http://www.chibiakumas.com/asm

Mnemonic	Function	Notes	N Z V C	Opcode (Octal) F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
HALT	Stop		- - - -	**0**		**0**			**0**			**0**			**0**			**0**	
WAIT	Stop until interrupt		- - - -	**0**		**0**			**0**			**0**			**0**			**1**	
RESET	Reset all IO devices		- - - -	**0**		**0**			**0**			**0**			**0**			**5**	
NOP	NoOp		- - - -	**0**		**0**			**0**			**2**			**4**			**0**	
CLR{B}	Set to zero		0 1 0 0	**0**		**0**			**5**			**0**			*D*			*D*	
INC{B}	Add 1		* * * -	**0**		**0**			**5**			**2**			*D*			*D*	
DEC{B}	Sub 1		* * * -	**0**		**0**			**5**			**3**			*D*			*D*	
ADC{B}	Add with carry		* * * *	**0**		**0**			**5**			**5**			*D*			*D*	
SBC{B}	Subtract with Carry		* * * *	**0**		**0**			**5**			**6**			*D*			*D*	
TST{B}	Set Condition Codes		* * 0 0	**0**		**0**			**5**			**7**			*D*			*D*	
NEG{B}	Negate		* * * *	**0**		**0**			**5**			**4**			*D*			*D*	
COM{B}	Ones compliment		* * 0 1	**0**		**0**			**5**			**1**			*D*			*D*	
ROR{B}	Rotate Right (through Carry)		* * * *	**0**		**0**			**6**			**0**			*D*			*D*	
ROL{B}	Rotate Left (through Carry)		* * * *	**0**		**0**			**6**			**1**			*D*			*D*	
ASR{B}	Arithmatic shift Right		* * * *	**0**		**0**			**6**			**2**			*D*			*D*	
ASL{B}	Arithmatic shift Left		* * * *	**0**		**0**			**6**			**3**			*D*			*D*	
SWAB	Swap Bytes in a word		* * * 0	**0**		**0**			**0**			**3**			*D*			*D*	
SXT	Sign Extend		- * 0 -	**0**		**0**			**6**			**7**			*D*			*D*	
MUL	Multiply		* * 0 *	**0**		**7**			**0**			*R*			*S*			*S*	
DIV	Divide		* * * *	**0**		**7**			**1**			*R*			*S*			*S*	
ASH	Arithmatic shift (by val in reg)		* * * *	**0**		**7**			**2**			*R*			*S*			*S*	
ASHC	Arithmatic shift combined		* * * *	**0**		**7**			**3**			*R*			*S*			*S*	
XOR	Flip bits		* * 0 -	**0**		**7**			**4**			*R*			*S*			*S*	
MOV{B}	Move			B		**1**			*S*			*S*			*D*			*D*	
ADD	Add			**0**		**6**			*S*			*S*			*D*			*D*	
SUB	Subtract			**1**		**6**			*S*			*S*			*D*			*D*	
CMP{B}	Compare (effective -)			B		**2**			*S*			*S*			*D*			*D*	
BIS{B}	Bit Set (OR)			B		**5**			*S*			*S*			*D*			*D*	
BIC{B}	Bit Clear (for AND use with COM to flip bits)			B		**4**			*S*			*S*			*D*			*D*	
BIT{B}	Bit Test (like AND but doesn't alter dest)			B		**3**			*S*			*S*			*D*			*D*	
BR	Branch Always			**0**		**0**			**0**		**1**	*B*	*B*		*B*			*B*	
BNE	Branch Not Equal	Z=0		**0**		**0**			**1**		**0**	*B*	*B*		*B*			*B*	
BEQ	Branch Equal	Z=1		**0**		**0**			**1**		**1**	*B*	*B*		*B*			*B*	
BPL	Branch if plus	N=0		**1**		**0**			**0**		**0**	*B*	*B*		*B*			*B*	
BMI	Branch if minus	N=1		**1**		**0**			**0**		**1**	*B*	*B*		*B*			*B*	
BVC	Branch if Overflow Clear	V=0		**1**		**0**			**2**		**0**	*B*	*B*		*B*			*B*	
