



SACRAMENTO STATE

Course-EEE 273
Instructor: Dr.Behnam Arad
Parametrized FIFO
team10
Spring-2019

Student's Name:
Karthik M Shekarappa
Mittai Sreelekha

TABLE OF CONTENTS

S-No	Topics	Page-No
1	Status Report	
2	Block Diagram	
	• Block diagram of DUT	
	• Block diagram of Fixture	
3	Write FSM	
4	Read FSM	
5	Source Code	
	• Memory	
	• Write Logic	
	• Read Logic	
	• Write to Read Synchronizer	
	• Read to Write Synchronizer	
	• Top Module	
6	Test Fixture	
7	FIFO Verification Model	
8	Simulation Results	
9	Synthesis	
	• Synthesis Script	

	<ul style="list-style-type: none"> • FIFO Schematic 	
	<ul style="list-style-type: none"> • Simulation Waveforms 	
	<ul style="list-style-type: none"> • Synthesis Reports 	
10	Code Coverage	
11	Experimented DUT Using ONE HOT Code	
	<ul style="list-style-type: none"> • Improvised Synthesis reports 	

CSc/EEE 273
Term Project Status Report
Team 10.

<i>Names</i>	<i>Signatures</i>	<i>Overall Grade</i>
<i>1 Karthik Mallappahalli Shekarappa</i>		
<i>2 Sreelekha Mittai</i>		
<i>3</i>		

Provide the required information as accurately as possible based on the status of your FIFO submitted on the due date. The completed form must appear after Table of Contents (TOC) in your report.

I. Project Report / Demo **/ 200 points**

Table 1: (To be filled by the instructor)

Item in the report	Issues	points
Cover sheet		
TOC		
Project Status Report		
Block diagrams for the DUT and Testbench		
FSM diagrams		
Source code		
Test bench <ul style="list-style-type: none"> • DUT instant • Model • Automated validation • Simulation results (submitted as part of the softcopy only) 		
Synthesis <ol style="list-style-type: none"> 1. Script 2. Synthesis report 3. check_design report 		
Tabulated area and timing results		
Synthesis reports		
Code Coverage results		

VALIDITY OF RESULTS IN THE REPORT		
-----------------------------------	--	--

II.Design and Modeling Phase:

/550 points

Table 2. Source code: Comment on the functionality of the source code you developed for each component of the FIFO as accurately as possible. The comments you provide here must be based on your simulation results. Add more rows as needed. (300 points)

Component	Name of the person who modeled and validated the component	Is this component fully functional?	State any functional issue
memory.v	Sreelekha Mittai	Yes	
Read_logic.v	Sreelekha Mittai	Yes	Does not support insert after full
Write_logic.v	Karthik MS	yes	Does not support remove after flush or reset
Syncrd2wr.v	Karthik MS	Yes	
Syncwr2rd.v	Karthik MS	Yes	
Fifo.v	Sreelekha Mittai	Yes	

Table 3 Top-level Design. Comment on the functionality of the FIFO you developed (The functionality of the top-level design). 250 points

State functional issues in DUT	Does not support insert after full and remove after flush or reset
State functional issues in the testbench	NONE

Table 4. Code Coverage: Fill the following table based on the coverage reports vcs generated for your FIFO design for applicable options.

Coverage Type	Percentage	Comments if any
Line	96.60	
Toggle	97.68	

Conditional	100	
FSM	100	

III. Synthesis: 25 %

250 points

Table 5. Fill out the following table based on your best synthesis trials: timing & area. In each case, state the sign of the timing and area parameters as provided by the Design Compiler tool. The first row is a summary your results based on given clock rates. You should use the following constrains in all synthesis attempts using high synthesis effort:

Max input/output delay based on clk_in 0.4ns
Min input/output delay based on clk_in 0.2 ns
Max input/output delay based on clk_out 0.4ns
Min input/output delay based on clk_out 0.2ns

You must provide synthesis results based on the clock frequencies listed for Trails 1and 2 listed below. You can list successful synthesis attempts at other frequencies under Trials 3 and 4. These attempts should be based on enhancements to RTL code to improve synthesis results.

Trial	Clk_in Period in ns	Clk_out Period in ns	Area slack from area report	Timing slack from timing report	data required time for max path from timing report	data arrival time for the max path from timing report path
1	2 ns	1 ns	193710.326746	Clk_in = 0.02 Clk_out = -0.53	Clk_in = 1.92 Clk_out = 0.91	Clk_in = 1.90 Clk_out = 1.44
2	1ns	2ns	247763.181569	Clk_in = -0.63 Clk_out = 0.01	Clk_in = 0.91 Clk_out = 1.92	Clk_in = 1.55 Clk_out = 1.9
3	1.65ns	3.3ns	247375.501146	Clk_in = 0.00 Clk_out = 0.31	Clk_in = 1.56 Clk_out = 3.24	Clk_in = 1.56 Clk_out = 2.94
4	1.6ns	3ns	247587.944318	Clk_in = 0.00 Clk_out = 0.41	Clk_in = 1.51 Clk_out = 3.11	Clk_in = 1.51 Clk_out = 2.70

Briefly describe any RTL changes you made to improve synthesis results.

Trial 3: Changed frequency in synthesis script for different iterations to meet slack

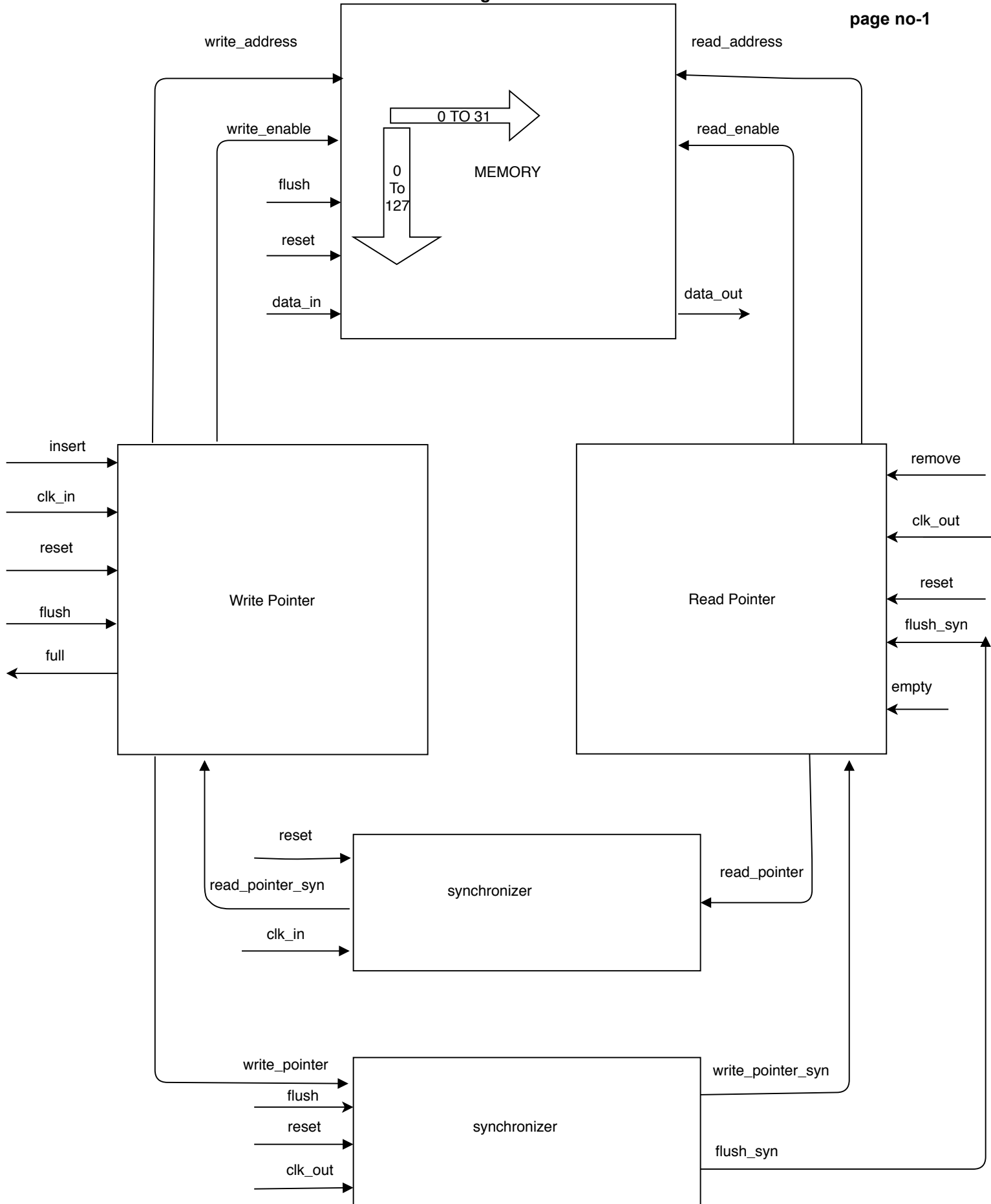
Trial 4: We used one hot for state representation and then used different iteration to meet Timing Slack at clk_in 1.6ns and clk_out 3.2 ns.

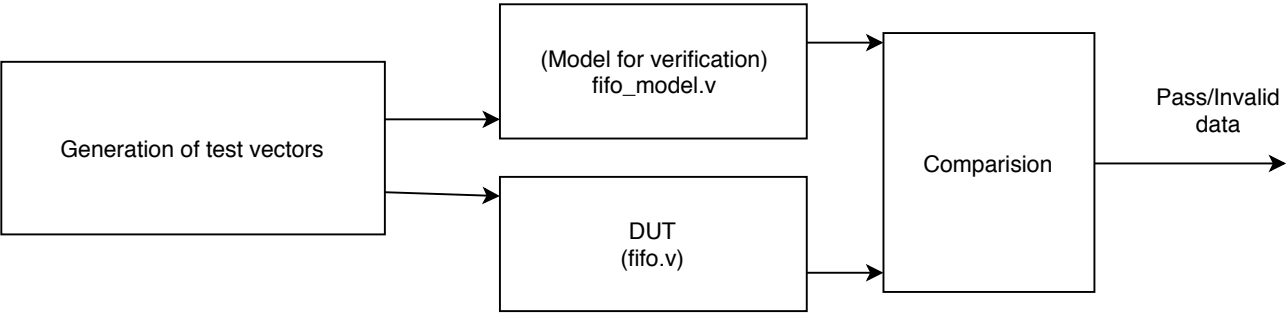
Tables 6: For each partner, state the contribution percentage for each task listed below: Please only provide a percentage.

