Formal Verification of an Asynchronous FIFO in SymbiYosys

Shidi Xi
Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver V6T 1Z4, Canada
xsd99@ece.ubc.ca

***Abstract*—yo**

I. Introduction

Asynchronous FIFO is a hardware buffer that allows data to be written and read at different clock speeds. It is vital for preventing data loss or corruption when crossing clock domains. SymbiYosys is an open-source tool for hardware formal verification. Its verification process involves two methods. One methods is bounded model check (BMC), which is a technique that uses SAT or SMT solvers to check the correctness of a system within a predefined number of transitions. The second is k-induction. To prove system properties, it first demonstrates that they hold in a base case (the BMC), and then it checks for future steps via inductive reasoning. In this project, an asynchronous FIFO was designed in SystemVerilog and subsequently formally verified in SymbiYosys. Section description.

II. ASYNCHRONOUS FIFO

A. IP Functionalities

Figure 1 presents the schematic of the asynchronous FIFO designed in this project. Table I lists its IOs. In Figure 1, there are 3 major components in the FIFO:

- **Dual-port RAM**: Stores the data.
- **Write and read pointer handlers**: Controls the pointers, generates their Gray codes (*g_wptr* and *g_rptr*) and the *full* and *empty* flags.
- **Synchronizers**: Takes the write pointer in Gray code into the read clock domain and vice versa.

On positive edge of *wclk*, if *full* is 0, and *w_en* is 1, *data_in* will be written to the address indicated by the binary write pointer *b_wptr*, followed by its increment. *data_out* will always have the data at *b_rptr*, and on positive edge of *rclk*, if *empty* is 0 abd *r_en* is 1, a read transaction will complete and *b_rptr* will increment. The FIFO is empty when *g_rptr* equals *g_wptr_sync* and is full when the two most significant bits of *g_wptr* is the complement of *g_rptr_sync* and the rest bits are equal.

III. FORMAL VERIFICATION IN SYMBIYOSYS

In SymbiYosys, formal verification can be either done using BMC or k-induction. Both methods require us to define the legal and the illegal states in our design.

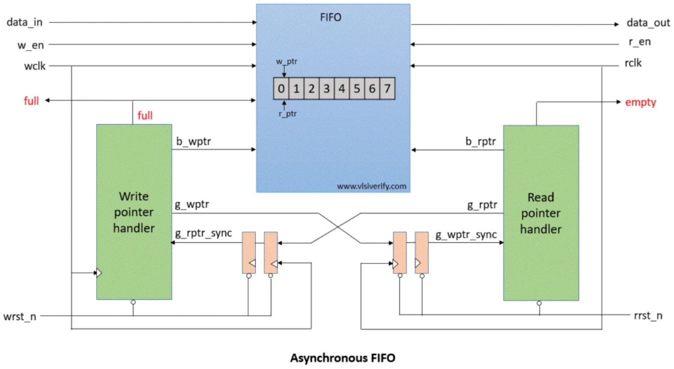

Fig. 1: Schematic of the asynchronous FIFO designed in this project (REF:https://vlsiverify.com/verilog/verilog-codes/asynchronous-fifo/).

TABLE I: List of IOs of the asynchronous FIFO.

Signal	I/O	Width	Description
wclk	Input	1	Write clock signal.
wrst_n	Input	1	Negative-high write reset signal.
data_in	Input	DATA_WIDTH	Input data to be written.
w_en	Input	1	Write enable.
full	Input	1	Full flag.
rclk	Input	1	Read clock signal.
rrst_n	Input	1	Negative-high read reset signal.
data_out	Input	DATA_WIDTH	Output data to be read.
r_en	Input	1	Read enable.
empty	Input	1	Empty flag.

IV. METHODOLOGY

V. Results

VI. Conclusion