ELEC 457

ASSIGNMENT – 2

Sait Berk VARIMLI

141024054

It took 6 hour for prepare.

1- Briefly describe each of the following terms in your own words.

PROM: PROM(Programmable read-only memory) is a computer memory chip that can be programmed once after it is created. Once the PROM is programmed, the information written is permanent and can't be erased or deleted.

PAL: It is one time programmable and it contains programmable AND gates with fixed OR gates.

PLA: It is also one time programmable and it contains programmble AND gates with programmable OR gates.

CPLD: It is different from PAL,PLA etc. because they were called Simple Programmable Logic Devices but, CPLD contains logic or functional blocks with a macrosell using PAL or PLA circuit.

ASIC : "ASIC" means application-specific integrated circuit is a microchip designed for a special application, such as a particular kind of transmission protocol or a hand-held computer.

ASSP : An ASSP (application-specific standart product) is a semiconductor device integrated circuit product that is a dedicated to a specific application market and sold to more than one user. It generally offers same performance characteristics as ASICs.

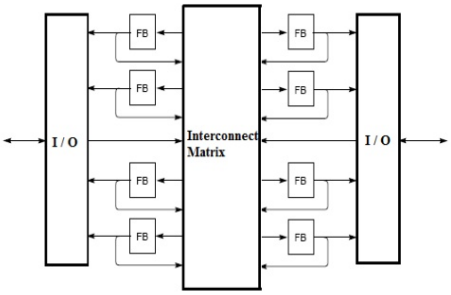
SoC : It means system-on-a-chip. It is a microchip with all the necessary electronic circuits and parts for a given system, such as a smartphone or wearable computer, on a single integrated circuit.

FPGA : Field Programmable Gate Arrays are semiconductor devices that are based around a matrix of configurable logic blocks connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionally requirements after manufacturing.

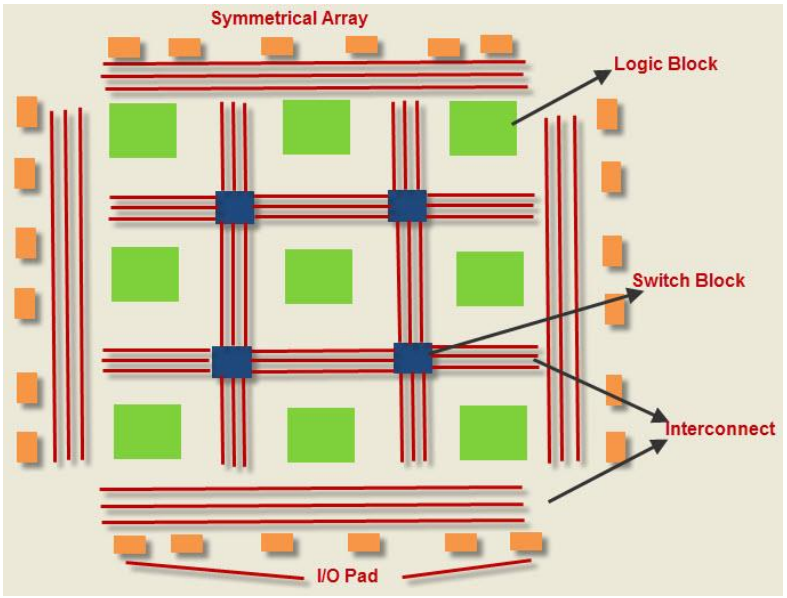
2-

- Explain the architecture of CPLDs and FPGAs. Support with a diagram

CPDL Architecture: A CPDL comprises of a group of programmable function blocks. The inputs and outputs of these functional blocks are connected together by a global interconnection matrix. This interconnection matrix is reconfigurable, so that we can modify the contacts between the functional blocks. Generally, the programmable function blocks looks like the array of logic gates, where an array of AND gates can be programmed and OR gates are stable. But, each manufacturer has their way of thinking to design the functional block. A listed o/p can be found by operating feedback signals attained from the OR gate outputs.



FPGA Architecture: General FPGA architecture consist of three types of modules. They are I/O blocks or Pads, Switch Matrix/Interconnection Wires and Configurable logic blocks. The basic FPGA architecture has two dimensional arrays of logic blocks with a means for a user to arrange the interconnection between the logic blocks. The functions of an FPGA architecture modules are: Configurable Logic Block, Interconnects provide direction between the logic blocks to implement the user logic. Depending on the logic, witch matrix provides switching between interconnects. I/O Pads used fort he outside World to communicate with different applications.



