

Basic microcontroller peripheral circuits

Semester 19L – Summer 2019

© Maciej Urbanski, MSc

email: M.Urbanski@elka.pw.edu.pl

Important remark

This material is intended to be used by the students during the Microprocessor Systems course for educational purposes only. The course is conducted in the Faculty of Electronics, Warsaw University of Technology.

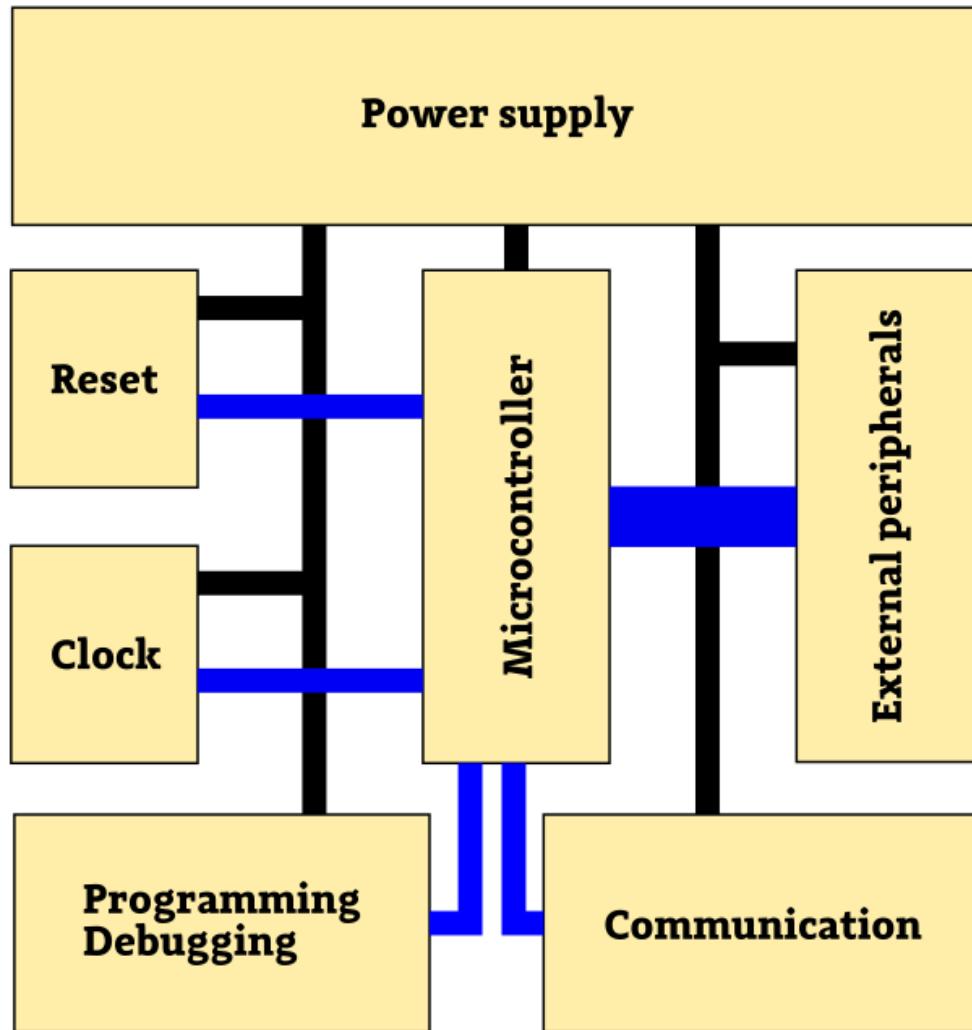
The use of this material in any other purpose than education is prohibited.

This material has been prepared based on many sources, considered by the author as valueable, however it is possible that the material contains errors and misstatements.

The author takes no responsibility for the usage of this material and any potential losses this usage can lead to. Furthermore the author will be very grateful for pointing out any errors found and also for any other useful remarks on the course material and potential upgrades.

Basic structure of typical microprocessor system

This is in fact the basic concept of most systems with microcontrollers.



Power supply – it delivers power to all the system devices. It is designed to be the source of all voltages needed in the circuit. Maximum drawn current has to be estimated.

Reset – this block allows to bring the system into preset state. It brings all the microcontroller's registers to default state.

Clock – microcontroller needs the clock signal for synchronization of data signals between internal and external blocks/devices

Programming & debugging – external interface used to write firmware in the program memory (internal or external) and also to verify the code execution. This will be covered during Lecture 12.

Communication (*) – all the interfaces used to connect the microprocessor system with the real world, for instance with PC

External peripherals (*) – one of the most essential parts of the system. This will be covered during next few lectures...

Not to forget about the microcontroller ☺

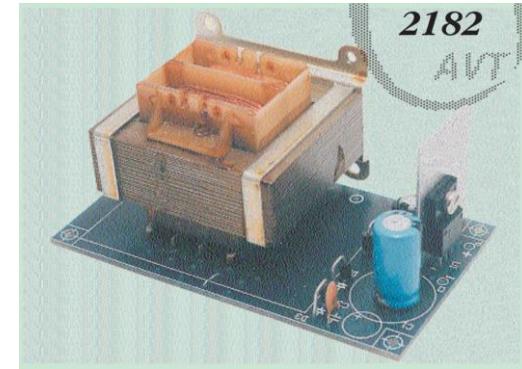
(*) optional

Power supply

It is believed that the electronic circuit works better when plugged into power supply...

Power supply is a device that converts the external power source parameters to fit the designed circuit requirements. These requirements are:

- Number of outputs (separated or not)
- Voltage levels for the outputs
- Current limits for the outputs



Simple power supply using a transformer

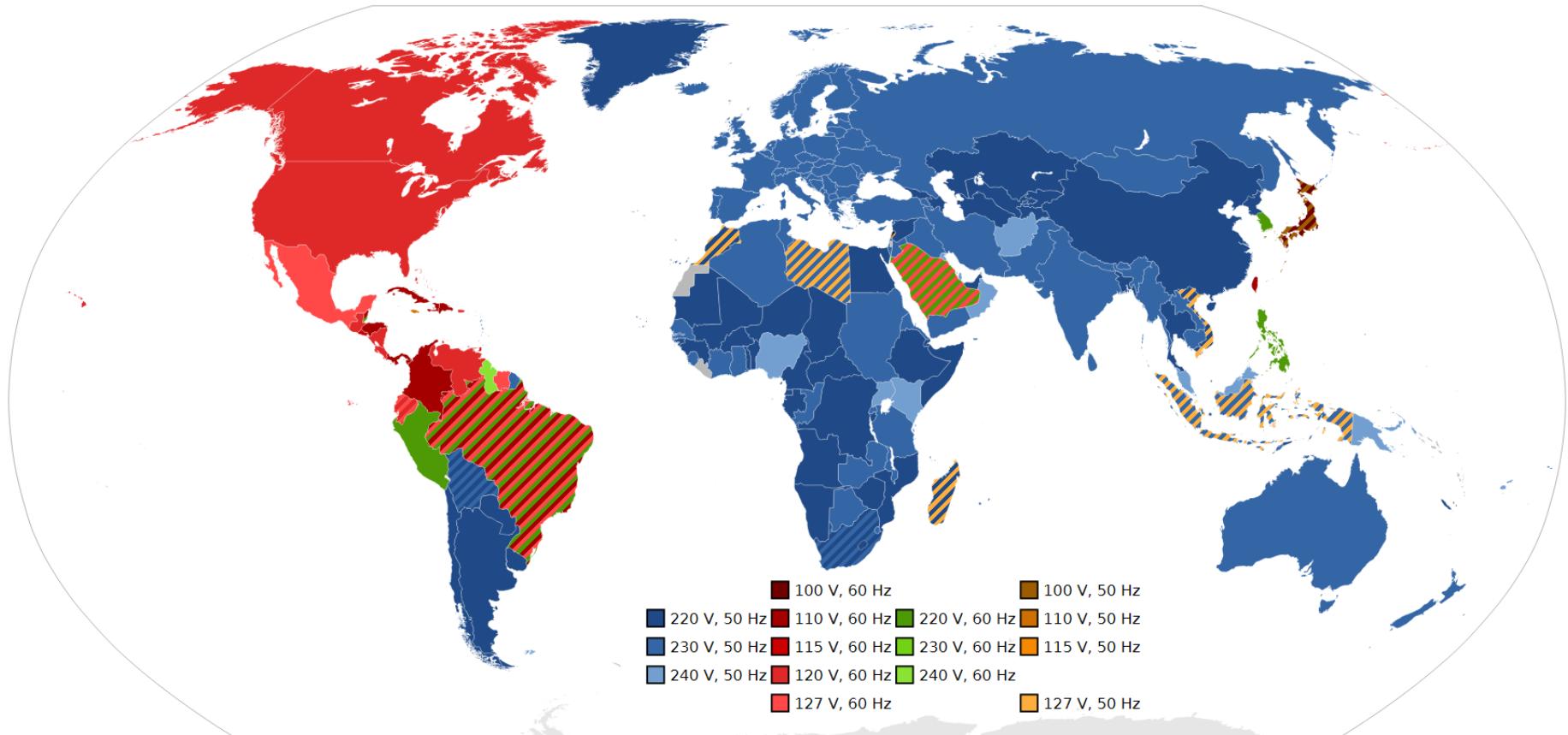


Commercial switching power supply

There are basically two types of power supplies:

- Linear power supplies
 - Low noise
 - Low efficiency – high thermal losses
- Switching power supplies
 - High efficiency
 - May be the source of high frequency noise in circuit

Power supply – mains voltage across the globe



The mains voltage is not equal around the world and this has to be considered during the project design stage. Where do you want to sell your product? Is your power supply compatible?

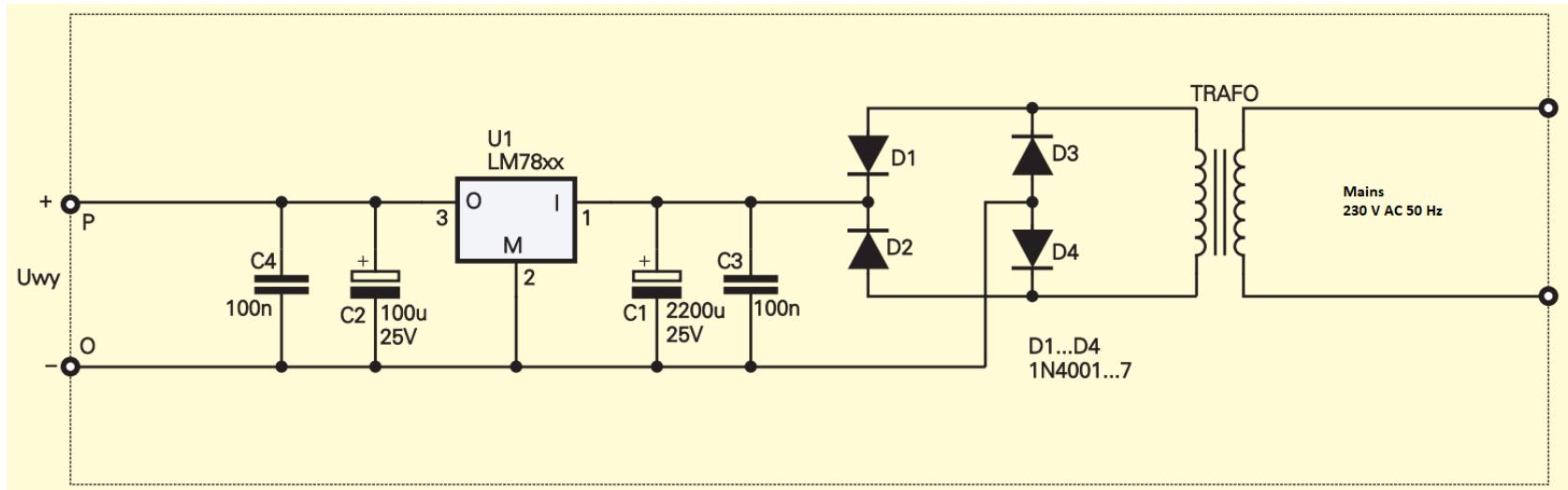
For linear power supplies there is a need for adjusting the transformer.

Switching power supply can be universal – working in range from 80 VAC to 260 VAC.

Do not forget about the certification of your product before you sell it. Certificate requirements are different across the globe!

