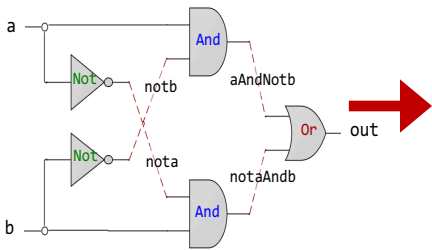
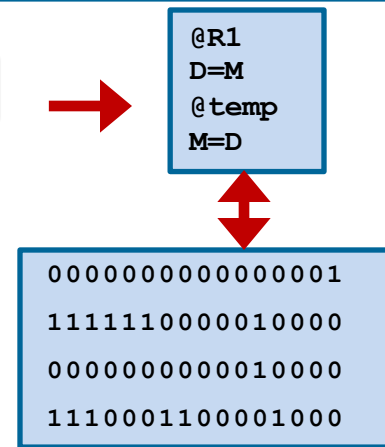
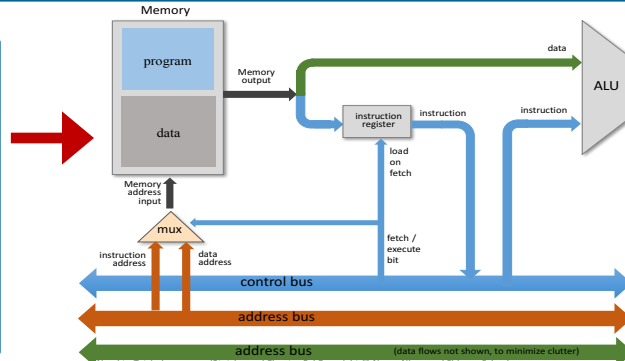




Computer Organization & Assembly Language Programming



```
CHIP Xor {
  IN a, b;
  OUT out;
  PARTS:
  Not(in=a, out=nota);
  Not(in=b, out=notb);
  And(a=nota, b=b, out=w1);
  And(a=a, b=notb, out=w2);
  Or(a=w1, b=w2, out=out);
}
```



Lecture # 01

Overview of the Course

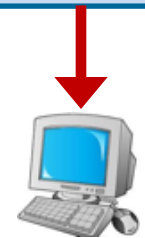
```
#include<stdio.h>
#include<stdlib.h>
int main(){
  printf("Learning is fun with Arif\n");
  exit(0);
}
```

```
global main
SECTION .data
  msg: db "Learning is fun with Arif", 0Ah, 0h
  len_msg: equ $ - msg
SECTION .text
main:
  mov rax,1
  mov rdi,1
  mov rsi,msg
  mov rdx,len_msg
  syscall
  mov rax,60
  mov rdi,0
  syscall
```

```
0: b8 01 00 00 00
5: bf 01 00 00 00
a: 48 be 00 00 00 00 00
11: 00 00 00
14: ba 1b 00 00 00
19: 0f 05
1b: b8 3c 00 00 00
20: bf 00 00 00 00
25: 0f 05
```

Slides of first half of the course are adapted from:
<https://www.nand2tetris.org>
 Download s/w tools required for first half of the course from the following link:
<https://drive.google.com/file/d/0B9c0BdDjz6XpZUh3X2dPR1o0MUE/view>

