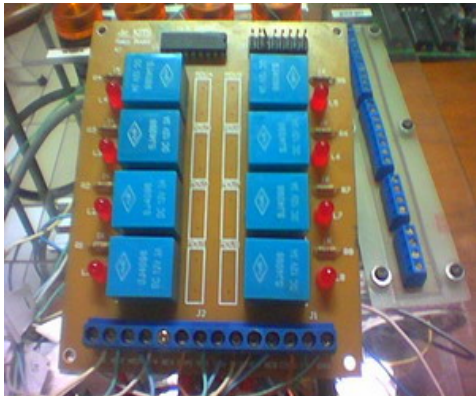
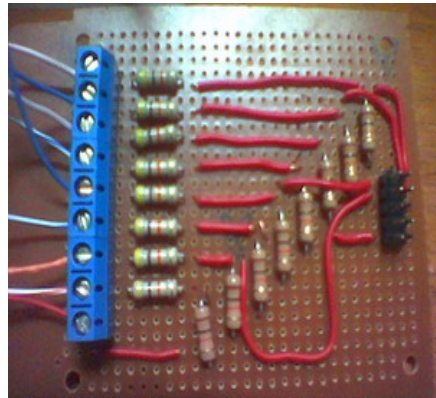


# **LAMPIRAN A**

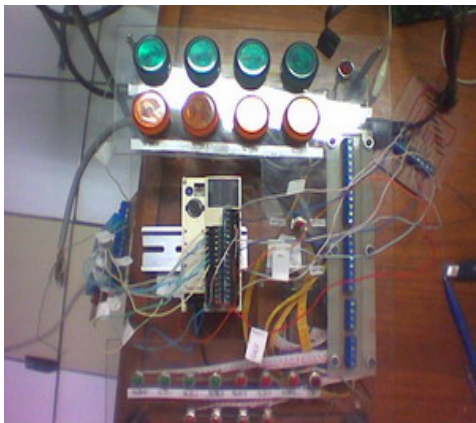
## FOTO ALAT



**Relay**



**Resistor**



**PLC**

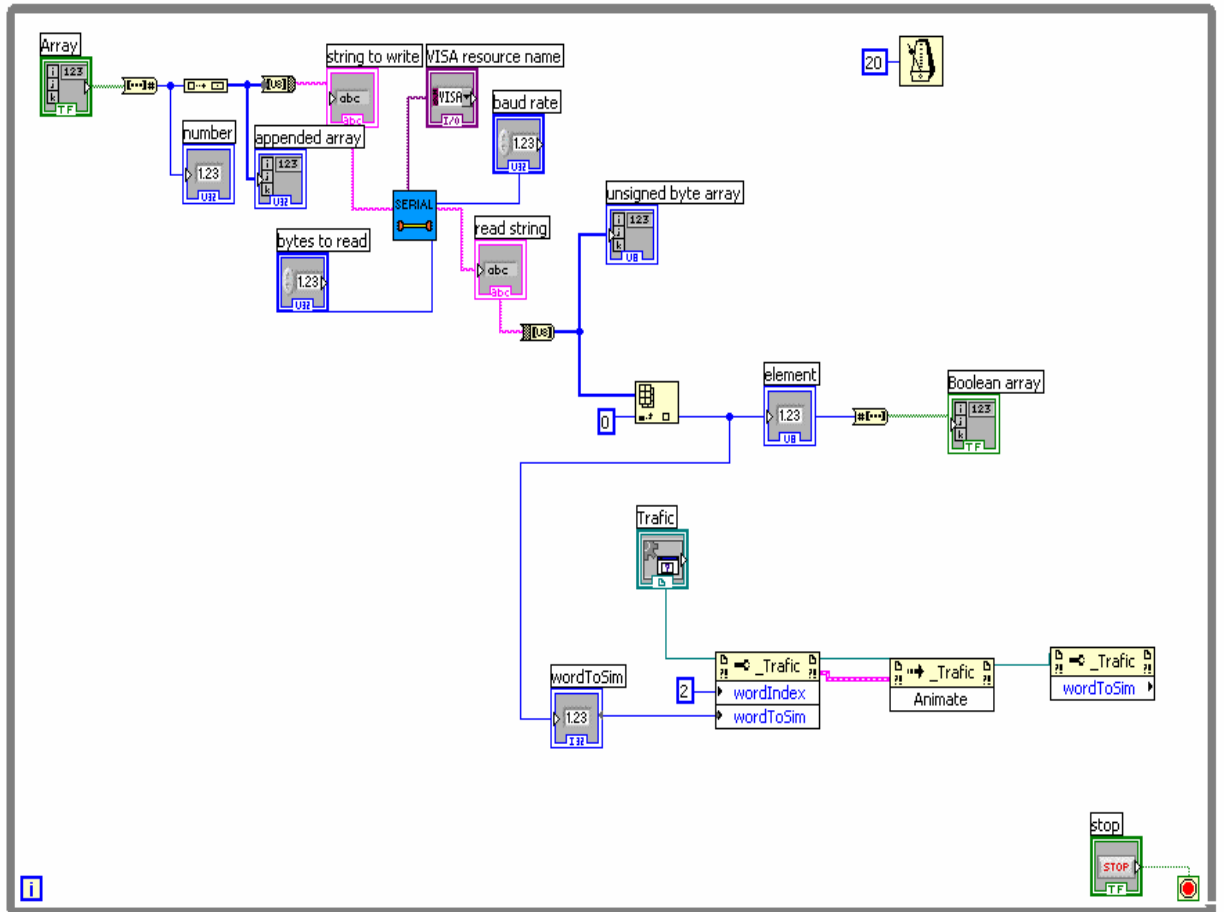


**MCS-51**



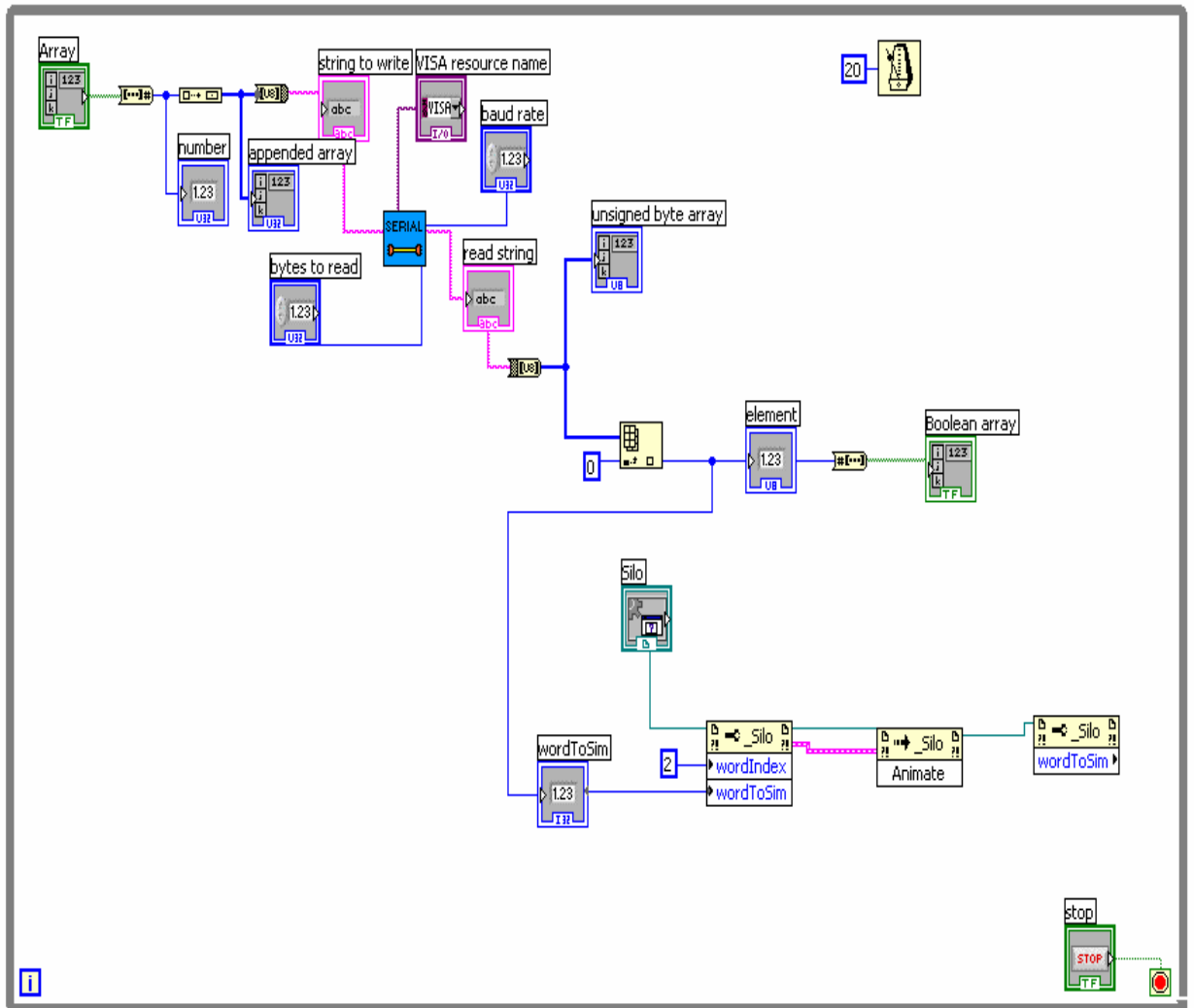
**MCS-51**

# **LAMPIRAN B**

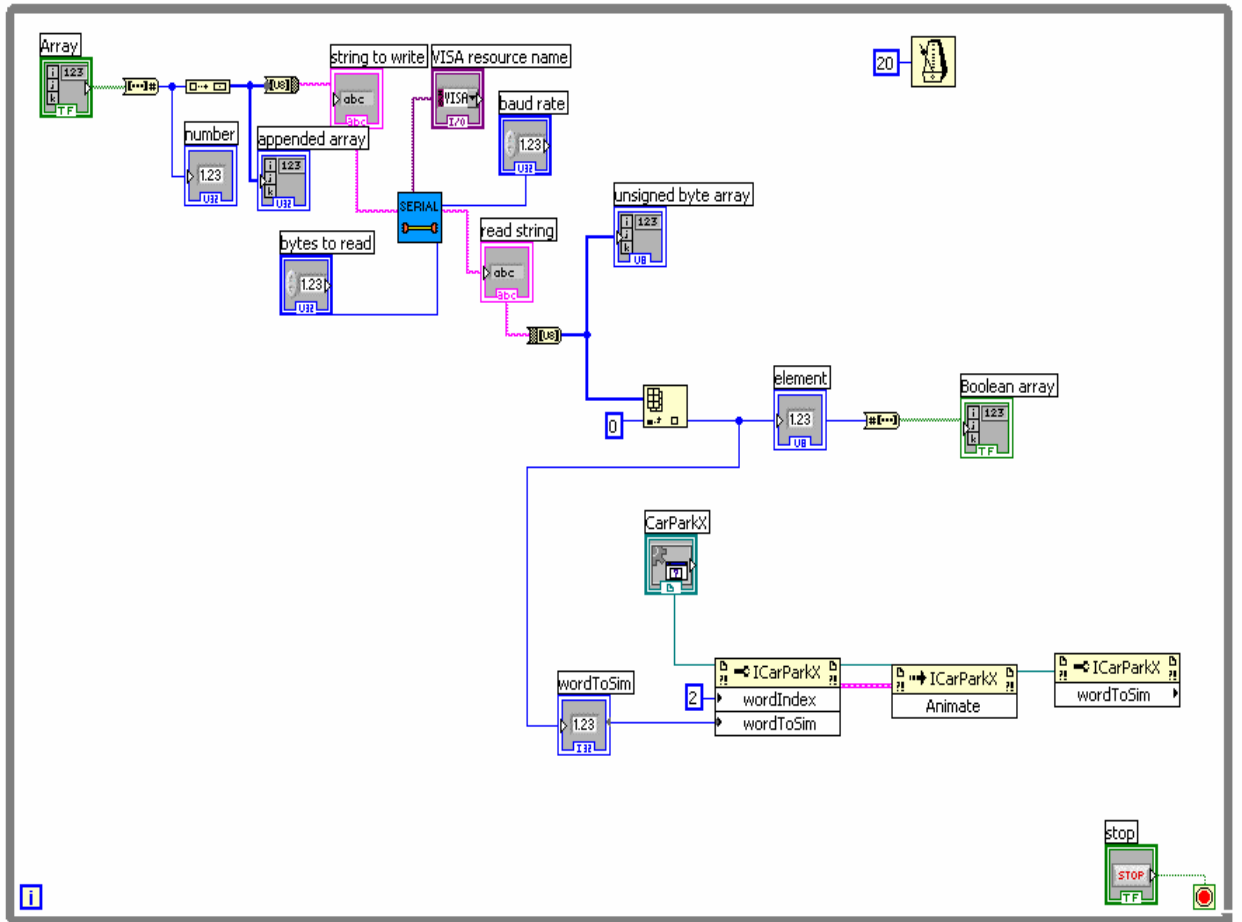


**Program LabVIEW *Traffic Light***





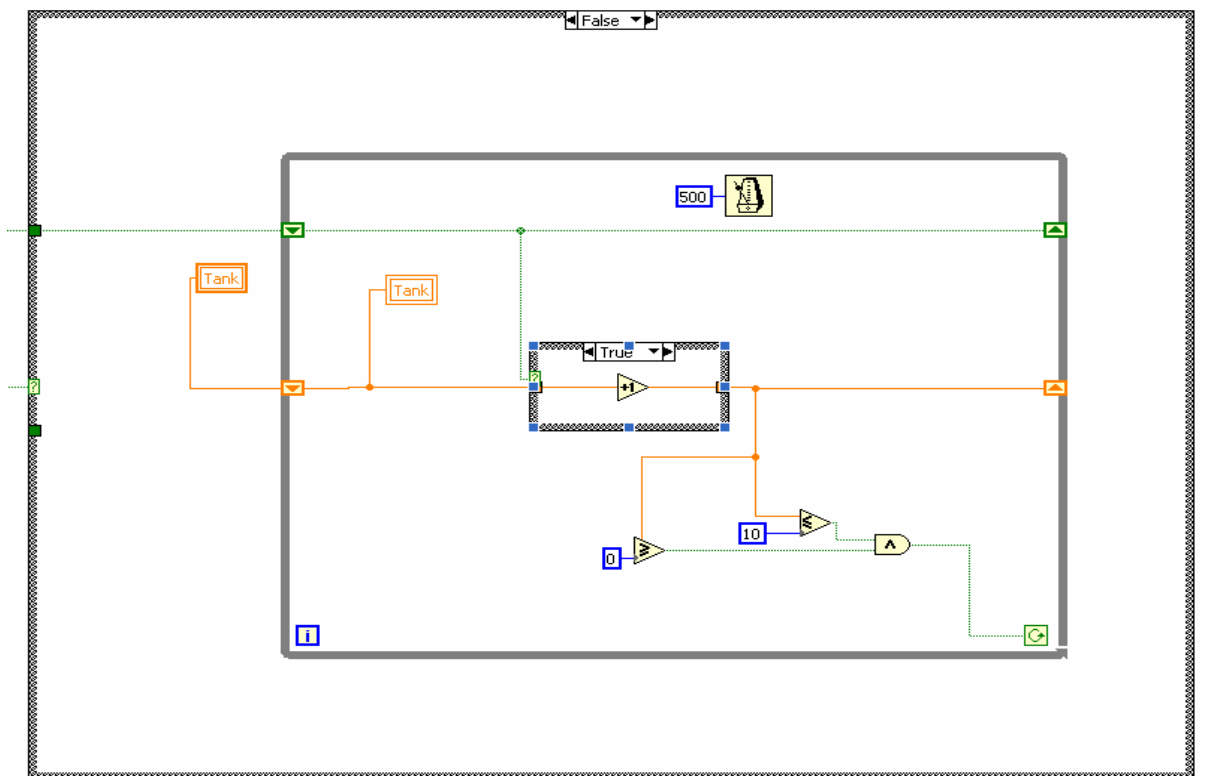
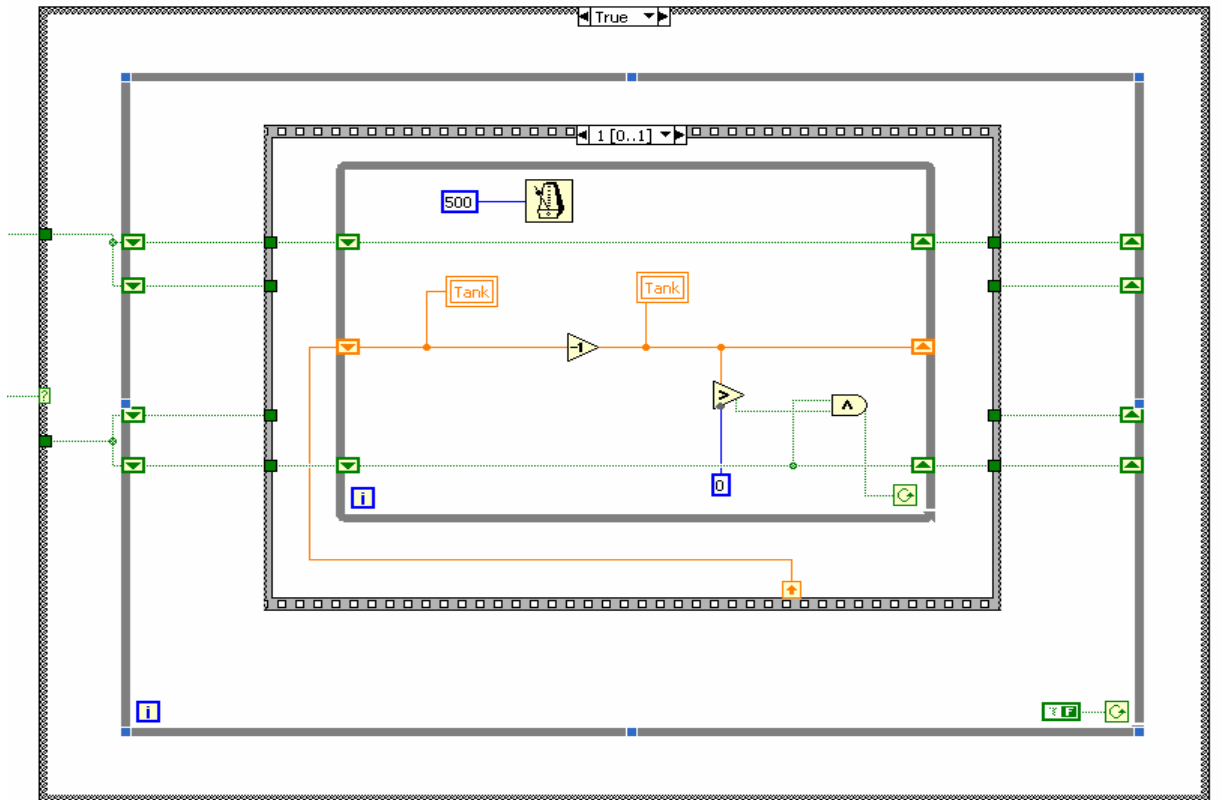
**Program LabVIEW Pada *Silo Simulation***

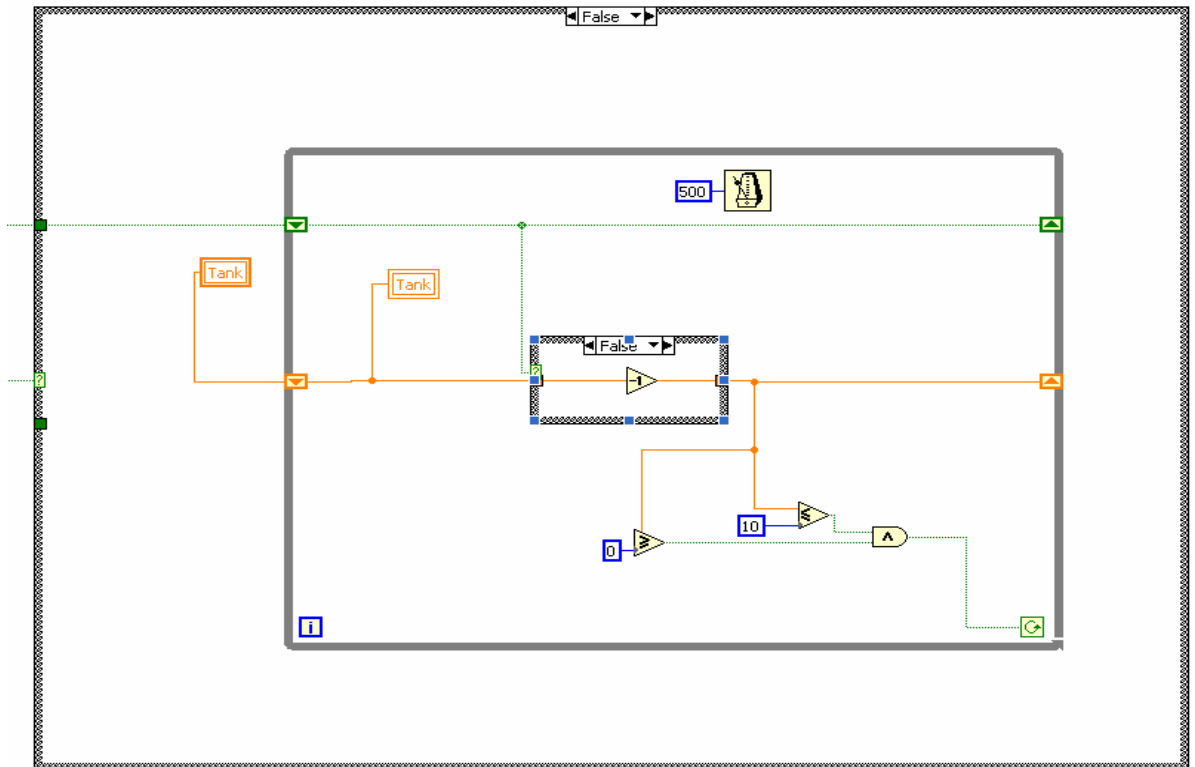


Program LabVIEW Pada *Parking Lot*









**Program LabVIEW Pada Tanki Air**

**Program Pada MCS-51**

```

$mod51
    CSEG
    ORG 4000H
    LJMP START

interrupt penerimaan
    ORG 4023H
    CLR RI
    MOV A, SBUF
    MOV DPTR,#2001H
    MOVX @DPTR, A

mengirimkan data ke PC
    MOV DPTR,#2000H
    MOVX A, @DPTR
    mov p1,a
    MOV SBUF, A
    JNB TI, $
    CLR TI
    RETI

```

interrupt diaktifkan dan serial berada pada mode 1 dengan REN = 1

```

START:    MOV SP, #30H
          MOV A, #90H
          MOV DPTR, #2003H
          MOVX    @DPTR, A
          MOV SCON, #50H
inisialisasi baud rate (9600 bps)
          MOV TMOD, #20H
          MOV TL1, #0FDH
          MOV TH1, #0FDH
          MOV PCON, #00H
          SETB TR1
          SETB ES
          SETB EA
pas:     MOV R7, #10H
LUPA:   MOV R6, #0FFH
LUPB:   MOV R5, #0FFH
          DJNZ R5, $
          DJNZ R6, LUPB
          DJNZ R7, LUPA
          MOV DPTR, #2001H
          MOVX    @DPTR, A
mengirimkan data ke PC
          MOV DPTR, #2000H
          MOVX    A, @DPTR
          mov  p1, a
          MOV SBUF, A
          JNB  TI, $
          CLR  TI
          sjmp pas
END

```

## LabVIEW Tutorial on Spectral Analysis

With the LabVIEW graphical programming environment, you can quickly and easily create many different types of measurement analysis applications. LabVIEW has built-in functions for measurement analysis including RMS calculation, peak detection, harmonic analysis, filtering, and frequency analysis functions, as well as a large number of complex mathematical and statistical functions.

**Purpose:** In this tutorial, you will create a LabVIEW virtual instrument (VI) that generates a sine wave, uses one of the LabVIEW analysis functions to calculate the power spectrum of the signal with a Fast Fourier Transform (FFT), and creates a plot of the frequency spectrum.

**Completion Time:** This tutorial will take approximately 45 minutes.

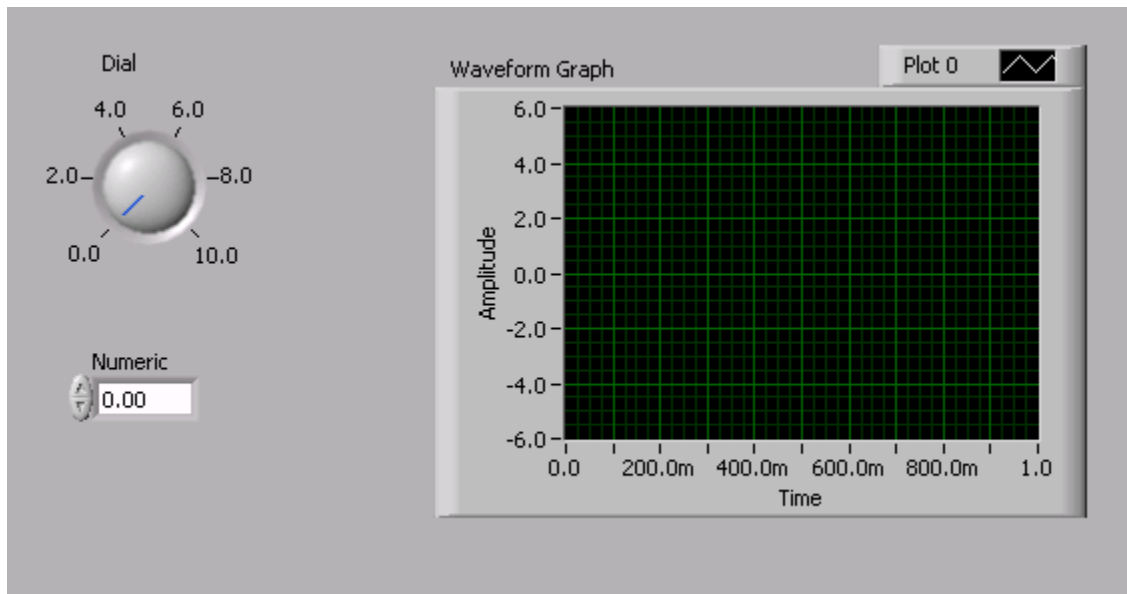
