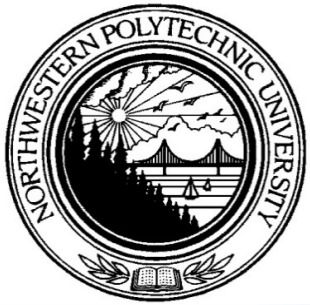



EE461 Verilog-HDL

Week 1 Introduction



Alex Yang, Engineering School, NPU

Week 1 Outlines

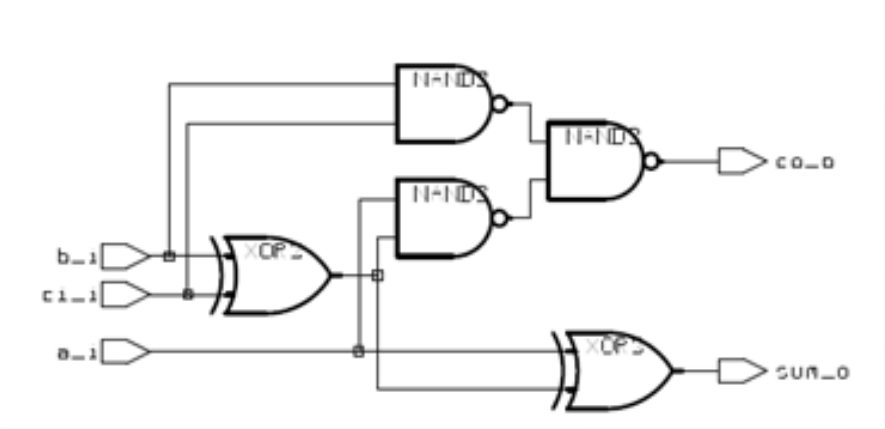
- Why needs Verilog?
- First Program “Hello World”
- Verilog Abstraction Levels
- ASIC(Application-Specific IC) Design Flow
- Introduction to Design Simulation Tools

Why needs Verilog?

- Traditional Full Adder Design (Paper & Pen)
  - Truth Table
  - Logic Simplification
  - Logic Function
  - Gate Level Circuit Built-up

Why needs Verilog?

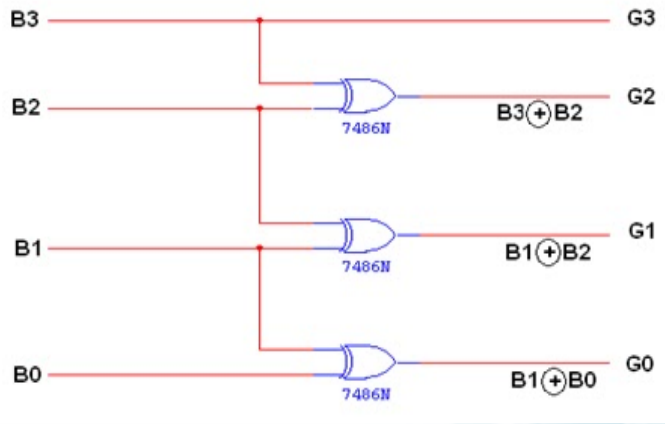
- Program-Based Design In Verilog
  - Write program in terms of design specification
  - Generate Digital Circuit in Design Tool (Logic Synthesizer)



Why needs Verilog?

- Design Gray code – Binary code Converter

Dec	Gray	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111



First Program “Hello World”

```
//------------------------------------------------------------
// File Name/Module Name: helloWorld.v
// Author: Alex Yang
// Northwestern Polytechnic University Fremont CA
// Description : Print 'hello world'
//------------------------------------------------------------
module helloWorld ;
        initial begin
                $display ("Hello World!!!");
                #10 $finish;
        end
endmodule // End of Module helloWorld
```

First Program “Hello World”

- Run the Program in VCS Compiler (Linux)

>> vi helloWorld.v

(type in the example code as above, and then save & quit)

