PIC32CX-BZ2

PIC32CX-BZ2 Programming Specification

Introduction

This programming specification applies to the PIC32CX-BZ2 developers of external programming tools.

In the development of a programming tool, it is necessary to understand the internal Flash program operations of the target device and the Special Function Registers (SFRs) that control the Flash programming, as an external programming tool and its software use the same operations and registers. For more details on these operations and control registers, see *Flash Program Memory* in the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504). It is highly recommended that the device-specific documents be used in conjunction with this programming specification.

An external tool for programming setup consists of an external programmer tool and a target PIC32CX-BZ2 device. The programmer tool is responsible for executing the necessary programming steps and completing the programming operation by sending Device Service Unit (DSU) commands over the Serial Wire Debug (SWD) interface of the PIC32CX-BZ2.

Table of Contents

1. Quick References

1.1 Reference Documentation

The following documents are referenced in this document for general understanding or detailed information.

- *ARMv7-M Architecture Reference Manual* – ARM DDI 0403E.c
- *Debug Interface v5 Architecture Specification* – IHI0031A or higher
- *Debug Interface v5 Architecture Specification ADIv5.1 Supplement* – DSA09-PRDC-008772 1.0
- *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504)

ROM Tables Handling

- *ARM® CoreSight™ Architecture Specification v2.0* – ARM IHI 0029D

ETM/ETB Trace

- *CoreSight™ ETM™ – M4 Technical Reference Manual* – ARM DDI 0440C

2. Overview

2.1 Architecture

The PIC32CX-BZ2 is based on an ARM Cortex M4F device (SAMD5x/SAME5x) with a CPU clock running up to 64 MHz. It includes other macros and peripherals from Microchip 'PIC' microcontrollers, including the Flash Controller (FC), which abstracts the flash topology for programming the single internal 1 MB embedded flash panel. The Device Service Unit (DSU) is the programming interface with ARM's 2-wire SWD interface.

Packages:

PIC32CX-BZ2 offers two variants:

- 39-pin LGA
- 48-pin QFN

For more package and pinout information, see the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504).

2.2 Programming Interface

This device uses the 2-wire Serial Wire Debug (SWD) for all user debug, programming and emulation features. For more information on the SWD interface, refer to the *ARM Core-sight Documentation*.

2.3 Memory

SRAM memories are accessed at their address in a memory space for both reads and writes at any granularity (byte, half word or word). The SRAM is 128 KB in size.

Flash memories are accessed for read at their address in memory space. Read granularity is byte, half word or word. FC erases and writes the Flash memories. The Flash memory is 1 MB in size and supports quad-word read (128-bit) and single word write (32-bit). The page size is 4 KB with 256 words per row and four rows per page.

The FC supports panel erase for the entire flash, or page erase for the unprotected pages. It supports Single Word Program (32-bit), Quad Word Program (128-bit), and row programming with a built-in DMA for reading data from the SRAM with a four- or eight-word buffer.

Additionally, there is a 32 KB Flash NVR block separate from the Main Panel block with eight pages (0 to 7) where:

- One page is allocated to Factory Test Memory
- One page is allocated to Cal Space (called OTP page)
- One page is allocated to Boot/Device Configuration (BCFG)
- Five pages are allocated to Boot Code

Each NVR page has write-protect capability.

The device memory supports a partial write feature, but using partial write-only bits that are '1' can be set to '0'. The user can erase the entire flash memory using a dedicated FC command.

2.4 Boot, BCFG and OTP Pages

Programming tools support reading and writing of NVR pages as they contain critical parameters for the device to run in the user-defined conditions, as well as to boot code.

The NVR pages start at address 0x0000_0000 and end at 0x0000_6FFF. For more details on each NVR page content, see the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504).

One page in the Flash boot region is implemented as a one-time programmable called OTP. The user can only write, and cannot erase by page or chip erase (only by a bulk erase) in the OTP region. This region stores the user system calibration data that must survive a flash erase. The user can use NVMOTP to store and preserve identification

values, such as, an Ethernet MAC address, OEM/ODM version numbers, board version numbers and so on. The other uses are for calibration values of circuits external to the chip but constant for a board design in which the chip exists.

