EMBEDDED SYSTEM DEVELOPMENT

Micro Control Logic

SAGAR PATEL

7441652

18TH

2017

CONESTOGA COLLEGE

2.0 INTRODUCTION

Control logic is a key part of a software program that controls the operations of the program. The control logic responds to commands from the user, and it also acts on its own to perform automated tasks that have been structured into the program. It can be modeled using a state diagram which can also combine with flow charts to explain complex control logic.

There are two types of control logic: 1) Hard wired and 2) Micro-programmed

Here, we just care about the micro-programmed control logic. In that, control information stored in to special memory. If we want to change the logic than we just need to make changes in memory contents.

Micro-programs consist of series of microinstructions, which control the CPU at a very fundamental level of hardware circuitry. Normally, it stored in ROM as it's contents is not changed frequently.

Microcode is a technique that imposes an interpreter between the hardware and the architectural level of a computer. The microcode is a layer of hardware-level instructions that implement higher-level machine code instructions. Microcode is used in general-purpose central processing units.

2.1 Design Objective

Make a control logic and to test it build a microcode program that does the following:

- Load various registers and perform some basic operations related to ALU and SHIFTER.

2.2 Block Diagram

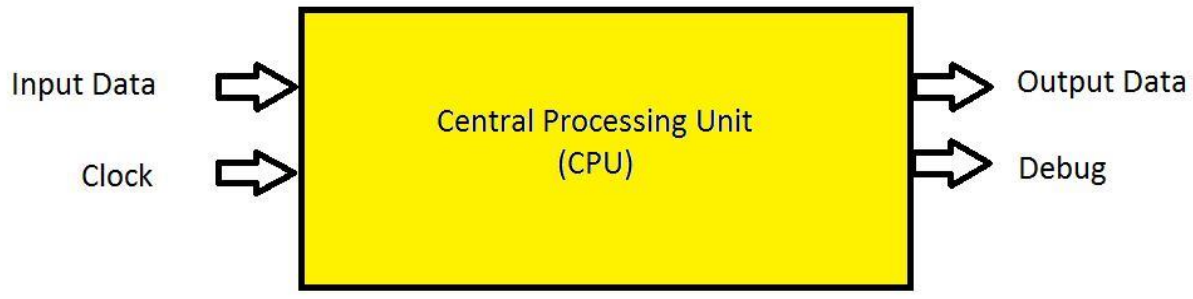


Figure 2.1 Simple block diagram shows input and output

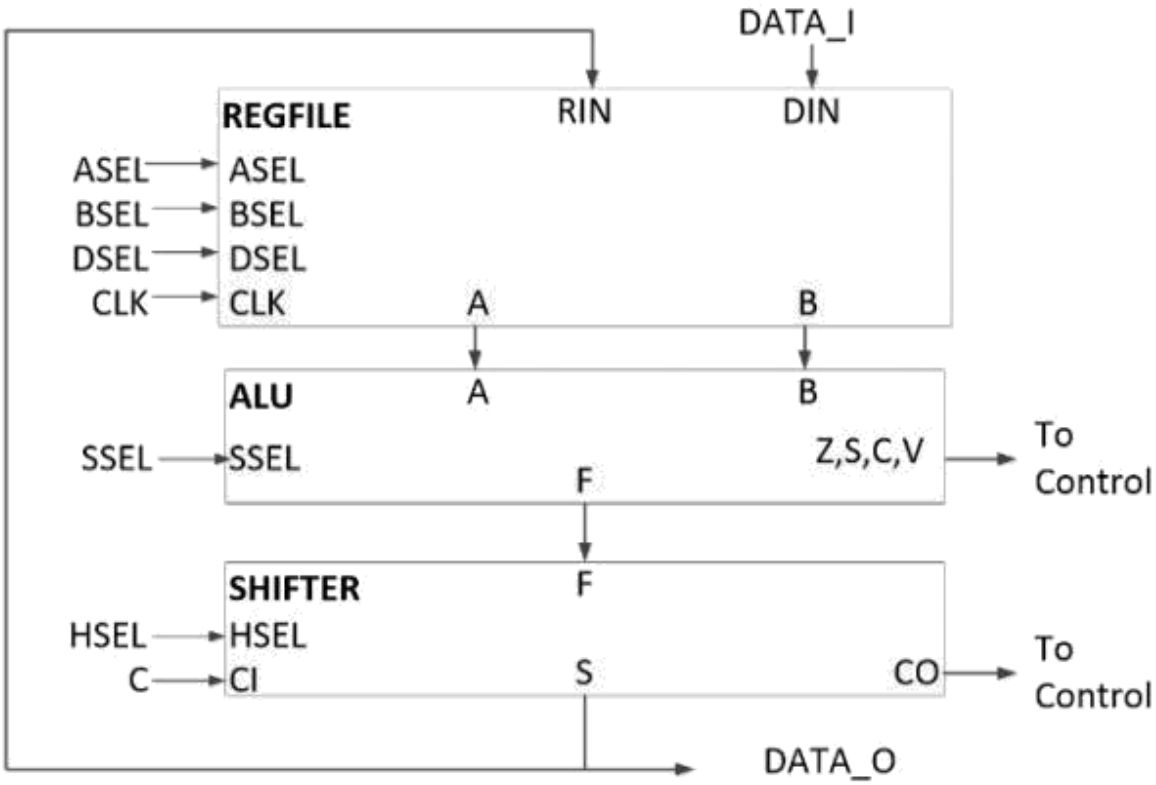


