```
`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(ful
l1),.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

$vcdpluson;

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 condiotion 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,ful
l);
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
```