SBC68K
68000 Single Board Computer User Manual

TABLE OF CONTENTS

1. INTRODUCTION

The SBC68K is a 68000 based single board microcomputer. The board contains necessary logic that is designed to provide an inexpensive tool for testing 68000 assembly language programs. For easy use, the board has a resident firmware packages that provides a self-contained programming and operating environment. The firmware provides the user with monitor/debug, program entry, assembly/disassembly, and I/O control functions.

1.1 The SBC68K specifications:

a. 8 MHz 68000 16-bit MPU
b. 16K SRAM (6264 x 2) or 64K SRAM (256 x 2)
c. 64K EPROM (27256 x 2) or 128K EPROM (27512 x 2)
d. Two RS-232e serial ports (9600 or 19200 baud)
e. One timer/counter and one 24-bit parallel port
f. Self-contained operating firmware that provides monitor and debug function
g. Individual LEDs indicate: 68000 free-run, DTACK, RAM/EPROM chip enable
h. Five volt only operation
i. Size 8" x 5"

1.2 HARDWARE DESCRIPTION

The SBC68K provides the SRAM, EPROM, timer/counter and I/O. The 68000 microprocessor has 16-bit data bus (DO - D15) and a 23-bit address bus (Al - A23), provides a memory addressing range of 16 megabytes. The processor also has eight 32-bit data register, seven 32-bit address registers, two 32-bit stack pointer, a 32-bit program counter and a 16-bit status register.

A 8 MHz 68000 is used on the SBC68K. All the memory and I/O devices communicate with the MPU via a common parallel bus. The building block of the SBC68K is shown as below:

a. SRAM
b. EPROM
c. 68000 MPU
d. PARALLEL I/O & TIMER/COUNTER
e. RS-232C SERIAL PORTS

a. An 68000 microprocessor Operating at the speed of 8 MHz
b. 64K EPROM to store the monitor called TUTOR. The range of the on board EPROM is 20000 - 2FFFF for 27256 x2, or 20000 - 3FFFF for 27512 x 2 and addresses from 00000 - 00007 are also read-only, which are reserved for the reset exception vector.
C· 16K or 64K SRAM to store the user's program and data. The on board SRAM occupies the addresses from 00008 - 03FFF for 6264 x 2, or 00008 OFFFF for 62256 x 2. However, the locations from 00008 to 003FF are allocated for the exception vector table and the area from 00400 to 008FF is reserved as temporary storage by TUTOR. Therefore the user's program must be loaded into the area from 00900 to 03FFF or 00900 to OFFFF.
d. Two asynchronous serial interface devices (ACIA) are used to provide two serial ports. Port 1 is for connection to PC as the system console. Port 2 provides a link to other devices with RS-232C interface.
e. An integrated parallel interface and timer device (PI/T). This device provides a 24 bit I/O port and a time /counter.

1.3 INSTALLATION AND SET UP

The following steps illustrated how to set up the SBC68K:

a. Connect the supplied serial cable to PC COM2 and SBC68K COMl
b. Use a Terminal or CRT software and set the speed to 19200 baud, 8N1 & VT100 mode
C· Connect the SBC68K with 5 volt DC

Once everything has been connected and double check, then turn on the PC and run terminal emulation program. Next apply 5 volt DC power to the SBC68K. If the TUTOR firmware is installed and everything is connected properly you should see the message:

TUTOR 1.3 >

Enter "HE" and press "Enter" to display a list of TUTOR commands.

```
TUTOR 1.3 >HE
·PC .SR .US .SS
·DO .D1 .D2 .D3 .D4 .D5 .D6 .D7
.AO .Al .A2 .A3 .A4 .A5 .A6 .A7
.RO .R1 .R2 .R3 .R4 .R5 .R6

BF   BM   BR   NOBR BS   BT   DC   DF
DU   G    GD   GO   GT   HE   LO   M
MD   MM   MS   OF   PA   NOPA PF   T
TM   TR   TT   VE

TUTOR 1.3 >
```

If you cannot obtain the appropriate display, reset the board by pressing the reset switch and try again. If there is still no response, double check all of *your* power and terminal connections and verify that your PC is set for the correct baud rate and character format.