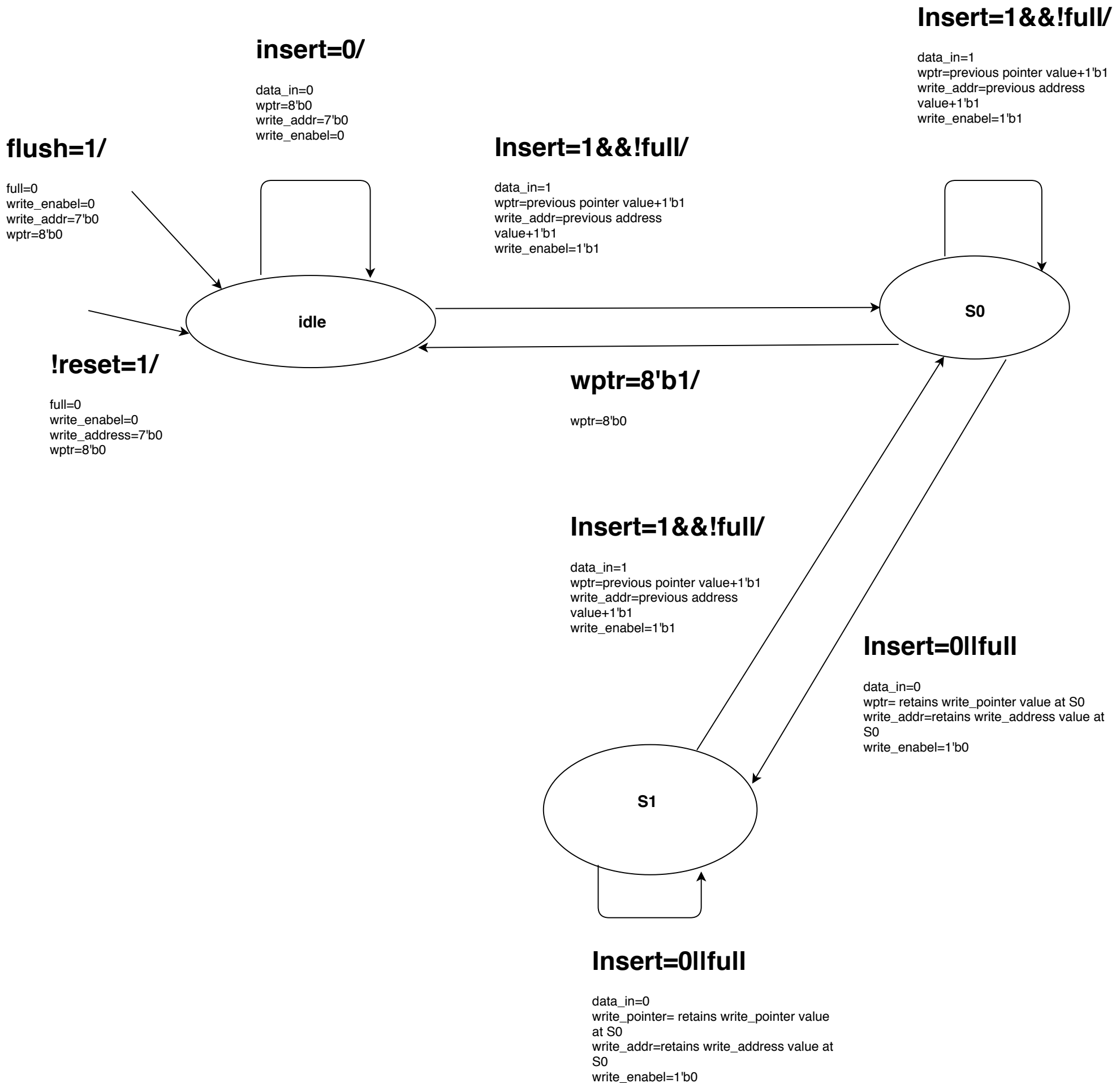
Name	Design	Simulation	Synthesis	Project report
Sreelekha Mittai	49%	49%	51%	54%
Karthik M Shekarappa	51%	51%	49%	46%

Block Diagram for DUT

page no-1







Syn_flush/

remove=0/

FSM For Read Logic

remove=1&&empty=0

empty=1
read_enabel=0
read_addr=7'b0
rptr=8'b0
data_out=0

empty=1
read_enabel=0
read_addr=7'b0
rptr=8'b0
data_out=0

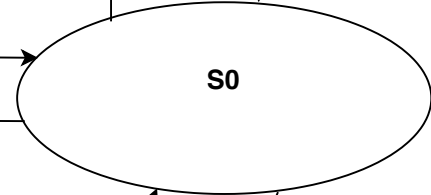
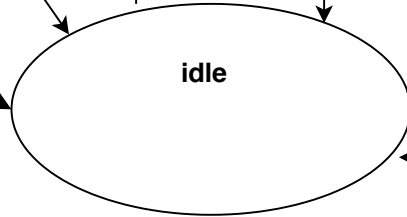
remove=1&&empty=0

data_out=data_in
read_enabel=1
rptr=previous pointer value+1'b1
read_addr=previous address value+1'b1

data_out=data_in
read_enabel=1
rptr=previous pointer value+1'b1
read_addr=previous address value+1'b1

!reset/

empty=1
read_enabel=0
read_addr=7'b0
rptr=8'b0
data_out=0



rprr=8'b1/

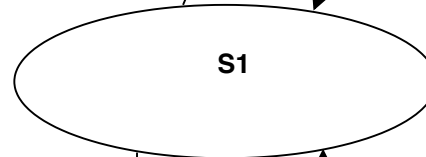
rprr=8'b0

remove=1&&empty=0

data_out=data_in
read_enabel=1
rptr=previous pointer value+1'b1
read_addr=previous address value+1'b1

remove=0||empty=1

read_enabel=0
data_out=0
read_addr=7'b0
rptr=8'b0



remove=0||empty=1

read_enabel=0
data_out=0
read_addr=7'b0
rptr=8'b0

```
module fifo_memory #(parameter mem_width =32, mem_depth = 128, depth= 7 )
(input[mem_width-1:0]temp_data_in,input[depth-1:0] read_addr, write_addr,
input write_enable,read_enable, flush, reset,output reg [mem_width-1:0]
data_out) ;
reg [mem_width-1:0] mem[0:mem_depth-1];
integer i;
always @(*)
begin
    if(!reset)
    begin
        data_out = 32'h0;
        for(i=0;i<128;i=i+1)
            mem[i] = 32'h0;
    end

    else if(flush)
    begin
        data_out = 32'h0;
        for(i=0;i<128;i=i+1)
            mem[i] = 32'h0;
    end

    else if(write_enable && read_enable)
    begin
        mem[write_addr] = temp_data_in;
        data_out = mem[read_addr];
    end

    else if(write_enable )
    begin
        mem[write_addr] = temp_data_in;
    end

    else if(read_enable )
        data_out = mem[read_addr];

end
endmodule
```

```
module write_logic#(parameter depth = 7, width = 8)(input
clk_in,insert,reset,flush,input [depth : 0]r2wsync_ff2,input[31:0]
data_in, output reg [depth-1 : 0]write_addr, output reg [depth : 0]wptr,
output reg write_enable,full,output reg[31:0] temp_data_in);
parameter[1:0] idle=2'b00, s0=2'b01,s1=2'b11;
reg[1:0] current_state, next_state;
reg [31:0] temp;
always@(posedge clk_in, negedge reset)
begin
    if(!reset)
        current_state<=idle;
    else if(flush)
        current_state<=idle;
    else
        current_state<=next_state;
end
always@(*)
begin

case(current_state)
    idle:
        if(insert)
            next_state=s0;
        else
            next_state=idle;
    s0:
        if(insert)
            next_state=s0;
        else
            next_state=s1;
    s1:
        if(insert)
            next_state=s0;
        else
            next_state=s1;
        default:    next_state=idle;
endcase
end
always@(posedge clk_in, negedge reset)
begin
    if(!reset)
    begin
        full<=0;
        write_enable <= 0;
        write_addr <= 0;
        wptr <=0;
    end
    else if(flush)
    begin
        full<=0;
        write_enable <= 0;
        write_addr <= 0;
        wptr <=0;
    end
end
```

```

else
case(current_state)
  idle:
  begin
    full<=0;
    write_enable <= 0;
    write_addr <= 0;
    wptr <=0;

  end
  s0:
  begin
    if(wptr[depth-1:0] == r2wsync_ff2[depth-1:0] &&
wptr[depth] != r2wsync_ff2[depth])
    begin
      full<=1'b1;
      write_enable <= 1'b0;
      write_addr <= write_addr;
      wptr <= wptr;

    end
    else
    begin
      full<=1'b0;
      write_enable <= 1'b1;
      write_addr <= wptr[width-2:0];
      if(wptr == 8'b11111111)
        wptr <= 8'h0;
      else begin
        wptr <= wptr+1;
      end

    end

  end
  s1:
  begin
    if(!(wptr[depth-1:0] == r2wsync_ff2[depth-1:0] &&
wptr[depth] != r2wsync_ff2[depth]))
    begin
      full<=0;
      write_enable <= 0;
      write_addr <= write_addr;
      wptr <= wptr;

    end
    else
    begin
      full<=full;
      write_enable <= 0;
      write_addr <= write_addr;
      wptr <= wptr;

    end

  end
endcase
end
always @ (posedge clk_in or negedge reset)

```

```
begin
    if(!reset)
        temp_data_in<=32'h0;
    else
        begin
            temp4<=data_in;
            temp_data_in<=temp4;
        end
    end
end
endmodule
```

```
module read_logic #(parameter depth = 7, width = 8) (input
clk_out,remove,reset,syn_flush,input [depth: 0]w2rsync_ff2, output reg
[depth-1 : 0]read_addr, output reg [depth : 0]rptr, output reg
read_enable, empty );
parameter[1:0] idle=2'b00, s0=2'b01,s1=2'b11;
reg[1:0] current_state, next_state;
always@(posedge clk_out, negedge reset)
begin
    if(!reset)
        current_state<=idle;
    else if(syn_flush)
        current_state<=idle;
    else
        current_state<=next_state;
end
always@(*)
begin

    case(current_state)
        idle:
            if(remove)
                next_state=s0;
            else
                next_state=idle;
        s0:
            if(remove)
                next_state=s0;
            else
                next_state=s1;
        s1:
            if(remove)
                next_state=s0;
            else
                next_state=s1;

        default:
            next_state=idle;
    endcase
end

always@(posedge clk_out, negedge reset)
begin
    if(!reset)
    begin
        empty<=1'b1;
        read_enable <= 0;
        read_addr <= 0;
        rptr <=0;
    end
    else if(syn_flush)
    begin
        empty<=1'b1;
    end
end
```



```

        read_enable <= 0;
        read_addr <= 0;
        rptr <=0;
    end
    else
    case(current_state)
        idle:
        begin
            if(w2rsync_ff2[depth:0]>0)
                empty<=1'b0;
            else
            begin
                empty<=1'b1;
                read_enable <= 0;
                read_addr <= 0;
                rptr <= 0;
            end
        end
        s0:
        begin
            if(rp[depth:0] == w2rsync_ff2[depth:0])
            begin
                empty<=1'b1;
                read_enable <= 1'b0;
                read_addr <= read_addr;
                rptr <= rptr;
            end

            else
            begin
                read_addr<=rp[width-2:0];
                empty<=1'b0;
                read_enable<=1'b1;
                if(rp==8'b11111111)
                    rptr<=8'h0;
                else
                    rptr<=rptr+1;
                end
            end

        end
        s1:
        begin
            if(rp[depth:0] !=w2rsync_ff2[depth:0])
                empty<=1'b0;
            else
            begin
                empty<=empty;
                rptr<=rptr;
                read_enable<=1'b0;
                read_addr<=read_addr;
            end
        end
    endcase
endcase

