# SERVICE MANUAL

# MODEL C-128 COMPUTER

***Preliminary***

**AUGUST 1985 PN-314001-07**

**Commodore Business Machines, Inc.**
1200 Wilson Drive, West Chester, Pennsylvania 19380 U.S.A.

# CONTENTS

# COMMODORE 128 PERSONAL COMPUTER

**GENERAL FEATURES**

- Advanced Styling • 100% Compatible with Commodore 64
- Built-in, Easy to Use DOS support • RAM Expandable up to 512K RAM Using RAM Disk Option • Upper and Lower Case Character Set
- Built-in BASIC • 3 Separate Modes of Operation

**64 MODE**

- 8502 Microprocessor (6502/6510 Compatible) • 6581 Sound Interface Chip • 64K RAM • 16K ROM • BASIC 2.0 • 40 x 25 Lines (320 x 200 resolution) • 16 Colors + 8 Sprites

**128 MODE**

- 8502 Microprocessor (6502/6510 Compatible)
- 6581 Sound Interface Chip • 128K RAM (Expandable to 512K Using RAM Disk Option) • 48K ROM + 16K ROM for DOS Support
- BASIC 7.0 • Machine Language Monitor • 40 x 25 Lines (320 x 200 resolution) • 80 x 25 Lines (640 x 200 resolution) • 16 Colors + 8 Sprites (40 Column Only)

**CP/M MODE**

- Z80 Microprocessor • CP/M™ Plus Version 3.0
- 128K RAM (Expandable to 512K Using RAM Disk Option)
- 40 x 25 Lines (320 x 200 resolution) • 80 x 25 Lines (640 x 200 resolution) • 16 Colors

**KEYBOARD**

- Full Size Typewriter Style • 92 Keys • 14 Key Numeric Keypad
- 8 Programmable Function Keys • 6 Cursor Keys • Help Key
- 40/80 Column Key • No Scroll • Line Feed • Escape • Tab
- Cap Lock • Alt (Not all accessible in 64 Mode)

**INPUTS/OUTPUTS**

- User Port
- Cassette Port
- RF/TV Port
- Audio Input
- Composite Video
- Serial Port
- 2 Game Ports
- Cartridge Port
- Audio Output
- Digital RGBI Video

**RECOMMENDED PERIPHERALS**

- MPS 802, MPS 803, MPS 1000 Printers
- 1541, 1571 Single Disk Drive
- 1901 Monochrome Monitor • 1902 Digital RGBI Color Monitor
- 1660 and 1670 Modems • Fully Compatible with Commodore 64 Software and Accessories in 64 Mode

**POWER REQUIREMENTS**

- 117 Volts AC, 60 Hz, 15 Watts

†Specifications subject to change without notice

*CP/M is a registered trademark of Digital Research, Inc.

# PARTS LIST
# C-128

**PLEASE NOTE: Commodore part numbers are provided for reference only and do not *indicate* the availability of parts from Commodore. Industry standard parts (Resistors, Capacitors, Connectors) should be secured locally. Approved cross-references for TTL chips, Transistors, etc. are available in manual form through the Service Department. Unique or non-standard parts will be stocked by Commodore and are indicated on the parts list by a "C".**

## TOP CASE ASSY

| | |
|---|---|
| Top Case | C 251987-01 |
| Keyboard | C 310401-01 |
| Nameplate | C 310400-01 |
| Lamp Holder Set | C 252013-01 |
| LED Assembly | C 250754-01 |

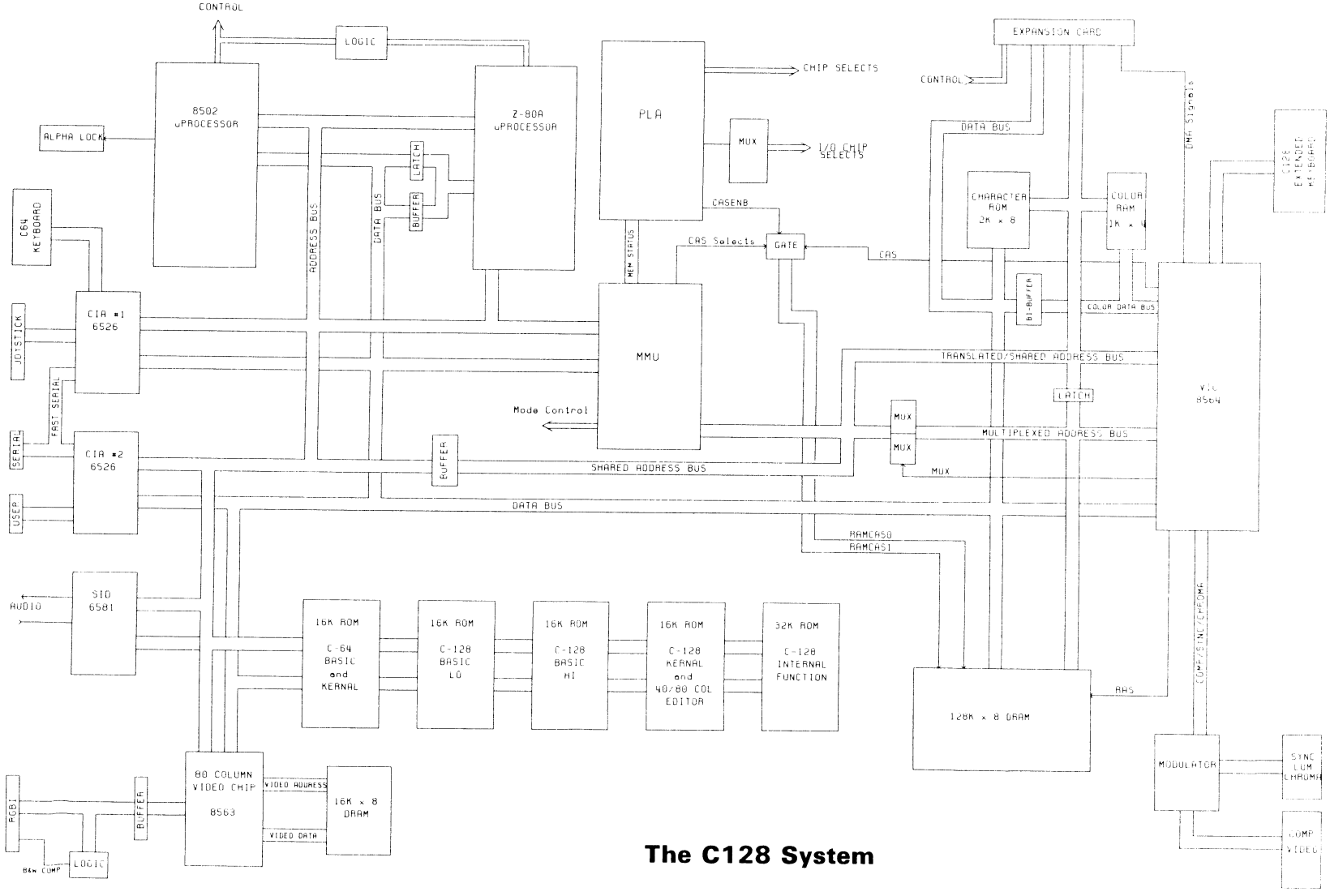
## BOTTOM CASE ASSY

| | |
|---|---|
| Bottom Case | C 251988-01 |
| Foot, Self-Adhesive | C 251993-01 |
| PCB Top Shield | C 252015-01 |
| PCB Bottom Shield | C 252016-01 |
| PCB Insulation Sheet | C 252017-01 |

## ACCESSORIES

| | |
|---|---|
| Users Manuals | |
| Introductory Guide | C 319773-01 |
| System Guide | C 310638-01 |
| Power Supply | C 310416-01 |
| RF Cable | C 326189-01 |
| Switch Box | C 904778-01 |
| Tutorial Diskette | C 317667-01 |
| CP/M Diskette #1 | C 317430-01 |
| CP/M Diskette #2 | C 317431-01 |

## BLOCK DIAGRAM




**The C128 System**

# BUS ARCHITECTURE

**FOLD OUT SCHEMATIC PAGES 64-67 FOR EASY REFERENCE.**

### The Processor Bus

The **Processor Bus** is the data and address buses that are directly connected to the 8502 processor. These buses are designated $D_0$ - $D_7$ for the eight bit data bus and $A_0$ -$A_{15}$ for the sixteen bit address bus. These buses tie the processor to most of the system ROM and I/O devices, including at least part of all System ROM, all built-in Function ROM, the MMU, the PLA, the 8563 Video Processor, the SID, and both CIA chips.

The Processor Bus is in direct communication with the Z-80 co-processor as well. All address lines are shared directly by both processors. In order to allow the Z-80 to operate on a 6502 family bus, it is necessary to latch data going into the Z-80 and gate the data leaving the Z-80. Thus, the Z-80 has a small local data bus, designated $ZD_0$ -$ZD_7$. During a write cycle, when AEC is high, Z-80 data is gated to the Processor Bus. During a read cycle, Processor Bus data is gated to the Z-80 data bus. This read data is transparently latched by the 1 MHz system clock.

The read and write cycles referred to are, unless otherwise specified, 8502 type bus cycles. The Z-80 Read Enable and Write Enable outputs are conditioned using logic to interface with an 8502 bus cycle, so no distinction is made as to the differences between cycles of the different processors.

As mentioned above, the Z-80 is not in direct communication with the Processor Data Bus, due to the necessity of adapting the Z-80 to 8502 bus protocol. Note, however, that every other device and the translated bus (except two that will be explained later) shares the Processor Data Bus as a common data bus.

### The Translated Address Bus

Another C128 system bus is the **Translated Address Bus**, which is produced by the MMU during AEC high. This bus consists of only high order addressing lines, designated $TA_8$ - $TA_{15}$. These lines reflect the action of the MMU on the normal high order address lines, which may or may not include some sort of translation. The MMU can translate the address of page zero or page one in normal operation, and it translates the Z-80 address from $0000 thru $0FFF in order to direct it to read the Z-80 BIOS. A more complete description of MMU translations can be found in the MMU section. Normally the Translated Address Bus indirectly drives the DRAMs and the VIC chip by driving the Multiplexed Address Buses. It directly drives System ROM 4 address line 12 to allow the Z-80 ROM relocation. Finally, this bus becomes address lines 8 thru 15 of the C64 compatible expansion port.

During a VIC cycle or a DMA, the MMU pulls $TA_{12}$ - $TA_{15}$ high, while $TA_8$ - $TA_{11}$ are tri-stated. This allows the VIC chip to drive $TA_8$ - $TA_{11}$ as VIC addresses $VA_8$ - $VA_{11}$.

### The Multiplexed Address Bus

This section actually describes two related address buses, the **Multiplexed Address Bus** and the **VIC Multiplexed Address Bus**, known respectively as $MA_0$ - $MA_7$ and $VMA_0$ - $VMA_7$. The VIC Multiplexed Address Bus is created during AEC high by multiplexing the high order Translated Address Bus ($TA_8$ - $TA_{15}$) with the low order Processor Address Bus ($A_0$ - $A_7$), controlled via the MUX signal. This bus, driven though series resistors, is called the Multiplexed Address bus. The VIC Multiplexed Address Bus is used in addressing the VIC chip registers while the Multiplexed Address Bus is the processor's DRAM address for both 64K banks of DRAM.

During a VIC cycle, AEC low, the VIC chip address lines must be asserted. There is no completely separate address bus for the VIC addresses, so it shares the $VMA_0$ - $VMA_7$ and $TA_8$ - $TA_{11}$ address lines that are otherwise tri-stated during AEC low. Most of the VIC addresses come out of the VIC chip already multiplexed, but two of them, $VA_6$ and $VA_7$. They do not supply column information, as the VIC chip supplies only fourteen bits of addressing. The higher order address bits $VA_{14}$ and $VA_{15}$ come from CIA 2, as in the C64. Thus, the VIC supplies complete $VMA_0$ - $VMA_7$ for a VIC DRAM access or DRAM refresh. The $TA_8$ - $TA_{11}$ supplied by VIC are used in conjunction with another addressing bus for non-multiplexed VIC cycle addresses, such as Character ROM and Color RAM accesses.

### The Shared Address Bus

The **Shared Address Bus** is a non-multiplexed address bus used by both the processor and the VIC chip. This is necessary to communicate with common resources, namely the Character ROM and Color RAM. During AEC high, the Shared Address Bus, designated $SA_0$ - $SA_7$, is driven by $A_0$ - $A_7$, the lower order Processor Address bits. The higher order bits needed are supplied by the Translated Address Bus, which is also a shared address bus. Thus, the processor is able to access both shared items.

During AEC low, the VIC addresses $VA_0$ - $VA_7$ ($VMA_0$ - $VMA_7$) must come onto the Shared Address Bus. Since $VA_0$ - $VA_6$ are actually multiplexed, the row address only must be sent to the Shared Address Bus. Thus, the Multiplexed VIC addresses are transparently gated when either $\overline{RAS}$ or MUX are low, but latched when both are high, which would indicate that a column address is about to be presented. The high order address bits, as well, are supplied by the shared Translated Address Bus. Note that the Shared Address Bus provides the lower eight bits of the expansion port address, allowing VIC access to cartridges and some additional drive capability by way of the TTL chips used to drive the Shared Address Bus.

### The Color Data Bus

The Color RAM is written to or read from by a nybble data bus called the **Color Data Bus**. During AEC high, the Color Data Bus is connected to the lower half of the Processor Data Bus via an analog switch, allowing the Processor full access to the Color RAM. During AEC low, that switch is opened, effectively isolating the Color Data Bus from the Processor Data Bus. In this state, it is driven by the VIC extended data bus $D_8$ - $D_{11}$.

### The Display Bus

The Display Bus is a bus local to the 8563 Video Controller VIC chip, consisting of the **Display Address**, $DA_0$ - $DA_7$, and the **Bus Display Data Bus**, $DD_0$ - $DD_7$. This local bus supports the 8563 display RAM, which is completely isolated from the rest of the C128 system. The Display Address Bus is a multiplexed address bus providing addressing to the display DRAM. The Display Data Bus provides communication between this DRAM and the 8563. The 8563 also provides row and column strobes and dynamic refresh to this DRAM.

# THE 8502 MICROPROCESSOR

**FOLD OUT SCHEMATIC SHEET 2, PAGE 65, FOR EASY REFERENCE.**

The 8502 is an HMOSII Technology microprocessor similar to the 6510/6502. It is the normal operating processor and is used in the C64 and the C128 modes. Fully software compatible with the 6510, hence the 6502, the 8502 also features a zero page port used in memory management and cassette implementations. The 8502 is also specified for operation at 2 MHz. The 2 MHz operation is made possible by removing the VIC from the system. The VIC chip is never completely removed from the C128 system, as it continues to function as clock generator and bus arbitrator. However, the VIC is removed as a display chip and co-processor, thus the full clock cycle can be devoted to processor functioning instead of sharing the cycle with the VIC. The task of video display processor is then taken over by the 8563, which can function without the need for bus sharing that the VIC required. Since the I/O devices, SID, etc., are rated at 1 MHz only, stretching of the 2 MHz clock is used to allow these parts to be directly accessed by the 2 MHz processor, and still keep throughput to a maximum. The I/O devices are not affected by the 2 MHz operation as they are still driven by a 1 MHz source (and as such, all timer operations remain unchanged), and clock stretching is only used to synchronize the 2 MHz machine cycle to the 1 MHz $\phi_0$ high time. The clock sources and clock stretching capabilities are generated by the VIC chip.

## CLOCK STRETCHING

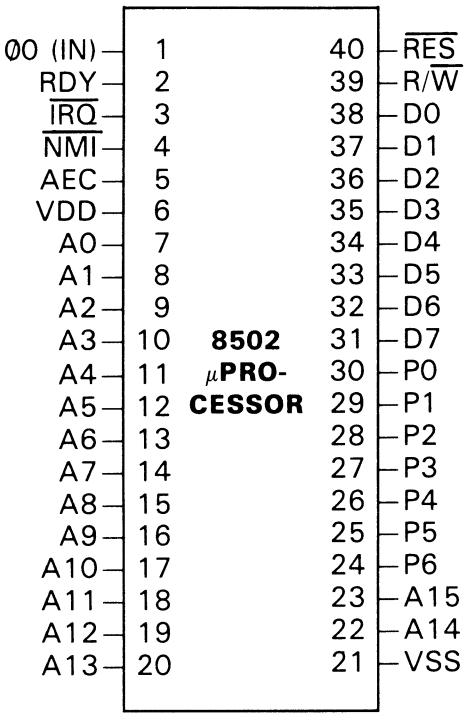
When running in 2 MHz mode, the processor clock sometimes must be stretched. This is handled by the VIC chip, the processor, and the PLA working together. When an I/O operation is decoded during a 2 MHz cycle, the phase relationship between the 2 MHz and the 1 MHz clocks must be considered. If the 2 MHz access occurs during 1 MHz $\phi_1$, the access to a clocked I/O chip would be out of synchronization with the 1 MHz clock that drives all I/O chips. Thus, during this phase relationship, $\overline{\text{IOACC}}$, from the PLA, signals the VIC chip to extend the 2 MHz clock. Should the 2 MHz cycles take place during the 1 MHz $\phi_2$ cycle, no special attention is necessary.

**Clock Stretching in 2 MHz Mode**

Please take note to consider the speed implications of this. In 2 MHz mode, half of the I/O accesses given will occur at an effective speed of 1 MHz. For time critical operations, then, accesses to I/O chips are kept at a minimum.

## THE 8502 MICROPROCESSOR (Continued)

### 315020
### 8502 MICROPROCESSOR




| | | |
|---|---|---|
| 1 | Ø0 | Phase 0 clock input. This is the dual speed system clock for the 128. |
| 2 | RDY | Ready. TTL level input, used to DMA the 8502. The processor operates normally while RDY is high. When RDY makes a transition to the low state, the processor will finish the operation it is on, and any subsequent operation if it is a write cycle. On the next occurrence of read cycle the processor will halt, making it possible to tri-state the processor to gain complete access to the system bus. |
| 3 | IRQ | The Interrupt Request input is a request that the processor initiate an interrupt sequence. The processor will complete execution of the current instruction before recognizing the request. At that time, the interrupt mask in the Status Code Register will be examined. If the interrupt mask is not set, the processor will begin an interrupt sequence. The Program Counter and the Processor status register will be stored on the stack and the interrupt disable flag is set so that no other interrupts can occur. The processor will then load the program counter from the memory location $FFFE and $FFFF. |
| 4 | NMI | The Non-Maskable Interrupt Request is a negative-edge sensitive request that the processor initiate an interrupt sequence. The processor will complete execution of the current instruction before recognizing the request. The Program Counter and the processor status register will be stored on the stack. The processor will then load the program counter from the memory locations $FFFA and $FFFB. |
| 5 | AEC | The Address Enable Control. The Address Bus is only valid when the AEC line is high. When low, the address bus is in a high impedance state. This allows DMA's for dual processor systems. |
| 6 | VDD | 5VDC input. |
| 7-20<br>22,23 | A0-A15 | Address bus outputs. Unidirectional bus used to address memory and I/O devices. The address bus can be disabled by controlling the AEC input. |
| 21 | VSS | DC ground. |
| 24-30 | P0-P6 | Bidirectional I/O port used for transferring data to and from the processor directly. The Data Register is located at location $0001 and the Data Direction Register is located at location $0000. |
| 31-38 | D0-D7 | Bi-directional bus for transferring data to and from the device and the peripherals. |
| 39 | R/W | The read/write line is a TTL level output from the processor to control the direction of data transfer between the processor and memory, peripherals, etc. This line is high for reading memory and low for writing. |
| 40 | RES | The Reset input is used to reset or start the μprocessor from a power down condition. During the time that this line is held low, writing to or from the μprocessor is inhibited. When a positive edge is detected on the input, the μprocessor will immediately begin the reset sequence. After a system initialization time of 6 cycles, the mask interrupt flag will be set and the processor will load the program counter from the contents of the memory locations $FFFC and $FFFD. This is the start location for program control. After $V_{CC}$ reaches 4.75 volts in a power up routine, reset must be held low for at least 2 cycles. At this time the R/W line will become valid. |

# THE Z-80 MICROPROCESSOR

**FOLD OUT SCHEMATIC SHEET 2, PAGE 65, FOR EASY REFERENCE.**

The Z-80 microprocessor is used as a secondary processor in the C128 to run CP/M based programs. The Z-80 is interfaced to the 8502 bus and can access all of the devices that the 8502 can access.

### Bus Interface

Since a Z-80 bus cycle is much different than a 65xx family bus cycle, a certain amount of interfacing is required for a Z-80 to control a 65xx type bus. Since the Z-80 has built-in bus arbitration control lines, it is possible to isolate the Z-80 by tri-stating its address line. Thus, the Z-80 and 8502 both share common address lines.

The data lines do not interface quite as easily. Due to the shared nature of the bus during Z-80 mode, it is necessary to isolate the Z-80 from the bus during AEC low. Thus, a tri-stable buffer must drive the Processor bus during Z-80 data writes. The reverse problem occurs during a Z-80 read — the Z-80 must not read things that are going on during AEC low. It must latch the data that was present during AEC high. Thus, a transparent latch drives the data input to the Z-80. It is gated by the Z-80 Read Enable output, and latched when the 1 MHz clock is low. It will be seen that the Z-80 actually runs during AEC low, but that the data bus interfaces with it only during AEC high.

### Control Interface

The Z-80 control interfacing must provide useful clock pulses to the Z-80 and must tailor the Z-80 Read and Write Enable signals for the 8502 type bus protocol. The Z-80 clock is provided by the VIC chip, and is basically a 4 MHz clock that only occurs during $\phi_0$ low, as seen in the Z-80 bus timing diagram. This insures that the Z-80 is only clocked when it is actively on the bus. One additional problem that arises in clocking the Z-80 is that while all of the 8502 levels, and most of the Z-80 levels, are TTL compatible, the Z-80 clock input expects levels very close to five volts. For that reason, the ouptut from the VIC chip is processed by a transistor switching circuit to give a full amplitude clock. This circuit uses the nine volt supply, thus, the nine volt circuit **must** be operational for the Z-80 to function.

