Exp. 6 : Decoders and Multiplexers

Engineering 357 - Digital Design Lab

Fall 2015

Farnam Adelkhani

Date Experiment Performed: 10-12-2015

Date Report Submitted: 10-15-2015

Farnam Adelkhani

Engineering 357 - Digital Design Lab – 10/1/15

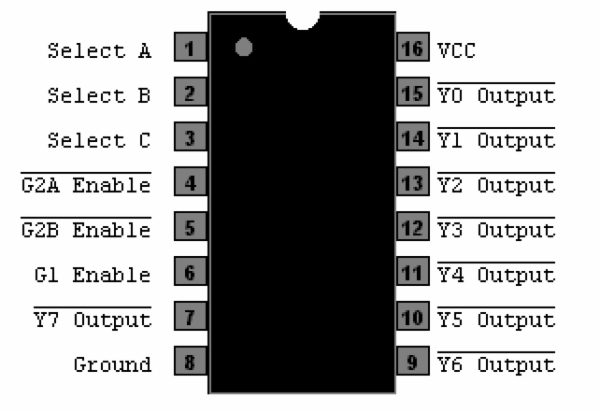
Experiment #6

Exp. 6 Decoders and Multiplexers

**Abstract and Objective:**

This lab is an introduction to decoders and multiplexers, also known as mux. The principle behind decoders and multiplexers is that some logic circuits are so widely used in digital systems that semiconductor manufacturers have packed them into convenient ICs. Some examples of these types of ICs are decoders (of course), multiplexers, counters and registers. Decoders and multiplexers are combinational circuits and are the focus of this experiment.

A decoder is a circuit that converts information from one coded form to another form and they generally have more outputs than inputs. An encoder on the other hand converts many independent input lines into a binary coded form. The '138 logic gate gets heavy use in the lab as it is our 3x8 decoder. The experimenter must be sure to be fluent in the decoder pins arrangement, which is pictured below.



**Components Used:**

* Xilinx software on PC
* The following gates/integrated circuits:

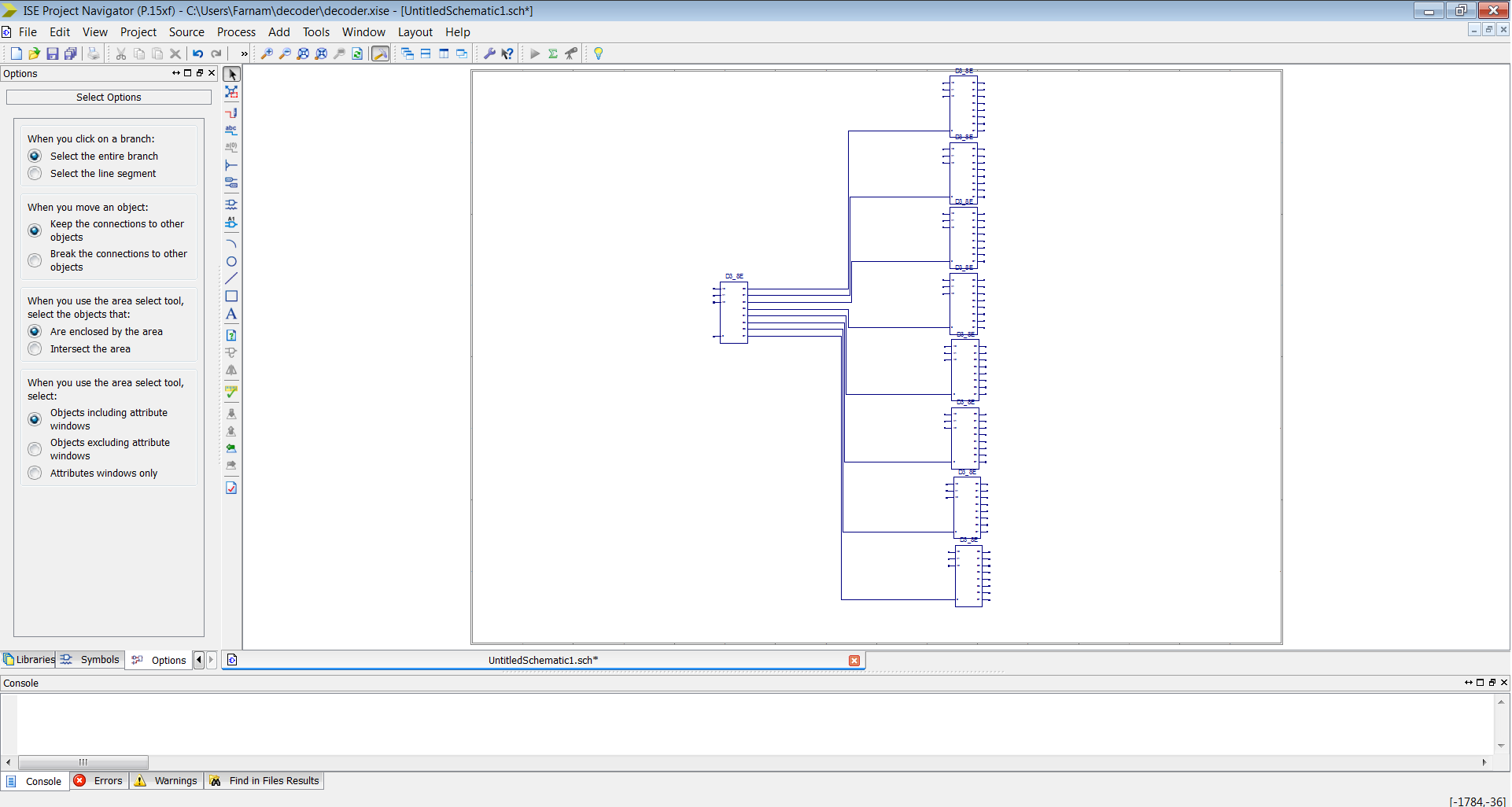
74LS04

74LS10

74LS138

74LS157

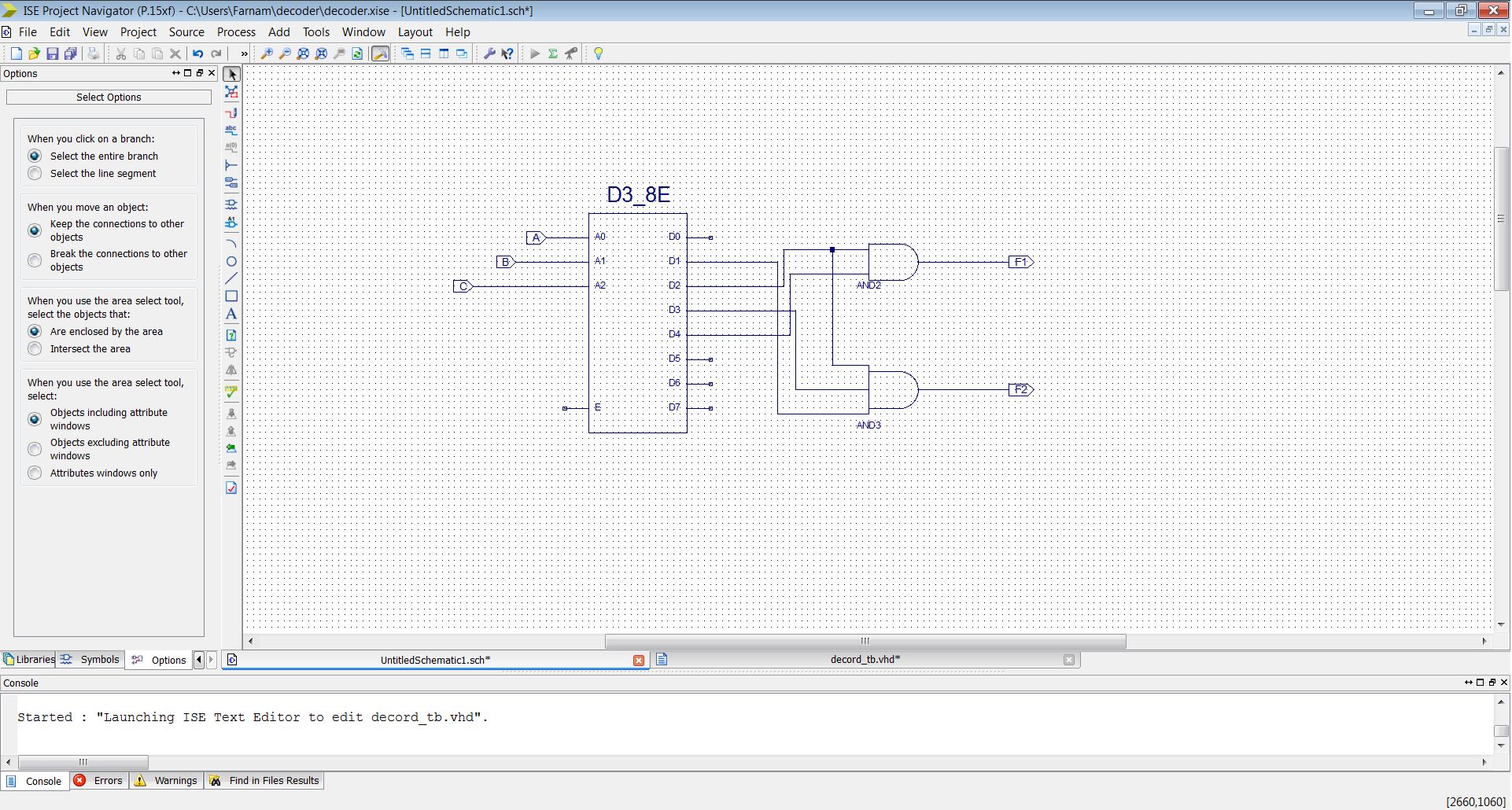
**Procedures and result:**

1. The first step in this lab is to deal with problem #1 which requires the experimenter to construct a decoding network that will recognize decimal addresses 40-47 in binary. This task was completed by connected 8 (3x8) decoders to a single '138 as the selector. See the picture on engineering paper as well as the diagram pictured below.

2. Step two asks to design a logic circuit to implement two functions, F1 & F2 described in the previous section, by making use of a decoder.

-F1 requires both memory write and output states.

