NI MATRIXx™

SystemBuild™ State Transition Diagram Block User Guide

April 2007
370766C-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 (0) 9 725 72511, France 33 (0) 1 48 14 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466, New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the info code feedback.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

Trademarks

National Instruments, NI, ni.com, MATRIXx, Xmath, and SystemBuild are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on `ni.com/legal` for more information about National Instruments trademarks.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

About This Manual

Chapter 1
Introduction to State Transition Diagrams

Chapter 2
Creating, Editing, and Saving STDs

Chapter 3
Using the STD Editor Window

Appendix A
Technical Support and Professional Services

Index

About This Manual

This manual introduces you to the State Transition Diagram block in SystemBuild. This manual focuses primarily on creating and editing states within this block.

This manual assumes you are familiar with basic Xmath capabilities such as plotting, printing, Xmath command and function syntax, and MathScript programming. For information about these topics, refer to the *Xmath User Guide,* available by selecting **Help»MATRIXx Bookshelf**.

This manual also assumes you are familiar with navigating the MATRIXx Integrated Design Environment (IDE) and the project system. For more information about these topics, refer to the *MATRIXx Getting Started Guide*.

For descriptions of dialog boxes, the user interface, and individual block functionality, refer to the *MATRIXx Help*. Access this help by entering `help` in the Xmath command area or by selecting **Help**»**MATRIXx Help**.

Conventions

The following conventions appear in this manual:

»

The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

📝

This icon denotes a note, which alerts you to important information.

💡

This icon denotes a tip, which alerts you to advisory information.

bold

Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic

Italic text denotes variables, emphasis, a cross-reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.
`monospace italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Related Documentation

You might find the following resources helpful as you use this manual.

- The *SystemBuild User Guide* provides information about creating, editing, and simulating SuperBlocks. Access this guide from the *MATRIXx Bookshelf*.
- The *SystemBuild BlockScript User Guide* provides information about using the BlockScript language for SystemBuild.

The following textbooks contain information about finite state machines.

Note The following textbooks offer useful background information on the general concepts discussed in this documentation. These references are provided for general informational purposes only and are not affiliated, sponsored, or endorsed by National Instruments. The content of these references is not a representation of, may not correspond to, and does not imply current or future functionality in MATRIXx or any other National Instruments product.

- Farhat, Hassan A. 2003. Chap. 8 in *Digital design and computer organization*. Boca Raton, FL: CRC Press.
- Fletcher, William I. 1997. *An engineering approach to digital design*. Upper Saddle River, NJ: Prentice-Hall, Inc.
- Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. 2000. *Introduction to automata theory, language, and computation*. 2d ed. Reading, MA: Addison-Wesley.

1

Introduction to State Transition Diagrams

A state transition diagram (STD) is a graphical representation of a finite state machine (FSM). By definition, an FSM exists in exactly one state at any given time. Transitions to other states are based solely on combinatorial logic that takes the states and inputs into account. In MATRIXx, STDs are deterministic.

Note This manual is not intended to provide a comprehensive discussion of FSMs. For information about this topic, refer to the documents listed in the *Related Documentation* section of this manual.

In SystemBuild, an FSM consists of a specified number of discrete states, together with logic for transitioning between the states and producing outputs depending on block inputs, the states, and the transitions. Each state in an FSM is graphically rendered as a bubble, rather than a block. You create bubble diagrams by using the STD Editor window.

You can obtain an STD block from the **References** palette of the **Blocks** view. In many respects, this block behaves like other SystemBuild blocks; in fact, it is stored and manipulated as a separate SystemBuild diagram just like a SuperBlock. The STD block is an interface between an FSM and a SuperBlock diagram. For more information about these aspects of SystemBuild, refer to the *SystemBuild User Guide*, available by selecting **Help»MATRIXx Bookshelf**.

The STD Block in the SuperBlock Editor

In the SuperBlock Editor window, an FSM is integrated into a block diagram as an STD block. This block is available on the **References** palette of the **Blocks** view. If you do not see this view, select **View»Blocks** to display this view.

You can drag an STD block from the palette into a block diagram and connect it to other blocks. This action creates a reference to an STD. You can edit the reference properties and the properties of the STD itself by using different dialog boxes. The STD block can be simulated in the same manner as any other block.

Note You can place an STD block within only discrete or procedure SuperBlocks. For information about these types of SuperBlocks, refer to Chapter 6, *SuperBlock Timing and Transformation*, of the *SystemBuild User Guide*.

Figure 1-1 shows the default STD block icon.

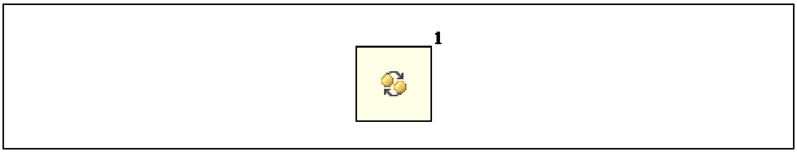


Figure 1-1. The STD Icon in the SuperBlock Editor Window

For more information about the STD block, enter `help std` in the Xmath command area.