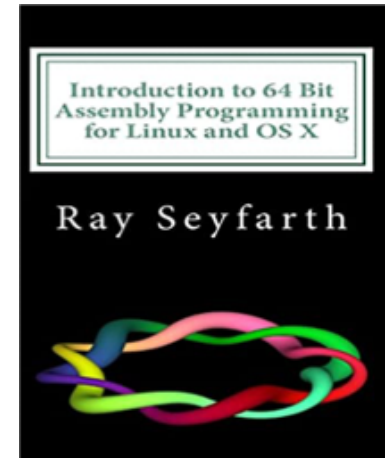
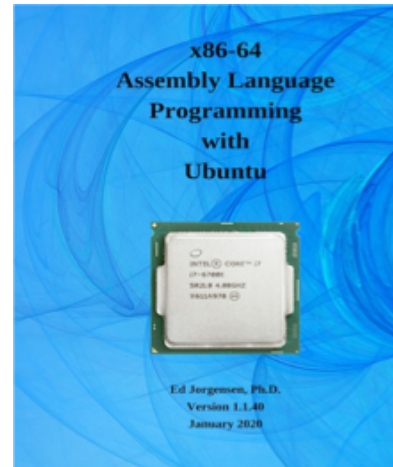
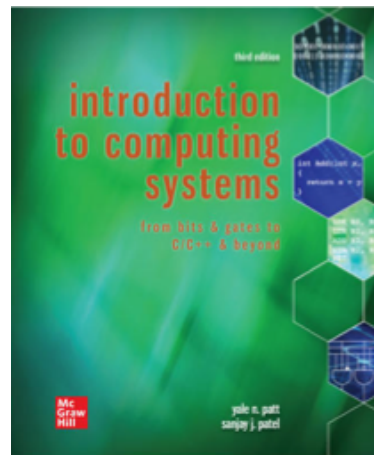
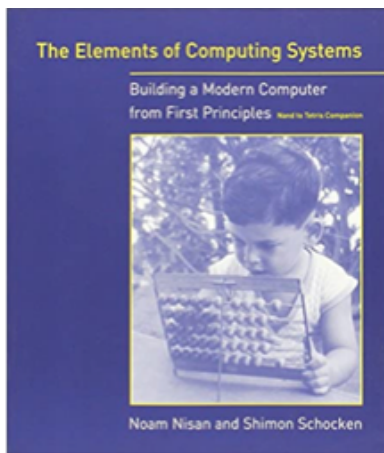
Instructor: Muhammad Arif Butt, Ph.D.





Course Information

- Required Textbooks:
 - The Elements of Computing Systems, Building a modern computer, by Noam Nisan and Shimon Schocken, 2nd Ed, Published in 2020, ISBN-13: 978-0262640688
 - Introduction to Computing Systems: from bits and gates to C and beyond, by Yale Patt and Sanjay Patel, 3rd Ed, Published in 2020, ISBN13: 9781260150537
 - X86_64 Assembly Language Programming with Ubuntu, by Ed. Jorgensen, January 2020
 - Intro to 64 bit Assembly Programming for Linux and OS X, by Ray Seyfarth, Published in 2013, ISBN-13: 978-1484921906





Course Information

- Grades Website: <http://online.pucit.edu.pk>
- Resources Website: <http://arifbutt.me>
- Course Prerequisites : Nil
- Students Counseling hours:
 - Mentioned on <http://arifbutt.me>
- Teaching Assistant info:
 - Mentioned on <http://arifbutt.me>
- 24 hour turnaround for email: arif@pucit.edu.pk



Where to find Stuff?

Where to find stuff?

<http://www.arifbutt.me>

- Lecture Slides
- Quizzes + Assignments + Labs
- Announcements
- Teaching Assistants
- SOPs and Course related Policies
- Download s/w tools, codes and other resources required for the course from the following link:

<https://bitbucket.org/arifpucit/coal-repo>



Lecture Format





Lab Format

- Please come to Labs (**in time**)
- Quizzes might be taken in class or in Lab, so don't miss
- Contents covered in the Lab will come in the Quizzes as well as in the Mid and Final exams



How Will You Be Evaluated?

- Final exam: 40%
- Mid-exam: 35%
- Sessional: 25%
 - Surprise Quizzes: 15%
 - Assignments / Home Tasks: 10%





Surprise Quizzes

- There will be surprise quizzes, given at the start of a lecture, during any lecture. The total number of quizzes could be anywhere between 4 and 40
- **NO LATE or MAKEUP SURPRISE QUIZZES**, under any circumstances whatsoever
- Surprise quizzes are completely individual efforts
- Your best strategy is to play it safe – attend every lecture and do the reading/programming assignments



Cheating Policy

- Academic integrity
- Both the cheater and the student who aided the cheater will be held responsible for the cheating
- The instructor may take actions such as:
 - require repetition of the subject work,
 - assign 'zero' or may be 'negative' marks for the subject work,
 - for serious offenses, assign an **F** grade for the **course**





