1.3. The C code is processed by the preprocessor and then the compiler to create assembly code. This assembly code is then sent to the assembler and linker to turn into machine code. Finally this is loaded into the memory and program counter.

1.4. a. $1280 \times 1024 \times 8 \times 8 \times 8$

$= 1280 \times 1024 \times 3$ bytes $= 3932160$ bytes
≈ 3.9 MB

b. $\frac{3932160 \times 8}{100\,000\,000} = 0.31$ secs

1.5. a. P1 $= 3/1.5 = 2$ G ips

P2 $= 2.5/1 = 2.5$ G ips

P3 $= 4/2.2 = 1.8$ G ips

b. P1 $= 3 \times 10 = 30$ G cycles $/1.5 = 20$ G instructions

P2 $= 2.5 \times 10 = 25$ G cycles $/1 = 25$ G instructions

P3 $= 4 \times 10 = 40$ G cycles $/2.2 = 18$ G instructions

c. $\text{exectime} \times 0.7 = \frac{\#\text{inst} \cdot 1.2 \cdot \text{CPI}}{\text{clkrate}}$

$\text{exectime} = \frac{1.2 \text{ cycles}}{0.7 \text{ clkrt}} \approx 1.71 \frac{\text{cycles}}{\text{clkrt}}$

So increase clkrt by 1.71

1.6. a. P1 $= (0.1) + (0.2 \cdot 2) + (0.5 \cdot 3) + (0.2 \cdot 3) = 2.6$ cpi

P2 $= (0.1 \cdot 2) + (0.2 \cdot 2) + (0.5 \cdot 2) + (0.2 \cdot 2) = 2$ cpi

b. P1 $= 1E6 \, \cancel{\text{inst}} \cdot \frac{2.6 \text{ cyc}}{\cancel{\text{inst}}} = 2.6$ M clkcycles

P2 $= 1E6 \, \cancel{\text{inst}} \cdot \frac{2 \text{ cyc}}{\cancel{\text{inst}}} = 2$ M clk cycles

1.7. a. $A = \frac{1.1 \cdot 1\,GHz}{1\ G\,instrs} = 1.1$ cpi

$\frac{\text{exec time} \times \text{clk rate}}{\#\ \text{inst}} = \text{cpi}$

$B = \frac{1.5 \cdot 1\,GHz}{1.2\ G\,instrs} = 1.25$ cpi

b. $\frac{1\cancel{E9} \cdot 1.1}{\text{clk rate}} = \frac{1.2\cancel{E9} \cdot 1.5}{\text{clk rate}}$

$\frac{1.8}{1.1} = 1.63\times$ faster

c. $C = \text{cpi} \cdot \text{inst} = 0.6\,G\,ins \cdot 1.1 = 0.66\,G$

$A = 1.1\,G$

$B = 1.25 \cdot 1.2 = 1.5\,G$

$\frac{A}{C} = 1.67\times$ faster than A

$\frac{B}{C} = 2.27\times$ faster than B