LOAD GROUP	Immediate	X Register	Y Register	A Register	8 Bit Memory	X 8 bit memory	Y 8 bit Memory	Y 8 bit Memory Y+Y+1	Y=Y-1 Y 8 bit Memory	8 bit Memory + X	8 Bit Memory + Y	16 bit Memory	16 Bit Memory XY	16 Bit Memory + X	16 Bit Memory + Y	XY Register + N
A Register	LDAI 73	LDAX 38	LDAY 39	LDAF ca	LDA(M) 6e	LDA(X) 78	LDA(Y) 79	LDA(Y)+ 7f	LDA-(Y) e7	LDA(M+X) 7c	LDA(M+Y) 7d	LDA(MM) 8b	LDA(XY) 74	LDA(MM+X) 80	LDA(MM+Y) 84	LDA(XY+N) 82
X Register	LDXI 75		LDXY 68	LDXA f3	LDX(M) 6f		LDX(Y) 7a				LDX(M+Y) 5b	LDX(MM) 8c			LDX(MM+Y) 50	
Y Register	LDYI 70	LDYX 58		LDYA 72	LDY(M) 76	LDY(X) 59				LDY(M+X) ef		LDY(MM) 8d				
X and Y Register	LDXYII 77				LDXY(M) a5	LDXY(X) a4	LDXY(Y) a3									
STORE GROUP																
A Register					STA(M) 88	STA(X) 69	STA(Y) e8	STA(Y)+ 7e	STA-(Y) e6	STA(M+X) a8	STA(M+Y) a9	STA(MM) a1	STA(XY) 95	STA(MM+X) 81	STA(MM+Y) 85	STA(XY+N) 83
X Register					STX(M) 89		STX(Y) 93				STX(M+Y) 5d	STX(MM) 9a			STX(MM+Y) 51	
Y Register					STY(M) 6c	STY(X) 94				STY(M+X) be		STY(MM) f0				
Immediate					STI(M) a0	STI(X) f1	STI(Y) f2									
Clear to 0		CLX 02	CLY 01	CLA 05	CL(M) 06	CL(X) 04	CL(Y) 07				CL(M+Y) 3f	CL(MM) 3c				
ALU Instructions (ADC, SBC, CMP, OR, XOR, AND) Result in Except CMP														XY Register		
A Register	*AI 08-0d	*AX 20-25	*AY d0-d5		*A(M) 48-4d	*A(X) d8-dd	*A(Y) 60-65			*A(M+X) 28-2d	*A(M+Y) 30-35	*(MM) b0-b5	*A(XY) c0-c5			
X Register	*XI 10-15		*XY 40-45													
Y Register	*YI 18-1d															
XY Register (No CMP or Flags)	*XYII e0-e5															
Y 8 Bit Memory (ADC SBC CMP)																
ROR / ROL		*X 8e/8f	*Y 96/97	*A 86/87	*(M) a6 a7	*(X) b6/b7	*(Y) d6/d7				*(M+Y) ae/af			*XY 9e/9f		
SHR / SHL		*X 16/17	*Y 1e/1f	*A 0e/0f	*(M) 66/67	*(X) 46/47	*(Y) 4e/4f				*(M+Y) 36/37			*XY 27/28		
INC		*X 99	*Y 9d	*A 98	*(M) 8a	*(X) 90	*(Y) 91				*(M+Y) 55	*(MM) 9c		*XY (No Flag) 54		
DEC (ZF only)		*X df	*Y ce	*A de	*(M) 92	*(X) 52	*(Y) 5c				*(M+Y) cd	*(MM) ff ed		*XY (No Flag) 3a		
SW (Swap values)		*XY ff eb		*AY ff ec												
TST (C Flag Reset, ZF)		*X ff 75	*Y ff 73	*A 6d	*(M) 3b	*(X) 6b	*(Y) 3d			*(M+X) 7b	*(M+Y) 9b	*(MM) - ff 05		*XY 03		
ADD / SUB without Carry																
X Register ADD	*XI ff f4		*XY ff 38	*XA c8												
Y Register ADD	*YI ff f5	*YX ff 39		*YA c9												
SP Stack Pointer ADD	*SPI ff ca			*SPA - ff 02												
A Register ADD No FLG	*AI ff 68															
A Register SUB	*AI e2															
A Register Bit 7 to CF				LDCAS cf												

Stack, Jump, Call	No Parameters							XY Register
PUSH / POP				*AF fe/ee				*XY b8/b9
JP	ab				*(M) 5f		*(Y) 5e	*XY ad
JPS (Same 256 byte page)	eb					*(X) ff f3	*(Y) ff ac	
CALL	2e							*XY 56
RET / RETI	57 / f8							

Special Register Functions	Save	Load	Constant	4x Roll Left
Stack Pointer	LDASP cc	LSPA 2f		
Vector	LDAV ff cc	LDVA 71		
A Register			LDAONE ac	ROLAM bf

Conditional Jump / Return	Carry	No Carry	Zero	No Zero
RET	*C f6	*NC f7	*Z e9	*NZ ea
JP	*C ba	*NC bb	*Z bc	*NZ bd
JPS (Same 256 byte page)	*C f4	*NC f5	*Z ec	*NZ ed
DECYJ (Same Page)			*Z c6	*NZ c7
CPAIJ (Same Page)	*C aa	*NC a2	*Z 5a	*NZ 6a
TSTXJ (Same Page)			*Z cb	*NZ 53

Miscellaneous Instructions	
No Operation	NOP 00
Enable Interrupts	EIT f9
Disable Interrupts	DIT fa
Complement Carry Flag	CCF fd
Set Carry Flag	SCF fb
Clear Carry Flag	CLC fc
Jump to Vector at vv	BRK 3e vv