香港中文大學
The Chinese University of Hong Kong

CENG3430 Rapid Prototyping of Digital Systems

Lecture 00: Course Information

Ming-Chang YANG

mcyang@cse.cuhk.edu.hk

CENG3430 Course Information

- **CENG3430 Rapid Prototyping of Digital Systems**

- **Course Time and Place**
  - **Lecture** (*2)
    - MON 12:30~14:15 (@SC LG23)
  - **Lab** (*2)
    - WED 12:30~14:15 (@SHB 102)
    - Note: Attendance at both lab sessions is mandatory!

- **Course Website**
  - https://blackboard.cuhk.edu.hk/

Course Instructor & Teaching Assistants

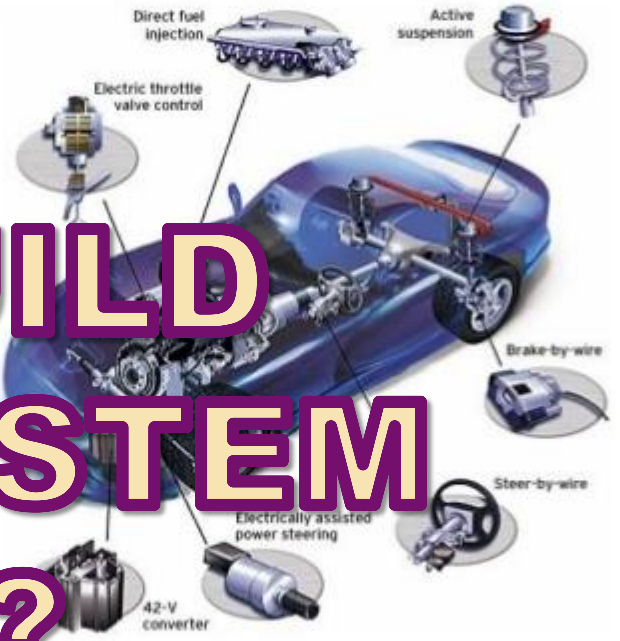
- **Course Instructor**
  - Prof. Ming-Chang YANG (楊明昌)
    - Office: SHB 906
    - Email: mcyang@cse.cuhk.edu.hk

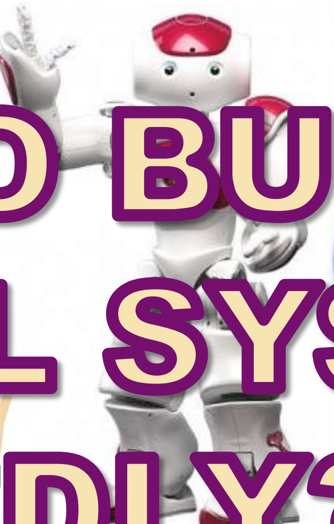
- **Teaching Assistant & Lab Helper**
  - Han ZHAO (趙涵)
    - Office: SHB 921
    - Email: hzhao@cse.cuhk.edu.hk

  - Zhirui ZHANG (張知睿)
    - Email: 1155174021@link.cuhk.edu.hk

We are surrounded by digital systems!

- **Mass Products**
  - Media players
  - Mobile phones

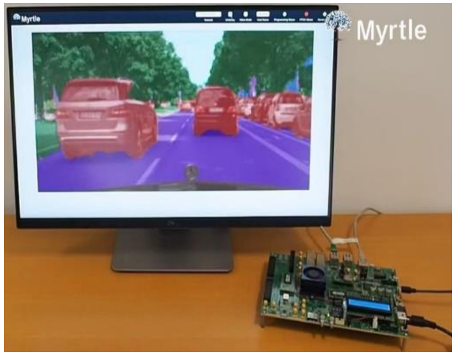




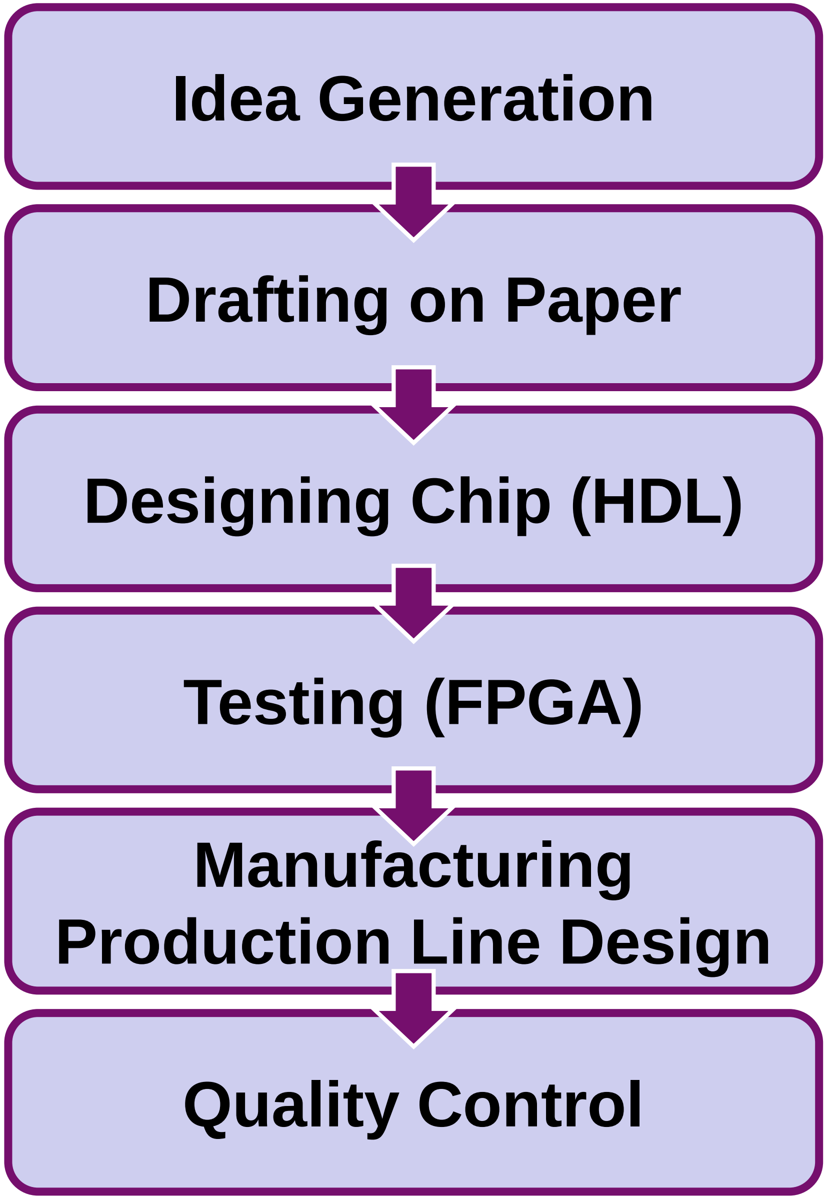
- **Novel Products**
  - Wearable devices
  - Robots