- List the differences between them.

-FPGA contains up to 100,000 of tiny logic blocks while CPLD contains only a few blocks of logic that reaches up to a few thousands.

-In terms of architecture, FPGAs are considered as “fine-grain” devices while CPLDs are “coarse-grain”.

-FPGAs are great for more complex applications while CPLDs are better for simpler ones.

-FPGAs are made up of tiny logic blocks while CPLDs are made of larger blocks.

-FPGA is a RAM-based digital logic chip while CPLD is EEPROM-based.

-Normally, FPGAs are more expensive while CPLDs are much cheaper.

-Delays are much more predictable in CPLDs than in FPGAs.

- Explain different FPGA architectures such as *SRAM*, *Flash*, or *anti-fuse* based FPGAs.

SRAM based FPGAs: This type of FPGA stores logic cells configuration data in the static memory. Since SRAM is volatile and can’t keep data without power source, such FPGAs must be programmed(configured) upon start. There are two basic modes of programming: Master mode and Slave mode. SRAM-based FPGAs include most chips of Xilinx Virtex and Spartan families and Altera Stratix and Cyclone.

Flash based FPGAs: Flash-based FPGA uses flash as a primary resource for configruation storage, and doesn’t require SRAM. This technology has an advantage of being less power consumptive. Flash-based FPGAs are also more tolerant to radiation effects. Flash-based FPGA faimilies such as Igloo and ProASIC3 are manufactured by Actel.

Antifuse based FPGA: The antifuse is a device that doesn’t conduct current initially, but can be “burned” to conduct current.(the antifuse behavior is thus opposite to that of the fuse, hence the name). The antifuse-based FPGA can’t be then reporgrammed since there is no way to return a burned antifuse into the initial state. Antifuse based device families include Axcelerator produced by Actel.

- List their differences and give a commercial FPGA example from vendors.

-Modern SRAM-based FPGAs have highest densities, but consume a lot of power and need an external non-volatile memory to store configuration bitstream.

-SRAM-based FPGAs with an internal flash modüle doesn’t need an external configuration memory.

-Flash-based and Antifuse-based FPGAs consume much less power than their SRAM-based counterparts.

-Antifuse-based FPGAs can only be programmed once.

3- Draw the given boolean expression for a PAL device. You can use the template given below.

```
O3 = (I3 and !I2) or ((I0 nand !I1) and (!I2 nand I3))
O2 = !((I3 xor I2) and (I0 xor I1))
O1 = (I0 and I1) or (!I2 and !I3)
O0 = O3 or O2
```

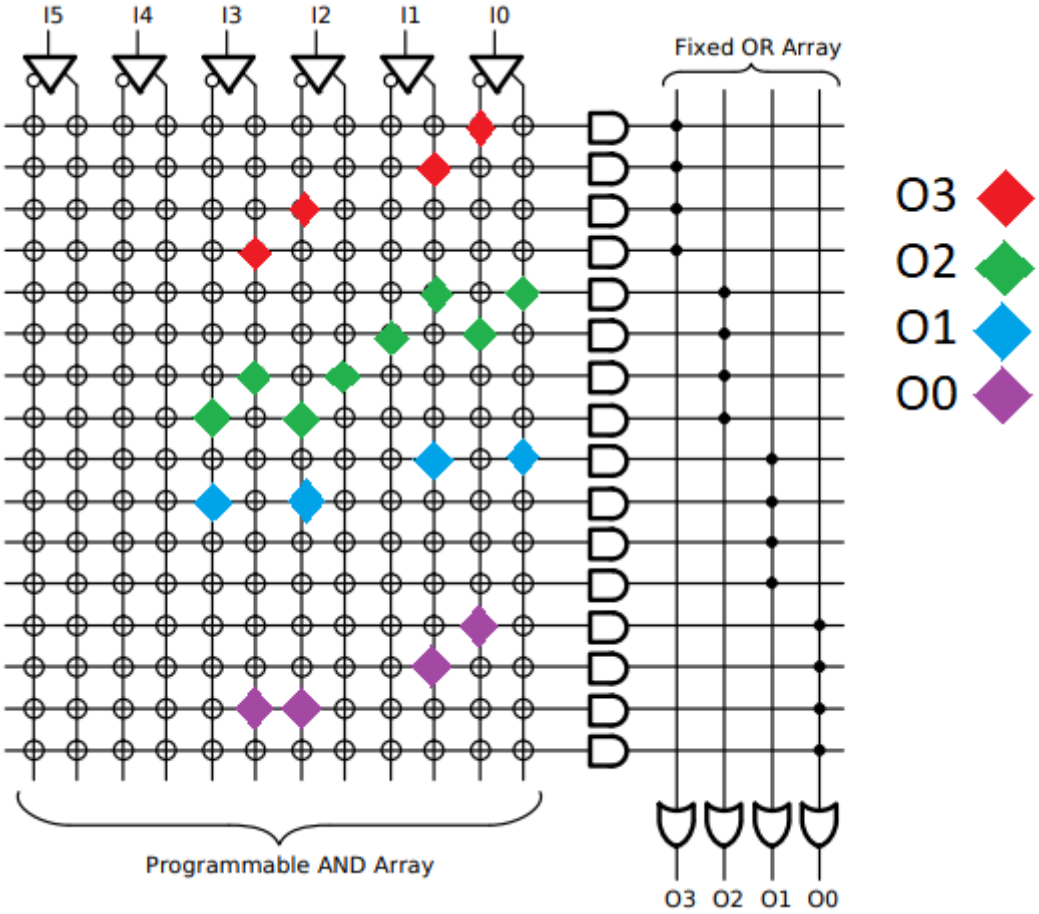
Firstly, i've simplified some expression for easy solution;

