VLSI POINT

Verilog (Vectors, Arrays, Memories, Parameters, Strings)

EXAMPLE:

```
wire x,y,z;              //single bit variable
wire [15:0] data;        //16-bit data
wire [0:31] sum          //MSB is sum[0], LSB is sum[31]
reg clock;               //scalar register, default
reg [7:0] d1,d2,d3;      //3 buses of 8-bit width
reg [1:40] bus;          //vector register, 40-bit bus
```

Arrays

- In Verilog arrays are allowed for reg, integer, time, real, vector register data types
- Multi-dimensional arrays can be declared with any number of dimensions

```
reg [7:0] register_bank [15:0]; // 16 8-bit registers
reg num [31:0] ;                // array of 32 one-bit numbers
```

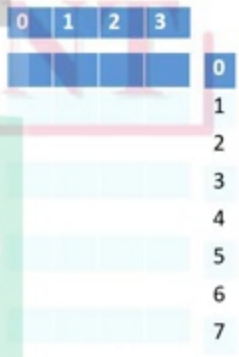
EXAMPLE:

1. **reg rega [0:7];**

   //an array of 8, 1-bit registers

2. **reg arraya [0:1] [0:3];**

   //Two dimensional array of 1-bit registers



3. **wire arrayw [0:7] [0:3];**

   //Two dimensional array of wires

EXAMPLE:

```
reg mem_1bit [0:1023];
//Memory mem_1bit with 1K 1-bit words

reg [7:0] membyte [0:1023];
//Memory membyte with 1K 8-bit words
```

Parameters

- Parameters cannot be used as variables, it is a constant value
- Parameter values for each module instance can be overridden individually at compile time
- This allows the module instances to be customized

```
parameter width_new = 8;
//Defines width_new as a constant value 8

parameter depth_new = 256;
 //Defines depth_new as a constant value 256
```

Strings

- Strings can be stored in reg
- The width of reg variables must be larger enough to hold the string
- Each string character takes up 8 bits(1 byte)
- A sequence of characters enclosed by double quotes(“ “)

```
reg [8*10 : 1] string_value;        //Declaration
string_value = “VLSI Point” ;       //assignment
```