HOW TO BUILD A DIGITAL SYSTEM RAPIDLY?

- **Research**
  - Real time edge detection
  - Deep learning acceleration



Common Design Flow of Digital System


Idea Generation

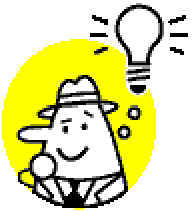
Drafting on Paper

Designing Chip (HDL)

Testing (FPGA)

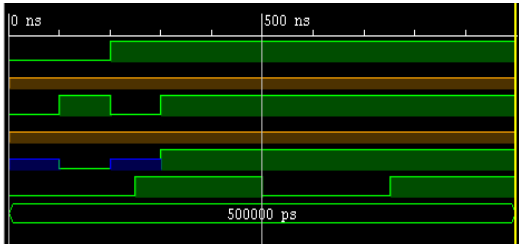
Manufacturing Production Line Design

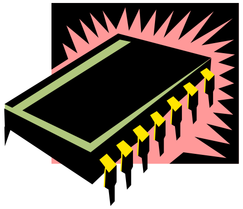
Quality Control



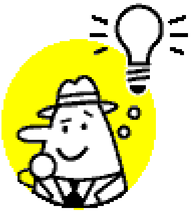
Ex: VHDL AND-Gate Program

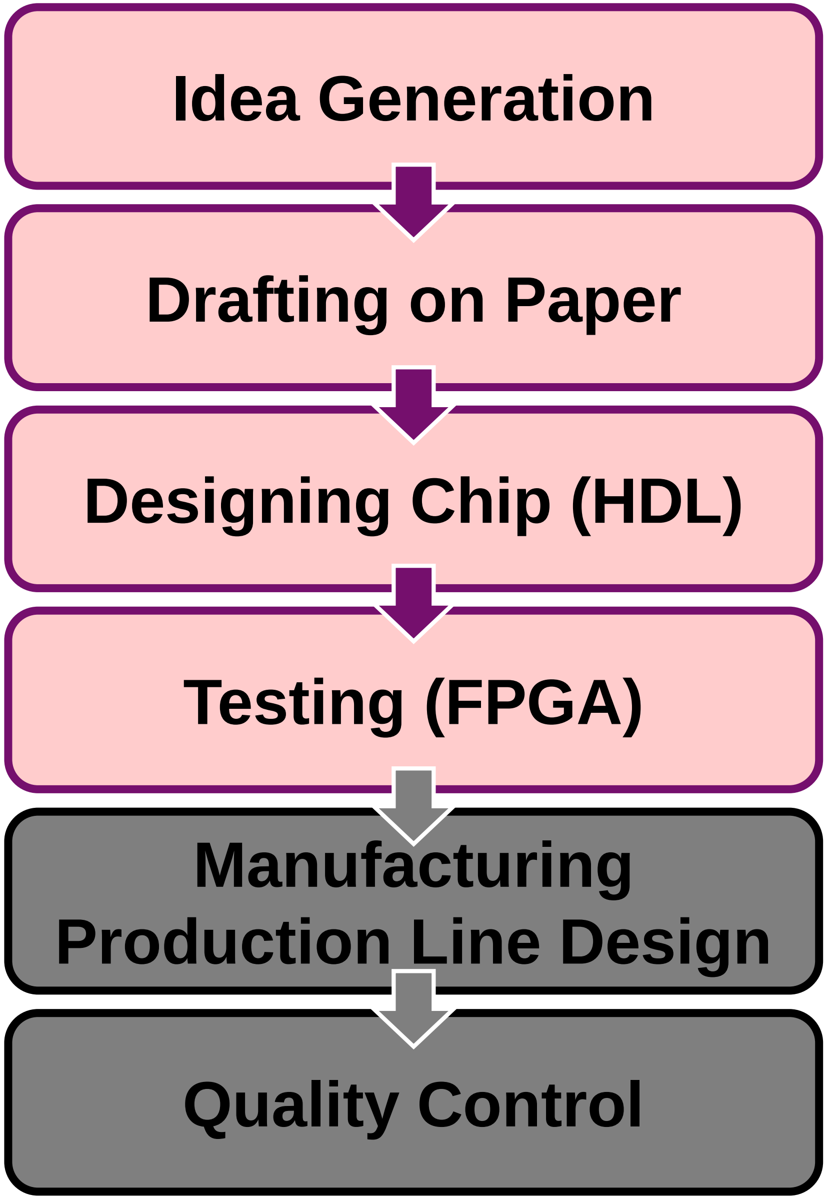
```
entity and2 is
port (a,b: in std_logic;
          c: out std_logic);
end and2
architecture arch of and2
begin
    c <= a and b;
end and2_arch
```