**Z-80 Bus Timing**

The Z-80 is designed to have explicit Read, Write and I/O cycles, where an I/O cycle is distinct from a memory cycle. The 65xx family uses only memory mapped I/O and thus, for a 65xx bus, all I/O devices appear as memory locations, and all non-write cycles appear as read cycles. The Z-80 communicates cycle information via two control lines, the Read Enable and Write Enable lines. The C128 uses the Read Enable line of the Z-80 to gate the Processor Bus data to the Z-80 data bus. The Write Enable interfacing is somewhat more complicated.

The Write Enable Circuitry consists of a rising-edge triggered D-Type flip-flop and an SR flip-flop. The D-flop is triggered by the rising-edge of the 1 MHz clock. The positive output of the SR drives the D-input, and the Q output gated with AEC drives an open collector inverter, which in turn drives the R/W line of the 8502 bus. The $\overline{S}$ input is driven by the Z-80 $\overline{WE}$, and the $\overline{R}$ input is driven by the Q output of the D-flop. Normally the D-input is low, resulting in an 8502 read cycle. When the Z-80 $\overline{WE}$ signal falls, it sets the SR flop, causing the D-input to rise. This line remains high, even if the Z-80 $\overline{WE}$ should rise again. When the 1 MHz clock rises, this high level is clocked, causing an 8502 write cycle that will last one complete 1 MHz cycle. When the Write signal is passed by the D-flop, the $\overline{Q}$ output will reset the SR flop. If no more $\overline{WE}$ signals come, the D-flop will once again set 8502 Read mode.

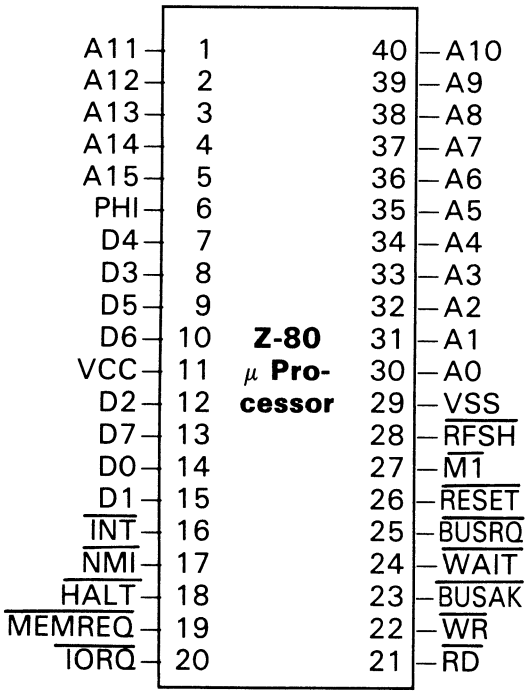
### Processor Switching

It is important in normal operation for the Z-80 and 8502 to operate as co-processors, communicating between each other. This is, however, only serial co-processing, not to be considered parallel co-processing or multiprocessing. Only one processor may have the bus at any one time. This is important in several ways. First, the C128 system must power up with the Z-80 as the master processor. This is because the Z-80 will not power up cleanly, and may accidently access the bus when powering up. Thus, it is made master on powerup and can do anything it likes to the bus. Also, the Z-80 can start up certain C64 applications that would cause the 8502 to crash, thus again it is the logical choice for startup processor. After some initializations, the Z-80 will start up the 8502 in either C128 or C64 mode, depending upon if a cartridge is present.

The second reason for processor switching is to allow the Z-80 to access 8502 Kernal routines. For standardized programs, or for any I/O operation not supported in the Z-80 BIOS, the Z-80 can pass on the task of I/O to the 8502. Since the Z-80 sees BIOS ROM where the 8502 sees its pages 0 through F, the Z-80 can operate without fear of disrupting any 8502 pointers or the stack in RAM Bank 0.

The Z-80 can receive a bus grant request from the MMU, via $\overline{Z80EN}$, or from the VIC chip, via BA. Since the VIC control line is used for DMAs, that is not of immediate concern. The $\overline{Z80EN}$ action, however, is, since it is the mechanism by which processors swap control. When the $\overline{Z80EN}$ line goes high, it triggers a Z-80 $\overline{BUSRQ}$. The Z-80 will relinquish the bus by pulling $\overline{BUSACK}$ low. This action drives the 8502 AEC high and, providing VIC does not request a DMA, will also drive the 8502 RDY line high, enabling the 8502. To switch back, a low on the Z-80 $\overline{BUSRQ}$ will result in Z-80 $\overline{BUSACK}$ going high, tri-stating and halting the 8502.

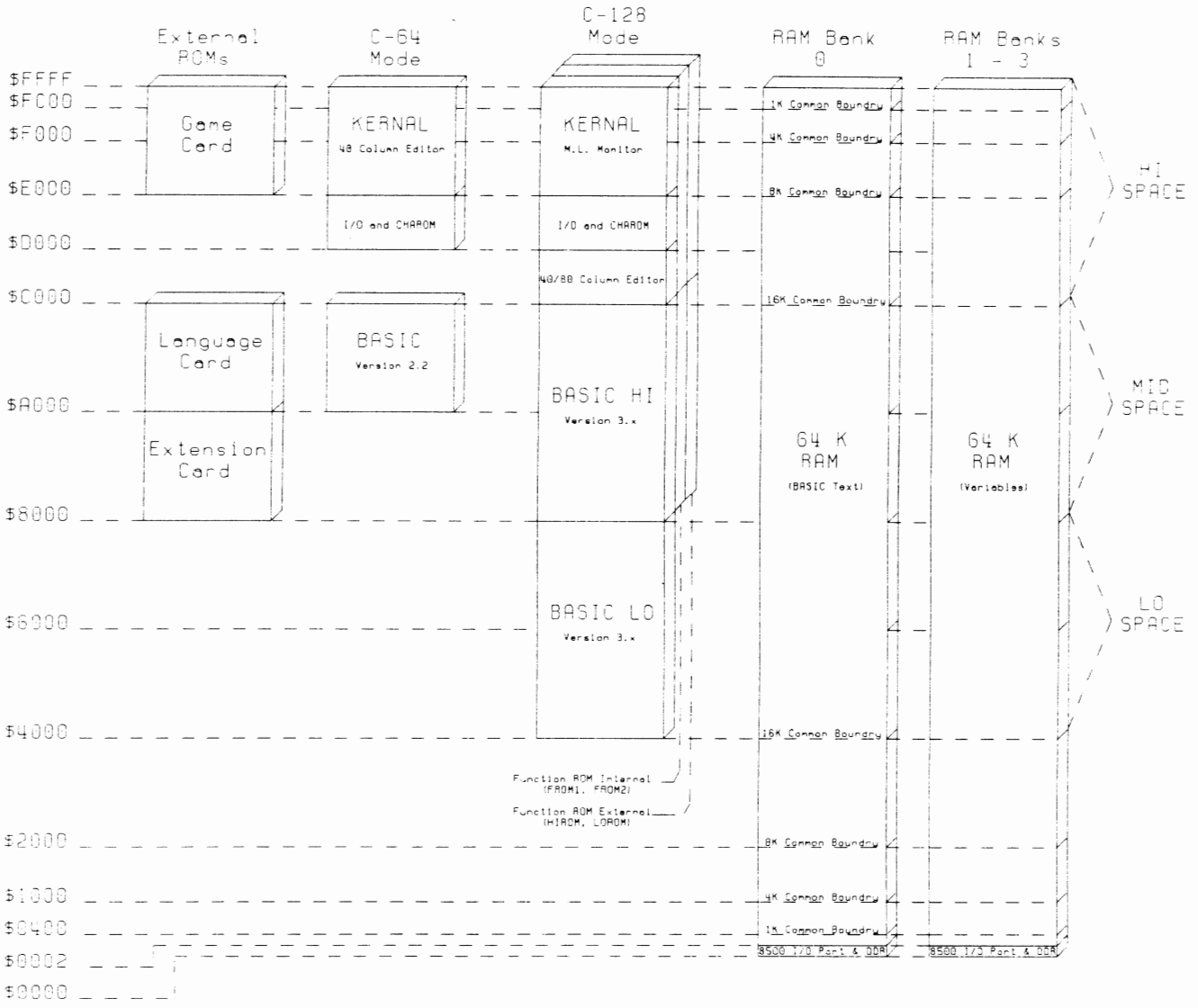
## 906150
## Z-80 MICROPROCESSOR




| | | |
|---|---|---|
| 1-5, 30-40 | A0-A15 | 16 Bit tri-stating Address Bus. Used for 16 bit memory address during memory cycles, used for 8 bit I/O address during I/O cycles. This allows up to 256 input or 256 output ports. During refresh time, the lower 7 bits contain a valid refresh address. |
| 6 | PHI | Single phase system clock. |
| 7-10, 12-15 | D0-D7 | Input/Output Data Bus, capable of tri-stating, used for 8-bit data exchanges with memory and I/O devices. |
| 11 | VCC | 5VDC input. |
| 16 | INT | Interrupt Request. Active low input, driven by external devices. If the interrupt flag IFF is enabled, and the $\overline{BUSRQ}$ line is not active, the processor will honor the requested interrupt at the end of the current instruction. When the Z-80 acknowledges an interrupt, it generates an interrupt acknowledge signal ($\overline{IORQ}$ during $M_1$) at the beginning of the next instruction cycle. There are three different modes of response to a given interrupt. |
| 17 | NMI | Non-Maskable Interrupt. Active low input. This interrupt is edge triggered and cannot be masked against. It is always recognized at the end of the current instruction, forcing the Z-80 to take a restart at location $0066. The program counter is automatically saved in the stack to allow a return from the interrupted program. Note that continuous WAIT cycles can delay an $\overline{NMI}$ by preventing the end of the current cycle, and that $\overline{BUSRQ}$ will override $\overline{NMI}$. |
| 18 | HALT | Halt State. Active low output indicating that the Z-80 has executed a HALT instruction and is awaiting some kind of interrupt before execution can continue. While in the $\overline{HALT}$ state, the CPU continuously executes NOPs to continue refresh activity. |
| 19 | MEMREQ | Memory Request. Active low, tri-state output that indicates that the address bus holds a valid address for a memory read or write operation. |
| 20 | IORQ | Input/Output Request. Active low, tri-state output. The $\overline{IORQ}$ signal indicates that the lower half of the address bus holds a valid address for an I/O read or write operation. An $\overline{IORQ}$ signal is also generated with an $\overline{M_1}$ signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. An interrupt can be acknowledged during $\overline{M_1}$; I/O operations never occur during $\overline{M_1}$. |
| 21 | RD | Memory Read. Active low, tri-state output. $\overline{RD}$ indicates that the CPU wants to read data from memory or from an I/O device. This signal is generally used to gate read data onto the data bus. |
| 22 | WR | Memory Write. Active low, tri-state output. $\overline{WR}$ indicates that the data bus holds valid data to be processed by memory or by an I/O device. |
| 23 | BUSAK | Bus Acknowledge. Active low output, used to indicate to any device taking over the bus that the Z-80 has gone into tri-state and the bus has been granted. While in this mode it cannot refresh dynamic memory. |
| 24 | WAIT | Wait. Active low input, used to drive the Z-80 into wait states. As long as this signal is low, the Z-80 will execute wait states, allowing this signal to be used to access slow memory and I/O devices. While the Z-80 is in a $\overline{WAIT}$ state, it cannot refresh dynamic memory. |
| 25 | BUSRQ | Bus Request. Active low input that is used to request the CPU address, data, tri-statable output control signals to all go tri-state for bus sharing and DMAs. The lines go tri-state upon termination of the current machine cycle. |
| 26 | RESET | Reset. Active low input which forces the program counter to zero and initializes the Z-80, which will set interrupt mode 0, disable interrupts, and set registers I and R to zero. During $\overline{RESET}$, address and data buses tri-state and all other signals go inactive. |
| 27 | $M_1$ | Machine Cycle One. Output, active low. This signal indicates that the current machine cycle is the OP code fetch of an instruction execution. During execution of a two byte op-code, $\overline{M_1}$ is generated as each byte is fetched. $\overline{M_1}$ also occurs with $\overline{IORQ}$ to indicate an interrupt acknowledge cycle. |
| 28 | RFSH | Refresh. Active low output used to indicate that the address bus holds a refresh address in its lower seven bits. Thus, the current $\overline{MREQ}$ signal should be used to do a refresh read to all dynamic memories not refreshed from an alternate source. $A_7$ is set to zero and the upper eight bits contain the I register at this time. |
| 29 | VSS | Ground. |

# MEMORY ARCHITECTURE

**FOLD OUT SCHEMATIC PAGES 64-67 FOR EASY REFERENCE.**




**C128 Memory Map**

## C128 ROM Memory Organization

The memory map is an important consideration in maintaining C64 compatibility. The standard map is shown for the C64 mode. The C128 basically becomes a C64 when in C64 mode.

C128 mode is achieved at system reset, and is controlled by a bit in the MMU configuration register (See MMU Circuit Theory, page 20). In C128 mode, the MMU asserts itself in the C128 memory map at $FF00 *and* in the I/O space starting at $D500. Use of MMU registers, located at $FF00, allows memory management without actually having the I/O block banked in at the time and with a minimum loss of contiguous RAM. The MMU is removed from the memory map in C64 mode but is still used by hardware to manage memory.

The ROMs in C64 mode, both internally and externally, look just like C64 ROMs. The internal BASIC and KERNAL provide the C64 mode with the normal C64 operating system in ROM. This ROM actually duplicates some of the ROM used in C128 mode, but is necessary, as it is not accessible from C128 mode. In C128 mode, up to 48K of Operating System is present, with the exact amount being set by software control. This allows quicker access to underlying RAM by turning off unneeded sections of the Operating System.

The External ROMs represented on the memory map are those used in the C64 mode, and obey the C64 rules for mapping, i.e., cartridges assert themselves in hardware via the $\overline{\text{EXROM}}$ and $\overline{\text{GAME}}$ lines. External ROMs in C128 mode are mapped as banked ROMs, such that when the system is initialized, all ROM slots are polled for the existance of a ROM and the ROM's priority if one exists. This allows much more flexibility than the hardwire ROM substitution method, since the Kernal and Basic ROMs can be swapped out for an application program, swapped out for external program control, or turned off all together. This banking manipulation is accomplished by writing to the Configuration Register at location $D500 or $FF00, in the MMU.

The hardware also features the ability to store preset values for the configuration and force a load of the Configuration Register by writing to one of the LCR (Load Configuration Register) registers.

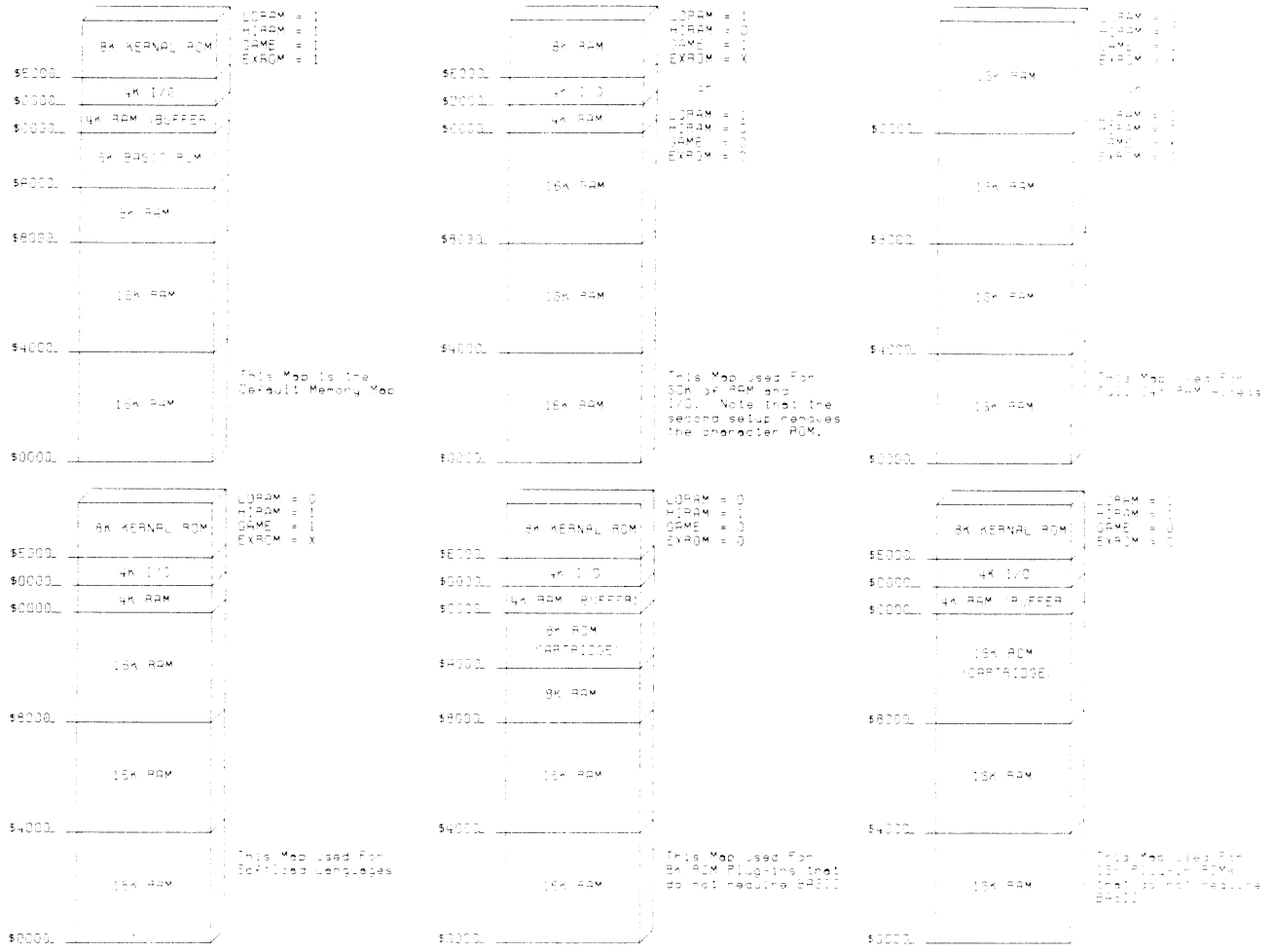
### C128 RAM Memory Organization

Refer again to the **C128 Memory Map.** The RAM present in the system is actually composed of two 64K by 8 bytes of contiguous DRAM. The RAM is accessed by selecting one of the two banks of 64K according to the RAM banking rules set in the RAM Configuration Register of the MMU. The area shown as RAM is representative of what the $\mu$Processor would see if all ROM were disabled. Bank switching can be accomplished in one of two ways.

The bank in use is a function of the value stored in the Configuration Register. A store to this register will always take effect immediately. An indirect store to this register, using preprogrammed bank configuration values, can be accomplished by writing to one of the **indirect load** registers, known as LCRs (Load Configuration Register), located in the $FF00 region of memory. By writing to an LCR the contents of its corresponding PCR (PreConfiguration Register) will be latched into the configuration register. Refer to the MMU section on page 20 and the Alternate Memory Configurations on the following page.

When dealing with 64K banks of memory at once, it may be desirable to bank in bank 1 but still retain the system RAM (Stack, Zeropage, Screen, etc.). The MMU has provisions for what is referred to as **common RAM.** This is the RAM that does not bank, and is programmable in size and as to whether it appears at the top, bottom, or both in the memory map. The size is set by bits 0 and 1 in the **RAM Configuration Register (RCR).** If the value of the bits is zero, 1K will be common. Values of one, two, and three produce common areas of 4K, 8K, and 16K respectively. If bit 2 of the RCR is set, bottom memory is held common, if bit 3 is set, then top memory is common. In all cases, common RAM is physically located in bank 0.

## MEMORY ARCHITECTURE (Continued)



**C64 Alternate Memory Configurations**

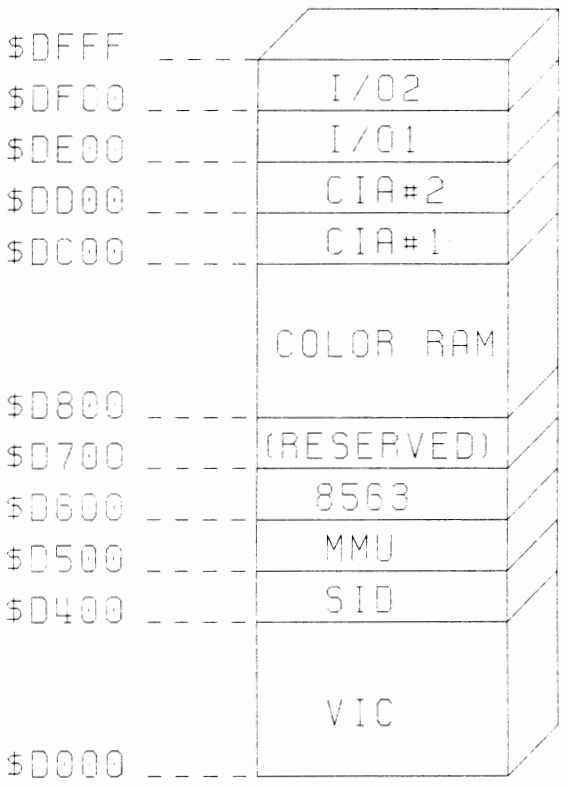
Zero page and page one can be located (or relocated) independently of the RCR. When the processor accesses an address that falls within zeropage or page one, the MMU adds to the high order $\mu$Processor address, the contents of the P0 register pair or the P1 register pair, respectively, and puts this new address on the bus, including the extended addressing bit $A_{16}$. RAM banking will occur as appropriate to access the new address. Writes to the P0 and P1 registers will be stored in a prelatch, until a write to the respective $P_{XL}$ register occurs. This prevents a $P_{XH}$ register from affecting the translated address until both high and low bytes have been written.

At the same time, the contents of the P0 and P1 registers are applied to a digital comparator, and a reverse substitution occurs if the address from the 8502 falls within the page pointed to by the register. This results in not just relocating the zero or one page but swapping the zero or one page with the memory that it replaced. Swapping only occurs if the swapped area is defined as RAM, i.e., System or Function ROM must always be at their assigned addresses and thus should not be back-substituted. Note that upon system reset, the pointers are set to true zero and true one page.

