Instruction Set proposal

Design of ISA:

- Architecture model :
  - von Nauman.
  - that uses the same memory for both data and instructions.

- Class:
  - RISC .
  - Fixed length instructions.

- Our ISA :
  - General-purpose register architecture.
  - the operands are either registers or memory locations.

- 16 general purpose registers
- ISA can access memory only with load and store instructions

General purpose registers:

0)$zero → constant that has the value zero
1)$one → constant that has the value one

2,3) $v0-v1→values for the functions return values

4)$a0-$a1→ arguments for the procedure

5)$ra→return address

6)$sp→stack pointer

7)$fp→frame pointer

8)$gp→global pointer

9,10,11,12)$t0-$t3→temporary variables

13,14,15)$s0-$s2→saved temporaries

Addressing modes and memory access:

- We use byte addressing to access memory operands
- Addressing modes : register , Immediate , O-type , J-type

General categories :

- Arithmetic and computational.
- logical operations
- branch and jump operations
- memory operations

R-type format :

Opcode(5bits)	$r3(5bits)	$r2(5bits)	$r1(5bits)	Shift(4bit)

1) ADD: r1←r2,r3

2) SUB:r1←r2,r3

3)AND:r1 ← r2,r3

4) OR:

5) XOR

6) MOD: a modulus operation (get remainder of division)

7) IDV : Integer Division.

8) MUL

9)MULA: multiply Accumulative:r1←(a*b)+r1

10) nCk: $C(n,r)=n!/(r!(n-r)!)$

11) SUBR : subtract after reversing operands. If x-y is typed it computes y-x

12) GCD: greatest common divisor of two numbers

13) LCM: lowest common multiple of two numbers

14) POW: execute the mathematical operation of power

15) SLT:set if less than

I-type format:

Opcode(5bits)	$r3(5bits)	$r2(5bits)	Immidiat(9bit)

1) ADDI: add immediate
2) SLTI :set less than immediate
3) BEQ: branch if equal
4) LDI: load immediate value.
5) NZERO: count the number of zeros in a register
6) NONE: count the number of ones in a register
7) LD: load from memory
8) SW: store to memory
9) MOV: move the value of register to another register

O-type format:

Opcode(5bits)	$r3(5bits)	0(5bits)	0(5bits)	Immidiat(9bit)

1)SLL :Shift left logical

2)SRL: shift right logical

3)SRA :shift right arithmetic

4)NOT : bitwise not

5)INC :Increment register value by one

6)DEC: decrement register value by one

7)MOVI : move immediate value

J-type format:

Opcode(5bits)	Word addressing

1)jump : J target