**3.1** Memory (contents in hex): 300: 3005; 301: 5940; 302: 7006

**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

**3.3 a.** $2^{24}$ = 16 MBytes

**b.** **(1)** If the local address bus is 32 bits, the whole address can be transferred at once and decoded in memory. However, because the data bus is only 16 bits, it will require 2 cycles to fetch a 32-bit instruction or operand.

**(2)** The 16 bits of the address placed on the address bus can't access the whole memory. Thus a more complex memory interface control is needed to latch the first part of the address and then the second part (because the microprocessor will end in two steps). For a 32-bit address, one may assume the first half will decode to access a "row" in memory, while the second half is sent later to access a "column" in memory. In addition to the two-step address operation, the microprocessor will need 2 cycles to fetch the 32-bit instruction/operand.

**c.** The program counter must be at least 24 bits. Typically, a 32-bit microprocessor will have a 32-bit external address bus and a 32-bit program counter, unless on-chip segment registers are used that may work with a smaller program counter. If the instruction register is to contain the whole instruction, it will have to be 32-bits long; if it will contain only the op code (called the op code register) then it will have to be 8 bits long.

**3.4** In cases **(a)** and **(b)**, the microprocessor will be able to access $2^{16}$ = 64K bytes; the only difference is that with an 8-bit memory each access will transfer a byte, while with a 16-bit memory an access may transfer a byte or a 16-byte word. For case **(c)**, separate input and output instructions are needed, whose execution will generate separate "I/O signals" (different from the "memory signals" generated with the execution of memory-type instructions); at a minimum, one additional output pin will be required to carry this new signal. For case **(d)**, it can support $2^8$ = 256 input and $2^8$ = 256 output byte ports and the same number of input and output 16-bit ports; in either case, the distinction between an input and an output port is defined by the different signal that the executed input or output instruction generated.

**3.13 a.** A 8 MHz clock corresponds to a clock period of 125 ns. Therefore, the Write signal has a duration of 93.75 ns. A 12 MHz clock corresponds to a clock period of 83.33 ns. Therefore, the Write signal has a duration of 62.49 ns.

**b.** In case of 8 MHz clock, the data remains valid for 93.75 + 15 = 108.75 ns.

In case of 12 MHz clock, the data remains valid for 62.49 + 15 = 77.49 ns.

**c.** In case of 8 MHz clock, no wait state should be inserted. In case of 12 MHz clock, one wait state should be inserted.

**3.14 a.** Without the wait states, the instruction takes 18 bus clock cycles. The instruction requires four memory accesses, resulting in 4 wait states. The instruction, with wait states, takes 22 clock cycles, for an increase of 22%.

**b.** In this case, the instruction takes 22 bus cycles without wait states and 26 bus cycles with wait states, for an increase of 18%.

**3.15 a.** When processor clock rate is 20 MHz, the clock period is 50 ns. One bus read cycle takes 200 ns = 0.2 μs. If the bus cycles repeat one after another, we can achieve a data transfer rate of 5 MB/s. When processor clock rate is 50 MHz, the clock period is 20 ns. One bus read cycle takes 80 ns = 0.08 μs. If the bus cycles repeat one after another, we can achieve a data transfer rate of 12.5 MB/s.

**b.** When processor clock rate is 20 MHz, the wait state extends the bus read cycle by 50 ns, for a total duration of 0.25 μs. The corresponding data transfer rate is 1/0.25 = 4 MB/s.
When processor clock rate is 50 MHz, the wait state extends the bus read cycle by 20 ns, for a total duration of 0.1 μs. The corresponding data transfer rate is 1/0.1 = 10 MB/s.