Glitch Works 6502 SBC rev 1
GW-6502SBC-1

User's Manual and Assembly Guide

First Revision, 2025-07-23

http://www.glitchwrks.com/

Contents

1 Introduction

The Glitch Works 6502 SBC rev 1 (GW-6502SBC-1) is a single-board computer based on the 6502 central processing unit (CPU). When shipped in kit or assembled form, it includes a Western Design Center W65C02S. In that configuration, it includes the following features:

- WDC W65C02S at 8 MHz (guaranteed operation, overclocking possible)
- 64 KB static RAM, FeRAM compatible
- Up to 512 KB Flash memory in 4K pages, in-board programmable
- WDC W65C51N ACIA serial console
- WDC W65C22N VIA for I/O and timer facilities
- Flash paging and switch-out
- Debounced reset and power supply supervisory circuit
- Glitchbus expansion header

The 6502 SBC rev 1 is a completely self-contained system with a ROM monitor (GWMON-65) and EhBASIC 2.22. It requires only a serial terminal (or terminal emulation software on a PC) for full operation. The large Flash memory included onboard serves as both the boot ROM and storage for EhBASIC.

This SBC supports all 6502 processors; however, the 8 MHz oscillator supplied with kits and assembled boards will not operate with many older processors.

Kits and assembled boards include a WDC W65C51 ACIA. This part is known to have a hardware bug affecting the `TDRE` flag bit in the ACIA status register. All Glitch Works software can detect and work around this bug, and will operate correctly with WDC W65C51 ACIAs, as well as fully functional ACIAs from other manufacturers.

2 Configuration

The 6502 SBC rev 1 uses software configuration for all board options. No switch packs or jumpers need to be set for general operation.

When assembling the 6502 SBC rev 1 with user-supplied parts, non-WDC 6502 processors are supported when processor pin 1 is grounded. There is a solder jumper provided on the solder side of the PCB for this purpose.

2.1 Flash Options

The 6502 SBC rev 1 supports 32-pin Flash chips, ranging in size from 128 KB to 512 KB. The Flash chip is paged into system memory in 4K segments through the ROM segment latch, and can also be disabled entirely. No configuration changes are required to support different Flash sizes; however, all Glitch Works kits ship with 512 KB parts.

Glitch Works parts kits and fully assembled boards use SST 39SF040 Flash chips in DIP-32 packaging. When substituting other manufacturers' parts, ensure that the write and software data protect (SDP) algorithms are the same, and that the chosen part supports 4K sector erase. *Flash chips that require entire device erasure for reprogramming cannot be used as in-system programmable Flash!*

2.2 Glitchbus Expansion

The 6502 SBC rev 1 is expandable through a Glitchbus expansion header. The Glitchbus is a generic 8-bit bus intended to be processor-agnostic. Glitchbus expansion cards are designed to stack above or below the SBC using PC/104 style stacking headers. Alternatively, the 6502 SBC rev 1 may be assembled with a right-angle connector for use in backplanes.

All available Glitchbus boards are compatible with the 6502 SBC rev 1.

3 Assembly

The 6502 SBC rev 1 is designed to be easy to assemble for anyone with moderate soldering ability. It is a medium complexity board and will typically require between one and three hours of assembly time, depending on the skill of the assembler. The following tools will be required:

- Soldering iron, 20-40 W recommended, grounded tip
- Solder, 63/37 leaded solder recommended, Kester "44 Core" or similar
- Diagonal cutters or flush cutters
- Solder braid, solder sucker, or desoldering station, in the event errors are made
- Needle-nose pliers for bending component leads
- 1/4 and 1/8 W resistor lead forms (optional)

This manual does not cover basic soldering technique. If you are new to soldering, we recommend the Adafruit soldering guide and plenty of practice on a piece of protoboard, before beginning assembly of the 6502 SBC rev 1. The Adafruit guide can be found at:

`https://learn.adafruit.com/adafruit-guide-excellent-soldering`

3.1 Assembling the 6502 SBC rev 1

If you purchased a full Glitch Works parts kit, we recommend completing all assembly sections, since extra features can be disabled as needed. If supplying your own parts, you may choose which sections to complete based on the functionality required.

Note that pin 1 is designated with a square pad for all ICs, resistor packs, switches, and most connectors. Except for the LEDs and U3, pin 1 is toward the top of the board, as seen from the front, for all ICs, oscillators, resistor packs, and switches. The component (front) side of the board is the side which contains the white silkscreen legend. It is recommended to install components from shortest to tallest, which makes assembly without an assembly vise or jig easier, assuming the board is flipped over and soldered with the component side resting on a table top.

