1. Project Overview

For this project, groups of two students will design and implement a 12-bit fast, arithmetic calculator using an Artix-7 FPGA. The calculator will support a 12-bit data bus, basic arithmetic operations, at least one finite state machine (FSM), and a special user function. An in-lab inspection of basic functionality is scheduled for Week-10. Your VHDL code will need to be finalised and submitted to a git repository by the day of your inspection. A group report is due in Week-12. This project brief provides details on resources, the project specifications to be met, and information on how your work will be submitted and assessed.

As stated in the ENEL373 *Course Summary*, 25% of your final course mark is based on your project assessment, which you and your group will work on over most of Semester 1. Following a tutorial in Week-1, two formal laboratories (PL-1 & PL-2) comprise 4 formal laboratory hours. These laboratories are designed to introduce students to basic language and FPGA resources, however tight time constraints mean that student groups, once formed, should **start on their project in Week-4**.

If you have any questions or comments regarding this project, a project forum has been setup on Learn.

Important Dates

Complete your group booking:	12:00 Monday 8 Mar. 2021
Formal Project Start Date[1]:	During Week-4
Project Milestone (6%):	**Week-5**, in your Tue. or Wed. Lab session
Project Inspection + Code Submission (13%):	**Week-10,** in your Tue. or Wed. Lab session
Group Report (6%):	17:00 Thu. 3 June (Week-12)

3. Facilities & Resource Requirements

3.1. Development boards & Xilinx Software

The ECE department has a class-set of Digilent Nexys-4 DDR FPGA development boards and these will be available to use in the period allocated to you during your formal weekly laboratory session during Weeks 1 to 12. Access to the Xilinx suite of software tools (Vivado) used in the Electronics lab, is also available in our CAE laboratory. Also, a Vivado WebPACK is available from Xilinx free of charge (after registration) and this supports the Artix-7 FPGA that is used on the Nexys-4 DDR laboratory boards (this is detailed in a Week-3 handout). To speed-up completion of this project, students should make use of the simulation tool in Vivado. The Nexyx-4 board only need be used sparingly for project development, testing and demonstration, however an FPGA remote laboratory is available to groups for lab or project development. A brief primer on how to login and use our remote labs, either on or off campus, is also covered in a Week-3 handout that is available on Learn. Given our 24/7 remote lab to access and use an FPGA board, Nexys-4 FPGA boards will not be available for loan to use outside of the Electronics laboratory.

3.2. Laboratory sessions

By 12:00 8 March you need to be booked into a 2-person group using our Learn group booking system. All members of each group should be booked into your respective Tuesday or Wednesday lab session. **To be fair to other students, i.e., allowing other groups access our TAs and their lab, students will be restricted to work in one, and only one, 2-hr lab session**. Note that there is a spare lab session (to all) available on Thursdays, but access to TAs will be limited.

3.3. Repositories

We highly recommend you frequently archive all work to your P drive using the Vivado Archive utility. In addition, your git repository will be available towards the end of Week-4. Lastly, to speed up processing times on lab PCs we recommend you insert a USB-3 (“pen” or SS) drive as your working directory in a rear USB slot (blue only) of your lab PC, i.e., don’t network to your P: drive for synthesis and implementation processes while working in the lab; do, however, **periodically save your work to your P drive, and latter, to your eng-git repository**.

[1] Formed groups can start on this project anytime.

4. Project Requirements

Your Week-10 FPGA project requirements will allow a user to enter 12-bit data operands into one or more registers (Item-A), implement the basic functionality of a simple 12-bit signed arithmetic module using two operands and an operation code (see Item-B). The results of your signed arithmetic operation will be displayed on the 7-segment display (Item-C).

A. Two 12-bit operands and an *n*-bit operator will be input into FPGA-configured registers using slider switches. Since there is only 16 switches, data will need to be input in binary and in stages. A single push-button on your Nexys-DDR (use either the middle button or an operator button to finish entering your operand). Either *in-fix* or *post-fix* notation can be used to enter operands. To facilitate user interface operations, a sequencer (see p.404-406 of [1]) or a simple finite state machine (FSM), as discussed in lectures and given examples, can be used.

B. Implement a simple 12-**bit** arithmetic module in your provided FPGA board. Required arithmetic operations include:
   I. Full addition and subtraction operations using signed, integer arithmetic,
   II. Multiplication and division, although limited support for these operations will be acceptable.
   III. An additional special arithmetic operation, other than the basic 4, is required.

   At least one source operand (typically, two will be used), an operator code, and a result should be stored in registers.

C. User feedback is required to confirm that operands and an operation code have been correctly entered, and that a result (destination operand) has been correctly processed by your arithmetic module. Given that operands and an opcode will be stored in registers, the **contents** of these registers, rather than simply the switch values, should be shown in binary on the provided array of LEDs, as each operand and operator is entered. Note that some groups may want to show operands as they are entered on the provided 7-segment displays.

