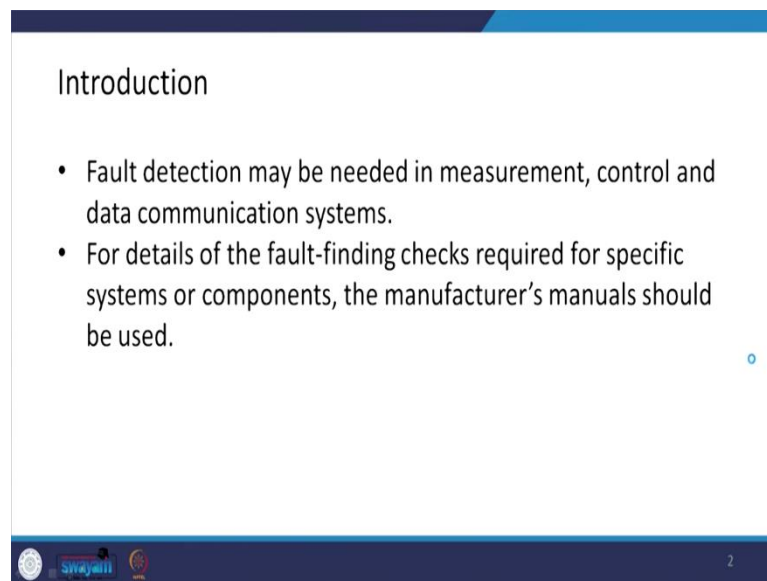


Mechatronics
Prof. Pushparaj Mani Pathak
Department of Mechanical and Industrial Engineering
Indian Institute of Technology, Roorkee

Lecture - 35
Fault Finding

I welcome you all to this NPTEL online certification course on Mechatronics. Today we are going to talk about Fault Finding. In a Mechatronics system, you see that it is very important that in case of a certain fault, we are able to identify the faults, at what place the fault has occurred, so and then we could take the remedial action. A piece of knowledge about the finding of the fault is essential from the maintenance point of view, that is, the maintenance of a mechatronic system. So, fault detection may be needed in the measurement unit control unit or data communication system.

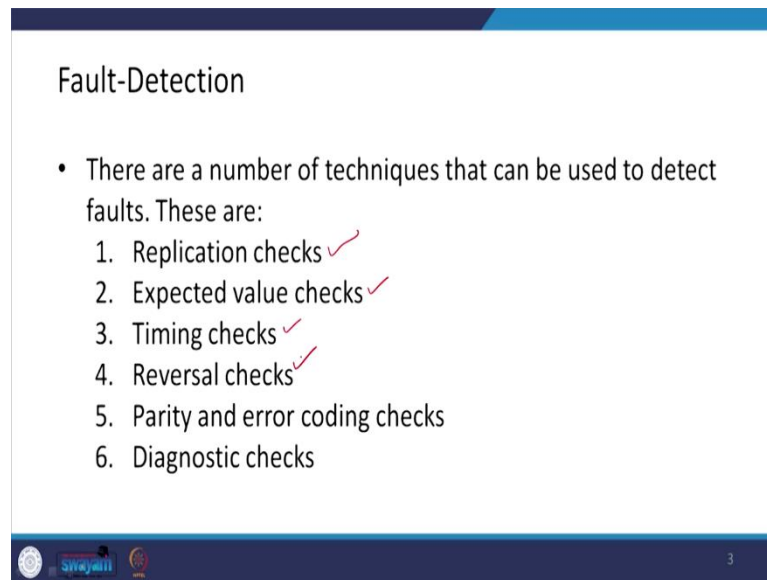
(Refer Slide Time: 01:24)



The slide is titled "Introduction" and contains two bullet points. The first bullet point states that fault detection may be needed in measurement, control, and data communication systems. The second bullet point states that for details of the fault-finding checks required for specific systems or components, the manufacturer's manuals should be used. The slide has a blue header and footer. The footer contains the NPTEL logo, the word "Swayam", and the number "2".

For details of the fault-finding check, it is always advised to go with the manufacturer manuals instructions, and that always helps in checking for the required fault. There are a number of techniques that can be used to detect faults.

(Refer Slide Time: 02:04)



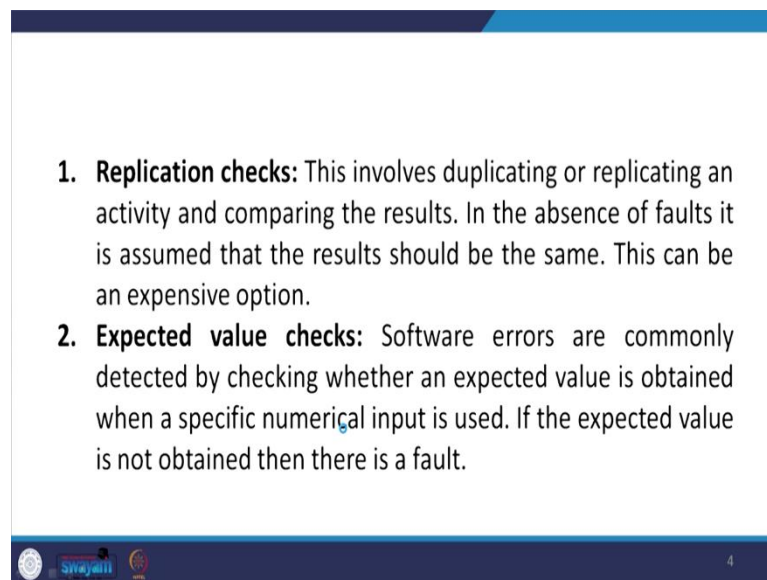
The slide is titled "Fault-Detection" and lists six techniques for detecting faults. The list is as follows:

- There are a number of techniques that can be used to detect faults. These are:
 1. Replication checks ✓
 2. Expected value checks ✓
 3. Timing checks ✓
 4. Reversal checks ✓
 5. Parity and error coding checks
 6. Diagnostic checks

The slide footer contains logos for Swayam and a small number '3'.

These are, , replication checks, expected value checks, timing checks, reversal checks, parity and error coding checks, and diagnostic checks.

(Refer Slide Time: 02:12)



The slide provides detailed descriptions for the first two techniques:

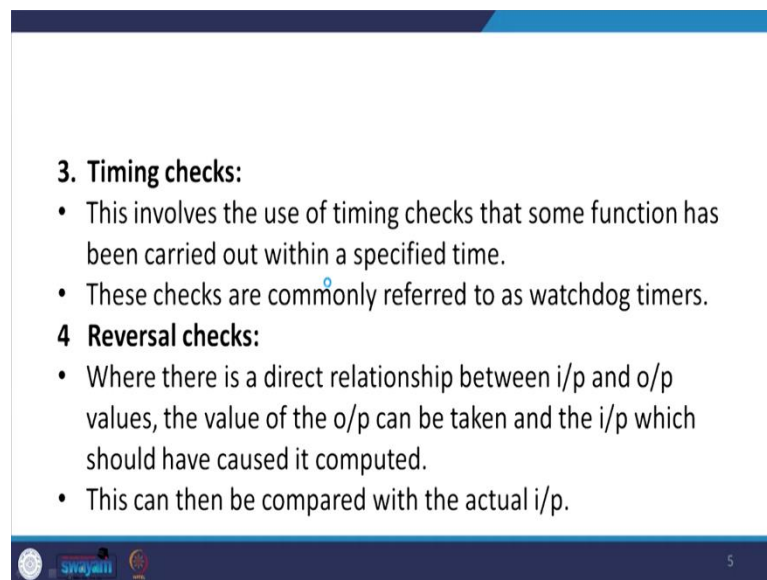
- 1. Replication checks:** This involves duplicating or replicating an activity and comparing the results. In the absence of faults it is assumed that the results should be the same. This can be an expensive option.
- 2. Expected value checks:** Software errors are commonly detected by checking whether an expected value is obtained when a specific numerical input is used. If the expected value is not obtained then there is a fault.

