ECE 448
FPGA Design with VHDL

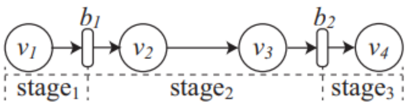
Spring 2024

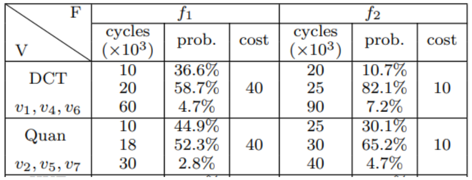
Weiwen [way-when] **Jiang** [jee-aang] /dʒiˈæŋ/

Research and teaching interests:

- SW/HW co-design
- FPGA design
- High-level synthesis
- Edge Machine Learning
- Quantum computing

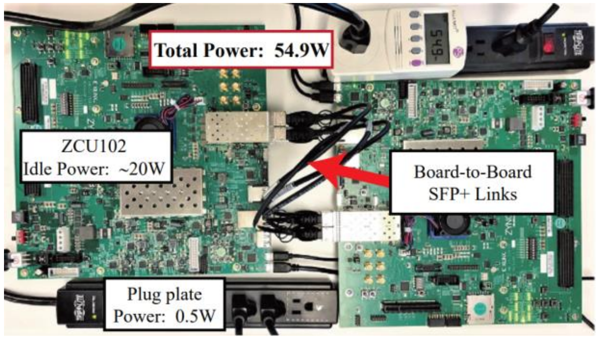
2013 - 2017





V \ F	f_1 cycles ($\times 10^3$)	f_1 prob.	f_1 cost	f_2 cycles ($\times 10^3$)	f_2 prob.	f_2 cost
DCT v_1, v_4, v_6	10 20 60	36.6% 58.7% 4.7%	40	20 25 90	10.7% 82.1% 7.2%	10
Quan v_2, v_5, v_7	10 18 30	44.9% 52.3% 2.8%	40	25 30 40	30.1% 65.2% 4.7%	10

2017 - 2021



Vivado™ HLS

2021 - now

Contact:

- wjiang8@gmu.edu
- **Subject: ECE448 …**

Course Web Page

https://jqub.ece.gmu.edu/2024/01/01/24SECE448/

ECE 448 FPGA Design with VHDL

TA Team

Miguel Medina

Lab TA

mmedin23@gmu.edu

Telecommunications and Computer Engineering graduate student

Corey Kaiser

Lab TA

ckaiser2@gmu.edu

Computer Engineering Graduate Student

Kabir Basnet

Course TA

kbasnet4@gmu.edu

Bachelors in Computer Engineering

Software Engineer

Course Hours

Lecture:

Monday, Wednesday
12:00-13:15 PM
Exploratory Hall L003

Lab Sessions:

Wednesday	**Friday**	**Monday**
7:20-10:00 PM	**8:40-11:20 AM**	**9:00-11:40AM**
Jan. 24	**Jan. 26**	**Jan. 29**
TA: Miguel	**TA: Corey**	**TA: Corey**

Location: ENGR 3208

General Section Assignment Rules

- You should do your best to attend **all lab meetings** of the section you are registered for

- If you have missed a meeting of your section, please attend a meeting of another section

In Person Office Hours

Instructor: Dr. Weiwen Jiang
Monday, Wednesday, 2:00-3:00 PM
ENGR 3247

TA: Corey Kaiser
Monday 3:00-5:00 PM,
Thursday 2:00-4:00 PM
ENGR 3208

TA: Miguel
TBD

Getting Help Outside of Office Hours

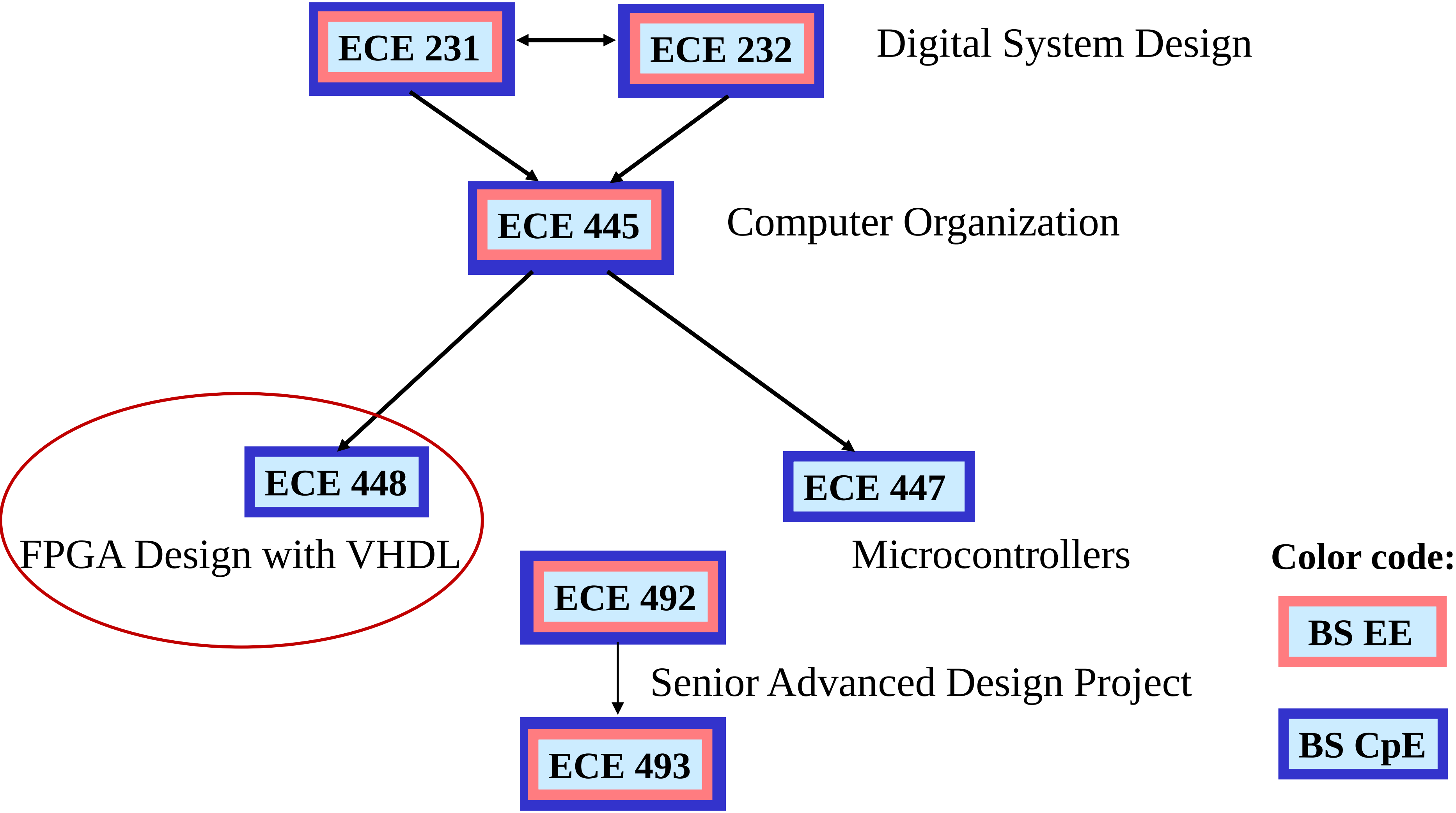
piazza

- **System for asking questions 24/7**
- **Answers can be given by students and instructors**
- **Student answers endorsed (or corrected) by instructors**
- **You can submit your questions anonymously**
- **You can ask private questions visible only to the instructors**

[Signup] https://piazza.com/gmu/spring2024/ece448
[Code] vba328kibjl

Contact: Kabir Basnet (kbasnet4@gmu.edu)

Undergraduate Computer Engineering Courses



ECE 231

ECE 232

Digital System Design

ECE 445

Computer Organization

ECE 448

FPGA Design with VHDL

ECE 447

Microcontrollers

ECE 492

Senior Advanced Design Project

ECE 493

Color code:

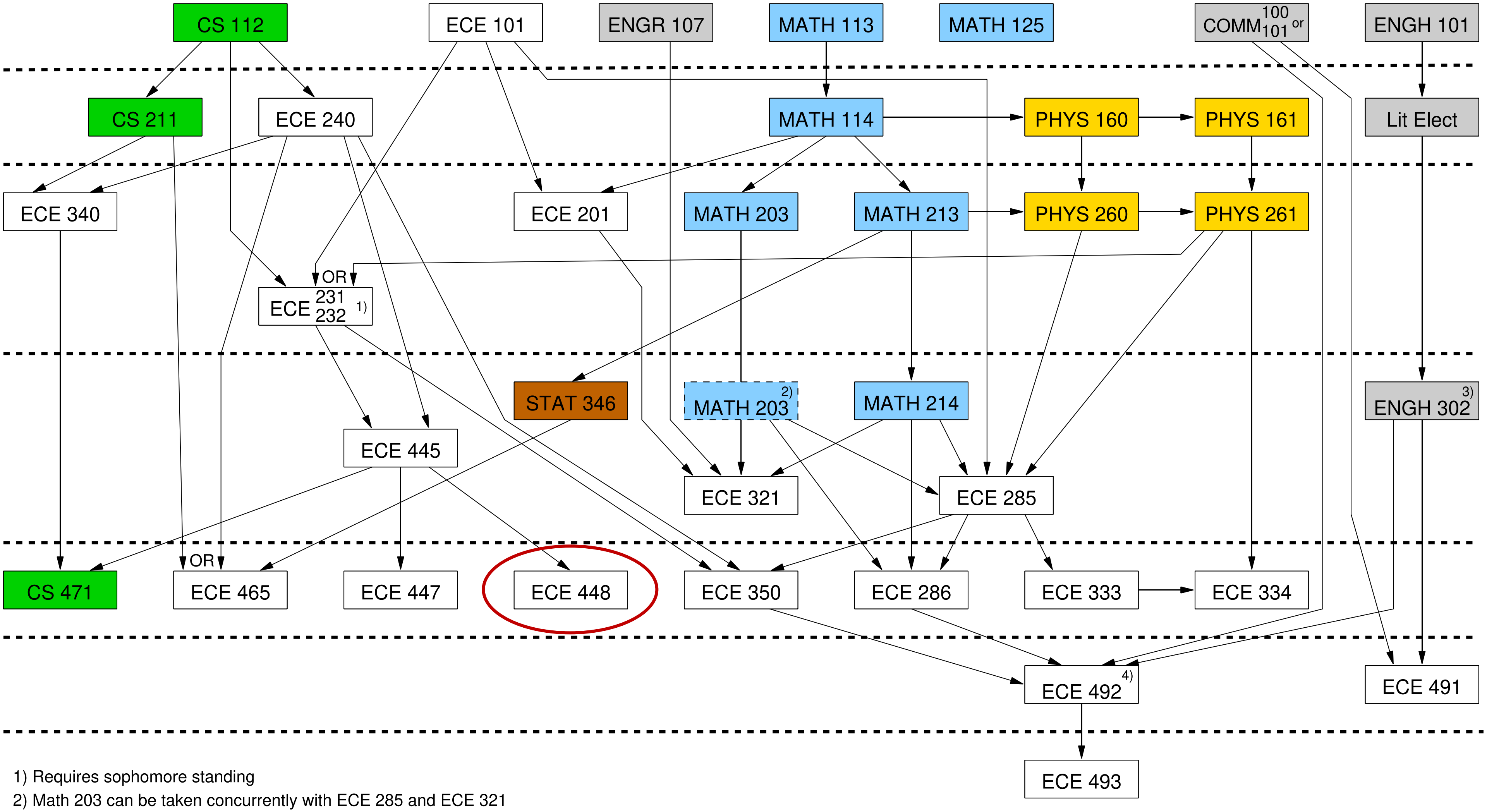
BS EE

BS CpE

Computer Engineering Progression of Core Courses

This is not a suggested schedule. It only illustrates dependencies and shows courses in earliest possible semester.

Not all courses required for the degree are shown.



1) Requires sophomore standing

2) Math 203 can be taken concurrently with ECE 285 and ECE 321

3) Requires completion of 45 credits

4) Requires completion of 90 credits applicable to your major and all 200 level coursework

Catalog Year 2022/2023

Prerequisite
Must be taken in sequence

Co–Requisite
Should be taken con–currenlty but not earlier

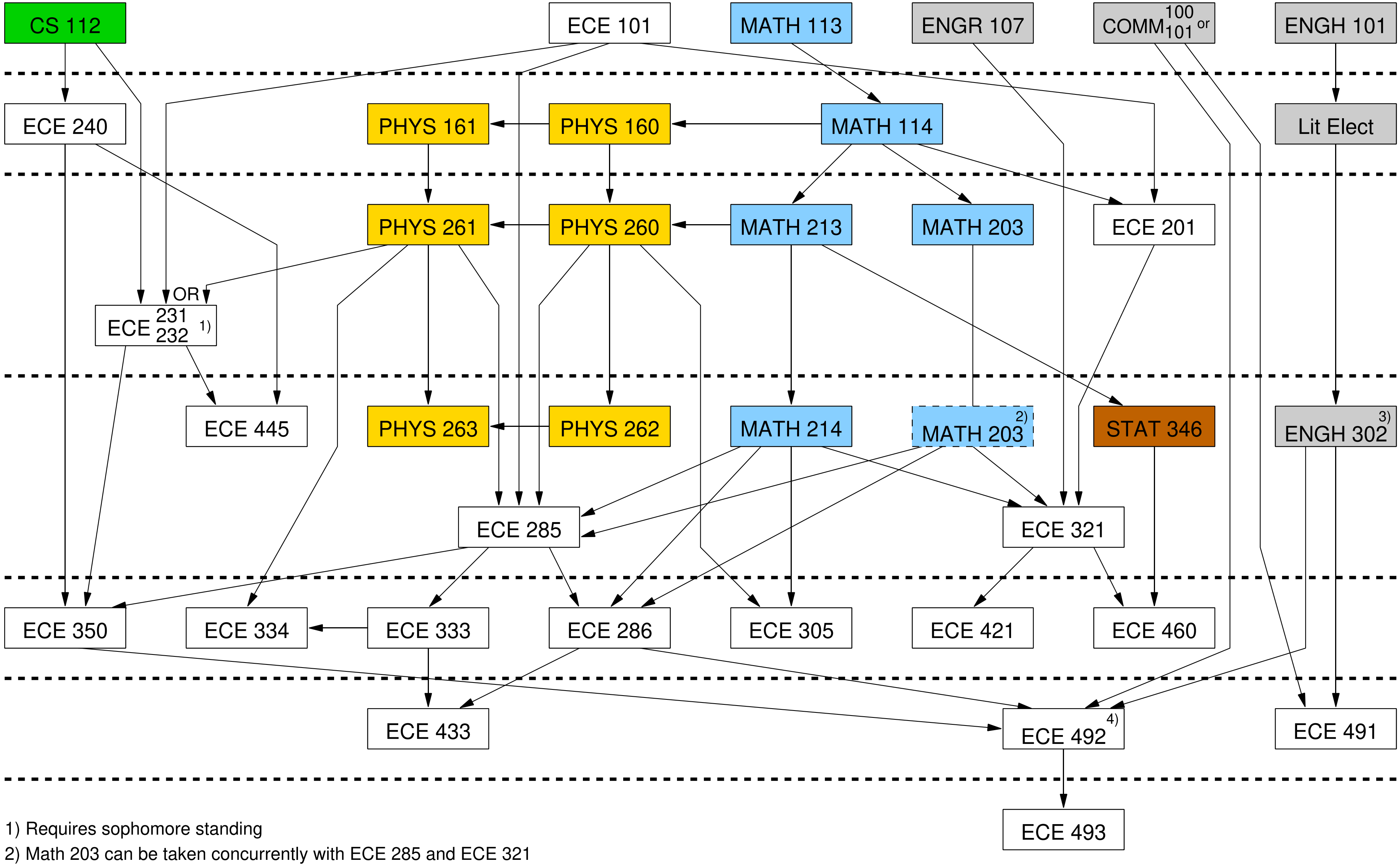
Co–Requisite +
Suggested to be taken in sequece

Semester
Courses between dashed lines can be taken concurrently

Electrical Engineering Progression of Core Courses

This is not a suggested schedule. It only illustrates dependencies and shows courses in earliest possible semester.

Not all courses required for the degree are shown.



1) Requires sophomore standing

2) Math 203 can be taken concurrently with ECE 285 and ECE 321

3) Requires completion of 45 credits

4) Requires completion of 90 credits applicable to your major and all 200 level coursework

Catalog Year 2022/2023

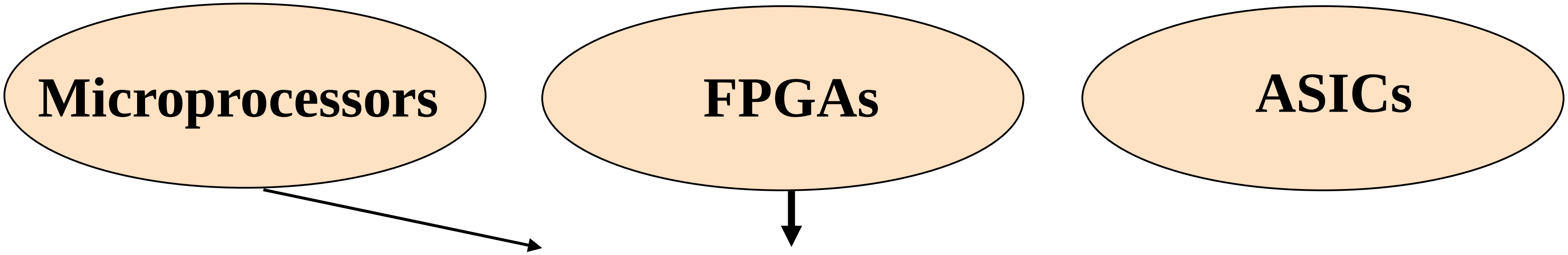
Prerequisite
Must be taken in sequence

Co–Requisite
Should be taken con–currenlty but not earlier

Co–Requisite +
Suggested to be taken in sequece

Semester
Courses between dashed lines can be taken concurrently

Digital system design technologies coverage in the CpE & EE programs at GMU



Microprocessors

FPGAs

ASICs

ECE 448 FPGA Design with VHDL

ECE 350 Embedded Systems and Hardware Interfaces

ECE 445 Computer Organization

ECE 447 Microcontrollers

ECE 431 Digital Circuit Design

ECE 511 Computer Architecture

ECE 611 Advanced Computer Architecture

ECE 612 Real-Time Embedded Systems

ECE 615 Software/Hardware Codesign

ECE 545 Digital System Design with VHDL

ECE 554 Machine Learning for Embedded Systems

ECE 505 Hardware Security

ECE 618 Hardware Accelerators for Machine Learning

ECE 586 Digital Integrated Circuits

ECE 681 VLSI Design for ASICs

Course Web Page

https://jqub.ece.gmu.edu/2024/01/01/24SECE448/

Organization

Instructor
Teaching Assistant
Office Hours
Grading

Lecture

Textbooks
Lecture Slides
Homework
Midterm Exams
Final Exams

Lab

Rules
Lab Assignments
Lab Slides & Examples
Software
Hardware
Useful References
Lab Exams

Grading criteria

First part of the semester:

Lab experiments - Part I
25%

Lab exercises – Part I
2%

Quizzes & homework: 3%

Midterm exam for the lecture: 10%
Midterm exam for the lab: 15%

Second part of the semester:

Lab experiments - Part II
15%

Lab exercises – Part II
2%

Quizzes & homework: 3%

Final exam
25%

Tentative Grading Scheme for the Labs

Lab 1: Developing VHDL Testbenches – 5 points

Lab 2: Combinational Logic – 5 points

Lab 3: Sequential Logic – 5 points

Lab 4: State Machines. Basic I/O Devices. – 10 points

Total: 25 points

Lab 5: MicroBlaze – Basic I/O Devices

Lab 6: MicroBlaze – Computer Graphics

Total: 15 points

Penalties and Bonus Points

Penalties:

one-week delay: **1/4 of points**

i.e., you can earn max. 7.5 out of 10 points

No submissions or demos will be accepted more than one week after the assignment is due!