The Finite State Machine and the STD Editor Window

The STD block acts as an interface between the SystemBuild block diagram and an FSM, which is created and edited in the STD Editor window. While the SuperBlock Editor produces block diagrams, the STD Editor window produces bubble diagrams. In the STD Editor window, each state is represented by a circular bubble. Throughout this document, the term *bubble* is used interchangeably with the term *state* to refer to the circular symbol on the screen. The lines that connect bubbles are called *transitions* or *transition lines*; they can be rendered as straight or curved lines in the diagram.

The STD Editor window supersedes the SuperBlock Editor window whenever you create a new STD from the **Solution Explorer** view or double-click an STD block to view its contents. Like a top-level SuperBlock, an STD can be referenced by name. Multiple instances of an STD can occur within a SystemBuild model.

Finite State Machine Behavior

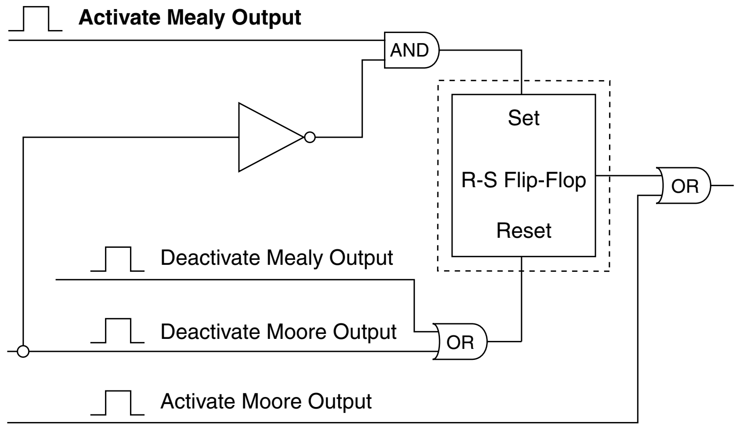
At every computation cycle, an FSM evaluates the current state based on a list of logical inputs. The FSM then uses a list of prioritized transition conditions to move to a new—but not necessarily a different—state. MATRIXx assigns transition priorities based on the order in which you create the transitions, but you can change this priority. For information about this action, refer to the *Setting Transition Priorities* section of Chapter 3, *Using the STD Editor Window*.

In every state, the machine may set or reset a list of logical output signals which are called the Moore mode outputs. Also, while transitioning, the FSM may activate or deactivate another list of logical outputs, called Mealy mode outputs. If a transition and state both activate the same output in different ways, the state activation takes precedence.

Moore and Mealy Modes

Each STD output is identified by an integer. The value of each output is zero by default, but an output can be asserted (set to 1) if its number is stated in a bubble (Moore mode) or transition (Mealy mode). How the output is turned off depends on whether the output was asserted in the Moore or Mealy mode.

Figure 1-2 illustrates the differences between Mealy and Moore modes.



Moore Output	Mealy Output
Activating a Moore output makes the output TRUE for the duration of the state.	Activating a Mealy output makes the output TRUE until deactivated.
Deactivating a Moore output forces the output to FALSE.	Deactivating a Mealy output makes the output FALSE unless otherwise activated.

Figure 1-2. Moore and Mealy Mode Finite State Machine Outputs

The difference between Moore and Mealy mode outputs is that Moore outputs appear on the bubbles, whereas Mealy outputs appear on the transitions.

In the Moore mode, outputs are updated whenever a new state is entered and remain stable as long as the system remains in that state. Thus, no explicit resets are usually required for Moore outputs. Resetting is allowed; if used, it overrides the effect of any input that may try to force the output to TRUE.

In Mealy mode, outputs are updated on the transitions between states. If an output is activated in Mealy mode, it remains activated until it is explicitly reset by the action of a transition or state. Figure 1-2 depicts these ideas and shows how the Moore and Mealy outputs are combined to produce the STD outputs.

An output may be activated (set to TRUE) in either the Mealy or Moore mode; after it is activated in one mode or the other, it remains in that mode until deactivated by the method appropriate to the mode.

- An output in Moore mode is reset when the bubble by which the output is set transitions to another bubble.
- An output in Mealy mode remains set until it is explicitly reset by another Mealy or Moore mode.

Later, the output may be activated and deactivated in the other mode.

Example

The following example makes use of the Cruise Control model. The model contains an STD with the outputs set in Moore Mode, For the second half of this example, you must convert the same model to use Mealy Mode outputs. You can simulate the two STDs with a suitable input matrix to verify that the outputs are the same.

Complete the following steps to open this example SystemBuild model.

