Durham University

Digital Electronics

Circuits

Dr. Eleni Akrida
eleni.akrida@durham.ac.uk

Overview of today's lecture

- Adder
- Subtractor
- Decoder
- Mux
- Tristates
- Simple ALU

Key circuits

Combinatorial/combinational circuits:

(Output depends only on current input)

Adders – add the contents of two registers

Decoders – use a binary number to activate a single line

Multiplexors – use a binary number to select an input

Sequential circuits

(Output depends on state and input, i.e. history of input)

Latches / flip-flops – basic memory element

Half-Adder

Based on 8 simple rules:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 with Carry

Carry + 0 + 0 = 1

Carry + 0 + 1 = 0 with Carry

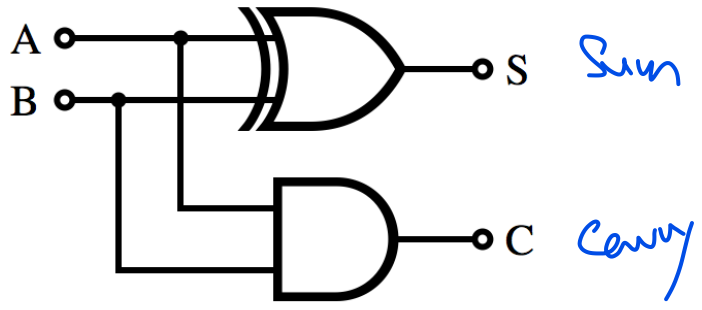
Carry + 1 + 0 = 0 with Carry

Carry + 1 + 1 = 1 with Carry

Inputs: A, B

Outputs: Sum, Carry

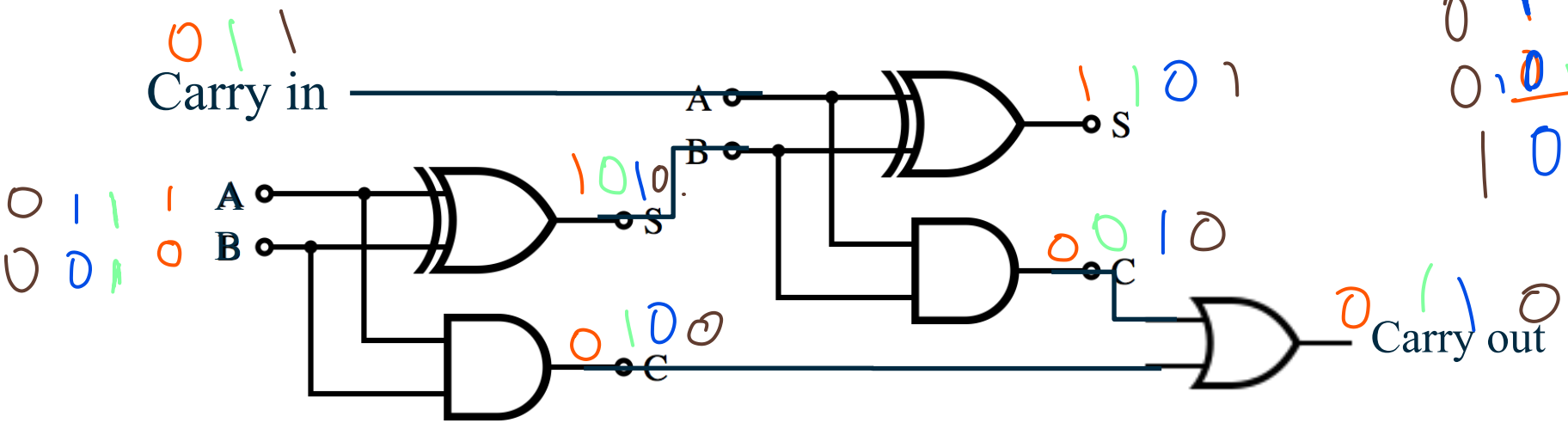
A	B	Sum	Carry
0	0	**0**	**0**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**0**	**1**



