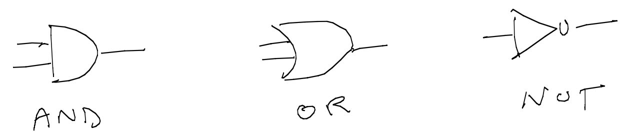
lecture 4

Combinational logic

Last lecture

- truth tables
  - $Y = f(A, B, \ldots)$
  - sums of products
- logical gates & circuits



Arithmetic Circuits

$$
\begin{array}{rcccc}
 & C_{n-1} \cdots & C_2 & C_1 & C_0 \\
 & A_{n-1} \cdots & A_2 & A_1 & A_0 \\
+ & B_{n-1} \cdots & B_2 & B_1 & B_0 \\
\hline
 & S_{n-1} \cdots & S_2 & S_1 & S_0
\end{array}
$$

Note
- $C_0 = 0$
- A, B could be positive or negative

Sum of products? $\Rightarrow$ fast ☺

$B_{31} A_{31} \ldots B_1 A_1 B_0 A_0$	$S_{31} \ldots S_2 S_1 S_0$
$2^{32 \cdot 2}$ rows	

$\sim 2^{2 \times 32}$ gates $\Rightarrow$ big ☹

$$
\begin{array}{rccc}
C_{n-1} \cdots & C_2 & C_1 & \\
A_{n-1} \cdots & A_2 & A_1 & A_0 \\
B_{n-1} \cdots & B_2 & B_1 & B_0 \\
\hline
S_{n-1} \cdots & S_2 & S_1 & S_0
\end{array}
$$

A_0	B_0	S_0	C_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

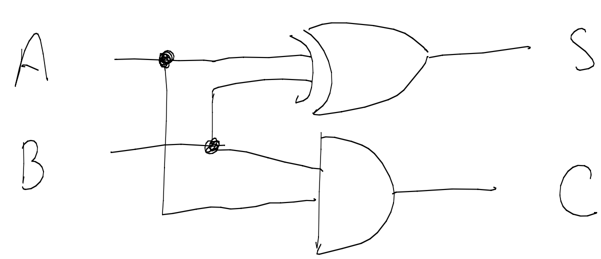
$$S_0 = A_0 \oplus B_0$$

$$C_1 = A_0 \cdot B_0$$

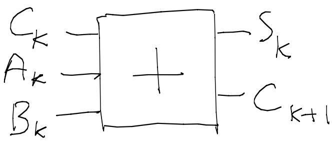
Half Adder

$$S = A \oplus B$$

$$C = A \cdot B$$



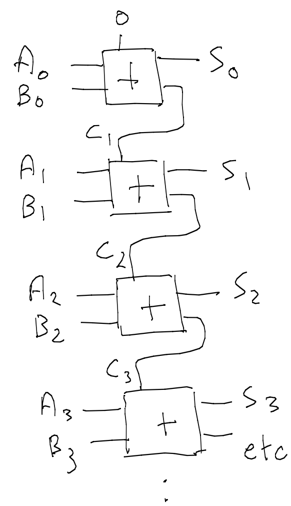
$$
\begin{array}{r}
C_{n-1} \cdots C_2 C_1 C_0 \\
A_{n-1} \cdots A_2 A_1 A_0 \\
B_{n-1} \cdots B_2 B_1 B_0 \\
\hline
S_{n-1} \cdots S_2 S_1 S_0
\end{array}
$$



Full adder

A_k	B_k	C_k	S_k	C_{k+1}
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

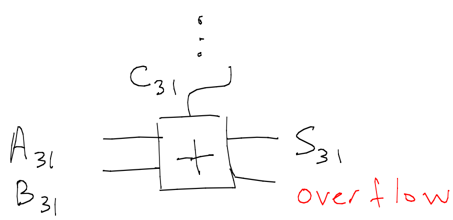
Ripple Adder



If $n = 32$, we can have a long delay as carries propagate through the circuit.

We'll return to this later.

