

# ***MATRIXx™***

---

## **SystemBuild™ State Transition Diagram Block User Guide**

## **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 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,  
Canada (Calgary) 403 274 9391, Canada (Ottawa) 613 233 5949, Canada (Québec) 450 510 3055,  
Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530, China 86 21 6555 7838,  
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11,  
France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427, India 91 80 51190000,  
Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400,  
Malaysia 603 9131 0918, Mexico 001 800 010 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 7 095 783 68 51, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,  
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227,  
Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to [techpubs@ni.com](mailto:techpubs@ni.com).

© 2000–2004 National Instruments Corporation. All rights reserved.

# 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 THEREOF 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

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

## Trademarks

MATRIX<sup>™</sup>, National Instruments<sup>™</sup>, NI<sup>™</sup>, ni.com<sup>™</sup>, SystemBuild<sup>™</sup>, and Xmath<sup>™</sup> are trademarks of National Instruments Corporation.

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](http://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.

# Conventions

---

The following conventions are used in this manual:

{ } Braces denote the optional arguments or keywords in Xmath syntax. For example: the disengage signal {U3}.

[ ] Square brackets enclose optional items—for example, [response].

» 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.

**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. This font 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.

# Contents

---

## Chapter 1

### State Transition Diagrams

Introduction to State Transition Diagrams.....	1-1
The STD Block in the SuperBlock Editor.....	1-1
The Finite State Machine and the STD Editor .....	1-2

## Chapter 2

### Finite State Machine Behavior

Moore and Mealy Modes.....	2-1
Examples.....	2-3
Moore Mode .....	2-3
Mealy Mode.....	2-5

## Chapter 3

### Creating and Editing an STD

Creating an STD from the Catalog Browser.....	3-1
Creating an STD from the SuperBlock Editor.....	3-2
Creating an STD Using SBA Commands.....	3-3

## Chapter 4

### Using the STD Editor

Creating Bubbles.....	4-2
Creating and Changing the Initial Bubble .....	4-2
Identifying Bubbles .....	4-3
Moving Bubbles .....	4-3
Providing Bubble Properties.....	4-4
Moore Outputs and Moore Description .....	4-4
Color.....	4-5
Diagram Option.....	4-6
Creating Transitions.....	4-6
Providing Transition Properties.....	4-6
Mealy Outputs and Mealy Description .....	4-7
Activation Conditions .....	4-8
Diagram Option.....	4-10
Editing Transition Lines .....	4-10

## **Chapter 5**

### **SuperBubbles**

Creating a SuperBubble.....	5-1
Navigating a Bubble Hierarchy .....	5-2
Returning the Contents of a Subhierarchy to the Parent Page.....	5-2
Examples .....	5-3

## **Appendix A**

### **Technical Support and Professional Services**

### **Index**

---

# State Transition Diagrams

This manual supports the State Transition Diagram (STD) block, which is a separately licensed block in SystemBuild, a product in the MATRIXx family.

## Introduction to State Transition Diagrams

---

A state transition diagram is a graphical representation of a finite state machine. By definition, a state machine exists in exactly one state at any given time, and transitions to other states are based solely on combinatorial logic that takes the states and inputs into account. In SystemBuild, a finite state machine 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 a finite state machine is graphically rendered as a bubble, rather than a block. You create bubble diagrams with the STD Editor.

You can obtain an STD block from the SuperBlocks palette of the SystemBuild Palette Browser. In many respects it behaves like other SystemBuild blocks; in fact, it is stored and manipulated as a separate SystemBuild catalog entity just like a SuperBlock. The STD block is an interface between a finite state machine and a SuperBlock diagram. For more information about these aspects of SystemBuild, refer to the *SystemBuild User Guide*.

## The STD Block in the SuperBlock Editor

In the SuperBlock Editor, a finite state machine is integrated into a block diagram as an STD block. You can drag an STD block from the palette into a block diagram and connect it to other blocks. You can edit its properties. It can be simulated in the same manner as any other block.

The default STD icon is shown on the left in Figure 1-1. You can display the STD icon with the diagram expanded and visible, but shrunk to fit inside the icon, as shown on the right. This icon represents the Model Selection State Diagram, which is part of the Cruise Control model introduced in Chapter 2, *Finite State Machine Behavior*.

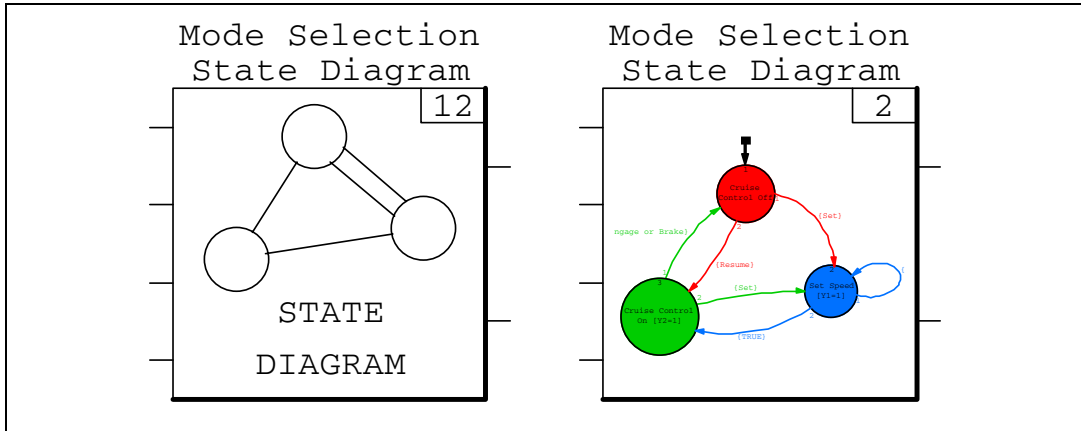


Figure 1-1. STD Icons in the SuperBlock Editor

To change the icon type, complete the following steps:

1. Select the STD block.
2. Select **User** or **Alternate** from the Icon Type pull-down menu on the SuperBlock Editor icon bar.

## The Finite State Machine and the STD Editor

The STD block acts as an interface between the SystemBuild block diagram and a finite state machine, which is created and edited in the STD Editor. While the SuperBlock Editor produces block diagrams, the STD Editor produces bubble diagrams. In the STD Editor, 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 supersedes the SuperBlock Editor whenever you create a new STD from the Catalog Browser or double-click an STD block to view its contents. The STD Editor is similar in appearance to the SuperBlock Editor, but the icon bar is different. For information on the different menus and icons, select **Help»Topics** from the STD Editor.

