ECE 550 Fall 2015

Homework 2

Due: 11:59PM ~~Sept 25,~~ 2015 October 2nd

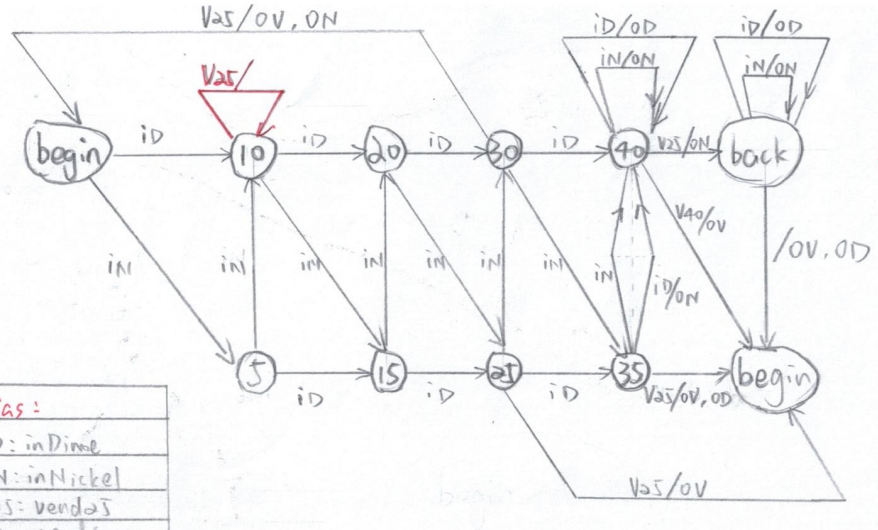
Demo by: 11:59PM ~~October 2nd~~, 2015 October 9th

For this homework, you will be answering two pencil and paper questions, as well as writing VHDL. You should submit your answers to the pencil and paper questions, along with your VHDL code in a .tar, .tar.gz, or .zip format (no other formats will be accepted) to Sakai before the deadline. Your answers to the pencil and paper questions **must** be in .pdf format—no Word documents will be accepted. For drawings, you are encouraged to draw in computerized tool, but may draw by hand and scan the drawings into a pdf format, as long as they are clear and easily readable.

Within one week of the submission deadline, you must demo your VHDL code to a TA, who will ask each group member to explain various aspects of how it works. **All** group members are responsible for understanding the entire submissions.

Shengxin Qian

Xinquan Liu



Φ: The input which caused self-loop with no output will not ~~be shown~~ be drawn.
e.x. the real self-loop in the diagram will be omitted

Alias:

input:	iD: inDime
	iN: inNickel
	V25: vend25
	V40: vend40
output:	0V: out Vend
	0D: out Dime
	0N: Out Nickel

The in/output format on 黑板
input / output

state	alias	current state	input (iD/iN/V25/V40) 0001	0010	0100	1000	0000
begin	0	0000	0/000	0/000	1/000	2/000	0/000
5	1	0001	1/000	1/000	2/000	3/000	1/000
10	2	0010	2/000	2/000	3/000	4/000	2/000
15	3	0011	3/000	3/000	4/000	5/000	3/000
20	4	0100	4/000	4/000	5/000	6/000	4/000
25	5	0101	5/000	0/100	6/000	7/000	5/000
30	6	0110	6/000	0/101	7/000	8/000	6/000
35	7	0111	7/000	0/110	8/000	8/001	7/000
40	8	1000	0/100	9/001	8/001	9/010	8/000
back	9	1001	9/000	9/000	9/001	9/010	0/110

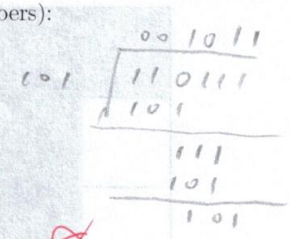
format in bank: next state / output

Output (0V/0D/0N)

Question 2:

Perform the binary division of 110111 divided by 101 (both are unsigned numbers). For each step (states are numbered by which bit of the divisor is examined) , show the contents of the remainder register and the answer register (both of which are 6 bit unsigned numbers):

State	Remainder	Answer
Start	000000	000000
5	110111	000000
4	110111	000000
3	001111	000001
2	001111	000010
1	000101	000101
0	000010	001011



Answer * 101 + Remainder = 110111

Question 3:

For this question, you will be writing VHDL for a "baby video card". The DE2 boards has pins which let you send digital color signals to a VGA connection (there is some digital to analog conversion involved, but you do not need to worry about that).

We have provided you with a quartus archive file (`vgacontroller_dist.qar`) which gives you the pin assignments, a clock at the proper frequency, and an active high reset signal (meaning you get a '1' for reset—by contrast the signals coming from the buttons on the board are active low, meaning they send a '0' when its pressed).

You should not need to edit any file other than `myvga.vhdl`, where you will fill in the `myvga` entity (currently it just has placeholders which do nothing useful).

Before we go into the details of the provided VHDL, it is useful to give you a description of how video cards work, and what needs to be done to properly control the VGA signals.

Video cards store pixel data as numeric values in memory (located on the video card). The pixels make up a 2D array of dots, with (0,0) in the upper-left corner of the screen. How many pixels there are depends on the screen resolution. You will be working with a resolution of 640x480 (so 640 pixels in the x direction (columns), and 480 in the y direction (rows)). Video memory is indexed linearly, the rows laid out one after the other—the index into the video memory for a pixel at (x, y) is $y * w + x$, where w is the width of the screen (in pixels). The drawing below illustrates for a resolution of 8x6: