California State University, Sacramento.

EEE273
Hierarchical Digital Design Methodology
Instructor: Dr. Behnam Arad
Term Project

Submitted by :-

Omkar Kachare
Date of submission: 1st May 2019

Table of Contents

Sr. No.	Content	Page No.
1	Introduction	2
2	Status report	5
3	Block diagram	7
4	Source Codes	10
5	Test Fixture	17
6	Simulation Results	20
7	Synthesis	25
8	Code Coverage	30

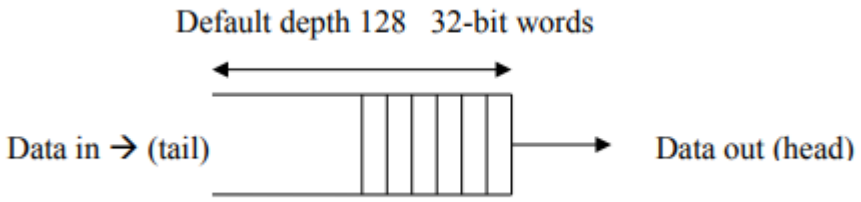
Introduction

The objective of this project is to design, validate and synthesize a parameterized first-in first-out (FIFO) buffer in Verilog. The default depth of the FIFO is 128 32-bit words. It supports flush, *insert*, and *remove*operations. During the insert operation, a new word is added to the tail (end) of the buffer and an internal write pointer is incremented on positive edge of ***clk_in*** clock. The data at the head of the FIFO is read during the remove operation, and an internal read pointer is incremented on positive edge ***clk_out***clock. A *flush* clears all words in the buffer. The buffer is considered empty after a flush. Here are the ports for the FIFO:

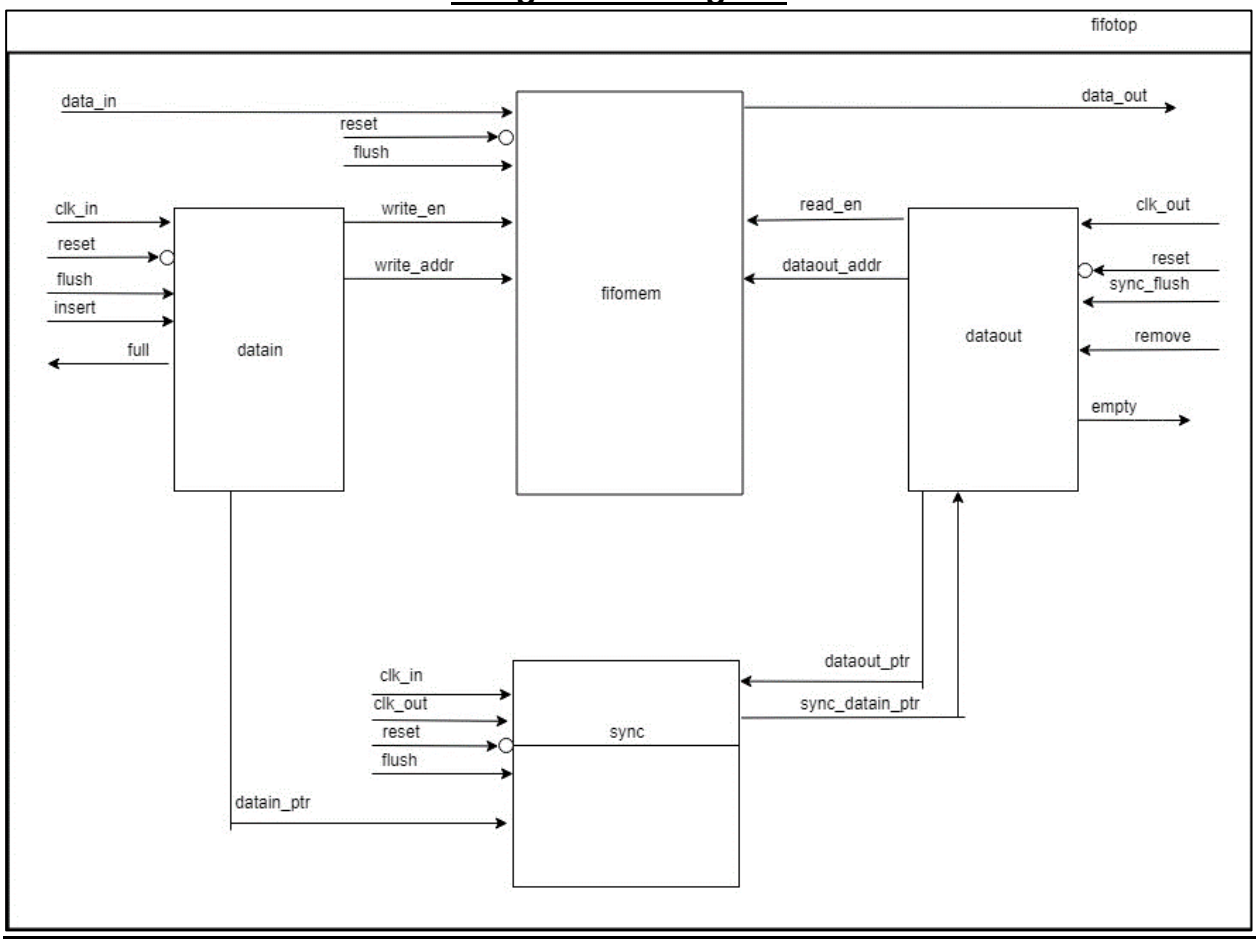
1-bit input ***reset:*** *An* asynchronous global system reset. Every sequential element in the system goes through a reset when this signal is asserted. The content of the FIFO is cleared. The status of the FIFO will be empty after a reset.

