Lecture 1

Computers, Computer Interfaces and Computer Programming

The Digital Computer

A computer is an automatic electronic machine that performs calculations according to a stored program. As shown in figure 1, a computer has four basic components:

(1) **The central processing unit (CPU).** This device performs the arithmetic and logical operations according to the operator's instructions.

(2) **The memory**. This component serves to store the sequence of instructions (program) and the data.

(3) **The system bus**. This is the communication pathway shared by all of the computer's components.

(4) **The input/output (I/O) devices.** These are facilities for accepting data from the outside world and providing results. The data acquisition card is a highly specialized I/O device.

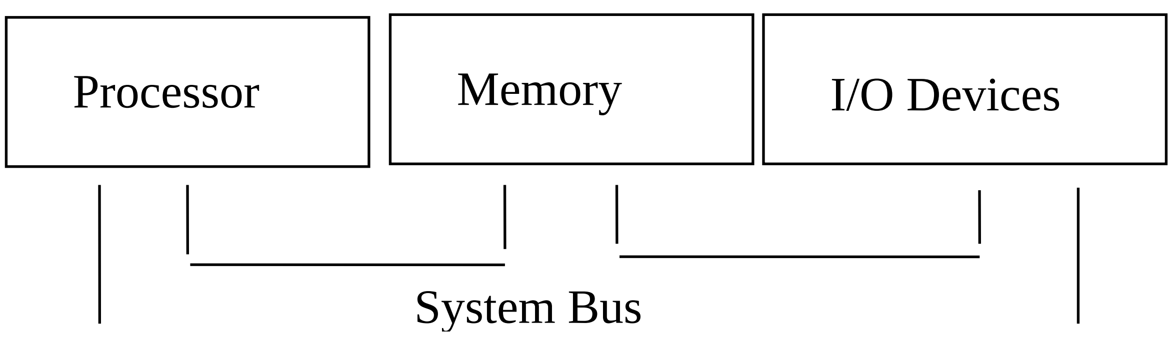


Figure 1-1. Components of a Computer

Digital computers represent numbers in binary (base 2) format. Thus there are only two numbers, 1 and 0. In the digital computer, these are represented as voltages. Typically, voltages less than 0.8 volts are zero and voltages greater than 2.00 volts are one. These two voltages or states can have an alternate interpretation when used for logical operations: they can represent the logical propositions of true (1) and false (0). An algebra of two state logic was invented by George Boule over 100 years ago. It provides the means whereby the basic operations: *and*, *or* and *negation* can be used to construct highly complex logical and arithmetic operations. In 1938, Shannon showed how Boolean algebra could be implemented with electronic switching devices. Shortly thereafter John von Neumann proposed that the sequence of instructions to be carried out by the computer be stored in memory. The instructions would be carried out by a CPU which would fetch the instructions from memory. Thus, the basis for modern digital computers was conceived.

How do we represent numbers other than 1 and 0 in the computer? The answer is by positional encoding. For the binary system, this can be illustrated by means of an example:

$$110101 = 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 53_{10}$$

where the subscript 10 means the number is written in the decimal number system. Note that the 2^3 and 2^1 are replaced by zeros because the positions for these powers contain a zero in the binary representation of the number.

The basic word length of the digital computer is a series of 8 bits called a byte. An 8 bit binary word has a resolution of $2^8 = 256_{10}$. In modern computers, eight bytes are assembled into a 32 bit word. This entity has a resolution of $2^{32} > 4{,}000{,}000{,}000_{10}$.

The contemporary CPU, such as found in the Pentium 4 microprocessor, is the heart of the computer. It is an enormously complicated device consisting of millions of transistors, all on a single silicon chip. CPUs are distinguished by the number and types of arithmetic and logic operations they can perform. A complex instruction set CPU (CISC) like the Pentium 4 processor is able to carry out over 100 types of operations.

The memory is the short-term storage area, as distinct from the archival storage devices such as magnetic or optical disks. The memory stores the program that is presently active as sequence of instructions and also stores the data for that program. When the processor executes the program, it gets its instructions sequentially from the memory and moves data in and out of memory as necessary. Logically, the memory is configured as a set of storage cells, each having a unique 32-bit address as illustrated in Figure 2. The processor may read and write to memory locations in any order. Thus, the term random access memory (RAM).

