tynemouth software

TYNEMOUTH SOFTWARE MINSTREL 4TH BUILD INSTRUCTIONS

PARTS LIST

CAPACITORS –ALL AXIAL, RATED 16V OR HIGHER

2 x 18-22pF (to match crystal) *(usually marked 18/18J, 20/20J or 22/22J)*
2 x 47nF axial *(usually marked 47n or 473)*
18 x 100nF axial *(usually marked 100n or 104)*
1 x 1uF axial *(usually marked 1u or 105)*
1 x 22µF *(axial electrolytic rated 25V)*

RESISTORS – ALL ¼W 5% OR BETTER (4 BAND RESISTOR COLOUR CODES SHOWN)

2 x 75Ω 
1 x 1KΩ
1 x 2.2KΩ 
1 x 4.7KΩ
16 x 10KΩ 
1 x 220KΩ

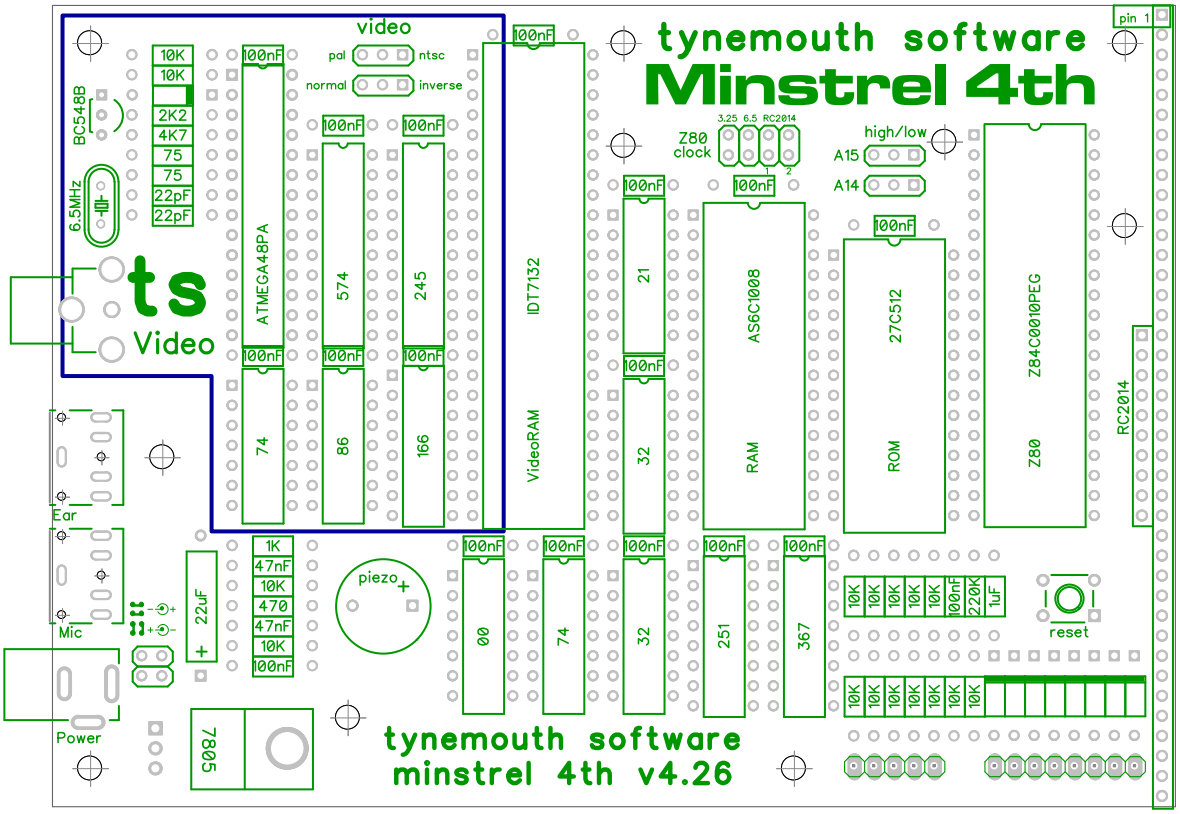
SEMICONDUCTORS - NEW TEXAS INSTRUMENTS 74HC SERIES CHIPS RECOMMENDED.

9 x 1N4148 diode
1 x BC548B or similar NPN transistor
1 x 74HC00
1 x 74HC21
2 x 74HC32
2 x 74HC74
1 x 74HC86
1 x 74HC166
1 x 74HC245
1 x 74HC251
1 x 74HC367
1 x 74HC574
1 x Z80 CPU / NEC D780 / Z84C0008PEG / Z84C0010PEG (*8MHz or higher rated*)
1 x 128K SRAM (AS6C1008 / 621024)
1 x 2K Dual Port RAM (IDT7132)
1 x 27C64 – 27C512 EPROM
1 x ATmega48PA (*pre-programmed*)
1 x 7805 or 7805 switching replacement (*rated at least 100mA, more if adding RC2014 modules*)
1 x 6.5 MHz Crystal (*HC-49/U package*)

