Instr	HEX	ASSM	Func	Bytes	Codes
0	00	AOUT	ACCA => DISPLAY	1	
1	01	BOUT	ACCB => DISPLAY	1	
2	02	MOUT	M[REGY] => DISPLAY	1	
3	03	INA	DATA => ACCA (wait for it)	1	Z
4	04	KEY	DATA => ACCA (real-time)	1	Z
5	05	TMR	Timer => ACCA elapsed secs	1	Z
6	06	RTMR	0 => Timer	1	
7	07	CLS	Clears the HEX digit display	1	
8	08	VBAT	Battery Voltage => ACCA	1	Z
9	09	LDA	d => ACCA	2	Z
10	0A	LDB	d => ACCB	2	Z
11	0B	LDY	d => REGY	2	Z
12	0C	CLA	0 => ACCA	1	
13	0D	CLB	0 => ACCB	1	
14	0E	INCA	ACCA + 1 => ACCA	1	Z,C
15	0F	INCB	ACCB + 1 => ACCB	1	Z,C
16	10	INCY	REGY + 1 => REGY	1	Z,C
17	11	DECA	ACCA - 1 => ACCA	1	Z,N
18	12	DECB	ACCB - 1 => ACCB	1	Z,N
19	13	DECY	REGY - 1 => REGY	1	Z,N
20	14	SWP	ACCA <=> ACCB	1	Z
21	15	SYZ	REGY <=> REGZ	1	Z
22	16	ROLA	ROLL ACCA 1 BIT LEFT	1	Z,C
23	17	RORA	ROLL ACCA 1 BIT RIGHT	1	Z,C
24	18	CMPA	(ACCA-d); Z if 0, N if <0	2	Z,N
25	19	CMPB	(ACCB-d); Z if 0, N if <0	2	Z,N
26	1A	CMAB	(ACCA-ACCB); Z if 0, N if <0	1	Z,N
27	1B	MIA	M[REGY] => ACCA	1	Z
28	1C	AIM	ACCA => M[REGY]	1	Z
29	1D	MIB	M[REGY] => ACCB	1	Z
30	1E	BIM	ACCB => M[REGY]	1	Z,N
31	1F	AAM	ACCA+M[REGY] => M[REGY]	1	Z,C

32	**20**	**AMA**	ACCA+M[REGY] => ACCA	1	Z,C
33	**21**	**SAM**	M[REGY]-ACCA => M[REGY]	1	Z,N
34	**22**	**JMP**	d=> PC	2	
35	**23**	**WAIT**	Stop until INC pressed	1	
36	**24**	**STOP**	Stop program execution	1	
37	**25**	**DLYA**	DELAY = ACCA x 0.1 sec	1	
38	**26**	**DLYB**	DELAY = ACCB x 0.1 sec	1	
39	**27**	**BEQ**	If Z, d => PC	2	
40	**28**	**BNE**	If not Z, d => PC	2	
41	**29**	**BLT**	If N, d => PC	2	
42	**2A**	**BLE**	If Z or N, d => PC	2	
43	**2B**	**BGT**	If not (Z or N), d => PC	2	
44	**2C**	**BGE**	If not N, d => PC	2	
45	**2D**	**BCS**	If C, d=>PC	2	
46	**2E**	**ADDA**	ACCA+d => ACCA	2	Z,C
47	**2F**	**ADAB**	ACCA+ACCB => ACCA	1	Z,C
48	**30**	**SUBA**	ACCA-d => ACCA	2	Z,N
49	**31**	**SBA**	ACCA-ACCB => ACCA	1	Z,N
50	**32**	**SFLA**	SHIFT LEFT ACCA 1 BIT	1	Z,C
51	**33**	**SFLB**	SHIFT LEFT ACCB 1 BIT	1	Z,C
52	**34**	**SFRA**	SHIFT RIGHT ACCA 1 BIT	1	Z,C
53	**35**	**SFRB**	SHIFT RIGHT ACCB 1 BIT	1	Z,C
54	**36**	**ANDA**	ACCA (AND) d => ACCA	2	Z
55	**37**	**ORA**	ACCA (OR) d => ACCA	2	Z
56	**38**	**XORA**	ACCA (XOR) d => ACCA	2	Z
57	**39**	**AAB**	ACCA (AND) ACCB => ACCA	1	Z
58	**3A**	**AOB**	ACCA (OR) ACCB => ACCA	1	Z
59	**3B**	**AXOB**	ACCA (XOR) ACCB => ACCA	1	Z
60	**3C**	**MULA**	ACCA X d =>M[254,255]	2	Z
61	**3D**	**MAB**	ACCA X ACCB => M[254,255]	1	Z
62	**3E**	**RNDA**	RANDOM (0-255) => ACCA	1	Z
63	**3F**	**PSHA**	ACCA =>STACK	1	Z
64	**40**	**POPA**	STACK=>ACCA	1	Z

65	**41**	**PSHB**	ACCB=>STACK	1	Z
66	**42**	**POPB**	STACK=>ACCB	1	Z
67	**43**	**PSHY**	REGY=>STACK	1	Z
68	**44**	**POPY**	STACK=>REGY	1	Z
69	**45**	**CALL**	PC=>d; rtn=>PC+1	2	
70	**46**	**RTN**	PC=>rtn	1	