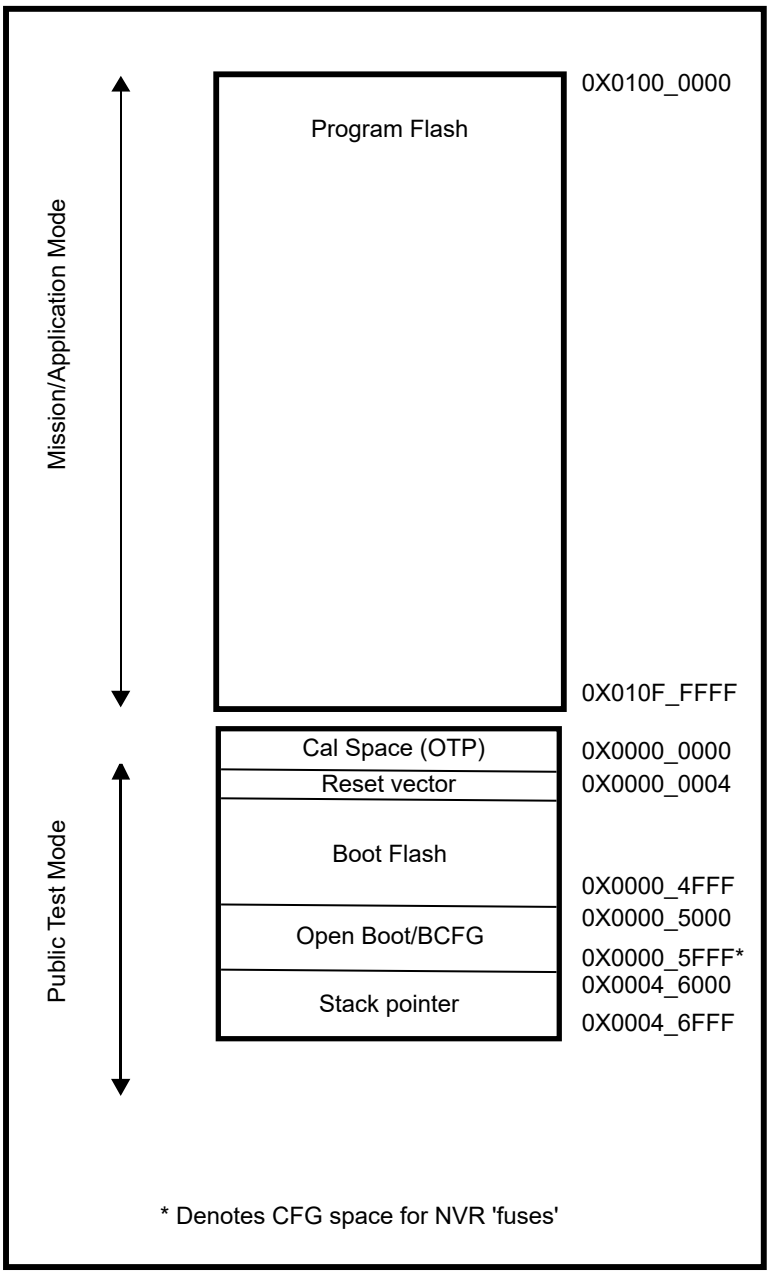
One page of the Flash boot region is called BCFG or open boot. This region stores the configuration bits. These bits are set with #pragma directives embedded in the source file(s), or the user can program them with external IDEs, such as, Microchip MPLAB® X, using the configuration bits window. In either case, the user must erase the page first before writing new configuration bits using the page erase function described in 6.1.8.1. Page Erase.

There is no difference in the programming algorithm for the Boot or NVR Flash pages from programming the 1 MB Flash panel pages except that the DSU.TESTMODE[3] (0x410000FC bit 3) must be set to '1'. QuadWord write programs the NVR FBCFGx 'fuses'. For more details, see 6.1.6. NVR Fuses. The user needs to manipulate the differences in memory protection registers. For more details, see 6. Programming.

2.5 Memory Map

The programming tools used refer to the `CMSIS *.SVD` files provided for each part or device data sheet. The CMSIS file contains the most up-to-date information on the device, memory and peripherals with its associated registers. The following figure illustrates the current flash memory map for the device.

Figure 2-1. Flash Memory Map



Boot Flash is where the CPU starts after a reset. This space contains the reset handler and interrupt vectors. The Boot Flash also contains the Microchip private boot code and user boot code (if any). After the boot code runs successfully, it jumps to the user Program Flash. For more details on the NVR memory, see the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504).

3. Device Reset

This section describes the device's available reset sources. For more detailed information, see the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504).

3.1 System Reset

The SWRST bit in the CRU.RSWRST register resets the software. The unlock sequence is needed to set this software reset bit. The address of this register is 0x4400_0a40. A write of logic '1' to this bit enables the software reset. A subsequent read of this register triggers the system reset sequence. The bit can be written once the system unlock sequence is complete. This bit always reads a value of logic '0'.

