OBSOLETE

Application Report

SNOA435B–March 2004–Revised April 2013

AN-1294 COP8TAB9/TAC9 ISP HANDBOOK—Intro to ISP

ABSTRACT

This application report describes the COP8TAB9/TAC9 In System Programming (ISP) Software Interface. The In System method of programming the flash memory from an external source is thoroughly discussed.

Contents

List of Figures

List of Tables

1 Introduction

In-System Programming (ISP) allows the user to re-program a microcontroller without physical removal. The COP8TAB9/TAC9 ISP Software allows the user to program the flash memory in three ways. A user may choose to program the flash memory by using the Boot ROM's user support portion, the emulation support portion (via the Flash emulator module) or the MICROWIRE/PLUS™ support portion. The MICROWIRE/PLUS support portion is fully documented and its requirements are specified.

2 Introduction To ISP—Software Topics

The COP8TAB9/TAC9 Flash Family provides the capability to program the Program Memory while installed in an application board. This feature is called In System Programming (ISP). It provides a means of ISP by using the MICROWIRE/PLUS, or the user can provide his own, customized ISP routine. This customized routine may use any of the capabilities of the device, such as parallel port, and so on. The factory installed ISP uses only the MICROWIRE/PLUS port.

2.1 Functional Description

The organization of the ISP feature consists of the user flash program memory, the factory Boot ROM, and some registers dedicated to performing the ISP function. See Figure 1 for a simplified block diagram. The factory installed ISP that uses MICROWIRE/PLUS is located in the Boot ROM. The size of the Boot ROM is 1K bytes and also includes the ICE™ monitor code. If a user chooses to write his own ISP routine, it must be located in the flash program memory.

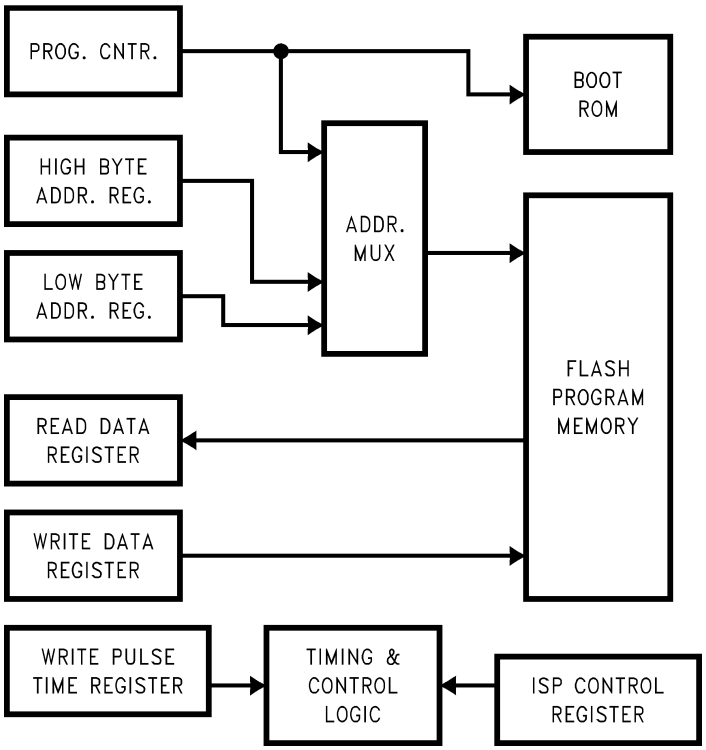

Figure 1. Block Diagram of ISP

In Section 3, a discussion regarding the FLEX bit is presented. The FLEX bit controls whether the device exits RESET executing from the flash memory or the Boot ROM. The user must program this Option Byte bit as appropriate for the application. In the erased state, the FLEX bit = 0 and the device will power-up executing from Boot ROM. When FLEX = 0, this assumes that either the MICROWIRE/PLUS ISP routine or external programming is being used to program the device. If using the MICROWIRE/PLUS ISP routine, the software in the Boot ROM will monitor the MICROWIRE/PLUS for commands to program the flash memory. When programming the flash program memory is complete, the FLEX bit will have to be programmed to a 1 and the device will have to be reset, either by pulling external Reset to ground or by software, before execution from flash program memory will occur.

If FLEX = 1, upon exiting Reset, the device will begin executing from location 0000 in the flash program memory. The assumption here, is that either the application is not using ISP, but is using MICROWIRE/PLUS ISP by jumping to it within the application code, or is using a customized ISP routine. If a customized ISP routine is being used, then it must be programmed into the flash memory by means of MICROWIRE/PLUS ISP or external programming as described in the preceding paragraph.

2.2 Registers

There are six registers required to support ISP: Address Register Hi byte (ISPADHI), Address Register Low byte (ISPADLO), Read Data Register (ISPRD), Write Data Register (ISPWR), Write Timing Register (PGMTIM), and the Control Register (ISPCNTRL).

2.2.1 ISP Address Registers

The address registers, ISPADHI (Table 1) and ISPADLO (Table 2), are used to specify the address of the byte of data being written or read. For page erase operations, the address of the beginning of the page should be loaded. When reading the Option register, 07FF (for COP8TAB9) or 0FFF (for COP8TAC9) should be placed into the address registers. Registers ISPADHI and ISPADLO are cleared to 00 on Reset. These registers can be loaded from either flash program memory or Boot ROM and must be maintained for the entire duration of the operation.

Table 1. High Byte of ISP Address

ISPADHI							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Addr 15	Addr 14	Addr 13	Addr 12	Addr 11	Addr 10	Addr 9	Addr 8

Table 2. Low Byte of ISP Address

ISPADLO							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Addr 7	Addr 6	Addr 5	Addr 4	Addr 3	Addr 2	Addr 1	Addr 0

2.2.2 ISP Read Data Register

The Read Data Register (ISPRD), Table 3, contains the value read back from a read operation. This register can be accessed from either flash program memory or Boot ROM. This register is undefined on Reset. This register is not directly available for external ISP access.

Table 3. ISP Read Data Register

ISPRD							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

2.2.3 ISP Write Data Register

The Write Data Register (ISPWR), Table 4, contains the value to be written to the Flash at the address specified in ISPADHI and ISPADLO. This register can be accessed from either flash program memory or Boot ROM. This register is undefined on Reset. This register is not directly available for external ISP access.

Table 4. ISP Write Data Register

ISPWR							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

2.2.4 ISP Write Timing Register

The Write Timing Register (PGMTIM) is used to control the width of the timing pulses for write and erase operations. The value to be written into this register is dependent on the frequency of CKI and is shown in Table 13. This register must be written before any write or erase operation can take place. It only needs to be loaded once, for each value of CKI frequency. This register can be loaded from either flash program memory or Boot ROM and must be maintained for the entire duration of the operation.

2.3 Forced Execution from Boot ROM

When the user is developing his own ISP routine, he may encounter code lockups due to mistakes in his software. There is a hardware method to get out of these lockups and force execution from the Boot ROM's MICROWIRE/PLUS routine, so that the customer can erase his flash code and start over. The method to shift a 24-code (0x5E38AC, LSB first) into the G0 pin, using G2 as a clock, while holding RESET low. This sequence is shown in Figure 2. This special condition will start execution from location 0000 in the Boot ROM where the user can input the appropriate commands, using MICROWIRE/PLUS, to erase the flash program memory and reprogram it.