Adder

Input is not just A and B, but A, B and **the carry from the previous bit**

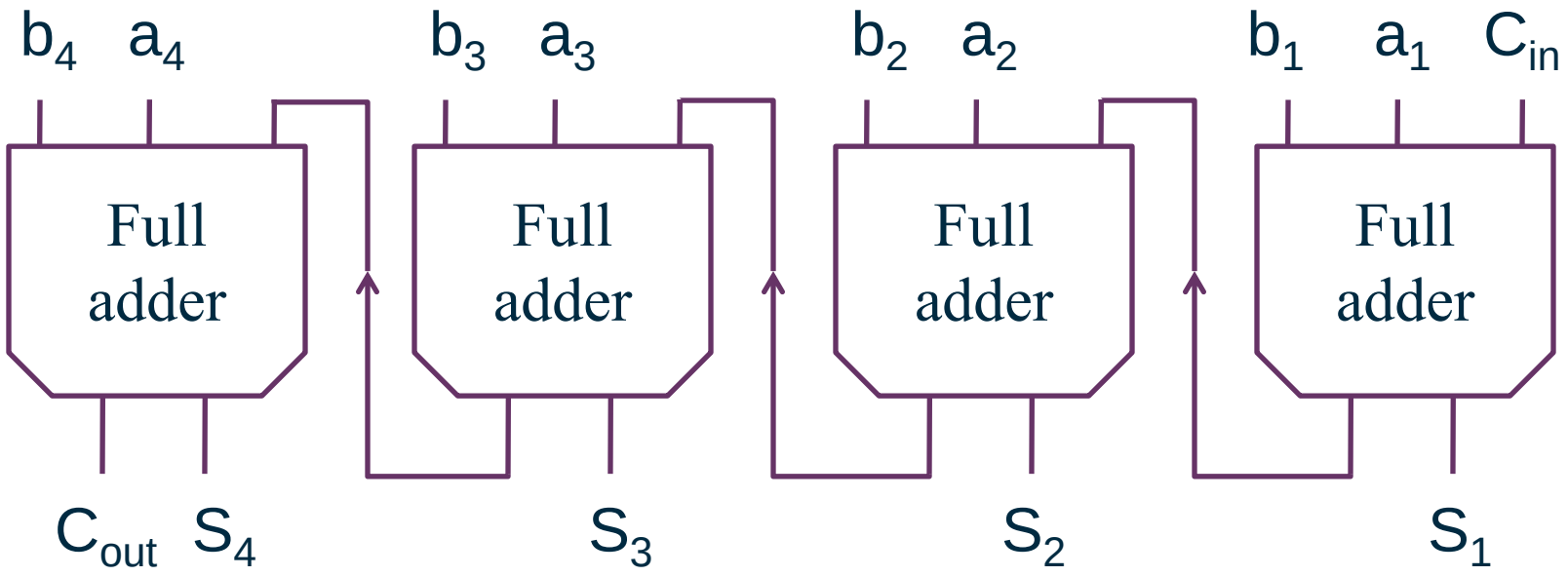
Use two half-adders: Add A and B first, then add in the carried bit:



“Full adder”

Chaining adders

How to add two 4 bit registers:

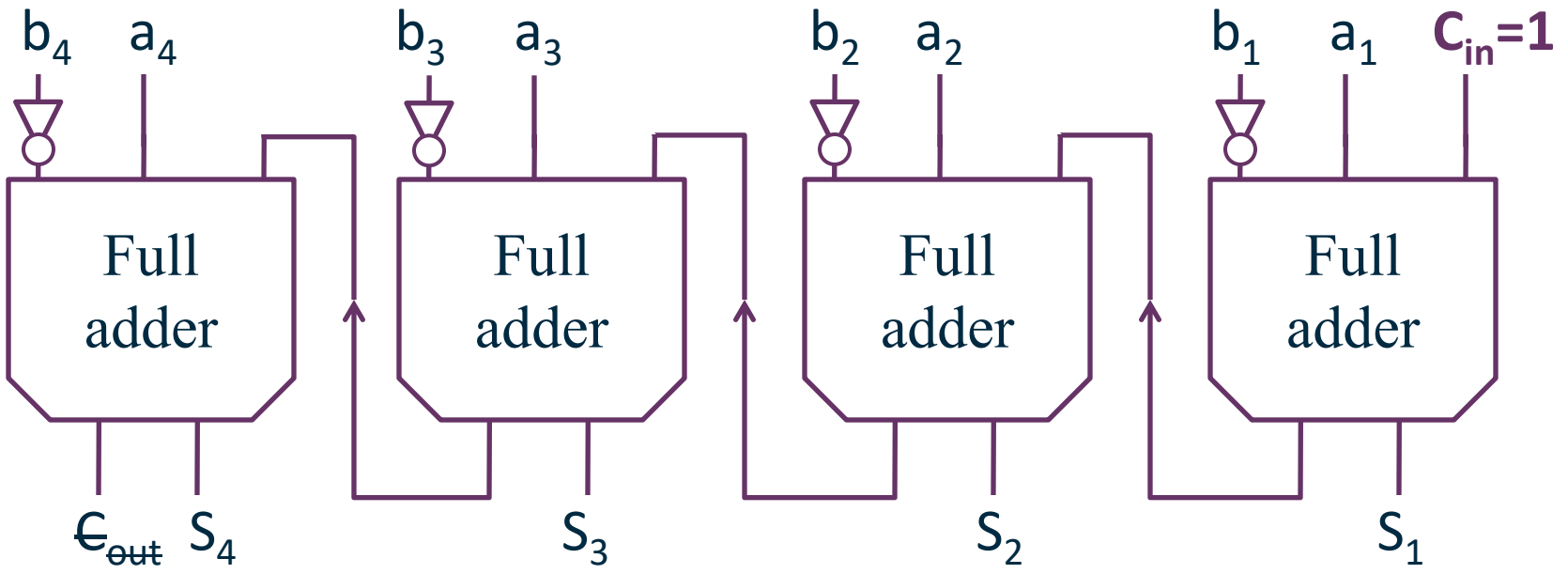


More full-adders can be chained together to give 8-, 16- or 32-bit adders.

Subtractor

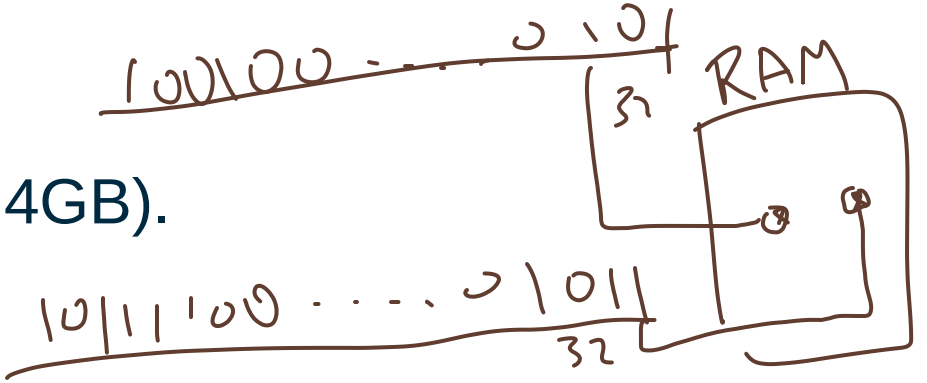
Recall that we can create a **twos-complement negative** by inverting each bit and adding 1.

We subtract b from a by **adding -b**.



The NOT gates and setting C_{in}=1 give the effect of converting b to -b.

Decoder



We have a large bank of memory (say 2^{32} bytes = 4GB).

We want to use 32 bits to address a byte.

How do we get 32 set bits representing a binary number (address), to activate a given byte from the 4 billion possible?

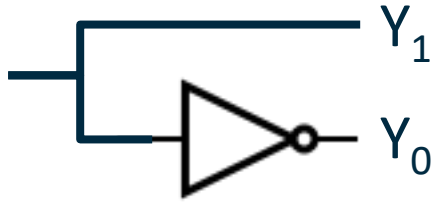
This is what a decoder does:

N inputs. $\mathbf{2^N}$ outputs.

Only one output is high at a time.

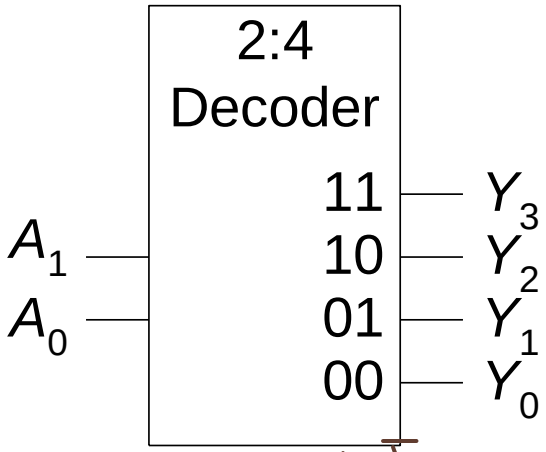
Which one depends on the input.

