

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**Continuous Assessment Test - II, April 2024**

**Winter Semester 2023-2024**

**Course Name, Code:**

**Programme:**

**Class Numbers:**

**Slot: D1+TD1**

**Faculty:**

**Duration: 90 mins**

**Max Marks: 50**

**General instruction(s):** Write question number properly on answer scripts.

Sl. No	Question
1	<p>a.</p> $CPI = \sum_i (\text{Fraction of instruction type } i \times \text{Cycles needed by instruction type } i)$ $\Rightarrow CPI = 0.45 * 1 + (0.32 + 0.08 + 0.15) * 2 = 1.55$ <p>Execution time = CPI * total number of instructions * Clock time</p> $= 1.55 * 10^5 * \frac{1}{40 * 10^6} \text{ sec}$ $= 3.875 \text{ ms}$ $MIPS = \frac{\text{Clock Frequency}}{CPI * 10^6}$ $= \frac{40 * 10^6}{1.55 * 10^6}$ $= 25.8$ <p>b.</p> <p>Micro programmed control is a control mechanism to generate control signals by using a memory called control storage (CS), which contains the control signals. Although micro programmed control seems to be advantageous to CISC machines, since CISC requires systematic development of sophisticated control signals, there is no intrinsic difference between these 2 control mechanisms.</p> <p>Hardwired control is a control mechanism to generate control signals by using appropriate finite state machine (FSM). The pair of “microinstruction-register” and “control storage address register” can be regarded as a “state register” for the hardwired control. Note that the control storage can be regarded as a kind of combinational logic circuit. We can assign any 0, 1 values to each output corresponding to each address, which can be regarded as the input for a combinational logic circuit.</p>

2

Suppose that a  $2M \times 16$  RAM memory is built using  $256K \times 8$  RAM chips and  $1K \times 8$  ROM memory is built using  $256 \times 8$  ROM chips with the word addressable memory, find the following:

a) 16 RAMS , 4 ROMS

b) Each RAM chip is  $256K \times 8$ , so to access one full word (2 bytes), 2 RAM chips would be involved.

c) The  $256K \times 8$  RAM chip has 18 address bits ( $2^{18} = 256K$ ), so each RAM chip would need 18 address bits.

d)8

e)module 0 – word 27  
module 3 - word 3

3

a.

16 Kb cache size.

4 way set associative.

Line size =  $4 \times 32$  bit = 16 bytes.

Here, in question "word" is mentioned and even

"Where in the cache is the **word** from memory location" is asked. So, word addressing is in use. So, offset bits = 2 for 4 words.

No. of sets=no of lines/p(way)

No. of lines=cache size /line size

$$\#sets = \frac{2^{10}}{4} = 256$$

tag(22 bit)	sets( 8 bit)	block size(2 bit)
-------------	--------------	-------------------

now address is ABCDE8F8

its binary form is :1010 1011 1100 1101 1110 1000 1111 1000

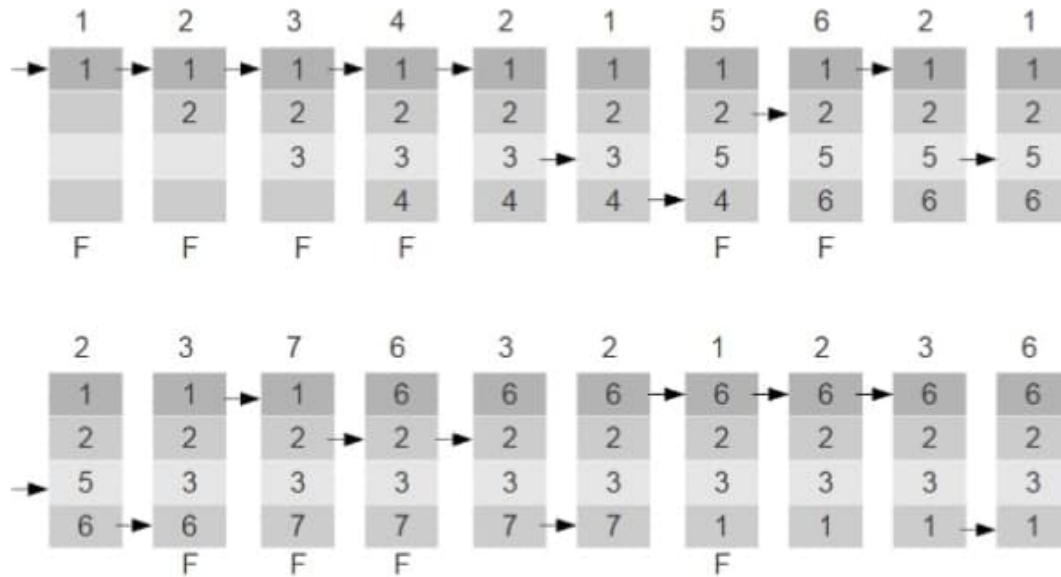
<1010 1011 1100 1101 1110 10> <00 1111 10> <00>

it is mapped to set number 62 in cache.

b. 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6

Total references: 20

LRU



Page faults= 10

4

a.