2. DEVICES ADDRESSES

Device	Address	Description
SRAM	00008 - 03FFF 00008 - OFFFF	6264 x 2 62256 x 2
EPROM	20000 - 2FFFF 20000 - 3FFFF	27256x 2 27512 x 2
Reset Interrupt vector	00000 - 00003 (stored in EPROM) 00005 - 00007 (stored in EPROM)	Supervisor stack pointer(SSP) program counter (PC)
6850 ACIA	10040	ACIA1 control register **(write only)** status register **(read only)**
	10041	ACIA2 control register **(write only)** status register **(read only)**
	10042	ACIA1 transmit data register **(write only)** receive data register **(read only)**
	10043	ACIA2 transmit data register **(write only)** receive data register **(read only)**

2. DEVICES ADDRESSES (Con’t)

Device	Address	Description
68230 PI/T		
register	10001	port general control
register	10003	port service request
register	10005	port A data direction
register	10007	port B data direction
register	10009	port C data direction
register	1000B	port interrupt vector
	1000D	port A control register
	1000F	port B control register
	10011	port A data register
	10013	port B data register
	10015	port A alternate register
	10017	port B alternate register
	10019	port C data register
	1001B	port status register
	10021	timer control register
	10023	timer interrupt vector
register	10027	counter preload register
high	10029	counter preload register
middle	1002B	counter preload register
low	1002F	counter register high
	10031	counter register middle
	10033	counter register low
	10035	timer status register

3. JUMPER SETTING

a. JP2 - baud rate

- Short pin 1 & pin 2 = 19200 baud
- Short pin 3 & pin 4 = 9600 baud

b. JP3 - SRAM size

- Short pin 1 & pin 2 = 62256
- Short pin 3 & pin 4 = 6264

C· JP4 - EPROM size

- Short pin 1 & pin 2 = 27256
- Short pin 3 & pin4 = 27512

4. CONNECTORS PIN LIST

4.1 PWR connector

1. NC	2. GND
3. GND	4. VCC

4.2 J1 connector

1. PAO	2. PA1
3. PA2	4. PA3
5. PA4	6. PA5
7. PA6	8. PA7
9. NC	10. NC
11. H1	12. H2
13. H3	14. H4
15. GND	16. GND
17. PBO	18. PB1
19. PB2	20. PB3
21. PB4	22. PB5
23. PB6	24. PB7
25. NC	26. NC
27. PCO	28. PC1
29. PC2	30. PC4
31. NC	32. NC
33. VCC	34. VCC

4.3 J2 connector

1. DO	2. D1
3. D2	4. D3
5. D4	6. D5
7. D6	8. D7
9. D8	10. D9
11. D10	12. D11
13. D12	14. D13
15. D14	16. D15
17. GND	18. GND
19. Al	20. A2
21. A3	22. A4
23. A5	24. A6
25. A7	26. A8
27. A9	28. A10
29. All	30. A12
31. A13	32. A14
33. A15	34. A16
35. A17	36. A18
37. NC	38. NC
39. VCC	40. VCC

4.4 J3 connector

1. IRQ7	2. IACK7
3. IRQ2	4. IACK2
5. IRQ1	6. IACK1
7. GND	8. GND
9. CLK	10. UDS
11. FOR	12. LDS
13. E	14. R/W
15. AS	16. VMA
17. VPA_E	18. DTACK_E
19. VCC	20. VCC

4.5 COMI & COM2 connectors

1. NC 3. TXD 5. GND 7. RTS 9. NC	2. RXD 4. NC 6. NC 8. CTS 10. NC

5. SERIAL CABLE CONNECTION LIST

PC (DB 9 FEMALE)	SBC68K (10PIN IDC HEADER)
PIN 2 TXD PIN 3 RXD PIN 5 GND PIN 4 DTR	PIN 3 RXD PIN 2 TXD PIN 5 GND PIN 8 CTS

6. BIBLIOGRAPHY

1. MC68000 Educational Computer Board User's Manual.
Motorola Inc., Austin, Texas; Motorola Inc. 1982.

2. 68000 Microcomputer Systems Designing and Troubleshooting.
Alan D. Wilcox, Prentice Hall Inc., 1987.

3. The M68000 Microprocessor Family.
Fundamentals of Assembly Language Programming and Interface Design
Yu - Cheng Liu, Prentice Hall International Editions.

