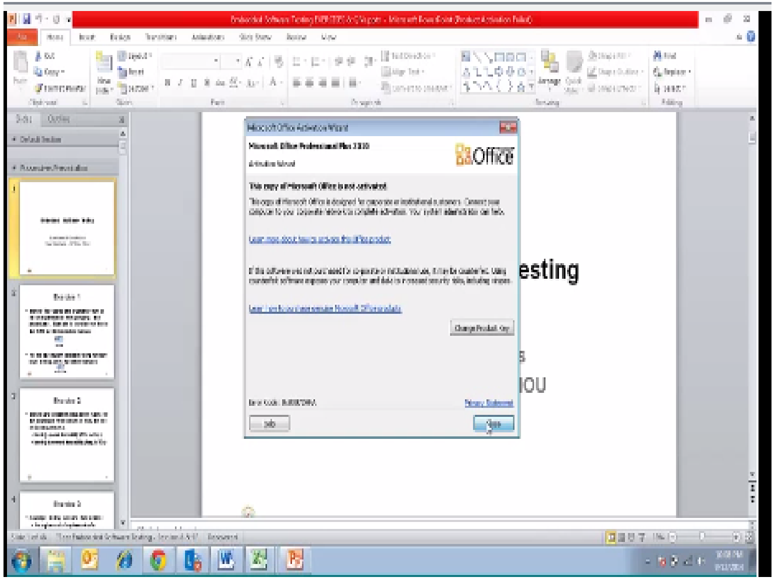
(Refer Slide Time: 00:02)



Microsoft Office Activation Wizard

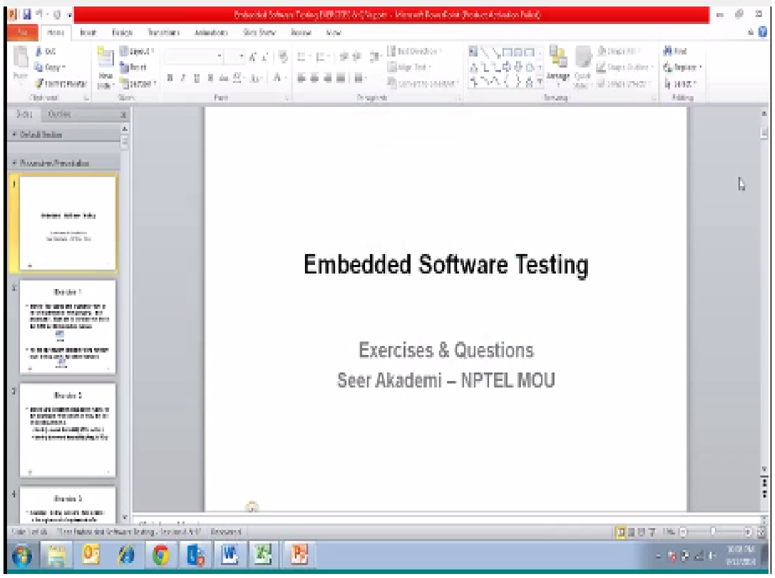
Microsoft Office Professional Plus 2010

Activation Wizard

This copy of Microsoft Office is not activated.

[illegible]

(Refer Slide Time: 00:04)



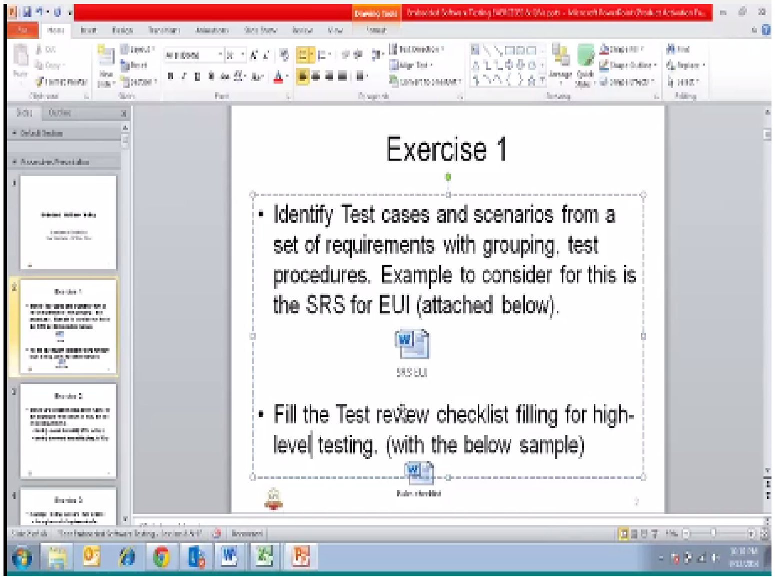
Embedded Software Testing

Exercises & Questions

Seer Akademi – NPTEL MOU

Welcome to the next practical session of embedded software testing today.

(Refer Slide Time: 00:14)



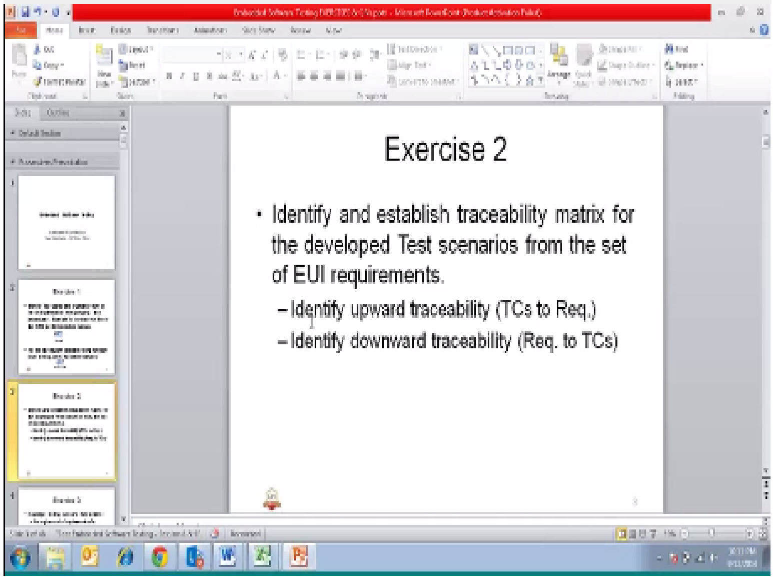
Exercise 1

- Identify Test cases and scenarios from a set of requirements with grouping, test procedures. Example to consider for this is the SRS for EUI (attached below).
- Fill the Test review checklist filling for high-level testing. (with the below sample)

Will see the exercises regarding test case resource and scenarios from a set of requirements we had a studied requirements for a lst class in try to study that with a requirements and the corresponding this cases so we had also gone through the example of SRS called a built in with a estimate so we just try to recap that we have a just we have a check list this is base form high level testing this is the sample here.

This typically being done or the completion step to take a top and software testing high level testing that is been done re set of requirement so this check list will talk about the rule that what we fallowed and whether there are emirates of the software testing can we taken for a next level like this for a next level for this to be used that is how the rules have been used.

(Refer Slide Time: 01:29)



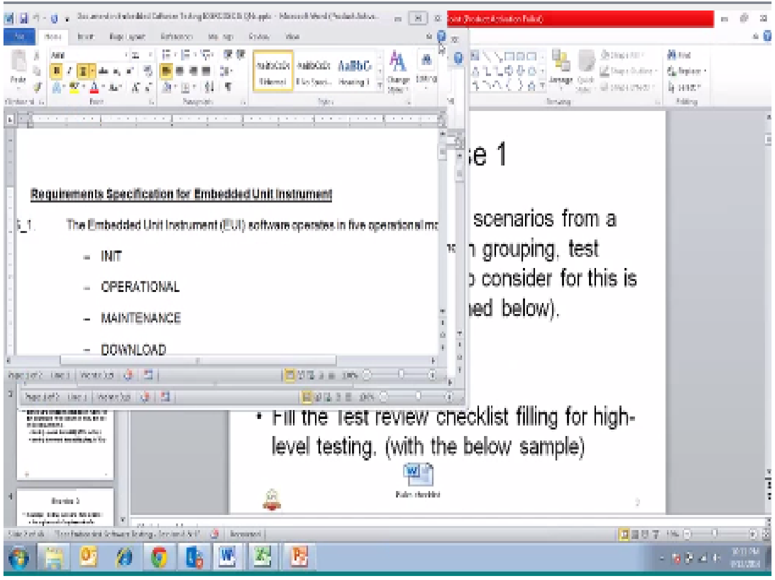
Exercise 2

- Identify and establish traceability matrix for the developed Test scenarios from the set of EUI requirements.
  - Identify upward traceability (TCs to Req.)
  - Identify downward traceability (Req. to TCs)

So once this exercise and all will try to continuous the exercise in terms of I need try to establish traceability matrix for the develop the scenarios the set of emirate scenarios you the testability the traceability basically require to see and report that from the requirements to test cases how the coverage is there how we arrival is trace requirements to test cases similarly test cases to requirements to are different basically.

I mean so we call it as traceability. So both are requires because the coverage define so the test cases requirements is called as a upward traceability and requirements test cases are called as downward traceability okay now, let us try to recap.

(Refer Slide Time: 02:31)



Requirements Specification for Embedded Unit Instrument

S_1. The Embedded Unit Instrument (EUI) software operates in five operational m

- INIT
- OPERATIONAL
- MAINTENANCE
- DOWNLOAD

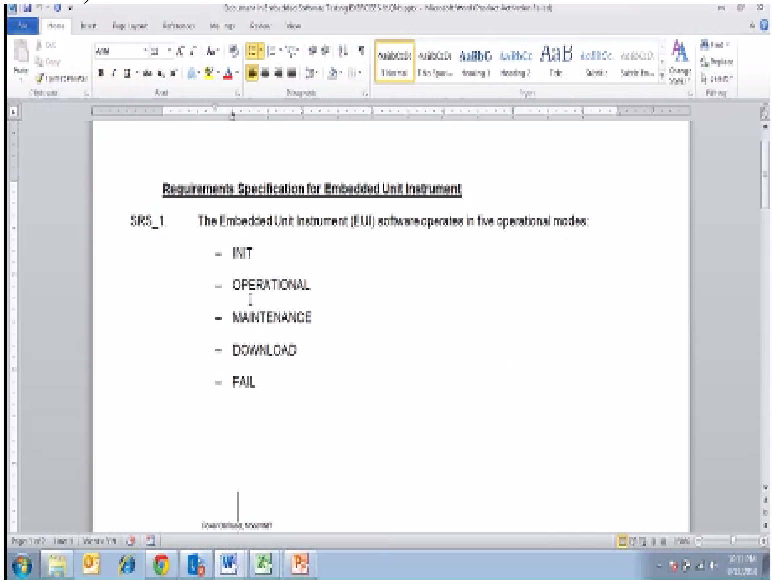
e 1

scenarios from a
n grouping, test
consider for this is
ed below).

- Fill the Test review checklist filling for high-level testing. (with the below sample)

The requirements so SRS1

(Refer Slide Time: 02:33)



Requirements Specification for Embedded Unit Instrument

SRS_1 The Embedded Unit Instrument (EUI) software operates in five operational modes:

- INIT
- OPERATIONAL
- MAINTENANCE
- DOWNLOAD
- FAIL

Talks about generating the instrument 5 operational modes such an INIT, an operational, maintenance, download and fail.

(Refer Slide Time: 02:42)

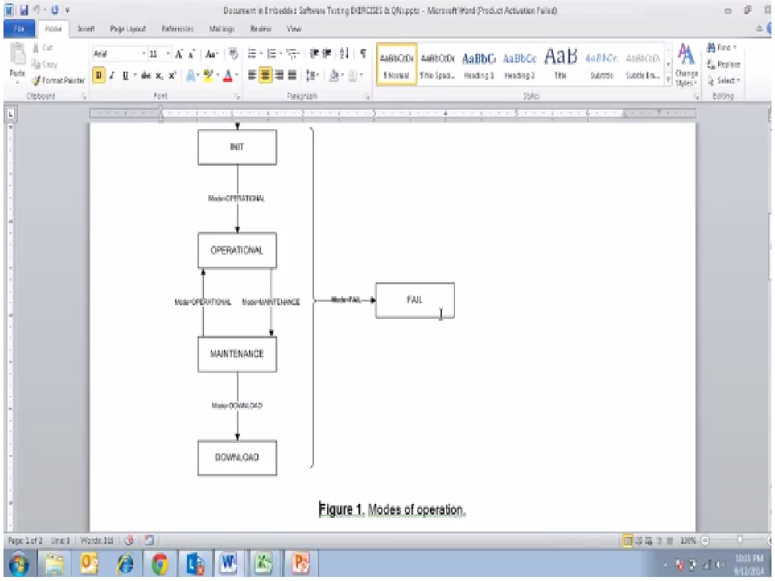
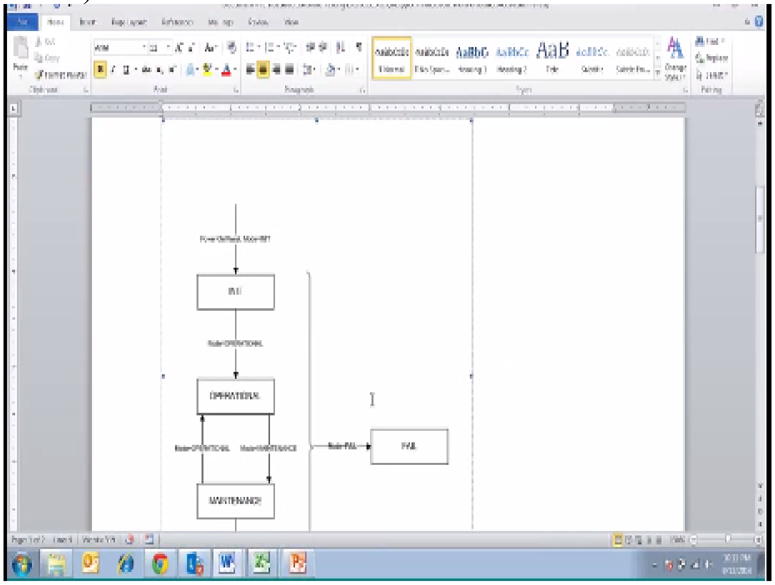


Figure 1. Modes of operation.

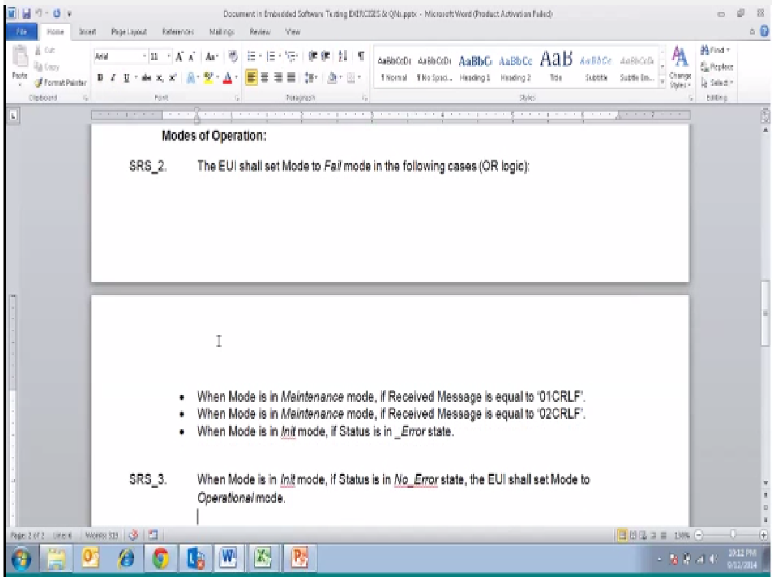
Here we can see different modes of operation.

(Refer Slide Time: 02:45)



How they can enter and exit so this is parallel modes of operation and fail which can go from any topics and downward it any point of time based on the different conditions.

(Refer Slide Time: 03:00)



Modes of Operation:

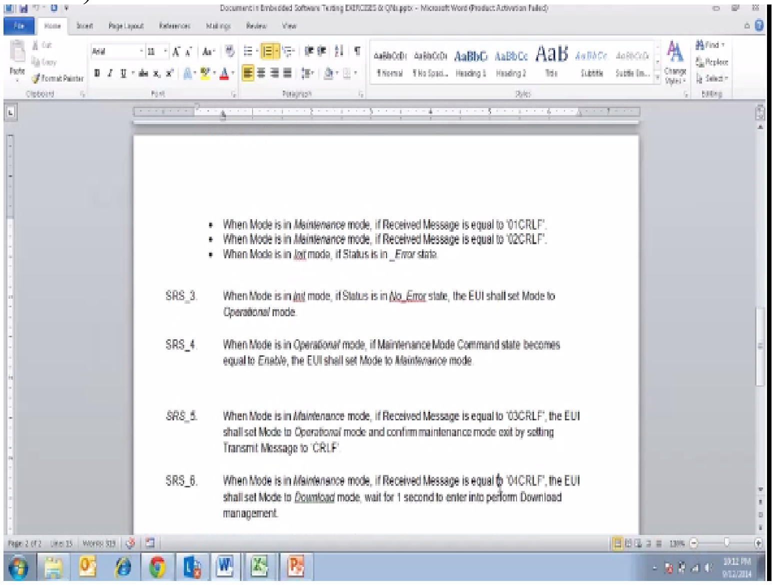
SRS_2. The EUI shall set Mode to *Fail* mode in the following cases (OR logic):

- When Mode is in *Maintenance* mode, if Received Message is equal to '01CRLF'.
- When Mode is in *Maintenance* mode, if Received Message is equal to '02CRLF'.
- When Mode is in *Init* mode, if Status is in *_Error* state.

SRS_3. When Mode is in *Init* mode, if Status is in *No_Error* state, the EUI shall set Mode to *Operational* mode.

So those conditions are nothing but the modes on operations typically SRS2 onwards we have what is the terms of operations requirements I think we had concept this SRS2regarging modes of operations such SRS3 as well as any to mode.

(Refer Slide Time: 03:16)



- When Mode is in *Maintenance* mode, if Received Message is equal to '01CRLF'.
- When Mode is in *Maintenance* mode, if Received Message is equal to '02CRLF'.
- When Mode is in *Init* mode, if Status is in *_Error* state.

SRS_3. When Mode is in *Init* mode, if Status is in *No_Error* state, the EUI shall set Mode to *Operational* mode.

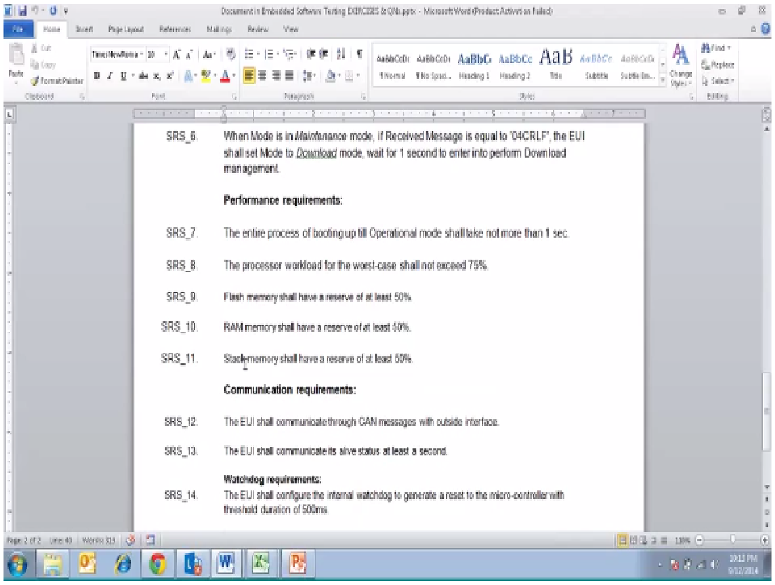
SRS_4. When Mode is in *Operational* mode, if Maintenance Mode Command state becomes equal to *Enable*, the EUI shall set Mode to *Maintenance* mode.

