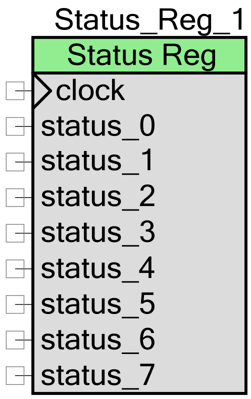
Status Register

1.50



Features

- Up to 8-bit Status Register

General Description

The Status Register allows the firmware to read digital signals.

When to Use a Status Register

Use the Status Register when the firmware needs to query the state of internal digital signals.

Input/Output Connections

This section describes the input connections for the status register. An asterisk (*) in the list of I/Os indicates that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

clock – Input

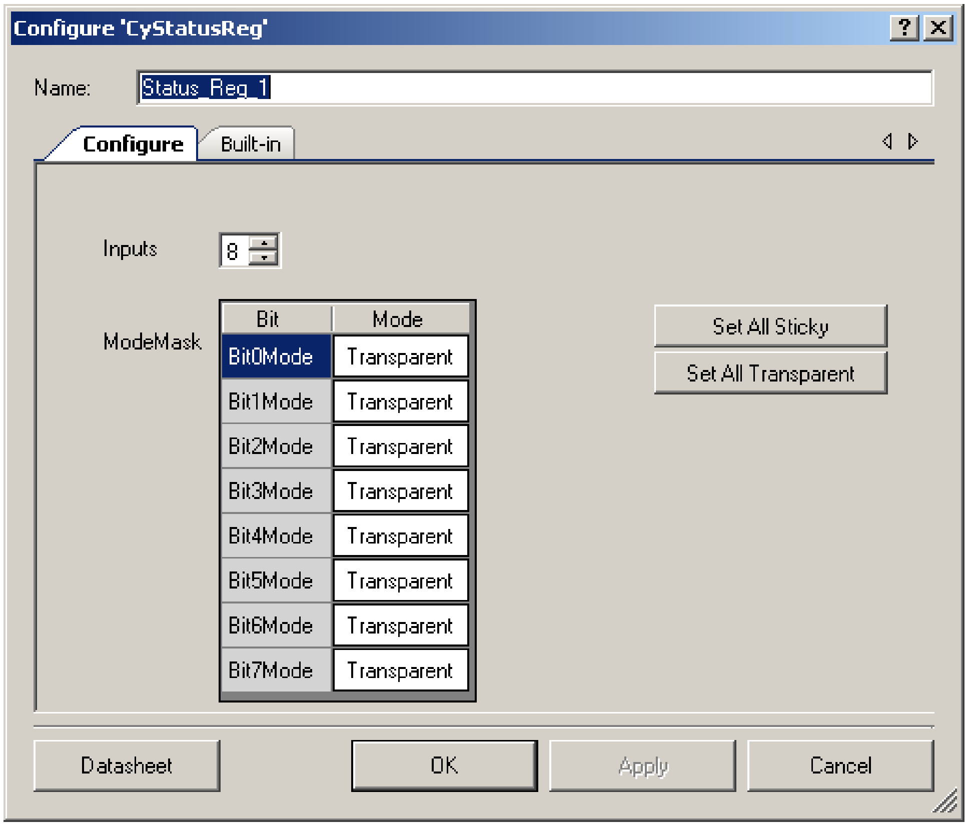
Status register clock. The clock input signal is ignored for bits configured as Transparent.

status_0 - status_7 – Input *

Status register input. The firmware queries the input signals by reading the status register. The number of inputs depends on the **Inputs** parameter.

Component Parameters

Drag a Status Register onto your design and double-click it to open the **Configure** dialog.



Name: Status_Reg_1

Inputs 8

ModeMask

Bit	Mode
Bit0Mode	Transparent
Bit1Mode	Transparent
Bit2Mode	Transparent
Bit3Mode	Transparent
Bit4Mode	Transparent
Bit5Mode	Transparent
Bit6Mode	Transparent
Bit7Mode	Transparent

Inputs

Number of input terminals (1 to 8). The default value is **8**.

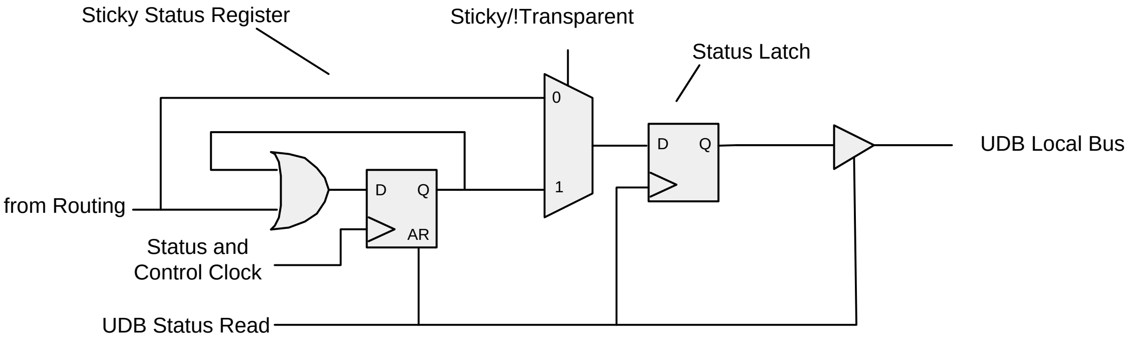
ModeMask (Bit0Mode – Bit7Mode)

These parameters are used to set specific bits of the Status Register to be held high after being registered, until a read is executed. That read clears all registered values. The settings are:

- **Transparent** – By default, a CPU read of this register transparently reads the state of the associated routing net and is asynchronous to the block clock. This mode can be used for transient state that is computed and registered internally in the UDB.

- **Sticky (Clear on Read)** – In this mode, the associated routing net is sampled on each cycle of the status and control clock. If the signal is high in a given sample, it is captured in the status bit and remains high, regardless of the subsequent state of the associated route. When CPU firmware reads the status register, the bit is cleared. The status register clearing is independent of mode and will occur even if the block clock is disabled; it is based on the bus clock and occurs as part of the read operation.

Figure 1. Behavior of Transparent versus Sticky Modes



Set All Sticky

This button sets all of the bits to Sticky mode.

Set All Transparent

This button sets all of the bits to Transparent mode.

Resources

Analog Blocks	Digital Blocks					API Memory (Bytes)		Pins (per External I/O)
	Datapaths	Macro cells	Status Registers	Control Registers	Counter7	Flash	RAM	
N/A	N/A	N/A	1	N/A	N/A	6	0	N/A

The status register requires one UDB Status Register.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software.

By default, PSoC Creator assigns the instance name “Status_Reg_1” to the first instance of a status register in any given design. You can rename the component to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following function is “StatusReg.”

uint8 StatusReg_Read (void)

Description: Reads the value of a status register.

Parameters: None

Return Value: Returns the current value of a status register.

Side Effects: None

Sample Firmware Source Code

PSoC Creator provides numerous example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.50.d	Minor datasheet edit.	
1.50.c	Minor datasheet edit.	
1.50.b	Datasheet edits	
1.50.a	Datasheet edits	
1.50	Updated the Configure dialog.	Created a customized interface. Added "Set All" buttons and changed Number of Inputs field to allow keyboard entry. Updated the dialog to comply with corporate standards.

© Cypress Semiconductor Corporation, 2010-2016. 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. 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, 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.