DIGITAL ALARM CLOCK REPORT

DR. Mohamed Shalan

Dareen Hussein - 900170917

Donia Ghazy - 900172124

22.07.2019

INTRODUCTION

Design and implement a digital clock with alarm function using Verilog HDL. The digital clock basically consists of a clock unit, time counter unit and a display unit. The clock unit generates the clock pulse of one second period. The FPGA kit has frequency oscillator which generates 50-MHz, 27 MHz and 24 MHz clock pulses. A frequency divider is used to reduce either of these clocks to a frequency of 1 Hz. The time counter unit counts the one second clock pulses and save them as seconds, minutes and hours in the memory. The display unit takes the data from the memory and displays them in the seven segment display. The alarm function is implemented as a separate block. The alarm unit asserts an output when the setup time is attained. The time of the digital clock has to be initially set by the user. So, the digital clock as a whole has user input and control unit. The input unit involves functions such as time reset, change of time, change of alarm time and stopping alarm. While, the control unit controls the blocks such as clock unit, time counter unit, display unit and alarm unit.

MATERIALS

1. Hardware: FPGA BASYS 3
2. Software: Verilog

SPECIFICATIONS

- The only used BASYS3 peripherals are: LD0, LD12, LD13, LD14, LD15, 7-Segment display and the push buttons.
- 7-Segment display usage: 2 digits (left) are used for hours and 2 digits (right) are used for minutes.
- There are two modes: “adjust” and “clock/alarm”.

  • Default mode is the “clock/alarm” (LD0 is OFF)

  • In the “clock/alarm” mode

o LD0 is OFF

o The 2nd decimal point from the left shall blink with a frequency of 2Hz.

o To adjust the clock press BTNC for a tleast 2 seconds to enter the “adjust” mode.

o When the current time matches the set alarm time the LED blinks; hitting any button stops blinking.

• In the “adjust” mode

o LD0isON.

o The 2nd decimal point from the left does not blink.

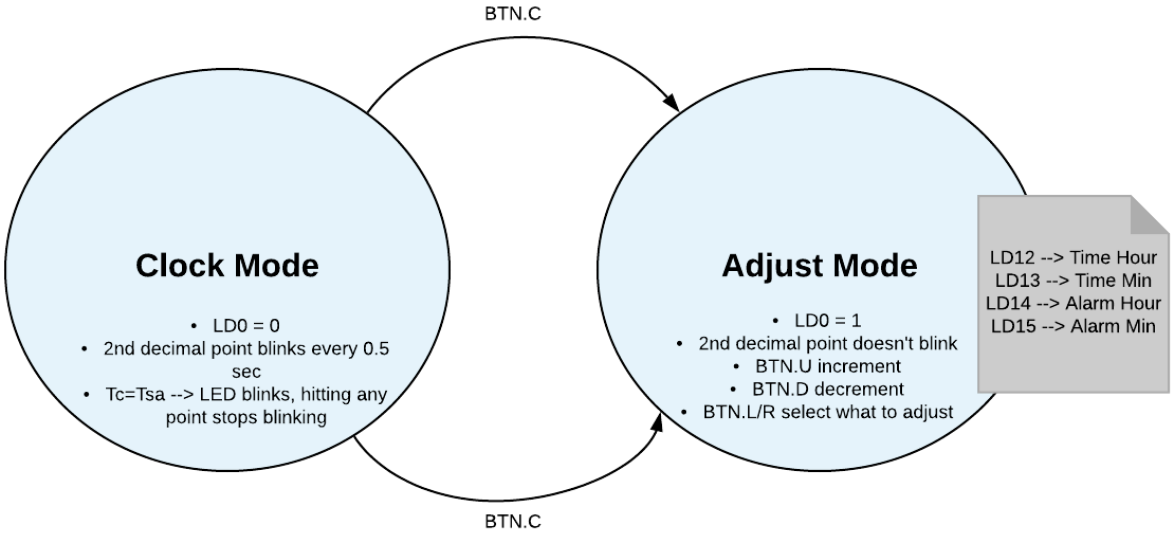
o Pressing BTNC exits the “adjust” mode to the “clock/alarm” mode.

o Pressing BTNL or BTNR selects what to adjust: “time hour”, “time minute”, “alarm hour”, and “alarm minute” (in sequence). LD12, LD13, LD14 and LD15 states reflect the parameter to be adjusted. For example, LD13 is ON while adjusting “alarm hour”; pressing BTNL change the parameter to be adjusted to “time minute”, turns LD14 on and turns LD12 off.

