Spring 2021

California State University, Northridge Department of Electrical & Computer Engineering

Experiment 9
Synchronous FIFO

April 24, 2021

ECE 526L

Written By: Jose Luis Martinez

Introduction

In this lab we are to design a Synchronous FIFO with parameterized width, depth, and almost triggers based on the model in **Fig. 1**. FIFO stands for first in first out and it is a form of buffer that has a certain amount of registers specified by depth. The FIFO will either read or write on the positive edge of the clock depending if they are enabled. The FIFO will present its contents on the data_out line in the order that it arrived on data_in. The FIFO will have empty, almost_empty, full, almost_full, valid, overflow, and underflow flags to show its current state. The FIFO will also specify how much data is stored in its registers on the count line.

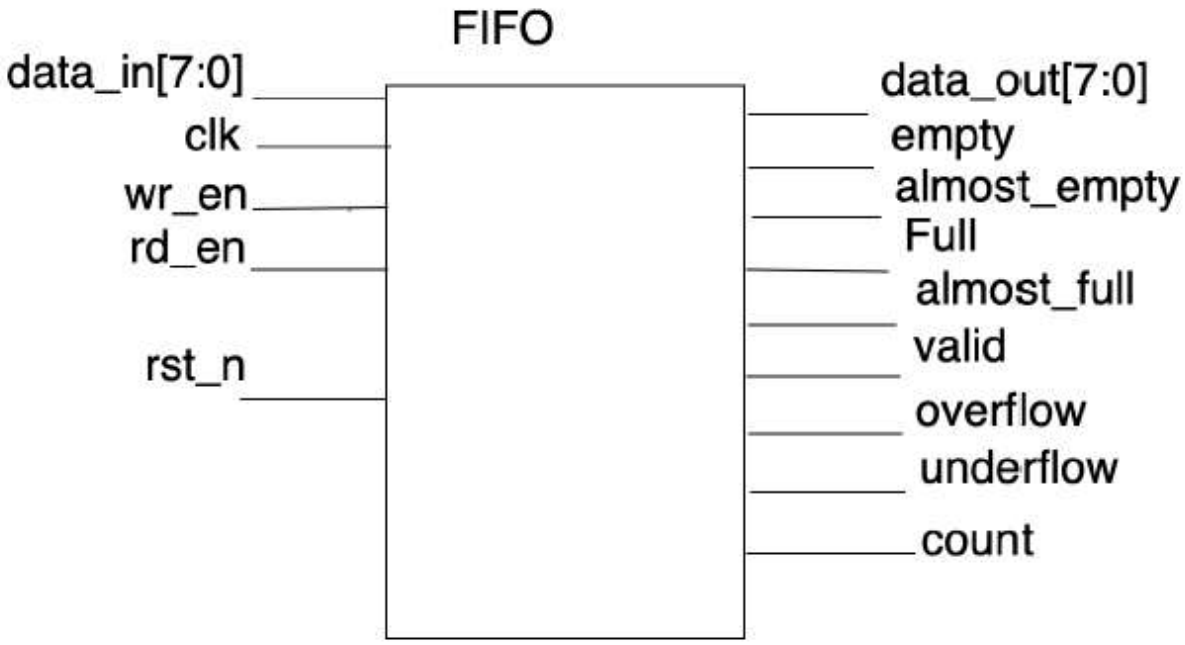

Fig. 1 FIFO Model

Methodology

I followed these steps in order to complete the lab:

1. Sync_fifo.v was created as shown in **Fig. 2**.
    a. An initial block is there to set everything to 0.
    b. The first always block is there to control the reset, writing, and reading logic.
        i. The writing portion checks if wr_en is on and if it is it will check if the FIFO stack is full. If full it will not write to the stack and set the flag to 1'b1. If not full data_in will be written to the fifo registers depending on wr_ptr's value and increment write count by one. Additionally if FWFT is defined during compilation then if FIFO is empty data_in will be set to the output.
        ii. The reading portion checks if rd_en is on and if the FIFO is empty. If it is empty it will not read the next item in the FIFO and set the uf flag to 1'b1. Otherwise the valid signal will be set to 1'b1 and data_out will be set to the value of whatever the rd_ptr points to and increment readcount by 1.
2. Lab9_tb.v was created as shown in **Fig. 3.**

   a. A portion of the testbench file tests for the FWFT whenever it is defined during compilation and the other test without FWFT.
   b. For both portions various for loops were used to test all possible test cases.
3. Compiled normally.
   a. Ran the simv command in the terminal then recorded the results as shown in **Fig. 4.**
   b. Ran the dve command and made a new wave view with the desired signals and then recorded the waveforms as shown in **Fig. 6.**
4. Compiled with +define+FWFT.
   a. Ran the simv command in the terminal then recorded the results as shown in **Fig. 5.**
   b. Ran the dve command and made a new wave view with the desired signals and then recorded the waveforms as shown in **Fig. 7.**

