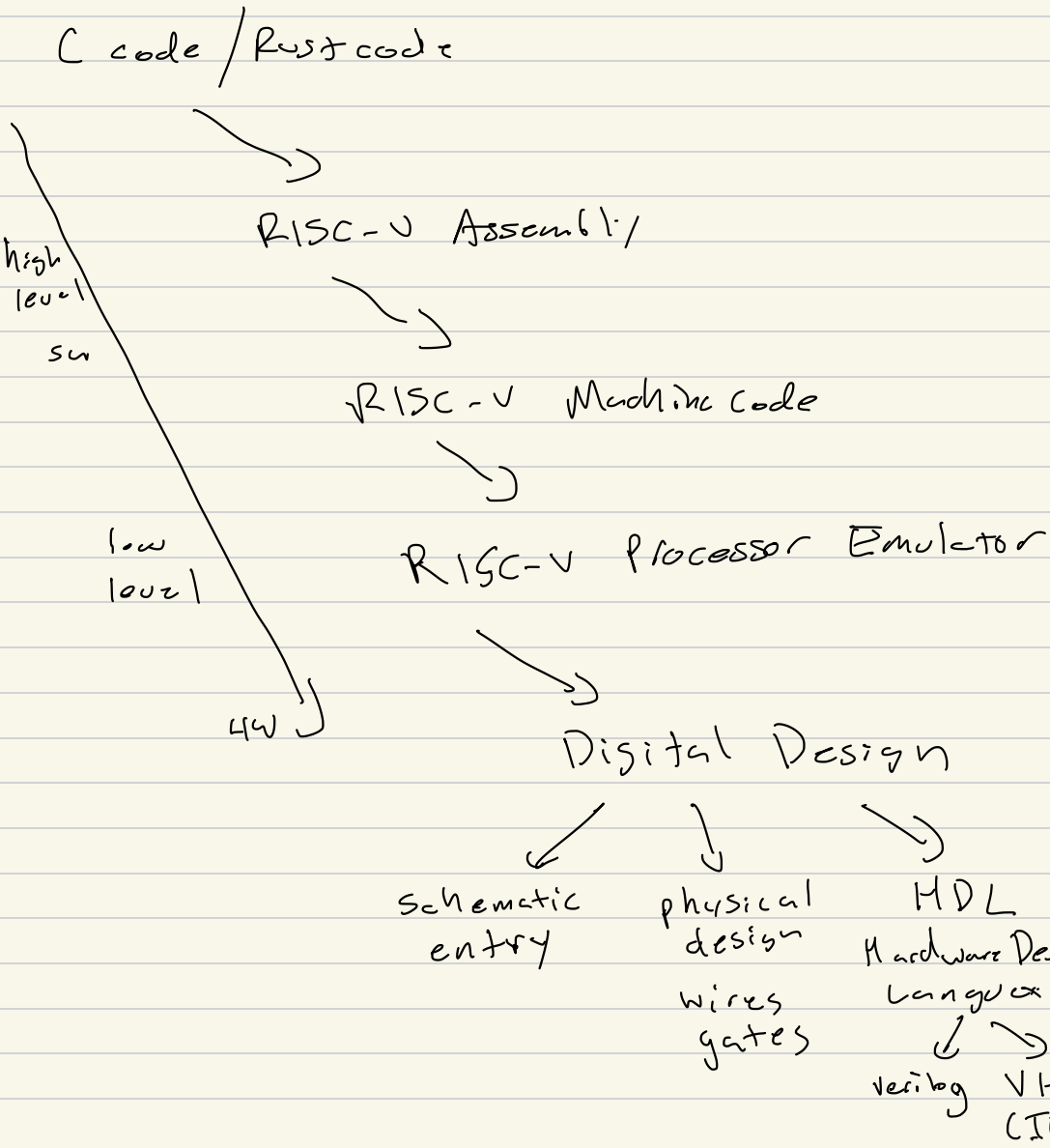
CS 631-02 Digital Design 2026-04-02

combinational logic / Sequential Logic



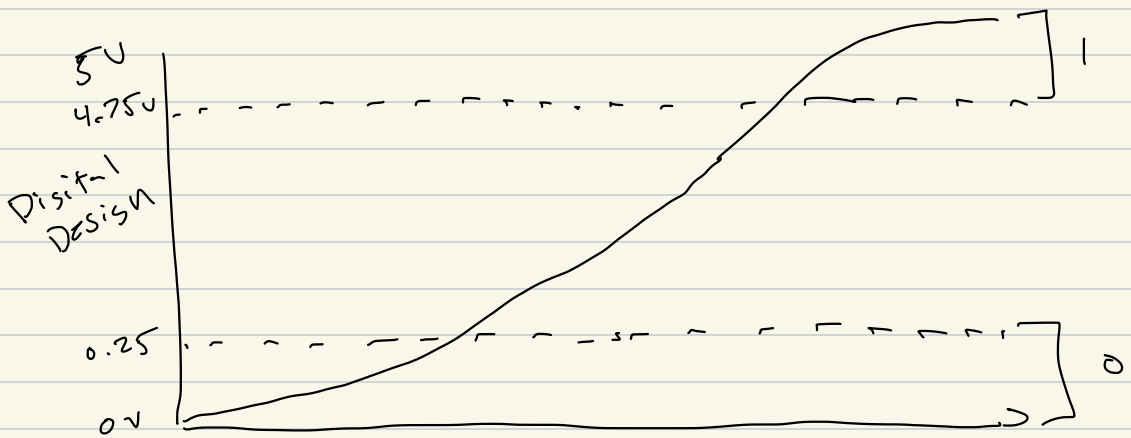
SW

C / Rust / ASM

ISA RISC-V

Digital Design

Analog → Digital

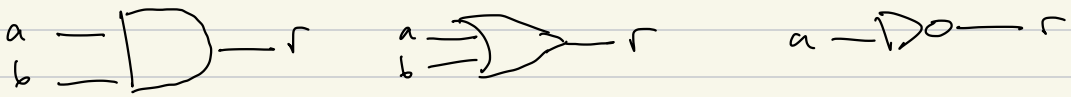


power source

wires

devices → gates

AND OR NOT



C / Rust

`r = a & b` `r = a | b` `r = !b`

Boolean Algebra

$r = a \cdot b$ $r = a + b$ $r = \overline{a}$

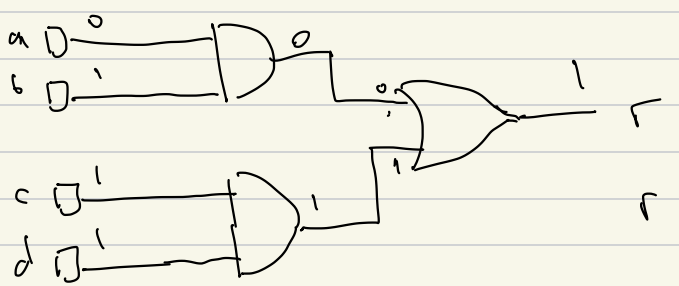
Logic

$r = a \wedge b$ $r = a \vee b$ $r = \neg a$

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

