

Directive	.bss	emit .bss section (if not present) and make current	
Directive	.comm sym,nam,sz,aln	emit common object to .bss section	
Directive	.common sym_name,sz,aln	emit common object to .bss section	
Directive	.section sect	emit section (if not present, default .text [[.text,.data,.rodata,.bss]])	
Directive	.option opt	RISC-V options	[rvc,norvc,pic,nopic,push,pop]
Directive	.macro name arg1 [, argn]	begin macro definition %argname to substitute	
Directive	.endm	end macro definition	
Directive	.file "filename"	emit filename FILE LOCAL symbol table	
Directive	.ident "string"	accepted for source compatibility	
Directive	.size symbol, symbol	accepted for source compatibility	
Directive	.type symbol, @function	accepted for source compatibility	
Pseudo	NOB	No operation	addi zero,zero,0
Pseudo	LI rd, expression	Load immediate	(several expansions)
Pseudo	LA rd, symbol	Load address	(several expansions)
Pseudo	MV rd, rs1	Copy register	addi rd, rs, 0
Pseudo	NOT rd, rs1	One's complement	xori rd, rs, -1
Pseudo	NEQ rd, rs1	Two's complement	sub rd, x0, rs
Pseudo	NEQW rd, rs1	Two's complement Word	subw rd, x0, rs
Pseudo	SEXT.W rd, rs1	Sign extend Word	addiw rd, rs, 0
Pseudo	SEQZ rd, rs1	Set if = zero	sitiu rd, rs, 1
Pseudo	SNEZ rd, rs1	Set if ≠ zero	situ rd, x0, rs
Pseudo	SLTZ rd, rs1	Set if < zero	sit rd, rs, x0
Pseudo	SQTZ rd, rs1	Set if > zero	sit rd, x0, rs
Pseudo	FMV.S frd, frs1	Single-precision move	fsgnj.s frd, frs, frs
Pseudo	FAB.S frd, frs1	Single-precision absolute value	fsgnj.s frd, frs, frs
Pseudo	FNEG.S frd, frs1	Single-precision negate	fsgjn.s frd, frs, frs
Pseudo	FMV.D frd, frs1	Double-precision move	fsgnj.d frd, frs, frs
Pseudo	FABS.D frd, frs1	Double-precision absolute value	fsgnj.d frd, frs, frs
Pseudo	FNEG.D frd, frs1	Double-precision negate	fsgjn.d frd, frs, frs
Pseudo	BEQZ rs1, offset	Branch if = zero	beq rs, x0, offset
Pseudo	BNEZ rs1, offset	Branch if ≠ zero	bne rs, x0, offset
Pseudo	BLEZ rs1, offset	Branch if ≤ zero	bge x0, rs, offset
Pseudo	BGEZ rs1, offset	Branch if ≥ zero	bge rs, x0, offset
Pseudo	BLTZ rs1, offset	Branch if < zero	blt rs, x0, offset
Pseudo	BGTZ rs1, offset	Branch if > zero	blt x0, rs, offset
Pseudo	BQT rs, rt, offset	Branch if >	blt rt, rs, offset
Pseudo	BLE rs, rt, offset	Branch if ≤	bge rt, rs, offset
Pseudo	BGTU rs, rt, offset	Branch if >, unsigned	bltu rt, rs, offset
Pseudo	BLEU rs, rt, offset	Branch if ≤, unsigned	bltu rt, rs, offset
Pseudo	J offset	Jump	jal x0, offset
Pseudo	JR offset	Jump register	jal x1, offset
Pseudo	RET	Return from subroutine	jair x0, x1, 0

Syntax: Imm [12][10:5] = Bits 12 & 10-5 of immediate (other bits in other part)