The system bus provides the means for the computer's components to communicate with each other by transferring data and instructions. The bus consists of three sub-buses: a 64 bit data bus, a 64 bit address bus and a set of control lines which provide instructions on the interpretation of the address, the direction of information flow etc.

Cell 0
Cell 1
Cell 2
Cell 3
. . .
Cell N-2
Cell N-1
Cell N

Figure 1-2 Memory Addresses

We are now in a position to see how the computer carries out its functions. The basic instructions of the CPU are very simple. From these, very complicated operations can be constructed by carrying out a sequence of instructions stored in memory and known as a program. For the sake of illustration, let us consider one particular instruction that might be contained in a program:

MOV AX, COUNTS

This command moves the contents of a memory location known as "COUNTS" to a register in the CPU known as "AX". In performing this instruction, the computer actually executes a sequence of actions, all controlled by the system clock. In the first step, the CPU loads the COUNTS address onto the address bus. At the same time the CPU sets the bus control lines to indicate that data is to be retrieved from memory. Then after a very short delay (nanoseconds), when the address has had time to reach the memory, the CPU sends out a control signal to load the data available in the selected memory location to the CPU over the data bus. After the data is loaded, all control signals, data and addresses are removed from the bus and a new instruction can be fetched and executed.

The simple computer consisting of a processor, a memory, and system bus can execute programs, but it can't do anything useful, because it has no way to communicate with the outside world. To do even the basic tasks of loading new programs and interacting with the user, the computer must have input/output devices. Normally, we are referring to keyboard, monitor, mouse, network interface and mass storage devices such as diskettes and hard drives.

Computer Interfaces for Data Acquisition and Control.

This course is devoted to the problem of computer-aided chemical experimentation. Thus, the need for specialized I/O devices that can acquire data from sensors and control experimental conditions. Fortunately, there exist computer data acquisition systems that can be purchased that allow ready use in chemical experiments. An excellent example is the PCI-MIO-16E-4 data acquisition (DAQ) board from National Instruments that we use in this class. A block diagram of this board is given in Figure 3 below. It plugs into the back plane of the computer. For experimental purposes, we supply a "break out" board that interfaces to the DAQ board and allows easy (screw-in) connection of all of the external lines. The DAQ board contains the following components:

Analog to digital converter (ADC). This device allows the continuously variable signals (voltages) of the analog world to be converted to the binary numbers of the computer's world. A typical use of the ADC is to digitize the output of a sensor such as a pH meter in a titration experiment or an electronic thermometer in a melting point determination. Since there are often many parameters to be measured in an experiment, the ADC is equipped with a "multiplexer" which allows the output of one of several sensors to be fed to the input of the ADC.

Digital to analog converter (DAC). This device allows digital numbers from the computer to be converted to analog signals (voltages). A typical use of the DAC is to control an experimental variable such as the potential in a voltammetry measurement or the amount of heat applied to a chemical compound in a melting point determination. There are two independent DACs in our board.

Digital timer/counter. This device is used as a frequency counter, i.e. to count the number of pulses occurring in a given time. We will use this for nuclear decay curve measurement. Also, individual quanta of light (photons) can be counted for detection of individual molecules. Finally, the device can be used to generate a sequence of pulses at preset intervals or to count the time between pulses.

Digital I/O lines. The board also contains eight digital I/O lines which can be configured for either digital input or output in four-bit blocks. Typical uses of this facility are for sensing switch settings and for driving stepper motors.

Special features of the National Instruments DAQ board:

1. Protection of all inputs and outputs against damage from inappropriate electrical connections.
2. On-board buffering of analog input and output data streams.
3. Built in calibration hardware.
4. Direct memory transfers of data.

