**VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY**

HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL ENGINEERING

**Advanced Program**

---------------o0o---------------

****

**MICROCOMPUTER LAB**

**MINIPROJECT 1 REPORT**

**Instructor: Nguyễn Trung Hiếu**

**Students: Lý Chí Học 1810930**

**Nguyễn Thanh Hiệp 1851064**

**Phạm Tấn Khải 1851081**

**Ho Chi Minh City, 5/5/2021**

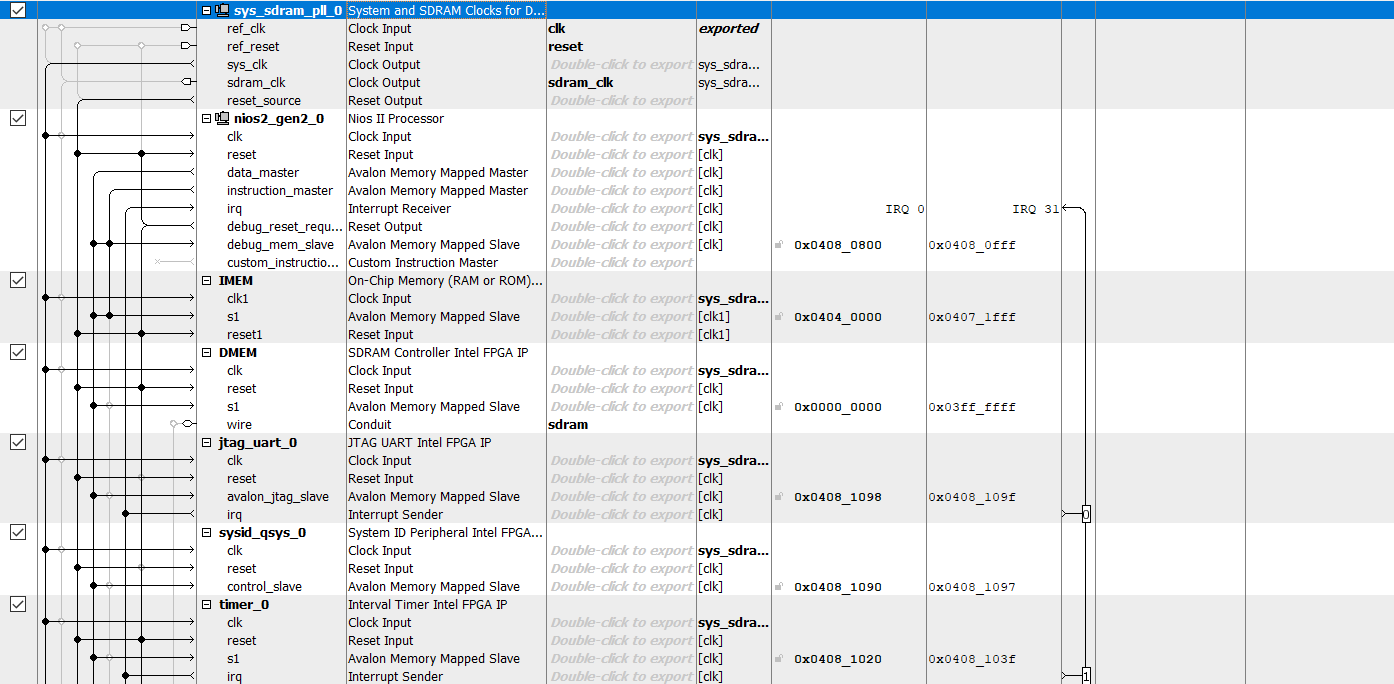
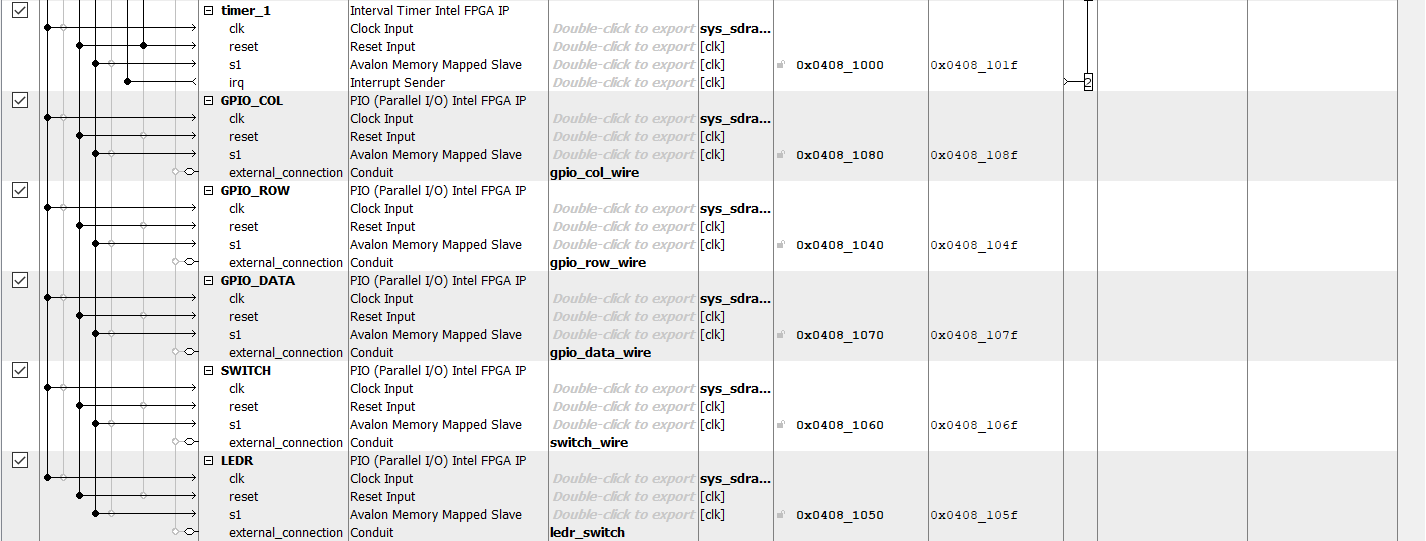
**CONTENT**

