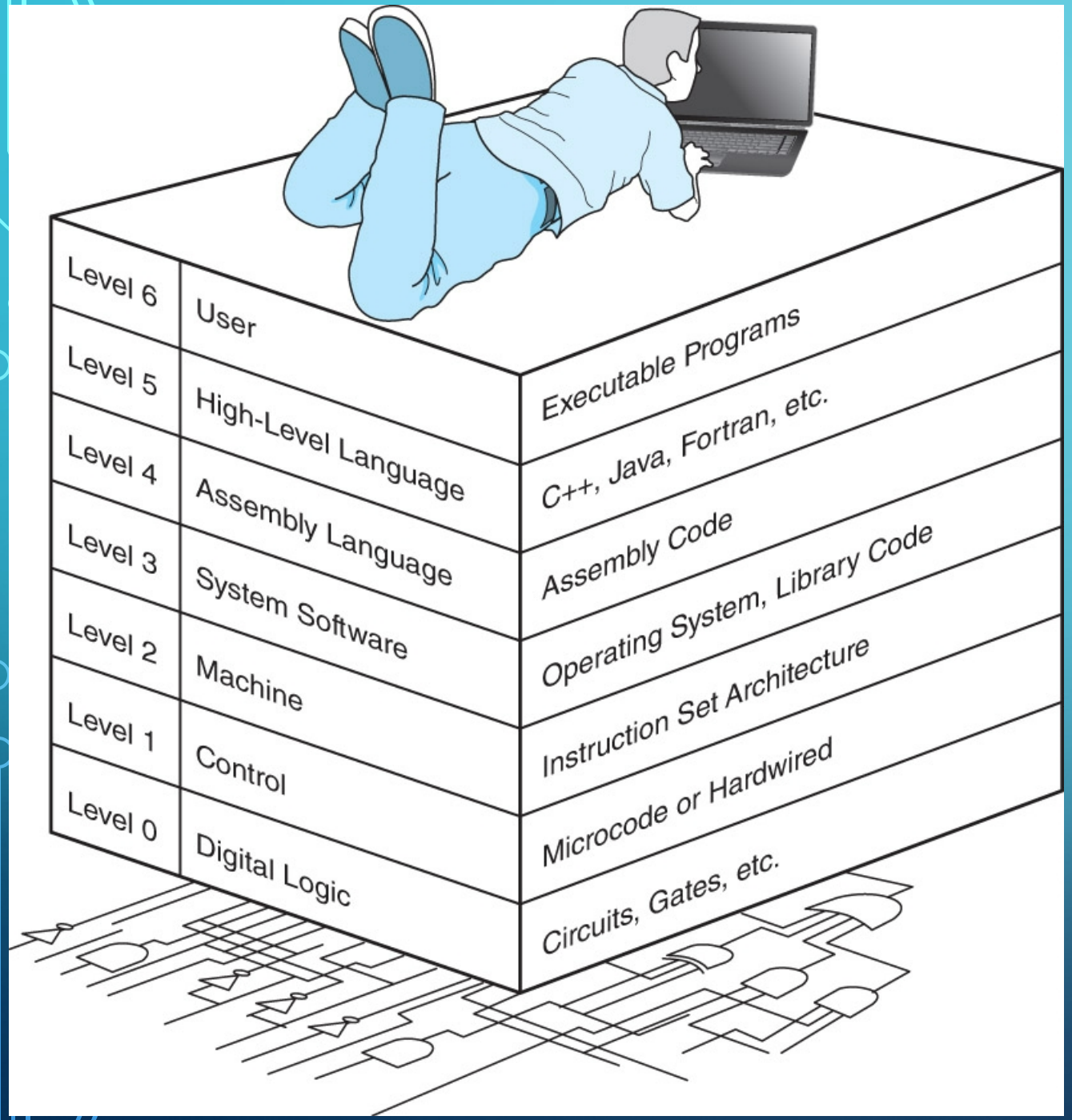
COMPUTER ARCHITECTURE

HARVARD AND VON NEUMANN

ABSTRACTION OF COMPUTING



OVERVIEW

- Architecture (Machine organization) overview
- CPU
- Harvard
- von Neumann
- CISC *vs*. RISC

COMPUTER COMPONENTS

- Major components of a computer
  - Central Processing Unit (CPU)
  - memory
  - Bus
    - to move information from one component to another
    - divided into three sub-buses, one each for data, addresses and control signals
  - Peripheral devices

