MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 23 02
F: +90 312 210 23 04
ee@metu.edu.tr
www.eee.metu.edu.tr

VERILOG TUTORIAL

SIMPLE ALARM CLOCK

In this tutorial, you are going to design a basic alarm clock by using Verilog HDL and implement it on FPGA. The system will have one top module including two submodules: the controller module and the seven segment display module. In order to implement alarm clock on the FPGA, leds, push buttons and seven segment display will be used. Details about these modules and desired specifications of the alarm clock are explained below.

a) Controller Module

The controller module should control all the functions of the alarm clock. Specifications of the controller module are defined as follows:

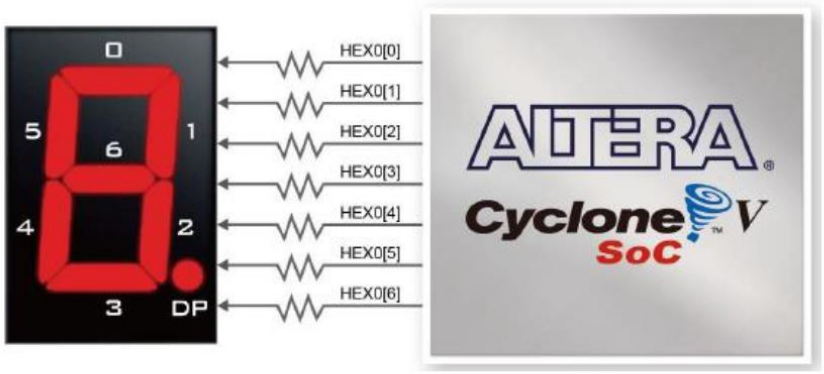
1) In the alarm clock, there will be three buttons: start/stop, snooze and set. In order to monitor timing (remaining second) we will use leds and seven segment display as shown in Figure 1.
2) When the system is powered up the seven segment display should show 0 and none of leds should be ON as shown in Figure 1.
3) The user first determine the time for the alarm clock. For this purpose, user should press set button; if the user press set button once then the time should be 1 and 1st led (Led9) should be ON, if set button is pressed four times then the time should be 2 and 1st to 4nd leds should be ON as shown in Figure 2.
4) The maximum set time of the alarm clock can be 9 so if the user continue pressing set button when the time is equal to 9 (9 leds are ON), it should be ignored by the program.
5) After specifying the time, the alarm clock should be started by pressing start button and timer should count down the value specified by the user (1-9). Also, leds should indicate the remaining time (if the time is 4 then 4 of the leds should be ON).

6) When, the time is equal to zero, the alarm clock should start ringing. Do not worry, we are not going to use any sound interface for ringing operation. Instead, all of the 9 leds should be ON and OFF continuously at 1 Hz that indicates the alarm is ringing as it is shown in Figure 3.
7) When the alarm is ringing, the user has 2 options: stop the clock or snooze the alarm for 5 seconds.
8) If the user presses start/stop button the alarm clock should be stopped (all of the leds should be OFF and the time should be 0 (the same case with when the system is powered up)). Therefore, note that start/stop button has dual functionality: starting the alarm clock or stopping the alarm clock.
9) If the user presses snooze button, extra 5 seconds should be added to the timer (timer should be equal to 5 and count down again) and leds should be adjusted accordingly (5 leds should be ON.).
10) If the user presses set button, then the same operation described in (part 3)) starts. Again, if the user presses the set button when the alarm is ringing, it **cannot** be snoozed.
11) Other additional specifications are; the user cannot stop the alarm clock after it is started counting down (ringing of the alarm clock should be waited) or cannot snooze the alarm clock when it is counting down.

Design and simulate the controller module and make sure that your module satisfies all of given specifications. Include both RTL schematics and simulation results in your report. Show the results to your lab assistant. You are free to use any clock frequency in your simulations. However, you should use the master clock on the FPGA and divide it in your controller module in order to achieve the desired frequency of the lights and seven segment display when you implement your code on the FPGA. Pin assignment for the clock inputs of the FPGA is given in appendix.

b) Seven Segment Display Module

The only function of seven segment display module is to get instantaneous time information from controller module and display it on the seven segment display (SSD) of the FPGA. Pin diagram for SSD is given below for your convenience.



Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_AE26	Seven Segment Digit 0[0]	3.3V
HEX0[1]	PIN_AE27	Seven Segment Digit 0[1]	3.3V
HEX0[2]	PIN_AE28	Seven Segment Digit 0[2]	3.3V
HEX0[3]	PIN_AG27	Seven Segment Digit 0[3]	3.3V
HEX0[4]	PIN_AF28	Seven Segment Digit 0[4]	3.3V
HEX0[5]	PIN_AG28	Seven Segment Digit 0[5]	3.3V
HEX0[6]	PIN_AH28	Seven Segment Digit 0[6]	3.3V

Design and simulate the seven segment display module that properly shows remaining time. Include both RTL schematics and simulation results in your report also show the results to your lab assistant.

c) Integration of the Modules

After you have designed both of the modules, you need to integrate them into a top module using hierarchical design. Finally, program the FPGA and show the proper operation of the alarm system to your lab assistant.

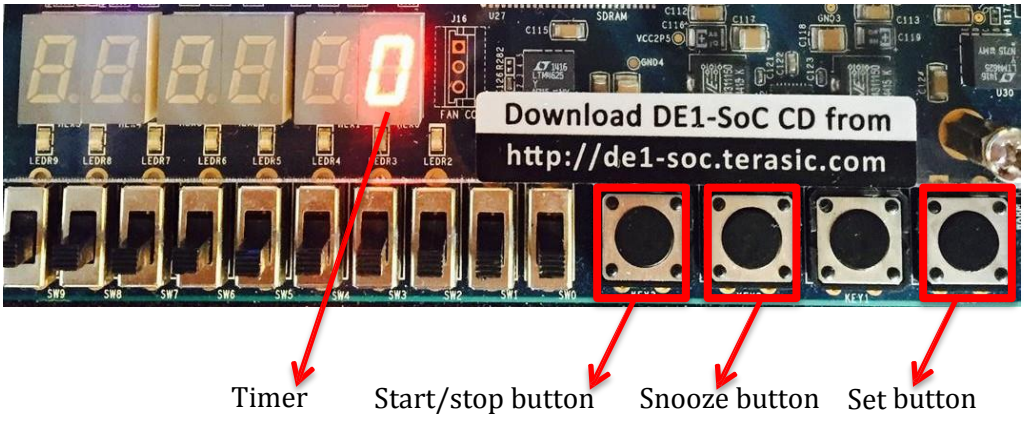
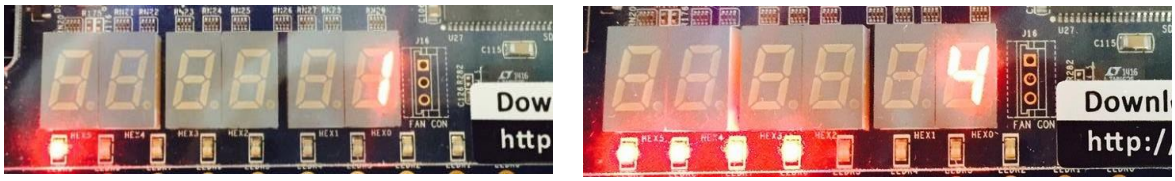


Figure 1. Screenshot of the game (after integration of the modules in part c.) when the power is up.



(a) (b)

Figure 2. Screenshot of the alarm clock when set button is pressed (a) once (b) 4 times.

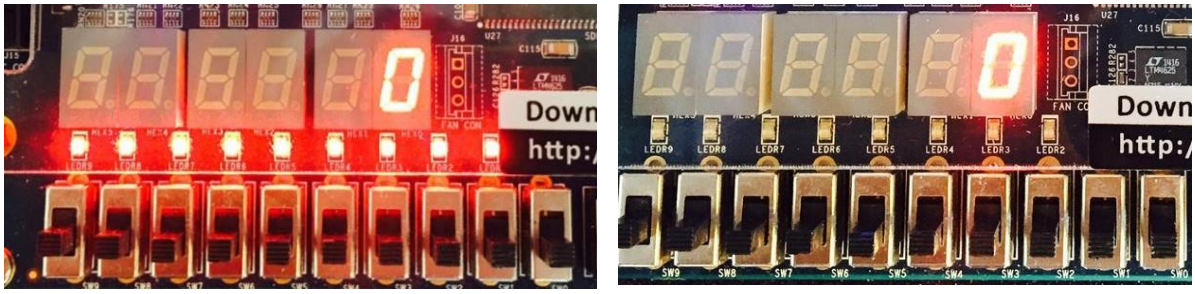


Figure 3. Screenshot of the alarm clock when it is ringing.

