Due Nov 19 by 11:59pm **Points** 100 **Submitting** a file upload
Available Oct 23 at 12am - Nov 19 at 11:59pm

This assignment was locked Nov 19 at 11:59pm.

. .oject 4 - SIC Clock

General Information

Purpose | Project Objectives | Possible Points | Feedback | Disclaimers

This project provides students an understanding of the SIC assembly language. In this project, students will create and implement a 12- and 24-hour clock program using the SIC assembly language.

Students are not allowed to use the SIC/XE assembly language for this project.

This project partially satisfies **Course Objective 1**. After completing this project, students will be able to:

1. Create and implement an efficient solution using the SIC assembly language
2. Streamline their assembly program to use the fewest bytes of memory possible
3. Efficiently use memory to store and manipulate values
4. Utilize system devices to transfer their program's output to other applications running in the computer
5. Produce the correct output based on flags provided in the input

This assignment is worth a maximum of 100 points.

Students can expect comments on their Project 4 submission within 5 days after the project deadline.

The hyperlinks provided throughout this project provide access to third-party resources that students will use to learn and support the material discussed within this project. The professor provides these links "**as is**"; therefore, the professor does not nor cannot guarantee or

endorse the information found at the linked sites beyond the scope of this course. Furthermore, the professor accepts no responsibility or liability for the content maintained at the linked sites. Students should report non-working links, as well as other issues with linked material, to the professor.

A thorough discussion on advanced topics related to this project are beyond the scope of this course. The professor has taken great care to ensure students have access to all the material needed to complete this project successfully. Although advanced topics related to this project are not covered, students are encouraged to explore the advanced topics related to this project, if time permits; however, failing to do so will not prevent students from completing the project successfully.

Directions & Important Notes

Directions | **Important Notes** | Required Tools | Academic Integrity

- Download the **Project 4 files (https://canvas.unf.edu/courses/93573/files/16038679?wrap=1)** **(https://canvas.unf.edu/courses/93573/files/16038679/download?download_frd=1)** , which you will need to complete the project
  - Watch the **SIC Simulator Introduction (https://youtu.be/7GcCD6wNGfs)** video
- Review the requirements listed in the **Project Requirements** section
- Create and implement a solution that meets all of the stated requirements
  - Students must create their solution using SIC assembly language
  - Students are not allowed to use the SIC/XE assembly language
- Test your project thoroughly using the provided **SIC Simulator**
  - The professor will use the **SIC Simulator** while grading your project
- Upload the **clock.sic** file to Canvas before the posted due date (refer to the Course Schedule)

Students should create a solution using the C language prior to attempting to implementing the solution in the SIC assembly language. Doing so will help students understand the concepts that are important for completing Project 4 successfully.

Students will be required to use one or more of the following tools to earn a passing grade on the project. Each of the tools listed below can be downloaded for free or already exists in the indicated operating system.

- C Development Environment
  - You are free to use your preferred C IDE and other tools
- Plain text editor
  Note: For creating your **clock.sic** file.

    - Windows
        - Notepad++
    - Mac OS
        - TextWrangler
        - TextEdit (plain-text mode)
    - Linux (terminal)
        - pico
        - vi
- SIC Simulator
  Note: Required to test your **clock.sic** file.
    - The SIC Simulator is provided in the **Project 4 files (https://canvas.unf.edu/courses/93573/files/16038679?wrap=1) (https://canvas.unf.edu/courses/93573/files/16038679/download?download_frd=1)**

Students are strongly encouraged to adhere to **UNF's Academic Integrity (https://www.unf.edu/catalog/policies/academic_integrity/)** policy while working on the projects for this course. Work that is too similar to another student's work (current or former) may receive 0 points.

Unless authorized to do so by the professor, students should avoid directly copying code from

- Websites like **Stack Overflow (https://stackoverflow.com/)**
- Repositories like **GitHub (https://github.com/)**
- Fellow students (including current or former COP3404 students)

While students are encouraged to work together to complete their projects,

- Share ideas - not code
- Do not provide direct access to files
- Do not share computing devices

Project Requirements

Preliminary Tasks

- Create a solution using your preferred C programming tools prior to attempting the project
    - This will help you understand the concepts that are important for completing Project 4 successfully
- An automatic 10-point (10%) **penalty** will be assessed for very **disorganized** code.
    - Remove the following items from your **clock.sic** file before submitting to Canvas
        - Blank lines
        - Unnecessary comments

Project Information

Introduction

UNFinished Business recently discovered a major flaw in the circuit board the company purchased for its new line of ovens. Since it would take too long to order new circuit boards based on an updated design, UNFinished Business has decided to replace the circuit board with a SIC computer and several programs written in the SIC assembly language.

UNFinished Business engineers have already started working on the timer and display programs, and you have been asked to design and implement the clock program. The timer program outputs the number of seconds since midnight, and the display program outputs the current time (12- or 24-hour format) to the oven's control panel. Your clock program must read the output of the timer program, convert and format the timer output as either 12-hour (i.e., 1015A or 1015P) or 24-hour (i.e., 1015 or 2215) time, and write the formatted time to the input of the display program.

