CS 315-02 Combinational Logic

Midterm

Autograder

Digital - splitter, subcircuits

Final

If you do better

$$\text{midterm}_{new} = \frac{\text{final} + \text{midterm}_{old}}{2}$$

70 midterm 90 final

$$\frac{70 + 90}{2} = \boxed{80} \leftarrow$$

0x FFAABBCC >> 4

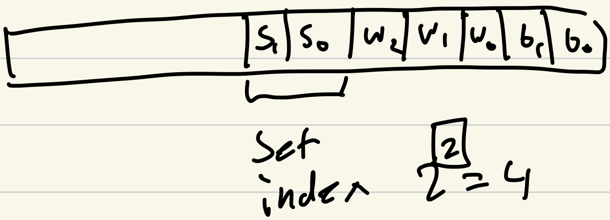
4 bits

= 0xFFAABBC

>> 16

0xFFAA & 0xFF = $\boxed{0xAA}$

0x 0000 FFAA



int sumarr_s (int arr[], int len);

prototype

Autograder

mac
- clone tests
- clone autograder
- install pip3
- pip3 install -r requirements.txt
- install java
  - brew install java11
- put grade on your PATH
- edit ~/.config/grade/config.toml

Windows

WSL Ubuntu

apt install python3-pip3

clone ~~tests~~ tests

clone autograder

put grade on PATH

config autograder

install java

sudo apt install java

wget Digital.zip

In WSL

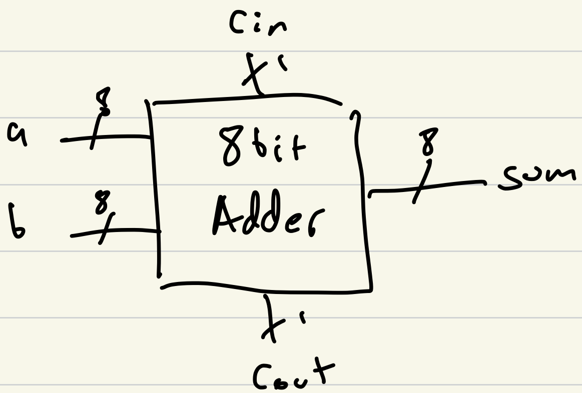
/mnt/c/Users/name/....

Install git-bash in Git Window

setup .ssh/config

keys

Ripple-carry Adder



110 even
111 odd
100 odd

sum-of-products

3-bit number $n_2 n_1 n_0$ (bits)

Two 1-bit outputs: even, odd

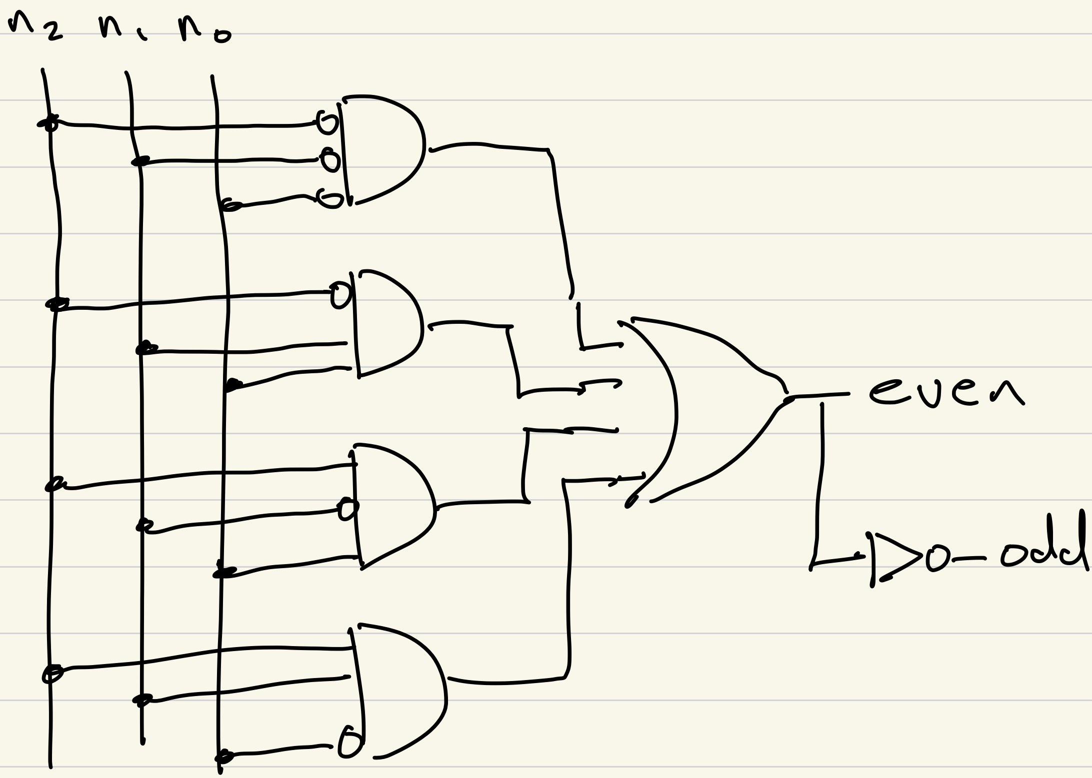
Goal: determine if the number of "1" bits is even or odd

