W65C02SOC64
EIT Microcontroller
Datasheet

Page Left Intentionally Blank

Table of Contents

Page Left Intentionally Blank

DOCUMENT REVISION HISTORY

Version	Date	Authors	Description
1.0	09/20/2024	Bill Mensch, David Gray	Initial Document Entry
1.1	01/22/2026	Bill Mensch, David Gray	Updated introduction, features, pin assignments, formatting

Page Left Intentionally Blank

1 INTRODUCTION

The **W65C02SOC64 EIT Microcontroller** is a customizable System-on-Chip (SoC) platform developed by The Western Design Center, Inc. (WDC) as an educational, research, and product-development vehicle for **Embedded Intelligence Technology (EIT)**. It is designed for adoption in **VLSI and SoC design courses**, for **self-paced learners**, and as a scalable foundation for **commercial embedded systems**.

This datasheet documents both the **W65C02SOC64 ASIC SoC** and the **W65C02SOC64M08SA FPGA-based microcontroller**, implemented on the Altera MAX10M08SA device for functional emulation and rapid prototyping. Portions of this datasheet describe implementation considerations for ASIC tapeout using the **GlobalFoundries GF180MCU Open Source PDK**, enabling a complete educational path from RTL to silicon.

WDC's definition of this complete System-on-Chip (SoC) includes the W65C02RTL microprocessor, serial and parallel IO, monitor ROM, and RAM for both code and data. The platform is intentionally structured to make the principles of Embedded Intelligence observable, measurable, and designable in hardware and software.

Embedded Intelligence is characterized by the functional cycle of **Sensing, Processing, Communicating, and Actuating (SPCA)**. The W65C02SOC64 architecture explicitly supports SPCA through memory-mapped, addressable hardware resources and deterministic control, allowing designers and students to directly map real-world system behavior to silicon implementations.

To support extensibility and experimentation, the W65C02SOC64 is composed of a **base microcontroller** and a set of **micromodules**. A micromodule is a well-defined function or group of functions that expands system capability while preserving architectural clarity. Examples include memory blocks, ADCs, DACs, timers, counters, multipliers, dividers, and custom accelerators. Micromodules are described in **Verilog HDL** and may be integrated for use in both FPGA and ASIC design flows.

This modular approach reflects the principles of the **Theory of Embedded Intelligence (TEI)**, in which intelligence increases capability and complexity over time through embedded experience. By enabling designers to incrementally add functionality, the W65C02SOC64 provides a practical framework for understanding how intelligence—natural or engineered—emerges from structured SPCA interactions.

For software development and system interaction, **MyMENSCH™** incorporates a **W65C51RTL** UART driving a **CH340 serial-to-USB interface**, supporting WDCTools for both **Assembly and C language development**. This configuration provides a transparent and deterministic interface between software, hardware, and the external environment.

The W65C02SOC64 platform intentionally emphasizes **clarity, determinism, and architectural visibility**. These attributes make it especially well-suited for education, research, and long-lifecycle embedded products, while serving as a practical embodiment of Embedded Intelligence Technology—where sensing, processing, communication, and actuation are not abstract concepts, but directly realized in silicon.

1.1 Key Features of the W65C02SOC64 EIT Microcontroller

- IO Operating Voltage – 3.3V/5V
- Core Operating Voltage – 1.8V (TSMC), 3.3V/5V (Global Foundries 180MCU)
- System Operation Speed – 14.7456 MHz
- W65C02RTL MPU
- W65C22RTL VIA (x2)
- W65C51RTL ACIA (x2) – 1.8432 MHz Created with divider logic from XCLK of MAX10
- W65CGPIO ports (8 pins per port)
- SPI Primary
- I2C Primary
- WDC ~2K byte Monitor for boot loading and debugging code
- 8K bytes User code SRAM boot loaded from USB or copied from external SPI serial FLASH memory
- 8K bytes for data SRAM
- JTAG available on MyMENSCH™

1.2 W65C02SOC64 Pin Function List for 64 Pin QFN

- 2x IO Ring VDD (TSMC 180nm uses 3.3V, GF180MCU is either 3.3V or 5V)
- 2x Core VDD (TSMC 180nm uses 1.8V Core, GF180MCU is either 3.3V or 5V)
- 4x VSS (2x Core, 2x IO Ring; GF180MCU will have all 4 VSS tied together)
- 1x XCLK Input used to generate other clocks - *ACIA Clock generated by dividing XCLK on MAX10
- 1x UCLK 1.8432MHz ACIA Clock for ASIC
- 1x RESB
- 1x NMIB
- 20x VIA_A
- 19x VIA_B for 8x LEDs (PA7-0); CB1, CB2, CA2 PB7-0 for mikroBUS connectors and RTC on MySPCA
- 4x ACIA_A with 2x Handshake (GPIOE7 for RTSB and GPIOE3 for CTSB)
- 2x ACIA_B – No Handshake
- 2x I2C
- 3x SPI
- 2x SPI Chip Selects (MySPCA Uses GPIOE4 for SPI SRAM and GPIOE5 for SPI FLASH)

