Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran
Department of
Computer Science and Engineering

PES UNIVERSITY ONLINE

Microprocessor & Computer Architecture (µpCA)

Overview

Dr. D. C. Kiran
Department of Computer Science and Engineering

What we Know?

Digital Design!

- In DDCO course, *increase logic circuit speed, decrease logic resources required* and *decrease power consumed*
- *Goal is to* Optimize fundamental physical quantities of *time, space and energy.*

Computer Organization !

- Hardware Abstraction above Digital Logic and below Operating System.
- Hardware details which are required by programmers, such as Datapath of Processor, Control signals, Interfaces and Memory technology used.

Microprocessor & Computer Architecture (µpCA)

What we study?

What is µicroProcessor?

- Basic Computing Unit on an Integrated Chip (IC)
- Brain of the Computer

What is Computer Architecture ?

- The engineering decisions and tradeoffs that must be made in order to produce a ***"Good" Design"***.

  i.e Execution Time, Space and Resource

Quote for the Day!

"Manufacturers offer family of Computer models with same Architecture but with different Organization"

Microprocessor & Computer Architecture (µpCA)

Syllabus

Unit 1: Basic Processor Architecture and Design

Unit 2: Pipelined Processor and Design

Unit 3: Memory Design

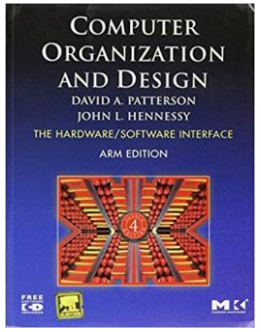
Unit 4: Input/Output Device Design

Unit 5: Advanced Architecture

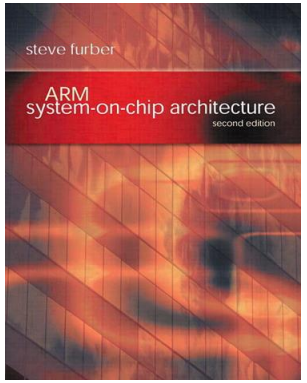
Microprocessor & Computer Architecture (µpCA)

Text and Reference Book(s)

T1: "Computer Organization and Design", Patterson, Hennessey, 5th Edition, Morgan Kaufmann, 2014.



T2: "ARM System-on-Chip Architecture", Steve Furber, 2nd Edition, 2015, Pearson India.



R1: ***"Computer Architecture: A Quantitative Approach", Hennessey, Patterson, 5th Edition, Morgan Kaufmann, 2011.***

R2: "The Definitive Guide to the ARM Cortex-M0 and Cortex MO+ processors", Joseph Yiu, 2nd Edition, Newnes, 2015.

Microprocessor & Computer Architecture (µpCA)

Evaluation

ISA 1: 24%

ISA 2: 16%

Quiz: 10%

ESA : 50 %

Administration of 10% Quiz

- Four regular Quiz, each 4% (Unit 1, Unit 2, Unit3&4, Unit 4&5)
- One improvement Quiz for 4% (Covering all the Units)
- Student should score maximum of 10% in cumulative fashion from all 5 Quiz

Example: Student can score 2 % +2% +2 % +2 % +2% = 10%

3 % +3 % +3 % +1 % = 10%

2 % +3 % +2 % +4 % = 11-1 = 10%

4 % +4 % +4 % = 12-2 = 10%

4 % +4 % +4 % +4 % +4 % = 20-10 = 10%

Note: No Extra or Make-up Quiz will be conducted if student miss quiz component. However, He/ She should put more effort to compensate the missed quiz in the remaining quiz

Microprocessor & Computer Architecture (µpCA)

Team of Faculty Members

EC Campus:

Dr Kiran. D. C.
Prof Shanthala. T. P.
Prof Deepthi. C.

RR Campus:

Prof V. R. Badri Prasad
Prof Roopa Ravish
Prof Supriya. M. C

Motivation

Who am I?

I am Computer Engineer

Why to Study µpCA?

Programming Requirements?

- Correct
- Speed
- Less Power Consumption
- Less or Fast Memory Utilization
- Less or Fast Communication

Microprocessor & Computer Architecture (µpCA)

What I / we Do?

- Programmer / Developer / Analyst

- Probably do not have to worry about CA /CO, like we need to know what our cars look like under the hood in order to drive them.

- Certainly, write high level language programs without understanding how these programs execute.

- Use various application packages without understanding how they really work

To Be Thought

A program written needs to be faster and more efficient.

Who is responsible for keeping CPU busy?

The application we are using does not do precisely what we want?

Example: 2+3*4 is 14 not 20

Microprocessor & Computer Architecture (µpCA)