1-bit decoder: A



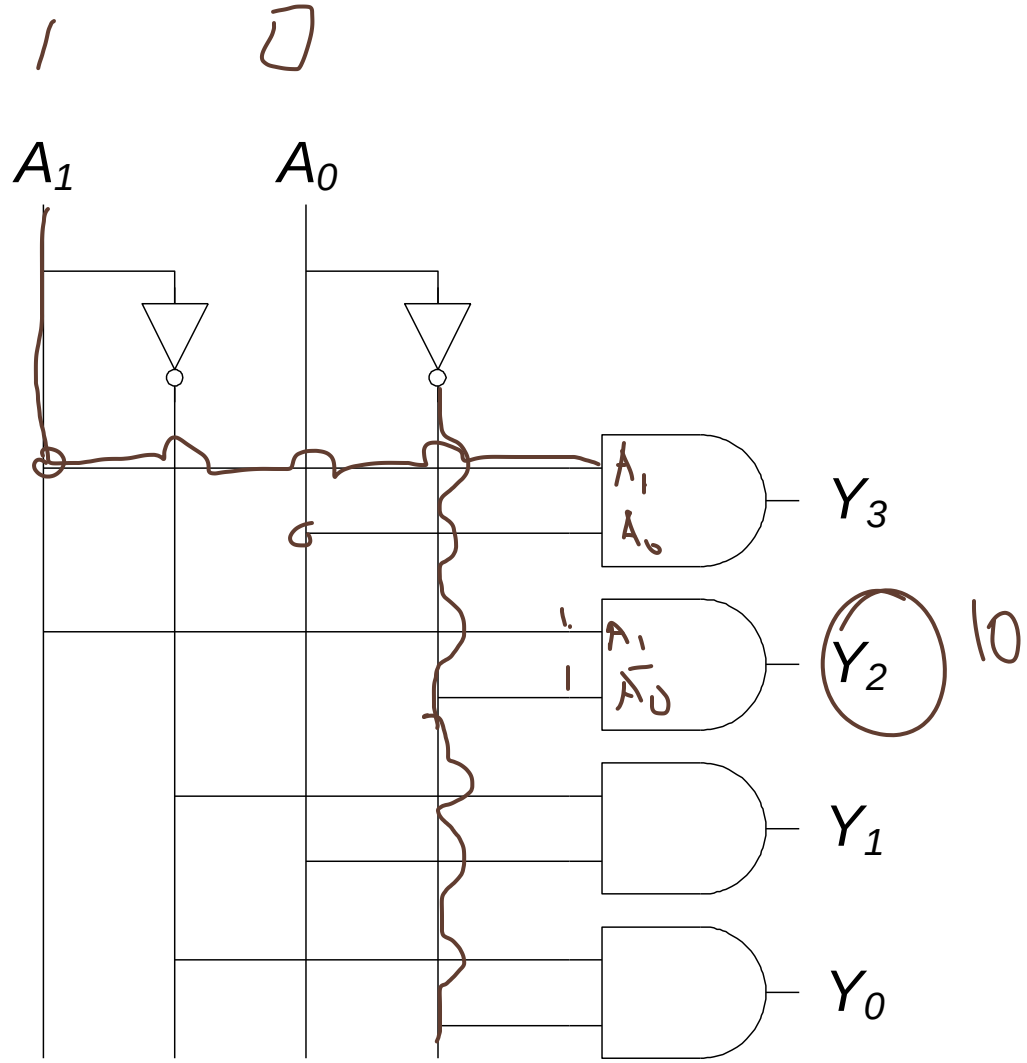
Decoder

2-bit decoder:



Unique 1 in each row

A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

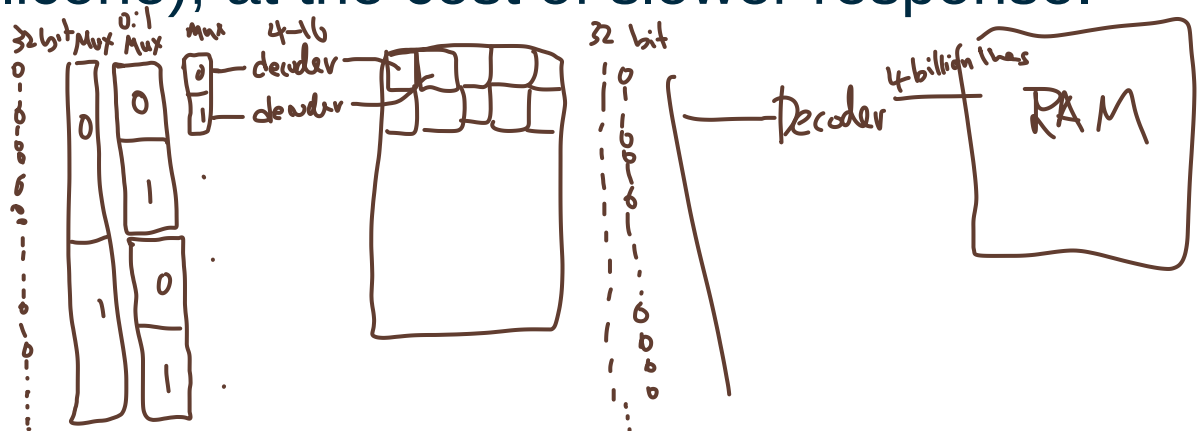


Decoder

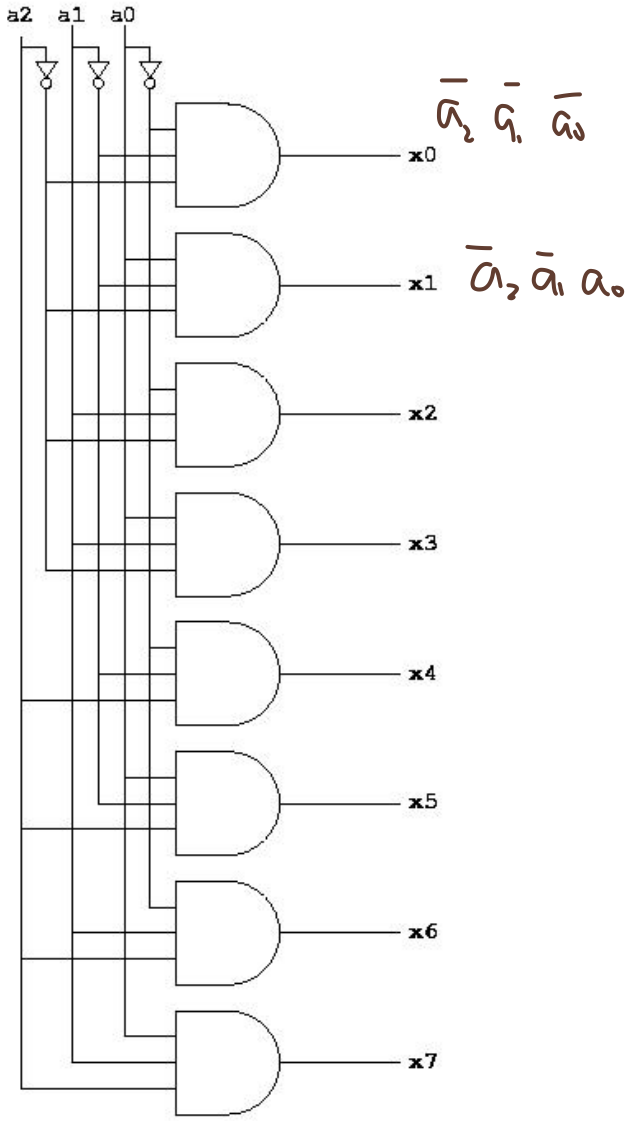
Larger decoders require multi-input AND gates to be constructed in two-level logic.

This requires a lot of circuitry for large decoders.

Can create deeper circuits with fewer transistors (i.e. less silicone), at the cost of slower response.

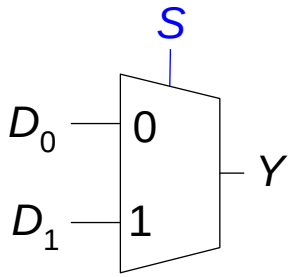


3-bit or '3-8' decoder



Durham University

Mux (multiplexor)

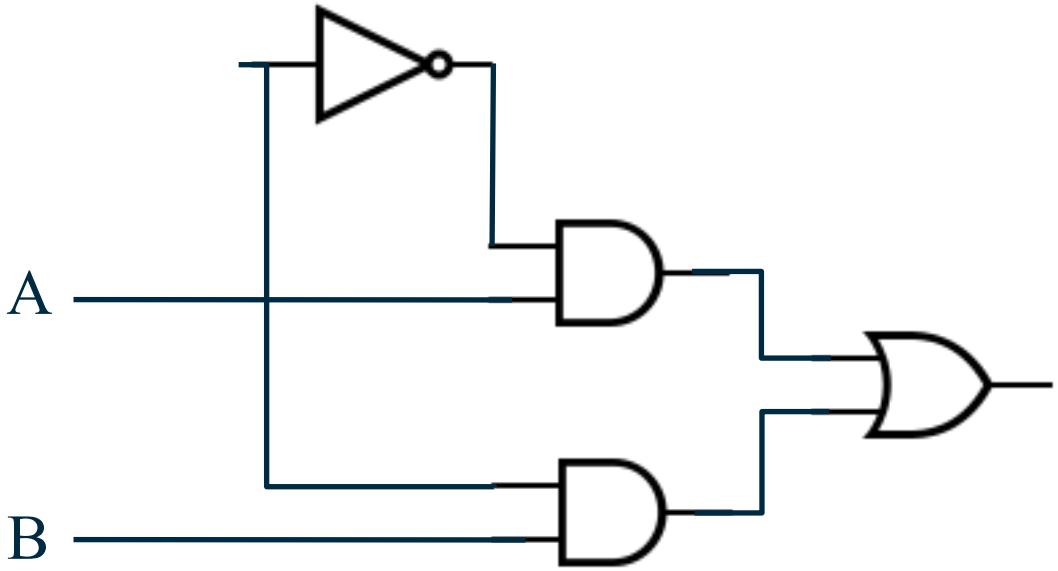


S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Compressed version of the truth table

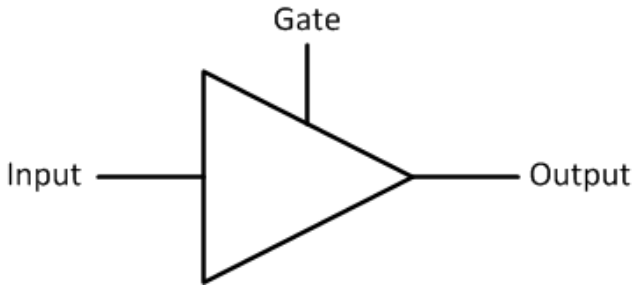
S	Y
0	D_0
1	D_1

2-to-1 line multiplexor



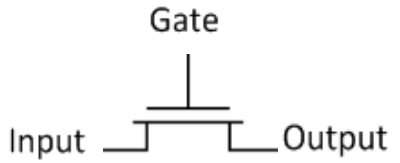
A multiplexor has **k+2^k inputs** and **1 output**.
The first k inputs (selector) represent a binary number.
The output takes the value of the remaining input indexed by this binary number.
Consider many memory locations connected to the inputs. Using the selector we can select which is loaded into a register connected to the output.

