XMICRO-6502

Technical Manual

V1.0 2018-03-01
Applies to hardware V1

Features:

- WDC 65C02 CPU @ 4MHz
- 65C02 WAI instruction support
- Memory bank switcher providing 1MB expanded address range
- Optional bank switching for zero-page/stack
- Individually maskable card interrupt lines with priority and vectoring
- DMA system
- System clock can be received from the bus or generated onboard
- Card ID register for system self-configuration
- Fully open-source

Functional Description:

The XMICRO-6502 is an XMICRO primary-master card based on the 65C02 CPU. It boasts an advanced feature set among 6502 microcomputers, while maintaining full compatibility with the XMICRO bus. Its open design is almost entirely based upon 74-series logic ICs, making it very approachable and ensuring long-term support. The WDC 65C02 CPU provides an enhanced instruction set over older NMOS 6502s, and its fully static design allows for a much wider range of clock speeds. The card's onboard interrupt handling circuitry includes prioritized IRQ vectoring for faster response times and easier driver loading, and a DMA circuit allows for high-performance peripherals. A sophisticated bank switching system provides a dramatically expanded memory space. Optional bank switching of the Zero-page/Stack region also allows for high-performance multitasking.

Tables of Information

Table 1 – System Memory Map

Address Range	Function
$00000-$000FF	6502 Zero-Page Memory
$00100-$001FF	6502 Stack Memory
$00200-$0023F	Interrupt Vector Table (Cards 0-7)
$0023F-$00FFF	Persistent Memory
$01000-$017FF	Card Address Space (Cards 0-7)
$017FF-$03FFF	Persistent Memory
$04000-$0DFFF	General-Purpose Memory
$0E000-$0FFFF	6502 ROM
$10000-$FFFFF	General-Purpose Memory (Alternate Banks)

Table 2 – Card Configuration

Setting	Function
JP1	Enable bank switcher
JP2	Enable bank switcher wait states (leave off)
JP3	Select $\overline{\text{CLK}}$ buffer output (see Fig.1)
JP4	Select $\overline{\text{CLK}}$ buffer input (see Fig.1)
JP5	$\overline{\text{NMI}}$ enable

Table 3 – Interrupt Vector Locations

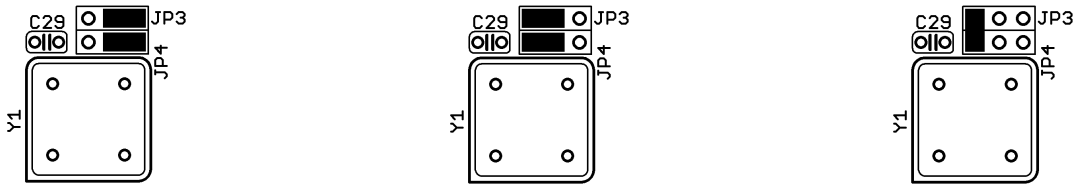
Highest Priority IRQ Signal	Vectored Address
$\overline{\text{IRQ0}}$	$0200
$\overline{\text{IRQ1}}$	$0204
$\overline{\text{IRQ2}}$	$0208
$\overline{\text{IRQ3}}$	$020C
$\overline{\text{IRQ4}}$	$0210
$\overline{\text{IRQ5}}$	$0214
$\overline{\text{IRQ6}}$	$0218
$\overline{\text{IRQ7}}$	$021C
No IRQ Signal	$023C

Table 4 – Onboard Registers

Address	D7	D6	D5	D4	D3	D2	D1	D0
$X00	N/A			ZP/Stack	A19 Value	A18 Value	A17 Value	A16 Value
$X01	IRQ7 Mask	IRQ6 Mask	IRQ5 Mask	IRQ4 Mask	IRQ3 Mask	IRQ2 Mask	IRQ1 Mask	IRQ0 Mask
$XFF	Card ID							

CPU

The system clock signal can either be received from the bus, or generated by the onboard oscillator Y1 and sent to the bus. Typically this signal will be buffered by U8, however there is also an option to bypass the buffer and connect to the bus $\overline{CLK}$ line directly. Note that the output of Y1 is always connected to the card's internal clock line, so Y1 must be removed if an external clock source is used. These options are configured with JP3 and JP4 as shown in Figure 1.