CONNECTORS / SWITCHES / SOUNDER

1 x 5 way, 1 x 8 way 0.1" connector to suit keyboard (right angled socket)
2 x Stereo 3.5mm Jack *(e.g. CUI SJ1-3525N – Digi-Key SJ1-3525N)*
1 x Phono jack *(e.g. CUI RCJ-011 – Digi-Key CP-1400-ND)*
1 x 2.1 mm DC Jack
1 x miniature tactile switch 6x6mm *(optional, e.g. Diptronics DTS-61N)*
1 x Piezo AC transducer *(not a buzzer or any sounder that has internal circuitry)*
2 x 2 way and 5 x 3 way headers with jumper *(optional or fit wire links)*
1 x 40 way connection for RC2014 bus (*male or female as required, optional*)
1 x 10 way connection for RC2014 enhanced bus (*male or female as required, optional*)
1 x 48pin, 1 x 40pin, 1 x 32pin, 1 x 28pin (600mil), 1 x 28pin (300mil) IC sockets *(turned pin recommended*)
7 x 14pin, 3 x 16pin, 2 x 20pin (300mil) IC sockets *(optional, turned pin recommended)*

COMPONENT PLACEMENT



JUMPERS

Jumpers or link wires need to be fitted to the following positions.

Jumper		
Video Standard	PAL 50Hz refresh rate	NTSC 60Hz refresh rate
Normal / Inverse	Normal (black on white)	Inverse (white on black)

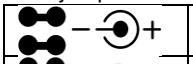
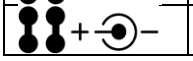
The Z80 clock is set using four clock jumpers. Setting a clock and an RC2014 clock also feeds that clock to the RC2014 bus

Clock Jumper				
Z80 Clock	3.25 MHz	6.5MHz	RC2014 Clock 1	RC2014 Clock 2

The ROM image is selected using the A14 and A15 jumpers that control the voltage on pins 27 and 1 of the ROM chip.

A15	A14	ROM	Use with clock
		*Reserved for future upgrade*	-
		*Reserved for future upgrade*	-
		Patched Forth	6.5MHz
		Original Forth	3.25MHz

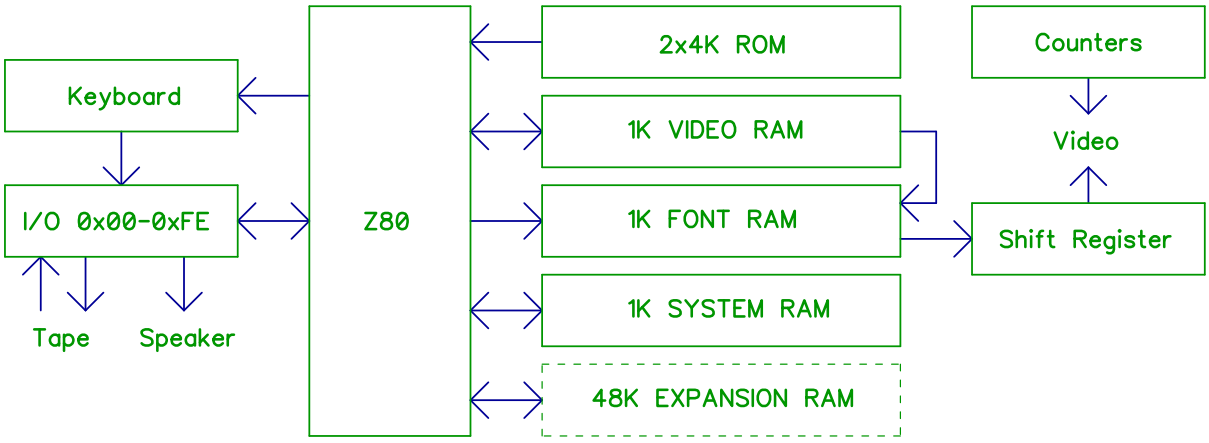
The jumpers near the power connector set the polarity of the power input

	Centre negative, used by optional TFW8b.com supply (also Spectrum and Commodore 16 computers)
	Centre positive, used by pretty much every other power supply

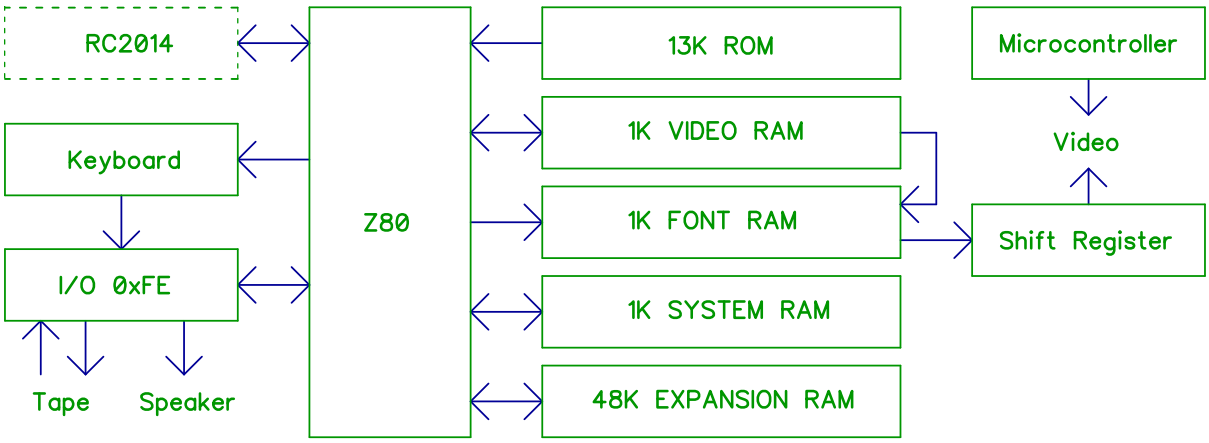
Power requirements are 9V DC. Current consumption is around 65mA for the board. More if you plan to use the RC2014 bus, in which case a heatsink may be required on the 7805 regulator. You can also power the board from the RC2014 bus, in which case you can omit the DC jack, polarity jumpers, 22uF capacitor and 7805 regulator.

