RVC and MC-Soft Framework In SystemC Model Development

(Ver1.2)

Red color: updated/added content

Summary :

This document describes the detail framework between RVC and MC-Soft in SystemC model development.

Reference Document

Model Development Flow and Deliverable Rule - Ver2.0 (Masashi Watanabe - Mar.2010)

Contents

1. Introduction....4
2. Term definition....4
3. Model development flow....4
3.1. Design plan....5
3.2. Internal specification....7
3.3. Design & Verification....8
3.4. Final review....9
4. Content of deliverable documents....10
4.1. Management table....10
4.2. Detail requirement....11
4.3. Internal specification....12
4.3.1.Model summary....12
4.3.2.Supported features....12
4.3.3.Block diagram....13
4.3.4.List of implemented registers....13
4.3.5.Port behavior....13
4.3.6.Direction for user....14
4.3.7.Flow diagram....14
4.3.8.Function description....15
4.3.9.Limitation....15
4.3.10.Related document....15
4.4. Verification specification....15
4.4.1.Verification description....15
4.4.2.Checklist....17
4.5. Verification result....17
4.5.1.Verification result description....17
4.5.2.Checklist with bug information....18
4.6. User manual....19
4.7. Traceability table....20
4.8. Deliverable table....20
4.9. Issue management list....20
4.10. Confirmation list....21
5. Management rule....21

5.1. Review and Approval....22
5.2. Document name....22

Index of Tables

Table 2.1: Term Definition....4
Table 3.1: Expected output of Design Plan's deliverable....6
Table 3.2: Expected output of Internal Specification's deliverable....7
Table 3.3: Expected output of Design & Verification's deliverable....9
Table 3.4: Expected output of Final Review's deliverable....10
Table 4.1: The content of management table....10
Table 4.2: The example of management table....11
Table 4.3: The content of Detail Requirement....11
Table 4.4: The content of Supported Features....12
Table 4.5: The content of Register List....13
Table 4.6: The content of Port List....13
Table 4.7: The content of Direction For User....14
Table 4.8: The content of Flow Diagram....14
Table 4.9: The content of Verification Description....15
Table 4.10: The content of Checklist....17
Table 4.11: The content of Verification Result Description....17
Table 4.12: The content of Bug Information....18
Table 4.13: The content of User Manual....19
Table 4.14: The content of Requirement Management....20
Table 4.15: The content of Deliverable Table....20
Table 4.16: The content of Common Issue Management List....21
Table 4.17: The content of Issue Management List for each model....21
Table 4.18: The content of Confirmation List....21
Table 5.1: The explanation about rule of file name....22

Index of Figures

Figure 3.1: Model development flow....4
Figure 3.2: The procedure of Design Plan phase....5
Figure 3.3: The procedure of Internal Specification phase....7
Figure 3.4: The procedure of Design & Verification phase....8
Figure 3.5: The procedure of Final Review phase....9
Figure 4.1: Example of sequence diagram....14
Figure 4.2: Example of operation flow....15
Figure 4.3: Example of state diagram....15
Figure 5.1: The rule of file name....22

1. Introduction

To design a model efficiency, both RVC and REL need to define the model development flow and deliverable clearly. This document describes all phases in model development flow and their procedure. All projects must take them as strict rules during model development.

2. Term definition

Table 2.1: Term Definition

No.	Term name	Explanation
1	DR	The design review.
2	DMS	The website to share documents between RVC and MC-Soft.
3	Redmine	The website to discuss the issue and manage the project status.
4	Requester	A person who requests to develop a model (MC-Soft site).
5	Designer	A person who develops a target model (RVC site).
6	HW designer	A person who answers unclear points related to hardware manual of target model.

3. Model development flow

The model development flow includes 4 phases as figure 3.1. The detail of these phases is described in chapters 3.1, 3.2, 3.3, 3.4

