ECS 409 : MIPS Assembly Programming
Assignment 1: Sequential Construct-I

Integer Arithmetic and Logical Instructions

Instructor: Dr. Sukarn Agarwal,
Assistant Professor,
EECS,
IISER Bhopal

October 12, 2025

Sequential Construct

The programming requires a dividing a task, into small unit of work. These unit of work are represented with programming construct that represents part of task. In the sequential construct, the designated task is broken into smaller task one follow by another.

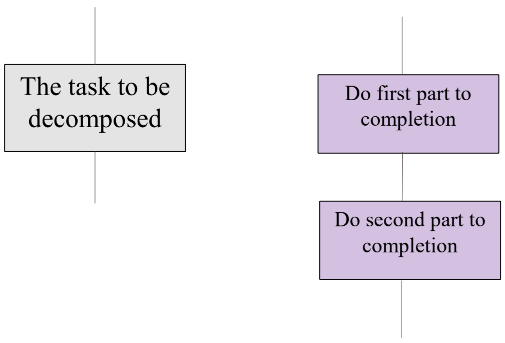
The task to be decomposed
Do first part to completion
Do second part to completion

Figure 1: Representational view of Sequential Construct

Arithmetic Instructions

Following are the set of arithmetic instruction that is to be used in this assignment.
In all the list of instruction, \$1, \$2 and \$3 represent the registers for the understanding purposes. In the assignment, you have to use the register name not the corresponding register number. Note that the details and list of the register is already provided in the instruction manual.

Instruction	Example	Meaning	Comments
add	add $1, $2, $3	$1=$2+$3	
subtract	sub $1, $2, $3	$1=$2-$3	
add immediate	addi $1,$2,100	$1=$2+100	”immediate” means a constant number
add unsigned	addu $1,$2,$3	$1=$2+$3	Values are treated as unsigned integers, not two’s complement integers
subtract unsigned	subu $1,$2,$3	$1=$2-$3	Values are treated as unsigned integers, not two’s complement integers
add immediate unsigned	addiu $1,$2,100	$1=$2+100	Values are treated as unsigned integers, not two’s complement integers
Multiply (without overflow)	mul $1,$2,$3	$1=$2*$3	Result is only 32 bit
Multiply	mult $2,$3	$hi,$low=2∗3	Upper 32 bits stored in special register hi Lower 32 bits stored in special register lo
Divide	div $2,$3	$hi,$low=$2/$3	Remainder stored in special register hi Quotient stored in special register lo
Unsigned Divide	divu $2,$3	$hi,$low=$2/$3	$2 and $3 store unsigned values. Remainder stored in special register hi Quotient stored in special register lo

Table 1: Arithmetic Instruction with their details and explanations

Logical Instructions

Following are the set of logical instructions that is to be used in this assignment.

Instruction	Example	Meaning	Comments
and	and $1, $2, $3	$1=$2&$3	Bitwise AND
or	or $1, $2, $3	$1=$2—$3	Bitwise OR
and immediate	andi $1,$2,100	$1=$2&100	Bitwise AND with immediate value
or immediate	ori $1,$2,100	$1=$2—100	Bitwise OR with immediate value
nor	nor $1,$2,$3	$1=$2↓$3	Bitwise NOR
shift left logical	sll $1,$2,10	$1=$2<<10	Shift left by constant number of bits
shift right logical	srl $1,$2,10	$1=$2>>100	Shift right by constant number of bits

Table 2: Logical Instructions with their details and explanations

Data Movement Instructions

Following are the set of data movement instructions that is to be used in this assignment.

Instruction	Example	Meaning	Comments
load word	lw $1, 100($2)	$1=Memory[$2+100]	Copy from memory to register
store word	sw $1, 100($2)	Memory[$2+100]=$1	Copy from register to memory
load upper immediate	lui $1, 100	$1=100x2ˆ16	Load constant into upper 16bits. Lower 16bits are set to zero.
load address	la $1, label	$1=Address of label	Pseudo-instruction (provided by the assembler, not processor!) Load computed address of label (not its contents) into register
load immediate	li $1, 100	$1=100	Loads immediate value into register
move from hi	mfhi $2	$2=hi	Copy from special purpose register hi to general register
move from lo	mflo $2	$2=lo	Copy from special purpose register lo to general register
move	move $1,$2	$1=$2	Copy from register to register

Table 3: List of Data Movement Instruction with their details and explanations

Note: Variation on load and store also exist for smaller data sizes.

1. 16-bit halfword: lh and sh
2. 8-bit byte: lb and sb

System Calls

The SPIM provide a large number of system call. These are the call to Operating System and do not represent MIPS process instruction. These call are either implemented by the OS or standard library.

System calls are used for input, output and to exit the program. These calls are commences with the help of *syscall* function. To use the instruction, the appropriate arguments in registers $v0, $a0-$a1, or $f12 are supplied depending on the specific call required. Following are the list of system calls that are to be required in this assignment.

Service	Operation	Code (in $v0)	Arguements
exit	stop program from running	10	none
exit2	stop program from running and return an integer	17	$a0=result (integer number)

Table 4: List of System Calls with their usage and explanations

Problem 1: For a given arithmetic and geometric progression sequence:

```
1,11,21,31,........ (A.P.)
4, 8, 16,........(G.P.)
```

Write a MIPS assembly program (using the list of instructions given in tables-1,3 and 4) to compute the eight-term (in A.P) / fourth term (in G.P.) and sum of the first six (in A.P) / first four numbers (in G.P.). Note that values of the first term (a), the common difference/ratio (d/r), and the number of terms (n) is stored in the registers or memory locations (refer assignment-1 assembler directive section). Further, all the digits used in the formulae are immediately provided in the instruction as an immediate value.

Problem 2: Write a assembly program to implement a half adder and subtractor (using the list of the instructions given in tables 2,3 and 4).

Problem 3: Write a assembly program to swap two register values using logical instructions given in tables 2 and 4. The values are loaded into the register using the instructions given in table 3.

Problem 4: Write a assembly program to compute a weighted average of four 32-bit numbers (A,B,C and D) stored in the registers $\hat{0}$, \$t1, \$t2 and \$t3 and the returned the value into \$a0.

```
(.125A + .25B + .5C+ .5D)
```

Note (for problem-4): Do not use mult/div and floating point register.

Note: Usage of other instructions (other than the instructions given in tables 1,2,3 and 4) to solve the problems results in zero marks. Submit all of your source code and a final screenshot of the register panel (integer) to the google classroom portal (in a pdf format, make sure that your name appear on the screen) on the end of the day of 19th October 2025 (Indian Standard Time). Further, any copy case between the assignment results in zero marks.