**Programming Level:** This tutorial is designed for LabVIEW users of any level. However, if this is your first time using LabVIEW, consider starting out with our introductory tutorial entitled “Acquire, Analyze, and Present with LabVIEW”. It contains detailed information about the fundamentals of graphical programming, so you can quickly become familiar with the LabVIEW environment. You can find it at <http://www.ni.com/tutorials>.

If you would like more in-depth technical information about spectral analysis, the National Instruments Web site offers a multimedia-based “FFT Interactive Tutorial” on the fundamentals of frequency domain measurements. It can also be found at <http://www.ni.com/tutorials>.

Follow these steps to create your own VI:

- 1.) Open your internet browser and enter <http://www.ni.com/info>. Enter “trylvnow” as the info code.
- 2.) If you have already created a profile, enter your e-mail address and password and click “Continue”. If not, creating one will only take a few seconds; to begin, simply select “No. I need to create a profile” and click “Continue”. Once you have created your profile, move on to step 3.
- 3.) Click on the “Start LabVIEW” button. A pop-up window will instruct you to do a one-time installation of a small client.
- 4.) From the LabVIEW 6.1 splash screen, choose New VI. This will open up two blank windows. The gray window titled “Untitled 1” is the front panel and will be your user interface, and the white window titled “Untitled 1 Diagram” is the block diagram, where you will create your graphical source code.
- 5.) Click on the front panel window and create the user interface shown below by performing steps 6-10.

**\*USAGE TIP:** You may also access the front panel by selecting Window»Show Panel from the block diagram menu bar.

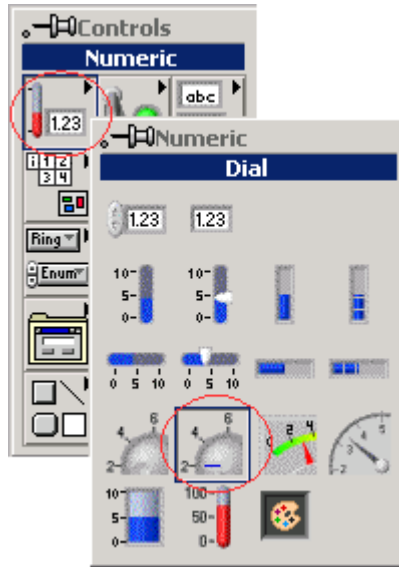


- 6.) Click on the Controls palette window on the left side of the screen. The Controls palette contains objects (i.e. controls and indicators) that can be placed on the front panel to form the user interface. Mouse over the sub-palettes and you will see their descriptions in the top of the window.

**\*USAGE TIP:** If you cannot see the Controls palette, you can view it by selecting Window»Show Controls Palette from the menu or by right-clicking on a blank area on the front panel window.

**\*DEFINITION:** A **control** is a front panel object that serves as a data source, getting data from the user via the front panel and sending it to the block diagram code (e.g. a knob). An **indicator** is a front panel object that serves as a data sink, obtaining data from the block diagram code and displaying it to the front panel (e.g. a graph). A front panel object cannot be both a control and an indicator.

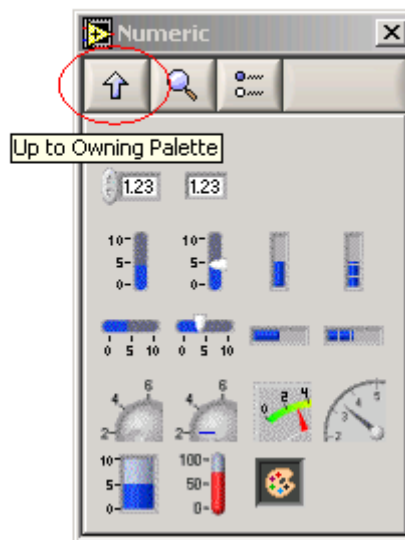
- 7.) From the Controls palette, select the Numeric sub-palette and click on the Dial control as shown in the figure below. Move the mouse to the front panel and click the left mouse button to drop the Dial onto the panel.



