

Shift Instructions

Shift amount is in (lower 5 bits of) a register

- sll: shift left logical
 - **Example:** sll t0, t1, t2 # t0 = t1 << t2
- srl: shift right logical
 - **Example:** srl t0, t1, t2 # t0 = t1 >> t2
- sra: shift right arithmetic
 - **Example:** sra t0, t1, t2 # t0 = t1 >>> t2

Shift instructions are used for logical shifts, sign extension and multiplication/division by powers of 2

Syntax

- # 1. First operand is the destination register
- # 2. Second operand is the source register
- # 3. Third operand contains the shift amount

Loading registers

```
li t1, 5      # 0101
li t2, 1      # Shift amount
```

1.) Shift Left Logical

```
sll t0, t1, t2 # Shift 5 left by 1 bit resulting in 10
```

2.) Shift Right Logical

```
srl t0, t1, t2 # Shift 5 right by 1 bit resulting in 2
```

3.) Shift Right Arithmetic

```
sra t0, t1, t2 # Shift right while preserving sign bit resulting in 2 as sign bit is 0
```

Immediate Shift Instructions

Shift amount is an immediate between 0 to 31

- slli: shift left logical immediate
 - **Example:** slli t0, t1, 23 # t0 = t1 << 23
- srli: shift right logical immediate
 - **Example:** srli t0, t1, 18 # t0 = t1 >> 18
- srai: shift right arithmetic immediate
 - **Example:** srai t0, t1, 5 # t0 = t1 >>> 5

Immediate shift takes the value to shift by instead of it being stored in another register

4.) Shift Left Logical Immediate

slli t0 , t1 , 23 # Shift 0101 to the left by 23 bits

5.) Shift Right Logical Immediate

srli t0 , t1 , 10 # Shift 0101 to the right by 18 bits

6.) Shift Right Arithmetic Immediate

srai t0 , t1 , 5 # Shift 0101 to the right by 5 bits while preserving sign bit