- 1-bit input ***flush***: When asserted, the FIFO is cleared, and the read/write pointers are reset on positive edge of clock ***clk_in***. The FIFO status is empty.
- 1-bit input ***insert:*** When asserted, data on ***data_in*** port is stored at the tail of the FIFO on positive edge of clock ***clk_in***
- 1-bit input ***remove***: When asserted, data at the head of the FIFO is removed from the FIFO and is placed on the on the ***data_out*** output port on positive edge of ***clk_out***
- 32-bit input ***data_in****:* input data port
- 32-bit output ***data_out****:* output data port
- 1-bit output ***full***: asserted on the positive edge of ***clk_in*** clock when the buffer is full and there is no more room for data
- 1-bit output ***empty***: asserted on the positive edge of ***clk_out*** clock when the buffer is empty (no data in the buffer)

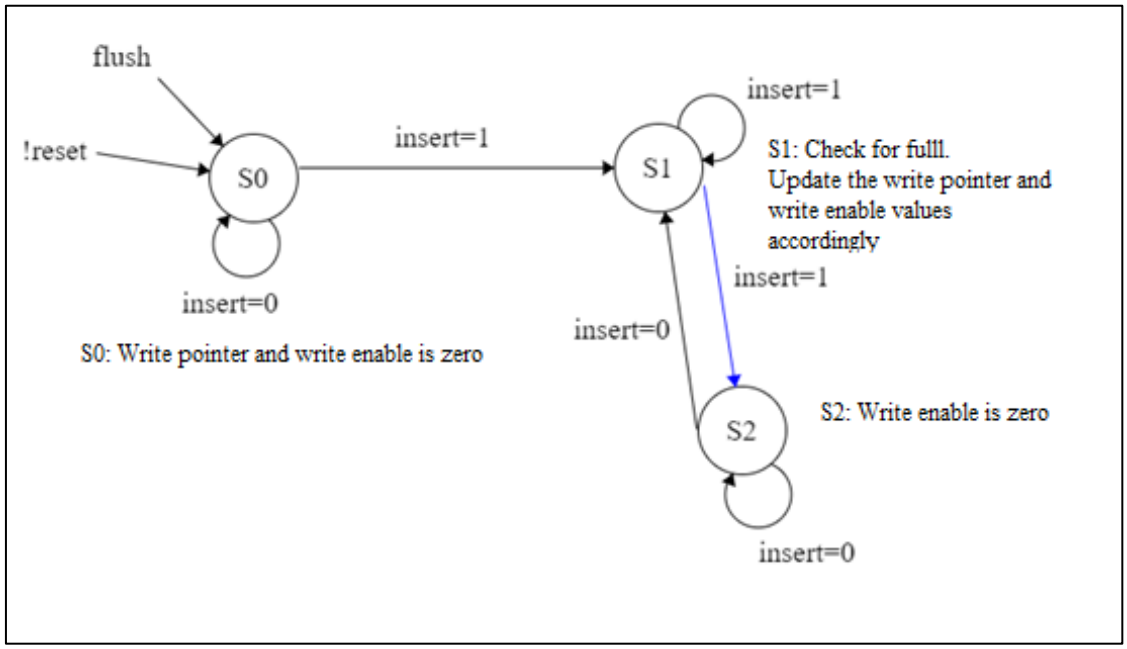
Full and empty flags are used by the external environment for proper use of the FIFO



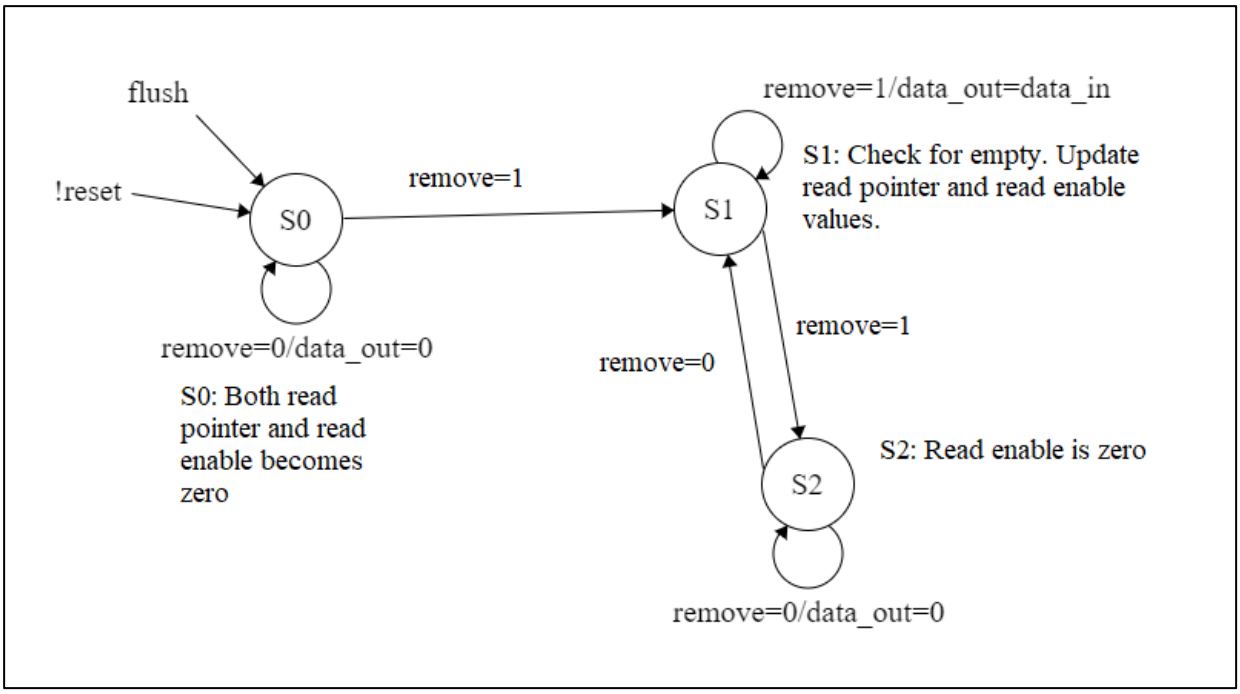
Design Block Diagram



FSM for Write Logic:



FSM for Read logic:



Source Code

1) **Memory: - fifomem.v**
2) **Write Logic: - datain.v**
3) **Read Logic: - dataout.v**
4) **Synchronizers: - sync.v**
5) **Top Module: - fifotop.v**

1) fifomem.v

```
// FIFO Memory module Depth-128,Word Size- 32.

module fifomem(reset,flush,write_en,read_en,        data_in,
      datain_addr, dataout_addr,         data_out);
parameter word=32;parameter addr=6;parameter depth=128;
input reset,flush,write_en,read_en;input [word-1:0] data_in;input [addr
:0] datain_addr, dataout_addr;output reg [word-1:0] data_out;
reg [word-1:0] mem [depth-1:0];reg [word-1:0] temp;integer i;
always @ (*)beginif (!reset)                  // Reset condition
      begin      data_out = 32'b0;      for (i=0;i<depth;i=i+1)
      begin                  mem[i] = 32'b0;
            end   end
      else if (flush)                  //Flush condition      begin
      data_out = 32'b0;      for (i=1;i<depth;i=i+1)           begin
                  mem[i] = 32'b0;
            end   end

      else if (write_en)                //Write Operationbegin
      mem[datain_addr] = data_in;  end
      else if (read_en)           //Read operation begin      data_out =
mem[dataout_addr];     end   else if (write_en && read_en)
      //Write and Read operations  begin       mem [datain_addr] =
data_in;         data_out = mem[dataout_addr];     end

endendmodule
```

2) datain.v

```
// Data_in Logic

module datain (clk_in,reset,flush,insert,
            sync_dataout_ptr,
            datain_addr,
            full, write_en,
            datain_ptr);

parameter ptr=7;
parameter addr=6;

parameter S0 = 2'b00;
parameter S1 = 2'b01;
parameter S2 = 2'b11;
```

```
input clk_in,reset,flush,insert;
input [ptr:0] sync_dataout_ptr;
output reg [addr:0] datain_addr;
output reg full, write_en;
output reg [ptr:0] datain_ptr;

reg [1:0] current_state, next_state;



always @(posedge clk_in,negedge reset)
begin
      if(!reset)
            current_state<=S0;
      else if (flush)
            current_state<=S0;
      else
            current_state<=next_state;
end



always @ (*)
begin
      case(current_state)

      S0: if(insert)
            next_state<=S1;
      else
            next_state<=S0;


      S1: if(insert)
            next_state<=S1;
      else
            next_state<=S2;


      S2: if(insert)
            next_state<=S1;
      else
            next_state<=S2;

      default: next_state<= S0;
      endcase
end


always @ (posedge clk_in,negedge reset)
begin
if(!reset)
begin
      if(!reset)
      begin
```

```
            full<=0;
            datain_ptr<=0;
            write_en<=0;
            datain_addr<=0;
      end

      if(flush)
      begin
            full<=0;
            datain_ptr<=0;
            write_en<=0;
            datain_addr<=0;
      end

      else

      case(current_state)

      S0: begin
            full<=0;
            datain_ptr<=0;
            write_en<=0;
            datain_addr<=0;
      end

      S1: begin
      if(datain_ptr[addr:0]==sync_dataout_ptr[addr:0] &&
datain_ptr[ptr]!=sync_dataout_ptr[ptr])             // Full condition
check
      begin
            full<=1'b1;
            datain_ptr<=datain_ptr;
            write_en<=1'b0;
            datain_addr<=datain_addr;
      end

      else
      begin
            datain_addr<=datain_ptr[addr:0];
            full<=1'b0;
            write_en<=1'b1;
            datain_ptr<=datain_ptr+1;
      end
      end

      S2:begin
      if(!(datain_ptr[addr:0]==sync_dataout_ptr[addr:0] &&
datain_ptr[ptr]!=sync_dataout_ptr[ptr]))
      begin
            full<=0;
            datain_ptr<=datain_ptr;
            write_en<=1'b0;
            datain_addr<=datain_addr;
      end
```

```
      else
      begin
            full<=full;
            datain_ptr<=datain_ptr;
            write_en<=1'b0;
            datain_addr<=datain_addr;
      end
      end

endcase
end
end
endmodule
```