After an STD is defined, it appears in a separate hierarchy within the SystemBuild Catalog Browser on the same level as top-level SuperBlocks. 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, a finite state machine evaluates the current state based on a list of logical inputs and progresses, depending on a list of prioritized transition conditions, to a new—but not necessarily a different—state. In every state, the machine may set or reset a list of logical output signals which are called the Moore mode outputs. Also, while transitioned it may activate or deactivate another list of logical outputs, called Mealy mode outputs.

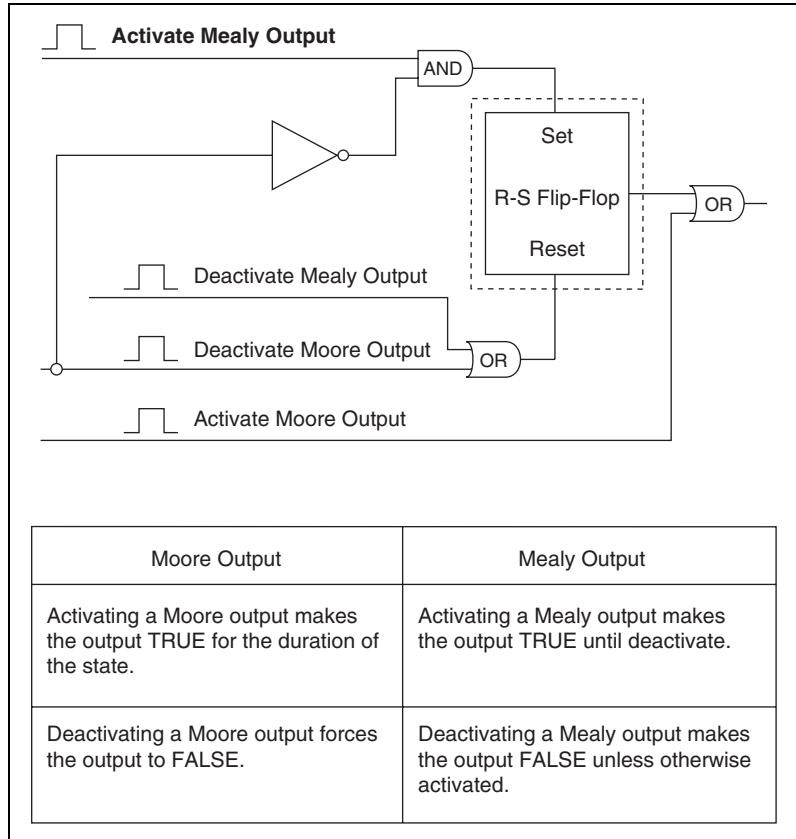
## Moore and Mealy Modes

---

Each STD output is identified by an integer. An output is zero by default, but it can be asserted (set to 1) if its number is stated in a bubble (Moore mode) or transition (Mealy mode). Its output is turned off (set to zero) depending upon whether it was asserted in the Moore or Mealy mode. These modes are illustrated in Figure 2-1.



**Note** Figure 2-1 is merely an illustration of state behavior; it does not depict a SystemBuild diagram. SystemBuild does not have a Flip-Flop block.



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

The difference between Moore and Mealy mode outputs is apparent from the bubble diagram: Moore outputs appear on the bubbles, while 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 2-1 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.

- A Moore mode output is reset when the bubble state it is set by transitions to another bubble.
- A Mealy mode output 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.

## Examples

---

The following examples make use of the Cruise Control model. The model contains an STD that contains Moore modes, which we examine in the first example. You must convert the same example to Mealy modes for the second example. You can simulate the two STDs with a suitable input matrix to verify that the outputs are the same.

To view a model containing Moore modes, complete the following steps:

1. Load the file `$SYSBLD/examples/auto/cruise_d.cat` into SystemBuild.
2. Select the State Diagrams folder in the Catalog Browser.
3. Open the Mode Selection State Diagram STD in the STD Editor.

For this model, it is helpful to know what the external input signals mean.

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

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

## Moore Mode

Figure 2-2 shows the diagram with the Diagram Option set to **Name & Outputs** for each bubble, and the Diagram Option 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.

