

TFE4152 - Lecture 20

Adders

Source

Week	Book	Monday	Book	Friday
34		Introduction, what are we going to do WH 1 , WH 15 in this course. Why do you need it?		Manufacturing of integrated circuits
35	CJM 1.1	pn Junctions	CJM 1.2 WH 1.3, 2.1-2.4	Mosfet transistors
36	CJM 1.2 WH 1.3, 2.1-2.4	Mosfet transistors	CJM 1.3 - 1.6	Modeling and passive devices
37		Guest Lecture - Sony	CJM 3.1, 3.5, 3.6	Current mirrors
38	CJM 3.2, 3.3,3.4 3.7	Amplifiers	CJM, CJM 2 WH 1.5	SPICE simulation
39		Verilog		Verilog
40	WH 1.4 WH 2.5	CMOS Logic	WH 3	Speed
41	WH 4	Power	WH 5	Wires
42	WH 6	Scaling Reliability and Variability	WH 8	Gates
43	WH 9	Sequencing and FSM	WH 10	Datapaths - Adders
44	WH 10	Datapaths - Multipliers, Counters	WH 11	Hacking circuits (Nordic). Memories
45	WH 12	Packaging	WH 14	Test
46		Guest lecture - Nordic Semiconductor		
47	CJM	Recap of CJM	WH	Recap of WH

Housekeeping

<https://folk.ntnu.no/carstenw/TFE4152/scoreboard.html>

[https://github.com/wulffern/dic2021/blob/main/
2021-10-19_project_report/project_report.pdf](https://github.com/wulffern/dic2021/blob/main/2021-10-19_project_report/project_report.pdf)

Want to make your own IC?

<https://efabless.com/>

<https://github.com/google/skywater-pdk>

<https://skywater-pdk.readthedocs.io/en/latest/index.html>

Honesty: Making integrated circuits is an expensive endeavor. As humans, we will always make mistakes, or fail to imagine how things will fail. The cost of failures increases exponentially the closer we are to a volume product. As such, finding mistakes early is necessary to reduce cost. If people are honest, and acknowledge immediately that something has gone wrong, that can give us time to fix the problem before it costs millions.

Responsibility: Time to market is essential in integrated circuits. We must meet the market window for a particular product. Complex projects will be broken down into pieces, and milestones. We require those milestones to be met, as such, we need people that take responsibility, and ensure that milestones are met. If their milestone depends on a delivery from someone else, they need to follow up, and help to ensure that the overall milestones are met.

Logical thinking: When things go wrong, we need people that will dig into the problem and find the real physical root cause. There is no place for “hand-wavy” arguments, or arguments from authority “I know everything, so you should listen to me”. We deal with physics, and in integrated circuits, how something fails always have a physical reason.

Humility: Realizing that humans are fallible, but we should strive for not doing the same mistake twice.

Diversity: An employee’s brain must have the right skills and the right qualities. We strive for diversity, free from discrimination due to gender, nationality, origin, and religion.

Ability to learn: As an engineer you’re never finished learning. Integrated circuits is a deep skill, and you will need the ability to learn new things.

Curiosity: We want engineers that like to understand how things work. There should be an inherent drive to figure things out.

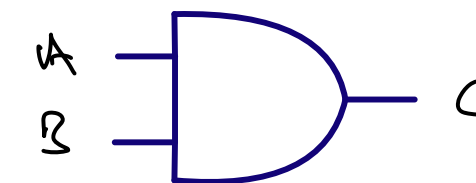
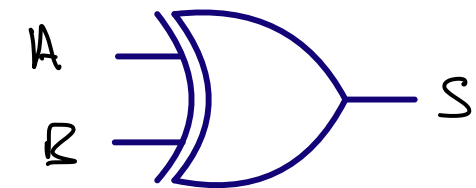
Be fearless: Don’t be afraid of asking questions. Be forward leaning. Act. Champion your ideas.