3.2 Vector Catch

The vector catch feature allows the CPU to halt after a reset or other exception. For more details, see the *ARMv7-M Architecture Reference Manual*.

3.3 External Reset

The debugger can also set the external reset device pin to low to reset the device. Set the $\overline{\text{MCLR}}$ pin of the device to a logic low level for 2 µS minimum pulse width.

Note: Do not set the SWDIO pins low using the debugger when the $\overline{\text{MCLR}}$ pin is high because it triggers a "CPU Reset Extension" (see 3.4. CPU Reset Extension).

3.4 CPU Reset Extension

All the devices with a DSU have the CPU reset extension feature that allows a connection to a debugger while the CPU is under reset. This is also known as cold-plugging or also core-hold reset. For more details, see the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504).

This feature requires the debug tool to control the level of SWCLK and $\overline{\text{MCLR}}$ pins.

This feature mainly avoids the execution of potentially corrupted code at boot and allows a connection to a device safely without any assumption on its current state. The debugger can read or write memory and peripherals without any delay from the CPU.

Note: After the initial connect sequence, the debugger must release the CPU reset extension to write to memory located outside the CPU (in other words, RAM, NVM and Peripherals) using Port 0 of the two Memory Access Port (MEM-AP). For more details, see 5. Access Port(s). The debugger can use the reset vector catch feature to prevent the CPU from executing the user's code.

3.5 CPU Boot

Boot memory starts at address 0x0000_0000 (this is the stack pointer) followed by 0x0000_0004 (this is the reset vector). For more details, see 2.5. Memory Map.

On a system reset, the vector table is fixed at address 0x00000000. Software can write to the Vector Table Offset Register (VTOR) to relocate the vector table's start address to a different memory location.

4. Serial Wire Debug Port

This section describes accessing the Serial Wire Debug Port (SW-DP). The SW-DP uses the following:

- SWD version 1 protocol. For more details, see *DSA09-PRDC-008772*.
- *ARM® Debug Interface v5 Architecture Specification* (ADIv5 revision)
- For more details on the serial wire debug protocol, see the *ARM® Debug Interface v5 Architecture Specification* (ARM IHI 0031 Revision A or higher).

4.1 Operation Sequence

4.1.1 Initialize SWD

There are no additional steps apart from what is described in the *ARM IHI 0031*. The SW-DP starts in the SWD mode and does not require a switching sequence.

4.1.2 Uninitialize

There are no additional steps apart from what is described in the *ARM IHI 0031*.

5. Access Port(s)

This section describes the Memory Access Port (MEM-AP) used to access the device memory. The debugger accesses the MEM-AP once access to Serial Wire Debug Port (SW-DP) is enabled. When the device security bit is cleared, 2 MEM-AP (AP0 and AP1) are accessible on those devices. The second AP may be used for non-intrusive debugging.

Even if it is not mandatory for this device, before accessing a MEM-AP, the debugger can wait for the CSW.DeviceEn bit to be set. A coresight ROM-table is attached to each AP, describing the available debug components. Alternatively, the debugger can rely on the DSU device ID to determine which debug components are available on the device.

5.1 Device Protected State

When the device is locked (the security bit is set), the MEM-AP access range is limited to the DSU's external range. For more details, see *Intellectual Property Protection* in the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504). It is recommended that the debugger first read the DSUEXT.STATUSB.PROT bit to handle this case. For more details on unprotecting the device, see 6.1.9. Chip Erase.

Note: The DSU external range is denoted as **DSUEXT** in this document.

5.2 Access Primitives

Based on the MEM-AP access, the host can rely on the following primitives:

Table 5-1. Access Primitives

Memory Access Primitive	Description
Status ReadD8(U32 Addr, Value)	Read a byte (8 bits) from device memory at address Addr Returns OK, FAULT, WAIT or ERROR
Status ReadD16(U32 Addr,Value)	Read a half-word (16 bits) from device memory at address Addr Returns OK, FAULT, WAIT or ERROR
Status ReadD32(U32 Addr,Value)	Read a word (32 bits) from device memory at address Addr Returns OK, FAULT, WAIT or ERROR
Status WriteD8(U32 Addr,Value)	Write a byte (8 bits) from device memory at address Addr Returns OK, FAULT, WAIT or ERROR
Status WriteD16(U32 Addr,Value)	Write a half-word (16 bits) from device memory at address Addr Returns OK, FAULT, WAIT or ERROR
Status WriteD32(U32 Addr,Value)	Write a word (32 bits) from device memory at address Addr Returns OK, FAULT, WAIT or ERROR

Note: The user can use the speed optimized primitives for block access using the auto-increment feature of the MEM-AP, but they are not mandatory to support all the device features.