>> source /ee/setup/synopsys/setup.cmd

>> vcs helloWorld.v

>>./simv

You will see “Hello World!!!” on the monitor

Verilog Abstraction Levels

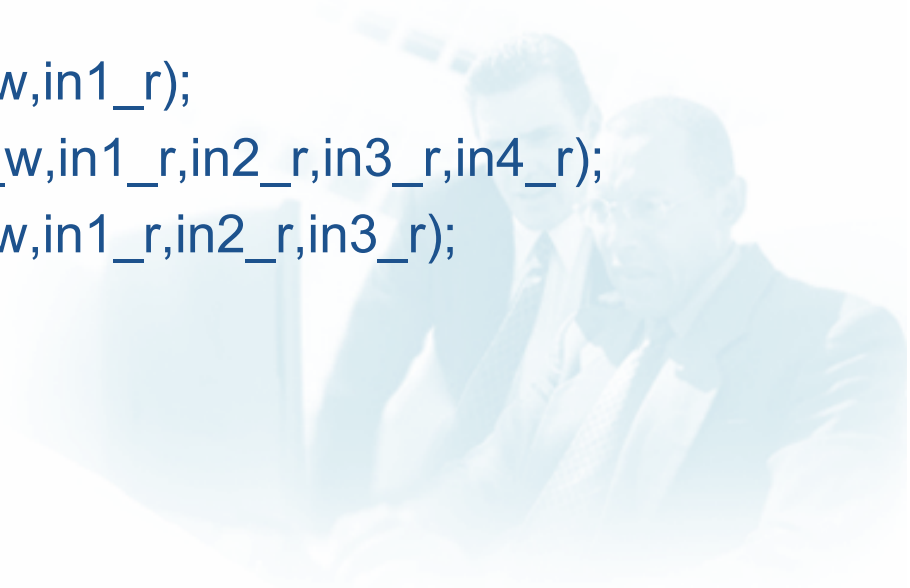
- Gate Level (Low Level)
  - Like Assembly Language
  - Design product
- RTL Level (Register Transistor Level)
  - Like C language
  - Design Language
  - Can Generate Gate Level Verilog Code
- Behavioral Level (High Level)
  - Like C++/Java,etc.
  - Design Ideas Verification
  - Can’t Generate Gate Level Verilog Code

Verilog Abstraction Levels

- Gate Level (Low Level)

```
module gates();
        wire out0_w;
        wire out1_w;
        wire out2_w;
        reg  in1_r,in2_r,in3_r,in4_r;

        not U1(out0_w,in1_r);
        and U2(out1_w,in1_r,in2_r,in3_r,in4_r);
        xor U3(out2_w,in1_r,in2_r,in3_r);
endmodule
```



Verilog Abstraction Levels

- RTL Level (Register Transistor Level)

```
module oneBitFA1(
        input wire a_i,
        input wire b_i,
        input wire ci_i,
        output wire sum_o,
        output wire co_o
);
        assign   {co_o,sum_o} = a_i + b_i + ci_i;
endmodule
```

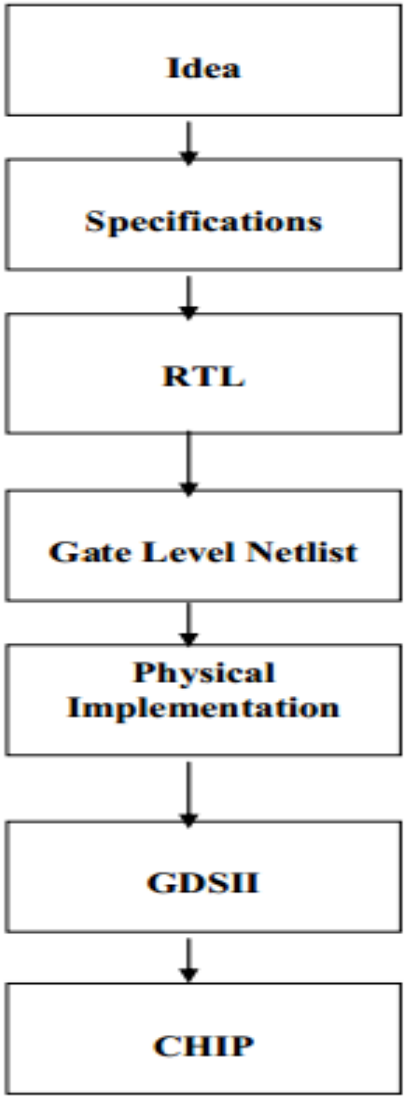
Verilog Abstraction Levels

Behavioral Level (High Level)

```
module GCDBehav  #( parameter W = 16 ) (
                                                input [W-1:0] inA, inB,
                                                output [W-1:0] out
);
                reg [W-1:0] A, B, out, swap;
                integer done;

                always @(*) begin
                                done = 0; A = inA; B = inB;
                                while ( !done ) begin
                                                if ( A < B )
                                                                swap = A;
                                                                A = B; B = swap;
                                                else if ( B != 0 )
                                                                A = A - B;
                                                else
                                                                 done = 1;
                                 end
                                out = A;
                end
endmodule
```

ASIC Design Flow Chart



Intro to Design Sim. Tools

- VCS Simulator – Synopsys
  - In Linux System
- ModelSim – Mentor Graphics
  - In Windows/Linux
- Quartus II – Xilinx
  - Windows
- Silos – Silvaco
  - Windows