Lecture/Laboratory #11/12

- Review lab
- Review FF and Timers
- Options with 555 Timers
- Crystal Oscillators
- Analog – to – Digital and Digital – to – Analog Conversions
  - Number Codes
  - Hexadecimal Representation
  - Binary – Coded Decimal
  - Gray Code
  - Signed Numbers

Laboratory: (a) Op – amp relaxation oscillator

(b) Clock (oscillator) with 555

(c) Sawtooth oscillator: f = 3 kHz

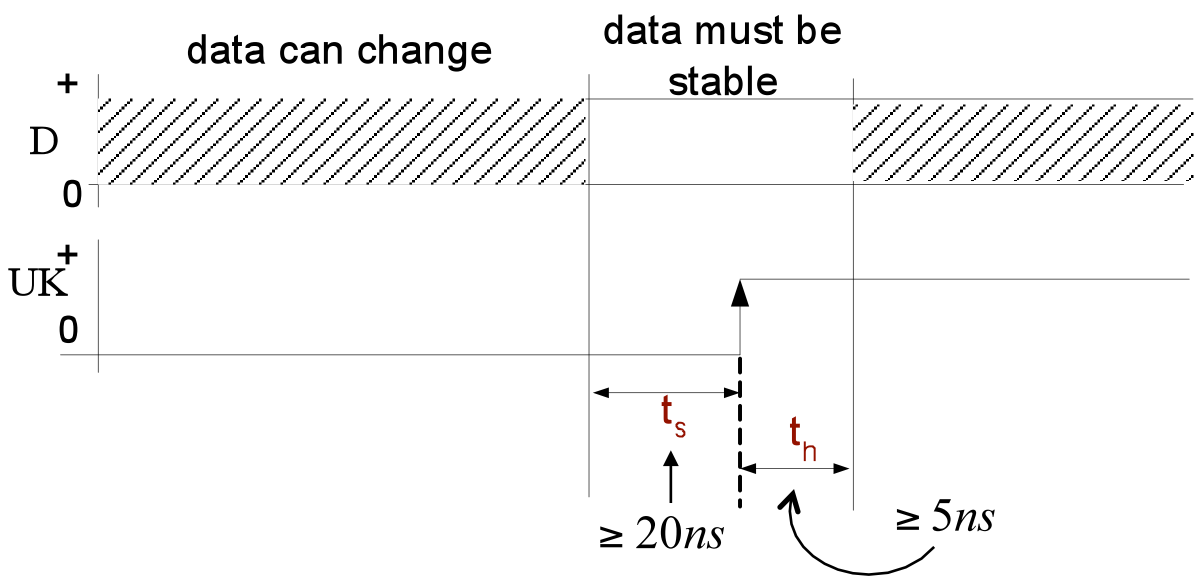
(d) One-shot $\Rightarrow$ driving servo

Data and Clock Timing

When does the D – FF (or any other FF) look at its input, relative to the clock pulse, in order to react?

There is a specified "setup time" t_s and "hold time" t_h for any clocked device. Input data must be present and stable from at least t_s before the clock transition until at least t_h after in order to have proper operation.

Example: D – FF "74" ⇒

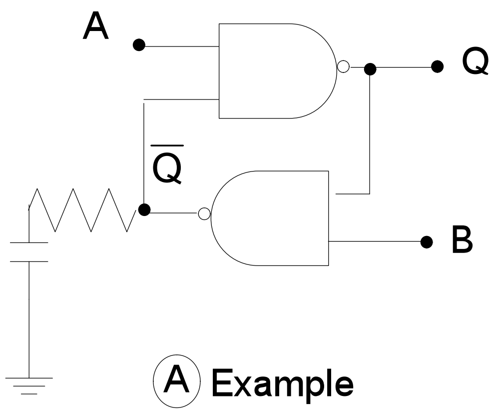


Time of propagation of signal from input clock to output is ≈ 10ns, which make D – input stable for 10 ns after the clock transition.