Truth tables: Tristate



Input	Enable	Output
0	0	**Floating**
0	1	**0**
1	0	**Floating**
1	1	**1**

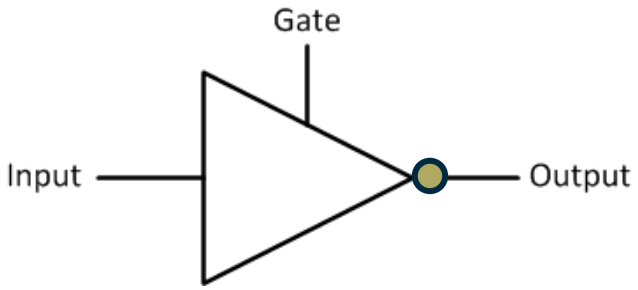
Linear truth table



A single transistor could work, but the output is not driven.

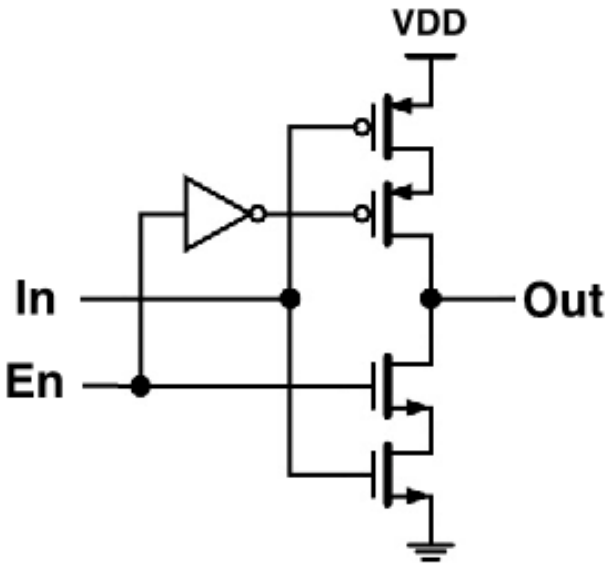
Issue when gate is ON and input is weak – the output will be weak as well

Truth tables: Inverting Tristate



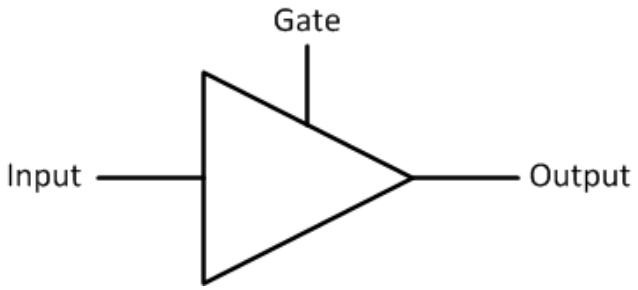
Input	Enable	Output
0	0	**Floating**
0	1	**1**
1	0	**Floating**
1	1	**0**

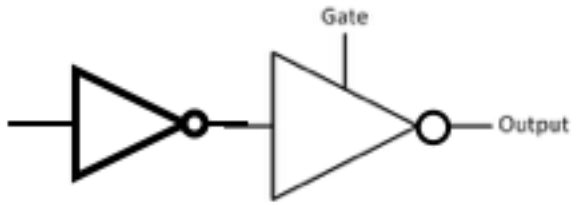
Linear truth table



Here the output is driven by direct connection to VDD or ground.

Truth tables: Tristate

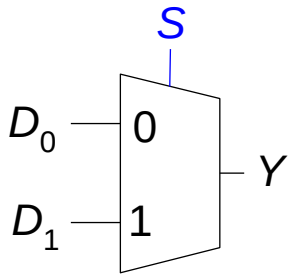




Input	Enable	Output
0	0	**Floating**
0	1	**0**
1	0	**Floating**
1	1	**1**

Linear truth table

Mux (multiplexor)



