CS150 Fall 2009

Final

NAME:__________________________ **SID**_______________________

Instructions

Read all of the instructions and all of the questions before beginning the exam.

There are 6 problems in this exam. The total score is 250 points. Points are given next to each problem to help you allocate time. Do not spend all your time on one problem.

Unless otherwise noted on a particular problem, you must show your work in the space provided, on the back of the exam pages or in the extra pages provided at the back of the exam**. Simply providing numerical answers will only result in partial credit**, even if the answers are correct.

Draw a BOX or a CIRCLE around your answers to each problem.

Be sure to provide units where necessary.

GOOD LUCK!

PROBLEM	POINTS	MAX
1		45
2		25
3		70
4		40
5		40
6		30

"Think of your happy place........"
-Happy Gilmore

Problem 1

Grab bag I ***45 points***

The error correction coding process can be viewed as a transformation from one space of bits (the unencoded data) to another (the coded data).

a) In the *(n,k)* notation for an error correction code, k is the unencoded data width in bits and n is the encoded width. Which is larger, n or k? Why must it be this way? *(5 points)*

b) Define the minimum Hamming distance, $\mathbf{d_{min}}$, of an error correction code: *(5 points)*

c) For a code with $\mathbf{d_{min}}$ minimum distance, what is the formula for the maximum number of errors, $\mathbf{E_{detect}}$, that can be *detected*? *(5 points)*

d) For a code with $\mathbf{d_{min}}$ minimum distance, what is the formula for the maximum number of errors, $\mathbf{E_{correct}}$, that can be *corrected*? *(5 points)*

e) One common type of linear code that is used in memory systems is the even parity code. This code adds one additional bit to each byte (for a total of 9 bits) so that the number of 1s in the result is even. What is the minimum distance of the (9,8) bit parity code? What are $\mathbf{E}_{\mathbf{detect}}$ and $\mathbf{E}_{\mathbf{correct}}$ for this code? *(5 points)*

f) Suppose that we have 9-bit drams (in fact, some of the RAMBUS DRAMS are like this). If we put 8 of them together we would have enough bits for 64 data bits and 8 parity bits. Suppose we code each group of 8 bits with a single parity bit. The result could be called a (72, 64) code. What is its minimum distance of this code and why? *(10 points)*

g) RAID level-5 is formed by xor-ing a number of data blocks together to produce a single parity block. Thinking of this code as grouping all of the first bits of every disk block together, then all of the second bits, etc, we see that this code is an extreme version of part f with 4K parity-encoded "bytes" concatenated together. What is its minimum distance of this code and why? *(10 points)*

"Sometimes I wish I had done more with my life than hanging out in front of places sellin' weed ... Like maybe be an animal doctor, I like seals...

-Clerks 2

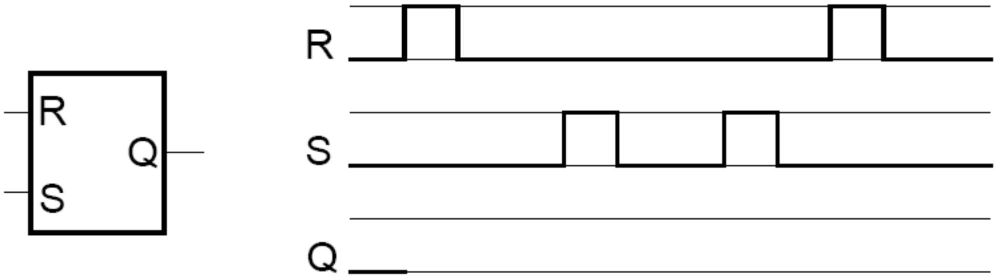
Problem 2

Grab bag II **25 points**

a) Given a word-addressable memory arranged as a stack of 256 words, what is the minimum **datapath** hardware required to calculate the average value of all of the words in the stack? Draw your **datapath** circuit below. Make sure to label any control bits an external controller would need. *(10 points)*

b) Provide RTL code that accomplishes the averaging function with the hardware you drew above. *(7.5 points)*

c) Draw in the output waveform for the asynchronous SR latch. *(7.5 points)*



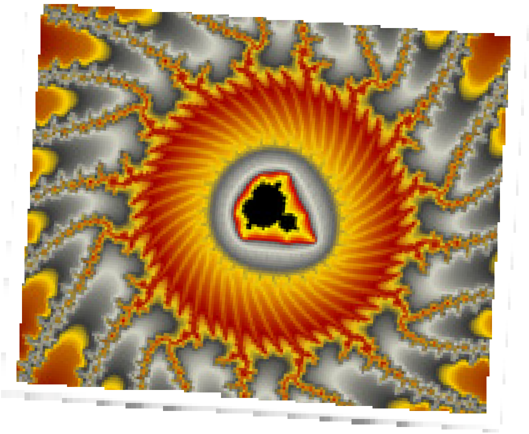
"Fight now, cry later."
-From Dusk Till Dawn

