

路科：System Verilog 系统验证技术第五课—数组

Verilog & C语言中数组的表示

定义

reg [x:0] data [y:0];对于[x:0]，这里是指的元素的大小。而data之后的为维数，就是[y:0]

type name[size1][size2]...[sizeN];

非组合数组

声明的方式 int c_style [16]//16个整数[0]..[15]

int lo_hi[0:15]//16个整数[0]..[15]

组合数组

必须在变量名前指定合并的位和数组大小 数组大小的定义格式必须是[msb:lsb]

仿真器中使用32bit的字边界，所以byte，shortint和int都是放在一个字中，而longint则存放在两个字中

存储形式的比较

bit [7:0] b_unpack[3]; // 非合并数组
b_unpack[0] 7 6 5 4 3 2 1 0
b_unpack[1] 7 6 5 4 3 2 1 0
b_unpack[2] 7 6 5 4 3 2 1 0

barray[0] [3] barray[0] [1] [6]
barray[0] 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
barray[1] 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
barray[2] 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

对数组操作的方式

for

foreach

动态数组是一个非组合数组，他的大小可以再运行时定义或者修改。动态数组支持所有数据类型作为数组的元素。

定义

bit [3:0] nibble[]; // Dynamic array of 4-bit vectors

举例

integer mem[2][]; // Fixed-size unpacked array composed
// of 2 dynamic subarrays of integers

动态数组

int idest[], isrc[3]='{5, 6, 7};
idest = new [3] (isrc); // set size and array element data
values (5, 6, 7)

动态数组自建方法

int j = addr.size;
size() ○ addr = new[addr.size() * 4] (addr); // quadruple addr array

ab.delete; // delete the array contents
delete() ○ \$display("%d", ab.size); // prints 0

关联数组产生的背景是当在对于处理数G字节寻址范围的处理器建模问题引起的。一般这个时候，我们需要存储的数据，就是需要访问的数据可能并没有这么多，因此sv提供了关联数组的方式来存储稀疏矩阵。可以理解为一个虚拟地址，只有存储的数据时候，才为该地址分配内存空间。

关联数组

bit [63:0] assoc[bit[63:0]],idx=1;

数组的方法

数组定位的方法

min

max

unique

reverse

数组排序的方法

sort

rsort

shuffle