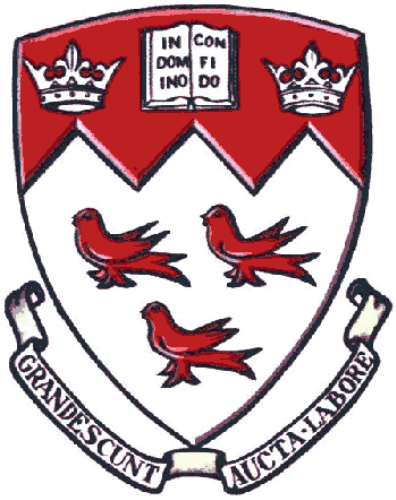
Lab Five



Enigma Machine, Enigma User Interface

Enigma Machine

Prepared by
Team 22

Ossama Samir Ahmed	**260549558**
Georges Assouad	**260567730**

Course Number: ECSE 323

Course Name: Digital System Design

Date: 2016/04/15

Table of Contents

1 Description of the enigma machine's features

The enigma machine is a cipher machine dating back to the early 1920s used to protect private communications; it was widely used during World War II. In this lab, we created a digital version of the enigma machine using several computer programs. The machine is implemented on the FPGA Altera board. The machine is able to cipher a message and only the person with the key will be able to read it. Based on the complexity of this machine you can be sure that your secret will be safe with us. The message coming in the machine will go through as much as nine permutations. Every permutation is different than the other, which makes it nearly impossible to decipher… unless your name is Alan Turin.

2 Description of the entire system

Figure 1. Simplified block diagram of the enigma machine.

