Final Project, ELEC 4320, 2025

Rules:

(1) **Teams**: the limit is **a maximum of three** students in a group.

(2) **Timeline** (23:59 by default for the deadlines below):

a) Decide your project topic and form your team. Empty groups will be created on *Canvas—People* in advance. By **Oct. 31**, all members of a team should join a Group on Canvas and submit your project proposal in a group assignment (will be published then). For groups selecting provided projects, the proposal only needs to include the project name. For groups to propose their own project, details should be included for difficulty evaluation.

b) Presentation (demo): **Dec. 16** and **Dec. 17**.

c) Submission of written report: **Dec. 15**, through Canvas.

(3) **Submission** of a **zip** file to Canvas assignment containing:

a) A debugged sequential logic design with complete functionalities on the BASYS3 board

i. Verilog code (Must copy to a .txt file, otherwise, 10% deduction and re-submit.)

ii. Complete Xilinx project folder

iii. .bit file

iv. Other software code running on the host PC (if any)

b) A written report (.pdf). A suggested organization is:

i. Introduction/Background

ii. Development environment, Input and Output

iii. Elaboration of the functionalities of your core modules

iv. Schematic diagram of your top design and core modules (the meaning of I/Os)

Note: please clearly define your effort and the parts borrowed from existing works

v. A flowchart or a visualized finite state machine for your top modules and/or the core modules

vi. Description of the debugging for top or core modules (I/O, test data, etc.)

vii. Simulation waveform for core modules

viii. Task distribution ({task: name}) and contribution ratio ([*integer*%] within your group)

ix. Reference list with proper citation format: papers, blogs, technical documents or existing projects

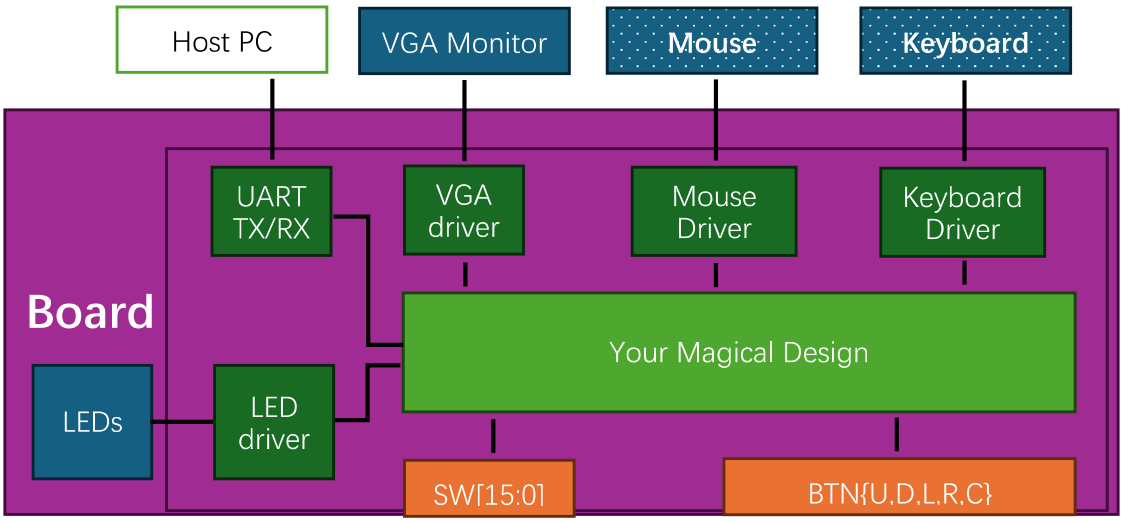
(4) **Resources**:

We provide BASYS3 FPGA and PYNQ-Z2 FPGA. You can select the one based on your requirements.

a) Complete constraints for BASYS3:
   i. Basys 3 - Digilent Reference
   ii. https://github.com/Digilent/digilent-xdc/blob/master/Basys-3-Master.xdc
b) Or select the PYNQ-Z2
   i. PYNQ | Python Productivity for AMD Adaptive Computing platforms

(5) Functional requirements for your design:

a) Functionalities:
   i. A meaningful sequential logic (finite state machine)
   ii. Access to on-chip memory (RAM, ROM)
   iii. Register read and write
   iv. Proper human-machine interaction (I/O) such as:
      1. Input: switches, buttons, UART, mouse/keyboard (usb, advanced choice), etc.
      2. Output: 7-segment LEDs, VGA, UART, etc.
b) An overview of a potential system design



(6) Scoring:

a) Basic criteria:
   i. Project completeness level by the deadline.
   ii. Design complexity and technical implementation difficulty (considering team size).
   iii. Interaction: such as how many I/O utilized (considering team size).
   iv. Optimization regarding performance, resource, power, etc. (if applicable)
   v. The quality of your written report.

vi. The quality of your presentation. All members must participate in the demo part.

b) Late policy:

i. 20% deduction for delay < 12 hour, 50% deduction for delay < 24 hour. No late hand-in will be accepted for late submission of more than one day

c) Unless a specific arrangement is requested, all members of a group will be awarded an equal score, ensuring a fair distribution of points.

