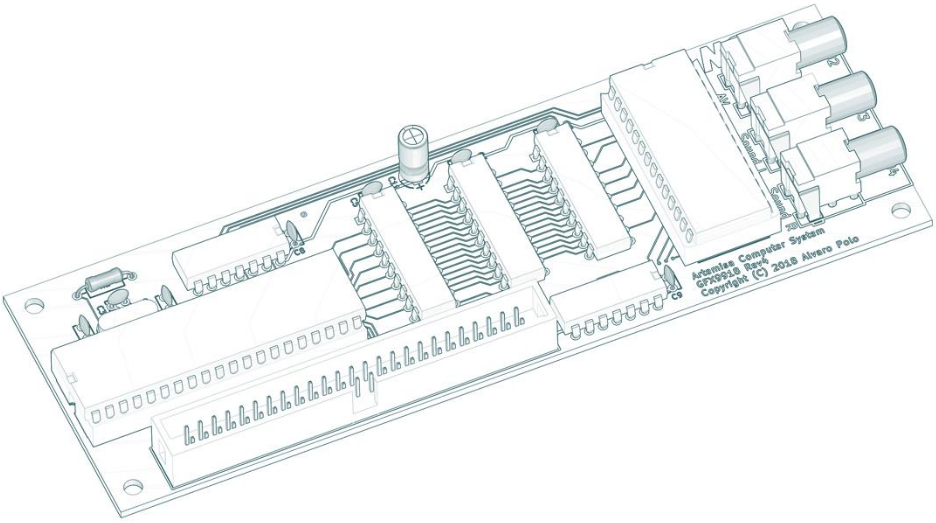
GFX9918
Assembly Manual



Rev04

Table of Contents

Introduction 5

About this document 5

Intended audience 5

How to Read This Manual 5

Special blocks 5

Theory of operation blocks 5

Warning blocks 6

Schematic diagrams 6

Number notations 7

General View of GFX9918 9

Architecture of Artemisa Graphic Cards 9

General Architecture 9

Architecture of GFX9918 11

Operation of GFX9918 11

Preparation 14

Contents of the Package 14

Required tools 14

Assembly 16

High Level Design 16

Card connector and power 16

Step 1 18

Step 2 19

Interface decoding logic 20

Step 3 21

Video Display Processor 21

Step 5 23

Step 6 24

VRAM decoding logic 24

Step 7 27

Step 8 27

Video RAM 28

Step 8 28

Output connectors 29

Step 9 30

Step 10 30

Step 11 31

Summary 31

Troubleshooting 32

Introduction

About this document

GFX9918 is the entry-level graphics card for Artemisa Computer System 100 series. This card provides the graphics capabilities based on the Texas Instruments TMS9918 chip.

This document provides a manual to assemble your GFX9918 card step by step.

Intended audience

If you have purchased an Artemisa 100 series computer, you are likely an electronic hobbyist or an MSX fan and you feel skilled enough to assemble a computer system by your own. This means soldering the components following the detailed instructions you will find in this manual.

The Artemisa 100 series is built using electronic components with large through-hole packages. Its integrated circuits are DIP (Dual Inline Package), with a pad separation of 2.54mm. Other components such as resistors and capacitors are also large ones. This resembles the design style of early 80s computers. But also lowers the entrance barrier for those that do not trust their own soldering skills. The grade of difficulty of assembling an Artemisa 100 series computer is low.

Thus, only a **basic knowledge in soldering** is required. If unsure, there are tons of guides and video tutorials available online to learn from. Assembling this kind of electronic components is really easy. Just forget your fears and enjoy the process!

One of the main incentives to purchase an Artemisa computer is to learn how the MSX computer works. If you only want to have an MSX machine, probably other options are better. On one hand, a refurbished MSX or a new FPGA-based one are likely less expensive. In the other hand, some other homebrew designs such as the Omega[1] are more powerful and MSX2-compliant. The offer of Artemisa is educative. The **goal is to make you learn** computer architecture and design at the same time you learn deeper about MSX computer internals.

In order to have a successful learning experience, some previous basic knowledge on computing is required. The user is assumed to understand basic concepts such as bit, byte, memory address, bus, CPU, peripheral, etc. You are not supposed to be an expert or have a degree in Computer Science. But at least to have the knowledge expected from users of microcomputers during the eighties.

How to Read This Manual

Special blocks

You will find some special text blocks all along this document. These blocks represent additional information or instructions that are not essential in the construction of your Artemisa computer. However, they represent information you may find useful.

Theory of operation blocks

These blocks will be represented as follows:

Theory of Operation

[1] https://github.com/skiselev/omega

Here goes a full detailed description of how a part of the circuit works.

The theory of operation blocks will describe the theory behind a part of the system. A basic knowledge on electronics and digital systems is required in order to fully understand these blocks. However, do not worry if you lack this knowledge or you find it difficult to understand. This information is not essential to assemble the system. You can still build your Artemisa computer without fully understanding what these text blocks tell. This is just for educational purposes.

Warning blocks

Some operations require special attention to things that, when unnoticed, could cause damage in the circuit. In these cases, a warning text block will indicate you have to pay special attention, like this one:

⚠ This is a warning message that requires your attention.

Schematic diagrams

Along this document, the different parts of the circuit are explained using schematic diagrams. They are not difficult to understand. But if they are totally unfamiliar to you, a brief introduction might be useful.

Most of the elements you will find in a schematic diagram, shown in Figure 1, are described as follows:

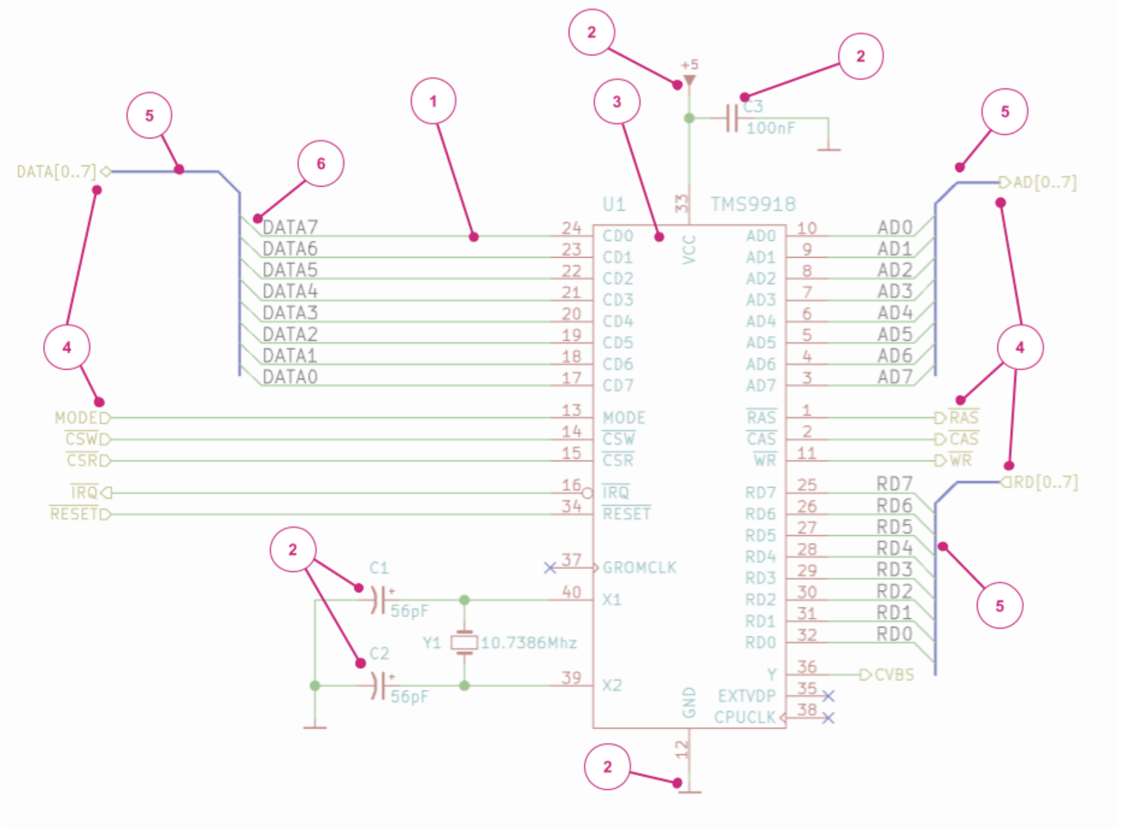


Figure 1: Elements of a schematic diagram

1. Wires. This is the most building block of a schematic diagram. It connects two points electrically. This means there will be a copper path between them that will let the current flow.
2. Basic components. Elements that have one or more pins where wires can be connected. These are power supply terminals, capacitors, resistors, oscillators, logic gates, etc. The different symbols used in this manual are described in Table 1.
3. Integrated circuits. Components that contains a whole and complex circuit inside it. These are processors, memories, flip-flops, decoders, etc.
4. Connections from/to other schematic diagrams. The pointing edge indicates the direction of the signal. If it has two pointed edges, it is bidirectional.
5. Buses. Collections of parallel wires that transmit a multibit data.
6. Wire connections from/to buses. A label indicates what is the wire that is obtained from the bus.