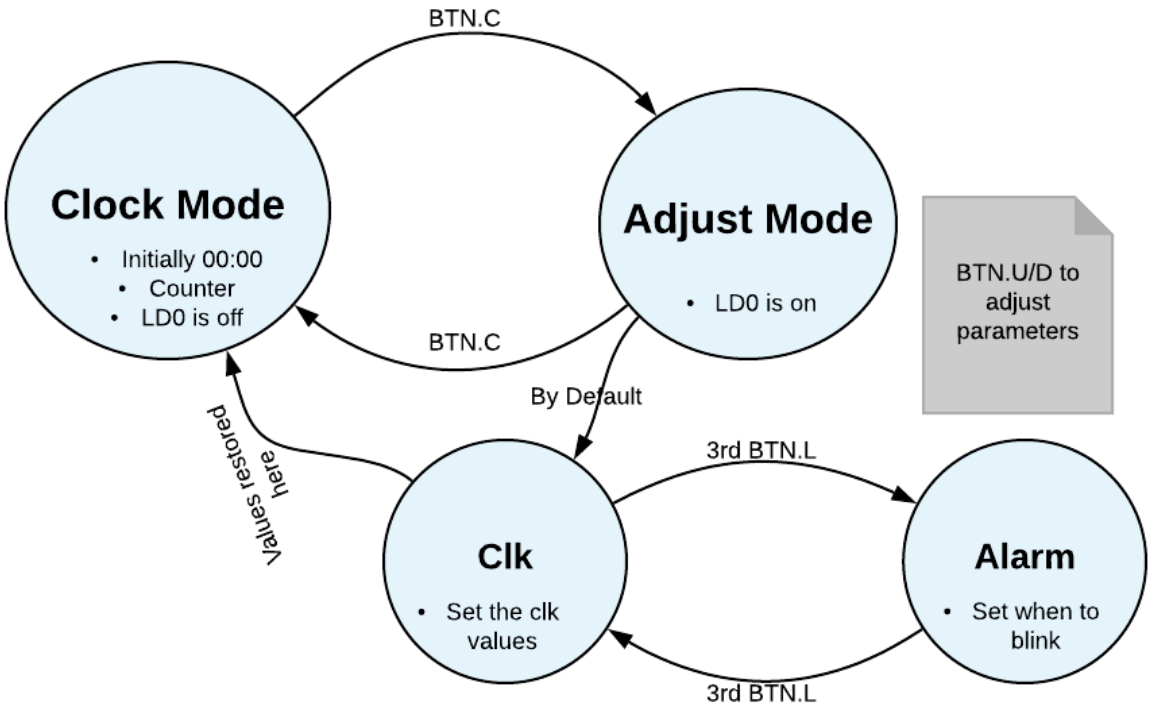
o Pressing BTNU increment the selected parameter (e.g.,“timehour”)

o Pressing BTND decrement the selected parameter (e.g.,“timehour”)

