Hex.	Binary		Dave OpCode	Address	Notes	Registers
00	0000	0000	hlt		Halts the CPU	AX/AH/AL
01	0000	0001	nop		No operation	BX/BH/BL
02	0000	0010	fsn		Set the sign flag	RT/RH/RL
03	0000	0011	fcn		Clear the sign flag	DX/DH/DL
04	0000	0100	fsi		Set the interrupt flag	EX/EH/EL
05	0000	0101	fci		Clear the interrupt flag	SP
06	0000	0110	fsc		Set the carry flag	PC (DON'T USE)
07	0000	0111	fcc		Clear the carry flag	EXTRA (DON'T USE)
08	0000	1000	fca		Clear all flags	
09	0000	1001	ret		Return from function	
0a	0000	1010	reti		Return from interrupt	
0b	0000	1011	??			
0c	0000	1100	??			
0d	0000	1101	??			
0e	0000	1110	dly		Delay 4.5ms	
0f	0000	1111	??			
10	0001	0000	bcs	PC+N	Branch if carry set	
11	0001	0001	bcc	PC+N	Branch if carry clear	
12	0001	0010	bns	PC+N	Branch if negative set	
13	0001	0011	bnc	PC+N	Branch if negative clear	
14	0001	0100	bzs	PC+N	Branch if zero set (Branch if equal)	
15	0001	0101	bzc	PC+N	Branch if zero clear (Branch if not equal)	
16	0001	0110	blt	PC+N	Branch if less than	
17	0001	0111	bge	PC+N	Branch if greater than or equal	
18	0001	1000	bgt	PC+N	Branch if greater than	
19	0001	1001	ble	PC+N	Branch if less than or equal	
1a	0001	1010	bs1	PC+N	Branch if sense switch 1 is set	
1b	0001	1011	bs2	PC+N	Branch if sense switch 2 is set	
1c	0001	1100	bs3	PC+N	Branch if sense switch 3 is set	
1d	0001	1101	bs4	PC+N	Branch if sense switch 4 is set	
1e	0001	1110				
1f	0001	1111				
20	0010	0000	inc	_L/H	Increment byte of explicit register	
21	0010	0001	dec	_L/H	Decrement byte of explicit register	
22	0010	0010	clr	_L/H	Clear byte of explicit register (22 32 = CPU ID)	
23	0010	0011	not	_L/H	Invert byte of explicit register	
24	0010	0100	lsl	_L/H	Shift byte of explicit register left	
25	0010	0101	lsr	_L/H	Shift byte of explicit register right	
26	0010	0110	rrc	_L/H	Rotate byte of explicit register right (wraps through carry)	
27	0010	0111	rlc	_L/H	Rotate byte of explicit register left (wraps through carry)	
28	0010	1000	inc	AL	Increment byte of implicit AL register	
29	0010	1001	dec	AL	Decrement byte of implicit AL register	
2a	0010	1010	clr	AL	Clear byte of implicit AL register	
2b	0010	1011	not	AL	Invert byte of implicit AL register	
2c	0010	1100	lsl	AL	Shift byte of implicit AL register left	
2d	0010	1101	lsr	AL	Shift byte of implicit AL register right	
2e	0010	1110			Memory mapping?	
2f	0010	1111			DMA	
30	0011	0000	inc	_X	Increment full word of explicit register	
31	0011	0001	dec	_X	Decrement full word of explicit register	
32	0011	0010	clr	_X	Clear full word of explicit register	
33	0011	0011	not	_X	Invert full word of explicit register	
34	0011	0100	lsl	_X	Shift full word of explicit register left	
35	0011	0101	lsr	_X	Shift full word of explicit register right	
36	0011	0110	rrc	_X	Rotate full word of explicit register right	
37	0011	0111	rlc	_X	Rotate full word of explicit register left	
38	0011	1000	inc	AX	Increment full word of implicit AX register	
39	0011	1001	dec	AX	Decrement full word of implicit AX register	
3a	0011	1010	clr	AX	Clear full word of implicit AX register	
3b	0011	1011	not	AX	Invert full word of implicit AX register	
3c	0011	1100	lsl	AX	Shift full word of implicit AX register left	
3d	0011	1101	lsr	AX	Shift full word of implicit AX register right	
3e	0011	1110	inc	RT	Increment full word of implicit RT register	
3f	0011	1111	dec	RT	Decrement full word of implicit RT register	
40	0100	0000	add	_L/H, _L/H	Add bytes of two explicit registers (left plus right stored in left)	
41	0100	0001	sub	_L/H, _L/H	Subtract bytes of two explicit registers (left minus right stored in left)	
42	0100	0010	and	_L/H, _L/H	AND bytes of two explicit registers (left AND right stored in left)	
43	0100	0011	or	_L/H, _L/H	OR bytes of two explicit registers (left OR right stored in left)	
44	0100	0100	xor	_L/H, _L/H	XOR bytes of two explicit registers (left XOR right stored in left)	

45	0100	0101	mov	_L/H, _L/H	Copy byte of one explicit register into other explicit register (right into left)
46	0100	0110			
47	0100	0111			Execute micro code?