Overflow



A_{31}	B_{31}	S_{31}	overflow

See Exercises 2

Encoder

$A_m \ldots A_2\ A_1\ A_0$	$Y_n \ldots Y_0$

events (many bits)

code word (few bits)

Encoder: Example 1

panel with five buttons

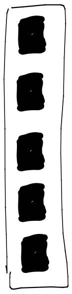
b_4	b_3	b_2	b_1	b_0	Y_2	Y_1	Y_0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	1
1	0	0	0	0	1	0	0

Suppose only one pressed at any time

Q: What if two buttons are pressed at same time?

A: none of lights go on. (assuming a sum-of-products implementation of each L_i)

panel with five buttons

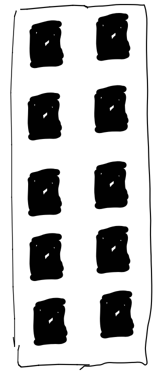


b_4	b_3	b_2	b_1	b_0	y_2	y_1	y_0
0	0	0	0	0	0	0	0
X	X	X	X	1	0	0	1
X	X	X	1	0	0	1	0
X	X	1	0	0	0	1	1
X	1	0	0	0	1	0	0
1	0	0	0	0	1	0	1

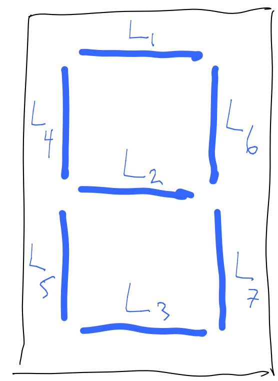
Allow any number of buttons to be pressed.

