*Name: Trung Kien Nguyen*

*Student ID: 104053642*

*From Lab 18 (10:30 a.m to 12:30 a.m on Friday)*



COS10004

COMPUTER SYSTEMS

Report

Assigntment 1



**Table of contents**

*Table of contents…………………………………………………………………………………………………….…2*

*Introduction………………………………………………………………………………………………………………3*

*Description………………………………………………………………………………………………………………..4*

*Used Components and Sub circuits description…….……………………………………………4*

*Mod 6 counter………….………………………………………………………………………………4*

*Mod 9 counter………….………………………………………………………………………………4*

*Minutes Circuit…………………………………………………………………………………………5*

*Hours Circuit…………………………………………………………………………………………….5*

*Mins and Hours Circuit……………………………………………….…………………………….6*

*Alarm sub Clock Circuit……………………………………………….…………………………….6*

*Enable Alarm Circuit………….……………………………………….…………………………….6*

*Main Circuit……………………………………………………………………………………………..8*

*How does the “Set Time” mode work?..................................................................9*

*How does the “Set Alarm” mode work?...............................................................10*

*Unsolved problems………………………………………………………………………………………….12*

*Conclusion……………………….……………………………………………………………………………………...12*

*References…………………………….…………………………………………………………………………………12*

**Introduction**

*Hi, I’m Trung Kien Nguyen, student ID 104053642, now I’m studying COS10004 Computer Systems, at lab 18 from 10:30 am to 12:30 am on Friday weekly.*

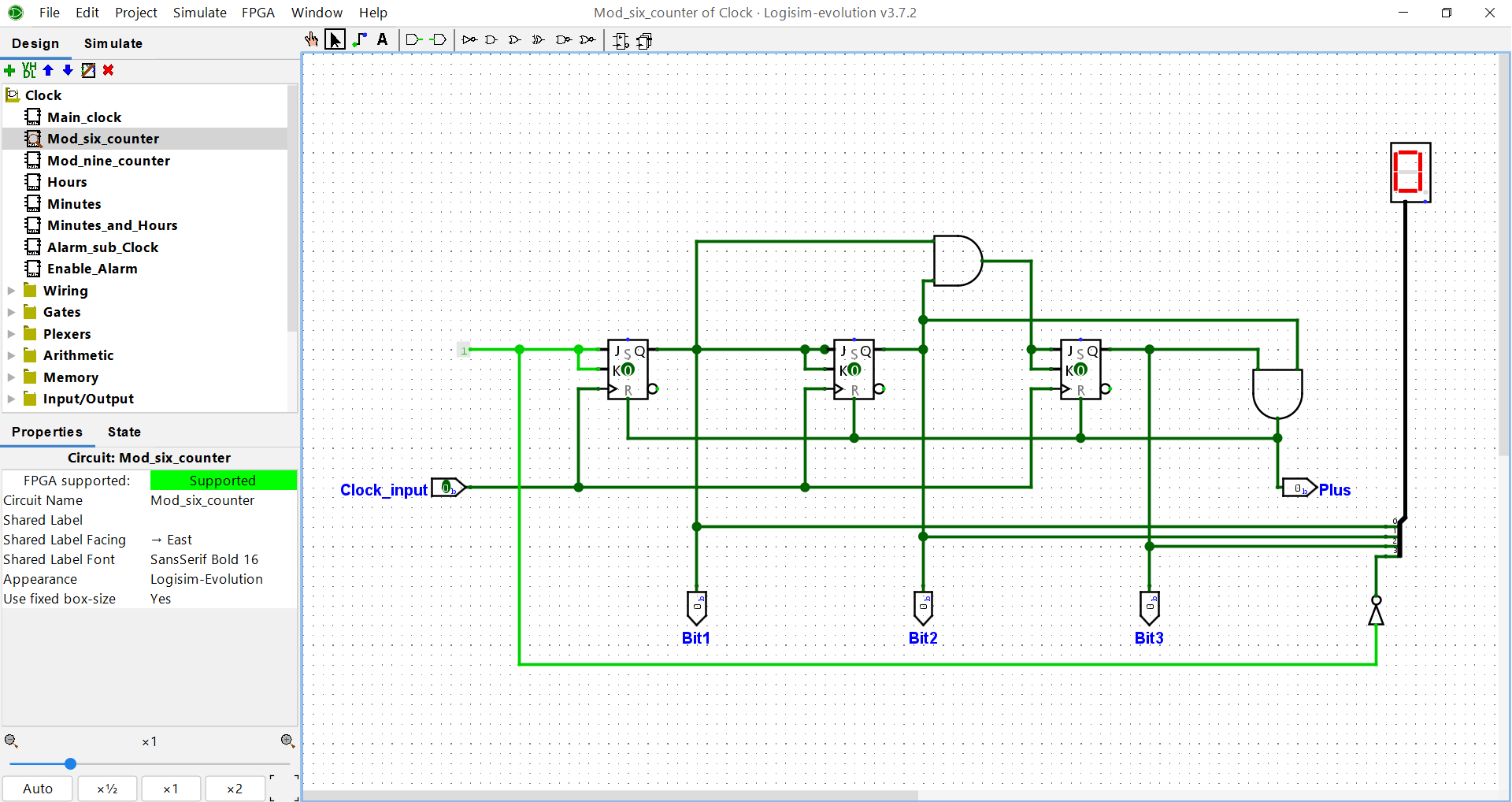
*This is my report for Assignment 1 – Alarm Clock, which will provide a clear description of my work in the recent weeks.*



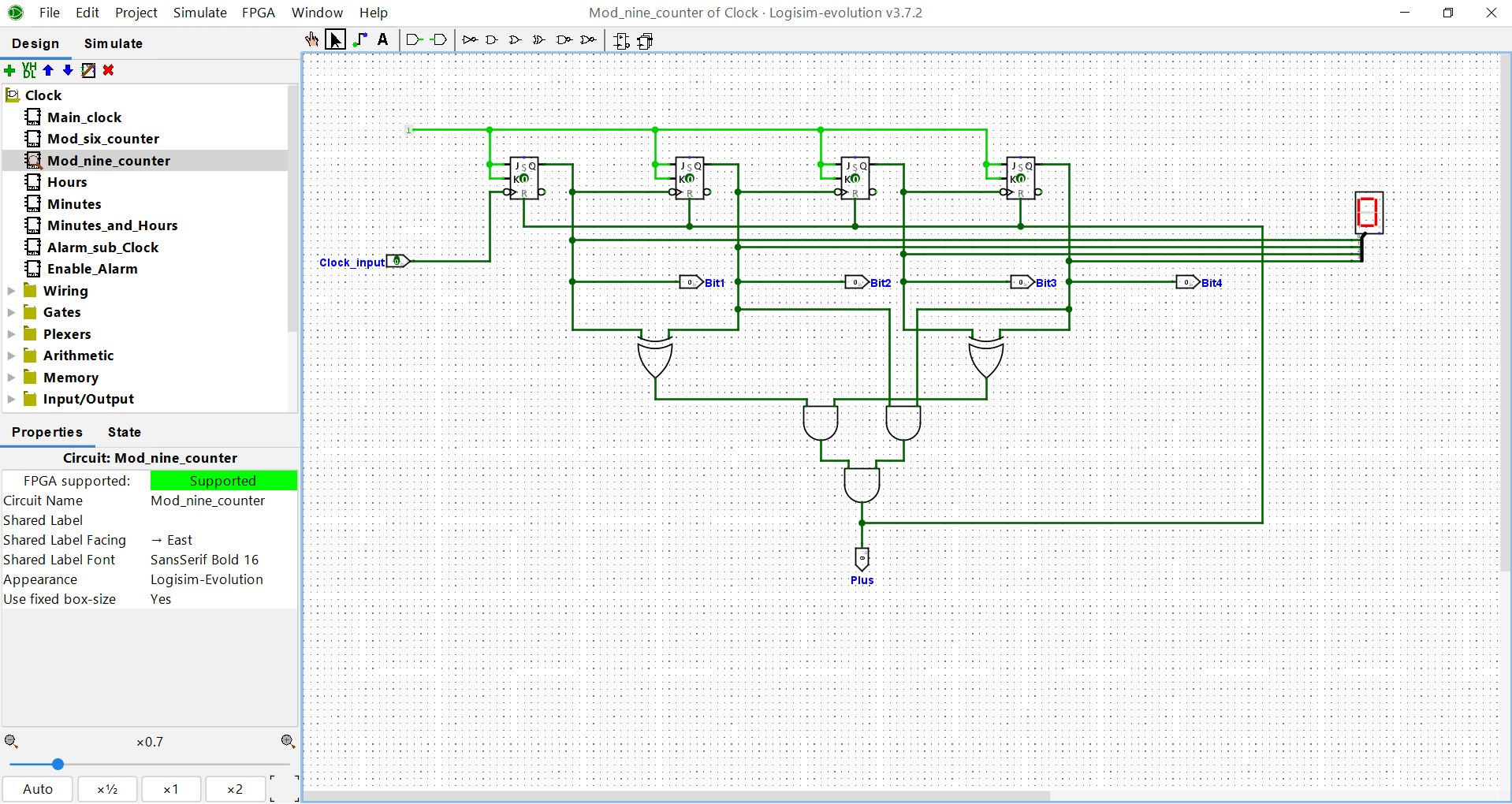
**Description**

*My work consists of a main circuit for the Main Clock and 7 auxiliary circuits, including a mod 6 counter, a mod 9 counter for displaying minutes, a mins circuit, an hours circuit, a clock circuit combining mins and hours ones, two circuits for alarm mode (Alarm Sub Clock, Enable Alarm).*