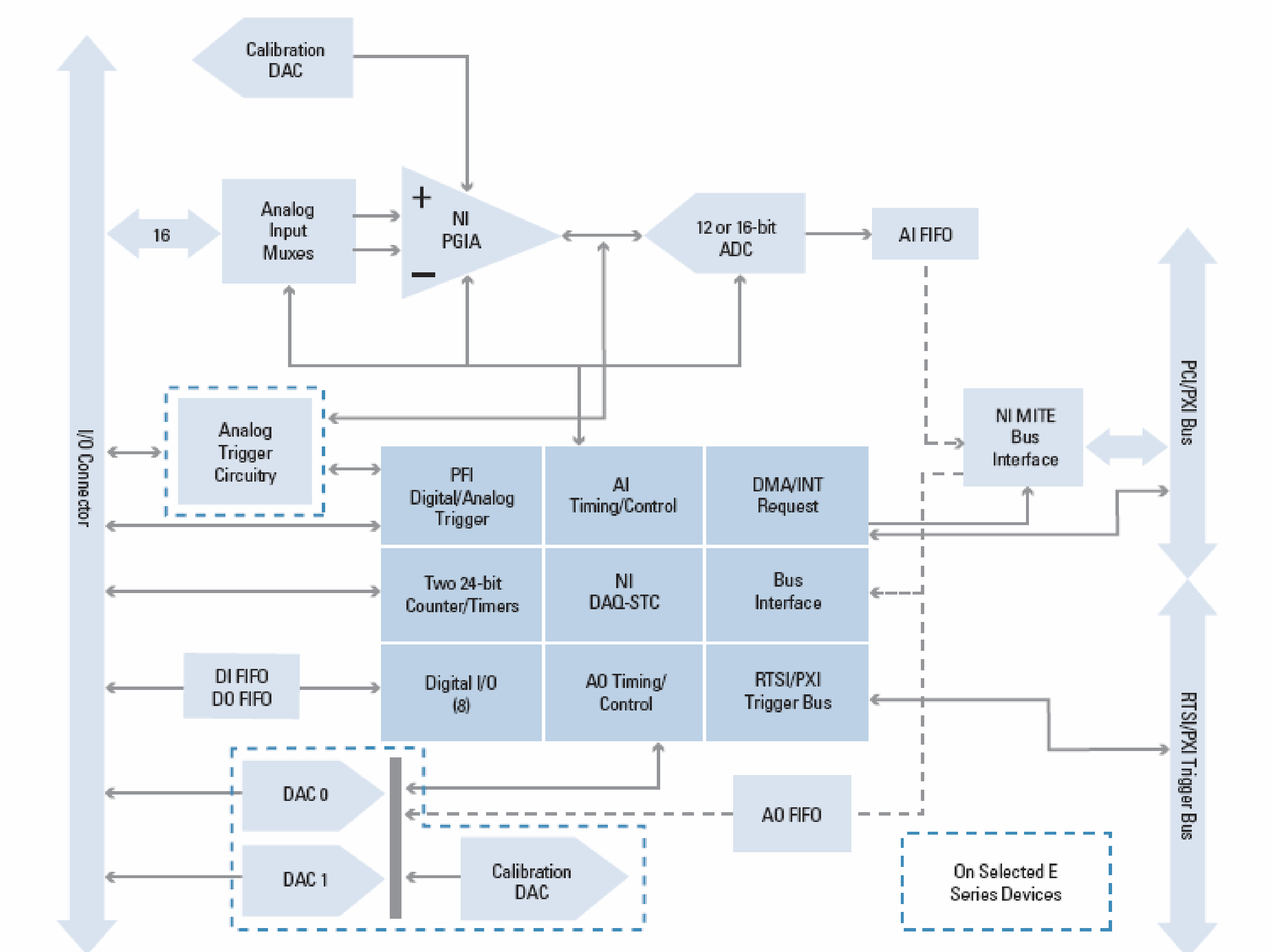


Figure 2. E Series Hardware Block Diagram