Late Policy for Home Works and PA

- Late policy for Assignment, Quizzes, and other deliverables
 - No late Assignment submissions!
 - No late quizzes or exams!
- Sticking to dates is your responsibility!
 - Check announcements on lecture notes regularly
- Your best strategy is to play it safe – submit everything on time



Playing Safe in CS-223

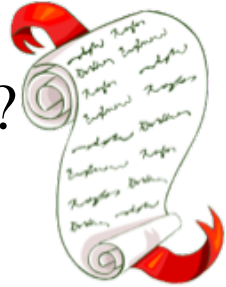
If you follow these 4 simple rules during the CS223 class, you'll make sure that you do well in the course:

1. Attend every lecture + Lab
2. Study/Understand the course material (textbook sections assigned + slides + Reading assignments), and practice the concepts on the provided tools (H/W simulator, CPU emulator, Assembler,...)
3. Submit everything (PAs, HWs, quizzes, exams) on time - don't be late
4. Don't cheat



Today's Agenda

- Why do Scuba Diving?
 - Why understand Computer Organization & Architecture?
 - Why understand Assembly Language?
 - Why program in Assembly when we have HLL(s)?
- Discussion on Course Matrix:
 - Pre-Mid Course
 - Post-Mid Course





Problem Solving on Computer

- Human Thought
- Algorithms
- Applications Software
- Systems Software / Compiler
- OS/Runtimes
- **Assembly Language**
- **Machine Language (Instruction Set Architecture)**
- **Microarchitecture (core + memory hierarchy)**
- **Logic Design**
- Device Level
- Physical Design
- Semiconductors/Silicon used to build transistors
- Properties of atoms, electrons, resistors, capacitors



Two Recurring Themes

Abstraction:

- Use of abstraction is all around us
- Take me to the air port
- Go straight 1.2 km, then make a right turn, go down 500 m, then take a left, then go straight for another 750 m, then take a right and so on
- Abstraction is a technique for establishing a simpler way for a person to interact with a system, removing the details that are unnecessary for the person to interact effectively with that system
- It is a productivity enhancer – don't need to worry about details, until some thing goes wrong! And then, it becomes important to understand the components and how they works together



Two Recurring Themes (cont...)

Hardware vs. Software:

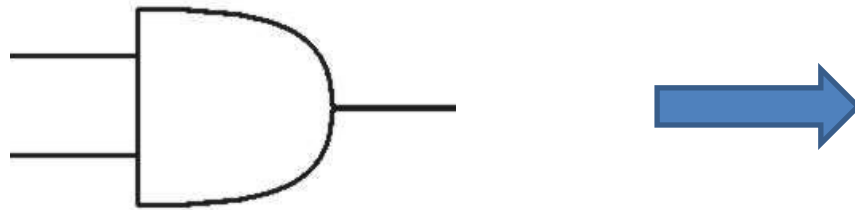
- Both are components of a computer system. Even if you specialize in one, you should understand capabilities and limitation of both
- Data types vs finite word length of a computer
- Functions vs function calling convention
- Recursion vs memory layout
- Pointers vs memory layout
- Data structures vs memory layout