1. Select **File»Open»SystemBuild Model**.
2. Navigate to the `matrixx\mx_`*`xx`*`\sysbld\examples\auto\` directory, where *`xx`* is the version of MATRIXx you installed.
3. Select `cruise_d.sbd-mtx` and click **Open**. MATRIXx displays the contents of the project in the **Solution Explorer** view. If you do not see this view, select **View»Solution Explorer**.
4. In this view, expand the **cruise_d** project, the **Diagrams** folder, and the **cruise_d** subfolder.
5. Double-click the Mode Selection State Diagram. MATRIXx displays the contents of this STD in the STD Editor window.

This STD has the following four inputs:

- **U1**—Set (cruise control on)
- **U2**—Resume (cruise control on)
- **U3**—Disengage (cruise control on)
- **U4**—Brake

Complete the following steps to display these inputs:

1. Double-click an empty space in the STD Editor window to launch the **StateTransition Properties** dialog box.
2. On the **Inputs** tab, select **Cruise Control System.33** from the **Signals Instance Path** pull-down list. Notice the labels of the signals.
3. To see these inputs on the block diagram, click **Cancel** and navigate to the parent SuperBlock by selecting **Navigation»Parents» cruise_d\Cruise Control System**.

The two outputs from the STD initiate the auto speed logic and indicate whether the cruise control is on or off.

Moore Mode

Figure 1-3 shows the diagram with the **Name Format** set to **Name & Outputs** for each bubble, and the **Name Format** set to **Name & Cond & Outputs** for each transition.

The initial bubble in the diagram is bubble 1, **Cruise Control Off**. The set signal {**U1**} is asserted to get the system running. The **Set** transition (1) forces the diagram to the **Set Speed** bubble (2), and the bubble asserts the set speed output signal [1] as a Moore mode output. As long as set input signal remains TRUE, the diagram keeps looping in bubble 2 because this transition has priority over the other one. When the set signal goes FALSE, the diagram transitions through the transition line labeled **True** to **Cruise Control On**, bubble 3. This state asserts the cruise control on output signal [2] and keeps it on until either the disengage signal {**U3**} or brake signal {**U4**} evaluates to TRUE. This forces the diagram back to **Cruise Control Off** to wait for the resume external input {**U2**} to force the transition back to **Cruise Control On**.

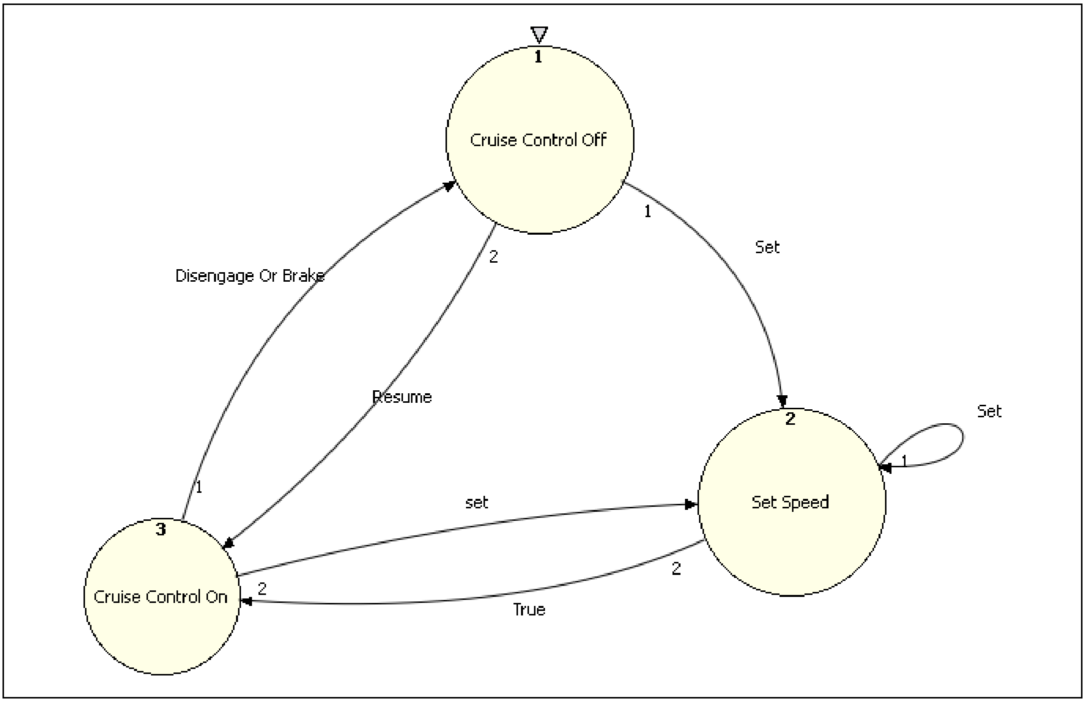


Figure 1-3. STD with Moore Mode Outputs Displayed

Mealy Mode

Figure 1-4 shows the same logic as Figure 1-3 using Mealy mode outputs. You can create this diagram from the model provided.

