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

Name: Prof. Josh Brake

Caltech

LETU LeTourneau University

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

Brake Family in San Diego this summer. Prof. Brake with his wife Abbey, Judah (11), Chloe (10), #3 due 10/23/23.

Learning Objectives

Embedded systems technical expertise

- 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 task.
- Read and understand complicated datasheets at a level that enables you to incorporate them into your designs.

Debugging and systems engineering

- Effectively and efficiently debug electrical systems with measurement tools such as an oscilloscope and logic analyzer.

Communication

- Clearly communicate technical results in a professional manner both through oral presentations and written reports.

Class Norms & Expectations

Instructor

- Be respectful of all students in the class
- Prepare engaging and challenging course material
- Timely feedback to student work and questions

Students

- Be respectful of instructors, grutors, and peers
- Engaged attention in class
- Radical candor

The classroom as an ecosystem

What you do has an impact on those around you, even if you might not realize it.

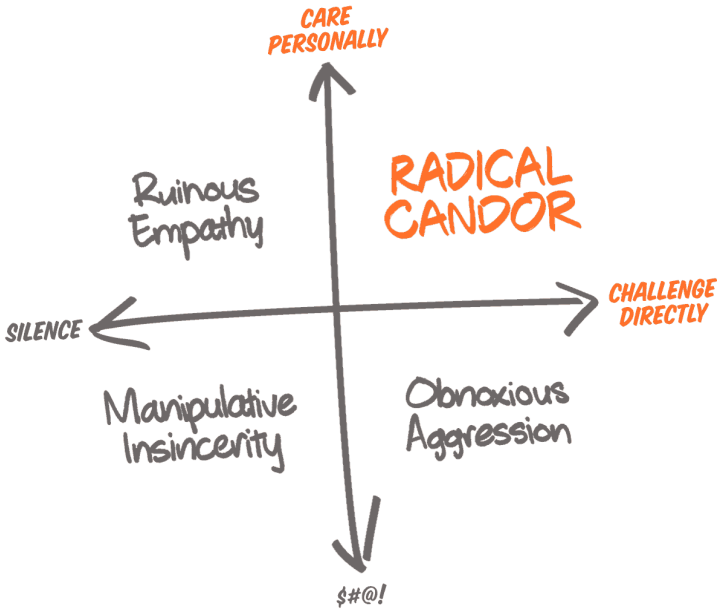
Radical Candor & Psychological Safety

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

Week	Tuesday's Class	Thursday's Class	Due
8/27	Intro & Analog Behavior of Digital Systems	Combinational and Sequential Logic	(Lab Demos)
9/3	Verilog Coding	Synchronous Design	Lab 1 - Development Board Assembly
9/10	FPGA Documentation	Architecture & Assembly Review	Lab 2 - Muxed 7-Segment Display
9/17	Assembly Programming	C Programming on an MCU	Lab 3 - Keypad
9/24	Clock Configuration	Timers	Lab 4 - ARM Assembly Sort
10/1	Serial Interfaces Overview & SPI	UART and the IoT	Lab 5 - Digital Audio
10/8	PCB Design	Advanced Encryption Standard (AES)	Lab 6 - SPI & The Internet of Things
10/15	**Happy Fall Break! No class**	Project Kickoff	
10/22	Graphics and Displays	Motors and Speakers	Lab 7 - AES & Project Proposal
10/29	Interrupts Pt. 1	Interrupts Pt. 2	Proposal Debriefs
11/5	Presentations	Presentations	
11/12	The Fast Fourier Transform (FFT)	Project Status Report and Demo	Project Status Reports & Demo
11/19	Emerging Topics in Embedded Systems	**Happy Thanksgiving! No class**	
11/26	Introduction to Real Time Operating Systems	Direct Memory Access	
12/3	TBD	Interview Questions & Life Beyond Mudd	Project Checkoffs, Report, Demo Day

Important Information

- Course website: https://pages.hmc.edu/brake/class/e155
- 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. A lot.
- **Interleaving (spaced repetition)** – Come back to things again and again, with a little time in between
- **Radical Candor** - care personally, challenge directly

Course Content

Three main sections: labs, project, and in-class 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 10/15-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.

In-class participation

- We will regularly open class with short low-stakes quizzes using Plickers.
- You get full credit for participating (i.e., graded on completion).
- The quizzes will often cover a short (~15 minute) reading to be completed before class on the topic for the day.

