8 Bit Registers	
Accumulator A	A
Accumulator B	B
AUX Register C	C
AUX Register D	D
Status Register	DK

16 Bit Registers	
Accumulator pair	AB
AUX register pair	CD
Index register	SK
Stack pointer	YG
Program counter	PS

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

K	T	S	N	Y	E

Addressing Methods	
Immediate	V
Immediate memory	Y
Register	L
Direct	D
Indirect	K
Relative	B
Indexed (SK)	S
Incremental SK	R
Decremental SK	Z
Indirect SK	U
Indexed (YG)	Y

Status Flags (DK)	
0	Becomes 0 after the execution.
1	Becomes 1 after the execution.
▬	No changes after the execution.
↕	Becomes 1 or 0 after the execution.

Initialization of PIA for Ready Input

D1	D0	Ready Input		Interrupt Output
0	0	From 1 to 0	D7 flag becomes 1	not genaretes interrupt
0	1	From 0 to 1	D7 flag becomes 1	not genaretes interrupt
1	0	From 1 to 0	D7 flag becomes 1	genaretes interrupt
1	1	From 0 to 1	D7 flag becomes 1	genaretes interrupt

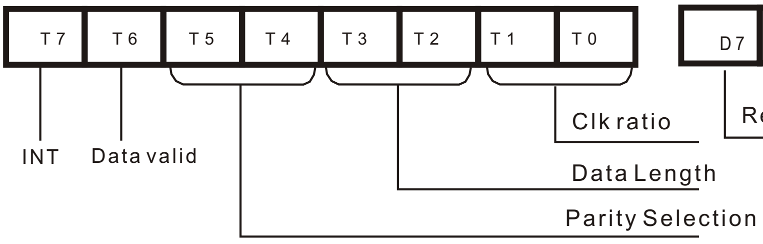
Initialization of PIA for ACK Input

D3	D2	ACK Input		Interrupt Output
0	0	From 1 to 0	D7 flag becomes 1	not genaretes interrupt
0	1	From 0 to 1	D7 flag becomes 1	not genaretes interrupt
1	0	From 1 to 0	D7 flag becomes 1	genaretes interrupt
1	1	From 0 to 1	D7 flag becomes 1	genaretes interrupt

Initialization of PIA for Data Valid output

D5	D4	Data Valid output
0	0	Becomes 0
0	1	Becomes 1
1	0	Becomes 0 after data writen to port.
1	1	Becomes 1 after data writen to port.

ASIA Control Register

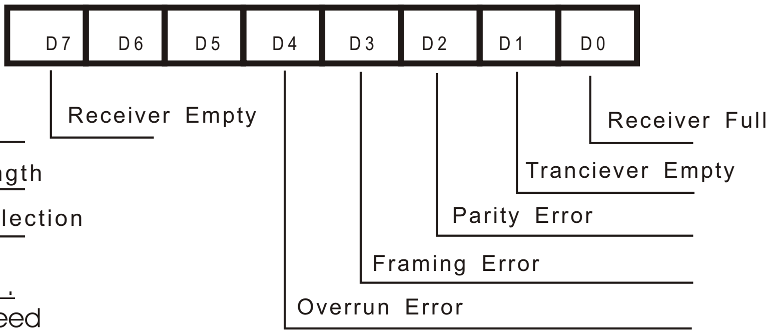


T5	T4	
0	0	no parity
0	1	odd parity
1	0	even parity

T1	T0	
0	0	1/1 speed
0	1	1/2 speed
1	0	1/4 speed
1	1	1/8 speed

T3	T2	
0	0	7 bit data 1 bit stop
0	1	7 bit data 2 bit stop
1	0	8 bit data 1 bit stop
1	1	8 bit data 2 bit stop

ASIA Status Register



ARITHMETIC INSTRUCTIONS -(8 bit)