At the transition from bubble 1 (labeled **Set**), output 1 is asserted using the Mealy mode. This transition places the diagram in the **Set Speed** bubble, where it remains until the set input signal {**U1**} goes FALSE. Then the diagram takes transition priority 2, Turn Set Off CC On, which sets the set speed output signal off [–1] and the cruise control signal on [2]. At a later time, the disengage signal {**U3**} or the brake signal {**U4**} may reset the cruise control on output signal to off [–2] and return the diagram to bubble 1. From there, the resume signal {**U2**} may restart the cruise control on [2] output and return the diagram to the **Cruise Control On** state. Likewise, if the set input signal {**U1**} goes TRUE when the diagram is in the **Cruise Control On** bubble, the state transitions to the **Set Speed** bubble and resets the set speed output signal to TRUE [1] and turns off the cruise control on output signal [–2].

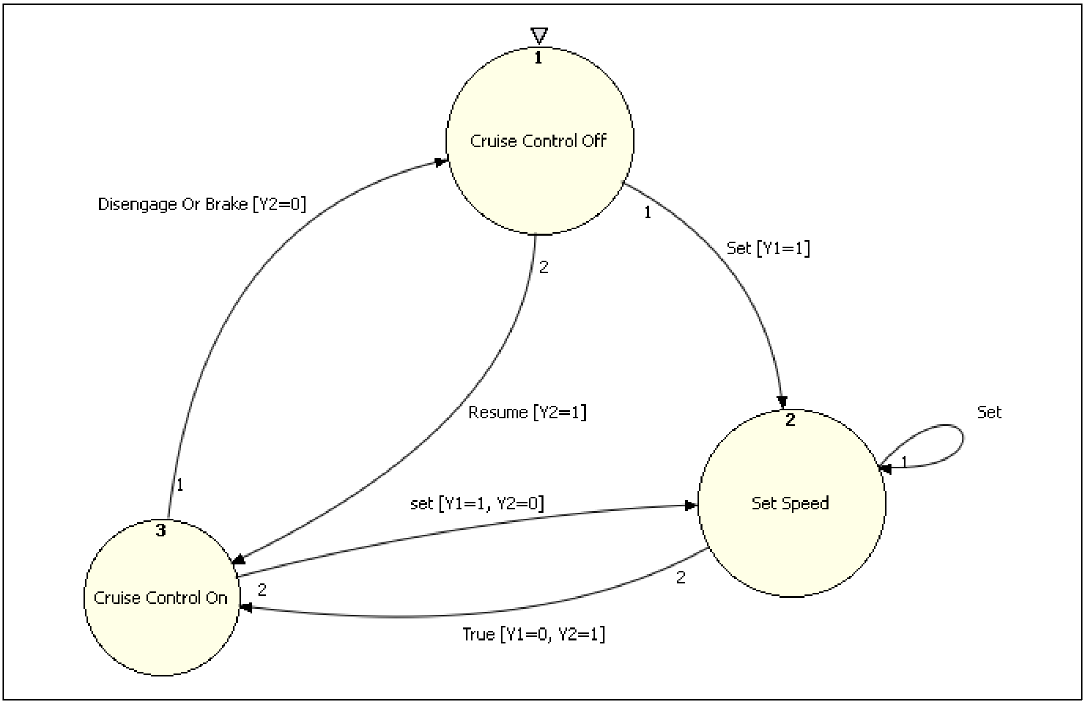


Figure 1-4. STD with Mealy Mode Outputs Displayed

2

Creating, Editing, and Saving STDs

You can create a state transition diagram (STD) by using any of the following methods:

- The **Solution Explorer** view
- The SuperBlock Editor window
- SystemBuild Access (SBA)

This chapter provides information about using each of these methods. This chapter also describes how to edit and save an STD.

Creating an STD from the Solution Explorer View

Complete the following steps to create an STD from the **Solution Explorer** view.

1. Load a project in the **Solution Explorer** view. If you do not see this view, select **View»Solution Explorer** to display this view.
2. Right-click the **Diagrams** folder and select **New Diagram»State Transition**. You can also perform this action on any subfolder of the **Diagrams** folder.

After you complete these steps, MATRIXx displays an empty STD Editor window. For information about using this window, refer to Chapter 3, *Using the STD Editor Window*.

Creating an STD from the SuperBlock Editor Window

Complete the following steps to create an STD from the SuperBlock Editor window.

1. Open any discrete or procedure SuperBlock.
2. Click the **References** palette in the **Blocks** view. If you do not see this view, select **View»Blocks** to display this view.