Flexibility in the Second Part of the Semester

Schedule A+:

- **Intended for students who do very well in the first part of the semester (preferably ≥ 90% of points for Labs 1-4)**
- **An open-ended project proposed by students, having a discussion with the TAs and/or the instructor**
- **Can be done individually or in groups of two students**
- **Schedule:**

Detailed Specification	**(1 week)**
Milestone 1	**(2 weeks)**
Milestone 2	**(2 weeks)**
Final Report & Deliverable	**(1 week)**

Total: 20 points

Exams

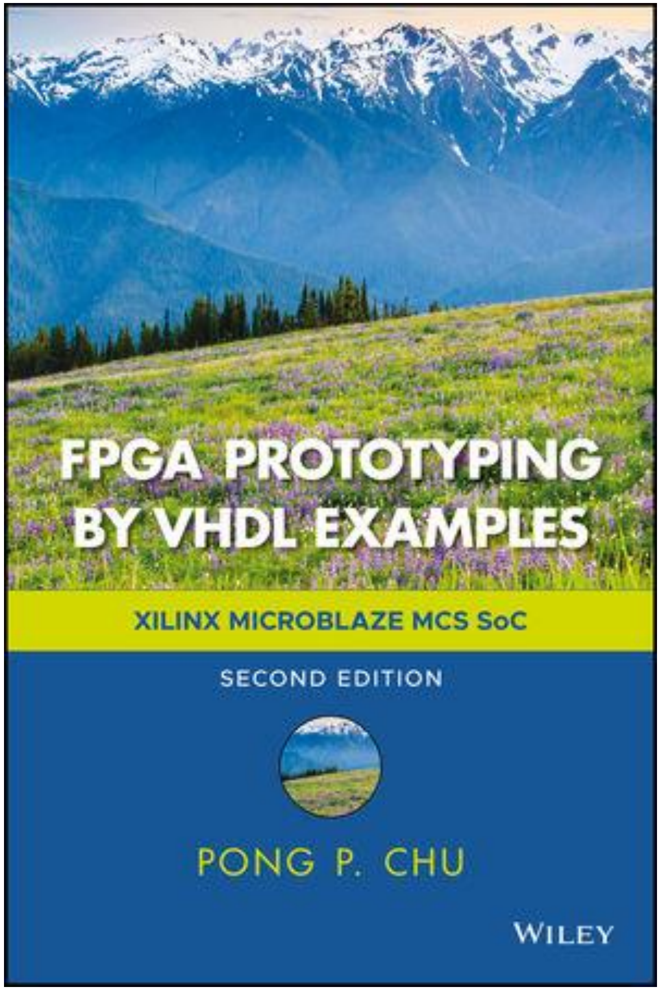
- **Midterm Exam for the Lecture – 10 points**
  Wednesday, Feb. 28, 12:00-13:15

- **Midterm Exam for the Lab (hands-on) – 15 points**
  March 13, 15, 18 --- According to which lab you selected

- **Final Exam – 25 points**
  Monday, May 06, 10:30 AM-1:15 PM

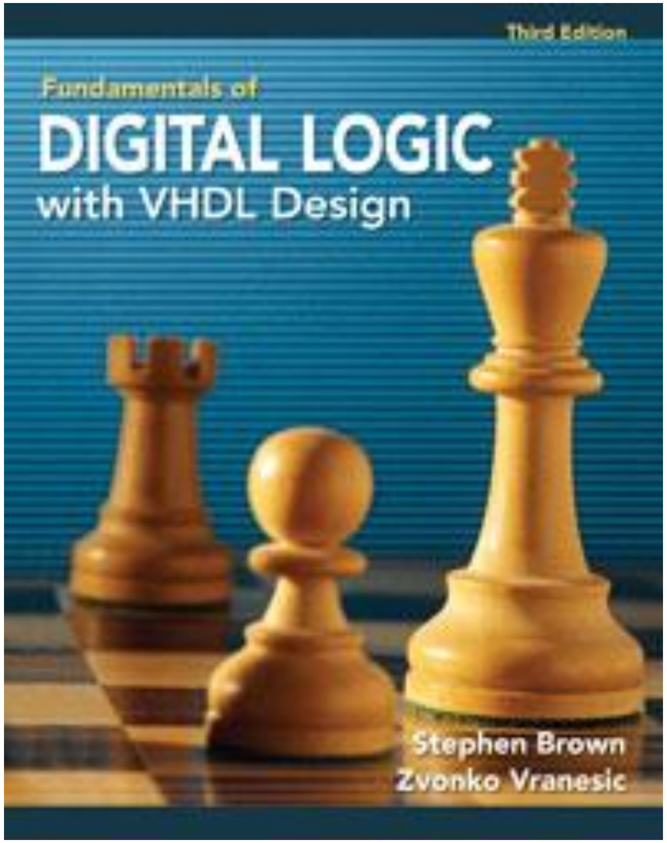
Required Textbook

Pong P. Chu, *FPGA Prototyping by VHDL Examples: Xilinx MicroBlaze MCS SoC*, Wiley, Oct. 2017.



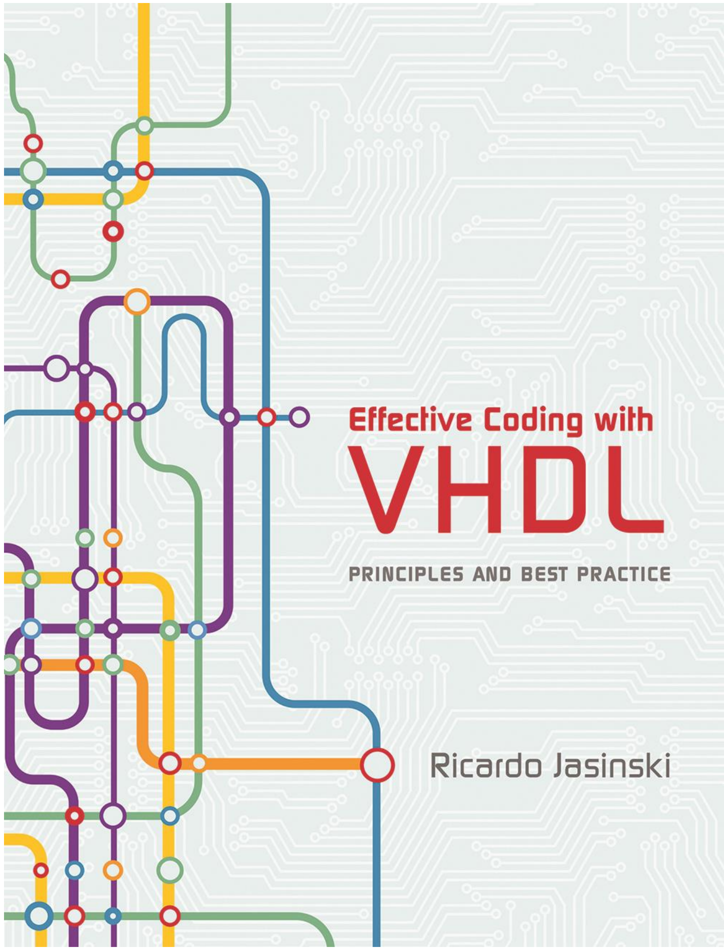
Supplementary Textbook – Basics Refresher

Stephen Brown and Zvonko Vranesic,
Fundamentals of Digital Logic with VHDL Design,
McGraw-Hill, 3rd Edition, 2008.