Oper	Op Code	Adr met	Instruction Format				Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	T	S	N	Y	E		
ADD	Ai,V	V	00000011	00000 Ai	Data		↕	↕	↕	↕	↕	3	Ai ← Ai +V
	Ai,Ki	L	01000011	00 Ai Ki			↕	↕	↕	↕	↕	3	Ai ← Ai + Ki
	Ai,<adr>	D	00000011	00100 Ai	Adr (H)	Adr (L)	↕	↕	↕	↕	↕	4	Ai ← Ai + <Adr>
	Ai,<CD>	K	00000011	01000 Ai			↕	↕	↕	↕	↕	6	Ai ← Ai + <<CD>>
	Ai,<SK+S>	S	00000011	01100 Ai	S		↕	↕	↕	↕	↕	7	Ai ← Ai + <SK+S>
	Ai,<SK+S>+R	R	00000011	10000 Ai	S	R	↕	↕	↕	↕	↕	7	Ai ← Ai + <SK+S> +R
	Ai,<SK+S>-R	Z	00000011	10100 Ai	S	R	↕	↕	↕	↕	↕	7	Ai ← Ai + <SK+S> - R
	Ai,<SK+CD+S>	U	00000011	11000 Ai	S		↕	↕	↕	↕	↕	8	Ai ← Ai + <SK+CD+S>
	Ai,<YG+S>	Y	00000011	11100 Ai	S		↕	↕	↕	↕	↕	7	Ai ← Ai + <YG+S>
ADC	Ai,V	V	00000100	00000 Ai	Data		↕	↕	↕	↕	↕	3	Ai ← Ai +V + E
	Ai,Ki	L	01000100	00 Ai Ki			↕	↕	↕	↕	↕	3	Ai ← Ai + Ki + E
	Ai,<adr>	D	00000100	00100 Ai	Adr (H)	Adr (L)	↕	↕	↕	↕	↕	4	Ai ← Ai + <Adr> + E
	Ai,<CD>	K	00000100	01000 Ai			↕	↕	↕	↕	↕	6	Ai ← Ai + <<CD>> + E
	Ai,<SK+S>	S	00000100	01100 Ai	S		↕	↕	↕	↕	↕	7	Ai ← Ai + <SK+S> + E
	Ai,<SK+S>+R	R	00000100	10000 Ai	S	R	↕	↕	↕	↕	↕	7	Ai ← Ai + <SK+S> + E +R
	Ai,<SK+S>-R	Z	00000100	10100 Ai	S	R	↕	↕	↕	↕	↕	7	Ai ← Ai + <SK+S> + E - R
	Ai,<SK+CD+S>	U	00000100	11000 Ai	S		↕	↕	↕	↕	↕	8	Ai ← Ai + <SK+CD+S> + E
	Ai,<YG+S>	Y	00000100	11100 Ai	S		↕	↕	↕	↕	↕	7	Ai ← Ai + <YG+S> + E
SUB	Ai,V	V	00000101	00000 Ai	Data		↕	↕	↕	▬	↕	3	Ai ← Ai - V
	Ai,Ki	L	01000101	00 Ai Ki			↕	↕	↕	▬	↕	3	Ai ← Ai - Ki
	Ai,<adr>	D	00000101	00100 Ai	Adr (H)	Adr (L)	↕	↕	↕	▬	↕	4	Ai ← Ai - <Adr>
	Ai,<CD>	K	00000101	01000 Ai			↕	↕	↕	▬	↕	6	Ai ← Ai - <<CD>>
	Ai,<SK+S>	S	00000101	01100 Ai	S		↕	↕	↕	▬	↕	7	Ai ← Ai - <SK+S>
	Ai,<SK+S>+R	R	00000101	10000 Ai	S	R	↕	↕	↕	▬	↕	7	Ai ← Ai - <SK+S> +R
	Ai,<SK+S>-R	Z	00000101	10100 Ai	S	R	↕	↕	↕	▬	↕	7	Ai ← Ai - <SK+S> - R
	Ai,<SK+CD+S>	U	00000101	11000 Ai	S		↕	↕	↕	▬	↕	8	Ai ← Ai - <SK+CD+S>
	Ai,<YG+S>	Y	00000101	11100 Ai	S		↕	↕	↕	▬	↕	7	Ai ← Ai - <YG+S>
SUE	Ai,V	V	00000110	00000 Ai	Data		↕	↕	↕	▬	↕	3	Ai ← Ai -V - E
	Ai,Ki	L	01000110	00 Ai Ki			↕	↕	↕	▬	↕	3	Ai ← Ai - Ki - E
	Ai,<adr>	D	00000110	00100 Ai	Adr (Y)	Adr (L)	↕	↕	↕	▬	↕	4	Ai ← Ai - <Adr> - E
	Ai,<CD>	K	00000110	01000 Ai			↕	↕	↕	▬	↕	6	Ai ← Ai - <<CD>> - E
	Ai,<SK+S>	S	00000110	01100 Ai	S		↕	↕	↕	▬	↕	7	Ai ← Ai - <SK+S> - E
	Ai,<SK+S>+R	R	00000110	10000 Ai	S	R	↕	↕	↕	▬	↕	7	Ai ← Ai - <SK+S> - E +R
	Ai,<SK+S>-R	Z	00000110	10100 Ai	S	R	↕	↕	↕	▬	↕	7	Ai ← Ai - <SK+S> - E - R
	Ai,<SK+CD+S>	U	00000110	11000 Ai	S		↕	↕	↕	▬	↕	8	Ai ← Ai - <SK+CD+S> - E
	Ai,<YG+S>	Y	00000110	11100 Ai	S		↕	↕	↕	▬	↕	7	Ai ← Ai - <YG+S> - E

ARITHMETIC INSTRUCTIONS -(16 bit)

