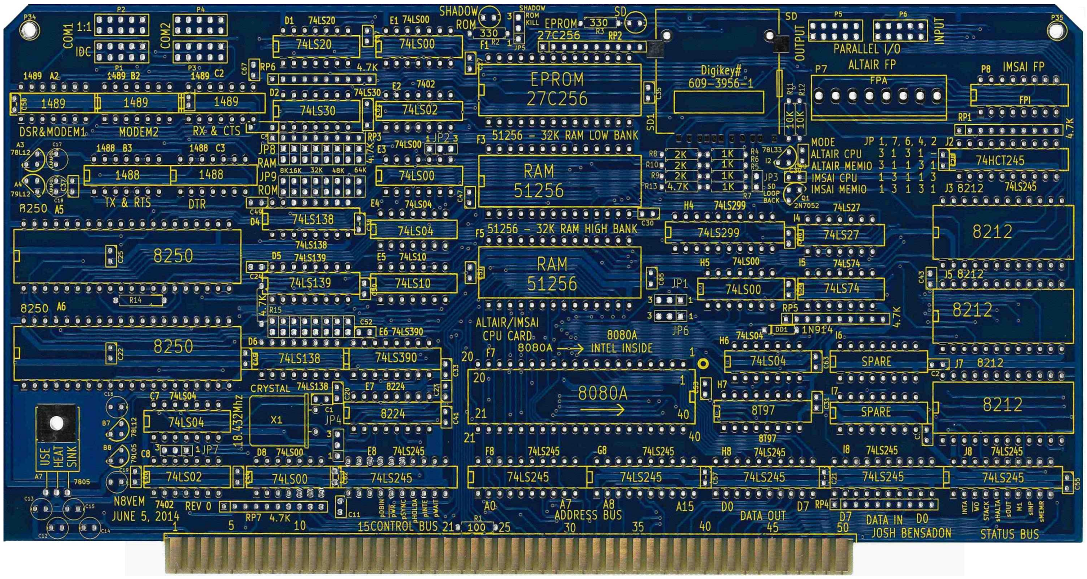


ALTAIR 8800 / IMSAI 8080
Replacement CPU Project

Josh Bensadon
June 5, 2014
Toronto Canada

I cannot thank everyone by name or else I may forget someone, but you know who you are and thank you for your help and friendship.

Special Thanks to everyone at the N8VEM group for their endless help and wonderful ideas that broadened the scope of this project.

Extra Special Thanks to Andrew Lynch for his hard work to build and distribute the boards in the N8VEM realm. Only after doing similar work did I realize how much labour is involved. Also Extra Special Thanks to John Monahan for all his hard work building www.S-100computers.com

Introduction

Thank you for your interest and participation in this project. It drew on several skills and took many hours to bring it to this point. As an electronics hobbyist, I enjoy the roots of my interest as formed by endless hours of studying Popular Electronics and Radio Electronics magazines from the 70's and 80's. Recently, I've been taking a larger interest in the history of personal computers. As such, I would like to include a short history of this project board.

Table of Contents

Short History

In October 2012, I visited the PC Museum of Ontario (www.pcmuseum.ca) and decided to get their IMSAI 8080 computer running. They did not have a disk drive and the installed boards looked mismatched. On the drive home I thought of building an all-in-one CPU board that would work with the least amount of on-site work. By my next visit in April 2013, I had built the first replacement board on one of Andrew Lynch's S-100 Unbuffered Prototyping boards[1]. This first board had 64K of SRAM and a PIC chip to provide a simulated Disk Drive, Console I/O and Boot ROM. An 8Meg Flash memory chip was connected to the PIC's SPI to host 4 possible disks. The PIC's USART provided the Console I/O. Upon reset, the PIC would "spoon feed" the 8080A. It supplied the 8080A CPU with instructions that would save a boot strap in RAM. Then a jump to that location would release this "spoon feed" mode and allow the 8080A execute this boot strap program. With the help of Rich Cini's paper on boot strapping CP/M, I was able to write a copy for this system. On site, I only needed to repair the front panel board and then install this all-in-one CPU board to get a working IMSAI system. After this success, I presented the idea of this board to the N8VEM group and many of your ideas were put on paper and I began working on this version of the board. Many of those ideas are listed in the next chapter that covers features, but one idea that is not listed there was the removal of the PIC chip.

In preparation for this project, I needed to learn KiCad software. I also saw the benefits of Andrew's Buffered Prototyping board, but I didn't like losing all that board real-estate and went to work on a buffered prototyping board that would offer more prototyping area. This side project served as an excellent stepping stone and produced a lot of space to create & debug the prototype.

Although this project did take many hours, I would like to note that it would have taken ten times longer if it was done back in the 70's or 80's because of the lack of all the wonderful tools and helpful people there are today.

[1] I found out about Andrew's prototyping boards from Ebay in the summer of 2012 and since then joined the N8VEM group.