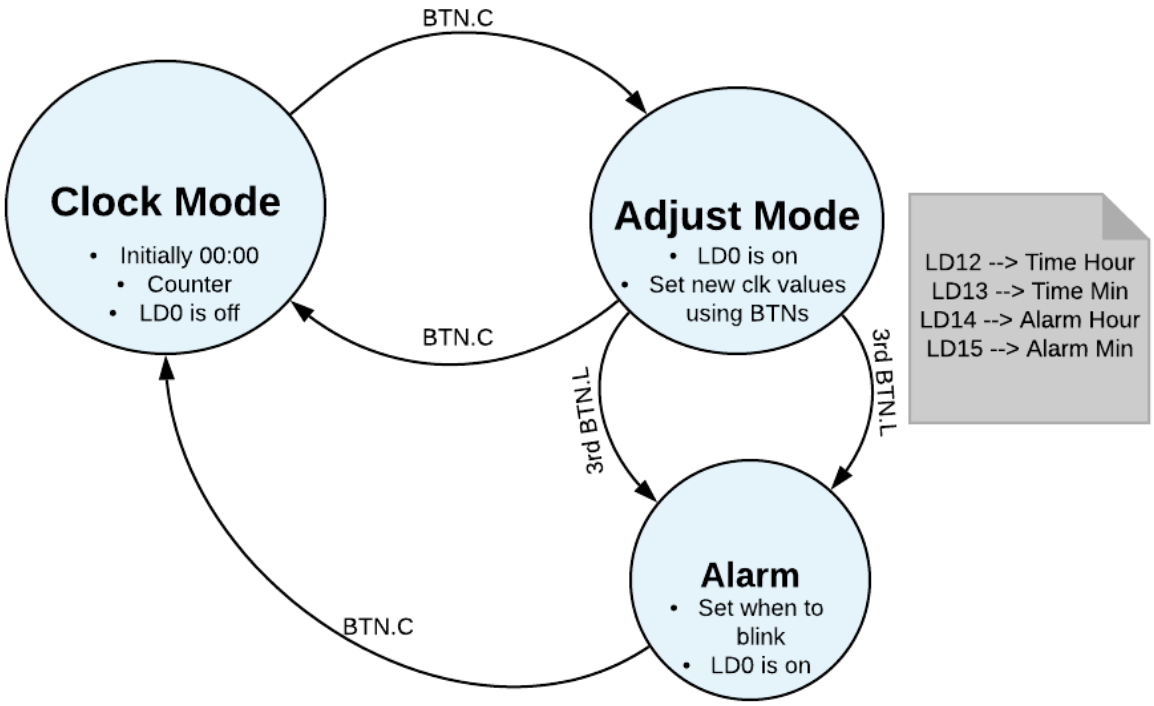
DESIGN PROCESS



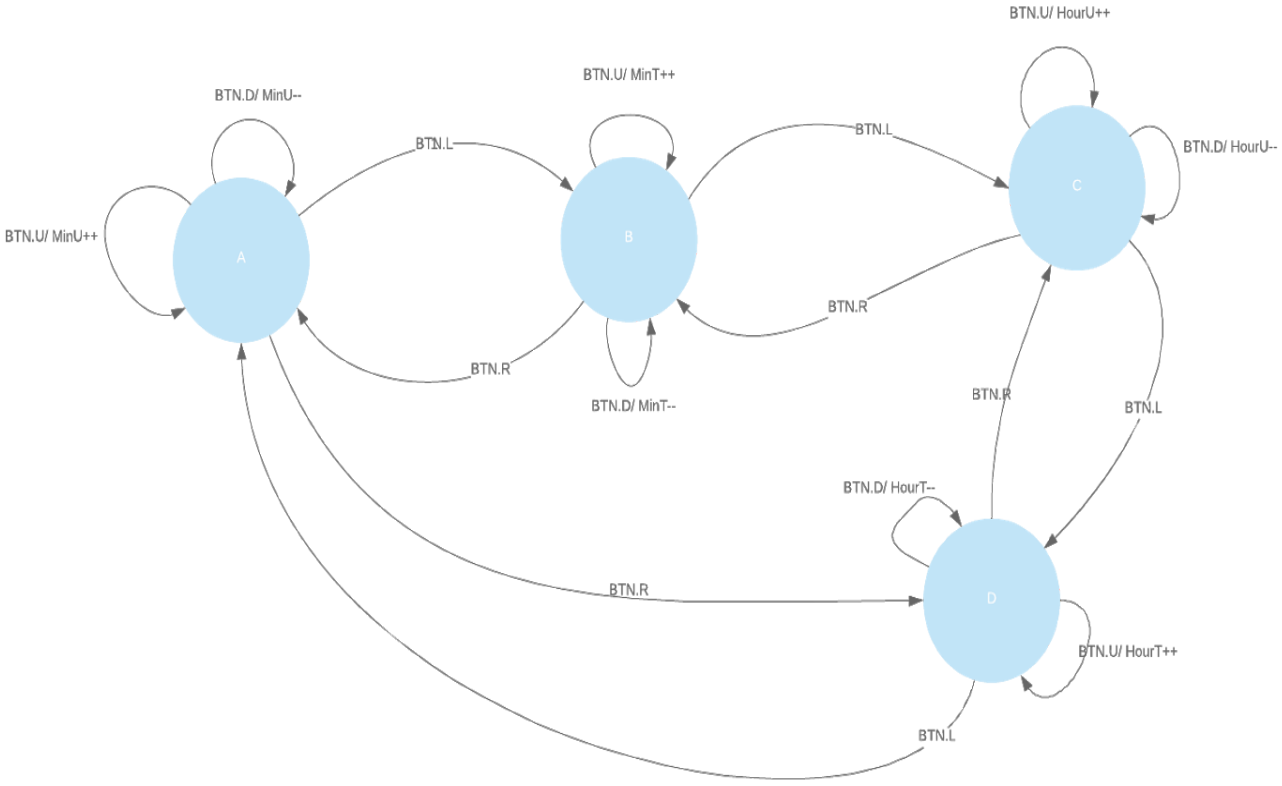
Fig(1) Shows the first proposed State Diagram



