1.8 Key Terms, Review Questions, and Problems

Key Terms

application processor

arithmetic and logic unit (ALU)

ARM

central processing unit (CPU)

chip

computer architecture

computer organization

control unit

core

dedicated processor

deeply embedded system

embedded system

gate

input–output (I/O)

instruction set architecture (ISA)

integrated circuit

Intel x86

Internet of things (IoT)

main memory

memory cell

memory management unit (MMU)

memory protection unit (MPU)

microcontroller

microelectronics

microprocessor

motherboard

multichip module (MCM)

multicore

multicore processor

printed circuit board

processor

registers

semiconductor

semiconductor memory

system bus

system interconnection

transistor

Review Questions

1.1 What, in general terms, is the distinction between computer organization and computer architecture?
1.2 What, in general terms, is the distinction between computer structure and computer function?
1.3 What are the four main functions of a computer?
1.4 List and briefly define the main structural components of a computer.
1.5 List and briefly define the main structural components of a processor.
1.6 What is a stored program computer?
1.7 Explain Moore's law.
1.8 What is the key distinguishing feature of a microprocessor?

Problems

1.1 You are to write an IAS program to compute the results of the following equation.

$$Y = \sum_{X=1}^{N} X$$

Assume that the computation does not result in an arithmetic overflow and that *X*, *Y*, and *N* are positive integers with $N \geq 1$. *Note*: The IAS did not have assembly language, only machine language.

a. Use the equation $\text{Sum}(Y) = \frac{N(N+1)}{2}$ when writing the IAS program.
b. Do it the "hard way," without using the equation from part (a).

1.2

a. On the IAS, what would the machine code instruction look like to load the contents of memory address 2 to the accumulator?
b. How many trips to memory does the CPU need to make to complete this instruction during the instruction cycle?

1.3 On the IAS, describe in English the process that the CPU must undertake to read a value from memory and to write a value to memory in terms of what is put into the MAR, MBR, address bus, data bus, and control bus.
1.4 Given the memory contents of the IAS computer shown below,

Address	Contents

08A	010FA210FB
08B	010FA0F08D
08C	020FA210FB

show the assembly language code for the program, starting at address 08A. Explain what this program does.

1.5 In **Figure 1.6** , indicate the width, in bits, of each data path (e.g., between AC and ALU).

1.6 In the IBM 360 Models 65 and 75, addresses are staggered in two separate main memory units (e.g., all even-numbered words in one unit and all odd-numbered words in another). What might be the purpose of this technique?

1.7 The relative performance of the IBM 360 Model 75 is 50 times that of the 360 Model 30, yet the instruction cycle time is only 5 times as fast. How do you account for this discrepancy?

1.8 While browsing at Billy Bob's computer store, you overhear a customer asking Billy Bob what is the fastest computer in the store that he can buy. Billy Bob replies, "You're looking at our Macintoshes. The fastest Mac we have runs at a clock speed of 1.2 GHz. If you really want the fastest machine, you should buy our 2.4-GHz Intel Pentium IV instead." Is Billy Bob correct? What would you say to help this customer?

1.9 The ENIAC, a precursor to the ISA machine, was a decimal machine, in which each register was represented by a ring of 10 vacuum tubes. At any time, only one vacuum tube was in the ON state, representing one of the 10 decimal digits. Assuming that ENIAC had the capability to have multiple vacuum tubes in the ON and OFF state simultaneously, why is this representation "wasteful" and what range of integer values could we represent using the 10 vacuum tubes?

1.10 For each of the following examples, determine whether this is an embedded system, explaining why or why not.

a. Are programs that understand physics and/or hardware embedded? For example, one that uses finite-element methods to predict fluid flow over airplane wings?
b. Is the internal microprocessor controlling a disk drive an example of an embedded system?
c. I/O drivers control hardware, so does the presence of an I/O driver imply that the computer executing the driver is embedded?
d. Is a PDA (Personal Digital Assistant) an embedded system?
e. Is the microprocessor controlling a cell phone an embedded system?
f. Are the computers in a big phased-array radar considered embedded? These radars are 10-story buildings with one to three 100-foot diameter radiating patches on the sloped sides of the building.
g. Is a traditional flight management system (FMS) built into an airplane cockpit considered embedded?
h. Are the computers in a hardware-in-the-loop (HIL) simulator embedded?
i. Is the computer controlling a pacemaker in a person's chest an embedded computer?
j. Is the computer controlling fuel injection in an automobile engine embedded?

2.7 Key Terms, Review Questions, and Problems

Key Terms

Amdahl's law

arithmetic mean (AM)

base metric

benchmark

clock cycle

clock cycle time

clock rate

clock speed

clock tick

cycles per instruction (CPI)

functional mean (FM)

general-purpose computing on GPU (GPGPU)

geometric mean (GM)

graphics processing unit (GPU)

harmonic mean (HM)

instruction execution rate

Little's law

many integrated core (MIC)

microprocessor

MIPS rate

multicore

peak metric

rate metric

reference machine

speed metric

SPEC

system under test

throughput

Review Questions

2.1 List and briefly define some of the techniques used in contemporary processors to increase speed.
2.2 Explain the concept of performance balance.
2.3 Explain the differences among multicore systems, MICs, and GPGPUs.
2.4 Briefly characterize Amdahl's law.
2.5 Briefly characterize Little's law.
2.6 Define MIPS and FLOPS.
2.7 List and define three methods for calculating a mean value of a set of data values.
2.8 List the desirable characteristics of a benchmark program.
2.9 What are the SPEC benchmarks?
2.10 What are the differences among base metric, peak metric, speed metric, and rate metric?

Problems

2.1 A benchmark program is run on a 40 MHz processor. The executed program consists of 100,000 instruction executions, with the following instruction mix and clock cycle count:

Instruction Type	Instruction Count	Cycles per Instruction
Integer arithmetic	45,000	1
Data transfer	32,000	2
Floating point	15,000	2
Control transfer	8000	2

Determine the effective CPI, MIPS rate, and execution time for this program.
2.2 Consider two different machines, with two different instruction sets, both of which have a clock rate of 200 MHz. The following measurements are recorded on the two machines running a given set of benchmark programs:

Instruction Type	Instruction Count (millions)	Cycles per Instruction
Machine A		
Arithmetic and logic	8	1
Load and store	4	3
Branch	2	4
Others	4	3
Machine A		
Arithmetic and logic	10	1

Load and store	8	2
Branch	2	4
Others	4	3

a. Determine the effective CPI, MIPS rate, and execution time for each machine.
b. Comment on the results.

2.3 Early examples of CISC and RISC design are the VAX 11/780 and the IBM RS/6000, respectively. Using a typical benchmark program, the following machine characteristics result:

Processor	Clock Frequency (MHz)	Performance (MIPS)	CPU Time (secs)
VAX 11/780	5	1	12 *x*
IBM RS/6000	25	18	*x*

The final column shows that the VAX required 12 times longer than the IBM measured in CPU time.

a. What is the relative size of the instruction count of the machine code for this benchmark program running on the two machines?
b. What are the *CPI* values for the two machines?

2.4 Four benchmark programs are executed on three computers with the following results:

	Computer A	Computer B	Computer C
Program 1	1	10	20
Program 2	1000	100	20
Program 3	500	1000	50
Program 4	100	800	100

The table shows the execution time in seconds, with 100,000,000 instructions executed in each of the four programs. Calculate the MIPS values for each computer for each program. Then calculate the arithmetic and harmonic means assuming equal weights for the four programs, and rank the computers based on arithmetic mean and harmonic mean.

