Continuing Verilog Introduction

Further examples of Verilog code are illustrated. Additionally, the simulation techniques are discussed.

Structural Modules

Behavioral

```
module par( p, r);
input [3:0] r;
output p;
reg p;
/*  Compute the parity of r, a nibble     */
        always @(r)
        p = (r[0] ^ r[1]) ^ (r[2] ^ r[3]
endmodule
```

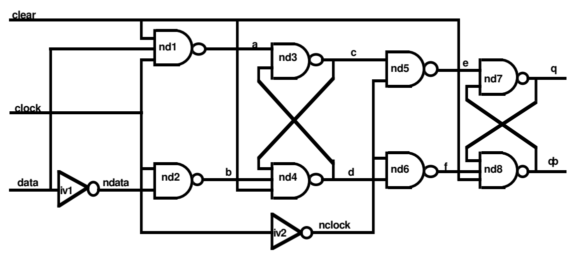
```
module xor (r, a, b);
input a, b; output r;
reg r;
always @(a or b)
 if (a == 'bx | b == 'bx)
     r = 1'bx;
 else if (a != b)
     r = 1'b1;
 else
     r = 1'b0;
endmodule
```

Structural

```
module par( p, r);
input [3:0] r;
output p;
/*  Compute the parity of r, a nibble     */
    xor x0 (w0, r[0], r[1]);
    xor x1 (w1, r[2], r[3]);
    xor x2 (p, w0,w1);
endmodule
```

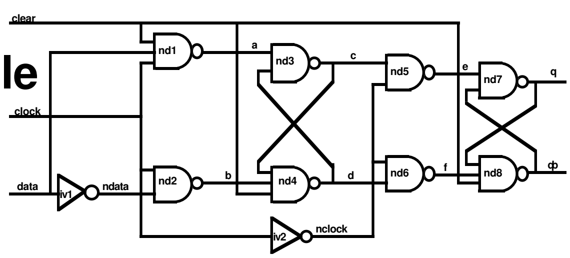
D-type Master-Slave FF

The design is intended to have all inverters and nand-gates propagate with a 10ns delay, except for nd3 and nd7 which are to have 9ns delay.



Inputs: clear, clock, data
Outputs: q, qb

D-FF Module



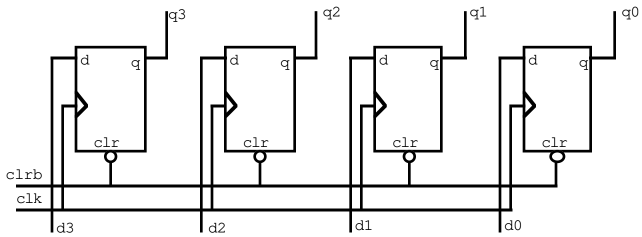

```
module flop ( data, clock, clear, q, qb);
input data, clock, clear;
output q, qb;
nand #10 nd1 (a, data, clock, clear),
         nd2 (b, ndata, clock),
         nd4 (d, c, b, clear),
         nd5 (e, c, nclock),
         nd6 (f, d, nclock),
         nd8 (qb, q, f, clear);
nand #9 nd3 (c, a, d);
nand #9 nd7 (q, e, qb);
not #10 iv1 (ndata, data);
not #10 inv (nclock, clock);
endmodule
```

4-Bit Register

```
module hardreg (d, clk, clrb, q);
input clk, clrb;
input [3:0] d;
output [3:0] q;

     flop f0 (d[0], clk, clrb, q[0]);
     flop f1 (d[1], clk, clrb, q[1]);
     flop f2 (d[2], clk, clrb, q[2]);
     flop f3 (d[3], clk, clrb, q[3]);

endmodule
```



Hardrive Module

```
module hardrive;
reg clr, clk;
reg [3:0] data;
wire [3:0] q;
`define stim #100 data = 4'b
event end_first_pass;
hardreg h1( data, clk, clr, q);

initial  begin
            clr = 1;
            clk = 0;  end
always #50 clk = ~clk;
always @(end_first_pass)  clr = ~ clr;
always @(posedge clk)
            $strobe("at time %0d clr=%b data=%d q=%d",
                        $time, clr, data, q);
initial begin
            repeat (2) begin
                  data = 4'b0000;
                  `stim 0001;
                  `stim 0010;
                  `stim 0011;
                  `stim 0100;
                  `stim 0101;
                  `stim 0110;
                  `stim 0111;
                  `stim 1000;
                  `stim 1001;
                  `stim 1010;
                  `stim 1011;
                  `stim 1100;
                  `stim 1101;
                  `stim 1110;
                  `stim 1111;
                  #200 -> end_first_pass;  end
      $finish; end
endmodule
```