“Data count register gives the number of words the DMA can transfer in a s

Here it is 16 bits.. so max  $2^{16}$  words can be transferred in one cycle..

Since memory is byte addressable.. 1 word = 1 byte

so  $2^{16}$  bytes in 1 cycle..

Now for the given file..

$$\text{File size} = 29154 \text{ KB} = 29154 \times 2^{10} \text{ B}$$

$$1 \text{ cycle} \rightarrow \text{DMA transfers } 2^{16} \text{ B}$$

i.e

$$1 \text{ B transferred by DMA} \rightarrow \frac{1}{2^{16}} \text{ cycles.}$$

Now, for full file of size 29154 KB,

$$\text{Minimum number of cycles} = \frac{(29154 \times 2^{10} \text{ B})}{2^{16}} = 455.53$$

But number of cycles is asked so  $455.53 \rightarrow 456$ .

b. i) destination initiated Handshaking – When source initiates the data transfer process. It consists of signals: DATA VALID: if ON tells data on the data bus is valid otherwise invalid. DATA ACCEPTED: if ON tells data is accepted otherwise not accepted.

ii) destination initiated strobe I/O

In many applications, valid data is present on an external device only at a certain time, so it must be read in at that time.

E.g. the ASCII-encoded keyboard. When a key is pressed, circuitry on the keyboard sends out the ASCII code for the pressed key on eight parallel data lines, and then sends out a strobe signal on another line to indicate that valid data is present on the eight data lines.

The sending device, such as a keyboard, outputs a parallel data on the data lines, and then outputs an STB signal to let you know that valid data is present. For low rates of data transfer, such as from a keyboard to a MP, a simple strobe transfer works well. However, for higher speed data transfer, this method does not work because there is no signal which tells the sending device when it is safe to send the next data byte.

5

a)

Big Endian:

<b>Address:</b>			0x100	0x101	0x102	0x103		
<b>Value:</b>			01	23	45	67		

Little Endian:

<b>Address:</b>			0x100	0x101	0x102	0x103		
<b>Value:</b>			67	45	23	01		

b)

Memory mapped IO allows peripheral devices to be accessed by mapping their control registers and data buffers to specific memory addresses. However, the address space available for memory mapped IO is limited by the size of the memory address space that the system can support.

For example, if a system has a 32-bit address bus, it can support a maximum of  $2^{32}$  addressable locations, which correspond to 4 GB of memory address space. If a significant portion of this address space is already allocated to system memory, there may not be enough address space left to map all the peripheral devices in the system.

Additionally, some peripheral devices may require a large amount of address space for their control registers and buffers. For example, graphics cards or network controllers may have a large number of registers that need to be

accessed for efficient operation. If each of these devices requires a significant portion of the available address space, it may not be possible to map all of them to memory addresses due to limitations in the address space.

Therefore, in systems with limited address space, there may not be enough address space available to address all the peripheral devices using memory mapped IO. In such cases, other IO mechanisms such as port-mapped IO or DMA may need to be used to access the peripheral devices.

## **2) Interrupt driven or DMA**

One technique to avoid wastage of CPU burst time in I/O transfer mechanisms is using interrupt-driven I/O. In interrupt-driven I/O, the CPU does not need to actively poll the I/O device for completion of data transfer. Instead, the I/O device sends an interrupt signal to the CPU when the data transfer is complete, allowing the CPU to perform other tasks in the meantime.

This technique reduces CPU utilization as the CPU does not have to continuously check the status of I/O operations. It allows the CPU to efficiently handle other tasks while waiting for I/O operations to complete, thereby avoiding wastage of CPU burst time.

Another technique is using direct memory access (DMA) for data transfer. DMA allows the I/O device to transfer data directly to and from the memory without involving the CPU. This offloads the data transfer task from the CPU, allowing it to focus on other tasks and reducing the wastage of CPU burst time.

Overall, utilizing interrupt-driven I/O and DMA can significantly reduce the wastage of CPU burst time in I/O transfer mechanisms, leading to improved system efficiency and performance.