```

```
end  
endmodule
```

```
module sync_wr2rd #(parameter width =8, depth = 7)
  (input[depth:0] wptr,input clk_out,reset,flush, output reg[depth:0]
  w2rsync_ff2, output reg syn_flush);
  reg [depth:0] w2rsync_ff1;
  reg sync_flush1;
  always @(posedge clk_out, negedge reset)
  begin
    if(!reset)
    begin
      w2rsync_ff2 <= 0 ;
      w2rsync_ff1 <= 0 ;
    end
    else
    begin
      w2rsync_ff1 <= wptr ;
      w2rsync_ff2 <= w2rsync_ff1;
    end
  end
  always @(posedge clk_out, negedge reset)
  begin
    if(!reset)
      syn_flush<=0;
    else
    begin
      sync_flush1 <= flush ;
      syn_flush<= sync_flush1;
    end
  end
endmodule
```

```
module sync_rd2wr #(parameter width =8, depth = 7)
(input[depth:0] rptr,input clk_in,reset,flush, output reg[depth:0]
r2wsync_ff2);
reg [depth:0] r2wsync_ff1;

always @(posedge clk_in, negedge reset)
begin
    if(!reset)
    begin
        r2wsync_ff2 <= 0 ;
        r2wsync_ff1 <= 0 ;
    end
    else
    begin // $display("b4syn r2w",rptr );
        r2wsync_ff1 <= rptr ;
        r2wsync_ff2 <= r2wsync_ff1; // $display("aftr syn
r2w",r2wsync_ff2 );
    end
end

endmodule
```

```
`include "read_logic.v"
`include "write_logic.v"
`include "fifo_memory.v"
`include "sync_rd2wr.v"
`include "sync_wr2rd.v"
module fifo #(parameter mem_width = 32, mem_depth = 128, depth = 7, width =
8)(input clk_in, clk_out, flush, reset, insert, remove, input [mem_width-
1:0] data_in, output [mem_width-1:0] data_out, output full, empty);

wire [depth:0] r2wsync_ff2, w2rsync_ff2, rptr, wptr;
wire write_enable, read_enable, syn_flush;
wire [depth-1:0] write_addr, read_addr;
wire [mem_width-1:0] temp_data_in;

//write_logic
write_logic #( 7, 8) w1(.clk_in(clk_in), .reset(reset), .flush(flush),
.insert(insert), .r2wsync_ff2(r2wsync_ff2), .full(full),
.write_enable(write_enable), .write_addr(write_addr),
.wptr(wptr), .temp_data_in(temp_data_in), .data_in(data_in));

//read_logic
read_logic #(7,8) r1(.clk_out(clk_out), .reset(reset), .remove(remove),
.syn_flush(syn_flush), .w2rsync_ff2(w2rsync_ff2),
.empty(empty), .read_enable(read_enable), .read_addr(read_addr),
.rptr(rptr));

//fifo_memory
fifo_memory #(32,128,7) m1(.reset(reset),
.flush(flush), .write_enable(write_enable), .read_enable(read_enable),
.write_addr(write_addr), .read_addr(read_addr),
.data_out(data_out), .temp_data_in(temp_data_in));

//sync_wr2rd
sync_wr2rd #(8,7) w2r1(.wptr(wptr), .clk_out(clk_out), .reset(reset),
.flush(flush), .w2rsync_ff2(w2rsync_ff2), .syn_flush(syn_flush));

//sync_rd2wr
sync_rd2wr #(8,7) r2w1(.rptr(rptr), .clk_in(clk_in), .reset(reset),
.r2wsync_ff2(r2wsync_ff2));

endmodule
```

```

`include "fifo_model.v"
`include "fifo.v"
module fifo_fixture;
parameter width_memory=32,depth_memory=128;
reg clk_in,clk_out,flush,reset,insert,remove; reg [width_memory-1:0]
data_in;
wire [width_memory-1:0] data_out;
wire full,empty,full1,empty1;
wire [width_memory-1:0] data_out_1;
wire full1,empty1; integer i,j;
reg[1:0] current_state, next_state;
//Design Model instance
fifo_model #(128,32,7)
u1(.clk_in(clk_in),.clk_out(clk_out),.flush(flush),.reset(reset),.insert(
insert),.remove(remove),.data_in(data_in),.data_out(data_out_1),.full(full
1),.empty(empty1));
//DUT instance
fifo #(32,128,7,8)
u2(.clk_in(clk_in),.clk_out(clk_out),.flush(flush),.reset(reset),.insert(
insert),.remove(remove),.data_in(data_in),.data_out(data_out),.full(full)
,.empty(empty));
initial

$vcddpluson;

initial
begin
clk_in=1'b1;
forever #1 clk_in=~clk_in;
end
initial
begin
clk_out=1'b0;
forever #2 clk_out=~clk_out;
end
//Initial conditiion on inputs to not see unknow at t=0
initial
begin
reset=1'b0;
flush =1'b0;
insert=1'b0;
remove=1'b0;
data_in=32'b0;
#8;
end
//task for reset
task reset_task();
begin
reset=1'b0;
#5 reset=1'b1;
end
endtask

//task for flush

```

```

task flush_task();
begin
flush=1'b1;
#10 flush=1'b0;
end
endtask

//task for insert
task insert_task(input ins);
begin
#1 insert=ins;
#1 ;
end
endtask

initial begin
reset_task();
for (i=0; i<150;i=i+1) //to check fullcondition
begin
@(negedge clk_in);
data_in=$random;
insert=1;
#2;
end
insert=0;//to check emptycondition
#10;
remove=1;
#500;
//insert=0; //to checkflush
remove=0;
for (i=0; i<60;i=i+1) //to check fullcondition
begin
if (full== 1'b1)
insert = 1'b0;
else
begin
@(negedge clk_in);
data_in=$random;
insert=1;
#2;
end
end
insert=0;
flush_task();
#10;
for (i=0; i<100;i=i+1) //random generation oftestinputs
begin
@(negedge clk_in);
data_in=$random;
insert =$random;
#2;
insert =0;
#2;

```

```

@(negedge clk_out);
remove =$random;
#4;
remove=0;
end
reset_task();
$finish;
end

initial //checker block
begin
forever @(posedge clk_out )
if(data_out == data_out_1)
$display($time," clk_in=%b, clk_out=%b, reset=%b, flush=%b,insert=%b,
remove=%b, datain=%h,dataout=%h,empty=%b, full=%b
PASS",clk_in,clk_out,reset,flush,insert,remove,data_in,data_out,empty,full);
else
$display($time," clk_in=%b, clk_out=%b, reset=%b, flush=%b,insert=%b,
remove=%b, datain=%h, dataout=%h,data_out_1=%h, empty=%b, full=%b not
valid Data",clk_in,
clk_out,reset,flush,insert,remove,data_in,data_out_1,data_out,empty,full)
;
end
endmodule

```



```

module fiffo_model #(parameter f_depth=128,f_width=32,ptr_wdth=7) (input
clk_in,clk_out,flush,reset,insert,remove,input [f_width-1:0]
data_in,output reg [f_width-1:0] data_out,output reg full,empty);
reg [ptr_wdth:0]rptr,wptr;
reg flush_sync;
reg read_enable, write_enable;
reg [f_width-1:0] temp_data_in;
reg [f_width-1:0]f_mem[127:0];
integer i;

always @(posedge clk_in, negedge reset)
begin
    temp_data_in<= data_in;
    if(!reset)
    begin
        empty <= 1;
        data_out<=0;
        full<=0;
        rptr <=0;
        wptr<=0;
        for(i=0; i<128;i=i+1)
            f_mem[i]<=32'b0;
    end

    else if(flush)
    begin
        empty <= 1;
        data_out<=0;
        full<=0;
        rptr <=0;
        wptr<=0;
        for(i=0; i<128;i=i+1)
            f_mem[i]<=32'b0;
    end

    else if(insert)
    begin
        if(wptr[ptr_wdth-1:0]-1==rp[ptr_wdth-1:0]-1 &&
wptr[ptr_wdth]!=rp[ptr_wdth])
        begin
            full<=1'b1;
            wptr<=wptr;

            end
        else
        begin
            #2 full<=1'b0;
            f_mem[wptr[ptr_wdth-1:0]]<= temp_data_in;
            if(wptr==8'b11111111)
                wptr<=8'h0;
            else
                wptr<=wptr+1;
            end
        end
    end
else

```

```

        begin
            if (! (wptr[ptr_width-1:0]-1==rptr[ptr_width-1:0]-1 &&
wptr[ptr_width]!=rptr[ptr_width]))
                full<=0;
            else
                begin
                    #2 full<=full;
                    wptr<=wptr;
                end
            end
        end
    end
always @(posedge clk_out , negedge reset)
begin
    if(!reset)
    begin
        empty<=1'b1;
        rptr<=1'b0;
        #4 data_out<=1'b0;
    end
    else if(flush_sync)
    begin
        empty<=1'b1;
        rptr<=1'b0;
        #4 data_out<=1'b0;
    end
    else if(remove)
    begin
        if(rp[ptr_width:0]-1==wptr[ptr_width:0]-1)
        begin
            empty<=1'b1;
            rptr<=rp;
            #4 data_out<=data_out;
        end
        else
        begin
            empty<=1'b0;
            #4 data_out<=f_mem[rptr[ptr_width-1:0]];
            if(rp==8'b11111111)
                rptr<=8'h0;
            else
                rptr<=rp+1;
        end
    end
    else
    begin
        empty<=empty;
        #4 data_out<=data_out;
        rptr<=rp;
    end
end
endmodule

```

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; Apr 29 19:55

2019

```
      2 clk_in=1, clk_out=1, reset=0, flush=0,insert=0,
remove=0, datain=00000000,dataout=00000000,empty=1, full=0 PASS
      6 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=12153524,dataout=00000000,empty=1, full=0 PASS
     10 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=8484d609,dataout=00000000,empty=1, full=0 PASS
     14 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=06b97b0d,dataout=00000000,empty=1, full=0 PASS
     18 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b2c28465,dataout=00000000,empty=1, full=0 PASS
     22 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=00f3e301,dataout=00000000,empty=0, full=0 PASS
     26 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3b23f176,dataout=00000000,empty=0, full=0 PASS
     30 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=76d457ed,dataout=00000000,empty=0, full=0 PASS
     34 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7cfde9f9,dataout=00000000,empty=0, full=0 PASS
     38 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=e2f784c5,dataout=00000000,empty=0, full=0 PASS
     42 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=72aff7e5,dataout=00000000,empty=0, full=0 PASS
     46 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=8932d612,dataout=00000000,empty=0, full=0 PASS
     50 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=793069f2,dataout=00000000,empty=0, full=0 PASS
     54 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=f4007ae8,dataout=00000000,empty=0, full=0 PASS
     58 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=2e58495c,dataout=00000000,empty=0, full=0 PASS
     62 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=96ab582d,dataout=00000000,empty=0, full=0 PASS
     66 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b1ef6263,dataout=00000000,empty=0, full=0 PASS
     70 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=c03b2280,dataout=00000000,empty=0, full=0 PASS
     74 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=557845aa,dataout=00000000,empty=0, full=0 PASS
     78 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=cb203e96,dataout=00000000,empty=0, full=0 PASS
     82 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=86bc380d,dataout=00000000,empty=0, full=0 PASS
     86 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=359fdd6b,dataout=00000000,empty=0, full=0 PASS
     90 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=81174a02,dataout=00000000,empty=0, full=0 PASS
     94 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=0effe91d,dataout=00000000,empty=0, full=0 PASS
     98 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=11844923,dataout=00000000,empty=0, full=0 PASS
```

```

102 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=e5730acā,dataout=00000000,empty=0, full=0 PASS
106 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7968bdf2,dataout=00000000,empty=0, full=0 PASS
110 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=20c4b341,dataout=00000000,empty=0, full=0 PASS
114 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3c20f378,dataout=00000000,empty=0, full=0 PASS
118 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=75c50deb,dataout=00000000,empty=0, full=0 PASS
122 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=634bf9c6,dataout=00000000,empty=0, full=0 PASS
126 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=de7502bc,dataout=00000000,empty=0, full=0 PASS
130 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=85d79a0b,dataout=00000000,empty=0, full=0 PASS
134 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=42f24185,dataout=00000000,empty=0, full=0 PASS
138 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9dcc603b,dataout=00000000,empty=0, full=0 PASS
142 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=bf23327e,dataout=00000000,empty=0, full=0 PASS
146 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=78d99bf1,dataout=00000000,empty=0, full=0 PASS
150 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=31230762,dataout=00000000,empty=0, full=0 PASS
154 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=4fa1559f,dataout=00000000,empty=0, full=0 PASS
158 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7c6da9f8,dataout=00000000,empty=0, full=0 PASS
162 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=cfc4569f,dataout=00000000,empty=0, full=0 PASS
166 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=adcbc05b,dataout=00000000,empty=0, full=0 PASS
170 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=a4ae3249,dataout=00000000,empty=0, full=0 PASS
174 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ebfec0d7,dataout=00000000,empty=0, full=0 PASS
178 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=4b212f96,dataout=00000000,empty=0, full=0 PASS
182 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=e12ccec2,dataout=00000000,empty=0, full=0 PASS
186 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=bb825a77,dataout=00000000,empty=0, full=0 PASS
190 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=090cdb12,dataout=00000000,empty=0, full=0 PASS
194 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=36e5816d,dataout=00000000,empty=0, full=0 PASS
198 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=0fd28f1f,dataout=00000000,empty=0, full=0 PASS
202 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=42d92f85,dataout=00000000,empty=0, full=0 PASS
206 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=2dda595b,dataout=00000000,empty=0, full=0 PASS

