CORSHAM
Corsham Technologies, LLC
www.corshamtech.com
617 Stokes Road, Suite 4-299
Medford, NJ 08055

6809 CPU Board

Introduction

Thank you for buying our 6809 CPU board!

This was a big project, and definitely the most complicated hardware project we've done so far. At the Vintage Computer Festival Midwest in 2014, I had our first SS-50 products on display and received a warm welcome. However, a lot of people said they really wanted a 6809 based board, so this is the result of all those requests.

Is this board vintage? Well, work started in 2014, so technically it is not. However, it uses a design very similar to the original SWTPC 6809 CPU board using parts available at that time. The large RAM and EPROMs are not vintage, particularly the 128K RAM chip. The board is vintage in that it uses the SS-50 bus and can plug into existing systems or work with other boards of that era.

Using older parts has been a problem because some of them have not been made in a long time, so prices are high, conditions of pulled chips are unknown, and we have to test a lot more components to verify they actually work as expected. Fortunately all the chips on this board are available from surplus inventories, but eventually they will be unavailable.

Features

- 6809 running at 2 MHz.
- Baud rate generator provides all standard SS-50/SS-30 clocks.
- One baud rate line can be jumpered for higher speed options.

- 2K or 4K of EPROM. SBUG uses 2K, but the other 2K can be enabled for user extensions.
- Dynamic Address Translation that is fully compatible with SWTPC's scheme.
- 128K of RAM, fixed in banks 0 and 1. Each 8K chunk can be enabled/disabled.
- A16 to A19 available on the SS-50 bus, individually selectable.

Reset and NMI

In the upper left hand corner of the board is a reset pushbutton switch along with jumpers JP1 (RESET) and JP9 (NMI). These two jumpers can be wired to external buttons on the chassis to provide reset and NMI signals to the processor.

Baud Rate Selection

The SS-50 bus used five lines for baud rate clocks, while the SS-50C bus allowed those lines to be used for either those clocks or the extended addressing lines A16 to A19. Our board allows individual jumper selection for each pin using five jumpers located on the lower left hand corner of the board:

Bus Pin	Jumper	SS-50 (6800) Use	SS-50C (6809) Use
46	JP2	110 baud	110 baud or BUSRQ
47	JP13	150 baud	150 baud or A19
48	JP12	300 baud	300 baud or A18
49	JP11	600 baud	600 baud or A17
50	JP10	1200 baud	1200 baud or A16

Note that the baud clocks are actually x16, that is, they are 16 times faster than the indicating baud.

Because we didn't want to tie up all those pins, our board is optimized to use pin 46, normally the 110 baud line, as a VARiable baud rate line. You'll see JP2 allows you to select either BUSRQ or VAR, and jumper J6 allows you to configure this line as 1200, 2400, 4800 or 9600 baud.

Phew, that's a lot of options, and might not be very clear at all, so here is our recommendation on how to set up those six jumpers to five you a desired baud rate and also the full 20 bits of address space:

Jumper	Suggested setting	Result
JP10	A16	Gives 17 bit addressing
JP11	A17	Gives 18 bit addressing
JP12	A18	Gives 19 bit addressing
JP13	A19	Gives 20 bit addressing
JP2	VAR	Makes the 110 line one of four selectable baud rates
JP6 (VAR)	Your choice	Select this to provide your designed console baud rate setting. We use 2400.

