Next Session – Variable Data Type

Net (Cont.) – tri0, tri1

TRI0

- model nets with resistive *pulldown* devices on them
- When no driver drives a tri0 net, its value is 0

	0	1	X	Z
0	0	X	X	0
1	X	1	X	1
X	X	X	X	X
Z	0	1	X	0

TRI1

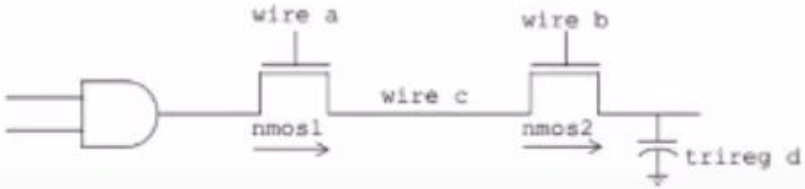
- model nets with resistive *pullup* devices on them
- When no driver drives a tri1 net, its value is 1

	0	1	X	Z
0	0	X	X	0
1	X	1	X	1
X	X	X	X	X
Z	0	1	X	1

Net (Cont.) - trireg

- used to model charge storage nodes

Driven State	At least one of driver has a value – 0,1,x ; Resolved value gets propagated and driven
Capacitive State	All drivers are having value z ; Retains last driven value, z doesn't get propagated.



simulation time	wire a	wire b	wire c	trireg d
0	1	1	strong 1	strong 1
10	0	1	HiZ	medium 1

Net (Cont.) - exercise

1. Take 2 bits vector A and B. Write a SV code that verifies all the possibilities of below block diagram, example - A = 2'b01, B = 2'bXZ, z = ? verify a,b with wand, wor truth-table, respectively.

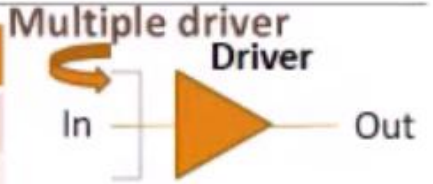
2. Sample version of SDA line of I2C protocol. Make one master and one slave module. Take inout tri port SDA in both of them. Connect both module in top module having wire SDA.

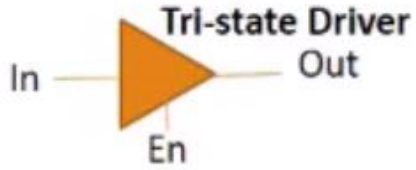
Make state diagram having 2 states in both master and slave, such that in 1st state master should be driving on SDA line & slave should be receiving it, in 2nd state slave should be driving on SDA line & master should be receiving the same. Try them with one by one – tri, triand, trior.

Hint : Both module will be having individual local En signal with individual SDA line, SDA will be connected in top.

Net (Cont.) – wire, tri, uwire, wor, trior, wand, triand

Net type	Description
wire	driven by a single continuous assignment
tri	where multiple drivers drive a net
uwire	Unidriver wire, allow only a single driver
wor, trior	when any of the drivers is 1, the resulting value of the net is 1
wand, triand	if any driver is 0, the value of the net is 0





For multiple drivers, truth table t1(wire/tri), t2(wand/triand), t3(wor/trior)

t1	0	1	X	Z
0	0	X	X	0
1	X	1	X	1
X	X	X	X	X
Z	0	1	X	z

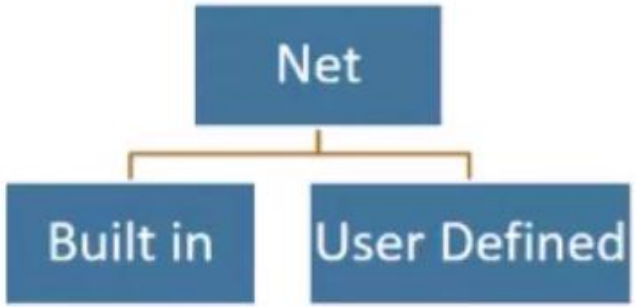
t2	0	1	X	Z
0	0	0	0	0
1	0	1	X	1
X	0	X	X	X
Z	0	1	X	Z