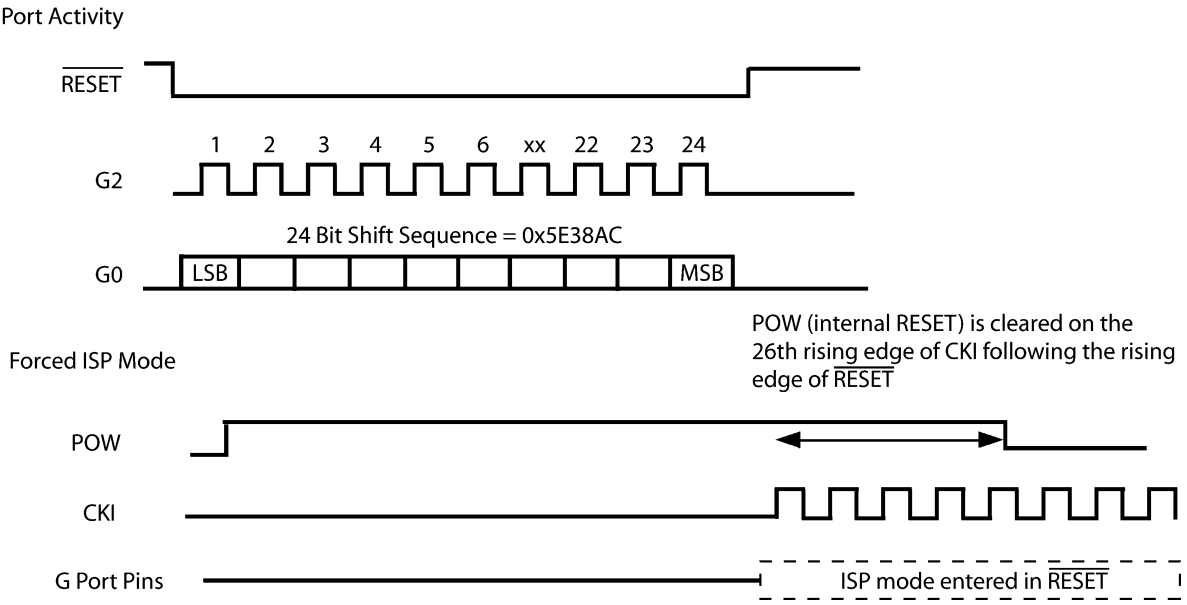

Figure 2. Timing of Shift Sequence for Entering Forced ISP Mode

2.4 MICROWIRE/PLUS ISP Commands

The MICROWIRE/PLUS ISP will support the following features and commands:

- Read a byte from a specified address.
- Write a byte from a specified address.
- Erase a page at a specified address.
- Erase the entire flash program memory (mass erase).
- Read multiple bytes starting at a specified address.
- Write multiple bytes starting at a specified address.
- Read Option register.
- Exit ISP by resetting the device and return execution to flash program memory if the FLEX bit is set in the Option Register.

3 Advanced ISP—Software Topics

3.1 In System Programming (ISP) Support Blocks

The COP8TAB9/TAC9's Boot ROM consists of three main blocks: The user support portion, the emulation support portion, and the MICROWIRE/PLUS support portion. Figure 3 shows the relative organization of these support blocks. Each command portion is both independent and self contained. The entire Boot ROM is 1 Kbytes. This document assumes that the reader is fluent in the use of MICROWIRE/PLUS and its transmission protocol. For reference please refer the MICROWIRE/PLUS section of the COP8TAB9/TAC9 datasheet.

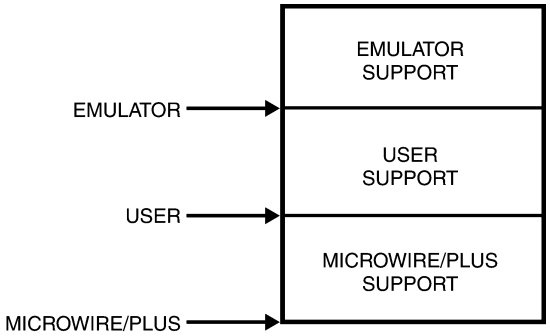

Figure 3. ISP Boot ROM Interface

3.1.1 Boot ROM Memory Layout

Figure 4 shows how the Boot ROM is organized. FLEX is a hardware bit that controls whether program execution occurs from flash memory of Boot_ROM. It uses data from the Option register. When the FLEX bit = 1, on exit from Reset. execution begins from the flash program memory. When the FLEX bit = 0, one exit from Reset. program execution begins from the Boot ROM.

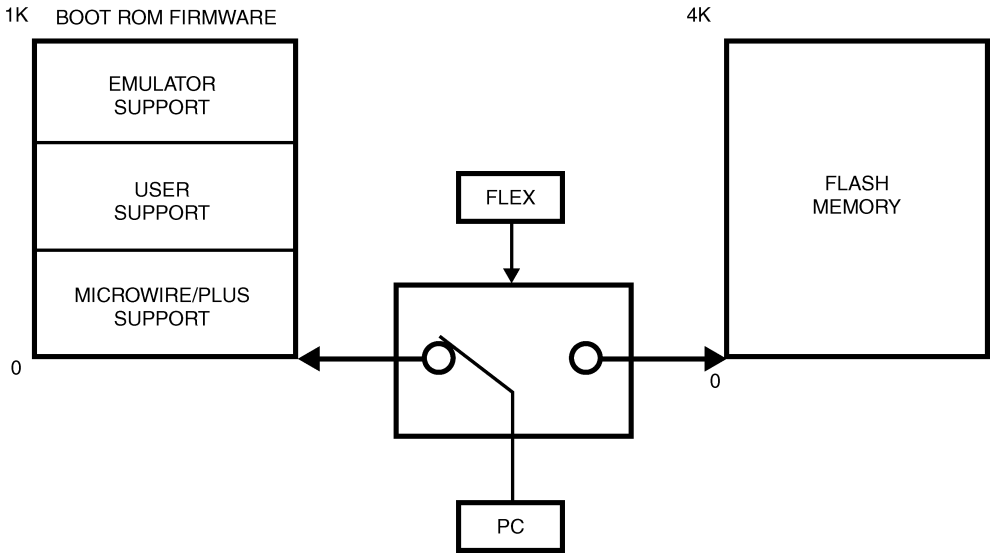

Figure 4. COP8 FLASH Memory Layout

3.2 Programmable Options Description

The programmable configuration options for this device are:

- Program Memory Security
- Oscillator selection
- Watchdog Feature Enable
- Halt Feature Enable
- Power-up execution selection

The options will be stored in the highest location in program memory. This location will be called the Option Byte. For devices with 4K of Program Memory, the options are stored at location 0FFF. For 2K devices, they will be stored at 07FF. The options are programmed with either external programming or ISP. The location must be erased before programming. The user must not store instructions in the Option register location. If the software tries to execute from the Option register, 00 data will be returned to the instruction register and the device will execute the Software Trap.