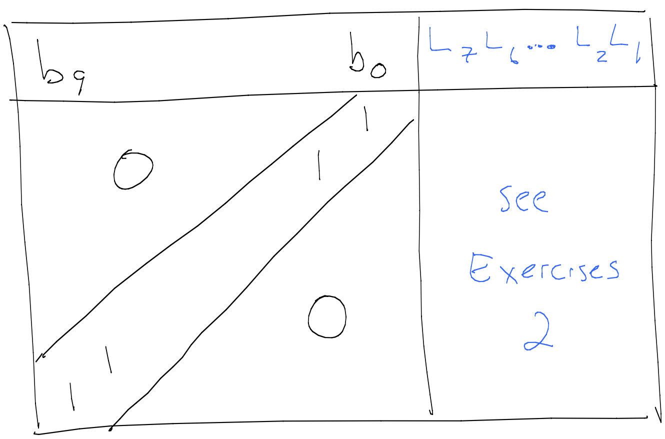
Encoder: Example 2

panel with ten buttons

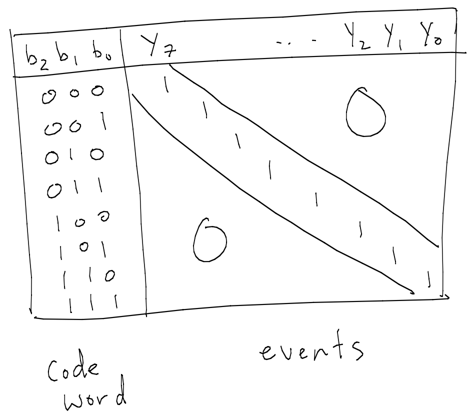


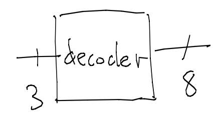
$b_9 \ldots b_0$

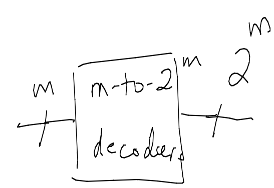




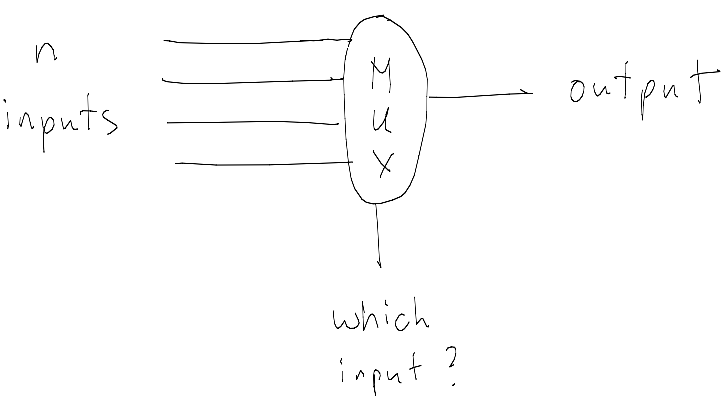
Decoder





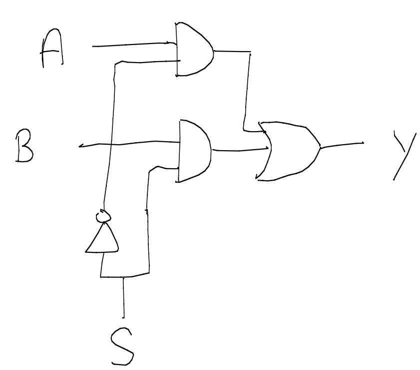


Multiplexor (also called selector)



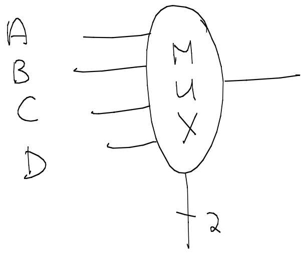
1-bit multiplexor

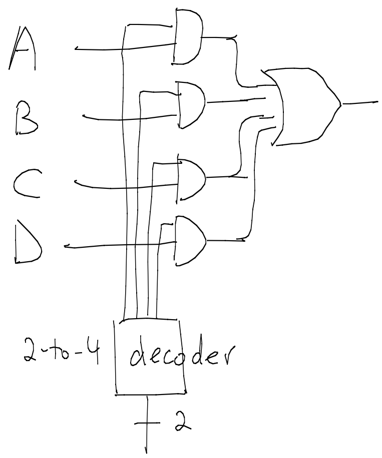
$$Y = \overline{S} \cdot A + S \cdot B$$



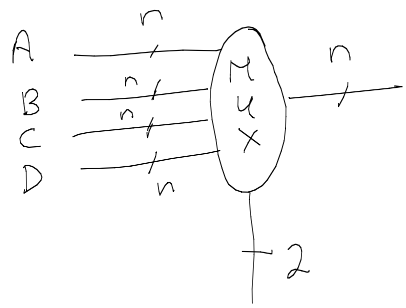
2-bit multiplexor

(What is the circuit?)





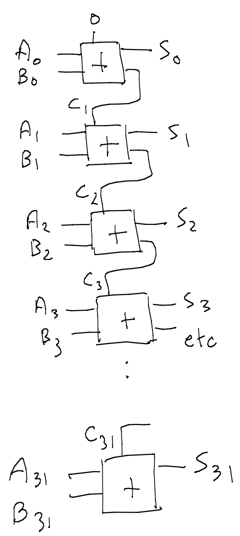
2-bit multiplexor



Selects from four n-bit inputs. For each $A_i B_i C_i D_i$, replicate circuit on previous slide (but use same decoder).

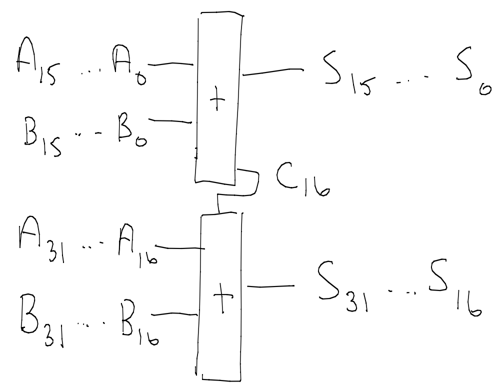
Ripple Adder

Recall arithmetic circuit



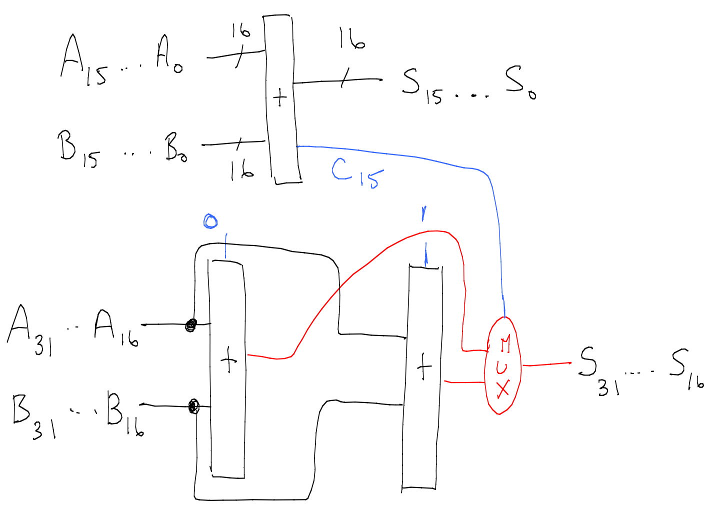
Slow!