4. Programming and Designing with the 68000 Family.
Tibet Mimer, Prentice Hall International Editions.

5. MC68000 Assembly Language and System Programming.
William Ford, William Topp, D.C. Heath and Company.

6.68000 Microcomputer Experiments
Using the Motorola Educational Computer Board
Alan D. Wilcox, Prentice Hall.

APPENDIX A   TUTOR COMMAND DETAIL

A.l Block of memory fill

This command fills memory words Starting from <addrl> through <addr2> with the data word given in <word>

BF <addrl> <addr2> <word>

```
TUTOR  1.3 >MD 2004
2004   0D 8E OF 35 07 36 OF 1B  07 09 87 6A 49 CO C9 6C

TUTOR  1.3 >BF 2004 200A 475A
PHYSICAL ADDRESS 00002004 0000200A

TUTOR  1.3 > MD 2004
2004   47 5A 47 5R 47 5A 47 5A  07 09 87 6A 19 CO C9 6C

TUTOR  1.3 > BF 2004 2012 7
PHYSICAL ADDRESS=00002004 00002012

TUTOR  1.3 > MD 2004
2004   00 07 00 07 00 07 00 07  00 07 00 07 00 07 00 07

TUTOR  1.3 >
```

A.2 Block of memory move

This command moves a memory block starting with <addrl> through *<addr2>* to another block beginning at <addr3>

BM <addrl> <addr2> <addr3>

```
TUTOR  1.3 > MD 1800 10
1800   OF 8E 1F OD 07 8B EE 83  2E OF OD 17 OD 5D OF OB

TUTOR  1.3 >BM 1806 1809 1804
PHYSICAL ADDRESS=00001806 00001809
PHYSICAL ADDRESS=00001804

TUTOR  1.3 > MD 1800 10
1800   OF 8E 1F OD EE 83 ZE OF  2E OF OD 17 OD 5D OF OB

TUTOR  1.3 >
```

A.3 Break point set

This command sets one to eight break point. <addr> indicates the location of a break point and <count> specifies a count that is decremented each time the break point is encounted until count = 0. If no parameters are given, the BR command display all break points.

BR <addr> [;<count>]

```
TUTOR  1.3 > BR

BREAKPOINTS

TUTOR  1.3 > BR 1010 2000;5 2040 4000

BREAKPOINTS
001010    001010
002000    002000;5
002040    002040
004000    004000
```

```
TUTOR  1.3 > NOBR 1010 2040

BREAKPOINTS
002000    002000;5
004000    004000

TUTOR  1.3 > NOBR
 BREAKPOINTS


 TUTOR  1.3 >
```

A.4 Block of memory search; 8, W, L

This command searches memory starting with <addrl> through <addr2> for the data given in <data>. The optional <mask>, if used, is ANDed to data.

BS <addrl> <addr2> <data> [<mask>]

```
TUTOR  1.3 > MD 1FFF 15
001FF0   09 49 1B EB CC C9 C9 41  49 C9 49 49 69 4B 49 2B
002000   0F 9D 0F 3A 00 07 00 07  00 07 00 07 00 07 00 07

TUTOR  1.3 > BS 1FF0 200F 'CC'
PHYSICAL ADDRESS=00001FF0 0000200F

TUTOR 1.3 > BS 1FF0 200F 41 ;W
PHYSICAL ADDRESS=00001FF0 0000200F

TUTOR  1.3 > BS 1FF0 200F 41 ;B
PHYSICAL ADDRESS=00001FF0 0000200F
001FF7 41

TUTOR  1.3 >
```

A.5 Block memory test

This command performs a memory test starting with <addrl> through <addr2>(both are even addresses). If no error is detected, the tested memory block is set to all zeros; otherwise, a message is displayed.

BT <addrl> <addr2>

```
TUTOR  1.3 > BT 1000 1FFE
PHYSICAL ADDRESS=00001000 00001FFE

TUTOR  1.3 > BT 2000 3FFE
PHYSICAL ADDRESS=00002000 00003FFE

TUTOR 1.3 >
```

A.6 Data conversion

This command computes <expression> and shows the result in both hexadecimal and decimal forms.