3.3 Option Register

The Option register, Table 5, located at address 0x0FFF (hex) in the Flash Program Memory, is used to configure the user selectable security, WATCHDOG, HALT and Oscillator selection options. The register can be programmed only in external Flash Memory programming or ISP Programming modes. Therefore, the register must be programmed at the same time as the program memory. The contents of the Option register shipped from the factory read 00 Hex.

Table 5. Option Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD	CLKSEL2	SEC	CLKSEL1	CLKSEL0	WATCHDOG	HALT	FLEX

Option Register Field Descriptions

Bit	Field	Description
7	RSVD	This bit is reserved and must be 0.
6	CLKSEL2	This bit defines the most significant bit of the oscillator selection. (See Table 6 for more information on Oscillator selection.)
5	SEC	0: Security disabled. Flash Memory read and write are allowed. 1: Security enabled. Flash Memory read and write are not allowed except in User ISP/Virtual E^2 commands. Mass Erase is allowed.
4-3	CLKSEL1, CLKSEL0	These bits define the two least significant bits of the oscillator selection. (See Table 6 for more information on Oscillator selection.)
2	WATCHDOG	0:WATCHDOG feature enabled. G1 pin is WATCHDOG output with weak pullup. 1: WATCHDOG feature disabled. G1 is a general purpose I/O.
1	HALT	0: HALT mode enabled. 1: HALT mode disabled.
0	FLEX	0:Flash Memory is erased. Execution following RESET will be from Boot ROM with the MICROWIRE/PLUS ISP routines. 1:Execution following RESET will be from Flash Memory.

Table 6. Oscillator Selection

Option Byte			Oscillator Selection
Bit 6	Bit 4	Bit 3	
0	0	0	Fully internal RC oscillator
0	0	1	RC oscillator with external frequency control resistor
0	1	0	Crystal oscillator with on-chip feedback resistor
0	1	1	Crystal oscillator with user-supplied feedback resistor
1	x	x	External oscillator

3.4 Security

The device has a security feature. When enabled, it prevents reading, writing, and page erases of the flash program memory by all external control sources. Bit-5 in the Option register determines whether security is enabled or disabled. If the security option is disabled, the content of the internal flash program memory is not protected. If the security feature is enabled;

When executing from user ISP:

1. Reads, writes, page erases, mass erases are all allowed. The user is expected to enforce security within the application code.

When executing from NSC (Boot ROM) ISP or ICE emulation. All writes, reads, and page erases are prohibited.

1. Reads will return FF.
2. Mass erase is permitted. This also erases the Option register.
3. The Option register is readable by reading location FFFF. (COP8TAC9 only)
4. Reads, writes, page erases are prohibited.

3.5 MICROWIRE/PLUS Support Blocks

3.5.1 Introduction

MICROWIRE/PLUS is a synchronous, SPI compatible, serial communication protocol that allows this device to communicate with any other device that also supports the MICROWIRE/PLUS system. Examples of such devices include A/D converters, comparators, EEPROMs, display drivers, telecommunications devices, and other processors. The MICROWIRE/PLUS serial interface uses a simple and economical 3-wire connection between devices.

Several MICROWIRE/PLUS devices can be connected to the same 3-wire system. One of these devices, operating in what is called the master mode, supplies the synchronous clock for the serial interface and initiates the data transfer. Other devices, operating in what is called the slave mode, respond by sending (or receiving) the requested data. The slave devices use the master's clock for serially shifting data out (or in), while the master device shifts the data in (or out).

On this device, the three interface signals are called SI (Serial Input), SO (Serial Output), and SK (Serial Clock). To the master, SO and SK are outputs (connected to slave inputs), and SI is an input (connected to slave outputs).

This device can operate either as a master or a slave, depending on how it is configured by the software. Figure 5 shows an example of how several devices can be connected together using the MICROWIRE/PLUS system, with the device (on the left) operating as the master, and other devices operating as slaves. The protocol for selecting and enabling slave devices is determined by the system designer.

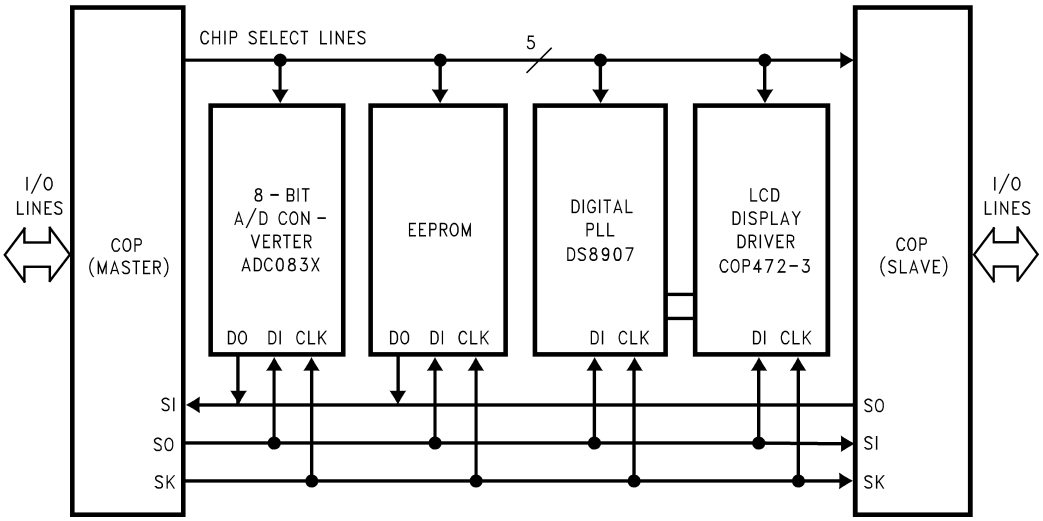

Figure 5. MICROWIRE/PLUS Example

3.5.2 Firmware—MICROWIRE/PLUS Initialization

The MICROWIRE/PLUS support block will initialize the internal communication block with the following parameters: CTRL.MSEL=1 (MICROWIRE/PLUS enabled), PORTGC.SK=0, PORTGD.SK=1 (Slave Mode with weak pullup on the clock), PORTGC.SO=1 (SO output enabled) and PORTGC.SI=1 (Alternate mode). Table 7 and Table 8 contain information about the MICROWIRE/PLUS mode. Figure 6 shows the waveforms that apply to the MICROWIRE/PLUS block.

