Technology in Action

Alan Evans • Kendall Martin

Mary Anne Poatsy

Eleventh Edition

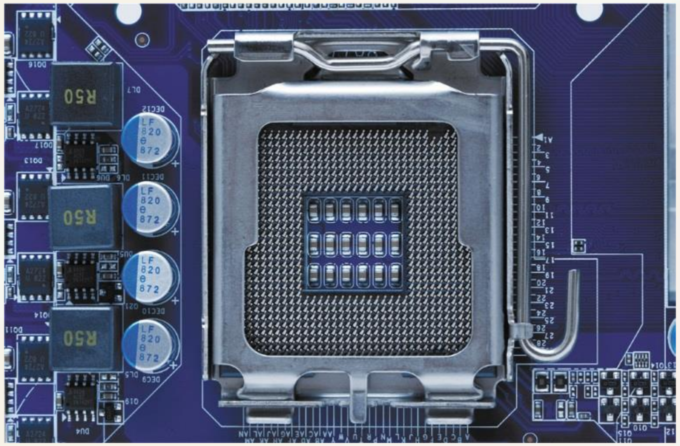
Switches
Electrical Switches

- Computers only understand two states
  - Binary language consists of two numbers: 1 or 0
  - Electrical switches can be switched between 1 and 0 signifying "On" and "Off"
  - Computers contain a huge collection of electrical switches

Lock

Switches
Integrated Circuits

- Tiny regions of semiconductor material
- Support huge number of transistors
- No more than ¼ inch in size
- Billions of transistors can fit on an integrated circuit
- CPUs are microprocessor chips



Number Systems
The Base 10 Number System

- Number system is organized plan for representing a number
- Base 10 uses 10 digits (0–9)
- To represent a number, break it down into groups of ones, tens, hundreds, etc.

6,954 =

10^3 1,000s place		10^2 100s place		10^1 10s place		10^0 1s place
6 * 1,000	+	9 * 100	+	5 * 10	+	4 * 1

Number Systems
The Base 2 (or Binary) Number System

- Base 2 or binary uses two digits (0 and 1)
- Describes value as sum of powers of 2: 1, 2, 4, 8, 16, 32, 64, and so on

11 =

2^3 8s place	2^2 4s place	2^1 2s place	2^0 1s place
1	0	1	1

Number Systems
Representing Integers

- Base-10 system
  - Whole number represented as sum of powers of 10
- Binary system
  - Whole number is represented as sum of powers of 2
- Windows 10
  - Scientific Calculator supports conversion between decimal and binary

Number Systems
Hexadecimal Notation

- Hexadecimal notation used to avoid working with long strings of 1s and 0s
- Base 16 uses 16 digits (0–9 and A–F)
  - A equals 10, B equals 11, etc.
- Easier for computer scientists to use 43 than 1000011

Sample Hexadecimal Values

DECIMAL NUMBER	BINARY VALUE	HEXADECIMAL VALUE
00	0000	00
01	0001	01
02	0010	02
03	0011	03
04	0100	04
05	0101	05
06	0110	06
07	0111	07
08	1000	08
09	1001	09
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Number Systems
Representing Characters: ASCII

- American Standard Code for Information Interchange (ASCII) represents each letter or character as 8-bit binary code
  - Each binary digit is a bit
  - 8 binary digits (or bits) create one byte

ASCII Standard Code for a Sample of Letters and Characters

ASCII CODE	REPRESENTS THIS SYMBOL	ASCII CODE	REPRESENTS THIS SYMBOL
01000001	A	01100001	a
01000010	B	01100010	b
01000011	C	01100011	c
01011010	Z	00100011	#
00100001	!	00100100	$
00100010	“	00100101	%

Number Systems
Representing Characters: Unicode

- ASCII can use only 256 codes
- Unicode uses 16 bits and can represent nearly 1,115,000 code points
- Currently assigns more than 96,000 unique character symbols

