|  |
| --- |
| **Basic Circuit for 8051** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **8051 PIN OUT**  **Power** - Vcc, Vss    **Reset**- RST    **Crystal** - XTAL[1,2]    **External device interfacing**        – EA, ALE, PSEN, WR, RD    **I/O Port**        – P0[7;0], P1[7:0], P2[7:0], P3    **P3 is shared with control lines**          – Serial I/O RxD, TxD,           – external interrupts INT0,  INT1           – Counter control T0, T1    **P0 and P2 are multiplexed with Address and Data bus**        **BASIC CIRCUIT -THAT MAKES 8051 WORKS.**      **EA/VP Pin**  The EA on pin 31 is tied high to make the 8051 executes program from Internal ROM    **Reset Circuit**  RESET is an active High input  When RESET is set to High, 8051 goes back to the power on state.  The 8051 is reset by holding the RST high for at least two machine cycles and then returning it low.  **Power-On Reset**      - Initially charging of capacitor makes RST High      - When capacitor charges fully it blocks DC.  **Manual reset**       -closing the switch momentarily will make RST High.   After a reset, the program counter is loaded with 0000H but the content of on-chip RAM is not affected.      |  |  |  |  | | --- | --- | --- | --- | | **Register** | **Content** | **Register** | **Content** | | Program counter | 0000h | IP | XXX00000b | | Accumulator | 00h | IEv | 0XX00000b | | B register | 00h | All timer registers | 00h | | PSW | 00h | SCON | 00h | | SP | 07h | SBUF | 00h | | DPTR | 0000h | PCON (HMOS) | 0XXXXXXXbv | | All ports | FFh | PCON (CMOS)v | 0XXX0000b |             Note: content of on-chip RAM is not affected by Reset.    **Oscillator Circuit**  The 8051 uses the crystal for precisely that: to synchronize it’s operation. Effectively, the 8051 operates using what are called "machine cycles." A single machine cycle is the minimum amount of time in which a single 8051 instruction can be executed. although many instructions take multiple cycles.   8051 has an on-chip oscillator. It needs an external crystal that’s decides the operating frequency of the 8051.   This can be achieved in two ways,,  The crystal is connected to pins 18 and 19 with stabilizing capacitors. 12 MHz(11.059MHz) crystal is often used and the capacitance ranges from 20pF to 40pF.  The oscillator can also be a TTL clock source connected with a NOT gate asshown    **How fast 8051 works ?**  A cycle is, in reality, 12 pulses of the crystal. That is to say, if an instruction takes one machine cycle to execute, it will take 12 pulses of the crystal to execute. Since we know the crystal is pulsing 11,059,000 times per second and that one machine cycle is 12 pulses, we can calculate how many instruction cycles the 8051 can execute per second:  11,059,000 / 12 = 921,583  **Why is such an oddball crystal frequency?**  11.0592 MHz crystals are often used because it can be divided to give you exact clock rates for most of the common baud rates for the UART, especially for the higher speeds (9600, 19200). Despite the "oddball" value, these crystals are readily available and commonly used.    **Power Supply**    C1-1000 mf ,C2-100 mf  The 78L05 is a 5V regulator. The input voltage ranges from 7V to 35V and the output voltage is about 5V.      **Using Ports for I/O Operation**  8051 is TTL logic device. TTL logic has two levels: Logic "High" (1) and logic "Low" (0). The voltage and current involved for the two levels are as follows:     |  |  |  | | --- | --- | --- | | **Level** | **Voltage** | **Current** | | High | Above 2.4V | Virtually no current flow | | Low | Below 0.9V | 1.6mA Sinking current from TTL input to ground (Depends on logic family) |     **Port functions**   |  |  | | --- | --- | | **Ports** | **Function** | | **Port 0**  (Pin 32-39) | **Dual-purpose port**- 1. general purpose I/O Port.                                  2. multiplexed address & data bus                                                            Open drain outputs | | **Port 1**  (Pin 1-8) | **Dedicated I/O port** – Used solely  for interfacing to external devices                                                 Internal pull-ups | | **Port 2**  (Pin 21-28) | **Dual-purpose port**-  1. general purpose I/O port.                                    2. a multiplexed address & data bus.                                    Internal pull-ups | | **Port 3**  (Pin 10-17) | **Dual-purpose port**-  1. general purpose I/O port.                                   2. pins have alternate purpose related to special features of the 8051                                                  Internal pull-ups |     The 8051 internal ports are partly bi-directional (Quasi-bi-directional). The following is the internal circuitry for the 8051 port pins:    **1.Configuring for output**  P0 is open drain.  – Has to be pulled high by external 10K resistors.  – Not needed if P0 is used for address lines  Writing to a port pin loads data into a port latch that drives a FET connected to the port pin.  **P0:** Note that the pull-up is absent on Port 0 except when functioning as the external address/data bus. When a "0" is written to a bit in port 0, the pin is pulled low. But when a "1" is written to it, it is in high impedance (disconnected) state. So when using port 0 for output, an external pull-up resistor is  needed, depending on the input characteristics of the device driven by the port pin    **P1, P2, P3 have internal pull-ups:** When a "0" is written to a bit in these port , the pin is pulled low ( FET-ON) ,also when 1 is written  to a bit in these port pin becomes high (FET-OFF) thus using port P1,P2,P3 is simple.  **2. Configuring for input**  At power-on all are output ports by default  To configure any port for input, write all 1’s (0xFF) to the port      Latch bit=1, FET=OFF, Read Pin asserted by read instruction  You can used a port for output any time. But for input, the FET must be off. Otherwise, you will be reading your own latch rather than the signal coming from the outside. Therefore, a "1" should be written to the pin if you want to use it as input, especially when you have used it for output before. If you don't do this input high voltage will get grounded through FET so you will read pin as low and not as high. An external device cannot easily drive it high  so, you should not tide a port high directly without any resistor. Otherwise, the FET would burn.  **Be Careful :**  Some port pins serve multiple functions. Be careful writing to such ports. For example, P3.0 is the UART RXD (serial input), and P3.1 is the UART TXD (serial output). If you set P3.0 to a '0', an external buffer (such as an RS232 level translator) cannot drive it high. Therefore you have prevented receiving any serial input.  If an external interrupt such as EX1 on P3.3 is enabled, and set to be level sensitive, and you clear this pin's output latch to a zero, guess what? You've just caused a perpetual interrupt 1. The pin's input buffer will read the output of it's latch as always low. Your controller will spend all of its time in the interrupt handler code and will appear to have crashed, since it will have very little time for other tasks. In fact, it will get to execute a single instruction before re-entering the interrupt handler, so the rest of your program will execute very, very slowly. |