

17. (a) In designing and evaluating any algorithm, two properties are of interest. What are they? (2 marks)

(i) Property 1: *The time taken – time complexity* \_\_\_\_\_

(ii) Property 2: *The memory used – space complexity* \_\_\_\_\_

(b) Express the following polynomial in nested product form and explain why this form is computationally useful.

$$P(x) = -4 + 2.1x^2 + 5x^3 + 7x^4 + 3.2x^6$$

(i) Nested Product Form: (1 mark)

$$P(x) = -4 + x(x(2.1 + x(5 + x(7 + x(3.2x))))))$$

(ii) It is useful because: (1 mark)

*the number of operations to evaluate it is  $O(N)$ . In the original form it is  $O(N^{**2})$*

(c) (i) Describe what is meant by a “collision” when using a hashed form of data storage and explain how it can be dealt with. (1 mark)

*In applying a typical hashing function to an incoming data stream, e.g.  $H_0(e) = (e \text{ Mod } M + 1)$ , two pieces of data may be allocated to the same location. This is a COLLISION.*

*The solution is to use a family of hashing functions such that, when a collision occurs, a second function is used to try to find an empty space. If another collision occurs, then a third function is used, etc. In all, there need to be  $M$  possible hashing functions to try all the possible locations in a storage area with  $M$  locations. (a typical family of functions might be  $H_k(e) = (H_0(e) + k) \text{ mod } M$ )*

(ii) Give an example of a situation in which a hashed storage scheme would be an appropriate solution to the data structure problem (1 mark)

*The example needs to look at a problem where the data being used is always a small subset of a large potential data space. Typical examples might be:*

*The set of variable names used in a particular computer program (a subset of all possible names)*

*The last names of the students in a particular class (a subset of all possible last names)*

*The books in a library (a subset of all possible books)*

(d) It is desired to sort a randomly ordered data set (consisting of numbers only) using a binary tree (a Tree Sort).

(i) Describe the algorithm for doing this (either as a flow chart or a set of steps) (1 mark)

*1. Place the first item at the root of the tree (the tree will be built from a set of nodes each of which has two onward pointers – a left pointer and a right pointer)*

*2. Take the next data item, if it is smaller than the root, follow the left pointer, else follow the right pointer.*

*3. If the space pointed to is empty, store the item. If it is occupied, check if the new item is smaller than the current occupant of the space. If it is take the left pointer, else take the right pointer.*

*4. Repeat step 3 until an empty space is found.*

*5. Repeat from step 2 until all the data is stored in the tree.*

*6. Once all the data is stored, extract the data from the tree using an Inorder Traversal. (left-root-right)*

7. Starting at the root follow the left pointers to the leftmost subtree. Extract the left leaf, then the root and then follow the right pointers to the leftmost subtree and extract the leaf.

8. recursively implement step 7 until all the data in the tree is extracted.

9. The extracted data will be ordered.

(ii) What would the time complexity of this process be? (0.5 marks)

*Storing the tree is  $O(N \log N)$ .*

*The traversal is  $O(N)$*

*The total time complexity is  $O(N \log N)$*

(iii) For the problem described, what is the ratio of memory needed for the Tree Sort compared to a Quicksort? (0.5 marks) (assume that all the data elements are integers)

*The Tree Sort needs at least 2 pointers per node. If the data are integers and so are the pointers, each node requires 3 integers. Thus to store  $N$  items requires  $3N$  space.*

*The Quick Sort, if it is done in the same array requires one extra storage location and thus its space needs are approximately  $O(N)$*

*Thus the ratio is  $3N/N = 3$*

*(note that this will differ depending on the assumptions made about the size of the various data elements)*

(e) A large data set represents the students at McGill University. It is required to access the data set frequently. If each student is to be accessed  $M$  times, derive an expression for  $M$  in terms of the total number of students,  $N$ , if it is to be worth sorting the data set before searching for a student. (2 marks)

*If unsorted, each access is  $O(N)$*

*Thus for  $M$  accesses, the time taken is approximately  $M.N$*

*If the data is sorted using a Quick Sort, the time taken to sort is  $O(N\log N)$  but each access is now  $O(\log N)$ .*

*Thus the time for  $M$  accesses is  $N\log N + M\log N = (M+N)\log N$*

*It is worth sorting at the point where the two processes are approximately equal:*

$$M.N = (M+N)\log N$$

$$M = N\log N / (N - \log N)$$

18. (a) Explain the roles of the following components of an interface:

