Embedded System
SKEM4223
2016/2017/1

Dr Yeong Che Fai
P19A-04-03-22, FKE UTM
Email: cfyeong@utm.my
Website: http://cfyeong.fke.utm.my

Course Synopsys

- This course introduces the principles and applications of embedded system.
- The topics emphasized are the microcontroller system architecture, software programming and the system design.
- The content covers internal peripherals such as general input and output, analogue to digital converter, serial communication interface, timer/counter and interrupt.
- The students will learn the technique to interface the microcontroller system with other devices in the embedded system for real world application.
- Students will also being introduce to ARM based embedded system and application.

Course Learning Outcome

- **CO1** – Able to describe and differentiate the component of an embedded system
- **CO2/P03** – Able to analysis and design the peripherals and the software for various application
- **CO3/P05** – Able to solve the design and development for a real world embedded system application through group project

Course Syllabus

- **Chapter 1** - Introduction to Embedded System
- **Chapter 2** - Microcontroller for Embedded System
- **Chapter 3** - Microcontroller Fundamental
- **Chapter 4** - Embedded System Programming
- **Chapter 5** - Real-time Interfacing
- **Chapter 6** - Single Board Computer
- **Chapter 7** - Case Studies on Embedded System Design

References

- ✓ Shibu K.V. Introduction to Embedded System. McGraw Hill.
- ✓ Frank Wahid, Tony Givargis. Embedded System Design, A unified software and hardware design, John Wiley.
- ✓ Jack Purdum, 2012. Beginning C for Arduino: Learn C Programming for the Arduino, APress.
- ✓ Brian Evans, 2011. Beginning Arduino Programming, Apress.
- ✓ http://www.arduino.cc
- ✓ http://www.atmel.com/devices/ATMEGA328.aspx
- ✓ Qing Li. 2011. Real time concept for Embedded System. Elsevier
- ✓ W. Bolton. Mechatronics Electronic control systems in mechanical and electrical 5th ed.. Peason Education
- ✓ Andrew, N. Sloss, Dominic Symes, Cris Wright, 2008. ARM System Developer's Guide-Designing & Optimizing System Software. Elsevier
- ✓ Craight Hollabaugh. Embedded Linux: Hardware, Software and Interfacing. Addison-Wesley Professional
- ✓ http://www.raspberrypi.org

Assessment

- ✓ **Final Exam:** **50%**
- ✓ **Course Work:** **50%**
  - ✓ Test 1 10%
  - ✓ Test 2 10%
  - ✓ Assignments / Quizzes 10%
  - ✓ Group Project 15%

