COMP1562 Logbook (Week 1)

Basic Information

1.1	Student name	**Trevor Kiggundu (001001720)**
1.2	Who did you work with? Name and/or id	**Maruf Hoque**
1.3	Which lab topic does this document relate to?	System Bit Processors and System Shells
1.4	How well do you feel you have done?	I have completed the exercise and am totally satisfied with my work.
1.5	Briefly explain your answer to question 1.4	My group and I were able to successfully follow and complete the tasks. Proof of that is shown below.

Task 1:

1.1:

In this task, we were asked to use the fetch-execute cycle to replicate pseudo program execution by loading, adding and storing; three tasks that are normally executed within a CPU. The specifications of the machine are shown below:

Hypothetical 16 bit processor

16 bit accumulator AC
16 bit instruction register IR
12 bit program counter PC

Instruction format:
4 bit opcode
12 bit address

opcodes:

0011 = Load AC from I/O **3**

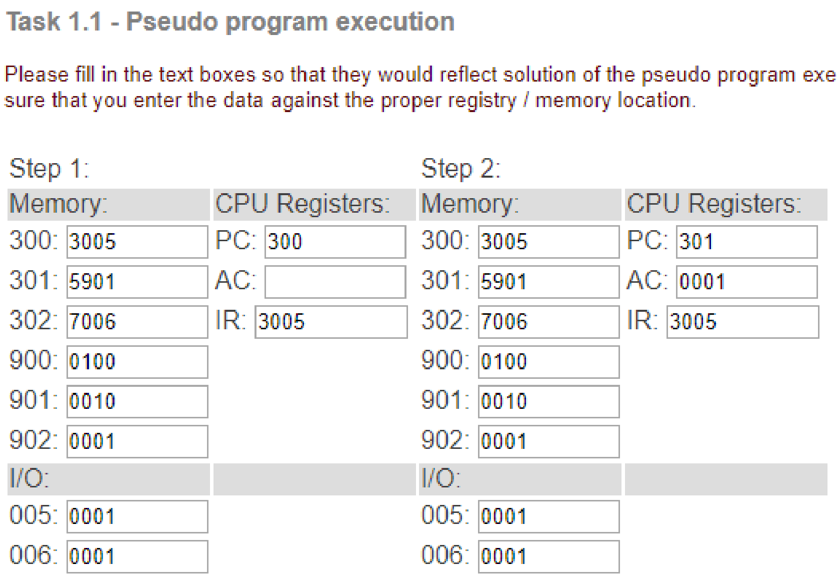
0101 = Add to AC (add date from the given memory location) **5**

0111 = Store AC to I/O **7**

Annotated screen shots demonstrating what you have achieved:

Screenshots showing my correct answers for task 1.1:

Steps 1 and 2 showing the task of loading AC from the device buffer:



Task 1.1 - Pseudo program execution

Please fill in the text boxes so that they would reflect solution of the pseudo program execution task. Make sure that you enter the data against the proper registry / memory location.

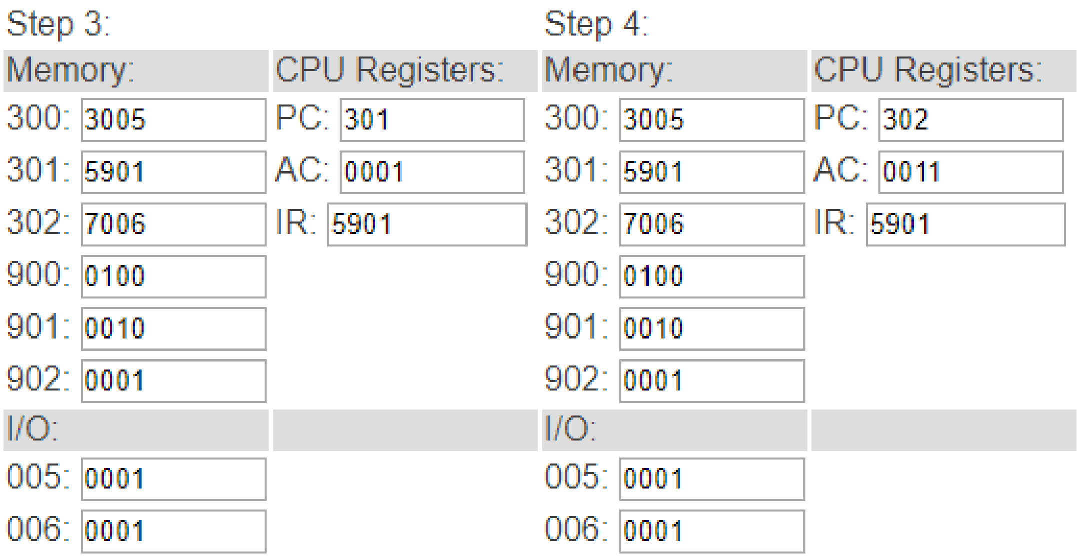
Step 1:

Memory:		CPU Registers:	
300:	3005	PC:	300
301:	5901	AC:	
302:	7006	IR:	3005
900:	0100		
901:	0010		
902:	0001		
I/O:			
005:	0001		
006:	0001		

