

---

---

# ON-CHIP DEBUGGER SPECIFICATION

Information contained in this publication regarding device applications and the like is intended by way of suggestion only. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip.

© 2001 Microchip Technology Incorporated. All rights reserved.

The Microchip logo, name, PIC, PICmicro, PICMASTER, PICSTART, and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. MPLAB and *Smart Serial* are trademarks of Microchip Technology in the U.S.A. and other countries.

All product/company trademarks mentioned herein are the property of their respective companies.

# On-Chip Debugger Specification

---

---

---

**Table of Contents**

---

---

**Chapter 1. Overview**

1.1 Introduction .....	5
1.2 Highlights .....	5
1.3 On-Chip Debugger Definition .....	5
1.4 Debugger vs. Emulator .....	6

**Chapter 2. PIC16F87X Debugger Control**

2.1 Introduction .....	7
2.2 Highlights .....	7
2.3 Enabling/Disabling the On-Chip Debugger .....	7
2.4 Using the On-Chip Debugger Registers .....	8
2.5 Communicating with the Target Device .....	9
2.6 Resetting and Running Programs .....	9
2.7 Halting Execution .....	9
2.8 Additional Considerations .....	11

<b>Index .....</b>	<b>13</b>
--------------------	-----------

# On-Chip Debugger Specification

---

---

---

## Chapter 1. Overview

---

---

### 1.1 Introduction

This chapter provides an overview of the on-chip debugger.



The information in this document is provided as-is and is subject to change. No guarantees are given or implied. The user assumes all risks associated with using this information.

### 1.2 Highlights

Topics covered in this chapter are:

- On-Chip Debugger Definition
- Debugger vs. Emulator

### 1.3 On-Chip Debugger Definition

An on-chip debugger is special hardware and software that works with specific PICmicro<sup>®</sup> devices to give developers a low cost method for debugging their code. These PICmicro MCU's must (1) contain special on-chip logic supporting debugging functionality and (2) provide In-Circuit Serial Programming<sup>™</sup> (ICSP<sup>™</sup>) capabilities.

The ICSP capability allows in-circuit programming of the PICmicro MCU via specific hardware pins designated for this purpose. ICSP is described in detail in the *In-Circuit Serial Programming (ICSP) Guide (DS30277)*. Additionally, ICSP-related technical briefs and application notes, as well as device programming specifications, may be found at our website ([www.microchip.com](http://www.microchip.com)).

The on-chip debug capability provides breakpointing, single stepping, and external breaking. Breaks are treated as a special form of interrupt that vector to debug executive software implemented by the developer. The same hardware pins used by the ICSP capability are preserved by the on-chip debug capability while servicing the debug break, allowing them to be used to implement communications between the debug executive software and a host processor.

# On-Chip Debugger Specification

---

## 1.4 Debugger vs. Emulator

Traditionally, embedded systems engineers use in-circuit emulators (ICE) to develop and debug their designs. The on-chip debug facilities in some PICmicro devices provide a low cost alternative to a more expensive ICE.

When implemented, the on-chip debugger logic is part of the actual microcontroller silicon. As a result, there is a trade-off between the cost of producing this silicon on every chip (even though only a few will actually use this logic) and the power of the logic itself. It would be possible to put an entire emulator with complex breakpoints and tracing capabilities on a chip, but the cost would be prohibitively high. Usually the on-chip debugger provides the means to set simple breakpoints, query the internal state of the chip, and single step through code.

A microcontroller utilizing its on-chip debug facilities has these benefits:

- Low cost
- Requires a minimum of extra hardware
- Does not require expensive sockets or adapters
- Able to debug a production line board

In order to have the above benefits, on-chip debuggers have these trade-offs compared to ICEs:

- Use of some target system resources such as pins, program and/or data memory, and stack space may mean that some portions of an embedded application cannot be debugged.
- Triggering and breakpointing are limited to the capabilities of the on-chip debug silicon.
- The target chip must actually be running with a clock and a supply voltage. Often an emulator probe can run without any external hardware.

Overall, an in-circuit emulator usually provides the most powerful debugging solution, but as chips get faster and more complex, emulators require increasingly expensive high speed logic and high priced adapters to connect to the denser chip packages.

## Chapter 2. PIC16F87X Debugger Control

### 2.1 Introduction

This chapter describes the hardware and software requirements for on-chip debugging using the PIC16F87X.