5. Project Demonstration Requirements

In Week-10 your group will demonstrate the project requirements, as outlined in Section 4. Groups will be expected to have developed a simple datapath to support their arithmetic module, with registers, controlled by a finite state machine (FSM), or sequencer, to synchronise arithmetic operations. Your group demonstration will take place in your lab session in Week-10. Project sign-up sheets for your respective lab session, i.e, either Tuesday or Wednesday, will be available in the Electronics lab starting in Week-9. One, and only one person from each group should sign-up for a short demonstration slot. Due to tight timelines, it will not be possible to change lab sessions, and this will be strictly enforced.

6. Reporting Requirements

At the conclusion to your FPGA project you will be required to submit a formal **group** report, i.e., **one report per group of two students near the end of Week-12**. This section will outline specific reporting requirements related to this project. Details on the general formatting and content requirements for your report are provided below.

As a general guide, your report should have no more that 7 pages of content, i.e., the body of the report from your introduction to conclusion, which will have numbered sections with tiered subsections (as in this brief), and should address the following:

A. A short introduction, outlining in your own words the project requirement and what was achieved.

B. A top-level block diagram summarising your system design. Either a user-generated diagram or an RTL schematic view would be acceptable.

C. An expanded design summary, to complement the outline in Item-A, should be provided. This should address methods your group used to implement your FPGA (about 1 to 2 pages), and **include design decisions** with **justifications**. Some examples include the design of your FSM, e.g., types of encoding methods used and modular design considerations, e.g., structural vs.

behavioural or dataflow abstractions. Specific references should be made to your VHDL source code in an Appendix. VHDL descriptions could, for example, include an RTL description.

D. A short section on how you tested a significant module, such as the ALU, is appropriate. **At least one Testbench and associated waveforms**, demonstrating the functionality of a module, ***should be included in an Appendix***.

E. Include a short section outlining any design and/or implementation problems you encountered during your development, and how these were solved by your group.

F. Provide a short section (a few sentences) on what you learned and any suggestions on how the project could be improved.

G. A brief conclusion should summarise your work.

H. Appendices & References: Since your code will be submitted via eng-git, inclusion of code will not be necessary in your report. However, reference to a code snippet provided in an appendix or a few lines of code given in the main body of your report, is quite appropriate. **Formal citations** to books, web articles, open source VHDL code, etc., should be formally listed in a reference section using the IEEE reference style format. As mentioned in Item-D, a section in the Appendix is required for your Test-bench waveforms but more than one appendix, but < 4, is allowed.

I. Your report can only be submitted in PDF. Word documents will not be accepted via Learn.

An eng-git project repository will be available for your code development. All code for your FPGA project must be **committed and pushed** to your repository by **the time you demonstrate your code**. **Note that code submitted after this date *will not be marked*.** Marks will be awarded on your design methodology and justifications to achieve the project specifications, in addition as to how your design was implemented. Note also that the quality of your VHDL code will be assessed, and this includes how well it is documented[2].

7. Assessment

The assessment for this project is worth 25% of your overall course mark. The contributions for each part of this assessment are as follows:

a.	**Project Milestone**: Enter a 12-bit number via switches and show this in binary coded decimal form on three 7-segment, multiplexed displays.	6%
b.	**Project demonstration** (several minutes in your lab session):	13%
	b.1. Code Submission via eng-git repository:	
	b.2. Signed arithmetic module (add, subtract, multiply, divide) operations.	
	b.3. A special function, e.g., an exponent, sine, etc.	
	b.4. One structural and several behavioural/dataflow modules.	
	b.5. A finite state machine (FSM) or sequencer.	
	b.6. A VHDL test-bench module.	
c.	**Group project report** submission:	6%
	c.1. Results from your VHDL Test-Bench module	
	c.2. As per Section 6: subsections A to I.	
	Total:	**25%**

8. Submissions

An eng-git code repository will be set-up for each group by the end of Week-4. Each repository should **ONLY include the following source files**:

- A top-level structural VHDL module.
- All behavioural and/or datapath VHDL modules, including a FSM.
- A test-bench VHDL module.
- A constraints file.

[2] Note that a VHDL style guide is available in Section 1 on your ENEL373 Learn pages.

Do NOT push any other files other than source files to your eng-git repository. In short, I do not want to see any large project files in your repository - only source files, please!!

A separate drop-box will be provided on Learn for groups to **submit their group project report in PDF, and only in PDF**. Lab books will not be assessed but all students are expected to maintain good engineering practice and take and use their lab books in the lab, and keep their books up to date.

References

[1] M.M. Mano and C.R. Kime, "Logic and computer design fundamentals", 2nd Ed., 2001

Book: In Short-Term Library section, TK 7888.4 .M285 2001, EPS Library
Book: In Short-Term Library section, TK 7888.4 .M285 2000, EPS Library (+1 More)
Book: In Short-Term Library section, TK 7888.4 .M285 2004, EPS Library.

[2] S. Brown and Z. Vranesic, "Fundamentals of Digital Logic with VHDL Design" (Latest Edition) by S. Brown and Z.

Book: In Short-Term Library section of the EPS Library

[3] R.E. Haskell, D.M. Hanna, "Introduction to Digital Design Using Digilent FPGA Boards: Block Diagram / VHDL Examples", Paperback, September, 2019

Book: In Short-Term Library section of the EPS Library

On-line version (reduced content to our EPS printed version):

https://reference.digilentinc.com/_media/textbooks:intro_digital_design-digilent-vhdl_online.pdf