MINISTERUL EDUCAȚIEI NAȚIONALE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

TIMER

STUDENT: COMAN ALECSIA COSTINA

GROUP: 30413

PROJECT SUPERVISOR: TIMAR MIHAI

TABLE OF CONTENTS

1. SPECIFICATIONS

The timer has four 7-segments displays. The first two screens are for the minutes, while the other two are destined to show the seconds. So, the maximum value that can be displayed on the screens is: 99 minutes and 59 seconds.

The device has three buttons: START/STOP, M (for minutes) and S (for seconds).

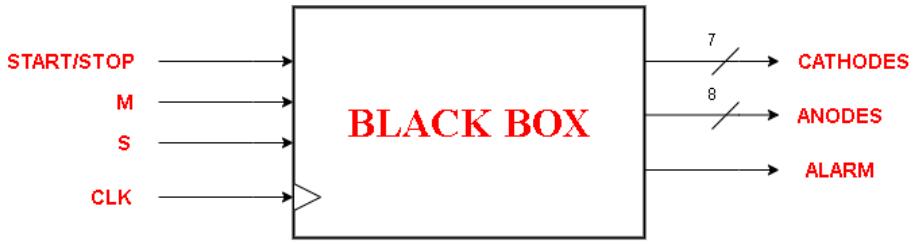
Considering that the initial state is IDLE (00:00), if the START/STOP button is pressed, the timer starts counting in ascending mode. If the START/STOP button is pressed again, the timer stops at the current value displayed on the screens. Then, at the third press of the START/STOP button, the timer continues its counting. If it reaches 99 minutes and 59 seconds, it returns to the initial state. If the buttons M and S are pressed simultaneously, the device resets (goes to state IDLE).

In any state, if the button M is pressed, the value of minutes will be incremented and displayed. This function is also applied to the button S. After the value is incremented for minutes and/or seconds, when the START/STOP button is pressed, the timer starts counting in descending mode, until it reaches state IDLE. After count-down, at the initial state, an alarm is issued, which in our case means lighting up a led.

It is considered available a periodic signal with frequency of 1Hz.

2. DESIGN

2.1. BLACK BOX



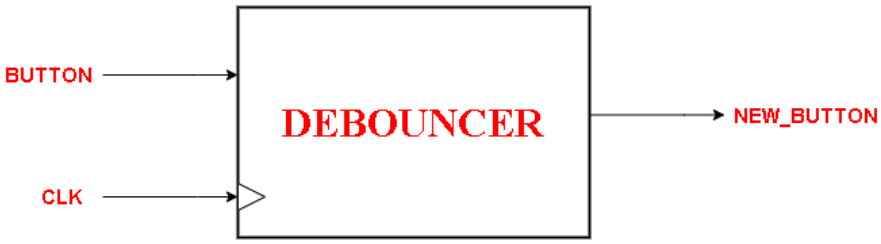
The timer's black box has as inputs the device's buttons and the clock. The initial clock is the board's one, of 100MHz, that is assigned to some of the components, more detailed in the next schemes.

As outputs, the device has the 7 cathodes that control the segments from a display, the 8 anodes that control which display to be active and receive the cathodes and an alarm, a led, which is destined to be turned on when the counter reaches state IDLE, "00:00", after count-down.