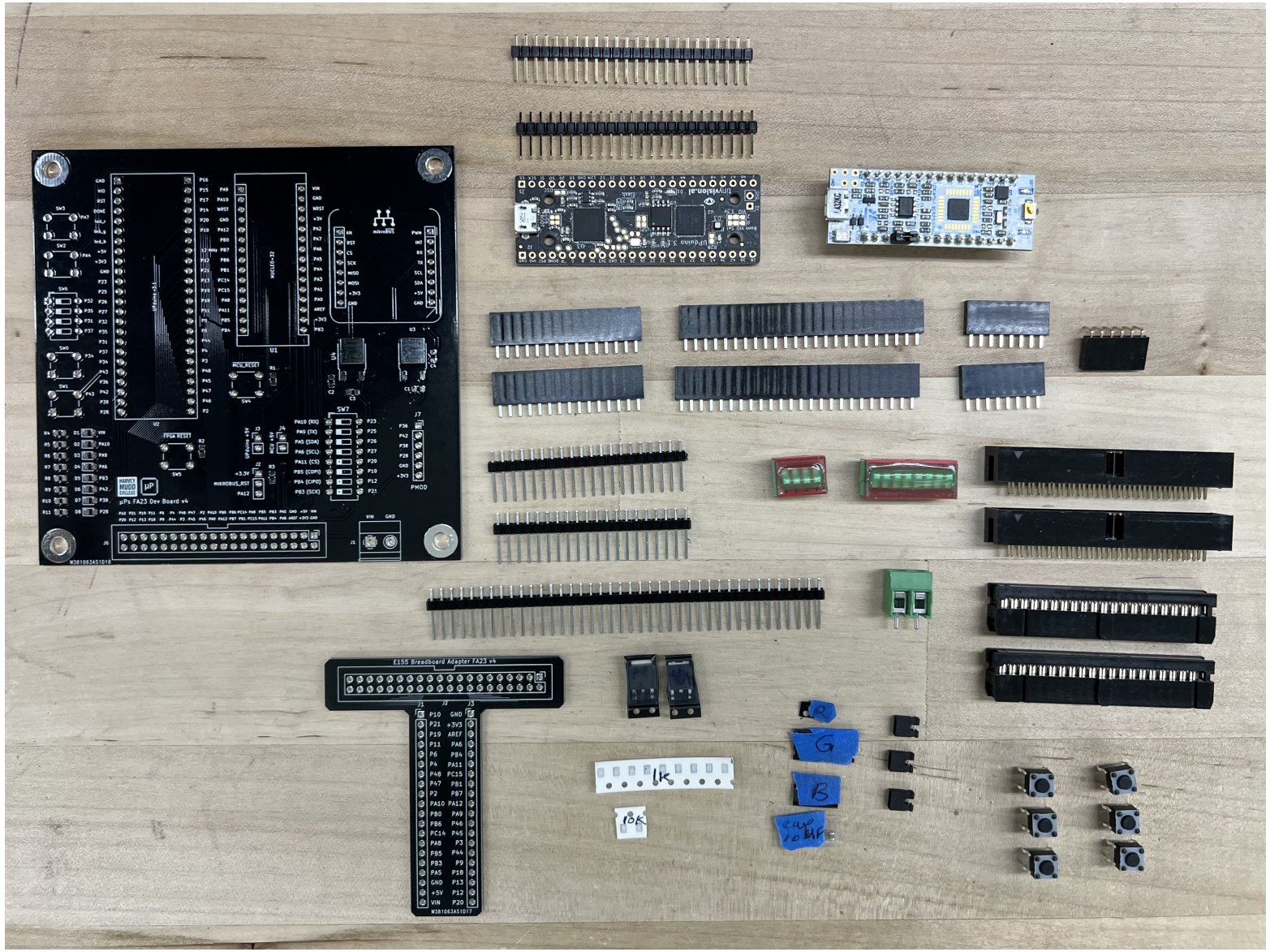
Labs

- Lab 1 - Board Assembly and Testing
- Lab 2 - Multiplexed 7-Segment Display
- Lab 3 - Keypad Scanner
- Lab 4 - ARM Assembly Sort
- Lab 5 - Digital Audio
- 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.