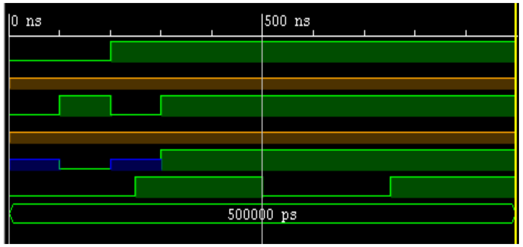
Our Focus: Prototyping

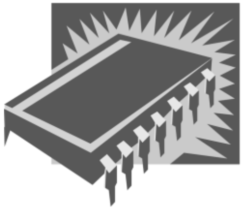




Ex: VHDL AND-Gate Program

```
entity and2 is
port (a,b: in std_logic;
        c: out std_logic);
end and2
architecture arch of and2
begin
   c <= a and b;
end and2_arch
```





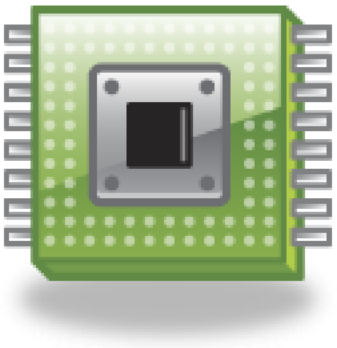
What We Will Learn: HDL + FPGA

- **Software**: Hardware Description Language (HDL)

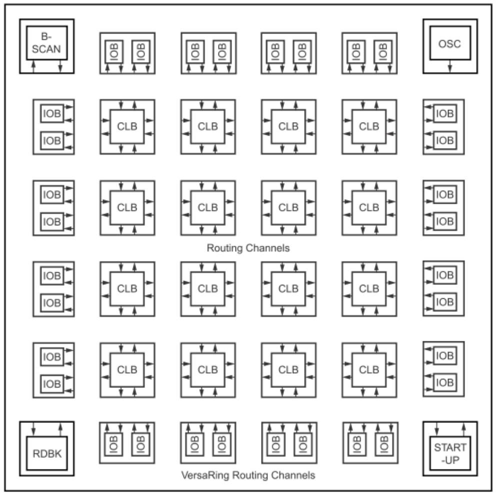
Ex: VHDL AND-Gate Program

```
entity and2 is
port (a,b: in std_logic;
        c: out std_logic);
end and2
architecture and2_arch of and2
begin
   c <= a and b;
end and2_arch
```

Write HDL code, then it will generate the hardware chip automatically

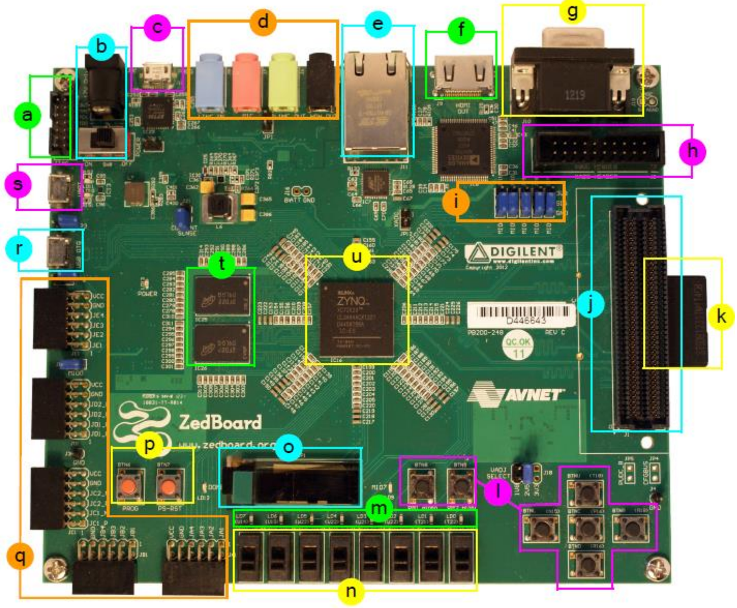


- **Hardware**: Field Programmable Gate Array (FPGA)
  - The hardware can be reprogrammable.
  - Designs can be changed easily.
  - No additional hardware manufacturing cost is needed.