(i) Device Data Register (1 mark): \_\_\_\_\_

---

*To move data between the bus and the interface*

(i) Data Address Register (1 mark): \_\_\_\_\_

---

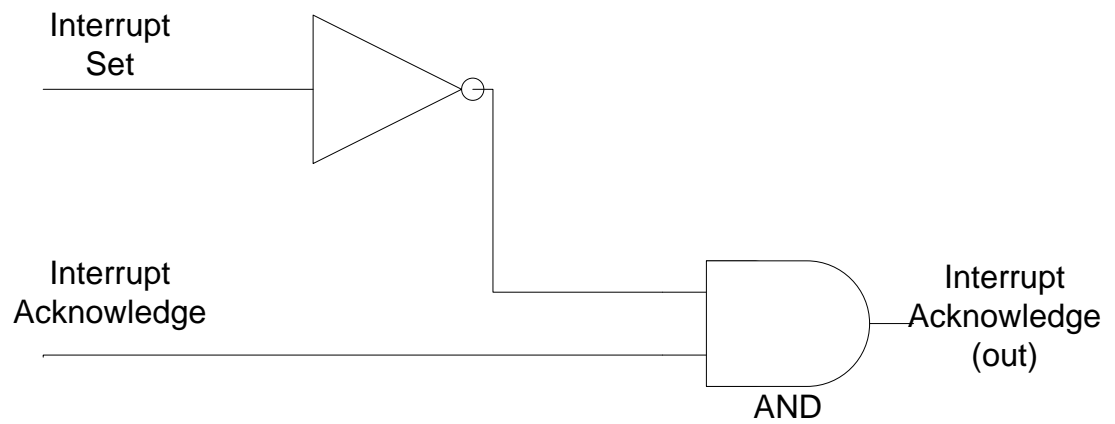
*To provide a comparison with the address on the bus and the address of the device The address on the bus is stored in the data address register.*

(b) In a vectored interrupt system, the CPU transmits an *Interrupt Acknowledge* signal to the interrupting device when it receives an *Interrupt* signal. If several devices have interrupted it is important that only one accepts the *Interrupt Acknowledge* signal. This is the first interrupting device that is encountered on the *Interrupt Acknowledge* line. This device blocks the *Interrupt Acknowledge* signal, i.e. if *Interrupt Acknowledge* is a "1" coming in to the interface, it should be set to "0" on exiting the interface. The two signals involved in this are the *Interrupt Set* and the *Interrupt Acknowledge*.

(i) Complete the truth table below (with signal names and logic values) for the logic that will generate the outgoing Interrupt Acknowledge signal (1 mark).

<b>Interrupt Set</b>	<b>Interrupt Acknowledge</b>	<b>Interrupt Acknowledge (Out)</b>
<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>

(ii) Draw the circuit that will implement the truth table in (i) (1 mark)



(c) A particular system takes (on the average) 40 microseconds to get to the pertinent ISR after the interrupt is received and the FETCH and EXECUTE cycle is broken. The overhead of the ISR (i.e. the time taken up with restoring the contents of the CPU registers, etc.) takes 15 microseconds.

- (i) If 25 % of the total interrupt time(from the break in CPU execution) is associated with the overhead, determine how much ISR execution time it takes to actually service the I/O device on average. (1 mark)

Solution:

If the total time for the ISR to execute takes  $x$  microseconds on average, then

$$(40+15)/(40+x) = 0.25$$

$$(0.25)*(40+x) = 10 + 0.25x = 55$$

$$x = 180 \text{ microseconds}$$

Alternate interpretation

If the total time for the ISR to service the I/O device takes  $x$  microseconds on average, then

$$(40+15)/(40+x + 15) = 0.25$$

$$(0.25)*(40+x + 15) = 13.75 + 0.25x = 55$$

$$x = 165 \text{ microseconds}$$

- (ii) Measurements on a real system suggest that only 22% of the CPU time is spent on interrupts. What is the average number of interrupts received by the CPU each second?  
(1 mark)

Solution:

If there are  $n$  interrupts per second, then the time taken is  $n$

$$n * 220 * 10^{-6} = 0.22$$

$n = 1000$  interrupts per second

(d) A computer is equipped with a character based display employing a 90 by 48 screen format (i.e. 90 characters by 48 rows). After examining a (screen) page, we have the option of requesting a hard copy, which is printed while examining the next page.

Assume the printer has a single character buffer, i.e. it prints characters as soon as they arrive. After a character is printed, the printer interface interrupts the CPU. If it takes an average of 51.8 seconds to print a page, and the printer can print 125 characters per second, calculate the time taken by the interrupt routine ( 2 marks)

Answer

*Total number of characters to be printed =  $90 * 48 = 4320$  characters*

*51.8 seconds to print page = 11.99 (i.e. 12) millisecs per character.*

The printer speed is 125 characters per second or 8 millisecond per character. The difference of (12-8)=4 millisecond is the time needed for the interrupt

(e) (i) Describe the steps necessary to move data from the CPU to a printer through the printer interface if interrupts do not exist. ( 2 marks)

1. CPU checks the status of the interface – if BUSY, it rechecks

2. Place the address of the Device Status Register on the bus

3. *Place the data to be transferred on the bus*
4. *Interface recognizes the address and latches data*
5. *The data is moved to a shift register to be sent down the serial line*
6. *The load signal from the Address Decoder is used to start a counter*
7. *While the counter has not reached the maximum number of bits to be sent, the BUSY signal is set.*

(ii) This process is known as (1 mark):

*Programmed Input and Output*