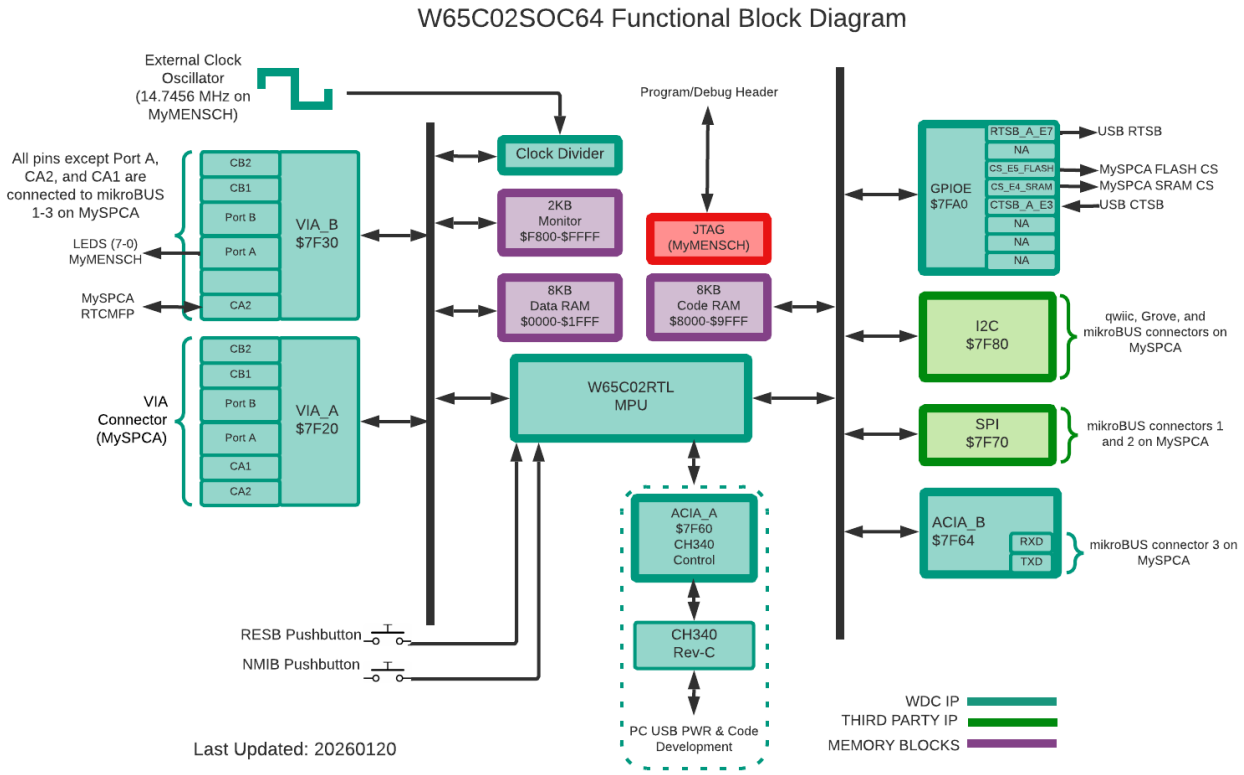
1.3 Optional MicroModule Considerations

The features that were chosen for the W65C02SOC64 took into consideration the limited number of pins available. WDC allows licenses to add their own custom MicroModules. Here are just a few examples.

1.) ADC
2.) RTC
3.) AI Applications (Quantized Neural Networks, FP8, etc.)
4.) PWM
5.) RGB LED Drivers

1.4 Functional Block Diagram

The following block diagram is for the W65C02SOC64.



2 MicroModule Descriptions

2.1 Clock and Reset MicroModule

This handles the clock and reset logic for the system. There are no registers accessible by the MPU. XCLK is the main input clock for the system.
On MyMENSCH™ is driven by a 14.7456MHz clock oscillator, and a divider derives the clocks for the system. PHI2 for the MPU is a divide by 4 or 3.6864MHz clock. The memory clocks on the SOC build use divide by 2 or 7.3728MHz. The UCLK input clock for the ACIA is a divide by 8 or 1.8432MHz. XTLI for the ACIA can also run at 3.6864MHz, which can utilize baud rates up to 115200 baud.
On MyCHIP, XCLK and UCLK are on separate pins. This allows for testing a faster system clock and allowing the ACIA input clock (UCLK) to remain 1.8432MHz.

2.2 W65C02RTL MicroProcessor MicroModule

Refer to the W65C02S Datasheet for the Microprocessor Programming Model, Status Register Coding and complete information. More information is found in ***Programming the 65816: Including the 6502, 65C02 and 65802*** Manual available through Amazon.

2.3 Priority Interrupt Controller MicroModule Information

The Priority Interrupt Control MicroModule controls the priority and memory map for interrupts. Each interrupt is connected to the Priority Interrupt Control MicroModule for prioritizing.

Interrupt Enable Registers for the various interrupts are the interrupt enable by the various enable bits. Reading the various IER and IFR bits determines the interrupt that occurred. By prioritizing the interrupts one can determine which interrupt occurred in the associated interrupt handler routine. Notice that any of the 8 interrupts for a GPIO 8-bit port will cause a GPIO vectored interrupt to occur.

Priority Encoded Interrupt Vector MicroModule

Vector Address	Label	Function
0xFFFE,F	IRQBRK	BRK – Software Interrupt
0xFFFC,D	IRQRES	RES – “REStart” Interrupt
0xFFFA,B	IRQNMI	Non-Maskable Interrupt
0xFFF8,9	IRQGPIO_HS	GPIO Interrupt for UART Handshaking (for all 8 pins)
0xFFF6,7	IRQVIA_B	VIA_B Interrupt
0xFFF4,5	IRQVIA_A	VIA_A Interrupt
0xFFF2,3	IRQSPI	SPI Interrupt
0xFFF0,1	IRQI2C	I2C Interrupt

2.4 Memory Map

Start	End	Size	Description
0xF800	0xFFFF	2048 B	2048 Byte Monitor
0x8000	0x9FFF	8192 B	8K Byte Protected Bootloaded RAM
0x7FA0	0x7FA3	4 B	GPIOE
0x7F80	0x7F8F	16 B	I2C
0x7F70	0x7F7F	16 B	SPI
0x7F64	0x7F67	4 B	ACIA_B
0x7F60	0x7F63	4 B	ACIA_A
0x7F50	0x7F5F	16 B	RESERVED
0x7F40	0x7F4F	16 B	RESERVED
0x7F30	0x7F3F	16B	VIA_B
0x7F20	0x7F2F	16 B	VIA_A
0x7F10	0x7F1F	16 B	RESERVED
0x7F08	0x7F0B	4 B	RESERVED
0x0000	0x1FFF	8192 B	8K Byte SRAM

