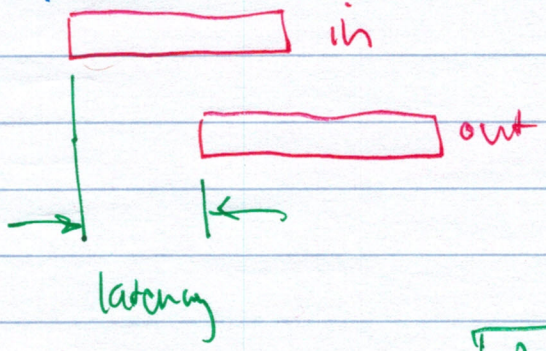
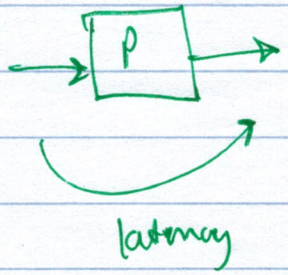
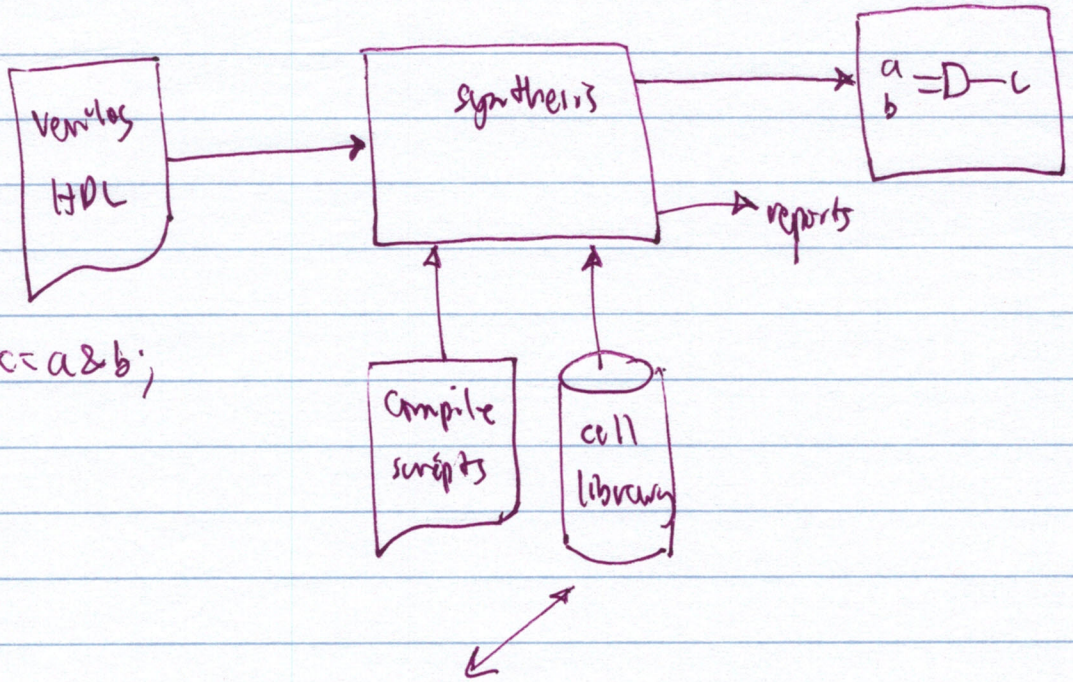
April 6, 2021



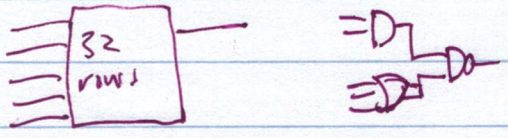




c=a&b;

1) Std Cell Library : INV, NANDs, NORs, XORs, FFs, AOI
Multiple strengths for each INV_X1, INV_X2, ...

2) FPGA - LUT, CLB, slice



Verilog (HDL)

- Digital Sys Design is important
- Verilog tells computer your design
+ Submitted work in verilog
+ Know lang. basics

Std. Prog. lang.: C, C++, python, etc.

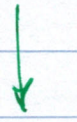
- code an algorithm
- calc. a result
- SW

HDL

- code Hw
- Hw designer
- Best when you use basics of lang

Design Process

- Think
- Draw diagrams
- Signal names
- Logic vs. FFs/registers
- Pipeline
- Think