SRS_5. When Mode is in *Maintenance* mode, if Received Message is equal to '03CRLF', the EUI shall set Mode to *Operational* mode and confirm maintenance mode exit by setting Transmit Message to 'CRLF'.

SRS_6. When Mode is in *Maintenance* mode, if Received Message is equal to '04CRLF', the EUI shall set Mode to *Download* mode, wait for 1 second to enter into perform Download management.

SRS4 when a mode of operational mode inter mode and fix again operational mode so like were we have different, modes of operation, SRS to the SRS5 six similarly we have performance requirements.

(Refer Slide Time: 03:40)



SRS_6. When Mode is in *Maintenance* mode, if Received Message is equal to '04CRLF', the EUI shall set Mode to *Download* mode, wait for 1 second to enter into perform Download management.

Performance requirements:

SRS_7. The entire process of booting up till Operational mode shall take not more than 1 sec.

SRS_8. The processor workload for the worst-case shall not exceed 75%.

SRS_9. Flash memory shall have a reserve of at least 50%.

SRS_10. RAM memory shall have a reserve of at least 50%.

SRS_11. Stack memory shall have a reserve of at least 50%.

Communication requirements:

SRS_12. The EUI shall communicate through CAN messages with outside interface.

SRS_13. The EUI shall communicate its alive status at least a second.

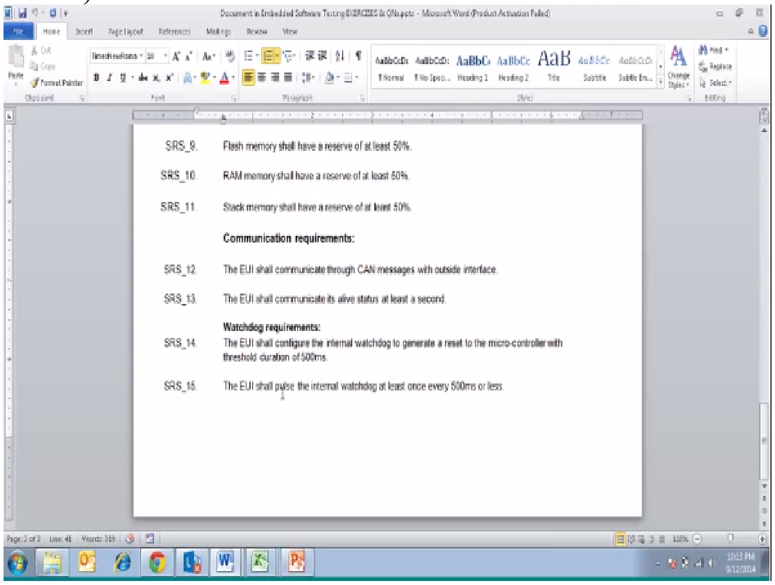
Watchdog requirements:

SRS_14. The EUI shall configure the internal watchdog to generate a reset to the micro-controller with threshold duration of 500ms.

It terms of SRS 7 to 11 here we know that booting time formats and work load for the performance of the processor of the CPU how much it should be an also along with the timing and work load we have the memory related performance requirements such as flash memory Ram memory, stack memory so how we are going to verify those let us try to views some example test cases.

So that a frame work on which the entire a testing technique they cannot further terms of the scenarios as well as execution similarly we have communication requirements.

(Refer Slide Time: 04:26)



SRS_9. Flash memory shall have a reserve of at least 50%.

SRS_10. RAM memory shall have a reserve of at least 50%.

SRS_11. Stack memory shall have a reserve of at least 50%.

Communication requirements:

SRS_12. The EUI shall communicate through CAN messages with outside interface.

SRS_13. The EUI shall communicate its alive status at least a second.

Watchdog requirements:

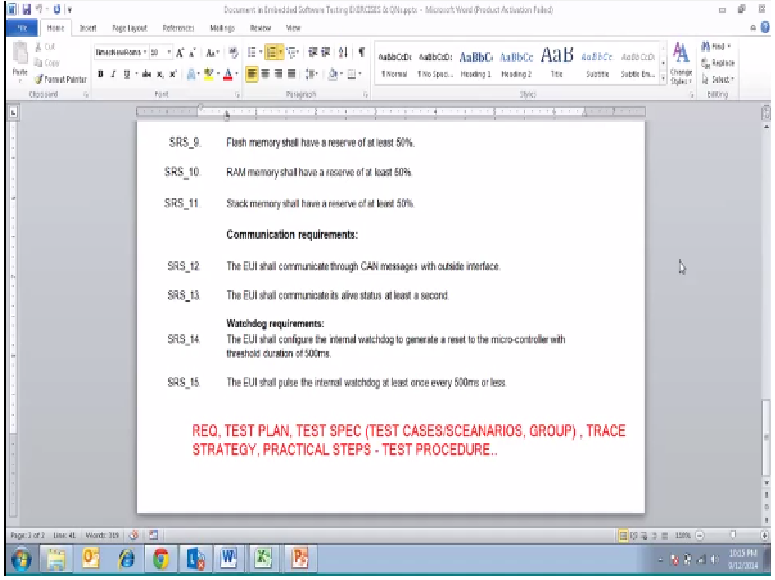
SRS_14. The EUI shall configure the internal watchdog to generate a reset to the micro-controller with threshold duration of 500ms.

SRS_15. The EUI shall pulse the internal watchdog at least once every 500ms or less.

Terms of how the instrument unit can communicate with the external word is through the can messages we know that because of the 1st requirement talks about the different messages it is the received based on different messages it will different modes of operations, so these are the watchdog requirements.

You know that watchdog because with the <500ms with in pass we have to resent these are some of the requirements for which test cases have been develop. Let us try to understand the test cases in one of the session we said about, how it has spec for the test cases should be existing.

(Refer Slide Time: 05:22)



SRS_9. Flash memory shall have a reserve of at least 50%.

SRS_10. RAM memory shall have a reserve of at least 50%.

SRS_11. Stack memory shall have a reserve of at least 50%.

Communication requirements:

SRS_12. The EUI shall communicate through CAN messages with outside interface.

SRS_13. The EUI shall communicate its alive status at least a second.

Watchdog requirements:

SRS_14. The EUI shall configure the internal watchdog to generate a reset to the micro-controller with threshold duration of 500ms.

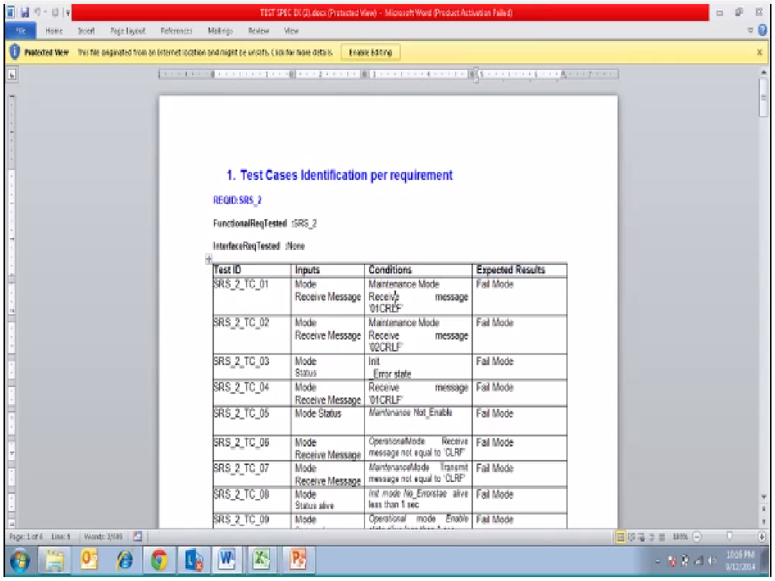
SRS_15. The EUI shall pulse the internal watchdog at least once every 500ms or less.

REQ, TEST PLAN, TEST SPEC (TEST CASES/SCEANARIOS, GROUP) , TRACE STRATEGY, PRACTICAL STEPS - TEST PROCEDURE..

So we need to have the requirement of the command test plan we assume the test plan is available test spec is nothing but the test cases scenarios, corresponding to the requirements then we are going to, group it then we are going to, trace it this are some of the specific activities that we do for developing the test cases and in test specification they want to start the strategy terms of how we can do it then practical steps.

Which are nothing but the test procedure this all we do so this are some of the things that we need to do we know we gone through the example of template how to reflect one of the practical class so let us try to understand the test case perspective how it can be develop okay.