6. Programming

This section describes the MEM-AP sequences required to program the areas of memory using the development tool.

Use the following procedure to program a device Flash using the DSU. Programming the Flash is only possible when the device is not code protected. If the device is already code protected, perform the chip erase procedure to make the device unprotected and programming ready.

1. Perform the cold plugging procedure to enter into the Debug Access Port (DAP) mode.
2. Perform a chip erase procedure (only if device is code protected).
3. Perform the system unlock sequence and write to the NVMLBWP register (0x440006F0) = 0x80000000. This turns off write protect to the boot flash.
   Note: At reset, all LBWPn and UBWPn bits are set to logic '1', write protecting all user accessible NVR pages. Clearing a boot page bit within its register removes write protect from the corresponding page.
4. Perform the procedures described in 6.1.7. Programming Sequences.
   If an attempt is made to program or erase a protected page or region, the FSM executes the sequence but it has no effect. Then, reset the write protection bits for the NVR flash by writing 0x80FFFFFF to the NVMLBWP register. Bulk or chip erase commands override the effect of the NVM*WP registers.

   Note: The PIC32CX-BZ2 devices have a single panel of Flash, therefore, the UBWPUNLK register has no functionality.

6.1 Operation Sequences

This section describes the MEM-AP sequences required to:

- Get Device ID
- Read memory (for each area of memory; requires programming operations to access)
- Write memory (for each area of memory; requires programming operations to access)
- Read Debug Halting Control and Status Register (DHCRS)

6.1.1 Get Device ID

The code is equivalent to reading a word at the address DSUEXT.DID (0x41000118) (DSU_DID register). The value is read from the DSUEXT range. This register is always accessible, even on protected devices.

Notes:

1. It is recommended that only the lower 16 bits of this register be used to identify the device.
2. The DSUEXT range starts at offset 0x0100. There are no restrictions when accessing DSU registers from the CPU. However, it is recommended that the user code running on the CPU access DSU registers in the internal address range only (0x0-0xFF).

Table 6-1. Reference

Step	mem_ap primitive
Read DSU DID register	ReadD32(DSUEXT.DID)

6.1.2 Read RAM and Peripherals

Reading from RAM or peripherals requires no specific handling.

Note: MEM-AP transactions may be handled differently than CPU transactions to provide "non-intrusive" debugging.

Table 6-2. Reference

Step	mem_ap primitive
Read 8-bit item from address	ReadD8(Addr,Value)

.....continued	
Step	**mem_ap primitive**
Read 16-bit item from address	ReadD16(Addr,Value)
Read 32-bit item from address	ReadD32(Addr,Value)

6.1.3 Write SRAM and Peripherals

Writing to SRAM or peripherals requires no specific handling.

See **Note** in 3.4. CPU Reset Extension .

Table 6-3. Reference

Step	mem_ap primitive
Write 8 bits item from address	WriteD8(Addr,Value)
Write 16 bits item from address	WriteD16(Addr,Value)
Write 32 bits item from address	WriteD32(Addr,Value)

6.1.4 Read Debug Halting Control and Status Register

Reading peripherals requires no specific handling. The code is equivalent to reading a word at the address 0xE000EDF0 (System Control Block DHCSR register).

Table 6-4. Reference

Step	mem_ap primitive
Read 32-bit item from address	ReadD32(Addr)

6.1.5 Flash Memory Programming

The Flash Controller (FC) works with panels made from 4 KB pages, with each page containing four rows of Flash data. A row is the largest selectable region for contiguous programming of write words. The number of write words in a row varies based on write word width. See the following figure.

Except for the JTAG Test Mode (TMOD12), the FC does not know about the other test modes. The FC state machine accepts commands from the NVMCON register NVMOP commands (NVMCON.NVMOP) in both the Mission mode and the Test Mode.

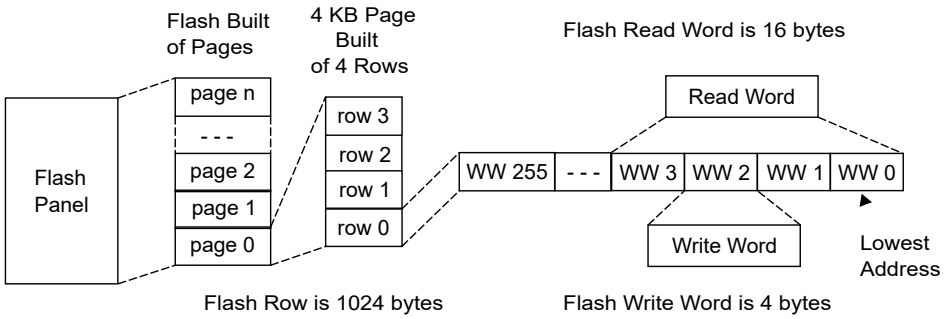
A page of Flash is the smallest unit of memory that can be erased in a single operation. A panel's program Flash space is the only other unit that can be erased in a single operation. All other erases use multiple operations either autonomously by the FC or by software.