BYTES from Software Professionals

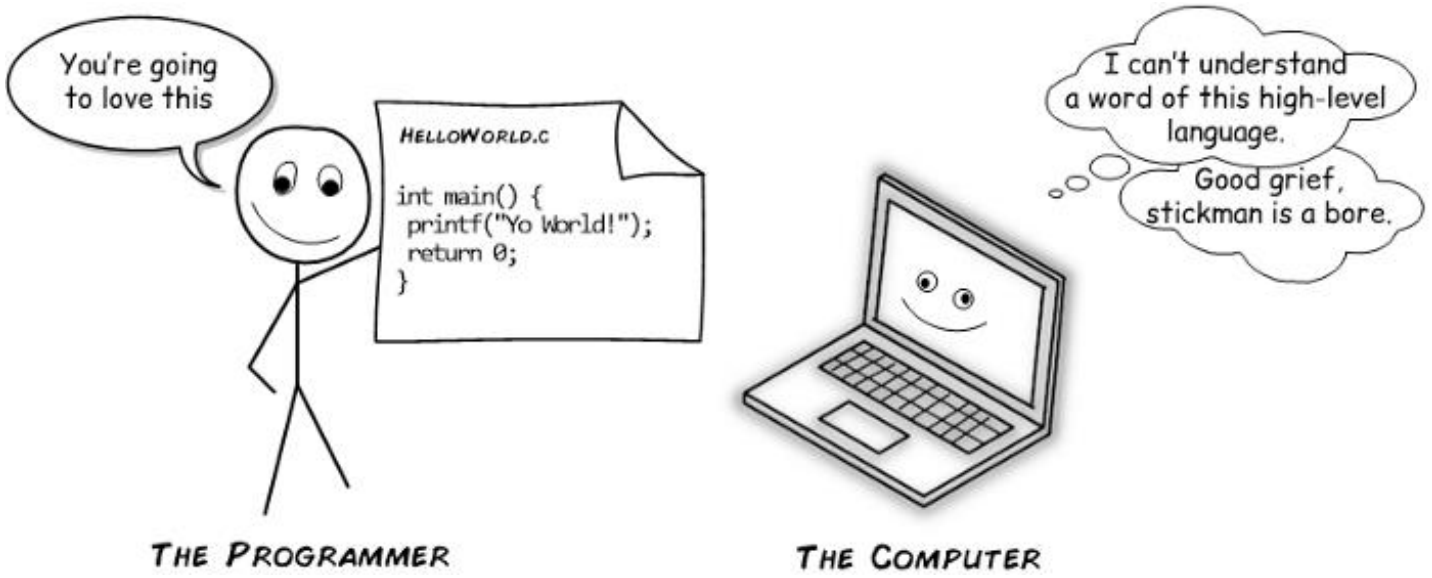
Knowing what is inside data structure and how it works will help you design, develop, and implement applications better, faster, cheaper, more efficient, and easier to use because you will be able to make informed decisions instead of guessing and assuming.

Better programming to understand what the compiler is translating your source code into that runs on the CPU. Not just understanding but also knowing how to write a more efficient program (both speed and size). And do not leave out security, which usually overrides speed and size.

The study of computer architecture and organization focuses on the interface between hardware and software and emphasizes the structure and behavior of the system.

Microprocessor & Computer Architecture (µpCA)

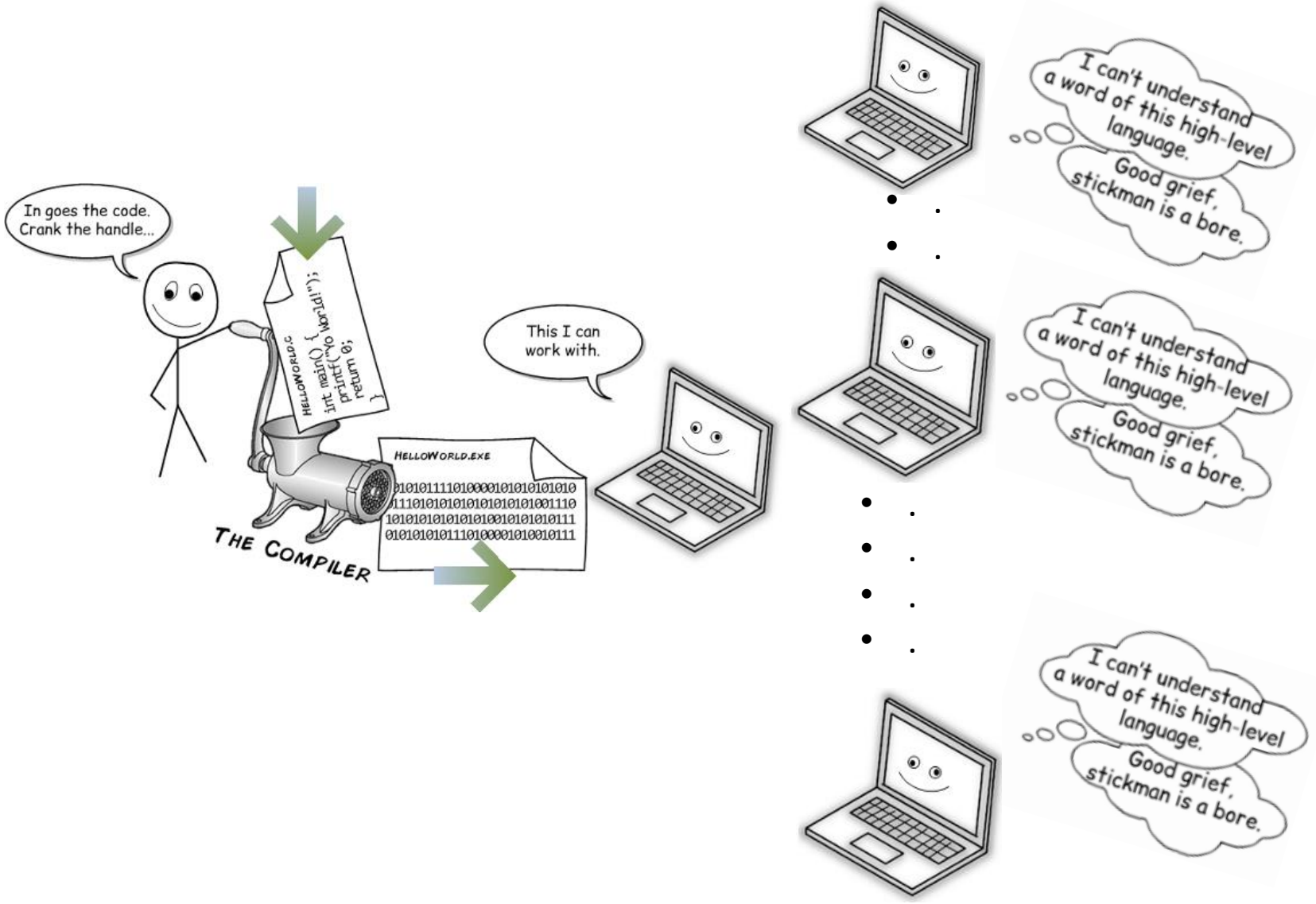
Who are all the players?



