Intro and Analog Behavior of Digital Systems

Lecture 01

Josh Brake
Harvey Mudd College

Outline

- Course Overview
  - Course learning objectives
  - Administrative details
  - Syllabus
  - Schedule
  - Important links and information
  - Labs and Checkoffs
- Final Project
- Embedded Systems Introduction
- Warp Speed Boolean Logic Review

Introduction

Caltech

Things other than E155 stuff you could get me talking about in office hours…

- Pizza making
- Coffee drinking & roasting
- Spending time with my family
- Running/Biking/Hiking
- Following sports: NY Mets, Tennessee Titans
- Reading

Learning Objectives: Technical Skills

- Design and implement combinational and sequential circuits on an FPGA.
- Use an ARM-based microcontroller to interface with the real world via sensors and actuators.
- Build an embedded system project of your own design from the ground up.
- Select appropriate embedded hardware for a given design challenge.
- Effectively and efficiently debug electrical systems with measurement tools such as an oscilloscope and logic analyzer.
- Read and understand complicated datasheets at a level that enables you to incorporate them into your designs.

Learning Objectives: Professional Skills

- Communicate technical results with clarity and in a professional manner through oral presentations and written reports.
- Communicate your work with a wide variety of audiences, including technical experts, your peers, and the general public.
- Articulate how you have developed character strengths throughout the semester.

Character Traits

Intellectual Virtues	Moral Virtues	Civic Virtues	Performance Virtues
Autonomy	Compassion (Empathy)	Citizenship	Confidence
Critical Thinking	Courage	Civility	Determination
Curiosity	Gratitude	Neighborliness	Motivation
Judgment	Honesty	Service	Perseverance
Reasoning	Humility	Volunteering	Resilience
Reflection	Integrity	Community Awareness	Teamwork (Collaboration)
Resourcefulness	Respect		
	Justice (Equity, Equality)		

Source: The Jubilee Centre Framework for Character Education in Schools

Class Norms & Expectations

How do we want to engage each other in these areas?

- Communication
- Collaboration
- Honor Code
- Mutual Respect
- Engagement and Attendance
- Feedback and Improvement

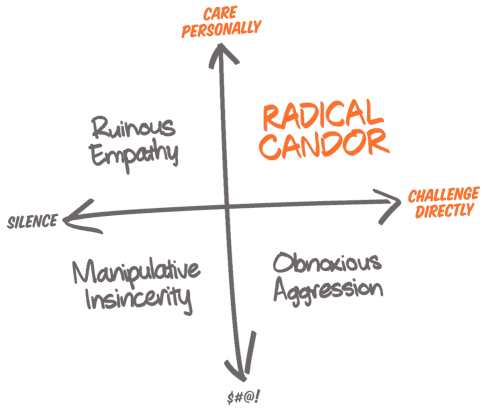
Radical Candor & Psychological Safety

Radical Candor

What is Radical Candor?

Defined by two attributes:

1. Care personally
2. Challenge directly



Psychological Safety

Create an environment where you can courageously challenge yourself.

Failure is ok, it means you’re learning!

But there are better and worse ways to fail.

Schedule

Monday Date	Tuesday's Class	Thursday's Class	Due
8/26	Intro & Analog Behavior of Digital Systems	Combinational and Sequential Logic	Git & Quarto Portfolio Setup
9/2	Verilog Coding	Synchronous Design	Lab 1
9/9	FPGA Documentation	C Programming on an MCU	Lab 2
9/16	Clock Configuration	Timers	Lab 3
9/23	Interrupts	Serial Interfaces Overview & SPI	Lab 4
9/30	TBD	TBD	
10/7	Project Kickoff	UART and the IoT	Lab 5
10/14	Happy Fall Break! **No Class**	Proposal Debriefs	Project Proposal
10/21	Advanced Encryption Standard (AES)	Graphics and Displays	Lab 6
10/28	Motors and Speakers	Direct Memory Access	Lab 7
11/4	Design Review Presentations	Design Review Presentations	Design Review Presentation & Memo
11/11	Introduction to Real Time Operating Systems	Project Status Report and Demo	Project Status Reports & Demo
11/18	The Fast Fourier Transform (FFT)	Connectors	
11/25	Emerging Topics in Embedded Systems	Happy Thanksgiving! **No class**	
12/2	Guest Lecture: Ben Chelf	TBD	Project Checkoffs, Report, Demo Day

Important Information

- Course website: https://hmc-e155.github.io/
- Discord Server (invite link in welcome email)

Office hours will be scheduled later this week based on clinic meeting schedule.

My Teaching Philosophy

- **Transparent Teaching** – Not making why you're doing something a mystery
- **Psychological Safety** – Create an environment where you can courageously challenge yourself
- **Prototyping Mindset** – Try to push yourself. Progressive overload.
- **Frequent, Low-stakes Testing** – Try, fail, try again
- **Interleaving (spaced repetition)** – Come back to things again and again, with a little time in between
- **Radical Candor** - care personally, challenge directly
- **Productive Struggle** - learning is hard work and requires effort

Course Content

Three main sections: labs, project, and participation.

Labs

Mini-design projects that you complete on your own time outside of the scheduled class time. You check your lab off with an instructor each week during a 10-minute meeting each week during lab on Tuesday or Thursday afternoon.

Project

- Final project of your own choosing completed in teams of 2.
- Two criteria: must do something fun or useful and meaningfully use the MCU and the FPGA.
- Start thinking about potential projects early.
- See the course website for inspiration from past years.

Reflection & In-class participation

- We will regularly open class with short low-stakes quizzes using Plickers.
- Weekly blog reflections.
- You get full credit for participating (i.e., graded on completion).