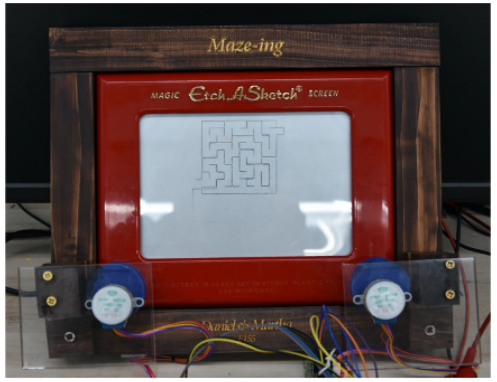
Lab Kit

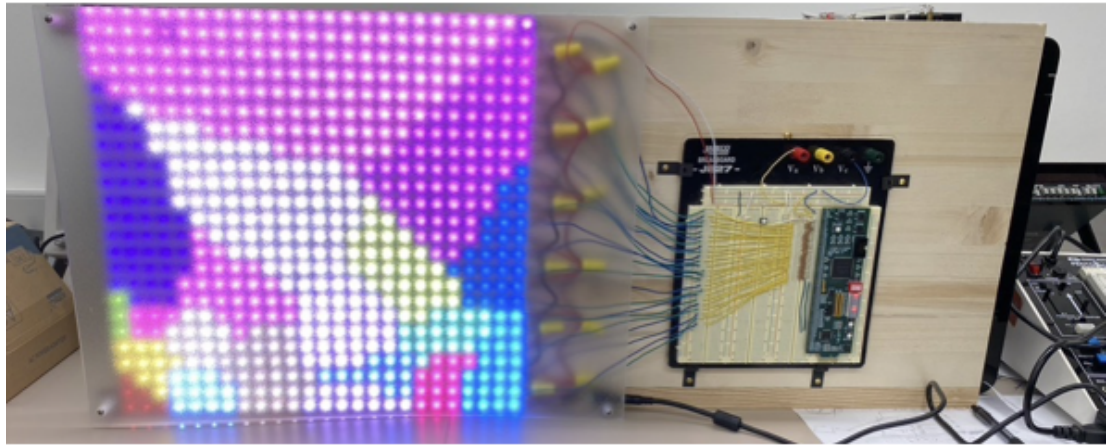

µPs FA23 Dev Board v4
E155 Breadboard Adapter FA23 v4
1K
10K
R
G
B

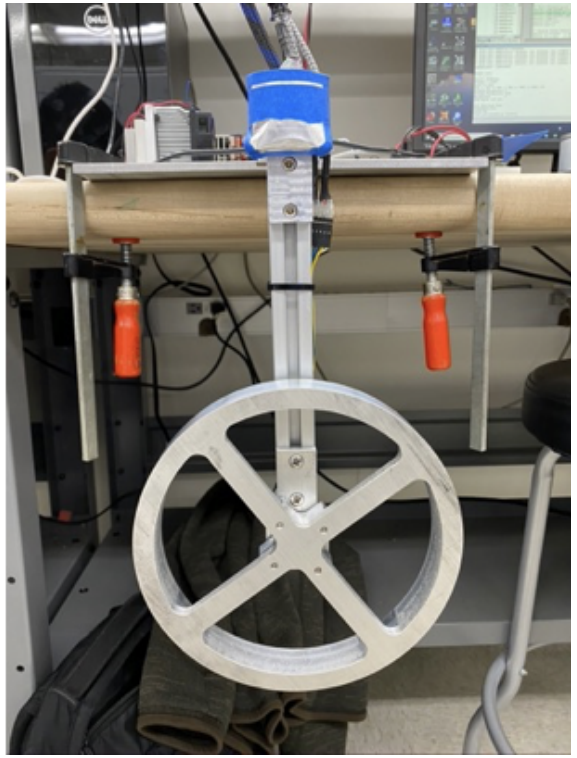
Project

There are 5 deliverables associated with the pro

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







Grading

What are grades for?

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. **No credit**: no attempt or poor attempt made. Design does not meet minimum specifications.
2. **Proficiency**: design meets proficiency specifications but not mastery specifications.
3. **Mastery**: design meets both proficiency and mastery specifications.

Grading

Assignments will be graded on a three point scale designed to reflect your level of mastery of the content contained within it. Every assignment in this class will be presented with a list of specifications with two levels:

1. No Credit
2. Proficiency
3. Mastery

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

Three main bundles where you can score points. For the grade category you must meet all criteria in the respective row.

Grading Breakdown

- PS = proficiency specs
- MS = mastery specs

Grade	Labs (7 total)	Project	Class Participation
D	PS < 4	Less than 4 elements meet PS.	More than 3 unexcused absences.
C	PS >= 5. MS > 3/5.	All elements meet PS. 2/5 meet MS.	No more than 3 unexcused absences.
B	PS >= 6. MS > 4/6.	All elements meet PS. 3/5 meet MS.	No more than 2 unexcused absences.
A	PS 7. MS > 5/7.	All elements meet PS. 4/5 meet MS.	No more than 1 unexcused absence.

Practices for Success

- Start early!
- Get stuck, but don’t stay stuck.
- Read documentation carefully and repeatedly.
- Ask for help (instructors, grutors, 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
- …