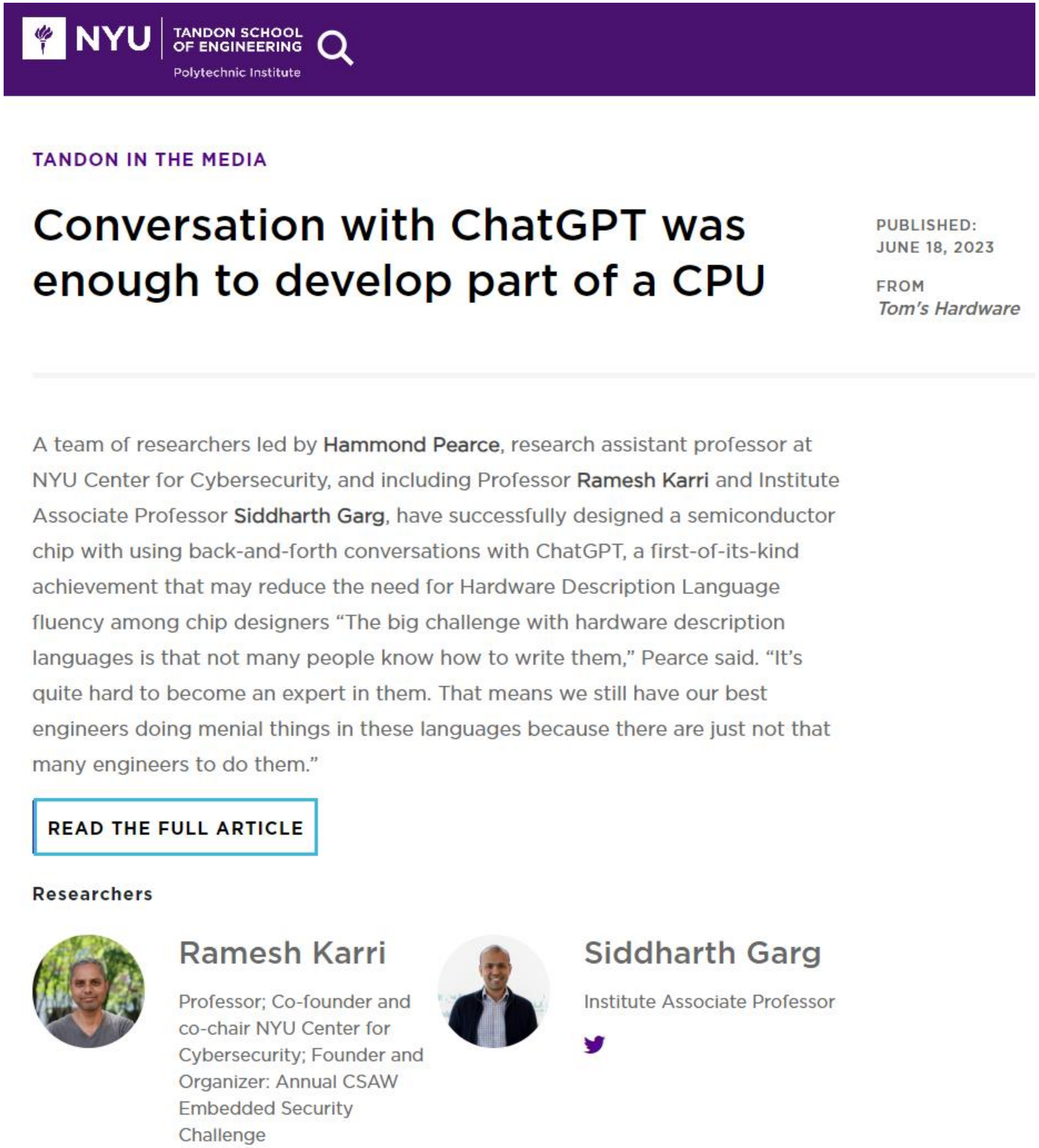


Supplementary Textbook – Advanced

Ricardo Jasinski, *Effective Coding with VHDL: Principles and Best Practice*,
The MIT Press; 1st Edition, 2016.



Supplementary: ChatGPT as an Assistant but not a Writer



TANDON IN THE MEDIA

Conversation with ChatGPT was enough to develop part of a CPU

PUBLISHED:
JUNE 18, 2023

FROM
Tom's Hardware

A team of researchers led by **Hammond Pearce**, research assistant professor at NYU Center for Cybersecurity, and including Professor **Ramesh Karri** and Institute Associate Professor **Siddharth Garg**, have successfully designed a semiconductor chip with using back-and-forth conversations with ChatGPT, a first-of-its-kind achievement that may reduce the need for Hardware Description Language fluency among chip designers "The big challenge with hardware description languages is that not many people know how to write them," Pearce said. "It's quite hard to become an expert in them. That means we still have our best engineers doing menial things in these languages because there are just not that many engineers to do them."

Researchers

Ramesh Karri

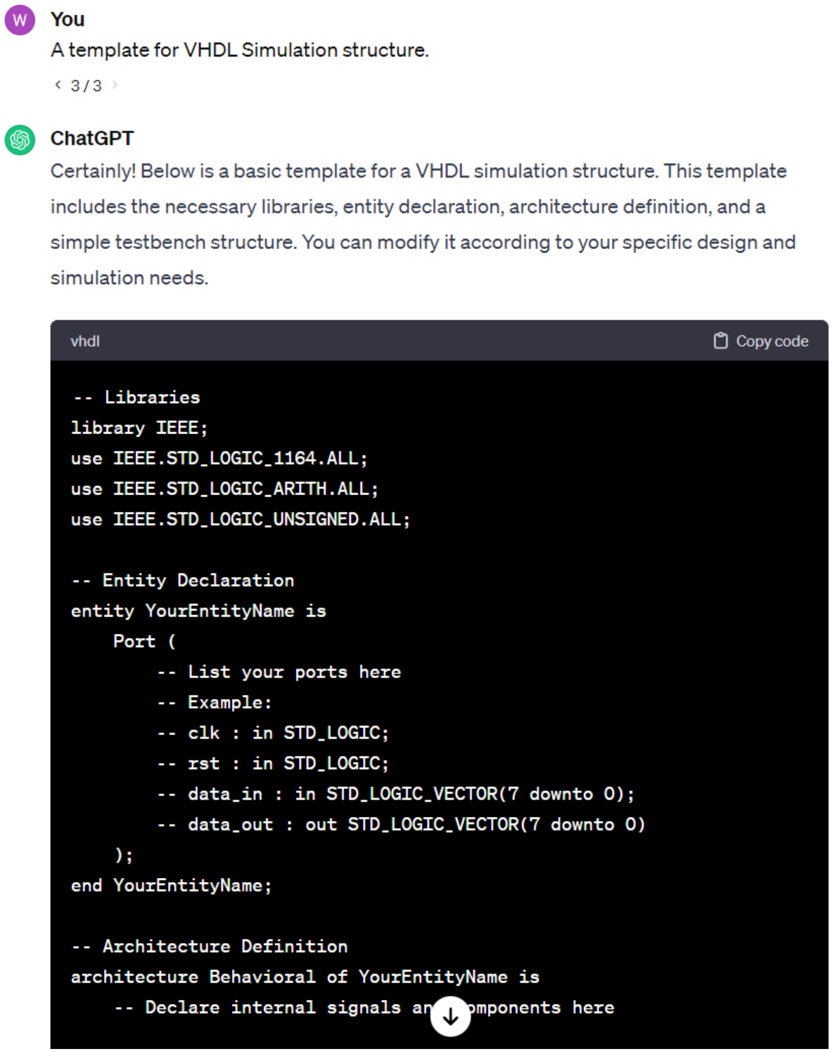
Professor; Co-founder and co-chair NYU Center for Cybersecurity; Founder and Organizer: Annual CSAW Embedded Security Challenge

Siddharth Garg

Institute Associate Professor

Supplementary: ChatGPT as an Assistant but not a Writer

Okay to use it for assistance



You

A template for VHDL Simulation structure.

ChatGPT

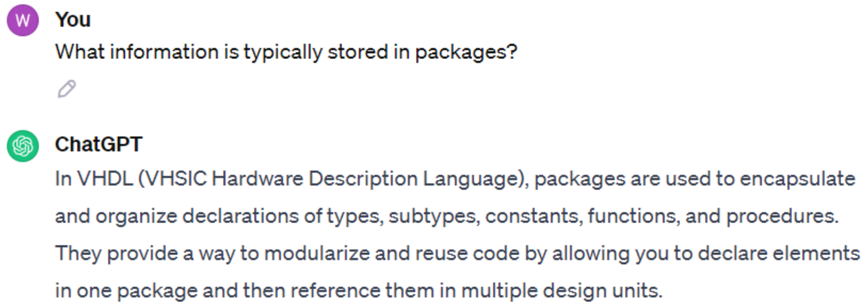
Certainly! Below is a basic template for a VHDL simulation structure. This template includes the necessary libraries, entity declaration, architecture definition, and a simple testbench structure. You can modify it according to your specific design and simulation needs.

```vhdl
-- Libraries
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Entity Declaration
entity YourEntityName is
    Port (
        -- List your ports here
        -- Example:
        -- clk : in STD_LOGIC;
        -- rst : in STD_LOGIC;
        -- data_in : in STD_LOGIC_VECTOR(7 downto 0);
        -- data_out : out STD_LOGIC_VECTOR(7 downto 0)
    );
end YourEntityName;

-- Architecture Definition
architecture Behavioral of YourEntityName is
    -- Declare internal signals an omponents here
```

Do not use it for writing lab, quiz, or homework



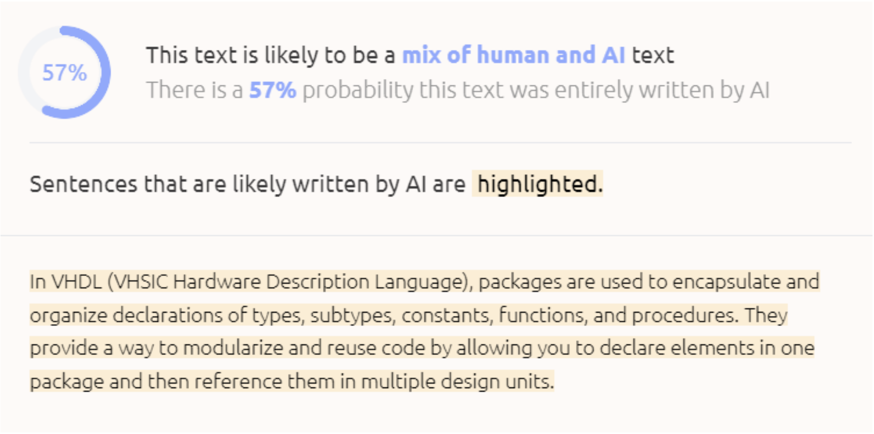
You

What information is typically stored in packages?

ChatGPT

In VHDL (VHSIC Hardware Description Language), packages are used to encapsulate and organize declarations of types, subtypes, constants, functions, and procedures. They provide a way to modularize and reuse code by allowing you to declare elements in one package and then reference them in multiple design units.

- **Generated stuff can be easily detected!**



This text is likely to be a **mix of human and AI** text

There is a **57%** probability this text was entirely written by AI

Sentences that are likely written by AI are highlighted.

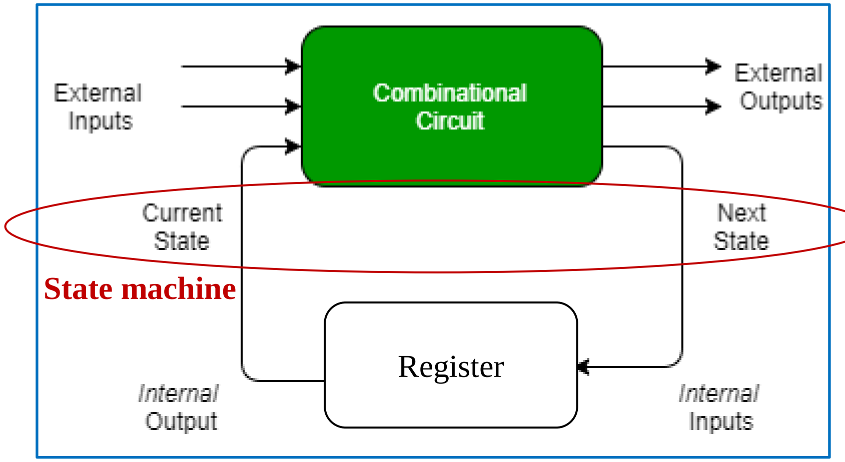
In VHDL (VHSIC Hardware Description Language), packages are used to encapsulate and organize declarations of types, subtypes, constants, functions, and procedures. They provide a way to modularize and reuse code by allowing you to declare elements in one package and then reference them in multiple design units.

