ENGN4213/6213
Digital Systems and Microprocessors

Semester 1, 2023

Week 1, Lecture A:

Part 1: Introduction to the course

Australian
National
University

Acknowledgement of Country

"We acknowledge and celebrate the First Australians on whose traditional lands we meet, and pay our respect to the elders past and present."

Overview

- Motivation for Studying this Course
- Course Description
- Learning Outcomes
- Teaching Team
- Course Information
- Course Overview
- Suggestions for Study
- ENGN4213/6213 Class Representatives
- ANU Access & Inclusion (A&I)

Motivation for Studying this Course

- Different career paths
  - Embedded/ Hardware electronic engineer
  - Consulting companies
  - Systems Engineer/ Leading Engineering Teams
  - Tec Start-ups
  - Leadership positions in Technology sector
- This course can be used as a foundation to become an embedded engineer or just to know/appreciate what is involved in digital hardware so that you can lead others/ see big picture
- No better way than having hand-on experience

Motivation for Studying this Course

Digital electronics are everywhere ..

- Computers, smartphones, smart speakers
- Smart home products
- TVs, microwaves, vacuum cleaners, fridges
- Cars, boats, trains, aeroplanes, rockets
- Robots, satellites, medical devices
- Intelligent Physical Systems/ Cyber Physical Systems
- Internet of Things (IoT)
- Smart grid, smart electricity meters

Pretty much most electrical items and large systems all have digital electronics inside.

Of course, you still need analog electronics!

Motivation for Studying this Course

Advantages of Digital vs. Analog

Just a few!

- An analogue circuit is hard to change once it is manufactured. Digital circuits can be reprogrammed to vary their behaviour (e.g. microprocessors) without changing the circuit hardware.
- Analogue circuits are less noise-tolerant and their performance can vary gradually with temperature and age. Digital circuits generally either function as intended or fail completely.
- Complex analogue circuits can take up a large amount of space on a PCB. Digital circuits can normally perform the same functions as their analogue equivalent in just a fraction of the size.
- After digitization, it is very easy to subject information to all sorts of mathematical or algorithmic operations (versatility).

But analog is a necessary part ! Digital signals are analog first!

Read Textbook (Wakerly, 5th edition): Section 1.2 (Analog versus Digital)

Course Description

- ENGN4213/6213 expands your knowledge of digital systems building on the background you acquired in ENGN2218 and ENGN2219. We will push well beyond logic gates and truth tables and learn about systems for which the time dimension is important.
  - Systems that can carry out complex operations.
  - Systems that can be useful in the real world, and that you can design and build by yourselves!
- ENGN4213/6213 will give you knowledge and tools to develop digital solutions using two popular platforms:
  - **Field-Programmable Gate Arrays (FPGA)**
    - Weeks 1 to 6 (Project Assessment in Week-7)
  - **Microcontrollers**
    - Weeks 7 to 12

Learning Outcomes

On successful completion of this course, students should have the skills and knowledge to:

- Explain the fundamental principles of sequential digital circuits and finite state machines.
- Compare and describe the architecture and fundamental concepts of the modern embedded microprocessor systems.
- Design complex digital systems using schematics and Verilog HDL, and implement these on commercial-grade field-programmable gate array (FPGA) development boards.
- Design an embedded system using C/C++ programming and microprocessor boards.
- Analyse critically, and evaluate the performance of systems against the design requirements.
- Plan, execute and report on small projects working in a group, communicating effectively in written form about their work

Teaching Team

- **Conveners**
  - *A/Prof. Nan Yang* (nan.yang@anu.edu.au)

    Room B154, Brian Anderson Building (Bldg. 115)
  - *A/Prof. Xiangyun (Sean) Zhou* (xiangyun.zhou@anu.edu.au)

    Room B158, Brian Anderson Building (Bldg. 115)

- **Lecturers**
  - *Ms. Amy Bastine* (amy.bastine@anu.edu.au)

    B147, Brian Anderson Building (Bldg. 115)
  - *Mr. Hanwen Bi* (hanwen.bi@anu.edu.au)

    B140-D, Brian Anderson Building (Bldg. 115)

Note: For email communication, please include “ENGN4213” or “ENGN6213” in the subject of the email.