3.2 Assembly Checklist

- ☐ Verify parts list against kit contents or builder-provided parts
- ☐ Consult the assembly drawing for component locations and values
- ☐ Bend and install all 0.1 µF bypass capacitors (yellow axial bead) in locations marked C in the assembly diagram – position 2 on a 1/8 W lead form
- ☐ Bend both 4.7 kΩ resistors and install in their locations – position 1 on a 1/4 W lead form
- ☐ Install non-socketed DIP ICs at their marked locations: U9, U10, U12
- ☐ Install 20-pin sockets at U9 and U11
- ☐ Install 28-pin sockets at U5, U6, U7
- ☐ Install a 32-pin socket at U4
- ☐ Install 40-pin sockets at U1 and U2
- ☐ Install a 22 µF 10V capacitor at C16, bend leads with needle-nose pliers
- ☐ Install a 1.8432 MHz oscillator at OSC1, `ACIA CLOCK`
- ☐ Install an 8 MHz oscillator at OSC2, `CPU CLOCK`
- ☐ Install a resistor pack at RP1
- ☐ Install a TO-92 DS1233 at U3
- ☐ Install a 10 nF radial ceramic capacitor at C3
- ☐ Install micro tact pushbutton at SW1 `RESET`
- ☐ Hairpin bend both 330 Ω resistors and install in their marked locations
- ☐ Install two LEDs at D1 and D2, align flat on LED with silkscreen
- ☐ Install two 6-pin headers at A1
- ☐ Install a 12-pin header at J1
- ☐ Install a 2-pin Molex KK-100 header section at J2
- ☐ Install a 40-pin Glitchbus expansion header at J3

3.3 Insert Socketed ICs

- ☐ Insert 6522 VIA into socket at U1
- ☐ Insert 6502 CPU into socket at U2
- ☐ Insert 39SF040 Flash chip into socket at U4
- ☐ Insert SRAMs into sockets at U5, U6
- ☐ Insert 6551 ACIA into socket at U7
- ☐ Insert GLUE1 16V8 GAL (labeled `65R1G1`) into socket at U9
- ☐ Insert GLUE2 16V8 GAL (labeled `65R1G2`) into socket at U10

3.4 Connect Serial Mezzanine

The 6502 SBC rev 1 uses a serial mezzanine to convert TTL `TX` and `RX` signals into the desired interface. It is compatible with Glitch Works mezzanines, as well as "FTDI cable" USB adapters. If using a USB adapter, be sure that its pinout matches the signals on the A1 connector before use.

The onboard system serial console supports hardware handshake, and requires that the `*CTS` input be connected. All tested "FTDI cable" USB adapters properly drive this signal. If using a Glitch Works RS-232 mezzanine, be sure your cable includes the `*CTS` line, or loops it back to the `*RTS` output.

If your terminal requires hardware handshake, ensure that your cabling satisfies its signal requirements. A serial light box will help debug potential serial wiring problems. We highly recommend the addition of a light box to your toolkit if you regularly interface RS-232 equipment; our personal favorite is the IQ Technologies SmartCable SC821 Plus.

3.5 Power Supply Considerations

The 6502 SBC rev 1 may be powered by either the 5V power connector (J2), or through the serial mezzanine connector, *but not both at once*. When using a "FTDI cable" style USB adapter, small systems may be powered from USB alone.

Any external power supply should provide 5V and at least 500mA. 5V rail regulation should be within 5% – outside of this range, the DS1233 EconoReset will hold the system in reset, preventing operation.

4 Initial Checkout and Testing

Once the 6502 SBC rev 1 is assembled, double-check all ICs for proper orientation, and check all solder joints for cold joints or solder bridges. Connect a power pigtail provided to a suitable regulated 5V power supply capable of providing at least 500mA, or use USB power. Connect your serial terminal or computer to the console port. Apply power to your system and press `RESET`. The system sign-on message should be printed to the console, and should respond to appropriate user input. `POWER` LED D1 should be on. `USER` LED D2 should be off.

Press `C` at the boot prompt. When asked `Memory size ?`, press `ENTER` to autosize memory:

```
GLITCH WORKS 6502 SBC REV 1

ACIA TDRE BUG PRESENT, USING TIMER 2
[C]OLD/[W]ARM/[M]ONITOR? C

Memory size ?

48127 Bytes free

Enhanced BASIC 2.22

Ready
```

Once EhBASIC is operational, type in the small BASIC program to blink the `USER` LED at D2. The BASIC program may be found in section 5.7.

4.1 Troubleshooting

If your 6502 SBC rev 1 fails to come up, recheck all solder joints for cold joints, bridges, or missed pins – this is by far the most common problem we've observed during assembly workshops. Ensure your serial terminal or terminal emulator software is properly configured and that your cable is wired correctly (a RS-232 light box is very helpful here).

Verify that `*BRESET` is properly strobing during power-up and when pressing the `RESET` button (SW1). Ensure that all socketed chips are installed at the appropriate locations, especially that the 16V8 GAL with the label `65R1G1` is installed in socket U9, and that `65R1G2` is installed in socket U11.

4.2 Repair and Service

If you purchased an assembled 6502 SBC rev 1 from Glitch Works, LLC, your board is warranted to work on arrival. If you have assembled a kit that fails to work, you may return it to Glitch Works, LLC for evaluation, repair, and testing. For questions concerning returns or configuration, please visit `http://www.glitchwrks.com/` and click the "Contact" link.

Do note that while we will attempt to help those who have purchased used boards, there is no warranty extended.