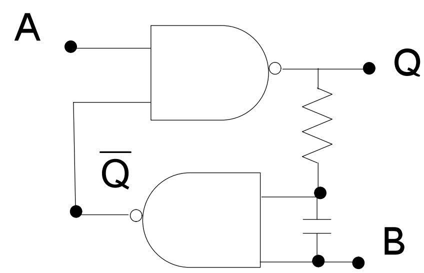
<u>Monostable Multivibrators ⇒ "One – Shot"</u>

Monostable multivibrator (MM) or "one – shot" (O.S.) is variation of the flip – flop in which the <u>output of one of the gate is capacitively coupled to the input of the other gate</u>. Result ⇒ circuit is in <u>one state</u>. If it is forced to the other state by a momentary input pulse, it will return to the original state after delay, which is determined by capacitor and circuit. It is <u>very useful for generation pulses of selectable width and polarity</u>. (It is best to use commercial one – shot as IC)

<u>Examples of O.S.:</u> (not to build)



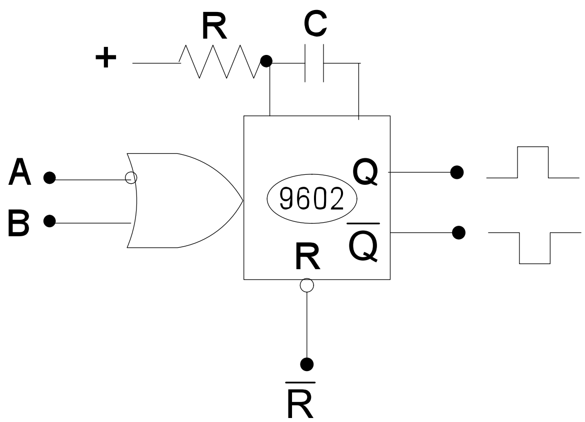
Ⓐ Example

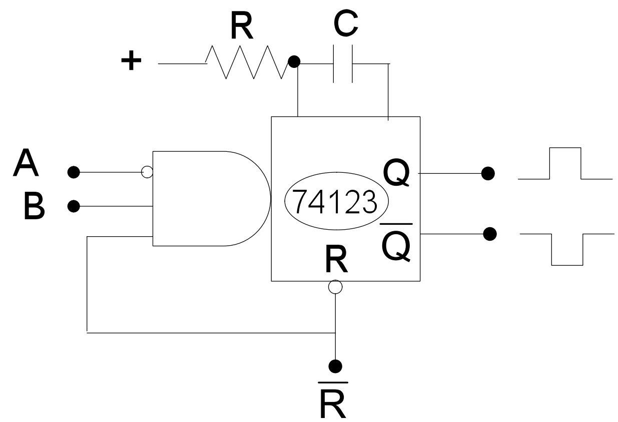


Ⓑ Example

One – Shot (cont.)

Examples (IC):





triggers on these inputs:

A	B
↓	L
H	↑

A	B	$\overline{R}$
↓	H	H
L	↑	H
L	H	↑

↓ : Hight – to – Low
↑ : Low – to - High

} edge triggering

The 9602 is a dual monostable with OR gate at the input, while 74123 is dual monostable with AND gating at inputs.

<u>One – shots</u> are triggered by a rising (↑) or falling (↓) edge at the appropriate inputs.

Triggering signal width: 25 ns $\leq \tau \leq$ 100 ns typically

<u>Oscillators and Timer Chip (555)</u>

Source of regular oscillations ⇒ necessary for any instrument which:

(a) Initiates measurements or processes

(b) Involves periodic states or periodic waveforems

<u>Examples</u>: oscilloscopes, calculators, computers, digital multimeters, almost all digital instruments, etc.