3) dataout.v

```
// Data_out Logic

module dataout (clk_out,reset,sync_flush,remove,
            sync_datain_ptr,
            dataout_addr,
            empty, read_en,
            dataout_ptr);

parameter ptr=7;
parameter addr=6;

parameter S0 = 2'b00;
parameter S1 = 2'b01;
parameter S2 = 2'b11;

input clk_out,reset,sync_flush,remove;
input [ptr:0] sync_datain_ptr;
output reg [addr:0] dataout_addr;
output reg empty, read_en;
output reg [ptr:0] dataout_ptr;

reg [1:0] current_state, next_state;

always @(posedge clk_out,negedge reset)
begin
      if(!reset)
            current_state<=S0;
      else if (sync_flush)
            current_state<=S0;
      else
            current_state<=next_state;
end

always @ (*)
begin
      case(current_state)
```

```
      S0: if(remove)
            next_state<=S1;
      else
            next_state<=S0;


      S1: if(remove)
            next_state<=S1;
      else
            next_state<=S2;


      S2: if(remove)
            next_state<=S1;
      else
            next_state<=S2;

      default: next_state<= S0;
      endcase
end


always @ (posedge clk_out,negedge reset)
begin
if(!reset)
      begin
      if(reset)
      begin
            empty<=0;
            dataout_ptr<=0;
            read_en<=0;
            dataout_addr<=0;
      end

      else
      case(current_state)

      S0: begin
            empty<=1'b1;
            dataout_ptr<=0;
            read_en<=0;
            dataout_addr<=0;
      end

      S1: begin
      if(dataout_ptr[ptr:0]==sync_datain_ptr[ptr:0])
      begin
            empty<=1'b1;
            dataout_ptr<=dataout_ptr;
            read_en<=1'b0;
            dataout_addr<=dataout_addr;
      end
```

```
      else if (dataout_ptr[ptr:0]!=sync_datain_ptr[ptr:0])
      begin
            dataout_addr<=dataout_ptr[addr:0];
            empty<=1'b0;
            read_en<=1'b1;
            dataout_ptr<=dataout_ptr+1;
      end
      end

      S2:begin
      if(dataout_ptr[ptr:0]!=sync_datain_ptr[ptr:0])
            empty<=0;
      else
      begin
            empty<=1'b1;
            dataout_ptr<=dataout_ptr;
            read_en<=1'b0;
            dataout_addr<=dataout_addr;
      end
      end
endcase
end
end
endmodule
```

4) sync.v

```
// Sync for FIFO

module sync( clk_in,clk_out,reset,

      dataout_ptr,datain_ptr,
      sync_dataout_ptr,
      sync_datain_ptr);

parameter ptr=7;

input clk_in, clk_out,reset;
input [ptr:0] dataout_ptr,datain_ptr;
output reg [ptr:0] sync_dataout_ptr,sync_datain_ptr;

reg[ptr:0] temp1, temp2;

always @ (posedge clk_out , negedge reset)
begin
      if(!reset)
            sync_datain_ptr<=8'b0;
      else
      begin
            temp2<= datain_ptr;
            sync_datain_ptr<=temp2;
      end
```

```
end


always @ (posedge clk_in , negedge reset)
begin
      if(!reset)
      sync_dataout_ptr<=8'b0;
      else
      begin
            temp1<=dataout_ptr;
            sync_dataout_ptr<=temp1;
      end
end

endmodule
```

5) fifotop.v

```
//FIFO TOP MODULE

`include "sync.v"
`include "datain.v"
`include "dataout.v"
`include "fifomem.v"


module fifotop
(clk_in,clk_out,flush,reset,insert,remove,data_in,data_out,full,empty);

parameter ptr= 7, depth=128,word=32;

input clk_in,clk_out,flush,reset,insert,remove;
input[word-1:0] data_in;
output full,empty;
output[word-1:0]data_out;

wire[ptr:0] sync_datain_ptr, sync_dataout_ptr, datain_ptr,dataout_ptr;
wire write_en, read_en;
wire[ptr-1:0] datain_addr,dataout_addr;


fifomem F1
(.reset(reset),.flush(flush),.write_en(write_en),.read_en(read_en),.data_
in(data_in),.datain_addr(datain_addr),.dataout_addr(dataout_addr),.data_o
ut(data_out));

datain W1
(.clk_in(clk_in),.reset(reset),.flush(flush),.insert(insert),.sync_dataou
t_ptr(sync_dataout_ptr),.full(full),.write_en(write_en),.datain_addr(data
in_addr),.datain_ptr(datain_ptr));

dataout R1
(.clk_out(clk_out),.reset(reset),.remove(remove),.sync_datain_ptr(sync_da
```

```
tain_ptr),.empty(empty),.read_en(read_en),.dataout_addr(dataout_addr),.da
taout_ptr(dataout_ptr));

sync S1
(.clk_in(clk_in),.clk_out(clk_out),.reset(reset),.dataout_ptr(dataout_ptr
),.datain_ptr(datain_ptr),.sync_dataout_ptr(sync_dataout_ptr),.sync_datai
n_ptr(sync_datain_ptr));

endmodule
```

Test Fixture

1) Test Fixture:-

```
`include "fifotop.v"
module fifotop_fixture2;
parameter word=32,depth=128;
reg clk_in,clk_out,flush,reset,insert,remove;reg [word-1:0] data_in;wire
full, empty;wire [word-1:0] data_out;
integer i;
fifotop F1
(.clk_in(clk_in),.clk_out(clk_out),.flush(flush),.reset(reset),.insert(in
sert),.remove(remove),.data_in(data_in),.data_out(data_out),.full(full),.
empty(empty));
//initial //$vcdpluson;
/////////////// clock in & clock out /////////////////initialbegin
      clk_in=1'b0;     forever #10 clk_in=~clk_in;end
initialbegin     clk_out=1'b0;    forever #5
clk_out=~clk_out;end/////////// reset task ////////////////////////////
task resettest();begin insert=1'b0;     remove=1'b0;     reset=1'b0;
      flush =1'b0;     #5 reset=1'b1;endendtask
task flushtest();begin insert=1'b0;     remove=1'b0;     reset=1'b1;
      flush =1'b1;     #5 flush=1'b0;endendtask


initialresettest();
/////////////// check for full/////////////////initialbegin       for
(i=0; i<128;i=i+1)     begin      @(posedge clk_in);
      data_in=$random;       insert=1;        #10;  endend
initialbegin           for (i=0; i<128;i=i+1)      begin      @(posedge
clk_out);        remove=1;        #10;  endend
/////////// check for empty /////////////////initialbegin     insert=0;
      remove=1;   #200;end/////////// check flush
/////////////////initialbegin    insert=0;   remove=0;   flush=1;    #5
flush=0;    #10;end/////////// testing random case
////////initialbegin
for (i=0; i<100;i=i+1)       begin      @(posedge clk_in);
      data_in=$random;       insert =$random;       #2;         insert =0;
            #2;
```

```
      @(posedge clk_out);    remove =$random; #4;   remove=0;   end
      resettest();     end

initialbegin           for (i=0; i<10;i=i+1)  begin       @(posedge
clk_in);          data_in=$random;       insert=1;        #10;
      endflushtest();#500 $finish;end

////////////////////// monitor
//////////////////////initialbegin$monitor($time," clk_in=%b | clk_out=%b |
reset=%b | flush=%b | insert=%b | remove=%b | datain=%h | dataout=%h |
empty=%b |
full=%b",clk_in,clk_out,reset,flush,insert,remove,data_in,data_out,empty,
full);
end
endmodule
```

