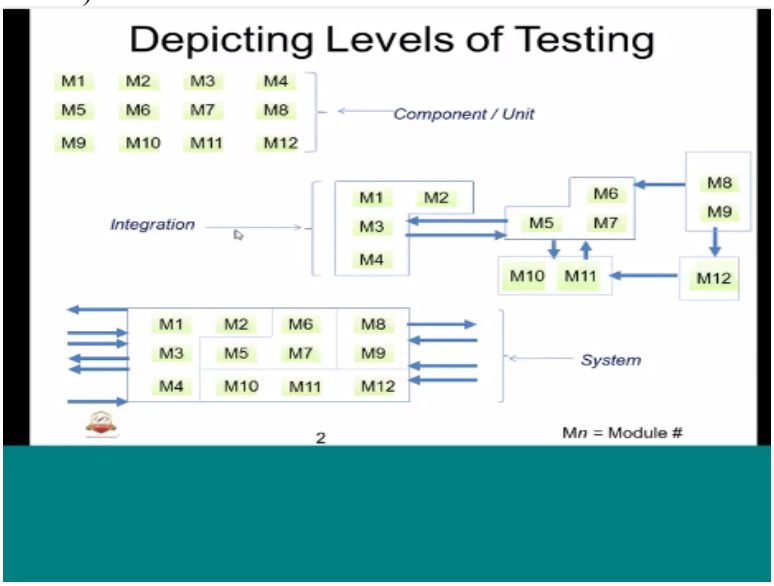
Welcome you to the next session, embedded software testing. In this session we will discuss some of the testing aspects and perspect of configuration and we will, from this session we will have, we will teach some of the test names what are the test that will be called discuss journalism, test journalism once if we test inward other definition. In this session we will try to understand the different types of testing employment, test attempt basically you have to task queue. We will go through in testing test effect called mechanism then the tools and how tools will be used in the embedded system category. Okay,

(Refer Slide Time: 01:15)



I have breathed earlier in the earlier session different levels of testing. Let us meet, deputized into screen level, the unit level testing, integration testing, which consult the, not included acceptance testing because that will not be core instance of the engineering deliverables that is something like user level acceptance testing that could be a subset of any of these three.

Could be a subset of 1, 2, 3 or basically we will use interesting aspects studied and in initial they will take this and from perspect this is perspect the instance of before engineering embedded software testing activities. These are the three important of testing. So, in this diagram you can see the different blocks. The same blocks have been rearranged or different types of testing. The first level of testing we will teach component or unit testing.

That is there are suppose 12 modules in the embedded software M1, M2, M3, M4 till M12. Each ellipse will have to be perspect operation as a component. What are the integrated components has to be or whatever the inputs it take and produces as per standard level we need to circle. Of course there are specific conditions which may have more components like M4 may have M4.1 called aspects. It will come and depict to the test strategy seen how they want to define it.

So, we net shelled the basic modules with their identified and they will be used for component testing or unit testing. Coming to the next integration testing, here what we do? We take this an example of 12 modules for an embedded systems with 12 modules are basically it is based on the, mostly based on the functionality of depict perspect this block1, this block2, block3, block4. So, here is the M12 as single block. May be using why M12 perspect here as well as here.

So, idea became integration be perspect. This M12 which have a binary on hardware or the respectful software, so that used to be integrated by that used to be individually we need to exercise as the module how it works with the other module. Okay, how M12 can be integrated to M8 and M9? How M12 can be integrated to M10 and M11? Discuss about. So, by going this picture the 4 blocks, each blocks is of interest, other blocks will be tested separately or all this integrated together how it is behaving represented on the scope of the integration testing.

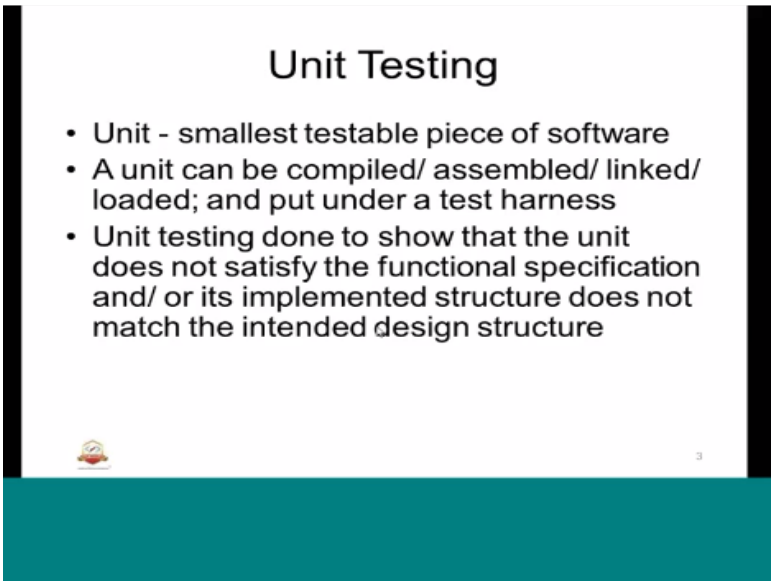
Next I will read the system testing. As I said earlier all the blocks are took together in the box which we consider these blocks as once embedded system target. So, that target will be basically sending to different testing aspects. This thing I will for the IO whatever it is will be fit and the aspect will be expected as the black box. Most of the realistic inputs will be fetched into here and that will be tested again.

So, that is the distance basically with integration here as we give so component level or unit level testing we do individual components or units tested separately. In integration we combine several blocks or component together to test it on the software level software to software or software to hardware and in system testing al blocks will should be offend on the, in the big target and that target will be provided with several inference and the expected modules will be read as an output value Tested against those values.

So, all these three levels of testing we have studied why because we did not do any test by using with the testing we may not be able to test rightly with the integration testing. We may not be able to test with the individuals that are not enough. So, putting all this together embedded system testing will be, is consider as the complaint. Please do not get confused with all the bottom of top of down that evolved cockroach is in the method which is done. Here basically given the embedded system how we are going to testing.

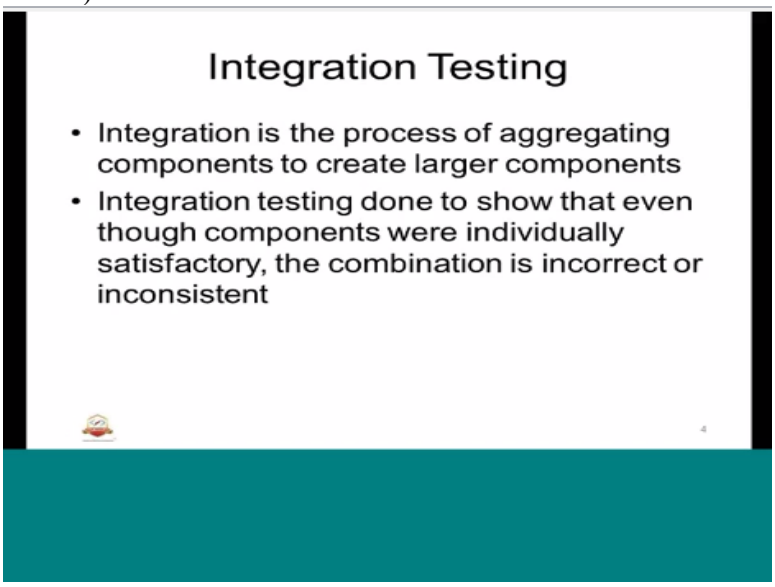
How we are going to break I mean this I took as knowledge. So that is the philosophy relends and the relevance of that we are doing depicting and M4.1, M4.2, M4.3 you can test it as an even for all this. So, like that we cross the execution level. Here in this case, picture personality is given. Or this must said 1, integrated 1 will have a feature that is 1 and this is 2, this will have a data function take read for some other leave. That is what we have. In the system testing it could be black box and the target are reveal under I will allot the depict testing are put together and afford on the, this could be the arrow mark and the inward arrow mark will be the called in the test is a black box. The outward is called outfit. That is how to take it out.

(Refer Slide Time: 08:52)



Okay, just to highlight what we do? Unit testing unit is the smallest testable piece of software. A unit can be compiled, assembled, linked, loaded and put under a test harness. I will explain what test harness is in a later session. A unit under testing will be separately addressed for unit testing. Unit testing done to show that the unit does not satisfy the functional specification or its implemented structure does not match the intended design structure. That is testing is done to show that is why the function is specification or predicts that we need. So, it is not satisfying that or respected or implemented it is not matching the implemented structure that is the bone of testing going under test.

(Refer Slide Time: 09:56)

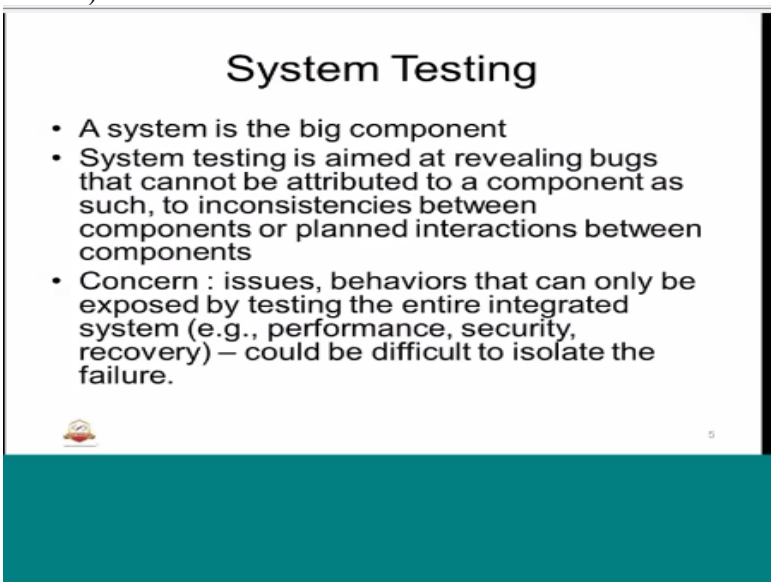


Next we are going to integration testing. Integration is the process of aggregating components to create larger components that as well so the components are aggregated to create larger components. So, the components could be M1 to M4, M5 and M7 and it may be M9 etc could be in component as the integration testing what we do integration is the process of aggregating components to create larger components which we get larger components as I said small

components are together blocks of1, 2,3 4 whatever it is M1, m2, m3, M4 these blocks are combined together and we will test against their interactions between the modules M1, M2, M3, M4 first then as I grouped for as block how it is interacted with the other blocks M5, M6, m7 or neither blocks of components.

So, that is the intention Integration testing done to show that even though components were individually satisfactory the combination is incorrect or inconsistent. That we as I said you would have tested individually in M3, M7 whatever it is but when you integrate M7 to M12 so whatever guarantee that it is working completely or correctly so in order to do that we will integrate. So, that is the purpose of integration testing. The combination should be correct and consistent that is the code.

(Refer Slide Time|: 11:52)



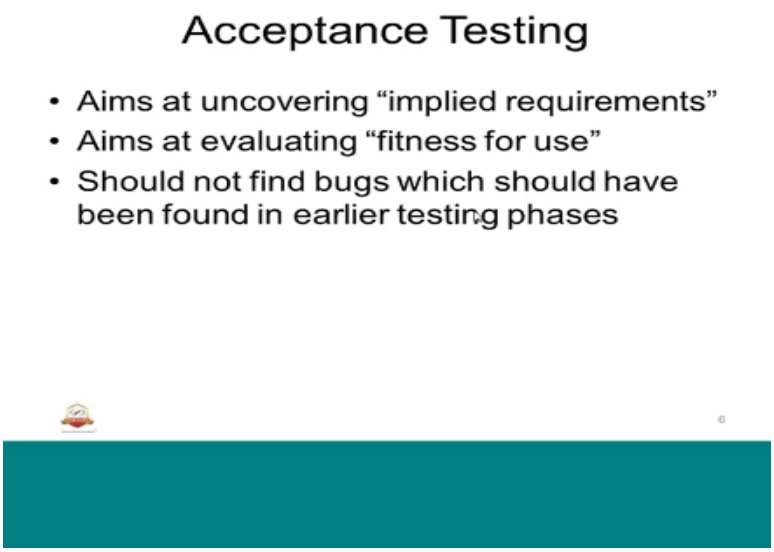
Last one is the system testing. A system is the big component as I said it is a black box. System testing is aimed at revealing bugs that cannot be attributed to a component as such to inconsistencies between components or planned interactions between components. Still you see there could be some issues after putting together all this blocks that means while first block is interacting with the second block and while the third block or the fourth block is interacting with the fifth block continuously within the system there could be some issues.

Especially when you trigger the coming post of the signal all together our queue should correctly and consistently, so, how do you bring up those issues? That is to the system testing. So, system testing is aimed at revealing bugs that cannot be attributed to a component as such to inconsistencies between components or planned interactions between components.

Concerns: issues behaviors that can only be exposed by testing the entire integrated system. So that is what the purpose of it. Specially the performance, security, timing, recovery these are some of the aspects will have issues when you test it to higher level. So, the problem still is that sometimes it is difficult to isolate suppose isolate the performance is not up to what you expected. In a system look under the system testing and isolate there is a memory leak. I say there is timing slack. I do not know where it is, the problems. So, how will you isolate? So that is one of the concerns that we have a system testing. Of course definitely this individual level or

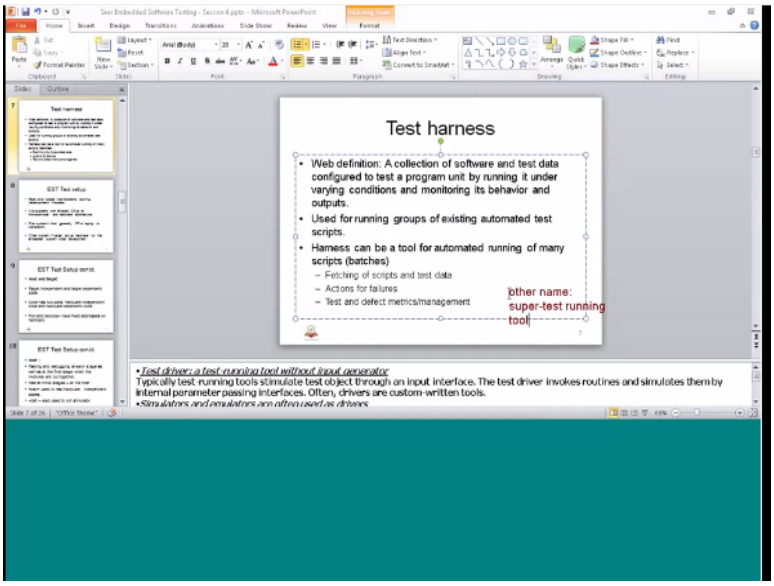
unit level or integration level, it may not be able to find out. So, how do we do it? So there are several mechanism tided through debugging where we do the white box set of mechanism be connect the debugger and a manual analysis. Who is giving IDE, code inspection or walkthrough etc. some of the issues are you may use IDE tools such as profilers or time machines etc. these are some of the alternates while dong the system testing because some of the critical things may not be found out easily we does system testing. So, there are alternates that we will use it. So, that is about system testing.

(Refer Slide Time: 15:25)



Of course I said acceptance testing uncovering implied requirements aims at evaluating fitness for use. There is selective level that means some of this system testing test can be reused to make sure that the product is accepted and definitely that should not find bugs which should have been found in earlier testing phases that means integration and unit testing or system testing these bugs should not be whatever the bugs are there these bug should be fixed. So, acceptance system is the last we do before we leave the product.