(Refer Slide Time: 06:33)



TEST SPEC DC (2).docx (Protected View) - Microsoft Word (Product Activation Failed)
Protected View
Enable Editing

1. Test Cases Identification per requirement

REQID:SRS_2

FunctionalReqTested :SRS_2

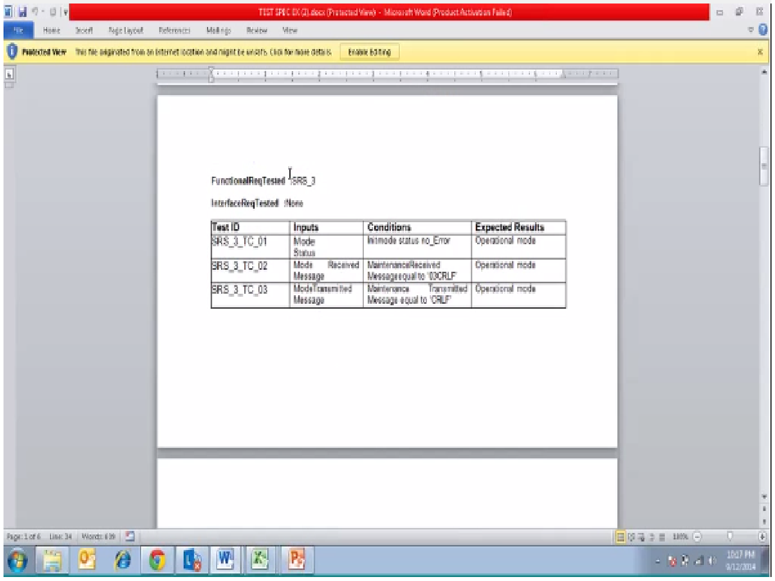
InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_2_TC_01	Mode Receive Message	Maintenance Mode Receive message '01CRLF'	Fail Mode
SRS_2_TC_02	Mode Receive Message	Maintenance Mode Receive message '02CRLF'	Fail Mode
SRS_2_TC_03	Mode Status	Init _Error state	Fail Mode
SRS_2_TC_04	Mode Receive Message	Receive message '01CRLF'	Fail Mode
SRS_2_TC_05	Mode Status	Maintenance Not_Enable	Fail Mode
SRS_2_TC_06	Mode Receive Message	OperationalMode Receive message not equal to 'CLRF'	Fail Mode
SRS_2_TC_07	Mode Receive Message	MaintenanceMode Transmit message not equal to 'CLRF'	Fail Mode
SRS_2_TC_08	Mode Status alive	Init mode No_Errorstae alive less than 1 sec	Fail Mode
SRS_2_TC_09	Mode	Operational mode Enable	Fail Mode

So we have 3 levels of software 4 levels of software requirements in terms of nodes of operation performance requirements and communication requirements for watchdog requirements okay, okay so we know the templates how we are going to use it the 1st section it should talk about test cases identification per each of the requirements we can see whereas each requirements it will 1st identify test case is we gone through the template show should replied like should identify the test id should be eventually for each test case.

And what is the input for that particular test case, so what is the condition? That for this input we should execute then what is the expected result that we going to have.

(Refer Slide Time: 07:34)



TEST SPEC DC (2).docx (Protected View) - Microsoft Word (Product Activation Failed)
Protected View
Enable Editing

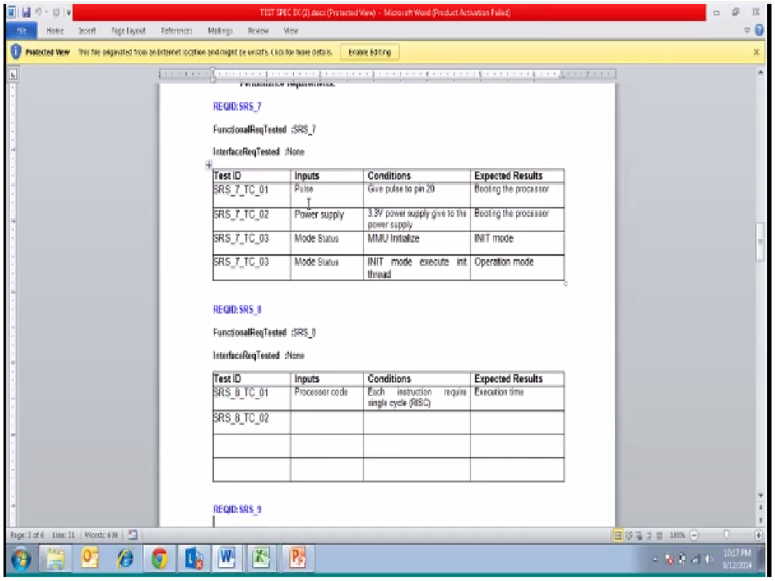
FunctionalReqTested :SRS_3

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_3_TC_01	Mode Status	Initmode status no_Error	Operational mode
SRS_3_TC_02	Mode Received Message	MaintenanceReceived Messageequal to '03CRLF'	Operational mode
SRS_3_TC_03	ModeTransmitted Message	Maintenance Transmitted Message equal to 'CRLF'	Operational mode

Similarly for each of the requirement like we have going to take up is SRS2, SRS3, SRS4 likewise we able to have for each of the requirement

(Refer Slide Time: 07:46)



REQID:SRS_7

FunctionalReqTested :SRS_7

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_7_TC_01	Pulse	Give pulse to pin 20	Booting the processor
SRS_7_TC_02	Power supply	3.3V power supply give to the power supply	Booting the processor
SRS_7_TC_03	Mode Status	MMU Initialize	INIT mode
SRS_7_TC_03	Mode Status	INIT mode execute init thread	Operation mode

REQID:SRS_8

FunctionalReqTested :SRS_8

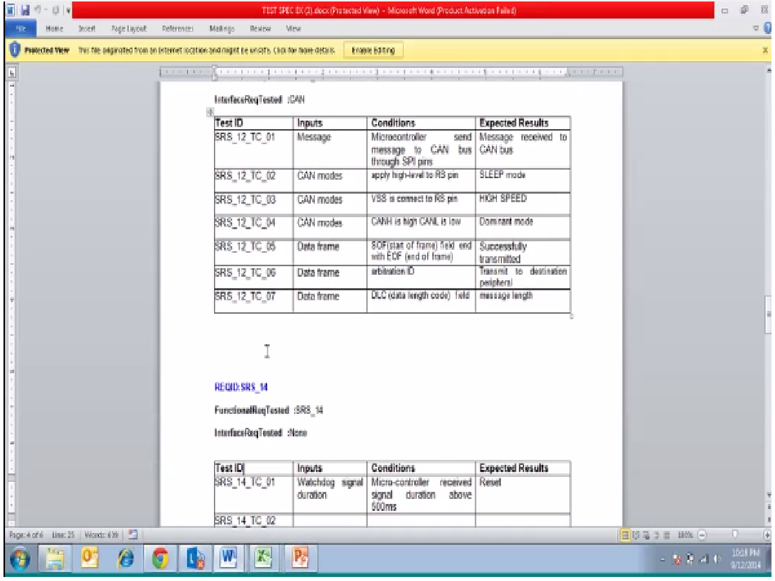
InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_8_TC_01	Processor code	Each instruction require single cycle (RISC)	Execution time
SRS_8_TC_02			

REQID:SRS_9

So let show we have to develop test cases and the grouping is based on, different functionality like 1st few cases we concentrate on the modes of operation next one we should, performance requirements and next grouping is based on the timing next group is based on the watchdog group it but before grouping 1st of all we need to have this the test cases.

(Refer Slide Time: 08:24)



InterfaceReqTested :CAN

Test ID	Inputs	Conditions	Expected Results
SRS_12_TC_01	Message	Microcontroller send message to CAN bus through SPI pins	Message received to CAN bus
SRS_12_TC_02	CAN modes	apply high-level to RS pin	SLEEP mode
SRS_12_TC_03	CAN modes	VSS is connect to RS pin	HIGH SPEED
SRS_12_TC_04	CAN modes	CANH is high CANL is low	Dominant mode
SRS_12_TC_05	Data frame	SOF(start of frame) field end with EOF (end of frame)	Successfully transmitted
SRS_12_TC_06	Data frame	arbitration ID	Transmit to destination peripheral
SRS_12_TC_07	Data frame	DLC (data length code) field	message length

REQID:SRS_14

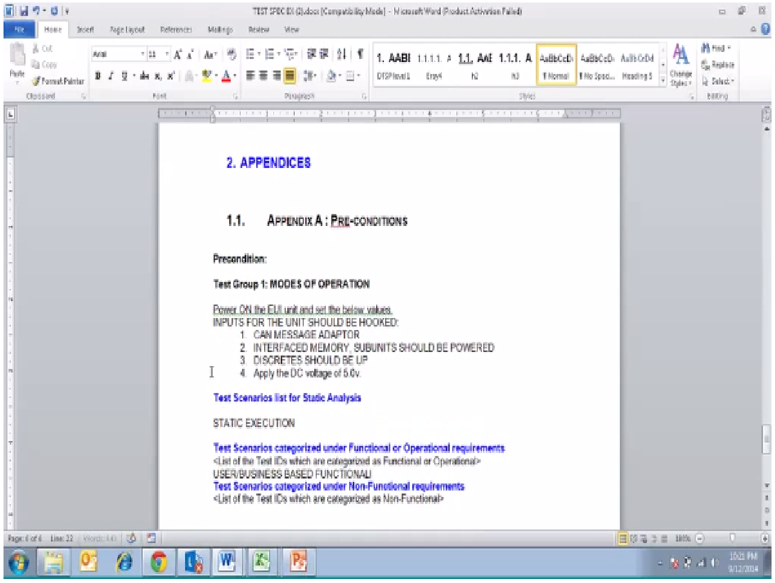
FunctionalReqTested :SRS_14

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_14_TC_01	Watchdog signal duration	Micro-controller received signal duration above 500ms	Reset
SRS_14_TC_02			

Individually identify for each of the requirements it is very important and basically we can adjust like what test case can go into what and hoe can we trouble and duplicate or redundant any aspects that we can add 1st term from as thing is there are the test cases.

(Refer Slide Time: 08:42)



2. APPENDICES

1.1. APPENDIX A : PRE-CONDITIONS

Precondition:

Test Group 1: MODES OF OPERATION

Power ON the EUI unit and set the below values.
INPUTS FOR THE UNIT SHOULD BE HOOKED:

1. CAN MESSAGE ADAPTOR
2. INTERFACED MEMORY, SUBUNITS SHOULD BE POWERED
3. DISCRETES SHOULD BE UP
4. Apply the DC voltage of 5.0v.

Test Scenarios list for Static Analysis

STATIC EXECUTION

Test Scenarios categorized under Functional or Operational requirements
<List of the Test IDs which are categorized as Functional or Operational>
USER/BUSINESS BASED FUNCTIONAL\
Test Scenarios categorized under Non-Functional requirements
<List of the Test IDs which are categorized as Non-Functional>

For each of the requirements also we have done with all that then we have to grouping for the grouping is done and we know that for each group this is set of conditions need to be existing before we start the practical step are the step procedures those conditions they are called as pre conditions so pre conditions for example for are the test case we can have, test group 1, so which is nothing but relating modes of operation.

So what are things that we need we need to power on the unit and set on the, and set the below values, below values could be something like a what are the inputs for INIT should be hook so what are those going could be can message adapter should be up similarly any interconnected interfaced in a array sub units, should be powered okay so what will describe could be up likewise.