BLOCK DIAGRAM

The original Jupiter Ace can be reduced to this block diagram:



There are two 4K ROMs giving an 8K block of ROM, and three 1K blocks of RAM. There is 1K which is the screen RAM, one byte per character, 32x24 characters taking 768 bytes (the remainder being used as scratch RAM by the system). Another 1K RAM is used to store the font. This is write only, and is initialised at boot time. There are 128 8x8 characters, which can be redefined by the user to create pixel graphics. The font pixels are clocked out of a shift register to generate the video signal, combined with counters to generate video timing. A third 1K block is the main system RAM. This is mirrored three times, and the two video RAM blocks are mirrored once each. That leaves up to 48K for expansion RAM. I/O is to tape, speaker and keyboard, at any even I/O address.



The Minstrel 4th implements a compatible architecture, but fills up gaps in the memory map to give more ROM, a total of 13K being available. All 48K of expansion RAM is present, giving 49K of system RAM, completely filling the 64K address range of the Z80. The I/O address is fully decoded, so only occupies a single address - 0xFE - the rest is available for RC2014 expansion bus. The counters and decoding logic of the video signal is replaced by a simple microcontroller. The extra control that gives over timing has allowed PAL / NTSC 50/60Hz video timing, and a back porch section to be added to the video signal which allows black on white as well as the original white on black video.

MEMORY MAP

Expanding that in more detail, the memory map is as follows. The ROM is a single 16K image, accessible as one 8K block (0000-1FFF) and a 5K block (2800-3BFF). The remaining 3K is not accessible as it is hidden behind blocks of RAM.

<table>
<tr><th></th><th colspan="2">Minstrel 4th Memory Map</th><th>ROM</th><th>RAM</th></tr>
<tr><th>Address Range</th><th>Read</th><th>Write</th><th>16K (one of four)</th><th>64K (half of chip)</th></tr>
<tr><td>0000-03FF</td><td rowspan="8">ROM (8K)</td><td rowspan="8">-</td><td rowspan="8">8K</td><td rowspan="11">Not Used (15K)</td></tr>
<tr><td>0400-07FF</td></tr>
<tr><td>0800-0BFF</td></tr>
<tr><td>0C00-0FFF</td></tr>
<tr><td>1000-13FF</td></tr>
<tr><td>1400-17FF</td></tr>
<tr><td>1800-1BFF</td></tr>
<tr><td>1C00-1FFF</td></tr>
<tr><td>2000-23FF</td><td colspan="2">Video RAM Mirror</td><td rowspan="2">Hidden (2K)</td></tr>
<tr><td>2400-27FF</td><td colspan="2">Video RAM (1K)</td></tr>
<tr><td>2800-2BFF</td><td rowspan="5">ROM (5K)</td><td>Font RAM Mirror</td><td rowspan="5">5K</td></tr>
<tr><td>2C00-2FFF</td><td>Font RAM (1K)</td></tr>
<tr><td>3000-33FF</td><td rowspan="3">-</td></tr>
<tr><td>3400-37FF</td></tr>
<tr><td>3800-3BFF</td></tr>
<tr><td>3C00-3FFF</td><td colspan="2">System RAM (1K)</td><td>Hidden (1K)</td><td rowspan="2">49K</td></tr>
<tr><td>4000-FFFF</td><td colspan="2">Expansion RAM (48K)</td><td>-</td></tr>
</table>