```

```

210 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9ff2ae3f,dataout=00000000,empty=0, full=0 PASS
214 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=2c156358,dataout=00000000,empty=0, full=0 PASS
218 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=c71a0c8e,dataout=00000000,empty=0, full=0 PASS
222 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7d3599fa,dataout=00000000,empty=0, full=0 PASS
226 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=39961773,dataout=00000000,empty=0, full=0 PASS
230 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9799a82f,dataout=00000000,empty=0, full=0 PASS
234 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=afd8565f,dataout=00000000,empty=0, full=0 PASS
238 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7bf8fdf7,dataout=00000000,empty=0, full=0 PASS
242 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=f3091ae6,dataout=00000000,empty=0, full=0 PASS
246 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=14cfc129,dataout=00000000,empty=0, full=0 PASS
250 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ed536cda,dataout=00000000,empty=0, full=0 PASS
254 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=da8ae2b5,dataout=00000000,empty=0, full=0 PASS
258 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3cf11979,dataout=00000000,empty=0, full=0 PASS
262 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=e8740cd0,dataout=00000000,empty=0, full=0 PASS
266 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=55f6adab,dataout=00000000,empty=0, full=1 PASS
270 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=6e5daddc,dataout=00000000,empty=0, full=1 PASS
274 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=fedf72fd,dataout=00000000,empty=0, full=1 PASS
278 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=2b0eed56,dataout=00000000,empty=0, full=1 PASS
282 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b3d97667,dataout=00000000,empty=0, full=1 PASS
286 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=5b6fb9b6,dataout=00000000,empty=0, full=1 PASS
290 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3cd18779,dataout=00000000,empty=0, full=1 PASS
294 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=4a74bf94,dataout=00000000,empty=0, full=1 PASS
298 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=823f2c04,dataout=00000000,empty=0, full=1 PASS
302 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=6dcb69db,dataout=00000000,empty=0, full=1 PASS
306 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a6fcde4d,dataout=00000000,empty=0, full=1 PASS
310 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a6fcde4d,dataout=00000000,empty=0, full=1 PASS
314 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a6fcde4d,dataout=00000000,empty=0, full=1 PASS

```

```

318 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=00000000,empty=0, full=1 PASS
322 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=00000000,empty=0, full=1 PASS
326 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d, dataout=e8740cd0,data_out_1=12153524, empty=0,
full=1 not valid Data
330 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=c0895e81,empty=0, full=0 PASS
334 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=8484d609,empty=0, full=0 PASS
338 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=b1f05663,empty=0, full=0 PASS
342 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=06b97b0d,empty=0, full=0 PASS
346 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=46df998d,empty=0, full=0 PASS
350 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=b2c28465,empty=0, full=0 PASS
354 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=89375212,empty=0, full=0 PASS
358 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=00f3e301,empty=0, full=0 PASS
362 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=06d7cd0d,empty=0, full=0 PASS
366 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=3b23f176,empty=0, full=0 PASS
370 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=1e8dcd3d,empty=0, full=0 PASS
374 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=76d457ed,empty=0, full=0 PASS
378 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=462df78c,empty=0, full=0 PASS
382 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=7cfde9f9,empty=0, full=0 PASS
386 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e33724c6,empty=0, full=0 PASS
390 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e2f784c5,empty=0, full=0 PASS
394 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=d513d2aa,empty=0, full=0 PASS
398 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=72aff7e5,empty=0, full=0 PASS
402 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=bdb27277,empty=0, full=0 PASS
406 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=8932d612,empty=0, full=0 PASS
410 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=47ecdb8f,empty=0, full=0 PASS
414 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=793069f2,empty=0, full=0 PASS
418 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e77696ce,empty=0, full=0 PASS

```

```

422 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=f4007ae8,empty=0, full=0 PASS
426 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e2ca4ec5,empty=0, full=0 PASS
430 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=2e58495c,empty=0, full=0 PASS
434 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=de8e28bd,empty=0, full=0 PASS
438 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=96ab582d,empty=0, full=0 PASS
442 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=b2a72665,empty=0, full=0 PASS
446 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=b1ef6263,empty=0, full=0 PASS
450 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=0573870a,empty=0, full=0 PASS
454 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=c03b2280,empty=0, full=0 PASS
458 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=10642120,empty=0, full=0 PASS
462 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=557845aa,empty=0, full=0 PASS
466 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=cecccc9d,empty=0, full=0 PASS
470 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=cb203e96,empty=0, full=0 PASS
474 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=8983b813,empty=0, full=0 PASS
478 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=86bc380d,empty=0, full=0 PASS
482 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=a9a7d653,empty=0, full=0 PASS
486 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=359fdd6b,empty=0, full=0 PASS
490 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=eaa62ad5,empty=0, full=0 PASS
494 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=81174a02,empty=0, full=0 PASS
498 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=d7563eae,empty=0, full=0 PASS
502 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=0effe91d,empty=0, full=0 PASS
506 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e7c572cf,empty=0, full=0 PASS
510 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=11844923,empty=0, full=0 PASS
514 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=0509650a,empty=0, full=0 PASS
518 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e5730aca,empty=0, full=0 PASS
522 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=9e314c3c,empty=0, full=0 PASS
526 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=7968bdf2,empty=0, full=0 PASS

```

```

530 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=452e618a,empty=0, full=0 PASS
534 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=20c4b341,empty=0, full=0 PASS
538 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=ec4b34d8,empty=0, full=0 PASS
542 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=3c20f378,empty=0, full=0 PASS
546 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=c48a1289,empty=0, full=0 PASS
550 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=75c50deb,empty=0, full=0 PASS
554 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=5b0265b6,empty=0, full=0 PASS
558 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=634bf9c6,empty=0, full=0 PASS
562 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=571513ae,empty=0, full=0 PASS
566 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=de7502bc,empty=0, full=0 PASS
570 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=150fdd2a,empty=0, full=0 PASS
574 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=85d79a0b,empty=0, full=0 PASS
578 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=b897be71,empty=0, full=0 PASS
582 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=42f24185,empty=0, full=0 PASS
586 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=27f2554f,empty=0, full=0 PASS
590 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=9dcc603b,empty=0, full=0 PASS
594 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=1d06333a,empty=0, full=0 PASS
598 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=bf23327e,empty=0, full=0 PASS
602 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=0aaa4b15,empty=0, full=0 PASS
606 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=78d99bf1,empty=0, full=0 PASS
610 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=6c9c4bd9,empty=0, full=0 PASS
614 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=31230762,empty=0, full=0 PASS
618 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=2635fb4c,empty=0, full=0 PASS
622 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=4fa1559f,empty=0, full=0 PASS
626 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=47b9a18f,empty=0, full=0 PASS
630 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=7c6da9f8,empty=0, full=0 PASS
634 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=dbcd60b7,empty=0, full=0 PASS

```



```

        638 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=cfc4569f,empty=0, full=0 PASS
        642 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=ae7d945c,empty=0, full=0 PASS
        646 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=adcbc05b,empty=0, full=0 PASS
        650 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=44de3789,empty=0, full=0 PASS
        654 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=a4ae3249,empty=0, full=0 PASS
        658 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e8233ed0,empty=0, full=0 PASS
        662 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=ebfec0d7,empty=0, full=0 PASS
        666 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=a8c7fc51,empty=0, full=0 PASS
        670 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=4b212f96,empty=0, full=0 PASS
        674 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=061d7f0c,empty=0, full=0 PASS
        678 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e12ccec2,empty=0, full=0 PASS
        682 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=6457edc8,empty=0, full=0 PASS
        686 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=bb825a77,empty=0, full=0 PASS
        690 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=1ef2ed3d,empty=0, full=0 PASS
        694 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=090cdb12,empty=0, full=0 PASS
        698 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=bf05007e,empty=0, full=0 PASS
        702 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=36e5816d,empty=0, full=0 PASS
        706 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=1cd9e739,empty=0, full=0 PASS
        710 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=0fd28f1f,empty=0, full=0 PASS
        714 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e9ebf6d3,empty=0, full=0 PASS
        718 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=42d92f85,empty=0, full=0 PASS
        722 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=bc148878,empty=0, full=0 PASS
        726 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=2dda595b,empty=0, full=0 PASS
        730 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=248b4b49,empty=0, full=0 PASS
        734 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=9ff2ae3f,empty=0, full=0 PASS
        738 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=150caf2a,empty=0, full=0 PASS
        742 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=2c156358,empty=0, full=0 PASS

```

```

746 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=c33f3886,empty=0, full=0 PASS
750 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=c71a0c8e,empty=0, full=0 PASS
754 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=ce2ff29c,empty=0, full=0 PASS
758 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=7d3599fa,empty=0, full=0 PASS
762 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=937dbc26,empty=0, full=0 PASS
766 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=39961773,empty=0, full=0 PASS
770 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=d18bb4a3,empty=0, full=0 PASS
774 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=9799a82f,empty=0, full=0 PASS
778 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=d9d292b3,empty=0, full=0 PASS
782 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=afd8565f,empty=0, full=0 PASS
786 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=22290d44,empty=0, full=0 PASS
790 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=7bf8fdf7,empty=0, full=0 PASS
794 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=e59b36cb,empty=0, full=0 PASS
798 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=f3091ae6,empty=0, full=0 PASS
802 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=2d28db5a,empty=0, full=0 PASS
806 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=14cfc129,empty=0, full=0 PASS
810 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=f682e2ed,empty=0, full=0 PASS
814 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a6fcde4d,dataout=ed536cda,empty=0, full=0 PASS
818 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b6a4266d,dataout=b29fb665,empty=0, full=0 PASS
822 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=653b49ca,dataout=da8ae2b5,empty=0, full=0 PASS
826 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=4a937195,dataout=da8ae2b5,empty=0, full=0 PASS
830 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=02749b04,dataout=da8ae2b5,empty=0, full=0 PASS
834 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=34980769,dataout=da8ae2b5,empty=0, full=0 PASS
838 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=44018d88,dataout=da8ae2b5,empty=0, full=0 PASS
842 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9690042d,dataout=da8ae2b5,empty=0, full=0 PASS
846 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=975c9c2e,dataout=da8ae2b5,empty=0, full=0 PASS
850 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=0e41451c,dataout=da8ae2b5,empty=0, full=0 PASS

```

```

854 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=149e0729,dataout=da8ae2b5,empty=0, full=0 PASS
858 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=43356786,dataout=da8ae2b5,empty=0, full=0 PASS
862 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9eb7c63d,dataout=da8ae2b5,empty=0, full=0 PASS
866 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b855c470,dataout=da8ae2b5,empty=0, full=0 PASS
870 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=5d7199ba,dataout=da8ae2b5,empty=0, full=0 PASS
874 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7d4779fa,dataout=da8ae2b5,empty=0, full=0 PASS
878 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=8d24f61a,dataout=da8ae2b5,empty=0, full=0 PASS
882 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=1b876137,dataout=da8ae2b5,empty=0, full=0 PASS
886 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=603921c0,dataout=da8ae2b5,empty=0, full=0 PASS
890 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=db461ab6,dataout=da8ae2b5,empty=0, full=0 PASS
894 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=6e5f0fdc,dataout=da8ae2b5,empty=0, full=0 PASS
898 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3c03ff78,dataout=da8ae2b5,empty=0, full=0 PASS
902 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ed8d80db,dataout=da8ae2b5,empty=0, full=0 PASS
906 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3ced2b79,dataout=da8ae2b5,empty=0, full=0 PASS
910 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b0bcee61,dataout=da8ae2b5,empty=0, full=0 PASS
914 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=d0f578a1,dataout=da8ae2b5,empty=0, full=0 PASS
918 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=a8639650,dataout=da8ae2b5,empty=0, full=0 PASS
922 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9ab48835,dataout=da8ae2b5,empty=0, full=0 PASS
926 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=60b175c1,dataout=da8ae2b5,empty=0, full=0 PASS
930 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=cc01b498,dataout=da8ae2b5,empty=0, full=0 PASS
934 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b98c4273,dataout=da8ae2b5,empty=0, full=0 PASS
938 clk_in=1, clk_out=1, reset=1, flush=1,insert=0,
remove=0, datain=b98c4273,dataout=00000000,empty=0, full=0 PASS
942 clk_in=1, clk_out=1, reset=1, flush=1,insert=0,
remove=0, datain=b98c4273,dataout=00000000,empty=0, full=0 PASS
946 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b98c4273,dataout=00000000,empty=0, full=0 PASS
950 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b98c4273,dataout=00000000,empty=1, full=0 PASS
954 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b98c4273,dataout=00000000,empty=1, full=0 PASS
958 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f622e6ec,dataout=00000000,empty=1, full=0 PASS

