Multiplexer with FPGA and Icoboard

Saurabh Vankalas and Susheel Halwale

I. Introduction

Design and Code for FPGA and Icoboard based multiplexer 2x1. With the help of the code, we can easily extend the multiplexer, for ex:4x1, 32x1,etc.

II. Components used

1. Jumper Wires.
2. LED.
3. Raspberry Pi.
4. FPGA Icoboard.
5. Register.

III. Installation

First, we need to make following installations on our system.

```
cd $home
git clone git://git.drogon.net/wiringPi
cd wiringPi && ./build
cd $home
sudo apt-get install subversion
svn co http://svn.clifford.at/handicraft/2015/icoprog
cd icoprog && make install
sudo apt-get install build-essential clang bison flex
libreadline-dev
sudo apt-get install gawk tcl-dev libffi-dev git
mercurial graphviz
sudo apt-get install xdot pkg-config python
python3 libftdi-dev
cd $home
git clone
https://github.com/cliffordwolf/icestorm.git
icestorm
cd icestorm && make && sudo make install
cd $home
git clone https://github.com/cseed/arachne-pnr.git
arachne-pnr
cd arachne-pnr && make && sudo make install
cd $home
git clone https://github.com/clifford/yosys.git yosys
cd yosys && make && sudo make install
```

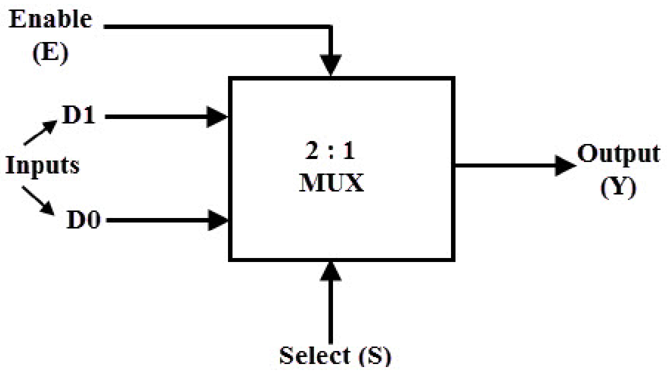

Fig. 1. 2 X 1 Multiplexer.

Now, we need to make a new file with name Makefile.

```
.PHONY: default
default: prog_sram
$(v_fname).blif: $(v_fname).v
    yosys -p ’synth_ice40 -blif $(v_fname).blif’
$(v_fname).v
$(v_fname).asc: $(v_fname).blif $(v_fname).pcf
    arachne-pnr -d 8k -p $(v_fname).pcf -o
$(v_fname).asc
    $(v_fname).blif
$(v_fname).bin: $(v_fname).asc
    icetime -d hx8k -c 25 $(v_fname).asc
    icepack $(v_fname).asc $(v_fname).bin
prog_sram: $(v_fname).bin
    icoprog -p ¡ $(v_fname).bin
```

IV. Hardware Connections

Figure 1 shows the multiplexer. Here, we have 2 inputs and one select input to determine the output. Basically, the select input decides the output would either be D0 or D1. We have to make the following connections with the icoboard.

Register : Vcc O : B5
A : B6
B : B7
sel : B4
Here, we have O as output and A and B as inputs. sel is the select input. These will be connected to

the pins B5, B6, B7, B4 respectively. The Register will be connected between Vcc of icoboard and LED.

V. Code

We create following file with name 2_1mux.pcf where we define the input output pins on the icoboard. Following is the code:

```
set_io O B5
set_io A B6
set_io B B7
set_io sel B4
```

Now, we create a file named 2_1mux.v where the actual logic for multiplexer will be written in verilog.

```
module mux2_1(O,A,B,sel);
// if sel = 0, O = A
// if sel = 1, O =B
output reg O;
input wire A;
input wire B;
input wire sel;
not #(50) not1(nsel,sel);
and #(50) and1(O1,A,nsel);
and #(50) and2(O2,B,sel);
or #(50) or2(O,O1,O2);
endmodule
```

VI. Run the project

We run the following command in the terminal to execute the code.

```
make v_fname=2_1mux
```

We observe from the output that the multiplexer is working. Also, we can easily extend the code for 4x1, 8x1, 32x1, etc.