WESTERN MICHIGAN UNIVERSITY
ECE 5510 APPLICATION SPECIFIC INTEGRATED CIRCUIT DESIGN

Assignment #4

STOPWATCH USING NEXYS3

By:

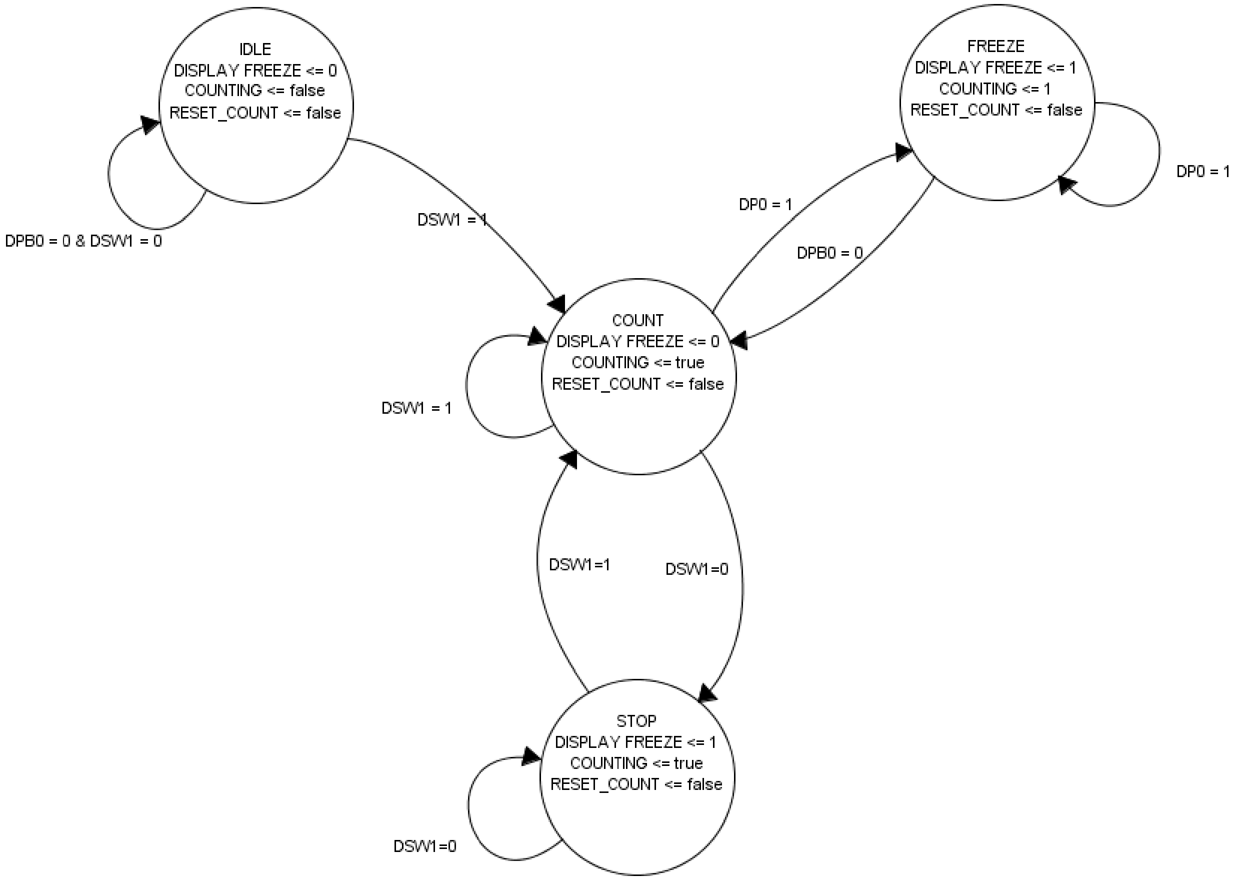
Erivelton Gualter dos Santos

1.Introduction

A stopwatch is an instrument to measure the amount of time. Generally, there are differences scales of range; however, I am using a range of 1 minute. The scale of the stopwatch is 0.01s. The first digit in the left shows, leftmost, shows hundredth of second, the second digit shows the tenth of a second, and the first and second; rightmost, shows the seconds.

Our stopwatch is controlled by a push buttons PB0 and a slid switches SW1. When SW1 is switched the watch resets and starts to count from 0. When SW1 is low-level, the watch stops and the final time is displayed. The function of the push button is to frozen the display, but the watch is not stoped; therefore, when PB0 is released, the display shows the current time.