Oper	Op Code	Adr met	Instruction Format				Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	T	S	N	Y	E		
ADD	AB,VV	V	00100011	00000 AB	Data(H)	Data(L)	↕	↕	↕	▬	↕	4	AB← AB +VV
	AB,Kii	L	01100011	00 AB Kii			↕	↕	↕	▬	↕	4	AB← AB + Kii
	AB,<adr>	D	00100011	00100 AB	Adr (H)	Adr (L)	↕	↕	↕	▬	↕	5	AB← AB + (<Adr>+<Adr+1>)
	AB,<CD>	K	00100011	01000 AB			↕	↕	↕	▬	↕	7	AB← AB +(<<CD>>+<<CD+1>>)
	AB,<SK+S>	S	00100011	01100 AB	S		↕	↕	↕	▬	↕	8	AB← AB +(<SK+S>+<SK+S+1>)
	AB,<SK+S>+R	R	00100011	10000 AB	S	R	↕	↕	↕	▬	↕	8	AB← AB +(<SK+S>+<SK+S>) +R
	AB,<SK+S>-R	Z	00100011	10100 AB	S	R	↕	↕	↕	▬	↕	8	AB← AB +(<SK+S>+<SK+S+1>) - R
	AB,<SK+CD+S>	U	00100011	11000 AB	S		↕	↕	↕	▬	↕	9	AB← AB +(<SK+CD+S>+<SK+CD+S+1>
	AB,<YG+S>	Y	00100011	11100 AB	S		↕	↕	↕	▬	↕	8	AB← AB +(<YG+S>+<YG+S>)
SUB	AB,VV	V	00100101	00000 AB	Data (H)	Data (L)	↕	↕	↕	▬	↕	4	AB← AB - VV
	AB,Kii	L	01100101	00 AB Kii			↕	↕	↕	▬	↕	4	AB← AB - Kii
	AB,<adr>	D	00100101	00100 AB	Adr (H)	Adr (L)	↕	↕	↕	▬	↕	5	AB← AB - (<Adr>+<Adr+1>)
	AB,<CD>	K	00100101	01000 AB			↕	↕	↕	▬	↕	7	AB← AB -(<<CD>>+<<CD+1>>)
	AB,<SK+S>	S	00100101	01100 AB	S		↕	↕	↕	▬	↕	8	AB← AB -(<SK+S>+<SK+S+1>)
	AB,<SK+S>+R	R	00100101	10000 AB	S	R	↕	↕	↕	▬	↕	8	AB← AB -(<SK+S>+<SK+S>) +R
	AB,<SK+S>-R	Z	00100101	10100 AB	S	R	↕	↕	↕	▬	↕	8	AB← AB -(<SK+S>+>SK+S+1>) - R
	AB,<SK+CD+S>	U	00100101	11000 AB	S		↕	↕	↕	▬	↕	9	AB← AB -(<SK+CD+S>+<SK+CD+S+1>
	AB,<YG+S>	Y	00100101	11100 AB	S		↕	↕	↕	▬	↕	8	AB← AB -(<YG+S>+<YG+S>)
MUL	A,V	V	00000111	00000 A	Data		▬	↕	↕	▬	▬	24	AB← A * V
	A,Ki	L	01000111	00 A Ki			▬	↕	↕	▬	▬	24	AB← A * Ki
	A,<adr>	D	00000111	00100 A	Adr (H)	Adr (L)	▬	↕	↕	▬	▬	26	AB← A * <Adr>
	A,<CD>	K	00000111	01000 A			▬	↕	↕	▬	▬	28	AB← A * <<CD>>
	A,<SK+S>	S	00000111	01100 A	S		▬	↕	↕	▬	▬	30	AB← A * <SK+S>
	A,<SK+S>+R	R	00000111	10000 A	S	R	▬	↕	↕	▬	▬	31	AB← A * <SK+S> +R
	A,<SK+S>-R	Z	00000111	10100 A	S	R	▬	↕	↕	▬	▬	31	AB← A * <SK+S> - R
	A,<SK+CD+S>	U	00000111	11000 A	S		▬	↕	↕	▬	▬	32	AB← A * <SK+CD+S>
	A,<YG+S>	Y	00000111	11100 A	S		▬	↕	↕	▬	▬	30	AB← A * <YG+S>
DIV	AB,V	V	00100111	00000 AB	Data		▬	↕	↕	▬	▬	32	AB← AB / V
	AB,Ki	L	01100111	00000 AB			▬	↕	↕	▬	▬	32	AB← AB / Ki
	AB,<adr>	D	00100111	00100 AB	Adr (H)	Adr (L)	▬	↕	↕	▬	▬	34	AB← AB / <Adr>
	AB,<CD>	K	00100111	01000 AB			▬	↕	↕	▬	▬	36	AB← AB / <<CD>>
	AB,<SK+S>	S	00100111	01100 AB	S		▬	↕	↕	▬	▬	38	AB← AB / <SK+S>
	AB,<SK+S>+R	R	00100111	10000 AB	S	R	▬	↕	↕	▬	▬	38	AB← AB / <SK+S>
	AB,<SK+S>-R	Z	00100111	10100 AB	S	R	▬	↕	↕	▬	▬	38	AB← AB / <SK+S>
	AB,<SK+CD+S>	U	00100111	11000 AB	S		▬	↕	↕	▬	▬	40	AB← AB / <SK+CD+S>
	AB,<YG+S>	Y	00100111	11100 AB	S		▬	↕	↕	▬	▬	38	AB← AB / <YG+S>

TRANSFER INSTRUCTIONS -(8 bit)

Oper	Op Code	Adr met	Instruction Format					Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	T	S	N	Y	E		
MOV	Ki,Kj	L	01000000	00 Ki Kj				—	↕	↕	—	—	1	Ki ← Kj
EXC	Ki,Kj	L	01000001	00 Ki Kj				—	—	—	—	—	3	Ki← Kj
CHN	Ki	L	01000010	01 ▨▨▨ Ki				—	↕	↕	—	—	5	D3 D2 D1 D0 D7 D6 D5 D4
LDA	Ki,V	V	00000000	00000 Ki	Data			—	↕	↕	—	—	1	Ki ← V
	Ki,<adr>	D	00000000	00100 Ki	Adr (H)	Adr (L)		—	↕	↕	—	—	2	Ki ←<Adr>
	Ki,<CD>	K	00000000	01000 Ki				—	↕	↕	—	—	3	Ki ←<<CD>>
	Ki,<SK+S>	S	00000000	01100 Ki	S			—	↕	↕	—	—	4	Ki ←<SK+S>
	Ki,<SK+S>+R	R	00000000	10000 Ki	S	R		—	↕	↕	—	—	5	Ki ←<SK+S> +R
	Ki,<SK+S>-R	Z	00000000	10100 Ki	S	R		—	↕	↕	—	—	5	Ki ←<SK+S> -R
	Ki,<SK+CD+S>	U	00000000	11000 Ki	S			—	↕	↕	—	—	6	Ki ←<SK+CD+S>
	Ki,<YG+S>	Y	00000000	11100 Ki	S			—	↕	↕	—	—	5	Ki ←<YG+S>
STR	V,Adr	V	00000001	00001 ▨▨▨	Datai	Adr (H)	Adr (L)	—	—	—	—	—	3	Adr← V
	Ki,<adr>	D	00000001	00100 Ki	Adr (H)	Adr (L)		—	—	—	—	—	2	Adr← Ki
	Ki,<CD>	K	00000001	01000 Ki				—	—	—	—	—	3	<<CD>> ← Ki
	Ki,<SK+S>	S	00000001	01100 Ki	S			—	—	—	—	—	4	<SK+S> ← Ki
	Ki,<SK+S>+R	R	00000001	10000 Ki	S	R		—	—	—	—	—	5	<SK+S> ← Ki + R
	Ki,<SK+S>-R	Z	00000001	10100 Ki	S	R		—	—	—	—	—	5	<SK+S> ← Ki - R
	Ki,<SK+CD+S>	U	00000001	11000 Ki	S			—	—	—	—	—	6	<SK+CD+S> ← Ki
	Ki,<YG+S>	Y	00000001	11100 Ki	S			—	—	—	—	—	5	<YG+S> ← Ki

TRANSFER INSTRUCTIONS -(16 bit)

