CHAPTER 10

Intelligent Devices

Chapter Outline

10.1 Introduction

We now find reference to devices with names like intelligent instruments, smart sensors, and smart transmitters whenever we open a technical magazine or browse through an

Measurement and Instrumentation. http://dx.doi.org/10.1016/B978-0-12-800884-3.00010-1

instrument manufacturer's catalog. This reflects the fact that intelligent devices have now achieved widespread use in measurement applications. The term "intelligent" is used to denote any measurement device that uses computational power to enhance its measurement performance.

We are probably aware that digital computers have been used in conjunction with measurement systems for many years in the typical control system scenario where a computer uses data on process variables supplied by a measurement system to compute a control signal that is then applied to an actuator in order to modify some aspect of the controlled process. In this case, the computer was not actually part of the measurement system but merely works with it by taking data from the system.

As the cost of computers fell and their power increased, it became a common practice to use the computer assigned to a process control function to enhance the quality of measurements by performing various signal processing operations digitally that were previously carried out by analog electronic circuits. However, in these early applications of digital signal processing, the computer remained as a distinctly separate component within the measurement system.

We have now moved on one stage further to the point where the computer that performs digital signal processing to enhance measurement quality is incorporated into the measurement device. Such devices which incorporate digital signal processing are given the generic name *intelligent devices*. Individual intelligent devices attract various names like *intelligent instrument*, *intelligent sensor*, *smart sensor*, and *smart transmitter*. There are no hard distinctions between the function of any of these, and which term is used to refer to an intelligent device is largely due to the preference adopted by different manufacturers for one name or another. Similar variation exists in the name used to describe the computational power within the intelligent device, with terms like microcomputer and microprocessor being common.

The subject of this chapter is therefore intelligent devices. However, to start off with, we will look more generally at some basic principles of digital computation, since this will enable us to better understand how intelligent devices function and what potential difficulties exist in their application and operation.

10.2 Principles of Digital Computation

10.2.1 Elements of a Computer

The primary function of a digital computer is the manipulation of data. The three elements that are essential for the fulfillment of this task are the central processing unit (CPU), the memory, and the input−output (I/O) interface, as shown in Figure 10.1. These elements are collectively known as the computer hardware, and each element exists physically as one or

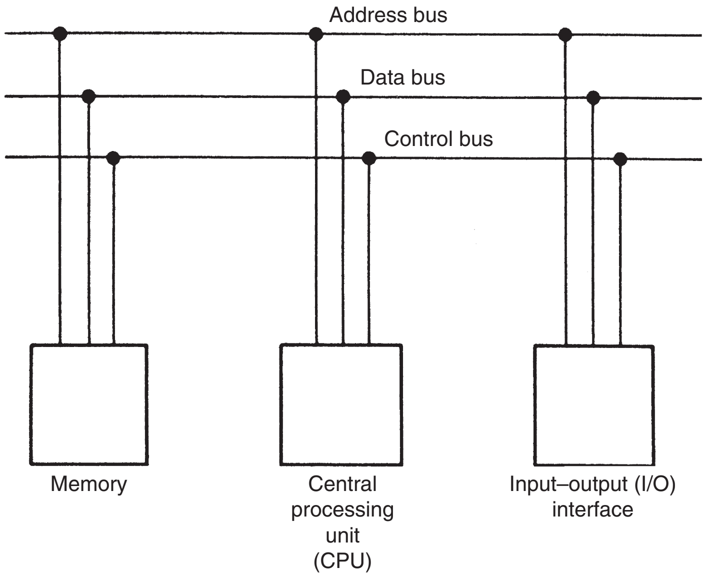

Figure 10.1
Elements of a microcomputer.

more integrated circuit chips mounted on a printed circuit board. Where the CPU consists of a single microprocessor, it is usual to regard the system as a microcomputer. The distinction between the terms "microcomputer," "minicomputer," and "mainframe-computer" is a very arbitrary division made according to relative computer power. However, this classification has become somewhat meaningless, with present day "microcomputers" being more powerful than mainframe computers of only a few years ago.

The CPU part of a computer can be regarded as the brain of the system. A relatively small CPU is commonly called a *microprocessor*. The CPU determines what computational operations are carried out and the sequence in which the operations are executed. During such operation, the CPU makes use of one or more special storage locations within itself known as *registers*. Another part of the CPU is the *arithmetic and logic unit* (ALU), which is where all arithmetic operations are evaluated. The CPU operates according to a sequential list of required operations defined by a computer program, known as the computer software. This program is held in the second of the three system components known as the computer memory.

The *computer memory* also serves several other functions besides this role of holding the computer program. One of these is to provide temporary storage locations that the CPU uses to store variables during execution of the computer program. A further common use of memory is to store data tables that are used for scaling and variable conversion purposes during program execution.

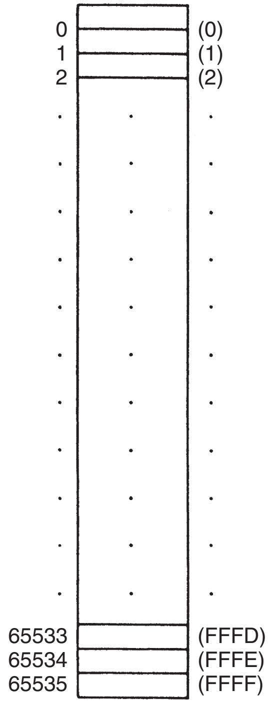

Figure 10.2
Schematic representation of computer memory (numbers in parentheses are memory addresses in hexadecimal notation).

Memory can be visualized as a consecutive sequence of boxes in which various items are stored, as shown in Figure 10.2 for a typical memory size of 65,536 storage units. If this storage mechanism is to be useful, then it is essential that a means be provided for giving a unique label to each storage box. This is achieved by labeling the first box as 0, the next one as 1, and so on for the rest of the storage locations. These numbers are known as the *memory addresses*. While these can be labeled by decimal numbers, it is more usual to use hexadecimal notation.

Two main types of computer memory exist and there are important differences between these. The two kinds are *random access memory* (RAM) and *read only memory* (ROM). The CPU can both read from and write to the former, but it can only read from the latter. The importance of ROM becomes apparent if the behavior of each kind of memory when the power supply is turned off is considered. At power-off time, RAM loses its contents but ROM maintains them, and this is the value of ROM. Intelligent devices normally use ROM for storage of the program and data tables and just have a small amount of RAM that is used by the CPU for temporary variable storage during program execution.

The third essential element of a computer system is the *input—output (I/O) interface*, which allows the computer to communicate with the outside world by reading in data values and outputting results after the appropriate computation has been executed. In the case of a microcomputer performing a signal processing function within an intelligent device, this means reading in the values obtained from one or more sensors and outputting a processed value for presentation at the instrument output. All such external peripherals are identified by a unique number, as for memory addresses.