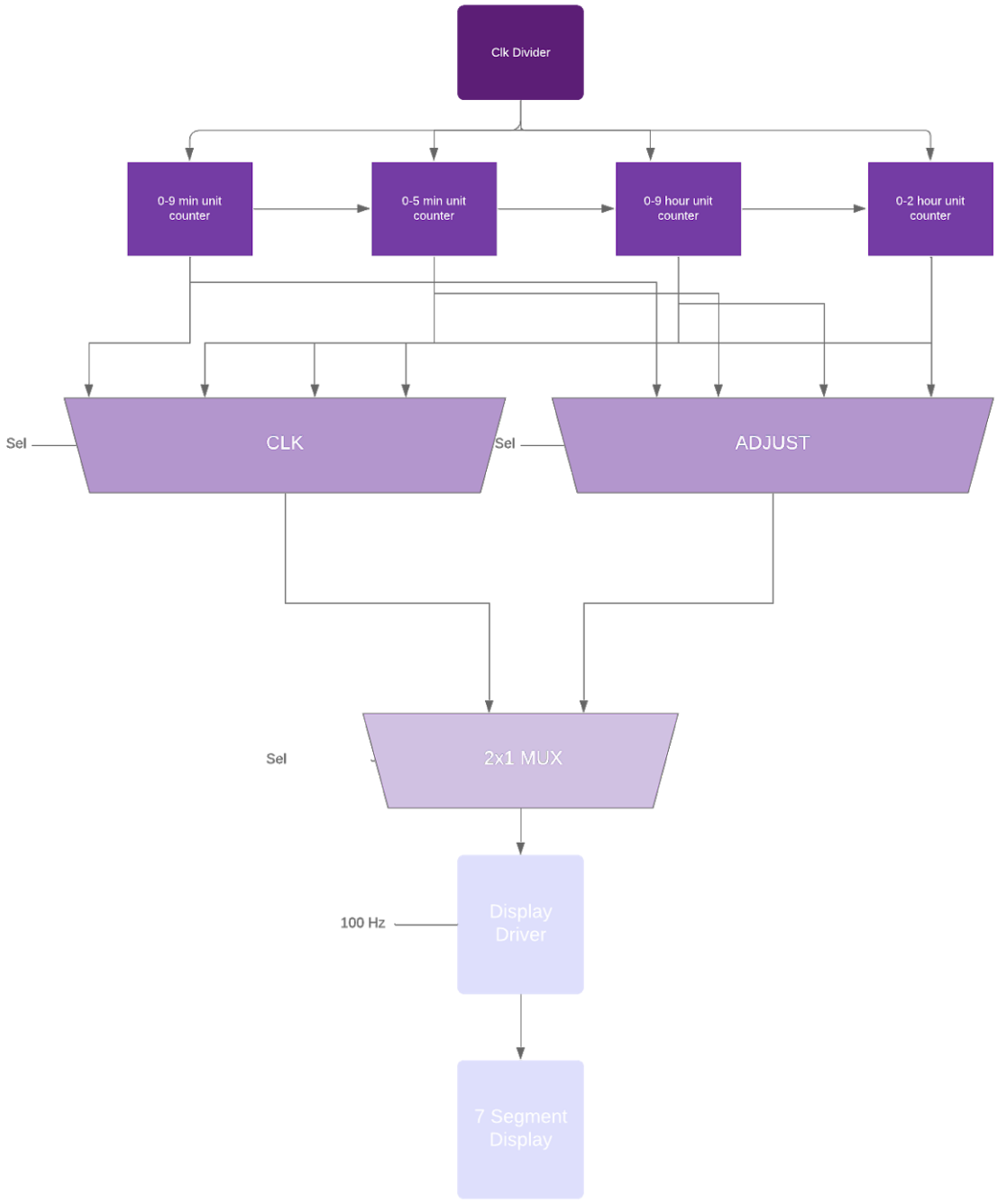
Fig(2) Shows the second proposed State Diagram



