

16

Troubleshooting Digital Circuits and Test Equipment

This chapter looks at two interrelated aspects of digital circuit troubleshooting, namely *troubleshooting* and the *test equipment*. The chapter is divided into two parts: the first part discusses troubleshooting guidelines for a variety of digital devices and circuits; the second part deals with test and measuring equipment. The chapter begins with general guidelines to troubleshooting digital circuits and then moves on to discuss techniques for troubleshooting specific digital building blocks such as logic gates, flip-flops, counters, registers, arithmetic circuits, memory devices and so on. In the second part of the chapter, some of the more commonly used test and measuring equipment is discussed at length. The test instruments covered here are not necessarily ones that are required by a troubleshooter during the course of fault finding. They also include instruments that are the result of advances in digital technology and have a digital-dominated internal hardware. In fact, this constitutes one of the most important areas where digital technology has so strongly manifested itself. Some of these instruments, such as the digital multimeter (DMM), the logic probe and the digital storage oscilloscope (DSO), are the essential tools of any digital circuit troubleshooter. The chapter is adequately illustrated with a large number of case studies related to digital circuit troubleshooting.

16.1 General Troubleshooting Guidelines

Irrespective of the type and complexity of the digital circuit to be troubleshooted, the following three-step procedure should be followed:

1. Fault detection or identification.
2. Fault isolation.
3. Remedial measures.

Fault detection means knowing the nature of the fault, which could be done by comparing the actual or present performance of the circuit with the ideal or desired performance. Complete knowledge about the nature of the fault often gives an idea about the nature of tests and measurements to be performed to isolate the fault. It is therefore important that the nature of the fault is properly understood and appreciated in terms of the functions performed by various parts of the overall digital circuit or system.

Fault isolation means performing tests and making measurements with the available diagnostic tools to know precisely where the fault lies. This could be in the form of a faulty component or a shorted or open track and so on. The level of documentation that is available plays an important role in deciding about the type of measurements to be made to isolate the fault. Comprehensive documentation helps in significantly reducing the time period required to actually latch on to the faulty component or area. Again, the faults could either be internal to the components and devices, digital integrated circuits, for instance, or external to the components. These two types of fault are discussed in the following paragraphs.

Remedial measures follow the fault isolation. This could mean repairing of tracks or replacement of one or more components.

16.1.1 Faults Internal to Digital Integrated Circuits

Digital circuits and systems are dominated by the use of digital integrated circuits (ICs). The number of discrete devices is usually much smaller than the number of ICs used. Therefore, the knowledge of typical faults that can occur in digital ICs is central to fault isolation in digital systems. The most commonly observed defects or failures in digital ICs are as follows:

1. Shorting of input or output pins to V_{CC} or ground terminals or shorting of tracks.
2. Open circuiting of input or output pins.
3. Shorting of two pins other than ground and V_{CC} pins.
4. Failure of the internal circuitry of the IC.

16.1.1.1 Internal Shorting of Input or Output Pins to GND or V_{CC}

This is one of the commonly observed faults internal to digital ICs. Shorting of one or more of the input or output pins internally to GND puts a permanent LOW on the pin(s). This could have several manifestations depending upon the nature of the IC and also upon the nature of the component driving these pins. Some of these manifestations are given in the following examples:

1. If an input pin that is internally shorted to GND is being driven from an output pin of another IC, that particular output pin will face a permanent ground and will be affected accordingly. A pulsating signal, if originally present at that pin, will vanish.
2. If the shorted input terminal happens to be that of a NAND gate, the output of the gate will permanently go to the logic HIGH state and will not respond to any changes on the other input.
3. If the shorted input pin is the PRESET input of a presetable, clearable $J-K$ flip-flop with active LOW PRESET and CLEAR inputs, the output of this particular flip-flop will always be in the logic HIGH state irrespective of the status of the J and K inputs.
4. Shorting of the output pin to GND puts a permanent logic LOW on that pin, and this particular output does not respond to changes on the corresponding input pins.

Shorting of input or output pins to V_{CC} puts a permanent HIGH on those pins. If it is the output pin, it again fails to respond to any changes on the corresponding input pins, and, if it is the input pin, it affects the output response of the IC depending upon the nature of the IC. The following examples illustrate this point further:

1. If it is the input of the NAND gate, a permanent HIGH on the input permanently transforms it into an inverter circuit, which means that the NAND gate no longer performs its intended function.
2. If it happens to be the input terminal of an OR gate, it drives its corresponding output to a permanent logic HIGH state.

16.1.1.2 Open Circuiting of Input or Output Pins

Open circuiting of input and output pins occurs for reasons internal to the IC when the fine wire that connects the IC pin to the relevant location on the chip breaks. The effects of open circuiting can be serious too. For instance, an open on the input or output pin makes it a floating terminal, and, if the IC belongs to the TTL logic family, it will be treated as logic HIGH. It could even lead to overheating and subsequent damage to the IC. An open on the input pin also prohibits any genuine changes on the pin from reaching the input on the chip, with the result that the output fails to respond to those changes. Similarly, an open on the output pin affects the response of the subsequent IC to whose input this particular output is connected.

16.1.1.3 Shorting of Two Pins Other than GND and V_{CC} Pins

This fault forces the affected pins to have the same logic status at all times. For obvious reasons, the output responds incorrectly. Such a situation also leads to shorting of the two pins from where these affected (internally shorted) pins are being fed. The ultimate effect on the performance depends upon the nature of the ICs involved.

16.1.1.4 Failure of the Internal Circuitry of the IC

Failure of the internal circuitry could be anything from damage to a certain active device to increase in the resistance value of a certain on-chip resistor. Bearing in mind the complexity of the internal circuitry of the present-day digital ICs, there could be numerous possibilities. However, the occurrence of such a fault is not very common.

16.1.2 Faults External to Digital Integrated Circuits

The commonly observed faults external to digital ICs include the following:

1. Open circuits.
2. Short circuits.
3. Power supply faults.

16.1.2.1 Open Circuit

An open circuit could be caused by any of a large number of factors, such as a broken track (usually a hairline crack that is very difficult to notice with the naked eye), a dry solder leading to a loose or intermittent connection, a bent or broken pin on the IC, which disallows the signal from reaching that pin, and even a faulty IC socket, where the IC pin does not make a good contact with the socket. Any of the above-mentioned fault conditions would produce a break in the signal path. Such a fault condition can be easily located by switching off the power to the circuit and then establishing the continuity in the suspected areas with the help of a multimeter.

16.1.2.2 Short Circuit

A short circuit could be caused by an improperly etched PCB leading to unetched copper between tracks, solder bridges tending to short two points that are close to each other, such as adjacent pins of an IC, and other similar factors reflecting poor-quality PCB making, wiring and soldering techniques. Such a fault could also be easily located with the help of a multimeter by switching off the power to the circuit.

16.1.2.3 Faulty Power Supply

The third commonly observed fault external to the ICs results from a faulty power supply. There are in fact two commonly observed conditions that generally lead to an apparent power supply fault. One of them could be a catastrophic failure of the power supply that feeds DC voltages to the V_{CC} or V_{DD} pins. The result could be either a complete absence of or a reduction in these DC voltages. The other possible condition could be the overloading of the power supply, which means that the power supply is being asked to deliver a current that is greater than it is designed for. Such a condition is usually due to a fault internal to the IC. In some cases, the fault could be external to the IC too. In such cases it would be good practice to check the power supply and ground status of all the digital ICs being used. An overloading caused by some kind of fault internal to the IC often leads to an increased ripple on the power supply line. Having confirmed such a situation, it would again be good practice firstly to rule out any possibility of a short or a very low resistance path external to the ICs. After that, the ICs could be removed one at a time until the situation is corrected. The IC whose removal restores normalcy is the one that has developed an internal fault. The next obvious step is that of replacing the faulty IC with a fresh one. Sometimes, more than one IC develops internal faults so as to load the power supply. In that case it is necessary to replace all of them to restore normal functioning.

The general guidelines outlined above are applicable to troubleshooting digital circuits using digital ICs of different complexities, from logic gates to counters, registers and arithmetic building blocks. Application of these guidelines to some simple case studies related to troubleshooting of combinational circuits is presented in the following examples.

Example 16.1

Refer to the simple combinational circuit of Fig. 16.1. The logic status of the different input and output pins of the ICs used in this circuit, as observed with the help of a logic probe, is as follows: pin 1 of IC-1 is LOW; pin 2 of IC-1 is pulsing; pin 3 of IC-1 is LOW; pin 4 of IC-1 is HIGH; pin 5 of IC-1 is pulsing; pin 6 of IC-1 is pulsing; pin 1 of IC-2 is indeterminate; pin 2 of IC-2 is pulsing; pin 3 of IC-2 is indeterminate. What in your opinion is the most probable cause of this faulty condition? Give justification wherever required. The ICs used here belong to the 74HC logic family.

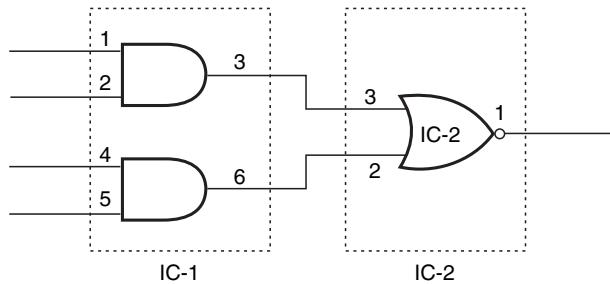


Figure 16.1 Combinational circuit (example 16.1).

Solution

At the outset, the functional status of each one of the building blocks used in this combinational logic circuit is looked at. The upper AND gate is disabled as one of its inputs is observed to have a logic LOW, with the result that its output should be a logic LOW. This is confirmed by the logic probe measurement at pin 3 of this IC. The lower AND gate is enabled as one of its inputs is in the logic HIGH state. Therefore, the output of this gate should be the same as the other input of this gate, which is a pulsed waveform. The output of this gate is a pulsed one, as confirmed by the logic probe measurement at pin 6 of IC-1.

Pin 6 of IC-1 is connected to pin 2 of IC-2. Pin 2 of IC-2 is one of the inputs of a two-input NOR gate. Pin 2 of IC-2 shows the presence of a pulsed waveform, which confirms that it is being properly fed from pin 6 of IC-1. Now, pin 3 of IC-1 is in the logic LOW state, and this is connected to pin 3 of IC-2. Therefore, pin 3 of IC-2 should have shown a logic LOW status. This is, however, not the case, as demonstrated by logic probe measurement. The indeterminate state at pin 3 of IC-2 also manifests itself at pin 1 of IC-2, which is understandable when CMOS ICs are being dealt with.

The indeterminate status of pin 3 of IC-2 only indicates that there is an open circuit somewhere in the path from pin 3 of IC-1 to pin 3 of IC-2. This can be verified with the help of a logic probe and tracing the path and identifying the spot where the genuine logic LOW status changes to an undesired indeterminate status. Remember that CMOS ICs treat floating inputs as indeterminate states.

Example 16.2

Figure 16.2(a) shows the implementation of a two-input multiplexer that is supposed to have the functional table of Fig. 16.2(b). Instead, it is behaving like the functional table of Fig. 16.2(c). The ICs used are from the TTL family. The observations made at different pins of the three ICs used in the circuit are listed in Table 16.1. What is the most probable cause of this faulty behaviour?

Solution

If we look at the logic status of various pins of IC-1, IC-2 and IC-3 for $S = 0$, we find that the inverter in IC-1 is not working properly. Its output should have been \bar{S} and not logic '0'. The two AND gates in IC-2 and the OR gate in IC-3 are functioning as per their respective truth tables. Even the inverter seems to be doing its job when the input is a logic '1'. Such behaviour of the inverter is possible only when the input to this inverter is always a logic '1', irrespective of the logic status of S .

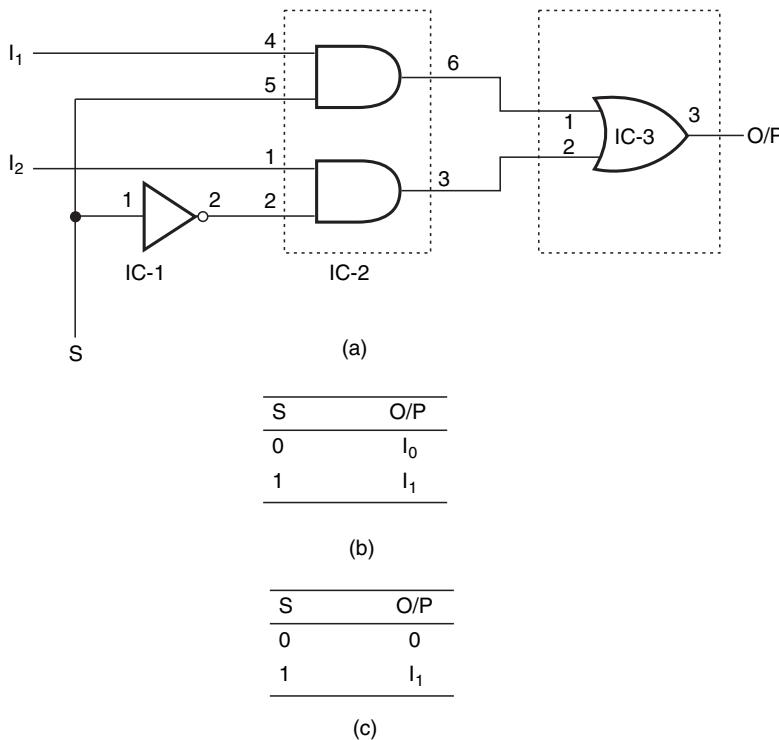


Figure 16.2 Combinational circuit (example 16.2).

Table 16.1 Example 16.2.

Pin/IC	$S = 0$	$S = 1$
Pin 1 (IC-1)	0	1
Pin 2 (IC-1)	0	2
Pin 1 (IC-2)	I_2	I_2
Pin 2 (IC-2)	0	0
Pin 3 (IC-2)	0	0
Pin 4 (IC-2)	I_1	I_1
Pin 5 (IC-2)	0	1
Pin 6 (IC-2)	0	I_1
Pin 1 (IC-3)	0	I_1
Pin 2 (IC-3)	0	0
Pin 3 (IC-3)	0	I_1

Probable reasons for such behaviour are as follows:

1. Pin 2 of IC-1 is internally shorted to GND.
2. Pin 2 of IC-2 is internally shorted to GND.
3. Pin 1 of IC-1 is internally open, which means that it is floating and is therefore treated as a logic HIGH input as the IC belongs to the TTL family.

The first two reasons can be ruled out one by one by checking the continuity between pin 2 of IC-1 and GND and also between pin 2 of IC-2 and GND. If the meter shows no continuity in the two cases, these reasons are ruled out. In such a case, the third reason seems to be the most probable cause.

16.2 Troubleshooting Sequential Logic Circuits

The troubleshooting guidelines for combinational circuits that have been outlined and illustrated in the previous pages with the help of troubleshooting exercises are equally valid in the case of sequential logic circuits such as flip-flops, counters, registers, etc. Faults such as open and short circuits affect all categories of digital building blocks, including both combinational and sequential circuits. However, the effects of open and short circuits in the case of sequential logic devices can be far more serious and difficult to analyse than they would be in the case of logic gates and other combinational building blocks. This is due to the memory characteristics of flip-flops, on account of which the output of a sequential device or circuit depends not only on the present inputs but also on the past inputs. A noise pulse, if large enough in amplitude and duration, could induce a change in the logic status at the output of a logic gate. However, the logic gate would get back to its original status after the noise pulse has vanished. On the other hand, the same noise pulse induced state transition in the case of a flip-flop is permanent.

Let us take the case of a floating input due to an internal or external open circuit. A floating input is highly prone to picking up noise. The most susceptible inputs from the viewpoint of noise pick-up in the case of flip-flops are the CLOCK, PRESET and CLEAR inputs, as these inputs, if activated by noise pick-up due to an internal or external open circuit, can cause the flip-flop to behave erratically. The other possible fault condition is a short circuit at one or more of the inputs of the flip-flop. Again, the symptoms in the case of flip-flops would be different from those in the case of combinational circuits.

Yet another condition that is particularly troublesome in the case of clocked sequential circuits arises from what is known as *clock skew*. Clock skew is basically the difference in the time of arrival of the clock signal at the clock inputs of various sequential devices such as flip-flops comprising a complex synchronous sequential circuit. This time delay, if more than the propagation delay associated with each of the individual devices, could cause serious problems. This can be best explained with the help of a simple illustration. Refer to Fig. 16.3(a). It shows a simple sequential circuit comprising a cascade arrangement of two *D* flip-flops. The outputs Q_1 and Q_2 are initially in the logic '0' state. With the occurrence of LOW-to-HIGH transition of the first clock pulse, Q_1 should go to the logic '1' state, whereas Q_2 should stay in the logic '0' state. However, if for some reason the clock signal reaching the clock input of FF-2 is delayed from the clock input of FF-1 input by more than the propagation delay associated with the individual flip-flops, the Q_2 output would also go to the logic '1' state. This is obvious from the waveforms shown in Fig. 16.3(b). The dotted block in Fig. 16.3(a) represents the clock signal delay. The reasons for this clock skew could be long connecting lines, parasitic capacitance at clock inputs and so on. Since these undesired parameters change with temperature and other circuit conditions, the behaviour of affected devices is usually erratic and unpredictable.

Example 16.3

Refer to the flip-flop circuit of Fig. 16.4. The *D* input to the flip-flop is tied to GND. The *Q* output of the flip-flop is expected to go to the logic '0' state with the application of a clock pulse. However, it does not do so, as shown by the observations recorded by the troubleshooter: pin 1 of IC-1 is in the logic '1' state; pin 2 of IC-1 is in the logic '1' state; pin 3 of IC-1 is in the logic '0' state; pin 2 of IC-2 is in the logic '0' state; pin 3 of IC-2 is pulsing; pin 5 of IC-2 is in the logic '1' state. List various possible causes of occurrence of this fault. Isolate them one by one to arrive at the actual fault.

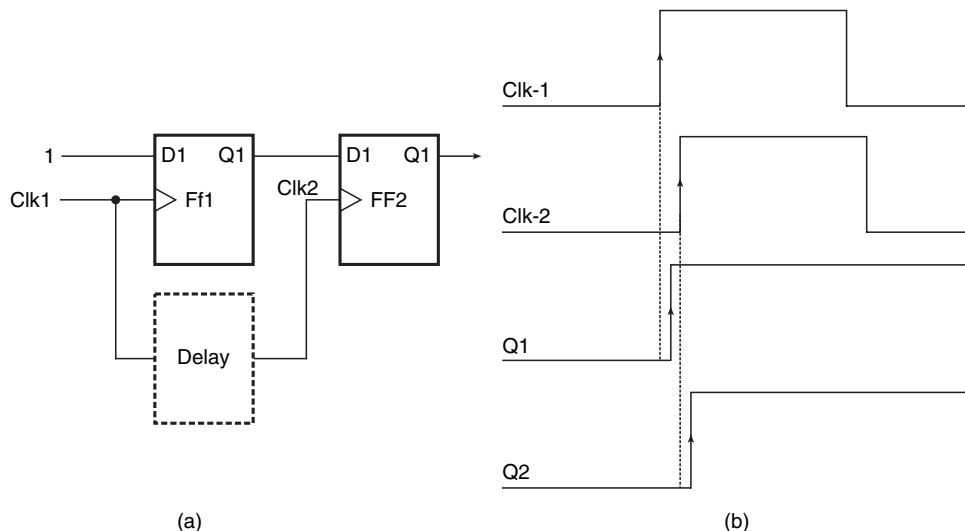


Figure 16.3 Clock skew problem.

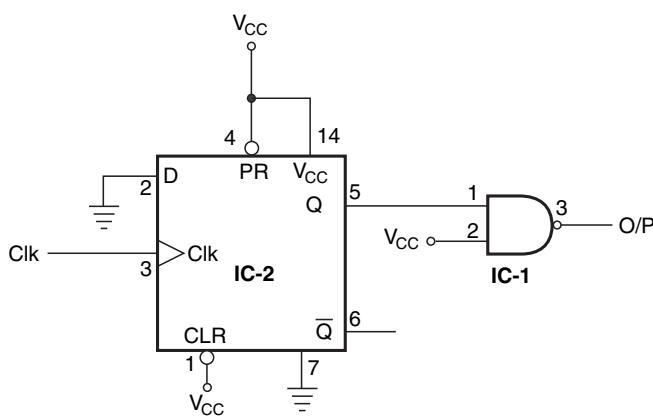


Figure 16.4 Sequential circuit (example 16.3).

Solution

Various possible causes that can lead to the above fault conditions are as follows:

1. Pin 1 of IC-1 is either externally or internally shorted to V_{CC} .
 2. Pin 5 of IC-2 is either externally or internally shorted to V_{CC} .
 3. Pin 4 of IC-2 is either externally or internally shorted to GND.
 4. IC-2 has some kind of internal failure, which stops it from responding to inputs.

5. Pin 6 of IC-2 is externally or internally shorted to GND.

A continuity check can be used to rule out one by one the first, second, third and fifth causes. Remember that, although pin 6 is not used in the circuit, if it is shorted to GND it will force the Q output permanently to go to the logic ‘1’ state owing to the cross-coupling arrangement in the internal structure of the flip-flop. Incidentally, pin 6 in IC-2 happens to be very close to the GND pin, which is pin 7. Even a solder bridge between pin 6 and pin 7 could lead to this. What is important to note is that a troubleshooter may tend to ignore IC pins that are not used, but even those unused pins in the IC can cause faulty conditions. Once the continuity check rules all these possibilities out, the IC can be replaced.

Example 16.4

Figure 16.5 shows a cascaded arrangement of three D flip-flops belonging to the TTL family of digital ICs. The circuit shown here is only a small part of a complex digital circuit. Each of the three flip-flops has a clock input-to-output propagation delay of 15 ns. The expected and observed outputs of the flip-flops for the first few clock cycles are listed in Table 16.2. Although the circuit shown here is that of a three-bit shift counter, the observed outputs are nowhere near to what they should have been in the case of a three-bit shift counter. Identify the possible cause for the observed outputs being different from the expected ones. All flip-flops are observed to be in the logic ‘0’ state just before application of

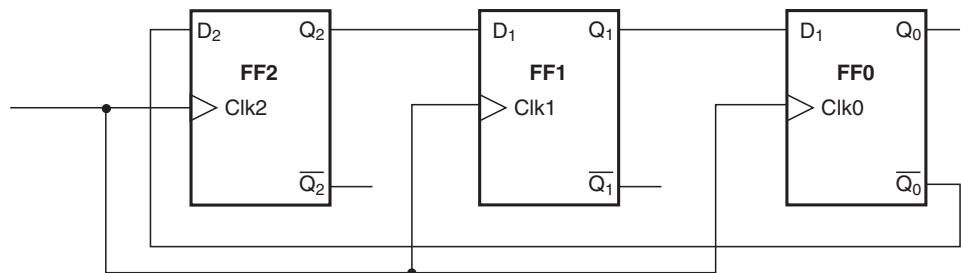


Figure 16.5 Three-bit shift counter (example 16.4).

Table 16.2 Example 16.4.

Clock pulse	Expected output			Actual output		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0
1	1	0	0	1	1	1
2	1	1	0	0	0	0
3	1	1	1	1	1	1
4	0	1	1	0	0	0
5	0	0	1	1	1	1
6	0	0	0	0	0	0
7	1	0	0	1	1	1

the clock signal. The clock signals appearing at the input terminals of the three flip-flops, when seen individually, are observed to be clean and free of any noise content.

Solution

Initially, $Q_2 = Q_1 = Q_0 = 0$ and $D_2 = 1$ as D_2 is fed from $\overline{Q_0}$. Therefore, with the occurrence of the first clock pulse, Q_2 is expected to go to the logic '1' state. Since $D_1 = D_0 = 0$, Q_1 and Q_0 are expected to remain in the logic '0' state. However, Q_2 , Q_1 and Q_0 are observed to make a transition to the logic '1' state. Now this could have been possible if $D_2 = D_1 = D_0 = 1$, which is not the case. This would be remotely possible if there were an external or an internal open at all the D inputs, making them floating inputs. Since the ICs used here are TTL ICs, these floating inputs would be treated as logic HIGH. All this seems to be valid for only the first clock pulse, because, if this were true, the three outputs would subsequently stay in the logic '1' state. Here, all outputs are observed to be toggling. Whether there is any internal or external open or short can be verified with a continuity check using a multimeter.

