Architecture of Computers and Parallel Systems

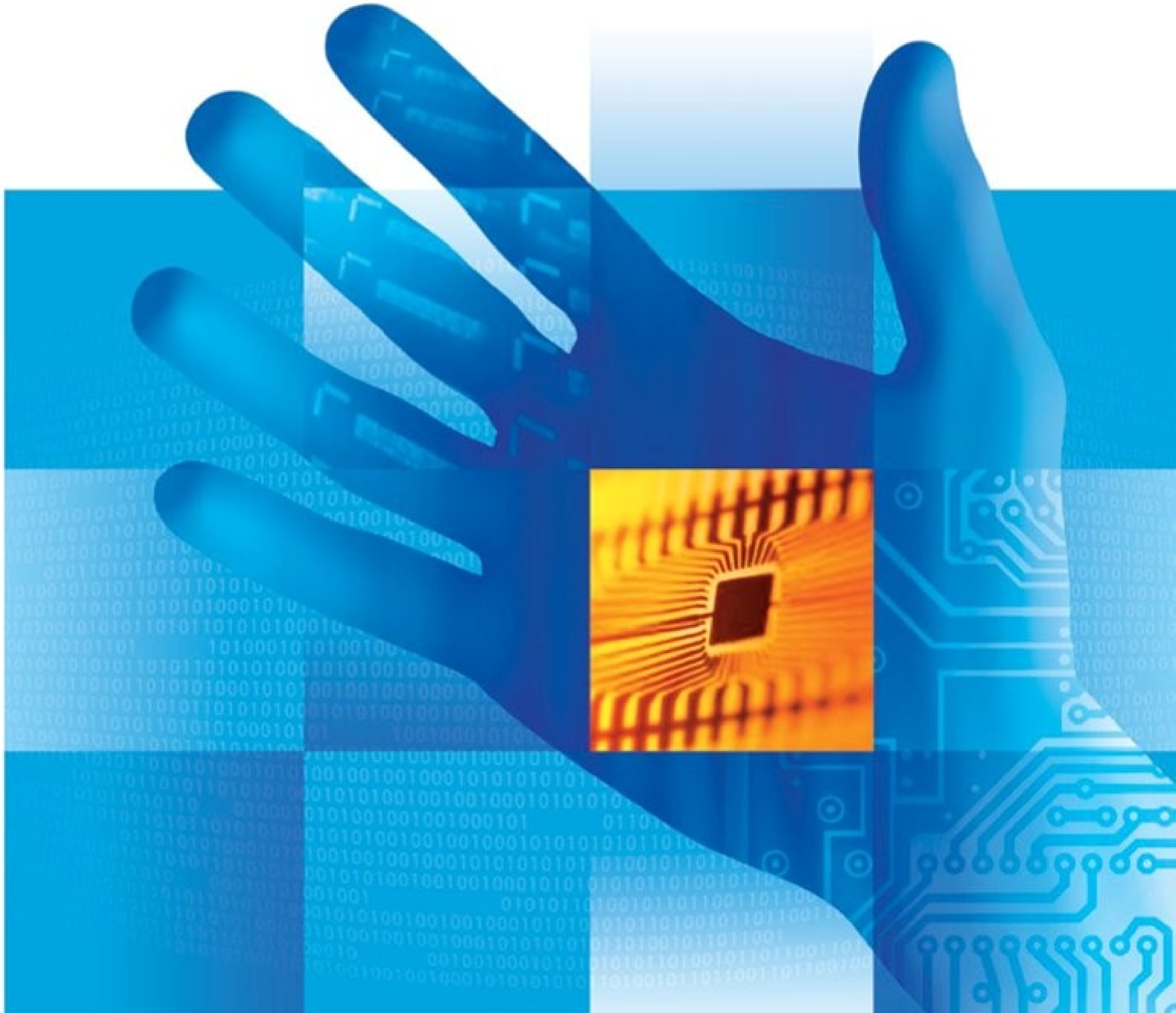
Part 6: Microcomputers

Ing. Petr Olivka

petr.olivka@vsb.cz

Department of Computer Science

FEI VSB-TUO



Architecture of Computers and Parallel Systems

Part 6: Microcomputers

Ing. Petr Olivka

petr.olivka@vsb.cz

Department of Computer Science

FEI VSB-TUO

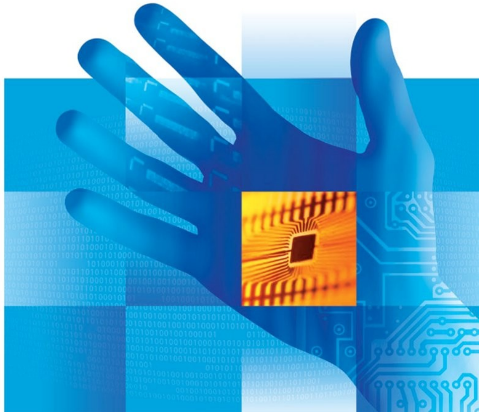
european social fund in the czech republic

EUROPEAN UNION

MINISTRY OF EDUCATION, YOUTH AND SPORTS

OP Education for Competitiveness

INVESTMENTS IN EDUCATION DEVELOPMENT



Architecture of Computers and Parallel Systems

Part 6: Microcomputers

Ing. Petr Olivka

petr.olivka@vsb.cz

Department of Computer Science

FEI VSB-TUO

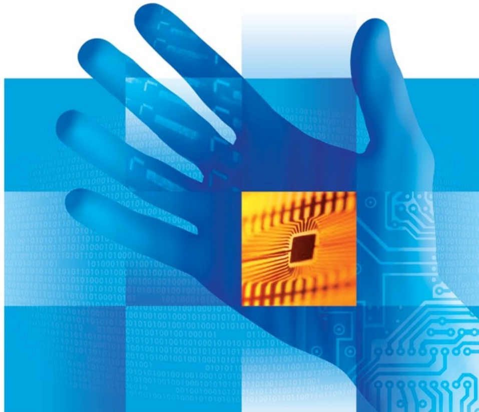
evropský sociální fond v ČR

EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Microcomputers

The microcontroller, microcomputer, monolithic computer, single-chip computer, SOC – System on Chip, etc... are technical terms used to describe computer integrated in one package. As we already know, a computer consists of three inseparable parts: the processor, the memory and devices.

Microcomputers are made for many low-end and middle complex applications for more than 30 years. There are tens of manufacturers of small computers, which produce a wide range of microcomputers.

Thanks to continually falling prices and improving technology, microcomputers are practically used in all devices around us: in battery chargers, lights dimmers, in bike computers, remote-controllers, washing machines, dishwashers, food processors, vacuum cleaners, cars and many others...