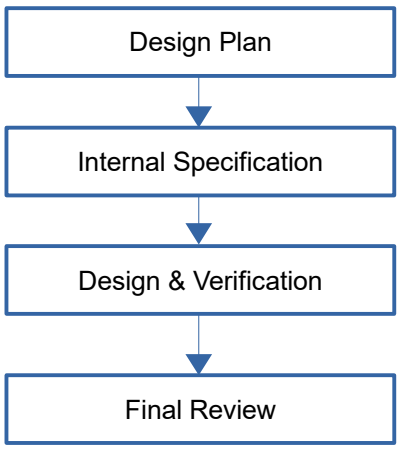

Figure 3.1: Model development flow

3.1. Design plan

Detail procedure of Design Plan phase is illustrated as figure 3.2

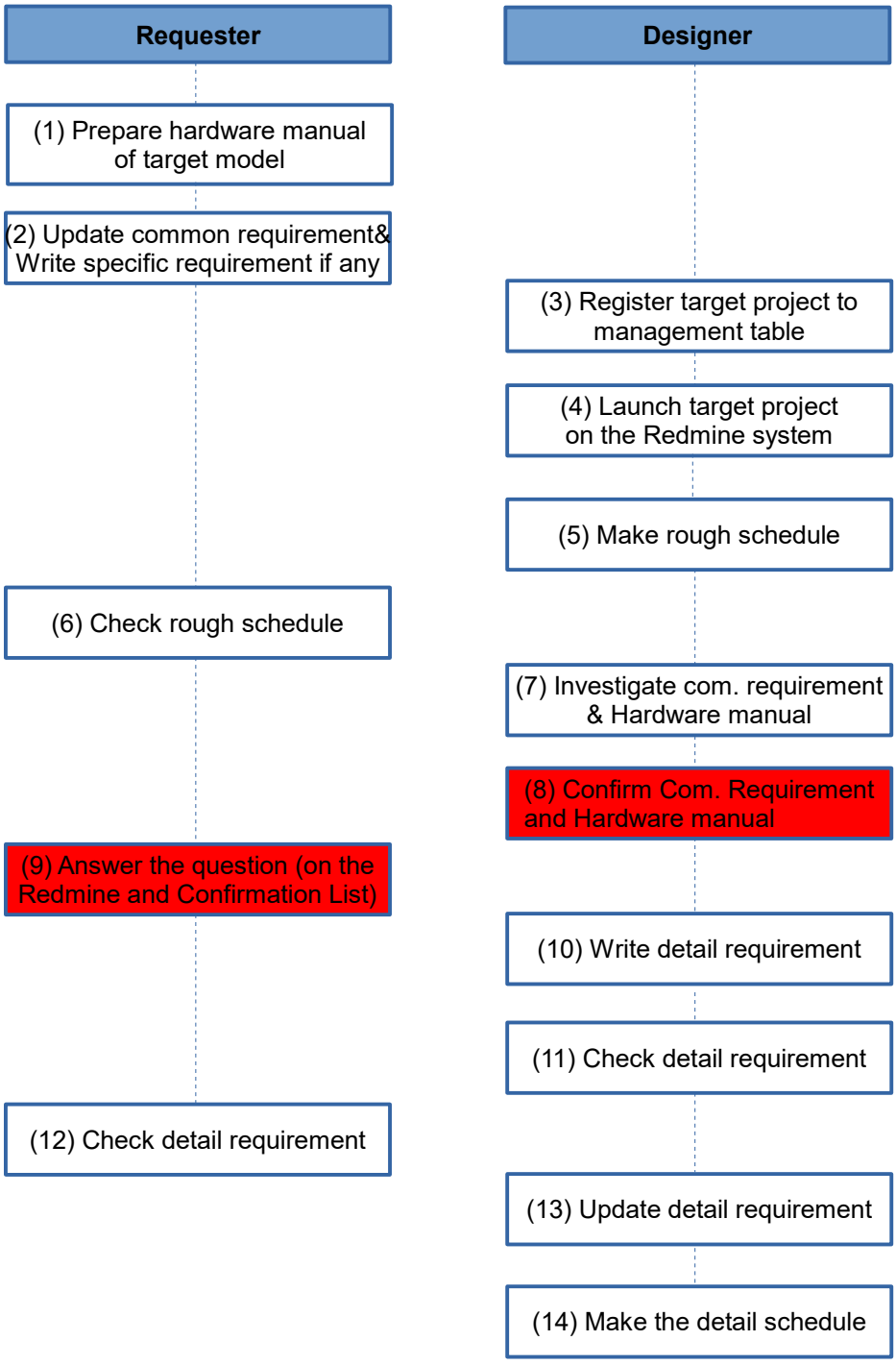

Figure 3.2: The procedure of Design Plan phase

- Step 1, 2: The requester prepares the hardware manual. Besides, if there is additional information about target model, the requester writes them in the specific requirement or update the common requirement.
- Step 3: The designer registers new project for target model to Management Table (defined

in section 4.1) by fulfilling all information such as project name, project ID, person in charge, started date.

- Step 4: The designer launches the project on the Redmine system and both the requester and the designer can use it to exchange question/issue/idea during the project.
- Step 5, 6: The rough schedule must be estimated in these steps. So the designer makes rough schedule and the requester checks the rough schedule. If there is any changing points, the designer must update the rough schedule for the requester approval.
- Step 7, 8, 9: The designer investigates target model specification (it includes the hardware manual, the common requirement if any). The designer's understanding (such as feature list, difference between the hardware manual and the model, unclear points) is described in Confirmation List file (defined in section 4.10). In these steps, the designer and requester exchange the question/issue/idea related to target model though the Redmine system and Confirmation List. If the HW designer's confirmation is required, the requester will send the question to the HW designer.
- Step 10: The designer writes the Detail Requirement (defined in section 4.2) to the requester.
- Step 11, 12, 13: The Detail Requirement will be checked at the designer side before sending it to the requester through the DMS system. The Detail Requirement must be checked by the requester and if there is any question/comment, the requester and designer can exchange them through the Redmine system. After the designer receives the comment from the requester, the designer must update the Detail Requirement and send its final version to the requester again.
- Step 14: The designer makes the detail schedule and send it to the requester though DMS system.

Table 3.1: Expected output of Design Plan's deliverable

No.	Expected output		Author
	Document	Data	
1	Hardware manual	-	Requester
2	Updated common requirement if any	-	Requester
3	Specific requirement if any	-	Requester
4	Confirmation List (defined in section 4.10)	-	Designer
5	Detail Requirement (defined in section 4.2)	-	Designer
6	Traceability Table (defined in section 4.7)	-	Designer
7	Deliverable Table (defined in section 4.8)	-	Designer
8	Round Schedule + Detail Schedule	-	Designer

