CS 631-02 Digital Design
Combinational Logic

C code

↓

RISC-V Assembly

↓

RISC-V Machine Code (Emu)

↓

RISC-V Processor

↓

HW

Digital Design

↓

Schematic

HDLs

Hardware Description Languages

Verilog

VHDL

High Level → Low Level

SW

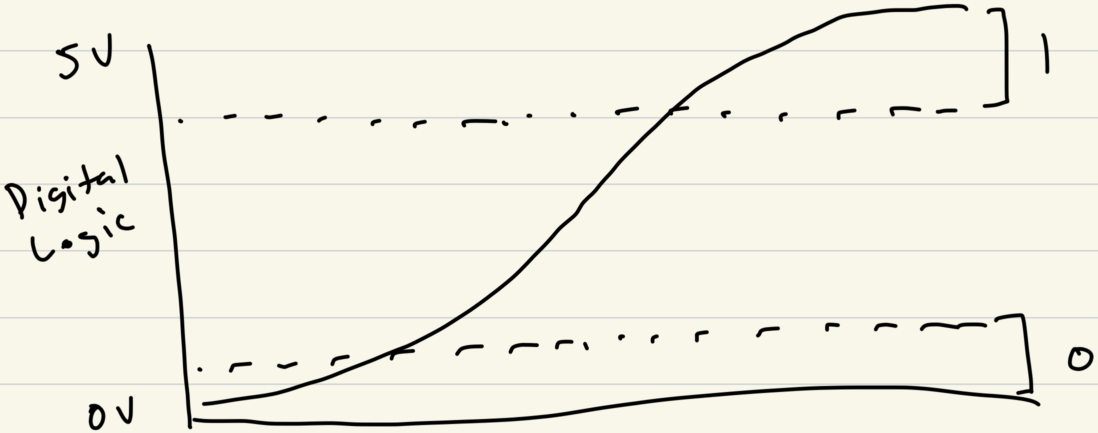
C / ASM

ISA

HW

Digital Logic

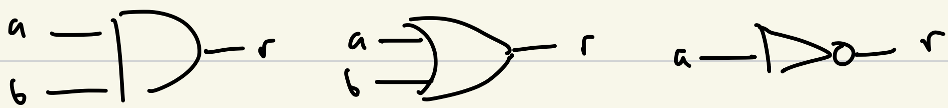
Analog



Wires

Device → gates

	AND	OR	NOT



	AND	OR	NOT
C	$r = a \& b$	$r = a \mid b$	$r = \sim a$
Boolean Algebra	$r = a \cdot b$	$r = a + b$	$r = \overline{a}$
Logic	$r = a \wedge b$	$r = a \vee b$	$r = \neg a$

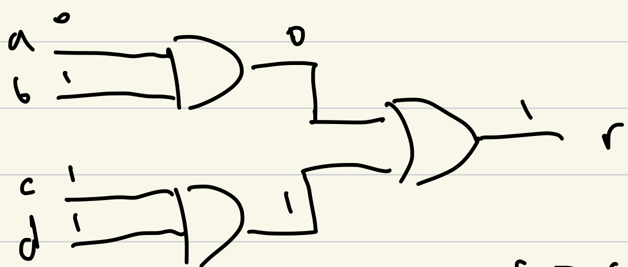
a	b	r
0	0	0
0	1	0
1	0	0
1	1	1

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

inclusive

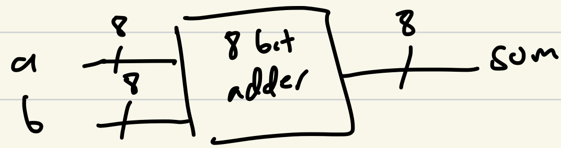
a	r
0	1
1	0

Combinational logic



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

Goal



Sum-of-products

Sum two 1-bit numbers

XOR $sum = a \oplus b$

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

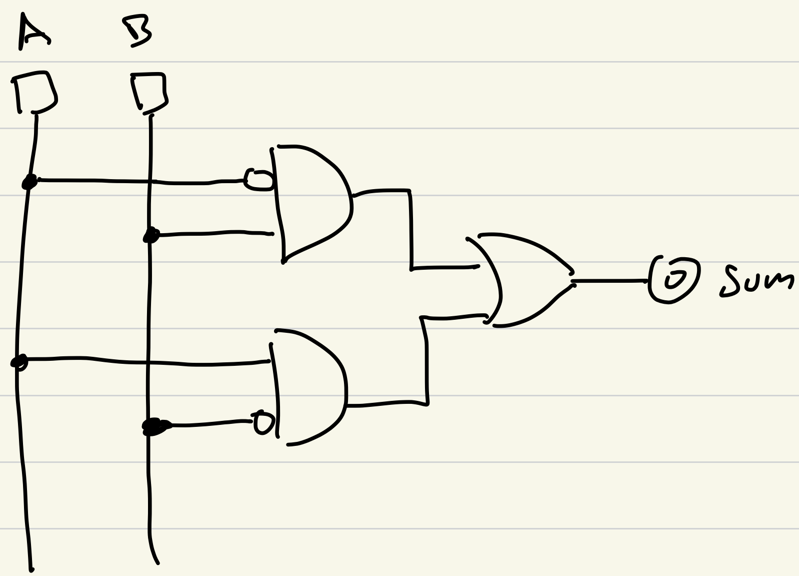
$sum = (\bar{a} \cdot b) + (a \cdot \bar{b})$

