EEE4084F: Digital Systems

10 May 2012

Quiz 4

SOLUTIONS

Question 1

(a) The main thing is to see what the problem and solution is described clearly and logically.
[4 marks]

(b) There are lots of right answers for this question; the answer generally need to make sense. For example, a design question could be:
"What is the maximum allowed latency?"
"Are hand-shaking connections required for the inputs and outputs?"
"Should the inputs and output be latched?"
"Should these be control lines, e.g. a reset line for the module?"
"What should happen to the output if two or more inputs of the same value are received?"
"The problem specifies that N input buts should be catered for; is this a genetic/parameter? The minimum N is probably 1, but what is the maximum N that the module should work for? "
4 marks

(c) Criteria should make sense. Examples are:
"The output value must be updated within 0.8 * (1/M) s after the input bits have changed."
"The output value must toggle if the same input value keeps getting fed in."
"If increasing numbers greater than 0 are fed into the module, the output bit must remain at 1."
4 marks

Question 2

(a) Verilog code sample solution with top level block diagram represented as module *toplevel1* :
10 marks

```
// file toplevel1.v
module Dev1 ( output y, input A, input B, input x );
 if (x == A) y <= B; else y <= ~B;
endmodule;

module Dev2 ( output x, input A, input clk);
 reg oldA = 0;
 always @ (posedge clk) begin
   oldA = A;
 end
 x = oldA;
endmodule;

module toplevel1 ( output Xout, input A, input B, input clk, input En );
 // create and wires and regs that may be needed
 wire x,y;
```

```
 Dev1 u1(y,A,B,x);
 Dev2 u2(x,A,clk);
 and(Xout,y,en);
end module;
```

(b) Simulation would be an obvious approach to testing. This would be done by sending sample inputs to the delta modulation module to see if the outputs are expected. The specific approach depends on the tool used. For example, with Xilinx ISIM, a Verilog or VHDL test bench would be implemented: This test bench would include code that exercises the inputs and either prints out results to the console (or text file) using the display command, or it can be graphed. Sample code for a Verilog test bench is illustrated below:

4 marks

```
// file toplevel1_tb.v
include " toplevel1.v"
module toplevel1_tb();
// create inputs as regs and outputs as wires
reg reset, A, B, clk, en;
reg Xout;

// Initialize all variables
initial begin
  // display column headings
  $display ("time\t reset en A B Xout");
  // Whenever one of these registers/wires changes print all of them:
  $monitor   ("%g\t %b     %b  %b %b %b", $time, reset, en, A, B, Xout);
  reset  = 0       // initial value of reset
  enable = 0;      // initial value of enable
  #10 reset = 1;  // Assert the reset
  #10 A = 0; B = 0; en = 0; // initialize the inputs for the module
  #10 reset = 0;  // De-assert the reset
  #10 en = 1;      // set enable line high
  // exercise the A and B inputs...
  #10 B = 1;
  #10 A = 1;
  #10 B = 0;
  #10 en = 0;      // De-assert enable
  #20 $finish;     // Terminate simulation
end

// Clock generator
always begin
  #5 clock = ~clock; // toggle clock every 5 ticks
end

 // Connect test bench to instance of toplevel
 toplevel1 toplevel1U (Xout, A, B, Clk, En);
endmodule
```

The code is more just for illustrative purposes. A visual will count only 1 mark; it is more the description that was asked for).

Question 3

(a) The determining factor is ability for the computer to change hardware datapaths and control flows by software control. This change could be either a post-process / compile time or dynamically during runtime. 3 marks

(b) The terms are:

Programmable logic device (PLD) = a chip (i.e. the physical package) that contains the PLBs and their composite PLEs.

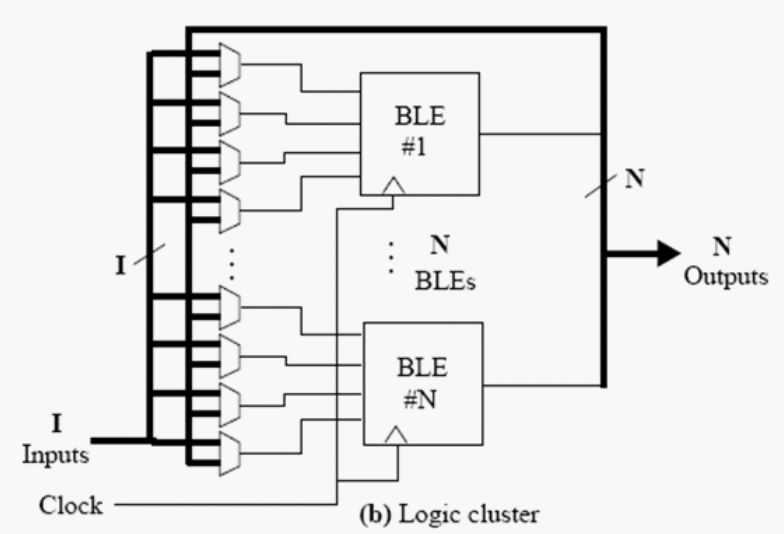
Programmable logic element (PLE) = the lowest level / smallest programmable component within the PLD (e.g., a LUT or multiplexor that decides which gates to connect up).

Programmable logic block (PLB) = collection (or clustr) of PLEs

4 marks (one mark each, one mark for presentation)

(c) Provide a sketch using multiplexors and flip flips that indicates how interconnects between PLBs within a FPGA are configured

5 marks



(d) The configuration architecture the circuitry used for programming the FPGA; typically implements a statemachine that toggles the streams a FPGA bit into the FPGA.

2 marks

Question 4

Student should discuss that the hybrid approach is about making tradeoffs between hardware- and software-centered designs, establishing a suitable balance between the factors of size, weight and power. Effectively, current technologies provide more flexibility for developers, allowing more hardware or more software to achieve the required SWAP needs.

5 marks