S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	Y
0	D_0
1	D_1

$S=1$

$Y = D_0\bar{S} + D_1S$

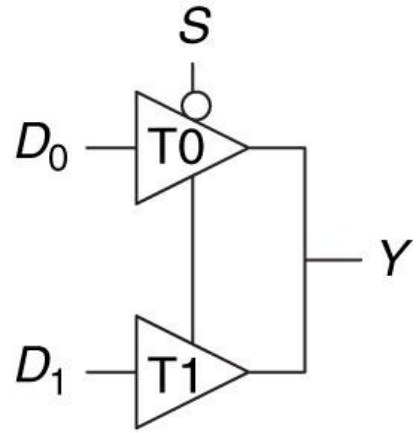
$Y = D_0 \cdot \bar{1} + D_1 \cdot 1$

$= D_0 \cdot 0 + D_1$

$= 0 + D_1$

$= D_1$

2-to-1 line multiplexor



$Y = D_0\bar{S} + D_1S$

$S=0$ $Y = D_0\bar{0} + D_1 \cdot 0$

$= D_0 \cdot 1 + 0$

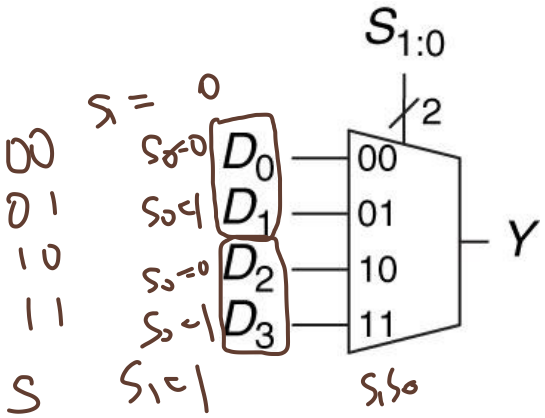
$= D_0$

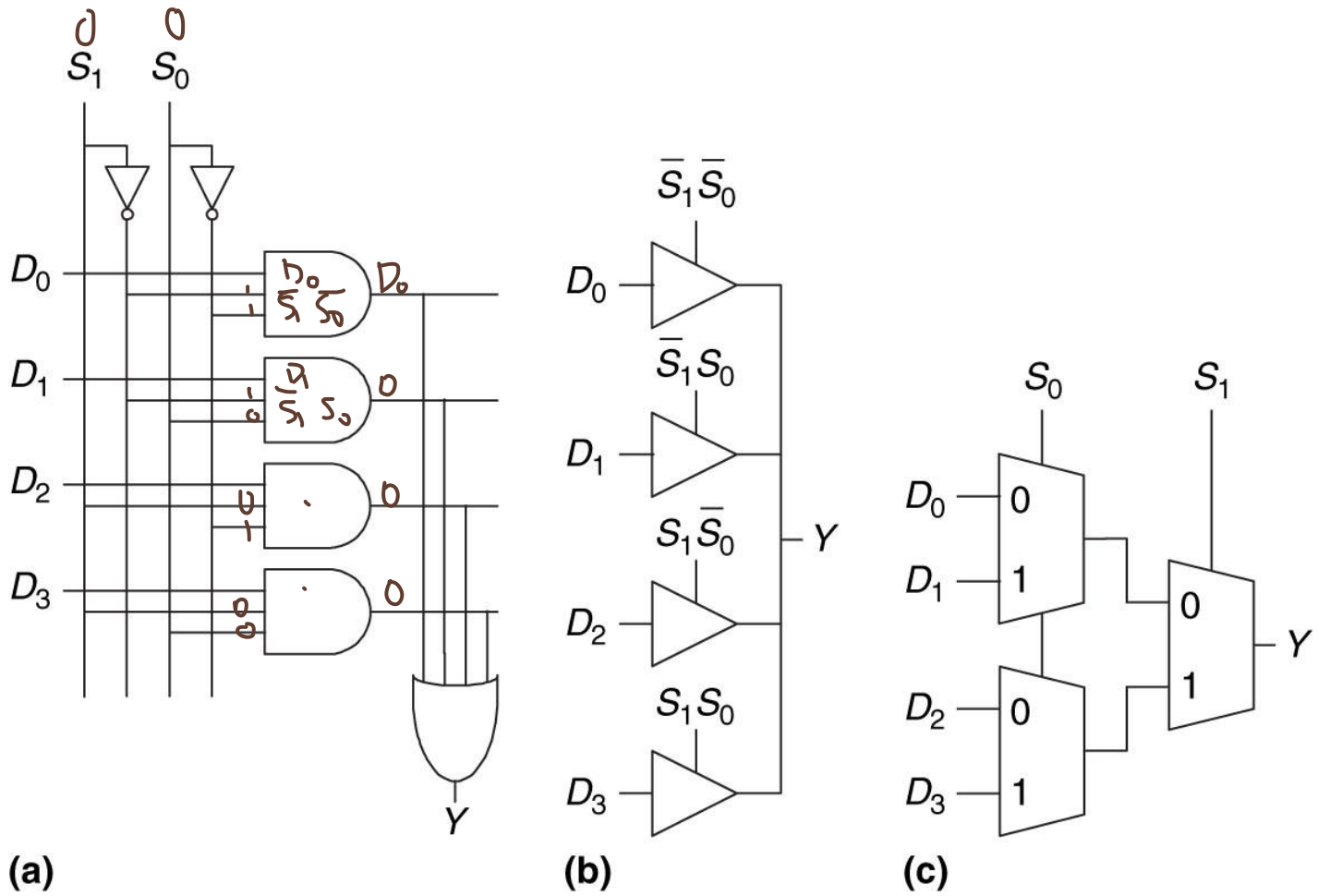
A multiplexor has **k+2^k inputs** and **1 output**.
The first k inputs (selector) represent a binary number.
The output takes the value of the remaining input indexed by this binary number.
Consider many memory locations connected to the inputs. Using the selector we can select which is loaded into a register connected to the output.

