

|  |  |
| --- | --- |
| **ASSIGNMENT** | |
| **Course Code** | 19CSC205A |
| **Course Name** | Microprocessor & Assembly Language. |
| **Programme** | B. Tech |
| **Department** | Computer Science & Engineering |
| **Faculty** | Faculty of Engineering Technology |

#### 

|  |  |
| --- | --- |
| **Name of the Student** | SUBHENDU MAJI |
| **Reg. No** | 18ETCS002121 |
| **Semester/Year** | 3RD / 2019 |
| **Course Leader/s** | Supriya M.S. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Declaration Sheet** | | | | | | | | |
| Student Name | SUBHENDU MAJI | | | | | | | |
| Reg. No | 18ETCS002121 | | | | | | | |
| Programme | B. Tech | | | | | Semester/Year | 3rd / 2019 | |
| Course Code | 19CSC205A | | | | | | | |
| Course Title | Microprocessor & Assembly Language | | | | | | | |
| Course Date |  | | To | |  | | | |
| Course Leader | Supriya M.S. | | | | | | | |
| **Declaration**  The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly. | | | | | | | | |
| Signature of the Student | |  | | | | | Date |  |
| Submission date stamp  (by Examination & Assessment Section) | |  | | | | | | |
| Signature of the Course Leader and date | | | | Signature of the Reviewer and date | | | | |
|  | | | |  | | | | |

# **Contents**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[**Declaration Sheet** ii](#_Toc22905149)

[**Contents** iii](#_Toc22905150)

[Marking Scheme iv](#_Toc22905151)

[**Question No. 1** 5](#_Toc22905152)

[A1.1 Assembly Language Program 5](#_Toc22905153)

[A1.2 Clock cycle time, Execution time of sequence recognizer, CPI 7](#_Toc22905154)

[A1.3 AMAT 9](#_Toc22905155)

[A1.4 Comparison of Execution time 10](#_Toc22905156)

[Bibliography 12](#_Toc22905157)

|  |  |  |  |
| --- | --- | --- | --- |
| **Faculty of Engineering & Technology** | | | |
| **Ramaiah University of Applied Sciences** | | | |
| **Department** | Computer Science and Engineering | **Programme** | B. Tech. |
| **Semester/Batch** | 3rd /2018 | | |
| **Course Code** | 19CSC205A | **Course** **Title** | Microprocessors and Assembly Programming |
| **Course Leader** | P.Padma Priya Dharishini , Supriya M.S. | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assignment - 1 | | | | | | | | |
| Name of Student | | |  | Register No | |  | | |
| Sections |  | Marking Scheme | | | Max Marks | | First Examiner Marks | Second Examiner Marks |
| Part-A | A1.1 | Assembly Language Program | | | 3 | |  |  |
| A1.2 | Clock cycle time, Execution time of sequence recognizer, CPI | | | 3 | |  |  |
| A1.3 | AMAT | | | 2 | |  |  |
| A1.4 | Comparison of Execution time | | | 2 | |  |  |
|  | **Max Marks** | | | **10** | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Marks Tabulation** | | | | |
| **Component- CET B Assignment** | **First Examiner** | **Remarks** | **Second Examiner** | **Remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Marks (out of 10 )** |  |  |  |  |
| Signature of First Examiner Signature of Second Examiner | | | | |

# **Question No. 1**

**Solution to Question No. 1:**

## A1.1 Assembly Language Program



Figure 1 ASM code

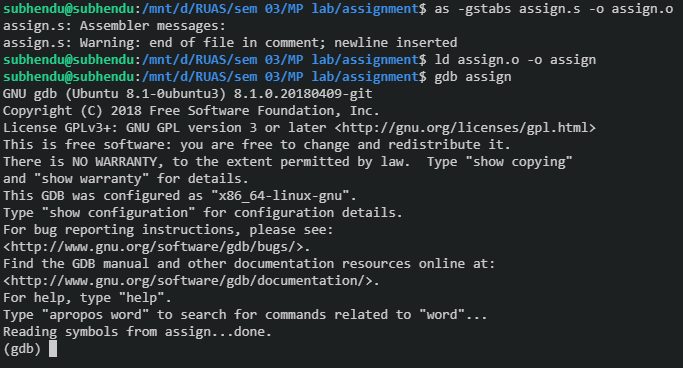


Figure 2 Making and linking file

Output when searching sequence **1011** when binary no. is **0010110110**:

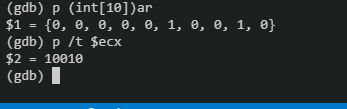


Figure 3 output for input 0010110110

Output when searching sequence **1011** when binary no. is **101101011010**:

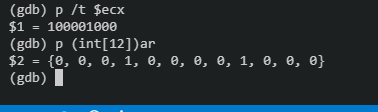


Figure 4 output for input 101101011010

Output when searching sequence **1100** and binary no. is **110011001**:

