

MELZG554 HSCD LAB ASSIGNMENT 1

HSCD of SIMPLE ISA



Submitted by,

Anand S – 2021HT80003

# Contents

[Contents 2](#_Toc118709052)

[Table of Figures 3](#_Toc118709053)

[Question 1: Intro to Partial Reconfiguration Flow Lab 4](#_Toc118709054)

[Q1.a: Screenshot of the adder VHDL module with updated code 5](#_Toc118709055)

[Q1.b: Screenshot of device view after floorplanning and partition pins are introduced 5](#_Toc118709056)

[PR\_verify result 8](#_Toc118709057)

[Q1.c: Screenshot from file manager 8](#_Toc118709058)

[Q1.d: Screenshot of verification of addition/subtraction operations at the serial terminal window 10](#_Toc118709059)

[Q1.e: Screenshot of Zedboard using the camera app in the RLab PC after programming Config\_addleft\_pblock\_reconfig\_leds\_partial.bit 10](#_Toc118709060)

# Table of Figures

[Figure 1 Assignment question with considerations 4](#_Toc118709061)

[Figure 2 File name: ./Sources/rModule\_addsub/adder/adder.vhd 5](#_Toc118709062)

[Figure 3 Introduction of Partition Pins (shown in white blocks). data\_out[5] is highlighted. This is the placed design view of ‘reconfig\_addsub’ module 6](#_Toc118709063)

[Figure 4 For reconfig\_leds(Bottom-left box), SNAPPING\_MODE was ON. This resulted in whole colum going through the pblock being covered, no static logic is placed here. 6](#_Toc118709064)

[Figure 5 Same picture as above, but with the region highlighted in yellow. Upper highlighted sector has Partition pins placed. Bottom sector has the actual pblock defined. 7](#_Toc118709065)

[Figure 6 PR\_Verify summary 8](#_Toc118709066)

[Figure 7 Screenshot of the files present as Bitstreams 9](#_Toc118709067)

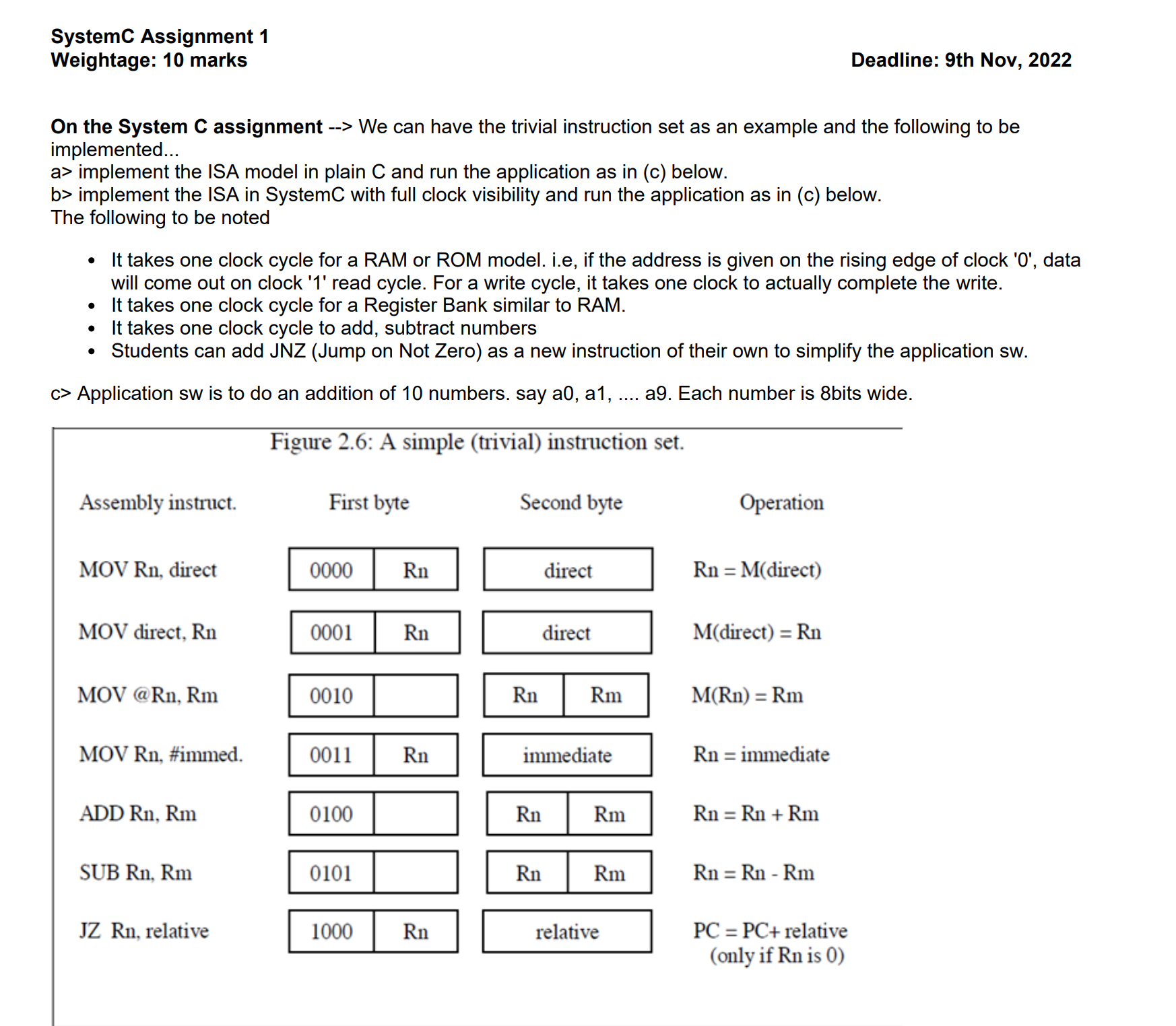
[Figure 8 md5 checksum of each bitstream file in file manager 9](#_Toc118709068)