Communication between these three computer elements is provided by three electronic highways known as the *data bus*, the *address bus*, and the *control bus*. At each data transfer operation executed by the CPU, two items of information must be conveyed along the electronic highway, the item of data being transferred and the address where it is being sent. While both of these items of information could be conveyed along a single bus, it is more usual to use two buses that are called the data bus and the address bus. The timing of data transfer operations is important, particularly when transfers take place to peripherals such as disc-drives and keyboards where the CPU often has to wait until the peripheral is free before it can initialize a data transfer. This timing information is carried by a third highway known as the control bus.

The current trend made possible by advances in very large scale integration (VLSI) technology is to incorporate all three functions of central processor unit, memory, and I/O within a single chip (known as a computer on a chip or *microcomputer*). The term “microprocessor” is often used to describe such an integrated unit, but this is strictly incorrect since the device contains more than just processing power.

10.2.2 Computer Operation

As has already been mentioned, the fundamental role of a computer is the manipulation of data. Numbers are used both in quantifying items of data and also in the form of codes that define the computational operations that are to be executed. All numbers that are used for these two purposes must be stored within the computer memory and also transported along the communication buses.

Programming and program execution

In most modes of usage, including use as part of intelligent devices, computers are involved in manipulating data. This requires data values to be input, processed, and output according to a sequence of operations defined by the computer program. However, in practice, programming the microprocessor within an intelligent device is not normally the province of the instrument user, indeed, there is rarely any provision for the user to create or modify operating programs even if he/she wished to do so. There are several reasons for this. Firstly, the signal processing needed within an intelligent device is usually well defined, and therefore it is more efficient for a manufacturer to produce this rather than to have each individual user produce near identical programs separately. Secondly, better program integrity and instrument operation is achieved if a standard program produced by the instrument manufacturer is used. Finally, use of a standard program allows it to be burnt into ROM, thereby protecting it from any failure of the instrument power supply. This also facilitates software maintenance and updates, by the mechanism of the manufacturer providing a new ROM that simply plugs into the slot previously occupied by the old ROM.

However, even though it is not normally a task undertaken by the user, some appreciation of microprocessor programming for an intelligent device is useful background knowledge. To illustrate the techniques involved in programming, consider a very simple program that reads in a value from a sensor, adds a prestored value to it to compensate for a bias in the sensor measurement, and outputs a corrected reading to a display device.

Let us assume that the addresses of the sensor and output display device are 00C0 and 00C1, respectively, and that the required scaling value has already been stored in memory address 0100. The instructions below are formed from the instruction set for a Z80[1] microprocessor and make use of CPU registers A and B.

```
IN A, C0
IN B, 100
ADD A, B
OUT C1, A
```

This list of four instructions constitutes the computer program that is necessary to execute the required task. The CPU normally executes the instructions one at a time, starting at the top of the list and working downward (though jump and branch instructions change this order). The first instruction (IN A,C0) reads in a value from the sensor at address C0 and places the value in CPU register A (often called the accumulator). The mechanics of the execution of this instruction consist of the CPU putting the required address C0 on the address bus and then putting a command on the control bus that causes the contents of the target address (C0) to be copied onto the data bus and subsequently transferred into the A register. The next instruction (IN B,100) reads in a value from address 100 (the prestored biasing value) and stores it in register B. The following instruction (ADD A,B) adds together the contents of registers A and B and stores the result in register A. Register A now contains the measurement read from the sensor but corrected for bias. The final instruction (OUT C1,A) transfers the contents of register A to the output device on address C1.

[1] The Z80 is now an obsolete 8-bit processor but its simplicity is well suited to illustrating programming techniques. Similar, but necessarily more complex, programming instructions are used with current 16- and 32-bit processors.

10.2.3 Computer I/O Interface

The I/O interface connects the computer to the outside world, and is therefore an essential part of the computer system. When the CPU puts the address of a peripheral onto the address bus, the I/O interface decodes the address and identifies the unique computer peripheral with which a data transfer operation is to be executed. The interface also has to interpret the command on the control bus so that the timing of the data transfer is correct. One further very important function of the I/O interface is to provide a physical electronic highway for the flow of data between the computer data bus and the external peripheral. In many computer applications, including their use within intelligent devices, the external peripheral requires signals to be in analog form. Therefore, the I/O interface must provide for conversion between these analog signals and the digital signals required by a digital computer. This is satisfied by analog-to-digital and digital-to-analog conversion elements within the I/O interface.

The rest of this section presents some elementary concepts of interfacing in simple terms.

Address decoding

A typical address bus in a microcomputer is 16 bits wide[2], allowing 65,536 separate addresses to be accessed in the range 0000−FFFF (in hexadecimal representation). Special commands on some computers are reserved for accessing the bottom end 256 of these addresses in the range 0000−00FF, and, if these commands are used, only 8 bits are needed to specify the required address. For the purpose of explaining address-decoding techniques, the scheme below shows how the lower 8 bits of the 16-bit address line are decoded to identify the unique address referenced by one of these special commands. Decoding of all 16 address lines follows a similar procedure but requires a substantially greater number of integrated circuit chips.

Address-decoding is performed by a suitable combination of logic gates. Figure 10.3 shows a very simple hardware scheme for decoding eight address lines. This consists of 256 eight-input NAND gates, which each uniquely decode one of 256 addresses. A NAND gate is a logic element that only gives a logic level 1 output when all inputs are 0, and gives a logic level 0 output for any other combination of inputs. The inputs to the NAND gates are connected onto the lower eight lines of the address bus and the computer peripherals are connected to the output of the particular gates that decode their unique addresses. There are two pins for each input to the NAND gates that respectively invert and do not invert the input signal. By connecting the eight address lines appropriately to

[2] Recently, 32-bit address fields have also become available in some devices.