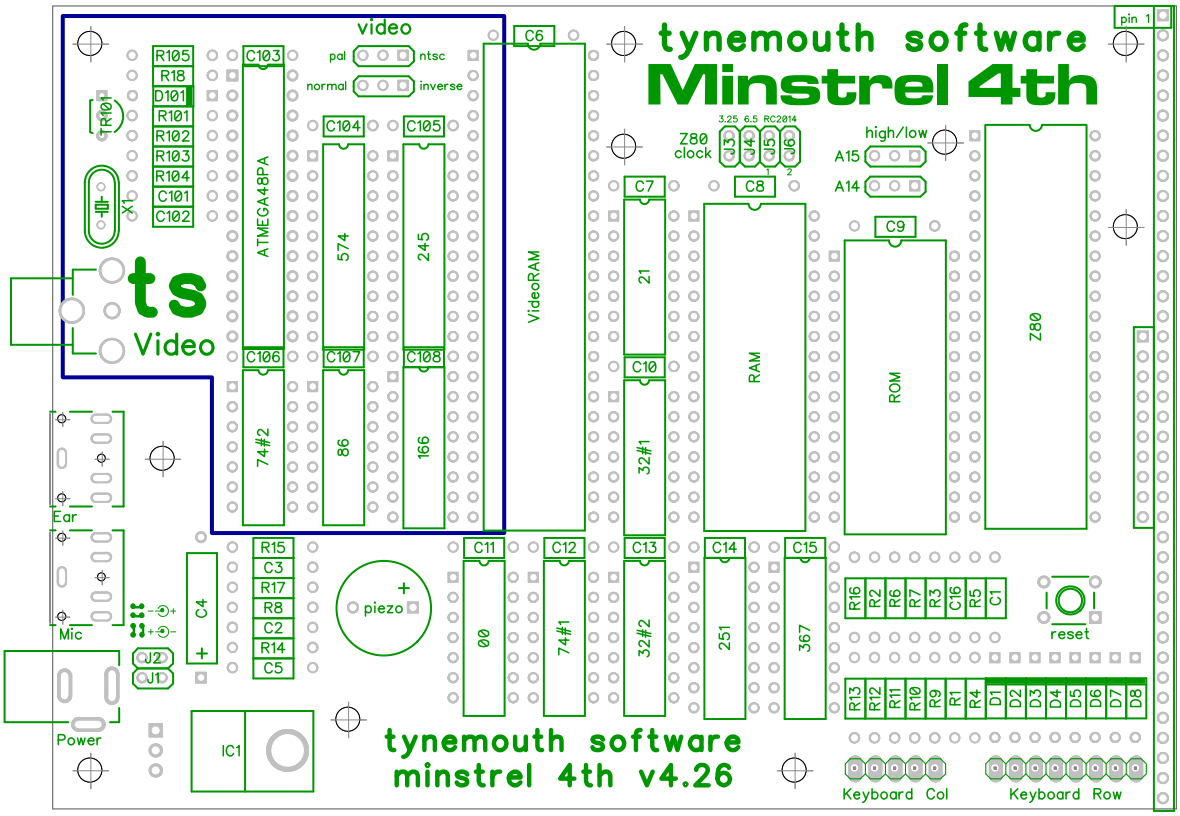
The Minstrel 4th implements the block diagram, rather than the specifics of the Jupiter Ace. Originally there were three pairs of 2114 SRAM chips, and the RAM access was multiplexed between the CPU and video circuitry with lots of 1K resistors and 74LS367 buffers. The mirrors of the video RAM and font RAM were used to give one option to access the RAM with CPU priority (snow on screen), and the other with video priority (CPU halted).

The Minstrel 4th has a single 128K RAM chip, 49K of which is used (the rest is not accessible - some banking could be added if it there was use for the extra RAM?). The video RAM is simplified down to a single 2K dual port RAM chip. Both sides can access the video RAM at the same time and there is only a conflict if both access the same address at the same time, in which case the Z80 is halted for a few cycles or the screen blanked for a few pixels whilst there is conflict.

The video timing circuitry was composed of a lot of counters and decoding logic dividing down the 6.5MHz clock to generate video sync and timing, and a different PCB was produced for PAL and NTSC timings. To simplify things, this has been replaced by a small microcontroller which does effectively the same thing. The final video is still generated by clocking pixels out of the font RAM via a 74LS166 shift register.

Earlier versions of the Minstrel 4th used two 1K dual port RAM chips, this was reduced to a single 2K dual port RAM chip, with the addition of a latch to store the character address to feed back into the font RAM.

COMPONENT DESIGNATORS

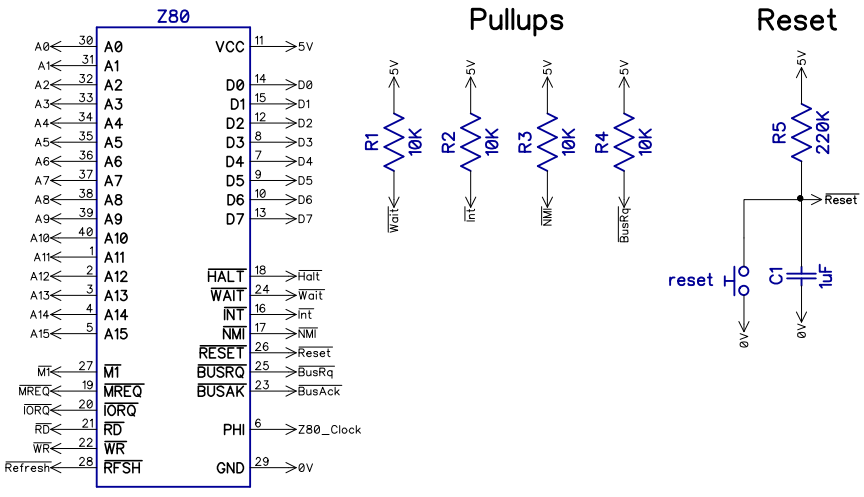


SCHEMATIC

The Schematic has been split into functional grounps for clarity.

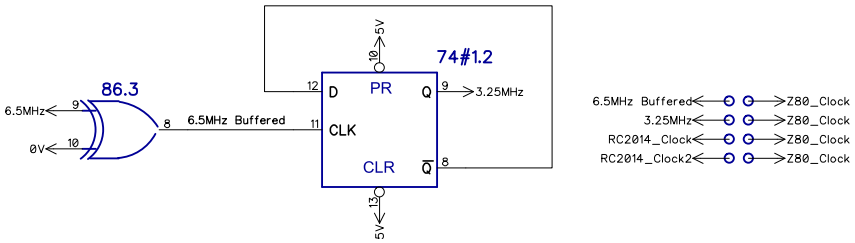
Z80

The heart of the Minstrel 4th is a Z80 processor. The control inputs are pulled up with 10K resistors. A power on reset pulse is generated from a resistor / capacitor circuit with a reset switch shorting out the capacitor to again generate a reset pulse.