There is another possibility. As we know, clock skew is a problem that quite often bothers flip-flop timing. Whether or not the fault could possibly be due to the clock skew problem will now be examined. This is not an arbitrary choice. If the statement of the problem is carefully read, it is stated there that the given circuit is only a part of a bigger circuit, and also that clock signals have been observed only individually at the relevant inputs of different flip-flops. It is therefore quite possible that the clock signals at the clock inputs of different flip-flops are not synchronous. If the clock inputs

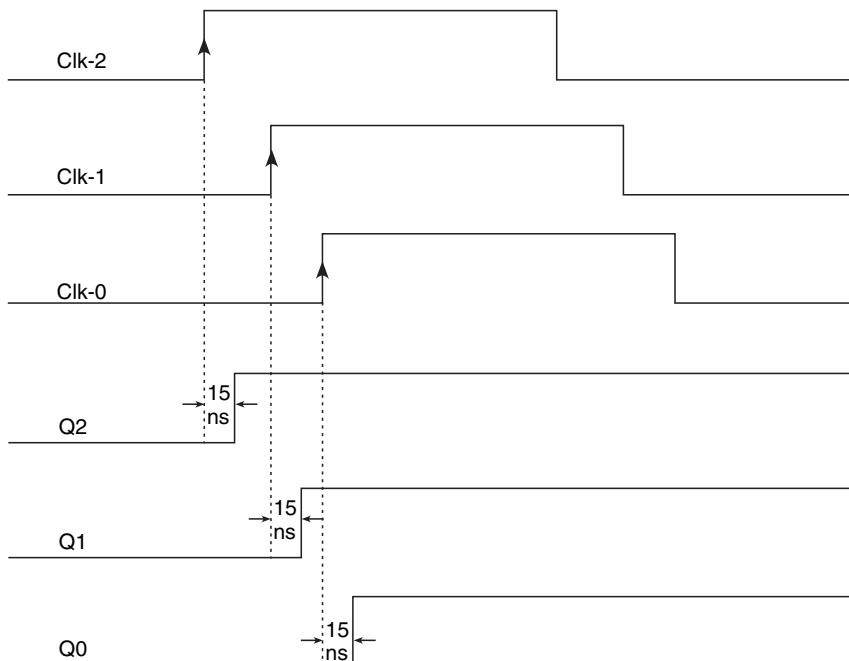


Figure 16.6 Waveforms (troubleshooting exercise 16.4).

to flip-flops FF-2 and FF-1 are examined simultaneously and it is discovered that the clock input to FF-1 is delayed from the clock input to FF-2 by more than 15 ns, FF-1 will make a transition to the logic ‘1’ state with the first clock transition. Similarly, if the clock input to FF-0 is delayed by more than 15 ns from that to FF-1 or by more than 30 ns from that to FF-2, even FF-0 is going to make a transition to the logic ‘1’ state with the first relevant transition of the clock signal. And what is more important is that all other observed outputs for subsequent clock pulses, as shown in Table 16.2, are also valid under these circumstances. The waveforms shown in Fig. 16.6 illustrate how this clock delay can cause a fault condition. Thus, this seems to be the most probable reason for the present fault condition.

16.3 Troubleshooting Arithmetic Circuits

The arithmetic circuits also fall into the category of combinational circuits. Therefore, the troubleshooting tips are similar to those described at length in the previous pages. It would be worth reiterating again that knowledge of the internal structure and functional aspects of the ICs used helps a lot in identifying the reasons for a fault. The following troubleshooting exercise illustrates the point.

Example 16.5

Figure 16.7 shows a four-bit binary adder–subtractor circuit configured around a four-bit parallel binary adder (type number 7483) and a quad two-input EX-OR gate (type number 7486). The arrangement works as an adder when the ADD/SUB input is in the logic ‘0’ state, and as a subtractor when ADD/SUB is in the logic ‘1’ state. The circuit has developed a fault. It is functioning satisfactorily as a subtractor. However, when it is used as an adder, it is observed that the SUM output is not $A + B$ but $A + B + 1$ instead. What do you think is the probable reason for this behaviour?

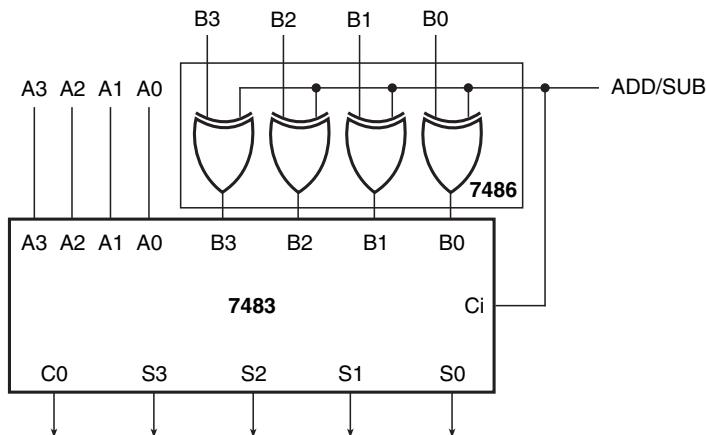


Figure 16.7 Adder–subtractor circuit (example 16.5).

Solution

- Since the circuit is functioning properly as a subtractor (when ADD/SUB = 1), this implies that:
 - (a) IC 7486 is functioning properly;
 - (b) the C_i input is in the logic '1' state.
- The present fault would occur only if the C_i input were in the logic '1' state, even when ADD/SUB = 0. This is possible in the case of either of the following two situations:
 - (a) There is an external open between the ADD/SUB input and the C_i input. This would make C_i a floating input, which would be treated as a logic '1' in a TTL IC.
 - (b) There is an internal open at the C_i input, which would have the same end result.
- The fault can be isolated with the help of a continuity check.

16.4 Troubleshooting Memory Devices

This section outlines the general procedure used for testing and troubleshooting memory devices. As will be seen in the paragraphs to follow, troubleshooting memory devices is far more complex than troubleshooting other digital building blocks. The procedure outlined earlier for digital building blocks such as logic gates, flip-flops, counters, registers, arithmetic circuits, etc., is not valid for testing memory devices such as RAM and ROM. One thing that is valid is that it is equally important fully to understand the operation of the system before attempting to troubleshoot. You must also remember that there is a lot of digital circuitry outside the memory device that is a part of the overall memory system. This may include a decoder circuit and some combinational logic.

16.4.1 Troubleshooting RAM Devices

The most common method of testing a RAM system involves writing known the pattern of 0s and 1s to each of the memory locations and then reading them back to see whether the location stored the pattern correctly. This way, both READ and WRITE operations are checked. One of the most commonly used patterns is the 'checkerboard pattern' where all memory locations are tested with a 01010101 pattern and then with a 10101010 pattern. There are many more patterns that can be used to check various failure modes in RAM devices. No check, however, guarantees 100 % accuracy. A chip that passes a checkerboard test may fail in another test. But if the chip fails in the checkerboard test, it is certainly not good.

RAM check is performed automatically. In the majority of computers and microprocessor-based systems, every time the system is powered, the CPU runs a memory-test program that is stored in the ROM. The operator can also execute this memory-test routine on request. The system displays some message after the test is over. After that, remedial action can be initiated.

16.4.2 Troubleshooting ROM Devices

ROM devices cannot be checked by writing and reading back known patterns of 0s and 1s, as was done in the case of RAM devices. ROM is a 'read only memory' device and its testing should basically

involve reading the contents of each location of the ROM and then comparing them with what it is actually supposed to contain.

ROM testing is done with the help of a special instrument that can be used to read the data stored in each location of the ROM. It cannot be tested, like a RAM, by writing some pattern of 0s and 1s and then reading them back. One of the methods is to read data in each location and produce a listing of those data for the user to compare with what the ROM is actually supposed to store. But, of course, the process becomes highly cumbersome for large-capacity ROM chips.

Another approach is to have a reference ROM plugged into the test instrument along with the test ROM. The instrument reads data in each of the locations on the test ROM and then compares them with the data stored on the corresponding locations of the reference ROM.

Yet another method is to use a CHECKSUM. Checksum is a code that is stored in the last one or two locations of the ROM. It is derived from the addition of different data words stored in different locations of the ROM under test. For instance, if the data words stored in the first three locations are 11001001, 10001110 and 11001100, then the checksum up to this point will be 00100011. When the test instrument reads data in the test ROM, it creates its own checksum. It compares the checksum with the one already stored in the test ROM. If the two match, the ROM may be considered to be a good one. We have used the word ‘may’ because even wrong data can possibly lead to a correct checksum. However, if the checksums do not match, it is definitely a faulty ROM.

16.5 Test and Measuring Equipment

As outlined at the beginning of the chapter, the test and measuring instruments discussed in this part of the chapter are not only the ones that a digital system troubleshooter or analyser has generally to make use of; some of the instruments described here have an internal hardware dominated by digital technology and its advances. The test equipment covered at length in the following pages include the digital multimeter, digital oscilloscope, logic probe, logic analyser, frequency counter, synthesized function generator and arbitrary waveform generator. Computerized instrumentation and equipment-computer interface standards are discussed towards the end of the chapter.

16.6 Digital Multimeter

In a *digital multimeter*, the analogue quantity to be measured (current, voltage, resistance) is firstly transformed into an equivalent voltage if the parameter to be measured is current or resistance. The transformed analogue voltage is then digitized using an A/D converter (ADC). To be more precise, the analogue voltage is converted into a pulse train whose frequency depends upon the magnitude of the voltage. The pulses are counted over a known gating period in a counter. The counter outputs are decoded and displayed. The displayed count represents the magnitude of the parameter under measurement.

In another approach that is also in common use the input analogue voltage is compared with a ramp from a ramp generator. The comparator generates a gating pulse whose width equals the time interval between the ramp amplitude rising from zero to the analogue voltage under measurement. The counter in the ADC counts clock pulses of a known frequency over this gating interval, and the counter count is decoded and displayed. Thus, while in the former method there is a voltage-to-frequency (V/F) conversion and the equivalent frequency representing the analogue voltage is counted over a fixed gating interval, in the latter method a fixed frequency is counted over a variable gating interval, with the gating interval being proportional to the analogue voltage. Different techniques of analogue-to-digital conversion have been discussed in detail in Chapter 12 on data conversion circuits.

16.6.1 Advantages of Using a Digital Multimeter

The digital multimeter has the advantages of offering unambiguous display with no allowance for any human error, improved accuracy ($\pm 0.1\%$ as against $\pm 3\%$ in analogue meters) and improved resolution ($+0.1\%$ as against 1% in analogue meters). Other advantages include easy incorporation of features such as *autoranging*, automatic polarity and diode/transistor test and so on. The cost advantage that used to exist in favour of analogue meters has narrowed down to a small amount with advances in IC technology. Digital multimeters are fast replacing analogue meters even for routine measurements. However, analogue meters are relatively immune to noise and are preferred in an electrically noisy environment.

16.6.2 Inside the Digital Meter

Figure 16.8 shows the schematic arrangement of a typical digital meter. The signal scalar at the input is basically an attenuator/amplifier block and is partly used for range selection function. In autoranging meters, the input signal level is sensed on application of the input signal, and the signal scalar gain is selected accordingly. The signal conditioner generates a DC voltage proportional to the input signal. The ADC employed is usually the integrating-type ADC, single slope or dual slope, with the latter being the preferred one because of its higher accuracy, insensitivity to changes in integrator parameters and low cost. All the building blocks depicted in Fig. 16.8, except for the display, are available on a single chip. ICL 7106/7107 is an example.

16.6.3 Significance of the Half-Digit

Digital multimeters (DMMs) invariably have a display that has an additional half-digit. We have $3\frac{1}{2}$ -, $4\frac{1}{2}$ - and $5\frac{1}{2}$ -digit digital multimeters rather than 3-, 4- and 5-digit multimeters. While the usually so-called full digits can display all digits from 0 to 9, a half-digit can display either a '0' or a '1'. The

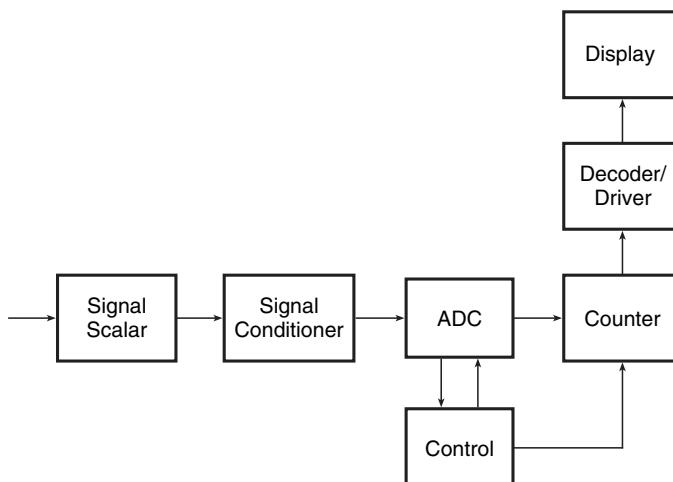


Figure 16.8 Block schematic of a digital meter.

addition of a half-digit in the MSB position of the display preserves the resolution of the multimeter up to a higher range. For instance, a three-digit multimeter has a resolution of 0.1 V up to 99.9 V. A $3\frac{1}{2}$ -digit meter with practically no additional hardware would give you a resolution of 0.1 V up to 199.9 V. This increase in resolution range comes with the addition of one additional seven-segment display and no change in hardware complexity. The display resolution is also sometimes expressed in terms of counts. The $3\frac{1}{2}$ -digit DMM has a 2000 count resolution. DMMs with a 4000 count resolution, referred to as $3\frac{3}{4}$ -digit meters, are also commercially available. These meters will also have four seven-segment displays but have some additional hardware.

Digital multimeters are made in a large variety of sizes, shapes and performance specifications, ranging from pen-type $3\frac{1}{2}$ -digit DMMs to $7\frac{1}{2}$ -digit high-resolution benchtop versions. Handheld versions are available, typically up to $4\frac{1}{2}$ -digit resolution. The majority of them have an in-built diode test, transistor test and continuity check features. Some of them even offer L-C measurement and frequency measurement without any significant change in price. Figure 16.9 shows a photograph of one such multimeter (the Fluke 115 multimeter). It has a 6000 count display and an in-built continuity check, diode test, frequency measurement, capacitance measurement, etc., in addition to conventional functions. Figure 16.10 shows a photograph of a high-end benchtop version of a digital multimeter (Fluke 8845A).

Example 16.6

The specification sheet of a certain $3\frac{1}{2}$ -digit digital multimeter lists its display to be a 4000 count display. Determine the resolution offered by the multimeter for the following measurements:

- the maximum DC voltage that can be measured with a resolution of 0.1 V;
- the maximum resistance value that can be measured with a resolution of 1 Ω .
- the maximum DC current that can be measured with a resolution of 10 μA .



Figure 16.9 Handheld digital multimeter. Reproduced with permission of Fluke Corporation.



Figure 16.10 Benchtop digital multimeter. Reproduced with permission of Fluke Corporation.

Solution

- (a) 399.9 V;
- (b) 3999 Ω ;
- (c) 39.99 mA.

16.7 Oscilloscope

After the multimeter, the oscilloscope is the most commonly used item of electronic test equipment. Be it the electronics industry or a research laboratory, the oscilloscope is an indispensable test and measurement tool for an electronics engineer or technician. Most of us regard the oscilloscope as an item of equipment that is used to see pulsed or repetitive waveforms. However, very few of us are familiar with the actual use of the multiplicity of front-panel controls on the oscilloscope and the potential that lies behind the operation of each one of these controls.

With the arrival of the digital storage oscilloscope (DSO), the functional potential of oscilloscopes has greatly increased. The digital storage oscilloscope enjoys a number of advantages over its analogue counterpart.

16.7.1 Importance of Specifications and Front-Panel Controls

It is very important to have a clear understanding of the performance specifications of oscilloscopes. The specification sheet supplied by the manufacturer contains scores of specifications. Each one of them is important in its own right and should not be ignored. Although some of them explain only the broad features of the equipment and do not play a significant role as far as measurements are concerned, these are important when it is required to choose one for a given application. In fact, the performance specifications of an oscilloscope and the operational features of its front-panel controls cannot be considered in isolation. One complements the other. Not only does the correct interpretation of specifications help in the selection of the right equipment for an intended application, their appreciation is almost a prerequisite to a proper understanding of the functional potential of front-panel controls.

16.7.2 Types of Oscilloscope

Technology is often the single most important criterion forming the basis of oscilloscope classification. Different types of oscilloscope include analogue oscilloscopes, CRT storage type analogue oscilloscopes, digital storage oscilloscopes and sampling oscilloscopes. Digital storage oscilloscopes and sampling oscilloscopes are often clubbed together under digital oscilloscopes. Analogue oscilloscopes are briefly described in the following paragraphs. This is followed up by a detailed description of digital oscilloscopes.

16.8 Analogue Oscilloscopes

The analogue oscilloscope displays the signal directly and enables us to see the waveform shape in real time. The signal update rate in an analogue oscilloscope is the fastest possible as there is only the beam retrace timing and the trigger rearm between two successive sweeps. Consequently, an analogue oscilloscope has a much higher probability of capturing the desired event than any other type of oscilloscope. Analogue oscilloscopes find wide application for viewing both repetitive and single-shot events up to a bandwidth of about 500 MHz. Analogue oscilloscopes do not give a desirable display when viewing very low-frequency repetitive signals or single-shot events. In such cases, the display is nothing but a bright dot moving slowly across the screen to trace the waveform. Such waveforms are not at all convenient to analyse and need some kind of photographic memory.

16.9 CRT Storage Type Analogue Oscilloscopes

A CRT storage type analogue oscilloscope overcomes this problem by using a special type of CRT. In one such type, the phosphor dots have higher persistence. As a result, the moving dot leaves behind a visible trail as it sweeps across the screen, even at much lower sweep speeds. There are two main types of storage mode currently in use for these oscilloscopes: the bistable storage mode, which is capable of storing signals for many hours, and the more popular variable-persistence storage mode, which can store signals for a maximum of 10 min. The majority of commercially available CRT storage oscilloscopes have the option of both the above-mentioned storage modes.

The CRT storage type oscilloscope is an excellent choice for slowly changing signals. As the writing rate is faster than that of the conventional analogue oscilloscopes, it is extremely good for viewing fast transient events. It can be used to store both repetitive and single-shot signals having a bandwidth of up to 500 MHz or so. Oscilloscope type 7934 from Tektronix, for instance, has a bandwidth of 500 MHz and a maximum writing speed of 4000 cm/ms. Even handheld versions of these scopes with a reasonably good writing speed are available. Analogue storage oscilloscope technology is fast being replaced by digital storage oscilloscope technology owing to the far superior performance features of the latter.

16.10 Digital Oscilloscopes

In a digital oscilloscope, the signal to be viewed is firstly digitized inside the scope using a fast A/D converter. The digitized signal is stored in a high-speed semiconductor memory to be subsequently retrieved from the memory and displayed on the oscilloscope screen. There are two digitizing techniques, namely real-time sampling and equivalent-time sampling. The digital storage oscilloscopes (DSOs) use real-time sampling, as shown in Fig. 16.11, so that they can capture both repetitive and

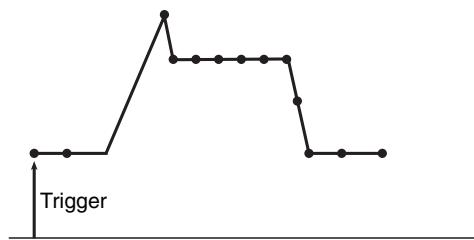


Figure 16.11 Real-time sampling.

single-shot signals. In digital storage oscilloscopes, the digitizer samples the entire input waveform with a single trigger. Sampling oscilloscopes use equivalent-time sampling and are limited to capturing repetitive signals. Some digital storage oscilloscopes also use equivalent-time sampling to extend their useful frequency range for capturing repetitive signals. The equivalent-time sampling technique is thus applicable to only stable repetitive signals and can be implemented in at least three different ways, namely sequential single-sample, sequential sweep and random interleaved sampling (RIS).

In the *sequential single-sample technique* (Fig. 16.12), the digitizer acquires a single sample with the first trigger pulse. It then waits for the second trigger, and, on receipt of the second trigger, a time delay equal to the reciprocal of the desired sampling rate is executed and then the second sample is acquired. The trigger-to-acquisition delay is incremented by the desired intersample period Δt for each subsequent acquisition. The resulting capture has thus an equivalent sample rate of $1/\Delta t$. Clearly, this method is slow, as N trigger cycles would be needed to gather N samples, and the scopes using this type of digitizing technique cannot provide real-time operation.

In *sequential sweep equivalent-time sampling* (Fig. 16.13), a sweep of samples spanning the desired display time range is acquired for each trigger. Here, N samples are acquired in M trigger cycles, where $N = kM$. On receipt of each trigger, k sequential samples are acquired at sample rate f_s . These are stored in every M th location of the acquisition memory allocated for N samples. k samples of the first sweep are acquired directly on receipt of the trigger. Subsequent sweeps have an increasing delay between trigger receipt and sweep initiation, with the delay increment being equal to $1/Mf_s$ with reference to trigger detection in order to give an apparent sample rate of Mf_s .

The *random interleaved sampling* (RIS) technique uses a memory distribution scheme that is philosophically similar to that of sequential sweep equivalent-time sampling, with the difference that the samples are random with respect to the trigger. Sampling in this case occurs on both sides of

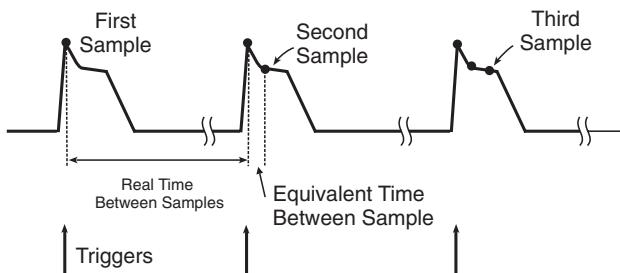


Figure 16.12 Sequential single-sample technique.

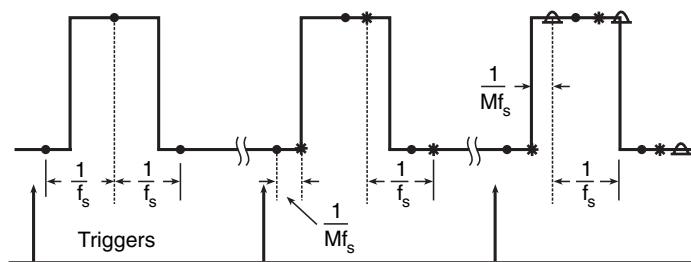


Figure 16.13 Sequential sweep equivalent-time sampling.

trigger points, which gives this technique a ‘pretrigger view’ capability not available in the first two equivalent-time sampling techniques, as both methods gather signals only following the receipt of a trigger.

If we wanted to view a 1 GHz signal, the sweep speed requirement would be enormous. Even if we were successful in achieving this high speed, the beam would be almost invisible. We have often noticed that, as the time-base setting is made faster, we are forced to adjust the intensity control to maintain an acceptable intensity level setting. Another major problem in designing a real-time oscilloscope for viewing very high-frequency signals (in the GHz range) is the difficulty in building such a high bandwidth in the vertical amplifier. A sampling oscilloscope using any of the equivalent-time sampling techniques outlined above is an answer to all these problems. In such scopes it is not imperative to take a sample or a group of samples from each cycle of the signal to be viewed. The next adjacent sample or group of samples may be 10 000 cycles away. As a result, the bandwidth of the vertical amplifier can afford to be much lower than the frequency of the signal.

Another type of sampling oscilloscope, although not very common in use, is the analogue sampling oscilloscope, where a conventional sample/hold circuit consisting of an electronic switch and a capacitor is used for signal acquisition (Fig. 16.14). It can be used to view high-frequency repetitive signals in nonstorage mode, unlike the digital sampling scopes where the signal is sampled digitally and then stored in semiconductor memory for subsequent retrieval. It can also be used for viewing high-frequency repetitive signals in storage mode, although not in real time (Fig. 16.15).

Digital storage oscilloscopes are also available in a large variety of sizes, shapes, performance features and specifications. Battery-operated, handheld digital storage oscilloscopes with a bandwidth as high as 200 MHz are common (Fig. 16.16). The digital phosphor oscilloscope (DPO) is a big step forward in DSO technology. It captures, stores, displays and analyses, in real time, three dimensions of signal information, i.e. amplitude, time and distribution of amplitude over time. This third dimension

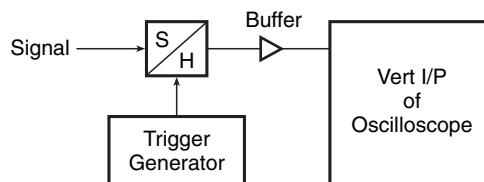


Figure 16.14 Analogue sampling oscilloscope (nonstorage mode). Reproduced with permission of Fluke Corporation.

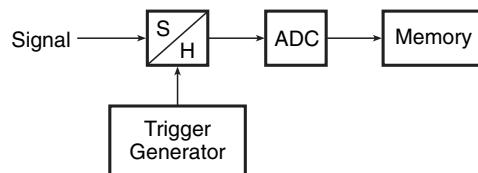


Figure 16.15 Analogue sampling oscilloscope (storage mode). Reproduced with permission of Fluke Corporation.



Figure 16.16 Handheld digital storage oscilloscopes. Reproduced with permission of Fluke Corporation.

offers the advantage of interpretation of signal dynamics, including instantaneous changes and the frequency of occurrence displayed in the form of quantitative intensity information.

16.11 Analogue Versus Digital Oscilloscopes

Almost all oscilloscopes available today use one or a combination of the technologies discussed above. Each technology has its own benefits and shortcomings. While signal manipulation and its consequent benefits are the strong point of the digital technology, extremely fast update rates coupled with low cost is a feature associated with analogue scopes. In fact, many state-of-the-art oscilloscopes are not simply analogue or digital. They offer advantages of both technologies.

16.12 Oscilloscope Specifications

Although oscilloscopes are characterized by scores of performance specifications, not all of them are important. Important specifications of analogue and digital oscilloscopes are briefly described in the following paragraphs.

16.12.1 Analogue Oscilloscopes

Key specifications include bandwidth (or rise time), vertical sensitivity and accuracy. Other features such as triggering capabilities, display modes, sweep speeds, etc., are secondary in nature.

16.12.1.1 Bandwidth and Rise Time

The bandwidth and rise time specifications of an oscilloscope are related to one another. Each can be calculated from the other. Bandwidth (in MHz) = $350/\text{rise time (in ns)}$.