3.2. Internal specification

Detail procedure of Internal Specification phase is illustrated as figure 3.3

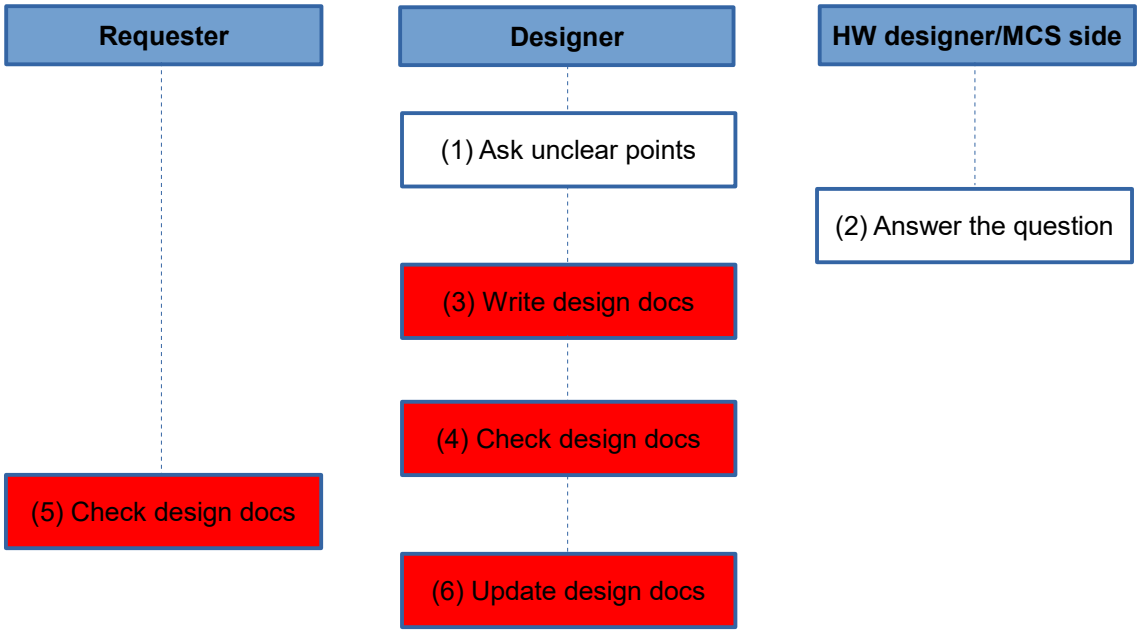

Figure 3.3: The procedure of Internal Specification phase

- Step 1, 2: The designer continues to ask questions about unclear points of target model and the requester answers questions through Redmine system (In some cases, the requester will get answers from the HW designer side). It recommended that all of unclear points should be cleared in advance.
- Step 3, 4, 5, 6: The designer writes design documents (they includes Internal Specification (defined in section 4.3), Verification Specification (defined in section 4.4), Checklist (defined in section 4.4.2)) to describe how to design/verify target model in detail. Besides, Issue Management List (defined in section 4.9) need to be created in these steps by the designer. All of document will be checked at the designer side before they are sent to the requester. The requester also check them and give the comment/question/idea through Redmine system. After the designer receives the comment from the requester, the designer must update the documents and send their final version to the requester again.

Table 3.2: Expected output of Internal Specification's deliverable

No.	Expected output		Author
	Document	Data	
1	Internal Specification (defined in section 4.3)	-	Designer
2	Verification Specification (defined in section 4.4)	-	Designer
3	Checklist (without bug report - defined in section 4.4.2)	-	Designer
4	Issue Management List (defined in section 20) – Note: This document is only used in	-	Designer

	the designer side.		
5	Traceability Table (defined in section 4.7)	-	Designer
6	Deliverable Table (defined in section 4.8)	-	Designer

