CHAPTER 2

2.1 This program is developed in [HAYE98]. The vectors A, B, and C are each stored in 1,000 contiguous locations in memory, beginning at locations 1001, 2001, and 3001, respectively. The program begins with the left half of location 3. A counting variable N is set to 999 and decremented after each step until it reaches –1. Thus, the vectors are processed from high location to low location.

Location	Instruction	Comments
0	999	Constant (count N)
1	1	Constant
2	1000	Constant
3L	LOAD M(2000)	Transfer A(I) to AC
3R	ADD M(3000)	Compute A(I) + B(I)
4L	STOR M(4000)	Transfer sum to C(I)
4R	LOAD M(0)	Load count N
5L	SUB M(1)	Decrement N by 1
5R	JUMP+ M(6, 20:39)	Test N and branch to 6R if nonnegative
6L	JUMP M(6, 0:19)	Halt
6R	STOR M(0)	Update N
7L	ADD M(1)	Increment AC by 1
7R	ADD M(2)	
8L	STOR M(3, 8:19)	Modify address in 3L
8R	ADD M(2)	
9L	STOR M(3, 28:39)	Modify address in 3R
9R	ADD M(2)	
10L	STOR M(4, 8:19)	Modify address in 4L
10R	JUMP M(3, 0:19)	Branch to 3L

2.4

Address	Contents
08A	LOAD M(0FA)
	STOR M(0FB)
08B	LOAD M(0FA)
	JUMP +M(08D)
08C	LOAD –M(0FA)
	STOR M(0FB)
08D	

This program will store the absolute value of content at memory location 0FA into memory location 0FB.

2.7 The discrepancy can be explained by noting that other system components aside from clock speed make a big difference in overall system speed. In particular, memory systems and advances in I/O processing contribute to the performance ratio. A system is only as fast as its slowest link. In recent years, the bottlenecks have been the performance of memory modules and bus speed.

2.8 As noted in the answer to Problem 2.7, even though the Intel machine may have a faster clock speed (2.4 GHz vs. 1.2 GHz), that does not necessarily mean the system will perform faster. Different systems are not comparable on clock speed. Other factors such as the system components (memory, buses, architecture) and the instruction sets must also be taken into account. A more accurate measure is to run both systems on a benchmark. Benchmark programs exist for certain tasks, such as running office applications, performing floating point operations, graphics operations, and so on. The systems can be compared to each other on how long they take to complete these tasks. According to Apple Computer, the G4 is comparable or better than a higher-clock speed Pentium on many benchmarks.

2.9 This representation is wasteful because to represent a single decimal digit from 0 through 9 we need to have ten tubes. If we could have an arbitrary number of these tubes ON at the same time, then those same tubes could be treated as binary bits. With ten bits, we can represent 2^{10} patterns, or 1024 patterns. For integers, these patterns could be used to represent the numbers from 0 through 1023.

2.13 *CPI* = 1.55; MIPS rate = 25.8; Execution time = 3.87 ns. Source: [HWAN93]

2.14 a. Ultimately, the user is concerned with the execution time of a system, not its execution rate. If we take arithmetic mean of the MIPS rates of various benchmark programs, we get a result that is proportional to the sum of the inverses of execution times. But this is not inversely proportional to the sum of execution times. In other words, the arithmetic mean of the MIPS rate does not cleanly relate to execution time. On the other hand, the harmonic mean MIPS rate is the inverse of the average execution time.

b.

	Arithmetic mean	**Harmonic Mean**	**Rank**
Computer A	25.3 MIPS	0.25 MIPS	2
Computer B	2.8 MIPS	0.21 MIPS	3
Computer C	3.25 MIPS	2.1 MIPS	1

CHAPTER 3

3.6 a. Input from the Teletype is stored in INPR. The INPR will only accept data from the Teletype when FGI=0. When data arrives, it is stored in INPR, and FGI is set to 1. The CPU periodically checks FGI. If FGI =1, the CPU transfers the contents of INPR to the AC and sets FGI to 0.

When the CPU has data to send to the Teletype, it checks FGO. If FGO = 0, the CPU must wait. If FGO = 1, the CPU transfers the contents of the AC to OUTR and sets FGO to 0. The Teletype sets FGI to 1 after the word is printed.

b. The process described in **(a)** is very wasteful. The CPU, which is much faster than the Teletype, must repeatedly check FGI and FGO. If interrupts are used, the

Teletype can issue an interrupt to the CPU whenever it is ready to accept or send data. The IEN register can be set by the CPU (under programmer control)

3.8 The whole point of the clock is to define event times on the bus; therefore, we wish for a bus arbitration operation to be made each clock cycle. This requires that the priority signal propagate the length of the daisy chain (Figure 3.26) in one clock period. Thus, the maximum number of masters is determined by dividing the amount of time it takes a bus master to pass through the bus priority by the clock period.