Simulation Result

```
Chronologic VCS simulator copyright 1991-2014
Contains Synopsys proprietary information.
Compiler version I-2014.03-2; Runtime version I-2014.03-2;  May  1
21:25 2019
                   0 clk_in=0 | clk_out=0 | reset=0 | flush=1 |
insert=0 | remove=0 | datain=xxxxxxxx | dataout=00000000 | empty=x |
full=0
                   5 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=xxxxxxxx | dataout=00000000 | empty=x |
full=0
                  10 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=b1f05663 | dataout=00000000 | empty=x |
full=0
                  12 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=b1f05663 | dataout=00000000 | empty=x |
full=0
                  15 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=b1f05663 | dataout=00000000 | empty=x |
full=0
                  19 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=b1f05663 | dataout=00000000 | empty=x |
full=0
                  20 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=b1f05663 | dataout=00000000 | empty=x |
full=0
```

```
                  25 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=b1f05663 | dataout=00000000 | empty=x |
full=0
                  30 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=00f3e301 | dataout=00000000 | empty=x |
full=0
                  32 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=00f3e301 | dataout=00000000 | empty=x |
full=0
                  35 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=00f3e301 | dataout=00000000 | empty=x |
full=0
                  39 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=00f3e301 | dataout=00000000 | empty=x |
full=0
                  40 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=00f3e301 | dataout=00000000 | empty=x |
full=0
                  45 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=00f3e301 | dataout=00000000 | empty=x |
full=0
                  50 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=462df78c | dataout=00000000 | empty=x |
full=0
                  52 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=462df78c | dataout=00000000 | empty=x |
full=0
                  55 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=462df78c | dataout=00000000 | empty=x |
full=0
                  59 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=462df78c | dataout=00000000 | empty=x |
full=0
                  60 clk_in=0 | clk_out=0 | reset=1 | flush=1 |
insert=0 | remove=0 | datain=462df78c | dataout=00000000 | empty=x |
full=0
                  65 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=462df78c | dataout=00000000 | empty=x |
full=0
                  70 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e2f784c5 | dataout=00000000 | empty=x |
full=0
                  75 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e2f784c5 | dataout=00000000 | empty=x |
full=0
                  79 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=e2f784c5 | dataout=00000000 | empty=x |
full=0
                  80 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=e2f784c5 | dataout=00000000 | empty=x |
full=0
```

```
                  85 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e2f784c5 | dataout=00000000 | empty=x |
full=0
                  90 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=8932d612 | dataout=00000000 | empty=x |
full=0
                  92 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=8932d612 | dataout=00000000 | empty=x |
full=0
                  95 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=8932d612 | dataout=00000000 | empty=x |
full=0
                 100 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=8932d612 | dataout=00000000 | empty=x |
full=0
                 105 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=8932d612 | dataout=00000000 | empty=x |
full=0
                 110 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=f4007ae8 | dataout=00000000 | empty=x |
full=0
                 112 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=f4007ae8 | dataout=00000000 | empty=x |
full=0
                 115 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=f4007ae8 | dataout=00000000 | empty=x |
full=0
                 120 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=f4007ae8 | dataout=00000000 | empty=x |
full=0
                 125 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=f4007ae8 | dataout=00000000 | empty=x |
full=0
                 130 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=96ab582d | dataout=00000000 | empty=x |
full=0
                 132 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=96ab582d | dataout=00000000 | empty=x |
full=0
                 135 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=96ab582d | dataout=00000000 | empty=x |
full=0
                 139 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=96ab582d | dataout=00000000 | empty=x |
full=0
                 140 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=96ab582d | dataout=00000000 | empty=x |
full=0
                 145 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=96ab582d | dataout=00000000 | empty=x |
full=0
```

```
                150 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=c03b2280 | dataout=00000000 | empty=x |
full=0
                155 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=c03b2280 | dataout=00000000 | empty=x |
full=0
                160 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=c03b2280 | dataout=00000000 | empty=x |
full=0
                165 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=c03b2280 | dataout=00000000 | empty=x |
full=0
                170 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=cb203e96 | dataout=00000000 | empty=x |
full=0
                172 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=cb203e96 | dataout=00000000 | empty=x |
full=0
                175 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=cb203e96 | dataout=00000000 | empty=x |
full=0
                179 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=cb203e96 | dataout=00000000 | empty=x |
full=0
                180 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=cb203e96 | dataout=00000000 | empty=x |
full=0
                185 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=cb203e96 | dataout=00000000 | empty=x |
full=0
                190 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=359fdd6b | dataout=00000000 | empty=x |
full=0
                192 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=359fdd6b | dataout=00000000 | empty=x |
full=0
                195 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=359fdd6b | dataout=00000000 | empty=x |
full=0
                200 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=359fdd6b | dataout=00000000 | empty=x |
full=0
                205 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=359fdd6b | dataout=00000000 | empty=x |
full=0
                210 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=0effe91d | dataout=00000000 | empty=x |
full=0
                212 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=0effe91d | dataout=00000000 | empty=x |
full=0
```

```
                215 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=0effe91d | dataout=00000000 | empty=x |
full=0
                219 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=0effe91d | dataout=00000000 | empty=x |
full=0
                220 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=0effe91d | dataout=00000000 | empty=x |
full=0
                225 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=0effe91d | dataout=00000000 | empty=x |
full=0
                230 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e5730aca | dataout=00000000 | empty=x |
full=0
                235 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=e5730aca | dataout=00000000 | empty=x |
full=0
                240 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=e5730aca | dataout=00000000 | empty=x |
full=0
                245 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e5730aca | dataout=00000000 | empty=x |
full=0
                250 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=20c4b341 | dataout=00000000 | empty=x |
full=0
                255 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=20c4b341 | dataout=00000000 | empty=x |
full=0
                260 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=20c4b341 | dataout=00000000 | empty=x |
full=0
                265 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=20c4b341 | dataout=00000000 | empty=x |
full=0
                270 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=75c50deb | dataout=00000000 | empty=x |
full=0
                275 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=75c50deb | dataout=00000000 | empty=x |
full=0
                280 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=75c50deb | dataout=00000000 | empty=x |
full=0
                285 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=75c50deb | dataout=00000000 | empty=x |
