***Ver notas del autor al final del documento / See author's notes at the end of this document
SC61860 (aka ESR-H) Instruction Set*******
Version 1.11, released 2014-07-31

Mnemonic	Size	Opcode	Clock	cz	Effect	Remarks
ADB	1	14	5	cz	(P)+A->(P), (P+1)+B+c->(P+1), P+1->P	
ADCM	1	C4	3	cz	(P)+A+c->(P)	
ADIA n	2	74 XX	4	cz	A+n->A	
ADIM n	2	70 XX	4	cz	(P)+n->(P)	
ADM	1	44	3	cz	(P)+A->(P)	
ADN	1	0C	7+3*I	cz	(P)+A->(P)..(P-I)+c->(P-I) P-I-1->P, BCD	
ADW	1	0E	7+3*I	cz	(P)+(Q)->(P)..(P-I)+(Q-I)+c->(P-I) P-I-1->P, Q-I-2->Q, BCD	
ANIA n	2	64 XX	4	.z	A&n->A	
ANID n	2	D4 XX	6	.z	(DP)&n>(DP), (DP)->(R-1)	(R-1) used
ANIM n	2	60 XX	4	.z	(P)&n->(P)	
ANMA	1	46	3	.z	(P)&A->(P)	
CAL nm	2	E0+n XX	7	..	PC+2->(R-1,R-2), R-2->R, nm->PC	n<=$1f
CALL nm	3	78 XX XX	8	..	PC+3->(R-1,R-2), R-2->R, nm->PC	
CASE1 ->PTC						
CASE2 ->DTC						
CDN	1	6F	1+4*I	.z	I->d, REPEAT d-1->d, P+1->P UNTIL I=FF or Xin=0	
CLRA	1	23	2	..	0->A, 0->H	=LDS, but S is always 0
CPIA n	2	67 XX	4	cz	A-n	
CPID n	2	D7 XX	6?	cz	(DP)-n	undocumented, (R-1) used, may not be fully implemented
CPIM n	2	63 XX	4	cz	(P)-n	
CPCAL ->PTC						
CPMA	1	C7	3	cz	(P)-A	