Features

Every effort was put into making this project as feature rich as possible. I cannot take credit for all the ideas as most came from the N8VEM group.

8080A CPU

- Since this board is intended to operate with the front panels of the ALTAIR 8800 and the IMSAI 8080, the original CPU of these computers was chosen.
- This CPU is still widely available on Ebay and other sources.
- Historically, the 8080 processor was a significant milestone for INTEL©. Wikipedia states "The architecture of the 8080 strongly influenced Intel's 8086 CPU architecture, which spawned the x86 family of processors."
- The 8080A hardware resembles Intel's first 8 bit processor the 8008 but its programming model & instructions was kept intact when Intel released the 8085
- Zilog copied and built on this programming model because the 8080A was a success.
- Runs at 2.048Mhz, which is slightly faster than the original CPU's that ran at 2.000Mhz. This was done to get the correct baud rates from the UART chips. Hopefully, this minor over clocking shouldn't affect your software. The original crystal value of 18.000Mhz can be used, only the baud rates will incur some error (which won't be an issue at lower baud rates).

Front Panel Connectors

ALTAIR 8800 – 8 pin Molex

- Original 8 pin Molex connector, 0.156" pin spacing
- Located a little to the left of the original, making it easier to plug in the somewhat short wiring harness.

IMSAI 8080 – 16 pin dip

- Standard 16 pin dip, machine socket suggested.
- Approximately in the same location as the original IMSAI CPU board
- Using the same orientation (pin 1 to the right)

Memory

- 32K SRAM chips which are cheap and available everywhere. With 2 chips on the board, the 8080A can have full access to its whole address range of 64K.
- A 32K EPROM chip adds the ability for storing Boot strap programs and possibly some block(s) of permanent memory.
- Selection of onboard RAM/ROM is done in blocks of 8K. Unselected blocks will permit the 8080A CPU to access those memory ranges from other cards on the S-100 BUS.
- All memory can be disabled by not installing any jumpers

I/O

Addressing

- All of the following I/O devices are mapped to one of eight (8) selectable blocks of thirty two (32) I/O addresses.
- All other I/O addresses permit the 8080A to access I/O from other cards on the S-100 BUS
- All I/O can be disabled by not installing the jumper

Serial Ports

- Uses another popular chip in the 82xx family, the 8250.
- Two Serial Ports provided, COM1 and COM2
- Baud rate software selectable
- RS-232 output drivers and receivers, configured in 3 steps.
    1. Minimal RS-232 (RX/TX + RTS/CTS) for both ports using 1 Driver and 1 Receiver chip
    2. Full DSR/DTR support on both ports plus Full modem (DCD + RI) on COM1 by adding another Driver and Receiver chip
    3. Full modem (DCD + RI) on COM2 by adding the last Receiver chip
- Choice of 10 Pin Header for connection to DB-9 or DB-25 connectors
    1. IDC connector maps the 10 pin header to DB-9 IDC pins that connect to a 9 wire ribbon cable. IDC = Insulation Displacement Connector.
    2. 1:1 connector maps the 10 pin header to commonly found prefabricated DB-9 and DB-25 connectors as used on recent PC's. Pin 1 on the header goes to Pin 1 on the DB-9 for DCD, Pin 2 to Pin 2 and so on.

SPI – Serial Peripheral Interface

- Uses discrete components, no special chips or programmable microcontrollers.
- Runs at the CPU clock of approximately 2Mhz
- Interfaces through voltage level shifters (5V to 3.3V) to an SD Memory Card
- Four lines are used for SPI communication:
    - Clock (Driven by the Master)
    - Master Out Slave In (MOSI)
    - Master In Slave Out (MISO)
    - Device Select (Driven by the Master)