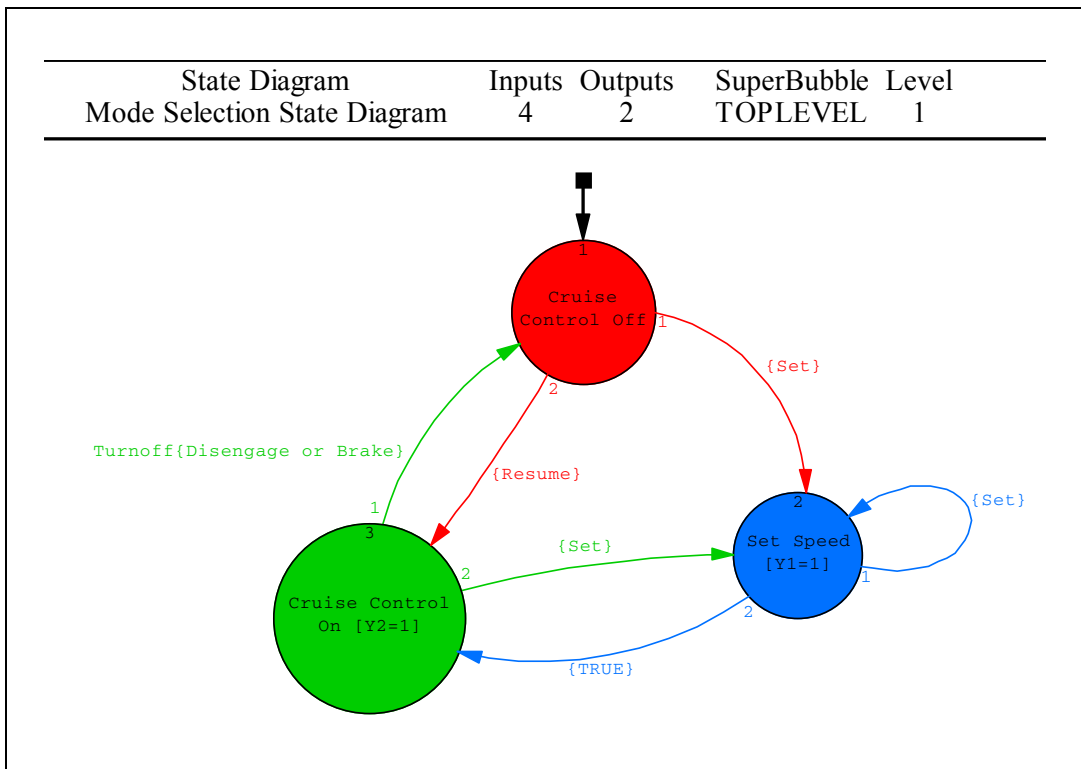
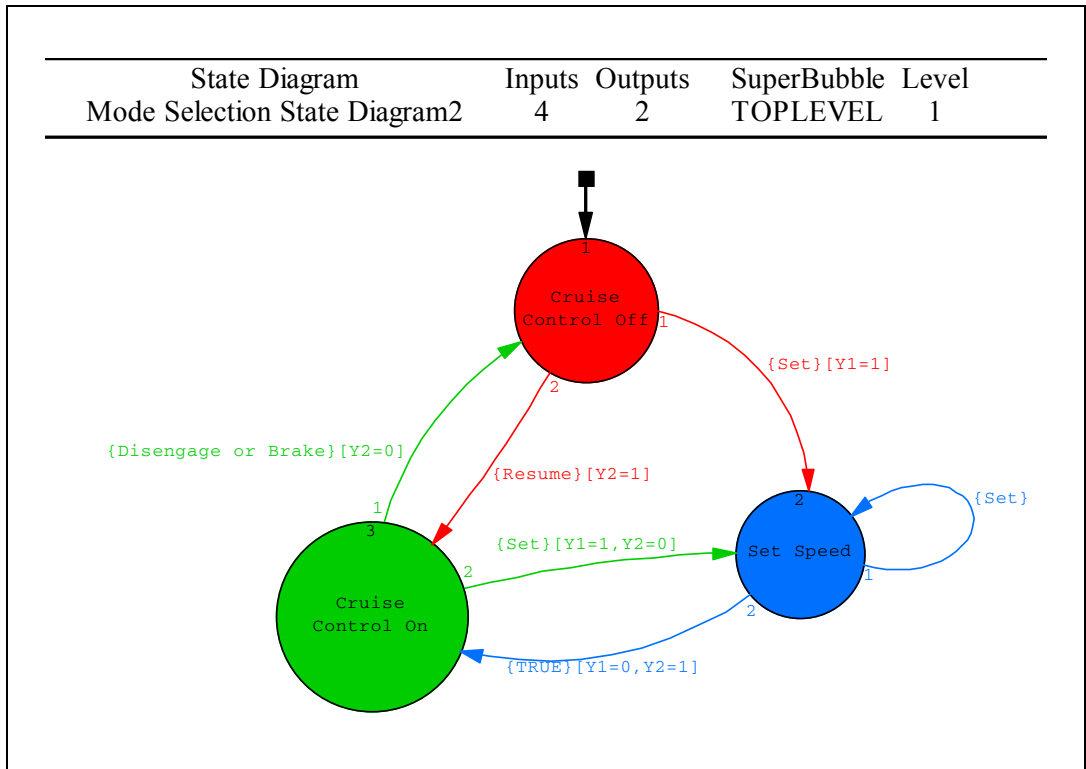


Figure 2-2. STD with Moore Mode Outputs

## Mealy Mode

Figure 2-3 shows the same logic as Figure 2-2 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 2-3.** STD with Mealy Mode Outputs

---

# Creating and Editing an STD

This chapter describes three different ways to create an STD:

- *Creating an STD from the Catalog Browser*
- *Creating an STD from the SuperBlock Editor*
- *Creating an STD Using SBA Commands*

If you use the Catalog Browser, you can define an STD by going directly to the STD Editor. If you start from the SuperBlock Editor, you must first create a block that acts as an interface between the finite state machine and the block diagram.

---

## Creating an STD from the Catalog Browser

To create an STD from the Catalog Browser, complete the following steps:

1. Select **File»New»STD**.

The STD Properties dialog box appears.

#	Higher Level Input Signal(s)	Input Label(s)
---	------------------------------	----------------

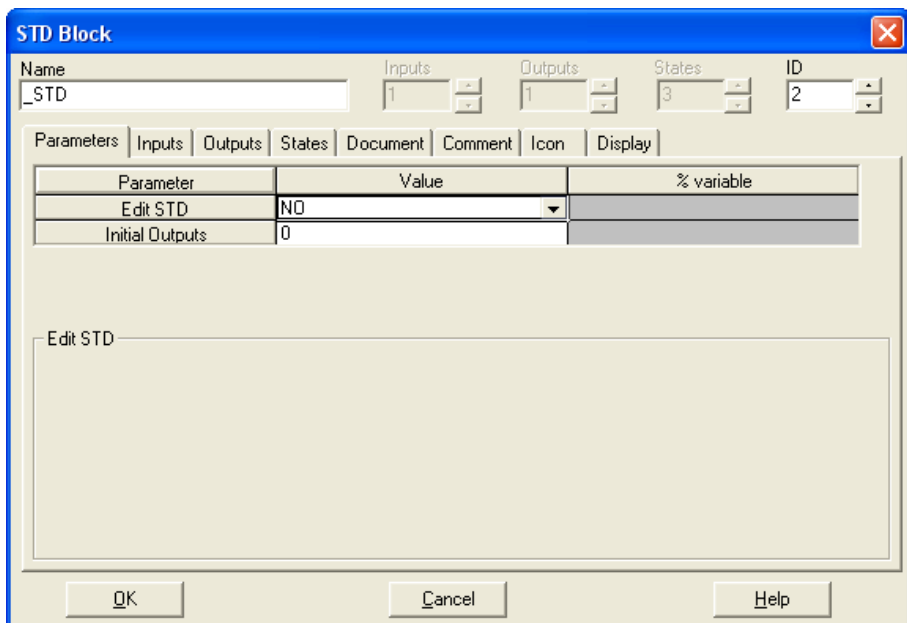
2. Enter a unique **Name** in the STD Properties dialog box and click **OK**.  
The STD Editor appears. Refer to Chapter 4, *Using the STD Editor*.

## Creating an STD from the SuperBlock Editor

In the SuperBlock Editor, an STD can be created only within a discrete SuperBlock. In the SuperBlock Editor, you first must create an STD block in a discrete SuperBlock.

To create an STD from the SuperBlock Editor, complete the following steps:

1. Edit any discrete SuperBlock.
2. Open the Palette Browser. From the SuperBlock palette, drag an STD block into the editor workspace.
3. When the icon appears in the block diagram, select it and press <Return> to view the dialog box.



4. Give the STD a unique Name. If necessary, change the number of inputs, outputs, and states.

References to STD blocks are made by name, the same as SuperBlocks. SystemBuild treats references to STD blocks that have

the same name and same number of inputs and outputs as references to the same STD block of code.