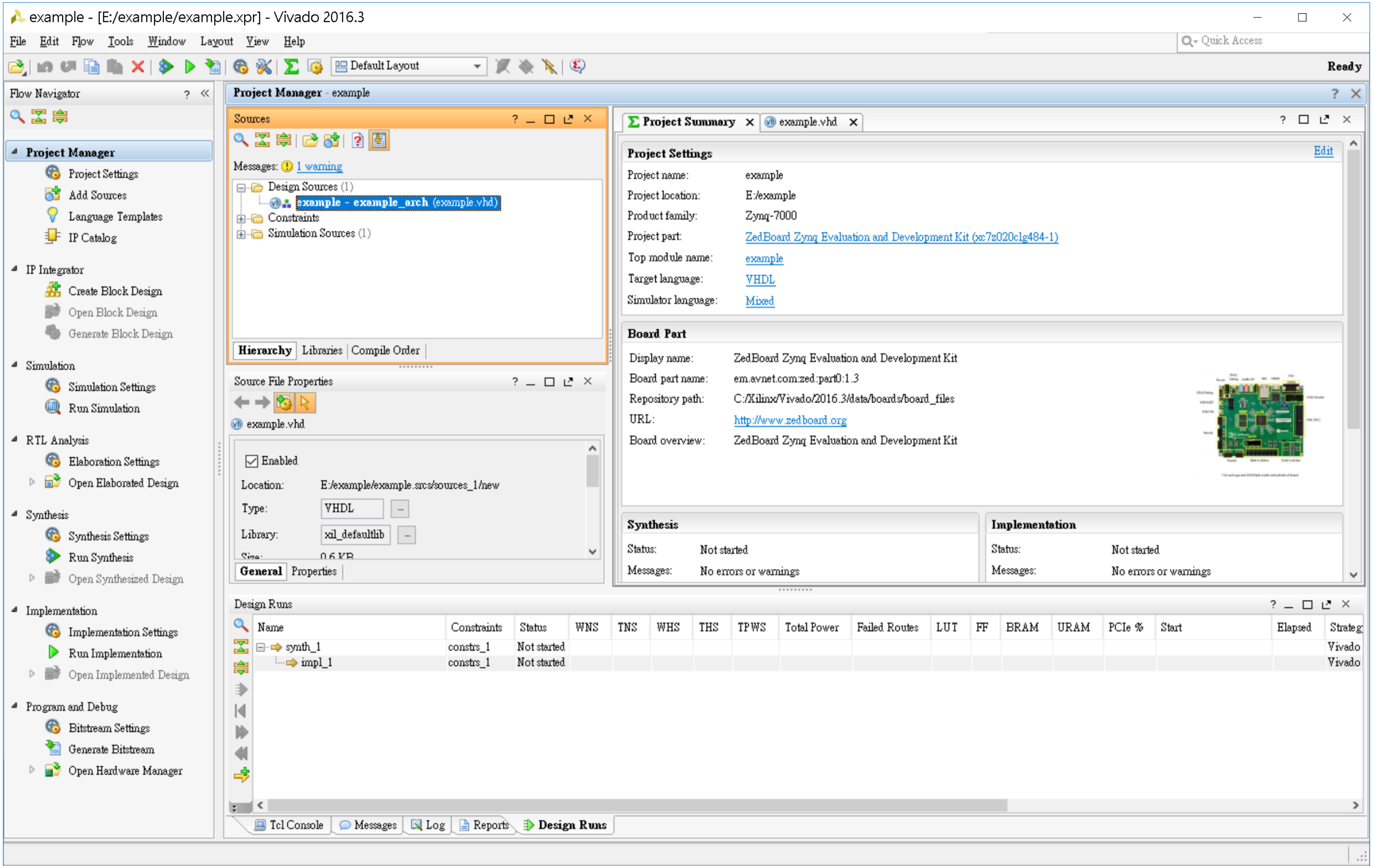
What We Will Use: VHDL + ZedBoard

- **Software**: Vivado WebPACK™ Edition (FREE)
  - It supports Windows or Linux operating systems.
  - **Hardware Description Language**: Very-High-Speed-Integrated-Circuits Hardware Description Language (**VHDL**)
- **Hardware**: Zynq ZedBoard
  - Dual-core ARM Cortex-A9 with traditional FPGA



a	Xilinx JTAG connector
b	Power input and switch
c	USB-JTAG (programming)
d	Audio ports
e	Ethernet port
f	HDMI port (output)
g	VGA port
h	XADC header port
i	Configuration jumpers
j	FMC connector
k	SD card (underside)
l	User push buttons
m	LEDs
n	Switches
o	OLED display
p	Prog & reset push buttons
q	5 x Pmod connector ports
r	USB-OTG peripheral port
s	USB-UART port
t	DDR3 memory
u	Zynq device (+ heatsink)

Software: Vivado WebPACK™ Edition



Messages: 1 warning

☑ Enabled

Location: E:/example/example.srcs/sources_1/new

Type: VHDL

Library: xil_defaultlib

Size: 0.6 KB

Project Settings

Project name: example

Project location: E:/example

Product family: Zynq-7000

Project part: ZedBoard Zynq Evaluation and Development Kit (xc7z020clg484-1)

Top module name: example

Target language: VHDL

Simulator language: Mixed

Board Part

Display name: ZedBoard Zynq Evaluation and Development Kit

Board part name: em.avnet.com:zed:part0:1.3

Repository path: C:/Xilinx/Vivado/2016.3/data/boards/board_files

URL: http://www.zedboard.org

Board overview: ZedBoard Zynq Evaluation and Development Kit

Synthesis

Status: Not started

Messages: No errors or warnings

Implementation

Status: Not started

Messages: No errors or warnings

Design Runs

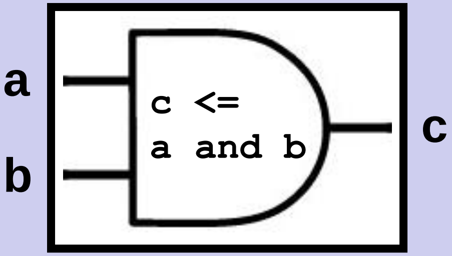
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	PCIe %	Start	Elapsed	Strateg
synth_1	constrs_1	Not started															Vivado
impl_1	constrs_1	Not started															Vivado