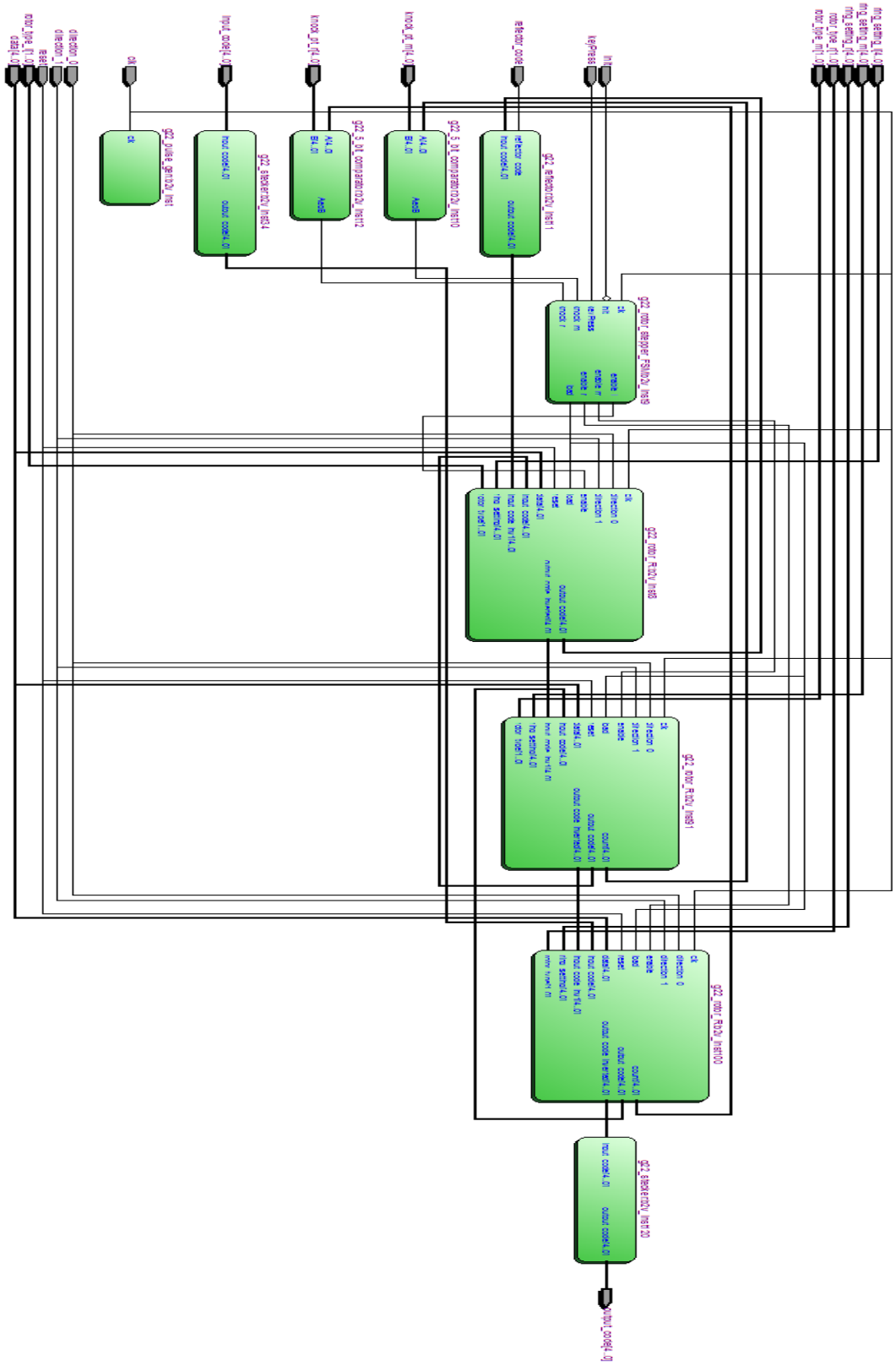


Figure 2. Block diagram of the enigma machine

The enigma machine was constructed using all the previous labs as components. Some new components were also created in this lab. In figure 1. you can see the simplified block diagram of the whole enigma machine with its inputs and outputs. The enigma machine takes as inputs: “clk”, “init”, “KeyPress”, “reflector_code”, “direction_0”, “direction_1” and “reset” which are of type std_logic; “input_code”, “knock_pt_m”, “knock_point_r”, “ring_setting_l”, “ring_setting_m”, “ring_setting_r” which are 5 bit long std_logic_vector ; “rotor_type_r”, “rotor_type_m”, “rotor_type_l” which are 2 bit long std_logic_vector. The output of the machine is simply “output_code” which is of type std_logic_vector and size 5, it is the scrambled/permutated/shifted letter (“input_code”). To better understand how the enigma machine will cipher the message, we take a look at figure 3. The machine takes as input a sequence of letters that will go through several stages of permutation and shifting to generate the scrambled message. We will study every component of this structure.