### 2.2 Highlights

Topics covered in this chapter are:

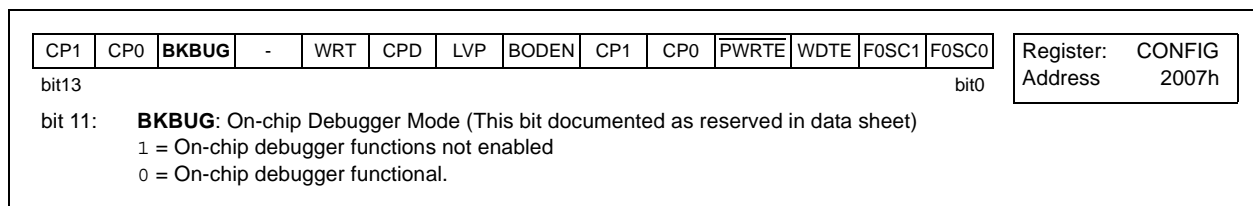
- Enabling/Disabling the On-Chip Debugger
- Using the On-Chip Debugger Registers
- Communicating with the Target Device
- Resetting and Running Programs
- Halting Execution
- Additional Considerations

### 2.3 Enabling/Disabling the On-Chip Debugger

The configuration bit **BKBUG** (bit 11 in the configuration word) is used to enable/disable the on-chip debugger (Figure 2.1).

See the programming specifications for PIC16C87X devices (DS39025) for more information on programming the configuration word.

**Figure 2.1: CONFIGURATION WORD**



# On-Chip Debugger Specification

## 2.4 Using the On-Chip Debugger Registers

There are two registers associated with debugger control: the `ICKBUG` and `BIGBUG` registers (Figure 2.2 and Figure 2.3.) Between these two registers, there are 3 bits for debugger control and 13 bits for breakpoint address specification.

**INBUG** - Set/cleared automatically when entering/exiting debug executive. A four (4) instruction cycle delay will prevent the new state of zero from being recognized by the core to allow a safe debugger exit. When set, interrupts are disabled (regardless of the state of the `GIE` bit) and the debugger `HALT` is disabled. This prevents reentry into the debugger while communicating with computer software.

**FREEZ** - When `FREEZ` is set, `TMR0`, `TMR1` and `TMR2` and their prescalers freeze at a break (suspend counting). The `SSP`, `A/D` converter and `USART` state machines will also suspend upon the assertion of the `FREEZ` bit. Status flag bits that are altered by register read operations are unaffected while `FREEZ` is asserted.

**SSTEP** - Enables single step operation. After a return to normal operation (`INBUG=0`), if `SSTEP` is set, the on-chip logic will allow one user instruction to execute before the on-chip debugger is reentered.

**BKAnn** - Break address `nn`. These bits are cleared at POR. No other reset clears the bits, (i.e., a `MCLR` cannot change the breakpoint address value).

**Figure 2.2: ICKBUG Register (Address 18Eh)**

	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INBUG	FREEZ	SSTEP	BKA12	BKA11	BKA10	BKA09	BKA08
bit7								bit0
bit 7:	<b>INBUG:</b> On-chip debugger activity status 1 = Device is executing on-chip debugger code 0 = Device is executing user code (This bit is read only; set from on-chip halt or breakpoint occurrence; only clear by RETURN)							
bit 6:	<b>FREEZ:</b> on-chip debugger freeze mode 1 = Peripherals will freeze when <code>INBUG=1</code> 0 = Peripherals will not freeze when <code>INBUG=1</code>							
bit 5:	<b>SSTEP:</b> Single Step Enable 1 = Program will execute 1 instruction word of user code upon RETURN from debug code 0 = Program will execute multiple instruction words of user code							
bit 4-0:	<b>BKA:</b> Breakpoint Address When writing, represents the requested breakpoint address. When reading, represents the value of the PC at last breakpoint, halt, or single step operation.							

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset



# PIC16F87X Debugger Control

Figure 2.3: BIGBUG Register (Address 18Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BKA07	BKA06	BKA05	BKA04	BKA03	BKA02	BKA01	BKA00
bit7							bit0

bit 7-0: **BKA**: Breakpoint Address  
When writing, represents the requested breakpoint address.  
When reading, represents the value of the PC at last breakpoint, halt, or single step operation.

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

## 2.5 Communicating with the Target Device

The debug executive establishes the communication channel between the target device (RB<7:6>) and computer software (e.g., MPLAB IDE.)