Hardware Description Language: VHDL

- **An Example: AND-Gate in VHDL**

Entity Declaration: Define I/Os

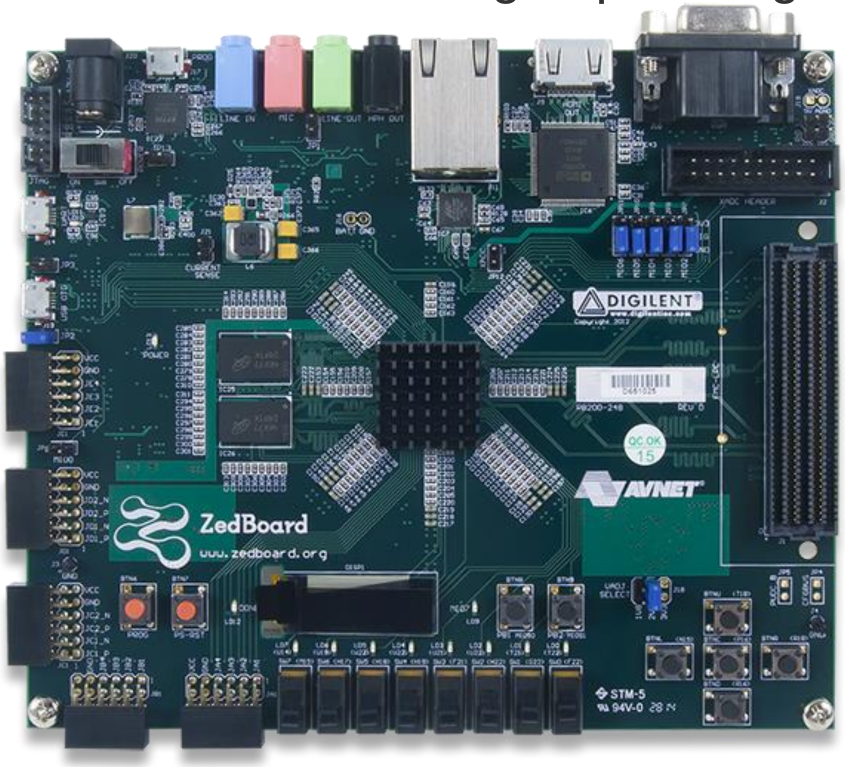
```
entity and2 is
port (a,b: in std_logic;
        c: out std_logic);
end and2
architecture and2_arch of and2
begin
   c <= a and b;
end and2_arch
```