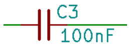
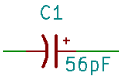
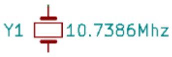
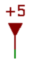
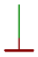
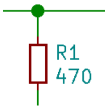
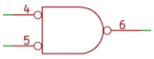
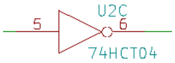
	Ceramic capacitor
	Electrolytic capacitor
	Clock oscillator
	5 volts power supply
	Ground power supply
	Resistor
	AND logic gate
	NOT logic gate

Table 1: Component symbols in schematic diagrams

Number notations

Some numerical values are expressed along this manual. The prefix formats shown in Table 2 are used to distinguish between different integer bases.

Prefix	**Description**

0b	Binary number format. Example: 0b01100011.
0x	Hexadecimal number format. Example: 0x4A0129FE.

Table 2: Number format prefixes

General View of GFX9918

Architecture of Artemisa Graphic Cards

As discussed in the introduction above, one of the key benefits of Artemisa is that it brings the opportunity to learn how an MSX computer is made.

But you will not learn that by just soldering some chips into a PCB. You have to understand what these chips do, and why they are there. And this requires a previous high-level description of how an MSX computer is made and how the components used to build them operate. This is what you will learn in this section.

General Architecture

As you might read in the Artemisa Motherboard Assembly Manual, Artemisa Computer 100 series moves the graphics processing unit into a separated board: the graphics card. This design enables the customization of the graphic devices while using the same motherboard as any other user. This derives in a reduction of production costs and leverages the upgrading of your computer.

The graphics card is connected to the Motherboard through a 50 pin IDC (Insulation Displacement Connector), as shown in Figure 2. You are probably familiarized with the IDC connector. It is the same used to connect motherboards to old IDE hard disks in the PC architecture. In this case, this connector is in fact a modified slot connector, similar to those used by MSX cartridges and present in some MSX computers as slot expansion port also using IDE like the Spectravideo SVI-728.

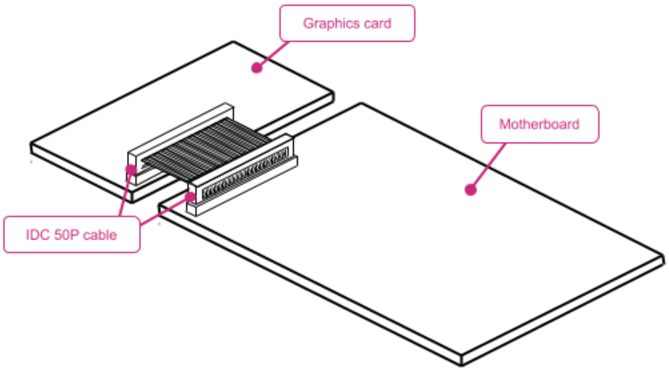


Figure 2: Graphics Card Connector

MSX slots are bus expansions

Roughly speaking, the graphics card connector, as the cartridge slot connectors, are direct expansions of the computer system bus. The standard 50 pins slot connector include:

- The address bus. These are 16 wires that carry the memory or IO address the CPU is accessing.
- The data bus. These are 8 wires that carry the byte read or write from or to the memory or the IO device.
- The control bus. These are multiple wires that carry control signals used by the CPU and the devices to signal specific actions (e.g., reset, read, write, interrupt, ...) and conditions (e.g., memory, IO, refresh, ...).

- An external audio line. For those devices that generate sound, this line can be used to transmit it to the system in order to be merged with the sound produced by the PSG.
- The power lines. +12V, -12V, +5V and GND. Artemisa is powered by +5V only to keep the design as simple as possible. +12V and -12V lines are connected to ground.
- A pair of signals not used and reserved.

The video card connector makes the following convenience modifications to the standard cartridge slot:

- The pin 16 is reserved, not used line in the standard connector. The Artemisa graphics card connector uses it for the /VDPIOSEL signal. This will tell the VDP when a IO request has been issued for it.
- The pin 49 transmits the external audio line from the peripheral to the computer. In Artemisa graphics card connector, the direction of this line is reversed. The audio line goes from the computer to the external device (the graphics card). The reason of this is to make it possible to combine video and sound in a single output signal for those graphic cards that require it (e.g., SCART connector).

Even though the connector carries a /SLTSL signal, it is never driven in this model of Artemisa computer. In other words, the card will respond to IO requests and not to memory accesses.

Any Artemisa graphics card will have the following basic elements:

- A Video Display Processor (VDP). This is an integrated circuit that implements the logic necessary to display graphics in your screen. MSX computers use VDPs of the TMS9918 family.
- Some Video RAM (VRAM) memory. The TMS9918 family chips use their own memory for video display, separated from the main memory of the computer.
- Input decoding logic. This will interpret the signals coming from the CPU and will adapt them to the VDP interface.
- Memory decoding logic. The VDP will typically access its memory by a set of signals that cannot be trivially connected to the VRAM. They have to be decoded and adapted to the interface of the memory chips.
- Video and audio signal processing. The VDP will typically produce an output video signal that is not ready yet to drive the monitor. Further transformation is needed. In some implementations, that will include the mix of the audio signal coming from the motherboard in the output.

VRAM is fully owned by the VDP

Having a separated VRAM instead of using the main memory of the computer is a really interesting tradeoff present in the TMS9918 family.

Those VDPs that use the main RAM memory to store video data have to share it with the CPU. In some other Z80 architectures, if both devices need to access the memory at the same time, one of them will have to yield. If the VDP stops reading memory, it will not be able to render the current frame in the screen. Hence it is the CPU the one that stops its job and have to wait for the VDP to finish.

Fortunately, there are some tricks to mitigate this problem. For example, the VDP does not need to access the memory all the time. There is a time window between the last line of a frame and the first line of the next frame are drawn in which there is no memory access. The CPU can use this time window to access the video memory to prepare the next frame. This technique is widely used in the Sinclair ZX Spectrum computer.

Computers based in TMS9918 do not have this problem. The VRAM memory is completely owned by the VDP. And the CPU can read or write to them by requesting IO operations to the VDP.

Architecture of GFX9918

In the case of GFX9918 video card, these basic elements, shown in Figure 3, are:

- A TMS9918A VDP chip.
- 16KB of static memory used as VRAM.
- A custom input decoding logic. A few logic gates to interpret the system bus signals according to the interface of the TMS9918.
- A custom memory decoding logic that will demultiplex the signals coming from the VDP and will adapt to the timings of the VDP.
- No specific video output processing. The TMS9918 generates a composite video signal suitable for any TV set. Both this video output, and the sound signal coming from the motherboard, are available through a triple RCA connector.

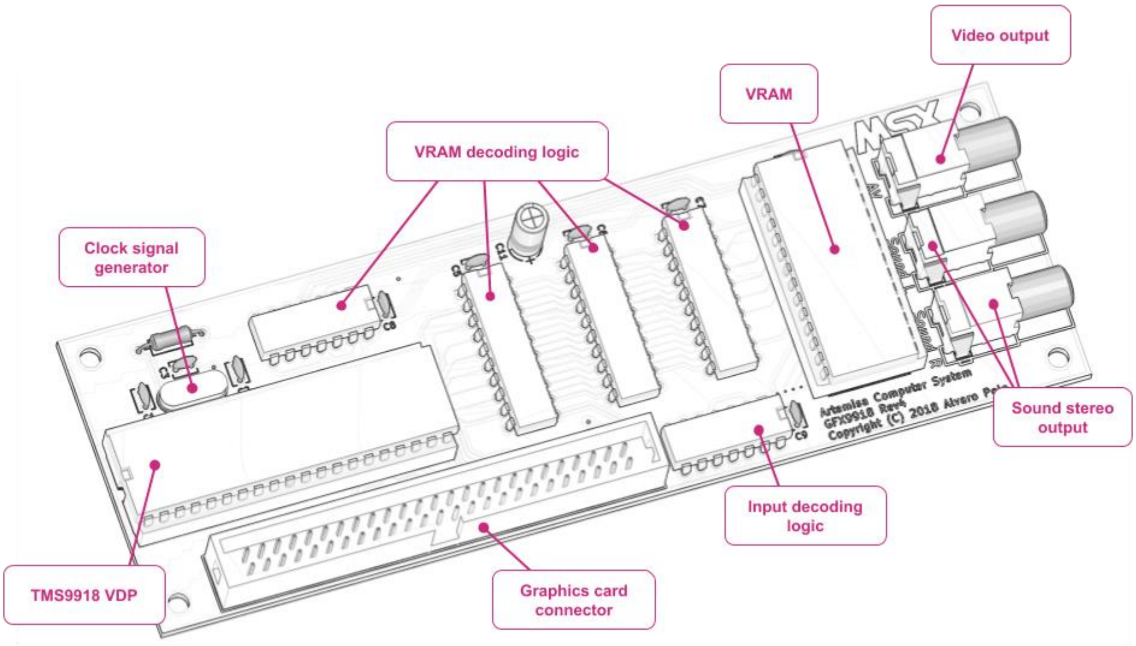


Figure 3: GFX9918 Basic Elements

Operation of GFX9918

The TMS9918 video display processor contains so internal register that can be programmed by the user:

- There are eight internal mode registers that can be used to configure things such as the screen mode, the VRAM base addresses for the different memory areas, the color palette, etc.
- There is one internal address register that will determine the VRAM address where read and write operations will be located.