CPU

- The Central Processing Unit (CPU)
  - Often referred to as the “brain” of the computer.
  - Responsible for controlling all activities of the computer system.
  - CPU are mounted is mounted on a **Motherboard** along with most of other electronics
- Features
  - performs arithmetic and logical operations
  - synchronous operation

CPU

- The three major components of the CPU are:
  1. **Arithmetic Unit** (Computations performed)

     Accumulator (Results of computations kept here)
  2. **Control Unit** (Has two locations where numbers are kept)

     Instruction Register (Instruction placed here for analysis)

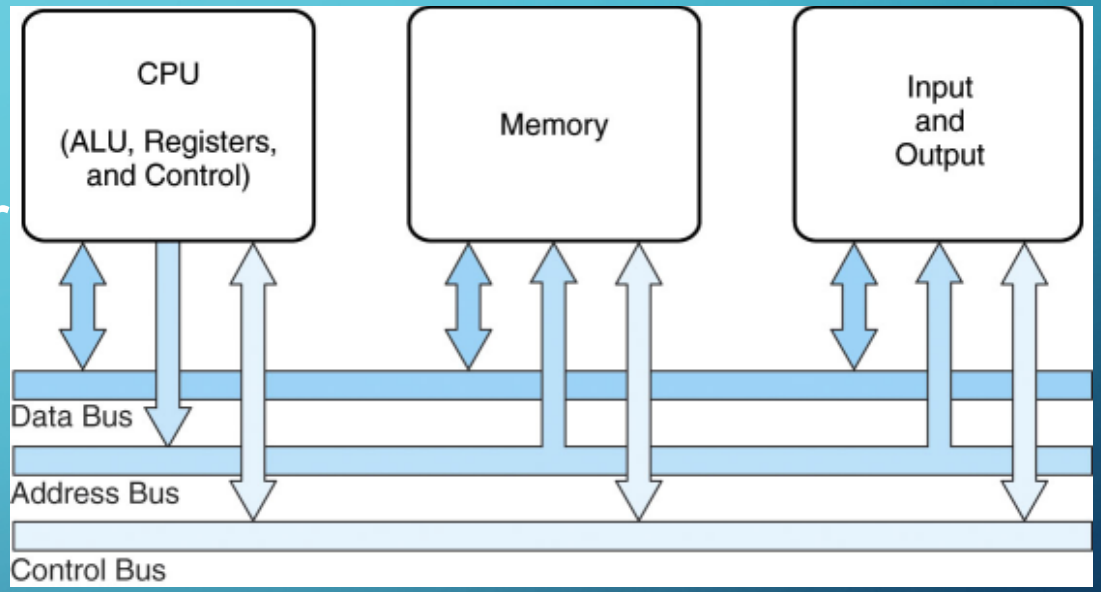
     Program Counter (Which instruction will be performed next?)

     General-purpose registers, etc.
  3. **Instruction Decoding Unit** (Decodes the instruction)

MEMORY

- Memory
  - stores programs and data
  - organized as
    - bit
    - byte = 8 bits (smallest addressable location)
    - word = 4 bytes (typically; machine dependent)
  - instructions consist of operation codes and addresses