Microcomputer Architecture

All computers are designed according to the von Neumann or Harvard architecture. In most cases the second one is used for microcomputers. Microcomputers are usually controlled by a fixed program and they use data from their environment. The program is stored in the nonvolatile memory and data are stored in registers or the SRAM memory. Therefore, Harvard architecture is the natural choice.

Older microcomputers were designed as CISC processors, but now all newest ones are designed as RISC. Programmers now prefer the C-language to the assembly language and so more complicated programming of RISC processors does not matter to anyone.

Every microcomputer is slightly different, but we will try to characterize the microcomputer and find their general features and properties.

Microcomputer Memories

As mentioned before, microcomputers contain two main memories. The first one is for program and the second one for data. The program is usually stored in some version of nonvolatile memory. In older microcomputers manufacturers implemented EEPROM and the chip had a typical small window on the top for UV light erasing. Now, the Flash memories are used in most microprocessors and for the final applications the OTP memory can be used.

RWM and Flash memories are used for data. While the program is running, it stores data in the RWM, but sometimes it is necessary to store some measurement, runtime data or corrections for the future usage in the Flash memory.

The RWM memory can be usually divided into three layers, depending on frequency of data usage.

… Microcomputer Memories

- **Working registers** – the first level of RWM. Generally, the microcomputer has one or two working registers, but it can have tens of them. The registers hold current (temporary) data All instructions from the instruction set can (or must) use them. Instructions are usually hardwired to use specific register(s).
- **Universal scratchpad registers** – here the program stores the most frequently used data. The part of instruction set is able to manipulate with these registers. But the format of instructions does not allow a wide address range. The limit is usually in the range from 16 to 256 registers. Some manufacturers solve this problem by creating register banks. The programmer can then switch between banks as required. In some processors, one selected bank of registers is used during the interrupt execution.

… Microcomputer Memories

- **RWM (or Flash) Memory –** is used for larger or less-used data storage. The instruction set usually does not allow direct manipulation with the memory content, except movement instructions. These data have to be moved to working registers. Instructions of some processors allow to use data from this memory as the second operand. The result can not be stored directly back to this memory. In some microcomputers, the RWM memory is not implemented and we have to use some external memory.

The microcomputer can implement some other types of registers or memories: instruction pointer, flags, control registers, stack for storing return address, etc.

Ports of devices are very often implemented in address space of scratchpad registers or RWM.

Microcomputer Synchronization

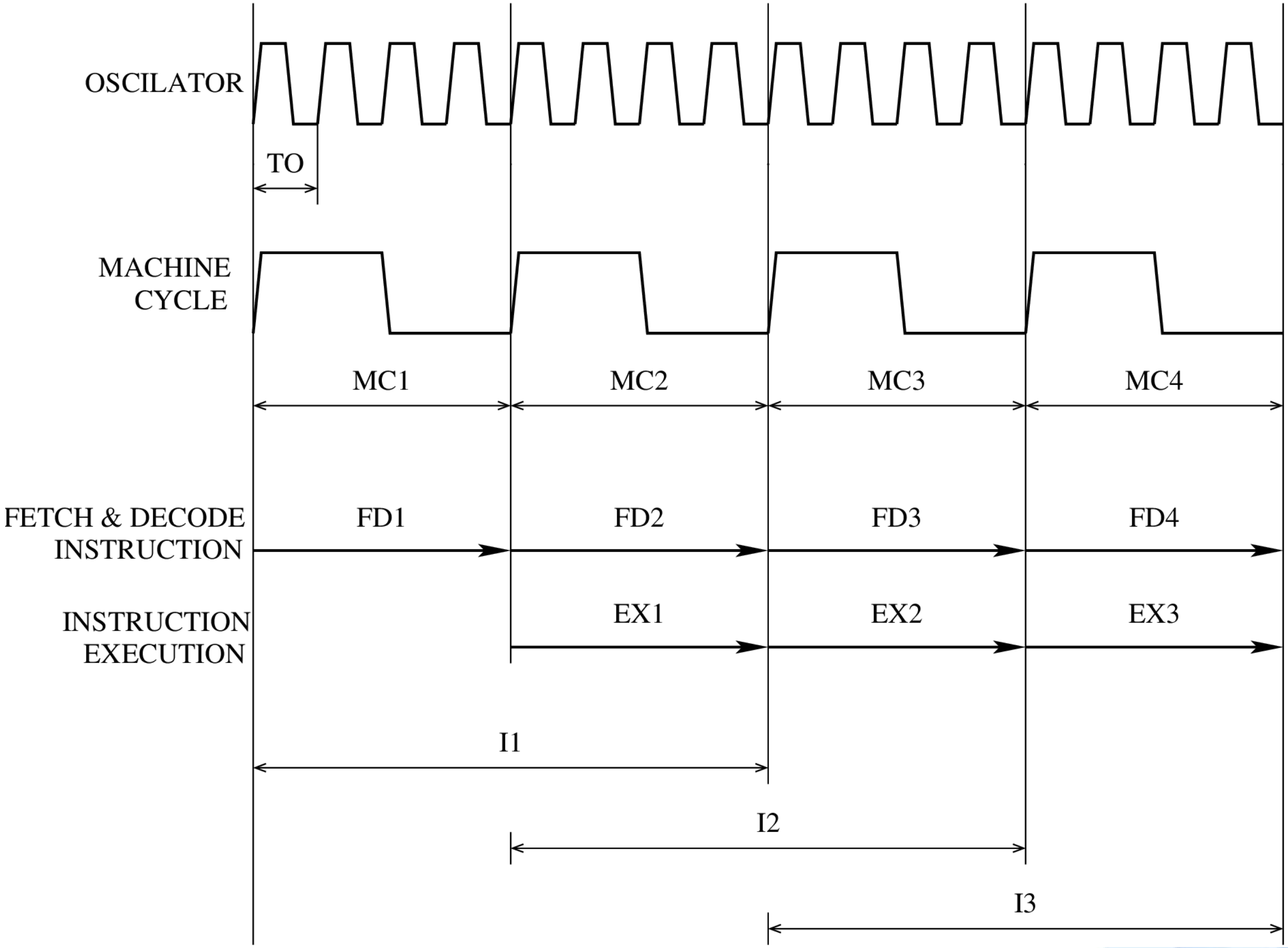
Processors are sequential circuits and they are controlled by the clock signal, including microcomputers. The synchronization is very important, because microcomputers are used to control technical applications, where the time plays often an important role. Manufacturers have to describe very exactly the timing of all circuits inside the computer.

The scheme on the next slide provides an example of how the processor is controlled. It shows how many clock periods the CPU needs for one machine cycle and how many machine cycles the CPU uses to execute one instruction. If the processor uses pipelining, the dependence between machine cycle and each stage of instruction execution has to be described.

It is the only way to accurately calculate, how long the instruction is performed.