2.5 The following table, based on data reported in the literature [HEAT84], shows the execution times, in seconds, for five different benchmark programs on three machines.

Benchmark	Processor		
	R	M	Z
E	417	244	134

F	83	70	70
H	66	153	135
I	39,449	35,527	66,000
K	772	368	369

a. Compute the speed metric for each processor for each benchmark, normalized to machine R. That is, the ratio values for R are all 1.0. Other ratios are calculated using **Equation (2.5)** with R treated as the reference system. Then compute the arithmetic mean value for each system using **Equation (2.3)** . This is the approach taken in [HEAT84].
b. Repeat part (a) using M as the reference machine. This calculation was not tried in [HEAT84].
c. Which machine is the slowest based on each of the preceding two calculations?
d. Repeat the calculations of parts (a) and (b) using the geometric mean, defined in **Equation (2.6)** . Which machine is the slowest based on the two calculations?

2.6 To clarify the results of the preceding problem, we look at a simpler example.

Benchmark	Processor		
	X	Y	Z
1	20	10	40
2	40	80	20

a. Compute the arithmetic mean value for each system using X as the reference machine and then using Y as the reference machine. Argue that intuitively, the three machines have roughly equivalent performance and that the arithmetic mean gives misleading results.
b. Compute the geometric mean value for each system, using X as the reference machine and then using Y as the reference machine. Argue that the results are more realistic than with the arithmetic mean.

2.7 Consider the example in **Section 2.5** for the calculation of average CPI and MIPS rate, which yielded the result of $CPI = 2.24$ and MIPS $\text{rate} = 178$. Now assume that the program can be executed in eight parallel tasks or threads, with roughly equal number of instructions executed in each task. Execution is on an 8-core system, with each core (processor) having the same performance as the single processor originally used. Coordination and synchronization between the parts adds an extra 25,000 instruction executions to each task. Assume the same instruction mix as in the example for each task, but increase the CPI for memory reference with cache miss to 12 cycles due to contention for memory.

a. Determine the average CPI.
b. Determine the corresponding MIPS rate.

c. Calculate the speedup factor.
d. Compare the actual speedup factor with the theoretical speedup factor determined by Amdhal's law.

2.8 A processor accesses main memory with an average access time of T_2. A smaller cache memory is interposed between the processor and main memory. The cache has a significantly faster access time of $T_1 < T_2$. The cache holds, at any time, copies of some main memory words and is designed so that the words more likely to be accessed in the near future are in the cache. Assume that the probability that the next word accessed by the processor is in the cache is *H*, known as the hit ratio.

a. For any single memory access, what is the theoretical speedup of accessing the word in the cache rather than in main memory?
b. Let *T* be the average access time. Express *T* as a function of T_1, T_2, and *H*. What is the overall speedup as a function of *H*?
c. In practice, a system may be designed so that the processor must first access the cache to determine if the word is in the cache and, if it is not, then access main memory, so that on a miss (opposite of a hit), memory access time is $T_1 + T_2$. Express *T* as a function of T_1, T_2, and *H*. Now calculate the speedup and compare to the result produced in part (b).

2.9 The owner of a shop observes that on average 18 customers per hour arrive, and there are typically 8 customers in the shop. What is the average length of time each customer spends in the shop?

2.10 We can gain more insight into Little's law by considering **Figure 2.8a** . Over a period of time *T*, a total of *C* items arrive at a system, wait for service, and complete service. The upper solid line shows the time sequence of arrivals, and the lower solid line shows the time sequence of departures. The shaded area bounded by the two lines represents the total "work" done by the system in units of job-seconds; let *A* be the total work. We wish to derive the relationship among *L*, *W*, and λ.

a. **Figure 2.8b** divides the total area into horizontal rectangles, each with a height of one job. Picture sliding all these rectangles to the left so that their left edges line up at $t = 0$. Develop an equation that relates *A*, *C*, and *W*.
b. **Figure 2.8c** divides the total area into vertical rectangles, defined by the vertical transition boundaries indicated by the dashed lines. Picture sliding all these rectangles down so that their lower edges line up at $N(t) = 0$. Develop an equation that relates *A*, *T*, and *L*.
c. Finally, derive $L = \lambda W$ from the results of (a) and (b).

2.11 In **Figure 2.8a** , jobs arrive at times $t = 0, 1, 1.5, 3.25, 5.25,$ and 7.75. The corresponding completion times are $t = 2, 3, 3.5, 4.25, 8.25,$ and 8.75.

a. Determine the area of each of the six rectangles in **Figure 2.8b** and sum to get the total area *A*. Show your work.
b. Determine the area of each of the 10 rectangles in **Figure 2.8c** and sum to get the total area *A*. Show your work.

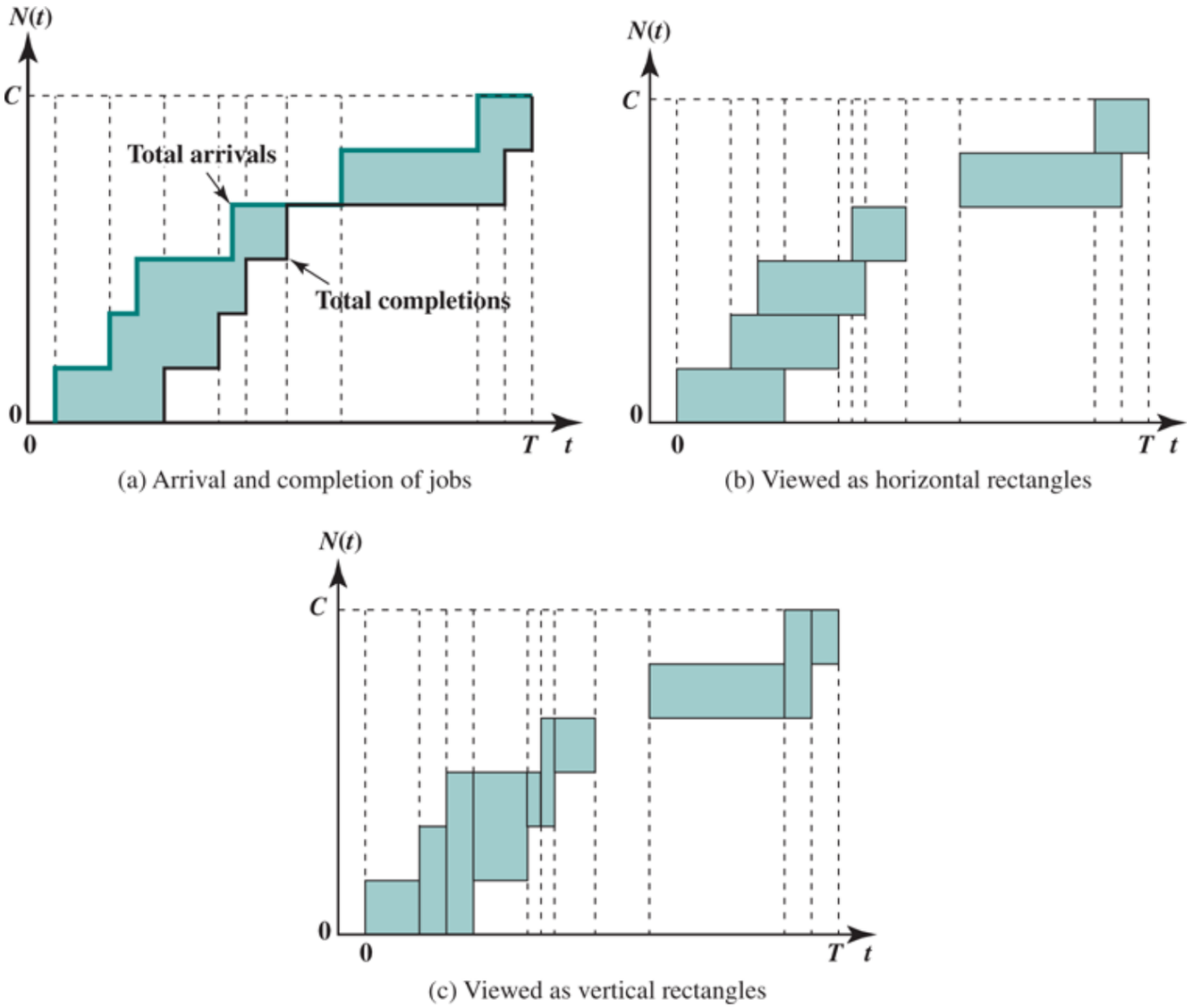


Figure 2.8 Illustration of Little's Law

2.12 In **Section 2.6** , we specified that the base ratio used for comparing a system under test to a reference system is:

$$r_i = \frac{Tref_i}{Tsut_i}$$

a. The preceding equation provides a measure of the speedup of the system under test compared to the reference system. Assume that the number of floating-point operations executed in the test program is I_i. Now show the speedup as a function of the instruction execution rate $FLOPS_i$.
b. Another technique for normalizing performance is to express the performance of a system as a percent change relative to the performance of another system. Express this relative change first as a function of instruction execution rate, and then as a function of execution times.

2.13 Assume that a benchmark program executes in 480 seconds on a reference machine A. The same program executes on systems B, C, and D in 360, 540, and 210 seconds, respectively.

a. Show the speedup of each of the three systems under test relative to A.
b. Now show the relative speedup of the three systems. Comment on the three ways of comparing machines (execution time, speedup, relative speedup).

2.14 Repeat the preceding problem using machine D as the reference machine. How does this affect the relative rankings of the four systems?
2.15 Recalculate the results in **Table 2.2** using the computer time data of **Table 2.4** and comment on the results.
2.16 **Equation 2.5** shows two different formulations of the geometric mean, one using a product operator and one using a summation operator.
a. Show that the two formulas are equivalent.
b. Why would the summation formulation be preferred for calculating the geometric mean?

2.17 **Project.** **Section 2.5** lists a number of references that document the "benchmark means wars." All of the referenced papers are available at **box.com/COA10e**. Read these papers and summarize the case for and against the use of the geometric mean for SPEC calculations.

3.7 Key Terms, Review Questions, and Problems

Key Terms

address bus

address lines

arbitration

balanced transmission

bus

control lines

data bus

data lines

differential signaling

disabled interrupt

distributed arbitration

error control function

execute cycle

fetch cycle

flit

flow control function

instruction cycle

interrupt

interrupt handler

interrupt service routine (ISR)

lane

memory address register (MAR)

memory buffer register (MBR)

multilane distribution

packets

PCI Express (PCIe)

peripheral component interconnect (PCI)

phit

QuickPath Interconnect (QPI)

root complex

system bus

Review Questions

3.1 What general categories of functions are specified by computer instructions?
3.2 List and briefly define the possible states that define an instruction execution.
3.3 List and briefly define two approaches to dealing with multiple interrupts.
3.4 What types of transfers must a computer's interconnection structure (e.g., bus) support?
3.5 List and briefly define the QPI protocol layers.
3.6 List and briefly define the PCIe protocol layers.

Problems

3.1 The hypothetical machine of **Figure 3.4** also has two I/O instructions:

0011 = Load AC from I/O
0111 = Store AC to I/O

In these cases, the 12-bit address identifies a particular I/O device. Show the program execution (using the format of **Figure 3.5**) for the following program:

1. Load AC from device 5.
2. Add contents of memory location 940.
3. Store AC to device 6.

Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.
3.2 The program execution of **Figure 3.5** is described in the text using six steps. Expand this description to show the use of the MAR and MBR.
3.3 Consider a hypothetical 32-bit microprocessor having 32-bit instructions composed of two fields: the first byte contains the opcode and the remainder the immediate operand or an operand address.

a. What is the maximum directly addressable memory capacity (in bytes)?
b. Discuss the impact on the system speed if the microprocessor bus has:
    1. 32-bit local address bus and a 16-bit local data bus, or
    2. 16-bit local address bus and a 16-bit local data bus.
c. How many bits are needed for the program counter and the instruction register?

3.4 Consider a hypothetical microprocessor generating a 16-bit address (for example, assume that the program counter and the address registers are 16 bits wide) and having a 16-bit data bus.

a. What is the maximum memory address space that the processor can access directly if it is connected to a "16-bit memory"?
b. What is the maximum memory address space that the processor can access directly if it is connected to an "8-bit memory"?
c. What architectural features will allow this microprocessor to access a separate "I/O space"?
d. If an input and an output instruction can specify an 8-bit I/O port number, how many 8-bit I/O ports can the microprocessor support? How many 16-bit I/O ports? Explain.

3.5 Consider a 32-bit microprocessor, with a 16-bit external data bus, driven by an 8-MHz input clock. Assume that this microprocessor has a bus cycle whose minimum duration equals four input clock cycles. What is the maximum data transfer rate across the bus that this microprocessor can sustain, in bytes/sec? To increase its performance, would it be better to make its external data bus 32 bits or to double the external clock frequency supplied to the microprocessor? State any other assumptions you make, and explain. *Hint:* Determine the number of bytes that can be transferred per bus cycle.

3.6 Consider a computer system that contains an I/O module controlling a simple keyboard/printer teletype. The following registers are contained in the processor and connected directly to the system bus:

INPR: Input Register, 8 bits

OUTR: Output Register, 8 bits

FGI: Input Flag, 1 bit

FGO: Output Flag, 1 bit

IEN: Interrupt Enable, 1 bit

Keystroke input from the teletype and printer output to the teletype are controlled by the I/O module. The teletype is able to encode an alphanumeric symbol to an 8-bit word and decode an 8-bit word into an alphanumeric symbol.

a. Describe how the processor, using the first four registers listed in this problem, can achieve I/O with the teletype.
b. Describe how the function can be performed more efficiently by also employing IEN.

3.7 Consider two microprocessors having 8- and 16-bit-wide external data buses, respectively. The two processors are identical otherwise and their bus cycles take just as long.

a. Suppose all instructions and operands are two bytes long. By what factor do the maximum data transfer rates differ?
b. Repeat assuming that half of the operands and instructions are one byte long.

3.8 **Figure 3.26** indicates a distributed arbitration scheme that can be used with an obsolete bus scheme known as Multibus I. Agents are daisy-chained physically in priority order. The left-most agent in the diagram receives a constant *bus priority in* (BPRN) signal indicating that no higher-priority agent desires the bus. If the agent does not require the bus, it asserts its *bus priority out* (BPRO) line. At the beginning of a clock cycle, any agent can request control of the bus by lowering its BPRO line. This lowers the BPRN line of the next agent in the chain, which is in turn required to lower its BPRO line. Thus, the signal is propagated the length of the chain. At the end of this chain reaction, there should be only one agent whose BPRN is asserted and whose BPRO is not. This agent has priority. If, at the beginning of a bus cycle, the bus is not busy (BUSY inactive), the agent that has priority may seize control of the bus by asserting the BUSY line.