2.5 VIA Port MicroModule

The W65C02SOC64 features two Versatile Interface Adapters (VIA) based on the W65C22S.See Memory Map for base addresses and W65C22S Datasheet for full register descriptions.
On the MySPCA board, all 20 IO pins from VIA_A are connected to the VIA connector. See section 3.3 for VIA Connector pinout. VIA_B IO are used in a variety of ways. Port A is connected to the 8 LEDS on MyCHIP. VIA_B_CA2 is connected to the RTC chip's MultiFunction Pin (MFP) on MySPCA. VIA_B_CA1 is not connected. All other pins are connected to the 3 mikroBUS module connectors. See section 3.4 for mikroBUS module connector pinouts.

2.6 GPIO Port MicroModules

One GPIO Port is included in the W65C02SOC64 design. This 5 Register (8-bit) version supports edge sense interrupts and has a PIO Register (PIOx), Data Direction Register (DDRx), Interrupt Flag Register (IFRx), Interrupt Enable Register (IERx), and Edge Sense Register (ESRx). This GPIO (GPIOE) is intended to be used as handshake logic for the 2 ACIA and Chip Select Outputs for SPI.
See Memory Map for base addresses.

GPIO MicroModule - 5 Register Version

Address = Base + 4		GPIO_ESR: GPIO Edge Sense Register					Reset Value = 0x00	
7:0->	ESR7	ESR6	ESR5	ESR4	ESR3	ESR2	ESR1	ESR0
Bit	**Name**	**Access**	**Description**					
7 – 0	ESR[7:0]	R/W	1 = Positive Edge Sense for PIO7-0					
			0 = Negative Edge Sense for PIO7-0					

Address = Base + 3		GPIO_IER: GPIO Interrupt Enable Register					Reset Value = 0x00	
7:0->	IER7	IER6	IER5	IER4	IER3	IER2	IER1	IER0
Bit	**Name**	**Access**	**Description**					
7 – 0	IER[7:0]	R/W	1 = Enable Interrupt on inputs for PIO7-0					
			0 = Disable Interrupts on inputs for PIO7-0					

Address = Base + 2		GPIO_IFR: GPIO Interrupt Flag Register					Reset Value = 0x00	
7:0->	IFR7	IFR6	IFR5	IFR4	IFR3	IFR2	IFR1	IFR0
Bit	**Name**	**Access**	**Description**					
7 – 0	IFR[7:0]	R/W	1 = Interrupt Occurred on inputs for PIO7-0					
			0 = Interrupts did not occur on inputs for PIO7-0					

Address = Base + 1		GPIO_DDR: GPIO Data Direction Register					Reset Value = 0x00	
7:0->	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
Bit	**Name**	**Access**	**Description**					
7 – 0	DDR[7:0]	R/W	1 = PIO data direction set to Output PIO7-0					
			0 = PIO data direction set to Input PIO7-0					

Address = Base		GPIO_DATA: GPIO Data Register					Reset Value = 0x00	
7:0->	PIO7	PIO6	PIO5	PIO4	PIO3	PIO2	PIO1	PIO0
Bit	**Name**	**Access**	**Description**					
7 – 0	PIO[7:0]	R/W	1 = PIO line is logic 1 value read and sets a 1 value on write for PIO7-0					
			0 = PIO line is logic 0 value read and sets a 0 value on write for PIO7-0					

2.7 ACIA MicroModules

The SoC has two Asynchronous Communications Interface Adapters (ACIA) used to transfer information to and from various communications devices such as LoRa, GSM, Bluetooth, Wi-Fi radios, etc. See the Memory Map for base addresses. The baud rates are derived from 1.8432MHz XTLI input.

The Micro USB connector is dual purpose. It is both the power connector that powers the board and the USB code port. The CH340 chip that interfaces to the connector has been pre-programmed so that when the board is power from a USB power on a computer, the chip will request 500mA of current from the host machine. The board can also be powered by any USB port that supplies 5V DC. Note that the board has a 3v3 and 1v8 voltage regulators.

In addition to the power, the USB port serves as an interface to WDC's tool suite for debugging and loading programs into the onboard SRAM.

ACIA Control Register

Address = Base + 3		ACIA_CTRL: ACIA Control Register					Reset Value = 0x00	
7:0->	SBN	WL1	WL0	RSC	SBR3	SBR2	SBR1	SBR0
HWRES	0	0	0	1	0	0	0	0
SWRES	-	-	-	1	-	-	-	-

Bit	Name	Access	Description
7	SBN	R/W	1 = 2 Stop bits, 1 ½ Stop bits for WL = 5, 1 Stop bit for WL = 8 and parity
			0 = 1 Stop bit
6	WL1	R/W	11 = 5 bits
			10 = 6 bits
5	WL0	R/W	01 = 7 bits
			00 = 8 bits
4	RSC	R/W	1 = Baud rate
			0 = RSC clock source
3	SBR3	R/W	1110 = 9600, 1111 = 19200
			1100 = 4800, 1101 = 7200
2	SBR2	R/W	1010 = 2400, 1011 = 3600
			1000 = 1200, 1001 = 1800
1	SBR1	R/W	0110 = 300, 0111 = 600
			0100 = 134.58, 0101 = 150
0	SBR0	R/W	0010 = 75, 0011 = 109.92
			0000 = 115.2K, 0001 = 50