Since the debugger will issue read-modify-write instructions (BSF, BCF) on PORTB for communication with the debugger interface module, addresses 0x106 and 0x186 should be used to modify PORTB<7:6> and TRISB<7:6> so that bits <5:0> are not corrupted. If the user requests a read or a write of PORTB or TRISB, addresses 0x006 and 0x086 can be used to access the RB<5:0> portions of the register.

## 2.6 Resetting and Running Programs

In the debug environment, the target device reset pin can be activated by the module/target connector or by the target system resources. After reset, the target device will begin running the target program. To begin debugger execution, initiate a hardware halt (on I/O pin.)

**Note:** A hardware halt after reset will execute location 0x0000 and then skip to 0x0001. Therefore, a NOP is recommended at location 0x0000.

## 2.7 Halting Execution

If BKBUG is programmed to 0, the INBUG flag is clear, and a halt signal occurs, normal program execution is halted and the on-chip debugger is entered. A halt signal will be generated when either:

- RB6 transitions from high to low
- the current PC value matches the value of BKA<12:0>
- the SSTEP bit is set

# On-Chip Debugger Specification

When normal program execution is halted, the current instruction will be executed and then the program will be stopped at the address of the next instruction. The processor will go through what appears to be a normal interrupt cycle, except it will not disturb the contents of the `GIE` bit. On-chip debugger logic will control sequencing of the key interrupt signals in the core logic.

Once in debug mode, the `INBUG` flag is set. The PC is pushed onto the stack and stored in `BKAnn`. This allows proper reporting of the current state of the PC (next instruction to be executed) when the on-chip debugger is entered. The PC is then set to `0x2004`.

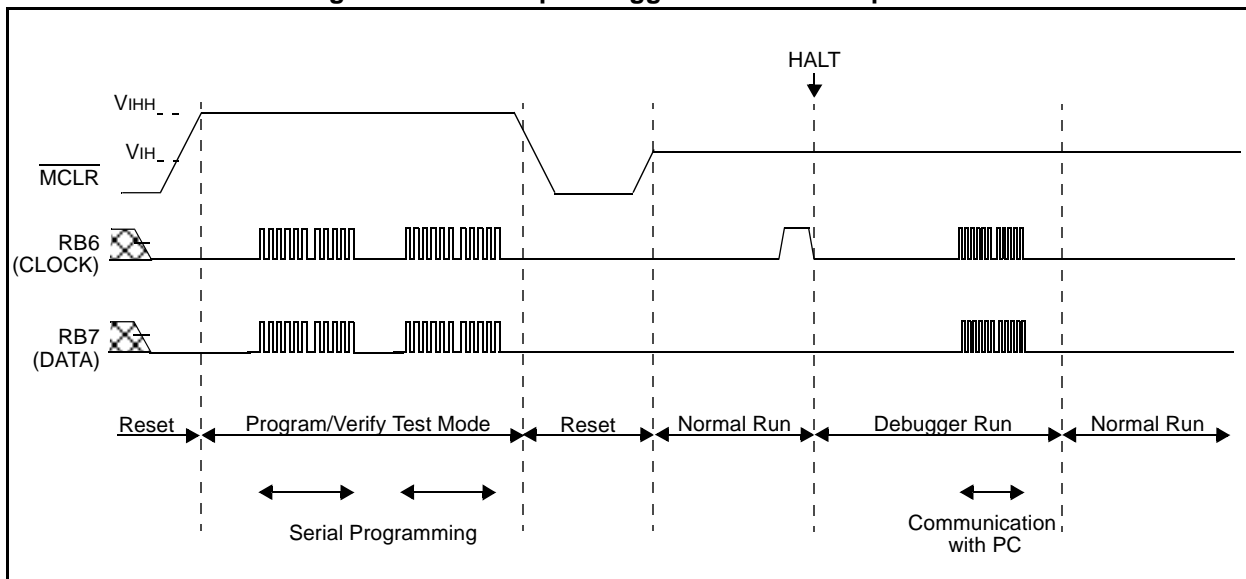
As the processor vectors to `0x2004`, the CPU will fetch and execute the instruction at this address. Therefore, address 2004 must be programmed with a `GOTO` opcode, where `opcode` is pointing to the beginning address of the debug code. Then the on-chip debugger routine will commence. `PCLATH` will be undisturbed, and the PC at breakpoint is at the top of the stack.

