

These two tables describe the SM213 ISA. The first gives the general form for instructions in assembly and machine language and describes instruction semantics. ‘s’ and ‘d’ refer to source and destination register numbers while ‘p’ and ‘i’ to refer to compressed-offset and index values. Offsets in assembly use ‘o’, stored in machine code as ‘p’ such that ‘o’ is either 2 or 4 times ‘p’ as indicated in the semantics column. The second table gives an example of each instruction. With the exception of pc-relative branches and shift, all immediate values in instructions are unsigned.

Operation	Machine Language	Semantics / RTL	Assembly
load immediate	0d-- vvvvvvvv	$r[d] \leftarrow v$	ld \$v, rd
load base+offset	1psd	$r[d] \leftarrow m[(o = p \times 4) + r[s]]$	ld o(rs), rd
load indexed	2sid	$r[d] \leftarrow m[r[s] + r[i] \times 4]$	ld (rs, ri, 4), rd
store base+offset	3spd	$m[(o = p \times 4) + r[d]] \leftarrow r[s]$	st rs, o(rd)
store indexed	4sdi	$m[r[d] + r[i] \times 4] \leftarrow r[s]$	st rs, (rd, ri, 4)
halt	F0--	(stop execution)	halt
nop	FF--	(do nothing)	nop
register move	60sd	$r[d] \leftarrow r[s]$	mov rs, rd
add	61sd	$r[d] \leftarrow r[d] + r[s]$	add rs, rd
and	62sd	$r[d] \leftarrow r[d] \& r[s]$	and rs, rd
inc	63-d	$r[d] \leftarrow r[d] + 1$	inc rd
inc addr	64-d	$r[d] \leftarrow r[d] + 4$	inca rd
dec	65-d	$r[d] \leftarrow r[d] - 1$	dec rd
dec addr	66-d	$r[d] \leftarrow r[d] - 4$	deca rd
not	67-d	$r[d] \leftarrow \sim r[d]$	not rd
shift	7dss (ss > 0) 7dss (ss < 0)	$r[d] \leftarrow r[d] \ll (v = s)$ $r[d] \leftarrow r[d] \gg (v = -s)$	shl \$v, rd shr \$v, rd
branch	8-pp	$pc \leftarrow (a = pc + p \times 2)$	br a
branch if equal	9cpp	if $r[c] == 0 : pc \leftarrow (a = pc + p \times 2)$	beq rc, a
branch if greater	Acpp	if $r[c] > 0 : pc \leftarrow (a = pc + p \times 2)$	bgt rc, a
jump	B--- aaaaaaaaa	$pc \leftarrow a$	j a
get program counter	6Fpd	$r[d] \leftarrow pc + (o = p \times 2)$	gpc \$o, rd
jump indirect	Cdpp	$pc \leftarrow r[d] + (o = p \times 2)$	j o(rd)
system call	F1nn	system call #n	sys \$n

Operation	Machine Language Example	Assembly Language Example
load immediate	0100 00001000	ld \$0x1000, r1
load base+offset	1123	ld 4(r2), r3
load indexed	2123	ld (r1, r2, 4), r3
store base+offset	3123	st r1, 8(r3)
store indexed	4123	st r1, (r2, r3, 4)
halt	f000	halt
nop	ff00	nop
register move	6012	mov r1, r2
add	6112	add r1, r2
and	6212	and r1, r2
inc	6301	inc r1
inc addr	6401	inca r1
dec	6501	dec r1
dec addr	6601	deca r1
not	6701	not r1
shift	7102 71fe	shl \$2, r1 shr \$2, r1
branch	(at address 0x1000) 8003	br 0x1008
branch if equal	(at address 0x1000) 9103	beq r1, 0x1008
branch if greater	(at address 0x1000) a103	bgt r1, 0x1008
jump	b000 00001000	j 0x1000
get program counter	6f31	gpc \$6, r1
jump indirect	c104	j 8(r1)
syscall read(r0=0, r1=buffer, r2=size)	f100	sys \$0
syscall write(r0=1, r1=buffer, r2=size)	f101	sys \$1
syscall exec(r0=buffer, r1=size)	f102	sys \$2