Birla Institute of Technology & Science, Pilani

Work Integrated Learning Programmes Division

First Semester 2020-2021

M.Tech (Data Science and Engineering)

Compre- Regular Exam (EC-3 Regular)

Course No. : DSECLZG516
Course Title : Computer Organization and Software Systems
Nature of Exam : Open Book
Weightage : 40%
Duration : 150 minutes
Date of Exam : 03/10/2021

Number of Pages: 6

Number of Questions: 8

Note to Students:

1. **All parts of a question should be answered consecutively. Each answer should start from a fresh page.**
2. **Assumptions** made if any, should be stated clearly at the beginning of your answer.
3. For all problems **relevant steps** are to be shown.

Q1 Answer the following Question [6 Marks]

A. Extensive research is being done keeping lateral thinking to assist in AI/ML approaches and how associative cache replacement algorithms could be beneficial. Keeping this in view a hybrid approach was adapted such that 70% follow LRU and next 30 % follow LFU. In order to have a reference the FIFO for the entire 100% was used for evaluation. The blocks generated are 3,2,5,7,2,3,6,4,6,5 and there are four lines. Using time line diagram show the replacements for both hybrid and FIFO along with the hit ratios of these two methods. Treat ties for replacement by replacing the one longest in cache.

Sol

LRU 70% and LFU 30% 3 Hits and 7 Miss [3 M]

	3	**2**	**5**	**7**	**2**	**3**	**6**	**4**	**6**	**5**
L1	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
L2		**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
L3			**5**	**5**	**5**	**5**	**6**	**6**	**6**	**6**
L4				**7**	**7**	**7**	**7**	**4**	**4**	**5**
H/M	M	M	M	M	H	H	M	M	H	M

FIFO

4 Hits and 6 Miss [1M]

	3	**2**	**5**	**7**	**2**	**3**	**6**	**4**	**6**	**5**
L1	**3**	**3**	**3**	**3**	**3**	**3**	**6**	**6**	**6**	**6**
L2		**2**	**2**	**2**	**2**	**2**	**2**	**4**	**4**	**4**
L3			**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
L4				**7**	**7**	**7**	**7**	**7**	**7**	**7**
H/M	M	M	M	M	H	H	M	M	H	H

B A block-set associative cache memory consists of 128 blocks divided into four block sets . The main memory consists of 16,384 blocks and each block contains 256 eight bit words.

a. How many bits are required for addressing the main memory? [1M]

b. How many bits are needed to represent the TAG, SET and WORD fields?[1M]

Solution:

Given-

· Number of blocks in cache memory = 128

· Number of blocks in each set of cache = 4

· Main memory size = 16384 blocks

· Block size = 256 bytes

· 1 word = 8 bits = 1 byte

Main Memory Size-

We have-

Size of main memory

= 16384 blocks

= 16384 x 256 bytes

= 2^{22} bytes

Thus, Number of bits required to address main memory = 22 bits

Number of Bits in Block Offset-

We have-

Block size

= 256 bytes

= 2^{8} bytes

Thus, Number of bits in block offset or word = 8 bits

Number of Bits in Set Number-

Number of sets in cache

= Number of lines in cache / Set size

= 128 blocks / 4 blocks

= 32 sets

= 2^5 sets

Thus, Number of bits in set number = 5 bits

Number of Bits in Tag Number-

Number of bits in tag

= Number of bits in physical address – (Number of bits in set number + Number of bits in word)

= 22 bits – (5 bits + 8 bits)

= 22 bits – 13 bits

= 9 bits

Thus, Number of bits in tag = 9 bits

Thus, physical address is-

Tag:9 bits

Set Number:5 bits

Word: 8 bits

Q2 Answer the following Questions [7 Marks]

A computer is being used by XYZ. This computer has a CPU which works at 1.5GHz, a hard disk rotating at 120 revolutions per second, having 4 platters,30000 tracks per surface and 128 sectors per track. Size of each sector is 1Kbyte and the head travels at 100 tracks per 1milli second.