Bandwidth is the most important specification of any oscilloscope. It gives us a fairly good indication of the signal frequency range that can be viewed on the oscilloscope with an acceptable accuracy. If we try to view a signal with a bandwidth equal to the bandwidth of the oscilloscope, the measurement error may be as large as 40 % (Fig. 16.17). As a rule, the oscilloscope bandwidth should be 3–5 times the highest frequency one is likely to encounter in order to keep the measurement error to less than 5 %.

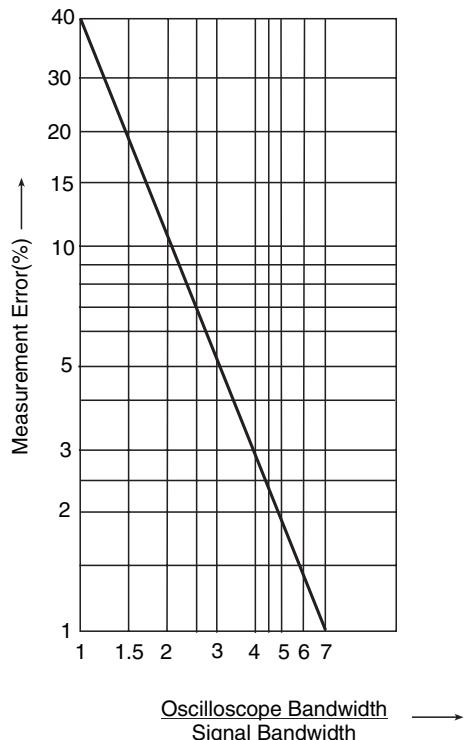


Figure 16.17 Measurement error as a function of oscilloscope bandwidth.

16.12.1.2 Vertical sensitivity

The vertical sensitivity specification tells us about the minimum signal amplitude that can fill the oscilloscope screen in the vertical direction. A 5 mV/div sensitivity is quite common. Oscilloscopes with a sensitivity specification of 1 mV/div are also available. Sensitivity and bandwidth are often trade-offs. Although a higher bandwidth enables us to capture high-frequency signals, there is a good possibility of unwanted high-frequency noise being captured if the oscilloscope has a higher sensitivity too. That is why most of the high-sensitivity scopes have bandwidth limit controls to enable a clear view of low-level signals of moderate frequencies. It is also important that the oscilloscope we choose has an adequate V/div range to make possible a full-screen or near-full-screen display for a wide range of signal amplitudes.

16.12.1.3 Accuracy

The accuracy specification indicates the degree to which our measurement conforms to a true and accepted standard value. An accuracy of $\pm 1 - 3\%$ is typical. Almost all oscilloscopes are provided with a $\times 5$ magnification in the V/div selector switch. This alters the nominal vertical deflection scale from say 5 mV/div–5 V/div to 1 mV/div–1 V/div. It may be mentioned here that the accuracy suffers with the magnifier pull. Most of the manufacturers list accuracy specifications separately for the two cases for the oscilloscopes manufactured by them.

16.12.2 Analogue Storage Oscilloscope

With the CRT storage-type oscilloscope, the stored *writing speed* is usually the main criterion for choosing the instrument. The speed of a CRT storage scope depends on the speed of the input signal (signal frequency) and the size of the trace it draws.

16.12.3 Digital Storage Oscilloscope

Just like an analogue scope, the specification sheet of a digital oscilloscope contains scores of specifications that at first sight may appear quite confusing. A closer look at these specifications, particularly the decisive ones, will make one appreciate the performance capabilities of digital oscilloscopes. The real strength of a digital oscilloscope lies in the following specifications: bandwidth, sampling rate, vertical resolution, accuracy and acquisition memory.

16.12.3.1 Bandwidth and Sampling Rate

The *bandwidth* is an important specification of digital oscilloscopes, just as it is for analogue oscilloscopes. The bandwidth, which is primarily determined by the frequency response of input amplifiers and filters, must exceed the bandwidth of the signal if the sharp edges and peaks are to be accurately recorded.

The *sampling rate* is another vital digital scope specification. In fact, the sampling rate determines the true usable bandwidth of the scope. While the bandwidth is associated with the analogue front end of the scope (amplifiers, filters, etc.) and is specified in Hz, the sampling rate is associated with the digitizing process and, if it is not adequate, degrades the bandwidth. A clear understanding of sample rate specification is thus important when it comes to establishing the adequacy of a particular sample

rate to achieve a given bandwidth. Digital oscilloscope specification sheets often contain two sample rates, one for single-shot events and the other for repetitive signals. In some cases, both repetitive and single-shot events are sampled at the same rate, although the bandwidth capability of the oscilloscope for the two cases is different. It is lower in the case of single-shot events.

Theoretically, the Nyquist criterion holds true, and this criterion states that at least two samples must be taken for each cycle of the highest input frequency. In other words, the highest input frequency (also called the Nyquist frequency) cannot exceed half the sample rate. Given this condition, a $\sin x/x$ interpolation algorithm can exactly reproduce a digitized signal. An interpolation algorithm is the mathematical function used by an oscilloscope to join two successive sample points while reconstructing the signal. The $\sin x/x$ interpolation has a tendency to amplify noise in the signal, particularly when each cycle is sampled only twice. With $(\sin x/x)$ interpolation, four samples per cycle are found to be quite adequate. The additional sample points effectively enhance the signal-to-noise ratio for $\sin x/x$ interpolation. With straight-line interpolation, at least ten samples are required per cycle for good results.

For repetitive signals, however, even a smaller sample rate does the job, as explained in the case of sampling oscilloscopes. Thus, it becomes important to look into the sample rate specification together with the interpolation algorithm used. For instance, in a digital storage oscilloscope with a single-shot sample rate of 400 MS/s (where MS stands for megasamples), using the $\sin x/x$ interpolation technique can give us a single-shot bandwidth of 100 MHz, while the same sample rate will provide a bandwidth of only 40 MHz if a straight-line interpolation algorithm is used instead. Thus, the single-shot bandwidth capability of a digital storage oscilloscope must always be gauged by its single-shot sample rate. The sample rate in samples per second should be at least twice the highest frequency component or 4 times the highest frequency component for good results, or anywhere between 2 and 4, assuming $\sin x/x$ interpolation. For repetitive signals, if it is not a real-time DSO, the sampling rate could be smaller.

16.12.3.2 Memory Length

Memory length is a vital digital oscilloscope specification and should not be considered to be an insignificant one. Not only does it affect the sample rate and consequently the single-shot bandwidth, longer memories also have many more peripheral benefits. The sample rate as quoted by the manufacturer always refers to the maximum digitizing rate attainable in single-shot mode. Interestingly, the quoted sample rate figure does not hold true for the entire range of time-base settings. For a given memory length, the attainable sample rate is observed to decrease as the time base is made slower. Some manufacturers offer record length, which is nothing but the size of the memory used while displaying the signal. Suppose a particular DSO has a memory length of 1K and a quoted sample rate specification of 100 MS/s. In the limit when the record length equals the memory length, we can store approximately 1000 samples. At the given sample rate, the displayed waveform will cover a time span of 10 ms, i.e. a time-base setting of 1 ms/div, if the waveform is to cover the full screen in the horizontal direction. If the time-base setting is changed to 10 ms/div, the effective sample rate would be limited to only 10 MS/s, thus reducing the single-shot bandwidth. The only method to maintain the sample rate at the quoted value for a larger time-base setting range is to have a longer acquisition memory. The effect of memory length on single-shot bandwidth as a function of time-base setting is expressed by

$$\text{Sample rate} = \text{memorylength}/(10 \times \text{time - base setting})$$

The '10' is the total number of divisions in the horizontal direction.

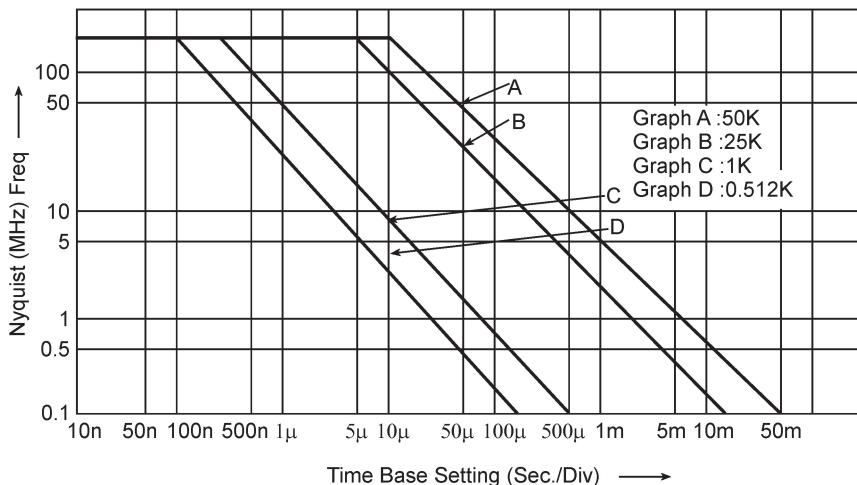


Figure 16.18 Effective sampling rate – time-base setting graph.

Figure 16.18 shows the changes in sample rate as a function of time-base setting for digital oscilloscopes of different memory lengths. Given two oscilloscopes with identical sample rate and single-shot bandwidth specifications, the one with the longer acquisition memory has a decisive edge. Hence, it must be accorded due importance when choosing one to meet your requirements. For a given time resolution, a longer memory enables events of longer duration to be recorded. For instance, a DSO with a 1K memory can record a 1 s transient with a time resolution of 1 ms, whereas a DSO with a 10K memory can record a 10 s long event with the same time resolution. In other words, for the same transient duration, longer memories give enhanced time resolution. Long memories also help in acquiring hard-to-catch signals and also minimize signal reconstruction distortion.

16.12.3.3 Vertical Accuracy and Resolution

The accuracy specification tells us how closely the measurement matches the actual value. The accuracy of a DSO is affected by various sources of error, including gain and offset errors, differential nonlinearity, quantization error and so on. The quantization error indirectly indicates vertical resolution, i.e. uncertainty associated with any reading or the ability of the oscilloscope to see small changes in amplitude measurements. Choosing a scope with fewer than eight bits of resolution is not recommended. Resolution specification must not be considered in isolation from accuracy specification. For instance, more than eight bits of resolution is meaningless when the overall accuracy itself is $\pm 1\%$. An eight-bit resolution gives a $\pm 0.4\%$ uncertainty, which is fairly acceptable if the overall accuracy is $\pm 1\%$, as can be seen from Table 16.3. Also, digital oscilloscopes with more than seven bits of resolution can resolve signal details better than visual measurements made with analogue oscilloscopes.

To sum up our discussion on the available oscilloscope types and the selection criteria for choosing the right one, it can be said that both analogue and digital oscilloscopes have their advantages and shortcomings. The suitability of a particular type must always be viewed in terms of intended application. Although digital oscilloscopes can perform many functions that analogue versions cannot, analogue oscilloscope technology, too, has reached high performance standards. It is important to

Table 16.3 Uncertainty of an oscilloscope as a function of the number of bits.

Number of bits	Uncertainty (%)
6	1.6
7	0.8
8	0.4
9	0.2
10	0.1
11	0.05
12	0.02

understand the critical specifications of each type and then decide whether it fits an intended application. The key specifications to look for in analogue scopes are bandwidth, vertical sensitivity and accuracy, whereas the strength of a digital oscilloscope must be ascertained from its bandwidth, sample rate, vertical resolution, accuracy and memory length.

16.13 Oscilloscope Probes

The oscilloscope probe acts as a kind of interface between the circuit under test and the oscilloscope input. The signal to be viewed on the oscilloscope screen is fed to the vertical input (designated as the Y input) of the oscilloscope. An appropriate probe ensures that the circuit under test is not loaded by the input impedance of the oscilloscope vertical amplifier. This input impedance is usually $1\text{ M}\Omega$, in parallel with a capacitance of $10\text{--}50\text{ pF}$. The most commonly used general-purpose probes are the 1X, 10X and 100X probes. These probes respectively provide attenuation by factors of 1 (i.e. no attenuation), 10 and 100. That is, if we are measuring a 10 V signal with a 10X probe, the signal actually being fed to the oscilloscope input will be 1 V . 10X and 100X probes are quite useful for measuring high-amplitude signals. Another significant advantage of using these probes is that the capacitive loading on the circuit under test is drastically reduced.

Refer to the internal circuit of the 10X probe as shown in Fig. 16.19. The RC time constant of the probe equals the input RC time constant of the oscilloscope. Since the resistance of the probe is 9 times the input resistive component of the oscilloscope, in order to provide attenuation by a factor of 10, the probe capacitance has got to be smaller than the input capacitance of the scope by the same amount. As a result, the circuit under test with a 10X probe will never see a capacitance of more than 5 pF .

16.13.1 Probe Compensation

The probe is compensated when its RC time constant equals the RC time constant of the oscilloscope input. With this, what we see on the screen of the scope is what we are trying to measure independent of the frequency of the input signal. If the probe is not properly compensated, the signal will be attenuated more than the attenuation factor of the probe at higher frequencies owing to reduction in the effective input impedance of the vertical input of the scope.

To check for probe compensation, the probe can be used to see the calibration signal (the CAL position on the front panel) available on the oscilloscope. If the probe is properly compensated, the

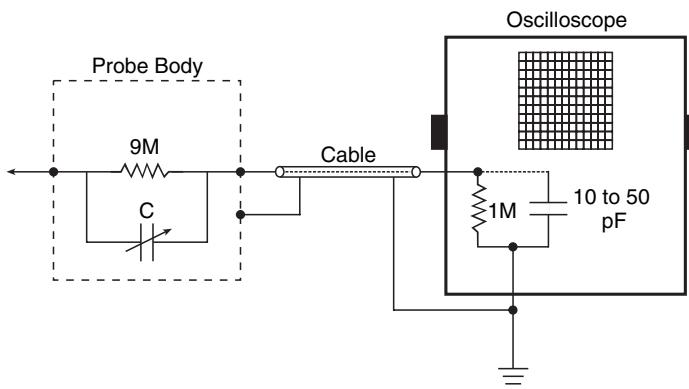


Figure 16.19 Internal circuit of 10X probe.

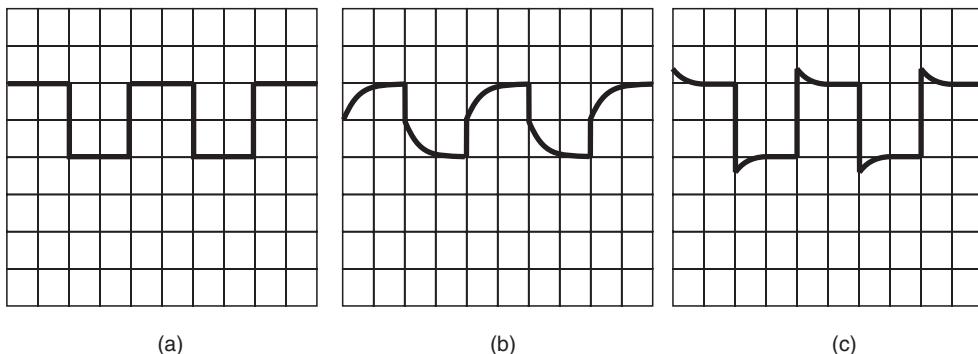


Figure 16.20 Probe compensation.

CAL signal will appear in perfect rectangular shape [Fig. 16.20(a)] with no rounding-off of edges [Fig. 16.20(b)] or any spikes on fast transitions [Fig. 16.20(c)]. Rounding-off of edges indicates too little a probe capacitance, while spikes indicate too large a probe capacitance. The probe capacitance can be adjusted by turning a screw or rotating the probe barrel after loosening the locking nut (in some probes) to get a perfect calibration signal.

16.14 Frequency Counter

The most basic function of a frequency counter is the measurement of an unknown frequency. Modern frequency counters, however, offer much more than just frequency measurement. Other related parameters such as the *time period*, which is the reciprocal of frequency, the *time interval* between two events and the *totalize count*, which is nothing but the cumulative count over a known period, are other functions that are available with present-day frequency counters. These instruments, offering a variety of measurement options, are usually referred to as *universal counters*.

16.14.1 Universal Counters – Functional Modes

The functions available with modern universal counters, other than measurement of an unknown frequency, are time interval measurement, period, time interval average, totalize, frequency ratio A/B, phase A relative to B and pulse width.

16.14.1.1 Time Interval Measurement

This mode measures the time that elapses between the occurrence of two events. One of the events, called the start signal, is usually fed into one of the channels, while the other, called the stop signal, feeds the second channel. The resolution of measurement is typically 10 ns or better. A typical application of this measurement mode is in determination of the propagation delay in logic circuits. Variations of this mode can be used to measure pulse width and rise/fall times.

16.14.1.2 Time Interval Average

This mode can be used to improve the measurement resolution in the time interval measurement mode for a given clock frequency. The resolution improves as the square root of the number of measurements. That is, an average of 100 measurements would give a 10-fold improvement in resolution.

16.14.1.3 Period

In this mode, the time period of the input signal is measured by counting clock pulses between two successive leading or trailing edges of the input signal. Again, the period average function can be used to improve upon the measurement resolution for a given clock. For instance, if the measurement were done for 100 periods instead of one period for a given clock frequency, the measurement resolution would also improve by a factor of 100.

16.14.1.4 Totalize

The totalize mode gives a cumulative count of events over a known time period.

16.14.1.5 Frequency Ratio A/B

This gives the ratio of the frequencies of signals fed to the A and B channels. This feature can be used to test the performance of prescalers and frequency multipliers.

16.14.1.6 Phase A Relative to B

This compares the phase delay between signals with similar frequencies.

16.14.2 Basic Counter Architecture

Figure 16.21 shows the architecture of a frequency counter when it is being used in the frequency measurement mode. The oscillator section, comprising a crystal-based oscillator and a frequency divider

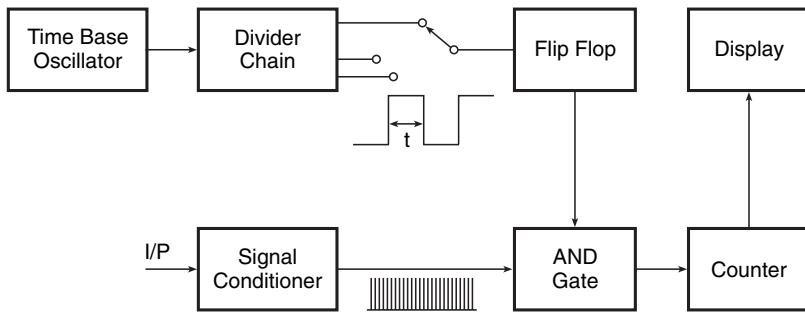


Figure 16.21 Counter architecture. Reproduced with permission of Fluke Corporation.

chain, generates the clock pulses. The clock pulses are used to trigger a flip-flop whose output serves to enable or disable the AND gate. When the AND gate is enabled, the input signal, after passing through the signal conditioning section comprising level shifting amplifiers, comparators, etc., reaches the counter. In the simplest case, if the AND gate is enabled for 1 s (which is the case when the flip-flop clock input is 1 Hz), then the counter count will represent the signal frequency. The measurement resolution in this case would be 1 Hz. The measurement resolution can be improved by enabling the AND gate for a longer time. For instance, a 0.1 Hz clock at the flip-flop input would give a 10 s gate time and a consequent 0.1 Hz resolution. Similarly, a shorter measurement for a gate time of 0.1 s (corresponding to a clock of 10 Hz) gives a measurement resolution of 10 Hz.

The same building blocks, when slightly rearranged as shown in Fig. 16.22, can be used to measure the time period. Enabling and disabling of the AND gate are now determined by the frequency of the input signal and not by the clock frequency. The number stored in the counter here is proportional to the number of clock pulses that reach the counter during the period of the input signal. The same set-up can be used for time interval (TI) measurement by having two input signal channels, with one enabling the AND gate by, say, setting the flip-flop and the other disabling the same by resetting the flip-flop.

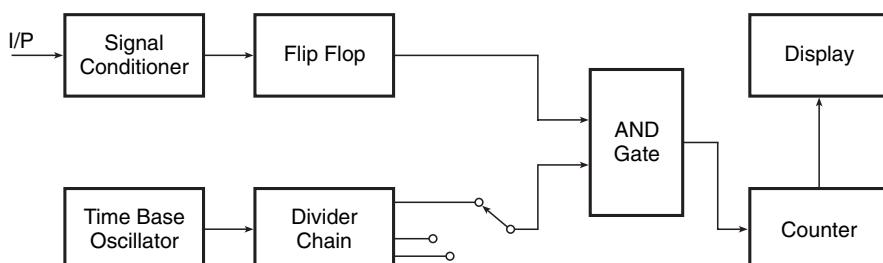


Figure 16.22 Time period measurement using a frequency counter.

16.14.3 Reciprocal Counters

The reciprocal counter overcomes some of the limitations of the basic counter architecture described in the previous paragraphs. Most important of all is its inadequate resolution, particularly when measuring low frequencies. The basic counter had a resolution of 1 Hz for a gate time of 1 s and the resolution could be enhanced only by increasing the gate time. If the gate time cannot be increased in a certain application, the resolution is restricted to 1 Hz. The basic counter measures frequency in terms of event count. Depending upon the gate time, which is 1 s or any other decade value such as 10, 100, etc., the decimal point appropriately placed in the count gives frequency. What is important to note here is that computation of frequency involves computation of the event count only. The frequency, which is given by the event count divided by the time taken, is calculable from the event count itself if the time is 1 s, 10 s, 100 s, etc.

In a reciprocal counter, both events as well as time are computed and the ratio of the two gives the frequency. The advent of the reciprocal counter was made possible owing to the availability of digital logic that could perform arithmetic division economically and with precision. Figure 16.23 shows the reciprocal counter hardware. The processor is the heart of the counter hardware and controls almost every other building block. The synchronizing and routing logic block routes the A and B channel inputs and the time-base signal to the *event* and *time* counters. The routing is determined by the measurement function. The computations are done in the processor block.

As a matter of comparison, let us see how the two counters having an internal clock of 10 MHz would respond to measurement of a signal frequency of 50.38752 Hz. The basic counter will display 50 Hz, assuming a gate time of 1 s as the event count will be 50. The reciprocal counter will also have an event count of 50 but it will also measure time with a resolution of 100 ns (for a 10 MHz clock), equal to 0.9923328 s. The measured frequency will therefore be 50.38752 Hz. The frequency resolution offered for a 10 MHz clock is seven digits, equal to 0.000005 Hz in the present case for a 1 s gate. The resolution could be further enhanced by increasing the clock frequency. Since clock frequencies of up to 500 MHz are practical, a reciprocal counter would give a resolution of 2 ns for a 1 s gate time.

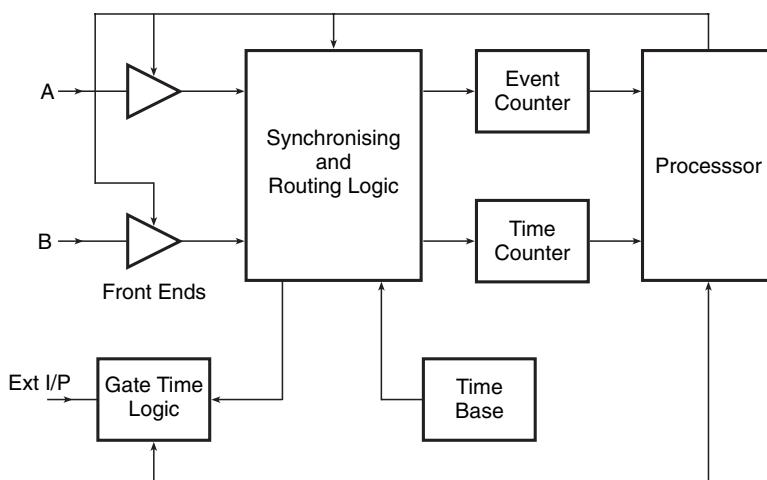


Figure 16.23 Reciprocal counter architecture.

The resolution of reciprocal counters can be further enhanced by using a technique called interpolation. It is possible to achieve a nine-digit resolution with a 10 MHz clock using interpolation techniques that otherwise would require a clock of 1 GHz. This is particularly important when we are looking for a given resolution in shorter gate times. The details of interpolation techniques are beyond the scope of this text.

16.14.4 Continuous-Count Counters

The counter architectures discussed in the previous paragraphs had a counter that counted for a known period equal to the gate time. These counters have a dead time when the gate is disabled. Such counters could miss vital information that could be important to the measurement. The continuous-count counter architecture is based on the fact that, if different measurements of a certain parameter of a signal were not disjoint and the relationship that they had were made use of, the measurement resolution could be significantly enhanced by applying what we call curve-fitting algorithms. These counters have all the attributes of reciprocal counters, with the additional ability of reading the event, the time and the counter without having to disable the gates.

16.14.5 Counter Specifications

The data sheets and manuals of universal counters contain detailed specifications of the instrument. The important ones include *sensitivity*, *bandwidth*, *resolution*, *accuracy* and *throughput*.

16.14.5.1 Sensitivity

This refers to the smallest signal that the instrument can measure and is usually expressed as mV (RMS) or peak-to-peak. A sensitivity of 10–20 mV (rms) is typical. In the majority of measurement situations, sensitivity is not the issue.

16.14.5.2 Bandwidth

The bandwidth of the counter is its front-end bandwidth and is not necessarily the same as the maximum frequency that the counter is capable of measuring. Measuring a signal frequency higher than the instrument's bandwidth only reduces its sensitivity specification and requires a larger minimum input signal. However, the bandwidth does affect the measurement accuracy in the case of some parameters. Rise time is one such parameter. Thus, it is always preferable to choose a counter with as high a bandwidth as possible. Bandwidth is not explicitly mentioned in the specifications. However, it can be estimated by looking at variation in sensitivity across the frequency range of the instrument.

