Tutorial 6
CSE 112 Computer Organization

ISA description:

First Block Instructions	Description	Operation
addi rd,rs, imm[11:0]	Add immediate	rd = rs + sext(imm[11:0])
sub rd, rs1, rs2	Subtract registers	rd = rs1 - rs2
add rd, rs1, rs2	Add registers	rd = rs1 + rs2
xor rd, rs1, rs2	Perform XOR boolean operation	rd = rs1 xor rs2
slt rd, rs1, rs2	Set Less than	rd = 1. If signed(rs1) < signed(rs2) else rd=0
sltu rd,rs1,rs2	Set Less than unsigned	rd = 1. If unsigned(rs1) < unsigned(rs2) else rd=0
lw rd,#imm[11:0](rs1)	Load Word	rd=mem(sext(rs1+imm))
sw rd,#imm[11:0](rs1)	Store Word	mem(sext(rs1+imm))=rd
beq rs1, rs2, imm[12:1]	Branch if equal	PC = PC + sext({imm[12:1],1'b0}) if rs1 == rs2

Second Block Instructions	Description	Operation
bne rs1, rs2, imm[12:1]	Branch if not equal	PC = PC + sext({imm[12:1],1'b0}) if rs1 =! rs2
bltu rs1,rs2,imm[12:1]	Branch if less than unsigned	PC = PC + sext({imm[12:1],1'b0}) If unsigned(rs1) < unsigned(rs2)
blt rs1,rs2,imm[12:1]	Branch if Less than	PC = PC + sext[imm[12:1],1'b0}] If signed(rs1) < signed(rs2)
bge rs1,rs2, imm[12:1]	Branch if Greater than or Equal	PC= PC + sext({imm[12:1],1'b0}) If signed(rs1) > signed(rs2)
bgeu rs1,rs2, imm[12:1]	Branch if Greater than or equal unsigned	PC= PC + sext({imm[12:1],1'b0}) If unsigned(rs1) > unsigned(rs2)

Note: The ISA is a subset of RISC-V ISA. Each instruction is of 32-bit size, we have 32 registers as standard from x0 to x31 with x0 hardwired to zero. The **symbol #** indicates an immediate value in the decimal format.

Apart from the above instructions, the assembler and the operating system support the following subroutines.

Name	Semantics	Syntax
Input	Reads immediate data from the user into reg	in reg
Output	Prints str on the console	out "str"
Halt	Halts the program execution	hlt

Q1: Nested Loops

Write an assembly program to print the following pattern for **m** lines, where **m** is read as input from the user. Note that performing **out "\n"** moves the cursor to the next line. Assume that **m** will always be greater than or equal to 1.

#
##
###
####
#####

(Hint: The above pattern would be printed when m = 5)

Q2: Switch Case Without Break

Write an assembly program that takes an input **n** from the user and prints
n n-1 n-2 n-3 … 0
If **n** is **0**, then only **0** is printed.
Assume that **n** will always be one of **0, 1, 2, 3, 4, or 5**.

Example 1: If n=3, print
3 2 1 0

Example 2: If n = 1, print
1 0

Use switch-case like structures in your solution.

Q3: Switch Case With Break

Write an assembly program that takes an input **n** from the user and prints back the number. Valid values of **n** are **0, 1, 2, 3, 4, or 5**. If any other value is given to **n,** print "**Invalid**"**.**

Example 1: If n=3, print
3

Example 2: If n = 1, print
1

Example 3: If n = 10, print
Invalid

Use switch-case like structures in your solution.

Q4. Write an assembly that performs the factorial of an integer 'n' given by a user.