University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS150 J. Wawrzynek
Spring 2002 5/14/02

Midterm Exam III

Name:_____

ID number:_____

This is a *closed-book, closed-note* exam. No calculators please. You have 3 hours. Each question is marked with its number of points (one point per expected minute of time).

Put your name and SID on each page.
Show your work. **Write neatly** and be well organized.
Good luck!

problem	maximum	score
1	15pts	
2	5pts	
3	5pts	
4	10pts	
5	10pts	
6	8pts	
7	15pts	
8	10pts	
9	15pts	
total	93pts	

1. [15 pts] Short Answer.

a) [1pt] How many bits are contained in a "1M x 8" memory block?

b) [1pt] Which is cheaper per bit, SRAM or DRAM?

c) [1pt] Which type of memory uses a "floating gate" technology, DRAM or EPROM?

d) [1pt] Flash memory is closely related to SRAM, DRAM, or EPROM?

e) [1pt] How many AND-plane rows are needed in a PLA that implements the following set of logic equations:

X1 = a'b + cdf
X2 = ab + cde'
X3 = a'b + cde'

f) [1pt] What is the minimum number of parity bits needed to detect single bit errors in an 8-bit data word?

g) [2pt] List three factors that limit the degree to which pipelining can be effectively applied to a circuit.

h) [1pt] When a pipeline uses the "c-slow" technique, what is the pipeline throughput in terms of clock frequency F?

i) [1pt] List two causes of clock skew.

j) [2pt] Draw the circuit diagram for a reliable asynchronous signal synchronizer. Label the asynchronous input and synchronous output.

k) [1pt] Flip-flop metastability resolution time is i) less than 1 clock cycle, ii) between 1 and 2 clock cycles, or iii) unbounded.

l) [2pt] Draw the waveforms for "data", "request", and "acknowledge" in a self-timed data transfer, using the “4-cycle” signaling protocol:

2. [5pt] A data word is encoded using a Hamming code designed to correct single errors. The code word is written to memory then read back. The bit pattern read back is "111010000111".

a) What is the bit pattern of the *code word* that was written to memory?

b) What is the bit pattern of the original *data word* before encoding?

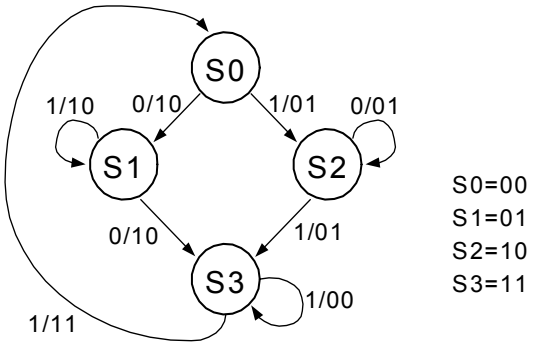
3. [5pt] A linear feedback shift register (LFSR) circuit produces the following sequence:

…, 011100, 111000, 110011, …

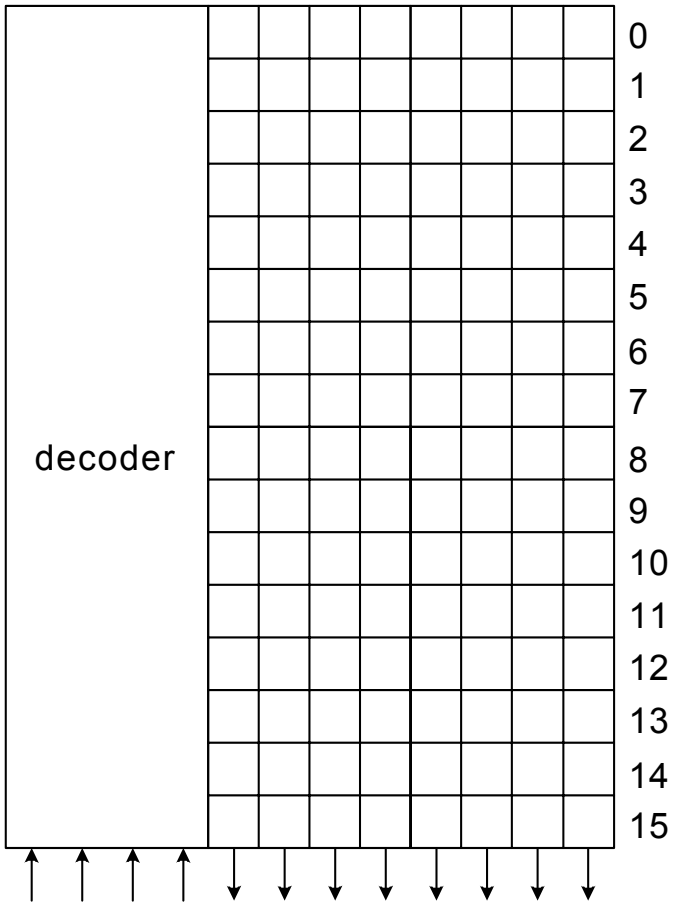
What is the primitive polynomial for the circuit?