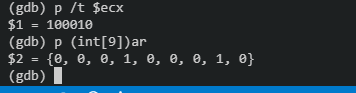


Figure 5 output for input 110011001

## A1.2 Clock cycle time, Execution time of sequence recognizer, CPI

Table 1 Clock Cycle and CPI Calculation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Line No. | | source | Destination | No. of Clock Cycles for each instruction | No. of clock cycles repeated per instruction | no. of clock cycles |
| 20 |  | Immediate | Register | 1 | 1 | 1 |
| 21 |  | Memory | Register | 7 | 1 | 7 |
| 22 |  | Immediate | Register | 1 | 1 | 1 |
|  | loop: |  |  |  |  |  |
| 25 |  | Register | Register | 1 | 10 | 10 |
| 26 |  | Immediate | Register | 1 | 10 | 10 |
| 27 |  | Immediate | Register | 1 | 10 | 10 |
|  |  |  |  |  |  |  |
| 29 |  | cmp |  | 5 | 10 | 50 |
| 30 |  | je |  | 5 | 10 | 50 |
|  |  |  |  |  |  |  |
| 32 |  | cmp |  | 5 | 10 | 50 |
| 33 |  | je |  | 5 | 10 | 50 |
| 34 |  | jne |  | 5 | 5 | 25 |
|  | equal: |  |  |  |  |  |
| 37 |  | Immediate | Memory | 1 | 5 | 5 |
| 38 |  | Immediate | Register | 1 | 5 | 5 |
| 39 |  | jmp |  | 5 | 5 | 25 |
|  | not\_equal: | |  |  |  |  |
| 42 |  | Immediate | Memory | 1 | 5 | 5 |
| 43 |  | Immediate | Register | 1 | 5 | 5 |
| 44 |  | jmp |  | 5 | 5 | 25 |
|  |  |  |  |  |  |  |
| 49 |  | Immediate | Register | 1 | 1 | 1 |
| 50 |  | Immediate | Register | 1 | 1 | 1 |
|  | loop1 |  |  |  |  |  |
| 53 |  | Memory | Register | 7 | 10 | 70 |
| 54 |  | Immediate | Register | 1 | 10 | 10 |
| 55 |  | Immediate | Register | 1 | 10 | 10 |
| 56 |  | cmp |  | 5 | 10 | 50 |
| 57 |  | jne |  | 5 | 10 | 50 |

We know,

Refer to Fig. 6 for Clock Rate.

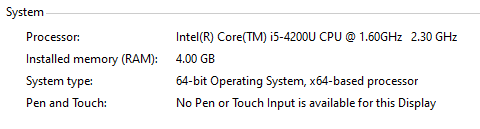


Figure 6 Clock Rate = 1.60 GHz

Note. This Calculation is only valid for specific input, i.e. **0b0010110110**. Refer to Fig. 1 for the source code.

From Table 1, The calculations are as follows:

We know,

We know,

## A1.3 AMAT

We know,

Two-level cache **AMAT formula**:

------**equation 1**

Where,

Given,

|  |  |  |
| --- | --- | --- |
| **Parameters** | **L1 cache** | **L2 cache** |
| Associativity | 2-way set Associative | 4-way set Associative |
| Block size | 32 bytes | 64 bytes |
| Cache size | 512 Kb | 1Gb |
| Hit rate | 40% | 60% |
| Miss Penalty | - | 30 |
| Hit time | 4 ns | 20 ns |

Substituting the values in equation 1,

## A1.4 Comparison of Execution time

The actual execution time can be shown by **time** command, which is as follows:



Figure 7 when using time command

Here, we are getting 0.00 sec , even after running program for twice or thrice.

And, as we can see, the inbuilt ‘**time**’ command is not so accurate. It can show up to milliseconds only. it is not capable of showing the time in Nano-seconds.

Hence, I have used a ‘**shell script**’ to find accurate execution time using ‘**date**’ command.

Refer to Fig. 5 for Source Code of the Shell script.

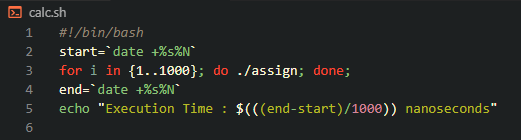


Figure 8 script file code for running the program 1000 times and getting time in Nano seconds

Basically, here in the Shell Script, I am storing a starting time and then running the program. then when the program ends, I am storing that time.

We can get an average Execution Time by **‘(end-start)/1000’**

The Shell Script is running the program for 1000 times. So, I can achieve an average accurate value up to some extent. As running the program for once or twice may not give accurate value.



Figure 9 output of running script file

Theoretically we are getting Execution time to be **328.75 ns** but practically, we are getting an average Execution time to be **5227178 ns**.

As we can see, there is a large difference between manually calculated Execution time and Execution time given by computer. This is because manually calculated time is calculated under ideal condition, but practically there a lot of background process happening which is slowing down the Execution.

An another reason can be that the program is running on WSL (windows Subsystem for LINUX). It can be faster if the main OS of the system is UBUNTU.

# Bibliography

* Refer to <https://github.com/subhendu17620/RUAS/tree/master/sem%2003/MP%20lab/assignment> for the ASM code
* https://cs.stackexchange.com/questions/95659/amat-calculation