DC <expression>

```
TUTOR  1.3 >DC &120
$78=&120

TUTOR  1.3 > DC &15+$4-$13
$0=&0

TUTOR  1.3 > DC -1000
$FFFFF000=-$1000=-&4096

TUTOR  1.3 > .RO 1000

TUTOR  1.3 > OF
RO=00001000 R1=00000000 R2=00000000 R3=00000000
R4=00000000 R5=00000000 R6=00000000 R7=00000000

TUTOR  1.3 > DC 10+10+30
$1050=&4176
```

```
TUTOR  1.3 > DC 10+10+30+R7
$50=&80

TUTOR  1.3 >
```

A.7 Display formatted registers

This command displays all 68000 registers: DO - D7, AO - A6, PC, SR, USP, and SSP.

DF

```
TUTOR  1,3 > DF
PC=00000000 SR=2700=.S7.....US=00000000 SS=00000786
DO=0000FF00 D1=00000002 D2=00000000 D3=00000000
D4=00000030 D5=00000000 D6=00000008 D7=00000000
AO=00010040 A1=000000C0 A2=00006000 A3=0000044C
A4=00000468 A5=00000540 A6=00000540 A7=00000786
--------------------000000    0000              DC.W  $0000

TUTOR  1.3 >
```

A.8 Dump memory(S - record)

This command sends out memory contents starting with <addrl> through <addr2> to <port no.> in the S - record format. The optional <text>, if used, is output as part of the header.

DU [<port no.>] <addrl> <addr2>, [<text>]

```
TUTOR  1.3 > DU1 2000 200E
PHYSICAL ADDRESS=00002000 0000200E
S0030000FC
S112200000000000000000000000000000000000CD
S9030000FC

TUTOR  1.3 >
```

A.9 Go direct

This command starts program execution from <addr> without setting break points. If the optional starting address is not used, execution begins at the address in PC.

GD [<addr>]

```
TUTOR  1.3 > GO 2000
PHYSICAL ADDRESS=00002000

TUTOR  1.3 >
```

A.10 GO

This command starts program execution from <addr> until a break point in the break point table is encountered or an exception occurs. If a break point has a count associated, program execution does not stop at the break point until its count reaches 0.

GO (or G) [<addr>]

```
TUTOR  1.3 > GO 2000
PHYSICAL ADDRESS=00002000

TUTOR  1.3 >
```

A.11 Go until break point

This command sets a temporary break point at <break point addr>, and starts program execution at the address in PC until the temporary break point or one with a zero count in the break point table is encountered.

GT <break point addr>

```
TUTOR  1.3 > BR 2010 3000

BREAKPOINTS
002010   002010
003000   003000

TUTOR 1.3 > DF
PC=00000000 SR=2700=.S7..... US=00000000 SS=00000786
DO=OOOOFFOC) D1=00000002 D2=0001000A D3=00000000
D4=00000032  D5=0000002C D6=00000002 D7=00000000
AO=00010040  A1=000000CO A2=00000414 A3=00000554
A4=0000387E  A5=00000540 A6=00000540 A7=00000786
--------------------000000    0000              DC.W  $0000

TUTOR  1.3 > GT 2006
PHYSICAL ADDRESS=00002006
PHYSICAL ADDRESS=00000000

TUTOR  1.3 > GT 2010
PHYSICAL ADDRESS=00002010
ERROR
002010   002010
003000   002000

TUTOR  1.3 >
```

A.12 Help

This command displays all the available commands.

HE

```
TUTOR 1.3 > HE
·PC .SR .US .SS
.D0 .D1 .D2 .D3 .D4 .D5 .D6 .D7
.A0 .Al .A2 .A3 .A4 .A5 .A6 .A7
.R0 .R1 .R2 .R3 .R4 .R5 .R6

BF   BM   BR   NOBR BS   BT   DC   DF
DU   G    GD   GO   GT   HE   LO   M
MD   MM   MS   OF   PA   NOPA  PF  T
TM   TR   TT   VE

TUTOR 1.3
```

A.13 Load S –record; options: X, -C

This command loads machine code or data in the S - record format from <port no.> to memory. The optional <=text>, if used with port 2, send a message to that port before loading starts.