We have going to have pre conditions so the preconditions can be unique or different based on the take the group, group that we are using so for modes of operation we can message should be coming because we cannot the test the most of operations requirements similarly, we need to apply power voltage of 5.1, 5.0 so like will we would not have not and some of the requirements we may not able to do dynamically.

Or when the target is running so those, list of static analysis where we need to do it static execution we seen the code so like performance requirements some of like memory map those things we need to execute on the target so those cases we should list out here telling that, that group of a that specific scenarios can be tested statically or static analysis based on that similarly the templates talks about the scenarios applied.

Under the functional of operation requirements let us the test cases scenarios the categories and the non functional requirements act that means basically whatever the user listen is, based functionality this various it should be doing some like we should, rotate the motor that is a actual requirement so the non functional requirements such as should power of within by seconds something of those test cases.

There are not this is related or user related. That is all performance or supporting of an requirements so likewise we going to have the appendices updated for the test case the test case document that is one section that we have and the 1st section what we have seen the test cases what we have seen in test cases identification for each requirement.

So these what we have going to do So for example will study if you requirements and few test cases, okay so SRS2 we know that what we talk about the modes of operation like we know that the instrument.

(Refer Slide Time: 13:04)

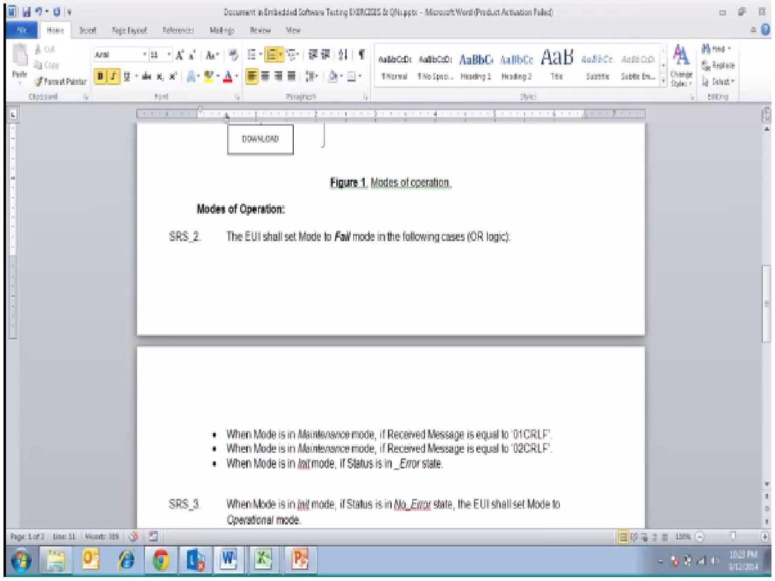

Document in Embedded Software Testing EXERCISES & QN.pptx - Microsoft Word (Product Activation Failed)
DOWNLOAD

Figure 1. Modes of operation.

Modes of Operation:

SRS_2. The EUI shall set Mode to *Fail* mode in the following cases (OR logic):

- When Mode is in *Maintenance* mode, if Received Message is equal to '01CRLF'.
- When Mode is in *Maintenance* mode, if Received Message is equal to '02CRLF'.
- When Mode is in *Init* mode, if Status is in _*Error* state.

SRS_3. When Mode is in *Init* mode, if Status is in *No_Error* state, the EUI shall set Mode to Operational mode.

Shell that we know said the mode failure mode in the fallowing cases based a logic so one of this conditions one of this 3 conditions if you it will occurs it will going to enter into it is the what we thing so here what is the input, if content of this having the inputs so let us try to understand content for one condition when is mode is in maintains node so mode is a, and received message so what is the output, fail that be second mode.

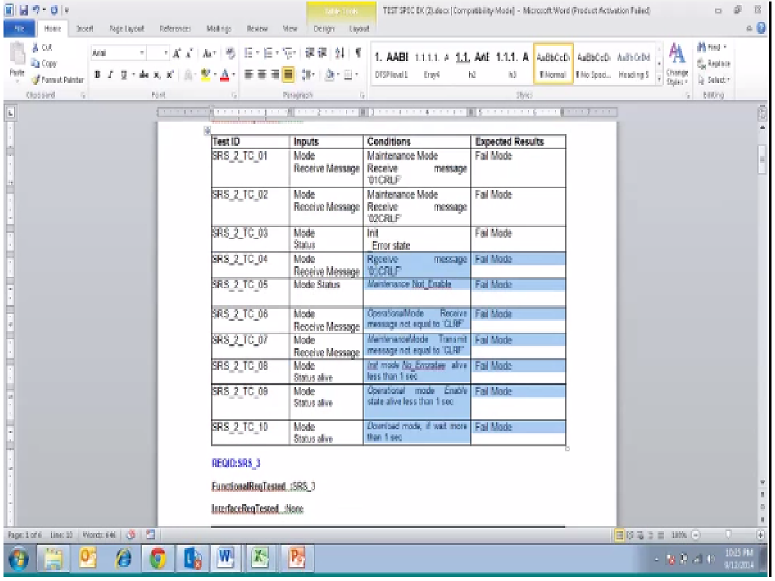
Set to failure condition is what this is the confusion the 1st condition, 2nd condition, 3rd condition one of this condition exit it is going to enter into failure mode okay let us try to draw a example case 1st case let us try to define we know that mode is an end what is an input receive message is an input so because can message if we having the dependency right so this one input the mode is available this 2 this are inputs there some apply some conditions on that input. And we are going to get the expect this so what will happen id mode is in maintain this input is satisfied.

When we get receive message are here once occurs do not worry about a what is CRL message it just an because we are focusing are a empathizes for test this potion not the actual data that maybe robustness in terms of a different message other than 0101 what will happen that may be define later but we will start with a normal case saying that. We fated t o inputs and maintenance mode and the message are 0.

And there under directly as per the requirement we should go to fail mode so this we have tested here and the next case we can see we are trying to test with us silent message o to CRLF that cases also fail mode similarly the 3 rd condition when the mode in is INIT mode in the status in error state that is doing initialization.

If any error occurs it will go for error so this are straight forward test cases that we have seen for this requirement similarly 4 conditions are there we are going to apply as per that okay, next the status define.

(Refer Slide Time: 16:13)



Test ID	Inputs	Conditions	Expected Results
SRS_2_TC_01	Mode Receive Message	Maintenance Mode Receive message '01CRLF'	Fail Mode
SRS_2_TC_02	Mode Receive Message	Maintenance Mode Receive message '02CRLF'	Fail Mode
SRS_2_TC_03	Mode Status	Init _Error state	Fail Mode
SRS_2_TC_04	Mode Receive Message	Receive message '0_CRLF'	Fail Mode
SRS_2_TC_05	Mode Status	Maintenance Not_Enable	Fail Mode
SRS_2_TC_06	Mode Receive Message	OperationalMode Receive message not equal to 'CLRF'	Fail Mode
SRS_2_TC_07	Mode Receive Message	MaintenanceMode Transmit message not equal to 'CLRF'	Fail Mode
SRS_2_TC_08	Mode Status alive	Init mode No_Errstate alive less than 1 sec	Fail Mode
SRS_2_TC_09	Mode Status alive	Operational mode Enable state alive less than 1 sec	Fail Mode
SRS_2_TC_10	Mode Status alive	Download mode, if wait more than 1 sec	Fail Mode

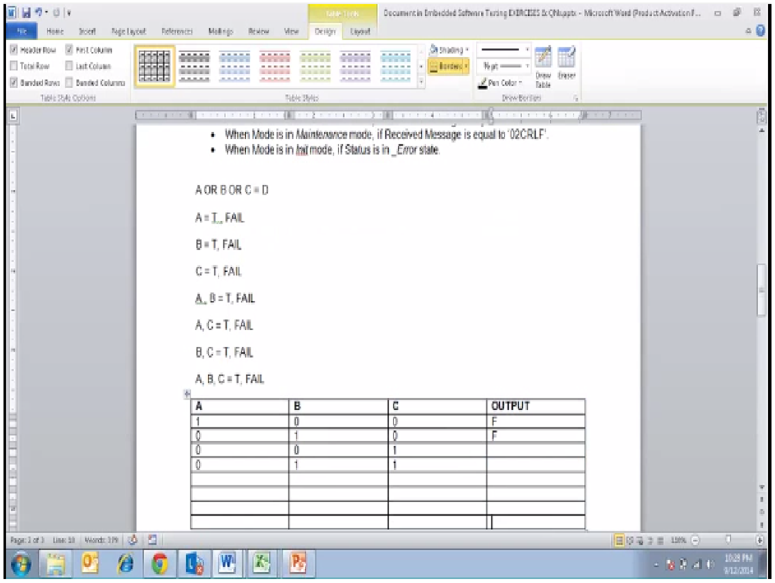
REQID:SRS_3

FunctionalReqTested :SRS_3

InterfaceReqTested :None

Now these or logic so that means any of this conditions multiple of this occurs so we have something like a or b or c =d, d means failure.

(Refer Slide Time: 16:32)



- When Mode is in *Maintenance* mode, if Received Message is equal to '02CRLF'.
- When Mode is in *Init* mode, if Status is in _*Error* state.

