

AoE Instruction Formats

19:16	15:12	11:8	7:4	3:0	
-------	-------	------	-----	-----	--

Arithmetic/Logical Instructions

not used	rs2	rs1	rd	opcode	Instruction Meaning
XXXX	4 bits	4 bits	4 bits	0100	add
XXXX	4 bits	4 bits	4 bits	0101	subtract
XXXX	4 bits	4 bits	4 bits	0110	AND
XXXX	4 bits	4 bits	4 bits	0111	OR

Load Instruction

immediate[7:4]	immediate[3:0]	rs1	rd	opcode	Instruction Meaning
4 bits	4 bits	4 bits	4 bits	0010	ld

Store Instruction

immediate[7:4]	rs2	rs1	immediate[3:0]	opcode	Instruction Meaning
4 bits	4 bits	4 bits	4 bits	0011	sd

Conditional Branch Instruction

immediate[7:4]	rs2	rs1	immediate[3:0]	opcode	Instruction Meaning
4 bits	4 bits	4 bits	4 bits	0001	beq

No Operation Instruction

not used	not used	not used	not used	opcode	Instruction Meaning
XXXX	XXXX	XXXX	XXXX	0000	nop

Examples

Machine Code	Assembly Code
0000 0110 0101 0111 0100	add x7, x5, x6
0000 0100 0101 0111 0010	ld x7, 4(x5)
0000 0110 0101 0100 0011	sd x6, 4(x5)
0000 0110 0101 0011 0001	beq x5, x6, 3
0000 0111 1010 1101 0000	nop

Important Notes on Interpreting Assembly Code

- “x7” indicates register 7
- “add x7, x5, x6” means add the **contents** of registers 5 and 6, and put the result in register 7
- “4(x5)” indicates a data memory address that is the sum of the **constant** 4 and the **contents** of register 5
- “ld x7, 4(x5)” means that we should go to the data memory location at the address 4(x5), and load the contents of that memory location into register 7
- Similarly, “sd x6, 4(x5)” means that we should store the contents of register 6 in the data memory location at address 4(x5)
- By default, after executing one instruction, the computer will go on to the next one - the one 1 address later in the instruction memory. “beq x5, x6, 3” means that if the contents of x5 and x6 are equal, we should instead branch to the instruction that is 3 addresses later than the current instruction

Important Notes on Interpreting Machine Code

- opcode tells the computer what type of instruction we are executing
- rs1 and rs2 are “source” registers, while rd is a “destination” register
- immediate is a constant value. For example, the 4 in the assembly code “4(x5)” would be stored in the immediate field
- immediate is 8 bits wide, so it is contained in two nibbles in each instruction that uses an immediate field
- Remember that immediate can be negative (if you are doing subtraction), but register numbers are always positive, so an rs1 field of 1000 represents register 8, not register -8