EE 337: Microprocessors Laboratory (Spring 2023)
Indian Institute of Technology Bombay
ENDSEM: 35 points (Duration: 3 hours) Date: April 8, 2023

Instructions:

- You are **ONLY** allowed to use your codes from previous labs. Sharing your previous codes and resources to other students during exam is strictly not permitted.
- Use of internet during the course of this examination will be considered as copying and strict action will be taken.
  - **You must put your laptops in airplane mode.**
  - **No internet browser should be opened during the exam**
  - **Install a non-browser pdf reader to read pdf files during the exam.** Examples are Acrobat Reader, Nitro PDF Reader, Sumatra PDF Reader.
- Any discussion during exam is not permitted.
- Lock your screen at the end of 3 hours at 5:15 PM sharp. Any student found writing code after the exam duration will face penalty.
- At the completion of **every question**, **show your solution to an evaluating TA** for evaluation.
- Check the rubrics and show the outputs as and when completed in-order to get partial markings.
- Create separate projects for each question. For submission, create a single zip file consisting of all your projects and name it as `rollnumber_name.zip`.

1. [15 points] Two students Elizabeth and Scarlet have designed different multiplication algorithms to be implemented on Pt-51 board.

   Algorithm designed by Elizabeth:

```
input p, q
p1 = p/10
p0 = p%10
q1 = q/10
q0 = q%10
r0 = p0*q0
r1 = (p0+p1)*(q0+q1)
r2 = p1*q1
r = r0 + (r1 - r0 - r2)*10 + r2*100
output r
```

   Algorithm designed by Scarlet:

```
input p, q
p1 = p/10
p0 = p%10
q1 = q/10
q0 = q%10
r0 = p0*q0
r1 = p0*q1
r2 = p1*q0
r3 = p1*q1
r = r0 + (r1 + r2)*10 + r3*100
output r
```

   Your task is to figure out whose algorithm performs better.

   - Implement both these algorithms in C independently (2 separate programs). Inputs can be assumed to be in the range 0 to $2^{16} - 1$. Use long unsigned int for all variables including inputs.

- Directly give inputs $p = 9653$ and $q = 43801$ for both the programs.
- Use timer to measure the time taken to perform a single multiplication. You need to start the timer just before the algorithm is started and stop the timer just after the algorithm is completed.
- Finally, display the multiplication result and the time measured (in μs) by the timer on LCD in the following format respectively after each algorithm execution. You need to display 9 digits of decimal product output and 5 digits of time measured (in μs) for each algorithm.

```
Prod1=_________        Prod2=_________
Time1=_____us          Time2=_____us
```

Which algorithm is faster? What do you think is the reason for the difference in performance?

2. [20 points] FPGAs available in Wadhwani Electronics Lab (WEL) are listed in Table 1.

FPGA	Index	Quantity (MAX)
Xenon-10	1	8
Arty-A7	2	5
De0-Nano	3	8
Pynq-Z2	4	2

Table 1: FPGAs

These FPGAs can be borrowed by students. Store this table database in your C-program. You can either use arrays (recommended) or just individual variables.

You need to read input from keyboard on RealTerm and send it to the Pt-51 kit using USB to UART converter. You need to take care of the following cases three states of operation.

* **Initial state**

- Upon resetting the Pt-51 kit, the program should enter the Initial state and display the maximum number of FPGAs available on RealTerm.
  `FPGAs available:  Xen10-8 Arty-5 De0-8 Pynq-2.`
- Whenever it is in the Initial state, two tasks are to be done:
  i The current availability of FPGAs must be displayed on RealTerm (number of current available micro-controllers after issuing/returning need to be displayed).
  `FPGAs available:  Xen10-x Arty-x De0-x Pynq-x.`
  ii Then RealTerm must show the following message.
  `Press I for Issue and R for Return`
- Based on the key pressed, one of the other two states must be entered. If a different key is entered, the program should remain in Initial state.

* **Issue state**

- This state is for issuing items. It is entered when the user types the character "I" through UART. When this happens, details must be asked from the user.
- Display the following message- `Enter FPGA to be borrowed:` and read character from UART. The index of the FPGA (refer Table 1.) to be borrowed must be entered.
- Then display the following message- `Enter Quantity:` and read character from UART. The quantity of FPGAs to be borrowed must be entered.
- Once the inputs are read, display `Requested FPGA allocated!` and decrement the count of the corresponding FPGA from the database.
- If the requested FPGA of requested quantity is/are not available, then display- `Requested FPGAs not available...`

- **Return state**
  - This state is for returning items. It is entered when the user types the character "R" through UART. When this happens, details must be asked from the user.
  - Display the following message- `Enter FPGA to be returned:` and read character from UART. The index of the FPGA (refer Table 1.) to be returned must be entered.
  - If the returned resource is already maximum (i.e it was not borrowed in the first place), then display- `You can't return what you don't have...` and exit the state. Else follow the next steps.
  - Then display the following message- `Enter Quantity:` and read character from UART. The quantity of FPGAs to be returned must be entered.
  - Once the inputs are read, display `Returned FPGA received!` and increment the count of the corresponding FPGA from the database.
  - If the sum of the returned FPGA count and existing FPGA count goes beyond maximum (i.e it is not possible to return that many FPGAs), then display- `Returned FPGA out of bounds...`

After the functions of Issue state or Return state, the execution must return to Initial state.
To convert character to integer, you can subtract 48 from the character received.

. . . .

RUBRICS:

Please show state-wise progress to your TA. This is to help us give you partial marks. You might have some part working and later your code may not work due to the changes you have made. In that case, you can get partial credit if you show the previous working part to your TA.

Question 1:

- 2 marks for displaying correct product for first algorithm.
- 4 marks for displaying time for first algorithm.
- 2 marks for displaying correct product for second algorithm.
- 4 marks for displaying time for second algorithm.
- 1 mark for correct answer for faster algorithm.
- 2 marks for correct reasoning.

Question 2:

- Correct working of Initial state and returning to Initial state : 6 marks
- Correct working of Issue state : 6 marks
- Correct working of Return state : 8 marks