ACIA Command Register

Address = Base + 2	ACIA_CMR: ACIA Command Register						Reset Value = 0x00	
7:0->	PCM1	PCM0	PME	REM	TIC1	TIC0	IRD	DTR
HWRES	0	0	0	0	0	0	0	0
SWRES	-	-	-	0	0	0	0	0

Bit	Name	Access	Description
7	PCM1	R/W	11 = Space parity
			10 = Mark parity
6	PCM0	R/W	01 = Odd parity
			00 = Even parity
5	PME	R/W	1 = Parity enabled
			0 = Parity disabled
4	REM	R/W	1 = Receiver Echo Mode not available
			0 = Receiver Echo Mode not available
3	TIC1	R/W	11 = RTSB = low, Transmitter interrupt disabled, Transmit Break
			10 = RTSB = low, Transmitter interrupt disabled
2	TIC0	R/W	01 = RTSB = low, Transmitter interrupt enabled
			00 = RTSB = high, Transmitter interrupt disabled
1	IRD	R/W	1 = Receiver Interrupt Disabled
			0 = Receiver Interrupt Enabled
0	DTR	R/W	1 = Data Terminal Ready
			0 = Data Terminal Transmitter Not Ready

ACIA Status Register

Address = Base + 1		ACIA_STR: ACIA Status Register					Reset Value = 0x10	
7:0->	IRQ	DSRB	DCDB	TDRE	RDRF	OVRN	FE	PE
HWRES	0	0	0	1	0	0	0	0
SWRES	-	-	-	1	-	-	-	-
Bit	**Name**	**Access**	**Description**					
7	IRQ	R/O	1 = Interrupt has occurred					
			0 = No Interrupt					
6	DSRB	R/O	1 = Not ready and not clear to send data					
			0 = Ready and clear to send data					
5	DCDB	R/O	1 = DCD Not Detected					
			0 = DCD Detected					
4	TDRE	R/O	1 = Empty					
			0 = Not Empty					
3	RDRF	R/O	1 = Full					
			0 = Not Full					
2	OVRN	R/O	1 = Overrun has occurred					
			0 = No overrun					
1	FE	R/O	1 = Framing error detected					
			0 = No framing error					
0	PE	R/O	1 = Parity error detected					
			0 = No parity error					
Address = Base + 1		W/O	Program Reset aka SWRES					

Address = Base		ACIA_DR: ACIA Data Register					Reset Value = 0x00	
7:0->	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
Bit	**Name**	**Access**	**Description**					
7 - 0	DR[7-0]	R/W	R = Read Receiver Data Register					
			W = Write Transmitter Data Register					

2.8 I2C Interface MicroModule

For complete I2C register descriptions and I2C Operation, refer to the "I2C Design Specification".

I2C Status Register

Address = Base + 5		SR: I2C Status Register					Reset Value = 0x00	
7:0->	RxACK	WCOL	BUSY	RESERVED	WFFULL	WFEMPTY	RFFULL	RFEMPTY

Bit	Name	Access	Description
7	RxACK	R/O	1 = No acknowledge received
			0 = Acknowledge received
6	BUSY	R/W	1 = After start I2C bus busy signal detected
			0 = After stop I2C bus busy signal detected
5	RSRVD	R/O	1 = Never
			0 = Always
4	RSRVD	R/O	1 = Never
			0 = Always
3	RSRVD	R/O	1 = Never
			0 = Always
2	RSRVD	R/O	1 = Never
			0 = Always
1	TIP	R/O	1 = Transfer in progress when transferring data
			0 = When transfer complete
0	IF	R/O	1 = Interrupt is set when one byte is transferred, processor interrupt request if IEN bit is set.
			0 = No interrupt

I2C Command Register

Address = Base + 4		CR: I2C Command Register					Reset Value = 0x00	
7:0->	STA	STO	RD	WR	ACK	RSVRD	RSVRD	IACK

Bit	Name	Access	Description
7	STA	R/W	1 = Generate start condition
			0 = Do not generate start condition
6	STO	R/W	1 = Generate stop condition
			0 = Do not generate stop condition
5	RD	R/W	1 = Read from slaver
			0 = Do not read from slave
4	WR	R/W	1 = Write slave
			0 = Do not write slave
3	ACK	R/W	1 = NACK
			0 = ACK
2	RSVRD	R/W	1 = Never
			0 = Always
1	RSVRD	R/W	1 = Never
			0 = Always
0	IACK	R/W	1 = Clear a pending interrupt
			0 = Don’t clear a pending interrupt

I2C Receive Register

Last byte received via I2C.

Address = Base + 3		RXR: I2C Receive Register					Reset Value = 0x00	
7:0->	RXR7	RXR6	RXR5	RXR4	RXR3	RXR2	RXR1	RXR0

Bit	Name	Access	Description
7 - 0	RXR[7-0]	R/O	**R = Read Receiver Data Register**
			W = no operation

I2C Transmit Register

7:1 RW Next byte to transmit via I2C 0 RW In case of a data transfer this bit represent the data's LSB. In case of a slave address transfer this bit represents the RW bit. '1' for reading from slave '0' for writing to slave

Address = Base + 2		TXR: I2C Transmit Register					Reset Value = 0x00	
7:0->	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
Bit	**Name**	**Access**	**Description**					
7 - 0	TXR[7-0]	R/W	R = Read Receiver Data Register					
			W = Write Transmitter Data Register					

I2C Control Register

The core responds to new commands only when the 'EN' bit is set. Pending commands are finished. Clear the 'EN' bit only when no transfer is in progress, i.e. after a STOP command, or when the command register has the STO bit set. When halted during a transfer, the core can hang the I2C bus.