Challenges

Dependability

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

Resource Awareness

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

Design Flow

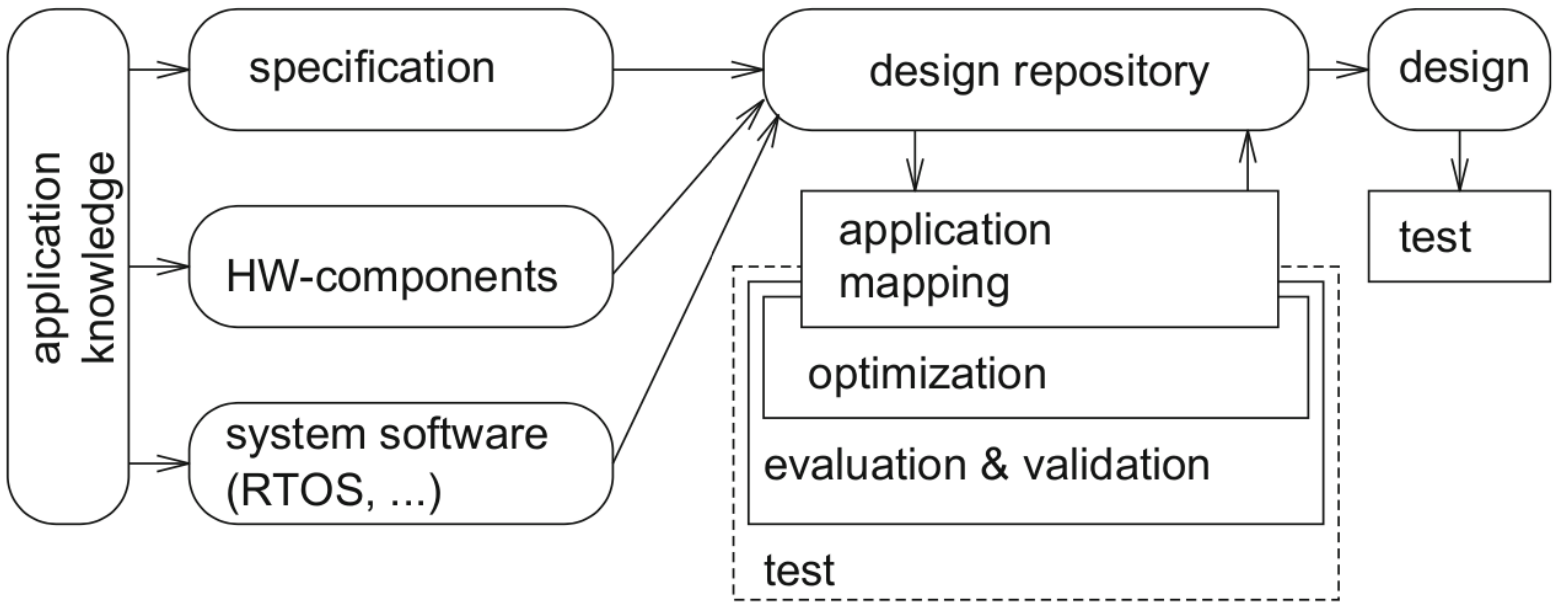


Fig. 1.8 Simplified design information flow

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

Design Flow Unfolded

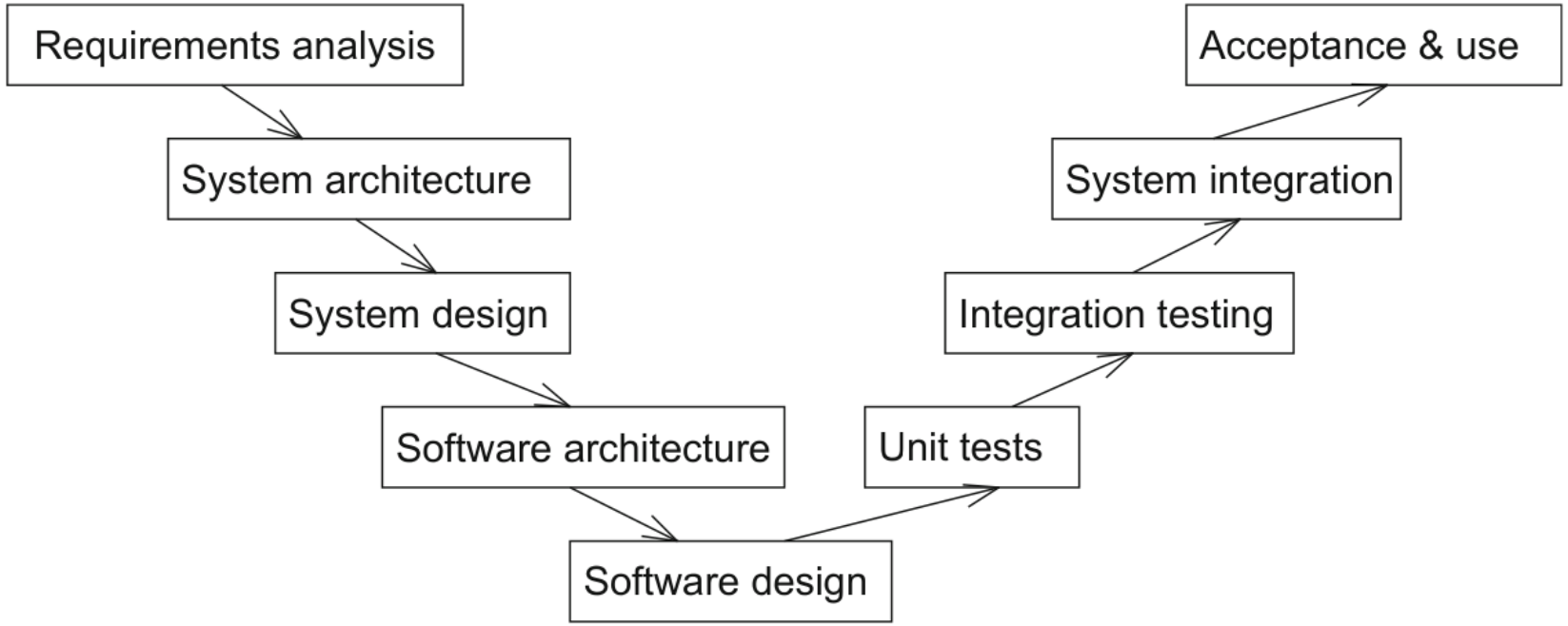
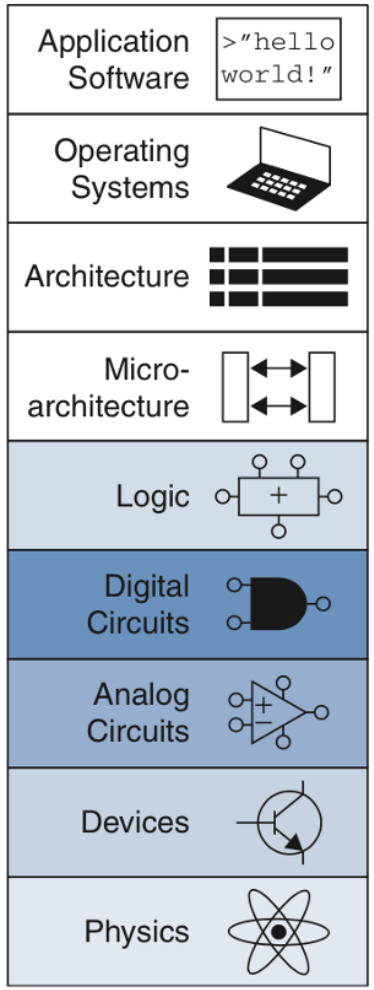