(Refer Slide Time: 16:12)



Okay coming to test harness, the embedded system test harness what it means the web definition about I like this. A collection of software and test data configured to test a program by running it under varying conditions and monitoring its behavior and outputs that means test harness will enable the test harness to feed the test data and test deprogram which is under subject by applying various conditions and the target should drive into support monitoring the output as well.

So, it is something like a test setup part of it. It is used for running groups of existing automated test scripts that means once we have individual tests done we should be able to run in speedier way that means we can group the existing tests in a batch sort of a mechanism where the environment also stops each apart. So, the harness itself can be a tool for automated running of many scripts like we as I said we can create batches or scripts and that batch can be run of the harness in a consistent and continuous manner.

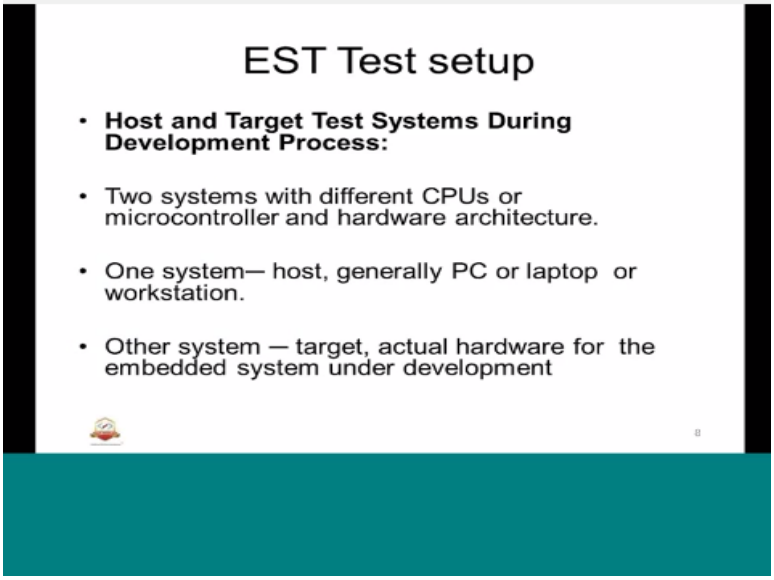
So, that the testing will be speedier and lot of automation is involved in. fetching of scripts and test data. What it does? The automated the link of batches is fetching of scripts and test data. Any actions of failures should be photograph or video or report log etc. the test ad defect metrics or management, how many number of passes are there? How much it as failed? Like in terms of overall automated batch output this will take care.

So, that is the test harness. Test harness could involve many things such as, working a simulator or test drivers could be involved as well. So, there is a difference from teat turning tool and test harness. So, test turning tool is something like an underneath thing of activity. Test harness is above that which will help in terms of automating and turning at a higher level. See typically a test harness is a super-test running tool.

The other name uses is super-test running tool. So, controlling collapse have you tested running tools I mean some other sub tools it can control or trigger or drive it? So, test harness can be used to run groups of existing automated test groups, a typical activities or fetching the, as I said scripts on test data should take actions for failures and should be able to manage the tests before reporting insolently and exponent reports defects as well criteria outputs and all that will be part

of this. So, that is about test harness because this is one of the important tele-factor that will be used across all this. Test harness is very much needed for infinite level testing, integration testing, system testing. Test harness has some sort of setups for each of these. Okay, coming to the next slide

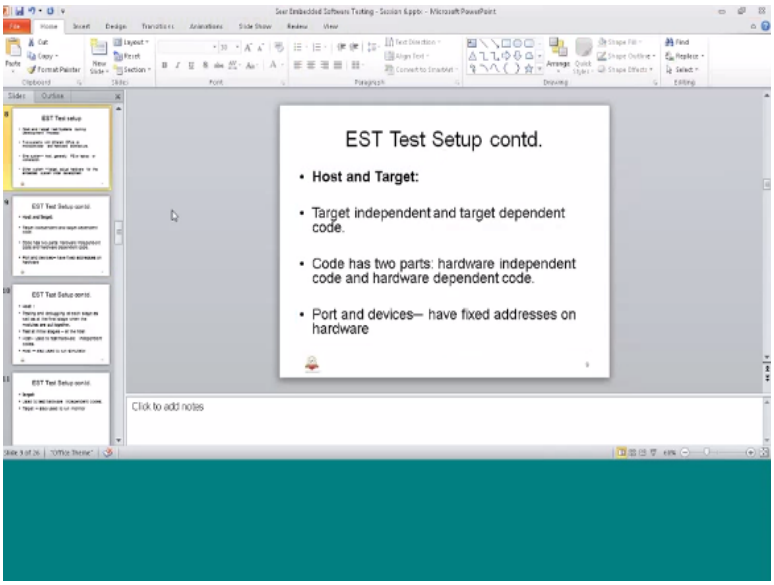
(Refer Slide Time: 21:09)



This is about the setup. We know that test harness is required underneath that what are the things that we need to take care. Embedded software testing setup, there are several ways the test setup can be done with high and defense on what level the system is required? I will explain a typical how it is used. Basically host and target test systems during development process that is one thing I will use it at least during development again, the developers have to have a components on how to develop.

So, they will have a set of host and target interactions. So, same thing can be used for testing also. So, we know that host and target earlier we have discussed two systems will be there, hosts will have CPUs will be there and target will be having it is on architecture or microcontroller target system. So, one system host generally PC or laptop or workstation will be used. Other system will have a target for the actual hardware. Actual hardware can be a microcontroller based or micro-controller bus any other hardware. Okay, JPEG or any flash device anything which is used under the development so continuation of that host and target.

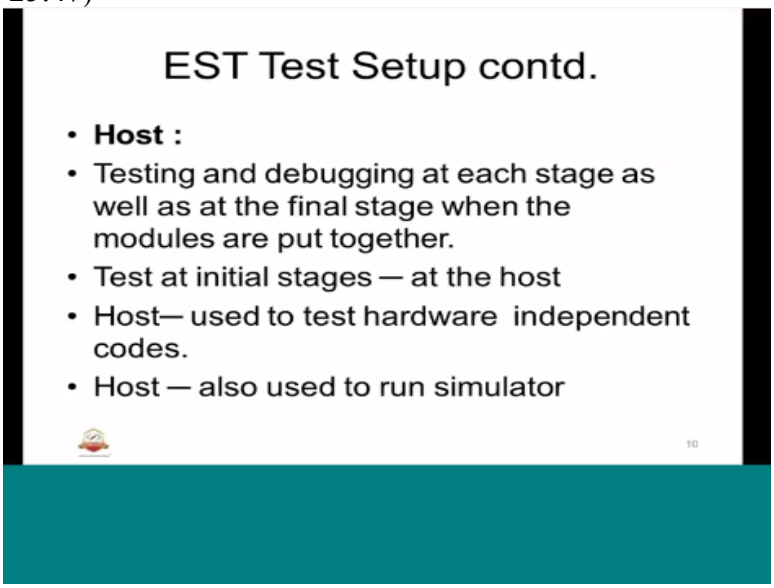
(Refer Slide Time: 22:50)



Target independent and target dependent code. So, what it means like there are few code like hardware related. Suppose take microcontroller related or registers which we may use for example target specific hardware registers, microcontroller registers, memory etc. those are all called target dependent. Target independent is something like we have algorithms, few utilities like conversion utilities, any glow, structure, data structure mechanism etc. these are all target independent.

Even target dependency under target for this piece of software that is how they categorize. So, centering is explained here code has two parts hardware independent and hardware dependent code and port is a timber a piece of interprecies which is presiding in the target and devices by tele-Fred could be flash device or any crystal or that will have addresses, the microcontroller using the addresses. So, that addresses are fixed so we cannot use it has general. So, this is one of the set up that is used that means host and target based systems. Same thing will be used for development as well as testing so, some details about host.

