Lecture-5

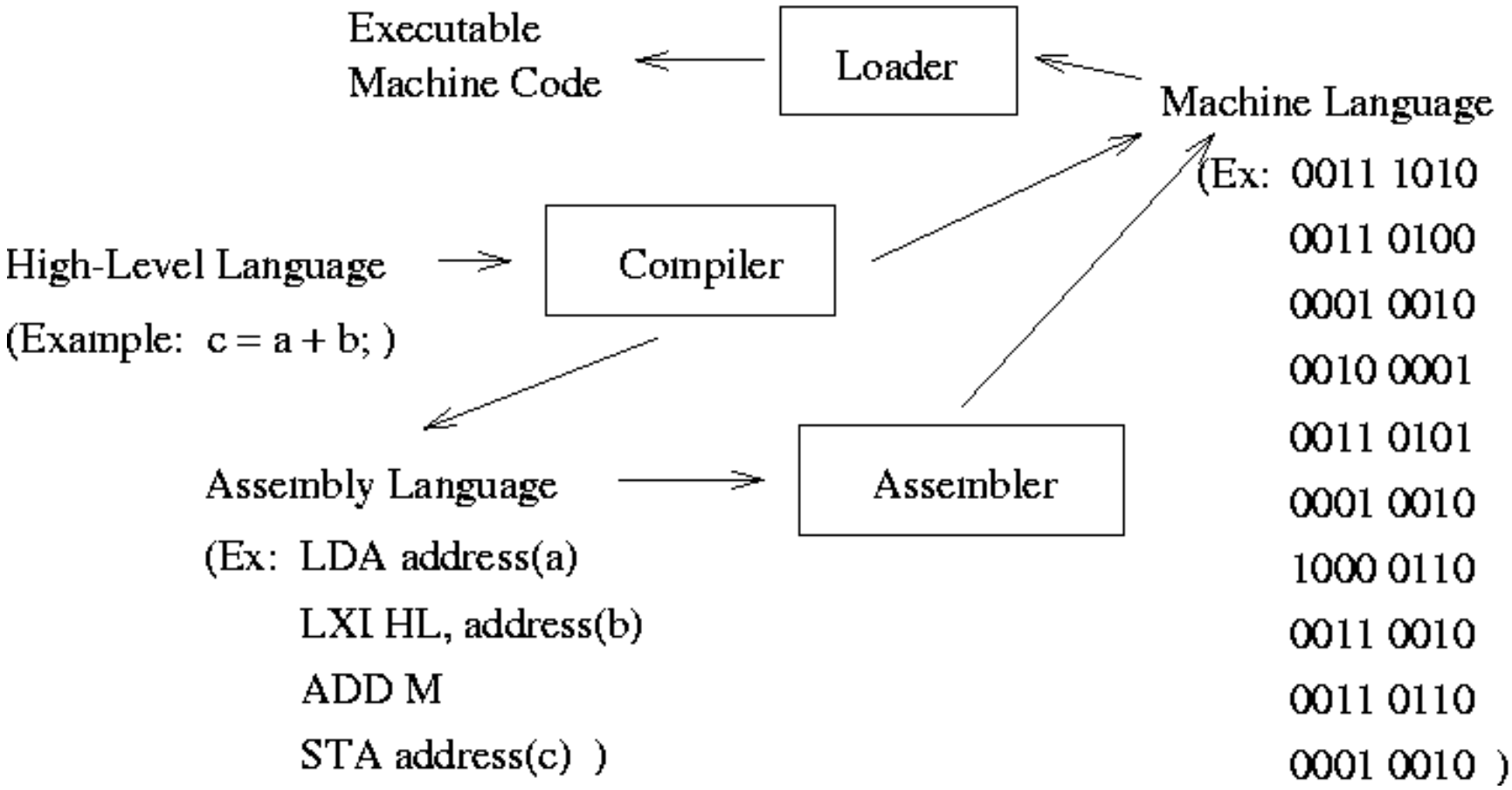
Chapter-3.3

Computer Architecture and Organization-
Jhon P. Hayes

Processor Basics

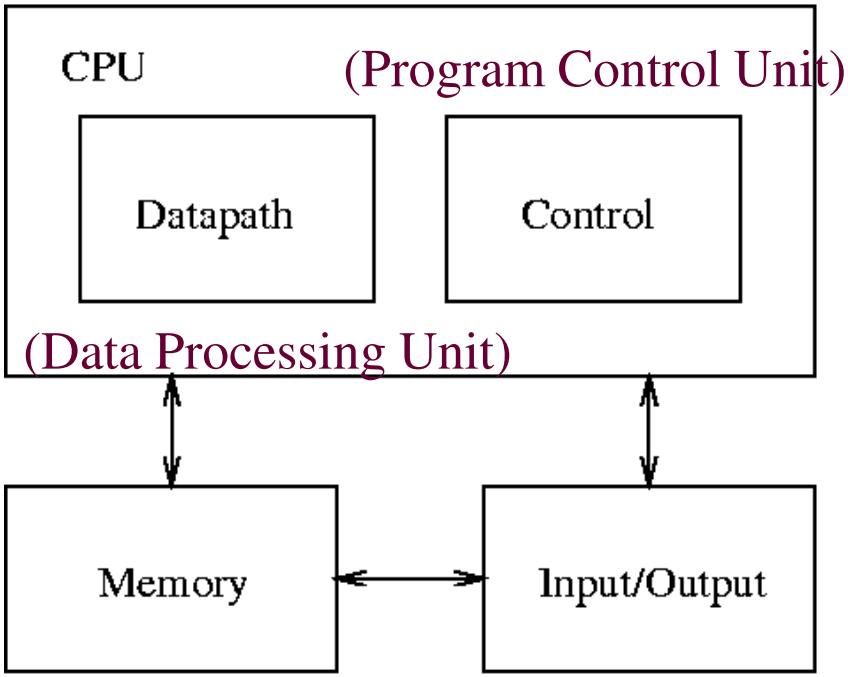
(Pipelining, Instruction Set Design)

Computer Program Execution

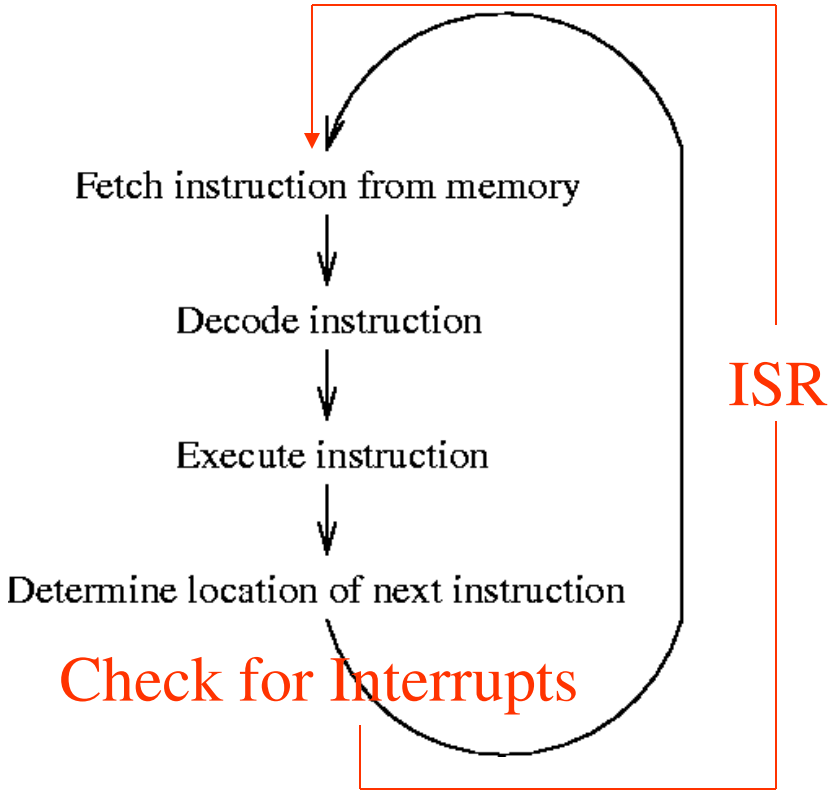


Block Diagram and Basic Operation

- Block Diagram



- Basic Operation



Instruction Set Design

- Fixed versus Variable-Length Instruction Formats
  - Fixed-length instruction format
    - use same number of bits to represent all instructions
    - leads to simpler (and faster) instruction decoding
    - facilitates fast and effective prefetching of instructions
  - Variable-length instruction format
    - use different numbers of bits for different instructions
    - can accommodate short and long instructions (requiring extra bits to represent data operands) without a waste of memory
    - prefetching and decoding of instructions is more difficult