Onboard Oscillator | Buffered External $\overline{CLK}$ | Bypass Buffer

Figure 1 – System Clock Source Selection

Bank Switcher

The Bank Switcher circuit allows switching between up to 16 banks of 64k each. It generates address bits A<19..16> based on the CPU's current address and the value stored in D<3..0> of the bank register ($X00). A persistent area is active when the CPU address is in the range of $0000-$3FFF. In this range, A<19..16> are always low, leaving a protected area that is accessible regardless of the current bank. In addition, a Zero-Page/Stack Mask bit D4 may be set to remove this protection from $0000-$01FF. The bank register is reset to $00 during a system reset.

When the bank register's value is changed, there can be a significant propagation delay to the output. An optional wait-state generator can be enabled by adding a jumper link to JP2. This causes the CPU to wait for one cycle after switching banks, to provide the next addressed device with greater setup time.

Bank switching can be disabled by removing the jumper link from JP1. This causes A<19..16> to remain low.

Interrupt Vector Generator

The Interrupt Vector Generator takes the eight card-specific interrupt requests ($\overline{IRQ}$<7..0>) and generates a prioritized interrupt vector. This vector is placed on a 16-bit interrupt vector register, which is only accessible by the CPU itself during a vector pull. Each interrupt request may be individually disabled by setting the corresponding bit of the interrupt mask register ($X01). $\overline{IRQ}$<7..0> originate at the $\overline{IRQX}$ pin of the first eight backplane slots and correspond to their slot number. $\overline{IRQ0}$ is the highest priority interrupt request.

When one or more IRQ lines are asserted, the priority encoder places the value of the highest priority line on D<5..2> of the interrupt vector register's least-significant byte ($FFFE). When the CPU enters its interrupt sequence, it jumps to the location stored in this register and begins execution. Each of the interrupt vector locations contains 4 bytes of memory, and must contain a JMP instruction to an interrupt service routine. Table 3 contains the locations of each interrupt vector address.

The output of the IVG is combined with $\overline{INT}$ to allow expanded interrupt handling circuitry on large backplanes. Because of this, when the $\overline{INT}$ signal is asserted the CPU will be sent to the vector at $023C, where a hardware-specific interrupt handler can be placed. The BRK instruction will also send the CPU to the vector at $023C. In systems with no additional interrupt circuitry, $023C will conveniently only be used by the BRK instruction.

DMA System

The onboard DMA (direct memory access) system allows other devices to directly manipulate the system's memory without the intervention of the CPU. Typically this is used to achieve high speed, low-overhead operations which would take significantly more time for the CPU to perform. These operations are asynchronous in nature and do not need to occur at the same frequency as the CPU clock. They are only limited by the timing of the system components being addressed.

An interrupt request is generated for the card when the $\overline{\text{BUSACK}}$ signal is asserted by the DMA circuit, indicating that the CPU has been stopped. This interrupt request is cleared upon the first IRQ vector pull, regardless of which vector is pulled. Because of this, it is possible for the DMA interrupt to be missed if it is not the highest priority interrupt request in the system.

These are the steps in a DMA operation:

1. (Optional) A register is set on a secondary master to initiate a DMA operation.
2. The secondary master asserts $\overline{\text{BUSRQ}}$.
3. The XMICRO-6502 asserts an internal $\overline{\text{WAIT}}$ signal to stop the CPU.
4. When safe to do so, the CPU is removed from the bus and the XMICRO-6502 asserts $\overline{\text{BUSACK}}$.
5. The secondary master temporarily controls the bus and performs a DMA operation.
6. The secondary master releases the $\overline{\text{BUSRQ}}$ signal.
7. The XMICRO-6502 places the CPU back on the bus and releases $\overline{\text{BUSACK}}$.
8. (Optional) The CPU responds to the DMA interrupt request generated by this process.

The following signals enter a high-impedance state during a DMA operation: A<19..0>, D<7..0>, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{FETCH}}$

Appendix A – Hardware V1.0

V1.0 contains a defective wait state generator for the bank switcher. The wait state is triggered by a write to the bank register and cleared at the end of the first cycle. By the nature of a wait state, the CPU continues to attempt the write during the next cycle, causing an infinite loop from which the CPU can not return. Thus, JP2 should be left open to disable these wait states. Alternative solutions will need to be used if this function is required.

V1.0 was originally designed with an 8kB persistent memory area. It was later decided to increase this area to 16kB. This can be done by cutting the traces on the bottom of the board to bypass U6 pin 13, then connecting pin 13 to pin 14.