A OR B OR C = D

A = T,, FAIL

B = T, FAIL

C = T, FAIL

A,, B = T, FAIL

A, C = T, FAIL

B, C = T, FAIL

A, B, C = T, FAIL

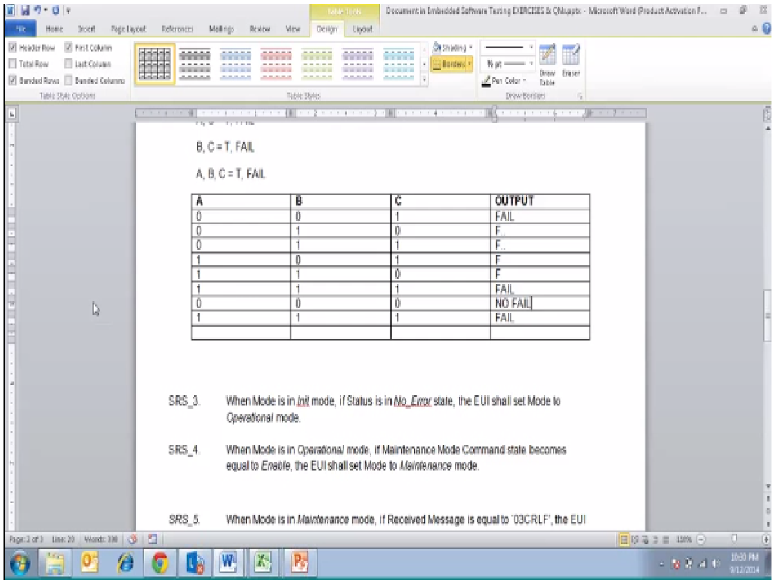
A	B	C	OUTPUT
1	0	0	F
0	1	0	F
0	0	1	
0	1	1	

So they go to have the true table are thing were either a can be is a true b can be set a true c can be set a true in that condition it should go to fail and next one a b set to a c set a true a b c set a true in this condition also we go to go for failure likewise we going to have this called enter this they are the modified condition mission logic based on the conditions different decisions are there, here.

A becomes true condition will be fail can b is true condition will fail. C is true will be a go for both are true condition will be failed right both are to fail similarly a b c last case it is becomes to true b fail that we have what was left everything will left right everything will done so this kind of things it is a god to have something like a true table we can define that way it will be easier actually, maybe to have one table fallow a added.

So we can a b c and the output this what is trying to set for different values so we know that combinations of a b c w try to set 0 10 0 output will be failed similarly 0 1 0 will be fail, became 0 01 next condition 011, typical route table we are going able to have a for all the combinations the moreover the may not require the we know the result of this.

(Refer Slide Time: 19:01)



B, C = T, FAIL

A, B, C = T, FAIL

A	B	C	OUTPUT
0	0	1	FAIL
0	1	0	F..
0	1	1	F..
1	0	1	F
1	1	0	F
1	1	1	FAIL
0	0	0	NO FAIL
1	1	1	FAIL

SRS_3. When Mode is in *Init* mode, if Status is in *No_Error* state, the EUI shall set Mode to *Operational* mode.

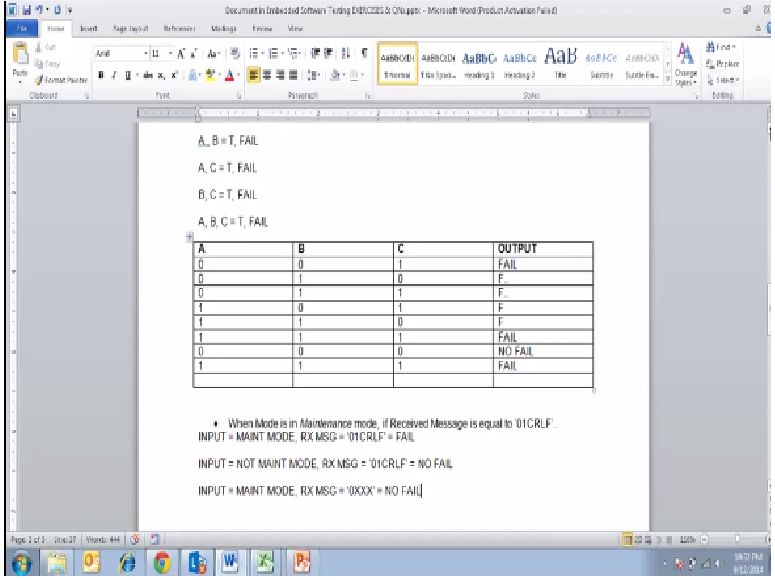
SRS_4. When Mode is in *Operational* mode, if Maintenance Mode Command state becomes equal to *Enable*, the EUI shall set Mode to *Maintenance* mode.

SRS_5. When Mode is in *Maintenance* mode, if Received Message is equal to '03CRLF', the EUI

Why we need why we may not need this independent we need to prove and I will table expand that, okay here we can see the multiple route table combinations like, c is true output is false and b is true what will be false similarly both are true will come false and now we are going to set a as 1 and here.

We are going to very clear this going to be still the last condition is 000 it will became true that means true means is a failure or not a failure here mis case this, this fail right one more condition is all should be here, failure f means basically failure, okay, so, so for this of case we need to develop the test cases adding inputs accordingly okay.

(Refer Slide Time: 20:18)



A,, B = T, FAIL

A, C = T, FAIL

B, C = T, FAIL

A, B, C = T, FAIL

A	B	C	OUTPUT
0	0	1	FAIL
0	1	0	F..
0	1	1	F..
1	0	1	F
1	1	0	F
1	1	1	FAIL
0	0	0	NO FAIL
1	1	1	FAIL

- When Mode is in *Maintenance* mode, if Received Message is equal to '01CRLF'.

INPUT = MAINT MODE, RX MSG = '01CRLF' = FAIL

INPUT = NOT MAINT MODE, RX MSG = '01CRLF' = NO FAIL

INPUT = MAINT MODE, RX MSG = '0XXX' = NO FAIL

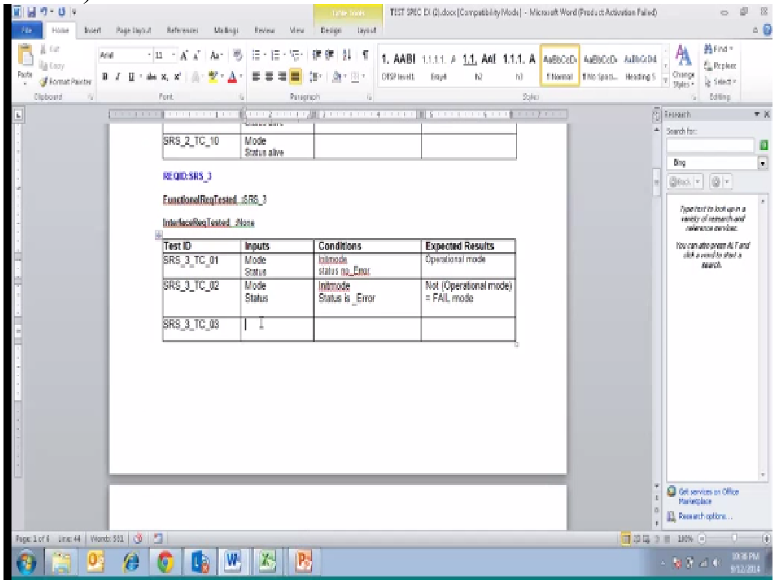
Now we have tested about this normal cases of the truth table and conditions next within each of day this we can identify other test case or other group of test case such as we draw that only so input is mode maintenance mode and receive message is o1 here CLRF that we already tested we should became fail okay, next condition with in this test case input mode is lot of maintenance that is not a we do not know what mode.

It is so it will became no fail correct because we are heard were in that the next one in put is maintenance mode there is the received message is not 01 this something else existence whatever it is so this also became a fail so like this the combination that we need to have that is very important, so this is the then we run with a test cases and whatever to occur okay the next thing we need to take care is independent, very important terms we can see the above case abc set here that occurs for now.

We have proven only b has change for 0 to 1 there are remaining things are constant there also we need to right because remaining things is not constant here we have seen b has changes along with c also we do not know because of what the failure would have occurs so better to have independency such as, that thing is back to 0 we are prove the independence this because of this only failure has occurred and not.

Because of this or combinations So it a independent were went to similarly for a other cases also like we have a, 000 of here is has this to basic because of this getting the fail so that so the affect of inputs we need to that tell it as properly that independence is one of the important thing that they look for when we are write the test cases that is are we going to have for different set of a combinations and, inputs for that particular requirements okay, next one

(Refer Slide Time: 23:57)



SRS_2_TC_10	Mode Status alive		

REQID:SRS_3

FunctionalReqTested :SRS_3

InterfaceReqTested :None

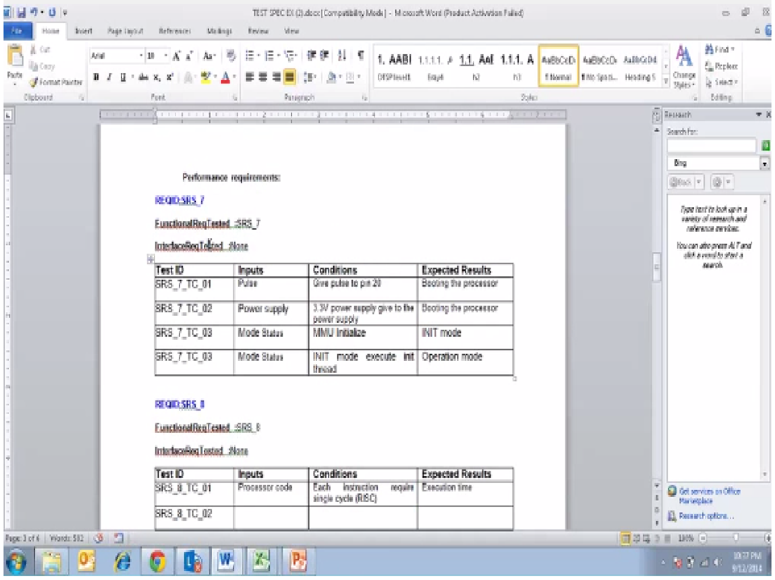
Test ID	Inputs	Conditions	Expected Results
SRS_3_TC_01	Mode Status	Initmode status no_Error	Operational mode
SRS_3_TC_02	Mode Status	Initmode Status is _Error	Not (Operational mode) = FAIL mode
SRS_3_TC_03			

Next one is SRS3 is talks about a what, in the mode is INIT mode and the status is no error state that means the error and the INIT is in, and the mode is INIT mode, the instrument shell said into operational so 2 inputs mode in status init mode no error with the operational mode, mode received message, this not a right test case maintenance of, okay let started with this, mode status okay on init mode okay, let put in similar.