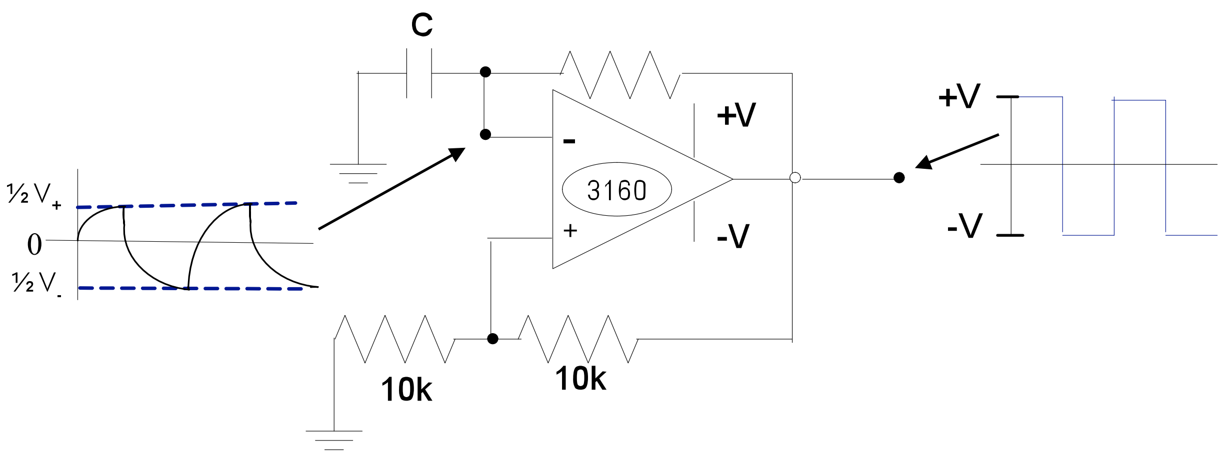
Oscillator may be used as:

- clock (source of regularly spaced pulses)
- frequency counter (time base → use stability and accuracy)
- producer of accurate waveforms (e.g. horizontal – sweep ramp generator in oscilloscope)

Relaxation Oscillators: Very simple oscillator by charging capacitor C through resistor R (or current source) and discharging it rapidly when voltage on C reaches some threshold, starting they cycle again. With appropriate external circuit ⇒ polarity of charging current can by revered when threshold is reached ⇒ generation <u>triangle wave</u> instead of <u>sawtooth</u>.

Oscillators and Timer Chip (555) (cont.)

These relaxation oscillators are simple, inexpensive and quite stable in frequency.



Op – amp relaxation oscillator (Lab)