For VIC chip access, one bit in the MMU status register is substituted for extended address line A16, selecting the proper CAS enable to make it possible to steer the VIC to anywhere in the 128K range. Note that AEC is the mechanism that the MMU uses to steer a VIC space address, i.e., when AEC is low a VIC access is assumed. This results in the VIC bank being selected as well for an outside DMA, since this too will pull the AEC line low.

### MMU and I/O Memory Organization

The block of memory represented by the **I/O Block** is an expanded view of the memory block entitled **I/O + CHAROM**, as shown in the C128 memory map. When the I/O block exists, access to VIC, SID, and I/O, as well as the addition of the MMU can be accomplished. All I/O functions remain as they were previously on the C64 with the exception that the MMU and the 80 Column chip have been added. With the exception of four registers that are asserted in the zero page in C128 mode, all new MMU registers appear in an unused slot in the I/O Memory block, though they will only appear in C128 mode. Detailed descriptions of the MMU registers can be found in the MMU section on page 20.




**I/O Block**

# READ ONLY MEMORY

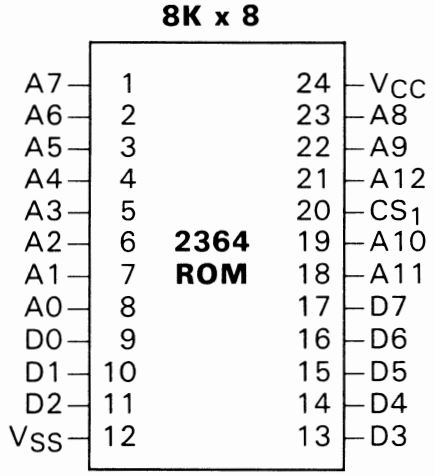
**FOLD OUT SCHEMATIC SHEETS 3 AND 4, PAGES 66 AND 67, FOR EASY REFERENCE.**

In C64 mode, the operating system resides in 16K of ROM, which includes approximately 8K for Kernal and 8K for Basic. In C128 mode, the operating system resides in 48K of ROM and includes advanced Kernal and Basic features. The Kernal, by definition, is the general operating system of the computer, with fixed entry points into usable subroutines. The entry table for the Kernal is located in memory at addresses $FF40 - $FFF9, exluding of course the MMU registers at $FF00 - $FF04. There is also a CHARACTER ROM, 8K x 8, which resides on the Shared Bus, shared by the VIC chip and the processor. The C64 OS ROM is wired so as to appear as two chunks of non-contiguous ROM, copying the actual C64 ROM memory map. Provision is included to handle system ROM as either four 16K x 8 ROMs or as two 32K x 8 ROMs. All internal C128 function ROMs will be the 32K x 8 variety.

### Rom Banking

Refer to the **MMU Register Map** on page 20. Note that the Configuration Register (CR) controls the type of ROM or RAM seen in a given address location. Dependent on the contents of the CR, ROM may be enabled and disabled to attain the most useful configuration for the application at hand. ROM is enabled in three memory areas in C128 mode, each consisting of 16K of address space. The lower ROM may be defined as RAM or System ROM, the upper two ROMs may be System ROM, Function ROM, Cartridge ROM, or RAM. In C64 mode the C64 memory mapping rules apply, which are primitive compared to those used in C128 mode. C64 ROM is banked as two 8K sections, BASIC and KERNAL, according to the page zero port and the cartridge in place at the time. No free banking can take place when a cartridge is in place.

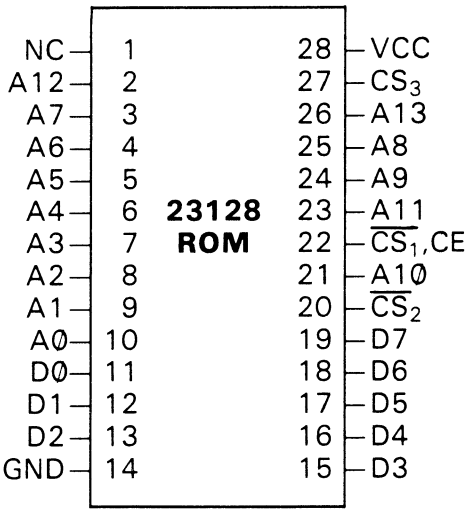
In the C128, if an address falls into the range of an enabled ROM, the MMU will communicate the status of ROM to the PLA decoder via the Memory Status lines. Essentially, the MMU looks up in the Configuration Register which ROM or RAM is set. The various combinations possible are shown on the **C128 Memory Map** found on page 11. The banking scheme, the way it is implemented, allows up to 32K of internal, bankable ROM for use such as Function Key Applications programs, and will support 32K of external bankable ROM. Various combinations of ROM are possible, and can be noted by studying the configurations for the Configuration Register.




| | | |
|---|---|---|
| 1-8, 18,19, 21-23 | A0-A12 | Address Bus Inputs. |
| 12 | $V_{SS}$ | Ground. |
| 9-11, 13-17 | D0-D7 | Data Outputs. |
| 20 | $CS_1$ | Chip Select. |
| 24 | $V_{CC}$ | 5VDC Input. |

## READ ONLY MEMORY (Continued)

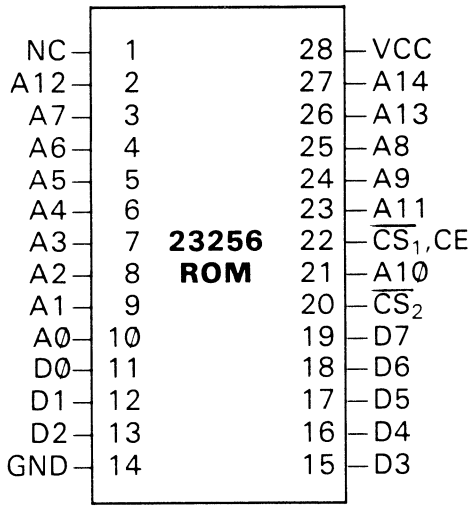
### PIN CONFIGURATION




**16K x 8 ROM**

| | | |
|---|---|---|
| 1 | NC | Not Connected. |
| 2-10, 21, 23-26 | A0-A13 | Address Bus Inputs. |
| 11-13, 15-19 | D0-D7 | Data Outputs. |
| 14 | GND | Ground. |
| 20 | $CS_2$ | Chip Select. |
| 22 | $CS_1$, CE | Output Enable. |
| 27 | $CS_3$ | Program Enable. |
| 28 | VCC | 5VDC Input. |

### PIN CONFIGURATION




**32K x 8 ROM**

| | | |
|---|---|---|
| 1 | NC | Not Connected. |
| 2-10, 21, 23-27 | A0-A14 | Address Bus Inputs. |
| 11-13, 15-19 | D0-D7 | Data Outputs. |
| 14 | GND | Ground. |
| 20 | $CS_2$ | Chip Select. |
| 22 | $CS_1$, CE | Output Enable. |
| 28 | VCC | 5VDC Input. |

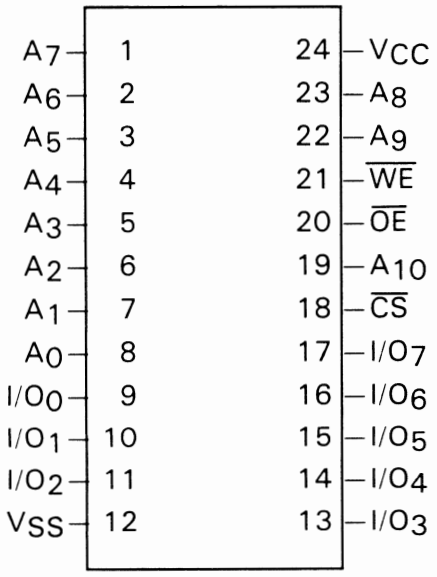
# RANDOM ACCESS MEMORY

**FOLD OUT SCHEMATIC SHEETS 3 AND 4, PAGES 66 AND 67 FOR EASY REFERENCE.**

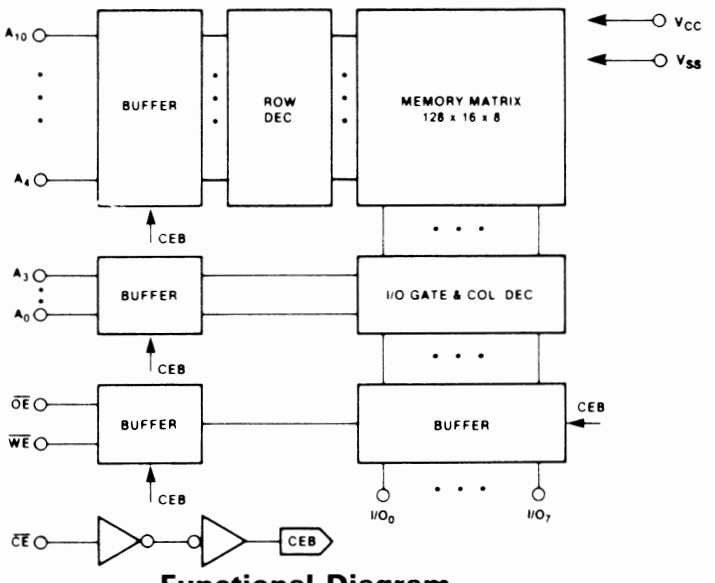
The C128 System contains 128K of processor-addressable 4164 DRAMs in the 64K x 1 configuration, organized into two individual 64K banks. Additionally, the system contains 16K of video display 4416 DRAMs (16K x 4) local to the 8563 CRT Controller, and 8K of STATIC RAM used as VIC COLOR RAM.

RAM banking, described in detail in the MMU section, is controlled by several MMU registers: the Configuration Register, the RAM Configuration Register, and the Page Zero and Page One Pointers. Simply put, the Configuration register controls which 64K bank of RAM is selected, the RAM Configuration Register controls if and how much RAM is kept in common between banks, and the Pointer registers redirect the zero and one pages to any page in memory, overriding the effect of the two configuration registers. In the system, RAM bank select is achieved via gated CAS control.

**2016**
**2K x 8 STATIC RAM**

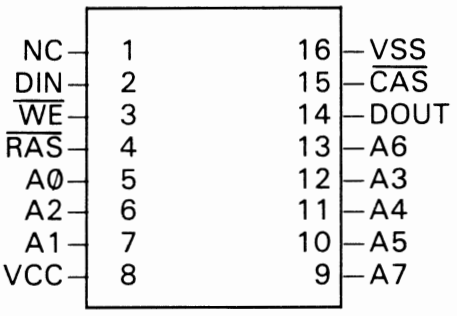



| | | |
|---|---|---|
| 1-8, 19, 22 23 | $A_0$-$A_{10}$ | Address Bus Inputs. |
| 9-11, 13-17 | $I/O_0$-$I/O_7$ | Common Data Input/Output Lines. |
| 12 | $V_{SS}$ | Ground. |
| 18 | CS | Chip Select Enable, Low Active. |
| 20 | OE | Output Enable, Low Active. |
| 21 | WE | Write (Input) Enable, Low Active. |
| 24 | $V_{CC}$ | 5VDC Input. |

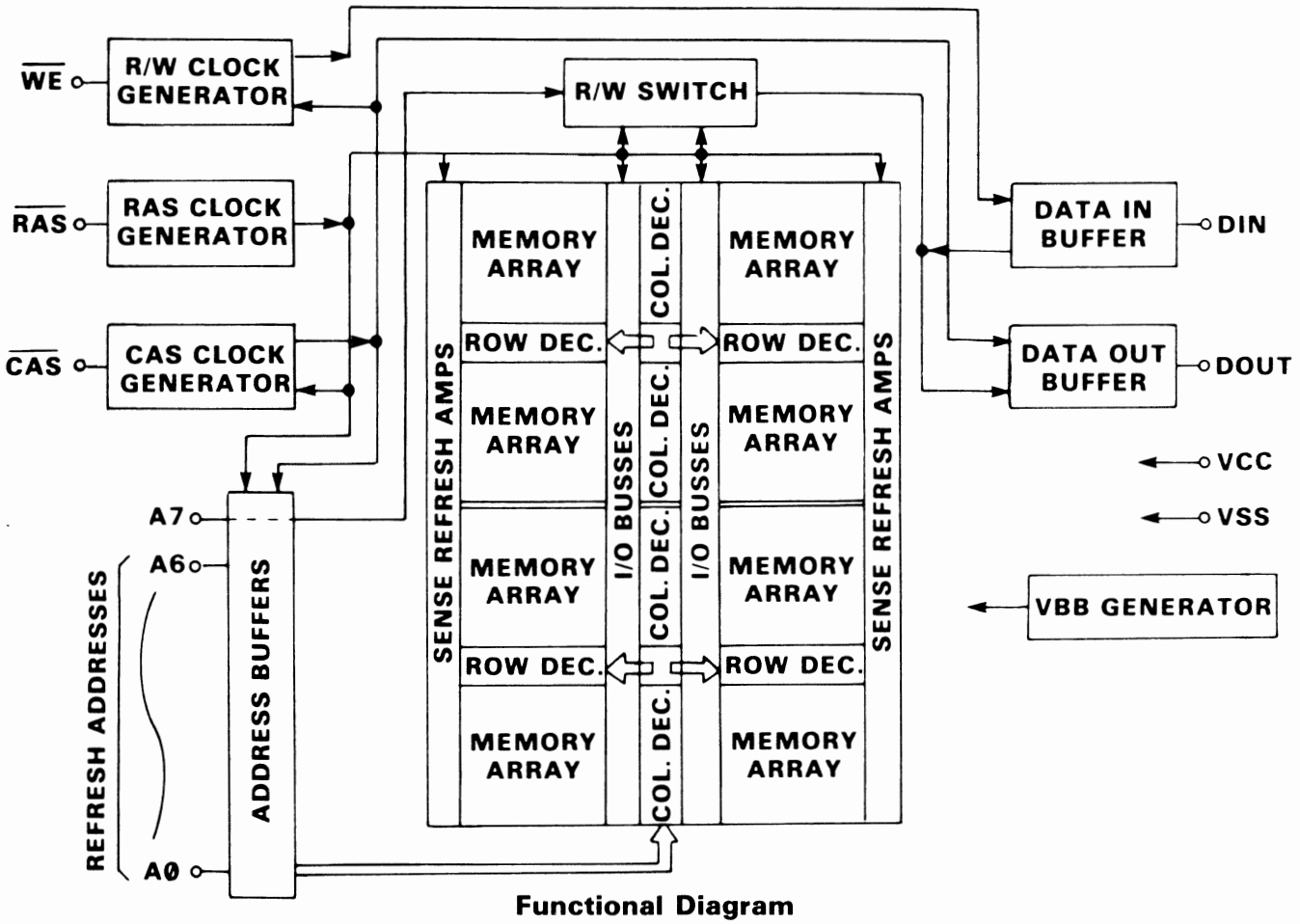



**Functional Diagram**

## 4164
## 64K x 1 DYNAMIC RAM

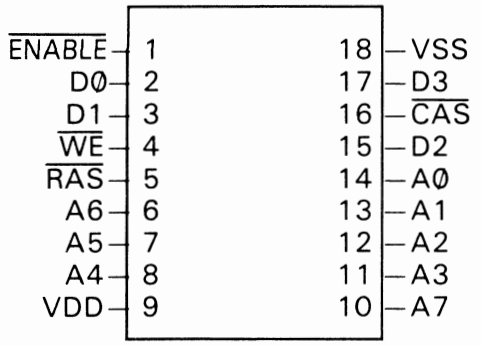



| | | |
|---|---|---|
| 1 | NC | Unused. |
| 2 | DIN | Data Input. |
| 3 | WE | Write Enable. Low active control input. |
| 4 | RAS | Row Address Strobe Input. Low active. |
| 5-7, 9-13 | A0-A7 | Address Bus Inputs. |
| 8 | VCC | 5VDC Input. |
| 14 | DOUT | Data Output. |
| 15 | CAS | Column Address Strobe Input. Low active. |
| 16 | VSS | Ground. |

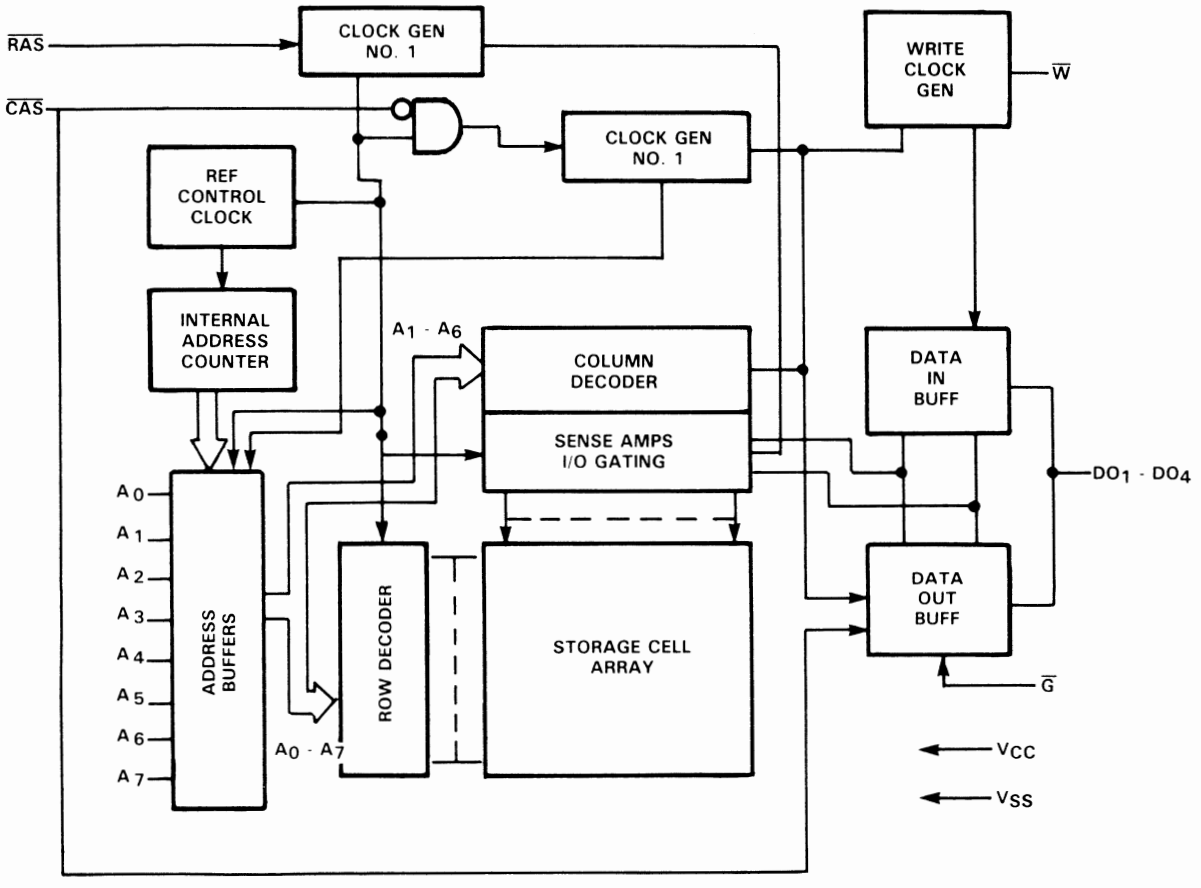



**Functional Diagram**

## 4416
## 16K x 4 DYNAMIC RAM




| | | |
|---|---|---|
| 1 | ENABLE | Output Enable ($\overline{G}$). |
| 2,3, 15,17 | D0-D3 | Common Data Input/Output Lines. |
| 4 | WE | Write (Input) Enable. Low Active. |
| 5 | RAS | Refresh Address. Low Active. |
| 6-8, 10-14 | A0-A7 | Address Bus Inputs. |
| 9 | VDD | 5VDC Input. |
| 16 | CAS | Column Address Strobe. Low Active. |
| 18 | VSS | Ground. |




**Functional Diagram**

# THE MEMORY MANAGEMENT UNIT

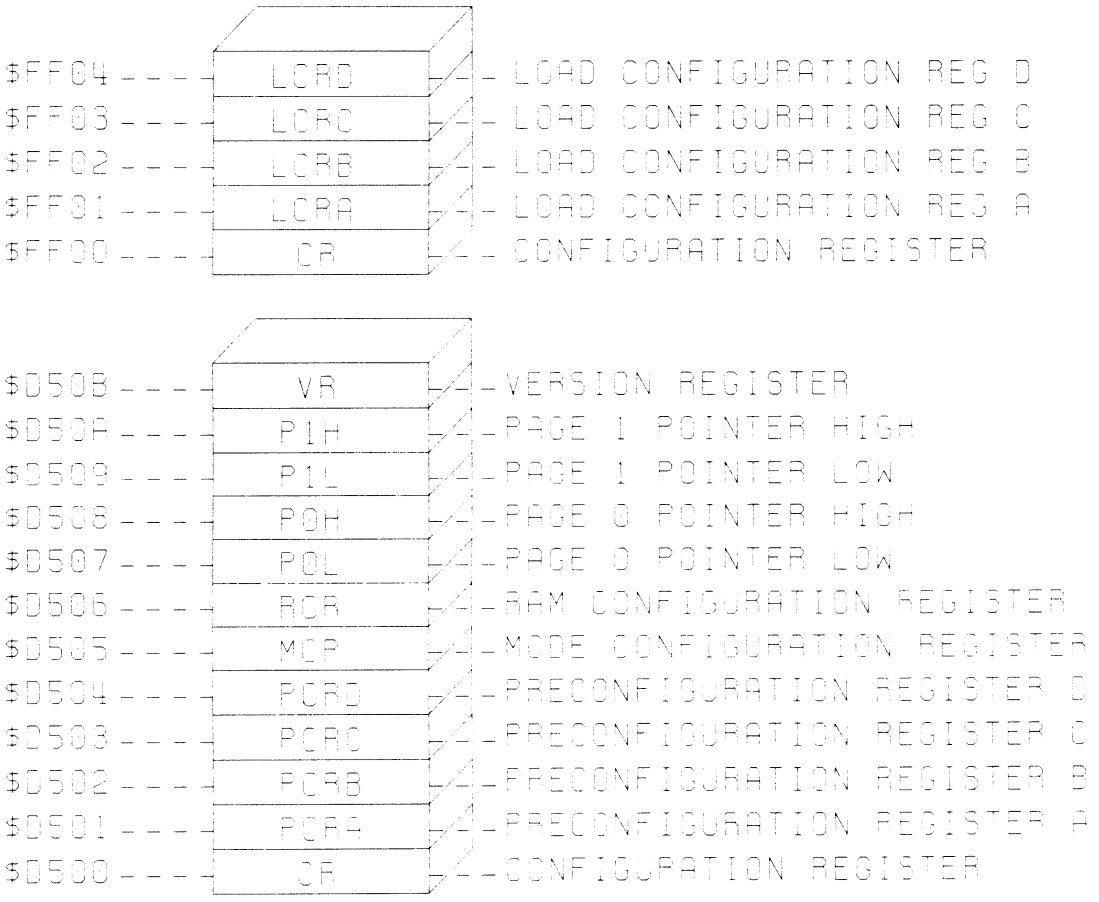
**FOLD OUT SCHEMATIC SHEET 2, PAGE 65, FOR EASY REFERENCE.**

The MMU is designed to allow complex control of the C128 system memory resources. It handles all of the standard **C64 modes of** operation in a fashion as to be completely compatible with the C64. Additionally, it controls the management of particular C128 modes including the Z-80 mode.

