CT1 Report

|  |  |
| --- | --- |
| Module | EE6621 ASICs(Digital ASICs) |
| Project Number | CT1 |
| Name | Jufeng Yang |
| ID | 20125011 |
| Data | 5/12/2021 |

**Declaration of authorship: I confirm that this lab report, submitted for assessment, is my own work**

1. **Project Features Overview**

|  |  |
| --- | --- |
| Modes & Functions | Progress States |
| Show ‘UL’ for 1s | Successful |
| Show ‘ECE’ for 1s | Successful |
| Show ‘UL ECE’ for 1s | Successful |
| Show ‘EE6621’ for 1s | Successful |
| Show ‘ct1’ for 1s | Successful |
| Show ‘125011’ for 2s | Successful |
| Stop Mode | Successful |
| Start time initial: ‘0325’ | Successful |
| Alarm Mode | Half Successful |
| Counter Down | Successful |
| Set Mode | Half Successful |
| RIGHT (move to right digit) | Successful |
| LEFT (move to left digit) | Successful |
| UP (add 1) | Successful |
| DOWN (minus 1) | Successful |
| ESC | Successful |
| OK | Successful |
| Reset Function | Successful |

1. **Code sources**

|  |  |
| --- | --- |
| Modules | **Source** |
| bcd\_counter\_4d | Adapted |
| bcd\_counter\_digit | Adapted |
| buzzer | Re-use |
| clkgen\_cmod\_a7 | Re-use |
| counter\_down\_rld | Re-use |
| ct1 | Adapted |
| debounce | Re-use |
| display\_7s | Re-use |
| display\_7s\_mux | Adapted |
| fpga\_wrapper\_ct1 | Adapted |
| fsm\_game | Adapted |
| mbutton | Adapted |
| Synchronizer\_3s | Re-use |
| fsm\_game\_states | Adapted |
| timing | Re-used |
| wordlength | Re-used |

1. **Project Specification Requirments**

The design and code implementation of this project is based on rt2. By adding some modes and functions to rt2, it will be possible to implement CT1. From the interface point of view, the interface of CT1 is the same as rt2 and all additional functions are implemented inside CT1. Some of the modules of the CT1 design are changed accordingly to meet the functionality (e.g. countdown timer, buzzer). From a functional point of view, CT1 includes "show" functions, stopped mode, set mode, countdown mode and alarm mode. Arranging these modes in a certain order will result in the correct display process. The ideal procedure would be to display 'UL', 'UL ECE', 'EE6621', 'ct1', '125011'. Then enter stop mode and display the initial value. Then press PB\_SS to enter countdown mode and press to write PB\_SET to enter setup mode. In countdown mode, every second the number will be subtracted by 1. Then when all the numbers are 0, the alarm mode will be entered and the buzzer will alarm. During this time, pressing any button resets the timer and enters Stopped Mode and the timed digits are reset to the value set afterwards. If nothing is done during the alarm, the alarm will stop automatically after 10s and return to the set value. In set mode the digit being set will flash, pressing PB\_SS will add one to the digit and when the digit is greater than 9 it will change to 0. Pressing PB\_SET will change the digit being adjusted and when all four digits have been adjusted it will enter Stopped Mode. pressing PB\_SS and PB\_SET at the same time will reset the whole system.

In addition to this, we have additional functions. In Set-up mode, the start value of the timer is adjusted by means of 6 buttons. The four buttons up, down, left and right make setting mode even easier.

