MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.111 Introductory Digital Systems Laboratory
Fall 2007

Midterm Exam: October 31, 2007

Name SOLUTIONS

This is an open book exam. Calculators can be used (but I don't think you'll need one).

Please write your answers legibly in the spaces provided. Please use the backs of the pages for scratch work and then make neat copies of your final answers in the space provided. It's hard to give credit if the answer is illegible!

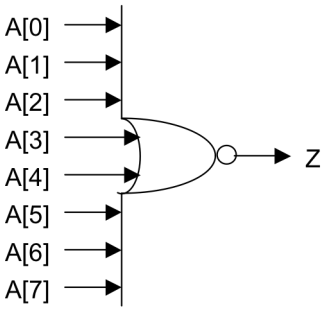
Problem	Score
#1	
#2	
#3	
Total	

Problem 1. (30 Points)

For each of the parts below write one or more statements of Verilog that implement the functionality shown in the schematic. Your Verilog just has to produce the same values for its outputs – it doesn't have to replicate the schematic logic gate-for-gate. Be sure to include the appropriate declarations for any wires or regs used in your code, and please follow our convention for the appropriate use of blocking and non-blocking assignments.

(A) (4 points) A circuit that tests an 8-bit value to see if it's zero.

Schematic:



Verilog:

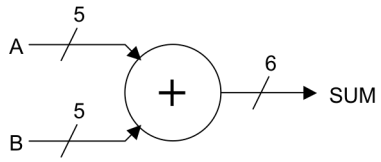
```
wire [7:0] A;
wire Z;

assign Z = (A == 0);

// alternatively…
assign Z = ~|A;
```

(B) (4 points) A circuit to compute the 6-bit sum of two 5-bit operands.

Schematic:



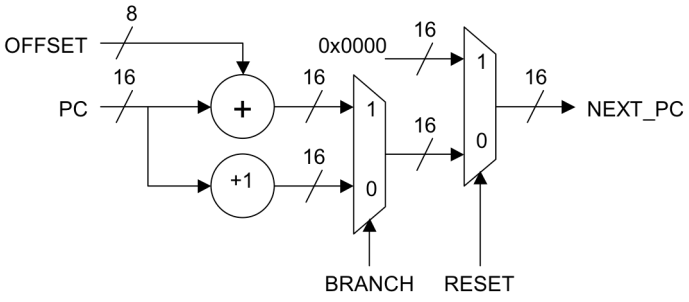
Verilog:

```
wire [4:0] A,B;
wire [5:0] SUM;

assign SUM = A + B;
```

(C) (6 points) A circuit to compute the next value for a 16-bit program counter. RESET and BRANCH are 1-bit signals, PC and NEXT_PC are 16-bit signals, and OFFSET is an 8-bit signal in 2's complement format (i.e., it can be either positive or negative).

Schematic:



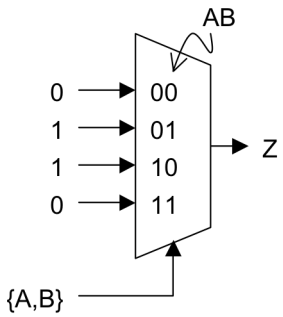
Verilog:

```
wire [15:0] PC, NEXT_PC;
wire [7:0] OFFSET;
wire BRANCH, RESET;

// priority logic
// don't forget to sign extend offset
assign NEXT_PC = RESET ? 16'h0000 :
                 BRANCH ? PC + {{8{OFFSET[7]}},
                                OFFSET[7:0]} :
                 PC + 1;
```

(D) (6 points) A logic function implemented as a multiplexer-based lookup table.

Schematic:



Verilog:

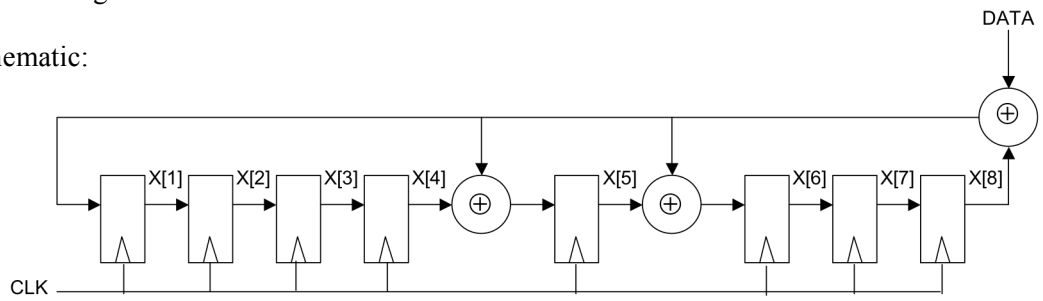
```
wire A,B,Z;

assign Z = A ^ B;  // an XOR!

// alternatively…
wire A,B;
reg Z;
always @ (A or B)
  case ({A,B})
  2'b00: Z = 0;
  2'b01: Z = 1;
  2'b10: Z = 1;
  2'b11: Z = 0;
  default: Z = 1'bx;
  endcase
```

(E) (10 points) A circuit to compute an 8-bit CRC on a serial bit stream using XOR gates (⊕) and 1-bit registers.

Schematic:



Verilog:

```
wire DATA,TMP,CLK;
reg [8:1] X;

assign TEMP = DATA ^ X[8];

always @ (posedge CLK)
  X <= {X[7:6],X[5]^TEMP,X[4]^TEMP,X[3:1],TEMP};
```

Problem 2. (40 Points)

You are an engineer working for NASA. They want you to design a FSM that will test their newest rover Fido on the MIT campus. NASA wirelessly transmits the travel plans to Fido, and then Fido moves according to that information.