Together with a partner (or alone),
figure out the logic circuit for the
sum of two 1-bit signals, $A + B$, with
the sum as a 1-bit output (S) and a
1-bit carry (C)

Half adder

A	B	C	S _H
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



$$S = \bar{A}B + \bar{B}A$$

$$C = A \cdot B$$

Together with a partner (or alone),
figure out the logic circuit for the
**sum of three 1-bit signals, A , B ,
and carry input (C_i)** with the sum
as a 1-bit output (S) and a 1-bit
carry (C)

Full adder

A	B	S _K	C _{in}	C _{out}	S _F
0	0	0	0	0	0
0	0	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	1

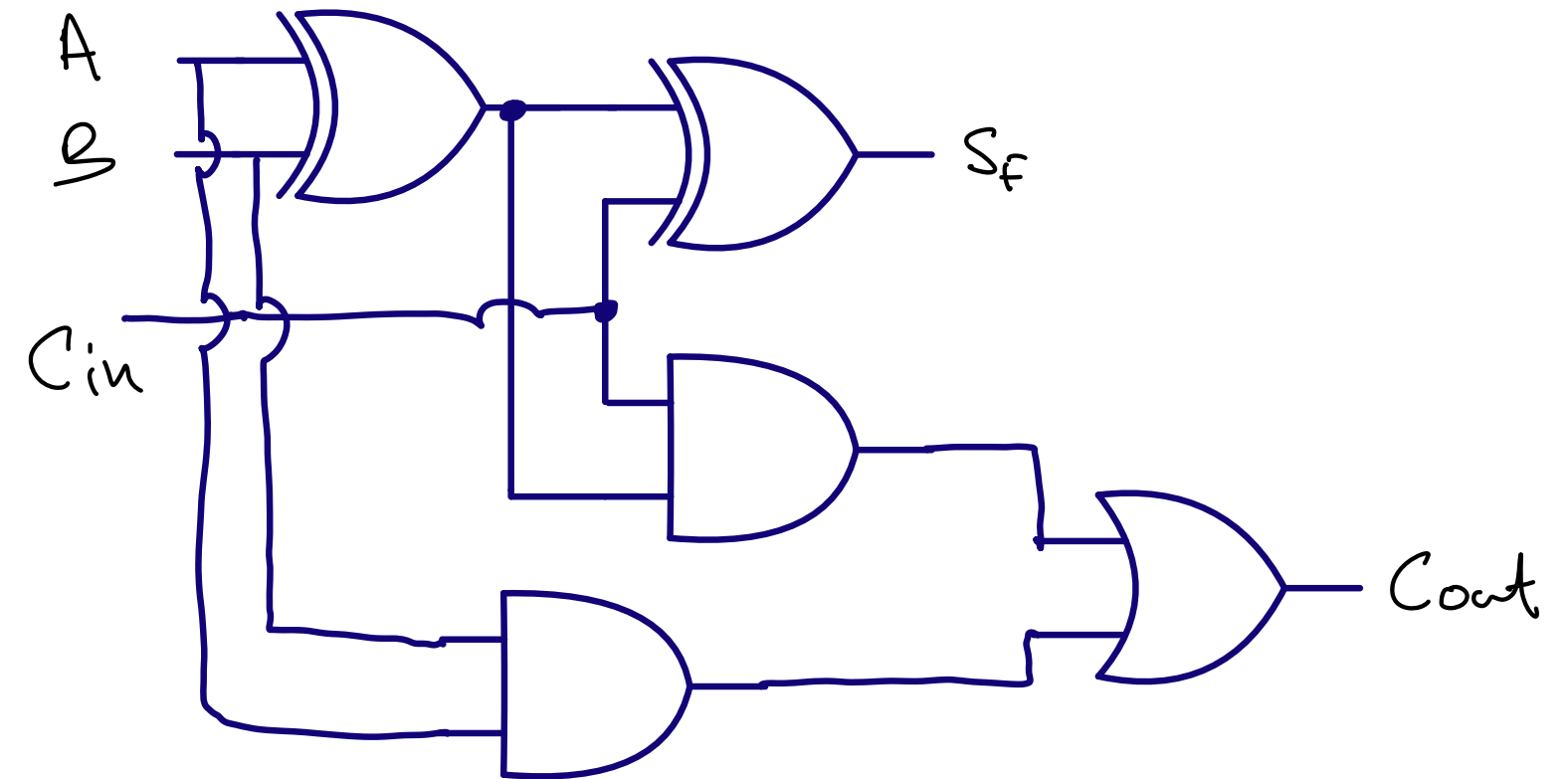
$$\begin{aligned}
 S &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} \\
 &\quad + ABC \\
 &= (\bar{A}B + \bar{B}A)\bar{C} + C(\bar{A}\bar{B} + AB) \\
 &\quad \text{XOR} \\
 &\quad \text{XOR} = D\bar{E} + E\bar{D} \\
 S &= A \oplus B \oplus C
 \end{aligned}$$