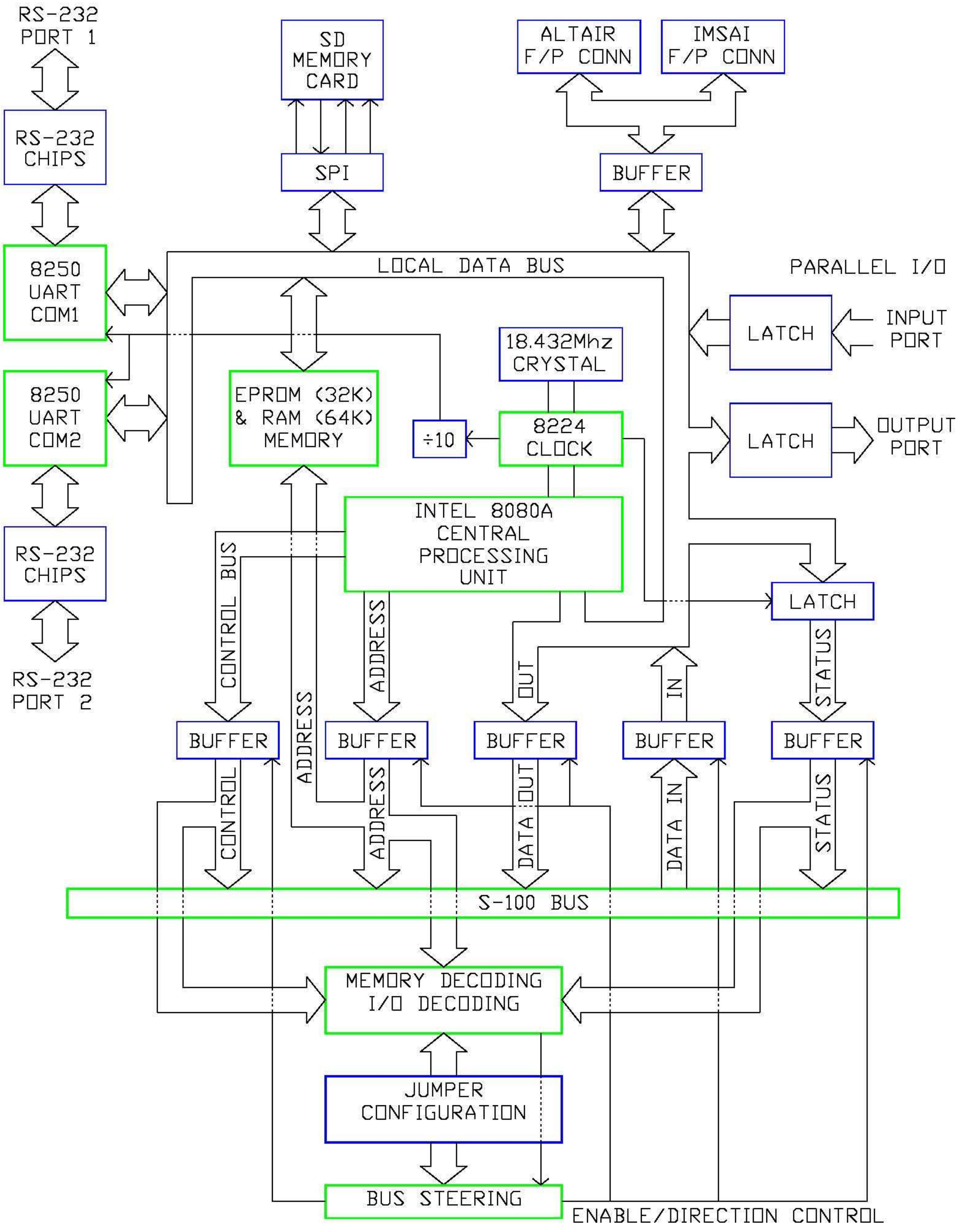
SPI – SD Memory Card

- Socket for a full size SD Memory card
- Recognizes Type 1, 2 and 3 cards
- Allows for mass storage system
- Firmware uses files on the SD Memory card as disk images to boot and access CP/M operating system and files.

Parallel Ports

- Uses the 8212 chip, which is still available and was originally used on the IMSAI 8080 CPU board. The strobe and interrupt pin functions offer a little more than just a simple 8 bit latch.
- One eight (8) bit Input port with the ability to strobe the data in.
- One eight (8) bit Output port with a strobe (interrupt) signal that pulses every time the port is written to.