CUP	1	4F	1+4*I	.z	I->d, REPT d-1->d, P+1->P UNTIL I=FF or Xin=1	
DATA	1	35	11+4*I	..	(BA)...(BA+1)->(P)...(P+1)	
DECA	1	43	4	cz	A-1->A, 2->Q	
DECB	1	C3	4	cz	B-1->B, 3->Q	
DECI	1	41	4	cz	I-1->I, 0->Q	
DECJ	1	C1	4	cz	J-1->J, 1->Q	
DECK	1	49	4	cz	K-1->K, 8->Q	
DECL	1	C9	4	cz	L-1->L, 9->Q	
DECM	1	4B	4	cz	M-1->M, 10->Q	
DECN	1	CB	4	cz	N-1->N, 11->Q	
DECP	1	51	2	..	P-1->P	
DTC n,nm	*	69	5+7*d	.z	FOR i=1 TO d: IF A=n nm->PC: NEXT i	n,nm are .db statements
DTLRA ->DTC					A->H	in most assemblers
DX	1	05	6	..	X-1->X, X->DP, 5->Q, Xl->H	
DXL	1	25	7	..	X-1->X, X->DP, (DP)->A, 5->Q, Xl->H	
DY	1	07	6	..	Y-1->Y, Y->DP, 7->Q, Yl->H	
DYS	1	27	7	..	Y-1->Y, Y->DP, A->(DP), 7->Q, Yl->H	
ETC -> DTC						
EXAB	1	DA	3	..	A<->B	
EXAM	1	DB	3	..	A<->(P)	
EXB	1	0B	6+3*J	..	(P)..(P+J)<->(Q)..(Q+J) P+J+1->P, Q+J+1->Q	
EXBD	1	1B	7+6*J	..	(P)..(P+J)<->(DP)..(DP+J) P+J+1->P, DP+J->DP	
EXW	1	09	6+3*I		(P)..(P+I)<->(Q)..(Q+I) P+I+1->P, Q+I+1->Q	
EXWD	1	19	7+6*I	..	(P)..(P+I)<->(DP)..(DP+I) P+I+1->P, DP+I->DP	
FILD	1	1F	4+3*I	..	A->(DP)..(DP+I), DP+I->DP	
FILM	1	1E	5+I	..	A->(P)..(P+I), P+I+1->P, A->H	
HALT	1	7B	n/a	?	Stops the CPU, possible side effects	undocumented
INA	1	4C	2	.z	IA-Port->A	

```
INB             1       CC       2      .z      IB-Port->A
INCA            1       42       4      cz      A+1->A, 2->Q
INCB            1       C2       4      cz      B+1->B, 3->Q
INCI            1       40       4      cz      I+1->I, 0->Q
INCJ            1       C0       4      cz      J+1->J, 1->Q
INCK            1       48       4      cz      K+1->K, 8->Q
INCL            1       C8       4      cz      L+1->L, 9->Q
INCM            1       4A       4      cz      M+1->M, 10->Q
INCN            1       CA       4      cz      N+1->N, 11->Q
INCP            1       50       2      ..      P+1->P
IPXH -> CDN
IPXL -> CUP
IX              1       04       6      ..      X+1->X, X->DP, 5->Q, Xh->H
IXL             1       24       7      ..      X+1->X, X->DP, (DP)->A, 5->Q
IY              1       06       6      ..      Y+1->Y, Y->DP, 7->Q, Yh->H
IYS             1       26       7      ..      Y+1->Y, Y->DP, A->(DP), 7->Q

JP nm           3       79 XX XX 6      ..      nm->PC
JPC nm          3       7F XX XX 6      ..      IF c=1 nm->PC
JPNC nm         3       7D XX XX 6      ..      IF c=0 nm->PC
JPNZ nm         3       7C XX XX 6      ..      IF z=0 nm->PC
JPZ nm          3       7E XX XX 6      ..      IF z=1 nm->PC
JRCM n          2       3B XX    7/4    ..      IF c=1 PC+1-n->PC
JRCP n          2       3A XX    7/4    ..      IF c=1 PC+1+n->PC
JRM n           2       2D XX    7      ..      PC+1-n->PC
JRNCM n         2       2B XX    7/4    ..      IF c=0 PC+1-n->PC
JRNCP n         2       2A XX    7/4    ..      IF c=0 PC+1+n->PC
JRNZM n         2       29 XX    7/4    ..      IF z=0 PC+1-n->PC
JRNZP n         2       28 XX    7/4    ..      IF z=0 PC+1+n->PC
JRP n           2       2C XX    7      ..      PC+1+n->PC
JRZM n          2       39 XX    7/4    ..      IF z=1 PC+1-n->PC
JRZP n          2       38 XX    7/4    ..      IF z=1 PC+1+n->PC
JST -> DTC

LDD             1       57       3      ..      (DP)->A
LDM             1       59       2      ..      (P)->A
```

LDP	1	20	2	..	P->A	
LDPC	1	56	3	..	(PC+1)->A	undocumented
LDQ	1	21	2	..	Q->A	
LDR	1	22	2	..	R->A	
LDS -> CLRA						
LEAVE	1	D8	2	..	0->(R)	(R-1) used
LIA n	2	02 XX	4	..	n->A	
LIB n	2	03 XX	4	..	n->B	
LIDL m	2	11 XX	5	..	m->DPl, m->H	
LIDP nm	3	10 XX XX	8	..	nm->DP, n->H	
LII n	2	00 XX	4	..	n->I	
LIIH n	2	72 XX	2	.0	H->I, n->H	undocumented, alt: 73/76/77
LIJ n	2	01 XX	4	..	n->J	
LIP n	2	12 XX	4	..	n->P, 0->H	
LIQ n	2	13 XX	4	..	n->Q, n->H	
LOOP n	2	2F XX	10/7	cz	(R)-1->(R), IF c=0 PC+1-n->PC	(R-1) used
LP n	1	80+n	2	..	n->P, 80+n->H	n<63
MVB	1	0A	5+2*J	..	(Q)..(Q+J)->(P)..(P+J) P+J+1->P, Q+J+1->Q	
MVBD	1	1A	5+4*J	..	(DP)..(DP+J)->(P)..(P+J) P+J+1->P, DP+J->DP	
MVDM	1	53	3	..	(P)->(DP)	
MVMD	1	55	3	..	(DP)->(P)	
MVMP	1	54	3	..	(PC+1)->(P)	
MVW	1	08	5+2*I	..	(Q)...(Q+I)->(P)...(P+I) P+I+1->P, Q+I+1->Q	
MVWD	1	18	5+4*I	..	(DP)...(DP+I)->(P)...(P+I) P+I+1->P, DP+I->DP	
MVWP -> DATA						
NOPT	1	CE	3	..	none	alt: 69/6A
NOPW	1	4D	2	..	none	alt: CD/D3/D9
ORIA n	2	65 XX	4	.z	A|n->A	