$$\begin{aligned}
 C_{out} &= \bar{A}B C + A\bar{B}C + AB\bar{C} \\
 &\quad + ABC \\
 &= (\bar{A}B + A\bar{B})C + AB(C + \bar{C}) \\
 &= (A \oplus B)C + AB \Rightarrow \text{Majority gate}
 \end{aligned}$$

Full adder

$$S = A \oplus B \oplus C_i$$

$$C = (A \oplus B)C_i + AB$$



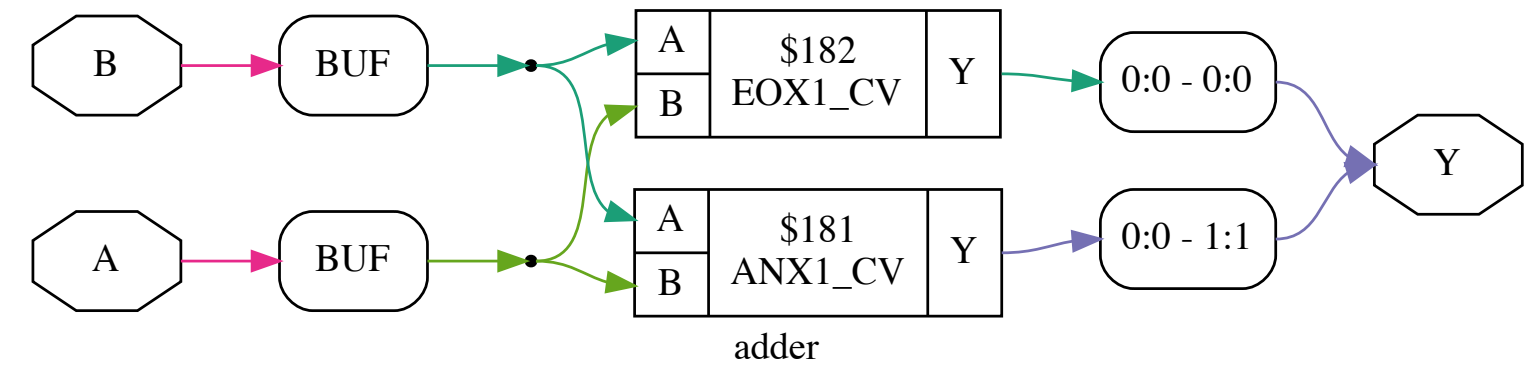
1-bit adder

```
module adder (input logic [WIDTH-1:0] A,  
              input logic [WIDTH-1:0] B,  
              output logic [WIDTH:0] Y  
              );
```

```
    localparam WIDTH=1;
```

```
    assign Y = A + B;
```

```
endmodule
```