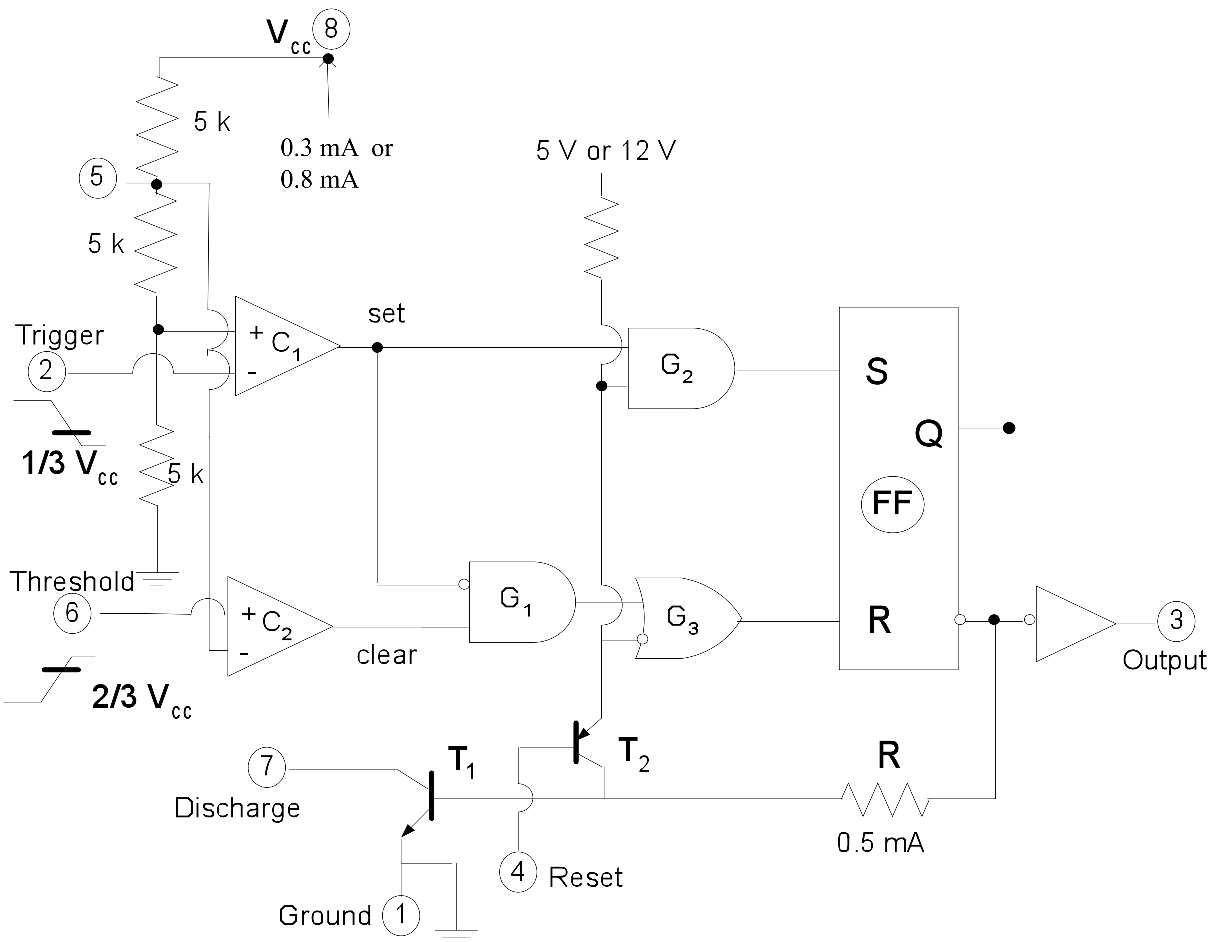
When power is on and op-amp goes to positive saturation (could go to negative, as well) capacitor begins charging toward V_+ with time constant $\tau = RC$. After reaching ½ V_+ of PS, op-amp switches into negative saturation (Schmitt trigger), and capacitor begins discharging → V_- with the same time constant.

IC 555 (Timer)

This 555 chip is very good oscillator with stability ≈ 1 % independent of voltage supply stability. There are other very good IC timers: 556 (dual 555), 7555 (low power → CMOS), 7756 is dual 7555.

Oscillators and Timer Chip (555) (cont.)

IC 555 (Timer)

