Documentation

Team name

Date

Important Notes:

- ***The descriptions in italics in this document (except for some section headings) are exemplary and explanatory and must be removed from the completed report.***
- ***Identify which section of this report was created by which team member***
- ***Your documentation should have ca. 4-8 pages***

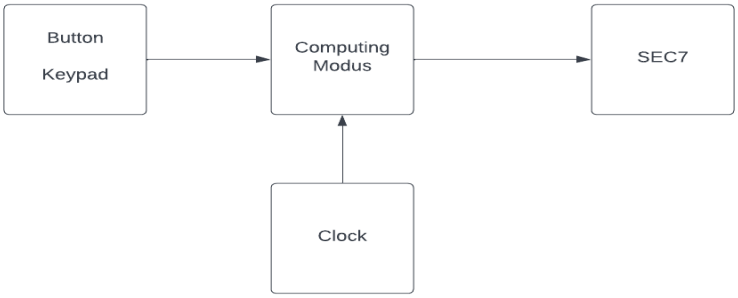
1 Team members

EVRARD LEUTEU FEUKEU

2 Concept/Idea description

Using a 16-button keypad module (Pmod KYPD) connected to the Pmod port, program the Nexys A7-100T board's FPGA to behave as a straightforward hexadecimal calculator capable of adding and subtracting four-digit hexadecimal values.

Block diagram of Hexadecimal calculator.



It functions as follows

1. Input: This block is used to input the hexadecimal numbers that need to be calculated.
2. Hexadecimal to binary conversion: This block is used to convert the hexadecimal numbers to binary, as most digital circuits use binary numbers for computation.
3. Arithmetic and Logic Unit (ALU): This block performs the arithmetic and logic operations on the binary numbers, such as addition, subtraction, multiplication, and division.
4. Binary to hexadecimal conversion: This block is used to convert the binary result back to hexadecimal.
5. Output: This block displays the final hexadecimal result on a display or sends it to external devices.

6. Control Unit: This block controls the flow of data and coordinates the operation of the different blocks in the calculator.

3 Project/Team management

Which project methods you used in your project?

Project Idea 2 was used which was the Implementation of a VHDL Project on the Nexys board based on the available peripherals and interfaces.

Breakdown: How you managed your tasks?

First, I design the overall architecture of the calculator, including the input and output interfaces and the processing unit.

Next, I wrote the VHDL code for each component of the calculator, including the state machine that controls the flow of data and the operations that are performed on the input data.

I Synthesize the VHDL code using a synthesis tool, such as Xilinx Vivado, to generate a gate-level netlist that can be loaded onto the Nexys A7 board.

I further Implement the design on the Nexys A7 board using the synthesis tool, and verify that the calculator is functioning correctly by testing it with a set of test inputs and comparing the output to the expected results.

Finally, I test the calculator with a set of test inputs and compare the output with the expected results.

What are the different tasks/roles of the team members in the project?
Describe which team member did which tasks.

Evrard Leuteu:
hexadecimal calculator
solutions for Lab 2

4 Technologies

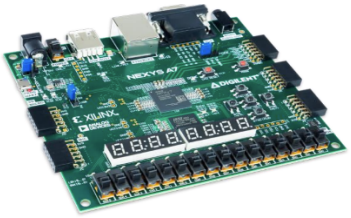
Describe the technological approaches you will use to implement your project.

- *Verilog/VHDL*
- *FPGA*
- *Any other HW/SW*

Team Name

For Hexadecimal calculator project, I used:
VHDL CODE

Nexys A7



The Nexys A7 FPGA.

- **Artux-7 FPGA**
  - 15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops (*8,150 slices)
  - 1,188 Kbits of fast block RAM (*600 Kbits)
  - Six clock management tiles, each with phase-locked loop (PLL)
  - 240 DSP slices (*120 DSPs)
  - Internal clock speeds exceeding 450 MHz
  - Dual-channel, 1 MSPS internal analog-digital converter (XADC)
- **Memory**
  - 128MiB DDR2
  - Serial Flash
  - microSD card slot
- **Power**
  - Powered from USB or any 4.5V-5.5V external power source
- **USB and Ethernet**
  - 10/100 Ethernet PHY
  - USB-JTAG programming circuitry
  - USB-UART bridge
  - USB HID Host for mice, keyboards and memory sticks
- **Simple User Input/Output**
  - 16 Switches
  - 16 LEDs
  - Two RGB LEDs
  - Two 4-digit 7-segment displays
- **Audio and Video**
  - 12-bit VGA output
  - PWM audio output
  - PDM microphone
- **Additional Sensors**
  - 3-axis accelerometer
  - Temperature sensor
- **Expansion Connectors**
  - Pmod connector for XADC signals
  - Four Pmod connectors providing 32 total FPGA I/O

Pmod kypd

5 Hardware Implementation

Describe the implementation of your digital design in VHDL/FPGA, IP-Blocks, etc.
Provide a detailed block diagram for this purpose and briefly explain the used modules.
Provide the results for your FPGA Implementation (Results summary + Hardware results if necessary)