APPENDIX A

Table 3-5 Pin Assignments for Clock Inputs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
CLOCK_50	PIN_AF14	50 MHz clock input	3.3V
CLOCK2_50	PIN_AA16	50 MHz clock input	3.3V
CLOCK3_50	PIN_Y26	50 MHz clock input	3.3V

Table 3-6 Pin Assignments for Slide Switches

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB12	Slide Switch[0]	3.3V
SW[1]	PIN_AC12	Slide Switch[1]	3.3V
SW[2]	PIN_AF9	Slide Switch[2]	3.3V
SW[3]	PIN_AF10	Slide Switch[3]	3.3V
SW[4]	PIN_AD11	Slide Switch[4]	3.3V
SW[5]	PIN_AD12	Slide Switch[5]	3.3V
SW[6]	PIN_AE11	Slide Switch[6]	3.3V
SW[7]	PIN_AC9	Slide Switch[7]	3.3V
SW[8]	PIN_AD10	Slide Switch[8]	3.3V
SW[9]	PIN_AE12	Slide Switch[9]	3.3V

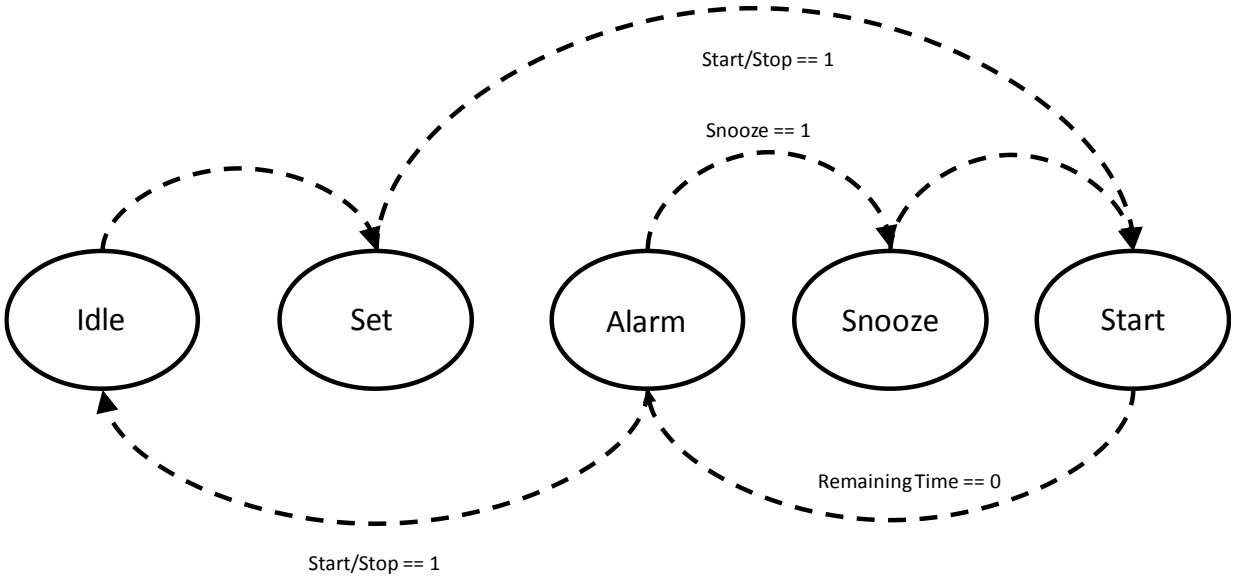
Table 3-7 Pin Assignments for Push-buttons

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
KEY[0]	PIN_AA14	Push-button[0]	3.3V
KEY[1]	PIN_AA15	Push-button[1]	3.3V
KEY[2]	PIN_W15	Push-button[2]	3.3V
KEY[3]	PIN_Y16	Push-button[3]	3.3V

Table 3-8 Pin Assignments for LEDs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LEDR[0]	PIN_V16	LED [0]	3.3V
LEDR[1]	PIN_W16	LED [1]	3.3V
LEDR[2]	PIN_V17	LED [2]	3.3V
LEDR[3]	PIN_V18	LED [3]	3.3V
LEDR[4]	PIN_W17	LED [4]	3.3V
LEDR[5]	PIN_W19	LED [5]	3.3V
LEDR[6]	PIN_Y19	LED [6]	3.3V
LEDR[7]	PIN_W20	LED [7]	3.3V
LEDR[8]	PIN_W21	LED [8]	3.3V
LEDR[9]	PIN_Y21	LED [9]	3.3V

APPENDIX B: STATE DIAGRAM OF THE CONTROLLER



APPENDIX C: SOURCE CODES

See ODTUCLASS