Address = Base +1		CTR: I2C Control Register					Reset Value = 0x10	
7:0->	EN	IEN	RSRVD	RSRVD	RSRVD	RSRVD	RSRVD	RSRVD
Bit	**Name**	**Access**	**Description**					
7	EN	R/W	1 = I2C Core enabled					
			0 = I2C Core disabled					
6	IEN	R/W	1 = I2C Core interrupt enabled					
			0 = I2C Core interrupt disabled					
5	RSRVD	R/W	1 = Never					
			0 = Always					
4	RSRVD	R/W	1 = Never					
			0 = Always					
3	RSRVD	R/W	1 = Never					
			0 = Always					
2	RSRVD	R/W	1 = Never					
			0 = Always					
1	RSRVD	R/W	1 = Never					
			0 = Always					
0	RSRVD	R/W	1 = Never					
			0 = Always					

I2C Clock Prescale Register

This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 4*SCL clock internally. The prescale register must be programmed to this 4*SCL bitrate. Change the value of the prescale register only when the 'EN' bit is cleared.
Example: CLK_I = 32MHz, desired SCL = 100 KHz
Prescale = 32MHZ = 80 (dec) = 50 (hex) 4 * 100 KHz
Reset value: 0xFFFF

Address = Base		PRER: I2C Clock Prescale Register					Reset Value = 0xFF	
7:0->	PRER7	PRER6	PRER5	PRER4	PRER3	PRER2	PRER1	PRER0

Bit	Name	Access	Description
7 - 0	PRER[7-0]	R/W	R = Read Receiver Data Register W = Write Transmitter Data Register

2.9 SPI MicroModule

The SPI MicroModule described in the standard SPI Specification found in this link.

SPI Extension Register

Address = Base + 3		SPER: SPI Extension Register						Reset Value = 0x00	
7:0->	ICNT1	ICNT0	RESERVED	RESERVED	RESERVED	RESERVED		ESPR1	ESPR0
Bit	**Name**	**Access**	**Description**						
7	ICNT1	R/W	11 = SPIF is set after every four completed transfers						
			10 = SPIF is set after every three completed transfers						
6	ICNT0	R/W	01 = SPIF is set after every two completed transfers						
			00 = SPIF is set after every completed transfer						
5	RESERVED	R/W	1 = Never						
			0 = Always						
4	RESERVED	R/W	1 = Never						
			0 = Always						
3	RESERVED	R/W	1 = Never						
			0 = Always						
2	RESERVED	R/W	1 = Never						
			0 = Always						
1	ESPR1	R/W	11 = Reserved, do not use						
			10 = Add these two bits to the SPI Clock Rate 0=512, 01=1024, 02=2048, 03=4096						
0	ESPR0	R/W	01 = Add these two bits to the SPI Clock Rate 0=8, 01=64, 02=128, 03=256						
			00 = Add these two bits to the SPI Clock Rate 00=2, 01=4, 02=16, 03=32						

SPI Data Register

Address = Base + 2		SPDR: SPI Data Register					Not Initialized on Reset	
7:0->	SPDR7	SPDR6	SPDR5	SPDR4	SPDR3	SPDR2	SPDR1	SPDR0
Bit	**Name**	**Access**	**Description**					
7 - 0	SPDR[7-0]	R/W	R = Read SPI Data buffer					
			W = Write SPI Data buffer					

SPI Status Register

Address = Base + 1		SPSR: SPI Status Register							Reset Value = 0x05
7:0->	SPIF	WCOL	RESERVED	RESERVED	WFFULL	WFEMPTY	RFFULL	RFEMPTY	
Bit	**Name**	**Access**	**Description**						
7	SPIF	R/W	1 = SPI Interrupt Flag is set on completion of a transfer block						
			0 = SPI not interrupting						
6	WCOL	R/W	1 = SPI Core write collision when SPI data register when Write FIFO is full						
			0 = SPI Core disabled						
5	RESERVED	R/O	1 = Never						
			0 = Always						
4	RESERVED	R/O	1 = Never						
			0 = Always						
3	WFFULL	R/O	1 = Write FIFO full						
			0 = Write FIFO not full						
2	WFEMPTY	R/O	1 = Write FIFO empty						
			0 = Write FIFO not empty						
1	RFFULL	R/O	1 = Read FIFO full						
			0 = Read FIFO not full						
0	RFEMPTY	R/O	1 = Read FIFO empty						
			0 = Read FIFO not empty						

SPI Control Register

Address = Base		SPCR: SPI Control Register					Reset Value = 0x10	
7:0->	SPIE	SPE	RESERVED	MSTR	CPOL	CPHA	SPR1	SPR0

Bit	Name	Access	Description
7	SPIE	R/W	1 = SPI Interrupt Enabled
			0 = SPI Interrupt Disabled
6	SPE	R/W	1 = SPI Core enabled
			0 = SPI Core disabled
5	RESERVED	R/W	
4	MSTR	R/W	1 = Master
			0 = Slave
3	CPOL	R/W	1 = Negative Clock Polarity
			0 = Positive Clock Polarity
2	CPHA	R/W	1 = Clock Phase Not Shifted
			0 = Clock Phase Shifted
1	SPR1	R/W	These values are used with the ESPR bits to determine the extended clock rate.
			Refer to the SPI Datasheet for detailed selection information.
0	SPR0	R/W	These values are used with the ESPR bits to determine the extended clock rate.
			Refer to the SPI Datasheet for detailed selection information.

3 Pinouts

3.1 Left IO Connector J3 on MyMENSCH™/MySPCA™

The J3 left connector has 46 IO, 2x 3v3 power and 2x VSS pins. Ball assignments labeled NA are Non-Assigned pins.