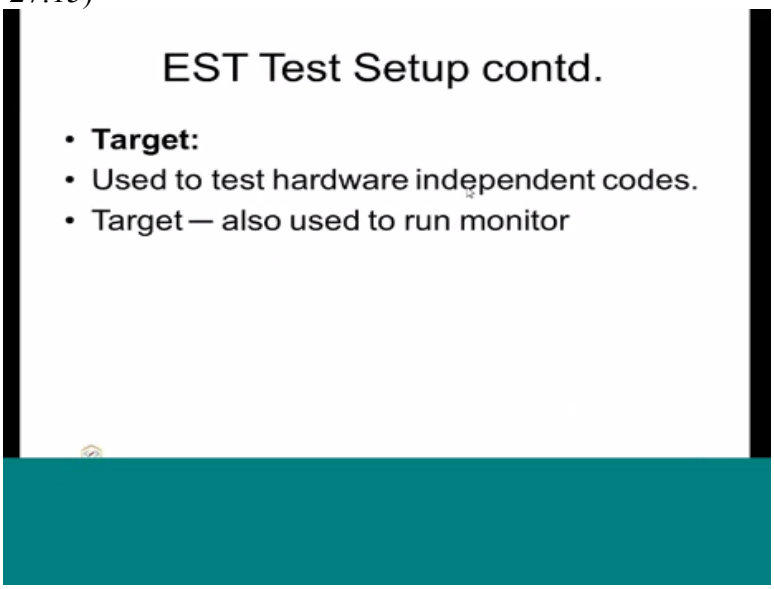
(Refer Slide Time: 25:47)



Testing and debugging at each stage as well as at the final stage when the modules are put together. That means which will be fixed, the different stages of developing and integrating on to the target with the help of host will be done at the same manner in different stages. So, testing can be done at each initial stages at the host side, that means having developed some piece of software we can use the same host for testing it like whether it is working at the expected of and in the host itself we can use like we do not need to program that onto the target as I said there are independent software required that can be used in the host itself where in those module like algorithm and hosting can be run and tested on the host for the PC.

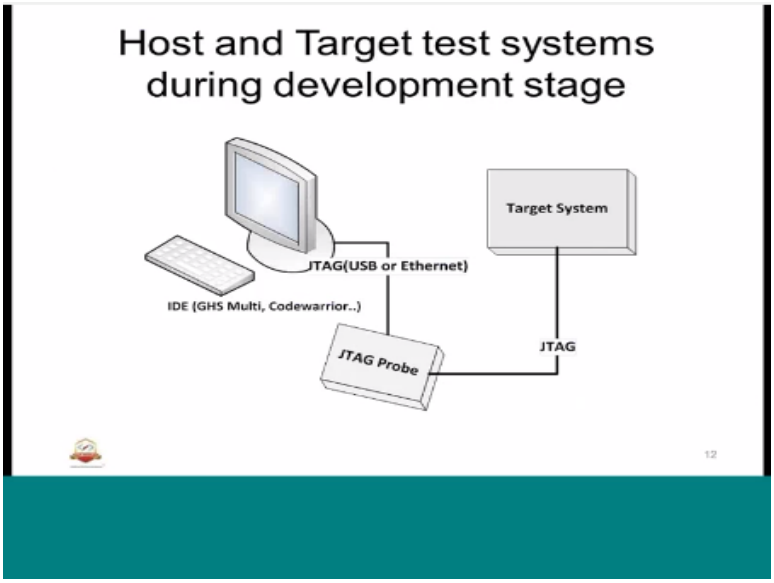
Also there are integrated development environment target based systems which will allow for user to simulate that means that simulator is an obligation which will be running on the host which can simulate the target between intended to be used. We will touch case each of these aspects in the next class.

(Refer Slide Time: 27:15)



So, this is the definition of the target. So, this is used to test hardware and software independent codes. So I think this is should be hardware dependent and the target is also used to run monitor that means from the target anything that we expect to see the output of the tests. So, we will need some monitoring mechanism. For that monitoring mechanism can be output to a target. That is how the target is used

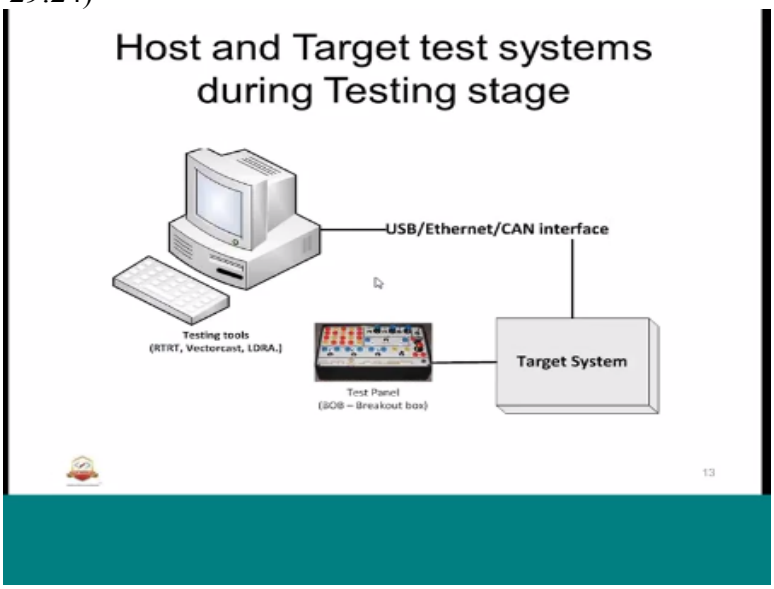
(Refer Slide Time: 27:52)



Here is a typical diagram of host and target how it is connected. So, typically it is used for development. Some of this can definitely used by the testing also like unit testing you may have to develop some scripts which will use the same interface for testing aspects arrogate system and the scripts will have dependent on two of them are the test development environment which will be used for debugging the target system and I brief about this host and target system and development systems is a laptop or a desktop which will have an interface with the target system with the help of an IDE.

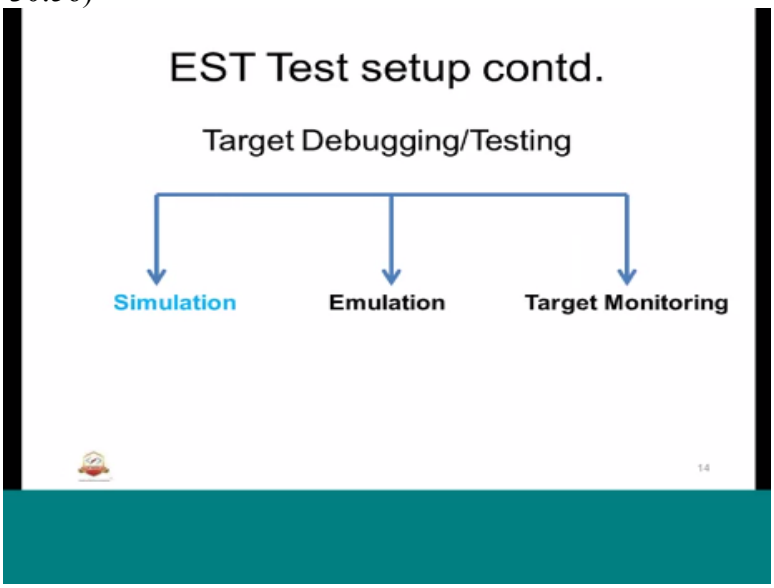
IDE certainly uses probes which will probe the registers, the memory or the hardware related information through a core box that is called J-tag and there are tools which will recruit this and will allow the user to monitor or put values and all that so, for example IDE integrated development environment we have use JHS to the code warrior etc. they interface with the J-tag will be usually through ISD or either net.