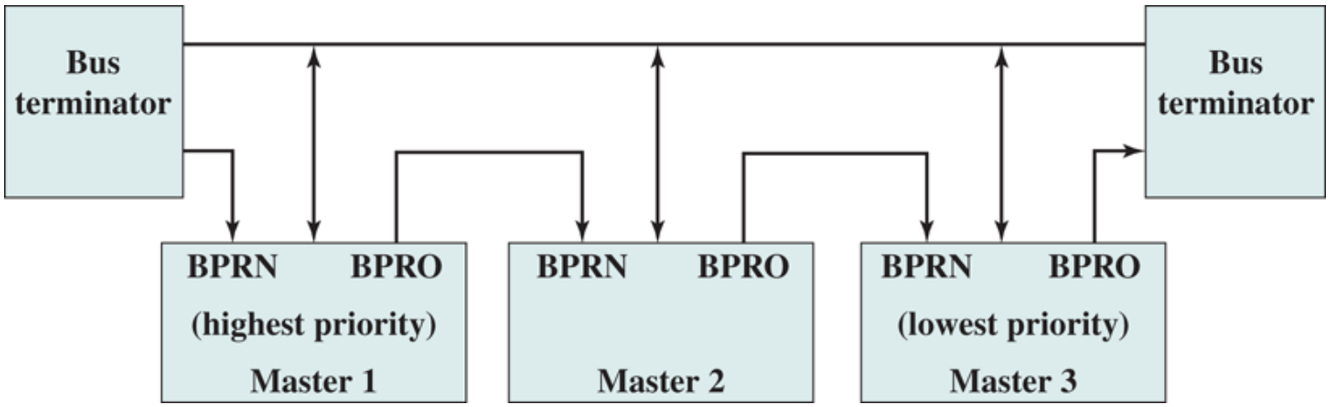


Figure 3.26 Multibus I Distributed Arbitration

It takes a certain amount of time for the BPR signal to propagate from the highest-priority agent to the lowest. Must this time be less than the clock cycle? Explain.

3.9 The VAX SBI bus uses a distributed, synchronous arbitration scheme. Each SBI device (i.e., processor, memory, I/O module) has a unique priority and is assigned a unique transfer request (TR) line. The SBI has 16 such lines (TR0, TR1, . . ., TR15), with TR0 having the highest priority. When a device wants to use the bus, it places a reservation for a future time slot by asserting its TR line during the current time slot. At the end of the current time slot, each device with a pending reservation examines the TR lines; the highest-priority device with a reservation uses the next time slot.

A maximum of 17 devices can be attached to the bus. The device with priority 16 has no TR line. Why not?

3.10 On the VAX SBI, the lowest-priority device usually has the lowest average wait time. For this reason, the processor is usually given the lowest priority on the SBI. Why does the priority 16 device usually have the lowest average wait time? Under what circumstances would this not be true?

3.11 For a synchronous read operation (Figure C.3 in **Appendix C**), the memory module must place the data on the bus sufficiently ahead of the falling edge of the Read signal to allow for signal settling. Assume a microprocessor bus is clocked at 10 MHz and that the Read signal begins to fall in the middle of the second half of T_3.

a. Determine the length of the memory read instruction cycle.
b. When, at the latest, should memory data be placed on the bus? Allow 20 ns for the settling of data lines.

3.12 Consider a microprocessor that has a memory read timing (Figure C.3 in **Appendix C**). After some analysis, a designer determines that the memory falls short of providing read data on time by about 180 ns.

a. How many wait states (clock cycles) need to be inserted for proper system operation if the bus clocking rate is 8 MHz?
b. To enforce the wait states, a Ready status line is employed. Once the processor has issued a Read command, it must wait until the Ready line is asserted before attempting to read data. At what time interval must we keep the Ready line low in order to force the processor to insert the required number of wait states?

3.13 A microprocessor has a memory write timing Figure A.3 in **Appendix A** . Its manufacturer specifies that the width of the Write signal can be determined by $T - 50$, where *T* is the clock period in ns.

a. What width should we expect for the Write signal if bus clocking rate is 5 MHz?
b. The data sheet for the microprocessor specifies that the data remain valid for 20 ns after the falling edge of the Write signal. What is the total duration of valid data presentation to memory?
c. How many wait states should we insert if memory requires valid data presentation for at least 190 ns?

3.14 A microprocessor has an increment memory direct instruction, which adds 1 to the value in a memory location. The instruction has five stages: fetch opcode (four bus clock cycles), fetch operand address (three cycles), fetch operand (three cycles), add 1 to operand (three cycles), and store operand (three cycles).

a. By what amount (in percent) will the duration of the instruction increase if we have to insert two bus wait states in each memory read and memory write operation?
b. Repeat assuming that the increment operation takes 13 cycles instead of 3 cycles.

3.15 The Intel 8088 microprocessor has a read bus timing similar to that of Figure C.3, but requires four processor clock cycles. The valid data is on the bus for an amount of time that extends into the fourth processor clock cycle. Assume a processor clock rate of 8 MHz.

a. What is the maximum data transfer rate?
b. Repeat, but assume the need to insert one wait state per byte transferred.

3.16 The Intel 8086 is a 16-bit processor similar in many ways to the 8-bit 8088. The 8086 uses a 16-bit bus that can transfer 2 bytes at a time, provided that the lower-order byte has an even address. However, the 8086 allows both even- and odd-aligned word operands. If an odd-aligned word is referenced, two memory cycles, each consisting of four bus cycles, are required to transfer the word. Consider an instruction on the 8086 that involves two 16-bit operands. How long does it take to fetch the operands? Give the range of possible answers. Assume a clocking rate of 4 MHz and no wait states.

3.17 Consider a 32-bit microprocessor whose bus cycle is the same duration as that of a 16-bit microprocessor. Assume that, on average, 20% of the operands and instructions are 32 bits long, 40% are 16 bits long, and 40% are only 8 bits long. Calculate the improvement achieved when fetching instructions and operands with the 32-bit microprocessor.

3.18 The microprocessor of Problem 3.14 initiates the fetch operand stage of the increment memory direct instruction at the same time that a keyboard activates an interrupt request line. After how long does the processor enter the interrupt processing cycle? Assume a bus clocking rate of 10 MHz.

Chapter 4 The Memory Hierarchy: Locality and Performance

Learning Objectives

After studying this chapter, you should be able to:

- Present an overview of the principle of locality.
- Describe key characteristics of a memory system.
- Discuss how locality influences the development of a memory hierarchy.
- Understand the performance implications of multiple levels of memory.

Although seemingly simple in concept, computer memory exhibits perhaps the widest range of type, technology, organization, performance, and cost of any feature of a computer system. No single technology is optimal in satisfying the memory requirements for a computer system. As a consequence, the typical computer system is equipped with a hierarchy of memory subsystems, some internal to the system (directly accessible by the processor) and some external (accessible by the processor via an I/O module).

This chapter focuses on the performance factors that drive the development of a computer memory system with multiple levels using different technologies. ***Section 4.1*** *introduces the key concept of locality of reference, which has a profound influence on both the organization of memory and on operating system memory management software. Following a brief discussion of key characteristics of memory systems, the chapter turns to a presentation of the concept of a memory hierarchy and indicates the typical components in contemporary systems. Finally,* ***Section 4.4*** *develops a simple but illuminating model of memory access performance.*

The next three chapters look at specific aspects of memory systems, using the insights provided in this chapter. Chapter 5 examines an essential element of all

4.5 Key Terms, Review Questions, and Problems

Key Terms

access time

addressable unit

associative memory

auxiliary memory

cache memory

coherence

data spatial locality

data temporal locality

direct access

dynamic instruction

hit ratio

horizontal coherence inclusion

instruction cache

instruction spatial locality

instruction temporal locality

L1 cache

L2 cache

L3 cache

L4 cache

locality

locality of reference

memory hierarchy

memory cycle time

multilevel cache

multilevel memory

random access

secondary memory

sequential access

spatial locality

static instruction

temporal locality

transfer rate

unit of transfer

vertical coherence

word

Review Questions

4.1 What are the differences among sequential access, direct access, and random access?
4.2 What is the general relationship among access time, memory cost, and capacity?
4.3 How does the principle of locality relate to the use of multiple memory levels?
4.4 What is the distinction between spatial locality and temporal locality?
4.5 In general, what are the strategies for exploiting spatial locality and temporal locality?
4.6 How do data locality and instruction locality relate to spatial locality and temporal locality?