2.FMS



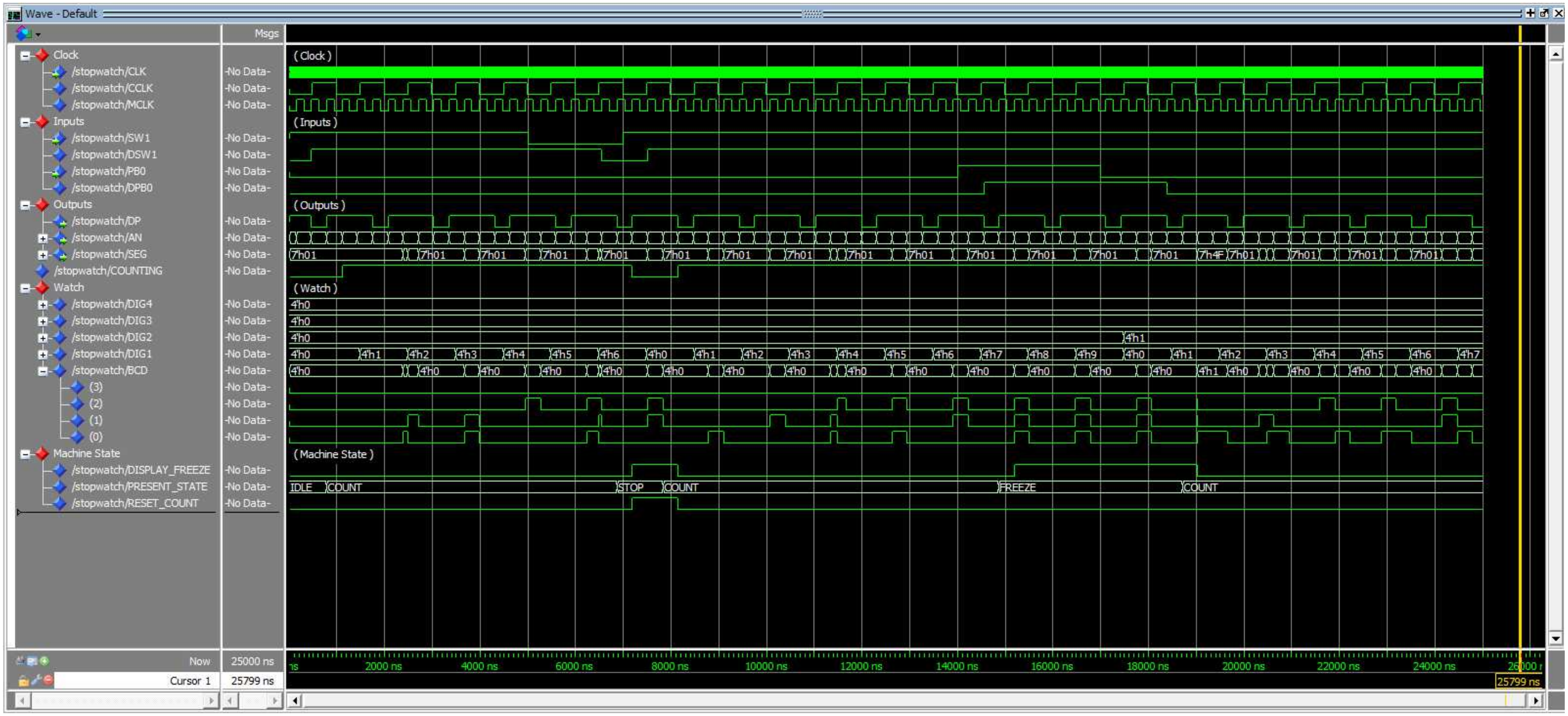
3. File.do

```
restart

force CLK 0 0, 1 10ns -r 20ns

force PB0 0
force SW1 1

force SW1 0 @5000ns
force SW1 1 @7000ns

force PB0 1 @14000ns
force PB0 0 @17000ns


run 25000ns
```

In this simulation, there is a different clock signal because is not viable to work with real clock. Also, it is testing the all conditions of the STOPWATCH.

