



Software Development

IEEE Northern Virginia Section

Hands-On Professional Development Series

October 29, 2016 Montgomery College

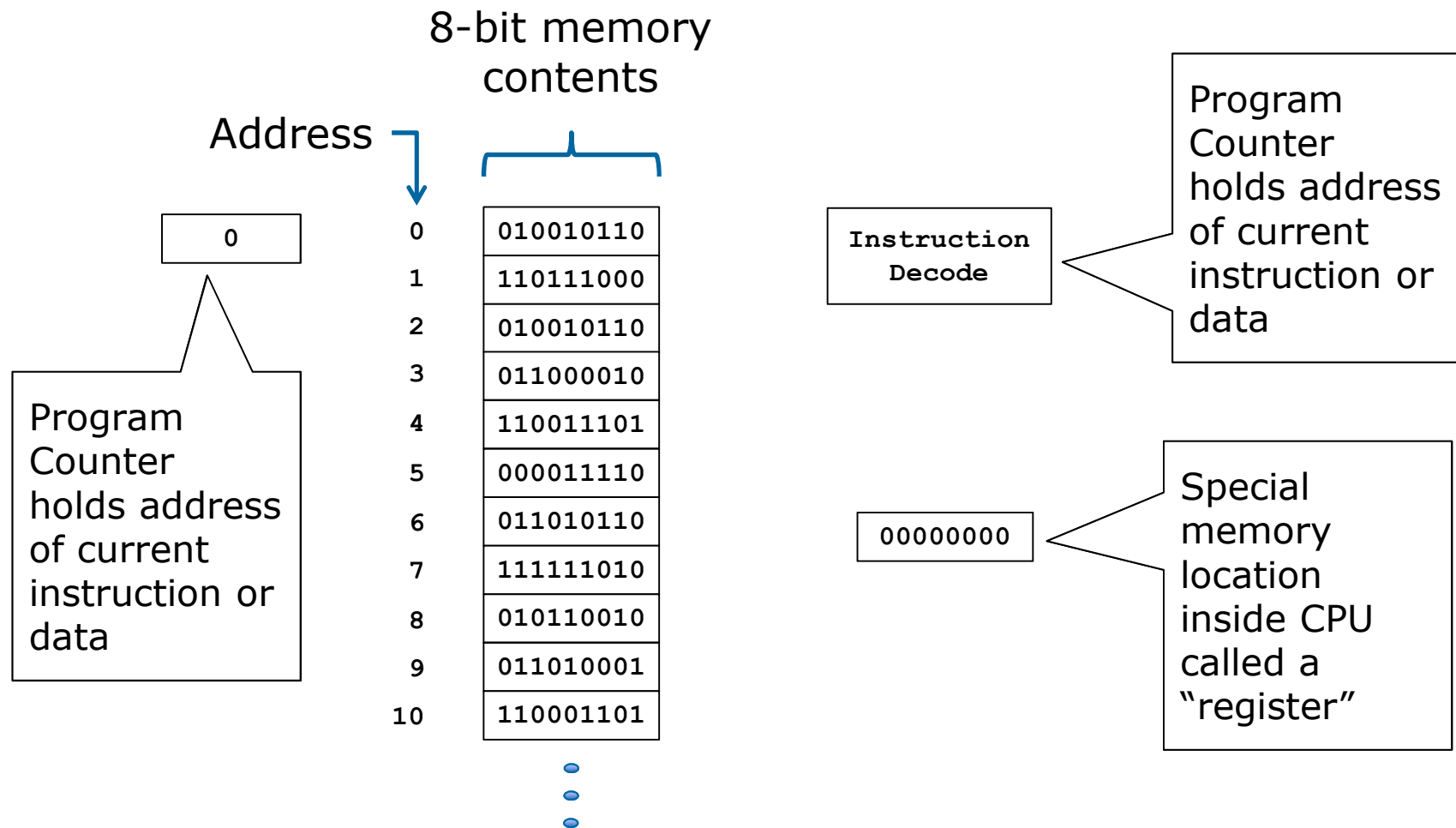
About These Slides

- ▶ This course is not about software
 - We use and modify it, but don't "develop" it
 - You can after the course if you want...
- ▶ We do however use software tools
 - In particular we use ones for Arduino
 - So we should know a little about it
- ▶ These slides give a very high level introduction
 - Software Development goes much, much deeper
 - Let's start with the hardware: microprocessors