COMPUTER COMPONENTS

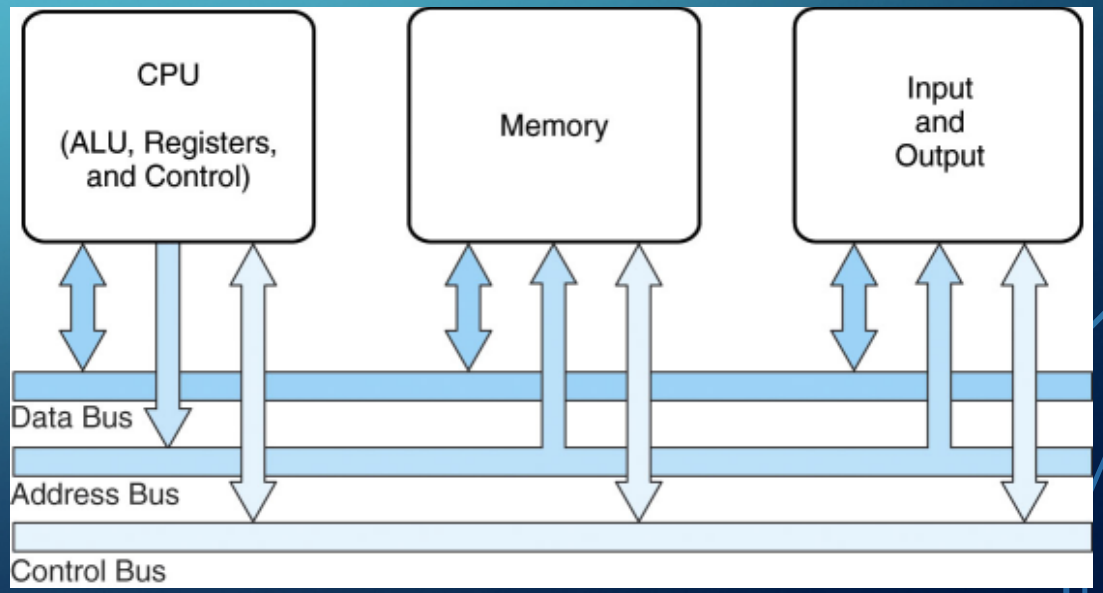


- Major components of a computer
  - Central Processing Unit (CPU)
  - memory
  - Bus or I/O System
    - to move information from one component to another
    - divided into three sub-buses, one each for data, addresses and control signals
  - *Peripheral devices e.g. secondary storage

- v

BUSES

- Function
  - to move information from one component to another
  - divided into three sub-buses, one each for data, addresses and control signals
- Types
  - Address bus
  - Data bus
  - Control Bus



COMPUTER ARCHITECTURE

- Architecture is concerned with
  - internal structures of each
  - Interconnections - speed and width
  - relative speeds of components
- Balance is often critical execution speed and connections