Problem 3

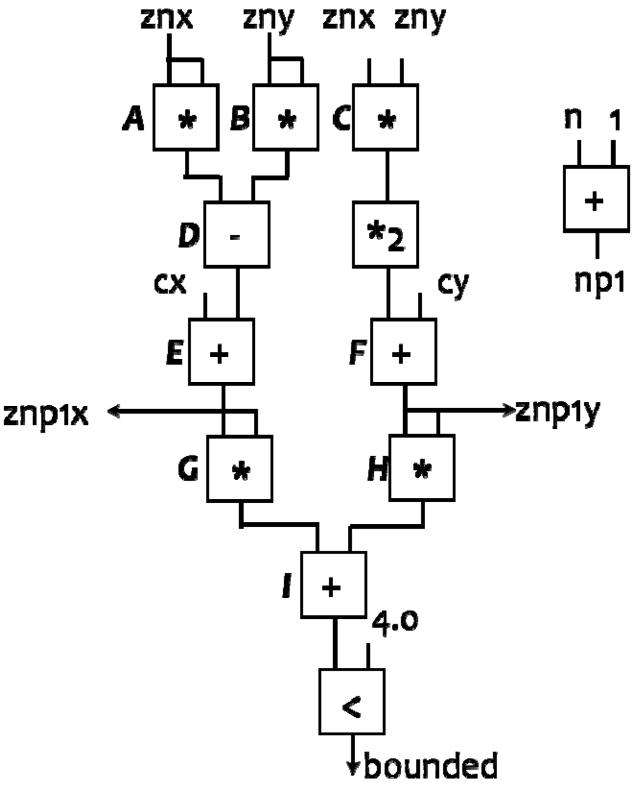
Controller for a custom datapath ***70 points***

A complex point c=x+jy is in the Mandelbrot set if the magnitude of the following sequence is bounded, as n goes to infinity:

$$z_0 = c$$
$$z_{n+1} = z_n^2 + c$$



We can estimate whether a point belongs to the Mandelbrot set by evaluating the sequence for a bounded number of iterations, and checking whether the magnitude of each element of the sequence remains below 4.0. Your task: design hardware to carry out one iteration of this computation.



Component	Latency
Multiply (*)	4 cycles
Add/Subtract (+/-)	3 cycles
Double (*2)	1 cycle
Compare (<)	1 cycle

Above is the data-flow graph for the computation of one iteration of the Mandelbrot set calculation.

Inputs: znx, the real part of z_n; zny, the imaginary part of z_n; cx, the real part of c; cy, the imaginary part of c; n, the current iteration count

Outputs: znp1x, the real part of z_{n+1}; znp1y, the imaginary part of z_{n+1}; bounded, which is a single bit which is true when z_{n+1} is still smaller than the threshold of 4.0; and np1, the resulting iteration count.

You need to design a module which implements the computation shown in the data-flow graph, using FP units with the specified latencies. The first step is to decide how many FP adders and multipliers you are going to use, and schedule the 9 FP operations onto your set of operators. Optimize the number of FP operators that you instantiate: if you use more operators than necessary, you will lose points.

a) In the table below, show your schedule by recording the initiation of each of the 9 operations, as mapped on to your set of components, by placing the letter (A-I) representing the operation, in the appropriate row and column. *(15 points)*

Cycle

Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

b) What is the overall latency of your design? *(7.5 points)*

c) Use this schedule to compute your schedule's efficiency. *(7.5 points)*
Define efficiency = (Realized floating point operations)/(Potential floating point operations).
What efficiency did your design achieve?

Complete your design by showing:
d) An FSM that outputs control signals to any multiplexers you need at the inputs of your FP operators *(20 points)*

e) A schematic of the new design, complete with all necessary pipeline registers and multiplexers. You don’t need to implement or draw circuits for the FSM, just denote outputs of the FSM as labeled wires in your schematic. *(20 points)*

"You may scream. There is no shame."
-Rambo, First Blood Part 2

Problem 4

Low-level logic design **40 points**

a) Using only inverters and passgates, construct a 2 input mux and an xor gate *(8 points)*

b) Draw the truth table for a full adder circuit, which takes in three 1-bit inputs (A, B, Cin) and has two 1-bit outputs: Sum and Carry. *(7 points)*

c) Using only inverters and passgates, or modules constructed of inverters and passgates, construct a full adder circuit. Hint: consider Carry as a function of (A xor B) and Cin. How many transistors did you use? *(15 points)*

d) Using your full adder circuit and your xor gate, construct a 4-bit wide ripple carry, twos-complement ALU that can compute either A+B or A-B for 4-bit inputs A, B. Use a 1 bit Subtract signal to choose the operation (0 = Add, 1 = Subtract). Produce a 4 bit Sum and a 1 bit Carry Out. *(15 points)*

"Kansas City Shuffle is when everybody looks right and you go left."