3.11 **a.** With a clocking frequency of 10 MHz, the clock period is 10^{-9} s = 100 ns. The length of the memory read cycle is 300 ns.

b. The Read signal begins to fall at 75 ns from the beginning of the third clock cycle (middle of the second half of T_3). Thus, memory must place the data on the bus no later than 55 ns from the beginning of T_3. Source: [PROT88]

3.12 **a**. The clock period is 125 ns. Therefore, two clock cycles need to be inserted.

b. From Figure 3.19, the Read signal begins to rise early in T_2. To insert two clock cycles, the Ready line can be put in low at the beginning of T_2 and kept low for 250 ns. Source: [PROT88]

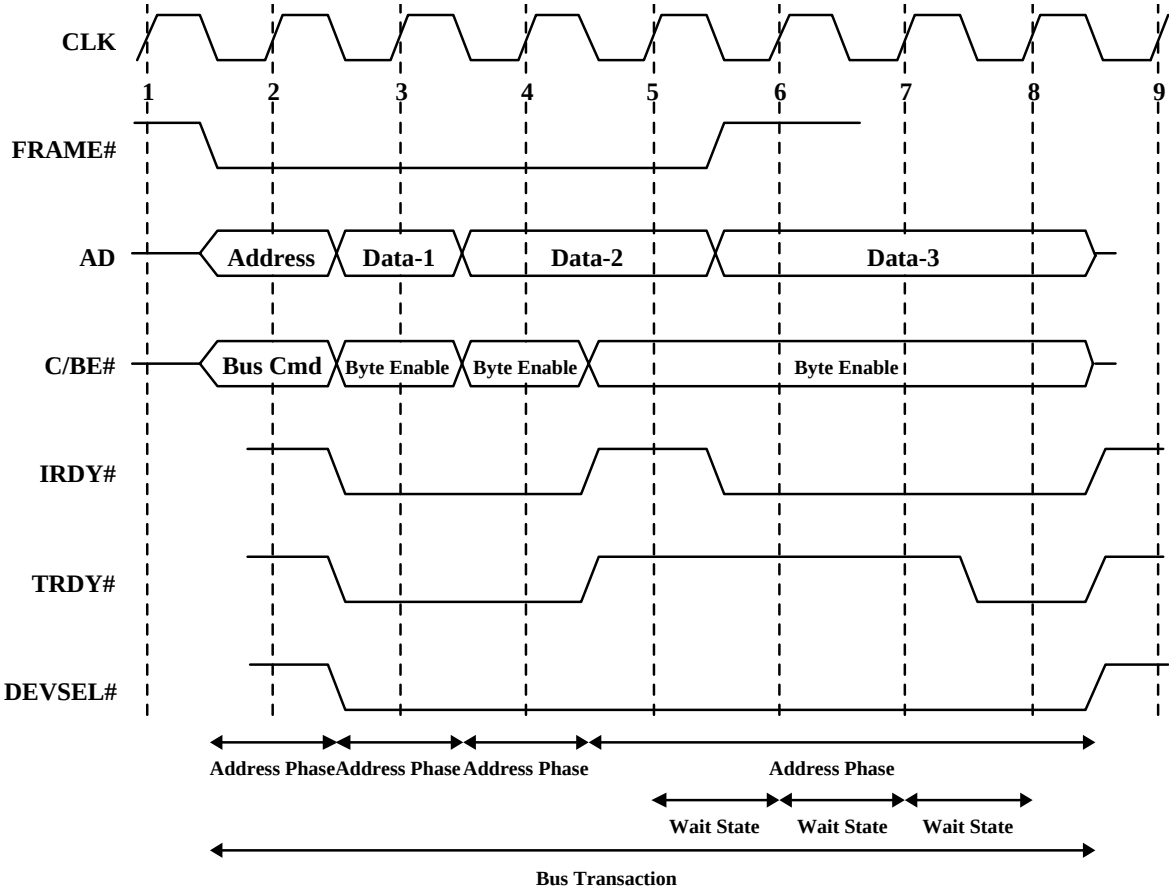
3.13 **a.** A 5 MHz clock corresponds to a clock period of 200 ns. Therefore, the Write signal has a duration of 150 ns.

b. The data remain valid for 150 + 20 = 170 ns.

c. One wait state. Source: [PROT88]

3.17 Consider a mix of 100 instructions and operands. On average, they consist of 20 32-bit items, 40 16-bit items, and 40 bytes. The number of bus cycles required for the 16-bit microprocessor is $(2 \times 20) + 40 + 40 = 120$. For the 32-bit microprocessor, the number required is 100. This amounts to an improvement of 20/120 or about 17%. Source: [PROT88].

3.19


CLK
1
2
3
4
5
6
7
8
9
FRAME#
AD
Address
Data-1
Data-2
Data-3
C/BE#
Bus Cmd
Byte Enable
Byte Enable
Byte Enable
IRDY#
TRDY#
DEVSEL#
Address Phase
Address Phase
Address Phase
Address Phase
Wait State
Wait State
Wait State
Bus Transaction