1st Programming Assignment, Introduction to Verilog Hardware Description Language (HDL)

The four diagrams below depicts different logic circuits that can be simulated with Verilog programs on the gate-logic level.

The example programs are for **`one-to-two (1x2) decoder`** and **`two-to-one (2x1) multiplexer`**. Program the other two circuits (with and without using arrays and module compositions) will be for you to work on. Submit your source files: ***d2x4.v, d2x4array.v, m4x1.v, m4x1array.v*** in your designated directory.

- After verifying that your code is all working, use the **`smbclient -U ...`** command and **`put`** to submit them into the folder that has your login name. Do not submit **a.out** or any other files such as the runtime output. If you do not see your login name folder, just make your own with **`mkdir`**.

- Use *`help`* to know different *`smb`* commands. If needed, issue *`mkdir`* command at the *`smb`* prompt to make a subfolder under your login folder for resubmission: e.g., ***V2*** (version 2) and ***cd V2*** to go into it, then perform submit files again. Make a *V3* folder and resubmit again if further correction is needed.

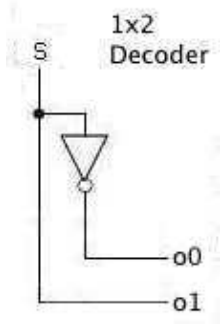
- At the start of each program, put your name in a comment section.

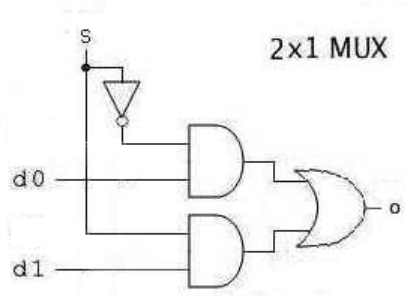
- To connect to the working server, launch two **`PuTTy`** shell terminals (one for editing programs and the other for compiling and running) from a Windows PC to server ***atoz, sp1, sp2, or sp3***. (Connect to ***titan.ecs.csus.edu*** or ***athena.ecs.csus.edu*** first if you are not in the ECS network, e.g., from home.)

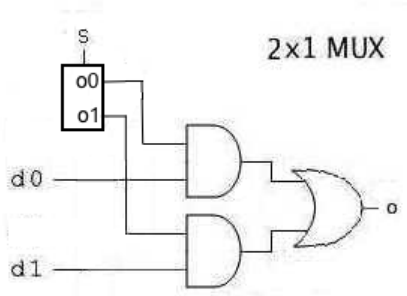
- Issue shell command to copy example programs, compile them, and run them:

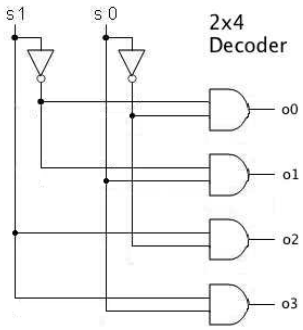
```
cp ~changw/html/137/prg/1/*.v  .   (note the space and "dot" at the end)
~changw/ivl/bin/iverilog d1x2.v
a.out  (or ./a.out)
~changw/ivl/bin/iverilog m2x1.v
a.out  (or ./a.out)
```

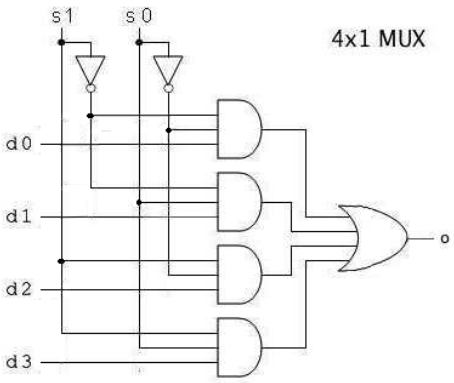
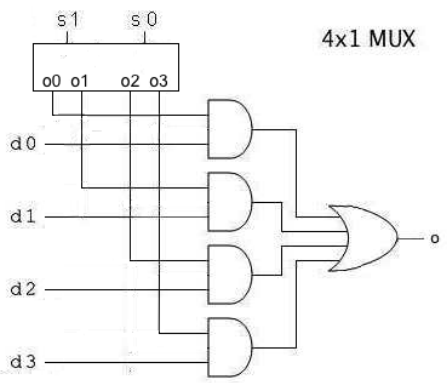
~changw/ivl/bin/iverilog d1x2array.v a.out (or ./a.out) ~changw/ivl/bin/iverilog m2x1array.v a.out (or ./a.out) Your programs should run the same way as **Runtime Output**

<table>
<tr>
<td> 

d1x2.v (no arrays)
d1x2array.v (arrays)</td>
<td> 

m2x1.v (no arrays)</td>
<td> 

m2x1array.v (arrays and modules)</td>
</tr>
<tr>
<td> 

Code/submit d2x4.v (no arrays)</td>
<td> 
</td>
<td> 
</td>
</tr>
</table>

Code/submit ***d2x4array.v (arrays and modules)***	Code/submit ***m4x1.v (no arrays)***	Code/submit ***m4x1array.v (arrays and modules)***

Use a Linux shell and issue shell commands to make subdirectories (subfolders) to organize your class work. Learn them and use program editor *vi* will enhance your software development skills. Useful materials are linked below.

- Useful Linux and vi Commands
- Compile Your Verilog Programs
- Access Dropbox from Shell
- Simple Verilog Handbook
- Complete Verilog Manual
- More Program Examples