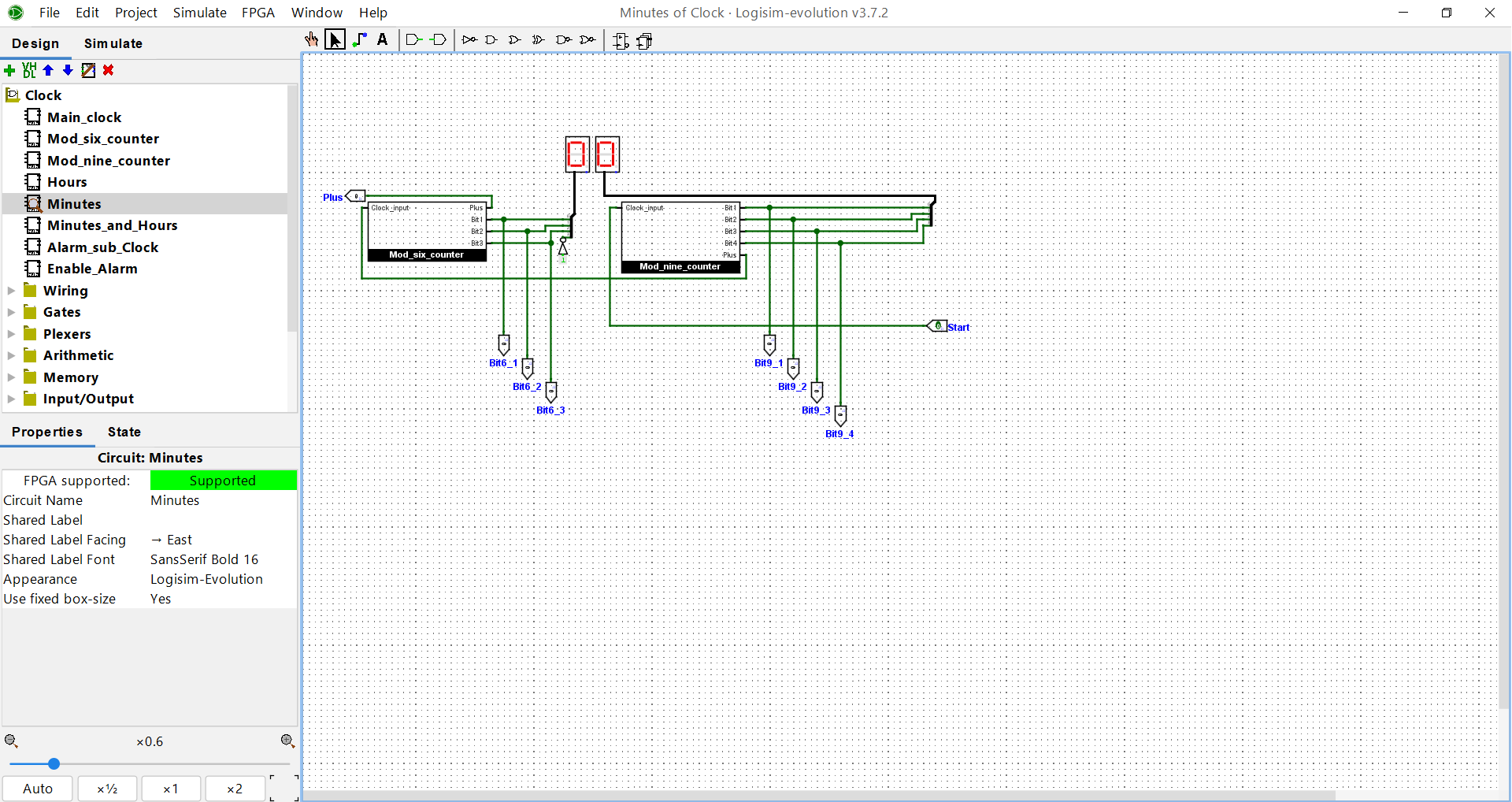
**Mod 6 Counter:** Simply designed with 3 JK Flipflop, that I have done in Lab 3



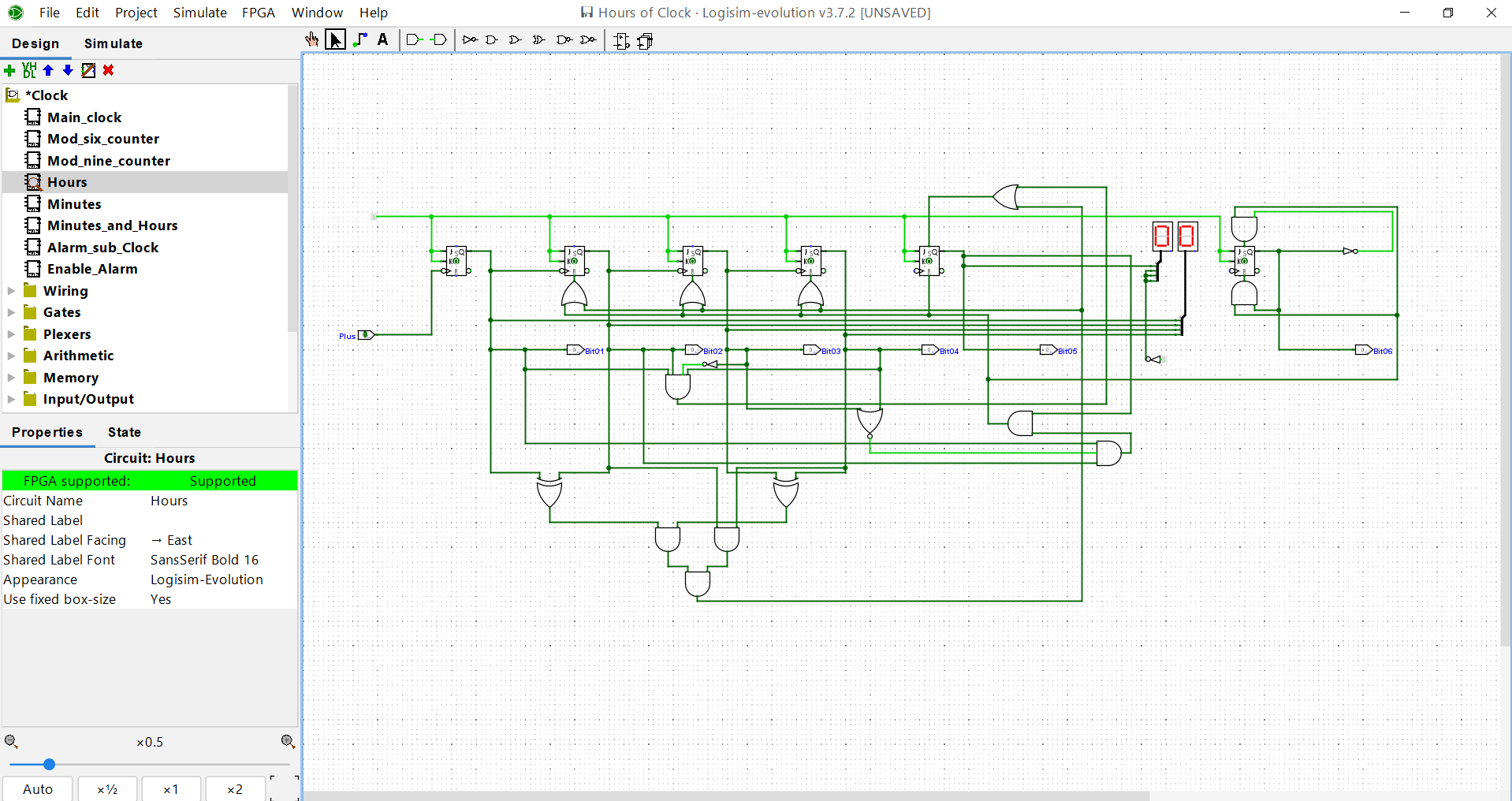
**Mod 9 Counter:** I developed it from the design of Mod 6 Counter, I used 4 JK Flip Flop, along with some gates to make sure the value will return to 0 after reaching 9.



