University of Nottingham
UK | CHINA | MALAYSIA

LECTURE 5

Digital Electronics 2

Electromechanical Devices

MMME2051

Module Convenor – Surojit Sen

Learning Outcomes

- Revision of Logic Gates
  - **Shaft Encoder**
- **Flip Flops**
  - Latch v Flip Flop
  - SR/JK/D/T Flip Flops
- Applications of Digital Circuit
  - **Series** v **Parallel** data & conversion (**Bit Shifter**)
  - Analog/Digital conversion (**R-2R Ladder** circuit)
  - **Flash Converter**

Digital v Analog

Digital

Information in form of **discrete** symbols, or **levels**

Variable can be only 1 out of a **finite number of options**

Humans interpret physical values in discrete levels

- **Alphabets**
- **Binary number**
- **Logic state**
- **Answer to the question** – "*Are you* enjoying this module?"

Analog

Information in form of **continuous** and **real-valued levels**

Variable can be only 1 out of an **infinite number of options**

The physical values exist naturally in continuous spectrum levels

- **Air pressure in this room**
- **Volume of my voice**
- **Battery voltage in your laptop**
- **Answer to the question** – "*How much* are you enjoying this module?"

Digital Methods of Information Exchange

Language – using letters

There are 26 alphabets in the English language – digital!

Numbers

Every number that we use, uses a distinct number of symbols (including the decimal point)

Binary

e.g.,
11100100

Octal

e.g.,
344

Decimal

e.g.,
228

Hexadecimal

e.g.,
E4

How does this actually relate to “numbers”?

Let us look at a number in the “Decimal” number-format, the one that we have grown up with.

Weight of the Formula Student 2021 car is

301 kg

MSB – Most Significant Bit

LSB – Least Significant Bit

$3 \times 10^{2} = 300$

$0 \times 10^{1} = 0$

$1 \times 10^{0} = 1$

How does this actually relate to “numbers”?

The same number in the Hexadecimal format will be

Weight of the Formula Student 2021 car is

12D kg

MSB – Most Significant Bit

LSB – Least Significant Bit

1 x 16^{2} = **256**	**2** x 16^{1} = **32**	**D** x 16^{0} = **13**

How does this actually relate to "numbers"?

How about in Binary?

Weight of the Formula Student 2021 car is

LSB – Least Significant Bit

MSB – Most Significant Bit

0001 0010 1101 kg

- $0 \times 2^{11} = 0$
- $0 \times 2^{10} = 0$
- $0 \times 2^{9} = 0$
- $1 \times 2^{8} = 256$

<u>**256**</u>

- $0 \times 2^{7} = 0$
- $0 \times 2^{6} = 0$
- $1 \times 2^{5} = 32$
- $0 \times 2^{4} = 0$

<u>**32**</u>

- $1 \times 2^{3} = 8$
- $1 \times 2^{2} = 4$
- $0 \times 2^{1} = 0$
- $1 \times 2^{0} = 1$

<u>**13**</u>

Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

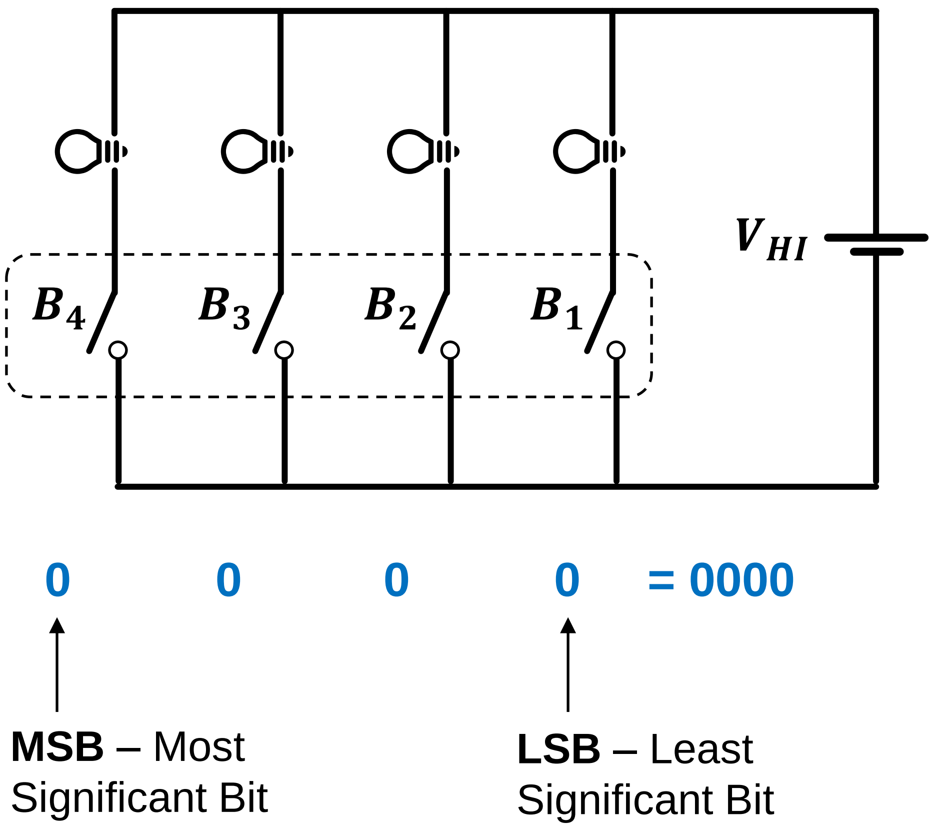
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

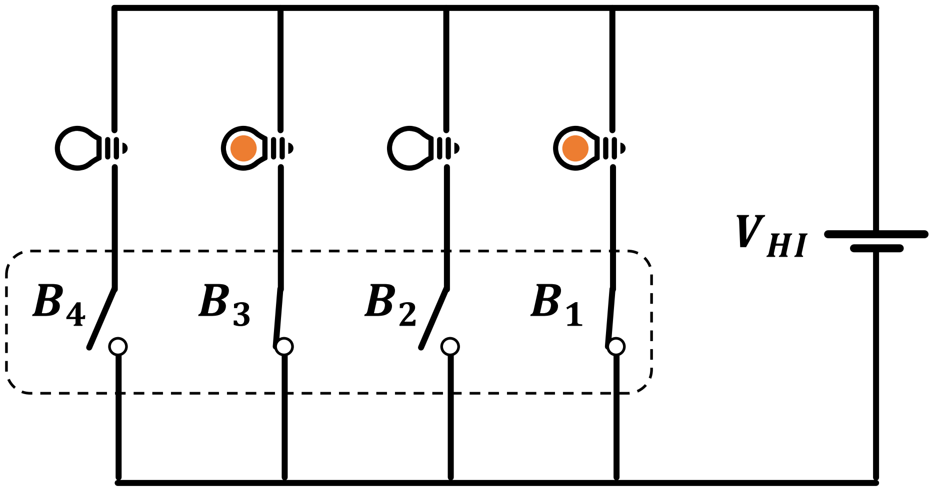
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



0 1 0 1 = 0101

MSB – Most Significant Bit

LSB – Least Significant Bit

Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

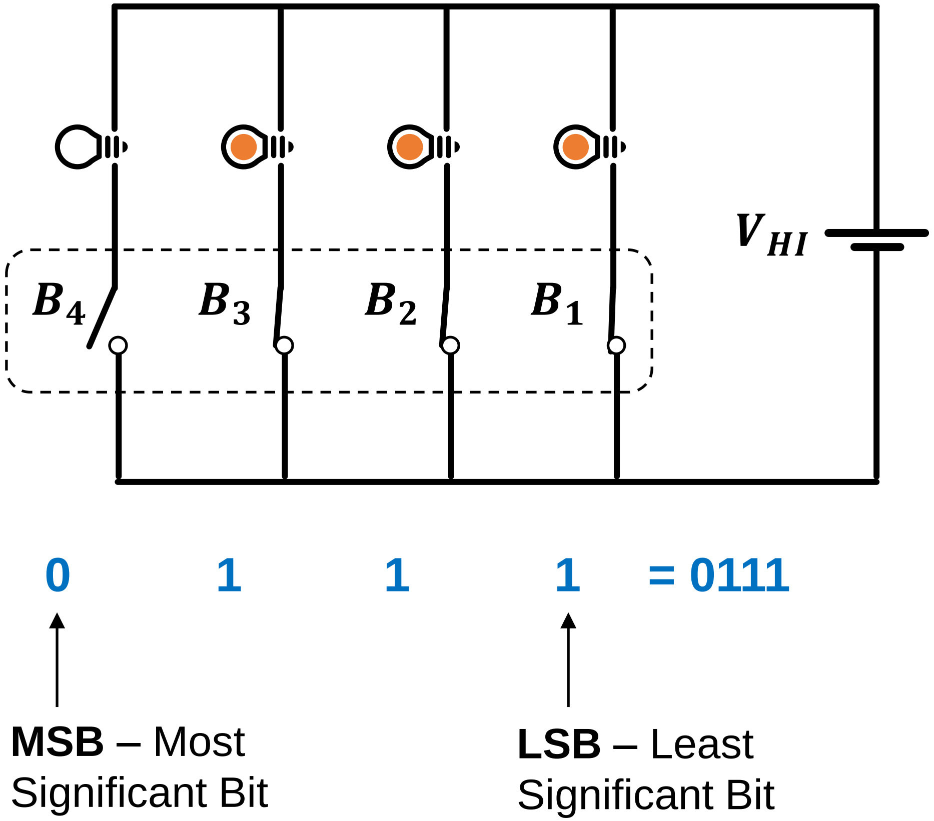
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



Information in Electronics

We use **binary number system** in logical circuits in electronics

This aligns with computer/software engineering – binary system used

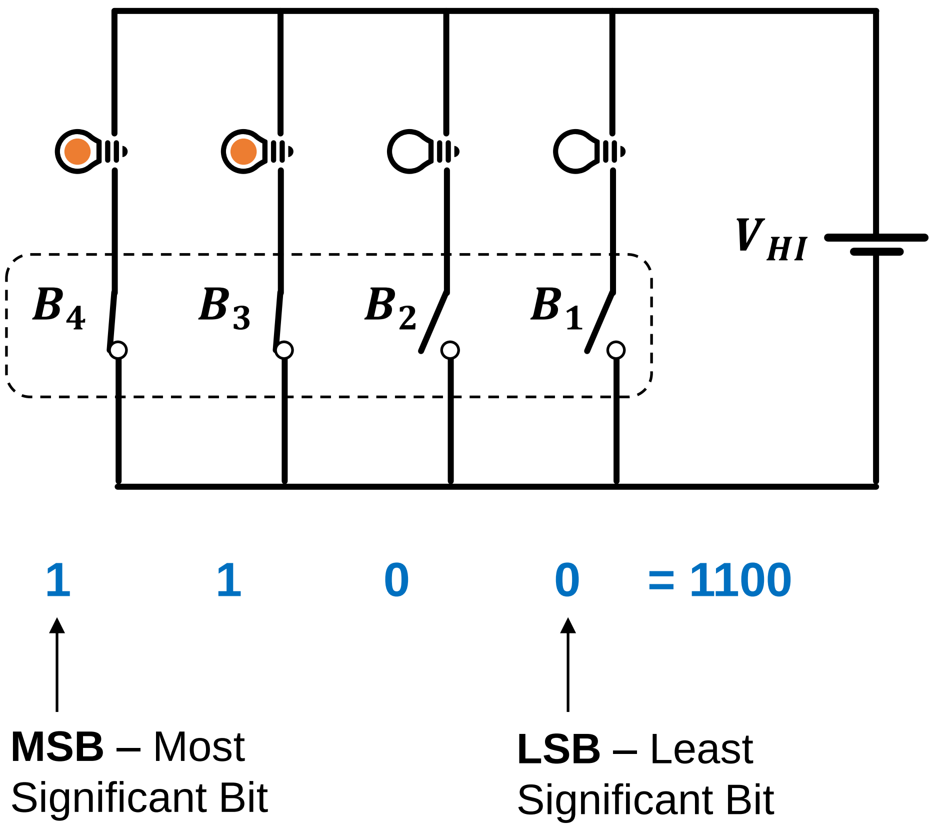
Logic – TRUE/FALSE

We said that **301** (weight of the FS21 in kg) is represented in binary as

0 0 0 1 0 0 1 0 1 1 0 1

How is this actually done in reality?

Two voltage levels – **High & Low**



How to Add/Subtract Binary Numbers?

Just the same way you do for decimal numbers!

Decimal	Binary
1	1 1 1 1 1
124	0111 1100
+229	+1110 0101
353	**1 0110 0001**
124	0111 1100
- 47	+0010 1111
77	**0100 1101**

We don't normal do **multiplication** and **division** operations on binary numbers

We shall study **Binary Algebra** later

4-bit Binary Number Range

Decimal	B_4	B_2	B_2	B_1	Binary
0	0	0	0	0	0000
1	0	0	0	1	0001
2	0	0	1	0	0010
3	0	0	1	1	0011
4	0	1	0	0	0100
5	0	1	0	1	0101
6	0	1	1	0	0110
7	0	1	1	1	0111
8	1	0	0	0	1000
9	1	0	0	1	1001
10	1	0	1	0	1010
11	1	0	1	1	1011
12	1	1	0	0	1100
13	1	1	0	1	1101
14	1	1	1	0	1110
15	1	1	1	1	1111

We would call this a 4-bit binary number – it is made of 4 bits

Maximum number we can count up to for a binary number is given by $\mathbf{2^n - 1}$

$$1\ byte \ = \ 8\ bits$$

Modern computers use **32-bit** or **64-bit** numbers in its operating system

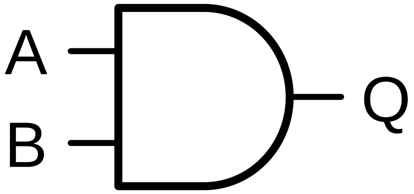
Remember the numeric data types you learnt in MATLAB last year?

- **Single** – 4 bytes
- **Double** – 8 bytes
- **Int8** – 1 byte

Logic Gates

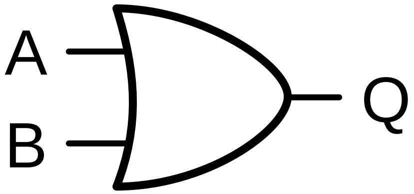
Truth Table

AND



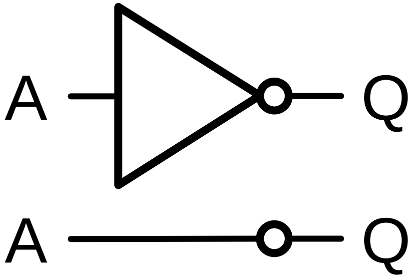
A	B	Q	Remark
0	0	**0**	HI if all inputs are HI
0	1	**0**	
1	0	**0**	
1	1	**1**	

OR



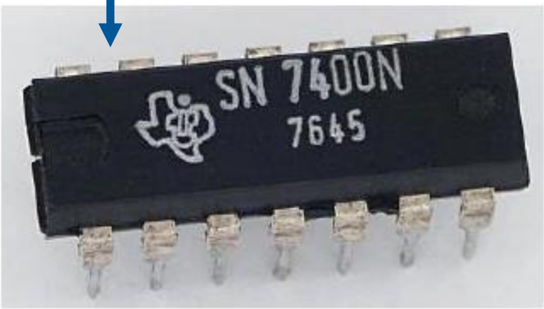
A	B	Q	Remark
0	0	**0**	HI if any input is HI
0	1	**1**	
1	0	**1**	
1	1	**1**	

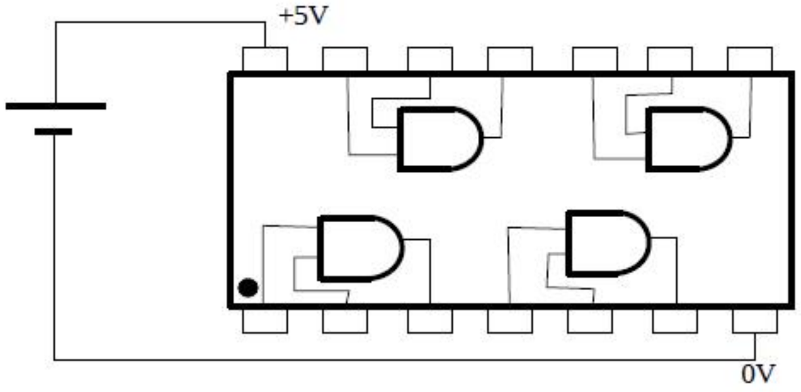
NOT



A	Q	Remark
0	**1**	Bit inversion
1	**0**	

This is an Integrated Circuit, or IC!

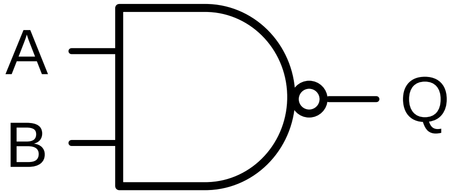




Logic Gates

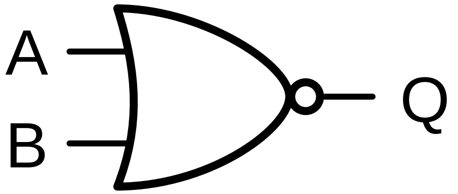
Truth Table

NAND



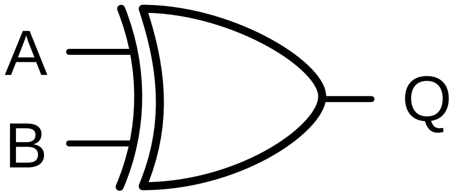
A	B	Q	Remark
0	0	**1**	LO if all inputs are HI
0	1	**1**	
1	0	**1**	
1	1	**0**	

NOR



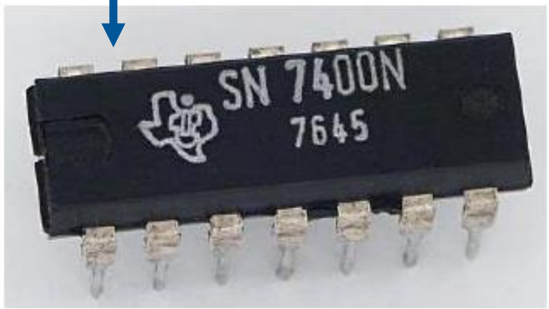
A	B	Q	Remark
0	0	**1**	LO if any input is HI
0	1	**0**	
1	0	**0**	
1	1	**0**	

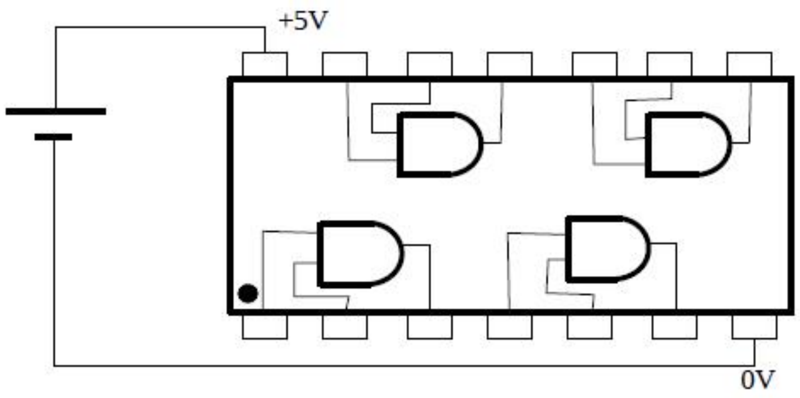
XOR



A	B	Q	Remark
0	0	**0**	HI if at least one input is HI and one is LO
0	1	**1**	
1	0	**1**	
1	1	**0**	

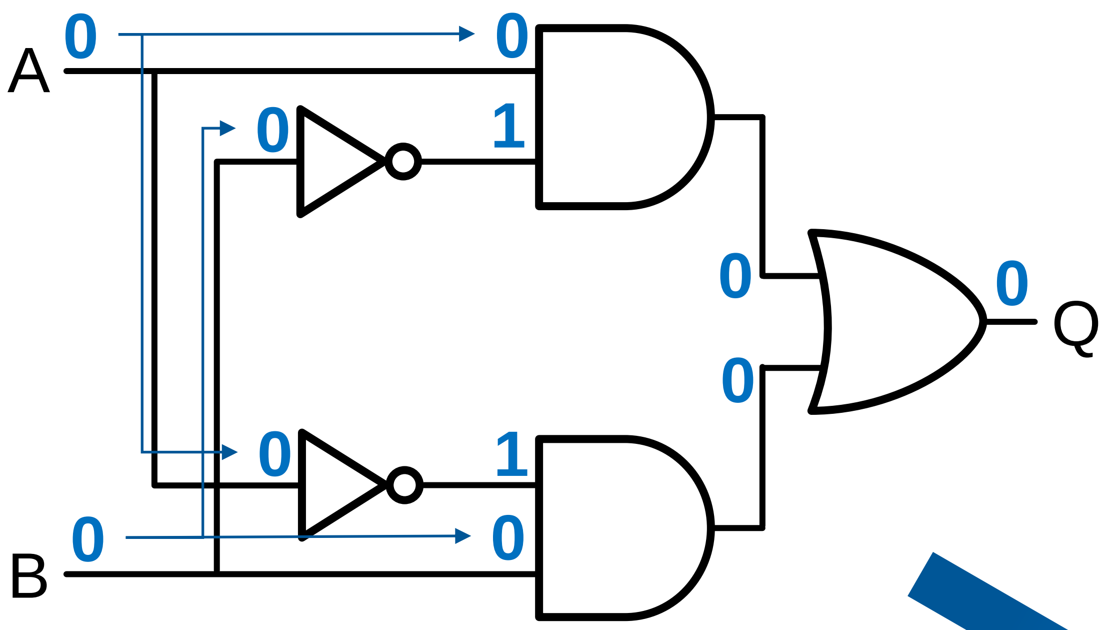
This is an Integrated Circuit, or IC!





Don't need to study this for exam

Example 3

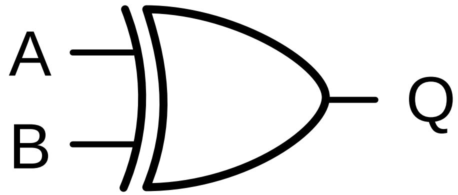


Total inputs = 2

Total combinations possible = $2^n = 4$

4 rows in truth table

A	B	Q	Remark
0	0	**0**	This is XOR gate
0	1	**1**	
1	0	**1**	
1	1	**0**	



~~**Step 1 – Identify how many inputs there are**~~

~~**Step 2 – Draw a truth table with as many number of rows as possible combinations of input bits**~~

~~**Step 3 – Try each input combination in the logic gate**~~

~~**Step 4 – Propagate the "logic" all the way to output**~~

~~**Step 5 – Fill the truth table row by row**~~

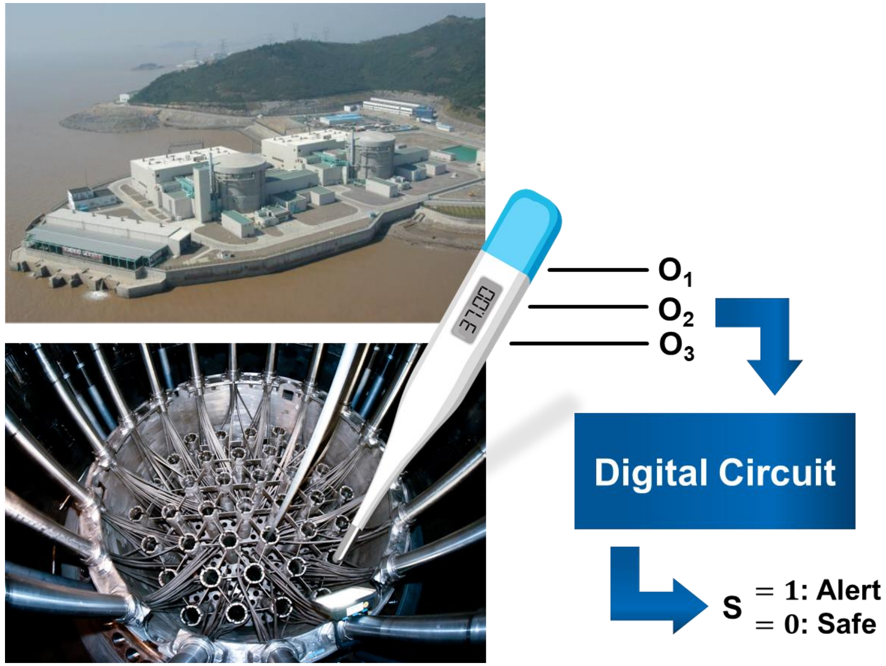
Example 4

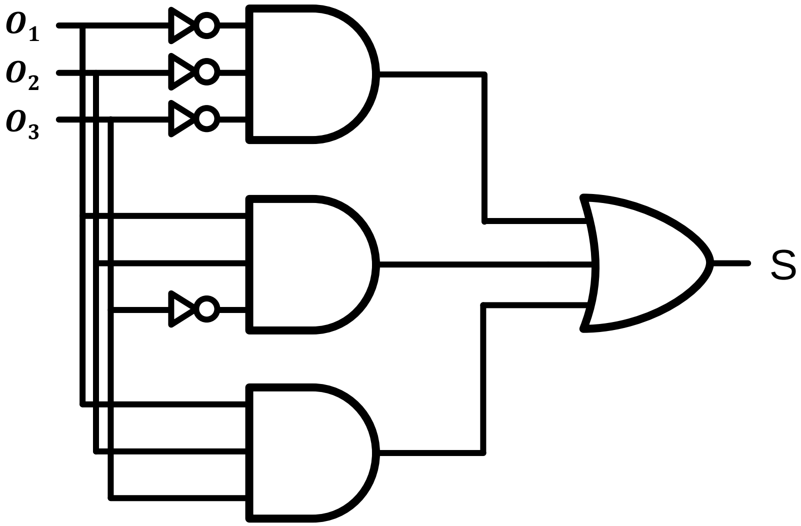
- Imagine you are designing a circuit to monitor a digital thermometer embedded in a nuclear reactor
- You want to automatically shut off the reactor when the cooling fluid rises above 50°C
- It would also be bad if the coolant froze – shut down the reactor!
- Thermometer gives a 3-bit binary output in 10°C steps –
  - $2^3 = \mathbf{8}$ **levels**
  - Count from 0 to $2^3 - 1 = \mathbf{7}$
  - **0°C** to **80°C** range of output

S=1 (as we said solving for HI) if:

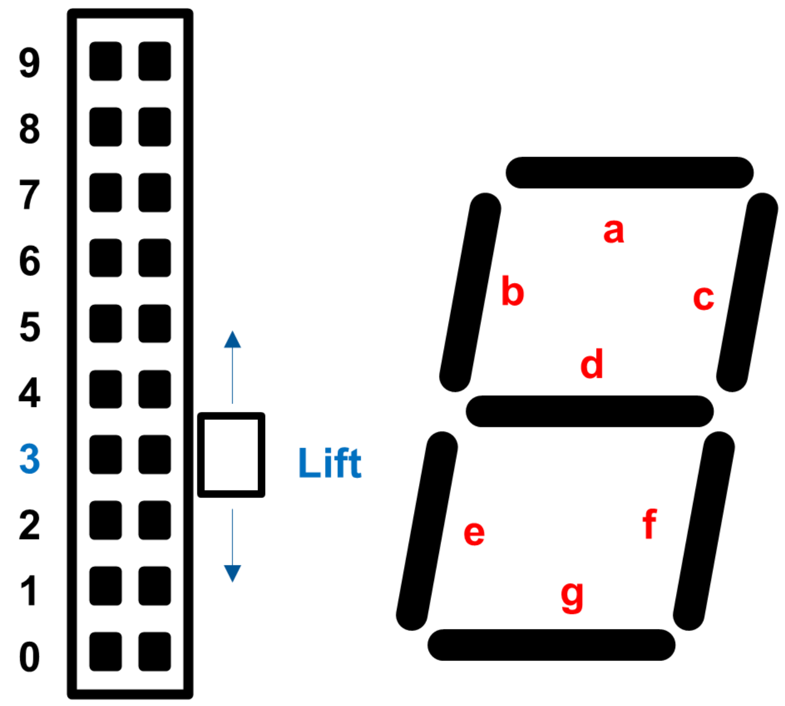
- $\boldsymbol{O_1} = \mathbf{0}$ AND $\boldsymbol{O_2} = \mathbf{0}$ AND $\boldsymbol{O_3} = \mathbf{0}$ **OR**
- $\boldsymbol{O_1} = \mathbf{1}$ AND $\boldsymbol{O_2} = \mathbf{1}$ AND $\boldsymbol{O_3} = \mathbf{0}$ **OR**
- $\boldsymbol{O_1} = \mathbf{1}$ AND $\boldsymbol{O_2} = \mathbf{1}$ AND $\boldsymbol{O_3} = \mathbf{1}$

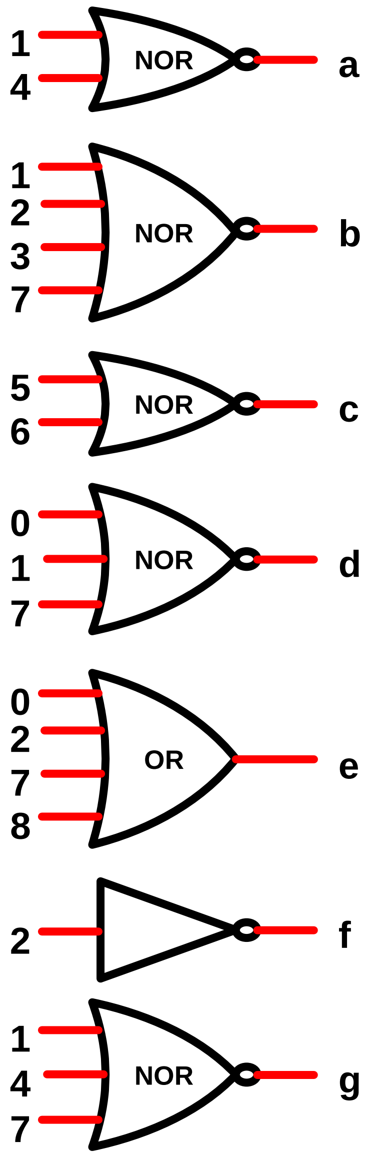
O_1	O_2	O_3	Dec	Temp	S
0	0	0	0	0°C	1
0	0	1	1	10°C	0
0	1	0	2	20°C	0
0	1	1	3	30°C	0
1	0	0	4	40°C	0
1	0	1	5	50°C	0
1	1	0	6	60°C	1
1	1	1	7	70°C	1





Example 5





	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Learning Outcomes

- Revision of Logic Gates
  - **Shaft Encoder**
- **Flip Flops**
  - Latch v Flip Flop
  - SR/JK/D/T Flip Flops
- Applications of Digital Circuit
  - **Series** v **Parallel** data & conversion (**Bit Shifter**)
  - Analog/Digital conversion (**R-2R Ladder** circuit)
  - **Flash Converter**

Shaft Encoder

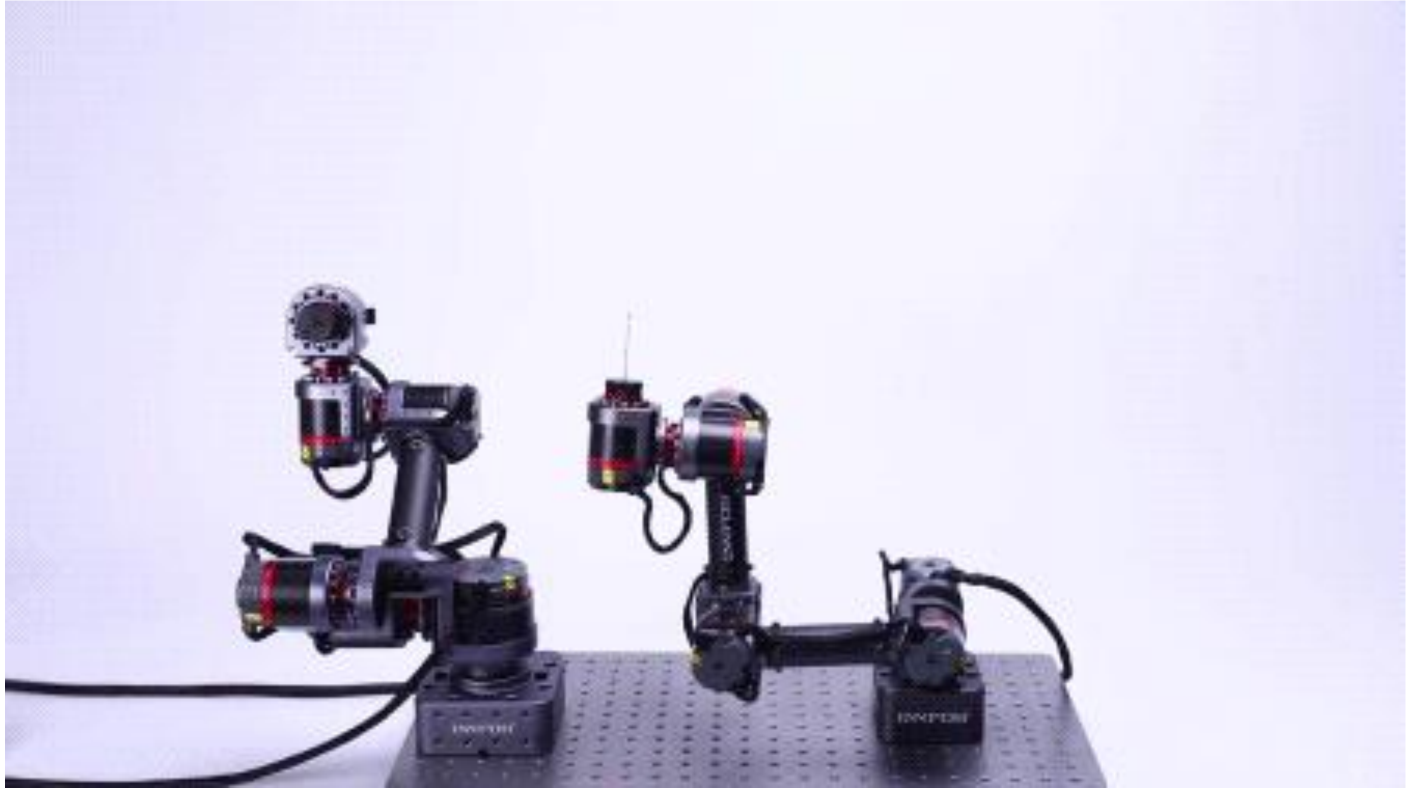
How does the controller know when to start/stop the motor controlling the robotic arm joint?