5 Technical Notes

5.1 Flash Segment Control

The 6502 SBC rev 1 includes a Flash segment port, which is implemented using `PORT A` of the 6522 VIA, and is located at address `0xEFF1`. `PORT A` direction is also controlled by `DDRA`, the data direction register, which defaults to input mode on reset. `PORT A` controls which 4K segment of the Flash chip is active, and whether the Flash chip is enabled or disabled. It is pulled up to `0xFF` when `DDRA` is loaded on reset. The bits of the Flash segment port are mapped as follows:

Register Bit	Function	Notes
7	Flash Enabled	Set (Flash enabled) on reset
6	Flash Segment Address, bit 6	Set on reset
5	Flash Segment Address, bit 5	Set on reset
4	Flash Segment Address, bit 4	Set on reset
3	Flash Segment Address, bit 3	Set on reset
2	Flash Segment Address, bit 2	Set on reset
1	Flash Segment Address, bit 1	Set on reset
0	Flash Segment Address, bit 0	Set on reset

Bits 0-6 control the selected 4K segment of Flash available at `0xF000 - 0xFFFF`. These bits are set to `1` on reset so that the last segment of Flash is available for booting. Writing to these bits immediately changes the Flash segment.

Bit 7 enables Flash when set, and disables Flash when cleared. This allows unmapping Flash for use of the full 64 KB of system memory. Writing a `0` to this bit clears it and immediately switches off Flash, writing a `1` to this bit switches Flash in. The selected Flash page is not affected if no other bits are modified. The reset state of bit 7 is `1`.

The Flash segment port provides readback. Reading from address `0xEFF1` returns its current value.

5.2 Default Memory Map

Address Range	Contents
`0x0000 - 0x7FFF`	Static RAM, IC socket U5
`0x8000 - 0xEDFF`	Static RAM, IC socket U6
`0xEEF0 - 0xEEFF`	Glitch Works software variables
`0xEF00 - 0xEFFF`	Glitchbus I/O page (`BIO**M` high)
`0xF000 - 0xFFFF`	Static RAM, IC socket U6, when Flash is disabled
`0xF000 - 0xFFFF`	4K page of Flash, IC socket U4, when Flash is enabled

All onboard I/O devices are addressed within the Glitchbus I/O page, from `0xEF00 - 0xEFFF`.

Onboard memory devices may be overlaid by external devices on the Glitchbus using the `*BMASK` signal.

5.3 Default I/O Map

The 6502 processor uses memory-mapped I/O (MMIO); however, a 256-byte page at `0xEF00 - 0xEFFF` has been reserved as a dedicated I/O page. All onboard I/O devices are addressed within the dedicated I/O page. Accesses within the I/O page drive the Glitchbus signal `BIO**M` high, allowing operation of standard Glitchbus I/O boards with the 6502's MMIO.

I/O Address	R/W	Function
`0x00`	R/W	6551 ACIA (U7) base address
`0xF0`	R/W	6522 VIA (U1) base address

Consult the datasheets for the 6551 ACIA and 6522 VIA for an explanation of their register sets.

5.4 6551 ACIA Serial Console

The console serial interface on the 6502 SBC rev 1 is implemented using a 6551 Asynchronous Communications Interface Adapter (ACIA) addressed at `0xEF00 - 0xEF03`. Serial bitrate is generated using the ACIA's internal bitrate generator, fed with a 1.8432 MHz clock from `OSC1`.

Glitch Works parts kits and assembled boards include a Western Design Center W65C51N ACIA, which is known to have a hardware bug affecting the `TDRE` flag bit in the status register. This bug causes `TDRE` to always be asserted. WDC datasheets have referred to this hardware bug in a number of ways, but currently describe it as a result of the transmit buffer being immediately loaded into the transmitter shift register. *Make no mistake, this is a bug!* It effectively breaks the ability to poll the ACIA status register to determine if the transmit buffer is empty (that is, that transmission of the previously loaded byte is complete). It also breaks interrupt-driven operation, resulting in a continuous stream of interrupts. The WDC datasheet suggests working around the bug with software delay loops, which is not practical for the 6502 SBC rev 1 as the CPU clock frequency may be changed by substitution of the CPU clock oscillator.

The 6502 SBC rev 1 works around the `TDRE` bug using a combination of software detection of the bug, and mitigation with the `TIMER2` element of the 6522 Versatile Interface Adapter (VIA).

In order to detect ACIAs with the `TDRE` bug, the bootstrap routine initializes the ACIA, then immediately writes two `NULL` bytes to the transmitter register. After writing the second `NULL`, the status register is immediately read, and the `TDRE` flag bit is checked. If the flag is clear (logic `0`), the `TDRE` bug is not present and normal use of the `TDRE` flag bit is possible. If the flag is set (logic `1`), the `TDRE` bug is present and must be mitigated.

`TDRE` bug mitigation is accomplished by using `TIMER2` of the 6522 VIA in pulse counting mode. In this mode, `TIMER2` counts down for every negative transition on VIA `PORT B` pin `PB6`. `PB6` is connected to the ACIA's receiver clock output, `RxC`, on ACIA pin 5. `TIMER2` is then loaded with the value `0x00B0`, decimal 176, which is equivalent to 11 transitions of the receiver clock, times 16. This configuration allows using the VIA's interrupt flag register (`IFR`) as a replacement for the `TDRE` flag bit. It may be used in polled or interrupt-driven mode without software wait loops, making this mitigation independent of CPU clock frequency and ACIA bitrate frequency.