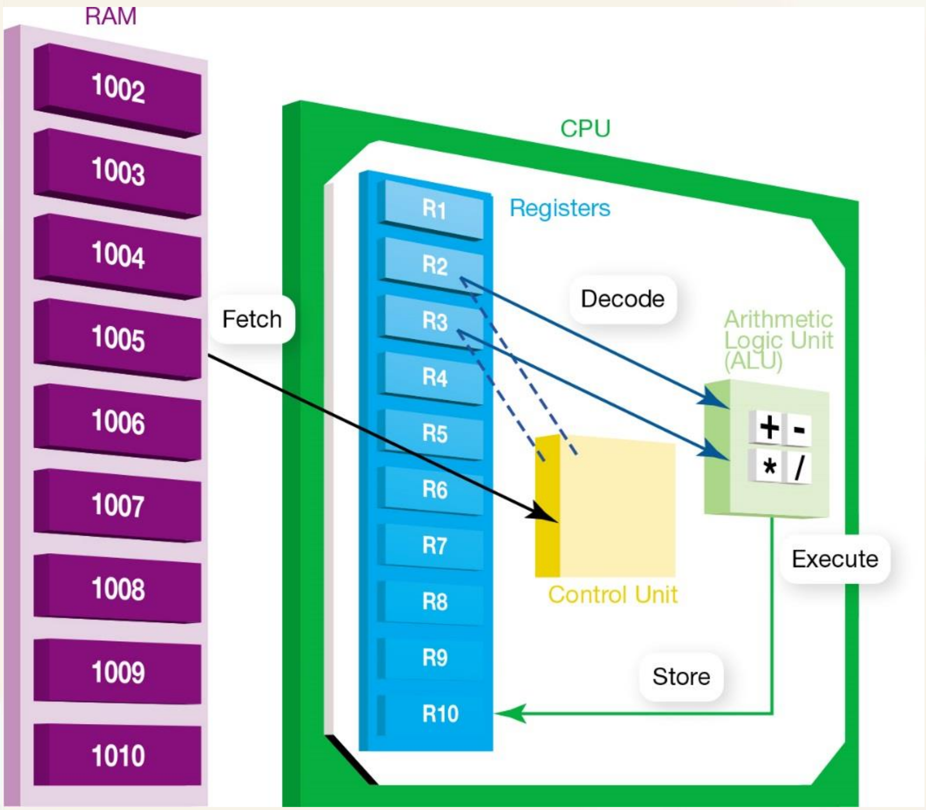
Number Systems
Representing Characters: Unicode (cont.)

- First 128 characters of Unicode are identical to ASCII
- Unicode can represent alphabets of all modern and historic languages
- Will probably replace ASCII as standard

Number Systems Interpretation

- Positive and negative integers stored using signed integer notation
- Decimal numbers stored according to IEEE floating-point standard
- Letters and symbols stored according to ASCII code or Unicode

How the CPU Works



- Machine cycle refers to series of general steps CPU performs
  - Fetch
  - Decode
  - Execute
  - Store

How the CPU Works
The Control Unit

- Manages switches inside CPU
- Remembers:
  - Sequence of processing stages
  - How switches are set for each stage
- With each beat of system clock, control unit moves each switch to correct on/off setting and performs work of that stage