Teaching Team

- **Tutors**
  - Mr. Jiarui (Frank) Wang (jiarui.wang@anu.edu.au)
  - Mr. Shaoheng (Henry) Xu (shaoheng.xu@anu.edu.au)
  - Mr. Manish Kumar (manish.kumar@anu.edu.au)
  - Mr. Huawei Zhang (huawei.zhang@anu.edu.au)
  - Mr. Zhifeng Tang (zhifeng.tang@anu.edu.au)
  - Mr. Wei-Ting Lai (wei-ting.lai@anu.edu.au)
  - Mr. Xingyu Chen (xingyu.chen1@anu.edu.au)
  - Ms. Zeinab Salehi (zeinab.salehi@anu.edu.au)
  - Ms. Angela Zhang (yile.zhang@anu.edu.au)
  - Mr. Skanda Dolphin (skanda.dolphin@anu.edu.au)

FPGA Labs					MCU Labs		
Lab 1 (Week 2)	**Lab 2** (Week 3)	**Lab 3** (Week 4)	**Lab 4** (Week 5)	**Lab 5** (Week 6)	**Lab 6** (Week 7)	**Lab 7** (Week 8)	**Lab 8** (Week 9)
Manish Kumar, Zhifeng Tang Jiarui (Frank) Wang	Angela Zhang Zhifeng Tang Jiarui (Frank) Wang	Angela Zhang Jiarui (Frank) Wang Shaoheng (Henry) Xu	Shaoheng (Henry) Xu Zhifeng Tang Manish Kumar	Jiarui (Frank) Wang Shaoheng (Henry) Xu	Zeinab Salehi Xingyu Chen Huawei Zhang	Zeinab Salehi Wei-Ting Lai Manish Kumar	Wei-Ting Lai Xingyu Chen Huawei Zhang

FPGA Tutorials				MCU Tutorials			
Tutorial 1 (Week 2)	**Tutorial 2** (Week 3)	**Tutorial 3** (Week 4)	**Tutorial 4** (Week 5)	**Tutorial 5** (Week 8)	**Tutorial 6** (Week 9)	**Tutorial 7** (Week 10)	**Tutorial 8** (Week 11)
Skanda Dolphin	Angela Zhang			Skanda Dolphin			

Course Information

Course Information

– **Course websites**

- ENGN4213: https://programsandcourses.anu.edu.au/2023/course/ENGN4213
- ENGN6213: https://programsandcourses.anu.edu.au/course/engn6213
- Wattle: http://wattle.anu.edu.au
  - ○ Make sure you regularly check Wattle (at least twice a week for course information and announcements)
  - ○ Follow appropriate netiquette when using communication forums and interactive tools.

– **Course Outline**

- All course logistics presented in this lecture are also included in the Course Outline document available on the Wattle course page.

Course Information

Prerequisites

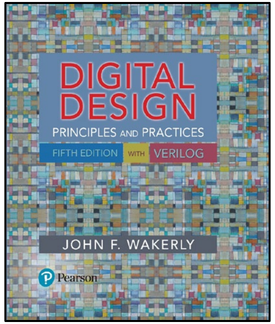
- Background knowledge about interpreting circuit drawings, a basic understanding of transistors and logic gates, in particular.
- Background knowledge in computer architecture and programming languages, especially C programming.

Please undertake independent review of key contents in courses like ENGN2218 (*Electronic Systems and Design*) and ENGN2219/COMP2300 (*Computer Organisation and Program Execution*), and C primer document**.**

Course Information

Prescribed Textbooks

- John F. Wakerly, "***Digital Design, Principles and Practices***", 5th edition, Pearson/Prentice Hall.
    - Main reference for the *first part* of the course
    - E-version can be purchased https://www.pearson.com.au/9780134460239



- Donald Norris, "***Programming with STM32: Getting Started with the Nucleo Board and C/C++***", McGraw-Hill Education, 2018.
    - Main reference for the *second part* of the course
    - Available online with ANU subscription (use your ANU email): https://www.oreilly.com/library/view/programming-with-stm32/9781260031324/?ar

Course Information

Timetable

	Session Number	Day	Duration	Location
Lectures: Weeks 1 – 12	Lecture session A	Tuesday	10am – 12am	SRES Lecture Theatre (Bldg. #: 48A)
	Lecture session B	Wednesday	12pm – 1pm	Gould Seminar Room 235 (Bldg. #: 116)
Tutorials: Weeks 2 – 5 & 8 – 11	Tutorial session 1	Wednesday	9am – 11am	Birch 1.43 Interactive Theatre (Bldg. #: 35)
	Tutorial session 2	Thursday	9am – 11am	Hancock 2.24 (Bldg. #: 43)
	Tutorial session 3	Friday	9am – 11am	PSYC G5 (Bldg. #: 38B)
Practical Labs: Weeks 2 – 11	Lab session 1	Tuesday	2pm – 5:30pm	Ian Ross R103 (Bldg. #: 31) *Lab session 5 will open only when the number of students exceeds the total capacity of four lab sessions.
	Lab session 2	Wednesday	2pm – 5:30pm	
	Lab session 3	Thursday	2pm – 5:30pm	
	Lab session 4	Friday	2pm – 5:30pm	
	Lab session 5*	Friday	9am – 12:30pm	

Note: Recordings of lectures will be uploaded to Echo360 in Wattle after each lecture.