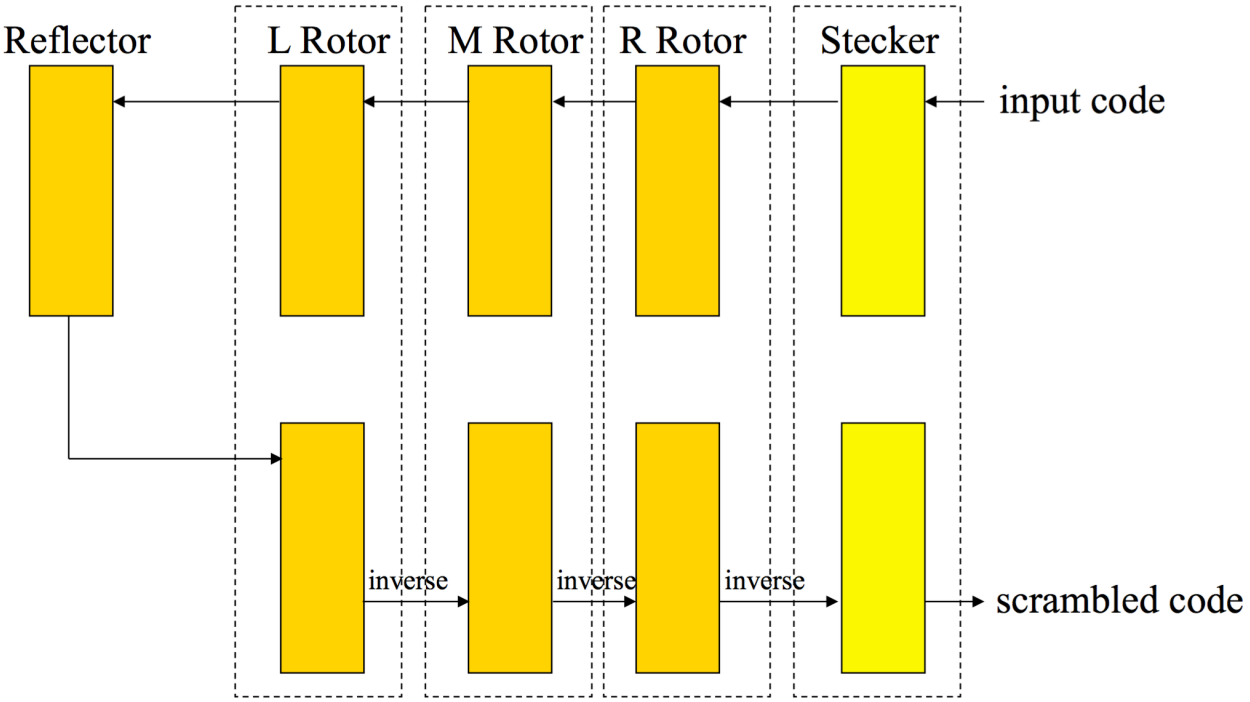


Figure 3. Overall structure of the enigma machine.

The stecker circuit is a simplified version of a fixed rotor and is able to swap codes for a small number of code pairs. It will be traversed twice, once in the right order and another time in the inverse order. This swapping is hardcoded and we chose which pair of codes should be swapped. In the simulation of the stecker circuit, figure 4., you can see several input codes (letters) being swapped/scrambled.

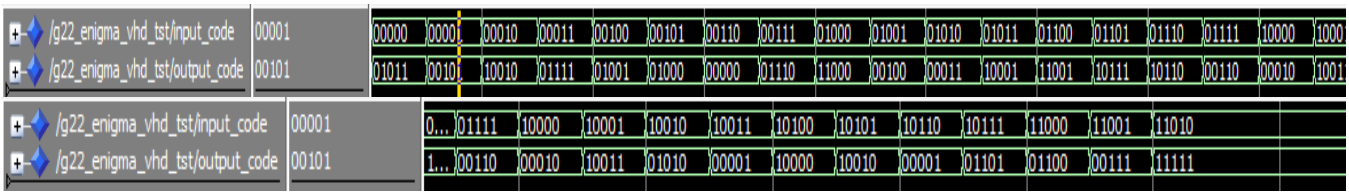


Figure 4. Simulation of the stecker circuit.

The reflector circuit simply takes the input code and gives the image of the sequence. It will be traversed once and its output will go the last rotor it has passed by. The message will go through the same permutations but inversely.

Furthermore, we will be talking about the rotors, which are the main encryptions of the enigma machine. We have three different rotors in the enigma machine. Each rotor will perform a different permutation. As seen in figure 4. this component takes as input “direction_0”, “direction_1”, “enable”, “load”, “reset”, “clk” (std_logic) and “data”, “input_code”, “input_code_inv”, “ring_setting” (std_logic_vector of size 5), “rotor_type” (std_logic_vector of size 2). It gives as output “count”, “output_code” and “output_code_inverted” (std_logic_vector of size 5).

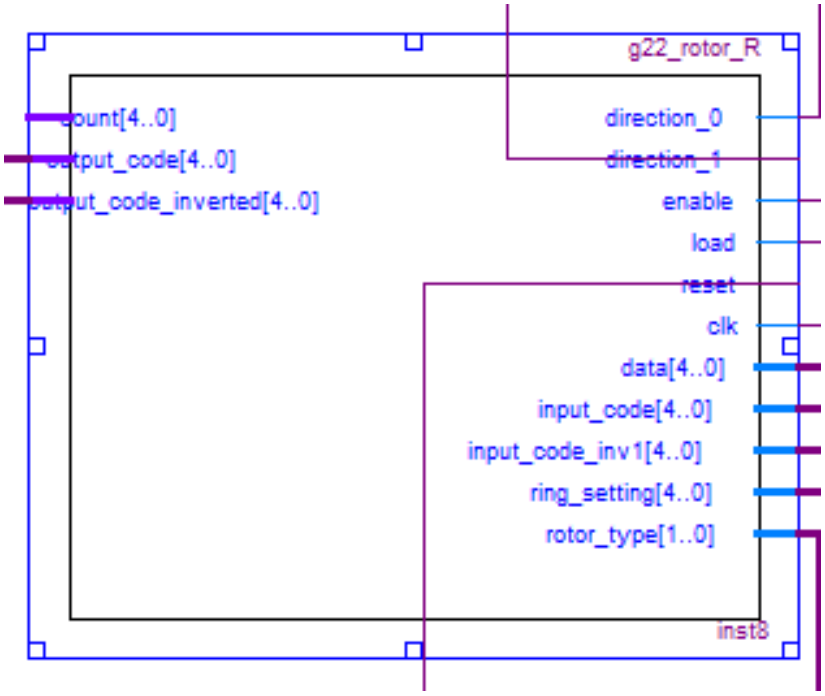


Figure 4. Block diagram of the rotor.

