In The Name of God

COMPUTER ENGINEERING DEPARTMENT OF AMIRKABIR UNIVERSITY OF TECHNOLOGY

Design Automation Homework - 5

Iman Tabrizian (9331032)

May 12, 2017

1 QUESTION 6

(a) **Profiling** is the

(b) Implementation using software has the benefit of sequential execution and also has the benefit of reusing the existing hardware components, however hardware implementation has the benefit of concurrency and faster execution time but it consumes more resources. It is recommended that for the parts that take a lot of time, use the hardware implementation while for the parts that don't consume much time use the software implementation.

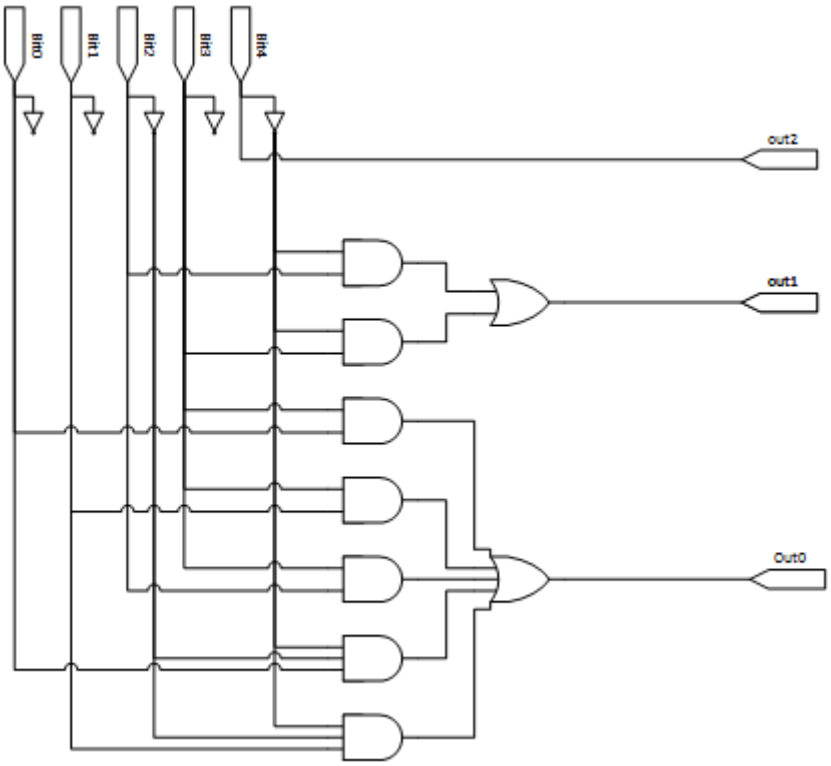
(c) **Hard Core** is an actual integrated circuit that does the job of central processing unit, while **Soft Core** is implemented in FPGA using the LUTs and there is no actual circuit.

If we implement a core by ourselves it has the overhead of not being optimized. Because both of the above cores are fully optimized. The hard core is optimized using electrical elements and the soft core is optimized by the vendor that provides it.

(d)

2 QUESTION 7

The circuit that I have found by my self is like below:



In the above circuit the MSB is *bit4* and in the output also the MSB is *out2*.

```
library ieee;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity square_root is
    port(
            input: in std_logic_vector(4 downto 0);
            output: out std_logic_vector(2 downto 0)
        );
end entity;

architecture rtl of square_root is
begin
    output(2) <= input(4);
    output(1) <= ((not input(4)) and input(2)) or ((not input(4)) and input(3));
    output(0) <= (input(3) and input(0)) or (input(3) and input(1)) or
                 ((input(3) and input(2))) or ((not input(4)) and
                 (not input(2)) and input(0)) or ((not input(4)) and
                 (not input(2)) and input(1));
```

```
end rtl;

architecture memory of square_root is

    type ram_type is array(31 downto 0) of std_logic_vector(2 downto 0);
    signal ram: ram_type := ("101", "101", "101", "101", "101", "101", "101",
                             "100", "100", "100", "100", "100", "100", "100",
                             "100", "100", "011", "011", "011", "011", "011",
                             "011", "011", "010", "010", "010", "010" , "010",
                             "001", "001", "001", "000");
begin

    output <= ram(to_integer(unsigned(input)));

end memory;
```