- Example of fixed-length instruction formats used in the MIPS (Millions of Instruction Per Second) microprocessor

R-type Instruction Format

opcode	source reg. 1	source reg. 2	destin. register	shift amount	function variant
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

I-type Instruction Format

opcode	source reg. 1	source reg. 2	address
6 bits	5 bits	5 bits	16 bits

Figure 6.4. Two instruction formats used in the MIPS architecture.
[Lee 2000]

MIPS = Microprocessor without Interlocked Pipeline Stages

- Variable-length instruction formats used in the Intel 8080 architecture

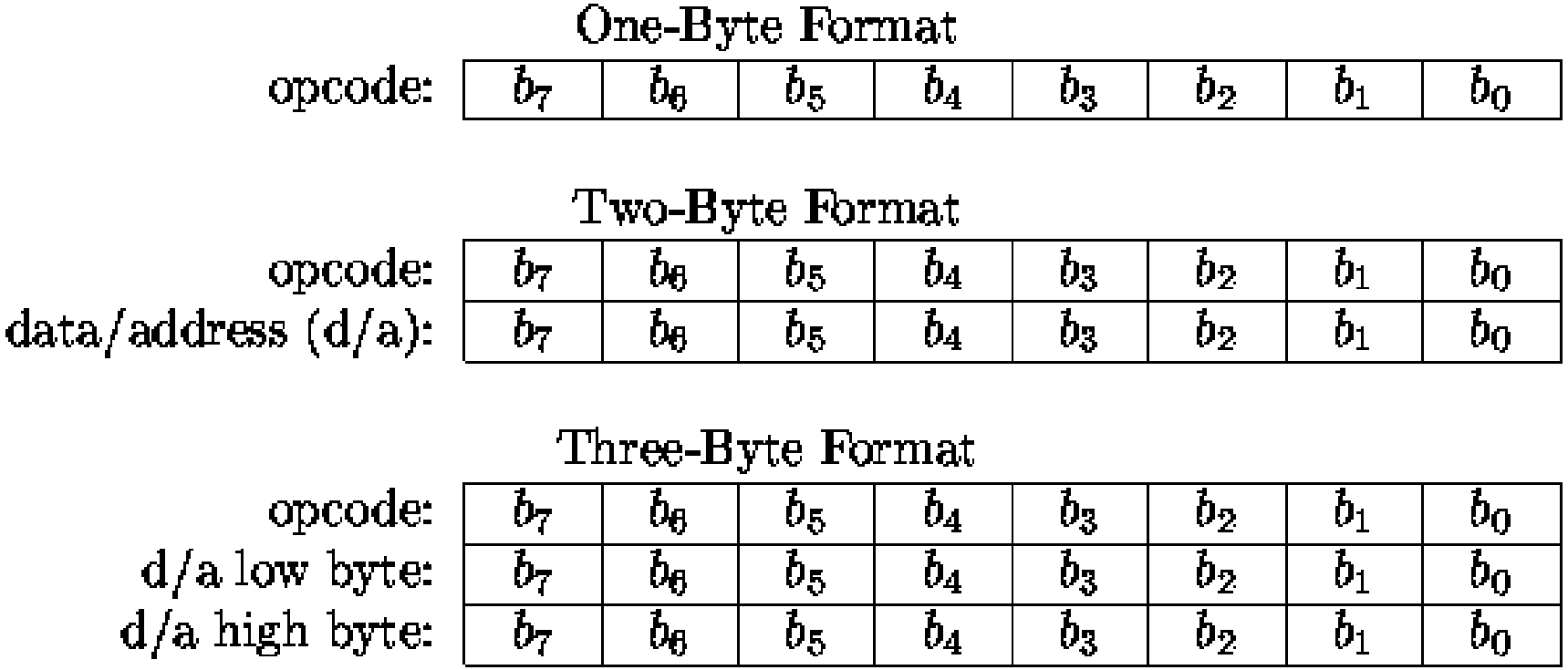


Figure 6.5. Instruction formats for the Intel 8080 architecture. [Lee 2000]

