Model Answers HW2 - Chapter #3

1. **The hypothetical machine of figure 3.4 also has two I/O instructions:**
    0011= Load AC fro 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 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 received from device 5 is 3 and that location 940 contains value of 2.
Answer:

	Memory
300	3 0 0 5
301	5 9 4 0
302	7 0 0 6
.	
.	
940	0 0 0 2
941	

We will assume that the memory (contents in hex) as the previous table:
300: 3005; 301: 5940; 302: 7006
Therefore, the steps will be as the following:
Step 1: 3005 → IR
Step 2: 3 → AC
Step 3: 5940 → IR
Step 4: 3 + 2 = 5 → AC
Step 5: 7006 → IR
Step 6: AC → Device 6

----

2. 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. a 32-bit local address bus and a 16-bit local data bus, or**
>   **2. a 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?

Answer:
a. $2^{(32-8)} = 2^{24} = 16,777,216$ bytes = 16 MB   ,(8 bits = 1 byte for he opcode).
b.1. a 32-bit local address bus and a 16-bit local data bus. Instruction and data transfers would take three bus cycles each, one for the address and two for the data. Since If the address bus is 32 bits, the whole address can be transferred to memory at once and decoded there; however, since the data bus is only 16 bits, it will require 2 bus cycles (accesses to memory) to fetch the 32-bit instruction or operand.

b.2. a 16-bit local address bus and a 16-bit local data bus. Instruction and data transfers would take four bus cycles each, two for the address and two for the data. Therefore, that will have the processor perform two transmissions in order to send to memory the whole 32-bit address; this will require more complex memory interface control to latch the two halves of the address before it performs an access to it. In

addition to this two-step address issue, since the data bus is also 16 bits, the microprocessor will need 2 bus cycles to fetch the 32-bit instruction or operand.

c. For the PC needs 24 bits (24-bit addresses), and for the IR needs 32 bits (32-bit addresses).

----

3. 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.
Answer:
a. The Maximum memory address space $= 2^{16} = 64$ Kbytes.

b. The Maximum memory address space $= 2^{16} = 64$ Kbytes.

Therefore, in (a) and (b), the microprocessor is to access 64K bytes, but the difference thing between them is that the access of 8-bit memory will transfer a 8 bits and the access of 16-bit memory may transfer 8 bits or 16 bits word.

c. Separate I/O instructions are needed because during its execution will generate separate its own signals I/O signals. That signals will be different from the memory signals which is generated during the execution for memory instructions. Therefore, one more output pin will be needed to carry I/O signals.

d. With an 8-bit I/O port number the microprocessor can support $2^8 = 256$ 8-bit input ports,
and $2^8 = 256$ 8-bit output ports.
With an 8-bit I/O port number the microprocessor can support $2^8 = 256$ 16-bit input ports,
and $2^8 = 256$ 16-bit output ports.

Thus, the size of the I/O port will not change the number of I/O ports since the number of I/O ports depends on the number of bits which is used to represent the I/O port number (equals to 8 bits in both cases).

----

4. 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/s? 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.
Answer:
Since minimum bus cycle duration = 4 clock cycles and bus clock = 8 MHz
Then, maximum bus cycle rate = 8 M / 4 = 2 M/s

Data transferred per bus cycle = 16 bit = 2 bytes
Data transfer rate per second = bus cycle rate * data per bus cycle = 2 M * 2 = 4 Mbytes/sec.

To increase its performance:
- By doubling the frequency, it may mean adopting a new chip manufacturing technology (assuming each instruction will have the same number of clock cycles);
- By doubling the external data bus, that means wider (may be newer) on-chip data bus drivers/latches and modifications to the bus control logic.

Therefore, in the first situation the speed of the memory chips will need to double, not to slow down the microprocessor. Regarding the second situation, the word length of the memory will must double to be able to send/receive 32-bit quantities.

----

5. 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.
Answer:
   a. Through a single bus cycle, the 8-bit microprocessor transfers one byte while the 16-bit microprocessor transfers two bytes. The 16-bit microprocessor has twice the data transfer rate.
   b. By assuming that we have to perform 50 transfers of operands and instructions which 25 are one byte long and 25 are two bytes long. The 8-bit microprocessor needs 25 + (2 x 25) = 75 bus cycles for the transfer. The 16-bit microprocessor needs 25 + 25 = 50 bus cycles. Therefore, the data transfer rates differ by a factor of 1.5.

----

6. 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?**

Answer:
The lowest-priority device has priority 16. That device have to defer to all the others. Therefore, it may transmit in any slot not reserved by the other SBI devices.

----

7. 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?
Answer:
From the beginning for any slot, only the priority 16 device may transmit when none of the TR lines is asserted. This provides it the lowest average wait time under most circumstances. We could say that when there is much demand on the bus and most of the time having at least one pending request, that will prevent the priority 16 device to have the lowest average wait time.

