8.1) Combinational system’s outputs are entirely defined by current inputs whereas sequential systems are dependent on previous input history in addition to the current inputs.

2) The history of inputs is represented by the system’s present state

3) Two stable states, set (1) or clear (0)

4) 10MHz and 25% duty cycle

6) Latches states update anytime the clock is high, flip-flops update only on the appropriate clock-edge. D-Latches and Flip-flops differ in the previously mentioned way and are similar in that they both only have one input, a clock, and one output.

8) In a process sensitive to a clock, you use an if statement with the clock value being ‘1’ for positive edge or ‘0’ for negative edge. Alternatively use rising\_edge for positive edge or falling\_edge for negative edge.

11)

Q \_\_\_-----\_\_\_----\_\_\_\_\_---\_\_\_\_

D \_\_\_----\_\_\_\_----\_\_\_\_----\_\_\_\_

Clk \_\_\_\_----\_\_\_\_\_\_\_\_----\_\_\_\_---

12) A latch is transparent if the input affects the output. A latch is transparent when the clock is in the state the latch is sensitive to. A positive d-latch would be transparent when the clock is high and a negative d-latch would be transparent when the clock is low.

16) if(clk = ‘0’) then

Q <= D;

Q’ <= not D;

End if;

22) if clk’event and clk = ‘0’ then; if falling\_edge(clk) then

23) Because clk’event evaluates to true for any clock edge, rather than only the edge the design uses.

24) A flip-flop can only change states once per cycle, on the sensitive clock edge. A latch’s state can change as many times as the input changes, as long as the latch is transparent.

25) Because a latch’s input can change the output of the latch even if the clock hasn’t changed, as long as it’s in the sensitive state (clock is high for positive latch, low for negative latch). A flip-flop’s state can only change on the clock-edge and is the only condition in which the output can be updated.

29) Because it suspends the process until clock has a high-to-low transition, or a negative edge.

32) No, but it can be used to describe a latch. With selected signal, the output would be updated anytime clk has the appropriate value (‘0’ for negative latch, ‘1’ for positive latch) instead of only on the clock edge.

41)

library ieee;

use ieee.std\_logic\_1164.all;

entity srff is

port( s, r, pre\_bar, clr\_bar, clk, t : in std\_logic;

q:out std\_logic );

end srff;

architecture behavioral of srff is

signal q\_sig : std\_logic ; -- local signal to assign and read

begin

process (clk, pre\_bar, clr\_bar)

begin

if pre\_bar = '0' then -- asserted low asynchronous preset

q\_sig <= '1';

elsif clr\_bar = '0' then -- asserted low asynchronous clear

q\_sig <= '0';

elsif rising\_edge(clk) and t = ‘1’ then

q\_sig<= not q\_sig; -- characteristic equation

end if;

end process;

q <= q\_sig; -- assign local signal to output

end behavioral

43) 74HC74: 3ns setup time

GAL22V10: Setup time is 2.5, 3, or 4ns depending on conditions; hold time is 0ns