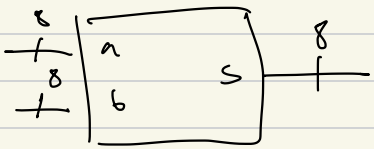
a	b	r
0	0	0
0	1	1
1	0	1
1	1	1

a	r
0	1
1	0



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

Goal: 8-bit adder



Sum-of-products

sum of two 1-bit values

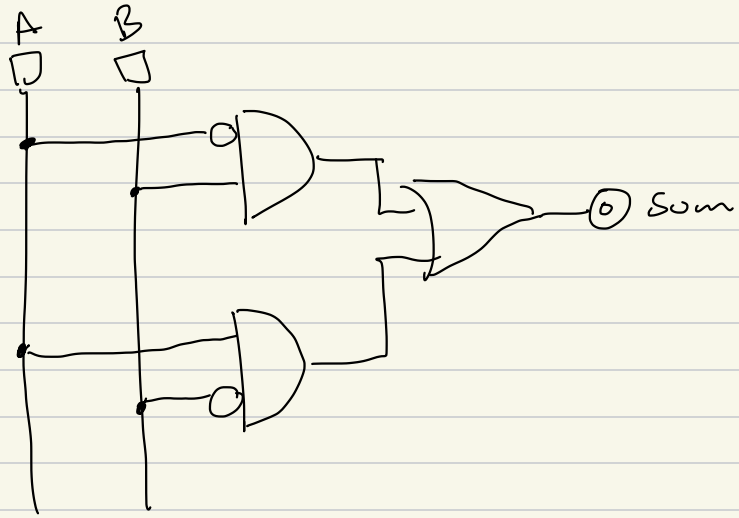
sum = a ⊕ b

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

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

sum

products

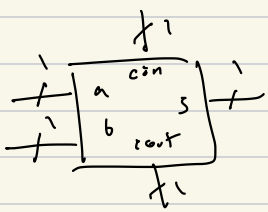


sum-of-products

1) define your function (on binary inputs)
2) Build the 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