Problems

4.1 Consider these terms: instruction spatial locality, instruction temporal locality, data spatial locality, data temporal locality. Match each of these terms to one of the following definitions:

a. Locality is quantified by computing the average distance (in terms of number of operand memory accesses) between two consecutive accesses to the same address, for every unique address in the program. The evaluation is performed in four distinct window sizes, analogous to cache block sizes.
b. Locality metric is quantified by computing the average distance (in terms of number of instructions) between two consecutive accesses to the same static instruction, for every unique static instruction in the program that is executed at least twice.
c. Locality for operand memory accesses is characterized by the ratio of the locality metric for window sizes mentioned in (a).
d. Locality is characterized by the ratio of the locality metric for the window sizes mentioned in (b).

4.2 Consider these two programs:

`for ( i = 1; i < n; i++) {`
`  Z[i] = X[i] - Y[i]`
`  Z[i] = Z[i] * Z[i]`
`}`
	`for ( i = 1; i < n; i++) {`
	`  Z[i] = X[i] - Y[i]`
	`}`
	`for ( i = 1; i < n; i++) {`
	`  Z[i] = Z[i] * Z[i]`
	`}`

Program A	Program B

a. The two programs perform the same function. Describe it.
b. Which version performs better, and why?

4.3 Consider the following code:

```
for (i = 0; i < 20; i++)
   for (j = 0; j < 10; j++)
      a[i] = a[i] * j
```

a. Give one example of the spatial locality in the code.
b. Give one example of the temporal locality in the code.

4.4 Consider a memory system with the following parameters:

$T_c = 100\text{ns}$	$C_c = 10^{-4}\$ / \text{bit}$
$T_m = 1200\text{ns}$	$C_m = 10^{-5}\$ / \text{bit}$

a. What is the cost of 1 MB of main memory?
b. What is the cost of 1 MB of main memory using cache memory technology?
c. If the effective access time is 10% greater than the cache access time, what is the hit ratio *H*?

4.5
a. Consider an L1 cache with an access time of 1 ns and a hit ratio of $H = 0.95$. Suppose that we can change the cache design (size of cache, cache organization) such that we increase *H* to 0.97, but increase access time to 1.5 ns. What conditions must be met for this change to result in improved performance?
b. Explain why this result makes intuitive sense.

4.6 Consider a single-level cache with an access time of 2.5 ns, a block size of 64 bytes, and a hit ratio of $H = 0.95$. Main memory uses a block transfer capability that has a first-word (4 bytes) access time of 50 ns and an access time of 5 ns for each word thereafter.
a. What is the access time when there is a cache miss? Assume that the cache waits until the line has been fetched from main memory and then re-executes for a hit.
b. Suppose that increasing the block size to 128 bytes increases the H to 0.97. Does this reduce the average memory access time?

4.7 A computer has a cache, main memory, and a disk used for virtual memory. If a referenced word is in the cache, 20 ns are required to access it. If it is in main memory but not in the cache, 60 ns are needed to load it into the cache, and then the reference is started again. If the word is not in main memory, 12 ns are required to fetch the word from the disk, followed by 60 ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time in nanoseconds required to access a referenced word on this system?
4.8 On the Motorola 68020 microprocessor, a cache access takes two clock cycles. Data access from main memory over the bus to the processor takes three clock cycles in the case of no wait state insertion; the data are delivered to the processor in parallel with delivery to the cache.

a. Calculate the effective length of a memory cycle given a hit ratio of 0.9 and a clocking rate of 16.67 MHz.
b. Repeat the calculations assuming insertion of two wait states of one cycle each per memory cycle. What conclusion can you draw from the results?

4.9 Assume a processor having a memory cycle time of 300 ns and an instruction processing rate of 1 MIPS. On average, each instruction requires one bus memory cycle for instruction fetch and one for the operand it involves.
a. Calculate the utilization of the bus by the processor.
b. Suppose that the processor is equipped with an instruction cache and the associated hit ratio is 0.5. Determine the impact on bus utilization.

4.10 The performance of a single-level cache system for a read operation can be characterized by the following equation:

$$T_a = T_c + (1 - H)\,T_m$$

where T_a is the average access time, T_c is the cache access time, T_m is the memory access time (memory to processor register), and *H* is the hit ratio. For simplicity, we assume that the word in question is loaded into the cache in parallel with the load to processor register. This is the same form as **Equation (4.2)** .
a. Define $T_b = \text{time}$ to transfer a block between cache and main memory, and $W = \text{fraction}$ of write references. Revise the preceding equation to account for writes as well as reads, using a write-through policy.
b. Define W_b as the probability that a block in the cache has been altered. Provide an equation for T_a for the write-back policy.

4.11 For a system with two levels of cache, define $T_{c1} = \text{first-level}$ cache access time; $T_{c2} = \text{second-level}$ cache access time; $T_m = \text{memory}$ access time; $H_1 = \text{first-level}$ cache hit ratio; $H_2 = \text{combined}$ first/second level cache hit ratio. Provide an equation for T_a for a read operation.

4.12 Assume the following performance characteristics on a cache read miss: one clock cycle to send an address to main memory and four clock cycles to access a 32-bit word from main memory and transfer it to the processor and cache.
a. If the cache block size is one word, what is the miss penalty (i.e., additional time required for a read in the event of a read miss)?
b. What is the miss penalty if the cache block size is four words and a multiple, nonburst transfer is executed?
c. What is the miss penalty if the cache block size is four words and a transfer is executed, with one clock cycle per word transfer?

4.13 For the cache design of the preceding problem, suppose that increasing the line size from one word to four words results in a decrease of the read miss rate from 3.2% to 1.1%. For both the nonburst transfer and the burst transfer case, what is the average miss penalty, averaged over all reads, for the two different line sizes?

4.14 Consider a two-level system with L1 instruction and data caches. For a given application, assume the following: instruction cache miss $\text{ratio} = 0.02$, data cache miss $\text{ratio} = 0.04$, and the fraction of instructions that are $\text{load / store} = 0.36$. The ideal value of CPI (cycles per instruction) without cache misses is 2.0. The penalty for a cache miss is 40 cycles. Calculate the CPI, taking misses into account.

4.15 Define $H_i = \text{probability}$ that the data for a memory access is resident in level M_i.

a. **Equation 4.5** uses the conditional probabilities h_i. Explain why in this form the equation is correct with the conditional probabilities rather than the unconditional probabilities H_i. That is, show that the following expression does not equal T_s.

$$\sum_{i-1}^{i} \prod_{j=1}^{1} (1 - h_j)\, h_I \times t_I$$

b. Rewrite **Equation 4.5** using H_i instead of h_i.

4.16 Define the access frequency f_i as the probability of successfully accessing (hit) M_i when there are misses at the preceding $i - 1$ levels.

a. Derive an expression for f_i.
b. Rewrite **Equation 4.5** using f_i instead of h_i.

