**13.1** Given the following memory values and a one-address machine with an accumulator, what values do the following instructions load into the accumulator?

- Word 20 contains 40.
- Word 30 contains 50.
- Word 40 contains 60.
- Word 50 contains 70.

20 40
30 50
40 60
50 70

a. LOAD IMMEDIATE 20 20
b. LOAD DIRECT 20 40
c. LOAD INDIRECT 20 60
d. LOAD IMMEDIATE 30 30
e. LOAD DIRECT 30 50
f. LOAD INDIRECT 30 70

**13.3** An address field in an instruction contains decimal value 14. Where is the corresponding operand located for

a. immediate addressing?
b. direct addressing?
c. indirect addressing?
d. register addressing?
e. register indirect addressing?

address field | 14

a. the address field
b. memory location 14
c. the memory location whose address is in memory location 14.
d. register 14
e. memory location whose address is in register 14.

**13.4** Consider a 16-bit processor in which the following appears in main memory, starting at location 200:

399 999
400 1000

| | | |
|---|---|---|
| 200 | Load to AC | Mode |
| 201 | 500 | |
| 202 | Next instruction | |

The first part of the first word indicates that this instruction loads a value into an accumulator. The Mode field specifies an addressing mode and, if appropriate, indicates a source register; assume that when used, the source register is R1, which has a value of 400. There is also a base register that contains the value 100. The value of 500 in location 201 may be part of the address calculation. Assume that location 399 contains the value 999, location 400 contains the value 1000, and so on. Determine the effective address and the operand to be loaded for the following address modes:

**a.** Direct
**b.** Immediate
**c.** Indirect
**d.** PC relative
**e.** Displacement
**f.** Register
**g.** Register indirect
**h.** Autoindexing with increment, using R1

**13.7** How many times does the processor need to refer to memory when it fetches and executes an indirect-address-mode instruction if the instruction is (a) a computation requiring a single operand; (b) a branch?

(a) 4 times (b) 2 times

**14.1** **a.** If the last operation performed on a computer with an 8-bit word was an addition in which the two operands were 00000010 and 00000011, what would be the value of the following flags?

- Carry
- Zero
- Overflow
- Sign
- Even Parity
- Half-Carry

**b.** Repeat for the addition of −1 (twos complement) and +1.

a. 00000010
+ 00000011
00000101

0, 0, 0, 0, 1, 0

b. 11111111
+ 00000001
100000000

1, 1, 0, 0, 1, 1

**14.2** Repeat Problem 14.1 for the operation A−B, where A contains 11110000 and B contains 0010100.

$\overline{B}$ 11101011

$\overline{B}$+1 11101100

11110000
+ 11101100
111011100

1, 0, 0, 0, 0, 0

$5\times10^9$

**14.3** A microprocessor is clocked at a rate of 5 GHz.

$\frac{1}{5\times10^9} = 0.2\,ns$

**a.** How long is a clock cycle?

**b.** What is the duration of a particular type of machine instruction consisting of three clock cycles?

$0.6\,ns$

**14.4** A microprocessor provides an instruction capable of moving a string of bytes from one area of memory to another. The fetching and initial decoding of the instruction takes 10 clock cycles. Thereafter, it takes 15 clock cycles to transfer each byte. The microprocessor is clocked at a rate of 10 GHz.

**a.** Determine the length of the instruction cycle for the case of a string of 64 bytes.

**b.** What is the worst-case delay for acknowledging an interrupt if the instruction is noninterruptible?

**c.** Repeat part (b) assuming the instruction can be interrupted at the beginning of each byte transfer.

| FI | DI | EI |
|---|---|---|
| 10 | | 15 |

64
+15
960

a. $\frac{1}{10\times10^9} = 0.1\,ns$

$10 + 15\times64 = 960$ $\quad 960\times0.1\,ns = 96\,ns$

b. $15\times64\times0.1 = 96\,ns$

c. $15\times0.1 = 1.5\,ns$

**14.5** The Intel 8088 consists of a bus interface unit (BIU) and an execution unit (EU), which form a 2-stage pipeline. The BIU fetches instructions into a 4-byte instruction queue. The BIU also participates in address calculations, fetches operands, and writes results in memory as requested by the EU. If no such requests are outstanding and the bus is free, the BIU fills any vacancies in the instruction queue. When the EU completes execution of an instruction, it passes any results to the BIU (destined for memory or I/O) and proceeds to the next instruction.

**a.** Suppose the tasks performed by the BIU and EU take about equal time. By what factor does pipelining improve the performance of the 8088? Ignore the effect of branch instructions.