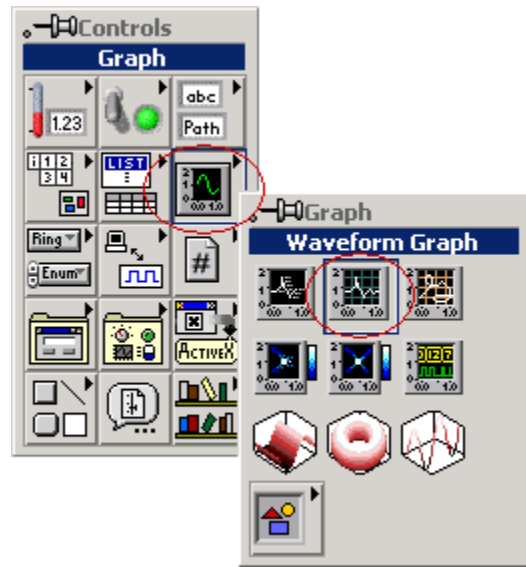
**\*USAGE TIP:** You can reposition the Dial control on the panel by clicking and dragging with the positioning tool, shown as an arrow on the Tools palette. To select the positioning tool, either click on it in the Tools palette or press the <TAB> key to switch the currently selected tool until the positioning tool is shown.

**\*USAGE TIP:** If you cannot see the Tools palette, select Window»Show Tools Palette from the menu.

- 8.) From the Numeric sub-palette, choose the Digital Control, and drop it onto the front panel.
- 9.) Click on the up arrow at the top of the Numeric sub-palette window to go back to the Controls palette, as shown below.

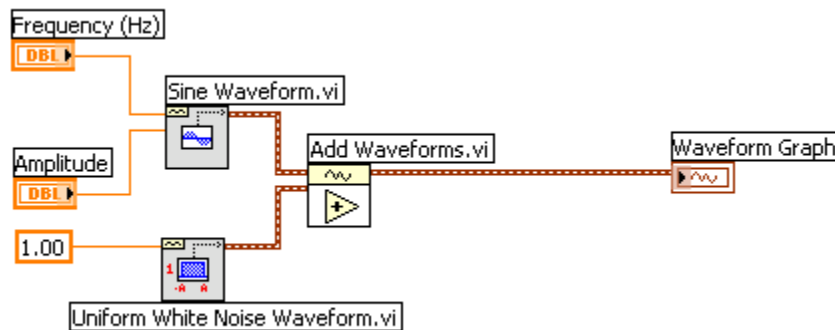


- 10.) From the Controls palette, select the Graph sub-palette and click on the Waveform Graph as shown below. Place it onto the front panel.



**\*DEFINITION:** The **Waveform Graph** is an indicator that accepts an array of data values and plots the entire array at once. This is different from the **Waveform Chart**, which scrolls data continuously, adding new data points to those already displayed.

The next step is to create the initial block diagram code. Build the block diagram pictured below by performing steps 11-28.



- 11.) Click on the block diagram window (behind the front panel). You should see three nodes on the block diagram. These nodes correspond to the three objects already placed on the front panel.

**\*INFO TIP:** The two control nodes (Dial and Numeric) have small black arrows on the right, and the Waveform Graph indicator has a similar arrow on the left. These arrows indicate the direction of data flow, either into or out of the node. All of the nodes are orange, indicating that they represent floating-point values, and they all contain the letters DBL, indicating that they are double precision.

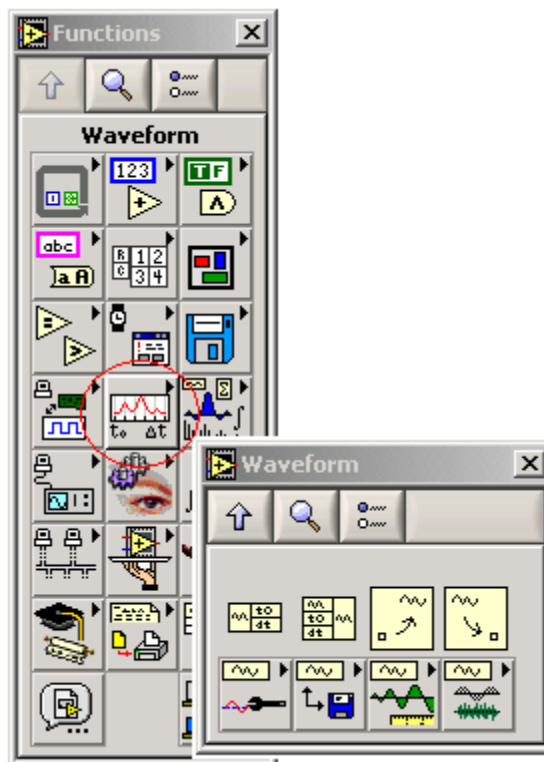
- 12.) Note that after clicking on the block diagram window, the Controls palette has been replaced by the Functions palette. The Functions palette is used to

build the block diagram. It contains all of the LabVIEW functions that can be combined to create your custom application.

- 13.) Reposition the Functions palette so that you can view the entire window on your screen.

**\*USAGE TIP:** If you cannot see the Functions palette, you may show it manually by selecting Window»Show Functions Palette from the menu.

- 14.) Within the Functions palette, select the Waveform sub-palette as shown below. This palette contains functions performed on the waveform data type. The waveform data type is a unique feature of LabVIEW. It is a special kind of cluster that succinctly contains both amplitude and timing information of a signal.



**\*DEFINITION:** A **cluster** is a data type that bundles together many different kinds of information, similar to several wires carrying various types of data wrapped together as one. This is also similar to the structure used in text-based programming languages.

**\*DEFINITION:** The **waveform data type** consists of a cluster of three elements. These three elements are an array of Y values containing the signal amplitude, a single value  $t_0$  that represents the start time of the signal, and a single value  $dt$  that represents the time interval between data points in the Y array. When the waveform data type is passed to the Waveform Graph, the graph interprets the data and plots the timing information on the X-axis along with the array of amplitudes on the Y-axis.



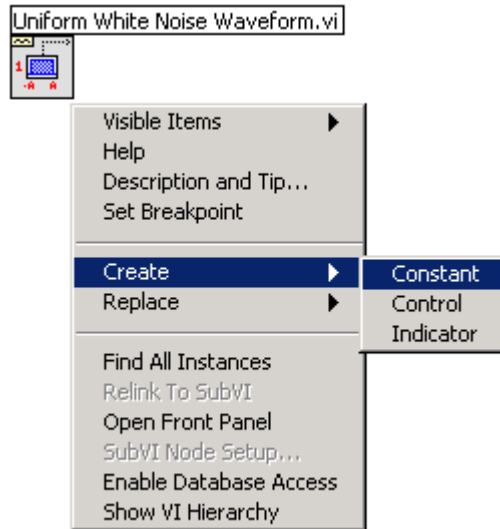
- 15.) Within the Waveform palette, select the Waveform Generation menu. The functions in this menu automatically generate many commonly used waveforms.
- 16.) Choose the Sine Wave.vi and drop it onto the block diagram.

**\*USAGE TIP:** If you would like more information on how Sine Wave.vi or any other VI works, simply activate the Context Help by selecting Help»Show Context Help or by pressing <CTRL><H>. Put the mouse pointer on any object in the block diagram, and the context help window will give a description of that VI or object. For additional in-depth information about a VI click on the “Click here for more help” button inside the Context Help window to open the LabVIEW Help page for that item. The LabVIEW Help page contains specific information about each input and output and about the function of the VI. You can also view the front panel and/or block diagram of the SineWave.vi by double-clicking on the node or right-clicking on it and selecting Open Front Panel.

- 17.) From the same Waveform Generation menu, choose the Uniform White Noise Waveform.vi and place it on the block diagram.
- 18.) Using the wiring tool (shown as a spool of wire in the Tools palette), place the mouse over the Sine Wave.vi and notice that the input and output terminals appear on the edges of the VI icon. The input or output terminal that the wiring tool is currently pointing to blinks on the block diagram as well as in the Context Help window (press <CTRL><H>), and its label is displayed in a tip-strip.
- 19.) Select the ‘frequency’ input by clicking on the upper left input terminal. A broken wire will now trail the mouse as you move it.
- 20.) Click on the Dial control and notice that a solid orange wire has been connected from the Dial control node to the Sine Wave.vi ‘frequency’ input. The orange color of the wire symbolizes a floating-point value.

**\*USAGE TIP:** You can click the left mouse button anytime during wiring to lock the current position of the wire onto the screen. Then you can move the mouse up or down to better control how the wire is placed. You can also move wires around by clicking and dragging them with the positioning tool (arrow). To cancel the placement of a wire, click the right mouse button during wiring or highlight the undesired wire and press <DELETE>. To eliminate all broken wires on the block diagram at once, press <CTRL><B>. The VI will not run if there are broken wires present in the block diagram.

- 21.) Rename the Dial control by double-clicking on the label with the labeling tool (capital “A”). This will highlight the label, allowing you to type in new text. Change the label to “Frequency (Hz)”, since this dial has been wired to control the frequency input to the Sine Wave.vi.
- 22.) Using the same method explained in steps 19-21 above, wire the ‘amplitude’ input of the Sine Wave.vi to the Numeric digital control node and change the label of the Numeric control to “Amplitude.”
- 23.) With the wiring tool, right-click on the ‘amplitude’ input of the Uniform White Noise Waveform.vi and select Create»Constant as shown below. The default amplitude of the noise is 1.00. You can change this value by double-clicking on it with the labeling tool (capital “A”) and typing in a new number. For this example, we will use the default value.



**\*DEFINITION:** A **constant** exists only on the block diagram and cannot be changed while the VI is running. This is different from a control, which exists on the front panel and can be changed at any time by the user.

- 24.) From the Waveform Generation palette, click the Up arrow at the top left of the window (described in step 9) to go back to the Waveform palette.
- 25.) From the Waveform palette, choose the Waveform Operations menu. Click on the Add Waveforms.vi and place it onto the block diagram.
- 26.) Using the wiring tool, connect the 'signal out' output of the Sine Wave.vi to the upper left 'waveform A' input of the Add Waveforms.vi
- 27.) Connect the 'signal out' output of the Uniform White Noise Waveform.vi to the 'waveform B' input of the Add Waveforms.vi.
- 28.) Connect the 'result' output terminal of the Add Waveforms.vi to the Waveform Graph indicator node. Notice that the wire used to connect these two is wide and brown in color. This type of wire represents a cluster, more specifically, the waveform data type cluster discussed previously.

At this point, you have created a working VI that generates a waveform containing a noisy sine wave and plots it on a Waveform Graph. If everything is wired correctly, you will see a white Run arrow button as the first button in the toolbar at the top of the front panel and block diagram windows. If the arrow is gray and broken, follow the debugging instructions in the tip below. If the Run arrow is solid white, there are no errors and you can now run your VI.

**\*DEBUGGING TIP:** If the Run arrow is broken, click on the broken Run arrow to view the error list. Click on an error description to highlight it and read the details given in the Details window to help solve the problem. Then, click the Show Error button. This will clearly show the problematic area on the block diagram.

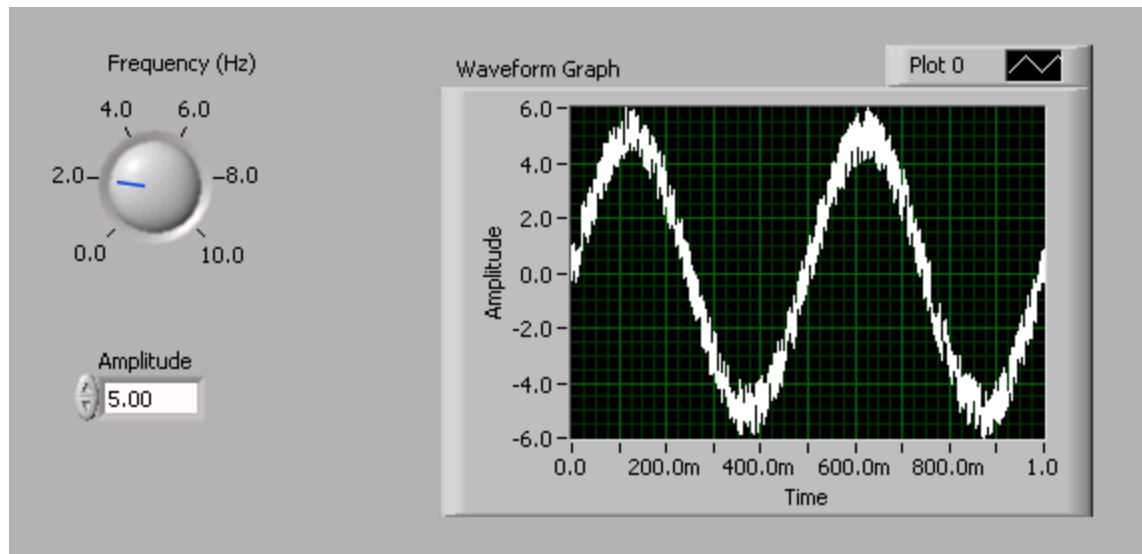
- 29.) Go to the front panel and enter values for the frequency and amplitude of your sine wave using the operating tool (hand) to click and drag the dial and

to increment or decrement the digital control arrows. Set the frequency to 2.0 Hz and the amplitude to 5.00.

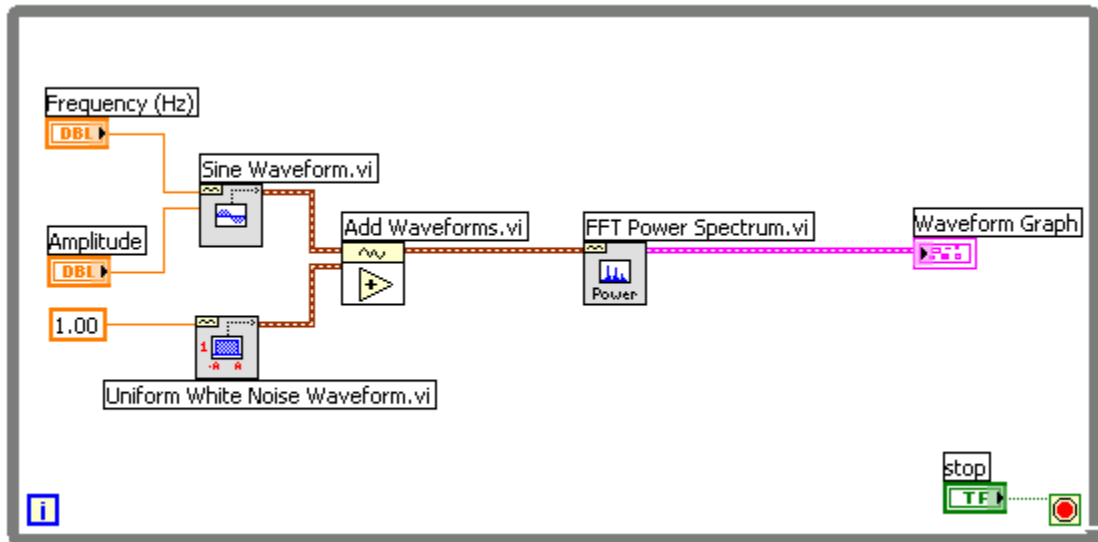
**\*USAGE TIP:** You can enter a value directly into the digital control by double-clicking the number in the field and typing in a new number.