16.14.5.3 Resolution

Resolution refers to the minimum resolvable frequency increment (in the case of frequency measurement) and time increment (in the case of time interval measurement). The resolution is usually very close to the least significant digit and is often ± 1 count or LSD. Noise in the input signal, noise in the front end and input signal slew rate are some of the factors that affect resolution.

16.14.5.4 Accuracy

Accuracy is related to resolution but is not the same as resolution. Factors such as time-base (or clock) accuracy and trigger accuracy must be considered along with the resolution specification to determine the ultimate accuracy of frequency measurement. Time-base error affects measurement accuracy as follows:

$$\text{Frequency accuracy} = \text{resolution} \pm \text{time base error} \times \text{frequency}$$

Trigger level accuracy is the precision with which the trigger level can be set. If there is an error in the trigger level setting, the trigger timing is changed, thus affecting measurement accuracy.

16.14.5.5 Throughput

Throughput is related to resolution. For instance, increasing the gate time of a certain frequency measurement increases the measurement resolution by the same factor, but it slows down the throughput by almost the same amount. Other factors affecting the throughput are more related to the speed of the microprocessor and the interface system. Two factors to be watched here are the number of measurements the counter can deliver through the interface and the speed with which the counter can switch between different functions or set-ups. If short gate times are being used and/or measurements are being switched between different functions repeatedly, these factors become important.

16.14.6 Microwave Counters

The counter architectures discussed in the preceding paragraphs (conventional, reciprocal, continuous count) are usually good enough up to 500 MHz or so. Counters meant for carrying out measurements at RF frequencies beyond 500 MHz and microwave frequencies employ a different architecture. There are two types of architecture in use for building microwave counters. One uses a prescaler while the other is based on down-conversion.

Prescaler counters use a prescaler placed between the front end and the gating circuitry of the counter. In fact, prescalers are available inside the counters as an optional channel to extend the frequency range of measurement. Extension up to 3 GHz is typically available with a prescaler. Prescalers are not used with pulsed microwave counters owing to their tendency to self-oscillation. When used with a basic counter, a prescaler causes degradation of resolution. This is because the frequency resolution of a basic counter is dependent upon the contents of the event counter and, owing to the location of the prescaler before the gating circuitry, its contents cannot be read. The resolution is not affected when the same is used in a reciprocal counter.

In a microwave counter based on down-conversion architecture, the input signal frequency is down-converted to produce an intermediate frequency (IF). The IF, which is the difference between the input signal frequency and the local oscillator (LO) frequency, is then counted. The actual frequency is then computed from LO + IF. Covering a frequency range of tens of GHz for an LO is an expensive proposition. The solution is to use a relatively lower-frequency LO (approximately 200 MHz). The LO drives a step recovery diode that produces a sharp pulse with usable harmonics up to the desired range. This pulse drives a sampler which samples points of the input signal. The resulting IF is low-pass filtered and counted. The actual input frequency is then given by $N \times \text{LO} + \text{IF}$, where N is the harmonic of the LO that goes through the mixing operation. One of the methods for determining N is to measure the IF at two slightly different LO frequencies. N is then given by $(\text{IF}_1 - \text{IF}_2) / (\text{LO}_2 - \text{LO}_1)$. However, all this is the instrument's headache and may take several tens of milliseconds only. Figure 16.24

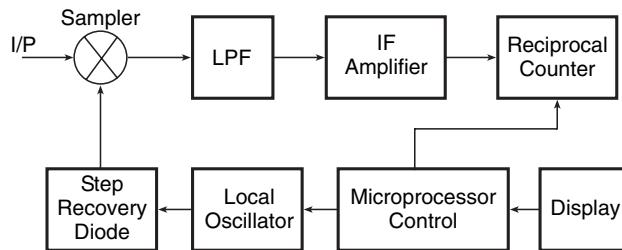


Figure 16.24 Microwave counter architecture.

shows the architecture of a microwave counter. Pulse microwave counters use similar architecture with additional gating circuitry to position the gate comfortably within the pulse.

16.15 Frequency Synthesizers and Synthesized Function/Signal Generators

Frequency synthesizers generate sinusoidal signals of extremely high frequency stability and exceptional output level accuracy. Frequency synthesizers and similar instruments such as synthesized function/signal generators are used to provide test signals for characterization of devices, subsystems and systems. Synthesized function generators, in addition to providing spectrally pure and accurate CW sinusoidal signals, also provide other waveforms such as ramp, triangle, square and pulse. Synthesized signal generators, in addition to providing spectrally pure and accurate CW signals, also have modulation capability and can be used to generate AM, FM, PM and pulse-modulated signals. There is another class of synthesized function generators called synthesized arbitrary waveform generators. The majority of synthesized function generators have a limited arbitrary waveform generation capability built into them. However, these are available as individual instruments also. All the above-mentioned instruments have one thing in common, that is, the synthesis of a signal that lends ultrahigh frequency stability and amplitude accuracy to the generated waveform. They therefore have more or less similar architecture for a given technique used for frequency synthesis.

16.15.1 Direct Frequency Synthesis

The frequency synthesizer in its basic form uses a reference oscillator, which is an ultrastable crystal oscillator, and other signal-processing circuits to multiply the oscillator frequency by a fraction M/N (where M and N are integers) in order to generate the desired output frequency. One such arrangement is shown in Fig. 16.25. It comprises an assortment of frequency multipliers and dividers, mixers and band-pass filters (BPFs). The diagram shows the use of this architecture to generate 17 MHz. In this arrangement, if the BPF has a pass band centred around 3 MHz, the output will be 3 MHz as the mixer produces both sum and difference components. This method of frequency synthesis has several disadvantages, not least that the technique is highly hardware intensive and therefore expensive. Another disadvantage is loss of phase continuity while switching frequencies, with the result that this technique has not found favour with designers.

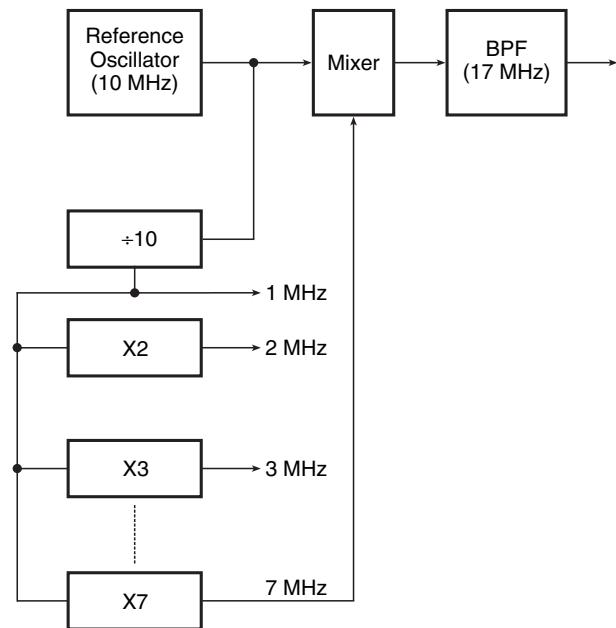


Figure 16.25 Frequency synthesizer architecture – direct frequency synthesis.

16.15.2 Indirect Synthesis

In indirect synthesis, the output is not directly derived from the quartz crystal based reference oscillator. Instead, the reference oscillator is used in a phase-locked loop wired as a frequency multiplier to generate an output frequency that is M/N times the reference oscillator frequency. The output is taken from the VCO of the phase-locked loop. Figure 16.26 shows the basic arrangement. If we insert a divide-by- N circuit between the reference oscillator and the phase detector signal input and a divide-by- M circuit between the VCO output and the phase detector VCO input, then the loop will lock with the VCO output as $f_{ref} \times (M/N)$. The frequency resolution of this architecture is f_{ref}/N , where f_{ref} is the frequency of the reference oscillator. The loop frequency switching speed is of the order of 10 times the period of reference frequency input to the loop phase detector. That is, if we desired a frequency resolution of 1 Hz, the switching time would be of the order of 10 s, which is highly unacceptable. Another disadvantage of this architecture is that frequency multiplier loops also multiply noise at the phase detector, which manifests itself in the form of noise sidebands at the VCO output. This restricts the maximum multiplication factor to a few thousands in this arrangement, which limits the resolution. If a finer resolution is needed, sequences of multiplication, division and addition are used that involve more than one phase-locked loop. One such arrangement is shown in Fig. 16.27. The synthesizer output in this case is given by $f_{ref} \times [m/(N_1 \times N_2) + 1]$.

This technique can be extended to get any desired resolution. Since the multiplication numbers are low and the loop frequency is high, the output will have low noise sidebands. Also, the synthesizer is capable of fast frequency switching. Another popular method of indirect synthesis is fractional N synthesis, where a single PLL is made to lock to the noninteger multiple of the loop reference. This

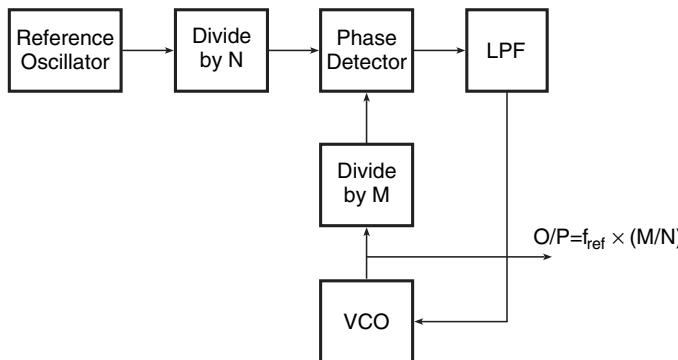


Figure 16.26 Frequency synthesizer architecture – indirect synthesis.

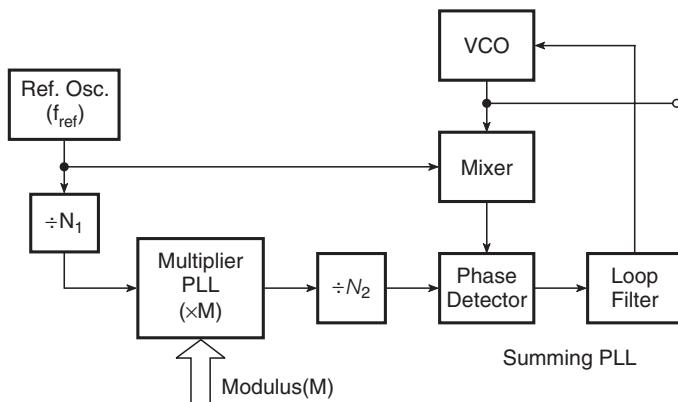


Figure 16.27 Indirect synthesis using more than one PLL.

technique can be used to achieve a frequency resolution of microhertz order at switching speeds of the order of a millisecond or so. Figure 16.28 shows the basic architecture. The configuration functions as follows.

The integer part of the desired multiplier is supplied to the digital divider placed between the VCO output and the phase detector in the form of its dividing factor. The fractional part is supplied to the accumulator. The accumulator is clocked by the reference source derived from the crystal oscillator. The quantum of fractional input is added to the accumulator contents every clock cycle. The VCO output is $N \times F$ times the reference input when the loop is locked. The circuit functions in such a way that the contents of the accumulator predict the expected phase detector output resulting from the frequency difference of the two phase detector input signals. The D/A converter is then so scaled and polarized that its output waveform cancels the phase detector output waveform. The two waveforms are added in the analogue adder, sampled and filtered to provide the oscillator control voltage. Also, to keep the phase detector output within its linear range, whenever the phase difference between the

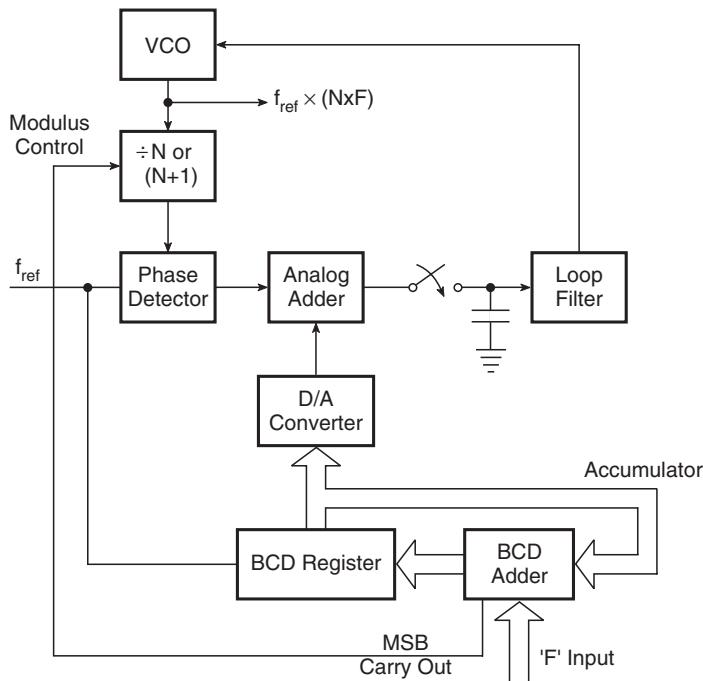


Figure 16.28 Fractional N synthesis.

two inputs to the phase detector tends to become 360° , which is the maximum the phase detector can tolerate without going out of range, the phase of the divider output (which is ahead of the reference input in phase) is retarded by 360° by either changing the divider modulus to $N + 1$ momentarily or by any other means. In the architecture shown, the modulus is changed to accomplish this on receiving a command from the BCD adder at the time of accumulator overflow.

16.15.3 Sampled Sine Synthesis (Direct Digital Synthesis)

This method of frequency synthesis is based on generating the waveform of desired frequency by first producing the samples as they would look if the desired waveform were sampled or digitized according to the Nyquist sampling theorem, and then interpolating among these samples to construct the waveform. As the frequency is the rate of change in phase, this information is made use of to generate samples. The sine of different phase values is stored in a memory, which is addressed by phase increment information stored in an accumulator. Figure 16.29 shows a simplified block schematic representation of direct digital synthesis. When the accumulator is clocked at a fixed frequency, the contents of the accumulator jump by the phase increment whose digital equivalent information is stored in the phase increment register (PIR). By changing the contents of the PIR, the output frequency can be changed. The rate at which the look-up table in the memory is addressed is given by the clock frequency and phase increment during one clock period as given by the PIR contents. For instance, if the contents of the PIR represented a phase angle of 36° , then the digital samples present at the output

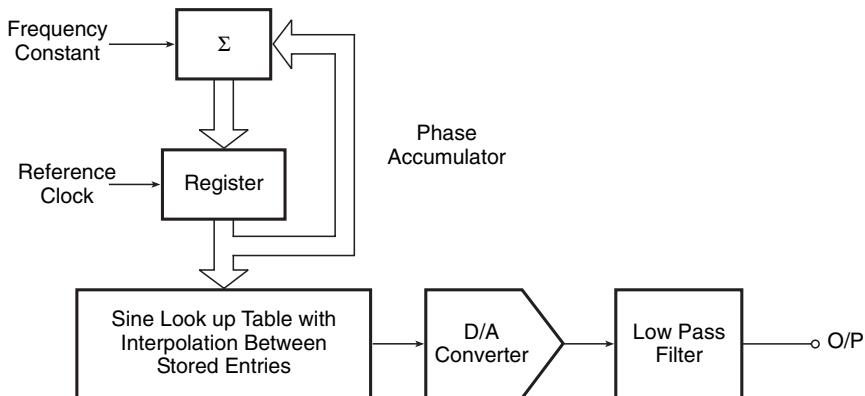


Figure 16.29 Direct digital synthesis.

of the memory would correspond to phase differences of 36, 72, 108, 144, 180, 216, 252, 288, 324 and 360° to complete one cycle of output waveform. The 10 samples will be produced in 10 clock cycles. Therefore, the output frequency will be one-tenth of the clock frequency. In general, the output frequency is given by

$$[\phi/2\pi] \times f_{clock} \quad (16.1)$$

where ϕ is the phase increment in radians.

The digital samples are converted into their analogue counterparts in a D/A converter and then interpolated to construct the waveform. The interpolator here is a low-pass filter. Relevant waveforms are shown in Fig. 16.30.

This method of synthesis derives its accuracy from the fact that both the phase increment information and the time in which the phase increment occurs can be computed to a very high degree of accuracy. With the frequency being equal to the rate of change in phase, the resulting waveform is highly

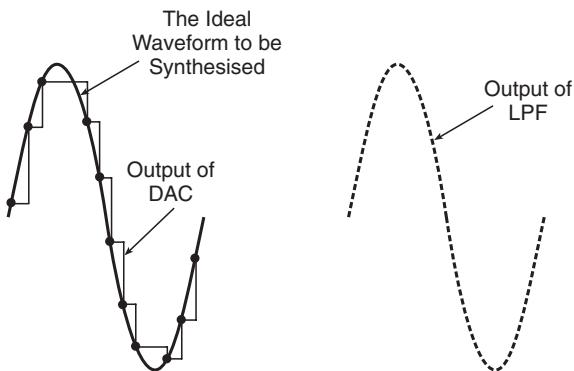


Figure 16.30 Direct digital synthesis – relevant waveforms.

stable. The most important feature of this technique, however, is its capability to provide instantaneous switching. This is possible because the size of the angle increments between two consecutive table look-ups may be changed instantaneously. The limitations of this technique are the *quantization noise* and *aliasing* inherent in any sampled data system. Another serious disadvantage is the presence of spurious components owing to imperfections and inaccuracies in the D/A converter. The highest frequency that can be synthesized is limited by the maximum speed of the available digital logic. The usable frequency range of the direct digital synthesis output may be extended by a variety of techniques. However, depending upon the technique used, some of the advantages of this technique may be lost. As in the case of more conventional synthesizers, the output of a direct digital synthesizer may be doubled, mixed with other fixed sources or used as a reference inside a PLL.

16.15.4 Important Specifications

Frequency range, resolution, frequency switching speed and signal purity are the important synthesizer specifications.

16.15.4.1 Frequency Range and Resolution

While considering the *frequency range*, it is important to note whether the claimed frequency range is being covered in a single band or a series of contiguous bands. This aspect is significant from the viewpoint of noise performance, which may be different in different bands in cases where the frequency range is covered in more than one band. This often leads to a larger transient when the frequency switching involves switchover of the band also. Frequency resolution is usually the same throughout the range. It is typically 0.1 Hz, although a resolution as fine as 1 mHz is also available in some specific instruments.

16.15.4.2 Frequency Switching Speed

The *frequency switching speed* is a measure of the time required by the source to stabilize at a new frequency after a change is initiated. In the PLL-based synthesizers it depends upon the transient response characteristics of the loop. The switching time is typically several hundreds of microseconds to tens of milliseconds in PLL-based synthesizers and a few microseconds in instruments using the direct digital synthesis technique.

16.15.4.3 Signal Purity

The *signal purity* tells how well the output signal approximates the ideal single spectral line. Phase noise is one parameter that affects signal purity. This refers to the sidebands that result from phase modulation of the carrier by noise. It is specified as the total sideband power (in decibels) with respect to the carrier. The presence of spurious signals resulting from undesired coupling between different circuits within the instrument and distortion products in the signal mixers also spoils signal purity.

16.15.5 Synthesized Function Generators

Synthesized function generators are function generators with the frequency precision of a frequency synthesizer. The hardware of a synthesized function generator is similar to that of a frequency

synthesizer with additional circuitry to produce pulse, ramp, triangle and square functions. These instruments with additional modulation capability are referred to as *synthesized signal generators*.

Direct digital synthesis described in the earlier pages of this chapter is almost invariably used in synthesized function/signal generator design. Advances in digital technology have made these synthesized function/signal generators truly versatile. Synthesized sine wave output up to 30 MHz and other functions such as pulse, ramp, triangle, etc., up to 100 kHz, all with a resolution of $1\mu\text{Hz}$, are available in contemporary synthesized function generators.

Figure 16.31 shows one such synthesized function generator (Fluke 271 DDS function generator) that employs direct digital synthesis for achieving a high level of stability. It offers sine, square, triangle and ramp outputs of up to 10 MHz.

16.15.6 Arbitrary Waveform Generator

The *arbitrary waveform generator* (AWG) is a signal source that is used to generate user-specified custom analogue waveforms. Using a custom stimulus waveform and measuring the response waveform provides realistic characterization of the device or system under test. The contemporary AWG allows generation of almost any conceivable waveform.

Direct digital synthesis again is the heart of an arbitrary waveform generator. Figure 16.32 shows the hardware. It looks very similar to the one shown in Fig. 16.29. The sequential amplitude values of the waveform to be generated are stored in the RAM. The size of the RAM decides the number of samples that can be stored, which in turn decides the maximum number of samples into which one period of the desired waveform can be divided. These sample values can be entered into the RAM from the keyboard. Once the sample values are loaded into the RAM, they can be stepped through at a repetition rate governed by the *frequency word* input to the phase accumulator in the same way as explained in the case of a frequency synthesizer. The complexity of the waveform that can be synthesized by this process is limited by the size of the RAM. As a rule of thumb, a minimum of about 3–4 samples per cycle of the highest frequency in the waveform should be used. This is intended to eliminate aliasing. Figure 16.33 shows a typical arbitrary waveform possible in a typical arbitrary waveform generator.



Figure 16.31 Synthesized function generator. Reproduced with permission of Fluke Corporation.

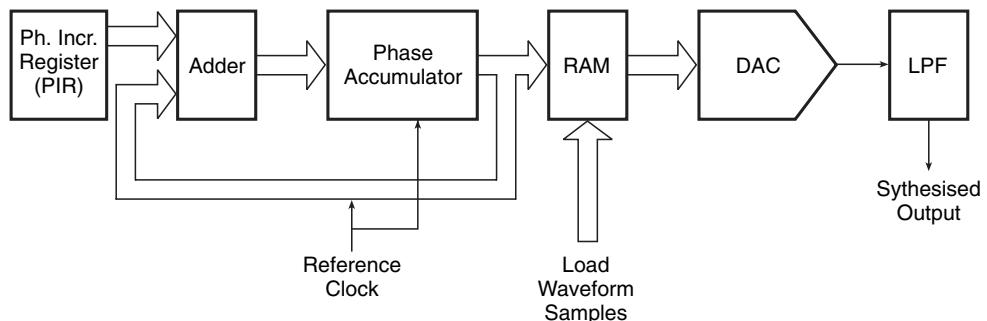


Figure 16.32 Arbitrary waveform generator architecture.

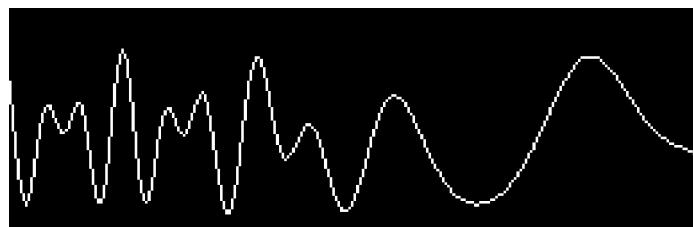


Figure 16.33 Arbitrary waveform generator – typical waveform.

16.16 Logic Probe

The *logic probe* is the most basic tool used for troubleshooting of digital circuits. It is a small, handheld pen-like test instrument with a metallic tip on one end (Fig. 16.34). The instrument can be used to ascertain the logic status of various points of interest such as the pins of digital integrated circuits in a digital circuit. The logic status is indicated by a glowing LED. There may typically be three LEDs



Figure 16.34 Logic probe. Reproduced with permission of Phoronix/Michael J. Larabel.

to indicate logic HIGH, logic LOW and Hi-Z states. Depending upon the actual logic status where the probe is touched, the corresponding LED comes on. The probe can be used to determine open and short circuits. Also, the probe has settings for different logic families to accommodate different acceptable voltage level ranges for logic LOW and HIGH status for different logic families.

16.17 Logic Analyser

The *logic analyser* is used for performance analysis and fault diagnosis of digital systems. Logic analysers have become a very relevant and indispensable diagnostic tool in the present-day instrumentation scenario, with the whole gamut of electronic instruments being centred on microprocessor/microcomputer-based digital architecture. In addition, most logic analysers can be configured to format their outputs as a sequence of microprocessor instructions, which makes them useful for debugging software too.

16.17.1 Operational Modes

The logic analyser works in one of two modes of operation, namely the asynchronous timing mode and the synchronous state mode. A brief description of each of these two modes is given in the following paragraphs.

16.17.1.1 Asynchronous Timing Mode

In this mode of operation, the signals being probed are recorded either as logic ‘0’ or logic ‘1’. The logic analyser provides the time base referred to as the ‘internal clock’. The time base determines when data values are clocked into the memory of the analyser. On screen, the asynchronous mode display looks similar to an oscilloscope display except for the number of channels that can be displayed, which is much larger in the case of a logic analyser.

16.17.1.2 Synchronous State Mode

In this mode of operation, samples of signals are stored in the memory on a clock edge, referred to as the external clock, supplied by the system under investigation. The logic analyser samples new data values or states only when directed by the clock signal. On a given clock edge, the logic states of various signals constitute a group. The logic analyser display in this mode shows progression of states represented by these groups.

16.17.2 Logic Analyser Architecture

Figure 16.35 shows the block schematic arrangement of a logic analyser. Important constituents of all logic analysers include probes, memory, trigger generator, clock generator, storage qualifier and user interface.

