ADD - ACCELERATOR DESIGN AND DEPLOY DOCUMENTATION

1.0

Thu Jun 29 2017 21:31:17

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

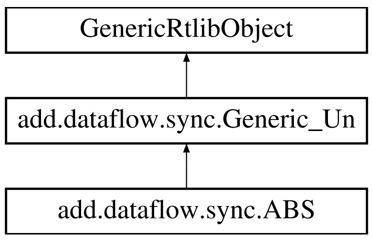
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 add.dataflow.sync.ABS Class Reference

Inheritance diagram for add.dataflow.sync.ABS:



Public Member Functions

- ABS ()
- int Compute (int data)

Additional Inherited Members

3.1.1 Detailed Description

ABS component for the UFV ADD synchronous data flow simulator.

The component is responsible for delivering the absolute value of the input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.1.2 Constructor & Destructor Documentation

3.1.2.1 add.dataflow.sync.ABS.ABS ( )

Object Constructor.

3.1.3 Member Function Documentation

3.1.3.1 int add.dataflow.sync.ABS.Compute (int *data*)

Method responsible for the component computation.

Parameters

data	- Value to be used for computing.

Returns

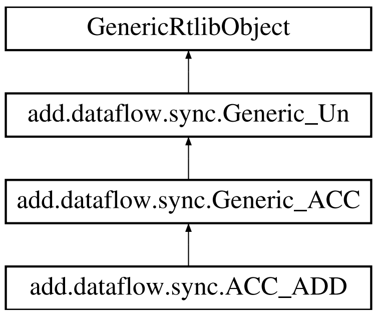
- Returns the result of the computation. In this case, returns the absolute value of the parameter.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ABS.java

3.2 add.dataflow.sync.ACC_ADD Class Reference

Inheritance diagram for add.dataflow.sync.ACC_ADD:


GenericRtlibObject
add.dataflow.sync.Generic_Un
add.dataflow.sync.Generic_ACC
add.dataflow.sync.ACC_ADD

Public Member Functions

- ACC_ADD ()

Additional Inherited Members

3.2.1 Detailed Description

ACC_ADD component for the UFV ADD synchronous data flow simulator.

The component implements an adder accumulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.2.2 Constructor & Destructor Documentation

3.2.2.1 add.dataflow.sync.ACC_ADD.ACC_ADD ( )

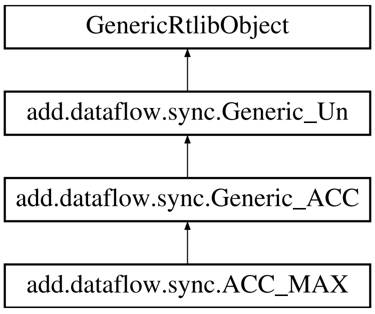
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ACC_ADD.java

3.3 add.dataflow.sync.ACC_MAX Class Reference

Inheritance diagram for add.dataflow.sync.ACC_MAX:


GenericRtlibObject
add.dataflow.sync.Generic_Un
add.dataflow.sync.Generic_ACC
add.dataflow.sync.ACC_MAX

Public Member Functions

- ACC_MAX ()
- void Reseted ()

Protected Member Functions

- void Accumulate (int data)

Additional Inherited Members

3.3.1 Detailed Description

ACC_MAX component for the UFV ADD synchronous data flow simulator.

The component implements a store for the highest input value.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.3.2 Constructor & Destructor Documentation

3.3.2.1 add.dataflow.sync.ACC_MAX.ACC_MAX ( )

Object Constructor.

3.3.3 Member Function Documentation

3.3.3.1 void add.dataflow.sync.ACC_MAX.Accumulate (int *data*) [protected]

Method that compares the parameter to the stored value. If the parameter is larger, it will override the stored value.

Parameters

data	- Value to be used for computing.

