Indian Institute of Information Technology Vadodara

CS/IT 429

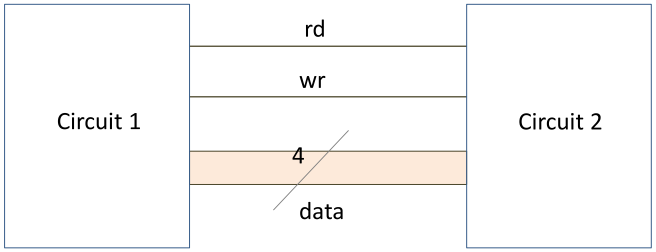
Scalars and Vectors

Dr. Yash Agrawal
Visiting Faculty, IIIT Vadodara
Associate Professor, DA-IICT Gandhinagar

Scalars and Vectors

Nets or reg data can be declared as scalars or vectors

- Scalars
- Vectors



Scalars and Vectors

Example

```
wire [3:0]  a;      // a is a four bit vector of net type. The bits are designated
                    as a[3], a[2], a[1], a[0]

reg [2:0]  b;

reg [4:2]  c;

wire [-2:2]  d;

wire [0:3]  e;
```

Scalars and Vectors

Concept of Little-endian and Big-endian notation

Example

```
input [10:0]   a;     // Little Endian Notation: Stores the LSB in smallest Address
Input [0:10]   b;     // Big Endian Notation: Stores the MSB in smallest Address
```

Store data '90ABCD12CD' in locations starting from 1000 using Little-endian and Big-endian

Address
1000
1001
1002
1003

⇨ Store the data…!

Scalars and Vectors

Example

```
module logic_vector1 (a, b, c, d, e, f, g, h, i, j, k);

input [4:0] a;
input [0:4] b;
input [-2:2] c;
input [2:-2] d;
output e, f, g, h, i;
output [4:0] j;
output [0:4] k;

assign  a = 5'b10101,       b = 5'b01100,
        c = 5'b00001,       d = 5'b01011;

assign  e = a[3],
        f = b[3],
        g = c[-2],
        h = d[-2],
        i = d[3];

assign  j = a + b,          k = c + d;

endmodule
```

⇨ Identify the output values…!

Variable Vector Part Select

Example

```
reg [255:0]  data1;       // Little Endian Notation

reg [0:255]  data2;      // Big Endian Notation

reg [7:0]  byte;


Byte = data1 [31 - :  8]          starting bit = 31, width = 8 => data[31:24]

Byte = data1 [24 + : 8]           starting bit = 24, width = 8 => data[31:24]

Byte = data2 [31 - :  8]          starting bit = 31, width = 8 => data[24:31]

Byte = data2 [24 + : 8]           starting bit = 24, width = 8 => data[24:31]
```

Scalars and Vectors

Example

```
module logic_vector2 (a, b, c, d, e, f, g, h, i, j, k, l, m, x, y);
input [10:0] a;
input [0:10] b;
output [3:0] c, d, e, f, g, h, I, j, k, l, m;
output x;
output [3:0] y;
assign  a = 11'b_011_0010_1010,
        b = 11'b_110_1010_1110,
        x = 5,       y = 4;
assign  c = a [3 : 6],
        d = a [6 : 3],
        e = b [3 : 6],
        f = a [ 10 - : 4],
        g = a [ 7 + : 4],
        h = b [ 10 - : 4],
         i = b [ 7 + : 4],
         j = b [7 + : 5],
        k = a [ (x+2) + : 4],
        l  = a [ (y+2) + : 4],
       m = a [7 + : y];
endmodule
```

⇨ Identify the output values…!

I am available/approachable at

email: yash_agrawal@iiitvadodara.ac.in
yash_agrawal@daiict.ac.in