A write word is the only unit of memory that can be programmed at a time. All other programming operations are comprised of several contiguous write word program events.

The FC supports the following functions:

- Row Programming – Word-by-word programming until the whole row is programmed; the FC reads the data from the system SRAM
- Quad Programming – The FC performs four writes of data from holding registers
- Single Programming – The FC performs one write of data from holding registers
- Page Erase with FC
- Chip Erase with DSU

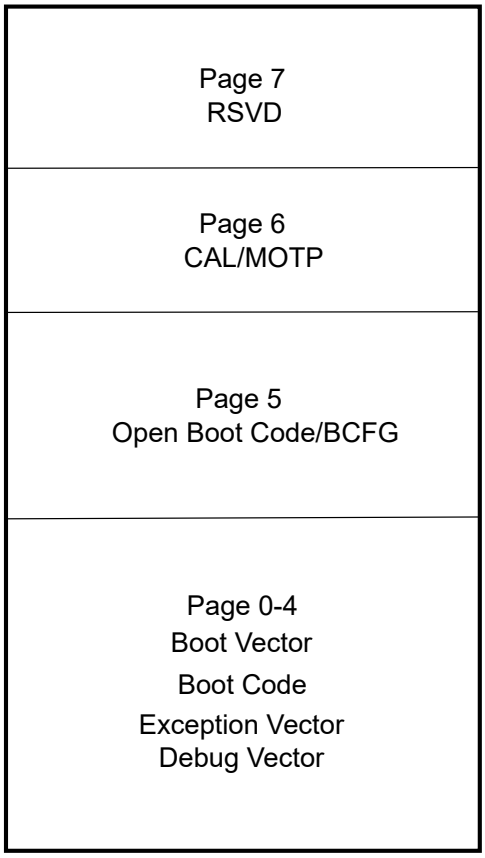
Figure 6-1. 128-Bit Wide Flash Module Application View



Small Flash memories support simple boot loaders and have eight total NVR pages (seven available for the user). The NVMOTP page is the highest user accessible page in the NVR region. NVMOTP is mapped to page NVMOTP_PAGE.

The NVMOTP page is always erase-protected by hardware. The associated LBWP[] and UBWP[] allow writes to the page(s) when logic '0', and prevent writes when logic '1'.

Figure 6-2. 128-Bit Wide Flash Module



6.1.6 NVR Fuses

The fuses in the PIC32CX-BZ2 devices are implemented in the NVR Flash. These flash locations are checked for ECC errors during power-up by the Flash controller. The configuration registers are loaded with the values from the Flash during pre-boot, only if it is trusted.

In order to initialize this area of the NVR such that the ECC is implemented, Quad Word writes must be done for all the FBCFGx, Alternate FBCFGx, FCPN0 and FSIGN0.

Notes: For the FBCFGx registers, the first Quad Word write starts at 0x0004_5F80 with the following contents:

1. FBCFG6/7 are not used, write all '1's.
2. MPLAB X register names display in square brackets.

```
FBCFG4[DEVCFG4] into 0x0004_5F8C
FBCFG5[FUSERID] into 0x0004_5F88
FBCFG6[value=0xFFFFFFFF] into 0x0004_5F84 --> not used
FBCFG7[value=0xFFFFFFFF] into 0x004_5F80 --> not used
```

This is, then, followed by a Quad Word Write to 0x0004_5F90 with the following contents:

```
FBCFG0[FBCFG0] into 0x0004_5F9C
FBCFG1[DEVCFG0] into 0x0004_5F98
FBCFG2[DEVCFG1] into 0x0004_5F94
FBCFG3[DEVCFG2] into 0x0004_5F90
```

Additionally, write the Alternate FBCFGx words; for example, copy the contents of Quad Word from 0x0004_5F80 to 0x0004_5E80, and the contents of Quad Word from 0x0004_5F90 to 0x0004_5E90, again using Quad Word writes only. This provides a backup in the NVR Flash if the primary FBCFGx words are corrupted.