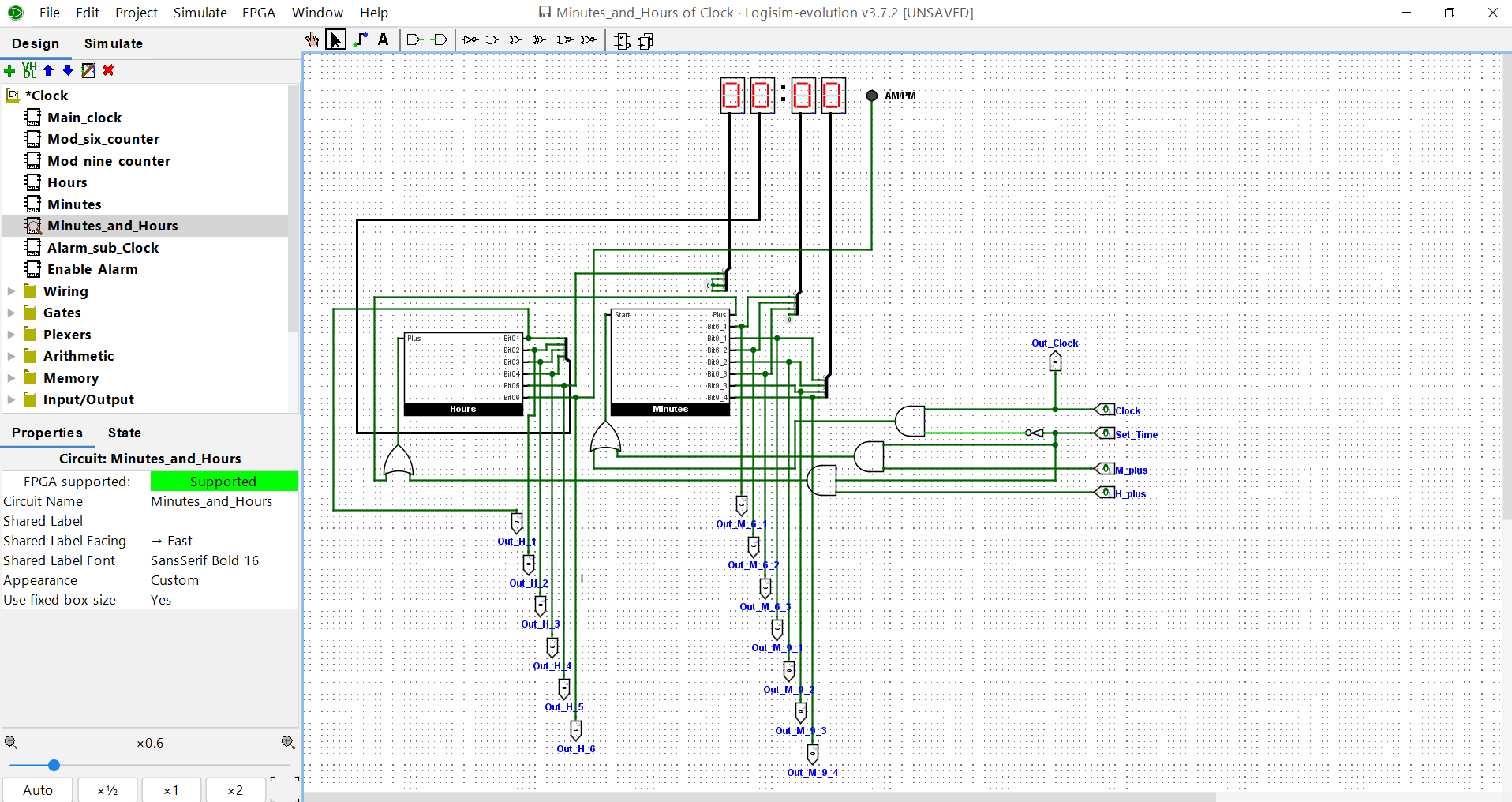
**Minutes Circuit:** Basically made from Mod 6 Counter and Mod 9 counter, to display the values between 00 and 59



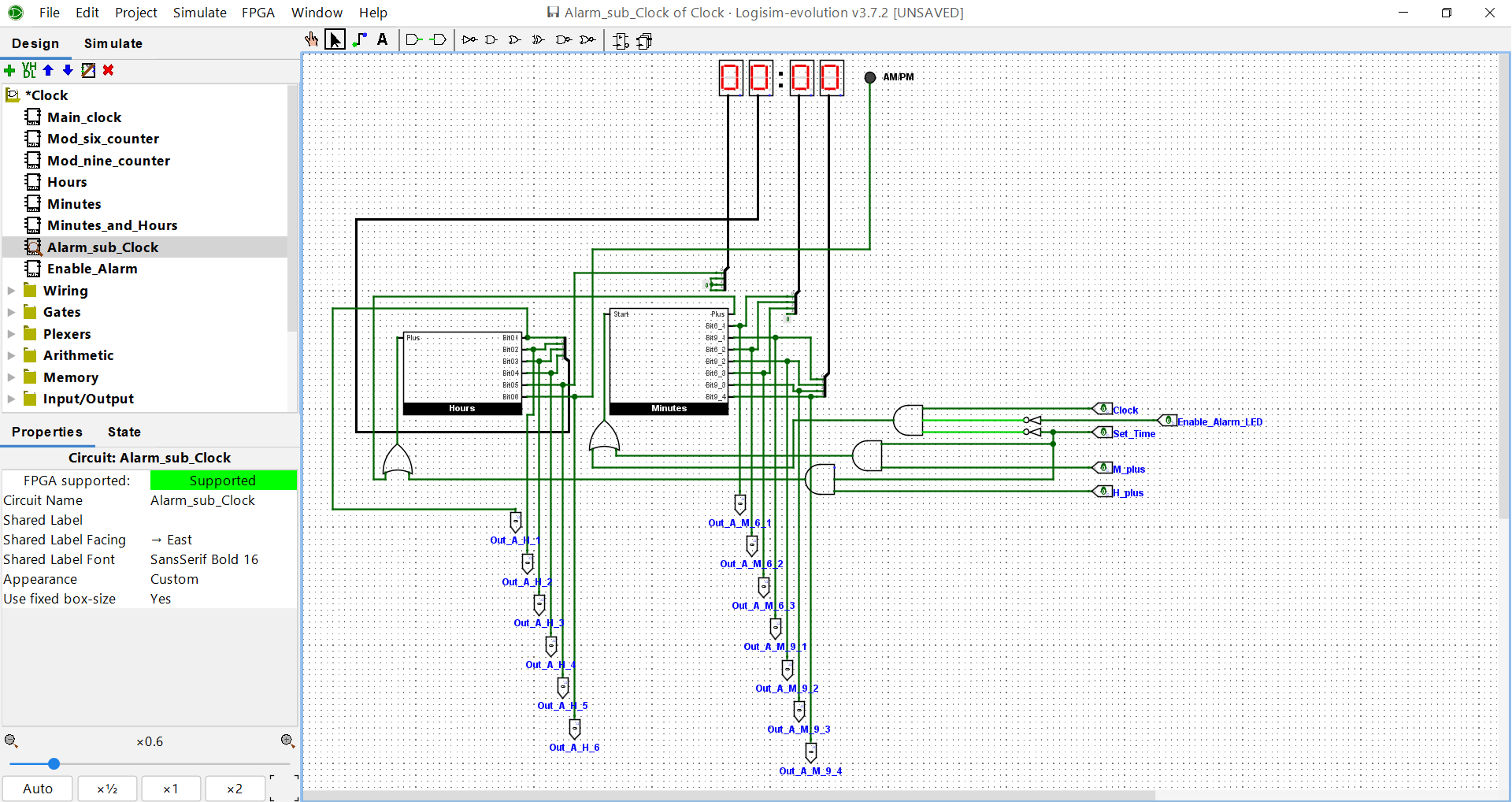
**Hours Circuit:** It is a complicated counter, the first display can show the values between 0 and 9. In the beginning, it will run from 0 to 9, then return to 1, then 2, and continue to return to 1 to count up to 9, at which the cycle begins, while the second display can only show the values of 0 and 1: 0 when the first one is from 1 to 9 (or 0 to 9 in the first running) and 1 when it is from 1 to 2. In addition, it also has an output to displayed AM/PM status, connected with a JK FlipFLop, which is “0” at the beginning, then changes each time the two other displays turn from 12 to 01.