Pins:

bt_clr	IN					N17	☑	14	LVCMOS33*	3.300	
bt_eq	IN					P17	☑	14	LVCMOS33*	3.300	
bt_plus	IN					M18	☑	14	LVCMOS33*	3.300	
bt_sub	IN					P18	☑	14	LVCMOS33*	3.300	

Bt_clr== “Clear”, Bt_eq==”=”, Bt_plus==”+”, Bt_sub==”-”

Team Name

SEG7_anode[7]	OUT					U13	☑	14	LVCMOS33* · 3.300
SEG7_anode[6]	OUT					K2	☑	35	LVCMOS33* · 3.300
SEG7_anode[5]	OUT					T14	☑	14	LVCMOS33* · 3.300
SEG7_anode[4]	OUT					P14	☑	14	LVCMOS33* · 3.300
SEG7_anode[3]	OUT					J14	☑	15	LVCMOS33* · 3.300
SEG7_anode[2]	OUT					T9	☑	14	LVCMOS33* · 3.300
SEG7_anode[1]	OUT					J18	☑	15	LVCMOS33* · 3.300
SEG7_anode[0]	OUT					J17	☑	15	LVCMOS33* · 3.300
SEG7_seg[6]	OUT					T10	☑	14	LVCMOS33* · 3.300
SEG7_seg[5]	OUT					R10	☑	14	LVCMOS33* · 3.300
SEG7_seg[4]	OUT					K16	☑	15	LVCMOS33* · 3.300
SEG7_seg[3]	OUT					K13	☑	15	LVCMOS33* · 3.300
SEG7_seg[2]	OUT					P15	☑	14	LVCMOS33* · 3.300
SEG7_seg[1]	OUT					T11	☑	14	LVCMOS33* · 3.300
SEG7_seg[0]	OUT					L18	☑	14	LVCMOS33* · 3.300

SEG7 == “Seven segment display”

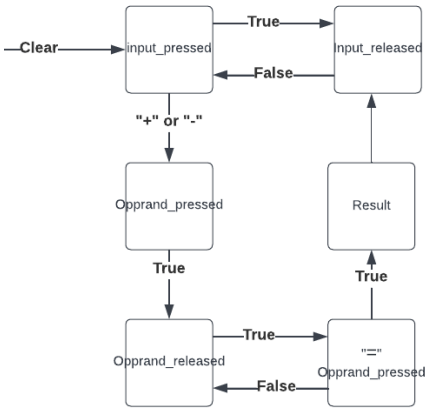
KB_col[4]	OUT					C17	☑	15	LVCMOS33*
KB_col[3]	OUT					D18	☑	15	LVCMOS33*
KB_col[2]	OUT					E18	☑	15	LVCMOS33*
KB_col[1]	OUT					G17	☑	15	LVCMOS33*
KB_row[4]	IN					D17	☑	15	LVCMOS33*
KB_row[3]	IN					E17	☑	15	LVCMOS33*
KB_row[2]	IN					F18	☑	15	LVCMOS33*
KB_row[1]	IN					G18	☑	15	LVCMOS33*

KB==”Keyboard”

6 Software Implementation

Describe the implementation of your software application Block Diagram/Flowchart, Vitis etc.

Hexadecimal calculator operation.



- Pushing keypad or operation buttons will cause the device to respond by updating variables, altering the output, or choosing the next state, depending on the current state.
- Pushing the clear button causes the device to go into the Input_pressed state.
  - The device is in this position and waiting for a keypad button to be depressed.
  - The code waits in the input_release state for the keypad button to be released after adding the new digit to the 16-bit word in the accumulator when a keypad button is pushed.
  - To wait for the following digit, it then transitions back to the Input_pressed state.
- until the "+" button is pressed, the procedure is continued.
  - When the first digit of the second operand is received, the machine enters the opprand_pressed state.
  - Pushing a keypad button causes it to record the hex digit and go into the opprand_released state where it waits for the keypad button to release.
  - When a keypad button is depressed and then released, the device enters the Opprand_pressed state, where it continues to receive operand digits.
- When the user clicks the "=" button, the procedure stops, adds the values, and switches to the Result state.

The following source files where used in the software implementation.

Hexcalculator.vhd

The keypad interface and the 7-segment decoder interface modules are created by the top-level source module hexcalculator.
Connects the external keypad, buttons, and display.
has a timing process that produces clock signals for the finite-state machine, display multiplexer, and keypad.
use a finite-state machine to carry out the calculator's operations in response to button presses.

Hexcal.xdc

Constraint file for the hardware.

Keypad.vhd

This process synchronously tests the state of the keypad buttons. On each edge of samp_ck

Leddec16.vhd

Turn on segments and select digit data to be displayed, corresponding to 4-bit data word

7 Sources/References

Provide the sources on the technologies/code you used in your project (Github).