Course Information

Classroom Model

- **Pre-recorded Video Lectures** will be uploaded in Wattle every week before the in-class lectures.
    - FPGA-related contents: *Dr. Nicolo Malagutti* (Recorded in 2020)
    - Microcontroller-related lectures: *Prof. Thushara Abhayapala* (Recorded in 2021)
    - These videos have detailed explanation of the key concepts and students are highly expected to watch them before the in-class lectures

- **In-class lectures** will discuss all the assessable contents of the course
    - Focus on applications, insights and consolidation of concepts delivered by video lectures.

- **Guest lectures** from industry experts will provide real-world insights and practical knowledge to help students prepare for future careers.

- **Tutorials** will provide exposure to exam-style questions and extension concepts.

- **Labs & Projects** are the most important learning activities of this course.

Course Information

Assessment

No.	Component	Weight	Due Date	Date of Return of Assessment	Linked Learning Outcomes
1.	**Practical Labs** (8 labs)	**26%**	Throughout semester	Marked during labs	3, 4, 5
2.	**FPGA Project**	**25%**	Week 7	Week 9/10	3, 5, 6
3.	C Primer Quiz	4%	Week 10	Marked after quiz being attempted	4
4.	**Micro-controller Project**	**25%**	Week 12	Week 14	2, 4, 5
5.	Final Exam	20%	Exam period		1, 2, 5

Course Information

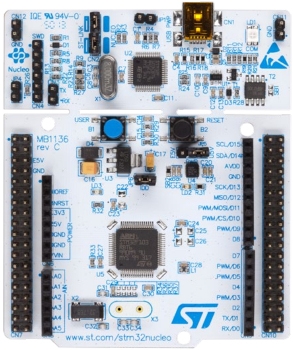
Assessment: Practical Labs

	Lab	Week	Weight	Topic
FPGA	Lab 1	Week 2	2 %	Introduction to Vivado
	Lab 2	Week 3	3 %	Implementing designs onto FPGA and Essential sequential designs
	Lab 3	Week 4	3 %	Seven Segment Displays & Switch Debouncers
	Lab 4	Week 5	4 %	Finite State Machines
	Lab 5	Week 6	3 %	TBA: Designed by CEA
Microcontrolle r	Lab 6	Week 7	3 %	Programming the STM32 Microcontroller : GPIO, Interrupts
	Lab 7	Week 8	4 %	Timers and Serial Communication of STM32 Microcontroller: UART
	Lab 8	Week 9	4 %	Serial Communication of STM32 Microcontroller: I2C LCD

Course Information

Assessment: Lab Activities

- The course aims to give you hands-on experience and hence **heavily geared towards lab activities** and project work with **real hardware and software systems**.
- There will be **5 FPGA** and **3 Micro-controller Labs** worth **26 %** in total. These will prepare you for the Project Assignments. So, it will be extremely hard to do the project without completing the labs.
- Lab manuals will be provided (via Wattle) at least one week before your lab session. You must get familiarized with the activities before attending the session.
- Tutors will be present in the Ian Ross 103 laboratory during the lab sessions to assess your lab activities.



Course Information

Assessment: Lab Activities

- Lab marks: Each lab will have a few “Activities” to be completed in order to evaluate your progress. After completing each Activity, show your results to the tutor to get marked. The tutor will then ask you a few questions to assess your understanding of the concepts involved in the lab.

- Labs run for three hours. In addition, a 30-minute extension is given for tutors to finish marking the working activities of remaining students in the lab session.
  - To complete all the required tasks within 3hrs, you are strongly encouraged to attempt them on your own time before attending the lab session.

- Lab activities will be done and assessed individually. However, you can work with your fellow students and seek help from them to learn. Students can teach each other, but blunt copying without any understanding will be penalized.

- If you cannot attend a lab or need to attend a session other than the one you signed-up for, please contact the course co-convenors with a valid reason.

Course Information

Assessment: FPGA Project

- The assignment will consist of a practical project to test your acquired digital design and FPGA implementation skills.
- The project is conducted in **groups of 2 students** and involves hands-on work using the board.
- Deliverables (worth **25** % of your grade):
  - Submit design files and a technical report summarizing your work (due date: *Friday 21 April*).
  - Live demo of your implementation followed by Q&A. (due date: *17 - 19 April*)
- Group marks are moderated based on group members' individual contributions.
- Students enrolled in **ENGN6213** will need to complete an individual reflective question on top of the assignment workload.

Course Information

Assessment: Microcontroller Project

- The assignment will consist of a practical project to develop an embedded solution using ARM Cortex Microcontroller board.
- The project is conducted in **groups of 2 students** and involves hands-on work using the board.
- Deliverables (worth **25** % of your grade):
  - Submit design files and a technical report summarizing your work (due date: *Thursday 1 June*).
  - Live demo of your implementation followed by Q&A. (due date: *24 - 26 May)*
- Group marks are moderated based on group members' individual contributions.