Results/Verilog Files

```
/*****************************************************************************
***                                                                        ***
*** ECE 526 L Experiment #9            Jose Luis Martinez, Spring, 2021    ***
***                                                                        ***
*** Experiment 9 - Parameterized Synchronous FIFO                          ***
***                                                                        ***
*****************************************************************************
*** Filename: sync_fifo.v     Created by Jose Luis Martinez, April 28, 2021 ***
***                                                                        ***
*****************************************************************************/

module sync_fifo(data_in, clk, wr_en, rd_en, rst, data_out,
                 empty, a_e, full, a_f, valid, of, uf, count);
      parameter WIDTH = 8;
      parameter DEPTH = 32;
      parameter ALMOST = 2;
      input clk, wr_en, rd_en, rst;
      input [WIDTH-1: 0]data_in;
      output reg empty, a_e, full, a_f;
      output reg of, uf, valid;
      output reg [$clog2(DEPTH):0] count;
      output reg [WIDTH-1:0]data_out;

      integer i;

      reg [WIDTH-1:0] fifo_data [0:DEPTH-1];
      integer writecount, readcount;
      reg [$clog2(DEPTH):0] wr_ptr, rd_ptr;

      initial begin
            data_out <= {WIDTH-1{1'b0}};
            of <= 0;
            uf <= 0;
            writecount <= {$clog2(DEPTH){1'b0}};
            readcount <= {$clog2(DEPTH){1'b0}};
      end
```

```
	always@(posedge clk, negedge rst) begin
		valid <= 0;
		if(!rst) begin
			writecount <= 0;
			readcount <= 0;
		end else begin

			if(wr_en) begin
				if(full) begin of <= 1; end
				else begin
					`ifdef FWFT
					if(empty) data_out <= data_in;
					`endif
					fifo_data[wr_ptr] <= data_in;
					writecount <= writecount + 1;
				end
			end

			if(rd_en) begin
				if(empty) begin uf <= 1; end
				else begin
					valid <= 1;
					data_out <= fifo_data[rd_ptr];
					readcount <= readcount + 1;
				end
			end
		end
	end

	always@(readcount, writecount) begin
		wr_ptr <= writecount%DEPTH;
		rd_ptr <= readcount%DEPTH;

		full <= (writecount - readcount) == DEPTH;
		empty <= (writecount - readcount) == 0;
		a_f <= (writecount - readcount) > DEPTH - ALMOST - 1;
		a_e <= (writecount - readcount) < ALMOST + 1;
		count <= writecount - readcount;
	end

endmodule
```

