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		r	mode 0 to 7		reg 0 to 7			mode 0 to 7			reg 0 to 7								
add	add	0	6		r	mode 0 to 7		reg 0 to 7			mode 0 to 7			reg 0 to 7								
subtract	sub	1	6		ı	mode 0 to 7			reg 0 to 7		mode 0 to 7	7		reg 0 to 7								
compare	cmp	b	2		ı	mode 0 to 7		reg 0 to 7			mode 0 to 7	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		r	mode 0 to 7		reg 0 to 7			mode 0 to 7			reg 0 to 7								
JUMPS/BRANCH				typ			flg			offset												
jump	jmp	0		1	0				1		mode 0 to 7			reg 0 to 7								
jump to subrout		0			0			reg 0 to 7		mode 0 to 7			reg 0 to 7			exception in SINGLE OPS, to be treated as SINGLE OP word in						
return from subr			0	0				2	0	0 0 0			reg 0 to 7			exception in FLAG OPS, the '2' overlaps w				PS		
swap byte	swab	0	0		0		3			mode 0 to 7			reg 0 to 7									
branch	br	0	0			0		1			off											
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 o				ntinue proce	essing	
if +ve	bpl	1	0					0		offset						(helps in	processin	g typ = 0	ınstr)			
if cy	bcs	1	0			3 1				offset												
if no cy	bcc	1	0			3 0					off											
if oflw	bvs	1	0			2		1			off											
if no oflw	bvc	1	0			2		0			off											
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 0			1 1		0			offset offset											
low some	blos	1	0			3		0			offset											
lower	blo	1		0		3		1														
SINGLE OPS	bio	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											
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	h	1			1		5						reg 0 to 7								
		b		1					5		mode 0 to 7	7										
sub cy	sbc	b b		1		1			6		mode 0 to 7											
sub cy rot rt						1 2						7		reg 0 to 7 reg 0 to 7								
	sbc	b		1					6 0		mode 0 to 7	7 7		reg 0 to 7								
rot rt	sbc ror	b b		1 1		2			6		mode 0 to 7	7 7 7		reg 0 to 7 reg 0 to 7								
rot rt rot lft	sbc ror rol	b b b		1 1 1		2 2			6 0 1 [1]		mode 0 to 7 mode 0 to 7 mode 0 to 7	7 7 7 7		reg 0 to 7 reg 0 to 7 reg 0 to 7								
rot rt rot lft arith rt	sbc ror rol asr asl	b b b		1 1 1 1	cons	2 2 2 2			6 0 1 [1] 2	flg	mode 0 to 2 mode 0 to 2 mode 0 to 2	7 7 7 7	Z	reg 0 to 7 reg 0 to 7 reg 0 to 7 reg 0 to 7	С							
rot rt rot Ift arith rt arith Ift	sbc ror rol asr asl	b b b		1 1 1 1	cons	2 2 2 2			6 0 1 [1] 2	flg 1	mode 0 to 7 mode 0 to 7 mode 0 to 7 mode 0 to 7	7 7 7 7	Z 0	reg 0 to 7 reg 0 to 7 reg 0 to 7 reg 0 to 7 reg 0 to 7	С 1							
rot rt rot Ift arith rt arith Ift Program Status 6	sbc ror rol asr asl	b b b b		1 1 1 1	cons	2 2 2 2 tant			6 0 1 [1] 2 3		mode 0 to 7	7 7 7 7 7 N		reg 0 to 7 reg 0 to 7 reg 0 to 7 reg 0 to 7 reg 0 to 7								
rot rt rot Ift arith rt arith Ift Program Status (	sbc ror rol asr asl  OPS clc	b b b b b		1 1 1 1 1	cons	2 2 2 2 stant			6 0 1 [1] 2 3	1	mode 0 to 7  S/C 0	7 7 7 7 7 N 0	0	reg 0 to 7 V 0	1							
rot rt rot Ift arith rt arith Ift Program Status of clr cy clr oflw	sbc ror rol asr asl  OPS clc clv	b b b b c d d d d d d d d d d d d d d d		1 1 1 1 1 0 0	cons	2 2 2 2 2 stant 0			6 0 1 [1] 2 3	1 1	mode 0 to 7 S/C 0 0	7 7 7 7 7 N 0	0	reg 0 to 7 reg 0 to 7 V 0 1	1 0							
rot rt rot Ift arith rt arith Ift Program Status ( clr cy clr oflw clr zero	sbc ror rol asr asl  OPS clc clv clz	b b b b c 0 0		1 1 1 1 1 0 0	cons	2 2 2 2 2 2 2 2 0 0 0			6 0 1 [1] 2 3	1 1 1	mode 0 to 7 mode 0	7 7 7 7 7 N 0 0	0 0 1	reg 0 to 7 0 1 0	1 0 0							