3.3. Design & Verification

Detail procedure of Design & Verification phase is illustrated as figure 3.4

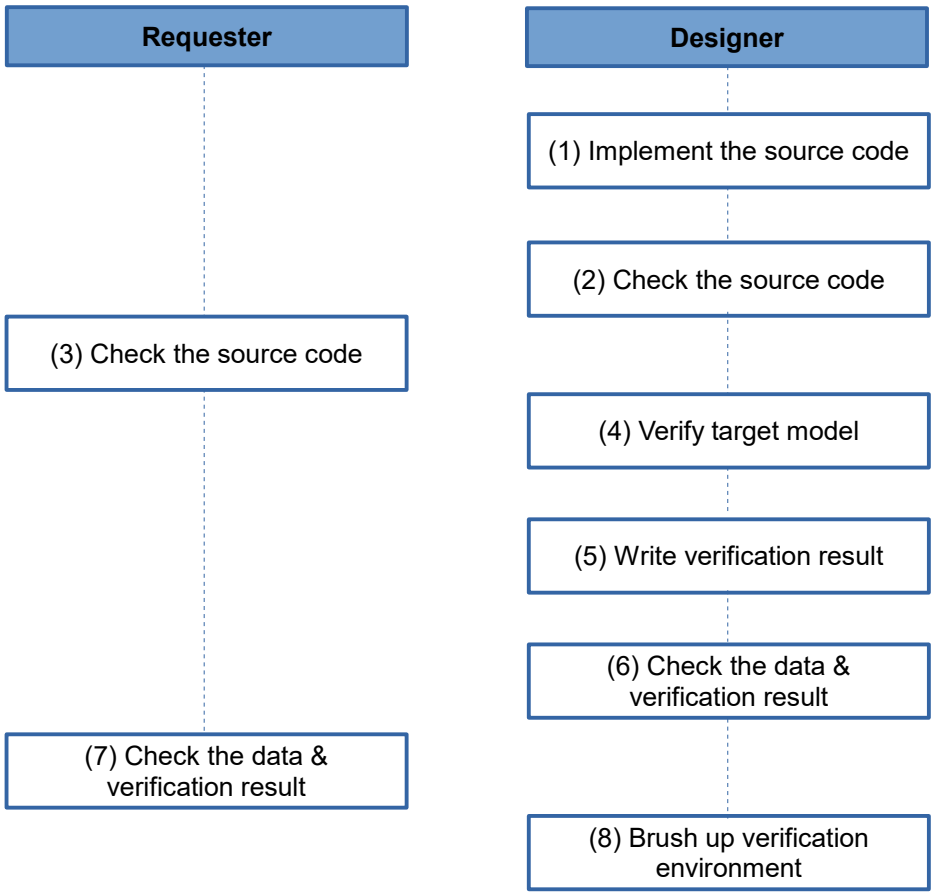

Figure 3.4: The procedure of Design & Verification phase

– Step 1: The designer implements the source code of target model based on the Internal Specification and checks syntax and style of source code by using compiler and 1Team system.
– Step 2, 3: Before verifying target model, the designer needs to request code review to other designer and the requester. All comments or bugs are recorded in Redmine system.
– Step 4: During checking source code by other designer and the requester, the designer starts to verify the source code of target model. All items in Issue Management List must be checked during this phase. The designer must keep bug list, checklist daily basis. To share development status between the designer and the requester, the designer must upload these documents (the schedule, the checklist) on the DMS system weekly.
– Step 5, 6, 7: After verifying target model, the designer writes Verification Result (defined in section 4.5). The data and Verification Result must be confirmed at the designer side before sending them to the requester. The requester also check them and give the

comment/question/idea through Redmine system. All comments are recorded in Redmine system.

– Step 8: After receiving comments from the requester, the designer must update the data and Verification Result and sends them to the requester again.

Table 3.3: Expected output of Design & Verification's deliverable

No.	Expected output		Author
	Document	Data	
1	-	Source code	Designer
2	Checklist (Included bug report – defined in section 4.4.2)	-	Designer
3	-	Release data (included test pattern)	Designer
4	Verification Result (defined in section 4.5)	-	Designer
5	Issue Management List (all items are checked – defined in section 4.9)	-	Designer
6	Traceability Table (defined in section 4.7)	-	Designer
7	Deliverable Table (defined in section 4.8)	-	Designer

3.4. Final review

Detail procedure of Final Review phase is illustrated as figure 3.5

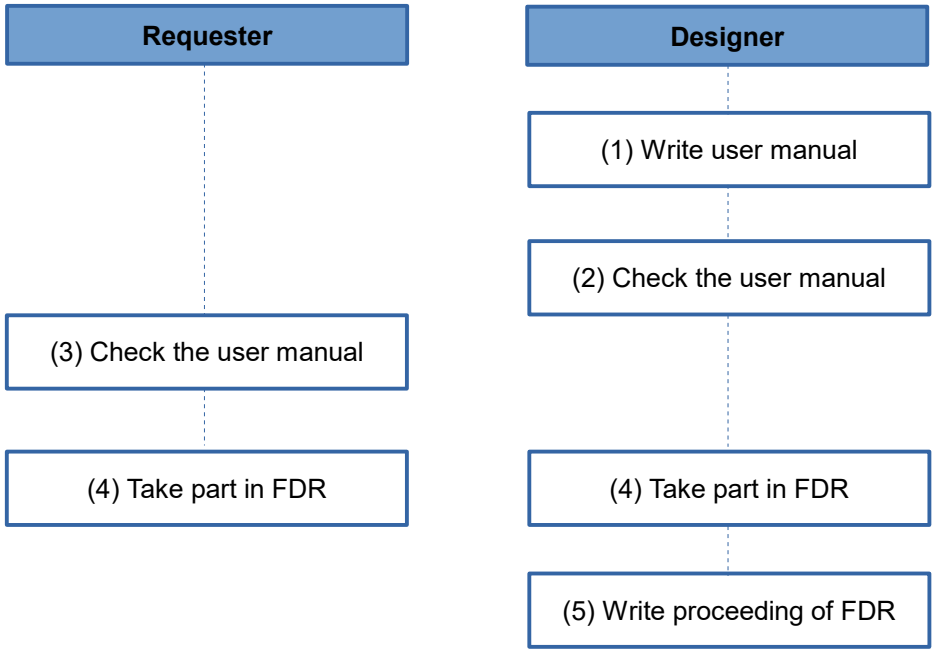

Figure 3.5: The procedure of Final Review phase

– Step 1, 2, 3: The designer writes User Manual (defined in section 4.6) to explain how to use target model to users. This document is checked at the designer side before sending it to the requester. The requester must check it and give comments on Redmine system.
– Step 4, 5: If necessity, both the requester and designer should take part a meeting (FDR) to discuss the problem occurred during model development. After that, designer writes proceeding for the FDR.

Table 3.4: Expected output of Final Review's deliverable

No.	Expected output		Author
	Document	Data	
1	User Manual (defined in section 4.6)	-	Designer
2	Traceability Table (defined in section 4.7)	-	Designer
3	Deliverable Table (defined in section 4.8)	-	Designer
4	Meeting minute of FDR if any	-	Designer

4. Content of deliverable documents

As MCS development flow is explained in chapter 3, there are some deliverable documents according to each phase as below. Thus, main content of these documents should be defined in this document.

- Management table
- Detail requirement
- Internal specification
- Verification specification
- Verification result
- User manual
- Traceability table
- Deliverable table
- Confirmation list
- Issue management list

4.1. Management table

Management Table is for managing the serial number of each project. After the designer receives the hardware manual from the requester, the designer writes requirement information into this document and gets serial number from it. Even if a project is canceled, reserved serial number is not be reused. When a project is done, the designer writes "Finishes day" into it.