2.2. CONTROL AND EXECUTION UNIT

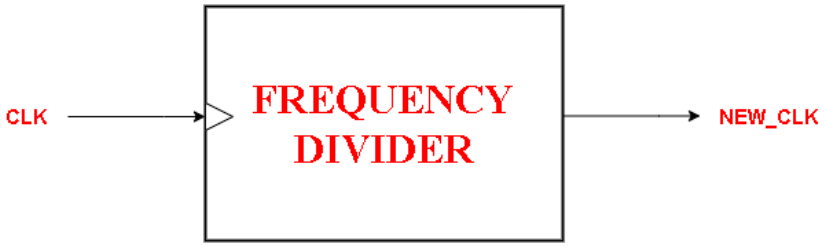
2.2.1. RESOURCES

- DEBOUNCER



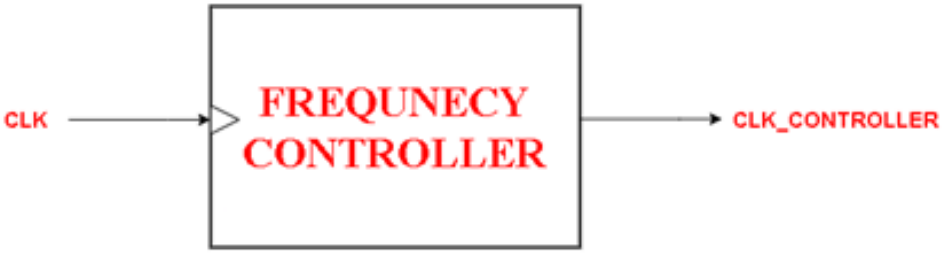
Physical buttons are mechanical devices with bouncing property. This means that when the button is pressed, it bounces from low to high/ high to low a few times, before it settles a proper output. In order to avoid this, the debouncer recieves the button's signal and emits an output by modifying the initial signal trhough flip-flops. This circuit is used for all the 3 timer's buttons.

- FREQUENCY DIVIDER



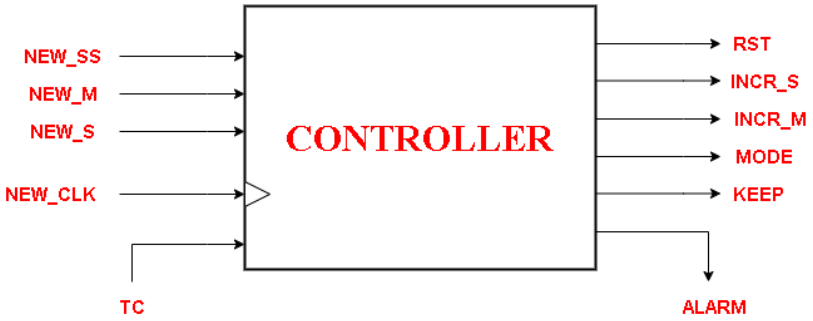
A frequency divider is a module that reduces the frequency of a signal. In this case, the Nexys4 DDR board has a 100MHz clock signal, while the timer has assigned a periodic signal with frequency of 1Hz. This component receives the board's frequency and emits a "new clock" that will be sent to the Counters.

- FREQUENCY DIVIDER FOR CONTROLLER



Different actions happen by pushing the buttons. This "pressing" is verified by the Controller component (explained next). In order to find out when the buttons are being pressed, the Controller should receive a faster clock, which would verify at a period of half a second this change. The "Frequency Controller" imitates the previously described component, the only exception being the higher frequency.

- CONTROLLER



The "Controller" represents the State Diagram of the Control Unit. This component sends signals to counters, depending on the current state of the device. It receives the debounced button's signals, the divided clock from "Frequency Controller" and the cascaded counters' terminal count (it "tells" the controller that the counters have finished counting in ascending or descending mode). Through **RST**, the counters are set to 0. By pressing the button **S** or **M**, **INCR_S** or **INCR_M** are activated, which will increment the seconds/ minutes as many times as the user presses the buttons. **MODE** decides if the counting is ascending or descending (0 – up, 1 – down) and **KEEP** maintains the current value on the displays, when it's high. **ALARM** is linked to a led on the board, instead of an audio alarm, as stated in the specifications.

- COUNTER S0

The Counter_S0 is responsible for the unit second. It receives inputs from the Controller and the divided clock signal, of 1Hz. If **RST** is active, the counter goes to the initial state, 0. In any mode, ascending (**MODE** = '0') or descending (**MODE** = '1'), if **INCR_S** is active, the seconds will be incremented with one. **HOLD** input, if it's '1', maintains the current value until future actions. As outputs, this counter activates terminal count, **TC**, in two cases: the number (**NR**) is 9 and **MODE** is '0' or the number (**NR**) is 0 and **MODE** is '1'. Also, the output **NR** contains the binary number of this counter, which will be ulterior concatenated to a final array of 16 bits.

