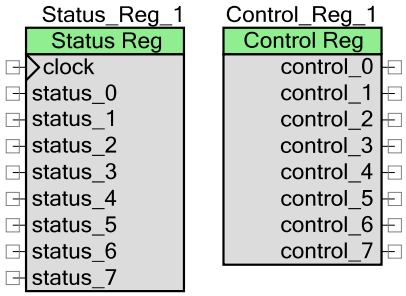
Status and Control Registers

1.0

Features

- Up to 8-bit Status Register
- Up to 8-bit Control Register



General Description

The status register allows the firmware to read digital signals. The control register allows the firmware to output digital signals.

When to use a status or control register

The status register should be used when the firmware needs to query the state of internal digital signals.

The control register should be used when the firmware needs to interact with a digital system. The control register may be used as a configuration register, allowing the firmware to specify the desired behavior of the digital system.

Input/Output Connections

This section describes the various input and output connections for the status and control registers. An asterisk (*) in the list of I/O's states 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 Only)

Status register clock (status register only).

status_0 - status_7 – Input *

Status register input (status register only). The firmware queries the input signals by reading the status register. The number of inputs depends on the NumInputs parameter.

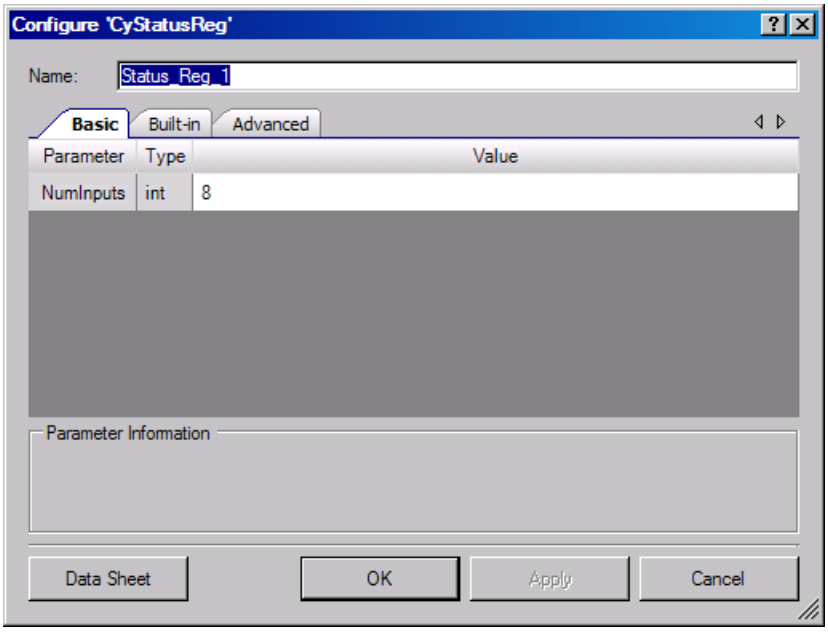
control_0 - control_7 – Output *

Control register output (control register only). The firmware sets the values of the output terminals by writing to the control register. The number of outputs depends on the NumOutputs parameter.

Component Parameters

Drag a status register or control register onto your design and double-click it to open the Configure dialog.

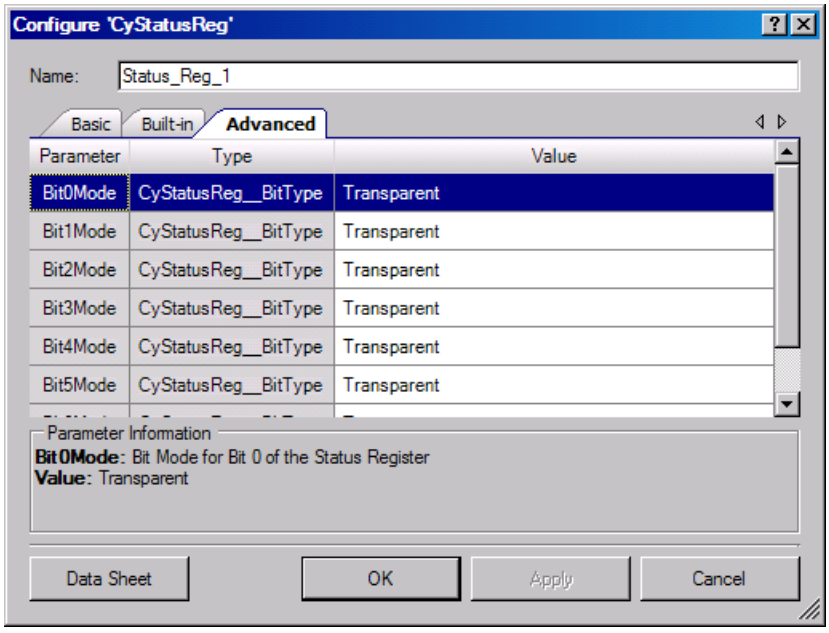
Configure Status Register Dialog – Basic Tab