ECE 448: FPGA Design with VHDL

Overview and Goal

Fundamentals:
Advanced digital system design with VHDL

Sequential circuit



Embedded Resources

Memory, DSP, etc.

Testbench

Process, variable, etc.

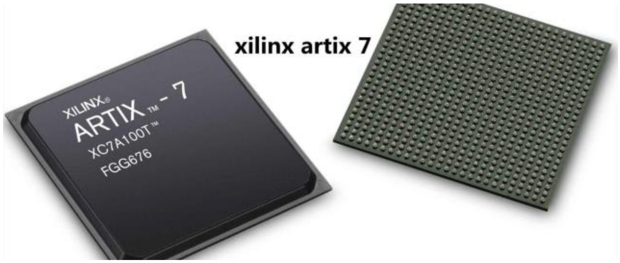
Processor

Soft-core: MicroBlaze

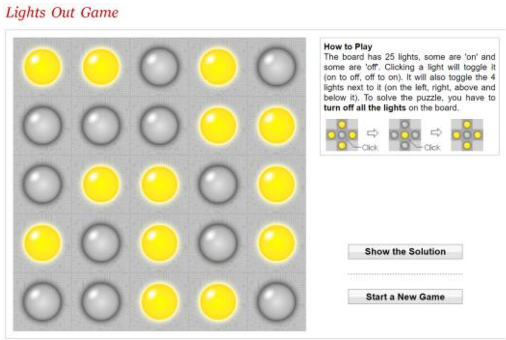
Hard-core: ZYNQ

I/O

Button, LED
Seven Segment Display

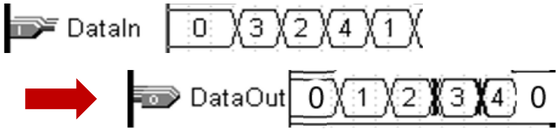


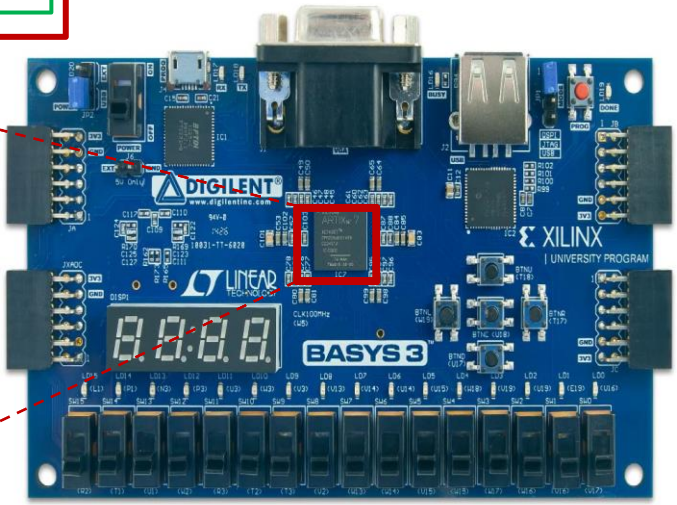
Application (1): LightsOut Puzzle on FPGA


Lights Out Game

How to Play
The board has 25 lights, some are 'on' and some are 'off'. Clicking a light will toggle it (on to off, off to on). It will also toggle the 4 lights next to it (on the left, right, above and below it). To solve the puzzle, you have to **turn off all the lights** on the board.

Application (2): Sorting on FPGA





Why ECE 448 is a challenging course?

- **need to refresh and strengthen your VHDL skills**
- **need to learn new tools**
- **need to perform practical experiments**
- **time needed to complete experiments**

Difficulties

(based on previous student survey)

- finding time to do the labs – 15

- learning VHDL – 2
- getting used to software – 1

Why is this course worth taking?

- **VHDL for synthesis:**
  one of the most sought-after skills
- **knowledge of state-of-the-art tools used in the industry**
- **knowledge of the modern FPGA technology**
- **knowledge of state-of-the-art testing equipment**
- **design portfolio that can be used during job interviews**
- **unique knowledge and practical skills that make you competitive on the job market**

ECE 448: FPGA Design with VHDL

Topics

VHDL:
- writing testbenches
- writing synthesizable code in VHDL

FPGAs:
- architecture of FPGA devices
- embedded resources (memories, DSP units)
- tools for the computer-aided design with FPGAs
- FPGA families

Applications:

- LightsOut puzzle
- basics of computer arithmetic

Platforms & Interfaces:

- FPGA boards
- I/O modules (seven segment display, etc.)

Design Techniques:

- microprocessors embedded in FPGAs (MicroBlaze)
- software/hardware co-design

Tasks of the course

Advanced course on digital system design with VHDL

- writing VHDL code for synthesis
- design using division into the datapath & controller
- testbenches
- SW/HW codesign
- I/O register map
- timing analysis