- Programmer
- Operating System
- Compiler or Interpreter
- Hardware (Processor, Memory, I/O…etc)

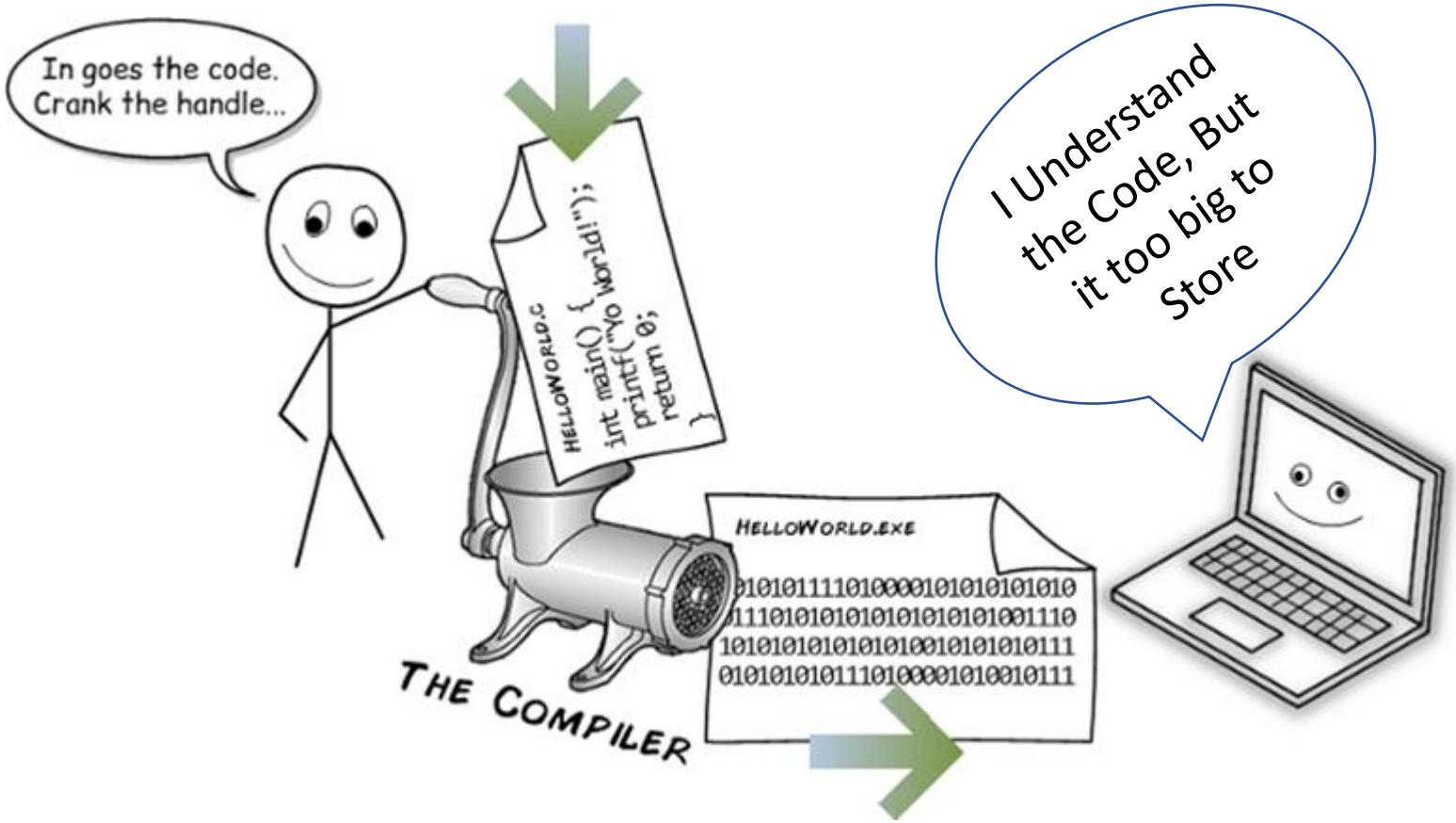
Microprocessor & Computer Architecture (µpCA)