How the CPU Works
The System Clock

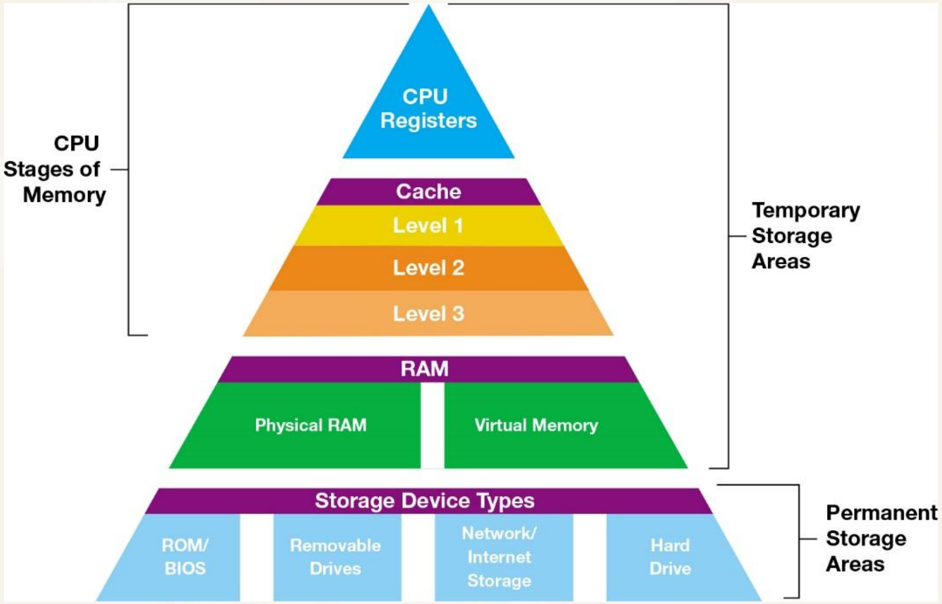
- Moves CPU from one stage of machine cycle to the next
- Acts as a metronome, keeping a steady beat or tick
  - Ticks, known as the clock cycle, set the pace
  - Pace, known as clock speed, is measured in hertz (Hz)
- Today’s speed is measured in gigahertz (GHz), 1 billion clock ticks per second

How the CPU Works
Stage 1: The Fetch Stage

- Data and program instructions are stored in various areas of computer system
- Program or data is moved to RAM from hard drive
- As instructions are needed, they are moved from RAM into registers
  - Storage areas located on CPU

How the CPU Works
Stage 1: The Fetch Stage (cont.)

- Cache memory—small blocks of memory located directly on and next to CPU chip
  - Stores recent or frequently used instructions
  - Faster access than RAM



How the CPU Works
Stage 1: The Fetch Stage (cont.)

- Level 1 Cache
  - Searched if next instruction is not in CPU register
  - Built onto CPU
  - Stores commands that have been used

How the CPU Works
Stage 1: The Fetch Stage (cont.)

- Level 2 Cache
  - Searched if instruction is not in Level 1
  - Located on CPU or chip next to CPU
  - Takes longer to access
  - Contains more storage than Level 1

How the CPU Works
Stage 1: The Fetch Stage (cont.)

- Level 3 Cache
  - Checks only if instruction is not in Level 1 or Level 2
  - Checked before making longer trip to RAM
  - Holds between 2 and 12 megabytes of data
  - Can hold entire programs

How the CPU Works
Stage 2: The Decode Stage

- CPU's control unit decodes program's instructions into commands
- Instruction set
  - Collection of commands CPU can execute
  - Written in assembly language
  - Assembly language translated into binary code
- Machine language—long strings of binary code

How the CPU Works
Stage 3: The Execute Stage

- Arithmetic logic unit (ALU)
  - Mathematical operations
    - Addition
    - Subtraction
    - Multiplication
    - Division
  - Test comparisons of values (<, >, =)
  - Logical OR, AND, and NOT operations
  - Word size is number of bits worked with at a time

How the CPU Works
Stage 4: The Store Stage

- Results produced by ALU are stored in registers
- Instruction explains which register to use
- When entire instruction is completed, next instruction will be fetched
- The fetch–decode–execute–store cycle begins again