A sensor is required that gives the **accurate position of the joint**

A **Shaft Encoder** can do that

Shaft encoder can provide:

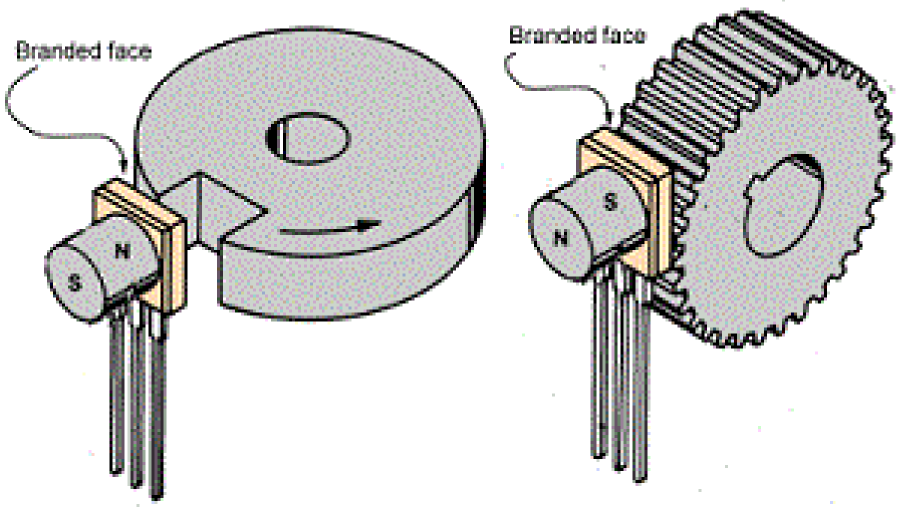
- **Angular Position**
- **Angular Speed**
- **Direction**

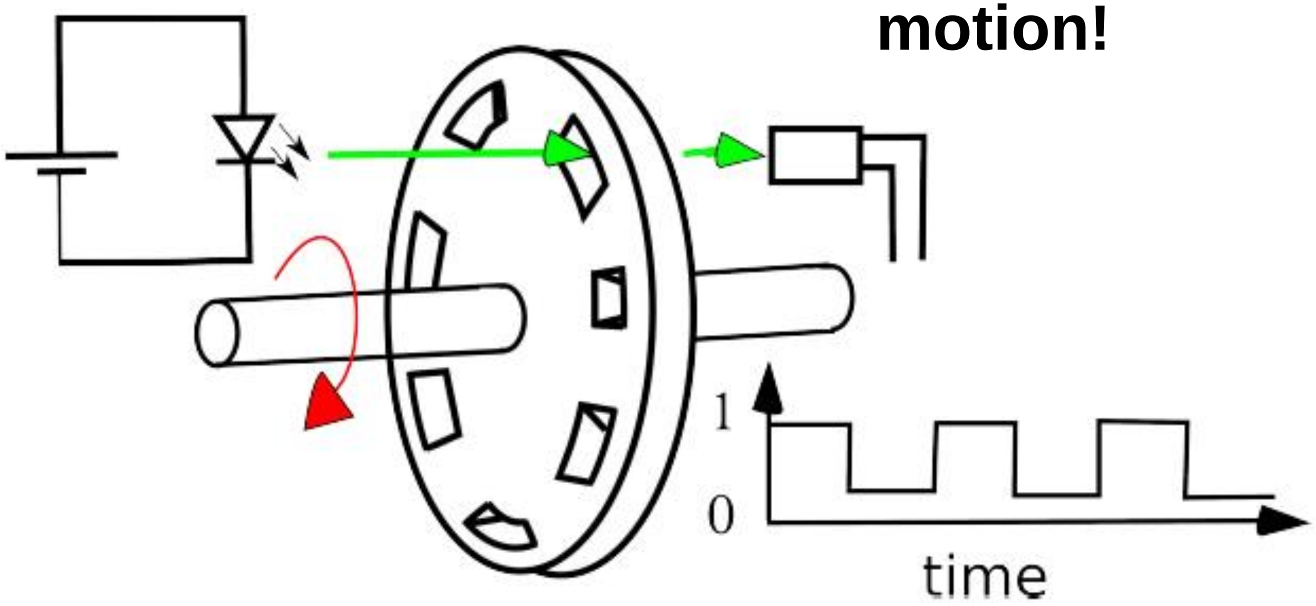


Shaft Encoder

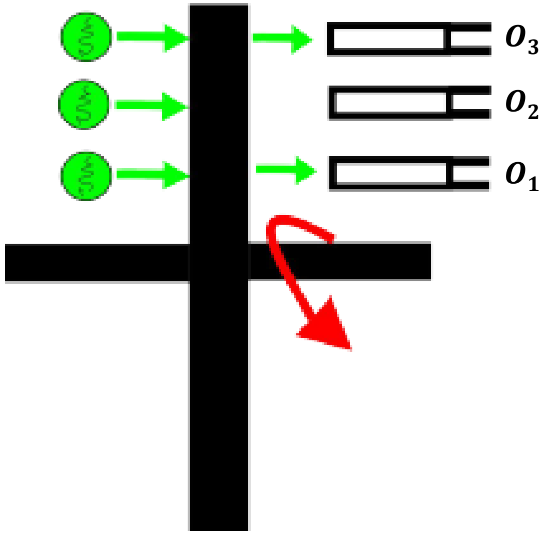
This is the most basic form of shaft encoder. It has some inherent problems though:

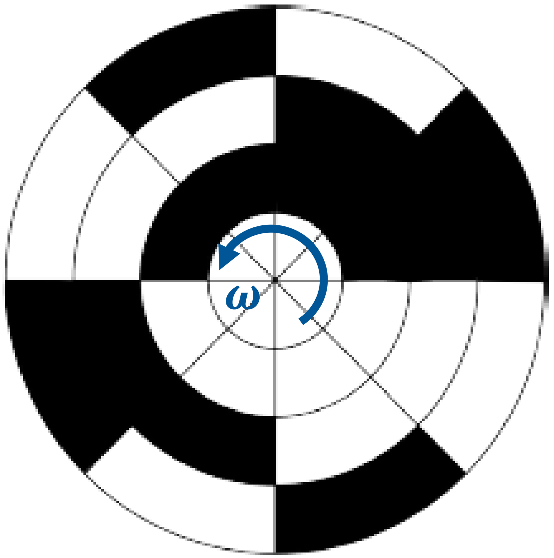
- **Only detects speed**
- **Not position**
- **Not direction of motion!**





Shaft Encoder





O_1	O_2	O_3	Dec	Angle
0	0	0	**0**	**0°-45°**
0	0	1	**1**	**45°-90°**
0	1	0	**2**	**00°-135°**
0	1	1	**3**	**135°-180°**
1	0	0	**4**	**180°-225°**
1	0	1	**5**	**225°-270°**
1	1	0	**6**	**270°-315°**
1	1	1	**7**	**315°-360°**

This is a motor **position encoder**

This solves all the problems in the previous design as it gives the position information

The **speed** and **direction** can be “figured out” programmatically

How do we increase the angular resolution?

Yes, add more bits!

Say there are n bits

Angular resolution$= \frac{360°}{2^n}$

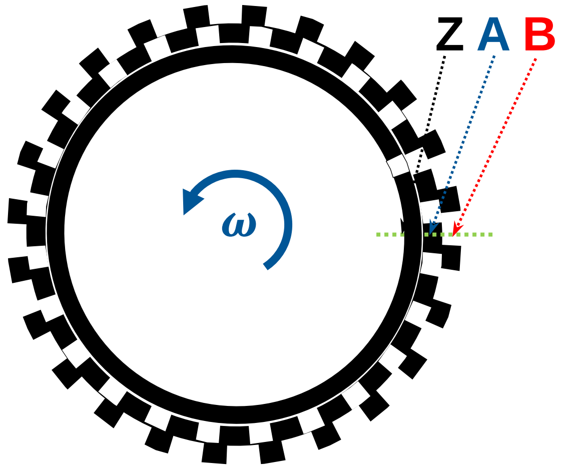
For 8-bit encoder, angular resolution$= \frac{360°}{2^8} = 1.406°$

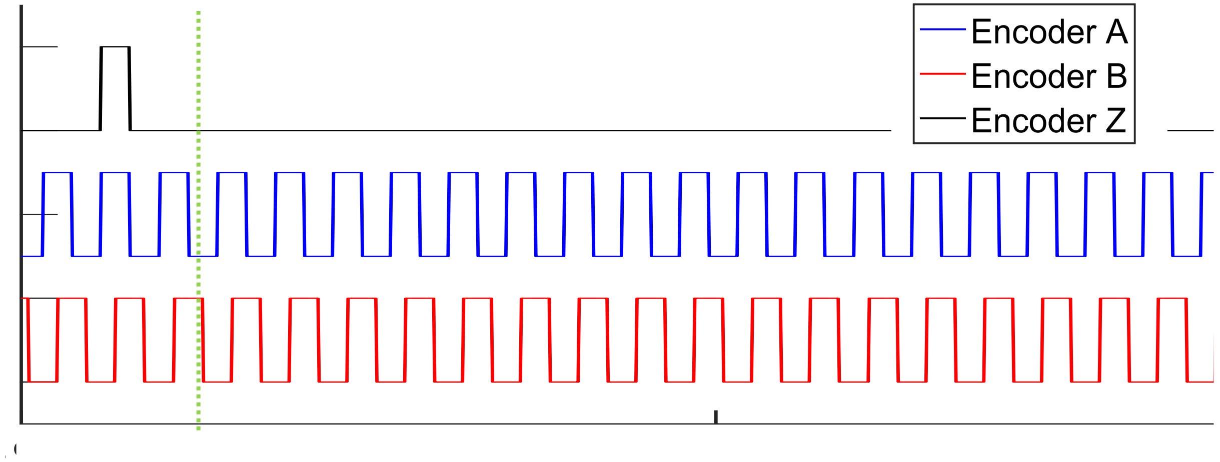
Shaft Encoder

This is an **incremental encoder**

Notice this has two incremental pulses **A** and **B** that are **90° phase shifted** from each other. This allows to detect **direction** of rotation

The third bit is the **Z** pulse which triggers once every revolution, indicating a **single revolution** has happened





Learning Outcomes

- Revision of Logic Gates
  - **Shaft Encoder**
- **Flip Flops**
  - Latch v Flip Flop
  - SR/JK/D/T Flip Flops
- Applications of Digital Circuit
  - **Series** v **Parallel** data & conversion (**Bit Shifter**)
  - Analog/Digital conversion (**R-2R Ladder** circuit)
  - **Flash Converter**

Memory in computers

What is a Computer?

It is essentially a really big and complex electronic circuit that **processes binary information** using **logical circuits**

Logical circuit (as the name suggests) uses logic (*if "A is happening" then "make B happen"*) to arrive at decisions

The basic building block of logical circuits is a **logic gate**

There are mainly 3 kinds of gates:

AND – outputs HI if all inputs are HI

OR – outputs HI if any input is HI

NOT – inverts the bi (HI becomes LO and LO becomes HI)

We studied about computers in the topic on Logic Gates last week

Can we make Computers **SMARTER?**

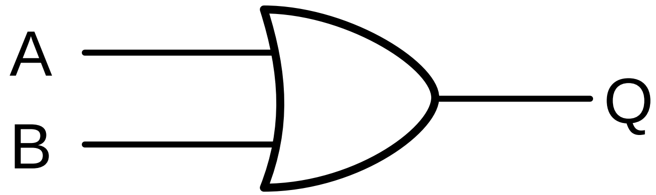
Yes, by giving it the power to **REMEMBER!**

With MEMORY, the computer can now **make decisions**, and **store** them!

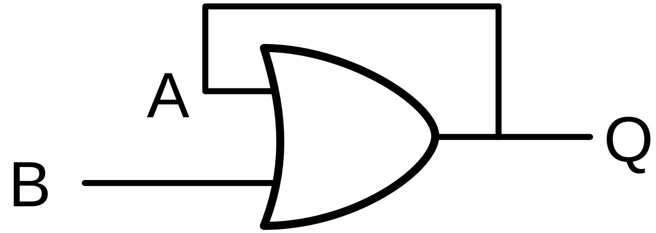
Data (decision is also data) in a computer is in form of 1s and 0s. So we need a circuit that can remember the value of a bit (0 or 1, LO or HI)

We use a **Latch** to do this

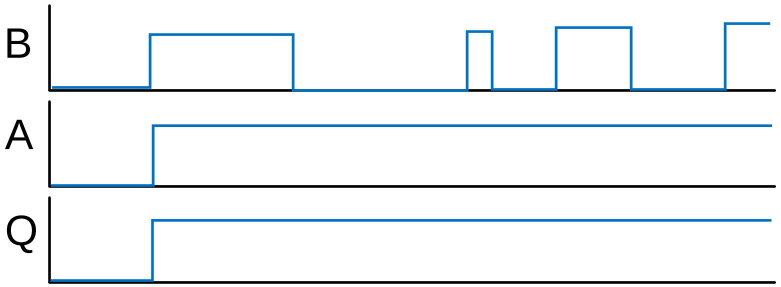
Memory in computers



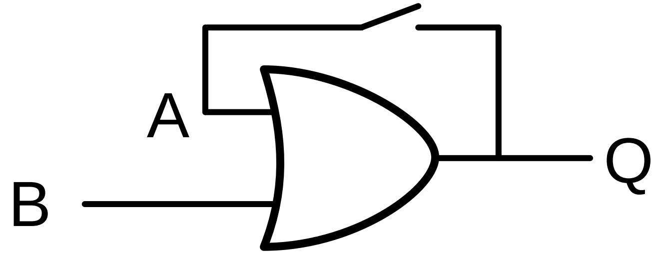
Let us take an **OR gate**



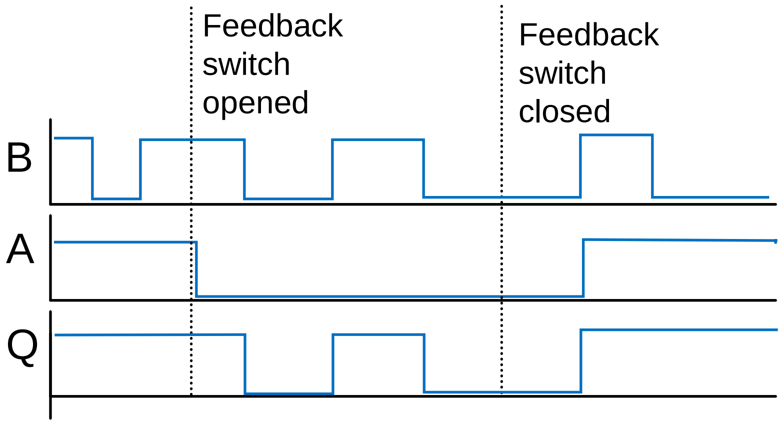
Let us “**feed back**” output Q as an input to the gate



Once Q is **“set”** HI, it will stay HI no matter what input B we apply – **memory!**



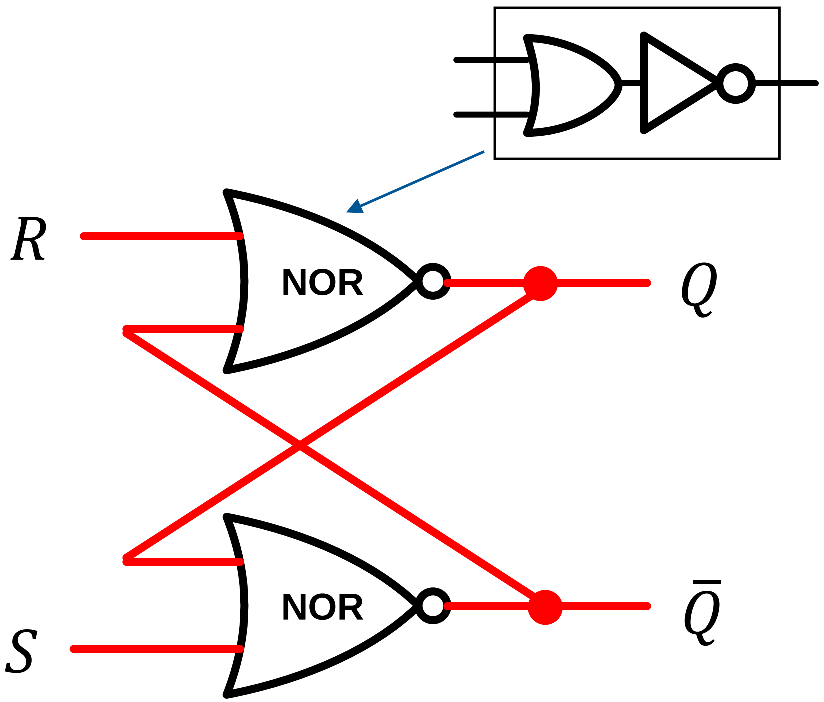
We can **“reset”** this “**memory block**” by breaking the feedback using some form of switch

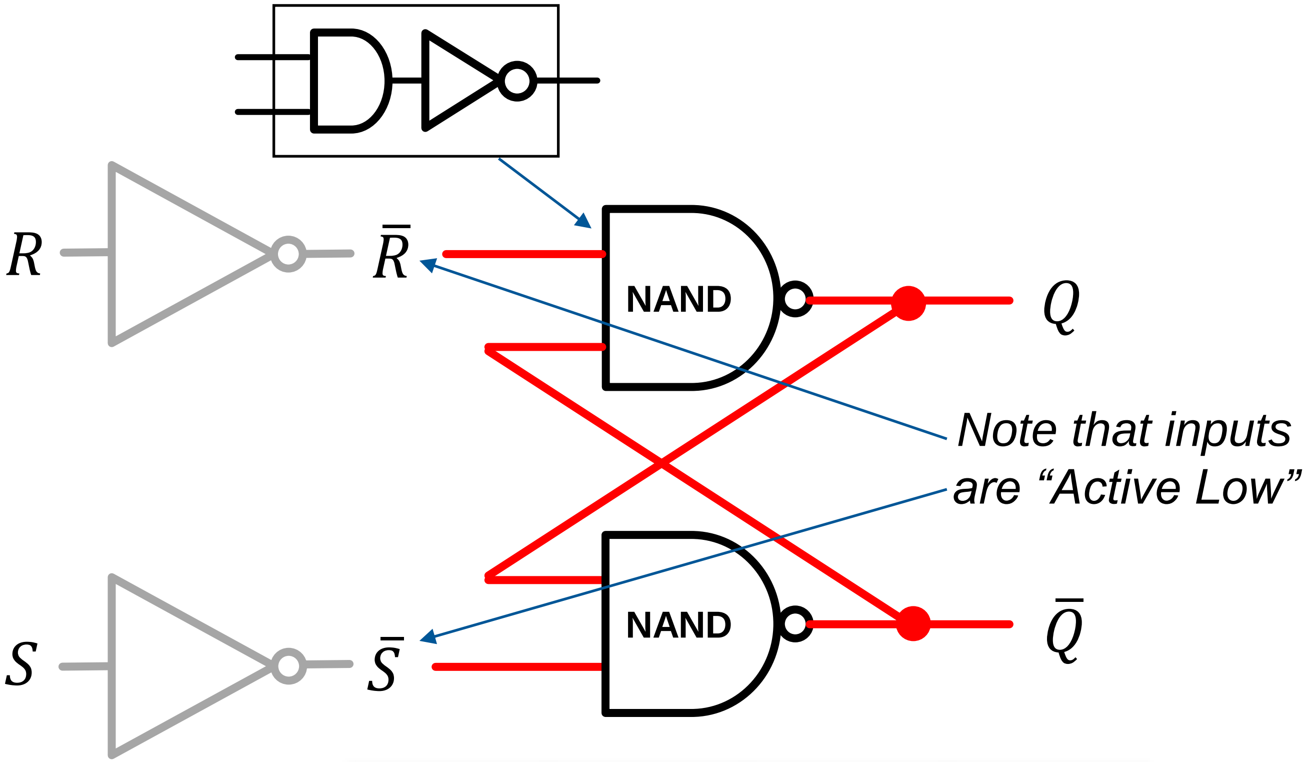


When feedback loop is broken, **output Q simply follows input B**

Closing the feedback loop again resumes the **latching functionality**

SR (“Set Reset”) Latch





R	S	Q_{next}	$\bar{Q}_{next}$
0	0	Q	$\bar{Q}$
0	1	1	0
1	0	0	1
1	1	0	0

These are invalid inputs, as it says $\bar{Q}$ = Q=0 !

$\bar{R}$	$\bar{S}$	Q_{next}	$\bar{Q}_{next}$
1	1	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
0	0	0	0

"Active Low" and "Active High"

A signal is called "Active High" when the physical voltage on the port/wire/contact is **high** (e.g., 5V, 3.3V, 12V depending on the application) when the signal is **active**

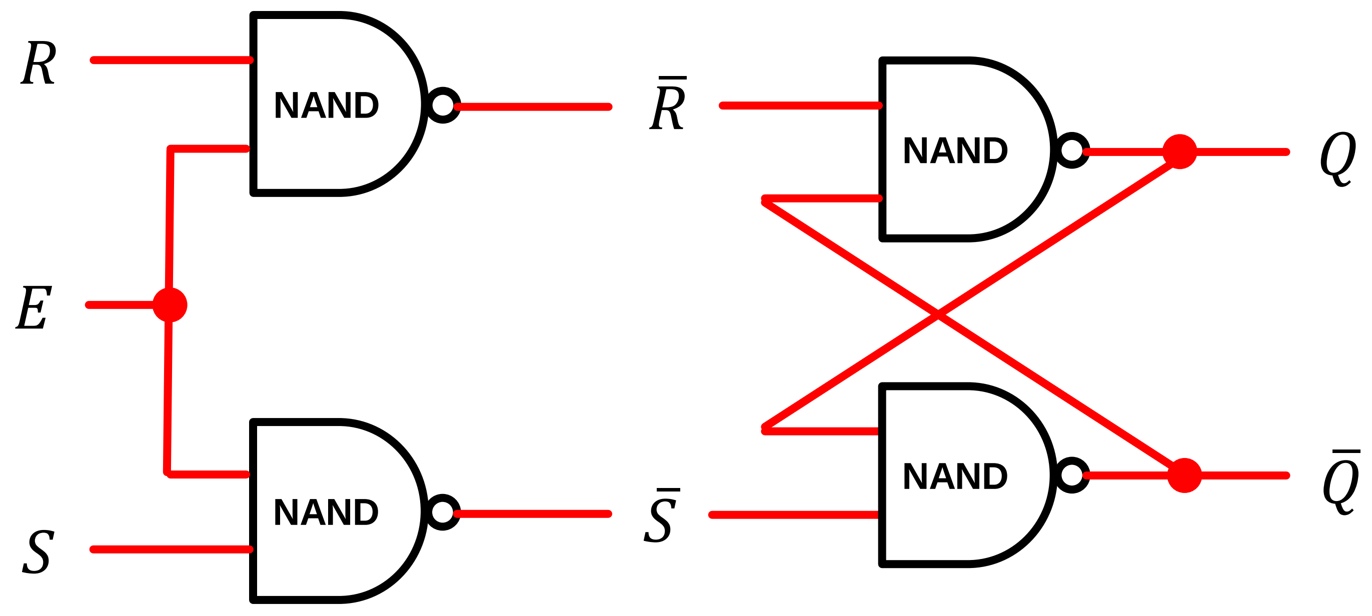
Similarly, a signal is called "Active Low" when the physical voltage on the port/wire/contact is **low** (e.g., 0V) when the signal is **active**

The concept of "Active" is purely for human interpretation. Say you name a signal "Set" (like the SR Latch). In the **NOR implementation**, if you want to "Set" the latch, you apply a High voltage to the S port.

In case of **NAND implementation**, if you want to "Set" the latch, you apply a Low voltage to the S port.

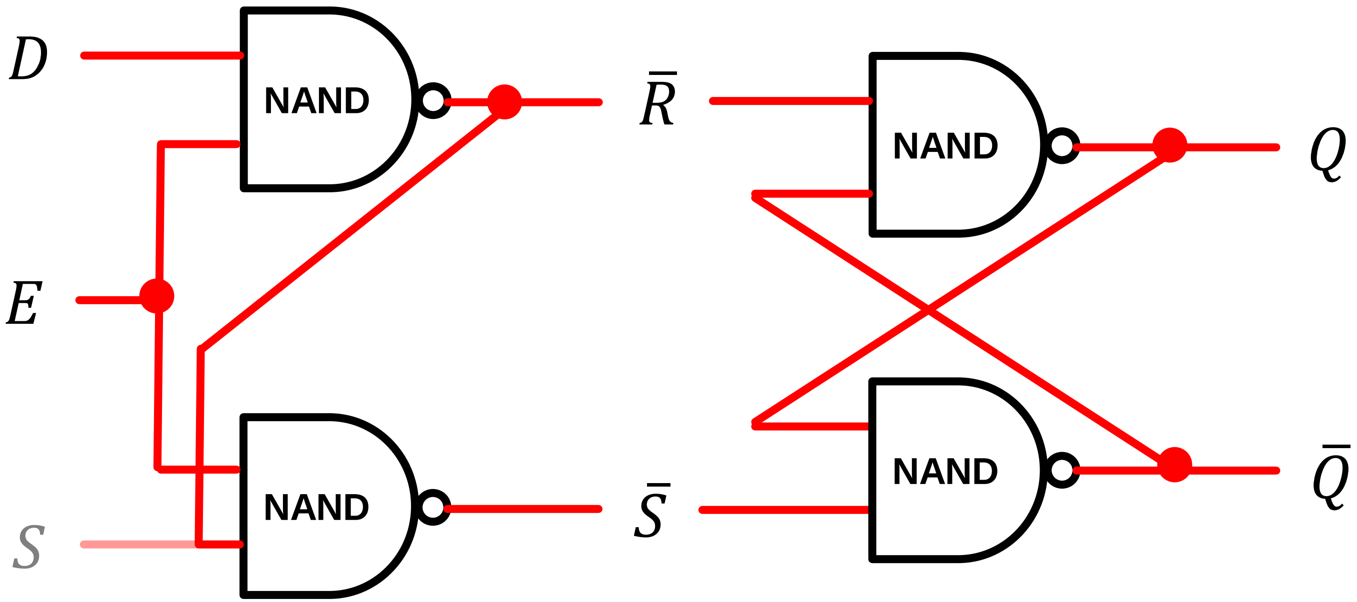
Gating a Latch to add "Enabling"

As the name suggests, an Enable input simply gives you the option to activate the Latch Set/Reset functionality



R	S	E	Q_{next}	$\bar{Q}_{next}$
0	0	0	Q	$\bar{Q}$
0	1	0	Q	$\bar{Q}$
1	0	0	Q	$\bar{Q}$
1	1	0	Q	$\bar{Q}$
0	0	1	Q	$\bar{Q}$
0	1	1	1	0
1	0	1	0	1
1	1	1	0	0

D (“Delay”) Gated Latch



E	D	Q_{next}	$\bar{Q}_{next}$
0	0	Q	$\bar{Q}$
0	1	Q	$\bar{Q}$
1	0	0	1
1	1	1	0

Synchronising the Latch with a Clock Signal

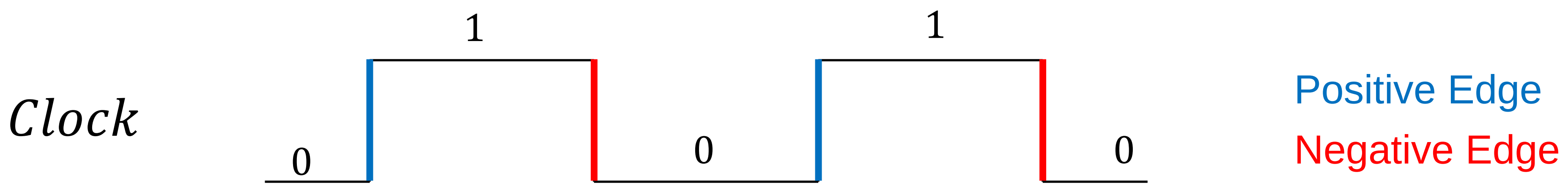
A Computer always works with a clock

A clock signal is simply a **square waveform** of a particular frequency – **faster clock** means **faster processing** power

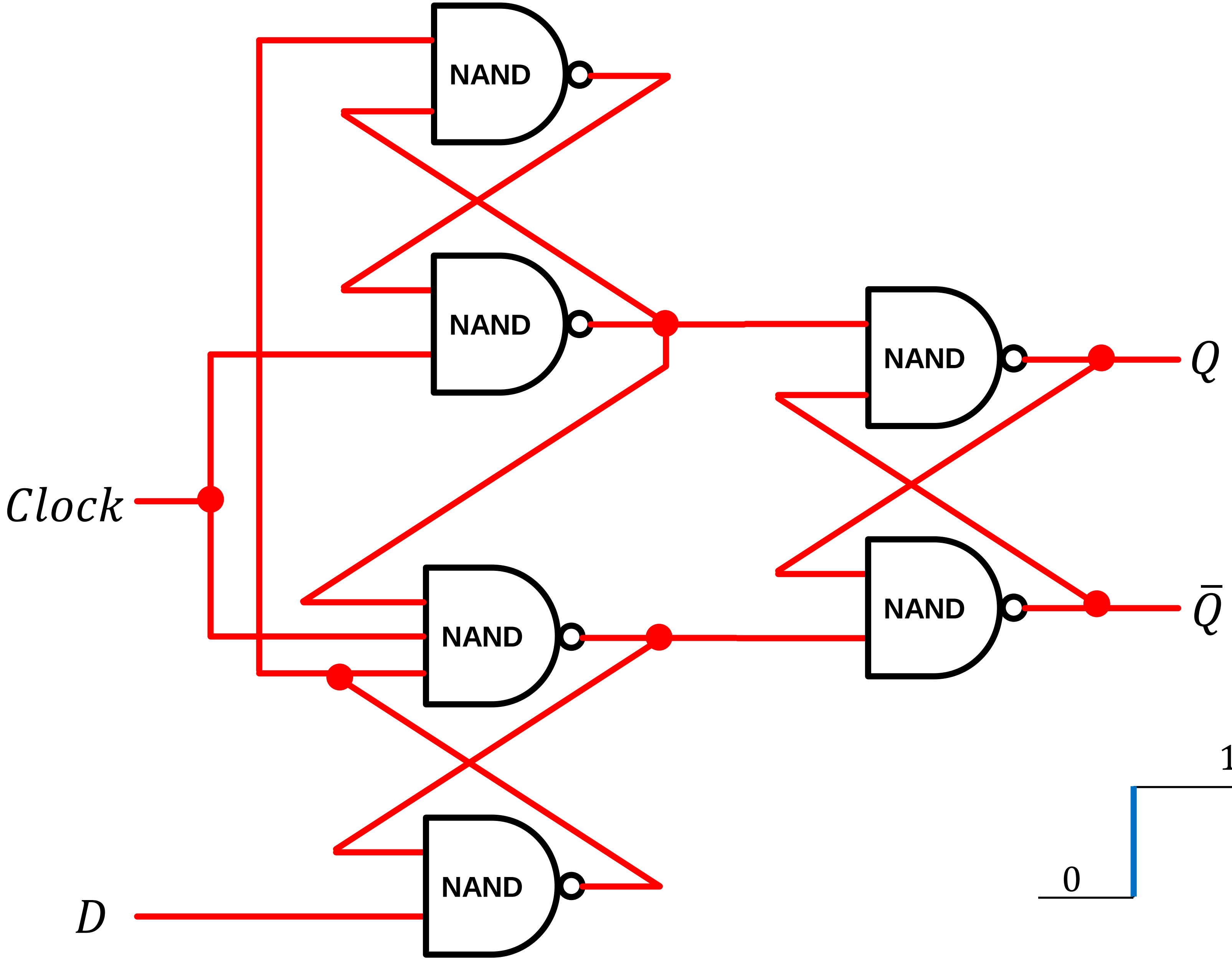
Propagation & **processing** of digital signals are expected to **happen at every clock pulse**

Clock pulse can be the **rising edge of the clock signal**

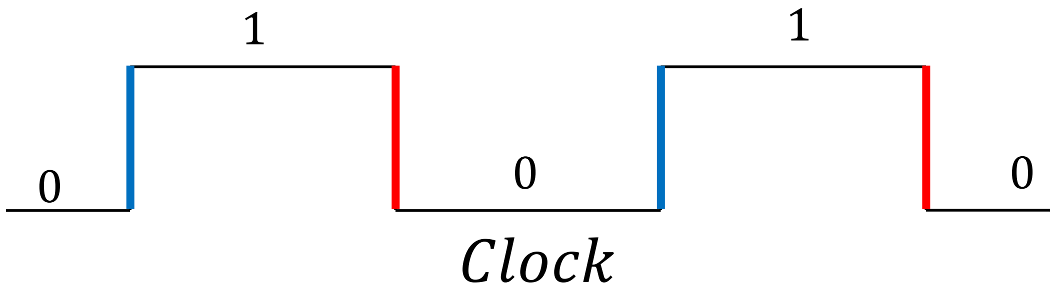
This is called Edge-Triggering



"Edge Triggered" D Flip-Flop



$Clock$	D	Q_{next}	$\bar{Q}_{next}$
0	0	Q	$\bar{Q}$
0	1	Q	$\bar{Q}$
1	0	Q	$\bar{Q}$
1	1	Q	$\bar{Q}$
$1 \rightarrow 0$	0	Q	$\bar{Q}$
$1 \rightarrow 0$	1	Q	$\bar{Q}$
$0 \rightarrow 1$	0	0	1
$0 \rightarrow 1$	1	1	0



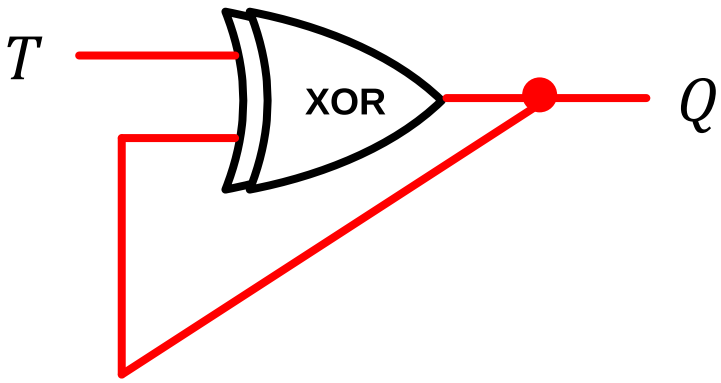
T Flip-Flop

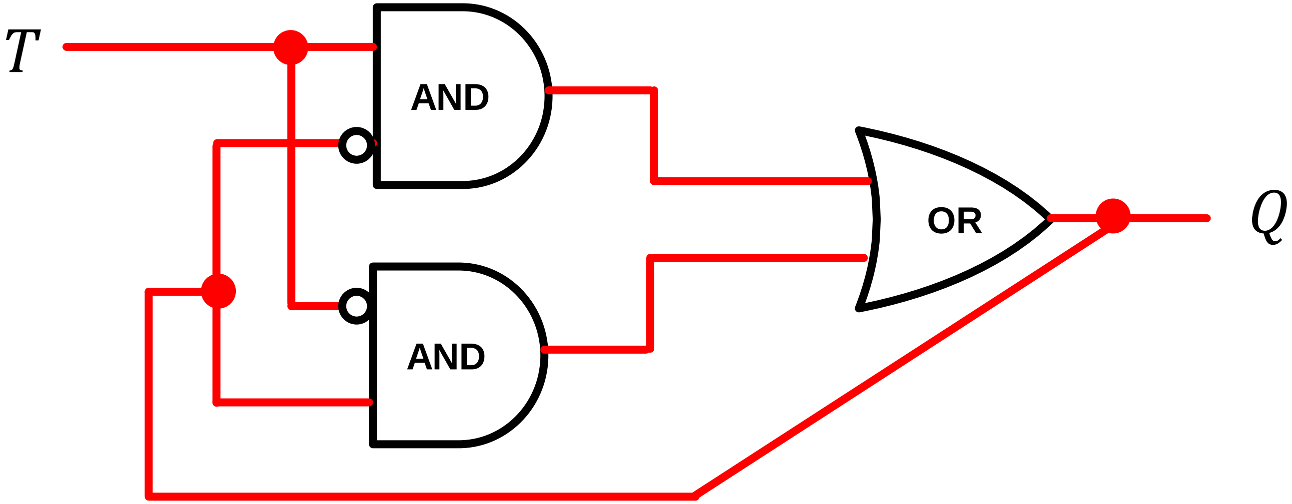
T	Q_{next}	$\bar{Q}_{next}$
0	1	0
1	0	1

XOR operator

Example of Binary Algebra (discuss later)

$$Q_{next} = T \oplus Q = T\bar{Q} + \bar{T}Q$$

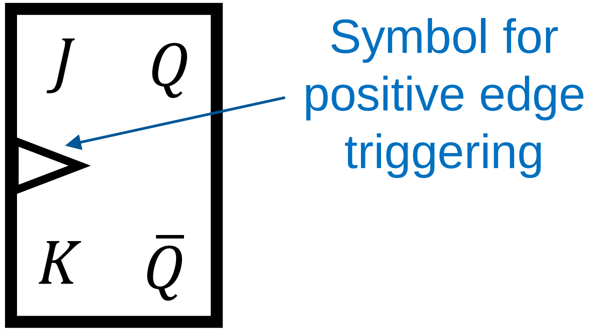




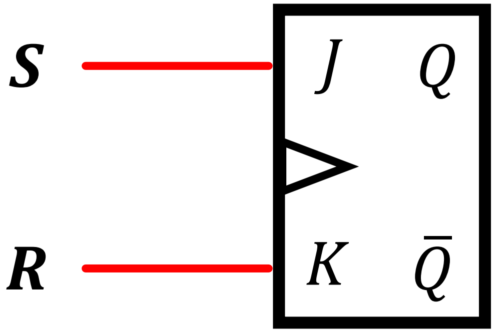
JK Flip-Flop

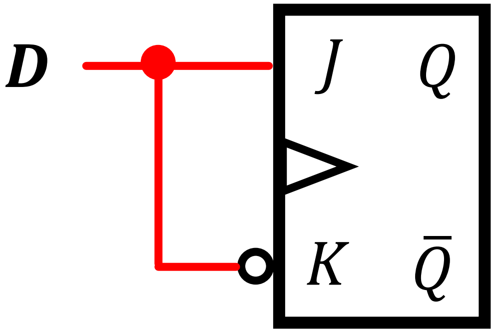
$Clock$	J	K	Q_{next}	$\bar{Q}_{next}$
Any other combination			Q	$\bar{Q}$
$0 \rightarrow 1$	0	0	Q	$\bar{Q}$
$0 \rightarrow 1$	0	1	0	1
$0 \rightarrow 1$	1	0	1	0
$0 \rightarrow 1$	1	1	$\bar{Q}$	Q