O3 = !I0 or I1 or (!I2 and I3)

O2 = (I0 and I1) or (!I0 and !I1) or (I2 and I3) or (!I2 and !I3)

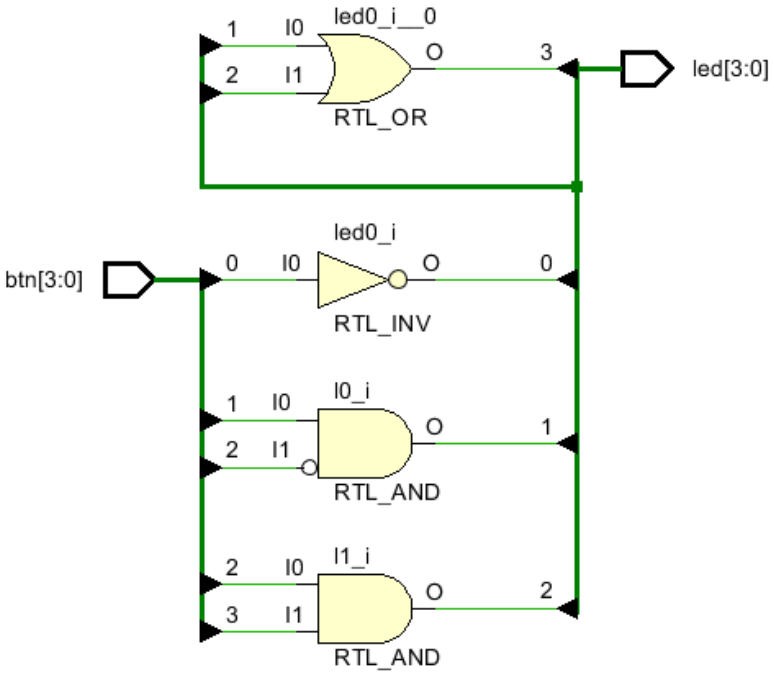
O0 = !I0 or I1 or !I2 or I3

Final device:



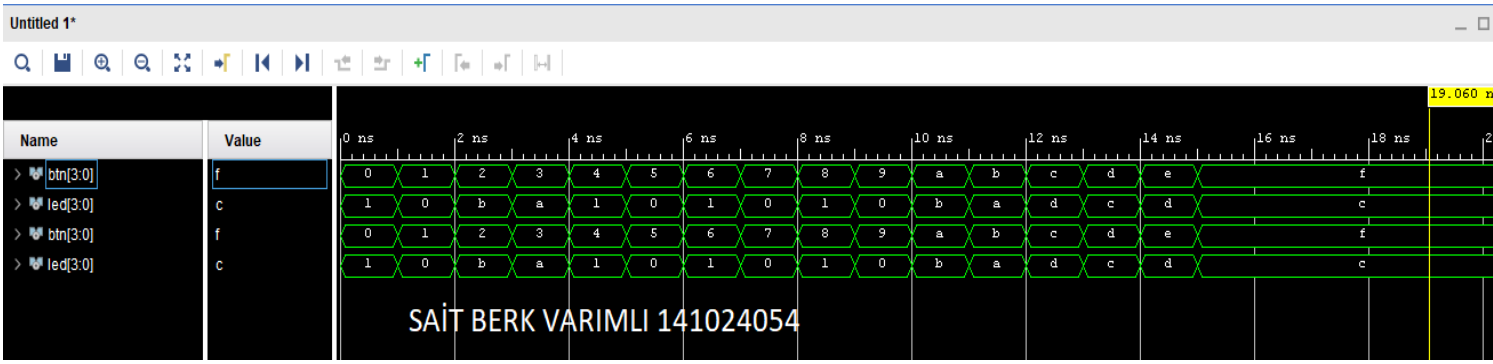
4- Download & Complete lab1. Use lab1.vhd and tb_lab1.vhd instead of the verilog source files. You do not need to program the board if you do not have one.