Course Information

Assessment: C Programming

- We use C programming in Labs 6 to 8 and in the Micro-controller project. So, you must be able to do write codes in C in Week 7.

- Self-learning:
  - Work through the exercises in the **C Primer** document in Wattle (under *C Programming (self-learning)* topic) at your own time to refresh your basics.
  - **5 weekly C Primer quizzes** for your self-assessment. Attempt them to monitor your progress and get feedback. You can even attempt them multiple times until you get all answers correct.

- Take the **C Primer Assessment Quiz** worth **4** % of your grade, with similar questions to those asked in the weekly quizzes.
  - Opens in Week 10 and due by *end of Week 11*
  - Only **single-attempt** allowed

Course Information

Assessment: Final Exam

- Worth **20** % of your grade
- Consists of a mix of multiple-choice, short answer and/or extended answer questions
  - Similar in difficulty and style to those you will encounter in the tutorials.
- Will be held during the examination period (*1 June to 17 June*)
- Students enrolled in ENGN6213 will have additional questions

Course Information

Assessment

- Group work is intended to teach you to work together, help each other to learn and understand better. Live demos and questioning will check your understanding.

- We will be **very unforgiving with “free-riders”.**

- You are encouraged to use the Internet to search and learn – there are tons of useful information (including YouTube videos) on both FPGA and Micro-controllers, including the specific boards we use.

- **Plagiarism will not be tolerated**. Depending on the severity of a breach, the penalties under *ANU policy* can be very serious. Make sure you properly acknowledge any work you cite or use, and ***do not claim anything but your work as your own***.

Course Information

Software Requirements for the First part of the course

– We use **Xilinx Vivado** integrated development environment exclusively in this course to design, simulate, debug, synthesize, and deploy our FPGA programs.

- Vivado ML Edition (free version), latest version 2022.2 from the webpage: https://www.xilinx.com/support/download.html
- Labs start in Week 2, and you need to have access to a computer with Vivado installed for performing the lab tasks. The installation can take 1 to 3 hours, so do it as early as possible.
- FPGA Software (Xilinx Vivado) Installation guide is available in Wattle under the *Practical Labs* topic.
- Note that the software needs a significant space in your hard disk. In case of difficulty in installation, the Ian Ross R103 PCs installed with Vivado can be used for labs and the project.

Course Information

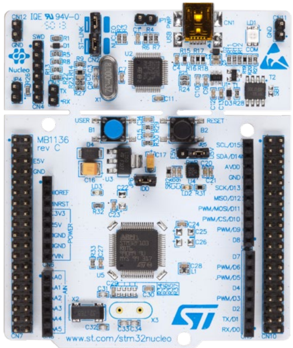
Hardware Requirements for the First part of the course

- We use **Digilent's Basys 3 Artix-7 FPGA Trainer Board** to implement your digital designs https://reference.digilentinc.com/reference/programmable-logic/basys-3/start
    - Sponsored by **CEA Technologies**
    - Each student is provided an FPGA development board, which is not needed to return.
    - Collect them during your Lab-1 session

Course Information

Software & Hardware Requirements for the Second part of the course

- STM32CubeIDE development platform (https://www.st.com/en/development-tools/stm32cubeide.html)
- STM32F411RE Nucleo board
  - Each student can borrow a micro-controller board from the School and needs to return this board after finishing the micro-controller project.
- More details will be provided later.