The slide footer contains logos for Swayam and a small number '4'.

Now, let us look at the first one, the replication checks. The replication check involved duplicating or replicating an activity and comparing the result. In the absence of faults, it is assumed that the result should be the same. So, if you are duplicating something, if there is no fault, the result is going to be the same. But the question is that this could be an expensive option because you have to replicate it.

The next way is the Expected value check. It is usually for the software errors we go for this type of check. So, software errors are commonly detected by checking whether an expected value is obtained when specific numerical input is used. If the expected value is not obtained, and then naturally, we know that this system has got a fault.

(Refer Slide Time: 03:39)



3. Timing checks:

- This involves the use of timing checks that some function has been carried out within a specified time.
- These checks are commonly referred to as watchdog timers.

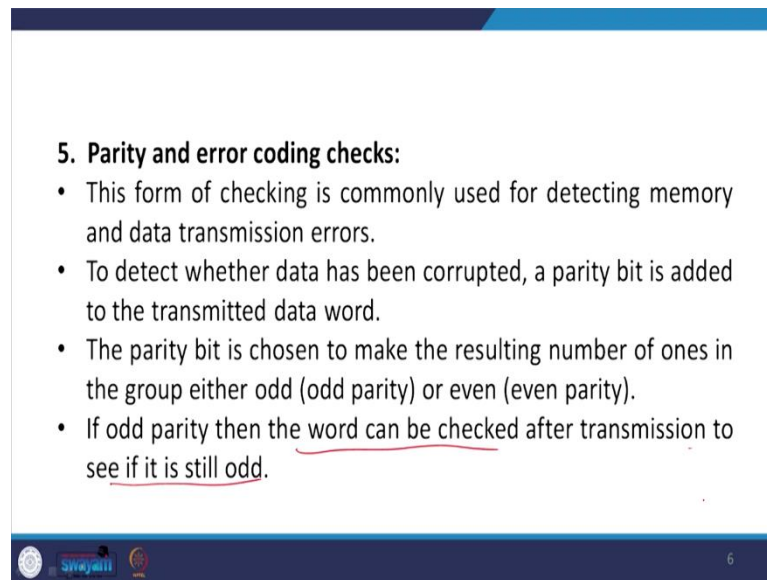
4 Reversal checks:

- Where there is a direct relationship between i/p and o/p values, the value of the o/p can be taken and the i/p which should have caused it computed.
- This can then be compared with the actual i/p.

The next way is by the timing checks, and this involves the use of a timing check that some function has been carried out within a specified time. These checks are commonly referred to as the watchdog timers.

Then we have the reversal check where there is a direct relationship between the input and output values, the values of the output can be taken, and the input which should have caused it is computed. And this can then be compared with the actual input, and this type of check is what is called the reversal check.

(Refer Slide Time: 04:28)



5. Parity and error coding checks:

- This form of checking is commonly used for detecting memory and data transmission errors.
- To detect whether data has been corrupted, a parity bit is added to the transmitted data word.
- The parity bit is chosen to make the resulting number of ones in the group either odd (odd parity) or even (even parity).
- If odd parity then the word can be checked after transmission to see if it is still odd.

The slide features a blue header and footer. The footer contains the Swajati logo and the number 6.

Then next is the parity and error coding check, so this form of checking is commonly used for detecting memory and data transmission errors. So, to detect whether data has been corrupted, a parity bit is added to the transmitted data word, and the parity bit is choosing to be making the resulting number of ones in the group either odd, which we call the odd parity, or it could be even which we call as the even parity.

So, if odd parity, then the word can be checked after transmission to see if it is still odd and if it is not odd, then we can that the data transmission error has occurred. Then we have the diagnostic checks, so diagnostic checks are used to test the behavior of a component in a system. So, input is applied to a component, and the output is compared with those we should occur. So, this is how we do the diagnostic check.

Now, let us look at the watchdog timer. A watchdog timer is a timer that the system must reset before it times out. And if the timer is not reset in time, then an error is assumed to have occurred. I could get an example from the PLC, that is, a PLC with a watchdog timer for an operation involving the movement of a piston in a cylinder.

(Refer Slide Time: 06:18)

Watchdog Timer

- A watchdog timer is basically a timer that the system must reset before it times out. If the timer is not reset in time then an error is assumed to have occurred.
- Example: A PLC with a watchdog timer for an operation involving the movement of a piston in a cylinder.

8

This is how it could be represented using the ladder diagram. So, you have to start, so a cylinder is moving plus and then above the if this is plus and this is another one another cylinder movement, then we can have a timer and if the from this timer we could have an alarm.

(Refer Slide Time: 06:44)

Common Hardware Faults

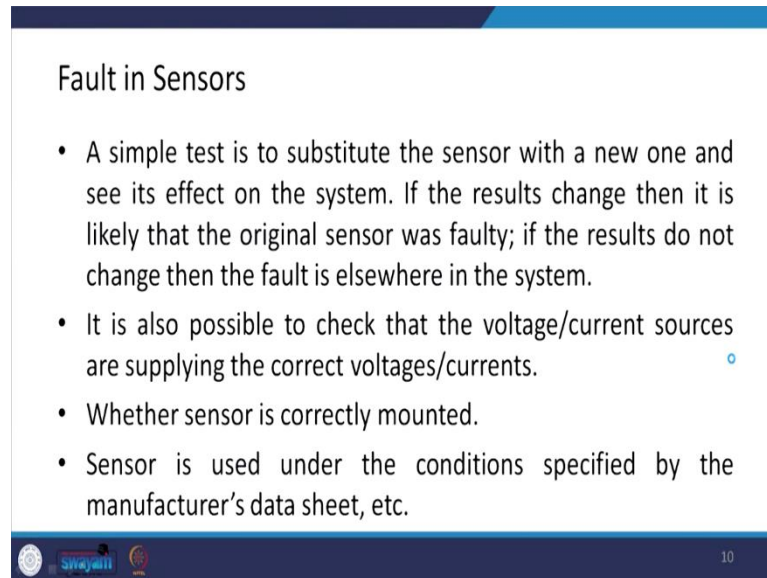
- The following are some of the commonly encountered faults that can occur with specific types of components and systems
 1. Fault in Sensors
 2. Faulty Switches and Relays
 3. Fault in Motors
 4. Fault in Hydraulic and Pneumatic Systems

9

Then common hardware faults. The following are some of the commonly encountered faults that occur with a specific type of component and system. As we are talking about the mechatronics system, the hardware faults could be there, in your actuators, or it could

be there in your sensors, or it could be said in switches and relays, or it could be a fault in the hydraulic and pneumatic systems; which we are using either for actuation purpose, or we are using for generating some of the control actions. So, these are the principal hardware type of faults that could be there. Now, let us look at the fault in sensors. A simple test is to substitute the sensor with a new one and see the effect on the system.