Power supply

Transformer based power supply



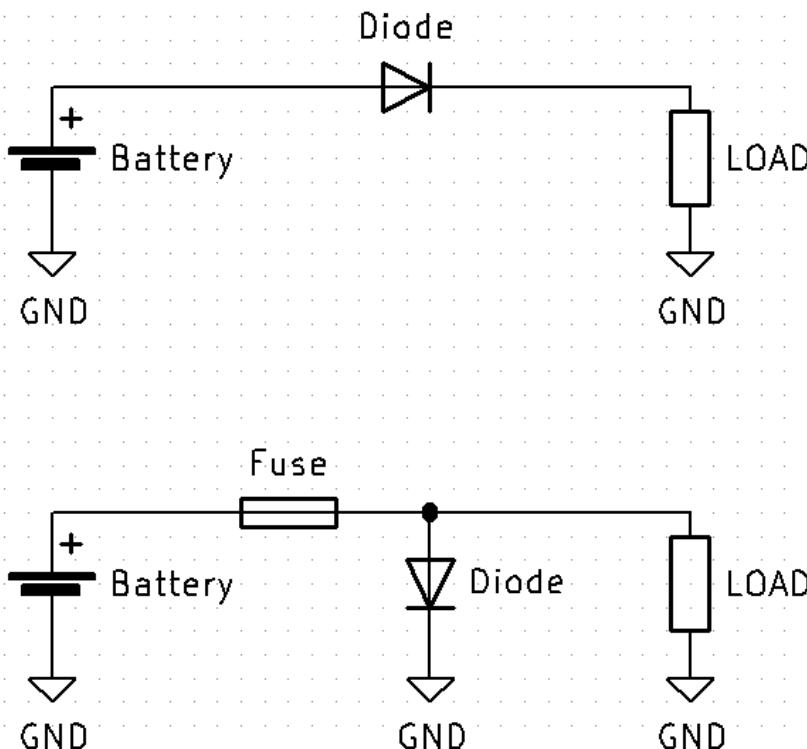
Transformer (TRAFO) converts input 230V AC sinewave 50 Hz into low voltage sinewave 50 Hz. The low voltage sinewave signal is then rectified with the full bridge rectifier (D1-D4). Capacitors C1 and C3 filters rectifier ripple voltage. U1 is linear voltage regulator. It converts DC rectifier voltage to required DC voltage (for example +5V) Capacitors C2 and C4 filters output DC voltage.

Battery supply

There are many ways to connect battery to the circuit.

Good advice. If something stupid is possible to be done in your circuit then you can be 100% sure that there is at least one user of your circuit that will do it!

Always make your circuit "user-proof".

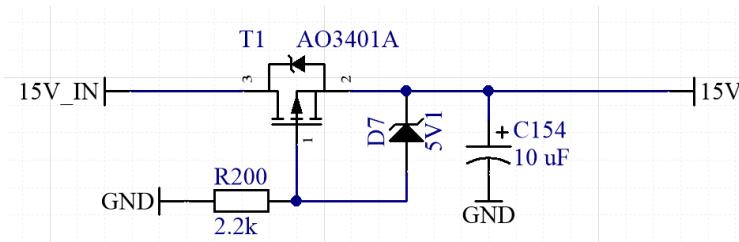


Diode is used to prevent LOAD (your circuit) from reverse polarity connection of the battery.

The drawback is that the voltage across the load is one diode voltage drop less than battery voltage.

Destructive protection. When the battery is connected the wrong way it causes the very high current to flow through the fuse and the diode. This very high current will blow the fuse and thus prevent the load, according to Kirchoff's laws.

There are more clever ways of protecting the device from reverse battery (or power supply) connection.



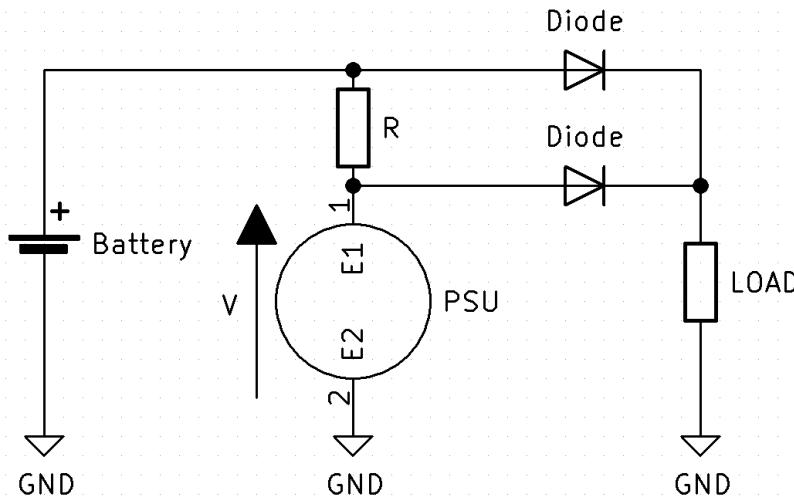
Emergency battery power supply connection

Sometimes there is a need to make your system invulnerable to power supply loss, like blackouts, electricity failures, etc.

This often happens in clock circuits. It would be nice to have a clock that will hold the time information even with the power supply turned off.

The device should detect that there is a problem with power supply and switch to power saving mode (Lecture 11). The battery should then supply power only to the Real Time Clock module (RTC) or selected blocks of microcontroller.

According to laws of physics you MUST NOT connect two voltage sources in parallel!
According to laws of physics you MUST NOT connect two current sources in series!



This is a simple way to connect emergency power supply to the load. The PSU voltage must be higher than battery voltage (and usually it is). When PSU is on, the upper diode is reverse biased and does not allow current to flow. The bottom diode is forward biased. When the PSU is off the bottom diode is reverse biased (or not biased at all)

The diode circuit creates OR gate.
The resistor R is optional and needed when there is a need to charge battery when it is not used.

Typical voltage levels for integrated circuits in microprocessor systems

First digital IC chips were designed to work with +5V DC power supply.

With the rapid development of electronics and digital logic, lower voltage standards were introduced.

The increase of complexity of integrated digital circuits forced the designers to reduce size of transistors in semiconductor designs. This caused technological layers, like oxidation layers for insulating transistors, to become thinner and thinner, thus more vulnerable to electrostatic field strength. The field strength is proportional to the supply voltage. This is why making complex digital chips forces to use lower voltage levels.

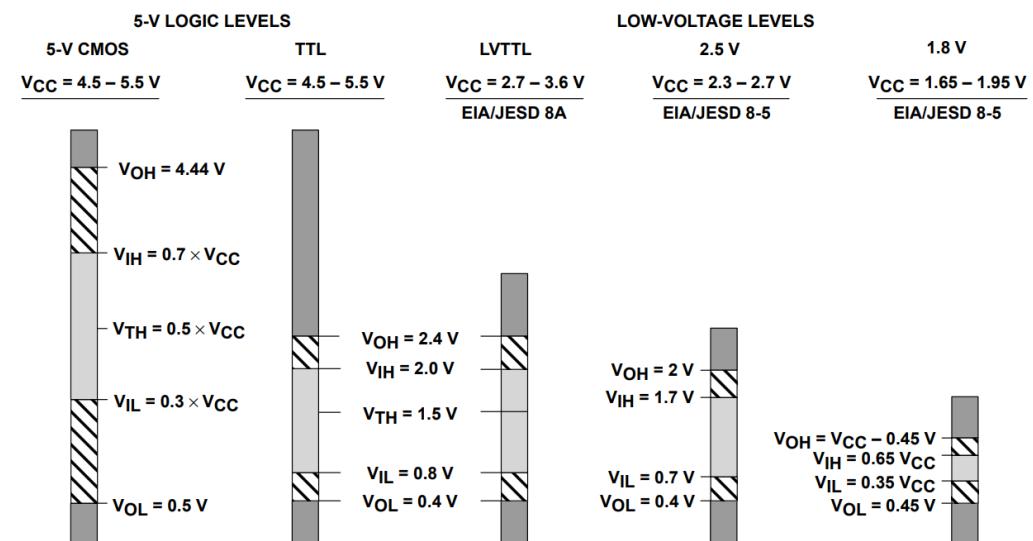
More can be read in:

<http://www.tij.co.jp/jp/lit/an/sdaa011a/sdaa011a.pdf>

3.3 V standard is used in various modern microcontrollers (STM32, AVR)

2.5 V and 1.8 V standards are used mostly in high speed digital circuits, featuring FPGAs, fast digital differential buses and fast ADCs

5V standard is still used in older digital circuits, but also in applications exposed to strong electromagnetic interference (EMI), for instance in industry applications.



Linear voltage regulators – fixed and adjustable

Voltage regulator is a device that converts input DC voltage to output DC voltage.

Voltage regulator has to:

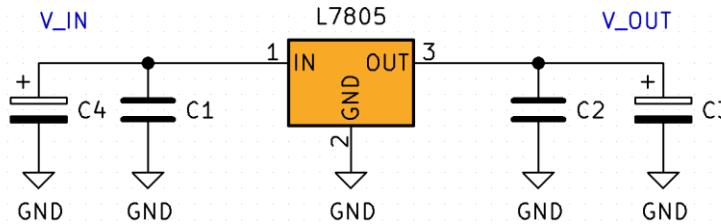
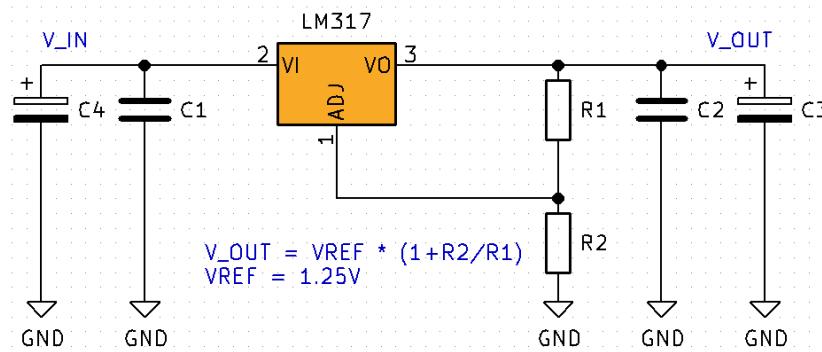
- Ensure proper output voltage stability in required voltage and current range
- Ensure proper output current limit
- Be invulnerable to output short
- Have the highest possible efficiency – in other words it should not generate too much heat
- Separate output from input noise and ripple

There are many different voltage regulator available. Usually they have fixed or adjustable output voltage.

In typical applications the most popular voltage regulators are classic LM78XX, LM79XX, LM317, LM337. These are fixed positive (78), fixed negative (79), adjustable positive (317) and adjustable negative (337). They are available in standard output voltages: +5V, +3,3V, and adjustable in range from 1,3 to 30 V.

They are old and robust devices, however not recommended for modern demanding applications.

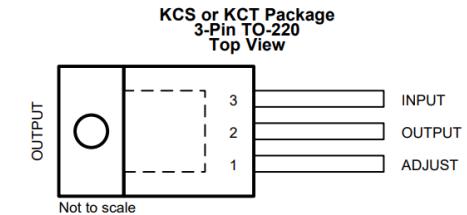
Linear voltage regulators – fixed and adjustable



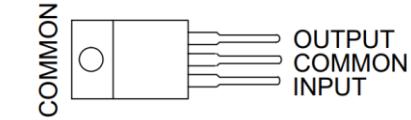
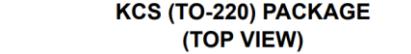
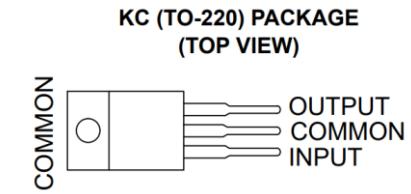
Capacitors and resistors values should be selected according to the datasheets and application – capacitors should provide required output ripple level, depending from output current.

Limitations:

- Input voltage – not higher than +35V DC
- Output current – not higher than 1.5 A (may differ between manufacturers and packages)
- **Maximum power dissipated –**
 $P_{TOT} = (V_{IN} - V_{OUT}) \cdot I_{OUT}$
- The voltage drop across the regulator will be the main source of heat



LM317 pinout (THT TO-220)



LM78XX pinout (THT TO-220)

Switching voltage regulators – solution for all the problems?

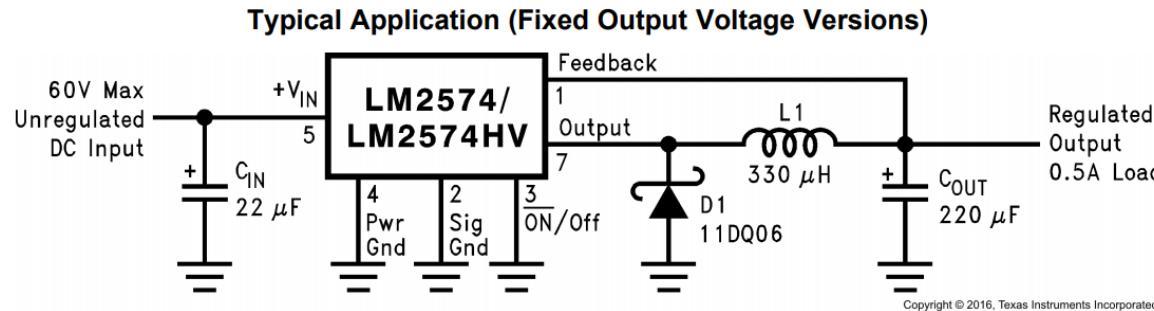
In switching voltage regulators the active element (transistor) is not driven all the time – the driving signal is switched. By proper setting of switching time (time on and off) the amount of charge transferred to the output is adjusted – voltage regulation.

Switching voltage regulators are significantly more efficient than linear regulators (efficiency up to 90%) – they usually do not require big heatsinks (if at all).

Switching mode regulators are based on pulse-width modulation (PWM) to control the average value of the output voltage.

Their application circuit is a bit more complicated than the linear regulator application.

They can be used to generate lower, higher or inverted output voltage, compared to input voltage.



Typical application of LM2574 fixed switching voltage regulator.

When to use switching voltage regulator and when linear one?

Linear regulators are better for low noise and precision applications, for instance for analog signal conditioning, operational amplifiers circuitry, audio preamplifiers, high resolution ADCs, etc.

Linear regulators are best when it is required to quickly react on input voltage change. It is achieved due to analog negative feedback loop.

Linear regulators are simpler and usually cheaper in low power applications.

Switching regulators are best when high efficiency is required – battery powered devices, low power devices.

Switching regulators are usually significantly smaller than linear regulators (with same max. output current).