48	0100	1000	add	BL, AL	Add bytes of implicit AL and BL (AL plus BL stored in BL)
49	0100	1001	sub	BL, AL	Subtract bytes of implicit AL and BL (AL minus BL stored in BL)
4a	0100	1010	and	BL, AL	AND bytes of implicit AL and BL (AL AND BL stored in BL)
4b	0100	1011	or	BL, AL	OR bytes of implicit AL and BL (AL OR BL stored in BL)
4c	0100	1100	xor	BL, AL	XOR bytes of two implicit registers (AL XOR BL stored in BL)
4d	0100	1101	mov	BL, AL	Copy byte of one implicit register into other explicit register (AL into BL)
4e	0100	1110			
4f	0100	1111			
50	0101	0000	add	_X, _X	Add two explicit registers (left plus right stored in left)
51	0101	0001	sub	_X, _X	Subtract two explicit registers (left minus right stored in left)
52	0101	0010	and	_X, _X	AND two explicit registers (left AND right stored in left)
53	0101	0011	or	_X, _X	OR two explicit registers (left OR right stored in left)
54	0101	0100	xor	_X, _X	XOR two explicit registers (left XOR right stored in left)
55	0101	0101	mov	_X, _X	Copy one explicit register into other explicit register (right into left)
56	0101	0110			
57	0101	0111			
58	0101	1000	add	BX, AX	Add implicit AX and BX (AX plus BX stored in BX)
59	0101	1001	sub	BX, AX	Subtract implicit AX and BX (AX minus BX stored in BX)
5a	0101	1010	and	BX, AX	AND implicit AX and BX (AX AND BX stored in BX)
5b	0101	1011	or	BX, AX	OR implicit AX and BX (AX OR BX stored in BX)
5c	0101	1100	mov	DX, AX	Copy byte of one implicit register into other explicit register (AX into DX)
5d	0101	1101	mov	BX, AX	Copy byte of one implicit register into other explicit register (AX into BX)
5e	0101	1110	mov	EX, AX	Copy byte of one implicit register into other explicit register (AX into EX)
5f	0101	1111	mov	SP, AX	Copy byte of one implicit register into other explicit register (AX into SP)
60	0110	0000	ld	CX, #Imm.	Load immediate address into full word CX
61	0110	0001	ld	CX, Addr.	Load direct address into full word CX
62	0110	0010	ld	CX, [Addr.]	Load indirect address into full word CX
63	0110	0011	ld	CX, PC+N	Load direct Program Counter offset by N address into full word CX
64	0110	0100	ld	CX, [PC+N]	Load indirect Program Counter offset by N address into full word CX
65	0110	0101	ld	CX, _[R]_	Load indexed mode register into full word CX
66	0110	0110			
67	0110	0111			
68	0110	1000	st	CX, #Imm.	Store full word of CX into immediate address
69	0110	1001	st	CX, Addr.	Store full word of CX into direct address
6a	0110	1010	st	CX, [Addr.]	Store full word of CX into indirect address
6b	0110	1011	st	CX, PC+N	Store full word of CX into direct Program Counter offset by N address
6c	0110	1100	st	CX, [PC+N]	Store full word of CX into indirect Program Counter offset by N address
6d	0110	1101	st	CX, _[R]_	Store full word of CX into indexed mode register
6e	0110	1110			
6f	0110	1111			
70	0111	0000	jump	#D	Jump to immediate address
71	0111	0001	jump	A	Jump to direct address
72	0111	0010	jump	[A]	Jump to indirect address
73	0111	0011	jump	PC+N	Jump to direct Program Counter offset by N address
74	0111	0100	jump	[PC+N]	Jump to indirect Program Counter offset by N address
75	0111	0101	jump	_[R]_	Jump to indexed mode register
76	0111	0110	syscall		Call interrupt level 15
77	0111	0111			
78	0111	1000	call	#D	Call immediate address
79	0111	1001	call	A	Call direct address
7a	0111	1010	call	[A]	Call indirect address
7b	0111	1011	call	PC+N	Call direct Program Counter offset by N address
7c	0111	1100	call	[PC+N]	Call indirect Program Counter offset by N address
7d	0111	1101	call	_[R]_	Call indexed mode register
7e	0111	1110			Memory banking?
7f	0111	1111			Memory banking?