The user needs to execute a program, named mfp.exe" on the hard disk which occupies one complete sector with its instructions. The average size of the instruction is 2bytes. The assortment of instruction group (percentages of instruction , Cycles per Instruction) are as follows:

ALU (43% ,1.2) , Load (22% , 4), Store (13% , 3), Branch (22% , 1.5).

a) What is the storage capacity of the hard disk? [0.5]
b) What is the average seek time?[0.5]
c) What is the average rotational latency?[0.5]
d) What is the time taken for a sector transfer request? [0.5]
e) Time to service a request to transfer content of any given sector[2M]

f) What are the average CPI, code execution in MFLOPS for the program mfp.exe[1M]

g) Compute the execution time of file"mfp.exe" from the time request received by OS for loading and till it completely executes. Ignore time taken by OS to do this operation? [2M]

Sol: [20 MIN]

a. Disk capacity =4platters*2 surfaces*30000 tracks* 128 sectors per track*1024bytes / sector

=4*2*30000*128*1024 bytes= 31,457,280,000

b. Assuming head initial position is on track 0:

average seek time=travel from (0 to 29999)/2=29999/(100*2)=149.99ms

c. Rotational delay is 1/120=0.00833 sec = 0.00833 * 1000 = 8.33 ms

Average Rotational latency=8.333 / 2 = 4.165 ms

d. Requested sector transfer time=disk revolution time / no. of sectors

=0.00833/128=65.1microseconds.

e. time to service a request to transfer content of a given sector

= Average seek time + Average rotational latency + Transfer time for a sector

=149.88us+4.165us+65.1us=219.145us

f. What are the average CPI, code execution in MFLOPS for the program mfp.exe

2.12

g. Sector has 1024 bytes and average instruction size is given as 2 bytes. There are 512 instructions in the sector

Time to execute= Average seek time + Average rotational latency + sector transfer time+No. of cycles to execute the program(average cpi per instruction*512 instructions)*time for one cycle(1/Freq)

=(149.88us+4.165us+65.10us)*2.12*512*(1/1.5)ns=154.22ms*1085.44*(0.666ns)

=154.22ms+1085.44*(0.000000666ms)=154.22ms+0.00072290304ms

=154.2207229030ms

Q3 Answer the following Questions [6 Marks]

A. consider the following COSS1 routine: [4 Marks]

```
typedef int array[2][2];

void COSS1(array dst, array src)
{
        int i, j;
        for (i = 0; i < 2; i++)
        {
                for (j = 0; j < 2; j++)
                {
                        dst[j][i] = src[i][j];
                }
        }
}
```

Assume this code runs on a machine with the following properties:

[Size of (int)==4, Source[src] array address starts at location 0 and destination[dst] array starts at location 16 (decimal) and There is single L1 data cache that is direct mapped, write through and write allocate with block size of 8 bytes. Assume that the cache has a total size of 16 data bytes and cache is initially empty.]

i. For each row and column ,indicate hit(h) and miss(m). (For example, reading src[0][0] is a miss and writing dst[0][0] is also a miss.)

A. The key to solving this problem is to visualize the picture in Figure 6.50. Notice that each cache line holds exactly one row of the array, that the cache is exactly large enough to hold one array, and that for all i, row i of `src` and `dst` maps to the same cache line. Because the cache is too small to hold both arrays, references to one array keep evicting useful lines from the other array. For example, the write to `dst[0][0]` evicts the line that was loaded when we read `src[0][0]`. So when we next read `src[0][1]`, we have a miss.

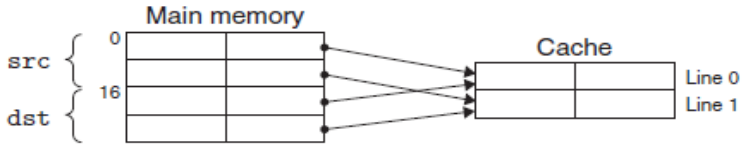
`dst` array

	Col 0	Col 1
Row 0	m	m
Row 1	m	m

`src` array

	Col 0	Col 1
Row 0	m	m
Row 1	m	h

Figure 6.50
Figure for Problem 6.18.



ii. Repeat the problem for a cache with 32 data bytes.

B. When the cache is 32 bytes, it is large enough to hold both arrays. Thus, the only misses are the initial cold misses.

dst array

	Col 0	Col 1
Row 0	m	h
Row 1	m	h

src array

	Col 0	Col 1
Row 0	m	h
Row 1	m	h

B. Identify who manages the below memory types [2 Marks]

i. CPU Registers : Compiler

ii. Virtual Memory : Hardware + OS

iii. Buffer cache : OS

iv. Web cache : Web proxy server

Q4: Answer the following questions. [5 MARKS]

A. As part of the COSS course, you had understood about Process Management done inside Operating Systems. Assume that you are writing a Web server process which is expected to look out for bytes arriving over the network on to your ethernet port. On which state this web server process will be in most of the times and WHY? (Note: It is important to give the correct reason for award of marks) [2M]*