Block Diagram



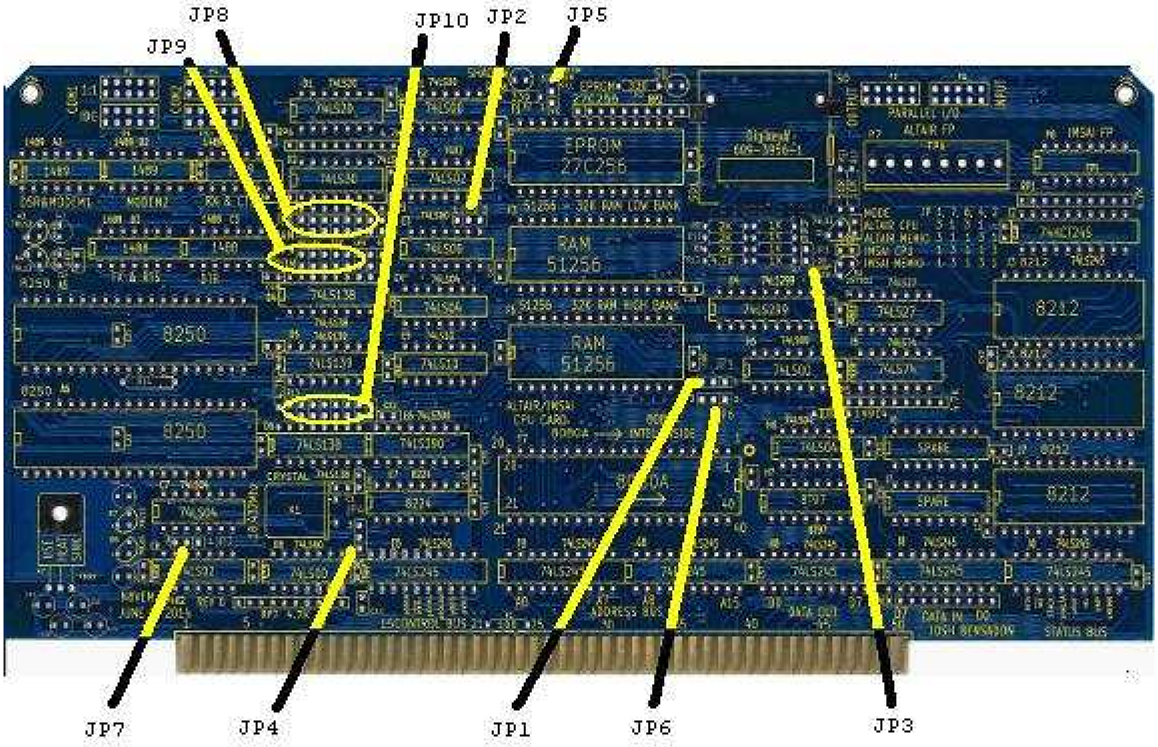
Operation Modes

This board has 4 modes of operation. It can serve as:

1. CPU for the ALTAIR
2. Memory and/or I/O card for the ALTAIR
3. CPU for the IMSAI
4. Memory and/or I/O card for the IMSAI

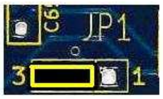
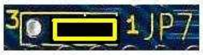
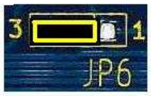
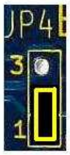
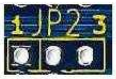
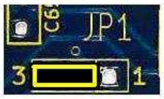
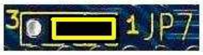
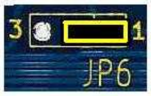
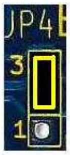
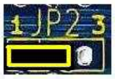
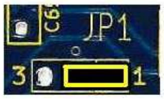
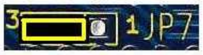
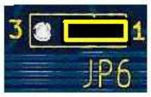
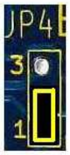
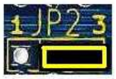
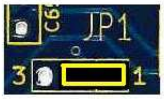
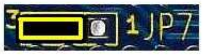
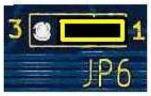
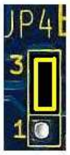
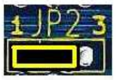
This means the card can either be the CPU card, where Data OUT means it's the originator or as a Memory/I/O card where Data Out means it's the recipient of the data. This is why all the Memory Address and Control line decoding happens from the S-100 side of the buffers.

Jumper Configuration



Board Mode Jumpers - Illustrated

ILLUSTRATED JUMPER CONFIGURATION

	JP1	JP7	JP6	JP4	JP2
ALTAIR CPU	 2-3	 1-2	 2-3	 1-2	 ---
ALTAIR MEM & I/O	 2-3	 1-2	 1-2	 2-3	 1-2
IMSAI CPU	 1-2	 2-3	 1-2	 1-2	 2-3
IMSAI MEM & I/O	 1-2	 2-3	 1-2	 2-3	 1-2

SD Card Diagnostic Jumper – JP3

JP3 is a loop back jumper that connects the MOSI to the MISO (TX to RX) of the SD Memory Card. Do not install.

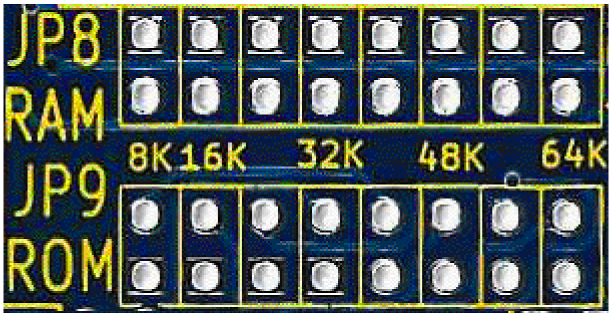
Shadow ROM Kill Jumper – JP5

JP5 selects which source will stop (kill) the Shadow ROM.