Oper	Op Code	Adr met	Instruction Format				Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	T	S	N	Y	E		
MOV	Kii,Kjj	L	01100000	00 Kii Kjj			—	↕	↕	—	—	2	Kii ← Kjj
EXC	Kii,Kjj	L	01100001	00 Kii Kjj			—	—	—	—	—	4	Kii ← Kjj
LDA	Kii,VV	V	00100000	00000 Kii	Datai	Data	—	↕	↕	—	—	2	Kii ← VV
	Kii,<adr>	D	00100000	00100 Kii	Adr (H)	Adr (L)	—	↕	↕	—	—	3	Kii ← <Adr>+<Adr+1>
	Kii,<CD>	K	00100000	01000 Kii			—	↕	↕	—	—	4	Kii ← <<CD>>+<<CD+1>>
	Kii,<SK+S>	S	00100000	01100 Kii	S		—	↕	↕	—	—	5	Kii ← <SK+S>+<SK+S+1>
	Kii,<SK+S>+R	R	00100000	10000 Kii	S	R	—	↕	↕	—	—	6	Kii ← <SK+S>+<SK+S+1> +R
	Kii,<SK+S>-R	Z	00100000	10100 Kii	S	R	—	↕	↕	—	—	6	Kii ← <SK+S>+<SK+S+1> - R
	Kii,<SK+CD+S>	U	00100000	11000 Kii	S		—	↕	↕	—	—	7	Kii ← <SK+CD+S>+<SK+CD+S+1>
	Kii,<YG+S>	Y	00100000	11100 Kii	S		—	↕	↕	—	—	6	Kii ← <YG+S>+<YG+S+1>
STR	Kii,<adr>	D	00100001	00100 Kii	Adr (H)	Adr (L)	—	—	—	—	—	3	Adr+(Adr+1) ← Kii
	Kii,<CD>	K	00100001	01000 Kii			—	—	—	—	—	4	<<CD>>+<<CD+1>> ← Kii
	Kii,<SK+S>	S	00100001	01100 Kii	S		—	—	—	—	—	5	<SK+S>+<SK+S+1> ← Kii
	Kii,<SK+S>+R	R	00100001	10000 Kii	S	R	—	—	—	—	—	6	<SK+S>+<SK+S+1> ← Kii + R
	Kii,<SK+S>-R	Z	00100001	10100 Kii	S	R	—	—	—	—	—	6	<SK+S>+<SK+S+1> ← Kii - R
	Kii,<SK+CD+S>	U	00100001	11000 Kii	S		—	—	—	—	—	7	<SK+CD+S>+<SK+CD+S+1>← Kii
	Kii,<YG+S>	Y	00100001	11100 Kii	S		—	—	—	—	—	6	<YG+S>+<YG+S+1>← Kii

LOGIC INSTRUCTIONS -(8 bit)

Oper	Op Code	Adr met	Instruction Format				Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	T	S	N	Y	E		
AND	Ai,V	V	00001000	00000 Ai	Data		—	↕	↕	—	—	3	Ai← Ai • V
	Ai,Ki	L	01001000	00 Ai Ki			—	↕	↕	—	—	3	Ai ← Ai • Ki
	Ai,<adr>	D	00001000	00100 Ai	Adr (H)	Adr (L)	—	↕	↕	—	—	4	Ai ← Ai • <Adr>
	Ai,<CD>	K	00001000	01000 Ai			—	↕	↕	—	—	6	Ai ← Ai • <<CD>>
	Ai,<SK+S>	S	00001000	01100 Ai	S		—	↕	↕	—	—	7	Ai ← Ai • <SK+S>
	Ai,<SK+S>+R	R	00001000	10000 Ai	S	R	—	↕	↕	—	—	7	Ai ← Ai • <SK+S>+R
	Ai,<SK+S>-R	Z	00001000	10100 Ai	S	R	—	↕	↕	—	—	7	Ai ← Ai • <SK+S> - R
	Ai,<SK+CD+S>	U	00001000	11000 Ai	S		—	↕	↕	—	—	8	Ai ← Ai • <SK+CD+S>
	Ai,<YG+S>	Y	00001000	11100 Ai	S		—	↕	↕	—	—	7	Ai ← Ai • <YG+S>
OR	Ai,V	V	00001001	00000 Ai	Data		—	↕	↕	—	—	3	Ai← Ai + V
	Ai,Ki	L	01001001	00 Ai Ki			—	↕	↕	—	—	3	Ai ← Ai + Ki
	Ai,<adr>	D	00001001	00100 Ai	Adr (H)	Adr (L)	—	↕	↕	—	—	4	Ai ← Ai + <Adr>
	Ai,<CD>	K	00001001	01000 Ai			—	↕	↕	—	—	6	Ai ← Ai + <<CD>>
	Ai,<SK+S>	S	00001001	01100 Ai	S		—	↕	↕	—	—	7	Ai ← Ai + <SK+S>
	Ai,<SK+S>+R	R	00001001	10000 Ai	S	R	—	↕	↕	—	—	7	Ai ← Ai + <SK+S>+R
	Ai,<SK+S>-R	Z	00001001	10100 Ai	S	R	—	↕	↕	—	—	7	Ai ← Ai + <SK+S> - R
	Ai,<SK+CD+S>	U	00001001	11000 Ai	S		—	↕	↕	—	—	8	Ai ← Ai + <SK+CD+S>
	Ai,<YG+S>	Y	00001001	11100 Ai	S		—	↕	↕	—	—	7	Ai ← Ai + <YG+S>
XOR	Ai,V	V	00001010	00000 Ai	Data		—	↕	↕	—	—	3	Ai← Ai ⊕ V
	Ai,Ki	L	01001010	00 Ai Ki			—	↕	↕	—	—	3	Ai ← Ai ⊕ Ki
	Ai,<adr>	D	00001010	00100 Ai	Adr (H)	Adr (L)	—	↕	↕	—	—	4	Ai ← Ai ⊕ <Adr>
	Ai,<CD>	K	00001010	01000 Ai			—	↕	↕	—	—	6	Ai ← Ai ⊕ <<CD>>
	Ai,<SK+S>	S	00001010	01100 Ai	S		—	↕	↕	—	—	7	Ai ← Ai ⊕ <SK+S>
	Ai,<SK+S>+R	R	00001010	10000 Ai	S	R	—	↕	↕	—	—	7	Ai ← Ai ⊕ <SK+S>+R
	Ai,<SK+S>-R	Z	00001010	10100 Ai	S	R	—	↕	↕	—	—	7	Ai ← Ai ⊕ <SK+S> - R
	Ai,<SK+CD+S>	U	00001010	11000 Ai	S		—	↕	↕	—	—	8	Ai ← Ai ⊕ <SK+CD+S>
	Ai,<YG+S>	Y	00001010	11100 Ai	S		—	↕	↕	—	—	7	Ai ← Ai ⊕ <YG+S>