3.3.3.2 void add.dataflow.sync.ACC_MAX.Reseted ( )

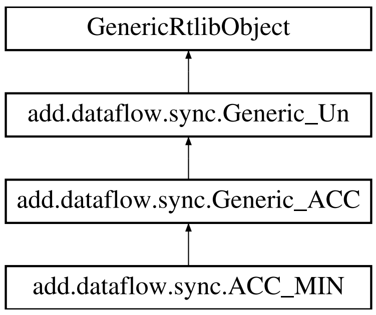
Method responsible for actions required when "Reset" occurs.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ACC_MAX.java

3.4 add.dataflow.sync.ACC_MIN Class Reference

Inheritance diagram for add.dataflow.sync.ACC_MIN:


GenericRtlibObject
add.dataflow.sync.Generic_Un
add.dataflow.sync.Generic_ACC
add.dataflow.sync.ACC_MIN

Public Member Functions

- ACC_MIN ()
- void Reseted ()

Protected Member Functions

- void Accumulate (int data)

Additional Inherited Members

3.4.1 Detailed Description

ACC_MIN component for the UFV ADD synchronous data flow simulator.

The component implements a store for the lowest input value.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.4.2 Constructor & Destructor Documentation

3.4.2.1 add.dataflow.sync.ACC_MIN.ACC_MIN ( )

Object Constructor.

3.4.3 Member Function Documentation

3.4.3.1 void add.dataflow.sync.ACC_MIN.Accumulate (int *data*) `[protected]`

Method that compares the parameter to the stored value. If the parameter is smaller, it will replace the stored value.

Parameters

data	- Value to be used for computing.

3.4.3.2 void add.dataflow.sync.ACC_MIN.Reseted ( )

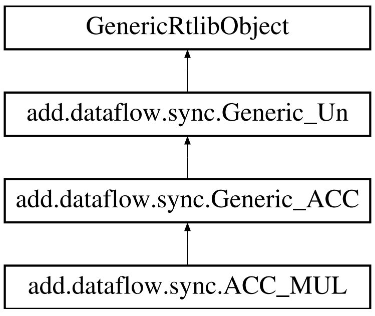
Method responsible for actions required when "Reset" occurs.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ACC_MIN.java

3.5 add.dataflow.sync.ACC_MUL Class Reference

Inheritance diagram for add.dataflow.sync.ACC_MUL:


GenericRtlibObject
add.dataflow.sync.Generic_Un
add.dataflow.sync.Generic_ACC
add.dataflow.sync.ACC_MUL

Public Member Functions

- ACC_MUL ()
- void Reseted ()

Protected Member Functions

- void Accumulate (int data)

Additional Inherited Members

3.5.1 Detailed Description

ACC_MUL component for the UFV ADD synchronous data flow simulator.

The component implements a multiplication accumulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.5.2 Constructor & Destructor Documentation

3.5.2.1 add.dataflow.sync.ACC_MUL.ACC_MUL ( )

Object Constructor.

3.5.3 Member Function Documentation

3.5.3.1 void add.dataflow.sync.ACC_MUL.Accumulate (int *data*) `[protected]`

Method that accumulates the input value with the stored. In this case, it multiplies the value stored by the input and stores it.

3.5.3.2 void add.dataflow.sync.ACC_MUL.Reseted ( )

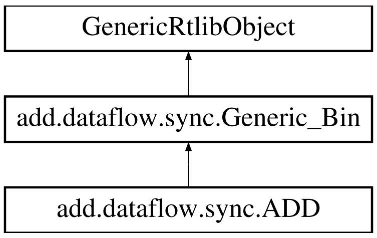
Method responsible for actions required when "Reset" occurs.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ACC_MUL.java

3.6 add.dataflow.sync.ADD Class Reference

Inheritance diagram for add.dataflow.sync.ADD:



Public Member Functions

- ADD ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.6.1 Detailed Description

ADD component for the UFV ADD synchronous data flow simulator.

The component is responsible for adding the inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.6.2 Constructor & Destructor Documentation

3.6.2.1 add.dataflow.sync.ADD.ADD ( )

Object Constructor.