Figure 2.2 Detailed block diagram

2.3 Program of my choice

I would like to do programming in VHDL because from beginning I am doing programming in this language. So, I can say that I have a good hand on this language rather than verilog.

It considers the below files:

- **Register File :**
  Input → ASEL, BSEL, DSEL, RIN, DIN, CLK
  Output → Contents of ASEL and BSEL
- **ALU :**
  Input → SSEL (Selection of arithmetic operation which we want to perform), contents of ASEL and BSEL from Register File
  Output → F (Result of operation), Status of carry, zero, sign and overflow flag

- **Shifter :**

Input → HSEL (Selection of shifter operation which we want to perform), Carry bit, F (From ALU file)
Output → S (Result of operation), Carry out bit

- **Microcode:**

It is provided by instructor.

- **CPU:**

It performs operations that described in microcode file.

3.0 SPECIFICATION & REQUIREMENTS

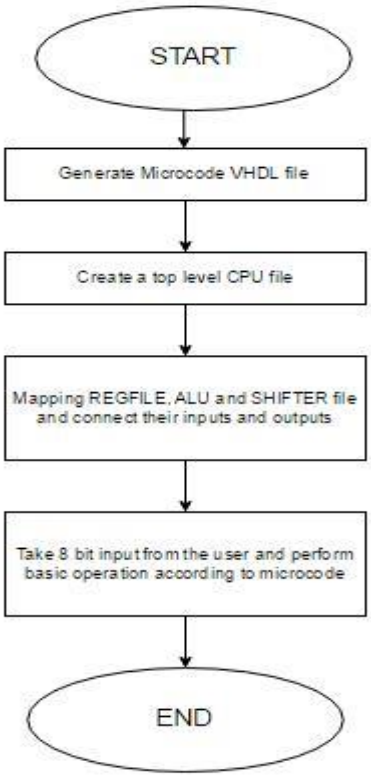
- Using a block sensitive to the rising edge of the clock implement:
  - The logic to implement MUX1 and MUX2
  - The logic to implement CAR
- Write a microcode that do some basic operations to prove REGFILE, ALU and SHIFTER work.

4.0 IMPLEMENTATION METHOD

I am going to follow below steps to achieve my objective:

- Test the previous lab code of REGFILE, ALU and SHIFTER.
- Create a top level entity named CPU and merge REGFILE, ALU and SHIFTER file under it.
- Generate a microcode file by the instruction given by instructor.
- Take the DIN (Input Data) from user.
- Perform the basic operations mentioned in microcode file.
- Show the output on 7 segment display that microcode is working.