Initially, the slide switch is turn ON (SW1) and the watch is initialized. You can see in the signals DIG4, DIG3, DIG2, DIG1 the corresponding values: 0000, 0001, 0002, 0003, 0004, 0005, and 0006. I can continue until 5999 because the next value is 0000. However, the SW1 is turn OFF for a short time, then turn ON. Note that the value was frozen and when SW1 was turned ON, the watch started to work from the 00.00.

Also, I simulated the push button PB0 that has the function to fix the value on the display when pushed. Therefore, in the moment that the PB0 is pressed the last value in the displayed is showed; in this case is 0007. However, the watch is working normally, and when it is released the display shows the current time; in this case is 00.10.


Wave - Default
Msgs
Clock
/stopwatch/CLK
/stopwatch/CCLK
/stopwatch/MCLK
Inputs
/stopwatch/SW1
/stopwatch/DSW1
/stopwatch/PB0
/stopwatch/DPB0
Outputs
/stopwatch/DP
/stopwatch/AN
/stopwatch/SEG
/stopwatch/COUNTING
Watch
/stopwatch/DIG4
/stopwatch/DIG3
/stopwatch/DIG2
/stopwatch/DIG1
/stopwatch/BCD
(3)
(2)
(1)
(0)
Machine State
/stopwatch/DISPLAY_FREEZE
/stopwatch/PRESENT_STATE
/stopwatch/RESET_COUNT
-No Data-
(Clock)
(Inputs)
(Outputs)
7'h01
7'h4F
(Watch)
4'h0
4'h1
4'h2
4'h3
4'h4
4'h5
4'h6
4'h7
4'h8
4'h9
(Machine State)
IDLE
COUNT
STOP
COUNT
FREEZE
COUNT
Now
25000 ns
Cursor 1
25799 ns
ns
2000 ns
4000 ns
6000 ns
8000 ns
10000 ns
12000 ns
14000 ns
16000 ns
18000 ns
20000 ns
22000 ns
24000 ns
26000 ns

```
----------------------------------------------------------------------------------
-- Company: Western Michigan University
-- Engineer: Erivelton Gualter dos Santos
-- 
-- Create Date:    21:50:25 06/03/2014 
-- Design Name: 
-- Module Name:    stopwatch - Behavioral 
-- Project Name: 
-- Target Devices: 
-- Tool versions: 
-- Description: 
--
-- Dependencies: 
--
-- Revision: 
-- Revision 0.01 - File Created
-- Additional Comments: 
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity stopwatch is
   port (   CLK : in std_logic;
            SW1 : in std_logic;
            PB0 : in std_logic;
            DP  : out std_logic;
            AN  : out std_logic_vector (3 downto 0);
            SEG : out std_logic_vector (6 downto 0));
end stopwatch;

architecture Behavioral of stopwatch is

   signal CCLK : std_logic;
   signal MCLK : std_logic;

   signal SHIFTR1,SHIFTR2 : std_logic_vector (3 downto 0) := "0000";
   signal DPB0 : std_logic := '0';
   signal DSW1 : std_logic := '0';

   type state is (IDLE, COUNT, STOP, FREEZE);
   signal PRESENT_STATE : state:=IDLE;

   signal DIG1: std_logic_vector (3 downto 0) := "0000";
   signal DIG2: std_logic_vector (3 downto 0) := "0000";
   signal DIG3: std_logic_vector (3 downto 0) := "0000";
   signal DIG4: std_logic_vector (3 downto 0) := "0000";

   signal BCD: std_logic_vector (3 downto 0);

   signal COUNTING   : boolean := false;
   signal RESET_COUNT: boolean := false;
   signal DISPLAY_FREEZE: std_logic := '0';

begin

--------------------------------------------------
--------         Count Clock       ---------------
--------------------------------------------------
clk_main: process(CLK, CCLK)
   variable DIVIDE : integer := 1000000;
   variable COUNTER_DIV : integer :=0;
begin
   if (rising_edge(CLK)) then
      if(COUNTER_DIV < DIVIDE/2-1) then
         COUNTER_DIV := COUNTER_DIV + 1;
         CCLK <= '0';
      elsif (COUNTER_DIV < DIVIDE-1) then
         COUNTER_DIV := COUNTER_DIV + 1;
         CCLK <= '1';
      else
         CCLK <= '0';
         COUNTER_DIV := 0;
      end if;
   end if;
end process clk_main;

--------------------------------------------------