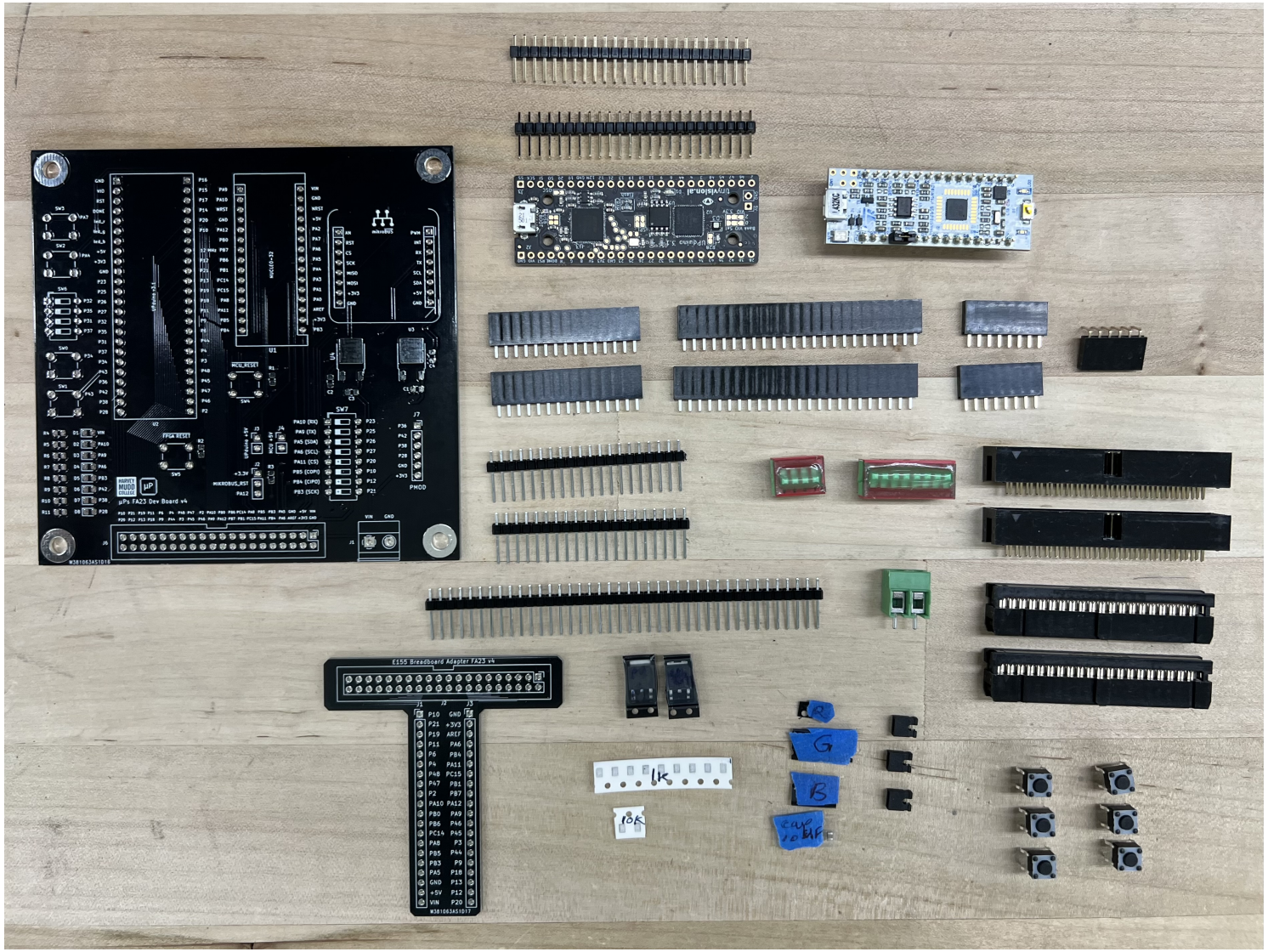
Labs

- Lab 1 - Board Assembly and Testing
- Lab 2 - Multiplexed 7-Segment Display
- Lab 3 - Keypad Scanner
- Lab 4 - Digital Audio
- Lab 5 - Interrupts
- Lab 6 - The Internet of Things and Serial Peripheral Interface (SPI)
- Lab 7 - The Advanced Encryption Standard (AES)

Lab Checkoffs

Each week you will demonstrate your lab to an instructor and have it checked off based on that lab's specifications.

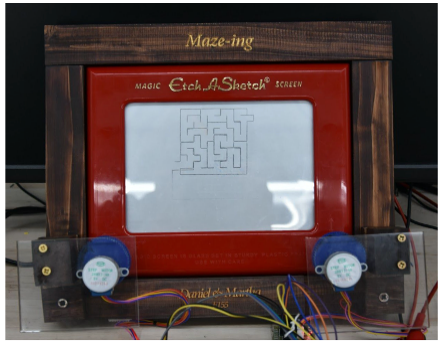
Lab Kit

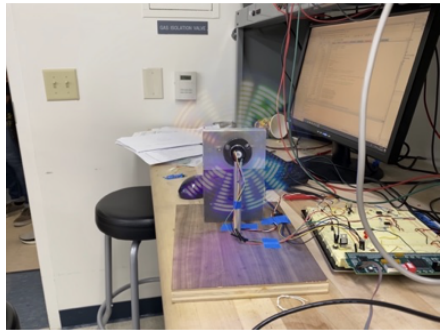


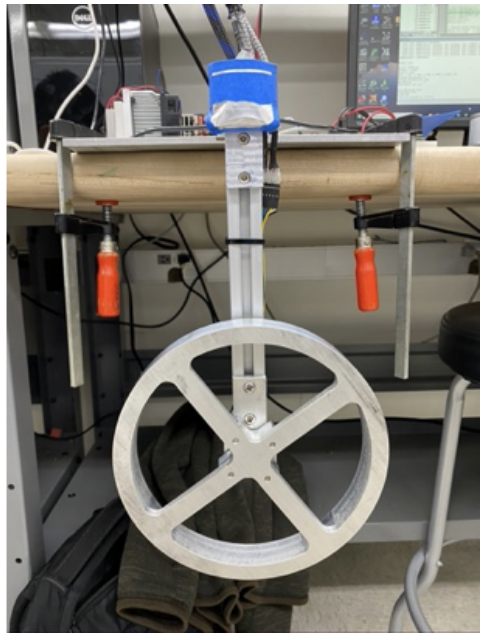
Project

There are 5 deliverables:

1. Proposal
2. Design Review Presentation
3. Mid-point checkoff
4. Final demo
5. Final portfolio website







What are grades for?

Grading

The grading in this course may be a bit different than what you have seen in other courses. Your grade will be determined based on the quality of the deliverables you submit.

1. **Not Yet**: no attempt or poor attempt made. Design does not meet minimum specifications.
2. **Proficiency**: design meets proficiency specifications but not excellence specifications.
3. **Excellence**: design meets both proficiency and excellence specifications.

Grading

Most assignments in this class will be presented with a list of specifications with two levels:

1. Proficiency
2. Excellence

Your grade will be assigned based on whether you meet only the proficiency specs or both the proficiency and excellence specs.

How grades are assigned

There are three main bundles where you can accrue points. For the grade category you must meet all criteria in the respective row.