4.1 Design Flow



4.1.1 Available File

- ✓ REGFILE
- ✓ ALU
- ✓ SHIFTER
- ✓ MICROCODE

4.1.2 Actions need to do to combine

I am going to use instantiation method to combine my files that shown in figure 2.2.Firstly, I need to create the CPU file named project. After that I need to create VHDL file of microcode. At the end, to combine REGFILE, ALU and SHIFTER in CPU I have to add those file in to the CPU project directory. Than go to setting and those file name and lastly I just need to write instantiation of those files in my main CPU file.

4.2 Technical Questions/Problems and Answers

4.2.1 TO DO LIST

Q.1 How to do instantiation?

Q.2 How can I generate microcode vhdl file?

Q.3 Where should I have to give CLK signal. Should I have to define commonly in program or somewhere else?

Q.4 What should I have to display on 7 segment display?

Q.5 Where the input DIN goes in REGFILE? If we go back to lab3 register file than we can see in lab diagram that DIN goes to ASEL and BSEL on 0 bit. So where we put the DIN value? And it's a 8 bit value so how can we put it onto a 0 bit on ASEL or BSEL.

Q.6 How to initialize the registers? Do I need to initialize in REGFILE or I need to do that by using microcode?

Q.7 What is a use of RIN?

Q.8 What to do with other status bits zero ,sign and overflow?

Q.9 Are we using a MUX1 in this microcode?

Q.10 If we get carryout in shifter output than where does that go? In figure it says to control, but what does that mean?

4.2.2 ANSWERS

A.1) http://insights.sigasi.com/tech/four-and-half-ways-write-vhdl-instantiations.html. On this website it shows step by step that how we can do instantiation of multiple files into one file.

A.2) After go through labs slides than I found that I have to first access microcode file from command window. Than it would automatically compile generate .v and .vhdl file.

A.3) From figure 2.2 , I can clearly see that I have to give clock in REGFILE. But in lab6 slides in assessment section it says that I need to declare in main program so that I can give a positive edge clock.

A.4) Professor told me to display anything you want. But you should have to display those kind of values from which we can say that your code is working properly or not.

A.5) Still I don't know how to do that. I discussed that my question with my lab partner Emily Adams but I don't get the explanation that I want.

A.6) We can initialize registers in REGFILE or we can initialize them by using switch input(DIN).

A.7) RIN is use to load the shifter output in register if we do any shifter operation.

A.8) other status bits used to load the address in CAR or use to increment the CAR.

A.9) No, for lab6 we are not using MUX1 but for our final project we need MUX1.

A.10) After doing some shifting operation if we get carryout than we used it to control the CAR register as per my think.

