## NuSRS 2.0 User Manual (ver.1.1)

http://dependable.kaist.ac.kr/~nusrs nusrs@dependable.kaist.ac.kr

#### Dependable Software lab. KAIST 한국과학기술원

Software Engineering Lab. Deptpartment of Computer Science



#### Contents

- NuSRS 2.0 Introduction
- NuSRS 2.0 Structural Overview
- NuSRS 2.0 Constructs
- Detailed description of NuSRS 2.0
  - Menu functions, Toolbar
  - Hierarchy Window
  - Description Window
  - FOD, FSM(TTS), SDT Edit Window
  - Type Window
  - Quick Check
  - Model Checking



### **NuSRS 2.0 Introduction**

- Automatic tool to support requirement specification and formal verification of NuSCR model
- Development language: Java
- Development environment: eclipse 3.1 (JDK 1.5)
- Environment
  - Windows OS(recommendation)
  - PC with JDK or JRE (ver. 1.5 or higher) installed







#### **NuSRS2.0 Constructs**



### Menu Functions, Toolbar





KAIS

#### Menu > File

#### ♦ File

- New Create new file
- Open Open file
- Close File Close file
- Save Save file
- Save As Save file as another name
- Save Without Graphical Info

Save file without graphical information

- Save as Image
  - Save file as image
- Print Print
- Exit Exit program





#### Menu > Edit

#### ♦ Edit

- Cut Cut Selected
- Copy Copy Selected
- Paste Paste Selected
- Delete Delete Selected
- SelectAll Select All nodes





#### Menu > View

#### View

- FIND Find node
- View IO Vars View list of IO Variables
- Zoom In Zoom In
- Zoom Out Zoom Out
- Zoom Selected Zoom Selected
- Fit to Window Zoom in or out depending on window size
- Auto Align Auto Align





#### Menu > Window

#### Window

- Close Close the editing Window
- Close All Windows Close all Windows





#### Menu > Help

#### ♦ Help

About – Give information on NuSRS





#### **Create new file (1/2)**

#### 1. Select File > New in Menu.

File Edit View Window Verifi	
🕞 Upen	
Save	
🛃 Save As	
🐻 Save Without Graphical Info	
🐻 Save as Image	-
🤪 Print	-
📲 Exit	
Description Window	
	Type Window 🖉 Console Window



### **Create new file (2/2)**

2. Under the hierarchy, the root is created inside the description window, and a window for drawing the root node is created

NuSRS Editor 2.0	
File Edit View Window Verification	Help
🞦 🗳 🛃 😼 🐤 🗎	
Hierarchy Window	O Root ×
- 🗋 Root	N
	0
Description Window	
P- 📑 Root	
P- ☐ Description	$\rightarrow$
e 📑 TemplateNumber	
- 🗅 Input	
– 🖸 Output	
Constant Constant Constant	🙄 Type Window 🧧 Console Window
–  Previous State Variable	
🗕 🗋 Mernoriazble Variable of Exter	



### Load file (1/3)

- 1. Select File > Open in Menu.
  - Open files that were created in NuSRS with .xml suffix

NuSRS Editor 2.0	т <sup>*</sup> 2 <sup>°</sup>
File       Eatt View Window Vermication         New       Image: Solution in the solutine solution in the solution in the solution	
Exit     Description Window	
	Type Window Console Window



### Load file (2/3)

KAIS

2. When the dialog box opens, select the file to load.

NuSRS Editor 2.0 - D:/LAB PR	OJECT'NuSRS - 합친거ttest.xml	<b>□</b> 🖸
File Edit View Window Ve	ification Help	
	• 🗎 📕 🗔 💌 🔍 🔍 🖾 🗖 💌 💌	
📲 Hierarchy Window		
	📓 open XML File 🔀	
	검색 위치: 📑 NuSRS - 합친거 🔹 🖬 🔂 🗂 🙁 🕒	
	🗋 d.xml	
	[] dd.xml	
	test.xml	
	nal-verification_0044_44_56 vml	
	nal Verification 1210 18 0/ vml	
📃 Desctiprion Window		
	파일 이름: test.xml	
	파일 종류: xml 💌	
	열기 취소	



#### **Save file**

1. Select File > Save in Menu. The current file is then saved.





#### Save As (1/2)

#### 1. Select File > Save As in Menu.





#### Save As (2/2)

2. When the dialog box opens, input a file name and click save. The current file is saved.

NuSRS Editor 2.0 - D:\LAB F	ROJECTWuSRS - 합친거\test.xml 📭 🔽 🔀	
File Edit View Window V	ferification Help	
2 🗳 🗐 🚳	* 🖹 🗷 🗔 🗟 < 🍳 🔍 🔯 🧵 💌 🚬	
Hierarchy Window		
P- ☐ Root	🛓 Save XML File 🔲	
y_aut	제장위치 🗂 NuSRS-1 🔻 🖬 🔂 🗂 🔀 🔚	
	L data h_1	
	Test.xml	
	h_3	Save as
🔄 Description Window		Cave as
P- 📑 Root		Toot1 ym
🕈 📑 Description	파일 종류: NuEditor XML data file (*.xml)	Iesti.xiii
	서상(S) 쥐소(C)	
🗣 🚍 Input		
– 🗋 f_1 : boolean		
- C f_2 : boolean		-
- 1 1_3 : poolean	f_1: boolean	
	↓ 1_17: boolean ↓ f_2: hoolean	
		10
		19

### **Save without Graphical Info (1/2)**

 Select File > Save Without Graphical Info in Menu.





### Save without Graphical Info (2/2)

2. When the dialog box opens, input a new file name and click save. The current file is saved without graphical information.

NuSRS Editor 2.0 - D:\L/	AB PROJECTINUSRS - 합친거test.xml 🗗 🗹 🛛	
File Edit View Window	Verification Help	
睯 New 選 Open		
🔒 Close File		
📙 Save		
🛃 Save As		
🐻 Save Without Graphic	🔹 Save XML File without graphical information 🛛 🔀	
🐻 Save as Image		
🍰 Print	제장위치 CINUSRS-1    제 슈 너 많이는	
📲 Exit	h_1	
	nuscrUl h_3	
Description Mindow		Save as
	Test1.xml	
P □ Root A □ Description		ucor
	<u>f_5</u>	usei
🔶 📑 TemplateNurub	ILU UIE: Test3-NoGraphicsInfo.xm	
⊷ ⊂⊐ Input	파일 종류: NuEditor XML data file (*.xml) 🗸 📃 🔽	assigned
f_1 : boolear		<b>J</b>
— 🗋 f_2 : boolear	저장(S) 취소(C) 📃 💷	name yml
— 🗋 f_3 : booldar		name
	The Astronom	
		<b>A</b> 4
		21

#### Save as Image (1/2)

#### 1. Select File > Save as Image in Menu.





#### Save as Image (2/2)

KALE

2. When the dialog box opens, input a new file name and click save. The current file is saved as an image file.

🗂 NuSRS Editor 2.0 - D:LAB PROJECTWUSRS - 합친거ttest.xml	a
File Edit View Window Verification Help	
Hierarchy Window	
P 🗇 Root Save current diagram to image file	
제장위치 [☐ NuSRS - 합친기	
i library	
<u>f_17</u>	
h_3	Save as
E Description Window	
	user
	assigned
●	accigited
	name.jpg
- D f_1 : boolean - D f_2 : boolean	
f_3: boolean f_1: boolean	
r_4: boolean f_17: boolean f 2: boolean f 2: boolean f 2: boolean	