As mentioned above, the TMS9918 chip requires a dedicated video memory separated from the main memory of the computer. This means the CPU will communicate to the VDP using exclusively the IO read and write operations.

The MSX1 standard establishes that the VDP is attached to the IO ports from 0x98 and 0x99, as shown in Table 3:

Port	Description
0x98	The data port. This is where the CPU will read or write when it wants to transfer data from or to the VDP.
0x99	The command port. On read, this will return the contents of the status register. On write, this will be interpreted as a command sent by the CPU to the VDP.

Table 3: VDP IO Ports

This is a summary of the typical operations the CPU requests to the VDP:

- Obtain the status of the VDP. This is done by reading the command port at IO address 0x99. This is useful to detect sprite collisions and frame rendering completed.
- Write into a mode register. This is done by writing two subsequent bytes to the command port at IO address 0x99. The first byte is the content to be written to the mode register. The second byte must match the binary pattern 0b10000xxx, where xxx is the 3-bit address of one of the eight available mode registers. The VDP knows this is a write request to a mode register because of the 0b10000 prefix.
- Write to VRAM address. This is done by first writing two subsequent bytes to the command port at IO address 0x99. These two bytes has the purpose of setting the contents of the internal address register mentioned above. The first byte contains the less significant bits of the address. The second byte must match the pattern 0b01xxxxxx, where xxxxxx are the six more significant bits of the address. The VDP knows this is a write request to VRAM because of the 0b01 prefix. After that, it expects the data to be written into the data port at IO address 0x98. The internal address register is autoincremented. After each write to the data port, it will increment its value pointing to the next byte in VRAM. This can be used by the CPU in order to transfer a large block of bytes by just setting the VRAM address once, and then writing the bytes in a row.
- Read to VRAM address. It works exactly as the write to VRAM but using the prefix 0b00 instead of 0b01. And reading from the data port instead of writing.

Preparation

Contents of the Package

The Artemisa GFX9918 package should contain the following elements:

Artemisa GFX9918 printed circuit board

Integrated circuits ESD bag, including:

1x TMS9918: Video Display Processor (VDP
1x 62256: 32,768-word × 8-bit High Speed CMOS Static RAM
1x 74HCT04: hex inverter
3x 74HCT574: 8-bit D-type flip-flop with 3-state outputs
1x 74HC32: quad 2-input OR gate

Sockets and connectors bag, including:

1x IC Socket 28p .600 mil
1x IC Socket 40p .600 mil.
3x RCA connectors (yellow, white and red)
1x IDC 50p PCB connector

Passive components bag, including:

2x Ceramic capacitor 56pF
7x Ceramic capacitor 100nF
1x Electrolytic capacitor 10uF
1x Resistor 470 ohms
1x Crystal 10.7386Mhz HC-49S CL=32pF

Triple RCA cable for composite video + stereo sound

If some part is missing or damaged, please contact support@artemisamsx.com.

Required tools

You will need the following tools to assembly your GFX9918 unit:

Electric soldering iron

One suitable for soldering electronic components.

Spool of solder cable

If you have a limited experience on soldering, lead-based alloy is recommended. It melts at lower temperature, facilitating the job.

Wire cutters

You will need to remove the remains of legs for resistors and capacitors.

The following tools may be useful during the assembly process but are not required. If you have access to them, it would be a good idea to prepare them. If not, do not worry. You can assemble your unit without them.

Multimeter

It might be useful to check circuit voltage and continuity.

Assembly

High Level Design

The schematic diagram in Figure 4 shows the main elements of the GFX9918:

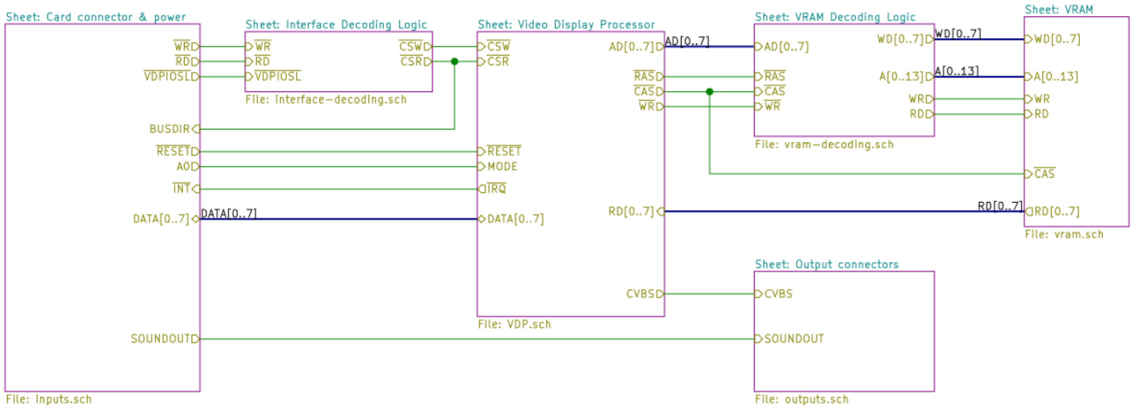


Figure 4: Main Schematic Diagram

- The Card Connector and Power. This is where card connector is placed, providing the control signals and the data bus to make the motherboard interface with the VDP.
- The Interface Decoding Logic. It receives a few control signals coming from the input connector and will generate the control signals adapted to the VDP interface.
- The Video Display Processor. This is the TMS9918 chip and the auxiliary elements that make it work. It receives the inputs from the Interface Decoding Logic and the Input Connector. And generates control signals and buses to interface with the VRAM memory through the VRAM Decoding Logic.
- The VRAM Decoding Logic. As its name suggests, it decodes the buses and control signals from the VDP into something the VRAM memory can understand.
- The Video RAM. The memory chip that will receive the inputs from the VRAM Decoding Logic and will respond to the corresponding memory accesses coming from the VDP.
- The Output Connectors, where the video output signal generated from the VDP and the audio signal coming from the Input Connector are mixed in a triple RCA connector.

We will discuss each block separately along this section of the manual. This brings an opportunity to understand how each part of the circuit works at the same time we solder the components to the board.

Card connector and power

We will start the assembly with the card connector and the power management elements.

As you can see in the schematic shown in Figure 5, this is quite simple. The 50P IDC connector is identified by J1. As you can see, we will use only the following control signals from the system bus:

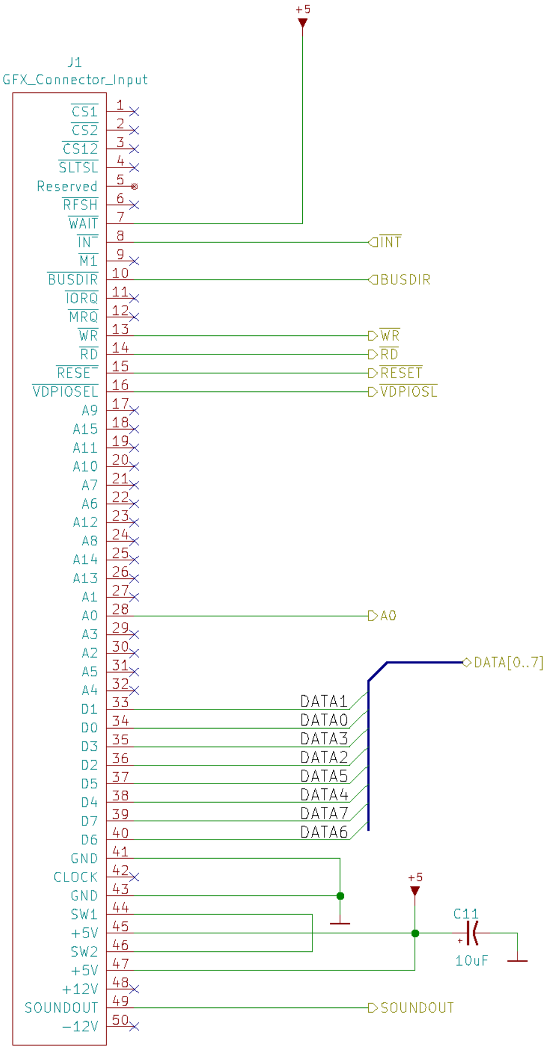


Figure 5: Card connector and power schematic diagram

- The /INT output signal. This will be activated by the VDP whenever it completes a whole screen frame.

CPU interruptions

Interruptions are a common way to communicate events from the peripherals to the CPU in any computer system. In this particular case, the VDP activates the interrupt signal when a whole screen frame is completely rendered. This is 60 times per second in GFX9918 due to the NTSC video encoding standard.

This interruption will cause the CPU to stop executing the current program and invoke an interruption routine, typically provided by the system BIOS. In the MSX system, this interruption is used as a system-wide timer that is triggered at known intervals (every 16.6 milliseconds). The default BIOS routine uses it to perform regular tasks, such as read the keyboard state. The user software can use it as well. For example to move the scene objects in a game.