1. Platform design………………………………………………………………
2. Quartus………………………………………………………………………
3. Code in Eclipse………………………………………………………………
4. Hardware…………………………………………………………………….
5. References…………………………………………………………………

***Topic:***

Build a Mini calculator using Nios II in kit DE10 to connect LCD 16x2 and Matrix Buttons. - This system can calculate +, -, \*,/ operators.

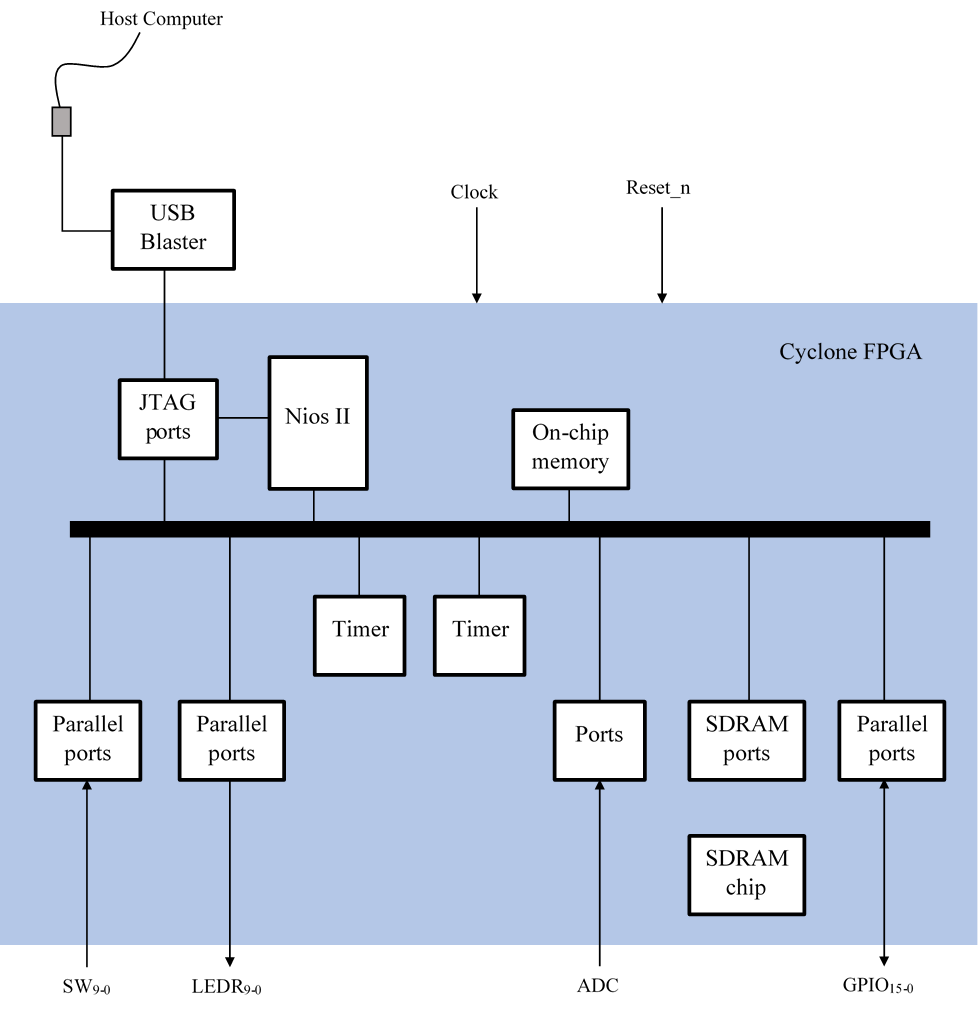
1. **Platform design**

****

We add System and SDRAM clocks, Nios II Processor, On-Chip Memory, SDRAM Controller, JTAG UART, System ID, 2 Interval Timer, 5 PIO ( 10 ledrs, 10 switchs,

