## HW11 Solutions

## CS150 Fall 2010

1. 

Problem 2. Use the mask respin to create the following circuit. After using the testing device to find the bad addresses, program them in to the 20 fuse bits.


Problem 3. Datapath:


The controller is straightforward: Simply step through 4 states where you calculate the min, max, and sum, and then a final state to round off the mean.

## 2.

| 00 | 00 | 01 | 001 |
| :--- | :--- | :--- | :--- |
| 01 | 10 | 11 | 110 |
| 10 | 11 | 11 | 101 |
| 11 | 00 | 10 | 100 |

There are 28 bits in the ROM.

The alternative implementation requires 2^3*5 = 40 bits.
The first implementation's ROM would require fewer bits because it would not have duplicated copies of the output.
3.


## 4.

## 4.1

a) You tell him that K-maps are a clean, easy, systematic, and FUN! way of applying the rules of boolean algebra, something that becomes very difficult for complicated many-variable expressions.
b) No she is not confused, Verilog synthesizers uses K-maps in addition to many other tools to reduce boolean logic.
c) Nope. With a microcontroller you are constrained to writing software to work with given interfaces, with an FPGA you can develop custom hardware circuits for interfacing with other devices. High Speed serial IO is a common application, FPGAs find a lot of use in high speed network equipment, and even high frequency trading.
d) False. If you don't see this, please email a TA and ask for help.
e) To implement a clock that is 4 times slower than a given clock, we merely use a counter to make each clock cycle take 4 times as long as the original clock, or in other words each half cycle takes 2 of the original cycles. A very simple way to implement this is to have a straight ring counter and or'ing together the first two flip flops of the counter. The output of that or gate will be the new clock signal.

## 4.4

a)

|  |  |
| :--- | :--- | :--- |
| 00 | 1101010 |
| $01 \mid 11$ | 11000 |

10|0001101
11|0100 111
000 | 11010
001 | 01010
010| 11000
011| 11000
100|00 101
101 | 01101
110|01 111
111 | 00111
b) The one on the right implements a Mealy machine, therefore the ROM needs to account for an extra bit of address (the input bit). Furthermore, in a Mealy machine there is an extra degree of freedom in describing output values based on input, but since we are transcribing a Moore machine that information is just replicated.
c)

For a Moore machine, Rows $=$ Number of States $\left(2^{\wedge} 10\right)=1024$, Columns $=1$ For Output(20 bits) + 1 For Each Input(4 * 10 bits) = 60 bits.
For a Mealy machine, Rows $=2^{\wedge}($ Number of State Bits(10) + Number of Input Bits(4)) $=16384$, Columns $=1$ For Next State(10 bits) +1 For Output(20 bits) $=30$ bits