In the next scheme we will use PIC processor to illustrate the timing.

… Microcomputer Synchronization



… Microcomputer Synchronization

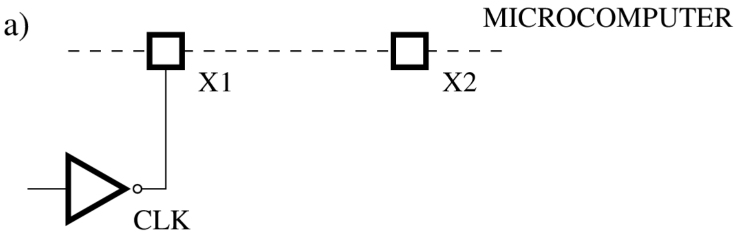
The previous scheme shows the clock signal with the period T0. Four clock periods are necessary for one machine cycle. Individual machine cycles are marked MCx. The scheme depicts RISC processor PIC with 2 stages (FDx and EXx) and both stages require one machine cycle. Thus one instruction cycle (Ix) needs two machine cycles (MCx).

As is evident from this description, the timing is primarily determined by the oscillator.

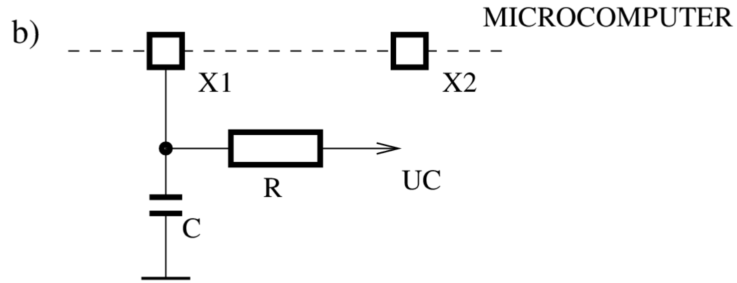
Microcomputers are currently equipped with the ability to use more types of clock sources, even the internal oscillator. The microprocessor has usually reserved two pins marked as X1 and X2 for clock source.

Three common clock sources are on the next scheme:

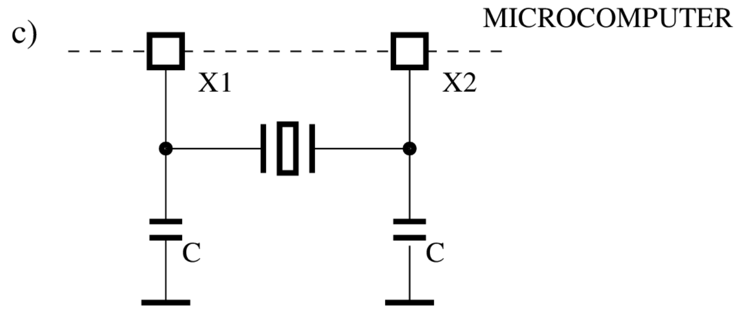
… Microcomputer Synchronization



- External clock source



- RC Cell



- Crystal

… Microcomputer Synchronization

- **RC circuit** – is the cheapest solution with higher inaccuracy. Because the timing depends on the value of capacitor and resistor, it is not possible to have accurate timing. The value of both passive parts changes with temperature and moreover with capacitors age.
- **Crystal** - is produced in wide range of frequencies and with very high accuracy. But its speed is fixed and it can not be changed. The disadvantage is to have other parts on the board: crystal plus two addition small capacitors.
- **External clock source** – is used in many applications as clock source from other parts of circuit. It can be also produced by special oscillating circuits that can be adjusted during operation. The dynamic decreasing of frequency can significantly reduce microprocessor's power consumption, which is important for battery-powered applications.

RESET of Microcomputer

The processor is a sequential circuit. The behavior of the processor does not depend only on machine instructions, but also on its internal states, which affects the results. Therefore it is very important to define an initial state of all circuits in the processor and computer.

Many think that the reset is only the button on the case. The reset is the initial state defined very exactly in a technical documentation for each processor and its peripherals.

The correct reset state is achieved by the reset signal with a minimum prescribed period, because the reset is the sequence of consecutive activities in all computer circuits. The most of registers are set to 0. Very important is the value in the instruction pointer. The first instruction of a program has to be on the given fixed address! There is the first program instruction.

The reset signal is usually external, but in some critical situations it can be generated internally by an auxiliary circuit.

Protection Against Noise

Microcomputers are used in many applications where an environment for its operation is not always good. Therefore the noise protection of the microcomputer must be focused on multiple perspectives:

- Physical impacts – the microcomputer has to be able to work in wide range of temperature, air pressure, humidity, acceleration,...
- Mechanical resistance – the chip and its package has to be strong enough to allow reliable mounting on the board.
- Electrical resistance – the chip has to be able to work in wide range of voltage, because during the operation the voltage can significantly fluctuate. Microcomputers are today working normally in the range of voltage from 2.5 to 6V.
- Electromagnetic resistance - high currents in environment can cause the unwanted current induction. Therefore it is necessary to shield processors.

… Protection Against Noise

Not only the environment surrounding the computer is noisy. The main problem is very often in the program and the programmer. Because the program in the computer runs many hours or days, some registers in the computer or device can unexpectedly change their own states. Many times programmers make errors in the program and it does not work correctly. In both cases the result is an infinite loop.

For this purpose the microcomputer has special independent circuit called WatchDog. If the program stops communicating with this circuit, the watchdog resets the microcomputer. The implementation is simple: the watchdog is a counter that is automatically and slowly decremented. When it comes to zero, it generates a reset signal. Therefore the program must only rarely rewrite value in the watchdog register to its maximum.

Power supply is monitored in a similar way. When it drops below the allowed level, then the specific circuit stops the computer and resets it. This guard is BROWNOUT.

Interrupt Subsystem

No computers can exist today without the interrupt support, including microcomputers. Interrupt allows effective responses to events from devices and it simplifies programming.

The programmer should know some basic features of an interrupt subsystem. The first one is how to enable and disable interrupts. Then how to recognize, which device is the source of the interrupt and what must be done in the program within the interrupt. Usually a sequence of instructions is recommended. Programmer would also be interested in whether the interrupt can be nested and whether it is possible to set the interrupt priority. The possibility of nesting is critical, because it can cause looping.

It is important to keep in mind that the main program can be interrupted at any time and anywhere. It is therefore important that the interrupt service always carries out only the necessary and then it quickly passes control back to the main program.