--------         Main Clock        ---------------
--------------------------------------------------
clk_segment: process(CLK, MCLK)
   variable DIVIDE : integer := 1600;
   variable COUNTER_DIV : integer :=0;
begin
   if (rising_edge(CLK)) then
      if(COUNTER_DIV < DIVIDE/2-1) then
         COUNTER_DIV := COUNTER_DIV + 1;
         MCLK <= '0';
      elsif (COUNTER_DIV < DIVIDE-1) then
         COUNTER_DIV := COUNTER_DIV + 1;
         MCLK <= '1';
      else
         MCLK <= '0';
         COUNTER_DIV := 0;
      end if;
   end if;
end process clk_segment;

--------------------------------------------------
--------      Debouncing SW1           -----------
--------------------------------------------------

process(MCLK,SW1)
```

```vhdl
begin
   if (MCLK='1' and MCLK'event) then
      SHIFTR1(2 downto 0) <= SHIFTR1(3 downto 1);
      SHIFTR1(3) <= SW1;
      if (SHIFTR1 = "0000") then
         DSW1 <= '0';
      else
         DSW1 <= '1';
      end if;
   end if;
   -- simulaco
   -- DSW1 <= SW1;
end process;

--------------------------------------------------
--------      Debouncing PB0           -----------
--------------------------------------------------

process(MCLK,PB0)
begin
   if (MCLK='1' and MCLK'event) then
      SHIFTR2(2 downto 0) <= SHIFTR2(3 downto 1);
      SHIFTR2(3) <= PB0;
      if (SHIFTR2 = "0000") then
         DPB0 <= '0';
      else
         DPB0 <= '1';
      end if;
   end if;
   -- simulaco
   -- DPB0 <= PB0;
end process;

--------------------------------------------------
--------      Display Controller       -----------
--------------------------------------------------

process(MCLK, DISPLAY_FREEZE, DIG1, DIG2, DIG3, DIG4)
   variable SEL: STD_LOGIC_vector(1 downto 0) := "00";
   variable FDIG1: std_logic_vector (3 downto 0) ;
   variable FDIG2: std_logic_vector (3 downto 0) ;
   variable FDIG3: std_logic_vector (3 downto 0) ;
   variable FDIG4: std_logic_vector (3 downto 0) ;

begin
   if (MCLK='1' and MCLK'event ) then
      SEL:=SEL+1;
   end if;
   if (DISPLAY_FREEZE = '1' and DISPLAY_FREEZE'event) then
      FDIG1 := DIG1;
      FDIG2 := DIG2;
      FDIG3 := DIG3;
      FDIG4 := DIG4;
   end if;

   case SEL is
      when "00" =>
```

```
         if (DISPLAY_FREEZE = '1') then
            BCD <= FDIG1;
         else
            BCD <= DIG1;
         end if;
         AN  <= "1110";
         DP <= '1';
      when "01"=>
         if (DISPLAY_FREEZE = '1') then
            BCD <= FDIG2;
         else
            BCD <= DIG2;
         end if;
         AN  <= "1101";
         DP <= '1';
      when "10"=>
         if (DISPLAY_FREEZE = '1') then
            BCD <= FDIG3;
         else
            BCD <= DIG3;
         end if;
         AN  <= "1011";
         DP <= '0';
      when others=>
         if (DISPLAY_FREEZE = '1') then
            BCD <= FDIG4;
         else
            BCD <= DIG4;
         end if;
         AN  <= "0111";
         DP <= '1';
   end case;

end process;

--------------------------------------------------
--------     7 segments Decoder        -----------
--------------------------------------------------
segment_process: process(BCD)
begin
   case  BCD is
   when "0000"=> SEG <="0000001";  -- '0'
   when "0001"=> SEG <="1001111";  -- '1'
   when "0010"=> SEG <="0010010";  -- '2'
   when "0011"=> SEG <="0000110";  -- '3'
   when "0100"=> SEG <="1001100";  -- '4'
   when "0101"=> SEG <="0100100";  -- '5'
   when "0110"=> SEG <="0100000";  -- '6'
   when "0111"=> SEG <="0001111";  -- '7'
   when "1000"=> SEG <="0000000";  -- '8'
   when "1001"=> SEG <="0000100";  -- '9'
   when others=> SEG <="1111111";
   end case;
end process segment_process;