4. [10pt] Sketch the design of the internal organization of a ROM block with 64 4-bit words. Assume that the shape of a single bit-cell is square and the cell array must be square. You do not need to show internal details of the bit-cell. Label inputs and outputs to the cell-array and mark its dimensions in terms of bit-cell. Label all ROM block input and output buses and indicate their widths.

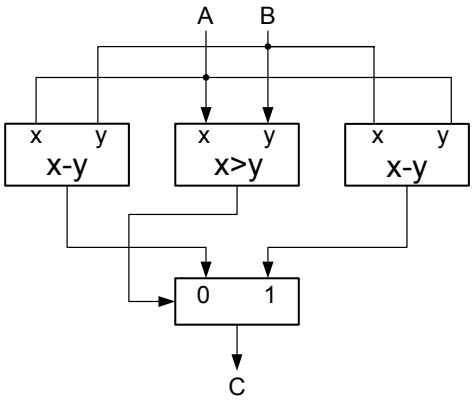
5. [10pt] Consider the Mealy finite state machine described by the following state transition diagram.



Below is drawn an 8-bit wide ROM block with 16 words. Using only this ROM block and flip-flops, draw a circuit that implements the FSM. Fill in the contents of the ROM and label all FSM inputs and output.



6. [8pts] The graph below is an abstract computation graph. It is composed of two subractors, a comparitor, and a multiplexor. **Draw a data-path and write out the RTL** level description for an implementation of this computation that attempts to minimize cost. Assume that the input A and B, and the output C, have associated registers. Label all signals that go to or come from the controller.



7. [15pt] Consider the design of a special purpose processor that computes:

$$y_i = bx_i + y_{i-1} + a$$

a, and *b* are constants supplied to the processor as inputs and held stable throughout the entire computation. x_i is stored in an array in memory beginning at address 0. y_i is output each cycle.

a) Below is the RTL description of the computation on a datapath with *only two registers*, Y and MAR, and no restrictions on the number of combinational logic blocks:

```
Y←0, MAR←0;
repeat n {
   Y←B × mem[MAR] + Y + A,  MAR←MAR+1;
    }
```

Assume that register setup and clk-to-Q times each = 0.5ns, adders take 10ns, multiplication 20ns, and memory reads take 10ns.

What is the minimum total time taken for the computation?

b) Now suppose we split the multiplier into two stages, separated by a register, each stage having a delay of 10ns. Furthermore, we restrict the data-path implementation to include 2 adders and the 2-stage multiplier, but with no restrictions on the number of registers or multiplexors.

What is the minimum total time taken for the computation?

Draw the resource utilization chart and write the RTL description:

Chart:

RTL:

8. [10pts] Consider the following computation described by the RTL:

```
Y←0;
repeat n {
   Y←Y⊕X;
}
Z←Y;
```

All registers are 1-bit wide. X is an input that arrives one value per cycle. After n cycles the output Z is generated. The value of Y is only used internally and not used as an output.

a) Draw the data-path that could be used for this computation. Show and label any control signals to/from the controller you may need:

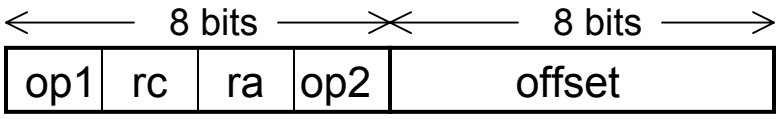
b) Draw the data-path that can accept 3 X values per cycle and reduce the total number of cycles needed by the computation by a factor of 3. Do not attempt to minimize the clock period.

c) Based on your answer from part b) draw a new data-path to show how to minimize the clock period, under the restriction that no gate has more than 2 inputs.