- P = proficiency specs
- E = excellence specs

Grade	Labs	Project
F	$P < 5$	$P < 4$
D	$P \geq 5, E \geq 1$	$P \geq 4$
C	$P \geq 6, E \geq 2$	$P \geq 4, E \geq 2$
B	$P = 7, E \geq 5$	$P = 5, E \geq 3$
A	$P = 7, E = 7$	$P = 5, E = 5$

MicroPs Portfolio

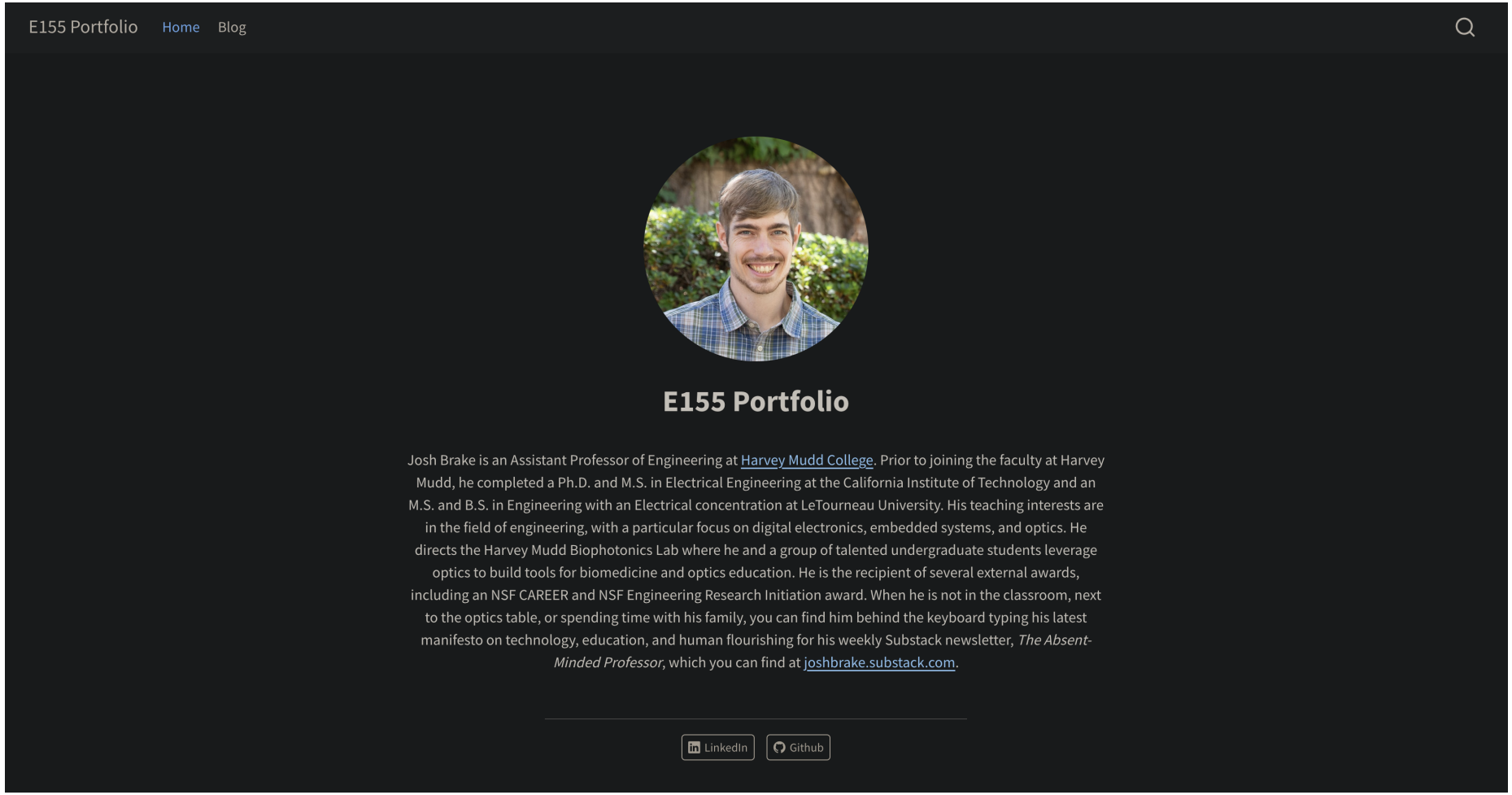
Who is the audience for your work in this class?

- Me
- Your peers
- Your future employers/companies

Success in MicroPs for me looks like helping you demonstrate a grasp of the material in this class to the level that you could get a job or start a company on the foundation of your skills. This means that your future employers (or investors) are an explicit audience for your work in this class.

Throughout the course you will be sharing your work in public. Note: You are *strongly* encouraged to publish your work publicly, but I will not require it.

MicroPs Portfolio Demo and Tech Stack



Quarto + Git

What about generative AI?

- What are the impacts of generative AI on the work we do in MicroPs?
- What should our norms be around generative AI use?

You **should**:

- Use AI on the AI prototypes throughout the course.

You **should not**:

- Use generative AI to directly generate any code on the main lab design challenges.
- Use generative AI to generate any of your technical documentation

Practices for Success

- Start early!
- Get stuck, but don’t stay stuck.
- Read documentation carefully and repeatedly.
- Ask for help (instructors, tutors, classmates)
- Block your time
- Practice good debugging strategies
  - Follow the signal flow (GIGO)
  - Try a simple case first
  - Use measurement tools to your advantage
  - Use software testing tools like test benches and unit tests.

Embedded Systems Overview

What is an embedded system?

Embedded systems are information processing systems embedded into a larger product

Table 1.2 Distinction between PC-like and embedded system design

	Embedded	PC-/Server-like
Architectures	Frequently heterogeneous very compact	Mostly homogeneous not compact (×86 etc.)
×86 compatibility	Less relevant	Very relevant
Architecture fixed?	Sometimes not	Yes
Model of computation (MoCs)	C+multiple models (data flow, discrete events, …)	Mostly von Neumann (C, C++, Java)
Optimization objectives	Multiple (energy, size, …)	Average performance dominates
Safety-critical?	Possibly	Usually not
Real-time relevant	Frequently	Hardly
Applications	Guarantees for several concurrent apps. needed	Best effort approaches to run application
Apps. known at design time	Yes, for real-time systems	Only some (e.g., WORD)

Marwedel, Peter. *Embedded System Design*. Springer, 2018.

What are some common fields where embedded systems are used?