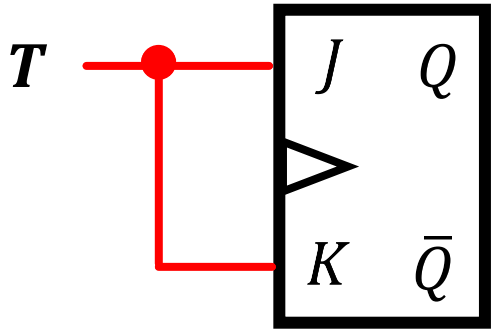
$$Q_{next} = J\bar{Q} + \bar{K}Q$$



Universal Flip-Flop

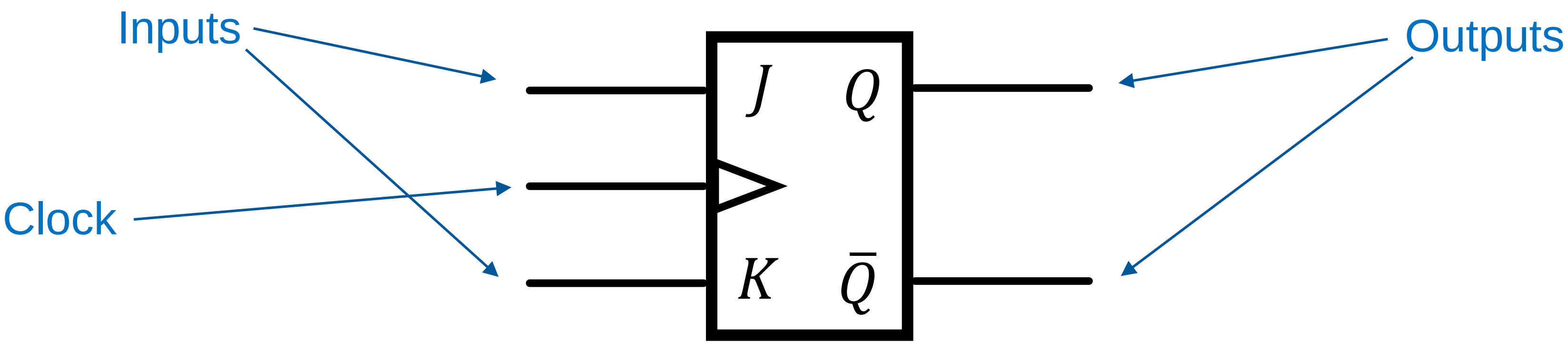


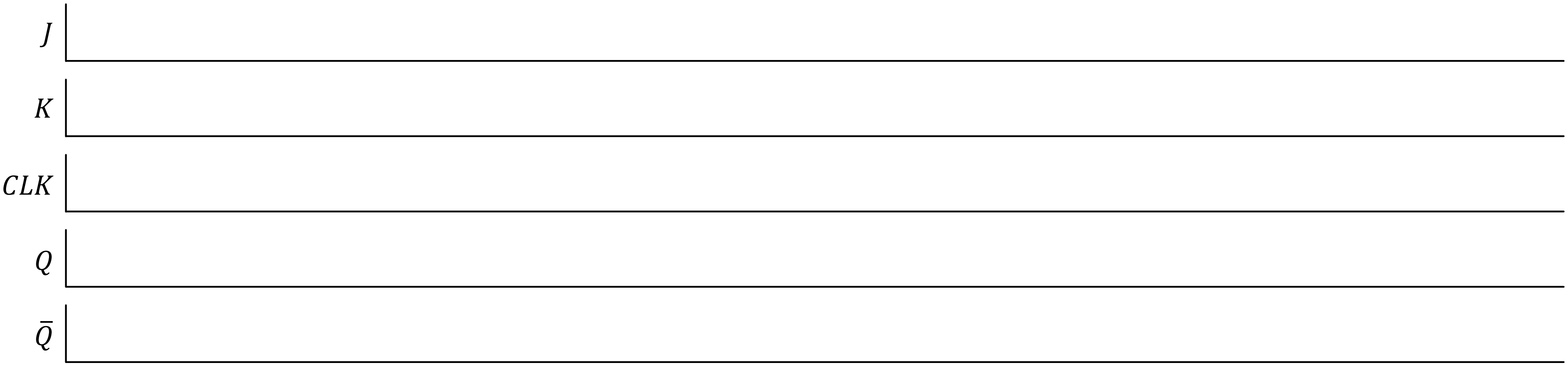




JK Flip-Flop

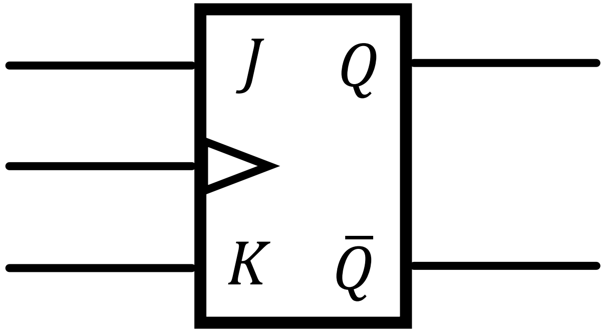
Let us revise how to operate the JK Flip Flop

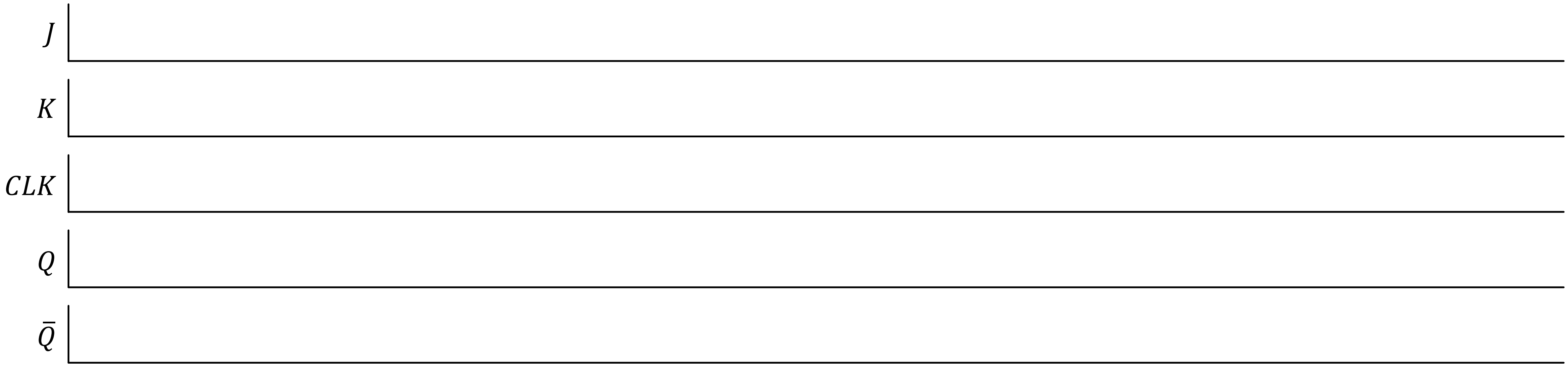




JK Flip-Flop

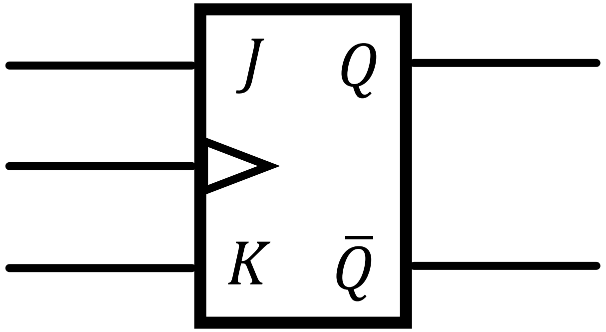
Let us revise how to operate the JK Flip Flop

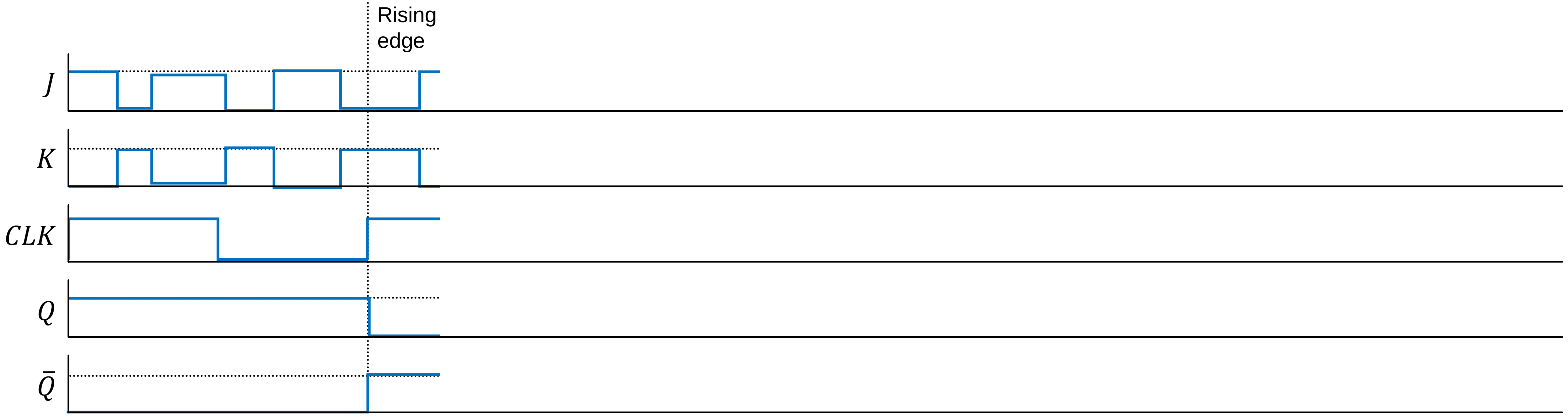




JK Flip-Flop

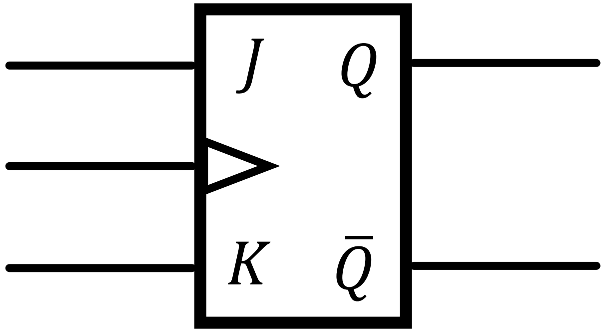
Let us revise how to operate the JK Flip Flop

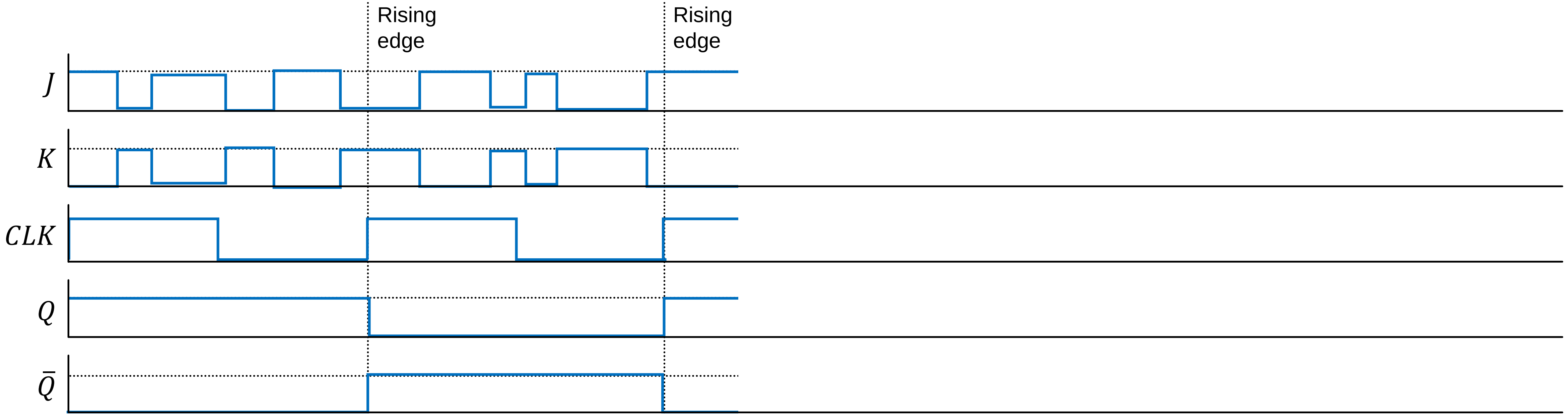




JK Flip-Flop

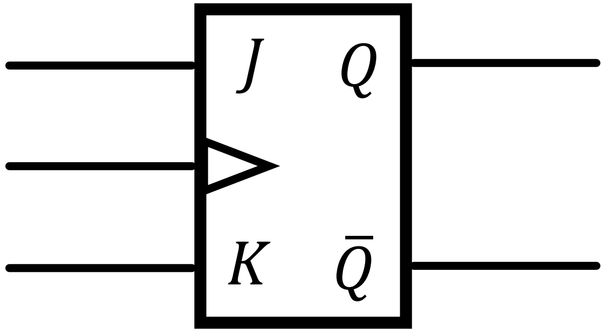
Let us revise how to operate the JK Flip Flop

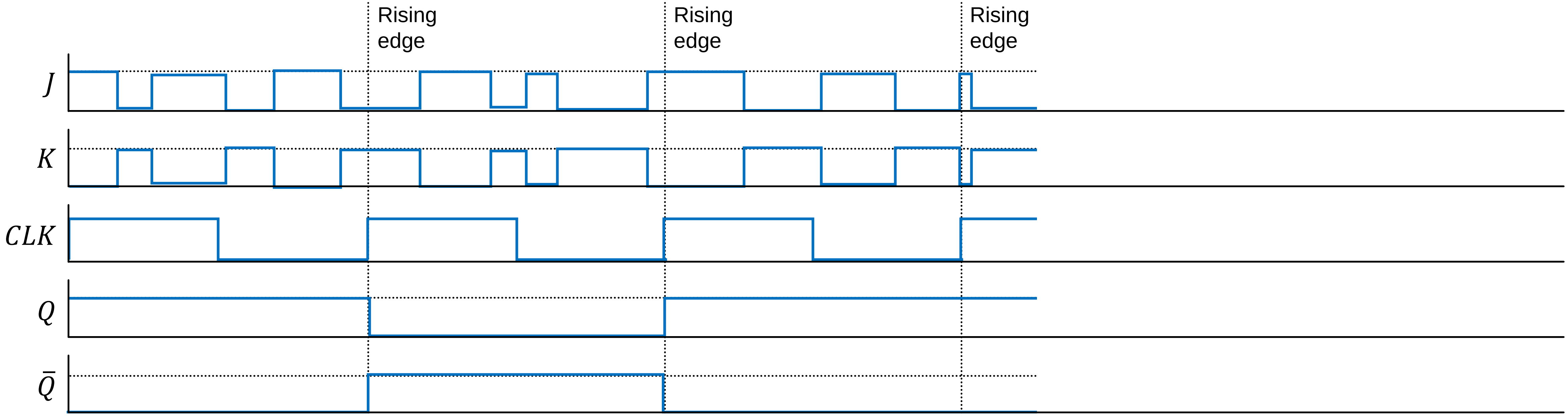




JK Flip-Flop

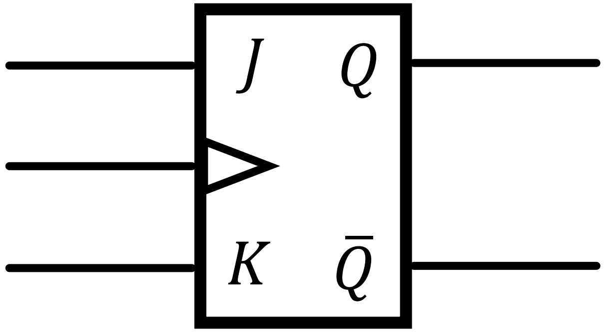
Let us revise how to operate the JK Flip Flop

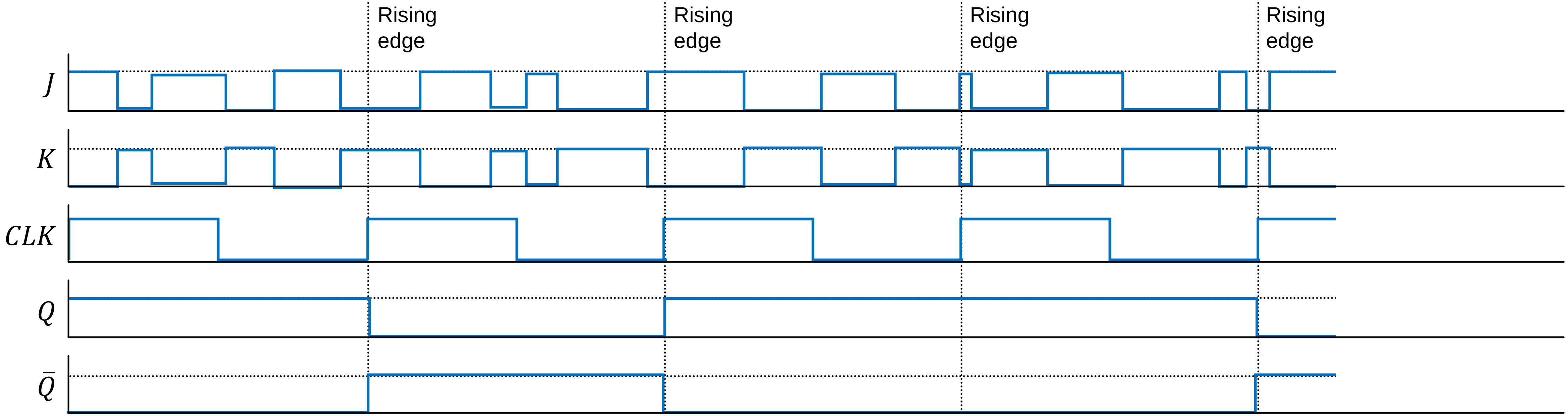




JK Flip-Flop

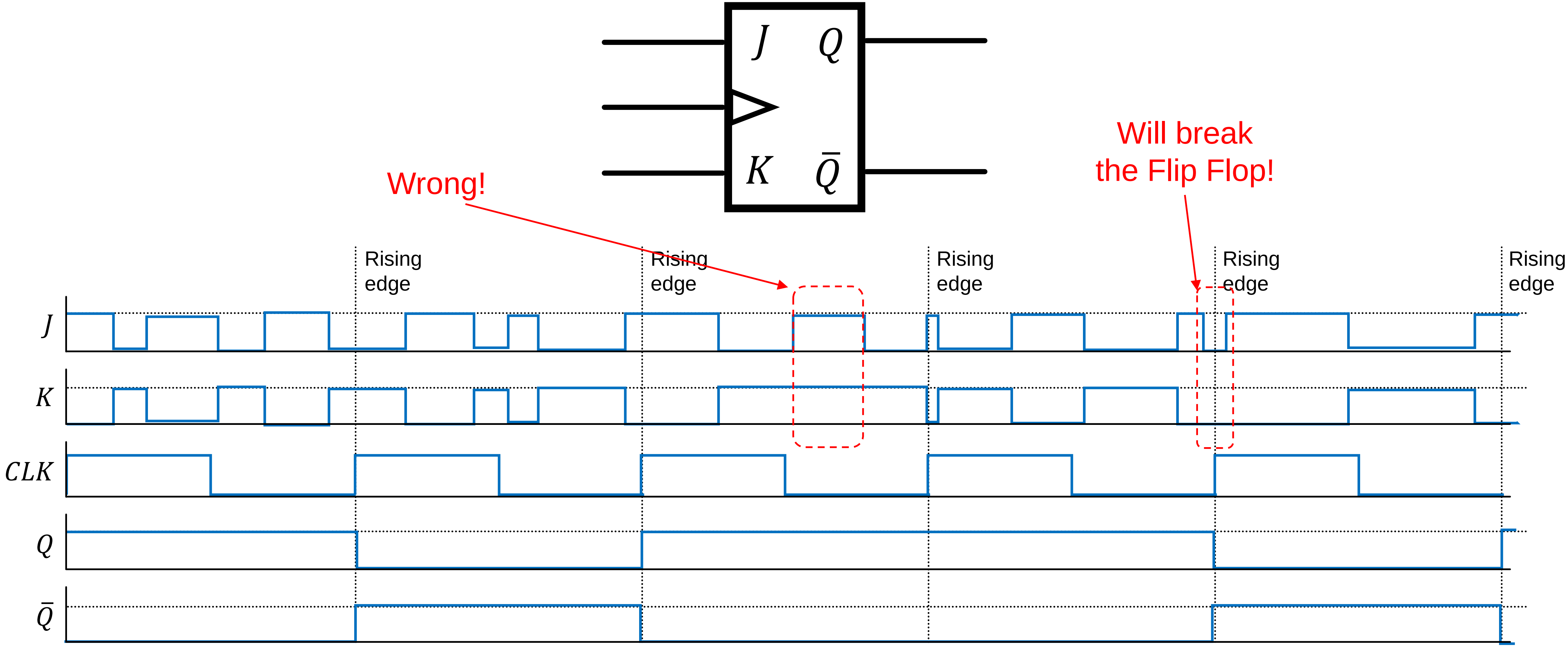
Let us revise how to operate the JK Flip Flop





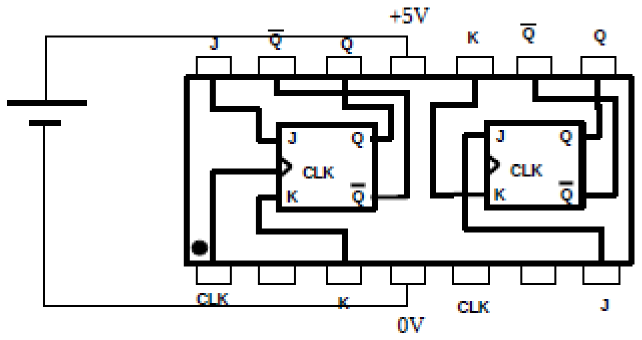
JK Flip-Flop

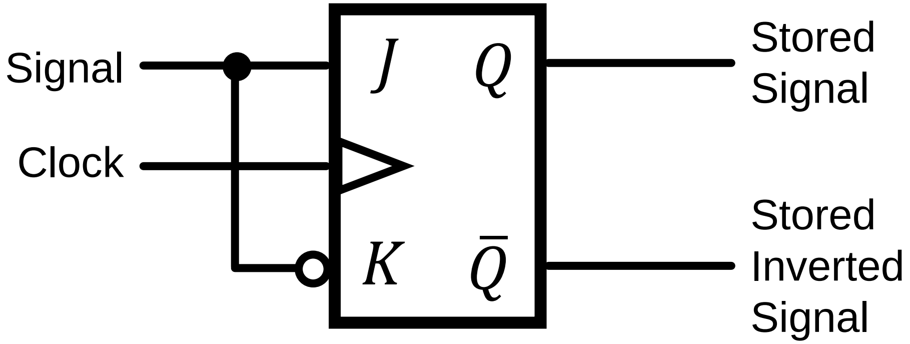
Let us revise how to operate the JK Flip Flop

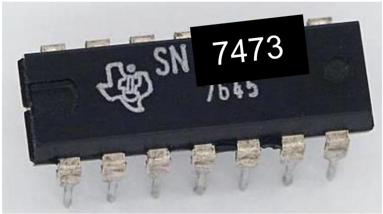


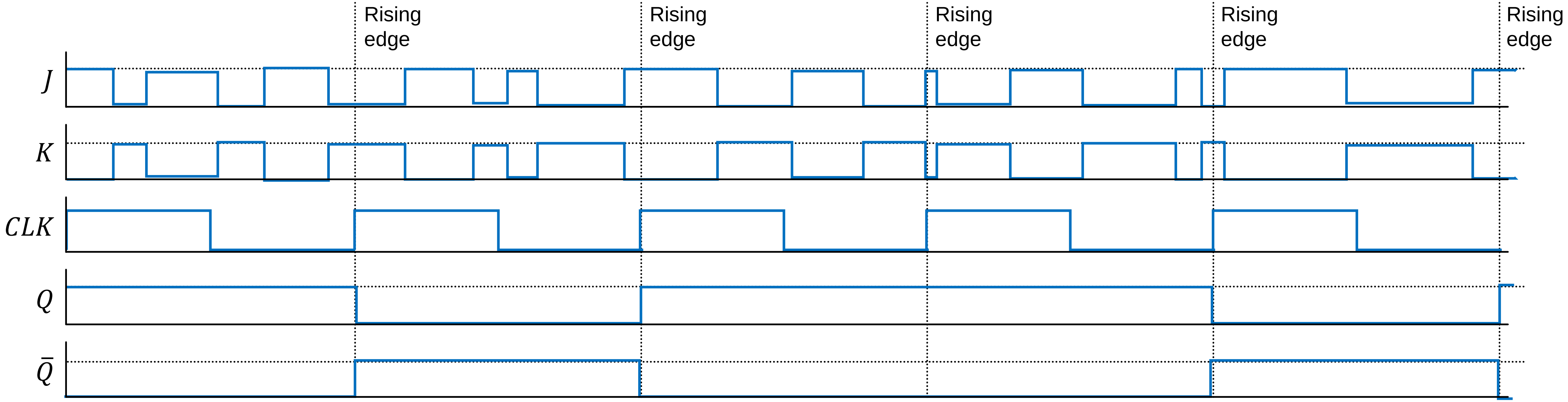
JK Flip-Flop

Let us revise how to operate the JK Flip Flop









What is a "Latch" and how is it different from "Flip-Flop"?

Latch v Flip-Flop

Latch

Does not have any CLOCK or ENABLE signal – *it is **always enabled**!*

Also called:

- **Level-triggered**
- **Asynchronous**
- **Transparent**
- **"Latch"**

Flip-Flop

Needs a periodic CLOCK signal – *it activates on the clock pulse **rising edge***

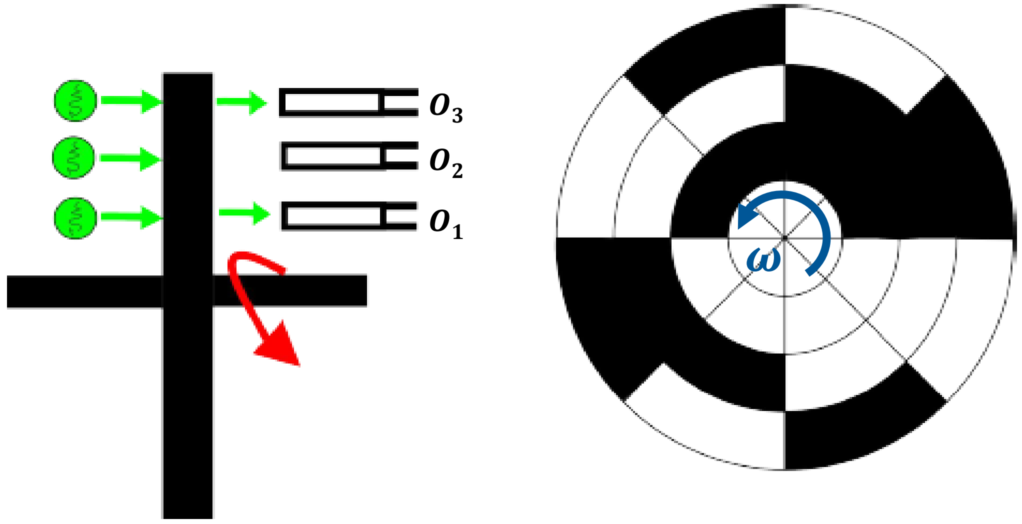
Also called:

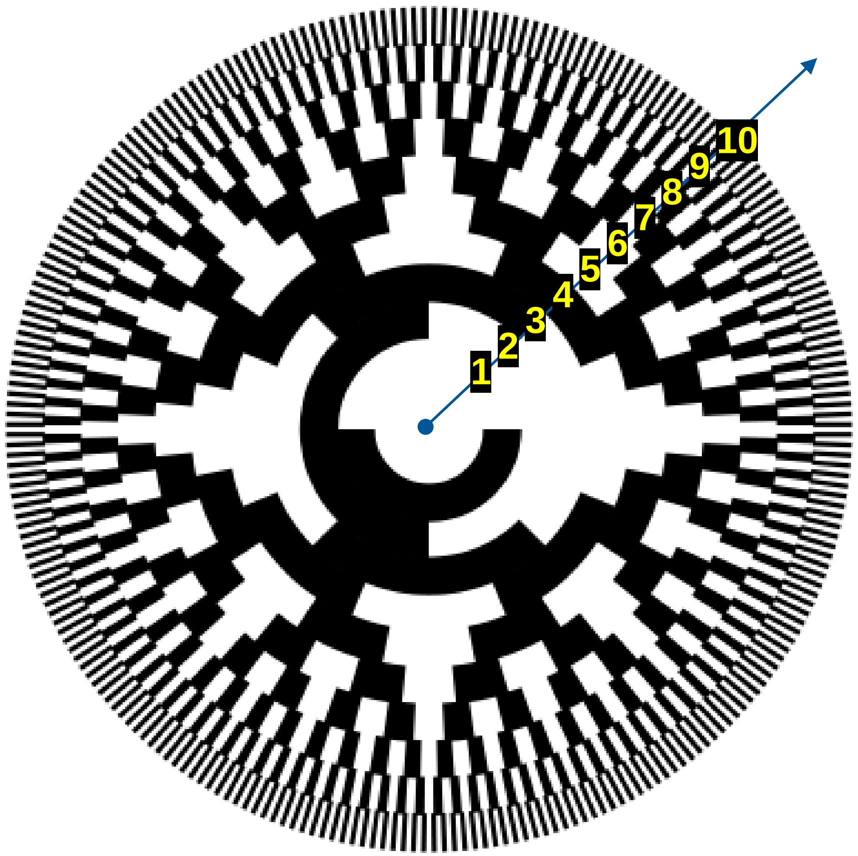
- **Edge-triggered**
- **Synchronous**
- **Opaque**
- **"Flip Flop"**

Learning Outcomes

- Revision of Logic Gates
  - **Shaft Encoder**
- **Flip Flops**
  - Latch v Flip Flop
  - SR/JK/D/T Flip Flops
- Applications of Digital Circuit
  - **Series** v **Parallel** data & conversion (**Bit Shifter**)
  - Analog/Digital conversion (**R-2R Ladder** circuit)
  - **Flash Converter**

Series v Parallel Data





We studied **Shaft Encoder** which uses ***n*-bits** to indicate the angular position

This ***n*-bit "word"** requires ***n* individual copper wires** to transmit the information

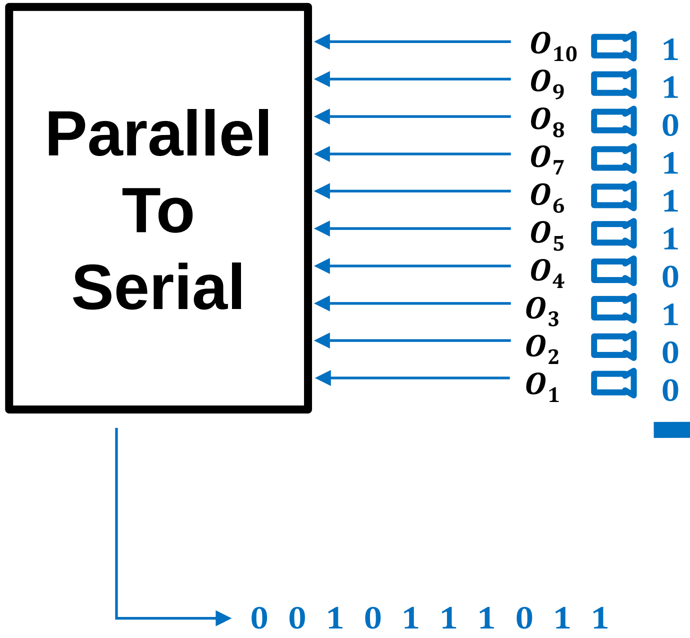
Could we save money here and transmit the same information over **just one wire**?

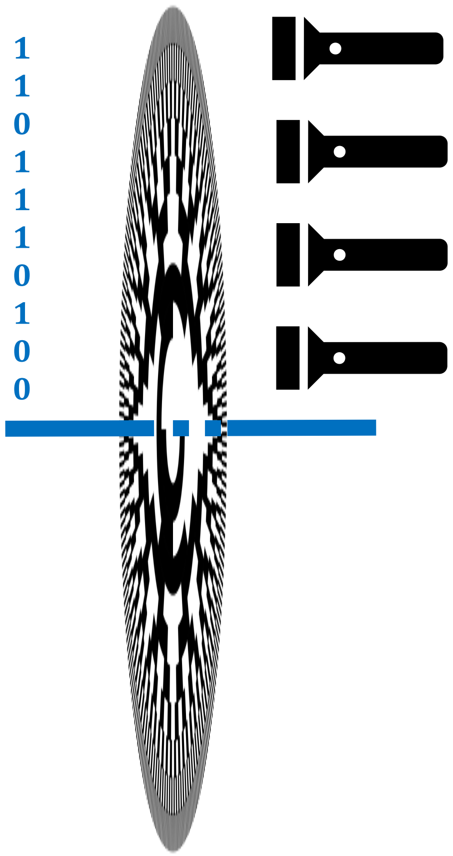
Yes we can! We use what is called:

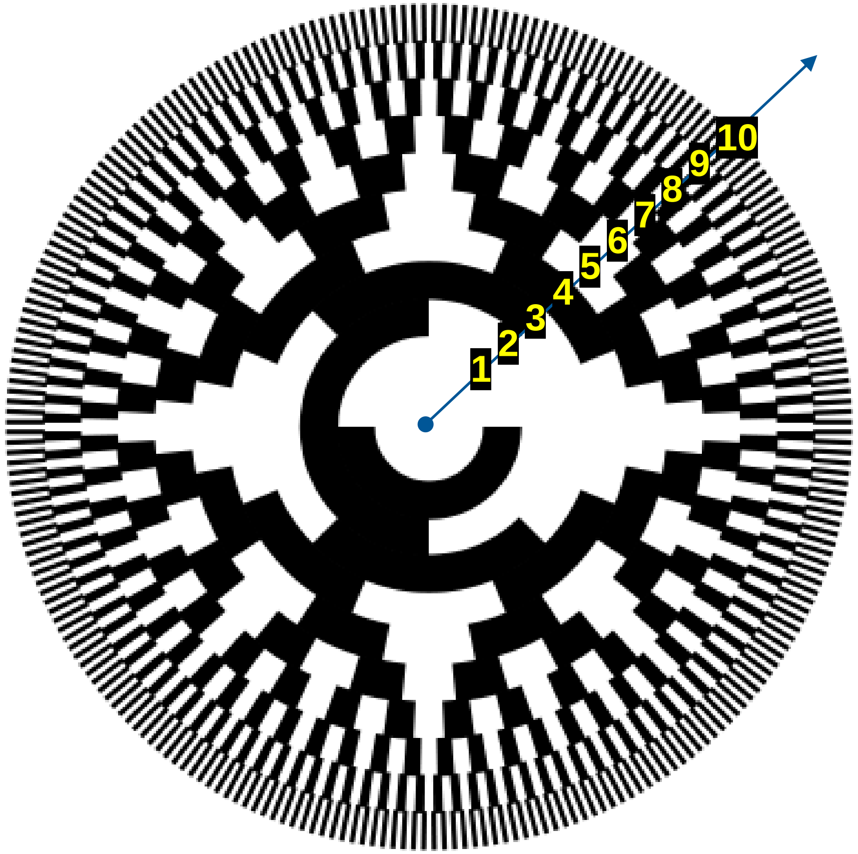
Serial Communication

Read about this absolute encoder – this does not use binary code. Instead, it uses **Gray Code.** The gray code is similar to binary code, but with every "code change", only a single bit inverts. This feature is used in error-checking. https://electronics.stackexchange.com/questions/15481/how-does-a-ball-mouse-know-the-direction

Series v Parallel Data







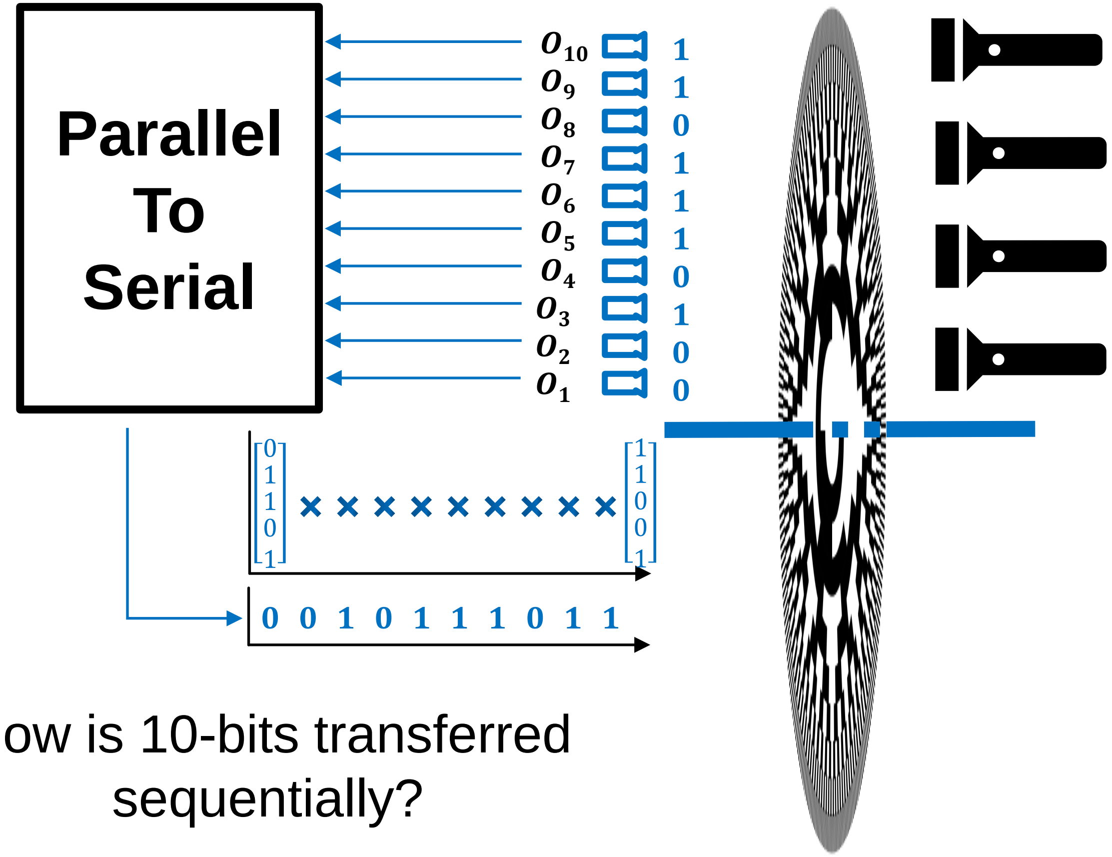
How is 10-bits transferred sequentially?