Fig. 1.10 Design flow for the V-model

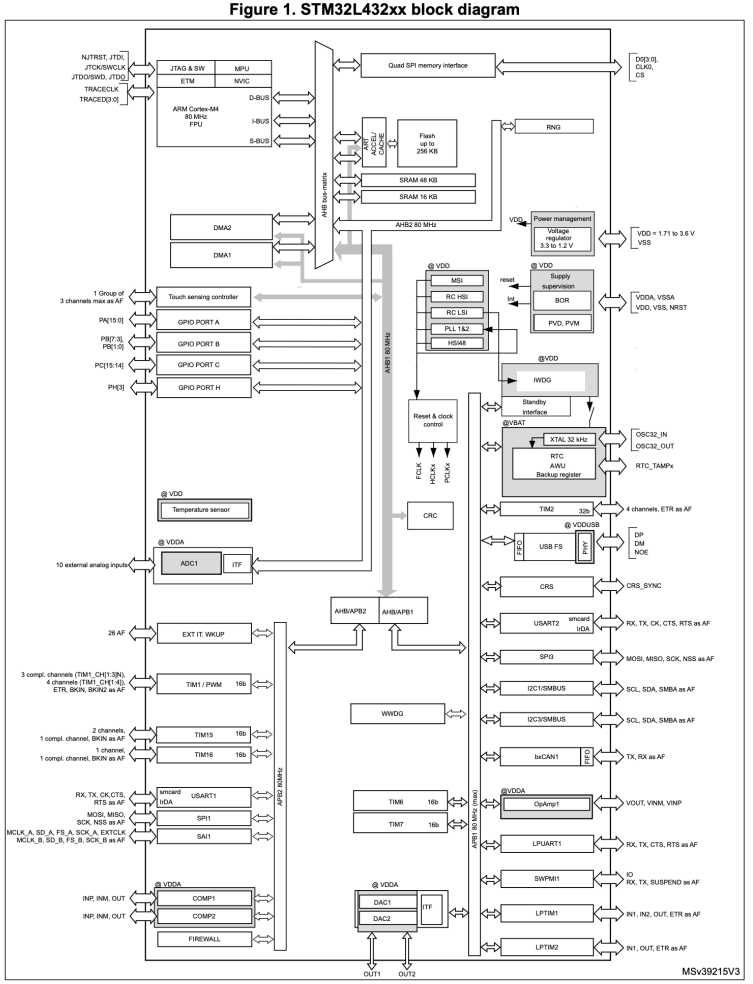
Marwedel, Peter. *Embedded System Design*. Springer, 2018.

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.

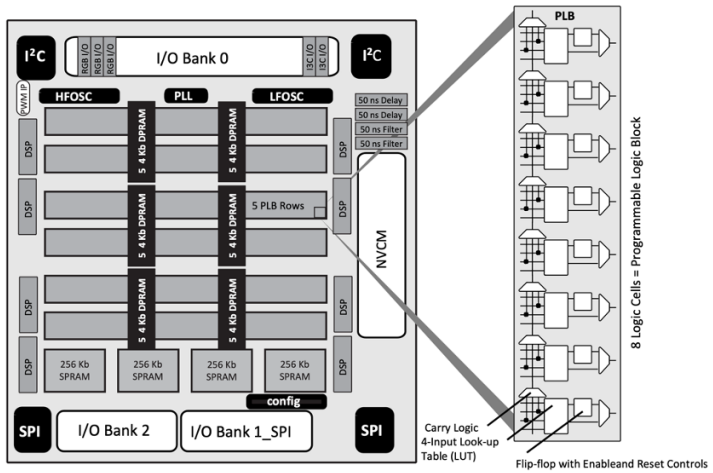


Figure 3.1. iCE40UP5K Device, Top View

iCE40 UltraPlus Family Data Sheet p. 11

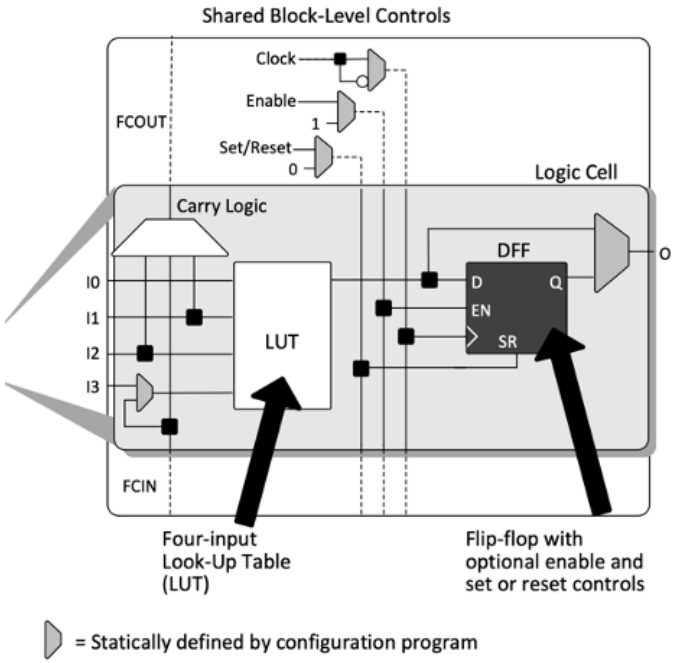


Figure 3.2. PLB Block Diagram

iCE40 UltraPlus Family Data Sheet p. 12

UPduino v3.1

What is an MCU?