	n_2	n_1	n_0	even	odd
①	0	0	0	1	0
	0	0	1	0	1
	0	1	0	0	1
②	0	1	1	1	0
	1	0	0	0	1
③	1	0	1	1	0
④	1	1	0	1	0
	1	1	1	0	1

0 000
1 001
2 010
3 011

$$even = (\bar{n}_2 \cdot \bar{n}_1 \cdot \bar{n}_0) + (\bar{n}_2 \cdot n_1 \cdot n_0) + (n_2 \cdot \bar{n}_1 \cdot n_0) + (n_2 \cdot n_1 \cdot \bar{n}_0)$$

$$odd = \overline{even}$$



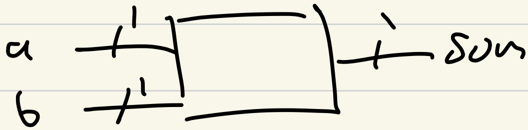
Lab04 part 2 max2

a_1	a_0	b_1	b_0	r_1	r_0
⋮					
1	0	0	1	1	0
⋮					

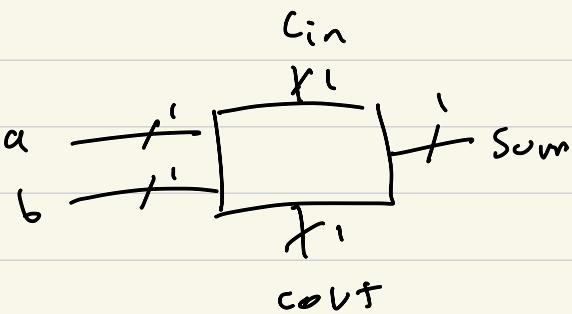
$r1 = $ __________