5.6 Key Terms, Review Questions, and Problems

Key Terms

associative mapping

cache block

cache hit

cache line

cache memory

cache miss

cache set

content-addressable memory

critical word first

data cache

direct mapping

dirty bit

frame

instruction cache

line

line size

logical cache

multilevel cache

no write allocate

physical cache

replacement algorithm

set-associative mapping

split cache

tag

unified cache

use bit

victim cache

virtual cache

write allocate

write back

write through

Review Questions

5.1 What are the differences among direct mapping, associative mapping, and set-associative mapping?
5.2 What is the difference between associative cache memory and content-addressable memory?
5.3 For a direct-mapped cache, a main memory address is viewed as consisting of three fields. List and define the three fields.
5.4 For an associative cache, a main memory address is viewed as consisting of two fields. List and define the two fields.
5.5 For a set-associative cache, a main memory address is viewed as consisting of three fields. List and define the three fields.
5.6 What is the distinction between spatial locality and temporal locality?
5.7 In general, what are the strategies for exploiting spatial locality and temporal locality?

Problems

5.1 A cache has a line size of 64 bytes. To determine which byte within a cache line an address points to, how many bits are in the Offset field?
5.2 A set-associative cache consists of 64 lines, or slots, divided into four-line sets. Main memory contains 4K blocks of 128 words each. Show the format of main memory addresses.
5.3 A two-way set-associative cache has lines of 16 bytes and a total size of 8 kB. The 64-MB main memory is byte addressable. Show the format of main memory addresses.
5.4 For the hexadecimal main memory addresses 111111, 666666, BBBBBB, show the following information, in hexadecimal format:

a. Tag, Line, and Word values for a direct-mapped cache, using the format of **Figure 5.7**
b. Tag and Word values for an associative cache, using the format of **Figure 5.10**
c. Tag, Set, and Word values for a two-way set-associative cache, using the format of **Figure 5.13**

5.5 List the following values:

a. For the direct cache example of **Figure 5.7** : address length, number of addressable units, block size, number of blocks in main memory, number of lines in cache, size of tag
b. For the associative cache example of **Figure 5.10** : address length, number of addressable units, block size, number of blocks in main memory, number of lines in cache, size of tag
c. For the two-way set-associative cache example of **Figure 5.13** : address length, number of addressable units, block size, number of blocks in main memory, number of lines in set, number of sets, number of lines in cache, size of tag

5.6 Consider a 32-bit microprocessor that has an on-chip 16-kB four-way set-associative cache. Assume that the cache has a line size of four 32-bit words. Draw a block diagram of this cache showing its organization and how the different address fields are used to determine a cache hit/miss. Where in the cache is the word from memory location ABCDE8F8 mapped?
5.7 Given the following specifications for an external cache memory: four-way set associative; line size of two 16-bit words; able to accommodate a total of 4K 32-bit words from main memory; used with a 16-bit processor that issues 24-bit addresses. Design the cache structure

with all pertinent information and show how it interprets the processor's addresses.

5.8 The Intel 80486 has an on-chip, unified cache. It contains 8 kB and has a four-way set-associative organization and a block length of four 32-bit words. The cache is organized into 128 sets. There is a single "line valid bit" and three bits, B0, B1, and B2 (the "LRU" bits), per line. On a cache miss, the 80486 reads a 16-byte line from main memory in a bus memory read burst. Draw a simplified diagram of the cache and show how the different fields of the address are interpreted.

5.9 Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

a. How is a 16-bit memory address divided into tag, line number, and byte number?
b. Into what line would bytes with each of the following addresses be stored?

0001	0001	0001	1011
1100	0011	0011	0100
1101	0000	0001	1101
1010	1010	1010	1010

c. Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?
d. How many total bytes of memory can be stored in the cache?
e. Why is the tag also stored in the cache?

5.10 For its on-chip cache, the Intel 80486 uses a replacement algorithm referred to as **pseudo least recently used**. Associated with each of the 128 sets of four lines (labeled L0, L1, L2, L3) are three bits B0, B1, and B2. The replacement algorithm works as follows: When a line must be replaced, the cache will first determine whether the most recent use was from L0 and L1 or L2 and L3. Then the cache will determine which of the pair of blocks was least recently used and mark it for replacement. **Figure 5.19** illustrates the logic.

a. Specify how the bits B0, B1, and B2 are set and then describe in words how they are used in the replacement algorithm depicted in **Figure 5.19** .
b. Show that the 80486 algorithm approximates a true LRU algorithm. *Hint:* Consider the case in which the most recent order of usage is L0, L2, L3, L1.
c. Demonstrate that a true LRU algorithm would require 6 bits per set.

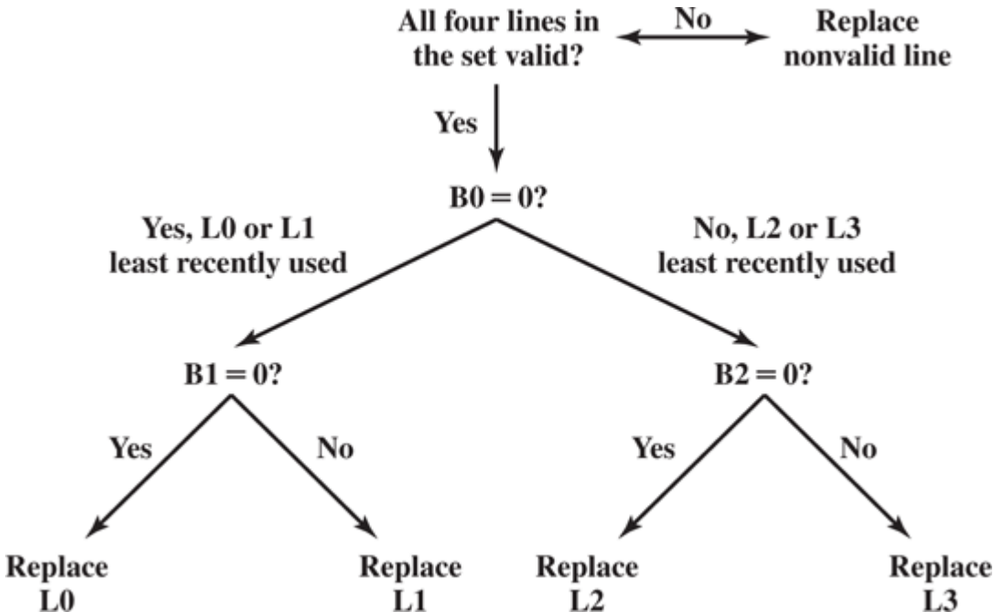


Figure 5.19 Intel 80486 On-Chip Cache Replacement Strategy

5.11 A set-associative cache has a block size of four 16-bit words and a set size of 2. The cache can accommodate a total of 4096 words. The main memory size that is cacheable is $64\text{K} \times 32$ bits. Design the cache structure and show how the processor's addresses are interpreted.

5.12 Consider a memory system that uses a 32-bit address to address at the byte level, plus a cache that uses a 64-byte line size.

a. Assume a direct mapped cache with a tag field in the address of 20 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.
b. Assume an associative cache. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.
c. Assume a four-way set-associative cache with a tag field in the address of 9 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in set, number of sets in cache, number of lines in cache, size of tag.

5.13 Consider a computer with the following characteristics: total of 1 MB of main memory; word size of 1 byte; block size of 16 bytes; and cache size of 64 kB.

a. For the main memory addresses of F0010, 01234, and CABBE, give the corresponding tag, cache line address, and word offsets for a direct-mapped cache.
b. Give any two main memory addresses with different tags that map to the same cache slot for a direct-mapped cache.
c. For the main memory addresses of F0010 and CABBE, give the corresponding tag and offset values for a fully-associative cache.
d. For the main memory addresses of F0010 and CABBE, give the corresponding tag, cache set, and offset values for a two-way set-associative cache.

