ABSTRACT

This paper will discuss the design of an asynchronous FIFO as implemented into Quicklogic’s QuickRAM family.

INTRODUCTION

Asynchronous FIFOs are widely used in the computer networking industry to receive data at a particular frequency and transmit them at another frequency. An asynchronous FIFO has two different clocks: one for read and one for write.

There are issues that arise when passing data over asynchronous clock boundaries. For example, when the write clock is faster than the read clock, data could be overwritten and hence lost. In order to overcome these problems, control signals like almost-empty, almost-full, empty, and full flags are required.

FUNCTIONAL DESCRIPTION

When the FIFO is full (or not empty), the POP signal indicates that the data will be read out of the RAM location indicated by the read address register, and the read address register will be incremented. When the read address register reaches the write address register, the FIFO is empty and the empty flag is active. The empty flag stays active as long as this condition is true.

When the FIFO is empty (or not full), the PUSH signal indicates that the data will be written to the RAM location indicated by the write address register, and the write address register will be incremented. When the write address register reaches the read address register the FIFO is full and the full flag is active. The full flag stays active as long as this condition is true.

The almost-empty flag is activated two locations before the FIFO is actually empty. Similarly, the almost-full flag is activated three locations before the FIFO is actually full. The user can initialize the read side and the write side counters to any desired value, e.g. if the counters are initialized to 7, the almost-full and almost-empty flags are activated 7 locations before the FIFO is full and empty respectively. The read side and the write side state machines receive the outputs of the comparators to determine the status of the flags. These state machines have two D-flipflops each in order to account for metastability.

Keeping in mind the necessity for asynchronous FIFOs, QuickLogic generated a code for a 32x32 asynchronous FIFO. The current asynchronous FIFO design is for a 32 wide by 32 deep FIFO implemented using RAM blocks in the QuickRAM family of devices. It is essentially a schematic (F32a32.sch). In order to configure the FIFO with a different width or depth, some modifications need to be made.
ILLUSTRATION

The figure below shows the file names corresponding to the relevant building blocks in the schematic. All the blocks shown in the figure can be scaled to meet the users' width and depth requirements. Please note that this figure does not show the read side and write side state machines.

RAM Block
In the enclosed schematic, a 64X32 RAM block symbol is used. This block was created from Verilog code generated by the RAM/ROM/FIFO Wizard in SpDE. The user can specify the required width and depth of the RAM block in the wizard, which generates the Verilog/VHDL code. Using the "New Block Symbol" in the Schematic Tools, you can create a schematic symbol for the Verilog code.

COMPARATOR
The 5-bit comparators shown in the top level schematic are purely schematic designs generated in the schematic editor. The user must note that the number of bits to be compared would change with the width of the RAM block used. For example, if the user
defined FIFO depth is 256, 8 address bits are required and hence an 8-bit comparator must be used.

GREY COUNTERS
The Grey code counters shown in the top level schematic are Verilog modules written for a 5-bit Grey counter. This code would have to be modified to 6-bits, 7-bits, 8-bits, and 9-bits for FIFO depths of 64, 128, 256, and 512 respectively. Writing a Verilog module for a Grey counter can be cumbersome, therefore we have provided an example that can generate the Grey code for such wide counters.

REGISTERS
The registers are Verilog modules, which can be easily modified to fit the user’s depth requirements. There are three registers for the write address and one for the read address. These are provided for the determination of the FIFO status i.e. Almost-Full, Almost-Empty, Full, and Empty.

PERFORMANCE
PART: QL4090 (QuickRAM family)
AREA: 7.1% (48 of 672 buffered cells)
SPEED: Write Clock (WCLK) = 136 MHz, Read Clock (RCLK) = 129 MHz

Conclusion
The implementation of the asynchronous FIFO in the QuickLogic QuickRAM family is fast and simple. The embedded RAM blocks keep the logic cell utilization to a minimum. The asynchronous FIFO design is scalable to user-defined widths and depths thereby making it very useful to designers with unconventional FIFO dimensions.