$r_0 = $ __________

1 bit full adder

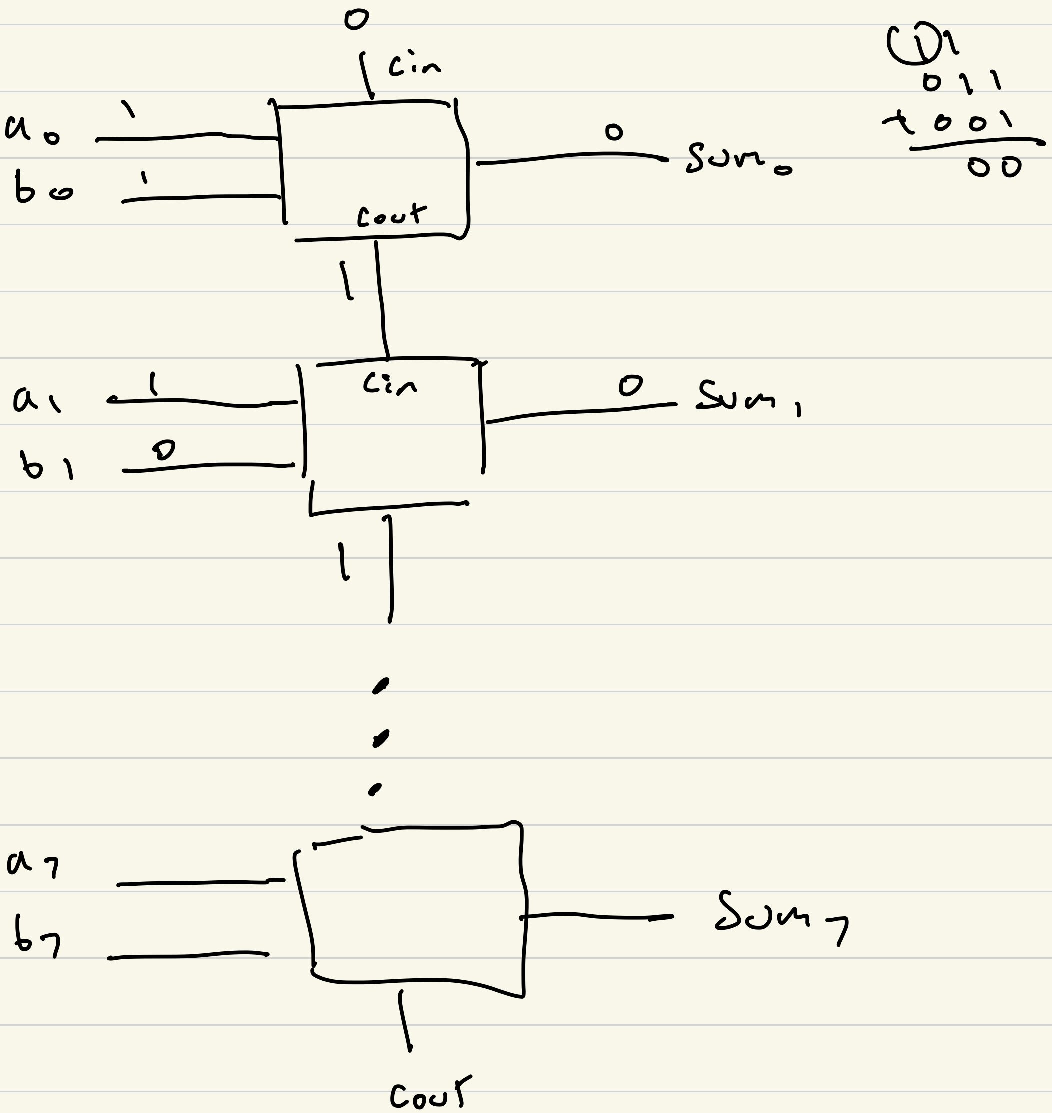


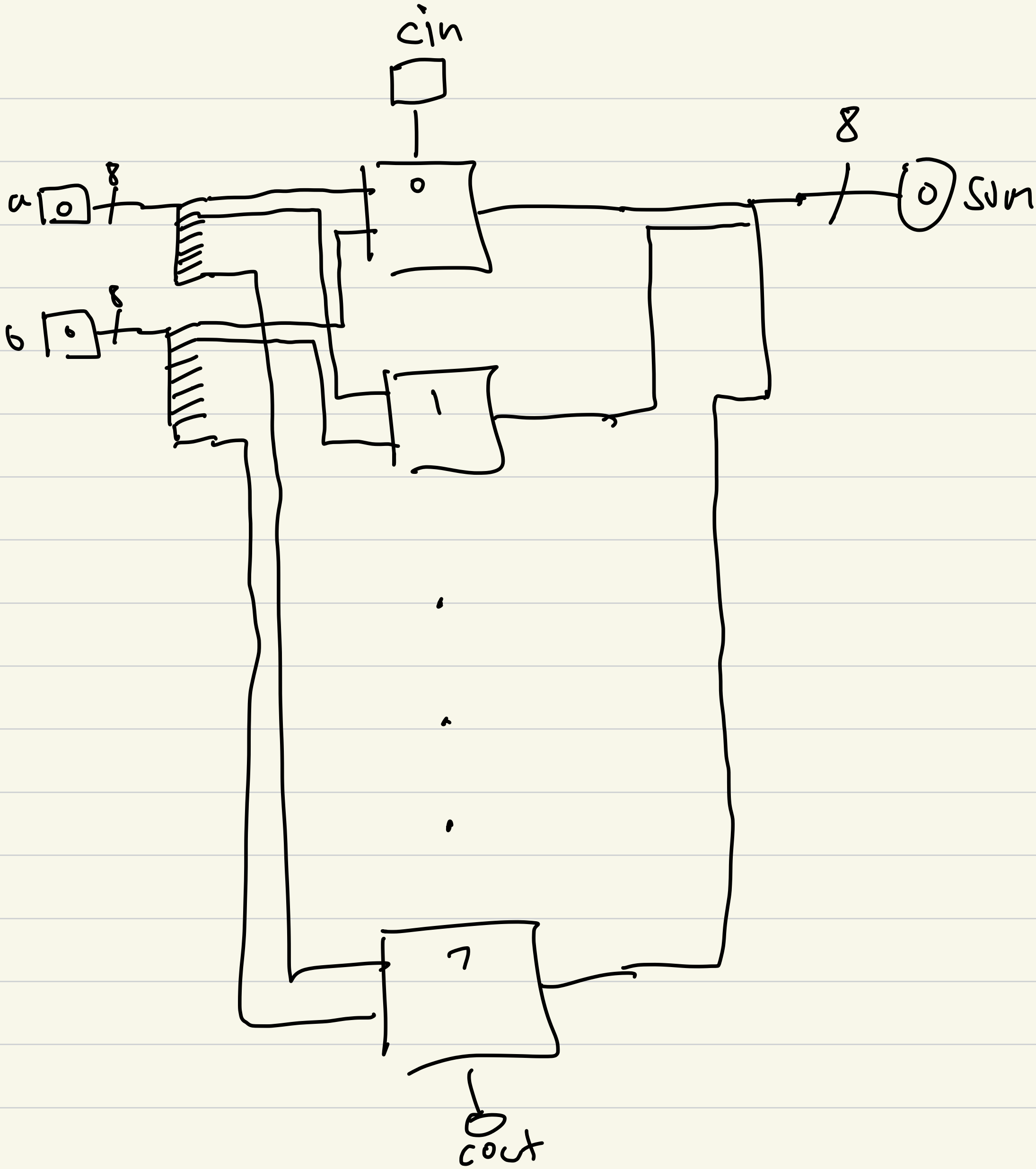
half adder



full adder

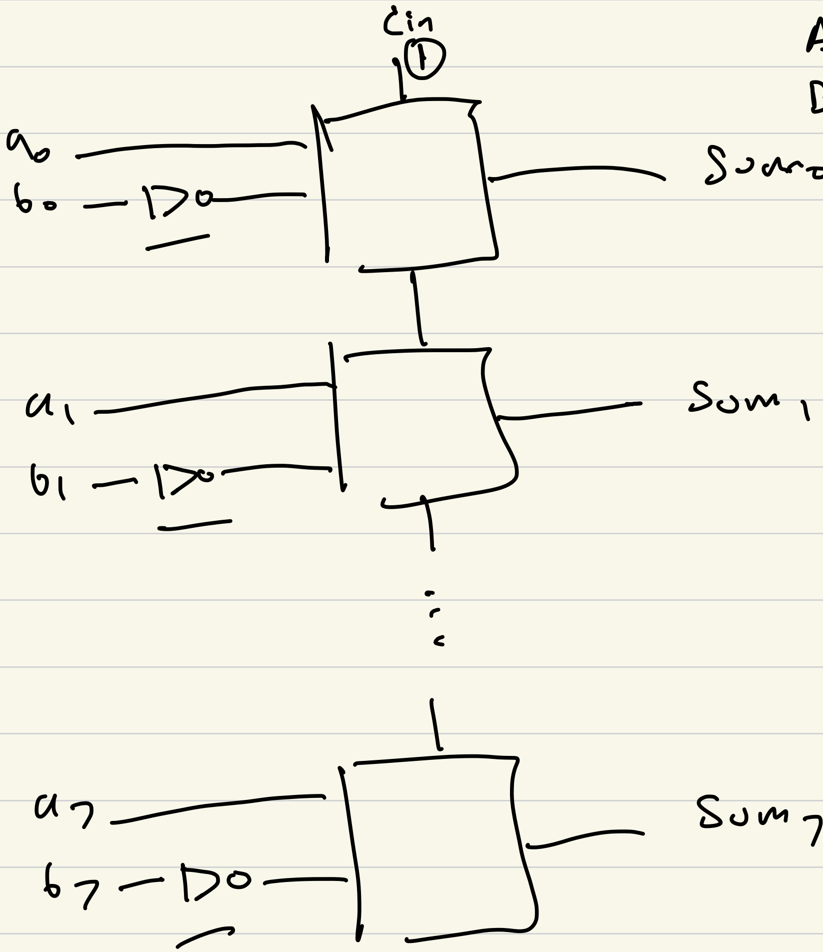
8 bit ripple carry adder



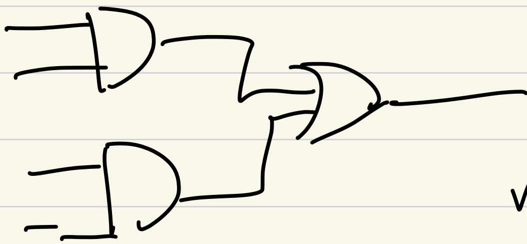

cin
