SC-61860 Instruction Set and Register Description

Compiled by Leonidas Tolias

Register Summary

Register	Address	Common Use	Processor Summary
I	0x00	Length of Block Operations	The SC-61860 has 96 bytes of internal Random Access Memory used for Registers and the stack. The last four bytes are used for I/O. The 61680 also has a 16 bit Program Counter (PC), and a 16 bit Data Pointer (DP). The Arithmetic Logic Unit has a carry flag (c) and a zero flag (z). There is an internal register D which cannot be accessed by machine instructions. It is filled from I or J during block operations. There are also three 7 bit registers used to address the internal RAM: the stack pointer R, and P and Q which point to the starting addresses of block operations.
J	0x01	Length of Block Operations	
A	0x02	Accumulator	
B	0x03	Accumulator	
XI	0x04	Low Byte of 16 bit Pointer for Read Operations	
Xh	0x05	High Byte of 16 bit Pointer for Read Operations	
YI	0x06	Low Byte of 16 bit Pointer for Write Operations	
Yh	0x07	High Byte of 16 bit Pointer for Write Operations	
K - N	0x08 - 0x0B	General Purpose registers	
Stack	0x0C - 0x5B	Stack	
IA	0x5C	Inport A	
IB	0x5D	Inport B	
FO	0x5E	Outport F	
COUT	0x5F	Control Port	

CPU Control

Mnemonic		Description	Operation	Hex	Flag	Cycle	Bytes
NOPW	N/A	No Operation	none	0x4d	N/A	2	1
NOPT	N/A	No Operation	none	0xce	N/A	3	1
WAIT	n	Wait 6 + n cycles	none	0x4e	N/A	6+n	2
WAITI	N/A	Wait 5 + 4 * I	none	0x4f	N/A	5+4*I	1
CUP	N/A	Synonym for WAITJ	comp of CDN?	0x4f	Z	???	1
SC	N/A	Set C and Z	1 -> C; 1 -> Z	0xd0	C,Z	2	1
RC	N/A	Clear C and Set Z	0 -> C; 1 -> Z	0xd1	C,Z	2	1
PUSH	N/A	Push A onto the Stack	R - 1 --> R, A --> (R)	0x34	N/A	3	1
POP	N/A	Pop top of Stack into A	(R) --> A, R + 1 --> R	0x5b	N/A	2	1
LEAVE	N/A	Clear top of Stack	0 --> (R)	0xd8	N/A	2	1
CALL	nm	Calls routine at adress	push PC; PC = nm	0x78,n,m	N/A	8	3
CAL	nm	nm <= 0x1fff, i byte less than CALL	push PC; PC = nm	0xe0+n,m	N/A	7	3
RTN	N/A	Return to call statement (ends routine)	[R,R+1] -> PC; R += 2	0x37	N/A	4	1

Jumps

Mnemonic		Description	Operation	Hex	Flag	Cycle	Bytes
JP	nm	Jump to Address	nm -> PC	0x79,n,m	N/A	6	3
JPZ	nm	Jump to Address if Z	if C=0 nm -> PC else PC += 3	0x7e,n,m	N/A	6	3
JPNZ	nm	Jump to Address if not Z	if C=0 nm -> PC else PC += 3	0x7c,n,m	N/A	6	3
JPC	nm	Jump to Address if C	if C=0 nm -> PC else PC += 3	0x7f,n,m	N/A	6	3
JPNC	nm	Jump to Address if not C	if C=0 nm -> PC else PC += 3	0x7d,n,m	N/A	6	3
PTC		Prepare Table Jump				9	
----->>	PTJ:0x7a/DTJ:0x69, CPCAL/DTLRA,CASE1/CASE2, SETT,JST are PTC synonyr			??????	????	????	
JRP	n	Jump Relative Forward (plus)	PC += 1 + n	0x2c,n	N/A	7	2
JRZP	n	Jump Relative if Z Forward (plus)	if Z=1 PC += 1 + n else PC += 2	0x38,n	N/A	7/4	2
JRNZP	n	Jump Relative if not Z Forward (plus)	if Z=0 PC += 1 + n else PC += 2	0x28,n	N/A	7/4	2
JRCP	n	Jump Relative if C Forward (plus)	if C=1 PC += 1 + n else PC += 2	0x3a,n	N/A	7/4	2
JRNCP	n	Jump Relative if not C Forward (minus)	if C=0 PC += 1 + n else PC += 2	0x2a,n	N/A	7/4	2
JRM	n	Jump Relative Backward (minus)	PC += 1 - n	0x2d,n	N/A	7	2
JRZM	n	Jump Relative if Z Backward (minus)	if Z=1 PC += 1 - n else PC += 2	0x39,n	N/A	7/4	2
JRNZM	n	Jump Relative if not Z Backward (minus)	if Z=0 PC += 1 - n else PC += 2	0x29,n	N/A	7/4	2
JRCM	n	Jump Relative if C Backward (minus)	if C=1 PC += 1 - n else PC += 2	0x3b,n	N/A	7/4	2
JRNCM	n	Jump Backward Relative to PC if C is 0	if C=0 PC += 1 - n else PC += 2	0x2b,n		7/4	2