Table 7. Initialization of the MICROWIRE/PLUS by the Firmware

Port G Config. Reg. Bits G5-G4	MICROWIRE/PLUS Operation	G4 Pin Function	G5 Pin Function	G6 Pin Function
0-1	Slave, Data Out and Data In	SO Output	SK Input	SI Input

Table 8. MICROWIRE/PLUS Mode Selected by the Firmware

Port G		SO Clocked Out On:	SI Sampled On:	SK Idle Phase
G6 (SKSEL) Config. Bit	G5 Data Bit			
1	1	SK Falling Edge	SK Rising Edge	High

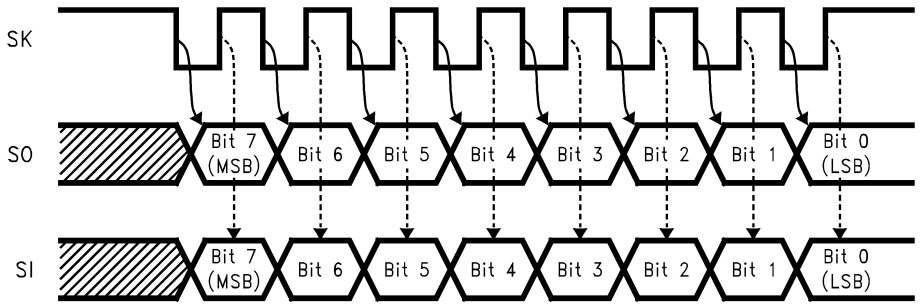

Figure 6. MICROWIRE/PLUS Interface Timing, Normal SK Mode, SK Idle Phase Being High

3.6 PC to Boot from MICROWIRE/PLUS Connection Diagram

Figure 7 shows the necessary connections to attach the MICROWIRE/PLUS to the PC's parallel port. The flash microcontroller connection to the PC will be accomplished via an eight wire interface.

Table 9 shows the necessary connections used in the building of the parallel adapter for the COP8TAB9/TAC9 Flash Family microcontroller.

NOTE: If the COP8–ISP-TAC-0 cable from Texas Instruments is used with any application other than, COP8–DB-TAC-0, resistor R10 must be provided on the application board.

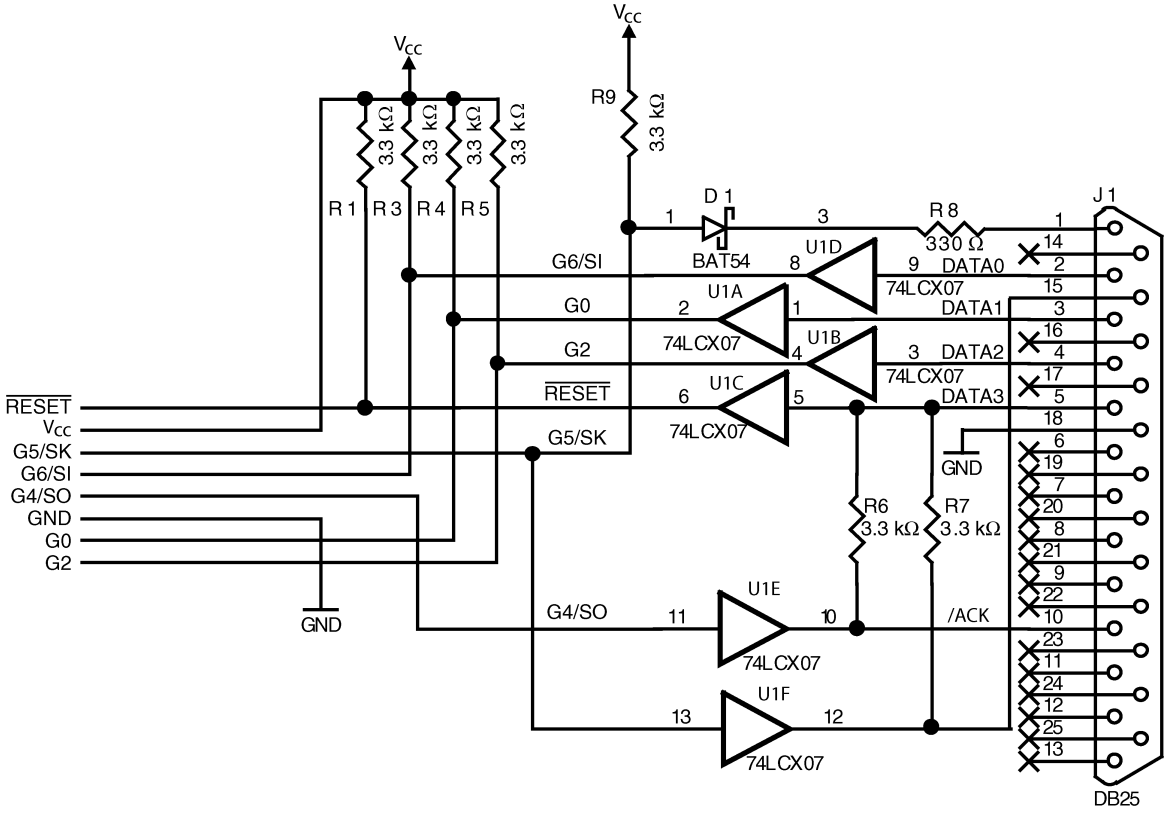

Figure 7. Parallel Port Connection Diagram

Table 9. Parallel Port <-> MICROWIRE/PLUS Conversion

Parallel Port Printer Port Pin Names	Parallel Printer Port Pin Numbers	MICROWIRE/PLUS Pin Names
STROBE	1	SK/G5
D0	2	SI/G6
D1	3	G0
D2	4	G2
D3	5	$\overline{\text{RESET}}$
NEG(ACK)	10	SO/G4
GND	18	GND
N/C	N/C	V_{CC}

3.7 Firmware—MICROWIRE/PLUS Operation

3.7.1 MICROWIRE/PLUS Packet Composition

A typical MICROWIRE/PLUS packet is composed of a three byte frame (although this varies with the chosen command). Figure 8 is a symbolic representation of the ISP-MICROWIRE/PLUS packet. A trigger byte is a value which will cause an ISP (In System Programming) command to be executed (for example, erase, read or write a byte of flash). The COMMAND Byte holds this trigger byte value. Refer to Table 12 for valid MICROWIRE/PLUS commands and their trigger byte values. Bytes ADDRESS_HI and ADDRESS_LO refer to the high and low bytes of the flash memory address that is to be operated upon. The symbol t_{delay} represents the delay that is required when sending the command, ADDRESS_HI and ADDRESS_LO bytes.

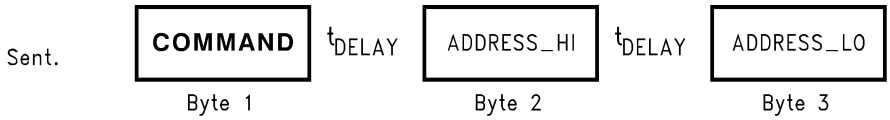

Figure 8. ISP Command Frame

3.7.2 Required Delays In Cascading Microwire Command Frames

A certain amount of delay must be observed when sending multiple command frames in a data stream. The symbol $t_{cascade\text{-}delay}$ represents the delay that is required when sending several commands in a data stream. The host must wait $t_{cascade\text{-}delay}$ cycles before sending the next command frame to the COP8 Flash Family device. Figure 9 shows the delay relationship. Refer to Table 10 for the values of $t_{cascade\text{-}delay}$. Refer to Table 11 for the values of t_{delay}. The symbol $t_1 ... t_N$ denotes individual delay requirements which vary among different commands and are described in detail in Section 3.8.