In high power applications switching regulators are cheaper than linear regulators.

Switching regulators are usually not suitable for low noise applications.

Sample power supply scheme for high precision microprocessor system

This is sample system supplied from +15V, -15V and +7V DC (coming from custom switching power supply).

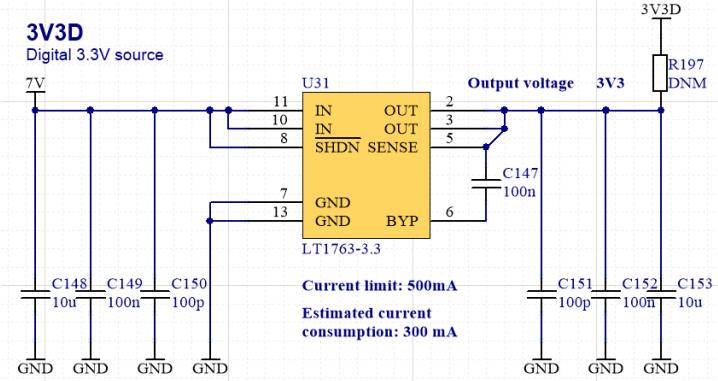
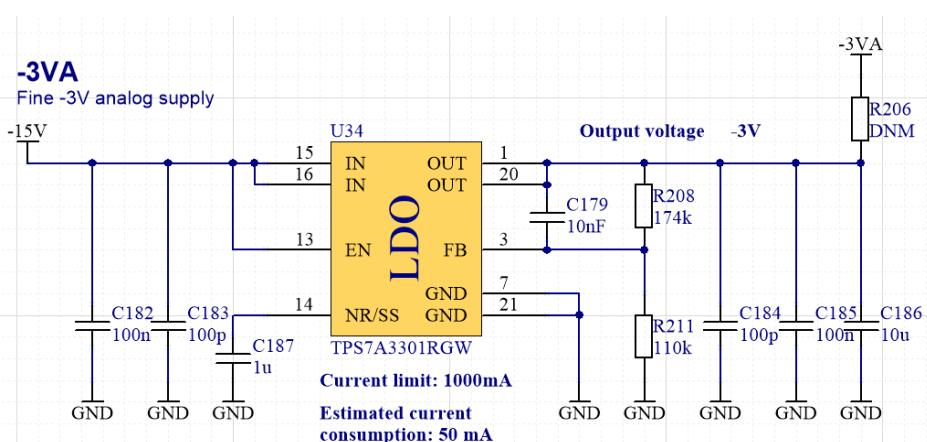
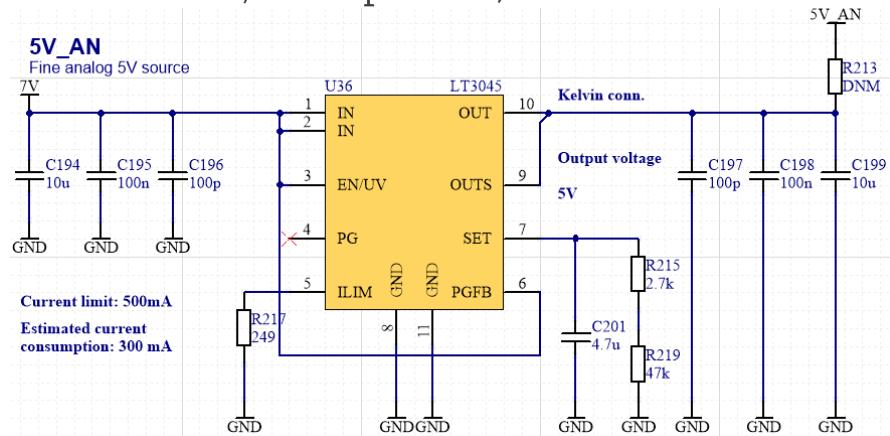
This power supply circuit has to provide +5V for analog circuits, +14V and -3V for operational amplifiers, +3.3V for digital circuits.

Low noise analog regulator is LT3045 (one of the best in the market, but expensive).

The negative regulator is TPS7A3301, also low noise.

For digital 3.3V LT1763 in fixed 3.3V version was chosen.

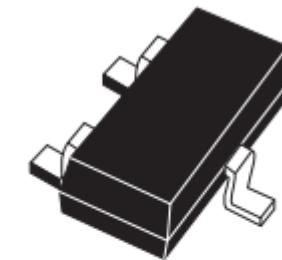
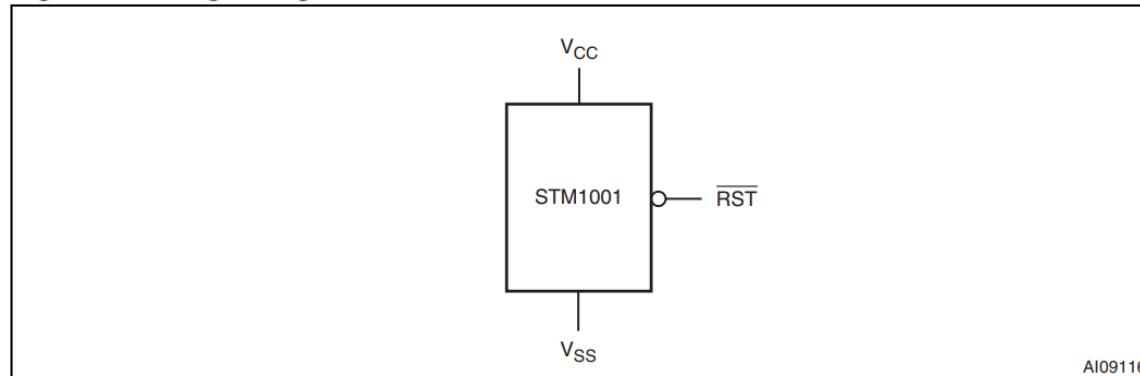
Estimated current consumption made it possible to use only linear regulators.



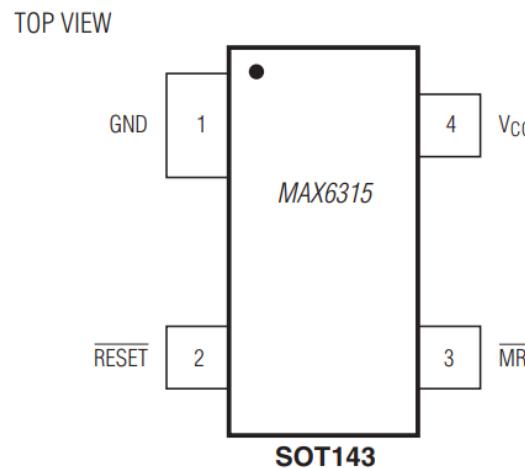
Power supply monitors and reset drivers

The STM1001 microprocessor reset circuit is a low-power supervisory device used to monitor power supplies. It performs a single function: asserting a reset signal whenever the V_{CC} supply voltage drops below a preset value and keeping it asserted until V_{CC} has risen above the preset threshold for a minimum period of time (t_{rec}).

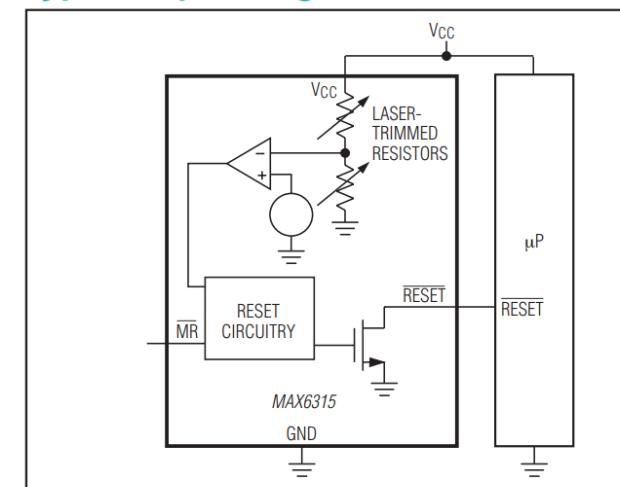
Figure 1. Logic diagram



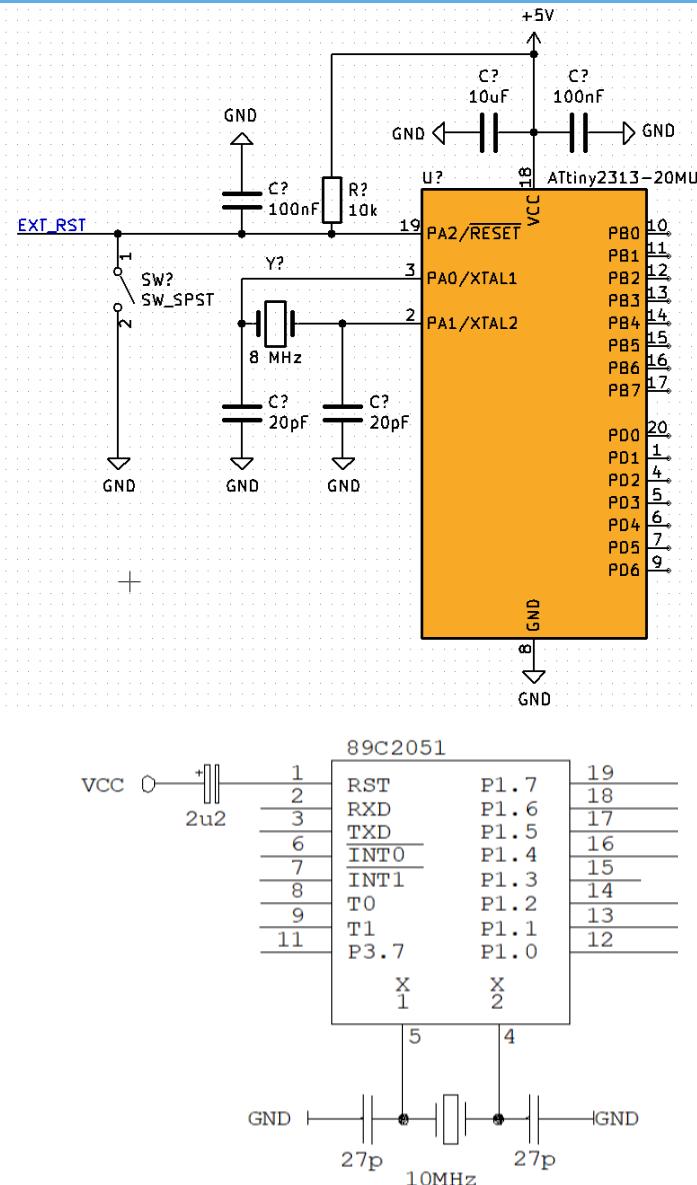
SOT23-3 (WX)



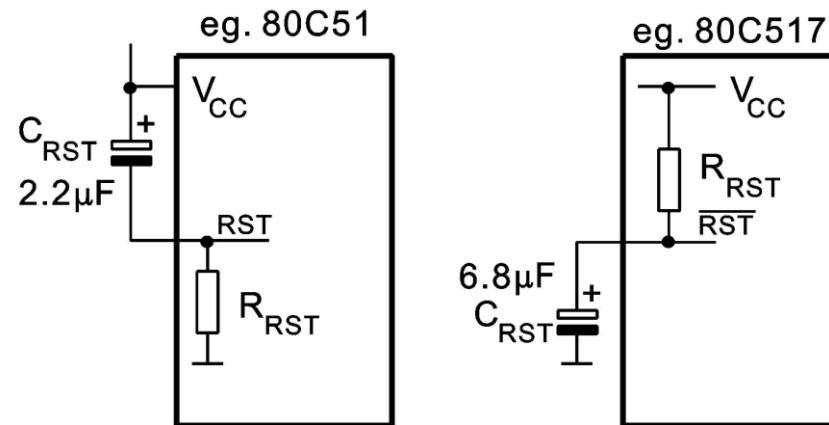
Typical Operating Circuit



Ways of connecting reset circuits to the microcontroller



Typical way of connecting reset circuit
For AVR and STM32 (for STM32 power supply
should be 3.3V)



Reset circuit for different 8051
microcontrollers. Refer to datasheet for
details.

Clock circuits

Every synchronous digital device, like microcontroller, requires synchronization signal for proper work. This signal is called clock signal and is used to manage data flow between microcontroller blocks and external peripherals. In most cases it is square wave or sine wave signal.

Clock signal determines the microcontroller working speed and current consumption.