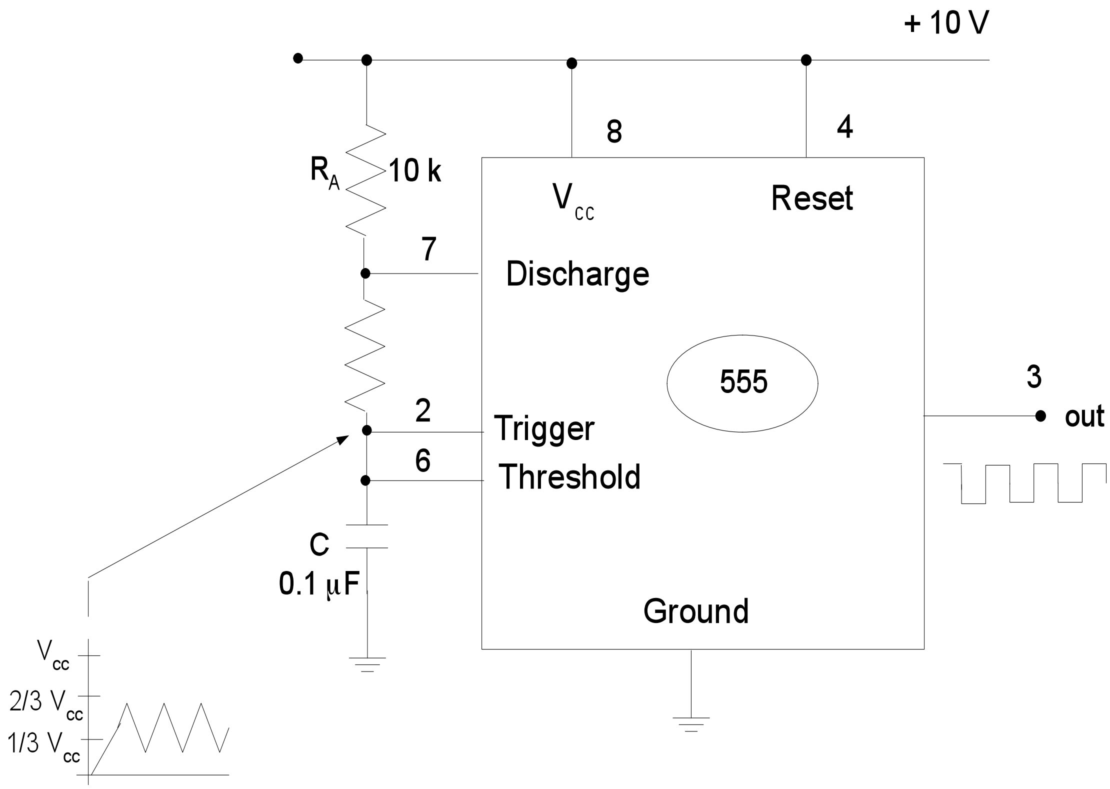


Analysis:

Trigger input → output HIGH stay HIGH until (~ V_{cc})

Threshold input is driven ⇒ output goes LOW (~ 0) and discharge transistor T_1 is turned ON Trigger input is activated by an input level below $1/3\ V_{cc}$, and the threshold is activated by an input level above $2/3\ V_{cc}$.

Timer 555 (cont.)