Table 4.1: The content of management table

Item	Contents
Gr.Name	Write group name. This keyword is 3 character.
A.D	The last two digits of the year.

Item	Contents
No.	3 digits serial number. It starts from 001. Even if the model development project is canceled, keep that number. (new project cannot use registered number).
Title	Title of development.
Entries day	The day when the designer writes project information to Management Table and gets the serial number.
Finishes day	The day when target project is finished.
Requester	Name of requester.
Designer	Name of designer.

Management table example is as following:

Table 4.2: The example of management table

Gr Name	A.D.	No.	Title	Entry day	Finish day	MCS	RVC		
						Contact Person	Contact Person	Member	
SLD	14	001	DNF	04/21/14	07/07/14	Arai-san	DuyMai	NganTran	-
SLD	14	002	FCLA	04/21/14	07/07/14	Arai-san	DungNguyen	BinhNguyen	-
SLD	14	003	ECCCMNU	04/21/14	07/07/14	Arai-san	DungNguyen	NganTran	SonVu

4.2. Detail requirement

The designer writes Detail Requirement based on the hardware manual. This document contains important information such as features of the model, difference between target model and hardware, limitation of the model regarding the hardware manual. Thus, the content of this document includes the items in table 4.3

Table 4.3: The content of Detail Requirement

Item	Comment
Project goal	Write the purpose for developing target model.
Model overview	Write summary of target model to share the image of target model before detailed explanation.
Feature list	Write feature list of target model. If feature is not supported in target model, also write the reason.
Register	Write register lit of target model. If register is not supported in target model, also write the reason
Input and output ports	Compare communication I/F between hardware manual and target model. If port is not supported in target model, also write the reason
Functions	Explain main function of target model. It also describes timing point if need.
Commands and Parameters	If target model needs to support parameters and/or commands, write their specification. Execution method (Python IF command or Heap configuration) should be

Item	Comment
	indicated.
Implement requirement	Describe the requirement of model implementation (such as unit of clock, synchronous/asynchronous reset, message format)
Related documents	Describe reference documents which is used for model implementation
Common model and Verification ENV	Describe common model which is used for model implementation. Besides, also describe the condition to verity the designed model.
Limitation	Describe limitation of model
Development schedule	-
Appendix	If there is additional information, also describe it (Ex: Timing format, relationship between HW reset (port reset) and Software reset (AssertReset command))

4.3. Internal specification

This document is written before coding phase. The designer designs target model on this document. Thus it is written in detail. This document includes the below items.

- Model summary
- Supported features
- Block diagram
- List of implemented registers
- List of implemented ports
- Direction for user
- Flow diagram
- Functions description
- Limitation
- Related document

4.3.1. Model summary

The designer writes model summary. For example, the purpose of the model is developed, product name which the model belongs. It is better to write the advantage to use this model.

4.3.2. Supported features

The feature list is written based on module' feature in the hardware manual. It describes that whether each feature is support or not.

The content of supported feature list includes the below items.

Table 4.4: The content of Supported Features

Item	Comment
Feature name	The name of feature of target model.
Explanation	Explanation of target model features compared with HWM.
Support	Yes or No. If target model supports both untimed and timed, this

	item should be separated.

4.3.3.Block diagram

The designer writes the model's block diagram with related other models to understand the position of target model in the system. Each block is explained at the bottom of block diagram. And fill in areas of each block which means the each model. The color of each block should mean the group (For example, target model, another model, ...). The block of target model had better include the major threads.

4.3.4.List of implemented registers

The register list is written based on module' register in the hardware manual. It describes that whether each bit in the register of module is support or not. The content of register list is as below

Table 4.5: The content of Register List

Item	Comment
Name	The name of the model register compared with HWM
Chapter	Chapter in hardware manual included register's information
Address	Offset address of the register
Initial	Initial value of the register
Bit name	It describes each bit of a register of target model
Access mode	Read access only, write access only, both access
Description	Brief description about function of the register or each bit in register
Support	Yes or No

4.3.5.Port behavior

The designer writes target model's port list and their behavior in this section. Port list includes items in table 4.6.

Table 4.6: The content of Port List

Item	Comment
Hardware	The port name is described as in the hardware manual.
Model	The port name in the hardware manual is modeled in target model.
I/O	Input port or output port.
Type	The data type of the port (bool, unsigned int, ...).
Initial	Initial value of the port.
Active	Active level of the port (high, low).
Description	Brief description of the port.
Support	Yes or No.

4.3.6.Direction for user

The designer explains how to use target model. It includes below items.

Table 4.7: The content of Direction For User

Item	Comment
File structure	Write all files composed target model with explanation.
Input and Output file	If there is any input file to target model or output file from target model, write them with explanation. The constructor's argument should be written too.
How to connect verification environment	Write the method and flow to connect target model to verification environment.
Commands and parameters	Write commands and parameters with explanation.
Defined macro and template	If target model has defined macro or template, write that name with explanation.
Error and debugging message	If target model puts message during simulation, write them.

4.3.7.Flow diagram

The designer describes detail operation of target model. It includes below items

Table 4.8: The content of Flow Diagram

Item	Comment
Sequence diagram	Describe interaction between target model and others.
State diagram	Describe internal states of model and transition of these states.
Operation flow	Describe each function operation of target model in detail.