**\*USAGE TIP:** To more precisely set the value of the frequency dial, right-click on the dial, select Visible Items and choose Digital Display. This will bring up a numeric window where you can view or change the dial value digitally.

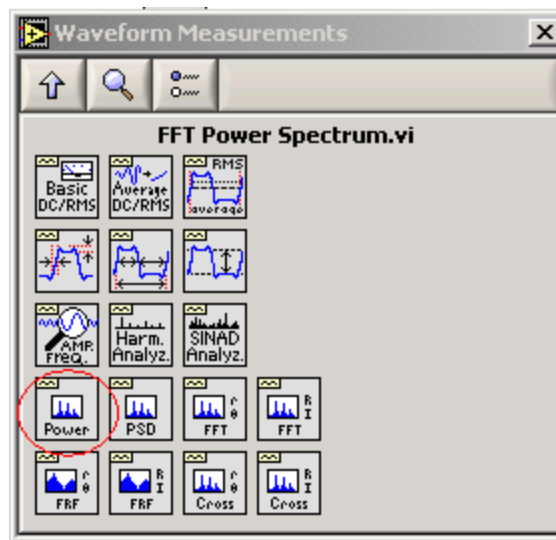
- 30.) Run the VI by clicking the Run arrow. On the front panel you should see a noisy sine wave with the proper frequency and amplitude plotted on the Waveform Graph, as shown in the figure below. If you do not see this, be sure to check that you have entered non-zero values for the input frequency and amplitude and that your block diagram mimics the figure given above.



Now that we have generated the noisy sine wave, the next step is to perform the frequency spectrum analysis. LabVIEW makes this very easy by providing a built-in FFT Power Spectrum function. Create the block diagram code pictured below to calculate and plot the power spectrum of the noisy sine wave by performing steps 31-45.



- 31.) Switch to the block diagram and delete the wire connecting the Add Waveforms.vi to the Waveform Graph node by highlighting it with the positioning tool and pressing <DELETE>.
- 32.) Click the Up arrow on the Waveform Operations palette to go back to the Waveform palette. Choose the Waveform Measurements menu. Inside this menu, select the FFT Power Spectrum.vi as shown below, and drop it onto the block diagram.



**\*INFO TIP:** In general, the FFT Power Spectrum.vi accepts any waveform as an input and sends the computed power spectrum of the waveform to the output. The parameters used to determine the power spectrum are changeable, including the time-domain window type, the specific averaging method, and the units applied to the output. Use the Context Help window (<CTRL><H>) to read more about how this VI works, or look at the underlying code by double-clicking on the VI to view its contents.

- 33.) Place the wiring tool on the FFT Power Spectrum.vi icon to view the input and output terminals.
- 34.) Connect the 'time signal' input of the FFT Power Spectrum.vi to the 'result' output of the Add Waveforms.vi.
- 35.) Connect the 'power spectrum' output of the FFT Power Spectrum.vi to the Waveform Graph indicator node. Notice that the wire connecting the FFT Power Spectrum.vi to the Waveform Graph is pink. The pink color in this case denotes a general cluster data type.

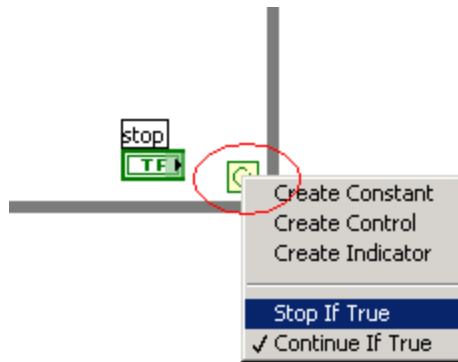
**\*DEFINITION:** The Waveform Graph node itself has also changed from brown to pink. This graph is **polymorphic**, meaning that it accepts many different types of data, including waveforms, clusters, and arrays. The ability to use polymorphic functions is a feature of LabVIEW that provides versatility in your programming by not restricting you to traditional, explicit type casting rules.

Next, we will add a While Loop to run continuous iterations of the sine wave generation and frequency analysis. This is necessary in order to utilize the averaging features of the FFT Power Spectrum.vi.

- 36.) From the Waveform Measurements palette, click the Up arrow twice to go back to the general Functions palette.
- 37.) Select the Structures sub-palette and choose the While Loop structure. Then, place the While Loop around the existing code by clicking and holding down the mouse button at the upper left corner of the block diagram window and dragging the box to enclose the entire diagram.
- 38.) Go to the front panel. Click the Up arrow until you see the general Controls palette.
- 39.) Select the Boolean sub-palette, and choose the STOP button. Drop the STOP button onto the front panel.

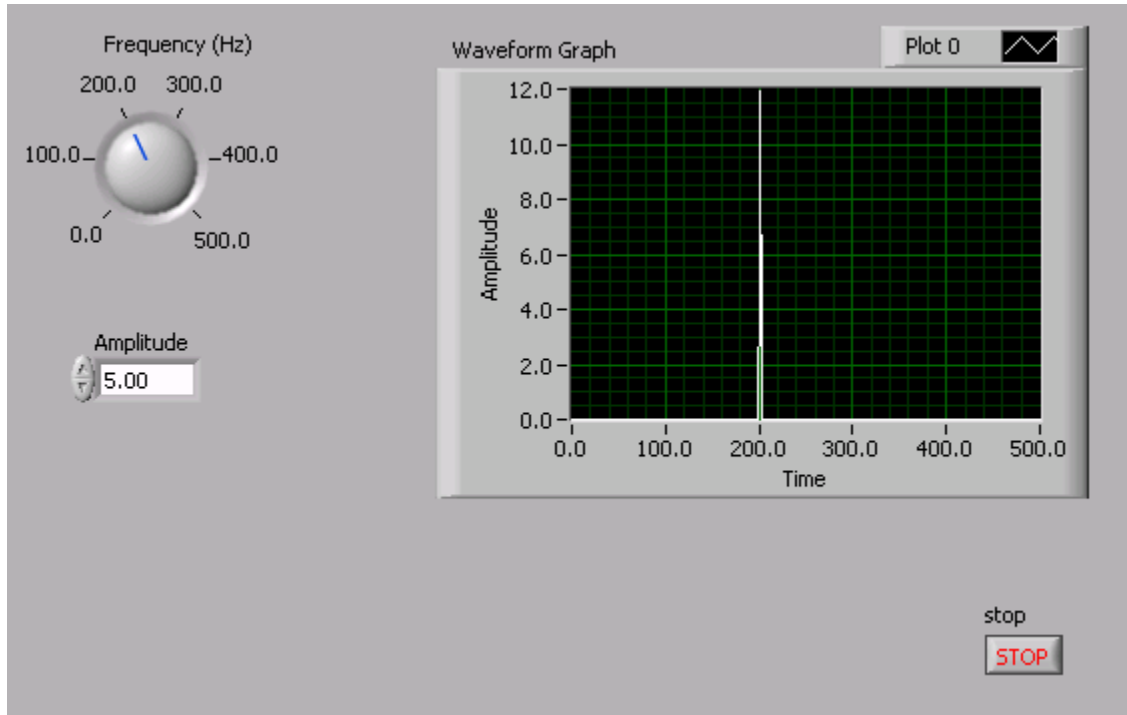
**\*USAGE TIP:** It is important to always have a way to stop any loops that you create in LabVIEW. The STOP button is the best way to easily control while loops.

- 40.) Go back to the block diagram and locate the STOP button node. Use the positioning tool to drag this node into the while loop, if it is not already there.
- 41.) Right-click on the continuation terminal (green circular arrow) of the while loop in the bottom right corner and select Stop if True, as shown below. You will see the terminal change from a green arrow to a red stop sign.

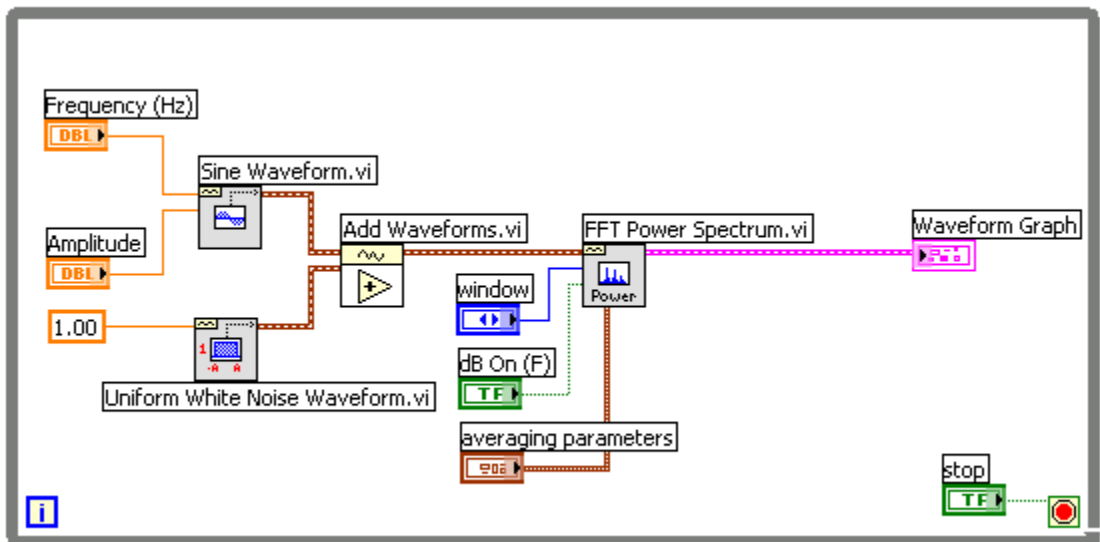


- 42.) Wire the STOP button node to the continuation terminal with the wiring tool.
- 43.) Go to the front panel and change the range of the frequency dial to 0-500 Hz using the labeling tool (capital "A"). Double-click on the highest dial value marker and type "500." Press Enter or click on the check box at the upper left corner of the window to confirm your entry.
- 44.) With the labeling tool, double-click on the 'Time' label for the X-axis of the Waveform Graph and change it to 'Frequency.'
- 45.) Set the dial to 200 Hz using the operating tool.
- 46.) If the run arrow is solid white and there are no errors in the block diagram, run the VI and observe its behavior. You should see a plot of the power spectrum on the Waveform Graph, with a spike in the plot at the frequency designated by the Dial control, as shown below. Turn the dial and notice the movement of the frequency spike.
- 47.) Push the STOP button to stop the VI and continue editing.

**\*DEBUGGING TIP:** Note that the VI now must be stopped using the STOP button. If the Run arrow is solid black, the VI is still running and must be stopped before it can be edited.



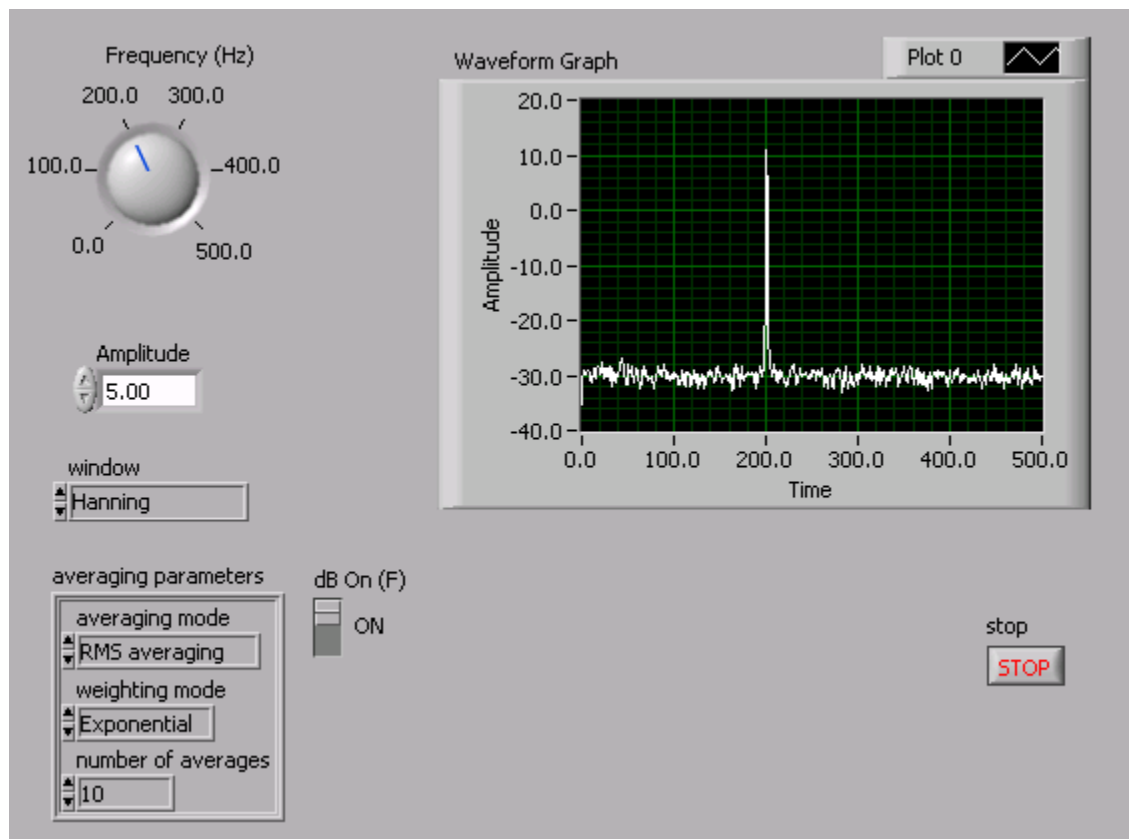
The FFT Power Spectrum.vi has built-in averaging capabilities to display the running average of multiple power spectrum calculations. To add these averaging capabilities and other features to your spectral analysis VI, modify your block diagram as shown below by performing steps 48-53.



- 48.) Go to the block diagram and place the wiring tool over the 'window' input of the FFT Power Spectrum.vi.
- 49.) Right-click and select Create»Control. This will create a 'window' control node on the block diagram and will automatically wire it to the terminal

from which it was created. It will also place a new control on the front panel.

- 50.) Use the positioning tool to click on the 'window' node and move it away from the other input terminals. Be sure when you are moving objects to highlight the entire object, not just the label.
- 51.) Locate the new ring control on the front panel (it may be off the screen below). Use the positioning tool to click on the 'window' control and drag it to a better location near the other front panel objects.
- 52.) Repeat the above steps to create controls for the 'averaging parameters' input, and the 'dB On' input. Doing this will add to the front panel a cluster control with elements specifying the 'averaging parameters', as well as a vertical slide switch control for 'dB On'.
- 53.) Set the front panel to the parameters shown below and run the VI.



Congratulations! You have created a fully functional sine wave spectral analyzer virtual instrument. Try adjusting the different averaging methods, the number of averages, and the windowing properties to see how they affect the power spectrum output.