(Refer Slide Time: 07:52)



The slide is titled "Fault in Sensors" and contains a bulleted list of diagnostic steps. The slide has a dark blue header and footer. The footer includes the Swayam logo and the number 10.

Fault in Sensors

- A simple test is to substitute the sensor with a new one and see its effect on the system. If the results change then it is likely that the original sensor was faulty; if the results do not change then the fault is elsewhere in the system.
- It is also possible to check that the voltage/current sources are supplying the correct voltages/currents.
- Whether sensor is correctly mounted.
- Sensor is used under the conditions specified by the manufacturer's data sheet, etc.

If the resulting change, then it is likely that the original sensor was faulty, and if the result does not change, then the fault is elsewhere in the system. It is also possible to check that the voltage or current sources are supplying the correct voltage or currents as the case may be. And we could also see whether my sensor is correctly mounted, and also the sensor is used under the condition is specified by the manufacturer's datasheet. So, this we have to ensure.

(Refer Slide Time: 08:31)

Faulty Switches and Relays

- Dirt and particles of waste material between switch contacts are a common source of incorrect functioning of mechanical switches.
- A voltmeter used across a switch should indicate the applied voltage when the contacts are open and very nearly zero when they are closed.
- If a relay fails to operate then a check can be made for the voltage across the coil. If the correct voltage is present then coil continuity can be checked with an ohmmeter. If there is no voltage across the coil then the fault is likely to be the switching transistor used with the relay.

Swajati 11

Next, let us look at the faulty switches and relays; see if dirt and particles of waste material between the switches contacts are a common source of incorrect functioning of the mechanical switches. A voltmeter used across a switch should indicate the applied voltage when the contacts are open and very nearly zero when they are closed. And if a relay fails to operate, then a check can be made on the voltage across the coil, and if the correct voltage is present, then the coil continuity can be checked with an ohmmeter. If there is no voltage across the coil, then the fault is likely to be there in the switching transistor used with the relays.

(Refer Slide Time: 09:26)

Fault in Motors

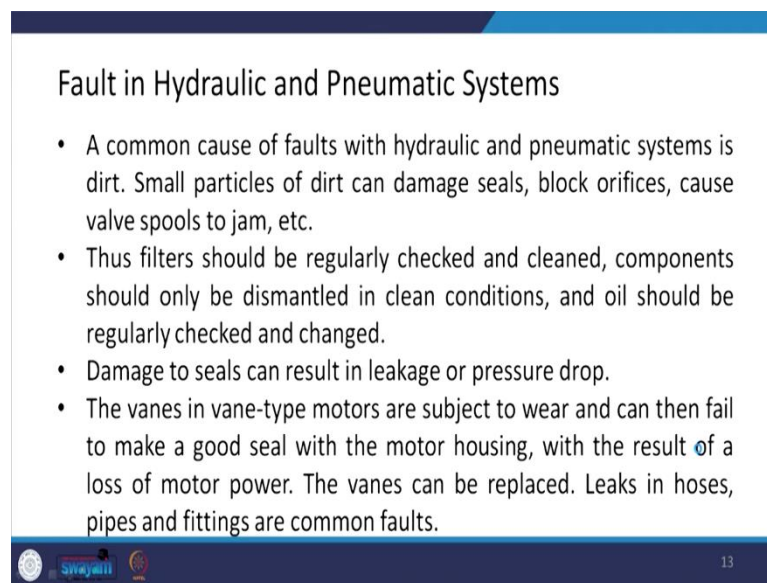
- Maintenance of both DC and AC motors involves correct lubrication.
- With DC motors the brushes wear and can require changing.
- A single phase capacitor start AC motor that is sluggish in starting probably needs a new starting capacitor.
- The three-phase induction motor required periodic lubrication.

Swajati 12

Next, let us look at the possible faults in the motor. So, maintenance of both DC and AC motors involves the correct type of lubrication. So, whatever motor we are using in our mechatronic system, whether it is AC or DC, its maintenance involves the correct lubrication. So, with the DC motor, the brushes wear and can require changing until and unless you are using the Brushless DC motor. In a Conventional DC motor, you know that commutation brushes are used. So, we need to periodically change these brushes because they wear. A single-phase capacitor start AC motor that is sluggish in starting probably needs a new starting capacitor in the case of AC motor. The three-phase induction motor requires periodic lubrication, so this is through these steps we can shape maintain our motors.

Then what are the likely faults in the hydraulic and pneumatic systems? A common cause of faults with the hydraulic and pneumatic system is dirt. A small particle of dirt can damage the seals which are provided to prevent leakage.

(Refer Slide Time: 11:09)



Fault in Hydraulic and Pneumatic Systems

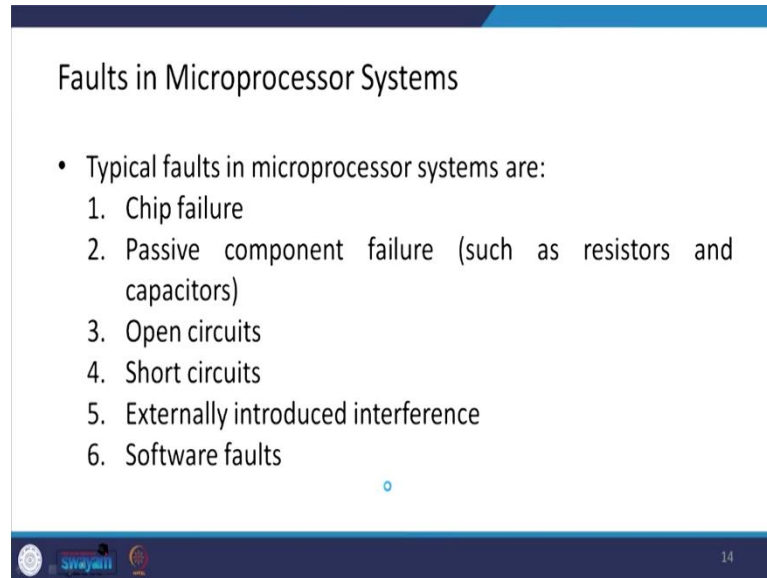
- A common cause of faults with hydraulic and pneumatic systems is dirt. Small particles of dirt can damage seals, block orifices, cause valve spools to jam, etc.
- Thus filters should be regularly checked and cleaned, components should only be dismantled in clean conditions, and oil should be regularly checked and changed.
- Damage to seals can result in leakage or pressure drop.
- The vanes in vane-type motors are subject to wear and can then fail to make a good seal with the motor housing, with the result of a loss of motor power. The vanes can be replaced. Leaks in hoses, pipes and fittings are common faults.