For FCPN0, use the following Quad Word write:

```
FCPN0[FPCN0] into 0x0004_5FBC
FCPN1[value = 0xFFFFFFFF] into 0x0004_5FB8 --> not used
FCPN2[value = 0xFFFFFFFF] into 0x0004_5FB4 --> not used
FCPN3[value = 0xFFFFFFFF] into 0x0004_5FB0 --> not used
```

For FSIGN0, use the following Quad Word write:

```
FSIGN0[FSIGN0] into 0x0004_5FDC
FSIGN1[value = 0xFFFFFFFF] into 0x0004_5FD8
FSIGN2[value = 0xFFFFFFFF] into 0x0004_5FD4
FSIGN3[value = 0xFFFFFFFF] into 0x0004_5FD0
```

Note: Follow the normal page unlock or lock sequences for all NVR Flash writes.

6.1.7 Programming Sequences

This section describes the register addresses and bit fields for the Flash controller.

Table 6-5. Register Addresses for the Flash Controller

Addresses
NVMCON = 0x4400_0600
NVMCONCLR = 0x4400_0604
NVMCONSET = 0x4400_0608
NVMKEY = 0x4400_0620
NVMADDR = 0x4400_0630
NVMDATA0 = 0x4400_0640
NVMDATA1 = 0x4400_0650
NVMDATA2 = 0x4400_0660
NVMDATA3 = 0x4400_0670
NVMSRCADDR = 0x4400_06C0

Table 6-6. Bit Fields for the Flash Controller

Bitfields
WREN_MASK = 0x4000
NVMWR_MASK = 0x8000
NVMERR_MASK = 0x2000
BORERR_MASK = 0x1000
NVMOP_MASK = 0x000F
WORD_NVMOP = 0b0001
QUAD_NVMOP = 0b0010
ROW_NVMOP = 0b0011
PAGE_ERASE_NVMOP = 0b0100
LOWER_ERASE_NVMOP = 0b0101
BULK_ERASE_NVMOP = 0b1111

6.1.7.1 Single or Quad Word Programming Sequence

Follow this sequence of steps for single word or quad word programming:

1. Write data to be programmed into the NVMDATA0 register for a single word, NVMDATA0-3 for a quad word.
2. Load NVMADDR with the address to be programmed.
3. Run the following sequencer start unlock sequence using a word (quad) program command to start the sequence:
    a. Set NVMCON.WREN = 1 (allow writes to NVMCON.NVMWR), and set NVMCON.NVMOP to the desired operation WORD or QUAD (using a single write).
    b. Set NVMKEY = 0x00000000 (reset key).
    c. Set NVMKEY = 0xAA996655.
    d. Set NVMKEY = 0x556699AA.
    e. Write to the target register NVMCONSET to set the NVMWR bit (not the NVMCON register itself). This starts the FC operation.

    Note: The program sequence completes when the hardware clears the NVMCON.NVMWR bit. This pulses the Flash event.
4. Clear the NMVCOM.WREN bit.
5. Check the NVMCON.NVMERR and NVMCON.BORERR bits to ensure that the programming is successful.

6.1.7.2 Row Programming Sequence

The largest block of data that can be programmed by a single NVMOP command is one row. A row is 1024 bytes of data. NVMADDR is the row-aligned address where the Flash address starts programming the data. The controller ignores the sub-row address bits and always starts programming at the beginning of a row.

Follow this sequence of steps for row programming:

1. Write the entire row of data to be programmed into the system SRAM. The source address must be word-aligned but is otherwise unrestricted.
2. Set NVMADDR with the address of the flash row to be programmed.
3. Set NVMSRCADDR with the 32-bit physical source address from step 1.
4. Run the following sequencer start unlock sequence using the row program command to start the sequence:
    a. Set NVMCON.WREN = 1 (allow writes to NVMCON.NVMWR), and set NVMCON.NVMOP to ROW (using a single write).
    b. Set NVMKEY = 0x00000000 (reset key).
    c. Set NVMKEY = 0xAA996655.

d. Set NVMKEY = 0x556699AA.
e. Write to the target register NVMCONSET to set the NVMWR bit (not the NVMCON register itself). This starts the FC operation.

Note: The program sequence completes when the hardware clears the NVMCOM.NVMWR bit. This pulses the Flash event.

5. Clear the NVMCON.NVMWREN bit.
6. Check the NVMCON.NVMERR and NVMCON.BORERR bits to ensure that the programming is successful.

6.1.8 Erase Sequences

6.1.8.1 Page Erase

A page erase performs an erase of a single page of the main program, boot/configuration or sector redundancy Flash. The page to be erased is selected using NVMADDR. The lower bits of the address given by NVMADDR are ignored in page selection. A page of flash can be erased if its associated page write protection is not enabled.

