Name ______________________________

Digital Design using Verilog

EE 4702-1

Practice Midterm Examination

31 March 2000   13:40–?

Problem 1	________	(20 pts)
Problem 2	________	(20 pts)
Problem 3	________	(20 pts)
Problem 4	________	(40 pts)
Exam Total	________	(100 pts)

Alias ______________________________

Good Luck!

Problem 1:

(*a*) Write a Verilog *behavioral* description of a four-bit adder module. The adder should have three inputs, `a`, `b`, and `cin`, and two outputs, `sum` and `cout`. Ports `cin` and `cout` are one bit, the other ports are four bits each. (10 pts)

(*b*) Write a Verilog explicit structural description of an eight-bit adder that uses two of the four-bit adders above. (That is, instantiate the modules designed above.) (10 pts)

Problem 2: The code fragment below implements the clock in the testbenches for homeworks 2 and 3 (and the solution to 4).

```
// Clock.
always
  begin:CLOCK
     wait( start === 1 && done === 0 );
     forever # (`timeunit * 0.5/s1.freq ) clk <= ~ clk;
  end
```

(*a*) Symbol ‘`timeunit` was set to the number of simulator time units per second. What kind of Verilog thing is ‘`timeunit` (register, integer, etc.) and how was it defined? (5 pts)

(*b*) Symbol `s1.freq` is the frequency of the clock provided to the tachometer. What kind of Verilog thing is it and how did it get its value? (5 pts)

(*c*) Symbol `start` is an input to the testbench module, it is set to `1` when the testbench is to start. Symbol `done` is an output which the testbench sets to one tests are completed. What would happen if `start === 1` were removed from the code fragment above? (5 pts)

(*d*) What would happen if `done === 0` were removed from the code fragment above? (5 pts)

Problem 3: An *accumulator* has three inputs, amt, reset, and clk, and an output, sum. The accumulator has an internal 32-bit register which is updated as follows: On a positive edge of clk it adds amt, a 32-bit integer, to the register; on the negative edge of clk it places the new sum on its outputs (until the next negative edge). Whenever reset is high the register is set to zero and the output changes immediately. Write a Verilog behavioral description of this module. (20 pts)

Problem 4: Answer each question below.

(*a*) The code below starts executing a $t = 0$. Show all changes in `a`, include the time of the change and the new value. (5 pts)

```
integer a;

initial begin
   a = 1;
   #1;
   a = 2;
   #1;
   a = 3;
   #1;
   a <= 4;
   #1;
   a <= 5;
   #1;
   #3 a = a+1;
   #1;
   a = #3 a+1;
   #1;
   a <= #3 a+1;
   #1;
   a = a+1;
end
```

(*b*) The programmer expected execution to exit the loop below when either i was 1000 or a[i] == c, but that's not what happened. What goes wrong and how can it be fixed? The loop must be exited using a disable statement. (5 pts)

```
 integer i, c;
 integer a[0:999];

 // ...

 i = 0;

 while( i < 1000 ) begin:LOOP

    if( a[i] == c ) disable LOOP;

    i = i + 1;

end
```

(*c*) Describe three uses for Verilog behavioral code. (5 pts)

(*d*) What is the difference between the following two declarations? (5 pts)

```
wire [7:0] w1;
wire [0:7] w2;
```

(*e*) What do the values `x` and `z` signify? (5 pts) (*f*) What is `10'had`? (5 pts)

(*g*) Name two features of primitives that are not available for modules. (5 pts)

(*h*) What is the difference between `case`, `casex`, and `casez`? (5 pts)