```

```

          962 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f622e6ec,dataout=00000000,empty=1, full=0 PASS
          966 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=d44b80a8,dataout=00000000,empty=1, full=0 PASS
          970 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d44b80a8,dataout=00000000,empty=1, full=0 PASS
          974 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=d44b80a8,dataout=00000000,empty=1, full=0 PASS
          978 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=070bb90e,dataout=00000000,empty=1, full=0 PASS
          982 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=070bb90e,dataout=d44b80a8,empty=0, full=0 PASS
          986 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=070bb90e,dataout=d44b80a8,empty=0, full=0 PASS
          990 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=152fb52a,dataout=d44b80a8,empty=0, full=0 PASS
          994 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=152fb52a,dataout=d44b80a8,empty=1, full=0 PASS
          998 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=152fb52a,dataout=d44b80a8,empty=1, full=0 PASS
         1002 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=4f75ff9e,dataout=d44b80a8,empty=1, full=0 PASS
         1006 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=4f75ff9e,dataout=d44b80a8,empty=1, full=0 PASS
         1010 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=4f75ff9e,dataout=d44b80a8,empty=1, full=0 PASS
         1014 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6464e3c8,dataout=d44b80a8,empty=1, full=0 PASS
         1018 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6464e3c8,dataout=d44b80a8,empty=1, full=0 PASS
         1022 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=6464e3c8,dataout=d44b80a8,empty=1, full=0 PASS
         1026 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=35a0c96b,dataout=d44b80a8,empty=1, full=0 PASS
         1030 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=35a0c96b,dataout=d44b80a8,empty=1, full=0 PASS
         1034 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=35a0c96b,dataout=d44b80a8,empty=1, full=0 PASS
         1038 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5d059dba,dataout=d44b80a8,empty=1, full=0 PASS
         1042 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5d059dba,dataout=d44b80a8,empty=0, full=0 PASS
         1046 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=5d059dba,dataout=d44b80a8,empty=0, full=0 PASS
         1050 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=492fd392,dataout=d44b80a8,empty=0, full=0 PASS
         1054 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=492fd392,dataout=35a0c96b,empty=0, full=0 PASS
         1058 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=492fd392,dataout=35a0c96b,empty=0, full=0 PASS
         1062 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=c3339086,dataout=35a0c96b,empty=0, full=0 PASS
         1066 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=c3339086,dataout=35a0c96b,empty=1, full=0 PASS

```

```

1070 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=c3339086,dataout=35a0c96b,empty=1, full=0 PASS
1074 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=19452132,dataout=35a0c96b,empty=1, full=0 PASS
1078 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=19452132,dataout=35a0c96b,empty=1, full=0 PASS
1082 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=19452132,dataout=35a0c96b,empty=1, full=0 PASS
1086 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f249a4e4,dataout=35a0c96b,empty=1, full=0 PASS
1090 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f249a4e4,dataout=35a0c96b,empty=0, full=0 PASS
1094 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=f249a4e4,dataout=35a0c96b,empty=0, full=0 PASS
1098 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d095a8a1,dataout=35a0c96b,empty=0, full=0 PASS
1102 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d095a8a1,dataout=19452132,empty=0, full=0 PASS
1106 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=d095a8a1,dataout=19452132,empty=0, full=0 PASS
1110 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=85e51e0b,dataout=19452132,empty=0, full=0 PASS
1114 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=85e51e0b,dataout=19452132,empty=1, full=0 PASS
1118 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=85e51e0b,dataout=19452132,empty=1, full=0 PASS
1122 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=1b60e536,dataout=19452132,empty=1, full=0 PASS
1126 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1b60e536,dataout=85e51e0b,empty=0, full=0 PASS
1130 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=1b60e536,dataout=85e51e0b,empty=0, full=0 PASS
1134 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=35cdbf6b,dataout=85e51e0b,empty=0, full=0 PASS
1138 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=35cdbf6b,dataout=1b60e536,empty=0, full=0 PASS
1142 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=35cdbf6b,dataout=1b60e536,empty=0, full=0 PASS
1146 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=4df3819b,dataout=1b60e536,empty=0, full=0 PASS
1150 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=4df3819b,dataout=1b60e536,empty=0, full=0 PASS
1154 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=4df3819b,dataout=1b60e536,empty=0, full=0 PASS
1158 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9684e02d,dataout=1b60e536,empty=0, full=0 PASS
1162 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=9684e02d,dataout=1b60e536,empty=0, full=0 PASS
1166 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=9684e02d,dataout=1b60e536,empty=0, full=0 PASS
1170 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=8f1cf61e,dataout=1b60e536,empty=0, full=0 PASS
1174 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=8f1cf61e,dataout=1b60e536,empty=0, full=0 PASS

```

```

1178 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=8f1cf61e,dataout=1b60e536,empty=0, full=0 PASS
1182 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=0c039d18,dataout=1b60e536,empty=0, full=0 PASS
1186 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0c039d18,dataout=1b60e536,empty=0, full=0 PASS
1190 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0c039d18,dataout=1b60e536,empty=0, full=0 PASS
1194 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=a0c02441,dataout=1b60e536,empty=0, full=0 PASS
1198 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a0c02441,dataout=1b60e536,empty=0, full=0 PASS
1202 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a0c02441,dataout=1b60e536,empty=0, full=0 PASS
1206 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=29efe953,dataout=1b60e536,empty=0, full=0 PASS
1210 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=29efe953,dataout=1b60e536,empty=0, full=0 PASS
1214 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=29efe953,dataout=1b60e536,empty=0, full=0 PASS
1218 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f166fae2,dataout=1b60e536,empty=0, full=0 PASS
1222 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f166fae2,dataout=9684e02d,empty=0, full=0 PASS
1226 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=f166fae2,dataout=9684e02d,empty=0, full=0 PASS
1230 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ec50b4d8,dataout=9684e02d,empty=0, full=0 PASS
1234 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ec50b4d8,dataout=8f1cf61e,empty=0, full=0 PASS
1238 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ec50b4d8,dataout=8f1cf61e,empty=0, full=0 PASS
1242 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=9c811239,dataout=8f1cf61e,empty=0, full=0 PASS
1246 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=9c811239,dataout=8f1cf61e,empty=0, full=0 PASS
1250 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=9c811239,dataout=8f1cf61e,empty=0, full=0 PASS
1254 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=15890f2b,dataout=8f1cf61e,empty=0, full=0 PASS
1258 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=15890f2b,dataout=0c039d18,empty=0, full=0 PASS
1262 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=15890f2b,dataout=0c039d18,empty=0, full=0 PASS
1266 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=13b55527,dataout=0c039d18,empty=0, full=0 PASS
1270 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=13b55527,dataout=0c039d18,empty=0, full=0 PASS
1274 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=13b55527,dataout=0c039d18,empty=0, full=0 PASS
1278 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=82223a04,dataout=0c039d18,empty=0, full=0 PASS
1282 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=82223a04,dataout=a0c02441,empty=0, full=0 PASS

```

```

1286 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=82223a04,dataout=a0c02441,empty=0, full=0 PASS
1290 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0a6e9314,dataout=a0c02441,empty=0, full=0 PASS
1294 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0a6e9314,dataout=a0c02441,empty=0, full=0 PASS
1298 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0a6e9314,dataout=a0c02441,empty=0, full=0 PASS
1302 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=d8ace2b1,dataout=a0c02441,empty=0, full=0 PASS
1306 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d8ace2b1,dataout=a0c02441,empty=0, full=0 PASS
1310 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=d8ace2b1,dataout=a0c02441,empty=0, full=0 PASS
1314 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=158b2b2b,dataout=a0c02441,empty=0, full=0 PASS
1318 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=158b2b2b,dataout=9c811239,empty=0, full=0 PASS
1322 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=158b2b2b,dataout=9c811239,empty=0, full=0 PASS
1326 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=93c12227,dataout=9c811239,empty=0, full=0 PASS
1330 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=93c12227,dataout=15890f2b,empty=0, full=0 PASS
1334 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=93c12227,dataout=15890f2b,empty=0, full=0 PASS
1338 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=f3d7a6e7,dataout=15890f2b,empty=0, full=0 PASS
1342 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f3d7a6e7,dataout=13b55527,empty=0, full=0 PASS
1346 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=f3d7a6e7,dataout=13b55527,empty=0, full=0 PASS
1350 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=6de5bbdb,dataout=13b55527,empty=0, full=0 PASS
1354 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6de5bbdb,dataout=d8ace2b1,empty=0, full=0 PASS
1358 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=6de5bbdb,dataout=d8ace2b1,empty=0, full=0 PASS
1362 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d0bc5ea1,dataout=d8ace2b1,empty=0, full=0 PASS
1366 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d0bc5ea1,dataout=158b2b2b,empty=0, full=0 PASS
1370 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d0bc5ea1,dataout=158b2b2b,empty=0, full=0 PASS
1374 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=a2e62045,dataout=158b2b2b,empty=0, full=0 PASS
1378 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a2e62045,dataout=158b2b2b,empty=0, full=0 PASS
1382 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a2e62045,dataout=158b2b2b,empty=0, full=0 PASS
1386 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b9461472,dataout=158b2b2b,empty=0, full=0 PASS
1390 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b9461472,dataout=158b2b2b,empty=0, full=0 PASS

```

```

1394 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b9461472,dataout=158b2b2b,empty=0, full=0 PASS
1398 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b7dfaa6f,dataout=158b2b2b,empty=0, full=0 PASS
1402 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b7dfaa6f,dataout=158b2b2b,empty=0, full=0 PASS
1406 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b7dfaa6f,dataout=158b2b2b,empty=0, full=0 PASS
1410 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1c719738,dataout=158b2b2b,empty=0, full=0 PASS
1414 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1c719738,dataout=158b2b2b,empty=0, full=0 PASS
1418 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1c719738,dataout=158b2b2b,empty=0, full=0 PASS
1422 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=7b0da9f6,dataout=158b2b2b,empty=0, full=0 PASS
1426 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=7b0da9f6,dataout=158b2b2b,empty=0, full=0 PASS
1430 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=7b0da9f6,dataout=158b2b2b,empty=0, full=0 PASS
1434 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=3a625f74,dataout=158b2b2b,empty=0, full=0 PASS
1438 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=3a625f74,dataout=158b2b2b,empty=0, full=0 PASS
1442 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=3a625f74,dataout=158b2b2b,empty=0, full=0 PASS
1446 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1e1c873c,dataout=158b2b2b,empty=0, full=0 PASS
1450 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1e1c873c,dataout=158b2b2b,empty=0, full=0 PASS
1454 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1e1c873c,dataout=158b2b2b,empty=0, full=0 PASS
1458 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=0aec3515,dataout=158b2b2b,empty=0, full=0 PASS
1462 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0aec3515,dataout=158b2b2b,empty=0, full=0 PASS
1466 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=0aec3515,dataout=158b2b2b,empty=0, full=0 PASS
1470 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=a18bee43,dataout=158b2b2b,empty=0, full=0 PASS
1474 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a18bee43,dataout=f3d7a6e7,empty=0, full=0 PASS
1478 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a18bee43,dataout=f3d7a6e7,empty=0, full=0 PASS
1482 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=1297cb25,dataout=f3d7a6e7,empty=0, full=0 PASS
1486 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1297cb25,dataout=f3d7a6e7,empty=0, full=0 PASS
1490 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=1297cb25,dataout=f3d7a6e7,empty=0, full=0 PASS
1494 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ad67e25a,dataout=f3d7a6e7,empty=0, full=0 PASS
1498 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ad67e25a,dataout=6de5bbdb,empty=0, full=0 PASS