Compiler



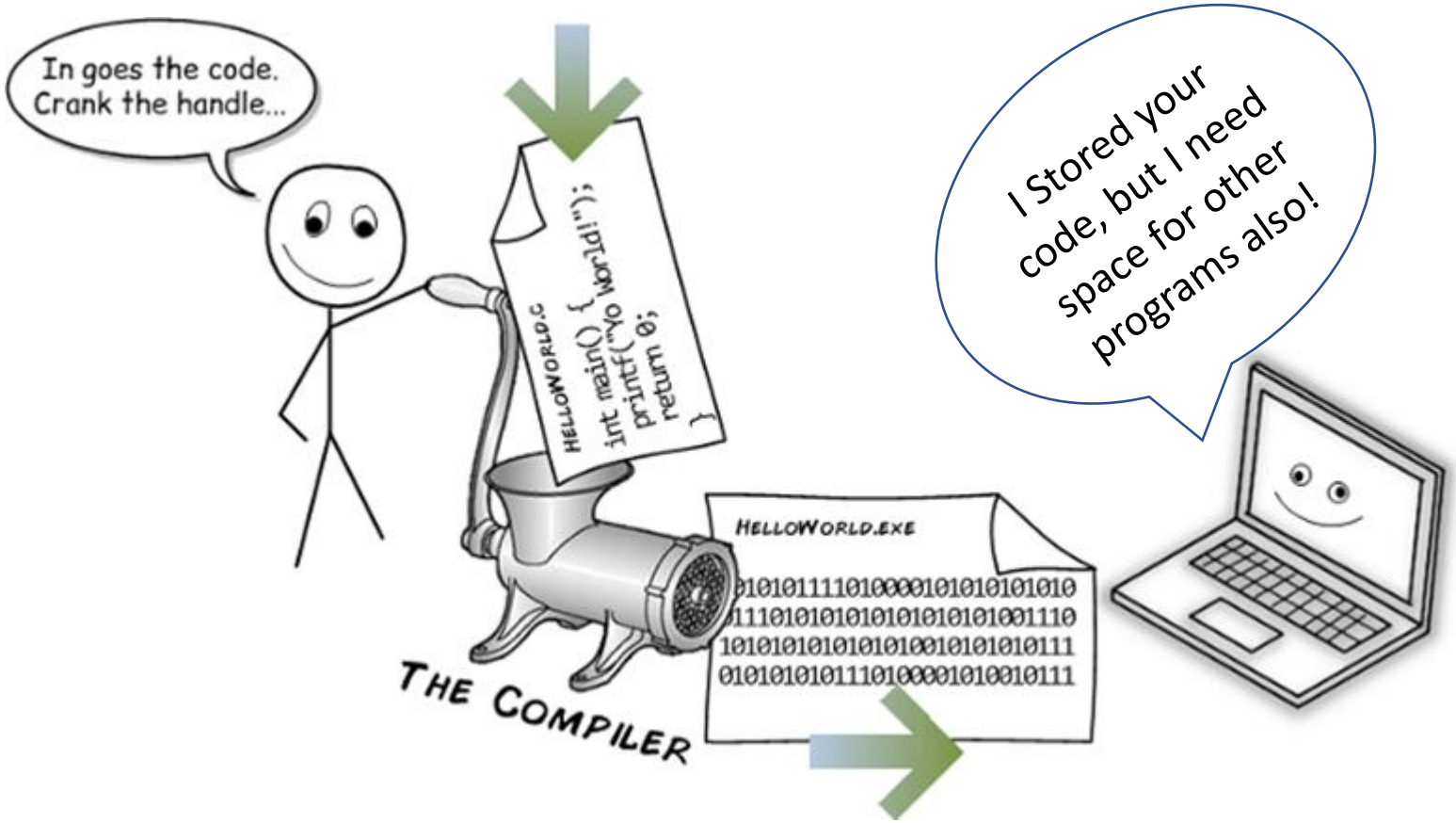
Microprocessor & Computer Architecture (µpCA)

