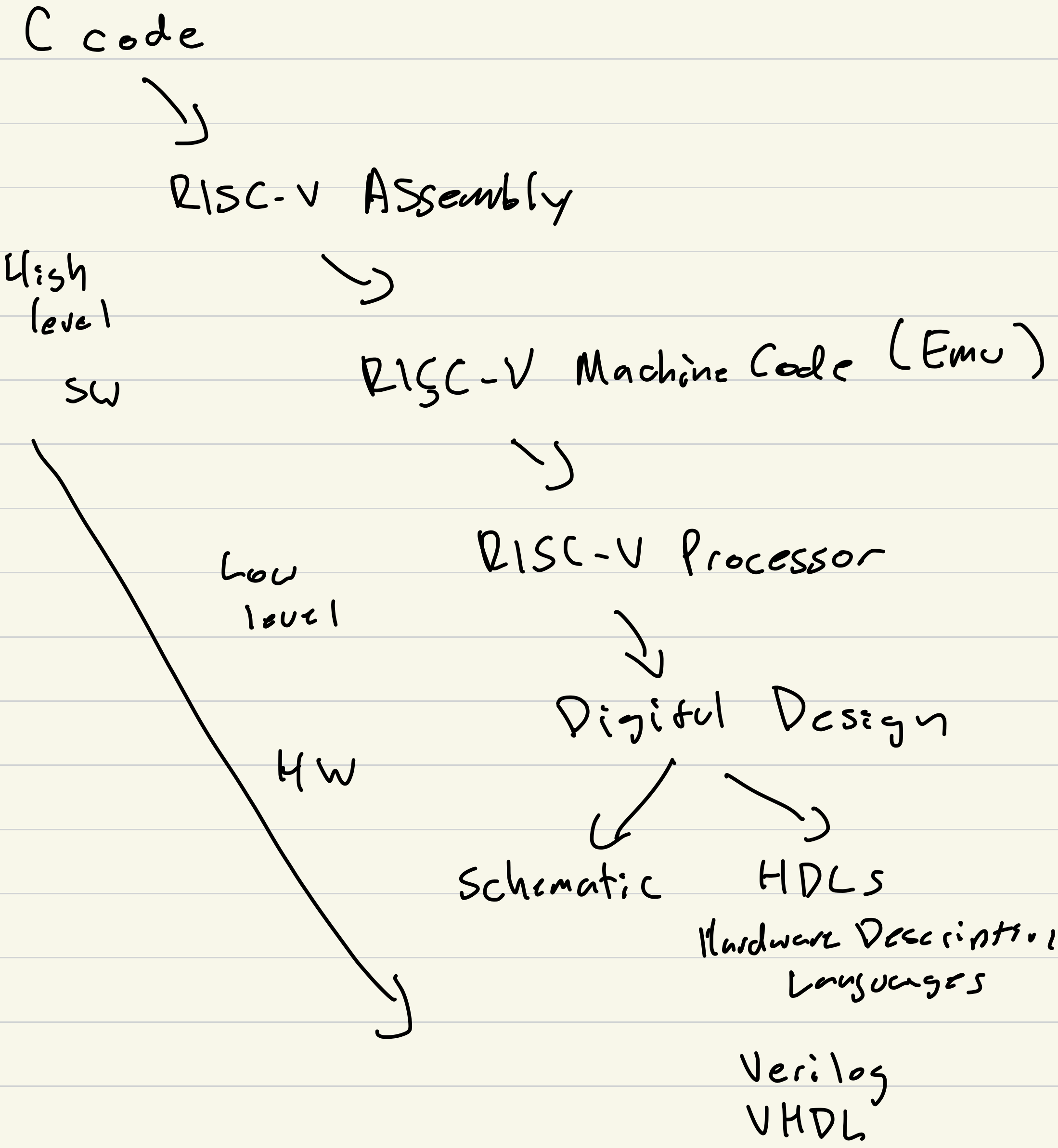
CS 631-01 Digital Design
Combinational Logic



C code

RISC-V Assembly

High level

SW

RISC-V Machine Code (Emu)