Time-multiplexing

At every clock pulse, the output signal changes (or "shifts") to the next bit in the sequence

Bit Shifter

Series v Parallel Data



How is 10-bits transferred sequentially?

Time-multiplexing

At every clock pulse, the output signal changes (or "shifts") to the next bit in the sequence

Bit Shifter

Disadvantage of Time Multiplexing

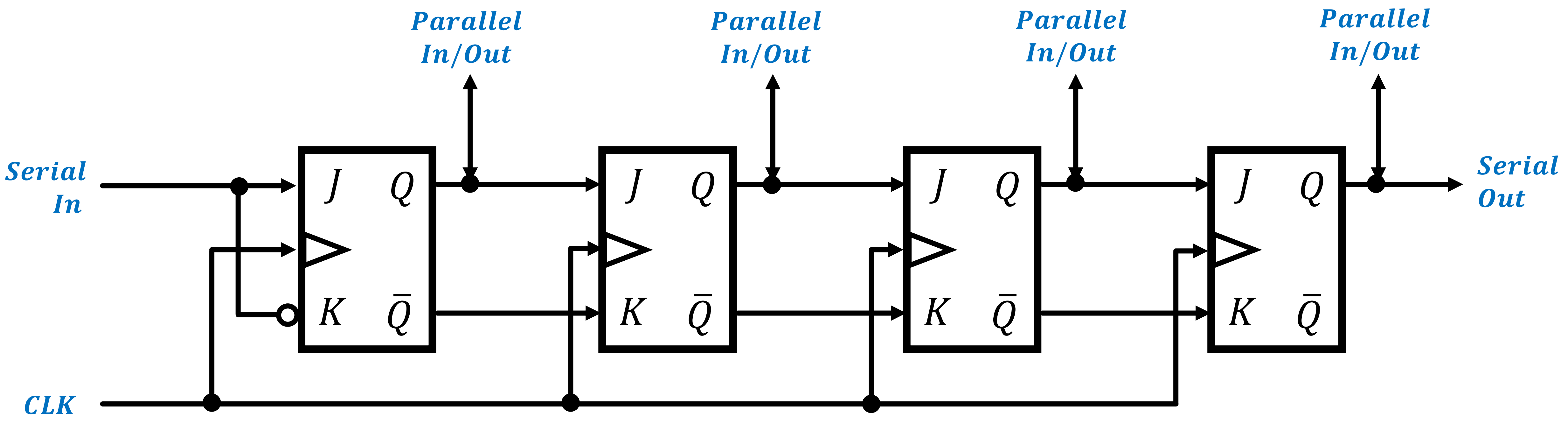
Time resolution gets divided!

If the encoder is producing a new 10-bit word every $1ms$, and your processor clock speed is also $1ms$, you need to sample-and-hold the word at the start, spend $10ms$ to produce the 10 bits sequentially, and then sample-and-hold the next 10-bit word

Effective time resolution of the encoder is now $10ms$ (even though the encoder has a resolution of $1ms$)

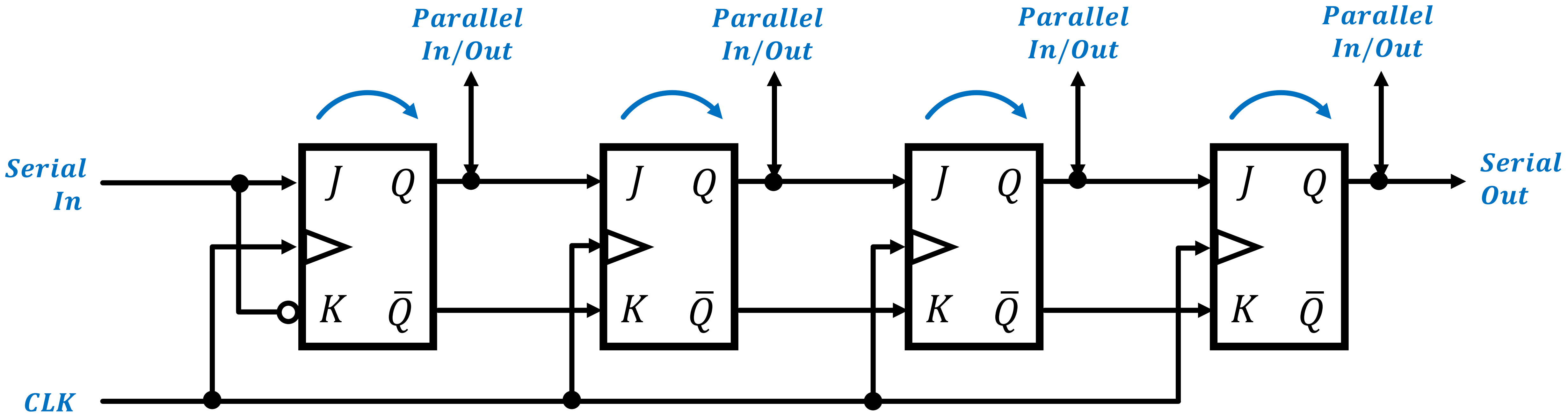
Bit Shifter

General Schematic of a Bit Shifter



Bit Shifter

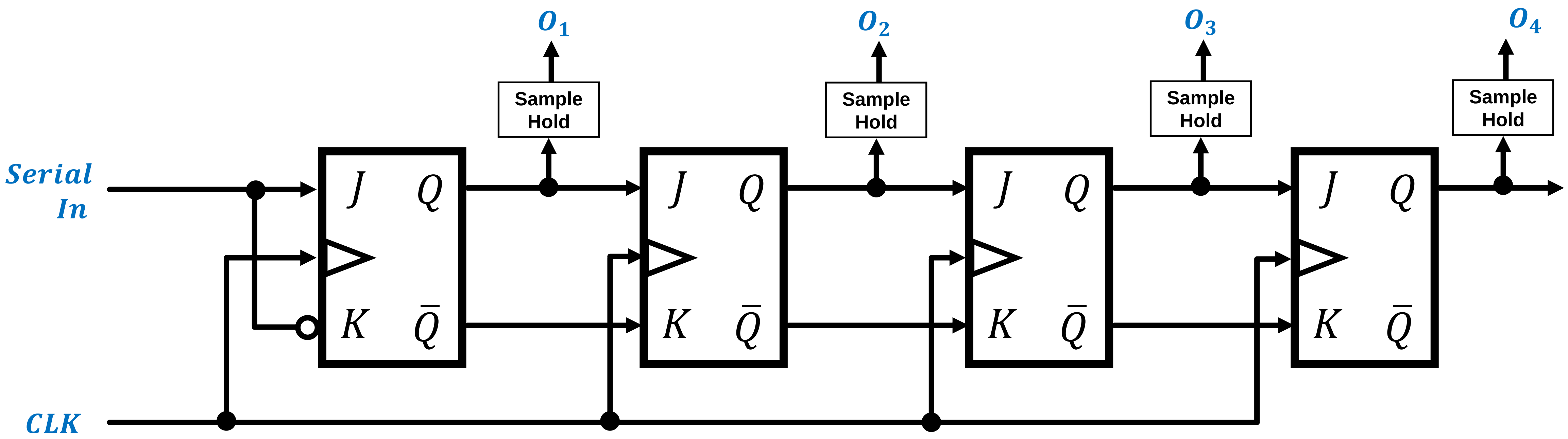
General Schematic of a Bit Shifter

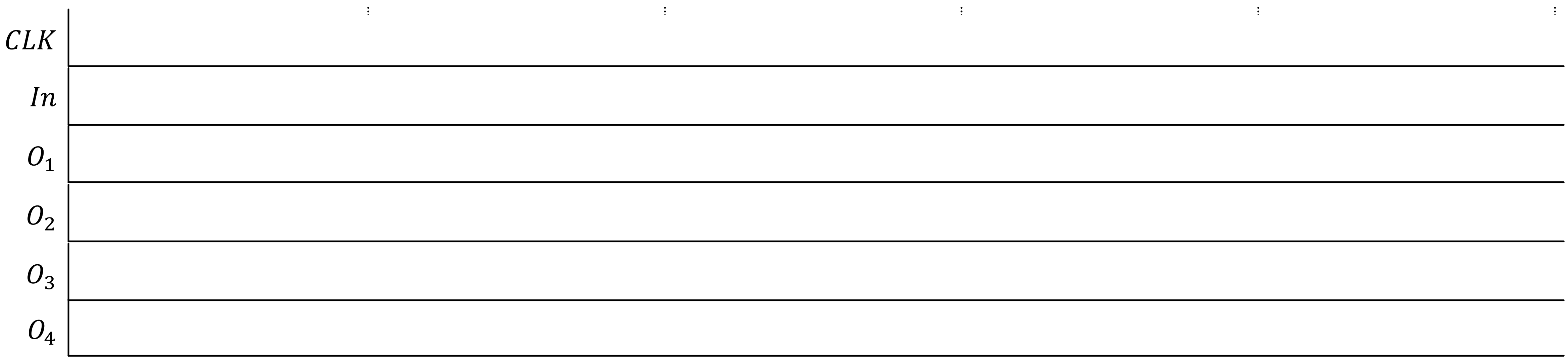


At every rising edge of the clock pulse, the bit gets shifted to the right

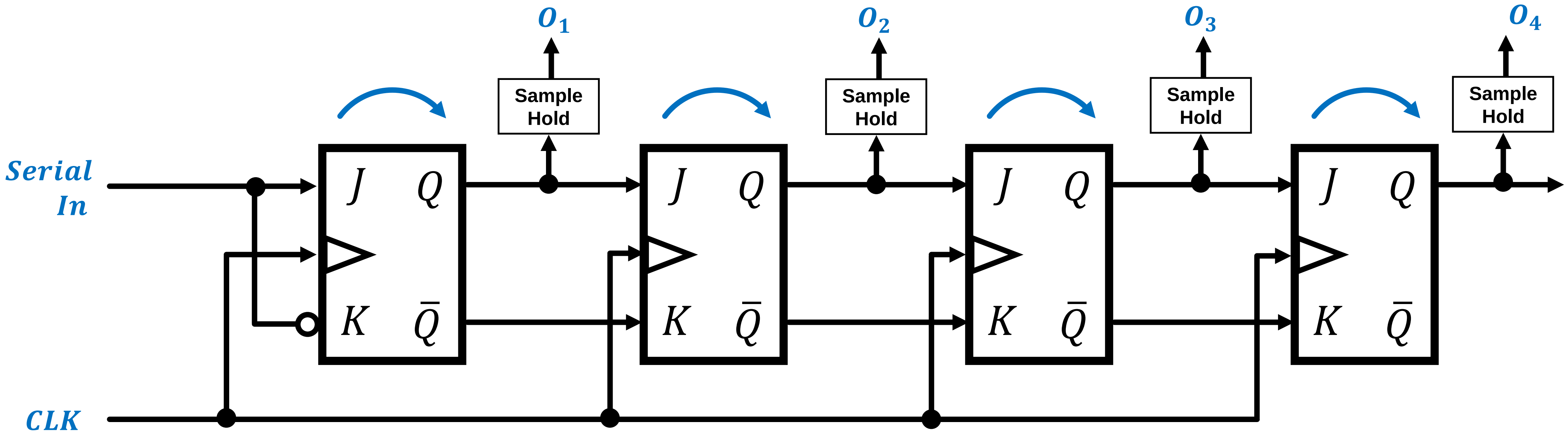
Otherwise, the JK flip flops (arranged in D flip flop configuration) holds their values

Bit Shifter – Serial-to-Parallel Conversion

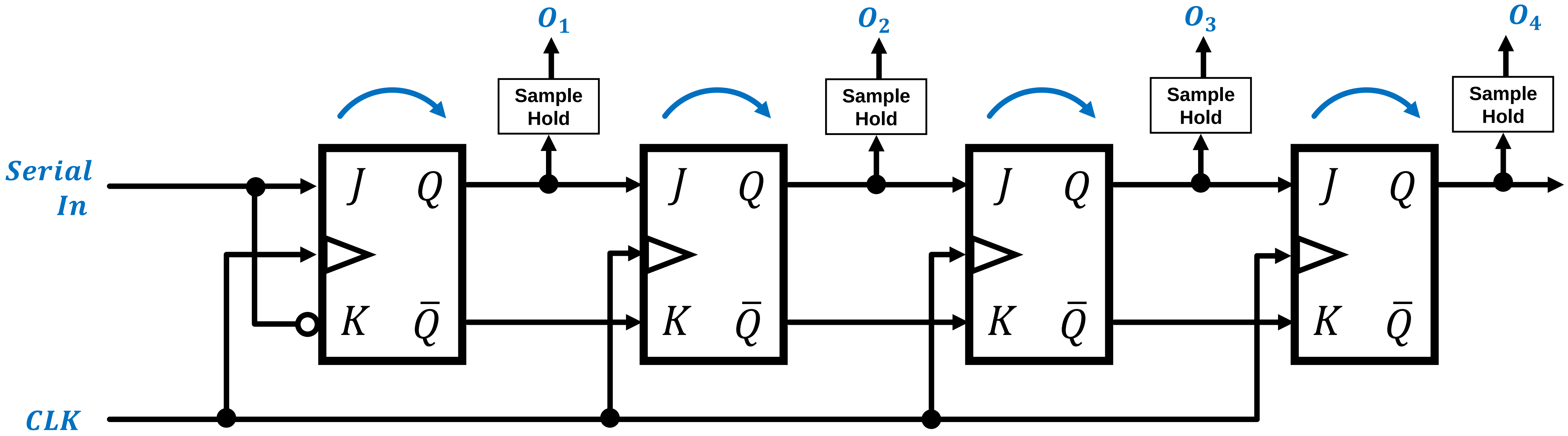




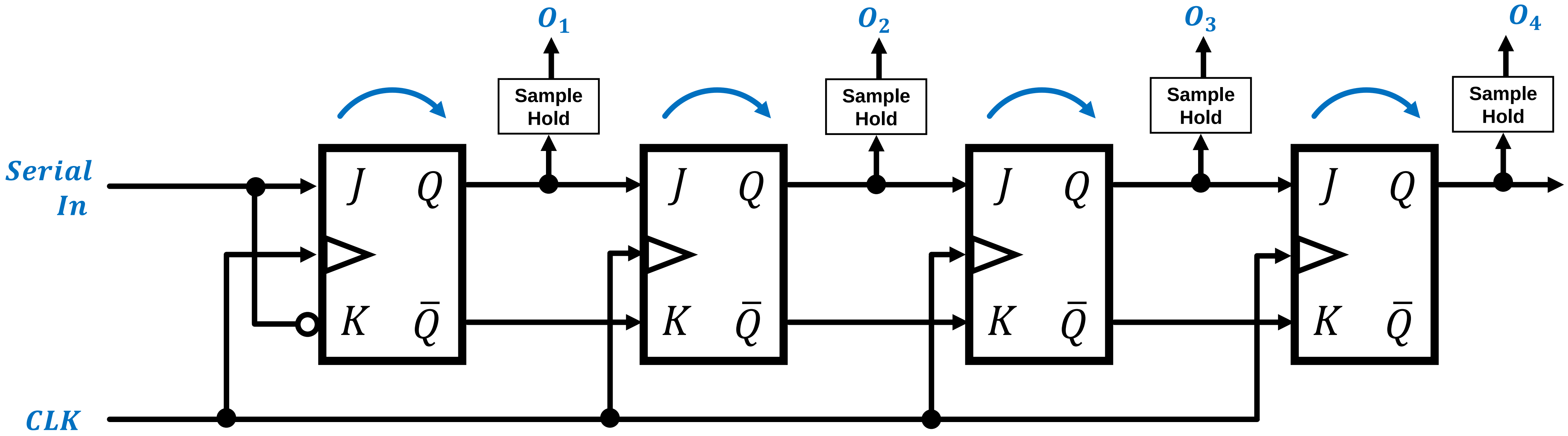
Bit Shifter – Serial-to-Parallel Conversion

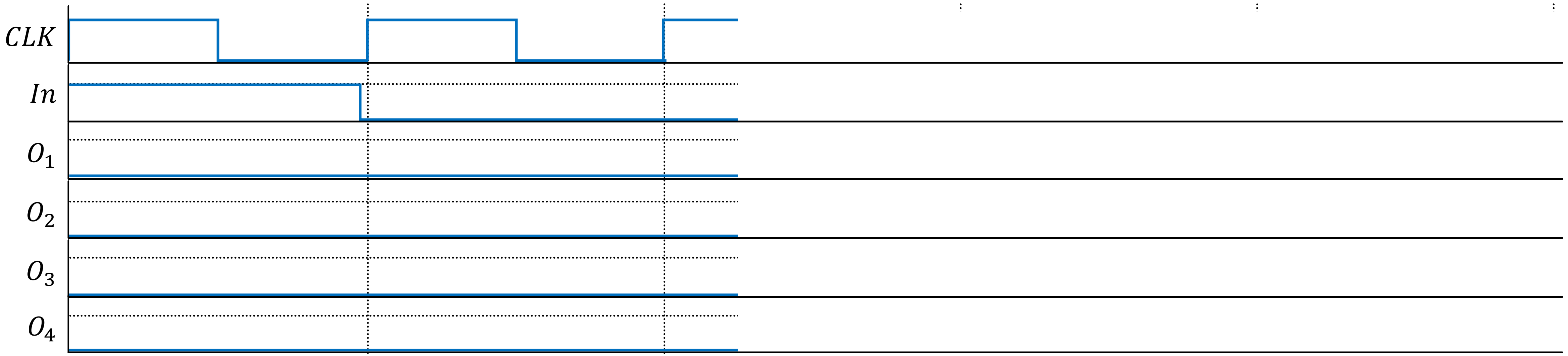


Bit Shifter – Serial-to-Parallel Conversion

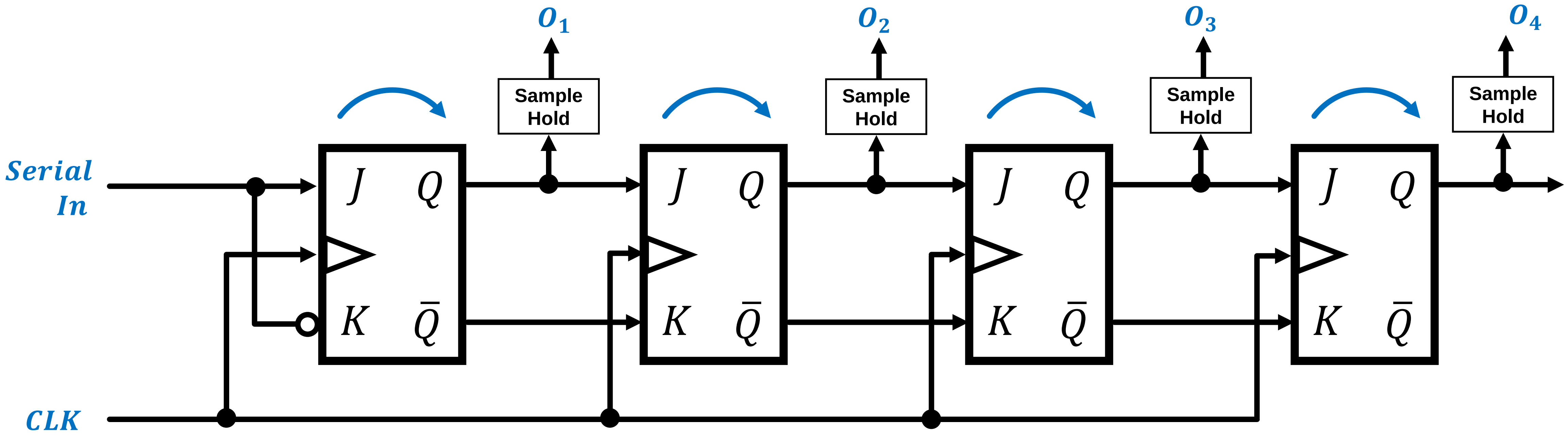


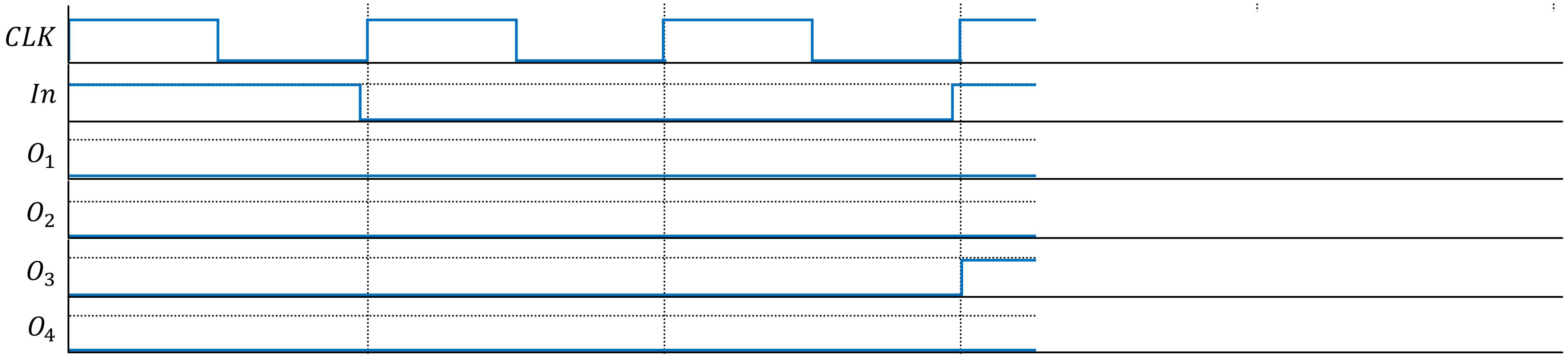
Bit Shifter – Serial-to-Parallel Conversion



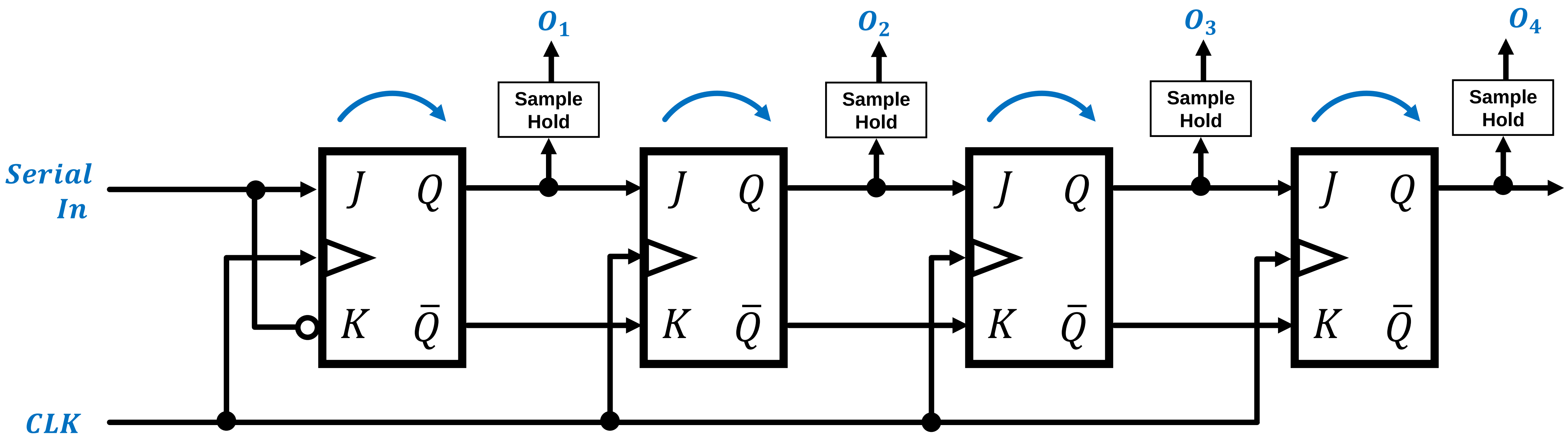


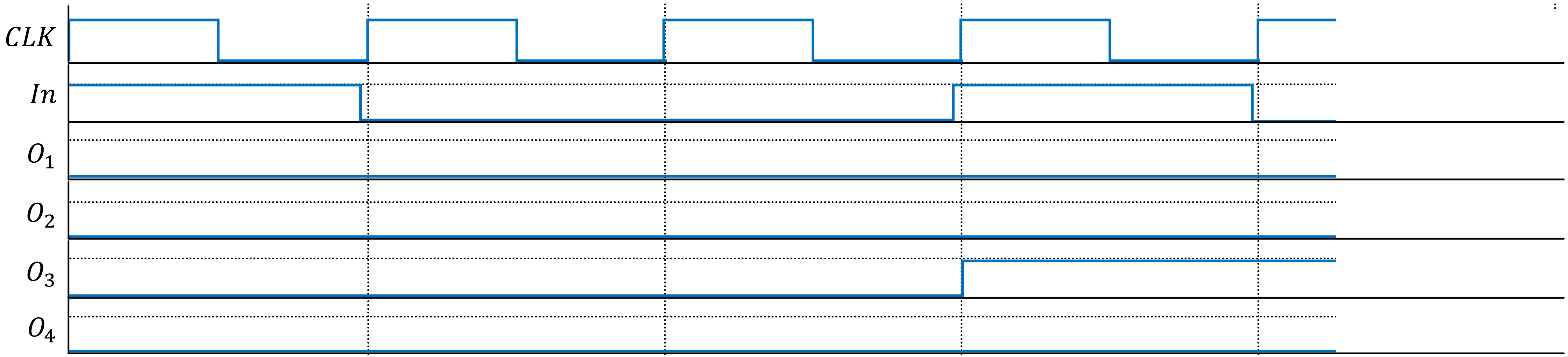
Bit Shifter – Serial-to-Parallel Conversion



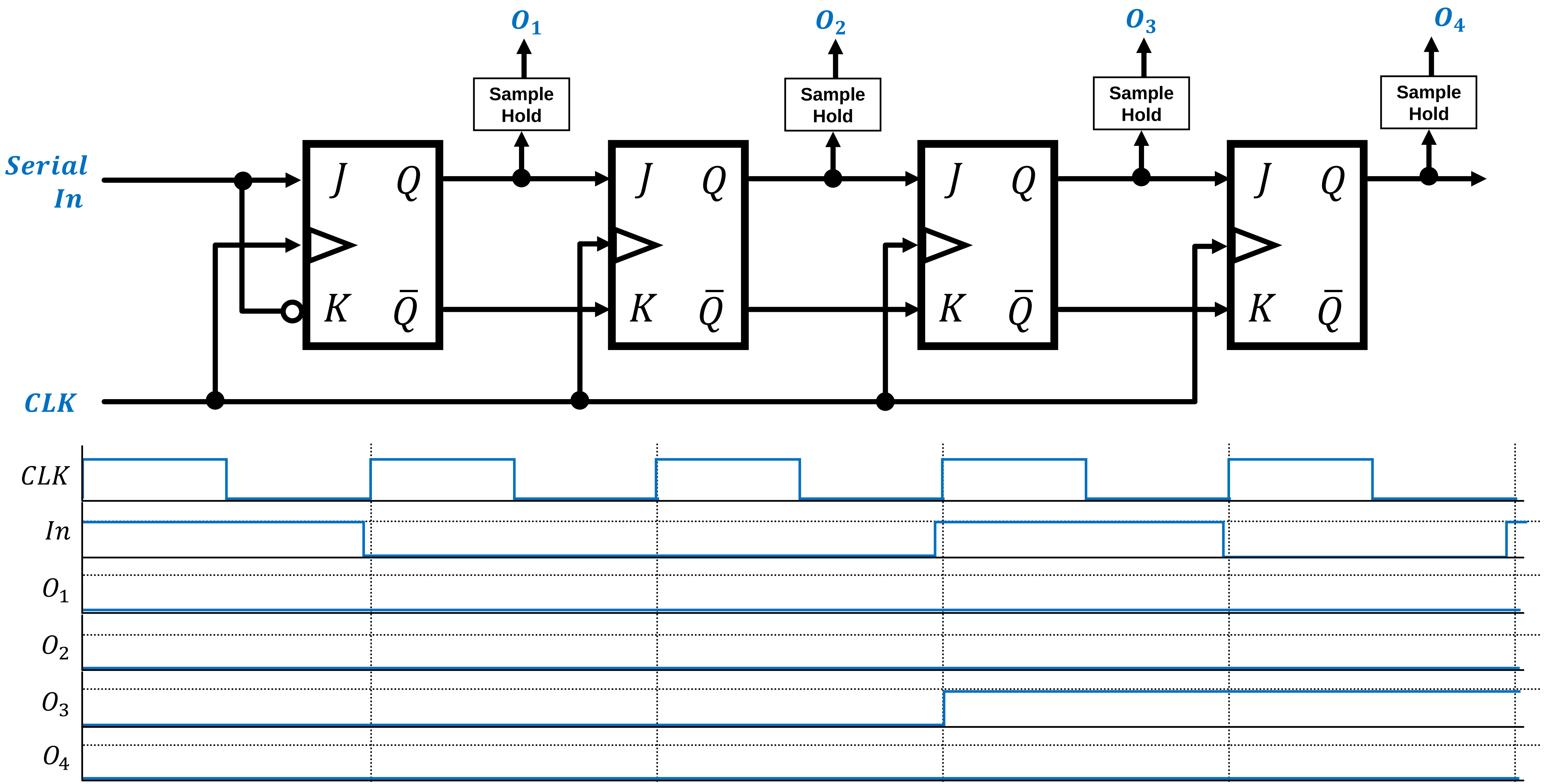


Bit Shifter – Serial-to-Parallel Conversion

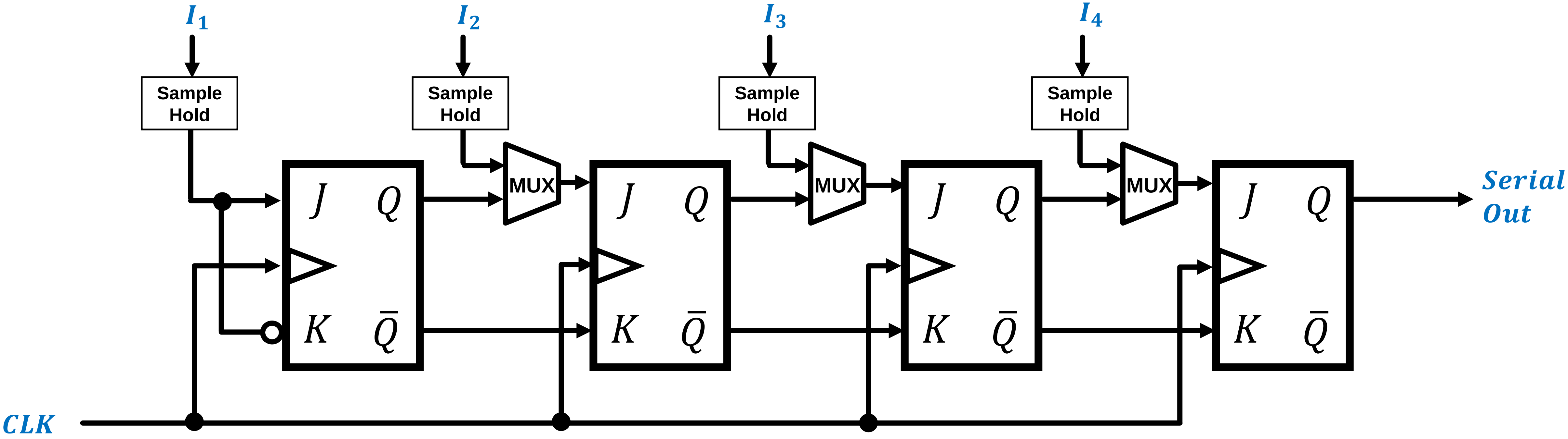


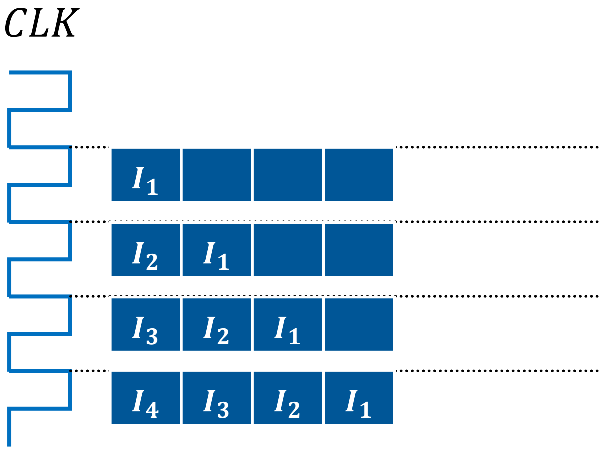


Bit Shifter – Serial-to-Parallel Conversion



Bit Shifter – Parallel-to-Serial Conversion





Parallel-to-serial conversion process is bit more detailed (we will not cover this in the module)

Multiplexer does a **bit-shift operation** at every clock pulse

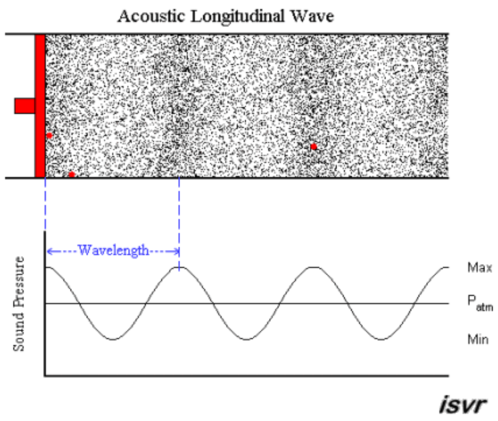
Final output is a ***n*-bit word**

Summary

- Revision of Logic Gates
  - **Shaft Encoder**
- **Flip Flops**
  - Latch v Flip Flop
  - SR/JK/D/T Flip Flops
- Applications of Digital Circuit
  - **Series** v **Parallel** data & conversion (**Bit Shifter**)
  - Analog/Digital conversion (**R-2R Ladder** circuit)
  - **Flash Converter**

Digital-Analog Conversion

Typical application of inter-conversion between analog and digital signals



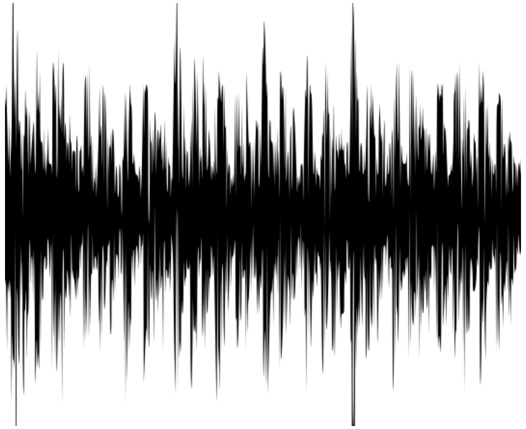
Air pressure waves gets picked up by a diaphragm in the mic

Diaphragm motion produces a small analog voltage signal proportional to the sound pressure waveform

ADC soundcard in the PC converts the analog waveform into digital signal

User does all kinds of processing to the sound digitally

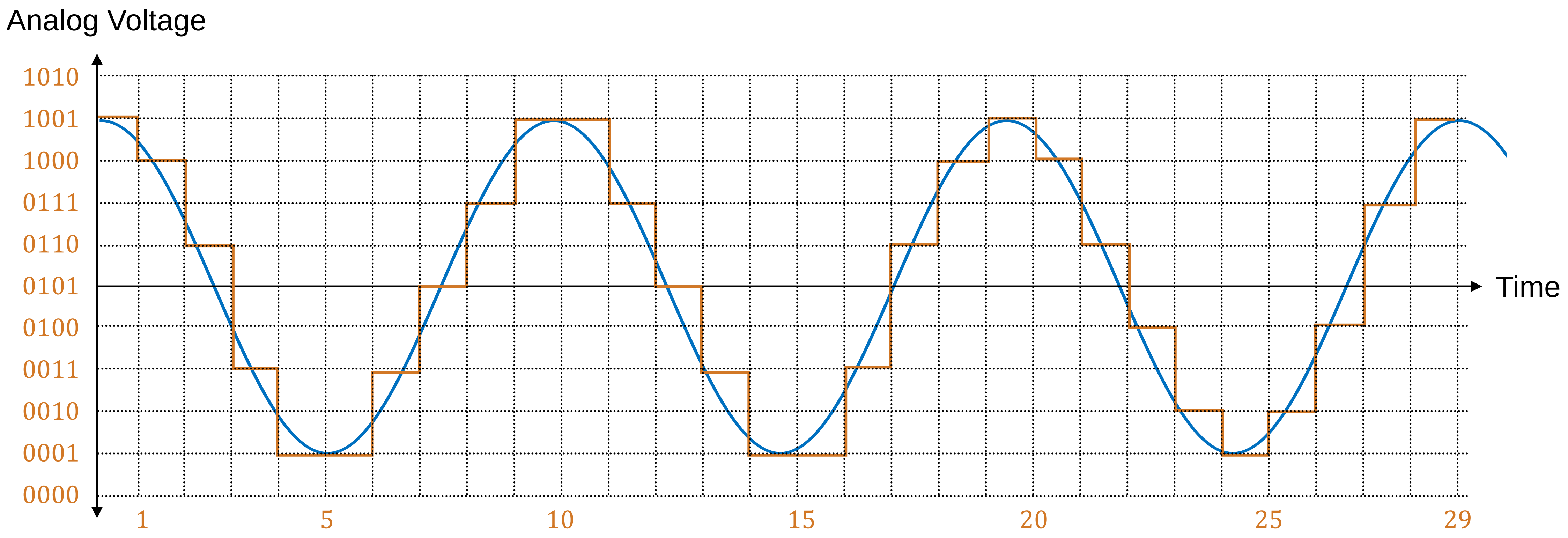
DAC produces an analog signal that can be read by speakers/amplifiers



Speakers convert the analog voltage signal into sound waves again

Digital-Analog Conversion

Digitalisation is simply recording **discreet values** at **discreet time intervals**

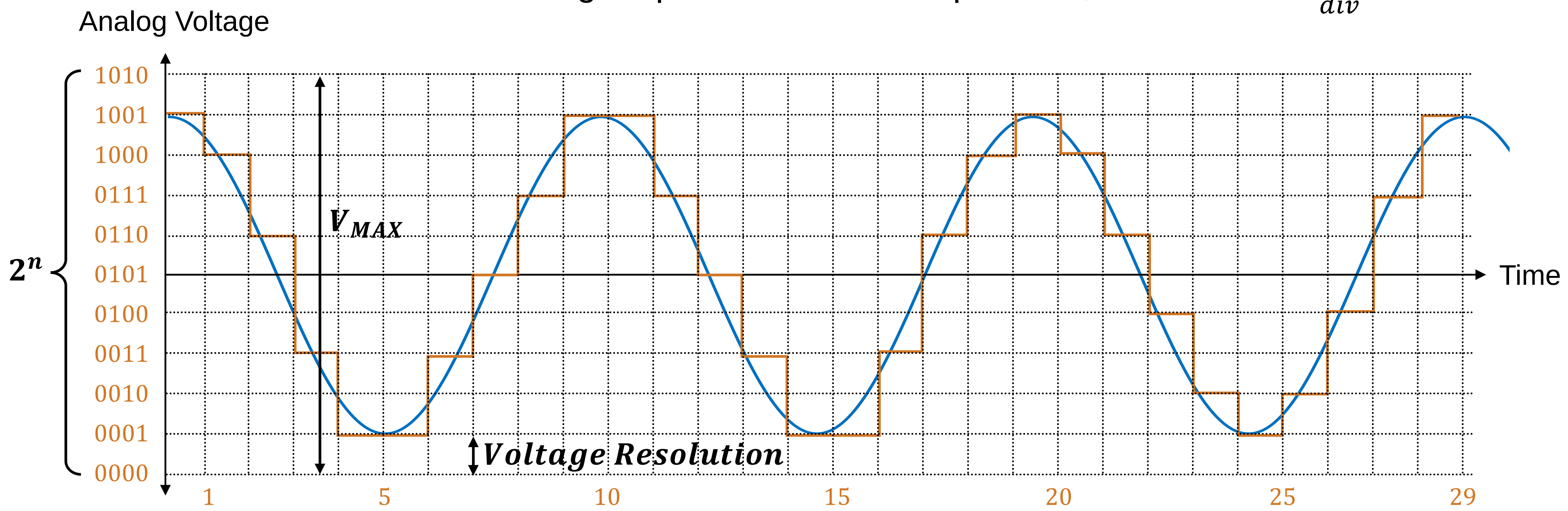


A **faster clock** would allow **finer divisions** on the **time** axis

A **larger word length** (n-bit) would allow **finer divisions** on the **amplitude** axis

What is Resolution?