LO [<port no.>] [;<optional>] [=text]

```
TUTOR  1.3 >LO1; x
```

A.14 Memory display; optional: DI

This command displays <count> bytes of memory starting with <addr> using <port no.>. The optional DI, if present, displays data in the dissembled form. Otherwise, the hexadecimal form is used.

MD [<port no.>] <addr> [<count>] [;<option>]

```
TUTOR  1.3 > md 1000 12 ;DI

001000   4FF900008000     LEA.L    S00008000,A7
001006   7E00             MOVEQ.L  #0,D7
001008   4EB900001170     JSR.L    $00001170
00100E   41F9000010AA     LEA.L    S000010AA,A0

TUTOR  1.3 >
```

A.15 Memory modify; options: w, L, O, V, N, DI

This command displays and, if needed, modifies memory content at <addr>. The current content after being displayed is replaced by, the new data if entered. A "CR” causes the command to repeat for the next location and a "." terminates the command.

MM <addr> [;<options>]

```
TUTOR  1.3 > MM 2000 ;W
002000    0F0F  ?1111
002002    0F0F  ?2222
002004    0F0F  ?3333
002006    0F0F  ?.

TUTOR  1.3 > MM 2000 04

002000 11 11 22 22 33 33 0F 0F  0F 0F 0F 0F 0F 0F 0F 0F

TUTOR  1.3 > MM 2000 ;DI
002000    2248     MOVE.L A0,A1

TUTOR  1.3 >
```

A.16 Memory set

This command sets memory beginning at <addr> with hexadecimal data or ASCII string given as <data...>.

MS <addr> <data...>

```
TUTOR 1.3 > MD 2000
2000     22 48 22 22 33 33 0F 0F  0F 0F 0F 0F 0F 0F 0F 0F

TUTOR  1.3 > MS 2000 ‘ABC’

TUTOR  1.3 > MS 2006  123 123456

TUTOR  1.3 > MD 2000
002000    41 42 43 22 33 33 01 23  12 34 56 0F 0F 0F 0F 0F
```

A.17 Remove break point

This command sets memory beginning at <addr> from the break point table. If no address is given, all break points are removed.

NOBR [<addr> <addr>.....]

```
TUTOR  1.3 > NOBR 2000;5 2030 3000;6 3060

BREAKPOINTS
002000   002000;5
002030   002030
003000   003000;6
003060   003060
```

```
TUTOR  1.3 > NOBR 3000

BREAKPOINTS
002000   002000;5
002030   002030
003060   003060

TUTOR  1.3 > NOBR

BREAKPOINTS

TUTOR  1.3 >
```

A.18 Reset printer attach

This command detaches the parallel printer from the system terminal connected to port 1.

NOPA

A.19 Display offsets

This command displays the offsets containad in registers R0 – R7.

OF

```
TUTOR  1.3 > OF
RO=00000000 R1=00000O00 R2=00000000 R3=00000000
R4=00000000 R5=00000000 R6=00000D00 R7=00000000

TUTOR  1.3 > .R1 1000

TUTOR  1.3 > .R3 3300

TUTOR  1.3 > OF
R0=00000000 R1=00001000 R2=00000000 R3=00003300
R4=00000000 R5=00000000 R6=00000000 R7=00000000

TUTOR 1.3 >
```

A.20 Printer attach

This command logically attaches the parallel printer to the system terminal connected to port 1. Once attached, the printer prints any information that is displayed on the system terminal.

PA

A.21 Port format

This command displays and, if needed, modifies the character format and other parameters of serial part 1 or 2 as specified by <port no.>.

PF [<port no.>]

A.22 Transparent mode

This command connects Serial ports 1 and 2 together. When <exit character> is entered from port l, direct connection between port 1 and 2 is terminated and <trailing character> is sent to port 2.

TM [<exit character>]

A.23 Trace

This command traces <count> instructions starting with the address in PC. A "CR" causes next instruction to be traced.

TR [<count>]

A.24 Temporary break point trace

This command sets a temporary break point at <break point addr>, and starts tracing at the address in PC until a break point with a zero count is encountered.

TT <break point addr>

A.25 Verify S – record

This command verifies memory contents with the data in the S - record format from <port no.>. The optional <=text>, if used, is sent to port 2.

VE [<port no.>] [=text]

A.26 Send message

This command sends <text> to port 2.

