

## ISA Design:

- 7-bit instructions with 1 parity bit
- 4 general-purpose registers R0 (00), R1 (01), R2 (10), and R3 (11)
- 1 special register R5 (for saving current PC)
- 16-bit data (Registers and memory)
- **RED** = RA, **BLUE** = RB, **GREEN** = imm

PC	Functionality	Instruction	Coding	Example	
PC ++	RA = RA + RB RA: [0,1]	add <b>RA</b> , <b>RB</b>	0000 <b>A</b> <b>BB</b>	add <b>R1</b> , <b>R3</b>	0000 <b>111</b>
PC ++	RA = RA + imm RA: [2,3] imm: [0,3]	addi <b>RA</b> , <b>imm</b>	0001 <b>A</b> <b>ii</b>	addi <b>R2</b> , <b>2</b>	0001 <b>011</b>
PC ++	RA = RA – RB RA: [0,1]	sub <b>RA</b> , <b>RB</b>	0010 <b>A</b> <b>BB</b>	sub <b>R1</b> , <b>R2</b>	0010 <b>110</b>
PC ++	RA = mem[RB] RA: [2,3]	load <b>RA</b> , ( <b>RB</b> )	0011 <b>A</b> <b>BB</b>	load <b>R3</b> , ( <b>R0</b> )	0011 <b>100</b>
PC ++	RA = RA or RB RA: [0,1]	or <b>RA</b> , <b>RB</b>	0100 <b>A</b> <b>BB</b>	or <b>R0</b> , <b>R1</b>	0100 <b>001</b>
PC ++	R0 = 1 if RA < RB R0 = 0 if RA > RB RA: [2,3]	slt <b>RA</b> , <b>RB</b>	0101 <b>A</b> <b>BB</b>	slt <b>R3</b> , <b>R2</b>	0101 <b>110</b>
PC ++	mem[RB] = RA RA: [0,1]	store <b>RA</b> , ( <b>RB</b> )	0110 <b>A</b> <b>BB</b>	store <b>R1</b> , <b>R3</b>	0110 <b>111</b>
PC ++	RA shift left imm bits RA: [2,3] imm: [0,3]	sll <b>RA</b> , <b>imm</b>	0111 <b>A</b> <b>ii</b>	sll <b>R2</b> , <b>1</b>	0111 <b>001</b>
If RA==0, PC = PC + imm Else, PC++	RA: [0,1] imm: [0,3]	beq <b>RA</b> , <b>imm</b>	1000 <b>A</b> <b>ii</b>	beq <b>R0</b> , <b>0</b>	1000 <b>000</b>
If RA!=0, PC = PC + imm Else, PC++	RA: [2,3] imm: [0,3]	bne <b>RA</b> , <b>imm</b>	1001 <b>A</b> <b>ii</b>	bne <b>R2</b> , <b>1</b>	1001 <b>001</b>
PC ++	RA = RA and imm RA: [0,1] imm: [0,3]	andi <b>RA</b> , <b>imm</b>	1010 <b>A</b> <b>ii</b>	andi <b>R1</b> , <b>1</b>	1010 <b>101</b>
PC ++	RA shift right imm bits RA: [2,3] imm: [0,3]	srl <b>RA</b> , <b>imm</b>	1011 <b>A</b> <b>ii</b>	srl <b>R2</b> , <b>1</b>	1011 <b>001</b>
PC ++	RA = RA xor RB RA: [0,1]	xor <b>RA</b> , <b>RB</b>	1100 <b>A</b> <b>BB</b>	xor <b>R0</b> , <b>R1</b>	1100 <b>001</b>

PC = PC + imm	imm: [0,7]	j imm	1101 iii	j 6	1101110
PC = R5 (Replaces PC with R5)	RA = 5	jr R5	1110 101	jr R5	1110101
Save PC+1 to R5 and jump PC = PC + imm	imm: [0,7]	jal imm	1111 iii	jal 7	1111111