J3 – Left Expansion Connector					
Pin	***Signal Name***	***FPGA Ball***	***Pin***	***Signal Name***	***FPGA Ball***
1	VSS	-	2	VDD	-
3	I2C_SCL	L4	4	NA	L3
5	I2C_SDA	K6	6	NA	K5
7	CS_SRAM_E4	H4	8	NA	N2
9	VIA_A_CB2	M4	10	NA	N3
11	VIA_A_CB1	M5	12	NA	N4
13	VIA_A_PB7	L5	14	NA	N5
15	VIA_A_PB6	N7	16	NA	N6
17	VIA_A_PB5	N8	18	NA	M7
19	VIA_A_PB4	M9	20	NA	M8
21	VIA_A_PB3	M10	22	NA	N9
23	VIA_A_PB2	M11	24	NA	N10
25	VIA_A_PB1	N12	26	NA	N11
27	VIA_A_PB0	M13	28	NA	M12
29	VIA_A_PA7	L13	30	NA	L12
31	VIA_A_PA6	K13	32	NA	K12
33	VIA_A_PA5	K8	34	NA	J8
35	VIA_A_PA4	J9	36	NA	L10
37	VIA_A_PA3	K10	38	NA	L11
39	VIA_A_PA2	K11	40	NA	J10
41	VIA_A_PA1	J12	42	NA	H10
43	VIA_A_PA0	K7	44	NA	J13
45	VIA_A_CA1	H13	46	NA	G12
47	VIA_A_CA2	G13	48	NA	F12
49	VDD	-	50	VSS	-

3.2 Right IO Connector J4 on MyMENSCH™/MySPCA™

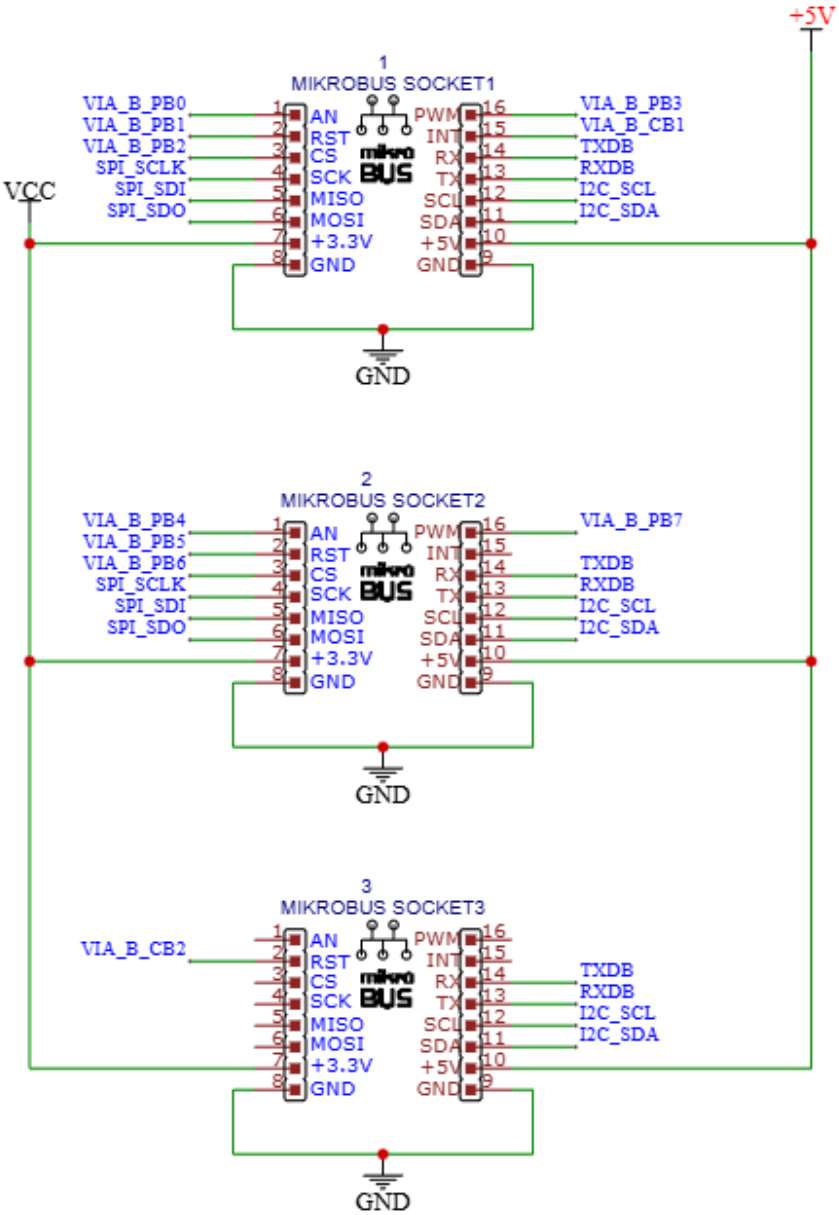
The J4 right connector has 46 IO, 2x 3v3 power and 2x VSS pins. Ball assignments labeled NA are Non-Assigned pins. The ADC inputs are not used for this build and should not be connected to anything.

