```
module fifo_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==rptr[ptr_wdth-1:0]-1 &&
wptr[ptr_wdth]!=rptr[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
      else
```

```
      begin
            if(!(wptr[ptr_wdth-1:0]-1==rptr[ptr_wdth-1:0]-1 &&
wptr[ptr_wdth]!=rptr[ptr_wdth]))
                  full<=0;
            else
            begin
                  #2 full<=full;
                  wptr<=wptr;
            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(rptr[ptr_wdth:0]-1==wptr[ptr_wdth:0]-1)
            begin
                  empty<=1'b1;
                  rptr<=rptr;
                  #4 data_out<=data_out;

            end
            else
            begin
                  empty<=1'b0;
                  #4 data_out<=f_mem[rptr[ptr_wdth-1:0]];
                  if(rptr==8'b11111111)
                        rptr<=8'h0;
                  else
                        rptr<=rptr+1;
            end
            end
            else
                  begin
                        empty<=empty;
                        #4 data_out<=data_out;
                        rptr<=rptr;
                  end
end
endmodule
```