You can bring any signal type into the provided LabVIEW signal analysis functions, whether the signal is generated, like the noisy sine wave in this example, or measured with hardware and acquired through the PC for your particular application. This example gives a small sample of the many powerful



tools that LabVIEW has to offer and demonstrates the ease with which you can create your own custom signal analysis applications.

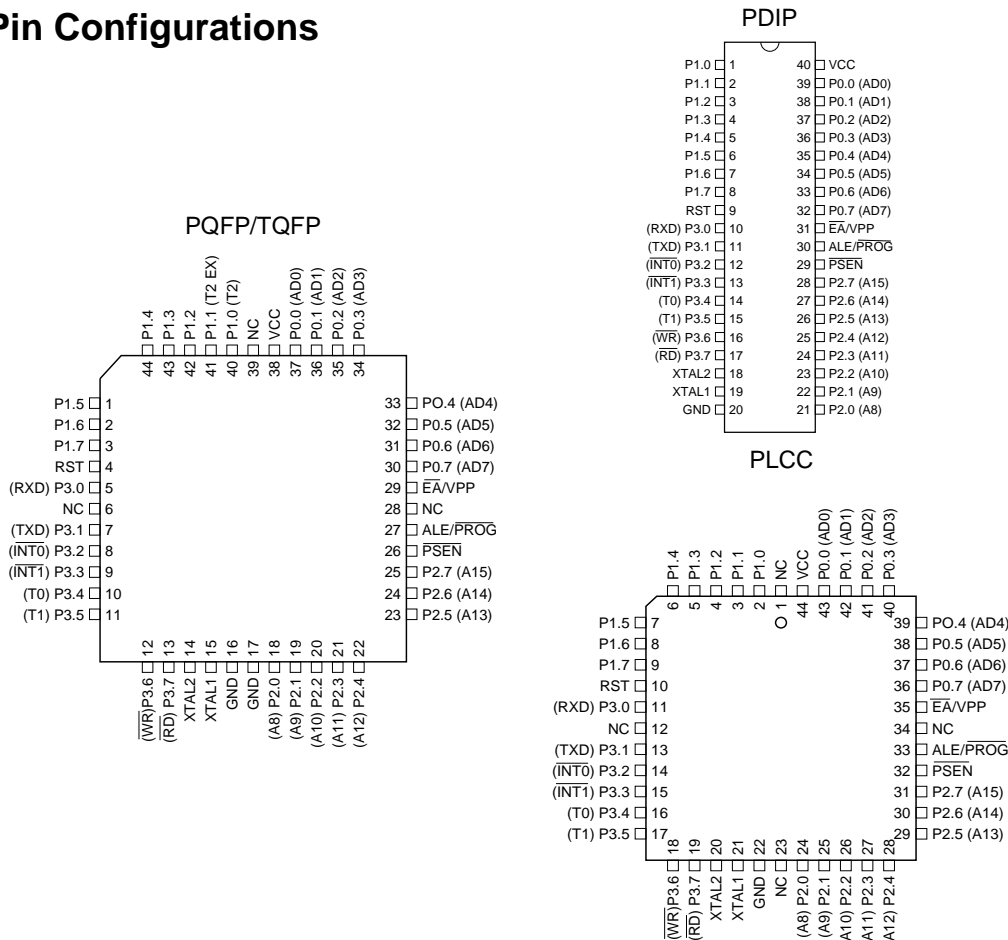
## Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

## Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

## Pin Configurations



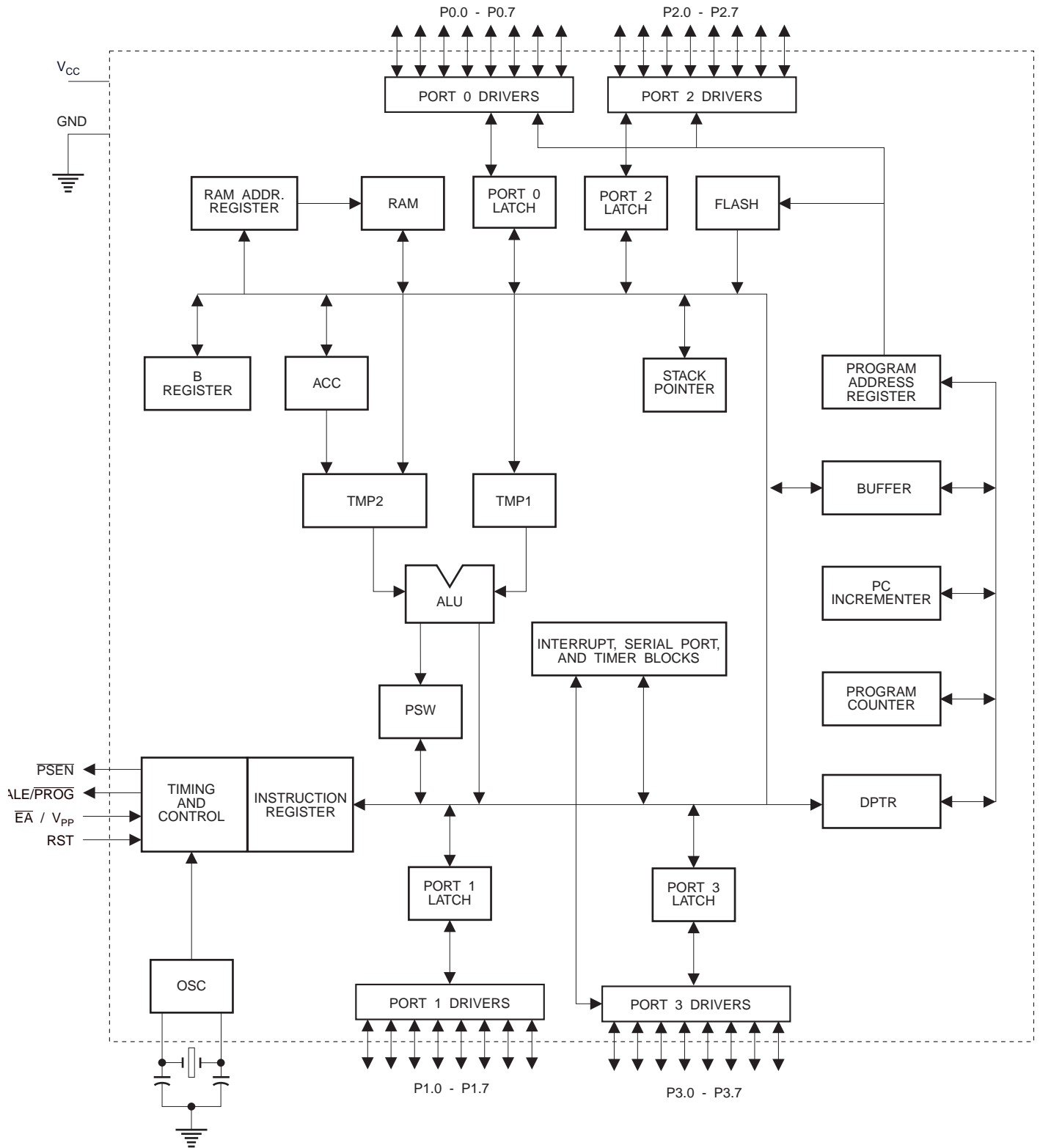
## 8-bit Microcontroller with 4K Bytes Flash

### AT89C51

**Not Recommended  
for New Designs.  
Use AT89S51.**



# Block Diagram



The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs,

Port 2 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

### Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

### ALE/ $\overline{\text{PROG}}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ( $\overline{\text{PROG}}$ ) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

### $\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory,  $\overline{\text{PSEN}}$  is activated twice each machine cycle, except that two  $\overline{\text{PSEN}}$  activations are skipped during each access to external data memory.

### $\overline{\text{EA}}/\text{VPP}$

External Access Enable.  $\overline{\text{EA}}$  must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed,  $\overline{\text{EA}}$  will be internally latched on reset.

$\overline{\text{EA}}$  should be strapped to  $V_{\text{CC}}$  for internal program executions.

This pin also receives the 12-volt programming enable voltage ( $V_{\text{PP}}$ ) during Flash programming, for parts that require 12-volt  $V_{\text{PP}}$ .

### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

### XTAL2

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left

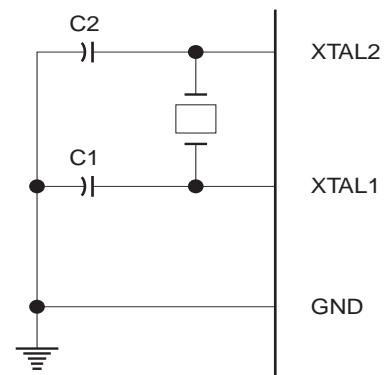
unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

## Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections

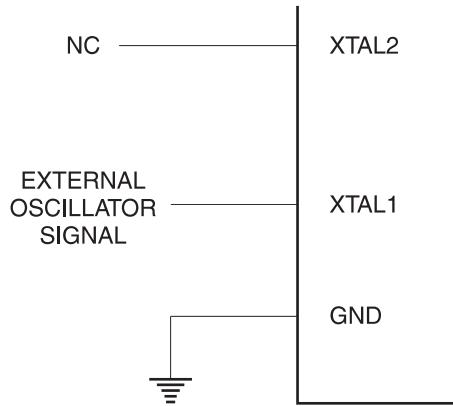


Note: C1, C2 = 30 pF ± 10 pF for Crystals  
= 40 pF ± 10 pF for Ceramic Resonators

## Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

**Figure 2.** External Clock Drive Configuration



## Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Regis-

ters retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before  $V_{CC}$  is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

## Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the  $\overline{EA}$  pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of  $\overline{EA}$  be in agreement with the current logic level at that pin in order for the device to function properly.

## Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

## Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage ( $V_{CC}$ ) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) = FFH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

**Programming Algorithm:** Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise  $\overline{EA}/V_{PP}$  to 12V for the high-voltage programming mode.
5. Pulse  $\overline{ALE}/\overline{PROG}$  once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address

and data for the entire array or until the end of the object file is reached.

**Data Polling:** The AT89C51 features  $\overline{Data}$  Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin.  $\overline{Data}$  Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming can also be monitored by the  $\overline{RDY}/\overline{BSY}$  output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:** The entire Flash array is erased electrically by using the proper combination of control signals and by holding  $\overline{ALE}/\overline{PROG}$  low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

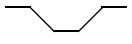
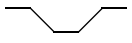
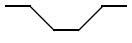
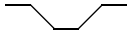
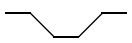
- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 51H indicates 89C51
- (032H) = FFH indicates 12V programming
- (032H) = 05H indicates 5V programming

## Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

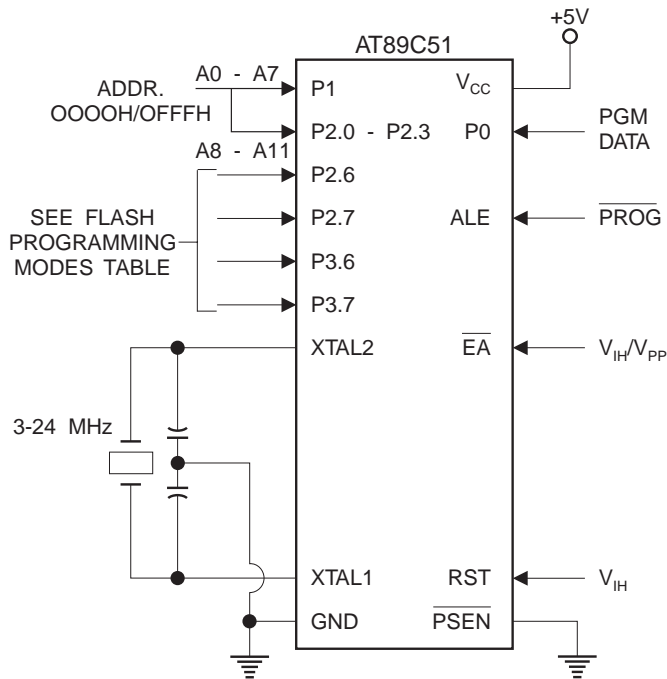
All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

## Flash Programming Modes

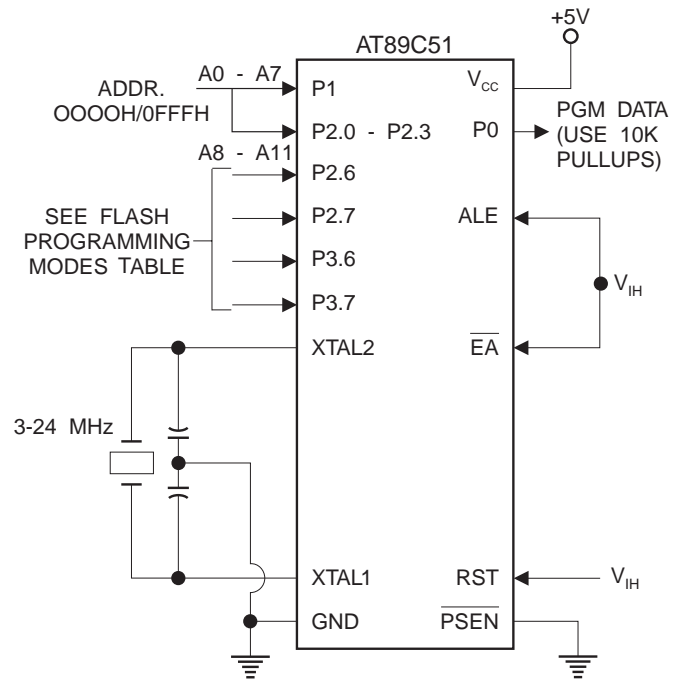
Mode	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	H	L		H/12V	H	H	H	H
								