Devices - Ports

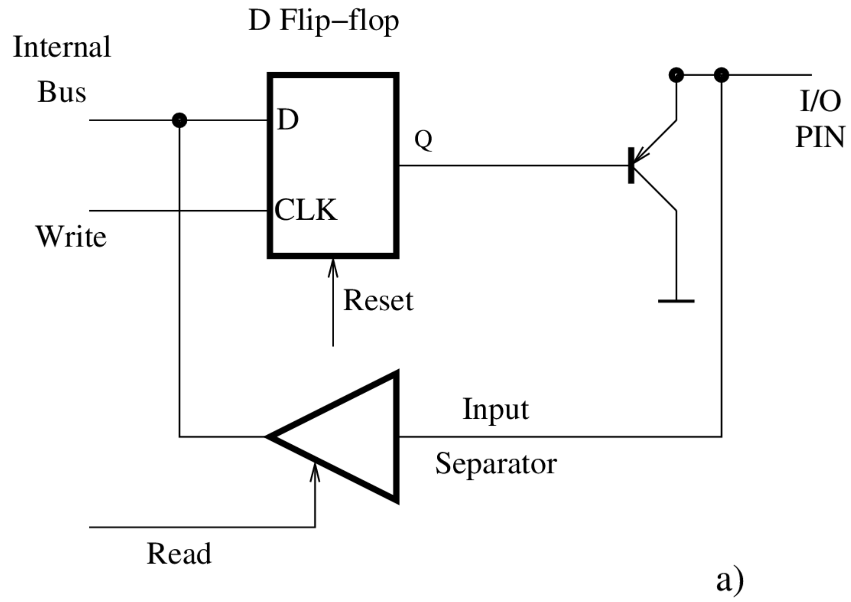
The computer is a processor, memory and - the third part – peripherals. Now we present several common peripherals. The first one, very important and the most widely used, is **Parallel Port** (or Gate). Usually they are organized in 4 or 8 bit groups, accessed by registers. For every bit, there is one pin assigned on microprocessor package. All ports can be set for the input or output (after reset the port is set to input mode for safety reasons).

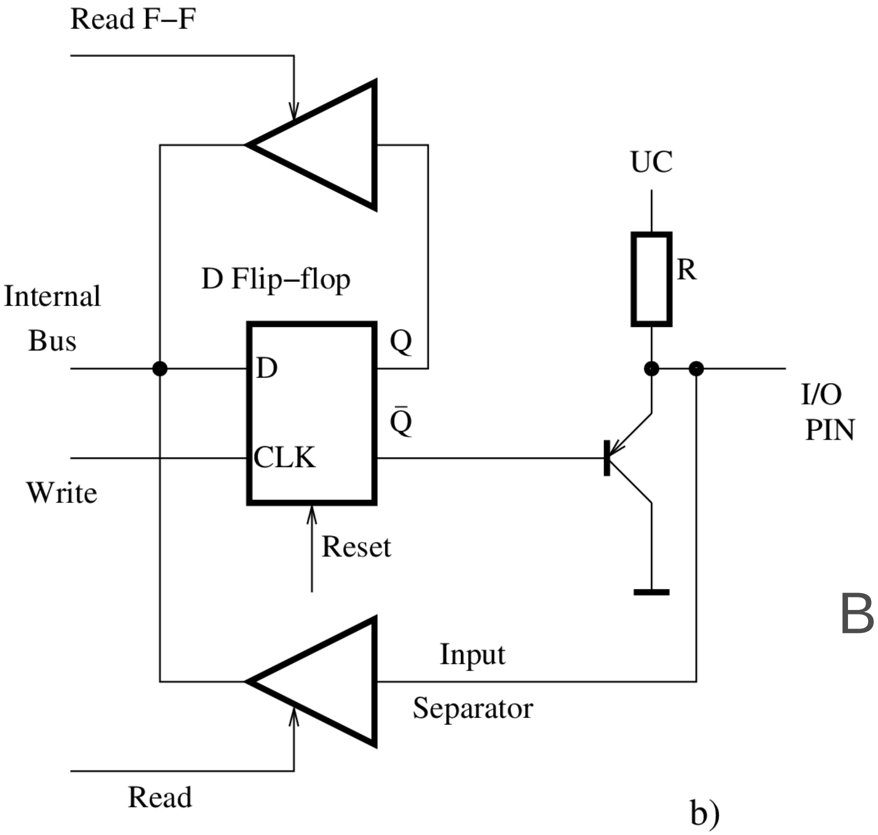
The internal wiring always significantly affects the electrical properties of the ports. Therefore manufacturers publish the wiring together with a table of electrical parameters. It is then easier to design the right electrical connection.

On the following two diagrams there are examples of internal I/O circuits. First one is with open collector and second one is typical example of bidirectional port.

... Devices - Ports

Port with open collector





Bidirectional port

… Devices - Ports

On the previous scheme a) is the open collector. It is very useful, because it allows connection to the circuit with the higher voltage than the microcomputer is powered. The closed output transistor set this output to a high-impedance state.

The diagram b) shows the example of bidirectional port. When the output transistor is closed, the output has a defined logical value. Programmers can read two values: the last written value to the D flip-flop or the real state on the output pin.

The port implementation allows in some computers to manipulate all bits separately. In other computers the programmer has to use bitwise operations.

Ports can be used for many purposes and they can simulate the behavior of other peripherals – serial ports, D/A converters, PWM, LCD controller, etc. But the simulation has to be done by the program and it decreases the available performance for the main application.