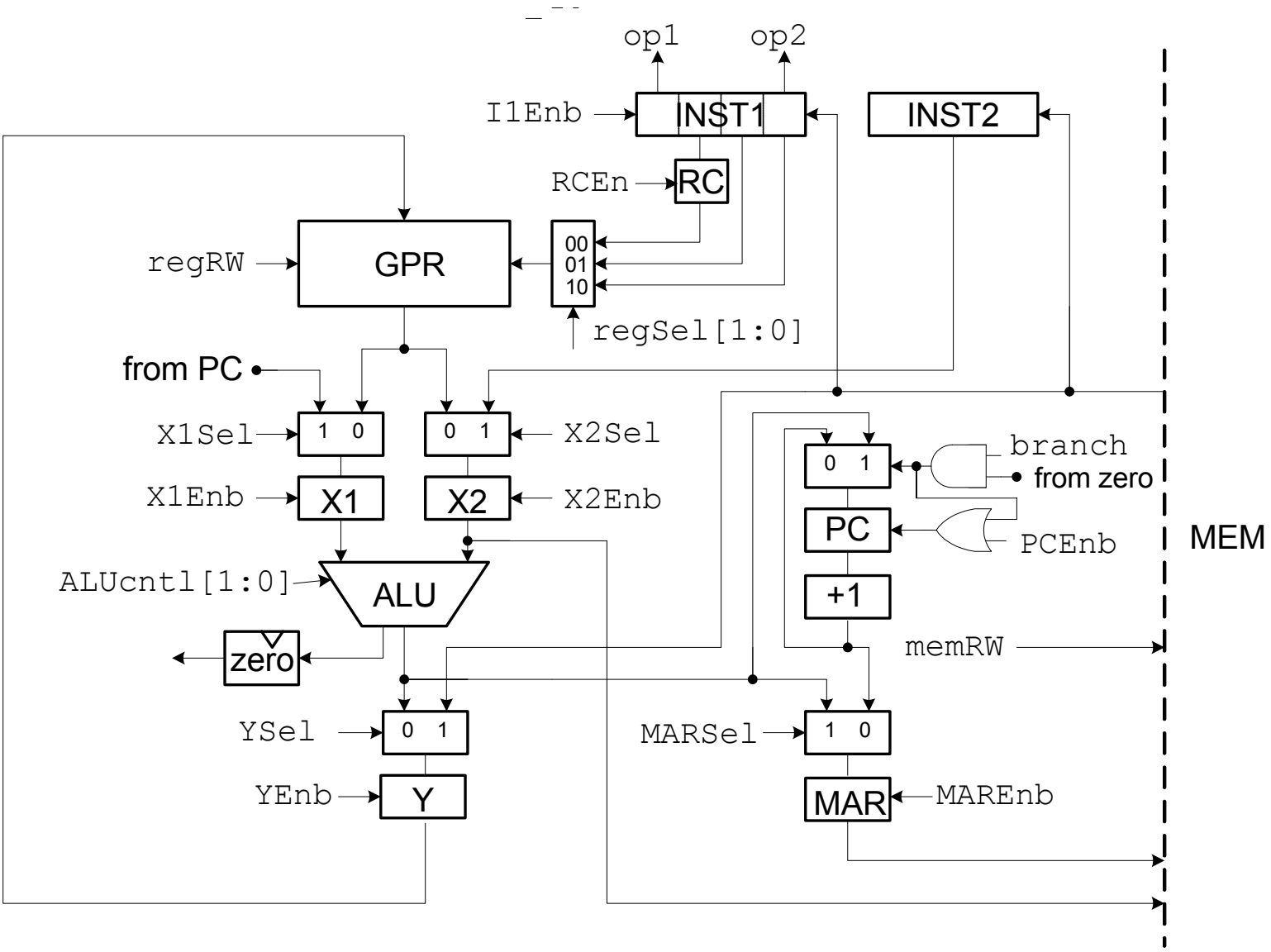
9. [15pts] The simple CPU core presented in class defined seven instructions; the eighth was reserved for future use. We have decided to let the new instruction be "add to memory (`atm`)", described as follows:

`atm rc,ra,offset` memory[ra+offset]←memory[ra+offset]+rc

It uses the "o-format" (as does the ldb, stb, and beq instructions):



a) Based on the data-path presented in class (reproduced below), write the RTL description for execution of the new `atm` instruction. Try to minimize the number of cycles needed.



b) The new instruction is not strictly necessary, because the same operation could be achieved using a sequence of the original 7 instructions. How many cycles would that take?