11 bits GPIO to work with LCD; 4 bits GPIO column and 4 bits GPIO row to work with 4x4 Keypad )

**2. Quartus:**



5V &GND from ADC ports

SW (9-0)

LEDR (9-0)

GPIO(32-0)

In this mini project, we build a microcontroller with Havard architecture which means data and program are stored in separate memory. SDRAM peripheral is used to store data memory. On-Chip Memory is used to store program memory.

* Parallel ports (GPIO): used for matrix 4x4 Keypad and 16x2 LCD.
* Timers: hardware block that counts clock sequence (0 - 232 ticks) and generate interrupt requests (alarm) (for future preparation).
* Ports: use A/D Converter ports available on DE10 Standard board to supply a potentiometer with 5V to adjust contrast of LCD.

**3.Nios Eclipse**

1. **LCD component:**

**Set Display Clear**

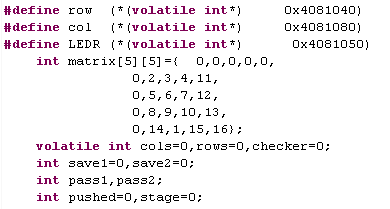
**Send Function set:**

**8 bits, 2 lines, 5x7 Dots**

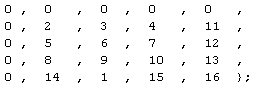
**Send Entry Mode set**

**Send Display Control:**

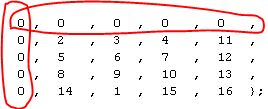
**Display on Cursor on**



The integer array “matrix[5][5]” is for labeling 16 keys on the key-pad from 0->9 and arithmetic keys.



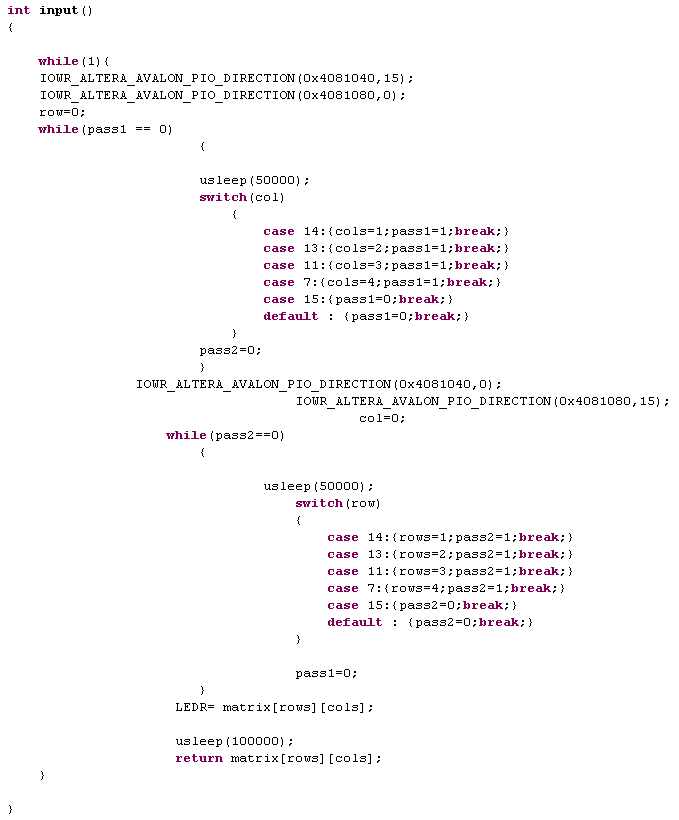
The zeros are for the matrix to filter the state that no button was pressed.



- Because of the algorithm for detecting button would return values for position key 1->16 and there are none that represent the non-pressed state which the “row” and “col” are both have the value ‘15’.

- The coordinates of the keys were increased by 1 (1->16) to distinguish the ‘0’ which is non-pressed state from the numbers ‘0’ then later subtracted with ‘1’ for the ‘temp’ variable further calculations .