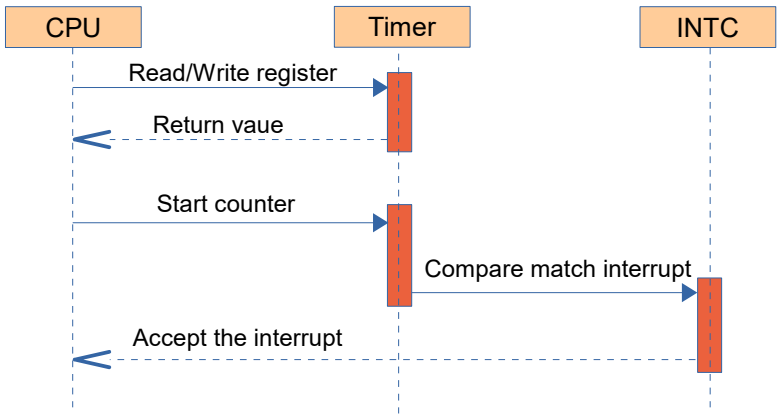

Figure 4.1: Example of sequence diagram

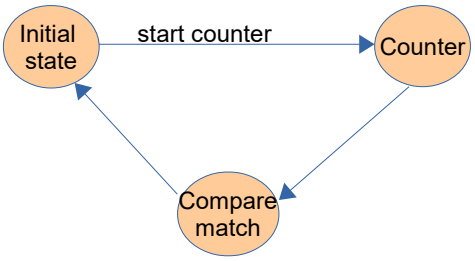

Figure 4.3: Example of state diagram

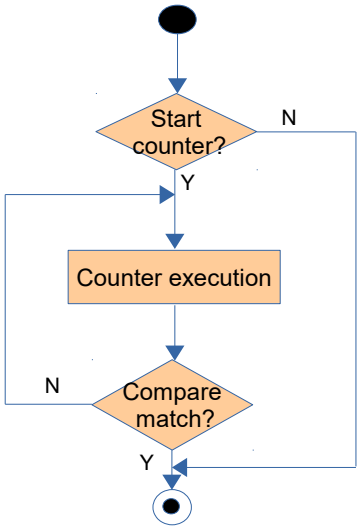

Figure 4.2: Example of operation flow

4.3.8.Function description

The designer describes all functions with explanation of each class.

4.3.9.Limitation

The designer describes limitation of target model if any.

4.3.10.Related document

Designer writes related document to design and understand target model. Hardware manual and Standard bus specification are one of them.

4.4. Verification specification

This document is written before verifying target model. Designer verifies target model based on this document. Thus it should be written in detail. This document should include the below items.

- Verification description
- Checklist

4.4.1.Verification description

This is Verification Specification document. In this document, the designer describes how to verify target model and It should include below items as table 4.9

Table 4.9: The content of Verification Description

Item	Comment
Introduction	Introduction of this document. Write the summary and purpose of it.

<table>
<tr><td>Block chart</td><td>Verification environment block chart. From this chart, reader gets the following information.
• The models composed this chart and those types (target mode, RTL or other model).
• Input and Output
• How to verify</td></tr>
<tr><td>Dummy model</td><td>Write dummy model information which communicates with the target model.</td></tr>
<tr><td>Verification environment</td><td>Write environment structure, how to execute the environment, how to confirm simulation result. Recommend that simulation result must be checked automatically. If there is any item which is checked by eyes, write them in detail.</td></tr>
<tr><td>Verification condition</td><td>Write the verification conditions. For a example:
<table>
<tr><th>Group</th><th>Condition</th><th>Content</th></tr>
<tr><td>Machine</td><td>EWS</td><td>Red Hat Enterprise 5 (32bit)</td></tr>
<tr><td></td><td>Windows</td><td>Windows-7 (32bit)</td></tr>
<tr><td>Simulation environment</td><td>The name of verification environment</td><td>SC-HEAP E3</td></tr>
<tr><td>Library</td><td>SystemC</td><td>OSCI SystemC v2.3</td></tr>
<tr><td></td><td>TLM</td><td>VWorks OSCAR (USK) v2.1.0</td></tr>
<tr><td>Tool</td><td>Compiler</td><td>GNU G++ v4.1.2</td></tr>
<tr><td></td><td></td><td>Microsoft Visual C++ 2010</td></tr>
<tr><td></td><td>gcov</td><td>v4.1.2</td></tr>
<tr><td></td><td></td><td>dummyins v1.01</td></tr>
<tr><td></td><td>1Team</td><td>v1.16.5</td></tr>
<tr><td></td><td>Valgrind</td><td>v3.7.0</td></tr>
<tr><td></td><td>Python I/F</td><td>Python 2.7.3</td></tr>
<tr><td></td><td>Embedded software tool</td><td>GHS MULTI 6.1.4 rteserv2</td></tr>
</table></td></tr>
<tr><td>Verification requirement</td><td>Write the verification requirement. For a example:
<table>
<tr><th>Requirement</th><th>Target</th></tr>
<tr><td>Compile</td><td>No error and no warning</td></tr>
<tr><td>Code coverage</td><td>Line coverage is 100%</td></tr>
<tr><td>Functional coverage</td><td>100% on traceability table (defined in section 4.7)</td></tr>
<tr><td>Style check</td><td>Run 1TeamSystem with option template=Renesas/Modeling”</td></tr>
<tr><td>Memory check</td><td>No error and warning bout target source code</td></tr>
</table></td></tr>
</table>

4.4.2.Checklist

The checklist describes all test items to verify target model in detail. In general, it is a very big

table, and it is not necessary to write the checklist in Verification Specification file. The designer can write it in another file. The content of checklist should include sheets as table 4.10