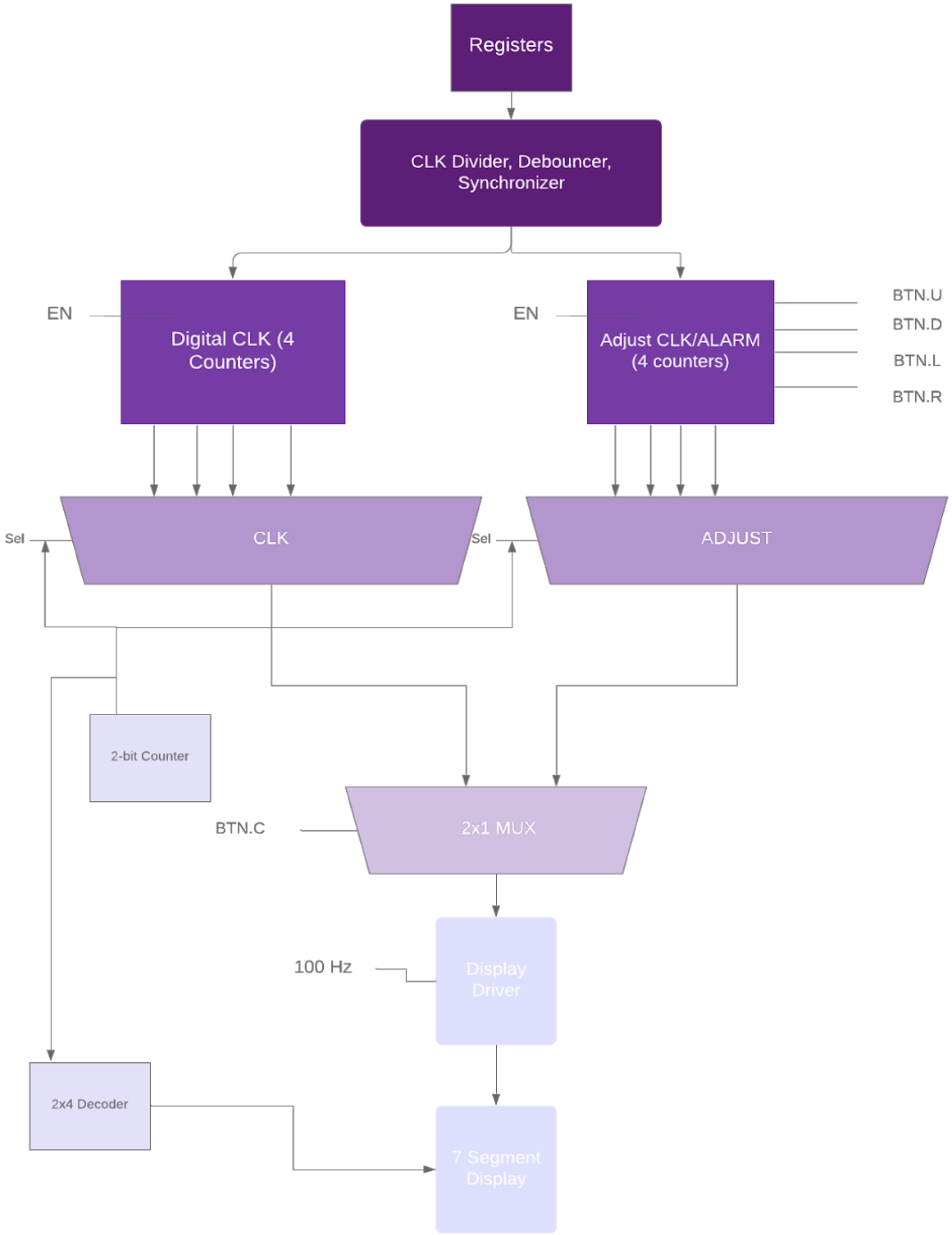
Fig(3) Shows the last proposed State Diagram and the one we implemented