full=0
                290 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=de7502bc | dataout=00000000 | empty=x |
full=0
```

```
                295 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=de7502bc | dataout=00000000 | empty=x |
full=0
                299 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=de7502bc | dataout=00000000 | empty=x |
full=0
                300 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=de7502bc | dataout=00000000 | empty=x |
full=0
                305 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=de7502bc | dataout=00000000 | empty=x |
full=0
                310 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=42f24185 | dataout=00000000 | empty=x |
full=0
                312 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=42f24185 | dataout=00000000 | empty=x |
full=0
                315 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=42f24185 | dataout=00000000 | empty=x |
full=0
                319 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=42f24185 | dataout=00000000 | empty=x |
full=0
                320 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=42f24185 | dataout=00000000 | empty=x |
full=0
                325 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=42f24185 | dataout=00000000 | empty=x |
full=0
                330 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=bf23327e | dataout=00000000 | empty=x |
full=0
                332 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=bf23327e | dataout=00000000 | empty=x |
full=0
                335 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=bf23327e | dataout=00000000 | empty=x |
full=0
                339 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=bf23327e | dataout=00000000 | empty=x |
full=0
                340 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=bf23327e | dataout=00000000 | empty=x |
full=0
                345 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=bf23327e | dataout=00000000 | empty=x |
full=0
                350 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=31230762 | dataout=00000000 | empty=x |
full=0
```

```
                355 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=31230762 | dataout=00000000 | empty=x |
full=0
                359 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=31230762 | dataout=00000000 | empty=x |
full=0
                360 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=31230762 | dataout=00000000 | empty=x |
full=0
                365 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=31230762 | dataout=00000000 | empty=x |
full=0
                370 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=7c6da9f8 | dataout=00000000 | empty=x |
full=0
                372 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=7c6da9f8 | dataout=00000000 | empty=x |
full=0
                375 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=7c6da9f8 | dataout=00000000 | empty=x |
full=0
                379 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=7c6da9f8 | dataout=00000000 | empty=x |
full=0
                380 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=7c6da9f8 | dataout=00000000 | empty=x |
full=0
                385 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=7c6da9f8 | dataout=00000000 | empty=x |
full=0
                390 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=adcbc05b | dataout=00000000 | empty=x |
full=0
                392 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=adcbc05b | dataout=00000000 | empty=x |
full=0
                395 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=adcbc05b | dataout=00000000 | empty=x |
full=0
                399 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=adcbc05b | dataout=00000000 | empty=x |
full=0
                400 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=adcbc05b | dataout=00000000 | empty=x |
full=0
                405 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=adcbc05b | dataout=00000000 | empty=x |
full=0
                410 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=ebfec0d7 | dataout=00000000 | empty=x |
full=0
```

```
                412 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=ebfec0d7 | dataout=00000000 | empty=x |
full=0
                415 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=ebfec0d7 | dataout=00000000 | empty=x |
full=0
                420 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=ebfec0d7 | dataout=00000000 | empty=x |
full=0
                425 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=ebfec0d7 | dataout=00000000 | empty=x |
full=0
                430 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e12ccec2 | dataout=00000000 | empty=x |
full=0
                435 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e12ccec2 | dataout=00000000 | empty=x |
full=0
                439 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=e12ccec2 | dataout=00000000 | empty=x |
full=0
                440 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=e12ccec2 | dataout=00000000 | empty=x |
full=0
                445 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=e12ccec2 | dataout=00000000 | empty=x |
full=0
                450 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=090cdb12 | dataout=00000000 | empty=x |
full=0
                455 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=090cdb12 | dataout=00000000 | empty=x |
full=0
                459 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=090cdb12 | dataout=00000000 | empty=x |
full=0
                460 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=090cdb12 | dataout=00000000 | empty=x |
full=0
                465 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=090cdb12 | dataout=00000000 | empty=x |
full=0
                470 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=0fd28f1f | dataout=00000000 | empty=x |
full=0
                472 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=0fd28f1f | dataout=00000000 | empty=x |
full=0
                475 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=0fd28f1f | dataout=00000000 | empty=x |
full=0
```

```
                479 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=0fd28f1f | dataout=00000000 | empty=x |
full=0
                480 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=0fd28f1f | dataout=00000000 | empty=x |
full=0
                485 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=0fd28f1f | dataout=00000000 | empty=x |
full=0
                490 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                492 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                495 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                499 clk_in=1 | clk_out=1 | reset=0 | flush=0 |
insert=0 | remove=0 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                500 clk_in=0 | clk_out=0 | reset=0 | flush=0 |
insert=0 | remove=0 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                504 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=0 | remove=0 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                505 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=0 | remove=1 | datain=2dda595b | dataout=00000000 | empty=x |
full=0
                510 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=150caf2a | dataout=00000000 | empty=x |
full=0
                515 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=150caf2a | dataout=00000000 | empty=x |
full=0
                520 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=150caf2a | dataout=00000000 | empty=x |
full=0
                525 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=150caf2a | dataout=00000000 | empty=x |
full=0
                530 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=2c156358 | dataout=00000000 | empty=x |
full=0
                535 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=2c156358 | dataout=00000000 | empty=x |
full=0
                540 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=2c156358 | dataout=00000000 | empty=x |
full=0
```

```
                545 clk_in=0 | clk_out=1 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=2c156358 | dataout=00000000 | empty=x |
full=0
                550 clk_in=1 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=c33f3886 | dataout=00000000 | empty=x |
full=0
                555 clk_in=1 | clk_out=1 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=c33f3886 | dataout=00000000 | empty=x |
full=0
                560 clk_in=0 | clk_out=0 | reset=1 | flush=0 |
insert=1 | remove=1 | datain=c33f3886 | dataout=00000000 | empty=x |
full=0
$finish called from file "fifotop_fixture2.v", line 133.
$finish at simulation time                  565
          V C S   S i m u l a t i o n   R e p o r t
Time: 565
CPU Time:      0.330 seconds;       Data structure size:   0.0Mb
Wed May  1 21:25:02 2019
```