- Submit the schematic

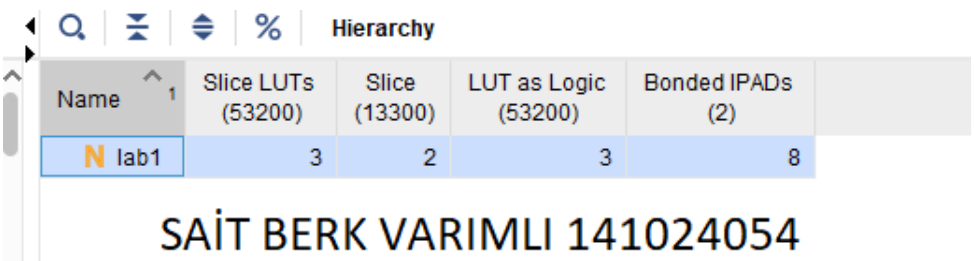


Sait Berk VARIMLI 141024054

- Submit the simulation screenshot



- Submit implementation reports on utilization

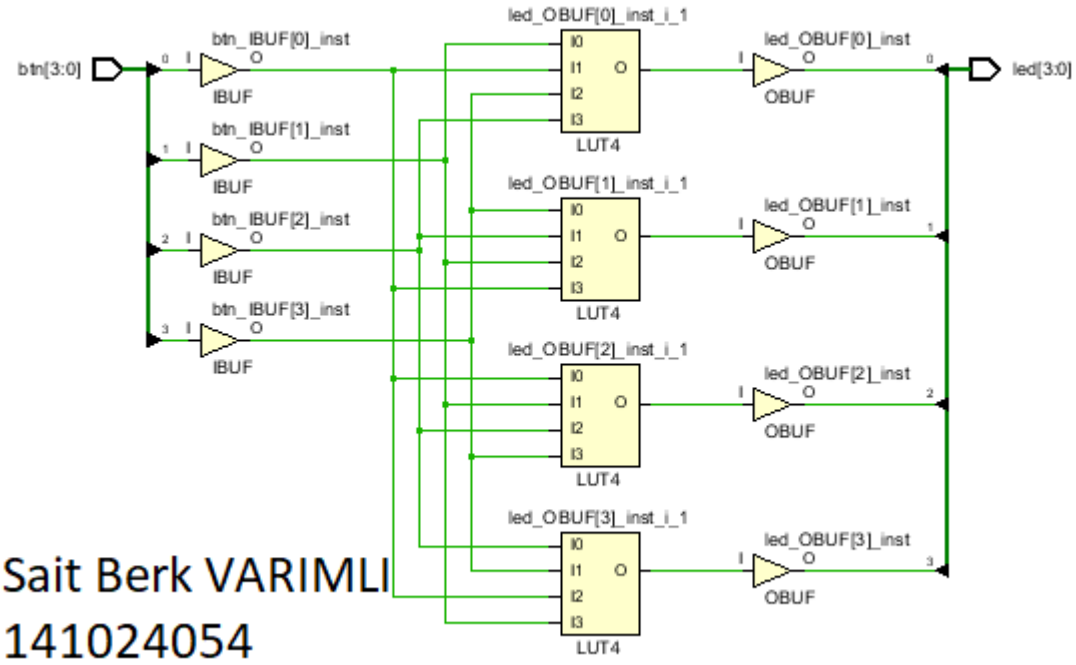


SAİT BERK VARIMLI 141024054

5- Create another project based on lab1. Change the `lab1.vhd` file to implement the boolean expressions given in Problem 3.