How Processors Work

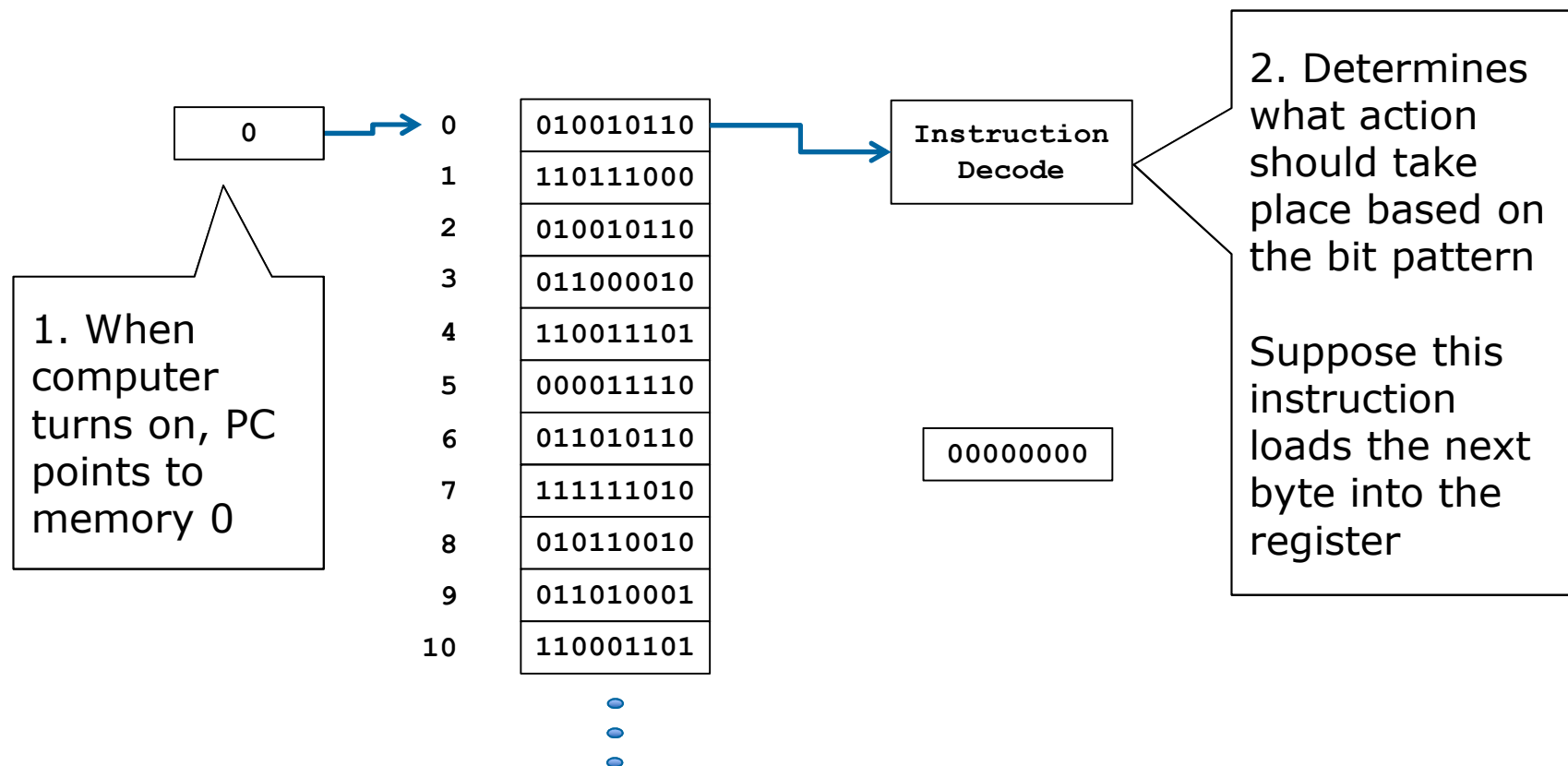
- ▶ Computers are sometimes said to work using numbers
 - More precisely, they work with binary digits (bits) usually in groups of 8 (bytes), 16 (words), 32, and 64
 - They are stored in memory and accessed by address
- ▶ Each group can represent instructions, data, or both
 - In Harvard architectures instructions and data are stored in separate spaces; Von Neumann machines share
 - Modern CPUs are somewhere in between
 - A typical processor reads one number at a time and decodes instruction bits to determine how to process the data