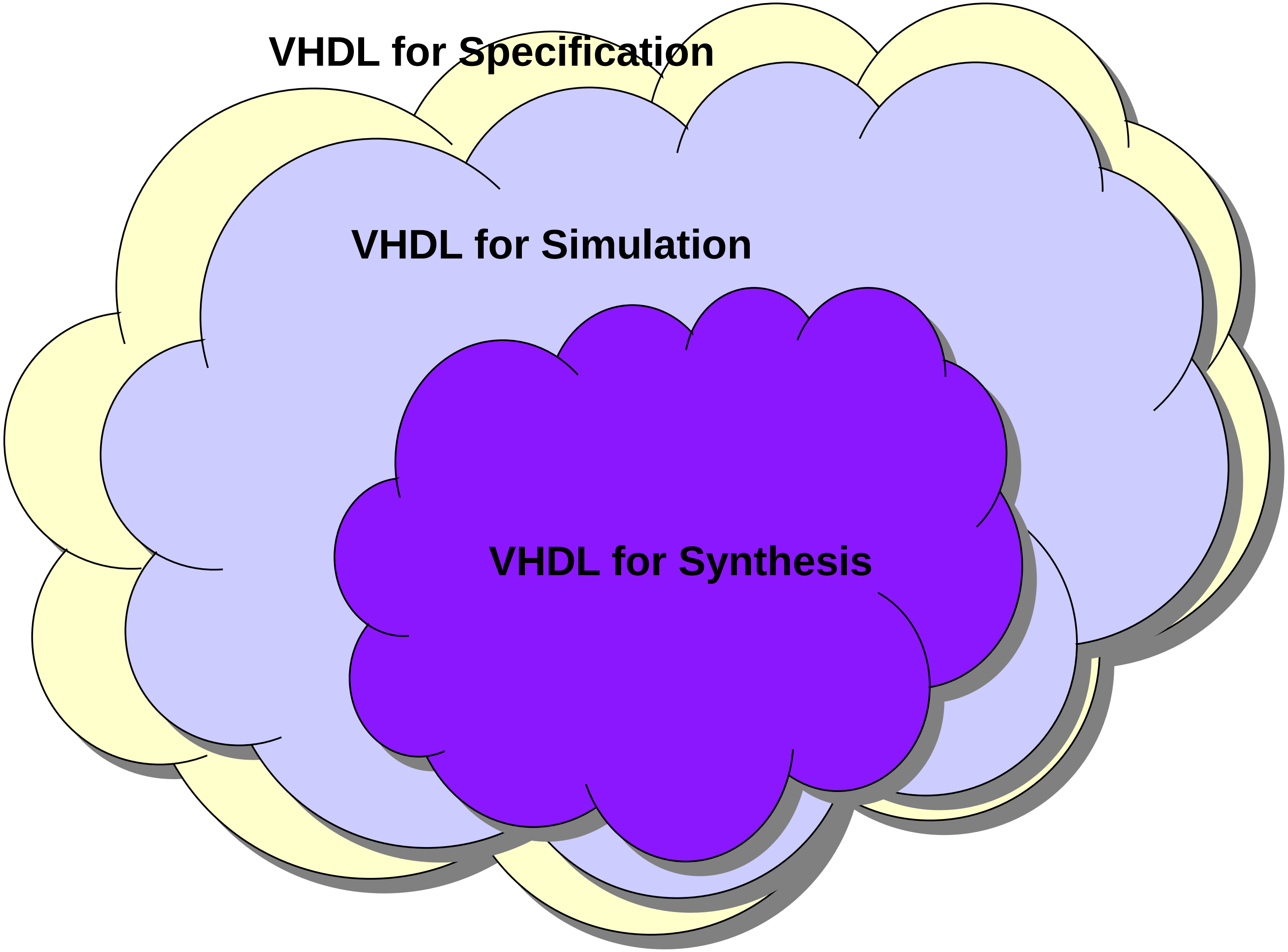
Comprehensive introduction to FPGAs

- hardware:
  - Xilinx FPGAs
  - MicroBlaze

- software:
  - Simulators
  - Synthesis tools
  - Implementation tools

Testing equipment

- oscilloscopes
- logic analyzers

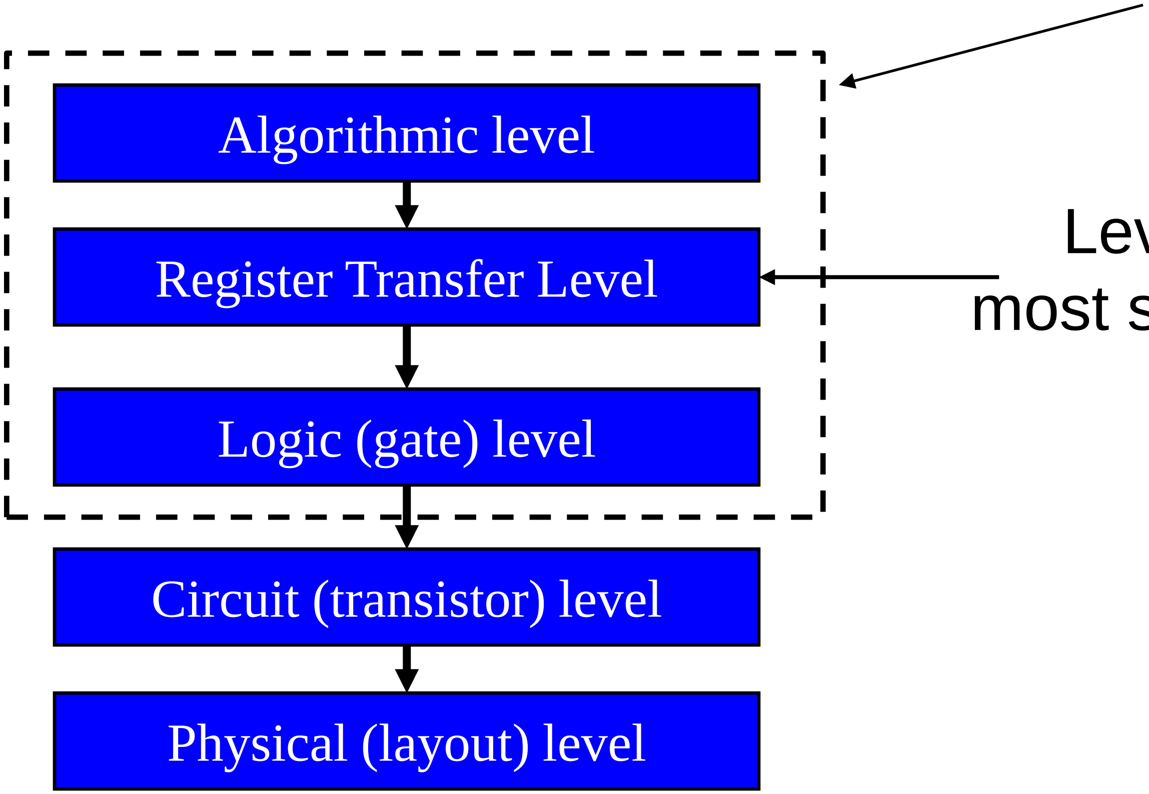


VHDL for Specification

VHDL for Simulation

VHDL for Synthesis

Levels of design description



Levels supported by HDL

Algorithmic level

Register Transfer Level

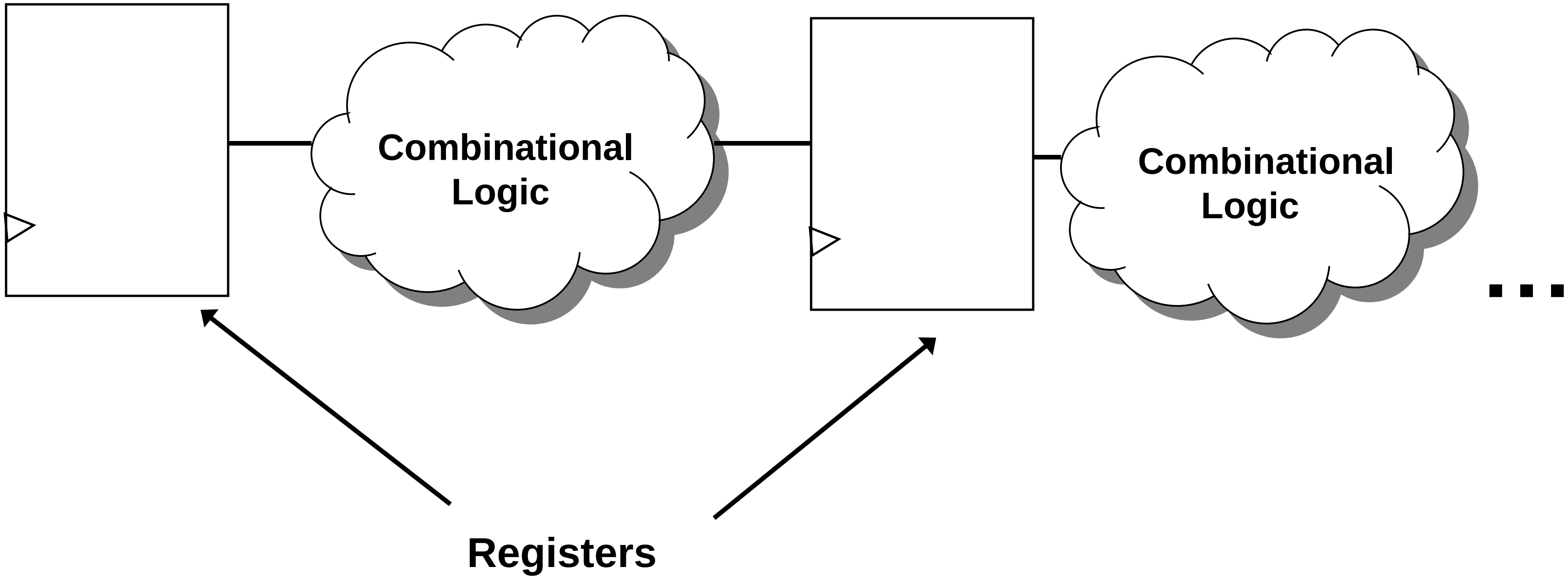
Logic (gate) level

Circuit (transistor) level

Physical (layout) level

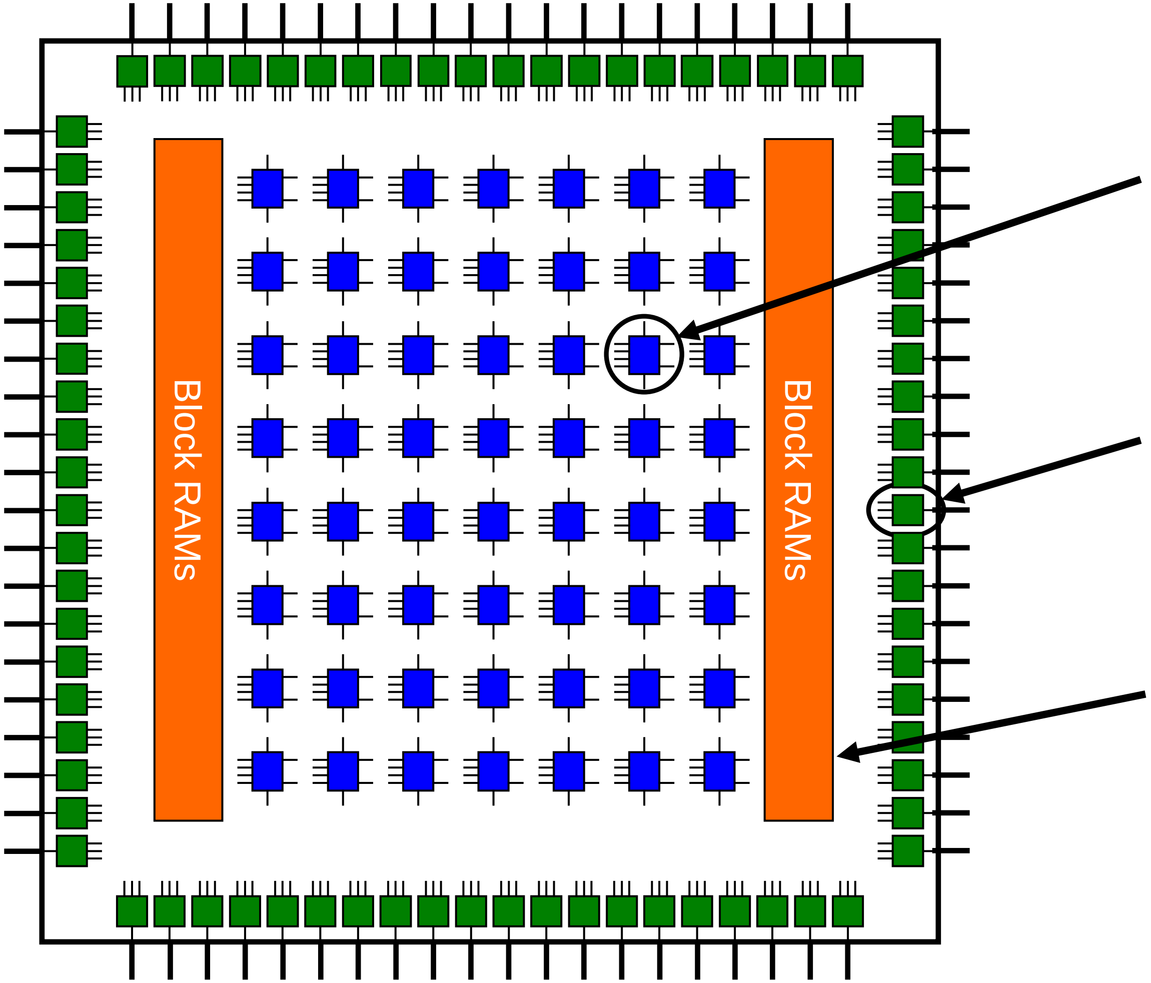
Level of description
most suitable for synthesis

Register Transfer Level (RTL) Design Description



- Use of **medium scale-components** (adders, multipliers, MUXes, ROMs, RAMs, registers, counters, etc.)
- The designer needs to **specify** what happens in the circuit in **every clock cycle**

What is an FPGA?