5. To create the STD this block will reference, open the STD Editor using one of the following methods:
  - On the **Parameters** tab, locate the Edit STD field. Click this field and drag down to select **Yes**. Click **OK**.
  - Double-click the STD block.

States 1 through  $n_0$ , where  $n_0$  = the number of outputs, contain the logical outputs of the state machine. At initialization, the states—and thus the initial outputs—are zero by default, but any of them can be set from the STD Block 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.

## 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 and the STD Editor. If you want to include an STD in a model, you must have the following elements:

- An STD (in the catalog)
 

To create an STD, use the `CREATESTD` command. Use `CREATEBUBBLE` to create a bubble for each state, 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
 

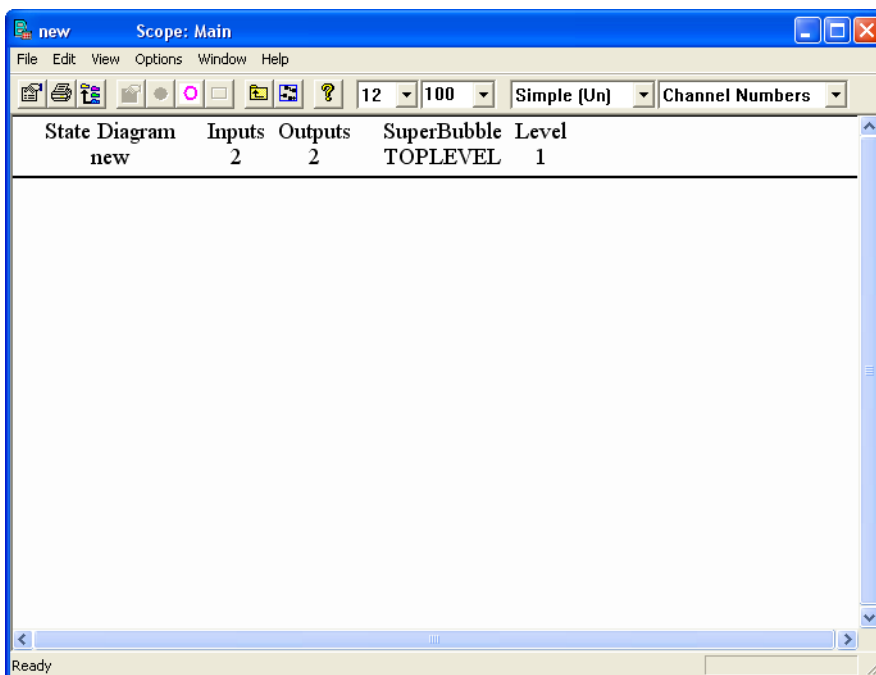
While editing a discrete SuperBlock, use `CREATEBLOCK` to create a block of type STD, and use the `name` keyword to reference an existing STD in the catalog.

Refer to the `CREATESTD` section of the *MATRIXx Help* for an example that performs each of these actions.



## Using the STD Editor

A full-screen STD Editor, shown in Figure 4-1, is provided for building state diagrams of arbitrary size.



**Figure 4-1.** Default STD Editor

The STD Editor resembles the SystemBuild Editor. It has File, Edit, View, Options, and Window menus but lacks the SuperBlock Editor Connect and Tools menus. Most functions for entering, editing, and manipulating bubble diagrams closely resemble the operations used for blocks in the SuperBlock Editor.

To get help on the STD Editor, select **Help»Topics**.

## Creating Bubbles

---

To insert a bubble on the screen, complete the following steps:

1. Select **Edit»New Bubble**.
2. Click the left mouse button in open space outside any bubble.

Each bubble represents a state, so you can define Moore outputs for each bubble.

Alternatively, move the cursor to an empty space and press <D> for Define bubble.

## Creating and Changing the Initial Bubble

The first bubble created in the diagram has a bold arrow at the top, as seen at bubble 1 in Figure 2-1, *Moore and Mealy Mode Finite State Machine Outputs*. This bubble is referred to as the *initial bubble*. At the time that the simulation or the application program is first executed, the finite state machine is started in the bubble state indicated by the initial bubble.

To change the initial bubble, complete the following steps:

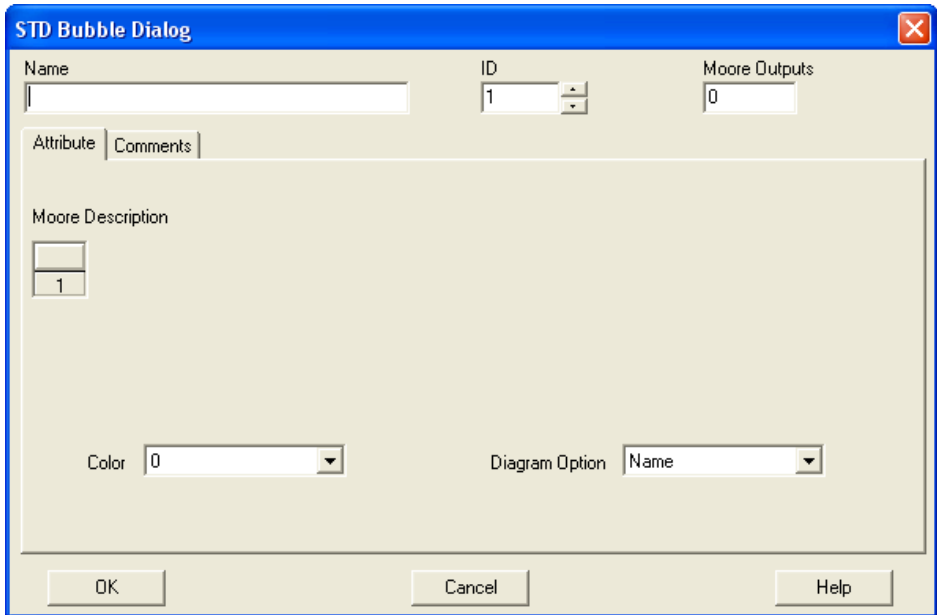
1. Select the bubble that you want to be the initial bubble.
2. Select **Edit»Initial Bubble**.

Each time a bubble is added to the diagram, the bubble appears on the screen.

To edit the bubble, complete the following steps:

1. Place the cursor over the bubble.
2. Press <Return> to raise the STD Bubble Dialog, as shown in Figure 4-2.

Refer to the *Providing Bubble Properties* section.



**Figure 4-2.** Default STD Bubble Dialog

## Identifying Bubbles

Every bubble is assigned an identification number by the STD Editor. The ID number, which is based on the order of creation, appears in small type inside each bubble. No provision is made for allowing you to change these numbers. However, when you delete a bubble, its serial number is returned to the pool, and it may be reused under these circumstances.