Step 2:

Memory:		CPU Registers:	
300:	3005	PC:	301
301:	5901	AC:	0001
302:	7006	IR:	3005
900:	0100		
901:	0010		
902:	0001		
I/O:			
005:	0001		
006:	0001		

Steps 3 and 4 showing the task of adding info to the next memory location:



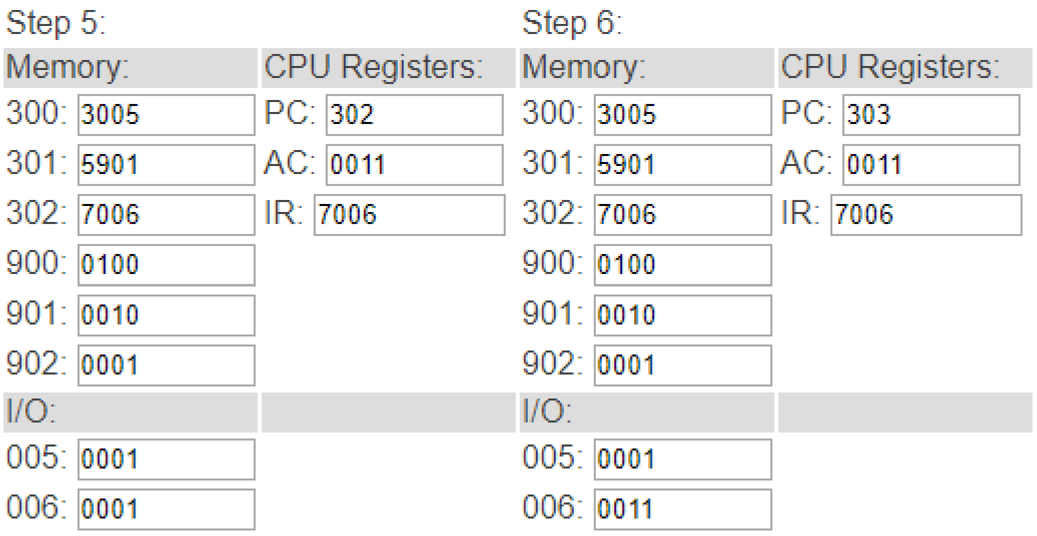
Step 3:

Memory:		CPU Registers:	
300:	3005	PC:	301
301:	5901	AC:	0001
302:	7006	IR:	5901
900:	0100		
901:	0010		
902:	0001		
I/O:			
005:	0001		
006:	0001		

Step 4:

Memory:		CPU Registers:	
300:	3005	PC:	302
301:	5901	AC:	0011
302:	7006	IR:	5901
900:	0100		
901:	0010		
902:	0001		
I/O:			
005:	0001		
006:	0001		

Steps 5 and 6 showing the task of storing the info from AC into the device buffer:



Step 5:

Memory:		CPU Registers:	
300:	3005	PC:	302
301:	5901	AC:	0011
302:	7006	IR:	7006
900:	0100		
901:	0010		
902:	0001		
I/O:			
005:	0001		
006:	0001		

Step 6:

Memory:		CPU Registers:	
300:	3005	PC:	303
301:	5901	AC:	0011
302:	7006	IR:	7006
900:	0100		
901:	0010		
902:	0001		
I/O:			
005:	0001		
006:	0011		

1.2:

In this task, we were asked to calculate the cache hit ratio using the machine specifications shown below:

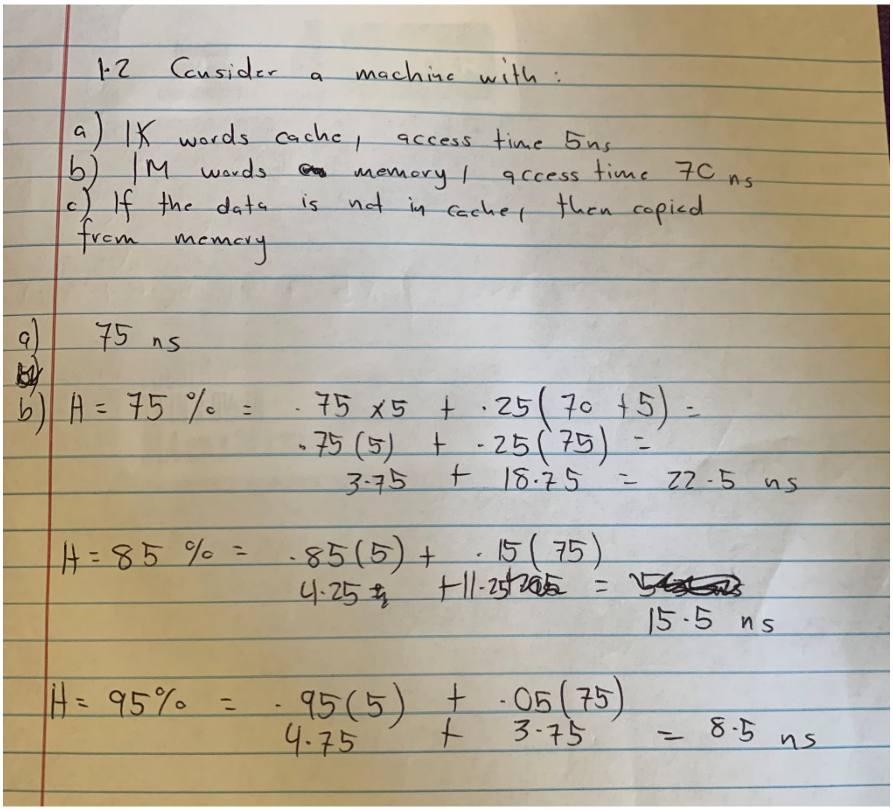
A) 1k Words Cache, Access Time 5ns

B) 1M words memory, Access time 70ns

C) If the data is not in the Cache, then it is copied from the memory

Annotated screen shots demonstrating what you have achieved:

My working out shown on paper:



1·2 Consider a machine with :

a) 1K words cache, access time 5ns
b) 1M words memory, access time 70 ns
c) If the data is not in cache, then copied from memory

a) 75 ns

b) H = 75 % = ·75 x5 + ·25(70 +5) =
·75 (5) + ·25(75) =
3·75 + 18·75 = 22·5 ns

H = 85 % = ·85(5) + ·15(75)
4·25 + 11·25 =
15·5 ns

H = 95% = ·95(5) + ·05(75)
4·75 + 3·75 = 8·5 ns

Calculations:

a) Cache miss time: (70 ns + 5 ns) = **75 ns**

b) H= 75% = .75 x 5 + .25 (70 + 5) =

.75 x 5 + .25 (75) =

3.75 + 18.75 = **22.5 ns**

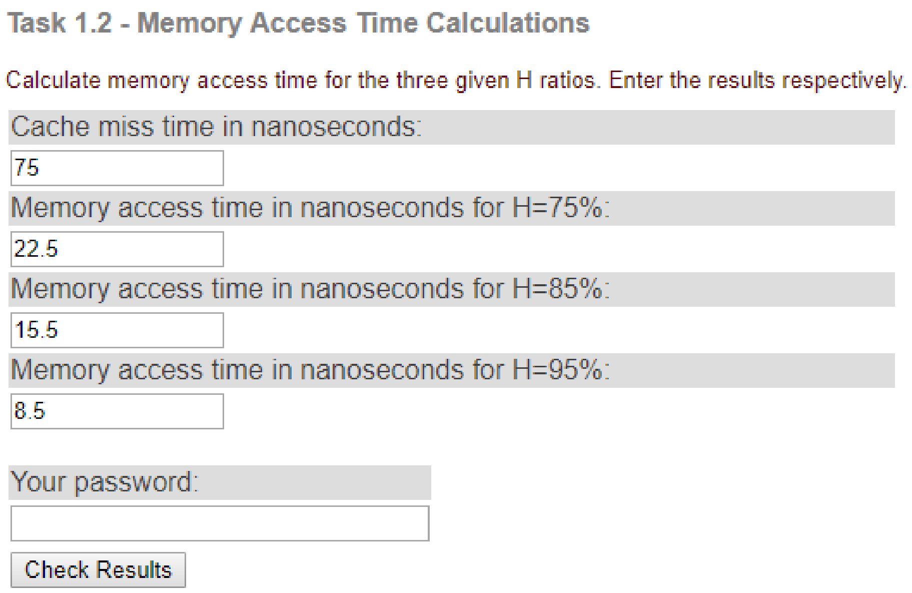
H= 85% = .85 x 5 + .15 (70 + 5) =

4.25 + 11.25 = **15.5 ns**

H= 95% = .95 x 5 + .05 (70 + 5) =

4.75 + 3.75 = **8.5 ns**

Screenshot showing correct answers for task 1.2:



Task 1.2 - Memory Access Time Calculations

Calculate memory access time for the three given H ratios. Enter the results respectively.

Cache miss time in nanoseconds:

75

Memory access time in nanoseconds for H=75%:

22.5

Memory access time in nanoseconds for H=85%:

15.5

Memory access time in nanoseconds for H=95%:

8.5

Your password:

Personal Reflection:

I learned a lot about system bit processors and the different types of storage during this week's lecture and lab sessions. This has definitely increased my knowledge about the way operating systems work, as I did not know some of this information beforehand. Learning about the different types of architectures was interesting, but not as helpful for the lab exercises. The easiest and most interesting part of my learning was the bit about calculating the cache hit ratio, as well as finding out that it is one of the fastest types of memory only behind registers. I was able to successfully calculate and finish the problems presented in task 1.2. The most challenging aspect I encountered was the pseudo-program execution. I was already familiar with the fetch execute cycle but found it difficult to integrate into the problem. However, I was not that familiar with the Von Neumann system, and it took me a few days to fully acclimatize to the requirements outside of the lecture slides. With the help of my group members and through trial and error, I was able to complete the task and I do feel like my understanding of operating systems has increased.