--------------------------------------------------
--------          Counter          ---------------
```

```
--------------------------------------------------
counter: process(CCLK, DSW1, RESET_COUNT)

begin
   if (CCLK'event and CCLK = '1') then
      if (COUNTING = true) then
         if (DIG4 = "0101" and DIG3 = "1001" and DIG2 = "1001" and DIG1 = "1001") then -- 59.99
            DIG4 <= "0000";
            DIG3 <= "0000";
            DIG2 <= "0000";
            DIG1 <= "0000";
         elsif (DIG3 = "1001" and DIG2 = "1001" and DIG1 = "1001") then -- X9.99
            DIG4 <= DIG4 + 1;
            DIG3 <= "0000";
            DIG2 <= "0000";
            DIG1 <= "0000";
         elsif (DIG2 = "1001" and DIG1 = "1001") then -- XX.99
            DIG3 <= DIG3 + 1;
            DIG2 <= "0000";
            DIG1 <= "0000";
         elsif (DIG1 = "1001") then -- XX.X9
            DIG2 <= DIG2 + 1;
            DIG1 <= "0000";
         else -- XX.XX
            DIG1 <= DIG1 + 1;
         end if;
      elsif (RESET_COUNT = true) then
         DIG4 <= "0000";
         DIG3 <= "0000";
         DIG2 <= "0000";
         DIG1 <= "0000";
      end if;
   end if;

end process counter;

--------------------------------------------------
--------         STATE MACHINE         -----------
--------------------------------------------------
machine: process(MCLK, DSW1, CCLK)
   --variable reset: integer :=0;
   --variable regreset: boolean := false;
begin
   if (MCLK'event and MCLK = '1') then
      case PRESENT_STATE is
      when IDLE =>
         DISPLAY_FREEZE <= '0';
         COUNTING       <= false;

         if (DPB0 = '1') then
            PRESENT_STATE <= FREEZE;
         elsif (DSW1 = '1') then
            PRESENT_STATE <= COUNT;
         else
            PRESENT_STATE <= IDLE;
         end if;

      when COUNT =>
         DISPLAY_FREEZE <= '0';
         COUNTING       <= true;
         RESET_COUNT    <= false;

         if(DSW1 = '1' and DPB0 = '1') then
            PRESENT_STATE <= FREEZE;
         elsif (DSW1 = '1') then
            PRESENT_STATE <= COUNT;
         elsif (DSW1 = '0') then
            PRESENT_STATE <= STOP;
         end if;

      when STOP =>
         COUNTING       <= false;
         RESET_COUNT    <= true;
         DISPLAY_FREEZE <= '1';

         if (DSW1 = '1') then
            PRESENT_STATE <= COUNT;
         else
            PRESENT_STATE <= STOP;
         end if;

      when FREEZE =>
         DISPLAY_FREEZE <= '1';
         COUNTING       <= true;

         if (DPB0 = '1') then
            PRESENT_STATE <= FREEZE;
         else
            PRESENT_STATE <= COUNT;
         end if;

      end case;
   end if;
end process machine;

end Behavioral;
```

```

# PlanAhead Generated IO constraints

NET "AN[3]" IOSTANDARD = LVCMOS33;
NET "AN[2]" IOSTANDARD = LVCMOS33;
NET "AN[1]" IOSTANDARD = LVCMOS33;
NET "AN[0]" IOSTANDARD = LVCMOS33;
NET "DP" IOSTANDARD = LVCMOS33;
NET "SEG[6]" IOSTANDARD = LVCMOS33;
NET "SEG[5]" IOSTANDARD = LVCMOS33;
NET "SEG[4]" IOSTANDARD = LVCMOS33;
NET "SEG[3]" IOSTANDARD = LVCMOS33;
NET "SEG[2]" IOSTANDARD = LVCMOS33;
NET "SEG[1]" IOSTANDARD = LVCMOS33;
NET "SEG[0]" IOSTANDARD = LVCMOS33;
NET "CLK" IOSTANDARD = LVCMOS33;
NET "PB0" IOSTANDARD = LVCMOS33;
NET "SW1" IOSTANDARD = LVCMOS33;

# PlanAhead Generated physical constraints

NET "AN[3]" LOC = P17;
NET "AN[2]" LOC = P18;
NET "AN[1]" LOC = N15;
NET "AN[0]" LOC = N16;
NET "DP" LOC = M13;
NET "SEG[6]" LOC = T17;
NET "SEG[5]" LOC = T18;
NET "SEG[4]" LOC = U17;
NET "SEG[3]" LOC = U18;
NET "SEG[2]" LOC = M14;
NET "SEG[1]" LOC = N14;
NET "SEG[0]" LOC = L14;
NET "CLK" LOC = V10;
NET "PB0" LOC = B8;
NET "SW1" LOC = T9;