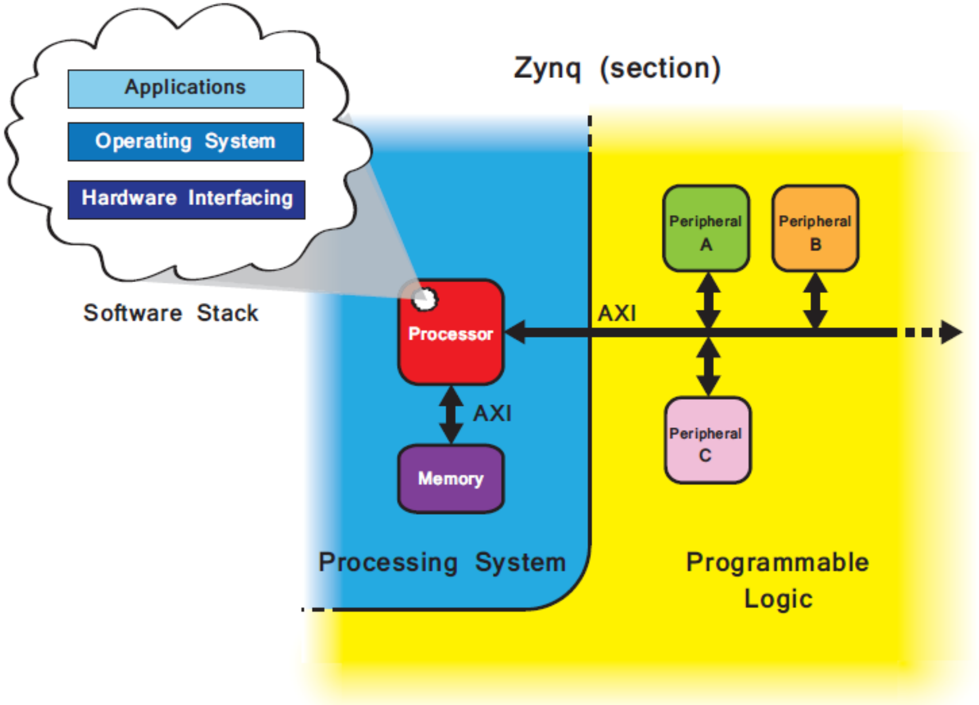


Architecture Body: Define functions

Hardware: Zynq ZedBoard

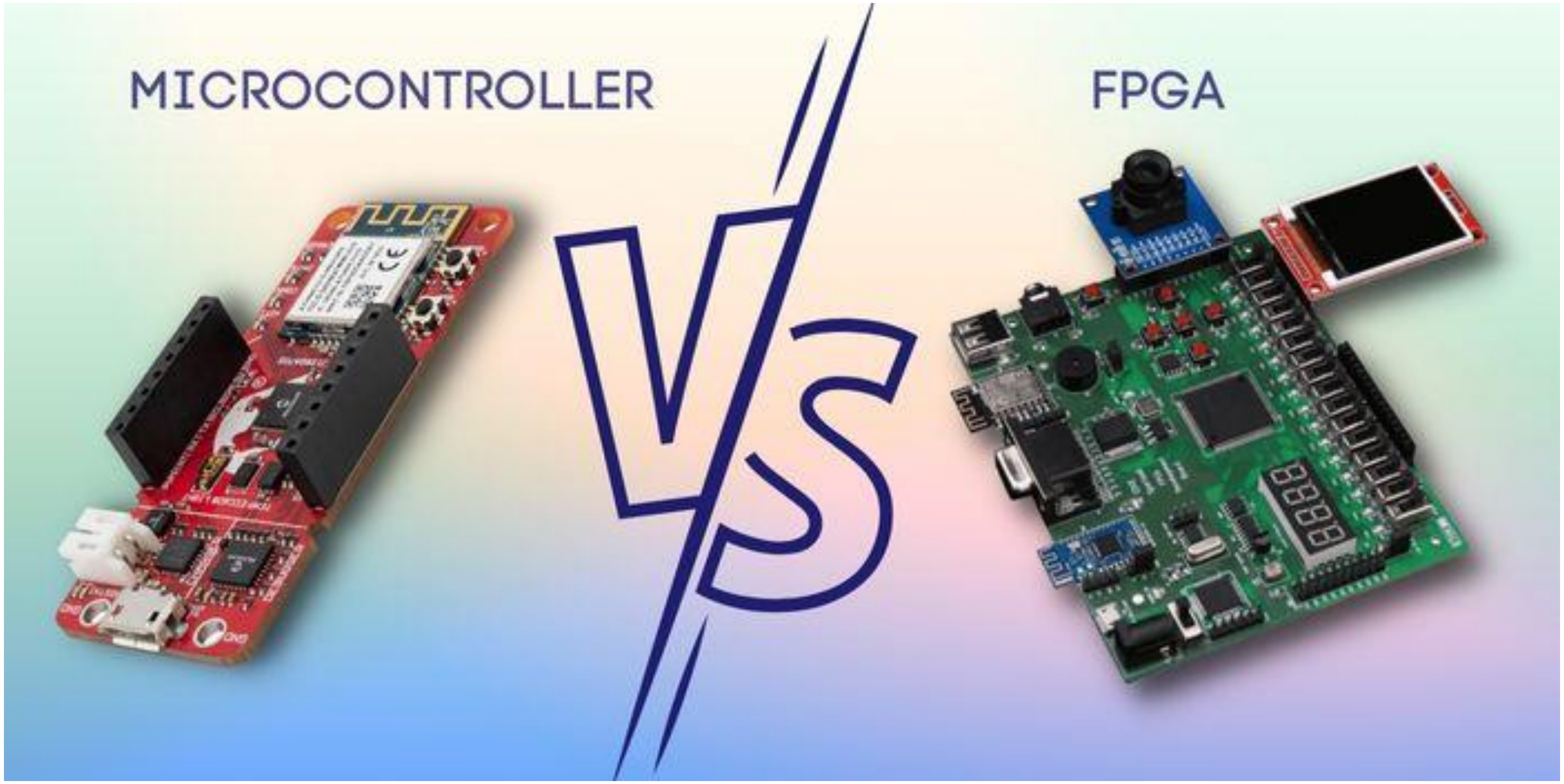
- **Zynq ZedBoard** combines
  - Processing System (PS): Dual-core ARM Cortex-A9 CPU
    - Supports software routines and/or operating systems
  - Programmable Logical (PL): Equivalent to trad. FPGA
    - Ideal for high-speed logic, arithmetic and data flow subsystems





Why FPGA (over Microcontroller)

- There is no golden rule!
  - FPGAs offer flexibility, programmability, and high-speed performance but come with higher cost and complexity.



Write Software VS. Design Hardware

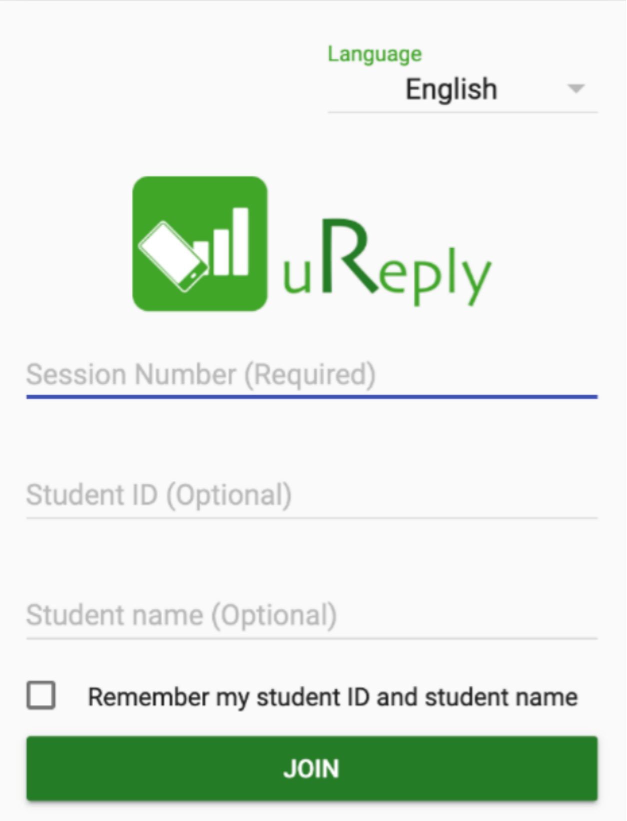
Course Assessment

- **Grading** *(subject to changes)*
  - **No Exam! 0%**
  - **Class Participation 10%**
    - Note: Unable to attend? Raise leave requests via email.
  - **Weekly Lab Exercises 40%**
    - Note: Required to submit individually.
  - **Final Project 50%**
    - Note: At most two students in a group.

- **Note**
  - Late submission is NOT acceptable (unless otherwise approved before the regular deadline).

