module clk\_rst\_gen

(

input logic clk\_i,

input logic rstn\_i,

input logic clk\_sel\_i,

input logic clk\_standalone\_i,

input logic testmode\_i,

input logic scan\_en\_i,

input logic scan\_i,

output logic scan\_o,

input logic fll\_req\_i,

input logic fll\_wrn\_i,

input logic [1:0] fll\_add\_i,

input logic [31:0] fll\_data\_i,

output logic fll\_ack\_o,

output logic [31:0] fll\_r\_data\_o,

output logic fll\_lock\_o,

output logic clk\_o,

output logic rstn\_o

);

logic clk\_fll\_int;

logic clk\_int;

cluster\_clock\_mux2

clk\_mux\_i

(

.clk\_sel\_i ( clk\_sel\_i ),

.clk0\_i ( clk\_i ),

.clk1\_i ( clk\_fll\_int ),

.clk\_o ( clk\_int )

);

//----------------------------------------------------------------------------//

// FLL

//----------------------------------------------------------------------------//

`ifdef ASIC

umcL65\_LL\_FLL

fll\_i

(

.FLLCLK ( clk\_fll\_int ),

.FLLOE ( 1'b1 ),

.REFCLK ( clk\_i ),

.LOCK ( fll\_lock\_o ),

.CFGREQ ( fll\_req\_i ),

.CFGACK ( fll\_ack\_o ),

.CFGAD ( fll\_add\_i ),

.CFGD ( fll\_data\_i ),

.CFGQ ( fll\_r\_data\_o ),

.CFGWEB ( fll\_wrn\_i ),

.RSTB ( rstn\_i ),

.PWDB ( clk\_sel\_i ),

.STAB ( clk\_standalone\_i ),

.TM ( testmode\_i ),

.TE ( scan\_en\_i ),

.TD ( scan\_i ),

.TQ ( scan\_o )

);

`else

assign fll\_ack\_o = fll\_req\_i;

assign fll\_r\_data\_o = 1'b0;

assign fll\_lock\_o = 1'b0;

assign scan\_o = 1'b0;

`endif

//----------------------------------------------------------------------------//

// Reset synchronizer

//----------------------------------------------------------------------------//

rstgen i\_rst\_gen\_soc

(

// PAD FRAME SIGNALS

.clk\_i ( clk\_int ),

.rst\_ni ( rstn\_i ),

// TEST MODE

.test\_mode\_i ( testmode\_i ),

// OUTPUT RESET

.rst\_no ( rstn\_o ),

.init\_no ( )

);

assign clk\_o = clk\_int;

endmodule