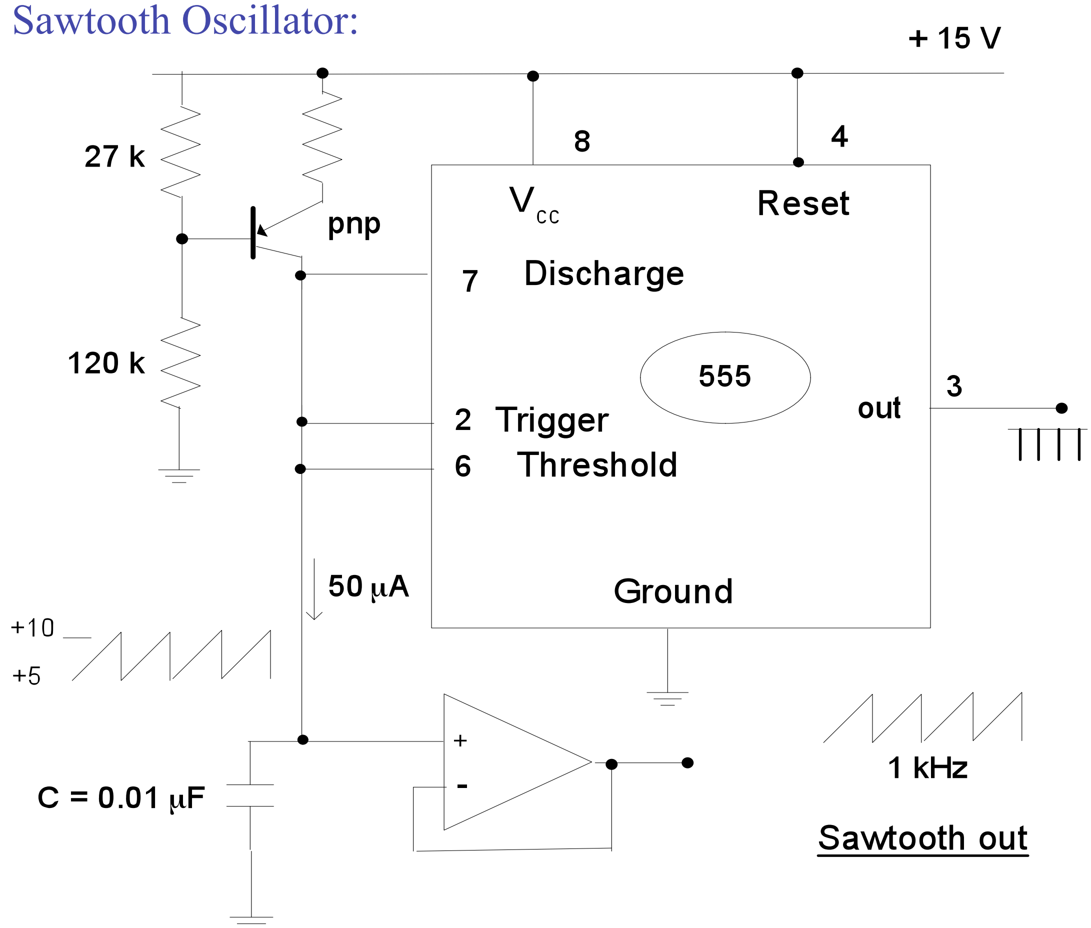


Example of connections provides another look: Power on $\rightarrow$ C is discharged $\Rightarrow$ 555 is triggered $\Rightarrow$ output goes HIGH, and discharge transistor T_1 $\leftarrow$ turn off. C start charging toward 10 V through $R_A + R_B$. When $V_c \geq 2/3\ V_{cc}$, threshold input is triggered $\Rightarrow$ output goes LOW , and T_1 – turns on, discharging C toward ground through R_B.

Timer chips (cont.)

Chip 555 has number of applications as a timer but even more often is used when connected such a way that provides different pulses (one – shot, different width, low-duty-cycle oscillator, sawtooth oscillator, triangle generator, etc.)

Sawtooth Oscillator:



Current source (pnp) for charging capacitor C (timing capacitor) → up to 2/3 V_{cc} . Rapidly discharges through 555's discharge transistor (npn). Op-amp provides buffering for high impedance on C.

Timer chips (cont.)

Presently CMOS are being used in 555 in order to decrease current supply and V_{cc} voltage as well as decrease trigger current.

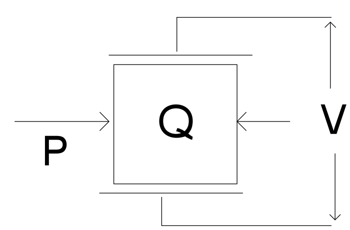
Comparison of specifications (some):

Type	Min V_{cc}	Typ I (μA)	Trig. I (nA)	Freq. (MHz)	I out (mA)
555	4.5 V	3000	100	0.5	200
ICL7555	2	60	~ 0	1	4
TLC 555	2	170	0.01	2.1	~ 0
LMC 555	1.5	100	0.01	3	~ 0

Crystal Oscillators

Crystal oscillators provide excellent frequency stability. Typical frequency stability: 0.001 ppm / °C and 0.1 ppm / V. Quartz crystal is piezoelectric $\Rightarrow$ a strain generates voltage, and vice versa.

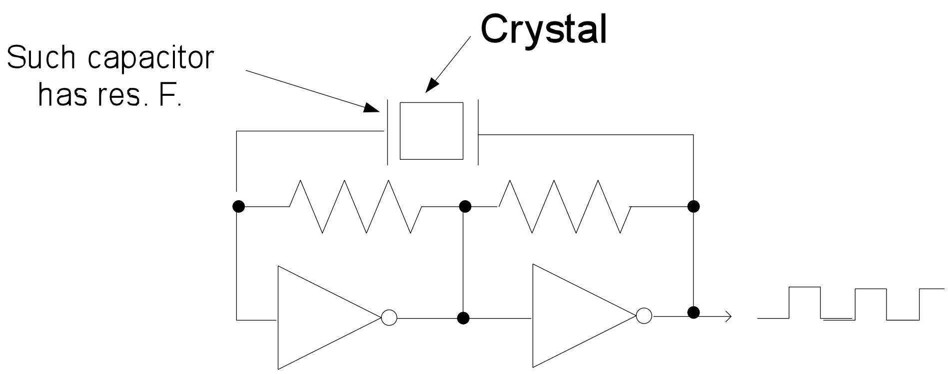
Natural mechanical frequency resonance depends on crystal material, size and cut (shape).