```

IOB Properties - Tue Jun 10 22:09:17 2014

IOB Name	Type	Direction	IO Standard	Diff Term	Drive Strength	Slew Rate	Reg (s)	Resistor	IOB Delay
AN<0>	IOB	OUTPUT	LVCMOS33		12	SLOW			
AN<1>	IOB	OUTPUT	LVCMOS33		12	SLOW			
AN<2>	IOB	OUTPUT	LVCMOS33		12	SLOW			
AN<3>	IOB	OUTPUT	LVCMOS33		12	SLOW			
CLK	IOB	INPUT	LVCMOS33						
DP	IOB	OUTPUT	LVCMOS33		12	SLOW			
PB0	IOB	INPUT	LVCMOS33						
SEG<0>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SEG<1>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SEG<2>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SEG<3>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SEG<4>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SEG<5>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SEG<6>	IOB	OUTPUT	LVCMOS33		12	SLOW			
SW1	IOB	INPUT	LVCMOS33						

stopwatch Project Status (06/10/2014 - 22:07:28)			
Project File:	STOPWATCH.xise	**Parser Errors:**	No Errors
Module Name:	stopwatch	**Implementation State:**	Placed and Routed
Target Device:	xc6slx16-3csg324	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	19 Warnings (0 new)
Design Goal:	Balanced	• **Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	All Constraints Met
Environment:	System Settings	• **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	147	18,224	1%	
Number used as Flip Flops	131			
Number used as Latches	16			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	232	9,112	2%	
Number used as logic	230	9,112	2%	
Number using O6 output only	118			
Number using O5 output only	60			
Number using O5 and O6	52			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number used exclusively as route-thrus	2			
Number with same-slice register load	0			
Number with same-slice carry load	2			
Number with other load	0			
Number of occupied Slices	108	2,278	4%	
Number of MUXCYs used	96	4,556	2%	
Number of LUT Flip Flop pairs used	275			
Number with an unused Flip Flop	131	275	47%	
Number with an unused LUT	43	275	15%	
Number of fully used LUT-FF pairs	101	275	36%	
Number of unique control sets	52			
Number of slice register sites lost to control set restrictions	357	18,224	1%	
Number of bonded IOBs	15	232	6%	
Number of LOCed IOBs	15	15	100%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	2	16	12%	
Number used as BUFGs	2			
Number used as BUFGMUX	0			

Number of DCM/DCM_CLKGENs	0	4	0%	
Number of ILOGIC2/ISERDES2s	0	248	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	248	0%	
Number of OLOGIC2/OSERDES2s	0	248	0%	
Number of BSCANs	0	4	0%	
Number of BUFHs	0	128	0%	
Number of BUFPLLs	0	8	0%	
Number of BUFPLL_MCBs	0	4	0%	
Number of DSP48A1s	0	32	0%	
Number of ICAPs	0	1	0%	
Number of MCBs	0	2	0%	
Number of PCILOGICSEs	0	2	0%	
Number of PLL_ADVs	0	2	0%	
Number of PMVs	0	1	0%	
Number of STARTUPs	0	1	0%	
Number of SUSPEND_SYNCs	0	1	0%	
Average Fanout of Non-Clock Nets	2.64			

Performance Summary			[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Tue Jun 10 22:05:52 2014	0	3 Warnings (0 new)	3 Infos (0 new)
Translation Report	Current	Tue Jun 10 22:06:47 2014	0	0	0
Map Report	Current	Tue Jun 10 22:07:01 2014	0	16 Warnings (0 new)	6 Infos (0 new)
Place and Route Report	Current	Tue Jun 10 22:07:19 2014	0	0	3 Infos (0 new)
Power Report					
Post-PAR Static Timing Report	Current	Tue Jun 10 22:07:26 2014	0	0	4 Infos (0 new)
Bitgen Report	Out of Date	Tue Jun 10 21:55:11 2014	0	16 Warnings (16 new)	0

Secondary Reports		[-]
Report Name	**Status**	**Generated**
WebTalk Report	Out of Date	Tue Jun 10 21:55:12 2014
WebTalk Log File	Out of Date	Tue Jun 10 21:55:16 2014

Date Generated: 06/10/2014 - 22:07:28