#### Print

#### 1. Select File > Print in Menu.





### Edit > Cut (1/2)

1. Select the area to be cut by clicking the left mouse button or by drag & drop.

File Edit View Window Verification Help
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $
Hierarchy Window     Root     g_abc     f_1     h_1     f_2     f_17
$f_3$ $g_{abc}$ $h_3$
Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints         Image: Secretarian constraints       Image: Secretarian constraints       Image: Secretarian constraints
Console Window Console Window
T 17: boolean
<pre></pre>



### Edit > Cut (2/2)

2. Select Edit > Cut (or press Ctrl + X). The selected area is then cut.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSR	S - 합친거test.xml	r 🛛 🖂
File Edit View Window Verification He	lp	
Cut 🛃 😽 🗎		
📲 🛅 Paste 🛛 🗤 🔽	O Root ×	
P C SelectAll		-
	$f_3$ $g_abc$ $h_3$	
📄 Description Window		
P-□ g_abc		
← 🗋 ┍─ 🗂 TemplateNumber 📰	$\rightarrow$	
← ☐ Input	•	•
- 🗋 [_] : boolean - 🗋 f_2 : boolean	🔀 Type Window 🖉 Console Window	ſ
– 🖸 f_3 : boolean	f_1 : boolean	4
L L 12 f_4 : boolean	f_17: boolean	=
	[	-



### Edit > Copy (1/2)

1. Select the area to be copied by clicking the left mouse button or by drag & drop.

NuSRS Editor 2.0 - D:LAB PROJECT/NuSRS	- 합친거test.xml ㅁ 껍	×
	, 1 🗷 🗔 😪 🔍 🔍 🔯 亘 💌 🖻 🗶	
Hierarchy Window	O Root ×	
P C Root	$\begin{array}{c} \hline \\ \hline $	
		•
- G f_2 : boolean	Type Window Type Window	
☐ f_3: boolean	f_1: boolean	
	r_1/:poolean f 2:boolean	
		•



### Edit > Copy (2/2)

# 2. Select Edit > Copy (or press Ctrl + C). The selected area is then copied.





#### **Edit > Paste (1/4)**

#### 1. Select Edit > Paste. (or press Ctrl + V)





### Edit > Paste (2/4)

KAIS

2. Selected constructs are pasted with different names.

NuSRS Editor 2.0 - D:/LAB PROJECT/NuSR	S - 합친거ttest.xml	ī 🖂
File Edit View Window Verification He	lp	
2 🖓 🗦 🖏 💊 🗎 🕻		
Hierarchy Window		
<pre>P-□ Root     □ g_abc</pre>	$\begin{array}{c} \hline \\ \hline $	
		-
f_1 : boolean		•
- 🗋 f_2 : boolean	🙄 Type Window 🗧 Console Window	
– 🗋 f_3 : boolean	f_1: boolean	<b></b>
f_4 : boolean	f_17 : boolean	
	f_17_: boolean	-

### Edit > Paste (3/4)

3. Select the pasted node, click the right mouse button and select rename.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	- 합친거test.xml	a' 🛛
File Edit View Window Verification Help		
2 6 6 6 6		
Hierarchy Window	O Root ×	
ዮ 급 Root └ Ŋ g_abc	$\begin{array}{c} \hline \\ \hline $	
	$\begin{array}{c} \hline \\ \hline $	
P- 🚍 Input	4	•
– 🗋 f_1 : boolean – 🗋 f_2 : boolean	🙄 Type Window 🧧 Console Window	
- i f_3: boolean	f_1 : boolean	
f_4:boolean	f_17 : boolean	
	f_17_; boolean	-



### Edit > Paste (4/4)

4. Change the name of the pasted node appropriately.

NuSRS Editor 2.0 - D:'LAB PROJECT'NuSRS	- 합친거itest.xml	' 🛛 🖂
File Edit View Window Verification Hel	p	
2 🗳 🛃 😼 🔓 👔		
Hierarchy Window		
Υ C Root C g_abc	$\begin{array}{c} \hline \\ \hline $	
	$f_4$ $h_5$ $f_7$	, , ,
L T_1 : boolean	🕞 Type Window 🗧 Console Window	G
f_3: boolean     f_4: boolean     f_4: boolean     f_4: boolean	f_1: boolean f_17: boolean f_17: boolean f_17_: boolean	



### Edit > Delete (1/3)

1. Select the area to be deleted by clicking the left mouse button or by drag & drop.

🗂 NuSRS Editor 2.0 - D:LAB PROJECT/NuSRS - 합친거/test.xml		
File Edit View Window Verification Hel	p	
2 🖗 🛃 🐐 🗎 🕻		
Hierarchy Window		
P- ☐ Root ☐ g_abc	$\begin{array}{c} \hline \\ \hline $	
Description Window      G g_abc      G Description      G TemplateNumber      G Input		
- n f 1 : boolean		
- n f 2 : boolean	🙄 Type Window 🖉 Console Window	
- n f 3 : boolean	f 1: hoolean	
h f 4; boolean	f 17 : hoolean	
- Output	f 2 hoolean	



#### Edit > Delete (2/3)

#### 2. Select Edit > Delete. (or press Delete key)



#### Edit > Delete (3/3)

#### 3. Selection is deleted.





#### Edit > Select All (1/2) Select Edit > Select All. (or press Ctrl + A) 1. r 🛛 🖂 👕 NuSRS Editor 2.0 - D:/LAB PROJECT/NuSRS - 합친거/test.xml File Edit View Window Verification Help 👆 Cut Ê 📉 📲 🔍 🔍 -× Q E 8 📄 Copy ○ Root × 📋 Paste 样 Delete . R. SelectAll f\_1 Ο f\_2 $\bigcirc$ g\_abc h 3 f\_3 f\_4 📰 Description Window . • □ g\_abc $\bigcirc$ 🔶 🚞 Description - 🗅 📹 TemplateNumber $\rightarrow$ LΠ - 🥅 Input • Þ 🗋 f\_1 : boolean 🔁 Type Window 🧰 Console Window 🗋 f 2 : boolean 🗋 f\_3 : boolean . f 1:boolean 🗋 f\_4 : boolean f\_2 : boolean 🔶 📺 Output f\_3:boolean KAF 4 Ŧ 36
#### Edit > Select All (2/2)

KAE

#### 2. All nodes in the current view are selected.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	- 합친거ttest.xml	⊏ ⊠ ⊻
File Edit View Window Verification Hel	p	
Hierarchy Window		
P C Root	$\begin{array}{c} \hline \\ \hline $	h_3
	→ Type Window Console Window f_1: boolean f_2: boolean f_3: boolean	

### View > Find (1/3)

#### 1. Select View > Find in Menu.





### View > Find (2/3)

2. When the dialog appears, input the name of a node which users want to find.



### View > Find (3/3)

#### 3. Focus moves to the found node.

NuSRS Editor 2.0	¤ <sup>⊄</sup> ⊠
File Edit View Window Verification H	lelp
2 🖗 🗦 🖇 🔒 1	
📲 Hierarchy Window	
<b>P</b> □ Root □ th_5 □ h_1 □ f_17	
E Description Window	인역 <b>?</b> Find node name (regex) [3 환민 취소 [4
Carbonia      Content      Content	
1_3 : bollean f_4 : boolean f_2 : boolean f_1 : boolean	Type Window Console Window  f_1: boolean  f_17: boolean
Output	f_2:boolean