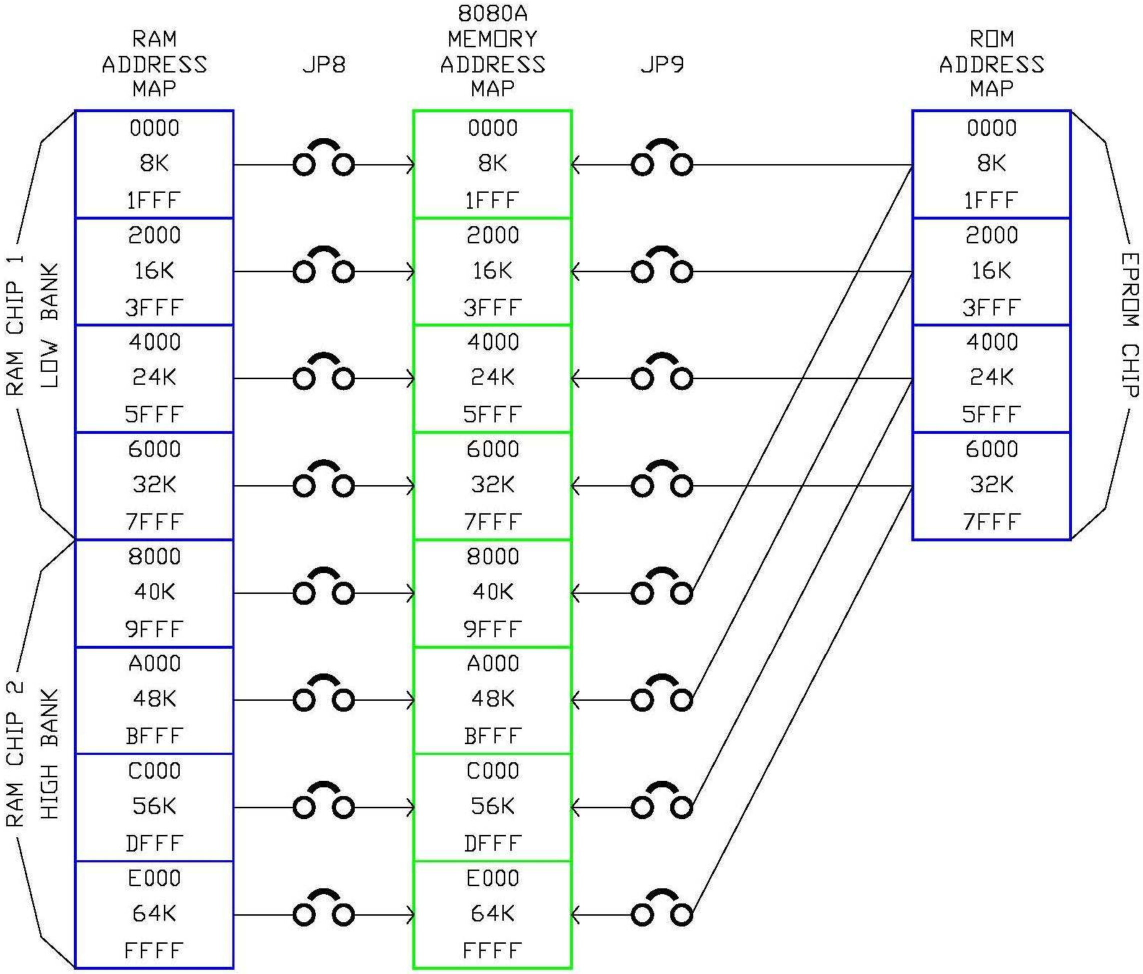
1-2 Shadow ROM killed by first Selection of the SD Memory Card.
Note, upon reset Shadow ROM is re-enabled.

2-3 Shadow ROM selectable by Parallel Output Port, bit 0. 1=Kill.
Note, upon reset, Parallel Port is cleared to 0x00 (Shadow ROM enabled).
With Parallel output control, memory can be “banked switched” between the ROM and RAM.

RAM / ROM Select Jumper – JP8 / JP9

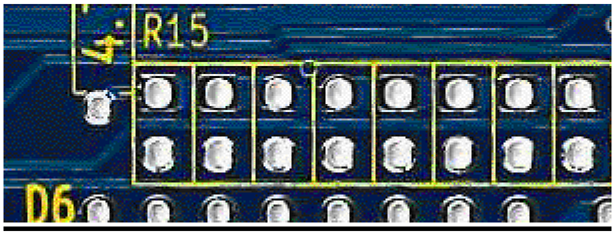


The onboard RAM can be selected in blocks of 8K by installing a jumper in that position. The onboard ROM is also selected in blocks of 8K, but since there is only 32K of ROM, it is duplicated for the upper half of the 8080's 64K address map.



JP8	JP9	Memory Configuration for that 8K block
OUT	OUT	External memory access to S-100 bus
IN	OUT	On board RAM access
OUT	IN	On board ROM access (writes go to RAM, but RAM never accessible)
IN	IN	On board Shadow ROM. After reset, Shadow ROM is enabled. While enabled, Reads are from ROM, Writes go to RAM. When Shadow ROM is disabled, Reads and Writes are to RAM only. See "**Shadow ROM**"

I/O Address Select Jumper – JP10



Not marked on the circuit board, but going from left to right the base I/O address for all the onboard I/O is selectable in steps of 32 (0x20). Please install only one (1) jumper to select the base address as given in the table below. I/O addresses not in the selected range will cause I/O from the S-100 bus. If no jumper is installed, then none of the onboard I/O will be selected and all I/O addresses are from the S-100 bus.