- Transportation
- Factory automation
- Health
- Smart buildings
- Smart grid
- Scientific experiments
- Public safety
- Structural health monitoring
- Disaster recovery
- Robotics
- Agriculture
- Military
- Telecommunication
- Consumer electronics

Marwedel, Peter. *Embedded System Design*. Springer, 2018.

Challenges

Dependability

- Safety
- Security
- Confidentiality
- Reliability
- Repairability
- Availability

Resource Awareness

- Energy
- Run-time
- Code size
- Weight
- Cost

Design Flow

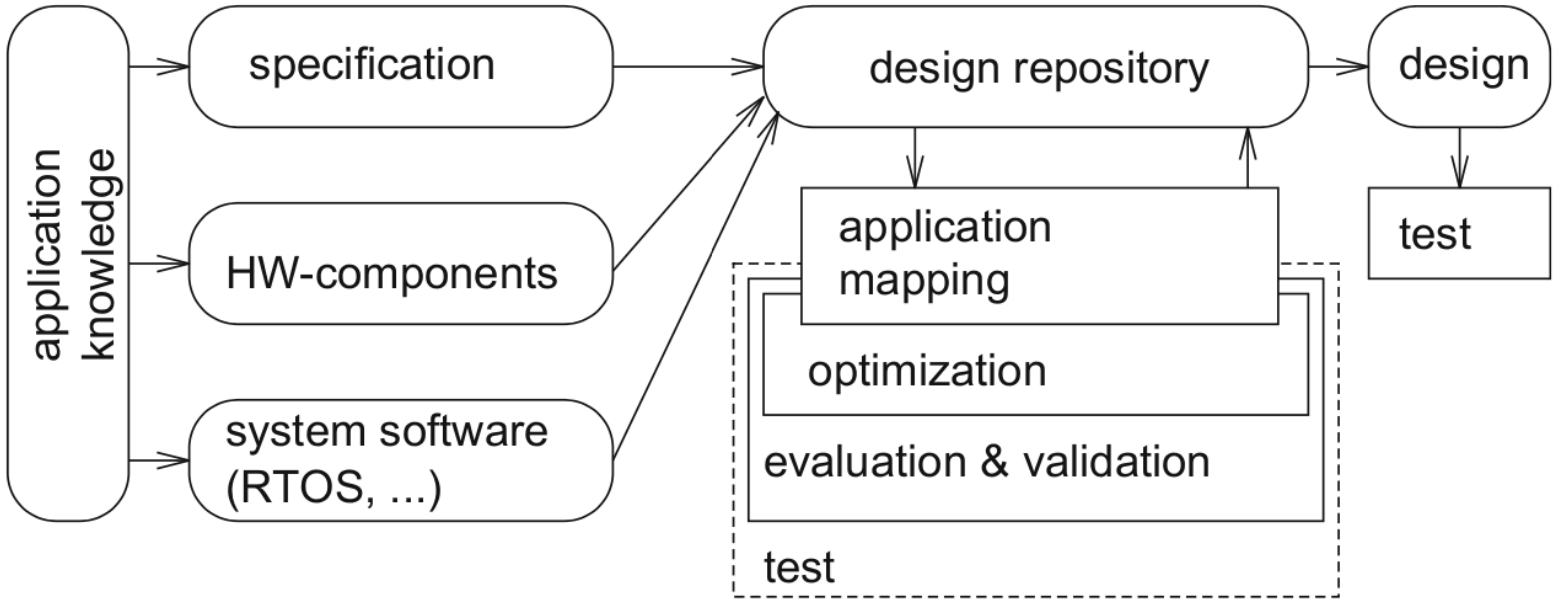


Fig. 1.8 Simplified design information flow

Marwedel, Peter. *Embedded System Design*. Springer, 2018.

Design Flow Unfolded

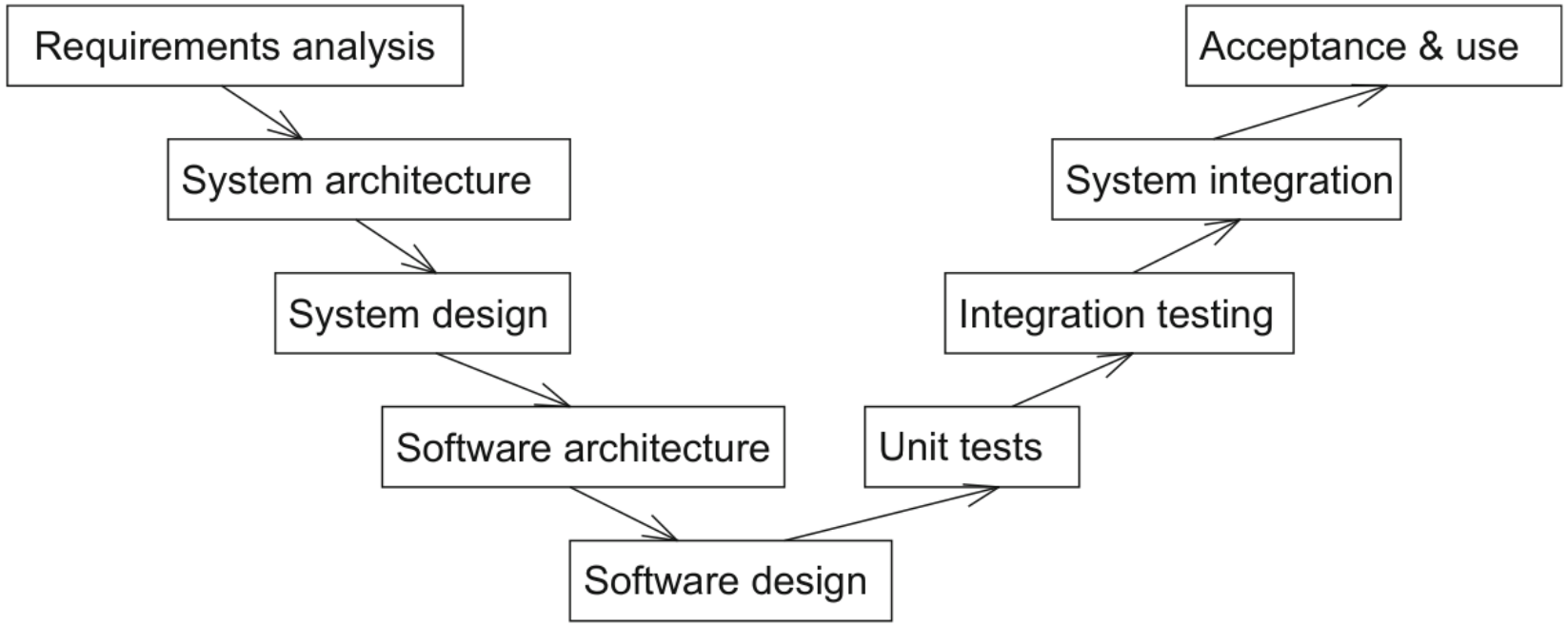
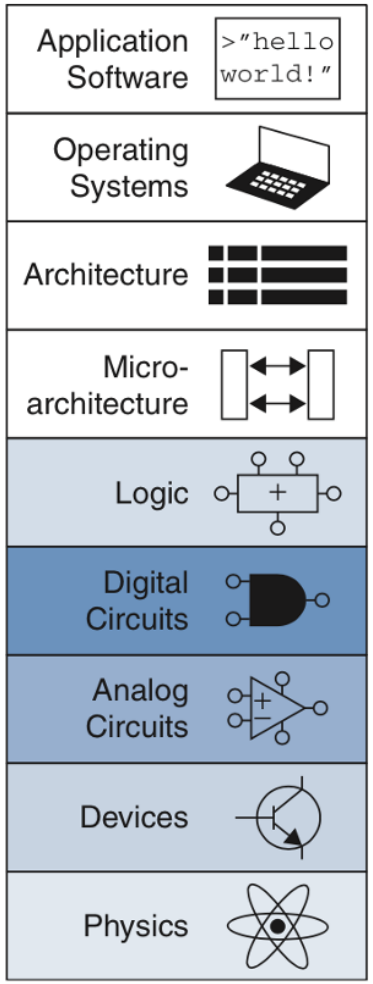