----

8. For a synchronous read operation (Figure 3.19), 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.
Answer:
a. Since the clocking frequency is 10 MHz, the clock period is 10^{-9} s = 100 ns. Therefore, the length of the memory read instruction cycle equals to 300 ns since the Read signal begins to fall in the middle of the second half of T_3.
.

b. Also, since the Read signal begins to fall in the middle of the second half of T_3, the Read signal begins to fall at 75 ns from the beginning of T_3. Therefore, the memory have to copy or put the data on the bus before 55 ns from the beginning of T_3.

----

9. Consider a microprocessor that has a memory read timing as shown in (Figure 3.19). 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?
Answer:
a. Actually, two clock cycles has be inserted proper system operation since the clock period is 125 ns.
b. We can notice from Figure 3.19 that the Read signal starts at the beginning of T_2. Therefore, to insert two clock cycles which is needed it, the Read line could be put in low at the beginning of T_2 and still like that in low for 125 ns + 125 ns = 250 ns.

----

10. A microprocessor has a memory write timing as shown in (Figure 3.19). 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?
Answer:
a. Since the clocking rate is 5 MHz, the clock period equals to 200 ns. Thus, the duration for the Write signal equals to T − 50 = 200 − 50 = 150 ns.
b. In fact, the data will remain valid for 150 + 20 = 170 ns.
c. Only one wait state for at least 190 ns.

----

11. 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.
Answer:
a. If the case has no wait states, the instruction requires around 16 bus clock cycles since it needs for:
    Four bus clock cycles to fetch opcode
    Three bus clock cycles to fetch operand address
    Three bus clock cycles to fetch operand
    Three bus clock cycles to add 1 to operand
    Three bus clock cycles to store operand

Therefore, 8 wait states will be required since the instruction needs four memory accesses.
If the case has wait states, the instruction takes 16 + 8 = 24 clock cycles which is increased to 8/16 = 50%

b. By assuming that (13 cycles to add 1 to operand) instead of (3 cycles to add 1 to operand), when there is no wait states, the instruction needs 26 bus cycles. When the case has wait states, the instruction needs 26 + 8 = 34 bus cycles which is increased to 8/26 = 31%.

----

12. The Intel 8088 microprocessor has a read bus timing similar to that of Figure 3.19, 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.
Answer:
a. Since The clock period is 125 ns. Thus, for each one read cycle is required 125 * 4 = 500 ns = 0.5 μs. If the bus cycles will be repeated respectively, the data transfer rate = 1/ 0.5 = 2 MB/s.
b. By assuming there is one wait state, the bus read cycle will be increased 125 ns. Thus, for each one read cycle is required 125 * 500 = 625 ns = 0.625 μs. Finally, the data transfer rate = 1/ 0.625 = 1.6 MB/s.

----

13. 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.
Answer:
Since the clocking rate 4 MHz, the bus cycle will need 0.25 μs. Thus, the memory cycle will take 0.25* 4 = 1 μs.
By applying this <u>"If an odd-aligned word is referenced, two memory cycles, each consisting of four bus cycles, are required to transfer the word"</u> from the question, so we will have 3 cases:
First case is if both operands are even-aligned, so the time required is 1*2 = 2 μs to fetch both operands.
Second case is if both operands are odd-aligned, so the time required is 1*4 = 4 μs to fetch both operands.
Third case is if only one is odd-aligned, so the time required is 1*3 = 3 μs to fetch both operands.

----

14. 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.

Answer:

By assuming that we have a mix of 100 instructions and operands.

From the question:

      20% of the operands and instructions are 32-bits long, so it is 20 32-bit.

      40% of the operands and instructions are 16-bits long, so it is 40 16-bit.

      40% of the operands and instructions are only 8-bits = 1 byte long, so it is 40 bytes.

The number of bus cycles needed for the 16-bit microprocessor will equal to:

$$(20 * 2) + 40 + 40 = 120 \text{ bus cycles.}$$

The number of bus cycles needed for the 32-bit microprocessor will equal to:

$$20 + 40 + 40 = 100 \text{ bus cycles.}$$

By calculating the improvement achieved with the 32-bit microprocessor to the 16-bit microprocessor will equal to 20/120 = 16.6%.

----

15. The microprocessor of (Problem 11) initiates the fetch operand stage of the increment memory direct instruction at the same time that a keyboard actives an interrupt request line. After how long does the processor enter the interrupt processing cycle? Assume a bus clocking rate of 10 MHz.

Answer:

Since the clocking rate is 10 MHz, the clock period is 10^{-9} s = 100 ns. Since the microprocessor of initiates the fetch operand stage of the increment memory direct instruction at the same time that a keyboard actives an interrupt request line, so no need for this:

      Four bus clock cycles to fetch opcode

      Three bus clock cycles to fetch operand address

Therefore, the processor requires nine clock cycles to keep executing the instruction. Actually, the Interrupt begins after 9*100 = 900 ns done.