Synthesis

1) Script file:

```
#Read the design in
read_file -format verilog {"fifotop.v"}

#set the current design
set current_design fifotop

#Link the design
link

#create clockand constrain the design
create_clock "clk_in" -period 2     -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_output_delay -clock clk_in  -max -rise 0.4 "full"
set_output_delay -clock clk_in  -min -rise 0.2 "full"
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"

create_clock "clk_out" -period 4     -name  "clk_out"
set_input_delay -clock clk_out  -max -rise 0.4 "remove"
set_input_delay -clock clk_in  -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
set_operating_conditions -library "saed90nm_typ" "TYPICAL"
#Synthesize and generate report
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_power > report5
```

2) Synthesis reports:

1) Area

```
****************************************
Report : area
Design : fifotop
Version: I-2013.12-SP5-4
Date   : Wed May  1 21:35:51 2019
****************************************

Information: Updating design information... (UID-85)
Information: Timing loop detected. (OPT-150)
     R1/U3/IN1 R1/U3/QN R1/U7/INP R1/U7/ZN R1/U6/IN1 R1/U6/QN
R1/dataout_ptr_reg[2]/CLK R1/dataout_ptr_reg[2]/Q R1/U38/IN2 R1/U38/Q
R1/U50/IN1 R1/U50/QN R1/U8/IN1 R1/U8/QN
Information: Timing loop detected. (OPT-150)
     R1/U5/INP R1/U5/ZN R1/U3/IN2 R1/U3/QN R1/U7/INP R1/U7/ZN
R1/U6/IN1 R1/U6/QN R1/dataout_ptr_reg[2]/CLK R1/dataout_ptr_reg[2]/Q
R1/U38/IN2 R1/U38/Q R1/U50/IN1 R1/U50/QN
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[2]'
        to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[2]'
        to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[7]'
        to break a timing loop. (OPT-314)
```

```
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[7]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[0]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[0]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[1]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[1]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[3]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[3]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[4]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[4]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[5]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[5]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'Q' on cell
'R1/dataout_ptr_reg[6]'
         to break a timing loop. (OPT-314)
Warning: Disabling timing arc between pins 'CLK' and 'QN' on cell
'R1/dataout_ptr_reg[6]'
         to break a timing loop. (OPT-314)
Library(s) Used:

    saed90nm_typ (File:
/netdisk/tmp/saed/SAED90_EDK/SAED_EDK90nm/Digital_Standard_cell_Libra
ry/synopsys/models/saed90nm_typ.db)

Number of ports:                          72
Number of nets:                          121
Number of cells:                           4
Number of combinational cells:             0
Number of sequential cells:                0
Number of macros/black boxes:              0
Number of buf/inv:                         0
```

```
Number of references:                     4

Combinational area:               39504.385325
Buf/Inv area:                      4975.718506
Noncombinational area:            93100.034393
Macro/Black Box area:                 0.000000
Net Interconnect area:             8362.053509

Total cell area:                 132604.419719
Total area:                      140966.473228
1
```

2) Constraints

```
****************************************
Report : constraint
        -all_violators
Design : fifotop
Version: I-2013.12-SP5-4
Date   : Wed May  1 21:35:51 2019
****************************************


   max_area

                             Required        Actual
   Design                      Area           Area           Slack
   -----------------------------------------------------------------------
   fifotop                     0.00       140966.47      -140966.47

(VIOLATED)


1
```

3) Timing

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : fifotop
Version: I-2013.12-SP5-4
Date   : Wed May  1 21:35:52 2019
****************************************

Operating Conditions: TYPICAL   Library: saed90nm_typ
Wire Load Model Mode: enclosed

