Western Engineering

ECE 9047 Sensor Networks & Embedded Systems

Laboratory 1

Course Instructor: Dr. John McLeod

Student Number: 251343797 .

Student Name: Yuanpeng Wang .

Table of Contents

1

Description of the problem

Write the first 10 Fibonacci numbers to a seven segment display. Show each number on the display for one second (1 s) before showing the next. Your program should endlessly loop, cycling through displaying these 10 numbers.

The Fibonacci numbers are defined as F_n:

$$F_0 = 0,\ F_1 = 1, and\ \ F_n = \ F_n - 1 + \ F_n - 2\ \ For\ n > 1$$

2

The flowchart of my solution

Initialization:

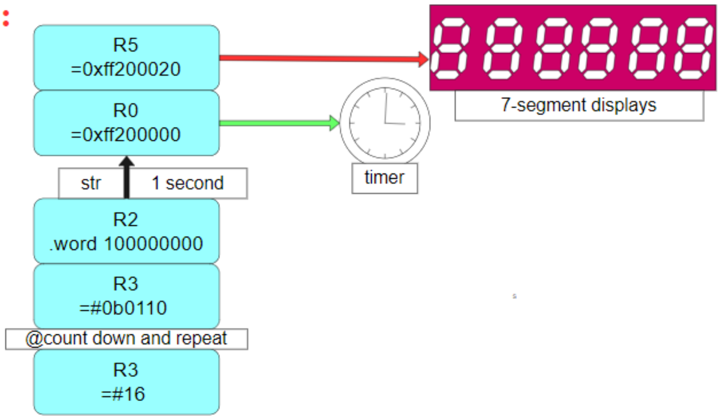

Fig. 1 Initialization

Main Loop:

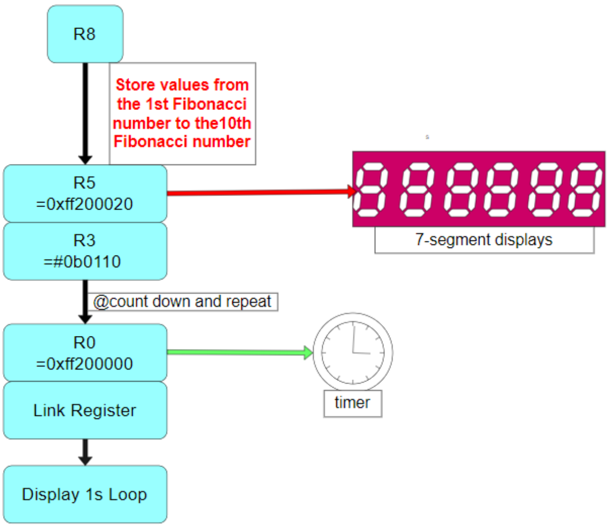

Fig. 2 Main Loop

Display 1s Loop:

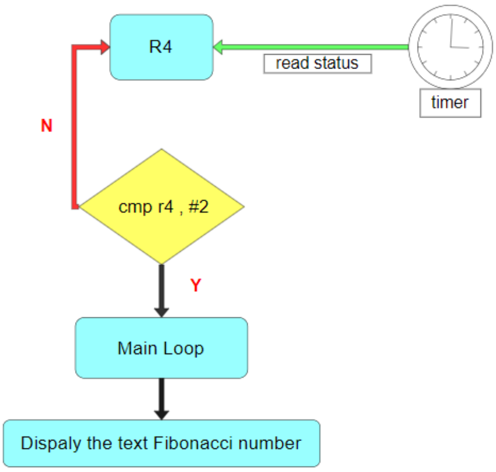

Fig. 3 Display 1s Loop

3

Description of my approach and validation

Description of my approach:

1. Initial Setup:

- Storing 7-segment display base address to the register.
- Storing timer base address to the register.
- Loading the timer interval.
- Setting the high period of the Timer.
- Initializing timer for the countdown.

2. Main Loop:

- Loading the display values sequentially.
- Writing to the display to the 7-segment display peripherals.
- Logic control for pausing for 1 second in the subroutine using the link register.

3. The ‘display_1_second’ subroutine

- Checking the timer status to determine whether or not to keep looping.
- Returning to Main Loop using the link register

Validation:

When I saw the 7-segment displays could display the first 10 Fibonacci numbers sequentially and pause 1 second for each interval.

4

A qualitative cost/benefit analysis of your solution

Design and Implementation:

The time spent designing this solution is relatively brief, as the approach is logically divided into three main parts: Initialization, Main Loop, and Delay 1s Loop, making it quite approachable. The design concept requires a clear understanding of the hardware interface, specifically the timer and the memory-mapped registers of the 7-segment display. It is essential to know how to initialize and control the timer, manage the display segments to represent numbers and structure the program to cyclically display a sequence of numbers at 1-second intervals. This understanding necessitates both knowledge of the hardware specifications and proficiency in assembly programming. Furthermore, this design approach demonstrates my comprehensive grasp of the first five weeks of the ECE 9047 course content.

The implementation phase also took a relatively short amount of time, as it primarily involved configuring registers and coding for memory mapping. This approach results in a lean, highly optimized solution that directly interacts with the hardware, which is crucial in embedded systems or real-time applications where performance and timing are of utmost importance.

Maintenance and Extension:

The time required for future maintenance or extension of this code could be minimal.

First, users can easily modify the code in the main loop to change to any desired number.

Second, the code's portability is excellent, as users can add other peripherals' addresses in the initialization and set the desired time interval for the timer, enabling the code's reuse in other projects to generate pauses.

Third, to generate the first 100 Fibonacci numbers, users only need to make slight modifications to the code in the main loop.

Last but not least, the code is highly readable, thanks to its nearly linear structure (it has only one branch) and industry-standard comments.

In summary, this code offers low operational overhead and saves time in future maintenance or extensions