Hypothetical 8 Bit CPU Example

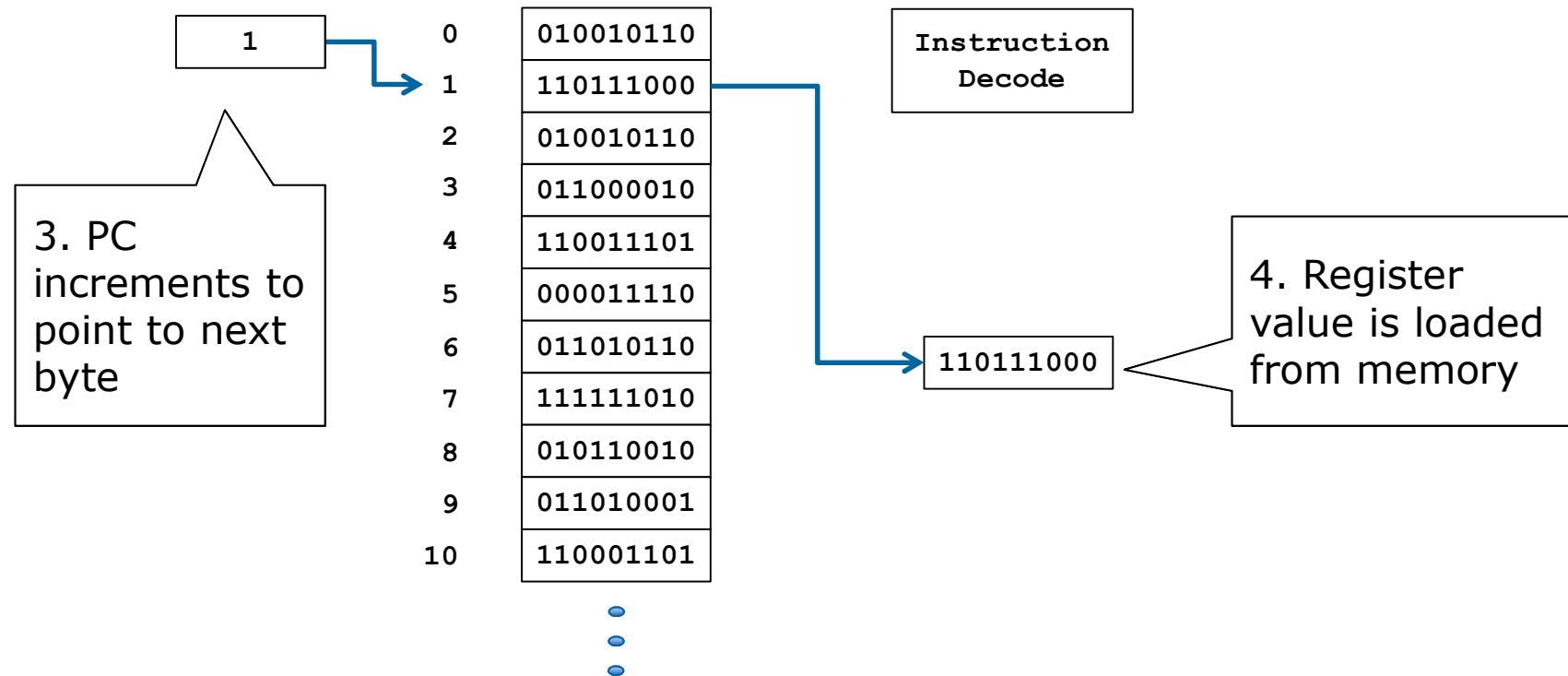


Up to 65,536 addressable locations

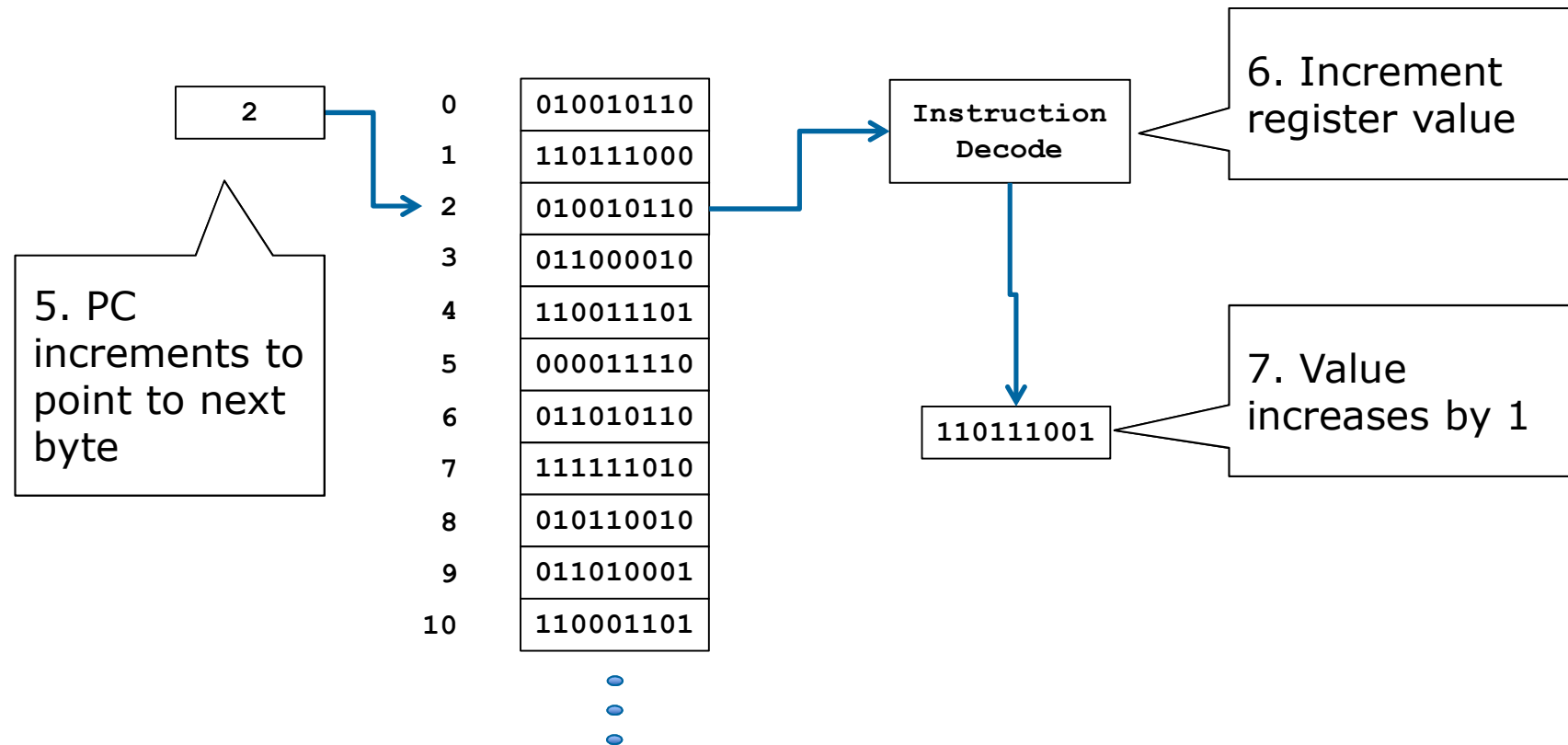
Hypothetical 8 Bit CPU Example



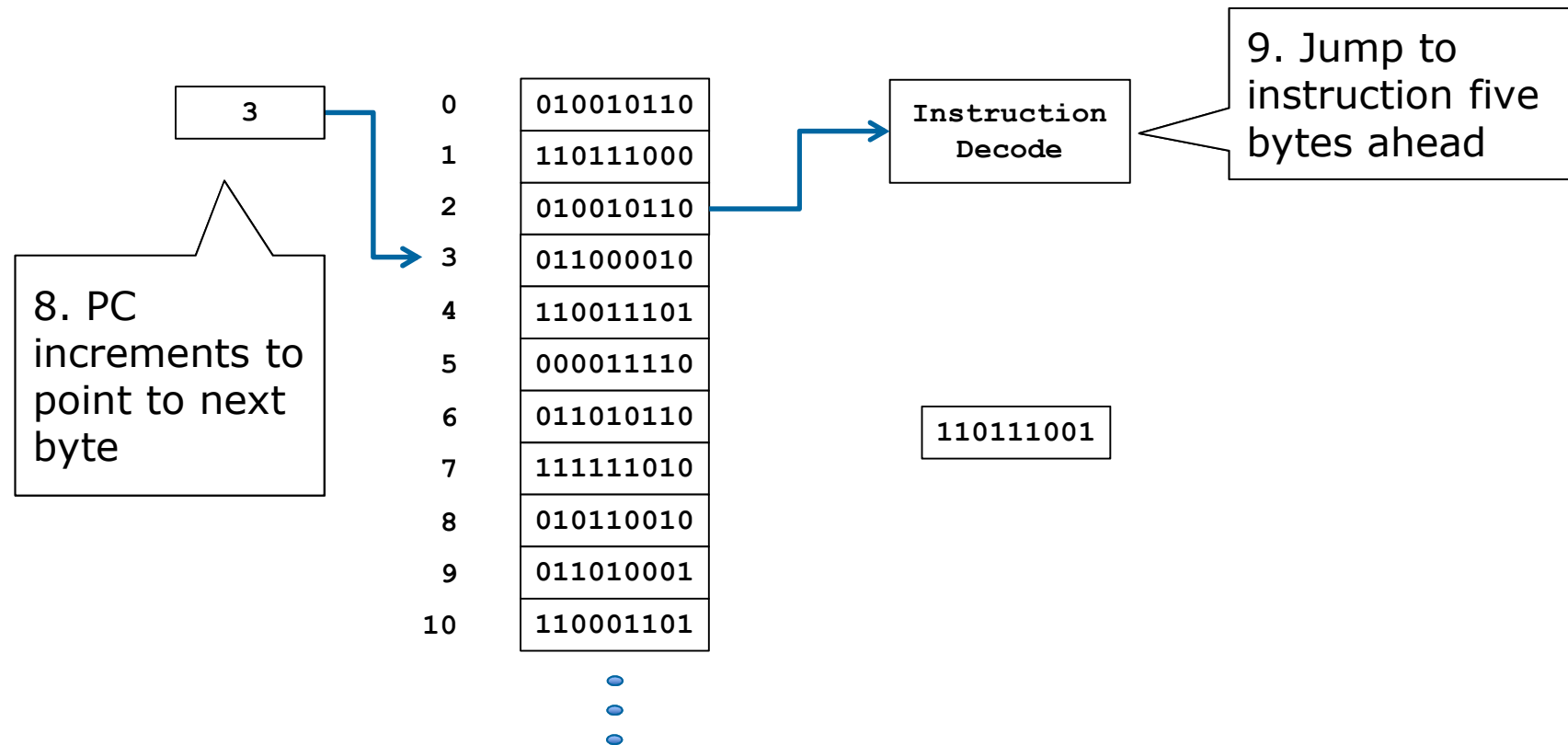
Hypothetical 8 Bit CPU Example



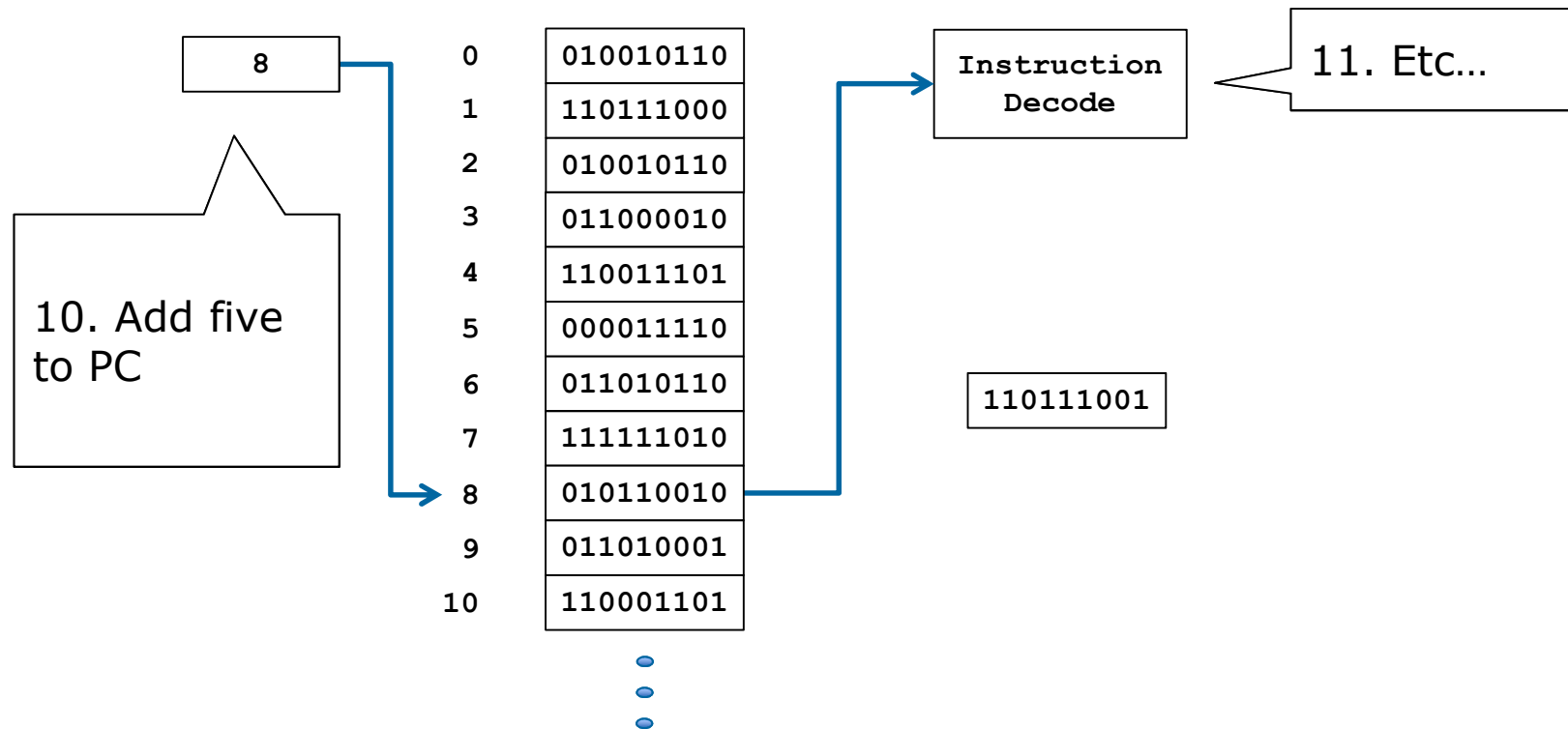
Hypothetical 8 Bit CPU Example



Hypothetical 8 Bit CPU Example

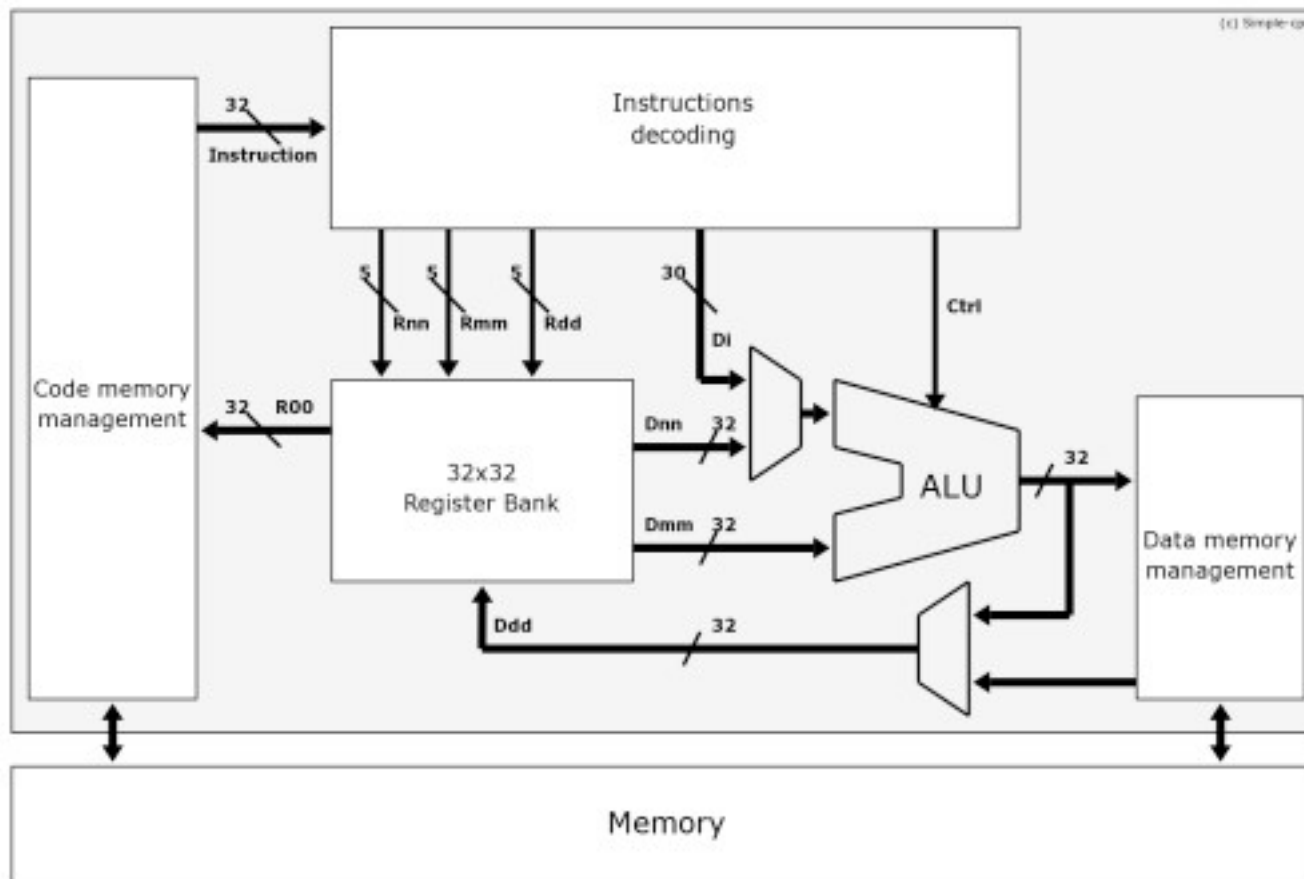


Hypothetical 8 Bit CPU Example



Example Processor Architecture

<http://www.simple-cpu.com/cpu-instruction-set-architecture-en.php>



Machine Instructions

- Most processors have a range of instruction types
 - Data movement
 - Arithmetic and logical instructions
 - Flow control
 - Specialized processing for encryption, video
- They also add many performance-enhancements
 - Instruction and data caches
 - Instruction pipelining
 - Multiple cores
- x86, ARM are common, each with variants

Machine Instructions

- ▶ Most of what your computer, table, and phone do
 - Word processing
 - Web and email
 - Video streaming

Happens as a series of these primitive operations

- Early processors did less than 1 million primitive per second
- Modern CPU's can perform several billion per second
 - Clock speeds are increasing much more slowly now than in the several decades

Assembly Language

- ▶ It is too difficult to develop complex software at the bit level
 - Not a good match for human cognition
- ▶ Assembly language creates a mapping from mnemonics to binary code for a CPU
 - An “Assembler” compiles it to binary
 - Often used for drivers, high-performance routines

```
LDA #<VTOC
STA A1
LDA #>VTOC
STA A1+1
LDA #<END
STA A2
LDA #>END
STA A2+1
LDA #$00
STA A4
LDA #$B5
STA A4+1
SEC
JMP AUXMOVE
```