Low level

RISC-V Processor

HW

Digital Design

Schematic

HDLs

Hardware Description Languages

Verilog

VHDL

SW

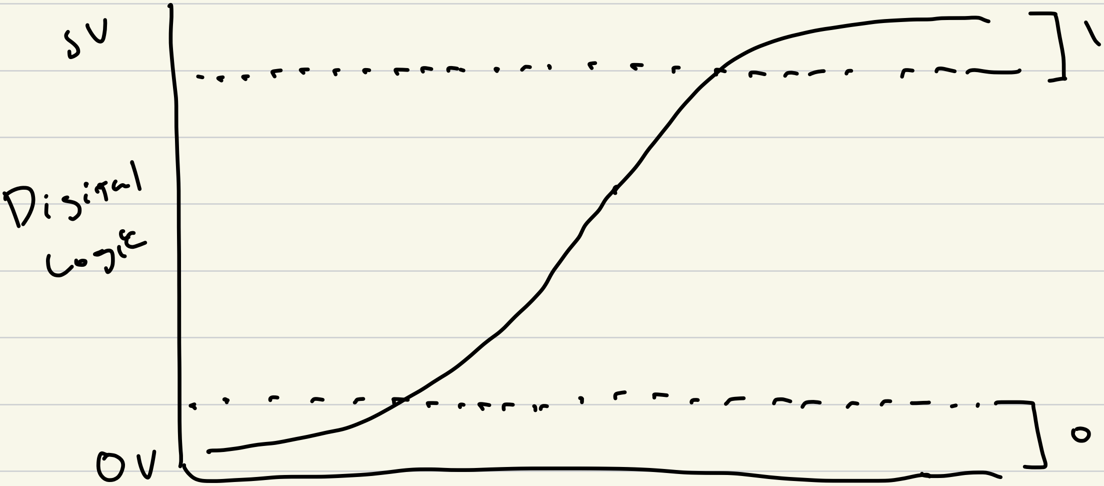
C / ASM

ISA

Digital Logic

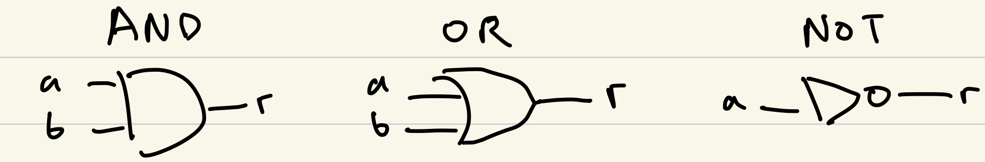
HW

Analog → Digital



Wires

devices → gates

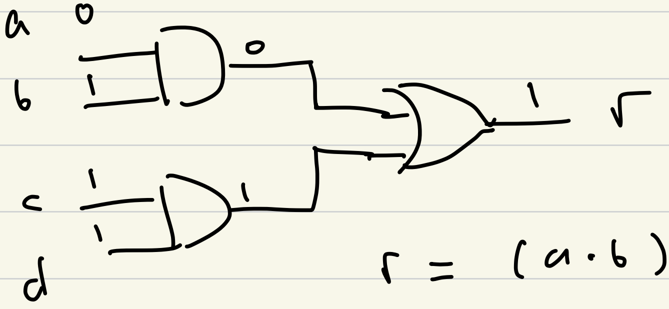


C	$r = a \& b$	$r = a \vert b$	$r = \sim a$
Boolean Algebra	$r = a \cdot b$	$r = a + b$	$r = \bar{a}$
Logic	$r = a \land b$	$r = a \lor b$	$r = \neg a$