(7) Project selection:

We provide three example projects for you as follows. All these projects **must** be implemented using Verilog on a Basys3 FPGA. You can also propose your own projects. For inspiration, you can refer to the projects on these website: https://openhw.org/gallery.html

Before you start working, you need to submit a proposal about your project. This will allow us to evaluate the complexity and ensure a fair workload for all students. For example, a complete 2D Snake Game is only eligible for 70% of the total score due to its relative simplicity. Additionally, for those groups selecting PYNQ-Z2, your project should be more complex, as PYNQ-Z2 allows users to implement using Python, that is faster and easier than Verilog.

Example projects

1. Electronic Calculator

Target

Design a simple electronic calculator circuit that includes data input processing module, arithmetic operation module (including addition, subtraction, multiplication, division, square root, cosine, sine, tangent, logarithm, power, and exponential operations), and result processing module. The design of all arithmetic circuits must not utilize IP cores or directly employ look-up table methods. The assessment will be based on the design block diagram and description, circuit design description for each module, timing descriptions, simulation results, resource report, design summary, and source code of the main program.

Requirements:

a) Design the data input processing circuit:
The data input processing circuit is used to handle input from keys and switches, with an input data range of -999 to 999. There are five keys on Basys3 FPGA boards, where the keys on the left and right are used to select the input position (i.e., units, tens, hundreds, thousands), and the two keys on the top and bottom are used to adjust the current position's data value, allowing the data to vary between 0 and 9, or sign "+" and "-". The middle key is used to confirm the current input and proceed to the next data entry step or obtain the calculation result. The switches are used to select the type of calculation. It is important to note that these keys do not stabilize immediately when closed, nor do they disconnect immediately when opened. Consequently, the closing and opening moments will be accompanied by a series of jitters, resulting in unstable button inputs. Therefore, a debouncing circuit needs to be designed. All these input data and operations should be displayed by digital cube.

b) Arithmetic Operation Circuit
The arithmetic operation circuit is used to perform various calculations, including Addition (a+b), Subtraction (a-b), Multiplication (a*b), Division (a/b), Square Root ($\sqrt{a}$), Cosine (cos(a)), Sine (sin(a)), Tangent (tan(a)), Arcsine (arccos(a)), Arccosine (arcsin(a)), Arctangent (arctan(a)), Logarithm ($\log_a b$), Power (a^b), Exponential Operations (e^b), Factorial (a!). It is important to note that IP cores and direct look-up tables must not be used in the operation circuit. Additionally, it is necessary to ensure the stability of the

calculation results. That is, when the computation is complete, a valid signal must be sent to notify the result processing circuit, and all output data during the computation process should be considered invalid. For Cosine, Sine, and Tangent, the input should be angle. For Arcsine, Arccosine, Arctangent, the output should be angle. All these operations should work at 300MHz.

c) Result Processing Circuit

The result processing circuit displays the output from the aforementioned arithmetic operation circuit. Since the Basys3 development board provides only a 4-digit seven-segment display, an LED is used to assist in the display. This means that the output results will be displayed in multiple stages, with the LED indicating the current display position. For example, when the LED shows 001, the seven-segment display will show "-012"; when the LED shows 000, the seven-segment display shows "3456,". This indicates a final output of -0123456. For operations resulting in decimal values, such as cosine, the decimal will retain significant digits equal to the average of the last digits of the student ID numbers of the group members.

The data input and display methods are not limited to a single approach. For example, communication can be established with a PC host using USB interface protocols to input and read data from the PC. If you use interfaces such as USB/UART/VGA for input and display, you can get an additional 5 points (no matter how many interfaces). The group can correspondingly reduce the implementation of the arithmetic operation module. For instance, if the group inputs data via a USB interface through the PC and displays the data on a screen using VGA, they can earn 5 points and may reduce the implementation of Square Root.

d) Expansion

Supports combined calculations of these operators. Your design must be capable of executing a series of sequential arithmetic and trigonometric operations on a given input string. To simplify the task, binary operators must be performed from left to right. For example, you can print 1, *, 2, +, 3, *, 4, then your calculator outputs the result of ((1*2)+3)*4. Another example is 1, +, 2, *, 3, -, sqrt, 4, the your calculator outputs the result of ((1+2)*3)-sqrt(4).

Design Report Contents

a) Overview of Design Ideas

b) Overall Design Block Diagram and Detailed Explanation

c) Timing Explanation: Provide Theoretical Analysis and Calculations

d) Module Design Block Diagrams, Pin Descriptions, and Relevant Timing

e) Program Code and Necessary Comments

f) Results: Provide Examples of Results for Each Type of Operation

g) Other Necessary Explanatory Content

Grading Criteria

<table>
<tr><th colspan="2">Items</th><th>Points</th></tr>
<tr><td colspan="2">Report</td><td>20</td></tr>
<tr><td rowspan="4">Design</td><td>Design the data input processing circuit</td><td>10</td></tr>
<tr><td>Operation Circuit Design:

Addition (2), Subtraction (2), Multiplication (2), Division (3), Square Root (5), Cosine (4), Tangent (4), Arcsine (6), Arccosine (4), Arctangent (6), Logarithm (5), Power (5), Exponential Operations (4), Factorial (3)

An error of less than 5% receives full points, with a deduction of 1 point for every additional 5% of error.

Show the maximum frequency that the project can successfully work without timing violation. A penalty of 1 point for every 5 MHz below 300MHz.</td><td>55</td></tr>
<tr><td>Result Processing Circuit</td><td>5</td></tr>
<tr><td>Extension</td><td>20</td></tr>
<tr><td colspan="2">Total</td><td>100</td></tr>
</table>