13

They can block the orifices, and they can cause the valve spool, etcetera to jam. Thus filters should be regularly checked and clean for dirt. And components if you are doing any maintenance activity and you are dismantling the components, so these components should only be dismantled in very clean condition, and oil should be regularly checked and changed. Damage to seals can result in leakage or pressure drop, as I said. The vanes in a vane-type of motor are subjected to wear and can then fail to make a good seal with the

motor housing with the result of loss of motor power. The vanes can be replaced leaks in hoses pipes, and fitting is the other common fault. Next, let us look at the faults in mechatronic systems.

(Refer Slide Time: 12:28)



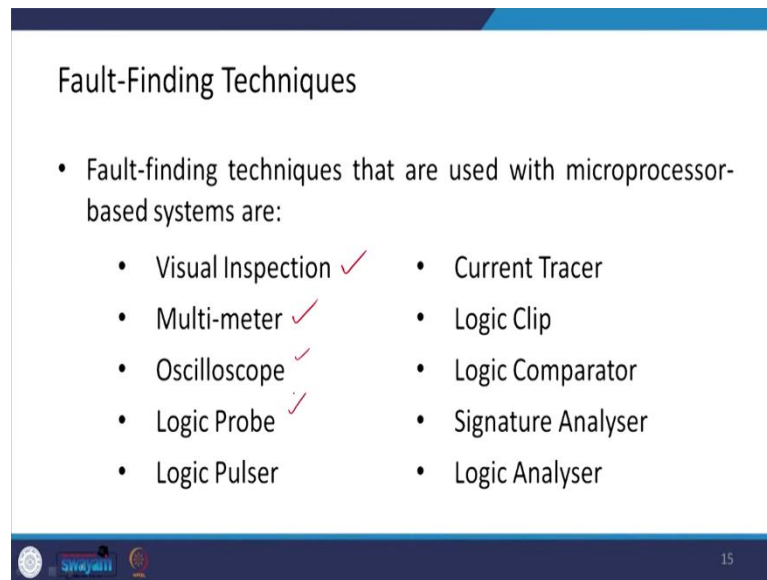
Faults in Microprocessor Systems

- Typical faults in microprocessor systems are:
 1. Chip failure
 2. Passive component failure (such as resistors and capacitors)
 3. Open circuits
 4. Short circuits
 5. Externally introduced interference
 6. Software faults

14

The mechatronics system is composed of actuators sensors, a signal conditioning unit, digital controller controllers, and a display unit. We have seen the issues possible issues with actuators sensors, and then there could also be situations where faults are there in the mechatronic microprocessor system. So, the typical faults in microprocessor systems are, you have the chip failure or the passive component failure. Such as resistors and capacitors or your circuit is open, or you have a short circuit, or externally some induced interference is there, or there could be software faults. So, these are the typical faults in the microprocessor system.

(Refer Slide Time: 13:52)



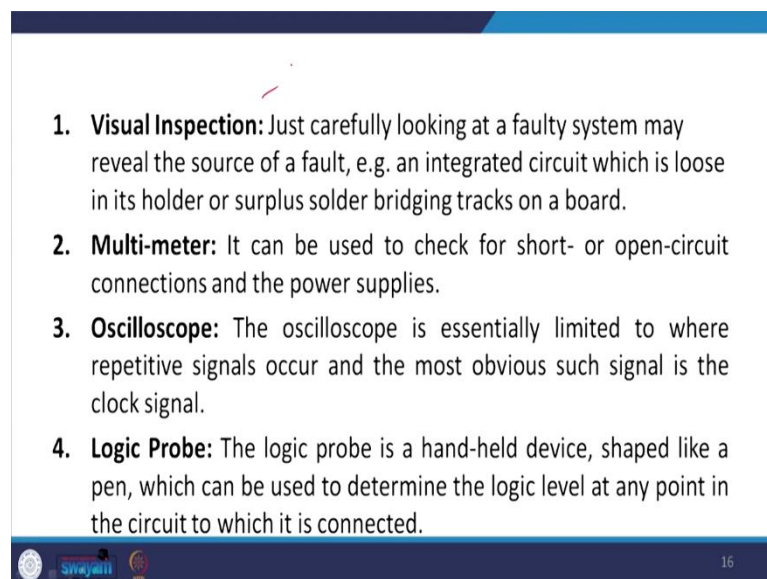
Fault-Finding Techniques

- Fault-finding techniques that are used with microprocessor-based systems are:
 - Visual Inspection ✓
 - Multi-meter ✓
 - Oscilloscope ✓
 - Logic Probe ✓
 - Logic Pulser
 - Current Tracer
 - Logic Clip
 - Logic Comparator
 - Signature Analyser
 - Logic Analyser

swayam 15

Now, what are the fault-finding techniques? So, the fault-finding techniques that are used with microprocessor-based systems are visual inspection, and we could have the multimeter, oscilloscope, logic probe, logic pulser, then-current tracer, logic clip, logic comparator, signature analyzer, and the logic analyzer. These are the techniques that could be used for finding out the faults in the microprocessor-based system. Let us look at these; the visual inspection just carefully looking at the faulty system may reveal a source of the fault.

(Refer Slide Time: 14:37)



1. **Visual Inspection:** Just carefully looking at a faulty system may reveal the source of a fault, e.g. an integrated circuit which is loose in its holder or surplus solder bridging tracks on a board.
2. **Multi-meter:** It can be used to check for short- or open-circuit connections and the power supplies.
3. **Oscilloscope:** The oscilloscope is essentially limited to where repetitive signals occur and the most obvious such signal is the clock signal.
4. **Logic Probe:** The logic probe is a hand-held device, shaped like a pen, which can be used to determine the logic level at any point in the circuit to which it is connected.

swayam 16

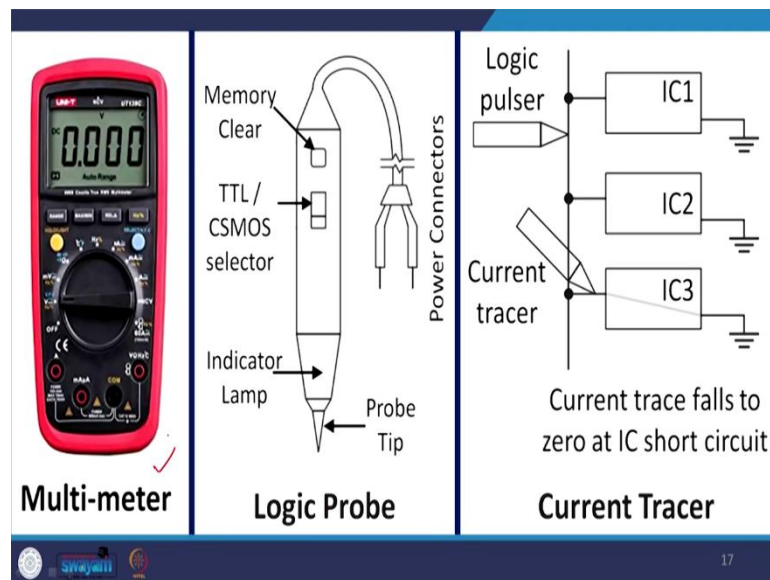
For example, an IC, which is loose in its holder or surplus solder bridging tracks on a track or on a board.