a	b	r
0	0	0
0	1	0
1	0	0
1	1	1

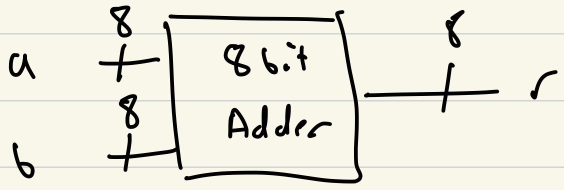
a	b	r
0	0	0
0	1	1
1	0	1
1	1	1

a	r
0	1
1	0



$r = (a \cdot b) + (c \cdot d)$

Goal



sum-of-products

Sum of two 1-bit numbers

XOR $\text{sum} = a \oplus b$

	a	b	sum
	0	0	0
①	0	1	1
②	1	0	1
	1	1	0

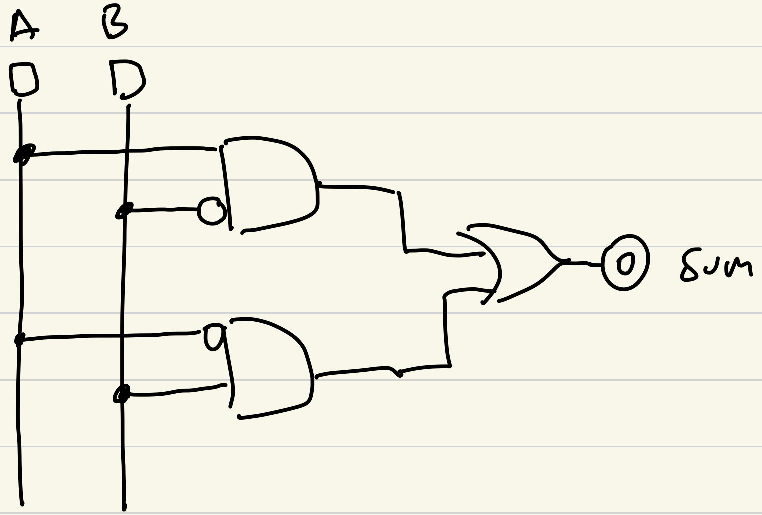
$$\text{sum} = (\bar{a} \cdot b) + (a \cdot \bar{b})$$

sum

products

$a = 0 \quad b = 1$

$$\begin{aligned} \text{Sum} &= (\bar{0} \cdot 1) + (0 \cdot \bar{1}) \\ &= (1 \cdot 1) + (0 \cdot 0) \\ &= 1 + 0 \\ &= 1 \end{aligned}$$



Sum-of-products

1) Define your function

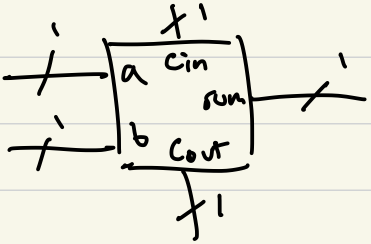
2) Build a truth table

3) Identify rows with output = 1

4) Construct product terms for each row
   a) don't invert if input is 1
   b) invert if input is 0

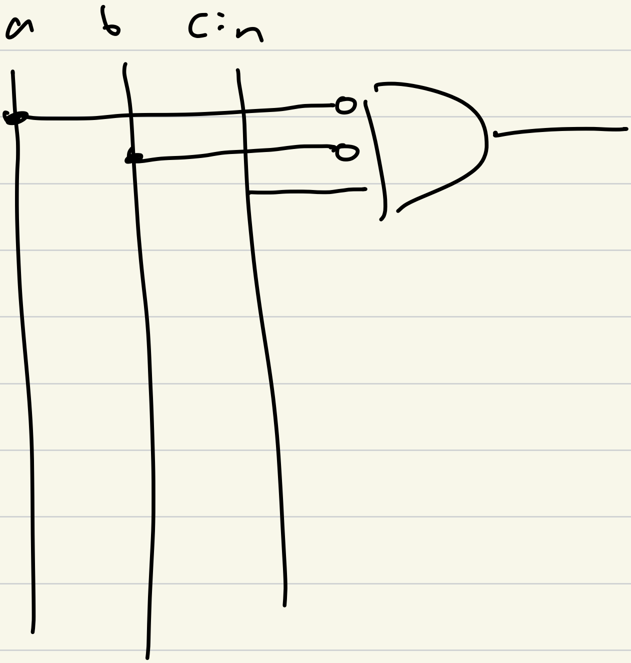
5) Sum (+) all product terms

1 bit full adder



	a	b	cin	sum	cout
	0	0	0	0	0
①	0	0	1	1	0
②	0	1	0	1	0
	0	1	1	0	1
③	1	0	0	1	0
	1	0	1	0	1
	1	1	0	0	1
	1	1	1	1	1

$$sum = (\bar{a}\cdot\bar{b}\cdot c_{in}) + (\bar{a}\cdot b\cdot\bar{c}_{in}) + (a\cdot\bar{b}\cdot\bar{c}_{in}) + (a\cdot b\cdot c_{in})$$


a
b
cin