CS-1216 – Monsoon 2023 – Assignment 6

Rutam Kathale,

December 17, 2023

Problem 1. A memory subsystem has the following specifications.
(i) Cache takes 80 nsec to access the index field of each set.
(ii) Cache takes 10 nsec to access the tag field of each set.
(iii) Main memory access time of 500 nsec.
(iv) 80% memory request are for reads.
(v) Hit ratio 0.9 for read access and 0.8 for write access.
Compute the following.
(a) Average access time for read.
(b) Average access time for write and overall (both read and write) for write through scheme.
(c) Average access time for write and overall (both read and write) for write back scheme.
(d) Average hit ratio.

(a) Average access time for read: The average access time for read can be calculated using the formula:
$T_{readavg} = T_{hit}.p + T_{miss}.(1-p)$
T_{hit} = time given block found in memory i.e. cache time = 80 + 10 = 90 ns.
Hit ratio to read = 0.9
If block not found, we access the memory, so our penalty would be the cost of both cache access and memory access = 500 + 90 = 590 ns.
So, AAT = 0.9 x 90 + (0.10 x 590) = 140ns.

(b) FOr write through, write time is just the memory access time since write performed on both cache and main memory.
AAT = T_r x P_r + T_w x P_w
$= 0.8 * 140 + (0.2 * 500) = 212$
(c) For write back, write operations on cache only so, we consider only the cache access time.
AAT = T_r x P_r + T_w x P_w
$= 0.8.140 * (0.2 * 90) = 130$
(d) AVerage hit ratio = $p_r * h_r * p_w * h_w$ = 0.8 x 0.9 + 0.2 x 0.8 = 0.88

Problem 2. A tape drive has the following parameters.
(i) Bit density: 1600 bits/inches.
(ii) Tape speed: 200 inches/sec.
(iii) Time spent at inter-record gap: 3 ms.
(iv) Average record length: 1000 characters.
How many bytes can be stored on a tape reel of length 1200 ft. written on such a drive.

number of bits storable on tabe reel = bit density $*$ length of tape reel = 1600 bits/inch $*$ 1200 ft = 1600 bits/inch * 14400 inch = 23040000 bits.
Time spent at inter-record gap: 3ms and tape speed = 200 inches/sec.
Length of inter-record gap = 3ms $*$ 200 inch/sec = 0.6 inch.
Considering average record length,
number of records on tape reel = total # bits / # bits per record = 23040000 / (1000 * 8 (bits$->$ byte) = 2800 records.
Total length of inter-record gaps: # records $*$ length of each gap = 2800 $*$ 0.6 = 1728 inch.
bits for inter record gaps = 1728 $*$ 1600 (density) = 2764800 bits.
For actual data storage, we need to subtract the bits for inter-record gaps => 23040000 - 2764800 = 20275200 bits (=2534400 bytes = 2.53 Mb)

Note that this calculations have a lot of rounding off, so the final answer may not be exactly same across approaches.

Problem 3. The message MENU is to be transmitted using standard asynchronous serial format of 7-bit data, one start bit (low), one odd parity bit and two stop bits (high), at a rate of 1200 bits/sec. Draw the complete waveform of the message and specify the time for transmission.

Let's use ASCII encoding for the characters in "MENU". Each character will begin with a start bit - 0, and we can show data with 7 bits and use a parity bit afterwards to ensure correct message sent (usually validated through a checksum). Also, all character transmissions will end with 11 (stop bits.). We have 1200 bits/second as transmission speed.
Representing MENU in ASCII:
M: 1001101
E: 1000101
N: 1001110
U: 1010101
Let's pad the representation. We'll add the start bit (0), calculate the odd parity bit, and append the two stop bits (11) for each character. We result in the representation:
M: 01001101111
E: 01000101011
N: 01001110111
U: 01010101111
The baud rate is 1200 bits per second, so the time to transmit one bit is the reciprocal of the baud rate. Multiplying it with total bits in message would give us the total transmission time.
So, the # of bits transmitted = 4 characters $*$ 11 bits/character = 44 bits.
The baud rate = 1200 bits/sec.
Time to transmit 1 bit = 1/1200 = 0.000833 sec/bit
So, the total time for transmission: 0.000833 $*$ 44 = 0.03667 seconds (36.67 ms).
For showing the waveform, we'll consider MENU (padded) concatenated:
01001101111 01000101011 01001110111 01010101111
(Given in last page)

Problem 4. How do we check the different ranges of the partial remainder in Radix-2 SRT division?