								
Chip Erase	H	L	 (1)	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

**Figure 3.** Programming the Flash

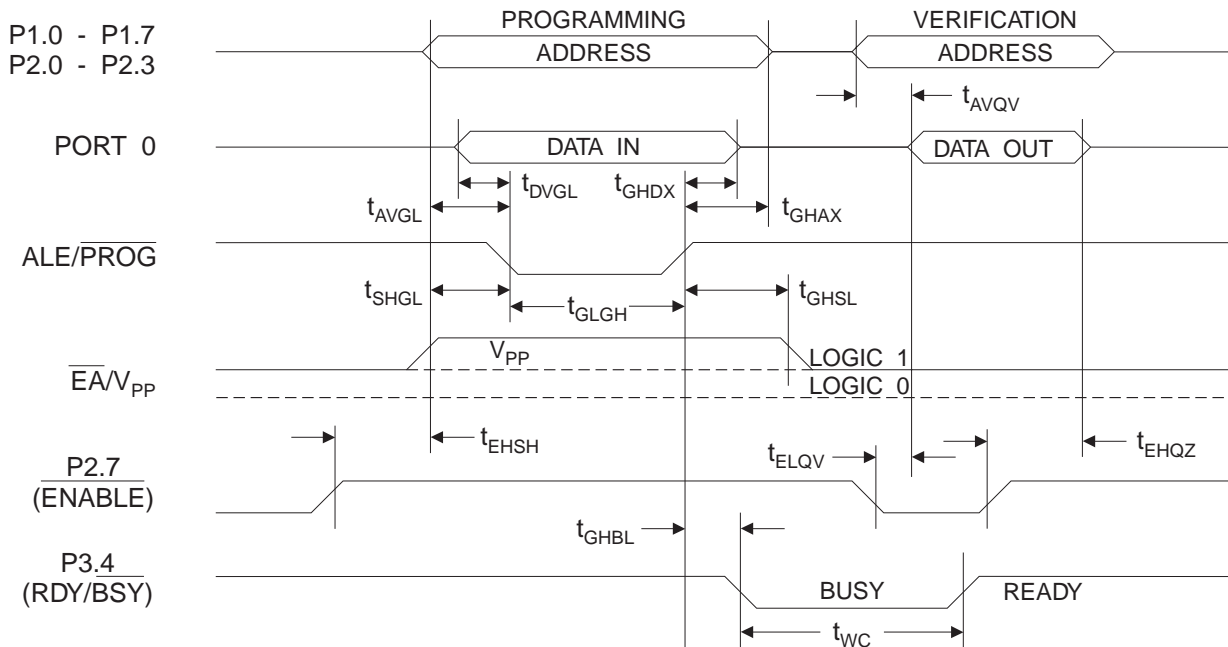


**Figure 4.** Verifying the Flash

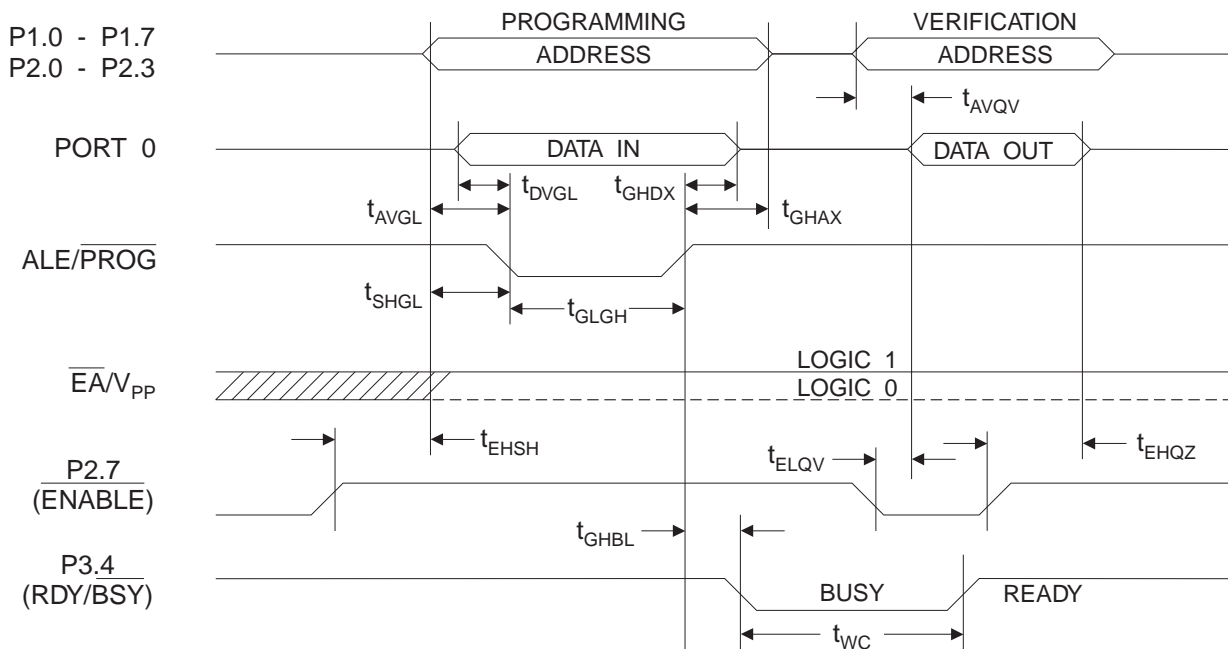




## Flash Programming and Verification Waveforms - High-voltage Mode ( $V_{PP} = 12V$ )



## Flash Programming and Verification Waveforms - Low-voltage Mode ( $V_{PP} = 5V$ )



## Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
$t_{AVGL}$	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
$t_{GHAX}$	Address Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{DVGL}$	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
$t_{GHDX}$	Data Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{EHSB}$	P2.7 ( $\overline{\text{ENABLE}}$ ) High to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ Setup to $\overline{\text{PROG}}$ Low	10		$\mu\text{s}$
$t_{GHSL}^{(1)}$	$V_{PP}$ Hold after $\overline{\text{PROG}}$	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{\text{PROG}}$ Width	1	110	$\mu\text{s}$
$t_{AVQV}$	Address to Data Valid		$48t_{CLCL}$	
$t_{ELQV}$	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data Float after $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
$t_{GHBL}$	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	$\mu\text{s}$
$t_{WC}$	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

## Absolute Maximum Ratings\*

Operating Temperature.....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground .....	-1.0V to +7.0V
Maximum Operating Voltage .....	6.6V
DC Output Current.....	15.0 mA

\*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 20\%$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low-voltage	(Except $\overline{EA}$ )	-0.5	$0.2 V_{CC} - 0.1$	V
$V_{IL1}$	Input Low-voltage ( $\overline{EA}$ )		-0.5	$0.2 V_{CC} - 0.3$	V
$V_{IH}$	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output Low-voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
$V_{OL1}$	Output Low-voltage <sup>(1)</sup> (Port 0, ALE, $\overline{PSEN}$ )	$I_{OL} = 3.2 \text{ mA}$		0.45	V
$V_{OH}$	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$ )	$I_{OH} = -60 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
$V_{OH1}$	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
$I_{IL}$	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port 0, $\overline{EA}$ )	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RRST	Reset Pull-down Resistor		50	300	$\text{K}\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$		40	$\mu\text{A}$

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 10 mA  
 Maximum  $I_{OL}$  per 8-bit port: Port 0: 26 mA  
 Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power-down is 2V.

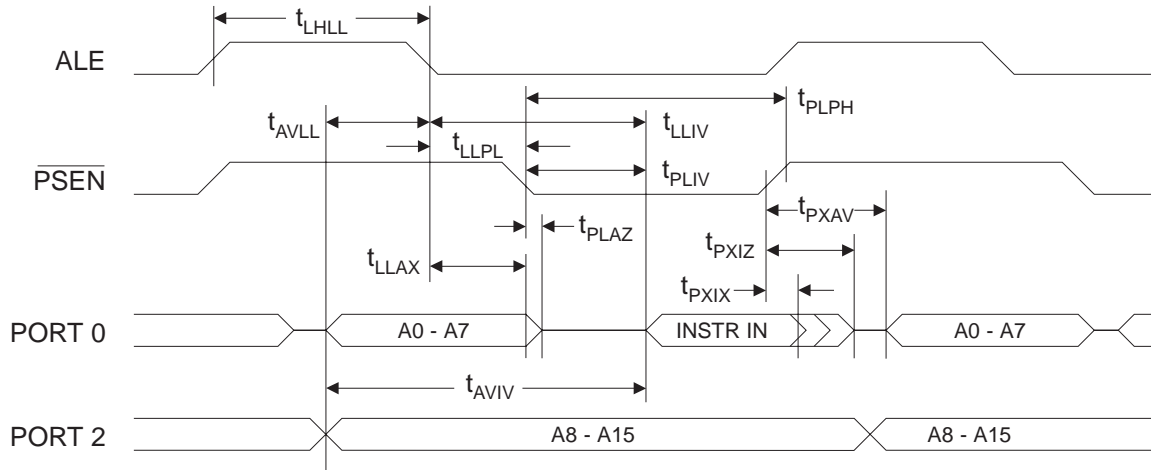
## AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$ , and  $\overline{\text{PSEN}}$  = 100 pF; load capacitance for all other outputs = 80 pF.

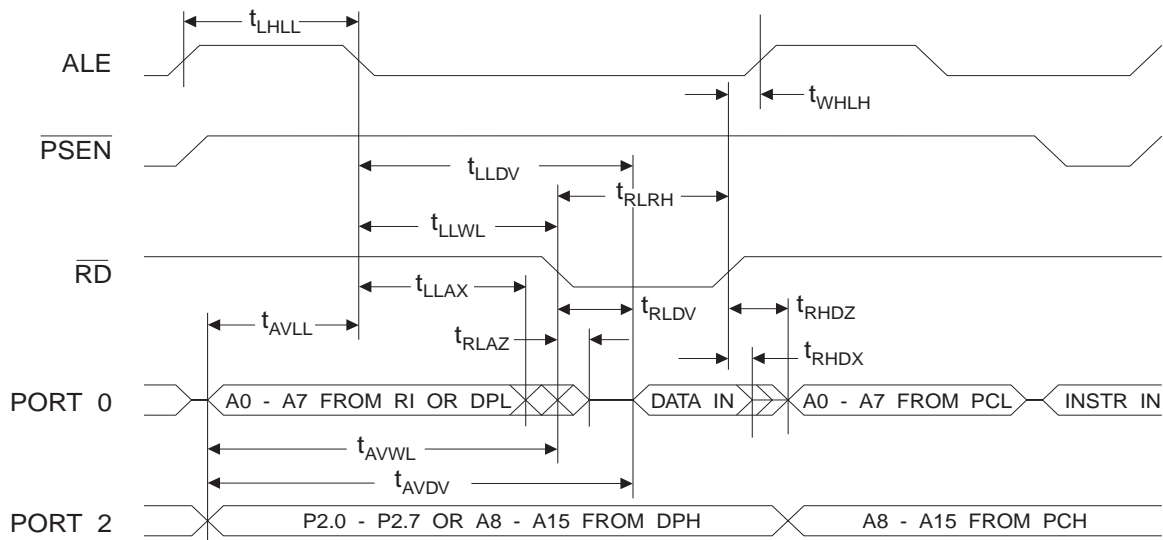
## External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency			0	24	MHz
$t_{\text{LHLL}}$	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
$t_{\text{AVLL}}$	Address Valid to ALE Low	43		$t_{\text{CLCL}}-13$		ns
$t_{\text{LLAX}}$	Address Hold after ALE Low	48		$t_{\text{CLCL}}-20$		ns
$t_{\text{LLIV}}$	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
$t_{\text{LLPL}}$	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-13$		ns
$t_{\text{PLPH}}$	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-20$		ns
$t_{\text{PLIV}}$	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-45$	ns
$t_{\text{PXIX}}$	Input Instruction Hold after $\overline{\text{PSEN}}$	0		0		ns
$t_{\text{PXIZ}}$	Input Instruction Float after $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-10$	ns
$t_{\text{PXAV}}$	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
$t_{\text{AVIV}}$	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-55$	ns
$t_{\text{PLAZ}}$	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
$t_{\text{RLRH}}$	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
$t_{\text{WLWH}}$	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
$t_{\text{RLDV}}$	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
$t_{\text{RHDX}}$	Data Hold after $\overline{\text{RD}}$	0		0		ns
$t_{\text{RHDZ}}$	Data Float after $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
$t_{\text{LLDV}}$	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
$t_{\text{AVDV}}$	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
$t_{\text{LLWL}}$	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
$t_{\text{AVWL}}$	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
$t_{\text{QVWX}}$	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-20$		ns
$t_{\text{QVWH}}$	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-120$		ns
$t_{\text{WHQX}}$	Data Hold after $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-20$		ns
$t_{\text{RLAZ}}$	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
$t_{\text{WHLH}}$	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-20$	$t_{\text{CLCL}}+25$	ns

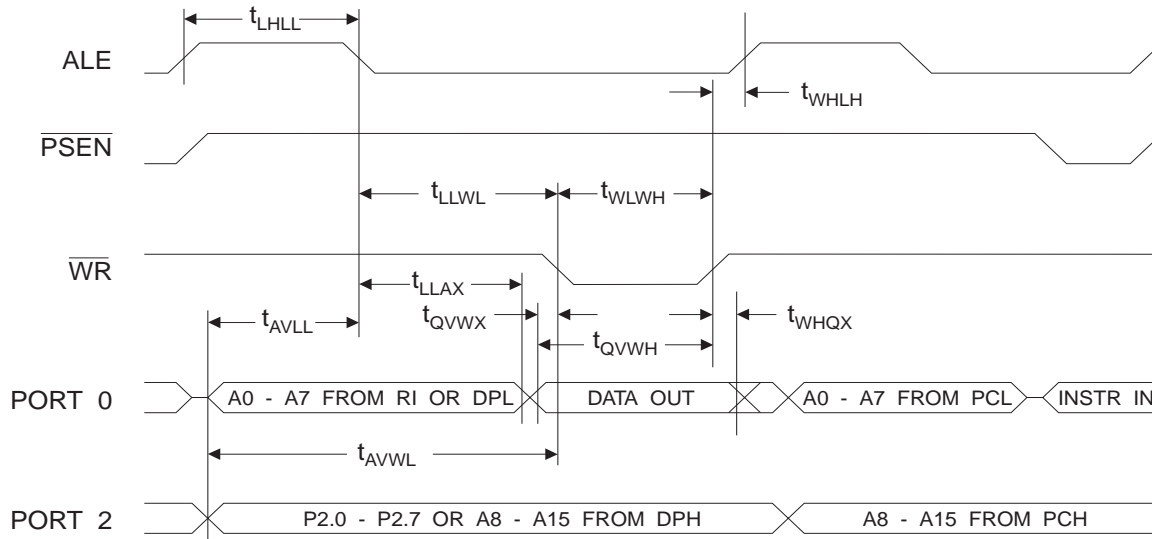
## External Program Memory Read Cycle



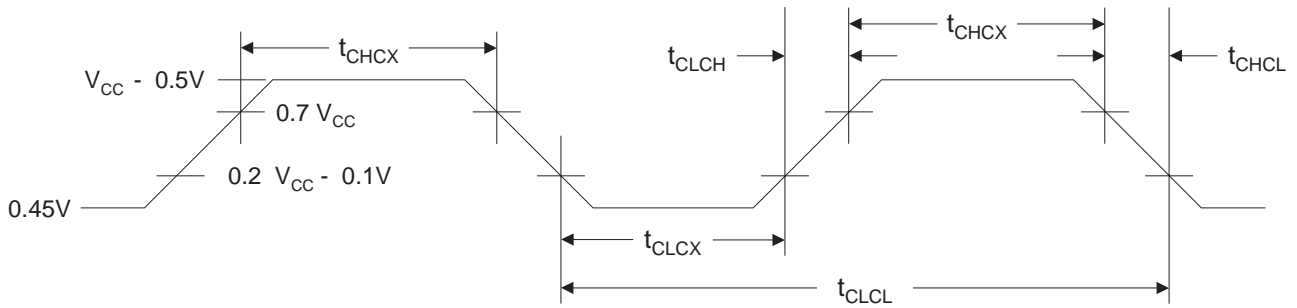
## External Data Memory Read Cycle



### External Data Memory Write Cycle



### External Clock Drive Waveforms



### External Clock Drive

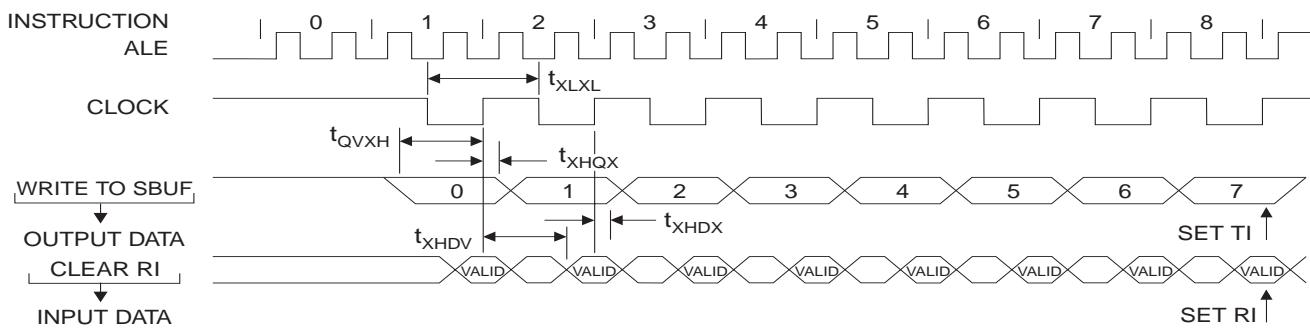
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
$t_{CLCL}$	Clock Period	41.6		ns
$t_{CHCX}$	High Time	15		ns
$t_{CLCX}$	Low Time	15		ns
$t_{CLCH}$	Rise Time		20	ns
$t_{CHCL}$	Fall Time		20	ns