- The second reason why the key array declared 5x5 instead of 4x4 is that the returned values for keys are row = [0;4] col = [0;4] whom ‘0’ is for non-pressed button.



The input function use many “while()” for scanning key forward and backward and waiting for keys stroke .

The “while(1)” is for scanning and waiting a value from keys to return for calculation purposes.



The two first lines are for forcing direction to the ports :

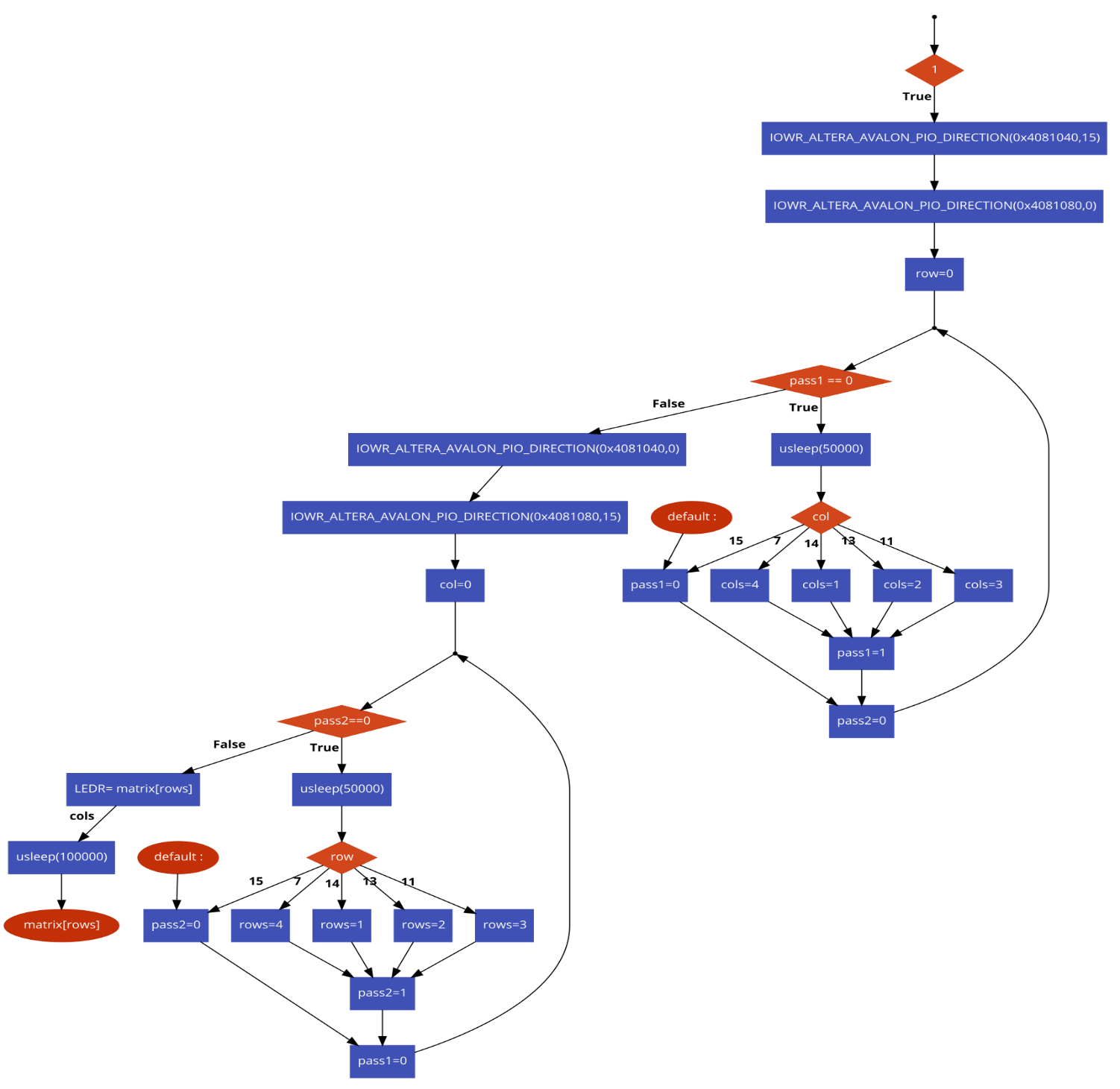
Row is now writing ‘0000’-(pull-up) to column which is reading from row. When keys are stroke the column would receive or read ‘1’ for each keys stroke.