3. Click the StateTransition block, move the mouse over the block diagram, and click again to place the block on the block diagram.
4. Right-click the StateTransition block and select **Properties** to launch the **State Diagram Properties** dialog box.
5. Give the STD a unique **Reference Name**.
6. Click **OK** to save changes and return to the block diagram.

   These steps result in a reference to an empty STD. The next step is to create the STD that this reference calls.
7. Right-click the STD block and select **Open Definition** to display the STD Editor window. For information about using this window, refer to Chapter 3, *Using the STD Editor Window*.

Note States 1 through n_0, where n_0 is the number of outputs, contain the logical outputs of the state machine. At initialization, the states and initial outputs are zero by default, but any of them can be set from the **State Diagram Properties** dialog box. This activates the output as a Mealy output at initialization time, but the Moore mode outputs of the initial bubble setting override these outputs. State $n_0 + 1$ is an integer, the most recent bubble number. State $n_0 + 2$ is another integer that is not used.

For information about how SystemBuild handles references to objects, refer to the *Creating and Manipulating SuperBlock References* section of Chapter 4, *SuperBlocks*, in the *SystemBuild User Guide*. This guide is available by selecting **Help»MATRIXx Bookshelf**.

Creating an STD Using SBA Commands

SystemBuild Access (SBA) provides a complete set of commands for creating STDs using a combination of the SuperBlock Editor window and the STD Editor window. To include an STD in a model, you must have the following elements:

- An STD in the project. To create an STD, use the `CREATESTD` command. Use `CREATEBUBBLE` to create a bubble, and use `CREATETRANSITION` to create each transition.
- A discrete parent SuperBlock. Use `CREATESUPERBLOCK` to create the parent SuperBlock, making sure to specify `{type="discrete"}`.
- An STD block in the discrete SuperBlock that references the STD. While editing a discrete SuperBlock, use `CREATEBLOCK` to create a block of type STD, and use the `name` keyword to reference an existing STD in the project.

For information about any of these commands, enter `help` *`command_name`* in the Xmath command area, where *`command_name`* is the name of the command for which you want to view help. For information about SBA, enter `help sba` in the Xmath command area.

Editing STD Properties

To edit the properties of an STD, use the **Selection** tool with any of the following methods:

- Double-click an empty area in the STD Editor window.
- Right-click an empty area in the STD Editor window and select **Properties**.

These actions launch the **StateTransition Properties** dialog box. You use this dialog box to configure the inputs and outputs of the STD. For information about the **StateTransition Properties** dialog box, click the **?** button in the title bar of this dialog box.

Note The **State Diagram Properties** dialog box is different from the **StateTransition Properties** dialog box. You use the former to edit the properties of an STD reference. You use the latter to edit the properties of the STD that reference calls.

Saving an STD

MATRIX handles STDs in the same way that MATRIXx handles SuperBlocks. Saving an STD involves the following two-step process:

1. With the STD Editor window open, select **State Transition»Update into Project**. This action updates the `.sbd-mtx` savefile associated with the STD.
2. Save the project.

For information about savefiles, enter `help savefile` in the Xmath command area.

3

Using the STD Editor Window

After you open the state transition diagram (STD) block, MATRIXx displays the STD Editor window. You use this editor to define states, or bubbles, and SuperBubbles. You also use this window to define the transitions between states.

This chapter provides information about using the STD Editor window to create and modify bubbles, transitions, and SuperBubbles.

Creating and Editing Bubbles

A bubble represents a state. After you launch the STD Editor window, you can use either of the following methods to create a bubble.

- Select **View»Bubbles** to display the **Bubbles** view. Click the Bubble, move the mouse over an empty space in the STD Editor window, and click again to place the bubble in the window.
- Right-click an empty area in the STD window and select **Add Bubble»Bubble**. Then, click the mouse to place the bubble in the window.

For information about changing the properties of a bubble, refer to the *Changing Bubble Properties* section of this chapter.

Defining the Initial Bubble

The initial bubble is the bubble that MATRIXx executes first. All STDs must have an initial bubble. By default, the initial bubble is the first bubble you create. To define another bubble as the initial bubble, right-click that bubble and select **Set as Initial Bubble**.

For example, in Figure 3-1, the initial bubble is defined as **State 2**.