Then we could have a multimeter it can be used to check for short or open circuit connections and the power supplies.

The oscilloscope is essentially limited to where repetitive signals occur, and the most obvious signal is the clock signal.

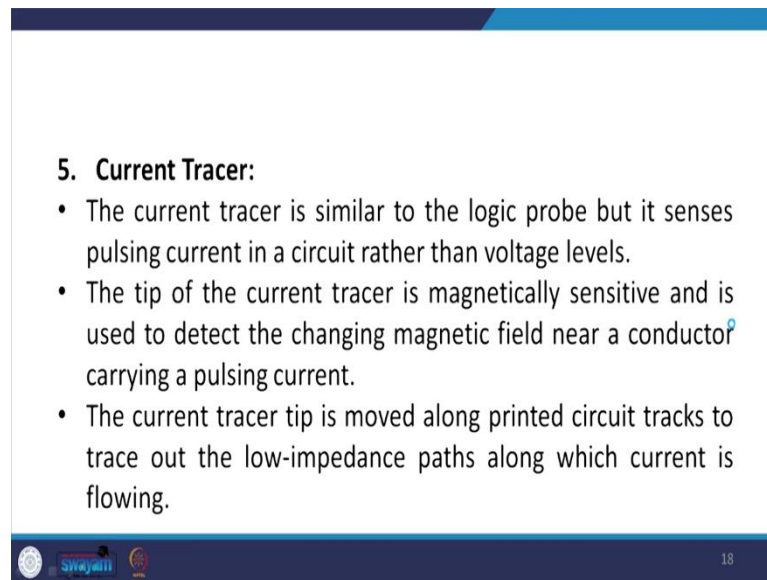
Then we could have the logic probe, and the logic probe is a handheld device shaped like a pen that can be used to determine the logic level at any point in the circuit to which it is connected.

(Refer Slide Time: 15:23)



This is how the multimeter looks like this is how the logic probe looks like, this is the probe tip you have an indicator lamp, you have the TTL or CSMOS selector, and then you have the memory clear, and then you have the power connections over here. The current tracer falls to 0 at IC circuit shorts. So, you have the logic pulser, and you could have the current tracer. The current tracer, as I said current tracer is very similar to the logic probe, but it senses pulsing currents in a circuit rather than the voltage level, the tip of the current tracer is magnetically sensitive and is used to detect the changing magnetic field near a conductor carrying a pulsing current.

(Refer Slide Time: 16:27)



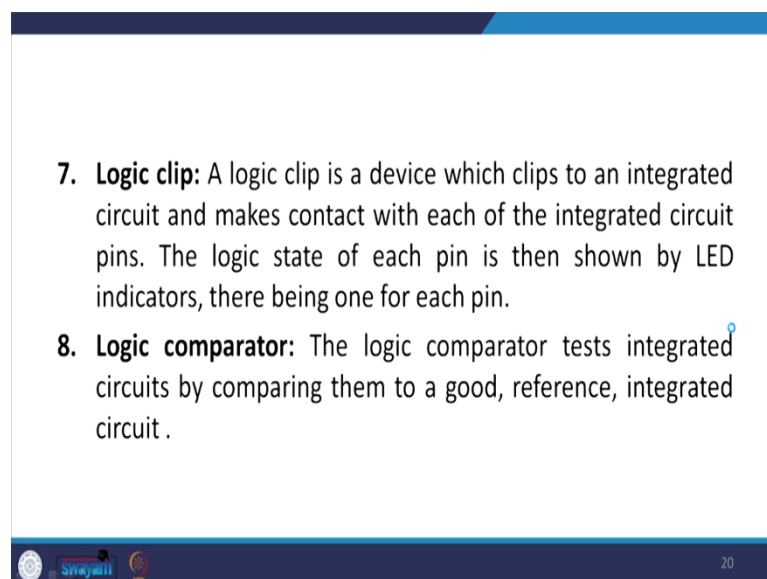
5. Current Tracer:

- The current tracer is similar to the logic probe but it senses pulsing current in a circuit rather than voltage levels.
- The tip of the current tracer is magnetically sensitive and is used to detect the changing magnetic field near a conductor carrying a pulsing current.
- The current tracer tip is moved along printed circuit tracks to trace out the low-impedance paths along which current is flowing.

18

The current tracer tip is moved along printed circuit tracks to trace out the low-impedance path along which the current is flowing. Coming back to the logic pulser, the logic pulser is a handheld pulse generator shaped like a pen that is used to inject control pulses into the circuit. The pulser probe tip is pressed against a node in the circuit, and the button on the probe press to generate a pulse. It is often used with the logic probe to check the functions of the logic gates.

(Refer Slide Time: 17:01)



7. Logic clip: A logic clip is a device which clips to an integrated circuit and makes contact with each of the integrated circuit pins. The logic state of each pin is then shown by LED indicators, there being one for each pin.

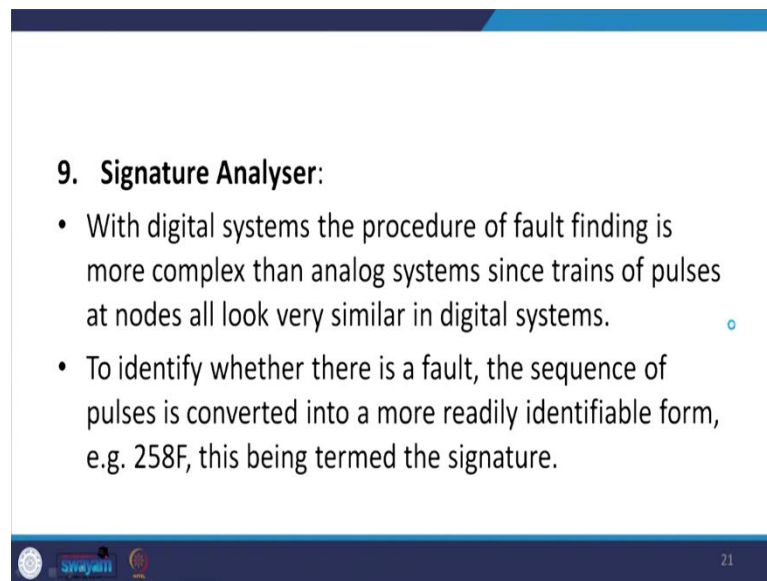
8. Logic comparator: The logic comparator tests integrated circuits by comparing them to a good, reference, integrated circuit .

20

The logic clip logic clip is a device that clips to an IC and makes contact with each of the IC's pins. The logic state of each pin is then shown by the LED indicator, and there is one for each pin.

Next, let see the Logic comparator. The logic comparator tests the integrated circuit by comparing them to a good reference integrated circuit.