Operating Systems

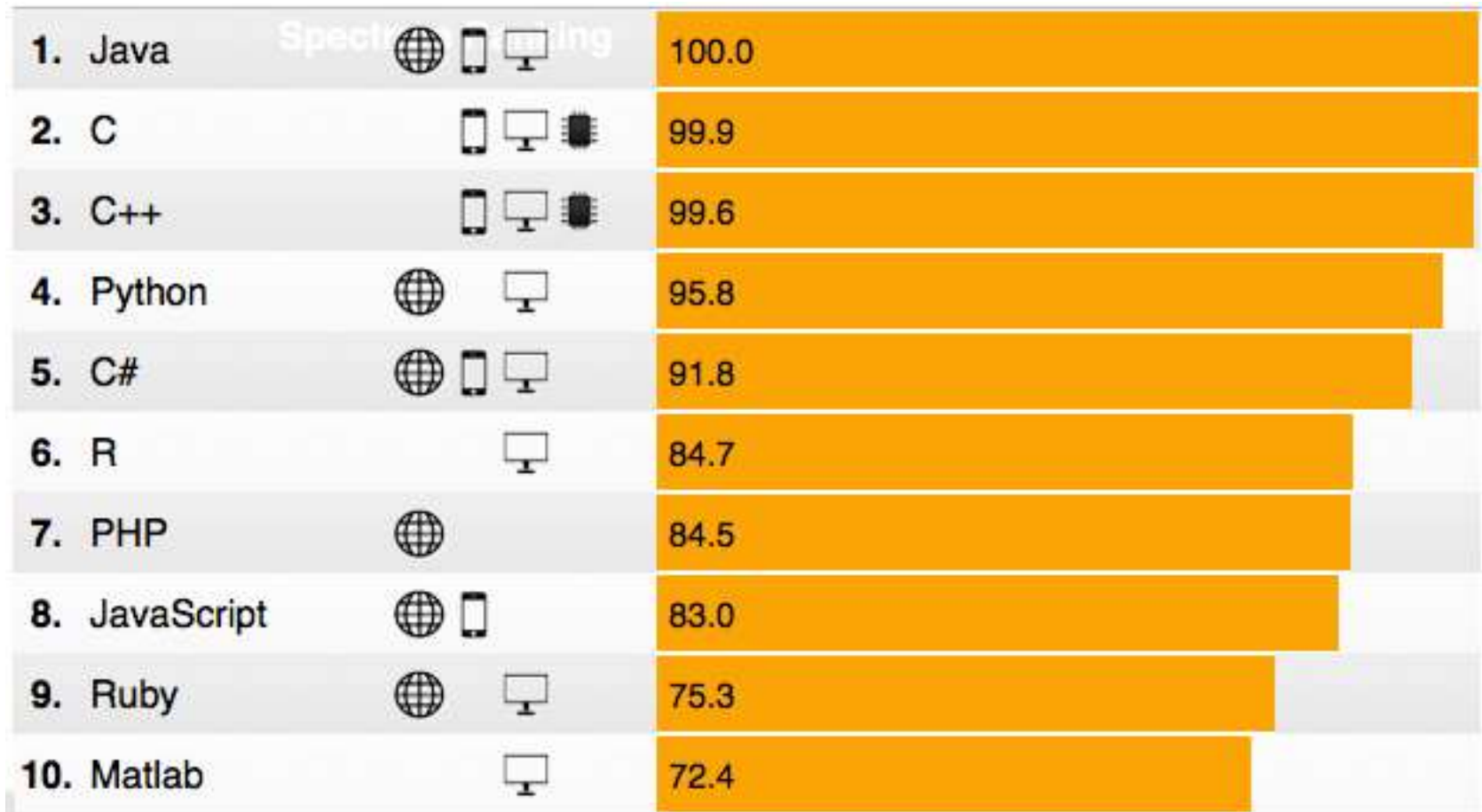
- Most programs share the need common functions
 - Memory management
 - Input/Output (disk, keyboard/video, network)
 - Multitasking
 - Though not for all embedded systems
- The Operating System provides a software environment in which to run individual programs
 - Individual programs don't need to reinvent the wheel
 - Makes hardware and formats compatible across programs
 - But OS's are too complex to write in assembly...

Programming Languages

- ▶ Programming languages allow software developers to work at a higher level of abstraction
 - Increasing productivity while reducing errors
 - Provide portability across hardware and OS's
- ▶ There are many from which to choose
 - Classification and terminology is a class unto itself
 - Some are domain specific – not all appropriate for embedded
- ▶ IEEE tracks the relative popularity of many