**A computer scientist can design much better solutions,
when he/she has a mastery of both the worlds!**

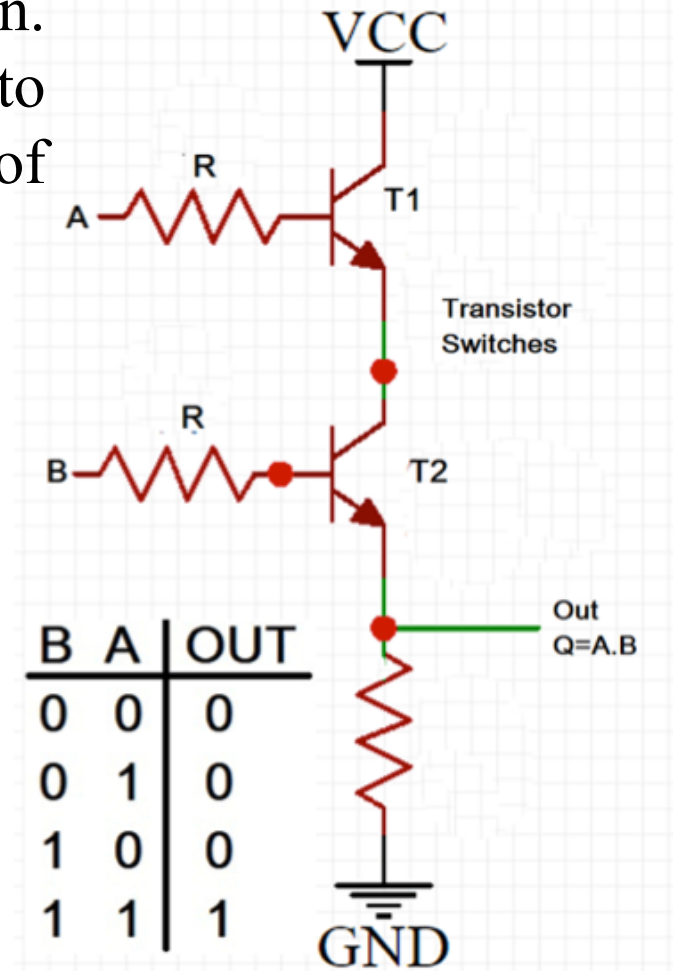


The Notion of Abstraction: (And Gate)

- Every layer in CS is an abstraction. Depending on which layer you want to live at, you will have different views of the computer



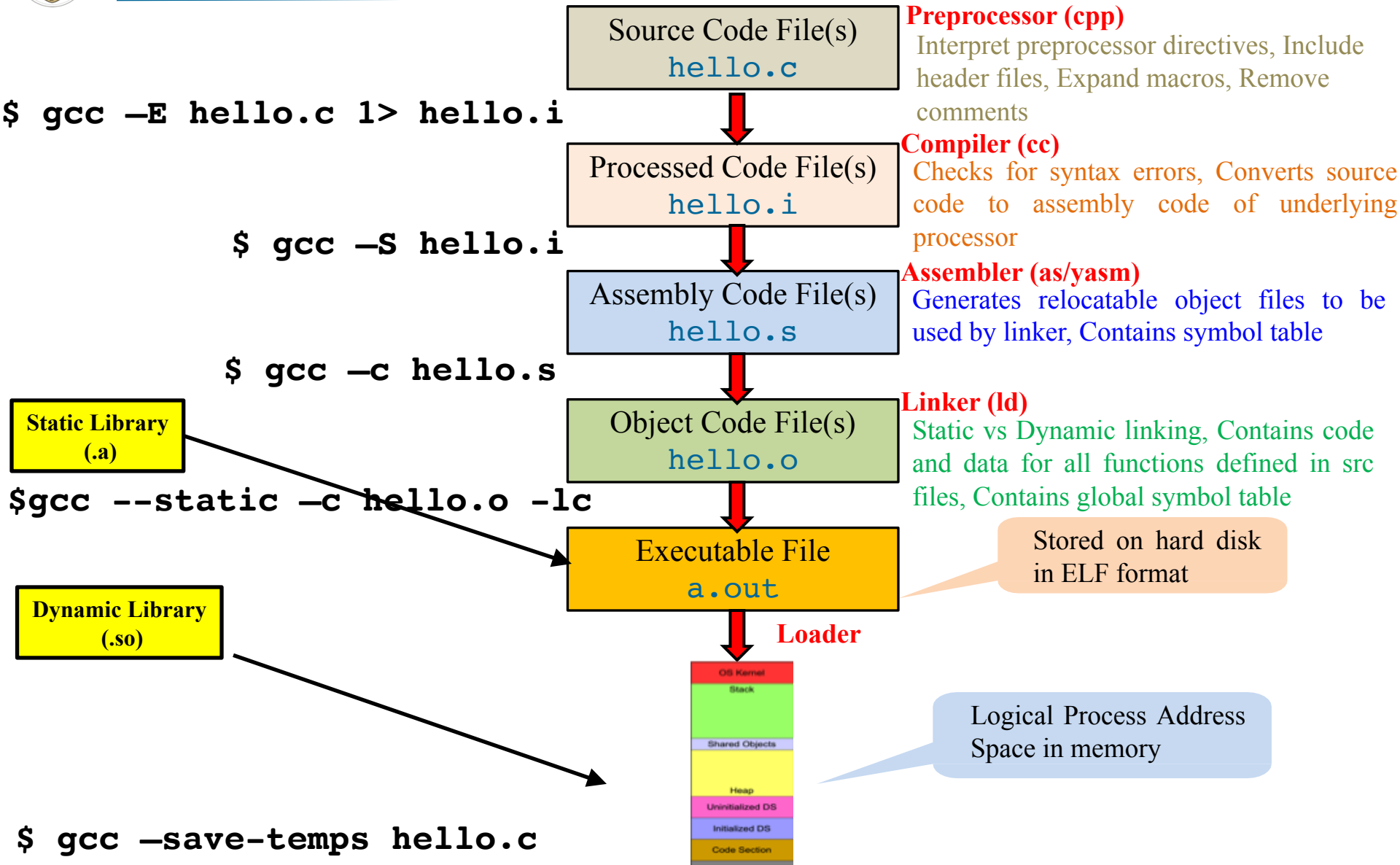
- A transistor is an electronic device that has three ends: a source, a sink, and a gate
- An Intel processor measuring less than a square inch has well over 1.5 billion transistors on it