Fig. 2 sync_fifo.v

```
/**************************************************************************
***                                                                     ***
*** ECE 526 L Experiment #9            Jose Luis Martinez, Spring, 2021   ***
***                                                                     ***
*** Experiment 9 - Parameterized Synchronous FIFO                       ***
***                                                                     ***
```

```
/*****************************************************************************
*** Filename: lab9_tb.v       Created by Jose Luis Martinez, April 28, 2021 ***
***                                                                         ***
*****************************************************************************/

module lab9_tb();
	reg clk, wr_en, rd_en, rst;
	reg [`WIDTH_TB-1:0] data_in;
	wire [`WIDTH_TB-1:0]data_out;
	wire empty, a_e, full, a_f, valid, of, uf;
	wire [$clog2(`DEPTH_TB):0] count;

	integer i;

	sync_fifo sf1(.data_in(data_in), .clk(clk), .wr_en(wr_en), .rst(rst),
			.rd_en(rd_en), .data_out(data_out), .empty(empty), .a_e(a_e),
			.full(full), .a_f(a_f), .valid(valid), .of(of), .uf(uf),
			.count(count));

	initial begin
		clk <= 0;
		forever begin
			#(`CLK_PER/2) clk <= ~clk;
		end
	end

	initial begin
		$vcdpluson;
		wr_en <= 1;
		rd_en <= 0;
		rst <= 1;
		`ifdef FWFT
		data_in <= 5;
		$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
					empty, a_e, full, a_f, valid, of, uf, count);

		for (i=0; i<`DEPTH_TB; i = i+1) begin
			#(`CLK_PER)
			data_in <= i;
			$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
					empty, a_e, full, a_f, valid, of, uf, count);
		end

		wr_en <= 0;
		rd_en <= 1;

		for (i=0; i<`DEPTH_TB; i = i+1) begin
			#(`CLK_PER)
			$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
					empty, a_e, full, a_f, valid, of, uf, count);
		end
```

```
#(`CLK_PER)
$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);

`else
data_in <= 0;
$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);

for (i=0; i<`DEPTH_TB; i = i+1) begin
      #(`CLK_PER)
      data_in <= i;
      $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);
end

wr_en <= 0;
rd_en <= 1;

for (i=0; i<`DEPTH_TB; i = i+1) begin
      #(`CLK_PER)
      $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);
end

#(`CLK_PER)
$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);

for (i=0; i<`DEPTH_TB; i = i+1) begin
      #(`CLK_PER)
      $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);
end

wr_en <= 1;
rd_en <= 0;

#(`CLK_PER)
$strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                       empty, a_e, full, a_f, valid, of, uf, count);
rd_en <= 1;

for (i=0; i<`DEPTH_TB; i = i+1) begin
      #(`CLK_PER)
      data_in <= i;
      $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                          empty, a_e, full, a_f, valid, of, uf, count);
end

#(`CLK_PER)
rd_en <= 0;
```

```
        for (i=0; i<`DEPTH_TB; i = i+1) begin
            #(`CLK_PER)
            data_in <= i;
            $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                                empty, a_e, full, a_f, valid, of, uf, count);
        end

        #(`CLK_PER)
        $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                                empty, a_e, full, a_f, valid, of, uf, count);

        #(`CLK_PER)
        $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                                empty, a_e, full, a_f, valid, of, uf, count);

        #(`CLK_PER)
        $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                                empty, a_e, full, a_f, valid, of, uf, count);

        #(`CLK_PER)
        $strobe(`STRING, $time, data_in, clk, wr_en, rd_en, rst, data_out,
                                empty, a_e, full, a_f, valid, of, uf, count);
        `endif
        $finish;
    end

endmodule
```

Fig. 3 lab9_tb.v

2. dcd142.ecs.csun.edu (jlm777)

```
$ simv -l lab9.log
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-2_Full64; Runtime version N-2017.12-SP2-2_Full64;  May  1 22:23 2021
VCD+ Writer N-2017.12-SP2-2_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
                   0, data_in = 00000000, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 1, a_e = 1, full = 0, a_f = 0, valid = x, of = 0, uf = 0, count =  0
                  20, data_in = 00000000, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  1
                  40, data_in = 00000001, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  2
                  60, data_in = 00000010, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  3
                  80, data_in = 00000011, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  4
                 100, data_in = 00000100, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  5
                 120, data_in = 00000101, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  6
                 140, data_in = 00000110, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  7
                 160, data_in = 00000111, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  8
                 180, data_in = 00001000, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  9
                 200, data_in = 00001001, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 10
                 220, data_in = 00001010, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 11
                 240, data_in = 00001011, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 12
                 260, data_in = 00001100, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 13
                 280, data_in = 00001101, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 14
                 300, data_in = 00001110, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 15
                 320, data_in = 00001111, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 16
                 340, data_in = 00010000, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 17
                 360, data_in = 00010001, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 18
                 380, data_in = 00010010, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 19
                 400, data_in = 00010011, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 20
                 420, data_in = 00010100, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 21
                 440, data_in = 00010101, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 22
                 460, data_in = 00010110, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 23
                 480, data_in = 00010111, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 24
                 500, data_in = 00011000, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 25
                 520, data_in = 00011001, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 26
                 540, data_in = 00011010, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 27
                 560, data_in = 00011011, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 28
                 580, data_in = 00011100, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count = 29
                 600, data_in = 00011101, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 1, valid = 0, of = 0, uf = 0, count = 30
                 620, data_in = 00011110, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 1, valid = 0, of = 0, uf = 0, count = 31
                 640, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 1, a_f = 1, valid = 0, of = 0, uf = 0, count = 32
                 660, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 1, valid = 1, of = 0, uf = 0, count = 31
                 680, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000000, empty = 0, a_e = 0, full = 0, a_f = 1, valid = 1, of = 0, uf = 0, count = 30
                 700, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000001, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 29
                 720, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000010, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 28
                 740, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000011, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 27
                 760, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000100, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 26
                 780, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000101, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 25
                 800, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000110, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 24
                 820, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00000111, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 23
                 840, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 22
                 860, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001001, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 21
                 880, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001010, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 20
                 900, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001011, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 19
                 920, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001100, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 18
                 940, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001101, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 17
                 960, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001110, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 16
                 980, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00001111, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 15
                1000, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 14
                1020, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010001, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 13
                1040, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010010, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 12
                1060, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010011, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 11
```

