Instruction Set

RISC 16-bit, 16-instruction 16-Register Design

The Processor features five instruction classes:

1. Arithmetic (Two's Complement) ALU operation (2)

Operation	Opcode	Destination	Source	Target	Description
ADD	0000	Rd	Rs	Rt	**ADD**: Rd = Rs + Rt Operands A and B stored in register locations Rs and Rt are added and written to the destination register specified by Rd.
SUB	0001	Rd	Rs	Rt	**SUB**: Rd = Rs - Rt Operand B (Rt) is subtracted from Operand A (Rs) and written to Rd.

2. Logical ALU operation (6)

Operation	Opcode	Destination	Source	Target	Description
AND	0011	Rd	Rs	Rt	**AND**: Rd = Rs & Rt Operand A (Rs) is bitwise anded with Operand B (Rt) and written into Rd.
OR	0011	Rd	Rs	Rt	**OR**: Rd = Rs | Rt Operand A (Rs) is bitwise ored with Operand B (Rt) and written into Rd.
XOR	0100	Rd	Rs	Rt	**XOR**: Rd = Rs ^ Rt Operand A (Rs) is bitwise Xored with Operand B (Rt) and written into Rd.
NOT	0101	Rd	Rs		**NOT**: Rd = ~Rs Operand A (Rs) is bitwise inverted and written into Rd.
SLA	0110	Rd	Rs		**SLA**: Rd = Rs << 1 Operand A (Rs) is arithmetically shifted to the left by one bit and written into Rd.
SRA	0111	Rd	Rs		**SRA**: Rd = Rs >> 1 Operand A (Rs) is arithmetically shifted to the right by one bit and written into Rd. The MSB (sign bit) will be preserved for this operation.

3. Memory operations (3)

Operation	Opcode	Destination	Source	Target	Description
LI	1000	Rd	Imm		**LI:** Rd = 8-bit Sign extended Immediate The 8-bit immediate in the Instruction word is sign-extended to 16-bits and written into the register specified by Rd.
LW	1001	Rd	Dir		**LW:** Rd = Mem[Dir] The memory word specified by the address Dir is loaded into register Rd.
SW	1010	Rt	Dir		**SW:** Mem[Dir] = Rt The data in register Rt is stored into the memory location Dir.

4. Conditional Branch operations (2)

Operation	Opcode	Destination	Source	Target	Description
BIZ	1011	Rs	Offset		**BIZ:** PC = PC + 1 + Offset **if** Rs = 0 If all the bits in register Rs are zero then the current Program Count (PC + 1) is offset to PC + 1 + Offset. The count is offset from PC + 1 because it is incremented and stored during the Fetch cycle.
BNZ	1100	Rs	Offset		**BNZ:** PC = PC + 1 + Offset **if** Rs! = 0 If all the bits in register Rs are not zero than the current Program Count (PC + 1) is offset to PC + 1 + Offset.

5. Program Count Jump operations (3)

Operation	Opcode	Destination	Source	Target	Description
JAL	1101	Rd	Offset		**JAL**: Rd = PC + 1 and PC = PC + 1 + Offset Jump and Link instruction would write current Program Count in register Rd and offset the program count to PC + 1 + Offset
JMP	1110		Offset		**JMP**: PC = PC + 1 + Offset Unconditional jump instruction will offset the program count to PC + 1 + Offset.
JR	1111		Rs		**JR**: PC = Rs Jump Return instruction will set the Program Count to the one previously stored in JAL.