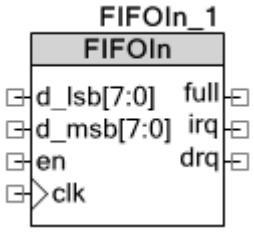
Parallel FIFO Input
1.0

Features

- 4-Entry FIFO to CPU or DMA
- Signaling for Interrupts, DMA and flow control
- 8 or 16-bit operation
- Selectable FIFO fill levels



General Description

Provides a mechanism to transfer parallel hardware signals to the CPU or DMA. Takes advantage of the built-in 4 entry FIFOs present in the UDB Datapaths to provide buffering and signaling.

When to use a Parallel FIFO Input

The component is used to provide a method to transfer a stream of data between hardware and CPU or DMA. To transfer non-streaming data the Status Register component can be used.

Input/Output Connections

This section describes the various input and output connections for FIFOIn. 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.

clk – Input

All data is captured on the rising edge of the **clk** input.

en – Input

The **en** signal enables the capture of the data input. The signal can either be a level sensitive signal or an edge sensitive signal. In level mode data is captured each clock when **en** is high. In edge mode data is captured on each clock after **en** has had a rising edge. In edge mode **en** must be low for one **clk** cycle before the rising edge.

d_lsb[7:0] – Input

Least significant 8-bits of data to capture.

d_msb[7:0] – Input *

Most significant 8-bits of data to capture. Displayed only in 16-bit mode.

full – Output

High when all 4 entries in the FIFO are populated. This signal can be used to provide flow control to the source of the data.

irq – Output

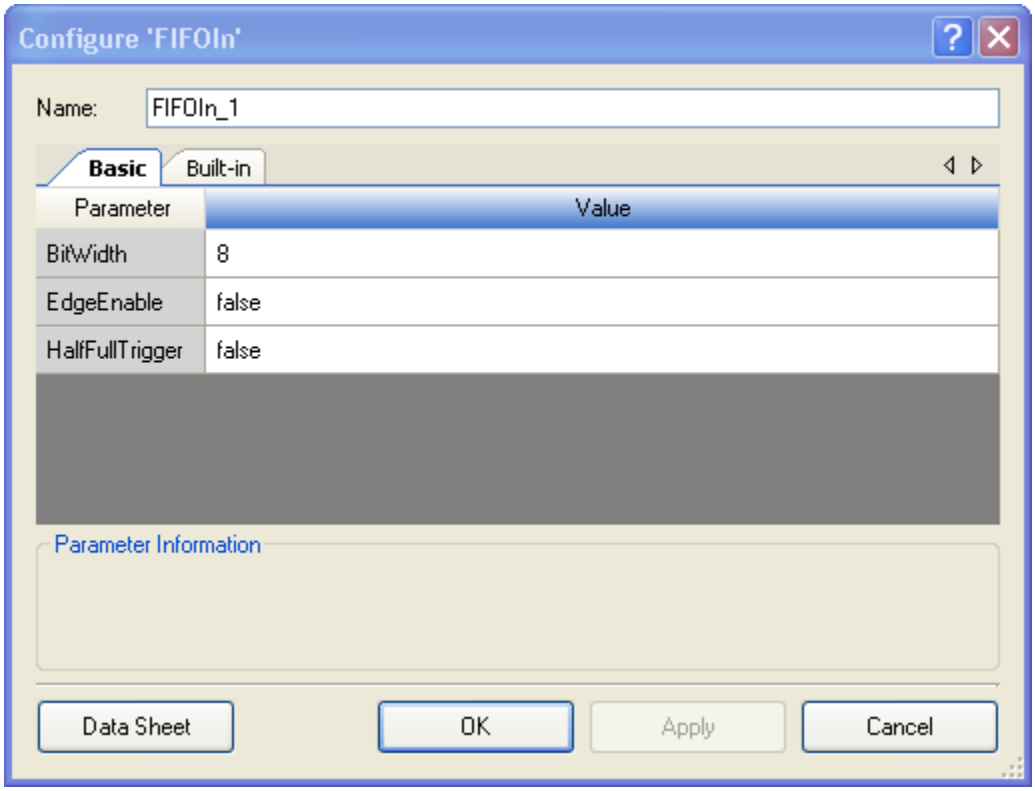
Interrupt request signal.

drq – Output

Level sensitive DMA signal.

Parameters and Setup

Drag a Parallel FIFO Input component onto your design and double-click it to open the Configure dialog.



BitWidth

Parallel data input width is either 8 (default) or 16 bits.

EdgeEnable

The enable input **en** is either level sensitive (default) or edge sensitive. Edge sensitive mode is used when only a single word should be captured on each rising edge of **en**.