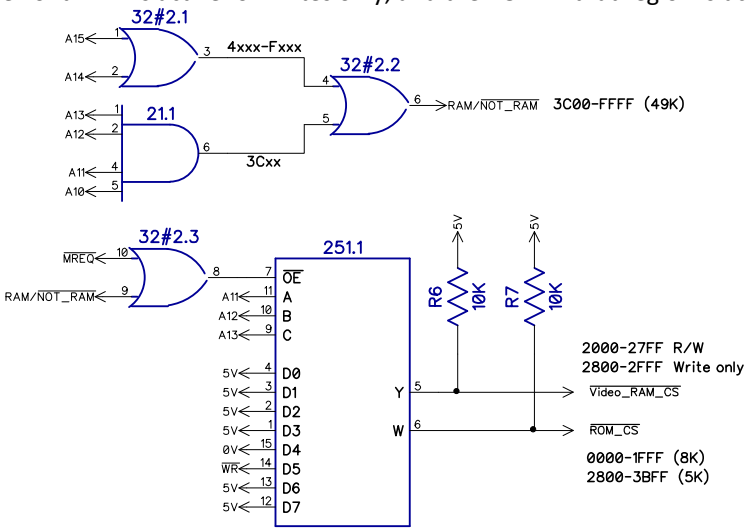
Z80 CLOCK

The clock for the Z80 can be fed from multiple sources. The 6.5MHz clock comes from the video circuitry; this is buffered and divided to give the 6.5MHz and 3.25MHz options respectively. It can also be fed to or from either of the two clocks on the RC2014 bus (Clock 2 only available on the enhanced bus)



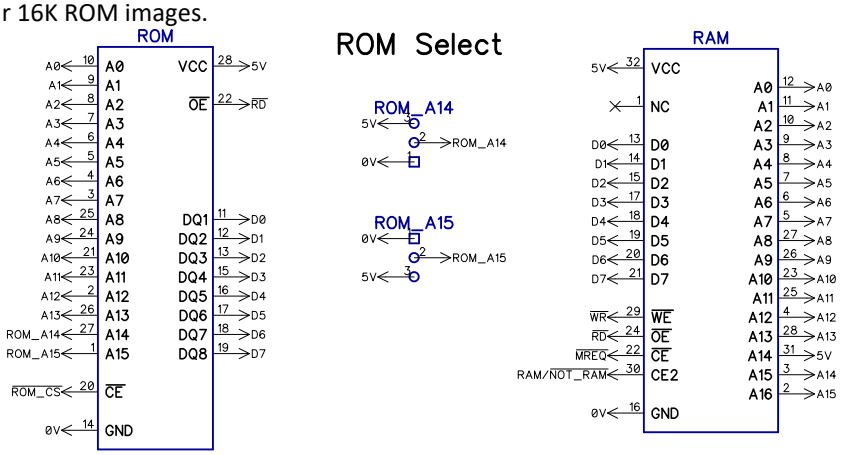
ADDRESS DECODING

The address decoding of the ROM and RAM starts with a few logic gates generating a signal which is high when in the range of the RAM (0x3C00-3FFF). That same signal is low when it is not in the range of RAM, and that is used to enable a decoder that generates the Video RAM and ROM select lines. Part of that range is controlled by the /WR line as the Font RAM is active for writes only, and the ROM in that region is active for reads only.



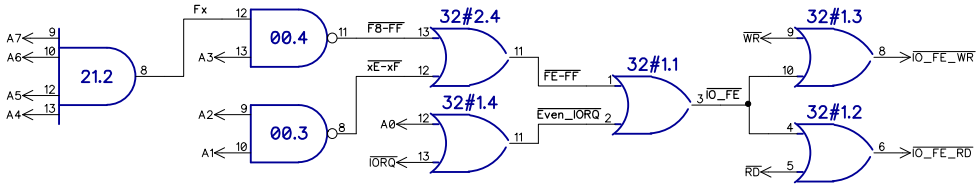
ROM AND RAM

A single ROM and a single RAM chip cover the majority of the address range. One address line on the RAM chip is tied high which reduces the usable capacity to 64K. The upper two address lines of the ROM chip are used to select one of four 16K ROM images.



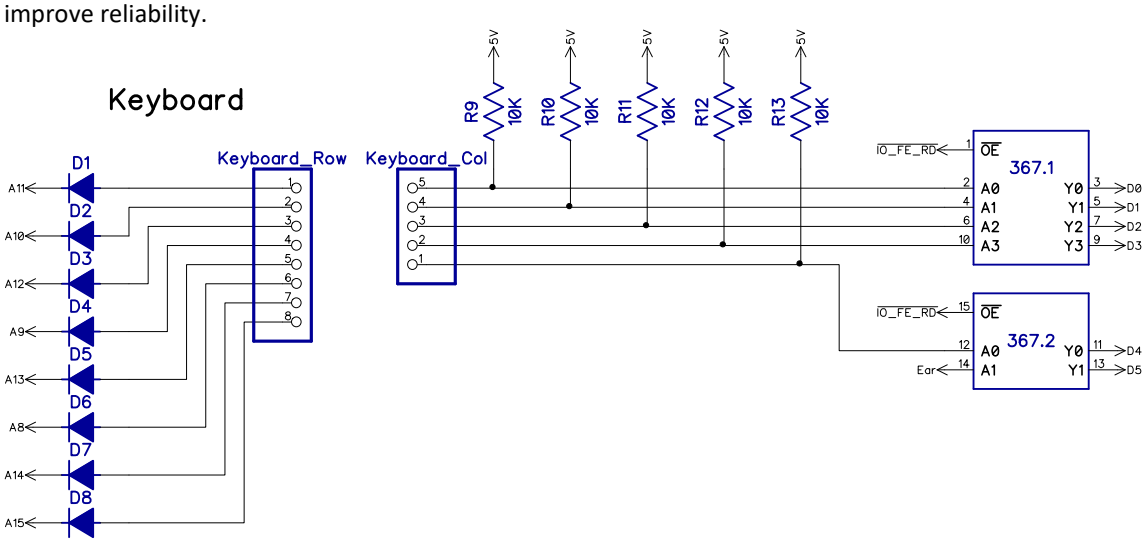
IO DECODING

The IO address of 0xFE is fully decoded by quite a lot of logic gates. This is combined with the read and write signals to generate signals which go low when the 0xFE address is read or written.



