```
`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
```