Class Participation? uReply

1) Enter the **Session Number**



Language: English

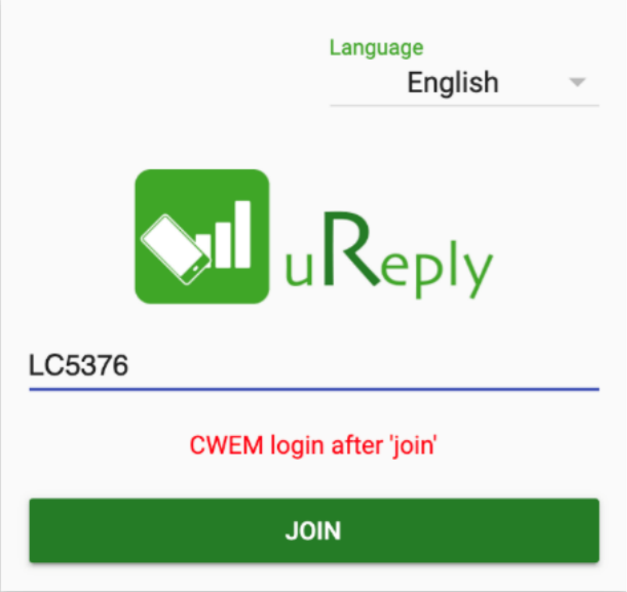
Session Number (Required)

Student ID (Optional)

Student name (Optional)

□ Remember my student ID and student name

2) Confirm the Session Number and Click **“Join”**

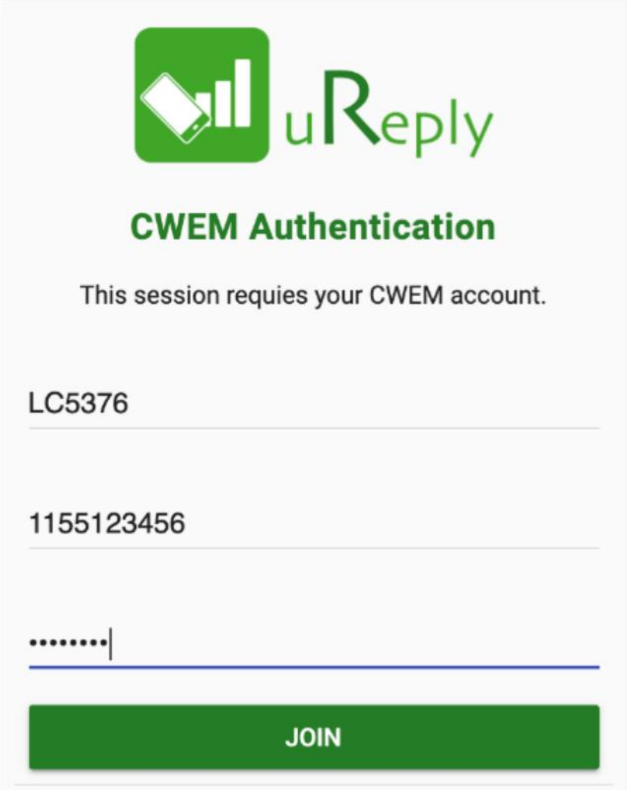


Language: English

LC5376

CWEM login after 'join'

3) Login with **Student ID** and **CWEM Password**



CWEM Authentication

This session requies your CWEM account.

LC5376

1155123456

•••••••••

[uReply Attendance User Guide]

Course Schedule *(subject to changes)*

W	Date	Lecture	Lab
1	Jan. 6, 8	Lec00: Course Information	Lec01: Introduction to VHDL
2	Jan. 13, 15	Lec02: Introduction to ZedBoard	Lab01: Vivado & Software Simulation
3	Jan. 20, 22	Lec03: Comb. Circuit and Seq. Circuit	Lab02: First Program on ZedBoard
4	Jan. 27, 29	Rescheduled to Jan. 8 (No Lecture)	Lunar New Year Vacation (No Class)
5	Feb. 3, 5	Lunar New Year Vacation (No Class)	Lab03: Shift Register
6	Feb. 10, 12	Lec04: Finite State Machine	Lab04: Finite State Machine
7	Feb. 17, 19	Lec05: Driving VGA Display	Lab05: Driving VGA Display
8	Feb. 24, 26	Lec06: Driving Peripheral Modules	Lab06: Driving Peripheral Modules
9	Mar. 3, 5	Reading Week (No Class)	Public Holiday (No Lab)
10	Mar. 10, 12	Lec07: Integration of ARM and FPGA	Lab07: Integration of ARM and FPGA
11	Mar. 17, 19	Lec08: High Level Synthesis	Lab08: High Level Synthesis
12	Mar. 24, 26	Lec09: Embedded Operating System	Lab09: Embedded Operating System
13	Mar. 31, Apr. 2	Lec10: VHDL versus Verilog	Lab10: Verilog Exercise
14	Apr. 7, 9	Final Project: Proposal Presentation (I)	Final Project: Proposal Presentation (I)
15	Apr. 14, 16	Final Project: Feedback on Proposal (I)	Final Project: Feedback on Proposal (II)
Mid of May (TBA)		Final Project Final Submission Deadline (Demo Video and Report)	

Full Final Project List (2023-24)

1. Multiplayer Billiards 🥈
2. Kahoot! - Wireless Communication App
3. Cooking with Chef Zed
4. Navigating Robot
5. Chopsticks Trick
6. RailOpti: FPGA-based Rail Route Optimization
7. BreakOut Clone
8. Keep Talking and Nobody Explodes
9. Smart Pet Feeding Machine
10. MIDI Music Synthesizer 🏆
11. Taiko Simulator
12. Air Fighter 🥈
13. Checkers
14. QR Code Scanner
15. VGA Game Tetris
16. The Spooktacular Surprise
17. Maze Runner
18. ZedPlant 🏆
19. Auto Car Control System
20. Fruit Catcher
21. Downward Escaping Game
22. The Day of Sagittarius

Full Final Project List (2022-23)