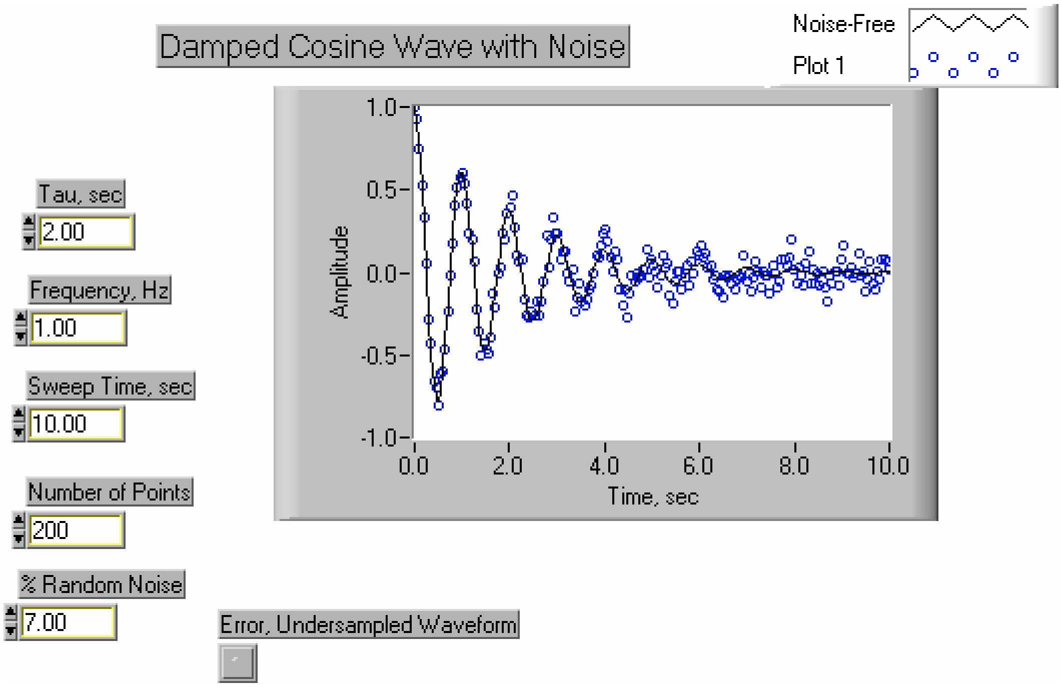
Computer Programming for Laboratory Data Acquisition and Control.

