**Computer Architecture**

Dr. Brian Nolan

Mr. Keith Maycock

Answer any **two** questions

Duration of exam: **90 minutes**

Attachments: **Boolean Algebra Identities**

Question one

1. How many nibbles does it take to represent AA base 16 in Binary? **4**
2. Convert 17 base 8 into a binary number. You must show the process of converting the number. **001 111**
3. Compare and contrast the advantages of using an assembly programming language in comparison to using an Object Oriented programming language.

**Ans= Advantages**

**1. Programs written in Assembly Language accounts for greater executional efficiency as they are faster**

**and consume little storage compared to higher-level programming language.**

**2. Programs written in this language is less tedious to read and understand unlike programs written in**

**Machine Language.**

**3. Codes used for Assembly Language programs are easier to learn, read, faster and easier to understand**

**than programs written in Machine Language using 0s and 1s.**

**4. More compact in form that programs written in High Level Language as less statements are used to**

**represent programming instructions.**

1. Congratulations you have won first prize in designing a suitable logic circuit to create an on-demand cloud computing component for mobile phones. Intel is going into production and is going to produce one million components. Before production you need to make sure that the circuit is optimal. If all logic chips and inverters cost one euro, how much money can you save Intel?

The function is described as:

\_ \_\_\_\_

F = AD(D + AC) + C(A+C + C)

Ans=

1. Prove that your solution is optimal by using truth tables.

Question two

1. Discuss the precautions that you should take when working inside a typical computer. Detail what could happen if these precautions are not followed.

**Ans=**

**1. Unplug the power cord to the computer before opening the case to work inside of it. With the cord unplugged you are sure there is no power to the motherboard.**

**2. Capacitors can still store electric charges for a short time from the when you turned off the power switch so it’s wise to wait a few minutes after pulling the plug before working on your PC.**

**3. A properly grounded wrist strap can be purchased from an electronics store to prevent ESD. Use one of these before touching any components inside such as hard drive, video card, memory stick, etc. If a grounded wrist strap is not available then make sure you touch a grounded object such as a metal pipe to discharge any built up static so you don’t “shock” your components.**

**4. To avoid damage due to static electric discharge, hold circuit boards by the edges and avoid touching the individual components, wires, chips, or capacitors.**

**5. If you are going to remove components from the system, connect an anti-static mat or use the bag the component came in to place them on and prevent static discharge. Also be aware the hard drive is heavy and can easily slip out of your hand and if dropped may very well not work again.**

**6. Never ever open items labeled as non-serviceable. Components labeled “Non Serviceable Components Inside” are not a challenge because opening them up could lead to serious injury. Some parts of your computer just aren’t meant to be repaired. The power supply is one such item and should be replaced if defective, don’t attempt to repair it.**

1. Define the computer term BIOS. What function does the system BIOS have in a typical computer system?

**Ans=**

**Basic Input/Output System**

**The BIOS software is built into the PC, and is the first code run by a PC when powered on ('boot firmware'). The primary function of the BIOS is to load and start an operating system. When the PC starts up, the first job for the BIOS is to initialize and identify system devices such as the video display card, keyboard and mouse, hard disk, CD/DVD drive and other hardware. The BIOS then locates software held on a peripheral device (designated as a 'boot device'), such as a hard disk or a CD, and loads and executes that software, giving it control of the PC. This process is known as booting, or booting up, which is short for bootstrapping.**

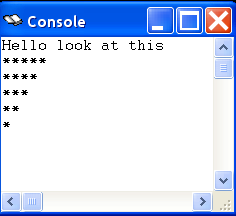
1. Discuss the POST program that typical BIOS would run after the computer has been turned on.

**Ans=**

**The first thing that the BIOS does when it boots the PC is to perform what is called the Power-On Self-Test, or POST for short. The POST is a built-in diagnostic program that checks your hardware to ensure that everything is present and functioning properly, before the BIOS begins the actual boot. It later continues with additional tests (such as the memory test) as the boot process is proceeding.**

**The POST runs very quickly, and you will normally not even notice that it is happening--unless it finds a problem. A PC that, when turned on, makes a beeping sound and then stops without booting up. That is the POST telling you something is wrong with the machine. The speaker is used because this test happens so early on, that the video isn't even activated yet! These beep patterns can be used to diagnose many hardware problems with your PC. The exact patterns depend on the maker of the BIOS; the most common are Award and AMI BIOSes.**

1. Write an assembly program that produces the output seen on the following page. Your solution should be produced using loops to determine the number of stars on each line.