JP10	I/O addressing
None	All I/O mapped to S-100 bus
Left	0x00 to 0x1F maps to onboard I/O, all other addresses are mapped to S-100
Left+1	0x20 to 0x3F maps to onboard I/O, all other addresses are mapped to S-100
Left+2	0x40 to 0x5F maps to onboard I/O, all other addresses are mapped to S-100
Left+3	0x60 to 0x7F maps to onboard I/O, all other addresses are mapped to S-100
Left+4	0x80 to 0x9F maps to onboard I/O, all other addresses are mapped to S-100
Left+5	0xA0 to 0xBF maps to onboard I/O, all other addresses are mapped to S-100
Left+6	0xC0 to 0xDF maps to onboard I/O, all other addresses are mapped to S-100
Right	0xE0 to 0xFE* maps to onboard I/O, all other addresses are mapped to S-100

Note: When selecting the right most jumper, onboard I/O is not accessed when the front panel switches or LED's are read/written to at address 0xFF.

This range of 32 (0x20) I/O addresses is then divided amongst the onboard I/O devices as follows:

Base Address to Base+7	The 8 registers of COM1's 8250 UART chip.
Base+8 to Base+0xF	The 8 registers of COM2's 8250 UART chip.
Base+0x10	SPI Shift Register, Writes trigger SPI transfer, Reads only read the Shift Register containing the received data.
Base+0x11	The lsb controls selecting the SD Card. It also can "Kill" the Shadow ROM based on JP5. 0=Select Card (& Kill).
Base+0x18 Reads Only	Reads the parallel input port from connector P6
Base+0x18 Writes Only	Writes the parallel output port to connector P5

Note: For Base+0x11, only the lsb is latched. The register latching this bit is SET or ON (1) upon a system RESET. This deselects the SD Memory Card and does not affect the Kill Shadow ROM flip/flop. When a "0" is sent to this latch, the SD Memory card is then selected and the Shadow ROM is killed (if configured to do so by JP5). Since it's a flip/flop controlling the Shadow ROM, there is no way to map the Shadow ROM back without a system reset. There is however the possibility to rewire pin 10 of IC E5. First it must be cut away from Pin 9 of IC E5, then it can be jumpered to some other means of resetting this flip/flop. See the schematics for more details.

Shadow ROM

By installing both JP8 and JP9 for the same block, it is possible to start out with ROM then convert that 8080 address space to RAM by “killing” the Shadow ROM. The kill to shadow ROM can be done in 2 ways, see Jumper 5 configuration. The advantage of using the Output port, is that the Shadow ROM can be brought back into the 8080 address space, in effect, this would be like bank switching. For example, you might put some floating point math routines in ROM then only switch in those routines when you need them. By far, the most practical use of Shadow ROM is to allow the system to boot up, copy the ROM code to RAM space as needed, then kill the Shadow ROM. The biggest reason for this approach is for using the CP/M system, which is commonly intended for systems with RAM in the lower memory addresses. However, since the 8080A begins program execution at 0000, it would be useful to have ROM there at 0000 (temporarily) to load CP/M (or a bootstrap) then switch that space to RAM to run CP/M. Note, part of loading CP/M will require this space to be RAM already.

Memory or I/O only mode

When operating in Memory or I/O only mode, the 8080A cpu does not need to be installed, even if it is installed, it will not run because the Hold line will go high to effectively disable the 8080A’s control over the bus. In this mode, all Memory and I/O address selections will continue to work but for an external processor on the S-100 bus. This is accomplished by reversing the Data-IN and Data-OUT buffer direction. The schematic shows this as M or S mode, where M is “master” ie CPU mode and S is “slave” ie Memory or I/O only mode.

Bus Steering

The Master/Slave jumper (JP4) and the ALTAIR/IMSAI jumper (JP6) with the External Memory jumper (JP2) control the Data-IN and Data-OUT bus direction and enabling.