Course Overview

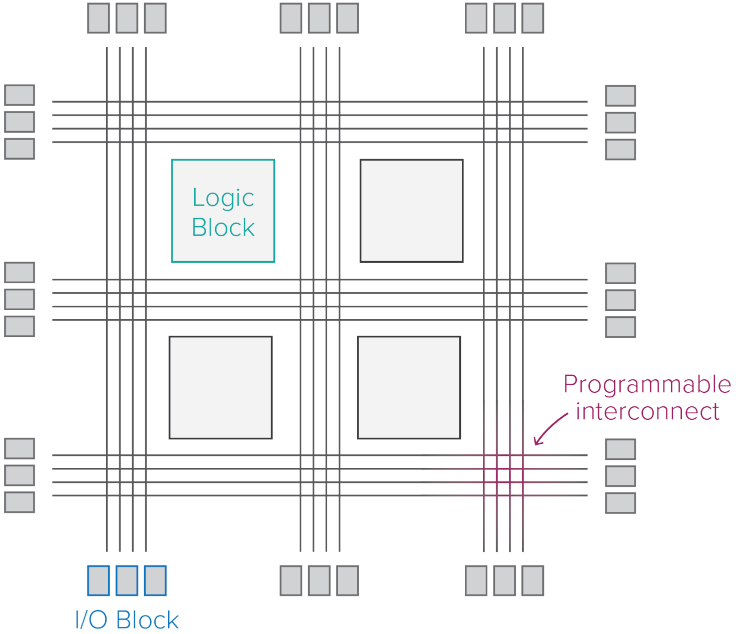
Course Overview

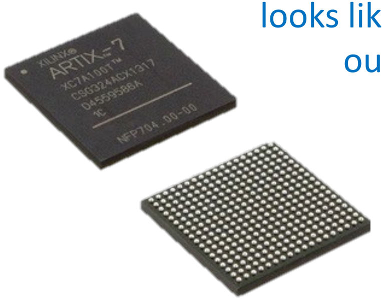
- Review of combinational logic analysis and design
- Analysis and design of synchronous finite state machines and register transfer level systems
- Computer-aided design of digital electronic systems using real-world software packages
- A detailed introduction to the Verilog hardware description language
- Implementation of digital systems in FPGA programmable logic devices
- Embedded system design on ARM microcontrollers using C programming

Course Overview

Field-Programmable Gate Arrays (FPGAs)

– FPGAs are integrated circuits that contain arrays of **configurable logic blocks** that can be wired together via **re-programmable interconnects**. They interface to the outside world via **I/O Blocks**.





This is how an FPGA looks like from the outside

Course Overview

Field-Programmable Gate Arrays (FPGAs)

- FPGAs are integrated circuits that contain arrays of **configurable logic blocks** that can be wired together via **re-programmable interconnects**. They interface to the outside world via **I/O Blocks**.
    - This allows FPGAs to be reprogrammed in the field (hence the name ***field-programmable***)
    - Individual logic blocks can be configured to perform basic combinatorial logic operations (e.g., AND, OR and XOR operations) and store information using registers.
    - Multiple logic blocks can be combined through series and/or parallel interconnections to perform complex digital signal processing algorithms.

Course Overview

Why should you take FPGAs seriously?

They are quickly becoming a core technology in automotive, telecommunications, defence, high-performance computing, 4K video-capture and real-time processing, high-frequency trading, intelligent systems (4th industry revolution)… everywhere, basically.

There are also many opportunities available right now for talented embedded systems engineers with experience developing FPGAs.

An increasing number of Australian companies now exploit the capabilities of FPGAs and microprocessors including CEA Technologies, Solinnov, Liquid Instruments, Seeing Machines, Vivian-Court, Metamako, Advanced Navigation and Black Magic Design, many of whom are growing and actively searching for talented embedded systems engineers, specifically those with FPGA experience.

- https://www.cea.com.au
- https://www.solinnov.com.au
- https://www.liquidinstruments.com
- https://www.seeingmachines.com
- https://www.vivcourt.com.au
- https://www.metamako.com
- https://www.advancednavigation.com.au
- https://www.blackmagicdesign.com/au

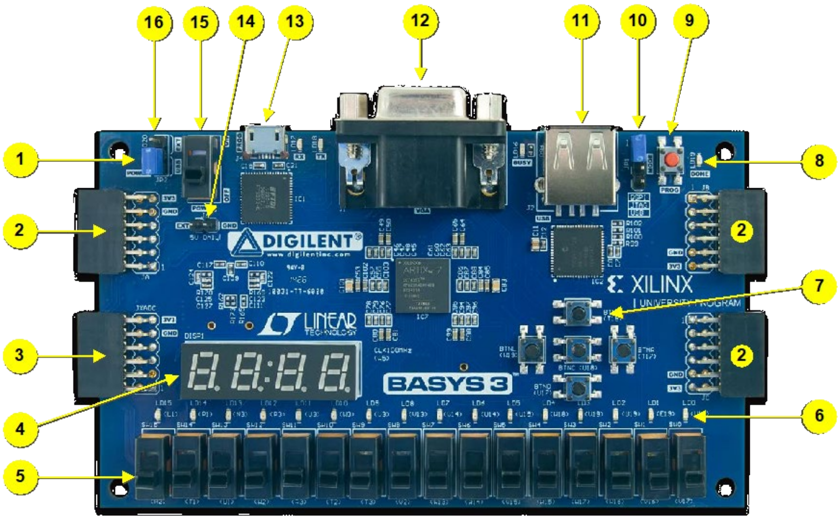
Course Overview

FPGAs – Major Manufacturers

- Xilinx (now AMD) and Intel (Altera) – 90% of Market share.
- Both Xilinx and Altera provide electronic design software which enables engineers to design, analyze, simulate and compile their designs.
- Others (Lattice Semiconductors, Microchip Technology, Gowin Semiconductors)

Course Overview

Xilinx Digilent’s Basys 3 Artix-7 FPGA Trainer Board



Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod port(s)	10	Programming mode jumper
3	Analog signal Pmod port (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