HalfFullTrigger

The DMA request is active either when the FIFO is not empty (default) or when it is at least half full. The half full mode allows a DMA burst of 2 words from the FIFO without concern for underflow since it indicates that at least 2 words are available. Note that if only one word is present in the FIFO, then the DMA request line will not become active until another word arrives.

DMA

This component is often used with DMA to transfer data from hardware to a destination available on the Peripheral Hub (PHUB). Configuration of the DMA channel depends on the configuration of the component.

In 8-bit mode the following configuration should be used:

Name	Direction	Request Signal	Request Type	Burst Length	Description
FIFOIn_FIFO_PTR	Source	drq	Level	1 (HalfFullTrigger false) 2 (HalfFullTrigger true)	FIFO Source

In 16-bit mode the following configuration should be used:

Name	Direction	Request Signal	Request Type	Burst Length	Description
FIFOIn_FIFO16_PTR	Source	drq	Level	2 (HalfFullTrigger false) 4 (HalfFullTrigger true)	FIFO Source

Note: In 16-bit mode the 16-bit address space of the UDB is used to access the FIFO. This address space should not be read with byte transactions (burst length should be 2 or 4).

Note: The PSoC 3 Keil compiler is a big endian compiler and the 16-bit FIFO is read in little endian format. If DMA endian swapping is desired the TD_SWAP_EN option for the CyDmaTdSetConfiguration() function can be used. When using CPU reads instead of DMA (FIFOIn_Read) byte swapping is automatically handled.

Application Programming Interface

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

By default, PSoC Creator assigns the instance name “FIFOIn_1" to the first instance of a component in a given design. You can rename it 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 table is "FIFOIn".

Function	Description
void FIFOIn_Start(void)	Starts (Initializes and enables) the FIFO
void FIFOIn_Stop(void)	Stops (Disables and Clears) the FIFO
void FIFOIn_Init(void)	Initializes to the customizer settings
void FIFOIn_Enable(void)	Enables capture

uint8/uint16 FIFOIn_Read(void)	Read one word from the FIFO
uint8 FIFOIn_ReadStatus(void)	Reads the status register
void FIFOIn_SetInterruptMode(uint8 interruptSource)	Sets the interrupt mask
void FIFOIn_ClearFIFO(void)	Clears the FIFO
void FIFOIn_Sleep(void)	Saves configuration and disables
void FIFOIn_Wakeup(void)	Restores configuration and enables
void FIFOIn_SaveConfig(void)	Saves the configuration
void FIFOIn_RestoreConfig(void)	Restores the configuration

void FIFOIn_Start(void)

Description: Starts the FIFO. Initializes the FIFO fill level mode and enables capture.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_Stop(void)

Description: Stops the FIFO. Disables capture of data and clears the FIFO.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_Init(void)

Description: Initializes to the customizer settings.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_Enable(void)

Description: Enables the capturing of data. To capture data the component must be enabled, the hardware enable signal (en) must be active, a clock must occur and the FIFO must not be full.

Parameters: None

Return Value: None

Side Effects: None

uint8/uint16 FIFOIn_Read(void)

Description: Reads a single word from the FIFO. The fill level of the FIFO is not checked by this function. The current fill level must be read using FIFOIn_ReadStatus() to determine if data is available.

Parameters: None

Return Value: (uint8) in 8-bit mode and (uint16) in 16-bit mode. One entry from the FIFO.

Side Effects: None

uint8 FIFOIn_ReadStatus(void)

Description: Reads the status register.

Parameters: None

Return Value: (uint8) state of the status register.

Status Masks	Value	Type
FIFOIn_FIFO_OVERFLOW	0x01	Clear on Read
FIFOIn_FIFO_NOT_EMPTY	0x02	Transparent

Side Effects: All Clear on Read status bits are cleared.

void FIFOIn_SetInterruptMode(uint8 interruptSource)

Description: Sets the interrupt mask register.

Parameters: (uint8) byte containing the constant with the following mask fields set.

Interrupt Source	Value
FIFOIn_FIFO_OVERFLOW	0x01
FIFOIn_FIFO_NOT_EMPTY	0x02

Return Value: None

Side Effects: None

void FIFOIn_ClearFIFO(void)

Description: Clears the contents of the FIFO.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_Sleep(void)

Description: Saves the configuration and non-rentention register values. Disables and clears the FIFO.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_Wakeup(void)

Description: Restores the configuration and non-retention register values. Enables depending on the state before going to sleep.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_SaveConfig(void)

Description: Saves the user configuration of non-rentention registers. Called by FIFOIn_Sleep routine to save the component state before entering sleep.

Parameters: None

Return Value: None

Side Effects: None

void FIFOIn_RestoreConfig(void)

Description: Restores the user configuration of non-rentention registers. Called by FIFOIn_Wakeup routine to restore the component state after returning from sleep.

Parameters: None

Return Value: None

Side Effects: None