Serial Communication

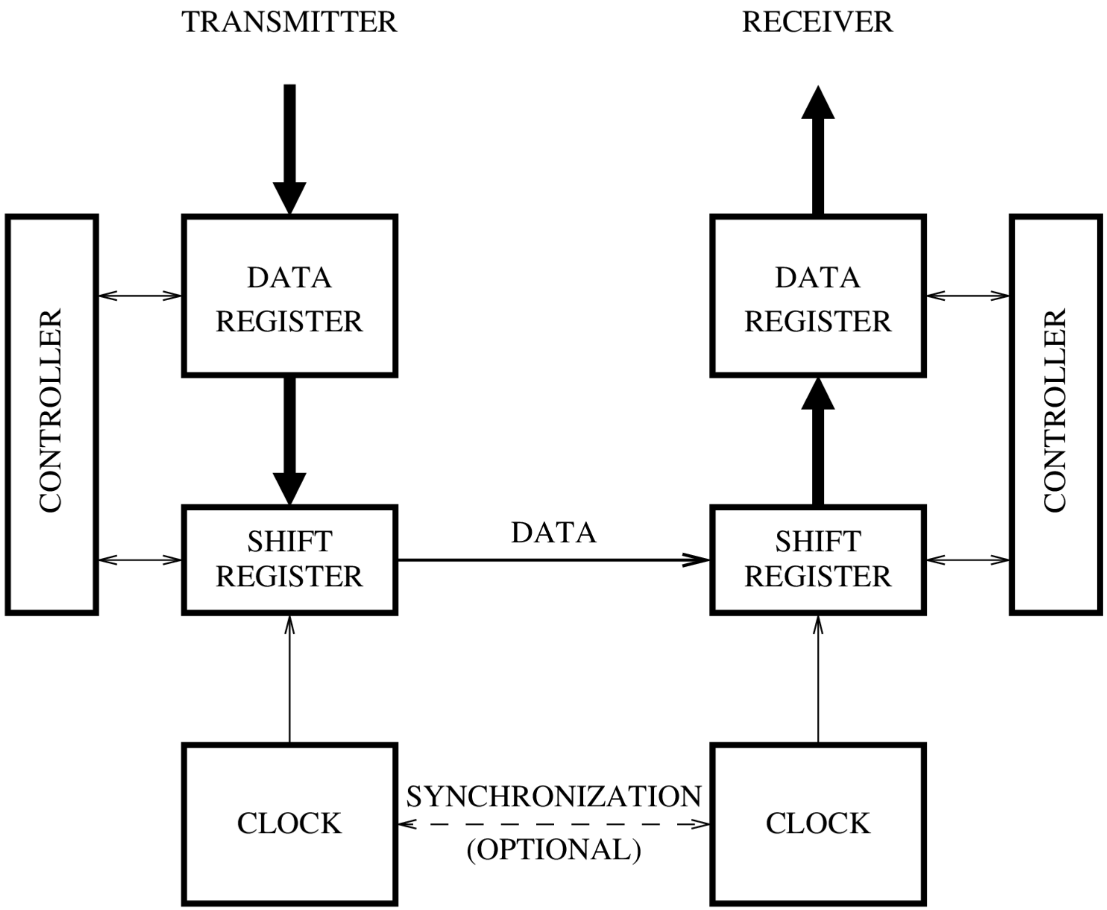
There are many types of serial interfaces. The parallel transfer on the bus is used only for short distances inside computers. The serial transfer needs only a few wires, it can be used for longer distances and it has a much easier controller. The disadvantage is the lower transfer speed. But today the newest semiconductor technology is able to use the highest speed.

In computers and industrial technologies there are many standards for serial transmissions. They differ with speed, voltage, number of wires, one-way or bidirectional, connectors, how many devices are connected, etc...

Standards used for serial communication on short distances are USB, (Universal Serial Bus), RS232 (12/5/3.3V), I^{2}C (Inter-Integrated Circuit), 1-Wire, etc...

Standards RS422, RS485, CAN Bus, Etherner or other networking technology are used for communication on longer distances.

Serial Transmission Scheme



… Serial Transmission Scheme

On the previous diagram there is the principle of serial communication. The processor puts data to the transmitter's data register (or buffer) and the controller in the serial interface passes it to the shift register. The data from this register are transmitted bit by bit to the shift register on receiver's side. When one transfer unit (byte, buffer, packet) is completed, then the receiver's controller passes received data to the data register (or buffer).

The serial transfer is usually performed in two basic modes:

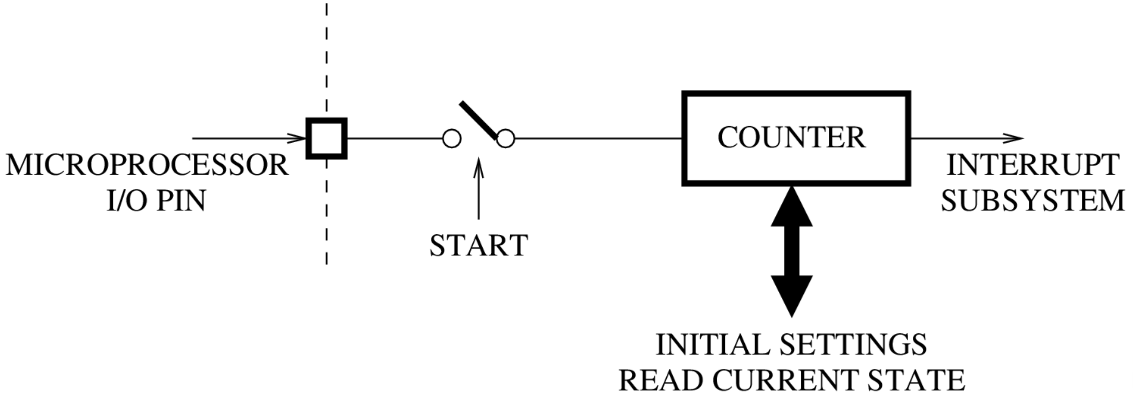
- Synchronous – one wire in serial line is used as clock signal.
- Asynchronous – usually first added bit synchronizes independent clock sources in both serial interfaces.

The communication with the clock signal is safer and easier. But it uses one extra wire and it is more expensive (especially for large distances). The asynchronous mode adds the synchronization data to the transmission, thus it decreases the transfer speed.

Timer, Counter

Another device implemented in all microcomputers is Timer and Counter. Both of them are very similar. They differ in one fundamental feature. The timer is incremented by the internal processor clock signal. Whereas the counter counts changes of the external signal.

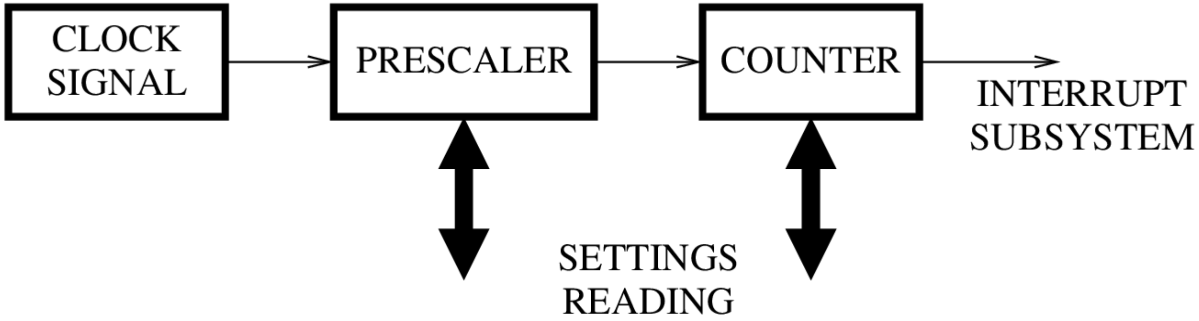
The diagram of counter is shown here:



Programmers set the initial value and turn ON counting. When the value inside the counter overflows, the interrupt is automatically generated. Then programmers decide what will follow. The counter is very useful when we need to count some external events (wheel revolutions, button presses and the like).

... Timer, Counter

The timer is used more often. It works in a similar way as the counter and its diagram follows:



The timer has more settings. Programmers can set a prescaler. It is a circuit in front of the counter to slow down the clock signal. The prescaler can be changed dynamically. In many microcomputers it is possible to select the clock source. The counter at the end of this chain works in the same way as the normal counter.

