XMICRO-6502

2024-09-29

Hardware V1.0B 2409

1. General Description

The XMICRO-6502 is an XMICRO primary master card based on the WDC W65C02S CPU. The design is optimized for high-speed operation at 8MHz, and may be overclocked significantly beyond that under nominal conditions.

1.1 Features

- W65C02S CPU
- 8MHz CPU clock
- 64kB flat memory map
- Onboard 8kB EEPROM
- Priority-vectored interrupts
- XMICRO DMA

1.2 System Outline

Table 1: System Memory Map

ADDRESS RANGE	FUNCTION
$0000-$00FF	6502 Zero-Page
$0100-$01FF	6502 Stack
$0200-$02FF	Interrupt Vectors
$0300-$CFFF	RAM
$D000-$D7FF	Bus I/O (Slot 0-7)
$D800-$DFFF	Bus I/O (Slot 8-15)
$E000-$FFFF	ROM

Table 3: Card Configuration

SETTING	FUNCTION
JP1	Disable onboard oscillator
JP2	CPU cycles per ROM access
JP3	Enable ROM wait states
JP4	Enable ROM
JP5	Number of backplane slots
JP6	Oscillator divisor
JP7	CPU clock delay

Table 2: Special Addresses

ADDRESS RANGE	FUNCTION
$0000	Card Control Register
$0200	Card 0 IRQ Vector
$0204	Card 1 IRQ Vector
$0208	Card 2 IRQ Vector
$020C	Card 3 IRQ Vector
$0210	Card 4 IRQ Vector
$0214	Card 5 IRQ Vector
$0218	Card 6 IRQ Vector
$021C	Card 7 IRQ Vector
$023C	BRK/Unknown Vector
$027C	DMA Interrupt Vector

2. Address Space

The 6502's 64kB address space is exposed to the XMICRO bus at $00000-$0FFFF. Bus address space $10000-$FFFFF is not directly addressable by the card. *Table 1* lists available address ranges and their purposes.

The Bus I/O and onboard ROM ranges can be disabled using the card's *Control Register*, allowing the CPU to access bus memory in these ranges instead.

2.1 I/O

Bus I/O space is mapped to $D000-$DFFF. Each slot's I/O range occupies $100, with Slot 0 at $D000-$D0FF, Slot 1 at $D100-$D1FF, and so on. When jumper JP5 is set for an 8-slot backplane, $D800-$DFFF is not mapped to I/O and therefore may be used as additional memory.

2.2 ROM

An optional 8kB EEPROM may be installed as a boot ROM, occupying $E000-$FFFF. When jumper JP4 is open, the onboard ROM is disabled and that range falls back to bus memory.

When enabled, this ROM is only accessed during read cycles, while the system RAM in the same address range is still accessed during write cycles. In this way, ROM data may be copied into RAM at the same address before it is disabled.

As the ROM may have slower timing requirements than RAM or other devices, jumpers JP2/JP3 are used to select how many clock cycles a ROM access will use. *Table 4* lists the required settings for these options.

Table 4: ROM Cycle Length

JP2	JP3	CLOCK CYCLES
Open	-	1
Closed	2	2
Closed	3	3

3. Control Register

Table 5: Control Register

ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
$0000	0	0	0	0	0	DMA Interrupt Enable (R/W)	I/O Enable (R/W)	ROM Enable (R/W)

The Control Register is used to control and monitor certain functions of the XMICRO-6502. CR bits indicate the following:

CR Bit 0:
0: Onboard ROM is disabled
1: Onboard ROM is enabled (default)

CR Bit 1:
0: I/O address range is disabled
1: I/O address range is enabled (default)

CR Bit 2:
0: DMA interrupt is disabled (default)
1: DMA interrupt is enabled

4. Interrupts

The CPU is vectored to a different address for each available interrupt. *Table 6* lists the system's interrupts, the vector address, and the order in which simultaneous interrupts will be serviced.

Table 6: Interrupts

SIGNAL	VECTOR ADDRESS	SERVICING PRIORITY
$\overline{\text{NMI}}$	$FFFA*	First
DMA	$027C	
$\overline{\text{V0}}$	$0200	
$\overline{\text{V1}}$	$0204	
$\overline{\text{V2}}$	$0208	
$\overline{\text{V3}}$	$020C	
$\overline{\text{V4}}$	$0210	
$\overline{\text{V5}}$	$0214	
$\overline{\text{V6}}$	$0218	
$\overline{\text{V7}}$	$021C	
BRK/Unknown	$023C	Last

*16-bit address only, no instruction required.

With the exception of NMI, the vector addresses are the location of the first instruction executed by the CPU when an interrupt is triggered. Generally, as part of the CPU's startup routine, these memory locations should each be populated with an RTI instruction to handle any errant interrupts.

When a particular interrupt is to be used, an absolute JMP $XXXX instruction is placed at the vector address, to jump to the correct interrupt service routine. Any instruction is valid here, however the narrow spacing between vector addresses prevents most other instructions from being very useful.