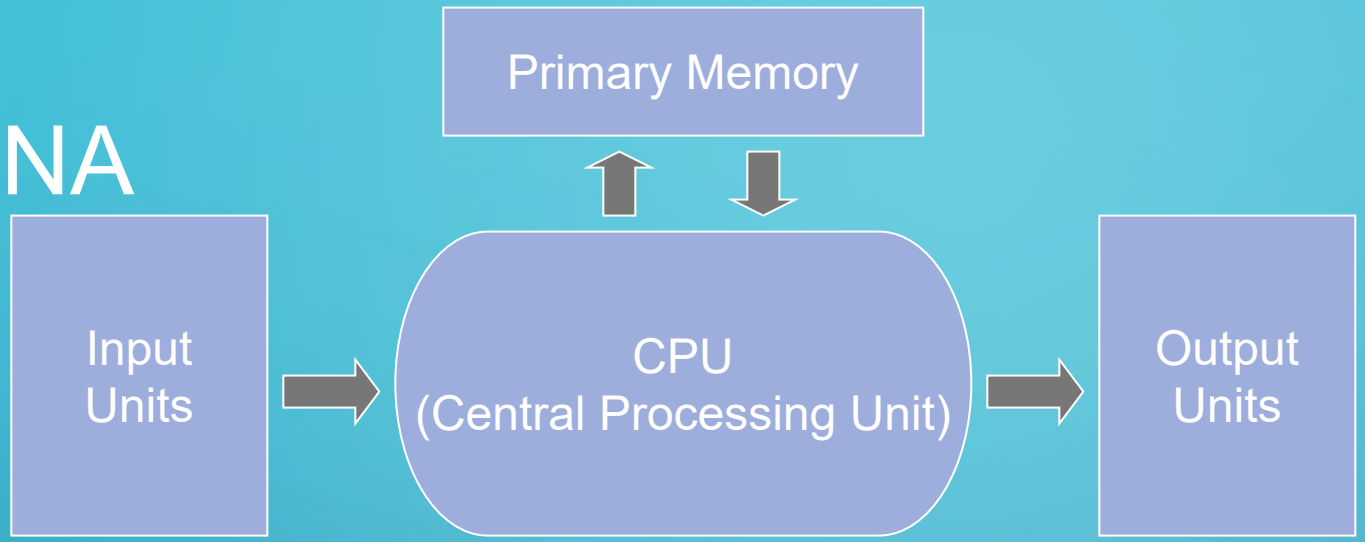
VON NEUMANN

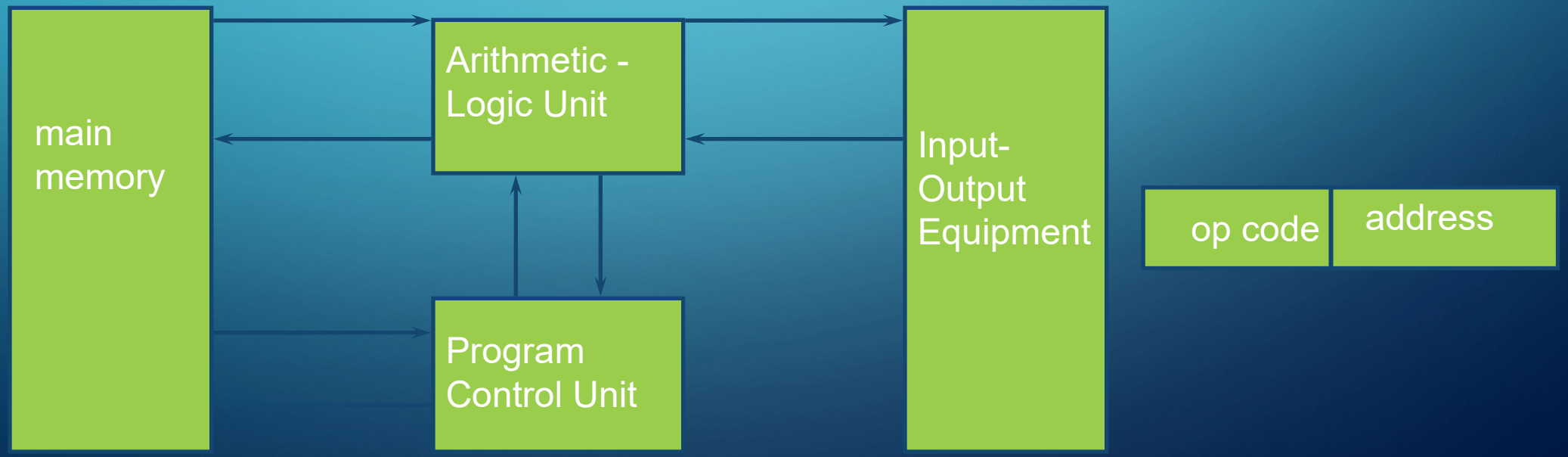
- Institute for Advanced Studies machine (1947)“von Neumann machine”
- This model of the typical digital computer is often called the **von Neumann** computer.
  - Programs and data are stored in the same memory: **primary memory.**
  - The computer can only perform one instruction at a time.
- Separate CPU and memory distinguishes programmable computer.
- Memory holds data, instructions.
- Central processing unit (CPU) fetches instructions from memory.
  - ALU performs transfers between memory and I/O devices
  - note two instructions per memory word