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

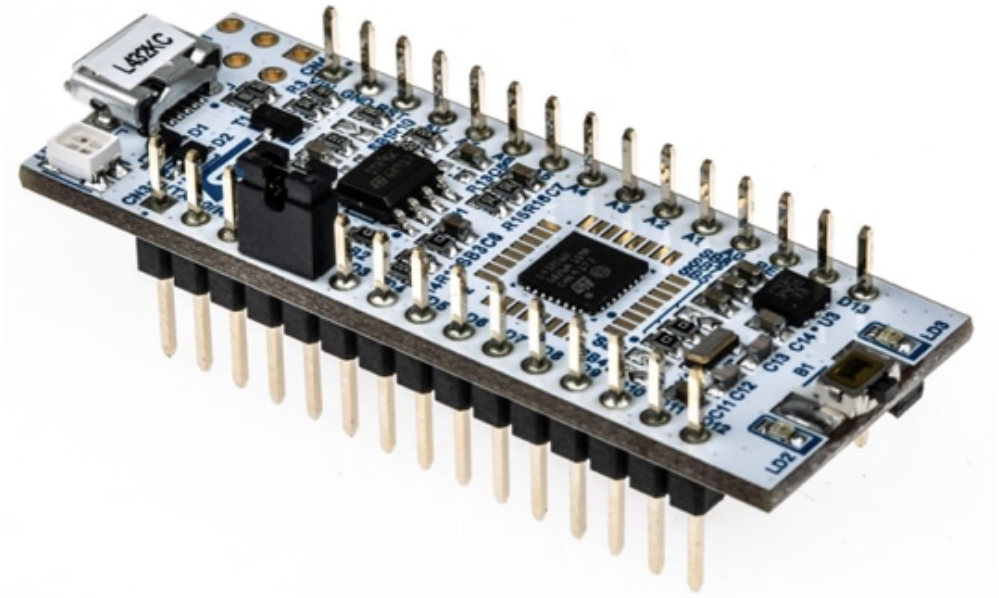
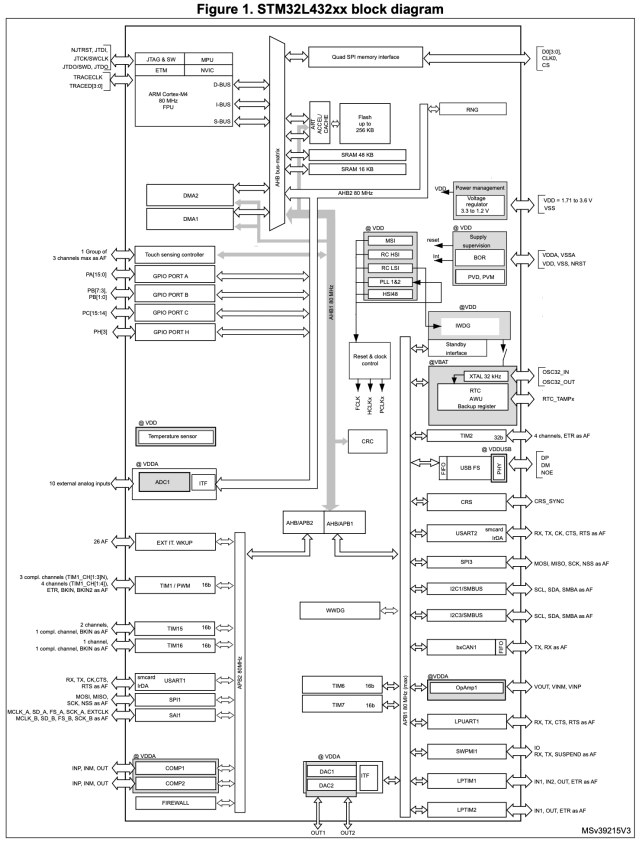


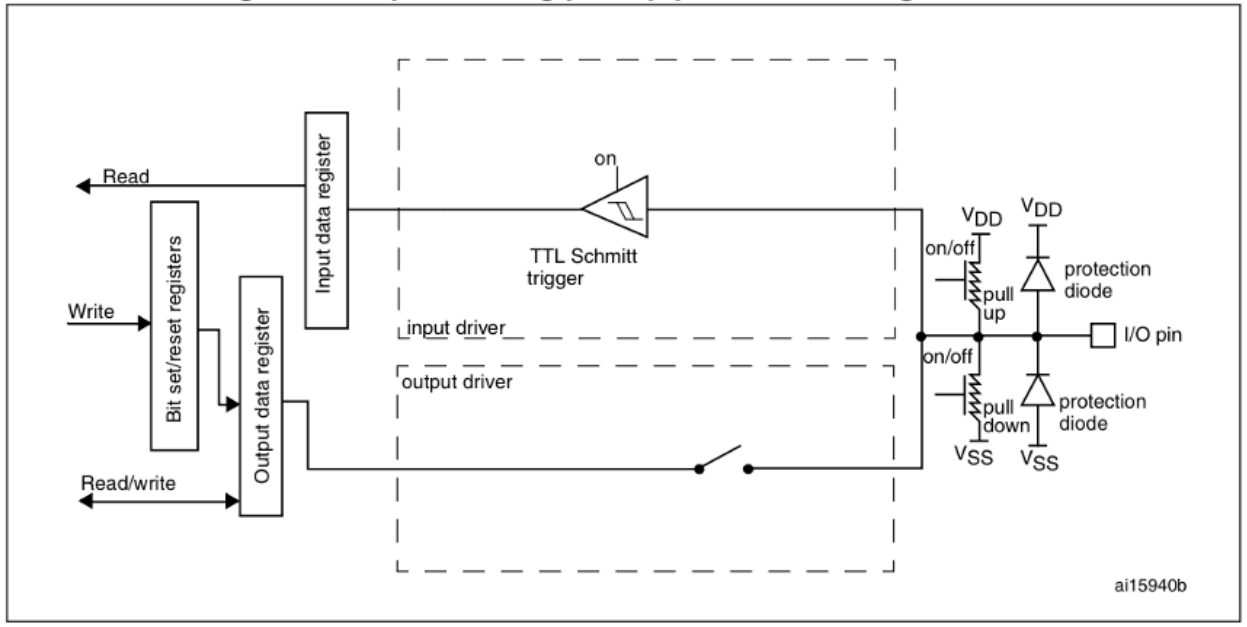
Figure 1. STM32L432xx block diagram



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 flow
  - Clock configuration
  - Peripherals
  - UART
  - SPI
  - Timers
  - PWM