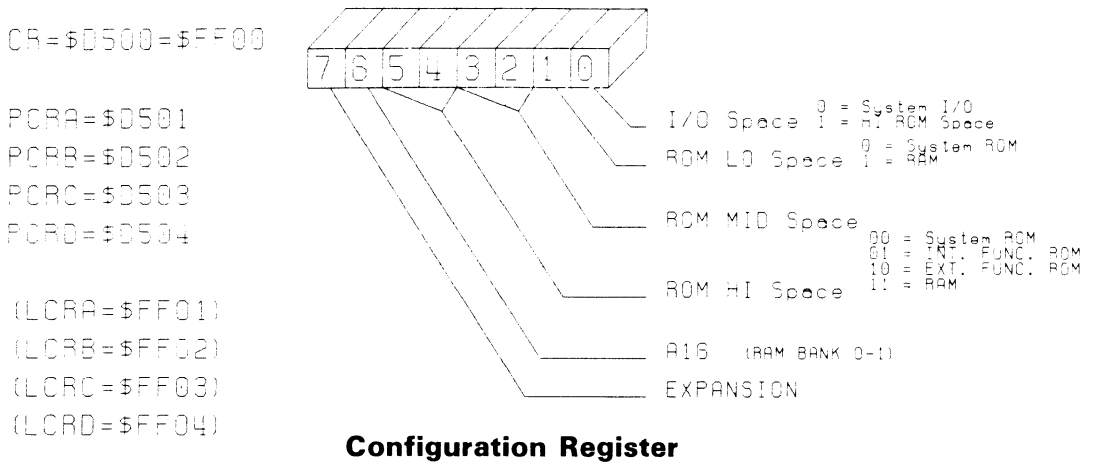
Summary of MMU functions:

- Generation of Translated Address Bus, $TA_8 - TA_{15}$.
- Generation of control signals for different processor modes — C128, C64, Z-80.
- Generation of CAS select lines for RAM banking.
- Generation of ROMBANK ($MS_0$, $MS_1$) lines for ROM banking.

The MMU is the mechanism by which the various memory modes shown in the C128 Memory Map are chosen. Additionally, the MMU provides for Z-80 mode, which was not shown on that diagram. Following is a description of the MMU register types. Note that in C64 mode the MMU completely disappears from the system's memory map. Note that the data out of the MMU is valid **only** on AEC high. This is necessary to avoid bus contention during a VIC cycle.




**MMU Register Map**




**Configuration Register**
**Preconfiguration Register**

## The Configuration Register

The **Configuration Register,** CR, controls the ROM, RAM, and I/O configuration of the C128 system. It is located at $D500 in I/O space and at $FF00 in system space. Some of the bits in this register are at times reflected by hardware lines $MS_0$ and $MS_1$ in C128 mode, depending upon how RAM and ROM have been set. These MS lines are used to inform the PLA about the type of memory in a particular address range. In C64 mode, $MS_0$ and $MS_1$ are always high, and the selection of RAM and ROM is done by the PLA using standard C64 banking methods. The MS lines are alternately referred to as ROMBANK lines. They will be referred to as MS lines in this section in the interest of simplicity.

In C128 mode, bit 0 controls whether an I/O space, $D000 — $DFFF, or a ROM/RAM access occurs. A low will select I/O, a high will enable some kind of ROM/RAM access, the nature of which is controlled by other bits in this register. The value of this bit is stored in a prelatch, until the fall of the clock, in order to prevent its changing in an unstable situation. Note that when not I/O space, the ROM/RAM access is controlled by the defined ROM Hi configuration bits, which are described later. This bit resets to 0. When the I/O bit is low, MMU registers $D500 to $D50B will assert themselves. When the bit is high, these registers disappear from the memory map. MMU registers $FF00 to $FF04 are always available in C128 mode. The hardware line I/OSE always reflects the polarity of this bit when in C128 mode. In C64 mode the I/OSE line, the hardware line driven by this bit, is completely ignored by the PLA, and the MMU is never asserted, even when C64 I/O is enabled. The C64 method of selecting I/O via HIROM and CHAREN takes over here. The I/O hardware line remains in its set state when in C128 mode, even though it has no effect in this mode.

Bit number 1 controls processor access to ROM low space, $4000 — $7FFF, in C128 mode. If the bit is high, the area will appear as RAM, and a RAM access, CAS enable, will be generated to the appropriate RAM bank, which is determined by other bits in this register. If low, system ROM will be located in the space. This bit affects the memory status lines $MS_0$ and $MS_1$ which are decoded by the PLA to generate ROM chip selects. Selecting ROM here will drive both memory status lines low when the processor address falls within the specified low space range. This bit resets low to include the C128 Basic Low ROM. Of course in C64 mode, this bit is ignored.

The next two bits, bits 2 and 3, determine for C128 mode the type of memory that will be located in the mid space, $8000 — $BFFF. If they are both low, system ROM will be located here. If bit 2 alone is high, internal function ROM is located here. External function ROM appears for bit 3 being alone high, and RAM appears, along with the proper CAS generation, for both bits set high. These bits also affect the hardware memory access lines. When in the aforementioned mid block address range, $MS_0$ will reflect the status of bit 3, and $MS_1$ will reflect the status of bit 2. These bits both reset low to start out with Basic Hi. C64 mode ignores these bits.

Bits 4 and 5 determine the contents of the Hi block, $C000 — $FFFF, for C128 mode, and have no effect on C64 mode. As with the mid space, both bits zero will set up system ROM, bit 4 high will set up internal function ROM, bit 5 high will set up external function ROM, and both bits high will set up RAM. Note that the I/O configuration bit, when set for I/O space, will leave the area from $D000 to $DFFF as I/O space, regardless of the values of these bits. If not set for I/O space, $D000 to $DFFF will contain the character ROM if the ROM chosen is System ROM. As with the other ROM selection bits, these bits are reflected by the memory status lines when this region of address is accessed. Bit 5 corresponds to $MS_0$ and bit 4 to $MS_1$. Both of these bits reset to low to permit Kernal and Character ROM to power up in this address space. Note that there is always a hole in high ROM during C128 mode for the MMU registers at $FF00 to $FF04. This hole is brought about by holding both MS lines high and both CAS enable lines high. These bits are ignored in C64 mode.

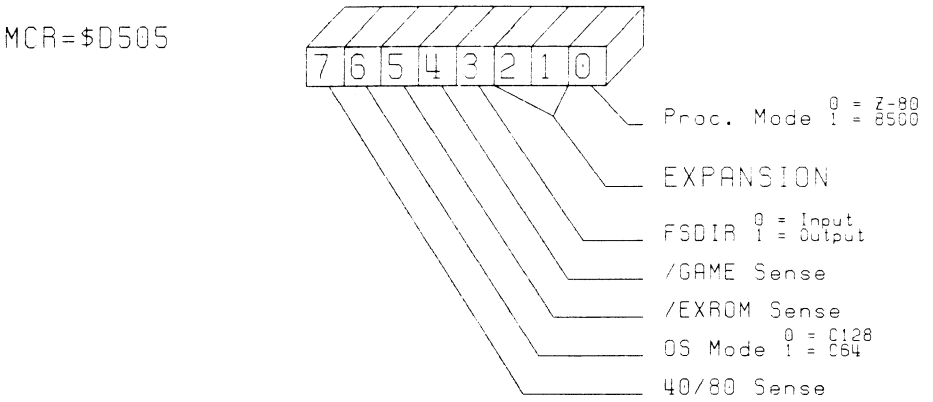
Finally, bit 6 controls the RAM bank selection. When low, it will select bank 0 by dropping $CAS_0$. When high, it will select bank 1 by dropping $CAS_1$. Bit 7 is unassigned at the present, left for future expansion. Note that a RAM share status that is non-zero will override the normal CAS enable generation to provide $CAS_0$ for all shared memory. Also, note that when the proper CAS enable is generated, any area of memory, even if that area does not have its ROM bank bits set for RAM, is accessed. It is up to the PLA to block CAS for a read from ROM. This allows RAM bleed through on a write to ROM. For any access to the MMU registers from $FF00 to $FF04, in any C128 mode configuration, both CAS enable lines and both MS lines will be high. Note that in C64 mode, the bank used follows the same rules as in C128 mode, though of course banks cannot be changed once in C64 mode.

### The Preconfiguration Mechanism

The Preconfiguration Mechanism is a feature of the MMU that allows the Configuration Register to be loaded with one of several memory configurations, with a minimum of time and memory on the part of the user. The scheme makes use of two sets of registers, the **Preconfiguration Registers** and the **Load Configuration Registers.**

The Preconfiguration Registers (PCRA — PCRD) are used to store several different memory configurations that may be accessed with a single store instruction. The format of each preconfiguration register is the same as for the Configuration Register but, when a value is stored to a preconfiguration register, no immediate effect takes place. They occupy I/O space from $D501 to $D504. These registers always reset to all zeros.

Load Configuration Registers (LCRA — LCRD) directly correspond with the preconfiguration registers on a one-to-one basis. A write to a Load Configuration Register causes the contents of the corresponding Preconfiguration Register to be transferred to the Configuration Register. A read of any Load Configuration Register returns the value of its corresponding Preconfiguration Register. Load Configuration Registers are located in system space from $FF01 to $FF04. Neither the Load Configuration Registers nor the Preconfiguration Registers have any effect in C64 mode. These registers reset to all zeros. Note that these, and the configuration register at $FF00, will **always** be available, completely independent of the ROM, RAM, or bank configuration defined for Hi ROM space. Any address in this range will cause the MMU to force both memory status lines and both CAS enable lines high.




**Mode Configuration Register**

## The Mode Configuration Register

The control of the current system mode is governed by the **Mode Configuration Register,** MCR. It controls which processor, 8502 or Z-80, and which operating system mode, C64 or C128, is currently in operation, and handles other overhead of the different operating modes. This register is located in the I/O space at $D505.

Several of the bits in this register function as bidirectional ports, including the FSDIR, GAME, EXROM, and 40/80 bits. This type of port functions like an output port. If a value is written to the port, its hardware line will reflect that value written, and a read will return that value. The only exception to this is if an external source is pulling down the corresponding port line. When pulled down, a read of the port will return a low. Once the external source has been removed, a read will return the value previously stored. Thus, as an input, the port can be driven low, but not high, by an external source. Under each bit description, both the input and output functions of each port bit will be described in detail.

The first bit, bit 0, controls which processor is enabled. It is reflected by the output line **Z80EN.** When low, it indicates that the processor is the Z-80. This is the reset configuration, and will cause the Z-80 processor to be active and all accesses to memory to follow the Z-80 mapping rules. In Z-80 mode, any address to RAM bank 0 in the range from $0000 to $0FFF will be translated to the corresponding address in the range from $D000 to $DFFF, where the Z-80 CP/M BIOS physically exists in System ROM. Additionally, the memory status lines $MS_0$ and $MS_1$, will reflect system ROM (both low) for accesses in the range of the BIOS, and the page zero and page one offset pointers will be disabled. RAM can still be banked by the CR $A_{16}$ bit, which controls $CAS_0$ and $CAS_1$. When in bank one, the BIOS ROM disappears, allowing the RAM from $0000 to $FFFF to be used by the system, and enabling the page zero and one offset pointers.

A change to this processor select bit is held in prelatch until a clock transition, in order to prevent processor changing in the midst of an instruction execution. Bringing this bit high will cause the Z-80 to be disabled and the 8502 to take over. Upon system power up, the Z-80 will turn itself off and bring up C128 mode by setting this bit and allowing the 8502 to take over.

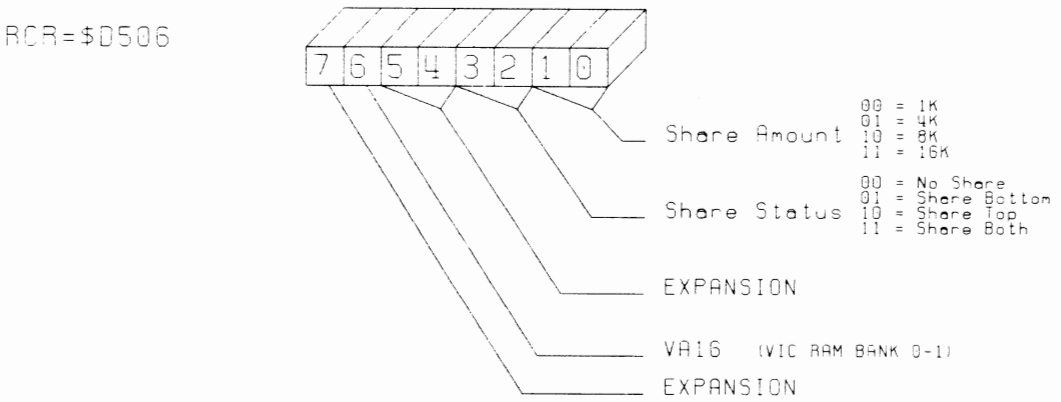
Bits 1 and 2 are unused, but are reserved for future expansion as possible port lines. Currently, they will return high if read, and cannot be written to.

Bit 3 is the **FSDIR** control bit. It is used as an output to control the fast serial disk data direction buffer hardware, and as an input to sense a fast disk enable signal. This bit is a bidirectional port bit as explained above, and its hardware line is called **FSDIR.**

Bits 4 and 5 are the $\overline{\text{GAME}}$ and $\overline{\text{EXROM}}$ sense bits, respectively, which are implemented as bidirectional ports as explained above. As inputs, they directly reflect the hardware cartridge control lines $\overline{\text{GAME}}$ and $\overline{\text{EXROM}}$ as used in C64 mode. C128 cartridges do not use $\overline{\text{EXROM}}$ and $\overline{\text{GAME}}$, so if they are detected in C128 mode, a C64 cartridge is present and C64 mode should be asserted. They have no dedicated C128 function.

The operating system mode is set by bit 6. This bit is cleared to zero upon reset and its presence enables all MMU registers and other C128 features, as well as asserting the C128 control line in hardware. Setting this bit removes the MMU from the memory map and sets the system up in C64 mode. Note that the C128 $MS_3$ hardware line reflects a logical inversion of the level of this bit.

Bit 7 is used to detect the status of the screen mode switch, as presented in hardware to the Sense40 column pin. If this bit is high, the 40/80 column switch is open, if low, the switch is closed. The display mode will be set according to a software interpretation of this bit. This bit is a bidirectional port bit, but its output function is undedicated at this time.




**RAM Configuration Register**

## The RAM Configuration Register

The **RAM Configuration Register** sets up the RAM segmenting parameters for both the processor and the block pointer for the VIC chip. This register is located in the I/O space at $D506.

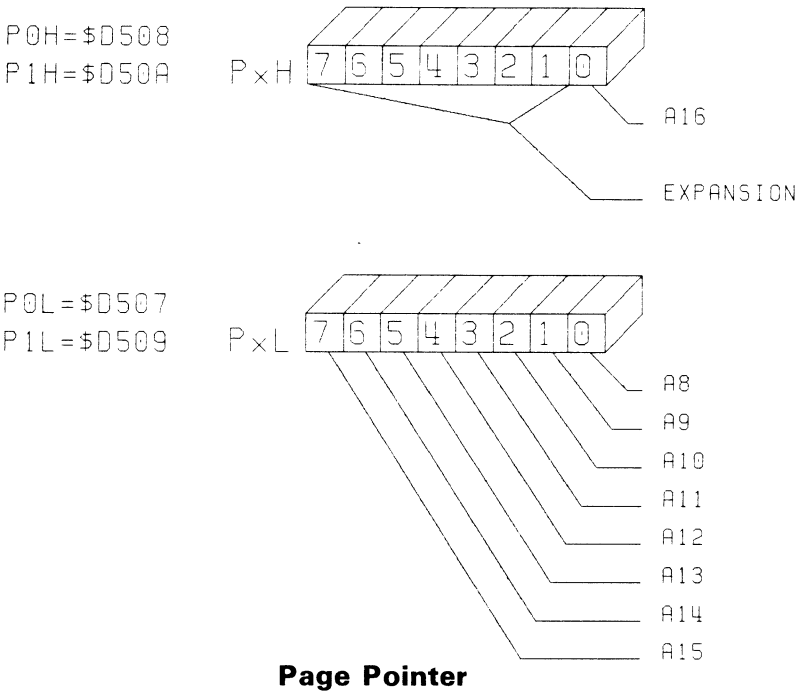
Bits 0 and 1 function together to determine the size of the RAM to be shared between banks, assuming that sharing is enabled. With common RAM, the RAM bank bits of the configuration register are basically overridden, as the selected bank of RAM will be used for the non-common areas, while bank 0 will be used for the specified common areas. ROM and I/O block configuration bits, however, are still important. If the value of the bits together is 0, then 1K of RAM is held common. If the value is 1, then 4K; 2, then 8K; 3, then 16K. These bits have no effect in C64 mode, and the reset value of both bits is defined to be zero.

Bits 2 and 3 function to determine how and if RAM is kept common. If both are low, no sharing takes place. If bit 2 is set, the bottom RAM is shared. If bit 3 is set, the top RAM is shared. Both may be set at the same time for sharing both top and bottom memory. The reset configuration sets both of these bits zero, such that no common memory is present.

The next two bits, numbers 4 and 5, are not used in this MMU. They are available for possible future expansion. They read low, and cannot be written to.

Bit 6 functions as a RAM bank pointer for VIC. It is used to drive $CAS_0$ low when set low or $CAS_1$ low when set high, thus selecting either RAM bank 0 or RAM bank 1 for the VIC, independently from the processor bank. When in 2 MHz mode the 80-column chip takes over, causing the VIC to be disabled. This disabling is affected by the VIC chip itself holding AEC constantly high, and thus is not directly effected by actions of the MMU. Note that since a VIC cycle is detected by AEC low, that any DMA will put the MMU into VIC configuration, as it too brings AEC low. This allows independent bank selection for DMAs in 80 column mode.

Bit 7 is currently unused.




**Page Pointer**

## The Page Pointers

The page pointers are four registers that allow independent relocation of pages zero and one, when running under either processor. These are especially useful when running under the 8502 as they help to remove some of the zero page and stack size limitations normally associated with 6502 family processors.

For zero page relocation, the MMU provides the **Page Zero Pointer High** ($P0_H$) and **Page Zero Pointer Low** ($P0_L$) registers. Bit 0 of the $P0_H$ register corresponds to translated addresses $TA_{16}$ for any zero page access, $0000 — $00FF, controlling the generation of $CAS_0$ or $CAS_1$ depending on whether it is low or high. The remaining bits are currently unused, and will always return zero. These bits override the RAM bank bits, the ROM block, and the I/O block bits to determine which physical page appears as zero page for all zero page accesses. A write to the $P0_H$ register is stored in prelatch until a write to the $P0_L$ register occurs. Bits 0 to 7 of the $P0_L$ correspond to Translated Addresses $TA_8$ to $TA_{15}$ for any zero page access, thus relocating the zero page. Any access to the area that has become the relocated zero page will be switched back to the original zero page if that area is mapped as RAM. If mapped as ROM, then the reverse mapping is not done, allowing access to the ROM. A write to this register sets up the zero page transfer, which can occur as soon as the next low clock cycle. Register $P0_L$ is located in the I/O space at $D507, while register $P0_H$ is located at $D508.