<http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2015>

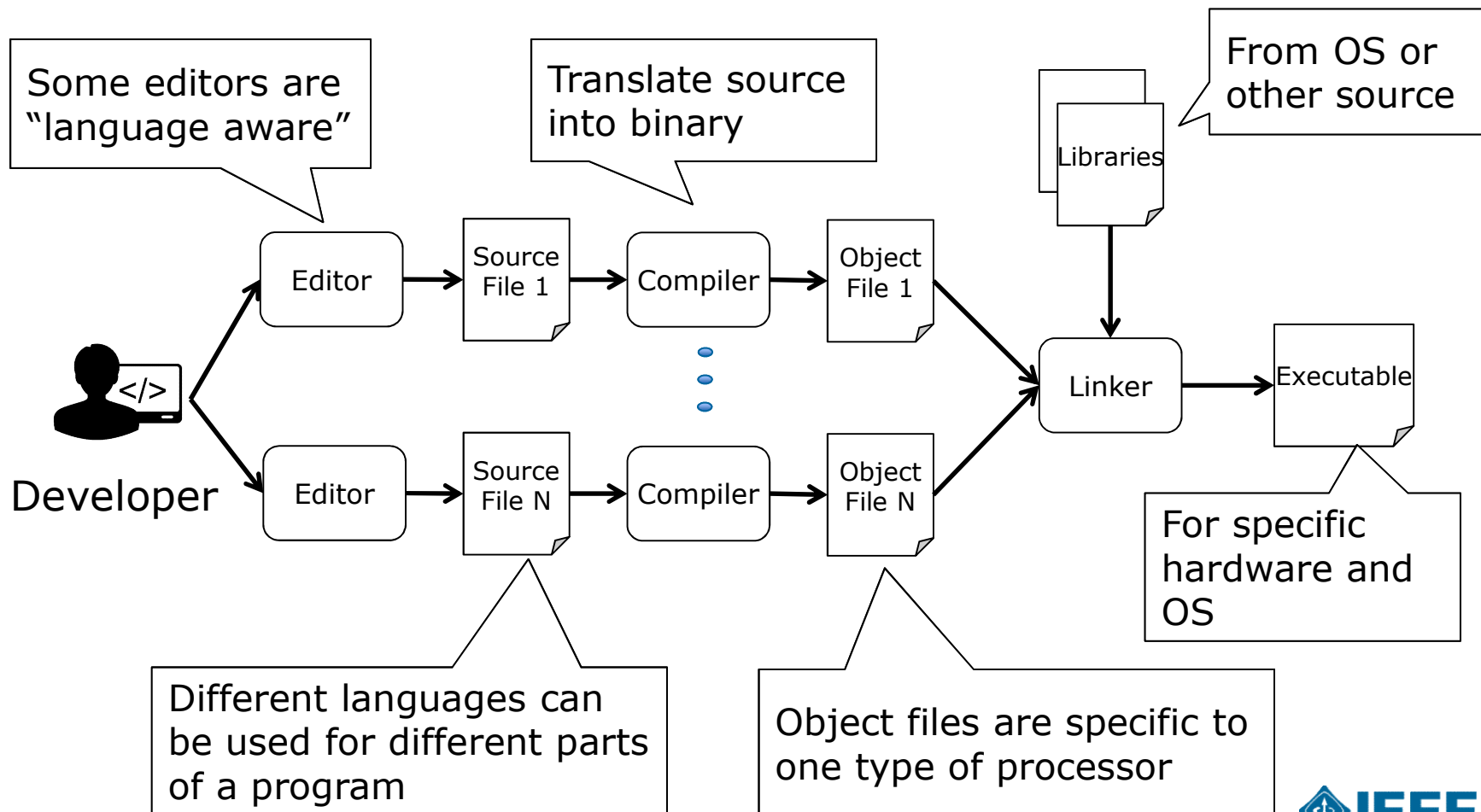
Top Ten Languages



Software Design

- Software development usually begins with requirements
 - Commercial software requirements can be extensive
 - Software programs to manage requirements exist
 - Or it can be an idea in someone's head
- What needs to be done is mapped to an architecture
 - A “good” architecture takes development resources, security, and future plans into account
 - The architecture is refined into smaller pieces
 - Languages, tools, libraries are chosen
 - Eventually coding begins

Development Tools



Debugging and Version Control

- Software doesn't always work the first time...
 - Debuggers provide diagnostic output as the program executes to help find errors
 - Some can “single step” a program, slowing down the processor's execution
- ...so developers change the source code
 - Version (or Source) Control systems allow developers to save their source code and undo/roll back changes
 - They can also coordinate changes when multiple developers work on the same project

Integrated Development Environments

- IDEs combine multiple functions into a single program
 - Typically language-aware editor, compiler/linker, debugger
 - Sometimes emulators, hardware loaders, hooks to version control, etc.
- Some companies offer proprietary IDEs for their CPUs
 - There are also cloud/web-based systems
- Many have switched to the popular Eclipse package
 - <https://eclipse.org/ide/>

Arduino Defined

- ▶ Arduino is an “eco-system” of related pieces
 - A set of low-cost embeddable processor boards
 - Some are “official” while others are just “compatible”
 - Various I/O and “shields” that add more
 - Different boards use different processor architectures
 - A programming language tailored to the above
 - Easy to use
 - Based on Processing and similar to C and C++
 - An IDE simplifying use of the language on the boards
 - That’s what we’ll be using
- ▶ Arduino.cc is also a company that supports the above

Arduino IDE

- ▶ A program in Arduino is called a "Sketch"
 - It is a high level language similar to C and C++
- ▶ It supports two styles of comments
 - `/* This is one type */`
 - `// This to end of line is another`
- ▶ It includes many intuitive functions for I/O
 - You can also define your own
 - Every program must define `setup()` and `loop()`

Weather Station Choice

- It's not the only way to achieve today's goal
 - But it may be the easiest!
- It's also a useful thing to learn
 - You can buy Arduino boards at Microcenter and online
 - There are many, many people using them
 - The concepts apply to many other languages and IDEs

Software Development Processes

- Software development is iterative
- There are many different approaches to create good quality software in predictable time
 - Waterfall
 - Spiral
 - Rapid Application Development
 - Agile
- The list goes on and on...
 - and it can get somewhat religious

Software Certifications

- ▶ If you're interested in learning more
 - Software development courses
 - Books
 - Free online emulators
 - IEEE Computer Society Certifications

<https://www.computer.org/web/education/certifications>

Now let's start putting things together!

Questions?