Digital Systems Implementation (ECE 4740)

Bob McLeod

E3-416 EITC

(204) 474-8886

N.B. You can contact me any time, best by email, bobmcleod247@gmail.com

Also you can drop by my lab. E3-416 anytime

(I am not in my office ever!)

Labs

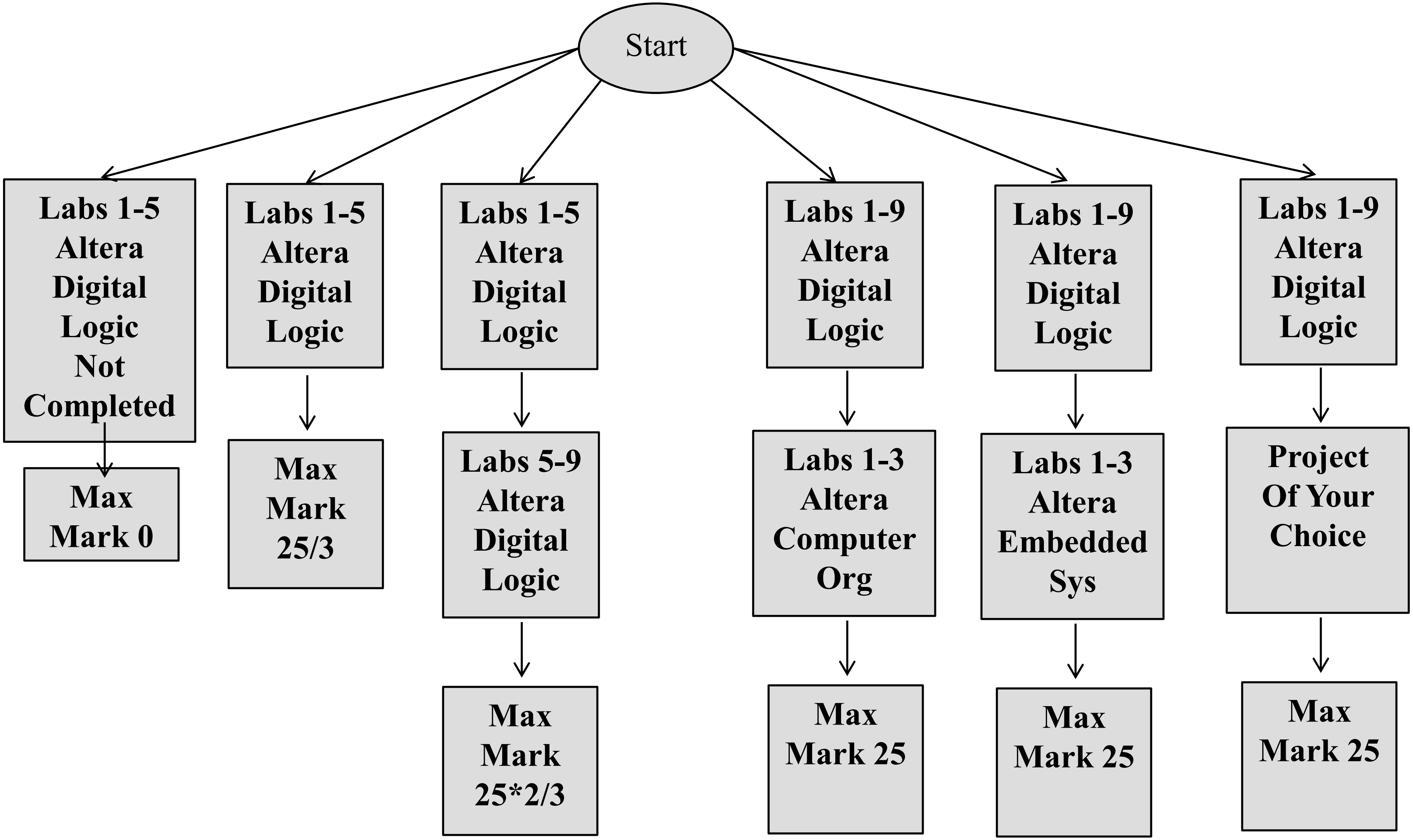
- Labs: Room E3-528
  - TA: TBD

Section	Dates
B01	Sept 11, Sept 25, Oct 09, Oct 23, Nov 06, Nov 20

- Laboratory max 25%
  - Make sure you understand the marking process and schedule.

Lab: remarks

- You are responsible to complete many labs from the Altera UP web site during the term.
  - ➢ These can be the first (9) Altera dig logic labs.
    - ❑ Roughly 3 per lab period to complete them by Nov. 1 (That will be for a total of 2/3 of the lab marks)
  - ➢ Then during November you can complete the first 3 from Altera's comp. org. or embedded sys. stream
    - ❑ Or complete one project of your choosing
    - ❑ (This will comprise the remaining 1/3 of the marks)
  - ➢ If you get 5 labs done, max mark is 1/3 of the available lab marks.
    - ❑ An incomplete in the course if these 5 are not done.