# COMMON LINE DEFINITIONS

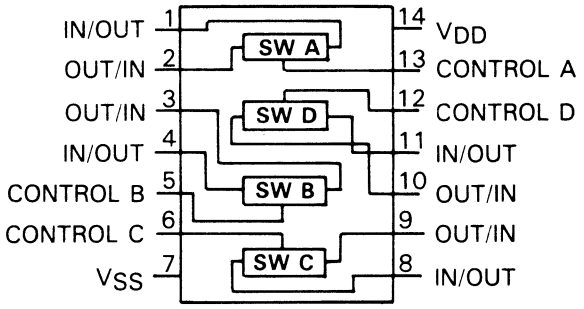
| | | | |
|---|---|---|---|
| A0-A7 | PROCESSOR ADDRESS BUS | LCR | LOAD CONFIGURATION REGISTER |
| AEC | ADDRESS ENABLE CONTROL | LP | LIGHT PEN INPUT |
| ATN | ATTENTION LINE | | |
| | | MA0-MA11 | MULTIPLEXED ADDRESS BUS |
| BA | BUS AVAILABLE | MMU | MEMORY MANAGEMENT UNIT |
| | | MS 0-4 | MEMORY STATUS, ALSO INDENTIFIED AS ROMBANK |
| C128/64 | C128 OR C64 MODE | | |
| CAP LK | CAPITAL LOCK | MUX | ADDRESS MULTIPLEX CONTROL |
| CAS | DRAM COLUMN ADDRESS STROBE | | MEMORY MULTIPLEX |
| CASENB | RAM COLUMN ADDRESS STROBE ENABLE | | |
| CASS SENSE | CASSETTE SENSE | NMI | NON-MASKABLE INTERRUPT |
| CASS WRT | CASSETTE WRITE | | |
| CASS MTR | CASSETTE MOTOR | PHI 0 | 2 MHZ 0 CLOCK |
| CHAROM | CHARACTER ROM SELECT | POT X,Y | JOYSTICK PORT INPUTS |
| CIA | COMPLEX INTERFACE ADAPTOR | | |
| CLR BNK | COLOR RAM BANK SELECT | RCR | RAM CONFIGURATION REGISTER |
| CNT | COUNT INPUT | RESET | SYSTEM RESET |
| COLORAM | COLOR RAM CHIP SELECT | ROM 1-4 | ROM CHIP SELECTS FOR OPERATING SYSTEM |
| D0-D7 | DATA BUS | ROM H,L | CHIP SELECTS FOR EXPANSION ROMS |
| DA0-DA7 | DISPLAY ADDRESS | ROMBANK | |
| DD0-DD7 | DISPLAY DATA BUS | 0,1 | MEMORY STATUS SELECT |
| DMA | DIRECT MEMORY ACCESS | RS | REGISTER SELECT |
| DOT CLK | 8.18 MHZ VIDEO DOT CLOCK | RSTR | RESTORE |
| DRAM | DYNAMIC RAM | R/W | READ/WRITE LINE |
| DRESET | DYNAMIC RAM RESET | | |
| DWE | DRAM WRITE ENABLE | SA0-SA7 | SHARED ADDRESS BUS |
| EXROM | EXTERNAL ROM ENABLE | TA8-TA15 | TRANSLATED ADDRESS BUS |
| EXTRES | EXTERNAL RESET | TOD | TIME OF DAY |
| | | VA 14,15 | VIC ADDRESSES |
| FROM | FUNCTION ROM | VIC | VERSATILE INTERFACE CHIP |
| FSDIR | FAST SERIAL DIRECTION | VMA0-VMA7 | VIC MULTIPLEXED ADDRESS BUS |
| GAME | GAME ROM ENABLE | Z80EN | Z-80 ENABLE |
| GWE | COLOR RAM WRITE ENABLE | Z80 PHI | Z-80 CLOCK |
| | | ZD0-ZD7 | Z-80 DATA BUS |
| I/O | I/O SELECT | | |
| IOACC | I/O ACCESS | 1 MHZ | MASTER CLOCK $\phi$ IN |
| IRQ | INTERRUPT REQUEST | 40/80 SENSE | 40/80 COLUMN STATUS SENSE |

# COMMON I.C.'S
# PIN ASSIGNMENTS AND LOGIC

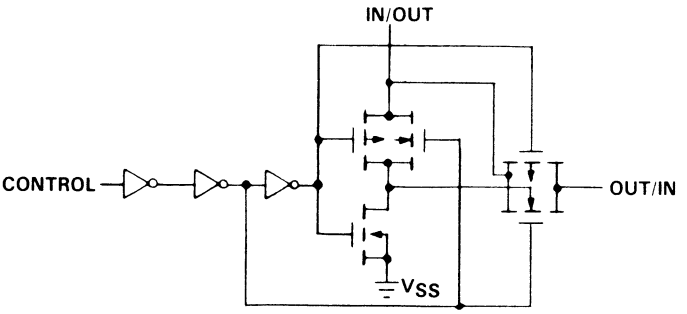
## 4066
## QUAD BILATERAL SWITCH

### PIN ASSIGNMENTS

DUAL-IN-LINE PACKAGE

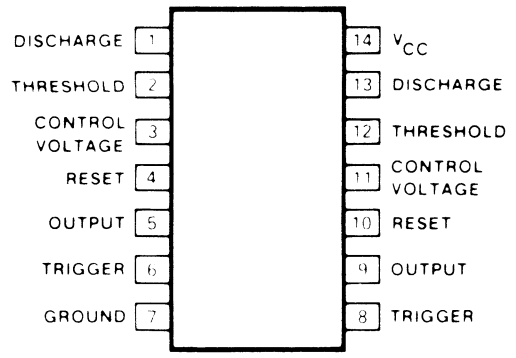


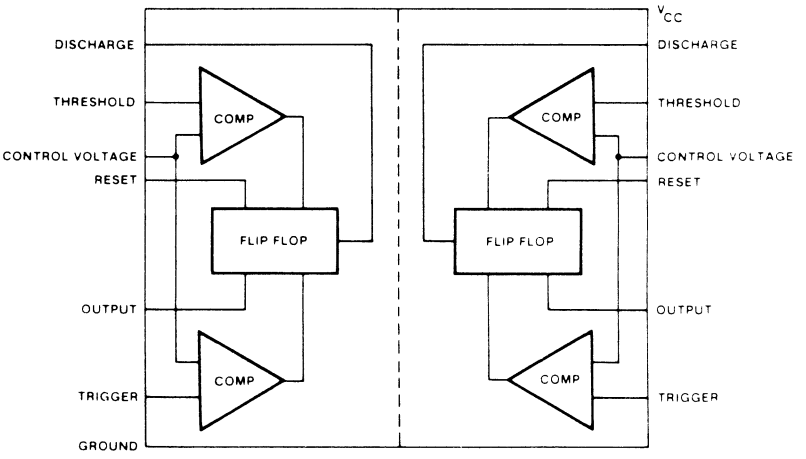

### INTERNAL DIAGRAM (EACH SWITCH)




## 556
## DUAL TIMER

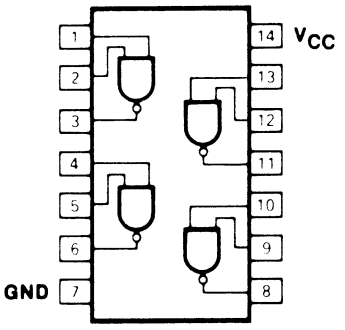
### PIN ASSIGNMENTS



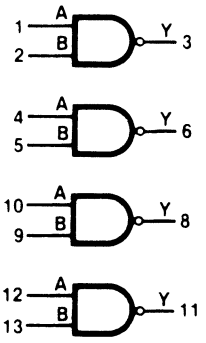



## 7400 • 74S00 • 74LS00
## QUAD 2-INPUT NAND GATE

### PIN ASSIGNMENT




### LOGIC DIAGRAM



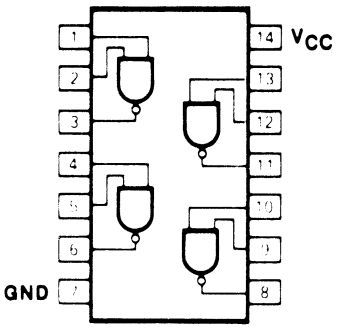

### TRUTH TABLE

| INPUTS | | OUTPUT |
|---|---|---|
| A | B | Y |
| L | L | H |
| L | H | H |
| H | L | H |
| H | H | L |

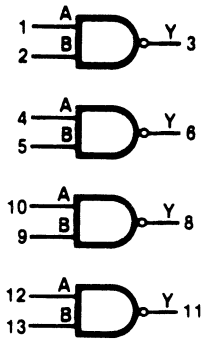
H = HIGH voltage level
L = LOW voltage level

## 74LS03
## QUAD 2-INPUT NAND GATE (OPEN COLLECTOR)

### PIN ASSIGNMENT




### LOGIC DIAGRAM



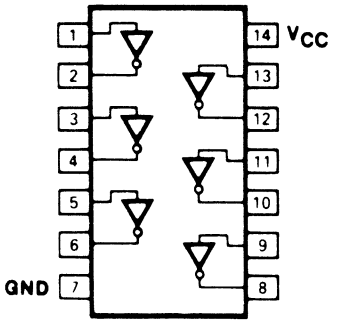

### TRUTH TABLE

| INPUTS | | OUTPUT |
|---|---|---|
| A | B | Y |
| L | L | H |
| L | H | H |
| H | L | H |
| H | H | L |

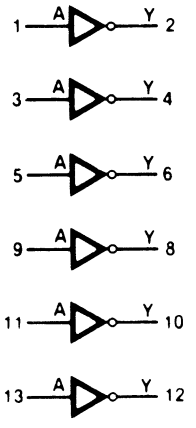
H = HIGH voltage level
L = LOW voltage level

## 7406
## HEX INVERTER BUFFER/DRIVER (OPEN COLLECTOR)

### PIN ASSIGNMENT




### LOGIC DIAGRAM



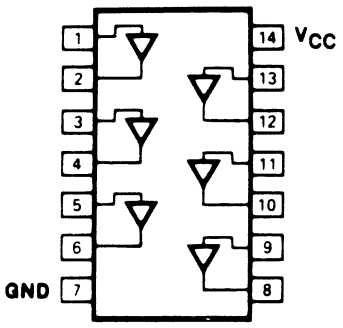

### TRUTH TABLE

| INPUT | OUTPUT |
|---|---|
| A | Y |
| H | L |
| L | H |

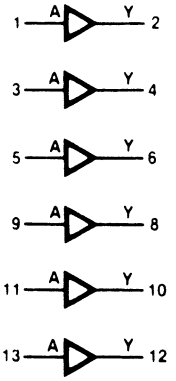
H = HIGH voltage level
L = LOW voltage level

## 7407
## HEX BUFFER/DRIVER (OPEN COLLECTOR)

### PIN ASSIGNMENT




### LOGIC DIAGRAM



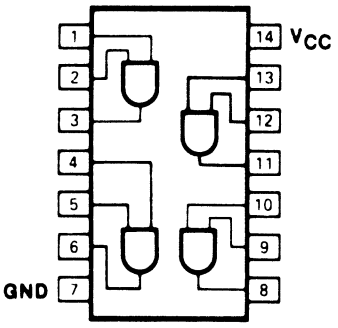

### TRUTH TABLE

| INPUT | OUTPUT |
|---|---|
| A | Y |
| H<br>L | H<br>L |

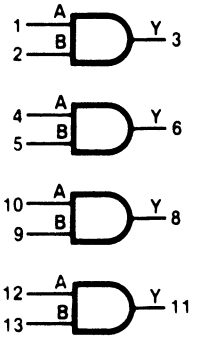
H = HIGH voltage level
L = LOW voltage level

## 7408 • 74S08 • 74LS08
## QUAD 2-INPUT AND GATE

### PIN ASSIGNMENT




### LOGIC DIAGRAM



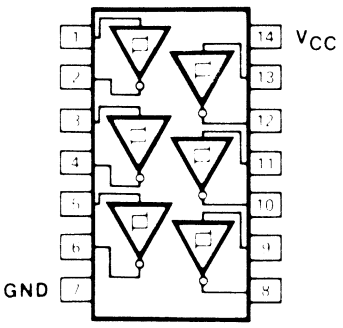

### TRUTH TABLE

| INPUTS | | OUTPUT |
|---|---|---|
| A | B | Y |
| L<br>L<br>H<br>H | L<br>H<br>L<br>H | L<br>L<br>L<br>H |

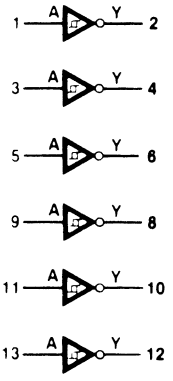
H = HIGH voltage level
L = LOW voltage level

## 7414 • 74LS14
## HEX INVERTER SCHMITT TRIGGER

### PIN ASSIGNMENT




### LOGIC DIAGRAM



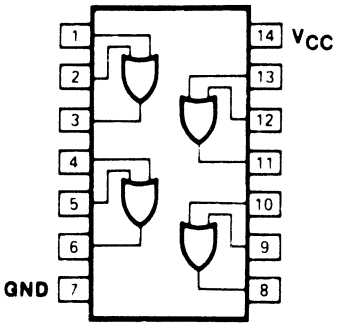

### TRUTH TABLE

| INPUT | OUTPUT |
|---|---|
| A | Y |
| 0<br>1 | 1<br>0 |

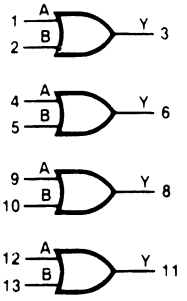
H = HIGH voltage level
L = LOW voltage level

## 7432 • 74S32 • 74LS32 • 74F32
## QUAD 2-INPUT OR GATE

### PIN ASSIGNMENT




### LOGIC DIAGRAM



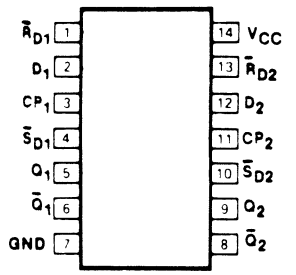

### TRUTH TABLE

| INPUTS | | OUTPUT |
|---|---|---|
| A | B | Y |
| L | L | L |
| L | H | H |
| H | L | H |
| H | H | H |

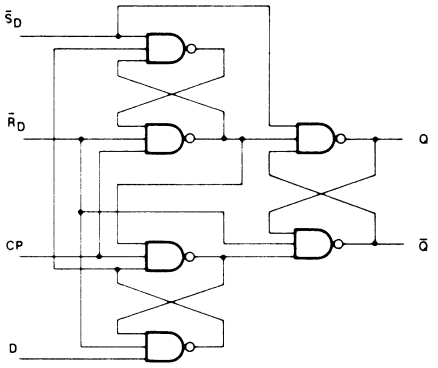
H = HIGH voltage level
L = LOW voltage level

## 7474 • 74S74 • 74LS74
## DUAL D-TYPE FLIP FLOP (POSITIVE EDGE TRIGGERED)

### PIN ASSIGNMENT




### LOGIC DIAGRAM




### TRUTH TABLE

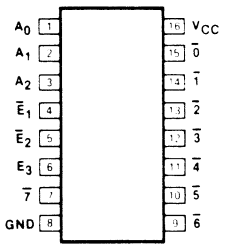
| OPERATING MODE | INPUTS | | | | OUTPUTS | |
|---|---|---|---|---|---|---|
| | $\bar{S}_D$ | $\bar{R}_D$ | CP | D | Q | $\bar{Q}$ |
| Asynchronous Set | L | H | X | X | H | L |
| Asynchronous Reset (Clear) | H | L | X | X | L | H |
| Undetermined[a] | L | L | X | X | H | H |
| Load "1" (Set) | H | H | ↑ | h | H | L |
| Load "0" (Reset) | H | H | ↑ | l | L | H |

H = HIGH voltage level steady state
h = HIGH voltage level one setup time prior to the LOW-to-HIGH clock transition
L = LOW voltage level steady state
l = LOW voltage level one setup time prior to the LOW-to-HIGH clock transition
X = Don't care
↑ = LOW-to-HIGH clock transition

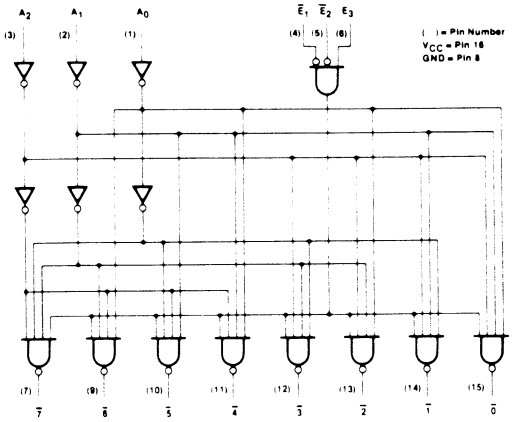
NOTE
(a) Both outputs will be HIGH while both $\bar{S}_D$ and $\bar{R}_D$ are LOW, but the output states are unpredictable if $\bar{S}_D$ and $\bar{R}_D$ go HIGH simultaneously.

## 74S138 • 74LS138
## DECODER/DEMULTIPLEXER

### PIN ASSIGNMENT




### LOGIC DIAGRAM



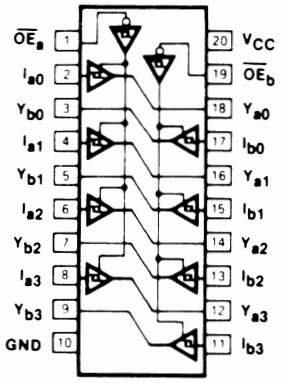

### TRUTH TABLE

| INPUTS | | | | | | OUTPUTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\bar{E}_1$ | $\bar{E}_2$ | $E_3$ | $A_0$ | $A_1$ | $A_2$ | $\bar{0}$ | $\bar{1}$ | $\bar{2}$ | $\bar{3}$ | $\bar{4}$ | $\bar{5}$ | $\bar{6}$ | $\bar{7}$ |
| H | X | X | X | X | X | H | H | H | H | H | H | H | H |
| X | H | X | X | X | X | H | H | H | H | H | H | H | H |
| X | X | L | X | X | X | H | H | H | H | H | H | H | H |
| L | L | H | L | L | L | L | H | H | H | H | H | H | H |
| L | L | H | H | L | L | H | L | H | H | H | H | H | H |
| L | L | H | L | H | L | H | H | L | H | H | H | H | H |
| L | L | H | H | H | L | H | H | H | L | H | H | H | H |
| L | L | H | L | L | H | H | H | H | H | L | H | H | H |
| L | L | H | H | L | H | H | H | H | H | H | L | H | H |
| L | L | H | L | H | H | H | H | H | H | H | H | L | H |
| L | L | H | H | H | H | H | H | H | H | H | H | H | L |

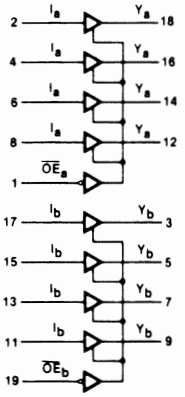
NOTES
H = HIGH voltage level
L = LOW voltage level
X = Don't care

## 74S244 • 74LS244
## OCTAL 3-STATE BUFFER

### PIN ASSIGNMENT




### LOGIC DIAGRAM



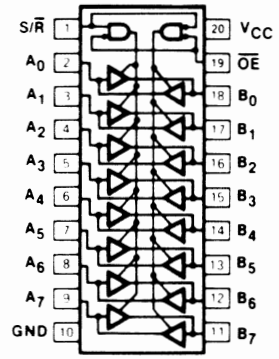

### TRUTH TABLE

| INPUTS | | | | OUTPUT | |
|---|---|---|---|---|---|
| $\overline{OE}_a$ | $I_a$ | $\overline{OE}_b$ | $I_b$ | $Y_a$ | $Y_b$ |
| L | L | L | L | L | L |
| L | H | L | H | H | H |
| H | X | H | X | (Z) | (Z) |

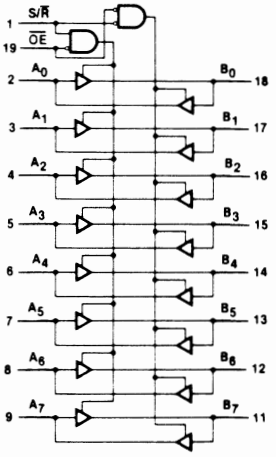
H = HIGH voltage level
L = LOW voltage level
X = Don't care
Z = HIGH impedance "off" state

## 74LS245 • 74F245
## OCTAL BUS TRANSCEIVER

### PIN ASSIGNMENT




### LOGIC DIAGRAM



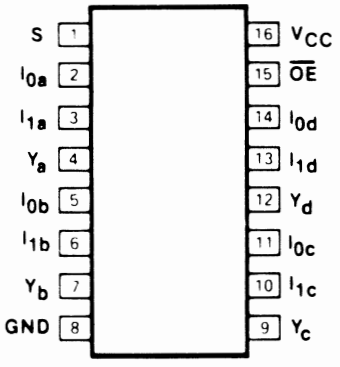

### TRUTH TABLE

| INPUTS | | INPUTS/OUTPUTS | |
|---|---|---|---|
| $\overline{OE}$ | S/$\overline{R}$ | $A_n$ | $B_n$ |
| I | I | A = B | INPUTS |
| L | H | INPUT | B = A |
| H | X | (Z) | (Z) |

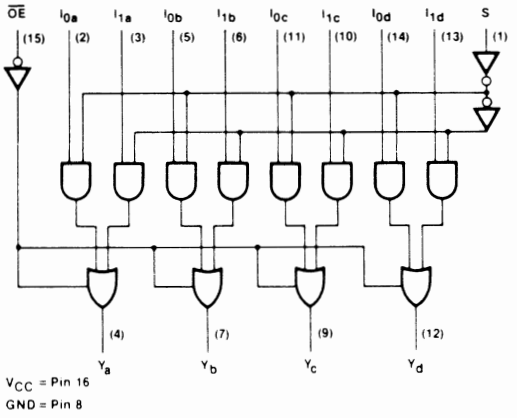
H = HIGH voltage level
L = LOW voltage level
X = Don't care
(Z) = HIGH impedance "off" state

## 72S257 • 74LS257

### PIN ASSIGNMENT




### LOGIC DIAGRAM




### TRUTH TABLE

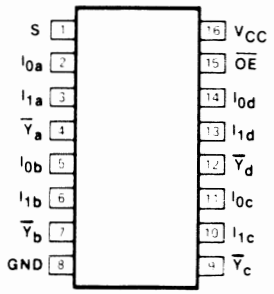
| ENABLE | SELECT INPUT |
|---|---|
| $\overline{OE}$ | S |
| H | X |
| L | H |
| L | H |
| L | L |
| L | L |

| INPUTS | | OUTPUT |
|---|---|---|
| $I_0$ | $I_1$ | Y |
| X | X | (Z) |
| X | L | L |
| X | H | H |
| L | X | L |
| H | X | H |

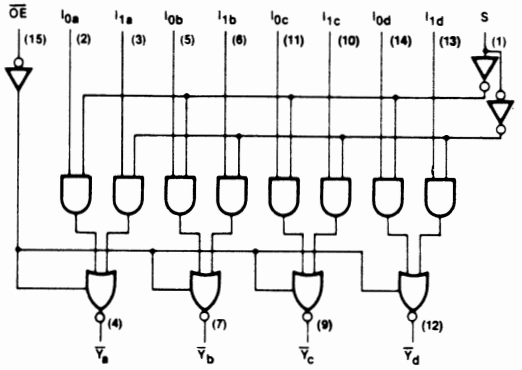
H = HIGH voltage level
L = LOW voltage level
X = Don't care
(Z) = HIGH impedance (off) state