In Radix-2 SRT division, the divisor is used to normalize the dividend, and then the division by basically calculating the new partial remainder based on the previous remainder and the current bits of the dividend, estimate next digit of quotient, update partial remainder, etc. The important bit is to estimate the quotient digit. Checking the differnet ranges is a part of estimating the quotient. So we predetermine these ranges, where the divisor is a parameter/dependency. The ranges are generally established by dividing the possible values of the partial remainder into segments. These segments are mapped to the specific quotient digit if the partial remainder was in that segment. Formally,
We first ensure the divisor is normalized (range [1,2] in binary).
Then, based on this divisor, we determine the limits of each range for the partial remainder.
Then, a lookup table is created that maps ranges of the partial remainder to the corresponding quotient digits. Behaving like a hash table, it can quickly determine quotient digit based on our input (partial remainder).
Then we check partial ranges against the table and update the table.

So, in a nutshell, the ranges of partial remainder depends on the quotient digit (generated from 2 most significant radix 2 digits of partial remainder). We calculate the new partial remainder as a function of old partial remainder (using comparison constants for selecting quotient digit). Based on this technique, we determine 2 most significant digits then quotient digits then based on that, the ranges of partial remainder.

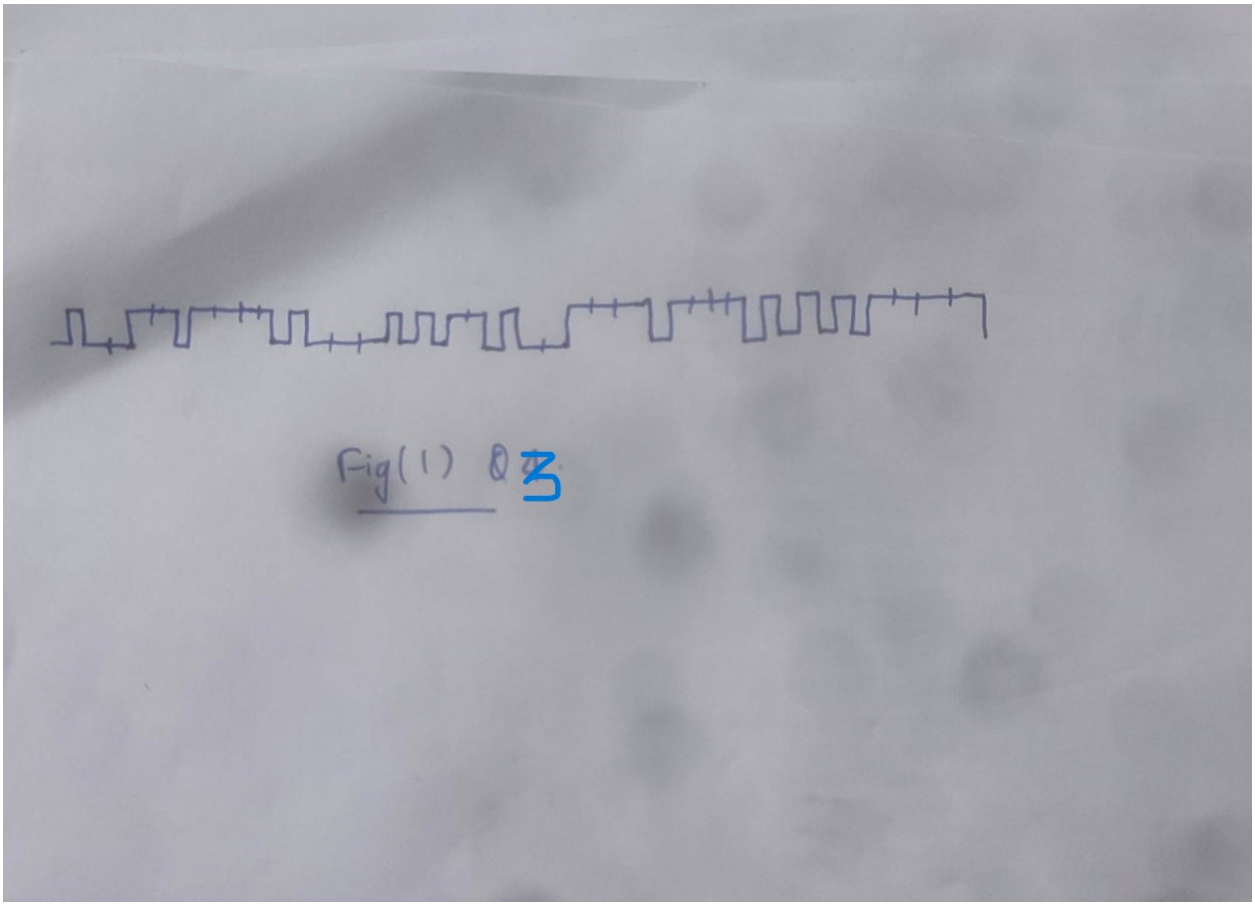

Figure 1: Q3