ORID n	2	D5 XX	6	.z	(DP)|n->(DP), (DP)->(R-1)	1 extra stack space used
ORIM n	2	61 XX	4	.z	(P)|n->(P)	
ORMA	1	47	3	.z	(P)|A->(P)	
OUTA	1	5D	3	..	(5C)->IA-Port, 5C->Q (?)	
OUTB	1	DD	2	..	(5D)->IB-Port, 5D->Q (?)	
OUTC	1	DF	2	..	(5F)->C-PORT	
OUTF	1	5F	3	..	(5E)->F0-Port, 5E->Q (?)	
POP	1	5B	2	..	(R)->A, R+1->R	
PTC n,nm	4	7A	9?	..	n->d,nm->(R-1,R-2), R-2->R	n,nm are .db statements
PTJ ->PTC						in most assemblers
PUSH	1	34	3	..	A->(R), R-1->R	
RA -> CLRA						
READ -> LDPC						
READM -> MVMP						
RC	1	D1	2	01	0->c, 1->z	
RTN	1	37	4	..	(R-1,R-2)->PC, R+2->R, PC-H->H	
RZ n -> LIIH						
SBB	1	15	5	cz	(P)-1->(P), (P+1)-B-c->(P+1), P+1->P	
SBCM	1	C5	3	cz	(P)-A-c->(P)	
SBIA n	2	75 XX	4	cz	A-n->A	
SBIM n	2	71 XX	4	cz	(P)-n->(P)	
SBM	1	45	3	cz	(P)-A->(P)	
SBN	1	0D	7+3*I	cz	(P)-A->(P), (P-I)-c->(P-I), P-I-1->P, BCD	
SBW	1	0F	7+3*I	cz	(P)-(Q)->(P), (P-I)-(Q-I)-c->(P-I) P-I-1->P, Q-I-2->Q, BCD	
SC	1	D0	2	11	1->c, 1->z	
SETT ->PTC						
SL	1	5A	2	c.	A<<1, c->A7, A0->c	
SLW	1	1D	5+I	..	(P)<<4 [I+1 bytes], P-I-1->P	
SR	1	D2	2	c.	A>>1, c->A0, A7->c	
SRW	1	1C	5+I	..	(P)>>4 [I+1 bytes], P+I+1->P	
STH	1	33	2	..	A->H, (theoretically A->S)	undocumented
STD	1	52	2	..	A->(DP)	

STP	1	30	2	..	A->P, A->H	
STQ	1	31	2	..	A->Q, A->H	
STR	1	32	2	..	A->R, A->H	
STS -> STH						
SWP	1	58	2	..	Ah<->Al	
SZ ->CPID						
TEST n	2	6B XX	4	.z	Test-byte&n	
TSCM -> TSMA						
TSIA n	2	66	4	.z	A&n	
TSID n	2	D6	6	.z	(DP)&n, (DP)->(R-1)	(R-1) used
TSIM n	2	62	4	.z	(P)&n	
TSIP -> TSMA						
TSMA	1	C6	3	.z	(P)&A	undocumented
WAIT n	2	4E	6+n	..	none	
WAITI -> CUP						
WRIT -> NOPW						

Notes*******

All numeric values in this document are assumed to be hexadecimal, unless stated otherwise.
Instructions marked as "undocumented" may not work as intended depending on the SC61860 version used.
Especially the functionality of the H register is unknown, the information provided here is mostly guesswork.