Fig. 1.10 Design flow for the V-model

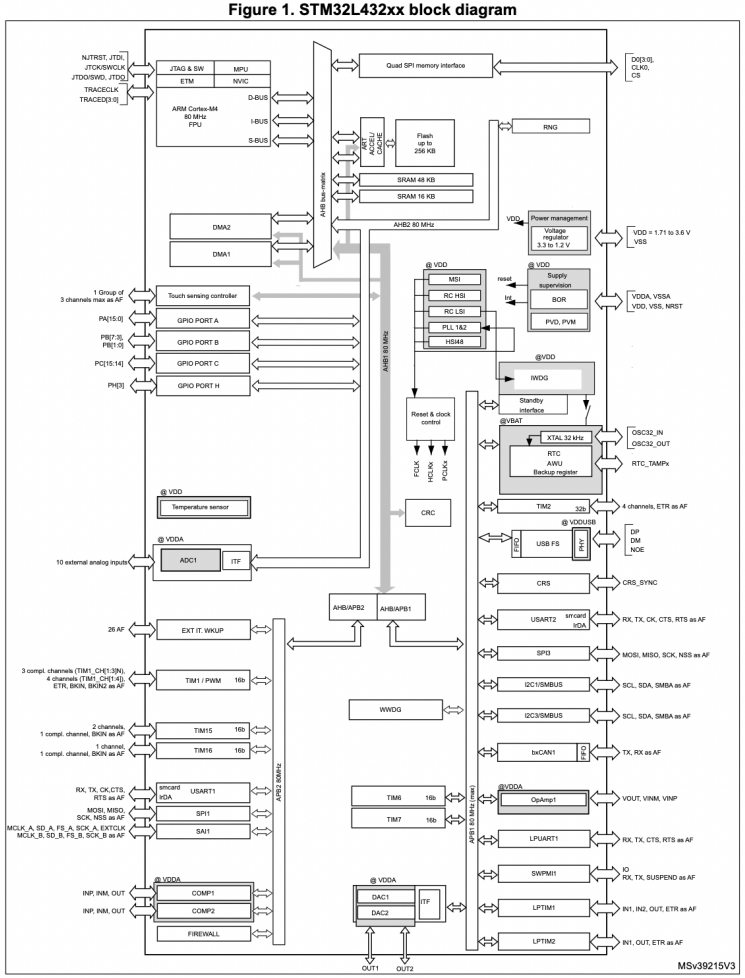
Marwedel, Peter. *Embedded System Design*. Springer, 2018.

MicroPs in Context



From DDCA ARM edition

Figure 1. STM32L432xx block diagram



STM32L432KC Datasheet - p. 13

What is an FPGA?

- FPGA stands for **F**ield **P**rogrammable **G**ate **A**rray.
- Programmed by configuring **Logic Cells** (**LC**) and connecting them.

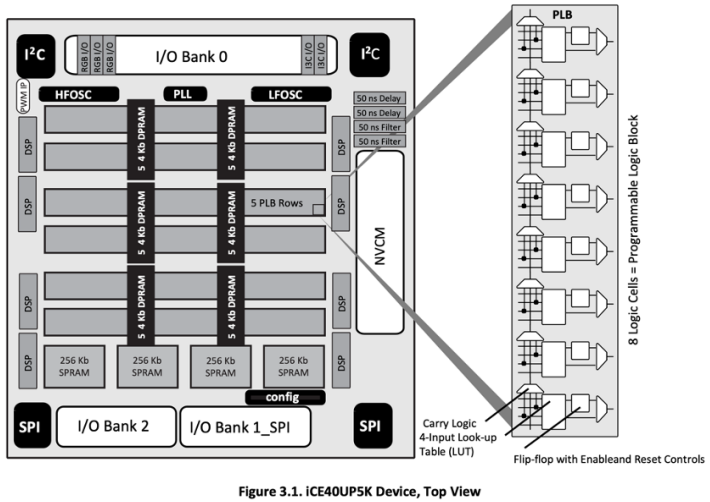


Figure 3.1. iCE40UP5K Device, Top View

iCE40 UltraPlus Family Data Sheet p. 11

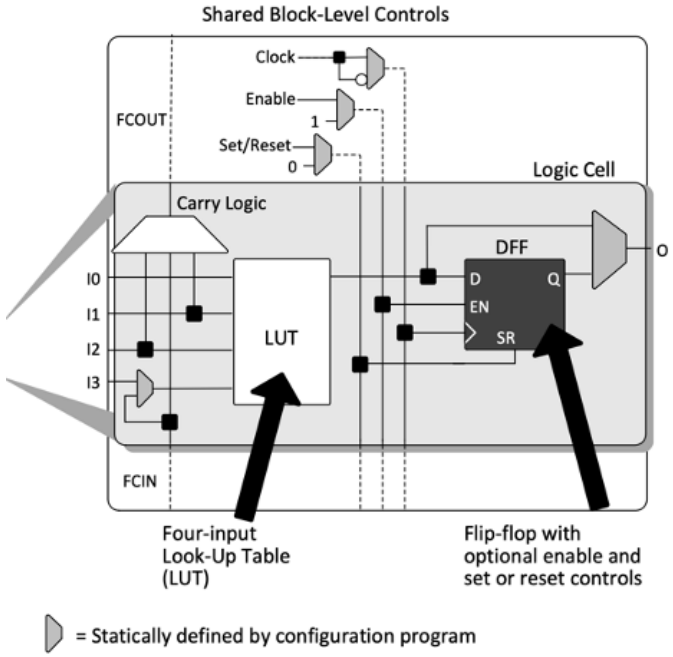


Figure 3.2. PLB Block Diagram

iCE40 UltraPlus Family Data Sheet p. 12

UPduino
v3.1

What is an MCU?