J4 – Right Expansion Connector					
Pin	***Signal Name***	***FPGA Ball***	***Pin***	***Signal Name***	***FPGA Ball***
1	VDD	-	2	VSS	-
3	NA	H3	4	AGND	(G2)
5	NA	H1	6	DNU/ADC_IN0	(F5)
7	NA	H2	8	3v3REF	(F6)
9	DNU/ADC_IN5	F1	10	5vBAT	(G1)
11	DNU/ADC_IN6	E1	12	NA	(E5)
13	DNU/ADC_IN7	C1	14	DNU/ADC_IN1	D1
15	DNU/ADC_IN8	B1	16	VIA_B_CA2	C2
17	NA	B2	18	VIA_B_CB2	E3
19	NA	A2	20	VIA_B_CB1	E4
21	NA	B3	22	CS_FLASH_E5	E6
23	NA	B4	24	VIA_B_PB0	A3
25	NA	B5	26	VIA_B_PB1	A4
27	NA	B6	28	SPI_SDI	A5
29	NA	B7	30	SPI_SDO	A6
31	NA	A7	32	SPI_SCLK	D9
33	NA	A8	34	VIA_B_PB2	E8
35	NA	C9	36	VIA_B_PB3	F8
37	NA	C10	38	VIA_B_PB4	A9
39	NA	B10	40	VIA_B_PB5	A10
41	NA	B11	42	TXD_B	A11
43	NA	B12	44	RXD_B	A12
45	NA	B13	46	VIA_B_PB6	C11
47	NA	C13	48	VIA_B_PB7	C12
49	VSS	-	50	VDD	-

3.3 VIA Connector on MySPCA

MySPCA VIA Connector			
Pin	***Signal Name***	***Pin***	***Signal Name***
1	VDD	2	VSS
3	VIA_A_CA1	4	VIA_A_CA2
5	VIA_A_PA0	6	VIA_A_PA1
7	VIA_A_PA2	8	VIA_A_PA3
9	VIA_A_PA4	10	VIA_A_PA5
11	VIA_A_PA6	12	VIA_A_PA7
13	NO CONNECT	14	NO CONNECT
15	VIA_A_CB1	16	VIA_A_CB2
17	VIA_A_PB0	18	VIA_A_PB1
19	VIA_A_PB2	20	VIA_A_PB3
21	VIA_A_PB4	22	VIA_A_PB5
23	VIA_A_PB6	24	VIA_A_PB7
25	VSS	26	VDD

3.4 mikroBUS Module Connectors on MySPCA



3.5 W65C02SOC64 QFN64 Pinout

Pin	WDC QFN64 Signal	FPGA Assignment	MyMensch Connector		Pin	WDC QFN64 Signal	FPGA Assignment	MyMensch Connector
1	VIA_A_CB2	M4	J3-9		33	VIA_B_PB7	C12	J4-48
2	VIA_A_CB1	M5	J3-11		34	VIA_B_PB6	C11	J4-46
3	VIA_A_PB7	L5	J3-13		35	RXD_B	A12	J4-44
4	VIA_A_PB6	N7	J3-15		36	TXD_B	A11	J4-42
5	VIA_A_PB5	N8	J3-17		37	VIA_B_PB5	A10	J4-40
6	VIA_A_PB4	M9	J3-19		38	VIA_B_PB4	A9	J4-38
7	VIA_A_PB3	M10	J3-21		39	VIA_B_PB3	F8	J4-36
8	VDD - RING	-	J3-2/49,J4-1/50		40	VSS - CORE	-	J3-1/50,J4-2/49
9	VSS - CORE	-	J3-1/50,J4-2/49		41	VDD - RING	-	J3-2/49,J4-1/50
10	VIA_A_PB2	M11	J3-23		42	VIA_B_PB2	E8	J4-34
11	VIA_A_PB1	N12	J3-25		43	SPI_SCLK	D9	J4-32
12	VIA_A_PB0	M13	J3-27		44	SPI_SDO	A6	J4-30
13	VIA_A_PA7	L13	J3-29		45	SPI_SDI	A5	J4-28
14	VIA_A_PA6	K13	J3-31		46	VIA_B_PB1	A4	J4-26
15	VIA_A_PA5	K8	J3-33		47	VIA_B_PB0	A3	J4-24
16	VIA_A_PA4	J9	J3-35		48	CS_FLASH_E5	E6	J4-22
17	VIA_A_PA3	K10	J3-37		49	VIA_B_CB1	E4	J4-20
18	VIA_A_PA2	K11	J3-39		50	VIA_B_CB2	E3	J4-18
19	VIA_A_PA1	J12	J3-41		51	VIA_B_CA2	C2	J4-16
20	VIA_A_PA0	K7	J3-43		52	VIA_B_PA0	J1	-
21	VIA_A_CA1	H13	J3-45		53	VIA_B_PA1	J2	-
22	VIA_A_CA2	G13	J3-47		54	VIA_B_PA2	K1	-
23	NMIB	H5	-		55	VIA_B_PA3	K2	-
24	VSS - RING	-	J3-1/50,J4-2/49		56	VDD - CORE	-	J3-2/49,J4-1/50
25	VDD - CORE	-	J3-2/49,J4-1/50		57	VSS - RING	-	J3-1/50,J4-2/49
26	RESB	H4	-		58	VIA_B_PA4	L1	-
27	XCLK	G5/PLL	-		59	VIA_B_PA5	L2	-
28	UCLK	PLL	-		60	VIA_B_PA6	M1	-
29	RTSB_A_E7	D13	-		61	VIA_B_PA7	M2	-
30	CTSB_A_E3	E13	-		62	I2C_SCL	L4	J3-3
31	RXD_A	G9	-		63	I2C_SDA	K6	J3-5
32	TXD_A	F13	-		64	CS_SRAM_E4	M3	J3-7

4 W65C02SOC64 Test and Evaluation Information

4.1 Monitor Operation, Wiggle Testing and BIST

The W65C02SOC64 has 2KB ROM at $F800. This memory holds a basic hardware test and monitor with firmware for basic hardware functions. For educational purposes of adding Built-In-Self Test (BIST) functions and application software, the monitor has functionality of adding
An example Memory BIST program that can be developed and loaded is a Serial March Memory Test using the following Algorithm:

Algorithm Steps (Serial March)