-Lucky Number Slevin

Problem 5

CMOS ***40 points***

Suppose I am interested in looking at the power consumption in a tiny wireless sensor mote. I am designing a tiny device (say <1 cm^3) which we'll call a **mote** that has:
a) a simple digital microcontroller,
b) a radio,
c) a MEMS sensor,
d) a battery.

Since I am an evil scientist, I am going to make hundreds of these motes and throw them into people's rooms to measure and report back on stuff (call me Prof. Big Brother). I need to know how long these motes will last before they run out of battery.

Given:

- The battery has 1000 J worth of energy in it and can put out 2.5 V as the power supply voltage (V_{DD}).
- I will make my digital microcontroller in CMOS.
- Every 10 minutes the mote computes something for 45 seconds.
- When not computing, all gates in the microcontroller remain connected to V_{DD} and GND.
- The IC module consists entirely of inverters (which is an awful assumption, btw)
- Each inverter sees a capacitance of 10fF at its output; this capacitance includes all contributions from wires, other inverters connected, etc.
- During the 45 seconds of activity, every inverter switches its output on each clock cycle with probability equal to 0.2.
- The clock speed is 200 MHz.
- The inverters have symmetrical rise and fall times (ie. they switch between GND and V_{DD} as fast as they switch between V_{DD} and GND).
- An inverter must force its output greater than 2.2 V for the next gate to "see" a logical 1
- An inverter must force its output less than 0.2 V for the next gate to "see" a logical 0
- On any cycle that an inverter is not switching, it "leaks" 1 nA from V_{DD} to GND.

a) What must the equivalent resistance of a single MOSFET be in order for the microcontroller to operate at 200 MHz? *(8 points)*

b) What would be the average current consumed by the microcontroller during the awake period? *(8 points)*

c) How many inverters, N, can I have in my design if I want to run the microcontroller for a year on the battery? *(14 points)*

d) At the suggestion of a colleague, you decide you will reduce the clock speed of your original design to 50 MHz during the activity burst. Assuming you make no other modifications to the circuit, what is the effect on power consumption for the *same computational task*? *(10 points)*

"As the sun sets slowly in the West, we bid you a fine farewell. "

-True Romance

Problem 6

FSMs and controllers ***30 points***

We want to design a smarter controller for our Evil Spy Mote© in Problem 4. This will sit outside the mote's microcontroller and be responsible for cycling power (and killing James Bond).

- Besides the standard system clock, assume there exists a clock, COMPCLOCK, which has a 50% duty cycle and a 1 minute period.

- The microcontroller, the radio and the audio sensor can be disconnected from the power supply independently via transistors.
  - The COMPON signal controls the state of the microcontroller power connection (connected or disconnected).
  - The SENSORON signal controls the state of the audio sensor power connection (connected or disconnected).
  - The RADIOON signal controls the state of the radio power connection (connected or disconnected).

- A separate temperature sensor cannot ever be turned off and reports on the ambient temperature to the power controller using two bits (TEMP):
  - 00 (unused)
  - 01 (normal office temperature)
  - 10 (night time office temperature)
  - 11 (body temperature)

- A normal computation is started when COMPCLOCK switches to high. This connects the mote block to the power rails (COMPON output) and powers up the sensor (SENSORON output).

- Once computation is finished, the microcontroller will set DONE to high.

- If TEMP = 01 when COMPCLOCK goes high, the radio must be turned on when the microcontroller issues the DONE signal and the microcontroller must be left on until the radio issues a TRANSDONE signal. Then everything must be turned off.

- If TEMP = 10 the radio should be left off and everything shut off once DONE is high.

- If TEMP = 11, follow the same procedure as TEMP = 01, except the EXPLODE_IN_BONDS_PANTS bit must be set to high once TRANSDONE is high.

- The BOND_BUTTON bit high disconnects power to all three blocks at any time.

a) Draw a Moore machine implementation of the controller. Please label all transitions and states neatly *(15 points)*

b) Draw a Mealy machine implementation of the controller. Please label all transitions and states neatly *(15 points)*