- MCU stands for **Micro**controller **U**nit
- Comprised of a processor (think E85) surrounded by a bunch of peripherals.
  - Serial communication: UART, SPI
  - Timers & Pulse Width Modulation (PWM)
  - General-purpose Input-Output (GPIO) Controllers

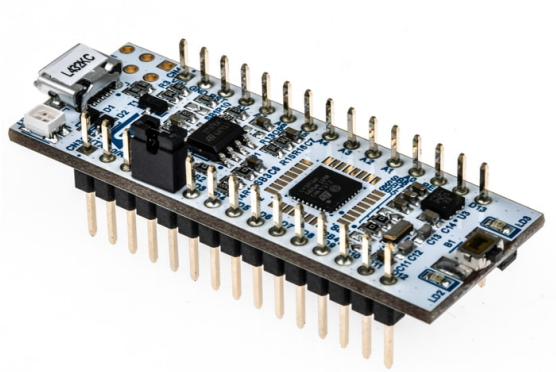
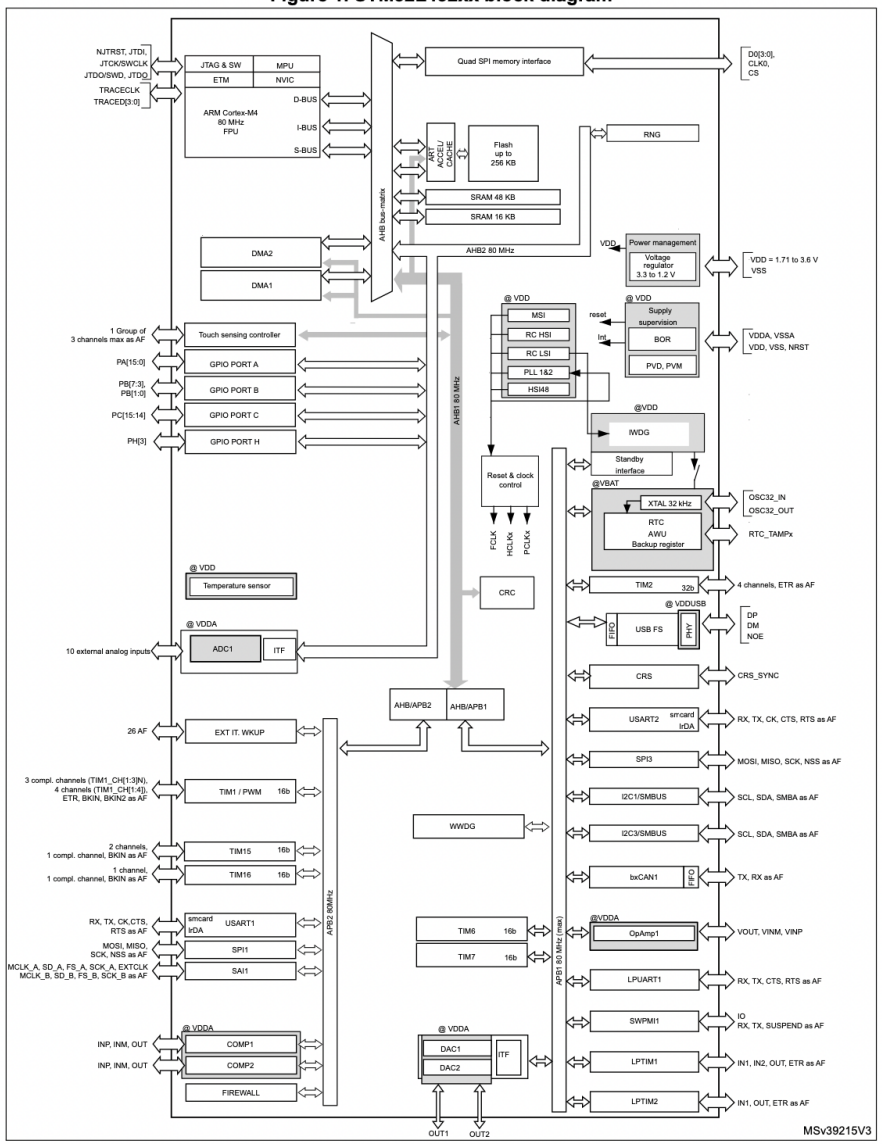


