INSTRUCTION SET REFERENCE

Transfer	
MOV	Move
LDA	Load
STA	Store
EXC	Exchange
CHN	Change

Shift/Rotate	
LSL	Logical shift left
LSR	Logical shift right
ASR	Arithmetic shift right
ROL	Rotate left
ROR	Rotate right

Logic	
AND	And
OR	Or
XOR	Exclusive or
CLR	Clear
SET	Set
COM	Complement
NEG	Negate

First Operands
Ri
Rii
V

Directives	
ORG	Origin
EQU	Equal
RMB	Reserve memory bytes
DAT	Data
END	End

Arithmetic	
ADD	Add
ADC	Add with carry
SUB	Subtract
SUE	Subtract with carry
MUL	Multiply
DIV	Divide
INC	Incremenet
DEC	Decrement

Second Operands
Rj
Rjj
V
VV
<Address>
<CD>
<SK+S>
<SK+S> + - R
<SK+CD+S>
<YG+S>

Operational	
DAA	Decimal adjust accumulator
PSH	Push
PUL	Pull
EIN	Enable interrupt
DIN	Disable interrupt
NOP	No operation
INT	Interrupt
RTS	Return from subroutine
RTI	Return from interrupt

Operand Symbols
V : Veri (8-bit data)
VV : 16-bit data
Ri, Rj : 8-bit register
Rii, Rjj : 16-bit register
S : Sıra (Index)
R: Range (incr/decr SK)

Branch - Compare	
CMP	Compare
BIT	Bit test
BRA	Branch (unconditional)
JMP	Jump (unconditional)
JMC	Jump conditionally
BEQ	Branch if equal
BNE	Branch if not equal
BGT	Branch if greater than
BGE	Branch if greater or equal
BLT	Branch if less than
BHI	Branch if higher
BHE	Branch if higher or equal

8-bit Registers
A, B, C, D
DK : Durum Kütüğü

16-bit Registers
AB, CD
SK : Sıralama Kütüğü (Index Register)
YG : Yığın Göstergesi (Stack Pointer)

Branch - Compare	
BLO	Branch if lower
BIO	Branch if overflow
BNO	Branch if not overflow
BIC	Branch if carry
BNC	Branch if not carry
BIH	Branch if half carry
BNH	Branch if not half carry
BSR	Branch to subroutine
JSR	Jump to subroutine
BSC	Branch to subroutine conditionally
JSC	Jump to subroutine conditionally
DBNZ	Decrease, branch if not zero

Status Flags (DK)
E : Carry
Y : Half carry
S : Zero
N : Negative
T : Overflow
K : Interrupt