1. **Specification for Advanced Project A**

|  |  |
| --- | --- |
| Button | Function |
| ESC | The ESC button is used to exit the current setup mode and enter S\_STOPPED mode. The currently changed value does not become the initial value, but when entering the setting mode again, the modified value is displayed. |
| OK | The OK button is used to confirm that the current value is set to the initial value of the timer. |
| UP | UP button to add one to the current number |
| DOWN | The UP button subtracts the current number by one |
| LEFT | Left is used to change the number of digits being adjusted to the left one |
| RIGHT | RIGHT is used to change the number of digits being adjusted to the right one |

1. **Implement Schematic and FSM Diagram**

**FPGA Project wrapper schematic.**

图示

描述已自动生成

***Figure.1: Schedule of CT1 Project.***

As can be seen in Figure 1, the package module combines clkgen\_cmod\_a7 and ct1. The inputs and outputs are shown in the figure. pio22 is the input signal for the multi-button, and the reset signal for ct1 comes from button1 & button2. buttton[1:0 is also the input signal for ct1. clk\_raw\_in is the input to the clock generation module. ct1's output signals are blink, buzzer\_p, d7\_anodes, d7cathodes respectively.

**CT1 Wrapper Schematic:**

图示, 示意图

描述已自动生成

***Figure.2: The whole Schematic Diagram of CT1.***

The main functions of the CT1 project are implemented in the CT1 module, which also contains a large number of sub-modules, the inputs and outputs of which are logically connected to achieve the correct functional display.

As can be easily seen from Figure.3, there are three timers in CT1, 100us, 1ms and 1s. 100us is used for the implementation of the buzzer function, 1ms for the time control of fsm\_game and 1s for the minimum timing interval of the countdown. The debounce function for the two buttons, debounce\_l for button1 and debounce\_r for button0. The display\_7s\_mux module takes as input the LED display value option returned in fsm\_game and outputs an 80 bit register variable corresponding to the LED display value of the option. Connect the output of this module to display\_7s. This module analyses the 80 binary bits of the variable, parses out the binary bits for blinking and digit decoding and displays these characteristics in the LEDT. bcd\_counter\_4d is the module used for timing, with four one-bit timers integrated in 4d. Combining the four one-bit timers completes the timing of the four bits. The most important is the fsm\_game module, which is used to arrange the display of the LEDs and the transition between modes when the button is pressed. These modules will be explained in detail later in the presentation.

**FSM Diagram:**

The details of the FSM diagram are shown in Figure 3. The FSM diagram shows all the states that the project will likely have to appear in, and the relationships between the different states. As can be seen in Figure 3, the first state is the S\_Reset state, which will be entered after pressing both fpga's buttons at the same time. After that the system will automatically enter the S\_SHOW\_UL state, then the S\_SHOW\_ECE state, then S\_SHOW\_MODULE, then S\_SHOW\_DESIGN, then S\_STUID and then S\_INITIAL. in the initialised state the initial value of the timer is initialised and the system will automatically jump into the S\_STOPPED state. S\_STOPPED mode is the key mode for the whole system, pressing button[0] in this mode will put the system into setup mode. Pressing button[1] will enter S\_RUN mode. Set mode contains four states, each of which controls a digit, and the size of the digit is adjusted by the additional buttons UP and DOWN, and Left and Right control the state in set mode. The system can be returned directly to S\_STOPPED mode regardless of the state of the system in SETUP mode. s\_RUN mode will enter S\_STOPPED mode if the timer is stopped. If the timer ends, the system enters the alarm mode and the alarm mode ends in S\_STOPPE mode.