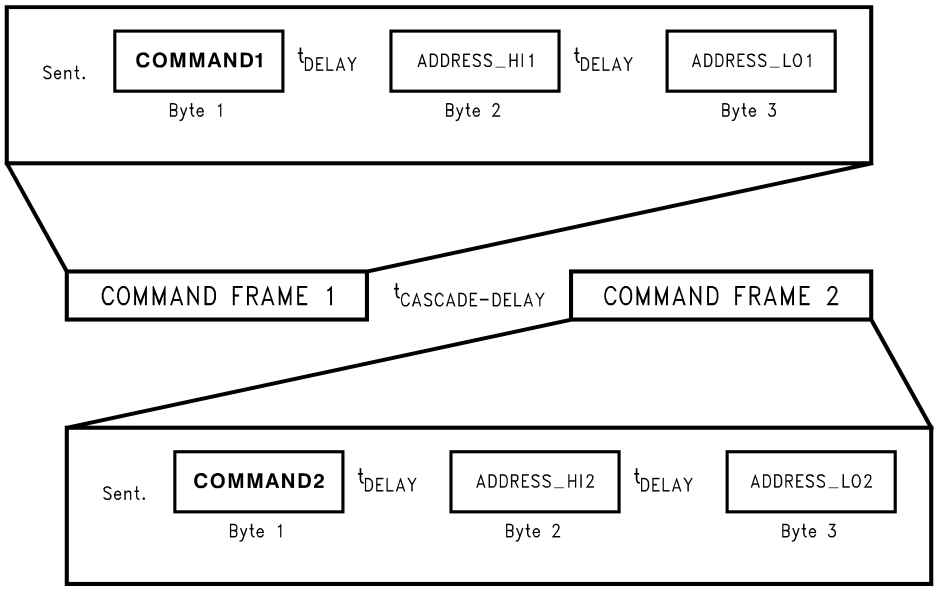

Figure 9. Cascade Delay Requirement

Table 10. Required Time Delays (in Instruction Cycles) for Cascading Command Frames After an Initial Command was Executed

Command	$t_{CASCADE-DELAY}$
READ_BYTE	48
WRITE_BYTE	34
BLOCKR	125
BLOCKW	34
PGERASE	34
MASS_ERASE	34
EXIT	N/A
PGMTIM_SET	51

Table 11. Required Time Delays (in Instruction Cycles)

Command	t_1	t_2	t_3	t_4	t_5	t_6	t_N
READ_BYTE	58	48	91	N/A	N/A	N/A	N/A
WRITE_BYTE	62	48	56	44	N/A	N/A	N/A
BLOCKR	70	48	56	48	97	162	162
BLOCKW	66	48	56	54	54	51	54
PGERASE	77	48	52	N/A	N/A	N/A	N/A
MASS_ERASE	73	41	N/A	N/A	N/A	N/A	N/A
EXIT	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PGMTIM_SET	66	N/A	N/A	N/A	N/A	N/A	N/A

3.7.3 Variable Host Delay

A special type of communication has been implemented in the device firmware in order to allow the microcontroller enough time to complete extended time operations such as write or erase. This type of communication was developed since the microcontroller may be used in situations where the clock is extremely slow and writes to the flash memory will take a large amount of time. This implementation relieves the user of having to manually change the write delays in host software. Figure 10 shows how the VARIABLE HOST DELAY configuration is implemented on a byte write. Figure 11 shows how the VARIABLE HOST DELAY configuration is implemented on a block write. Figure 12 shows how the VARIABLE HOST DELAY configuration is implemented on a page erase. Figure 13 shows how the VARIABLE HOST DELAY configuration is implemented on a mass erase. Since the SK (Serial CLOCK) is normally high, the microcontroller brings SK low to indicate to the host that a WAIT condition (that is, the SK pin is low) exists. The host then goes into a loop until the WAIT condition changes to a READY condition (that is, the SK pin is high again). The controller then returns to command decode and waits for the next command.

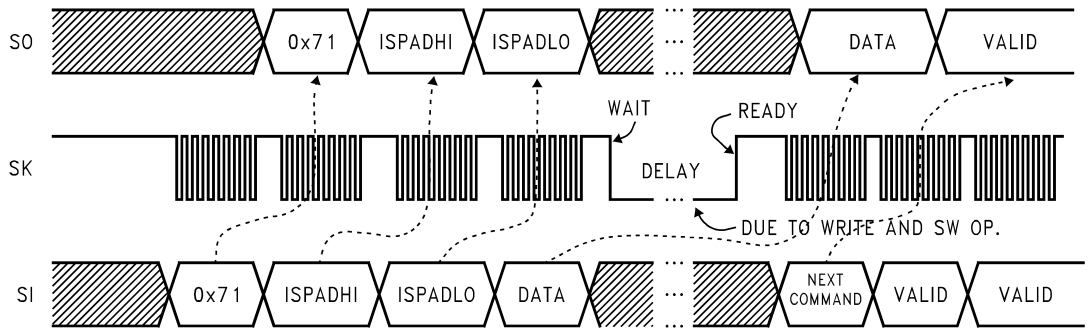

Figure 10. Byte Write Waveform (Relative Bytes are Shown)

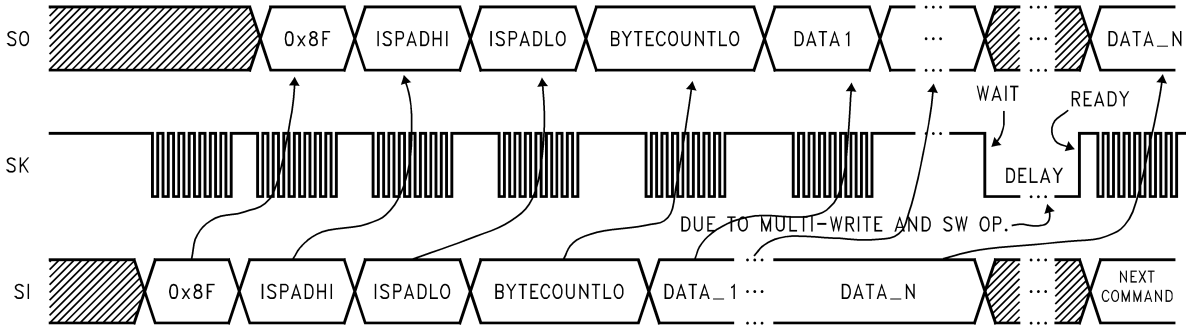

Figure 11. Block Write Waveform (Relative Bytes are Shown)