The `TDRE` bug detection in the 6502 SBC rev 1 bootstrap stores the results of its test at memory location `0xEEF7`. When this location is zero, `TDRE` works as intended and may be used. When `0xEEF7` is nonzero, the `TDRE` bug is present and must be mitigated. Use of `0xEEF7` is consistent across Glitch Works software, and the address may be read, *but not written,* by user software.

5.5 6522 VIA

Various board functions make use of some of the features of the 6522 Versatile Interface Adapter (VIA) addressed at `0xEFF0 - 0xEFFF`. `PORT A` is used for Flash control. `PORT A` handshake signal `CA2` is connected to the `USER` LED at D2. `TIMER2` is used for 6551 ACIA `TDRE` bug mitigation (see above section, "6551 ACIA Serial Console"), which requires `PORT B` pin `PB6` to be connected to the ACIA's receiver clock output (`RxC`, ACIA pin 5).

All `PORT B` pins are brought out to header `J1`, including the in-use signal `PB6` and handshake lines `CB1` and `CB2`. Pins `PB0` through `PB5`, and pin `PB7` may be used as inputs or outputs as desired. Take care to leave `PB6` configured as an input in `DDRB`.

5.6 Using All Pins on VIA Port B

6522 VIA `PORT B` pin `PB6` is used for 6551 ACIA `TDRE` bug mitigation (see above section, "6551 ACIA Serial Console") when a buggy ACIA is detected by the SBC's bootstrap routine. If the 6502 SBC rev 1 is to be assembled and used with fully functional ACIAs, and buggy ACIA compatibility is not desired, `PB6` may be reclaimed for general use.

To fully utilize `PORT B` when a fully functional 6551 ACIA is installed, cut the trace from ACIA pin 5 to `PB6`. This cut is most easily accomplished on the top side, to the left of U7 pin 5. Be careful not to cut the heavy power trace to the left of it. *Performing this modification will disable `TDRE` bug mitigation permanently, do not perform it if WDC ACIAs will be used with the 6502 SBC rev 1!*

5.7 Controlling the USER LED

The USER LED at D2 is connected to the VIA PORT A handshake line CA2 (VIA pin 39). It is software-controllable through the VIA's Peripheral Control Register (PCR). The PCR also controls the operation of handshake lines CA1, CB1, and CB2. Care must be taken to preserve the state of the control bits related to the other handshake lines when changing CA2.

CA2 is connected to the cathode of D2, a low value on CA2 will illuminate D2, and a high value on CA2 will extinguish D2.

To change the state of CA2 without affecting the other handshake lines:

- Read the PCR at address 0xEFFC
- Bitwise OR the value with 0x0C to set bits 2 and 3
- Clear bit 1 to illuminate USER LED, set bit 1 to extinguish it
- Write the value back to 0xEFFC

The following code snippet illustrates the process in 6502 assembly:

```
;Illuminate the USER LED at D2
USRON:  LDA     $EFFC           ;Get VIA PCR
        ORA     #$0C            ;Set bits 2,3
        AND     #$FD            ;Clear bit 1
        STA     $EFFC           ;Write to VIA PCR
        RTS

;Extinguish the USER LED at D2
USROFF: LDA     $EFFC           ;Get VIA PCR
        ORA     #$0E            ;Set bits 1,2,3
        STA     $EFFC           ;Write to VIA PCR
        RTS
```

Controlling the USER LED in EhBASIC can become a one-liner with compound BASIC statements:

```
REM Illuminate the USER LED at D2
POKE $EFFC,(PEEK($EFFC) OR $0C) AND $FD

REM Extinguish the USER LED at D2
POKE $EFFC,(PEEK($EFFC) OR $0E)
```

This EhBASIC program will flash the USER LED by first setting up the PCR, and then toggling PCR bit 1:

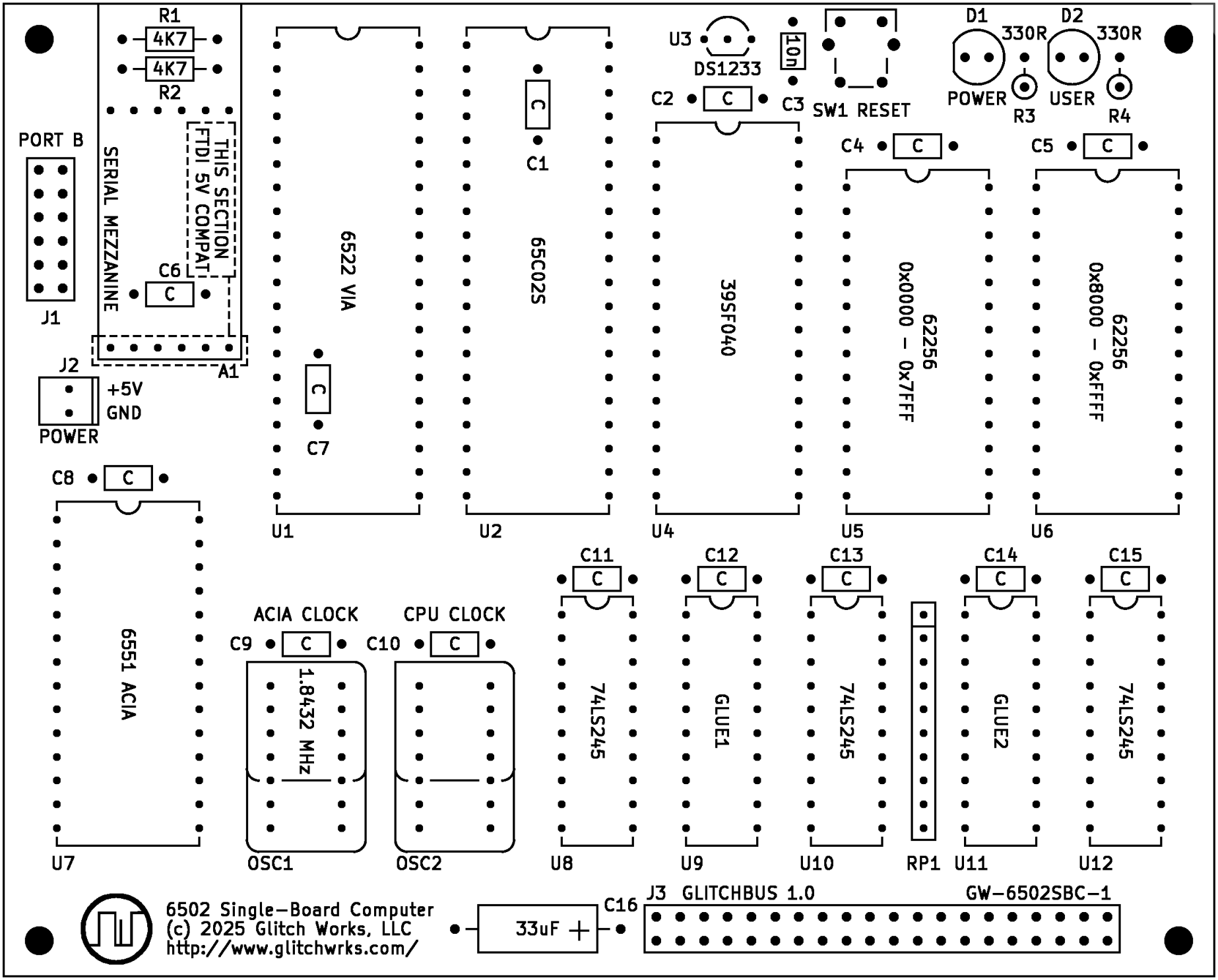
```
10 POKE $EFFC,(PEEK($EFFC) OR $0C)
20 BITCLR $EFFC,1
30 FOR I = 1 TO 1000
40 NEXT I
50 BITSET $EFFC,1
60 FOR I = 1 TO 1000
70 NEXT I
80 GOTO 20
```