Firmware

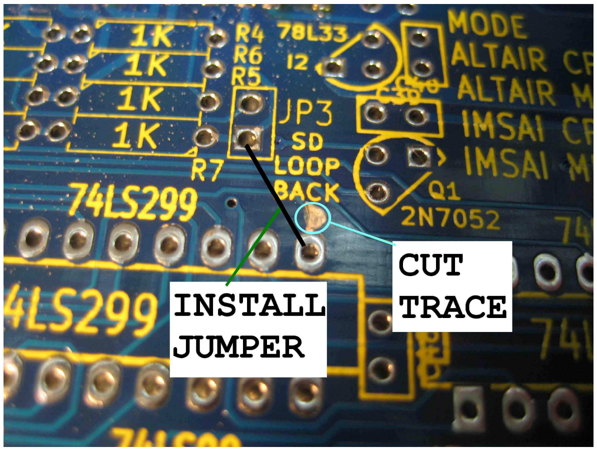
Please find V1 of the firmware on the N8VEM wiki pages.
http://n8vem-sbc.pbworks.com/w/file/83614753/JAIR-Firmware%20V1.zip
Included in the zip file are the source code and all the tools necessary to assemble it. There are 2 parts, the first is a 48 byte boot loader that will relocate the next part into RAM at the high addresses. The next part holds a monitor and the BIOS for CP/M. The two parts are concatenated by the 2nd batch file to produce the EPROM image. Also included, is a file that is 256,256 bytes long, which is the size of an IBM 3740 formated disk (77 tracks of 26 sectors of 128 bytes each). This file has CP/M 2.2 on it and the firmware can load it and run CP/M. There is an issue with selecting another drive, but that will be debugged in the future.
To run this firmware, you must jumper at least the first 8K block of RAM and the two top blocks of 8K RAM. Unless you have external RAM, you will need to install all the RAM jumpers. The first 8K block of ROM should also be jumpered, this will allow the ROM to be in control until after it copies and turns off the SHADOW ROM feature.

There will definitely be improvements made to this firmware, but let it suffice for now.

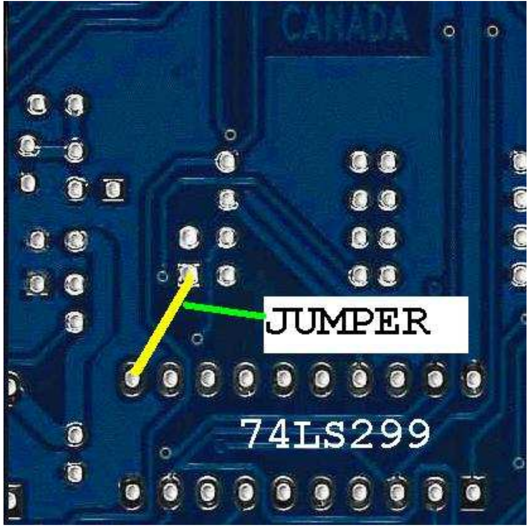
Modifications Required – Engineering Change Orders

ECO #1

Due to propagation issues with the Q1 transistor, it was necessary to direct connect the SD Memory card Data Output directly to the SPI MOSI. The V_{IH} of the 74LS299 is 2V and the Data Output is 3.3V, that will be enough for logic "1". I regret this change requires cutting 1 trace and installing 1 jumper wire. Cut the trace on the component side between Pin 11 of IC H4 and the silk screen text "SD LOOP BACK". Install one jumper wire on the solder side between JP3 and Pin 11. Omit Q1, R7 and R13.



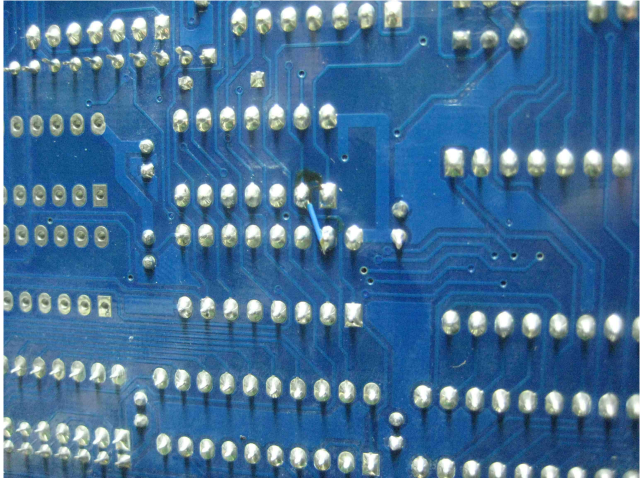
Component Side



Solder Side

ECO #2

Missed connecting pin 15 of IC-H7 (8T97). Install jumper from pin 15 to pin 2 of IC-H6.



ECO #3

Trouble with some CMOS UARTs require the local data bus pull up resistor to be increased to 47K. Change RP2 from 4.7K to 47K resistor network.

Board Assembly

It's always easier to install the components that have the lowest profile first. Here's an approximate list in that order.

1. Cut Trace as per ECO #1
2. SD Memory Card Socket, Digikey # 609-3956-1
3. Discrete resistors and diode.
4. Ceramic capacitors and resistor networks.
5. IC's and/or IC Sockets. Sockets are suggested for the LSI chips but it's your choice. A machine socket would work best for the IMSAI FP connector P8.
6. All Voltage Regulators.
7. Polarized capacitors
8. The ALTAIR FP connector.
9. Crystal. It may not be the highest profile, but it's a delicate component, so it's recommended to go in last.

