



## Digital Design Verification

**Weekly Task # 1 – RISC-V Assembler**

**Release: 1.0**

**Date: 21-May-2024**

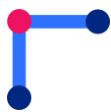
**NUST Chip Design Centre (NCDC), Islamabad, Pakistan**



**Copyrights** ©, NUST Chip Design Centre (NCDC). All Rights Reserved. This document is prepared by NCDC and is for intended recipients only. It is not allowed to copy, modify, distribute or share, in part or full, without the consent of NCDC officials.

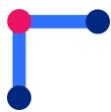
## Revision History

Revision Number	Revision Date	Revision By	Nature of Revision	Approved By
1.0	22/05/2024	Ali Aqdas	Complete manual	-



## Contents

Objectives.....	3
Tools.....	3
Lab Task.....	



## Objectives

The objectives of this lab are to:

- Understand Computing Systems
- Design a RISC-V Assembler

## Tools

1. C++ (GNU Toolchain)
2. VSCode

## Instructions

The submissions must include the appropriately commented source code file and a compiled binary.

## Task

Design an assembler to convert RISC-V Assembly to machine code. The assembler shall have the following specifications

1. Assemble the instructions listed in the [RISC-V Summary](#).
2. Be able to output the machine code either in hexadecimal format “**0x12345637**” or in binary stream.
3. Shall take command-line arguments to specify the assembly source file, destination binary file and output type.

```
$ ./assembler <source-file> <destination-file> <output-type>
```

The output type **hexadecimal** shall be identified with the flag **-h** and binary stream shall be identified with the flag **-b**. More information the using command line arguments in C++ is available in [Command Line Arguments in C](#).

## Expected Input

```
main:  
    lui a2,0x12345  
    addi a2,a2,0x678  
    .  
    .  
    .
```

## Expected Output

0x12345637 0x67860613 . . .	OR	#Ecpg . . .
-----------------------------------	----	----------------