- The BUSDIR signal. It will be activated by the graphics card whenever a data is sent by the VDP to the CPU.
- The /WR and /RD signals, indicating a write or read operation, respectively.
- The /RESET signal, which is activated when the system is powered on or restarted. This will be used by the VDP to reset to its initial state.
- The custom /VDPIOSL signal, which indicates an IO operation in any of the IO ports designated for the VDP.
- The A0 signal. This is the least significant bit of the address bus. Hence it distinguishes between IO ports 0x98 and 0x99, which are the same binary numbers except their last digit.
- The data bus, which is used to transfer data between the CPU and the VDP.
- The +5V and GND power connections.

As you can see in the schematic, there is also an electrolytic capacitor connected between 5V and GND. This is used to decouple the power circuit from AC fluctuations at low frequencies.

Decoupling capacitors

Electronic components do not consume the same current all the time. They experiment fluctuations. When a current spike occurs, the voltage provided by the power supply also changes. This could cause a voltage drop that can affect not only the component but also some other elements of the same board.

The way to prevent this is to use a decoupling capacitor. Think about it as a local battery connected to some part of the circuit. If the current drawn from the wire changes, the voltage drop can be compensated by the load stored in the capacitor. The volts lost by the power supply are provided by the capacitor. Maintaining the same voltage level until the power supply adapts its level to the new demand, or the extra current demand ends.

Typically, every integrated circuit has its own small decoupling capacitor to protect itself from the voltage fluctuations. There is also a single bulk capacitor aimed to decouple the whole power circuit of the board from the power line coming from the card connector.

Step 1

We will start by soldering the 50P IDC connector J1 in the board, as shown in Figure 6.

Please take care of the orientation of the connector. The notch must look to the bottom edge of the card, where the ribbon cable comes.

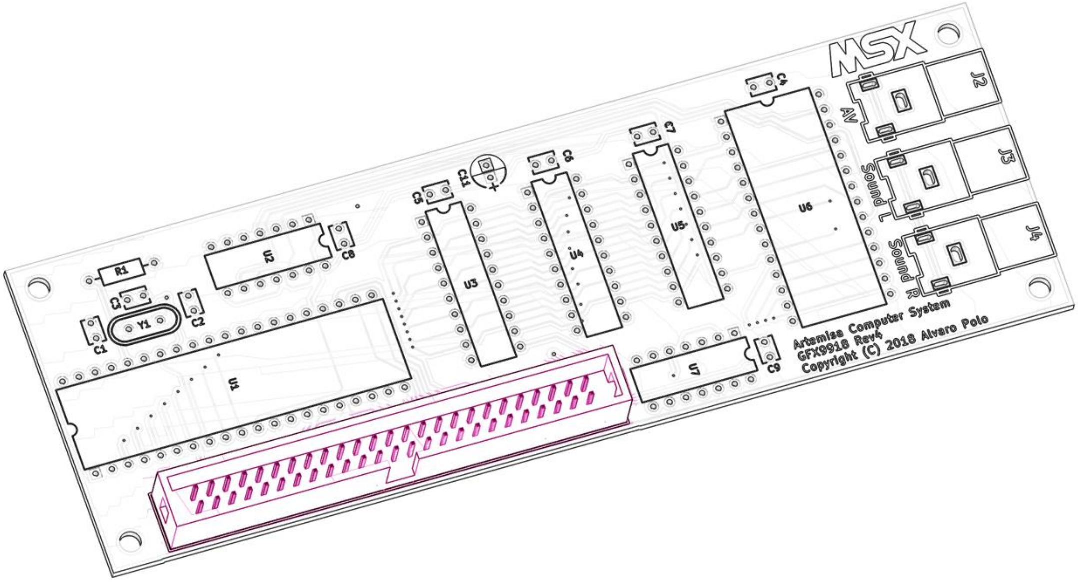


Figure 6: Placement of card connector

Step 2

Now we can solder the electrolytic capacitor C11, as shown in Figure 7.

⚠ Take care of the polarity of the electrolytic capacitors. If soldered incorrectly, the part might explode. The longest pad must be soldered in the hole marked with a plus sign.

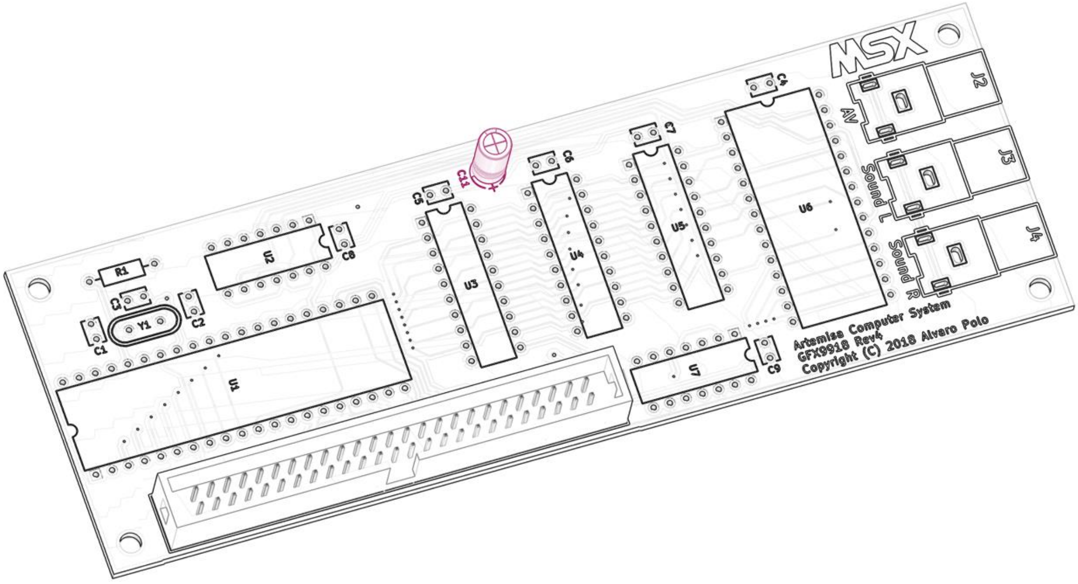


Figure 7: Placement of capacitor C11

Interface decoding logic

The interface decoding logic adapts the signals coming from the input connector into something adapted to the interface of the TMS9918.

This sounds complex, but it is as simple as a pair of AND logic gates as shown in Figure 7.

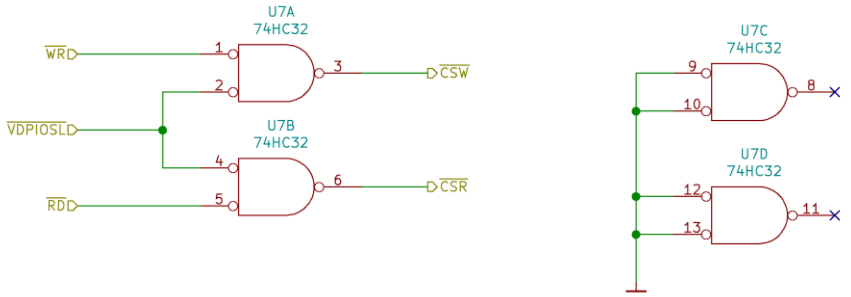


Figure 8: Interface decoding schematic diagram

Unused gate inputs

As you can see in Figure 8, there are two AND gates on the right side that are unused. However, their inputs are connected to GND instead of left disconnected. This is to prevent floating values that could damage the circuit due to electrostatic currents. Also, a floating value might cause fluctuations in the logic inputs, that in turn increases the power consumption in case of CMOS components. Any unused logic gate should be connected to either GND of VCC for a fixed input value.

The following inputs and outputs are handled in this decoding phase:

- The /WR input signal is activated when a write operation is requested by the CPU.
- The /RD input signal is activated when a read operation is requested by the CPU.
- The /VDPIOSL input signal is activated when an IO request to the VDP ports is requested by the CPU.
- The /CSW output signal indicates the VDP is selected for a write operation. This is the logic expression: /CSW = /WR *and* /VDPIOSL.
- The /CSR output signal indicates the VDP is selected for a read operation. This is the logic expression: /CSR = /RD *and* /VDPIOSL.

Please note all these control signals are active-low.

Active low signals

There are two different conventions used to consider a digital signal is active or inactive. One is to assume a high level means active and low level means inactive. This is called active-high. It is the most natural convention due to the assumption that low (0) means not while high (1) means yes. However, the reverse logic, called active-low, also exists. It considers a low level means active and a high level means inactive. All signals whose name is prepended with a slash (e.g., /CSR) are active-low.

In despite of active-high to be more natural, active-low is widely used in digital electronics. There are two main reasons for that. One of them is that old TTL family gates consume significantly more power when they receive a low level input. Considering the signals are most of the time inactive, active-low convention saves a lot of power.

The other reason is that combining active-low logic with open-collector outputs and pull-up resistor gives an OR function without using any logic gate. This combination of elements is not present in this circuit.

Step 3

Solder the integrated circuit U7 (74HC32) and its decoupling capacitor C9 (100nF) as indicated in Figure 9.

Take care with the orientation of the integrated circuit. The IC has a a notch at the top that must match the drawings in the board.

