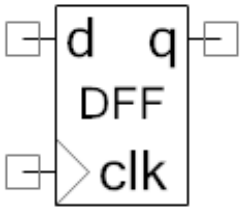
D Flip Flop
1.30



Features

- Asynchronous reset or preset
- Synchronous reset, preset, or both
- Configurable width for array of D Flip Flops

General Description

The D Flip Flop stores a digital value.

When to Use a D Flip Flop

Use the D Flip Flop to implement sequential logic.

Input/Output Connections

This section describes the various input and output connections for the D Flip Flop. An asterisk (*) in the list of I/Os states that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

d – Input

This input determines the next value of the output. The output does not change until the next rising edge of the clock.

clock – Input

The clock signal determines when the output will change. The output changes when a rising edge of the clock is detected.

ar – Input *

Asynchronous reset. When this input is true, the output immediately changes to false without waiting for the positive edge of the clock. Asynchronous reset functions regardless of the status of the clock signal. This input only appears if the **PresetOrReset** parameter is set to **Asynchronous Reset**.

ap – Input *

Asynchronous preset. When this input is true, the output immediately changes to true without waiting for the positive edge of the clock. Asynchronous preset functions regardless of the status of the clock signal. This input only appears if the **PresetOrReset** parameter is set to **Asynchronous Preset**.

sr – Input *

Synchronous reset. When this input is true, the output changes to false on the positive edge of the clock. This input only appears if the **PresetOrReset** parameter is set to **Synchronous Reset** or **Synchronous Preset & Reset**.

sp – Input *

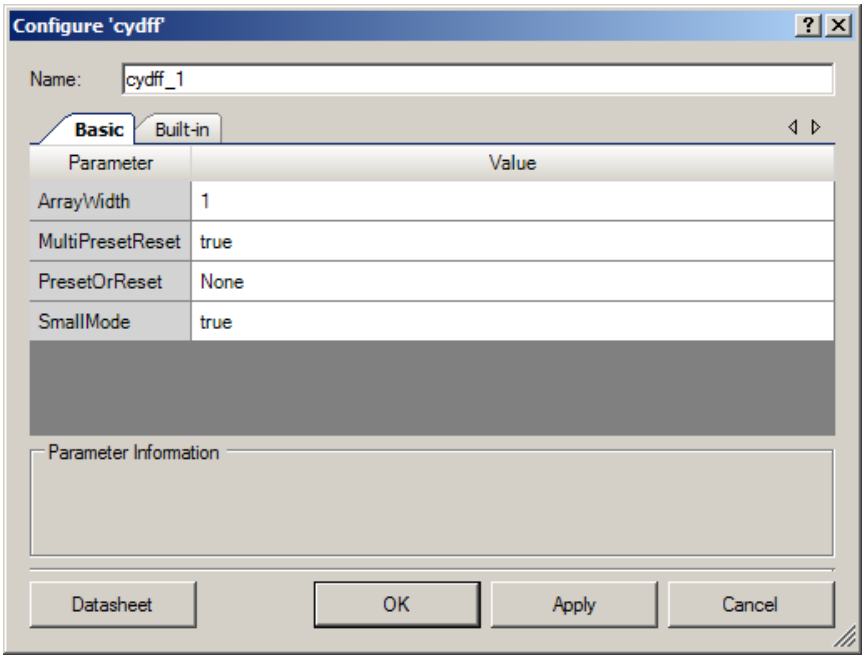
Synchronous preset. When this input is true, the output changes to true on the positive edge of the clock. This input only appears if the **PresetOrReset** parameter is set to **Synchronous Preset** or **Synchronous Preset & Reset**.

q – Output

The stored value of the D Flip Flop.

Component Parameters

Drag a D Flip Flop onto your design and double-click it to open the **Configure** dialog.



The D Flip Flop provides the following parameters.

ArrayWidth

You can create an array of D Flip Flops, which is useful if the input or output is a bus. This parameter defines the bus width of the d and q terminals. The value must be between 1 and 32. The default is **1**.

MultiPresetReset

This parameter controls whether the preset and reset inputs are implemented as a bus the size of **ArrayWidth** (if **true**) or are implemented as a single bit (if **false**).

PresetOrReset

This parameter controls whether the asynchronous preset (ap) input, asynchronous reset (ar), synchronous preset (sp) input, or synchronous reset (sr) is visible. The default is **None**.

SmallMode

This parameter controls the size of the component's symbol on the schematic. The default is **true**.

Functional Description

The D Flip Flop is implemented in PLD macrocells. All macrocell flip-flops are initialized to a 0 value at power up and after any reset of the device.

Asynchronous Preset and Reset is implemented directly in the macrocell.

Synchronous Preset is implemented using the following logical equation implemented in macrocell product terms:

Q = D | SP

Synchronous Reset is implemented using the following logical equation implemented in macrocell product terms:

Q = D & ~SR

Synchronous Preset & Reset is implemented using the following logical equation implemented in macrocell product terms:

Q = (D | SP) & ~SR

Note that Reset dominates Preset when Synchronous Preset & Reset is used.

Table 1. 1-ArrayWidth D Flip Flop Truth Table

Preset	Reset	D	Q
0	-	0	0
0	-	1	1
1	-	X	1
-	0	0	0
-	0	1	1
-	1	X	0
1	1	X	0

A letter ‘X’ in the truth table indicates that the input does not affect the output.

Resources

The D Flip Flop uses one macrocell. If the ArrayWidth parameter is greater than 1, the D Flip Flop uses a number of macrocells equal to ArrayWidth. All D Flip Flop components in the same PLD must have the same clock signal for clocking.

MISRA Compliance

This section describes the MISRA-C:2004 compliance and deviations for the component. There are two types of deviations defined:

- project deviations – deviations that are applicable for all PSoC Creator components
- specific deviations – deviations that are applicable only for this component

This section provides information on component-specific deviations. Non PSoC 6 project deviations are described in the MISRA Compliance section of the *System Reference Guide* along with information on the MISRA compliance verification environment. For PSoC 6, refer to PSoC Creator Help > Building a PSoC Creator Project > Generated Files (PSoC 6) for information on MISRA compliance and deviations for files generated by PSoC Creator.

The D Flip Flop component does not have any C source code APIs.

DC and AC Electrical Characteristics

The D Flip Flop component supports the maximum device frequency.

Component Changes

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

Version	Description of Changes
1.30.b	Updated MISRA section. Added PSoC 6 support.
1.30.a	Minor datasheet update.
1.30	Added Synchronous Preset & Reset option Added SmallMode parameter Added Misra Compliance section.
1.20.b	Minor datasheet edits and updates
1.20.a	Minor datasheet edits and updates
1.20	Added Synchronous Reset and Synchronous Preset option
	Added MultiPresetReset parameter
	Minor datasheet edits and updates
1.10	Replaced NeedAP and NeedAR parameters with PresetorReset parameter.

© Cypress Semiconductor Corporation, 2012-2017. 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, 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.