图示

描述已自动生成

***Figure.3: FSM diagram.***

1. **Verification Check-List**

|  |  |  |  |
| --- | --- | --- | --- |
| Features | Fail | Complete | Comments |
| Show “UL ”, Duration is 1s |  | Yes |  |
| Show “UL ECE”, Duration is 1s |  | Yes |  |
| Show “EE6621”, Duration is 1s |  | Yes |  |
| Show “ ct1”, Duration is 1s |  | Yes |  |
| Show student ID last six digit “125011”, Duration is 2s |  | Yes | The timing parameter WAIT\_LONG is 1999. |
| Initialize the timer with the initial value " 0325". |  | Yes |  |
| STOPPED mode, keep "0325" displayed and wait for input. |  | Yes | Press button[0] to enter SET mode , button[1] to enter RUN mode. |
| Press button[0] to enter SET mode and the first digit flashes. |  | Yes | Press UP plus one, DOWN minus one. |
| Press RIGHT and the second digit starts flashing and changes 3 to 0. |  | Yes |  |
| Press RIGHT again and the third digit starts flashing and changes 2 to 0. |  | Yes |  |
| Press RIGHT a third time and the fourth digit starts flashing and keep the value 5. |  | Yes |  |
| Press ESC to exit SET mode and display the initial value "0325". |  | Yes | It can exit in anytime in SET mode. |
| Enter the setting mode again and display the modified value "0005". |  | Yes |  |
| Press OK to enter STOPPED mode and the initial timer value is modified. |  | Yes | It can conform in any time in SET mode. |
| Press button[0] to enter RUN mode.  Show “c xxxx”. |  | Yes |  |
| Press button[1] to pause the timer and enter STOPPED mode. Maintains current digital display |  | Yes |  |
| Press button[1] again to start the timer from the current number. |  | Yes |  |
| After 5s it enters alarm mode, the four digits "0000" flash and a pulse-type alarm starts. | NO | Yes | The alarm is a continuous sound, not a pulse. |
| Press any button during an alarm to stop the alarm and redisplay the modified initial value. |  | Yes |  |
| If no button is pressed, the alarm stops automatically after 10s. |  | Yes |  |
| In any mode, press button[0] and button[1] at the same time to enter RESET mode and reset the system. |  | Yes |  |

1. **Summary of Implement information**

Synthesis: Implement:

图形用户界面, 应用程序

描述已自动生成

***Figure.4: Synthesis and Implement.***

DRC Violations: Timing:

图形用户界面, 文本, 应用程序

描述已自动生成

***Figure.5: DRC Violations and Timing.***

From the compile message above, there are no errors, so the bit file can be generated. We can use simulation to check if the code flow is correct, or use the FPGA to check if the flow makes sense.

The utilization and power:

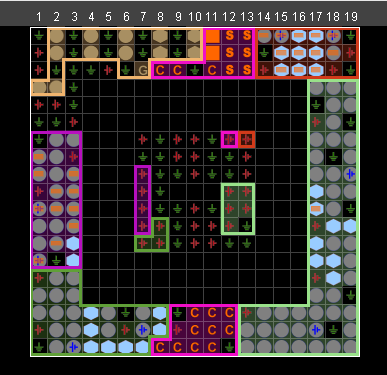
图形用户界面

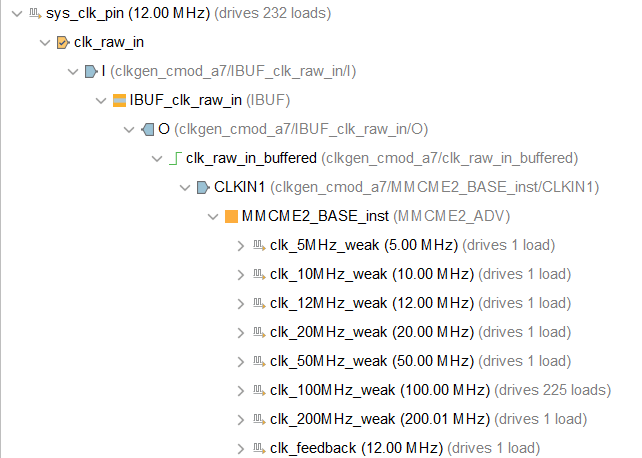
描述已自动生成

***Figure.6: utilization and power.***

Clock information: Package:

电脑萤幕画面

描述已自动生成



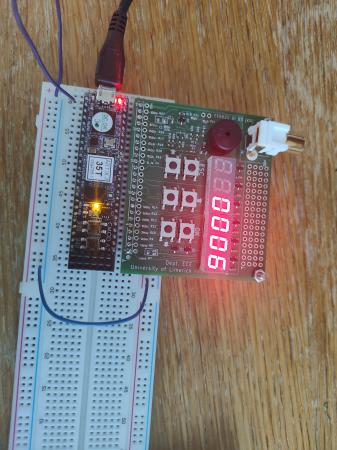
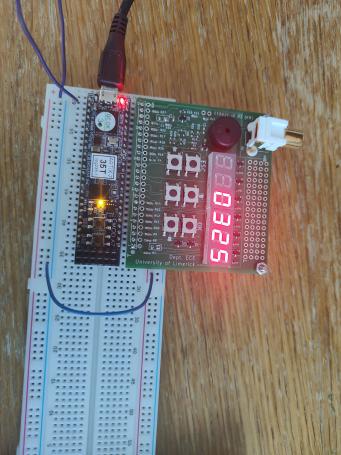
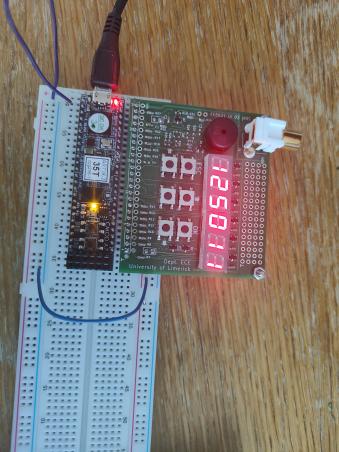
***Figure.16: Clock information.***

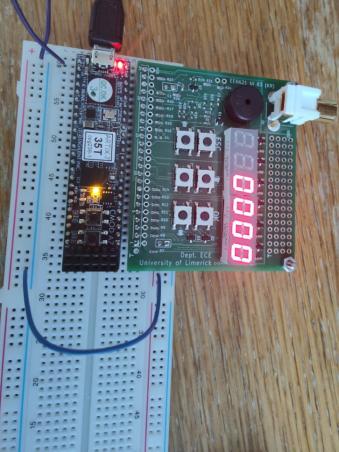
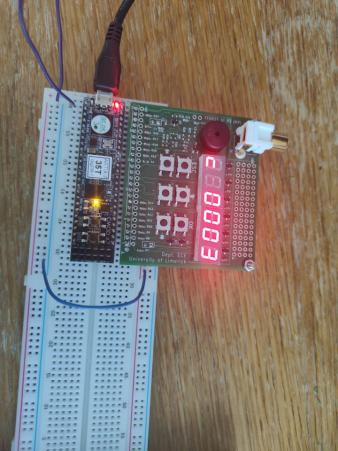
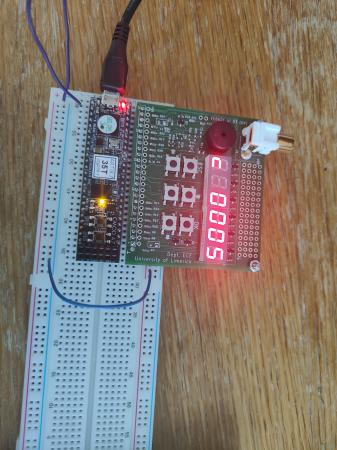
The main frequency here is 120Hz. The other modules are synthesised from 120Hz.

1. **FPGA Test**

After all the compilation is done, we can open the hardware manager and connect to the FPGA and click on tool ->Generate bitstream. After the bit file is generated, the FPGA is programmed. If the logic of the code is in order, it will be ready for operation on the FPGA.

The process diagram for the test is shown in the diagram.





***Figure.17: Test in FPGA***

Part of the process is shown in the diagram. show school number in SHOW mode works fine, initial value shows normal operation. SET mode works fine, set 0325 to 0005, start the timer and when it reaches 0000 in time, the buzzer starts to work. The ‘0000’ is flashing.