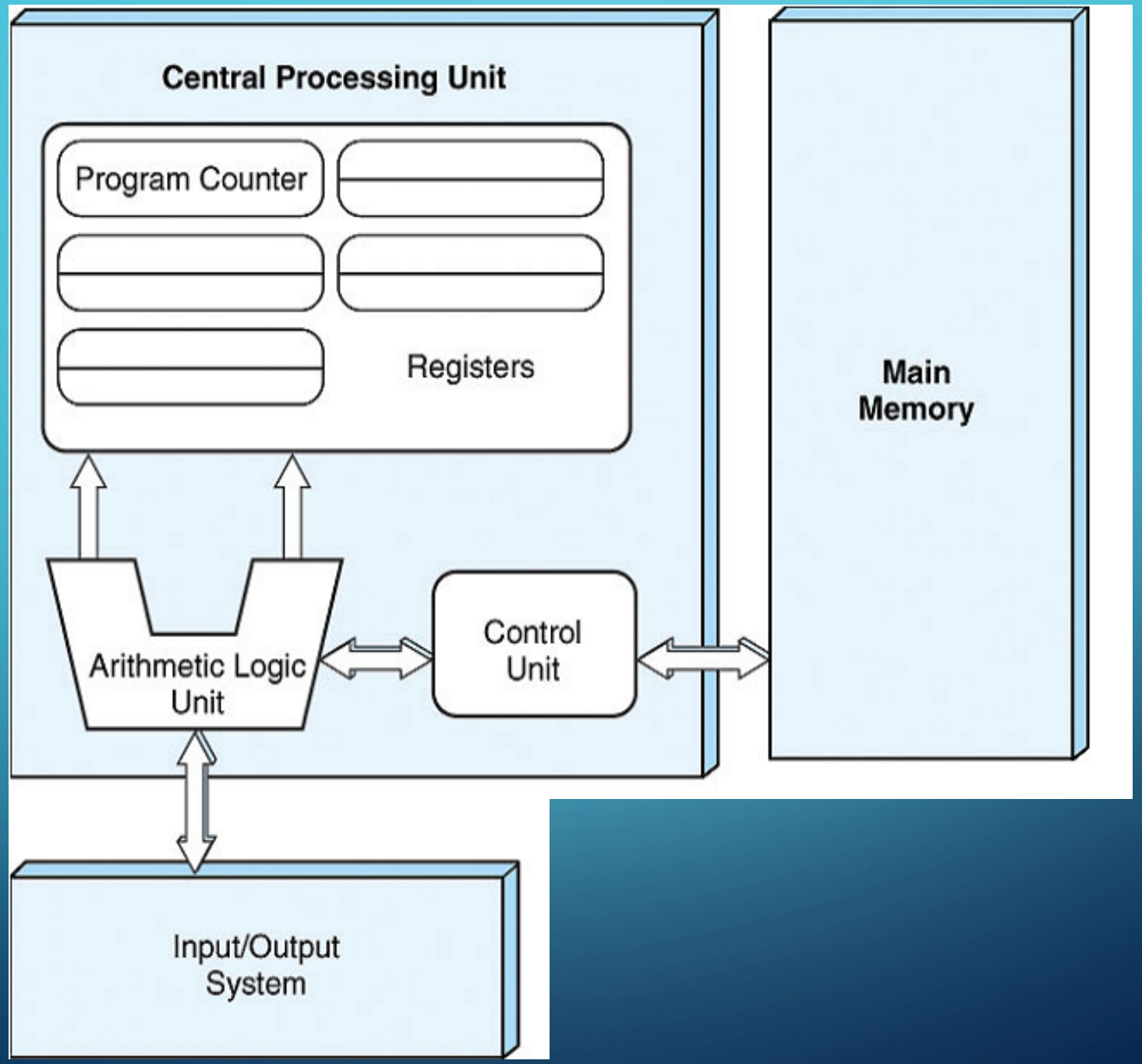
VON NEUMANN

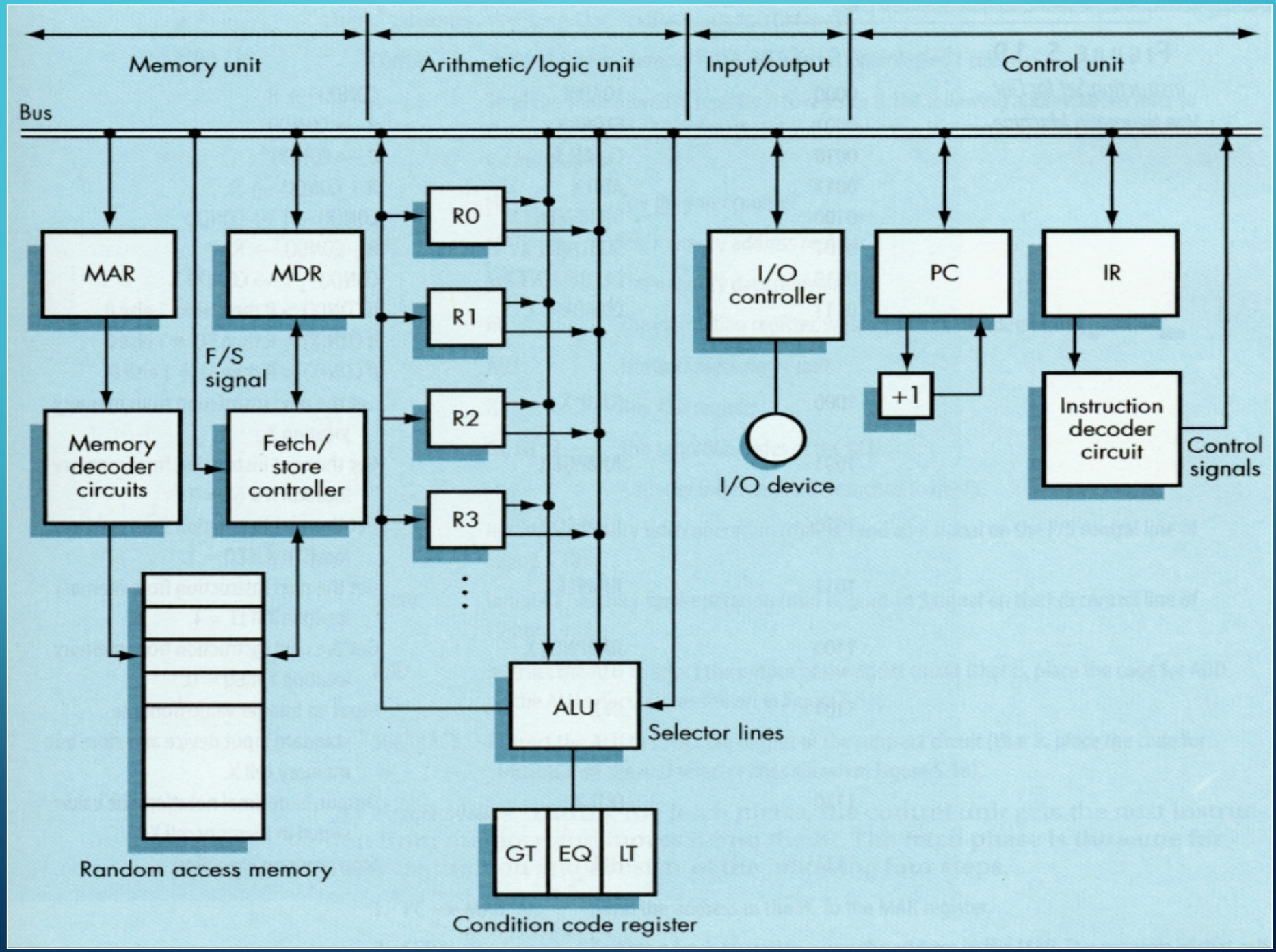
- Named after John von Neumann, Princeton, he designed computer architecture whereby data and instructions would be retrieved from memory, operated on by an ALU, and moved back to memory (or I/O)
- This architecture is the basis for most modern computers (only parallel processors and a few other unique architectures use a different model)
- Hardware consists of 3 units: CPU (control unit, ALU, registers) , Memory (stores programs and data) and I/O System (including secondary storage)
- Instructions in memory are executed sequentially unless a program instruction explicitly changes the order