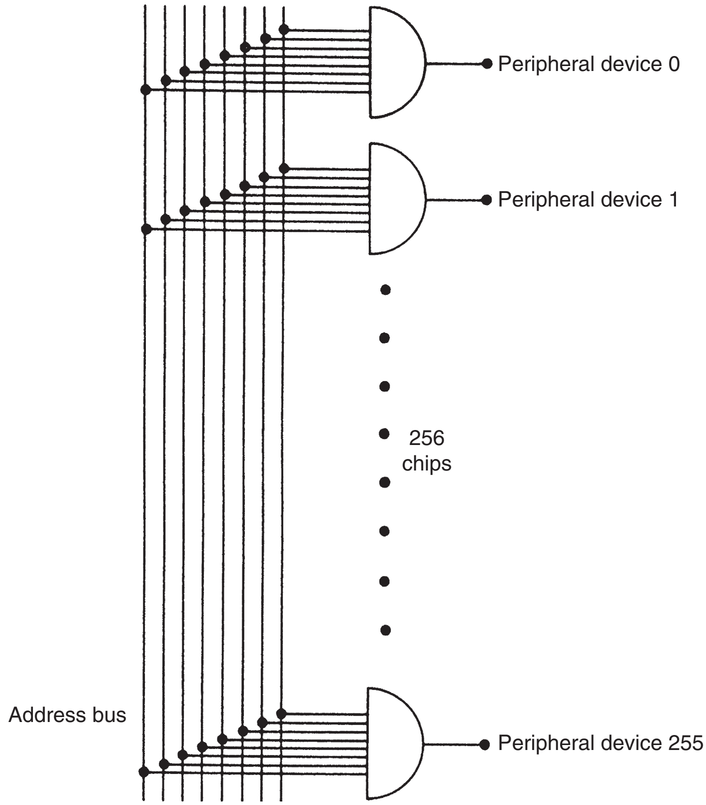

Figure 10.3
Simple hardware scheme for decoding eight-address lines.

these two alternative pins at each input, the gate is made to decode a unique address. Consider for instance the pin connections shown in Figure 10.4. This NAND gate decodes address C5 (hexadecimal), which is 11000101 in binary. Because of the way in which the input pins to the chip are connected, the NAND gate will see all zeros at its input when 11000101 is on the lower 8 bits of the address bus and therefore will have an output of 1. Any other binary number on the address bus will cause this NAND gate to have a zero output.

Data transfer control

The transfer of data between the computer and peripherals is managed by control and status signals carried on the control bus that determine the exact sequencing and timing of I/O operations. Such management is necessary because of the different operating speeds of the computer and its peripherals and because of the multitasking operation of many computers. This means that, at any particular instant when a data transfer operation is requested, either the computer or the peripheral may not be ready to take part in the

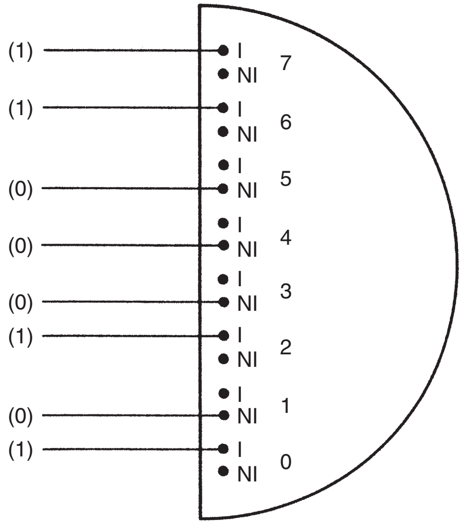

Figure 10.4
Pin connections to NAND gate to decode address C5.

transfer. Typical control and status lines, and their meanings when set at a logic level of 1, are shown below.

- BUSY — Peripheral device busy
- READY — Peripheral device ready for data transfer
- ENABLE — CPU ready for data transfer
- ERROR — Malfunction on peripheral device

Similar control signals are set up by both the computer and peripherals, but different conventions are often used to define the status of each device. Differing conventions occur particularly when the computer and peripherals come from different manufacturers, and might mean, for instance that the computer interprets a logic level of 1 as defining a device to be busy but the peripheral device uses logic level 0 to define "device busy" on the appropriate control line. Therefore, translation of the control lines between the computer and peripherals is required, which is achieved by a further series of logic gates within the I/O interface.

10.2.4 Practical Considerations in Adding Computers to Measurement Systems

The foregoing discussion has presented some of the necessary elements in an I/O interface in a relatively simplistic manner that is just sufficient to give the reader the flavor of what is involved in an interface. Much fine detail has been omitted, and the amount of work involved in the practical design of a real interface should not be underestimated. One significant omission so far is discussion of the scaling that is generally required within the analog–digital interface of a computer. The raw analog input and output signals are generally either too large or too small for compatibility with the operating voltage levels of a digital computer and they have to be scaled upward or downward. This is normally achieved by operational amplifiers and/or potentiometers. The main features of an operational amplifier are its high gain (typically × 1,000,000) and its large bandwidth (typically 1 MHz or better). However, when one is used at very high frequencies, the bandwidth becomes significant. The quality of an amplifier is often measured by a criterion called the gain-bandwidth product, which is the product of its gain and bandwidth. Other important attributes of the operational amplifier, particularly when used in a computer I/O interface or within intelligent devices are its distortion level, overload recovery capacity, and offset level. Special instrumentation amplifiers that are particularly good in these attributes have been developed for instrumentation applications, as described in Chapter 6.

Suitable care must always be taken when introducing a computer into a measurement system to avoid creating sources of measurement noise. This applies particularly where one computer is used to process the output of several transducers and is connected to them by signal wires. In such circumstances, the connections and connecting wires can create noise through electrochemical potentials, thermoelectric potentials, offset voltages introduced by common mode impedances, and AC noise at power, audio, and radio frequencies. Recognition of all these possible noise sources allows them to be eliminated in most cases by employing good practice when designing and constructing the measurement system.

10.3 Intelligent Devices

The term “intelligent device” is used to describe a package containing either a complete measurement system, or else a component within a measurement system, which incorporates a digital processor. Processing of the output of measurement sensors to correct for errors inherent in the measurement process brings about large improvements in measurement accuracy. Such intelligent devices are known by various names such as *intelligent instrument*, *intelligent sensor*, *smart sensor*, and *smart transmitter*. Recently, *smart MEMS devices (Micro-electro-mechanical-systems devices)* have started to become available (these are described in Chapter 13 and also in later chapters covering measurement of specific physical variables). There is no formal definition for any of these names, and there is considerable overlap between the characteristics of particular devices and the name given to them. The name used for any particular device depends largely on the whims and style of device manufacturers. The discussion below tries to lay out the historical development of intelligent devices, and it summarizes the general understanding of the sort of characteristics possessed by the various forms of intelligent device. Details of their application to measure particular physical variables will then be covered in more detail in later chapters of this book.

10.3.1 Intelligent Instruments

The first intelligent instrument appeared over 30 years ago, although high prices when such devices first became available meant that their use within measurement systems initially grew very slowly. However, since these early days, there has been a dramatic reduction in the price of all intelligent devices, and the cost differential between intelligent and conventional devices is now very small. Indeed, an intelligent device is sometimes now cheaper than its nonintelligent equivalent because of the greater sales volume for the intelligent version. Thus, intelligent devices are now routinely bought instead of nonintelligent versions in many cases.