The timer is used to generate periodic events – blinking LEDs, refresh display, check input signal, scan buttons state, etc...

D/A Converters

Computer world is digital. The surrounding environment contains continuously changing physical quantities. That is why it is necessary to convert the analog value to digital ones - that computers understand and vice versa.

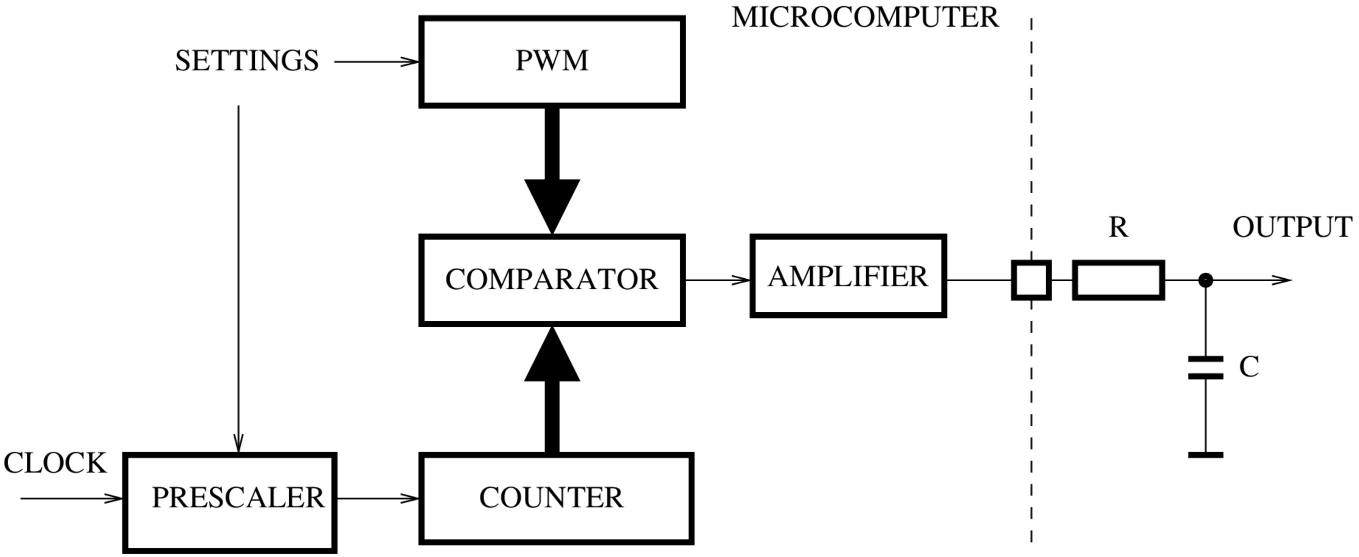
The easiest D/A converter is PWM – Pulse Wide Modulation. The principle is simple. The output signal from the microcomputer has two values: logical “0” - 0V and logical 1 – 5V. And how do we create the continuous signal from them? By switching the 0 and 1 in the unequal ratio of the time. For example, if we need 1V on the output, then the time period of 0 and 1 must be divided in the ratio of 4:1 (Duty Cycle). Average value is then 1V.

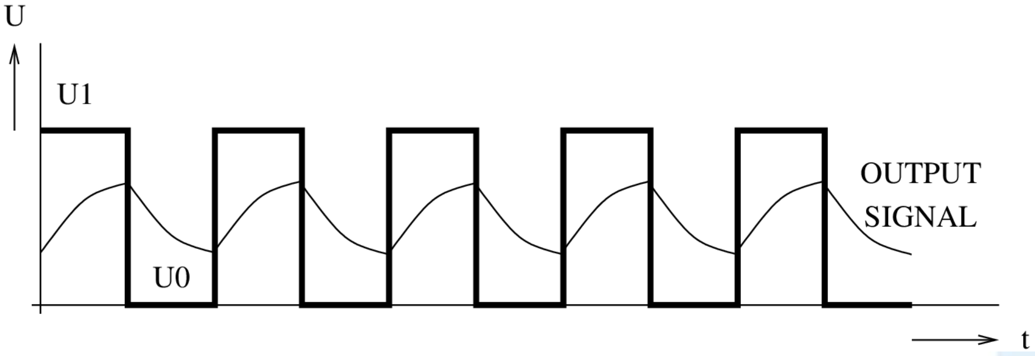
To smooth the signal we have to connect the RC circuit. This circuit must satisfy one important condition:

$$R \cdot C \gg T$$

But the RC circuit is causing small delay. The diagram follows:

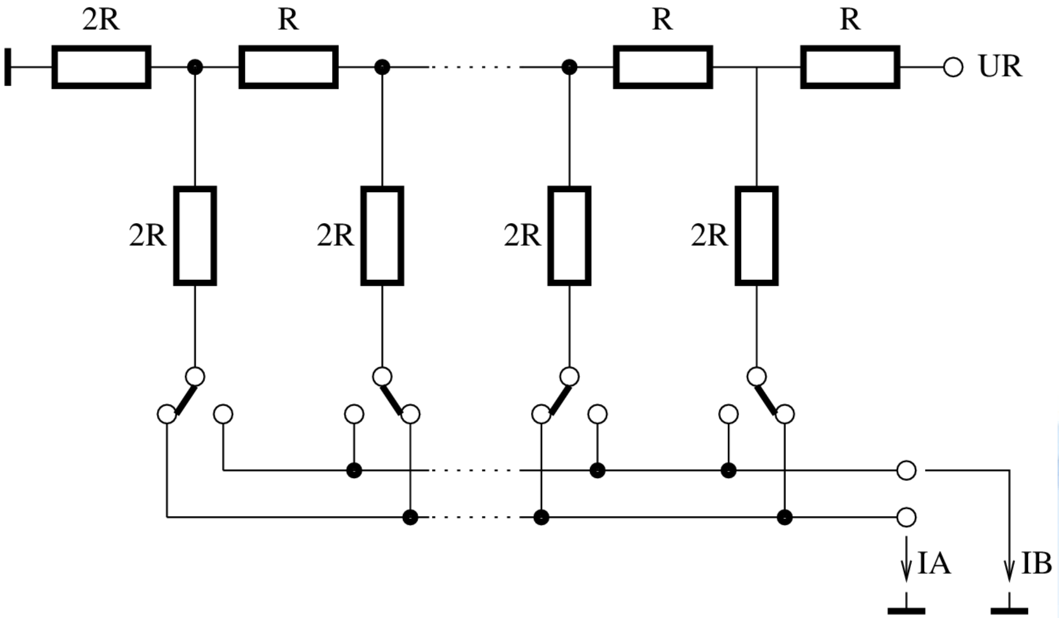
D/A Converters - PWM





… D/A Converters

The often used D/A converter is parallel converter. It is very fast and it uses only passive components – resistors. It is possible to build the converter with resistors in the ratio of 1:2:4:8 … :128…. But these resistances, however, are hard to produce. Thus the solution with only two different resistors – R and 2xR – is used for converter. The circuit is showed below:

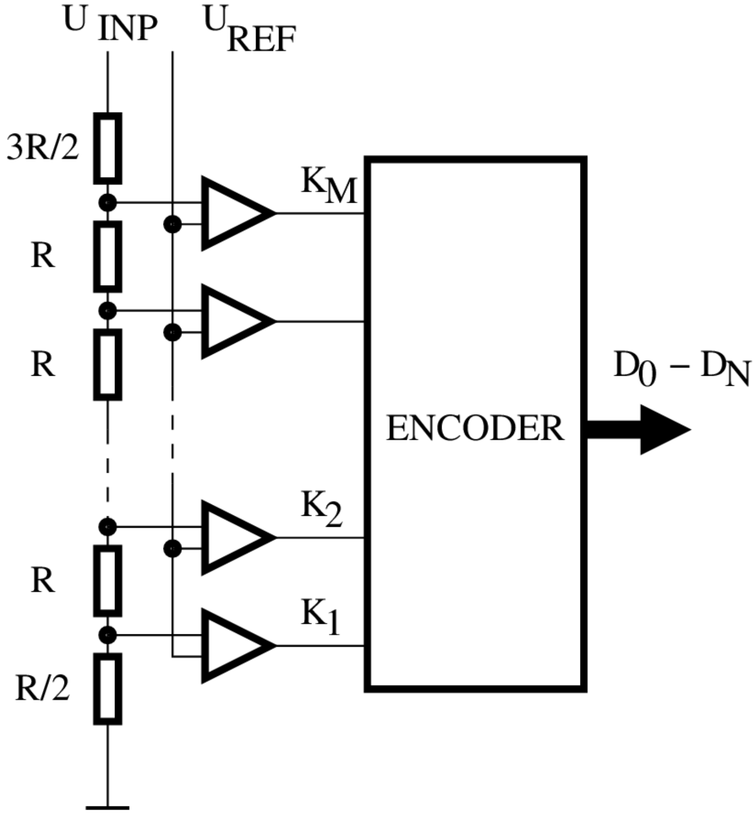


A/D Converters

Many applications use in their own control system physical quantities from the environment. Measured values are in most cases represented by current, voltage or resistance. They are able to convert all values to voltage and then use A/D converter. The conversion has always small delay. The result is then a number in digital form.

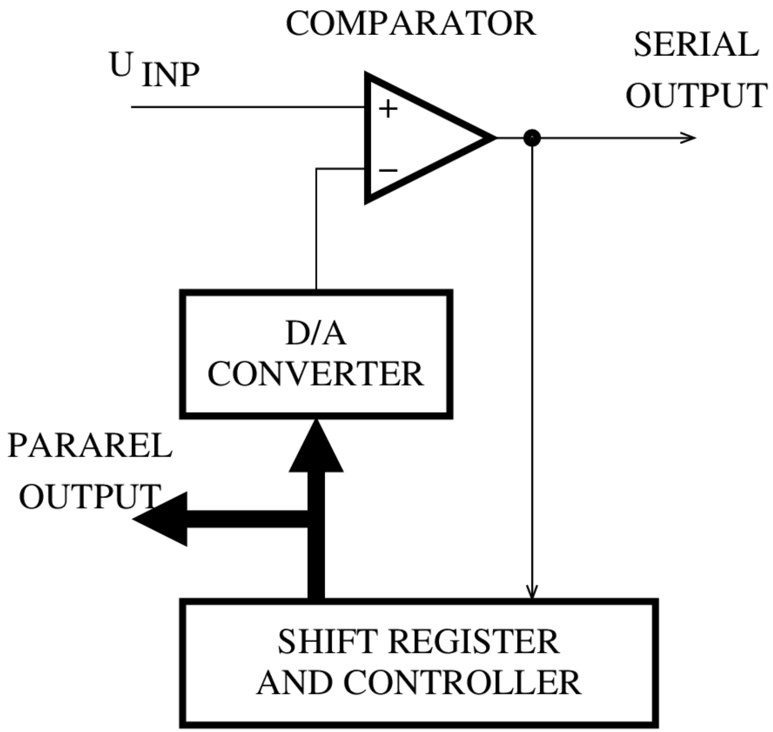
The first simple and fast A/D converter is Comparative converter (see next slide). The measured voltage is divided by the series of resistors to many levels. Comparators are in all nodes between resistors. All comparators accept the same voltage. Two adjoining comparators detect required voltage on one resistor. The converter output value is a sequence number of that comparator. The comparative converter works very fast, because the voltage on resistors changes immediately, all comparators work parallelly and the result is generated directly without delay. But there is a problem with accuracy. The converter with 8 bits resolution requires 2^8 (256) comparators and it is not suitable for better resolution.

Comparative A/D Converters



The comparative A/D converter is on the scheme above. The reference voltage is U_{REF} and input voltage is U_{INP}. K_M comparators are implemented in circuit and the result is D_0-D_N bits ($M=2^N$).

A/D Converters with D/A Conversion



The basis of this converter is only one comparator and auxiliary D/A converter. The production of D/A converter is easier, so why not use it for opposite operation?