VNA


Primary Memory
Input Units
CPU (Central Processing Unit)
Output Units


main memory
Arithmetic - Logic Unit
Program Control Unit
Input-Output Equipment
op code
address


Central Processing Unit
Program Counter
Registers
Main Memory
Arithmetic Logic Unit
Control Unit
Input/Output System


Memory unit
Arithmetic/logic unit
Input/output
Control unit
Bus
MAR
MDR
R0
R1
R2
R3
I/O controller
PC
IR
F/S signal
Memory decoder circuits
Fetch/ store controller
+1
Instruction decoder circuit
Control signals
I/O device
ALU
Selector lines
GT
EQ
LT
Random access memory
Condition code register

VARIATIONS

- single pathway used to move both data and instructions between memory, I/O and CPU
  - the pathway is implemented as a bus
  - the single pathway creates a bottleneck
    - known as the *von Neumann bottleneck*
  - A variation of this architecture is the *Harvard architecture* which separates data and instructions into two pathways
  - Another variation, used in most computers, is the system bus version in which there are different buses between CPU and memory and memory and I/O

OPS

- The von Neumann architecture operates on the *fetch-execute cycle*
  - Fetch an instruction from memory as indicated by the Program Counter register
  - Decode the instruction in the control unit
  - Data operands needed for the instruction are fetched from memory
  - Execute the instruction in the ALU storing the result in a register
  - Move the result back to memory if needed

HARVARD ARCHITECTURE

- **Harvard architecture** is a computer architecture with physically separate storage and signal pathways for instructions and data.
- Originated from Harvard Mark I relay-based computer,
  - stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters (23 digits wide).
  - had limited data storage, entirely contained within the data processing unit,
  - provided no access to the instruction storage as data,
  - made loading and modifying programs an entirely offline process.