(Refer Slide Time: 17:35)



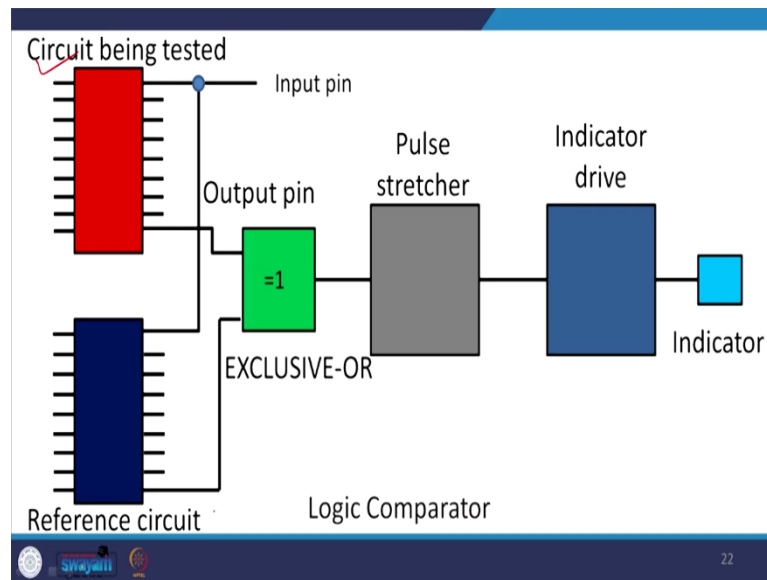
9. Signature Analyser:

- With digital systems the procedure of fault finding is more complex than analog systems since trains of pulses at nodes all look very similar in digital systems.
- To identify whether there is a fault, the sequence of pulses is converted into a more readily identifiable form, e.g. 258F, this being termed the signature.

21

Signature analyzer with a digital system, the procedure of fault-finding is more complex than the analog system since train trains of pulses at nodes all look very similar in the digital system. So, to identify whether there is a fault, the sequence of pulses is converted into a more readily identifiable form. We could have to 258F this being termed as the signature.

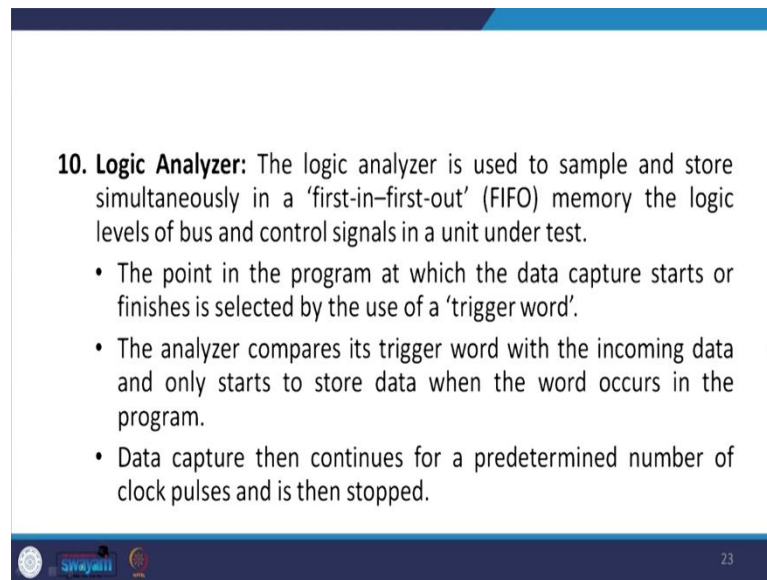
(Refer Slide Time: 18:19)



This is the logic comparator that could work like this. So, this is the circuit being tested, and this is your reference circuit, so you have an input pin. So, you give the input to the circuit being tested as well as to the reference circuit, and you have the output from the circuit being tested, and you have the output from the reference circuit. And this is provided to the exclusive-or gate, here then you have the pulse stretcher, and then you have the indicator drive, and then it gives you the indication.

Next, let us see the logic analyzer; the logic analyzer is used to sample and store simultaneously in a first in first out or FIFO memory the logic levels of bus and control signals in a unit under test. The point in the program at which the data capture starts or finishes is selected by the use of a trigger word.

(Refer Slide Time: 19:27)



10. Logic Analyzer: The logic analyzer is used to sample and store simultaneously in a 'first-in-first-out' (FIFO) memory the logic levels of bus and control signals in a unit under test.

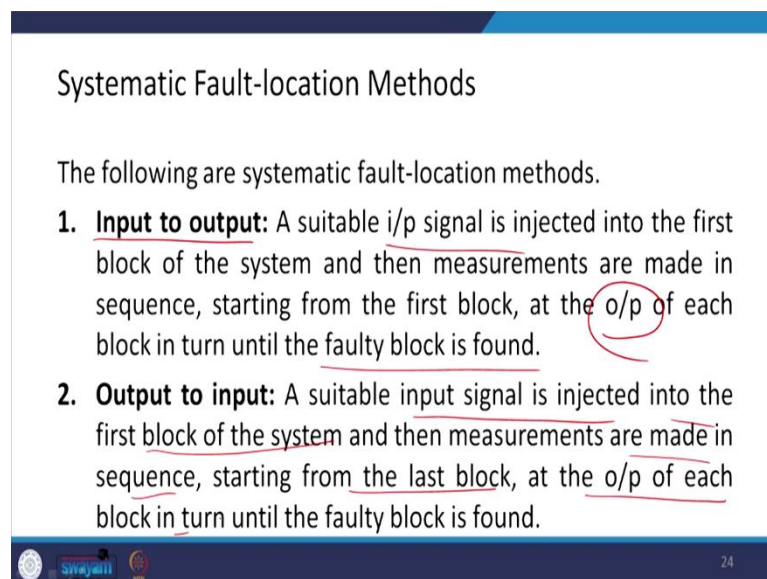
- The point in the program at which the data capture starts or finishes is selected by the use of a 'trigger word'.
- The analyzer compares its trigger word with the incoming data and only starts to store data when the word occurs in the program.
- Data capture then continues for a predetermined number of clock pulses and is then stopped.

23

The analyzer compares its trigger word with the incoming data and only starts to store data when the word occurs in the program. Data capture then continues for a predetermined number of clock pulses, and then it stops. So, this is how the logic analyzer works.

Now, let us look at some of the systematic fault location methods. So, there are various methods, for example, input to output.

(Refer Slide Time: 20:06)



Systematic Fault-location Methods

The following are systematic fault-location methods.

1. **Input to output:** A suitable i/p signal is injected into the first block of the system and then measurements are made in sequence, starting from the first block, at the o/p of each block in turn until the faulty block is found.
2. **Output to input:** A suitable input signal is injected into the first block of the system and then measurements are made in sequence, starting from the last block, at the o/p of each block in turn until the faulty block is found.