Number of Data Operands

- Zero-operand instructions
  - data is accessed from the “stack” (a LIFO (last-in-first-out) queue)
- One-operand instructions
  - the accumulator (ACC) register is used as the default second data input and destination
- Two-operand instructions
  - one of the data inputs is the default destination
- Three-operand instructions

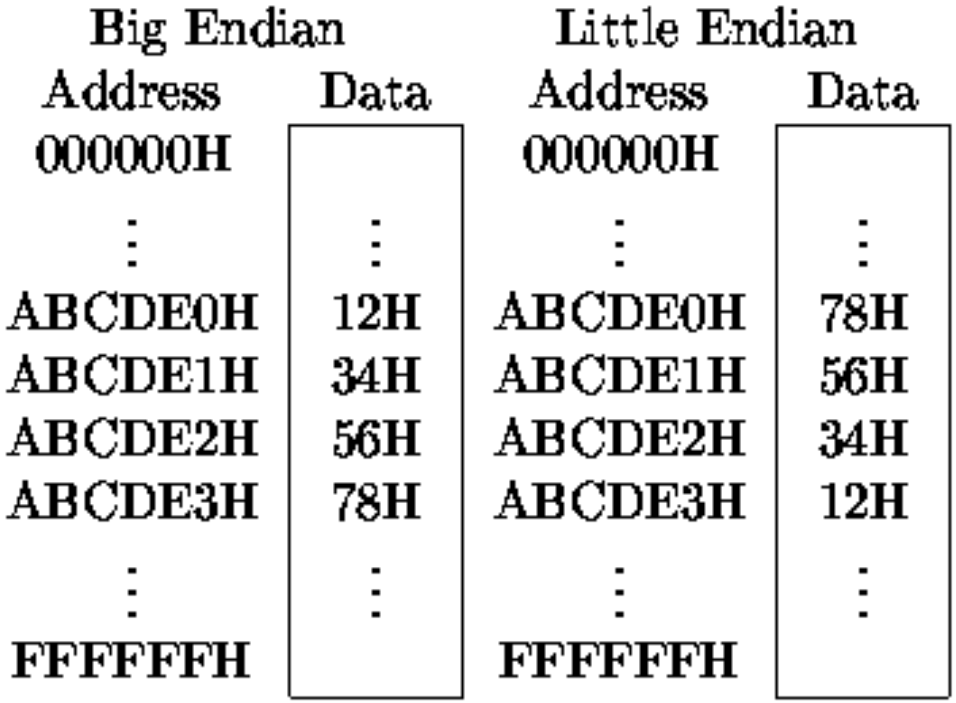
0-operand		1-operand		2-operand		3-operand	
PUSH	Rx	MOVE	ACC, Rx	MOVE	Rz, Rx	ADD	Rz, Rx, Ry
PUSH	Ry	ADD	Ry	ADD	Rz, Ry		
ADD		MOVE	Rz, ACC				
POP	Rz						

Figure 6.6. Examples of 0, 1, 2, and 3-operand instruction sequences for the operation $\mathbf{z \leftarrow x + y}$ assuming all data are in registers.

Endian Mode

Big-**endian** and little-**endian** are terms that describe the order in which a sequence of bytes are stored in computer memory.

Example Instruction: Store 12345678H, ABCDE0H



[Lee 2000]

Figure 6.7. An example of big and little endian addressing modes.

Addressing Modes

	Location of Data
• register (or register-direct) addressing:	R1
• register indirect addressing:	M[R1]
• immediate addressing:	data
• direct (or absolute) addressing:	M[address]
• indirect addressing:	M[M[address]]
• implicit addressing:	default location
• relative & indexed addressing:	M[R1+address]
• pre-decrement, post-decrement, pre-increment, ...	

Modern CPUs

- Employ a variety of speedup techniques
  - cache memories
  - special purpose instructions (e.g., test-and-set, MMX)
  - redundant hardware modules -> superscalar, VLIW, ...
  - support for parallel processing
  - pipelining

Modern CPUs

There are two advanced features of the commercial microprocessor:

- Reduced Instruction Set Computer (RISC) and
- Complex Instruction Set Computer (CISC)

There are many way in which basic design of CPU can be improved. Most recent CPUs contain the following extensions which improve their performance and case of Programming.