It is in init mode, status is error so what we will became it should not be part of operational that is 1st shown because the requirements is it should be there in operational only when we require so they going to the operational requirements, is clear for this probably by seeing if it is not of operational mode what is the mode?

This try to go through the requirement will come to mode requirement is a, this is go for the requirement because if it is going to be no error state then it will be operational otherwise we can get as failure that is what it said right, okay so nothing but it is, failure, so likewise we go to add test cases by looking at the requirement so for 3 we have done, okay.

(Refer Slide Time: 26:33)



Performance requirements:

REQID:SRS_7

FunctionalReqTested :SRS_7

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_7_TC_01	Pulse	Give pulse to pin 20	Booting the processor
SRS_7_TC_02	Power supply	3.3V power supply give to the power supply	Booting the processor
SRS_7_TC_03	Mode Status	MMU Initialize	INIT mode
SRS_7_TC_03	Mode Status	INIT mode execute init thread	Operation mode

REQID:SRS_8

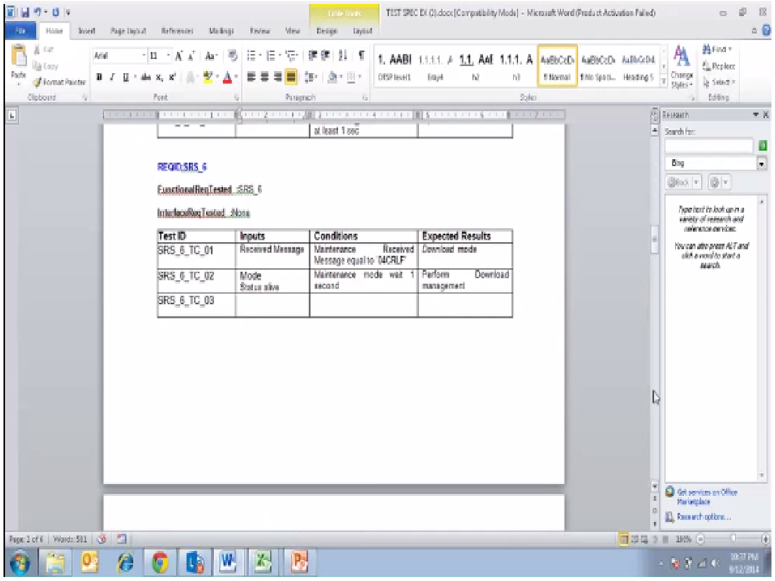
FunctionalReqTested :SRS_8

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_8_TC_01	Processor code	Each instruction require single cycle (RISC)	Execution time
SRS_8_TC_02			

For this regarding maintenance status the status live messages and 6 is about.

(Refer Slide Time: 26:40)



at least 1 sec

REQID:SRS_6

FunctionalReqTested :SRS_6

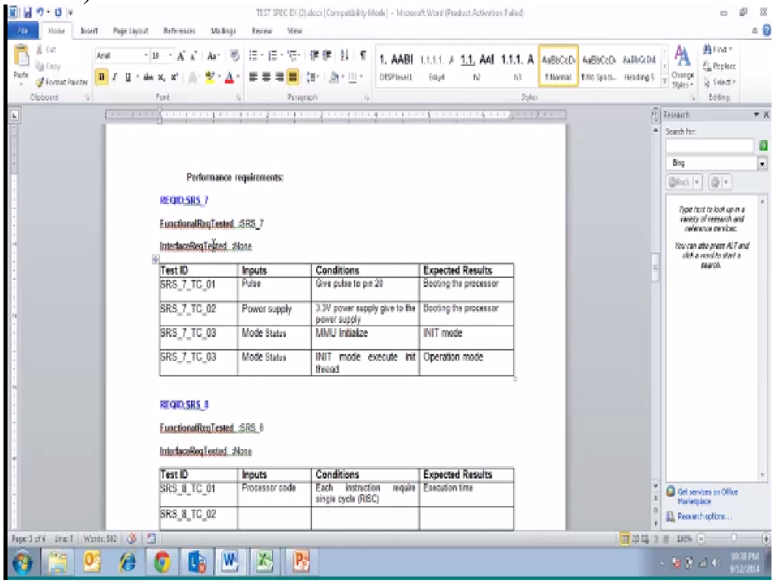
InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_6_TC_01	Received Message	Maintenance Received Message equal to '04CRLF'	Download mode
SRS_6_TC_02	Mode Status alive	Maintenance mode wait 1 second	Perform Download management
SRS_6_TC_03			

A the status of messages is 04 likewise we going to added I can see particular SRS and for that test cases we want to add one more thing we would have notice this interface because tested so what is happen is some of the signals that has having the dependency or a, particular requirements suppose for this functional requirement for the signal dependency is there some describe and that immediate.

The list and why listing that we are also trying to test particular interface requirements that is also saw it as an input may not be a functional requirements this is a dependent in the message, okay.

(Refer Slide Time: 27:32)



Performance requirements:

REQID:SRS_7

FunctionalReqTested :SRS_7

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_7_TC_01	Pulse	Give pulse to pin 20	Booting the processor
SRS_7_TC_02	Power supply	3.3V power supply give to the power supply	Booting the processor
SRS_7_TC_03	Mode Status	MMU Initialize	INIT mode
SRS_7_TC_03	Mode Status	INIT mode execute init thread	Operation mode

REQID:SRS_8

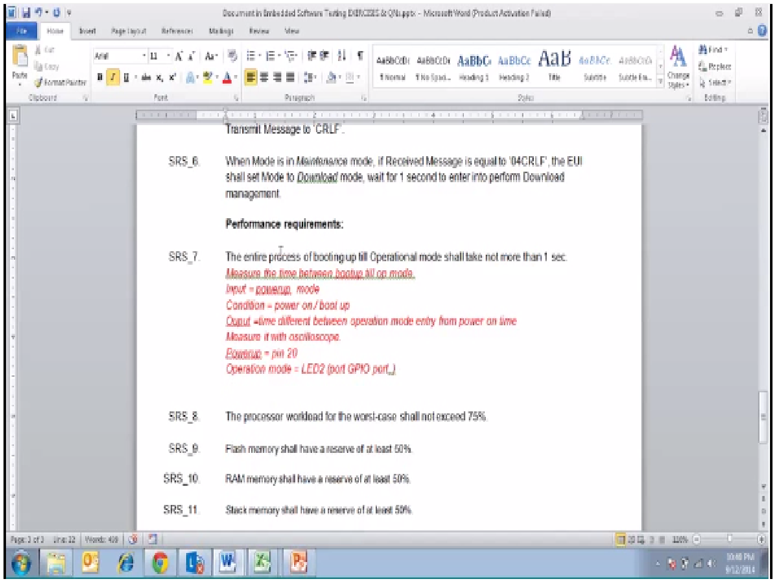
FunctionalReqTested :SRS_8

InterfaceReqTested :None

Test ID	Inputs	Conditions	Expected Results
SRS_8_TC_01	Processor code	Each instruction require single cycle (RISC)	Execution time
SRS_8_TC_02			

So next for is performance requirements let try to understand what performance that ports the entire process of booting up till operational mode.

(Refer Slide Time: 27:42)



Transmit Message to 'CRLF'.

SRS_6. When Mode is in *Maintenance* mode, if Received Message is equal to '04CRLF', the EUI shall set Mode to *Download* mode, wait for 1 second to enter into perform Download management.

Performance requirements:

SRS_7. The entire process of booting up till Operational mode shall take not more than 1 sec.
Measure the time between bootup till op mode.
Input = powerup, mode
Condition = power on / boot up
Ouput =time different between operation mode entry from power on time
Measure it with oscilloscope.
Powerup = pin 20
Operation mode = LED2 (port GPIO port_)

SRS_8. The processor workload for the worst-case shall not exceed 75%.

SRS_9. Flash memory shall have a reserve of at least 50%.

SRS_10. RAM memory shall have a reserve of at least 50%.

SRS_11. Stack memory shall have a reserve of at least 50%.

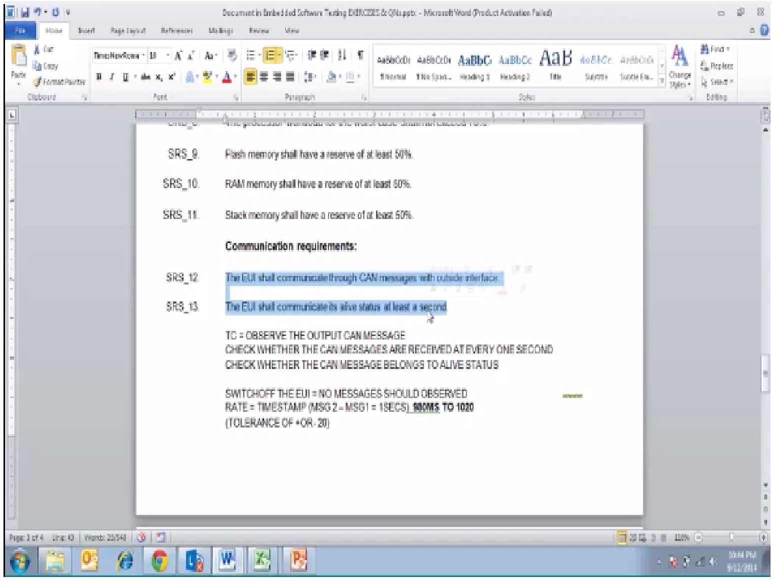
Should will not take more than second this upon power of w should not take more than 1 second to became operational mode so how I going to do there is a 1 inputs that means we need to measure right, measure the time that is what is says between boot up till or upward this is something like assert the test case we do not have a negative or were what is called a we are expecting something which some actions here.

Actions is we need boot up how is going to happen upon power of that is an input, input is power up and mode and condition is, or entire the sub and boot up how it occurs it is a power up, power on we can say the action is called as power on, so output is, time difference between operational mode entry on power on time so, something is called okay this we need to topic points or e do not need to having points are depending on.