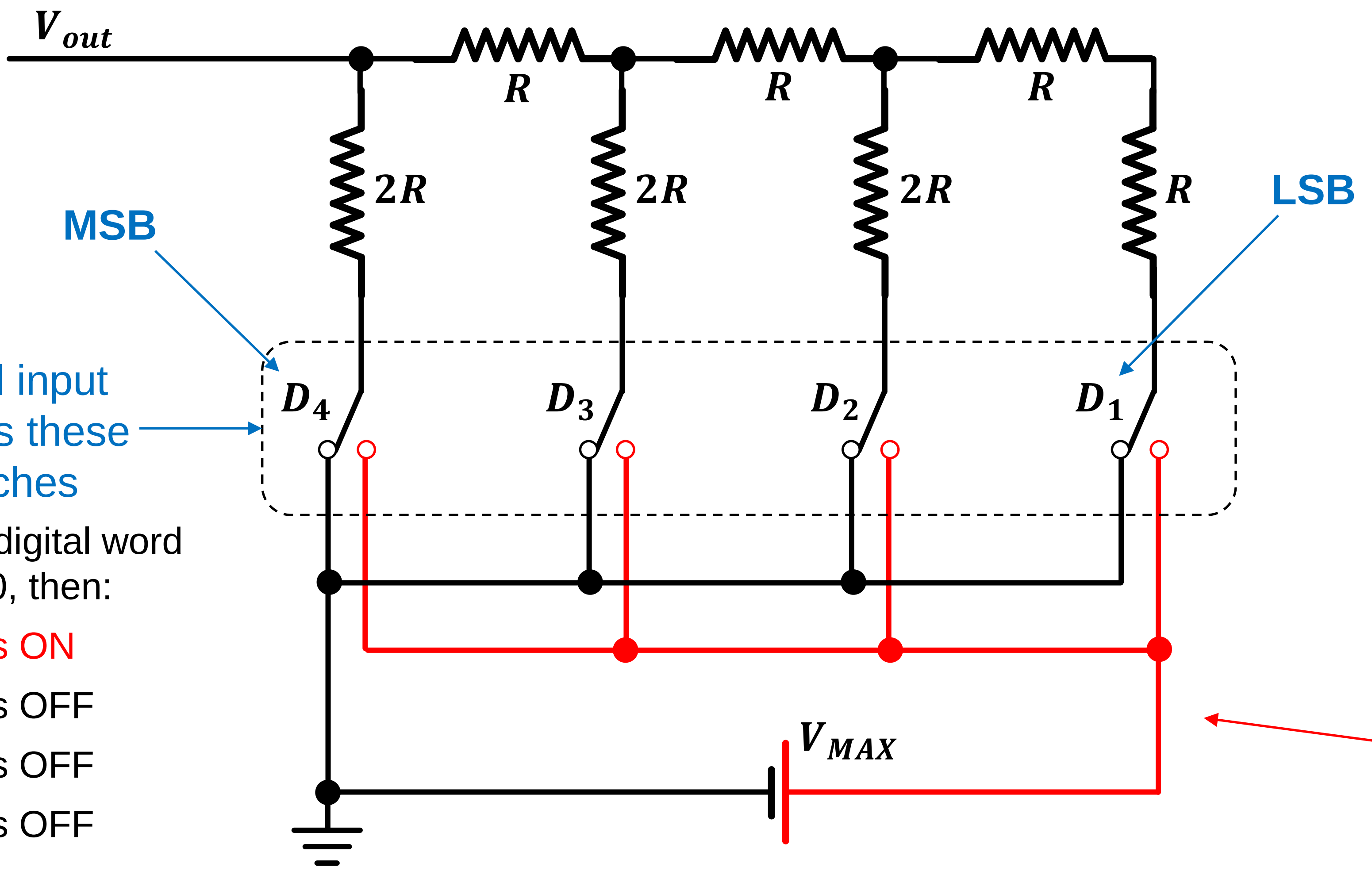
The smallest analog output the device can produce, measured in $\frac{units}{div}$



$$Voltage\ Resolution = \frac{V_{MAX}}{2^n}\frac{volts}{div}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



Digital input controls these switches

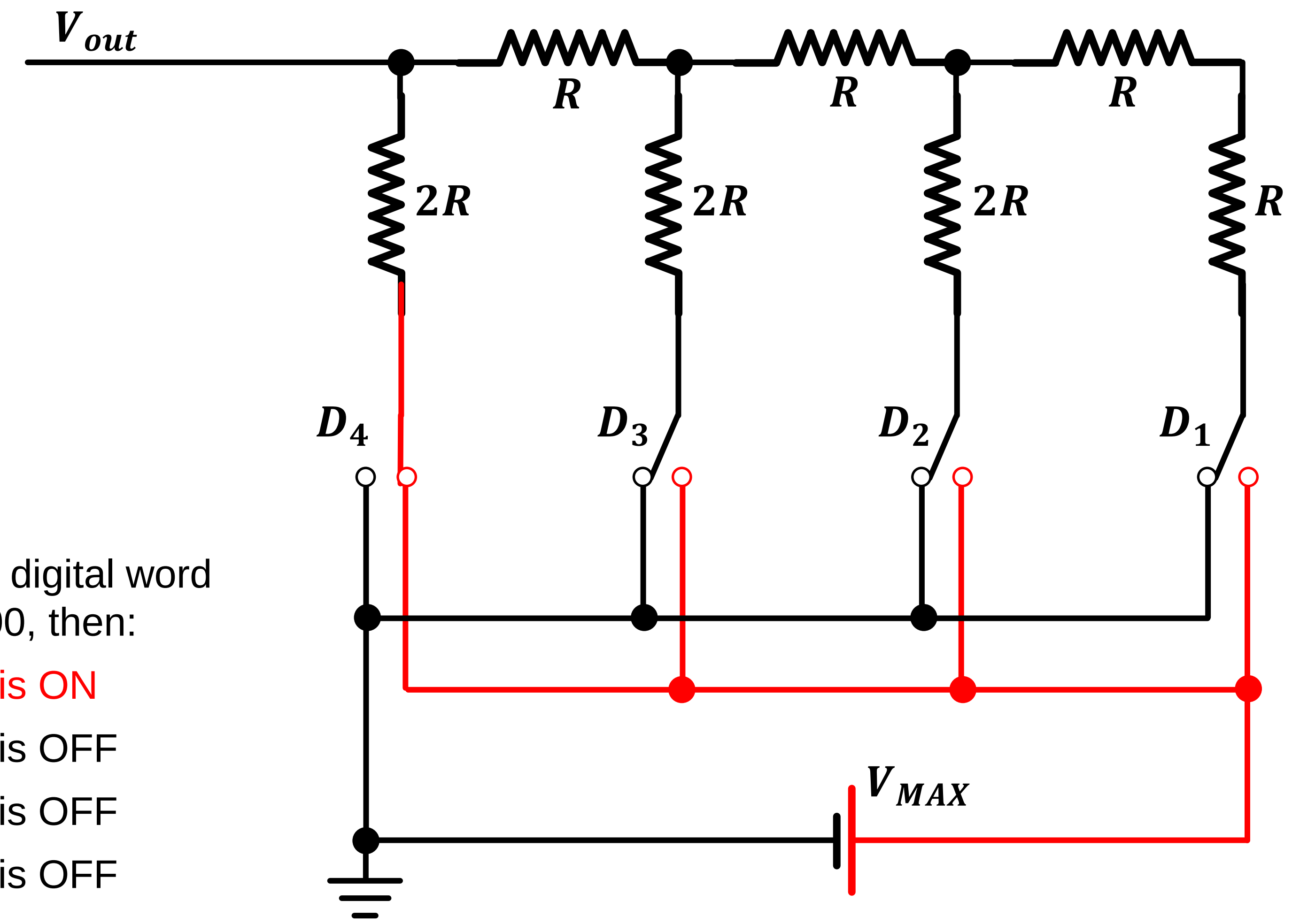
e.g., if the digital word is 1000, then:

- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Note: Positive side is drawn in red to aid understanding

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 1000, then:

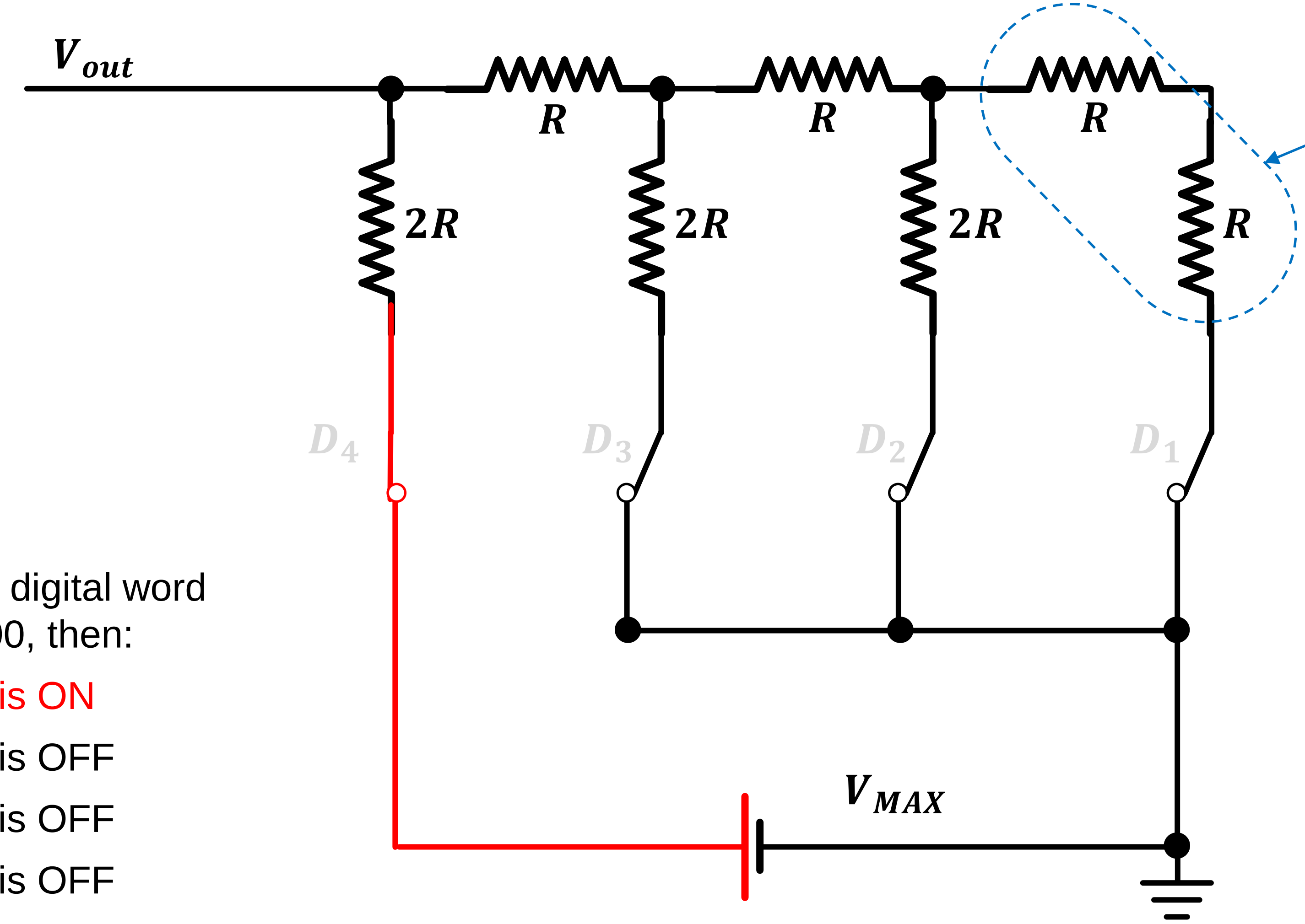
- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Let us try to see how the R-2R Ladder circuit works

Simplifying this circuit:

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



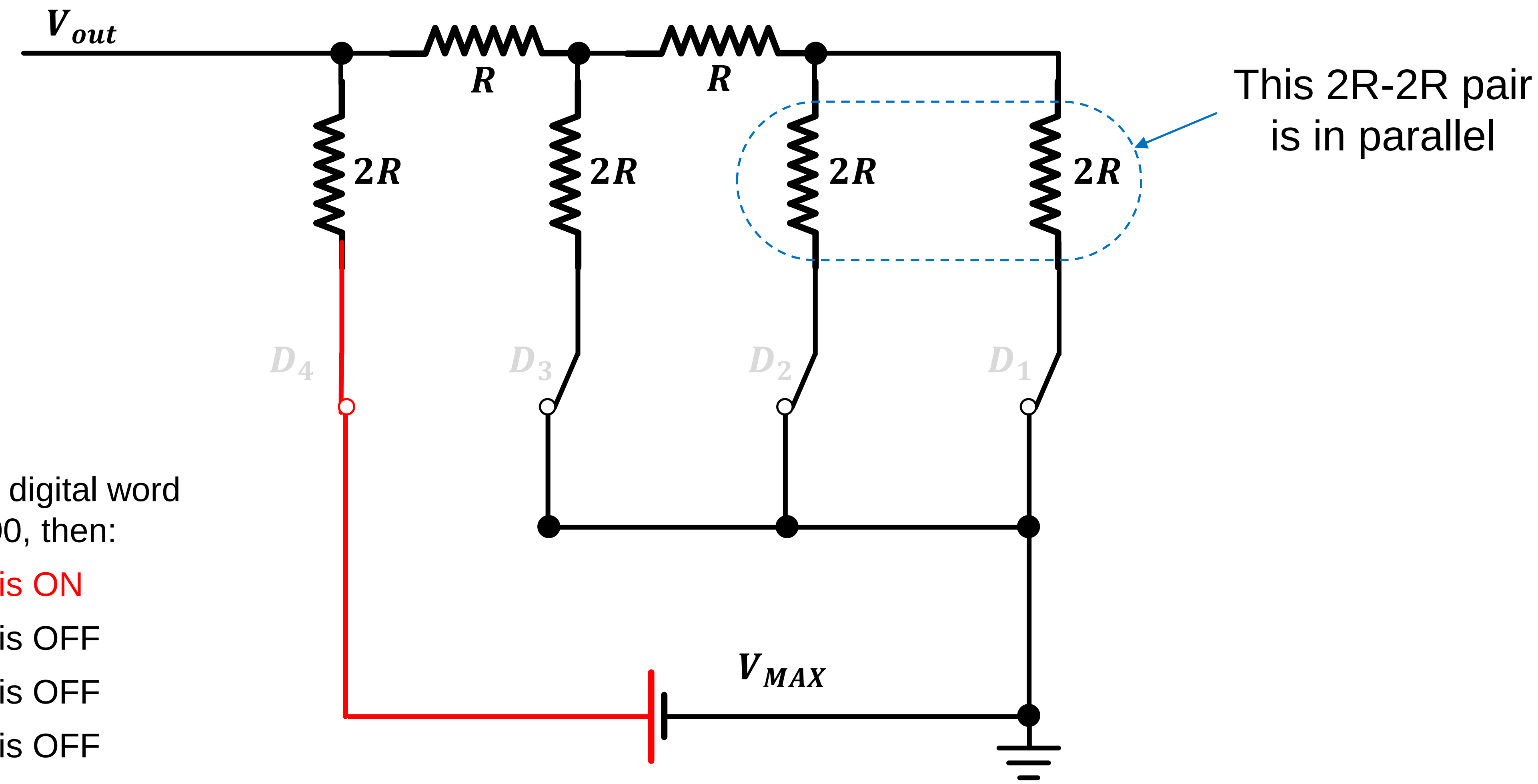
This R-R pair is in series

e.g., if the digital word is 1000, then:

- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

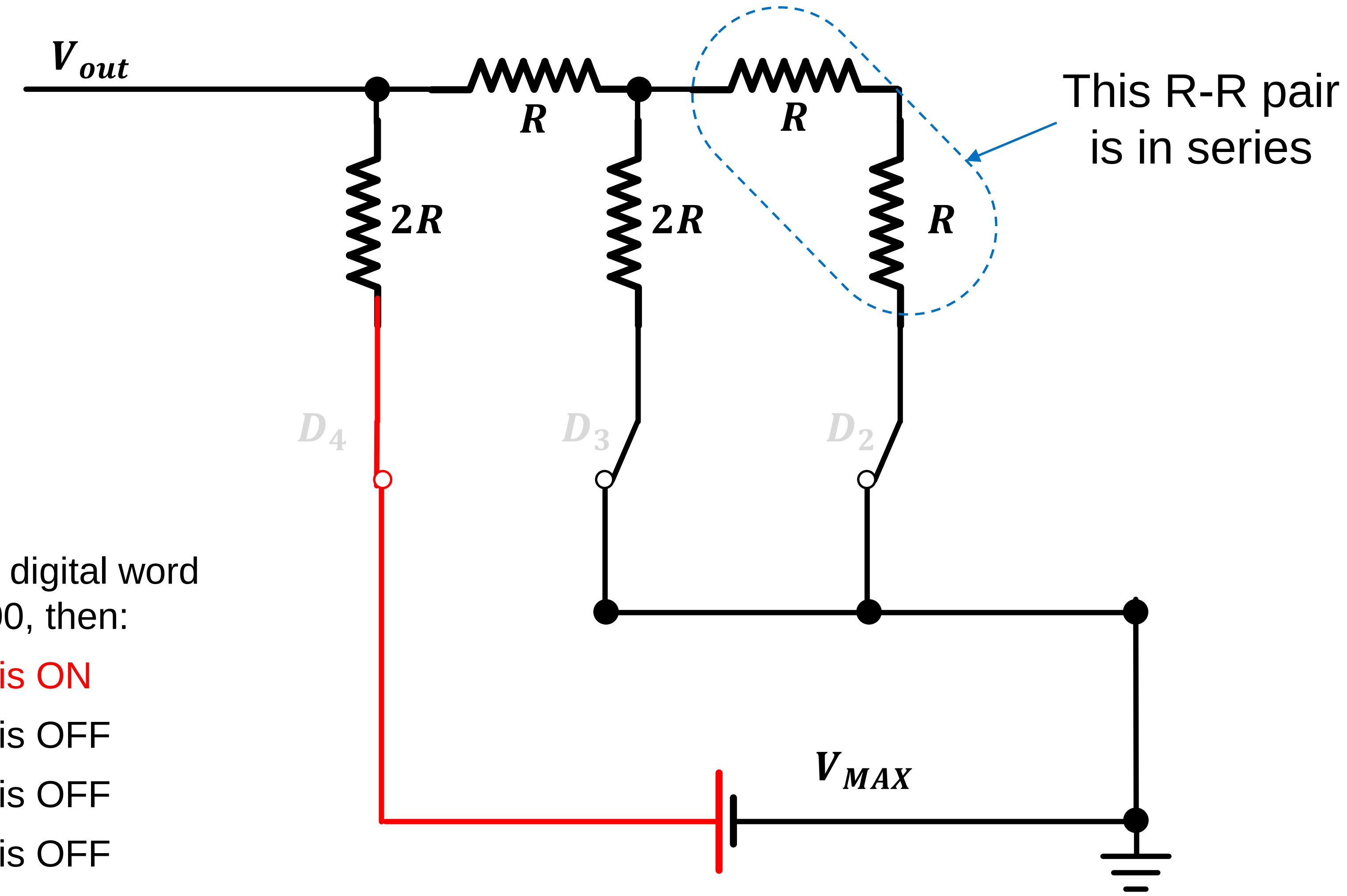


e.g., if the digital word is 1000, then:

- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

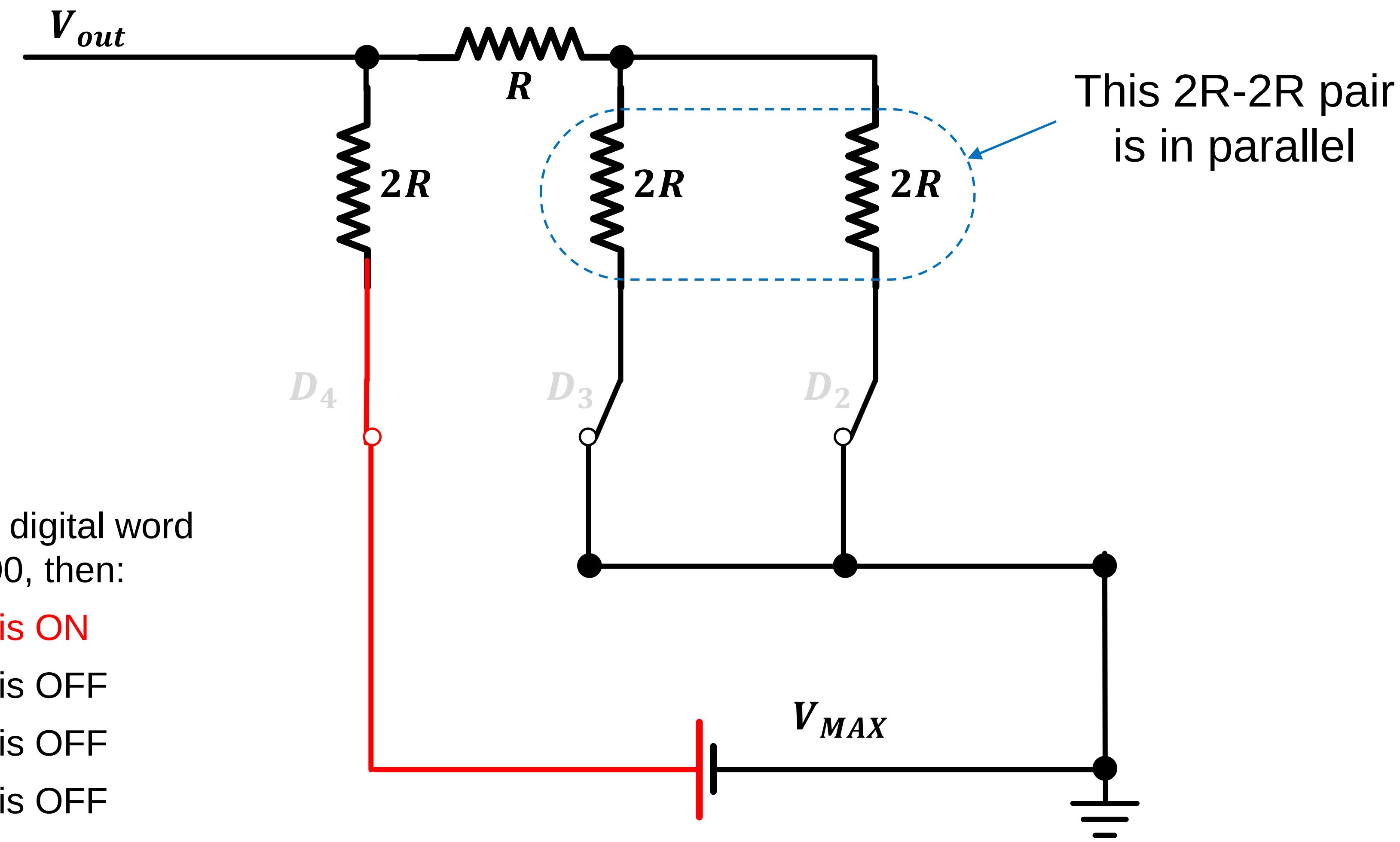
R-2R Ladder Circuit



e.g., if the digital word is 1000, then:

- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

R-2R Ladder Circuit

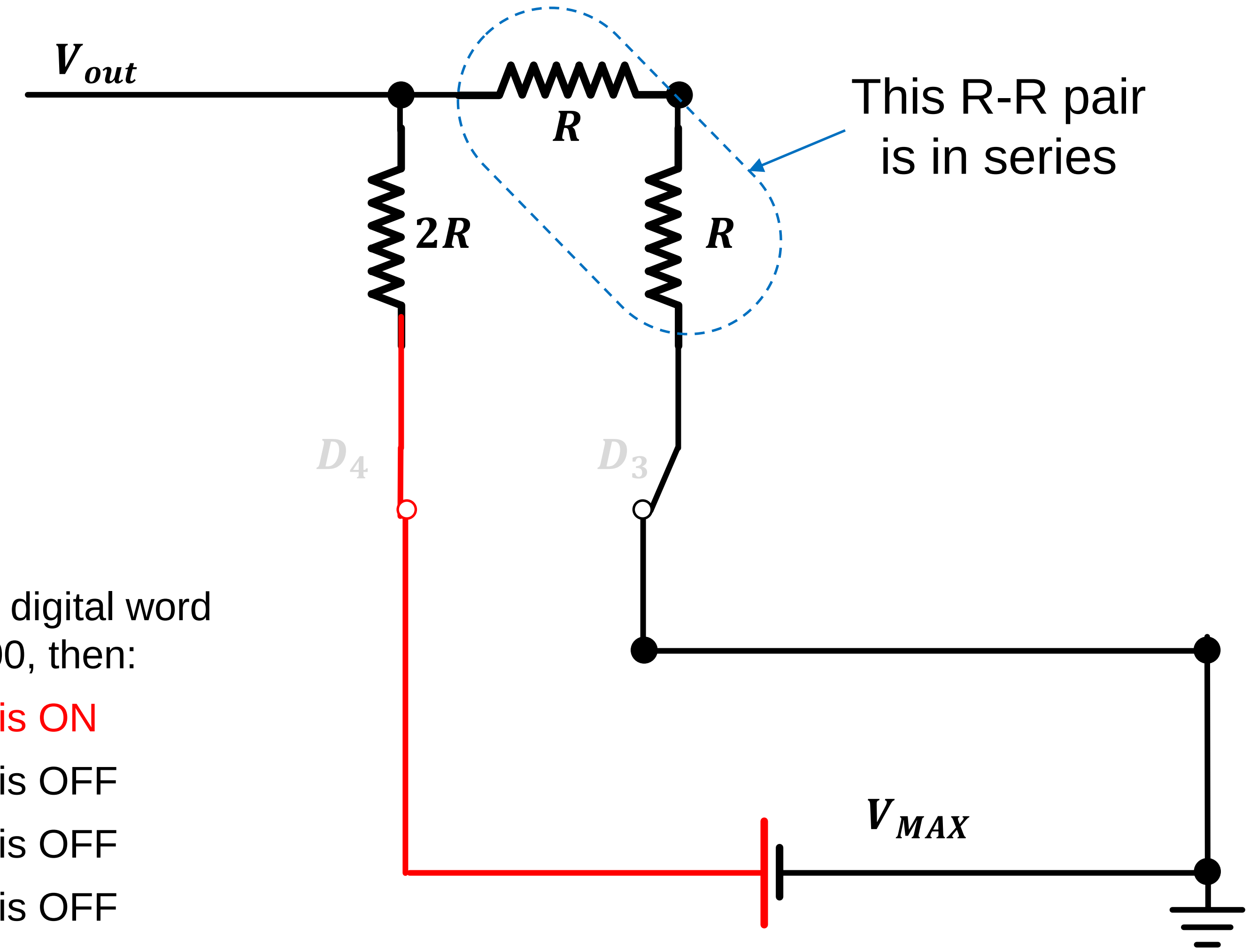


e.g., if the digital word is 1000, then:

- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

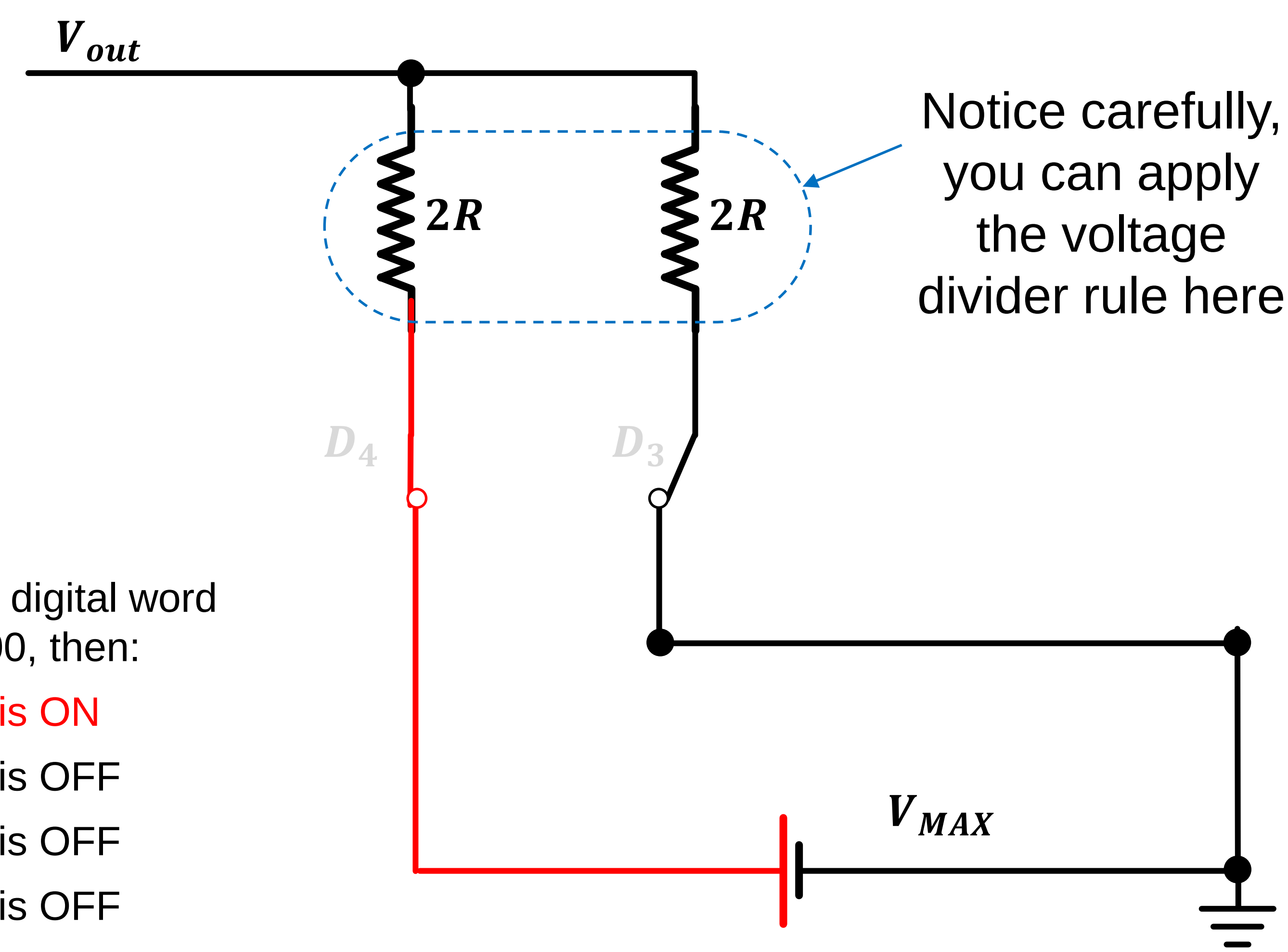


e.g., if the digital word is 1000, then:

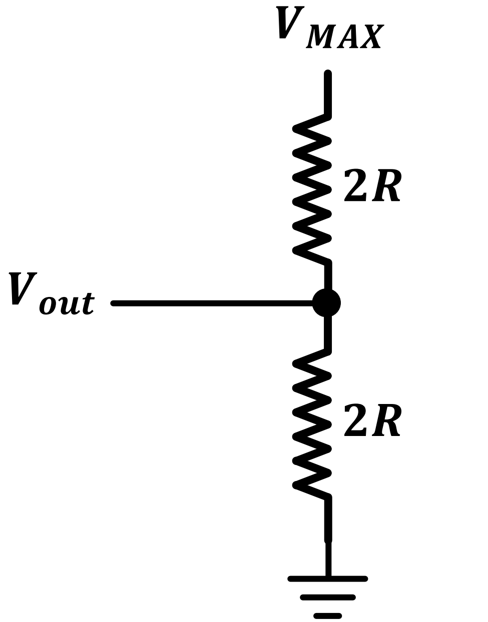
- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



Notice carefully, you can apply the voltage divider rule here



e.g., if the digital word is 1000, then:

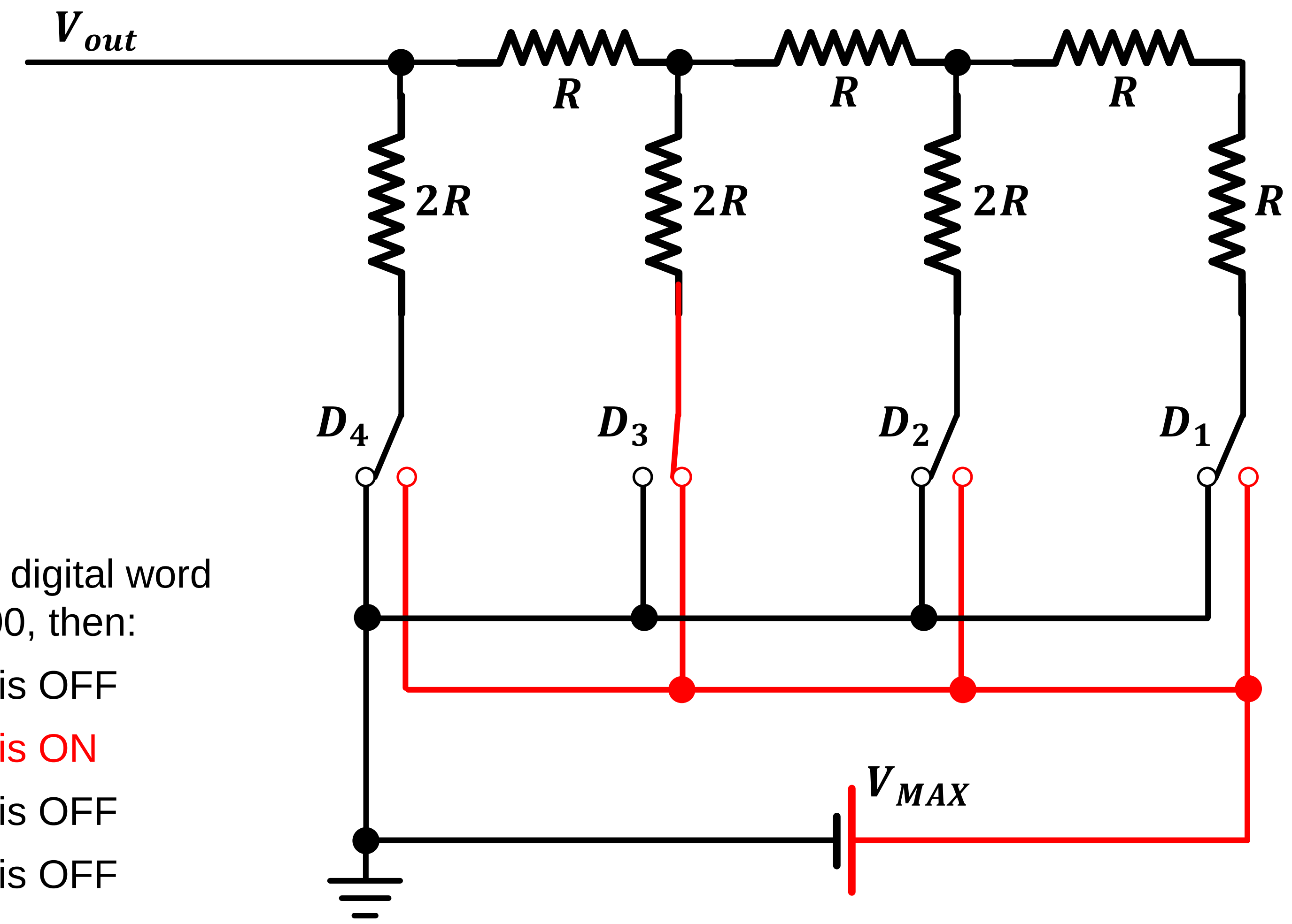
- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

$$V_{out} = \frac{2R}{2R + 2R} V_{MAX}$$

$$V_{out} = \frac{1}{2} V_{MAX}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

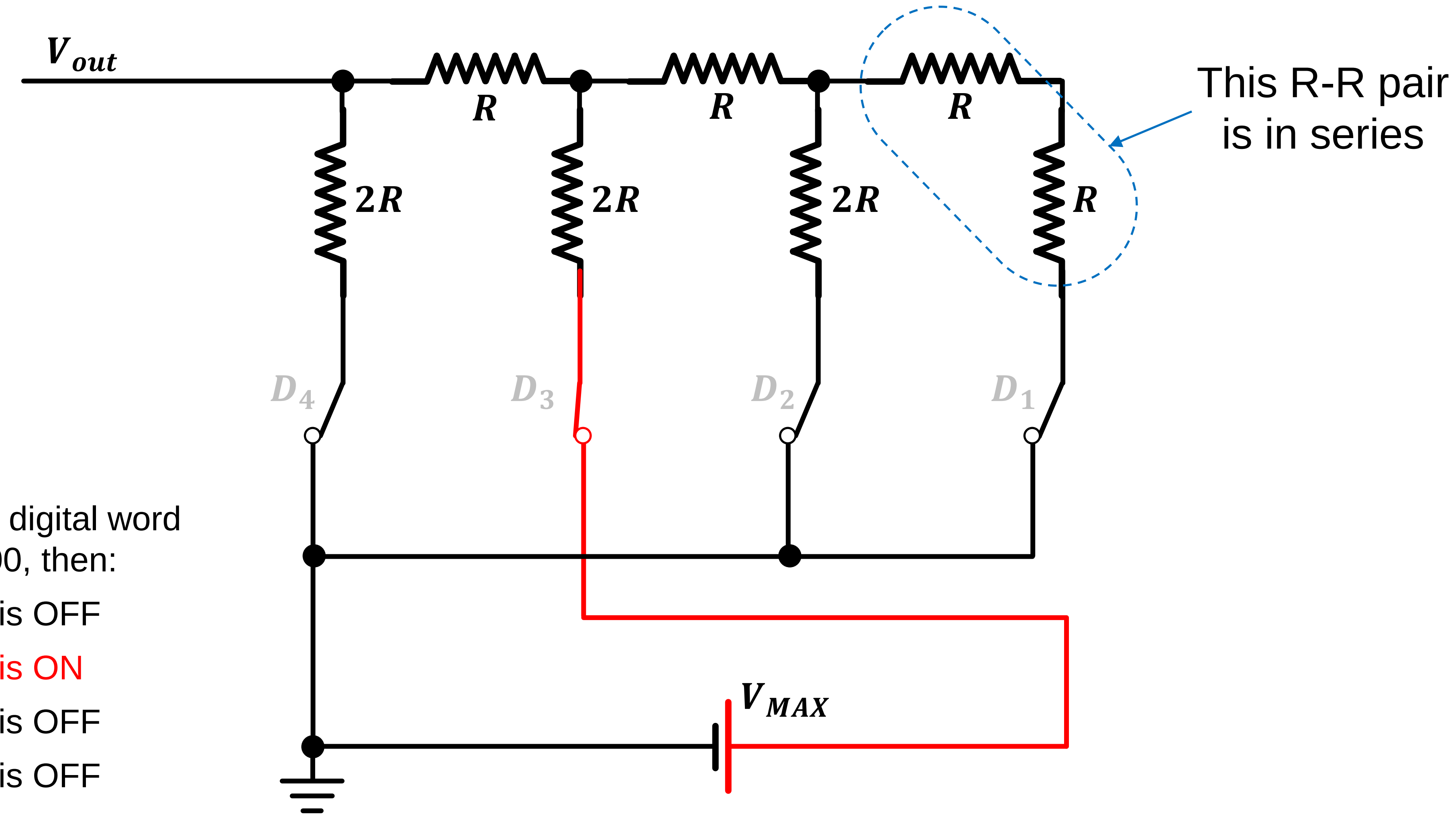
- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Let us do another example, but with a different bit activated!

Again, let us simplify this circuit:

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

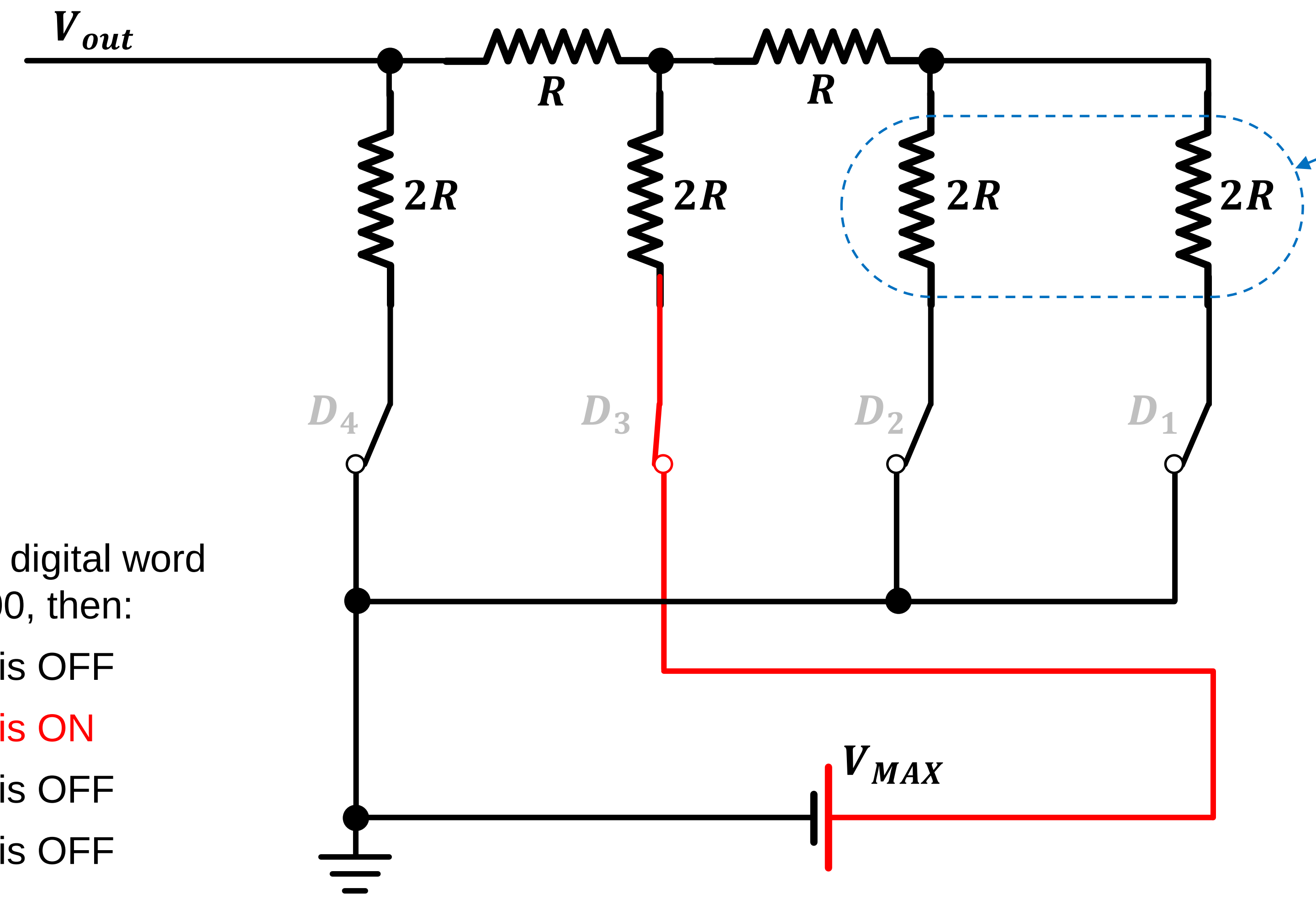


e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



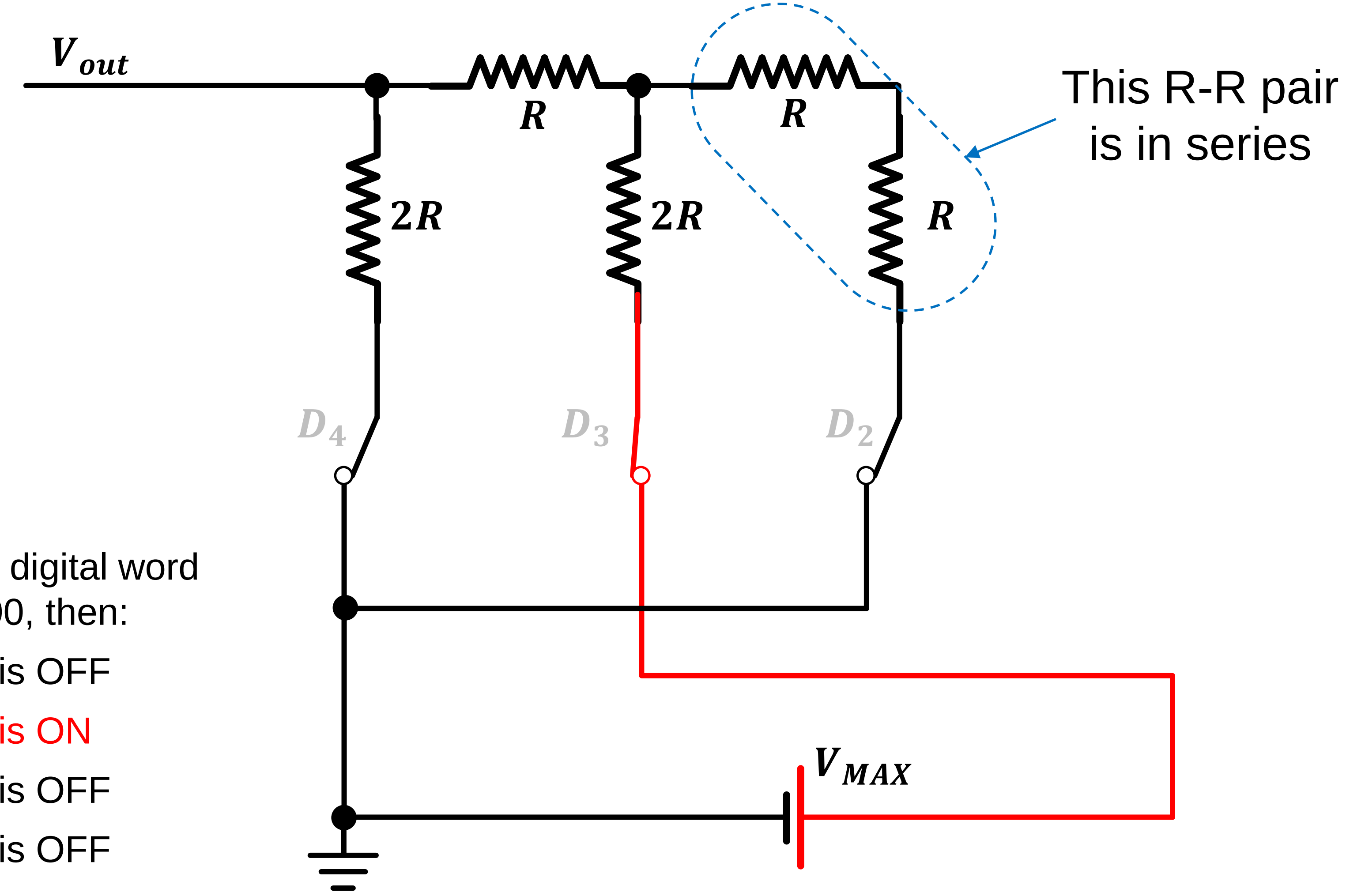
This 2R-2R pair is in parallel

e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

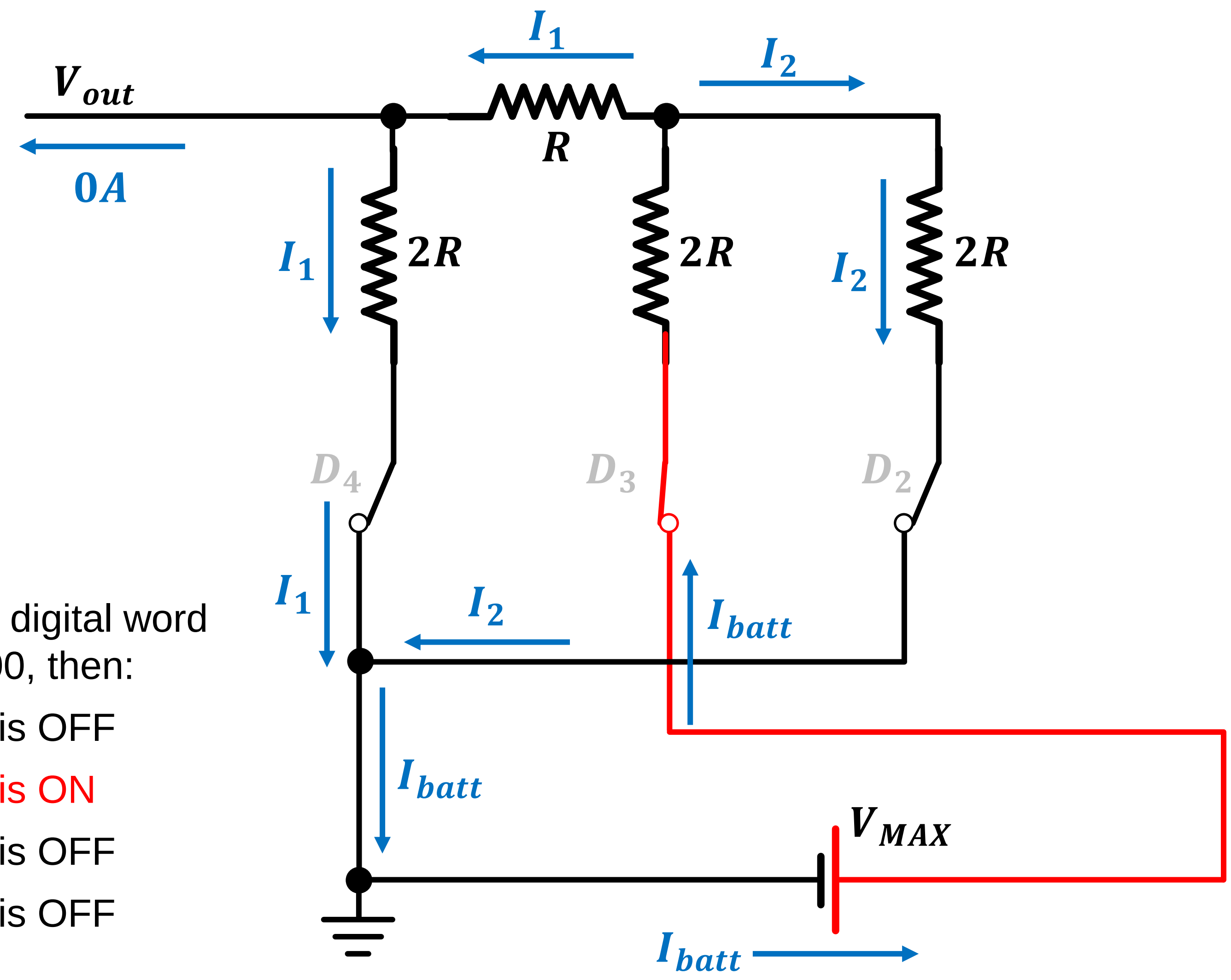


e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

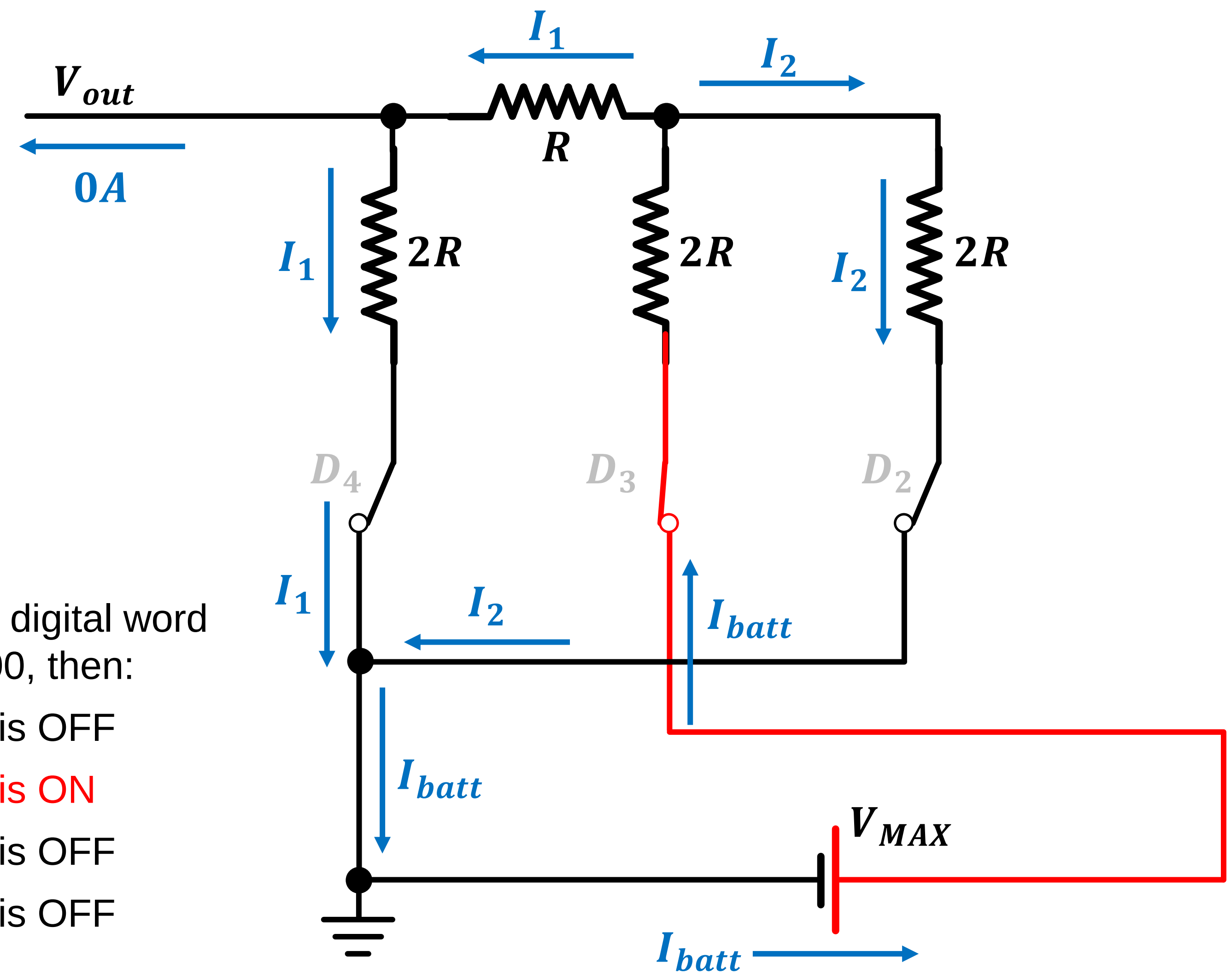
- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Now, as you can see, we cannot easily use the series/parallel rules to simplify this circuit further

Time to break out the Kirchhoff's Rules!

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

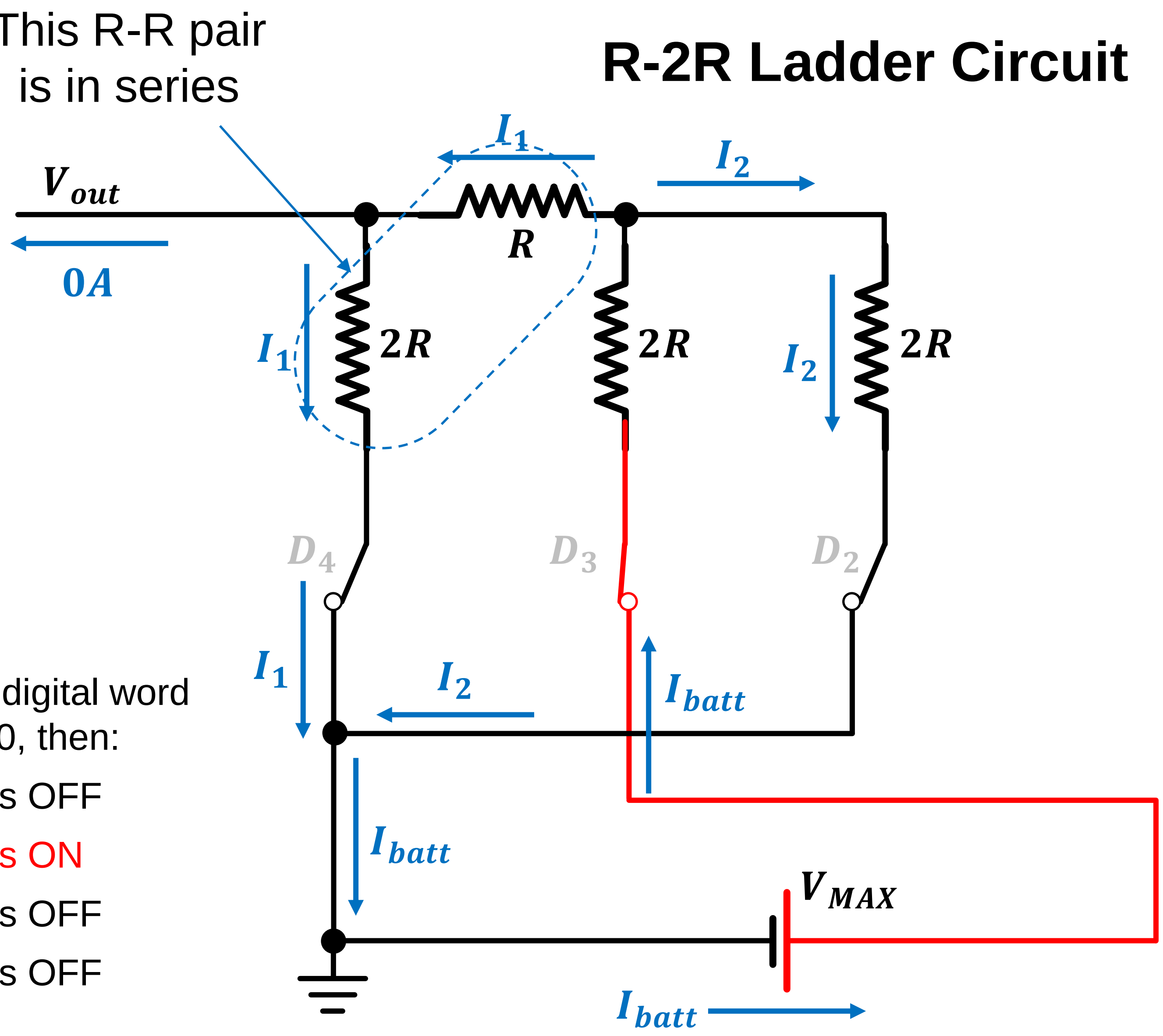
Kirchhoff's Current Law

$$I_{batt} = I_1 + I_2$$

Current Divider Rule – Current in a parallel branch divides as per the conductance offered by that branch as a fraction of the overall conductance

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Kirchhoff's Current Law

$$I_{batt} = I_1 + I_2$$

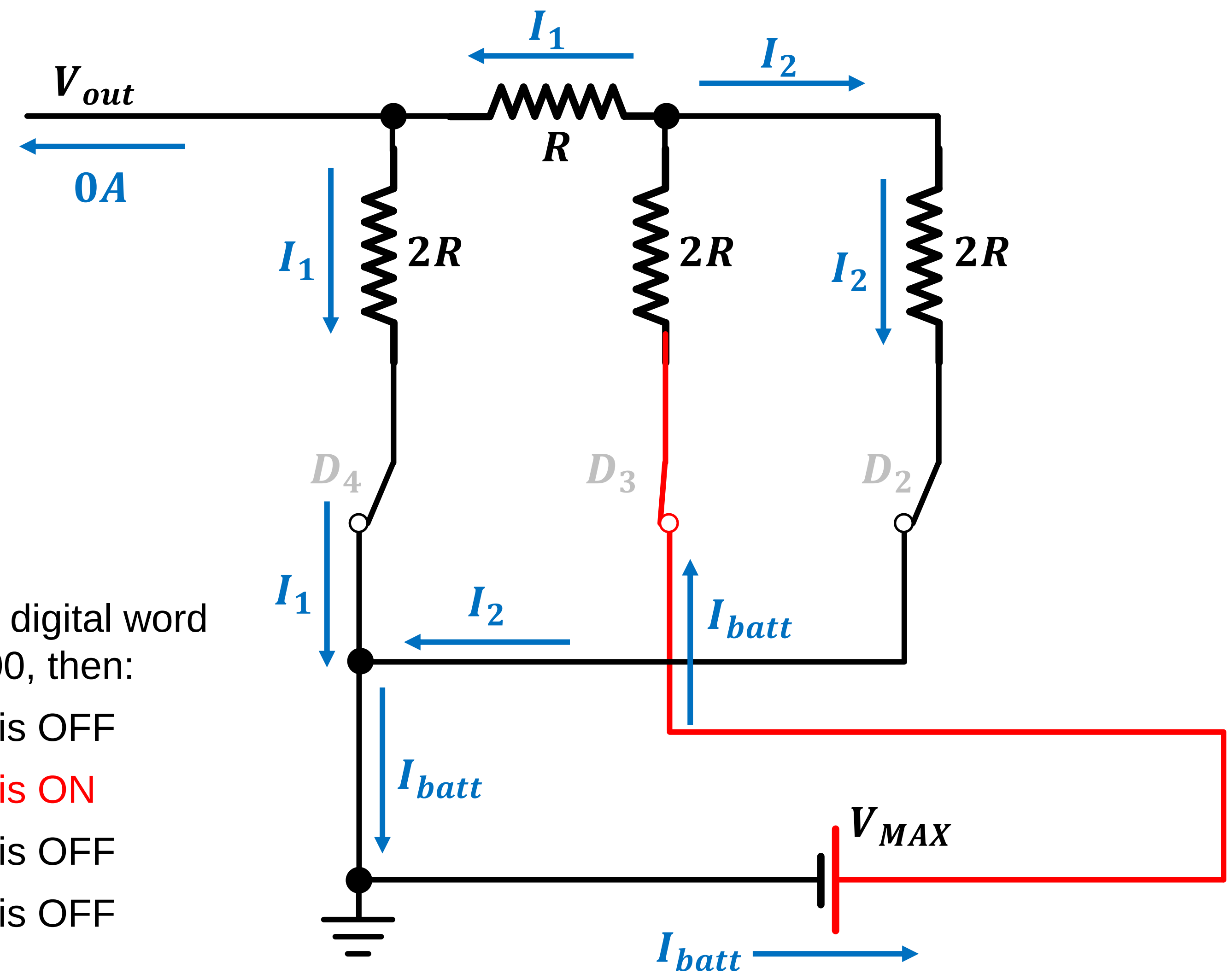
Current Divider Rule – Current in a parallel branch divides as per the conductance offered by that branch as a fraction of the overall conductance

$$G_1 = \frac{1}{R + 2R} = \frac{1}{3R}$$

$$G_2 = \frac{1}{2R}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Kirchhoff's Current Law

$$I_{batt} = I_1 + I_2$$

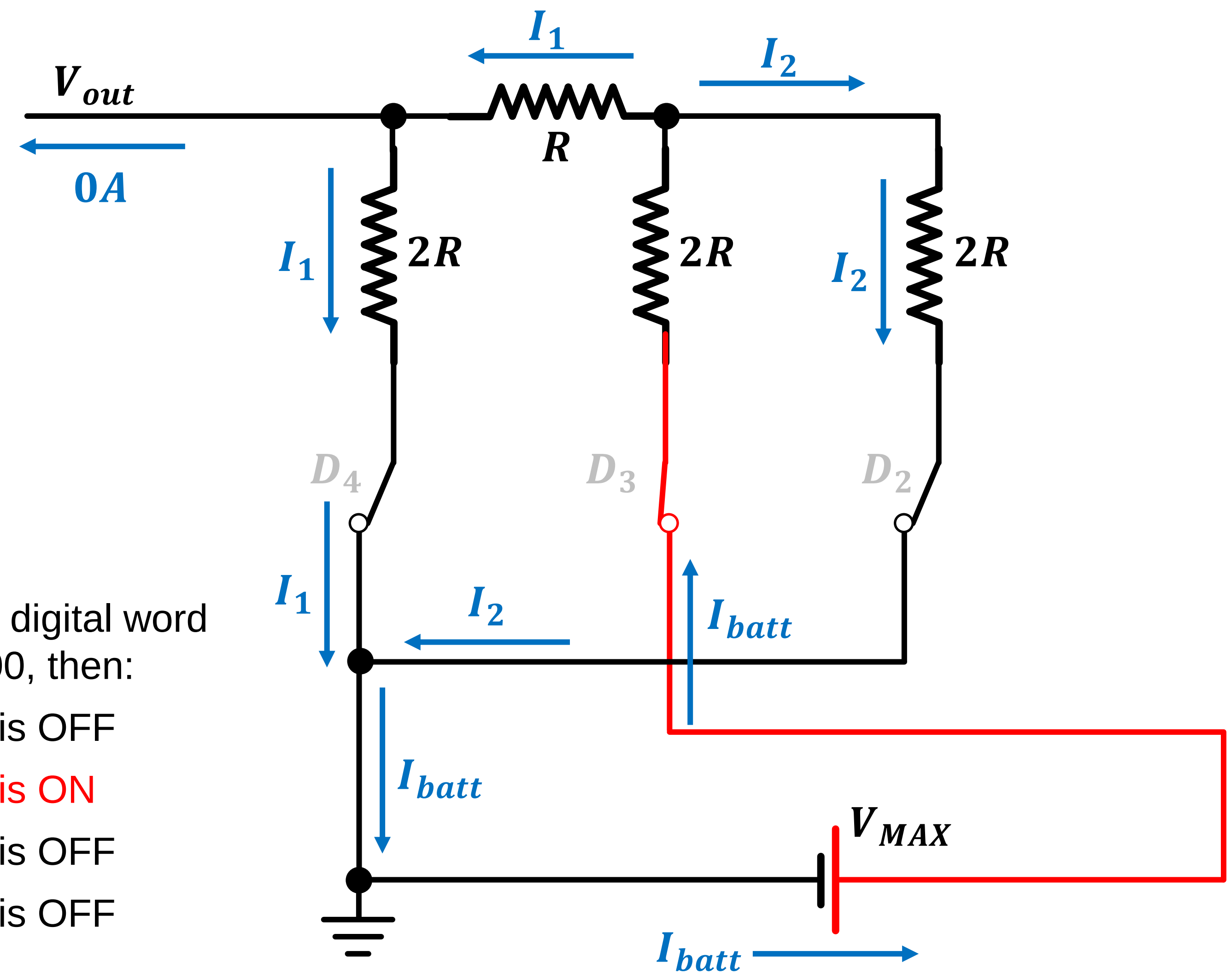
Current Divider Rule – Current in a parallel branch divides as per the conductance offered by that branch as a fraction of the overall conductance

$$I_1 = \frac{\frac{1}{3R}}{\frac{1}{3R}+\frac{1}{2R}} I_{batt} = \frac{\frac{2}{6R}}{\frac{2}{6R}+\frac{3}{6R}} I_{batt}$$

$$I_1 = \frac{2}{5} I_{batt}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Kirchhoff's Current Law

$$I_{batt} = I_1 + I_2$$

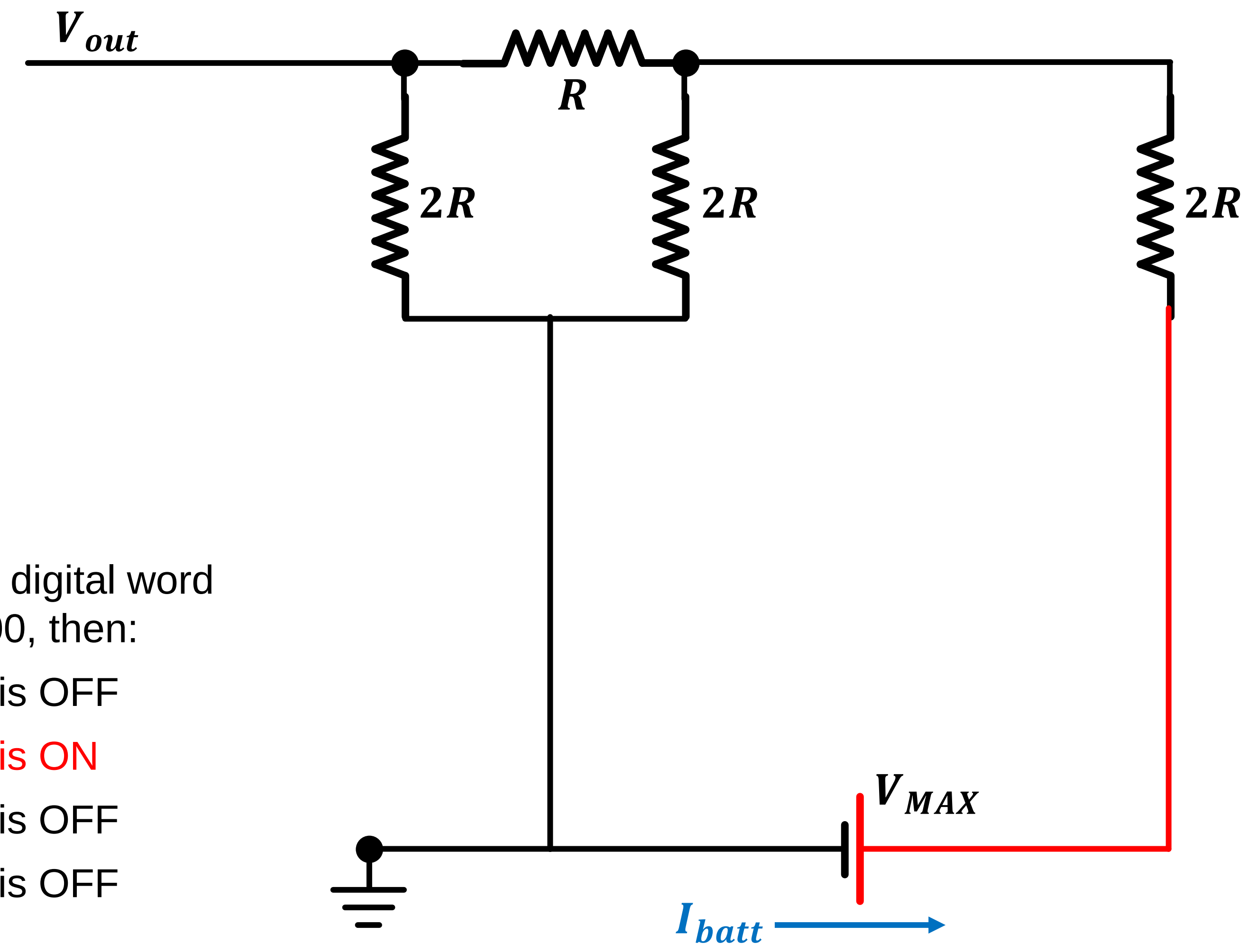
Current Divider Rule

$$I_1 = \frac{2}{5} I_{batt}$$

Now we need to find out I_{batt}

For that, we need to find out the equivalent resistance R_{eq} that is seen by the battery