- COUNTER S1

Counter_S1 is responsible for the tenth second. It has the same behavior as Counter_S0 and almost all its inputs. All 4 counters are synchronously cascaded, but each of them counts only if the ones before reached final states and have the terminal count active. So, this counter also receives the Counter_S0's **TC**, which allow it to count or increment the seconds if needed. It will increment only if **INCR_S** is active and the counter before has reached 9 (**TC_S0** = '1'). This component counts only until 5, since the timer's final state in ascending mode is 99:59. As outputs, **TC** activates if: **NR** = 5 and **MODE** = '0' or **NR** = 0 and **MODE** = '1', and **NR** transmits the current counter's number in binary.

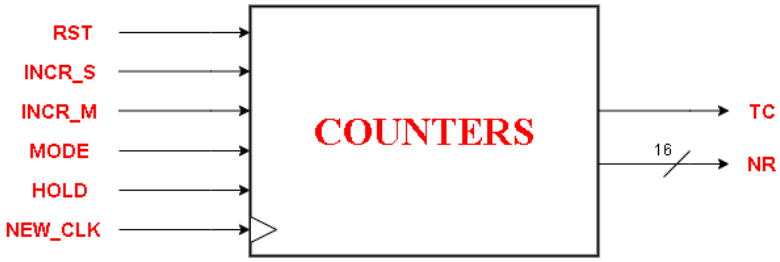
- COUNTER M0

The Counter_M0 is responsible for the unit minute, counting until 9. It has the same behavior as the previous counters, functioning only if the counters before have reached 9, respectively 5, in ascending mode, or 0, in descending mode. If **INCR_M** is active, the minutes are incremented as many times as the button **M** is pressed by the user. The minutes must increment also if the seconds counters reach the final state, 59. This is done by receiving the **INCR_S** input, together with **TC_S1**, which is active if both Counter_S0 and Counter_S1's terminal counts are '1'. The outputs are the same as the ones presented before.

- COUNTER M1

Counter_M1 works as the previous presented components. It represents the tenth minute, counting-up until 9 or down until 0, only if the other three have the terminal counts active, visible with **TC_M0**. It also needs a signal that tells the component if Counter_M0 has reached 9 in INCREMENT state, which is represented through **INCR_M0**. This operation must be taken in consideration beside the usual counting, because the minutes can be incremented separately from seconds, as many times as the user wants to.

- COUNTERS



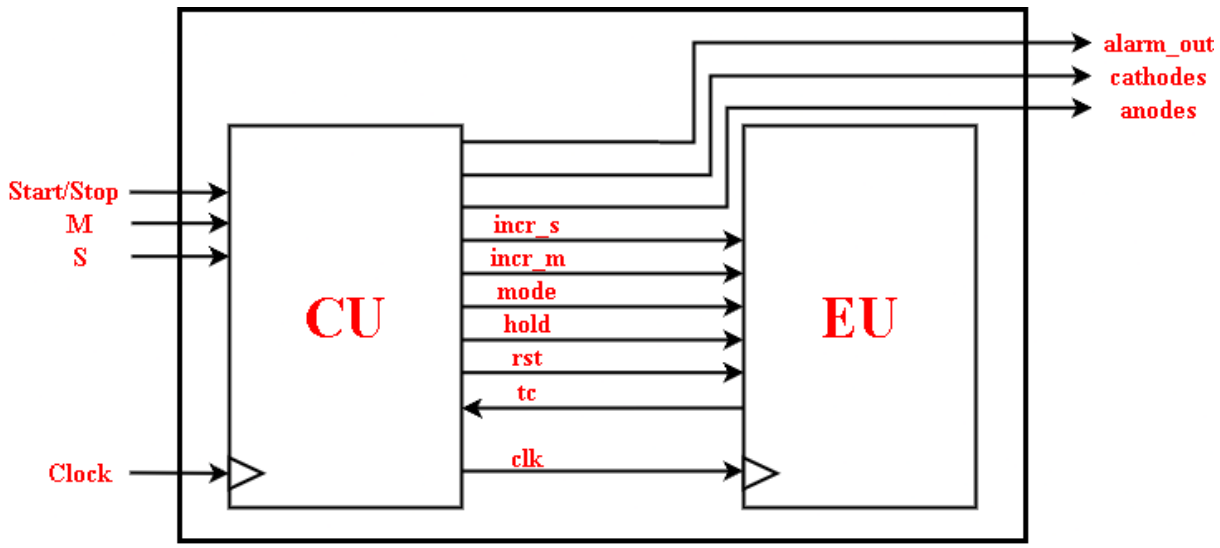
This component contains the four counters presented above, cascaded. This means that all of them work on the rising edge of the divided clock, with the frequency of 1 Hz. It receives the Control Unit's outputs and sends them to each counter, depending on their needs. As outputs, **TC** is '1' when all counters have their terminal counts active. This means they reached 99:59 in ascending mode or 00:00 in descending mode. This signal is sent back to the Control Unit, "telling" it that the timer has reached a final state. **NR** output is a 16 bits array, containing all the seconds and minutes in the following order: M1M0S1S0. This number is sent to the Display component (explained next) to be converted as cathodes.