5.14 Describe a simple technique for implementing an LRU replacement algorithm in a four-way set-associative cache.

5.15 Consider again **Example 5.2** . How does the answer change if the main memory uses a block transfer capability that has a first-word access time of 30 ns and an access time of 5 ns for each word thereafter?
A computer system contains a main memory of 32K 16-bit words. It also has a 4K word cache divided into four-line sets with 64 words per line. Assume that the cache is initially empty. The processor fetches words from locations 0, 1, 2, . . . , 4351 in that order. It then repeats this fetch sequence nine more times. The cache is 10 times faster than main memory. Estimate the improvement resulting from the use of the cache. Assume an LRU policy for block replacement.
5.16 Consider a cache of 4 lines of 16 bytes each. Main memory is divided into blocks of 16 bytes each. That is, block 0 has bytes with addresses 0 through 15, and so on. Now consider a program that accesses memory in the following sequence of addresses:
Once: 63 through 70.
Loop ten times: 15 through 32; 80 through 95.

a. Suppose the cache is organized as direct mapped. Memory blocks 0, 4, and so on are assigned to line 1; blocks 1, 5, and so on to line 2; and so on. Compute the hit ratio.
b. Suppose the cache is organized as two-way set associative, with two sets of two lines each. Even-numbered blocks are assigned to set 0 and odd-numbered blocks are assigned to set 1. Compute the hit ratio for the two-way set-associative cache using the least recently used replacement scheme.

5.17 Consider a cache with a line size of 64 bytes. Assume that on average 30% of the lines in the cache are dirty. A word consists of 8 bytes.

a. Assume there is a 3% miss rate (0.97 hit ratio). Compute the amount of main memory traffic, in terms of bytes per instruction for both write-through and write-back policies. Memory is read into cache one line at a time. However, for write back, a single word can be written from cache to main memory.
b. Repeat part a for a 5% rate.
c. Repeat part a for a 7% rate.
d. What conclusion can you draw from these results?

5.18 The level below a cache in the memory hierarchy requires 60 ns to read or write a word of data. If the cache line size is 8 words, how many times does the average line have to be written (counting only lines that are written at least once) before a write-back cache is more efficient than a write-through cache?

6.7 Key Terms, Review Questions, and Problems

Key Terms

bank group

double data rate DRAM (DDR DRAM)

dynamic RAM (DRAM)

electrically erasable programmable ROM (EEPROM)

erasable programmable ROM (EPROM)

error correcting code (ECC)

error correction

flash memory

Hamming code

hard failure

magnetic RAM (MRAM)

NAND flash memory

nonvolatile memory

NOR flash memory

phase-change RAM (PCRAM)

programmable ROM (PROM)

random access memory (RAM)

read-mostly memory

read-only memory (ROM)

resistive RAM (ReRAM)

semiconductor memory

single-error-correcting (SEC) code

single-error-correcting, double-error-detecting (SEC-DED) code

soft error

spin-transfer torque RAM (STT-RAM)

static RAM (SRAM)

synchronous DRAM (SDRAM)

syndrome

timing diagram

volatile memory

Review Questions

6.1 What are the key properties of semiconductor memory?
6.2 What are two interpretations of the term *random-access memory*?
6.3 What is the difference between DRAM and SRAM in terms of application?
6.4 What is the difference between DRAM and SRAM in terms of characteristics such as speed, size, and cost?
6.5 Explain why one type of RAM is considered to be analog and the other digital.
6.6 What are some applications for ROM?
6.7 What are the differences among EPROM, EEPROM, and flash memory?
6.8 Explain the function of each pin in **Figure 5.4b** .
6.9 What is a parity bit?
6.10 How is the syndrome for the Hamming code interpreted?
6.11 How does SDRAM differ from ordinary DRAM?
6.12 What is DDR RAM?
6.13 What is the difference between NAND and NOR flash memory?
6.14 List and briefly define three newer nonvolatile solid-state memory technologies.

Problems

6.1 Suggest reasons why RAMs traditionally have been organized as only one bit per chip whereas ROMs are usually organized with multiple bits per chip.
6.2 Consider a dynamic RAM that must be given a refresh cycle 64 times per ms. Each refresh operation requires 150 ns; a memory cycle requires 250 ns. What percentage of the memory's total operating time must be given to refreshes?
6.3 **Figure 6.22** shows a simplified timing diagram for a DRAM read operation over a bus. The access time is considered to last from t_1 to t_2. Then there is a recharge time, lasting from t_2 to t_3, during which the DRAM chips will have to recharge before the processor can access them again.

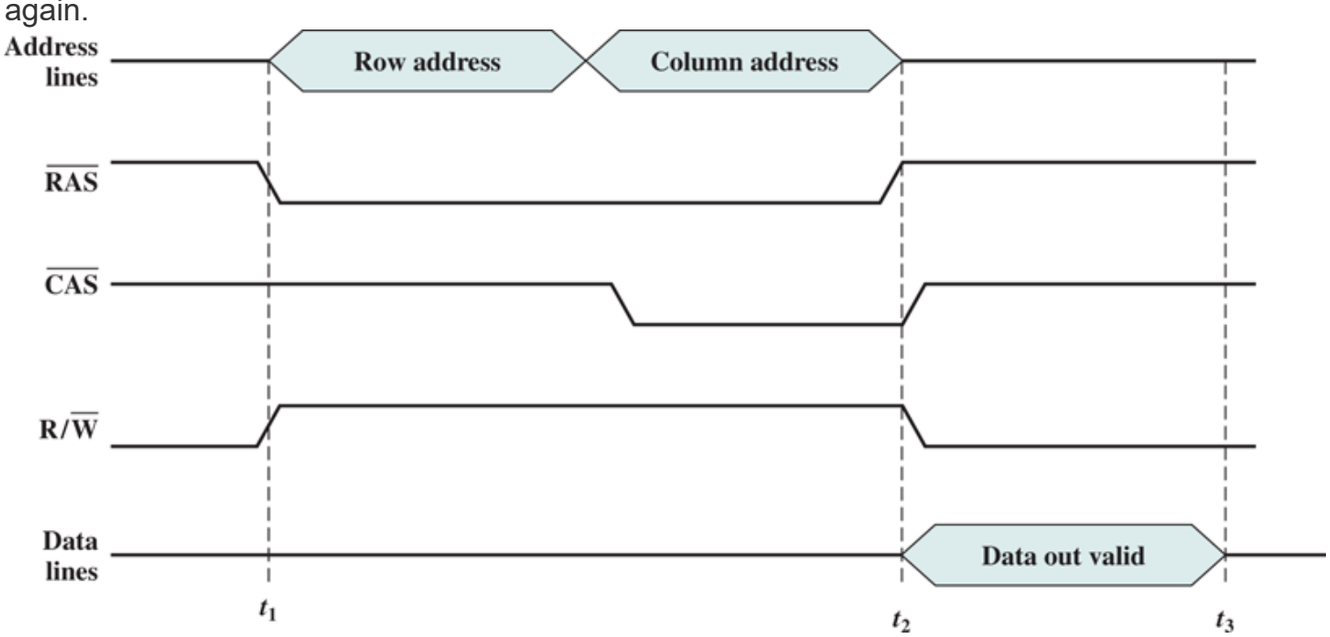


Figure 6.22 Simplified DRAM Read Timing

a. Assume that the access time is 60 ns and the recharge time is 40 ns. What is the memory cycle time? What is the maximum data rate this DRAM can sustain, assuming a 1-bit output?
b. Constructing a 32-bit wide memory system using these chips yields what data transfer rate?