### View > View I/O Vars (1/2)

#### 1. Select View > View IO Vars in Menu.





### View > View I/O Vars (2/2)

#### 2. List of I/O variables appears.

	NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	- 합친거ttest.xml
	File Edit View Window Verification Hel	p
	2 😜 🛃 🍕 🕞 😭	
	Hierarchy Window	
	P- ☐ Root	
		📓 I/O variables at Root 🛛 🗙
		// Input variables
		1_1 h_1
		<u>f_4</u> <u>f_17</u>
		// Output variables
		h_1
		f_17 h_3 th_5
	P-□ Root	<u>t_</u> 5 <u>t_</u> 5
	🗣 🚍 TemplateNumber 🔤	
	- 🗋 f_1 : boolean	
	- C f_2 : boolean	Uppe window Console Window
	□ □ f_3 : boolean	f_1: boolean
		f_1/: boolean
KAIST		

### View > Zoom In (1/2)

# Select View > Zoom In. (or press + in the keypad)





### View > Zoom In (2/2)

#### 2. Zoom in on the current active window.





### View > Zoom Out (1/2)

# Select View > Zoom Out. (or press – in the keypad)





### View > Zoom Out (2/2)

#### 2. Zoom out of the current active window.



### View > Zoom Selected (1/3)

1. Select the area to be zoomed by clicking the left mouse button or by drag & drop.

File Edit ) Gene Möndern ) ferification II	S - 압진 거제est.xmi	
File Edit View Window Verification H		
1 🗳 🛃 🚳 🕑 🗎	🗎 💹 🔝 🛰 🔍 🔍 🔯 💷 💌 🔄 🗮	
Hierarchy Window	O Root ×	
P-☐ Root └ Ŋg_abc		<u> </u>
	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	
Construction     C		
<ul> <li>f_2: boolean</li> <li>f_3: boolean</li> <li>f_4: boolean</li> <li>Output</li> </ul>	Type Window Console Window  f_1: boolean f_17: boolean f 2: boolean	



#### View > Zoom Selected (2/3)

#### 2. Select View > Zoom Selected.



### View > Zoom Selected (3/3)

3. The selected image is zoomed in to fill the current window space.



### View > Fit to Window (1/2)

#### 1. Select View > Fit to Window.





### View > Fit to Window (2/2)

KALE

2. The current active image is adjusted to fit the window space.

NuSRS Editor 2.0 - D:'LAB PROJECT'NuSR	S - 합친거test.xml ·	`ø' (
P- Root g_abc	$\begin{array}{c} \hline \\ \hline $	
	$f_4$ $f_5$ $f_5$	
f_1 : boolean     f_2 : boolean     f_3 : boolean     f_4 : boolean     f_4 : boolean     f_4 : boolean	Type Window     Console Window       f_1: boolean     f_17: boolean       f_2: boolean     f_2: boolean	(

### View > Auto Align (1/2)

#### 1. Select View > Auto Align.





### View > Auto Align (2/2)

2. Nodes and arrows in the current active window are automatically aligned and placed.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	6 - 합친거test.xml	r 🛛 🗅
		G
P Root   g_abc     Bescription Window     P   Root   P   Description   P   TemplateNumber	$\begin{array}{c} \mathbf{k} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{c} \\ \mathbf{f}_{1} \\ \mathbf{f}_{2} \\ \mathbf{f}_{3} \\ \mathbf{f}_{4} \\ \mathbf{f}_{4} \\ \mathbf{f}_{4} \\ \mathbf{f}_{5} \\ \mathbf{f}_{6} \\ \mathbf{f}_{6} \\ \mathbf{f}_{7} $	_1 _7 _3 _5 _5
C 1 : boolean C 1 : boolean C 2 : boolean C 1 : boolean C 2 : boolean	Image: Type Window     Image: Console Window       f_1: boolean     f_17: boolean       f_2: boolean     f_2: boolean	



#### **Close Window**

1. Select Window > Close. The current active window is then closed.





### **Close all Windows (1/2)**

#### 1. Select Window > Close All.



### Close all Windows (2/2)

#### 2. All open windows are then closed.

	🗍 NuSRS Editor 2.0 - D:LAB PROJECT/NuSRS - 합친거\test_3.xml	- d 🛛
	File Edit View Window Verification Help	
	▶ 3 4 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
	Hierarchy Window	
	P ← □     Root       - □     th_5       - □     h_1       - □     f 17	
	E Description Window	
	← □ [17]         ← □ Description         ← □         ← □ TemplateNumber         ← □         ← □         ← □	
KAIST	General Fairbonean fairbonean fairbonean	

### **Inquire Program Information (1/2)**

#### 1. Select Help > About.





### **Inquire Program Information(2/2)**

#### 2. Information of the NuSRS program is given.









# **Hierarchy Window**



KAIS1

Software Engineering Lab. Deptpartment of Computer Science



### **Hierarchy Window**



- Hierarchy Window shows the hierarchical structure of the opened file
- Root is the highest FOD.
- By double clicking a tree branch, the corresponding FOD is opened or becomes active.





# **Description Window**



Software Engineering Lab. Deptpartment of Computer Science

KAIS

**한국과학7**[술원

## **Description (1/9)**



 Gives detailed description of the current active window.

#### FOD name

- Description window has the current active window's name (node name) as the root.
- Maximize Button

### **Description (2/9)**

Desctiprion
<b>P</b> - □ th_5
🗛 - 🗂 Description
🗌 🗋 th_5 노드에 대한 설명 삽입
←  ☐ TemplateNumber
🕈 📑 Input
f_2
— 🗋 th_1
🗏 🗋 h_3
P
📙 🖵 🗅 th_5
🛉 🚍 Memoriazble Variable of External input
└─ <b>``</b> f_2_t0 : {10,20,40}
•-
└_ <b>`</b> ) k_1 ≔ 4
🕈 🚍 Local Clock Variable
└─ <b>L</b> ) time_1 : 04
Previous State Variable
🗆 🗋 prev : boolean

#### Description

- Shows information of the current active diagram
- The node can be edited by double clicking the paper shaped icon.
- (Applies to all FOD, SDT, FSM, TTS)

### **Description (3/9)**



#### Template Number

- Logic represented by FSM or TTS has a certain type of template. Input the corresponding template number.
- The node can be edited by double clicking the paper shaped icon.
- (Applies only to FSM, TTS)

### **Description (4/9)**



#### Input

- Shows all the input variables of the current active window.
- Impossible to make direct changes in the description window.
- When the input variable name in the parent FOD is changed, the same variable in the child FOD is automatically adjusted.
- (Applies to all FOD, SDT, FSM, TTS)

### **Description (5/9)**



#### Output

- Shows all the output variables of the current active window.
- Impossible to make direct changes from the description window.
- When the output variable name in the parent FOD is changed, the same variable in the child FOD is automatically adjusted
- f\_, h\_, and th\_ nodes have only one output variable.
- (Applies to all FOD, SDT, FSM, TTS)

### **Description (6/9)**

Desctiprion 🔲
<b>P</b> − <b>□</b> th_5
- C Description
🗆 🗋 th_5 노드에 대한 설명 삽입
←  ☐ TemplateNumber
←
f_2
— 🖺 th_1
h_3
→  →  →  →  →  → → → → → → → → → → → →
th 5
🗛 📑 Memoriazble Variable of External input
🗆 🗋 f_2_t0 : {10,20,40}
🗣 🗂 Constant
└─ 🗋 k_1 := 4
👇 🗂 Local Clock Variable
🗌 🖳 time_1 : 04
🔶 🗂 Previous State Variable
🗕 🗋 prev : boolean

#### Memorizable Variable of External input

- Declare a variable that describes the previous scan value of an input variable
- By adding a suffix such as \_t0, \_t1 to the input variable name, define which scan value it is.
  - t0 is one scan before, t1 is two scans before etc.
- When "Memorizable variable of External input" is selected, Add, Delete and Edit can be performed by clicking the right mouse button.
- (Applies only to FSM,TTS)