- DISPLAY

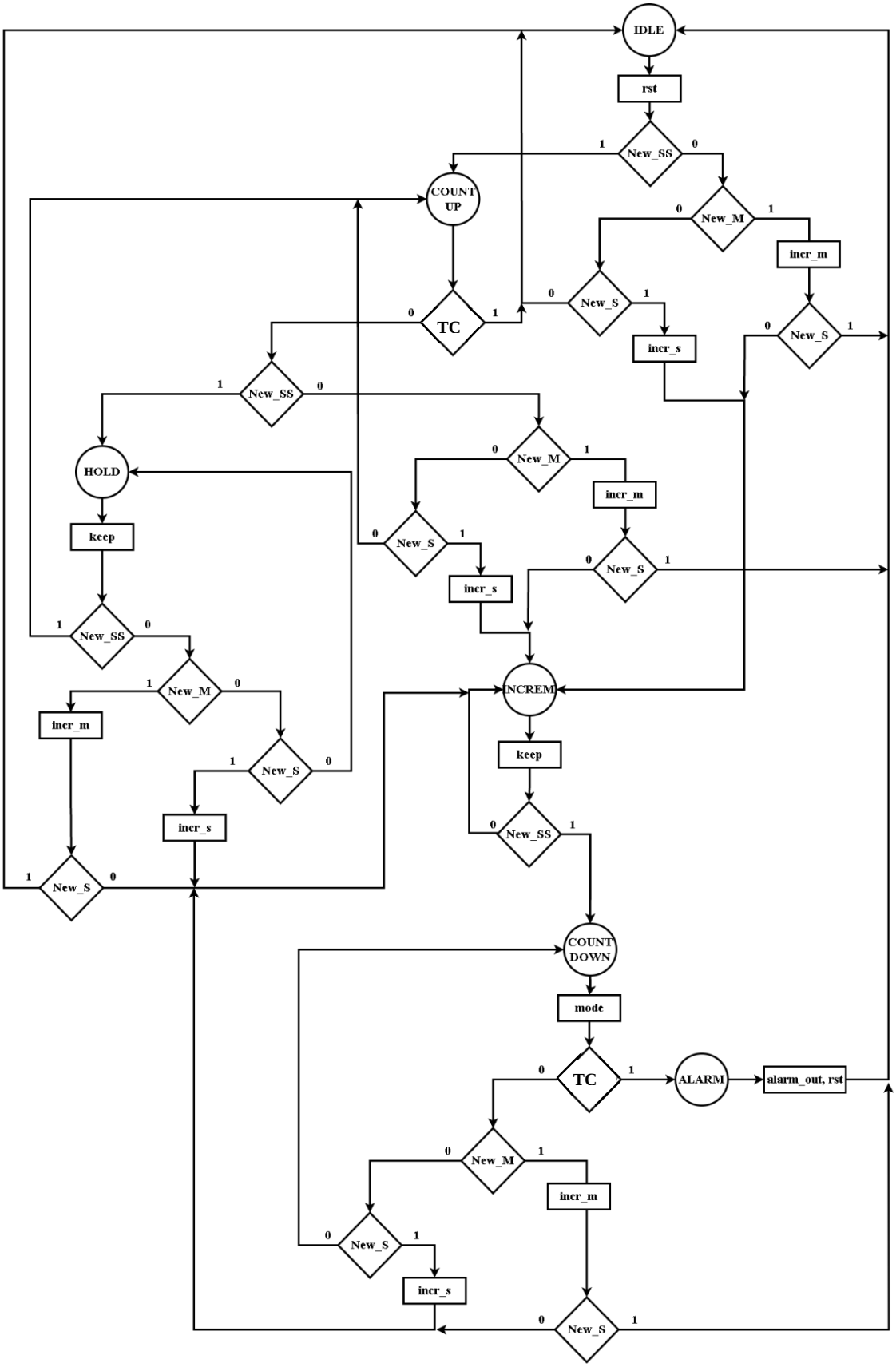
The Display is a very important component, and the last one, that will show the time on the SSD. These displays cannot be simultaneously light up. This is the reason we need a 100 MHz clock signal, namely **CLK**, to create the illusion that the screens are lit up concurrently. In fact, they will actually be lit up one at a time, but at a very fast rate, which can't be detected by the human eye.

