Synchronous FIFO Buffer Memory

Shirsendu B Acharyya
Catch the Bug Hackathon
NIELIT Calicut, IIT Madras,Vyoma Systems, VLSI System Design
Kolkata, India
acharyya.nelson@gmail.com

***Abstract*—This paper describes the design and working of synchronous first in first out buffer Queue memory. We provide the input data and other required control signals in the input side and in output we expect to memory to store the definite amount of input as provided and produce them as output whenever the appropriate input control signal is applied. The closest operation can resemble to program counter in computer architecture. Here the synchronous refers to all the control action, input and output occurs at positive edge of clock.**

Keywords—synchronous, FIFO, queue, control signal, memory.

I. DESCRIPTION

The block diagram can be imagined like-.

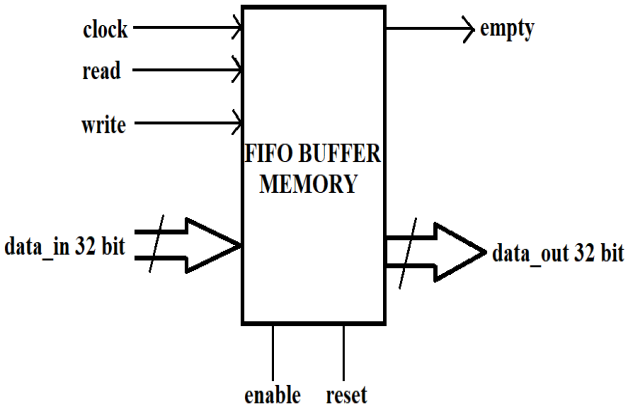

Fig. 1. Block Diagram

The "clock, read, write, enable, reset" is the one-bit input. To be noted all the signal is active high. The FIFO memory consist of 8 word (depth) and each of the word is of 32 bits, i.e. "reg [31:0] memory [0:7]". Other than that, there are internal registers like "count" to keep track the depth of the memory, so quite obviously when count is zero the output "empty" signal goes high. From the general knowledge we know in queue memory the reading and writing operation occurs through different position and to satisfy the condition of FIFO two more counters are used they are "read_pointer" and "write_pointer". When the data is to be written into the memory the "write" is made high, again when its time to read the memory then make "read" signal high and output will be available.

II. WORKING AND RESULTS

A. *Working with Input and Output*

Let us take a sample input function as "Input ()" and the status of memory can be observed as output

For Input (0,1,2,3,4,5,6,7) The memory status is -

0	1	2	3	4	5	6	7

For Input (8) The memory status is -

1	2	3	4	5	6	7	8

For Input (9) The memory status is -

2	3	4	5	6	7	8	9

As new data is fed to the memory the oldest data is overwritten as soon as the 8 words are completely filled. This mechanism is achieved by **circularly travelling** the "write_pointer" from the last index [7] of memory to first index position [0]; Similar **circular travelling** is also followed during the reading mechanism, so that we get continuous repetition of output as long as the "read" signal is active.

B. *Expected Output Waveform*

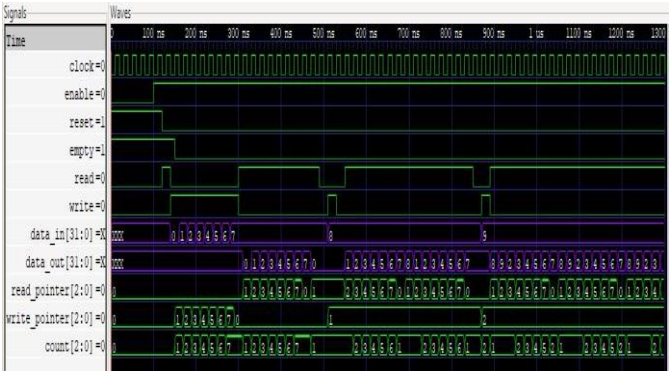

Fig. 2. Output waveform

III. LITERATURE SURVEY REFERENCE

1. Verilog Synchronous FIFO https://github.com/desirajusantosh/FIFO
2. Verilog for Beginners https://esrd2014.blogspot.com