Follow this sequence of steps to erase a page:

1. Set NVMADDR with the address of the page to be erased.
2. Run the following unlock sequence using the page erase command to start the sequence:
    1. Set NVMCON.WREN = `1` (allow writes to NVMCON.NVMWR), and set NVMCON.NVMOP to PAGE_ERASE (using a single write).
    2. Set NVMKEY = 0x00000000 (reset key).
    3. Set NVMKEY = 0xAA996655.
    4. Set NVMKEY = 0x556699AA.
    5. Write to the target register NVMCONSET to set the NVMWR bit (not the NVMCON register itself). This begins the FC operation.

    Note: The erase sequence completes when the hardware clears the NVMCOM.NVMWR bit. This generates an interrupt.
3. Clear the NVMCON.NVMWREN bit.
4. Check the NVMCON.NVMERR and NVMCON.BORERR bits to ensure that the erase sequence completes successfully.

6.1.8.2 Panel Erase

The Program Flash Memory (PFM) can be erased using the respective Higher Addressed or Lower Addressed Panel Erase NVMOP command. These commands leave the boot/config Flash intact. A single panel can be erased if its associated page write protection is not enabled.

Follow this sequence of steps to erase a panel:

1. The PFM runs the following unlock sequence using the selected panel erase command to start the sequence:
    1. Set NVMCON.WREN = `1` (allow writes to NVMCON.NVMWR), and set NVMCON.NVMOP to LOWER_ERASE (using a single write).
    2. Set NVMKEY = 0x00000000 (reset key).
    3. Set NVMKEY = 0xAA996655.
    4. Set NVMKEY = 0x556699AA.
    5. Write to the target register NVMCONSET to set the NVMWR bit (not the NVMCON register itself). This begins the FC operation.

    Note: The erase sequence completes when the hardware clears the NVMCOM.NVMWR bit. This generates an interrupt.
2. Clear the NVMCON.NVMWREN bit.
3. Check the NVMCON.NVMERR and NVMCON.BORERR bits to ensure that the erase sequence completes successfully.

6.1.9 Chip Erase

Erase the entire Flash using a DSU command.

Note: The debugger checks that the chip erase lock is not set before issuing this command. For more details, see the DSU.STATUSB.CELCK bit description in the *PIC32CX-BZ2 and WBZ45 Family Data Sheet* (DS70005504).

The chip erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended that a chip erase be issued after a cold-plugging procedure to ensure that the device is in a known and safe state.

The recommended sequence is as follows:

1. Perform the cold plugging procedure (refer to the *Cold Plugging*). The device then:
   a. Detects the debugger probe.
   b. Holds the CPU in reset.
2. Perform the chip erase command by writing a '1' to CTRL.CE. The device then:
   a. Clears the system volatile memories.
   b. Erases the whole Flash array (including the EEPROM emulation area, not including auxiliary rows).
   c. Erases the lock row, removing the NVMCTRL security bit protection.
3. Check for completion by polling STATUS.DONE (read as one when completed).
4. Reset the device to allow the NVMCTRL update fuses.

Table 6-7. Reference

Step	Commands
Clear flags in STATUSA	WriteD8(@DSUEXT.STATUSA DSU_STATUSA_MASK)
Issue Chip Erase	WriteD8(@DSUEXT.CTRL, DSU_CTRL_CE)

Wait until Erase is done	ReadD8(@DSUEXT.STATUSA,StatusValue)
	While ((StatusValue & DSU_STATUSA_DONE) == 0)
	{
	ReadD8(@DSUEXT.STATUSA,StatusValue)
	}

Device Specific Definitions:

- DSU_CTRL_CE = 0x10
- DSU_STATUSA_MASK = 0x1F
- DSU_STATUSA_DONE = 0x1
- DSUEXT.CTRL = 0x41000100
- DSUEXT.STATUSA = 0x41000101

Note: The recommended time-out for waiting for erase completion is 20 seconds (the data sheet specifies a maximum of 12s).

6.1.9.1 Bulk Erase

Note: The entire Bulk Erase section is internal only.

Bulk erase performs a complete erase of the Main Program (PFM), sector redundancy and all of the NVR Flash. It may only be used in the private test mode.

A bulk erase sequence comprises the following steps:

1. Run the following unlock sequence using the bulk erase command to start the sequence:
   1. Set NVMCON.WREN = 1 (allow writes to NVMCON.NVMWR), and set NVMCON.NVMOP to BULK_ERASE (using a single write).
   2. Set NVMKEY = 0x00000000 (reset key).
   3. Set NVMKEY = 0xAA996655.
   4. Set NVMKEY = 0x556699AA.
   5. Write to the target register NVMCON to set the NVMWR bit. This starts the FC operation.