Arithmetic

Mnemonic		Operation	Function	Hex	Flag	Cycle	Bytes
ADIA	n	Add Immediate to A	A + n --> A	0x74, n	C,Z	4	2
SBIA	n	Subtract Immediate from A	A - n --> A	0x75, n	C,Z	4	2
ADIM	n	Add Immediate to (P)	(P) + n --> (P)	0x70, n	C,Z	4	2
SBIM	n	Subtract Immediate from (P)	(P) - n --> (P)	0x71, n	C,Z	4	2
ADM	N/A	Add Accumulator into (P)	(P) + A --> (P)	0x44	C,Z	3	1
SBM	N/A	Subtract Accumulator from (P)	(P) - A --> (P)	0x45	C,Z	3	1
ADCM	N/A	ADM with carry	(P) + A --> (P), with carry	0xc4	C,Z	3	1
SBCM	N/A	SBM with carry	(P) - A --> (P), with carry	0xc5	C/Z	3	1
ADB	N/A	16 bit ADM	(P) + A --> (P), 16 bit	0x14	C,Z	5	1
SBB	N/A	16 bit SBM	(P) - A --> (P), 16 bit	0x15	C,Z	5	1
ADN	N/A	ADM with BCD addition for I + 1 bytes	(P) + A --> (P), BCD, I+1 bytes	0x0c	C,Z	7+3*I	1
SBN	N/A	SBM with BCD addition for I + 1 bytes	(P) - A --> (P), BCD, I+1 bytes	0x0d	C,Z	7+3*I	1
ADW	N/A	Add (Q) to P, BCD, I + 1 bytes	(P) + (Q) --> (P), BCD, I+1 bytes	0x0e	C,Z	7+3*I	1
SBW	N/A	Subtract (Q) from P, BCD, I + 1 bytes	(P) - (Q) --> (P), BCD, I+1 bytes	0x0f	C,Z	7+3*I	1
INC[Reg]	N/A	Increment Register P,I,J,A,B,K,L,M,N	[Reg] + 1 --> [Reg]	See Next	C,Z	4	1
----->>	**A**: 0x42 **B**: 0xc2 **I**: 0x40 **J**: 0xc0 **K**: 0x48 **L**: 0xc8 **M**: 0x4a **N**: 0xca **P**: 0x50						1
DEC[Reg]	N/A	Decrement Register P,I,J,A,B,K,L,M,N	[Reg] - 1 --> [Reg]	See Next	C,Z	4	1
----->>	**A**: 0x43 **B**: 0xc3 **I**: 0x41 **J**: 0xc1 **K**: 0x49 **L**: 0xc9 **M**: 0x4b **N**: 0xcb **P**: 0x51						1
IX	N/A	Increment X, Load X into Data Pointer	X + 1 --> X, X --> DP	0x04	N/A	6	1
DX	N/A	Decrement X, Load X into Data Pointer	X - 1 --> X, X --> DP	0x05	N/A	6	1
IY	N/A	Increment Y, Load Y into Data Pointer	Y + 1 --> Y Y --> DP	0x06	N/A	6	1
DY	N/A	Decrement Y, Load Y into Data Pointer	Y - 1 --> Y, Y --> DP	0x07	N/A	6	1
IXL	N/A	IX plus LDD	X + 1 --> X, X --> DP, (DP) --> A	0x24	N/A	7	1
DXL	N/A	DX plus LDD	X - 1 --> X, X --> DP, (DP) --> A	0x25	N/A	7	1
IYS	N/A	IY plus STD	Y + 1 --> Y Y --> DP, A --> (DP)	0x26	N/A	6	1
DYS	N/A	DY plus STD	Y - 1 --> Y, Y --> DP, A --> (DP)	0x27	N/A	6	1