Crystal Oscillators (cont.)

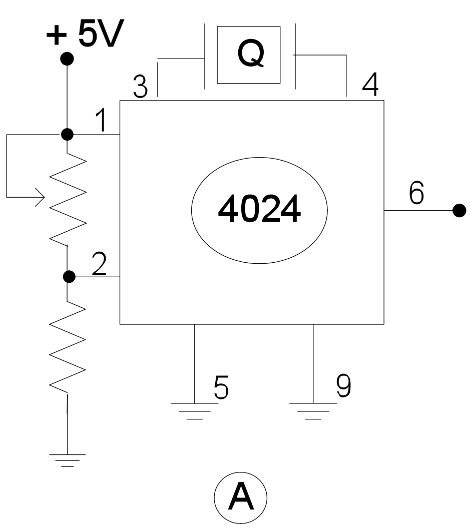
Principle of work $\Rightarrow$ to produce rapidly changing reactance with frequency.

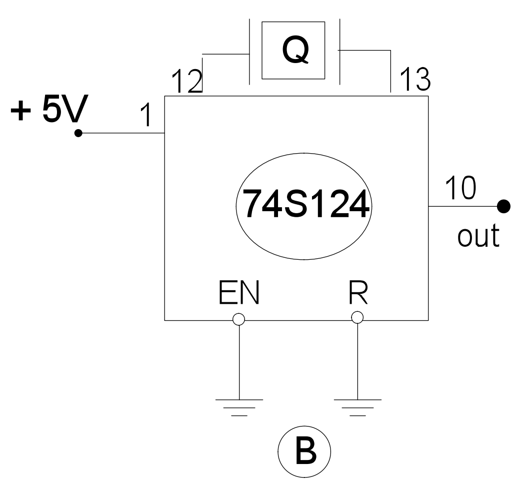
Example (Principle of work):



Oscillations start by noise

Examples (with ICs):





Analog – to – Digital and Digital – to – Analog Conversions

Analog – to – Digital Converter (A/DC) is converting analog signal to an accurate digital number proportional to its amplitude.

For reverse procedure ⇒ Digital – to – Analog Converter (D/AC) converts digital numbers into analog signal (e.g. displays).

A/DC ⇒ for error – and noise – free transmission A/D and D/A converters ⇒ applications in number of everyday electronics (not necessarily complicated electronic apparatus).

We will not build A/D or D/A converters, but we will provide analysis of some conversion methods : looking for advantages and disadvantages of such methods ⇒ selection of commercially available chips (ICs).

Digital numbers ⇒ related to number codes.

Number Codes

Decimal : base 10 $\Rightarrow$ string of integers that are understood to multiply successive powers of 10 and individual products being added together.

Example:

$$116.08 = 1 \times 10^2 + 1 \times 10^1 + 6 \times 10^0 + 0 \times 10^{-1} + 8 \times 10^{-2}$$

Number of symbols $\Rightarrow$ 10 (0 through 9).

Binary : base 2 $\Rightarrow$ two symbols: 0, 1 . Each (0 or 1) symbol multiplies a successive power of 2

Example: Conversion binary to decimal

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

The individual 1 and 0 are called “bits” (binary digits).

The subscript indicates number system (2 for binary and 10 for decimal).♪

Number Codes (cont.)

Conversion decimal to binary $\Rightarrow$ number is dividing by 2 and write down the remainders until last digit

Example: Conversion 13_{10} to binary:

$13/2 = 6$	remainder 1	$\leftarrow$ LSB (least significant bit)
$6/2 = 3$	remainder 0	
$3/2 = 1$	remainder 1	
$1/2 = 0$	remainder 1	$\leftarrow$ MSB (most significant bit)

$$13_{10} = 1\,1\,0\,1_2$$

Hexadecimal ("Hex") representation

In the binary representation the numbers are long $\Rightarrow$ solution: hexadecimal (base – 16) representation. In "hex" each position represents successive power of 16 with symbols having value 0 $\rightarrow$ 15. In order to have a single symbol for each hex position.