NMI

Non-maskable interrupt. This is a special maximum-priority interrupt signal that can't be disabled and will supersede any in-progress interrupt. Unlike the other interrupts, NMI's vector address is directly set in ROM (or RAM if the onboard ROM has been disabled). Because of this, the NMI vector does not require a JMP instruction.

DMA

When the 6502's clock is stopped for a DMA operation, an interrupt is triggered. This allows a service routine to be performed afterward in case any clean-up/recovery is required. The DMA interrupt is disabled by default, and can be enabled using the *Control Register*. Reading the CR will clear the DMA interrupt, however note that the CR does not contain a status bit for this so its value is irrelevant.

Vectored IRQ

The XMICRO bus routes the general interrupt signal ($\overline{\text{IRQX}}$) from each card to the $\overline{\text{V<7..0>}}$ signal corresponding to the backplane slot that card occupies. Using these signals, the XMICRO-6502 prioritizes and vectors each card's interrupt to a unique address. This can be used along with Card ID registers to identify which card is in a slot and configure drivers accordingly.

BRK/Unknown

When an interrupt is triggered from a BRK instruction, or from the bus $\overline{\text{SINT}}$ signal with no accompanying $\overline{\text{V<7..0>}}$ signal indicating the source, this vector is used. In an 8-slot backplane, there can never be an unknown $\overline{\text{SINT}}$, so this is exclusively a software interrupt vector.

5. Recommended Settings

8MHz Asymmetrical Clock
- Y1: 24MHz
- JP1: Open
- JP6: /3
- JP7: 6ns

Onboard ROM
- JP2: 2
- JP3: Closed
- JP4: Closed

8-Slot Backplane
- JP5: 8

6. Theory of Operation

6.1 Clock Generator

The XMICRO-6502 uses a complex clock generator circuit which handles several of the CPU’s functions:

CPU Clock Divider

At the heart of the clock generator is oscillator Y1, which supplies the system with a base frequency signal (OSC). OSC enters a divider circuit at U13A-3. Jumper JP6 selects between divide-by-two and divide-by-three operation, and the resulting system clock signal (CLK) leaves U13B-9.

Divide-by-three creates an asymmetrical clock signal which extends the 6502’s phase 2 (data transfer). This can provide timing advantages at high clock frequencies above 4MHz. Ultimately, the suitability of each mode will depend on many system configuration factors affecting bus timing.

For development purposes, the onboard oscillator can be disabled by closing jumper JP1 and a TTL-level clock signal of arbitrary frequency may then be injected at connector J1.

The output CLK signal is used by the $\overline{\text{RD}}$/$\overline{\text{WR}}$ strobe generator directly, while jumper JP7 allows the CPU clock to be delayed by up to two additional gates. This added delay can be used to fine-tune data hold times.

Bus WAIT Signal

When $\overline{\text{BWAIT}}$ is asserted, U18A prevents low transitions on the CLK output, extending data hold time by multiples of OSC’s period. This stretches the 6502’s phase 2 (data transfer) indefinitely during bus wait states. CLK will remain high until the next falling edge on OSC where $\overline{\text{BWAIT}}$ is not asserted.

$\overline{\text{BWAIT}}$ is directly connected to U18A’s Schmitt-trigger input instead of a buffer. This signal requires the minimum possible delay to the CPU’s clock input during high-frequency operation. The clock generator’s wait state is also designed in such a way that CLK phases are never shortened.

DMA System

Stage 1: Secondary master requests the bus, CPU clock is stopped

If the CPU’s $\overline{\text{MLB}}$ (memory lock) signal is asserted, flip-flop U1A is held in a clear state to prevent DMA during a 6502 read-modify-write instruction. The system will wait until the next safe cycle before allowing a DMA cycle.

On the rising edge of OSC, when CLK is low (6502 phase 1, address setup), gate U7B will output a rising edge to flip-flop U1A’s clock input. At this time, if a secondary master is asserting $\overline{\text{BBUSRQ}}$, U1A is set. U1A-6 then outputs low to gate U6C-10, which locks out the divider circuit’s input and stops CLK in a low state. U1A’s output is fed into U1B, and clocked in at the next rising edge of OSC.

Stage 2: XMICRO-6502 releases the bus and acknowledges the secondary master

When U1A and U1B are both set, CPUINH and $\overline{\text{BUSAK}}$ are asserted by gates U7D and U18D respectively. These produce the same signal, inverted. CPUINH puts address/control buffers U21B, U22, U24, and U26 in a high-impedance state, while $\overline{\text{BUSAK}}$ is used to put data buffer U25 in a high-impedance state. This removes the XMICRO-6502’s address, control, and data signals from the bus to allow the secondary master to control those bus lines.

When $\overline{\text{BUSAK}}$ is asserted, flip-flop U3A is set, triggering a DMA interrupt.