OPERATION - I

Oper	Op Code	Adr met	Instruction Format				Status Flags					A	Explanation
			1. Byte	2. Byte	3. Bytei	4. Byte	T	S	N	Y	E		
CLR	Ki	L	01001011	01▨▨▨ Ki			0	1	0	0	0	3	Ki ← 0
	<Adr>	D	00001011	00101▨▨▨	Adr (Yük)	Adr (Düş)	0	1	0	0	0	4	<Adr> ← 0
	<CD>	K	00001011	01001▨▨▨			0	1	0	0	0	6	<<CD>> ← 0
	<SK+S>	S	00001011	01101▨▨▨	S		0	1	0	0	0	7	<SK+S> ← 0
	<SK+S>+R	R	00001011	10001▨▨▨	S	R	0	1	0	0	0	7	<SK+S> ← 0, + R
	<SK+S>-R	Z	00001011	10101▨▨▨	S	R	0	1	0	0	0	7	<SK+S> ← 0, - R
	<SK+CD+S>	U	00001011	11001▨▨▨	S		0	1	0	0	0	8	<SK+CD+S> ← 0
	<YG+S>	Y	00001011	11101▨▨▨	S		0	1	0	0	0	7	<YG+S> ← 0
INC	Ki	L	01010000	01▨▨▨ Ki			↕	↕	↕	—	↕	3	Ki ← Ki + 1
	<Adr>	D	00010000	00101▨▨▨	Adr (Yük)	Adr (Düş)	↕	↕	↕	—	↕	4	<Adr> ← <Adr> + 1
	<CD>	K	00010000	01001▨▨▨			↕	↕	↕	—	↕	6	<<CD>> ← <<CD>> + 1
	<SK+S>	S	00010000	01101▨▨▨	S		↕	↕	↕	—	↕	7	<SK+S> ← <SK+S> + 1
	<SK+S>+R	R	00010000	10001▨▨▨	S	R	↕	↕	↕	—	↕	7	<SK+S> ← <SK+S> + 1, + R
	<SK+S>-R	Z	00010000	10101▨▨▨	S	R	↕	↕	↕	—	↕	7	<SK+S> ← <SK+S> + 1, - R
	<SK+CD+S>	U	00010000	11001▨▨▨	S		↕	↕	↕	—	↕	8	<SK+CD+S> ← <SK+CD+S> + 1
	<YG+S>	Y	00010000	11101▨▨▨	S		↕	↕	↕	—	↕	7	<YG+S> ← <YG+S> + 1
DEC	Ki	L	01010001	01▨▨▨ Ki			↕	↕	↕	—	↕	3	Ki ← Ki - 1
	<Adr>	D	00010001	00101▨▨▨	Adr (Yük)	Adr (Düş)	↕	↕	↕	—	↕	4	<Adr> ← <Adr> - 1
	<CD>	K	00010001	01001▨▨▨			↕	↕	↕	—	↕	6	<<CD>> ← <<CD>> - 1
	<SK+S>	S	00010001	01101▨▨▨	S		↕	↕	↕	—	↕	7	<SK+S> ← <SK+S> - 1
	<SK+S>+R	R	00010001	10001▨▨▨	S	R	↕	↕	↕	—	↕	7	<SK+S> ← <SK+S> - 1, + R
	<SK+S>-R	Z	00010001	10101▨▨▨	S	R	↕	↕	↕	—	↕	7	<SK+S> ← <SK+S> - 1, -R
	<SK+CD+S>	U	00010001	11001▨▨▨	S		↕	↕	↕	—	↕	8	<SK+CD+S> ← <SK+CD+S> - 1
	<YG+S>	Y	00010001	11101▨▨▨	S		↕	↕	↕	—	↕	7	<YG+S> ← <YG+S> - 1
COM	Ki	L	01010010	01▨▨▨ Ki			—	↕	↕	—	—	3	Ki ← com<Ki>
	<Adr>	D	00010010	00101▨▨▨	Adr (Yük)	Adr (Düş)	—	↕	↕	—	—	4	<Adr> ← com<Adr>
	<CD>	K	00010010	01001▨▨▨			—	↕	↕	—	—	6	<<CD>> ← com<<CD>>
	<SK+S>	S	00010010	01101▨▨▨	S		—	↕	↕	—	—	7	<SK+S> ← com<SK+S>
	<SK+S>+R	R	00010010	10001▨▨▨	S	R	—	↕	↕	—	—	7	<SK+S> ← com<SK+S>, + R
	<SK+S>-R	Z	00010010	10101▨▨▨	S	R	—	↕	↕	—	—	7	<SK+S> ← com<SK+S>, - R
	<SK+CD+S>	U	00010010	11001▨▨▨	S		—	↕	↕	—	—	8	<SK+CD+S> ← com<SK+CD+S>
	<YG+S>	Y	00010010	11101▨▨▨	S		—	↕	↕	—	—	7	<YG+S> ← com<YG+S>

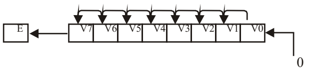
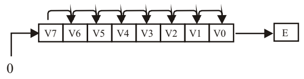
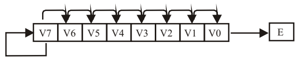
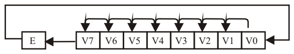
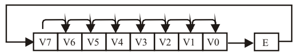
OPERATION - II