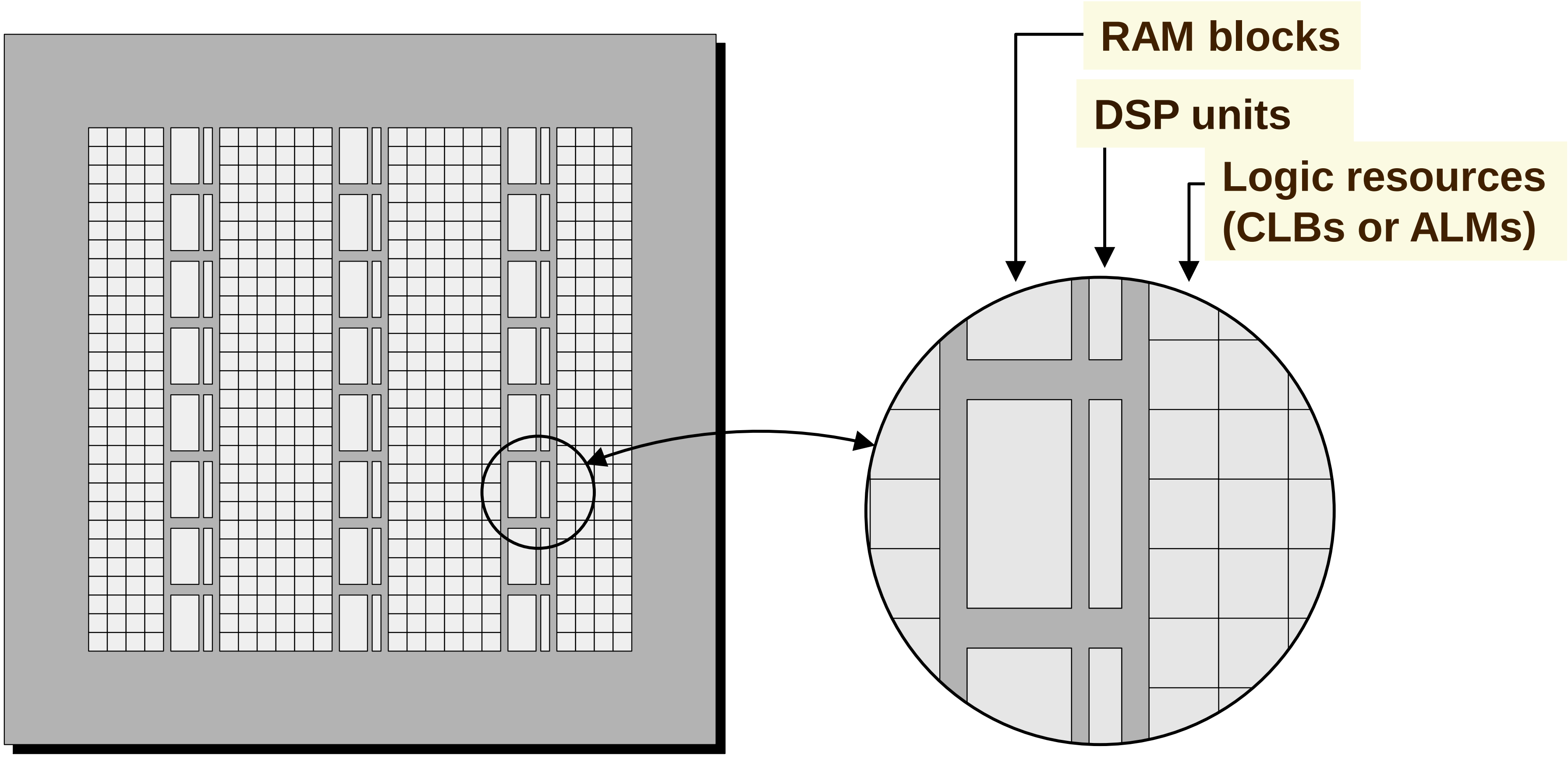


Configurable Logic Blocks

I/O Blocks

Block RAMs

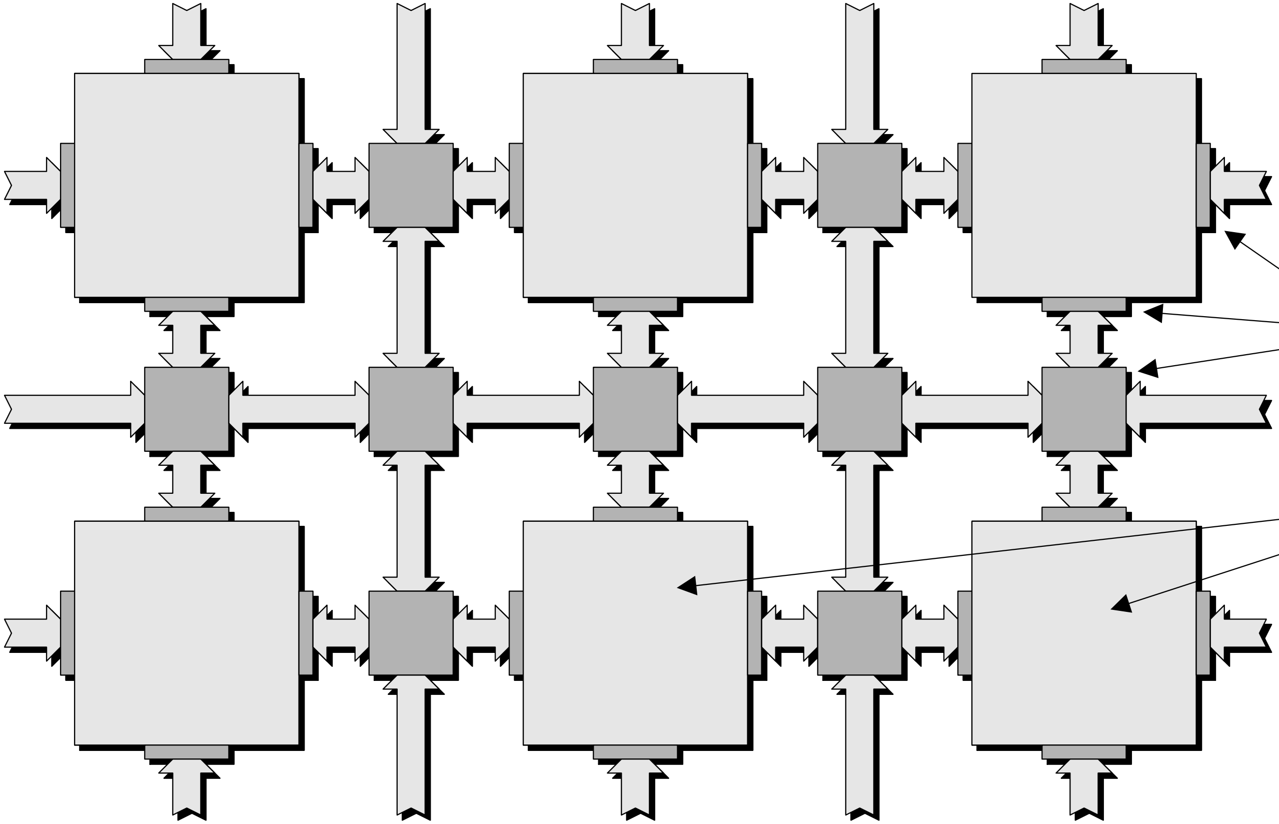
Modern FPGA



(#Logic resources, #DSP units, #RAM_blocks)

Graphics based on The Design Warrior's Guide to FPGAs
Devices, Tools, and Flows. ISBN 0750676043
Copyright © 2004 Mentor Graphics Corp. (www.mentor.com)

General structure of an FPGA



Programmable interconnect

Programmable logic blocks

The Design Warrior’s Guide to FPGAs
Devices, Tools, and Flows. ISBN 0750676043
Copyright © 2004 Mentor Graphics Corp. (www.mentor.com)

Two competing implementation approaches

ASIC
Application **S**pecific
Integrated **C**ircuit

- designed all the way from behavioral description to physical layout
- designs must be sent for expensive and time consuming fabrication in semiconductor foundry

FPGA
Field **P**rogrammable
Gate **A**rray

- no physical layout design; design ends with a bitstream used to configure a device
- bought off the shelf and reconfigured by designers themselves

FPGAs vs. ASICs

ASICs

- High performance
- Low power
- Low cost (but only in high volumes)

FPGAs

- Off-the-shelf
- Low development costs
- Short time to the market
- Reconfigurability

FPGA Design process (1)

Design and implement a simple unit permitting to speed up encryption with RC5-similar cipher with fixed key set on 8031 microcontroller. Unlike in the experiment 5, this time your unit has to be able to perform an encryption algorithm by itself, executing 32 rounds…..

Specification (Lab Assignments)

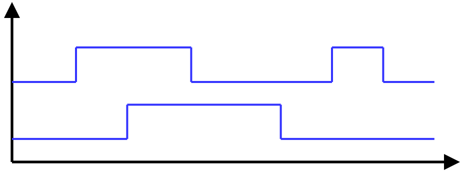
On-paper hardware design
(Block diagram & Algorithmic State Machine chart)

VHDL description (Your Source Files)

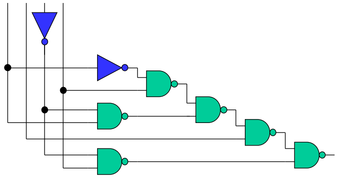
```
Library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity RC5_core is
        port(
            clock, reset, encr_decr: in std_logic;
            data_input: in std_logic_vector(31 downto 0);
            data_output: out std_logic_vector(31 downto 0);
            out_full: in std_logic;
            key_input: in std_logic_vector(31 downto 0);
            key_read: out std_logic;
        );
end AES_core;
```