  Startpoint: flush (input port clocked by clk_in)
  Endpoint: W1/current_state_reg[0]
```

```
          (rising edge-triggered flip-flop clocked by clk_in)
Path Group: clk_in
Path Type: max

Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
fifotop            140000                saed90nm_typ
datain             8000                  saed90nm_typ

Point                                    Incr       Path
-----------------------------------------------------------
clock clk_in (rise edge)                 0.00       0.00
clock network delay (ideal)              0.00       0.00
input external delay                     0.40       0.40 r
flush (in)                               0.00       0.40 r
W1/flush (datain)                        0.00       0.40 r
W1/U24/QN (NOR2X0)                       0.72       1.12 f
W1/current_state_reg[0]/D (DFFARX1)      0.03       1.16 f
data arrival time                                   1.16

clock clk_in (rise edge)                 2.00       2.00
clock network delay (ideal)              0.00       2.00
W1/current_state_reg[0]/CLK (DFFARX1)
                                         0.00       2.00 r
library setup time                      -0.06       1.94
data required time                                  1.94
-----------------------------------------------------------
data required time                                  1.94
data arrival time                                  -1.16
-----------------------------------------------------------
slack (MET)                                         0.79


Startpoint: remove (input port clocked by clk_out)
Endpoint: R1/current_state_reg[1]
          (rising edge-triggered flip-flop clocked by clk_out)
Path Group: clk_out
Path Type: max

Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
fifotop            140000                saed90nm_typ
dataout            8000                  saed90nm_typ

Point                                    Incr       Path
-----------------------------------------------------------
clock clk_out (rise edge)                0.00       0.00
clock network delay (ideal)              0.00       0.00
input external delay                     0.40       0.40 r
remove (in)                              0.00       0.40 r
R1/remove (dataout)                      0.00       0.40 r
R1/U9/QN (NOR2X0)                        0.57       0.97 f
R1/current_state_reg[1]/D (DFFARX1)      0.03       1.01 f
data arrival time                                   1.01
```

```
  clock clk_out (rise edge)                4.00       4.00
  clock network delay (ideal)              0.00       4.00
  R1/current_state_reg[1]/CLK (DFFARX1)
                                           0.00       4.00 r
  library setup time                      -0.06       3.94
  data required time                                  3.94
  ----------------------------------------------------------
  data required time                                  3.94
  data arrival time                                  -1.01
  ----------------------------------------------------------
  slack (MET)                                         2.93


1
```

4) Power

```
Loading db file
'/netdisk/tmp/saed/SAED90_EDK/SAED_EDK90nm/Digital_Standard_cell_Libr
ary/synopsys/models/saed90nm_typ.db'
Information: Propagating switching activity (low effort zero delay
simulation). (PWR-6)
Warning: Design has unannotated primary inputs. (PWR-414)
Warning: Design has unannotated sequential cell outputs. (PWR-415)

****************************************
Report : power
        -analysis_effort low
Design : fifotop
Version: I-2013.12-SP5-4
Date   : Wed May  1 21:35:53 2019
****************************************


Library(s) Used:

    saed90nm_typ (File:
/netdisk/tmp/saed/SAED90_EDK/SAED_EDK90nm/Digital_Standard_cell_Libra
ry/synopsys/models/saed90nm_typ.db)


Operating Conditions: TYPICAL   Library: saed90nm_typ
Wire Load Model Mode: enclosed

Design        Wire Load Model            Library
------------------------------------------------
fifotop                140000            saed90nm_typ
fifomem                140000            saed90nm_typ
datain                 8000              saed90nm_typ
dataout                8000              saed90nm_typ
sync                   8000              saed90nm_typ
```

```
dataout_DW01_inc_0     ForQA             saed90nm_typ
datain_DW01_inc_0      ForQA             saed90nm_typ


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  = 181.9435 uW   (40%)
  Net Switching Power  = 268.3800 uW   (60%)
                         ---------
Total Dynamic Power    = 450.3235 uW  (100%)

Cell Leakage Power     = 543.0214 uW


                 Internal         Switching           Leakage
Total
Power Group      Power            Power               Power
Power   (   %    )  Attrs
--------------------------------------------------------------------
----------------------------
io_pad             0.0000           0.0000            0.0000
0.0000  (   0.00%)
memory             0.0000           0.0000            0.0000
0.0000  (   0.00%)
black_box          0.0000           0.0000            0.0000
0.0000  (   0.00%)
clock_network      0.0000           0.0000            0.0000
0.0000  (   0.00%)
register           0.6644           0.7860        4.2690e+06
5.7194  (   0.58%)
sequential     -3.6947e-01           0.4621        4.1726e+08
417.3539 (  42.01%)
combinational    181.6497          267.1319        1.2149e+08
570.2721  (  57.41%)
--------------------------------------------------------------------
----------------------------
Total            181.9446 uW       268.3799 uW     5.4302e+08 pW
993.3454 uW
1
```

Code Coverage Report

Design Module List

dashboard | **hierarchy** | **modlist** | groups | **tests** | asserts

Total Module Definition Coverage Summary

SCORE	LINE	COND	TOGGLE	FSM
23.86	67.06	0.00	28.37	0.00

Total modules in report: 6

SCORE	LINE	COND	TOGGLE	FSM	NAME
15.91	50.00	0.00	13.64	0.00	datain
16.44	37.21		12.12	0.00	dataout
31.25			31.25		fifotop
34.24	73.33	0.00	29.39		fifomem
52.94	100.00		5.88		sync
76.04	100.00		52.08		fifo_fixture

dashboard | **hierarchy** | **modlist** | groups | **tests** | asserts

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%