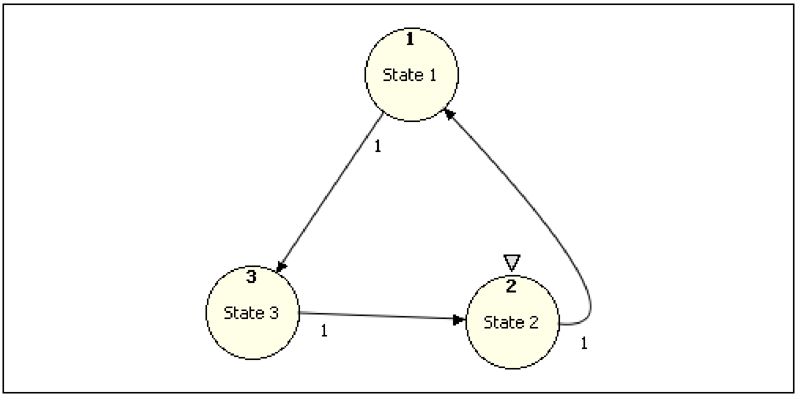


Figure 3-1. An STD where State 2 is the Initial Bubble

Identifying Bubbles

Every bubble is assigned an identification number by the STD Editor window. The ID number, which is initially based on the order of creation, appears inside each bubble. When you delete a bubble, its ID number is returned to the pool, and it may be reused under these circumstances. You can change a bubble ID number by accessing the **Bubble Properties** dialog box, which the following section describes.

Changing Bubble Properties

You use the **Bubble Properties** dialog box to enter or review information about a selected bubble. You can use either of the following methods to launch this dialog box.

- Double-click a bubble.
- Right-click a bubble and select **Properties**.

The following sections describe some common tasks you can complete by using this dialog box. For more information about this dialog box, click the **?** button in the title bar.

Moore Outputs and Moore Description

The **Bubble Properties** dialog box contains text boxes that relate to Moore mode. For each bubble, the number of **Moore Outputs** defines the number of possible entries in the **Moore Description** text box. For information about Moore mode, refer to the *Moore and Mealy Modes* section of Chapter 1, *Introduction to State Transition Diagrams*.

To specify that an output is to be asserted (set = 1), enter the pin number of the output on the STD block in the **Moore Description** text box.

To specify that an output is to be turned off (set = 0), place a – sign before the pin number on the STD block in the **Moore Description** text box. The order in which you place the outputs in the **Moore Description** text box is irrelevant.

For example, consider the following **Moore Description** text box:

```
4, -6, 10
```

These values indicate that STD outputs 4 and 10 are asserted while STD output 6 is turned off.

Name Format

Use the **Name Format** pull-down list to specify the information to be displayed about this bubble. If you give a name to a bubble, the name is always displayed on the bubble, but you may control how Moore mode outputs are displayed by selecting either **Name** or **Name & Outputs**.

Creating and Editing Transitions

Transitions allow you to specify the Mealy output conditions between states. You can use self-looping transitions to activate an output or a delay. This section provides information about creating and editing transitions between bubbles.

Creating Transitions

You use the **Connection** tool to create transitions. If you do not see the SystemBuild tools, select **View»Tools** to display these tools. Then, click the **Connection** button to activate the **Connection** tool.

Note You can also press <Tab> to cycle between the tools.

After you select this tool, complete the following steps to create a transition between two bubbles.

1. Click the bubble at which the transition starts. Notice that when you move the mouse now, the transition line trails the mouse cursor.
2. (Optional) Click an empty area in the STD Editor window to create the point at which the transition line bends.
3. Click the bubble at which the transition ends. This bubble can be the same bubble as the bubble you selected in step 1.

After you complete these steps, MATRIXx updates the STD Editor window to display a transition line between the two bubbles.

Editing Transitions

You use the **Transition Properties** dialog box to edit transitions. To launch this dialog box, use the **Selection** tool to double-click a transition line. You can also right-click a transition line and select **Properties**.

Mealy Outputs and Mealy Description

The **Transition Properties** dialog box contains text boxes that relate to Mealy mode. For each transition, the number of **Mealy Outputs** defines the number of possible entries in the **Mealy Description** text box.

To specify that an output is to be asserted (set = 1), enter the pin number of the output on the STD block in the **Mealy Description** text box.

To specify that an output is to be turned off (set = 0), place a – sign before the pin number on the STD block in the **Mealy Description** text box. The order in which you place the outputs in the **Mealy Description** text box is irrelevant.

For example, consider the following **Mealy Description** text box:

```
20, –2, 11
```

These values indicate that STD block outputs 20 and 11 are asserted while STD block output 2 is turned off.

Activation Conditions

Each **Transition Properties** dialog box has an **Activation Condition** field that must be filled in. Data entered into this field determines the events that cause the transition to be executed.

Activation Condition field entries are logical functions, just like those that form the right-hand sides of logical expressions in the SystemBuild LogicalExpression block. For information about these expressions, enter `help equations` in the Xmath command area.

The logical function accepts input signals of the STD as variables and evaluates them according to the condition(s) you define in the **Activation Condition** field. Each input channel can be a different data type, but input channel usage must be consistent across all transitions. This restriction implies that a logical input channel cannot also be used in a numeric expression, and a numeric input channel cannot also be used in a logical expression.

