module fifo\_sync

#( parameter FIFO\_DEPTH = 8,

parameter DATA\_WIDTH = 32)

(input clk,

input rst\_n,

input cs,

input wr\_en,

input rd\_en,

input [DATA\_WIDTH-1:0] data\_in,

output reg [DATA\_WIDTH-1:0] data\_out,

output empty,

output full);

localparam FIFO\_DEPTH\_LOG = $clog2(FIFO\_DEPTH);

reg [DATA\_WIDTH-1:0] fifo [0:FIFO\_DEPTH-1];

reg [FIFO\_DEPTH\_LOG:0] write\_pointer;

reg [FIFO\_DEPTH\_LOG:0] read\_pointer;

always @(posedge clk or negedge rst\_n)

begin

if(!rst\_n)

write\_pointer <= 0;

else if (cs && wr\_en && !full) begin

fifo[write\_pointer[FIFO\_DEPTH\_LOG-1:0]] <= data\_in;

write\_pointer <= write\_pointer + 1'b1;

end

end

always @(posedge clk or negedge rst\_n)

begin

if(!rst\_n)

read\_pointer <= 0;

else if (cs && rd\_en && !empty) begin

data\_out <= fifo[read\_pointer[FIFO\_DEPTH\_LOG-1:0]];

read\_pointer <= read\_pointer + 1'b1;

end

end

assign empty = (read\_pointer == write\_pointer);

assign full = (read\_pointer == {~write\_pointer[FIFO\_DEPTH\_LOG], write\_pointer[FIFO\_DEPTH\_LOG-1:0]});

endmodule