Notes:

1. The erase sequence completes when the hardware clears the NVMCON.NVMWR bit.
2. Do not pulse the output after a bulk erase.

2. Clear the NVMCON.NVMWREN bit.
3. Check the NVMCON.NVMERR and NVMCON.BORERR bits to ensure that the erase sequence completes successfully.

Table 6-8. Erase Options

Flash Partition	Memory Region	Page Erase	Panel Erase	Chip Erase	Bulk Erase
BFM	Boot Flash	Yes, Configurable	No	Yes	Yes
	Device/Boot Config	Yes, Configurable	No	Yes	Yes
	OTP (One Time Programmable)	No	No	No	Yes
PFM	Program Flash Memory	Yes, Configurable	Yes	Yes	Yes

7. Appendix A: Device JTAG IDs and DSU Device ID

Table 7-1. Device JTAG ID

Part Number	JTAG ID
PIC32CX1012BZ25048/WBZ451	0x09B8F053

Table 7-2. DSU Device ID

Part Number	DSU Device ID
PIC32CX1012BZ25048/WBZ451	0x9B8F

Note: The first four bits of the 32-bit device ID indicates the silicon revision. For more details on the differences, see 8. Appendix B: DSU.DID vs JTAG ID .

8. Appendix B: DSU.DID vs JTAG ID

The JTAG ID at memory location 0x4400_0090 is left in the device for legacy reasons and/or for being read over JTAG during private test modes. The device ID to be used by an external debugger or programmers must be within the DSU [DSU.DID register] located at address 0x4100_00118 (external address) or 0x4100_0018 (internal address). Only the lower 16 bits of this register are used for the purpose of external tools to identify the device. It is recommended that the upper 16 bits be masked off on read.

Many of the fields in the JTAG ID and the DSU.DID register are the same, but positioned differently.

The DSU.DID register has some of its bit fields loaded from the TCFG registers in the Flash RSVD Page 7 (Factory Test Memory), which includes addresses from 0x0004_7000 to 0x0004_7FFF.

Refer to the following figure for how the JTAG ID and DSU.DID are correlated, and refer to 7. Appendix A: Device JTAG IDs and DSU Device ID.

The DSU.DID register bits for the family/series/DEVSEL come from FTCFG2 in the factory test memory page.

Figure 8-1. DSU.DID VS JTAG ID

REGISTER 5-10 JTAG ID	addr:0x4400_0090	value:0x09B[8F|0F|0B]053	eg: 48-pin device: 0x09B8F053
REGSITER 3-70 DSU DID	addr:0x4100_0118	value:0x00009B[8F|0F|0B]	eg: 48-pin device: 0x00009B8F

REGISTER 5-10 JTAG ID	VERSION CODE: 0000 = A.0				MASK ID: 9B = SG405								PART NUMBER DEV ID = 8F/0F/0B			
	VER3	VER2	VER1	VER0	PN15	PN14	PN13	PN12	PN11	PN10	PN9	PN8	PN7	PN6	PN5	PN4
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	1	0	0	1	1	0	1	1	1/0	1/0	1/0	1/0

	PART NUMBER DEV ID = 8F/0F/0B				JEDEC MAN ID: MCHP = 29											FIXED
	PN3	PN2	PN1	PN0	MANID11	MANID10	MANID9	MANID8	MANID7	MANID6	MANID5	MANID4	MANID3	MANID2	MANID1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1/0	1/0	1/0	1/0	0	0	0	0	0	1	0	1	0	0	1	1

REGSITER 3-70 DSU DID	REVISION: 0000 = A.0				FAMILY: ZBT COMBO					N/A	SERIES: 0x0 = FIRST IN SERIES					
	VER3	VER2	VER1	VER0	FAM4	FAM3	FAM2	FAM1	FAM0	RSVD	SER5	SER4	SER3	SER2	SER1	SER0
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	X	0	0	0	0	0	0

	DIE (MASK ID): 9B = SG405								DEVSEL = 8F/0F/0B							
	DIE7	DIE6	DIE5	DIE4	DIE3	DIE2	DIE1	DIE0	DEVSEL7	DEVSEL6	DEVSEL5	DEVSEL4	DEVSEL3	DEVSEL2	DEVSEL1	DEVSEL0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	0	0	1	1	0	1	1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0

9. Document Revision History

Revision	Date	Section	Description
A	10/2022	Document	Initial revision

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded

by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-7827-0

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820