### **Description (7/9)**



#### Constant

- Declare a constant value to be used in the corresponding diagram
- When the "Constant" is selected, Add, Delete and Edit can be performed by clicking the right mouse button.
- (Applies to all FOD, SDT, FSM, TTS)

### **Description (8/9)**

E Desctiprion
P- □ th_5
🔶 📺 Description
🗌 🗌 🗋 th_5 노드에 대한 설명 삽입
🛉 🗂 TemplateNumber
P− □ Input
— 🗋 f_2
— 🗋 th_1
h_3
🚽 👝 📺 Output
📙 📙 🗋 th_5
📔 👇 🗂 Memoriazble Variable of External input
📙 🔚 f_2_t0 : {10,20,40}
🛉 🗂 Constant
└─ 🗋 k_1 := 4
🗛 🗂 Local Clock Variable
📙 🖳 time_1 : 04
Previous State Variable
📙 🦳 🗋 prev : boolean
[ <u> </u>

#### Local Clock Variable

- When TTS requires a time related transition label, a clock variable must be declared.
- When the "Local Clock Variable" is selected, Add, Delete and Edit can be performed by clicking the right mouse button.
- (Applies only to TTS)

### **Description (9/9)**



#### Previous State Variable

- Declare a variable that describes the previous scan value of an output variable
- By adding a suffix such as \_t0, \_t1 to the output variable name, define which scan value it is.
  - t0 is one scan before, t1 is two scans before etc.
- When the "Previous State Variable" is selected, Add, Delete and Edit can be performed by clicking the right mouse button.
- (Applies only to FSM,TTS)

## FOD,FSM(TTS),SDT Edit Window



한국과학기술원 Software Engineering Lab. Deptpartment of

KAIST

**Computer Science**
### **FOD Edit Window – Full Screen**

- When a group node starting with g\_ is double clicked
- Window that can edit the FOD

NuSRS Editor 2.0 - D:LAB PROJECT/NuSRS	- 합친거itest.xml	a' D
The Lat View Window Vernation The	,	٦
Ŷ- ☐ Root ☐ g_abc	$\begin{array}{c} \mathbf{h} \\ \mathbf{h} \\ 0 \\ \underline{\mathbf{f}_{1}} \\ \underline{\mathbf{f}_{2}} \\ \underline{\mathbf{f}_{2}} \\ \mathbf{f}_{1} \mathbf{f}_$	
	$f_{-4}$ $f_{-4}$ $f_{-4}$ $f_{-5}$ $f_{-5}$ $f_{-5}$ $f_{-5}$ $f_{-5}$ $f_{-5}$ $f_{-5}$ $f_{-5}$ $f_{-7}$ $f_{-1}$ $f$	



### **FOD Edit Window – Details**



### FOD Edit Window – Button (1/9)

#### Select

- Used to select each node and transition
- After clicking the selection button, click the node and transition in the edit window with the left mouse button
- Drag & drop is available for multiple selections.



### FOD Edit Window – Button (2/9)

### IO node

- Used to edit input output nodes.
- IO node can only be connected to one outgoing transition or incoming transition.
- IO node can only be created in the top most FOD; once created in the top most FOD it is automatically created in the lower FOD.
- When an IO node is deleted from a lower FOD, the transition from the top most FOD is deleted, but the IO node itself is not deleted.



### FOD Edit Window – Button (3/9)

### IO node (cont.)

- 1. Click the IO node button and click the left mouse button on the editing window.
- 2. When a dialog box for entering the node's name is shown, enter a node name and click ok.
- 3. As soon as the node's name is entered the node is edited.



### FOD Edit Window – Button (4/9)

### • Group node

- Used to edit group nodes starting with g\_.
- If the name does not begin with a g\_, an error message is given and the name is re-entered.
- 1. Click the group node button and click the left mouse button on the edit window.
- 2. When a dialog box for entering the node's name is shown, enter a node name and click ok.
- 3. As soon as the node's name is entered the node is edited.



### FOD Edit Window – Button (5/9)

#### Function node

- Used to edit function nodes starting with f\_.
- If the name does not begin with a f\_, an error message is given and the name is re-entered.
- 1. Click the function node button and click the left mouse button on the edit window.
- 2. When a dialog box for entering the node's name is shown, enter a node name and click ok.
- 3. When a dialog box for entering the node's name is shown, enter a node name and click ok.



### FOD Edit Window – Button (6/9)

### History node

- Used to edit history nodes starting with h\_.
- If the name does not begin with a h\_, an error message is given and the name is re-entered.
- 1. Click the history node button and click the left mouse button on the edit window.
- 2. When a dialog box for entering the node's name is shown, enter a node name and click ok.
- 3. When a dialog box for entering the node's name is shown, enter a node name and click ok.



### FOD Edit Window – Button (7/9)

#### Timed History node

- Used to edit timed history nodes starting with th\_.
- If the name does not begin with a th\_, an error message is given and the name is re-entered.
- 1. Click the timed history node button and click the left mouse button on the edit window.
- 2. When a dialog box for entering the node's name is shown, enter a node name and click ok.
- 3. When a dialog box for entering the node's name is shown, enter a node name and click ok.



 $\bigcirc$ 

### FOD Edit Window – Button (8/9)

#### Transition

- Used to edit transitions between nodes.
- A Self-Transition in FOD is not allowed.
- After clicking the Transition button once, multiple transitions can be drawn until another button is clicked.
- As soon as a transition from an IO node is drawn, we decide whether it is an input or output node, and this information change automatically in the description window.



### FOD Edit Window – Button (9/9)

### Transition (cont.)

- 1. Click the Transition button.
- 2. Click and drag the Source node, and drop it at the Destination node.
  - When making a transition between group nodes, must input a transition name.
    - A transition name must be one of the output nodes' names from the source group node.
    - When a group node → group node transition is drawn, a input node with the same name is automatically created in the target group node.
    - When a transition name is not one of the output node's names in the source group node, then an output node with the corresponding name is automatically created.



 $\rightarrow$ 

### **FOD Edit Window – Rename Node**

1. Select the node, right click and select Rename.

🔲 NUSRS Editor 2.0 - D:'LAB PROJECT/NUSRS	S - 합친거ttest.xml	î 🗵
File Edit View Window Verification Hel	lp	
Hierarchy Window		
<pre>P □ Root</pre>		
		=
E Description Window	$f_4$ $th_5$	
P C Root ▲ P C Description		
← ☐ ← ☐ TemplateNumber =	$\rightarrow$	
P □ Input		
– 🗋 f_1 : boolean – 🗋 f_2 : boolean	Type Window Console Window	
- C f_3 : boolean	f_1:boolean	
	f_1/: boolean f_2: boolean	•

### **FOD Edit Window – Rename Node**

2. Change the name when the node name becomes a changeable state, then click the left mouse button and the renaming is done.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSI	RS - 합친거itest.xml	- 0 X
File Edit View Window Verification H	elp	
P 🖗 📙 🚳 😽 🗎 I		
nterarchy Window		
P- ☐ Root > ☐ g_abc		
		h_1
		f_17
		h_3 =
Description Window		th_5
← C Root		
	$\rightarrow$	
P ☐ Input	-	
- D f 2 : boolean	🔀 Type Window 🖉 Console Window	
- 🗋 f_3 : boolean	f 1: boolean	
🗆 🗋 f_4 : boolean	f_17 : boolean	
Output	f_2: boolean	•



### **SDT Edit Window – Full Screen**

# When double clicked a node starting with f\_ A table-like window that can edit SDT

ile Edit View Window Verification Help				
🞦 🜮 🛃 🐝 😭 😭	🗶 🗔 💀 🔍 🔍 🖾 🖃 🖉 🗶 🔜			
Hierarchy Mindow				
<pre>P □ Root</pre>	Structured Decision Table:			
	Conditions	1	2	3
	h_3 > th_1		2	
	Action	1	2	3
	$f_{-6} := 1$			
	f_6 := 0			
🛐 Description Window				
₽- 🚍 f_17				
🕈 🚍 Description				
TemplateNumber				
e 📑 Input				
f_3:boolean				
f_17 : boolean				
🕈 🚍 Constant	Type Window E Console Window			
false = 0	f_17 : boolean			
Local Clock Variable	f_2:boolean			
– <u> </u> Previous State Variable	T_3: poolean			
Memoriazble Variable of External input				



86



# SDT Edit Window – Details (2/2)



#### Structured Decision Table

- A table based on NuSCR semantics.
- By clicking the right mouse button at each row/column, a row/column can be added, copied, pasted, deleted etc.