Oper	Op Code	Adr met	Instruction Format				Status Flags					A	Explanation
			1. Byte	2. Byte	3. Bytei	4. Byte	T	S	N	Y	E		
NEG	Ki	L	01010011	01▨▨▨ Ki			—	↕	↕	—	—	3	Ki ← neg<Ki>
	<Adr>	D	00010011	00101▨▨▨	Adr (Yük)	Adr (Düş)	—	↕	↕	—	—	4	<Adr> ← neg<Adr>
	<CD>	K	00010011	01001▨▨▨			—	↕	↕	—	—	6	<<CD>> ← neg<CD>>
	<SK+S>	S	00010011	01101▨▨▨	S		—	↕	↕	—	—	7	<SK+S> ← neg<SK+S>
	<SK+S>+R	R	00010011	10001▨▨▨	S	R	—	↕	↕	—	—	7	<SK+S> ← neg<SK+S>, + R
	<SK+S>-R	Z	00010011	10101▨▨▨	S	R	—	↕	↕	—	—	7	<SK+S> ← neg<SK+S>, - R
	<SK+CD+S>	U	00010011	11001▨▨▨	S		—	↕	↕	—	—	8	<SK+CD+S> ← neg<SK+CD+S>
	<YG+S>	Y	00010011	11101▨▨▨	S		—	↕	↕	—	—	7	<YG+S> ← neg<YG+S>
CLR	N,Ki	L	01001101	11 N Ki			—	↕	↕	—	—	3	Ki ← Ki(N=0)
	N,<adr>	D	00001101	00110 N	Adr (Yük)	Adr (Düş)	—	↕	↕	—	—	4	<Adr> ← <Adr,N=0>
	N,<CD>	K	00001101	01010 N			—	↕	↕	—	—	6	<<CD>> ← <CD,N=0>>
	N,<SK+S>	S	00001101	01110 N	S		—	↕	↕	—	—	7	<SK+S> ← <SK+S,N=0>
	N,<SK+S>+R	R	00001101	10010 N	S	R	—	↕	↕	—	—	7	<SK+S> ← <SK+S,N=0>, + R
	N,<SK+S>-R	Z	00001101	10110 N	S	R	—	↕	↕	—	—	7	<SK+S> ← <SK+S,N=0>, - R
	N,<SK+CD+S>	U	00001101	11010 N	S		—	↕	↕	—	—	8	<SK+CD+S> ← <SK+CD+S,N=0>
	N,<YG+S>	Y	00001101	11110 N	S		—	↕	↕	—	—	7	<YG+S> ← <YG+S,N=0>
SET	N,Ki	L	01001111	11 N Ki			—	↕	↕	—	—	3	Ki ← Ki(N=1)
	N,<adr>	D	00001111	00110 N	Adr (Yük)	Adr (Düş)	—	↕	↕	—	—	4	<Adr> ← <Adr,N=1>
	N,<CD>	K	00001111	01010 N			—	↕	↕	—	—	6	<<CD>> ← <CD,N=1>>
	N,<SK+S>	S	00001111	01110 N	S		—	↕	↕	—	—	7	<SK+S> ← <SK+S,N=1>
	N,<SK+S>+R	R	00001111	10010 N	S	R	—	↕	↕	—	—	7	<SK+S> ← <SK+S,N=1>, + R
	N,<SK+S>-R	Z	00001111	10110 N	S	R	—	↕	↕	—	—	7	<SK+S> ← <SK+S,N=1>, - R
	N,<SK+CD+S>	U	00001111	11010 N	S		—	↕	↕	—	—	8	<SK+CD+S> ← <SK+CD+S,N=1>
	N,<YG+S>	Y	00001111	11110 N	S		—	↕	↕	—	—	7	<YG+S> ← <YG+S,N=1>

OPERATION INSTRUCTIONS

Oper	Op Code	Adr met	Instruction Format		Status Reg.					A	Explanation
			1. Byte	2. Byte	T	S	N	Y	E		
CLR	E	L	01001100	10▨▨▨000	▬	▬	▬	▬	0	1	E ← 0
	Y	L	01001100	10▨▨▨001	▬	▬	▬	0	▬	1	Y ← 0
	N	L	01001100	10▨▨▨010	▬	▬	0	▬	▬	1	N ← 0
	S	L	01001100	10▨▨▨011	▬	0	▬	▬	▬	1	S ← 0
	T	L	01001100	10▨▨▨100	0	▬	▬	▬	▬	1	T ← 1
SET	E	L	01001110	10▨▨▨000	▬	▬	▬	▬	1	1	E ← 1
	Y	L	01001110	10▨▨▨001	▬	▬	▬	1	▬	1	Y ← 1
	N	L	01001110	10▨▨▨010	▬	▬	1	▬	▬	1	N ← 1
	S	L	01001110	10▨▨▨011	▬	1	▬	▬	▬	1	S ← 1
	T	L	01001110	10▨▨▨100	1	▬	▬	▬	▬	1	T ← 1
INC	Kii	L	01110000	01▨▨▨Kii	↕	↕	↕	▬	↕	2	Kii ← Kii + 1
DEC	Kii	L	01110001	01▨▨▨Kii	↕	↕	↕	▬	↕	2	Kii ← Kii - 1
DAA	Ai	L	01010100	01▨▨▨Ai	▬	▬	▬	0	0	2	Binary to Decimal
PSH	Ai	L	01010101	01▨▨▨Ai	▬	▬	▬	▬	▬	2	Push Ai to Stack
PUL	Ai	L	01010110	01▨▨▨Ai	▬	↕	↕	▬	▬	2	Pull Stock to Ai
EIN		L	11000000		▬	▬	▬	▬	▬	1	Enable Interrupt
DIN		L	11000001		▬	▬	▬	▬	▬	1	Disable Interrupt
NOP		L	11000010		▬	▬	▬	▬	▬	1	No Operation
RTS		L	11000100		▬	▬	▬	▬	▬	5	Return fron subroutine
RTI		L	11000101		▬	▬	▬	▬	▬	15	Return from interrupt
INT		L	11000011		▬	▬	▬	▬	▬	15	Software Interrupt