BVS	Branch if Overflow Set	V=1		**1**		**0**			**2**		**1**	*B*	*B*		*B*			*B*	
BHIS	Branch if higher or same	C=0		**1**		**0**			**3**		**0**	*B*	*B*		*B*			*B*	
BCC	Branch if carry clear	C=0		**1**		**0**			**3**		**0**	*B*	*B*		*B*			*B*	
BLO	Branch if lower	C=1		**1**		**0**			**3**		**1**	*B*	*B*		*B*			*B*	
BCS	Branch if carry set	C=1		**1**		**0**			**3**		**1**	*B*	*B*		*B*			*B*	
BGE	Branch if greater than or equal to	N xor V=0		**0**		**0**			**2**		**0**	*B*	*B*		*B*			*B*	
BLT	Branch if less than	N xor V=1		**0**		**0**			**2**		**1**	*B*	*B*		*B*			*B*	
BGT	Branch if greater than	not (N xor V)=0		**0**		**0**			**3**		**0**	*B*	*B*		*B*			*B*	
BLE	Branch on less than or equal to	not (N xor V)=1		**0**		**0**			**3**		**1**	*B*	*B*		*B*			*B*	
BHI	Branch on higher than	C not Z =0		**1**		**0**			**1**		**0**	*B*	*B*		*B*			*B*	
BLOS	Branch on lower than or same as	C not Z = 1		**1**		**0**			**1**		**1**	*B*	*B*		*B*			*B*	
JMP	Jump			**0**		**0**			**0**			**1**			*A*			*A*	
SOB	Subtract 1 and branch			**0**		**7**			**7**			*R*			*N*			*N*	
JSR r,Label	Jump to subroutine			**0**		**0**			**4**			*R*			*A*			*A*	
RTS r	Return from subroutine			**0**		**0**			**0**			**2**			**0**			*R*	
RTI	Return from interrupt/trap			**0**		**0**			**0**			**0**			**0**			**2**	
TRAP	Trap	T>=400		**1**		**0**			**4**			*T*			*T*			*T*	
BPT	Breakpoint trap			**0**		**0**			**0**			**0**			**0**			**3**	
IOT	I/O Trap			**0**		**0**			**0**			**0**			**0**			**4**	
EMT	Emulator Trap	T<=400		**1**		**0**			**4**			*T*			*T*			*T*	
RTT	Return from trace trap			**0**		**0**			**0**			**0**			**0**			**6**	
SPL	Set priority level			**0**		**0**			**0**			**2**			**3**			*N*	
-	Clear Multiple			**0**		**0**			**0**			**2**		**1**	**0**	*N*	*Z*	*V*	*C*
CLC	Clear Carry flag	C=0		**0**		**0**			**0**			**2**			**4**			**1**	
CLV	Clear Overflow flag	V=0		**0**		**0**			**0**			**2**			**4**			**2**	
CLZ	Clear Zero flag	Z=0		**0**		**0**			**0**			**2**			**4**			**4**	
CLN	Clear Negative flag	N=0		**0**		**0**			**0**			**2**			**5**			**0**	
CCC	Clear Condition codes	All=0		**0**		**0**			**0**			**2**			**5**			**7**	
-	Set Multiple			**0**		**0**			**0**			**2**		**1**	**1**	*N*	*Z*	*V*	*C*
SEC	Set Carry flag	C=1		**0**		**0**			**0**			**2**			**6**			**1**	
SEV	Set Overflow flag	V=1		**0**		**0**			**0**			**2**			**6**			**2**	
SEZ	Set Zero flag	Z=1		**0**		**0**			**0**			**2**			**6**			**4**	
SEN	Set Negative flag	N=1		**0**		**0**			**0**			**2**			**7**			**0**	
SCC	Set condition codes	All=1		**0**		**0**			**0**			**2**			**7**			**7**	
{B}=Byte (0=word / 1 = Byte)				1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1