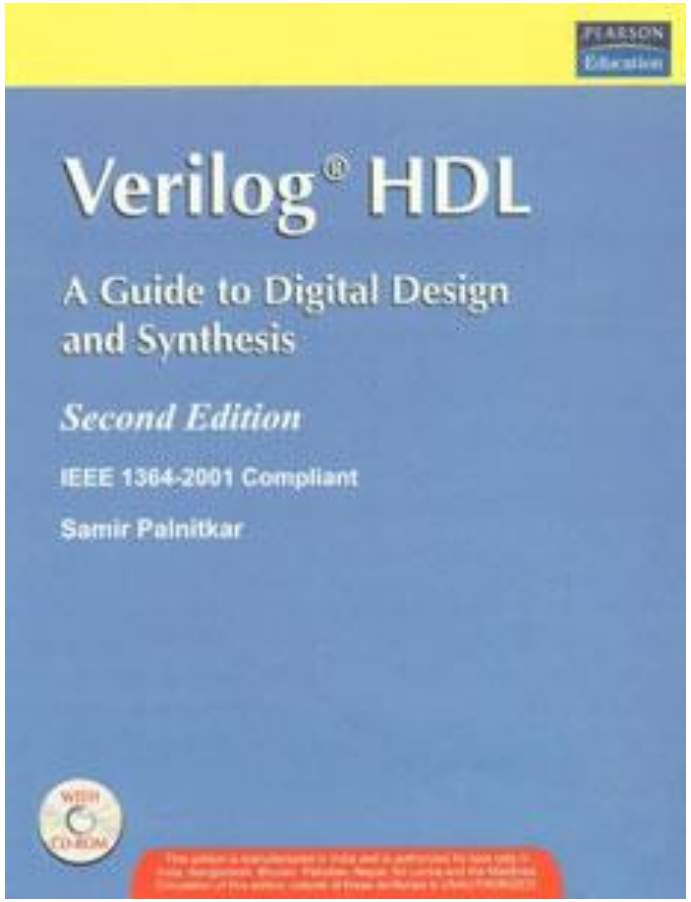
Verilog HDL

GATE-LEVEL MODELING

Reference

Verilog HDL: A Guide to Digital Design and Synthesis, 2e, Samir Palnitkar

Chapters 5

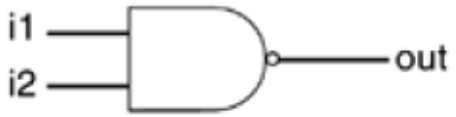


Basic logic gates in Verilog

- Verilog supports basic logic gates as predefined primitives.
- These primitives are instantiated like modules.
- But as they are predefined, a module definition is not needed.

Basic gates

and



nand

or

nor

xor

xnor

Instantiating gates

```
wire OUT, IN1, IN2;

// basic gate instantiations.
and a1(OUT, IN1, IN2);
nand na1(OUT, IN1, IN2);
or or1(OUT, IN1, IN2);
nor nor1(OUT, IN1, IN2);
xor x1(OUT, IN1, IN2);
xnor nx1(OUT, IN1, IN2);

// More than two inputs; 3 input nand gate
nand na1_3inp(OUT, IN1, IN2, IN3);

// gate instantiation without instance name
and (OUT, IN1, IN2); // legal gate instantiation
```