The Display receives the 16 bits vector, which holds the values of the concatenated minutes and seconds and they will be shown on the 7 segments display. The output **ANODES** represents the number of SSD used, 8 (the first 4 are always kept off) in this case, while the output **CATHODES** states for the number of segments on each of the screens, respectively 7. For the display to know what segments to light up, it needs a set of instructions. For example, to display '4', because the cathodes are active low, the vector should be "1001100".

2.2.2. BLOCK DIAGRAM FOR FIRST BREAKDOWN



2.2.3. STATE DIAGRAM OF CONTROL UNIT

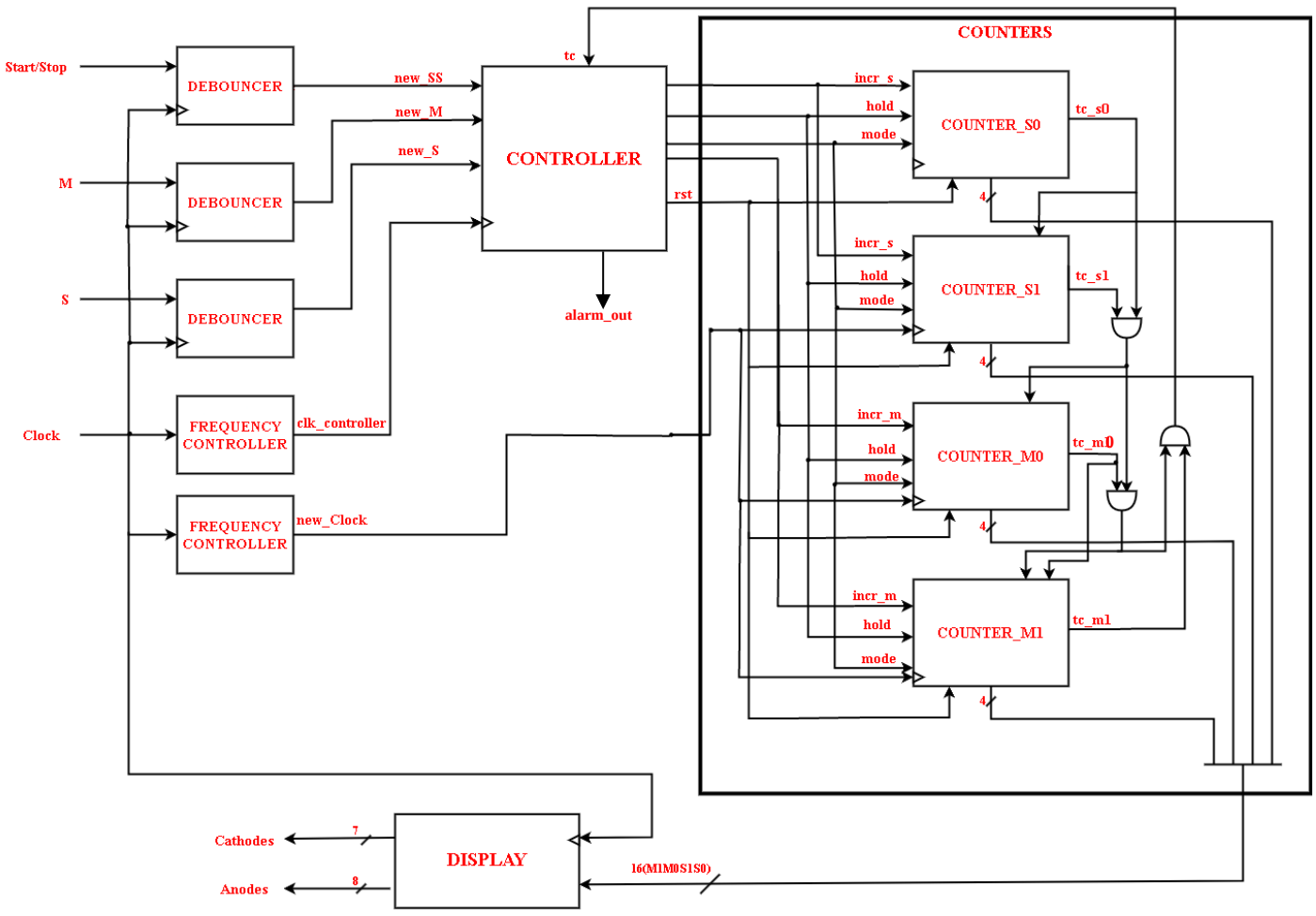


The above scheme stands for the state diagram of the control unit. In the project, the state machine is represented through the "Controller" component. Each circle represents a state. The timer has six: IDLE, COUNT-UP, HOLD, INCREMENT, COUNT-DOWN and ALARM. The control unit, as it has been seen at point 2.2.2, sends to the execution unit the following signals: rst, incr_s, incr_m, mode, hold and a divided clock. Each of them is initially set to '0'. So, on the diagram, the rectangles represent activating the respective outputs, set them to '1'. The triangles state for the asynchronous inputs, the debounced buttons: New_SS (Start/Stop), New_M (M), New_S (S) and the terminal count, TC, sent by the EU. Depending on their value, '0'/'1', a new state is following.

The outputs activate in the next situations:

- Rst – when current state is: IDLE or ALARM, or when **M** and **S** are pressed
- Incr_s, incr_m – when **S**, respectively **M**, is pressed
- Mode – when it is 1, the counters count down by themselves, else, they count up; Though, when the time is incremented, the mode doesn't matter anymore, since incrementing has priority
- Hold – when current state is: IDLE, HOLD, INCREMENT; it keeps the value on SSD