Fig(4) Shows the Final State Diagram of Adjust Mode



Fig(5) Shows the first proposed Block Diagram



Fig(6) Shows the second and final proposed Block Diagram and the one that was implemented

IMPLEMENTED MODULES

1. **Module debouncer**

   This module is used to stabilize the signal of the push buttons.

2. **Module sync**

   This module synchronize the button signal with every rising clock edge.

3. **Module risingEdge**

   This module detects the posedge of the clock.

4. **Module ClockDivider**

   This module set the frequency of the FPGA to be adjusted as needed.

5. **Module count5**

   This module counts until modulo 5 for the minutes tens.

6. **Module count9**

   This module counts until modulo 9 for the minutes/hours units.

7. **Module count2**

   This module counts until modulo 2 for the hour tens.

8. **Module hms**

   This module has six counters; two for seconds, two for minutes and two for hours. The enable of each counter is set according to the value of the previous counter in order to accommodate for change in all counters in the right time. Moreover, the digital clock module counters count according to a clock that comes out of a clock divider that has a frequency of 1 Hz. Furthermore, the digital clock module only outputs the outputs of four counters [Hour_tens, Hour_units, Min_tens,Min_units] of the six.

9. **Module alarm22**

This module is responsible for setting the alarm requirements and compare it with the clock timing.

10. **Module segment7**

This module is responsible for driving the 7 segments on the FPGA and adjust the 7 leds to be displayed, then display the driver.

11. **Module switcher**

This module is a multiplexer responsible for controlling if the FPGA is ON or not.

12. **Module adjustSwitcher**

This module is a counter that is considered with adjusting the position of the leds according to the values of BTNR, BTNL.

13. **Module binAdj**

This module control the state of the clock, if it is paused or working or getting adjusted.

14. **Module modes**

This module is responsible for setting and switching between the different modes of the FPGA. It is also composed of four counters, each is responsible for adjusting one of the following: Hours_tens, Hours_units, Min_Tens, Min_units. The driving modules of this module are binary counters that has two additional inputs which are BTNU and BTND in order to increment or decrement the values of the Counter.

15. **Module all**

The main module drives all the other modules. It contains a mod 2 binary counter, a digital clock module, an alarm module, display driver module, two 4x1 multiplexers and one 2x1 multiplexer and four registers. The flow of the design goes as the following. The main module has a binary counter whose count is initially set to zero which refers to

the state of the digital clock, and this counter is enabled by BTNC which switches the state between the digital clock and the adjust mode. The outputs of the digital clock or the adjust module are forwarded to a 2x1 mux that chooses which one is going to get displayed according to the select value which is an output of the mod2 binary counter. In order to enable the leds , four enables are used to differentiate between the values alarm_hours, alarm_minutes , clock_hours , clock_minutes. The four registers are used to store the outputs of the adjust mode, and if they are alarm values then they are compared constantly with the values of the clock in order to blink LD0 and if they are clock values then they are loaded to the clock registers.

IMPLEMENTATION ISSUES

1. The user can't get directly to the alarm mode he has to pass by the adjust clock mode first.
2. The Buttons don't stop the alarm.

VALIDATION ACTIVITIES

1. Observing the clock count and making sure it stops at 23:59 and restart with 00:00.
2. Comparing the clock count to a real clock to validate our used frequency/clock divider.
3. Switching between the two modes using BTNC.
4. Setting different values to adjust the clock.
5. Setting different values to adjust the alarm.
6. Testing the increment and decrement options and making sure that they ring around the values.