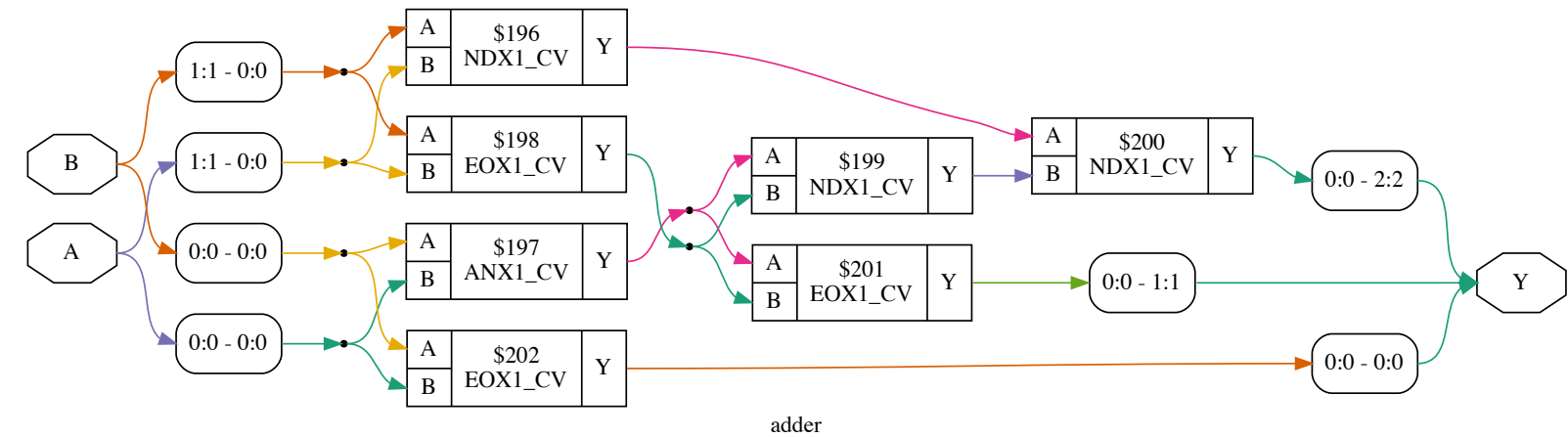
2-bit adder

```
module adder (input logic [WIDTH-1:0] A,
              input logic [WIDTH-1:0] B,
              output logic [WIDTH:0] Y
              );
```

```
    localparam WIDTH=2;
```

```
    assign Y = A + B;
```

```
endmodule
```



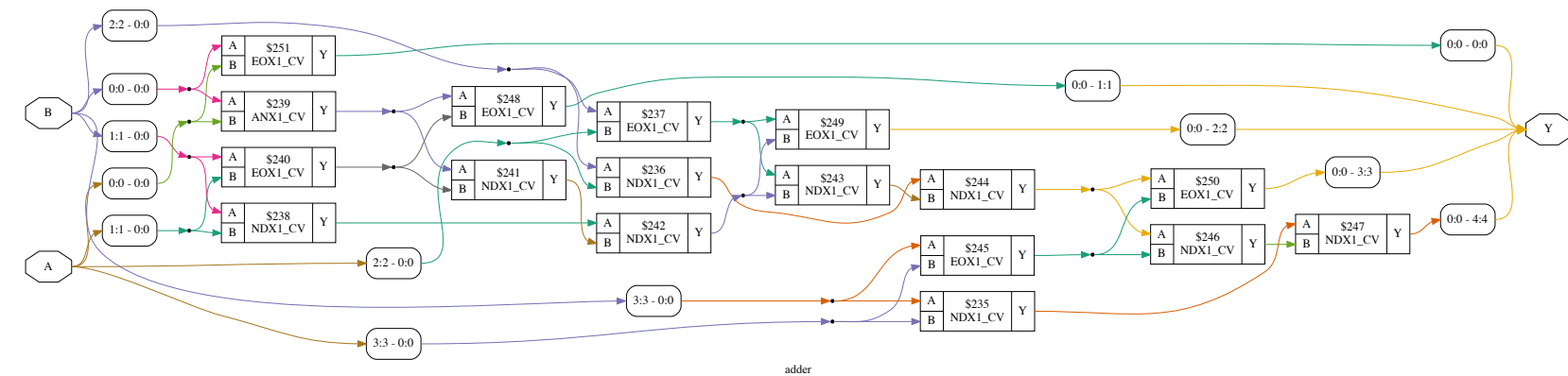
4-bit adder

```
module adder (input logic [WIDTH-1:0] A,  
             input logic [WIDTH-1:0] B,  
             output logic [WIDTH:0] Y  
             );
```

```
    localparam WIDTH=4;
```

```
    assign Y = A + B;
```

```
endmodule
```