Stage 3: Secondary master takes the bus

The secondary master, upon receiving the $\overline{\text{BBUSAK}}$ signal, may now begin controlling the bus. When finished, the secondary master releases the address/control/data lines, and de-asserts $\overline{\text{BBUSRQ}}$.

Stage 4: XMICRO-6502 takes the bus back

When $\overline{\text{BBUSRQ}}$ is de-asserted, U1A is cleared on the next OSC rising edge. This causes gates U7D/U18D to de-assert CPUINH/$\overline{\text{BUSAK}}$, enabling the address/control/data buffers. U1B keeps U6C-11 low for one additional OSC cycle, keeping CLK stopped to allow the bus signals to stabilize after the buffers are enabled. This ensures the setup times for the XMICRO-6502’s returning bus signals are no less than in a normal cycle.

$\overline{\text{RD}}$/$\overline{\text{WR}}$ Strobes

A 6502 data transfer cycle is fundamentally different from that of the XMICRO bus. The 6502’s transfer is relative to the rising edge of its input clock signal, and a R/$\overline{\text{W}}$ signal indicates which direction data is being transferred. The XMICRO bus uses separate $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals with no additional clock, where the transfer occurs relative to a rising edge of either signal. The XMICRO-6502 uses a “strobe generator” circuit to create these signals as required by the bus.

During CLK’s positive phase, gate U14A asserts $\overline{\text{RD}}$ if the CPU’s R$\overline{\text{WB}}$ signal is high, while U14B asserts $\overline{\text{WR}}$ if R$\overline{\text{WB}}$ is low.

Gate U6A is used to prevent $\overline{\text{RD}}$ and $\overline{\text{WR}}$ from being asserted when the CPU is accessing onboard resources, to avoid invalid read/write operations on the bus. The timing of U6A’s inputs and output are critical to prevent invalid assertions of $\overline{\text{RD}}$/$\overline{\text{WR}}$, which may cause memory corruption or unwanted I/O access.

6.2 Interrupt Vectors

To simplify interrupt handling and improve response times, an interrupt vectoring system is used to direct the CPU to a unique address for each available interrupt signal. In this way, an interrupt request from a card may directly invoke an interrupt service routine for itself, avoiding complex and time-consuming routines to discover which device is calling an interrupt.

Bus signals $\overline{\text{V<7..0>}}$ as well as the onboard DMA interrupt are fed into priority encoder U30. The highest-priority interrupt needing to be serviced is encoded into part of an address and clocked into Interrupt Vector Register U29 at the beginning of each CPU cycle. This produces the least-significant byte of the hardware-vectored address, while the most-significant byte on U28 is hard-wired to $02.

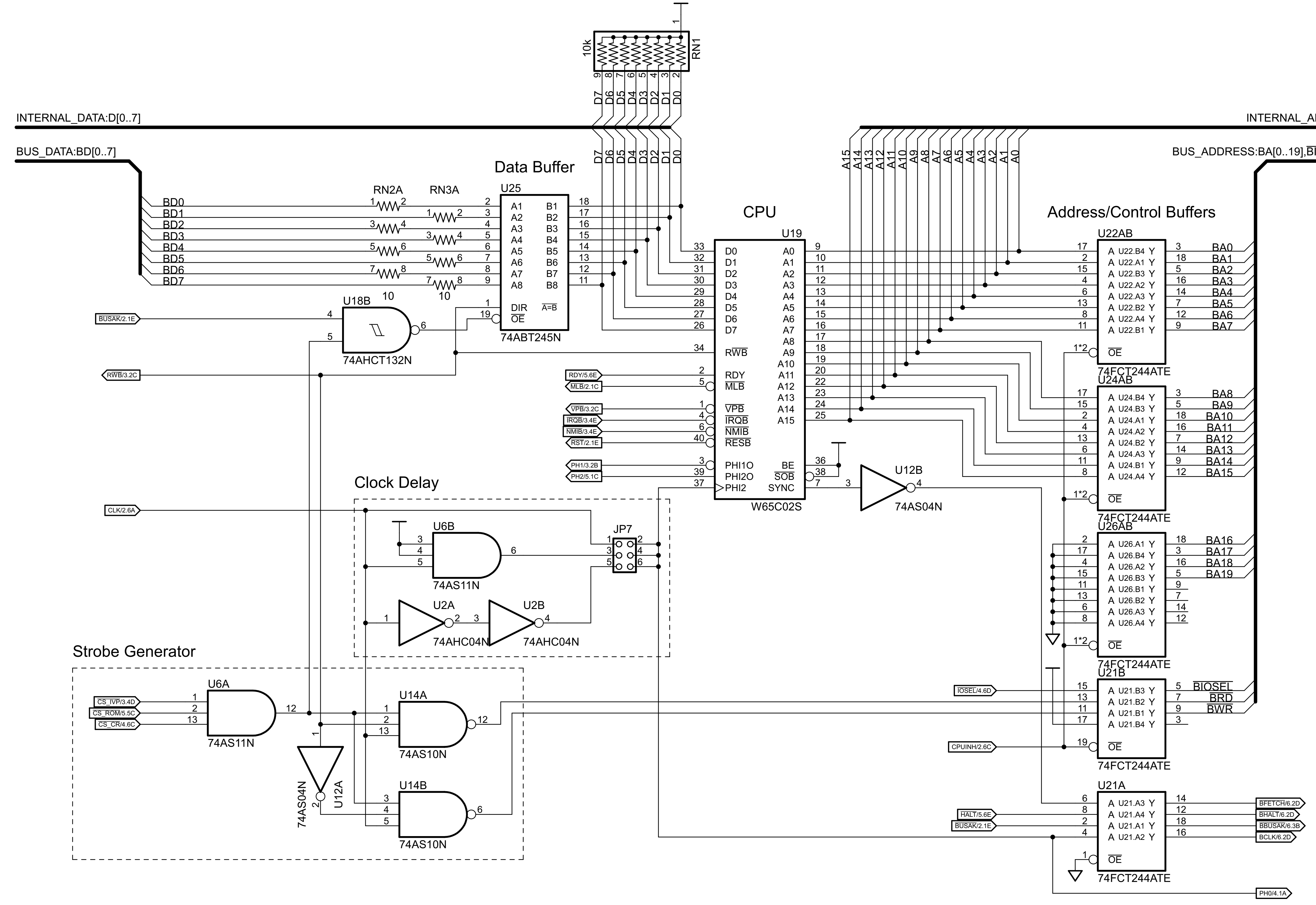
When any interrupt line is asserted (with the exception of $\overline{\text{NMI}}$), the CPU’s $\overline{\text{IRQB}}$ line is asserted, triggering an interrupt at the end of the current instruction. U11 detects when a 16-bit IRQ vector is being accessed, and overrides any other memory to supply the encoded interrupt vector at address $FFFE-$FFFF. Note that the interrupt vector register is only accessible during this “vector pull” operation.

