	Even address		Odd address		TIME µsec	Comments
	0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15		
MVM2 R2, R1	Ø Logic	DEST	SOURCE	Ø MOVE −2	3.97	1) The Source Reg is unchanged
MVM1 R2, R1		REG	REG	1 −1		2) Arithmetic Borrow/Carry
MVP1 R2, R1				2 +1	(1.52)	Propagates into the high order by
MVP2 R2, R1				3 +2		3) Add SPL's add Source Reg LO &
MOVE R2, R1				4 +Ø		Dest REG HI Result to Dest Reg
AND R2, R1				5 AND Byte		Dest REG HI is set as follows:
ORB R2, R1				6 OR	4.63	ADD SPL #1 & carry − HEX 01
XOR R2, R1				7 XOR		ADD SPL #2 & no carry " FF
ADD R2, R1				8 add	(1.72)	~~otherwise~~ " ØØ
SUB R2, R1				9 Sub		4) ~~Get to~~ Reg loads BUS IN to
ADDS1 R2, R1				A add SPL#1		Dest Reg LO
ADDS2 R2, R1				B add SPL #2		5) ~~Get~~ & ADD adds BUS IN bits
HTL R2, R1				C Copy H→L		2, 4, 8 to ~~the~~ Dest
L[illegible] R2, R1				D Copy L→H	(1.52) 3.97	Reg
GETR DA, R1		DEVICE	DEST	E GET TO Reg	(1.70)	
GETA DA, R1		ADDR	REG	F GET & ADD	(1.92) 5.30	
CTL DA, ##	1 Control	Device	Command (1.13)		2.65	Command is User Defined
PUTB DA, R1, #	4 Put Byte	Address	address	Address	(2.05) 3.97	1) The Addr Reg content is
GETB DA, R1, #	E Get Byte		Reg	modifier	4.63	modified after use
LDHI R1, R2, #	D INDIRECT HW FETCH	TO		0-3: add 1-4	(2.05)	2) Byte Fetch loads Hex ØØ
LDBI R1, R2, #	6 INDIRECT BYTE FETCH	REG		4-7: Sub 1-4		in the TO REG HI byte
STHI R1, R2, #	5 INDIRECT HW STORE	FROM		>7 no chng		
STBI R1, R2, #	7 INDIRECT BYTE STORE	REG				
LDHD R1, ##	2 DIRECT HW ~~STORE~~ FETCH	TO REG	Address		3.97	The high order Reg Byte is stored/
STHD R1, ##	3 DIRECT HW STORE	FROM REG	256 HW		(1.52)	fetched @ Addr X2
EMIT R1, ## or EA	8 EMIT	TO			(1.13) 2.65	The high order Reg Byte is
CLRI R1, ##	9 CLR BIT	REG	DATA/MASK			unchanged except for carrys &
ADDI R1, ## or EA	A ADD IMMED +1		byte		3.97	Borrows generated by the
SUBI R1, ## or EA	F SUB IMM −1	FROM REG			(1.52)	immediate operations
SE[illegible] R1, ##	B SET BIT	TO REG				
JHI/JLE R1, R2	C JUMP	DATA	MASK	Ø DATA LO/EQ	5.30	1) Tests are performed on LO order
JHE/JLO R1, R2		REG	REG	1 DATA LO	(1.92)	Reg Bytes only
JHL/JEQ R1, R2				2 EQUAL		2) Content of Regs is unchanged

	DATA REG	MASK REG		
JSB 8/JNO R1			3 no data bits	3) 8-F cause the same tests, But Jump on False
JSN J/JALL R1			4 all Data Bits	
JSNM 1/JALLM R1, R2			5 Data = all mask bits	4) If no Jump, Reg 1 is loaded with Jump instr Addr +4
JSM 1/JNOM R1, R2			6 Data = no mask bits	
JHSNM 1/JHAM R1, R2			7	

same as modifier 5 but on high order byte of R1 using low order byte of R2 as a mask (2.12 usec)

PUTB	2.1
LDHI } SHI LIB	2.2
GETB	2.2
CTL } EHM	2.3
CLRI } [illegible] LDAB SHIL MOVES	2.6
LTH	
AND OR XOR JUMP	2.7

LONGEST microstep 2.05 µs

New Instructions

SHIFTR
SHTR REG, 1 GETB ∅, { C = shift right 1, fill with high order byte low bit
D = rotate right 1 1.72 µsec
E = rotate right 3
F = rotate right 4

ROTR
RTR REG, 4