Operating System



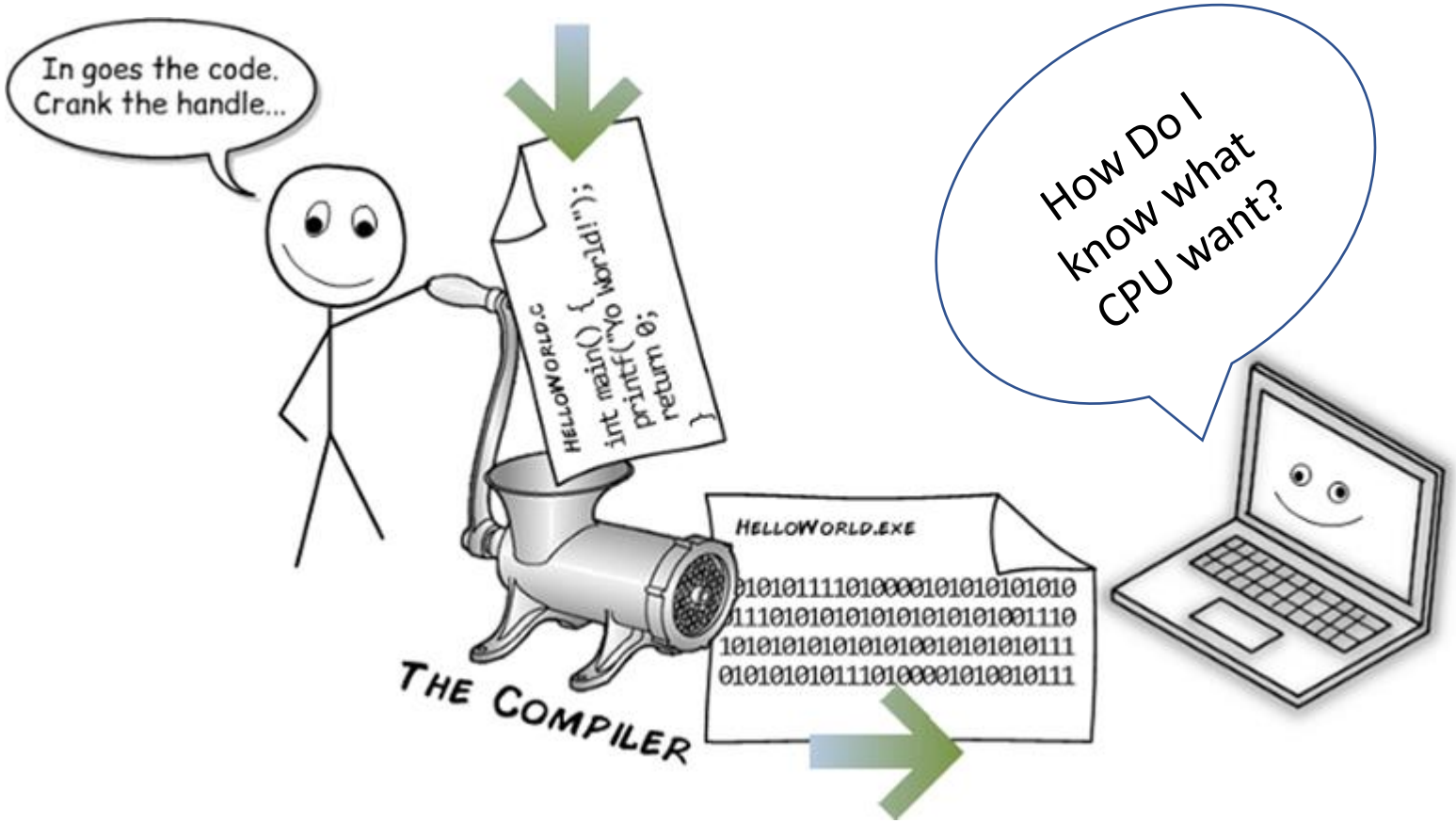
Microprocessor & Computer Architecture (µpCA)

Operating System



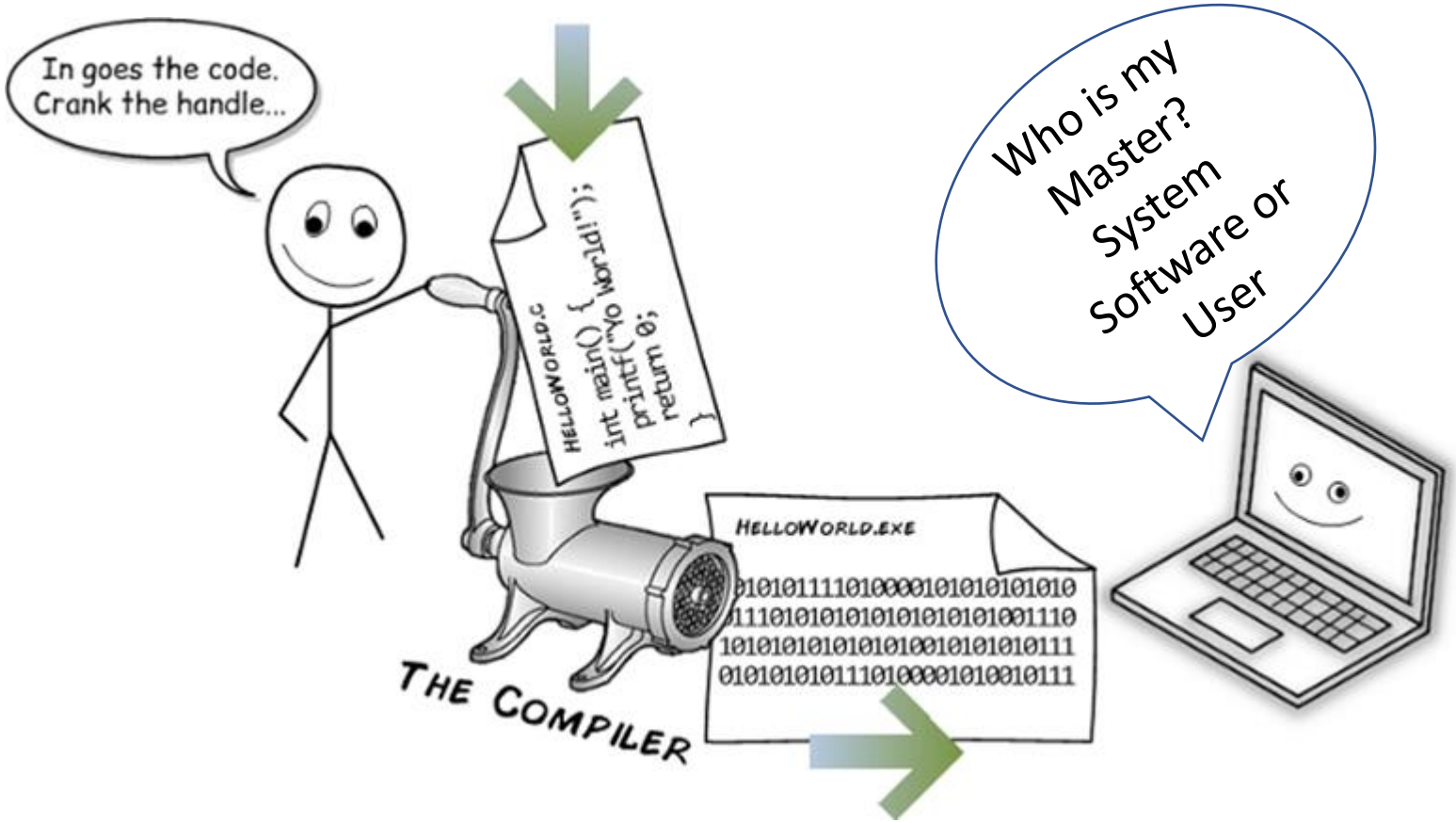
Microprocessor & Computer Architecture (µpCA)