You can provide names for bubbles, but this is optional. A name is a string of alphanumeric characters. The first character must be an alpha character, and the length must not exceed 32 characters. Punctuation characters cannot be used in names.

## Moving Bubbles

To move a bubble, press and hold the left mouse button on the bubble and drag it to the new position.

To move a group of bubbles, complete the following steps:

1. Lasso or press <Ctrl> while clicking to highlight multiple bubbles.
2. Click and drag any of the selected bubbles to move the entire selection.

To cancel the selection of a group of bubbles, click the left mouse button in open space outside the selected group.

## Providing Bubble Properties

The STD Bubble Dialog allows you to enter or review information about a selected bubble.

To raise the dialog box, complete the following steps:

1. Select a bubble.
2. Press <Return>.

The dialog box shown in Figure 4-2 opens. In the dialog box, you can enter values for the name and the number of Moore Outputs. You can enter optional comments on the Comments tab.

3. To accept changes and return to the STD Editor, click **OK**. To reject changes and return to the STD Editor, click **Cancel**.

## Moore Outputs and Moore Description

The number of Moore Outputs specified affects the number of cells displayed in the Moore Description spreadsheet shown on the Attributes tab.

To specify that an output is to be asserted (set = 1), type the pin number of the output on the STD block in a cell in the Moore Description field.

To specify that an output is to be turned off (set = 0), type the negative of the pin number on the STD block in a cell.

Example 4-1 illustrates these methods.

### Example 4-1 Turning Pins On and Off

To turn pins 4 and 17 on but pin 6 off, complete the following steps:

1. Set the number of Moore Outputs to 3.
2. Type 4, -6, 17 in the Moore Description field.

The order in which you place the outputs in the field is irrelevant.

## Color

For color workstations, you may specify a color for each bubble. The color is specified through the Color combo box from which you can choose a number between -14 and 14. Table 4-1 shows the colors that appear on the bubble, any transitions that come from the bubble, and documentation information associated with the transitions. If the color number is positive, the color is applied as a fill; if negative, as an outline.

**Table 4-1.** Color Assignments

Number	Color
0	Black outline
1	Red
2	Yellow
3	Green
4	Blue
5	Magenta
6	Cyan
7	Orange
8	Pink
9	Yellow-green
10	Blue-green
11	Lt. Blue
12	Purple
13	Brown
14	Gray

On UNIX platforms, you can change default settings for the color assignments. Refer to the *SystemBuild User Guide* discussion on the SystemBuild resource file.

## Diagram Option

The Diagram Option field allows you 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 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.

To add a transition between two states using the mouse only, complete the following steps:

1. Click the middle mouse button in the sending bubble.
2. Click the middle mouse button in the receiving bubble.

SystemBuild draws the transition.

To add a transition between two states using the mouse and the Edit menu, complete the following steps:

1. Lasso the bubbles or select the sending bubble, and then press <Ctrl> while clicking the receiving bubble.  
Lassoing the bubbles does not give you the same control over the sending and receiving bubbles.
2. When the bubbles are selected, select **Edit»New Transition**.

To create a transition that loops back to the same state, complete the following steps:

1. Click the middle mouse button within the bubble.
2. Click again after a moment. Do *not* double-click.

## Providing Transition Properties

By default, transitions are on.

To view the handles (represented as boxes) on all transitions in an STD, select **Enable Options»Transition Markers**.

To view the STD Transition Dialog, select a transition marker and press <Return>.

The STD Transition Dialog opens. You can change information in this dialog box at any time.

All the dialog box entries are optional. The STD Editor accepts any or all entries at a later time.

You can assign a Name to a transition for documentation purposes.

By default, the From Bubble and To Bubble fields, which are identified by Bubble ID number, are determined by the graphical transitions already existing in the STD Editor. The From Bubble field is determined by the origin of the current transition and is read-only. You can change the To Bubble field in the dialog box, and the diagram changes accordingly.

You can enter a Comment for documentation purposes. Comments are included with the detail or documentation outputs for the STD.

## Mealy Outputs and Mealy Description

The number of Mealy Outputs specified controls the number of cells in the Mealy Description spreadsheet.

To specify that an output is to be asserted (set = 1), type the pin number of the output on the STD block in a cell in the Moore Description field.

To specify that an output is to be turned off (set = 0), type the negative of the pin number on the STD block in a cell.

Example 4-2 illustrates these methods.

#### Example 4-2 Turning Pins On and Off

To turn pins 11 and 20 on, but pin 2 off, complete the following steps:

1. Set the number of Mealy Outputs to 3.
2. Type 20, -2, 11 in the Mealy Description field.

The order in which you place the outputs in the field is irrelevant.

## Activation Conditions

Each transition dialog box has an Activation Conditions field that must be filled in. Data entered into the conditions field determines the events that cause the transition to be executed. Activation Conditions field entries are logical functions, just like those that form the right-hand sides of logical expressions in the SystemBuild LogicalExpression block. Refer to the *MATRIXx Help* for the LogicalExpression block, or type `help equations` in the Xmath Commands window.

The logical function accepts input signals of the STD as variables and evaluates them according to the relations that you express in the Activation Conditions field. Each input channel can be a different data type, but input channel usage must be consistent across all transitions. This 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 labelled **U1** through **Un**, 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. They 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 equals and not equals operators, logical values.



The relational operators are:

<b>lt</b>	or	<	Less than
<b>gt</b>	or	>	Greater than
<b>ge</b>	or	>=	Greater or equal
<b>le</b>	or	<=	Less than or equal
<b>eq</b>	or	=	Equals
<b>ne</b>	or	<>	Not equal

- 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
<b>U1 le U2</b>	Arithmetic compare
<b>Not(u3 or U4)</b>	Logical compare
<b>True</b>	Unconditional
<b>U5 gt 0</b>	Compare a literal
<b>U6**2 + 1 &gt;= u7</b>	Evaluate, then compare
<b>u8 &lt;&gt; 0</b>	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.

The default transition condition for a transition is TRUE. Always transition if this bubble is reached and this transition has priority. Transition priorities are assigned consecutively when the transitions are created. Priority 1 is highest. You can change this priority in the Transition dialog box. All other transitions from this bubble are automatically updated to ensure that the priorities form a list of consecutive integers from 1 to the number of transitions.

## Diagram Option

As with the STD Bubble Dialog, the Diagram Option field allows you to specify what information is to be displayed about each transition individually. If you give a name to a transition, it always will be displayed next to the handle of the transition line, but you also may choose to have activation conditions and/or Mealy outputs displayed. Click **Diagram Option**, and choose the appropriate combination of Name, Outputs, and Cond. In this display, activation conditions always appear in curly braces ({ }), and Mealy outputs appear in square brackets ([ ]).