**Ans=**

**.text #same start as usual**

**.globl \_\_start**

**start:**

**la $a0, str #printing a string**

**li $v0, 4**

**syscall**

**addi $s0, $s0, 5 #setting the maximum number of stars**

**outerloop: #defining the outer loop label**

**mul $t0, $t0, 0 # ensuring the value stored in register t0 is zero**

**add $t0, $t0, $s0 # setting the number in t0 to be equal to the number in s0**

**loop: #inner loop label**

**la $a0, star #printing a star**

**li $v0, 4**

**syscall**

**addi $t0, $t0, -1 # decrementing the loop counter**

**bnez $t0, loop #branching if not equal to zero**

**la $a0, nline #moving to a newline**

**li $v0, 4**

**syscall**

**addi $s0, $s0, -1 #decrementing the outer loop counter**

**bnez $s0, outerloop #branching if not equal to zero**

**li $v0, 10 #exiting the program**

**syscall**

**.data**

**str: .asciiz "Hello look at this\n"**

**nline: .asciiz "\n"**

**star: .asciiz "\*"**

1. Convert 10 base 2 into a decimal number. Note: you must show the method you used for the conversion.

**Ans=**

**(1\*2^3) + (0\*2^2) + (1\*2^1) + (0\*2^0)= 10**

Question three

1. Discuss the protocol you would use when converting from decimal to hexadecimal.

**Ans=**

**Keep dividing by 16, noting the remainder until the quotient is 0.**

**Example**

**54401 / 16 = 3400 R1**

**3400 / 16 = 212 R8**

**212 / 16 = 13 R4**

**13 / 16 = 0 R13**

**0**

**=> 54401 = D481**

1. What are Turing machines used for? What is the relationship between Turing machines and Finite State Machines?

**Ans=**

**Turing machines A mathematical model of a hypothetical computing machine that can use a predefined set of rules to determine a result from a set of input variables**

1. Construct a Finite State Machine to accept the following words, L ={a, aab, abab, abb}.

Ans=

1. Write an assembly program to print out the factorial (factorial 3 = 3\*2\*1 = 6) of a user entered number

**Ans=**

**.text #same start as always**

**.globl \_\_start**

**start:**

**la $a0, str #printing a string**

**li $v0, 4**

**syscall**

**li $v0, 5 #reading in an integer**

**syscall**

**addi $t0, $t0, 1 #initializing the value in register t0 to be 1**

**loop: # defining the loop position**

**mul $t0, $t0, $v0 # calculating the factorial of the entered number**

**addi $v0, $v0, -1 # decrementing the counter**

**bnez $v0, loop # branching back to the loop if != 0**

**la $a0, final**

**li $v0, 4**

**syscall**

**move $a0, $t0 #moving the result to register a0, ready for printing.**

**li $v0, 1**

**syscall**

**li $v0, 10**

**syscall**

**.data**

**str: .asciiz "Please enter a number\n"**

**final: .asciiz "The factorial of the number entered is "**

1. Differentiate between parallel and serial data transfer.

**Ans=**

**Parallel transfers use multiple "lanes" for data and programs, and in keeping with the 8 bits = 1 byte nature of computer information, most parallel transfers use multiples of 8.**

**Multiple bits of information are sent at the same time.**

**At identical clock speeds, parallel transfers are faster than serial transfers because more data is being transferred.**

**Serial transfers have the following characteristics:**

**One bit at a time is transferred to the device.**

**Transmission speeds can vary greatly, depending on the sender and receiver.**

**Very few connections are needed in the cable and ports (one transmit, one receive, and a few control and ground wires).**

**Cable lengths can be longer with serial devices. For example, an UltraDMA/66 ATA/IDE cable can be only 18 inches long for reliable data transmission, whereas a Serial ATA cable can be almost twice as long.**

**Although RS-232 serial ports are slow, newer types of serial devices are as fast or faster than parallel devices. The extra speed is possible because serial transfers don't have to worry about interference or other problems caused by running so many data lines together.**

1. Explain two different types of cooling systems used to keep a CPU at a reasonable temperature.

**Ans=**

* 1. **Heatsink is a finned metal device (occasionally plastic on very old processors) that radiates heat away from the processor.**
  2. **Liquid cooling systems attach a liquid cooling unit instead of an active heatsink to the processor. A pump moves the liquid (which might be water or a special solution, depending upon the cooling system) through the computer to a heat exchanger, which uses a fan to cool the warm liquid and back to the processor.**