The processor within an intelligent instrument allows it to apply preprogrammed signal processing and data manipulation algorithms to measurements. This prewritten software is often known by the name of *embedded software*. One of the main functions performed by the first intelligent instruments to become available was compensation for environmental disturbances to measurements that cause systematic errors. Thus, apart from a primary sensor to measure the variable of interest, intelligent instruments usually have one or more secondary sensors to monitor the value of environmental disturbances. These extra measurements allow the output reading to be corrected for the effects of environmentally induced errors, subject to the following preconditions being satisfied:

1. The physical mechanism by which a measurement sensor is affected by ambient condition changes must be fully understood and all physical quantities that affect the output must be identified.
2. The effect of each ambient variable on the output characteristic of the primary sensor must be quantified.
3. Suitable secondary sensors for monitoring the value of all relevant environmental variables must be available that will be operate satisfactorily in the prevailing environmental conditions.

Condition (1) above means that the thermal expansion and contraction of all elements within a sensor must be considered in order to evaluate how it will respond to ambient temperature changes. Similarly, the sensor response, if any, to changes in ambient pressure, humidity, gravitational force, or power supply level (active instruments) must be examined.

Quantification of the effect of each ambient variable on the characteristics of the measurement sensor is then necessary, as stated in condition (2). Analytic quantification of ambient condition changes from purely theoretical consideration of the construction of a sensor is usually extremely complex and so is normally avoided. Instead, the effect is quantified empirically in laboratory tests. In such tests, the output characteristic of the sensor is observed as the ambient environmental conditions are changed in a controlled manner.

One early application of intelligent instruments was in volume flow rate measurement, where the flow rate is inferred by measuring the differential pressure across an orifice plate placed in a fluid-carrying pipe (see Chapter 15). The flow rate is proportional to the square root of the difference in pressure across the orifice plate. For a given flow rate, this relationship is affected both by the temperature and by the mean pressure in the pipe, and changes in the ambient value of either of these cause measurement errors. A typical intelligent flowmeter therefore contains three sensors, a primary one measuring pressure difference across the orifice plate and secondary ones measuring absolute pressure and temperature. The instrument is programmed to correct the output of the primary differential-pressure sensor according to the values measured by the secondary sensors, using appropriate physical laws that quantify the effect of ambient temperature and pressure changes on the fundamental relationship between flow and differential pressure. Even 30 years ago, such intelligent flow-measuring instruments achieved typical inaccuracy levels of $\pm 0.1\%$, compared with $\pm 0.5\%$ for their nonintelligent equivalents.

Although automatic compensation for environmental disturbances is a very important attribute of intelligent instruments, many versions of such devices perform additional functions, and this was so even in the early days of their development. For example, the orifice-plate flowmeter just discussed usually converts the square root relationship between flow and signal output into a linear one, thus making the output much easier to interpret. Other examples of the sort of functions performed by intelligent instruments are as follows:

- Correction for the loading effect of measurement on the measured system.
- Signal damping with selectable time constants.
- Switchable ranges (using several primary sensors within the instrument that each measure over a different range).
- Switchable output units (e.g., display in Imperial or SI units).
- Linearization of the output.
- Self-diagnosis of faults.
- Remote adjustment and control of instrument parameters from up to 1500 m away via 4-way, 20 mA signal lines.

These features will be discussed in greater detail under the later headings of smart sensors and smart transmitters.

Over the intervening years since their first introduction, the size of intelligent instruments has gradually reduced and the functions performed have steadily increased. One particular development has been the inclusion of a microprocessor within the sensor itself, in devices that are usually known as *smart sensors*. As further size reduction and device integration has taken place, such smart sensors have been incorporated into packages with other sensors and signal processing circuits etc. While such a package conforms to the definition of an intelligent instrument given previously, most manufacturers now tend to call the package a *smart transmitter* rather than an intelligent instrument, although the latter term has continued in use in some cases.

10.3.2 Smart Sensors

The name *smart sensor* is most commonly used to describe any sensor that has local processing power that enables it to react to local conditions without having to refer back to a central controller. Smart sensors are usually at least twice as accurate as nonsmart devices, have reduced maintenance costs and require less wiring to the site where they are used. In addition, long-term stability is improved, reducing the required calibration frequency.

The functions possessed by smart sensors vary widely, but consist of at least some of the following:

- Remote calibration capability.
- Self-diagnosis of faults.
- Automatic calculation of measurement accuracy and compensation for random errors.
- Adjustment for measurement nonlinearities to produce a linear output.
- Compensation for the loading effect of the measuring process on the measured system.

Calibration capability

Self-calibration is very simple in some cases. Sensors with an electrical output can use a known reference voltage level to carry out self-calibration. Also, load-cell types of sensor, which are used in weighing systems, can adjust the output reading to zero when there is no applied mass. In the case of other sensors, two methods of self-calibration are possible, use of a look-up table and an interpolation technique. Unfortunately, a *look-up table* requires a large memory capacity to store correction points. Also, a large amount of data has to be gathered from the sensor during calibration. In consequence, the interpolation calibration technique is preferable. This uses an interpolation method to calculate the correction required to any particular measurement and only requires a small matrix of calibration points.

Self-diagnosis of faults

Smart sensors perform self-diagnosis by monitoring internal signals for evidence of faults. While it is difficult to achieve a sensor that can carry out self-diagnosis of all possible faults that might arise, it is often possible to make simple checks that detect many of the more common faults. One example of self-diagnosis in a sensor is measuring the sheath capacitance and resistance in insulated thermocouples to detect breakdown of the insulation. Usually, a specific code is generated to indicate each type of possible fault (e.g., a failing of insulation in a device).

One difficulty that often arises in self-diagnosis is in differentiating between normal measurement deviations and sensor faults. Some smart sensors overcome this by storing multiple measured values around a set-point and then calculating minimum and maximum expected values for the measured quantity.

Uncertainty techniques can be applied to measure the impact of a sensor fault on measurement quality. This makes it possible in certain circumstances to continue to use a sensor after it has developed a fault. A scheme for generating a validity index has been proposed that indicates the validity and quality of a measurement from a sensor (Henry, 1995).

Automatic calculation of measurement accuracy and compensation for random errors

Many smart sensors can calculate measurement accuracy online by computing the mean over a number of measurements and analyzing all factors affecting accuracy. This averaging process also serves to greatly reduce the magnitude of random measurement errors.

Adjustment for measurement nonlinearities

In the case of sensors that have a nonlinear relationship between the measured quantity and the sensor output, digital processing can convert the output to a linear form, providing that the nature of the nonlinearity is known so that an equation describing it can be programmed into the sensor.

