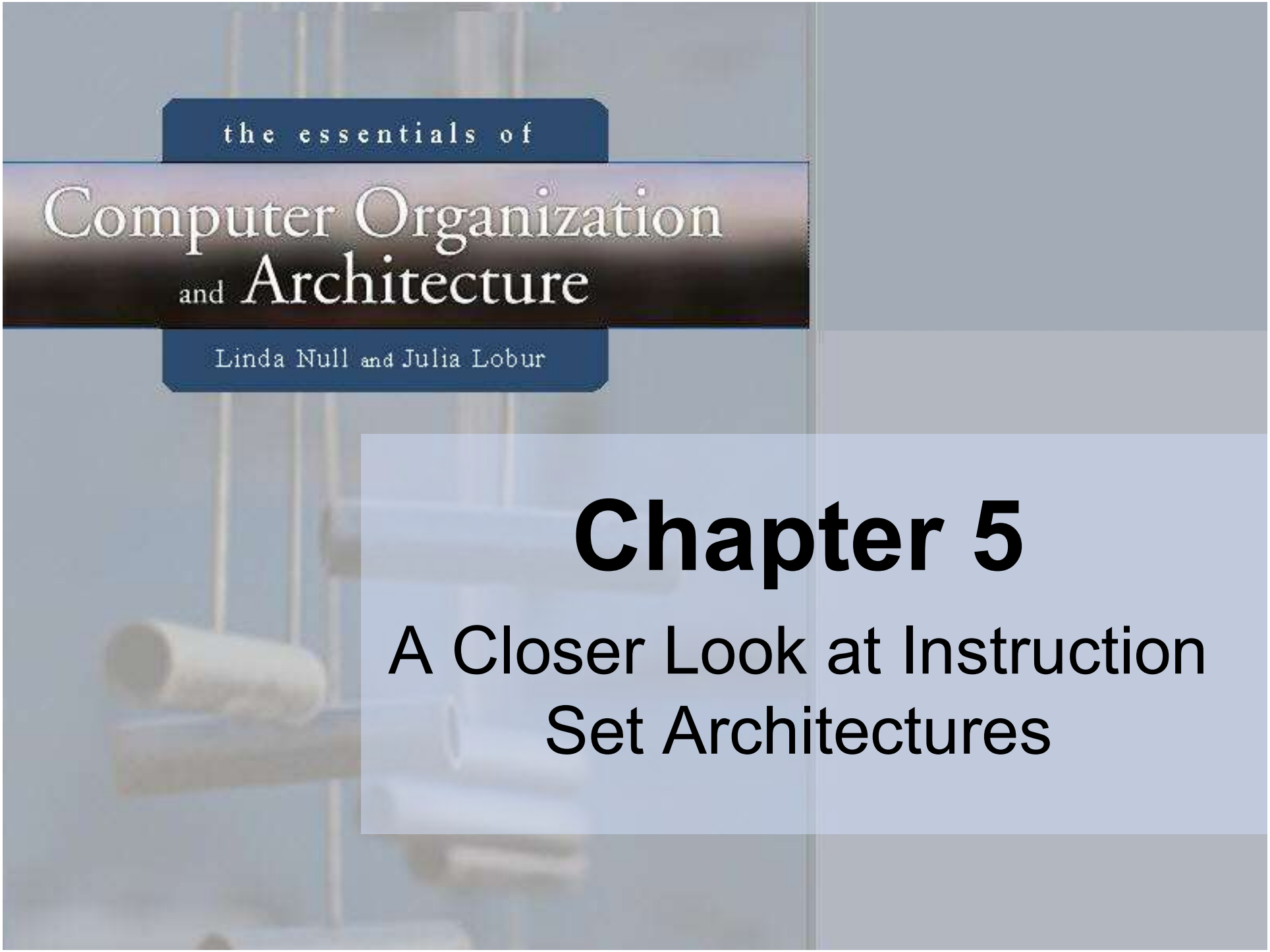

Chapter 5

A Closer Look at Instruction Set Architectures

Chapter 5
REVIEW OF ESSENTIAL TERMS AND CONCEPTS

7. What are the pros and cons of fixed-length and variable-length instructions? Which is currently more popular?

14. What is an address mode?

15. Give examples of immediate, direct, register, indirect, register indirect, and indexed addressing.

18. Explain the concept behind pipelining.

Chapter 5
EXERCISES

2. Show how the following values would be stored by machines with 32-bit words, using little endian and then big endian format. Assume each value starts at address 10_{16}. Draw a diagram of memory for each, placing the appropriate values in the correct(and labeled) memory locations.

- **a)** $456789A1_{16}$
- **b)** $0000058A_{16}$
- **c)** 14148888_{16}

Chapter 5
EXERCISES

3. Fill in the following table to show how the given integers are represented, assuming that 16 bits are used to store values and the machine uses two's complement notation.