### FSM, TTS Edit Window – Full Screen

- When a node starting with h\_ or th\_ is double clicked
- A window that can edit FSM and TTS

NuSRS Editor 2.0 - D:\LAB PROJECT\ File Edit View Window Verification	uSRS - 합친거itest_3.xml Help
P P R P P P P	· · · · · · · · · · · · · · · · · · ·
P-□ Root □ th_5 □ h_1 □ f_17	$ \begin{array}{c} \mathbf{k} \\ \mathbf$
	Type Window Console Window
← ☐ th_5 : boolean ← 급 Constant ← ☐ true := 1	T_3: boolean f_4: boolean th_5: boolean



89

# FSM, TTS Edit Window – Details





# FSM, TTS Edit Window – Button (1/3)

#### Select

- Used to select the State and transition.
- After clicking the Selection button, click the left mouse button on the state and transition to be edited in the edit window.
- Drag & drop is possible for Multiple Selection.



# FSM, TTS Edit Window – Button (2/3)

#### State

 $\bigcirc$ 

- Used to draw States.
- A State name must not include blank spaces or symbols.
- 1. Click the State button and click the left mouse button on the edit window.
- 2. When a dialog box for entering the State name pops up, enter the State name and click ok.
  - If the State name is entered wrong, an error message is given and the name is re-entered.



As soon as the State name is inputted, the state is edited.

# FSM, TTS Edit Window – Button (3/3)

#### Transition

- Used to edit transitions between States.
- In FSM and TTS, at most one self-Transition can be drawn.
- Once clicking the Transition button once, many transitions can be drawn until another button is pressed.
- 1. Click the Transition button.
- 2. Click the Source state and drag, drop at the Target state.



#### FSM, TTS Edit Window – Rename State (1/2)

#### 1. Click the state, right click and select Rename.



#### FSM, TTS Edit Window – Rename State (2/2)

2. Change the name when the node name becomes a changeable state, then click the left mouse button and the renaming is done.

NuSRS Editor 2.0 - D:\LAB PROJECTWus	RS - 합친거\test_3.xml
File Edit View Window Verification	Help
264	
📲 Hierarchy Window	
<b>P</b> □ Root □ th_5 □ h_1 □ f_17	$\overrightarrow{r_{2,10} > f_{2,2,10} > f_{$
	(k_1, k_1) prev <u>state_1</u> 5 := 0
	=
P→ □ Input	
$ \int f_4$ : boolean	
Providencial of the second and t	Type Window Console Window
th_5 : boolean	f_3: boolean
- Constant	f_4 : boolean
-   <u>`</u> true := 1	th_5 : boolean

95

#### FSM, TTS Edit Window – Set Initial State (1/2)

1. Click the state, then right click and select set initial state.

NuSRS Editor 2.0 - D:'LAB PROJEC	NuSRS - 합친거test_3.xml n Helo	
E	) <u> </u>	<b>k</b>
📲 Hierarchy Window	□ O Root O th_5 ×	
<b>P</b> — ☐ Root - ☐ th_5 - ☐ h_1 - ☐ f_17		2_t0 > f_2 & th_1 >= h_3 / th_5 := 1
Description Window	(k_1, k_1) prev <= th 54	fh_5 := 0 Set initial state
P ☐ th_5 P ☐ th_5 P ☐ Description P ☐ TemplateNumber P ☐ TemplateNumber P ☐ Input		
f_3 : boolean		
└─ 🗋 f_4 : boolean •- 📼 Output	🔀 Type Window 🖉 Console Window	
L h_5 : boolean	f_3 : boolean	
🕈 📑 Constant	f_4 : boolean	
— 🗋 true := 1	➡ th_5 : boolean	



#### FSM, TTS Edit Window – Set Initial State (2/2)

2. The selected state becomes an initial state, and the original initial state becomes a normal state.

File Edit View Window Verification	n Helb
Hierarchy Window	
P- ☐ Root - ☐ th_5 - ☐ h_1 - ☐ f_17	
E Description Window	(k_1, k_1) prev <= th_5/th_5 := 0
←	
	yve mindow VE CONSOLE WINDOW
P ☐ 1_4 : boolean P ☐ Output	
<pre></pre>	f_3:boolean
← ☐ Output ← ☐ Output ← ☐ Constant ← ☐ Constant	f_3: boolean f_4: boolean



97

# **Type Window**



한국과학기술원 Software Engineering Lab.

KAIST

Deptpartment of Computer Science

### **Type Window**

 A window that records all variables used in the diagram, and allows editing of the variable types

🔲 NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS - ğ	f친거(test_3.xm)	01
File Edit View Window Verification Help		
Hierarchy Window		
<pre>P-□ Root - □ th_5 - □ h_1 - □ f_17</pre>	$\begin{array}{c} \\ \hline \\ $	
		=
Description Window      Call Root      Call Description      Call Content of the second		-
- C f_3: boolean		►
	Type Window  f_1: boolean  f_17: boolean  f_2: boolean  f_2. boolean	



## Edit Type Window (1/3)

#### 1. Click the variable and double click.



## Edit Type Window (2/3)

2. In the variable type dialog box select the type and click ok.

f_1 🔀	
name: f_1	change the original setting
🔾 Value	of boolean into range type
🔾 Boolean	
🖲 Range Start: 0 End: 100	
Enumeration     Add     Remove	
OK Cancel	



### Edit Type Window (3/3)

#### 3. The variable's type has been changed.



### **Other supported item functions**

 Check whether variable names are repeated or not.



# **Quick Check**





**KAIST** 

<u>한국과학7</u>]술원

Software Engineering Lab. Deptpartment of **Computer Science** 

# Verification > Quick Check (1/4)

#### 1. Select Verification > Quick Check.





Verification > Quick Check (2/4)	
2. See the Result on the right of the Type Wind	low.
Image: Section 2.0. C:Documents and Settings/GRS/HE/2 319/WisrRS 2007/2028/P-2005/1019-Final-Verification.xml       p* p* p*         File Edit View Window Verification Help       Image: Section 2.0. P/2R_PRESS x         Image: Herarchy Window       Image: Section 2.0. P/2R_PRESS x         Image: Type Window       Image: Section 2.0. P/2R_PRESS x         Image: Section 2.0. P/2R_PRESS_IM_SP       Image: Section 2.0. P/2R_PRESS_IM_SP         Image: Section 2.0. P/2R_PRESS_IM_SP       Image: Section 2.0. P/2R_PRESS_Val_V         Image: Section 2.0. P/2R_PRESS_IM_SP       Image: Section 2.0. P/2R_PRESS_IM_SP         Error At f_LO_P/2R_PRESS_IM_SP, Action Row : 0 Col : 0 - Undefined Variable : f_LO_P/2R_PRESS_IM_SP         Error At f_LO_P/2R_PRESS_IM_SP       Image: Section 2.0. P/2R_PRESS_IM_SP	Du_t1t
Imput       Imput <td< td=""><td>106</td></td<>	106

## Verification > Quick Check (3/4)

 Quick Check checks diagram which is being edited and child nodes of the diagram.

