The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205373

FR, MB91460, Bit Search

This application note describes the functionality of the Bit Search and gives some examples.

Contents

1 Introduction....1
1.1 Key Features....1
2 Bit Search....1
2.1 Block Diagram....1
2.2 Registers....3
2.3 Operation....3
3 Bit Search Examples....4
3.1 Restoring Previous Bit Search State....4
4 Additional Information....5

1 Introduction

This application note describes the functionality of the Bit Search and gives some examples.

1.1 Key Features

- Detects the first ‘0’ position
- Detects the first ‘1’ position
- Detects the first position where data changes from ‘0’ to ‘1’ or vice versa
- Previous bit search result can be restored

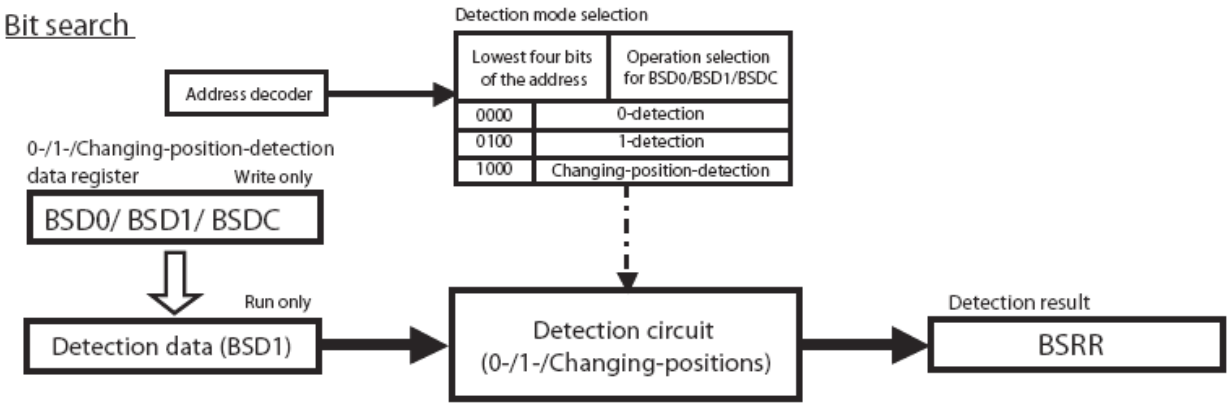
2 Bit Search

The basic functionality of Bit Search

2.1 Block Diagram

Figure 1 shows the internal block diagram of the Bit Search unit.

Figure 1. Bit Search Block Diagram



2.2 Registers

2.2.1 0 Detection Register (BSD0)

If the first '0' position needs to be detected in a word then it should be loaded in to `BSD0` register.

2.2.2 1 Detection Register (BSD1)

If the first '1' position needs to be detected in a word then it should be loaded in to `BSD1` register.

It should be noted that this register can be read as well as written. The reading of this register can be used restore previous bit search result.

2.2.3 Changing Position Detection Data Register (BSDC)

If the first position where data changes from '0' to '1' or vice versa needs to be detected in a word then it should be loaded in to the BSDC register.

2.2.4 Detection Result Register (BSRR)

The BSRR register contains the latest detection result for the data written in either BSD0 or BSD1 or BSDC. The type of detection result cannot be determined i.e. the information related to whether the result is related to 0 detection or 1 detection or changing position detection cannot be determined from BSRR.

2.3 Operation

The detection result in BSRR register is basically with reference to the MSB (bit-31) of the data written in either BSD0 or BSD1 or BSDR. That means in case of 0 position detection if the data written to BSD0 is 0xFFFFFFFE then the BSSR register would have the result as 0x0000001F. This is because the first occurrence of 0 is at the 31st bit with reference to MSB. When there is no 0 or 1 or changing position detected respectively in the case of BSD0 or BSD1 or BSDR then the BSRR register would have the result as 0x00000020.

The result is available immediately after writing either BSD0 or BSD1 or BSDR.

Table 1. Bit Search Examples

Sr. No.	Type of Detection	Register Used	Data	Result in BSRR
1	Zero	`BSD0`	`0x00000000`	`0x00000000`
2	Zero	`BSD0`	`0x800055AA`	`0x00000001`
3	Zero	`BSD0`	`0xFFFF0000`	`0x00000010`
4	Zero	`BSD0`	`0xFFFFFFFF`	`0x00000020`
5	One	`BSD1`	`0x00000000`	`0x00000020`
6	One	`BSD1`	`0x800055AA`	`0x00000000`
7	One	`BSD1`	`0x0000FFFF`	`0x00000010`

Sr. No.	Type of Detection	Register Used	Data	Result in BSRR
8	One	BSD1	0xFFFFFFFF	0x00000000
9	Changing Position	BSDC	0x00000000	0x00000020
10	Changing Position	BSDC	0xFFFFFFFF	0x00000020
11	Changing Position	BSDC	0xFF0000FF	0x00000008
12	Changing Position	BSDC	0x000005AA	0x00000015
13	Changing Position	BSDC	0x01234567	0x00000007

3 Bit Search Examples

Examples for Bit Search

3.1 Restoring Previous Bit Search State

The following example demonstrates to restore the previous bit search results when the bit search unit is simultaneously used in the application and the interrupt service routine. Here the bit search unit is used the main routine “normally” whereas in order to use it in the ISR first the contents of the BSD1 register is saved in a temporary variable, the required bit search is performed and then the original contents of the BSD1 register is copied back to it. This way previous bit search state and corresponding results are restored no matter the previous bit search was 0-detection, 1-detection or changing position detection.

```
void main(void)
{
       unsigned int Data[3] = {0x12340000, 0xFFAA5500, 0xFFF3230F};
       unsigned int Result[3];
       __EI();           // Enable interrupts
       __set_il(31);     // Allow all levels
       InitIrqLevels();  // Init interrupts
       PORTEN = 0x3;     // Enable I/O Ports
                         // This feature is not supported by MB91V460A
                         // For all other devices the I/O Ports must be enabled
       . . .
   BSD0 = Data[0];   // Detect the first 0-position in Data[0]
       Result[0] = BSRR; // Detection result copied to Result[0]
       . . .
   BSD1 = Data[1];   // Detect the first 1-position in Data[1]
       Result[1] = BSRR; // Detection result copied to Result[1]
       . . .
   BSD2 = Data[2];   // Detect the changing position in Data[2]
       Result[2] = BSRR; // Detection result copied to Result[2]
              . . .
   while(1)           // endless loop
   {
             HWWD_CL = 0;
          }
   }
   __interrupt void Resource1_IRQ(void)
   {
       unsigned int temp,
       unsigned int I_Data = 0xFEDCBA98;
       unsigned int I_Result;
       temp = BSD1;       // Evacuate BSD1

       BSD2 = I_Data;     // Perform the intended bit search
   I_Result = BSRR;

   BSD1 = temp;       // Restore BSD1
   }
```

4 Additional Information

Information about CYPRESS Microcontrollers can be found on the following Internet page:

http://www.cypress.com/cypress-microcontrollers

Document History

Document Title: AN205373 - FR, MB91460, Bit Search

Document Number: 002-05373

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	06/17/2008	First Version; MPi
*A	5132492	NOFL	02/10/2016	Converted Spansion Application Note “MCU-AN-300079-E-V10” to Cypress format
*B	5862900	AESATMP9	08/24/2017	Updated logo and copyright.
*C	6070279	NOFL	02/13/2018	Sunset Review Migrated to new template Updated links

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2008-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.
TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.
Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.