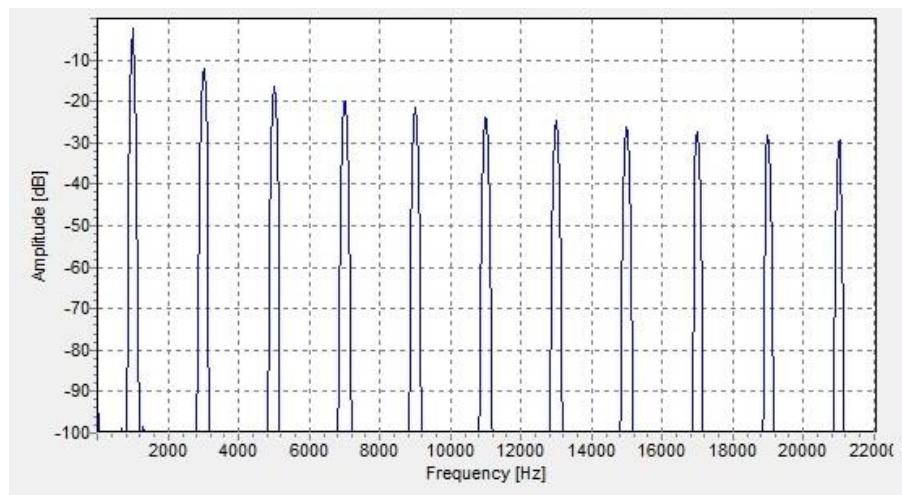
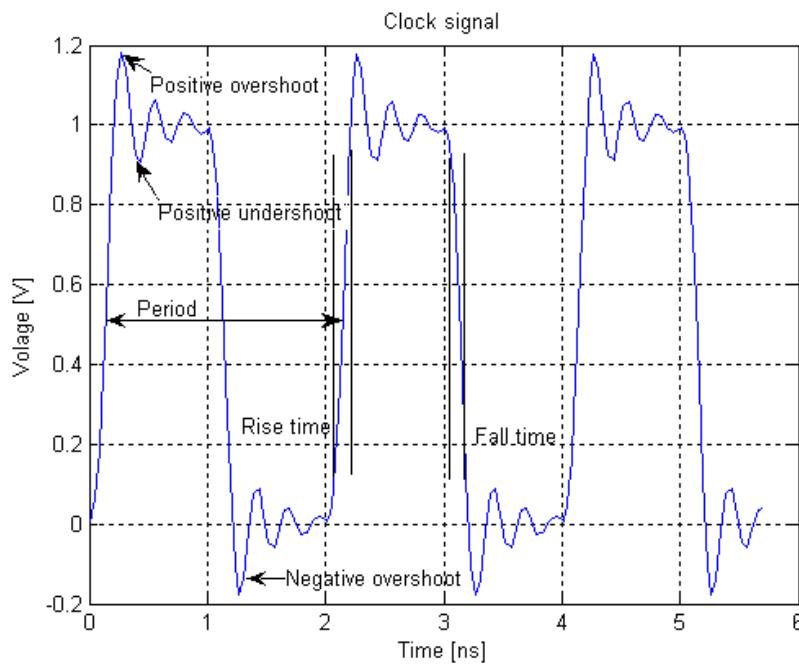
Clock signal frequency usually must be chosen carefully, to fit external peripherals used in the system.

It is often required in the code to generate precision time delays. For the purpose the proper clock frequency can be very useful – especially if the frequency can be divided by 2^n , where n is positive integer value.

Maximum clock frequency is limited by the type of microcontroller. For instance AVR microcontrollers can work with clock signal frequency up to 20 MHz and STM32 with input clock signal frequencies (to be more precise – crystal resonator frequencies) up to 48 MHz. STM32 can use internal PLL to work with clock frequency up to 172 MHz, depending on family (F0 to F7).

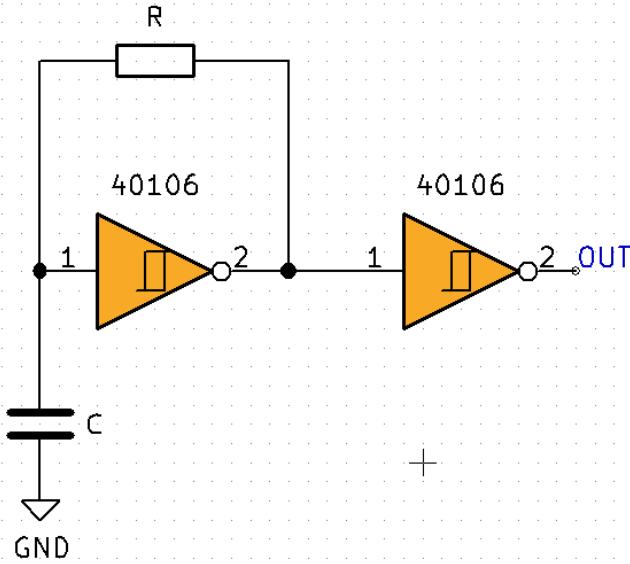
Clock signal – ideal square wave versus real square wave

The world is not perfect. Due to signal integrity issues, mainly impedance mismatch, the square wave is significantly deformed by overshoots and undershoots. These effects are particularly annoying when working with high frequency signals – in such cases careful PCB design and signal integrity simulations have to be done. If this cannot be done, then it should be considered to reduce rise and fall times of the clock signal (usually by a series resistor)

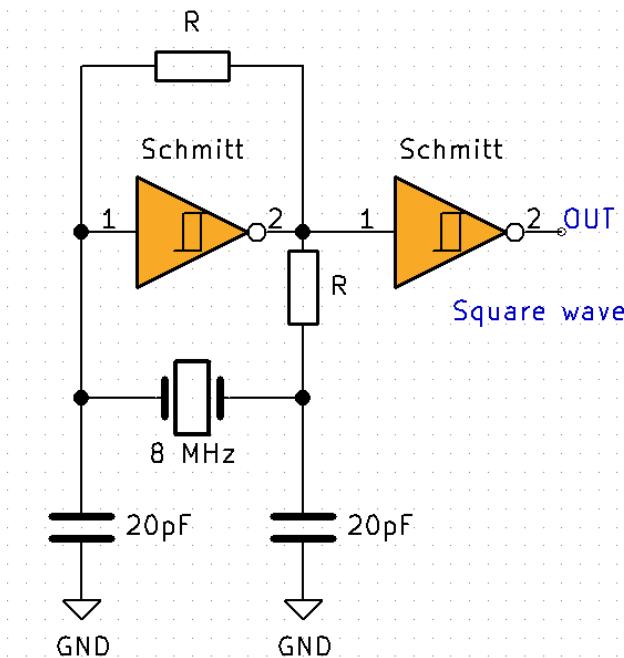


The square wave signal can also be a source of high frequency EMI – electromagnetic interference – it can affect other analog signals in the system (or worse, in some other system), causing malfunctions.

Examples of clock generator circuits



Simple Schmitt inverter RC generator
Similar generator may be embedded in the MCU



Simple Schmitt inverter quartz generator
Similar generator may be used for external crystal oscillator in MCU

Many other generator circuits are possible and not mentioned here
(NE555, CD4060, transistor based circuits, DDS synthesizers, etc.)

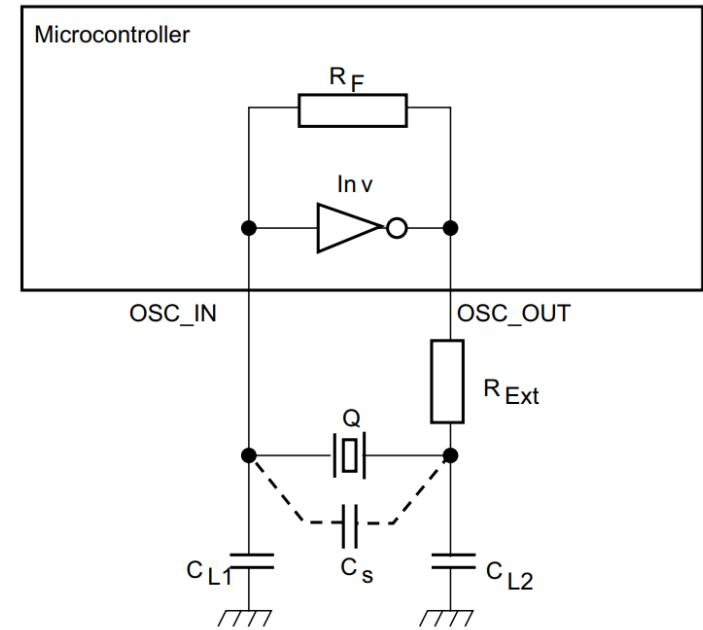
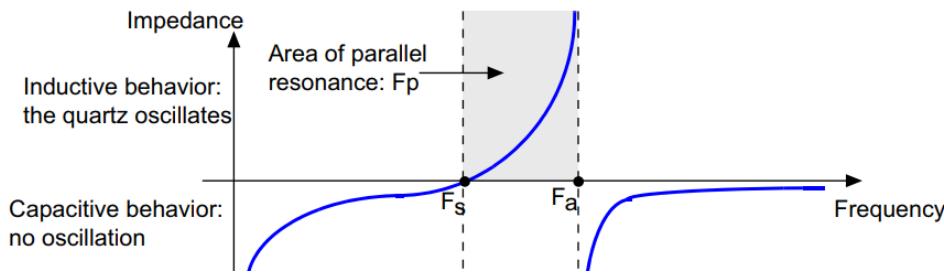
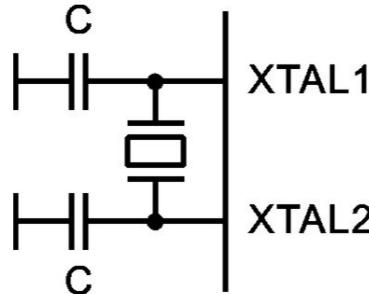
More information:

<https://www.electronics-tutorials.ws/oscillator/crystal.html>

Clock signal oscillators in the microcontrollers – crystal resonator based

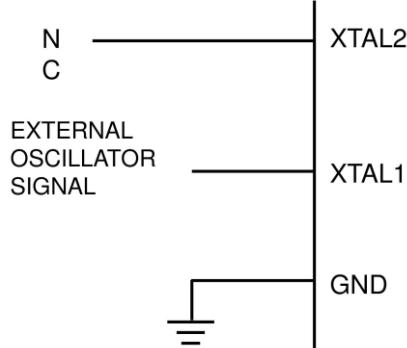
A quartz crystal resonator is a piezoelectric device transforming electric energy into mechanical energy and vice versa. The transformation occurs at the resonant frequency (or its harmonic).

Figure 5. Pierce oscillator circuitry

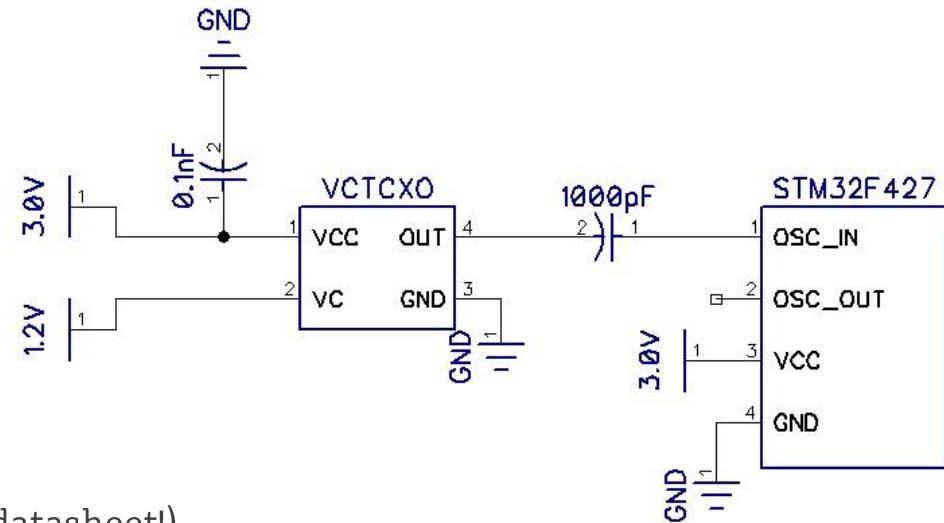


Values of C_{L1} and C_{L2} capacitors should be selected according to required quartz oscillation frequency and microcontroller specification. Refer to ST note AN2867 for details and equations. Usually $C_{L1}=C_{L2}$ = between 10-20 pF (not a rule!!)

Using external clock signal



External clock signal for 8051
(some 8051 chips should have it connected
To XTAL1 – be aware of that and check the datasheet!)