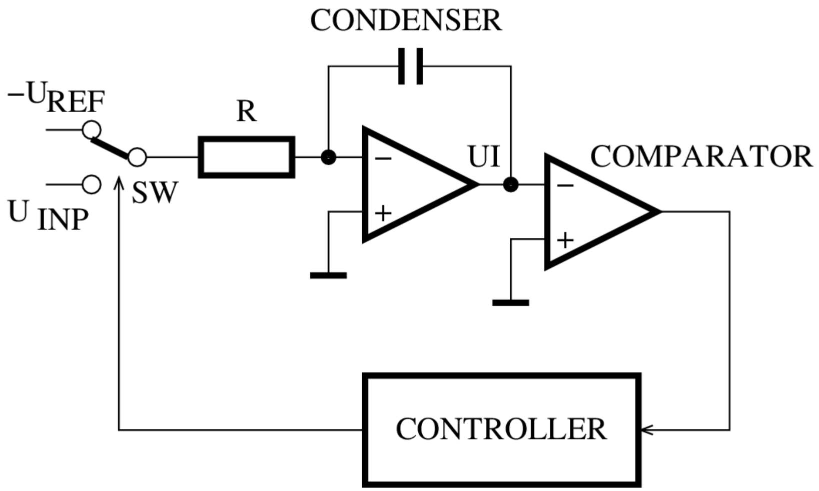
… A/D Converters with D/A Conversion

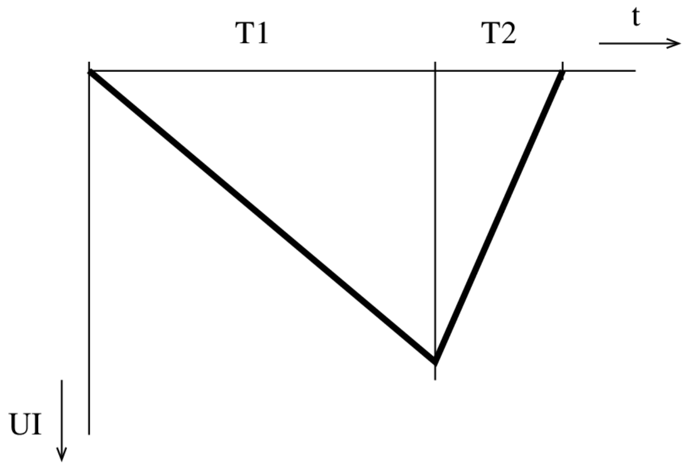
The principle of operation is based on the incremental approximation of the auxiliary D/A converter. Its output is compared with the input signal by a comparator. The output of the comparator indicates whether auxiliary voltage is greater or smaller than the input voltage.

The incremental approximation is performed by a controller with a shift register. The 8 bit resolution requires to perform up to 256 comparisons. These converters are therefore suitable for measuring slowly changing physical quantities (temperature, humidity).

But the converter can work much faster using a better approximation algorithm. Every programmer knows the algorithm for searching value in sorted array – binary search (half-interval search). This algorithm does not change value in the D/A register incrementally, but it starts looking it from the middle. When it is above or below, it moves by half of the interval in that direction. So only 8 steps are necessary for 8 bit resolution.

Integrating A/D Converters





The scheme above shows an integrating A/D converter and timing diagram.

… Integrating A/D Converters

The integrating A/D converters do not directly measure the voltage, but they measure the time of charging and discharging the capacitor.

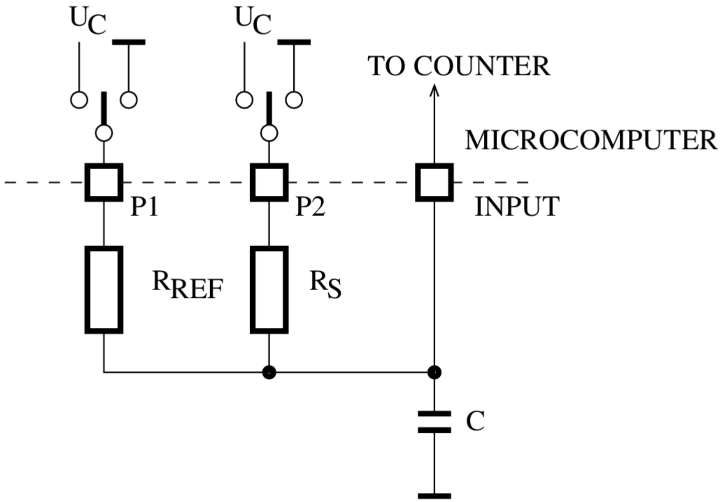
In the first step the capacitor is connected via the switch SW and resistor R to the input voltage U_{INP}. The controller allows charging for a time period T1. Then the SW connects the capacitor to negative reference voltage $-U_{REF}$ and the controller measures the time of the discharging T2. We say that this is a “method of double integration”.

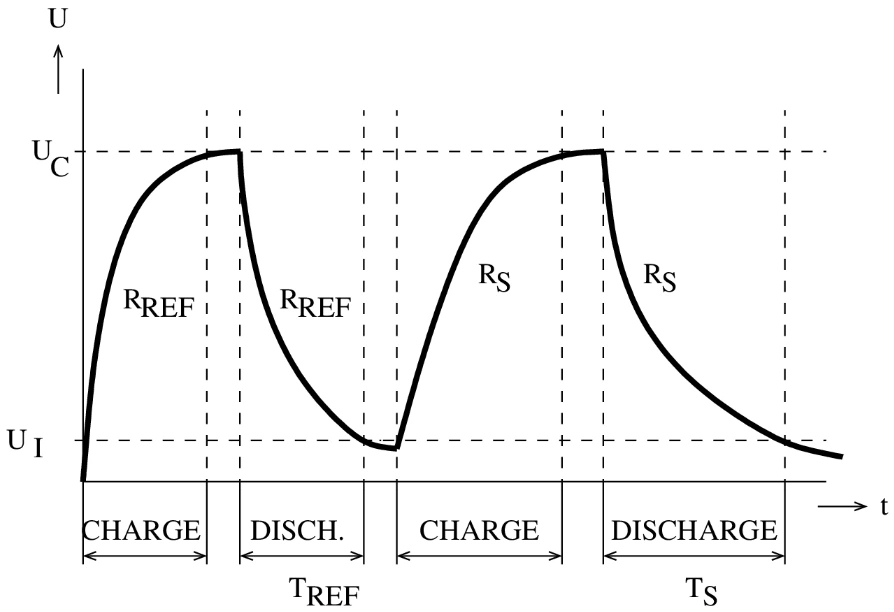
Because the product of time and voltage must be the same for both time periods, we can easily calculate the input voltage:

$$U_{INP} = U_{REF} * \frac{T1}{T2}$$

The input voltage can be very accurately computed from this formula. Therefore, these converters are used for more-bit (16-24) resolution.

A/D Converters with RC Cell





… A/D Converters with RC Cell

An analog quantity is in some applications represented by resistance. In these cases, we can construct an A/D converter with the RC circuit.

It is necessary to have one pin with level voltage detection, e.g. Schmitt trigger circuit. The sensor R_S and the reference resistance R_{REF} are connected to two pins. Both resistances are connected to capacitor with capacity C.

Now the program can connect the pin P1 to the voltage U_C and charge the capacitor to this voltage. Then it is discharging the condenser until its voltage is U_I and it measures the time T_{REF}. The same process is made for pin P2 with the sensor. The result is then the time T_S. We can calculate now the unknown sensor resistance R_S from the following formula:

$$R_S = R_{REF} * \frac{T_S}{T_{REF}}$$

Special Devices

Microcomputers can be integrated many others devices:

- RTC – Real Time Clock circuit for real time. Usually is supplied by a separate battery. It can be synchronized by Long-wave signal.
- Charger controller for acid, NiCd, NiMH or Lion batteries.
- Controller for LCD displays.
- Remote IR controller for TV and Audio.
- On-screen menu for TV.
- Temperature sensors.
- Reference voltage source.
- Comparators.
- PWM generator.