 To check entire model, execute Quick Check when you are editing root node.



# Verification > Quick Check (4/4)

#### Errors checked by Quick Check

- 1. Undefined variables in SDT's Conditions and Actions
- 2. Syntax errors in SDT conditions and actions
- 3. More than one action allocated in SDT conditions
- 4. Undefined variables in FSM and TTS transitions
- 5. Syntax errors in FSM and TTS Transitions
- 6. FOD, FSM and TTS nodes which have no transition
- 7. Nodes unreachable from initial state in FSM and TTS
- 8. Output variable and its connected node have different names


## **Model Checking**



KAIS1

Software Engineering Lab. Deptpartment of Computer Science



## Verification > Model Checking (1/6)

### 1. Select Verification > Model Checking.



### Verification > Model Checking (2/6)

2. A result window that converts the current window's NuSCR contents into SMV input language is given.



### Verification > Model Checking (3/6)

3. After entering the verification properties, click the Apply button and it is applied to the SMV input (SMV input window is editable)

SMV			
init(cycle) :=0;		<b>_</b>	
next(cycle) := case			
cycle < 20 : cycle + 1;			
cycle = 20 : 0;			
1 : cycle;			
esac;			
DEFINE			
sec := cycle = 20;			
true:= 1;			
false:= 0;			
		►	-
Save	Close	Execution	
PROPERTY			Input vorification
PROPERTY AGIT HILLOG POWER On Byn Initif HI	LOG POWER On Byn Init=true	e& of Mod Errf Mod Err=true I	<ul> <li>Input verification</li> </ul>
PROPERTY AG(f_HI_LOG_POWER_Op_Byp_Init.f_HI_	LOG_POWER_Op_Byp_Init = true	العلم (f_Mod_Err.f_Mod_Err = true	Input verification
PROPERTY AG(f_HI_LOG_POWER_Op_Byp_Init.f_HI_	LOG_POWER_Op_Byp_Init = true	v & (f_Mod_Err.f_Mod_Err = true	<ul> <li>Input verification properties (CTL)</li> </ul>
PROPERTY	LOG_POWER_Op_Byp_Init = true	e & (f_Mod_Err.f_Mod_Err = true	<ul> <li>Input verification properties (CTL)</li> </ul>
PROPERTY	LOG_POWER_Op_Byp_Init = true	e & (f_Mod_Err.f_Mod_Err = true	<ul> <li>Input verification properties (CTL)</li> </ul>
AG(f_HI_LOG_POWER_Op_Byp_Init.f_HI_	LOG_POWER_Op_Byp_Init = true	e & (f_Mod_Err.f_Mod_Err = true	<ul> <li>Input verification properties (CTL)</li> </ul>