KEYBOARD

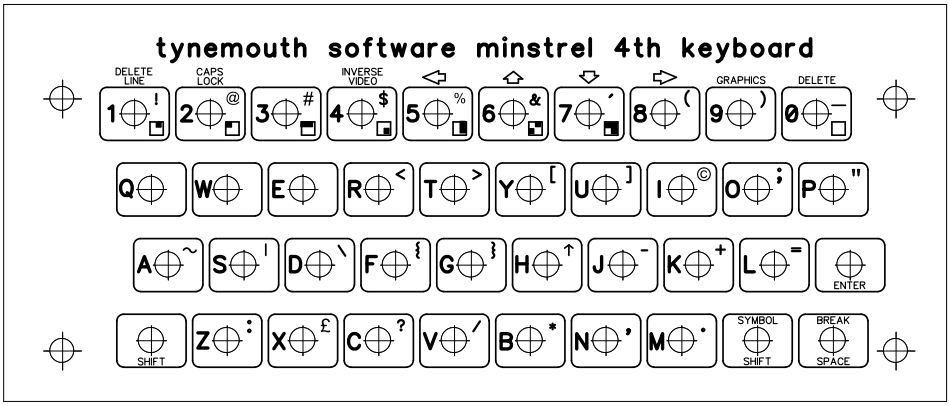
The input port is only six bits wide. Bits 0-4 are the keyboard column reads. The rows are generated from the upper address lines. The pull up resistors were not present on the Jupiter Ace, but have been added to improve reliability.



The keyboard is mapped as follows, this is very similar to the ZX80/ZX81, other than most of the bottom row is shifted one character to the right, at least logically. The Symbol Shift key is physically located between M and Space, but logically it sits between Shift and Z in the keyboard matrix.

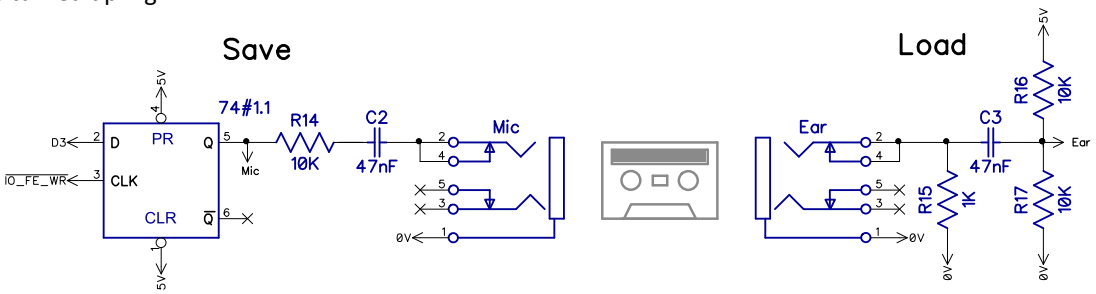
	Col 5	Col 4	Col 3	Col 2	Col 1
Row 1	1	2	3	4	5
Row 2	Q	W	E	R	T
Row 4	A	S	D	F	G
Row 6	SHIFT	SYMBOL SHIFT	Z	X	C

	Col 1	Col 2	Col 3	Col 4	Col 5
Row 3	6	7	8	9	0
Row 5	Y	U	I	O	P
Row 7	H	J	K	L	ENTER
Row 8	V	B	N	M	SPACE



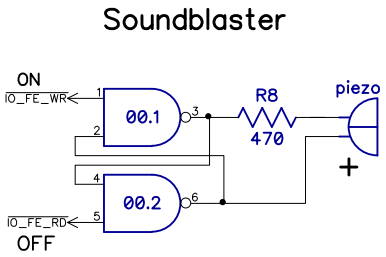
CASSETTE IO

The 6th bit of the input port is the Ear signal from cassette port, with some filtering to clean up the signal. There is no amplification of the input signal (as is the case with as Ace, ZX80 etc.), so the cassette signal needs to be turned up high.