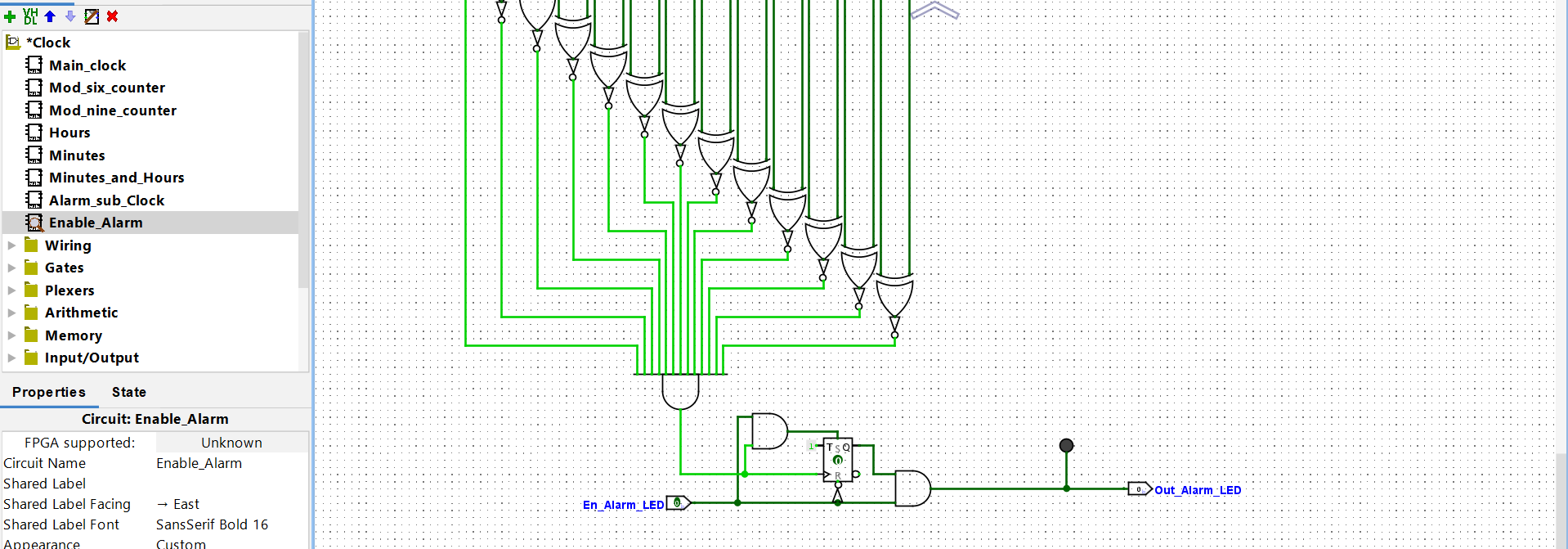
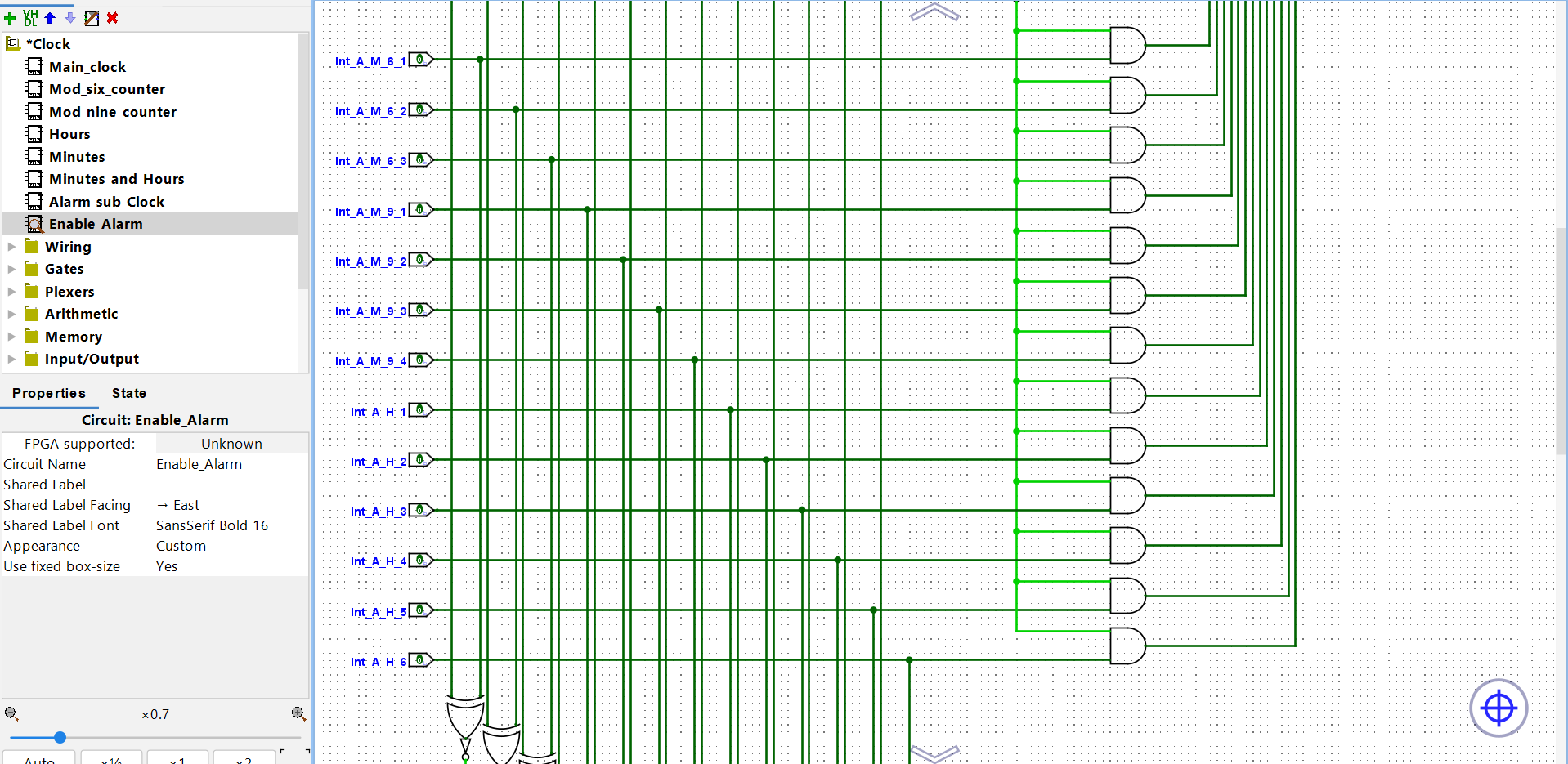
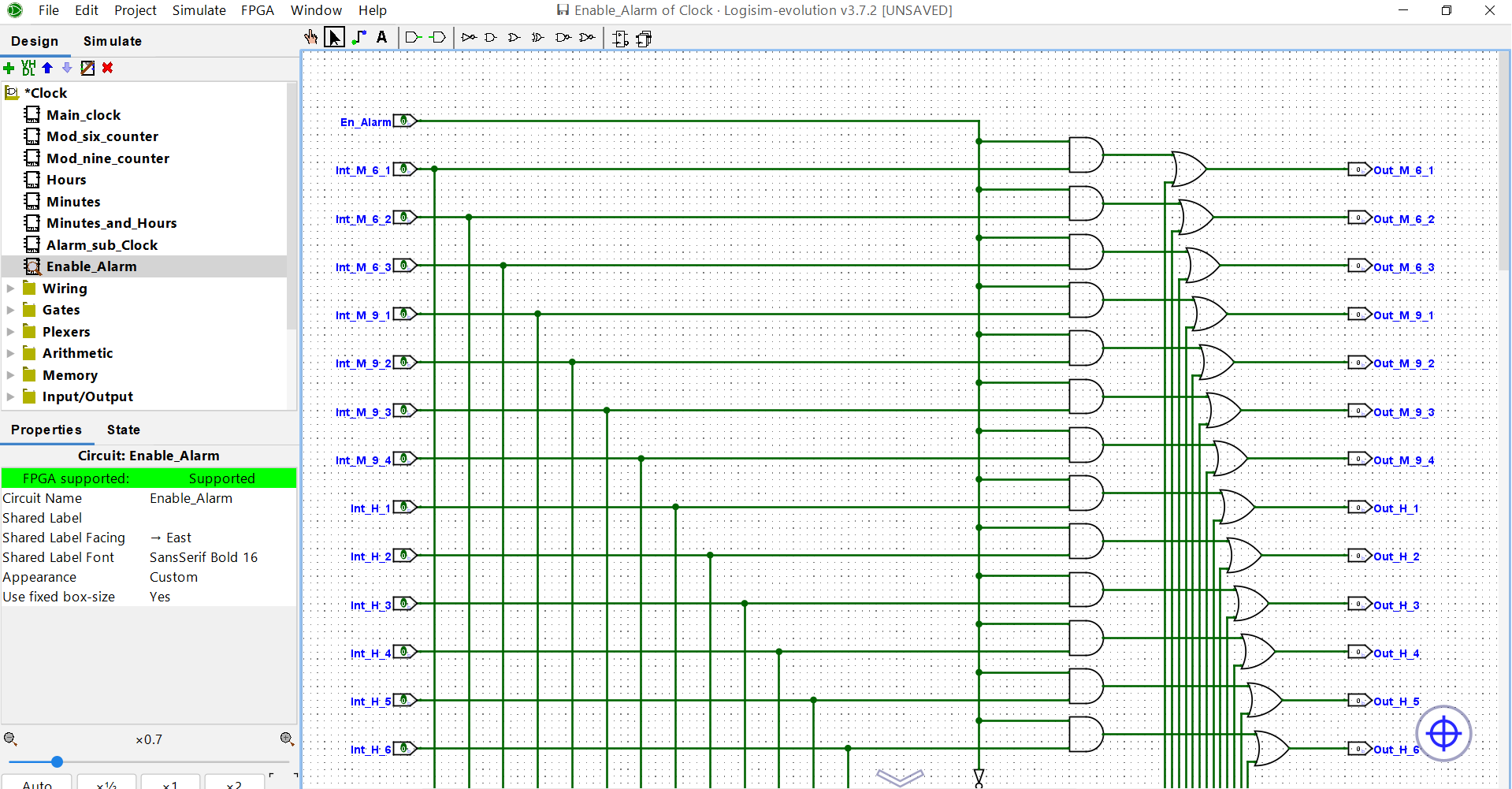
**Minutes and Hours Circuit:** As its name, it simply is a combination of hours and minutes circuits, with some functional buttons except “Set Alarm” and “Enable Alarm LED”.