### Verification > Model Checking (4/6)

# 4. Click the Execution button and the SMV tool is executed.

<pre>SWOocuments and SettingSWAIE3WMy DocumentsWAIAIWKNICS 프로24,, X</pre> <pre>SM/</pre> <pre>SM</pre>		
SMV         int(cycle) = 0;         next(cycle) = case         option = 0;         DEFINE         sec = cycle = 20;         intes = 1;         false = 0;         SPEC AG(f_H_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true & (f_Mod_Enrf_Mod_E         PROPHENTY         NG(f_H_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true & (f_Mod_Enrf_Mod_Err=true)         NG(f_H_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true & (f_Mod_Enrf_Mod_Err=true)         NG(f_H_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true & (f_Mod_Enrf_Mod_Err=true)         NG(f_H_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true & (f_Mod_Errf_Mod_Err=true)         NG(f_H_LOG_POWER_O	🎂 C:₩Documents and Settings₩지은경₩My Documents₩과제₩KNICS 프로젝 🔀	File Prop View Goto History Abstraction
mit(cycle) =0; net(cycle) = case cycle = 20: cycle + 1; cycle = 20: 0; 1: cycle; esa;       Name Layer         DEFINE sec = cycle = 20; nue= 1; false= 0;       Source Irace Log         Source Irace Log       File Show         File Show       File Show         FORM-Normal-TO-Waiting-taken : Waiting; 1: STATE; esac;       I STATE; esac;         PROPERIV       FROM-DOWER_Op_Byp_Init_HILLOG_POWER_Op_Byp_Init= true & (f_Mod_Errf_Mod_Err_f_Mod_Err = true)         Nome Layer       Cose         PROPERIV       FROM-Normal-TO-Waiting-taken : Waiting; 1: STATE; esac;         Nome Log       FROM-Normal-TO-Waiting-taken : false; FROM-Normal-taken : false; FROM-Normal-taken : false; FROM-Normal-taken : false; I : th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Op_Byp_Init_HILLOG_POWER_OP_Byp_Init= true & (f_Mod_Errf_Mod_Err=true)         Nome Layer       I : th_HI_LOG_POWER_OP_Byp_Init_HILLOG_POWER_OP_Byp_Init	SMV	File Flob Hen Zoto History Providence
<pre>int(cycle) = 0; next(cycle) = case cycle = 20:0; 1:cycle = 20:0; 1:cycle = 20; tue=1; talse=0; SPEC AG(f_HL_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true &amp; (f_Mod_Errf_Mod_E PROM-Normal-TO-Waiting-Taken : Waiting; 1 : STATE; esac; PROM-Normal-TO-Waiting-To-Normal-taken : false; PROM-Normal-TO-Waiting-To-Normal-taken : false; PROM-Waiting-TO-Normal-taken : false; 1 : th HL_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true &amp; (f_Mod_Errf_Mod_Err = true) No(f_HL_LOG_POWER_Op_Byp_Intf_HL_LOG_POWER_Op_Byp_Int = true &amp; (f_Mod_Errf_Mod_Err = true) No(f_HL_LOG_POWER_Op_Byp_Int = true &amp; (f_Mod_Errf_Mod_Err = true) No(f_HL_LOG_POWER_OP_BIP_Errf_LOG_POWER_OP_BPIP_Errf_LOG_POWER_OP_BPIP_Errf_LOG_POWER_OP_BPIP_Errf_LOG_POWER_OP_BPIP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_OP_Errf_LOG_POWER_O</pre>		Browser Properties Pecults Cope Heing Groups
<pre>Mid(cycle) = case cycle = 20: cycle = 1; cycle = 20: cycle = 20; ticycle; esa; bEFINE sec = cycle = 20; true = 1; mise= 0; SPEC AG(f,H_LOG_POWER_Op_Byp_Init_HL_OG_POWER_Op_Byp_Init= true &amp; (f_Mod_Errf_Mof_Errf_Mod_Errf_Mod_Errf_Mod_Errf_Mod_</pre>		Elonser   Hebernes   Kesuns   Cone   csing   Cloubs
<pre>next(cycle) = case cycle &gt; 20:0(e + 1; cycle + 1; cycle + 1; cycle &gt; 20:0; 1:cycle; esac; DEFINE sec=cycle = 20; true= 1; false= 0; SPEC AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err_f_Mod_E SPEC AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err_f_Mod_E PROPERIY NG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err_f_Mod_Err= true) NG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err_f_Mod_Err= true) Apply</pre>	init(cycle) := 0,	
<pre>cycle 20:cycle +1; cycle 20:cycle +1; cycle 20:cycle +1; esa; DEFINE sec = cycle 20; twe=1; false= 0; SPEC A0(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err.f_Mod_E Save Close Execution FROM-Normal-TO-Waiting-taken : Waiting; 1 : STATE; esac;  Outputs init(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(the_I) := 0; next(time_I) := 0; ne</pre>	next(cycle) := case	Name Layer
<pre>cycle = 20:0, 1:cycle; esac; DEFINE sec = cycle = 20; true= 1; false= 0; SPEC AO(f_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_E Save Close Execution PROPERIY NG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) AG(f_HL_LOG_POWER_Op_Byp_Init_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Errf_Mod_Err=true) ABPDY</pre>	cycle < 20: cycle + 1;	⊞ <u>(, (top level)</u> )
<pre>1 : cycle; esac; DEFINE sec : cycle = 20; true = 1; false = 0; SPEC AG(f_HL_LOG_POWER_Op_Byp_Init=true &amp; (f_Mod_Err.f_Mod_E Save Close Execution FROH-Normal-TO-Waiting-taken : Waiting; 1 : STATE; esac;  Outputs init (th HI_LOG_POWER_Ptrp_Logic) := 0; hext (th HI_LOG_POWER_Ptrp_Logic) := 0; hext (th HI_LOG_POWER_Ptrp_Logic) := case FROH-Waiting-TO-Normal-taken : false; FROH-Waiting-TO-Normal-taken : false; FROH-Waiting-TO-Normal-taken : true; FROH-Vaiting-TO-Normal-taken : true; FROH-Ptertip-To-Normal-taken : true; FROH-Ptertip-To-Normal-taken : false; 1 : th HI_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err:f_Mod_Err=true) Apply</pre>	cycle = 20 : 0;	
<pre>esac; DEFINE sec = cycle = 20; true= 1; false= 0; SPEC AG(f_HI_LOG_POWER_Op_Byp_Init=HIL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err.f_Mod_E</pre>	1 : cycle;	
DEFINE sec := cycle = 20; true = 1; false = 0; SPEC AG(f_HL_LOG_POWER_OP_Byp_Init_HL_LOG_POWER_OP_Byp_Init = true & (f_Mod_Errf_Mod_E Save Close Execution FROM-Normal-TO-Waiting-taken : Waiting; I : STATE; esac; Outputs init(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Ptrp_Logic) := case FROM-Waiting-TO-Normal-taken : false; FROM-Waiting-TO-Pretrip-taken : true; FROM-Waiting-TO-Pretrip-taken : true; FROM-Waiting : true; FROM-Pretrip-taken : true; FROM-	esac;	
DEFINE         sec:= cycle = 20; true= 1; false= 0;         SPEC AG(f_HI_LOG_POWER_OP_Byp_Init_HIL_OG_POWER_OP_Byp_Init= true & (f_Mod_Errf_Mod_E         FROM-Normal-TO-Waiting-taken : Waiting; 1 : STATE; esac;         Save       Close         PROPERTY         RG(f_HI_LOG_POWER_OP_Byp_Init_HIL_OG_POWER_OP_Byp_Init= true & (f_Mod_Errf_Mod_Err= true)         PROPERTY         AG(f_HI_LOG_POWER_OP_Byp_Init_HIL_OG_POWER_OP_Byp_Init= true & (f_Mod_Errf_Mod_Err= true)         Image: true = 1         Close         Execution         PROPERTY         AG(f_HI_LOG_POWER_OP_Byp_Init_HIL_LOG_POWER_OP_Byp_Init= true & (f_Mod_Errf_Mod_Err= true)         Image: true = 1         The main true = 1         The main true = 1         The main true = 1         Image: true = 1		
<pre>sec := cycle = 20; true= 1; false= 0; SPEC AG(<u>T</u>_HI_LOG_POWER_Op_Byp_Init_HIL_LOG_POWER_Op_Byp_Init= true &amp; (<u>Mod_Errf_Mod_E</u> Save Close Execution FROM-Normal-TO-Waiting-taken : Waiting; i : STATE; esac;  Outputs init(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Ptrp_St) : the_1 + 1; 1:0;</pre>	DEFINE	Source Trace Log
<pre>fue=1; false=0; SPEC AG(f_HL_LOG_POWER_Op_Byp_Init_f_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err.f_Mod_E Save Close Execution PROPERTY AG(f_HL_LOG_POWER_Op_Byp_Init_f_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err.f_Mod_Err = true) AG(f_HL_LOG_POWER_Op_Byp_Init_f_HL_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err.f_Mod_Err = true) Apply</pre> File Show FROM-Normal-TO-Waiting-taken : Waiting; 1 : STATE; esac; Outputs init (th_HI_LOG_POWER_Ptrp_Logic) := 0; next (th_HI_LOG_POWER_Ptrp_Logic) := case FROM-Waiting-TO-Normal-taken : false; FROM-Waiting-TO-Normal-taken : false; 1 : th_HI_LOG_POWER_Ptrp_Logic; esac; Increment local clock time_1 init(time_1) := 0; next (time_1) := 0;	sec := cycle = 20;	
<pre>false= 0; SPEC AG(f_HL_LOG_POWER_Op_Byp_Init=fue &amp; (f_Mod_Err.f_Mod_E Save Close Execution PROPERTY AG(f_HL_LOG_POWER_Op_Byp_Init=fue &amp; (f_Mod_Err.f_Mod_Err=fue) AG(f_HL_LOG_POWER_Op_Byp_Init_fue &amp; (f_Mod_Err.f_Mod_Err=fue) AG(f_HL_LOG_POWER_Op_Byp_Init=fue &amp; (f_Mod_Err.f_Mod_Err=fue) AG(f_HL_LOG_POWER_Op_Byp_Init=fue) AG(</pre>	true:= 1;	File Show
<pre>SPEC AG(T_HI_LOG_POWER_Op_Byp_Init_HI_LOG_POWER_Op_Byp_Init= true &amp; (f_Mod_Err.f_Mod_E</pre>	false:= 0;	The show
<pre>SPEC AG((_Hi_LOG_POWER_Op_Byp_Init_fHi_LOG_POWER_Op_Byp_Init= true &amp; ((_Mod_Err.f_Mod_E</pre>		FROM-Normal-TO-Waiting-taken : Waiting;
<pre>esac;  Outputs init(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_P</pre>	SPECAG(FHILOG POWER Op Byp Init.fHILOG POWER Op Byp Init=true&(fMod Err.fMod E	1 : STATE;
<pre> Outputs init(th HI_LOG_POWER_Ptrp_Logic) := 0; next(th HI_LOG_POWER_Prop_Logic) := case FROM-Waiting-TO-Normal-taken : false; FROM-Waiting-TO-Pretrip-taken : true; FROM-Pretrip-TO-Normal-taken : false; 1 : th HI_LOG_POWER_Ptrp_Logic; esac;  Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( fHI_LOG_POWER_Val_Out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>		esac;
<pre> Outputs init(th_HI_LOG_POWER_Ptrp_Logic) := 0; init(th_HI_LOG_POWER_Ptrp_Logic) := case FROM-Waiting-TO-Normal-taken : false; FROM-Waiting-TO-Pretrip-taken : true; FROM-Pretrip-TO-Normal-taken : false; I : th_HI_LOG_POWER_Ptrp_Logic; esac;  Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( f_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>		
Save       Close       Execution         PROPERTY       Init (th HI_LOG_POWER_Ptrp_Logic) := case         AG((_HI_LOG_POWER_Op_Byp_Init_fHI_LOG_POWER_Op_Byp_Init= true & ((_Mod_Err=true)       FROM-Waiting-TO-Normal-taken : false;         FROM-Pretrip-TO-Normal-taken : false;       FROM-Pretrip-To-Normal-taken : false;         I : th HI_LOG_POWER_Op_Byp_Init= true & ((_Mod_Err=true)       I : th HI_LOG_POWER_Ptrp_Logic;         esac;       Increment local clock time_1         Init (time_1) := 0;       sec 4 in-Waiting 5 time_1 < k_HI_LOG_POWER_Ptrp_Dly 4 (		Outputs
Save       Close       Execution         PROPERTY       next(th_HI_LOG_POWER_Ptrp_Logic) := case         AG(f_HI_LOG_POWER_Op_Byp_Init=frue & (f_Mod_Err.f_Mod_Err=true)       Ref (f_HI_LOG_POWER_Op_Byp_Init=frue & (f_Mod_Err.f_Mod_Err=true)         AG(f_HI_LOG_POWER_Op_Byp_Init=frue & (f_Mod_Err.f_Mod_Err=true)       I : th_HI_LOG_POWER_Ptrp_Logic;         esac;       Increment local clock time_1         init(time_1) := 0;       next(time_1) := case         sec & in-Waiting & time_1 < k_HI_LOG_POWER_Ptrp_Dly & (f_HI_LOG_POWER_Val_Out > k_HI_LOG_POWER_Ptrp_Dly & (f_HI_LOG_POWER_Ptrp_Set) : time_1 + 1;         1:0;       I:0;		init(th_HI_LOG_POWER_Ptrp_Logic) := 0;
<pre>FROM=Waiting=TO=Normal=taken : false; FROM=Waiting=TO=Normal=taken : false; FROM=Waiting=TO=Pretrip=taken : false; FROM=Waiting=TO=Pretrip=taken : false; FROM=Waiting=TO=Pretrip=taken : false; I : th_HI_LOG_POWER_OP_Byp_Init=true&amp;(f_Mod_Err=true) I : th_HI_LOG_POWER_Ptrp_Logic; esac; Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in=Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( HI_LOG_POWER_Val_Out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>	Save Close Execution	next(th_HI_LOG_POWER_Ptrp_Logic) := case
AG((_HI_LOG_POWER_Op_Byp_Init_HI_LOG_POWER_Op_Byp_Init_true & ((_Mod_Err_f_Mod_Err=true)       FROM-Waiting-TO-Pretrip-To-Normal-taken : fralse;         I : th_HI_LOG_POWER_Op_Byp_Init_true & ((_Mod_Err_f_Mod_Err=true)       I : th_HI_LOG_POWER_Ptrp_Logic;         esac;       Increment local clock time_1         init(time_1) := 0;       next(time_1) := 0;         next(time_1) := case       sec & in-Waiting & time_1 < k_HI_LOG_POWER_Ptrp_Dly & (	PROPERTY	FROM-Waiting-TO-Normal-taken : false;
AG((_HI_LOG_POWER_Op_Byp_Init_HI_LOG_POWER_Op_Byp_Init= true & ((_Mod_Err:f_Mod_Err= true         I : th_HL_OG_POWER_Ptrp_Logic;         I : th_I_LOG_POWER_Op_Byp_Init= true & ((_Mod_Err:f_Mod_Err= true         I : th_I_LOG_POWER_Ptrp_Logic;         esac;       Increment local clock time_1         init(time_1) := 0;       next(time_1) := case         sec & in-Waiting & time_1 < k_HI_LOG_POWER_Ptrp_Dly & (		FROM-Waiting-TO-Pretrip-taken : true;
<pre>1 : th HI LOG POWER Ptrp Logic; esac;  Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( f_HI_LOG_POWER_Val_out &gt; k_HI_LOG_POWER_Val_out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>	AG(FHLLOG POWER On Byn Initf HLLOG POWER On Byn Init=true&(fMod Frrf Mod Frr=true)	FROM-Pretrip-TO-Normal-taken : false;
<pre>esac;  Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( f_HI_LOG_POWER_Val_Out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>	Lo() "Troo" over Cob" ave " "Troo" over Cob" ave " " and " " " and " " " and "	1 : th_HI_LOG_POWER_Ptrp_Logic;
<pre> Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( f_HI_LOG_POWER_Val_Out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>		esac;
<pre>- Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( f_HI_LOG_POWER_Val_Out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;</pre>		
Apply		Increment local clock time_1
Apply		<pre>init(time_1) := U;</pre>
Apply     sec & in-waiting & time_1 < K_H1_LOG_POWER_Ptrp_D1y & (ffH1_LOG_POWER_Val_Out > k_H1_LOG_POWER_Val_Out > k_H1_LOG_POWER_Ptrp_Set) : time_1 + 1;		[next(time_1) := case
Apply k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0;		sec & in-waiting & time_1 < K_HI_LOG_POWER_Ptrp_Dly & (
R_HI_LOG_POWER_Ptrp_set) : time_1 + 1; 1:0;	Annk	II HI LOG POWER VALOUE >
	мрру	[K_ni_bog_Powsk_Ptrp_set) : time_i + 1;
		1:0;