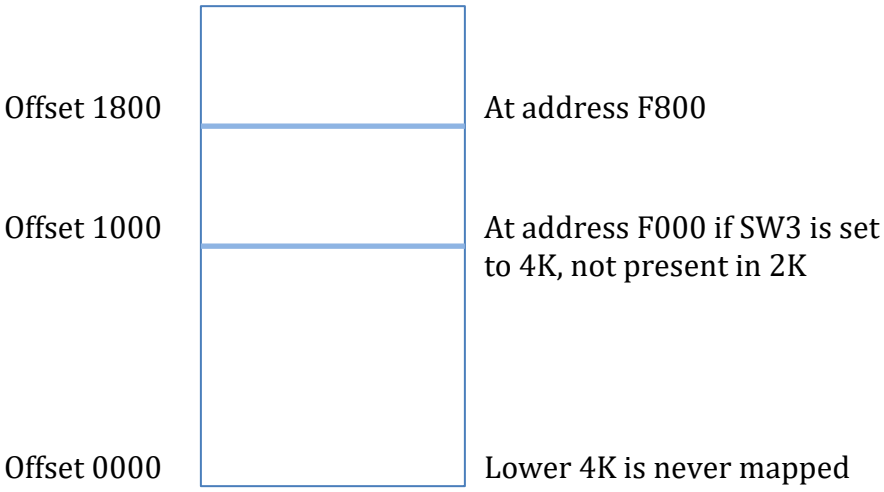
EPROM

The original SWTPC 6809 board had a 2K EPROM with SBUG in the top of memory, from F800 to FFFF, along with additional EPROM sockets. Our board has a single 8K EPROM and the user can select to use just the top 2K for SBUG or the top 4K for an extended monitor or other applications. For vintage computer shows we tend to use an EPROM with a Tiny BASIC interpreter in the lower 2K so people can write programs in BASIC, or include drivers for our SD Card System, but that lower 2K can be used for whatever you want.

Prior to rev 4 boards, switch SW3 (EPROM SIZE) is located along the lower portion of the board and can be set to 2K or 4K. Rev 4 boards have a jumper labeled JP3 which performs the same function. Since this is a little used option, the jumper might not be physically installed.

If you wish to burn your own EPROM, this is where things are in it:

The first 4K is completely unused and is not visible. The upper 4K is mapped to F000-FFFF.



RAM

The board has 128K of RAM available but must be configured via SW1 and SW1 on the upper left hand side of the board.

Banks 0 and 1 are controlled by two 8 position DIP switches. A recommended setting is to turn on the "0XXX" to "CXXX" switches for each bank. Do not turn on the EXXX switch! Please reference the Dynamic Address Translation section for a better explanation of how this works.

Dynamic Address Translation

You don't really need to read this section unless you plan on writing software that uses the extended memory, in which case it's good to understand how SWTPC mapped 1 MB of address space into a processor with only 64K of address space. They did this with Dynamic Address Translation, or DAT. DAT uses 16 RAM locations to map a 16 bit address from the processor into a 20 bit address space.

The top four address lines (A12 to A15) are used as address select lines to 16 bytes of memory. The lower 4 bits of each address map to A12 through A15. The upper 4 bits are A16 to A19.

The top page of memory (FF00 to FFFF) is always mapped to the top 256 bytes of the EPROM. When SBUG starts, it loads up the DAT registers to map 56K of memory from 0000 to DFFF.

Addresses FFF0 to FFFF are the write-only DAT registers. If you read those locations you'll get the contents of EPROM, not the DAT registers. Each register maps one 4K block of memory:

Address	Block	Default value
FFF0	0xxx	0F
FFF1	1xxx	0E
FFF2	2xxx	0D
FFF3	3xxx	0C
FFF4	4xxx	0B
FFF5	5xxx	0A
FFF6	6xxx	09
FFF7	7xxx	08
FFF8	8xxx	07
FFF9	9xxx	06
FFFA	Axxx	05

FFFB	Bxxx	04
FFFC	Cxxx	03
FFFD	Dxxx	02
FFFE	Exxx	01
FFFF	F000	00

That's as clear as mud, right? Okay, the value written into the registers is the inverse of the value for the lower 4 bits, and the true value fro the upper 4 bits. Still not clear, I know, so let's take an example:

FFF0	0xxx	0F

The value 00001111 (binary) is written into the register. When the upper four bits of the address (A12 to A15) are 0000, the entry above is used. The inverse of the lower four bits of DAT register at FFF0 is 0000 (since it has 00001111). So the values for A12 to A15 put onto the bus will be 0000.

So how do we use that? Well, let's assume you want to load and use two programs that are both start at address 0000 hex. You can select bank 0's memory by writing 0F to FFF0 and load the first program.

Now there are multiple ways to put another block of memory at address 0xxx. You can map another block from bank zero, such as moving the memory currently at 8000 down to 0000 by writing 07 hex to FFF0. The inverse of 7 (0111) is 8 (1000), so now when any address with 0000 as the top four bits is selected, the top four bits put onto the address bus will be 1000.