Integer	Binary	Hex	4 Byte Big Endian (Hex value as seen in memory)	4 Byte Little Endian (Hex value as seen in memory)
28				
2216				
–18675				
–12				
31456				

Chapter 5
EXERCISES

6. The first two bytes of a 2M X 16 main memory have the following hex values:.

- **Byte 0 is FE**
- **Byte 1 is 01**

If these bytes hold a 16-bit two's complement integer, what is its actual decimal value if:

- **a) memory is big endian?**
- **b) memory is little endian?**

Chapter 5
EXERCISES

10. A computer has 32-bit instructions and 12-bit addresses. Suppose there are 250 two-address instructions. How many one-address instructions can be formulated? Explain your answer.

Chapter 5
EXERCISES

12. Convert the following expressions from infix to reverse Polish (postfix) notation.

a) $X \times Y + W \times Z + V \times U$

b) $W \times X + W \times (U \times V + Z)$

c) $(W \times (X + Y \times (U \times V)))/(U \times (X + Y))$

Chapter 5
EXERCISES

14. Convert the following expressions from reverse Polish notation to infix notation.

a) W X Y Z − + ×

b) U V W X Y Z + × + × +

c) X Y Z + V W − × Z + +

Chapter 5
EXERCISES

16. a) Write the following expression in postfix (reverse Polish) notation. Remember the rules of precedence for arithmetic operators!

$$X = \frac{A - B + C \times (D \times E - F)}{G + H \times K}$$

b) Write a program to evaluate the above arithmetic statement using a stack-organized computer with zero-address instructions (so only Pop and Push can access memory).

Chapter 5
EXERCISES

17. a) In a computer instruction format, the instruction length is 11 bits and the size of an address field is 4 bits. Is it possible to have

5 2-address instructions

45 1-address instructions

32 0-address instructions

using the format? Justify your answer.

b) Assume that a computer architect has already designed 6 two-address and 24 zero-address instructions using the instruction format given in Problem 11. What is the maximum number of one-address instructions that can be added to the instruction set?

Chapter 5
EXERCISES

19. Given 16-bit instructions, is it possible to use expanding opcodes to allow the following to be encoded assuming we have a total of 32 registers? If so, show the encoding. If not, explain why is it not possible.

- **60 instructions with two register operands**
- **30 instructions with one register operand**
- **3 instructions with one 10-bit address**
- **26 instructions with zero operands**

Chapter 5
EXERCISES

21. Suppose we have the instruction Load 1000. Given that memory and register R1 contain the values below:

Memory

1000	1400
...	
1100	400
...	
1200	1000
...	
1300	1100
...	
1400	1300

R1	200

Assuming R1 is implied in the indexed addressing mode, determine the actual value loaded into the accumulator and fill in the table below:

Mode	Value Loaded into AC
Immediate	
Direct	
Indirect	
Indexed	

Chapter 5
EXERCISES

22. Suppose we have the instruction Load 500. Given that memory and register R1 contain the values below:

Memory

100	600
...	
400	300
...	
500	100
...	
600	500
...	
700	800

R1: 200

Assuming R1 is implied in the indexed addressing mode, determine the actual value loaded into the accumulator and fill in the table below:

Mode	Value Loaded into AC
Immediate	
Direct	
Indirect	
Indexed	

Chapter 5
EXERCISES

23. A nonpipelined system takes 200ns to process a task. The same task can be processed in a 5-segment pipeline with a clock cycle of 40ns. Determine the speedup ratio of the pipeline for 200 tasks. What is the maximum speedup that could be achieved with the pipeline unit over the nonpipelined unit?

Chapter 5
EXERCISES

24. A nonpipeline system takes 100ns to process a task. The same task can be processed in a 5-stage pipeline with a clock cycle of 20ns. Determine the speedup ratio of the pipeline for 100 tasks. What is the theoretical speedup that could be achieved with the pipeline system over a nonpipelined system?

Chapter 5
EXERCISES

27. A digital computer has a memory unit with 24 bits per word. The instruction set consists of 150 different operations. All instructions have an operation code part (opcode) and an address part (allowing for only one address). Each instruction is stored in one word of memory.:

a) How many bits are needed for the opcode?

b) How many bits are left for the address part of the instruction?

c) What is the maximum allowable size for memory?

d) What is the largest unsigned binary number that can be accommodated in one word of memory?

Chapter 5
EXERCISES

28. The memory unit of a computer has 256K words of 32 bits each. The computer has an instruction format with 4 fields: an opcode field; a mode field to specify 1 of 7 addressing modes; a register address field to specify 1 of 60 registers; and a memory address field. Assume an instruction is 32 bits long. Answer the following:

a) How large must the mode field be?

b) How large must the register field be?

c) How large must the address field be?

d) How large is the opcode field?

Chapter 5
EXERCISES

29. Suppose an instruction takes four cycles to execute in a nonpipelined CPU: one cycle to fetch the instruction, one cycle to decode the instruction, one cycle to perform the ALU operation, and one cycle to store the result. In a CPU with a 4-stage pipeline, that instruction still takes four cycles to execute, so how can we say the pipeline speeds up the execution of the program?