REDUCED 8088 INSTRUCTION SET

Edited by Phil Townshend 2003

INSTRUCTION		HEX CODE			FUNCTION	CYCS	Z-Flag
ADD	AL,n	04	n		AL << AL + n	2	*
AND	AL,n	24	n		AL << AL AND n	2	*
CALL	BX	FF	D3		(SP) << IP, IP << BX	2	
DEC	CX	49			CX << CX - 1	1	*
DEC	DX	4A			DX << DX - 1	1	*
DEC	AL	FE	C8		AL << AL - 1	2	*
HALT		76			Halts processor	1	
IN	AL,DX	EC			AL << IN (DX)	1	*
INC	DX	42			DX << DX + 1	1	*
INC	BX	43			BX << BX + 1	1	*
INC	AL	FE	C0		AL << AL + 1	2	*
JMP	fg	EB	e		IP << IP + e = fg	2	
JNZ	fg	75	e		IP << IP + e = fg IF F=0	2	
JZ	fg	74	e		IP << IP + e = fg IF F=1	2	
MOV	[BX],AL	88	07		(BX) << AL	2	*
MOV	CX,AX	89	C1		CX << AX	2	
MOV	BX,AX	89	C3		BX << AX	2	
MOV	AL,[BX]	8A	07		AL << (BX)	2	*
MOV	AL,n	B0	n		AL << n	2	*
MOV	AX.mn	B8	n	m	AX << mn	3	*
MOV	CX,mn	B9	n	m	CX << mn	3	*
MOV	DX,mn	BA	n	m	DX << mn	3	*
MOV	BX,mn	BB	n	m	BX << mn	3	*
MOV	SP,mn	BC	n	m	SP << mn		
OR	AL,n	0C	n		AL << AL OR n	2	*
OUT	DX,AL	EE			OUT (DX) << AL	1	*
POP	AX	58			AX << (SP)	1	
POP	CX	59			CX << (SP)	1	
POP	DX	5A			DX << (SP)	1	
POP	BX	5B			BX << (SP)	1	
PUSH	AX	50			(SP) << AX	1	
PUSH	CX	51			(SP) << CX	1	
PUSH	DX	52			(SP) << DX	1	
PUSH	BX	53			(SP) << BX	1	
RET		C3			IP << (SP)	1	
SHL	AL,1	D0	E0		AL << AL x 2	2	*
SHR	AL,1	D0	E8		AL << AL / 2	2	*
SUB	AL,n	2C	n		AL << AL - n	2	*
XOR	AL,n	34	n		AL << AL EOR n	2	*

REDUCED 8088 INSTRUCTION SET

Edited by Phil Townshend 2003

INSTRUCTION		HEX CODE			FUNCTION	CYCS	Z-Flag
ADD	AL,n	04	n		AL << AL + n	2	*
AND	AL,n	24	n		AL << AL AND n	2	*
CALL	BX	FF	D3		(SP) << IP, IP << BX	2	
DEC	CX	49			CX << CX - 1	1	*
DEC	DX	4A			DX << DX - 1	1	*
DEC	AL	FE	C8		AL << AL - 1	2	*
HALT		76			Halts processor	1	
IN	AL,DX	EC			AL << IN (DX)	1	*
INC	DX	42			DX << DX + 1	1	*
INC	BX	43			BX << BX + 1	1	*
INC	AL	FE	C0		AL << AL + 1	2	*
JMP	fg	EB	e		IP << IP + e = fg	2	
JNZ	fg	75	e		IP << IP + e = fg IF F=0	2	
JZ	fg	74	e		IP << IP + e = fg IF F=1	2	
MOV	[BX],AL	88	07		(BX) << AL	2	*
MOV	CX,AX	89	C1		CX << AX	2	
MOV	BX,AX	89	C3		BX << AX	2	
MOV	AL,[BX]	8A	07		AL << (BX)	2	*
MOV	AL,n	B0	n		AL << n	2	*
MOV	AX.mn	B8	n	m	AX << mn	3	*
MOV	CX,mn	B9	n	m	CX << mn	3	*
MOV	DX,mn	BA	n	m	DX << mn	3	*
MOV	BX,mn	BB	n	m	BX << mn	3	*
MOV	SP,mn	BC	n	m	SP << mn		
OR	AL,n	0C	n		AL << AL OR n	2	*
OUT	DX,AL	EE			OUT (DX) << AL	1	*
POP	AX	58			AX << (SP)	1	
POP	CX	59			CX << (SP)	1	
POP	DX	5A			DX << (SP)	1	
POP	BX	5B			BX << (SP)	1	
PUSH	AX	50			(SP) << AX	1	
PUSH	CX	51			(SP) << CX	1	
PUSH	DX	52			(SP) << DX	1	
PUSH	BX	53			(SP) << BX	1	
RET		C3			IP << (SP)	1	
SHL	AL,1	D0	E0		AL << AL x 2	2	*
SHR	AL,1	D0	E8		AL << AL / 2	2	*
SUB	AL,n	2C	n		AL << AL - n	2	*
XOR	AL,n	34	n		AL << AL EOR n	2	*