These hardware vectors are not directly programmable, but send the CPU program counter to a normal memory location where typically a JMP or RTI instruction are placed by software.

6.3 I/O

Because the 6502 does not have a separate I/O address space, the XMICRO bus’s I/O addresses are mapped to memory. Comparator U15 detects when an address in the I/O range is accessed, and asserts the $\overline{\text{IOSEL}}$ signal to activate the bus’s I/O systems. Jumper JP5 selects whether the card supports eight or sixteen slots. When configured for eight slots, the address range for slots 9-15 ($D800-$DFFF) remains addressable as memory.

Bit 1 (IO Enable) of the CSR can be used to disable the I/O comparator, leaving the address range available as normal memory. Because of this, the CSR is not located in the card’s I/O space, as this setting would make it impossible to access again once disabled.



Title: CPU

XMICRO

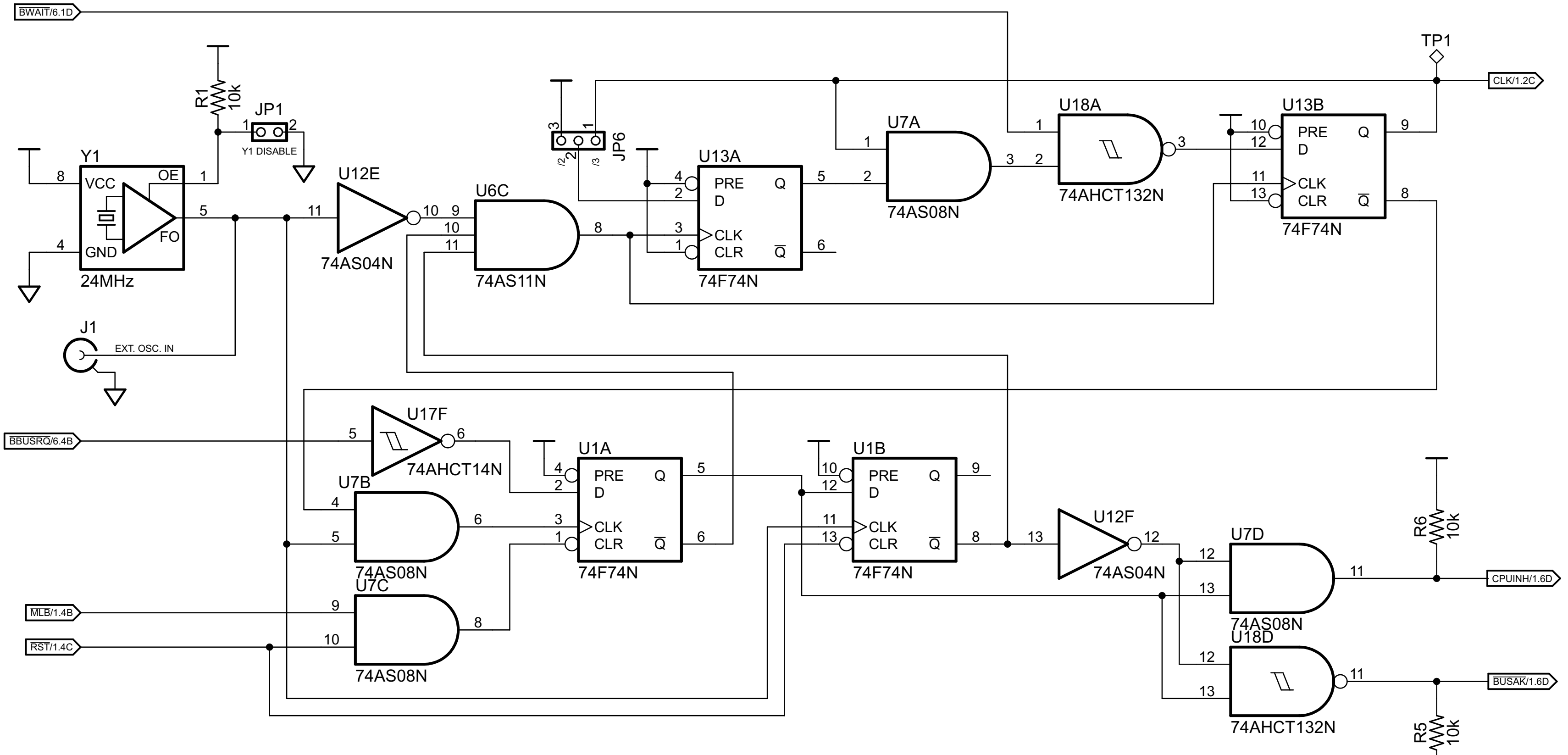
Project: XMICRO-6502

Revision: 1.0B

Drawn by: Chris Weir

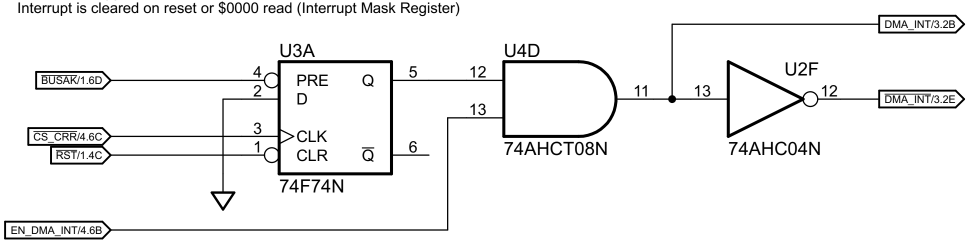
not saved!