## Editing Transition Lines

When you initially create a transition, the transition lines appear as straight lines. Where two lines connect two bubbles (in one direction or both), they appear as one. To pull the lines into arcs, drag the transition marker or handle rectangles on each transition. Figure 4-3 shows this behavior on the transitions between bubbles 1 and 3; the transitions appear as a single two-headed arrow.

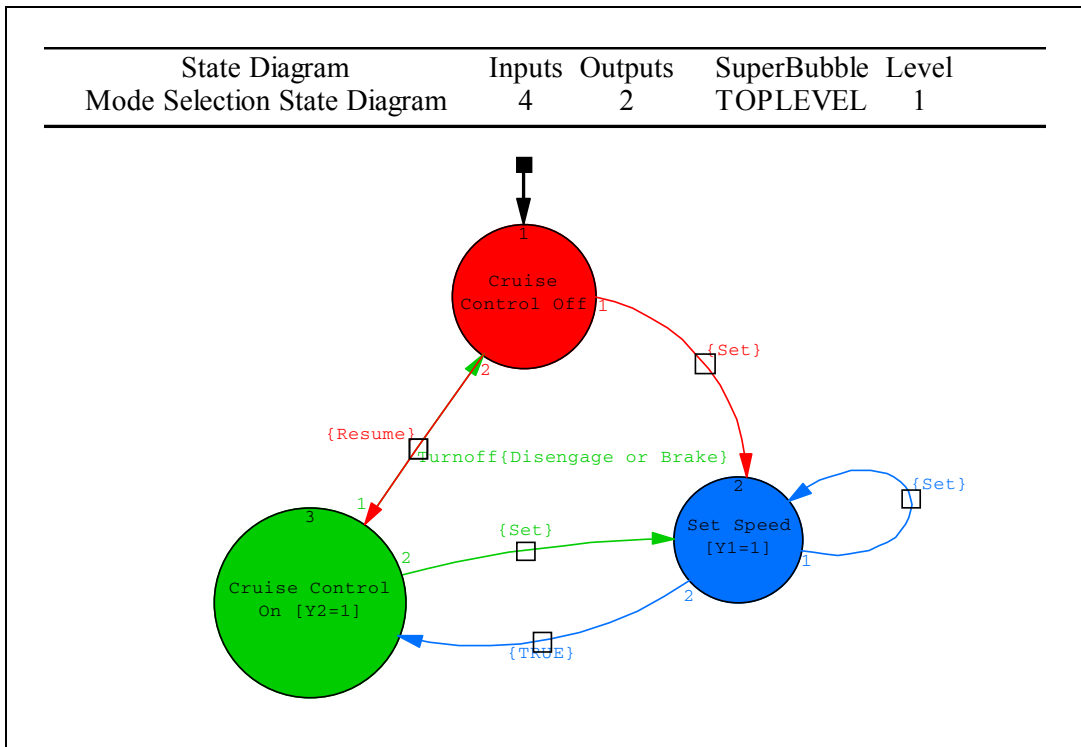


Figure 4-3. STD with Double Transition between Bubbles 1 and 3

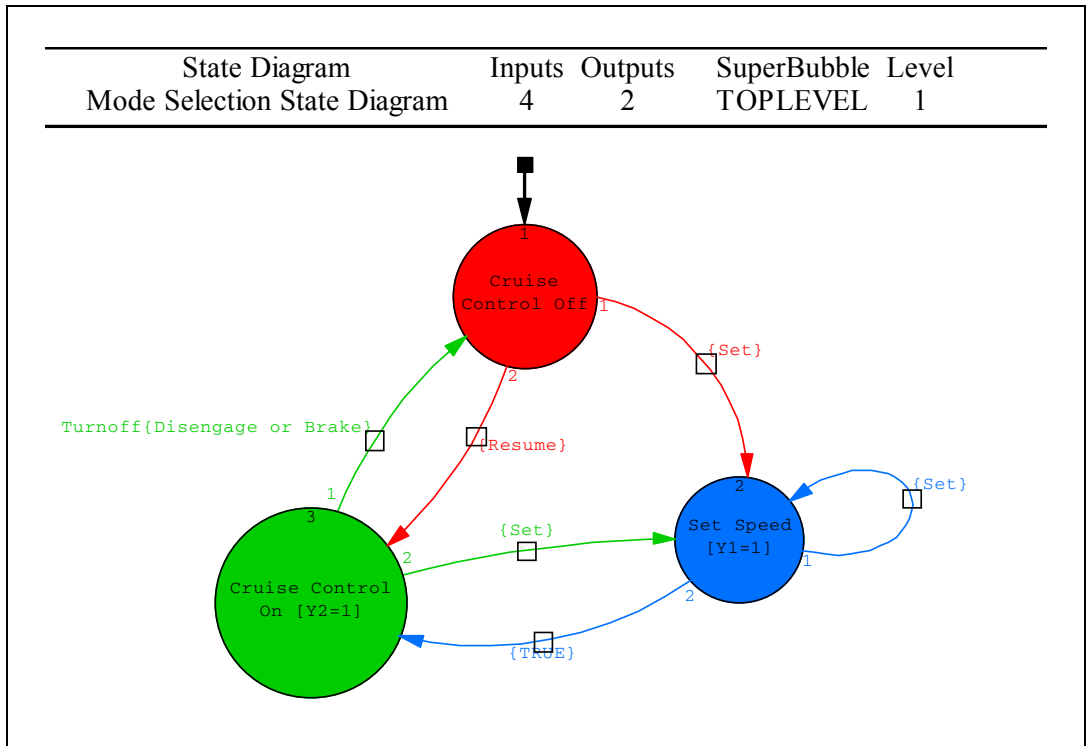
To separate two transitions represented as a single line into two arcs, complete the following steps:

1. Click the line to be moved and hold down the left mouse button.
2. Move the cursor in a direction away from the line.

The first transition line that you entered is the first one moved.

3. Repeat steps 1 and 2 to move multiple lines.

Refer to Figure 4-4 for the result of this process.



**Figure 4-4.** STD with Double Transition Resolved

Transition markers appear at the center of the transition lines, but you can move them along the lines. Click a marker and drag it in any direction. The transition markers determine the location of the transition name.

---

# SuperBubbles

A *SuperBubble* is a single bubble that represents a group of bubbles. SuperBubbles are used to visually simplify large bubble diagrams by introducing hierarchies. SuperBubbles may be nested up to 200 levels deep.