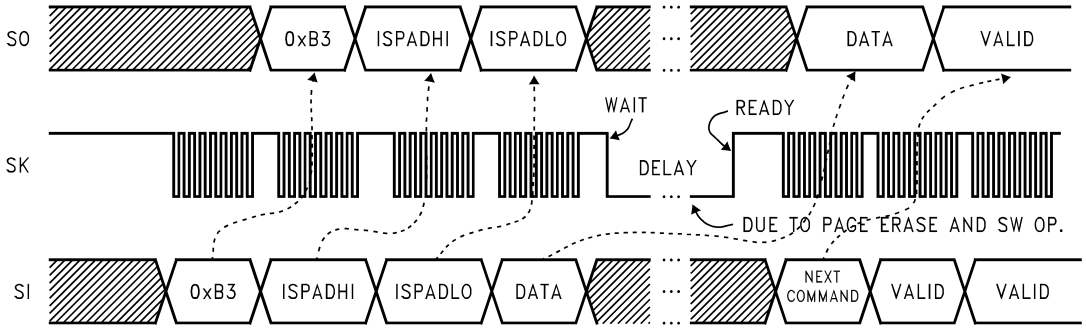

Figure 12. Page Erase Waveform (Relative Bytes are Shown)

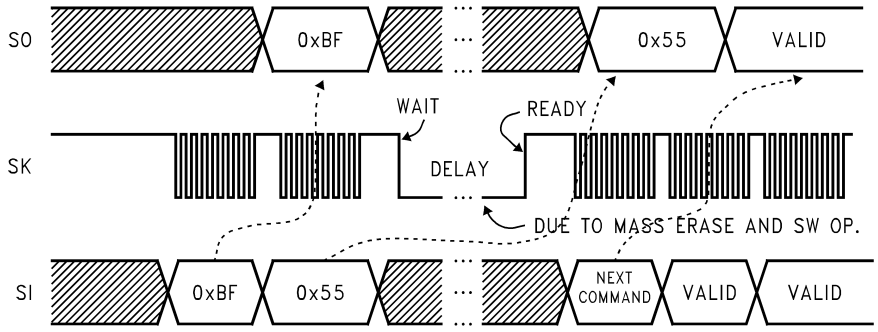

Figure 13. Mass Erase Waveform (Relative Bytes are Shown)

3.7.4 MICROWIRE/PLUS—Boot ROM Startup Behavior

Upon start-up, the ISP Boot ROM will detect if the G6 pin is high. By using this technique the Boot ROM avoids any bit that may be inadvertently entered on to the SI pin. If the G6 pin is not high at start-up, the ISP Boot ROM will try to detect if a valid command is received on a transmission. If a valid command is received, the Boot ROM firmware will check to see if the SECURITY bit is set. Table 12 shows the valid MICROWIRE/PLUS commands. If security is set, the Boot ROM will disable all ISP functions except reading the OPTION register at 0xFFFF (COP8TAC9 only), the execution of a mass erase on the flash memory and setting the PGMTIM Register. Read attempts of flash memory, other than location 0xFFFF, Option Register of COP8TAC9, while security is set, will result with a 0xFF sent back through the MICROWIRE/PLUS. In general, the Boot ROM firmware will decode the command, check security, execute the command (if security is off) and execute the MICROWIRE/PLUS Main Support Block (for example, triggering the PSW.BUSY bit in order to send the data back to the host.) See Figure 14 for the ISP—MICROWIRE/PLUS Control flow.

Table 12. MICROWIRE/PLUS Commands

Command	Function	Byte Value	Parameters	Variable Host Delay Implemented?	Return Data
PGMTIM_SET	Write Pulse Timing Register	0x3B	Value	No	N/A
PAGE_ERASE	Page Erase	0xB3	Starting Address of Page	Yes	N/A
MASS_ERASE	Mass Erase	0xBF	Confirmation Code	Yes	N/A (The entire Flash Memory will be erased)
READ_BYTE	Read Byte	0x1D	Address High, Address Low	No	Data Byte if Security not set. 0xFF if Security set.
BLOCKR	Block Read	0xA3	Address High, Address Low, Byte Count (n) High, Byte Count (n) Low (0 ≤ n ≤ 32767)	No	n Data Bytes if Security not set. n Bytes of 0xFF if Security set
WRITE_BYTE	Write Byte	0x71	Address High, Address Low, Data Byte	Yes	N/A
BLOCKW	Block Write	0x8F	Address High, Address Low, Byte Count (0 ≤ n ≤ 16), n Data Bytes Data location must be within a 64 byte segment (1/8 page) due to multi-byte write limitation	Yes	N/A
EXIT	EXIT	0xD3	N/A	No	N/A (Device will Reset)
INVALID	N/A		Any other invalid command will be ignored	N/A	N/A

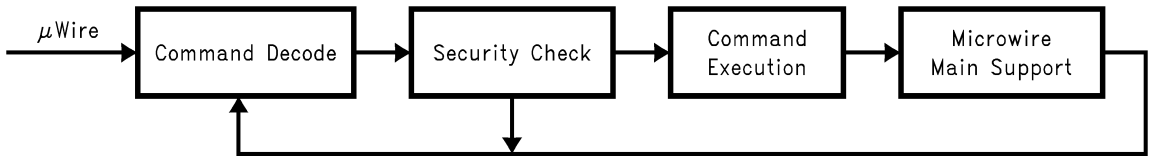

Figure 14. ISP—MICROWIRE Control

3.8 MICROWIRE Commands Available

3.8.1 PGMTIM_Set

Sets the flash write timing register to match that of the CKI frequency.

Figure 15 shows the format of the PGMTIM_SET command. The PGMTIM_SET command will transfer the next byte sent into the flash programming time register. No acknowledgment will be sent. The symbol t_1 denotes the time delay between the command byte and the setting of the PGMTIM register. This command is always available. This command must be used before any “writes” or "erases" can occur (that is, page erase, mass erase, write byte or block write). See Table 13 for the value(s) of t_1 and t_2. Table 13 shows valid values for the PGMTIM register. This command is security independent.

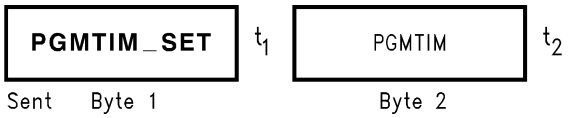

Figure 15. Set PGMTIM Command

Table 13. Valid PGMTIM Values

Bit Values for the PGMTIM Register								Hex Value	CKI Frequency Range
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	0	0x00	25 kHz–50 kHz
0	0	0	0	0	0	0	1	0x01	50 kHz–100 kHz
0	0	0	0	0	0	1	0	0x02	75 kHz–150 kHz
0	0	0	0	0	1	0	0	0x04	125 kHz–250 kHz
0	0	0	0	0	1	1	1	0x07	200 kHz–400 kHz
0	0	0	0	1	0	1	1	0x0B	300 kHz–600 kHz
0	0	0	1	0	0	0	1	0x11	450 kHz–900 kHz
0	0	0	1	0	1	1	1	0x17	600 kHz–1.2 MHz
0	0	1	0	0	1	1	1	0x27	1.0 MHz–2.0 MHz
0	0	1	1	1	1	1	1	0x3F	1.6 MHz–3.2 MHz
0	1	0	0	1	0	1	0	0x4A	2.75 MHz–5.5 MHz
0	1	0	0	1	1	1	0	0x4E	3.75 MHz–7.5 MHz
0	1	0	1	0	1	0	1	0x55	5.5 MHz–11 MHz
0	1	0	1	1	0	1	0	0x5A	6.75 MHz–13 MHz
0	1	0	1	1	1	0	1	0x5D	7.5 MHz–15 MHz
0	1	1	0	1	1	0	0	0x6C	11.25 MHz–22.5 MHz
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

