1.1 Aside from the smart cell phones used by a billion people, list and describe four other types of computers.

- **Personal computers** were built over 35 years ago. Used in home and business applications. Generally, the speed and computing power are high compared to smartphones. Personal computers deliver good results for single users.
- **Servers** are used for controlling groups of computers and are usually accessed only via a network. They are generally multi-user, multi-tasking systems.
- **Supercomputers** consist of groups of processors that can perform high computing within seconds. Generally, they are used in high-end scientific and engineering applications like NASA.
- **Embedded Computers** consist of a wide range of applications. Embedded computers mean integrating the processor with hardware.

1.3 Describe the steps that transform a program written in a high-level language such as C into a representation directly executed by a computer processor.

- Compiling is converting the source code of the C language into machine code. Since C is a mid-level language, it needs a compiler to convert it into an executable code to run the program on a machine.

1.4 Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 × 1024

A) What is the minimum size in bytes of the frame buffer to store a frame?
   - Total bits per pixel = 8 bits (red) + 8 bits (green) + 8 bits (blue) = 24 bits per pixel.
     Total bits per frame = Total bits per pixel × Number of pixels per frame
     Total bits per frame = 24 bits/pixel × (1280 pixels × 1024 pixels) = 24 bits/pixel × 1,310,720 pixels = 31,337,280 bits.
     Total bytes per frame = Total bits per frame / 8 = 31,337,280 bits / 8 = 3,917,160 bytes.

B) How long would it take, at a minimum, for the frame to be sent over a 100Mbit/s network?
   - Frame size in bits = Total bits per frame = 31,337,280 bits
   - Network speed = 100 Mbit/s = 100,000,000 bits/s
   - Time (in seconds) to send the frame = 31,337,280 bits / 100,000,000 bits/s ≈ 0.313 seconds.

1.5 Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3GHz clock rate and a CPI of 1.5. P2 has a 2.5GHz clock rate and a CPI of 1.0. P3 has a 4.0GHz clock rate and has a CPI of 2.2.

A) Which processor has the highest performance expressed in instructions per second?
   - For P1: Clock Rate = 3 GHz = 3 * 10^9 Hz CPI = 1.5
   - MIPS for P1 = (3 * 10^9 Hz) / (1.5) * 10^-6 = 2,000 MIPS
   - For P2: Clock Rate = 2.5 GHz = 2.5 * 10^9 Hz CPI = 1.0
   - MIPS for P2 = (2.5 * 10^9 Hz) / (1.0) * 10^-6 = 2,500 MIPS
   - For P3: Clock Rate = 4.0 GHz = 4.0 * 10^9 Hz CPI = 2.2
   - MIPS for P3 = (4.0 * 10^9 Hz) / (2.2) * 10^-6 ≈ 1,818.18 MIPS

P2 has the highest performance expressed in instructions per second with 2,500 MIPS.

B) If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
   - For P1: Execution Time = 10 seconds Clock Rate = 3 GHz = 3 * 10^9 Hz CPI = 1.5
   - Cycles for P1 = (10 seconds) * (3 * 10^9 Hz) = 30 * 10^9 cycles
   - Instructions for P1 = (30 * 10^9 cycles) / 1.5 = 20 * 10^9 instructions
   - For P2: Execution Time = 10 seconds Clock Rate = 2.5 GHz = 2.5 * 10^9 Hz
   - CPI = 1.0
   - Cycles for P2 = (10 seconds) * (2.5 * 10^9 Hz) = 25 * 10^9 cycles
   - Instructions for P2 = (25 * 10^9 cycles) / 1.0 = 25 * 10^9 instructions
   - For P3: Execution Time = 10 seconds Clock Rate = 4.0 GHz = 4.0 * 10^9 Hz
   - CPI = 2.2
   - Cycles for P3 = (10 seconds) * (4.0 * 10^9 Hz) = 40 * 10^9 cycles
   - Instructions for P3 = (40 * 10^9 cycles) / 2.2 ≈ 18.18 * 10^9 instructions

C) We are trying to reduce the execution time by 30%, which leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?
   - New Clock Rate = Old Clock Rate * (1 + Change in CPI) / (1 - Change in Execution Time)
   - Change in CPI = 20% (0.20 increase)
   - Change in Execution Time = -30% (-0.30 decrease)
   - New Clock Rate ≈ 1.714 * Old Clock Rate

1.9 Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also, assume that on a single processor, a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2GHz clock frequency.