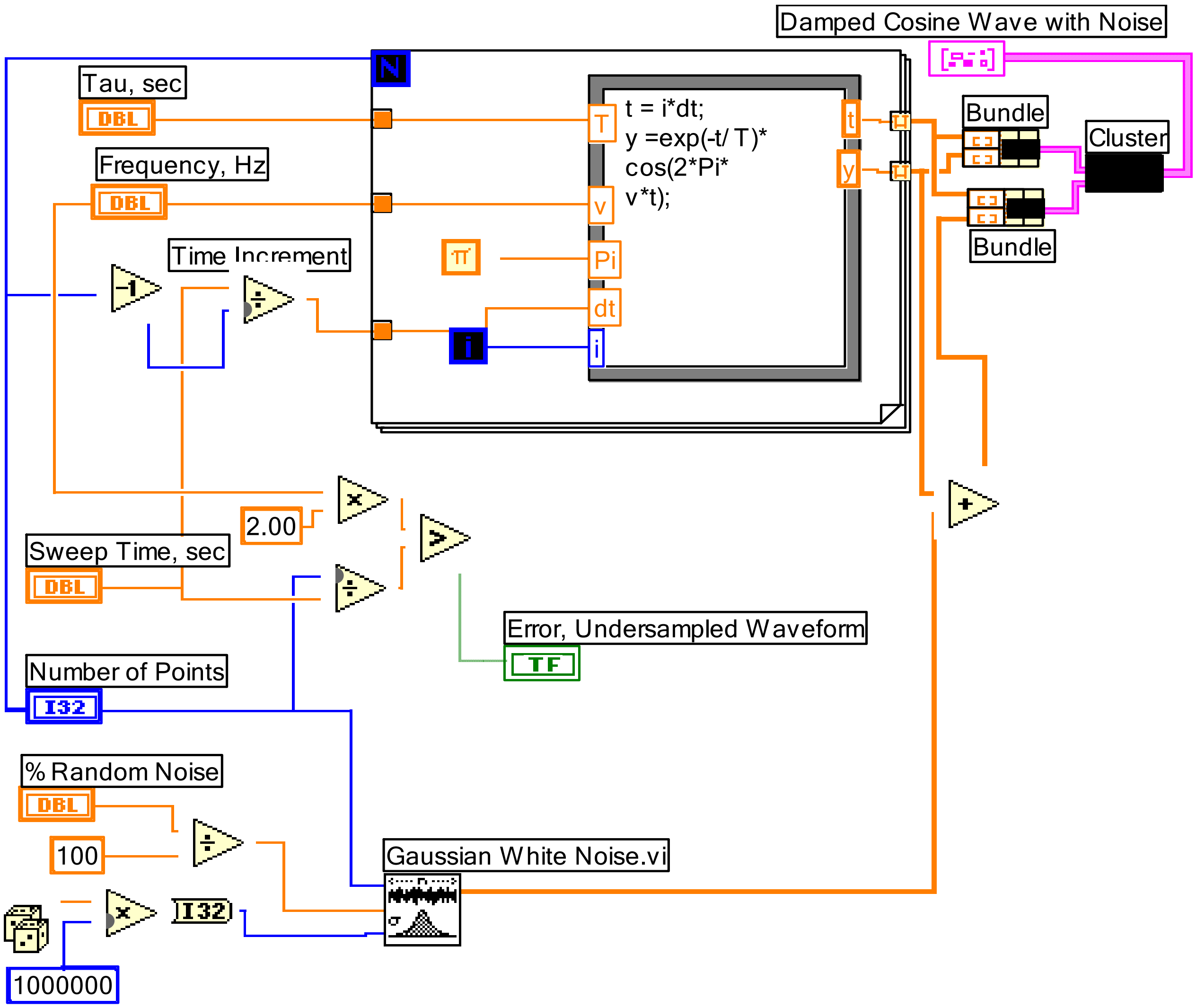
Ultimately, the computer requires the stored program and the data to be encoded in the binary representation and detailed step-by-step instructions in terms of the instruction set of the particular cpu being employed - such direct machine code programs are extremely difficult to compose, enter into memory and debug. As an alternative, the manufacturer of the computer supplies an assembler facility in which the binary code has been translated into mnemonic code and memory addresses can be given names. The example: MOV AX, COUNTS is in assembler code.

Assembler coding is seldom done by the end-user. Instead high-level languages such as C, Java or Visual Basic are used. In a high level language, the allocation to memory of variables, programs and data are performed automatically. The languages are based on key words and symbols such as print, do while, + and /. Programs based on these high level languages are composed with an editor and then are translated into appropriate machine code, either by a compiler before running or they are translated and run immediately.

We shall not be using the conventional text-based languages to create obscure lines of code which are difficult to create and easy to forget. Instead, we will be using the development environment LabVIEW to create laboratory data acquisition and control programs in a flow-chart program called a block diagram. LabVIEW uses terminology, icons and ideas familiar to scientists and engineers. It relies on graphic symbols rather than textual language to describe programming actions. You can learn LabVIEW even if you have little or no programming experience.

LabVIEW has extensive libraries of functions devoted to application specific code for data acquisition, control and analysis. Your computer, DAQ board and LabVIEW comprise a completely configurable virtual instrument to accomplish your task. You will find using LabVIEW to be truly easy and fun.


Damped Cosine Wave with Noise
Noise-Free
Plot 1
Tau, sec
2.00
Frequency, Hz
1.00
Sweep Time, sec
10.00
Number of Points
200
% Random Noise
7.00
Error, Undersampled Waveform
Amplitude
1.0
0.5
0.0
-0.5
-1.0
0.0
2.0
4.0
6.0
8.0
10.0
Time, sec


Damped Cosine Wave with Noise
Tau, sec
DBL
Frequency, Hz
DBL
Time Increment
t = i*dt;
y =exp(-t/ T)*
cos(2*Pi*
v*t);
T
v
Pi
dt
i
t
y
Bundle
Cluster
Bundle
Sweep Time, sec
DBL
2.00
Error, Undersampled Waveform
TF
Number of Points
I32
% Random Noise
DBL
100
I32
1000000
Gaussian White Noise.vi