10.3.3 Smart Transmitters

In concept, a smart transmitter is almost identical to other intelligent devices describer earlier. While the name “smart transmitter” is sometimes used interchangeably with the name “smart sensor,” it is perhaps more commonly used to describe an intelligent device that has greater functionality than just the computer-assisted sensing of a variable that a smart sensor conventionally does, particularly in respect of output functions and ability to compensate for environmental disturbances. In some instances, smart transmitters are known alternatively as *intelligent transmitters*. The term *multivariable transmitter* is also sometimes used, particularly for a device like a smart flow-measuring instrument. This latter device measures absolute pressure, differential pressure, and process temperature, and computes both the mass flow rate and volume flow rate of the measured fluid.

Many of the smart transmitters that are presently available still have an analog output, because of the continuing popularity and investment in 4–20 mA current transmission systems. While most devices now available have a digital output, many users convert this to analog form to maintain compatibility with existing instrumentation systems.

Comparison of performance with other forms of transmitter

The capabilities of smart transmitters can perhaps best be emphasized by comparing them with the attributes of analog transmitters and also with devices known as *programmable transmitters*. The latter have computational power but do not have a bidirectional communication ability, meaning that they are not truly intelligent. The respective attributes of these devices are as follows:

1. Analog transmitters:
   a. Require one transmitter for every sensor type and every sensor range.
   b. Require additional transmitters to correct for environmental changes.
   c. Require frequent calibration.
2. Programmable transmitters:
   a. Include a microprocessor but do not have bidirectional communication (hence are not truly intelligent).
   b. Require field calibration.
3. Smart transmitters:
   a. Include a microprocessor and have bidirectional communication.
   b. Include secondary sensors that can measure, and so compensate, for environmental disturbances.
   c. Usually incorporate signal conditioning and a–d conversion.
   d. Often incorporate multiple sensors covering different measurement ranges and allow automatic selection of required range. The range can be readily altered if initially estimated incorrectly.
   e. Have a self-calibration capability that allows removal of zero drift and sensitivity drift errors.
   f. Have a self-diagnostic capability that allows them to report problems or requirements for maintenance.
   g. Can adjust for nonlinearities to produce a linear output.

Summary of advantages of smart transmitters

The main disadvantage that could be cited for using a smart transmitter instead of a nonsmart one is that it is usually a little larger and heavier than its nonsmart equivalent, but this is not a problem in most applications. There is also normally a greater associated purchase cost. However, these potential disadvantages are minor in most circumstances and greatly outweighed by the advantages that smart transmitters have, which can be summarized as follows:

- Improved accuracy and repeatability.
- Automatic calculation of measurement accuracy and compensation for random errors.
- Compensation for the loading effect of the measuring process on the measured system.
- Long-term stability is improved and required recalibration frequency is reduced.
- Adjustment for measurement nonlinearities to produce a linear output.
- Reduced maintenance costs.
- Self-diagnosis of faults.
- Large range coverage, allowing interoperability and giving increased flexibility.
- Remote adjustment of output range, on command from a portable keyboard or from a PC. This saves on technician time compared with carrying out adjustment manually.
- Reduction in number of spare instruments required, since one spare transmitter can be configured to cover any range and so replace any faulty transmitter.
- Possibility of including redundant sensors, which can be used to replace failed sensors and so improve device reliability.
- Allowing remote recalibration or reranging by sending a digital signal to them.
- Ability to store last calibration date and indicate when next calibration is required.
- Single penetration into the measured process rather than the multiple penetration required by discrete devices, making installation easier and cheaper.
- Ability to store data so that plant and instrument performance can be analyzed. For example, data relating to the effects of environmental variations can be stored and used to correct output measurements over a large range.

Self-calibration

The common use of multiple primary sensors and secondary sensors to measure environmental parameters mean that the self-calibration procedure for smart transmitters is more complicated than that for simpler smart sensors. While the general approach to self-calibration remains similar to that explained earlier for smart sensors, the calibration procedure has to be repeated for each primary and secondary sensor within the transmitter. Recommended practice is to use the simplest calibration procedures available for each sensor in the transmitter. However, care has to be taken to ensure that any interaction between measured variables is taken account of. This often means that look-up tables in a smart transmitter have to have a particularly large memory requirement in order to take the cross-sensitivity to other parameters (e.g., temperature) into account, because a matrix of correction values has to be stored. This means that interpolation calibration is even more preferable to look-up table calibration than it is in the case of calibrating smart sensors.

Self-diagnosis and fault detection

Fault diagnosis in the sensors within a smart transmitter is often difficult because it is not easy to distinguish between measurement deviation due to a sensor fault and deviation due to a plant fault. The best theoretical approach to this difficulty is to apply mathematical modeling techniques to the sensor and plant in which it is working, with the aim of detecting inconsistencies in data from the sensor. However, there are very few industrial applications of this approach to fault detection in practice, firstly, because of the cost of implementation and, secondly, because of the difficulty in obtaining plant models that are robust to plant disturbances. Thus, it is usually necessary to resort to having multiple sensors and using a scheme such as two-out-of-three voting. Further advice on self-checking procedures can be found elsewhere (Brignell and White, 1996).

10.4 Communication with Intelligent Devices

The inclusion of computer processing power in intelligent instruments and intelligent actuators creates the possibility of building an instrumentation system where several intelligent devices collaborate together, transmit information to one another, and execute process control functions. Such an arrangement is often known as a *distributed control system*. Additional computer processors can also be added to the system as necessary to provide the necessary computational power when the computation of complex control algorithms is required. Such an instrumentation system is far more fault-tolerant and reliable than older control schemes where data from several discrete instruments is carried to a centralized computer controller via long instrumentation cables. This improved reliability arises from the fact that the presence of computer processors in every unit injects a degree of redundancy into the system. Therefore, measurement and control action can still continue, albeit in a degraded form, if one unit fails.

In order to effect the necessary communication when two or more intelligent devices are to be connected together as nodes in a distributed system, some form of electronic highway must be provided between them that permits the exchange of information. Apart from data transfer, a certain amount of control information also has to be transferred. The main purpose of this control information is to make sure that the target device is ready to receive information before data transmission starts. This control information also prevents more than one device trying to send information at the same time.

In modern installations, all communication and data transmission between processing nodes in a distributed instrumentation and control system is carried out digitally along some form of electronic highway. The highway can either be a parallel interface, a local area network (LAN), a digital fieldbus, or a combined LAN/fieldbus. A parallel interface protocol is commonly used for connecting a small number of devices spread over a small geographical area, typically a single room. In the case of a large number of devices that are spread over larger geographical distances, typically a single building or site, an electronic highway in the form of either a LAN or a digital fieldbus is used. Instrumentation networks that are geographically larger than a single building or site can also be built, but these generally require transmission systems that include telephone lines as well as local networks at particular sites within the large system.