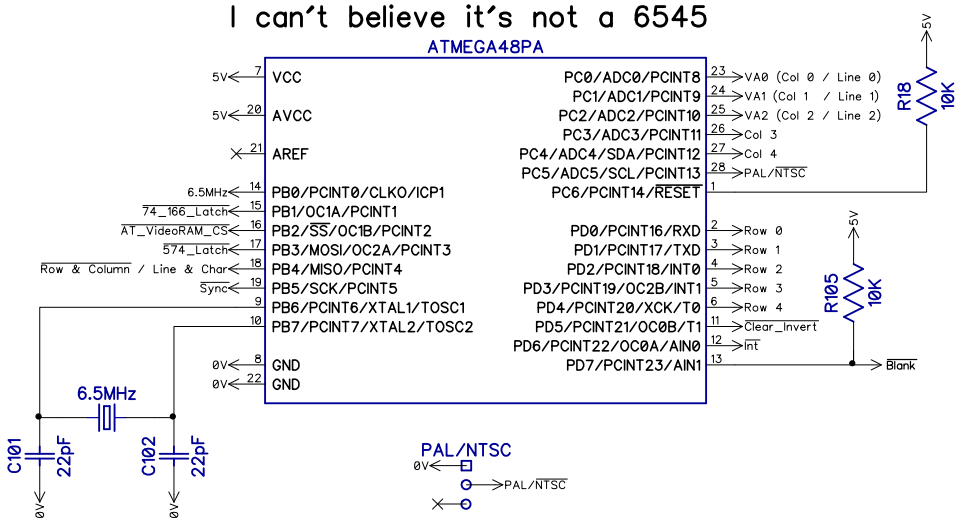
SOUND

Sound is generated by a 1 bit output which is toggled on and off with IO reads and writes. The piezo transducer is driven differentially between the two outputs to increase the volume.



VIDEO

Video timing is controlled by an ATmega48PA microcontroller performing a similar job to a 6845 CRT controller chip. This is essentially a large state machine which generates all the video sync and timing pulses. This microcontroller generates the 6.5MHz clock used to drive the CPU. A jumper input selects PAL or NTSC timing. The blank signal is triggered when not displaying character data and also if there is any address contention in the video RAM.

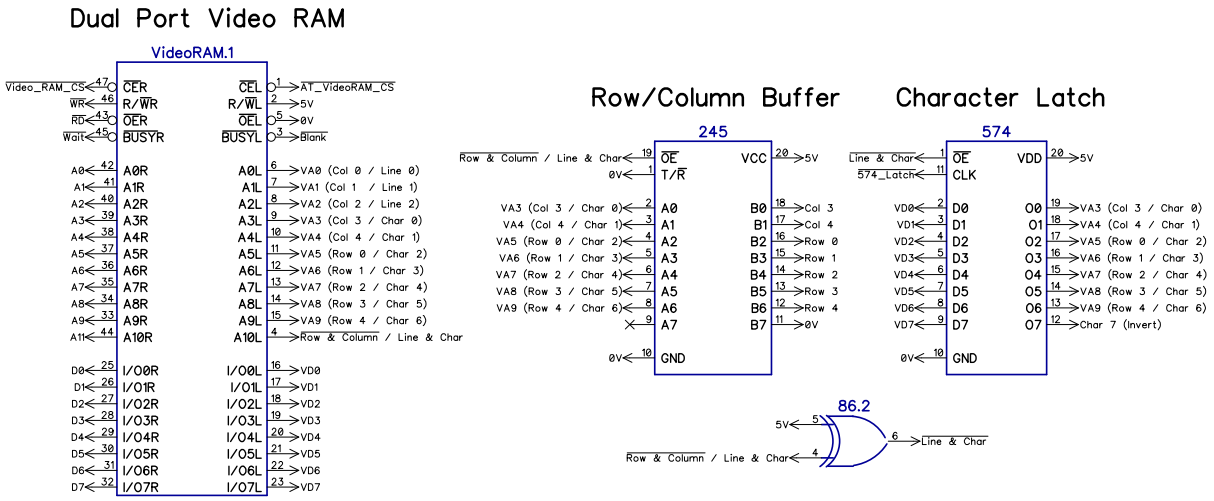


Is this cheating? Well, it's not doing anything more than you could do with a large amount of logic, it just takes up a lot less space. I did consider using a second Z80, a small ROM and some basic IO ports, that would have also taken up too much space, but I bet you wouldn't call that cheating, so I challenge your double standards and stick by my little microcontroller. So there.

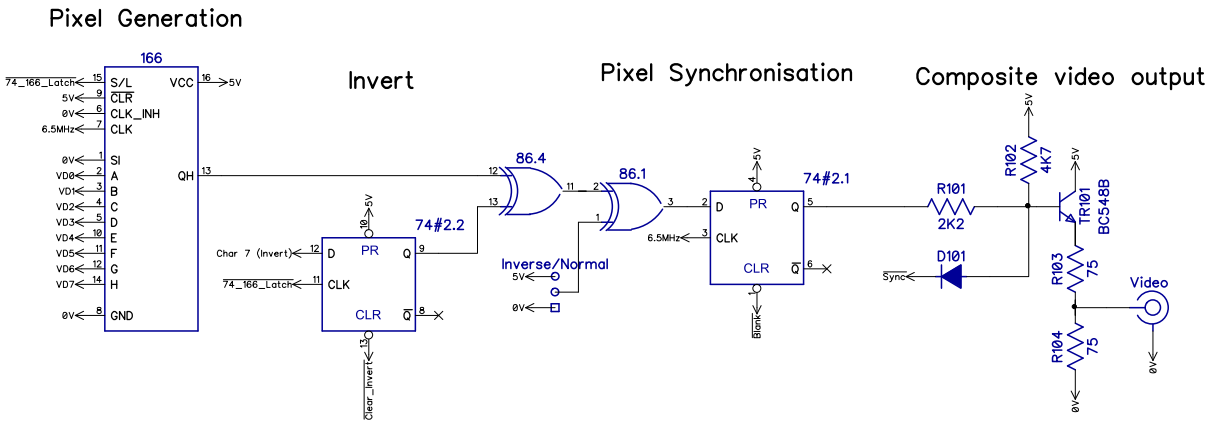
VIDEO RAM

The video RAM is a large 48 pin chip with completely separate address and data busses on each side. One side is connected to the Z80 bus, and the other to the video circuitry. The video side is addresses in two modes, one for the 768 bytes of video RAM, and the other for the 1K of font RAM. The mode also acts as the A10 line to select the block of RAM. The A10 pin on the Z80 side is actually attached to A11, as the 1K blocks are mirrored within those 2K windows. The two mirrors were originally used to select priority, but since contention is now done on the address level rather than the whole bank, the two banks now act identically but the mirroring is left in place for compatibility reasons.