80	1000	0000	ld	AL, #Imm.	Load immediate address into byte of AL register
81	1000	0001	ld	AL, Addr.	Load direct address into byte of AL register
82	1000	0010	ld	AL, [Addr.]	Load indirect address into byte of AL register
83	1000	0011	ld	AL, PC+N	Load direct Program Counter offset by N address into byte of AL register
84	1000	0100	ld	AL, [PC+N]	Load indirect Program Counter offset by N address into byte of AL register
85	1000	0101	ld	AL, _[R]_	Load indexed register into byte of AL register
86	1000	0110			
87	1000	0111			
88	1000	1000	ld	AL, [AX]	Load byte from memory address stored in AX into AL register
89	1000	1001	ld	AL, [BX]	Load byte from memory address stored in BX into AL register
8a	1000	1010	ld	AL, [RT]	Load byte from memory address stored in RT into AL register

8b	1000	1011	ld	AL, [DX]	Load byte from memory address stored in DX into AL register
8c	1000	1100	ld	AL, [EX]	Load byte from memory address stored in EX into AL register
8d	1000	1101	ld	AL, [SP]	Load byte from memory address stored in SP into AL register
8e	1000	1110			
8f	1000	1111			
90	1001	0000	ld	AX, #Imm.	Load immediate address into full word of AX register
91	1001	0001	ld	AX, Addr.	Load direct address into full word of AX register
92	1001	0010	ld	AX, [Addr.]	Load indirect address into full word of AX register
93	1001	0011	ld	AX, PC+N	Load direct Program Counter offset by N address into full word of AX register
94	1001	0100	ld	AX, [PC+N]	Load indirect Program Counter offset by N address into full word of AX register
95	1001	0101	ld	AX, _[R]_	Load indexed register into full word of AX register
96	1001	0110			
97	1001	0111			
98	1001	1000	ld	AX, [AX]	Load word from memory address stored in AX into AX register
99	1001	1001	ld	AX, [BX]	Load word from memory address stored in BX into AX register
9a	1001	1010	ld	AX, [RT]	Load word from memory address stored in RT into AX register
9b	1001	1011	ld	AX, [DX]	Load word from memory address stored in DX into AX register
9c	1001	1100	ld	AX, [EX]	Load word from memory address stored in EX into AX register
9d	1001	1101	ld	AX, [SP]	Load word from memory address stored in SP into AX register
9e	1001	1110			
9f	1001	1111			
a0	1010	0000			
a1	1010	0001	st	AL, Addr.	Store byte of AL register into direct address
a2	1010	0010	st	AL, [Addr.]	Store byte of AL register into indirect address
a3	1010	0011	st	AL, PC+N	Store byte of AL register into direct Program Counter offset by N address
a4	1010	0100	st	AL, [PC+N]	Store byte of AL register into indirect Program Counter offset by N address
a5	1010	0101	st	AL, _[R]_	Store byte of AL register into indexed register
a6	1010	0110			
a7	1010	0111			
a8	1010	1000	st	AL, [AX]	Store byte from AL register to memory address stored in AX
a9	1010	1001	st	AL, [BX]	Store byte from AL register to memory address stored in BX
aa	1010	1010	st	AL, [RT]	Store byte from AL register to memory address stored in RT
ab	1010	1011	st	AL, [DX]	Store byte from AL register to memory address stored in DX
ac	1010	1100	st	AL, [EX]	Store byte from AL register to memory address stored in EX
ad	1010	1101	st	AL, [SP]	Store byte from AL register to memory address stored in SP
ae	1010	1110			
af	1010	1111			
b0	1011	0000			
b1	1011	0001	st	AX, Addr.	Store AX register into direct address
b2	1011	0010	st	AX, [Addr.]	Store AX register into indirect address
b3	1011	0011	st	AX, PC+N	Store AX register into direct Program Counter offset by N address
b4	1011	0100	st	AX, [PC+N]	Store AX register into indirect Program Counter offset by N address
b5	1011	0101	st	AX, _[R]_	Store AX register into indexed register
b6	1011	0110			
b7	1011	0111			
b8	1011	1000	st	AX, [AX]	Store word from AX register to memory address stored in AX
b9	1011	1001	st	AX, [BX]	Store word from AX register to memory address stored in BX
ba	1011	1010	st	AX, [RT]	Store word from AX register to memory address stored in RT
bb	1011	1011	st	AX, [DX]	Store word from AX register to memory address stored in DX
bc	1011	1100	st	AX, [EX]	Store word from AX register to memory address stored in EX
bd	1011	1101	st	AX, [SP]	Store word from AX register to memory address stored in SP
be	1011	1110			
bf	1011	1111			