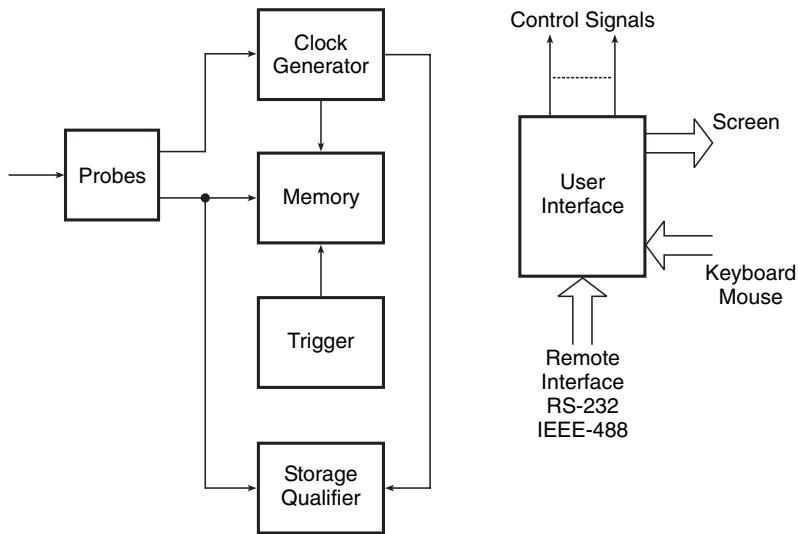


Figure 16.35 Logic analyser architecture.

16.17.2.1 Probes

Probes are used to provide physical connection to the circuit under test without causing any loading effects, so that the logic signal of interest is not unduly affected and its timing integrity is maintained. The probes usually operate as voltage dividers. By this, the comparators inside the probe are asked to handle the lowest possible voltage slew rate that enables higher-speed signals to be captured. These comparators have an adjustable threshold to make the probes compatible with different logic families as different families have different voltage thresholds. The comparators transform the input signals into logic 1s and 0s.

16.17.2.2 Memory

The memory stores the sampled logic values. Addresses for given samples are supplied internally. In a typical measurement using a logic analyser, the user is interested in observing the logic signals around some event called the measurement trigger and the samples have a timing relationship with this trigger event. These samples are placed in the memory, depending upon the instantaneous value of the internally supplied address.

16.17.2.3 Trigger

Logic analysers have both a combinational (or word-recognized) trigger mode and an external trigger mode. In the combinational trigger mode, the trigger circuitry compares the incoming data with a word programmed by the user from the front panel. A trigger signal is generated when the incoming data match with the programmed word. Data are being sampled and stored in the memory by either an internal or an external clock. On the occurrence of a trigger, the stored data samples are displayed on the screen.

16.17.2.4 Clock Generator

As stated earlier, the clock is either internal or external, depending upon whether the selected operational mode is the asynchronous timing mode or the synchronous state mode. The two modes were described in Section 16.17.1. Again, in the timing mode there are two commonly used approaches. Some logic analysers offer both approaches.

In the first approach, called the continuous storage mode, the clock is generated at the selected rate irrespective of the activity occurring on the input signals. The logic status of the input signal is stored in the memory on every clock cycle [Fig. 16.36(a)]. In the second approach, called the transitional timing mode, the input signals are again sampled at the selected rate, but the clock generator circuitry allows the samples to be stored in the memory only if one or more signals change their logic status. Thus, the memory storage locations are used only if inputs change, leading to more efficient use of memory. For each sample, however, a time marker is recorded, as shown in Fig. 16.36(b). This approach offers a distinct advantage when long time records of infrequent or bursts of finely timed events are to be recorded.

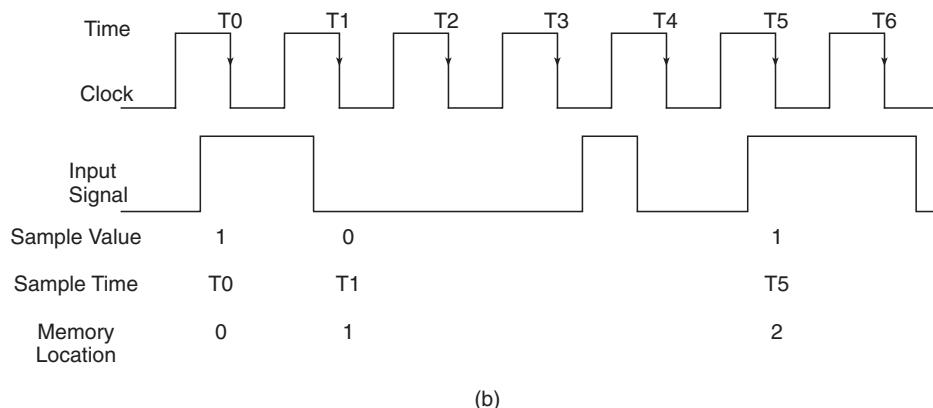
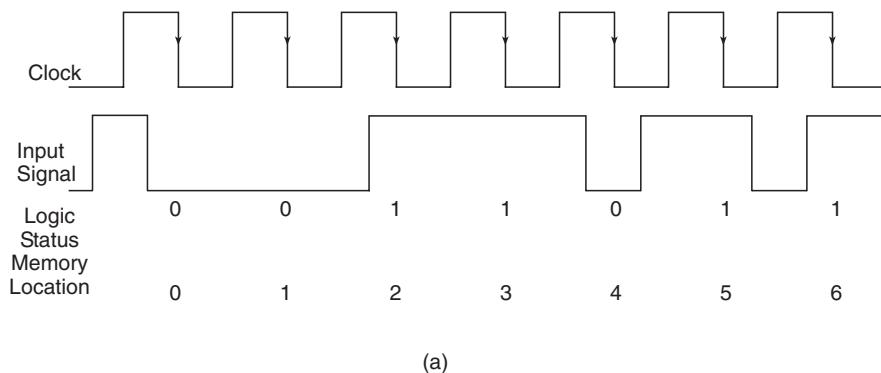


Figure 16.36 Logic analyser – relevant waveforms.

16.17.2.5 Storage Qualifier

The function of the storage qualifier is to determine which data samples are clocked into the memory. The storage qualifier block looks at the sampled data and tests them against a criterion. If the criterion is met, the clocked sample is stored in the memory. This feature is particularly useful in troubleshooting microprocessor architectures. For instance, if the circuit under test is a microprocessor bus, this function can be used to separate bus cycles to a specific I/O port from cycles to all other ports or from instruction cycles.

16.17.2.6 User Interface

Benchtop logic analysers typically use a dedicated keyboard and CRT display. Graphical user interfaces (GUIs) similar to those available on personal computers are also available with many products. Also, interfaces such as RS-232, IEEE-488 or local area network (LAN) enable the use of the instrument from a personal computer or a workstation. Remote interfaces are important in manufacturing applications. LAN interfaces have emerged as critical links in research and development activities where these instruments can be tied to project databases.

16.17.3 Key Specifications

Some of the important specifications of logic analysers include sample rate, set-up and hold times, probe loading, memory depth and channel count. Trigger resources, the availability of preprocessors/inverse assemblers, nonvolatile storage and the ability of the logic analyser to store time value along with captured data are the other key features.

16.17.3.1 Sample rate

The *sample rate* in the timing mode determines the minimum resolvable time interval. Since the relationship of the sample clock and the input signal transition is random, two edges of the same signal can be measured to an accuracy of two sample periods. Measuring a transition on one signal with respect to a transition on another signal can also be done with an accuracy of two sample periods plus whatever skew exists between the channels. In the state mode, the sample rate determines the maximum clock rate that can be measured in the target state machine.

16.17.3.2 Set-up and Hold Times

The *set-up and hold time* specification in the case of logic analysers is similar to that in the case of flip-flops, registers and memory devices. Like these devices, a logic analyser also needs stable data for a specified time before the clock becomes active. This specified time is the set-up time. The *hold time* is the time interval for which the data must be held after the active transition of the clock to enable data capture. The hold time is typically zero for logic analysers.

16.17.3.3 Probe Loading

It is desired that the target system not be perturbed by probe loading. Logic analysers with a sampling rate of equal to or less than 500 MHz have probe specifications of typically 100K and 6–8 pF. Analysers

having a sample rate greater than 1 GHz usually come with SPICE models for their probes so as to enable the users to know the true impact of probe loading on signal integrity.

16.17.3.4 Memory depth

The *memory depth* determines the maximum time window that can be captured in the timing mode or the total number of states or bus cycles that can be captured in the state mode. Most of the logic analysers offer 4K to 1M samples of memory.

16.17.3.5 Channel count

Channel count is the number of available input channels. Together with maximum rate, channel count determines the cost of instrument.

16.18 Computer–Instrument Interface Standards

Quite often, in a complex measurement situation, more than one instrument is required to measure a parameter. In another situation, the system may require a large number of parameters to be measured simultaneously, with each parameter being measured by a dedicated instrument. In such measurement situations, the management of different instruments becomes very crucial. This has found a solution in automated measurement set-ups where various instruments are controlled by a computer. Another reason for instruments being placed into such automated measurement set-ups is to achieve capabilities that the individual instruments do not have. If there were a single instrument that did all the measurements the user required, automated them and compiled all the data in the required format, probably there would be no need for an integrated system. The probability of a single system doing all this is extremely remote when there are a large number of different measurements to be made. Yet another reason for having a computer-controlled instrument system is that it enables the user to make measurements faster and free of any human error.

In an integrated measurement set-up there has to be transfer of data back and forth between different instruments and also between individual instruments and computer. Different interface standards have evolved to allow transfer of data. The IEEE-488 interface is the most commonly used one for the instrument–computer interface. This and some of the other popular interface standards are briefly discussed in the following paragraphs.

16.18.1 IEEE-488 Interface

The IEEE interface has evolved from the Hewlett-Packard interface bus (HP-IB), also called the general-purpose interface bus (GP-IB). Presently, it is the standard interface bus used internationally for interconnecting programmable instruments in an automated measurement set-up.

Figures 16.37(a) and (b) show the general interface and bus structure of IEEE-488/HP-IB. Figure 16.37(a) shows the interconnection of different types of programmable device such as talkers, listeners, controllers, etc. A listener is an instrument that can only receive data from other instruments. A printer is an example of a listener-type instrument. A talker such as a frequency counter is capable of transmitting data to the other instruments connected to the bus. There are some instruments that may perform both the functions. In the listening mode, they receive instructions to carry out certain

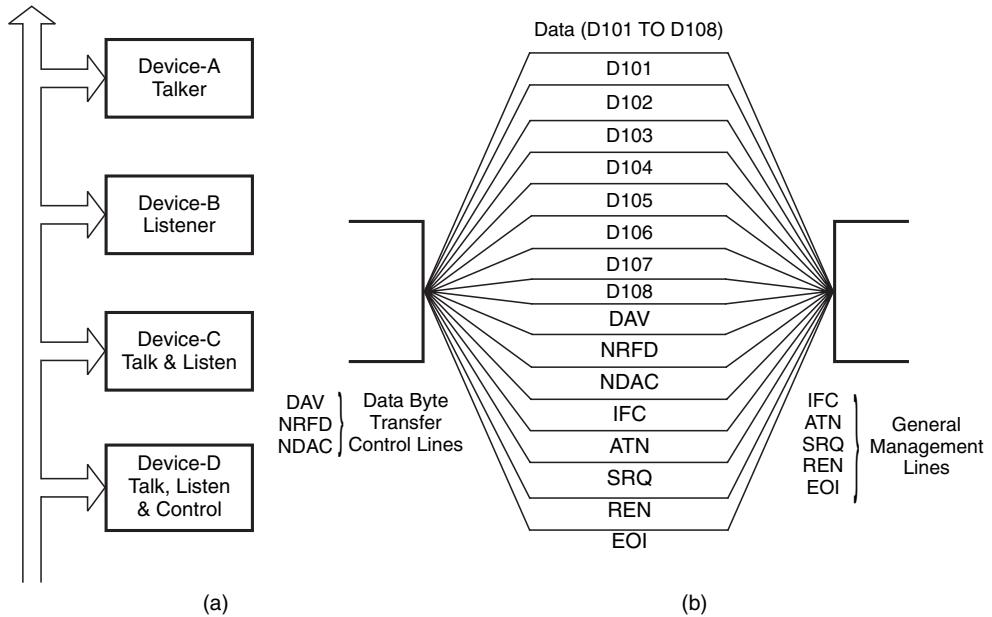


Figure 16.37 IEEE-488 interface standard.

measurements, and in the talking mode they transmit the results of measurements. A controller is supposed to manage the interface bus.

The interface bus has 16 lines and uses a 24-pin connector. A maximum of 15 devices/instruments can be connected to this interface bus in parallel. A typical data rate is 250–500 kbytes/s over the full transmission path which ranges from 2 to 20 m. The maximum data rate is 1 Mbyte/s. There are eight lines dedicated for data transfer (D-101 to D-108) in bit parallel format. There are three lines for data byte transfer control (DAV, NRFD and NDAC) and five lines for general interface management (IFC, ATN, SRQ, REN and EOI). Different lines in the interface bus carry addresses, program data, measurement data, universal commands and status bytes to and from the devices interconnected in the system. The data lines are held at +5 V for logic ‘0’ and pulled to ground for logic ‘1’.

The other popular instrument interface buses that allow interconnection of stand-alone instruments and computers are the VXI-bus, the PCI bus and the MXI-bus. These interface buses are more relevant to the fast-growing concept of virtual instrumentation and therefore are discussed in the next section on virtual instruments.

16.19 Virtual Instrumentation

Advances in software development and rapid increase in the functional capabilities available on the PC platform have changed the traditional instrumentation scenario. The scene is fast changing from the box-like conventional stand-alone instruments to printed circuit cards offering various instrument functions. These cards are inserted either into a card cage, called the mainframe, or into a PC slot. These acquire the measurement data which are then processed in the computer and subsequently displayed on

the monitor in a format as required by the user. Such an instrumentation concept is commonly referred to as *virtual instrumentation*.

16.19.1 Use of Virtual Instruments

There are four types of virtual instrumentation set-up:

1. A set of instruments used as a virtual instrument.
2. A software graphical panel used as a virtual instrument.
3. Graphical programming techniques used as a virtual instrument.
4. Reconfigurable building blocks used as a virtual instrument.

16.19.1.1 Set of Instruments as a Virtual Instrument

In complex measurement situations, usually more than one instrument is required to do the intended measurement. An instrumentation set-up that is used to qualify various subsystems and systems for electromagnetic compatibility (EMC) is an example. In such a set-up, as shown in Fig. 16.38, the computer receives measurement data from all the stand-alone instruments, works on the data and then displays the measurement results. Another similar set-up that has been customized to perform a certain test on a certain specific product, however, would not be classified as a virtual instrument.

16.19.1.2 Software Graphical Panel as a Virtual Instrument

In this type of virtual instrumentation set-up, the instrumentation hardware is controlled by a personal computer from a keyboard or a mouse. The PC screen is used to display the measurement results (Fig. 16.39). The instrumentation hardware could be a traditional box-like instrument or a PC card offering the desired measurement function. The computer control of the instrument is through an interface bus such as IEEE-488.

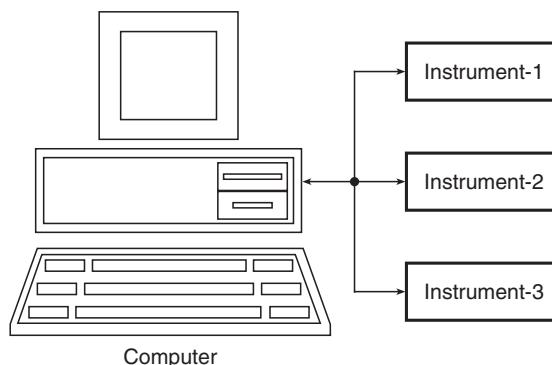


Figure 16.38 Set of instruments as a virtual instrument. Reproduced with permission of Fluke Corporation.

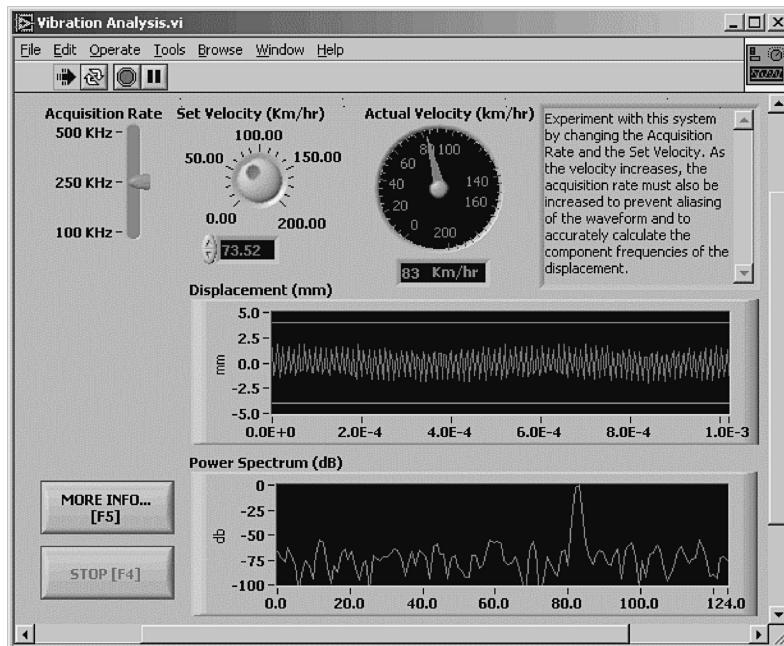


Figure 16.39 Software graphical panel as a virtual instrument. Photo courtesy of National Instruments Corporation.

16.19.1.3 Graphical Programming Technique as a Virtual Instrument

In a typical computer-controlled instrument set-up, the software to do the job is written using a textual programming language such as C, BASIC, Pascal and FORTRAN. Owing to the constant increase in computer power and instrument capabilities, the development of software that makes full use of the instrumentation setup has become a tedious and time-consuming job if it is done using one of the available textual programming languages. There has been a distinct trend to move away from the conventional programming languages and to move towards graphical programming languages. A graphical programming equivalent of a program is a set of interrelated icons (graphical objects) joined by lines and arrows. The use of a graphical programming language leads to a drastic reduction in programming time, sometimes by a factor as large as 10.

Having written a graphical program for a certain test, all icons appear on the screen with programmed interactions. It may be mentioned here that with graphical language the instrument control as well as the program flow and execution are determined graphically. A graphical programming product lists the interface buses and instruments that are supported by it. Graphical programming languages are typically used where one wants to decrease the effort needed to develop a software for instrument systems. However, they require substantial computing power, and the size of these programs can reduce the speed of application in some cases.

16.19.1.4 Reconfigurable Building Blocks as a Virtual Instrument

If one looks into the building blocks of various instruments, one is sure to find a lot of commonality. Building blocks such as front ends, A/D converters, D/A converters, DSP modules, memory modules,

etc., are the commonly used ones. One or more of these building blocks are invariably found in voltmeters, oscilloscopes, spectrum analysers, waveform analysers, counters, signal generators and so on. In an instrumentation set-up comprising more than one instrument function there is therefore likely to be lot of redundant hardware.

A fast-emerging concept is to have instrument hardware in the form of building blocks that can be configured from a graphical user interface (GUI) to emulate the desired instrument function. These building blocks could be reconfigured at will to become voltmeters, oscilloscopes, spectrum analysers, waveform recorders and so on. A graphical panel would represent each virtual instrument.

16.19.2 Components of a Virtual Instrument

The basic components of a virtual instrument as shown in Fig. 16.40 are the computer and display, the software, the bus structure and the instrument hardware.

16.19.2.1 Computer and Display

The majority of virtual instruments are built around personal computers or workstations with high-resolution monitors. The chosen computer should meet the system requirements as dictated by the software packages.

16.19.2.2 Software

The software is the brain of any virtual instrument set-up. The software uniquely defines the functional capabilities of the instrument set-up, and in most cases it is designed to run industry-standard operating systems for personal computers and workstations.

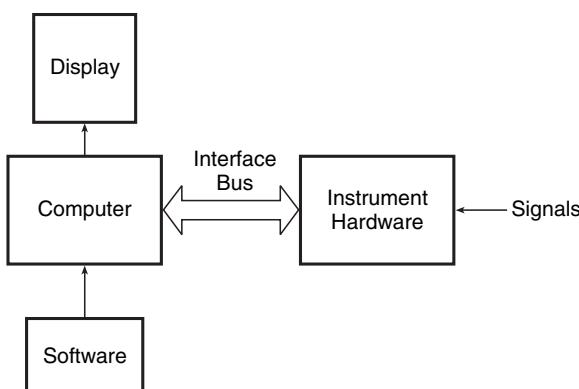


Figure 16.40 Components of a virtual instrument.

16.19.2.3 Interface Bus Structure

Commonly used interface bus structures for a computer–instrument interface are the IEEE-488, described in Section 16.18.1, the PC-bus and the VXI-bus. The other two are described here.

In a PC-bus virtual instrument set-up, the instrument function available on a printed circuit card (Fig. 16.41) is inserted directly into a vacant slot in the personal computer. Since these cards are plugged directly into the computer back plane and contain no embedded command interpreter as found in IEEE-488 instruments, these cards are invariably delivered with driver software so that they can be operated from the computer. PC-bus instruments offer a low-cost solution to building a data acquisition system. Owing to the limited printed circuit space and close proximity to sources of electromagnetic interference, PC-bus instruments offer a lower performance level than their IEEE-488 counterparts.

VXI-bus instruments are plug-in instruments that are inserted into specially designed card cages called mainframes (Fig. 16.42). The mainframe contains power supplies, air cooling, etc., that are common to all the modules. VXI-instruments combine the advantages of computer back-plane buses and IEEE-488. A VXI-bus instrument has high-speed communication as offered by computer back-plane buses (such as the VME-bus) and a high-quality EMC environment that allows high-performance instrumentation similar to that found in IEEE-488 instruments.

One of the methods to communicate with VXI instruments is via IEEE-488, as shown in Fig. 16.43. In this case, an IEEE-488 to VXI-bus converter module is plugged into the VXI-bus mainframe. The mainframe then interfaces with the IEEE-488 interface card in the computer using the standard interface cable. The set-up is easy to program, but the overall speed is limited by the IEEE-488 data transfer rate.

Another technique is to use a higher-speed interface bus between the hardware mainframe and the computer. One such bus is the MXI-bus, which is basically an implementation of the VXI-bus on a flexible cable. In this case, the VXI-MXI converter is plugged into the mainframe and an MXI-interface

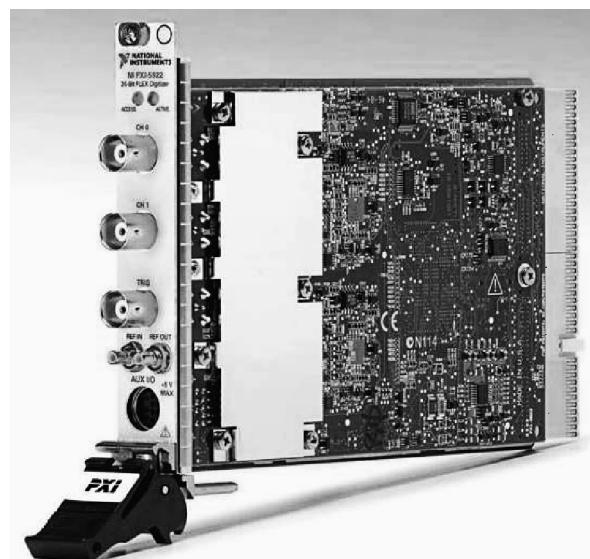


Figure 16.41 Instrument function on a PC card. Photo courtesy of National Instruments Corporation.



Figure 16.42 VXI-bus instruments. Photo courtesy of National Instruments Corporation.

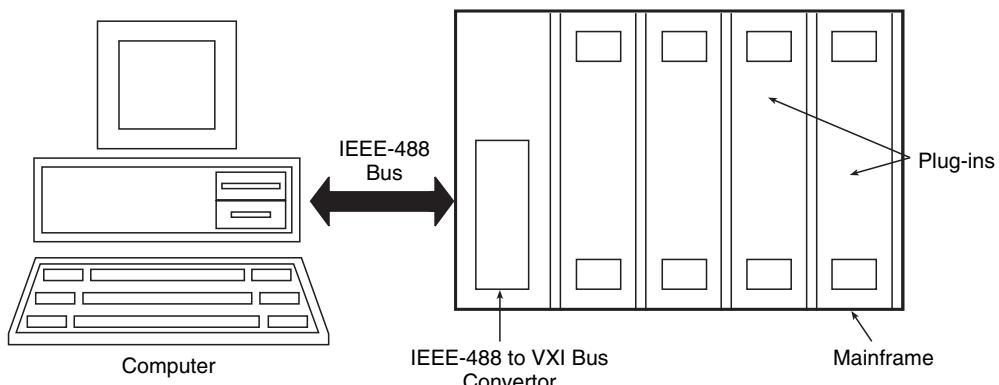


Figure 16.43 VXI instrument-PC interface using IEEE-488.

card with software is installed in the computer. This set-up allows the use of off-the-shelf PCs to communicate with VXI instruments at speeds much faster than IEEE-488 instruments.

Yet another approach is to insert a powerful VXI-bus computer in the hardware mainframe to take full advantage of the VXI-bus instruments. The disadvantage of such a set-up is that, owing to the low volume requirement of VXI computers, these may not be able to match the industry standard personal computers on the price performance criteria. The set-up is shown in Fig. 16.44.