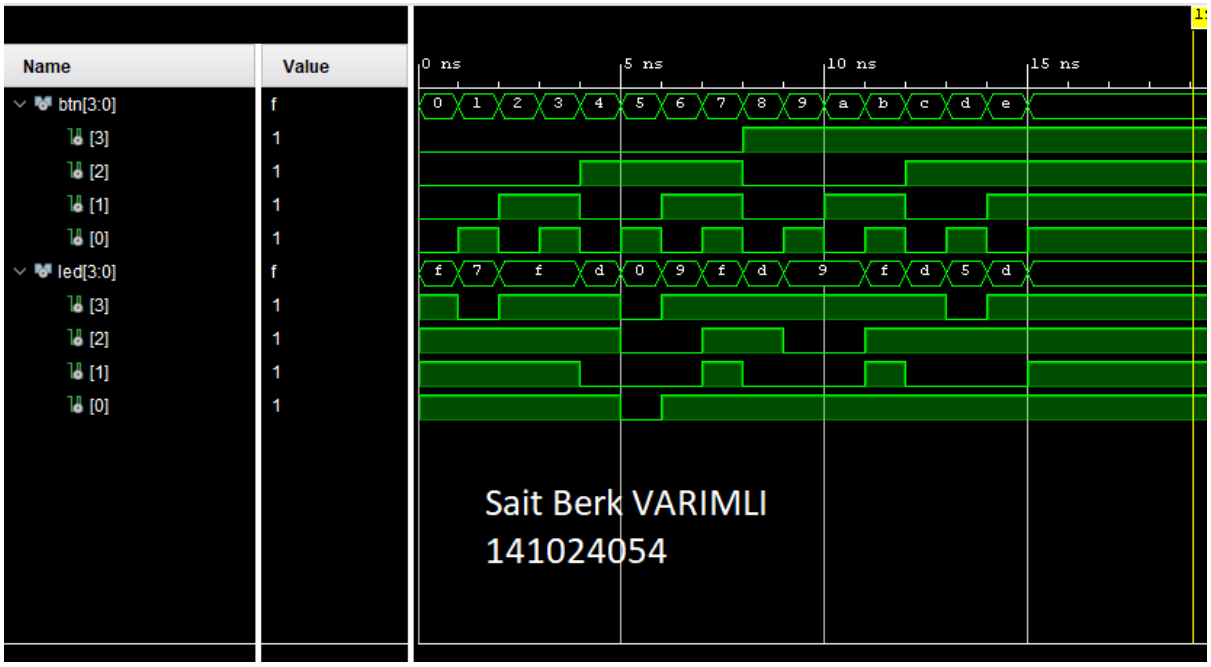
> Replace `Ox` with `led(x)` and `Ix` with `btn(x)` in the equations to match with lab1 circuit port names
>
> You cannot write `a nand b`, instead you should do `not (a and b)`. Do not forget to put parenthesis in proper places just like regular math.
>
> You cannot read from `led(x)` for the last equation. Instead use the intermediate signals defined in the file.

- **Submit the Schematic**



Sait Berk VARIMLI
141024054

- Submit the simulation screenshot



- Submit implementation reports on utilization

Name	Slice LUTs (53200)	Slice (13300)	LUT as Logic (53200)	Bonded IPADs (2)
lab1	2	1	2	8

Sait Berk VARIMLI 141024054

Problem 5 Code:

```
-- Assignment–2 Problem 5 for FPGA course

library ieee;

use ieee.std_logic_1164.all;


entity lab1 is

-- btns are inputs of the circuit

-- leds are outputs of the circuit

port (

    btn : in std_logic_vector(3 downto 0);

    led : out std_logic_vector(3 downto 0)

);

end lab1;


architecture rtl of lab1 is

    -- intermediate signals

    --signal l0, l1 : std_logic := '0';

    signal signal_led : std_logic_vector (1 downto 0);

begin



    -- problem 5


    signal_led(1) <= not ( btn(0)) or btn(1) or (not(btn(2)) and btn(3));

    signal_led(0) <= (btn(0) and btn(1)) or (not(btn(0)) and not(btn(1))) or
(btn(2) and btn(3)) or (not (btn(2)) and not(btn(3)));

    led(1) <= (btn(0) and btn(1)) or (not(btn(2)) and not(btn(3)));

    led(3) <= signal_led(1);

    led(2) <= signal_led(0);

    led(0) <= signal_led(1) or signal_led(0);




end rtl;
```