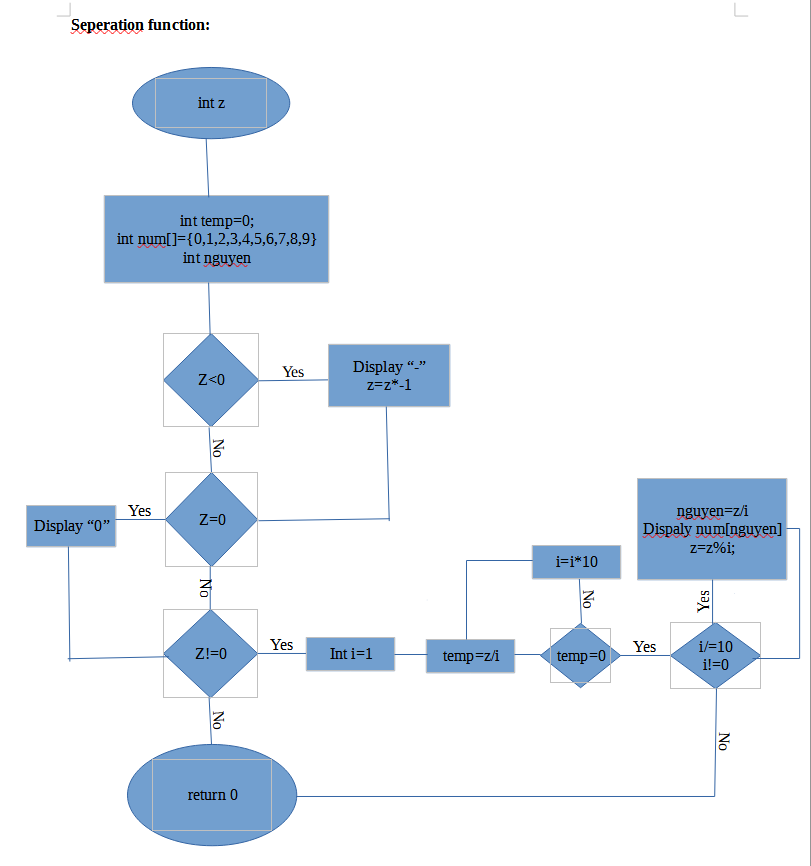
Then the role of the two were switched, row is reading from column written with col=’0000’.

So when no button were pressed the returned value gets a coordinate matrix [0][0] which is ‘0’ and when any keys pressed it would return a coordinate that fit 16 keys

Below is the flow chart of the code :



**b.Seperation Function:**

****

In this function first we check integer z whether its negative or not. If it is negative, we display minus sign on the screen and turn the integer z into positive value.

Next, checking integer z whether its value is zero or not. If it is zero, we display “0” on the screen.

If integer z is not zero and have value more than 0, we create a loop that output of it is i (base 10 of z). Next we decrease the base 10 of z and we extract the number of that base and make z become the remaining of that base. That process repeat until the base 10 of z is 0.

**c.Main Program**

**true**

**False**

**true**

**true**

**False**

**False**

**False**

Set Function: 8-bit, 2 Line, 5x7 Dots

And Display on Cursor on

Clear Display

And Move cursor right by one character twice

Clear Display

And Move cursor right by one character twice

Set an array include number 0-9 and 6 calculation signs in calculator

While loop (1)

There is a number before signs?

Display this number

And save as the first number

There is a calculation signs?

Display this number

And save as the second number

Display this sign

And save value in “cal”

There is a number after signs?

Dot signs??

Clear Display

Reset

Switch (cal) 

Case:10

Case:11

There is “=” sign?

Display “=”

and End line

Case:12

Case:15

Default:

Do Addition

Do Subtraction

Do Multiplication

Do Division

Clear Display

**False**

**true**

**true**

**true**

**true**

**true**

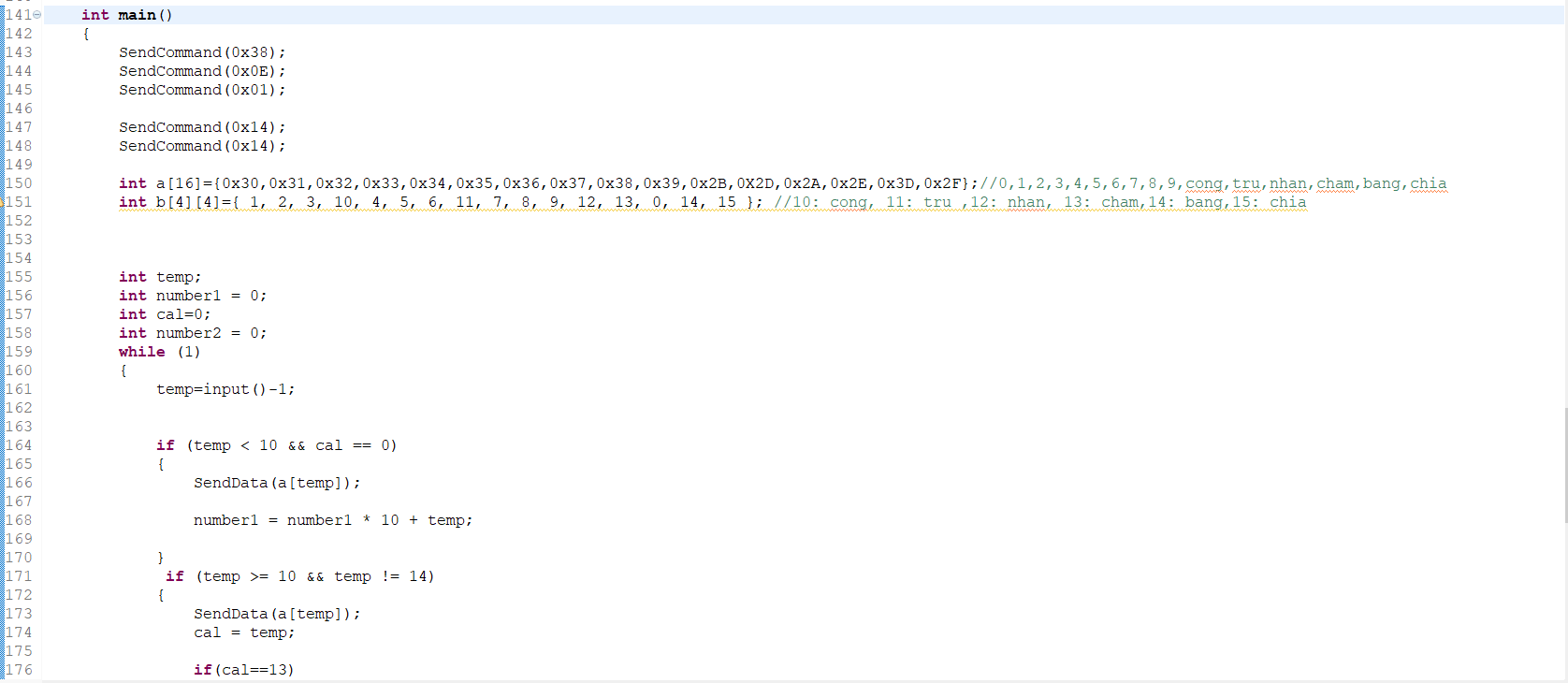
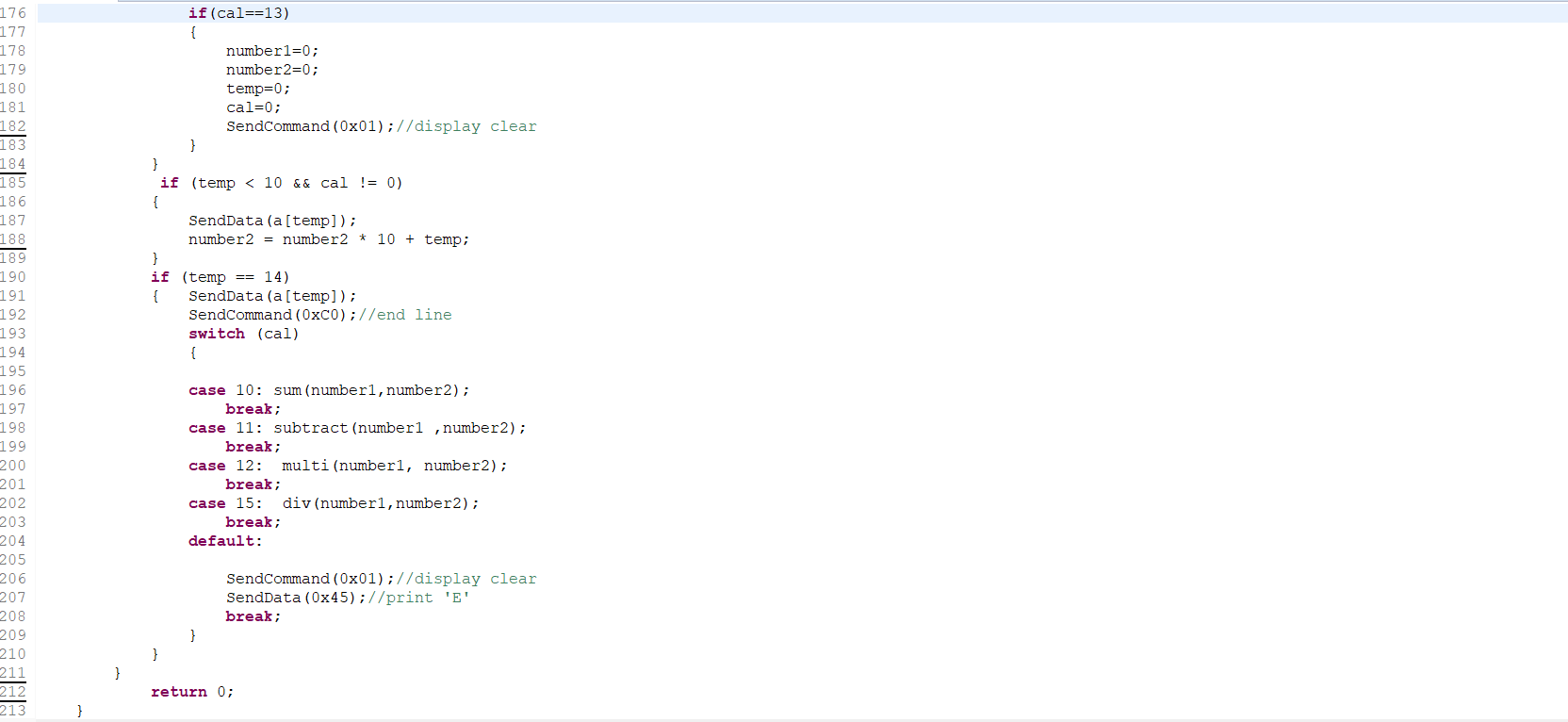
**true**

