CS220
Introduction to Computer Organisation
Lab 4

Amey Karkare

1 Introduction

This is the second lab for the design of SMIPS processor. This lab requires you to complete the implementation of Decoder.

2 Using the provided code

Inside the lab directory, you should find the following files and directories:

```
Decode.bsv    Proc.bsv       TestBench.bsv  smips
MemTypes.bsv  ProcTypes.bsv  Types.bsv      share
```

for this lab, you only have to modify one bsv file: `Decode.bsv`. Other bsv files are for your reference to understand the design of SMIPS. **Do not** modify other bsv files as the compilation ignores them, and uses appropriate libraries from `share` directory.

The `smips` script will take care of everything from compiling the SMIPS core and cross-compiling the benchmarks for SMIPS, to running the benchmarks. The commands remain the same, and are summarized here for quick reference:

```
./smips -c all -r                    # to compile and run all benchmarks
./smips -c _path_to_benchmark_ -r    # to compile and run a specific benchmark
./smips -x                           # to clean up the build
./smips -h                           # help
```

3 SMIPS Decoder implementation

The *incomplete* `Decoder.bsv` contains several lines marked `// fixme`. Your job is to fix these lines.
NOTE: You need to consult the SMIPS manual (provided with lab 3) to fix the decoder. You also need to consult BSV files `MemTypes.bsv, ProcTypes.bsv, Types.bsv` to use the named constants, opcodes, helper functions properly.

Exercise (30 point): Complete the implementation of function `decode` in `Decode.bsv`.

Discussion (0 point): No discussion question :-)

While the implementation of register file and execute module is provided in binary format in the `share` directory, you can use your own `RFile.bsv` and `Exec.bsv` from lab 3. Ask on canvas for how to do it.