SHIFT & ROTATE INSTRUCTIONS

Oper	Op Code	Adr met	Instruction Format				Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	T	S	N	Y	E		
LSL	Ki	L	01010111	01▨▨▨Ki			▬	↕	↕	▬	↕	1	
	<Adr>	D	00010111	00101▨▨▨	Adr (H)	Adr (L)	▬	↕	↕	▬	↕	2	
	<CD>	K	00010111	01001▨▨▨			▬	↕	↕	▬	↕	3	
	<SK+S>	S	00010111	01101▨▨▨	S		▬	↕	↕	▬	↕	4	
	<SK+S>+R	R	00010111	10001▨▨▨	S	R	▬	↕	↕	▬	↕	5	
	<SK+S>-R	Z	00010111	10101▨▨▨	S	R	▬	↕	↕	▬	↕	5	
	<SK+CD+S>	U	00010111	11001▨▨▨	S		▬	↕	↕	▬	↕	6	
	<YG+S>	Y	00010111	11101▨▨▨	S		▬	↕	↕	▬	↕	4	
LSR	Ki	L	01011000	01▨▨▨Ki			▬	↕	↕	▬	↕	1	
	<Adr>	D	00011000	00101▨▨▨	Adr (H)	Adr (L)	▬	↕	↕	▬	↕	2	
	<CD>	K	00011000	01001▨▨▨			▬	↕	↕	▬	↕	3	
	<SK+S>	S	00011000	01101▨▨▨	S		▬	↕	↕	▬	↕	4	
	<SK+S>+R	R	00011000	10001▨▨▨	S	R	▬	↕	↕	▬	↕	5	
	<SK+S>-R	Z	00011000	10101▨▨▨	S	R	▬	↕	↕	▬	↕	5	
	<SK+CD+S>	U	00011000	11001▨▨▨	S		▬	↕	↕	▬	↕	6	
	<YG+S>	Y	00011000	11101▨▨▨	S		▬	↕	↕	▬	↕	4	
ASR	Ki	L	01011001	01▨▨▨Ki			▬	↕	↕	▬	↕	1	
	<Adr>	D	00011001	00101▨▨▨	Adr (H)	Adr (L)	▬	↕	↕	▬	↕	2	
	<CD>	K	00011001	01001▨▨▨			▬	↕	↕	▬	↕	3	
	<SK+S>	S	00011001	01101▨▨▨	S		▬	↕	↕	▬	↕	4	
	<SK+S>+R	R	00011001	10001▨▨▨	S	R	▬	↕	↕	▬	↕	5	
	<SK+S>-R	Z	00011001	10101▨▨▨	S	R	▬	↕	↕	▬	↕	5	
	<SK+CD+S>	U	00011001	11001▨▨▨	S		▬	↕	↕	▬	↕	6	
	<YG+S>	Y	00011001	11101▨▨▨	S		▬	↕	↕	▬	↕	4	
ROL	Ki	L	01011010	01▨▨▨Ki			▬	↕	↕	▬	▬	1	
	<Adr>	D	00011010	00101▨▨▨	Adr (H)	Adr (L)	▬	↕	↕	▬	▬	2	
	<CD>	K	00011010	01001▨▨▨			▬	↕	↕	▬	▬	3	
	<SK+S>	S	00011010	01101▨▨▨	S		▬	↕	↕	▬	▬	4	
	<SK+S>+R	R	00011010	10001▨▨▨	S	R	▬	↕	↕	▬	▬	5	
	<SK+S>-R	Z	00011010	10101▨▨▨	S	R	▬	↕	↕	▬	▬	5	
	<SK+CD+S>	U	00011010	11001▨▨▨	S		▬	↕	↕	▬	▬	6	
	<YG+S>	Y	00011010	11101▨▨▨	S		▬	↕	↕	▬	▬	4	
ROR	Ki	L	01011011	01▨▨▨Ki			▬	↕	↕	▬	▬	1	
	<Adr>	D	00011011	00101▨▨▨	Adr (H)	Adr (L)	▬	↕	↕	▬	▬	2	
	<CD>	K	00011011	01001▨▨▨			▬	↕	↕	▬	▬	3	
	<SK+S>	S	00011011	01101▨▨▨	S		▬	↕	↕	▬	▬	4	
	<SK+S>+R	R	00011011	10001▨▨▨	S	R	▬	↕	↕	▬	▬	5	
	<SK+S>-R	Z	00011011	10101▨▨▨	S	R	▬	↕	↕	▬	▬	5	
	<SK+CD+S>	U	00011011	11001▨▨▨	S		▬	↕	↕	▬	▬	6	
	<YG+S>	Y	00011011	11101▨▨▨	S		▬	↕	↕	▬	▬	4	