Sheet: 1/6

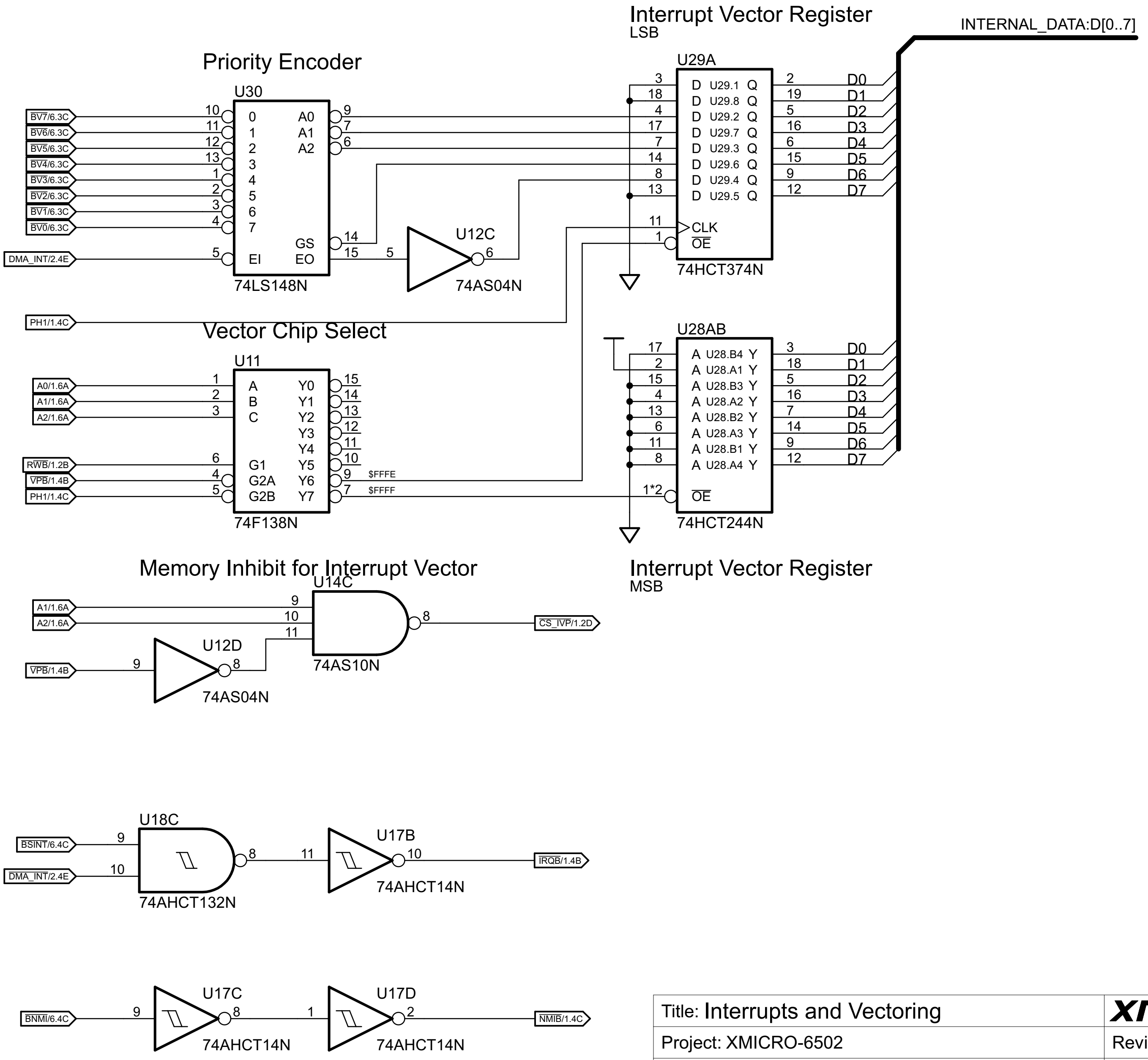


DMA Interrupt

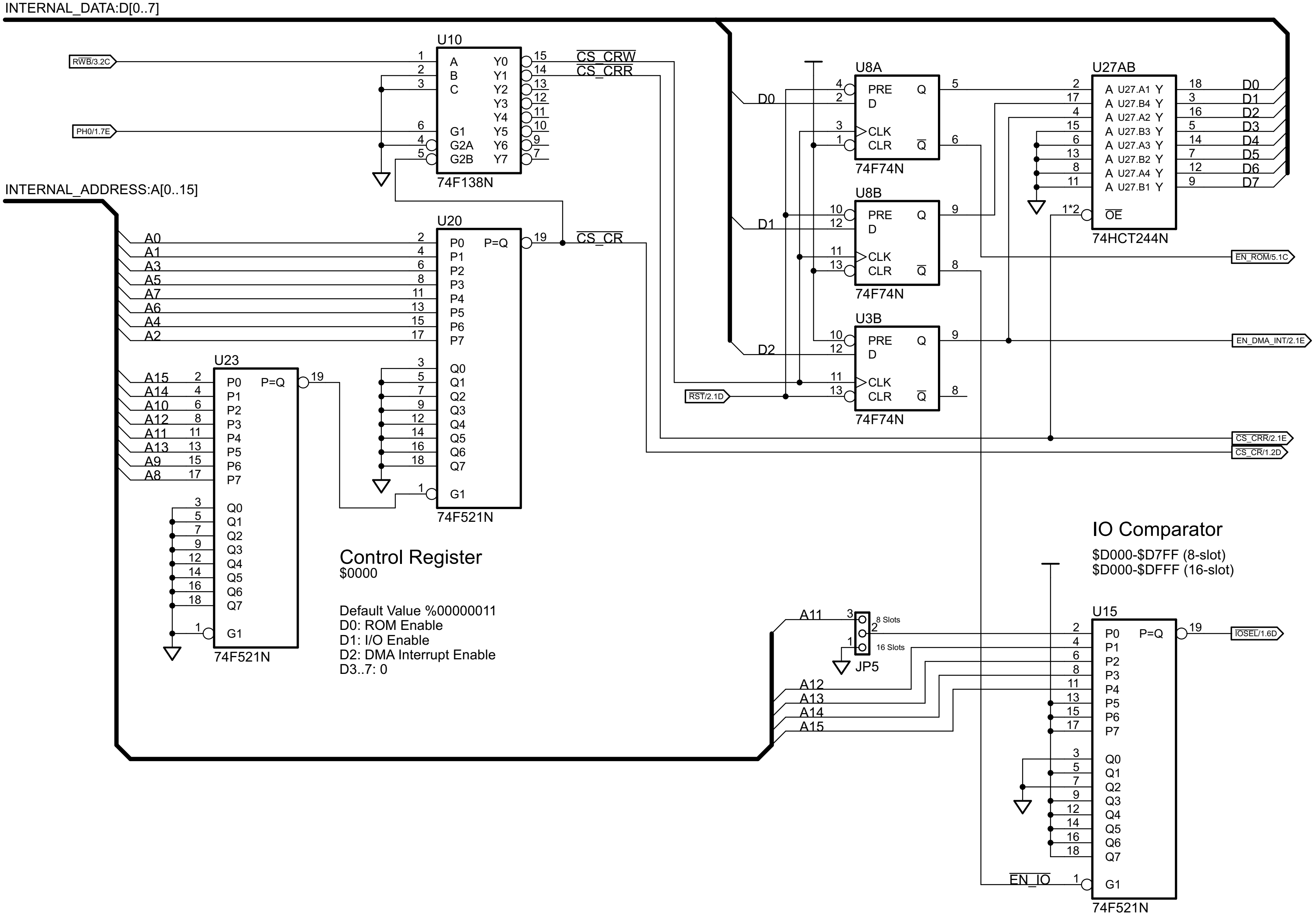
Interrupt is triggered upon BUSACK.
Interrupt is cleared on reset or $0000 read (Interrupt Mask Register)