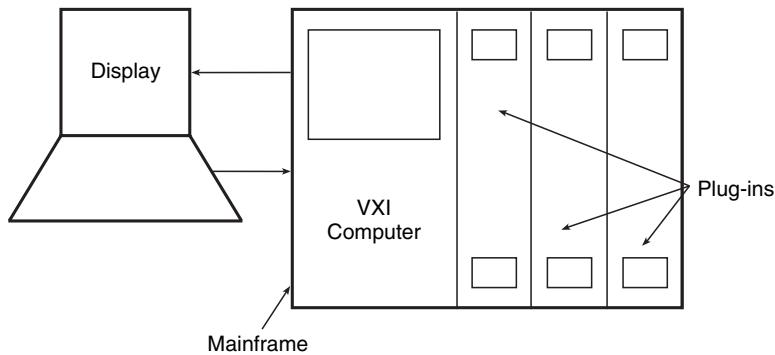


Figure 16.44 VXI-bus computer inside the hardware mainframe.

16.19.2.4 Instrument Hardware

The instrument hardware comprises of sensors and other hardware components that acquire the data and condition it to a level and form so that it can be processed in the computer to extract the desired results.

Review Questions

1. Briefly outline the different steps involved in the troubleshooting of digital circuits. In what possible ways can an internal open or short circuit in a digital IC manifest itself?
2. Why is the troubleshooting of sequential logic circuits a more cumbersome task than in the case of combinational logic? Explain with the help of a suitable illustration.
3. Briefly describe commonly used methods of diagnosing faulty ROM and RAM devices.
4. Distinguish between an analogue storage oscilloscope and a digital storage oscilloscope. Briefly describe the major performance specifications of analogue and digital scopes.
5. With reference to a digital storage oscilloscope, briefly explain the following:
 - (a) How does the effective sampling rate depend upon the acquisition memory?
 - (b) What do you understand by real-time sampling and equivalent-time sampling?
 - (c) What is the difference between bandwidth and sampling rate?
6. Briefly describe the counter architecture when it is used in:
 - (a) frequency measurement mode;
 - (b) time interval measurement mode.
7. What are reciprocal counters? How does a reciprocal counter provide a much higher resolution even when the frequency of the signal is very low?
8. Briefly describe the following with respect to frequency counters:
 - (a) bandwidth;
 - (b) resolution;

- (c) accuracy;
 (d) throughput.
9. Write short notes on:
- sampled sine synthesis;
 - virtual instrumentation.
10. Briefly describe various test and measurement functions that can be performed by a logic analyser. Distinguish between asynchronous and synchronous modes of operation of a logic analyser.

Problems

1. Figure 16.45 shows a D flip-flop wired around a $J-K$ flip-flop that belongs to the TTL family of devices. The D input in this circuit has been permanently tied to V_{CC} . The logic probe observations at the J and K inputs respectively show logic HIGH and logic LOW status, as expected. The Q output of this circuit is supposed to go to logic HIGH status with the first LOW-to-HIGH transition of the clock input. However, the Q output is observed to be a pulsed waveform with the frequency of the signal being one-half of the clock frequency. What is the most probable cause of this unexpected behaviour of the circuit?

The K input of the J-K flip-flop is internally open. The K input is therefore floating and behaves as if it were in the logic HIGH state. This converts it into a toggle flip-flop

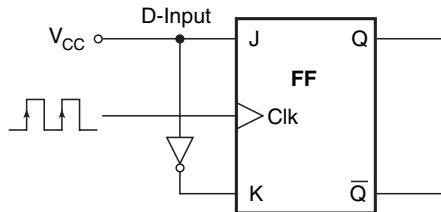


Figure 16.45 D flip-flop (problem 1).

2. Figure 16.46 shows the block schematic arrangement of a three-bit ring counter configured around three D flip-flops. The expected and actual outputs of the flip-flops for the first few clock cycles are listed in Table 16.4. Each of the flip-flops has a propagation delay of 15 ns. Identify the possible cause of observed outputs being different from the expected outputs. The clock signals appearing at the clock input terminals of the flip-flops when seen individually are observed to be clean and free of any noise content. Flip-flops FF-1 and FF-0 are initially cleared to the logic '0' state. The Q output of FF-2 is initially in the logic '1' state.

The fault is possibly due to the clock skew problem. The clock input to FF-1 is delayed from the clock input to FF-2 by a time period that is greater than 15 ns. Also, the clock input to FF-0 is delayed from the clock input to FF-1 by a time period that is greater than 15 ns

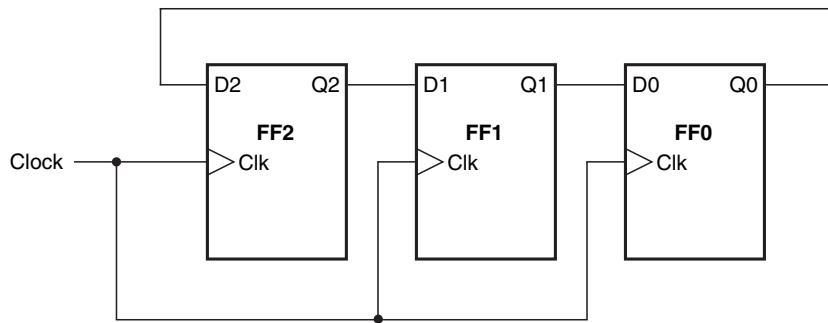


Figure 16.46 Three-bit ring counter (problem 2). Photo Courtesy of National Instruments Corporation.

Table 16.4 Problem 2.

Clock pulse	Expected output			Actual output		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	1	0	0	1	0	0
1	0	1	0	0	0	0
2	0	0	1	0	0	0
3	1	0	0	0	0	0
4	0	1	0	0	0	0
5	0	0	1	0	0	0
6	1	0	0	0	0	0

3. A digital storage oscilloscope is specified to have a sample rate of 400 MS/s and an acquisition memory of 20K. (a) Determine the slowest possible time-base setting for which the specified sample rate is achievable. (b) If the time-base setting were 1 ms per division, what sampling rate would be achievable in this case?

(a) $5\mu\text{s}/\text{div}$; (b) 2 MS/s

4. A transient of 100 ms is to be captured on a digital storage oscilloscope on full screen in the horizontal direction. If the transient is to be recorded at a sampling rate of 100 kS/s, what should the minimum size of the acquisition memory be?

10K

Further Reading

1. Tomal, D. and Widmer, N. S. (2004) *Electronic Troubleshooting*, McGraw-Hill, USA.
2. Coombs Jr, C. F. (1999) *Electronics Instrument handbook*, McGraw-Hill Inc., USA.
3. Webster, J. G. (1999) *The Measurement, Instrumentation and Sensors Handbook*, CRC Press (in cooperation with IEEE Press), FL, USA.
4. Whitaker, J. C. (1996) *The Electronics Handbook*, CRC Press (in cooperation with IEEE Press), FL, USA.

Index

- 1's complement 5, 6
- 10's complement 5
- 15's complement 5
- 16's complement 5
- 16-bit microcontrollers 588–90
 - 68HC12 family 588, 589, 590
 - 68HC16 family 588
- 16-bit microprocessor 547–51
 - 80186 microprocessor 548
 - 80286 microprocessor 548–9
 - 8086 microprocessor 547–8
 - MC68000 microprocessor 549–51
- 16XX series PAL 340–1, 342, 343, 344
- 2's complement 5, 6
- 22V10 PAL 340
- 32-bit microcontrollers 590–2
 - 683XX family 590–1
 - LPC3000 family 591–2
 - MCFXXX family 591
 - MCORE family 591
 - MPC500 family 591
 - TRICORE family 592
- 32-bit microprocessors 551–7
 - 80386 microprocessor 551–3
 - 80486 microprocessor 551, 555–7
 - MC68020 microprocessor 551, 553–4
 - MC68030 microprocessor 551, 554–5
 - PowerPC RISC microprocessor 557
- 4000 CMOS series 117, 165–7
 - 4000 A CMOS series 117, 165
 - 4000 B CMOS series 117, 165–7
 - 4000 UB CMOS series 117, 165–7
- 4004 527
- 4040 527
- 4221 BCD code 20
- 4585 magnitude comparator 262, 263, 264
- 54/74AC CMOS series 117, 168
- 54/74ACT CMOS series 117, 168
- 54/74C CMOS series 117, 167
- 54/74HC CMOS series 117, 167
- 54/74HCT CMOS series 117, 167
- 5421 BCD code 20
- 555 monoshot triggering on leading edges 371
- 555 monoshot triggering on trailing edges 370
- 555 Timer IC 363–71
- 6800 microprocessor 528, 544–6
 - addressing modes 545
 - instruction set 545
 - registers 545
- 683XX family of microcontrollers 590–1
- 68HC05 family of microcontrollers 583
- 68HC11 family of microcontrollers 584–5, 586
- 68HC12 family of microcontrollers 588, 589, 590
- 68HC16 family of microcontrollers 588

- 7's complement 5
 74164 SIPO shift register 452, 453
 74166 PISO shift register 452, 454, 455
 74182 look ahead carry generator 256, 259
 74194 universal shift register 455–8
 74199 PIPO shift register 453, 456
 74284 multiplier 260, 261
 74285 multiplier 260, 261
 74293 counter 418, 419
 7483 adder 246
 7485 magnitude comparator 262, 263, 264
 7491 SISO shift register 451
- 8's complement 5
 8008 527
 80186 microprocessors 528, 548
 80286 microprocessor 528, 548–9
 80386 microprocessor 528, 551–3
 addressing modes 552–3
 instruction set 551–2
 registers 551
 80486 microprocessor 528, 555–7
 instruction set 556
 operating modes 556–7
 registers 556
 8080 528
 8085 microprocessor 528, 541–4
 addressing modes 542
 instructions 542
 registers 542
 8086 microprocessor 528, 547–8
 addressing modes 547
 instruction set 547
 internal architecture 547
 pin-out configuration 547
 registers 547
 8088 528
 80C31FA microcontroller 582
 80C31RA+ microcontroller 583
 80C32 microcontroller 583
 80C51/87C51/80C31 580–2
 addressing modes 581
 architecture 580, 581
 instruction set 582
 interrupts 582
 microcontroller related features 580
 peripheral related features 580
 pin connection diagram 580, 582
 power modes 582
 registers 580–1
 8421 BCD code 19, 20, 21, 22
 89C51 microcontroller 583
- architecture 583
 microcontroller related features 583
 peripheral related features 583
 pin-connection diagram 583
 8-bit microcontrollers 580–8
 68HC05 583
 68HC11 584–5, 586
 80C31RA+/8XC51RA+/RB+/RC+ 583
 80C32/8XC52/54/58 583
 80C51/87C51/80C31 580–2
 89C51 583
 8XC51 RD+ 583
 PIC 16X84 family 585–6, 587
 XC-800 family 586–8
 8-bit microprocessors 541–6
 6800 microprocessor 544–6
 8085 microprocessor 541–4
 Z80 microprocessor 546
 8XC51FA/FB/FC microcontroller 582
 8XC51RA+/RB+/RC+ microcontroller 583
 8XC51RD+ microcontroller 583
 8XC52/54/58 microcontroller 583
- 9's complement 5
- A/D converter 473, 495–522
 A/D converter terminology 499–500
 applications 520–2
 integrated circuit A/D converters 513–20, 521
 specifications 495–9
 types 500–11
 A/D converter applications 520–2
 data acquisition 520, 521–2
 A/D converter specifications 495–9
 accuracy 495, 496
 acquisition time 495, 498
 aliasing phenomenon 495, 496
 aperture time 495, 498
 code width 495, 499
 conversion time 495, 498
 differential nonlinearity 495, 497–8
 gain drift 495, 496
 gain error 495, 496
 nonlinearity 495, 497
 offset drift 495, 496
 offset error 495, 496
 quantization error 495, 496–7
 resolution 495–6
 sampling frequency 495, 496
 A/D converter terminology 499–500
 bipolar mode operation 499
 coding 499
 command register 500

- A/D converter terminology (*Continued*)
control lines 500
high byte 499
left justified data 499
low byte 499
right justified data 499
status register 500
unipolar mode operation 499
- A/D converter types 500–11
counter-type A/D converter 500, 504–5
dual-slope A/D converter 500, 506, 507–8
flash A/D converter 500–3
half-flash A/D converter 500, 503–4
multi-slope A/D converter 500, 506, 508–9
pipeline A/D converter, *see* Half-flash A/D converter
quad-slope A/D converter 509
sigma-delta A/D converter 500, 509–11
simultaneous A/D converter, *see* Flash A/D converter
single-slope A/D converter 500, 506–7
successive approximation type A/D converter 500, 505–6
tracking-type A/D converter 500, 505
triple-slope A/D converter 508–9
- ABEL-Hardware Description Language 338, 339
- Absolute addressing mode 534–5
- Absorption law 197
- Accumulator 530
- Accumulator-based architecture 577
- Accumulator register 260
- Accuracy-A/D converter 495, 496
- Accuracy-analogue oscilloscope 673, 674
- Accuracy-D/A converter 476, 477
- Accuracy-digital storage oscilloscope 674, 676–7
- Accuracy-frequency counter 682, 683
- Acquisition time-A/D converter 495, 498
- Act-1 FPGA 350–1
- Act-2 FPGA 350–1
- Act-3 FPGA 350–1
- Actel antifuse 336
- Active HIGH decoding 429
- Active LOW decoding 429
- AD-5240 A/D converter 513, 516–518
- AD-7524 D/A converter 486, 489
- AD 7820 A/D converter 513, 516, 519
- AD ADC-80 A/D converter 513, 515–516, 517
- AD ADC-84 A/D converter 513, 516, 518
- AD ADC-85 A/D converter 513, 516, 518
- ADC-0800 A/D converter 513–514, 515
- ADC-0808 A/D converter 513, 514–515, 516
- ADC-80 A/D converter 513, 515–516, 517
- ADC-84 A/D converter 513, 516, 518
- ADC-85 A/D converter 513, 516, 518
- Adder-subtractor 245–6, 247
- Addition of larger-bit binary numbers 49–52
- Addition using the 2's complement method 49–52
- Address bus 526
- Addressing modes 534–7
absolute addressing mode 534–5
immediate addressing mode 534, 535
implicit addressing mode 534, 537
indexed addressing mode 534, 536–7
memory direct addressing mode, *see* Absolute addressing mode
register direct addressing mode 534, 535, 536
register indirect addressing mode 534, 535–6
relative addressing mode 534, 537
- Addressing modes-80C51 581
- Advanced low power Schottky TTL 116, 124, 137–9
- Advanced Schottky TTL 116, 124, 137, 139–40
- Aliasing phenomenon-A/D converter 495, 496
- Alphanumeric codes 27–38
American Standard Code for Information Interchange, *see* ASCII code
ASCII code 27, 28–31, 37
EBCDIC code 27–8, 31–7
Extended Binary Coded Decimal Interchange Code, *see* EBCDIC code
Unicode 28, 37–8
- Alphanumeric terminal 645
- ALU 260, 528, 529
- AMD64 528
- American Standard Code for Information Interchange, *see* ASCII code
- Analogue computer 608
- Analogue oscilloscope 669, 672, 673–4
- Analogue-to-digital converter, *see* A/D converter
- Anatomy of a computer 605–7
central processing unit 605, 606
input/output ports 605, 607
memory 605, 606–7
- AND gate 69, 71, 73–5
- AND-OR-INVERT gates 87, 89, 90
- ANSI symbols, *see* IEEE/ANSI standard symbols
- ANSI X3.4-1968 28
- ANSI X3.4-1977 28
- ANSI X3.4-1986 28
- ANSI/IEEE STD 754-1985 *see* IEEE-754 standard
- Antifuse 300, 308, 334, 335–6, 337
- Antifuse-based FPGA 349
- Aperture time-A/D converter 495, 498
- Application relevant information-arithmetic circuits 266
- Application relevant information-counters and registers 466, 468–9

- Application relevant information-flip flops 407–8
 Application relevant information-level translator
 ICs 184
 Application relevant information-logic gates 107–9
 Application relevant information-microcontroller
 580–92
 16 bit microcontrollers 588–90
 32 bit microcontrollers 590–2
 8 bit microcontrollers 580–8
 Application relevant information-PLDs 340–52
 application relevant information-CPLD 343–9
 application relevant information-FPGA 349–52
 application relevant information-SPLD
 340–3, 344
 Arbitrary waveform generator 690, 691
 Arithmetic circuits 236–66
 adder-subtractor 245–6, 247
 application relevant information 266
 arithmetic logic unit 260
 BCD adder 246–9
 controlled inverter 236, 244–5
 full adder 236, 237–40, 241
 full subtractor 236, 242–4, 245
 half adder 236–7
 half subtractor 236, 240–2
 look ahead carry generator 254–7, 258, 259
 magnitude comparator 261–4
 multipliers 260–1
 Arithmetic instructions 531, 532–3
 Arithmetic logic unit, *see* ALU
 Arithmetic operations 47
 ARM926EJ core 591
 ASCII code 27, 28–31, 37
 ASCII-8 *see* US ASCII-8
 Assembler 540
 Assembly language 540
 Associative laws 194–5
 Astable multivibrator 357, 362–3, 364–7, 368
 Astable multivibrator using timer IC 555 364–7, 368
 Asynchronous communication interface 571–2
 Asynchronous counter, *see* Ripple counter
 Asynchronous input active pulse width 399, 401–2
 Asynchronous inputs 398
 Asynchronous SRAM 612, 613–616, 617
 Asynchronous SRAM timing diagram 614–616, 617
 Asynchronous timing mode-logic analyser 692
 Auxiliary storage, *see* secondary storage
- Bandwidth-analogue oscilloscope 673
 Bandwidth-digital storage oscilloscope 674–5
 Bandwidth-frequency counter 682
 Basic logic gates 71
 Basic multilingual plane 38
 BCD 19–21
 BCD-to-binary conversion 20
 binary-to-BCD conversion 20
 higher density BCD encoding 21
 packed BCD numbers 21
 unpacked BCD numbers 21
 BCD adder 246–9
 BCD addition in excess-3 code 57
 BCD counter 426
 BCD input D/A converter 482–3
 BCD subtraction in Excess-3 code 57
 BCD-to-binary conversion 20
 BCD-to-decimal decoder 431, 432
 BiCMOS inverter 171
 BiCMOS logic 115, 116, 170–2
 BiCMOS inverter 171
 BiCMOS NAND gate 171–2
 BiCMOS NAND gate 171–2
 Bidirectional shift register 455
 Binary addition 47–51
 Binary coded decimal, *see* BCD
 Binary codes 19–43
 alphanumeric codes 27–38
 ASCII code 27, 28–31, 37
 binary coded decimal 19–21
 cyclic redundancy check code 41–2
 EBCDIC code 27–8, 31–7
 error detection and correction codes 40–3
 excess-3 code 21–22
 gray code 23–6
 Hamming code 42–3
 parity code 41
 repetition code 41
 seven-segment display code 38–40
 Unicode 28, 37–8
 binary digit, *see* Bit
 Binary division 60–2
 repeated right-shift and subtract algorithm 61
 repeated subtract and left-shift algorithm 62
 Binary-Gray code conversion 24
 Binary ladder network 475–6
 Binary multiplication 58–60
 repeated add and right-shift algorithm 58, 59–60
 repeated left-shift and add algorithm 58, 59
 Binary number system 2, 3–5
 Binary reflected Gray code 23, 25
 Binary ripple counter 413–420
 Binary ripple counter with a modulus less than 2^N
 416–418
 Binary subtraction 47, 48–9, 52–5
 Binary-to-BCD conversion 20
 Binary-to-decimal conversion 6

- Binary-to-hex conversion 10, 11
Binary-to-octal conversion 9, 11
BIOS 611
Bipolar-CMOS Logic, *see BiCMOS Logic*
Bipolar families 116
Bipolar mode operation-A/D converter 499
Bipolar output D/A converter 479, 480
Bistable multivibrator 357–8
Bit 4
Boolean algebra 189–226
 postulates 192
 simplification techniques 204–26
 theorems 192–202
Boolean algebra-simplification techniques 204–26
 Karnaugh map method 204, 216–26
 Quine–McCluskey tabular method 204, 208–14
Boolean expressions 190–1, 204–8
 canonical form of Boolean expression 206
 complement of a Boolean expression 190–1
 dual of a Boolean expression 191
 equivalent of a Boolean expression 190
 expanded forms of Boolean expression 206
 literals 190
 Pi (Π) nomenclature 207–8
 product-of-sums Boolean expression 204,
 205–6
 sigma (Σ) nomenclature 207–8
 sum-of-products Boolean expression 204–5
 terms 190
 variables 190
Borrow-in 48–9
Borrow-out 48–9
Branch instructions, *see* Control transfer
 instructions
Buffer register 529–30
Buffers 98
Burst extended data output (BEDO) DRAM 622
Bus 526
Byte 4

Cache memory 606, 622, 623
Call instruction 533, 534
CAN bus 569, 573
Canonical form of Boolean expressions 206
Carry generate 254, 256
Carry-in 48, 238–9
Carry-out 47–8, 238–9
Carry propagate 254, 256
Carry propagation 254
Carry propagation time 254
Cascading counters 433–6
 cascading BCD counters 435–6
 cascading binary counters 433–5
Cascading decoders 288
Cascading magnitude comparator 263–4
Cascading multiplexers 280
CD-R 649
CD-ROM 648, 649
CD-RW 649
Celeron processor 560
Central processing unit, *see* CPU
Channel count-logic analyser 695, 696
Character codes, *see* Alphanumeric codes
Characteristic equations 377
Characteristic parameters of logic families, *see* Logic
 families-characteristic parameters
Characteristic table 376, 378
Check bits 40
Checksum method 665
Chen-Ho encoding 21
Circulating register, *see* Ring counter
CISC 528, 531, 541
Classification-digital ICs 183–4
 large scale integration 183, 184
 medium scale integration 183, 184
 small scale integration 183, 184
 ultra large scale integration 183, 184
 very large scale integration 183, 184
Classification of computers 607–10
 on the basis of applications 607–8
 on the basis of size and capacity 609–10
 on the basis of technology used 608
CLEAR input 398
Clock generator 560, 563
Clock pulse HIGH Time-flip flop 399, 401
Clock pulse LOW Time-flip flop 399, 401
Clock skew 659
Clock transition time-flip flop 399, 402
Clocked flip flop 377
Clocked R-S flip flop 377–81
 Clocked R-S flip flop with active HIGH inputs 379
 Clocked R-S flip flop with active LOW inputs 379,
 380–1
CML, *see* ECL
CMOS AND gate 154, 155
CMOS AND-OR-INVERT gate 157–9
CMOS bilateral switch, *see* CMOS Transmission
 gate
CMOS exclusive-NOR gate 156, 158
CMOS exclusive-OR gate 154, 156, 157
CMOS-floating inputs 162–3
CMOS-guidelines 179
CMOS-input protection 163, 164
CMOS inverter 151–2
CMOS-latch up condition 163–5

- CMOS logic family 116, 117, 151–68
 AND gate 154, 155
 AND-OR-INVERT gate 157–9
 Exclusive NOR gate 156, 158
 Exclusive OR gate 154, 156, 157
 floating inputs 162–3
 guidelines to using CMOS devices 179
 input protection 163, 164
 inverter 151–2
 latch up condition 163–5
 NAND gate 152–3
 NOR gate 153–4, 155
 open drain output 161–2
 OR-AND-INVERT gate 157–60
 OR gate 154, 156
 subfamilies 117, 165–8
 transmission gate 160–1
 tristate outputs 162, 163
 unused inputs 162–3
 CMOS NAND gate 152–3
 CMOS NOR gate 153–4, 155
 CMOS open drain output 161–2
 CMOS OR-AND-INVERT gate 157–60
 CMOS OR gate 154, 156
 CMOS subfamilies 117, 165–8
 4000 series 117, 165–7
 74AC/ACT series 117, 168
 74C series 117, 167
 74HC/HCT series 117, 167
 CMOS-to-ECL interface 183, 184
 CMOS-to-TTL interface 179–80, 181
 CMOS transmission gate 160–1
 CMOS-tristate output 162, 163
 CMOS-unused inputs 162–3
 Code density 539
 Code table 37, 38
 Code width-A/D converter 495, 499
 Code width uncertainty-A/D converter 499
 Coded keyboard 594
 Coding-A/D converter 499
 ColdFire Version 2 core 591
 Combinational logic 69
 Combinational logic circuit 233–6
 Command register-A/D converter 500
 Common anode seven-segment display 39, 40
 Common cathode seven-segment display 39, 40
 Commutative laws 194
 Compact Disc Read, *see* CD-R
 Compact Disc Read/Write, *see* CD-RW
 Compact Disc ROM, *see* CD-ROM
 Companding D/A converter 478, 479, 480
 Compiler 540
 Complement of a Boolean expression 190–1
 Complementary Metal Oxide Semiconductor logic family, *see* CMOS logic family
 Complementary output gates 91, 93
 Complementation law 193
 Complex Instruction Set Computer, *see* CISC
 Complex programmable logic device, *see* CPLD
 Components of a microcontroller 567–74
 A/D converters 567, 569–70
 central processing unit 567, 568
 controller area network (CAN) bus 569, 573
 counters/timers 567, 569, 570–1
 I/O ports 569, 570
 interintegrated circuit (I²C) bus 569, 572–3
 local interconnect network (LIN) bus 569, 573–4
 peripheral components 567, 569–74
 random access memory 567, 569
 read only memory 567, 569
 serial communication interfaces 569, 571–2
 special function registers 567, 569
 Computer-instrument interface standards 696–7
 IEEE-488 interface 696–7
 Computer memory 610–635, 645–50
 auxiliary storage, *see* secondary storage
 expanding memory capability 632–5
 magnetic storage devices 645–8
 magneto-optical storage devices 648
 optical storage devices 648–50
 primary memory 610, 611–35
 random access memory 611–22
 read only memory 611, 612, 622–31
 secondary storage 610, 611, 645–50
 Computer system 607–10
 mainframe computers 609
 microcomputers 609
 minicomputers 609
 personal computers 610
 supercomputer 609, 610
 types 607–10
 workstation 610
 Computer terminal 645
 Computer word 4
 Conditional jump instruction 533, 534
 Consensus theorem 198–9
 Contact bounce 594, 595
 Continuous-count counter 682
 Control bus 526
 Control lines-A/D converter 500
 Control transfer instructions 531, 533–4
 Control unit 528, 531
 Controlled inverter 236, 244–5
 Controller area network, *see* CAN
 Conversion speed-D/A converter 476, 477–8
 Conversion time-A/D converter 495, 498