**Alarm Sub Clock Circuit:** A sub clock that is almost similar to the Minutes and Hours Circuit, except a button for “Enable Alarm LED” (This is made for “Set Alarm Mode”)

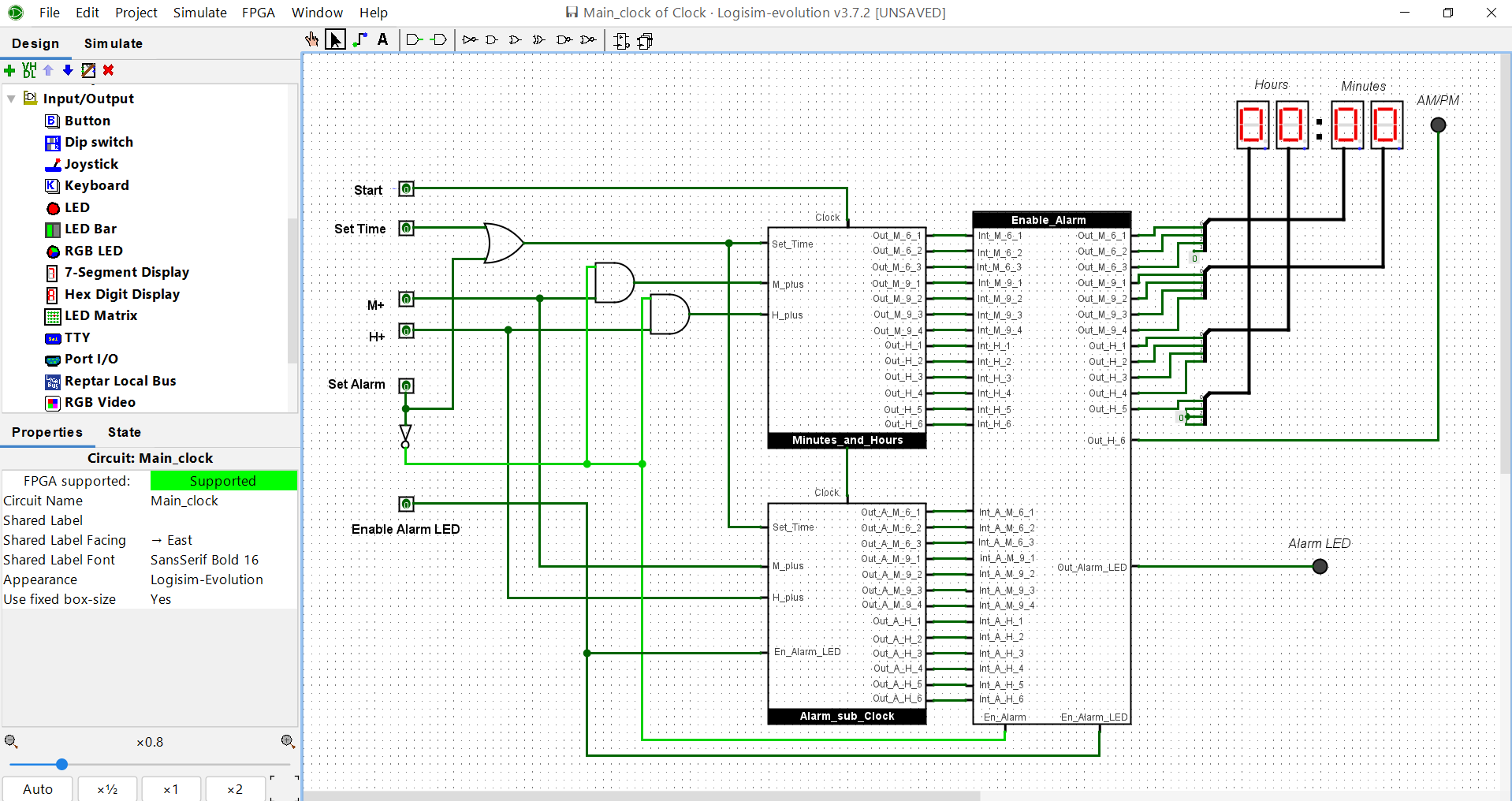


**Enable Alarm Circuit:** This circuit has 28 inputs, including 13 inputs getting values from the Minutes and Hours Circuit, 13 from Alarm Sub Clock, and two inputs for “Enable Alarm” and “Enable Alarm LED” respectively. I will describe in detail these three alarm circuits’ functions in the following words.