Title: Clock Generator	XMICRO
Project: XMICRO-6502	Revision: 1.0B
Drawn by: Chris Weir not saved!	Sheet: 2/6



Title: Interrupts and Vectoring		XMICRO	
Project: XMICRO-6502		Revision:	1.0B
Drawn by: Chris Weir	not saved!	Sheet:	3/6

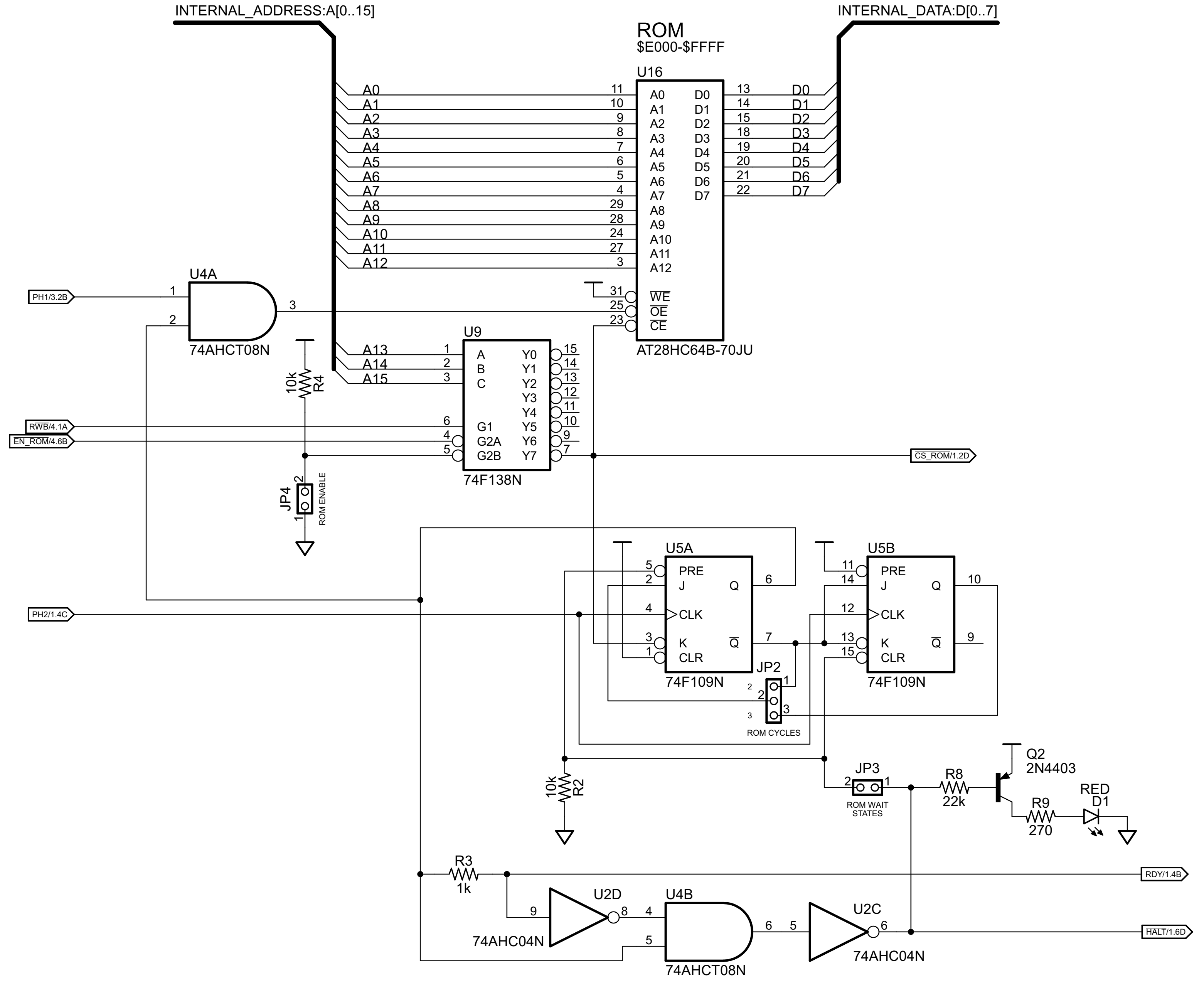


Control Register
$0000

Default Value %00000011
D0: ROM Enable
D1: I/O Enable
D2: DMA Interrupt Enable
D3..7: 0