**False**

**False**

**False**

**False**

****

In main function of this program,

0x38: Function Set: 8-bit, 2 Line, 5x7 Dots. We format LCD display.

0x0E: Display on Cursor ON

0x01: Clear Display (also clear DDRAM content) before working with LCD.

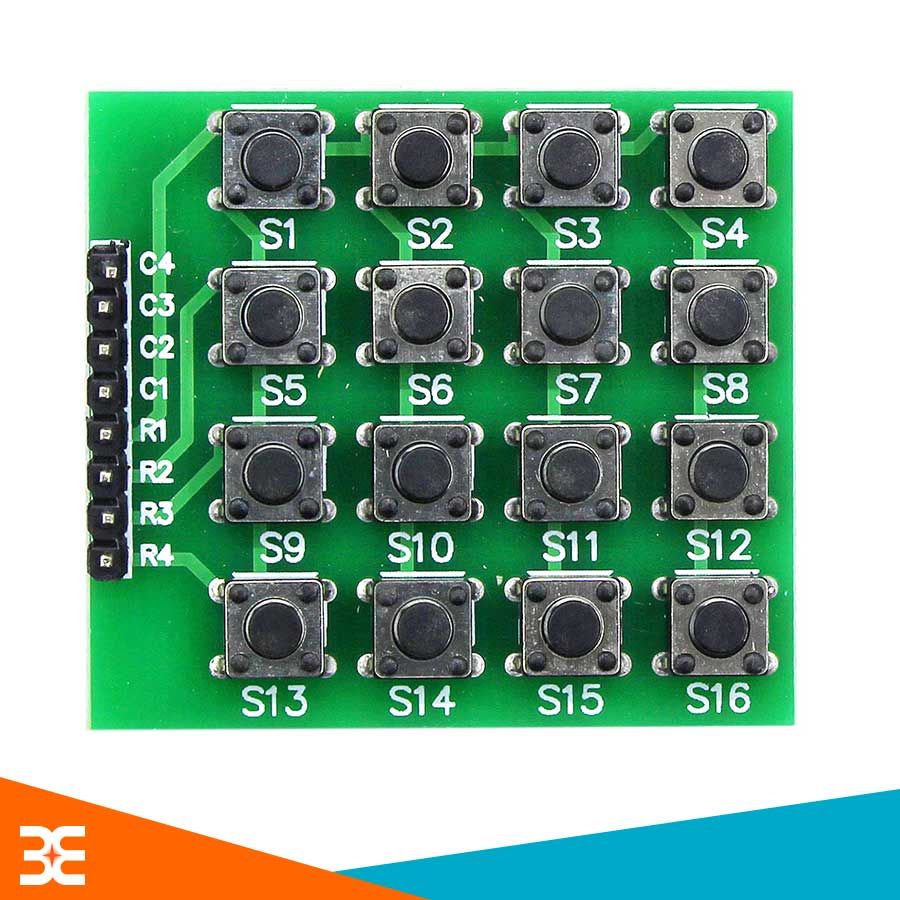
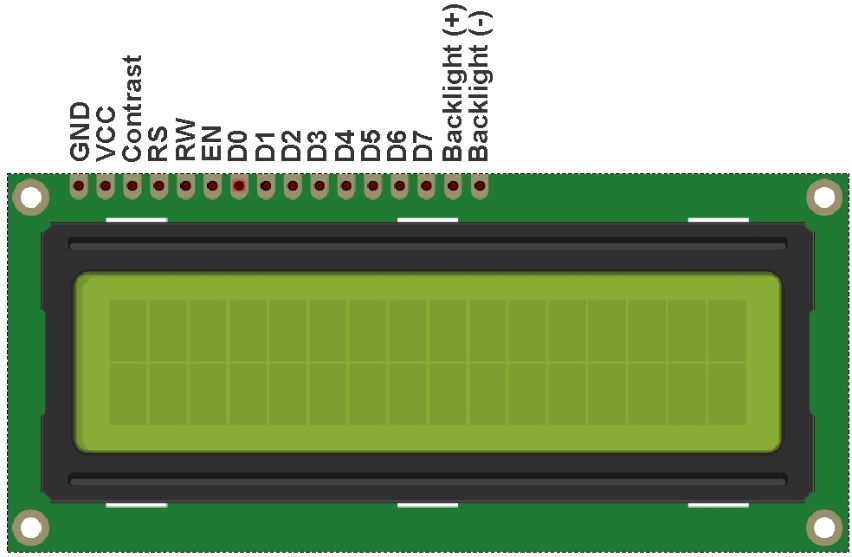
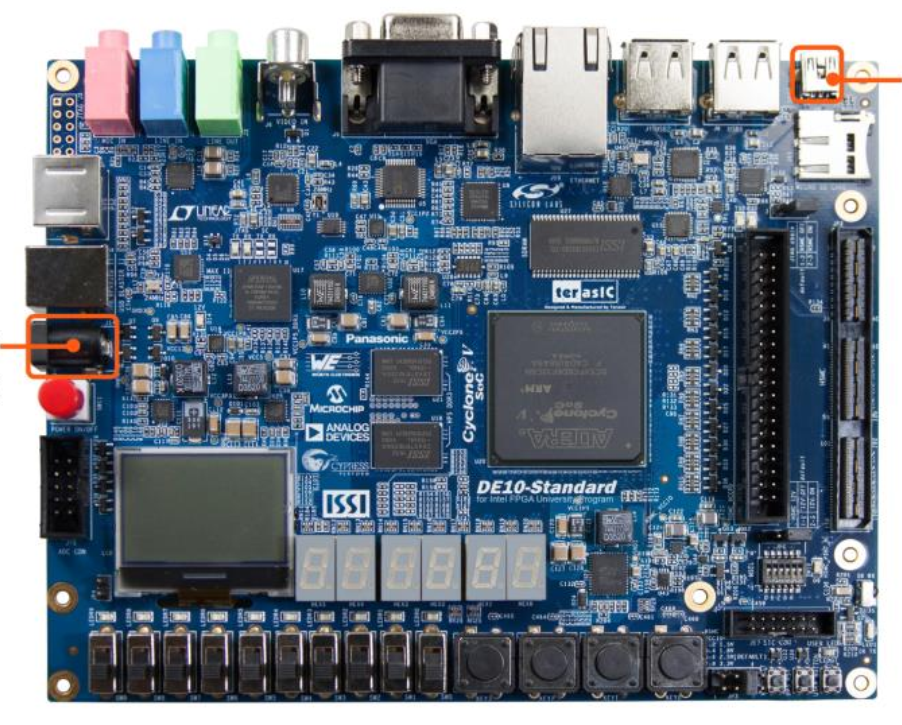
We set an array with 16 value include 9 numbers from 0 to 9 and 6 calculation signs (+ , -, \* , \ , = , . ) in HEX

Next, we set “number1, number2, cal “ =0 (number1 save the first number, number2 save the second number, cal save value that represent a calculation sign)

Then we use “IF” with temp= input()-1 to check if there is a number or a signs?

* “if (temp < 10 && cal == 0)” in line 164 means that if there is a number and no calculation signs => display number and add value to “number1”
* “if (temp >= 10 && temp != 14)” in line 171 means that if there is a sign expect for “=” => display this signs and if there is a dot, we will reset number1,number2,cal=0 and clear display.\
* “if (temp < 10 && cal != 0)” in line 185 means that if there is a number after pressing a sign => display number and add value to “number2”
* “if (temp == 14)” means if pressing “=” => we will end line of LCD and do calculation with Switch-Case loop.

**4.HardWare**



**REFERENCE**

[1] HD44780U (LCD-II) (Dot Matrix Liquid Crystal Display Controller/Driver)

[2] Experiment 9 Parallel Interfacing: Interfacing LCD Display

[3] Cyclone V Nios II Embedded “Hello World” Lab: Cyclone V E FPGA Development Kit