

Aristotle Reference Data

Instructions

Mnemonic	Type	Name	Description	Opcode
add	S	ADD	$R[cr] = R[cr] + M[sp+\{s1, 1b'0\}]$	00010
addc	C	ADD Const	$R[cr] = R[cr] + const$	00001
addcsp	C	ADD Const Stack Pointer	$SP = SP + \{const, 00101 1b'0\}$	
and	S	AND	$R[cr] = R[cr] \& M[sp+\{s1, 1b'0\}]$	10110
andc	C	AND Const	$R[cr] = R[cr] \& const$	10101
or	S	OR	$R[cr] = R[cr] M[sp+\{s1, 1b'0\}]$	01110
orc	C	OR Constant	$R[cr] = R[cr] const$	10001
sub	S	SUBtraction	$R[cr] = R[cr] - M[sp+\{s1, 1b'0\}]$	00110
sl	C	Shift Left	$R[cr] \ll const$	01001
sr	C	Shift Right	$R[cr] \gg const$	01101
lds	S	LoaD from Stack	$R[cr] = M[sp+\{s1, 11010 1b'0\}]$	
wrs	S	WRite to Stack	$M[sp+\{s1, 1b'0\}] = R[cr]$	10010
ssp	C	Set Stack Pointer	$SP = R[cr]$	00011
lpc	S	LoaD PC	$R[cr] = PC$	00111
lsp	S	LoaD SP	$R[cr] = SP$	01011
lu	C	Load Upper bits	$R[cr] = \{const, 6b'0\}$	10100
b	C	Branch	$PC = R[cr]$	11100
beq	S	Branch Equal	if $(M[sp+\{s1, 1b'0\}] == M[sp+\{s2, 1b'0\}])$ $PC = R[cr]$	00000
bge	S	Branch Greater than or Equal	if $(M[sp+\{s1, 1b'0\}] \geq M[sp+\{s2, 1b'0\}])$ $PC = R[cr]$	00100
bgt	S	Branch Greater Than	if $(M[sp+\{s1, 1b'0\}] > M[sp+\{s2, 1b'0\}])$ $PC = R[cr]$	01000
bneq	S	Branch Not Equal	if $(M[sp+\{s1, 1b'0\}] \neq M[sp+\{s2, 1b'0\}])$ $PC = R[cr]$	01100
bpc	C	Branch relative to PC	$PC = PC + const$	11111
end	C	END of the program	Marks the end of execution	11011

Pseudo instruction	Name	Description
addow	ADD OverWrite	$R[cr] = M[sp+s1] + M[sp+s2]$
call	Call branch	$R[cr] = PC + 6$ $M[SP] = R[cr]$ $PC = PC + s1$
ld	Load label	$R[cr] = PC + s1$

Core Instruction Formats

	15	14		10	9		5	4		0
C	const								opcode	
S	X	s2			s1			opcode		

RTL Instructions

Branches	Literal	PC-Relative
Fetch & Increment	$IR \leq Mem[PC]$ $PC \leq PC + 1$	
Decode & Early Writeback	$ALU1 \leq SP + imm(IR[9:5])$ $ALU2 \leq SP + imm(IR[14:10])$	$PC \leq PC + imm(IR[15:5])$
Memory	$Mem1 \leq Mem[ALU1]$ $Mem2 \leq Mem[ALU2]$	
Writeback	if $(Mem1 == Mem2)^*$ $PC \leq CR$	

*Can be replaced with \geq , $>$, \neq , or no condition

ALU	S Type	C Type
Fetch & Increment	$IR \leq Mem[PC]$ $PC \leq PC + 1$	
Decode & Early Writeback	$ALU1 \leq SP + imm(IR[9:5])$	$CR \leq CR op imm(IR[15:5])^*$
Memory	$Mem1 \leq Mem[ALU1]$	
Writeback	$CR \leq CR op Mem1$	

*CR can be replaced with SP

Stack	lds	wrs
Fetch & Increment	$IR \leq Mem[PC]$ $PC \leq PC + 1$	
Decode & Early Writeback	$ALU1 \leq SP + imm(IR[9:5])$	
Memory	$CR \leq Mem[ALU1]$	$Mem[ALU1] \leq CR$
Writeback		

Register	lpc	lsp
Fetch & Increment	$IR \leq Mem[PC]$ $PC \leq PC + 1$	
Decode & Early Writeback	$CR \leq PC$	$CR \leq SP$
Memory		
Writeback		

Stack Frame

Caller Local		SP During Caller
Caller Setup for Callee	arg...	
	arg1	
	arg0	
	Return Address	SP During Callee
Callee Local		