(Refer Slide Time: 29:24)



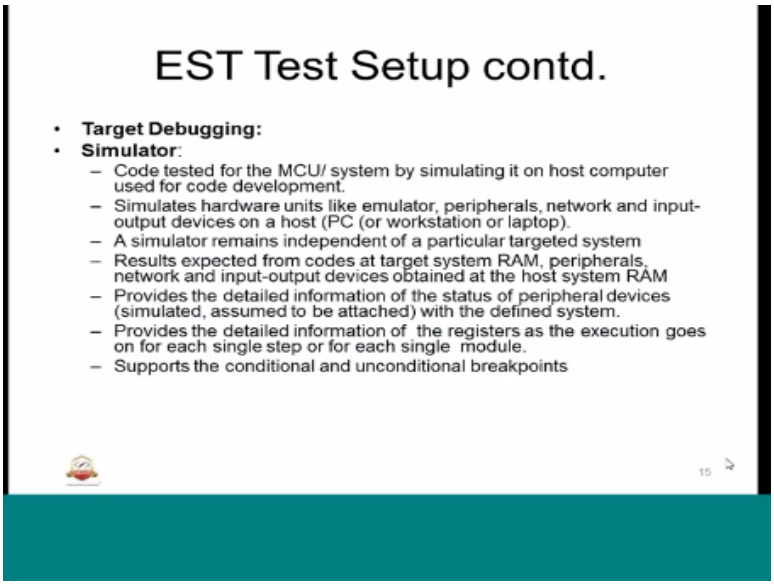
Now as I said in earlier slide how the target system will be used for testing? So, once we have development we no more require any of the IDE or J-tag which is connected to the target system. In leave of that what we will use is the target system actually connected with the real inputs that is the realistic values with the help of test panel or grey code box that can be used to provide the inputs and to test the values or the monitor the values we use interfaces jet scam, internet or it could be oristherat also. I have seen separate system connected to target system I mean separate interface as well some of the monitoring can be done with the help of that. For example RS2 22 and that all can be segregated in the host with the help of something like testing tools. Those testing tools will help user identifying the failures or run scripts and batches etc. okay, so we know that there is a target system how can be setup with the help of this. There are different mechanisms that can be used for doing the test setup.

(Refer Slide Time: 30:56)



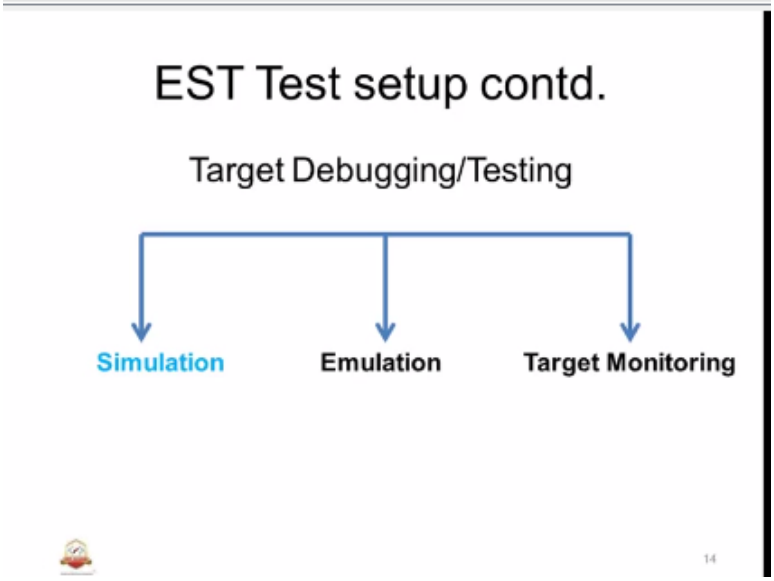
Those are broadly three categories having said that we use different setups for the development and testing. The setup is broadly categorized in three ways testing in which simulation, testing with emulation and with the help of target monitoring that means the target debugging testing is categorized simulation, emulation and target monitoring. What are those? So we will see each one.

(Refer Slide Time: 31:50)



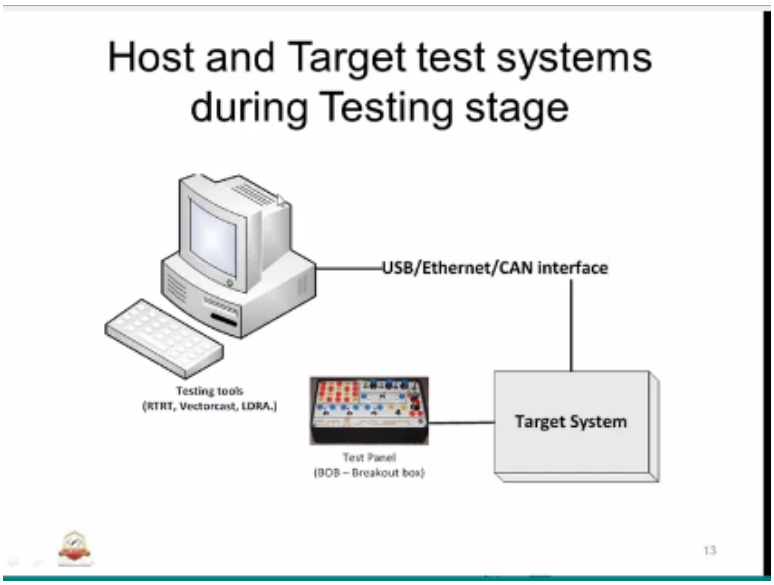
So, target, the embedded system target can be tested with the help of simulator so in simulation what it has basically, it uses the knowledge of target processor or make it controller or target system architecture at the host side that means on the host processor.

(Refer Slide Time: 32:24)



That means we have seen the host is the desktop of the user station or the laptop.

(Refer Slide Time: 32:27)



So all the information required from the target side will be available in a simulation manner we do not need the target should be running but the host has a knowledge, so for that what we have to do is first the dialoged code or that code that we have should be complied cross complied I think I have explained the what is what is cross compilation and embedded? Anyway that is the part of the embedded system basics we use cross complier where we have the code developed in the host and complied for the target.

That is cross complier. So what we do is we do the cross complier and code and place it in the host system RAM. That means host system will have the complied code. Then behavior of the target system host registers is all simulation with this simulator. It use the linker and locate to cross complied course in RAM, and functions like that code that would have run, at the actual target system.

That means it has this the target is running it will be running in the host with the knowledge of whatever the target it is having in the host so simulates code tested off for the MCU system by simulating it on host computer used for code development simulates the hardware units like emulator, peripherals, network and input –output devices on the host or the workstation or laptop it is all the peripherals that it has which tries to use it.

Try to emulate it basically with the help of what are the emulation characteristics that has. The simulator remains independent of a particular target system as I said it does not have a dependency on the target. That is all expected from the code set and the target system RAM, peripherals, network or input-output devices or obtained of the host system RAM. I would say, may not be 100% possible with all the devices and peripherals but at least generic nature of all this memory parallel, and all that should be an able to simulate so that is how they develop the simulate. To provide the detailed information of the peripherals devices like it is running or stop etc.