The debugger software will use a `RETURN` instruction to return to the mainline code. The `RETURN` will cause the `INBUG` bit to be cleared and will also release peripheral freeze at the proper time, if the `FREEZ` bit had been set.

## 2.7.1 Halting Execution by I/O Pin

For the conditions specified in Section 2.7, an edge detection circuit will generate a `HALT` signal pulse when `RB6` transitions from high to low. This is also known as external or hardware halting.

Figure 2.4: On-chip Debugger Execution Sequence



## 2.7.2 Halting Execution by Address Breakpoint

For the conditions specified in Section 2.7, another HALT method is by address breakpoint. The `ICKBUG` and `BIGBUG` registers contain a 13-bit value (`BKAnn`) that is compared against the current PC value. When the values are equal, then the circuit will generate a halt signal on that cycle.

Power-up initializations the state of the `ICKBUG` and `BIGBUG` registers to `0x0000`, equal to the reset vector.

**Note:** A hardware halt after reset will execute location `0x0000` and then skip to `0x0001`. Therefore, a `NOP` is recommended at location `0x0000`.

Disabling the breakpoints is implemented by setting a breakpoint address that lies in the address space of the on-chip debugger (ex: `0x1F00`.)

## 2.7.3 Halting Execution by Single Stepping

For the conditions specified in Section 2.7, another HALT method is by single step. Upon exit of the debug routine by the `return` instruction, if the `SSTEP` bit is set, the on-chip debugger logic will generate a HALT signal timed to allow one instruction execution. After this, the device will reenter the on-chip debugger routine in the same fashion as the external (I/O pin) halt.

## 2.8 Additional Considerations

In addition to the logic required to support the on-chip debugger as described in the above sections, there are several issues of which to be aware.

- Watchdog Timer will still be functional with no special modes associated with on-chip debugger.
- The reset, POR and BOR logic will still be functional with no special modes associated with the on-chip debugger.

# On-Chip Debugger Specification

---

NOTES:

**Index**

<b>A</b>		On-Chip Debugger Registers .....	8
A/D .....	8	On-Chip Debugger, Enabling/Disabling .....	7
Address 18E .....	8	Overview .....	5
Address 18F .....	9	<b>P</b>	
Address 2004 .....	10	PIC16F87X Debugger Control .....	7
Address 2007 .....	7	POR .....	8, 11
<b>B</b>		PORTB .....	9
BIGBUG Register .....	9, 11	<b>R</b>	
BKAnn .....	8, 9, 10	RB6 .....	9, 10
BKBUG .....	7, 9	Register	
BOR .....	11	BIGBUG .....	9, 11
<b>C</b>		ICKBUG .....	8, 11
Communicating with the Target Device .....	9	Reset .....	11
Configuration Word .....	7	Resetting Programs .....	9
<b>D</b>		Running Programs .....	9
Debugger vs. Emulator .....	6	<b>S</b>	
<b>E</b>		SSP .....	8
Enabling/Disabling On-Chip Debugger .....	7	SSTEP .....	8, 9, 11
Execution Sequence .....	10	<b>T</b>	
<b>F</b>		Timers .....	8
FREEZ .....	8, 10	<b>U</b>	
<b>G</b>		USART .....	8
GIE Bit .....	8, 10	Using On-Chip Debugger Registers .....	8
<b>H</b>		<b>W</b>	
Halting Execution		Watchdog Timer .....	11
Breakpoint .....	11		
I/O Pin .....	9, 10		
Single Stepping .....	11		
Hardware Considerations .....	11		
<b>I</b>			
I/O Pin .....	9		
ICE .....	6		
ICKBUG Register .....	8, 11		
ICSP .....	5		
INBUG .....	8, 9, 10		
<b>M</b>			
MCLR .....	8		
<b>O</b>			
On-Chip Debugger Definition .....	5		



# MICROCHIP

## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200 Fax: 480-792-7277  
Technical Support: 480-792-7627  
Web Address: <http://www.microchip.com>

#### Rocky Mountain

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7966 Fax: 480-792-7456

#### Atlanta

500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770-640-0034 Fax: 770-640-0307

#### Austin

Analog Product Sales  
8303 MoPac Expressway North  
Suite A-201  
Austin, TX 78759  
Tel: 512-345-2030 Fax: 512-345-6085