IO Comparator
$D000-$D7FF (8-slot)
$D000-$DFFF (16-slot)

Title: Control Register and I/O	XMICRO
Project: XMICRO-6502	Revision: 1.0B
Drawn by: Chris Weir not saved!	Sheet: 4/6



Title: ROM

XMICRO

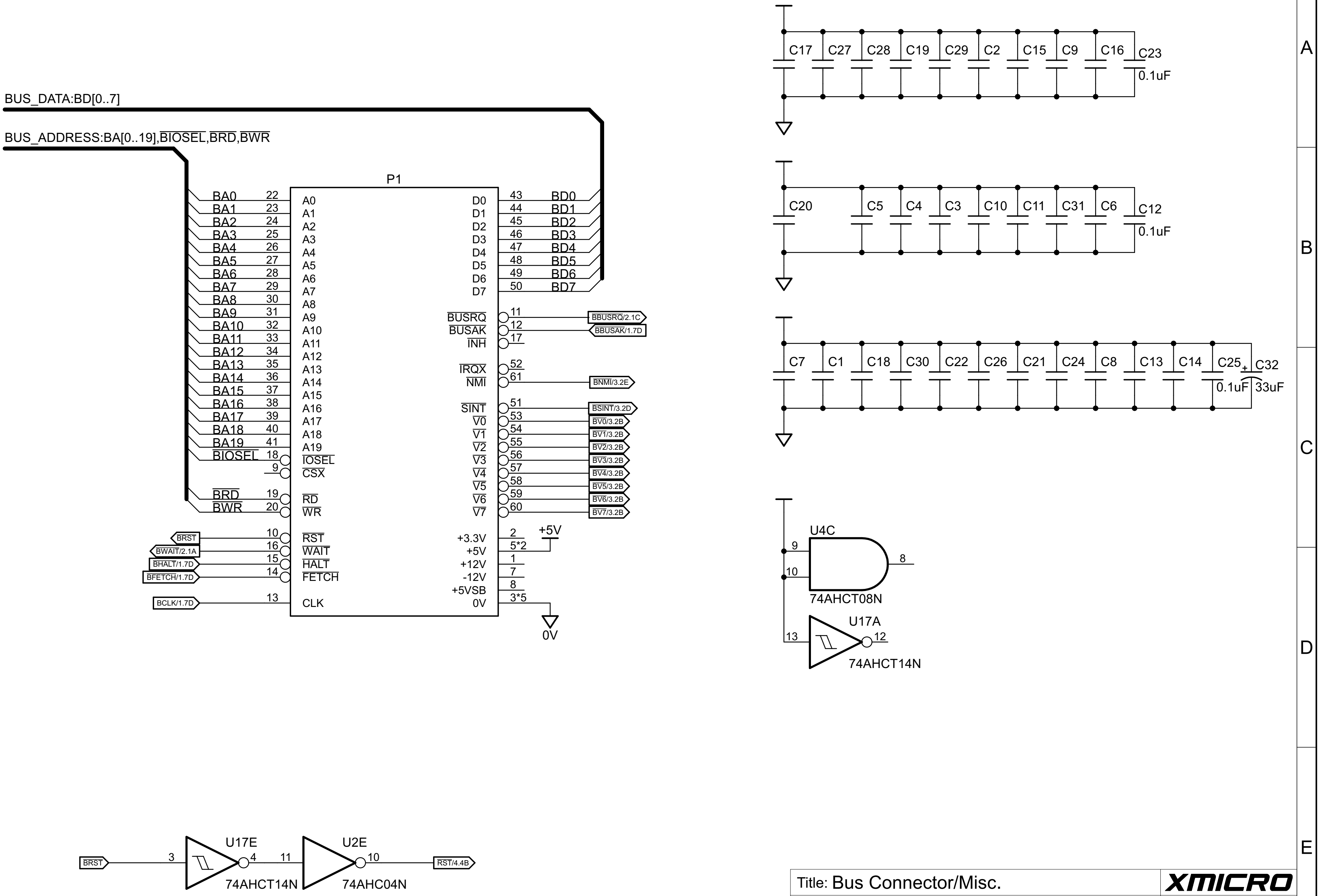
Project: XMICRO-6502

Revision: 1.0B

Drawn by: Chris Weir

not saved!

Sheet: 5/6



Title: Bus Connector/Misc.	XMICRO
Project: XMICRO-6502	Revision: 1.0B
Drawn by: Chris Weir not saved!	Sheet: 6/6