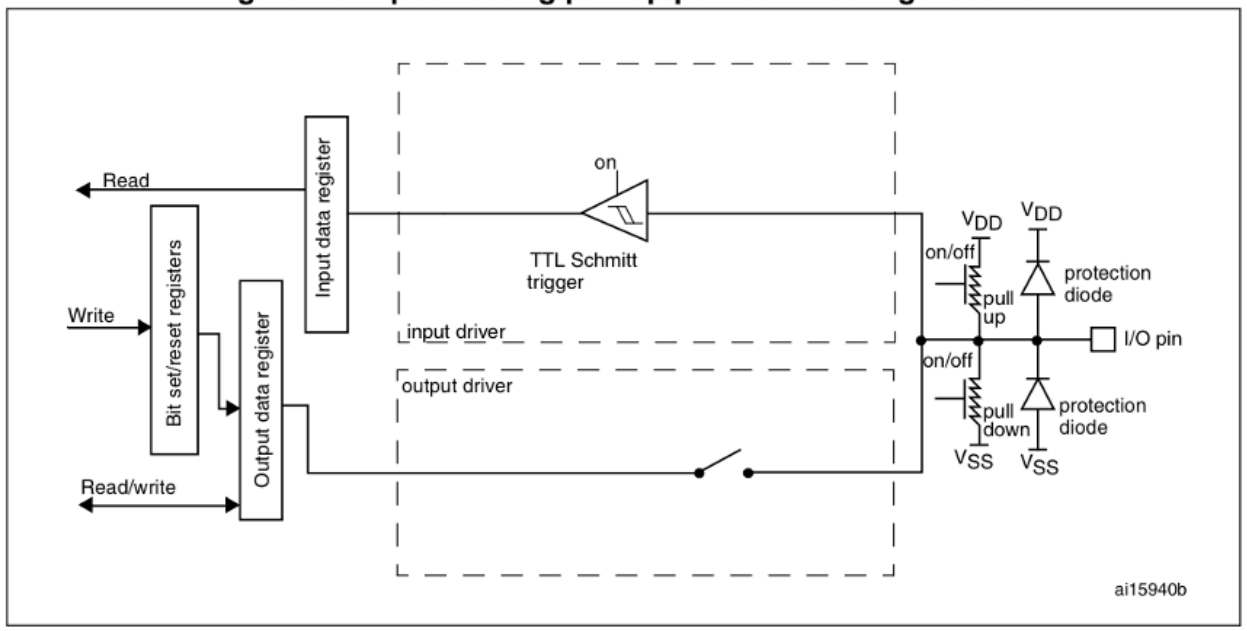
Figure 1. STM32L432xx block diagram



MicroPs in Context

- Focus on learning general concepts – specifics are obsolete tomorrow!
- Some core concepts endure
  - Understanding device layout and how to read documentation
  - Basic configuration/programming flow
  - Memory-mapped I/O
  - Peripheral configuration
  - Clock configuration
  - Peripherals
  - UART
  - SPI
  - Timers
  - PWM

Figure 18. Input floating/pull up/pull down configurations

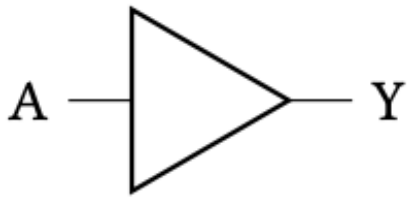


Technical Kickoff: E85 Warp Speed Review

Boolean Logic & Verilog Idiom Review

- BUF
- NOT
- AND
- OR
- NAND
- NOR
- XOR
- XNOR
- AND3
- 2:1 Mux
- 2:4 Decoder

BUF



Truth Table

A	Y
0	0
1	1

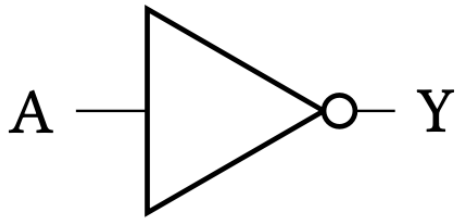
Boolean Expression

$$Y = A$$

Verilog Idiom

```
assign y = a;
```

NOT



Truth Table

A	Y
0	1
1	0

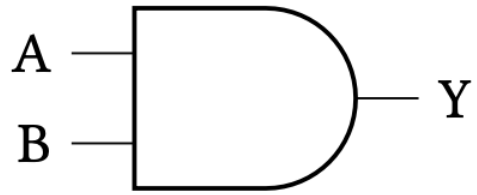
Boolean Expression

$$Y = \bar{A}$$

Verilog Idiom

```
assign y = ~a;
```

AND



Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

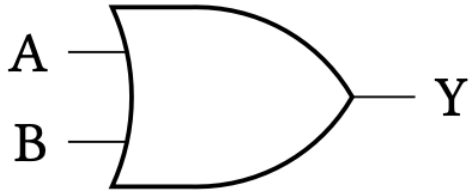
Boolean Expression

$$Y = AB$$

Verilog Idiom

```
assign y = a & b;
```

OR



Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

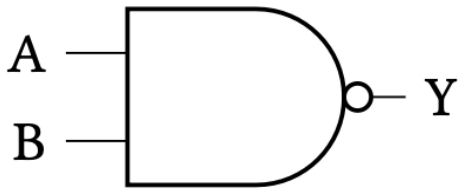
Boolean Expression

$$Y = A + B$$

Verilog Idiom

```
assign y = a | b;
```

NAND



Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

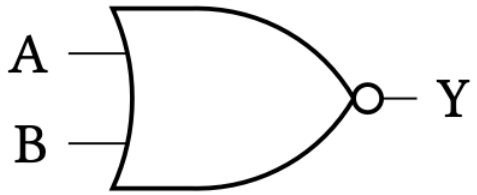
Boolean Expression

$$Y = \overline{AB}$$

Verilog Idiom

```
assign y = ~(a & b);
```

NOR



Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

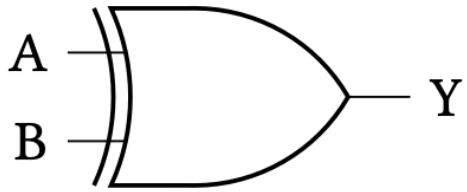
Boolean Expression

$$Y = \overline{A + B}$$

Verilog Idiom

```
assign y = ~(a | b);
```

XOR



Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

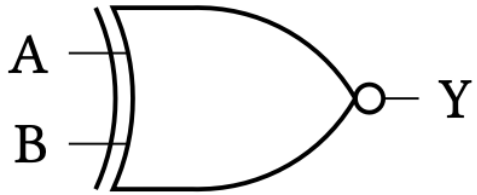
Boolean Expression

$$Y = A \oplus B$$

Verilog Idiom

```
assign y = a ^ b;
```

XNOR



Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

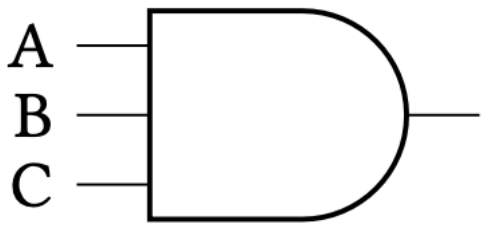
Boolean Expression

$$Y = \overline{A \oplus B}$$

Verilog Idiom

```
assign y = ~(a ^ b);
```

AND3



Truth Table

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

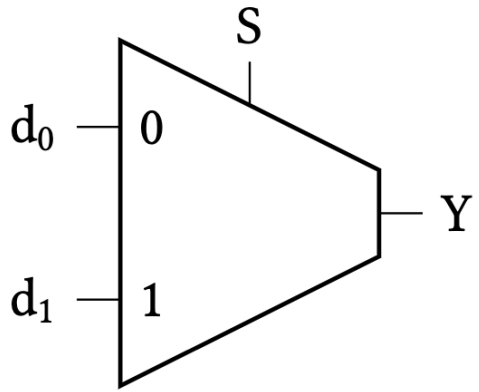
Boolean Expression

$$Y = ABC$$

Verilog Idiom

```
assign y = a & b & c;
```

2:1 Multiplexer (MUX)