1.9.1
Total Execution Time (T_1) for 1 processor = Total instructions * CPI_single / Clock Frequency = (3.84E9) * ((2/3) * 1 + (1/3) * 12 + (1/15) * 5) / (2E9) = 19.2 seconds

Total Execution Time (T_2) for 2 processors = Total instructions * CPI_parallel / (Clock Frequency * 2) = (3.84E9) * ((2/3) * 1 + (1/3) * 12 + (1/15) * 5) / (2E9 * 2) = 9.6 seconds

Total Execution Time (T_4) for 4 processors = Total instructions * CPI_parallel / (Clock Frequency * 4) = (3.84E9) * ((2/3) * 1 + (1/3) * 12 + (1/15) * 5) / (2E9 * 4) = 4.8 seconds

Total Execution Time (T_8) for 8 processors = Total instructions * CPI_parallel / (Clock Frequency * 8) = (3.84E9) * ((2/3) * 1 + (1/3) * 12 + (1/15) * 5) / (2E9 * 8) = 2.4 seconds

1.9.2
Impact on Execution Time for 1 processor = (T_1_new - T_1_original) / T_1_original = (28.8 - 19.2) / 19.2 ≈ 0.5 (50% increase)

Impact on Execution Time for 2 processors = (T_2_new - T_2_original) / T_2_original = (14.4 - 9.6) / 9.6 ≈ 0.5 (50% increase)

Impact on Execution Time for 4 processors = (T_4_new - T_4_original) / T_4_original = (7.2 - 4.8) / 4.8 ≈ 0.5 (50% increase)

Impact on Execution Time for 8 processors = (T_8_new - T_8_original) / T_8_original = (3.6 - 2.4) / 2.4 ≈ 0.5 (50% increase)

Doubling the CPI for arithmetic instructions results in a 50% increase in execution time for all processor configurations.

1.9.3
The CPI of load/store instructions should be reduced to approximately -1.27 to match the performance of four processors using the original CPI values. Since CPI values should be positive, reducing the CPI of load/store instructions alone will not be

sufficient to match the performance of four processors, and other optimizations may be needed.

1.11 The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona have an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.

1.11.1 Find the CPI if the clock cycle time is 0.333ns.

- CPI = (Total Cycles) / (Total Instructions)
- CPI = (2.250E18 cycles) / (2.389E12 instructions)

CPI ≈ 942.69

The SPEC CPU2006 bzip2 benchmark CPI on the AMD Barcelona with a clock cycle time of 0.333 ns is approximately 942.69.

1.11.4 Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.

- Increase in CPU Time = (New Instruction Count / Old Instruction Count) * (New CPI / Old CPI) * Old CPU Time
- Increase in CPU Time = (2.6279E12 / 2.389E12) * (989.82 / 942.69) * 750 s
- Increase in CPU Time ≈ 1.1 * 1.0492 * 750 s ≈ 871.53 s

1.11.6 Find the new CPI.

- New CPI = Old CPI + (5% of Old CPI)
- New CPI = 942.69 + (0.05 * 942.69)
- New CPI = 942.69 + 47.1345
- New CPI ≈ 989.8245

1.11.11 Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.

- New Clock Rate = (Old Clock Rate) * (Old CPI) / (New CPI) * (Old CPU Time / New CPU Time)
- New Clock Rate = (Old Clock Rate) / (0.85 * 0.80)
- New Clock Rate = (Old Clock Rate) / 0.68
- New Clock Rate ≈ 1.47 * Old Clock Rate

1.12 Section 1.10 cites the utilization of a subset of the performance equation as a performance metric as a pitfall. To illustrate this, consider the following two processors. P1 has a clock rate of 4GHz, an average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

1.12.1

- Execution Time (P1) = (5.0E9 instructions / 4E9 Hz) * 0.9 = 1.125 seconds
- Execution Time (P2) = (1.0E9 instructions / 3E9 Hz) * 0.75 = 0.25 seconds
- Processor P2 has a significantly lower execution time, which means it performs the given task faster than Processor P1.

1.12.2

- Execution Time (P1) = (1.0E9 instructions / 4E9 Hz) * 0.9 = 0.225 seconds
- Clock Rate (P2) = 3 GHz = 3E9 Hz
- Average CPI (P2) = 0.75
- Total Instructions (P2) = 0.225 seconds * 3E9 Hz / 0.75 = 0.225E9 instructions
- So, Processor P2 can execute 0.225E9 instructions (225 million instructions) in the same amount of time that Processor P1 needs to execute 1.0E9 instructions.

