

arithmetic and logical

Arithmetic & Logical Instructions

Instructions(Group1)	Brief
ADDI/ADD	Addition
SUB	Subtraction
SLT/SLTI/SLTU/SLTIU	Comparison

Instructions(Group2)	Brief
SLL/SLLI	Shift left logical
SLA/SLAI	Shift left arithmetic
SRA/SRAI	Shift right arithmetic
AND/ANDI	Bitwise AND
OR/ORI	Bitwise OR
XOR/XORI	Bitwise exclusive or

shift and bitwise

Shift Left Logical (sll)

t0: 0x00000724



t1: 15

t0: 0x03920000



Sll t0, t0, t1

Bits beyond boundary are lost, NOT wrapped around

Shift Left Logical Immediate(slli)

0x00000724

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0

23 ←

1 1 1 0 0 1 0 0 1 0

0x92000000

Slli t0, t0, 23

**Bits beyond boundary are lost, NOT wrapped around
<imm5>**

Shift Right Logical (srl)

t0: 0x00000724



t0: 0x00000072



Always Zero Filled

Srl t0, t0, t1

Bits beyond boundary are lost, NOT wrapped around

FISTT

Shift Right Logical Immediate (srli)

t0: 0xC0000724



t0: 0x0C000072



Always Zero Filled

Srli t0, t0, 4
<imm5>

Shift Right Arithmetic (sra)

Sign



t0: 0x0000724

Sign extension



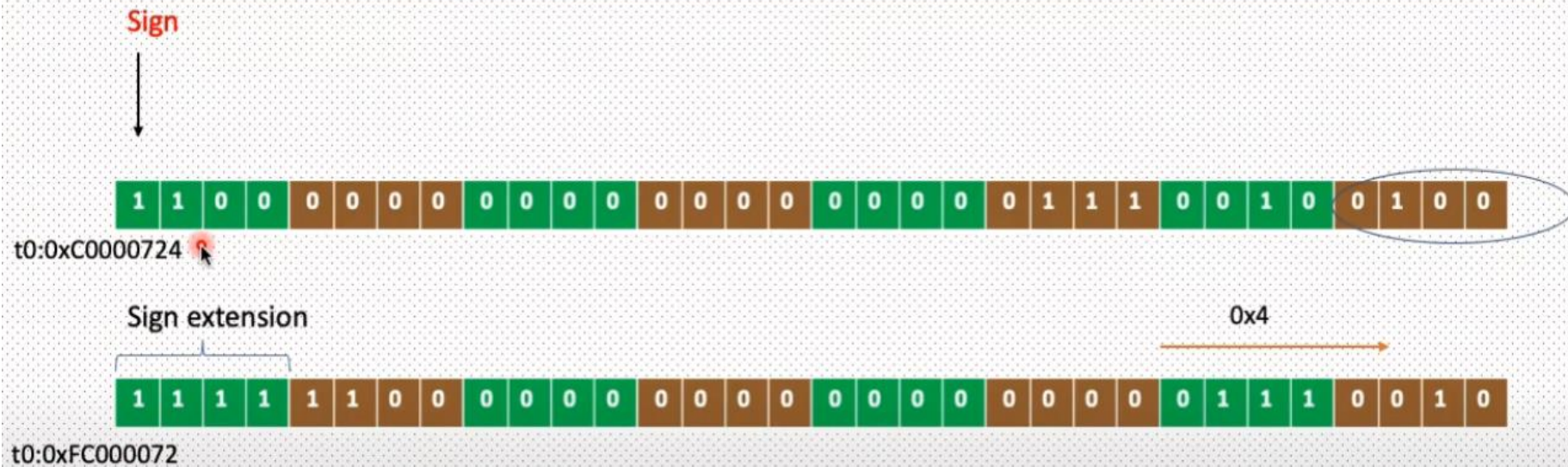
t0: 0x00000072

t1: 0x4



Sra t0, t0, t1

Shift Right Arithmetic Immediate(srai)

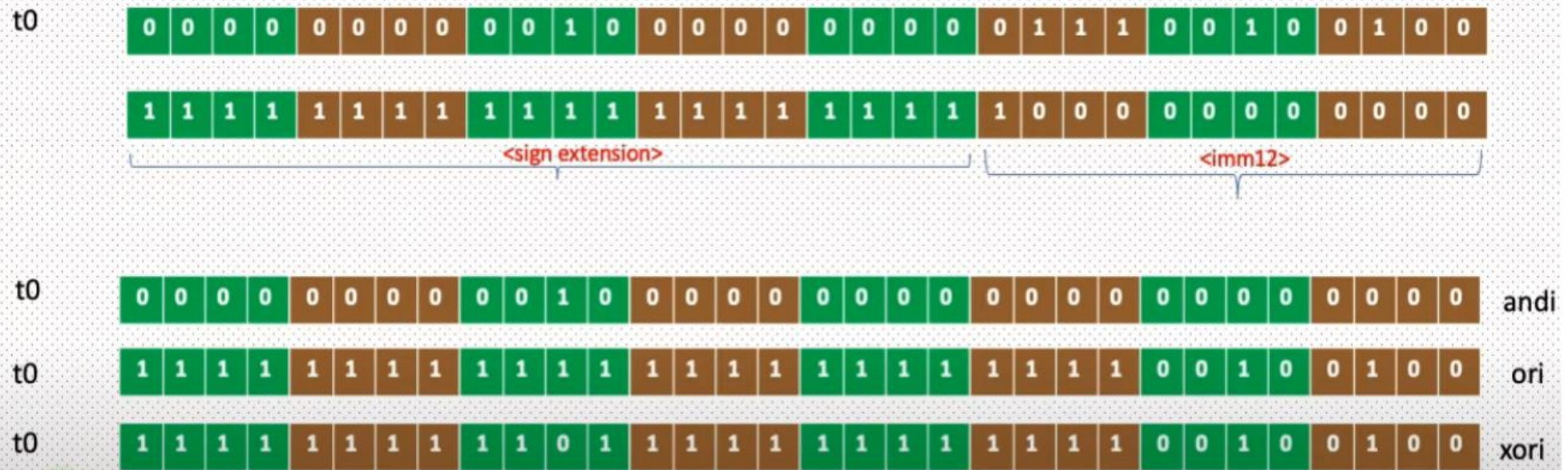


Srai t0, t0, 4

Imm value width 5 bits

SLA missing ??

Bitwise operations (andi, ori, xori)



jump

- jal rd, offset

```
loop:  addi x5, x4, 1      #  $x5 \leftarrow x4 + 1$   
jal x1, loop             # Goto loop  $x1 \leftarrow address[loop]$ 
```


- jalr rd, offset

```
addi x1, x0, 3          #  $x1 \leftarrow x0 + 3$   
loop: addi x5, x0, 1     #  $x5 \leftarrow x0 + 1$   
jalr x0, 0(x1)          #  $x0 \leftarrow mem[x1 + 0]$ 
```