Increasing Address

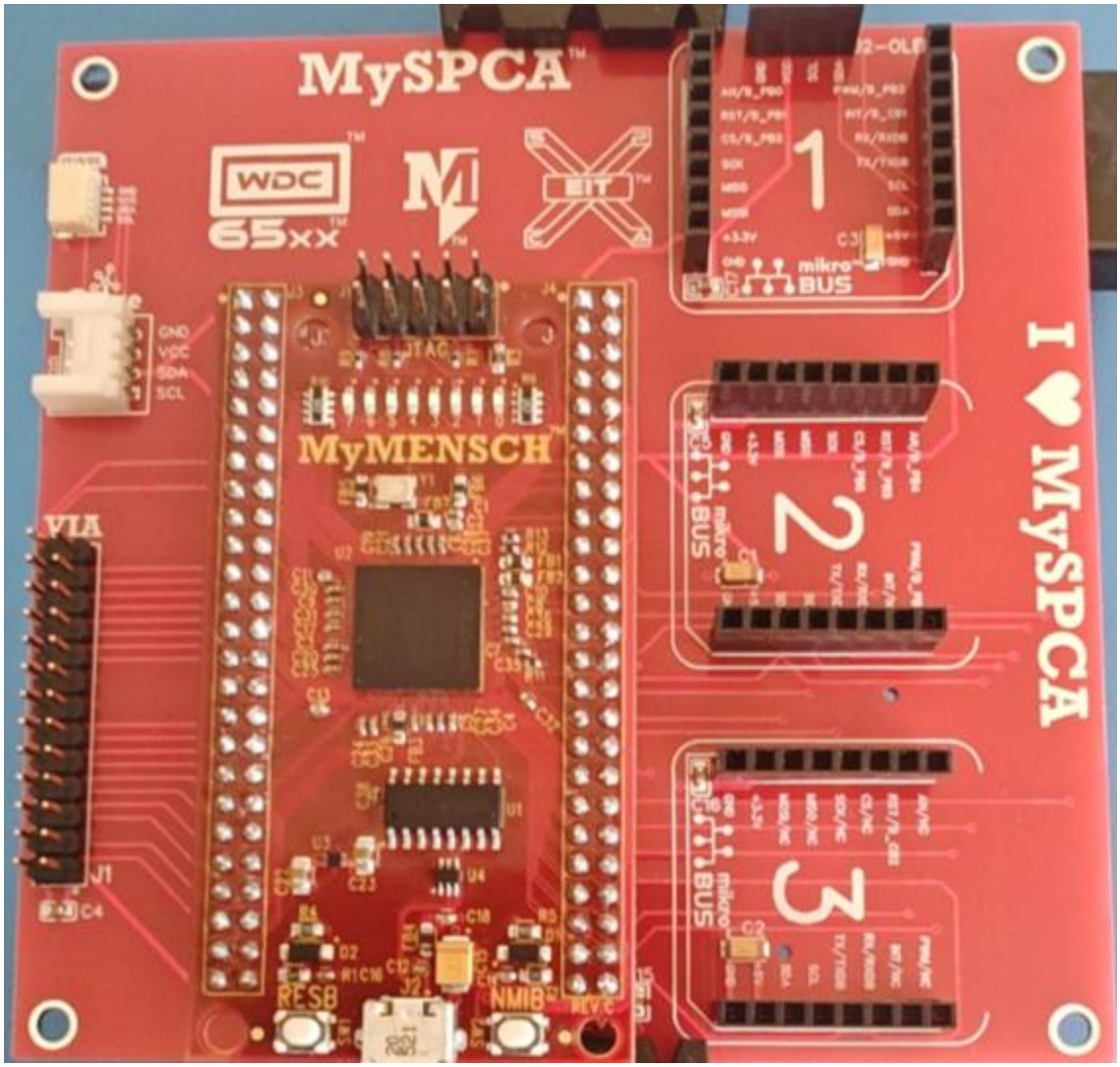
- write 0s with up addressing order (to initialize)
- Read 0s, write 1s with up addressing order
- Read 1s, write 0s with up addressing order

Decreasing address

- Read 0s, write 1s with down addressing order
- Read 1s, write 0s with down addressing order
- Read 0s with down addressing order

4.2 W65C02SOC64 MySPCA™ Board

The MySPCA™ is a board that MyMENSCH™ plugs into for prototyping a more complete system around the W65C02SOC64 FPGA EIT Microcontroller. This board features I2C connectors compatible with either SparkFun's Qwiic system or SEEED Studio's Grove system. There is also a general pin head for I2C OLED Display. For SPI, I2C and UART connectivity there are three MikroBUS connectors. The 20 IO from VIA_A are located in the lower left corner (J1-VIA).



4.3 W65C02SOC64 MyCHIP™ Board

The MyCHIP™ is a carrier board for the taped out W65C02SOC64 Microcontroller. Below is a picture of the board showing the location of the QFN64 package die.

The MyCHIP™ has the same connectors as the MyMENSCH™. MyCHIP™ has the MicroUSB connector, onboard 5V0 to 3V3/1V8 voltage regulation, a 14.7456MHz clock oscillator, 1.8432MHz oscillator. NMIB and RESB pushbuttons all on the board. Design files are available from WDC.

5 FCC Compliance

The Western Design Center, Inc. (WDC) provides the enclosed product under the following conditions: All MyMENSCH™, MySPCA™ and MyCHIP™ boards are intended for use for Education, Engineering Development or Evaluation Purposes ONLY and are not considered by WDC to be a finished consumer product. This board should be handled with caution using good electronics handling practices. This board is compliant per RoHS/Green directives. It does not fall within the scope of directives such as FCC, CE, and UL. It generates uses and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC rules.

6 Ordering Information

The W65C02SOC64 Microcontroller is ordered through the MPW tapeout service provider. MyMENSCH™, MySPCA™ and MyCHIP are available from WDC.