```



```

1502 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ad67e25a,dataout=6de5bbdb,empty=0, full=0 PASS
1506 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=060a5d0c,dataout=6de5bbdb,empty=0, full=0 PASS
1510 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=060a5d0c,dataout=6de5bbdb,empty=0, full=0 PASS
1514 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=060a5d0c,dataout=6de5bbdb,empty=0, full=0 PASS
1518 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=5b60e5b6,dataout=6de5bbdb,empty=0, full=0 PASS
1522 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5b60e5b6,dataout=a2e62045,empty=0, full=0 PASS
1526 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5b60e5b6,dataout=a2e62045,empty=0, full=0 PASS
1530 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ae78585c,dataout=a2e62045,empty=0, full=0 PASS
1534 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ae78585c,dataout=a2e62045,empty=0, full=0 PASS
1538 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ae78585c,dataout=a2e62045,empty=0, full=0 PASS
1542 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d00b12a0,dataout=a2e62045,empty=0, full=0 PASS
1546 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d00b12a0,dataout=a2e62045,empty=0, full=0 PASS
1550 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d00b12a0,dataout=a2e62045,empty=0, full=0 PASS
1554 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=6e8af5dd,dataout=a2e62045,empty=0, full=0 PASS
1558 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6e8af5dd,dataout=a2e62045,empty=0, full=0 PASS
1562 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=6e8af5dd,dataout=a2e62045,empty=0, full=0 PASS
1566 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=bccc4279,dataout=a2e62045,empty=0, full=0 PASS
1570 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=bccc4279,dataout=7b0da9f6,empty=0, full=0 PASS
1574 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=bccc4279,dataout=7b0da9f6,empty=0, full=0 PASS
1578 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=bde0d27b,dataout=7b0da9f6,empty=0, full=0 PASS
1582 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=bde0d27b,dataout=3a625f74,empty=0, full=0 PASS
1586 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=bde0d27b,dataout=3a625f74,empty=0, full=0 PASS
1590 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=71c129e3,dataout=3a625f74,empty=0, full=0 PASS
1594 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=71c129e3,dataout=0aec3515,empty=0, full=0 PASS
1598 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=71c129e3,dataout=0aec3515,empty=0, full=0 PASS
1602 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=22119f44,dataout=0aec3515,empty=0, full=0 PASS
1606 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=22119f44,dataout=a18bee43,empty=0, full=0 PASS

```

```

1610 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=22119f44,dataout=a18bee43,empty=0, full=0 PASS
1614 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f6a178ed,dataout=a18bee43,empty=0, full=0 PASS
1618 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f6a178ed,dataout=a18bee43,empty=0, full=0 PASS
1622 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=f6a178ed,dataout=a18bee43,empty=0, full=0 PASS
1626 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=46dcb78d,dataout=a18bee43,empty=0, full=0 PASS
1630 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=46dcb78d,dataout=a18bee43,empty=0, full=0 PASS
1634 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=46dcb78d,dataout=a18bee43,empty=0, full=0 PASS
1638 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=236afd46,dataout=a18bee43,empty=0, full=0 PASS
1642 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=236afd46,dataout=a18bee43,empty=0, full=0 PASS
1646 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=236afd46,dataout=a18bee43,empty=0, full=0 PASS
1650 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0beac117,dataout=a18bee43,empty=0, full=0 PASS
1654 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0beac117,dataout=a18bee43,empty=0, full=0 PASS
1658 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=0beac117,dataout=a18bee43,empty=0, full=0 PASS
1662 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d55bbcaa,dataout=a18bee43,empty=0, full=0 PASS
1666 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d55bbcaa,dataout=a18bee43,empty=0, full=0 PASS
1670 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d55bbcaa,dataout=a18bee43,empty=0, full=0 PASS
1674 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=5d4a4dba,dataout=a18bee43,empty=0, full=0 PASS
1678 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5d4a4dba,dataout=a18bee43,empty=0, full=0 PASS
1682 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5d4a4dba,dataout=a18bee43,empty=0, full=0 PASS
1686 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=92831e25,dataout=a18bee43,empty=0, full=0 PASS
1690 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=92831e25,dataout=a18bee43,empty=0, full=0 PASS
1694 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=92831e25,dataout=a18bee43,empty=0, full=0 PASS
1698 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a48f7c49,dataout=a18bee43,empty=0, full=0 PASS
1702 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=a48f7c49,dataout=a18bee43,empty=0, full=0 PASS
1706 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=a48f7c49,dataout=a18bee43,empty=0, full=0 PASS
1710 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=25f2034b,dataout=a18bee43,empty=0, full=0 PASS
1714 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=25f2034b,dataout=1297cb25,empty=0, full=0 PASS

```

```

1718 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=25f2034b,dataout=1297cb25,empty=0, full=0 PASS
1722 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=433e9786,dataout=1297cb25,empty=0, full=0 PASS
1726 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=433e9786,dataout=ad67e25a,empty=0, full=0 PASS
1730 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=433e9786,dataout=ad67e25a,empty=0, full=0 PASS
1734 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6851e5d0,dataout=ad67e25a,empty=0, full=0 PASS
1738 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6851e5d0,dataout=ad67e25a,empty=0, full=0 PASS
1742 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6851e5d0,dataout=ad67e25a,empty=0, full=0 PASS
1746 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=03e9b707,dataout=ad67e25a,empty=0, full=0 PASS
1750 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=03e9b707,dataout=ad67e25a,empty=0, full=0 PASS
1754 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=03e9b707,dataout=ad67e25a,empty=0, full=0 PASS
1758 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=746affe8,dataout=ad67e25a,empty=0, full=0 PASS
1762 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=746affe8,dataout=060a5d0c,empty=0, full=0 PASS
1766 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=746affe8,dataout=060a5d0c,empty=0, full=0 PASS
1770 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=76295bec,dataout=060a5d0c,empty=0, full=0 PASS
1774 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=76295bec,dataout=5b60e5b6,empty=0, full=0 PASS
1778 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=76295bec,dataout=5b60e5b6,empty=0, full=0 PASS
1782 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=64e165c9,dataout=5b60e5b6,empty=0, full=0 PASS
1786 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=64e165c9,dataout=5b60e5b6,empty=0, full=0 PASS
1790 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=64e165c9,dataout=5b60e5b6,empty=0, full=0 PASS
1794 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ea5814d4,dataout=5b60e5b6,empty=0, full=0 PASS
1798 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ea5814d4,dataout=ae78585c,empty=0, full=0 PASS
1802 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=ea5814d4,dataout=ae78585c,empty=0, full=0 PASS
1806 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=583125b0,dataout=ae78585c,empty=0, full=0 PASS
1810 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=583125b0,dataout=6e8af5dd,empty=0, full=0 PASS
1814 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=583125b0,dataout=6e8af5dd,empty=0, full=0 PASS
1818 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ecb91ad9,dataout=6e8af5dd,empty=0, full=0 PASS
1822 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ecb91ad9,dataout=bde0d27b,empty=0, full=0 PASS

```

```

1826 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ecb91ad9,dataout=bde0d27b,empty=0, full=0 PASS
1830 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=49b16f93,dataout=bde0d27b,empty=0, full=0 PASS
1834 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=49b16f93,dataout=bde0d27b,empty=0, full=0 PASS
1838 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=49b16f93,dataout=bde0d27b,empty=0, full=0 PASS
1842 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=e471f8c8,dataout=bde0d27b,empty=0, full=0 PASS
1846 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=e471f8c8,dataout=bde0d27b,empty=0, full=0 PASS
1850 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=e471f8c8,dataout=bde0d27b,empty=0, full=0 PASS
1854 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=4226a984,dataout=bde0d27b,empty=0, full=0 PASS
1858 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=4226a984,dataout=bde0d27b,empty=0, full=0 PASS
1862 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=4226a984,dataout=bde0d27b,empty=0, full=0 PASS
1866 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=897f1c12,dataout=bde0d27b,empty=0, full=0 PASS
1870 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=897f1c12,dataout=71c129e3,empty=0, full=0 PASS
1874 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=897f1c12,dataout=71c129e3,empty=0, full=0 PASS
1878 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=e82b96d0,dataout=71c129e3,empty=0, full=0 PASS
1882 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=e82b96d0,dataout=22119f44,empty=0, full=0 PASS
1886 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=e82b96d0,dataout=22119f44,empty=0, full=0 PASS
1890 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6d8b87db,dataout=22119f44,empty=0, full=0 PASS
1894 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6d8b87db,dataout=5d4a4dba,empty=0, full=0 PASS
1898 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=6d8b87db,dataout=5d4a4dba,empty=0, full=0 PASS
1902 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=80797c00,dataout=5d4a4dba,empty=0, full=0 PASS
1906 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=80797c00,dataout=433e9786,empty=0, full=0 PASS
1910 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=80797c00,dataout=433e9786,empty=0, full=0 PASS
1914 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=8653620c,dataout=433e9786,empty=0, full=0 PASS
1918 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=8653620c,dataout=746affe8,empty=0, full=0 PASS
1922 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=8653620c,dataout=746affe8,empty=0, full=0 PASS
1926 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=67d735cf,dataout=746affe8,empty=0, full=0 PASS
1930 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=67d735cf,dataout=76295bec,empty=0, full=0 PASS

```

```

1934 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=67d735cf,dataout=76295bec,empty=0, full=0 PASS
1938 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b4f9a469,dataout=76295bec,empty=0, full=0 PASS
1942 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b4f9a469,dataout=64e165c9,empty=0, full=0 PASS
1946 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b4f9a469,dataout=64e165c9,empty=0, full=0 PASS
1950 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=ec3758d8,dataout=64e165c9,empty=0, full=0 PASS
1954 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ec3758d8,dataout=64e165c9,empty=0, full=0 PASS
1958 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=ec3758d8,dataout=64e165c9,empty=0, full=0 PASS
1962 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=5c78b1b8,dataout=64e165c9,empty=0, full=0 PASS
1966 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5c78b1b8,dataout=64e165c9,empty=0, full=0 PASS
1970 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=5c78b1b8,dataout=64e165c9,empty=0, full=0 PASS
1974 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=984d5a30,dataout=64e165c9,empty=0, full=0 PASS
1978 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=984d5a30,dataout=64e165c9,empty=0, full=0 PASS
1982 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=984d5a30,dataout=64e165c9,empty=0, full=0 PASS
1986 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=6a15f5d4,dataout=64e165c9,empty=0, full=0 PASS
1990 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6a15f5d4,dataout=ea5814d4,empty=0, full=0 PASS
1994 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6a15f5d4,dataout=ea5814d4,empty=0, full=0 PASS
1998 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=74a1ade9,dataout=ea5814d4,empty=0, full=0 PASS
2002 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=74a1ade9,dataout=ea5814d4,empty=0, full=0 PASS
2006 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=74a1ade9,dataout=ea5814d4,empty=0, full=0 PASS
2010 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6d808bdb,dataout=ea5814d4,empty=0, full=0 PASS
2014 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6d808bdb,dataout=49b16f93,empty=0, full=0 PASS
2018 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6d808bdb,dataout=49b16f93,empty=0, full=0 PASS
2022 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=e2ecdac5,dataout=49b16f93,empty=0, full=0 PASS
2026 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=e2ecdac5,dataout=49b16f93,empty=0, full=0 PASS
2030 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=e2ecdac5,dataout=49b16f93,empty=0, full=0 PASS
2034 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=b302da66,dataout=49b16f93,empty=0, full=0 PASS
2038 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=b302da66,dataout=e471f8c8,empty=0, full=0 PASS

```

```

2042 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=b302da66,dataout=e471f8c8,empty=0, full=0 PASS
2046 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=7c41aff8,dataout=e471f8c8,empty=0, full=0 PASS
2050 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=7c41aff8,dataout=4226a984,empty=0, full=0 PASS
2054 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=7c41aff8,dataout=4226a984,empty=0, full=0 PASS
2058 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=70ef37e1,dataout=4226a984,empty=0, full=0 PASS
2062 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=70ef37e1,dataout=80797c00,empty=0, full=0 PASS
2066 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=70ef37e1,dataout=80797c00,empty=0, full=0 PASS
2070 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=304e4d60,dataout=80797c00,empty=0, full=0 PASS
2074 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=304e4d60,dataout=80797c00,empty=0, full=0 PASS
2078 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=304e4d60,dataout=80797c00,empty=0, full=0 PASS
2082 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=322f7d64,dataout=80797c00,empty=0, full=0 PASS
2086 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=322f7d64,dataout=80797c00,empty=0, full=0 PASS
2090 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=322f7d64,dataout=80797c00,empty=0, full=0 PASS
2094 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=bbbbc5277,dataout=80797c00,empty=0, full=0 PASS
2098 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=bbbbc5277,dataout=8653620c,empty=0, full=0 PASS
2102 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=bbbbc5277,dataout=8653620c,empty=0, full=0 PASS
2106 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6a9fb9d5,dataout=8653620c,empty=0, full=0 PASS
2110 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6a9fb9d5,dataout=b4f9a469,empty=0, full=0 PASS
2114 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=6a9fb9d5,dataout=b4f9a469,empty=0, full=0 PASS
2118 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=d57800aa,dataout=b4f9a469,empty=0, full=0 PASS
2122 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=d57800aa,dataout=b4f9a469,empty=0, full=0 PASS
2126 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=d57800aa,dataout=b4f9a469,empty=0, full=0 PASS
2130 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=be9bbc7d,dataout=b4f9a469,empty=0, full=0 PASS
2134 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=be9bbc7d,dataout=ec3758d8,empty=0, full=0 PASS
2138 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=be9bbc7d,dataout=ec3758d8,empty=0, full=0 PASS
2142 clk_in=1, clk_out=1, reset=1, flush=0,insert=1,
remove=0, datain=1e664d3c,dataout=ec3758d8,empty=0, full=0 PASS
2146 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=0, datain=1e664d3c,dataout=ec3758d8,empty=0, full=0 PASS

```

2150 clk_in=1, clk_out=1, reset=1, flush=0,insert=0,
remove=1, datain=1e664d3c,dataout=ec3758d8,empty=0, full=0 PASS

2154 clk_in=1, clk_out=1, reset=0, flush=0,insert=0,
remove=0, datain=1e664d3c,dataout=00000000,empty=1, full=0 PASS

\$finish called from file "fifo_fixture.v", line 108.