3 Description of the user interface

To implement the enigma machine on the altera board, we created a user interface since there are not enough options on the board to generate all the functions. The user interface, the block diagram on the left of figure 5., takes 4 inputs. The inputs are “clk”, “keyPress”, “reset” of type std_logic and “input” of type std_logic_vector and size 5 bits. The outputs of the user interface represent the inputs of the enigma machine already mentioned in the previous section.

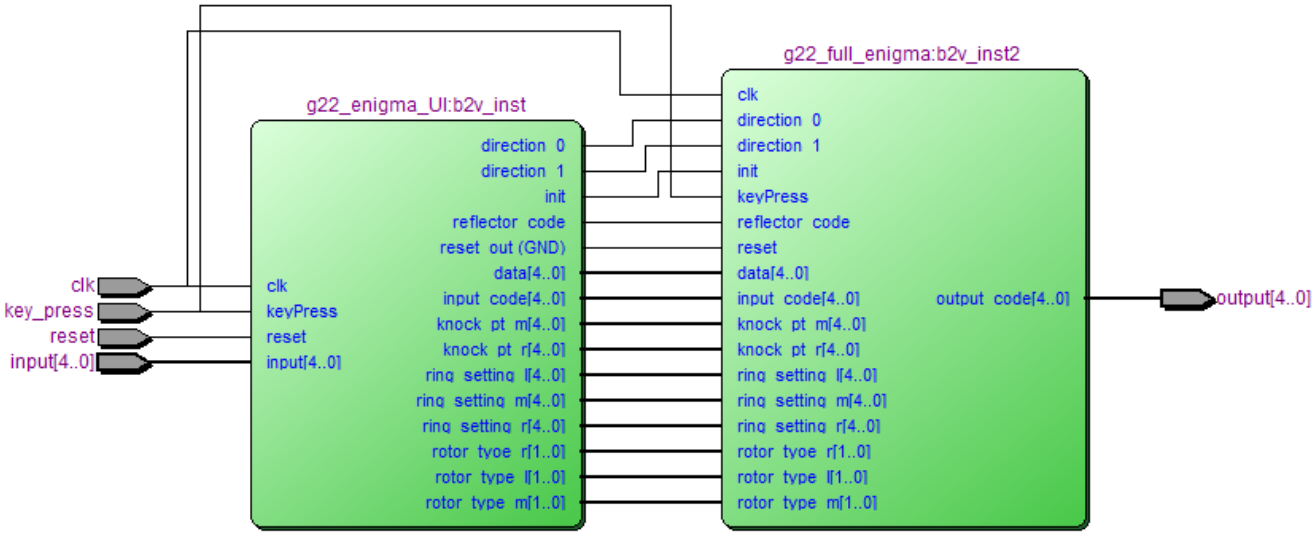


Figure 5. Block diagram of the user interface and the enigma machine.