## Serial Port Timing: Shift Register Mode Test Conditions

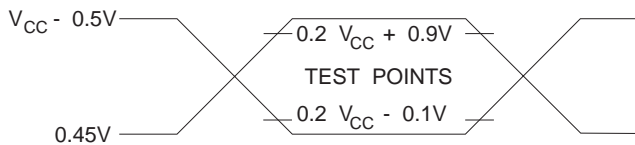
( $V_{CC} = 5.0\text{ V} \pm 20\%$ ; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
$t_{XLXL}$	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
$t_{XHQX}$	Output Data Hold after Clock Rising Edge	50		$2t_{CLCL}-117$		ns
$t_{XHDX}$	Input Data Hold after Clock Rising Edge	0		0		ns
$t_{XHDV}$	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

## Shift Register Mode Timing Waveforms

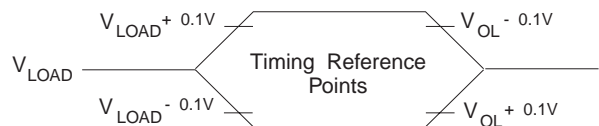


## AC Testing Input/Output Waveforms<sup>(1)</sup>



Note: 1. AC Inputs during testing are driven at  $V_{CC} - 0.5\text{V}$  for a logic 1 and  $0.45\text{V}$  for a logic 0. Timing measurements are made at  $V_{IH}$  min. for a logic 1 and  $V_{IL}$  max. for a logic 0.

## Float Waveforms<sup>(1)</sup>



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded  $V_{OH}/V_{OL}$  level occurs.

## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range	
12	5V ±20%	AT89C51-12AC	44A	Commercial (0° C to 70° C)	
		AT89C51-12JC	44J		
		AT89C51-12PC	40P6		
		AT89C51-12QC	44Q		
			AT89C51-12AI	44A	Industrial (-40° C to 85° C)
			AT89C51-12JI	44J	
			AT89C51-12PI	40P6	
			AT89C51-12QI	44Q	
16	5V ±20%	AT89C51-16AC	44A	Commercial (0° C to 70° C)	
		AT89C51-16JC	44J		
		AT89C51-16PC	40P6		
		AT89C51-16QC	44Q		
			AT89C51-16AI	44A	Industrial (-40° C to 85° C)
			AT89C51-16JI	44J	
			AT89C51-16PI	40P6	
			AT89C51-16QI	44Q	
20	5V ±20%	AT89C51-20AC	44A	Commercial (0° C to 70° C)	
		AT89C51-20JC	44J		
		AT89C51-20PC	40P6		
		AT89C51-20QC	44Q		
			AT89C51-20AI	44A	Industrial (-40° C to 85° C)
			AT89C51-20JI	44J	
			AT89C51-20PI	40P6	
			AT89C51-20QI	44Q	
24	5V ±20%	AT89C51-24AC	44A	Commercial (0° C to 70° C)	
		AT89C51-24JC	44J		
		AT89C51-24PC	40P6		
		AT89C51-24QC	44Q		
			AT89C51-24AI	44A	Industrial (-40° C to 85° C)
			AT89C51-24JI	44J	
			AT89C51-24PI	40P6	
			AT89C51-24QI	44Q	

Package Type	
<b>44A</b>	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
<b>44J</b>	44-lead, Plastic J-leaded Chip Carrier (PLCC)
<b>40P6</b>	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
<b>44Q</b>	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)







## Atmel Headquarters

### *Corporate Headquarters*

2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### *Europe*

Atmel U.K., Ltd.  
Coliseum Business Centre  
Riverside Way  
Camberley, Surrey GU15 3YL  
England  
TEL (44) 1276-686-677  
FAX (44) 1276-686-697

### *Asia*

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### *Japan*

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Operations

### *Atmel Colorado Springs*

1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

### *Atmel Rousset*

Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

---

### *Fax-on-Demand*

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

### *e-mail*

[literature@atmel.com](mailto:literature@atmel.com)

### *Web Site*

<http://www.atmel.com>

### *BBS*

1-(408) 436-4309

#### © Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

0265G-02/00/xM

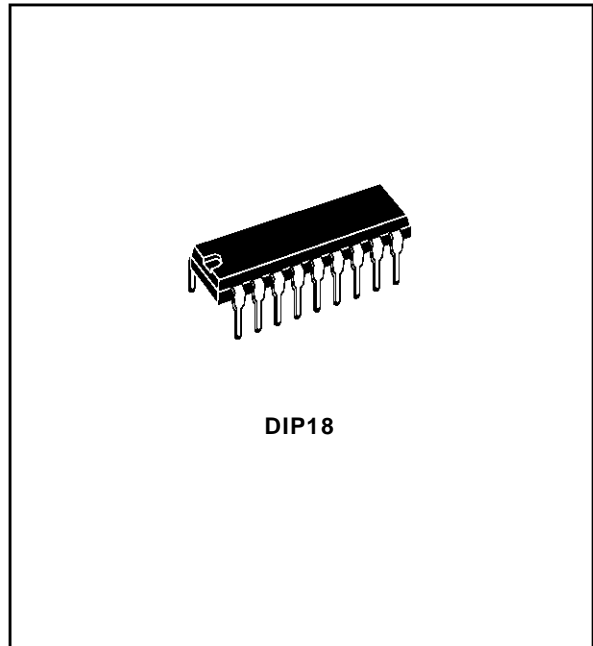
This datasheet has been download from:

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

Datasheets for electronics components.

## EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



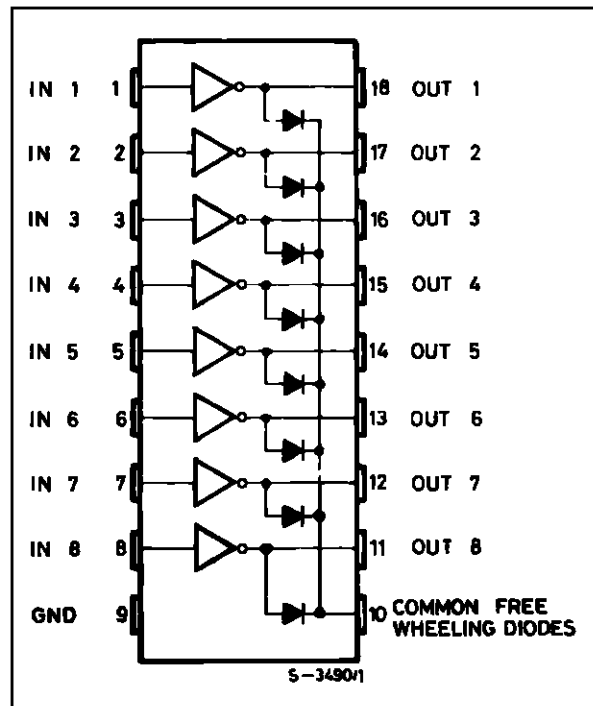
### DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

Five versions are available to simplify interfacing to standard logic families: the ULN2801A is designed for general purpose applications with a current limit resistor; the ULN2802A has a 10.5k $\Omega$  input resistor and zener for 14-25V PMOS; the ULN2803A has a 2.7k $\Omega$  input resistor for 5V TTL and CMOS; the ULN2804A has a 10.5k $\Omega$  input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

All types are supplied in a 18-lead plastic DIP with a copperlead from and feature the convenient input-opposite-output pinout to simplify board layout.

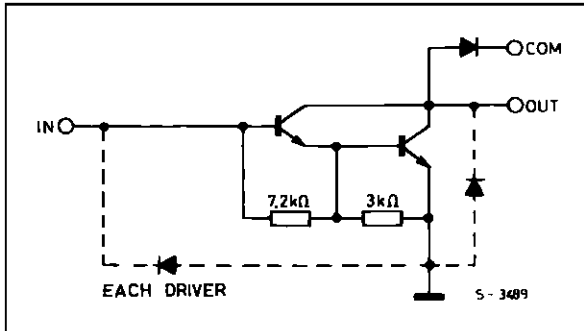
### PIN CONNECTION (top view)



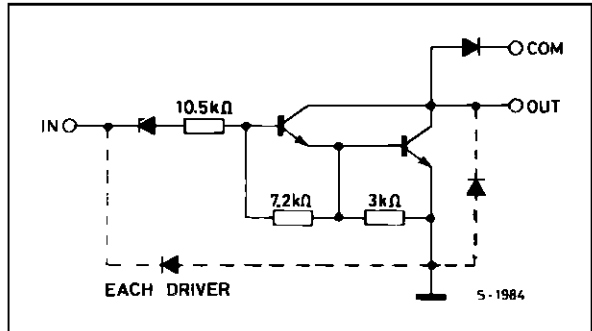
# ULN2801A - ULN2802A - ULN2803A - ULN2804A - ULN2805A

## SCHEMATIC DIAGRAM AND ORDER CODES

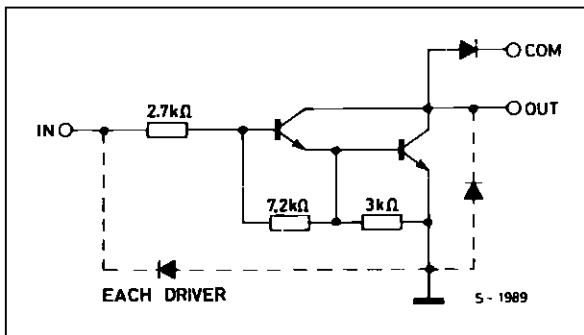
For ULN2801A (each driver for PMOS-CMOS)



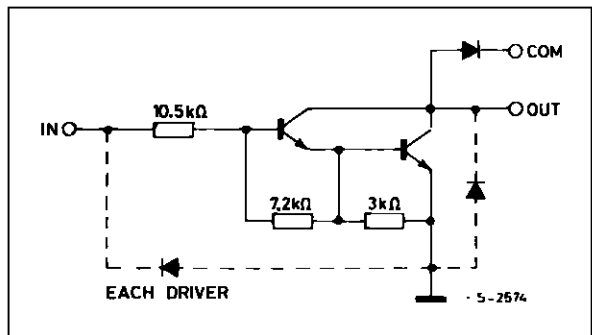
For ULN2802A (each driver for 14-15 V PMOS)



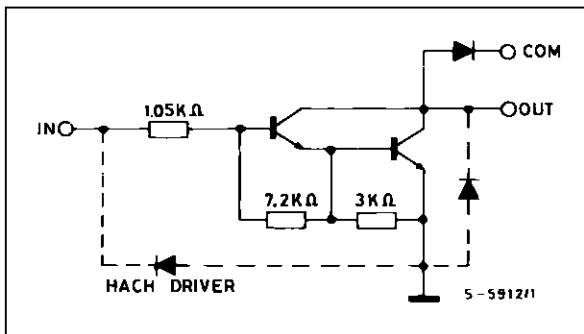
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS)



For ULN2805A (each driver for high out TTL)



## ULN2801A - ULN2802A - ULN2803A - ULN2804A - ULN2805A

### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_o$	Output Voltage	50	V
$V_i$	Input Voltage for ULN2802A, UL2803A, ULN2804A for ULN2805A	30 15	V
$I_C$	Continuous Collector Current	500	mA
$I_B$	Continuous Base Current	25	mA
$P_{tot}$	Power Dissipation (one Darlington pair) (total package)	1.0 2.25	W
$T_{amb}$	Operating Ambient Temperature Range	- 20 to 85	°C
$T_{stg}$	Storage Temperature Range	- 55 to 150	°C
$T_j$	Junction Temperature Range	- 20 to 150	°C

### THERMAL DATA

Symbol	Parameter	Value	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max. 55	°C/W

### ELECTRICAL CHARACTERISTICS ( $T_{amb} = 25^{\circ}C$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
$I_{CEX}$	Output Leakage Current	$V_{CE} = 50V$ $T_{amb} = 70^{\circ}C, V_{CE} = 50V$ $T_{amb} = 70^{\circ}C$ for ULN2802A $V_{CE} = 50V, V_i = 6V$ for ULN2804A $V_{CE} = 50V, V_i = 1V$			50 100 500 500	$\mu A$ $\mu A$ $\mu A$ $\mu A$	1a 1a 1b 1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_C = 100mA, I_B = 250\mu A$ $I_C = 200mA, I_B = 350\mu A$ $I_C = 350mA, I_B = 500\mu A$		0.9 1.1 1.3	1.1 1.3 1.6	V V V	2
$I_{i(on)}$	Input Current	for ULN2802A $V_i = 17V$ for ULN2803A $V_i = 3.85V$ for ULN2804A $V_i = 5V$ for ULN2805A $V_i = 12V$ $V_i = 3V$		0.82 0.93 0.35 1 1.5	1.25 1.35 0.5 1.45 2.4	mA mA mA mA mA	3
$I_{i(off)}$	Input Current	$T_{amb} = 70^{\circ}C, I_C = 500\mu A$	50	65		$\mu A$	4
$V_{i(on)}$	Input Voltage	$V_{CE} = 2V$ for ULN2802A $I_C = 300mA$ for ULN2803A $I_C = 200mA$ $I_C = 250mA$ $I_C = 300mA$ for ULN2804A $I_C = 125mA$ $I_C = 200mA$ $I_C = 275mA$ $I_C = 350mA$ for ULN2805A $I_C = 350mA$			13 2.4 2.7 3 5 6 7 8 2.4	V V V V V V V V V	5
$h_{FE}$	DC Forward Current Gain	for ULN2801A $V_{CE} = 2V, I_C = 350mA$	1000			-	2
$C_i$	Input Capacitance			15	25	pF	-
$t_{PLH}$	Turn-on Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	$\mu s$	-
$t_{PHL}$	Turn-off Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	$\mu s$	-
$I_R$	Clamp Diode Leakage Current	$V_R = 50V$ $T_{amb} = 70^{\circ}C, V_R = 50V$			50 100	$\mu A$ $\mu A$	6 6
$V_F$	Clamp Diode Forward Voltage	$I_F = 350mA$		1.7	2	V	7

TEST CIRCUITS

Figure 1a.