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

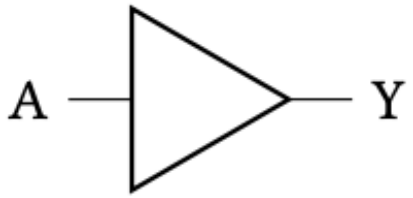


Technical Kickoff: E85 Warp Speed Review

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

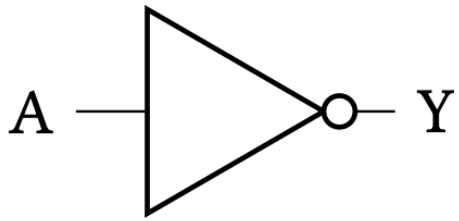
Boolean Expression

$$Y = A$$

Verilog Idiom

```
assign y = a;
```

NOT



Truth Table

A	Y
0	1
1	0

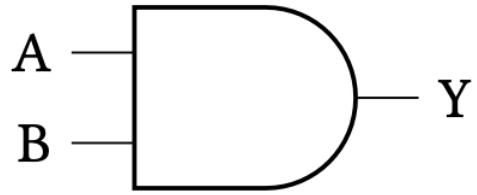
Boolean Expression

$$Y = \bar{A}$$

Verilog Idiom

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

AND



Truth Table

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

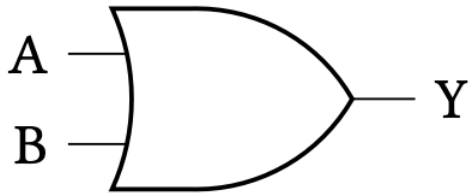
Boolean Expression

$$Y = AB$$

Verilog Idiom

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

OR



Truth Table

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

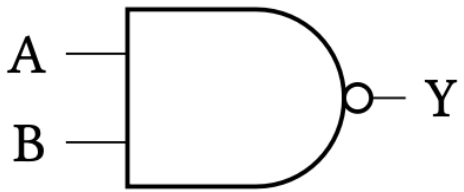
Boolean Expression

$$Y = A + B$$

Verilog Idiom

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

NAND



Truth Table

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

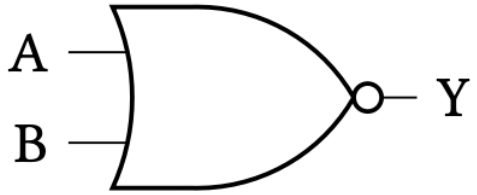
Boolean Expression

$$Y = \overline{AB}$$

Verilog Idiom

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

NOR



Truth Table

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

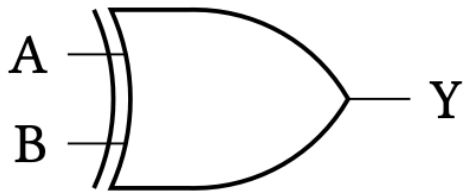
Boolean Expression

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

Verilog Idiom

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

XOR



Truth Table

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

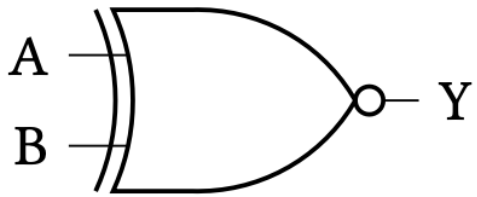
Boolean Expression

$$Y = A \oplus B$$

Verilog Idiom

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

XNOR



Truth Table

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

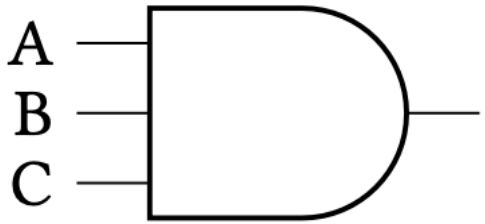
Boolean Expression

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

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

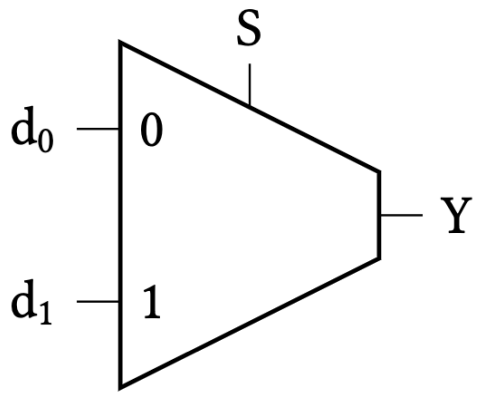
Boolean Expression

$Y = ABC$

Verilog Idiom

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

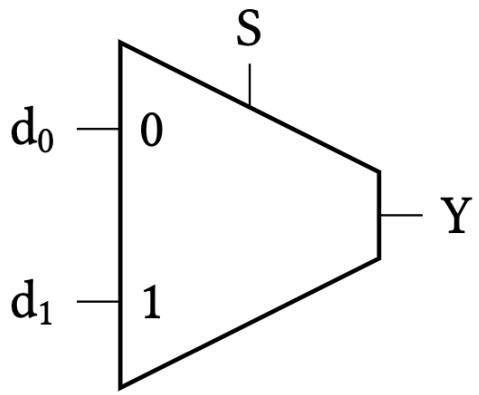
2:1 Multiplexer (MUX)