Table 4.10: The content of Checklist

<table>
<tr><th>Sheet</th><th>Explanation</th></tr>
<tr><td>INT</td><td>Describe relationship between chapter of Internal Specification and test items.</td></tr>
<tr><td>CFM</td><td>Describe relationship between each feature and HWM's chapter, Detail Requirement's chapter, Internal Specification's chapter and test items.</td></tr>
<tr><td>Register_RW
Operation
PythonIF_HeapConf</td><td>Each sheet should include items as below table:
<table>
<tr><th>Item</th><th>Comment</th></tr>
<tr><td>Category</td><td>Write each items categorized by some groups.</td></tr>
<tr><td>What to check</td><td>What the designer wants to check. For example, "copy source address's data to destination address.”</td></tr>
<tr><td>How to check</td><td>Detailed checking method. For example, write the register setting</td></tr>
<tr><td>Expected result</td><td>Write expected value, behavior message, and so on.</td></tr>
<tr><td>Test pattern name</td><td>-</td></tr>
<tr><td>Num of items</td><td>The number of checked items</td></tr>
<tr><td>By who</td><td>-</td></tr>
<tr><td>Schedule</td><td>Write finished date</td></tr>
<tr><td>Actual</td><td>Write finished date</td></tr>
<tr><td>Update History</td><td>The date when the designer adds/removes/changes the check item</td></tr>
<tr><td>Remark</td><td>-</td></tr>
</table></td></tr>
<tr><td>Message</td><td>Describe the model's message between the Checklist and Internal Specification.</td></tr>
<tr><td>Bug_record</td><td>Record bugs during model development period.</td></tr>
</table>

4.5. Verification result

This document is used to explain the verification result to the requester. Verification Result should include the following items

- Verification result description
- Checklist with bug information

4.5.1.Verification result description

The content of Verification Result Description is as table 4.11

Table 4.11: The content of Verification Result Description

Item	Comment
Functional verification	The summary of check list. For example, the number of total item, unchecked item and checked item. If the model works under some environment (ex. UNIX and Windows), these results should be separated.

Timing verification	Detail timing report. It should include timing diagram.
Code coverage	Code coverage percentage. If it cannot reach 100%, write that reason one by one.
Coding rule check	1TeamSystem result. If there are errors or warnings, write that reason.
Memory check	Valgrind result. If there are errors or warning, write that reason one by one.
Known Bug	If there are known bugs, write them in detail.

4.5.2.Checklist with bug information

The designer updates the checklist through the verification. Besides, bug information also include into the checklist as table 4.12 and Redmine system.

Table 4.12: The content of Bug Information

Item	Comment
Error-caused process	Please choose below items: • Requirement • Internal spec • Verification spec • Coding • Test environment • Test • User manual
Bug-detected process	Please choose below items: • Requirement • Internal spec • Verification spec • Coding • Test environment • Test • User manual
Bug cause	Please choose below items: • HW/SW It makes a mistake in the interface among. • SW It makes a mistake in the interface among. • Instruction use mistake • Logic mistake • Lapse of judgment • Judgment leakage • Data operation mistake • Data mistake • Declaration mistake • Specifications mistake • Correct
When/Who find	-
Status	Please choose following items:

Item	Comment
	• Fixed • Pending • Ignored
TM found	The test pattern that found the bug.
Symptom	Detail description of bug's symptom.
Cause & How to fix	Detail description of bug's cause and how to fix the bug.
When/Who fix	-
Updated Version	Source code version fixed the bug.
When/Who checked	-

4.6. User manual

This document describes “how to use target model”, users need to read it before using target model. Thus, the content should include items as table 4.13

Table 4.13: The content of User Manual

Item	Comment
File	File name with CVS version. If the model refers to common class, write file name and version number of them.
Register	Register list of target model (register name, initial value, offset address, each bit description).
Feature	Feature list of target model (feature name, feature description).
Port	Port list of target model (it should be compared with port name of hardware manual). These ports is explained with timing chart.
Parameters and Commands	If the target model needs to support parameters, write that specification in detail. Besides, configuration method should be also described (Python I/F and HEAP configuration)
Message	All messages of target model (It should include detail content and meaning).
Internal state	Internal operation states of target model.
Built option	Options which are declared when users compile target model.
Limitation	Limitation for using target model.
Reference documents	Description of reference documents

4.7. Traceability table

Before the designer sends a document to the requester, the designer ensures all items in Requirement Management are satisfied. Thus, requirement items should be defined in this document in advance. The content of Requirement Management is as table 4.14

Table 4.14: The content of Requirement Management

Item	Comment
Requirement	It should be separated into each group and write detail items in that group.
When/Who check	The date and the name of person in charge.
Documents with version	Write file name of document (with version)

4.8. Deliverable table

When the designer sends the data or the document to the requester, the information is recorded into this document. The content includes items as table 4.15.

Table 4.15: The content of Deliverable Table

Item	Comment
Deliverable name	This describes the name of deliverable items. Ex: Internal Specification, Verification Specification
File name	The file name of deliverable Ex: INT-MCS-12001-Guard.odt, VRF-MCS-12001-Guard.odt
Deliverable number	The ID of documents. It includes following information: + The kind of document as Internal Specification (abbreviation is INT), Verification specification (abbreviation is INT) + Group name: MCS + The ID of project (Year+XXX) : 12001, 12002 Ex: INT-MCS-12001-Guard, VRF-MCS-12001-Guard
Version	The version of deliverable (If deliverable is the data which is managed by the tool CVS, we writes CVS tag) Ex: ver.1.0, ver.1.2, ...
Delivery Date	The date which the deliverable is released Ex: 11/15/2012, ...