rot rt rot Ift arith rt arith Ift  Program Status ( clr cy clr oflw clr zero clr neg	sbc ror rol asr asl  OPS clc clv clz cln	b b b b c c c c c c c c c c c c c c c c		1 1 1 1 1 1 0 0 0	cons	2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1			6 0 1 [1] 2 3 2 2 2 2 2 2 2	1 1 1 1	mode 0 to 7 mode 0	7 7 7 7 7 N 0 0 0	0 0 1 0	reg 0 to 7  V 0 1 0 0	1 0 0							
rot rt rot Ift arith rt arith Ift  Program Status ( clr cy clr oflw clr zero clr neg set cy	sbc ror rol asr asl OPS clc clv clz cln sec	b b b b c 0 0 0 0 0 0		1 1 1 1 1 1 0 0 0 0 0 0 0 0	cons	2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0			6 0 1 [1] 2 3 2 2 2 2 2 2 2 2	1 1 1 1	mode 0 to 6 mode 0	7 7 7 7 7 7 N 0 0 0 0	0 0 1 0	reg 0 to 7  V 0 1 0 0 0	1 0 0 0 1 0							
rot rt rot Ift arith rt arith Ift  Program Status ( clr cy clr oflw clr zero clr neg set cy ser oflw	sbc ror rol asr asl  OPS clc clv clz cln sec sev	b b b b b c c c c c c c c c c c c c c c		1 1 1 1 1 1 0 0 0 0 0 0	cons	2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0			6 0 1 [1] 2 3 2 2 2 2 2 2 2 2 2	1 1 1 1 1 1	mode 0 to 6 mode 0	7 7 7 7 7 7 7 7 7 7 8 8 9 9 9 9 9 9 9 9	0 0 1 0 0	reg 0 to 7  V 0 1 0 0 0 1	1 0 0 0 1							
rot rt rot Ift arith rt arith Ift Program Status ( clr cy clr zero clr neg set cy ser oflw set zero	sbc ror rol asr asl  OPS  clc clv clz cln sec sev sez	b b b b b b b c c c c c c c c c c c c c		1 1 1 1 1 1 0 0 0 0 0 0 0 0 0	cons	2 2 2 2 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4			6 0 1 [1] 2 3 2 2 2 2 2 2 2 2 2 2	1 1 1 1 1 1 1	mode 0 to 3 mode 0	7 7 7 7 7 7 7 7 7 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0	0 0 1 0 0 0	reg 0 to 7 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 1 0							
rot rt rot Ift arith rt arith Ift Program Status ( clr cy clr oflw clr zero clr neg set cy ser oflw set zero set neg	sbc ror rol asr asl  OPS  clc clv clz cln sec sev sez	b b b b b b c c c c c c c c c c c c c c		1 1 1 1 1 1 0 0 0 0 0 0 0 0 0	Cons	2 2 2 2 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4			6 0 1 [1] 2 3 2 2 2 2 2 2 2 2 2	1 1 1 1 1 1 1 1	mode 0 to 3 mode 0	7 7 7 7 7 7 7 7 7 8 8 0 0 0 0 1 0 0 0 0 1 1	0 0 1 0 0 0 0	reg 0 to 7 v 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	1 0 0 0 1 0 0							
rot rt rot Ift arith rt arith Ift Program Status ( clr cy clr oflw clr zero clr neg set cy ser oflw set zero set neg no oprn	sbc ror rol asr asl  OPS  clc clv clz cln sec sev sez	b b b b b b b c c c c c c c c c c c c c		1 1 1 1 1 1 0 0 0 0 0 0 0 0 0	Cons	2 2 2 2 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4	constant		6 0 1 [1] 2 3 2 2 2 2 2 2 2 2 2 2	1 1 1 1 1 1 1 1 1	mode 0 to 3 mode 0	7 7 7 7 7 7 7 7 7 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0	0 0 1 0 0 0 0 1 0	reg 0 to 7 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 1 0 0 0							

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			×			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 - 2	.0					
ex io trap	iox	0	0			0			0		0			3			vector - 1	4					
intr retrn	rti	0	0				0 0			0			2										

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