sum

products

$a = 0 \quad b = 1$

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



Sum-of-products

1) Define your function

2) Build a truth table

3) Identify rows with output == 1

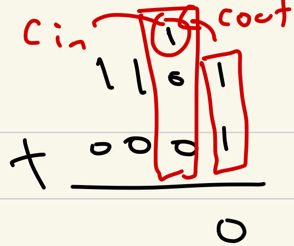
4) Construct product terms for each row (3)

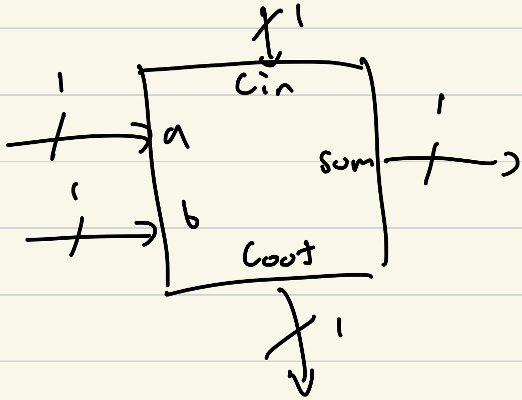
a) don't invert if input is 1

b) invert if input is 0

5) Sum (+) all product terms

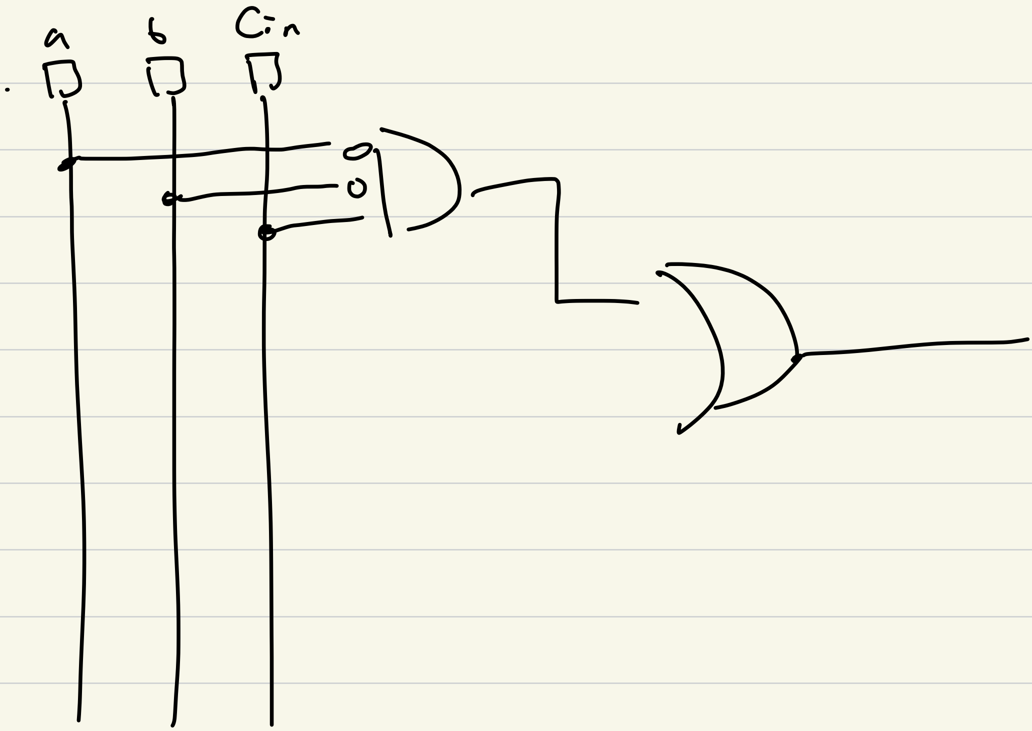
1 bit Full adder





	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
④	1	1	1	1	1

$$Sum = (\bar{a} \cdot \bar{b} \cdot c_{in}) + (\bar{a} \cdot b \cdot \overline{c_{in}}) + (a \cdot \bar{b} \cdot \overline{c_{in}}) + (a \cdot b \cdot c_{in})$$


a
b
Cin