Durham University

Mux

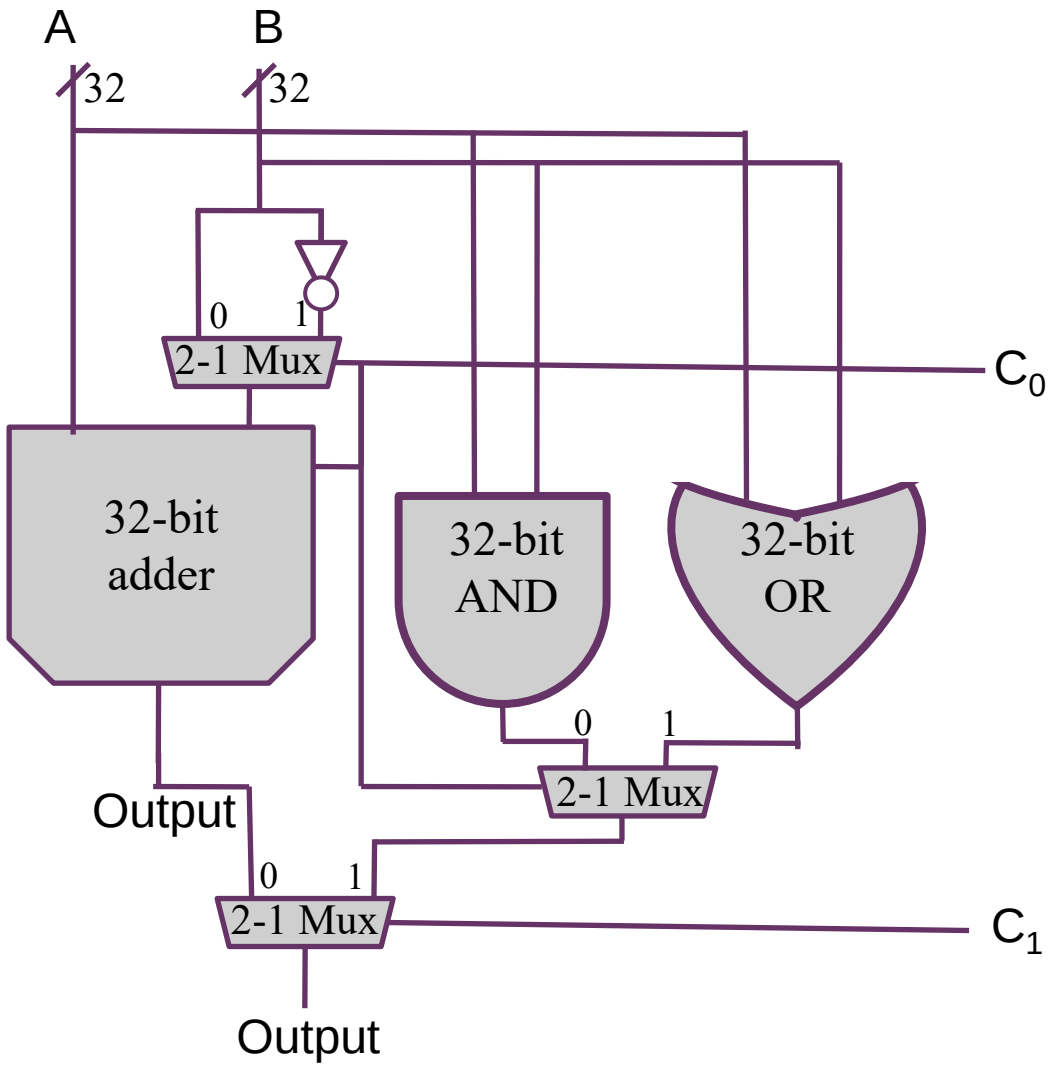
4-to-1 line multiplexor





a) two-level logic, requiring multiple input gates.
b) Using tri-state buffers – take value of input if activated, otherwise floating value.
c) Using hierarchical logic

Building a simple ALU



C_0	Fn
0	Add
1	Subtract