4.3 Test plan for End-to-End testing

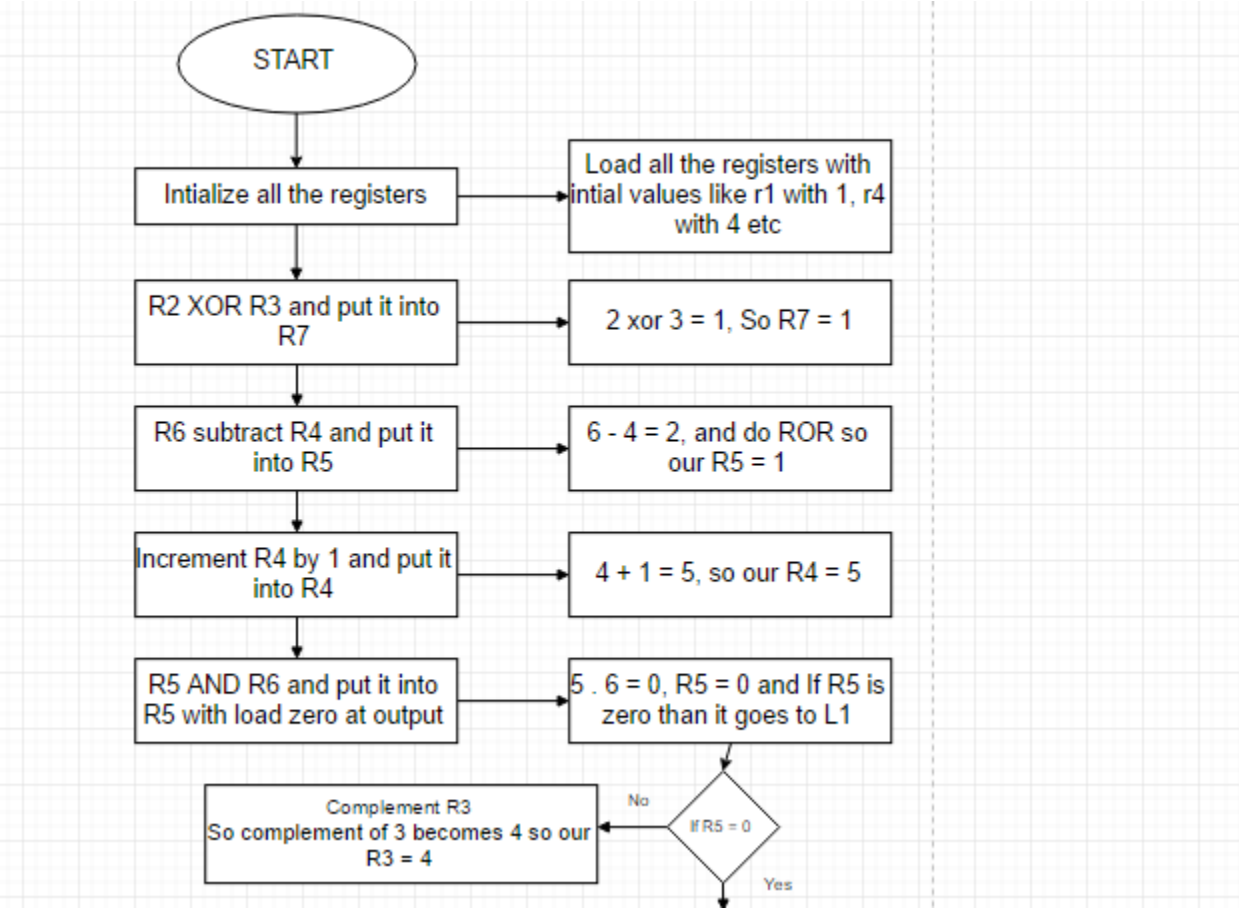
After getting result we need to check is it correct or not. So, for that I am going to follow below steps:

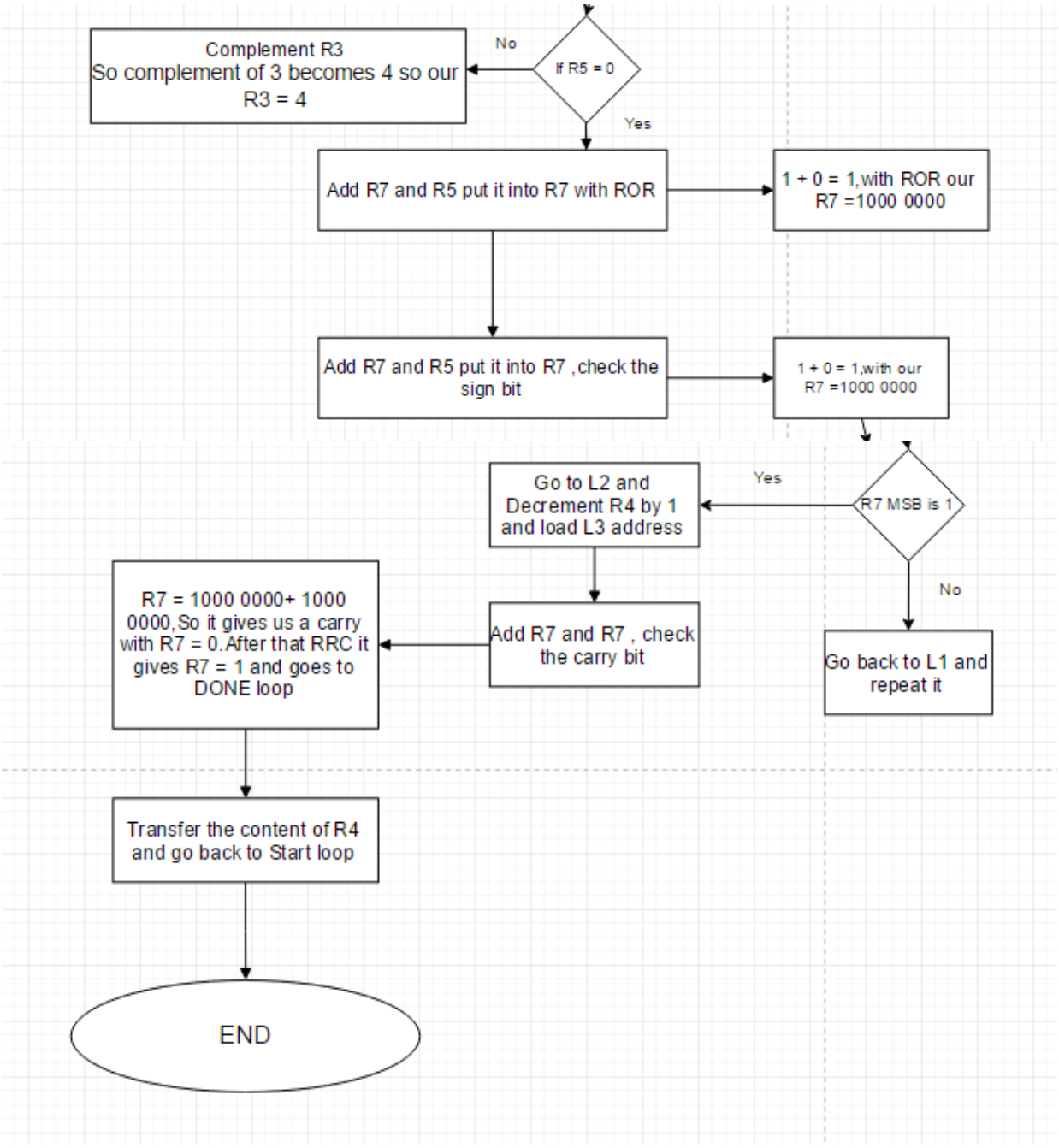
1. Compile the VHDL code that I had written
2. Show the value of ASEL and BSEL on 7 segment display
3. Show the contents of ASEL and BSEL 7 segment display
4. Show the result of ALU on 7 segment display
5. Show the result of shifter on 7 segment display
6. Store the result of shifter in DSEL and verify the values stored in proper address of DSEL or not
7. Check the MUX2 conditions and verify it

I am going to use following microcode as my test plan:

```
START:  -  \ -  \ -   \ TSF \ ZERO \ -  \ NEXT \ -    \ - ;  -- 0
        R2  \ R3  \ R7   \ XOR \ NSH \ -  \ NEXT \ -    \ - ;  -- 1
        R6  \ R4  \ R5   \ SUB \ ROR \ -  \ NEXT \ -    \ - ;  -- 2
        R4  \ -  \ R4   \ INC \ NSH \ -  \ NEXT \ -    \ - ;  -- 3
        R5 \ R6 \ R5   \ AND \ ZERO \ -   \ LZ \ L1    \ - ;  -- 4
        R3 \ - \ R3   \ COM \ NSH \ - \ NEXT  \ - \ - ;       -- 5
L1:     R7  \ R5  \ R7 \ ADD \ ROR \ - \ NEXT  \ -   \ - ;    -- 6
        R7  \ R5  \ R7 \ ADD \ NSH \ - \ LS  \ L2   \ - ;     -- 7
L3:     R7 \ R7  \ R7   \ ADD \ RLC \ -   \ LC  \ DONE \ - ;  -- 8
L2:     R4 \ - \ R4   \ DEC \ NSH \ -   \ LAD \ L3    \ - ;  -- 9
DONE: R4 \ -  \ R4 \ TSF \ NSH \ - \ LAD  \ START     \ - ; --A
```

RESULT FLOW:





5.0 Resources and reference

1) https://en.wikipedia.org/wiki/Microcode
2) Computer Engineering Hardware Design, M. Morris Mano, Prentice Hall, ISBN 0-13-162926-3