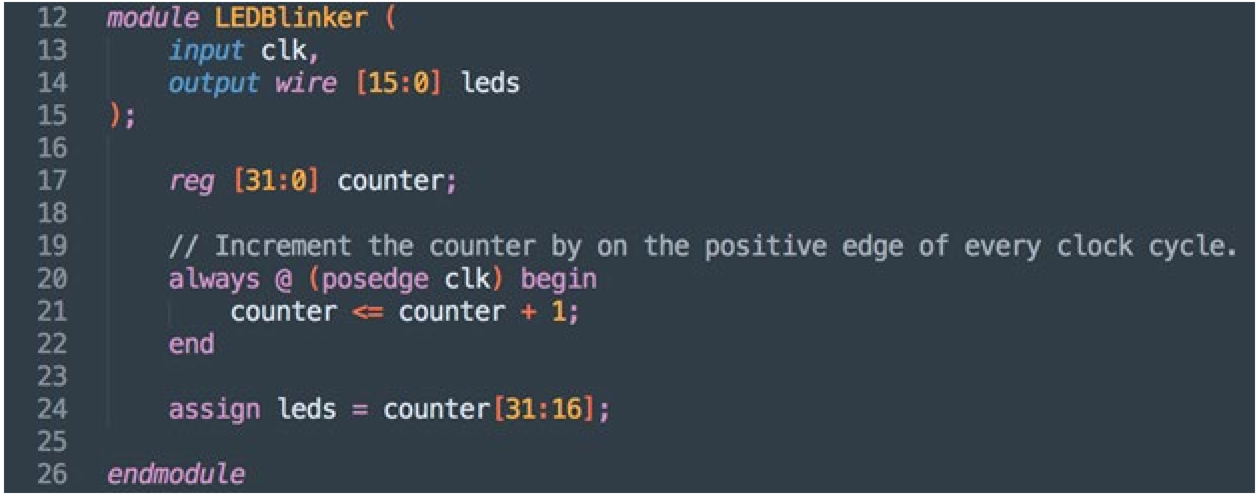
Source: Basys 3 FPGA Board Reference Manual

Course Overview

Hardware Description Language

- FPGAs are configured using code written in a **hardware description language (HDL)** such as Verilog and VHDL, it is translated into hardware level logic by a **synthesizer**.
- In this course, you will be learning how to code in Verilog.

Verilog looks like this…

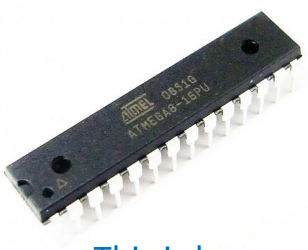


```
module LEDBlinker (
    input clk,
    output wire [15:0] leds
);

    reg [31:0] counter;

    // Increment the counter by on the positive edge of every clock cycle.
    always @ (posedge clk) begin
        counter <= counter + 1;
    end

    assign leds = counter[31:16];

endmodule
```

Course Overview

Microprocessors

- Microprocessors are special type of digital systems designed to perform a limited set of operations called **instructions**
  - Instructions are basic general operations, such as performing sums, subtractions, accessing memory locations, etc.
  - Many basic instructions can be combined in sequence to perform complex tasks. Programming languages facilitate humans in creating algorithms that can be fed to microprocessors.
  - The same microprocessor can carry out different tasks through different algorithms



This is how a microprocessor looks like from the outside

Course Overview

FPGAs vs. Microprocessors

- FPGAs have ***parallel and deterministic architectures***, making them well suited to low-latency, high-throughput applications that would otherwise not be implementable using a central processing unit (CPU).

- Certain numerical operations (e.g., floating-point division) are better suited to algorithmic resolution, e.g., through arithmetic logic units found in CPUs. This is just one of a number of reasons why today ***CPUs are often deployed alongside FPGAs in embedded applications***.

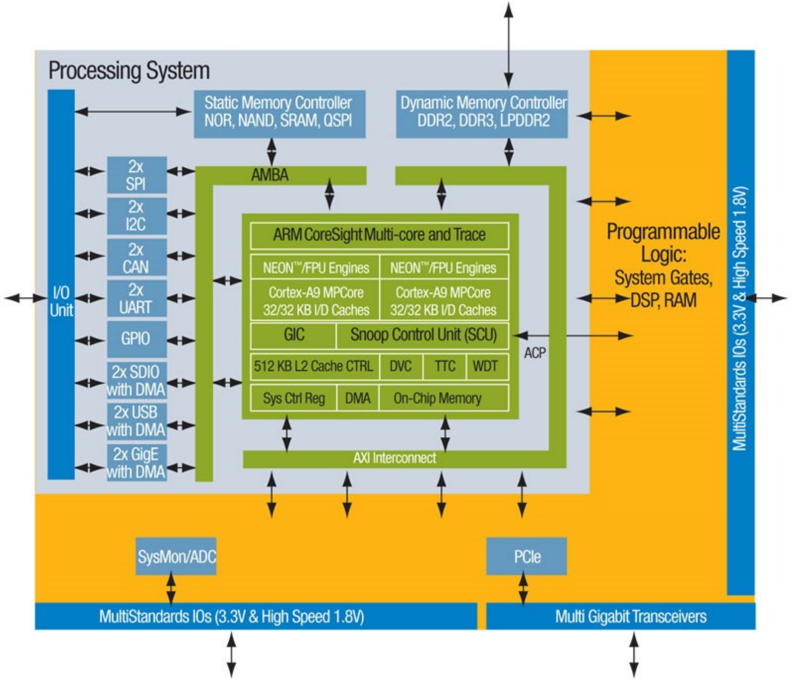
- The popularity of combining CPU and FPGA architectures ultimately led to the development of ***System-on-Chip (SoC) FPGAs, which combine a microprocessor (such as an ARM Cortex-A9) and FPGA on the same silicon chip***.