NumInputs

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

Configure Status Register Dialog – Advanced Tab



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
- Sticky (Clear on Read)

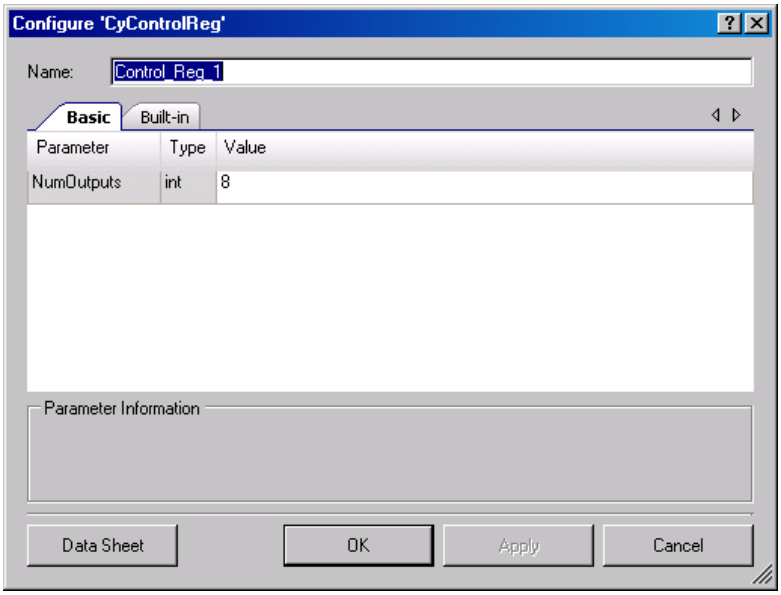
Transparent Status Read

By default, a CPU read of this register will transparently read the state of the associated routing net. This mode can be used for transient state that is computed and registered internally in the UDB.

Sticky Status, with Clear on Read

In this mode, the associated routing net is sampled on each cycle of the status and control clock and 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.

Configure Control Register Dialog



NumOutputs

Number of output terminals (1-8). The default value is 8. bit0 is the LSB and corresponds to the control_0 terminal.

Placement

There are no placement restrictions for the status or control registers.

Resources

The status register requires one UDB status register.

The control register requires one UDB control register.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following tables list and describe the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "Status_Reg_1" to the first instance of a status register and "Control_Reg_1" to the first instance of a control 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 names used in the following tables are "StatusReg" and "ControlReg".

Status Register

Function	Description
StatusReg_Read	Get the state of the status register's inputs (status register only).

uint8 StatusReg_Read (void)

Description: Read the value of a status register.

Parameters: None

Return Value: uint8: State of the status register's inputs.

Side Effects:

Control Register

Function	Description
ControlReg_Write	Set the state of the control register's outputs (control register only).
ControlReg_Read	Get the state of the control register's outputs (control register only).

void ControlReg_Write (uint8)

Description: Set the state of the control register's outputs.

Parameters: uint8: Control register value.

Return Value: None

Side Effects: Sets the state of the control register's outputs.

uint8 ControlReg_Read (void)

Description: Read the value of a control register.

Parameters: None

Return Value: uint8: State of the control register's inputs.

Side Effects: None

Sample Firmware Source Code

Status Register

The following is a C language example demonstrating the basic functionality of the status register component. This example assumes the component has been placed in a design with the default name "Status_Reg_1."

Note If you rename your component you must also edit the example code as appropriate to match the component name you specify.

```
#include <device.h>

void main()
{
    uint8 value;
    value = Status_Reg_1_Read();
}
```

Control Register

The following is a C language example demonstrating the basic functionality of the control register component. This example assumes the component has been placed in a design with the default name "Control_Reg_1."

Note If you rename your component you must also edit the example code as appropriate to match the component name you specify.

```
#include <device.h>

void main()
{
    uint8 value;
    Control_Reg_1_Write(0x3E);
    value = Control_Reg_1_Read();
}
```

Interrupt Service Routine

Not Applicable

Functional Description

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data.

5.0V/3.3V DC and AC Electrical Characteristics

Parameter	Typical	Min	Max	Units	Conditions and Notes
Input					
Maximum Clock Rate	---		67	MHz	