- CoolRunner 347–8
Counter type A/D converter 500, 504–5
Counter/timer 567, 569, 570–1
Counters 411–42, 464–6, 467, 468–9
 application relevant information 466, 468–9
 asynchronous counter, *see* Ripple counter
 BCD counter 426
 binary ripple counter 413–419
 binary ripple counter with a modulus of less than 2^N 416–18
 cascading counters 433–6
 decade counter 426
 decoding a counter 428–33
 designing counters with arbitrary sequences 438–42
 DOWN counter 421
 IEEE/ANSI symbology 464–6
 modulus of a counter 413
 parallel counter, *see* Synchronous counter
 presettable counters 426–8
 propagation delay in ripple counters 412–413
 ripple counter 411–419
 ripple counters in IC form 418, 419
 synchronous counter 413, 423–5
 UP counter 423–4
 UP/DOWN counter 425–6
 variable modulus with presettable counters 428, 429
CPLD 306, 328–31
 applications 330–1
 internal architecture 328–30
CPLD-applications 330–1
CPLD-I/O control block 328
CPLD-internal architecture 328–30
CPLD-logic block 328
CPLD-programmable interconnect matrix 328, 329
CPU 567, 568, 605, 606
CRC code 41–2
CRT display 644
Current injection logic, *see* Integrated Injection Logic
Current mode logic, *see* ECL
Current output version-DAC-80 D/A converter 487, 488
Current steering mode-D/A converter 480–1
Current transients 142
Customized computing 333
Cyclic property of Gray code 23
Cyclic redundancy check code, *see* CRC code
- D flip flop 394–5, 396, 397
 J–K flip flop as D flip flop 395, 397
D latch 395, 397
- D/A converter 473–95
 applications 490–5
 BCD input D/A converter 482–3
 binary ladder network 475–6
 bipolar-output D/A converter 479, 480
 companding type D/A converter 478, 479, 480
 current steering mode 480–1
 integrated circuit D/A converter 486–9
 modes of operation 480–2
 multiplying-type D/A converter 479–80
 resistive divider network 474–5
 specifications 476–9
 types 479–80
 voltage steering mode 480, 481–2
D/A converter applications 490–5
 audio attenuator 490
 digitally controlled filters 490, 493–5
 divider 490, 491
 low-frequency function generator 490, 492–3
 multiplier 490
 programmable gain element 490
 programmable integrator 490, 491–2
D/A converter-modes of operation 480–2
 current steering mode 480–1
 voltage steering mode 480, 481–2
D/A converter specifications 476–9
 accuracy 476, 477
 conversion speed 476, 477–8
 differential nonlinearity 476, 478
 dynamic range 476, 478
 monotonicity 476, 478, 479
 nonlinearity 476, 478
 resolution 476–7
 settling time 477–8
D/A converter types 479–80
 bipolar output D/A converter 479, 480
 companding D/A converter 479, 480
 multiplying type D/A converter 479–80
DAC-08 D/A converter 486–7
DAC-0808 D/A converter 486, 487
DAC-1408/ DAC-1508 D/A converter 486, 489
DAC-80 D/A converter 486, 487, 488
Data acquisition 520, 521–2
Data integrity 573
Data selector, *see* Multiplexer
Data storage and transfer circuits 402
Data transfer instructions 531
Debugger 540
Decade counter 426
Decimal digits of precision 13

- Decimal equivalent 6–7
 binary-to-decimal conversion 6
 hexadecimal-to-decimal conversion 6, 7
 octal-to-decimal conversion 6–7
- Decimal number system 1, 2–3, 5
- Decimal-to-binary conversion 7–8
- Decimal-to-hexadecimal conversion 9
- Decimal-to-octal conversion 8
- Decimation filter 511
- Decoder 285–8
- Decoding a counter 428–33
- Delay flip flop, *see* D flip flop
- Delay line 98
- Delta encoded A/D converter, *see* Tracking type A/D converter
- Delta modulator 509, 511
- DeMorgan's theorem 199–200
- Demultiplexer 285
- Densely packed decimal 21
- Design entry 337
- Designing counters with arbitrary sequences 438–42
- Detecting the sequence of edges 404–5
- Differential nonlinearity-A/D converter 495, 497–8
- Differential nonlinearity-D/A converter 476, 478
- Digital arithmetic 47–66
- BCD addition 57
 - BCD subtraction 57
 - binary addition 47–51
 - binary division 60–2
 - binary multiplication 58–60
 - binary subtraction 47, 48–9, 52–5
 - floating-point arithmetic 64–6
- Digital computer 608
- Digital IC-based monostable multivibrator 363, 364, 365
- Digital multimeter 665–7, 668
- advantages 666
 - inside the digital multimeter 666
 - significance of the half digit 666–7
- Digital oscilloscope 669–72
- Digital storage oscilloscope 668, 669–70, 674–7
- Digital-to-analogue converter, *see* D/A converter
- Digital Versatile Disc ROM, *see* DVD-ROM
- Digitally controlled filter 490, 493–5
- Diode logic, *see* DL
- Diode transistor logic, *see* DTL
- Direct digital synthesis 687–9
- Direct frequency synthesis 684–685
- Direct memory access, *see* DMA
- Display 644–5
- Distributive laws 195–6
- DL 116, 117
- DMA 526
- DMA controller 526, 560, 561–2
- Dot-matrix printer 644
- Double-extended precision format 14
- Double precision format 14, 15
- DOWN counter 421
- DRAM 612, 619–22
- burst extended data output (BEDO) DRAM 622
 - DRAM architecture 620
 - extended data output (EDO) DRAM 622
 - fast page mode (FPM) DRAM 622
 - synchronous (S) DRAM 622
 - timing diagram 621
 - types of DRAM 622
- DRAM architecture 620
- DRAM timing diagram 621
- DTL 116, 117
- Dual of a Boolean expression 191
- Dual slope A/D converter 500, 506, 507–8
- Dumb terminal 645
- DVD-ROM 649, 650
- Dynamic RAM, *see* DRAM
- Dynamic range-D/A converter 476, 478
- EBCDIC code 27–28, 31–7
- ECL 116, 118, 147–50
- logic gate implementation 148–50
 - salient features 150
 - subfamilies 118, 147–8
- ECL-OR/NOR gate 148–50
- ECL-salient features 150
- ECL subfamilies 118, 147–8
- ECLinPS 118, 147, 148
 - ECLinPSLite 118, 147, 148
 - MECL 10E 118, 147, 148
 - MECL 10H 118, 147, 148
 - MECL 10K 118, 147, 148
 - MECL-I 118, 147
 - MECL-II 118, 147
 - MECL-III 118, 147
- ECL-to-CMOS interface 183, 184
- ECL-to-TTL interface 180, 183
- ECLinPS 118, 147, 148
- ECLinPSLite 118, 147, 148
- Edge detector circuit 381–2, 383
- Edge-triggered Flip Flop 377, 381–2, 383, 384
- Editor 540
- EEPROM 612, 624, 628
- Electrically Erasable Programmable Read Only Memory, *see* EEPROM
- Embedded microprocessors 560
- Embedded system 526–7
- Emitter coupled logic, *see* ECL

- Encoder 280–3
EPROM 612, 628
Equivalent of a Boolean expression 190
Equivalent time sampling 669, 670
Erasable Programmable Read Only Memory, *see* EPROM
Error detection and correction codes 40–3
 cyclic redundancy check code 41–2
 Hamming code 42–3
 parity code 41
 repetition code 41
Essential prime implicants 211
Even parity 41
Evolution of microprocessors 527–8
Ex-NOR gate, *see* Exclusive-NOR gate
Ex-OR gate, *see* Exclusive-OR gate
Excess-3 code 21–2
Excitation table of a flip flop 438–9
Exclusive-NOR gate 69, 80–1
Exclusive-OR gate 69, 76–7
Expanded form of Boolean expressions 206
Expanding memory capacity 632–5
 memory location expansion 634–5
 word size expansion 632–3
Exponent 12, 13
Extended Binary Coded Decimal Interchange Code,
 see EBCDIC code
Extended data output (EDO) DRAM 622
- Fairchild Advanced schottky TTL, *see* Fast TTL
Fall time 121
Fan-out 95–7, 98, 122
Fast page mode (FPM) DRAM 622
Fast TTL 116, 124, 140–1
Fault detection 653, 654
Fault isolation 653, 654
Field programmable gate array, *see* FPGA
Field programmable logic array, *see* FPLA
FireWire 638, 639
Fixed logic 299–302
Fixed logic device 300
FLASH-370 CPLD 347, 349
Flash A/D converter 500–3
Flash memory 628–9, 630
FLEX-10000 FPGA 349–50
FLEX-8000 FPGA 349–50
Flip flop 357, 373–408
 application relevant information 407–8
 applications 402–5
 D flip flop 394–5, 396, 397
 J–K flip flop 382–5, 387, 388, 391, 393, 395, 397
 master–slave flip flop 382–4, 388
R-S flip flop 373–82
 timing parameters 399–402
 toggle flip flop 390–1, 392, 393
Flip flop applications 402–5
 data storage and transfer circuits 402
 detecting the sequence of edges 404–5
 flip flop synchronization 404, 405
 frequency division and counting
 circuits 402
 switch debouncing 402–4
Flip flop synchronization 404, 405
Flip flop timing parameters 399–402
 asynchronous input active pulse width 399, 401–2
 clock pulse HIGH time 399, 401
 clock pulse LOW time 399, 401
 clock transition times 399, 402
 hold time 399, 400
 maximum clock frequency 399, 402
 propagation delay 399, 401
 set-up time 399, 400
Floating-gate transistor switch 334–5
Floating inputs 141–2
Floating-point addition 65
Floating-point arithmetic 64–6
 addition 65
 division 65–6
 multiplication 65–6
 subtraction 65
Floating-point division 65–6
Floating-point multiplication 65–6
Floating point number format 13–16
 ANSI/IEEE-754 format 13, 14–15
 IEEE-754r format 14, 15–16
 IEEE-854 standard 14, 16
Floating point numbers 12–16
 exponent 12, 13
 formats 13–16
 mantissa 12, 13
 normalized binary numbers 13
 normalized numbers 12–13
 precision 13
 range of numbers 13
 significand, *see* Mantissa
Floating-point subtraction 65
Floppy disk 647
Floppy disk controller 560, 563
Four axioms 11
FPGA 307–8, 331–3, 349–52
 application relevant information 349–52
 applications 333
 internal architecture 331–3
FPGA-applications 333
FPGA-internal architecture 331–3

- FPLA 312
 Fractional binary value 480
 Frank Gray 23
 Free-running multivibrator 365
 Frequency counter 678–84
 architecture 679–80
 continuous-count counters 682
 frequency ratio A/B 679
 microwave counters 683–4
 period 679
 phase A relative to Phase B 679
 reciprocal counters 681–2
 specifications 682–3
 time interval average 679
 time interval measurement 679
 totalize 679
 universal counter-functional modes 679
 Frequency counter specifications 682–3
 accuracy 682, 683
 bandwidth 682
 resolution 682
 sensitivity 682
 throughput 682, 683
 Frequency range-frequency synthesizer 689
 Frequency switching speed-frequency synthesizer 689
 Frequency synthesizer 684–9
 direct digital synthesis 687–9
 direct frequency synthesis 684–5
 indirect synthesis 685–7
 sampled sine synthesis, *see* Direct digital synthesis
 specifications 689
 Frequency synthesizer specifications 689
 frequency range 689
 frequency switching speed 689
 resolution 689
 signal purity 689
 Front panel controls-oscilloscope 668
 Full adder 236, 237–40, 241
 Full scale error-D/A converter, *see* Gain error-D/A converter
 Full subtractor 236, 242–4, 245
 Fundamental quantization error 497
 Fuse 300, 308, 334

 Gain drift-A/D converter 495, 496
 Gain error-A/D converter 495, 496
 Gain error-D/A converter 477
 GAL 305–6, 325–8
 GAL 22V10 342–3, 345, 346
 General purpose computer 607–8

 General purpose registers 529, 530
 Generic array logic, *see* GAL
 George Boole 3
 Glitches-decoder 431–3
 Graphic terminal 645
 Gray code 23–6
 applications 25–6
 binary-gray code conversion 24
 binary reflected gray code 23, 25
 cyclic property 23
 gray code-binary conversion 25
 N-ary gray code 25
 Gray code applications 25–6
 Gray code-binary conversion 25
 Guidelines-CMOS devices 179
 Guidelines-TTL devices 176–8

 HAL 317
 Half adder 236–7
 Half digit-digital multimeter 666–7
 Half-flash A/D converter 500, 503–4
 Half subtractor 236, 240–2
 HALT instruction 534
 Hamming (7,4) code 43
 Hamming code 42–3
 Hamming distance 42
 Hard Array Logic, *see* HAL
 Hard disk 645–7
 Hard wired control unit 531
 Hardware compilation 338
 Hardware description language, *see* HDL
 Harvard architecture 574, 575
 HDL 337, 338–40
 ABEL 338, 339
 Java HDL 339, 340
 Verilog 339–40
 VHDL 339
 Hexadecimal number system 2, 4, 5
 Hexadecimal-to-decimal conversion 6, 7
 Hex-to-binary conversion 10, 11
 Hex-to-octal conversion 10, 11
 High byte-A/D converter 499
 HIGH level input current 118, 119, 120
 HIGH level input voltage 119, 120
 High level language 540
 HIGH level off-state output current 118
 HIGH level output current 118, 119, 120
 HIGH level output voltage 120
 High power TTL 116, 124, 134–5
 Higher density BCD encoding 21
 Hold time 399, 400
 Hold time-logic analyser 695
 Hollerith code 27

- Hybrid computer 608
Hysteresis 90–1, 360
- I/O blocks 307
I/O port-microcontroller 570
I^L logic 115, 116, 172, 174–6
IC timer-based multivibrators 363–71
 astable multivibrator using timer IC 555 364–7,
 368
 monostable multivibrator using timer IC 555
 367–71
ICL 7106 513, 517–20, 521
ICL 7107 513, 517–20, 521
Idempotent laws 193
Identity laws, *see* Idempotent laws
Idle mode 579
IEEE/ANSI standard 100–1
IEEE/ANSI standard symbol-logic gates 101–2
IEEE/ANSI standard symbols 100–2
IEEE/ANSI symbology-counters 464–6
IEEE/ANSI symbology-registers 464, 466, 467
IEEE-1284 standard 640, 641–2
IEEE-488 interface 640, 641, 696–7
IEEE-754 standard 13, 14–15
IEEE-754r standard 14, 15–16
IEEE-854 standard 14, 16
Image input devices 643
Immediate addressing mode 534, 535
Impact printer 644
Implementing Boolean functions with decoders 286–8
Implementing Boolean functions with multiplexers 273–7
Implicit addressing mode 534, 537
Indexed addressing mode 534, 536–7
Indirect synthesis 685–7
Industry standard architecture bus, *see* ISA bus
Infrared port (IrDA) 569
Inhibit gate 82–3
Inkjet printer 644
Input devices 526, 642, 643
 image and video input devices 643
 keyboard devices 643
 pointing devices 643
Input port 605, 607, 637–42
 parallel ports 607, 637, 640–2
 serial ports 607, 637–40
Instruction cycle 542
Instruction register 529
Instruction set 542
Instruction set-80C51 582
Integral nonlinearity-A/D converter 497, 498
- Integrated circuit A/D converter 513–20, 521
 AD 7820 513, 516, 519
 ADC-0800 513–514, 515
 ADC-0808 513, 514–515, 516
 ADC-80/AD ADC-80 513, 515–516, 517
 ADC-84/ADC-85/AD ADC-84/AD
 ADC-85/AD-5240 513, 516, 518
 ICL 7106/ICL 7107 513, 517–20, 521
Integrated circuit D/A converter 486–9
 AD7524 486, 489
 DAC-08 486–7
 DAC-0808 486, 487
 DAC-1408/DAC-1508 486, 489
 DAC-80 486, 487, 488
Integrated circuit multivibrators 363–71
 IC based monostable multivibrator 363, 364, 365
 IC timer based multivibrator 363–71
Integrated injection logic, *see* I^L logic
Intel's Itanium 528
Intelligent terminal 645
Interface bus structure-virtual instruments 701–3
Interintegrated (I²C) bus 569, 572–3
Internal bus 642
 ISA bus 642
 local bus 642
 PCI bus 642
Internal data bus 526
International Standards Organization (ISO) 28, 37
Interrupt 526
Interrupt instruction 533, 534
Interrupts-80C51 582
Inverting type programmable integrator 491–2
Involution law 202
ISA bus 642
ISO/IEC 8859 31
ISO-10646 37, 38
ISO-8859-1 31, 38
ispLSI series CPLD 346
- Java HDL 339, 340
- J-K flip flop 382–5, 387, 388, 391, 393, 395, 397
 J-K flip flop as a toggle flip flop 391, 393
 J-K flip flop as D flip flop 395, 397
 J-K flip flop with PRESET and CLEAR inputs 382, 387
 master-slave flip flops 382–4, 388
- Johnson counter, *see* Shift counters
- Karnaugh map method 204, 216–26
 construction 216–21
 Karnaugh map for Boolean expressions with a large umber of variables 222–3
 Karnaugh maps for multi-output functions 225–6

- Key hold 595
 Keyboard devices 643
- L1 cache, *see* Primary cache
 L2 cache, *see* Secondary cache
 Large scale integration, *see* LSI
 Laser printer 644
 Latency time 647
 Left justified data-A/D converter 499
 Level amplifier-D/A converter 476
 Level-triggered flip flop 377, 381
 LIN bus 569, 573–4
 Linker 540
 Liquid crystal display 644
 Literal 190
 LOAD instruction 531
 Local bus 642
Local interconnect network, see
 LIN bus
 Logarithmic D/A converter 490
 Logic analyser 692–6
 architecture 692–5
 operational modes 692
 specifications 695–6
 Logic analyser-architecture 692–5
 clock generator 692, 694
 memory 692, 693
 probes 692, 693
 storage qualifier 692, 695
 trigger 692, 693
 user interface 692, 695
 Logic analyser-operational modes 692
 asynchronous timing mode 692
 synchronous state mode 692
 Logic analyser-specifications 695–6
 channel count 695, 696
 hold time 695
 memory depth 695, 696
 probe loading 695–6
 sample rate 695
 set-up time 695
 Logic blocks 307
 Logic devices 299
 Logic families 115–84
 BiCMOS logic 115, 116, 170–2
 characteristic parameters 118–23
 CMOS logic family 116, 117, 151–68
 CMOS subfamilies 117, 165–8
 comparison 176, 177
 ECL 116, 118, 147–50
 ECL subfamilies 118, 147–8
 guidelines 176–9
 I²L logic 115, 116, 172, 174–6
 interfacing 179–84
 NMOS logic 116, 172, 174, 175
 PMOS logic 116, 172–4
 significance 115–116
 TTL 115, 116, 124–43
 TTL subfamilies 116, 124–41
 types 116
Logic families-characteristic parameters 118–23
 disable time from the HIGH state 121
 disable time from the LOW state 121
 enable time from the HIGH state 121
 enable time from the LOW state 122
 fall time 121
 fan-out 122
 HIGH-level input current 118, 119, 120
 HIGH-level input voltage 119, 120
 HIGH-level off-state output current 118
 HIGH-level output current 118, 119, 120
 HIGH-level output voltage 120
 LOW-level input current 118, 119, 120
 LOW-level input voltage 120
 LOW-level off-state output current 119
 LOW-level output current 118, 119, 120
 LOW-level output voltage 120
 maximum clock frequency 122
 noise margin 122–3
 power dissipation 122
 propagation delay 121
 rise time 121
 speed-power product 122
 supply current 121
Logic families-comparison 176, 177
Logic families-interfacing 179–84
 CMOS-to-ECL interface 183, 184
 CMOS-to-TTL interface 179–80, 181
 ECL-to-CMOS interface 183, 184
 ECL-to-TTL interface 180, 183
 TTL-to-CMOS interface 180, 182
 TTL-to-ECL interface 180, 183
Logic gate-applications 102–7
Logic gates 69–109
 AND gate 69, 71, 73–5
 AND-OR-INVERT gates 87, 89, 90
 ANSI symbols 101–2
 application relevant information 107–9
 applications 102–7
 Exclusive NOR gate 69, 80–1
 Exclusive OR gate 69, 76–7
 fan-out 95–7
 gates with open collector/drain outputs 85–7
 inhibit gate 82–3

- Logic gates (*Continued*)
NAND gate 69, 79, 85
NOR gate 69, 79–80, 85, 86
NOT gate 69, 71, 75, 76
OR-AND-INVERT gates 87–8, 89
OR gate 69, 71–2
Schmitt gates 88–91, 92
special output gates 91, 93
tristate logic gates 87, 88, 89
universal gates 85
Logic instructions 531, 533
Logic operations 47
Logic probe 691–2
Look-ahead carry 255
Look ahead carry generator 254–7, 258, 259
Look-up table, *see* LUT
Low byte-A/D converter 499
Low frequency function generator 490, 492–3
LOW level input current 118, 119, 120
LOW level input voltage 120
LOW level off-state output current 119
LOW level output current 118, 119, 120
LOW level output voltage 120
Low Power Schottky TTL 116, 124, 136–7
Low Power TTL 116, 124, 133–4
LPC3000 family of microcontrollers 591–2
LSI 172, 183, 184
LUT 331–3
- Mach series CPLD 344–6
Machine control instructions 531, 534
Machine cycle 542
Macrocell 328
Magnetic ink character recognition, *see* MICR
Magnetic storage devices 645–8
 floppy disks 647
 magnetic hard disks 645–7
 magnetic tapes 647–8
Magnetic tape 647–8
Magneto-optical storage devices 648
Magnitude comparator 261–4
Mainframe computer 609
Mantissa 12, 13
Mask programmable gate arrays, *see* MPGA
Mask programmable PLA 312
Mask-programmed ROM 624–6, 627
Master–slave flip flops 382–4, 388
Math coprocessor 560, 562
Mathematics of logic 3
Matrix keypad 594, 595–6
MAX-5000 CPLD family 343–4
MAX-7000 CPLD family 343–4, 347
MAX-9000 CPLD family 343–4
Maximum clock frequency-flip flop 399, 402
MC1600, *see* MECL III
MC6800 528, 544–6
 addressing modes 545
 instruction set 545–6
 registers 545
MC68000 528, 549–51
 addressing modes 549, 551
 instruction set 549
 registers 549
MC68010 528
MC68012 528
MC68020 551, 553–4
 addressing modes 553
 instruction set 553
 registers 553
MC68030 551, 554–5
MC6809 528
MCFXXX family of microcontrollers 591
MCORE family of microcontrollers 591
MECL 10100 series 148
MECL 10200 series 148
MECL 10E 118, 147, 148
MECL 10H 118, 147, 148
MECL 10K 118, 147, 148
MECL-I 118, 147
MECL-II 118, 147
MECL-III 118, 147
Medium scale integration, *see* MSI
Memory 525, 605, 606–7, 610–635, 645–50
Memory address register 529–30
Memory buffer register 529–30
Memory cell-DRAM 619–20
Memory cell-flash memory 628–9
Memory depth-logic analyser 695, 696
Memory direct addressing mode, *see* Absolute addressing mode
Memory length-digital storage oscilloscope 675–6
Memory location expansion 634–5
Memory-mapped I/O 576
Metal oxide semiconductor logic, *see* MOS logic
MICR 643
Microcoded control unit 531
Microcomputer 609
Microcontroller 527, 565–601
 16-bit microcontrollers 588–90
 32-bit microcontrollers 590–2
 8-bit microcontrollers 580–8
analogue-to-digital converters 569–70
application relevant information 580–92
applications 567
architecture 574–9