4.9. Issue management list

To ensure common issues do not occur frequently, we should record them in a document called Common Issue Management List. During model development, we must confirm them by create Issue Management List for each model. The content of Common Issue Management List as table 4.16 and Issue Management List for each model as table 4.17.

Table 4.16: The content of Common Issue Management List

Item	Comment

Bug list	Common bug must be described in this item, it should include following information: • Characteristic of model which the bug occurred • Bug's content/symptom • Which model detected the bug • Reason/how to fix the bug
Model bug status	The status of bug's fixing for each model should be described in this item.

Table 4.17: The content of Issue Management List for each model

Item	Comment
Bug list	Not only common bugs from “Common Issue Management List”), but also specific bugs of the model must be described in this item. It should include following information: • Bug's content/symptom • Reason/how to fix the bug • Who fixed/checked • Checking method • Workload
Document checking	The relationship between documents (REQ, INT, VRF, VRF-01 , etc.) is described in this item. The designer should refer to it to determine which part in a document need to update.

4.10. Confirmation list

When the designer investigate the hardware manual and common requirement (if any), the designer should collect all information for confirmation from the requester. The content includes items as table 4.18

Table 4.18: The content of Confirmation List

Item	Comment
Summary	Number and status of question should be described.
Features	Question about features of the model
Ports	Question about ports of the model
Registers	Question about registers of the model
Functions	Question about functions of the model
Commands_Parameters	Question about commands and parameters of the model
Others	Other question about the model

5. Management rule

This chapter defines some rules for managing the project.

5.1. Review and Approval

All documents must be uploaded to DMS system after they are reviewed and approval by manager.

5.2. Document name

Document name has simple rule, all documents must follow it. A document is composed by 5 items as below:

XXX-YYY-ZZVVV-D***.doc

(Ex: REQ-MCS-12001-Guard.doc, INT-MCS-12001-Guard.doc, ...)

Figure 5.1: The rule of file name

Table 5.1: The explanation about rule of file name

<table>
<tr><th>Items</th><th>Comment</th><th>Example</th></tr>
<tr><td>XXX</td><td>Reserved keyword. It is indicated as below table
<table>
<tr><th>Document</th><th>Reserved Keyword</th></tr>
<tr><td>Detail Requirement</td><td>REQ</td></tr>
<tr><td>Internal Specification</td><td>INT</td></tr>
<tr><td>Verification Specification</td><td>VRF</td></tr>
<tr><td>Checklist</td><td>CHK</td></tr>
<tr><td>Verification Result</td><td>INR</td></tr>
<tr><td>User Manual</td><td>USR</td></tr>
<tr><td>Traceability table</td><td>TRA</td></tr>
<tr><td>Deliverable Table</td><td>DEV</td></tr>
<tr><td>Issue Management Table</td><td>ISM</td></tr>
<tr><td>Confirmation List</td><td>CFM</td></tr>
</table></td><td rowspan="5">- REQ-MCS-12001-Guard.ods
- INT-MCS-12001-Guard.odt
- VRF-MCS-12001-Guard.odt
- CHK-MCS-12001-Guard.ods
- INR-MCS-12001-Guard.odt
- USR-MCS-12001-Guard.ods
- TRA-MCS-12001-Guard.ods
- DEV-MCS-12001-Guard.ods
- ISM-MCS-12001-Guard.ods
- CFM-MCS-12001-Guard.ods</td></tr>
<tr><td>YYY</td><td>Group name. It should be MCS (MC-Soft)</td></tr>
<tr><td>ZZ</td><td>A.D (The last two digits of the year). Ex: “12“ - year 2012</td></tr>
<tr><td>VVV</td><td>Serial number of project. Ex: 001, 002, 003, ...</td></tr>
<tr><td>D***</td><td>Character of document such model name, ...(it can is omitted if need)</td></tr>
</table>

Revision History				
Rev.	**Modified Contents**	**Approval**	**Reviewed by**	**Created by**
1.0	New creation		ThaiNguyen 11/19/2012	DungNguyen 11/15/2012
1.1	- Added “rough schedule” in step 5 of section 3.1 - Added “hardware designer role” in figure 3.3 - Changed “Requirement Management” to “Traceability Table” in section 4 - Added “Communication I/F” in table 4.3 - Deleted “Class explanation” in section 4.3 - Added “Parameter, Python I/F, HEAP configuration” in section 4.3 - Added “Parameter, Python I/F, HEAP configuration” in section 4.6 - Updated Verification requirement and Verification condition in table 4.9		ThaiNguyen 12/04/2012	DungNguyen 12/04/2012
1.2	- Removed “(4) Write reception verification” in Final Review phase (section 3.4) - Added Confirmation List in Design Plan phase (section 3.1) - Added Issue Management List in Internal Specification phase (section 3.2) and Design & Verification phase (section 3.3) - Updated the content of Detail Requirement (section 4.2) - Updated the content of Internal Specification (section 4.3) - Updated the content of Verification Specification (section 4.4): + Added “How to confirm simulation result” step in table 4.9. + Added sheets INT, CFM, Message, Bug_record for the Checklist. - Updated the content of User Manual (section 4.6) + Changed “Communication I/F” item to “Port” item + Changed “Paramter, Python I/F, HEAP configuration” item to “Parameters and Commands” item + Added “Reference documents” item - Added “Issue management” section 4.9 - Added “Confirmation list” section 4.10	-	-	DungNguyen 07/07/2014