Note: The 8080A chip is reversed to all the other chips (pin 1 to the right).
The 7805 voltage regulator requires a decent heat sink.

Prior to powering up the board, it's always advisable to look for solder shorts and ohm out the VCC and GND lines across any IC to ensure there isn't a power supply short. Also, ohm out all the other voltage regulator outputs to ground to prevent excessive current and damage in case of shorts. Ohm readings to GND should be anything higher than 100 ohms, it varies depending on your meter since these are semiconductors.
Power up the board without any IC's, measure the voltages to the 8080A socket. You should only get +12V on pin 28 and -5V on pin 11. You will also get +5V on pin 20 and the data lines (pins 3 to 10). If you get the +12 or -5 on any other pin, look for solder shorts and/or micro shorts on the board to correct the problem. This will save you from destroying the 8080A CPU.

John Monahan wrote a very comprehensive guide to building this board:
http://s100computers.com/My%20System%20Pages/8080%20CPU%20Board/8080%20CPU%20Board.htm

BOM – Bill of Materials

Don Caprio wrote a much nicer parts list:
http://n8vem-sbc.pbworks.com/w/page/82798540/Build%20Notes%20and%20Bill-of-materials%20-%20REV%201

Qty	Part	Description	Part Number (Digi-Key)
1	C1	39pf Ceramic Disk Capacitor	
2	C12,C14	22uF Electrolytic Capacitor	
1	C13	100uF Electrolytic Capacitor	
5	C15-C17	3.3uF Tantalum Capacitor	
43	Cxx	0.1uF Ceramic Disk Capacitor	
1	R1	100 ohm Resistor, see notes	
2	R2-R3	330 ohm Resistor	
3	R4-R6	1K ohm Resistor	
3	R8-R10	2K ohm Resistor	
2	R14,R15	4.7K ohm Resistor	
1	RP2	47K ohm Resistor Network 10 pin	4310R-1-473LF-ND
1	RP3	4.7K ohm Resistor Network 9 pin	4609X-101-472LF-ND
5	RP1-RP7	4.7K ohm Resistor Network 10 pin	4610X-1-472LF-ND
4	P1-P6	10 position Pin Header, see notes	A113801-ND
3	JP8-10	16 position Pin Header, see notes	952-1385-ND
7	JP1-7	3 position Pin Header, see notes	609-3461-ND
23	Misc	Mini Jumpers for Pin headers	
1	P8	16 pin socket	ED90034-ND
1	P7	8 pin Molex Pin Header	WM4626-ND
1	SD1	SD Socket	609-3956-1-ND
1	DD1	1N914 Diode	
1	A7	7805, TO-220, 5V Regulator	
1	Misc	Heat Sink for TO-220	HS107-ND
2	A3,B7	78L12, TO-92, 12V Regulator	
1	A4	79L12, TO-92, -12V Regulator	
1	B8	79L05, TO-92, -5V Regulator	
1	I2	78L33, TO-92, 3.3V Regulator	497-7288-ND
3	A2,B2,C2	1489 RS-232 Chips Receivers	
2	B3,C3	1488 RS-232 Chips Drivers	
2	A5,A6	8250 UART	
1	D1	74LS20	
1	D2	74LS30	
2	D4,D6	74LS138	
1	D5	74LS139	
3	C7,E4,H5	74LS04	
2	C8,E2	74LS02	
4	D8,E1,H5,	74LS00	
1	E6	74LS390	
1	E7	8224 Clock Generator	
1	E5	74LS10	

6	E8-J8	74LS245	
1	F1	27C256 32K EPROM	
2	F3,F5	51256 32K Static RAM	
1	F7	Intel 8080A CPU	
1	H4	74LS299	
1	I4	74LS27	
1	I5	74LS74	
1	H7	8T97 or 74LS367	
1	J2	74HCT245	
3	J3,J5,J7	8212, 8-bit I/O Port	
1	X1	18.432Mhz Crystal ECS-184-20-1X	X1036-ND
19		14 Pin IC Sockets	
6		16 Pin IC Sockets	
8		20 Pin IC Sockets	
3		24 Pin IC Sockets, 0.6”	
3		28 Pin IC Sockets, 0.6”	
3		40 Pin IC Sockets, 0.6”	
2		9 Pin DB connector, IDC	S9597-ND
2		10 Pin Header, IDC	MSC10A-ND
1		Ribbon Cable	

Notes:

- Resistors can be 1/8 to 1/4 watt, the 1/4 watt are easier to handle but 1/8 watt are smaller, it’s your choice.
- Only 4 pin headers are needed between P1 to P6, only one header for either 1:1 or IDC needs to be installed. No harm to install both, in which case you will need 6 of these headers.
- Recommend buying some long pin headers and just cutting to lengths needed.
- Optional to add IC sockets where desired, Recommended for all the LSI chips.