3.6.3 Member Function Documentation

3.6.3.1 int add.dataflow.sync.ADD.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case performs a addition of the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

- Returns the result of the computation. In this case the value of the addition of the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ADD.java

3.7 add.examples.dataflow_sync.ADD_AB_HELLOWORLD_DATAFLOW_FPGA Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.7.1 Detailed Description

ADD_AB data flow example in FPGA Board in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/ADD_AB_HELLOWORLD_DATAFLOW_FPGA.java

3.8 add.examples.dataflow_sync.ADD_AB_HELLOWORLD_DATAFLOW_HADES Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.8.1 Detailed Description

ADD_AB data flow example in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/ADD_AB_HELLOWORLD_DATAFLOW_HADES.java

3.9 add.examples.dataflow_sync.ADD_AB_HELLOWORLD_DATAFLOW_HADES_WITH_GENERATED_HDS Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.9.1 Detailed Description

ADD_AB data flow (Automatically generated) example in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

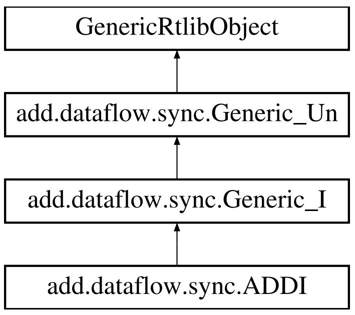
∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/ADD_AB_HELLOWORLD_DATAFLOW_HADES_WITH_GENERATED_HD-S.java

3.10 add.dataflow.sync.ADDI Class Reference

Inheritance diagram for add.dataflow.sync.ADDI:



Public Member Functions

- ADDI ()
- int Compute (int data)

Additional Inherited Members

3.10.1 Detailed Description

ADDI component for the UFV ADD synchronous data flow simulator.

The component is responsible for adding the input by a (Immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.10.2 Constructor & Destructor Documentation

3.10.2.1 add.dataflow.sync.ADDI.ADDI ( )

Object Constructor.

3.10.3 Member Function Documentation

3.10.3.1 int add.dataflow.sync.ADDI.Compute (int *data*)

Method responsible for the component computation: in this case performs a addition of the parameter by an (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

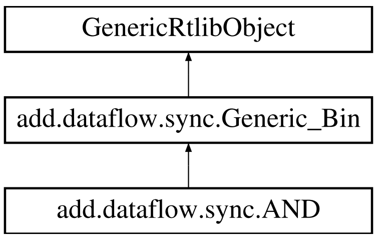
- Returns the result of the computation. In this case the value of the addition of the parameter by the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ADDI.java

3.11 add.dataflow.sync.AND Class Reference

Inheritance diagram for add.dataflow.sync.AND:



Public Member Functions

- AND ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.11.1 Detailed Description

AND component for the UFV ADD synchronous data flow simulator.

The component is responsible for the logical operation "AND" between the input

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.11.2 Constructor & Destructor Documentation

3.11.2.1 add.dataflow.sync.AND.AND ( )

Object Constructor.

3.11.3 Member Function Documentation

3.11.3.1 int add.dataflow.sync.AND.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case it performs the logical operation "AND" between the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

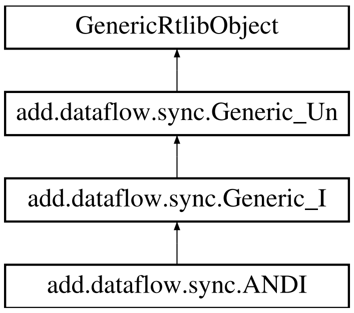
- Returns the result of the computation. In this case the result of the logical operation "AND" between the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/AND.java

3.12 add.dataflow.sync.ANDI Class Reference

Inheritance diagram for add.dataflow.sync.ANDI:



Public Member Functions

- ANDI ()
- int Compute (int data)

Additional Inherited Members

3.12.1 Detailed Description

ANDI component for the UFV ADD synchronous data flow simulator.

The component is responsible for the logical operation "AND" between the input and a constant (Immediate)

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.12.2 Constructor & Destructor Documentation

3.12.2.1 add.dataflow.sync.ANDI.ANDI ( )

Object Constructor.