- Microcontroller (*Continued*)
- central processing unit 567, 568, 569
 - controller area network bus 569, 573
 - counters/timers 569, 570–1
 - I/O ports 569, 570
 - interfacing peripheral devices 592–601
 - interintegrated circuit (I²C) Bus 569, 572–3
 - local interconnect network bus 569, 573–4
 - peripheral components 569–74
 - power-saving modes 579–80
 - random access memory 569
 - read only memory 569
 - serial communication interfaces 569, 571–2
 - special function registers 569
- Microcontroller applications 567
- Microcontroller architecture 574–9
- accumulator-based architecture 577
 - architecture to access memory 574–5
 - Harvard architecture 574, 575
 - mapping special function registers into memory space 574, 576
 - memory-mapped I/O 576
 - pipeline architecture 577, 579
 - processor architecture 574, 577–9
 - register-based architecture 577, 578
 - separate I/O and memory spaces 576
 - stack-based architecture 577, 578
 - Von Neumann architecture 574, 575
- Microcontroller-A/D converter interface 600, 601
- Microcontroller-D/A converter interface 600, 601
- Microcontroller-keyboard interface 594–6
- Microcontroller-LCD display interface 598–600
- Microcontroller-LED interface 592–3
- Microcontroller-relay interface 593–4
- Microcontroller-seven segment display interface 596–8, 599
- Microprocessor 525–63
- 16-bit microprocessors 547–51
 - 32-bit microprocessors 551–7
 - 8-bit microprocessors 541–6
 - addressing modes 534–7
 - ALU 528, 529
 - CISC processors 528, 531, 541
 - control unit 528, 531
 - embedded microprocessors 560
 - evolution 527–8
 - inside a microprocessor 528–31
 - instructions 531–4
 - Pentium series 557–60
 - peripheral devices 560–3
 - programming 540
 - registers 528, 529–30
- RISC processors 528, 531, 541
- selection criteria 537–9
- Microprocessor instructions 531–4
- arithmetic instructions 531, 532–3
 - branch instructions, *see* Control transfer instructions
 - control transfer instructions 531, 533–4
 - data transfer instructions 531
 - logic instructions 531, 533
 - machine control instructions 531, 534
 - program control instructions, *see* Control transfer instructions
- Microprocessor selection criteria 537–9
- Microwave counter 683–4
- Miller-killer configuration 141
- Minicomputer 609
- Minterm expression, *see* Sum-of-products Boolean expression
- Minuend 48
- MOCRE family of microcontrollers 591
- MOD number, *see* Modulus of a counter
- Modulus of a counter 413
- Monoshot, *see* Monostable multivibrator
- Monostable multivibrator 357, 360–2, 363, 364, 365, 367–71
- digital IC based monostable multivibrator 363, 364, 365
 - monostable multivibrator using Timer IC 555 367–71
 - retriggerable monostable multivibrator 361–2, 363, 365
- Monostable multivibrator using Timer IC 555 367–71
- Monotonocity-D/A converter 476, 478, 479
- Moore's law 527
- MOS logic 115
- MOVE instruction 530, 531
- MPC500 family of microcontrollers 591
- MPGA 331
- MSI 172, 183, 184
- Multi-master support arrangement 573
- Multi-output function 212, 225
- Multi slope A/D converter 500, 506, 508–9
- Multiple keys 595
- Multiplexer 269–80
- cascading 280
 - implementing Boolean functions with multiplexers 273–7
 - inside a multiplexer 271–3
- multiplexers for parallel-to-serial data conversion 277
- Multiplier-D/A converter 490
- Multipliers 260–1
- Multiplying type D/A converter 479–80

- Multivibrator 357–71
 astable multivibrator 357, 362–3, 364–7, 368
 bistable multivibrator 357–8
 integrated circuit multivibrators 363–71
 monostable multivibrator 357, 360–2, 363, 364,
 365, 367–71
 retriggerable monostable multivibrator 361–2,
 363, 365
 Schmitt trigger 358–60
MUX, *see* Multiplexer
- NAND gate 69, 79, 85
N-ary Gray code 25
N-key rollover 595
Negative AND 70
Negative edge triggered flip flops 381–2, 383, 384
Negative logic system 69–70
Negative NAND 70
Netlist 338
Nexperia platform 591
NMOS logic 116, 172, 174, 175
Non-Boolean Gray code 25
Non-inverting type programmable integrator 492
Non linearity error-D/A converter 477
Nonimpact printer 644
Nonlinearity-A/D converter 495, 497
Nonlinearity-D/A converter 476, 478
NOP instruction 534
NOR gate 69, 79–80, 85, 86
Normalized binary numbers 13
Normalized numbers 12–13
NOT gate 69, 71, 75, 76
Number representation in binary 5–6
 1's complement 5, 6
 2's complement 5, 6
 sign-bit magnitude 5
Number systems 1, 2–5
 binary number system 2, 3–5
 common terms 4–5
 decimal number system 1, 2–3, 5
 hexadecimal number system 2, 4, 5
 octal number system 2, 4, 5
- OCR 643
Octal bus transceiver 563
Octal number system 2, 4, 5
Octal-to-binary conversion 9, 11
Octal-to-decimal conversion 6–7
Octal-to-hex conversion 10, 11
Odd parity 41
Offset drift-A/D converter 495, 496
Offset error-A/D converter 495, 496
Offset error-D/A converter 477
- OLED display 644, 645
OLMC 327–8
OMR 643
Open collector gate-TTL 131–2
Open-collector output 85–7
Open-drain output 85–7
Operational modes-logic analyser 692
Operations with ‘0’ and ‘1’ 192–3
Optical character recognition, *see* OCR
Optical mark recognition, *see* OMR
Optical storage devices 648–50
 CD-R 649
 CD-ROM 648, 649
 CD-RW 649
 DVD-ROM 649, 650
 USB flash drive 650
 WORM 649
Optimized reconfigurable cell array, *see* ORCA
OR-AND-INVERT gates 87–8, 89
OR gate 69, 71–2
ORCA 350
Organic light-emitting diode display, *see* OLED
 display
Oscilloscope 668–78
 analog vs digital oscilloscopes 672
 analogue oscilloscopes 669, 672, 673–4
 CRT storage type analogue oscilloscopes 669
 digital oscilloscopes 669–72
 digital storage oscilloscopes 668, 669–70, 674–7
 front panel controls 668
 probes 677–8
 specifications 672–7
 types 669–72
Oscilloscope probe 677–8
 probe compensation 677–8
Oscilloscope specifications 672–7
 accuracy-analogue oscilloscope 673, 674
 accuracy-digital storage oscilloscope 674, 676–7
 analogue oscilloscope specifications 673–4
 analogue storage oscilloscope specifications 674
 bandwidth and rise time-analogue oscilloscope
 673
 bandwidth and sampling rate-digital storage
 oscilloscope 674–5
 digital storage oscilloscope specifications 674–7
 memory length-digital storage oscilloscope 675–6
 resolution-digital storage oscilloscope 674, 676–7
 vertical sensitivity-analogue oscilloscope 673, 674
Output devices 526, 642, 643–5
 computer terminals 645
 displays 644–5
 plotters 644
 printers 644

- Output logic macrocell, *see* OLMC
 Output port 605, 607, 637–42
 parallel ports 607, 637, 640–2
 serial ports 607, 637–40
 Oversampling 509, 510
- Packed BCD numbers 21
 PAL 300, 304–5, 317–23
 architecture 319–20, 321
 numbering system 320–3
 PAL architecture 319–20, 321
 PAL numbering system 320–3
 Parallel binary adder 254
 Parallel counter, *see* Synchronous counter
 Parallel-in parallel-out shift register 449, 453, 456
 Parallel-in serial-out shift register 449, 452, 454, 455
 Parallel port 607, 637, 640–642
 IEEE-1284 640, 641–2
 IEEE-488 640, 641
 small computer system interface 640, 641
 Parallel-to-serial data conversion 277
 Parity bit 41
 Parity checking circuits 104, 105
 Parity code 41
 Parity generation circuits 104
 pASIC-2 FPGA 352
 pASIC FPGA 352
 PC incrementer 569
 PCI bus 642
 PECL 149
 PEEL 306
 Pentium D 559–60
 Pentium Extreme Edition 559–60
 Pentium II 528, 559
 Pentium III 528, 559
 Pentium IV 528, 559
 Pentium M 559–60
 Pentium Pro 528, 559
 Pentium series of microprocessors 528, 557–60
 Celeron microprocessor 560
 Pentium D microprocessor 559–60
 Pentium Extreme Edition microprocessor 559–60
 Pentium II microprocessor 528, 559
 Pentium III microprocessor 528, 559
 Pentium IV microprocessor 528, 559
 Pentium M microprocessor 559–60
 Pentium Pro microprocessor 528, 559
 salient features 558
 Xeon microprocessor 560
 Percentage resolution-BCD input D/A converter 482
 Perfect induction 192
 Peripheral control interconnect bus, *see* PCI bus
 Peripheral devices 560–3
 clock generator 560, 563
 DMA controller 560, 561–2
 floppy disk controller 560, 563
 math coprocessor 560, 562
 octal bus transceiver 563
 programmable communication interface 560, 562
 programmable CRT controller 560, 562–3
 programmable interrupt controller 560, 561
 programmable keyboard/display interface 560, 562
 programmable peripheral interface 560, 561
 programmable timer/counter 560, 561
 Personal computer 610
 Pi (Π) nomenclature 207–8
 PIC 16X84 family of microcontrollers 585–6, 587
 Pipeline A/D converter, *see* Half-flash A/D converter
 Pipeline architecture 577, 579
 PLA 302–4, 312–314
 Place and route 338
 Plasma display 644, 645
 PLD 299–352
 application relevant information 340–52
 complex programmable logic device 306, 328–31
 design and development 337–8
 field programmable gate array 307–8, 331–3, 349–52
 generic array logic 305–6, 325–8
 programmable array logic 300, 304–5, 317–23
 programmable interconnect technologies 333–7
 programmable logic array 302–4, 312–314
 programmable ROMs 302, 308–11
 programming languages 338–40
 PLICE 336
 Plotter 644
 pLSI series CPLD 346
 PMOS logic 116, 172–4
 inverter 173–4
 NOR gate 173–4
 Pointing devices 643
 POP instruction 530
 Positive ECL, *see* PECL
 Positive edge triggered flip flops 381–2, 383, 384
 Positive feedback 90
 Positive logic system 69–70
 Positive NOR 70
 Positive OR 70
 Postulates of Boolean algebra 192
 Power consumption-microprocessor 537, 538
 Power dissipation 122
 Power down mode 579
 Power modes-80C51 582
 Power saving modes 579–80

- Power supply decoupling 142–3
PowerPC 620 528
PowerPC RISC microprocessor 557
Practical D/A converter 476
Precision 13
Precomposed characters 38
PRESET input 398
Presetable counters 426–8
Presetable counters-variable modulus 428, 429
Primary cache 622
Primary memory 525, 606, 610, 611–35
 cache memory 606, 622, 623
 dynamic RAM 612, 619–22
 erasable ROM 612, 624, 628
 expanding memory capability 632–5
 flash memory 628–9, 630
 mask-programmed ROM 624–6, 627
 programmable ROM 612, 624, 626–7, 628
RAM applications 622, 623
random access memory 606, 610, 611–22
read only memory 606, 610, 611, 612, 622–31
ROM applications 629–31
ROM architecture 623–4
static RAM 612–619
 types of ROM 624–9, 630
Prime implicant table 211
Prime implicants 208
Printer 644
Priority encoder 281–3
Probe compensation 677–8
Probe loading-logic analyser 695–6
Processor architecture 574, 577–9
Product-of-sums Boolean expression 87, 204, 205–6
Program address register 569
Program control instructions, *see* Control Transfer instructions
Program counter 529, 569
Programmable array logic, *see* PAL
Programmable communication interface 560, 562
Programmable CRT controller 560, 562–3
Programmable Electrically Erasable Logic, *see* PEEL
Programmable integrator 491–2
Programmable interconnect technologies 333–7
 antifuse 300, 308, 334, 335–6, 337
 floating-gate transistor switch 334–5
 fuse 300, 308, 334
 static RAM-controlled programmable switches 334, 335, 336
Programmable interconnects 307
Programmable interrupt controller 560, 561
Programmable keyboard/ display interface 560, 562
Programmable logic 299–302
Programmable logic array, *see* PLA
Programmable logic devices, *see* PLD
Programmable logic devices-advantages 301–2
Programmable logic devices-disadvantages 301–2
Programmable logic hardware-design 337–8
Programmable peripheral interface 560, 561
Programmable Read Only Memory, *see* PROM
Programmable ROM, *see* PROM
Programmable timer/counter 560, 561
Programming languages 338–40
 ABEL-hardware description language 338, 339
 Java HDL 339, 340
 Verilog 339–40
 VHDL-VHSIC hardware description language 339
PROM 302, 308–11, 612, 624, 626–7, 628
Propagation delay-flip flop 399, 401
Propagation delay-ripple counters 412–413
PS/2 connector 640
Pull-up resistor 85, 86, 87
PUSH instruction 530

Quad-precision format 15
Quad slope A/D converter 509
Quantization error-A/D converter 495, 496–7
Quasi-stable state 360, 361, 362
Quaternary Gray code 25
Quine–McCluskey tabular method 204, 208–14
 for multioutput functions 212–214

R-S flip flop 373–82
 clocked R-S flip flop 377–81
 edge triggered flip flops 377, 381–2, 383, 384
 level triggered flip flops 377, 381
 R-S flip flop with active HIGH inputs 375–7, 378
 R-S flip flop with active LOW inputs 374–5, 377, 378
 R-S flip flop with active HIGH inputs 375–7, 378
 R-S flip flop with active LOW inputs 374–5, 377, 378
 R-S latch 374
Race problem 384
Radix 2
Radix-2 3
RAM 569, 606, 610, 611–22
 applications 622, 623
 asynchronous SRAM 612, 613–616, 617
 cache memory 606, 622, 623
 DRAM architecture 620
 dynamic RAM 612, 619–22
 static RAM 612–619

- RAM (*Continued*)**
- synchronous SRAM 612–613, 616–619
 - types of DRAM 622
- RAM address register 569
- Random access memory, *see* RAM
- Random interleaved sampling 670–1
- Range of numbers 13
- Read only memory, *see* ROM
- Read/write memory 525
- Real time sampling 669–70
- Reciprocal counter 681–2
- Reconfigurable computing 333
- Reduced Instruction Set Computer, *see* RISC
- Redundancy law, *see* Absorption law
- Reflect-and-prefix method 23, 24
- Reflected Gray code 23
- Register based architecture 577, 578
- Register direct addressing mode 534, 535, 536
- Register file 528, 529–30
 - buffer register 529–30
 - general purpose registers 529, 530
 - instruction register 529
 - program counter 529
 - stack pointer 529, 530
 - status register 529, 530
 - temporary registers 529, 530
- Register indirect addressing mode 534, 535–6
- Registered outputs 317, 319
- Registers 411
- Relative addressing mode 534, 537
- Repeated add and right-shift algorithm 58, 59–60
- Repeated left-shift and add algorithm 58, 59
- Repeated right-shift and subtract algorithm 61
- Repeated subtract and left-shift algorithm 62
- Repetition code 41
- Resistive divider network 474–5
- Resistor transistor logic, *see* RTL
- Resolution-A/D converter 495–6
- Resolution-D/A converter 476–7
- Resolution-digital storage oscilloscope 674, 676–7
- Resolution-frequency counter 682
- Resolution-frequency synthesizer 689
- Retriggerable monostable multivibrator 361–2, 363, 365
- Return instruction 533, 534
- Right justified data-A/D converter 499
- Ring counter 459–60
- Ripple counter 411–419
 - binary ripple counter 413–419
 - binary ripple counter with a modulus of less than 2^N 416–418
 - propagation delay in ripple counters 412–413
 - ripple counters in IC form 418, 419
- Ripple counters in IC form 418, 419
- RISC 528, 531, 541
- Rise time-analogue oscilloscope 673
- ROM 308, 569, 606, 610, 611, 612, 622–31
 - applications 629–31
 - architecture 623–4
 - erasable ROM 612, 628
 - flash memory 628–9, 630
 - mask-programmed ROM 624–6, 627
 - programmable ROM 612, 624, 626–7, 628
 - types 624–9, 630
- ROM architecture 623–4
- ROM read operation-timing diagram 625
- Rotate operation 533
- RS 232 port 569, 638–9
- RTL 116, 117
- Sample and hold circuit-A/D converter 498
- Sample rate-logic analyser 695
- Sampled sine synthesis, *see* Direct digital synthesis
- Sampling frequency-A/D converter 495, 496
- Sampling oscilloscope 670
- Sampling rate-digital storage oscilloscope 674–5
- Schmitt gates 88–91, 92
- Schmitt trigger 358–60
- Schmitt trigger-transfer characteristics 360
- Schottky TTL 116, 124, 135–6
- SCI 569, 571–2
- SCSI standard 640, 641
- Secondary cache 622
- Secondary memory, *see* Secondary storage
- Secondary storage 525, 606, 610, 611, 645–50
 - CD-R 649
 - CD-ROM 648, 649
 - CD-RW 649
 - DVD-ROM 649, 650
 - floppy disks 647
 - magenta-optical storage devices 648
 - magnetic hard disks 645–7
 - magnetic storage devices 645–8
 - magnetic tapes 647–8
 - optical storage devices 648–50
 - USB flash drive 650
 - WORM 649
- Seek time 647
- Sensitivity-frequency counter 682
- Sequential logic circuit 233
- Sequential single sample technique 670
- Sequential sweep equivalent time sampling 670, 671
- Serial communication interface, *see* SCI
- Serial counter, *see* Ripple counter
- Serial-in parallel-out shift register 449, 452

- Serial-in serial-out shift register 449–51
Serial peripheral interface, *see* SPI
Serial port 607, 637–40
 FireWire 638, 639
 PS/2 connector 640
 RS-232C port 638–9
 universal serial bus 639–40
Set-up time-flip flop 399, 400
Settling time-D/A converter 477–8
Setup time-logic analyser 695
Seven-segment display code 38–40
Seven-segment displays 38–40
Shift counters 411, 459, 460–1
Shift operation 533
Shift register 447–58
 application relevant information 466, 468–9
 bidirectional shift register 455
 IEEE/ANSI symbology 464, 466, 467
 parallel-in parallel-out shift register 449, 453, 456
 parallel-in serial-out shift register 449, 452, 454, 455
 serial-in parallel-out shift register 449, 452
 serial-in serial-out shift register 449–51
 universal shift register 455–8
Shift register counters 459–61
 ring counter 459–60
 shift counter 459, 460–1
Sigma (Σ) nomenclature 207–8
Sigma-delta A/D converter 500, 509–11
Sign-bit magnitude 5
Signal purity-frequency synthesizer 689
Significand, *see* Mantissa
Simple programmable logic device, *see* SPLD
Simplification techniques 204–26
 Karnaugh-map method 204, 216–26
 Quine–McCluskey tabular method 204, 208–14
Simulation 338
Simultaneous A/D converter, *see* Flash A/D converter
Single-extended Precision format 14
Single precision format 14, 15
Single slope A/D converter 500, 506–7
Small computer system interface standard, *see* SCSI standard
Small scale integration, *see* SSI
Smart terminal 645
Special function registers-microcontroller 567, 569
Special output gates 91, 93
Special purpose computer 608
Speed-power product 122
SPI 569, 572
SPLD 306, 328, 340–3, 344, 345
Square wave oscillators 105
SRAM 612–619
 asynchronous SRAM 612, 613–616, 617
 synchronous SRAM 612–613, 616–619
SRAM-based FPGA 349
SSI 172, 183, 184
Stack-based architecture 577, 578
Stack pointer 529, 530
Standard TTL 116, 124, 125–33
 AND gate 128–9
 AND-OR-INVERT gate 131
 characteristic parameters 127
 circuit operation 125–6
 Exclusive-OR gate 130
 NAND gate 125–7
 NOR gate 127–8
 NOT gate 127, 128
 open collector gate 131–2
 OR gate 129–30
 totem pole output stage 126–7, 131–2
 tristate gate 132–3
State transition diagram of a counter 438, 439, 440
Static RAM, *see* SRAM
Static RAM-controlled programmable switches 334, 335, 336
Status register 529, 530
Status register-A/D converter 500
Stop clock mode 579
STORE instruction 531
Strobe signal 432
Subtraction of larger-bit binary numbers 52–5
Subtraction using 2's complement arithmetic 53–5
Subtrahend 48
Successive approximation A/D converter 500, 505–6
Sum 47–8, 238
Sum-of-products Boolean expression 87, 204–5
Supercomputer 609, 610
Switch bounce phenomenon 402–3
Switch debouncing 402–4
Switch matrix 306
Synchronous communication interface 571–2
Synchronous counter 413, 423–5
Synchronous DOWN counter 423
Synchronous DRAM 622
Synchronous inputs 398
Synchronous SRAM 612–613, 616–619
Synchronous state mode-logic analyser 692
Synthesis 338
Synthesized function generator 684, 689–90
Synthesized signal generator 684, 690

- T-flip flop 390–1, 392, 393
 J-K flip flop as a toggle flip flop 391, 393
 Temporary registers 529, 530
 Term 190
 Ternary Gray code 25
 Test and measuring equipment 665–703
 arbitrary waveform generator 690, 691
 computer-instrument interface standards 696–7
 digital multimeter 665–7, 668
 frequency counter 678–84
 frequency synthesizers 684–9
 logic analyser 692–6
 logic probe 691–2
 oscilloscope 668–78
 oscilloscope probes 677–8
 synthesized function generators 684, 689–90
 synthesized signal generator 684, 690
 virtual instrumentation 697–703
 Theorems of Boolean algebra 192–202
 Throughput-frequency counter 682, 683
 Time interval average 679
 Time interval measurement 679
 Time period 679
 Timer input block 570, 571
 Timer output block 570, 571
 Toggle flip flop, *see* T-flip flop
 Totalize 679
 Totem pole output stage 126–7, 131–2
 Tracking type A/D converter 500, 505
 Transceiver 98–100
 Transistor logic, *see* TTL
 Transposition theorem 200–1
 TRICORE family of microcontrollers 592
 Triple slope A/D converter 508–9
 Tristate gate-standard TTL 132–3
 Tristate logic gates 87–8, 89
 Troubleshooting arithmetic circuits 663
 Troubleshooting guidelines 653–65
 faults external to digital integrated circuits 655–6
 faults internal to digital integrated circuits 654–5
 troubleshooting arithmetic circuits 663
 troubleshooting memory devices 664–5
 troubleshooting RAM devices 664
 troubleshooting ROM devices 664–5
 troubleshooting sequential logic circuits 659
 Troubleshooting memory devices 664–5
 troubleshooting RAM devices 664
 troubleshooting ROM devices 664–5
 Troubleshooting RAM devices 664
 Troubleshooting ROM devices 664–5
 Troubleshooting sequential logic circuits 659
 Truth table 70–1
 TTL 115, 116, 124–43
 advanced low power Schottky TTL 116, 124,
 137–9
 advanced Schottky TTL 116, 124, 137,
 139–40
 current transients 142
 Fairchild Advanced Schottky TTL 116, 124,
 140–1
 floating and unused inputs 141–2
 guidelines to using TTL devices 176–8
 high power TTL 116, 124, 134–5
 low power Schottky TTL 116, 124, 136–7
 low power TTL 116, 124, 133–4
 power supply decoupling 142–3
 Schottky TTL 116, 124, 135–6
 standard TTL 116, 124, 125–33
 subfamilies 116, 124–41
 TTL subfamilies 116, 124–41
 advanced low power Schottky TTL 116, 124,
 137–9
 advanced Schottky TTL 116, 124, 137, 139–40
 fairchild advanced Schottky TTL 116, 124, 140–1
 high power TTL 116, 124, 134–5
 low power Schottky TTL 116, 124, 136–7
 Schottky TTL 116, 124, 135–6
 standard TTL 116, 124, 125–33
 TTL-to-CMOS interface 180, 182
 TTL-to-ECL interface 180, 183
 Two-key lock-out 595
 Types of A/D converter 500–11
 Types of D/A converter 479–80
 Types of ROM 624–9, 630
 ULSI 183, 184
 Ultra large scale integration, *see* ULSI
 Unconditional jump instruction 533–4
 Unicode 28, 37–8
 Unicode consortium 28, 37
 Unipolar mode operation-A/D converter 499
 Universal counter 679
 frequency ratio A/B 679
 functional modes 679
 period 679
 phase A relative to phase B 679
 time interval average 679
 time interval measurement 679
 totalize 679
 Universal gates 85
 Universal serial bus port, *see* USB port
 Universal shift register 455–8
 Unpacked BCD numbers 21
 Unweighted binary code 23
 UP counter 423–4

- UP/DOWN counters 425–6
US ASCII-8 28
USB flash drive 650
USB port 569, 639–40
- Variables 190
Verilog 339–40
Vertical sensitivity-analogue oscilloscope 673, 674
Very High-Speed Integrated Circuit, *see* VHSIC
Very large scale integration, *see* VLSI
VHDL-VHSIC hardware description language 339
VHSIC 339
Video input devices 643
Virtual instrument components 700–3
 computer and display 700
 instrument hardware 703
 interface bus structure 701–3
 software 700
Virtual instrumentation 697–703
 components 700–3
 graphical programming technique as a virtual instrument 699
 reconfigurable building blocks as a virtual instrument 699–700
set of instruments as a virtual instrument 698
software graphical panel as a virtual instrument 698, 699
use 698
VLSI 183, 184
Voltage output version-DAC-80 D/A converter 487, 488
Voltage steering mode-D/A converter 480, 481–2
- Von Neumann architecture 574, 575
Von Neumann bottleneck 575
- Weighted BCD code 20
Weighted code 19
Windows-1252 31
WIRE-AND connection 86–7, 131
Word length 4
Word size, *see* Word length
Word size expansion 632–3
Workstation 610
WORM 649
Write once read many, *see* WORM
- XC-2000 FPGA 349
XC-3000 FPGA 349
XC-4000 FPGA 349, 350
XC-5000 FPGA 349
XC-7000 series CPLD 347–8
XC-800 family of microcontrollers 586–8
 XC-866 586–8
 XC-886/888 586–8
XC-8100 FPGA 349
XC-9500 CPLD 347–8
Xeon processor 560
- Z80 microprocessor 528, 546
 instruction set 546
 registers 546
Z800 microprocessor 528
Z8000 microprocessor 528
Z80000 microprocessor 528
Zero scale error-D/A converter, *see* Offset error-D/A converter