Manufacturers normally provide all the hardware and software necessary in order to create an instrumentation network using the various intelligent devices in their product range. However, problems usually occur if the designer of an instrumentation network wish to use components sourced from different manufacturers where quite serious compatibility issues can arise. To overcome this, the Institute of Electronics and Electrical Engineers has developed IEEE 1451, 2007. This is a series of smart device interface standards that allow components from different manufacturers to be connected onto the same network.

10.4.1 I/O Interface

An *I/O interface* is required to connect each intelligent device onto the electronic highway. Sensors with a digital output pose little interfacing problems. However, many intelligent devices still have an analog output that uses standard 4–20 mA protocol and requires an analog-to-digital converter in the I/O interface. For these, a protocol known as *HART* (Highway Addressable Remote Transducer) is the one that is most widely used to provide the necessary connection of such devices onto a digital network. HART is a bus-based networking protocol that has become a de facto standard for intelligent devices with an analog sensor output. HART-compatible devices are provided by all major instrument manufacturers.

HART was always intended to be an interim network protocol to satisfy communication needs in the transitional period between the use of analog communication with nonintelligent devices and fully digital communication with intelligent devices according to digital fieldbus protocol. Because of this need to support both old and new systems, HART supports two modes of use, a hybrid mode and a fully digital mode.

In *hybrid mode*, status/command signals are digital but data transmission takes place in analog form (usually in 4–20 mA format). One serious limitation of this mode is that it is not possible to transmit multiple measurement signals on a single bus, since the analog signals would corrupt each other. Hence, when HART is used in hybrid mode, the network must be arranged in a star configuration, using a separate line for each field device rather than a common bus.

In *fully digital mode*, data transmission is digital as well as status/command signals. This enables one cable to carry signals for up to 15 intelligent devices. In practice, the fully digital mode of HART is rarely used, since the data transmission speed is very limited compared with fieldbus protocols. Therefore, the main application of the HART protocol has been to provide a communication capability with intelligent devices when existing analog measurement signal transmission has to be retained because conversion to fully digital operation would be too expensive.

10.4.2 Parallel Data Bus

There are a number of different parallel data buses in existence. All of these have the common feature of transmitting data in parallel, that is, several bits are transmitted simultaneously. They also have separate data and control lines, which means that the data lines are used solely for data transmission and all control signals are routed onto dedicated control lines. This optimizes data transmission speed. However, apart from having this common functionality, there is little compatibility between the different parallel data buses available, with significant differences existing in the number of data lines used, the number of control lines used, the interrupt structure used, the data timing system, and the logic levels used for operation. Equipment manufacturers tend to keep to the same parallel interface protocol for all their range of devices, but different manufacturers use different protocols. Thus, while it will normally be easy to connect together a number of intelligent devices that all come from the same manufacturer, interfacing difficulties are likely to be experienced if devices from different manufacturers are connected together. Fortunately, the situation in the field is not as bad as it sounds, because the IEEE 488 bus has now gained prominence as the preferred parallel databus for instrumentation networks and has been adopted by a large number of manufacturers. Since it was first introduced in 1975, the published standard for this bus has been revised on several occasions, the most recent being in 2004 when the Institute of Electrical and Electronics Engineers (IEEE) published a standard jointly with the International Electrotechnical Commission (IEC) as standard IEC 60488/IEEE488 (IEC/IEEE, 2004, parts 1 and 2). The bus provides a parallel interface that facilitates the connection of intelligent instruments, actuators, and controllers within a single room. Physically, the bus consists of a shielded, 24-conductor cable. For a standard IEEE 488 bus, the maximum length of bus allowable is 20 m, with no more than 15 instruments distributed along its length. However, this limit on length and number of instruments can be overcome by using an active (i.e., with auxiliary power supply) bus extender. The maximum distance between two particular units on the bus should not exceed about 2 m. The maximum data transfer rate permitted by the bus is 1 Mbit/s in theory, though the maximum data rate achieved in practice over a full 20 m length of bus is more likely to be in the range of 250–500 kbit/s.

10.4.3 Local Area Networks

LANs transmit data in digital format along serial transmission lines. Synchronous transmission is normally used because this allows relatively high transmission speeds by transmitting blocks of characters at a time. A typical data block consists of 80 characters: this is preceded by a synchronization sequence and followed by a stop sequence. The synchronization sequence causes the receiver to synchronize its clock with that of the transmitter. The two main standards for synchronous, serial transmission are RS422 and RS485. These are now formally published by the ANSI Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA) with the codes ANSI/TIA/EIA-422-B and ANSI/TIA/EIA-485. A useful comparison between the performance and characteristics of each of these and the older RS232 standard (asynchronous serial transmission) can be found in Brook and Herklot (1996).

LANs have particular value in the monitoring and control of systems that have a number of separate sensors, actuators, and control units that are dispersed over a large area. Indeed, for such large instrumentation systems, a LAN is the only viable transmission medium in terms of performance and cost. Parallel data buses, which transmit data in analog form, suffer from signal attenuation and noise pickup over large distances, and the high cost of the long, multicore cables that they need is prohibitive.

However, the development of instrumentation networks is not without problems. Careful design of the network is required to prevent corruption of data when two or more devices on the network try to access it simultaneously and perhaps put information onto the data bus at the same time. This problem is solved by designing a suitable network protocol that ensures that network devices do not access the network simultaneously, thus preventing data corruption.

In a LAN, the electronic highway can take the form of either copper conductors or fiber optic cable. Copper conductors are the cheapest option and allow transmission speeds up to 10 Mbit/s, using either a simple pair of twisted wires or a coaxial cable. However, fiber optic cables are preferred in many networks for a number of reasons. The virtues of fiber optic cables as a data transmission medium have been expounded in Chapter 8. Apart from the high immunity of the signals to noise, a fiber optic transmission system can transfer data at speeds up to 240 Mbit/s. The reduction in signal attenuation during transmission also means that much longer transmission distances are possible without repeaters being necessary. For instance, the allowable distances between repeaters for a fiber optic network are quoted as 1 km for half-duplex operation and up to 3.5 km for full-duplex operation. In addition, the bandwidth of fiber-optic transmission is higher than for electrical transmission. Some cost saving can be achieved by using plastic fiber-optic cables, but these cannot generally be used over distances greater than about 30 m because signal attenuation is too high.