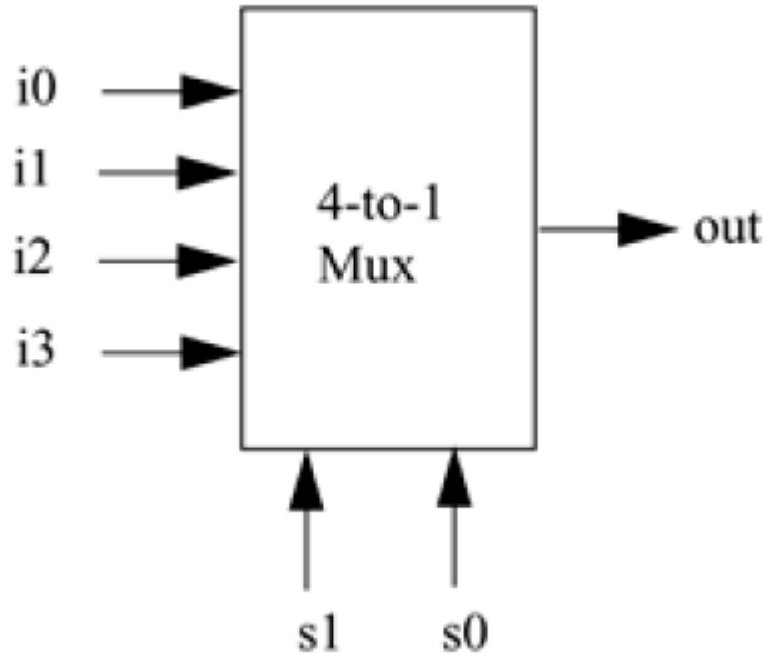
Instantiating gates

- Instance name does not need to be specified for primitives.
- This lets the designer instantiate hundreds of gates without giving them a name.
- More than two inputs can be specified in a gate instantiation.
- Gates with more than two inputs are instantiated by simply adding more input ports in the gate instantiation.

Array of Instances

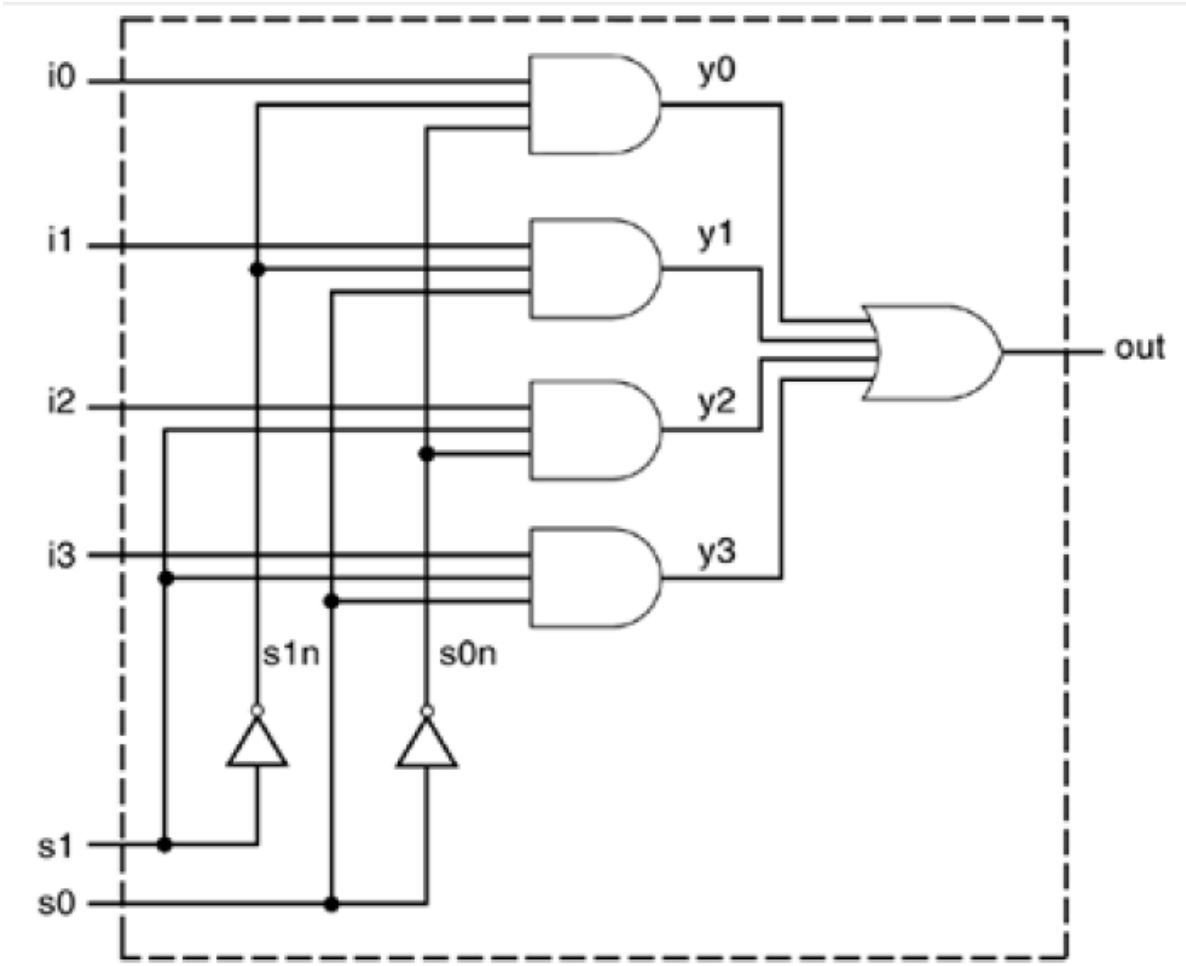
```
wire [7:0] OUT, IN1, IN2;

// basic gate instantiations.
nand n_gate[7:0](OUT, IN1, IN2);

// This is equivalent to the following
8 instantiations
nand n_gate0(OUT[0], IN1[0], IN2[0]);
nand n_gate1(OUT[1], IN1[1], IN2[1]);
nand n_gate2(OUT[2], IN1[2], IN2[2]);
nand n_gate3(OUT[3], IN1[3], IN2[3]);
nand n_gate4(OUT[4], IN1[4], IN2[4]);
nand n_gate5(OUT[5], IN1[5], IN2[5]);
nand n_gate6(OUT[6], IN1[6], IN2[6]);
nand n_gate7(OUT[7], IN1[7], IN2[7]);
```