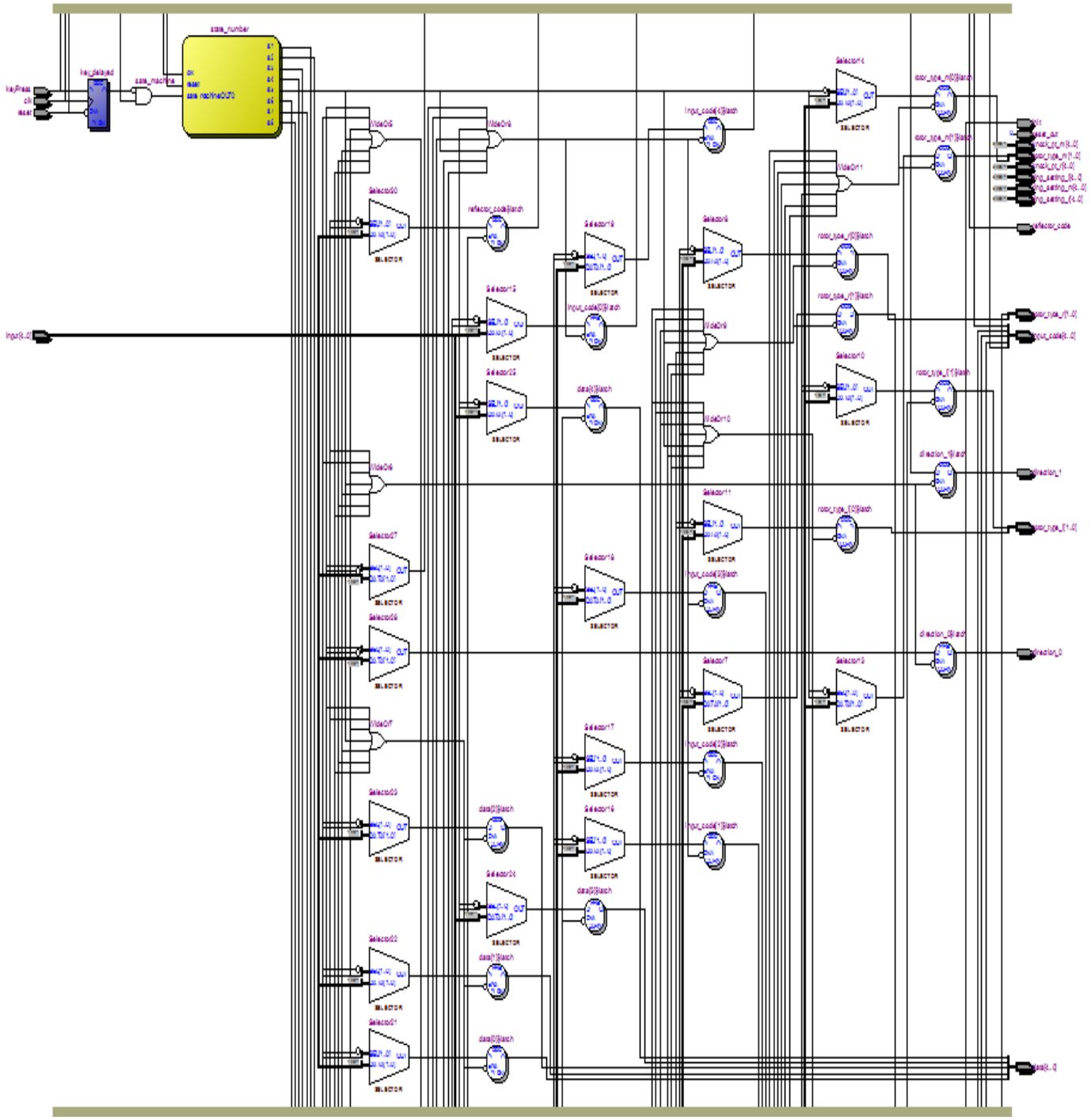


Figure 6. Symbol diagram of the user interface.

4 Testing of the enigma machine

Before implementing the code of the enigma machine on the Altera board, we created a test bench and implemented it on model sim. To see that our machine works correctly we entered a word like “DSD” which converts to “00011”, “10010” and “00011” in binary. Unfortunately, we did not get the right output and we encountered problems getting the decoded letters. We experienced a shift of 2 letters for every decoded letter but we could not figure out the source of the problem due to the time limitations. We suspected it is in the barrel shifter but we could not find it.

5 Conclusion

We had several issues during our design process, which were not expected. We had to go back to previous labs to debug some of the codes that we thought were working perfectly like the barrel shifter. We ended losing a lot of time debugging our system before it finally worked. A possible enhancement to our system is to fix the barrel shifter so that we obtain the wanted shift of letters.

Grade Sheet for Lab #5

Winter 2016.

Group Number: 22

Group Member Name: Georges Assaad — Student Number: 26056730

Group Member Name: Ossama Ahmed — Student Number: 26054958

Marks			TA Signatures
2	1.	VHDL Description of the reflector circuit	
2	2.	Simulation of the reflector circuit	
2	3.	VHDL description of the Stecker circuit	AU
2	4.	Simulation of the Stecker circuit	AW
2	5.	VHDL for the rotor	
2	6.	VHDL for the complete Enigma Machine	
1	7.	Simulation of the complete Enigma Machine	AW
1	8.	Demonstration of the Enigma Machine on the Altera board	AW

Each part should be demonstrated to one of the TAs who will then give a grade and sign the grade sheet. Grades for each part will be either 0, 1, or 2. A mark of 2 will be given if everything is done correctly. A grade of 1 will be given if there are significant problems, but an attempt was made. A grade of 0 will be given for parts that were not done at all, or for which there is no TA signature.