AND gate using NPN transistor

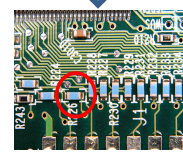
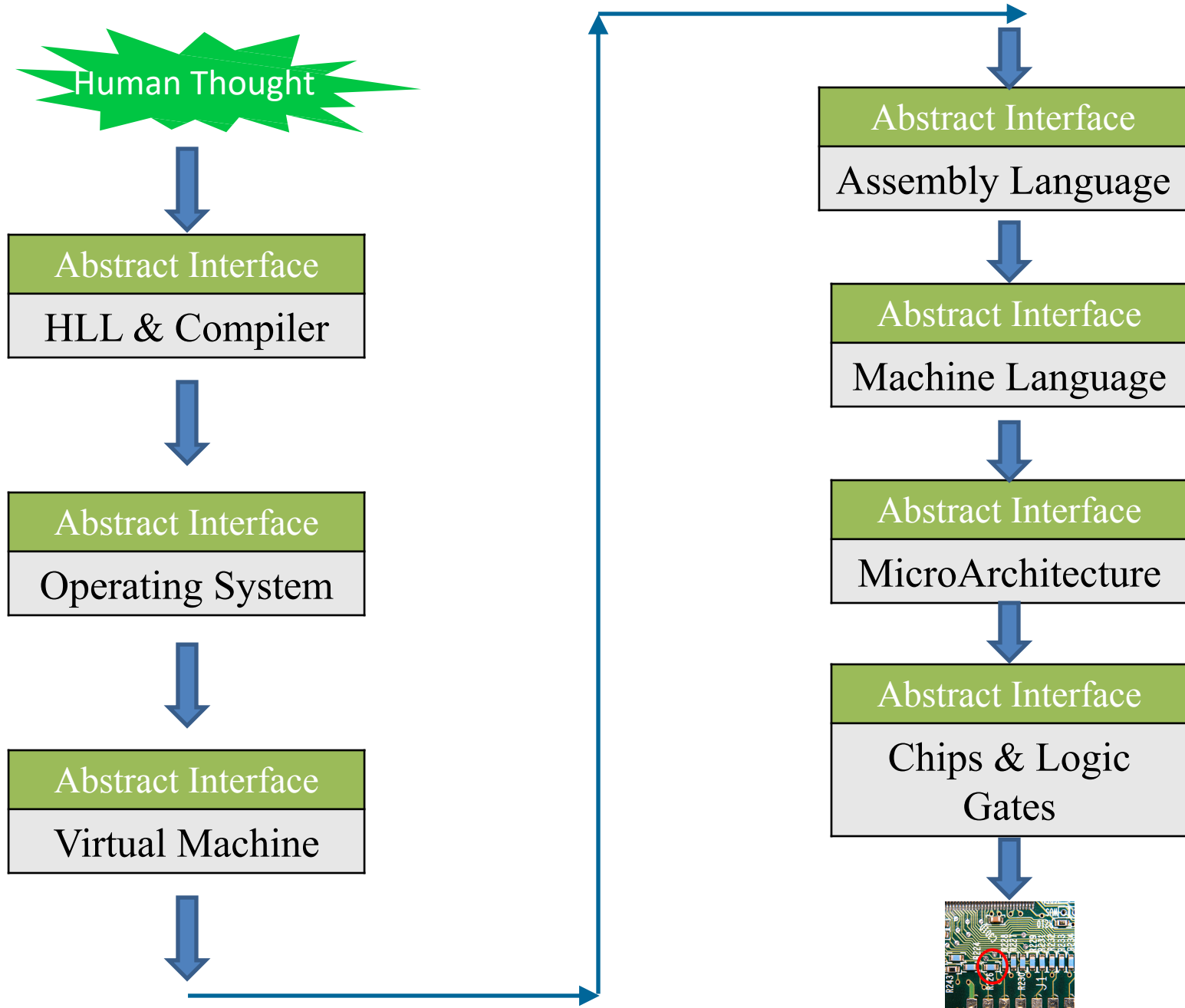