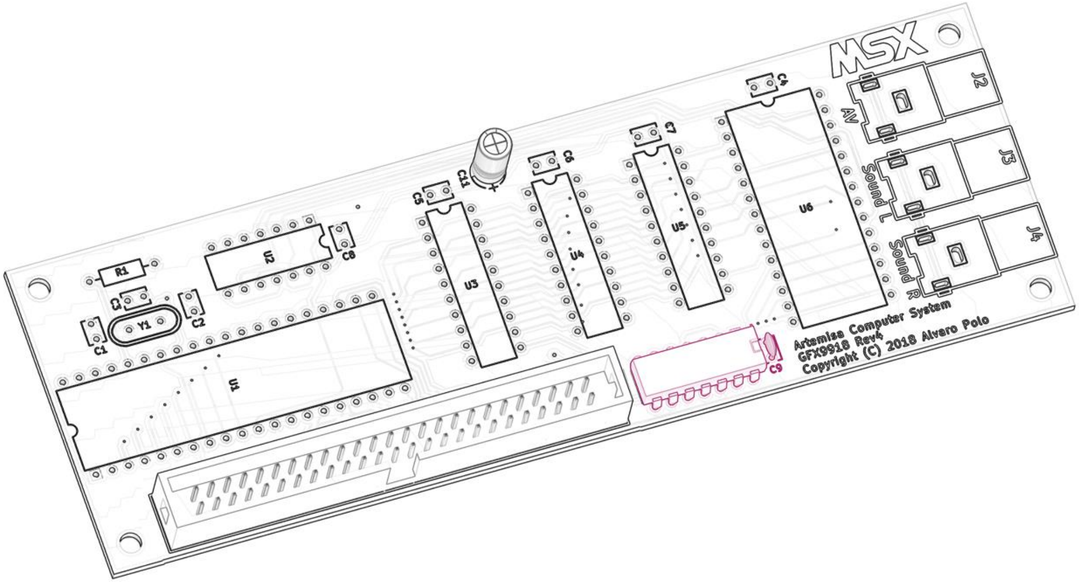


Figure 9: Placement of integrated circuit U7

Video Display Processor

The Video Display Processor is mainly the TMS9918 chip and the associated clock generator circuit, as shown in Figure 10.

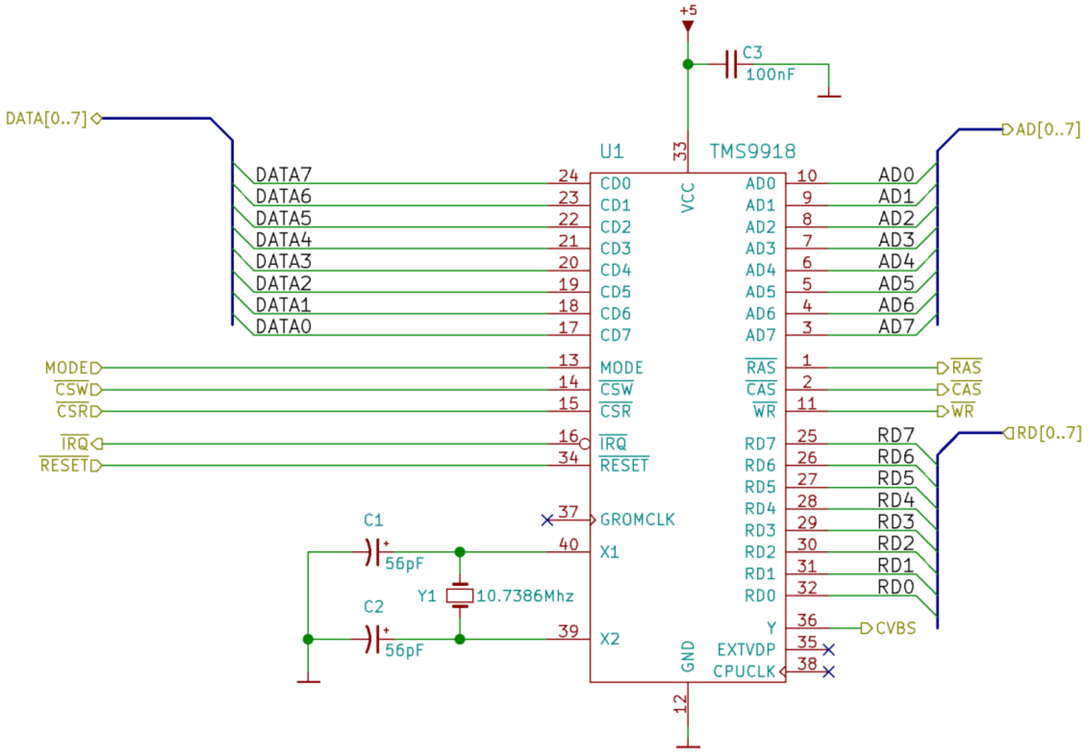


Figure 10: Video Display Processor schematic diagram

The VDP receives the following upstream signals:

- The data bus coming from the CPU. It comes from the graphics card connector.
- The /CSW and /CSR signals, which are generated by the Interface Decoding Logic. Active when write and read operations, respectively, are requested by the CPU.
- The mode signal, which determines whether an IO request for the chip refers to the data register (low) or the command port (high). This is connected to the lowest bit of the address bus, or A0. Thanks to this, the data register will be selected on IO port 0x98 (0b10011000), and command register on port 0x99 (0b10011001).
- The /IRQ signal generated by the VDP. This will be active when the VDP wants to interrupt the CPU. This happens every time a new screen frame is drawn. This signal is directly connected to the graphics card connector.
- The /RESET signal. This comes from the graphics card connector and will be activated when the system boots up or is reset. This is used by the VDP to reset into an initial state.

The VDP also generates the following downstream signals:

- The AD bus, which multiplexes the VRAM higher 7 bits of the address (the Column Address), its lower 7 bits (the Row Address) and the data output (in case of writes) when the VDP accesses its memory.
- The RD bus, which is the input data bus coming from the VRAM. It is only used for reading.
- The /RAS (Row Address Strobe) and /CAS (Column Address Strobe) signals. They indicate a row and a column memory access, respectively. They are part of the memory access sequence performed by the VDP.
- The /WR signal. This is active when the current memory access is for writing, and inactive if it is for reading.

- The CVBS signal. This is the composite video output signal generated by the VDP. This is connected to the Output Connectors.

Memory buses multiplexing

In many computer systems, it is very common to split memory address by row and column. Therefore, you might see the /RAS and /CAS lines present in many other designs.

There are two reasons for this. First, because of the internal construction of the memory chips. They are disposed as a matrix of bit cells. Hence, it is straightforward to assume each bit is accessed by its column and row addresses.

In the other hard, having row and column address is good to reduce the size of IC encapsulation and hence save space in the electronic boards. This is possible because both row and column addresses can be multiplexed in a single bus. Multiplexing means that different information is transmitted on the same channel at different instants. Thanks to this, we can save half of the width of address buses. For example, eight pins are enough to address 64K elements by multiplexing 8-bit row and 8-bit column addresses.

In the case of TMS9918, the 8-bit bus AD multiplexes the row and column addresses, along the output data for write operations.

Along the upstream and downstream signals, there is a clock generation circuit that can be seen in the schematic diagram. This is comprised by a 10.73 Mhz crystal and a pair of ceramic capacitors. They conform a resonator circuit used internally by the VDP to generate the internal clock signal used for its operation.

⚠ Please note the lines from the data bus are inversely connected to the VDP. This is because the Texas Instruments notation assigns the first line number (0) to the most significant bit, and the last line number (7) to the less significant bit. The rest of the world does it the other way around. First number (0) is the less significant bit and last number (7) the most significant bit. Thus, the data bus seems to be reversed but it is not.

Step 5

Solder the 40P DIP socket, as shown in Figure 11.

⚠ Just solder the socket, but do not plug the VDP chip yet in it. We will not need it at this point. And having it connected while soldering increases the risk of causing damage to the chips by overheat.

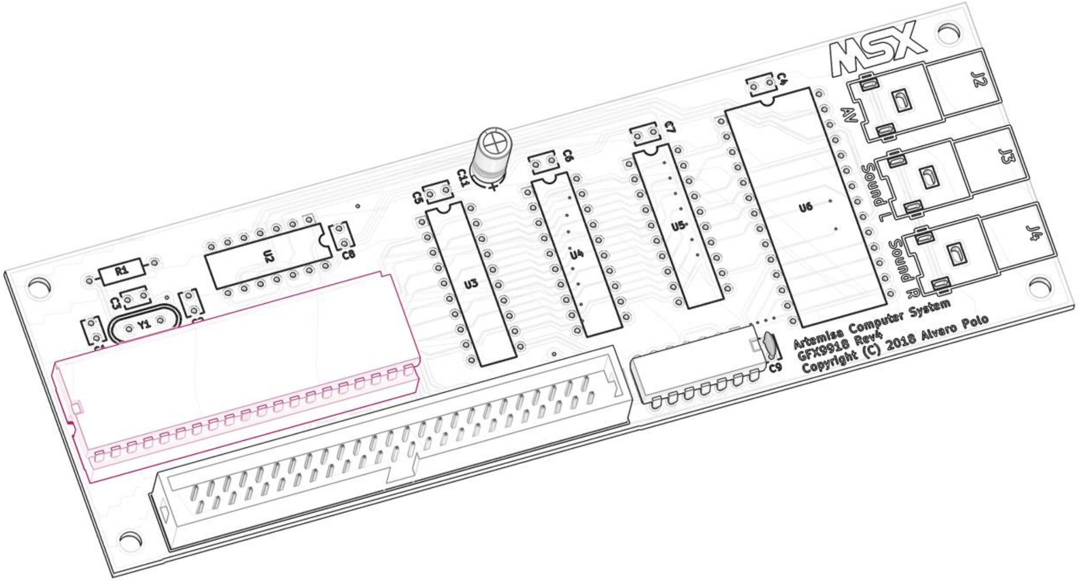


Figure 11: Placement of Video Display Processor

Step 6

Solder the components of the clock signal generation, as shown in Figure 12:

- Crystal oscillator Y1. Value 10.7386 Mhz.
- Capacitors C1 and C2. Value 56 pF.