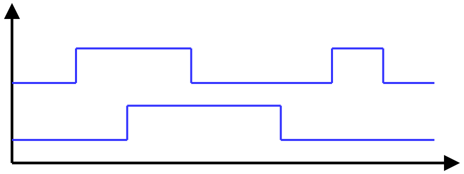
Functional simulation



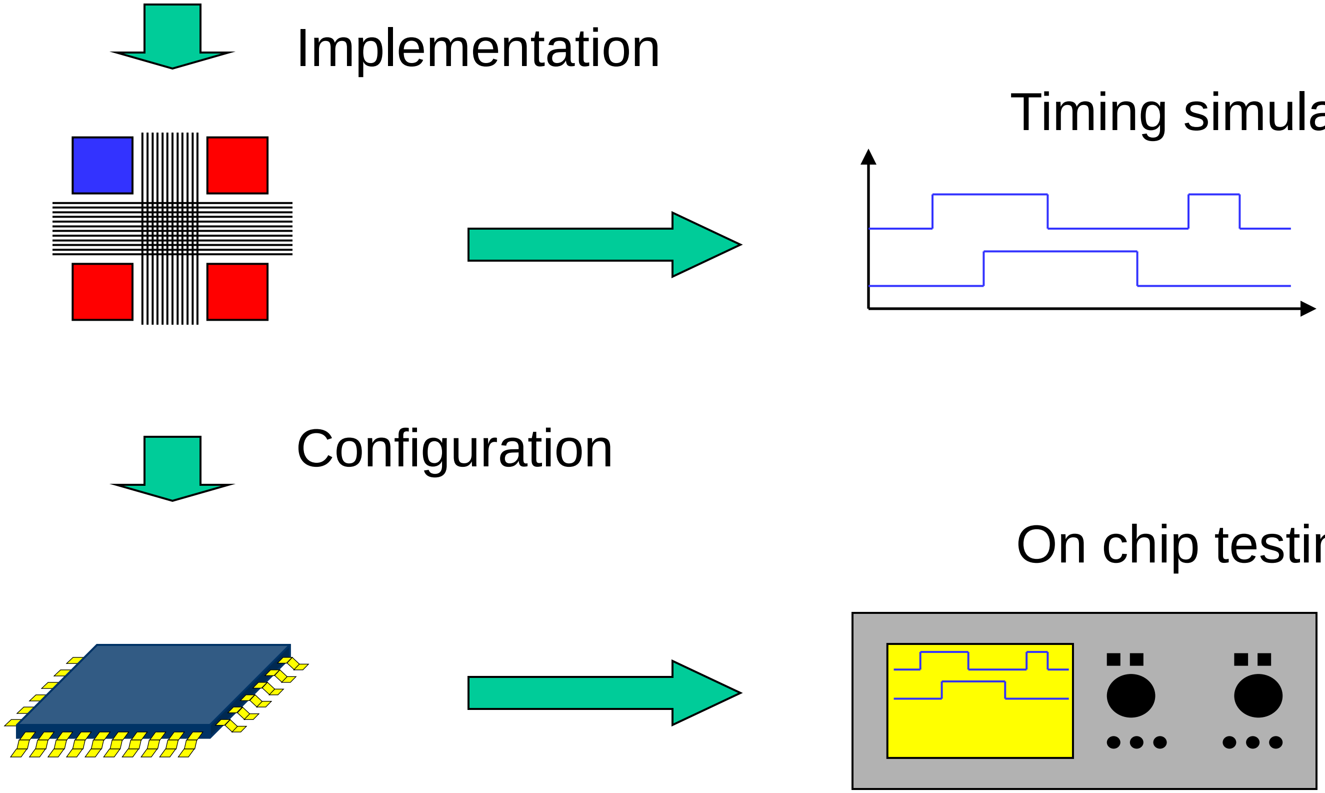
Synthesis



Post-synthesis simulation



FPGA Design process (2)



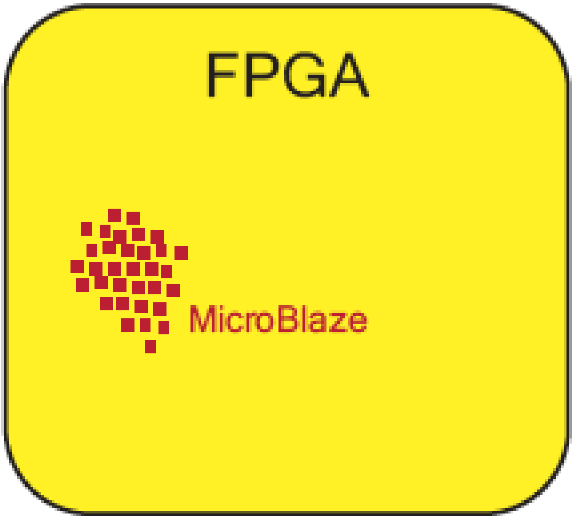
Implementation

Timing simulation

Configuration

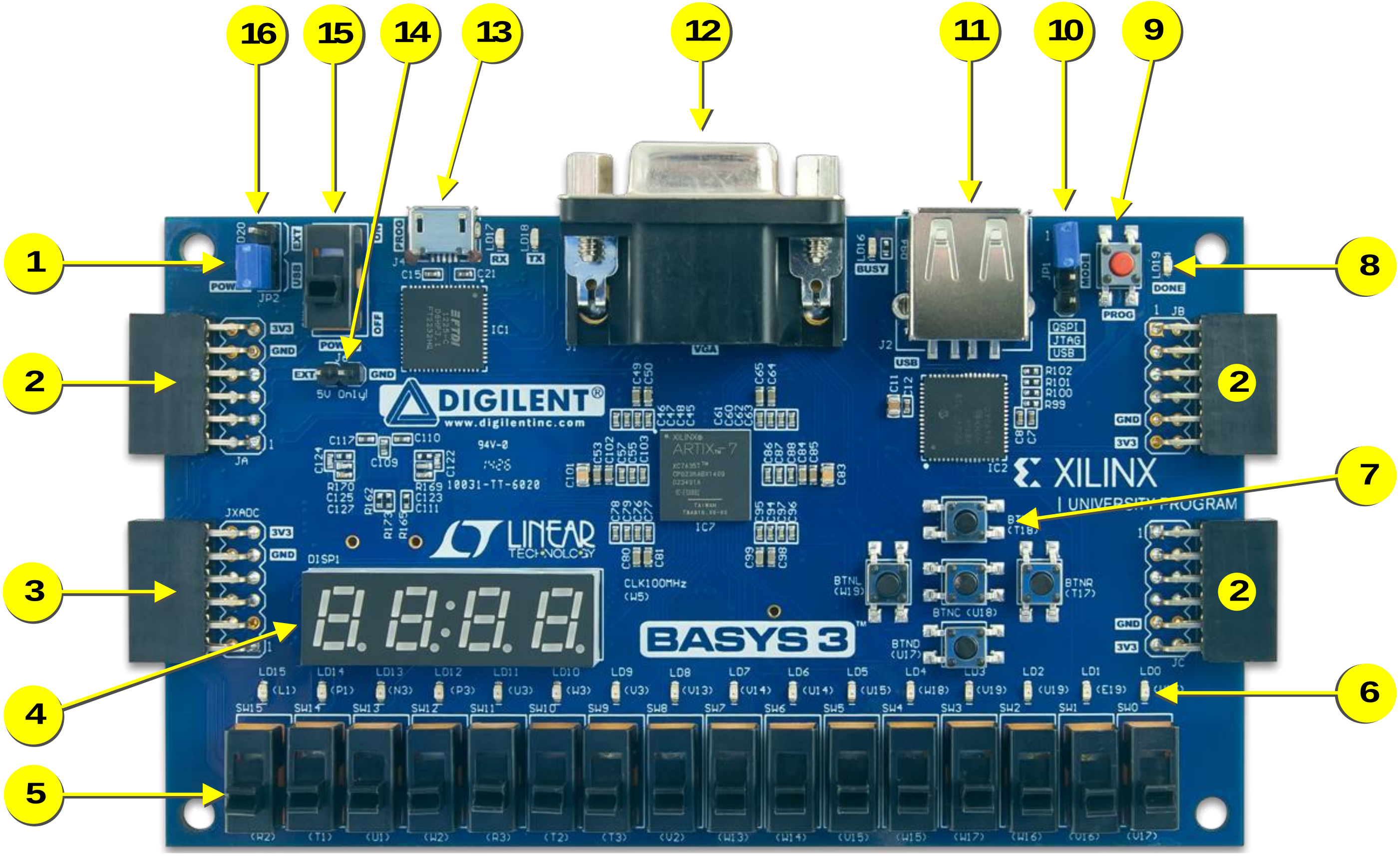
On chip testing

Software/Hardware Codesign
FPGA with a Soft-Core Processor



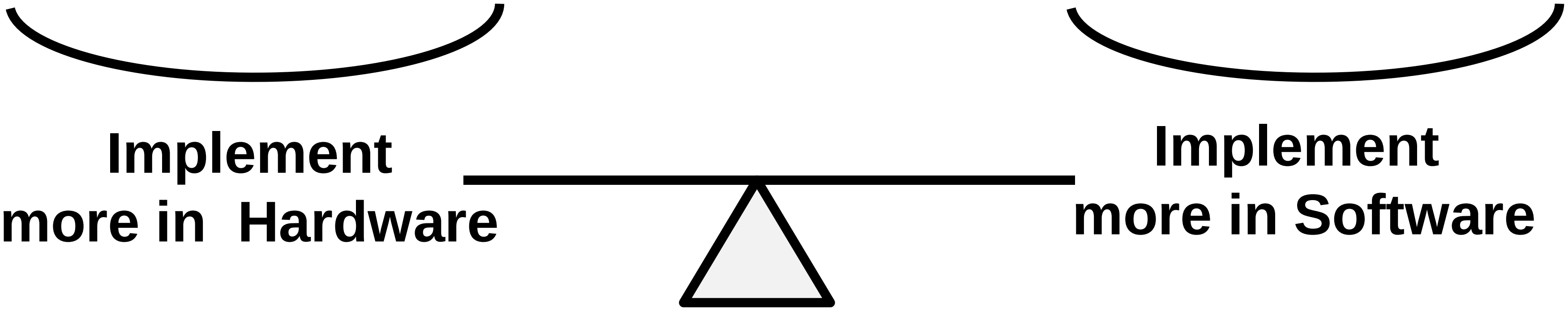
Source: The Zynq Book

FPGA Boards



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

Software vs. Hardware Trade-offs



Improve Performance
Improve Energy Efficiency
Reduce Power Density

Manage Design Complexity
Reduce Design Cost
Stick to Design Schedule

Implement more in Hardware

Implement more in Software

Source: A Practical Introduction to Hardware/Software Codesign

wjiang8@gmu.edu

George Mason University

4400 University Drive
Fairfax, Virginia 22030
Tel: (703)993-1000