Another way is to use bank 1 so that all of bank 0's memory remains in place. To do this, put the value 0001 in the top 4 bits by writing 1F to FFF0. Now bank 1 will be selected for all 0xxx addresses.

Load up your second program to 0000 and you're set! To select the initial program again, write 0F to FFF0.

Switches SW1 and SW2 allow you to enable/disable a given block of the on-board RAM. Each position is labeled with a bank number, a period, and then a base address such as 0.0xxx and 1.6xxx. Moving the switch to the on position will enable that bank. If you select a bank that is disabled, it won't respond.

Summary of Jumpers and Switches

There are a number of jumpers and switches on the board that change the behavior. While many of them are discussed in other sections of the manual, here is a summary:

Label	Use
JP1	External RESET button connection. Short these two pins together to force a reset.
JP2	Chooses the pin connected to SS-50C bus line 46. It can be set to either VAR to select the baud rate from JP6, or BUSRQ to put the BUSRQ signal onto the bus.
JP3	Rev 4 boards: Selects if the EEPROM covers F000 to FFFF (jumper not installed) or F800 to FFFF (if installed).
JP4	
JP5	
JP6	VAR – This jumper block should have no more than one jumper installed to select the desired baud rate for the VAR line. Currently available baud rates are 1200, 2400, 4800 and 9600. The actual speed of these lines are 16 times faster.
JP7	
JP8	
JP9	External NMI button connection. Short these two pins together to force a non maskable interrupt (NMI).
JP10	Selects the signal present on SS-50C pin 50. Can be the 300 baud (x16) clock or A16.
JP11	Selects the signal present on SS-50C pin 49. Can be the 600 baud (x16) clock or A17.
JP12	Selects the signal present on SS-50C pin 48. Can be the 1200 baud (x16) clock or A18.
JP13	Selects the signal present on SS-50C pin 47. Can be the output from the JP VAR jumper block or A19.
SW1	Used to select which segments in memory bank 1 are active using on-board RAM. The switch labeled 1.EXXX must always remain off.
SW2	Used to select which segments in memory bank 0 are active using on-board RAM. The switch labeled 0.EXXX must always remain off.
SW3	Prior to rev 4 boards, selects the available size of the EPROM. It can be set to either 2K or 4K.

Initial Terminal Settings

8N2. Eight data bits, no parity and two stop bits.

SBUG/EEPROM

The contents of the EPROM with any given board might change over time, but we always include a version of SBUG, which was SWTPC's 6809 debugger and mini command line interface. Documentation for it is easily found on-line. Source code for our version can be found on the www.corshamtech.com website, along with the assembly listing showing which options were built.

TINY BASIC

If your EPROM has a label that has "TINY BASIC" on it, then there are two commands for doing a cold start and warm start to BASIC:

- ! = Cold start BASIC. Do this first.
- @ = Warm start BASIC.

Do a cold start first, then you'll be able to write simple programs using a Tiny BASIC dialect. The BASIC was slightly reworked to make it fit into the bottom 2K of the 4K EPROM on the board. It has one new command ("!") which exits back to SBUG. You can re-enter BASIC with the "@" command, which keeps all variables and your BASIC programs intact.

SD UTILS

More recently, we've been installing our low-level drivers for the SD Card System into the EPROM, along with adding a "B" (Boot) command in SBUG. The B command will load the first sector from drive 0 into memory at C100, then jump to it. We provide a version of 6809 FLEX that can be run directly from the B command.

¡Viva Fiesta!

All of our circuit boards have something unusual on them, and since SWTPC was in San Antonio, it seemed the city would make for some interesting additions. Fortunately, I have a friend who is a native of San Antonio, so I asked her for some ideas or else I'd resort to Googling for something appropriate. She said that ¡Viva Fiesta! is a big festival held in San Antonio each year, so that seemed like a good

choice. I was also excited about this board, so the exclamation points fit into my enthusiasm for this project.

Revision History

Version	Changes
A	Initial Beta.
1	Initial release
2	Not released
3	Most common board shipped for several years
4	Replaced SW3 with JP3. Fixed IC4 on the schematic and board to properly show it is a 74LS244 device. Added more silkscreen information to identify pins on the SS-50 bus.