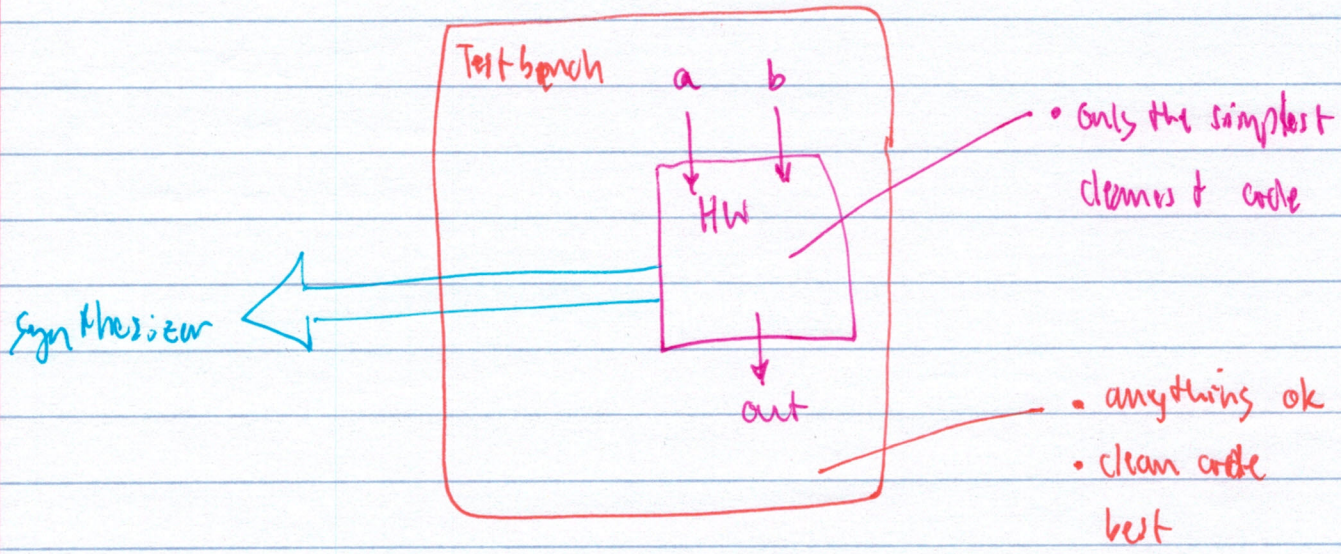
- Then
  - write verilog
  - test
  - optimize it

Verilog

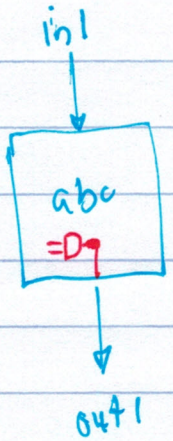
- invented 1983
- versions 2001 IEEE 1364-2001
  - signed
  - always @(*)
- 2005
- similar to C
- high-tech companies

VHDL

- 1987 by US DoD
- similar to Ada
- govt and defense
  east coast
  universities



Modules



1) module abc (in1, out1)
   input in1;
   output out1;

   wire out1;
   assign out1 = a & b;

   end module

* 2) module abc (
   input in1,
   output wire out1 // declare wire
   );
   wire out1;

   endmodule

Comments

~~~~ // comment

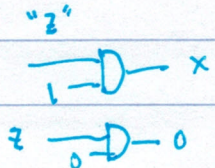
/* comment

*/

Values 0 or 1

x or X // unknown

z or Z // floating, high impedance


~~~~

Constants

- binary b
- hex h
- octal o
- decimal d

↕

[size in bits]' [base][value]

"_" helpful but ignored

	binary
- 1'b0	0
- 1'b1	1
- 4'b0101	0101
- 5'h0B	01011
- 16'h3F09	0011 1111 0000 1001
- 8'b0101_1010	01011010
- ~~8'd3~~	
8'd003	
- 8 bits 4	
8'd004	
- 6'h0B	00 1011

8 bit range [0, +255] = 3 digits

Concatenation

all = {g, h, j, k, m} ← 6 bits each

↖ 30 bits

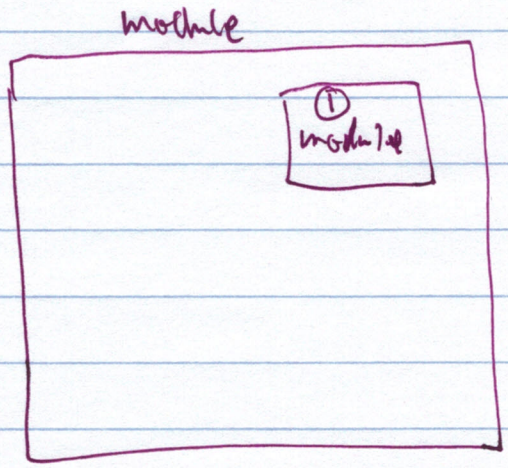
ex: b = {a[3], a[3], a[3:0]}; // sign extension

3 Ways to specify HW

① instantiate a module

② wire, declare with assign

③ reg, declare with always block



$a_3\ a_2\ a_1\ a_0$

$a_3\ a_3\ a_3\ a_2\ a_1\ a_0 = b$

```
b = {a[3], a[3], a};
```