6.4 **Figure 6.6** indicates how to construct a module of chips that can store 1 MB based on a group of four 256-Kbyte chips. Let's say this module of chips is packaged as a single 1-MB chip, where the word size is 1 byte. Give a high-level chip diagram of how to construct an 8-MB computer memory using eight 1-MB chips. Be sure to show the address lines in your diagram and what the address lines are used for.

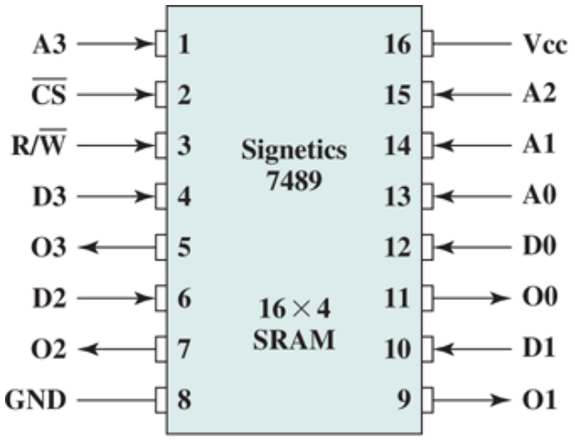
6.5 On a typical Intel 8086-based system, connected via system bus to DRAM memory, for a read operation, $\overline{\text{RAS}}$ is activated by the trailing edge of the Address Enable signal (**Figure A.1** in **Appendix A**). However, due to propagation and other delays, $\overline{\text{RAS}}$ does not go active until 50 ns after Address Enable returns to a low. Assume the latter occurs in the middle of the second half of state T_1 (somewhat earlier than in **Figure A.1**). Data are read by the processor at the end of T_3. For timely presentation to the processor, however, data must be provided 60 ns earlier by memory. This interval accounts for propagation delays along the data paths (from memory to processor) and processor data hold time requirements. Assume a clocking rate of 10 MHz.

a. How fast (access time) should the DRAMs be if no wait states are to be inserted?
b. How many wait states do we have to insert per memory read operation if the access time of the DRAMs is 150 ns?

6.6 The memory of a particular microcomputer is built from $64\text{K} \times 1\text{DRAMs}$. According to the data sheet, the cell array of the DRAM is organized into 256 rows. Each row must be refreshed at least once every 4 ms. Suppose we refresh the memory on a strictly periodic basis.

a. What is the time period between successive refresh requests?
b. How long a refresh address counter do we need?

6.7 **Figure 6.23** shows one of the early SRAMs, the 16×4 Signetics 7489 chip, which stores 16 4-bit words.

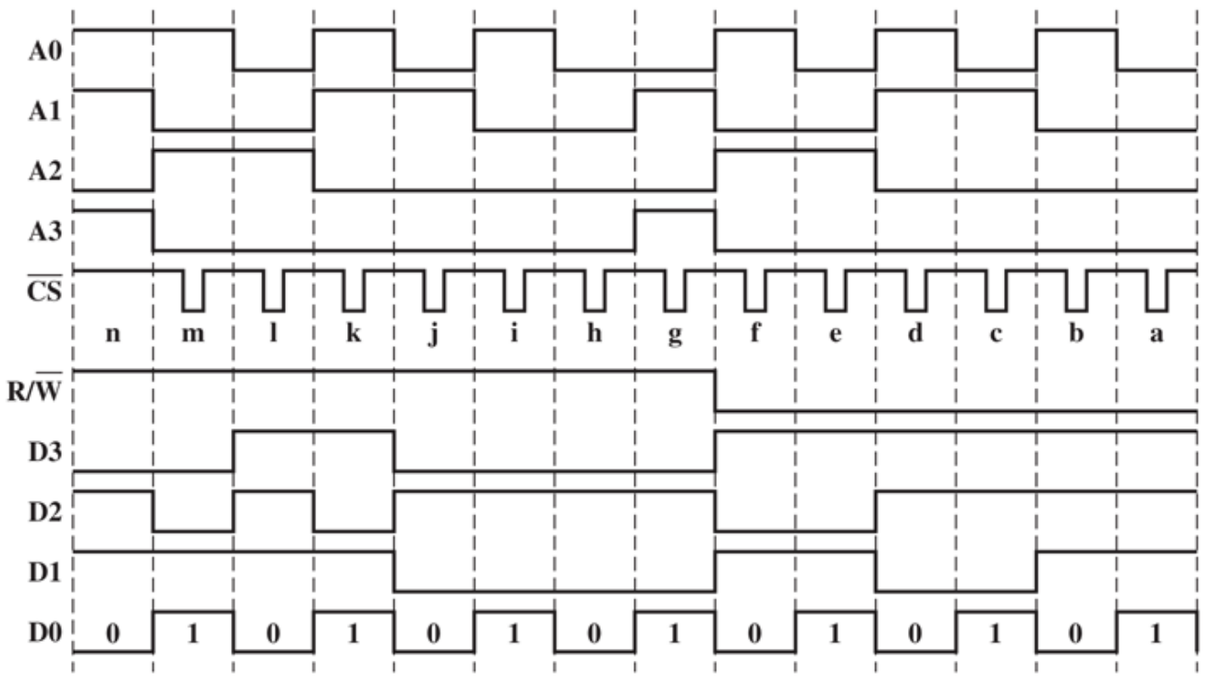


(a) Pin layout

Operating Mode	Inputs			Outputs
	$\overline{CS}$	R/$\overline{W}$	D*n*	O*n*
Write	L	L	L	L
	L	L	H	H
Read	L	H	X	Data
Inhibit writing	H	L	L	H
	H	L	H	L
Store - disable outputs	H	H	X	H

H = high voltage level
L = low voltage level
X = don't care

(b) Truth table



(c) Pulse train

Figure 6.23 The Signetics 7489 SRAM

a. List the mode of operation of the chip for each CS input pulse shown in **Figure 6.23c** .
b. List the memory contents of word locations 0 through 6 after pulse n.
c. What is the state of the output data leads for the input pulses h through m?

6.8 Design a 16-bit memory of total capacity 8192 bits using SRAM chips of size 64×1 bit. Give the array configuration of the chips on the memory board showing all required input and output signals for assigning this memory to the lowest address space. The design should allow for both byte and 16-bit word accesses.

6.9 A common unit of measure for failure rates of electronic components is the **Failure unIT** (FIT), expressed as a rate of failures per billion device hours. Another well known but less used measure is **mean time between failures (MTBF),** which is the average time of operation of a particular component until it fails. Consider a 1 MB memory of a 16-bit microprocessor with $256\text{K} \times 1$ DRAMs. Calculate its MTBF assuming 2000 FITS for each DRAM.

6.10 For the Hamming code shown in **Figure 6.10** , show what happens when a check bit rather than a data bit is in error?

6.11 Suppose an 8-bit data word stored in memory is 11000010. Using the Hamming algorithm, determine what check bits would be stored in memory with the data word. Show how you got your answer.

6.12 For the 8-bit word 00111001, the check bits stored with it would be 0111. Suppose when the word is read from memory, the check bits are calculated to be 1101. What is the data word that was read from memory?

6.13 How many check bits are needed if the Hamming error correction code is used to detect single bit errors in a 1024-bit data word?

6.14 Develop an SEC code for a 16-bit data word. Generate the code for the data word 0101000000111001. Show that the code will correctly identify an error in data bit 5.