Provide the information of the execution goes on for each single step for each single module. I think one session I have in terms of d3ebugging testing with the help of ID integrated development and it will be probably mark clear with that and similar manner simulation also it is

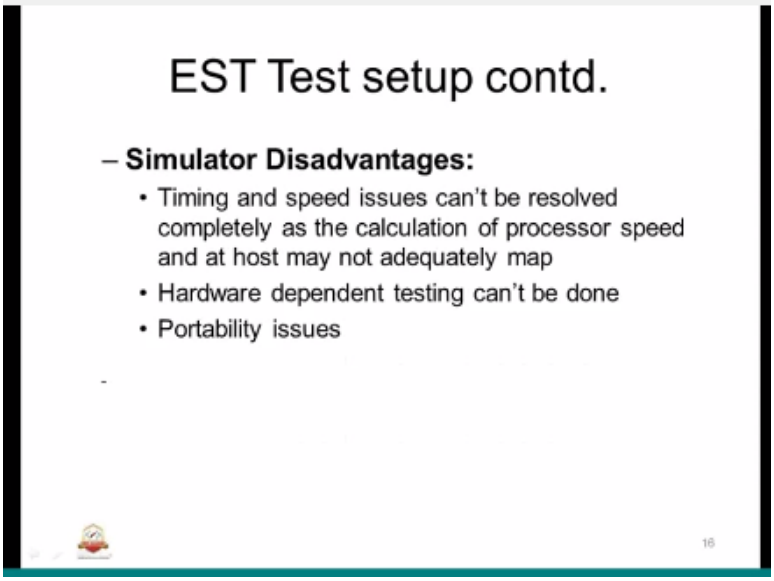
not too much away from what we usually use in the debugging and normal. Instead of the target system the host will have a simulation based system it is also supports all the break points like conditional and unconditional break points.

These are all some of the simulation break point of the software break points it is used for simulate. So the advantage is it is helps in the development of from the final target system of with the help of only PC at the tool for development that means we have need to the actual target system at least we can progress some development operation before we actually get the target. In general in an industry what will happen I think I will explain this in different models strategy host.

The hardware and software planning and requirement and development course parallel. Hardware is bit hide of software, hardware is should be ready but what will happen is it is likely that before the hardware the actual hardware is ready the software piece could have been developed. So get the confidence of what is be develop? We use mechanisms this such as simulator etc.

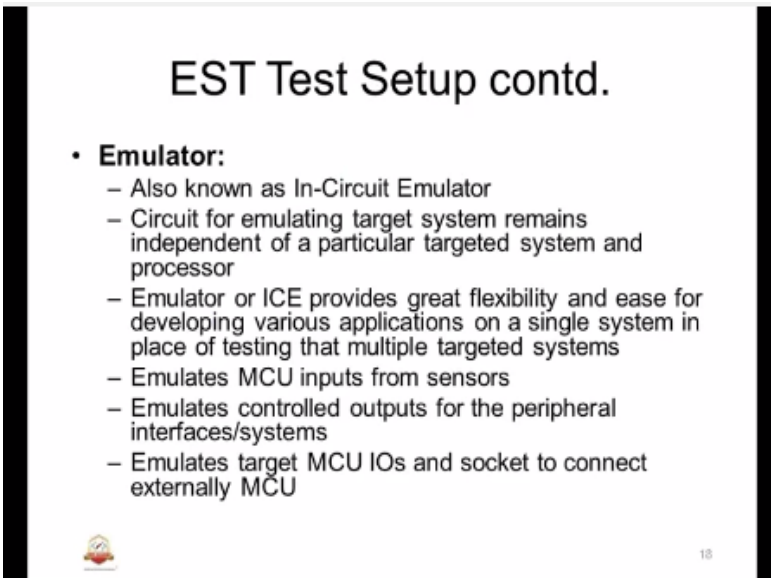
So that will of very much use and it helps of the initial development step. And simulator for available fir different hoisting devices, which are targeted for embedded systems so most of the processor of the development. Definitely there is some location available most of them like let we take micro complier such as TI based or motor based or whatever it is they will have simulators we can use for simulation. The same simulator can we used for especially the same code we can do the target and that code can be tested. Especially for any testing this will be useful. Where we use individual modules can be simulator impossible like algorithms or software flows etc., can be definitely tested with simulator.

(Refer Slide Time: 38:25)



So but simulator will have some disadvantage those are timing and speed misuses cannot be resolved as I said the actual timing of the micro controller or the processor. We the actual load of the signals or the actual peripherals devices connected so definitely vary that simulation. So those issues we may not resolved completely because the calculation of the process can be host we may not be adequately map we may map one piece or one processor or peripherals all cannot

be sink easily with the help of simulator. And of course as I said hardware dependent test like we have some of the border sate or the interfaces connected for the spy interfaces we may not be have to test it.
And of course we have developed the code in the host environment and cross complied and done it simulation and that should be able to port it as it is on the target without difficulty but since there are some issues especially on the binding so or accessing memory or issues. Definitely there are few things which have to simulator that is about simulator and we go to emulation next so that is the next technique which will actually use the target. Something remembers in the simulator for example target system may have 8 bit data bus between RAM and un-pipeline or every map processor.
Host may be having a pipeline processor. And it could be so this is the difference with the targe5t system and host. Similarly simulator as it raises from an interrupt some particular situation only. So like you never aware of interrupt some interrupts can come with the real time but simulator we may be not to do it. So this is some of the disadvantage which I think that we cannot use in simulator, we should definitely use the next techniques. This is emulation
(Refer Slide Time: 41; 18)

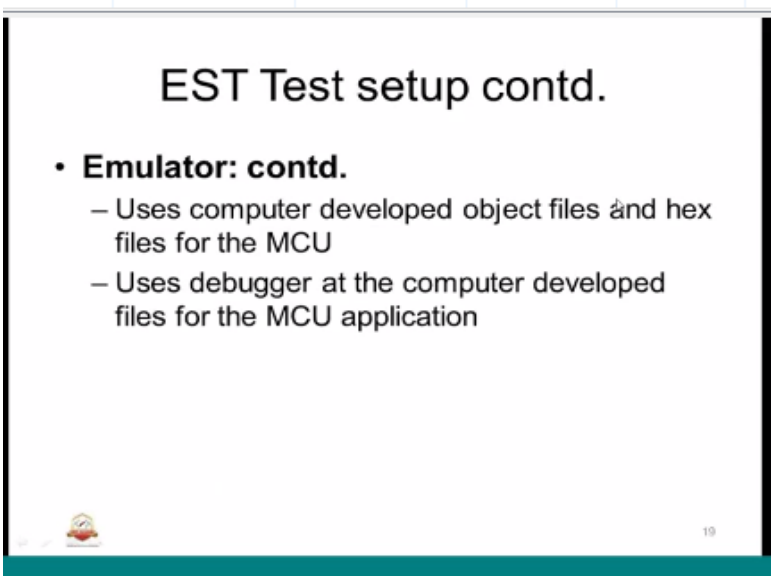


So what is an emulator? This is also called as an in-circuit emulator so basically which actually takes the in-circuit values and we will provide the user what are the values that is going on with the target so which actually emulate so what is needed similarly, user can put the value in to that or just a with the help of emulators. So emulator also known as in-circuit system ICE circuit for emulating target just come to independent of the target type system and processor. Simulate the ICE provide great flexibility and ease of developing various applications on a single system in place of testing that multiple target system.
That means we have batched of dependent target system that many if the components are same and can used an emulator on one system and probably it will ease out for the developing the

different applications of target systems. Suppose some sense of circuit is connected that can be emulated.

Emulates control output of the peripherals interfaces that means actually use peripherals which can be trigger or which can be controlled with the help of simulator. Emulates the target MCU IOs and socket to connect externally MCU suppose one more MCU or other MCU use so we can have multiple emulator or single emulator connect to the host MCU we can be look for the socket other MCU because there are complex system where how micro control. So both have to be used and both will have some interface some inputs and output and how do we do it. Definitely this is not possible with the help of simulator we have to use the emulator techniques.