Help

### Verification > Model Checking (5/6)

### 5. Click Prop > Verify or Verify all in SMV.

74 default.smv			
<u>F</u> ile <u>P</u> rop <u>V</u> iew <u>G</u> oto I	H <u>i</u> story <u>A</u> bstraction	<u>H</u> elp	
Bro     Options     I       I     Copy options       Group siblings       Uerify .       Verify all       State count	<u>Results</u> <u>Cone</u> Usi <u>ng</u> <u>G</u> roups		
<u>Source Trace Log</u>			
Fil <u>e</u> Sho <u>w</u>			
<pre>FROM-Normal-TO-Waiting-taken : Waiting; 1 : STATE; esac;  Outputs init(th_HI_LOG_POWER_Ptrp_Logic) := 0; next(th_HI_LOG_POWER_Ptrp_Logic) := case FROM-Waiting-TO-Normal-taken : false; FROM-Waiting-TO-Pretrip-taken : true; FROM-Pretrip-TO-Normal-taken : false; 1 : th_HI_LOG_POWER_Ptrp_Logic; esac;</pre>			
<pre> Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec &amp; in-Waiting &amp; time_1 &lt; k_HI_LOG_POWER_Ptrp_Dly &amp; ( f_HI_LOG_POWER_Val_Out &gt; k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0; esac:</pre>			
Jesac;		<u> </u>	



## Verification > Model Checking (6/6)

### 6. Check the verification results and analyze.

7 default.smv 📃 🗖 🔀	🎀 default.smy
<u>F</u> ile <u>P</u> rop <u>V</u> iew <u>G</u> oto H <u>i</u> story <u>A</u> bstraction <u>H</u> elp	<u>File Prop View Goto Hi</u> story <u>A</u> bstraction <u>H</u> elp
<u>B</u> rowser Properties <u>R</u> esults <u>C</u> one Usi <u>ng</u> <u>G</u> roups	Browser Properties Results Cone Using Groups
All results 📃	All results
Property Result	Property Result
(AG (((f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=1)&(((f_M_false Tue	(AG (((f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=1)&(((f_M_falseTue
Source Trace Log	Source Trace Log
Fil <u>e</u>	Fil <u>e</u> E <u>d</u> it R <u>u</u> n Vie <u>w</u>
Model checking time: 0.093750 user time0.1	
system time0.0	f HI LOG POWER PV Err \in- init
15625 s	f HI LOG POWER PV Err.\in-s0 0
Model checking results	f_HI_LOG_POWER_PV_Err.\in-s1 1
(AG (((f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=1) &(((f_M	f_HI_LOG_POWER_PV_Err.true 1
false	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s0-enabled
See file "default.warn" for warnings.	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s0-taken 0
user time 0.1	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s1-enabled
40625 s	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s1-taken
system time0.0	f_HI_LOG_POWER_Trip_Out.\FROM-s0-TO-s0-enabled
13023 5	