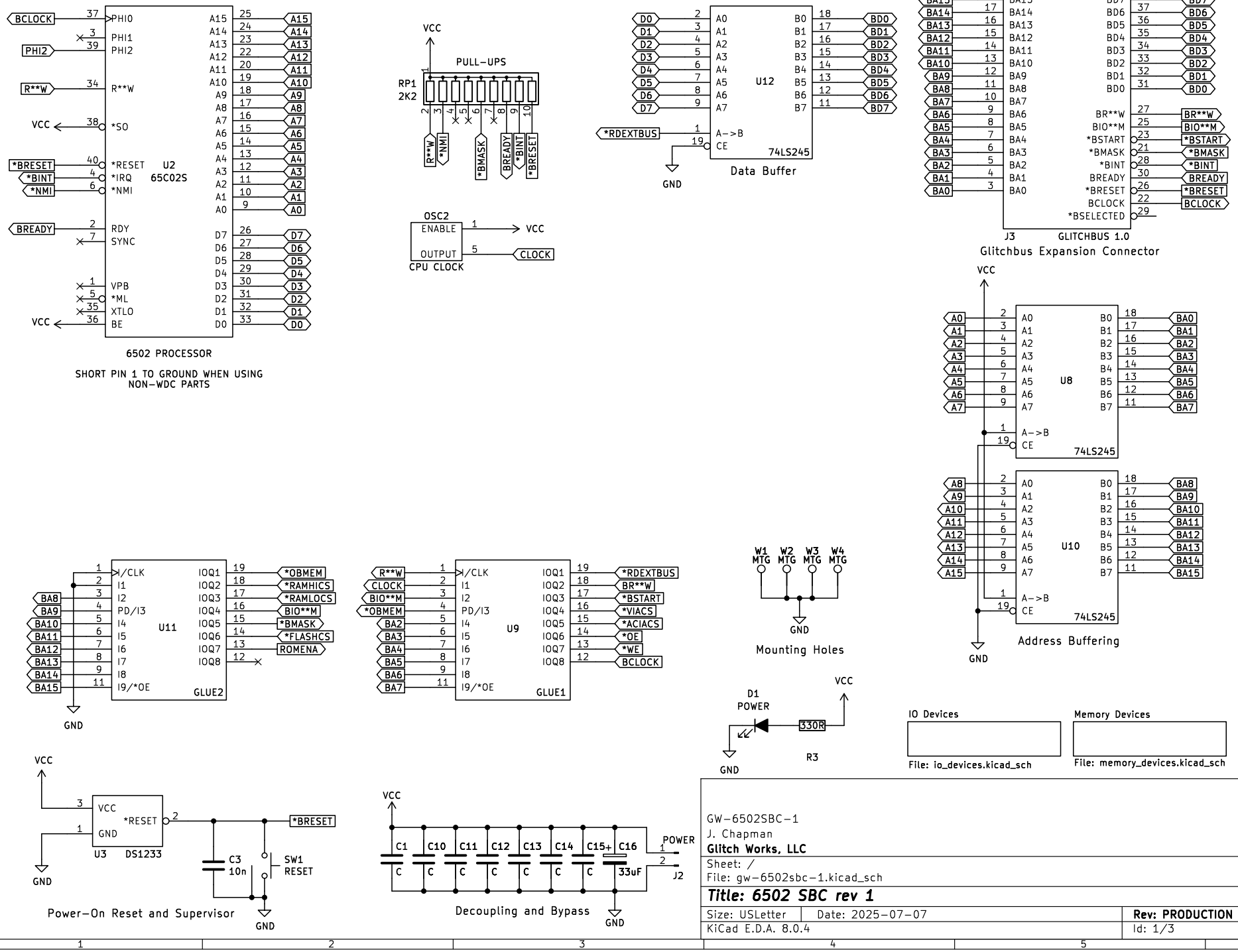
6 Parts List

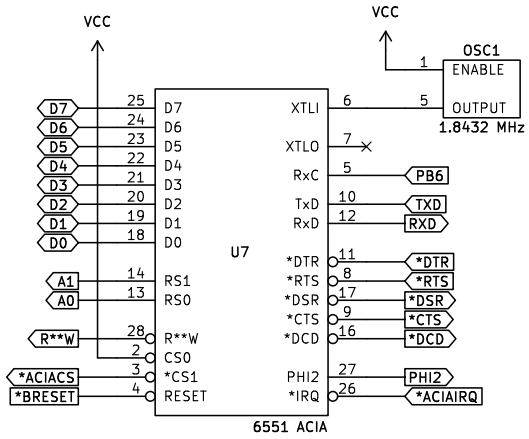
If you purchased a full Glitch Works parts kit, be sure it includes the following:

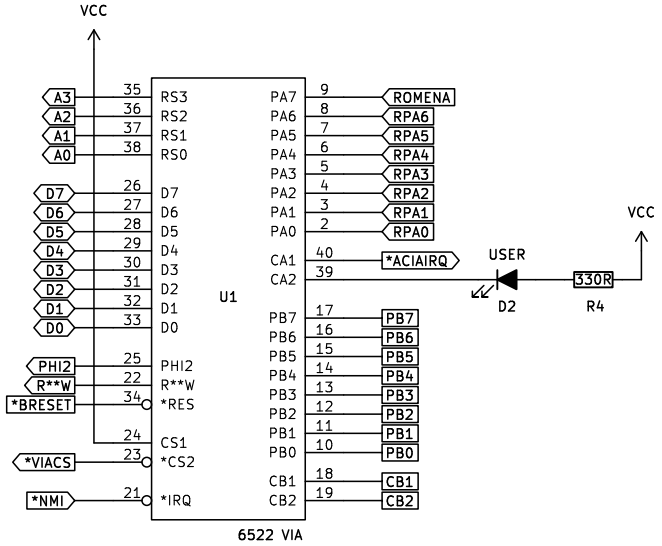
- ☐ 14x 0.01 µF axial ceramic capacitor (yellow bead)
- ☐ 1x 10 nF 16 V radial ceramic capacitor
- ☐ 1x 22 µF 10 V axial tantalum capacitor
- ☐ 2x 330 Ω 1/4 W resistor
- ☐ 2x 4.7 kΩ 1/4 W resistor (see above note)
- ☐ 1x 2.2 kΩ x 9 SIP resistor packs (see note above)
- ☐ 1x 1.8432 MHz crystal oscillator (ACIA CLOCK)
- ☐ 1x 8 MHz crystal oscillator (CPU CLOCK)
- ☐ 1x WDC W65C02S CPU
- ☐ 1x WDC W65C22N Versatile Interface Adapter (VIA)
- ☐ 1x WDC W65C51N Asynchronous Communications Interface Adapter (ACIA)
- ☐ 1x 39SF040 512K x 8 Flash chip, preloaded with GWMON-65 and EhBASIC 2.22
- ☐ 2x JEDEC 62256-type 32K x 8 static RAM
- ☐ 1x 16V8-type GAL, labeled `65R1G1`, preloaded with default `GLUE1` configuration
- ☐ 1x 16V8-type GAL, labeled `65R1G2`, preloaded with default `GLUE2` configuration
- ☐ 3x 74LS245 transceiver
- ☐ 1x DS1233 EconoReset
- ☐ 2x red T-5 LED
- ☐ 1x mini tact pushbutton switch
- ☐ 2x 6-pin header strip
- ☐ 1x 2-position Molex KK-100 header
- ☐ 1x 12-position right angle header
- ☐ 2x 40-pin IC sockets
- ☐ 1x 32-pin IC socket
- ☐ 3x 28-pin IC sockets
- ☐ 2x 20-pin IC sockets

