`include "config.sv"

module dp\_ram\_wrap

#(

parameter ADDR\_WIDTH = 8

)(

// Clock and Reset

input logic clk,

input logic en\_a\_i,

input logic [ADDR\_WIDTH-1:0] addr\_a\_i,

input logic [31:0] wdata\_a\_i,

output logic [31:0] rdata\_a\_o,

input logic we\_a\_i,

input logic [3:0] be\_a\_i,

input logic en\_b\_i,

input logic [ADDR\_WIDTH-1:0] addr\_b\_i,

input logic [31:0] wdata\_b\_i,

output logic [31:0] rdata\_b\_o,

input logic we\_b\_i,

input logic [3:0] be\_b\_i

);

`ifdef PULP\_FPGA\_EMUL

xilinx\_mem\_32768x32\_dp

dp\_ram\_i

(

.clka ( clk ),

.rsta ( 1'b1 ),

.clkb ( clk ),

.rstb ( 1'b1 ),

.ena ( en\_a\_i ),

.addra ( addr\_a\_i ),

.dina ( wdata\_a\_i ),

.douta ( rdata\_a\_o ),

.wea ( be\_a\_i & {4{we\_a\_i}} ),

.enb ( en\_b\_i ),

.addrb ( addr\_b\_i ),

.dinb ( wdata\_b\_i ),

.doutb ( rdata\_b\_o ),

.web ( be\_b\_i & {4{we\_b\_i}} )

);

`else

dp\_ram

#(

.ADDR\_WIDTH ( ADDR\_WIDTH )

)

dp\_ram\_i

(

.clk ( clk ),

.en\_a\_i ( en\_a\_i ),

.addr\_a\_i ( addr\_a\_i ),

.wdata\_a\_i ( wdata\_a\_i ),

.rdata\_a\_o ( rdata\_a\_o ),

.we\_a\_i ( we\_a\_i ),

.be\_a\_i ( be\_a\_i ),

.en\_b\_i ( en\_b\_i ),

.addr\_b\_i ( addr\_b\_i ),

.wdata\_b\_i ( wdata\_b\_i ),

.rdata\_b\_o ( rdata\_b\_o ),

.we\_b\_i ( we\_b\_i ),

.be\_b\_i ( be\_b\_i )

);

`endif

endmodule