(Refer Slide Time: 43:31)

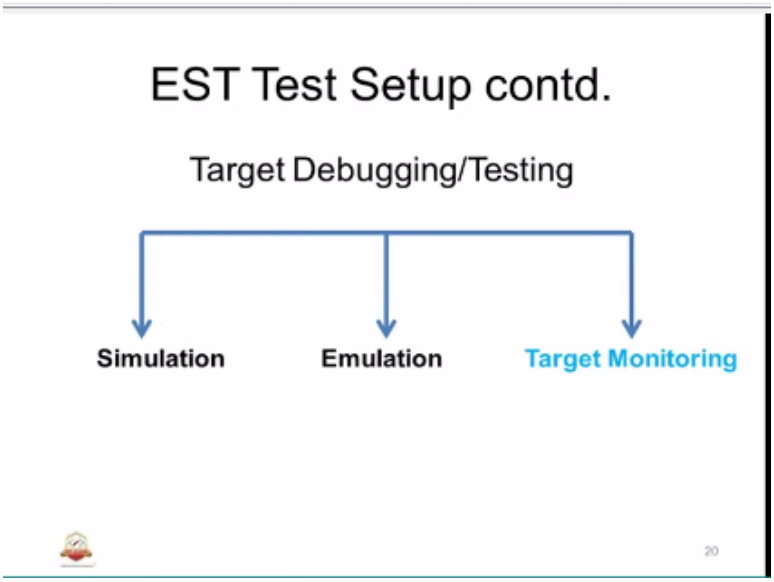


So what emulator target the developed we develop we compile of the target and it produced the object files those object files and the binary or the image the micro controllers which is required will be used by the emulator it is used debugger at the computer development files of the MCU application that means the debugger will have a knowledge of the entire MCU and MCU corresponding the applications can be debugged with the help of the emulator.

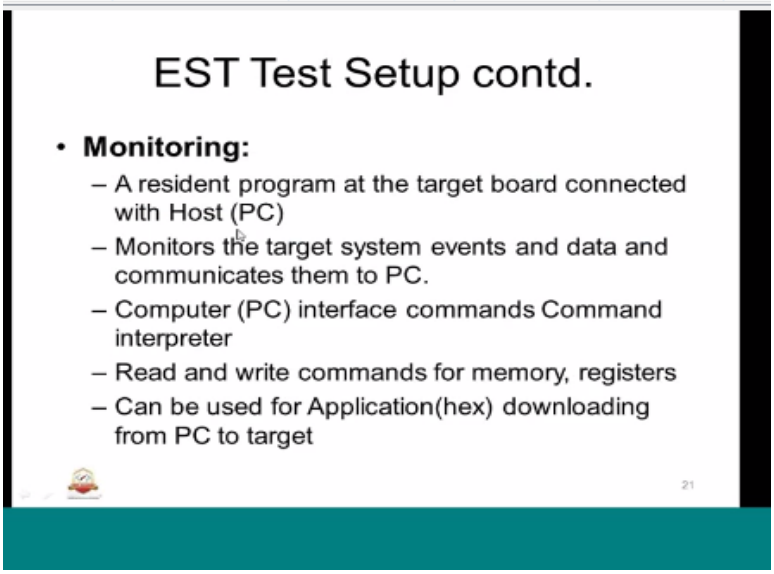
This emulator why it come for testing you may ask because emulator is definitely use for developing it can also the used for testing also it must some of the test may not be possible with the help of automated we have an irregular bugs to the with the target not having connected with the development environment. Why because some of the incase of some of the target system need to be inspected with the help of code

So use the test I has to go through the test code or the code on the target so how do you with the help of emulation. So emulation that is why t is used for testing also that is why we taken up has been one of the technique in terms of target debugging testing.

(Refer Slide Time: 45:15)



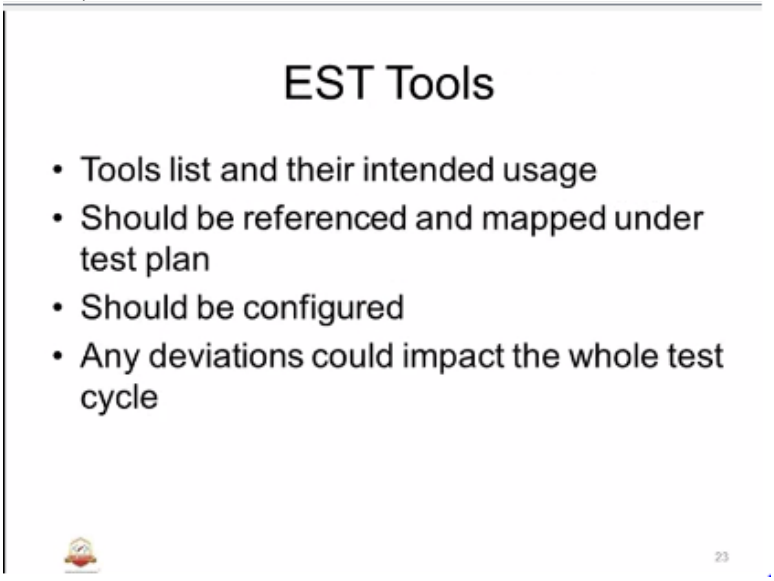
The next one is target monitoring
(Refer Slide Time: 45:29)



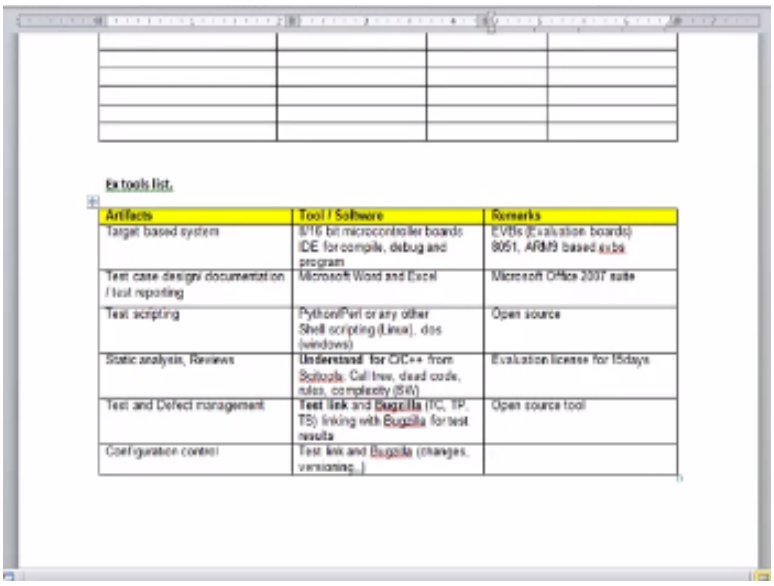
So what is monitoring? Monitoring as an output activity from the target system used for embedded system testing so what are the elements used to be monitoring is the resident program that is based system which is connected with the hoist as I said the connection could be RS232 or it could be any of the interface system. The interface could be generally used or can is used in many places control area.

So for this thing I mean we definitely need some program which is residing in the hardware system because the test is interface with the RS232, and it has top pumping some data to the host and the host has to work knowledge and host can be pressed that the system also. For that it has to respond so there is the hand shake involved for the between the target system and the host for that dedicated monitoring application might needed the both side as well as on the target. Monitoring as the target system is some data and commands in this interface and comes in the data to the target.

The computer interface commands command the interrupted that means for retrieving the data or it may have to send some commands and respond part of the data. So it uses some commands it is the protocol of the hand shake involved the target board and the host. Generally the commands are mostly they are used for memory registers operations.
And also there are systems which uses in the same interface for the target application that is the image or the binary to the target to be download with the help of PC. So this are some of the target based debugging or testing with the help of test set up. Test set ups we have seen how it will be.
(Refer Slide Time: 48:43)



Next one is the tools. So we will conclude this session, the tools with tolls this example and in the end we will touch base up to embedded system what we saw in the it is something like r3ecap of what we have studied so far or what we have study so this is all some of the words that has to be the code of the tester. It test should be knowledgeable enough any time to talk on this. Okay, so embedded system testing has to have a list of tools and it should be a well-documented tools list with additional information of the tools will say the guidelines etc.
And this all the tolls should be referred or mapped in the test plan. We have seen how the test plan looks like. That should have a reference the all tools or tools document. And that should be controlled with the help of comprehension management. And any change over the period of testing or development or the life system defiantly well has an impact. Why it has an impact? Because whatever we intend to use whatever we want to devolve slips and one version of tools may not work on the next version.
So that is fir example so similarly re check the part of the usage of the tool any changes happen that has to be controlled. So any deviations that have to be taken care with the help of deviation process along with the impact. So definitely there is an impact of the whole test cycle, wherever the impact that means to be executed. So, embedded testing software testing tools will for example how it is tested?
(Refer Slide Time: 51:11)



Ex tools list.