## 72S258 • 74LS258A
## QUAD 2:1 MULTIPLEXER (3-STATE)

### PIN ASSIGNMENT




### LOGIC DIAGRAM



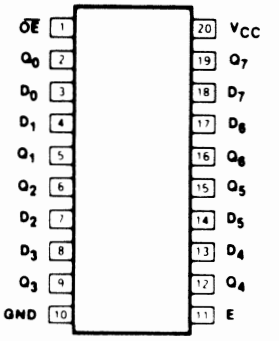

### TRUTH TABLE

| OUTPUT ENABLE | SELECT INPUT | DATA INPUTS | | OUTPUTS |
|---|---|---|---|---|
| $\overline{OE}$ | S | $I_0$ | $I_1$ | $\overline{Y}$ |
| H | X | X | X | (Z) |
| L | H | X | L | H |
| L | H | X | H | L |
| L | L | L | X | H |
| L | L | H | X | L |

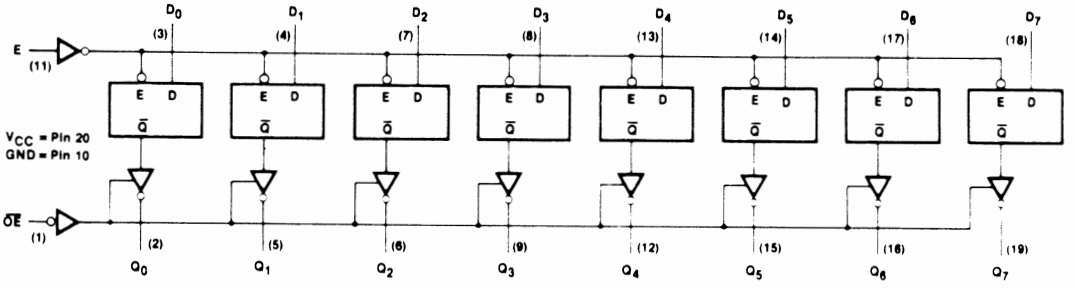
H = HIGH voltage level
L = LOW voltage level
X = Don't care
(Z) = HIGH impedance (off) state

## 74S373 • 74LS373
## OCTAL LATCH (3-STATE)

### PIN ASSIGNMENT




### LOGIC DIAGRAM



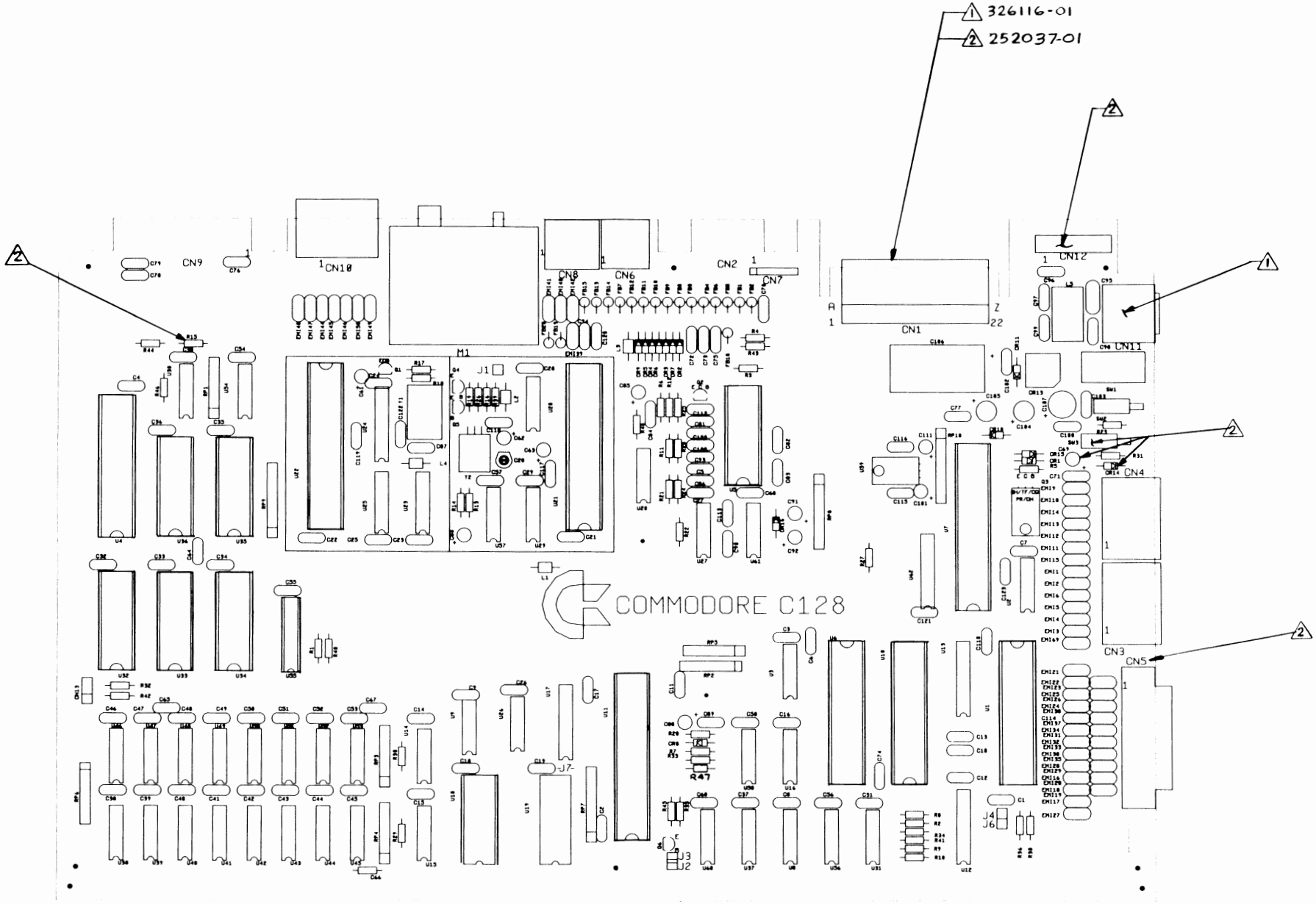

### TRUTH TABLE

| OPERATING MODES | INPUTS | | | INTERNAL REGISTER | OUTPUTS |
|---|---|---|---|---|---|
| | $\overline{OE}$ | E | $D_n$ | | $Q_0$-$Q_7$ |
| Enable and read register | L | H | L | L | L |
| | L | H | H | H | H |
| Latch and read register | L | L | l | L | L |
| | L | L | h | H | H |
| Latch register and disable outputs | H | L | l | L | (Z) |
| | H | L | h | H | (Z) |

H = HIGH voltage level
h = HIGH voltage level one setup time prior to the LOW-to-HIGH clock transition or HIGH-to-LOW $\overline{OE}$ transition
L = LOW voltage level
l = LOW voltage level one setup time prior to the LOW-to-HIGH clock transition or HIGH-to-LOW $\overline{OE}$ transition
(Z) = HIGH impedance "off" state
↑ = LOW-to-HIGH clock transition

# INITIAL SERVICE POLICY

DURING THE FIRST 90 DAYS, THE C-128 WILL BE REPAIRED AT THE ASSEMBLY LEVEL UNDER WARRANTY. THROUGHOUT THIS TIME PERIOD, TROUBLESHOOTING CHARTS AND AIDS WILL BE DEVELOPED FROM THE INFORMATION GATHERED. ANY COMMON FAILURES WILL RECEIVE SPECIAL ATTENTION. THIS INFORMATION WILL BE AVAILABLE APPROXIMATELY NOVEMBER 1, 1985 TO FACILITATE THE TRANSITION INTO COMPONENT LEVEL REPAIR. ALL CQS CENTERS WILL AUTOMATICALLY RECEIVE AN UPDATE AT THAT TIME.




NOTE:
1 LOW END ONLY
2 HIGH END ONLY

**BOARD LAYOUT**
**PCB ASSY #310379 Revision 6**
**IDENTIFYING FACTOR: On solder side of board at the EXPANSION BUS, CN1, the artwork #310381 REV. 6 appears.**

# PARTS LIST
# PCB ASSEMBLY #310379
# REV. 6

| INTEGRATED CIRCUITS | | |
|---|---|---|
| U1 | 6526 CIA | 906108-01 |
| U2 | 4066 | |
| U3 | 74LS138 | |
| U4 | 6526 CIA | 906108-01 |
| U5 | 6581 SID | 906112-01 |
| U6 | 8502 Microprocessor | 315020-01 |
| U7 | 8722-R1 MMU | 310389-01 |
| U8 | 74LS08 | |
| U9 | 74LS32 | |
| U10 | Z80B Microprocessor 6 MHz | 906150-02 |
| U11 | 8721-R3 PLA | 315012-01 |
| U12 | 74LS373 | |
| U13 | 74LS244 | |
| U14,15 | 74LS257A | |
| U16 | 74LS14 | |
| U17 | 74ALS373 | |
| U18 | ROM 64K 128 Char | 390059-01 |
| U19 | 2016 16K RAM – 200ns | |
| U20 | 4066 | |
| U21 | 8564-R4 VIC | 315009-01 |
| U22 | 8563-R7 CRT Cntrl | 315014-01 |
| U23 | 4416 Dynamic RAM – 150ns | |
| U24 | 74LS244 | |
| U25 | 4416 Dynamic RAM – 150ns | |
| U26 | 74LS257A | |
| U27 | 556 | |
| U28 | 8701 Clock Generator | 251527-01 |
| U29,30 | 7406 | |
| U31 | 74LS00 | |
| U32 | ROM 1 – C64 Kern & Basic | 251913-01 |
| U33 | ROM 2 – Basic $4000 | 318018-02 |
| U34 | ROM 3 – Basic $8000 | 318019-02 |
| U35 | ROM 4 – Kernal $C000 | 318020-03 |
| U36 | ROM Mem Function | Blank |
| U37 | 7406 | |
| U38-53 | 4164 Dynamic RAM – 200ns | |
| U54 | 74LS32 | |
| U55 | 74LS245 | |
| U56 | 74LS74 | |
| U57 | 7407 | |
| U58 | 74LS03 | |
| U59 | 7812 Regulator 12V, To-220 Case | |
| U60 | 7407 | |
| U61 | 74LS08 | |
| U62 | 74LS244 | |

| TRANSISTORS | |
|---|---|
| Q1,2 | 2SC1815 |
| Q3 | 2SD880 |
| Q4,5,6 | 2SC1815 |
| Q100 | 2N4403 |
| Q101 | 2SC1815 |
| Q102 | 2N4403 |

| DIODES | | |
|---|---|---|
| CR1 | RD6.8EB | Zener 6.8V 400MW |
| CR2-7 | 1N914 | Sub: IN4148 |
| CR8 | 1N4371 | Zener 2.7V 500MW |
| CR9 | 1N914 | Sub: IN4148 |
| CR10,11 | 1N4001 | Rect 50V 1A |

| DIODES (Continued) | | |
|---|---|---|
| CR13 | Bridge Rect 100V 2A | 251026-01 |
| CR15,16 | 1N914 | |
| CR100,101 | 1N914 | |

**RESISTORS** – All values in ohms, 1/4W, 5% unless noted otherwise.

| | |
|---|---|
| R1 | 68 |
| R2 | 100 |
| R3 | 1K |
| R4 | 100 |
| R5 | 470 |
| R6 | 47K |
| R7 | 560 |
| R8 | 3.3K |
| R9 | 10K |
| R10 | 3.3K |
| R11,12 | 1K |
| R13,14 | 330 1W 5% |
| R16 | 120 |
| R17 | 47 |
| R18 | 82 |
| R19 | 330 1W 5% |
| R20 | 10K |
| R21 | 470K |
| R22 | 47K |
| R23 | 100K |
| R24 | 47K |
| R25 | 100K |
| R26 | 100 |
| R27 | 4.7K |
| R28 | 10K |
| R29,30 | 68 |
| R31 | 100 |
| R32 | 180 |
| R33,34 | 10K |
| R35 | 100 |
| R36 | 10K |
| R38 | 10K |
| R39 | 120 |
| R40 | 180 |
| R41 | 680 |
| R42 | 180 |
| R43,44 | 3.3K |
| R45 | 1.2K |
| R46 | 3.3K |
| R47 | 1K |
| R48 | 10K |
| R100,101 | 1K |
| R102 | 47K |
| R103,04 | 1K |
| R105 | 470 |
| R106 | 1K |

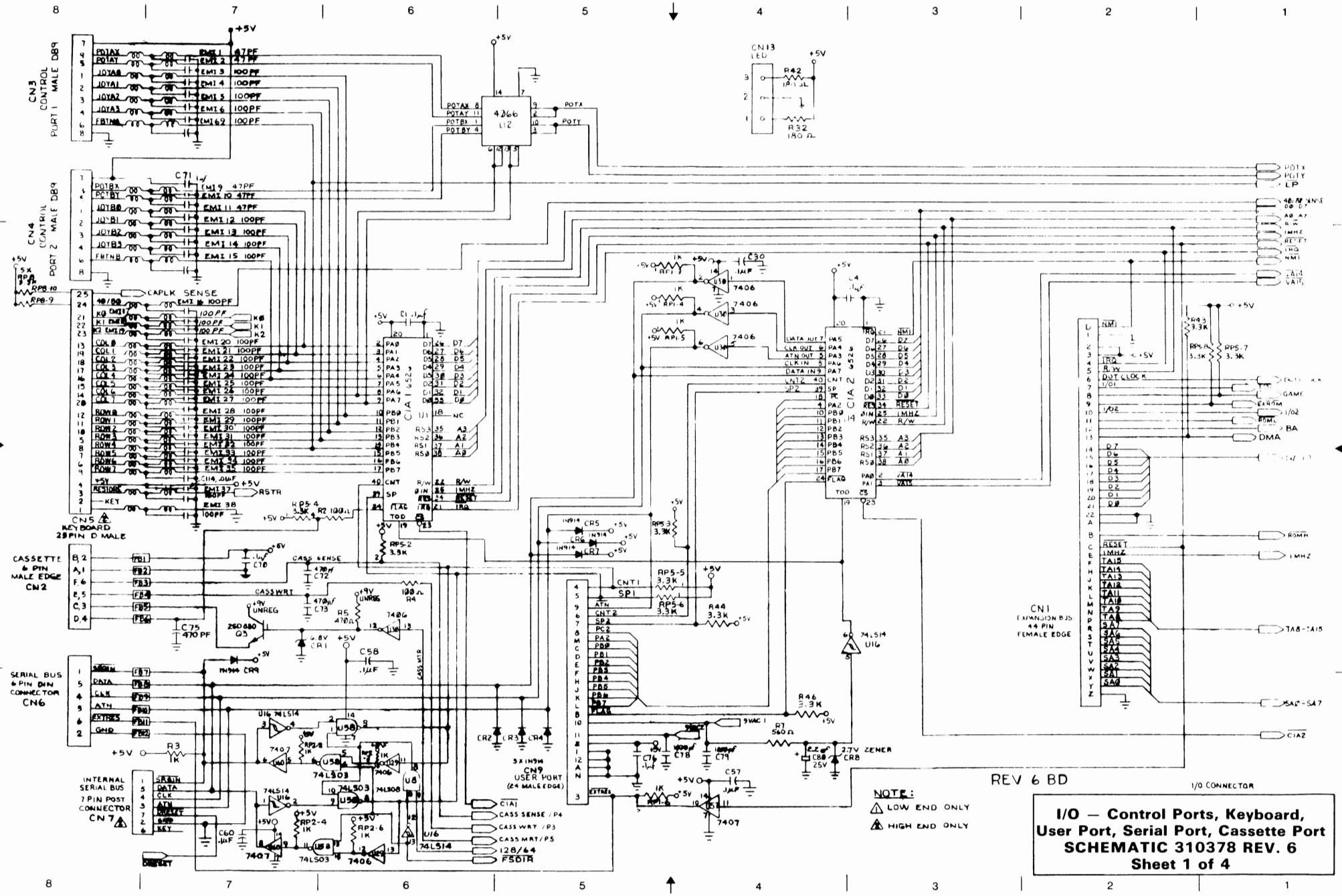
| RESISTOR PACKS | | |
|---|---|---|
| RP1 | 1K +/–2% | 8 Pin, SIP, Pin 1 Com |
| RP2 | 1K 1/8W +/–10% | 9 Pin, SIP, Pin 1 Com |
| RP3,4 | 33 +/–2% | 8 Pin, SIP, Isolated |
| RP5 | 3.3K +/–10% | 8 Pin, SIP, Pin 1 Com |
| RP6 | 1K 1/8W +/–10% | 9 Pin, SIP, Pin 1 Com |
| RP7,8 | 3.3K | 10 Pin, SIP, Pin 1 Com |
| RP9,10 | 10K | 10 Pin, SIP, Pin 1 Com |