Course Overview

FPGAs vs. Microprocessors

FPGAs and microprocessors are ***not*** competing technologies.

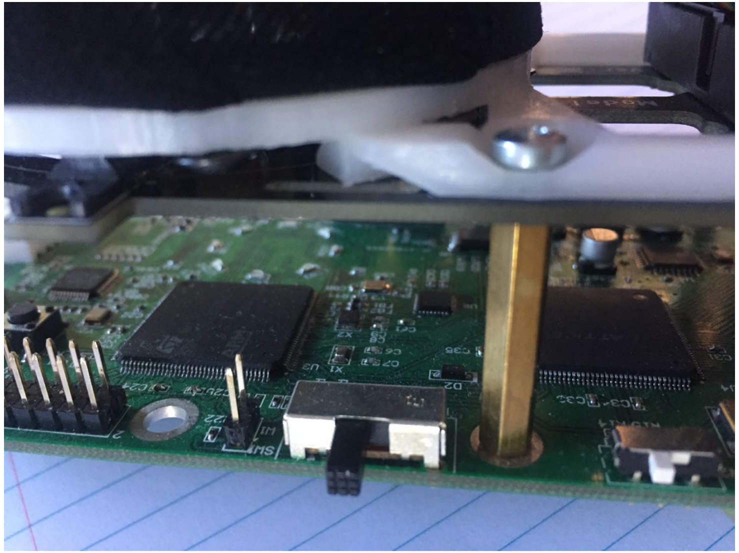
In fact, they synergise so well that it is now rare to find an FPGA unaccompanied by a microprocessor or CPU of some kind. This is why it is now important that you gain experience with both.



Course Overview

FPGAs vs. Microprocessors

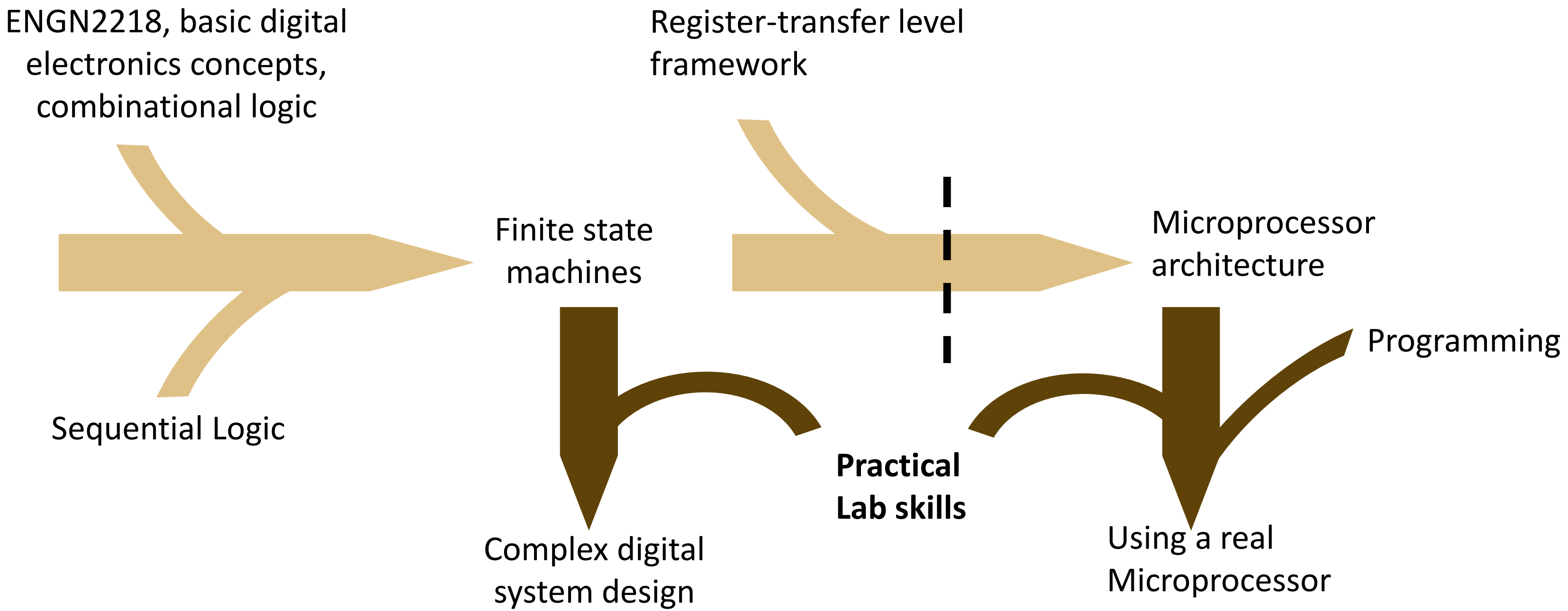
FPGAs and microprocessors are ***not*** competing technologies.



