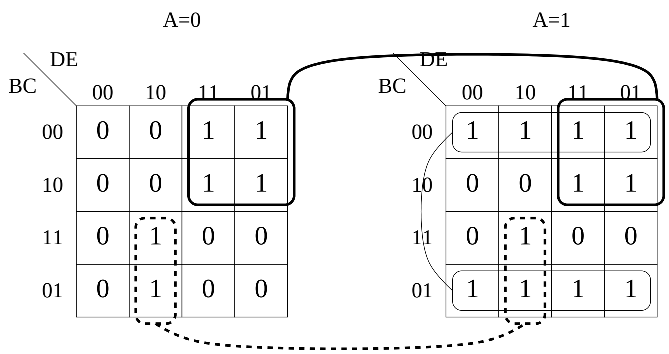
# Midterm Exam Solutions

Prof. Loh

*CS3220 - Processor Design - Spring 2005*

1. **Logic Design** Use the Karnaugh map given below for this problem.

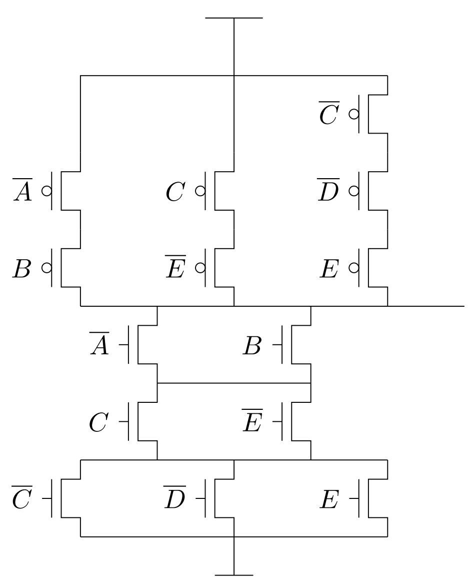



(a) Write a minimized boolean expression that corresponds to the function given by the Karnaugh map. [10 points]

$$A\overline{B} + \overline{C}E + CD\overline{E}$$

(b) Draw the CMOS circuit that implements the function. [15 points] Inverting the function gives us:

$$(\overline{A} + B).(C + \overline{E}).(\overline{C} + \overline{D} + E)$$



## 2. **Numbers**

(a) What is the largest value for a 5-bit *unsigned* integer? [5 points]

31

(b) What is the most negative (largest magnitude, farthest from zero) value for a 5-bit signed integer using 1's complement? [5 points]

-15

(c) What is the most negative value for a 5-bit signed integer using 2's complement? [5 points]

-16

Consider the 5-bit floating point format:

| Sign | Exp | | Mantissa | |
|---|---|---|---|---|
| s | e | e | m | m |

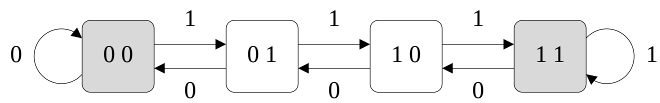
(d) What is the value of 11001? (You may leave the answer in fractional form) [5 points]

$$-1 \cdot 2^{2-1} \cdot 1.01_2 = -1 \cdot 2 \cdot \frac{5}{4} = -\frac{5}{2}$$

(e) What is the value of 01100? [5 points]

$$+\infty$$

3. **Finite State Machines**




Each box represents a state, with the name of the state given by the two bits ($\text{state}_1$ and $\text{state}_0$ – you can just write $s_1$ and $s_0$ for short). The arcs represent state transitions, and the value above the arc is the input value that causes the transition. The shaded states cause an output value of 1, and the unshaded states cause an output value of 0.

(a) Fill in the state transition table for the FSM shown above: [9 points]

| | Current State | | Current | Next State | |
|---|---|---|---|---|---|
| Input | $\text{state}_1$ | $\text{state}_0$ | Output | $\text{state}_1$ | $\text{state}_0$ |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

(b) Write the boolean equation corresponding to computing the "next state" value of $\text{state}_0$: [8 points]

| $Input$ \ $s_1 s_0$ | 00 | 10 | 11 | 01 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 0 |
| 10 | 1 | 1 | 1 | 0 |

$$Input.\overline{s_0} + Input.s_1 + s_1.\overline{s_o}$$

(c) Write the boolean equation to compute the output: [8 points]

| $Input$ \ $s_1 s_0$ | 00 | 10 | 11 | 01 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 |

$$s_1.s_0 + \overline{s_1}.\overline{s_o}$$
$$= \overline{s_1 \oplus s_0}$$

4. **Multi-Level Logic**

You have just moved out to California and you are working for a company that makes devices for gathering statistics on earthquakes. An employee recently quit his job and left an unfinished project. The device is supposed to be placed in the field to track the strength of earthquakes.

Whenever the earth moves even a little bit, a sensitive instrument measures the force and converts the force to an eight-bit number $X = x_7, x_6, ..., x_0$. The conversion works out such that the base-2 log of $X$ is the strength of the earthquake on the Richter scale.

The device that you have been assigned to tracks the *cumulative* distribution of earthquake intensities. The device has eight counters, where $Ri$ is the total number of earthquake measurements that were $i$ or higher on the Richter scale. For example, a measurement of $X = 5$ corresponds to just over 2 on the Richter scale. In this case, counters $R0, R1$ and $R2$ should be incremented, and all remaining counters remain the same.

Using only AND, OR and NOT gates, implement a circuit that generates the correct inputs to the adders (you may not need to use all three types of gates). Due to the timing constraints of the device, the logic you implement can only add at most five gate delays to the circuit.