Operating System



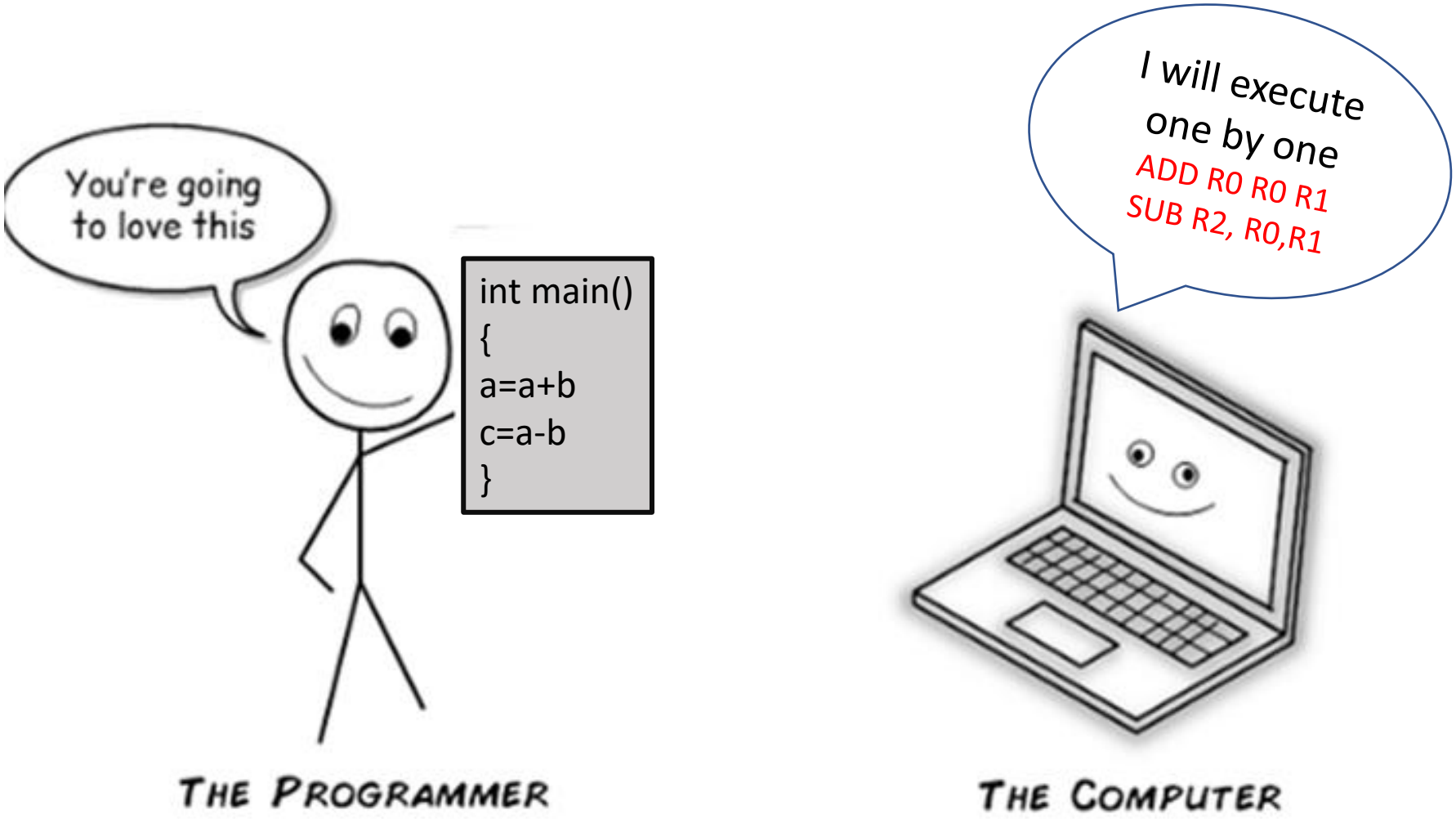
Microprocessor & Computer Architecture (µpCA)