SuperBubbles resemble SuperBlocks in that they are graphics abstractions that enable you to impose a visible hierarchy on a state diagram. Unlike SuperBlocks, however, the hierarchies of SuperBubbles impose no timing properties or other attributes on the bubbles under them. The hierarchy is completely disregarded when the system is analyzed for simulation. They are not carried as entries in the SystemBuild catalog, and they cannot be called or reused by other STDs.

## Creating a SuperBubble

---

To create a SuperBubble, complete the following steps:

1. Select the group of bubbles to be combined into a SuperBubble. To select, lasso a group of bubbles, or select a bubble and then press <Ctrl> while clicking the left mouse button to select each additional bubble in the group.

2. Select **Edit>Make SuperBubble**.

A SuperBubble replaces the selected bubbles in the current diagram. The selected bubbles and all related transitions are moved to a subhierarchy.

On the screen, a SuperBubble looks like an ordinary bubble with a dotted border. Transitions to and from the bubbles in the SuperBubble are shown in the usual manner, but with labels that indicate which bubble and transition number is at the terminus within the SuperBubble.

## Navigating a Bubble Hierarchy

---

To traverse down the hierarchy to the child bubbles, double-click in the SuperBubble.

To return to the parent, press <P>, or select the **View Parent** icon from the STD editor icon bar.

## Returning the Contents of a Subhierarchy to the Parent Page

---

SuperBubbles cease to exist when they are expanded or when the bubbles under them are all deleted.

To return child bubbles to the next level up the hierarchy, complete the following steps:

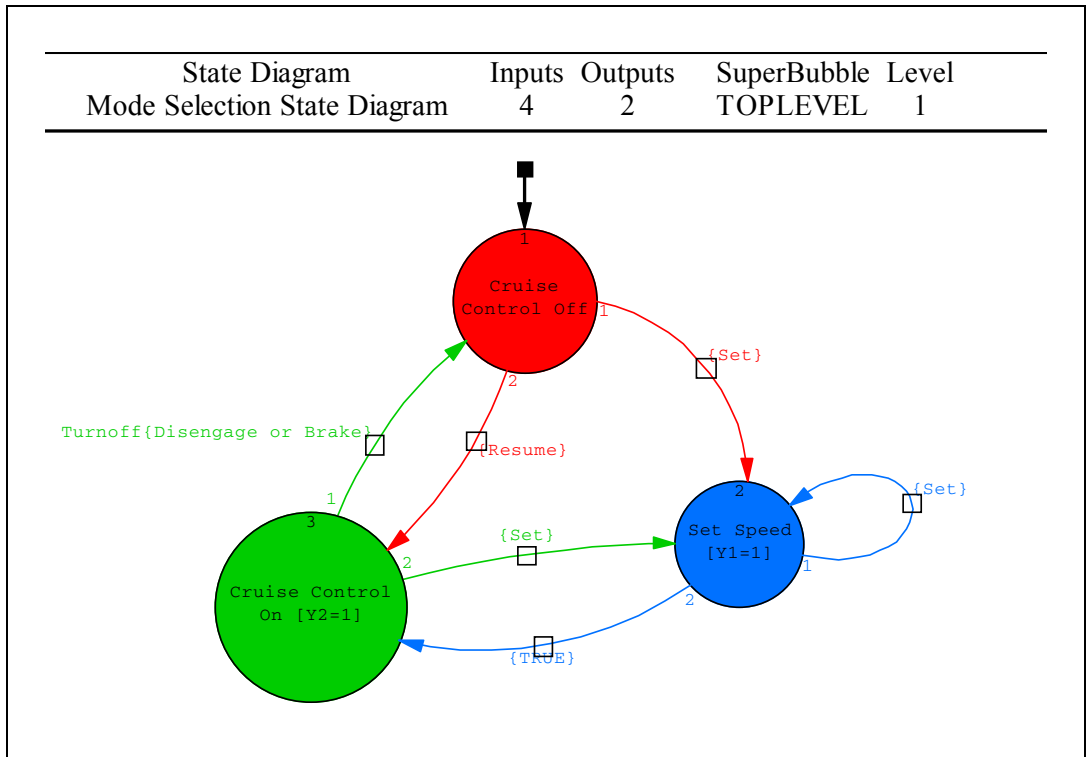
1. Select the Parent SuperBubble.
2. Select **Edit»Expand SuperBubble**.

The SuperBubble is replaced by the individual bubbles, and that level of the hierarchy no longer exists.

3. Repeat steps 1 and 2 to eliminate any desired levels of bubble hierarchy.

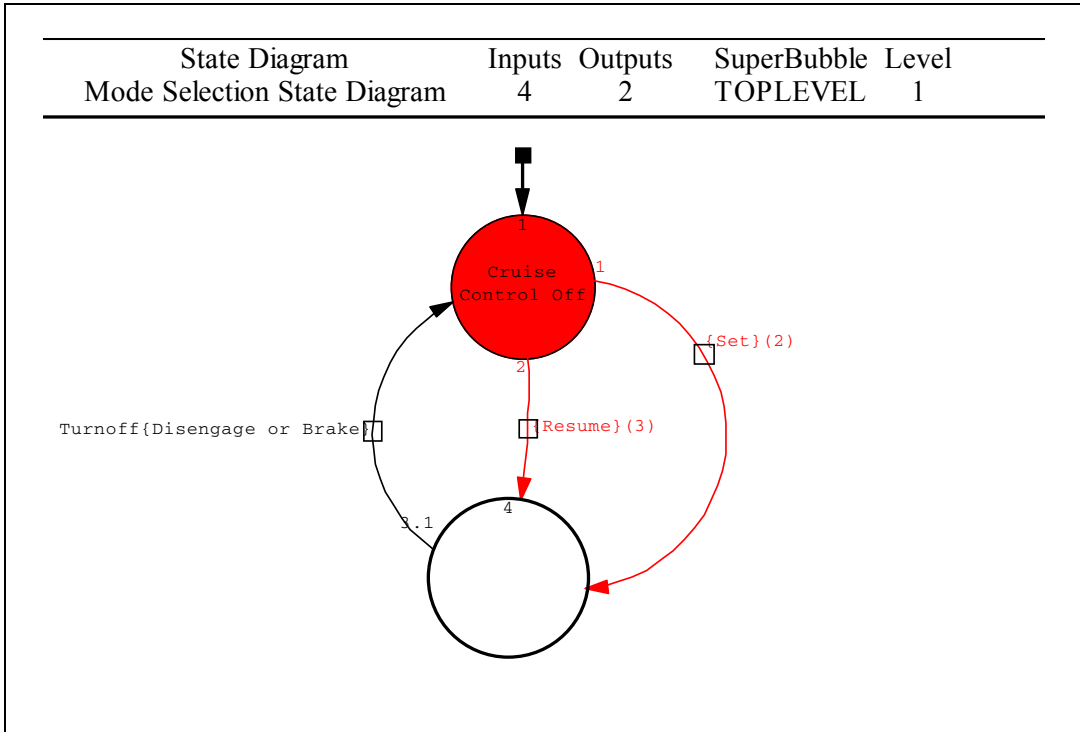
# Examples

Figure 5-1 shows the Mode Selection State Diagram with no SuperBubbles.