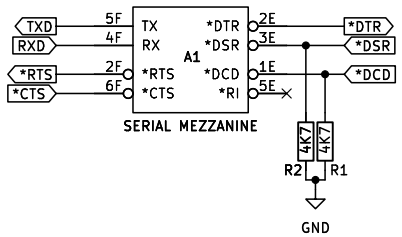
Any compatible 7400 series family logic may be used for the buffers at U8, U10, and U12; for example, 74LS245, 74HCT245, or 74ALS245.

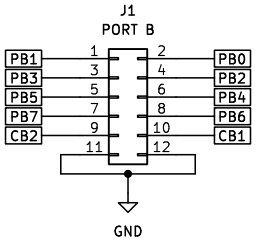

R1
4K7
4K7
R2
PORT B
J1
SERIAL MEZZANINE
THIS SECTION
FTDI 5V COMPAT
C6
C
A1
J2
+5V
GND
POWER
6522 VIA
C7
U1
C1
65C02S
U2
U3
DS1233
10n
C2
C3
SW1 RESET
39SF040
U4
D1
330R
POWER
R3
D2
330R
USER
R4
C4
62256
0x0000 – 0x7FFF
U5
C5
62256
0x8000 – 0xFFFF
U6
C8
6551 ACIA
U7
ACIA CLOCK
CPU CLOCK
C9
C10
1.8432 MHz
OSC1
OSC2
C11
74LS245
U8
C12
GLUE1
U9
C13
74LS245
U10
RP1
C14
GLUE2
U11
C15
74LS245
U12
6502 Single-Board Computer
(c) 2025 Glitch Works, LLC
http://www.glitchwrks.com/
33uF
C16
J3 GLITCHBUS 1.0
GW-6502SBC-1

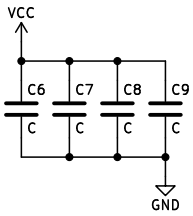

BCLOCK
PHI2
R**W
VCC
*BRESET
*BINT
*NMI
BREADY
VCC
*SO
*RESET
*IRQ
*NMI
RDY
SYNC
VPB
*ML
XTLO
BE
U2
65C02S
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
D7
D6
D5
D4
D3
D2
D1
D0
6502 PROCESSOR
SHORT PIN 1 TO GROUND WHEN USING
NON-WDC PARTS
VCC
PULL-UPS
RP1
2K2
R**W
*NMI
*BMASK
BREADY
*BINT
*BRESET
OSC2
ENABLE
OUTPUT
CPU CLOCK
VCC
CLOCK
D0
D1
D2
D3
D4
D5
D6
D7
*RDEXTBUS
U12
74LS245
Data Buffer
GND
BA15
BA14
BA13
BA12
BA11
BA10
BA9
BA8
BA7
BA6
BA5
BA4
BA3
BA2
BA1
BA0
BD7
BD6
BD5
BD4
BD3
BD2
BD1
BD0
BR**W
BIO**M
*BSTART
*BMASK
*BINT
BREADY
*BRESET
BCLOCK
*BSELECTED
J3
GLITCHBUS 1.0
Glitchbus Expansion Connector
VCC
U8
U10
74LS245
Address Buffering
GND
*OBMEM
*RAMHICS
*RAMLOCS
BIO**M
*BMASK
*FLASHCS
ROMENA
U11
GLUE2
GND
CLOCK
*VIACS
*ACIACS
*OE
*WE
U9
GLUE1
W1 MTG
W2 MTG
W3 MTG
W4 MTG
GND
Mounting Holes
D1
POWER
330R
R3
VCC
GND
IO Devices
File: io_devices.kicad_sch
Memory Devices
File: memory_devices.kicad_sch
U3
DS1233
*RESET
GND
C3
10n
SW1
RESET
Power-On Reset and Supervisor
C1
C10
C11
C12
C13
C14
C15+
C16
33uF
POWER
J2
Decoupling and Bypass
GW-6502SBC-1
J. Chapman
Glitch Works, LLC
Sheet: /
File: gw-6502sbc-1.kicad_sch
Title: 6502 SBC rev 1
Size: USLetter
Date: 2025-07-07
Rev: PRODUCTION
KiCad E.D.A. 8.0.4
Id: 1/3