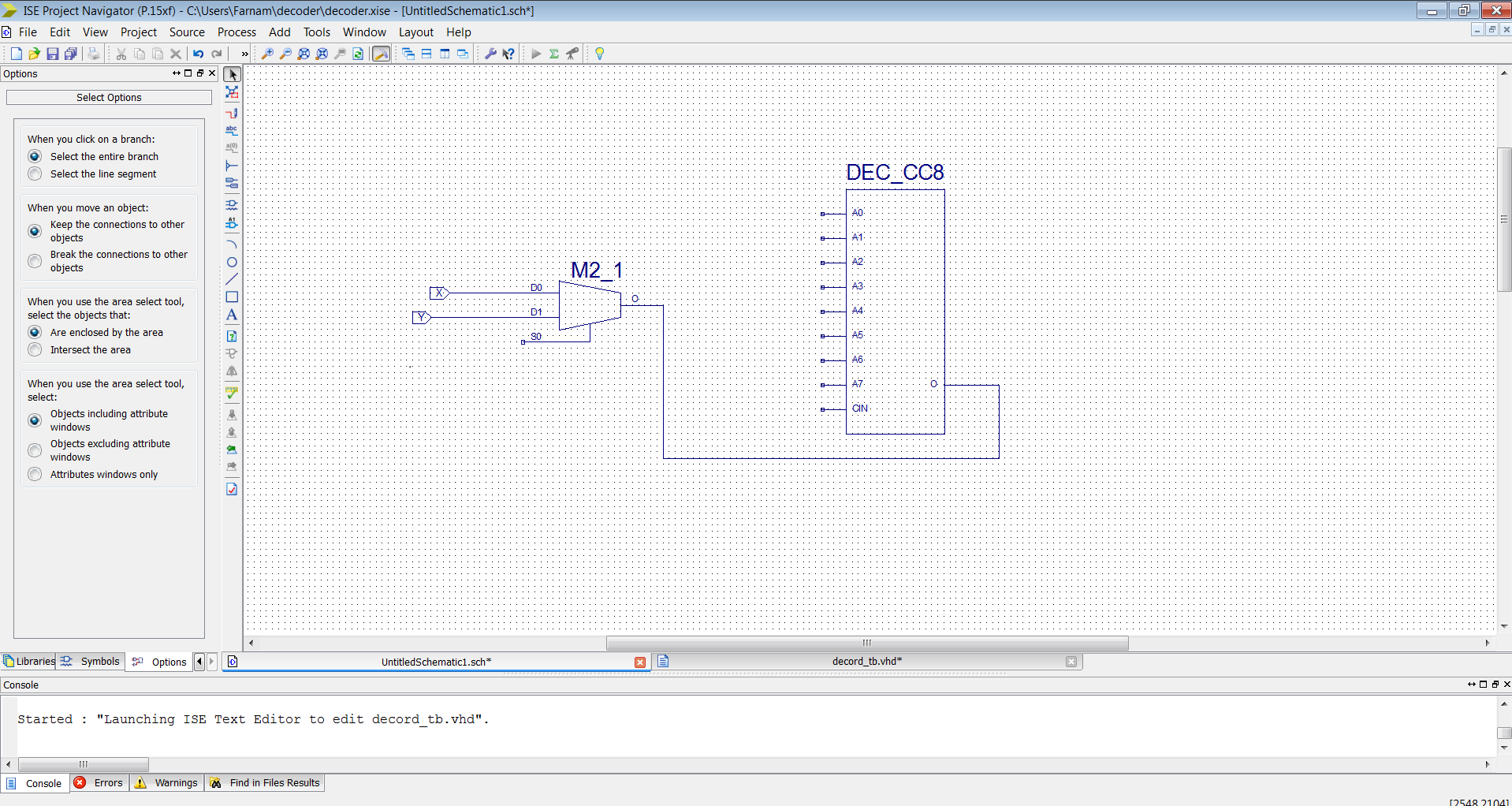
-F2 would encompass memory read, memory write and op-code fetch.



... cont #2

This circuit was implemented by simply anding the appropriate outputs to get F1 and F2.

3. The third task is to design a logic ckt that implements four functions, G1 - G4 described in this section, simultaneously using only a single '157. By connecting the appropriate inputs to the '157, the ability to switch between the signals becomes relatively simple. See the circuit diagram pictured on engineering paper.

4. Lastly is the task to design a data routing circuit which can send a single bit of information from any one of two inputs to any one of eight outputs by cascading a 2-to-1 multiplexer and a 1-to-8 demultiplexer. The selected output must follow exactly the signal applied at the selected input. See the diagram below from Xilinx.

**Conclusion and Summary:**

Now the experimenters have gained familiarity with encoder circuit and mux circuit design while also learning to implement our circuit to perform a task. It is useful to be given data points that can be confirmed by hand in order to confirm our system is working as expected. There are often issues that arise in the Xilinx software itself. Often the best fix is to save the circuit and reload it. I do have some concern that these software issues with Xilinx may become overbearing as a problem once the circuit is much more complex.

It is clear to understand why semiconductor companies have chosen to fill the market gap with convenient ICs that already contain the logic gates we commonly use in circuit design. Implementing a circuit with a decoder or MUX greatly simplifies our design but leaves gaps open for wasted hardware which can be a costly mistake.

