How can I use the "clock enable pin" instead of gated clocks in my HDL designs?

The use of gated clocks can introduce glitches, increased clock delay, clock skew, and other undesirable effects, unless you pay close attention to your design. The first two examples in this article (VHDL and Verilog) illustrate a design that uses a gated clock.

Following these examples are VHDL and Verilog designs that show how you can modify the gated clock design to use the clock enable pin of the CLB.

VHDL Example of a Gated Clock

```
-------------------------------------------------
-- GATE_CLOCK.VHD Version 1.1                   --
-- Illustrates clock buffer control             --
-- Better implementation is to use              --
-- clock enable rather than gated clock         --
-- May 1997                                     --
-------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
entity gate_clock is
port (IN1,IN2,DATA,CLK,LOAD: in STD_LOGIC;
OUT1: out STD_LOGIC);
end gate_clock;

architecture BEHAVIORAL of gate_clock is
signal GATECLK: STD_LOGIC;
begin
GATECLK <= (IN1 and IN2 and CLK);
GATE_PR: process (GATECLK,DATA,LOAD)
begin
if (GATECLK'event and GATECLK='1') then
if (LOAD='1') then
OUT1 <= DATA;
end if;
end if;
end process; --End GATE_PR
end BEHAVIORAL;
```

Verilog Example of a Gated Clock

```
//////////////////////////////////////////
// GATE_CLOCK.V Version 1.1             //
// Gated Clock Example                  //
// Better implementation to use clock   //
// enables than gating the clock        //
// May 1997                             //
//////////////////////////////////////////
module gate_clock(IN1, IN2, DATA, CLK, LOAD,
 OUT1) ;
input IN1 ;
input IN2 ;
input DATA ;
input CLK ;
input LOAD ;
output OUT1 ;
reg OUT1 ;
wire GATECLK ;
assign GATECLK = (IN1 & IN2 & CLK);
always @(posedge GATECLK)
begin

if (LOAD == 1'b1)
OUT1 = DATA;
end
endmodule
```

VHDL Example of a Clock Enable Pin

```
-------------------------------------------------
-- CLOCK_ENABLE.VHD                           --
-- Illustrates clock use of clock             --
-- enable rather than gated clock             --
-- May 1997                                   --
-------------------------------------------------
-- CLOCK_ENABLE.VHD
-- May 1997
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
entity clock_enable is
port (IN1,IN2,DATA,CLOCK,LOAD: in STD_LOGIC;
DOUT: out STD_LOGIC);
end clock_enable;
architecture BEHAV of clock_enable is
signal ENABLE: STD_LOGIC;
begin
ENABLE <= IN1 and IN2 and LOAD;
EN_PR: process (ENABLE,DATA,CLOCK)
begin
if (CLOCK'event and CLOCK='1') then
if (ENABLE='1') then
DOUT <= DATA;
end if;
end if;
end process; -- End EN_PR
end BEHAV;
```

Verilog Example of a Clock Enable Pin

```
/*
//////////////////////////////////////////
// CLOCK_ENABLE.V                       //
// Example of use of clock enables      //
// rather than gating the clock         //
// May 1997                             //
//////////////////////////////////////////
*/
module clock_enable (IN1, IN2, DATA, CLK,
 LOAD, DOUT);
input IN1, IN2, DATA;
input CLK, LOAD;
output DOUT;
wire ENABLE;
reg DOUT;
assign ENABLE = IN1 & IN2 & LOAD;
always @(posedge CLK)
begin
if (ENABLE)
DOUT <= DATA;
end
endmodule
```

◆

THE XILINX HDL ADVISOR

by Roberta Fulton, Technical Marketing Engineer, Alliance Series, roberta.fulton@xilinx.com