a
8
0
b
8
0
0
1
7
8
0
sum
cout

Subtraction: A - B invert add 1

A 0110
B ~~0010~~
1101
+ 1



Combinational Logic

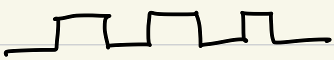


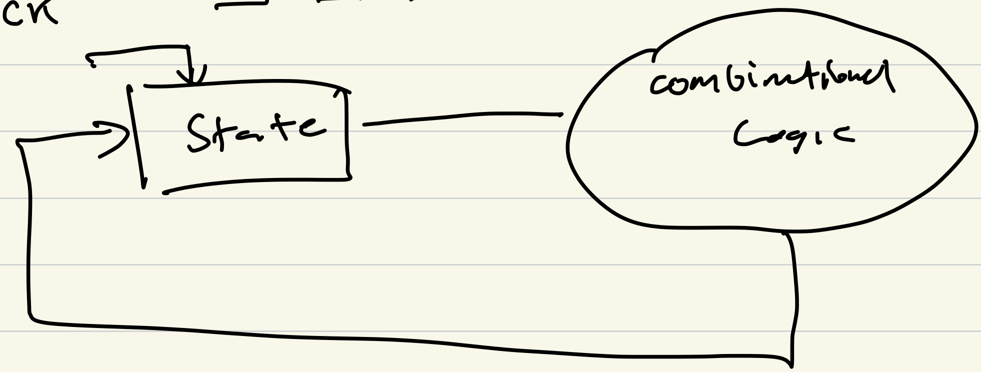
no cycles
DAG

Sequential Logic

State

CLOCK





CPU