*** Students are recommended to buy a "Starter Kit for Arduino UNO and Raspberry Pi B+"**

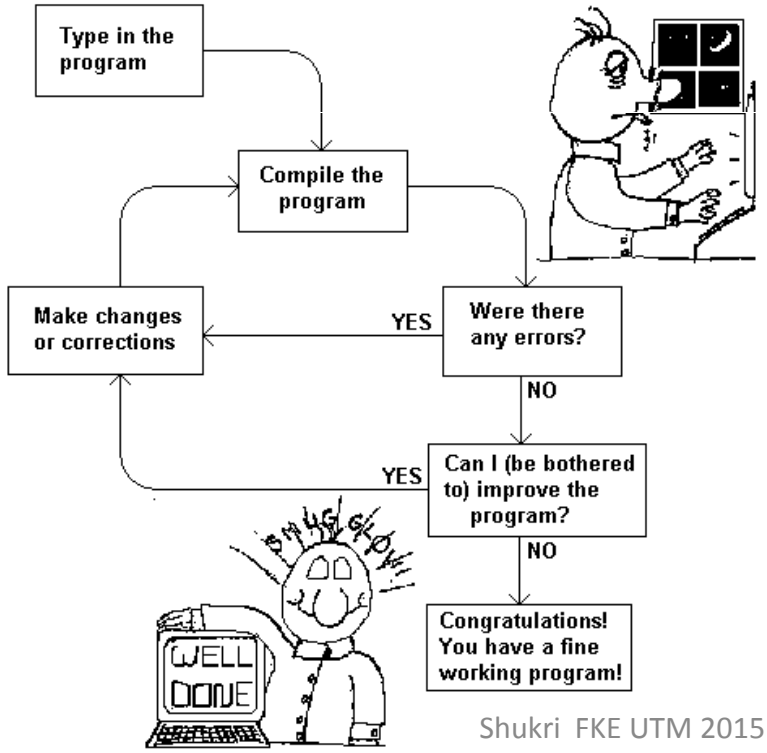
Embedded System Design Process

1. Specifications requirement
2. Conceptualization
3. Analysis
4. Synthesis
5. Verification
6. Documentation

Our Course Learning Process

- ✓ Requirements Specification
  - ✓ Given : Quiz, Assignment , Project, Exam, etc.
- ✓ Conceptualization
  - ✓ Developed by you and your group
- ✓ Iteration Design Cycle
  - ✓ Write program
  - ✓ Compile program
  - ✓ Debug compilation errors
  - ✓ Examine output results
  - ✓ Debug logical errors
  - ✓ Examine hardware results
  - ✓ Debug hardware errors

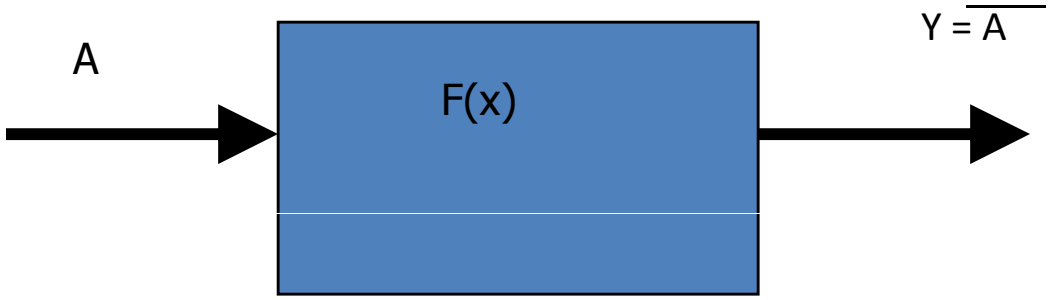
Iteration



Design Abstraction

(How do we "describe" a system?)

Example: Design a "system" which will complement input A



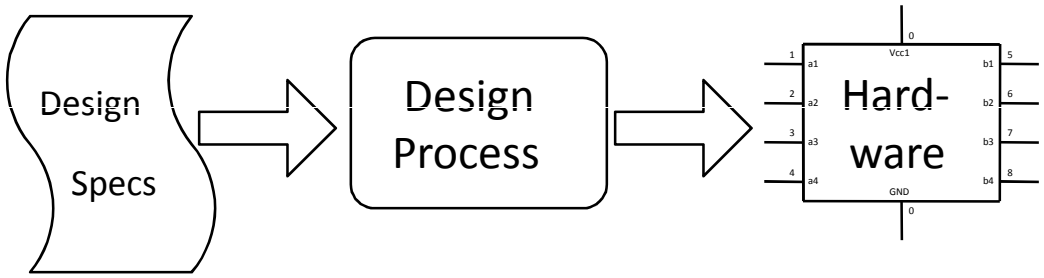
where A **and** Y are single bit values

We can "describe" this design using a logical Truth Table

A	Y
0	1
1	0

Design Abstraction

Our goal is physical or hardware implementations of the design.

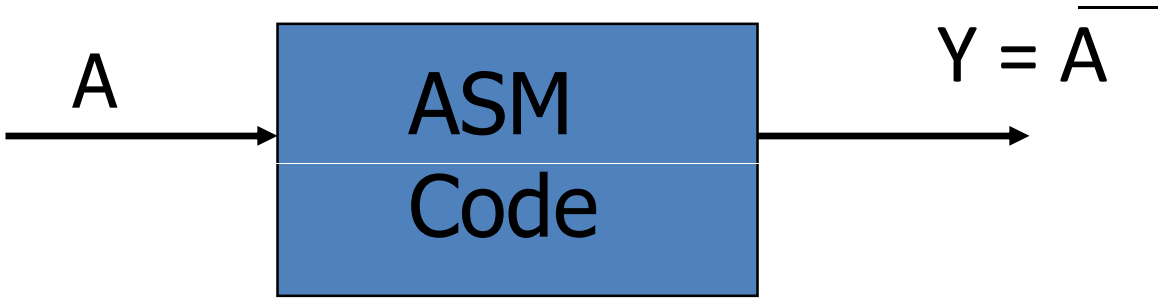


Here, we "design" at several levels of "abstraction"