*** text....**

A.27 Display/set address register

This command displays the address register selected by <register no.> and, as an option, modifies the content with <expression>.

.A<register no., [<expression>]

A.28 Display/set data register

This command displays the data register selected by <register no.>, and as an option, modifies the content with <expression>.

.D<register no.> [<expression>]

A.29 Display/set relative offset register

This command displays the relative offset register selected by <register no.> and, as an option, modifies the content with <expression>.

.R<register no.> [<expression>]

A.30 Display/set program counter

This command displays the program counter and, as an option, modifies the content with <expression>.

.PC [<expression>]

A.31 Display/set status register

This command displays the status register and, as an option, modifies the content with <expression>.

.SR [<expression>]

A.32 Display/set supervisor stack pointer

This command displays the supervisor stack pointer and, as an option, modifies the content with <expression>.

.SS [<expression>]

A.33 Display/set user stack pointer

This command displays the user stack pointer and, as an option, modifies the content with <expression>.

.US [<expression,]

A.34 Abort command

This key aborts any command currently doing console I/O.

(BREAK)

A.35 Delete character

This key deletes the last entered character.

(DEL)

A.36 Redisplay line

This key redisplays the entire line.

(CTRL D)

A.37 Delete character

This key is equivalent to the DEL key.

(CTRL H)

A.38 Suspend output

This key suspends output to the terminal. Depressing any other key resumes the output.

(CTRL W)

A.39 Cancel command line

This key cancels the entire line.

(CTRL X)

A.40 Process command line

This key causes the command to start.

(CR)

APPENDIX B TUTOR MESSAGES

B.1 PRINTER NOT READY
Printer is not properly connected or cannot receive output.

B.2 SYNTAX ERROR
Error in command line

B.3 ERROR
Error

B.4 ILLEGAL INSTRUCTION
Instruction used an illegal op-code during program execution

B.5 ADDR TRAP ERROR
PC on odd address, word access on odd address

B.6 BUS TRAP ERROR
PC on odd address, word access on odd address

B.7 IS NOT A HEX DIGIT
Improper character entered in a field that requires a hexadecimal digit

B.8 DATA DID NOT STORE
Data did not go where intended

B.9 INVALID ADDRESS
Too big (1 in bits 24-31) or odd for .W or .L (1 in bit 0)

B.10 WHAT
Program does not recognize user's entry

B.11 NOT HEX=
Same as IS NOT A HEX DIGIT

B.12 FAILED AT.. WROTE=.. READ=..
Read or write command failure output by BT

B.13 UNDEFINED TRAP 14
Trap function code is not defined

B.14 CHKSUM=
Indicates received checksum is incorrect, correct checksum is given

B.15.TUTOR 1.3 >
TUTOR prompt

B.16 TIMEOUT
Displayed if port 2 does not respond to LO or VE within 10 seconds

B.17 FORMAT=
Displayed by PF command

B.18 CHAR NULL=
Displayed by PF command

B.19 C/R NULL=
Displayed by PF command

B.20 OPTIONS@XXXXXX
Displayed by PF command

B.21 "TRANSPARENT"
EXIT=$01=CTRL A
Display by TM command

B.22 SOFTWARE ABORT
Displayed when ABORT button is press (Not support by SBC68K)

B.23 BREAK
BREAK key has been used (Not support by SBC68K)

B.24 AT BREAKPOINT
Indicates program has stopped at breakpoint

B.25 BREAKPOINT
Displayed by BR command

B.26 PHYSICAL ADDRESS=
Actual address by command

B.27 PC within "DEBUGGER"
Displayed by trace commands

APPENDIX C   DEMONSTRATION

Set up the SBC68K as described in Section 3, and upload the file 'T3.HEX' in the root directory of the supplied disk (please uses ASCII format). Then, key in the following commands:

```
TUTOR  1.3 >LO1

TUTOR  1.3 > GO 1000
PHYSICAL ADDRESS=00001000=HHDO

**** WELCOME TO THE ECB ****

ENTER THE PASSWORD:
00000001  00000002  00000003  00000004  00000005  00000006
00000007  00000008
TUTOR  1.3 >

Note : Password = ECB, to end the program type 'CTRL T'
```