Example : Gate level multiplexer

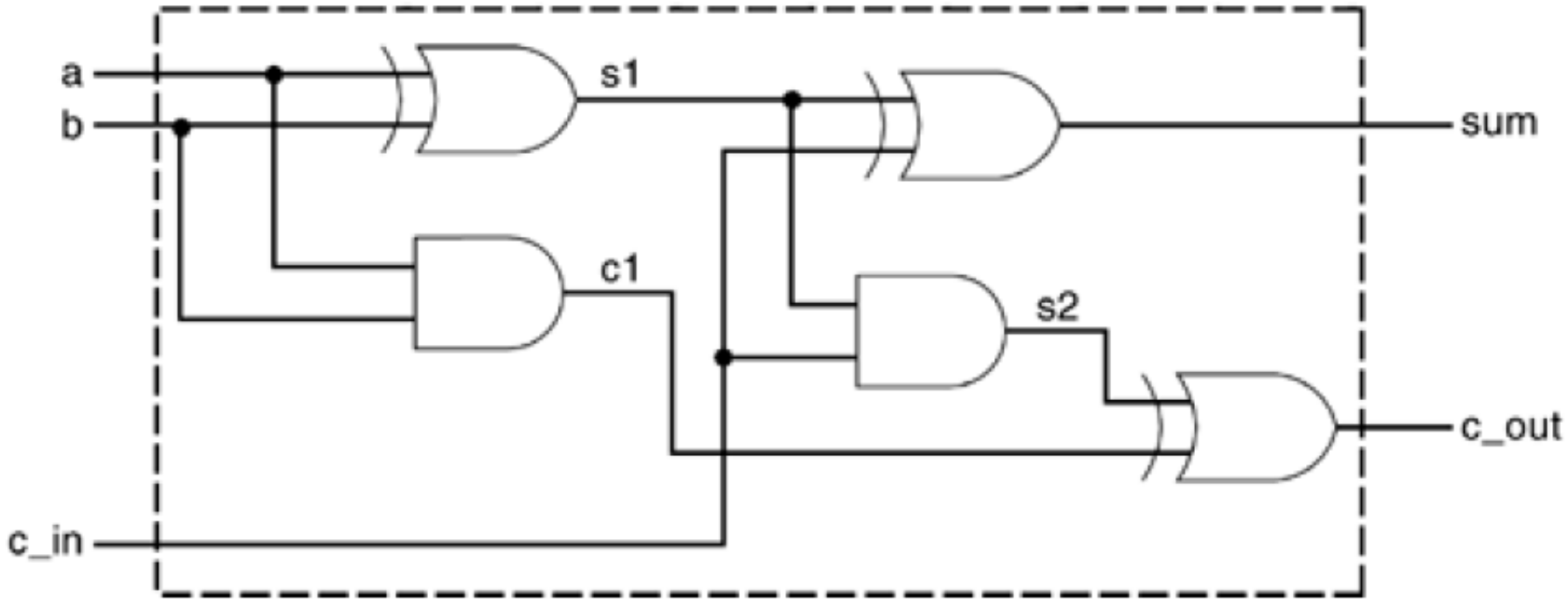


s1	s0	out
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Example : Gate level multiplexer

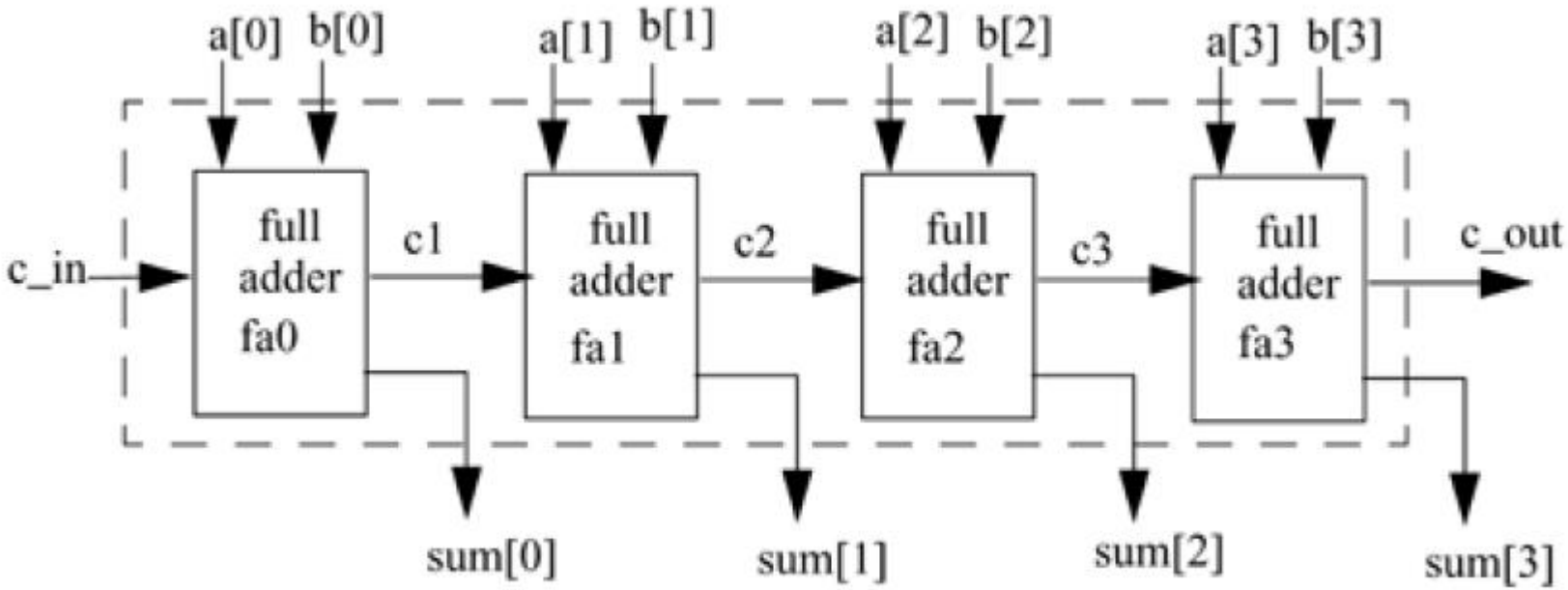


Example : 4-bit adder



1 bit full adder

Example : 4-bit adder



4-bit adder