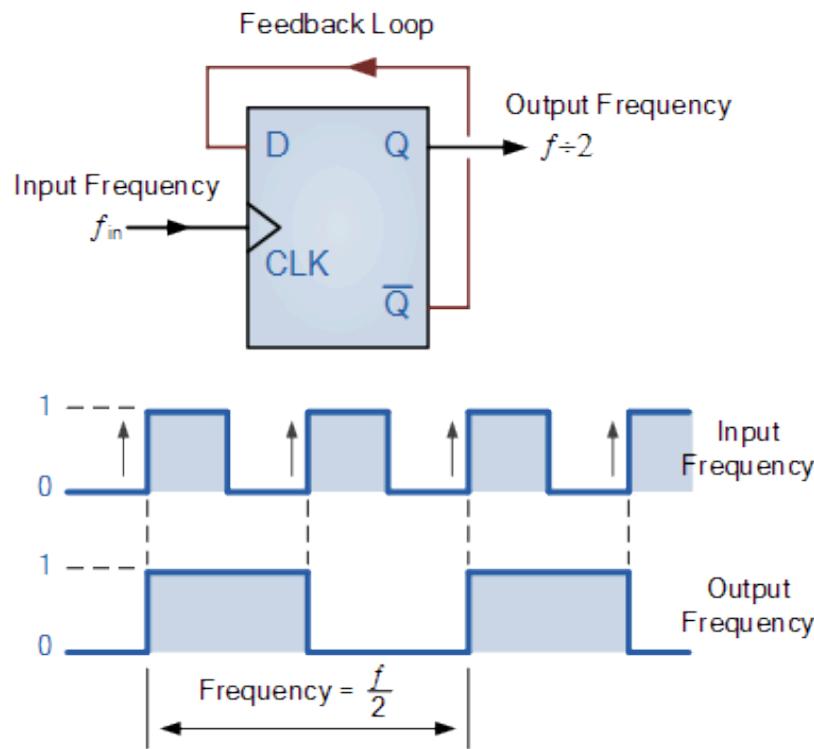


STM32 working with external TCXO

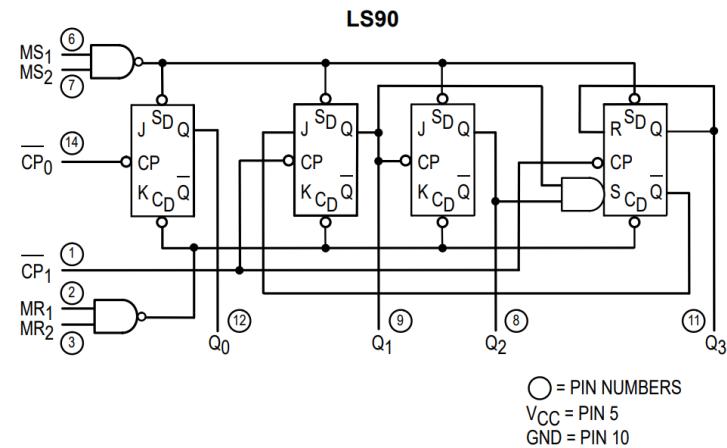


TCXO (Temperature Compensated Crystal Oscillator)

Frequency dividers



Principle of frequency division in D flip flop (divide : 2)



74LS90 4 bit counter, may be used as frequency divider

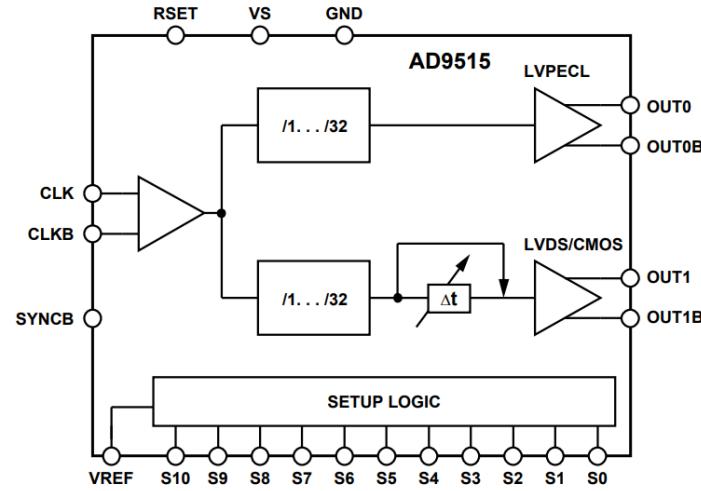


Figure 1.

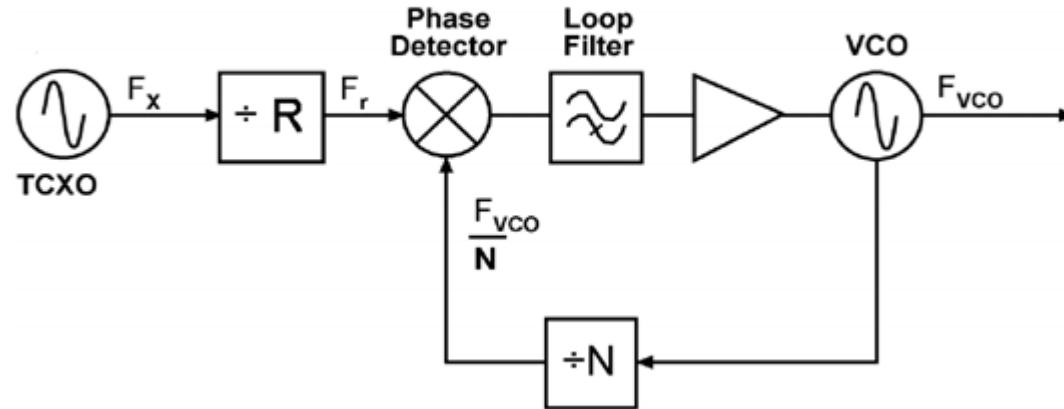
AD9515 clock divider, capable of working with 1.5 GHz input signal

Frequency multiplier - PLL

The PLL circuit performs frequency multiplication via comparing phase difference between reference signal (TCXO) and signal from frequency divider ($:N$). The negative feedback loop will try to drive the VCO (Voltage Controlled Oscillator) in such way to achieve $F_{VCO} = NF_r$. This way the phase detector is driven with two signals of same frequency – this is a stable state, we say that the phase loop is **locked**. As can be seen the circuit allows for integer-n multiplication of input frequency.

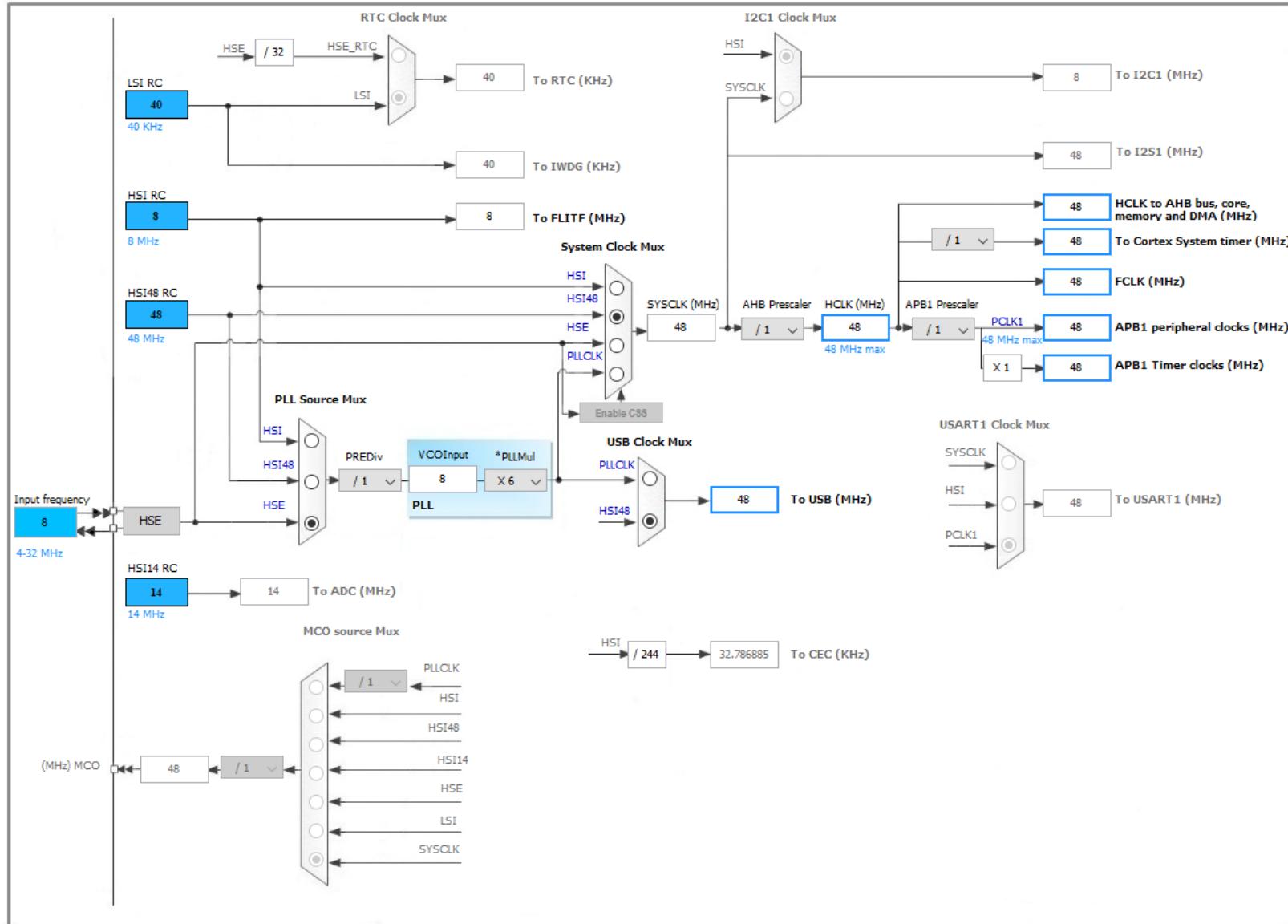
In digital systems the VCO is replaced with DCO (Digital Controlled Oscillator) and the output signal is square wave.

Integer-N (classical) PLL Block Diagram



Such devices are embedded in modern microcontrollers, like STM32 series.

More sophisticated internal oscillator circuit in microcontroller (STM32F042G4U6)



Time dependencies

Checking time dependencies:

- Determine connections between the circuits being checked
- Get appropriate time charts from the datasheets
- Find corresponding time markers on the time charts
- For each required time dependency define formulas describing the same timing on both time charts
- From each pair of such formulas calculate value of the parameter being analyzed
- From all the calculated values select the worst case