R-2R Ladder Circuit



The circuit has been simplified:

- The positive voltage branch (red) has been moved to the right side
- The grounded branches (black) have been moved to the left

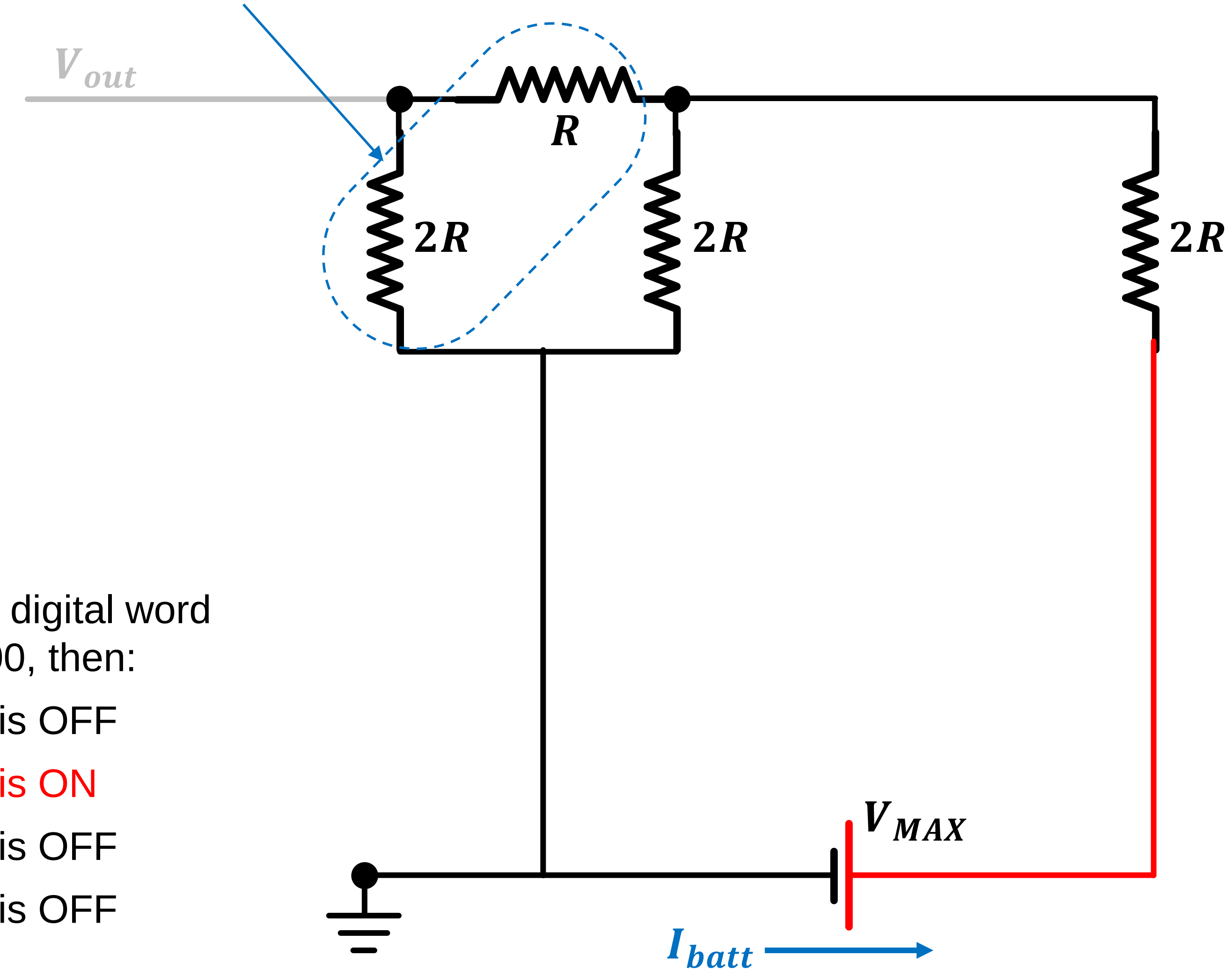
e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

This R-2R pair is in series

R-2R Ladder Circuit



The circuit has been simplified:

- The positive voltage branch (red) has been moved to the right side
- The grounded branches (black) have been moved to the left
- We can ignore V_{out} for the moment, as we are calculating the equivalent resistance – as there is no current flowing in/out of the V_{out} branch, it is effectively an infinite resistance, i.e., open circuit

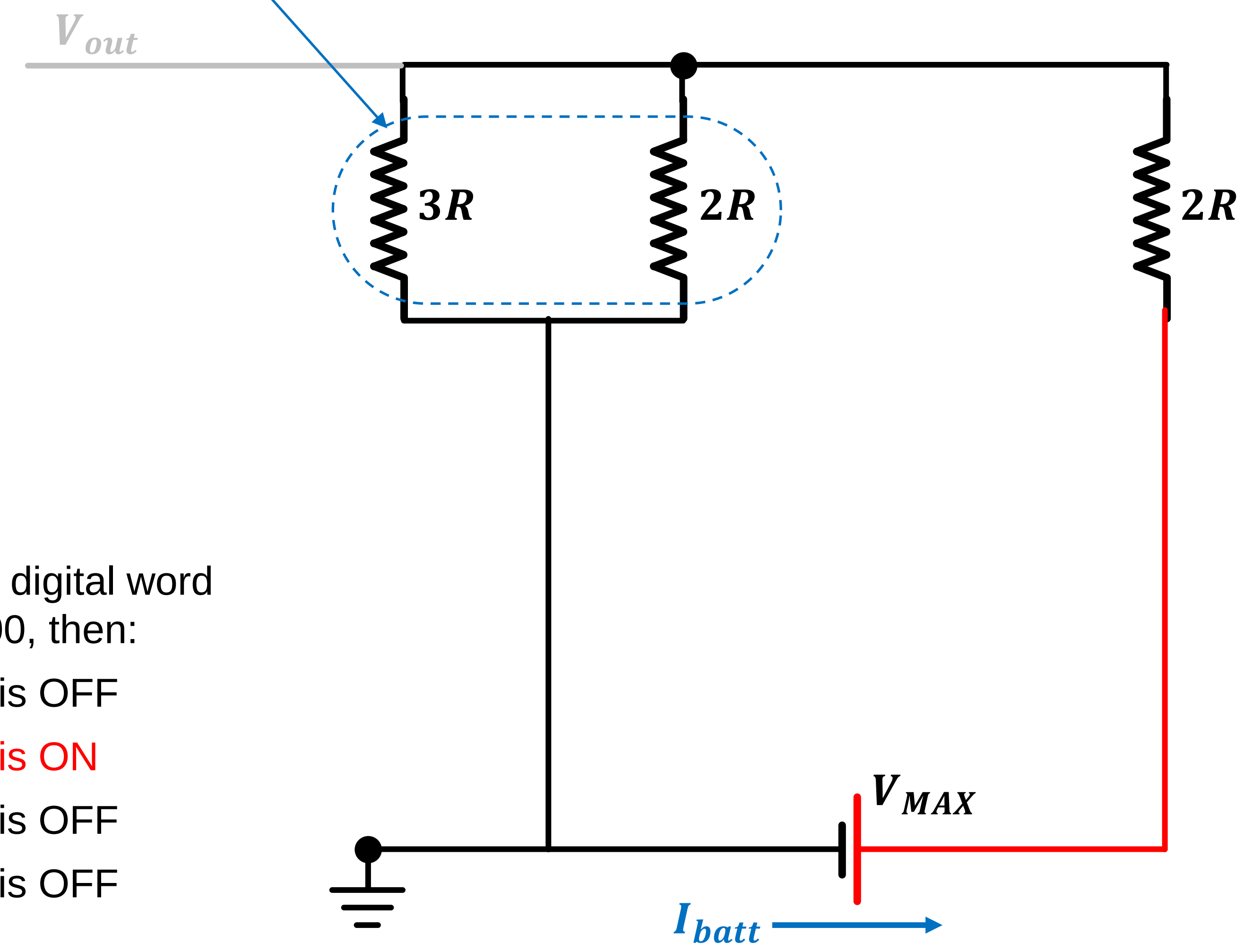
e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

This 3R-2R pair is in parallel



e.g., if the digital word is 0100, then:

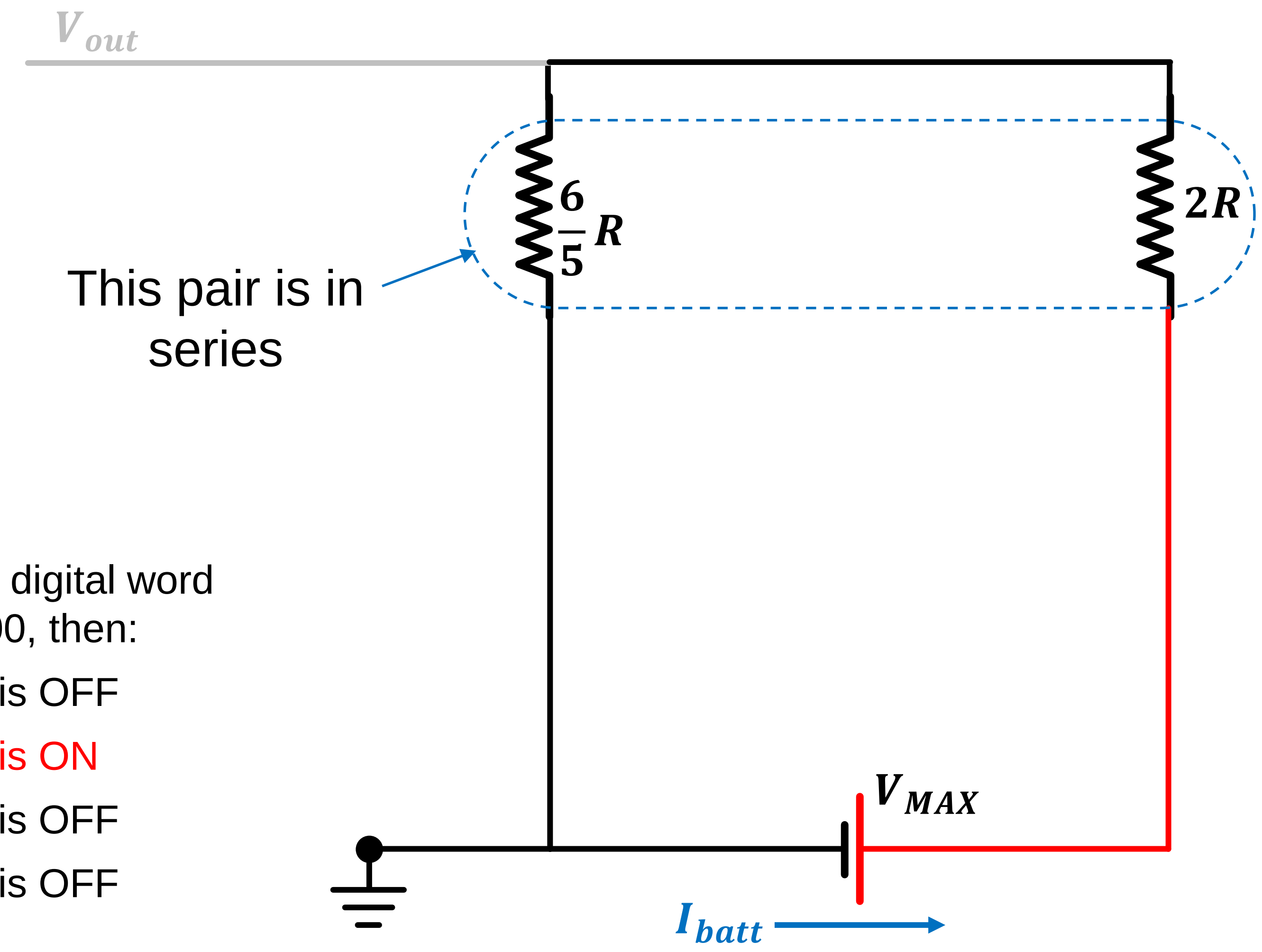
- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

The circuit has been simplified:

- The positive voltage branch (red) has been moved to the right side
- The grounded branches (black) have been moved to the left
- We can ignore V_{out} for the moment, as we are calculating the equivalent resistance – as there is no current flowing in/out of the V_{out} branch, it is effectively an infinite resistance, i.e., open circuit

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

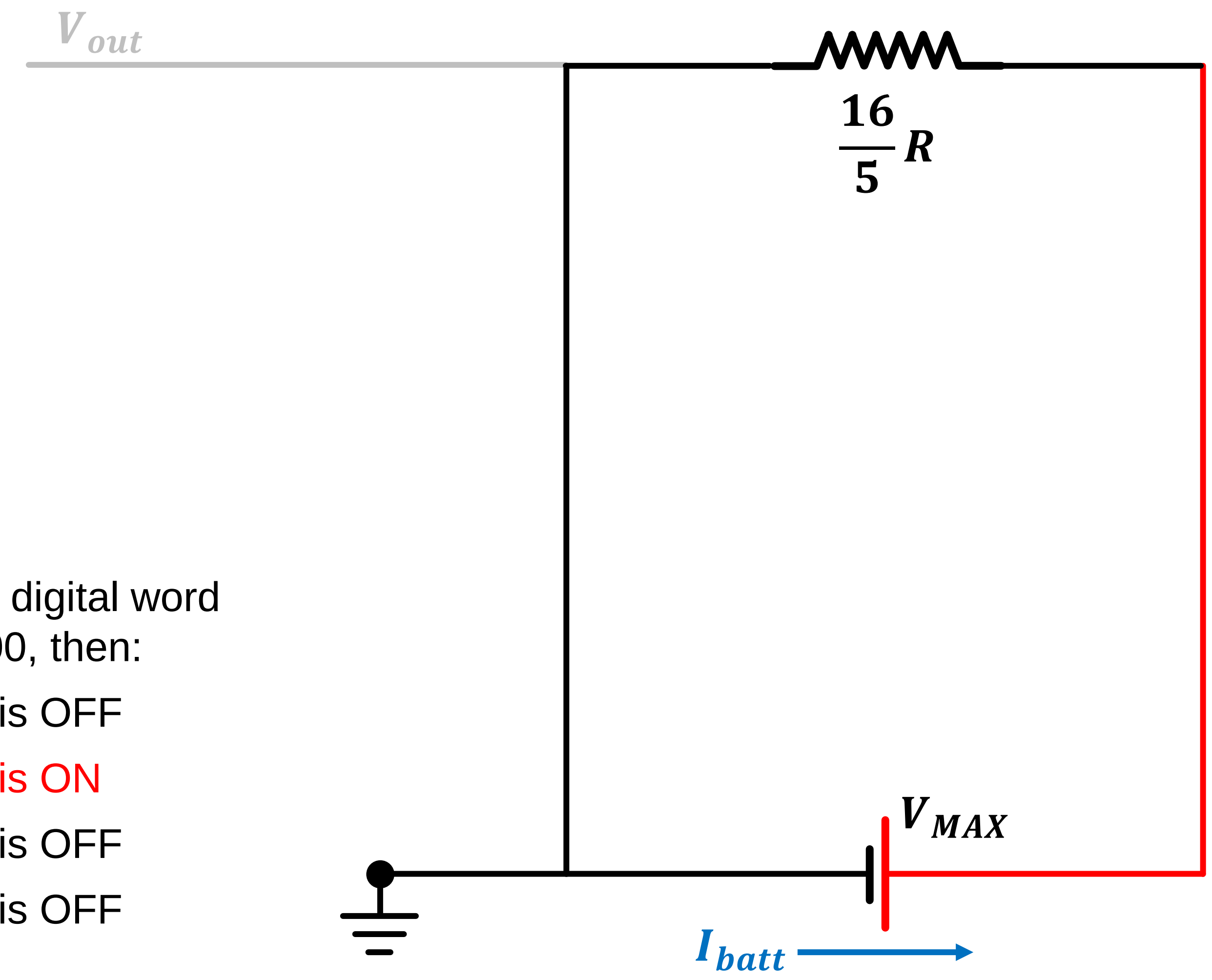
- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

The circuit has been simplified:

- The positive voltage branch (red) has been moved to the right side
- The grounded branches (black) have been moved to the left
- We can ignore V_{out} for the moment, as we are calculating the equivalent resistance – as there is no current flowing in/out of the V_{out} branch, it is effectively an infinite resistance, i.e., open circuit

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



$$R_{eqv} = \frac{16}{5}R$$

Applying Ohm's Law:

$$I_{batt} = \frac{V_{MAX}}{\frac{16}{5}R} = \frac{5}{16}\frac{V_{MAX}}{R}$$

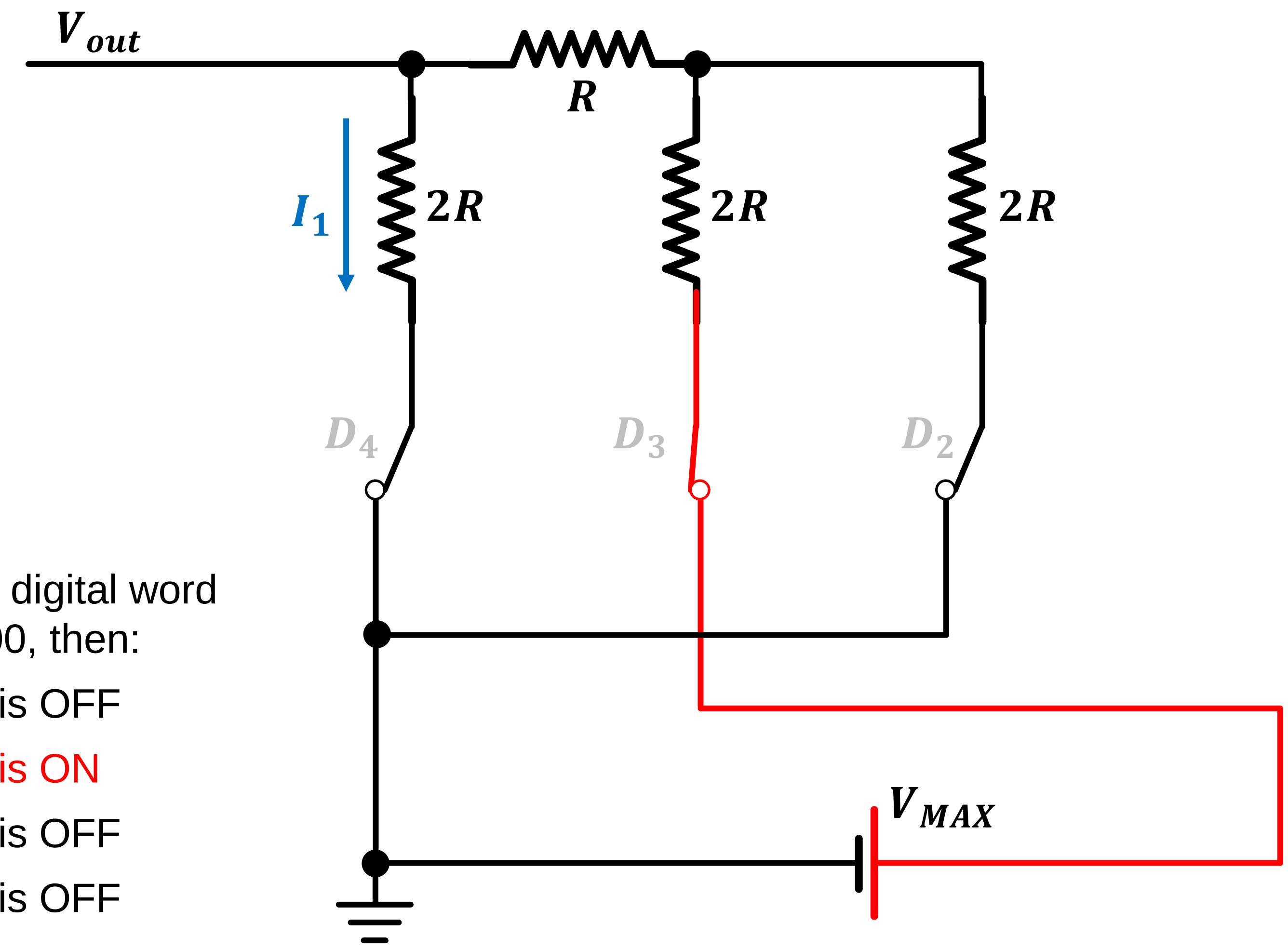
Let us go back to the original circuit now:

e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

$$R_{eqv} = \frac{16}{5} R$$

Applying Ohm's Law:

$$I_{batt} = \frac{V_{MAX}}{\frac{16}{5} R} = \frac{5}{16} \frac{V_{MAX}}{R}$$

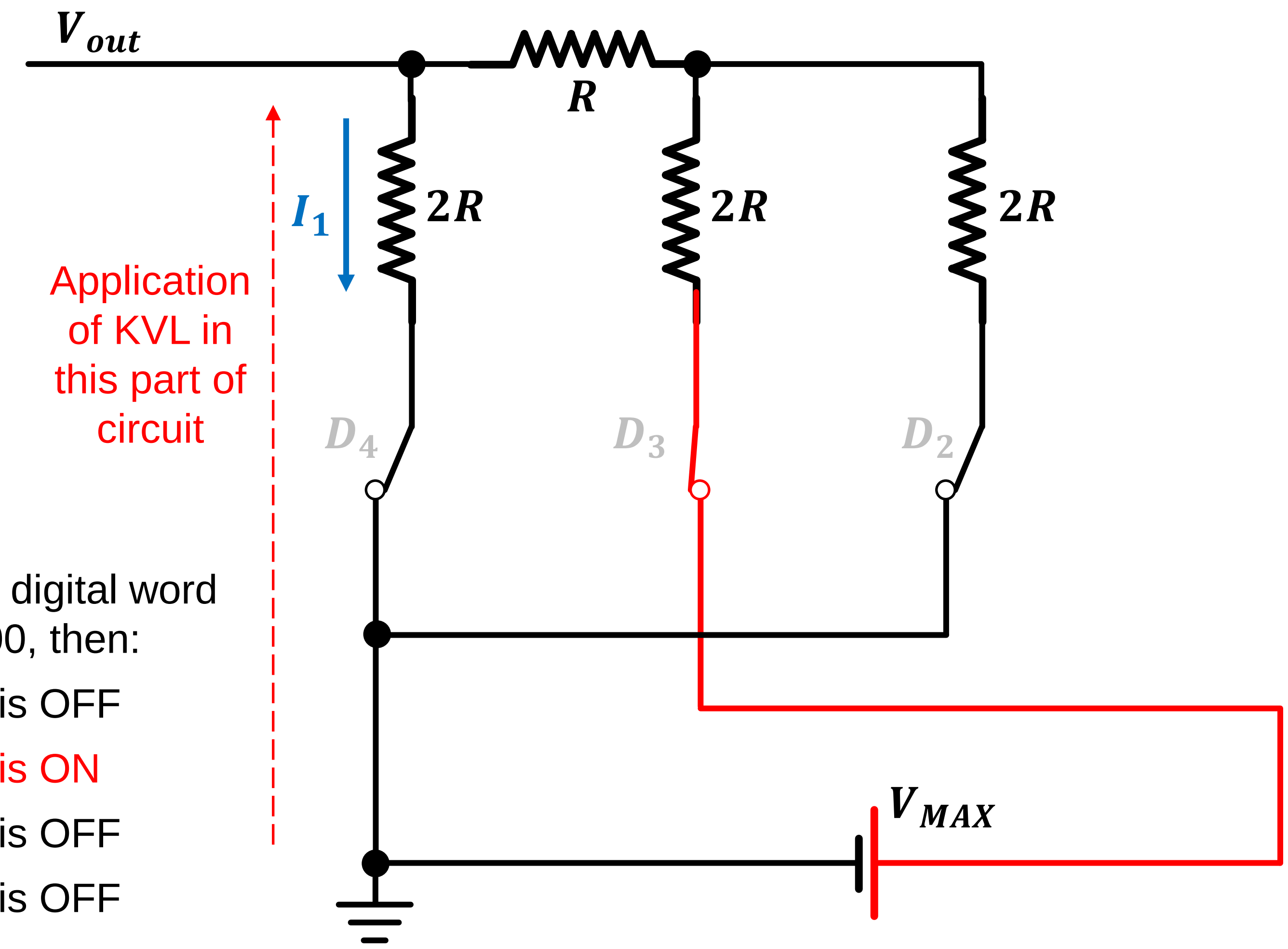
Let us go back to the original circuit now:

$$I_1 = \frac{2}{5} I_{batt} = \frac{2}{5} \times \frac{5}{16} \frac{V_{MAX}}{R}$$

$$I_1 = \frac{1}{8} \frac{V_{MAX}}{R}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 0100, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

$$I_1 = \frac{1}{8}\frac{V_{MAX}}{R}$$

To find V_{out}, we must apply KVL:

$$V_{out} = V_{2R} + 0$$

Now apply Ohm's Law:

$$V_{out} = V_{2R} = I_1 \times 2R$$

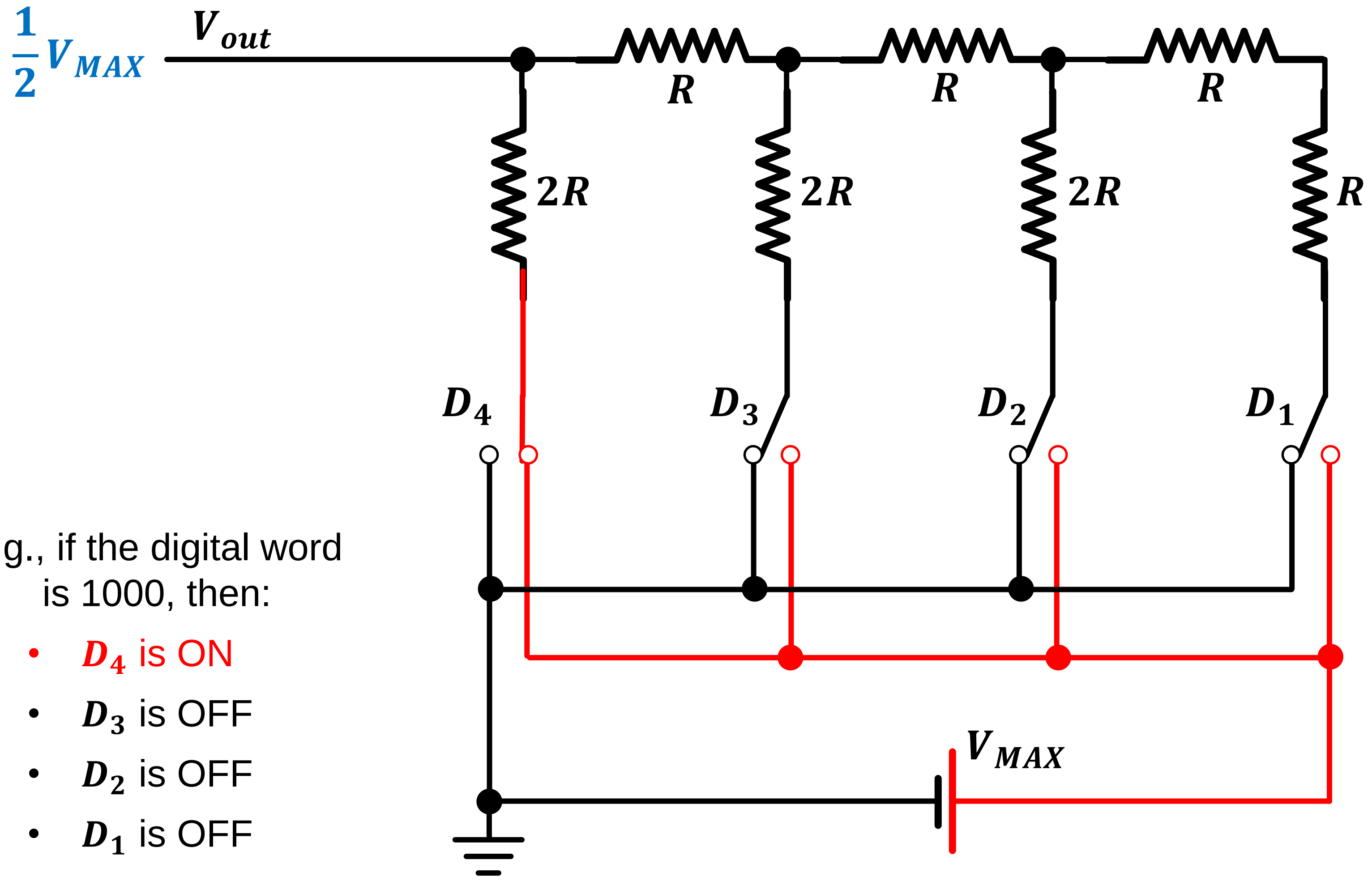
$$V_{out} = \frac{1}{8}\frac{V_{MAX}}{R} \times 2R$$

$$V_{out} = \frac{1}{4}V_{MAX}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

Now we can extrapolate the logic



When D_4 is ON

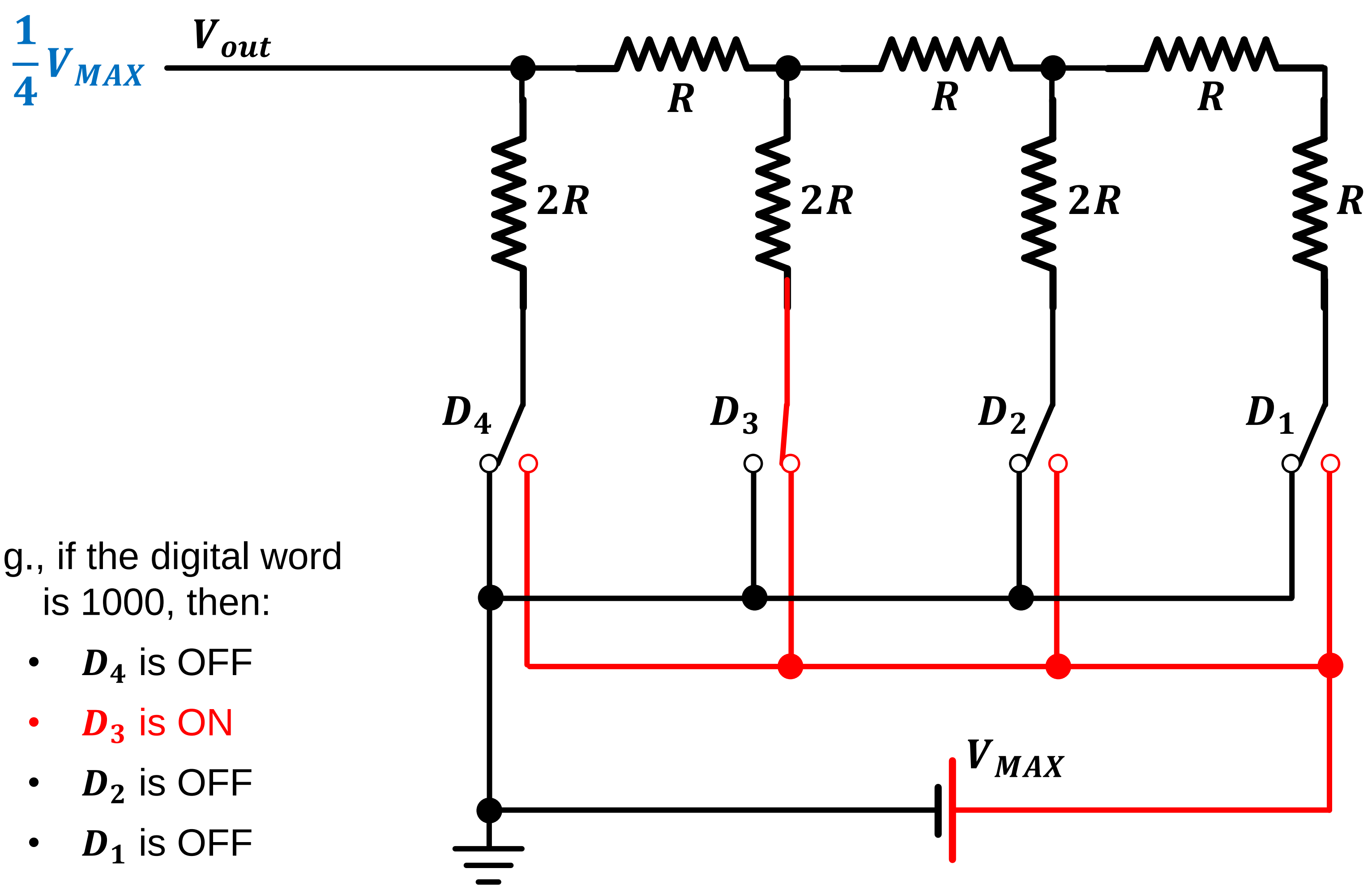
$$V_{out} = \frac{1}{2}V_{MAX}$$

e.g., if the digital word is 1000, then:

- D_4 is ON
- D_3 is OFF
- D_2 is OFF
- D_1 is OFF

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 1000, then:

- D_4 is OFF
- D_3 is ON
- D_2 is OFF
- D_1 is OFF

Now we can extrapolate the logic

When D_4 is ON

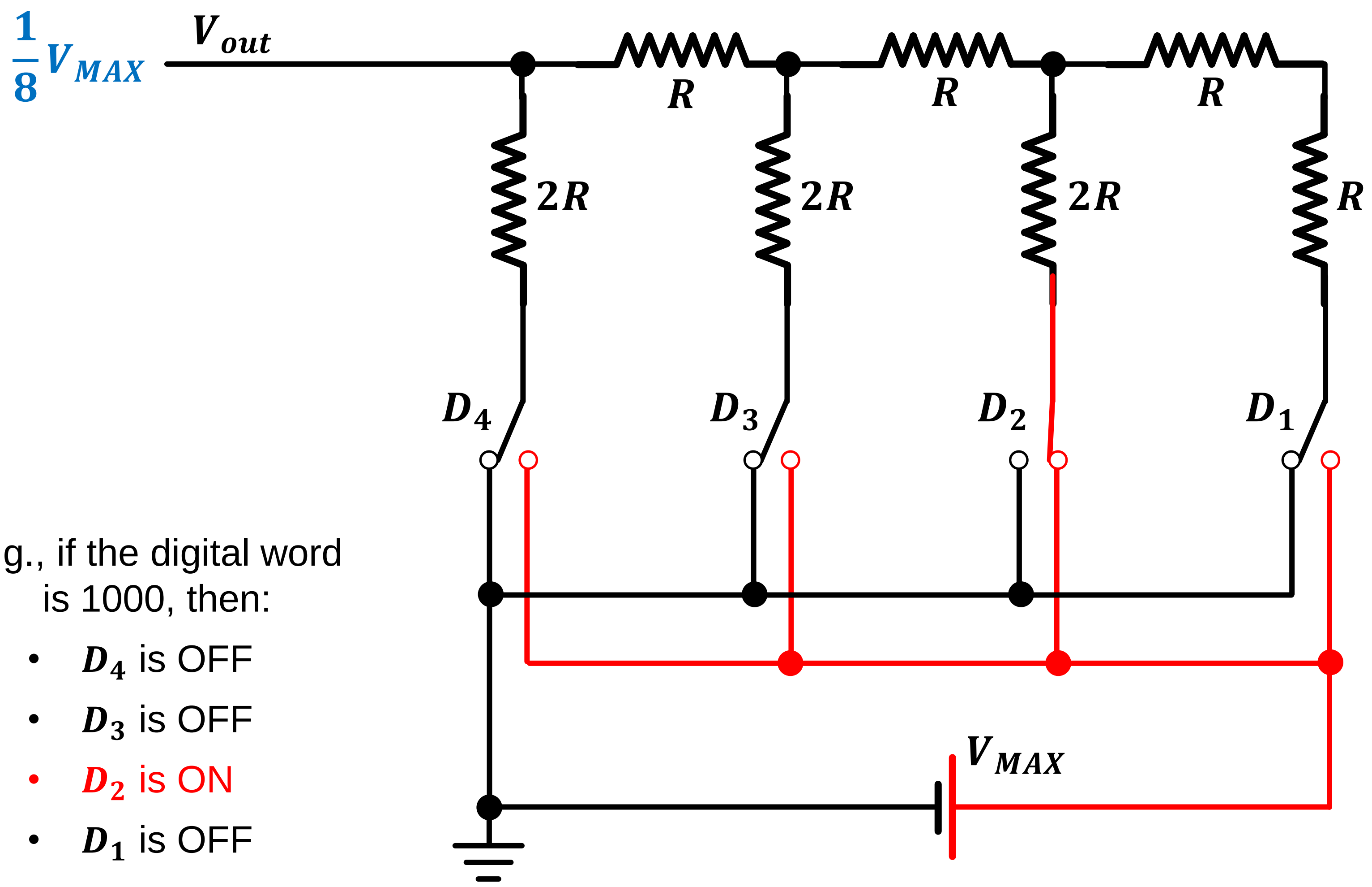
$$V_{out} = \frac{1}{2}V_{MAX}$$

When D_3 is ON

$$V_{out} = \frac{1}{4}V_{MAX}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 1000, then:

- D_4 is OFF
- D_3 is OFF
- D_2 is ON
- D_1 is OFF

Now we can extrapolate the logic

When D_4 is ON

$$V_{out} = \frac{1}{2}V_{MAX}$$

When D_3 is ON

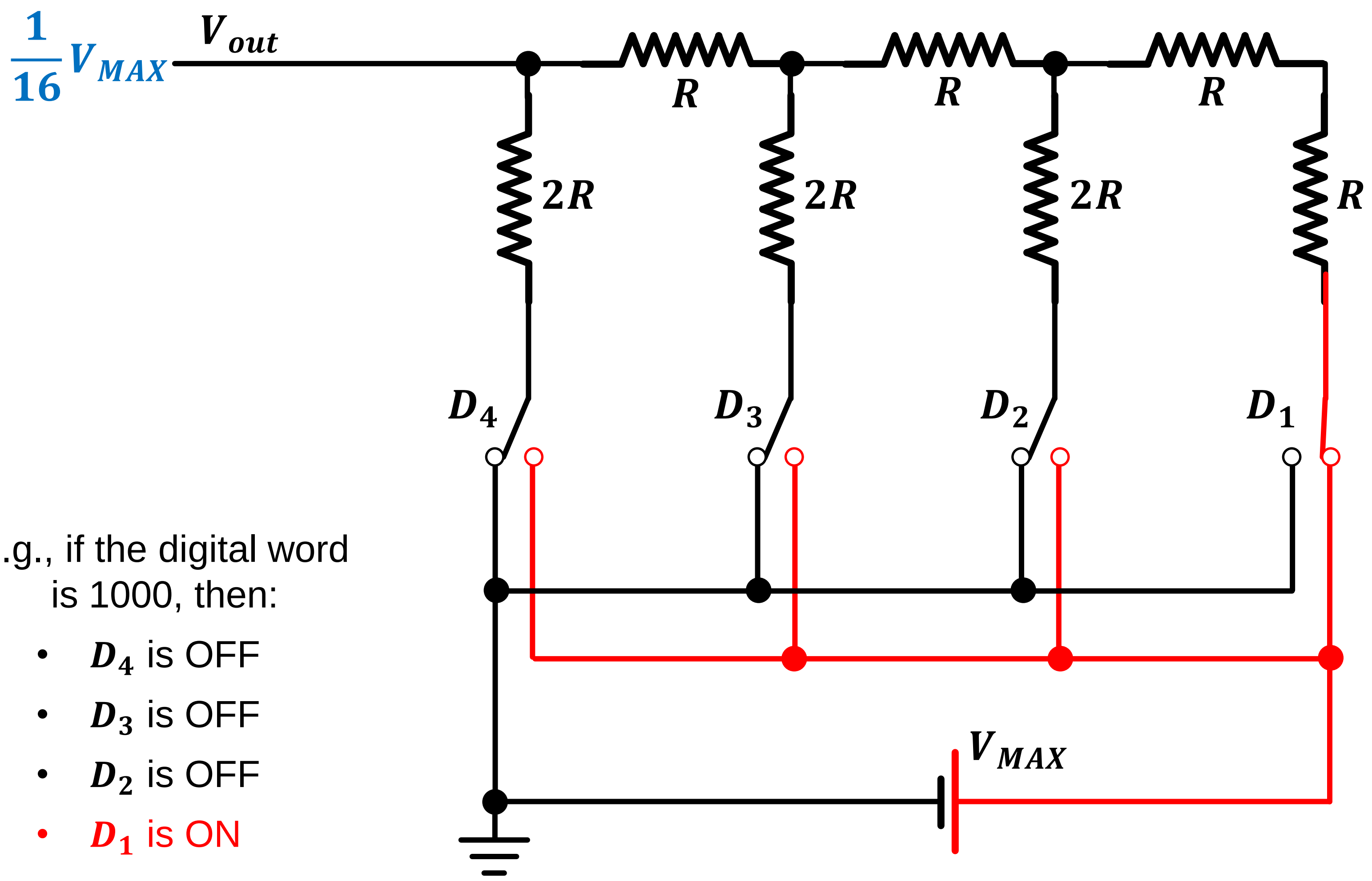
$$V_{out} = \frac{1}{4}V_{MAX}$$

When D_2 is ON

$$V_{out} = \frac{1}{8}V_{MAX}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



e.g., if the digital word is 1000, then:

- D_4 is OFF
- D_3 is OFF
- D_2 is OFF
- D_1 is ON

Now we can extrapolate the logic

When D_4 is ON

$$V_{out} = \frac{1}{2}V_{MAX}$$

When D_3 is ON

$$V_{out} = \frac{1}{4}V_{MAX}$$

When D_2 is ON

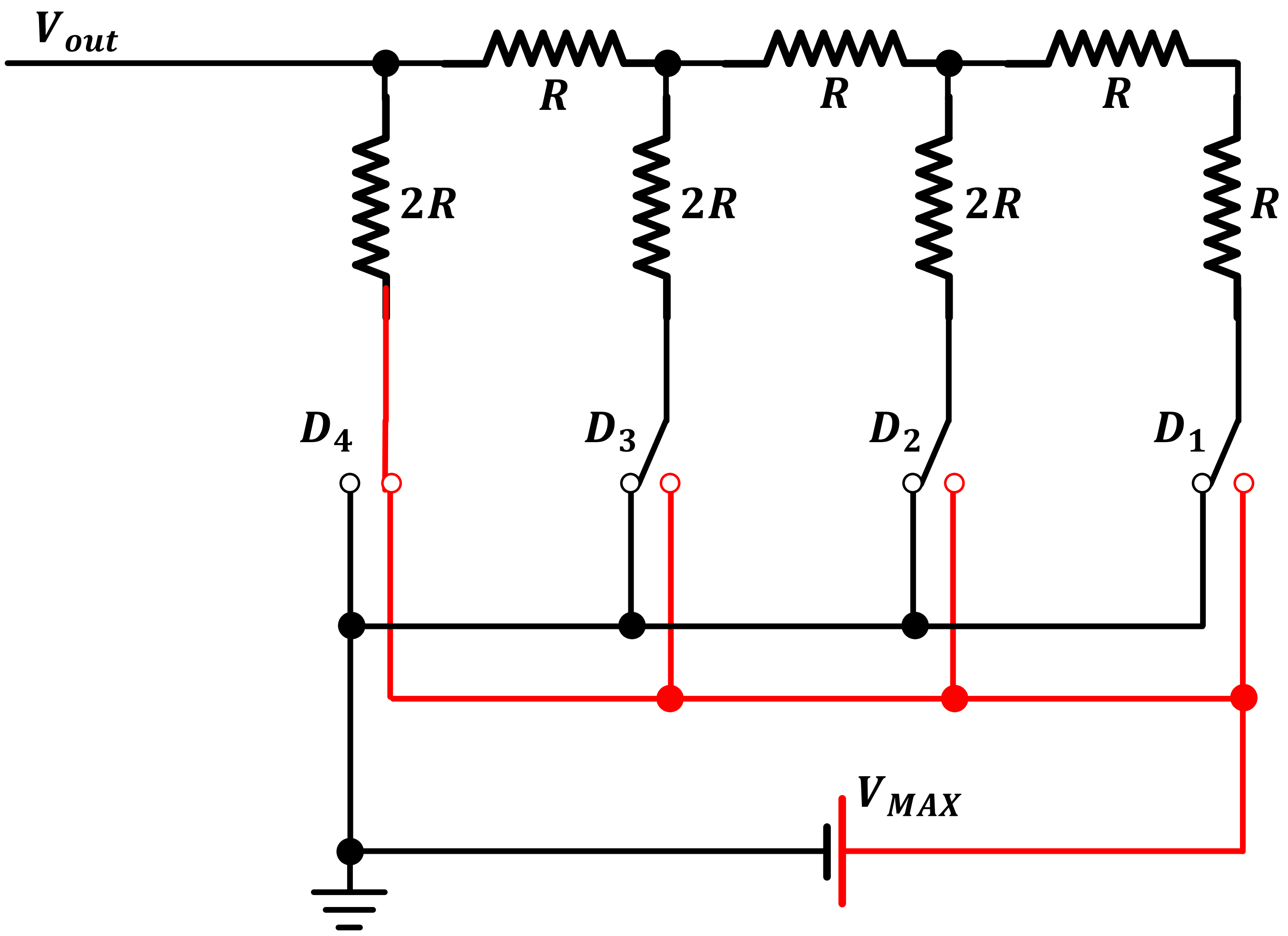
$$V_{out} = \frac{1}{8}V_{MAX}$$

When D_1 is ON

$$V_{out} = \frac{1}{16}V_{MAX}$$

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit



If there are n branches in the R-2R Ladder circuit:

$$V_{out} = \frac{1}{2^n} V_{MAX}$$

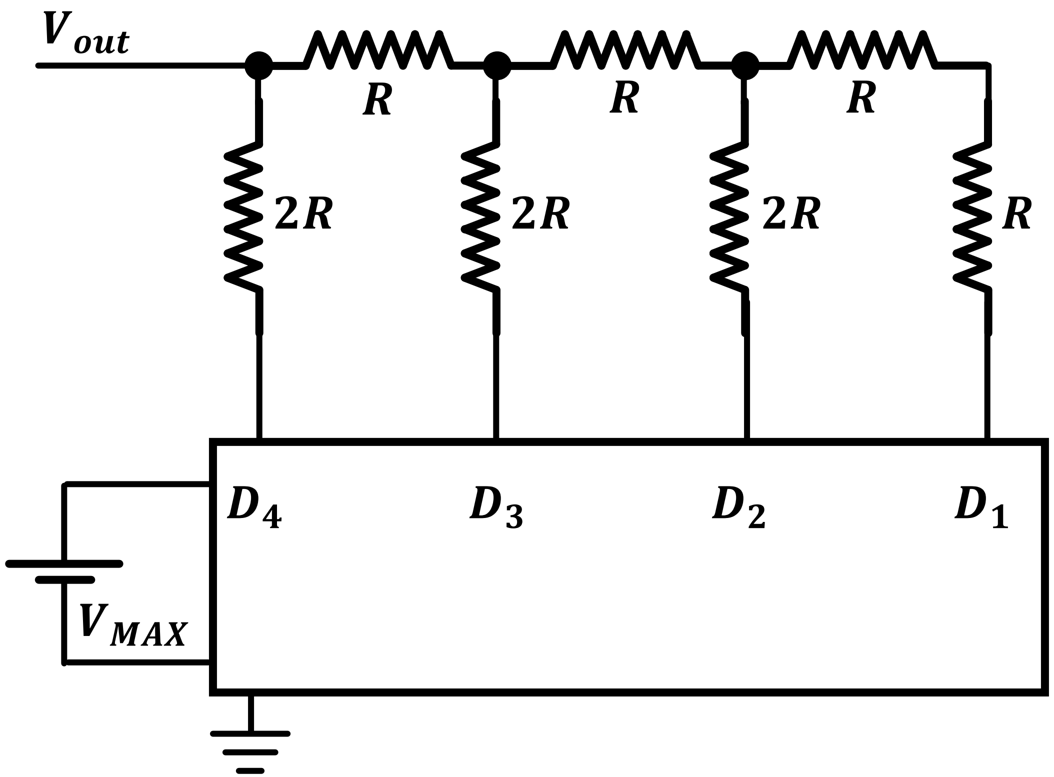
With multiple switches on at the same time, the individual contribution of each branch gets added to V_{out}

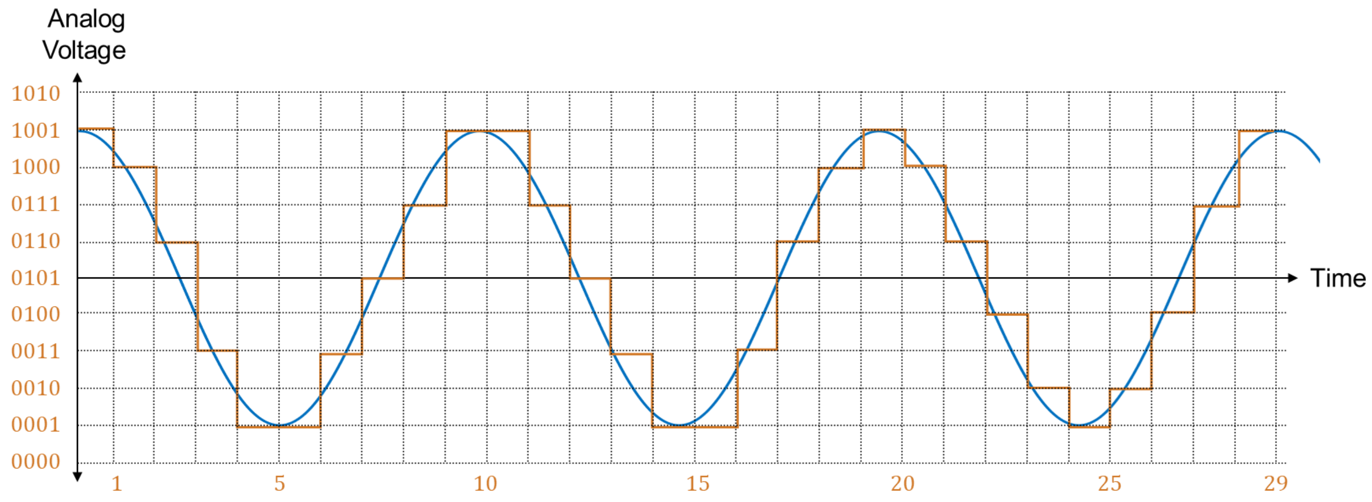
$$V_{out} = \sum D_n \frac{1}{2^n} V_{MAX}$$

Where D_n is the bit value, 1 or 0

Digital-to-Analog Converter (ADC)

R-2R Ladder Circuit

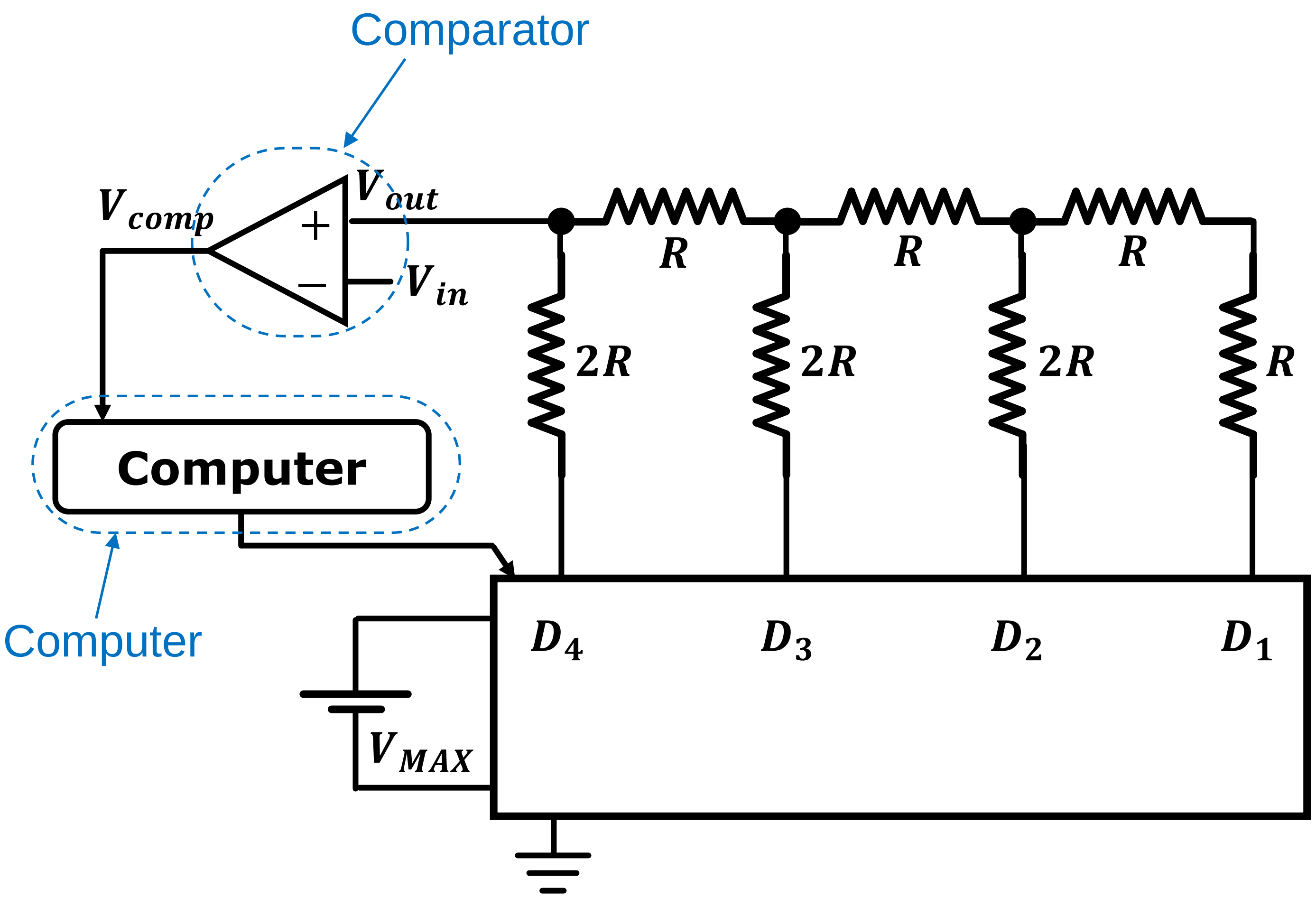




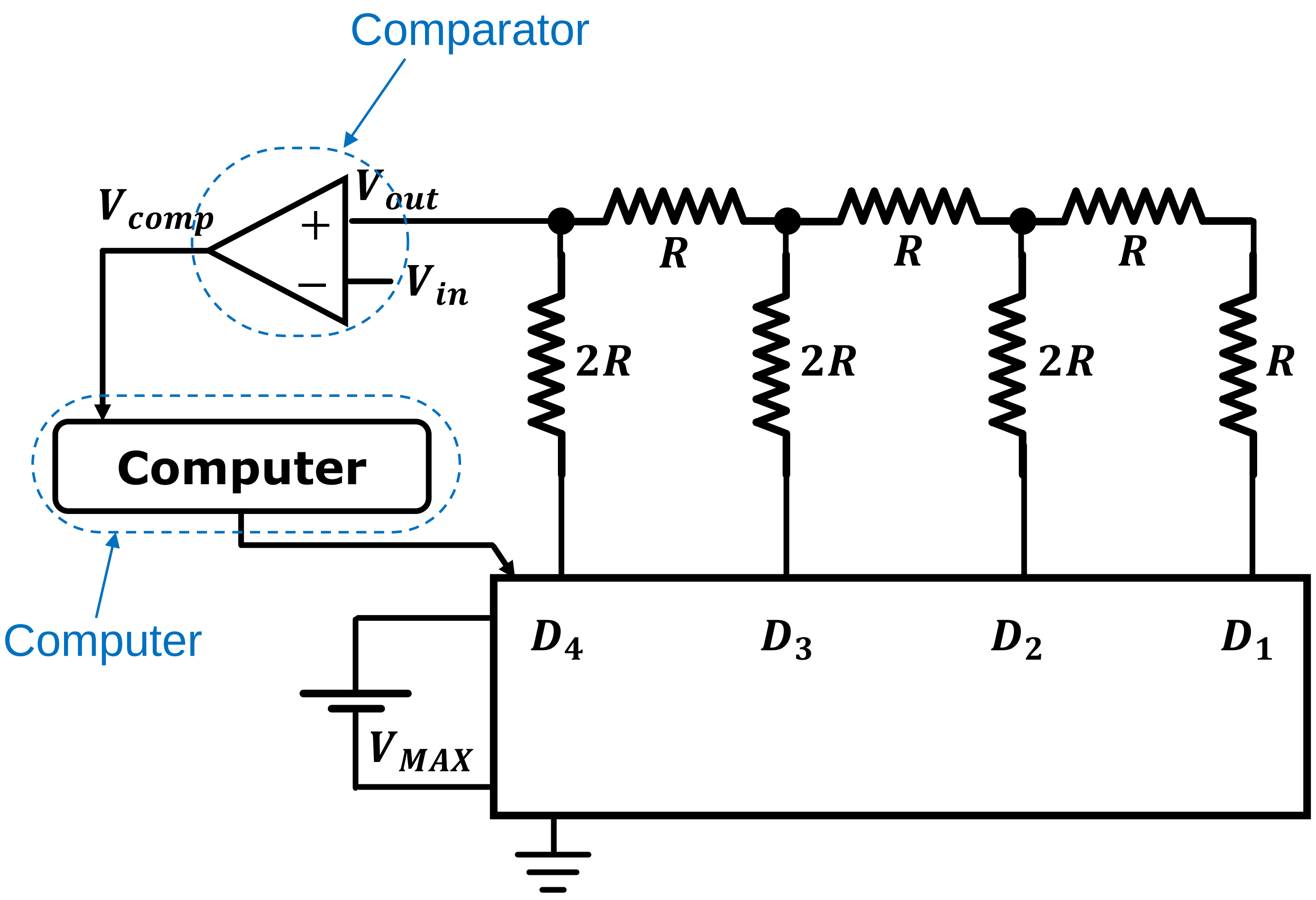
A **faster clock** would allow **finer divisions** on the **time** axis

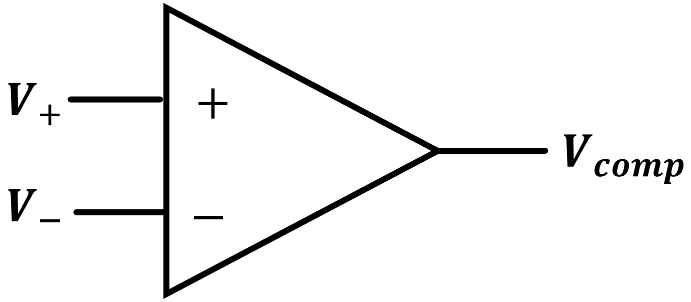
A **larger word length** (n-bit) would allow **finer divisions** on the **amplitude** axis

Analog-Digital-Converter (ADC)



Analog-Digital-Converter (ADC)





Comparator

Compares the input voltages

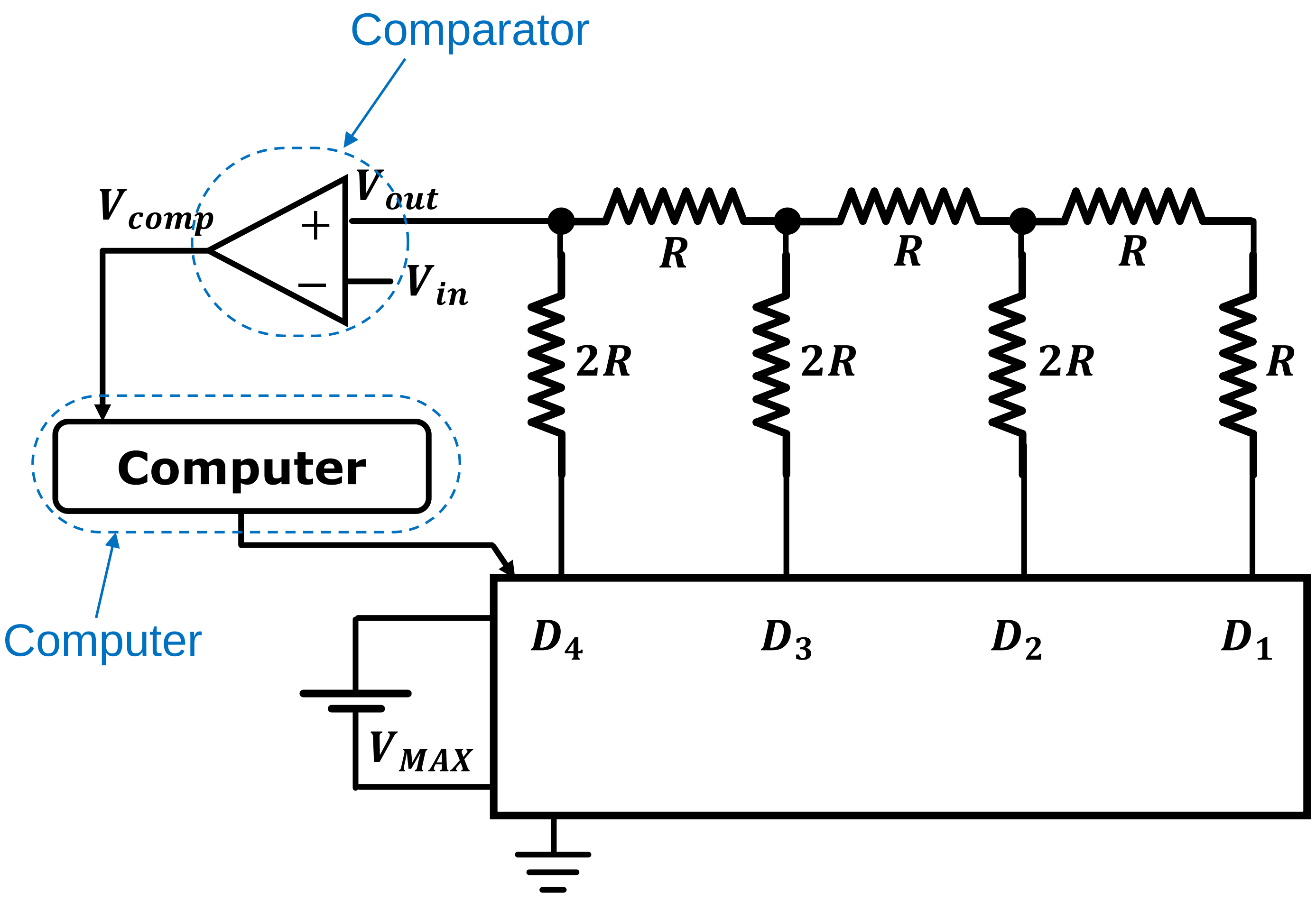
If:

- $V_+ > V_-$ then $V_{comp} = 1$
- $V_+ < V_-$ then $V_{comp} = 0$

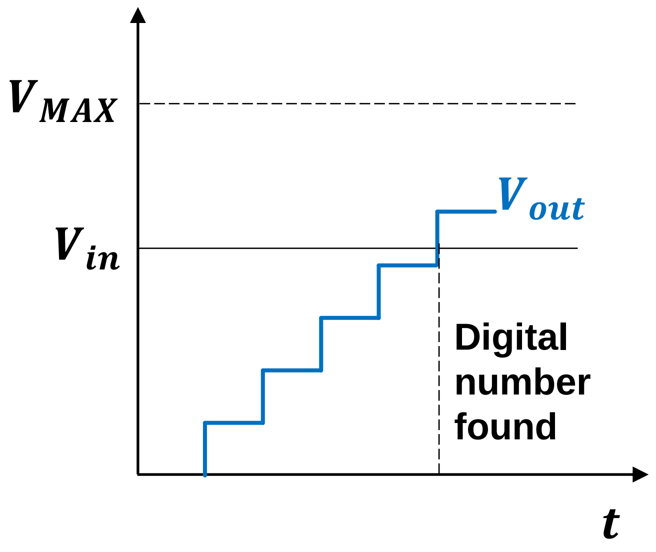
This is a special form of Operational Amplifier (or Op-Amp)

We will cover this in next lecture

Analog-Digital-Converter (ADC)



Notice that this takes up to 2^n clock pulses to do the conversion – often this is too slow for some applications



Computer

- Increases the R-2R ladder analog voltage output by counting up in binary
- Waits for comparator output to turn 1
- Records the binary state as the digital equivalent of input analog voltage V_{in}

Flash Converter

Voltage Divider

Notice we have divided the full scale voltage into 4 divisions

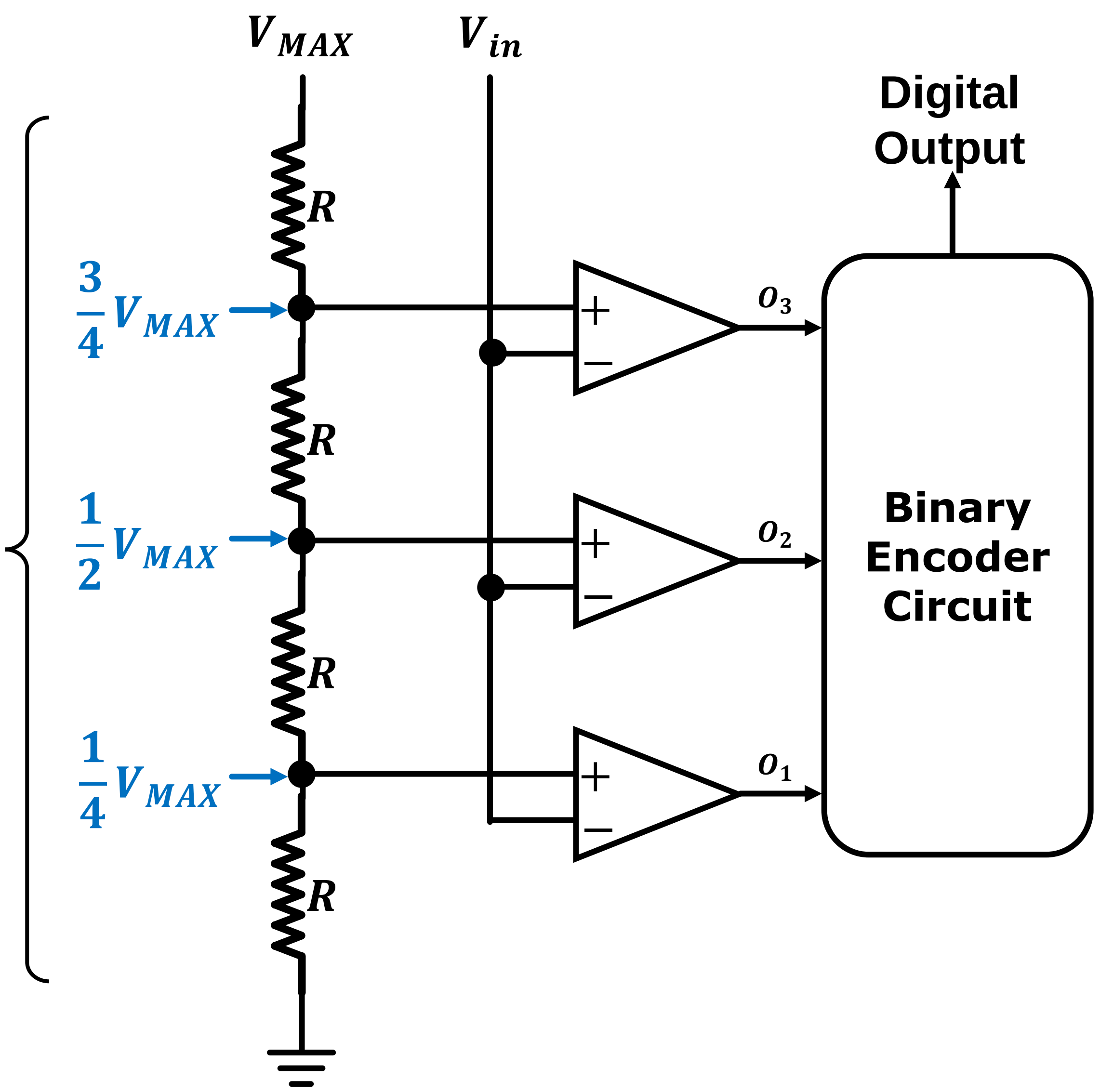
In order to convert this into a binary number, we should aim to make this an exponent of 2

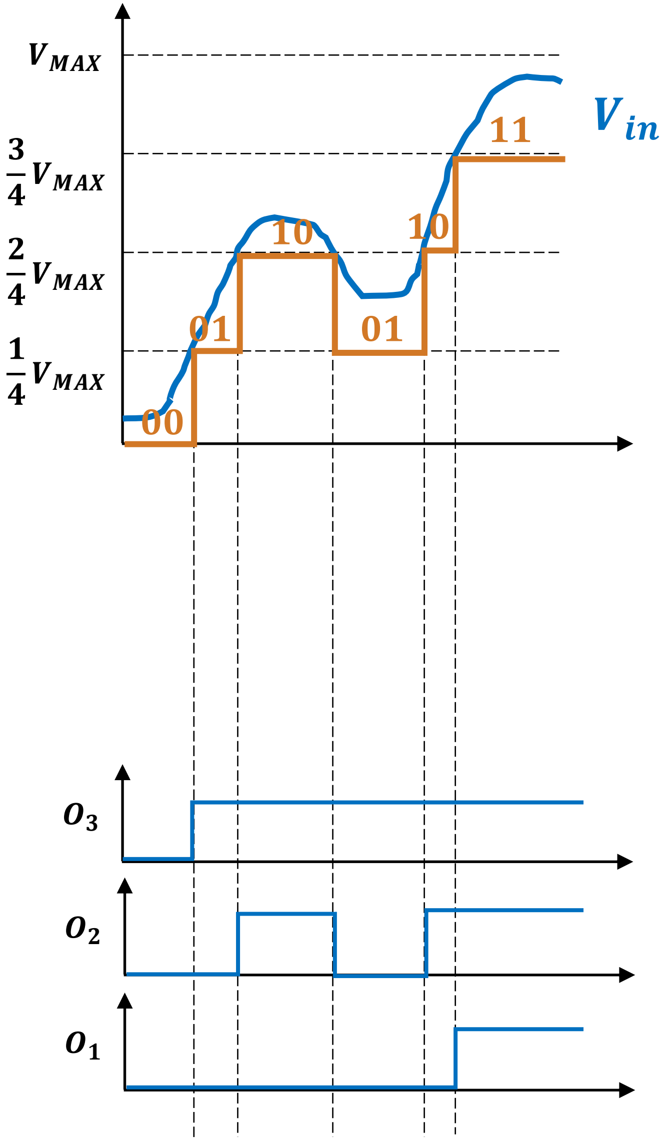
$$2^n$$

There is no necessity to do that – you will end up with an under-utilised binary word

e.g., say you have 6 divisions

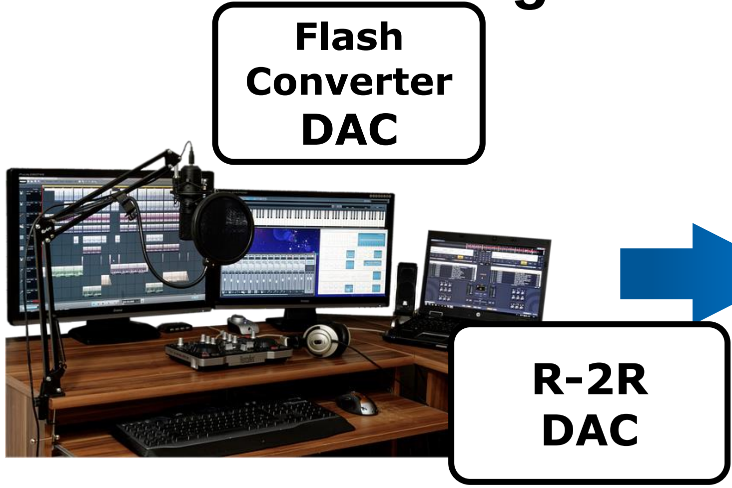
You would need a 3-bit word which would have given you 8 divisions

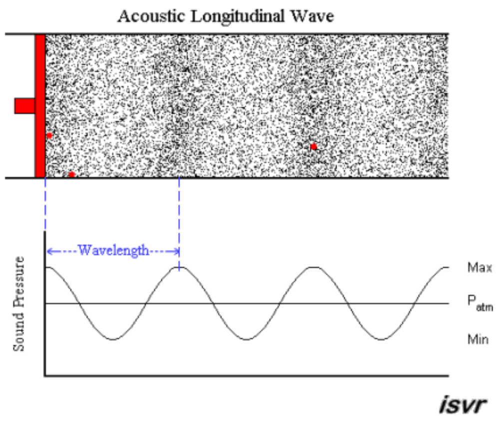




Digital-Analog Conversion

Typical application of inter-conversion between analog and digital signals





Air pressure waves gets picked up by a diaphragm in the mic

Diaphragm motion produces a small analog voltage signal proportional to the sound pressure waveform

ADC soundcard in the PC converts the analog waveform into digital signal

User does all kinds of processing to the sound digitally

DAC produces an analog signal that can be read by speakers/amplifiers

Speakers convert the analog voltage signal into sound waves again

Summary

- Revision of Logic Gates
  - **Shaft Encoder**
- **Flip Flops**
  - Latch v Flip Flop
  - SR/JK/D/T Flip Flops
- Applications of Digital Circuit
  - **Series** v **Parallel** data & conversion (**Bit Shifter**)
  - Analog/Digital conversion (**R-2R Ladder** circuit)
  - **Flash Converter**

Attendance

UP|PHY|B1-MMME2051EMD