Fill, Shift, Compare

Mnemonic		Description	Operation	Hex	Flag	Cycle	
FILM	N/A	Load A into I + 1 bytes of (P)	A --> (P), I+1 bytes	0x1e	N/A	5+I	1
FILD	N/A	Load A into I + 1 bytes of (DP)	A --> (DP), I+1 bytes	0x1f	N/A	4+3*I	1
SRW	N/A	Shift I + 1 bytes in (P) a word 4 bits right	(P) >> 4, I+1 bytes	0x1c	N/A	5+I	1
SLW	N/A	Shift I + 1 bytes in (P) a word 4 bits left	(P) << 4, I+1 bytes	0x1d	N/A	5+I	1
SR	N/A	Shift A 1 bit right with carry	A >> 1, w/carry	0xd2	C	2	1
SL	N/A	Shift A 1 bit left with carry	A << 1, w/carry	0x5a	C	2	1
SWP	N/A	Swap high and low nibble of A	Ah <--> Al	0x58	N/A	2	1
CPIA	n	Compare A and n and set c,z	A - n --> c,z	0x67, n	Z,C	4	2
CPIM	n	Compare (P) and n and set c,z	(P) - n --> c,z	0x63, n	Z,C	4	2
CPMA	N/A	Compare (P) and A and set c,z	(P) - A --> c,z	0xc7	Z,C	3	1
TSIA	n	Logical AND A and n and set z if 1	A & n --> z	0x66, n	N/A	4	2
TSIM	n	Logical AND (P) and n and set z if 1	(P) & n --> z	0x62, n	N/A	4	2
TSID	n	Logical AND (DP) and n and set z if 1	(DP) & n --> z	0xd6, n	N/A	6	2
TSIP	N/A	Logical AND (P) and A and set z if 1	(P) & A --> z				1

Logic

Mnemonic		Operation	Function	Hex	Flag	Cycle	
ANIA	n	Logical AND A and n into A	A & n --> A	0x64, n	Z	4	2
ORIA	n	Logical OR A and n into A	A | n --> A	0x65, n	Z	4	2
ANIM	n	Logical AND (P) and n into (P)	(P) & n --> (P)	0x60, n	Z	4	2
ORIM	n	Logical OR (P) and n into (P)	(P) | n --> (P)	0x61, n	Z	4	2
ANID	n	Logical AND (DP) and n into (DP)	(DP) & n --> (DP)	0xd4, n	Z	6	2
ORID	n	Logical OR (DP) and n into (DP)	(DP) | n --> (DP)	0xd5,n	Z	6	2
ANMA	N/A	Logical AND (P) and A into (P)	(P) & A --> (P)	0x46	Z	3	1
ORMA	N/A	Logical OR (P) and A into (P)	(P) | A --> (P)	0x47	Z	3	1