3.8.2 PAGE_ERASE—Erase a Page of Flash Memory

Figure 16 shows the format of the PAGE_ERASE command. The PAGE_ERASE command will erase a 512 byte page of the flash memory. The next two bytes after the PAGE_ERASE byte refer to the beginning high and low bytes of the beginning address of the target flash page. A WAIT/READY (Variable Host Delay) technique is used to delay the host when the controller is executing and erasing the flash memory. For a full description of the WAIT/READY command, refer to Section 3.7.3. The symbol t_1, t_2 denote the time delay between the command byte, the delay required after loading the high address byte, and the delay after loading the low address byte. The symbol t_3 denotes the time delay after loading the ADDRESS_LO value. The PAGE_ERASE command is NOT always available (that is, it is security dependent). If security is set, then the command will be aborted and no acknowledgment will be sent. See Table 11 for the value(s) of t_1, t_2, and t_3.

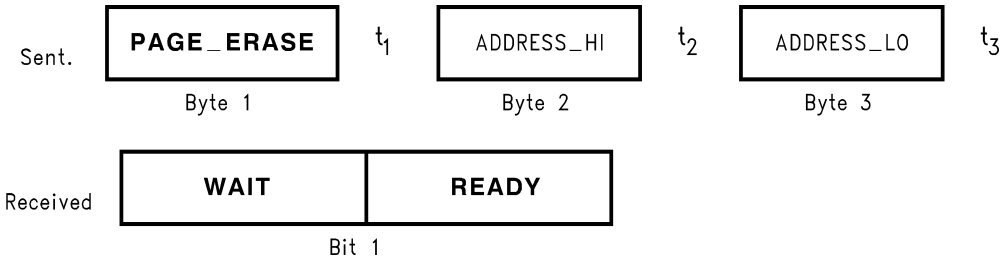

Figure 16. PAGE ERASE Command

3.8.3 MASS_ERASE—Erase the Entire Flash Memory Array

Figure 17 shows the format of the MASS_ERASE command. The MASS_ERASE command will erase the entire flash memory, including the Option Register. The next byte after the MASS_ERASE command refers to the confirmation key used to double check that a mass erase request was actually sent. The confirmation key must equal 0x55 in order for the MASS_ERASE command to continue. The symbol t_1 denotes the time delay between the command byte and the transmission of the CONFIRM_KEY. The symbol t_2 denotes the time delay after the CONFIRM_KEY has been checked. A WAIT/READY technique is used to delay the host when the controller is executing and writing to the flash memory. For a full description regarding the WAIT/READY command, refer to Section 3.7.3. The MASS_ERASE command is always available. It is security independent. See Table 11 for the value(s) of t_1, and t_2.

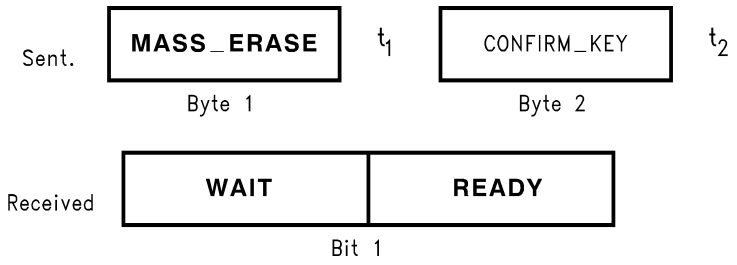

Figure 17. MASS_ERASE Command

3.8.4 READ_BYTE—Read a Byte from the Flash Memory Array

Figure 18 shows the format of the READ_BYTE command. The READ_BYTE command will read a byte from the flash memory. The next two bytes after the READ_BYTE refer to the address of the target flash location. The symbol t_1, t_2 denotes the time delay between the command byte, the delay after loading of the high address byte. Data is sent back after t_3 delay(s) has elapsed. If security is set, the user is only allowed to read location 0xFFFF (Option Register of COP8TAC9). In other words, if security is set and ADDRESS_HI and ADDRESS_LO=0xFFFF then the firmware will allow that operation, otherwise it will send back a 0xFF in the DATA_RTN byte. See Table 11 for the value(s) of t_1, t_2, and t_3.

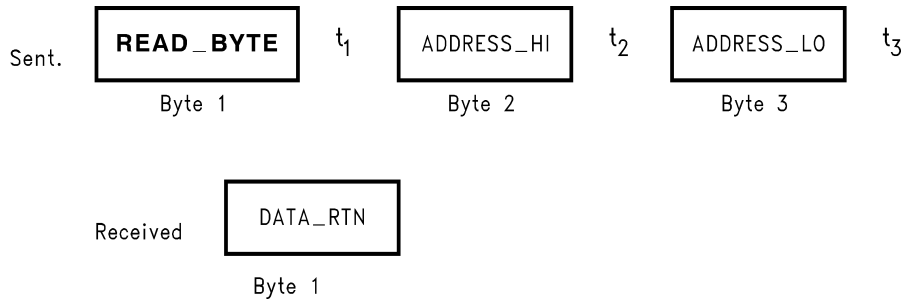

Figure 18. READ_BYTE Command

3.8.5 WRITE_BYTE—Write a Byte to the Flash Memory Array

Figure 19 shows the format of the WRITE_BYTE routine. The WRITE_BYTE command will write a byte to the flash memory. The next two bytes after the WRITE_BYTE byte refer to the high and low byte address of the target flash location. The next byte (DATA_REC) after the ADDRESS_LO byte will contain the value that will be stored into the flash location. The symbols t_1, t_2 denote the time delay between the command byte and the delay after loading of the high address byte. The symbol t_3 denotes the time delay after loading the ADDRESS_LO value. Data is saved into the flash location after a t_4 delay. A WAIT/READY signal is used to delay the host. For a full description of the WAIT/READY command, refer to Section 3.7.3. The WRITE_BYTE command is NOT always available (that is, it is security dependent.) If security is set, then the command will be aborted and no acknowledgment will be sent back. See Table 11 for the value(s) of t_1, t_2, t_3, and t_4.

