--------------------------------------------------------------------------------

--

-- FileName: debounce.vhd

-- Dependencies: none

-- Design Software: Quartus II 32-bit Version 11.1 Build 173 SJ Full Version

--

-- HDL CODE IS PROVIDED "AS IS." DIGI-KEY EXPRESSLY DISCLAIMS ANY

-- WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT

-- LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A

-- PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DIGI-KEY

-- BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL

-- DAMAGES, LOST PROFITS OR LOST DATA, HARM TO YOUR EQUIPMENT, COST OF

-- PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES, ANY CLAIMS

-- BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF),

-- ANY CLAIMS FOR INDEMNITY OR CONTRIBUTION, OR OTHER SIMILAR COSTS.

--

-- Version History

-- Version 1.0 3/26/2012 Scott Larson

-- Initial Public Release

--

--------------------------------------------------------------------------------

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_unsigned.all;

ENTITY debounce IS

GENERIC(

counter\_size : INTEGER := 19); --counter size (19 bits gives 10.5ms with 50MHz clock)

PORT(

clk : IN STD\_LOGIC; --input clock

button : IN STD\_LOGIC; --input signal to be debounced

result : OUT STD\_LOGIC); --debounced signal

END debounce;

ARCHITECTURE logic OF debounce IS

SIGNAL flipflops : STD\_LOGIC\_VECTOR(1 DOWNTO 0); --input flip flops

SIGNAL counter\_set : STD\_LOGIC; --sync reset to zero

SIGNAL counter\_out : STD\_LOGIC\_VECTOR(counter\_size DOWNTO 0) := (OTHERS => '0'); --counter output

BEGIN

counter\_set <= flipflops(0) xor flipflops(1); --determine when to start/reset counter

PROCESS(clk)

BEGIN

IF(clk'EVENT and clk = '1') THEN

flipflops(0) <= button;

flipflops(1) <= flipflops(0);

If(counter\_set = '1') THEN --reset counter because input is changing

counter\_out <= (OTHERS => '0');

ELSIF(counter\_out(counter\_size) = '0') THEN --stable input time is not yet met

counter\_out <= counter\_out + 1;

ELSE --stable input time is met

result <= flipflops(1);

END IF;

END IF;

END PROCESS;

END logic;