The Notion of Abstraction: (C-Compilation Process)



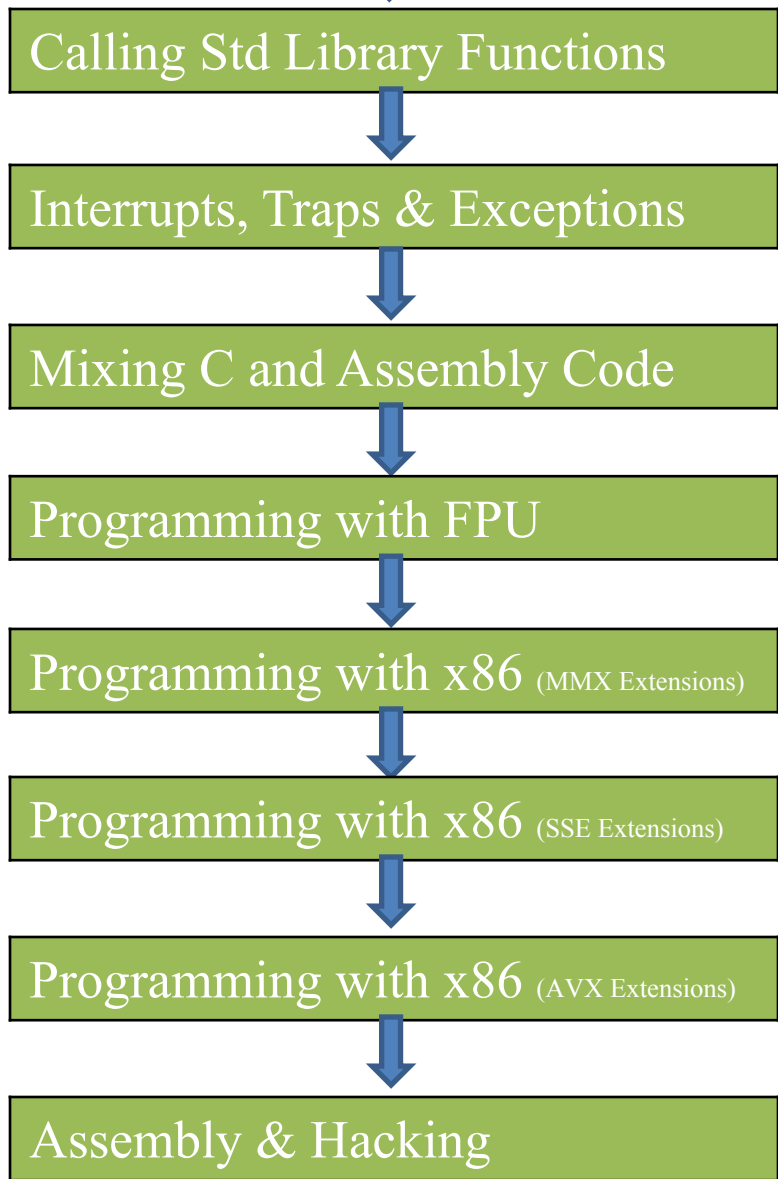
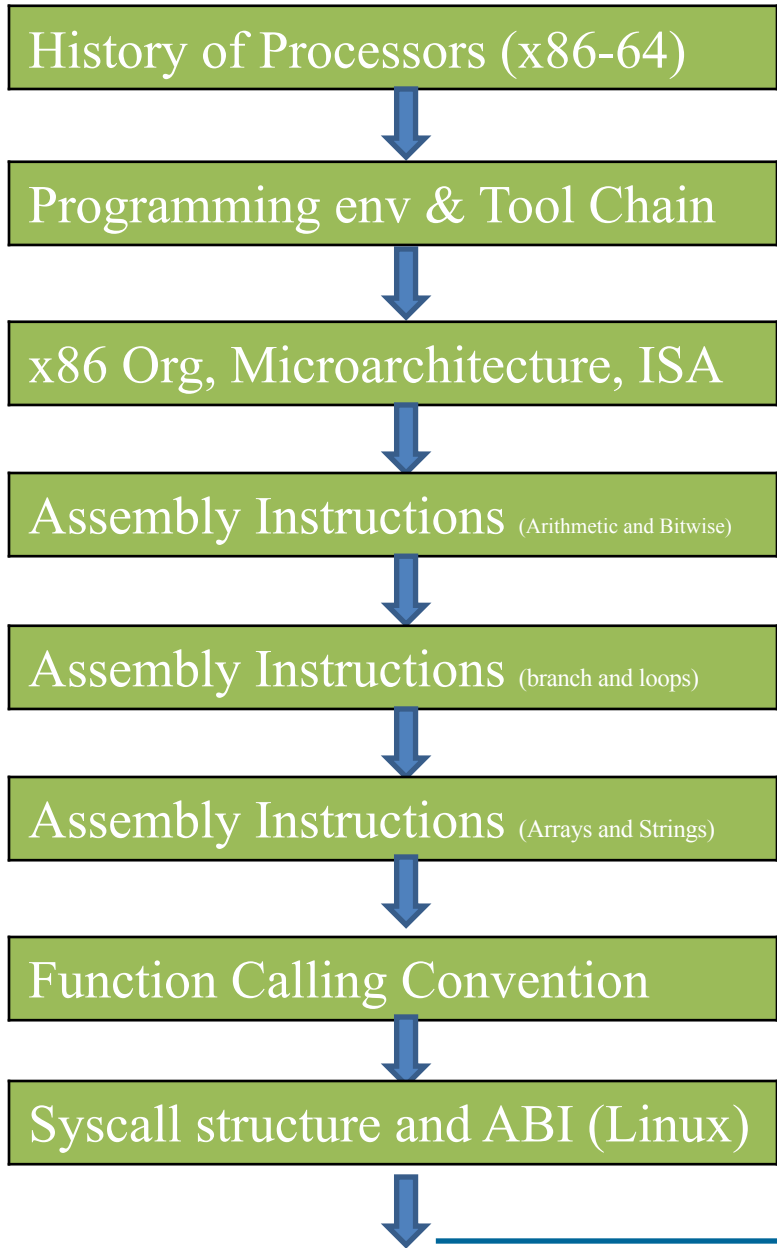


Pre-Mid Course Matrix Discussion





Post-Mid Course Matrix Discussion





Things To Do

- Download Books from course bitbucket repository:

<https://bitbucket.org/arifpucit/coal-repo>

and give their contents a bird's eye view

- Visit <http://www.arifbutt.me> and check out the resources available over there for this course. Keep visiting for announcements of quizzes and assignments
- Prepare a summary of your notes for this video session and make it your habit for all the remaining videos. This will really help you during your exams



Coming to office hours does NOT mean you are academically week!