There are many different protocols for LANs but these are all based on one of three network structures known as star networks, bus networks, and ring networks, as shown in Figure 10.5. A LAN operates within a single building or site and can transmit data over distances up to about 500 m without signal attenuation being a problem. For transmission over greater distances, telephone lines are used in the network. Intelligent devices are interfaced to the telephone line used for data transmission via a modem. The modem converts the signal into a frequency-modulated analog form. In this form, it can be transmitted over either the public-switched telephone network or over private lines rented from telephone companies. The latter, being dedicated lines, allow higher data transmission rates.

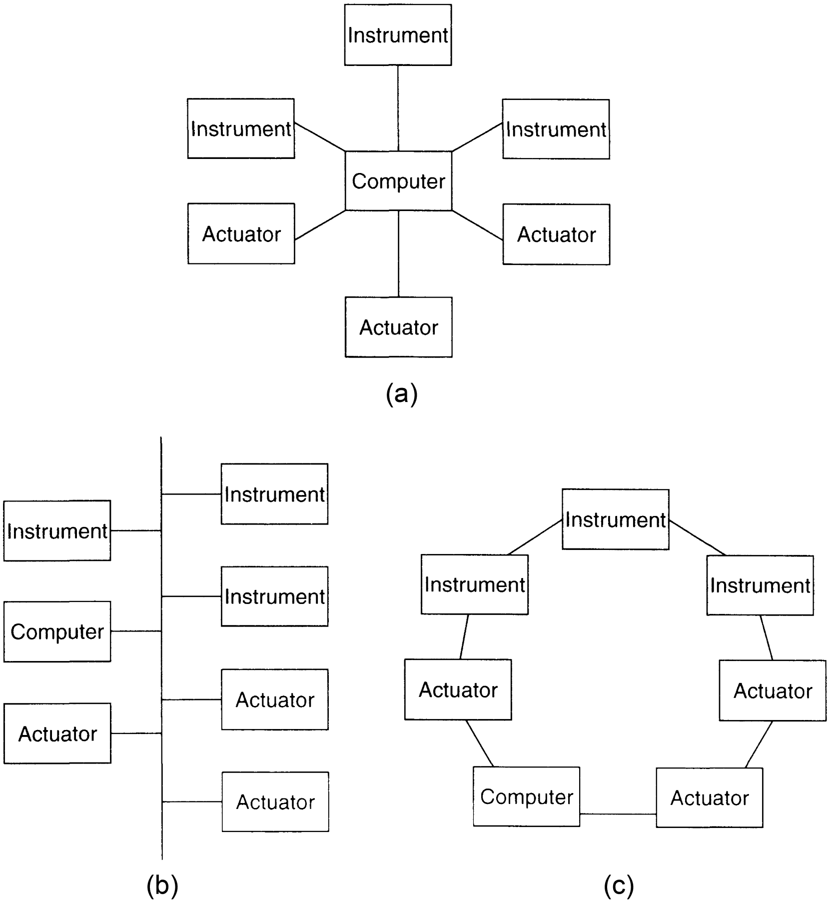

Figure 10.5
Network protocols: (a) Star; (b) bus; (c) ring.

Star networks

In a *star network*, each instrument and actuator is connected directly to the supervisory computer by its own signal cable. One apparent advantage of a star network is that data can be transferred if necessary using a simple serial communication protocol such as RS232. This is an industry-standard protocol and so compatibility problems do not arise, but it represents very old technology in which data transfer is very slow. Because of this speed problem, parallel communication is usually preferred even for star networks.

While star networks are simple in structure, the central supervisory computer node is a critical point in the system and failure of this means total failure of the whole system. When any device in the network needs to communicate with another device, a request has to be made to the central supervisory computer and all data transferred is routed through this central node. If the central node is inoperational for any reason then data communication in the network is stopped.

Ring and bus networks

In contrast to star networks, both ring and bus networks have a high degree of resilience in the face of one node breaking down. Hence, they are generally preferred to star networks. If the processor in any node breaks down, the data transmission paths in the network are still maintained. Thus, the network can continue to operate, albeit at a degraded performance level, using the remaining computational power in the other processors. Most computer and intelligent instrument/actuator manufacturers provide standard conversion modules that allow their equipment to interface to one of these standard networks.

In a *ring network*, all the intelligent devices are connected to a bus that is formed into a continuous ring. Ring protocol sends a special packet (or token) continuously round the ring to control access to the network. A station can only send data when it receives the token. During data transmission, the token is attached to the back of the message sent so that, once the information has been safely received, the token can continue on its journey round the network.

A *bus network* is similar to a ring network but the bus that the devices are connected onto is not continuous. Bus networks are also resilient toward the breakdown of one node in the network. A *contention protocol* is normally used. This allows any station to have immediate access to the network unless another station is using it simultaneously, in which case the protocol manages the situation and prevents data loss/corruption. *Ethernet* is the most common form of bus network and this has now gained a dominant position in the LAN marketplace.

10.4.4 Digital Fieldbuses

"Fieldbus" is a generic word that describes a range of high-speed, bus-based, network protocols that support two-way communication in digital format between a number of intelligent devices in a LAN. All forms of transmission are supported including twisted pair, coaxial cable, fiber optic, and radio links.

Intelligent devices in an automated system comprise a range of control elements, actuators, information processing devices, storage systems, and operator displays as well as measurement devices. Hence, any fieldbus protocol must include provision for the needs of all system elements, and the communication requirements of field measurement devices cannot be viewed in isolation from these other elements. The design of a network protocol also has to cater for implementation in both large and small plants. A large plant may contain a number of processors in a distributed control system and have a large number of sensors and actuators. On the other hand, a small plant may be controlled by a single personal computer that provides an operator display on its monitor as well as communicating with plant sensors and actuators.

Since fieldbus technology was first introduced in 1988, there was no rapid move to develop an international standard and, in consequence, different manufacturers all developed their own versions. This resulted in more than 50 different fieldbus protocols, with the more prominent ones being Foundation Fieldbus, Profibus, WorldFIP, ControlNet, P-net, and Interbus. Each fieldbus version supports all devices within the product range of one or several manufacturers, but there is little compatibility between the different protocols on offer. They differ in many major respects such as message format, access protocols, and rules for performance prediction. In recognition of the difficulties inherent in attempting to connect devices from different manufacturers that use a variety of incompatible interface standards and network protocols, the International Electrotechnical Commission (IEC) set up a working part that was charged with defining a standard interface protocol. However, individual manufacturers continued to develop their own versions of fieldbus in parallel with the IEC initiative. The result of this is that, when the IEC published its first fieldbus standard in 1999 (IEC 61,158), the document had more than 4000 pages and covered 8 different protocol sets, these defining a standard for each of the 8 main fieldbus systems then in operation (Foundation Fieldbus, Profibus, ControlNet, P-Net, Foundation fieldbus HSE (High-Speed Ethernet), SwiftNet, WorldFIP, Interbus). In the period following 1999, addendums to IEC61158 were published covering further new fieldbus standards, particularly in respect of safety buses and high-speed, Ethernet-based fieldbuses.