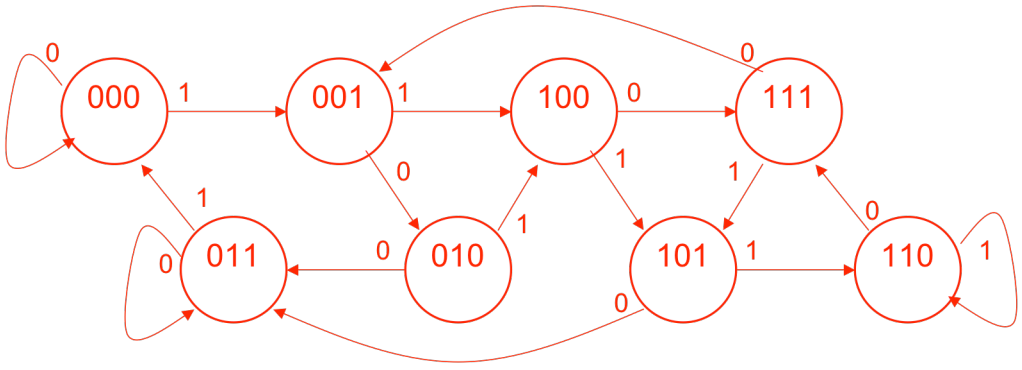
To design your FSM, you first select the following locations around the MIT campus and assign each location with a state in 3-bit binary representation: Killian[**000**], Kresge[**001**], Z-Center[**010**], Syd-Pac[**011**], Student Center[**100**], Building 34[**101**], 6.111 Lab[**110**], and the Stata Center[**111**].

To simplify your test, you inform NASA to send Fido's FSM a binary sequence for travel plans (e.g.'1-0-0-0-1' to cause Fido to move five times). In other words, Fido receives either '0' or '1' for each move and travels to the next destination as specified below. Fido starts off at Killian Court for each test run, and your FSM should output Fido's current location.

Killian [000]:	If 0, stay at Killian.	If 1, go to Kresge.
Kresge [001]:	If 0, go to Z-Center.	If 1, go to Student Center.
Z-Center [010]:	If 0, go to Syd-Pac.	If 1, go to Student Center.
Syd-Pac [011]:	If 0, stay at Syd-Pac.	If 1, go to Killian.
Student Center [100]:	If 0, go to Stata Center.	If 1, go to Building 34.
Building 34 [101]:	If 0, go to Syd-Pac.	If 1, go to 6.111 Lab.
6.111 Lab [110]:	If 0, go to Stata Center.	If 1, stay at 6.111 Lab.
Stata Center [111]:	If 0, go to Kresge.	If 1, go to Building 34.

(A) (14 points) Draw the state transition diagram for this FSM. Please use the back of an exam page for scratch space and make a neat copy of your final diagram below.

Draw state transition diagram.



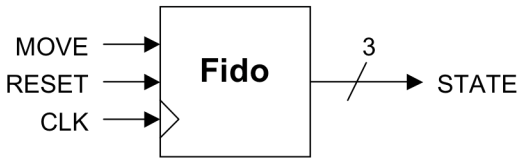
(B) (3 points) If Fido is forever given a sequence of ones (i.e. 11111…), where will it eventually end up?

Final location: 6.111 Lab

(C) (3 points) If Fido is forever given a sequence of 01s (i.e. 010101…), which location(s) will it never visit?

Location(s) never visited: 6.111 Lab

(D) (20 points) Design a module in Verilog for this FSM – the following schematic shows the appropriate inputs and outputs. Please use the back of an exam page as scratch space and make neat copy of the final code here.



Show your Verilog code.

```
module Fido(CLK,RESET,MOVE,STATE);
  input CLK,RESET,MOVE;
  output reg [2:0] STATE;

  always @ (posedge CLK) begin
    if (RESET) STATE <= 3'b000;
    else case (STATE)
      3'b000: STATE <= MOVE ? 3'b001 : 3'b000;
      3'b001: STATE <= MOVE ? 3'b100 : 3'b010;
      3'b010: STATE <= MOVE ? 3'b100 : 3'b011;
      3'b011: STATE <= MOVE ? 3'b000 : 3'b011;
      3'b100: STATE <= MOVE ? 3'b101 : 3'b111;
      3'b101: STATE <= MOVE ? 3'b110 : 3'b011;
      3'b110: STATE <= MOVE ? 3'b110 : 3'b111;
      3'b111: STATE <= MOVE ? 3'b101 : 3'b001;
      default: STATE <= 3'b000;
    endcase
  end
endmodule
```

Problem 3. (30 Points)

Consider the following Verilog module that uses Euclid's algorithm to iteratively compute the greatest common divisor of two 16-bit unsigned integer values Ain and Bin where Ain ≥ Bin.

```
module gcd(clk,start,Ain,Bin,answer,done);
  input clk,start;
  input [15:0] Ain,Bin;
  output reg [15:0] answer;
  output reg done;

  reg [15:0] a,b;
  always @ (posedge clk) begin
    if (start) begin a <= Ain; b <= Bin; done <= 0; end
    else if (b == 0) begin answer <= a; done <= 1; end
    else if (a > b) a <= a - b;
    else b <= b - a;
  end
endmodule
```

Please neatly complete the timing diagram below as the module computes the gcd of 21 and 15. Use "???" to indicate values that cannot be determined from the information given.