How to speed up the adder?



Does this speed it up? No.

"Conditional Sum" Adder



Many adders have been proposed:

small, slow

- ripple

  ⋮

- conditional sum (previous slide shows basic idea only)

  ⋮

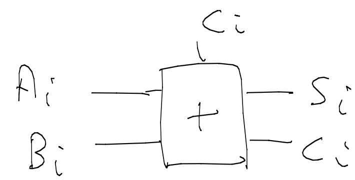
- sum of products

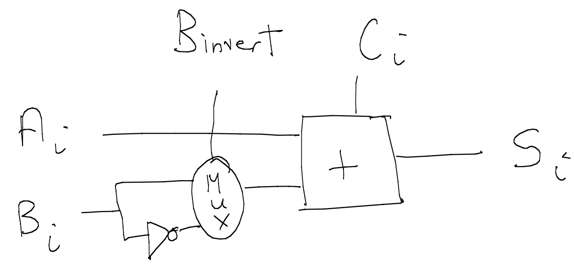
big, fast

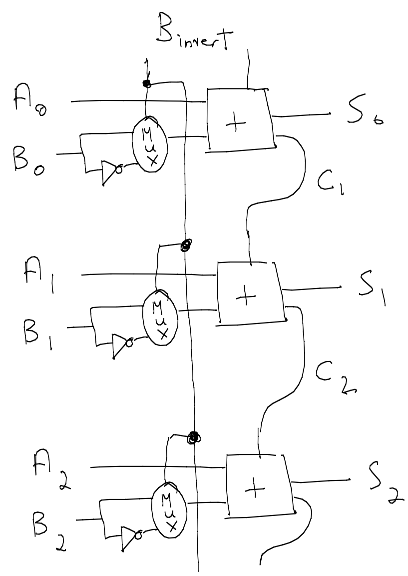
Subtraction

$$
\begin{array}{rl}
 & A_{n-1} \cdots A_2\, A_1\, A_0 \\
- & B_{n-1} \cdots B_2\, B_1\, B_0 \\
\hline
 & S_{n-1} \cdots S_2\, S_1\, S_0
\end{array}
$$

$$x - y = x + \underset{\substack{\uparrow \\ \text{invert bits and add 1}}}{(-y)}$$

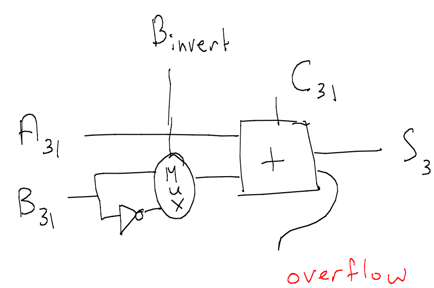






$$
\begin{array}{l}
A_{n-1} \cdots A_2\, A_1\, A_0 \\
B_{n-1} \cdots B_2\, B_1\, B_0 \\
\hline
S_{n-1} \cdots S_2\, S_1\, S_0
\end{array}
$$

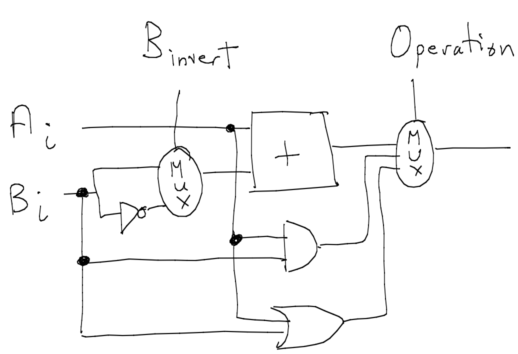
$n = 32$



B_{invert}	A_{31}	B_{31}	S_{31}	overflow
0				
1				

See Exercises 2

More operators:
bit-wise AND, OR



the fast adder implementation
to the adder part

Arithmetic Logic Unit (ALU)