Levels of Design Abstraction

Example: Design a "system" which will complement input A

1. System Level: Microprocessor / Microcontroller



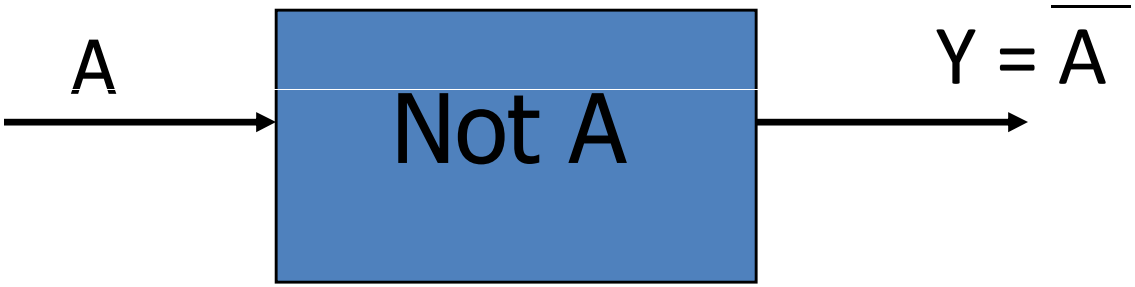
Microprocessor / Microcontroller **Assembly Language:**

```
COMA    ; Complement A
STAA Y  ; Store Accumulator A at Y
```

Levels of Design Abstraction

Example: Design a "system" which will complement input A

2. Behavioral Level: VERILOG / VHDL



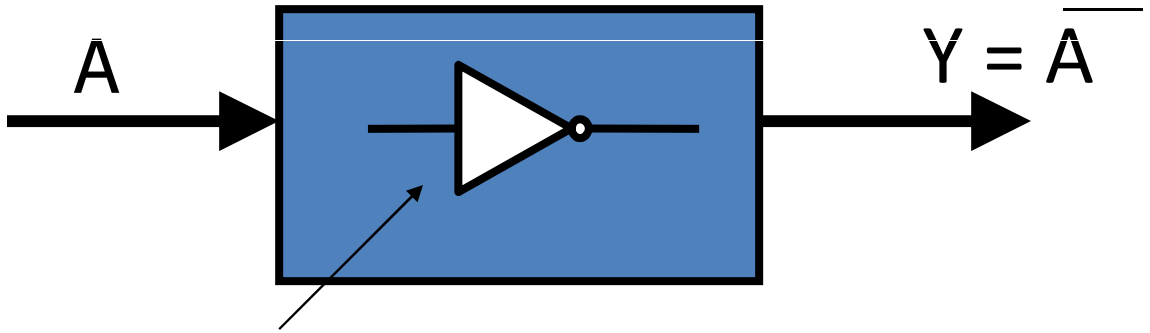
VHDL code: Y <= not A;