The process of generating a character starts by addressing the video RAM using a row and column, two 5 bit addresses, part of which is enabled in this mode via the 245 buffer. This address selects a character on the video data bus. This character number is latched into a 574 character latch. The mode is then switched to one where it is addresses using the character number and a line number, which selects 8 bits of pixel data for one line of the character. All this happens within 8 clock cycles ready for the next character.

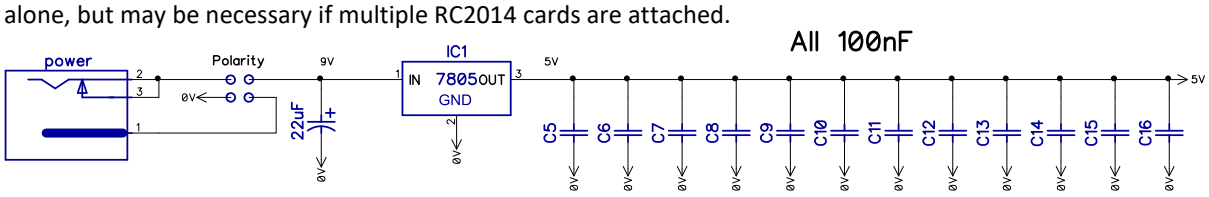


Those 8 bits of pixel data are latched into a 166 shift register, which clocks them out at 6.5Mhz. The 7th bit of the character code is the inverted / not inverted signal, which is latched in a 74 latch. The pixel data is inverted 0, 1 or 2 times at this stage, then synchronised with the 6.5MHz clock to give clean transitions. The composite video signal is then generated from the luminance and sync signals.



POWER

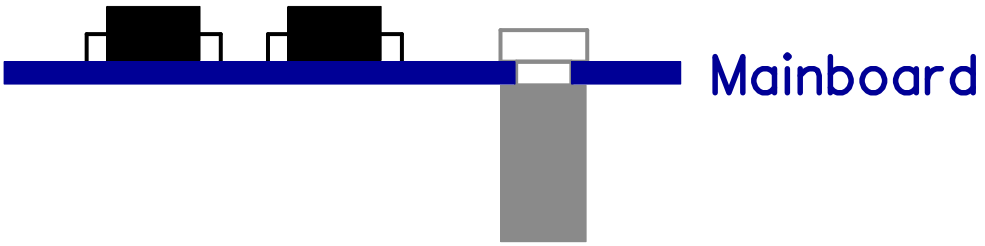
All the circuitry runs from 5V. Current consumption with HC chips and a modern EPROM and Z80 is around 65mA. This is provided by a 7805 regulator fed from a 9V input with selectable polarity. Power can also be provided to or from the RC2014 bus as required. The regulator does not need a heatsink if running the board alone, but may be necessary if multiple RC2014 cards are attached.



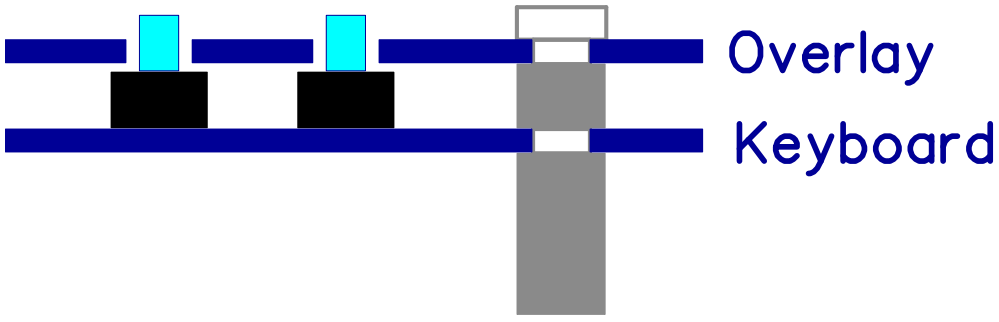
ASSEMBLY

Assembly should follow standard procedure for this type of board. Begin with the lowest profile components, the resistors, diodes and capacitors. Then fit the ICs and / or sockets and finally the higher profile items such as the connectors, jumpers and piezo sounder.

The open frame board can be mounted on the nylon M3 pillars, use the holes closest to the edges for greater stability (the extra mounting holes are to maintain ZX81 shape compatibility). These can be used as feet, or the complete unit can be screwed to a baseboard from below with M3 screws.



If building the keyboard with overlay, the spacer is placed between the keyboard and the overlay, and the screws go through these into the pillars.



RC2014 BUS

There is a full RC2014 bus connector on the side of the board. This can be fitted with a 40 way connector for standard RC40 bus cards, or with a 40 way and 10 way connectors, the enhanced bus. This can be used to attach a single RC2014 module, or an RC2014 backplane for multiple modules. This cannot be used in the ZX81 case configuration.

RC2014 is copyright RFC2795 Ltd, and the Minstrel 4th is '*designed for RC2014*'.