c0	1100	0000	ld	BL, #Imm.	Load immediate address into byte ofBL register
c1	1100	0001	ld	BL, Addr.	Load direct address into byte of BL register
c2	1100	0010	ld	BL, [Addr.]	Load indirect address into byte of BL register
c3	1100	0011	ld	BL, PC+N	Load direct Program Counter offset by N address into byte of BL register
c4	1100	0100	ld	BL, [PC+N]	Load indirect Program Counter offset by N address into byte of BL register
c5	1100	0101	ld	BL, _[R]_	Load indexed register into byte of BL register
c6	1100	0110			
c7	1100	0111			
c8	1100	1000	ld	BL, [AX]	Load byte from memory address stored in AX into BL register
c9	1100	1001	ld	BL, [BX]	Load byte from memory address stored in BX into BL register
ca	1100	1010	ld	BL, [RT]	Load byte from memory address stored in RT into BL register
cb	1100	1011	ld	BL, [DX]	Load byte from memory address stored in DX into BL register
cc	1100	1100	ld	BL, [EX]	Load byte from memory address stored in EX into BL register
cd	1100	1101	ld	BL, [SP]	Load byte from memory address stored in SP into BL register
ce	1100	1110			
cf	1100	1111			
d0	1101	0000	ld	BX, #Imm.	Load immediate address into full word of BX register

d1	1101	0001	ld	BX, Addr.	Load direct address into full word of BX register
d2	1101	0010	ld	BX, [Addr.]	Load indirect address into full word of BX register
d3	1101	0011	ld	BX, PC+N	Load direct Program Counter offset by N address into full word of BX register
d4	1101	0100	ld	BX, [PC+N]	Load indirect Program Counter offset by N address into full word of BX register
d5	1101	0101	ld	BX, _[R]_	Load indexed register into full word of AX register
d6	1101	0110			
d7	1101	0111			
d8	1101	1000	ld	BX, [AX]	Load word from memory address stored in AX into BX register
d9	1101	1001	ld	BX, [BX]	Load word from memory address stored in BX into BX register
da	1101	1010	ld	BX, [RT]	Load word from memory address stored in RT into BX register
db	1101	1011	ld	BX, [DX]	Load word from memory address stored in DX into BX register
dc	1101	1100	ld	BX, [EX]	Load word from memory address stored in EX into BX register
dd	1101	1101	ld	BX, [SP]	Load word from memory address stored in SP into BX register
de	1101	1110			
df	1101	1111			
e0	1110	0000			
e1	1110	0001	st	BL, Addr.	Store byte of BL register into direct address
e2	1110	0010	st	BL, [Addr.]	Store byte of BL register into indirect address
e3	1110	0011	st	BL, PC+N	Store byte of BL register into direct Program Counter offset by N address
e4	1110	0100	st	BL, [PC+N]	Store byte of BL register into indirect Program Counter offset by N address
e5	1110	0101	st	BL, _[R]_	Store byte of BL register into indexed register
e6	1110	0110			
e7	1110	0111			
e8	1110	1000	st	BL, [AX]	Store byte from BL register to memory address stored in AX
e9	1110	1001	st	BL, [BX]	Store byte from BL register to memory address stored in BX
ea	1110	1010	st	BL, [RT]	Store byte from BL register to memory address stored in RT
eb	1110	1011	st	BL, [DX]	Store byte from BL register to memory address stored in DX
ec	1110	1100	st	BL, [EX]	Store byte from BL register to memory address stored in EX
ed	1110	1101	st	BL, [SP]	Store byte from BL register to memory address stored in SP
ee	1110	1110			
ef	1110	1111			
f0	1111	0000			
f1	1111	0001	st	BX, Addr.	Store BX register into direct address
f2	1111	0010	st	BX, [Addr.]	Store BX register into indirect address
f3	1111	0011	st	BX, PC+N	Store BX register into direct Program Counter offset by N address
f4	1111	0100	st	BX, [PC+N]	Store BX register into indirect Program Counter offset by N address
f5	1111	0101	st	BX, _[R]_	Store BX register into indexed register
f6	1111	0110			
f7	1111	0111			
f8	1111	1000	st	BX, [AX]	Store word from BX register to memory address stored in AX
f9	1111	1001	st	BX, [BX]	Store word from BX register to memory address stored in BX
fa	1111	1010	st	BX, [RT]	Store word from BX register to memory address stored in RT
fb	1111	1011	st	BX, [DX]	Store word from BX register to memory address stored in DX
fc	1111	1100	st	BX, [EX]	Store word from BX register to memory address stored in EX
fd	1111	1101	st	BX, [SP]	Store word from BX register to memory address stored in SP
fe	1111	1110			
ff	1111	1111			