INSTR BITS	mnemoni	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DOUBLE OPS		sz	op code			mode			reg			mode			reg			
move	mov	b	1			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
add	add	0	6			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
subtract	sub	1	6			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
compare	cmp	b	2			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
bit set	bis	b	5			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
bit clear	bic	b	4			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
bit test	bit	b	3			mode 0 to 7			reg 0 to 7			mode 0 to 7			reg 0 to 7			
JUMPS/BRANCHES		sz	constant				typ		flg	offset								
jump	jmp	0	0				0		1			mode 0 to 7			reg 0 to 7			
jump to subroutine	jsr	0	1				0		reg 0 to 7			mode 0 to 7			reg 0 to 7			exception in SINGLE OPS, to be treated as SINGLE OP word inst
return from subrout	rts	0	0				0		2			0	0	0	reg 0 to 7			exception in FLAG OPS, the '2' overlaps with FLAG OPS
swap byte	swab	0	0				0		3			mode 0 to 7			reg 0 to 7			
branch	br	0	0				0		1	offset								
if equal	beq	0	0				1		1	offset								
if not eq	bne	0	0				1		0	offset								
if -ve	bmi	1	0				0		1	offset								For bracnhes, first check [15] bit then continue processing
if +ve	bpl	1	0				0		0	offset								(helps in processing typ = 0 instr)
if cy	bcs	1	0				3		1	offset								
if no cy	bcc	1	0				3		0	offset								
if oflw	bvs	1	0				2		1	offset								
if no oflw	bvc	1	0				2		0	offset								
less than	blt	0	0				2		1	offset								
grt eq	bge	0	0				2		0	offset								
less eq	ble	0	0				3		1	offset								
grt than	bgt	0	0				3		0	offset								
higher	bhi	1	0				1		0	offset								
low some	blos	1	0				1		1	offset								
high some	bhis	1	0				3		0	offset								
lower	blo	1	0				3		1	offset								
SINGLE OPS		sz	constant				typ		op code			mode[2:0]			reg[2:0]			
clear	clr	b	1				1		0			mode 0 to 7			reg 0 to 7			
incre	inc	b	1				1		2			mode 0 to 7			reg 0 to 7			
decr	dec	b	1				1		3			mode 0 to 7			reg 0 to 7			
negate	neg	b	1				1		4			mode 0 to 7			reg 0 to 7			
test	tst	b	1				1		7			mode 0 to 7			reg 0 to 7			
1s comp	com	b	1				1		1			mode 0 to 7			reg 0 to 7			
add cy	adc	b	1				1		5			mode 0 to 7			reg 0 to 7			
sub cy	sbc	b	1				1		6			mode 0 to 7			reg 0 to 7			
rot rt	ror	b	1				2		0			mode 0 to 7			reg 0 to 7			
rot lft	rol	b	1				2		1 [1]			mode 0 to 7			reg 0 to 7			
arith rt	asr	b	1				2		2			mode 0 to 7			reg 0 to 7			
arith lft	asl	b	1				2		3			mode 0 to 7			reg 0 to 7			
Program Status OPS		constant										flg	S/C	N	Z	V	C	
clr cy	clc	0	0				0		2			1	0	0	0	0	1	
clr oflw	clv	0	0				0		2			1	0	0	0	1	0	
clr zero	clz	0	0				0		2			1	0	0	1	0	0	
clr neg	cln	0	0				0		2			1	0	1	0	0	0	
set cy	sec	0	0				0		2			1	1	0	0	0	1	
ser oflw	sev	0	0				0		2			1	1	0	0	1	0	
set zero	sez	0	0				0		2			1	1	0	1	0	0	
set neg	sen	0	0				0		2			1	1	1	0	0	0	
no oprn		0	0				0		2			1	0	0	0	0	0	
no oprn		0	0				0		2			1	1	0	0	0	0	
system		constant													op code			
rst bus	reset	0	0				0		0			0			5			

INSTR BITS	mnemoni	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
intr wait	wait	0	0				0		0			0			1			
halt	halt	0	0				0		0			0			0			
emulat	emt	1	1				0		0			x			x			exception SINGLE OP, vector - 30, SOP byte instr with typ=0
emult	trap	1	1				0		4			x			x			exception SINGLE OP, vector - 34, SOP byte instr with typ=0
io trap	iot	0	0				0		0			0			4			vector - 20
ex io trap	iox	0	0				0		0			0			3			vector - 14
intr retrn	rti	0	0				0		0			0			2			

[1] shashwat:
typo in specification document given as 1063 for byte rotate left