Start
Labs 1-5 Altera Digital Logic Not Completed
Max Mark 0
Labs 1-5 Altera Digital Logic
Max Mark 25/3
Labs 1-5 Altera Digital Logic
Labs 5-9 Altera Digital Logic
Max Mark 25*2/3
Labs 1-9 Altera Digital Logic
Labs 1-3 Altera Computer Org
Max Mark 25
Labs 1-9 Altera Digital Logic
Labs 1-3 Altera Embedded Sys
Max Mark 25
Labs 1-9 Altera Digital Logic
Project Of Your Choice
Max Mark 25

- http://www.altera.com/education/univ/materials/digital_logic/labs/unv-labs.html
  - Digital Logic Labs
- http://www.altera.com/education/univ/materials/comp_org/labs/unv-labs.html
  - Computer Organization Labs
- http://www.altera.com/education/univ/materials/embedded_systems/labs/unv-labs.html
  - Embedded Systems Labs

Signoff Schedule 30 Mark 20 Mark

- Digital Logic Labs
  - Lab 1 - Switches, Lights, and Multiplexers
  - Lab 2 - Numbers and Displays Target: Sept. 11
  - Lab 3 - Latches, Flip-flops, and Registers Target: Sept. 11
  - Lab 4 - Counters Target: Sept. 25
  - Lab 5 - Timers and Real-Time Clock
  - Lab 6 - Adders, Subtractors, and Multipliers Target: Sept. 25 Oct. 09
  - Lab 7 - Finite State Machines
  - Lab 8 - Memory Blocks Target Nov. 6
  - Lab 9 - A Simple Processor Target: Oct. 09 Target Nov. 20
- Computer Organization Labs (or 3 Labs Embedded Sys or Project)
  - Lab 1 - Using Altera's Nios II Processor, Lab 2 - Subroutines and Stacks, Lab 3 - Using Logic Instructions Target: Nov. 20

Levels (done and in on time)

- Laboratory Levels
  - Level 1 (complete DL labs 1-5) 1/3*25 marks max
  - Level 2 (complete DL labs 1-9) 2/3*25 marks max
  - Level 3 (complete DL labs + CO/ES labs 1-3 or Project) 25 marks max

- Assignment levels:
  - Level 1 < ½ ICAs: 0 marks
    - Level 2 ½ <= ICAs <= all 2.5: marks max
  - Level 1 < ½ Assignments: 0 marks
    - Level 2 ½ <= Assignments <= all: 2.5 marks max

Preliminaries

- Get tools running at your home.
  - Immediately (i.e. by this weekend)
- Specifically:
  - Quartus II (Altera Design Tools)
  - Modelsim Simulation Software
- Official Language:
  - Verilog

Caveats

- This lab schedule is very aggressive, perhaps too much so.
- But, it is where we are starting from.
- In any event, following the more aggressive signoff schedule will facilitate completion in a timely manner.
- Downloading the solutions benefit you nil.
- There are DE2 boards you can borrow as well.

Most Important

- If you do the labs as instructed on the Altera University web site.
  - It will take forever
  - The logic labs were developed to teach logic, courses you have already taken.
- Thus
  - Use higher level constructs and behavioral code

For Example

- Lab 1 Part II you are asked to build a 2x1 mux and use it in the project of a 8 bit wide 2-to-1 mux. That is a lot of work and basically illustrates instances of modules reused.
- Instead you can directly implement the same function at a much higher level.

```
module eightBitMux (SW, LEDR, LEDG);
input [17:0]SW;
output [17:0]LEDR;
output [7:0]LEDG;
assign LEDR = SW;
assign LEDG[7:0] = (SW[17] == 0 )?SW[7:0]:SW[15:8] ;
endmodule
```

- Same applies throughout.

Eg 2. Lab 2 part 3

```
module part3 (SW, LEDG, LEDR);
  input [17:0] SW;
  output [8:0] LEDR, LEDG;

  assign LEDR[8:0] = SW[8:0];

  wire c1, c2, c3;

  fulladder A0 (SW[0], SW[4], SW[8], LEDG[0], c1);
  fulladder A1 (SW[1], SW[5], c1, LEDG[1], c2);
  fulladder A2 (SW[2], SW[6], c2, LEDG[2], c3);
  fulladder A3 (SW[3], SW[7], c3, LEDG[3], LEDG[4]);
endmodule

module fulladder (a, b, ci, s, co);
  input a, b, ci;
  output co, s;

  wire d;

  assign d = a ^ b;
  assign s = d ^ ci;
  assign co = (b & ~d) | (d & ci);
endmodule
```

```
module part3 (SW, LEDR, LEDG);
  input [8:0] SW;
  output [8:0] LEDR;
  output [4:0] LEDG;
  assign LEDR=SW;
  adder4 A4 (SW[8],SW[7:4],SW[3:0],LEDG[4:0]);

endmodule


module adder4 (carryin,X,Y,S);
  input carryin;
  input [3:0] X,Y;
  output reg [4:0] S;

  always @(X,Y,carryin)
  S=X+Y+carryin;

endmodule
```