**Figure 5-1.** Bubble Diagram with No SuperBubbles

Figure 5-2 shows the same bubble diagram with two of the bubbles organized into a SuperBubble. Observe that the transitions into and out of the SuperBubble are shown clearly, but the internal transitions are not visible.



**Figure 5-2.** Bubble Diagram with a SuperBubble

Figure 5-3 shows the SuperBubble opened for editing. Bubbles 3 and 4 have returned to the screen. Their transitions to the outside world are visible as stubs, and the transitions of bubble 1 are shown with square blocks to indicate that they are inputs to this diagram. The transition out of bubble 3 is shown as an arrow to “nowhere,” with a number next to the transition handle (1) indicating the serial number of the receiving bubble.

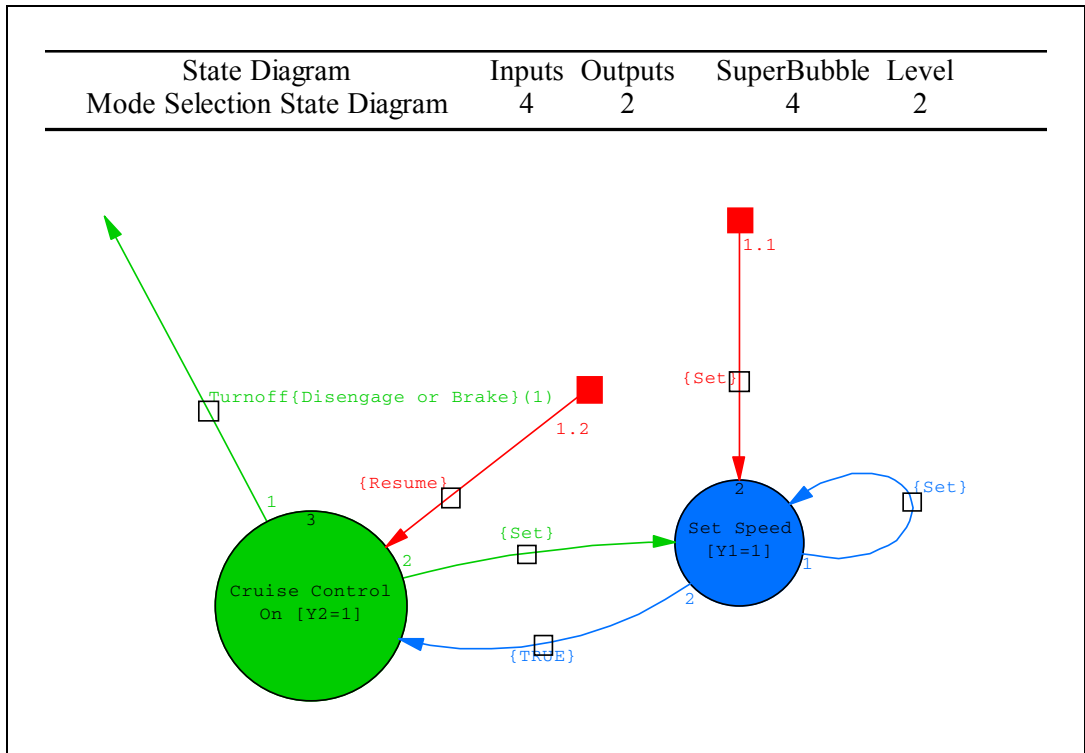


Figure 5-3. Expanded SuperBubble





---

# Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at [ni.com](http://ni.com) for technical support and professional services:

- **Support**—Online technical support resources at [ni.com/support](http://ni.com/support) include the following:
  - **Self-Help Resources**—For immediate 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 Developer Exchange at [ni.com/exchange](http://ni.com/exchange). National Instruments Application Engineers make sure every question receives an answer.
- **Training and Certification**—Visit [ni.com/training](http://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, NI Alliance Program members can help. To learn more, call your local NI office or visit [ni.com/alliance](http://ni.com/alliance).

If you searched [ni.com](http://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](http://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

---

## A

arithmetic expressions, 4-8  
asserting signals, 4-4, 4-7

## B

block, STD, 1-1  
bubble  
    creating diagrams, 4-2  
    definition, 1-2  
    displaying in diagrams, 1-1  
    hierarchy, traversing, 5-2  
    moving, 4-3

## C

conventions used in the manual, *iv*

## D

diagnostic tools (NI resources), A-1  
Diagram Option  
    bubble properties, 4-6  
    transition properties, 4-10  
dialog  
    STD Bubble, 4-4  
    STD Transition, 4-6  
documentation  
    conventions used in the manual, *iv*  
    NI resources, A-1  
drivers (NI resources), A-1

## E

examples (NI resources), A-1

## F

finite state machines, definition, 1-1

## H

help, technical support, A-1

## I

initial bubble, 4-2  
instrument drivers (NI resources), A-1

## K

KnowledgeBase, A-1

## L

logical  
    expressions, 4-9  
    functions, 4-8

## M

Mealy Mode  
    example, 2-5  
    illustrated, 2-2  
    outputs, 2-1, 2-2, 2-5  
Moore Mode  
    example, 2-4  
    illustrated, 2-2  
    outputs, 2-1

## N

National Instruments support and services, A-1  
navigating, bubble hierarchy, 5-2

## P

programming examples (NI resources), A-1

## R

relational operators, 4-9

## S

software (NI resources), A-1

state, 1-2

*See also* STD

transition diagram, 1-1

## STD

block, 1-1

Bubble Dialog, 4-4

Editor, 4-1

icon, 1-1

outputs, Moore and Mealy modes, 2-2

Transition Dialog, 4-6

SuperBubble, 5-1

and SuperBlock, compared, 5-1

eliminating, 5-2

expanding, 5-2

traversing its hierarchy, 5-2

support, technical, A-1

SystemBuild resource file, 4-5

## T

technical support, A-1

training and certification (NI resources), A-1

transition

definition, 1-2

lines, 4-10

markers, 4-11

priorities, 4-9

viewing, 4-6

troubleshooting (NI resources), A-1

## W

Web resources, A-1