

Языки описания схем

mk.cs.msu.ru → Лекционные курсы → Языки описания схем

Блок 19

Verilog:

Примеры комбинационных схем
с непрерывным присваиванием

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2023/2024, осенний семестр

Пример 1

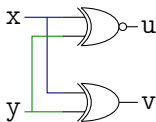
```
module M(input wire x, y, output wire u, v);  
    assign u = x && y || !x && !y;  
    assign v = !u;  
endmodule
```

Программная семантика модуля:

		u		
		y	0	1
x	0	1	X/Z	0
	X/Z	X	X	X
	1	0	X	1

		v		
		y	0	1
x	0	0	X	1
	X/Z	X	X	X
	1	1	X	0

Аппаратная семантика модуля может быть, например, такой:



Пример 2

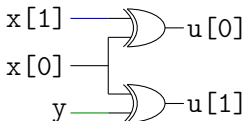
```
module M(input wire [1:0] x, input wire y,  
         output wire [1:0] u);  
    assign {u[0], u[1]} = x ^ {x[0], y};  
endmodule
```

Программная семантика модуля:

		u[0]		
		x[1]	0	X
x[0]	0	0	X/Z	1
	X/Z	X	X	X
	1	1	X	0

		u[1]		
		y	0	X/Z
x[0]	0	0	X	1
	X/Z	X/Z	X	X
	1	1	X	0

Аппаратная семантика модуля может быть, например, такой:



Пример 3

Реализация сумматора трёхразрядных чисел:¹

```
module adder_cell(input wire x, y, cin,
                  output wire sum, cout);
    assign {cout, sum} = x + y + cin;
endmodule

module adder(input wire [2:0] x, y, output wire [3:0] z);
    adder_cell cell1(.x(x[0]), .y(y[0]), .cin(1'b0),
                    .sum(z[0]), .cout(c1)); // 2
    adder_cell cell2(.x(x[1]), .y(y[1]), .cin(c1),
                    .sum(z[1]), .cout(c2));
    adder_cell cell3(.x(x[2]), .y(y[2]), .cin(c2),
                    .sum(z[2]), .cout(z[3]));
endmodule
```

1 Не реализуйте сумматор так!

Это просто демонстрация возможностей языка

2 **ВНИМАНИЕ!** Каждая необъявленная точка *по определению* имеет тип wire — в том числе точки c1 и c2