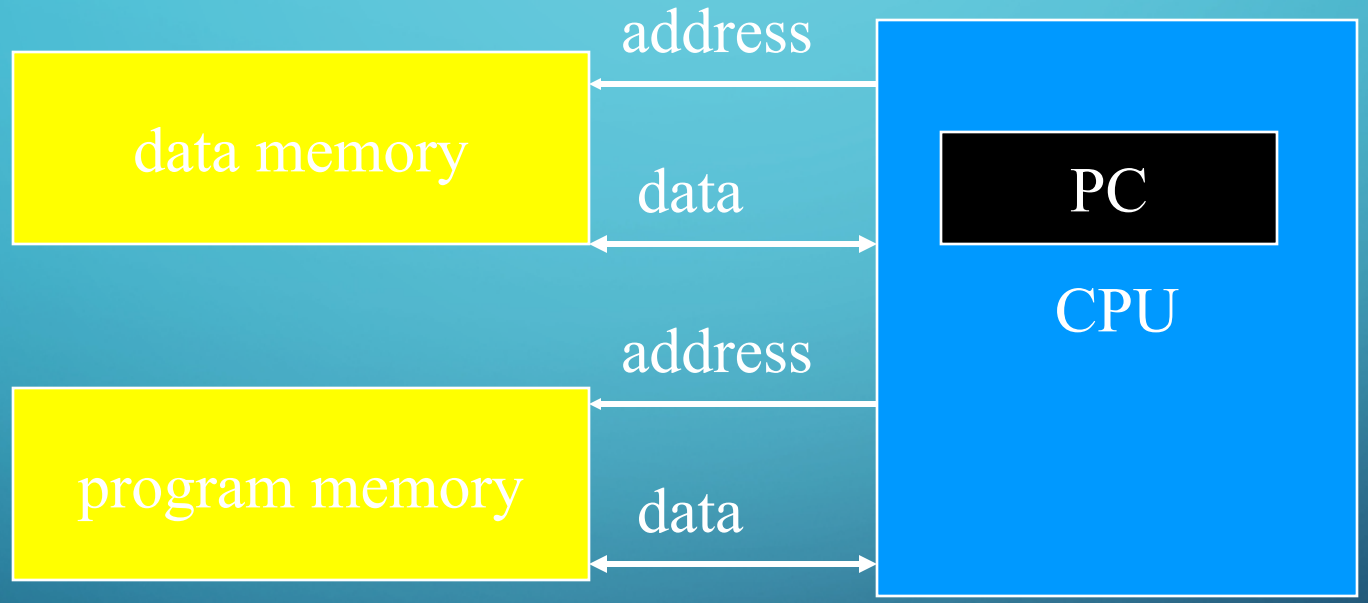
THE HARVARD ARCHITECTURE (2)

- In a computer with a von Neumann architecture (and no cache), the CPU can be either reading an instruction or reading/writing data from/to the memory.
  - Both cannot occur at the same time since the instructions and data use the same bus system.
- In a computer using the Harvard architecture, the CPU can read both an instruction and perform a data memory access at the same time, even without a cache.
- A Harvard architecture computer can thus be faster for a given circuit complexity because instruction fetches and data access do not contend for a single memory pathway.