2. dcd142.ecs.csun.edu (jlm777)

```
                1080, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010100, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count = 10
                1100, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010101, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  9
                1120, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010110, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  8
                1140, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00010111, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  7
                1160, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011000, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  6
                1180, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011001, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  5
                1200, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011010, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  4
                1220, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011011, empty = 0, a_e = 0, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  3
                1240, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011100, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  2
                1260, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011101, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                1280, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  0
                1300, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1320, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1340, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1360, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1380, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1400, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1420, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1440, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1460, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1480, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1500, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1520, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1540, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1560, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1580, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1600, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1620, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1640, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1660, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1680, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1700, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1720, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1740, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1760, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1780, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1800, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1820, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1840, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1860, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1880, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1900, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1920, data_in = 00011111, clk = 0, wr_en = 0, rd_en = 1, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1940, data_in = 00011111, clk = 0, wr_en = 1, rd_en = 0, rst = 1 | data_out = 00011110, empty = 1, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 1, count =  0
                1960, data_in = 00011111, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00011110, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 0, of = 0, uf = 0, count =  1
                1980, data_in = 00000000, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00011111, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2000, data_in = 00000001, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00011111, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2020, data_in = 00000010, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000000, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2040, data_in = 00000011, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000001, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2060, data_in = 00000100, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000010, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2080, data_in = 00000101, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000011, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2100, data_in = 00000110, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000100, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2120, data_in = 00000111, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000101, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2140, data_in = 00001000, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000110, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2160, data_in = 00001001, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00000111, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2180, data_in = 00001010, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00001000, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2200, data_in = 00001011, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00001001, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2220, data_in = 00001100, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00001010, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
                2240, data_in = 00001101, clk = 0, wr_en = 1, rd_en = 1, rst = 1 | data_out = 00001011, empty = 0, a_e = 1, full = 0, a_f = 0, valid = 1, of = 0, uf = 0, count =  1
```