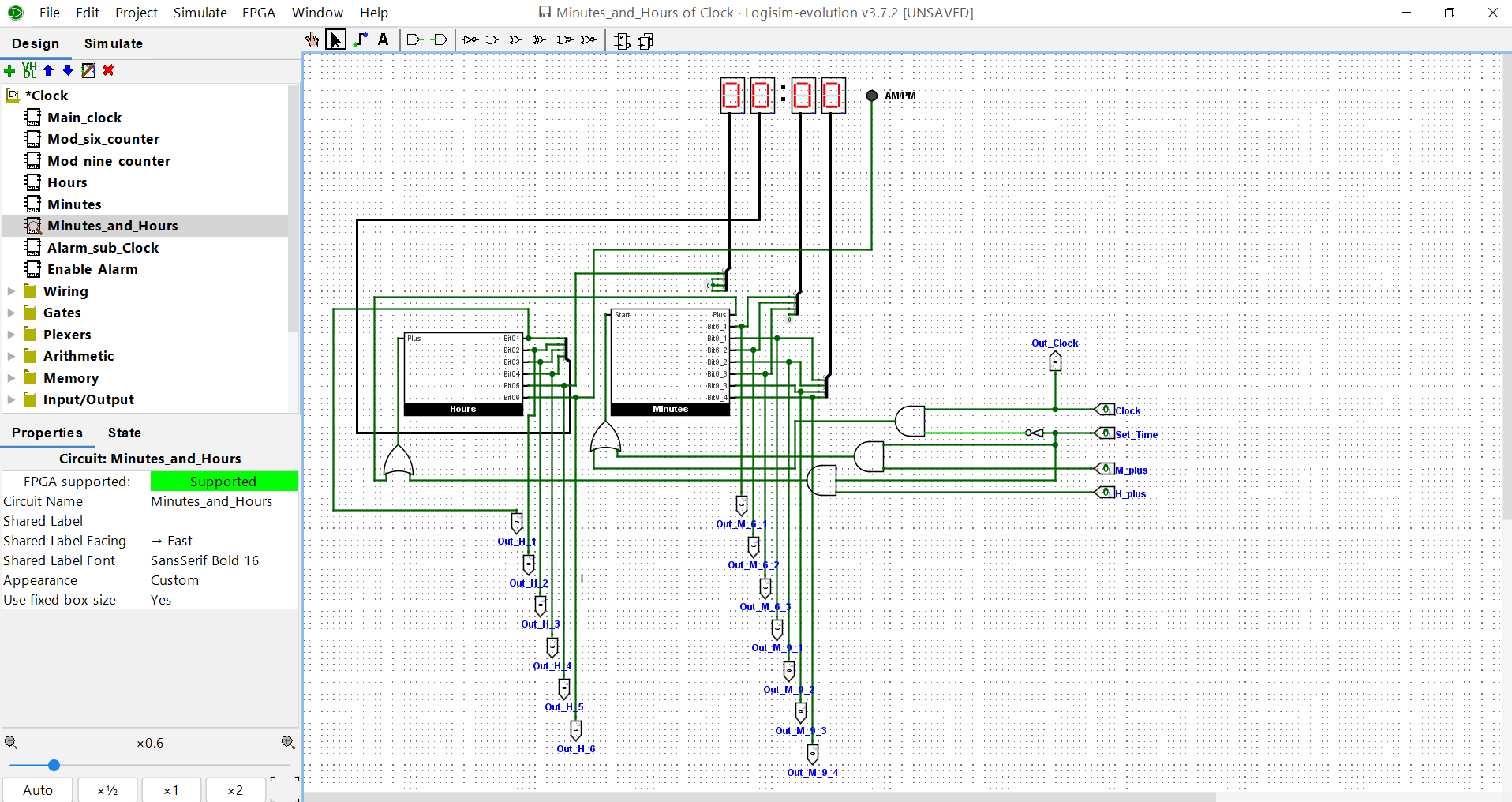
**Main Circuit:** This consists of 3 sub-circuits that I have mentioned: “Mins and Hours”, “Alarm Sub Clock” and “Enable Alarm”, four hex digit displays to illustrate hours and mins values, a LED for AM/PM status and another one for Alarm. Also, the main circuit has a total of 6 functional buttons, along with some different gates to make it works reasonably and follows these rules:

* A “Start” button, to let the users start the time in the beginning.
* A “Set Time” button to switch the clock to “Set Time” mode. When this mode is enabled:
  + The time display pauses.
  + Users can tick in the“M+” button to increase the minute displaying value by 1.
  + Users can tick in the “H+” button to increase the hour displaying value by 1.
  + When this mode is disabled, the clock will resume from the displaying time after manually incrementing.
* A “Set Alarm” button to switch the clock to “Set Alarm” mode. When this mode is enabled:
  + The time display pauses.
  + Users can tick in the “M+” and “H+” buttons to increase the minute and hour displaying value by 1 respectively, to choose the time they want to set the alarm.
  + Users also need to enable the “Enable Alarm LED” button first so that the alarm LED will work when the alarm time comes.
  + When this mode is disabled by turning off the “Set Alarm” button, the clock will resume from the time that was on the display prior to entering the “Set Alarm” mode. When the set alarm time comes, the LED will on until the users turn off the “Enable Alarm LED”.
* A “M+” button and a “H+” button for the two mentioned modes.



***How does the “Set Time” mode work?***

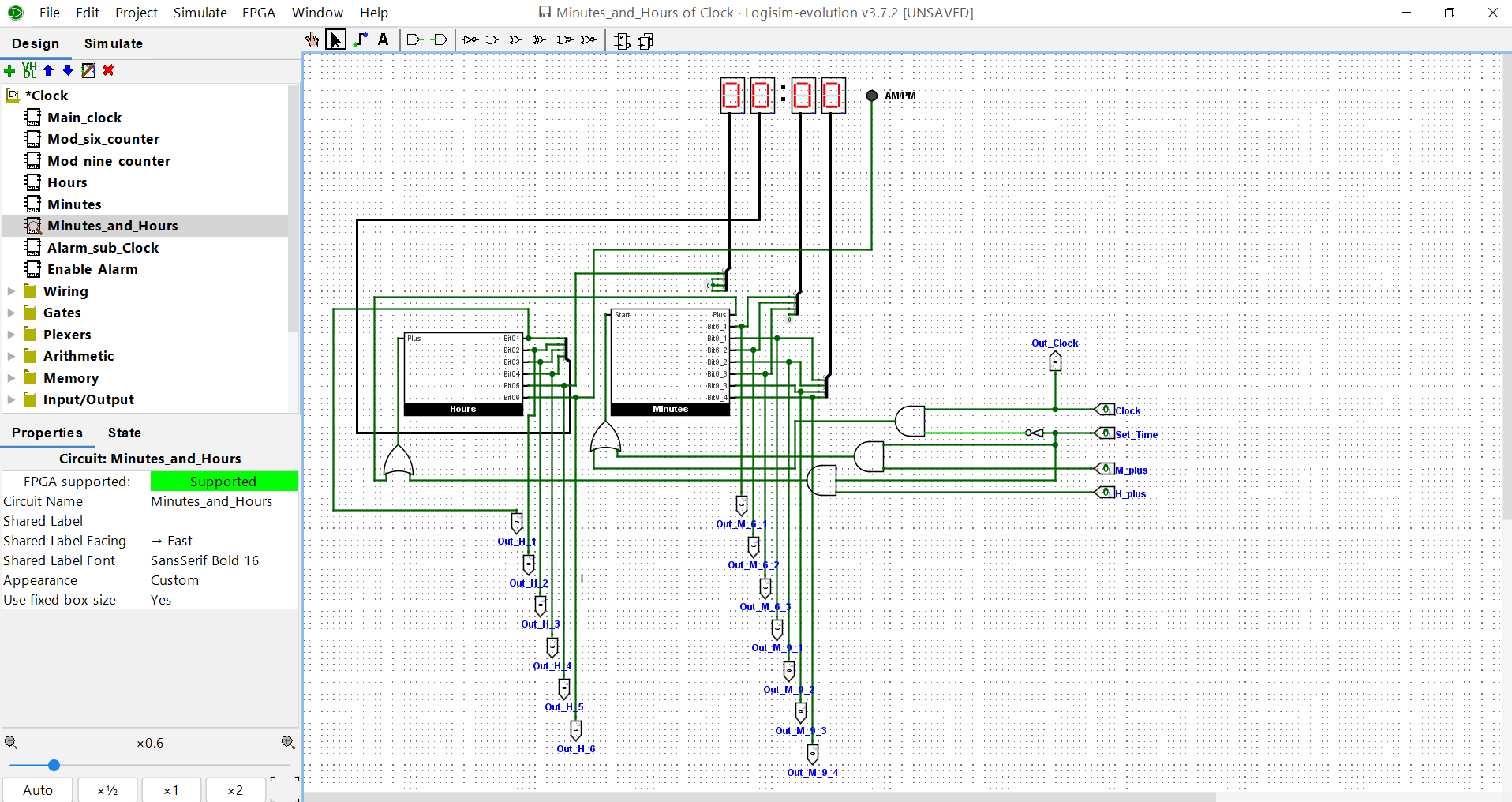
In the event of no “Alarm” mode, the “Mins and hours” sub circuit is responsible for displaying the time, with the main inputs of “Start”. In this sub-circuit, I use another input called “Set Time”, with an AND gate, and a NOT gate to make sure that values from the “Start” input cannot go through the start gate of the “Minutes” sub-circuit to begin the time when the “Set Time” button is on.





