

# CSE 331/503

## Computer Organization

### Final Project – MiniMIPS Design

**Due Date 09/01/2022 Sunday 23:59**

In this assignment you will design a small version of MIPS processor. Its name is MiniMIPS. In MiniMIPS, there are 8 registers each holding 32 bit numbers. The instruction width is 16 bits instead of 32 bits in MIPS. There are two types of instructions in MiniMIPS.

#### R'-type:

Op (4bits)	Rs (3bits)	Rt (3bits)	Rd(3bits)	Func(3bits)
------------	------------	------------	-----------	-------------

#### I'-type:

Op (4bits)	Rs (3bits)	Rt (3bits)	Imm (10bits)
------------	------------	------------	--------------

The instructions of MiniMIPS are given in below table:

Instr	Opcode	Func
AND	0000	000
ADD	0000	001
SUB	0000	010
XOR	0000	011
NOR	0000	100
OR	0000	101
ADDI	0001	XXX
ANDI	0010	XXX
ORI	0011	XXX
NORI	0100	XXX
BEQ	0101	XXX
BNE	0110	XXX
SLTI	0111	XXX
LW	1000	XXX
SW	1001	XXX

They all work similar to the actual MIPS Green Sheet. As immediate field is 10 bits you should extend it to 32 bits. Do not forget, only the instruction width and number of registers are different remaining part is same as MIPS. The register contents are 32 bits. Zero register is always 0. All other rules in MIPS are also rules of MiniMIPS.

**Submission Rules:**

- **Platform: Teams** CSE 331 page
- For late submission: -20pt for each day, end of the 5 days submission will be closed.
  - Submission format must be as follows:
  - StudentNo.zip -> StudentNo (folder) -> StudentNo.qar, Report\_StudentNo.pdf

**Report file must include:**

- You need to write a testbench module and show testbench results for each module with SS.
- For MiniMIPS part, you need to try at least two different numbers for all operations (OpCode:0000 x1+x2 result= , ..., OpCode =1001 x1|x2 resut=...,  
OpCode:0000 x3+x4 result= , ..., OpCode =1001 x3|x4 resut=...)
- Explain if you have missing parts, bonus parts or adding parts.
- 2-3 pages will suffice

**Design & Codes Rules:**

- You will use Structural or Dataflow Verilog.
  - Behavioral Verilog is not allowed other than memory components. Only register and memory modules can use behavioral Verilog because of register assignment.
  - For testbench structure, please follow DraftForHW4.qar project.
-