Errata

REV 1 through REV 3: Incorrect Chip

IC4 is incorrectly identified as a 74LS241 on both the schematic and PC board. The device is actually a 74LS244.

REV 1 Resistor Value Change

Rev 1 boards had an R4 value of 2.2k ohms, but that value is now changed to 680 ohms. The problem manifests itself as odd RAM issues.

Updating Rev 1 PC Boards

Users who buy assembled boards do not need to do any of these steps, as they were applied when we built your board. For someone building from a bare board, these steps will ensure the extended memory works properly. Without the mods, the board functions except that writes to the DAT registers will result in memory corruption. I.e., if you only use the base 64K RAM then none of these mods are needed.

To perform the modifications, you will need a sharp hobby knife to cut some traces, some #30 wire, a stripper for the wire, and a soldering iron. It is recommended that all cuts be made prior to installing any components, as at least one trace is obscured once an IC socket is installed, necessitating several more cuts and jumpers.

Cuts

1. Cut trace on IC6 between pins 30 and 32 (bottom side) close to pin 30 (leave trace from pin 32 to IC7 pins 1 and 28.

2. Cut trace between IC4 pin 19 and IC20 pin 7 (bottom).

3. On top of board, cut short trace from IC4 pin 19 to the via immediately adjacent to it.

4. On bottom, cut trace from IC4 pin 13 to ground.

5. Locate IC10. On top, between pins 6 and 7, and 8 and 9 is a trace. Cut that trace. It does not matter if it is cut above or below IC10. Follow the trace down to a via right above SS-50 pin30. You'll need to know where this pad is for the next step.

Jumpers (install IC sockets before doing these)

1. On the bottom of the board, solder a wire from IC4 pin 7 to the pad located in the previous step. Verify continuity from IC4 pin 7 to SS-50 pin 41.

2. Install jumper on bottom side from IC6 pin 30 to IC14 pin 8.

3. Install jumper from IC4 pin 19 to IC15 pin 1.

4. Install jumper from IC4 pin 13 to pin 14 (adjacent pins).

5. On bottom, install jumper from IC4 pin 1 to IC20 pin 7.

New Part

1. Install a 6.8K resistor on bottom side of board on IC1 between pins 7 and 32.

Parts List

Part	Number	Description
PCB	1	Printed Circuit Board (Corsham Tech)
J1	5	Molex 09-52-3101

JP1, JP9	2	1x2 jumper block
JP2, JP10-13	4	1x3 jumper block
JP6	1	2x4 jumper block
S1	1	4 pin SPST pushbutton
SW1, SW2	2	8 position DIP switch
SW3	1	1 position DIP switch – Not present on Rev 4
C1	1	220uf, 25v electrolytic capacitor
C2-6, C8-11	9	.1 uf disc capacitor
C7, C14	1	22pf
C12, C13	2	1.5uf, 10v electrolytic capacitor
C15, C17	2	.47uf tantalum
C16, C18	2	.01uf
C21	1	100pf
R1, R2, R6, R7	4	1M ¼ watt
R3, R5	2	1K
R4	1	680
R8-10	3	10K
R11	1	220
R12	1	470
R13, R14	2	6.8K
X1	1	1.8432 MHz crystal
X2	1	8 MHZ crystal
LED1	1	3mm LED (usually red, but does not matter)
VR1	1	7805 +5 VDC regulator, TO-220 case
IC1	1	MC68B09 CPU
IC2, IC3, IC4	3	74LS244 (see errata)
IC5, IC11	2	74LS240
IC6	1	628128 128K SRAM
IC7	1	27C64 EPROM
IC8	1	74159
IC9	1	74LS640
IC10	1	74LS10
IC12	1	74LS157
IC13	1	LM556
IC14	1	74LS30
IC15	1	74LS02
IC16	1	MC14411
IC17	1	74LS00
IC18, IC19	2	74LS189
IC20	1	74LS32
IC21	1	74LS74
	7	14 pin IC sockets for IC10, IC13-15, IC17, IC20, IC21
	3	16 pin IC sockets for IC12, IC18, IC19
	6	20 pin IC sockets for IC2-5, IC9, IC11
	2	24 pin wide IC socket for IC16, IC8
	1	28 pin IC sockets for IC7

1	32 pin IC socket for IC6
1	40 pin socket for IC1