Artifacts	Tool / Software	Remarks
Target based system	8/16 bit microcontroller boards IDE for compile, debug and program	EVBs (Evaluation boards) 8051, ARM9 based ex:tbs
Test case design/ documentation / test reporting	Microsoft Word and Excel	Microsoft Office 2007 suite
Test scripting	Python/Perl or any other Shell scripting (Linux), .dos (windows)	Open source
Static analysis, Reviews	**Understand** for C/C++ from Scitools. Call tree, dead code, rules, complexity (SW)	Evaluation license for 15days
Test and Defect management	**Test link** and **Bugzilla** (TC, TP, TS) linking with Bugzilla for test results	Open source tool
Configuration control	Test link and Bugzilla (changes, versioning..)	

So you can use I have what I did in earlier stages like test plan and procedures cases I am not put the template probably as one of the have all this tools this is the example I have put here may be that is an explain we can take it off during the critical session to list out all the tools and as part of the templates. So templates look like something a tolling permission the tools name, when and how to use, and of course we can have tool version part of the configuration information also gathered.

So this is the templates that they generally use but tools they have for this course planning to use some of this tools listed below. Different of the availability and other example I have listed below typically target based system as I said, we will have a 8 bit, 16, bit, 32 bit micro complier along with that you should have an ID target environment for compiling debugging and testing the development software. For that we use evolution boards we can target use bits ARM 6 or 7 or 9 this are available.

We can use for testing once we have keep the system we need to have the requirements other input that is the part of the space we will develop the test case can we will do the test case design documentation. This can be with the help of documentation Microsoft word or excel we can must was to lay out that the test procedure we need to have scripting mechanism. Scripting mechanism, can be done with the help of python I am not sure how many of you all aware of this.

This is near to any other scripting tool like you can take for example, scripting lines or dots and windows that also we will have command and arguments and all that so this can be used for scripting. And this is the open source also download and use it and for testing we to use static analysis review and all that for that one recommend to some of sky tools this also the revolution give it actually this is the good tools why because it use for complete code picture something like a call tree and all that producers. It also gives a configuration output how the code is organized in terms of code you can apply the rules the coding rules, and we can see many of the other aspects like complexity.

So this is called software complexity, I will like one of the session I will explain what is software complexity, this is basically, how complex that are get system about. It takes the numbers basically for any embedded system has to be some limit of the complexity beyond that it cannot be developed. So we use that static analysis but test defect management we use the tools call test link testing we use test like the mangers all the test cases, procedures, test shorts also and also the script also we have input with the help of testing tools it does not test but it helps us to manage this is called test links.

Similarly, once we have tested with the help of this tested tools use and all that. We can link them the pass fail results or reports, like failures can be controlled I mean can be managed with the help of tools is called buzilla, so one of the open source go to test link also open source tool. And we can use for same test link buzilla for configuration that means we will know changes will happen the version should be there all these can be taken care with the help of test link as an buzilla. It is as the favor of configuration. So this is an example of test tools. So that is about tools.

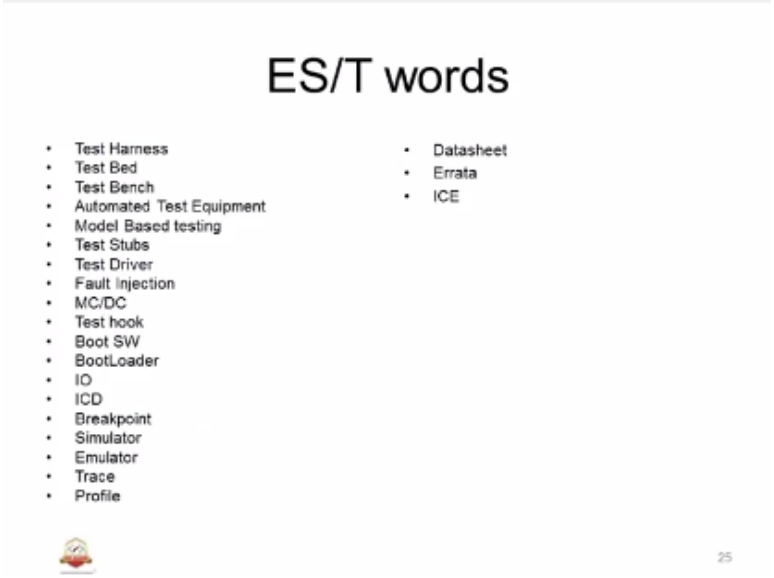
Okay, so this session we have go through different level of testing component or any integration system testing, what is integration testing, what is system testing, test or it means set of typical embedded software testing set of how it will be? Host target based system how it will be develop or hoe it will be used? Is a development as a set of this is the real environment testing set up and how we can test having set all this environment is developed with the help of techniques simulation test target based monitoring.

So emulation sorry simulation is used the target based knowledge available in the host and all the interfaces and the processor or the host. May be known that always simulate have been used that is the disadvantages in terms of hardware. In these cases we use emulation which will actually use the in-circuit simulation technique, in terms of all the hardware the values in to the host. With the help of the host we will write and all that while it is in to the target.

In the same target can also be used for programming the multiple object files or different programs. The next techniques is target monitoring where there is the already built program running on the target, you should have interface with the host. And there are commands for with the help of which the target is interacted.

And those set up has to be configure and controlled with the help of software environment configuration it is called environment configuration and index, which will list out all the tools and what is the environment us completely this should be mapped with the test plan. And embedded software tools, we have seen the example, what are the tools and what are the information should be used for listing out and how to use it and any changes for the period of testing has to be taken in the deviation.

(Refer Slide Time: 1:00:33)



Okay, so we will we have listed out some of the testing words probably, the next excise will be to put at least single word definition for this. So my next exercise are question is something like okay, so that means put a one sentence answer for some of the words which we have gone through this course.

Test hardware, what is the test hardware? It is the one line definition or put a small paragraph test bed and test bench both are same it gives the different contest. In some they use test bed and some industry test bench and the actual environment that it used will have something name something like automatic testing and this w have not done what is we will touch base extras I will list out all the words. So what are the words? That you have gone through, you can put one sentence answer.

I think test drivers we not have seen, fault injection nothing but error injection or fault triggering so we have MCDC let we cover and the testing in the next stage test hook we know that test hooks used for testing the targets system there the doubts will accept the user inputs provides user and monitory use of aware the input of program with the help in new application IO is a input ICD is the interface control document, basically the embedded that is used for document to defined the interface this is separate requirements sort of a document sort if a thing this will define all the inputs and outputs for the system as well as the requirements.

We know what is break point, we know what is simulator, what is emulator breaks this is part of emulation or emulator you can define or describes profile link and memory and all we can and for target processor used for data sheet data sheet is used for the reference manual for ARM micro controller such as ARM etc.

So, this is one of the need that embedded system tester as per have why because it is needs have the knowledge of the micro controller which is under test. And every micro controller will have a data it is an error news information which will explain about any of the bugs or errors still it is available in the micro controller which we should be aware of you know what is the ICE in-circuit emulator the question is please provide an one line answer of some of the words that we

are used in this course. So that is all about today's course and next time we will talk in next session.