Page 183 Exercises 1.2, 1.3, 1.4, 1.5

1. 1.2
    A. Assembly Lines in Automobile manufacturing.
        a. Performance via pipelining.
    B. Suspension bridge cables.
        a. Performance via parallelism.
    C. Aircraft and marine navigation systems that incorporate wind information.
        a. Performance via prediction.
    D. Express elevators in buildings.
        a. Make the common case fast.
    E. Library reserve desk.
        a. Hierarchy of memories.
    F. Increasing the gate area on a CMOS transistor to decrease its switching time.
        a. Dependability via redundancy.
    G. Building self-driving cars whose control systems partially rely on existing sensor systems already installed into the base vehicle, such as lane departure systems and smart cruise control systems.
        a. Use Abstraction to simplify design.
2. 1.3
    A. First Step:

a. A Compiler takes the C program, and translates it into Assembly Language program.

B. Second Step:

a. An Assembler takes the Assembly language program, and translates it into machine language (1s and 0s).

C. Third Step:

a. An executable file is then created and can be run on the specified machine in which the language was written for.

3. 1.4

A. 1280x1024 = 1310720 pixels

a. 1310720x3 = 3932160 bytes

B.

a. 3932160 Bytes = 3.93 MB

b. 100 MBits/sec = 12.5 MB/s

c. 3.93 MB/frame x (1/12.5 MB/s) = 0.3145728 sec

4. 1.5

A. P1: 3GHz / 1.5 = 2 * 10^9 instructions per second.

P2: 2.5GHz / 1.0 = 2.5 * 10^9 instructions per second

P3: 4GHz / 2.2 = 1.82 * 10^9 instructions per second

Therefore: P2 has the highest performance among the three.

B. Cycles:

P1: 3GHz * 10 = 3 * 10^10 cycles

P2: 2.5GHz * 10 = 2.5 * 10^10 cycles

P3: 4GHz * 10 = 4 * 10^10 cycles

Num of instructions:

P1: 3GHz * 10 / 1.5 = 2 * 10^10 instructions

P2: 2.5GHz * 10 / 1.0 = 2.5 * 10^10 instructions

P3: 4GHz * 10 / 2.2 = 1.82 * 10^10 instructions.

C.

- Execution time = (Num of instructions * CPI) / (Clock rate).
- So if we want to reduce the execution time by 30%, and CPI increases by 20%, we have:
- Execution time * 0.7 = (Num of instructions * CPI * 1.2) / (New Clock rate).
- New Clock rate = Clock rate * 1.2 / 0.7 = 1.71 * Clock rate
- New Clock rate for each processor:
  - P1: 3GHz * 1.71 = 5.13 GHz
  - P2: 2.5GHz * 1.71 = 4.27 GHz
  - P3: 4GHz * 1.71 = 6.84 GHz