Load/Store, Move, Exchange, I/O

Mnemonic		Description	Operation	Hex	Flag	Cycle	
LI[Reg]	n	Load Immediate to Register I,J,A,B,P,Q	n --> [Reg]	See Next	N/A	4	2
----->>	**I**: 0x00 **J**: 0x01 **A**: 0x02 **B**: 0x03 **P**: 0x12 **Q**: 0x13						**2**
LIDP	nm	Load Immediate to DP	n --> DP	0x10	N/A	8	3
LIDL	n	Load Immediate to low byte of DP	n --> DL	0x11	N/A	5	2
LP	n	One Byte Version of LIP	n --> P				2
RA	N/A	Same as LIA 0 but takes One Byte	0 --> A	0x23	N/A	2	1
CLRA	N/A	Synonym of RA	0 --> A				1
LD[Reg]	N/A	Load Register P,Q,R, into Accumulator	[Reg] --> A		N/A	2	1
----->>	**P**: 0x20 **Q**: 0x21 **R**: 0x22						1
LDM	N/A	Load (P) into Accumulator	(P) --> A	0x59	N/A	2	1
LDD	N/A	Load (DP) into Accumulator	(DP) --> A	0x57	N/A	3	1
ST[Reg]	N/A	Store Accumulator in Register P,Q,R	A --> [Reg]	See Next	N/A	2	1
----->>	**P**: 0x30 **Q**: 0x31 **R**: 0x32						1
STD	N/A	Store Accumulator in (DP)	A --> (DP)	0x52	N/A	2	1
MVDM	N/A	Move (P) into (DP)	(P) --> (DP)	0x53	N/A	3	1
MVMD	N/A	Move (DP) into (P)	(DP) --> (P)	0x55	N/A	3	1
EXAB	N/A	Exchange A and B	A <--> B	0xda	N/A	3	1
EXAM	N/A	Exchange A and (P)	A <--> (P)	0xdb	N/A	3	1
MVW	N/A	Move I + 1 bytes of (Q) into (P)	(Q) --> P, I+1 bytes	0x08	N/A	5+2*J	1
MVB	N/A	Move J + 1 bytes of (Q) into (P)	(Q) --> P, J+1 bytes	0x0a	N/A	5+2*J	1
MVWB	N/A	Move I + 1 bytes of (DP) into (P)	(DP) --> (P), I+1 bytes	0x18	N/A	5+4*J	1
MVBD	N/A	Move J + 1 bytes of (DP) into (P)	(DP) --> (P), I+1 bytes	0x1a	N/A	5+4*J	1
DATA	N/A	Move I + 1 bytes of (B,A) into (P)	(B,A) --> (P), I+1 bytes	0x35	N/A	11+4*I	1
EXW	N/A	Exchange I + 1 bytes of (Q) and (P)	(Q) <--> (P), I+1 bytes	0x09	N/A	6+3*J	1
EXB	N/A	Exchange J + 1 bytes of (Q) and (P)	(Q) --> P, J+1 bytes	0x0b	N/A	6+3*J	1
EXWD	N/A	Exchange I + 1 bytes of (DP) and (P)	(DP) <--> (P), I+1 bytes	0x19	N/A	7+6*J	1
EXBD	N/A	Exchange J + 1 bytes of (DP) and (P)	(DP) <--> (P), J+1 bytes	0x1b	N/A	7+6*J	1
INA	N/A	Load IA into Accumulator	IA --> A	0x4c	N/A	2	1
INB	N/A	Load IB into Accumulator	IB --> A	0xcc	N/A	2	1
OUTA	N/A	Load Accumulator into OUTA	A --> AOUT	0x5d	N/A	3	1
OUTF	N/A	Load Accumulator into OUTF	A --> FOUT	0x5f	N/A	3	1
OUTC	N/A	Load Accumulator into Control Port	A --> COUT	0xdf	N/A	2	1

Unknown

Mnemonic		Description	Operation	Hex	Flag	Cycle	
READM	N/A	Load (PC + 1) into (P)	(PC + 1) -> (P)	0x54	N/A	3	1
READ	N/A	Load (PC + 1) into Accumulator	(PC + 1) -> A	0x56	N/A	3	1
LPI			I -> P ?	0x80	N/A	2	
TSMA	N/A		(P) and A -> Z	0xc6	N/A	3	1
OUTB	N/A		(5D) -> IB-Port	0xdd	N/A	2	1
TEST	n			0x6b, n	Z	4	2
LOOP	n			0x2f,n			2