Program memory timing example: determination of connections

$O\ 0..7 = \text{PORT}\ 0$

$\overline{OE} = \overline{\text{PSEN}}$

$\overline{CE} = "0"$

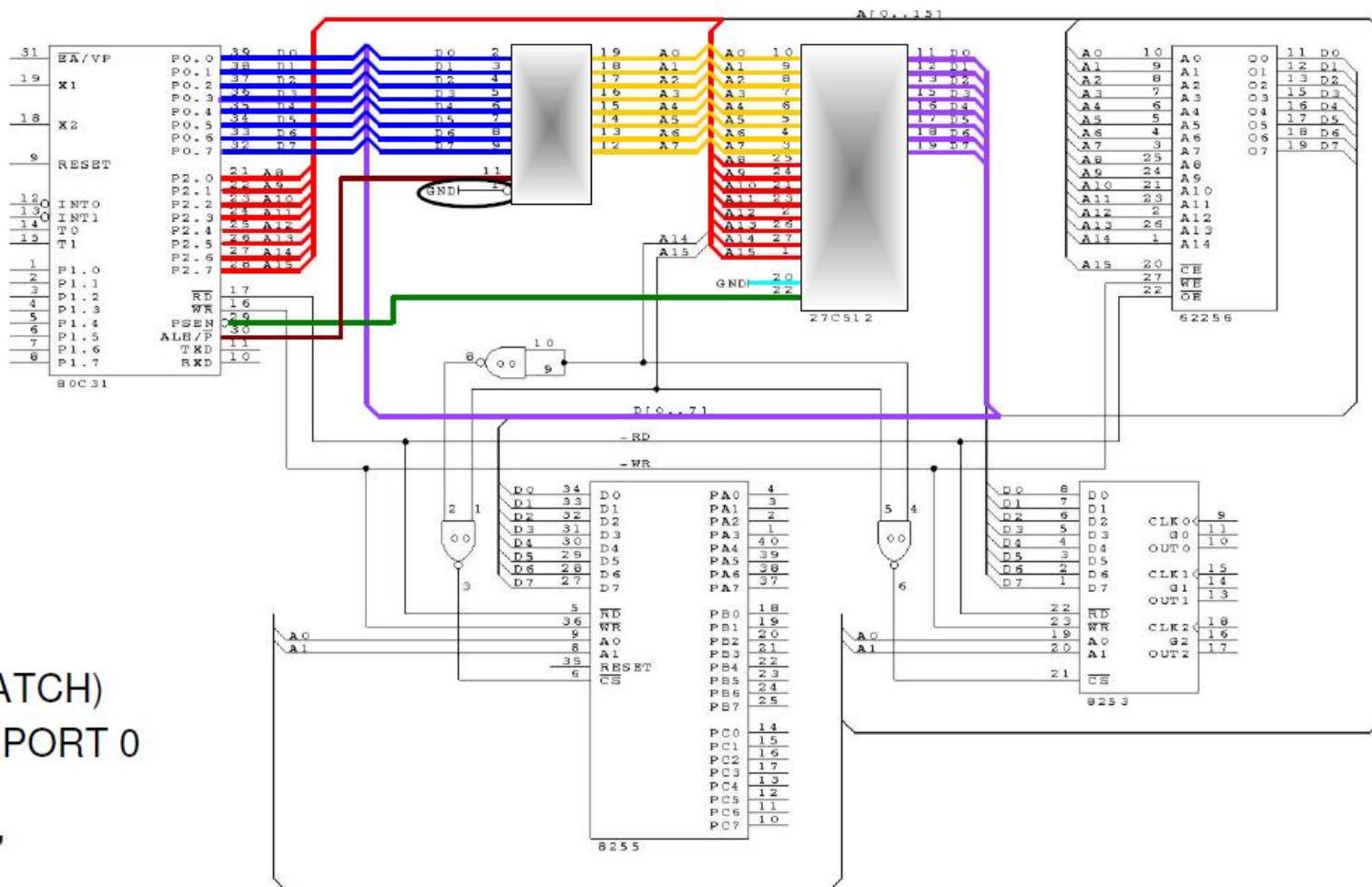
$A\ 8..15 = \text{PORT}\ 2$

$A\ 0..7 = Q\ 0..7 \text{ (LATCH)}$

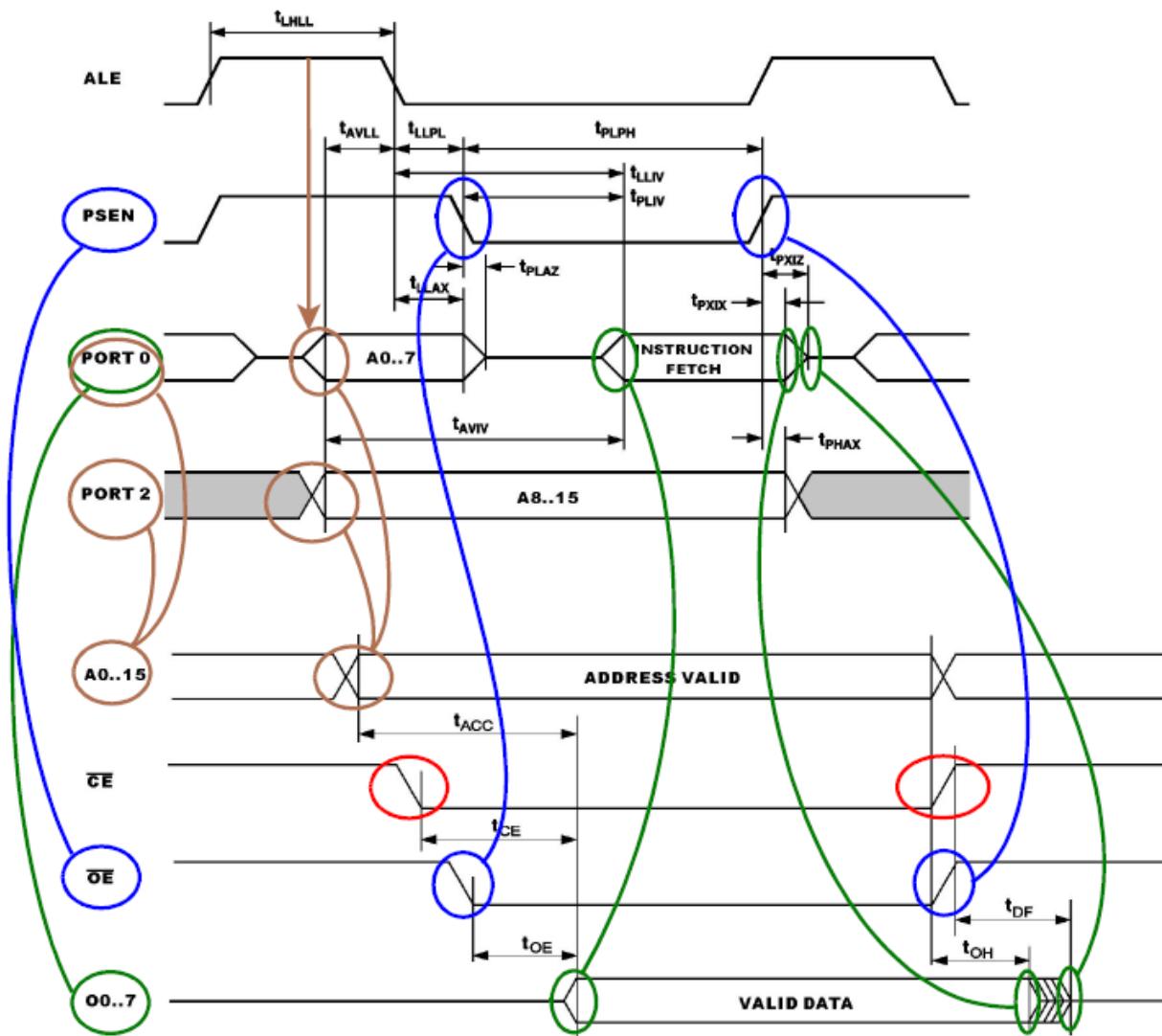
$D\ 0..7 \text{ (LATCH)} = \text{PORT}\ 0$

$C \text{ (LATCH)} = \text{ALE}$

$OC \text{ (LATCH)} = "0"$



Program memory timing example: finding corresponding time markers



$t_{ACC} = t_{AVIV}$
 $t_{OE} = t_{PLIV}$
 $t_{OH} = t_{PXIX}$
 $t_{DF} = t_{PXIZ}$

© Tomasz Starecki

Program memory timing example – using microprocessor timing specs

	Parameter	Variable Clock Min	Variable Clock Max	Unit
t_{LHLL}	ALE Pulsewidth	$2t_{CK} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	$t_{CK} - 40$		ns
t_{LLAX}	Address Hold after ALE Low	$t_{CK} - 30$		ns
t_{LLIV}	ALE Low to Valid Instruction In		$4t_{CK} - 100$	ns
t_{LLPL}	ALE Low to <u>PSEN</u> Low	$t_{CK} - 30$		ns
t_{PLPH}	<u>PSEN</u> Pulsewidth	$3t_{CK} - 45$		ns
t_{PLIV}	PSEN Low to Valid Instruction In		$3t_{CK} - 105$	ns
t_{PXIX}	Input Instruction Hold after <u>PSEN</u>	0		ns
t_{PXIZ}	Input Instruction Float after <u>PSEN</u>		$t_{CK} - 25$	ns
t_{AVIV}	Address to Valid Instruction In		$5t_{CK} - 105$	ns
t_{PLAZ}	PSEN Low to Address Float		25	ns
t_{PHAX}	Address Hold after <u>PSEN</u> High	0		ns

$$t_{ACC} = t_{AVIV} = 5 t_{CK} - 105 \text{ [ns]}$$

$$t_{OE} = t_{PLIV} = 3 t_{CK} - 105 \text{ [ns]}$$

$$t_{OH} = t_{PXIX} = 0$$

$$t_{DF} = t_{PXIZ} = t_{CK} - 25 \text{ [ns]}$$

Program memory timing example – finding appropriate memory

For $f_{\text{osc}} = 10 \text{ MHz}$, $t_{\text{CK}} = 1/f_{\text{osc}} = 100 \text{ ns}$
 For $f_{\text{osc}} = 20 \text{ MHz}$, $t_{\text{CK}} = 1/f_{\text{osc}} = 50 \text{ ns}$

$$\begin{aligned} t_{\text{ACC}} &= t_{\text{AVIV}} = 5 t_{\text{CK}} - 105 \text{ [ns]} & = 500 - 105 = 395 \text{ [ns]} \\ t_{\text{OE}} &= t_{\text{PLIV}} = 3 t_{\text{CK}} - 105 \text{ [ns]} & = 300 - 105 = 195 \text{ [ns]} \\ t_{\text{OH}} &= t_{\text{PXIX}} = 0 \\ t_{\text{DF}} &= t_{\text{PXIZ}} = t_{\text{CK}} - 25 \text{ [ns]} & = 100 - 25 = 75 \text{ [ns]} \end{aligned} \quad \begin{aligned} &= 250 - 105 = 145 \text{ [ns]} \\ &= 150 - 105 = 45 \text{ [ns]} \\ &= 50 - 25 = 25 \text{ [ns]} \end{aligned}$$

項目	記号	条件	27512-12		27512-15		27512-20		27512-25		単位
			最小	最大	最小	最大	最小	最大	最小	最大	
アドレスアクセスタイム	t_{ACC}	$\text{CE} = \text{OE} = V_{\text{IL}}$	–	120 ✓	–	150 ✗	–	200 ✗	–	250 ✗	ns
CE アクセスタイム	t_{CE}	$\text{OE} = V_{\text{IL}}$	–	120	–	150	–	200	–	250	ns
OE アクセスタイム	t_{OE}	$\text{CE} = V_{\text{IL}}$	–	50 ✗	–	60	–	70	–	100	ns
出力ディスエーブルタイム	t_{DF}	$\text{CE} = V_{\text{IL}}$	0	40 ✗	0	50	0	55	0	60	ns

$$t_{\text{CE}} = t_{\text{PLIV}} = 3 t_{\text{CK}} - 105 \text{ [ns]} = 150 - 105 = 45 \text{ [ns]}$$

項目	記号	条件	27C256H-45		27C256H-55		27C256H-70		単位
			最小	最大	最小	最大	最小	最大	
アドレスアクセスタイム	t_{ACC}	$\text{CE} = \text{OE} = V_{\text{IL}}$	–	45	–	55	–	70 ✓	ns
CE アクセスタイム	t_{CE}	$\text{OE} = V_{\text{IL}}$	–	45 ✓	–	55 ✗	–	70 ✗	ns
OE アクセスタイム	t_{OE}	$\text{CE} = V_{\text{IL}}$	–	25	–	25	–	30 ✓	ns
出力ディスエーブルタイム	t_{DF}	$\text{CE} = V_{\text{IL}}$	0	20	0	20	0	25 ✓	ns

Program memory timing example – finding appropriate memory

For $f_{\text{osc}} = 10 \text{ MHz}$, $t_{\text{CK}} = 1/f_{\text{osc}} = 100 \text{ ns}$
 For $f_{\text{osc}} = 20 \text{ MHz}$, $t_{\text{CK}} = 1/f_{\text{osc}} = 50 \text{ ns}$

$$\begin{aligned} t_{\text{ACC}} &= t_{\text{AVIV}} = 5 t_{\text{CK}} - 105 \text{ [ns]} & = 500 - 105 = 395 \text{ [ns]} \\ t_{\text{OE}} &= t_{\text{PLIV}} = 3 t_{\text{CK}} - 105 \text{ [ns]} & = 300 - 105 = 195 \text{ [ns]} \\ t_{\text{OH}} &= t_{\text{PXIX}} = 0 \\ t_{\text{DF}} &= t_{\text{PXIZ}} = t_{\text{CK}} - 25 \text{ [ns]} & = 100 - 25 = 75 \text{ [ns]} \end{aligned} \quad \begin{aligned} &= 250 - 105 = 145 \text{ [ns]} \\ &= 150 - 105 = 45 \text{ [ns]} \\ &= 50 - 25 = 25 \text{ [ns]} \end{aligned}$$

項目	記号	条件	27512-12		27512-15		27512-20		27512-25		単位
			最小	最大	最小	最大	最小	最大	最小	最大	
アドレスアクセスタイム	t_{ACC}	$\text{CE} = \text{OE} = V_{\text{IL}}$	–	120 ✓	–	150 ✗	–	200 ✗	–	250 ✗	ns
CE アクセスタイム	t_{CE}	$\text{OE} = V_{\text{IL}}$	–	120	–	150	–	200	–	250	ns
OE アクセスタイム	t_{OE}	$\text{CE} = V_{\text{IL}}$	–	50 ✗	–	60	–	70	–	100	ns
出力ディスエーブルタイム	t_{DF}	$\text{CE} = V_{\text{IL}}$	0	40 ✗	0	50	0	55	0	60	ns

$$t_{\text{CE}} = t_{\text{PLIV}} = 3 t_{\text{CK}} - 105 \text{ [ns]} = 150 - 105 = 45 \text{ [ns]}$$

項目	記号	条件	27C256H-45		27C256H-55		27C256H-70		単位
			最小	最大	最小	最大	最小	最大	
アドレスアクセスタイム	t_{ACC}	$\text{CE} = \text{OE} = V_{\text{IL}}$	–	45	–	55	–	70 ✓	ns
CE アクセスタイム	t_{CE}	$\text{OE} = V_{\text{IL}}$	–	45 ✓	–	55 ✗	–	70 ✗	ns
OE アクセスタイム	t_{OE}	$\text{CE} = V_{\text{IL}}$	–	25	–	25	–	30 ✓	ns
出力ディスエーブルタイム	t_{DF}	$\text{CE} = V_{\text{IL}}$	0	20	0	20	0	25 ✓	ns

Program memory timing example – calculating maximum SYS CLK frequency

項目	記号	条件	27512-12		27512-15		27512-20		27512-25		単位
			最小	最大	最小	最大	最小	最大	最小	最大	
アドレスアクセスタイム	t_{ACC}	$CE = OE = V_{IL}$	–	120	–	150	–	200	–	250	ns
CE アクセ스타イム	t_{CE}	$OE = V_{IL}$	–	120	–	150	–	200	–	250	ns
OE アクセ스타イム	t_{OE}	$CE = V_{IL}$	–	50	–	60	–	70	–	100	ns
出カティスエーブルタイム	t_{DF}	$CE = V_{IL}$	0	40	0	50	0	55	0	60	ns

項目	記号	条件	27C256H-45		27C256H-55		27C256H-70		単位
			最小	最大	最小	最大	最小	最大	
アドレスアクセスタイム	t_{ACC}	$CE = OE = V_{IL}$	–	45	–	55	–	70	ns
CE アクセ스타イム	t_{CE}	$OE = V_{IL}$	–	45	–	55	–	70	ns
OE アクセ스타イム	t_{OE}	$CE = V_{IL}$	–	25	–	25	–	30	ns
出カティスエーブルタイム	t_{DF}	$CE = V_{IL}$	0	20	0	20	0	25	ns

$$t_{ACC} = t_{AVIV} = 5 t_{CK} - 105 \text{ [ns]}$$

$$t_{OE} = t_{PLIV} = 3 t_{CK} - 105 \text{ [ns]}$$

$$t_{OH} = t_{PXIX} = 0$$

$$t_{DF} = t_{PXIZ} = t_{CK} - 25 \text{ [ns]}$$

$$t_{CK} = (t_{ACC} + 105) / 5 = (120 + 105) / 5 = 45 \text{ [ns]}$$

$$t_{CK} = (t_{OE} + 105) / 3 = (50 + 105) / 3 = 51.67 \text{ [ns]}$$

$$t_{CK} = t_{DF} + 25 = 40 + 25 = 65 \text{ [ns]}$$

The worst case is 65 ns, which corresponds to 15.38 MHz.