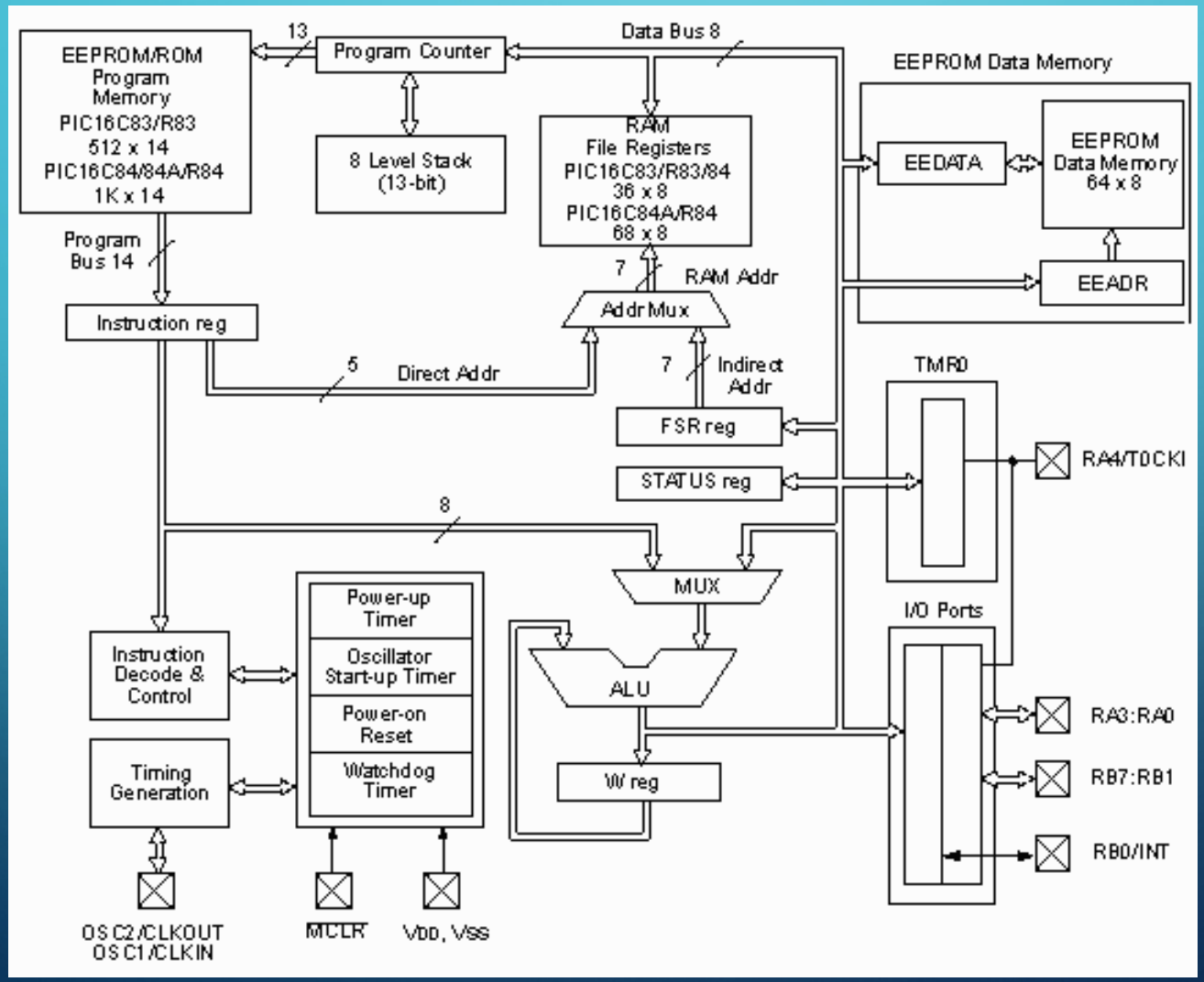
THE HARVARD ARCHITECTURE (3)

- In a Harvard architecture, there is no need to make the two memories share characteristics. In particular, the word width, timing, implementation technology, and memory address structure can differ.
- In some systems, instructions can be stored in read-only memory while data memory generally requires read-write memory.
- Instruction memory is often wider than data memory.

HARVARD ARCHITECTURE



HARVARD ARCHITECTURE EXAMPLE



Block Diagram of the PIC16C8X

MODIFIED HARVARD ARCHITECTURE

- The Modified Harvard architecture is very like the Harvard architecture but provides a pathway between the instruction memory and the CPU that allows words from the instruction memory to be treated as read-only data.
  - This allows constant data, particularly text strings, to be accessed without first having to be copied into data memory, thus preserving more data memory for read/write variables.
- Special machine language instructions are provided to read data from the instruction memory.
- Standards-based high-level languages, such as the C language, do not support the Modified Harvard Architecture, so that in-line assembly or non-standard extensions are needed to take advantage of it.
- Most modern computers that are documented as Harvard Architecture are, in fact, Modified Harvard Architecture.

VON NEUMANN VS. HARVARD

- Harvard can’t use self-modifying code.
- Harvard allows two simultaneous memory fetches.
- Most DSPs use Harvard architecture for streaming data:
  - greater memory bandwidth;
  - more predictable bandwidth.

CISC V RISC

- CISC = Complex Instruction Set Computer
- RISC = Reduced Instruction Set Computer

RISC VS. CISC

- Complex instruction set computer (CISC):
  - many addressing modes;
  - many operations.
- Reduced instruction set computer (RISC):
  - load/store;
  - pipelinable instructions.

CISC *VS.* RISC (CONTINUED)

- Historically, machines tend to add features over time
  - instruction opcodes
    - IBM 70X went from 24 opcodes to 185 in 10 years
    - same time performance increased 30 times
  - addressing modes
  - special purpose registers
- Motivations are to
  - improve efficiency, since complex instructions can be implemented in hardware and execute faster
  - make life easier for compiler writers
  - support more complex higher-level languages

CISC *VS.* RISC

- Examination of actual code indicated many of these features were not used
- RISC advocates proposed
  - simple, limited instruction set
  - large number of general purpose registers
    - and mostly register operations
  - optimized instruction pipeline
- Benefits should include
  - faster execution of instructions commonly used
  - faster design and implementation