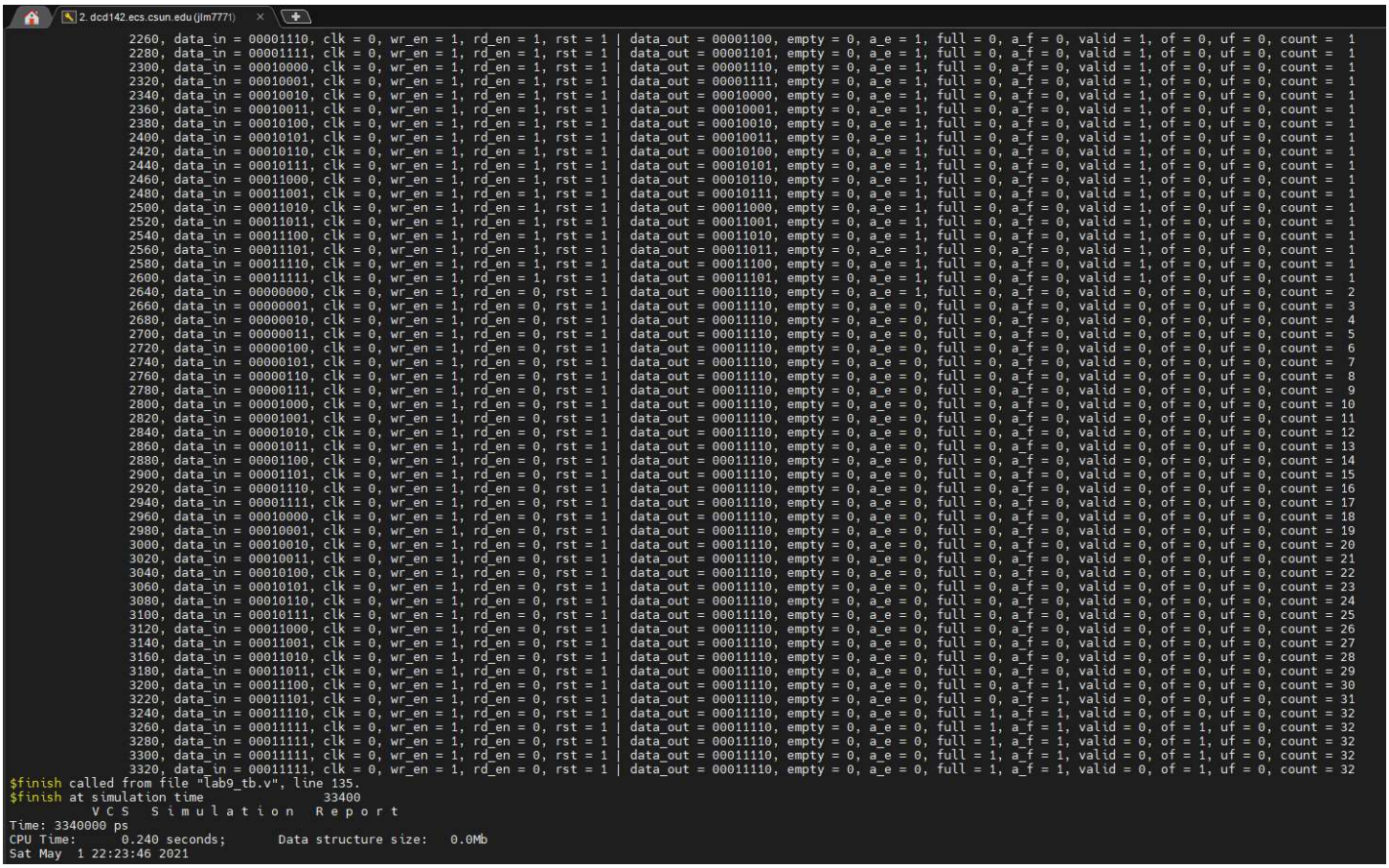

Fig. 4 simv output

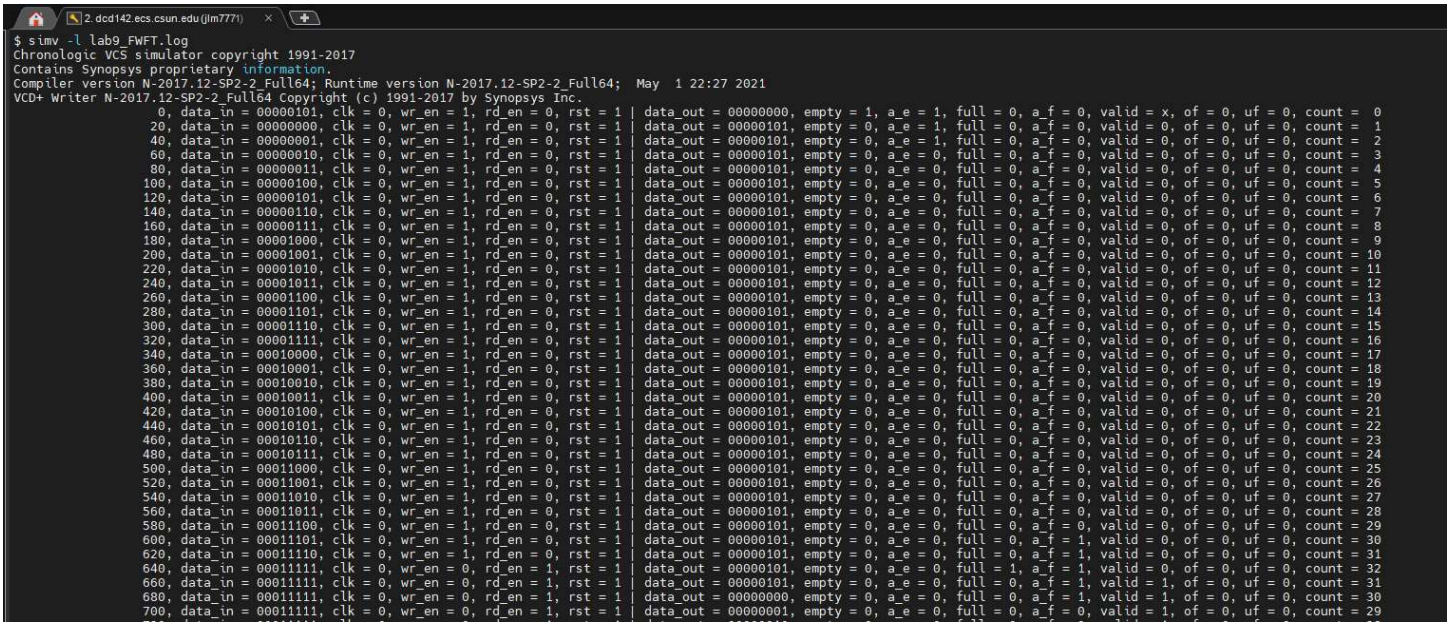

Fig. 5 simv FWFT output

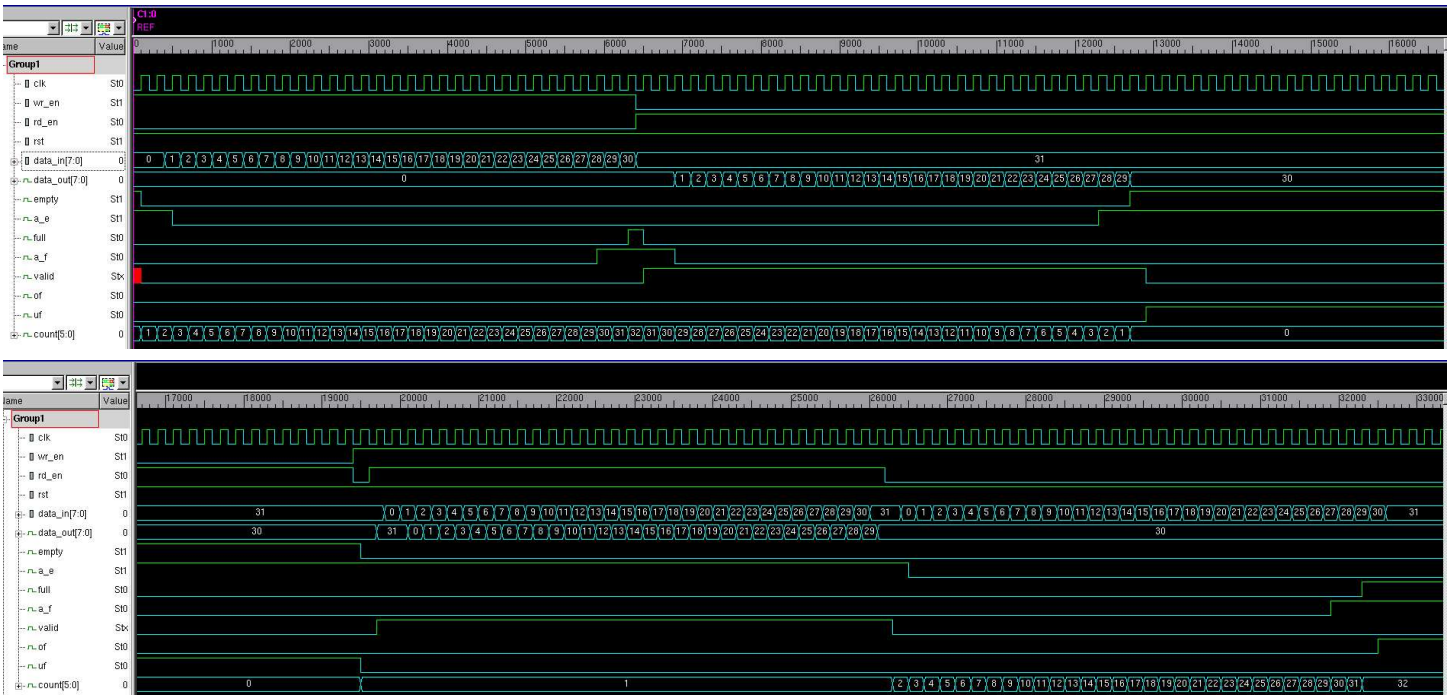

Fig. 6 waveforms

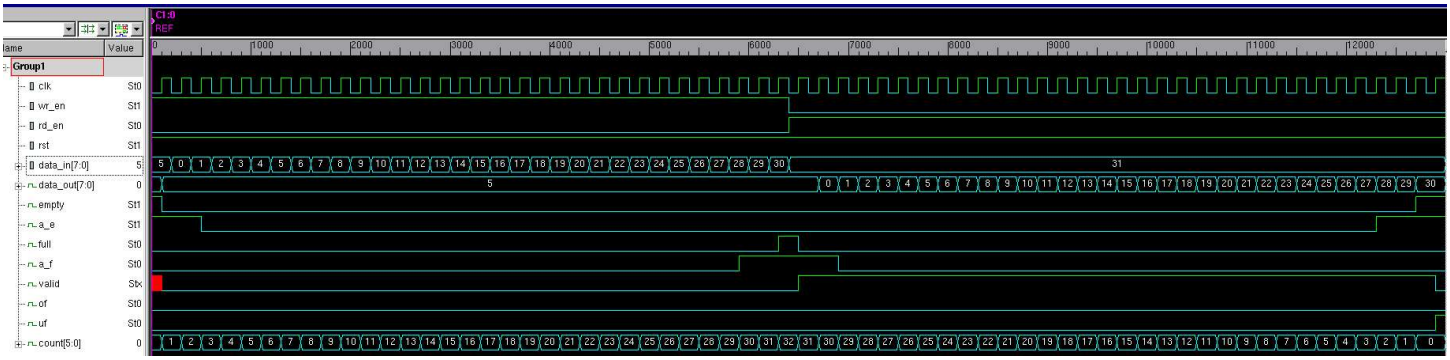

Fig. 7 FWFT waveforms

Analysis

As we can see from **Fig 4** & **6** our FIFO module is writing to the FIFO whenever the wr_en is 1'b1 and the FIFO is not full. We can see at time 3260ns the wr_en signal is 1'b1 and the FIFO was full in the previous period so when we tried to write to it the overflow flag was enabled. From 0ns to 620ns our wr_en is 1'b1 so every following clock cycle we see that our count is going up meaning that we are successfully writing to the FIFO. From 640ns to 1280ns our rd_en is 1'b1 and wr_en is 1'b0 and we can see that our count is going down meaning that our FIFO is counting down once per item read. Also the entire time it is reading the valid signal is 1'b1 meaning that the data on data_out is valid. However from 1300ns to 1940ns the rd_en signal is kept at 1'b1 and since the FIFO is empty at this point the underflow flag is set to 1'b1 and since the data on data_out is no longer valid, the valid signal is 1'b0. 