The values 10 – 15 are represented by A – F symbols.

Binary number in hexadecimal $\Rightarrow$ 4 – bit group with hex equivalent for each group.

Hexadecimal ("Hex") representation (cont.)

Example:

$$707_{10} = 1011000011_2 \Rightarrow \overbrace{10}^{2}\ \overbrace{1100}^{12}\ \overbrace{0011_2}^{3}$$

$$\underline{2\ C\ 3_{16}} = 2\ C\ 3_H \quad (H \Rightarrow 16)$$

$$2\ C\ 3_{16} = 2 \times 16^2 + 12 \times 16^1 + 3 \times 16^0 = 707_{10}$$

Hexadecimal representation is well suited to the popular "byte" (8 – bit) organization of computers $\Rightarrow$ 16 – bit or 32 – bit computer "word" $\Rightarrow$ hence word is 2 – or 4 – byte unit.

In hexadecimal → byte is 2 hex digits, and 16 – bit word is 4 hex digits.

Octal – base 8 $\Rightarrow$ not in use anymore.

Binary – coded Decimal (BCD)

BCD $\rightarrow$ encoded each decimal digit into binary; it requires 4 – bit group for each digit.

Example:

$$126_{10} \Rightarrow \underbrace{0001}_{1}\ \underbrace{0010}_{2}\ \underbrace{0110}_{6}\ \text{(BCD)}$$

Of course BCD is not the same as binary representation

$$126_{10} \Rightarrow 1111110_2$$

In BCD bit position (starting from right) can be considered as: 1, 2, 4, 8 (or $2^0, 2^1, 2^2, 2^3$) (in above example { 6 } is $0 + 2 + 4 + 0$), next as 10, 20, 40, 80, and next as 100, 200, 400, 800, etc. (in above example: { 2 } $\Rightarrow$ 20 ; { 1 } $\Rightarrow$ 100).

In BCD $\Rightarrow$ wasteful of bits $\rightarrow$ 4 – bit group can represent numbers $0 \rightarrow 15$, whereas in BCD only $0 \rightarrow 9$. However, BCD is ideal for displaying a number in decimal $\Rightarrow$ simple conversion BCD character to appropriate decimal number. Difficult conversion between binary and BCD.

Gray Code

In gray Code only one bit changes when going from one state to the next.

It is simple and prevents errors.

Often is used in mechanical shaft – angle encoders.

Generating Gray – code states: start with state of all zeros; next state → change single least significant bit, which provides as new state:

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000

No limit on number of bits.

Useful in "parallel encoding" → high speed A/DC.

American Standard Code for Information Interchange ⇒ ASCII

Signed Numbers
- Offset binary representation and
- 2's Complement representation

Signed numbers $\Rightarrow$ for computations (we need "+" and "-" numbers).

Offset binary and 2's complement will be discussed here. (Offset binary popular for A/DC and D/AC).

2's complement $\rightarrow$ in integer computation.

In offset binary numbers are running from most negative to most positive (e.g. from – 8 to + 7 in 4- bit singed integers) numbers.

In 2's complement numbers are running from zero to most positive number and from zero to most negative number by "complement" to positive numbers $\Rightarrow$ positive numbers are simple binary (unsigned) numbers.

Signed Numbers

4 – Bit signed integers in offset binary and 2's complement representation

Integer	Offset binary	2's complement
+7	1111	0111
+6	1110	0110
+5	1101	0101
+4	1100	0100
+3	1011	0011
+2	1010	0010
+1	1001	0001
0	1000	0000
-1	0111	1111
-2	0110	1110
-3	0101	1101
-4	0100	1100
-5	0011	1011
-6	0010	1010
-7	0001	1001
-8	0000	1000

MSB

complemented;   MSB carries the sign information

In 2's complement:

negative number ⇒ negative number & positive number = 0; e.g.: +3 – 3 ⇒ 0011 + 1101 = 0000

Negative number : (a) complement each bit of the positive number (1 for 0 and 0 for 1 → "1's complement"), (b) add 1 ("2's complement").