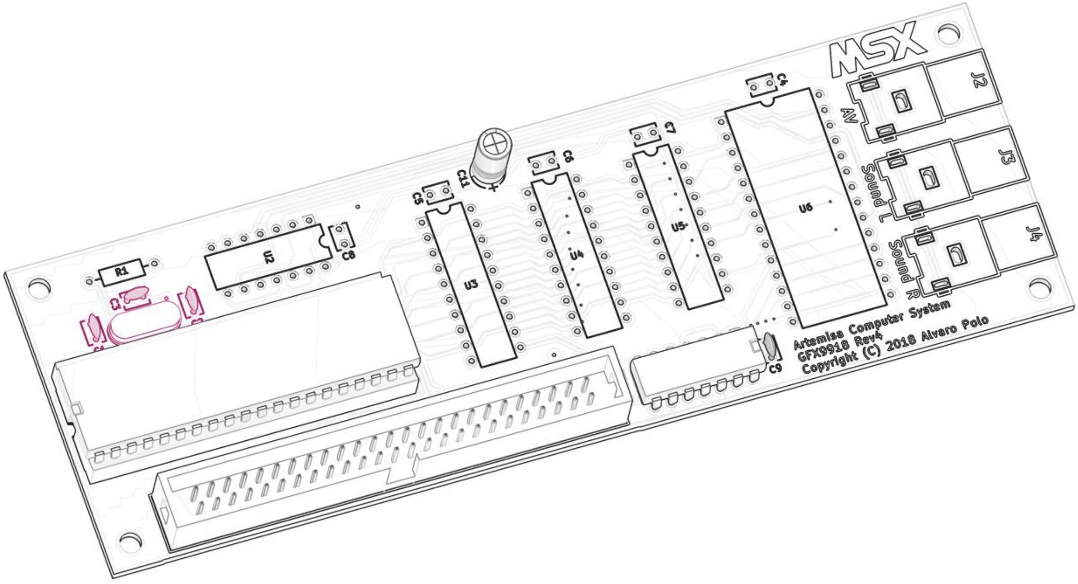


Figure 12: Placement of clock generation circuit

VRAM decoding logic

This is probably the most complex part of the design. This is based upon the work of Tom LeMense[2].

[2] https://cdn.hackaday.io/files/5789247676576/9918-SRAM.pdf

The main goal of this circuit is to strobe the address and data from the AD bus of the VDP at the right instant. As mentioned above, the TMS9918 multiplexes the following information in this buffer while accessing the VRAM:

- The higher 7-bits of the VRAM memory address, which is known as Column Address.
- The lower 7-bits of the VRAM memory address, which is known as Row Address.
- The output data in case the VDP is writing to the VRAM.

The VDP drives the AD bus with each of these items at different instants through a memory access. First, it puts the Row Address, followed by the Column Address, and finally the data to be written in case of a write.

The circuit fetches each item that travels through the AD bus and saves it in three different registers: one for the row address, one for the column address, and one for the data to write. These registers are D-type flip-flops with positive edge-trigger and 3-state output buffers with part name 74HCT574. They are U3, U4 and U5 as shown in the Figure 13.

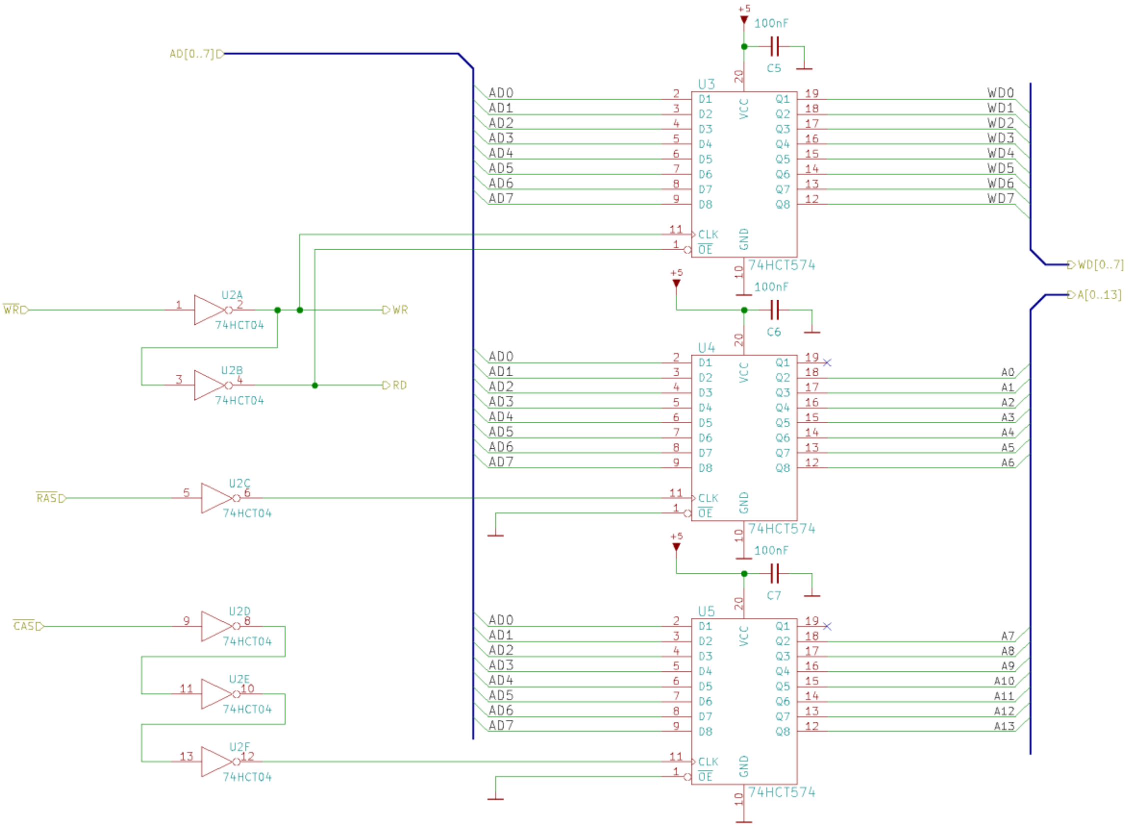


Figure 13: VRAM decoding logic schematic diagram

The flip-flips will demultiplex the signals from AD bus to produce the full 14-bit address in A bus, and the data to be written in the WD bus.

VRAM bus multiplexing

The AD bus is multiplexed in every access of the VDP to the VRAM. For read operations, the /RAS and the /CAS signals determine the contents present in the bus, as shown in Figure 12.

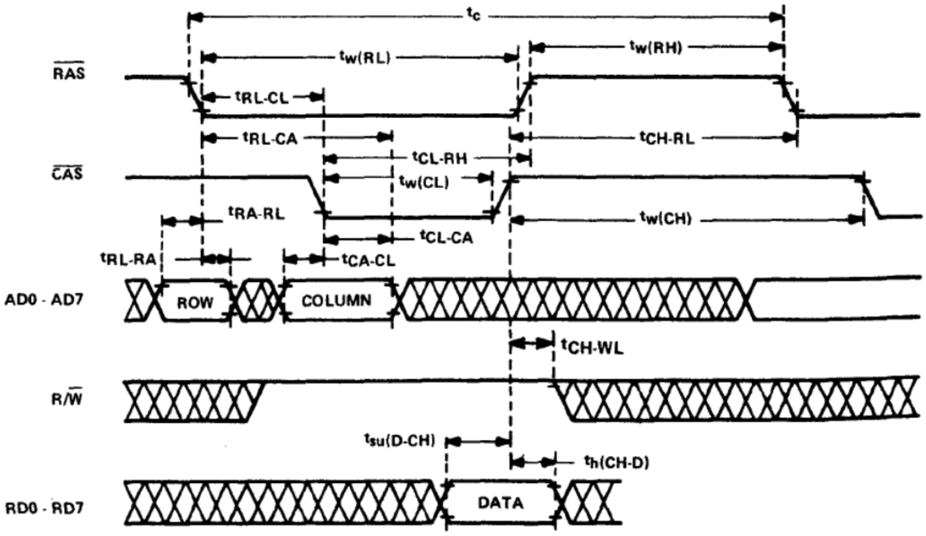


Figure 14: Timing diagram of VRAM read access

- When /RAS activates on its falling edge, the AD bus contains the row address. This signal is inverted and then connected to U4 clock input. This means U4 will register the row address when /RAS is asserted.
- When /CAS activates on its falling edge, the AD bus contains the column address. This signal is inverted and then connected to the U5 clock input. This means U5 will register the column address when /CAS is asserted. The reason to have multiple inverters is to generate large propagation delays from /CAS to the signal input of U5. This is required to satisfy the timing constraints of the VDP.