» Multipurpose register set for storing data and address

» Additional data, instruction and address type

» Register to indicate computation status

» Program control stack

Pipeline

Modern CPUs have a variety of speedup techniques, including cache memories, and several forms of instruction level parallelism. Such parallelism may be present in the internal organization of DPU or in the overlapping of the operations carried out by the DPU and PCU. Overlapping of instruction fetching and execution is an example of instruction pipelining which is an important speedup feature of RISC processor.

Pipeline

Each instruction can be thought of as passing through two consecutive stages of processing: a fetch stage implemented mainly by PCU and an execution stage implemented mainly by DPU. Hence two instructions can be processed simultaneously in every CPU clock cycle, with one completing its fetch phase and other completing it's execute phase. This Simultaneous process is called two stage instruction pipelines. [Fig 3.8]

Instruction Pipelining

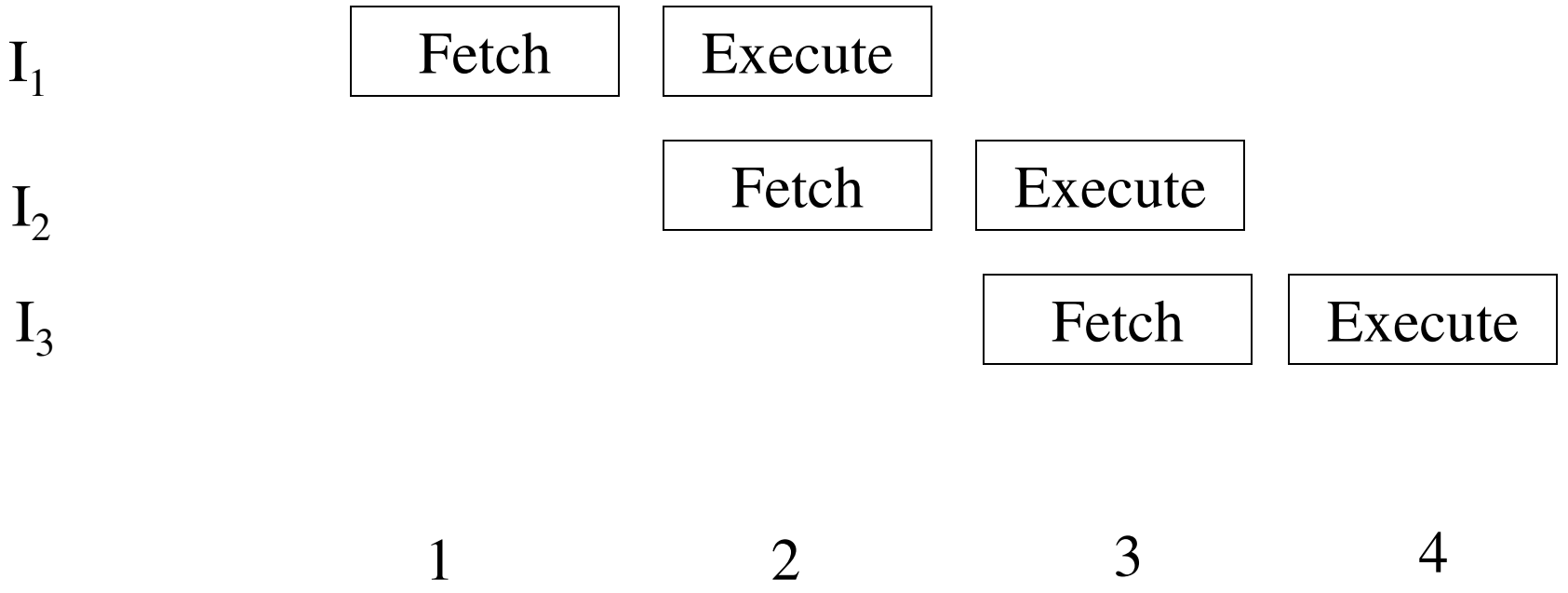


Fig 3.8: Overlapping instruction in a two stage instruction pipeline

Instruction Pipelining

The two principal number formats are **fixed-point** and floating-point. Fixed-point formats allow a limited range of values and have relatively simple hardware requirements.

Floating-point numbers, on the other hand, allow a much larger range of values but require either costly processing hardware or lengthy software implementations.