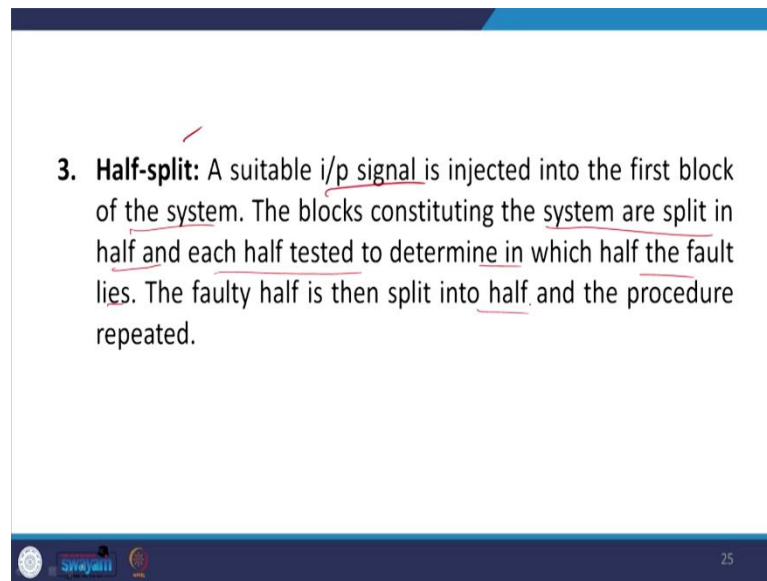
24

So, in this method, what we do is that a suitable input signal is injected into the first block of the system, and then the measurements are made in sequence starting from the first

block till the output of each block in turn until the faulty block is found. So, this is the input to output.

We could also have the output to input. So, in the case of output to the input of what is done is that a suitable input signal is injected into the first block of the system, and then measurements are made in sequence starting from the last block; we do it in a reverse way at the output of each block in turn until the faulty block is found.

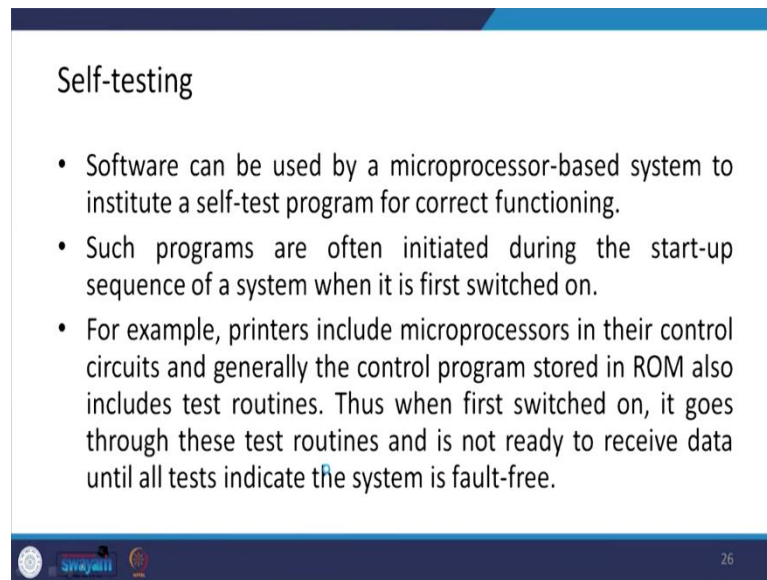
(Refer Slide Time: 20:49)



The other method is the half split. So, as the name indicates here, a suitable input signal is injected in the first block of the system. The block constituting the systems are split in half and each half tested to determine in which half the fault lies. The faulty half is then split into half, and the procedure is repeated until and unless we zero down to the actual block.

There now, let us look at the self-testing. Software's can be used by a microprocessor-based system to institute a self-test program for correct functioning. Such programs are often initiated during the start-up sequence of a system when it is first switched on.

(Refer Slide Time: 21:52)



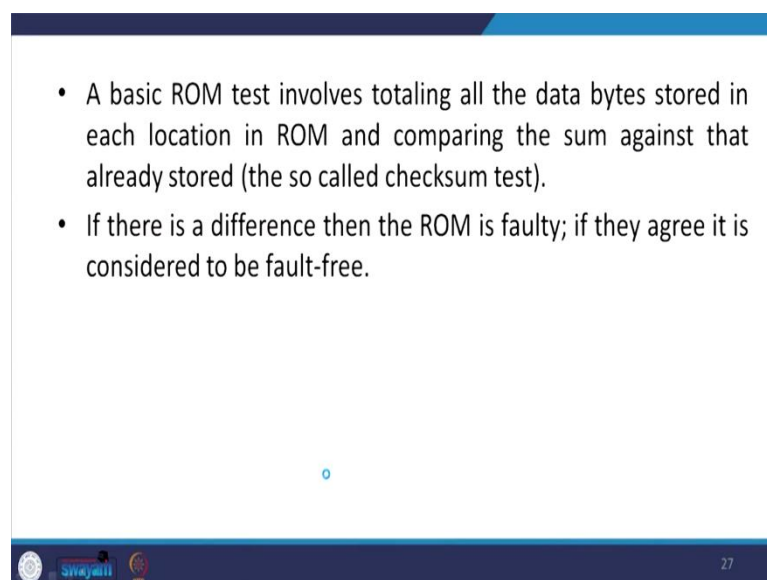
Self-testing

- Software can be used by a microprocessor-based system to institute a self-test program for correct functioning.
- Such programs are often initiated during the start-up sequence of a system when it is first switched on.
- For example, printers include microprocessors in their control circuits and generally the control program stored in ROM also includes test routines. Thus when first switched on, it goes through these test routines and is not ready to receive data until all tests indicate the system is fault-free.

26

For example printer, we all know a printer includes a microprocessor printer which we use in our offices. So, such a printer includes a microprocessor in their control circuit, and generally, the control program stored in the ROM also includes the test routine. Thus when we switch on the printer, it goes through these test routine and is not ready to receive data until all test indicates the system is fault-free.

(Refer Slide Time: 22:44)



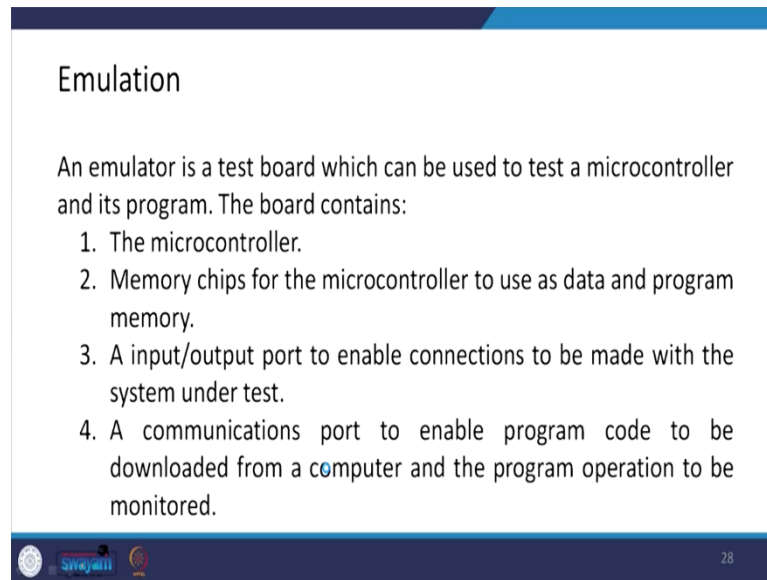
- A basic ROM test involves totaling all the data bytes stored in each location in ROM and comparing the sum against that already stored (the so called checksum test).
- If there is a difference then the ROM is faulty; if they agree it is considered to be fault-free.