Influence of latch on the timing characteristics

$$t_{ACC} = t_{AVIV}$$

$$t_{ACC} = t_{AVIV} - t_{LATCH}$$



MM74HC573 3-STATE Octal D-Type Latch

General Description

The MM74HC573 high speed octal D-type latches utilize advanced silicon-gate P-well CMOS technology. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the 3-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

Features

- Typical propagation delay: 18 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μ A maximum
- Low quiescent current: 80 μ A maximum (74HC Series)
- Compatible with bus-oriented systems
- Output drive capability: 15 LS-TTL loads

AC Electrical Characteristics

$V_{CC} = 5V$, $T_A = 25^\circ C$, $t_f = t_r = 6 \text{ ns}$

MM74HC573

Symbol	Parameter	Conditions		Typ	Guaranteed Limit	Units
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Data to Q	$C_L = 45 \text{ pF}$		16	20	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay, LE to Q	$C_L = 45 \text{ pF}$		14	22	ns
t_{PH2}, t_{PL2}	Maximum Output Enable Time	$R_L = 1 \text{ k}\Omega$ $C_L = 45 \text{ pF}$		15	27	ns
t_{PH2}, t_{PL2}	Maximum Output Disable Time	$R_L = 1 \text{ k}\Omega$ $C_L = 5 \text{ pF}$		13	23	ns
t_B	Minimum Set Up Time, Data to LE			10	15	ns
t_H	Minimum Hold Time, LE to Data			2	5	ns
t_W	Minimum Pulse Width, LE or Data			10	16	ns

AC Electrical Characteristics

Symbol	Parameter	Conditions	V_{CC}	$T_A = 25^\circ C$	$T_A = -40 \text{ to } 85^\circ C$	$T_A = -55 \text{ to } 125^\circ C$	Units	
				Typ	Guaranteed Limits			
t_{PHL}, t_{PLH}	Maximum Propagation Delay Data to Q	$C_L = 50 \text{ pF}$	2.0V	45	110	138	165	ns
		$C_L = 150 \text{ pF}$	2.0V	58	150	188	225	ns
		$C_L = 50 \text{ pF}$	4.5V	17	22	28	33	ns
		$C_L = 150 \text{ pF}$	4.5V	21	30	38	40	ns
		$C_L = 50 \text{ pF}$	6.0V	15	19	24	29	ns
		$C_L = 150 \text{ pF}$	6.0V	19	26	33	39	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay, LE to Q	$C_L = 50 \text{ pF}$	2.0V	46	115	138	165	ns
		$C_L = 150 \text{ pF}$	2.0V	60	155	194	233	ns
		$C_L = 50 \text{ pF}$	4.5V	14	23	29	35	ns
		$C_L = 150 \text{ pF}$	4.5V	21	31	47	47	ns
		$C_L = 50 \text{ pF}$	6.0V	12	20	25	30	ns
		$C_L = 150 \text{ pF}$	6.0V	19	27	34	41	ns

AC ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS					UNIT		
			$T_{Amb} = +25^\circ C$ $V_{CC} = +5V$ $C_L = 50\text{pF}$, $R_L = 500\Omega$		$T_{Amb} = 0^\circ C \text{ to } +70^\circ C$ $V_{CC} = +5V \pm 10\%$ $C_L = 50\text{pF}$, $R_L = 500\Omega$					
			MIN	Typ	MAX	MIN	MAX			
t_{PHL} t_{PLH}	Propagation delay Dn to Qn	74F573	Waveform NO TAG	3.0 1.0	5.5 3.5	8.0 6.0	2.5 1.0	9.0 7.0	ns	
			Waveform NO TAG	4.5 3.0	8.5 5.0	11.5 7.0	4.0 2.5	12.5 8.0		
	Output Enable time to High or Low level		Waveform NO TAG	2.5 2.5	5.5 5.5	9.5 8.0	2.0 2.0	10.5 8.5	ns	
			Waveform NO TAG	1.0 1.0	3.0 2.5	6.0 5.5	1.0 1.0	6.5 5.5		
	Output Disable time from High or Low level	74F574	Waveform NO TAG	1.0 1.0	3.0 2.5	6.0 5.5	1.0 1.0	6.5 5.5	ns	
			Waveform NO TAG	160	180		150		MHz	
t_{PHL} t_{PLH}	Propagation delay CP to Qn		Waveform NO TAG	3.5 3.5	5.0 5.0	7.5 7.5	3.0 3.0	8.0 8.0	ns	
			Waveform NO TAG	2.5 3.0	4.5 5.0	7.5 8.0	2.0 3.0	7.5 6.5		
	Output Enable time to High or Low level		Waveform NO TAG	1.0 1.0	3.0 2.5	6.0 5.5	1.0 1.0	6.0 6.0	ns	
			Waveform NO TAG	1.0 1.0	3.0 2.5	6.5 5.5	1.0 1.0	6.0 6.0		

External peripherals – General Purpose Input Output ports

One of the most important feature of the microcontroller is the number of available General Purpose Input Output pins.

To make the microcontroller more universal many of its GPIO lines have alternate functions. One pin can act as a GPIO, or USB data line, or SPI clock line, etc.

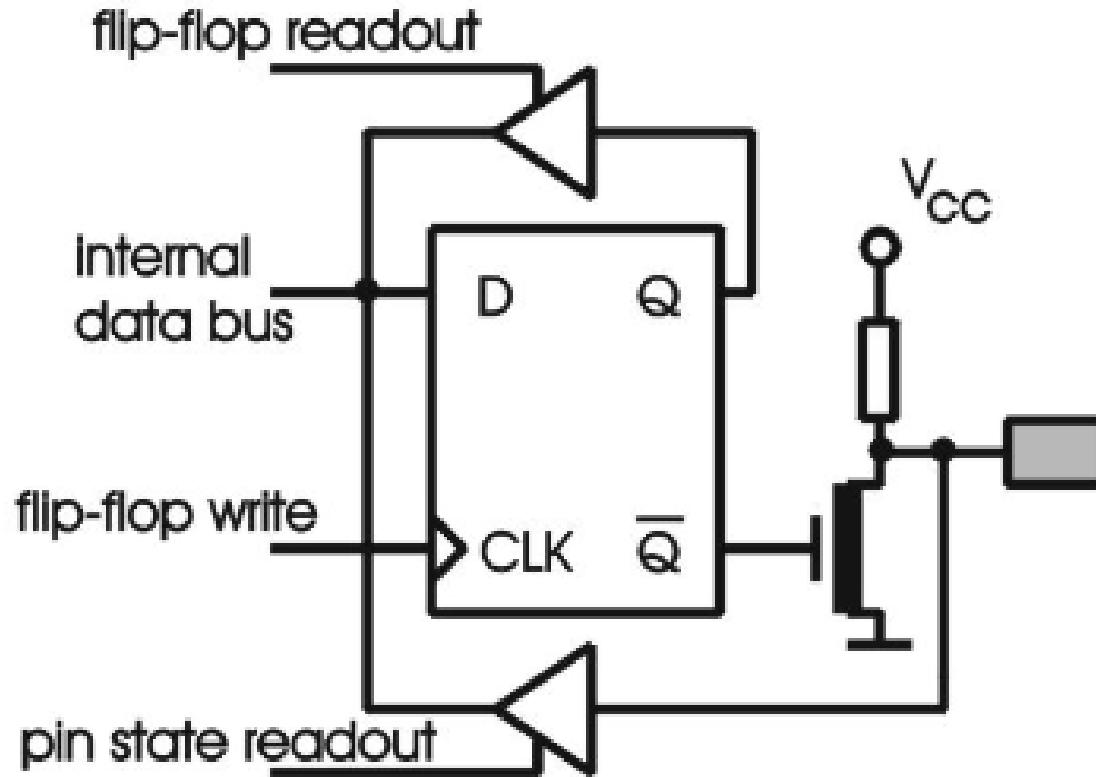
GPIO function type is defined in firmware in most cases.

GPIO pins are grouped in ports. These ports are then configured with SFR registers and can be driven/read using these registers.

Usually GPIO pins are bi-directional – they can be inputs or outputs.

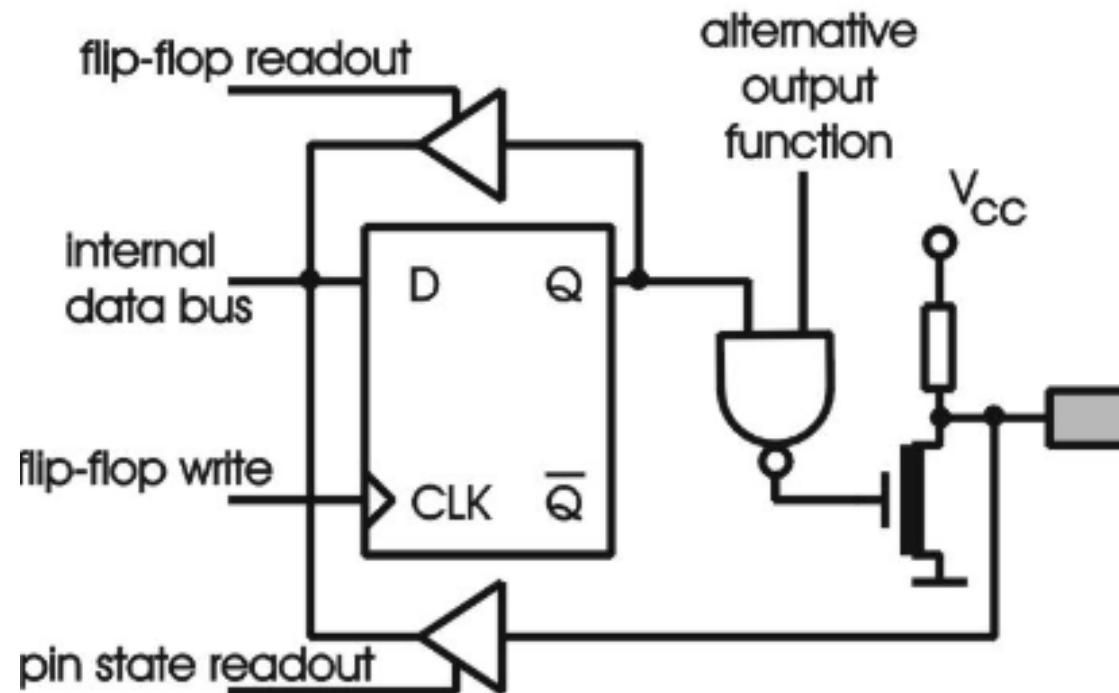
External peripherals – General Purpose Input Output ports in 8051

Simplified diagram of bi-directional GPIO pin



External peripherals – General Purpose Input Output ports in 8051

Simplified schematic of GPIO pin with alternate function for the output



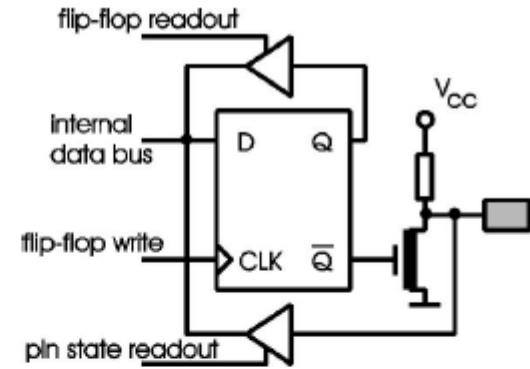
External peripherals – General Purpose Input Output ports in 8051

Read-modify-write instructions

GPIO read operation can be done in two ways:

- By reading port state register
 - By reading input line state

Instructions that reads the state registers are called read-modify-write instructions.



External peripherals – General Purpose Input Output ports in 8051

Read-modify-write instructions

In some cases the unawareness may lead to read errors – instead of reading status register the current state of input lines may be read.

read/write instructions:

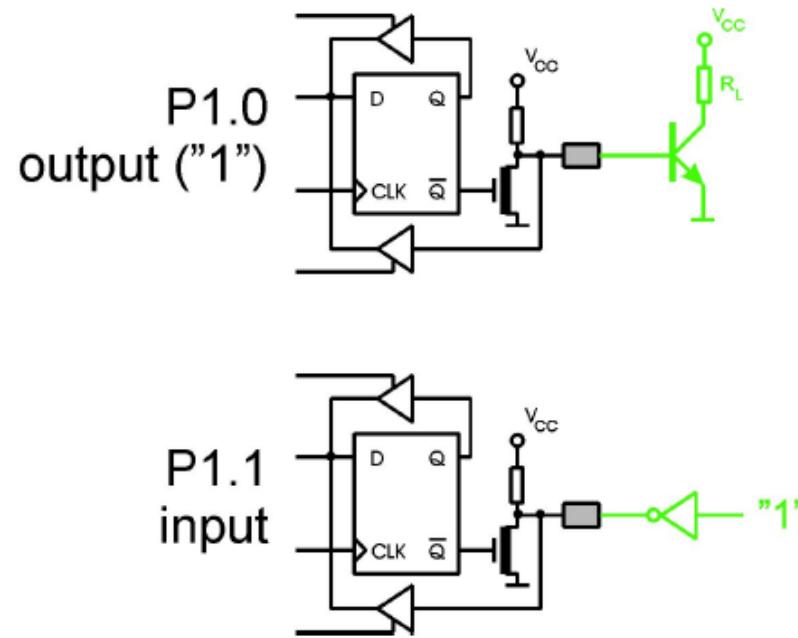
MOV A, P1

ANL A, #0FFH

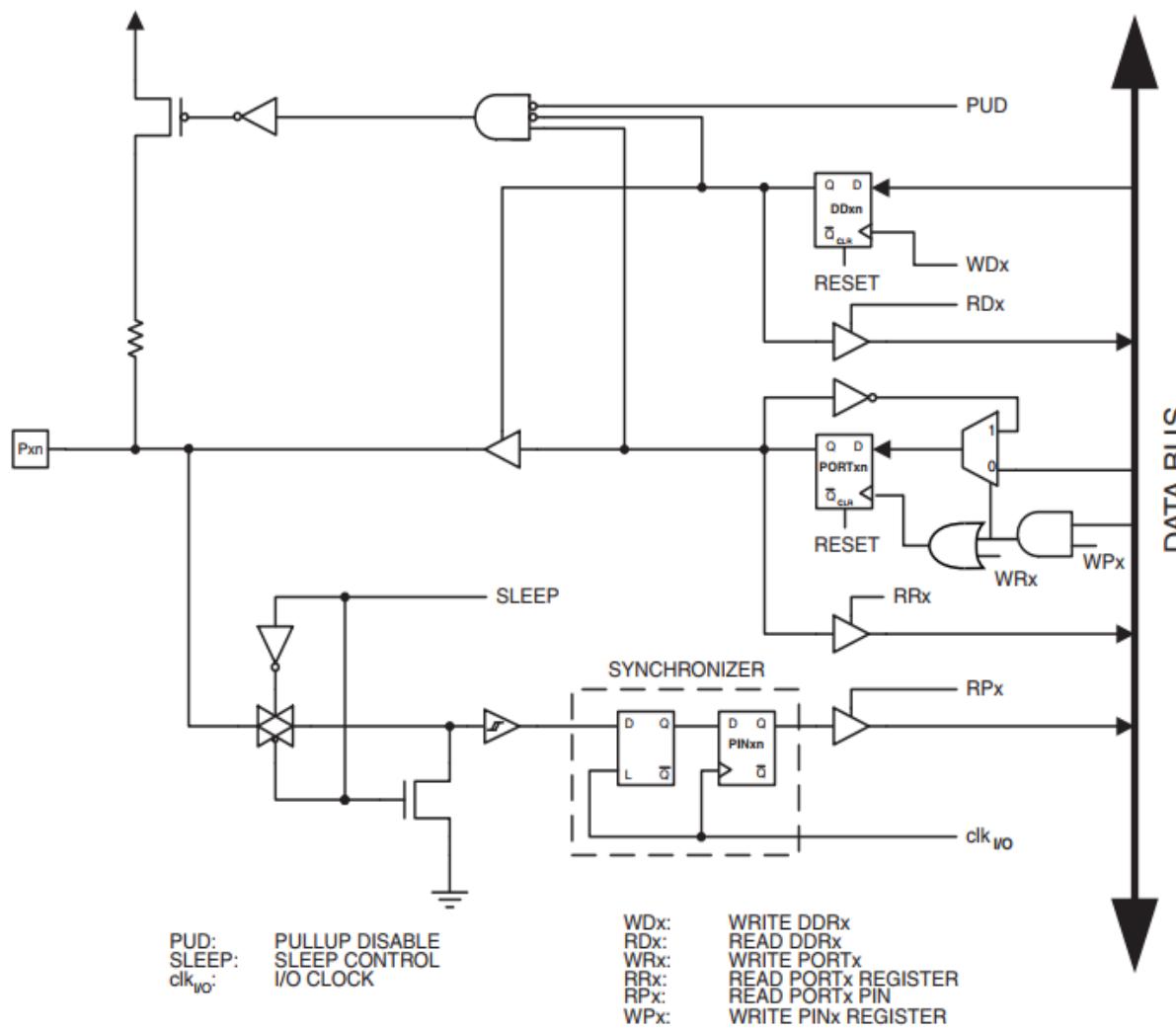
MOV P1, A

read-modify-write instruction:

ANL P1, #0FFH



External peripherals – General Purpose Input Output ports in AVR



GPIO port is driven by three registers:

PORTx – port data register

DDRx – port data direction register

PINx – port input pins address register

Ports are bi-directional, with optional internal pull-up resistors.

If DDRxn is written logic one, then Port X pin n is output pin, otherwise it is input pin.

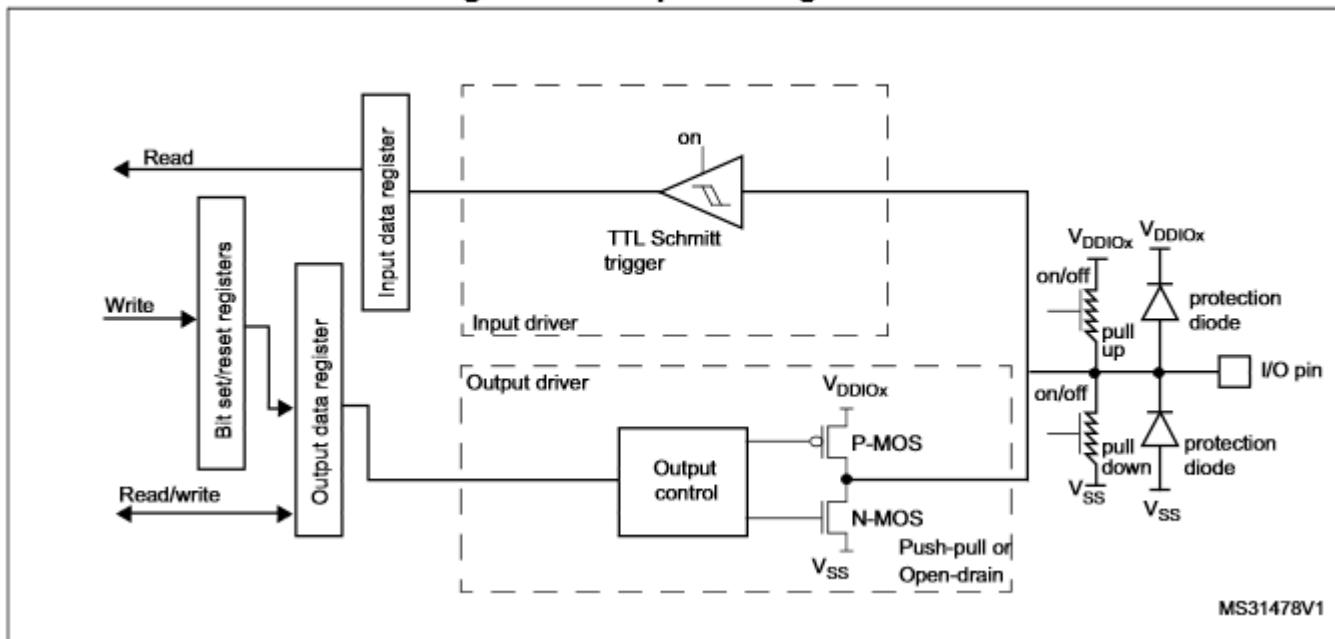
If PORTxn is written logic one and DDRxn is logic zero, the pull-up is activated.

If PORTxn is written logic one and DDRxn is one, then the Port X pin n output state is logic one, otherwise zero.

Writing logic one to PINxn toggles the value of PORTxn, independent on DDRxn.

External peripherals – General Purpose Input Output ports in STM32

Figure 25. Output configuration



In STM32 to make the GPIO port operational the designer has to:

- Turn on clock for the port (RCC_IOPENR)
- Set the port mode (GPIOx_MODER)
- Set the output type (GPIOx_OTYPER)
- Set speed (GPIOx_OSPEEDR)
- Set internal pull up/down resistors (GPIOx_PUPDR)

External peripherals – General Purpose Input Output ports

Alternate functions

Table 14. Alternate functions selected through GPIOA_AFR registers for port A

Pin name	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	-	USART2_CTS	TIM2_CH1_ETR	TSC_G1_IO1	-	-	-	-
PA1	EVENTOUT	USART2_RTS	TIM2_CH2	TSC_G1_IO2	-	-	-	-
PA2	-	USART2_TX	TIM2_CH3	TSC_G1_IO3	-	-	-	-
PA3	-	USART2_RX	TIM2_CH4	TSC_G1_IO4	-	-	-	-
PA4	SPI1 NSS, I2S1 WS	USART2_CK	USB_NOE	TSC_G2_IO1	TIM14_CH1	-	-	-
PA5	SPI1_SCK, I2S1 CK	CEC	TIM2_CH1_ETR	TSC_G2_IO2	-	-	-	-
PA6	SPI1_MISO, I2S1_MCK	TIM3_CH1	TIM1_BKIN	TSC_G2_IO3	-	TIM16_CH1	EVENTOUT	-
PA7	SPI1_MOSI, I2S1_SD	TIM3_CH2	TIM1_CH1N	TSC_G2_IO4	TIM14_CH1	TIM17_CH1	EVENTOUT	-
PA8	MCO	USART1_CK	TIM1_CH1	EVENTOUT	CRS_SYNC	-	-	-
PA9	-	USART1_TX	TIM1_CH2	TSC_G4_IO1	I2C1_SCL	MCO	-	-
PA10	TIM17_BKIN	USART1_RX	TIM1_CH3	TSC_G4_IO2	I2C1_SDA	-	-	-
PA11	EVENTOUT	USART1_CTS	TIM1_CH4	TSC_G4_IO3	CAN_RX	I2C1_SCL	-	-
PA12	EVENTOUT	USART1_RTS	TIM1_ETR	TSC_G4_IO4	CAN_TX	I2C1_SDA	-	-
PA13	SWDIO	IR_OUT	USB_NOE	-	-	-	-	-
PA14	SWCLK	USART2_TX	-	-	-	-	-	-
PA15	SPI1 NSS, I2S1 WS	USART2_RX	TIM2_CH1_ETR	EVENTOUT	-	USB_NOE	-	-

Available functions for STM32 GPIO pins (STM32F042G4U6)

External peripherals – General Purpose Input Output ports

Each GPIO pin can sink or source some current. Its value is defined by the manufacturer and is given in datasheets (as maximum current per pin and maximum current total for all pins).

The current characteristic depends on microcontroller type and can be asymmetric – one of the logic states can sink/source more current than the other one.

8051 GPIOs can work with no more than 10mA of current per pin and 26 mA per port.

AVR microcontrollers can sink up to 20 mA and source 3 to 20mA (5V supply), depending on output type and AVR model. The second limitation is that the overall current flowing through the power supply pins of the AVR must not exceed 100mA or 200mA.

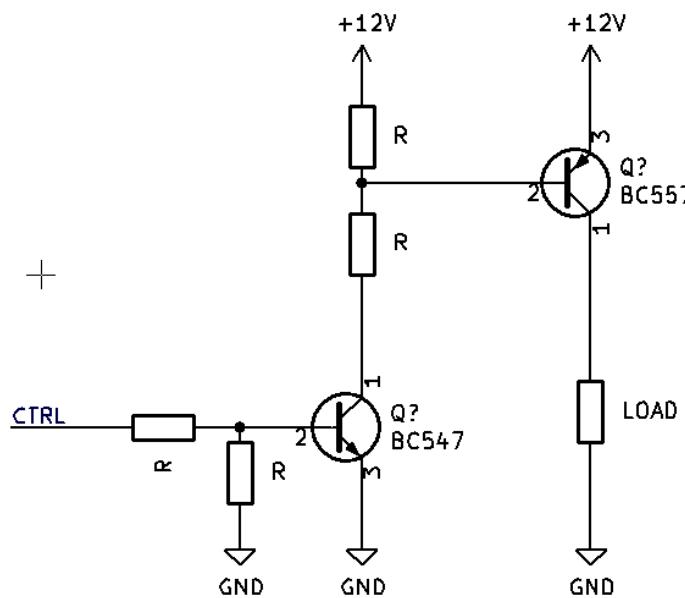
STM32 can sink/source up to 25 mA per pin and 75 mA total for all GPIO pins and control pins.

This clearly shows that we are able to drive LEDs directly from the GPIO pins, but to drive something more demanding (fan, motor, heater, etc.) a special current switch circuit is needed.

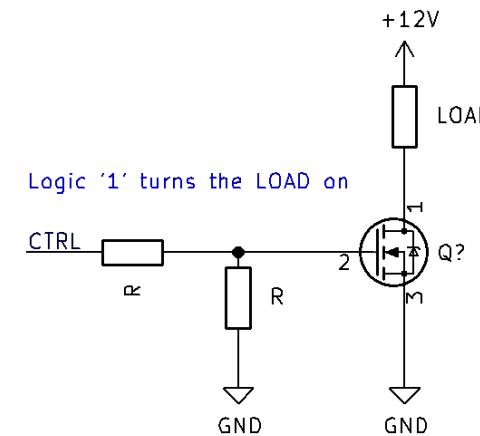
External peripherals – General Purpose Input Output ports

BJT and MOSFET current switches - examples

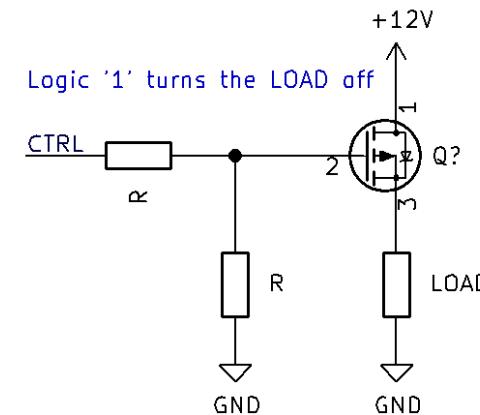
Pay attention to V_{TH} of the MOSFET!



BJT two stage current switch,
Non-inverting



NMOS current switch, when LOAD
is connected to power supply



PMOS current switch, when LOAD
is connected to GND