\$finish at simulation time 2157

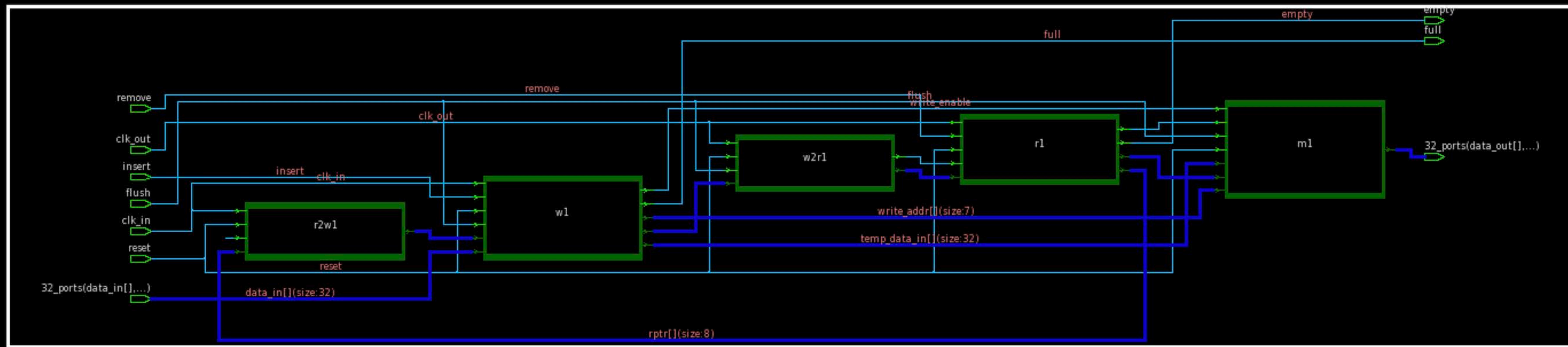
V C S S i m u l a t i o n R e p o r t

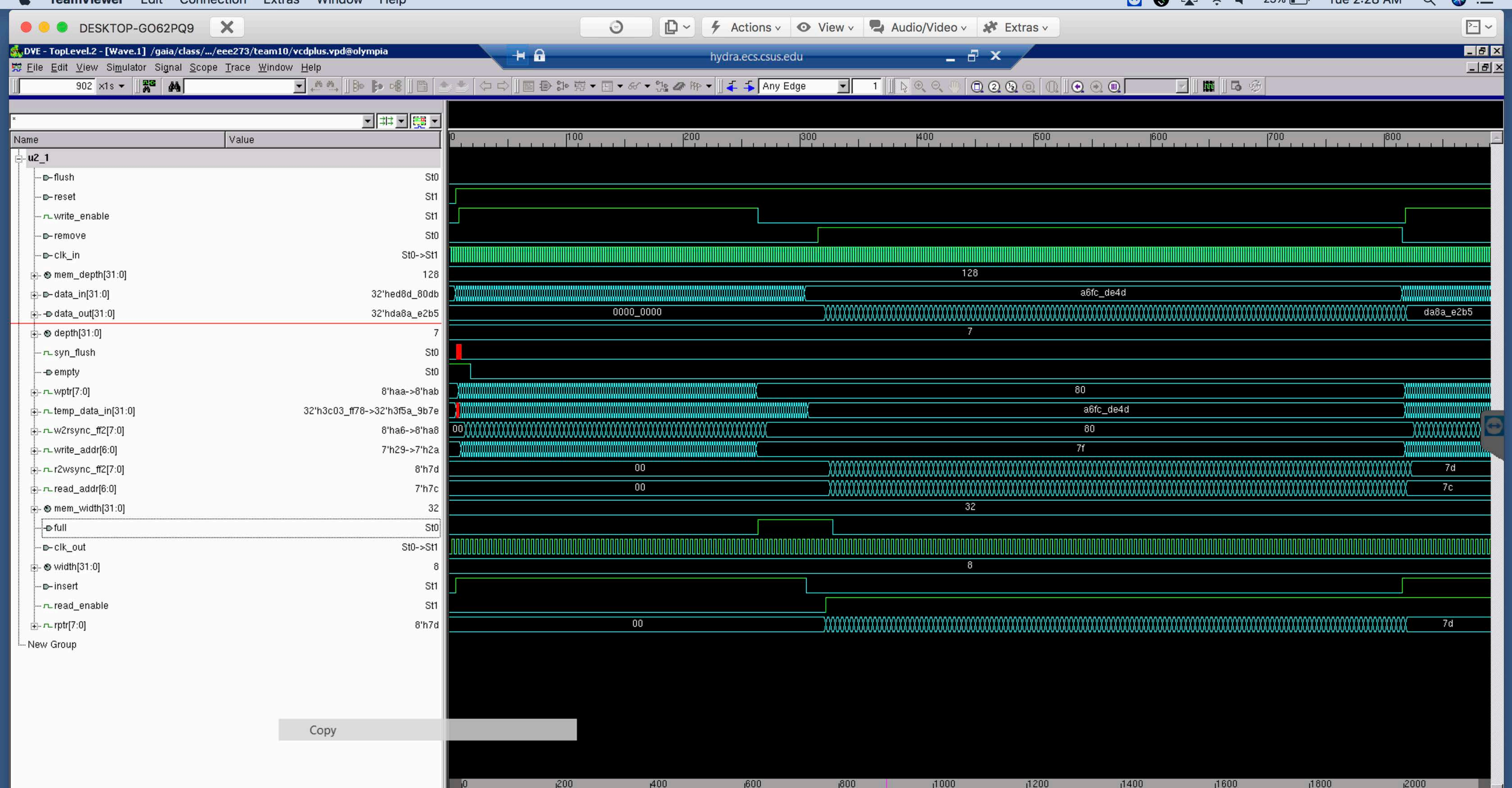
Time: 2157

CPU Time: 0.350 seconds; Data structure size: 0.0Mb

Mon Apr 29 19:55:09 2019

```
read_file -format verilog {"fif0.v"}
analyze -format verilog {fif0.v}
elaborate fif0 -architecture verilog
set current_design fif0
link
create_clock "clk_in" -period 1 -name "clk_in"
set_input_delay -clock clk_in -max -rise 0.4 "insert"
set_input_delay -clock clk_in -min -rise 0.2 "insert"
set_input_delay -clock clk_in -max -rise 0.4 "flush"
set_input_delay -clock clk_in -min -rise 0.2 "flush"
set_input_delay -clock clk_in -max -rise 0.4 "data_in"
set_input_delay -clock clk_in -min -rise 0.2 "data_in"
set_output_delay -clock clk_in -max -rise 0.4 "full"
set_output_delay -clock clk_in -min -rise 0.2 "full"
create_clock "clk_out" -period 2 -name "clk_out"
set_input_delay -clock clk_out -max -rise 0.4 "remove"
set_input_delay -clock clk_out -min -rise 0.2 "remove"
set_output_delay -clock clk_out -max -rise 0.4 "empty"
set_output_delay -clock clk_out -min -rise 0.2 "empty"
set_output_delay -clock clk_out -max -rise 0.4 "data_out"
set_output_delay -clock clk_out -min -rise 0.2 "data_out"
set_dont_touch_network "clk_in"
set_dont_touch_network "clk_out"
set_max_area 0
set_false_path -from [get_clocks {clk_in}] -to [get_clocks {clk_out}]
set_operating_conditions -library "saed90nm_typ" "TYPICAL"
compile -map_effort high -boundary_optimization
report_attribute > report1
report_area > report2
report_constraints -all_violators > report3
report_timing -path full -delay max -max_paths 1 -nworst 1 > report4
report_timing -path full -delay min -max_paths 1 -nworst 1 > report5
```



Area Report

Report : area

Design : fifo

Version: I-2013.12-SP5-4

Date : Tue Apr 30 01:15:05 2019

Information: Updating design information... (UID-85)

Library(s) Used:

saed90nm_typ (File:
/netdisk/tmp/saed/SAED90_EDK/SAED_EDK90nm/Digital_Standard_cell_Library/s
ynopsys/models/saed90nm_typ.db)

Number of ports:	72
Number of nets:	154
Number of cells:	5
Number of combinational cells:	0
Number of sequential cells:	0
Number of macros/black boxes:	0
Number of buf/inv:	0
Number of references:	5

Combinational area:	128402.843143
Buf/Inv area:	12374.323473
Noncombinational area:	95447.349611
Macro/Black Box area:	0.000000
Net Interconnect area:	23500.373792

Total cell area:	223850.192755
Total area:	247350.566547

1

Constraint Report

Report : constraint

-all_violators

Design : fifo

Version: I-2013.12-SP5-4

Date : Tue Apr 30 01:15:06 2019

max_area

Design	Required Area	Actual Area	Slack
-----	-----	-----	-----
fifo	0.00	247350.56	-247350.56

(VIOLATED)

Timing Report

Report : timing

-path full

-delay max

-max_paths 1

Design : fifo

Version: I-2013.12-SP5-4

Date : Tue Apr 30 01:15:06 2019

Operating Conditions: TYPICAL Library: saed90nm_typ

Wire Load Model Mode: enclosed

Startpoint: r2w1/r2wsync_ff2_reg[4]

(rising edge-triggered flip-flop clocked by clk_in)

Endpoint: w1/wp1r_reg[3]

(rising edge-triggered flip-flop clocked by clk_in)

Path Group: clk_in

Path Type: max

Des/Clust/Port	Wire Load Model	Library
-----	-----	-----
fifo	280000	saed90nm_typ
write_logic_depth7_width8	8000	saed90nm_typ

Point	Incr	Path
-----	-----	-----

clock clk_in (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
r2w1/r2wsync_ff2_reg[4]/CLK (DFFARX1)	0.00	0.00
r		
r2w1/r2wsync_ff2_reg[4]/Q (DFFARX1)	0.19	0.19
f		
r2w1/r2wsync_ff2[4] (sync_rd2wr_width8_depth7)	0.00	0.19
f		
w1/r2wsync_ff2[4] (write_logic_depth7_width8)	0.00	0.19
f		
w1/U66/Q (XOR2X1)	0.83	1.02
f		
w1/U50/Q (AND4X1)	0.13	1.15
f		
w1/U59/QN (NAND2X0)	0.07	1.22
r		
w1/U5/Q (AND3X1)	0.10	1.32
r		
w1/U90/Q (OA21X1)	0.10	1.42
r		
w1/U55/Q (AO22X1)	0.11	1.53
r		

w1/wptra_reg[3]/D (DFFARX1)	0.03	1.56
r		
data arrival time		1.56
clock clk_in (rise edge)	1.65	1.65
clock network delay (ideal)	0.00	1.65
w1/wptra_reg[3]/CLK (DFFARX1)	0.00	1.65
r		
library setup time	-0.09	1.56
data required time		1.56

data required time		1.56
data arrival time		-1.56

slack (MET)		0.00

Startpoint: r1/rptra_reg[0]
 (rising edge-triggered flip-flop clocked by clk_out)
 Endpoint: r1/rptra_reg[0]
 (rising edge-triggered flip-flop clocked by clk_out)
 Path Group: clk_out
 Path Type: max