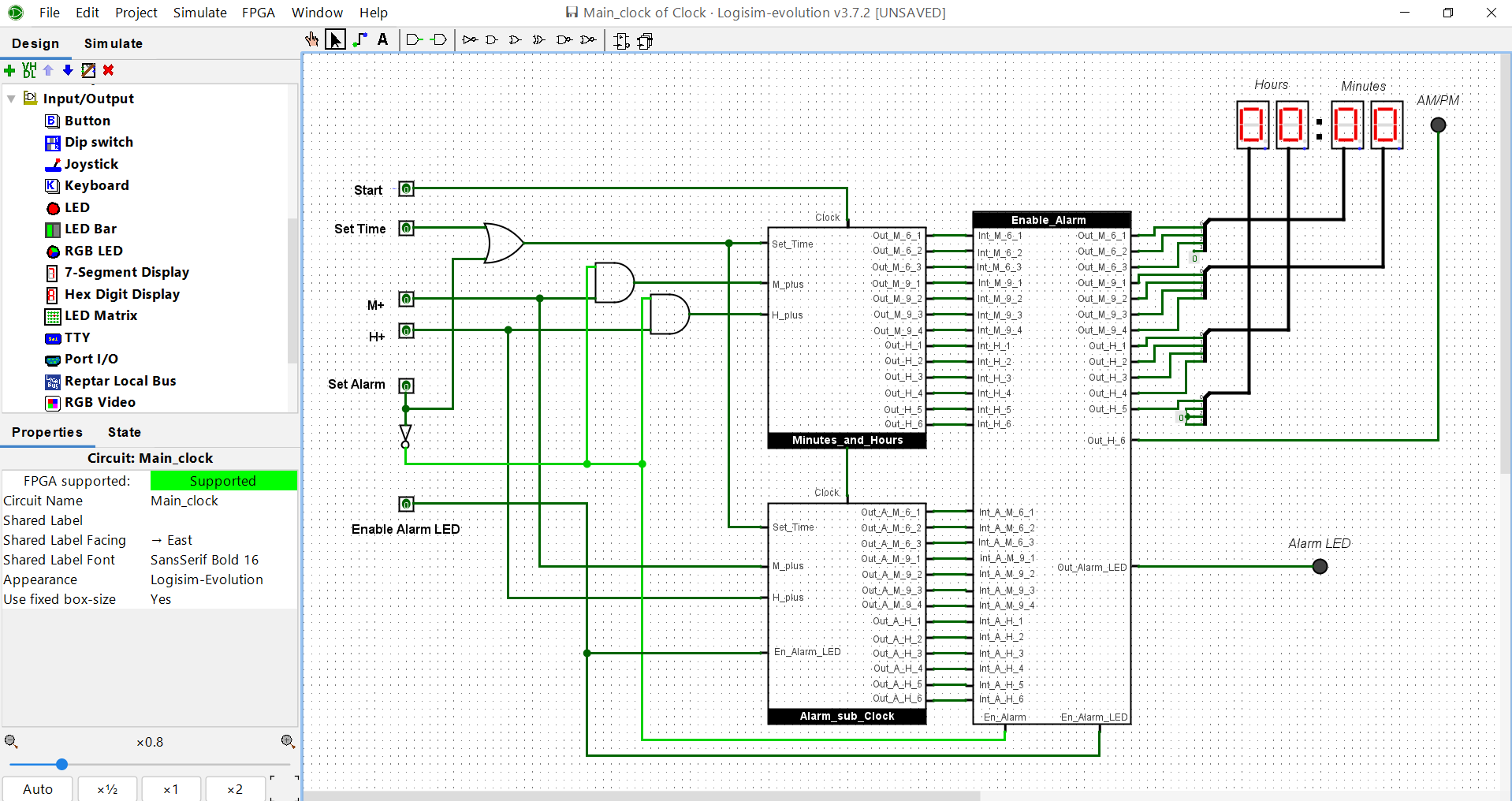
In addition, this sub-circuit has two smaller sub-circuits names “Minutes” and “Hours” with two separated inputs gates to increase the value of displaying minutes and hours respectively. When the “M+” or “H+” button is activated, I use AND gates and then OR gates to activate these inputs to increase the value manually.