1. Space Race
2. Printed Digits Recognition
3. People Counting
4. Rhythm Game - Osu!
5. Dog Tamer
6. Sign Language Learning Bot 🏆
7. Whac A Number
8. Digital Clawing Machine
9. 3D Object Renderer
10. Dance Machine
11. Simon Game
12. Pong Game 🏆
13. Hot Drinks on The Go 🏆
14. Automatic Teller Machine
15. 2D Aircraft Game
16. Automate Car
17. Yelling Game
18. Debomb
19. Smart Air Purifier
20. Brick Breaker Game
21. Computer Vision on Zedboard 🏆

Full Final Project List (2021-22)

1. Hit to Dance
2. Digital Companion Pet 🏆
3. Wordle
4. Pixel Space Invader
5. CUHKO
6. Smart Home Device
7. Breakout Clone
8. Vending Machine 🏆
9. Mario with ZedBoard
10. Vending Machine
11. Color Vision Game
12. Vending Machine
13. Math Game
14. Mirrored Snake
15. Audio Recording & Playback
16. Don’t Stop Photon
17. Color Memory Game
18. T-Rex Runner
19. Sliding Puzzle Game
20. Health Monitoring Cell
21. Snack Game
22. Duck Hunt 🏆
23. Sliding Game
24. FPonGA
25. Le Cheers

Full Final Project List (2020-21)

1. Digital Locker 🏆
2. Weather Report Generation 🏆
3. Reaction Game
4. Snake Game
5. Audition (Rhythm Game)
6. Kernel Convolution
7. GOMOKU
8. Football Shooting Game
9. Catch the Thief
10. The Snake Game
11. FLAPPY BIRD
12. Mota Game
13. Flappy Bird
14. Snake Game

Full Final Project List (2019-20)

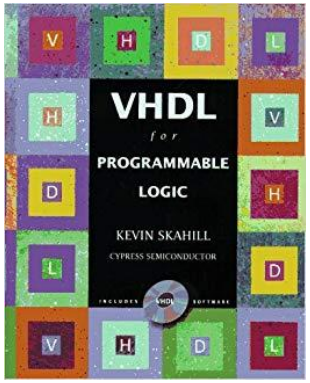
1. AVG Game
2. Door Locking System
3. The Flappy Bird 🏆
4. Immigration System
5. Traffic Horn Punishment
6. How Fast Is Your Finger
7. Mastermind
8. Drum Looper
9. Gobang
10. Morse Code from Torch
11. Indoor Monitoring Station
12. Rhythm Game
13. Cat Dog Fight Game
14. Portable Freezer 🏆
15. Color Detector
16. Mastermind
17. Tic Tac Toe
18. UFO Catcher
19. Sound Recorder

Full Final Project List (2018-19)

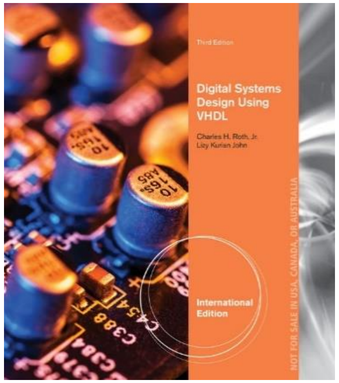
1. Piano and Music Player
2. Color Recognition
3. RGB Meter
4. Lie Detector 🏆
5. Snake Battle
6. Space War
7. The Dodge Game
8. Space Impact
9. Get It at Once
10. Elevator 🏆
11. Super Pads
12. Tetris
13. Morse Code Interpreter
14. The Flash
15. Multifunctional Display
16. Rolling Down!

References

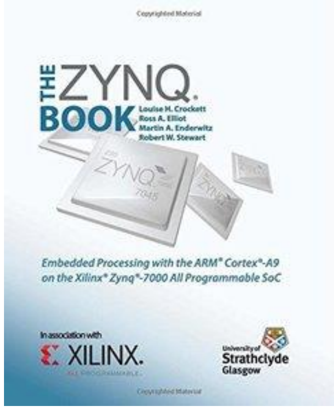
- **VHDL for Programmable Logic**
  - Kevin Skahill
  - Addison-Wesley



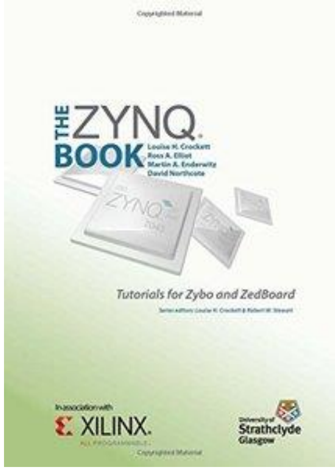
- **Digital Systems Design Using VHDL**
  - Charles H. Roth Jr., Lizy Kurian John
  - Cengage Learning



- **The Zynq Book**: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc
  - Louise H Crockett, Ross A Elliot, Martin A Enderwitz, Robert W Stewart
  - Strathclyde Academic Media


THE ZYNQ BOOK
Louise H. Crockett
Ross A. Elliot
Martin A. Enderwitz
Robert W. Stewart
Embedded Processing with the ARM® Cortex®-A9 on the Xilinx® Zynq®-7000 All Programmable SoC
In association with
XILINX
University of Strathclyde Glasgow

- **The Zynq Book**: Tutorials for Zybo and ZedBoard
  - Louise H Crockett, Ross A Elliot, Martin A Enderwitz
  - Strathclyde Academic Media



Important Notes

- **Plagiarism** will **NOT** be tolerated!
  - Do **NOT** copy!
  - Do **NOT** let other(s) copy!
  - **Can** discuss but write up the solutions by yourself!

- **Honesty** in Academic Work: A Guide
  - http://www.cuhk.edu.hk/policy/academichonesty/

The best way to learn is through