1.12.3

- MIPS (P1) = (Clock Rate (P1) / 1E6) / (Average CPI (P1))
- MIPS (P1) = (4E9 / 1E6) / 0.9
- MIPS (P1) = 4,444.44 MIPS
- MIPS (P2) = (Clock Rate (P2) / 1E6) / (Average CPI (P2))
- MIPS (P2) = (3E9 / 1E6) / 0.75
- MIPS (P2) = 4,000 MIPS
- Based on MIPS alone, Processor P1 might have a higher MIPS value and, therefore, higher performance. To get a more accurate performance comparison, it's essential to consider other factors such as execution time, CPI, clock rate, and the actual workload being executed.

1.12.4

- MFLOPS= Execution time/No. of FP operations ×1E6
- MFLOPS (P1)≈1.7777E9 MFLOPS
- MFLOPS (P2)= 0.25 seconds/ 4.0E8 floating-point operations ×1E6
- MFLOPS (P2) ≈1.6E9 MFLOPS

1.13 Another pitfall cited in Section 1.10 is expecting to improve the overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 250 s, with 70 s spent executing FP instructions, 85 s executed L/S instructions, and 40 s spent executing branch instructions.

1.13.1

- New time spent on FP operations = 70 s - 14 s = 56 s
- Original total time = 70 s + 85 s + 40 s = 195 s
- New total time = 56 s + 85 s + 40 s = 181 s
- Reduction in total time = 195 s - 181 s = 14 s
- So, if the time for FP operations is reduced by 20%, the total time is reduced by 14 seconds.

1.13.2

- Time for INT operations = New total time - (Time for FP operations + Time for L/S operations + Time for branch instructions)
- Time for INT operations = 156 s - (56 s + 85 s + 40 s) =156 s - 181 s = -25 secs

1.13.3

- Percentage Reduction = [(Original Total Time - New Total Time) / Original Total Time] * 100
- Percentage Reduction = [(195 s - 156 s) / 195 s] * 100 ≈ 20%
- Yes, the total time can be reduced by approximately 20% by reducing only the time for branch instructions by 39 seconds.

1.15 When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical sections and/or to send data from one processor to another.

For 2 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 2) + 4 s = 50 s + 4 s = 54 s
Speedup relative to a single processor = 100 s / 54 s ≈ 1.85
Ideal speedup without overhead = 2 (since there are 2 processors)
The ratio of actual speedup to ideal speedup = 1.85 / 2 ≈ 0.925
For 4 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 4) + 4 s = 25 s + 4 s = 29 s
Speedup relative to a single processor = 100 s / 29 s ≈ 3.45
Ideal speedup without overhead = 4 (since there are 4 processors)
The ratio of actual speedup to ideal speedup = 3.45 / 4 ≈ 0.8625

For 8 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 8) + 4 s = 12.5 s + 4 s = 16.5s
Speedup relative to a single processor = 100 s / 16.5 s ≈ 6.06
Ideal speedup without overhead = 8 (since there are 8 processors)
The ratio of actual speedup to ideal speedup = 6.06 / 8 ≈ 0.7575

For 16 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 16) + 4 s = 6.25 s + 4 s = 10.25 s
Speedup relative to a single processor = 100 s / 10.25 s ≈ 9.76
Ideal speedup without overhead = 16 (since there are 16 processors)
The ratio of actual speedup to ideal speedup = 9.76 / 16 ≈ 0.61

For 32 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 32) + 4 s = 3.125 s + 4 s = 7.125 s
Speedup relative to a single processor = 100 s / 7.125 s ≈ 14.04
Ideal speedup without overhead = 32 (since there are 32 processors)
The ratio of actual speedup to ideal speedup = 14.04 / 32 ≈ 0.4388

For 64 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 64) + 4 s = 1.5625 s + 4 s = 5.5625 s
Speedup relative to a single processor = 100 s / 5.5625 s ≈ 17.97
Ideal speedup without overhead = 64 (since there are 64 processors)
The ratio of actual speedup to ideal speedup = 17.97 / 64 ≈ 0.2809

For 128 processors:
Per-processor execution time = (t/p) + overhead = (100 s / 128) + 4 s = 0.78125 s + 4 s = 4.78125 s
Speedup relative to a single processor = 100 s / 4.78125 s ≈ 20.92
Ideal speedup without overhead = 128 (since there are 128 processors)
The ratio of actual speedup to ideal speedup = 20.92 / 128 ≈ 0.1634