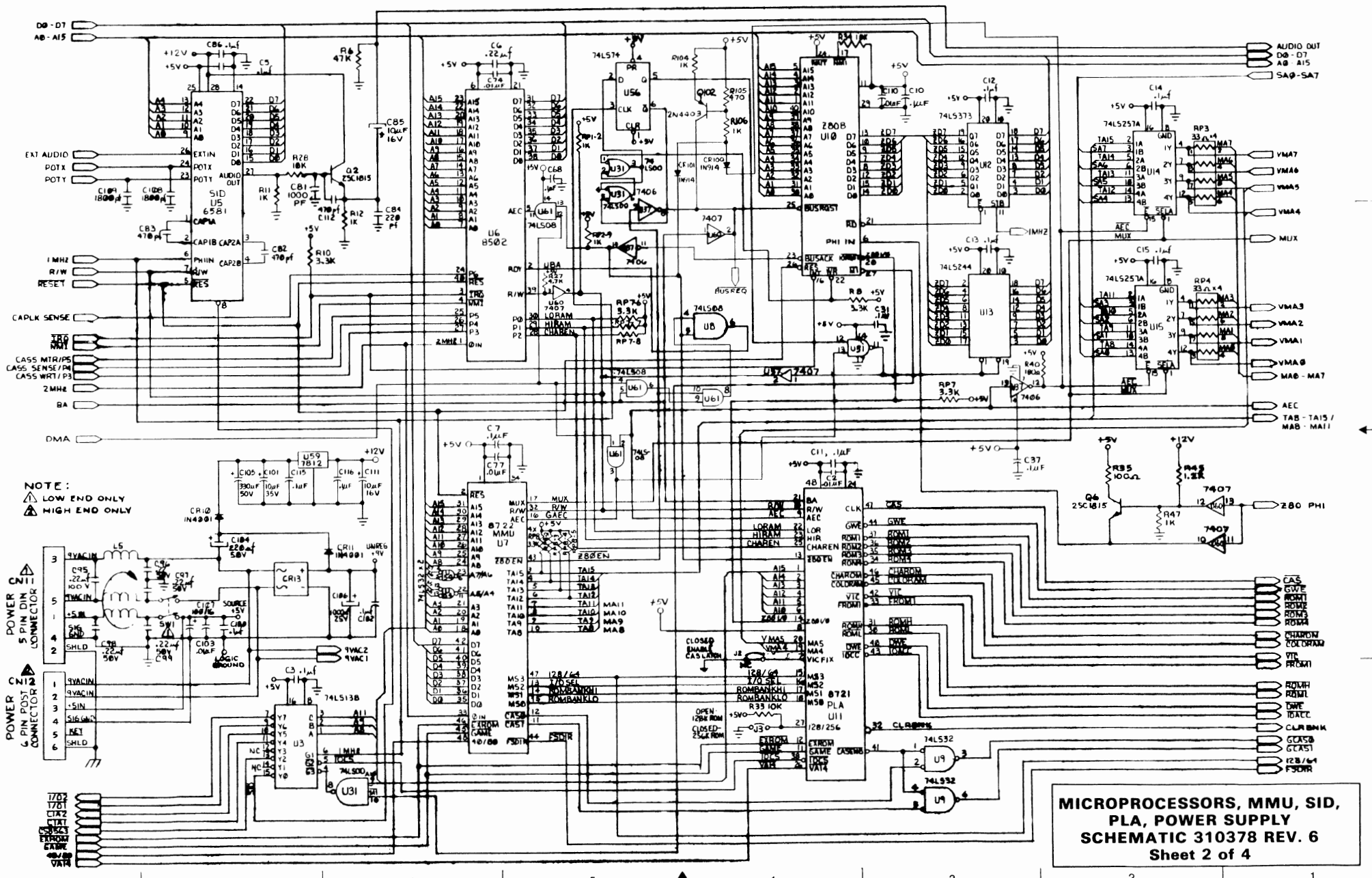
# PART LIST PCB ASSEMBLY 310379 REV. 6

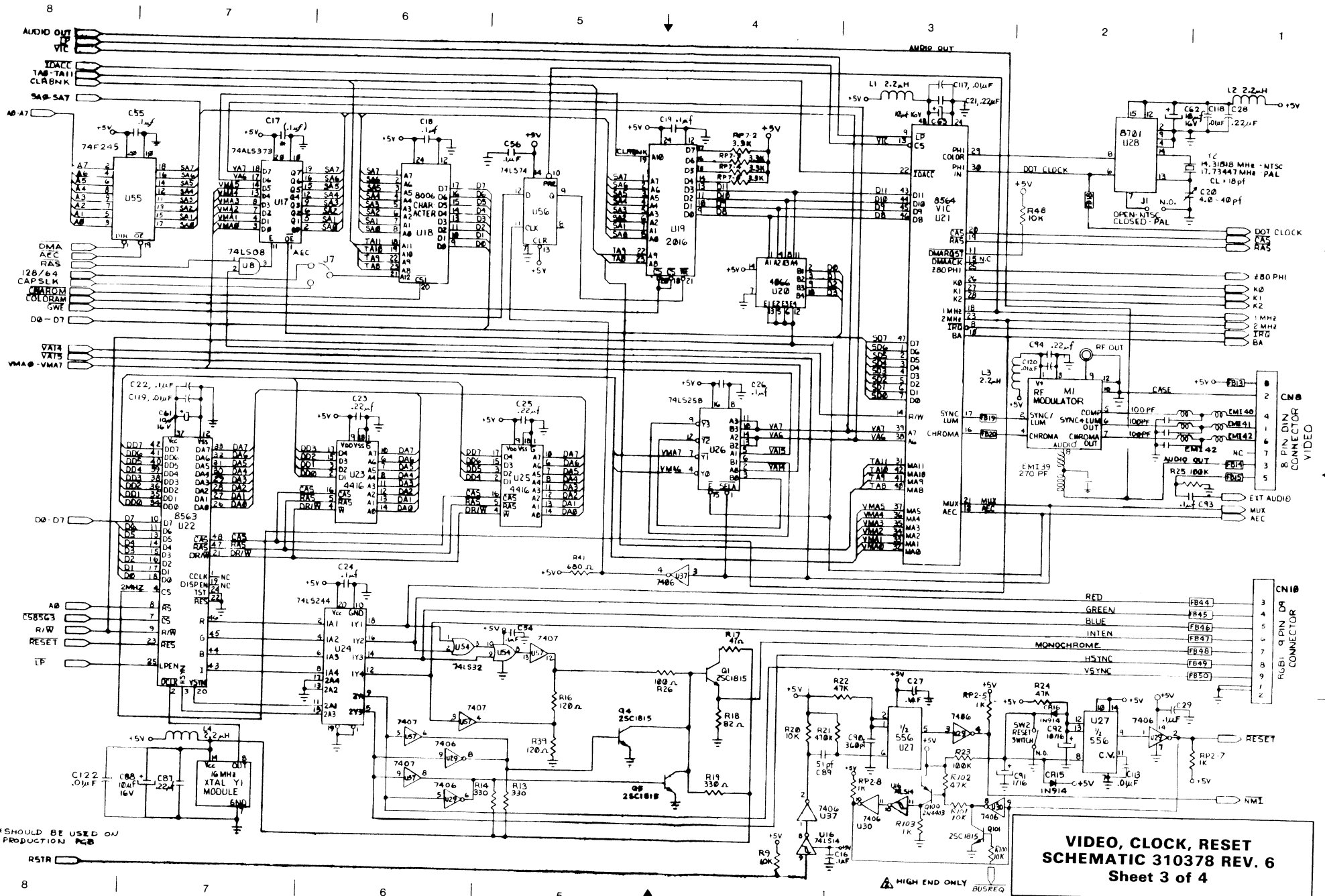
| **CAPACITORS** — All caps are 25v, +80%, -20% unless noted otherwise. | | |
|---|---|---|
| C1 | Cer | .1μF |
| C2 | Cer | 10000pF |
| C3-5 | Cer | .1F |
| C6 | Cer | .22μF |
| C7-19 | Cer | .1μF |
| C20 | Trim | 4-40pF |
| C21 | Cer | .22μF |
| C22 | Cer | .1μF |
| C23 | Cer | .22μF |
| C24 | Cer | .1μF |
| C25 | Cer | .22μF |
| C26,27 | Cer | .1μF |
| C28 | Cer | .22μF |
| C29-37 | Cer | .1μF |
| C38-53 | Cer | .22μF |
| C54-58 | Cer | .1μF |
| C60 | Cer | .1μF |
| C61-63 | Elect Alum | 10μF, 16V |
| C64,65 | Cer | 10000pF |
| C66 | Mylar | .01μF, 250V, +/-20% |
| C67 | Cer | 10000μF |
| C68 | Cer | .1μF |
| C69 | Elect Alum | 10μF, 16V |
| C70,71 | Cer | .1μF |
| C72,73 | Cer | 470pF, 50V, 10% |
| C74 | Cer | 10000pF |
| C75 | Cer | 470pF, 50V, 10% |
| C76 | Cer | .1μF |
| C77 | Cer | 10000pF |
| C78,C79 | Cer | 1000pF, 50V, 10% |
| C80 | Elect | 2.2μF, 25V |
| C81 | Cer | 1000pF, 50V, 10% |
| C82,83 | Cer | 470pF, 50V, 10% |
| C84 | Cer | 220pF, 50V, 10% |
| C85 | Elect Alum | 10μF, 16V |
| C86 | Cer | .1μF |
| C87 | Cer | .22μF |
| C88 | Elect Alum | 10μF, 16V |
| C89 | Cer | 51pF, 50V, +/-5% |
| C90 | Cer | 360pF, 50V, +/-5% |
| C91 | Elect | 1μF, 16V |
| C92 | Elect Alum | 10μF, 16V |
| C93 | Cer | .1μF |
| C94 | Cer | .22μF |
| C95 | Cer Mono | .22μF, 100V, +80%, -20% |
| C96-99 | Cer | .22μF, 50V |
| C100 | Cer | .1μF |
| C101 | Elect | 10μF, 35V, +/-20% |
| C102 | Cer | .1μF |
| C103 | Cer | 10000pF |
| C104 | Elect | 220μF, 50V |

| **CAPACITORS (Continued)** | | |
|---|---|---|
| C105 | Elect | 330μF, 50V, +/-20% |
| C106 | Elect | 1000μF, 25V |
| C107 | Elect | 100μF, 16V |
| C108,9 | Cer | 1800pF, 50V 10% |
| C110 | Cer | 10000pF |
| C111 | Elect Alum | 10μF, 16V |
| C112 | Cer | 470pF, 50V, 10% |
| C113,14 | Cer | 10000pF |
| C115,16 | Cer | .1μF |
| C117-23 | Cer | 10000pF |
| C124 | Cer | .1μF, 16V |

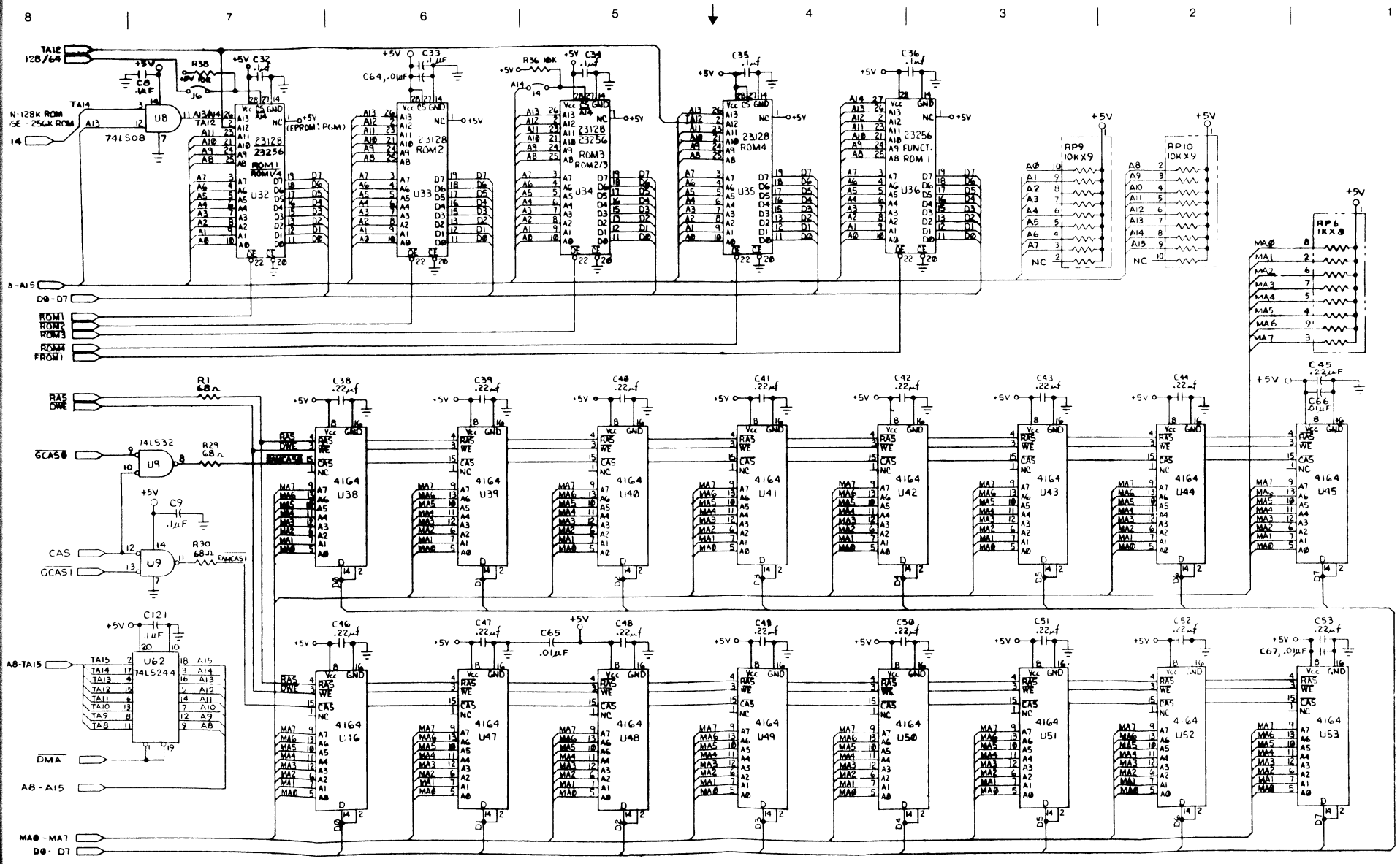
| **MISCELLANEOUS** | | |
|---|---|---|
| Y1 | 16 MHz Clock Oscillator | 325566-01 |
| Y2 | 14.31818 Crystal | 251467-01 |
| L1-4 | Inductor 2.2μH | |
| L5 | Line Filter Assy | 251878-01 |
| FB1-15, 18-20 | Ferrite Beads | |
| EMI1,2 | EMI Filter | 47pF |
| EMI3-6 | EMI Filter | 100pF |
| EMI9-11 | EMI Filter | 47pF |
| EMI12-35, 37,38 | EMI Filter | 100pF |
| EMI39 | EMI Filter | 270pF |
| EMI40-42 | EMI Filter | 100pF |
| EMI44-50 | Ferrite Bead | |
| EMI69 | EMI Filter | 100pF |
| M1 | Modulator | 251917-01 |
| SW1 | Rocker Switch | 252182-01 |
| SW2 | Push BT SPDT | 251260-01 |
| CN1 | RT Angle Card Edge | 906100-02 |
| CN3,4 | Mini D Cnnct Joy1,2 | 251057-01 |
| CN6,7 | 6 Pin Din Serial Cnnct Shld | 252166-01 |
| CN8 | 8 Pin Din Video Cnnct Shld | 252168-01 |
| CN10 | D Cnnct 9 Pin Fem Rgbi | 252024-01 |
| CN11 | 5 Pin Square Din Shielded | 252167-01 |
| CN13 | 3 Pin Header .1 Center | |
| | Shield Box | 326265-02 |
| | Shield Cap | 310407-01 |


CN3 CONTROL PORT 1 MALE DB9
CN4 CONTROL PORT 2 MALE DB9
CN5 KEYBOARD 28PIN D MALE
CASSETTE 6 PIN MALE EDGE CN2
SERIAL BUS 6 PIN DIN CONNECTOR CN6
INTERNAL SERIAL BUS 7 PIN POST CONNECTOR CN7
CN9 USER PORT (24 MALE EDGE)
CN1 EXPANSION BUS 44 PIN FEMALE EDGE
CN13 LED
I/O CONNECTOR
CIA1
CIA2
REV 6 BD
NOTE:
LOW END ONLY
HIGH END ONLY
I/O — Control Ports, Keyboard, User Port, Serial Port, Cassette Port
SCHEMATIC 310378 REV. 6
Sheet 1 of 4

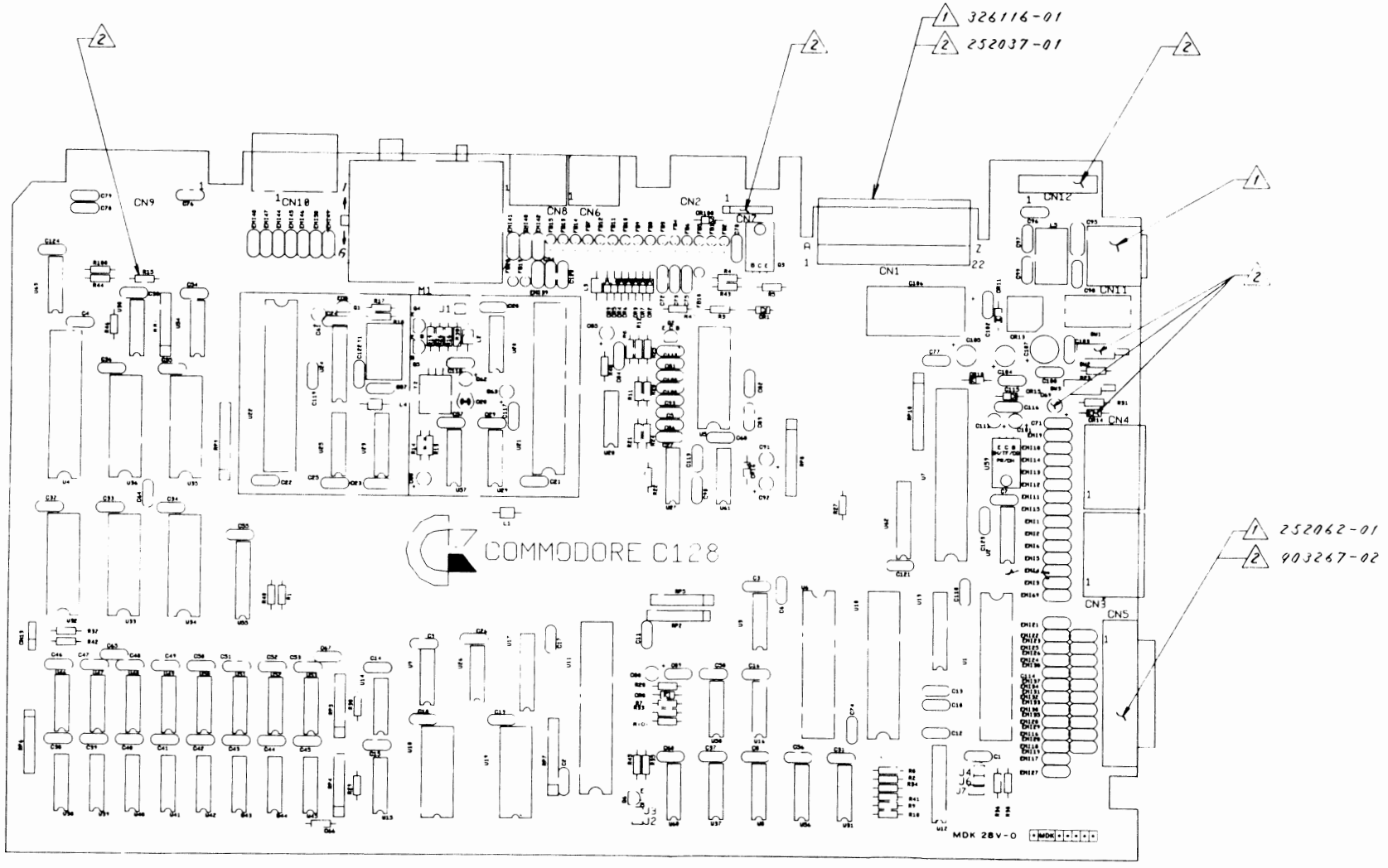

NOTE:
⚠ LOW END ONLY
⚠ HIGH END ONLY
POWER 5 PIN DIN CONNECTOR CN11
POWER 6 PIN POST CONNECTOR CN12
MICROPROCESSORS, MMU, SID, PLA, POWER SUPPLY
SCHEMATIC 310378 REV. 6
Sheet 2 of 4




**VIDEO, CLOCK, RESET**
**SCHEMATIC 310378 REV. 6**
**Sheet 3 of 4**



**RAM, ROM**
**SCHEMATIC 310378 REV. 6**
**Sheet 4 of 4**




NOTE :

1 : LOW END ONLY

2 : HIGH END ONLY

**BOARD LAYOUT**
**PCB ASSY #310379 Revision 7**
**IDENTIFYING FACTOR: On solder side of board at the EXPANSION BUS, CN1, the artwork #310381 REV. 7 appears.**

# PARTS LIST
# PCB ASSEMBLY #310379
# REV. 7

| | |
|---|---|
| | 906108-01 |
| | 906108-01 |
| | 906112-01 |
| ›cessor | 315020-01 |
| J | 310389-01 |
| ›cessor 6 MHz | 906150-02 |
| | 315012-01 |
| Char | 390059-01 |
| M — 200ns | |
| | 315009-01 |
| Cntrl | 315014-01 |
| : RAM — 150ns | |
| : RAM — 150ns | |
| Generator | 251527-03 |
| ⊦ Kern & Basic | 251913-01 |
| ic $4000 | 318018-02 |
| ic $8000 | 318019-02 |
| nal $C000 | 318020-03 |
| nction | Blank |
| : RAM — 150ns | |
| or 12V, TO-220 CASE | |
| Zener 6.8V 400MW | |
| Sub: IN4148 | |
| Zener 2.7V 500MW | |
| Sub: IN4148 | |
| Rect 50V 1A | |

**DIODES (Continued)**

| | | |
|---|---|---|
| CR13 | Bridge Rect 100V 2A | 251026-01 |
| CR15,16 | 1N914 | Sub: IN4148 |
| CR100 | 1N914 | Sub: IN4148 |

**RESISTORS** — All values in ohms, 1/4W, 5% unless noted otherwise.

| | |
|---|---|
| R1 | 68 |
| R2 | 100 |
| R3 | 1K |
| R4 | 100 |
| R5 | 470 |
| R6 | 47K |
| R7 | 560 |
| R8 | 3.3K |
| R9 | 10K |
| R10 | 3.3K |
| R11,12 | 1K |
| R13,14 | 330 1W 5% |
| R16 | 120 |
| R17 | 47 |
| R18 | 82 |
| R19 | 330 1W 5% |
| R20 | 10K |
| R21 | 470K |
| R22 | 47K |
| R23 | 100K |
| R24 | 47K |
| R25 | 100K |
| R26 | 100 |
| R27 | 4.7K |
| R28 | 10K |
| R29,30 | 68 |
| R31 | 100 |
| R32 | 180 |
| R33,34 | 10K |
| R35 | 100 |
| R36 | 10K |
| R38 | 10K |
| R39 | 120 |
| R40 | 180 |
| R41 | 680 |
| R42 | 180 |
| R43,44 | 3.3K |
| R45 | 1.2K |
| R46 | 3.3K |
| R47 | 1K |
| R48 | 10K |
| R100,101 | 1K |
| R102 | 68 |

**RESISTOR PACKS**

| | | |
|---|---|---|
| RP1 | 1K +/−2% | 8 Pin, SIP, Pin 1 Com |
| RP2 | 1K 1/8W +/−10% | 9 Pin, SIP, Pin 1 Com |
| RP3,4 | 33 +/−2% | 8 Pin, SIP, Isolated |
| RP5 | 3.3K +/−10% | 8 Pin, SIP, Pin 1 Com |
| RP6 | 1K 1/8W +/−10% | 9 Pin, SIP, Pin 1 Com |
| RP7,8 | 3.3K | 10 Pin, SIP, Pin 1 Com |
| RP9,10 | 10K | 10 Pin, SIP, Pin 1 Com |

caps are 25v, +80%, 0% unless noted otherwise.