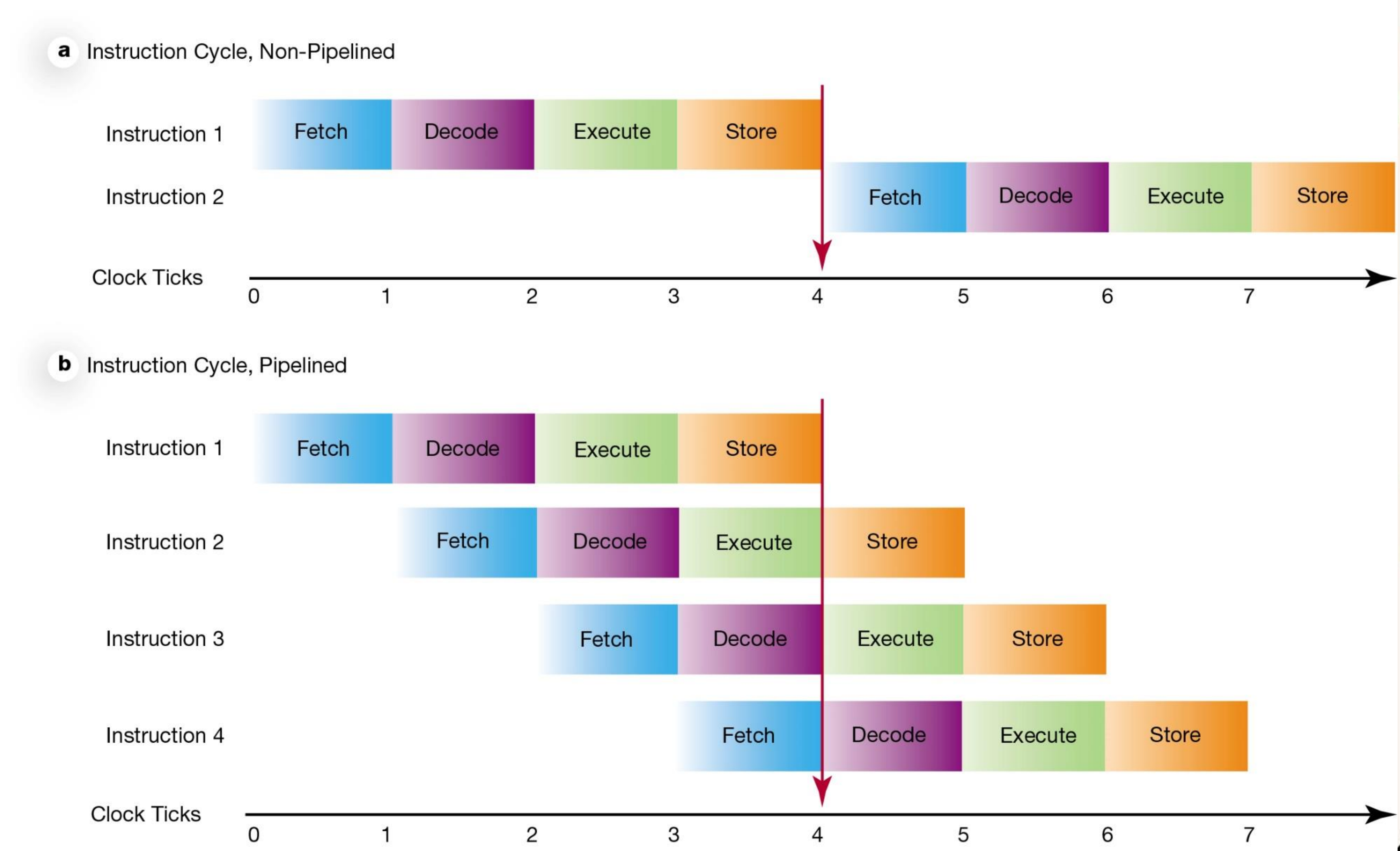
Making CPUs Even Faster

- Manufacturers can increase CPU performance
  - Pipelining (hyperthreading)
  - Specialized, faster instructions for multimedia and graphics
  - Multiple independent processing paths inside the CPU

Making CPUs Even Faster
Pipelining

- CPU works on more than one stage or instruction at a time
- Boosts CPU performance
- System clock indicates when instructions move to next process
- Can potentially run four times faster

Making CPUs Even Faster
Pipelining



Making CPUs Even Faster
Specialized Multimedia Instructions

- New processors incorporate multimedia instructions into basic instruction set
- Multimedia-specific instructions work to accelerate video and audio processing
- New instructions work to allow CPU to deliver faster data protection

Making CPUs Even Faster
Multiple Processing Efforts

- Many high-end server systems use large number of processors
- Multicore processing
  - Quad core processors have four separate parallel processing paths
  - Six- and eight-core processors are available
- Parallel processing uses multiple computers to work on portion of same problem simultaneously