The Web server process will be WAITING for bytes to come in over the network most of the times. Therefore this process will be in WAIT / BLOCKED state for a major chunk of its existence in the system. (All processes waiting for some resource are in WAIT / BLOCKED state)

B. Identify the mode to which the below statement belongs to [1M]

i. The executing code has no ability to directly access hardware or reference memory
Sol: User Mode

ii. The CPU memory system is effectively managed.
Sol: Kernel Mode

C. Your reporting manager is asking you to put your learnings on Computer Organization in this course to practice by asking you to recommend a hardware system configuration that will be suitable for both compute intensive as well as I/O intensive application. What parameters will you consider to arrive at the best configuration from your learnings in this COSS course? (NOTE: You should provide justification for each of the parameters that you mention here for marks to be awarded) [2M]

Sol:

Number of processor cores, Clock speed of the processor, Cache size for each core, Amount of SRAM, DRAM, Usage of SAS/SSD disks (and NOT SATA or magnetic disks). (Atleast 4 of these parameters should be explained with justification for marks)

Q5: Answer the following questions. **[6 MARKS]**

A. Consider a system with four process P0,P1,P2,P3 whose arrival time and CPU-I/O burst are given in the table (NOTE: CPU Burst is indicated with a number which is underlined) Assume that system uses round robin scheduling algorithm with time quantum 3. Draw Gantt chart and fill the table and also find the average waiting time, turnaround time and response time.

Processes	Arrival Time	CPU-I/O Burst	Finish Time	Waiting time	Turnaround Time	Response Time
P0	0	4+3+6				
P1	0	3+2+3				
P2	5	5+1+5				
P3	7	8+3+4				
Average						

Sol:

Processes	Arrival Time	CPU-I/O Burst	Finish Time	Turnaround time	Waiting Time	Response Time
P0	0	4+3+6	27	27	14	0
P1	0	3+2+3	16	16	8	3
P2	5	5+1+5	34	29	18	2
P3	7	8+3+4	39	32	17	3
Average				26.00	14.25	2.00

B. Consider a microprocessor generating a 16-bit address and having a 8-bit data bus. If an input and an output instruction can specify an 16-bit I/O port number, how many 8-bit I/O ports can the microprocessor support ?

Sol: 64K I/O ports

Q6: Answer the following questions. **[3 MARKS]**

Consider P0,P1,P2 are three processes synchronised with semaphores S0=1,S1=0,S2=0 as shown initialised. The table below gives the code of the processes

Process P0	Process P1	Process P2

While (true) {
wait(so);
print '0';
signal(S1);
signal(S2);
}
	wait(S1);
	signal(S1);
		Wait(S2);
		print '1';
		signal(S1);

a. If the sequence of processes schedule is P1,P2,P0,P0,P2,P0,P2,P0. What are the values printed?

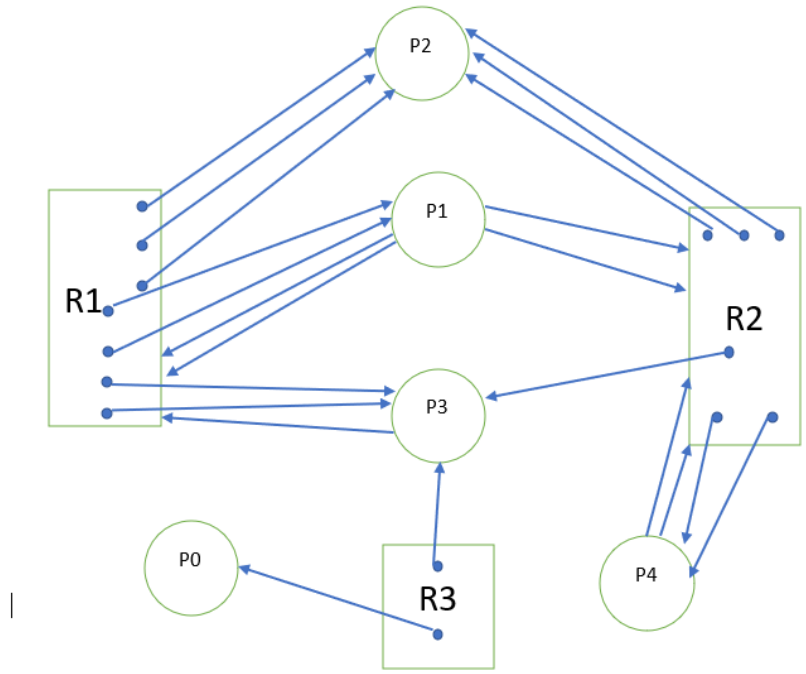
Seq		S0=1	S1=0	S2=0	Prints
1	**P1** runs first , waits on S1 BLOCKS	S0=1	S1=0	S2=0	-
2	**P2** runs waits 0n S2 BLOCKS	S0=1	S1=0	S2=0	-
3	**P0** runs waits S0 takes it Prints '0' Signals S1 & S2	S0=0	S1=1	S2=1	0 (1)
4	**P0** runs waits on S0 BLOCKS	S0=0	S1=1	S2=1	-
5	**P2** runs wait S2 takes it Signals S0	S0=0	S1=1	S2=0	1(1)
6	**P0** waiting on S0 takes it Prints '0' Signals S1 & S2	S0=0	S1=1	S2=0	-
7	**P2**takes S2 Prints '1' Signals S0	S0=1	S1=1	S2=0	-
8	**P0**runs takes S0 Prints '0' Signals S1 , S2	S0=0	S1=1	S2=1	-