Course Overview

Logical structure

Where do you come from and where are you heading?



ENGN2218, basic digital electronics concepts, combinational logic

Sequential Logic

Finite state machines

Register-transfer level framework

Microprocessor architecture

Programming

Complex digital system design

Practical Lab skills

Using a real Microprocessor

Suggestions for Study

- This course is different to most (if not all) ENGN courses you have completed at ANU. **Your learning in this course is mainly done by hands-on practice** through labs and projects.
- Do your **homework regularly**
  - Listen to the lecture recordings and study the lecture slides regularly, especially to prepare for the lab tasks.
  - Install necessary software in your personal PC, so that you can learn at your convenience
  - Read and attempt lab tasks and tutorial problems ahead of time.
- **Self-learning**: As the course progresses, you must increasingly learn yourselves to work based on higher-level instructions.
  - To do this, develop your skills to find information from lengthy data sheets, search relevant websites effectively to see how others have used similar hardware and solved similar applications
- Spend at least **10 hours per week** and try to be ahead. ***Unlike other theory-based subjects, it is very difficult to catch up with hands-on learning.***
  - No individual topic in this course is hard. The course as a whole, however, has a lot of topics. It will be **very hard** if you do not keep up.

Suggestions for Study

- **FPGA and Microcontroller projects**
  - Start looking for your project partner as soon as possible
  - Read the project document thoroughly and clearly identify the deliverables and assessment criteria
  - Start the project activities as early as possible and make regular progress
  - Make use of project drop-in sessions for getting effective guidance and feedback
  - You will not be able to complete the project tasks if you start just few days before the due date

ENGN4213/6213 Class Representatives

Class Student Representation is an important component of the teaching and learning quality assurance and quality improvement processes within the ANU College of Engineering, Computing and Cybernetics (CECC).

The role of Student Representatives is to provide ongoing constructive feedback on behalf of the student cohort to Course Conveners and to Associate Directors (Education) for continuous improvements to the course

Roles and responsibilities:

- Act as the official liaison between your peers and convener.
- Be available and proactive in gathering feedback from your classmates.
- Attend regular meetings and provide reports on course feedback to your course convener.
- Close the feedback loop by reporting back to the class the outcomes of your meetings.

ENGN4213/6213 Class Representatives

– Why become a class representative?

- **Ensure students have a voice** to their course convener, lecturer, tutors, and College.
- **Develop skills sought by employers**, including interpersonal, dispute resolution, leadership and communication skills.
- **Become empowered**. Play an active role in determining the direction of your education.
- **Become more aware of issues influencing your University** and current issues in higher education.
- **Course design and delivery.** Help shape the delivery of your current courses as well as future improvements for following years.

Note: Class representatives will need to be comfortable with their contact details being made available via Wattle to all students in the class.

Want to be a class representative? Nominate today!

Please nominate yourself to the course conveners (*A/Prof. Nan Yang & A/Prof. Xiangyun Zhou*) through email by **5pm on Friday 3 March**. In this email, please indicate your name and Uni ID, your enrolled course (ENGN4213 or ENGN6213), and a brief justification on why you are suitable for this role.

ANU Access & Inclusion (A&I)

- The ANU A&I is a team of Disability and Equity Advisors (DEA) who support students within the ANU community whose participation in academic studies is impacted by physical and learning disabilities, mental health conditions, chronic medical conditions and short-term illnesses/conditions. They also support carers, international under 18 students and elite athletes
- The Team provides advice, evidence based supports and adjustments to minimize the impact on an ANU student's studies. The Team also helps students to participate fully in their program of study, be academically successful and achieve their personal best through the development of an Education Access Plan
- Please check the A&I website if you have such needs

THANK YOU

Acknowledgement

These presentation slides are the modified version of slides prepared by Dr. Jihui (Aimee) Zhang based on the original version by Dr. Nicolo Malagutti

Contact:

Amy Bastine
School of Engineering
ANU College of Engineering, Computing and Cybernetics
The Australian National University

Office: B147, Brian Anderson Building (Bldg. 115)
Email: amy.bastine@anu.edu.au