Verilog Idiom

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

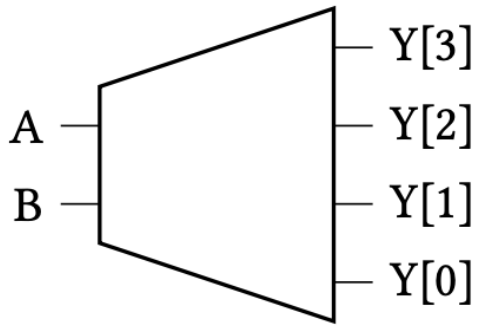
2:1 Multiplexer (MUX)



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

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

```
always_comb
  case(s)
    2'b00:    y = 4'b0001;
    2'b01:    y = 4'b0010;
    2'b10:    y = 4'b0100;
    2'b11:    y = 4'b1000;
    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

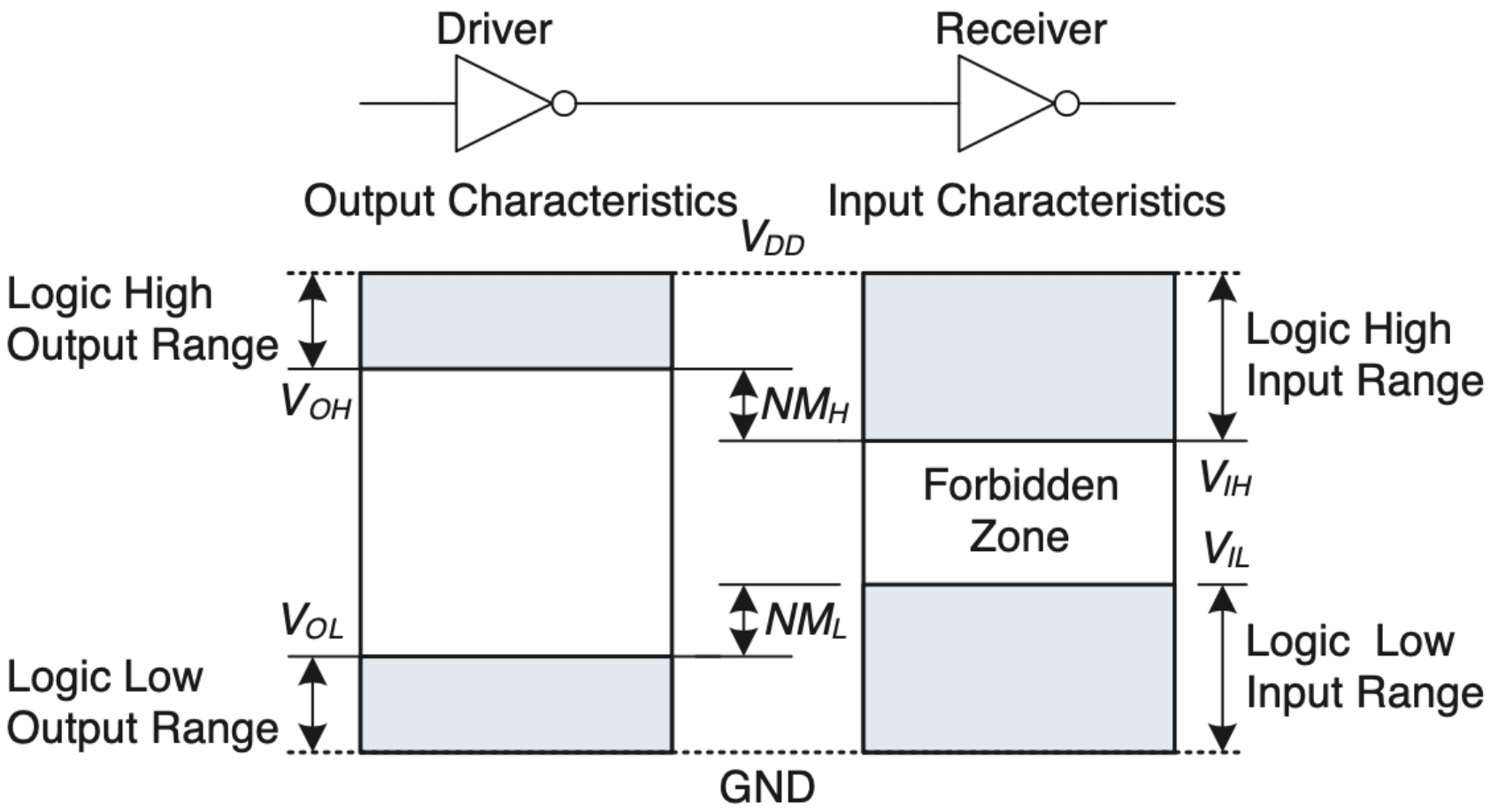


Figure 1.23 from *Digital Design and Computer Architecture: RISC-V Edition* by Harris & Harris

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 Sam 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
  - Soldering refresh
  - Basic FPGA and MCU programming (see tutorials on the website)