In case of a write operation, the mechanism is similar. But after /RAS and /CAS signals, the VDP will drive the AD bus with the data to be written, and will assert it with the RW signal, as shown in Figure 15:

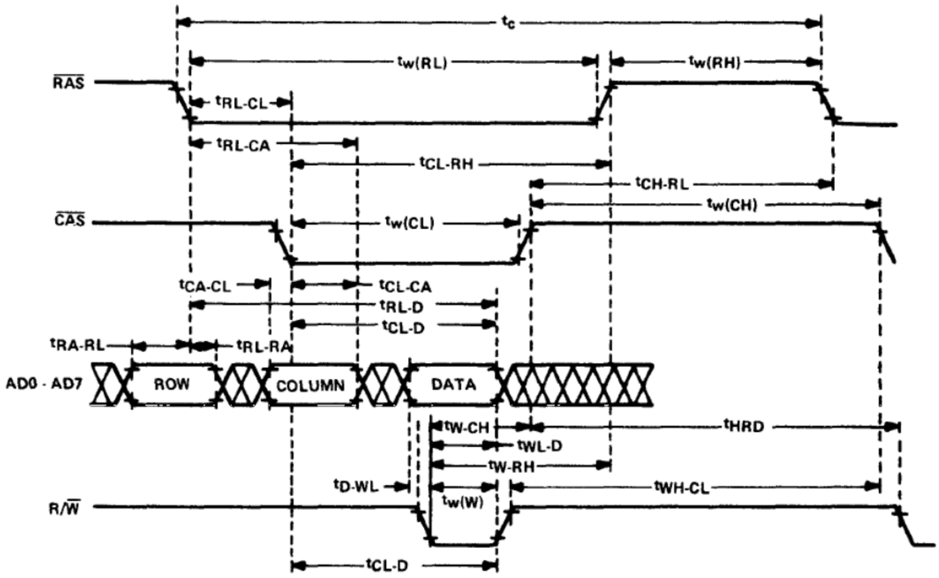


Figure 15: Timing diagram of VRAM write access

The RW signal is inverted and then connected to the clock input of U3. This means U3 will register the data to be written on RW falling edge.

As result, the A bus in the right hand of the schematic diagram contains the full 14-bit address selected by the VDP. The WD bus contains the data to be written. Please note the OE input of U3 will ensure the bus is driven only during the write cycles.

Step 7

Solder the integrated circuits U3, U4 and U5 with their decoupling capacitors C5, C6 and C7 of 100nF, as shown in Figure 16.

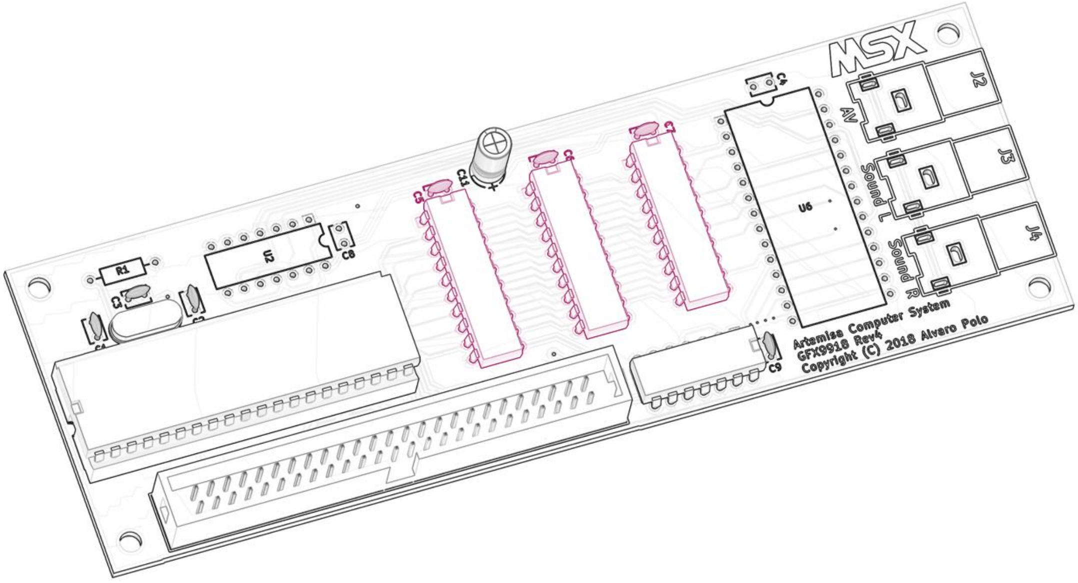


Figure 16: Placement of U3, U4 and U5

Step 8

Solder the integrated circuit U2 and its decoupling capacitor C8 of 100nF, as shown in Figure 17.

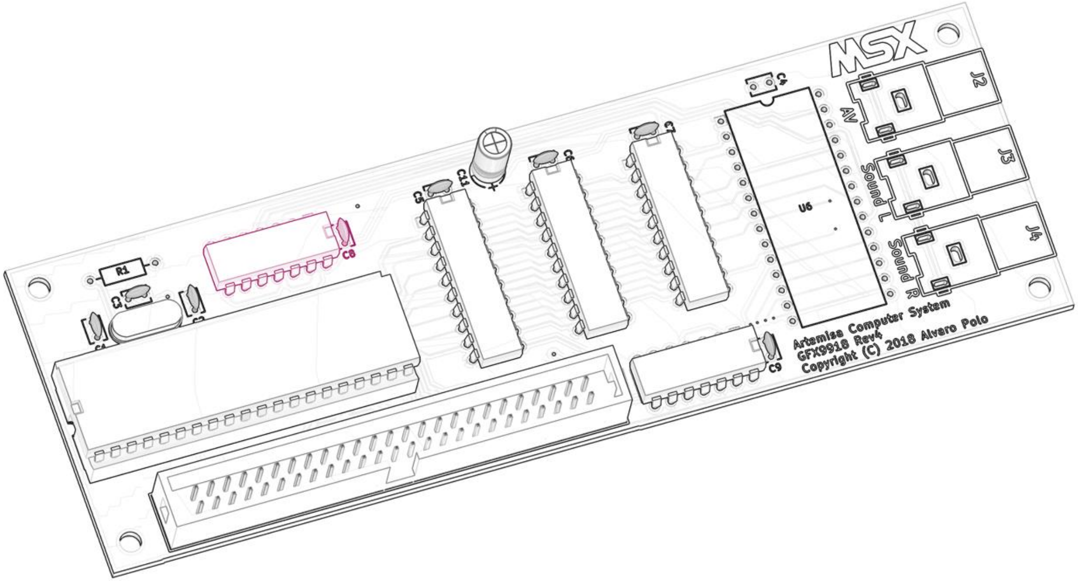


Figure 17: Placement of U2 and C8

Video RAM

The Video RAM circuit is pretty straightforward once the signals generated by the VDP are appropriately decoded. As we have seen, the VRAM decoding logic generates two buses that are directly connected to the VRAM:

- The A bus, which contains a completely demultiplexed 14-bit address.
- The WD bus, which contains the data sent by the VDP for write operations.

As shown in Figure 18, these buses are connected to U6, the static RAM chip with part name 62256.

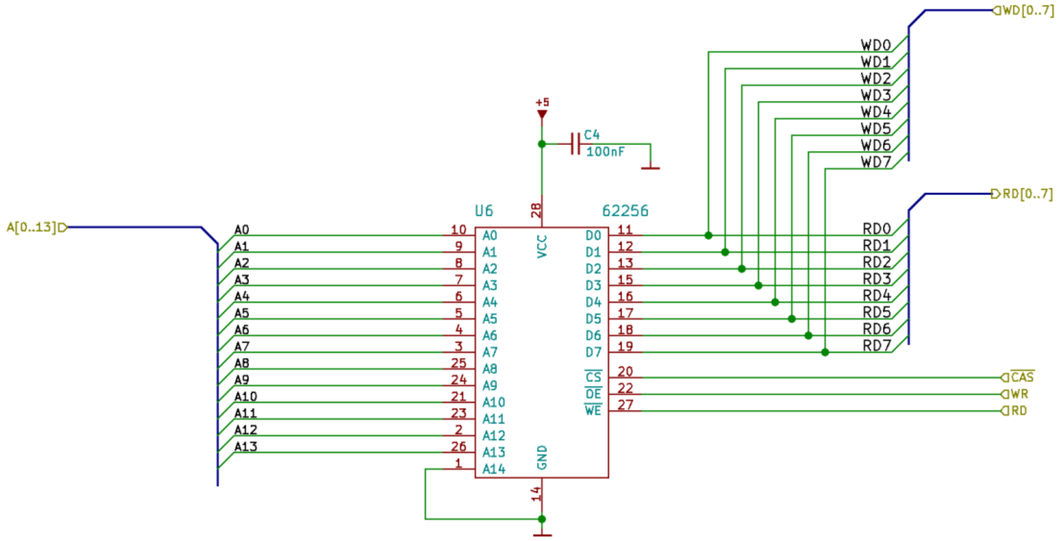


Figure 18: VRAM schematic diagram

In addition, there is another bus RD, which carries the data to the VDP in case of read operations. Both buses, WD and RD are connected to the data pins of the memory chip.