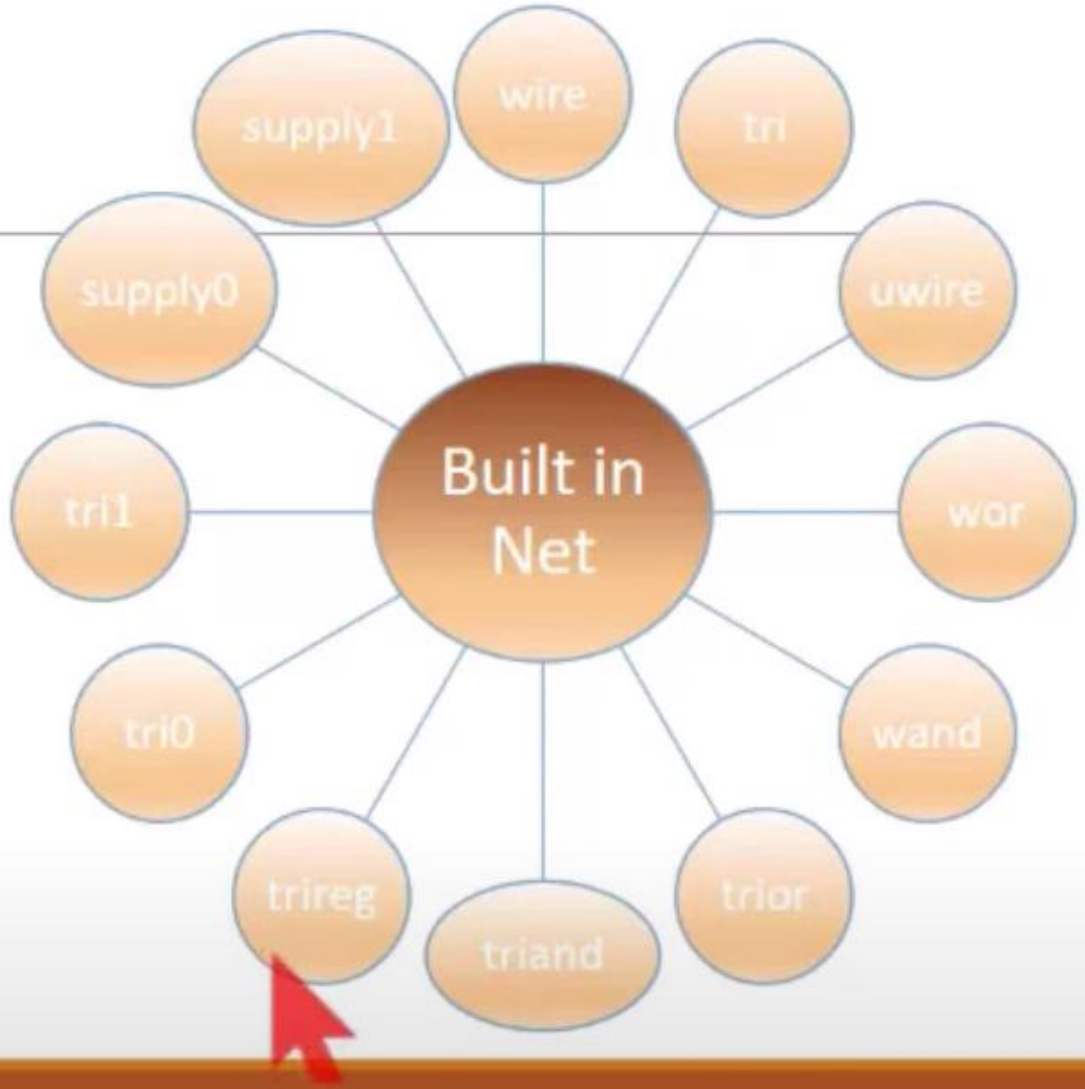
t3	0	1	X	Z
0	0	1	X	0
1	1	1	1	1
X	X	1	X	X
Z	0	1	X	Z

tri	In	En	out
	0	1	0
	1	1	1
	x	0	z

High Impedance State

Net


Net
Built in
User Defined


supply1
wire
tri
supply0
uwire
Built in Net
tri1
wor
tri0
wand
trireg
triand
trior

Groups of Data Object

NET	VARIABLE
*Differ in the way in which they are **assigned and hold values***	
➢ written by one or more continuous assignments	➢ written by one or more procedural statements
➢ Cannot be procedurally assigned	➢ written by one continuous assignment
➢ shall not store a value	➢ Stores and holds value; Last write determines the value

Data Type Classification

SINGULAR DATA TYPE

- a single value, symbol, or handle
- Integral types, string data type, class object
- though can be sliced

AGGREGATE DATA TYPE

- a set or collection of singular values
- Any unpacked structure, unpacked union, or unpacked array

classified so that
singular datatype can refer as
collective group
(treated as single vector)

Logic

- **logic** replaces **reg** of Verilog
- reg keyword gets confused with flipflop
- logic can be net or reg depending on usage

Logic – *4 state data type*		
4 – State value	Represents	Equivalent 2 – State value
0	logic zero or a false condition	0
1	logic one or a true condition	1
X	unknown logic value (don't care)	0
Z	high-impedance state (open connection)	0

Logic

- **logic** replaces **reg** of Verilog
- reg keyword gets confused with flipflop
- logic can be net or reg depending on usage

từ khóa reg quá dễ bị nhầm lẫn

Agenda

- ✓ Logic
- ✓ Data Type Classification
- ✓ Groups of Data Object
- ✓ Net