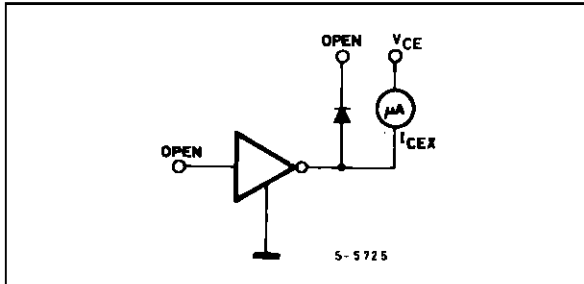


Figure 1b.

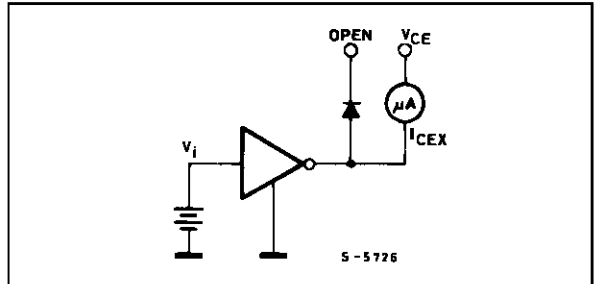


Figure 2.

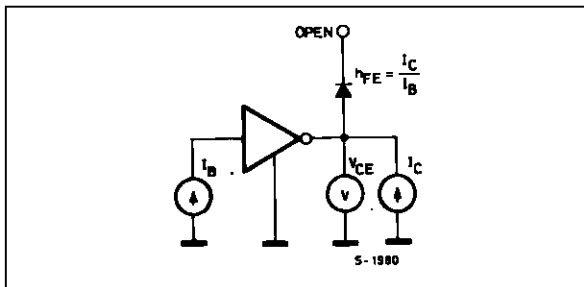


Figure 3.

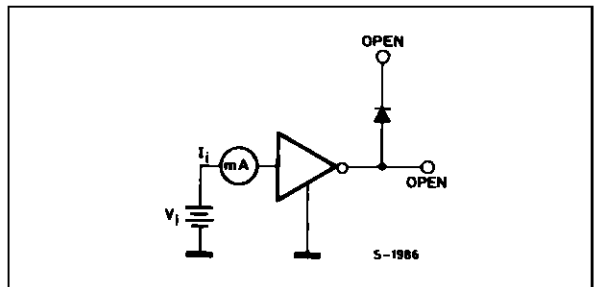


Figure 4.

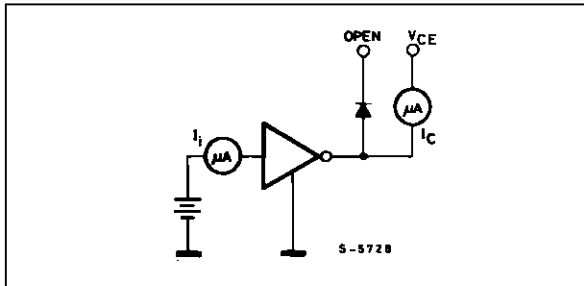


Figure 5.

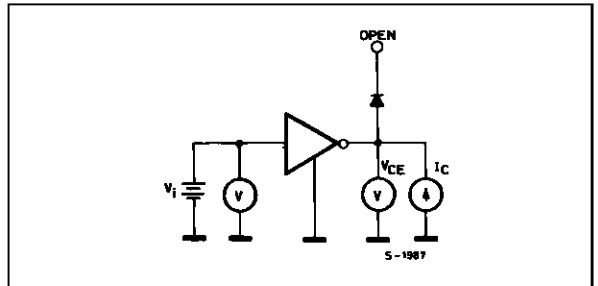


Figure 6.

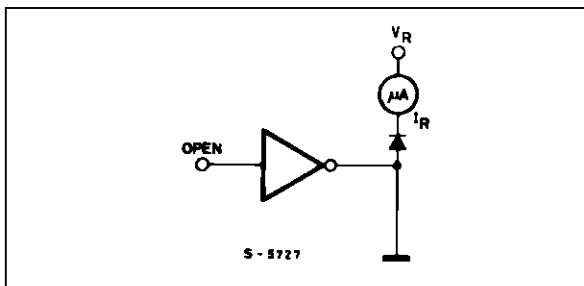
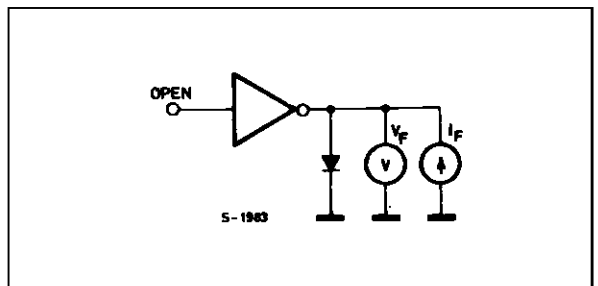
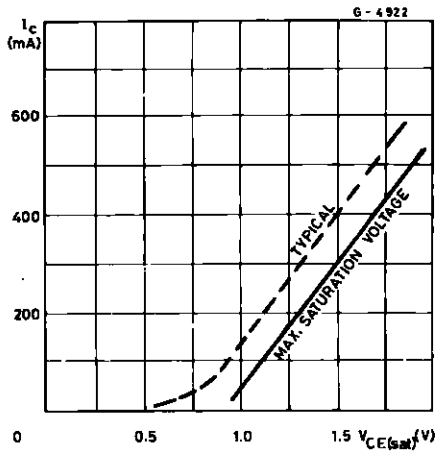


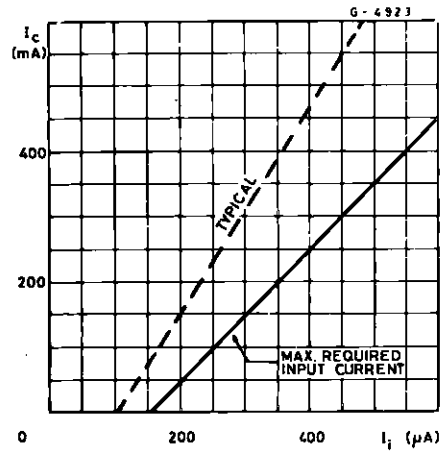
Figure 7.



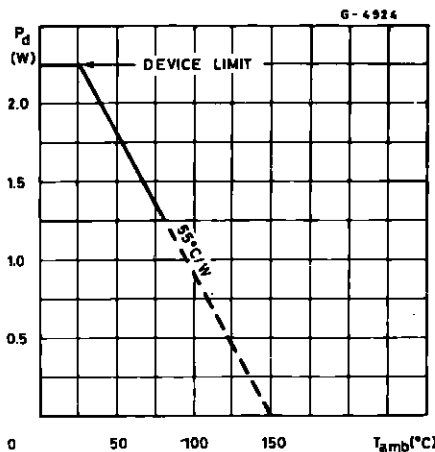
**Figure 8 :** Collector Current as a Function of Saturation Voltage.



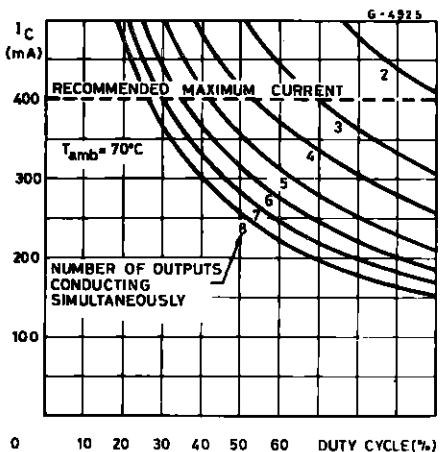
**Figure 9 :** Collector Current as a Function of Input Current.



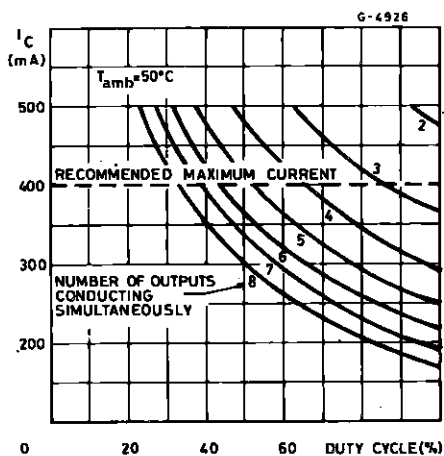
**Figure 10 :** Allowable Average Power Dissipation as a Function of Ambient Temperature.



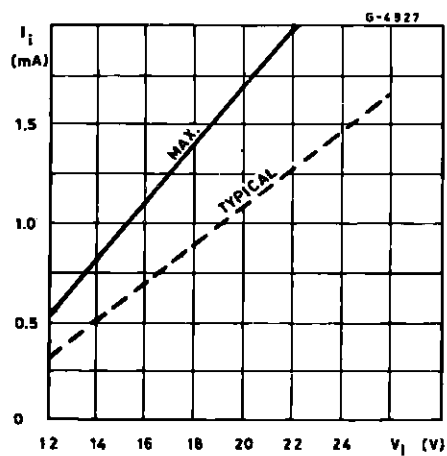
**Figure 11 :** Peak Collector Current as a Function of Duty Cycle.



**Figure 12 :** Peak Collector Current as a Function of Duty.

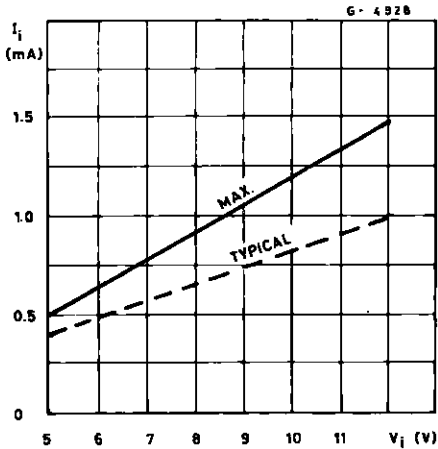


**Figure 13 :** Input Current as a Function of Input Voltage (for ULN2802A).

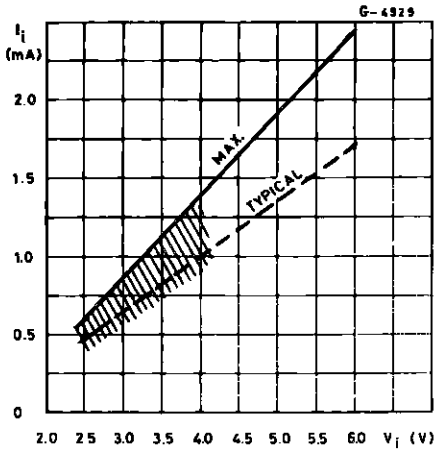




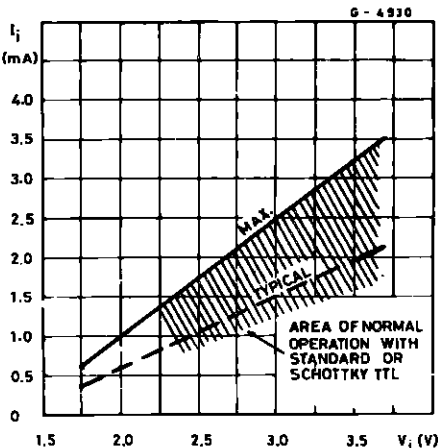
**Figure 14 :** Input Current as a Function of Input Voltage (for ULN2804A)



**Figure 15 :** Input Current as a Function of Input Voltage (for ULN2803A)

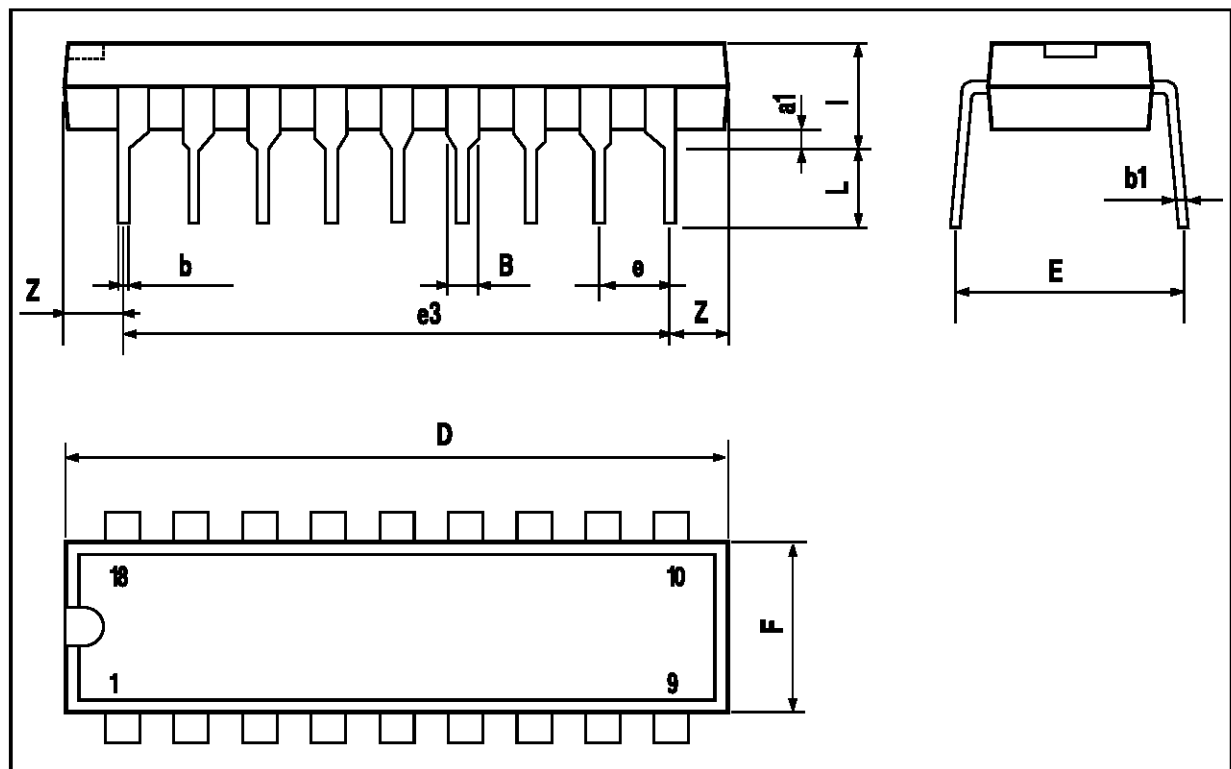


**Figure 16 :** Input Current as a Function of Input Voltage (for ULN2805A)



DIP18 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.254			0.010		
B	1.39		1.65	0.055		0.065
b		0.46			0.018	
b1		0.25			0.010	
D			23.24			0.915
E		8.5			0.335	
e		2.54			0.100	
e3		20.32			0.800	
F			7.1			0.280
l			3.93			0.155
L		3.3			0.130	
Z		1.27	1.59		0.050	0.063



Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1997 SGS-THOMSON Microelectronics – Printed in Italy – All Rights Reserved

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco -  
The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

This datasheet has been download from:

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

Datasheets for electronics components.