

EE-313 LAB #6
ALTERA QUARTUS II DESIGN SOFTWARE

JK FLIP-FLOP USING VHDL

1. Purpose:

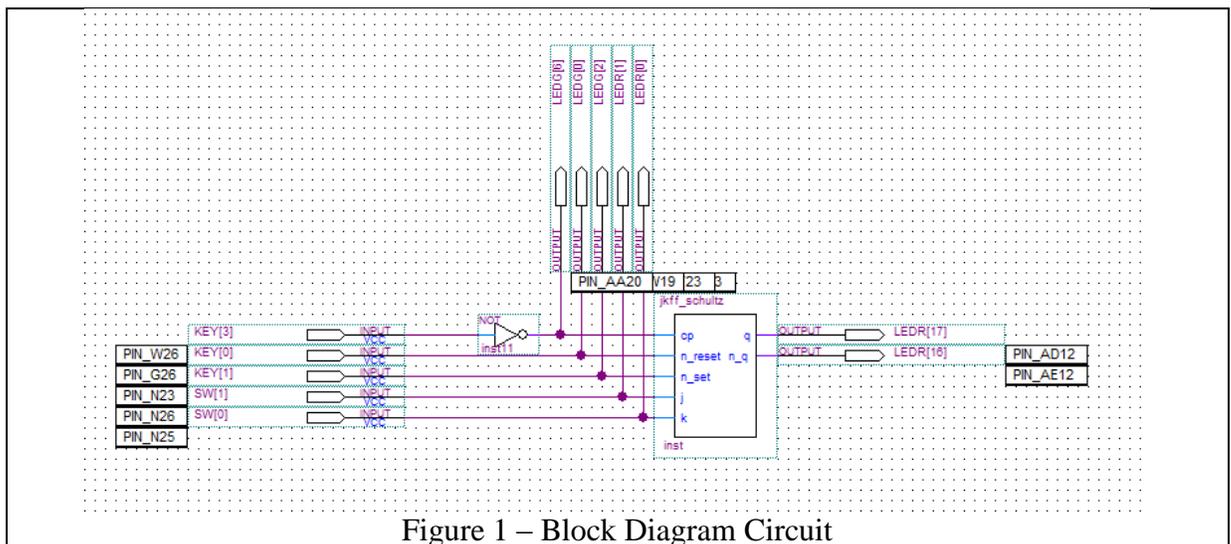
Introduce Asynchronous active low set and reset functions.
Understand how a Positive and Negative Edge Triggering is implemented using VHDL.
Review the operation of a JK Flip Flop.

2. Equipment: Desktop with Altera Software

Altera PLD Board
Altera Quartus II Quick Reference

3. Procedure

- a. Examine the two VHDL examples that demonstrate Asynchronous Active Low Set/Reset in Figure 2 and Negative Edge Triggering in Figure 3. Using VHDL design a JK Flip-Flop that
 - i. Has asynchronous active low set and reset and set functions
 - ii. Is triggered by the using the positive edge of a clock Cp for the “J” and “K” synchronous operations.
- b. Create a Symbol file for your flip flop.
- c. Create a Block Diagram Circuit using your symbol file as Shown in Figure 1. Ensure you use the same input and output pins.
- d. Program your altera board and demonstrate proper operation of the following features to your instructor.
 - i. Asynchronous input reset.
 - ii. Asynchronous input set.
 - iii. Synchronous hold, set, reset and toggle.



```

ex10_10.vhd
LIBRARY ieee;
USE ieee.std_logic_1164.ALL; -- D FF with Reset and Set--

ENTITY ex10_10 IS
    PORT(cp, n_reset, n_set, d : IN std_logic;
         q : OUT std_logic);
END ex10_10;

ARCHITECTURE arc OF ex10_10 IS
BEGIN
    PROCESS (cp, n_reset, n_set)
    BEGIN
        IF (n_reset = '0' AND n_set='1') THEN -- Asynchronous Reset
            q<='0';
        ELSIF (n_reset = '1' AND n_set='0') THEN -- Asynchronous Set
            q<='1';
        ELSIF (cp'EVENT AND cp= '1') THEN -- Synchronous operation
            q<=d; -- Positive clock edge
        END IF;
    END PROCESS;
END arc;

```

Figure 2 – D-FF with Asynchronous Set and Reset

```

ex10_14.vhd
LIBRARY ieee;
USE ieee.std_logic_1164.ALL; -- J-K FF --

ENTITY ex10_14 IS
    PORT(n_cp, j, k : IN std_logic;
         q : BUFFER std_logic);
END ex10_14;

ARCHITECTURE arc OF ex10_14 IS
    SIGNAL jk : std_logic_vector (1 DOWNTO 0);
BEGIN
    jk<=j&k; -- Concatenate j and k into a 2-bit vector
    PROCESS (n_cp, j, k)
    BEGIN
        IF (n_cp'EVENT AND n_cp= '0') THEN --Neg edge trigger
            CASE jk IS
                WHEN "00" => q <= q; --Hold
                WHEN "01" => q <= '0'; --Reset
                WHEN "10" => q <= '1'; --Set
                WHEN "11" => q <= NOT q; --Toggle
                WHEN OTHERS => q <= q;
            END CASE;
        END IF;
    END PROCESS;
END arc;

```

Figure 3 – JK-FF with Negative Edge Trigger

4. Deliverables Your lab report shall include:

- a. JK Flip Flop Source Code
- b. Block Diagram Schematic
- c. Lessons Learned