The inputs are labeled **U1** through **U*n***, where ***n*** is the number of inputs to the STD, and the inputs are numbered from top to bottom on the input side of the STD block. In general, as long as the function may be evaluated as TRUE or FALSE, any conventionally valid combination of relational operators linking arithmetic and logical expressions may be used. The properties of these operations are as follows:

- **Arithmetic Expressions**—Use conventional combinations of the operations `**`, `*`, `/`, `+`, and `-`. Parentheses are used for grouping. Expressions and parentheses are used to compute a number, which can be compared with another using relational operators.
- **Relational Operators**—Used to compare numbers or, in the case of Equal to and Not Equal to operators, logical values.

  The following table shows the relational operators in both textual and symbolic formats.

Text	Symbol	Meaning
`lt`	`<`	Less than
`gt`	`>`	Greater than
`ge`	`>=`	Greater than or equal to
`le`	`<=`	Less than or equal to
`eq`	`=`	Equal to
`ne`	`<>`	Not equal to

- **Logical Expressions**—Built up from inputs or relational comparisons using any conventional combination of **NOT**, **AND**, **NAND**, **OR**, **NOR**, **EQV**, and **NEQV**. The order of precedence for evaluation also follows usual conventions: arithmetic, relational, and then logical. For logical relations the order is **NOT**, **AND/NAND**, **OR/NOR**, **EQV/NEQV**.

The following table shows typical logical relations.

Relations	Action
U1 le U2	Arithmetic compare
Not(U3 or U4)	Logical compare
True	Unconditional
U5 gt 0	Compare a literal
U62 + 1 >= u7**	Evaluate, then compare
U8 <> 0	Logic test

Within the active bubble, each of the transitions is evaluated in the prioritized order until a condition is detected as TRUE. Then the state machine transitions to the new bubble pointed to by the particular transition. The Mealy outputs from the transition are set, and the Moore outputs from the new bubble are asserted.

Setting Transition Priorities

Transition priorities are assigned consecutively as you create transitions. The STD Editor window displays transition priorities near the starting point of each transition line. Lower numbers indicate higher priority, with 1 being the highest-priority transition.

You can change transition priorities by using the **Bubble Properties** dialog box of the bubble at which that transition starts. When you change a transition priority, all other transitions from that bubble are automatically updated to ensure that the priorities form a list of consecutive integers from 1 to the number of transitions.

Complete the following steps to change the priority of a transition.

1. Double-click the bubble from which that transition originates. This action launches the **Bubble Properties** dialog box.
2. Click the **Transitions** tab, which displays all outgoing transitions from the bubble.

3. Enter the new priority in the **Priority** column of the row that applies to that transition.
4. Click **OK** to save changes and return to the block diagram.

Moving and Bending Transition Lines

As described in the *Creating and Editing Transitions* section of this chapter, you can create either bent or straight transitions. You can bend a straight transition, and you can move transitions between bubbles. This section provides information about these tasks.

Before you edit a transition, you must select it. To select a transition, click the transition line once with the **Selection** tool. The transition handles appear, as shown in Figure 3-2.

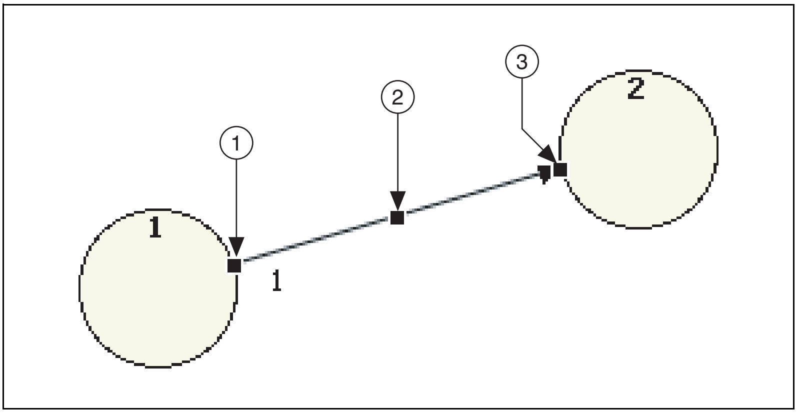


Figure 3-2. A Straight Transition with Transition Handles

To bend the transition, move the mouse over the middle handle (2) and drag this handle to a new location. Release the mouse button to accept changes.

You can also move the transition itself. To change the endpoint of a transition, drag the handle closest to the transition arrow (3). To change the starting point of a transition, drag the handle at the beginning of the transition (1).

Note When two straight transitions go between the same two states, MATRIXx displays the transition as a single double-headed transition connection line.

Creating and Editing SuperBubbles

You use SuperBubbles to organize groups of bubbles within the STD Editor window. These groups are for visual purposes only and do not affect the behavior of an STD.

Creating a SuperBubble

You can create a SuperBubble by using either of the following methods:

- Select **View»Bubbles** to display the **Bubbles** view. Click the SuperBubble icon, move the mouse over an empty space in the STD Editor window, and click again to place the SuperBubble in the window.