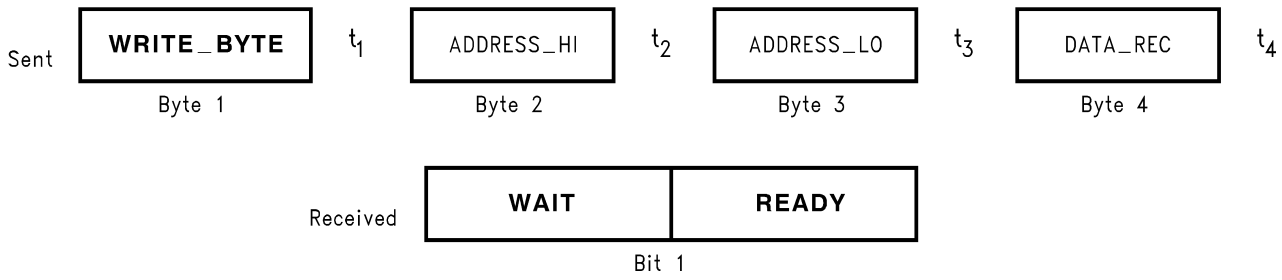

Figure 19. WRITE_BYTE Command

3.8.6 BLOCK WRITE—Write a Block of Data to the Flash Memory Array

Figure 20 is a symbolic representation of the BLOCK_WRITE routine. Data is written in sequential order. This routine is intended to write bytes of data which will reside in a page of flash memory. The next two bytes after the BLOCK_WRITE byte refer to the beginning high and low byte address of the target flash location. The next byte after the ADDRESS_LO byte refers to the BYTECOUNTLO variable. The BYTECOUNTLO variable is used by the microcontroller to transfer N bytes (that is, N=BYTECOUNTLO). The maximum number of bytes that can be written is 16. If the number of bytes exceeds 16, it may not be guaranteed that all of the bytes were written. Block Writes cannot cross row boundaries. Data must be placed within the same 1/8 page segment, 64 bytes. If N=0 then the firmware will abort. The symbols t_1 and t_2 denotes the time delay between the command byte and the delay after loading of the high address byte. The symbol t_3 denotes the time delay after loading the ADDRESS_LO value. The symbol t_4 denotes the necessary time delay after loading the BYTECOUNTLO variable. Data arrives at t_5 cycles after the ADDRESS_LO value is loaded (that is, DATA1 - DATA2 have the same delay as DATA2 - DATA3). After the last byte (DATA_N) is received, a WAIT/ READY signal will be sent to delay the host. For a full description of the WAIT/READY command, refer to Section 3.7.3. The command (BLOCK_WRITE) is NOT always available (that is, it is security dependent). If security is set, then the command will be aborted after the last data (DATA_N) is received and no acknowledgment will be sent back. See Table 11 for the value(s) of t_1, t_2, t_3, t_4, t_5, and t_6.

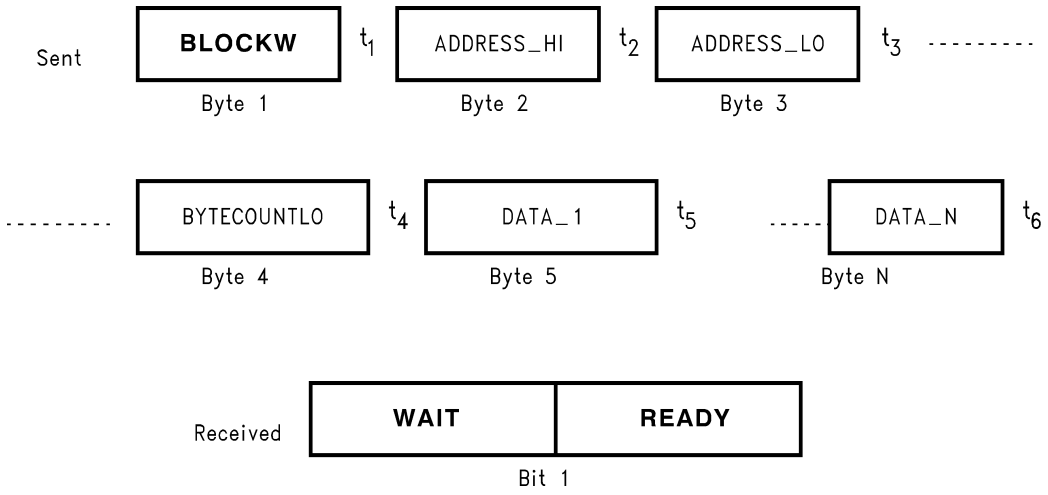

Figure 20. Block Write Routine

3.8.7 BLOCK_READ—Read a Block from the Flash Memory Array

Figure 21 shows the format of the BLOCK_READ command. The BLOCK_READ command will read multiple bytes from the flash memory. The next two bytes after the BLOCK_READ byte refer to the beginning high and low byte address of the target flash location. The next two bytes after the ADDRESS_LO byte refer to the upper and lower byte of BYTECOUNT. The BYTECOUNT variable is used by the microcontroller to send back N number of bytes (that is, N=BYTECOUNT). The maximum value of N is 4 kBytes. If N=0 then the firmware will abort. The symbols t_1, $t_2$and , t_3 denotes the time delay between the command byte, the delay in loading of the ADDRESS_HI, and the delay after loading the ADDRESS_LO. The symbol t_4 denotes the required time delay between loading BYTECOUNTHI and BYTECOUNTLO. Subsequent data is sent to the host at t_5 cycles after BYTECOUNTLO (that is, DATA1–DATA2 have the same delay as DATA2–DATA3). This command is capable of sending up to 32 kB of flash memory through the MICROWIRE/PLUS. This command is always available however, if security is set, the user is only allowed to read 0xFFFF (Option Register of COP8TAC9). In other words, if at anytime ADDRESS_HI and ADDRESS_LO=0xFFFF, the firmware will allow that operation. If at any time ADDRESS_HI and ADDRESS_LO do not equal 0xFFFF and security is set, then the firmware will return 0xFF. This routine will acknowledge by returning data to the host.

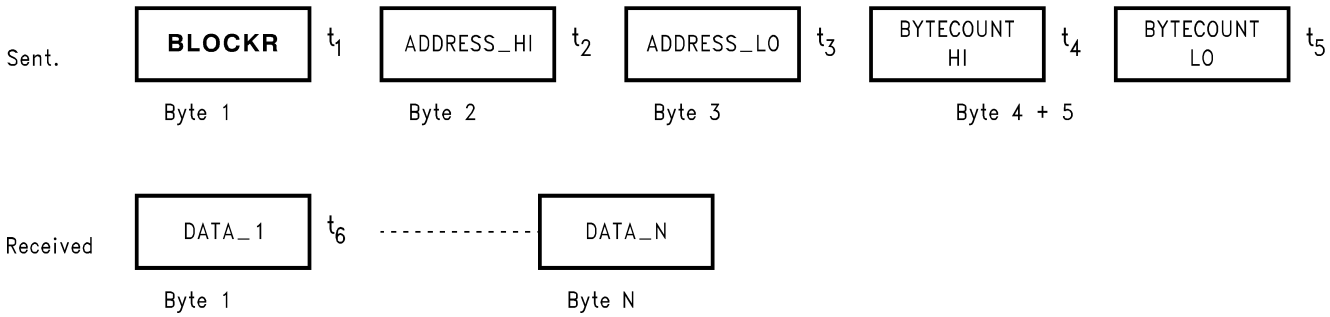

Figure 21. Block Read Command

3.8.8 EXIT—Reset the Microcontroller

Figure 22 shows the format of the EXIT command. The EXIT command will reset the microcontroller. There is no additional information required after the EXIT byte is received. No acknowledgment will be sent back regarding the operation. This command is always available. It is security independent.

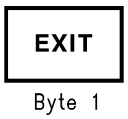

Figure 22. EXIT Command

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265