Compilation

```
VERILOG-XL 1.6c.5   Oct 15, 1996  10:42:02
 * Copyright Cadence Design Systems, Inc. 1985, 1988.    *
 *     All Rights Reserved.       Licensed Software.     *
 * Confidential and proprietary information which is the *
 *      property of Cadence Design Systems, Inc.         *

 ********************************************************
 *                                                      *
 *  WARNING : +speedup WAS NOT USED ON THE COMMAND LINE  *
 *                                                      *
 *  In order to obtain maximum behavioral simulation     *
 *  performance use +speedup on the command line.        *
 *  This will become the default in future releases.     *
 *                                                      *
 ********************************************************
Compiling source file "hardrive.v"
Compiling source file "hardreg.v"
Compiling source file "flop.v"

Warning!  Too few module port connections          [Verilog-TFMPC]
        "hardreg.v", 6: f1(d[0], clk, clrb, q[0])

Warning!  Too few module port connections          [Verilog-TFMPC]
        "hardreg.v", 7: f2(d[1], clk, clrb, q[1])

Warning!  Too few module port connections          [Verilog-TFMPC]
        "hardreg.v", 8: f3(d[2], clk, clrb, q[2])

Warning!  Too few module port connections          [Verilog-TFMPC]
        "hardreg.v", 9: f4(d[3], clk, clrb, q[3])
Highest level modules:
hardrive
```

```
module hardreg (d, clk, clrb, q);
input clk, clrb;
input [3:0] d;
output [3:0] q;

      flop f0 (d[0], clk, clrb, q[0],);
      flop f1 (d[1], clk, clrb, q[1],);
      flop f2 (d[2], clk, clrb, q[2],);
      flop f3 (d[3], clk, clrb, q[3],);

endmodule
```

Do not connect

Execution

```
at time 50 clr=1 data= 0 q= x
at time 150 clr=1 data= 1 q= 0
at time 250 clr=1 data= 2 q= 1
at time 350 clr=1 data= 3 q= 2
at time 450 clr=1 data= 4 q= 3
at time 550 clr=1 data= 5 q= 4
at time 650 clr=1 data= 6 q= 5
at time 750 clr=1 data= 7 q= 6
at time 850 clr=1 data= 8 q= 7
at time 950 clr=1 data= 9 q= 8
at time 1050 clr=1 data=10 q= 9
at time 1150 clr=1 data=11 q=10
at time 1250 clr=1 data=12 q=11
at time 1350 clr=1 data=13 q=12
at time 1450 clr=1 data=14 q=13
at time 1550 clr=1 data=15 q=14
at time 1650 clr=1 data=15 q=15
at time 1750 clr=0 data= 0 q= 0
at time 1850 clr=0 data= 1 q= 0
at time 1950 clr=0 data= 2 q= 0
at time 2050 clr=0 data= 3 q= 0
at time 2150 clr=0 data= 4 q= 0
at time 2250 clr=0 data= 5 q= 0
at time 2350 clr=0 data= 6 q= 0
at time 2450 clr=0 data= 7 q= 0
at time 2550 clr=0 data= 8 q= 0
at time 2650 clr=0 data= 9 q= 0
at time 2750 clr=0 data=10 q= 0
at time 2850 clr=0 data=11 q= 0
at time 2950 clr=0 data=12 q= 0
at time 3050 clr=0 data=13 q= 0
at time 3150 clr=0 data=14 q= 0
at time 3250 clr=0 data=15 q= 0
at time 3350 clr=0 data=15 q= 0
L35 "hardrive.v": $finish at simulation time 3400
4 warnings
1517 simulation events
CPU time: 0.5 secs to compile + 0.1 secs to link + 0.3 secs in simulation
End of VERILOG-XL 1.6c.5   Oct 15, 1996  10:53:52
```