2.2.4. DETAILED SCHEME OF THE CIRCUIT



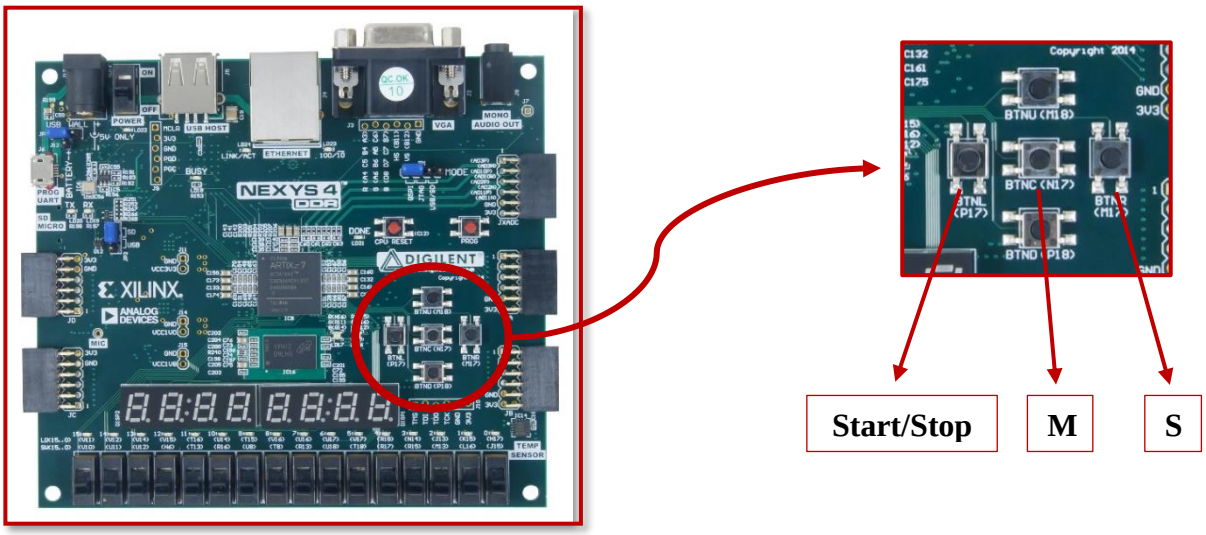
The presented scheme encapsulates all the components of the device. It can be seen that the initial buttons' signals are transmitted through three debouncers. The initial clock, of 100Mhz is divided to 1Hz by the frequency divider and it is sent to the counters. The controller gets a higher frequency in order to verify faster if the buttons are being pressed. The four counters, for unit seconds, tenth seconds, unit minutes and tenth minutes are cascaded synchronously. The counters component, that contains the four elements mentioned before, send back to Control Unit the terminal count and to display, the 16 bits vector. The display is the final component of the device, transmitting the cathodes and anodes to the SSD.

3. USER MANUAL

The timer's purpose is to display, as its name suggests, the time. It can count-up, from 00:00 to 99:59, and down. Initially, when the device is powered on, on the display will be shown 00:00, meaning that the system waits for a command.

There are only three buttons that can be used. **Start/Stop** is situated on the left. If it is pressed, it starts the counting. If it is pressed again, while the timer counts-up, the current time is being hold, until **Start/Stop** is for the third time. **M** is situated in the center. It can be pressed in any state and it will increment the time with one minute. **S** has the exact same function, but for the second. The time can be incremented as much as the user wants to. After incrementation, if **Start/Stop** is pressed, the timer starts counting-down until it reaches 00:00. It can be seen that after count-down, the right-most led is being light-up. It can be turned off only if **Start/Stop** button is pushed. The device can then be reused.

The timer can be anytime restarted by pressing simultaneously **M** and **S**.



4. TECHNICAL JUSTIFICATION FOR THE DESIGN

The reason I chose to implement the described design is because of its simplicity and good performance assurance. It uses easy to understand and apply components, as the debouncer, frequency divider and counters are. I chose to use a counter for each second/minute, because it was more accessible to verify the outputs and functioning. Each component was created and tested separately and then combined to form the final device. This way, the operations can be easily understanded.

5. FUTURE DEVELOPMENTS

For future developments, the timer could:

- Show the hours, by adding an extra button, like **H**, and using additional displays
- Hold the current value when it is in COUNT-DOWN state, not only in ascending mode
- Use an audio alarm
- Set from an external hardware the value for time, M1M0:S1S0, instead of repeatedly pressing the buttons
- Let the user choose if, after incrementing, the device should count-up or down

6. REFRENCES

https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual

https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Electronics/Microwave_and_RF_Design_IV%3A_Modules_(Steer)/06%3A_Mixer_and_Source_Modules/6.08%3A_Frequency_Divider

https://www.fpga4student.com/2017/09/vhdl-code-for-seven-segment-display.html

https://surf-vhdl.com/vhdl-syntax-web-course-surf-vhdl/vhdl-structural-modeling-style/