ECS 409 : MIPS Assembly Programming
Iterative Constructs

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

November 2, 2025

Iterative 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. The iterative construct is used if the designated task consists of doing a sub-task a number of times, but only as long as some condition is true.

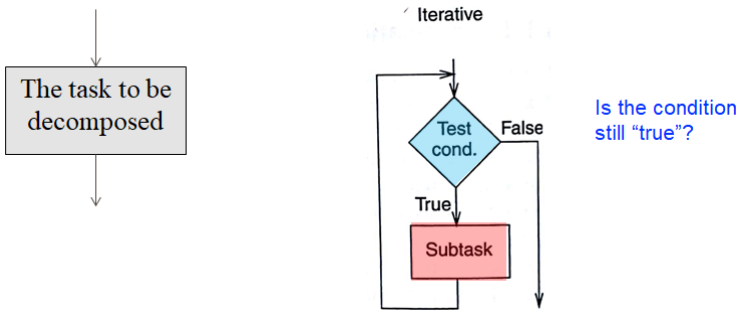

Figure 1: Representational view of Iterative Construct

Iterative Jump Instructions

MIPS provides four iterative jump instructions. All iterative instructions unconditionally jump into a specific address determined within the instruction. It is much easier to use a label for the jump instructions instead of an absolute number. For example, `j loop`, the label `loop` here should be defined somewhere else in the code.

Instruction	Example	Meaning	Comments
jump	`j 2000`	Go to address 2000	Jump to target address
jump register	`jr $2`	Go to address stored in $2	For switch, procedure return
jump and link	`jal 2000`	$ra=PC+4; go to address 2000	Use when making procedure call. This save the return address in $ra.
jump and link register	`jalr $2`	$ra=PC+4; go to address stored in $2	Use when making procedure call and return

Table 1: List of Iterative Jump Instructions with their details and explanations

Problem 1: The Fibonacci Sequence is the series of numbers:

```
0,1,1,2,3,5,8,13,21.........
```

that is produced by the adding up two number appear before the sequence. Write a MIPS assembly program that takes one number as an input, let say n and print the first n number of Fibonacci series by first storing into the memory. The output format (for $n = 6$) should be like this:

```
1 number of fibonacci series is stored at memory location 3000
2 number of fibonacci series is stored at memory location 3004
3 number of fibonacci series is stored at memory location 3008
.....
fibonacci series: 0,1,1,2,3,5
```

Problem 2: Write an MIPS assembly program that lets the use to input an integer, then output a countdown from that number to 1 (inclusively), line by line. For example if user entered 4, a correct program would output the following sequence:

```
4
3
2
1
```

Problem 3: Write a complete MIPS program that implements the algorithm shown below (in C):

```
void main()
{
int J=1;
int a;

a=0;
while(1){
if(a>20)
break;
J++;
a+=2;
}

do {
J++;
} while (J<100);

while(a > 0) {
J--;
a--;
}
```

Problem 4: Write an MIPS assembly program that let the user to input as a integer number, let say k and prompt the following output in the screen if the value of $k = 4$

```
   1
  121
 12321
1234321
```

Note: Usage of other instructions (other than the instructions given in the table 1 and the instructions given in assignment-1, 2, 3 and 4) to solve the problem results into the zero marks. Submit all of your source code and final screen shot of the register panels or console screen (both integer and floating point) to the google classroom portal by the end of the day of 9th Nov 2025 (Indian Standard Time). Further any copy case between the assignments results into the zero marks.