What should occurs processor will be suppose power up have a pin number 20 where we can see the glitch of the signal pulse that certain level 5 volts if it is we can absorb that with one pin captured there and operational mode if it is tied up with the LED say LED2 that have fault something like a GPI port whatever have it is so this 2 we can see it so that we can see the reference as 1 for a pin other one has a LED port when it is going to became operational mode so that difference.

We can find out the help of this so that will became a test case inters of visible so input is, power up or so we needed to give a pulse to for pulse are reset it been 20 input the processor here using pulse scope, what will happen is expected result is, <1 it is what is a expect we see right so that we have to measure it with a force apply try to validate the test case, okay next we will take of requirement we have communication requirements okay,

(Refer Slide Time: 31:22)


Document in Embedded Software Testing EXERCISES & QNs.pptx - Microsoft Word (Product Activation Failed)

SRS_9. Flash memory shall have a reserve of at least 50%.

SRS_10. RAM memory shall have a reserve of at least 50%.

SRS_11. Stack memory shall have a reserve of at least 50%.

Communication requirements:

SRS_12. The EUI shall communicate through CAN messages with outside interface.

SRS_13. The EUI shall communicate its alive status at least a second.

TC = OBSERVE THE OUTPUT CAN MESSAGE
CHECK WHETHER THE CAN MESSAGES ARE RECEIVED AT EVERY ONE SECOND
CHECK WHETHER THE CAN MESSAGE BELONGS TO ALIVE STATUS

SWITCHOFF THE EUI = NO MESSAGES SHOULD OBSERVED
RATE = TIMESTAMP (MSG 2 – MSG1 = 1SECS) **980MS TO 1020**
(TOLERANCE OF +OR- 20)

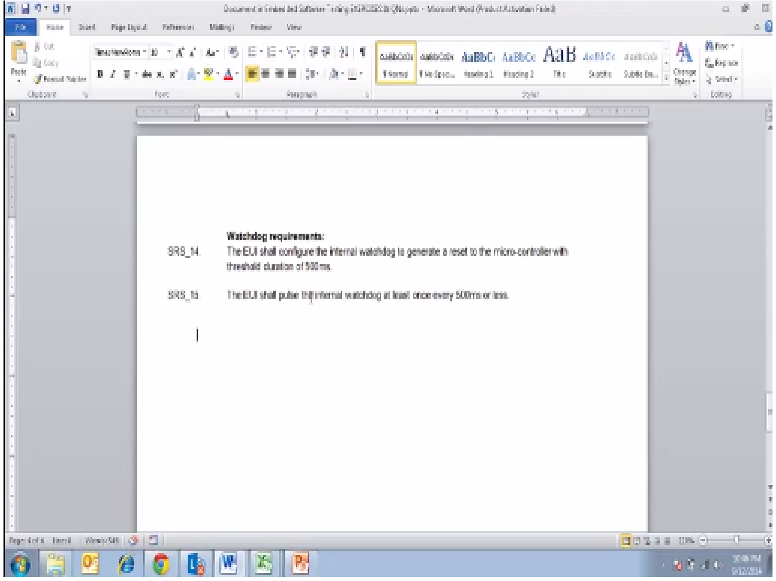
Here we can see 2 requirements are there instruments shell communicate through scan messages without interface the instrumental communicate alive status, one second atleast 1 second at these 2 things are there we can combine this requirements we have multiple cases and a one group we saying that proving this also prove that is communicating and this is also communicate which status one second so it is just a validation of a this basic we know verification is like there is no input or that we can tree call we can modify.

So what should have test cases we have absorb the output can message check whether it can message are receive at every 1 second next 1 check whether we can message belongs to alive status is are the important steps to validate this requirement right okay next negative type for, when the instrument is powered this is going to happen so switch of EUI what will happen is no messages or absorb, no messages should be absorb outside. So this is proving that the instrumental will communicate call outside.

The box and , the related which can messages are coming how we can verify the is based on the time stand because, we can message will have a time stage taking a longer outside in the host that longer that make difference between the SH1 and SH2 it should be 1 see it should be may not be exactly 1 sec something like it depends again to 100 here tolerance of a ,+ or-20 we can see it can fall anything between a to 1020 so depends on type of system tolerance will be there that exact here.

So we will specify format accordingly saying that for this tolerance it should work so that way we need to verify the such requirements any timing and the un log or any of the outputs that away the tolerance we can verify that so this know communication requirements are the tested with the test cases us.

(Refer Slide Time: 34:32)



Watchdog requirements:

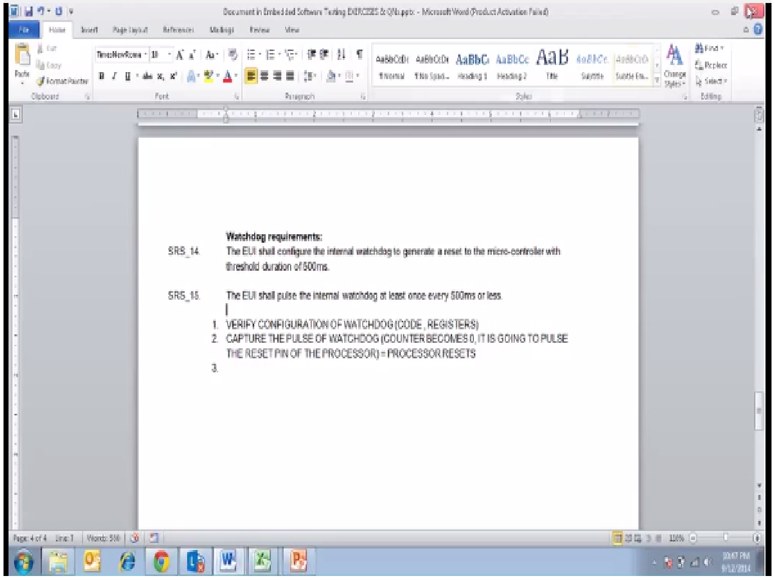
SRS_14. The EUI shall configure the internal watchdog to generate a reset to the micro-controller with threshold duration of 500ms.

SRS_15. The EUI shall pulse the internal watchdog at least once every 500ms or less.

The last one being the project of requirements so there are 2 requirements here the instrumentally configure the what a generate the reset to make a control with a result to this one so what it means is it has a configuration saying that we need it is upend operational, we should be configure for a final set that the watchdog will be satisfied with the pulse of <100 sec that is the next requirement.

If it is not watchdog is groping to reset. Because it is configure for 5 minutes right okay so, here a cases we need to verify in since that,

(Refer Slide Time: 35:39)



Watchdog requirements:

SRS_14. The EUI shall configure the internal watchdog to generate a reset to the micro-controller with threshold duration of 500ms.

SRS_15. The EUI shall pulse the internal watchdog at least once every 500ms or less.

1. VERIFY CONFIGURATION OF WATCHDOG (CODE , REGISTERS)
2. CAPTURE THE PULSE OF WATCHDOG (COUNTER BECOMES 0, IT IS GOING TO PULSE THE RESET PIN OF THE PROCESSOR) = PROCESSOR RESETS
3.

1st is verify the configuration of the watchdog this is theoretically or facing the code or spacing the registers we can validate dynamically we can validate such a weather we can capture the pulse of the watch dog usually watch dog is the circuitry , it has a counter and that counter is configure 500 and this is the decrement counter.

This when it comes 0 it is going to, it is going to reset or pulse the reset pin of the processor such that the processor gets resets, how we are going to eliminate this so make sure that you r configure the watch dog or you make it hang that program is that going to be or capture the pulse so what is the responsible for pulsing he watch dog let should be <500 sec so that is what the 2nd requirements.

Atleast once in every 500 sec this in should be there so 1 is configuration and other one is that the pulse is properly able to verify this though we can verify the watchdog certain kind of requirement okay so likewise we are going to have test cases define test id inputs conditions expect the results, and, appendices we have study now having seen all these let us try to understand the review check list.

We have do not it has a testing available or requirements whatever have last the test case and the requirements and how procedures can be written of this these got we understood now having done this we need to understand this checklist usually this will be a getting criteria for the test plan of test cases to measure that the rules of the particular verification of the test cases or adding to the requirement have as per the name.

That is what we are going to, okay so these are an example template generally we use it is again specify to the specific process that is been allowed the type of elements so let us go through this generate to probably, on what we have done so far okay 1st one an templates fallowed the this case system this we had a document this template which will have prove on right, are the validated classes are like all inputs of final requirements.

We are seen the equivalence causes in terms of the complete truth table, and set of range in terms inputs were we consider are limit values used for the cases are here this know the signal can used so there is a limit in terms of the CRO the values coming down where we have used for inputs structure in a table are bounce 1 terminate value of the table we used in the class there is no structured input tables this is not allocate here.

Essentially see the functional analysis do the allocations, are particular test cases created for time related functions and state missions this put we have here done the machines here, is one term words test, test case inputs here we have the negative different values of the we try to put it this what we have this identification format of test cases it we have sent he every test cases of its own unique this reach high level requirement covered by atleast 1.

Test case and correct 1 which very important this is also so based on the review we need to till this templates have done for each of the requirements this reach the test case template covered by scenario and very this, the test cases like we have a test case for correct identify, are values are inputs and outputs define and reach as this is a qualified as normal mode this probably we are not the this list have do.

We done revaluation of test cases which will not value thing in a order when we have a different modes that was test indications should report saying that know what is my action of this basics it is a test coverage analysis I think we do not have so there is a 2 session guideline and other wise is proves, proves are mandatory for the elements is know also called as deviation and functional without which the report will be acceptable this designated and authority as well as guidelines description of the scenarios what's it is I would say know we can add on description are the expected results.

Define for each , or the specification define these test case what we have to use and identify the test match or the set of the 3 condition is define for each case in the exits so which is test case check list there we have test cases the object the description that was saying is an that is what we require of terms of the develop test case of the requirements that is are test case is different, this will continue the next practical session in terms of upward a traceability as we set the test cases requirements is called as upward traceability this requirement of test cases, okay.