CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

Please hand write and sign statements affirming that you will not cheat here and in your submission:

"I will neither give nor receive unauthorized assistance on this LAB. I will use only one computing device to perform this LAB".

OBJECTIVE

1. *Implement in VHDL MIPS Instructions shown in the table.*
2. *Verify correctness in ModelSim Simulation using waveforms for several select cases*
3. *Compare the execution of instructions you have designed with corresponding MIPS instructions in MARS environment.*

What to do:

First, Create registers:

- *Three 32-bit registers RT,RS,RD,*
- *16-bit IMM16,*
- *32-bit Memory Address Register(MAR)*
- *32-Bit Memory Data Register (MDR)*

Second, Design arithmetic logic unit comprised of ADD/SUB and Bitwise operations, with flags.

Third, Run simulations and verification s for all instructions and compare with MIPS instructions in MARS.

CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

The instructions are taken from "green pages"

Name	Mnemonic	Format	Operation
Arithmetic			
Add	add	R	R[rd] = R[rs] + R[rt]
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + SignExtImm
Add Unsigned	addu R	R	R[rd] = R[rs] + R[rt
Subtract	sub	R	R[rd] = R[rs] - R[rt]
Subtract Unsigned	subu	R	R[rd] = R[rs] - R[rt]
Bitwise Logical			
And	and	R	R[rd] = R[rs] & R[rt]
And Immediate	andi	I	R[rt] = R[rs] & ZeroExtImm
Nor	nor	R	R[rd] = ~ (R[rs] |R[rt])
Or Immediate	ori	I	R[rt] = R[rs] | ZeroExtImm
Shift Left	sll	R	R[rd] = R[rt] << shamt
Shift Right	srl	R	R[rd] = R[rt] >> shamt
Shift Right Arith	sra	R	R[rd] = R[rt] >>> shamt
Memory access			
Store Word	sw	I	M[R[rs]+SignExtImm] = R[rt]
Load Word	lw	I	= R[rt] R[rt] = M[R[rs]+SignExtImm]

Components you need:

5 32-bit registers: RS,TR,TD,MAR,MDR
1 16-bit register for immediate field, or for shift amount

1 2:1 32-bit MUX
1 1:2 32-bit DEMUX

1 ADD/SUB unit with flags Overflow, Negative, Zero

1 Bitwise operation unit

CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

Part I.A

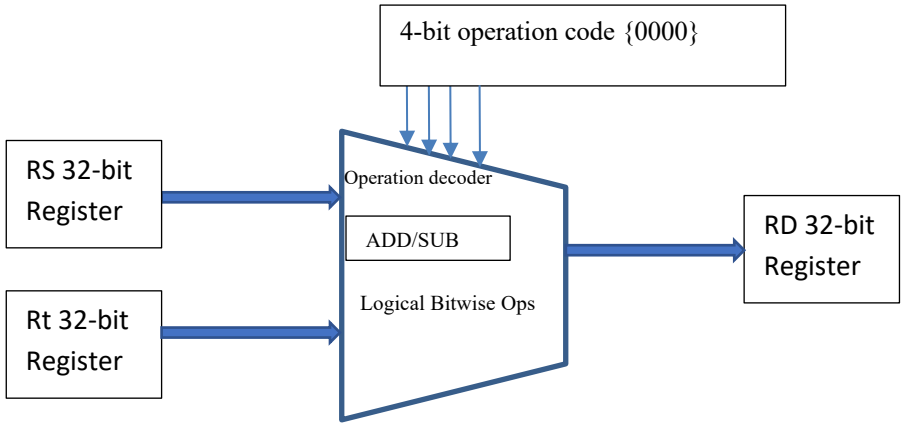
Design R Format Instructions

DATA FLOW (DATA PATH FOR R FORMAT INSTRUCTIONS)

Operations

1. *add*
2. addu
3. sub
4. subu
5. and
6. nor
7. or

R[rd] = R[rs] operation R[rt]



ToDO:

1. Use or Design add/sub including flags Z,N,O as in self-check lab you have done
2. Design Logical bitwise ops unit
3. Design operation decoder unit that will select one operation to execute (out of 7 shown above), based on 4 - bit operations code.
4. Verify the correctness of all instructions in simulation by comparing the results with MIPS instructions in MARS.

What to submit: VHDL code printout, Waveforms showing opcode, operands, results, and compare with corresponding MIPS instructions.

CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

Part I.B

Design R Format Instructions

DATA FLOW (DATA PATH FOR R FORMAT SHIFT INSTRUCTIONS)

Operations

1. *Shift Left Logical* **SLL**

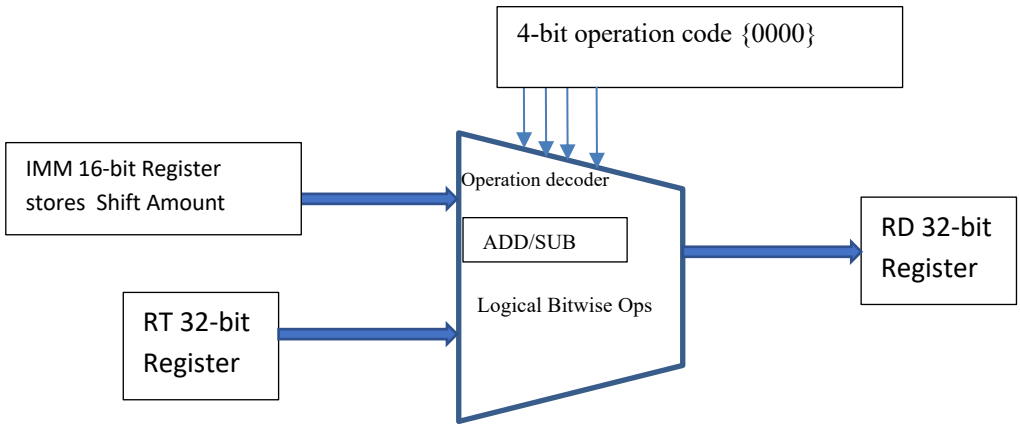
R[rd] = R[rt] << shmat(shift amount) Specified in IMM field

2. *Shift Right Logical* **SRL**

R[rd] = R[rt] >> shmat(shift amount) Specified in IMM field

3. *Shift Right Arithmetic* **SRA**

R[rd] = R[rt] >>> shmat(shift amount) Specified in IMM field



ToDO:

1. **Extend the design in part Ia to include shift operations as shown above.**
2. **Verify the correctness of all instructions in simulation by comparing the results with MIPS instructions in MARS.**

What to submit: VHDL code printout, Waveforms showing opcode, operands, results, and compare with corresponding MIPS instructions.

CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

Part II

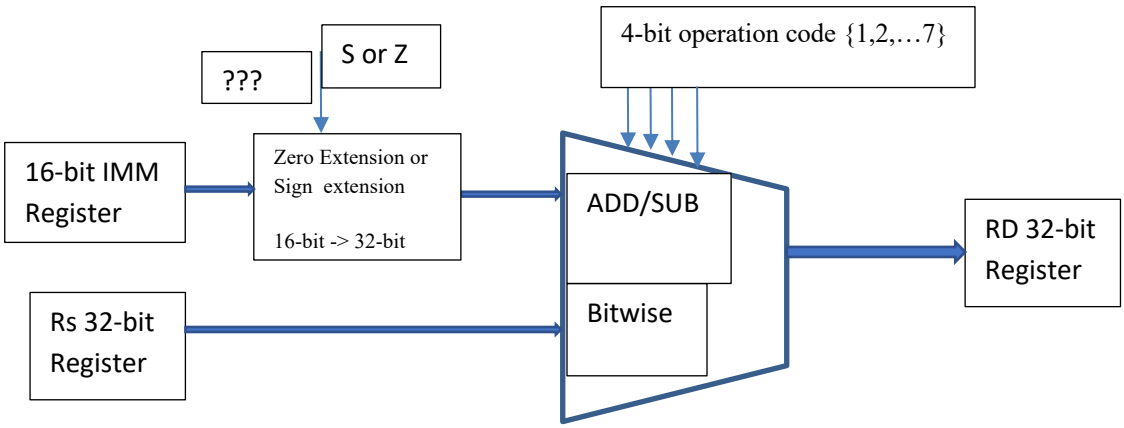
Design I Format Instructions

DATA FLOW (DATA PATH FOR I Format Arithmetic/Logic INSTRUCTIONS)

Operations

1. ***addi***
2. **addiu**
3. **subu**
4. **andi**
5. **ori**

R[rt] = R[rs] operation (SignExtImm or ZeroExtImm)



ToDO:

Extend code in Part I to include Zero Extension or Sign Extension unit as show in the figure. The 16 to 32 bit extension is controlled by signal if (S=1) the do sign extension else do zero extension.

What to Submit:

Same as in Part I for Format I instructions.

Suggested operands:

N=32 bits using Most positive, Most negative integer as a first operand, and integers +1, -1, +2, -2 as a second operand. You have to demonstrate that flags OVERFLOW, ZERO, NEGATIVE are set correctly.

CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

Part III

DATA FLOW (DATA PATH FOR MEMORY ACCESS INSTRUCTIONS FORMAT I)

Operations

Load Word	*lw*	*I*	R[rt] = M[R[rs]+SignExtImm]

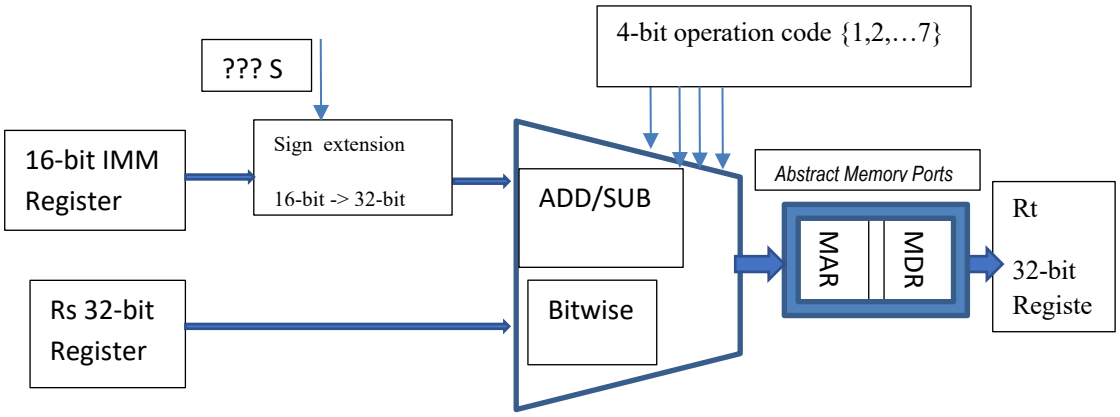
MAR -Memory Address Register

MDR -Memory Data (for the purpose of lab demo initialize MDR to any 32-bit data)

1. Memory Address computation

MAR = R[rs] operation *SignExtImm*

2. Copy data from MDR to Rt, R[rt] <= MDR



ToDO:

Modify Part II to compute effective address.

What to submit:

Same as in Part II PLUS Show Content of register MAR, and RT. Compare with MIPS LW instruction.

CSC 343 Spring 2021

Laboratory Exercise: DESIGN ARITHMETIC LOGIC UNIT GRADED!

Submit only VHDL code, and waveforms as described in each part. No Video, No Report

Instructor :Professor Isidor Gertner

Submission Date By April 6, 2021

Part IV

DATA FLOW (DATA PATH FOR MEMORY ACCESS INSTRUCTIONS)

Operations

Store Word	*sw*	*I*	M[R[rs]+SignExtImm] = R[rt]

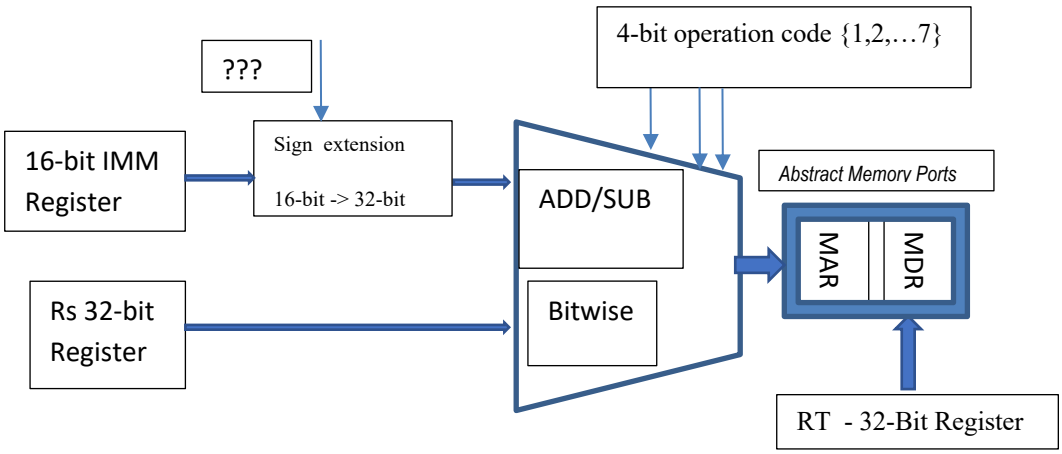
MAR -Memory Address Register

MDR -Memory Data Register

1. Memory Address computation

MAR = R[rs] operation *SignExtImm*

2. Copy data from Rt to MDR, R[rt] <= MDR



TODO:

Same as in part III.

What Submit:

ToDO:

Modify Part II to compute effective address.

What to submit:

Same as in Part III PLUS Show Content of register MAR,MDR and RT. Compare with MIPS SW instruction.