#### Boston

2 Lan Drive, Suite 120  
Westford, MA 01886  
Tel: 978-692-3848 Fax: 978-692-3821

#### Boston

Analog Product Sales  
Unit A-8-1 Millbrook Tarry Condominium  
97 Lowell Road  
Concord, MA 01742  
Tel: 978-371-6400 Fax: 978-371-0050

#### Chicago

333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071 Fax: 630-285-0075

#### Dallas

4570 Westgrove Drive, Suite 160  
Addison, TX 75001  
Tel: 972-818-7423 Fax: 972-818-2924

#### Dayton

Two Prestige Place, Suite 130  
Miamisburg, OH 45342  
Tel: 937-291-1654 Fax: 937-291-9175

#### Detroit

Tri-Atria Office Building  
32255 Northwestern Highway, Suite 190  
Farmington Hills, MI 48334  
Tel: 248-538-2250 Fax: 248-538-2260

#### Los Angeles

18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888 Fax: 949-263-1338

#### Mountain View

Analog Product Sales  
1300 Terra Bella Avenue  
Mountain View, CA 94043-1836  
Tel: 650-968-9241 Fax: 650-967-1590

#### New York

150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 631-273-5305 Fax: 631-273-5335

#### San Jose

Microchip Technology Inc.  
2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408-436-7950 Fax: 408-436-7955

#### Toronto

6285 Northam Drive, Suite 108  
Mississauga, Ontario L4V 1X5, Canada  
Tel: 905-673-0699 Fax: 905-673-6509

### ASIA/PACIFIC

#### Australia

Microchip Technology Australia Pty Ltd  
Suite 22, 41 Rawson Street  
Epping 2121, NSW  
Australia  
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

#### China - Beijing

Microchip Technology Beijing Office  
Unit 915  
New China Hong Kong Manhattan Bldg.  
No. 6 Chaoyangmen Beidajie  
Beijing, 100027, No. China  
Tel: 86-10-85282100 Fax: 86-10-85282104

#### China - Shanghai

Microchip Technology Shanghai Office  
Room 701, Bldg. B  
Far East International Plaza  
No. 317 Xian Xia Road  
Shanghai, 200051  
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

#### Hong Kong

Microchip Asia Pacific  
RM 2101, Tower 2, Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2401-1200 Fax: 852-2401-3431

#### India

Microchip Technology Inc.  
India Liaison Office  
Divyasree Chambers  
1 Floor, Wing A (A3/A4)  
No. 11, O'Shaughnessey Road  
Bangalore, 560 025, India  
Tel: 91-80-2290061 Fax: 91-80-2290062

#### Japan

Microchip Technology Intl. Inc.  
Benex S-1 6F  
3-18-20, Shinyokohama  
Kohoku-Ku, Yokohama-shi  
Kanagawa, 222-0033, Japan  
Tel: 81-45-471-6166 Fax: 81-45-471-6122

### ASIA/PACIFIC (continued)

#### Korea

Microchip Technology Korea  
168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-Ku  
Seoul, Korea  
Tel: 82-2-554-7200 Fax: 82-2-558-5934

#### Singapore

Microchip Technology Singapore Pte Ltd.  
200 Middle Road  
#07-02 Prime Centre  
Singapore, 188980  
Tel: 65-334-8870 Fax: 65-334-8850

#### Taiwan

Microchip Technology Taiwan  
11F-3, No. 207  
Tung Hua North Road  
Taipei, 105, Taiwan  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### Denmark

Microchip Technology Denmark ApS  
Regus Business Centre  
Lautrup høj 1-3  
Ballerup DK-2750 Denmark  
Tel: 45 4420 9895 Fax: 45 4420 9910

#### France

Arizona Microchip Technology SARL  
Parc d'Activite du Moulin de Massy  
43 Rue du Saule Trapu  
Batiment A - 1er Etage  
91300 Massy, France  
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

#### Germany

Arizona Microchip Technology GmbH  
Gustav-Heinemann Ring 125  
D-81739 Munich, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

#### Germany

Analog Product Sales  
Lochhamer Strasse 13  
D-82152 Martinsried, Germany  
Tel: 49-89-895650-0 Fax: 49-89-895650-22

#### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-039-65791-1 Fax: 39-039-6899883

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Winnersh Triangle  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/30/01

All rights reserved. © 2001 Microchip Technology Incorporated. Printed in the USA. 3/01 Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, except as maybe explicitly expressed herein, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.