Finally, the memory chip receives the following inputs:

- /CS, the chip select signal. It indicates the memory chip is selected for a read or write operation. This is connected to the /CAS signal produced in the VRAM decoding logic.
- /OE, the output enabled signal. It indicates the memory chip can drive the data bus. This is connected to the WR signal generated by the VRAM decoding logic, which ensures the data bus is driven only during write operations.
- /WE, the write enabled signal. It indicates a write operation when active-low or read operation otherwise.

Step 8

Solder the 28 pin DIP socket in the placement of U6 and the decoupling capacitor C4 of 100nF, as shown in Figure 19:

As we did for the VDP, just solder the socket with the memory chip unplugged in order to prevent any damage caused by overheat.

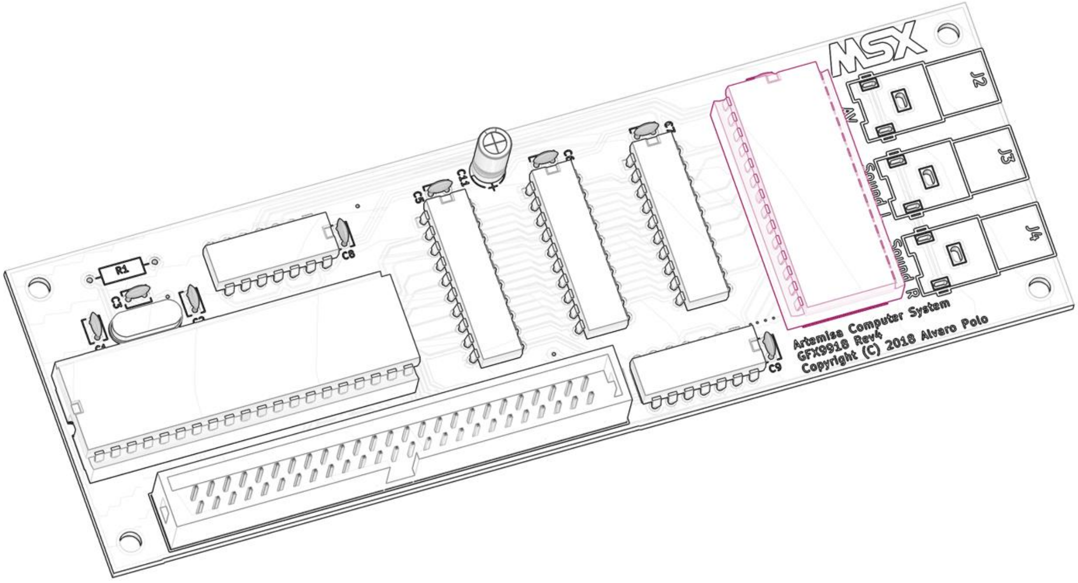


Figure 19: Placement of U6 and C4

Output connectors

Our final stage is to connect the outputs of the circuit to have suitable video and sound through RCA connectors as shown in Figure 20.

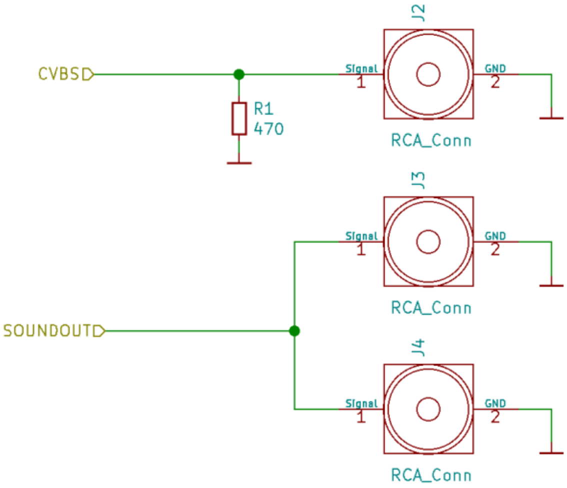


Figure 20: Output connectors schematic diagram

The CVBS signal requires a pull-down resistor of 470Ω to reduce the voltage of the composite video signal. After that, it is directly connected to the yellow RCA terminal.

The sound output from the card connector is directly connected to the white and red RCA terminals for a mono sound setup with two speakers.

These RCA terminals can be connected to a TV set with a NTSC composite video input.

Step 9

Solder the RCA connectors to the board, as shown in Figure 21.

> ⚠ Mind the colors of the RCA connectors. From the board top: yellow, white and red.

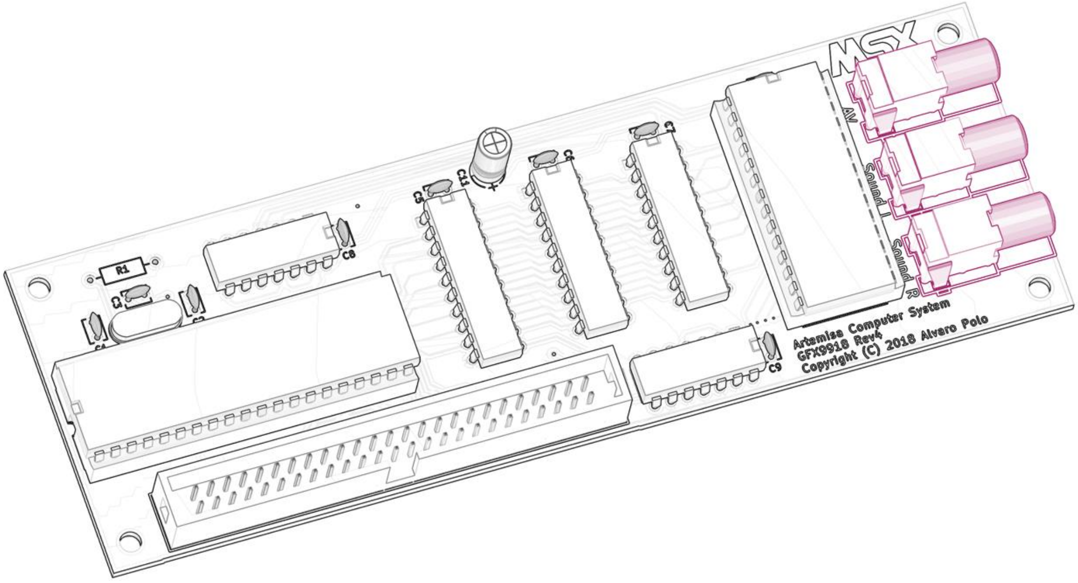


Figure 21: Placement of RCA connectors

Step 10

Solder the resistor R1 of 470Ω as shown in Figure 22.

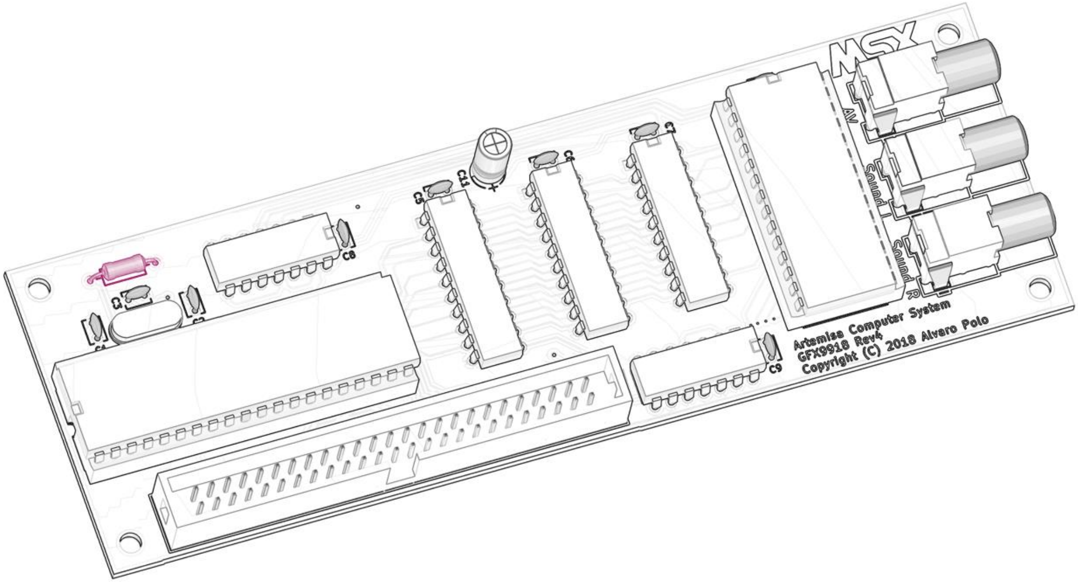


Figure 22: Placement of R1

Step 11

Just one more step and we are done. Now it is time to plug the VDP and the memory chip in their corresponding sockets.

⚠ Make sure the chips are plugged in the right direction. Look for the notch in the IC and the socket and ensure they are in the same side.

Summary

TBD

Troubleshooting