Despite the failure of IEC 61,158 to establish a single fieldbus standard in 1999 and the following years, a consortium of major international instrumentation manufacturers set up the Fieldbus Foundation in an attempt to move toward one worldwide fieldbus standard. This resulted in a version of Foundation Fieldbus that provided a common standard for, and interchangeability between, all devices manufactured by members of the consortium. However, competing standards, in particular Profibus, remained.

The basic architecture of Foundation Fieldbus has two levels, an upper and a lower. The lower level provides for communication between field devices and field I/O devices while the upper level enables field I/O devices to communicate with controllers. These two levels have quite different characteristics. The lower level generally requires few connections, only needs a relatively slow data-transfer rate, and must support intrinsically safe working. On the other hand, the upper level requires numerous connections and fast data transfer, but does not have to satisfy intrinsic-safety requirements. Three standard bus speeds are currently specified for the Foundation Fieldbus lower level of 31.25 kbit/s, 1 Mbit/s, and 2.5 Mbit/s. Maximum cable lengths allowed are 1900 m at 31.25 kbit/s, 750 m at 1 Mbit/s, and 500 m at 2.5 Mbit/s. For the upper Foundation Fieldbus layer, a high-speed ethernet protocol provides a data transfer rate up to 100 Mb/s.

The most recent attempt by the IEC to establish a fieldbus standard was published in 2007 (IEC 61158, 2007). This sets general standards for specification of physical layers, data links, and application layers, but has given up on earlier attempts to establish one common, worldwide fieldbus protocol. In the marketplace, Foundation Fieldbus and Profibus remain the dominant fieldbus versions.

10.5 Summary

The primary purpose of this chapter has been to introduce the subject of intelligent devices. However, since computational power is the component that separates intelligent devices from their nonintelligent counterparts, we started the chapter off by reviewing the main principles of digital computation. This led us to study the main elements in a computer, how computers operate particularly in respect of program execution, and how computers interface to outside components. We ended this introduction to digital computation with a review of the practical issues that have to be considered when incorporating computers into measurement devices.

Moving on to the subject of intelligent devices, we found that several different terms are used to describe these. Prominent among these terms are names like intelligent instrument, intelligent sensor, smart sensor, and smart transmitter. We learned that there are no industry-wide definitions of what any of these names mean, except that all are distinguished by incorporating some form of computational power. As a consequence, the same kind of device, even with very similar attributes, may be known by two or more names. Therefore, the name that is used to describe a particular intelligent device is subject to the whims and style of its manufacturer.

Having explained this arbitrary nature in the way that intelligent devices are named, we went on to describe some commonly held views about the sort of functions performed by devices known as smart sensors and the functions typically performed by devices known as smart transmitters. The conclusion drawn from this comparison of functions was that devices called smart transmitters tended to have a greater functionality than those called smart sensors.

Our investigation into the features of intelligent devices led us to conclude that these have significant advantages compared with nonintelligent devices. Perhaps the biggest single benefit is improved measurement accuracy. This is achieved by use of a computer processor within each device that performs actions like compensation for random errors, adjustment for measurement nonlinearities, and compensation for the loading effect of the measuring process on the measured system. Processing power also enables devices to perform functions like remote self-calibration and self-diagnosis of faults. Smart transmitters typically have additional features such as incorporation of multiple primary sensors covering different measurement ranges and allowing automatic selection of required range, inclusion of secondary sensors that can measure and compensate for environmental disturbances, plus incorporation of signal conditioning and analog-to-digital conversion functions. Sometimes smart transmitters also have redundant sensors, which can be used to replace failed sensors and so improve device reliability.

We then went on to look at the issues surrounding the communication between intelligent devices and other elements in a measurement/process control system. We noted that all communication and data transmission between processing nodes in a distributed instrumentation and control system required the use of some form of electronic highway, which can be either a parallel interface, a LAN, a digital fieldbus, or a combined LAN/fieldbus. We then concluded the chapter with a look in broad detail at the various features in these alternative forms of electronic highway, but observed that there was little point in studying the fine details of any particular form of highway because there were continuing developments in the format of highways, and particularly in the protocols used in LANs and digital fieldbuses. This means that the inclusion of any detailed study in the book would become out of date very quickly.

10.6 Problems

10.1 Explain the principal components in the computational element contained within an intelligent instrument.

10.2 What are the two main types of computer memory? Which type is predominantly used in intelligent instruments and why?

10.3 What are the mechanisms for programming and program execution within an intelligent instrument?
10.4 Discuss the operation of the I/O interface within the processor of an intelligent instrument, mentioning particularly the mechanisms of address decoding and data transfer control.
10.5 What are the practical considerations involved in about implementing a computer processor within a measurement system?
10.6 How does an intelligent instrument correct for environmentally induced errors in measurements? What preconditions must be satisfied to allow an intelligent instrument to correct for such errors? How are these preconditions satisfied?
10.7 Explain how adding intelligence to an instrument improves the accuracy of volume flow rate measurements.
10.8 What additional functions does an intelligent instrument typically perform apart from the correction of environmentally induced errors in measurements?
10.9 Describe the typical function of devices known as *smart sensors*.

References

Brignell, J., White, N., 1996. Intelligent Sensor Systems. Institute of Physics Publishing.
Brook, N., Herklot, T., June 1996. Choosing and Implementing a Serial Interface. Electronic Engineering pp. 46−54.
Henry, M., May 1995. Self-validation improves Coriolis flowmeter. Control Engineering pp. 81−86.
IEC61158, 2007. Digital Data Communications for Measurement and Control − Fieldbus for Use in Industrial Control Systems. International Electrotechnical Commission.
IEC-60488−1/IEEE488.1, 2004. Higher Performance Protocol for the Standard Digital Interface for Programmable Instrumentation. Part 1: General. Institute of Electrical and Electronic Engineers/International Electrotechnical Commission.
IEC-60488−2/IEEE488.2, 2004. Standard Digital Interface for Programmable Instrumentation. Part 2: Codes, Formats, Protocols and Common Commands. Institute of Electrical and Electronic Engineers/International Electrotechnical Commission.
IEEE 1451, 2007. Standard for a Smart Transducer Interface for Sensors and Actuators (Related parts of this standard referring to particular aspects of the interface are published separately as IEEE 1451−0, 2007, IEEE 1451-1, 1999, IEEE 1451-2, 1997, IEEE 1451-3, 2003, IEEE 1451-4, 2004, IEEE 1451-5, 2007 and IEEE 1451-7, 2010).