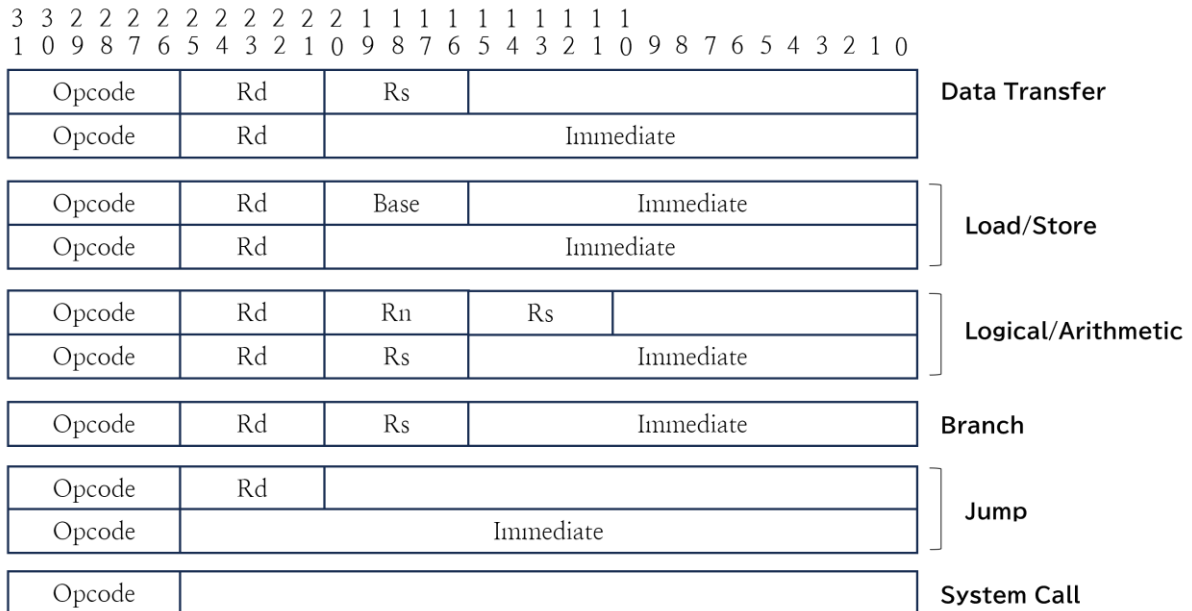


2.1 Instruction Set Summary

2.1.1 Format Summary

The AMO instruction set formats are shown below.



2.1.2 Instruction Summary

Mnemonic	Example	Instruction	DEFAULT
MOV	mov Rd, Rs/imm21	$Rd \leftarrow Rs/imm21$	
	mov Rd, imm32/symbol	literal pool (pseudo)	
LDR	ldr Rd, [base, offset]	$Rd \leftarrow mem[base + offset]$	
	ldr Rd, [relative]	$Rd \leftarrow mem[pc + relative]$	
STR	str [base, offset], Rs	$mem[base + offset] \leftarrow Rs$	
	str [relative], Rs	$mem[pc + relative] \leftarrow Rs$	
LDRB	ldrb Rd, [base, offset]	$Rd \leftarrow mem[base + offset]$	
	ldrb Rd, [relative]	$Rd \leftarrow mem[pc + relative]$	
STRB	strb [base, offset], Rs	$mem[base + offset] \leftarrow Rs$	
	strb [relative], Rs	$mem[pc + relative] \leftarrow Rs$	
LDRH	ldrh Rd, [base, offset]	$Rd \leftarrow mem[base + offset]$	
	ldrh Rd, [relative]	$Rd \leftarrow mem[pc + relative]$	
STRH	strh [base, offset], Rs	$mem[base + offset] \leftarrow Rs$	
	strh [relative], Rs	$mem[pc + relative] \leftarrow Rs$	

ADD	add Rd, Rn, Rs/imm16	$Rd \leftarrow Rn + Rs/imm16$
ADC	adc Rd, Rn, Rs/imm16	$Rd \leftarrow Rn + Rs/imm16 + \text{Carry}$
SUB	sub Rd, Rn, Rs/imm16	$Rd \leftarrow Rn - Rs/imm16$
AND	and Rd, Rn, Rs/imm16	$Rd \leftarrow Rn \text{ AND } Rs/imm16$
OR	or Rd, Rn, Rs/imm16	$Rd \leftarrow Rn \text{ OR } Rs/imm16$
XOR	xor Rd, Rn, Rs/imm16	$Rd \leftarrow Rn \text{ XOR } Rs/imm16$
NOT	not Rd, Rs/imm16	$Rd \leftarrow \text{NOT } Rs/imm16$
LSL	lsl Rd, Rn, Rs/imm16	$Rd \leftarrow Rn \ll Rs/imm16$
LSR	lsr Rd, Rn, Rs/imm16	$Rd \leftarrow Rn \gg Rs/imm16$
ASR	asr Rd, Rn, Rs/imm16	$Rd \leftarrow Rn \ggg Rs/imm16$
BEQ	beq Rd, Rs, imm16	$PC \leftarrow PC + (imm16 \ll 2) \text{ if } Rd == Rs$
BNE	bne Rd, Rs, imm16	$PC \leftarrow PC + (imm16 \ll 2) \text{ if } Rd \neq Rs$
BLT	blt Rd, Rs, imm16	$PC \leftarrow PC + (imm16 \ll 2) \text{ if } Rd < Rs$
BLE	ble Rd, Rs, imm16	$PC \leftarrow PC + (imm16 \ll 2) \text{ if } Rd \leq Rs$
BLTU	bltu Rd, Rs, imm16	$PC \leftarrow PC + (imm16 \ll 2) \text{ if } Rs < Rn$
BLEU	bleu Rd, Rs, imm16	$PC \leftarrow PC + (imm16 \ll 2) \text{ if } Rs \leq Rn$
JMP	jmp Rs/imm26	$PC = Rs/(imm26 \ll 2)$
JAL	jal Rs/imm26	$PC = (imm26 \ll 2)$ $LR = PC$
SWI	swi imm6	Jump to Interrupt Vector (Trap)
EXT	ext Rd, Rs, Opt	$Rd \leftarrow [\text{Sign/Unsign}] \text{Extend } (Rs)$
SETVT	setvt Rs, Type	Set the Vector Table Type 0: Interrupt Vector Table
RET	ret Rs	$PC = Rs/(imm26 \ll 2)$ InterruptBlocking = false
LOCK	lock	InterruptBlocking = !InterruptBlocking (for atomic operation)