library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity divider2\_0 is

generic (N: integer := 8);

port(clk,clear: in std\_logic ;

quot : buffer std\_logic\_vector(N-1 downto 0) ;

remainder : buffer std\_logic\_vector(N\*2-1 downto 0);

divisor, dividend: in std\_logic\_vector(N-1 downto 0);

green\_led: out std\_logic\_vector(2 downto 0));

end divider2\_0 ;

architecture behavior of divider2\_0 is

type state\_type is(A,B,C,D,E,F,BB);

signal state: state\_type ;

signal time1: integer ;

signal tdivisor: std\_logic\_vector(N\*2-1 downto 0);

signal rem\_zero : std\_logic\_vector(N\*2-1 downto 0) ;

begin

now\_state\_logic: process(clk,state)

begin

if clear = '0' then

tdivisor <=(others=>'0');

quot <=(others=>'0');

remainder <=(others=>'0');

state <= A ;

time1 <= 0;

green\_led <= "111" ;

elsif (clk'event and clk = '1') then

case state is

when A =>

remainder(N\*2-1 DOWNTO N) <=(others=>'0');

remainder(N-1 DOWNTO 0) <=dividend;

quot <=(others=>'0');

tdivisor(N\*2-1 DOWNTO N) <=divisor;

tdivisor(N-1 DOWNTO 0) <=(others=>'0');

state <= B ;

green\_led <="000";

when B =>

rem\_zero <=(others =>'0');

remainder <= (remainder - tdivisor) ;

state <= BB ;

green\_led <= "001";

when BB =>

if remainder(2\*N-1) = '1' then

state <= D ;

elsif remainder(2\*N-1) = '0' then

state <= C ;

else

remainder <=(others=>'1');

quot <=(others =>'1') ;

end if ;

green\_led <= "010";

when C =>

genbits1: for i in N-1 downto 1 loop

quot(i) <= quot(i-1) ;

end loop;

quot(0)<= '1';

state <= E;

green\_led <= "011";

when D =>

remainder <= remainder + tdivisor ;

genbits2: for i in N-1 downto 1 loop

quot(i) <= quot(i-1) ;

end loop;

quot(0)<= '0';

state <= E;

green\_led <= "100";

when E =>

genbits3: for i in 0 to N\*2-2 loop

tdivisor(i) <= tdivisor(i+1) ;

end loop;

tdivisor(N\*2-1)<= '0' ;

if time1 < N then

time1 <= time1 +1 ;

state <= B ;

else

state <= F ;

end if ;

green\_led <= "101";

when F =>

state <= F ;

green\_led <= "110";

end case ;

end if ;

end process ;

end behavior ;