b. If the semaphores are initialised with S0=0,S1=1 ,S2=1. What are the values printed for the above sequence?*

Sol: 1 is printed once.

Q7: Answer the following questions. **[3 MARKS]**

Consider the above Resource Allocation Graph with 5 processes P0 to P4 and 3 resources R1 to R3. Find the System is in deadlock or safe state and also find the safe sequence..



Sol: Table for Resource Request, Allocation and Available. [1.5M]

	Allocated			Max Need (allocated + requested)			Available			Remaining Need (Max Need - Allocated)		
Process	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
P0 (1)	0	0	1	0	0	1	0	0	0	0	0	0
P1 (5)	2	0	0	4	2	0	0	0	1	2	2	0
P2 (2)	3	3	0	3	3	0	3	3	1	0	0	0
P3 (3)	2	1	1	3	1	1	5	4	2	1	0	0
P4 (4)	0	2	0	0	4	0	5	6	2	0	2	0
Total	7	6	2				7	6	2			

Applying Deadlock Detection Algorithm and finding the safe state [1.5M]

Safe Sequence=<<P0,P2,P3,P4,P1>

Q8: Answer the following questions. **[4 MARKS]**

A. What are the different optimizations that you could do on the given program . You can assume that the given two functions are present in a single file and given for compilation. (Note: Reasons are important to be specified)*

```
const int value = 39;

static void my_func(int *val1, int *val2, int val) {
int i, p[20];
const int val3 = 5;
int var;

var = 3 + 4 + 11;

if (val >= value)
        for (i=0; i<=val2; i++)
                p[i] = val1;
else
        p[i] = val2;
}

void double_caller(int *val1, int *val2)  {
        short val = 7;
        my_func(val1, val2, val);
}
```

Sol: The static function indicates that my_func is local to this file and is not getting called from any other file. Since my_func is a 'static' function, compiler can assume that the value of val is always going to be 7 only. Because of this assumption, the variable val is never going to be greater than the value of the constant variable 'value' set to 39 and hence by the principle of removing dead code, the compiler optimizer can remove the 'if (val >= value)' check completely from the code.

Constant Propagation:

1. Value – is used only once so the variable can be replaced with constant '39'
2. Val - In function call 'val' can be replaced with '7'
3. Val3 - is used only once so the variable can be replaced with constant '5'

Constant Folding:

1. Var = 3+4+11 can be replaced with 18

Dead Code:

1. Var - is declared and value assigned not used in code
2. Sum – is declared and value assigned and not used in code
3. If (val >= value) this condition is never true, so for loop is never executed

Funtion In lining

4. My_Func can be part of double_caller, provided it is not invoked from any other procedure

B. Consider a computer system with 64K physical memory and a page size of 4096 bytes. The page table of a process is given below:

Page number(Hex)	Frame Number(Hex)
0	7
1	2
2	5
3	D
4	C
5	6

For the following logical address(Hex), determine the corresponding physical address:

i. 7512

ii. 3720

Solution:

i. 7512

0111 0101 0001 0010

0111 page no, rest 12 bits offset

7 has no entry in the page table, illegal address

ii. 3720

0011 0111 0010 0000

0011 page no, rest 12 bits offset

D720 is the physical address