COMPARE INSTRUCTIONS

Oper	Op Code	Adr met	Instruction Format				Status Reg.					A	Explanation
			1. Byte	2. Byte	3. Byte	4. Byte	T	S	N	Y	E		
CMP	Ki,V	V	00011100	00000 Ki	Data		⬍	⬍	⬍	⬍	⬍	2	Ki - V
	Ki,Kj	L	01011100	00 Ki Kj			⬍	⬍	⬍	⬍	⬍	2	Ki - Kj
	Ki,<adr>	D	00011100	00100 Ki	Adr (H)	Adr (L)	⬍	⬍	⬍	⬍	⬍	3	Ki - <Adr>
	Ki,<CD>	K	00011100	01000 Ki			⬍	⬍	⬍	⬍	⬍	3	Ki - <<CD>>
	Ki,<SK+S>	S	00011100	01100 Ki	S		⬍	⬍	⬍	⬍	⬍	4	Ki - <SK+S>
	Ki,<SK+S>+R	R	00011100	10000 Ki	S	R	⬍	⬍	⬍	⬍	⬍	5	Ki - <SK+S>+R
	Ki,<SK+S>-R	Z	00011100	10100 Ki	S	R	⬍	⬍	⬍	⬍	⬍	5	Ki - <SK+S> - R
	Ki,<SK+CD+S>	U	00011100	11000 Ki	S		⬍	⬍	⬍	⬍	⬍	6	Ki - <SK+CD+S>
	Ki,<YG+S>	Y	00011100	11100 Ki	S		⬍	⬍	⬍	⬍	⬍	5	Ki - <YG+S>
CMP	Kii,VV	V	00111100	00000 Kii	Data (H)	Data (L)	⬍	⬍	⬍	▬	⬍	4	Kii - VV
	Kii,Kjj	L	01111100	00 Kii Kjj			⬍	⬍	⬍	▬	⬍	4	Kii - Kjj
	Kii,<adr>	D	00111100	00100 Kii	Adr (H)	Adr (L)	⬍	⬍	⬍	▬	⬍	5	Kii - (<Adr>+<Adr+1>)
	Kii,<CD>	K	00111100	01000 Kii			⬍	⬍	⬍	▬	⬍	5	Kii - (<<CD>>+<<CD+1>>)
	Kii,<SK+S>	S	00111100	01100 Kii	S		⬍	⬍	⬍	▬	⬍	6	Kii - (<SK+S>+<SK+S+1>)
	Kii,<SK+S>+R	R	00111100	10000 Kii	S	R	⬍	⬍	⬍	▬	⬍	7	Kii - (<SK+S>+<SK+S>) +R
	Kii,<SK+S>-R	Z	00111100	10100 Kii	S	R	⬍	⬍	⬍	▬	⬍	7	Kii - (<SK+S>+>SK+S+1>) - R
	Kii,<SK+CD+S>	U	00111100	11000 Kii	S		⬍	⬍	⬍	▬	⬍	8	Kii - (<SK+CD+S>+<SK+CD+S+1>
	Kii,<YG+S>	Y	00111100	11100 Kii	S		⬍	⬍	⬍	▬	⬍	7	Kii - (<YG+S>+<YG+S>)
BIT	Ki,V	V	00011101	00000 Ki	Data		▬	⬍	⬍	▬	▬	2	Ki • V
	Ki,Kj	L	01011101	00 Ki Kj			▬	⬍	⬍	▬	▬	2	Ki • Ki
	Ki,<adr>	D	00011101	00100 Ki	Adr (H)	Adr (L)	▬	⬍	⬍	▬	▬	3	Ki •<Adr>
	Ki,<CD>	K	00011101	01000 Ki			▬	⬍	⬍	▬	▬	3	Ki •<<CD>>
	Ki,<SK+S>	S	00011101	01100 Ki	S		▬	⬍	⬍	▬	▬	4	Ki •<SK+S>
	Ki,<SK+S>+R	R	00011101	10000 Ki	S	R	▬	⬍	⬍	▬	▬	5	Ki •<SK+S> +R
	Ki,<SK+S>-R	Z	00011101	10100 Ki	S	R	▬	⬍	⬍	▬	▬	5	Ki •<SK+S> - R
	Ki,<SK+CD+S>	U	00011101	11000 Ki	S		▬	⬍	⬍	▬	▬	6	Ki •<SK+CD+S>
	Ki,<YG+S>	Y	00011101	11100 Ki	S		▬	⬍	⬍	▬	▬	5	Ki • <YG+S>

JUMP & BRANCH INSTRUCTIONS

Op Code	Adr met	Instruction Format				A	Explanation
		1. Byte	2. Byte	3. Byte	4. Byte		
BRA V	B	10000000	Step count			2	Branch Always (V step)
JMP Adr	D	00011110	00101▨▨▨	Adr (H)	Adr (L)	2	Jump Always (To address)
JMC S,Adr	D	00011111	00111011	Adr (H)	Adr (L)	3	S=1 => jump to address
JMC N,Adr	D	00011111	00111010	Adr (H)	Adr (L)	3	N=1 => jump to address
JMC E,Adr	D	00011111	00111000	Adr (H)	Adr (L)	3	E=1 => jump to address
JMC T,Adr	D	00011111	00111100	Adr (H)	Adr (L)	3	T=1 => jump to address
BEQ	B	10000001	Step count			2	Branch if equal (V step)
BNE	B	10000010	Step count			2	Branch if not equal (V step)
BGT V	B	10000011	Step count			2	Branch if greater (V step)
BGE V	B	10000100	Step count			2	Branch if greater or equal
BLS V	B	10000101	Step count			2	Branch if less than
BHI V	B	10000110	Step count			2	Branch if higher
BHE V	B	10000111	Step count			2	Branch if higher or equal
BLO V	B	10001000	Step count			2	Branch if lower
BIO V	B	10001001	Step count			2	T=1 => jump V step
BNO V	B	10001010	Step count			2	T=0 => jump V step
BIC V	B	10001011	Step count			2	E=1 => jump V step
BNC V	B	10001100	Step count			2	E=0 => jump V step
BIH V	B	10001101	Step count			2	Y=1 => jump V step
BNH V	B	10001110	Step count			2	Y=0 => jump V step
BSR V	B	10001111	Step count			2	Branch to subprogram (V step)
JSR Adr	D	00010100	00101▨▨▨	Adr (H)	Adr (L)	5	Branch to subprogram (Address)
BSC S,V	B	10010011	Step count			6	S=1 => jump to subprogram (V step)
BSC N,V	B	10010010	Step count			6	N=1 => jump to subprogram (V step)
BSC E,V	B	10010000	Step count			6	E=1 => jump to subprogram (V step)
BSC T,V	B	10010100	Step count			6	T=1 => jump to subprogram (V step)
BSC S,Adr	D	00010101	00111011	Adr (H)	Adr (L)	6	S=1 => jump to subprogram (Address)
BSC N,Adr	D	00010101	00111010	Adr (H)	Adr (L)	6	N=1 => jump to subprogram (Address)
BSC E,Adr	D	00010101	00111000	Adr (H)	Adr (L)	6	E=1 => jump to subprogram (Address)
BSC T,Adr	D	00010101	00111100	Adr (H)	Adr (L)	6	T=1 => jump to subprogram (Address)
DBNZ Ki,V	B	11000110	01▨▨▨ Ki	Step count		8	Decrease Ki, branch if not zero (V step)
DBNZ <Adr>,V	B	11000111	Step count	Adr (H)	Adr (L)	9	Decrease M[adr], branch if not zero (V step)