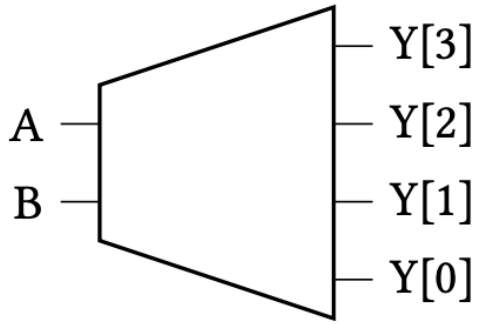
Truth Table

S	d1	d0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Verilog Idiom

```
assign y = s ? d1 : d0;
```

2:4 Decoder



Truth Table

A	B	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Verilog Idiom

```
1  always_comb
2    case(s)
3      2'b00:    y = 4'b0001;
4      2'b01:    y = 4'b0010;
5      2'b10:    y = 4'b0100;
6      2'b11:    y = 4'b1000;
7      default:  y = 4'bxxxx;
```

Analog Behavior of Digital Systems

DC Logic Gate Transfer Characteristics

- VIH - Lowest input voltage recognized as logical 1
- VIL - Highest input voltage recognized as logical 0
- VOH - Lowest output voltage indicating logical 1
- VOL - Highest output voltage indicating logical 0
- Noise margins - Difference between the high and low values for the input or output levels.

DC Logic Gate Transfer Characteristics

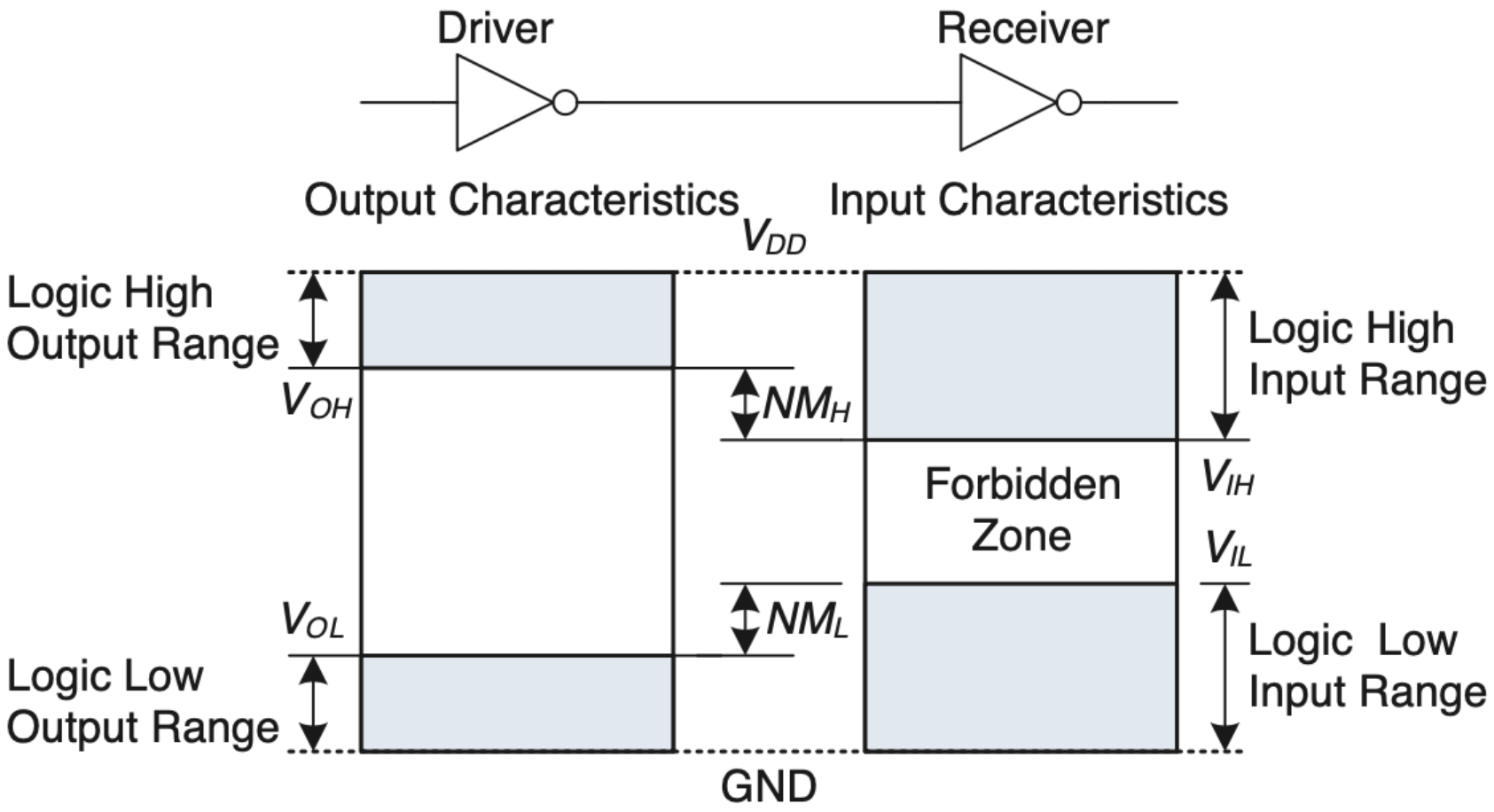


Figure 1.23 from *Digital Design and Computer Architecture: RISC-V Edition* by Harris & Harris

Important Specs

- Iin – Input leakage current (a current flowing in or out of the pin toward a power rail. This can cause invalid logic levels if you're not careful and use too big of a pulldown/up resistor.)
- Iout – Maximum output driving current
- Cin – Input pin capacitance
- IDD – Supply current. DD stands for from drain to drain.

Wrap Up

- Get kit and start on Lab 1 ASAP
- Also, checkout breadboard from the stockroom to use for the semester.
- Schedule checkoff time (try to stay within your scheduled section. Help each other out if others need to swap).
- Lab demos this week during the first hour of the lab slot
  - Git and Quarto setup help
  - Soldering refresh
  - Basic FPGA and MCU programming (see tutorials on the website)