Operating System



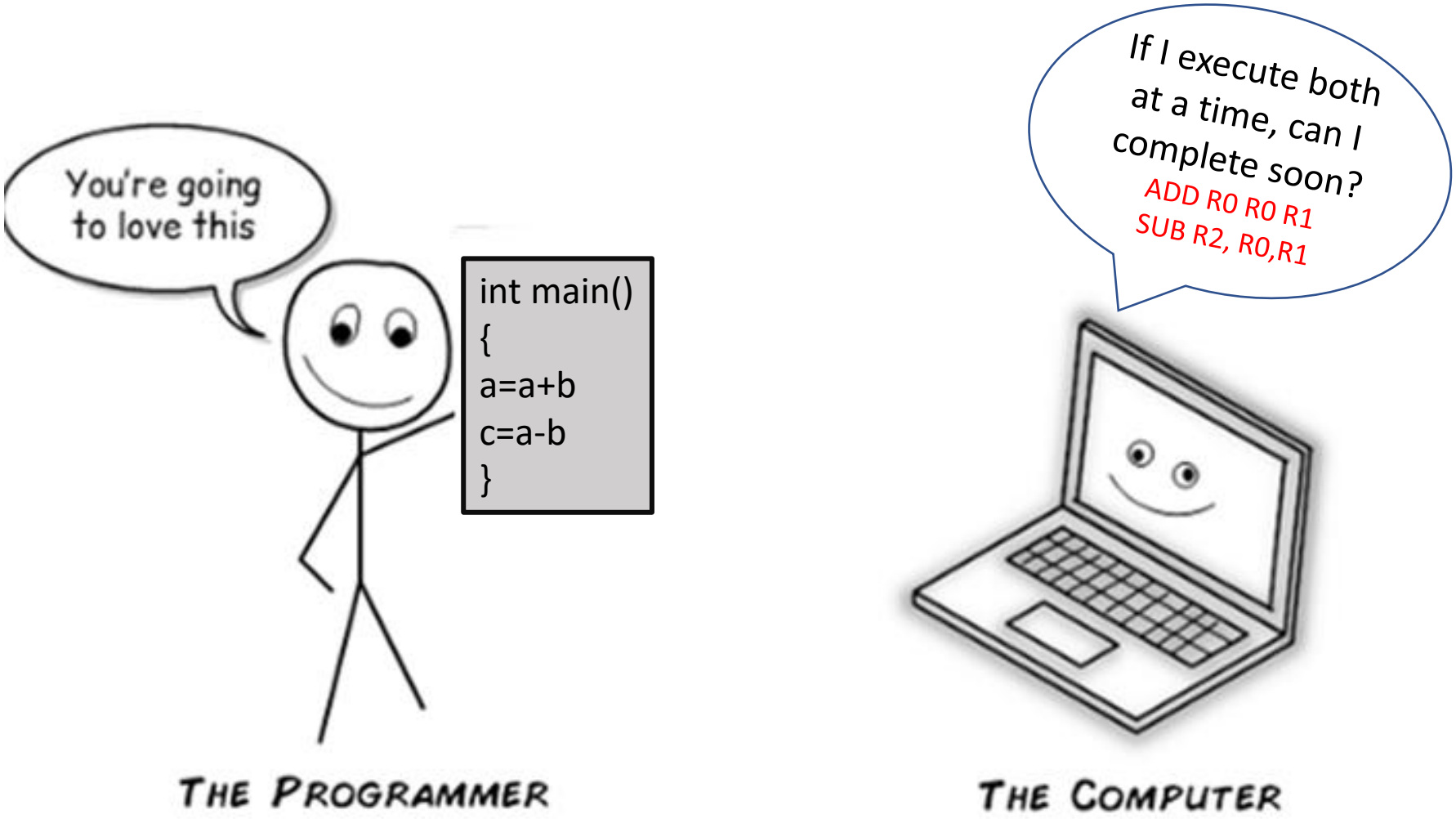
Microprocessor & Computer Architecture (µpCA)

Processor



Microprocessor & Computer Architecture (µpCA)

Processor



Microprocessor & Computer Architecture (µpCA)

What does the processor do?

Processor **Happily** spend its time computing, i.e., carrying out arithmetic operations on user data.

Misconception

Reality

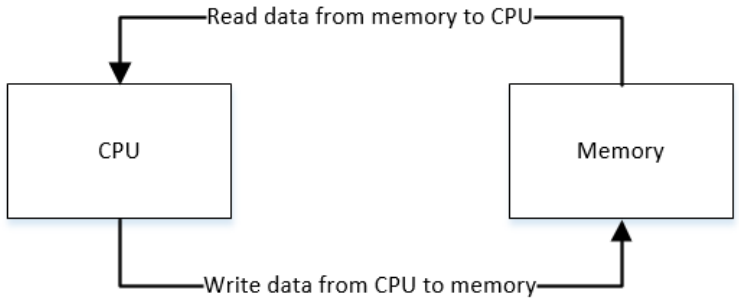
Microprocessor & Computer Architecture (µpCA)

What does the processor do?

Sample Assembly Program

```
Ldr R0, =a
Ldr R1, =b
Ldr R3, =c
Ldr R4, [r1]
Ldr R5, [r0]
Add R6, R4, R5
Str R6, [r3]
```

Instruction type	Dynamic usage
Data movement	43%
Control flow	23%
Arithmetic operations	15%
Comparisons	13%
Logical operations	5%
Other	1%



Microprocessor & Computer Architecture (µpCA)

Goal of the Course!

Learn to apply all Engineering Process to take decisions to ensure that the User Program Execute fast without keeping CPU idle

Quote

"What Processor spend their time doing, is there a way to make the processor work faster"

Microprocessor & Computer Architecture (µpCA)

Course Outcome

Unit 1: Basic Processor Architecture and Design

CO1: Demonstrate ability to understand the design of different instruction sets like RISC/ CISC and their addressing modes.

Unit 2: Pipelined Processor and Design

CO 2 &3: Design of Pipelined Processor, Performance Evaluation of Different Architecture.

Unit 3: Memory Design

CO4: Memory Hierarchy, Performance Evaluation and Optimizations.

Unit 4: Input/Output Device Design

CO4: Design of I/O devices, Performance Evaluation.