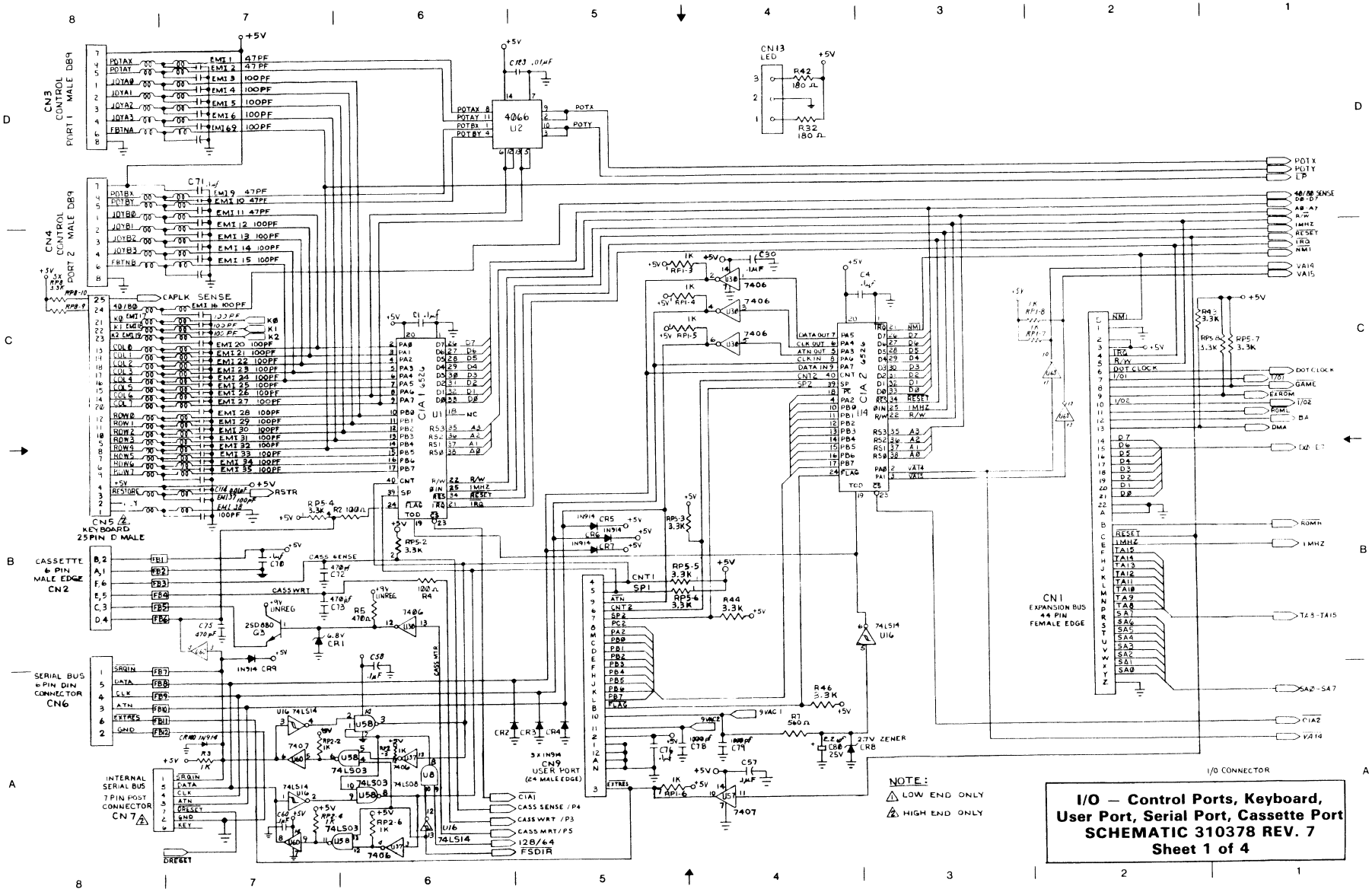
| | | | | |
|---|---|---|---|---|
| | .1μF | | | |
| | .01μF | | | |
| | .1F | | | |
| | .22μF | | | |
| | .1μF | | | |
| | 4-40pF | | | |
| | .22μF | | | |
| | .1μF | | | |
| | .22μF | | | |
| | .1μF | | | |
| | .22μF | | | |
| | .1μF | | | |
| | .22μF | | | |
| | .1μF | | | |
| | .22μF | | | |
| | .1μF | | | |
| | .1μF | | | |
| m | 10μF, | 16V | | |
| m | 100μF, | 6.3V | | |
| | .01μF | | | |
| | .01μF, | 25V | A | |
| | .01μF | | | |
| | .1μF | | | |
| | .1μF | | | |
| | 470pF, | 50V, | 10% | |
| | .01μF | | | |
| | 470pF, | 50V, | 10% | |
| | .1μF | | | |
| | .01μF | | | |
| | 1000pF, | 50V, | 10% | |
| | 2.2μF, | 25V | | |
| | 1000pF, | 50V, | 10% | |
| | 470pF, | 50V, | 10% | |
| | 220pF, | 50V, | 10% | |
| m | 10μF, | 16V | | |
| | .1μF | | | |
| | .22μF | | | |
| m | 10μF, | 16V | | |
| | 51pF, | 50V, | +/−5% | |
| | 360pF, | 50V, | +/−5% | |
| | 1μF, | 50V | | |
| m | 10μF, | 16V | | |
| | .1μF | | | |
| | .22μF | | | |
| ) | .22μF, | 100V, | +80%, −20% | |
| | .1μF, | 16V | | |
| | .1μF | | | |
| | 10μF, | 35V, | +/−20% | |
| | .1μF | | | |
| | .01μF | | | |
| | 220μF, | 50V | | |

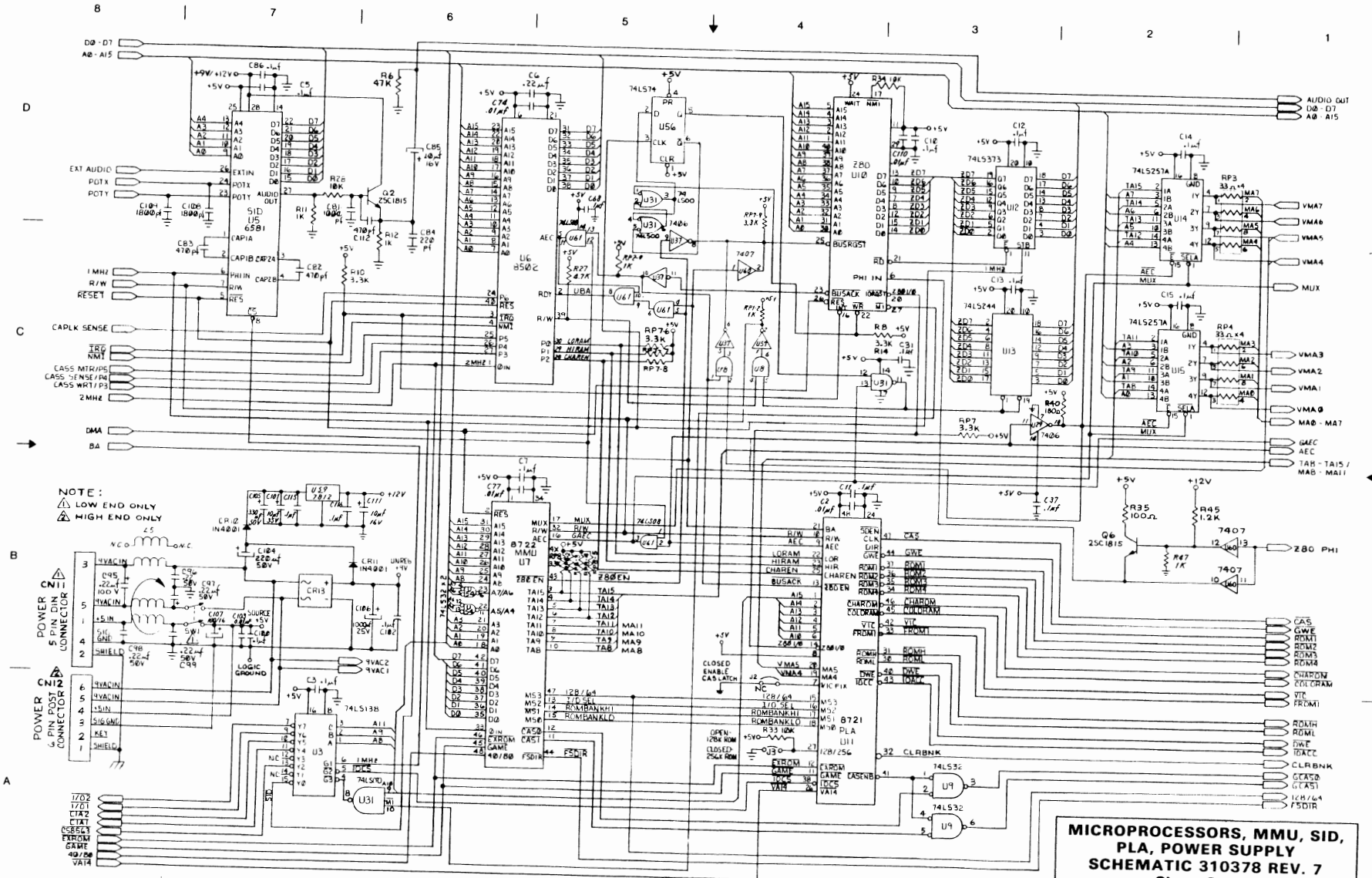
**CAPACITORS (Continued)**

| | | | | |
|---|---|---|---|---|
| C105 | Elect | 330μF, | 50V, | +/−20% |
| C106 | Elect | 1000μF, | 25V | |
| C107 | Elect | 100μF, | 16V | |
| C108,9 | Cer | .01μF | | |
| C110 | Cer | .01μF | | |
| C111 | Elect Alum | 10μF, | 16V | |
| C112 | Cer | 470pF, | 50V, | 10% |
| C113,14 | Cer | .01μF | | |
| C115,16 | Cer | .1μF | | |
| C117-23 | Cer | .01μF | | |
| C124 | Cer | .1μF, | 16V | |
| C125,126 | Cer | .01μF | | |

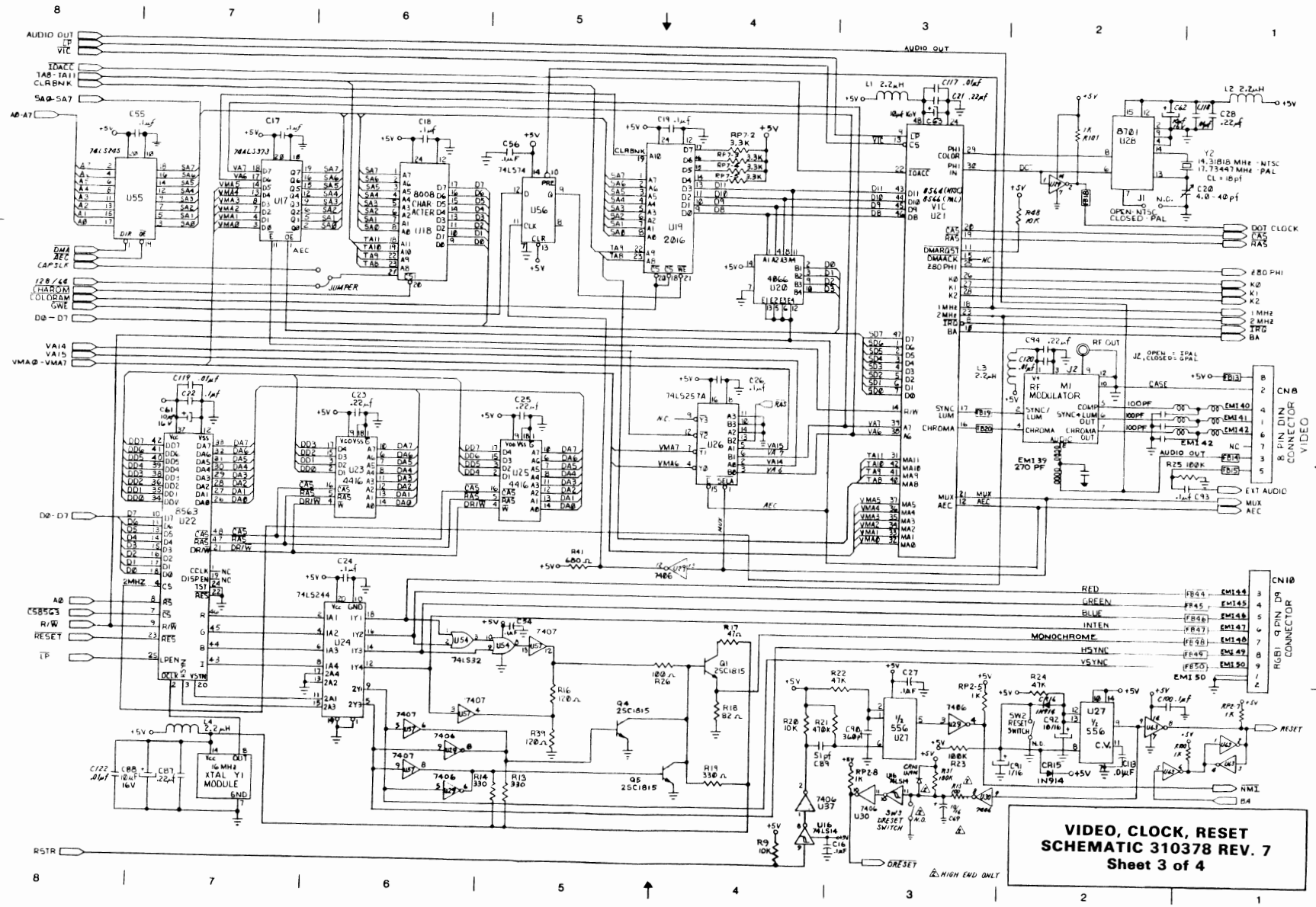
**MISCELLANEOUS**

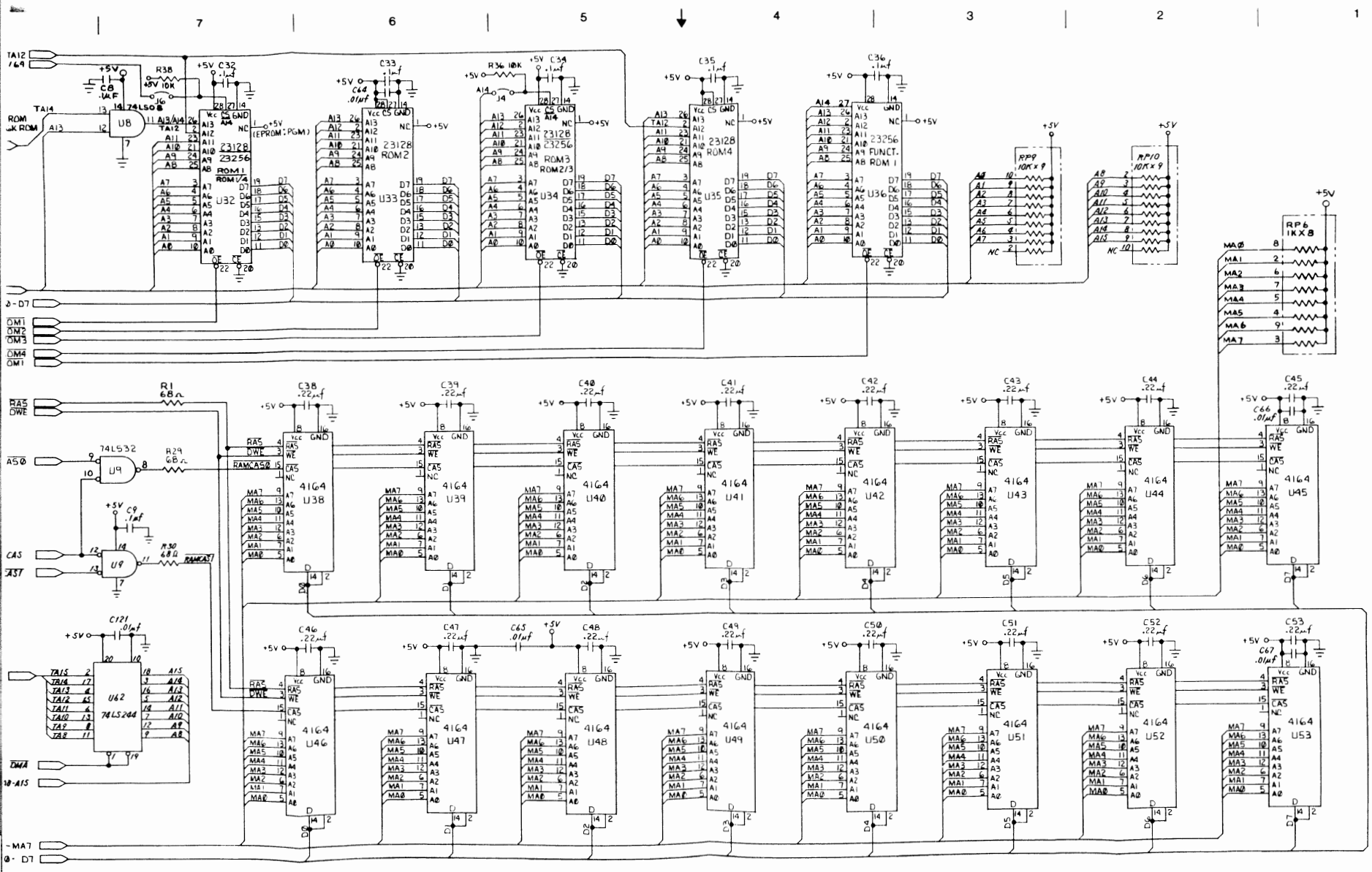
| | | |
|---|---|---|
| Y1 | 16 MHz Clock Oscillator | 325566-01 |
| Y2 | 14.31818 Crystal | 251467-01 |
| L1-4 | Inductor 2.2μH | |
| L5 | Line Filter Assy | 251878-01 |
| FB1-15, 18-20 | Ferrite Beads | |
| EMI1,2 | EMI Filter | 47pF |
| EMI3-6 | EMI Filter | 100pF |
| EMI9-11 | EMI Filter | 47pF |
| EMI12-35, 37,38 | EMI Filter | 100pF |
| EMI39 | EMI Filter | 270pF |
| EMI40-42 | EMI Filter | 100pF |
| EMI44-50 | Ferrite Bead | |
| EMI69 | EMI Filter | 100pF |
| M1 | Modulator | 251917-01 |
| SW1 | Rocker Switch | 252182-01 |
| SW2 | Push BT SPDT | 251260-01 |
| CN1 | RT Angle Card Edge | 906100-02 |
| CN3,4 | Mini D Cnnct Joy1,2 | 251057-01 |
| CN5 | Keybd Cnnct | 252062-01 |
| CN6 | 6 Pin Din Serial Cnnct Shld | 252166-01 |
| CN8 | 8 Pin Din Video Cnnct Shld | 252168-01 |
| CN10 | D Cnnct 9 Pin Fem Rgbi | 252024-01 |
| CN11 | 5 Pin Square Din Shielded | 252167-01 |
| CN13 | 3 Pin Header .1 Center | |
| | Shield Box | 326265-02 |
| | Shield Cap | 310407-01 |


CN3 CONTROL PORT 1 MALE DB9
CN4 CONTROL PORT 2 MALE DB9
CN5 KEYBOARD 25PIN D MALE
CASSETTE 6 PIN MALE EDGE CN2
SERIAL BUS 6 PIN DIN CONNECTOR CN6
INTERNAL SERIAL BUS 7 PIN POST CONNECTOR CN7
CN9 USER PORT (24 MALE EDGE)
CN1 EXPANSION BUS 44 PIN FEMALE EDGE
I/O CONNECTOR
CN13 LED
4066 U2
CIA 1 6526 U1
CIA 2 6526 U14
7406
74LS14
74LS03
7407
NOTE:
LOW END ONLY
HIGH END ONLY
I/O — Control Ports, Keyboard, User Port, Serial Port, Cassette Port
SCHEMATIC 310378 REV. 7
Sheet 1 of 4



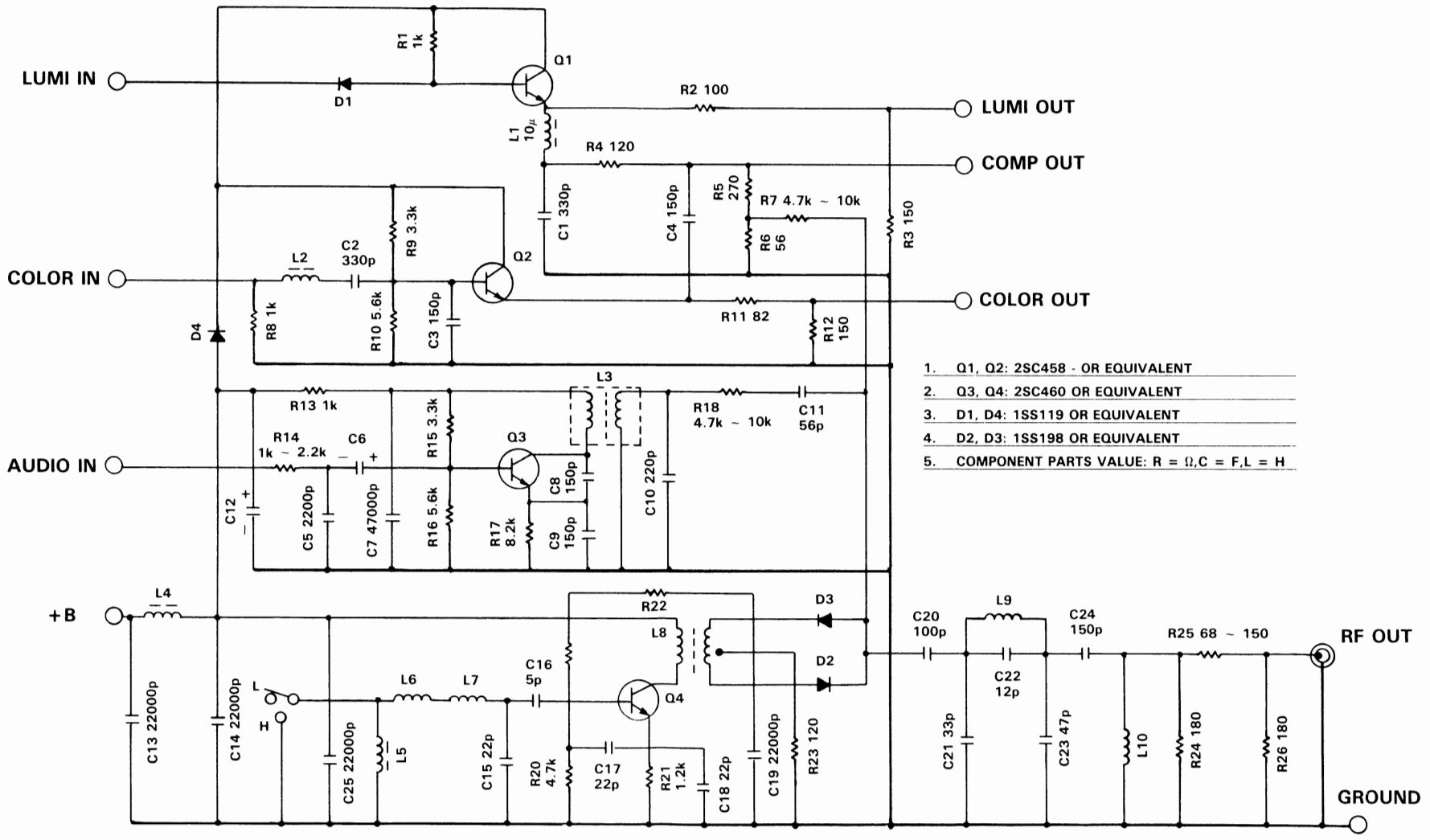

**MICROPROCESSORS, MMU, SID, PLA, POWER SUPPLY SCHEMATIC 310378 REV. 7**
**Sheet 2 of 4**


VIDEO, CLOCK, RESET
SCHEMATIC 310378 REV. 7
Sheet 3 of 4



**RAM, ROM**
**SCHEMATIC 310378 REV. 7**
**Sheet 4 of 4**

# RF MODULATOR




1. Q1, Q2: 2SC458 - OR EQUIVALENT
2. Q3, Q4: 2SC460 OR EQUIVALENT
3. D1, D4: 1SS119 OR EQUIVALENT
4. D2, D3: 1SS198 OR EQUIVALENT
5. COMPONENT PARTS VALUE: R = Ω,C = F,L = H