Des/Clust/Port	Wire Load Model	Library
fifo	280000	saed90nm_typ
read_logic_depth7_width8	8000	saed90nm_typ

Point	Incr	Path
clock clk_out (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
r1/rptra_reg[0]/CLK (DFFARX1)	0.00	0.00 r
r1/rptra_reg[0]/Q (DFFARX1)	0.19	0.19 r
r1/U47/Q (XOR2X1)	1.56	1.75 r
r1/U43/QN (NOR4X0)	0.11	1.86 f
r1/U42/QN (NAND4X0)	0.12	1.98 r
r1/U41/QN (NOR2X0)	0.12	2.10 f
r1/U28/QN (NOR3X0)	0.12	2.22 r
r1/U27/Q (AO21X1)	0.16	2.38 r
r1/U24/QN (NOR2X0)	0.23	2.61 f
r1/U14/Q (OA21X1)	0.15	2.76 f
r1/U5/Q (AO22X1)	0.14	2.90 f
r1/rptra_reg[0]/D (DFFARX1)	0.03	2.94 f
data arrival time		2.94
clock clk_out (rise edge)	3.30	3.30
clock network delay (ideal)	0.00	3.30
r1/rptra_reg[0]/CLK (DFFARX1)	0.00	3.30 r
library setup time	-0.06	3.24

data required time	3.24

data required time	3.24
data arrival time	-2.94

slack (MET)	0.31

Unified Coverage Report :: Dashboard - Mozilla Firefox@olympia

Unified Coverage Report :: Dashboard

file:///gaia/class/cs2731/cs273120/fifo_project/team10/urgReport/dashboard.html

Dashboard

dashboard | hierarchy | modlist | groups | tests | asserts

Date: Mon Apr 29 23:23:21 2019

User: cs273120

Version: l-2014.03-2

Command line: urg -dir simv.vdb

Total Coverage Summary

SCORE	LINE	COND	TOGGLE	FSM
98.56	96.59	100.00	97.67	100.00

Hierarchical coverage data for top-level instances

SCORE	LINE	COND	TOGGLE	FSM	NAME
98.56	96.59	100.00	97.67	100.00	fifo_fixture

dashboard | hierarchy | modlist | groups | tests | asserts

0%

10%

20%

30%

40%

50%

60%

70%

80%

90%

100%


```

module write_logic#(parameter depth = 7, width = 8)(input
clk_in,insert,reset,flush,input [depth : 0]r2wsync_ff2,input[31:0]
data_in, output reg [depth-1 : 0]write_addr, output reg [depth : 0]wptr,
output reg write_enable,full,output reg[31:0] temp_data_in);
//parameter[1:0] idle=2'b00, s0=2'b01,s1=2'b11;
//reg[1:0] current_state, next_state;
parameter[2:0] idle=3'b001, s0=3'b010,s1=3'b100;
reg[2:0] current_state, next_state;
reg [31:0] temp4;
always@(posedge clk_in, negedge reset)
begin
    if(!reset)
        current_state<=idle;
    else if(flush)
        current_state<=idle;
    else
        current_state<=next_state;
end
always@(*)
begin

case(current_state)
    idle:
        if(insert)
            next_state=s0;
        else
            next_state=idle;
    s0:
        if(insert)
            next_state=s0;
        else
            next_state=s1;
    s1:
        if(insert)
            next_state=s0;
        else
            next_state=s1;
        default:    next_state=idle;
endcase
end
always@(posedge clk_in, negedge reset)
begin
    if(!reset)
    begin
        full<=0;
        write_enable <= 0;
        write_addr <= 0;
        wptr <=0;
    end
    else if(flush)
    begin
        full<=0;
        write_enable <= 0;
        write_addr <= 0;
    end
end

```

```

        wptr <=0;
    end
    else
    case(current_state)
        idle:
        begin
            full<=0;
            write_enable <= 0;
            write_addr <= 0;
            wptr <=0;

        end
        s0:
        begin
            if(wptr[depth-1:0] == r2wsync_ff2[depth-1:0] &&
wptr[depth] != r2wsync_ff2[depth])
            begin
                full<=1'b1;
                write_enable <= 1'b0;
                write_addr <= write_addr;
                wptr <= wptr;

            end
            else
            begin
                full<=1'b0;
                write_enable <= 1'b1;
                write_addr <= wptr[width-2:0];
                if(wptr == 8'b11111111)
                    wptr <= 8'h0;
                else begin
                    wptr <= wptr+1;
                end

            end

        end
        end
        s1:
        begin
            if(!(wptr[depth-1:0] == r2wsync_ff2[depth-1:0] &&
wptr[depth] != r2wsync_ff2[depth]))
            begin
                full<=0;
                write_enable <= 0;
                write_addr <= write_addr;
                wptr <= wptr;

            end
            else
            begin
                full<=full;
                write_enable <= 0;
                write_addr <= write_addr;
                wptr <= wptr;

            end

        end
    endcase
end

```

```
end
always @ (posedge clk_in or negedge reset)
begin
    if(!reset)
        temp_data_in<=32'h0;
    else
        begin
            temp4<=data_in;
            temp_data_in<=temp4;
        end
    end
end
endmodule
```

```

module read_logic #(parameter depth = 7, width = 8) (input
clk_out,remove,reset,syn_flush,input [depth: 0]w2rsync_ff2, output reg
[depth-1 : 0]read_addr, output reg [depth : 0]rptr, output reg
read_enable, empty );
//parameter[1:0] idle=2'b00, s0=2'b01,s1=2'b11;
//reg[1:0] current_state, next_state;
parameter[2:0] idle=3'b001, s0=3'b010,s1=3'b100;
reg[2:0] current_state, next_state;
always@(posedge clk_out, negedge reset)
begin
    if(!reset)
        current_state<=idle;
    else if(syn_flush)
        current_state<=idle;
    else
        current_state<=next_state;
end
always@(*)
begin

    case(current_state)
        idle:
            if(remove)
                next_state=s0;
            else
                next_state=idle;
        s0:
            if(remove)
                next_state=s0;
            else
                next_state=s1;
        s1:
            if(remove)
                next_state=s0;
            else
                next_state=s1;

        default:
            next_state=idle;
    endcase
end

always@(posedge clk_out, negedge reset)
begin
    if(!reset)
    begin
        empty<=1'b1;
        read_enable <= 0;
        read_addr <= 0;
        rptr <=0;
    end
    else if(syn_flush)

```

```

begin
    empty<=1'b1;
    read_enable <= 0;
    read_addr <= 0;
    rptr <=0;
end
else
case(current_state)
    idle:
    begin
        if(w2rsync_ff2[depth:0]>0)
            empty<=1'b0;
        else
        begin
            empty<=1'b1;
            read_enable <= 0;
            read_addr <= 0;
            rptr <= 0;
        end
    end
    s0:
    begin
        if(rp[depth:0] == w2rsync_ff2[depth:0])
        begin
            empty<=1'b1;
            read_enable <= 1'b0;
            read_addr <= read_addr;
            rptr <= rptr;
        end

        else
        begin
            read_addr<=rp[width-2:0];
            empty<=1'b0;
            read_enable<=1'b1;
            if(rp==8'b11111111)
                rptr<=8'h0;
            else
                rptr<=rptr+1;
        end
    end

    s1:
    begin
        if(rp[depth:0] !=w2rsync_ff2[depth:0])
            empty<=1'b0;
        else
        begin
            empty<=empty;
            rptr<=rptr;
            read_enable<=1'b0;
            read_addr<=read_addr;
        end
    end
end

```

```
endcase
end
endmodule
```

Improvised Timing Report with OneHot

Report : timing

-path full

-delay max

-max_paths 1

Design : fifo

Version: I-2013.12-SP5-4

Date : Tue Apr 30 01:00:15 2019

Operating Conditions: TYPICAL Library: saed90nm_typ

Wire Load Model Mode: enclosed

Startpoint: r2w1/r2wsync_ff2_reg[2]

(rising edge-triggered flip-flop clocked by clk_in)

Endpoint: w1/wp1tr_reg[6]

(rising edge-triggered flip-flop clocked by clk_in)

Path Group: clk_in

Path Type: max

Des/Clust/Port	Wire Load Model	Library
-----	-----	-----
fifo	280000	saed90nm_typ
write_logic_depth7_width8	8000	saed90nm_typ

Point	Incr	Path
-----	-----	-----

clock clk_in (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
r2w1/r2wsync_ff2_reg[2]/CLK (DFFARX1)	0.00	0.00
r		
r2w1/r2wsync_ff2_reg[2]/Q (DFFARX1)	0.19	0.19
f		
r2w1/r2wsync_ff2[2] (sync_rd2wr_width8_depth7)	0.00	0.19
f		
w1/r2wsync_ff2[2] (write_logic_depth7_width8)	0.00	0.19
f		
w1/U82/Q (XNOR2X2)	0.66	0.84
r		
w1/U66/QN (NOR2X0)	0.09	0.93
f		
w1/U62/QN (NAND2X0)	0.07	1.00
r		
w1/U60/QN (NOR2X0)	0.09	1.10
f		
w1/U78/QN (NAND2X0)	0.07	1.17
r		
w1/U91/QN (NAND2X0)	0.07	1.24
f		

	w1/U92/QN (AOI21X1)	0.13	1.36
r	w1/U51/Q (AO22X1)	0.12	1.48
r	w1/wptra_reg[6]/D (DFFARX1)	0.03	1.51
r	data arrival time		1.51
	clock clk_in (rise edge)	1.60	1.60
	clock network delay (ideal)	0.00	1.60
	w1/wptra_reg[6]/CLK (DFFARX1)	0.00	1.60
r	library setup time	-0.09	1.51
	data required time		1.51

---	data required time		1.51
	data arrival time		-1.51

---	slack (MET)		0.00

Startpoint: r1/rptra_reg[1]
(rising edge-triggered flip-flop clocked by clk_out)
Endpoint: r1/rptra_reg[7]
(rising edge-triggered flip-flop clocked by clk_out)
Path Group: clk_out
Path Type: max

Des/Clust/Port	Wire Load Model	Library

fifo	280000	saed90nm_typ
read_logic_depth7_width8_DW01_inc_0	ForQA	saed90nm_typ
read_logic_depth7_width8	8000	saed90nm_typ

Point	Incr	Path

clock clk_out (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
r1/rptra_reg[1]/CLK (DFFARX1)	0.00	0.00
r		
r1/rptra_reg[1]/Q (DFFARX1)	0.21	0.21
f		
r1/add_90/A[1] (read_logic_depth7_width8_DW01_inc_0)	0.00	0.21
f		
r1/add_90/U1_1_1/C1 (HADDX1)	1.50	1.71
f		
r1/add_90/U1_1_2/C1 (HADDX1)	0.15	1.85
f		

r1/add_90/U1_1_3/C1 (HADDX1)	0.15	2.00
f r1/add_90/U1_1_4/C1 (HADDX1)	0.15	2.15
f r1/add_90/U1_1_5/C1 (HADDX1)	0.15	2.29
f r1/add_90/U1_1_6/C1 (HADDX1)	0.15	2.44
r1/add_90/U2/Q (XOR2X1)	0.13	2.57
r r1/add_90/SUM[7] (read_logic_depth7_width8_DW01_inc_0)	0.00	2.57
r r1/U7/Q (AO22X1)	0.10	2.67
r r1/rptr_reg[7]/D (DFFARX1)	0.03	2.70
r data arrival time		2.70
clock clk_out (rise edge)	3.20	3.20
clock network delay (ideal)	0.00	3.20
r1/rptr_reg[7]/CLK (DFFARX1)	0.00	3.20
r library setup time	-0.09	3.11
data required time		3.11

data required time		3.11
data arrival time		-2.70

slack (MET)		0.41