  Instead of clicking to place the SuperBubble, you can click and drag to define a SuperBubble of arbitrary size. Any bubbles within the borders of the SuperBubble become part of the SuperBubble.
- Right-click an empty area in the STD window and select **Add Bubble»SuperBubble**. Then, click the mouse to place the SuperBubble in the window. You can also click and drag to define a SuperBubble of arbitrary size.

Note You can create SuperBubbles within SuperBubbles.

Adding Bubbles to a SuperBubble

Complete the following steps to add bubbles to a SuperBubble.

1. Click the SuperBubble to display resizing handles.
2. Drag the resizing handles until the SuperBubble is the size you want.
3. Drag bubbles inside the borders of the SuperBubble.

Viewing a SuperBubble

You can display a SuperBubble in expanded or collapsed form. In expanded form, the SuperBubble is transparent and displays the bubbles it contains. In collapsed form, the SuperBubble is opaque. Toggle this option by right-clicking the SuperBubble border and selecting **Expand**.

Figure 3-3 shows an expanded SuperBubble that contains bubbles **State 2** and **State 3**.

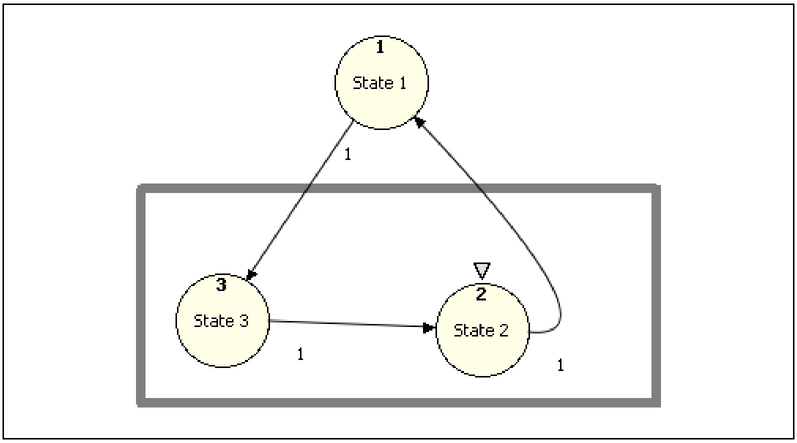


Figure 3-3. An STD with an Expanded SuperBubble

Figure 3-4 shows how this SuperBubble appears when collapsed.

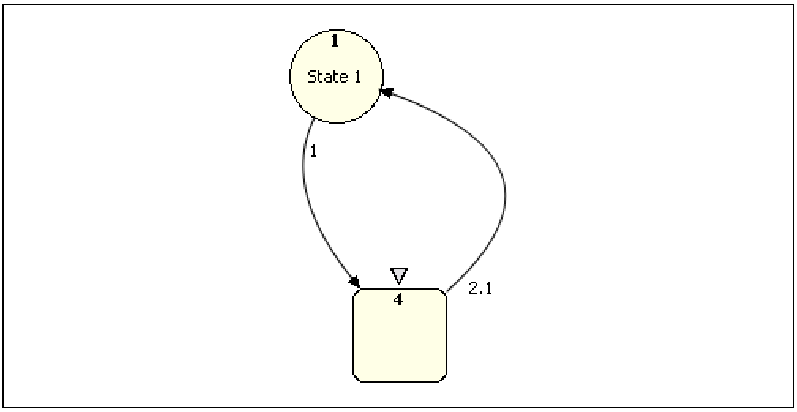


Figure 3-4. An STD with a Collapsed SuperBubble

In Figure 3-4, notice that the STD Editor window displays transitions to and from bubbles within SuperBubbles. These transitions appear to come from the SuperBubble itself. Similarly, if a collapsed SuperBubble contains the initial bubble, that SuperBubble displays the initial marker on the SuperBubble itself.

Editing SuperBubble Properties

To edit the properties of a SuperBubble, right-click that SuperBubble and select **Properties**.

Removing a SuperBubble

To remove a SuperBubble without deleting the bubbles inside it, right-click the SuperBubble and select **Remove Structure**.

To remove a SuperBubble and delete the bubbles inside it, right-click the SuperBubble and select **Delete**. You can also select the Super Bubble and press <Delete>.

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at `ni.com` for technical support and professional services:

- **Support**—Online technical support resources at `ni.com/support` include the following:
  - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Application Engineers worldwide in the NI Discussion Forums at `ni.com/forums`. National Instruments Application Engineers make sure every question receives an answer.

    For information about other technical support options in your area, visit `ni.com/services` or contact your local office at `ni.com/contact`.
- **Training and Certification**—Visit `ni.com/training` for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit `ni.com/alliance`.

If you searched `ni.com` and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Index