**b.** Repeat the calculation assuming that the EU takes twice as long as the BIU.

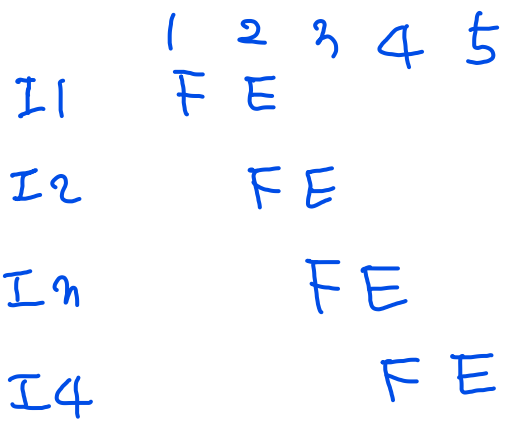
a. non-pipe line $\quad EU+BIU = 2t \quad t+2t = 3t$

pipe line $\quad t \quad 2t$

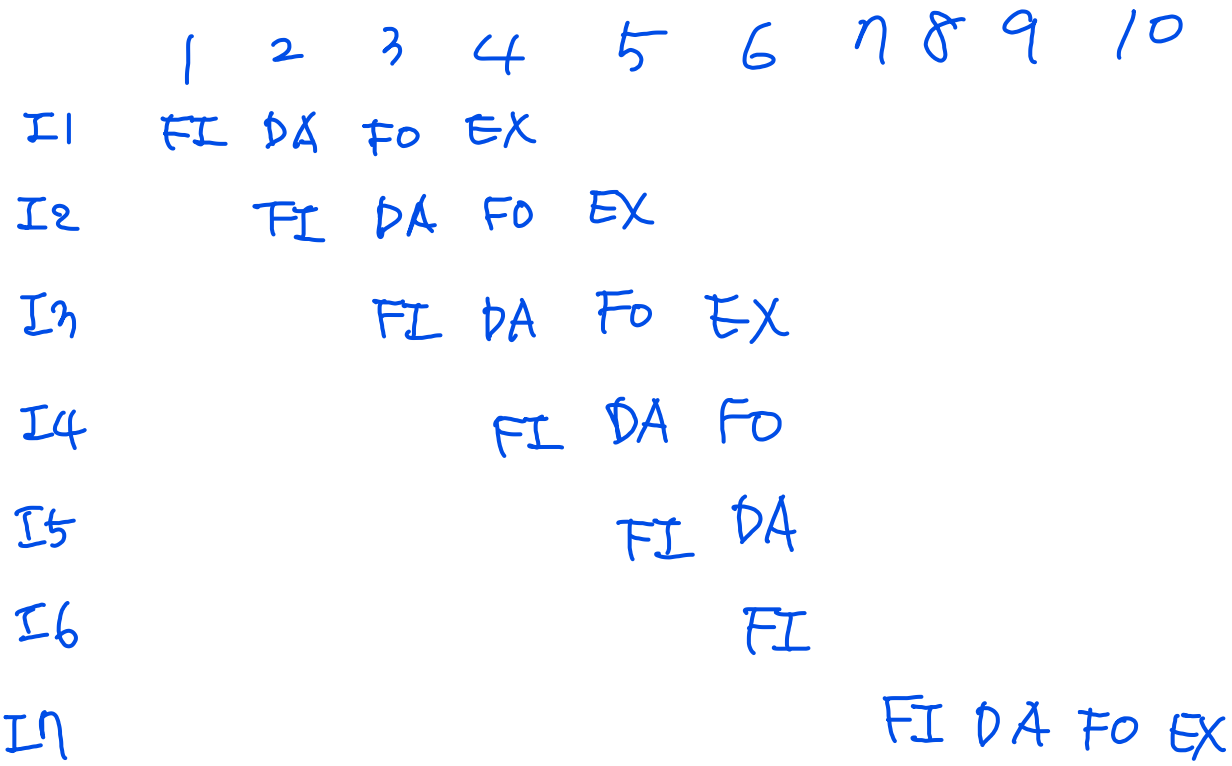
Speed up $\quad 3t/t = 2 \quad 1.5$

**14.7** Consider the timing diagram of Figures 14.10. Assume that there is only a two-stage pipeline (fetch, execute). Redraw the diagram to show how many time units are now needed for four instructions.

F E




**14.8** Assume a pipeline with four stages: fetch instruction (FI), decode instruction and calculate addresses (DA), fetch operand (FO), and execute (EX). Draw a diagram similar to Figure 14.10 for a sequence of 7 instructions, in which the third instruction is a branch that is taken and in which there are no data dependencies.



**14.9** A pipelined processor has a clock rate of 2.5 GHz and executes a program with 1.5 million instructions. The pipeline has five stages, and instructions are issued at a rate of one per clock cycle. Ignore penalties due to branch instructions and out-of-sequence executions.

**a.** What is the speedup of this processor for this program compared to a nonpipelined processor, making the same assumptions used in Section 14.4?

**b.** What is throughput (in MIPS) of the pipelined processor?

a. speed up $= \frac{kn}{k+n-1} = \frac{5 \times 1.5 \times 10^6}{5 + 1.5 \times 10^6 - 1} \approx 5$

b. MIPS = $2.5\text{GHz} / 10^6 = 2500$ MIPS.

**14.11** Consider an instruction sequence of length $n$ that is streaming through the instruction pipeline. Let $p$ be the probability of encountering a conditional or unconditional branch instruction, and let $q$ be the probability that execution of a branch instruction I causes a jump to a nonconsecutive address. Assume that each such jump requires the pipeline to be cleared, destroying all ongoing instruction processing, when I emerges from the last stage. Revise Equations (14.1) and (14.2) to take these probabilities into account.