To save money after the circuit board fiasco, UNFinished Business decided not to purchase the SIC/XE computer, which means that your CLOCK program must be completed using the SIC assembly language.

Clock Program Requirements and Limitations

- Timer program output (seconds past midnight)
  - Written to the **last word** of SIC memory
  - Range of 0 to 86399
  - **Most Significant Byte** (MSB) used to signal 12- or 24-hour format
    - 0 - **12-hour** format (i.e., 1015A or 1015P)
    - 1 - **24-hour** format (i.e., 1015 or 2215)
- Clock program process
  Note 1: The steps listed below can be performed in any logical order.
  - Define the constants and variables needed
    Note: Constants and variables can exist anywhere in the program, as needed.
    - Constant: **BYTE** or **WORD**
    - Variable: **RESB** or **RESW**
  - Read input from the **last word** of SIC memory
    - You are not required to **validate** the input value
  - Determine format to use (12- or 24-hour)
    - Refer to the MSB of the **Timer program output**
  - If 12-hour format is used, determine if **AM** or **PM**
  - Calculate the first digit of the **hour value** (i.e., 1 for **1015A** or **1015P** and 2 for **2215**)
    - If the **hour value** is less than 10, the **first digit** must be 0
  - Calculate the second digit of the **hour value** (i.e., 0 for **1015A** or **1015P** and 2 for **2215**)
  - Calculate the first digit of the **minute value** (i.e., 1 for **1015A** or **1015P** and 1 for **2215**)
    - If the **minute value** is less than 10, the **first digit** must be 0
  - Calculate the second digit of the **minute value** (i.e., 5 for **1015A** or **1015P** and 5 for

2215)
    - Write the **hour** and **minute** digits to the **output device**
        - You are not required to **test** if the device is ready
        - Use SIC Simulator device 42
    - If 12-hour format, write the appropriate **time indicator** (A or P) to the **output device**
        - You are not required to **test** if the device is ready
        - Use SIC Simulator device 42
    - Notify the display program the clock output is ready
        - Place the value 0 in the **Linkage** register (i.e., LDL ZERO)
        - Perform the **RSUB** command
- Clock program limitations
    - The submitted program cannot be **larger** than 300 bytes
        - A larger clock program will be accepted with a penalty
    - The clock program must **load** itself in the highest memory address possible
        - Compute the appropriate address for the **START directive**
        - Do not store **instructions** or **data** in the last word of SIC memory

Extra Credit

You can earn 25 points of extra credit by streamlining the size of your **clock.sic** SIC assembly language program to 256 bytes or less when loaded into the SIC Simulator. You can earn an additional 25 points of extra credit (**50 points total**) by streamlining the size of your **clock.sic** SIC assembly language program to less than 230 bytes (the size of the professor's clock.sic program).

Important Notes

- Do not attempt the extra credit until your **clock.sic** SIC assembly language program produces the expected results
- Create a backup of your **clock.sic** file before attempting the extra credit
- Your **clock.sic** SIC assembly language program must meet all project requirements and pass all tests to be eligible for extra credit

Extra Credit Requirements

- 25 points - Size of **clock.sic** program 256 bytes or less
    - Size of the program will be computed by the SIC Simulator
- 50 points - Size of **clock.sic** program below 230 bytes
    - Size of the program will be computed by the SIC Simulator

Important Material

The following material will be helpful in completing and testing your **clock.sic** program. Most of these documents are provided in the **Project 4 files (https://canvas.unf.edu/courses/93573/files**

/16038679?wrap=1) (https://canvas.unf.edu/courses/93573/files/16038679/download?download_frd=1) download.

- Example Program.sic
  - A documented example SIC assembly program to illustrate basic functionality of the SIC assembly language
- SIC Instruction Set
  - Provides a thorough discussion of the SIC instruction set and registers
- Time Datasheet
  - Provides the 12- and 24-hour output examples for a variety of times
  - Your clock.sic program may be tested with values not listed in this document
- **OnlineTimeTools.com (https://onlinetimetools.com/calculate-seconds-since-midnight)**
  - This website may be helpful in thoroughly testing your **clock.sic** program

Examples

The following examples demonstrate the output to the **Display** program your **clock.sic** program should produce given the input from the **Timer** program.

Note: Your clock.sic program may be tested with values not listed in these examples.

```
Input (HEX) Output     Comments
-----       ------     --------
000000      1200A      12-hour, MSB set to 0
000830      1234A      12-hour, MSB set to 0
002450      0234A      12-hour, MSB set to 0
00A35C      1137A      12-hour, MSB set to 0
00B0F0      1234P      12-hour, MSB set to 0
00CD10      0234P      12-hour, MSB set to 0
014C1C      1137P      12-hour, MSB set to 0
800000      0000       24-hour, MSB set to 1 (0x800000)
800830      0034       24-hour, MSB set to 1 (0x800000)
802450      0234       24-hour, MSB set to 1 (0x800000)
80A35C      1137       24-hour, MSB set to 1 (0x800000)
80B0F0      1234       24-hour, MSB set to 1 (0x800000)
80CD10      1434       24-hour, MSB set to 1 (0x800000)
814C1C      2337       24-hour, MSB set to 1 (0x800000)
```