Unit 5: Advanced Architecture

CO5: Demonstrate and appreciate modern trends in architecture such as multicore architectures.

Microprocessor & Computer Architecture (µpCA)

Job Description 1

Job Description

Software Developer for Microprocessor based embedded system

- Perform Board Bring-up on ARM/Intel SoC architecture.
- Reverse engineer HW-SW issues by referring schematics & data sheet.
- Realize digital design on Intel/Xilinx FPGA.
- Optimize device drivers (Ethernet, PCIe, DMA, CAN) taking into consideration security aspects.
- Migrate software components (Device Drivers & Middleware) from Linux to QNX.
- Design systems using SoC, PCIe Switch, FPGA & SoC's.
- Design software architecture using state of the art tools (like Rapsody).
- Port open source software & benchmark performance on development kits.
- Positively influence Bosch position by contributing to open source software.
- Create quick demonstrators/PoC & represent Bosch offerings in trade shows

Qualifications

- Qualification: B.E./B.Tech./ M.E./ M.Tech. – Electronics & Communication or Computer Science
- Work Experience: 1 to 4 years

Technical Know-how:

- Background experience in using Digital Storage Oscilloscope.
- Awareness of Intel and ARM SoC architecture from secure boot point of view.
- Awareness of ARM Instruction Set Architecture & Vectorization techniques.
- Practical experience in real-time Linux/ RTOS based embedded software development using C & C++.
- Awareness of latest technical happenings in open source software (OSADL, Linux mainline, Autoware, Apollo, Adaptive AUTOSAR).
- Practical experience in debugging software for embedded systems.
- Practical overview (via. academic literatures) of technology trends in SoC, Ethernet Switch, PCIe & FPGAs.

Microprocessor & Computer Architecture (µpCA)

Job Description 2

NVIDIA is searching for a highly motivated, creative engineer with experience in system software to join the GPU Software team. As someone who is hardworking and passionate about their work, you will design key aspects of our production GPU kernel drivers and embedded Software. You should demonstrate the ability to excel in an environment with complex software and hardware designs.

What you'll be doing:

- Define, design, develop and verify features for our GPUs; collaborating with hardware engineers and fellow software engineers
- You will follow the devices all the way through the development process to the customer desktops, notebooks, workstations, and gaming console products that are used throughout the world
- Heavily involved with the early modeling and simulation required to produce our world-class products
- Multiple opportunities to collaborate and communicate effectively with teams from all around the globe

What we need to see:

- BS or MS degree in Computer Engineering, Computer Science, or related degree
- Strong C programming skills as well as having shown initiative in pursuing independent coding projects
- Familiarity with computer system architecture, microprocessor, and microcontroller fundamentals (caches, buses, memory controllers, DMA, etc.)
- Kernel experience with Linux, Android, Chrome, or Windows systems
- 2+ years of meaningful software development experience

Ways to stand out from the crowd:

- Background and strength with complex system-level debugging is invaluable
- Deep understanding of memory management and virtualization concepts
- Familiarity with kernel level security concepts
- Experience with embedded system SW concepts, e.g.: RTOS and overlay programming models

Microprocessor & Computer Architecture (µpCA)

Job Description 3

Capgemini
Firmware Developer

Role Description (Role & Responsibilities)

1. **Mandatory skills 16bit 32bit Microcontroller Microprocessor**
2. Embedded systems Firmware Device driver development experience Programming Strong in C
3. Communication protocols UART CAN SPI Ethernet Modbus TCP IP IDE usage
4. Code composer studio IAR workbench Code warrior RTOS VxWorks FreeRTOS Ti RTOS Bootloader
5. Multi threading concepts Preferred skills Strong in Cplusplus Assembly language experience

Mobiveil: CPU Processor Design

Job Summary:
Bachelors or Masters degree in **Computer Science** or Electrical/Computer Engineering.

Understanding of general purpose CPU micro architecture, including knowledge of areas such as processor pipelines, caches, memory hierarchy, and multi-processor systems.

Knowledge and or Experience in RTL Design hardware development using Verilog, ideally block design in a CPU design project or similar high performance project.

Understanding of CPU instruction set architecture and assembly language.

Familiarity with ARM architecture and micro-architecture for current ARM CPU cores is helpful but not required.

Software development skills and/ or experience is helpful (C/C++, Python/Perl, Shell scripting)
Experience modelling microprocessors using higher-level languages, like C/C++, is helpful but not required
Effective communication skills and the ability to collaborate with a team

Microprocessor & Computer Architecture (µpCA)

Job Description 3

Open-Silicon: **Software Engineer**

Job Description:

- Solid programming experience in C or C++.
- Adequate knowledge of Object oriented programming and design concepts.
- **Should have good knowledge of System software, Microprocessors/Microcontrollers, Memory subsystem, Hardware IPs .**
- Proven experience in embedded systems design with low level driver programming , bootloader, preemptive, multitasking real-time operating systems
- Should have exposure to scripting language e.g. Perl/Shell/Python etc.
- Experience in hands-on development and troubleshooting on embedded boards.
- Familiarity with software configuration management tools, defect tracking tools, and peer review
  Good team player, should be able to handle task independently

Next Class

- What is Microprocessor
- Why Study Microprocessor
- Evolution of Microprocessor
- Classification of Processor: CISC vs RISC

THANK YOU

Dr. D. C. Kiran
Department of Computer Science and Engineering

dckiran@pes.edu

9829935135