Unknown Opcodes*******

In addition to the above, the following opcodes are also undocumented, and their effect is unknown.

16,17,2E,36,3C,3D,3E,3F,5C,5E,6C,6D,6E,CF,DC,DE

Registers*******

**

Name	Location int. RAM	Function	Remarks
A	02	Accumulator	
B	03	Secondary Accumulator	
I	00	Index/Counter	
J	01	Index/Counter	
H	n/a	Internal Aux. Register	undocumented
K	08	General Purpose/Counter	
L	09	General Purpose/Counter	
M	0A	General Purpose/Counter	aka V, undocumented
N	0B	General Purpose/Counter	aka W, undocumented
P	n/a	Internal RAM pointer	treated as 7 bit
Q	n/a	Internal RAM pointer	treated as 7 bit
R	n/a	Stack pointer	treated as 7 bit
S	n/a	unused	undocumented
X	04-05	16 bit RAM pointer	Xh=04, Xl=05
Y	06-07	16 bit RAM pointer	Yh=06, Yl=07
DP	n/a	16 bit Data pointer	
PC	n/a	16 bit Program counter	
c		Carry flag	
d (pseudo register)		Internal counter	
z		Zero flag	

There are also 4 8 byte pseudo registers used internally for BCD calculations.
They are located at 10-17, 18-1f, 20-17, and 28-2f respectively.

The P, Q, and R registers are treated as 7 bit internally, but seen as 8-bit from the data bus.
Hence, they can be read and written to as 8 bit registers.

***Using the LOOP
command**********

The LOOP command allows the execution of conditional loops without the use of a register. It is also 1 t-state faster than looping with DECr, JRNZM lloop.
Use it as follows:

```
                LIA n           ;n=[number of iterations]-1
                PUSH            ;put it in (R)
lloop:          ...
                ...
                LOOP lloop      ;(R)-1->(R), do JRNCM lloop
```

The loop can be broken by executing the LEAVE instruction, which will set (R) to 0.

Using the PTC/DTC instructions*******

The PTC/DTC (aka CASE1/CASE2, PTJ/DTJ, CPCAL/DTLRA, SETT/JST) instructions can be used to implement a switchcase.

This is a rather size efficient solution, but usually slower than using a jump table.

A lot of confusion regarding this instruction seems to be caused by fact that it is usually called "Prepare/Do Table Jump". PTC/DTC does not JumP, it CALLs subroutines.

Handling varies between the various available assemblers. In most cases, the arguments need to be entered as data

byte statements (.db), but there are some (like AS6186), which expects PTC with 3 argument bytes.

Use the commands as follows

```
                PTC
                .db k           ;where k is the number of cases
                .dw nm          ;where nm is the return address of the subroutines
                DTC
                .db n           ;where n is the condition to evaluate
                .dw nm          ;where nm is the call address
```

```
                ...             ;repeat k times
                .dw nm          ;where nm is a default dummy address to be called
```

Sources*******

**

http://www.lehmayr.de/PC-1360/e_Manuals/ml.htm
http://www.leonidastolias.com/Site/SharpEL5500III_files/61860%20Instruction%20Set.pdf
http://shop-pdp.kent.edu/ashtml/as6186.htm
http://page.freett.com/osamus/esrhinst.html
http://destroyedlolo.info/sharp/LM.html
http://www.oit.ac.jp/bme/~yagshi/misc/pocketcom/sc61860op.html
https://github.com/pockemul/PockEmul/blob/master/src/cpu/sc61860.cpp
PC-1350 ML Reference Manual
Information obtained in conversation with Torsten MŌcker

****

**

This document is written and maintained by utz^irrlichtproject. Visit me at www.irrlichtproject.de!