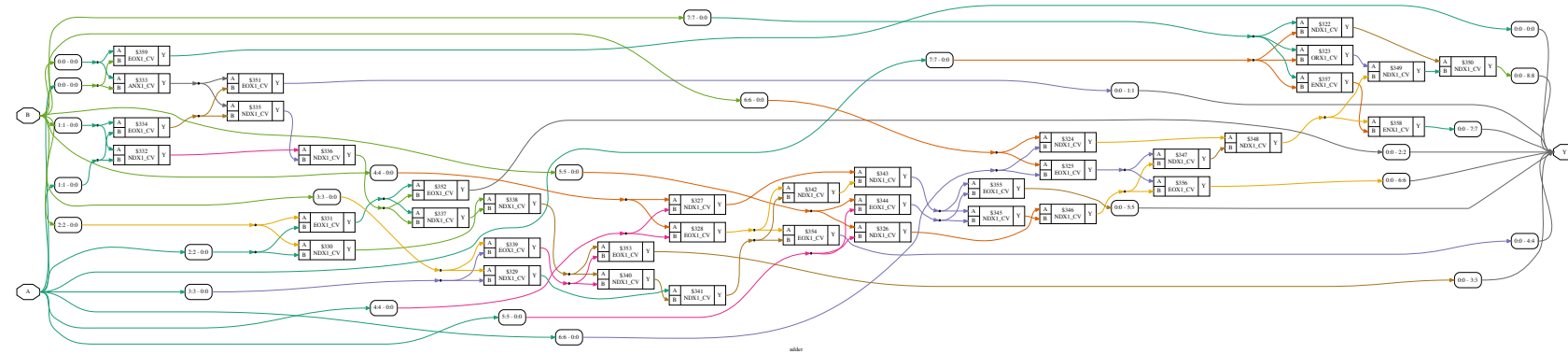


8-bit adder

```
module adder (input logic [WIDTH-1:0] A,  
             input logic [WIDTH-1:0] B,  
             output logic [WIDTH:0] Y  
             );
```

```
    localparam WIDTH=8;
```

```
    assign Y = A + B;  
endmodule
```

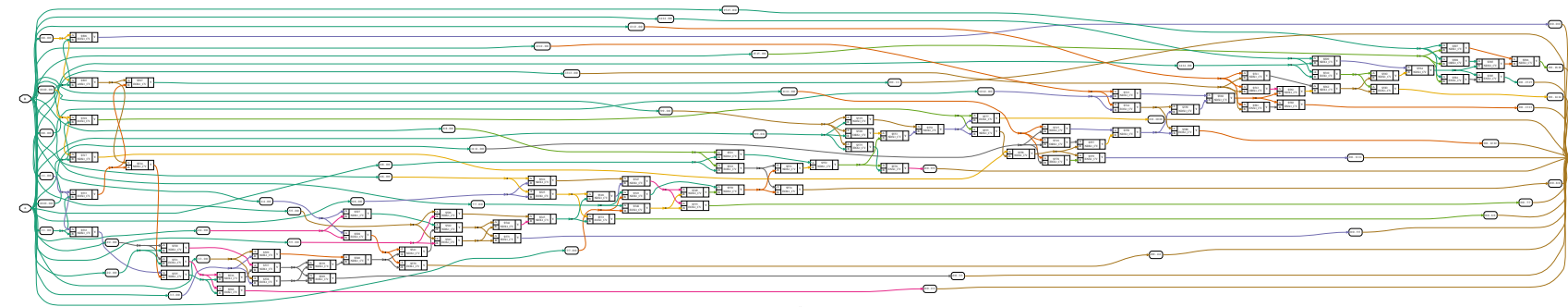


16-bit adder

```
module adder (input logic [WIDTH-1:0] A,  
              input logic [WIDTH-1:0] B,  
              output logic [WIDTH:0] Y  
              );
```

```
    localparam WIDTH=16;
```

```
    assign Y = A + B;  
endmodule
```



Straightforward adders become deep, can we make it better?