1960ns to 1600ns both the wr_en and rd_en are set to 1'b1 so the FIFO should write and read at the same time. As we can see from the simv logs and waveforms the count is staying the same and data that is presented on data_out is changing meaning that the design is functioning correctly. For the period between 600ns and 680ns the almost full flag is set to 1'b1 as the count values are 30 or higher. The full flag will be set to 1'b1 whenever the count is equal to 32 and that can be verified by just looking at simv output whenever count = 32. For the periods between 0ns and 40ns the almost empty flag is set to 1'b1 because the value of count is 2 or lower. The empty flag will be set to 1'b1 whenever the count is equal to 0 and can be verified the same way as full.

For the First Word Fall Through we can look at **Fig. 5** & **7** to verify its functionality. As we can see at time 0ns the value of data_in is 5, wr_en is 1'b1. At 20ns even though the rd_en is 1'b0, data_out will still show the first word being 5. After that the FIFO will wait for the rd_en to be 1'b1 in order to read more of the FIFO contents.

Conclusion

In conclusion I learned about First in First out buffers and how to implement them. I learned how to use pointers and counters to keep track of where I am in the FIFO and how many items there currently. This lab also helped me practice behavioral modeling. I also had a lot of bugs in this lab which I spent a lot of time trying to fix. So I learned how to better tackle bugs and find the root issue. One thing I am proud of this lab is the use of ifdef. With this compiler directive i was able to save some time by just having one testbench and fifo module.

Academic Dishonesty

Submitting any report that is not entirely your own work is a form of academic dishonesty and will not be tolerated. Each and every lab report must include the following statement, signed and dated by the student. Lab reports without the statement will be summarily rejected.

I hereby attest that this lab report is entirely my own work. I have not copied either code or text from anyone, nor have I allowed or will I allow anyone to copy my work.

Name (printed) _____

Name (signed) _____ Date _____