Verilog - Signal Objects

- There are two types of signal object, *net* and *variable*

**Nets**
- Model connections (wires and busses) in structural descriptions
- Defaults to 4-state data type
- Has conflict resolution with multiple drivers
- `wire [7:0] mybus;`
- Use `wire` to connect modules
- Use `logic` to create logic

**Variables**
- Stores values assigned in initial, always, task and function blocks
- Used to describe logic behavioral modeling
- `logic [4:0] nibble_en;`
signal objects

variables store values assigned in initial, always, task, and function blocks. Used in behavioral modeling.

4-state

logic ush defined size, unsigned, init to 'X'
reg ush defined size, unsigned, init to 'X'
φ, 1, 2 integer 32-bit, unsigned, init to 'X'
time 64-bit, unsigned, init to 'X'

2-state

bit ush defined size, unsigned, init to φ
byte 8-bit, signed, init to φ
shortint 16-bit, signed, init to φ
int 32-bit, signed, init to φ
longint 64-bit, signed, init to φ

net

models connections (wires, busses) in structural descriptions. Can also be target of wire assignment.

wire
uwire ( unrereased wire, only one device)
wand
wor
tri
triand
triNor
true
false
supply φ
supply 1

net types:
supply strength (Vss)
supply strength (Vdd)

defaults to 4-state logic data type
wire can be explicitly defined as 4-state
- wire logic 0, 1, 2, 3 DATA

conflict resolution for wire assignments:

<table>
<thead>
<tr>
<th>wire</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>tri</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>z</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

logic uses 'X' by itself implies it is a variable (var logic [2:0] data)
"var" by itself implies "logic"

use logic to describe logic
use int for loop variables
declare all nets at top of module

*Default nettype none

within implicit declarations