

# Assignment 1

CIS 480/580

**Due Date: Monday, September 29<sup>th</sup>, 2025**

**Total points = 20**

**Name: Swalpesh Kotalwar**

**CSU ID: 2918419**

**Chapter 1:**

**Q1 (Q13 of textbook):**

**Name three appliances that are candidates for being run by an embedded CPU. (2 points)**

Answer:

A microwave oven (controls timing, power level, keypad, safety interlocks).

A digital thermostat / HVAC controller (reads sensors, runs control loop, drives display).

A washing machine (runs wash cycles, motors, sensors, and user interface).

**Q2 (Q14 of textbook)**

**At a certain point in time, a transistor on a chip was 0.1 micron in diameter. According to Moore's law, how big would a transistor be on next year's model? (2 points)**

Answer:

Moore's law says transistor **count** (density) roughly doubles in a period (commonly quoted as every year or every ~18–24 months). If transistor density doubles, the *area per transistor* halves, so the linear dimension scales by  $1/\sqrt{2}$

Which is  $0.07071 \mu\text{m}$ .

**Chapter 2**

**Q3 (Q3 of textbook)**

**On a computer 1, all instructions take 10 nsec to execute. On computer 2, they all take 5 nsec to execute. Can you say for certain that computer 2 is faster? Discuss. Hint: pipeline. (2 points)**

Answer:

Not necessarily. The quoted “time per instruction” needs interpretation: if 5 ns is the clock period of a deeply pipelined processor, throughput can be higher but *latency* or effective performance depends on pipeline hazards, instruction mix, CPI (cycles per instruction), and pipeline overheads (stalls, branch penalties). A 5 ns clock with many pipeline stages may give higher *throughput*, but if pipeline causes frequent stalls or if the 10 ns machine executes each instruction in fewer cycles or with lower overhead, the 10 ns machine could be as fast or faster for some workloads. So you cannot say for certain you must compare overall CPI and instruction mix, not just the raw cycle time.

#### **Q 4 (Q11 of textbook)**

**A certain computer can be equipped with 1,073,741,824 bytes of memory. Why would a manufacturer choose such a peculiar number, instead of an easy-to-remember number like 1,000,000,000? (2 points)**

Answer:

1,073,741,824 bytes = 2 raised to 30 bytes I.e 1 gigabytes. Manufacturers use powers of two because binary addressing and memory hardware naturally align on powers of two (easier for address decoding, page tables, etc.). So the “peculiar” number is a round power-of-two in binary.

#### **Q5 (not from textbook)**

**Device a (11,7) Hamming codeword for a 7-bit data field 1010101. Show and explain how the transmitted codeword will be generated? How will it be confirmed at the receiving end that no errors have occurred during the transmission? Show with an example, how errors and the locations of the errors will be detected and corrected. (6 points)**

Answer:

We build standard Hamming (11,7) with 4 parity bits at positions 1,2,4,8 (1-based indexing). Data bits occupy positions: 3,5,6,7,9,10,11.

Steps: Place data bits (given data = 1 0 1 0 1 0 1) into positions:

pos3 = 1

pos5 = 0

pos6 = 1

pos7 = 0

pos9 = 1

pos10 = 0

pos11 = 1

So the positions 1..11 (with parity bits blank for now) look like:

— — 1 — 0 1 0 — 1 0 1

How the receiver checks for errors (syndrome method):

The receiver recomputes each parity  $s_k$  across the same sets (including the received parity bits). If all parities are even syndrome = 0  $\Rightarrow$  no error.  
If syndrome not equal to 0, the syndrome value gives the numeric index of the bit in error. Flip that bit to correct.

Example:

Suppose bit at position 6 (was 1) is flipped during transmission to 0. The receiver computes parity checks and gets a syndrome value = 6. That tells the receiver “bit 6 is wrong”; it flips bit 6 back and recovers the correct codeword.

#### **Q6 (Q14 of textbook)**

**In a Hamming code, some bits are “wasted” in the sense that they are used for checking and not for information. What is the percentage of wasted bits for messages whose total length (data + check bits) is  $2^n - 1$ ? Evaluate this expression for values of n from 3 to 10. (3 points)**

Answer:

$n = 3$ : total = 7  $\Rightarrow$  Wasted =  $3/7 * 100\% = 42.9\%$

$n = 4$ : total = 15  $\Rightarrow$  Wasted =  $4/15 * 100\% = 27\%$

$n = 5$ : total = 31  $\Rightarrow$  Wasted =  $5/31 * 100\% = 16\%$

$n = 6$ : total = 63  $\Rightarrow$  Wasted =  $6/63 * 100\% = 9.5\%$

$n = 7$ : total = 127  $\Rightarrow$  Wasted =  $7/127 * 100\% = 5.5\%$

$n = 8$ : total = 255  $\Rightarrow$  Wasted =  $8/255 * 100\% = 3.13\%$

$n = 9$ : total = 511  $\Rightarrow$  Wasted =  $9/511 * 100\% = 1.76\%$

$n = 10$ : total = 1023  $\Rightarrow$  Wasted =  $10/1023 * 100\% = 0.97\%$

#### **Q7 (Q17 of textbook) The disk, with 4096 bits/sector illustrated in Fig 2-19 of textbook/PowerPoint slides, has 1024 sector/track and a rotation rate of 7200 RPM. What is the sustained transfer rate of the disk over one track? (3 points)**

Answer:

Bits per sector = 4096.

Sectors per track = 1024.

So bits per track =  $4096 * 1024 = 4,194,304$  bits

Rotation rate = 7200 RPM =  $7200 / 60 = 120$  rotations per second.

Bits per second = bits per track  $\times$  rotations/sec

$$= 4,194,304 * 120 = 503,316,480 \text{ bits/sec.}$$

Convert to bytes/sec:  $503,316,480 / 8 = 62,914,560$  bytes/sec.

In Binary MiB/s:  $62,914,560 / 2^{20} = 60$  MiB/s exactly.

In Decimal MB/s ( $10^6$ ) = 62.915 MB/s