LogiCORE™ FIFO Generator v2.3

Release Notes
January 18, 2006

General Usage Information

FIFO Generator Guidelines

- The FIFO Generator supports data widths from 1 to 256 bits, and data depths up to 4,194,304 words.

- When the FIFO is empty and a read is requested, the read operation is ignored, the underflow flag is asserted and there is no change in the state of the FIFO (i.e. underflowing the FIFO is non-destructive.)

- When the FIFO is full and a write is initiated, the request is ignored, the overflow flag is asserted and there is no change in the state of the FIFO (i.e. overflowing the FIFO is non-destructive.)

Independent Clocks Guidelines

- For independent clocks FIFO implementations, the full and empty flags are pessimistic flags. Refer to Chapter 6 Special Design Considerations in the FIFO Generator User Guide for details.

- For independent clocks FIFO implementations, the write and read depths of the FIFO will vary slightly from the depth selected in the GUI. Refer to page 6 of the FIFO Generator GUI for the actual depths of the FIFO implemented.

Reset Requirements

- There are no reset requirements for block RAM or distributed RAM based FIFO implementations. In these configurations, the FIFO is guaranteed to power-on correctly after the FPGA is configured. This initial power-on state is also correctly modeled in both the behavioral and structural simulation models. The reset input (RST) is an optional pin that can be disabled in the GUI.

- For shift register based FIFO configurations, the FIFO Generator must be reset before use with a pulse three or more clock cycles in length. Refer to Chapter 5 Designing with the Core Independent Clocks: Block RAM, Distributed RAM, Shift Register FIFO Reset Requirements in the FIFO Generator User Guide for details.

- For built-in FIFO configurations, the FIFO Generator must be reset before use with a pulse three or more clock cycles in length. Read or write operations can not occur until four clock cycles after reset is released. Refer to Chapter 5 Designing with the Core Independent Clocks: Built-in FIFO Reset Requirements in the FIFO Generator User Guide for details.

Supported Features

All FIFO Generator configurations feature a variety of handshaking and status signals. Features that are not supported in all configurations are outlined in the table below.

FIFO Feature	Independent Clocks FIFO			Common Clock FIFO			
	Block RAM	Distributed RAM	Built-in FIFO	Block RAM	Distributed RAM	Shift Register	Built-in FIFO
Non-symmetric Aspect Ratios	Yes	No	No	No [1]	No	No	No
Almost Full/Empty	Yes	Yes	No	Yes	Yes	Yes	No
Data Count	Yes	Yes	No	Yes	Yes	Yes	No
First-Word Fall-Through	Yes	Yes	No	No [1]	No [1]	No	No
DOUT Reset Value	Yes [2]	Yes	No	Yes [2]	Yes	Yes	No

[1] For applications with a single clock, use the independent clocks FIFO configuration and connect the write and read clocks to the same source. A dedicated solution for common clock FIFO will be available in a future release. Contact your Xilinx representative for more details.
[2] All architectures except for Virtex, Virtex-E, Spartan-II, and Spartan-IIE.

Additional FIFO Generator Features

- **Non-symmetric Aspect Ratios**: For independent clocks FIFOs implemented with block RAM, write and read ports of different widths are supported for ratios ranging from 1:8 to 8:1. Refer to Chapter 5 Independent Clocks: Block RAM and Distributed RAM Non-symmetric Aspect Ratios in the FIFO Generator User Guide for details.

- **Built-In FIFOs**: In the Virtex-4 architecture, dedicated logic in the block RAM enables users to implement FIFOs with no additional logic required. The FIFO Generator supports and augments the Virtex-4 built-in FIFO modules in the following ways (refer to Chapter 5 Independent Clocks: Built-In FIFO in the FIFO Generator User Guide for details):
    - Cascading built-in FIFOs in both width and depth to implement larger FIFOs
    - Using additional FPGA fabric resources to provide status flags not available in the built-in FIFO

- **First-Word Fall-Through (FWFT):** For independent clocks FIFOs implemented with block or distributed RAM, the FWFT feature provides the ability to look-ahead to the next word available from the FIFO without issuing a read operation. When data is available in the FIFO, the first word falls through the FIFO and appears automatically on the output bus (DOUT). FWFT is useful in applications that require low latency access to data and applications that require throttling based on data content. Refer to Chapter 5 Independent Clocks: Block RAM and Distributed RAM Read Operation in the FIFO Generator User Guide for details.

- **Programmable Flags:** The FIFO Generator supports programmable flags to indicate that the FIFO has reached a user defined fill level or threshold. The user can set the thresholds to a constant value or the thresholds can be connected to dedicated input ports, enabling the thresholds to change dynamically in circuit. Refer to Chapter 5 Programmable Flags in the FIFO Generator User Guide for details.

Port Summary

All ports of the FIFO Generator are listed in the table below. For more information on these ports, refer to Chapter 3 Core Architecture in the FIFO Generator User Guide.

Port Name	Input or Output	Port is Optional	Port Available	
			Independent Clocks	Common Clock
RST	I	Yes	Yes	Yes
CLK	I	No	No	Yes
DATA_COUNT[C:0]	O	Yes	No	Yes
Write Interface Signals				
WR_CLK	I	No	Yes	No
DIN[N:0]	I	No	Yes	Yes
WR_EN	I	No	Yes	Yes
FULL	O	No	Yes	Yes
ALMOST_FULL	O	Yes	Yes	Yes
PROG_FULL	O	Yes	Yes	Yes
WR_DATA_COUNT[D:0]	O	Yes	Yes	No
WR_ACK	O	Yes	Yes	Yes
OVERFLOW	O	Yes	Yes	Yes
PROG_FULL_THRESH	I	Yes	Yes	Yes
PROG_FULL_THRESH_ASSERT	I	Yes	Yes	Yes
PROG_FULL_THRESH_NEGATE	I	Yes	Yes	Yes
Read Interface Signals				
RD_CLK	I	No	Yes	No
DOUT[M:0]	O	No	Yes	Yes
RD_EN	I	No	Yes	Yes
EMPTY	O	No	Yes	Yes
ALMOST_EMPTY	O	Yes	Yes	Yes
PROG_EMPTY	O	Yes	Yes	Yes
RD_DATA_COUNT[C:0]	O	Yes	Yes	No
VALID	O	Yes	Yes	Yes
UNDERFLOW	O	Yes	Yes	Yes
PROG_EMPTY_THRESH	I	Yes	Yes	Yes
PROG_EMPTY_THRESH_ASSERT	I	Yes	Yes	Yes
PROG_EMPTY_THRESH_NEGATE	I	Yes	Yes	Yes

Known Issues

If using the Virtex-4 built-in FIFO with common or independent clocks, please refer to Answer Record 22462 before using the built-in FIFO.

For a list of the current known issues, please refer to Answer Record 22302.

Technical Support

The fastest method for obtaining specific technical support for the FIFO Generator is through the http://support.xilinx.com/ website. Questions are routed to a team of engineers with specific expertise in using the FIFO Generator. Xilinx will provide technical support for use of this product as described in the FIFO Generator User Guide. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.