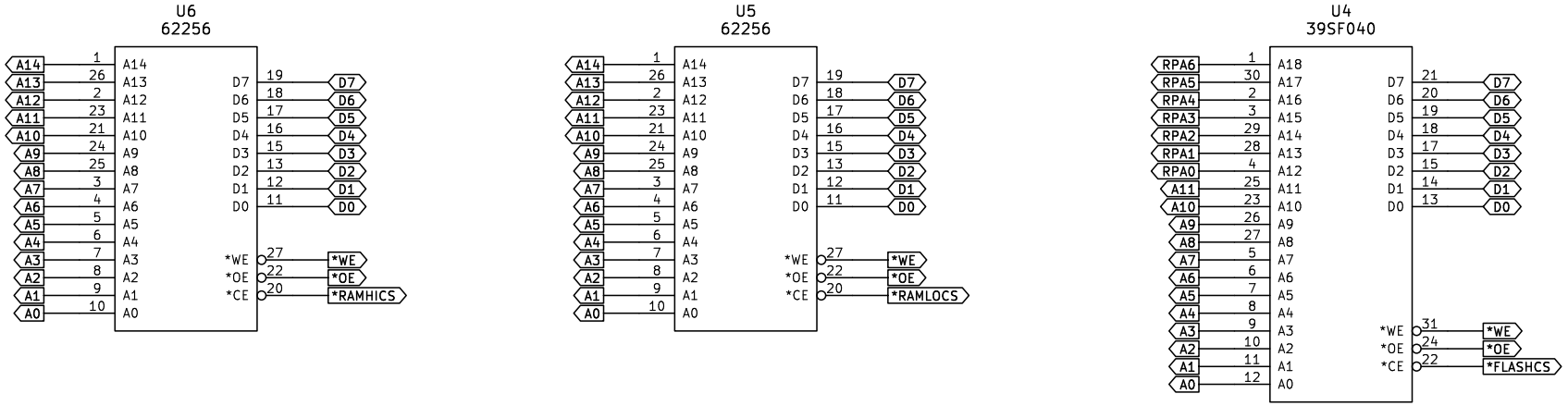


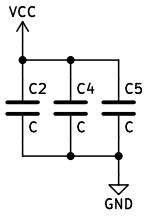
GW-6502SBC-1
J. Chapman
Glitch Works, LLC

Sheet: /IO Devices/
File: io_devices.kicad_sch

Title: 6502 SBC rev 1

Size: USLetter	Date: 2025-07-07	**Rev: PRODUCTION**
KiCad E.D.A. 8.0.4		Id: 2/3





GW-6502SBC-1
J. Chapman
Glitch Works, LLC

Sheet: /Memory Devices/
File: memory_devices.kicad_sch

Title: 6502 SBC rev 1

Size: USLetter	Date: 2025-07-07	**Rev: PRODUCTION**
KiCad E.D.A. 8.0.4		Id: 3/3

```
GAL16V8         ; 6502 SBC rev 1 GLUE1 Logic
6502SBC1        ; PROTOTYPE 1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Pin Declarations
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
RW
CLOCK
BIOM
OBMEM
BA2
BA3
BA4
BA5
BA6
GND

BA7
BCLOCK
WE
OE
ACIACS
VIACS
BSTART
BRW
RDEXTBUS
VCC

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Equations
;
;6522 VIA at 0xF0 - 0xFF I/O page
;6551 ACIA at 0x00 - 0x03 I/O page
;
;OBIO output active when any onboard I/O devices addressed
;
;*WE and *OE are qualified with *BSTART for memory devices
;
;BRW is just buffered RW
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
BSTART          =       /CLOCK
BCLOCK          =       /BSTART
BRW             =       RW
/WE             =       /BSTART * /RW
/OE             =       /BSTART * RW
/VIACS          =       BA7 * BA6 * BA5 * BA4 * BIOM
/ACIACS         =       /BA7 * /BA6 * /BA5 * /BA4 * /BA3 * /BA2 * BIOM
/RDEXTBUS       =       VIACS * ACIACS * OBMEM * RW * /BSTART
```

```
GAL16V8         ; 6502 SBC rev 1 GLUE2 Logic
6502SBC1        ; PROTOTYPE 1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Pin Declarations
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SPARE1
SPARE3
BA8
BA9
BA10
BA11
BA12
BA13
BA14
GND

BA15
SPARE2
ROMENA
FLASHCS
BMASK
BIOM
RAMLOCS
RAMHICS
OBMEM
VCC

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Equations
;
;Low RAM at 0x0000 - 0x7FFF
;High RAM controlled as follows:
;       * 0x8000 - 0xEEFF always active
;       * 0xF000 - 0xFFFF when ROMENA is low
;Flash ROM controlled as follows:
;       * 0xF000 - 0xFFFF when ROMENA is high
;       * Disabled when ROMENA is low
;I/O page at 0xEF00 - 0xEFFF
;
;OBMEM output is low when onboard memory is addressed.
;BMASK low inhibits any onboard memory.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
BIOM            =       BA15 * BA14 * BA13 * /BA12 * BA11 * BA10 * BA9 * BA8
/FLASHCS        =       BA15 * BA14 * BA13 * BA12 * ROMENA * BMASK
/RAMLOCS        =       /BA15 * /BIOM * BMASK
/RAMHICS        =       BA15 * /BIOM * FLASHCS * BMASK
OBMEM           =       FLASHCS * RAMHICS * RAMLOCS
```