* VHDL : VHSIC Hardware Development Language

Levels of Design Abstraction

Example: Design a "system" which will complement input A

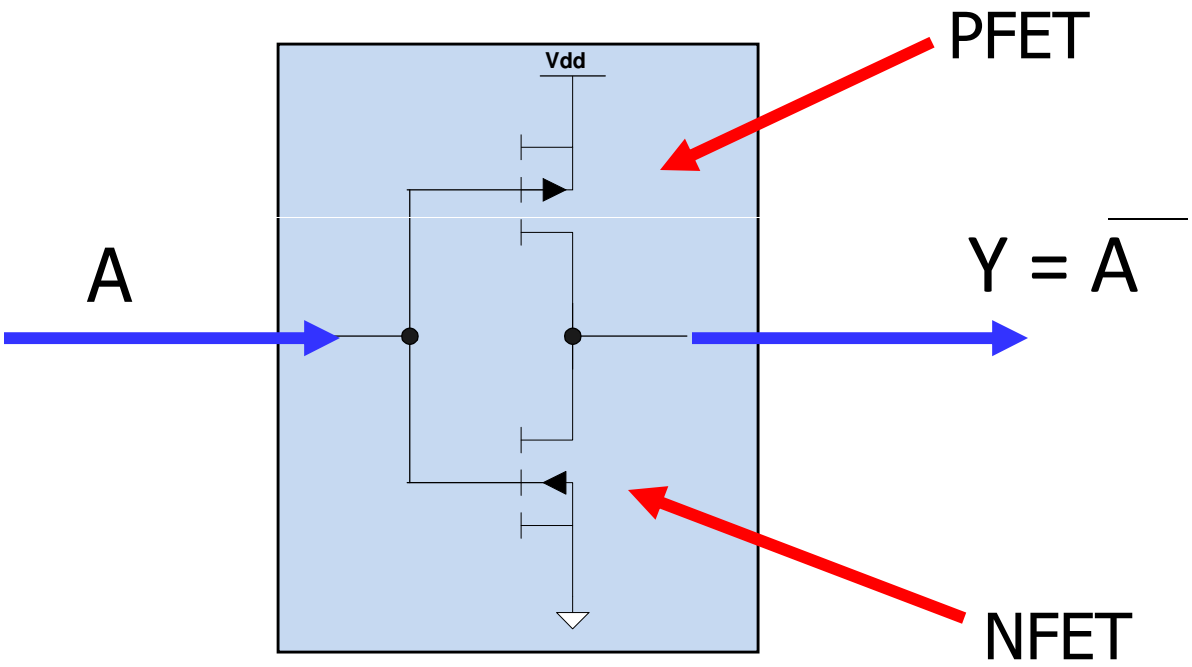
3. Gate Level: Digital Logic Design



Levels of Design Abstraction

Example: Design a "system" which will complement input A

4. Circuit Level: Electronics



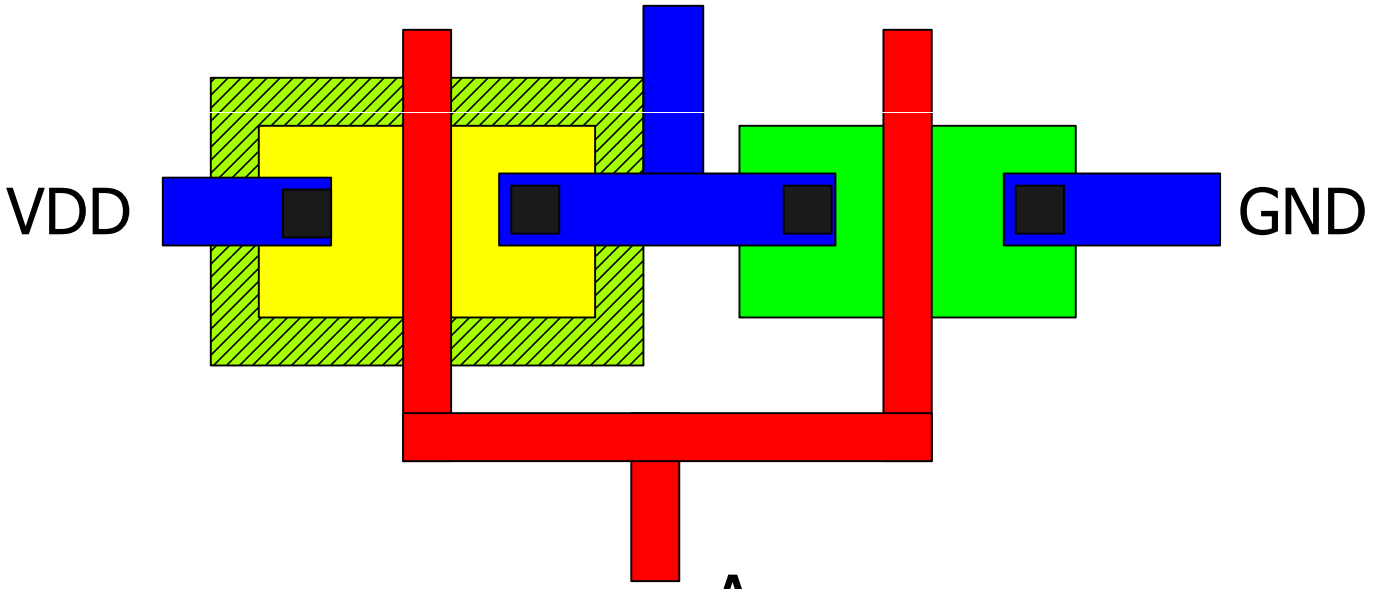
CMOS Technology

Levels of Design Abstraction

Example: Design a "system" which will complement input A

5. Digital IC Design: Digital IC Design

$$Y = \overline{A}$$



CMOS Technology

Summary of Levels

- **"System"**: Assembly Language
- **Behavioral**: VHDL
- **Logical**: Gates
- **Electronic Circuit**: Transistors
- **Integrated Circuit**: IC Layout
- **Fabrication**: IC Processing

  - *All "levels" give you the same result.*
  - *We will learn how to use the **"system"** level to its highest degree of effectiveness.*