27

A basic ROM test involves totaling all the data bytes stored in each location in ROM and comparing the sum against that already stored. And this is also what is called the checksum

test, and this checksum test many times we also carry out in checking our pen drives, if you are copying some data from a pen drive to another pen drive, we carry out that check to see whether the complete data has been copied or not. If there is a difference, then the ROM is faulty, and if they agree, it is considered to be fault-free.

(Refer Slide Time: 23:19)



The slide is titled "Emulation" and contains the following text and list:

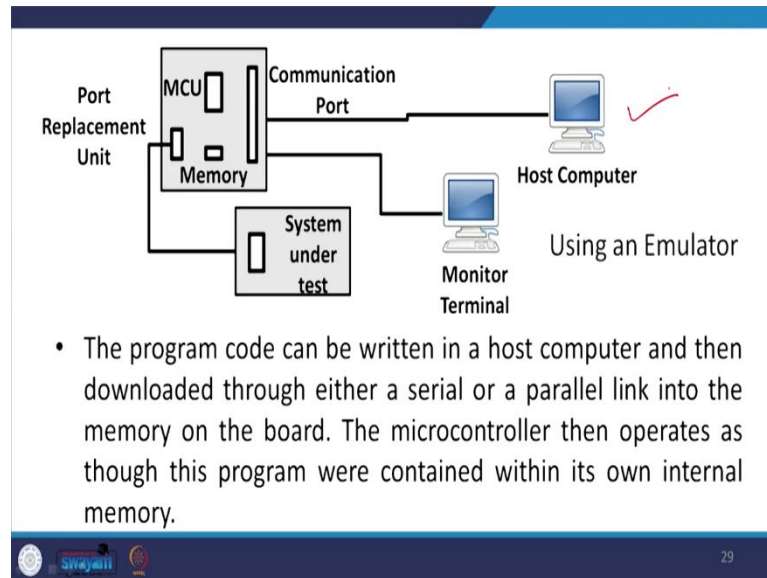
An emulator is a test board which can be used to test a microcontroller and its program. The board contains:

1. The microcontroller.
2. Memory chips for the microcontroller to use as data and program memory.
3. A input/output port to enable connections to be made with the system under test.
4. A communications port to enable program code to be downloaded from a computer and the program operation to be monitored.

The slide footer includes a logo on the left, the text "Syrjani" in the center, and the number "28" on the right.

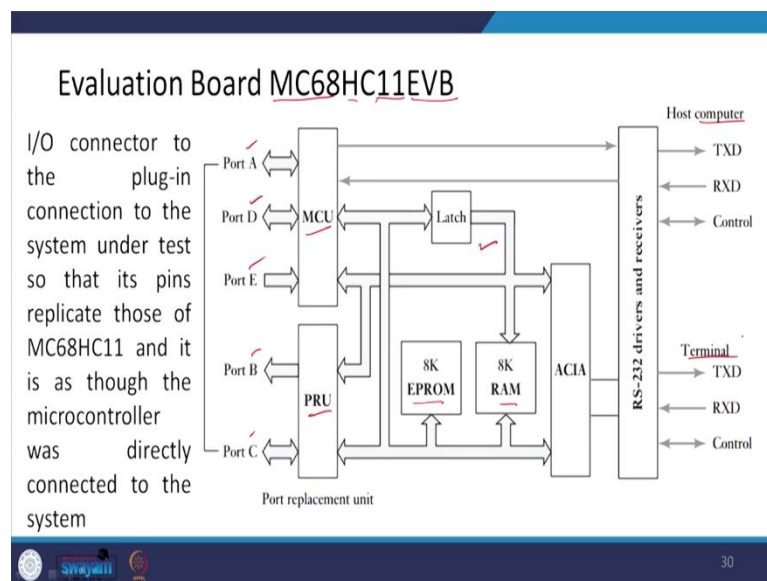
Emulation: an emulator is a test board that can be used to test a microcontroller and its program. This board contains the microcontroller, the memory chip for the microcontroller to use as data and program memory, an input-output port to enable connections to be made with the system under test, and a communication port to enable program code to be downloaded from a computer and the program operation to be monitored.

(Refer Slide Time: 23:55)



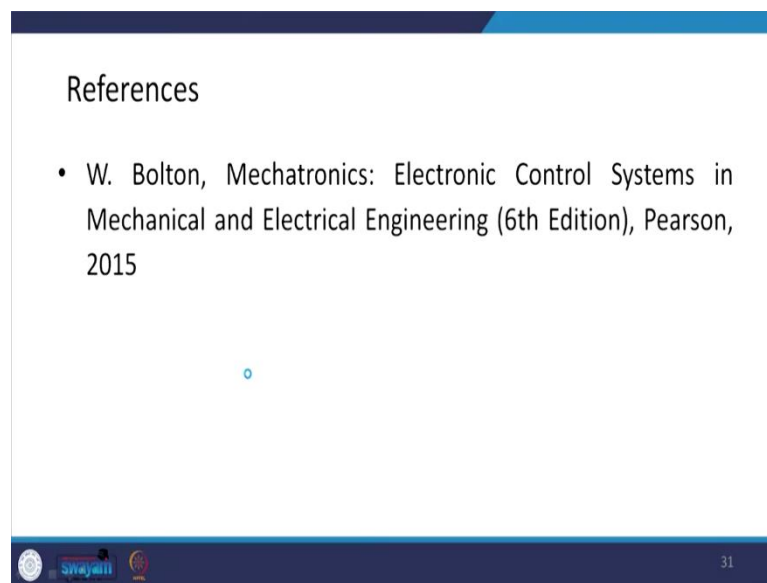
So, you have a host computer, you have a monitor terminal over here, and suppose this is the system under test. So, you so the host computer and the monitor terminal can be connected through the communication port over here. So, you have the port replacement unit, there is a memory, and there is an MCU. So, the program code can be written in the host computer and then downloaded through either a serial or parallel link into the memory on the board. The microcontroller then operates as though this program were contained within its own internal memory.

(Refer Slide Time: 24:50)



This is the evaluation board, MC68HC11EVB. So, here you can see that this is how we have ports A, D, and E connected to MCU B and C connected to PRU. You have a latch, you have EPROM, or you have RAM, and then you have the RS 232 drivers and the receivers. So, you have the host computer over here, and you have the terminal over here. So, the connector of the input-output connection to the plug-in connection to the system is under test. So, its pin replicates those of an MC68HC11 and is as though the microcontroller was directly connected to the system, and this way, we can use the evaluation board for testing.

(Refer Slide Time: 26:07)



So, if you want to read further on this, you can refer the Mechatronics by Bolton.

Thank you.