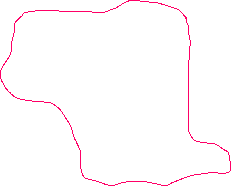




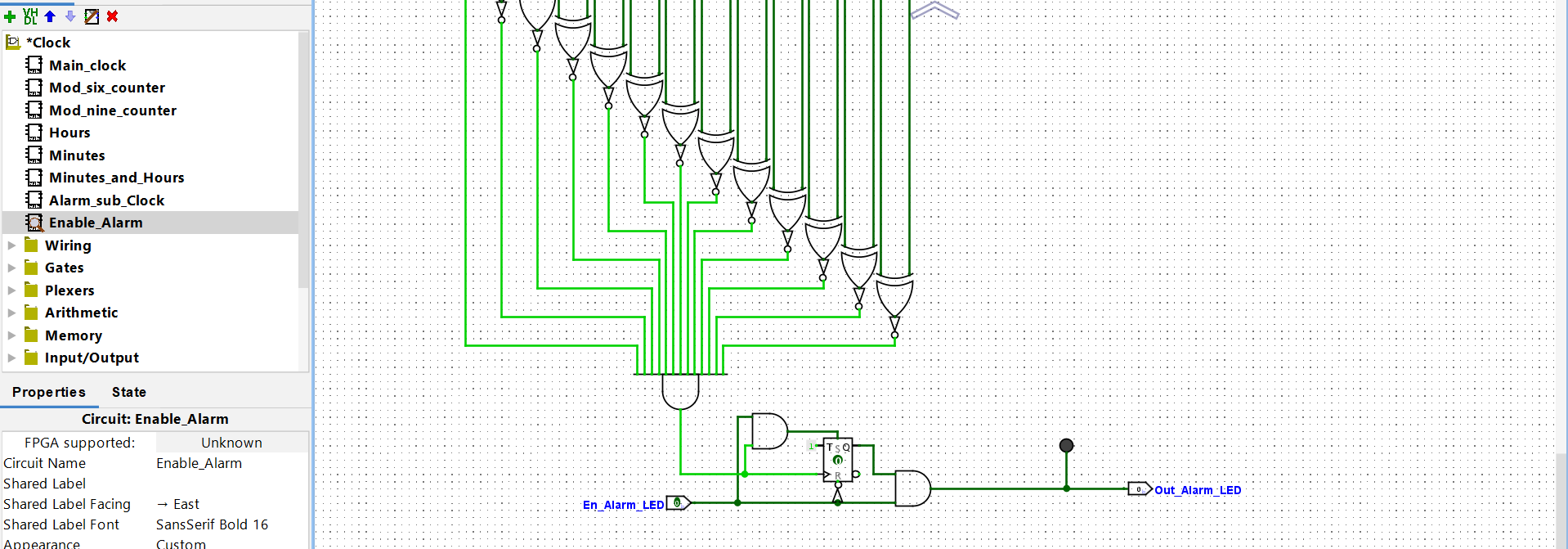
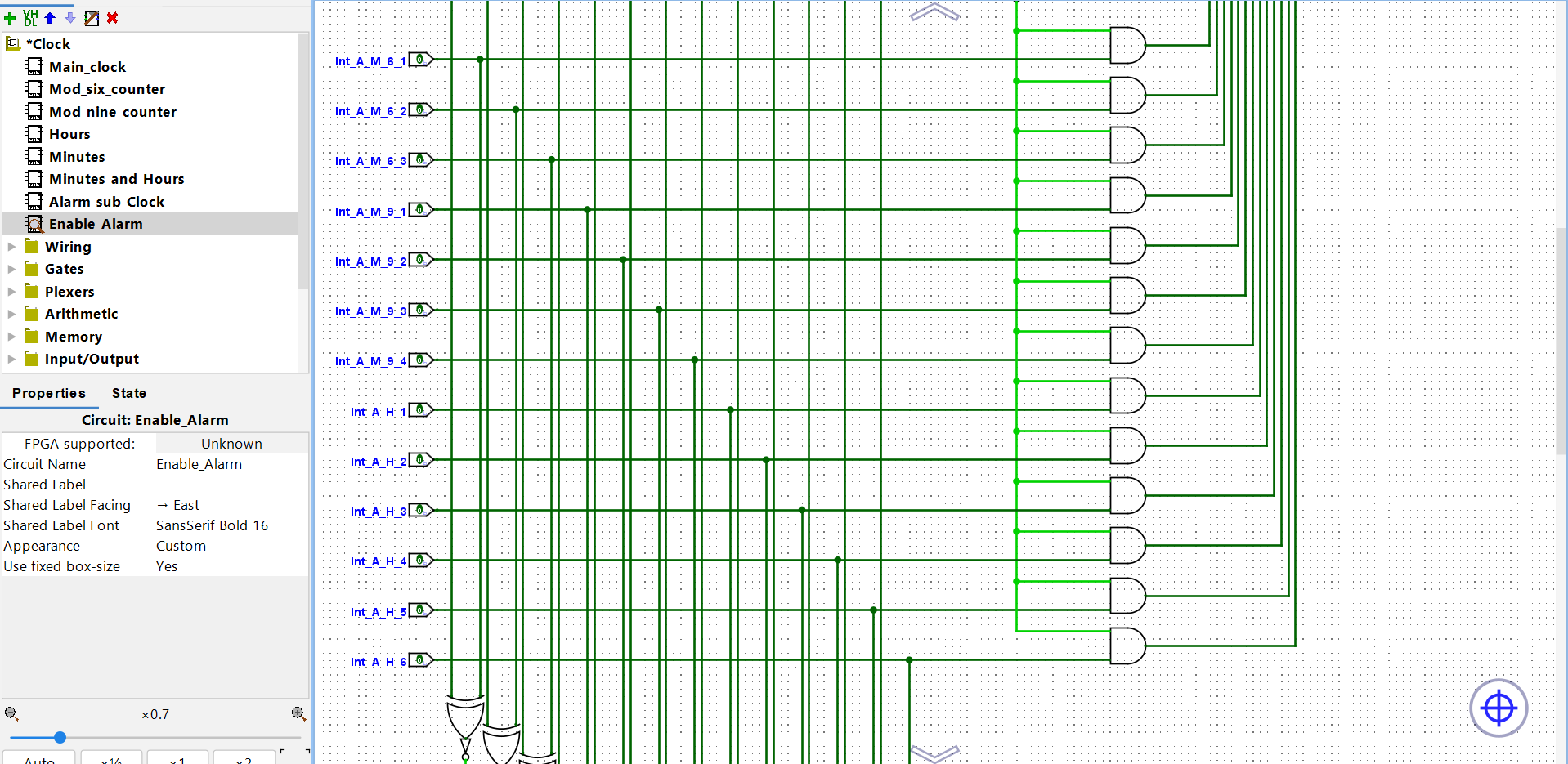
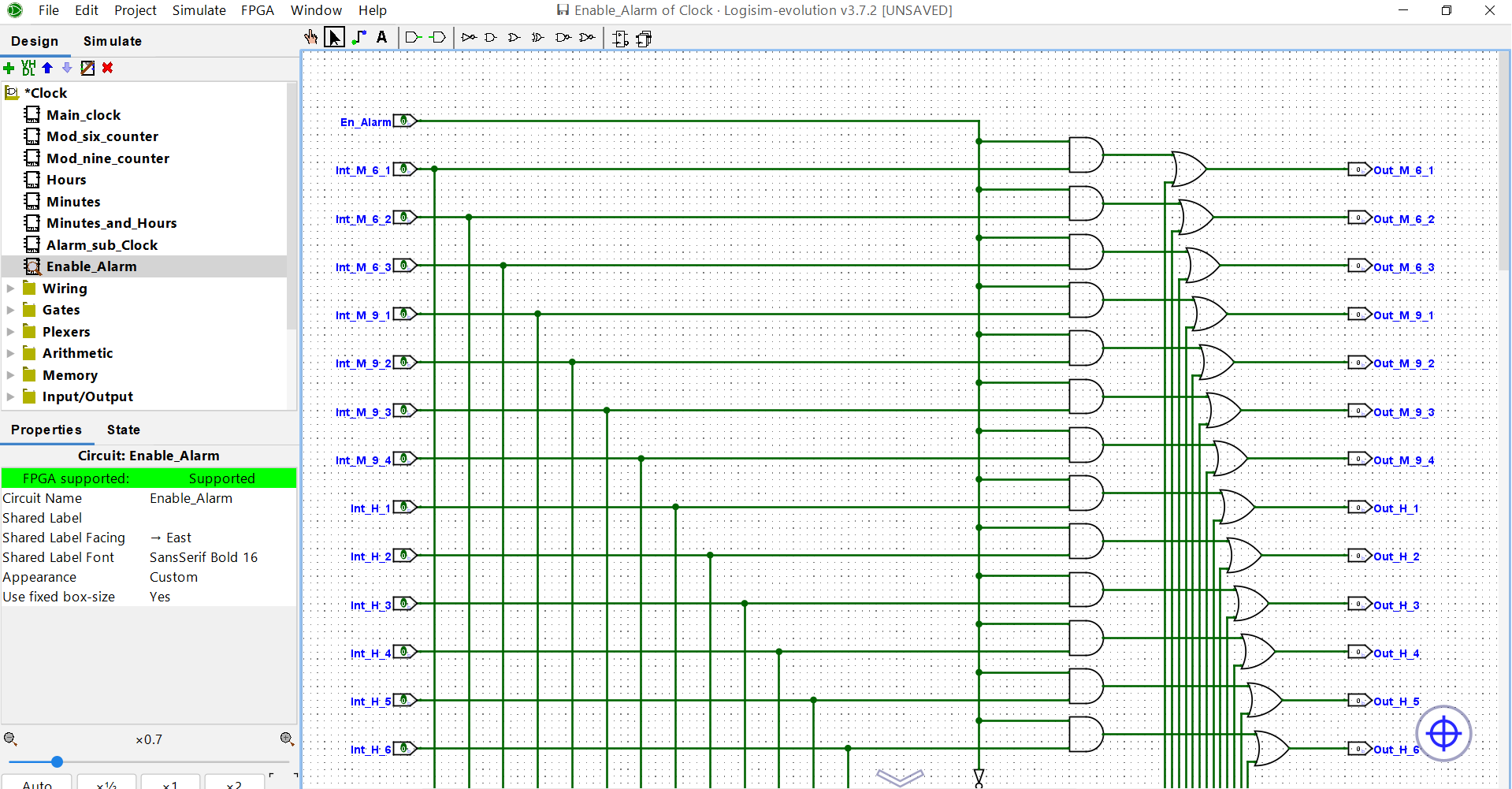
***How does the “Set Alarm” mode work?***

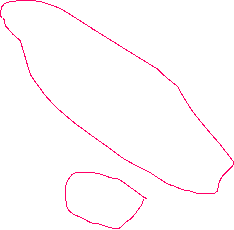
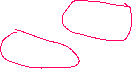
We have two functional sub-circuit to make this work. In the main circuit, we already have the “Mins and Hours” circuit to process the inputs and display time. As I have mentioned, there is a Sub Clock named “Alarm Sub Clock”, which almost has the same inputs and outputs, that running at the same time when “Set Alarm” is disabled, but only “Mins and Hours” outputs can display in the 4 hex digit displays and the AM/PM LED. When the “Set Alarm” button is enabled, the two clocks will stop, displaying responsibility is now taken by the second alarm sub clock, and when the users use the “M+” and “H+” button, they can only make changes in the second sub clock, due to some specific following AND and NOT gates:





And because the first clock has not been changed during the “Set Alarm” mode when this button is off, the first clock is now responsible for time display, and so the time will continue from before this mode.

Now is the task to decide if the Alarm LED turns on or not. After the “Set Alarm” mode is enabled, the first clock circuit remains the values of the present time, while the second one remains the values of alarm time. I simply make a sub-circuit to compare the values between these two clocks. First, this circuit gets the values from the two mentioned clocks, then compares each pair of bits by an XOR gate (similar values mean 1, different values mean 0), when all results of these XOR gates are similar, which means it’s now the alarm time, ensuring by a AND gate with 13 inputs, the LED now should be turned on, and it is also connected with a JK FlipFlop to remain the LED even after the time has passed the alarm time. The status of “Enable Alarm LED” is also important, so I “AND” it with the 13-input AND gate’s result, to make sure the LED turn on and off validly.



***Unresolved problems***

In the “Mod 6 Counter”, “Mod 9 Counter” and “Hours” circuits I cannot be able to use the buffers (D Flipflop, or synchronous counters). It is because when I try to build a counter with these buffers in my counter, there is a delay of 1 clock pulse. The “Mod 9 counter” is ok, but the problem rises when it turns to the “Mod 6 counter” when the first hex digit display runs from 0 to 9 and returns to 0, the second display will not turn to 1 immediately, but it will take another cycle. The problem is similar to the section of “Hours”.

**Conclusion**

In recent weeks, I have made a great deal of effort to complete Assignment 1. However, due to my limited ability, there are some issues that I cannot solve at the moment I write this report. This is a disadvantage that is quite serious for me in the future assignments of this unit, so in the following weeks, I should focus more on training and improving my knowledge and skills, to perform well in the next lab sessions, as well as the exams.

Thank you for taking the time to read this report, if you have any questions about it, please contact me by sending an email to the address: [104053642@student.swin.edu.au](mailto:104053642@student.swin.edu.au)

**References**

*[1] Lectures and recordings from week 1 to week 5*

*[2] Lab Instructions from 1 to 4*