---

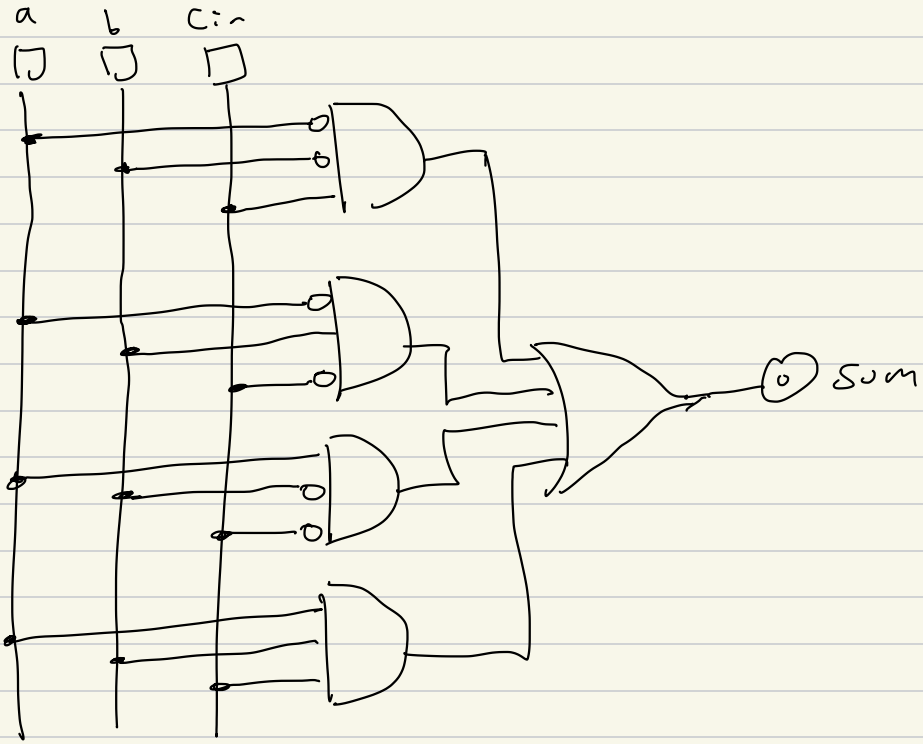
1 bit full adder



a	b	cin	sum	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

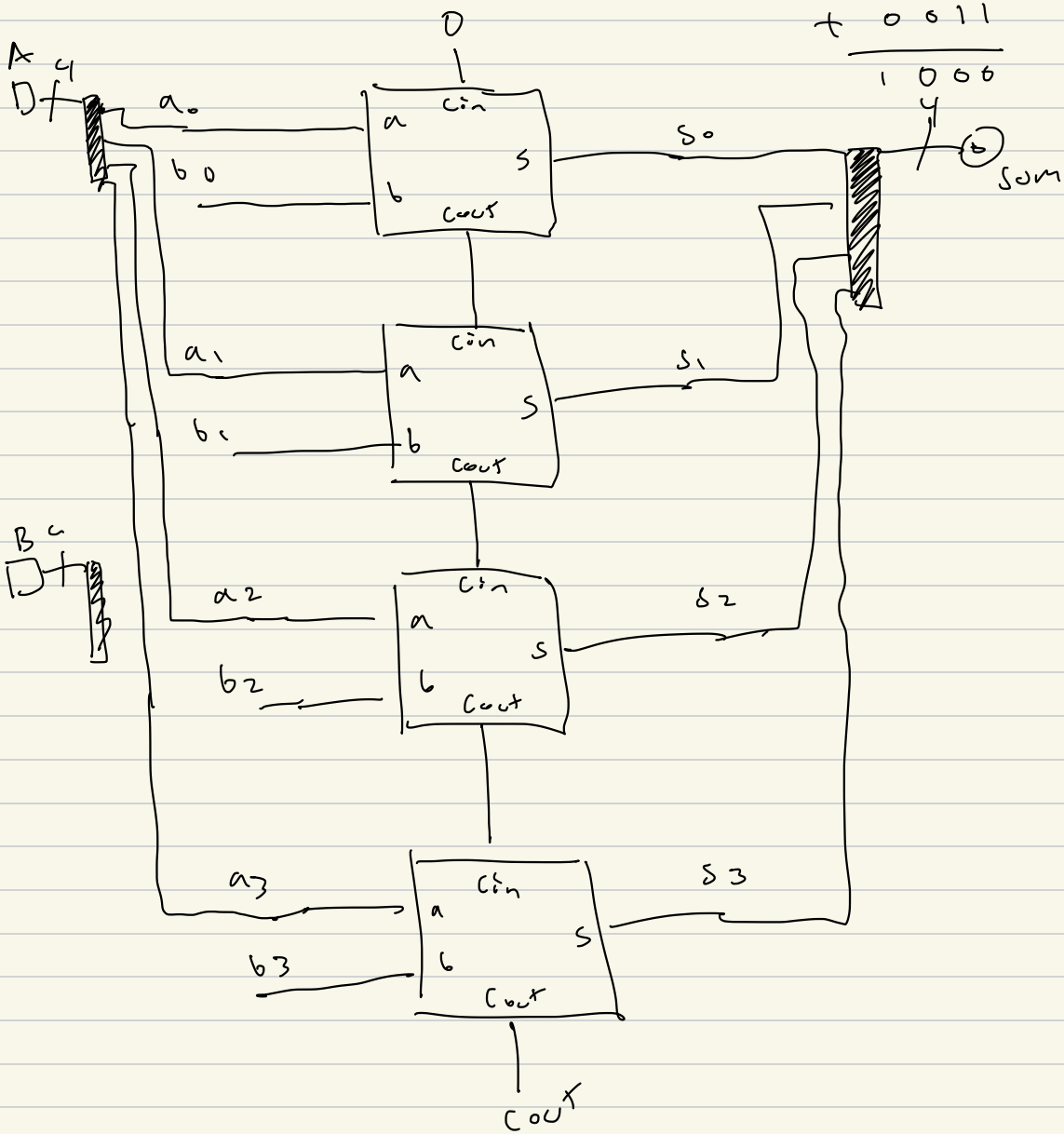
$cout = \ldots$


a
b
Cin
Sum

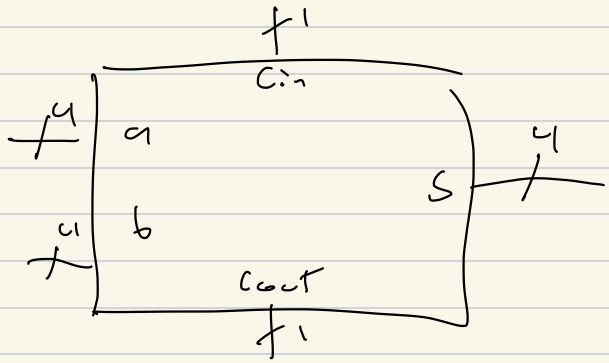
4 bit adder - 4 bit Ripple Carry Adder

a $a_3 a_2 a_1 a_0$ b $b_3 b_2 b_1 b_0$

```
  1 1 1
  0 1 0 1
+ 0 0 1 1
---------
1 0 0 0
```



4 bit ripple carry adder



Goal: N-bit register

state

SR Latch