[Figure 9 Screenshot of addition and subtraction 10](#_Toc118709069)

[Figure 10 ZedBoard screenshot 11](#_Toc118709070)

[Figure 11 Close-up screenshot of zedboard functioning 11](#_Toc118709071)

# Question



# PART C: Application code to add ten numbers



Figure Application code to add 10 numbers

The SW adds 10 numbers 0, 1, 2, 3, … 9 to get a sum of 45 (0x2d)

# PART A: Implement the ISA model in plain C and run the application code in PART C







Figure C code for the simple ISA

Text

Description automatically generated

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

Figure Execution log of the C model output

Thus the C code was executed for the application SW by coding the assembly into the instruction memory array. The execution of the code showed the final value of R3 = 8’d45 = 8’h2d, which is the expected output. Thus the execution was verified.

# PART B: Implement the ISA as timing accurate HW model

## Architecture of the simple\_isa CPU

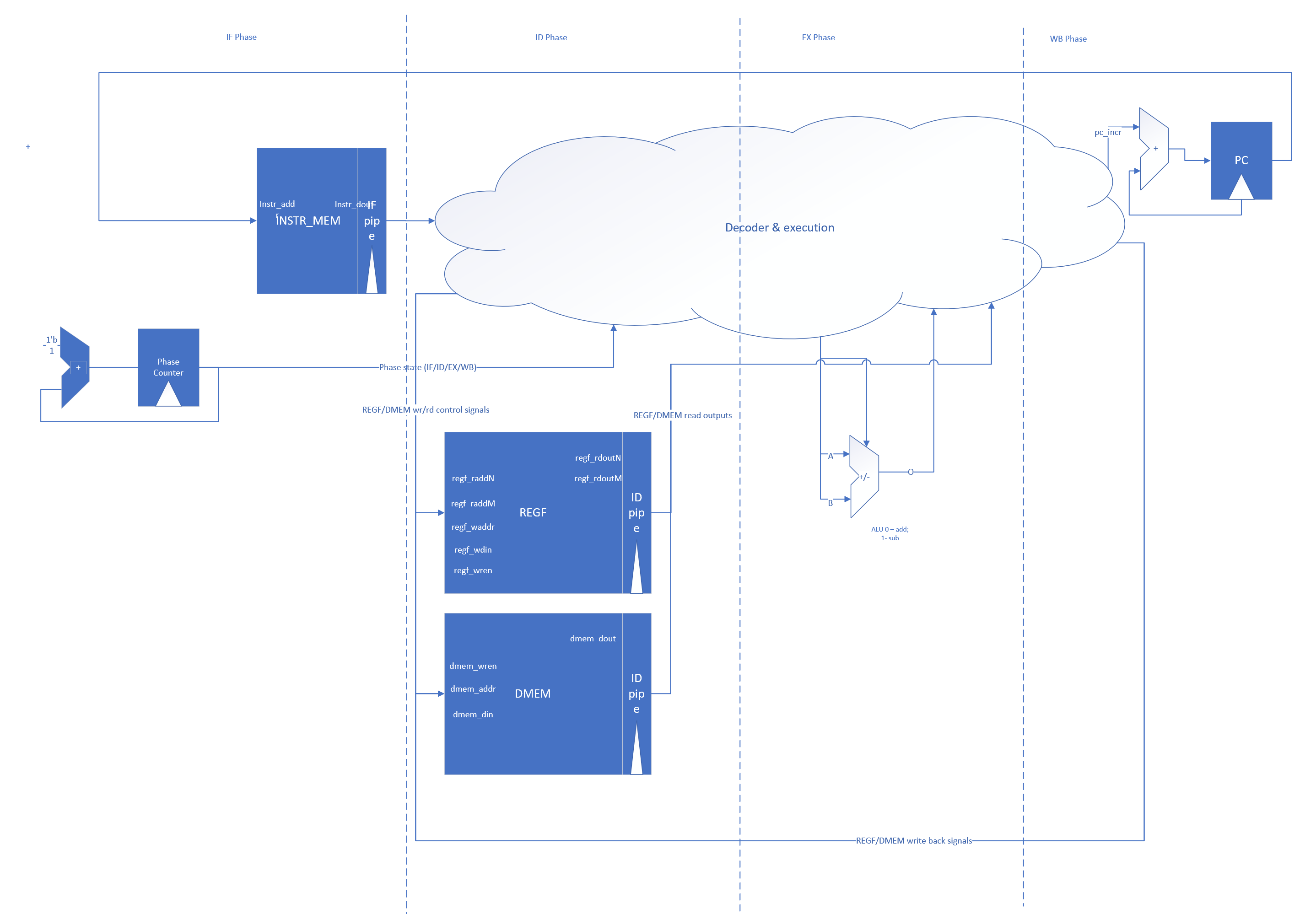


Figure High level arch of the simple\_isa CPU

## Implementing systemVerilog design for accurate timing

THE END.

ANAND S

2021HT80003