3.12.3 Member Function Documentation

3.12.3.1 int add.dataflow.sync.ANDI.Compute (int *data*)

Method responsible for the component computation: in this case it performs the logical operation "AND" between the parameter and the (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

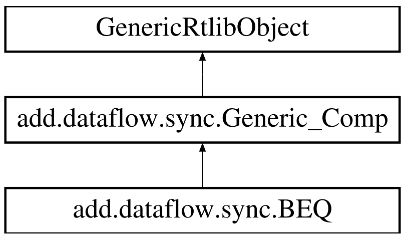
- Returns the result of the computation. In this case the result of the logical operation "AND" between the parameter and the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ANDI.java

3.13 add.dataflow.sync.BEQ Class Reference

Inheritance diagram for add.dataflow.sync.BEQ:



Public Member Functions

- BEQ ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.13.1 Detailed Description

BEQ component for the UFV ADD synchronous data flow simulator.

The component is responsible for comparing equality between the input. Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.13.2 Constructor & Destructor Documentation

3.13.2.1 add.dataflow.sync.BEQ.BEQ ( )

Object Constructor.

3.13.3 Member Function Documentation

3.13.3.1 int add.dataflow.sync.BEQ.Compute (int *data1,* int *data2*)

Method responsible for component computing: in this case performs a comparison of equality between the input. Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

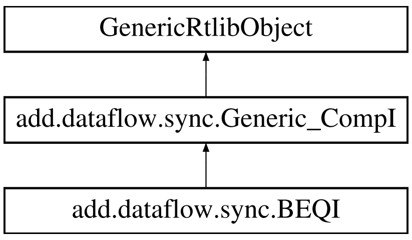
- Returns the result of the computation. In this case "1" if the parameters are equal or "0" if they are different.

The documentation for this class was generated from the following file:

- add/dataflow/sync/BEQ.java

3.14 add.dataflow.sync.BEQI Class Reference

Inheritance diagram for add.dataflow.sync.BEQI:



Public Member Functions

- BEQI ()
- int Compute (int data)

Additional Inherited Members

3.14.1 Detailed Description

BEQI component for the UFV ADD synchronous data flow simulator.

The component is responsible for comparing equality between the input and a constant (Immediate). Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.14.2 Constructor & Destructor Documentation

3.14.2.1 add.dataflow.sync.BEQI.BEQI ( )

Object Constructor.

3.14.3 Member Function Documentation

3.14.3.1 int add.dataflow.sync.BEQI.Compute (int *data*)

Method responsible for component computing: in this case performs a comparison of equality between the input and a constant. Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Parameters

data	- Value to be used for computing.

Returns

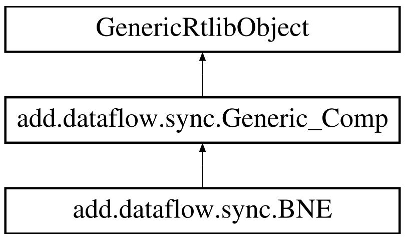
- Returns the result of the computation. In this case "1" if the parameter is equal to the constraint or "0" if they are different.

The documentation for this class was generated from the following file:

- add/dataflow/sync/BEQI.java

3.15 add.dataflow.sync.BNE Class Reference

Inheritance diagram for add.dataflow.sync.BNE:



Public Member Functions

- BNE ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.15.1 Detailed Description

BNE component for the UFV ADD synchronous data flow simulator.

The component is responsible for comparing inequality between the input. Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.15.2 Constructor & Destructor Documentation

3.15.2.1 add.dataflow.sync.BNE.BNE ( )

Object Constructor.

3.15.3 Member Function Documentation

3.15.3.1 int add.dataflow.sync.BNE.Compute (int *data1*, int *data2*)

Method responsible for component computing: in this case performs a comparison of inequality between the input. Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

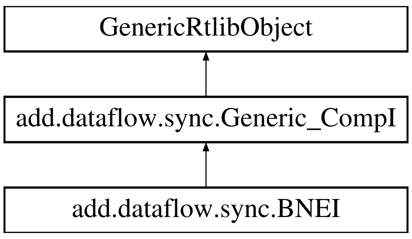
- Returns the result of the computation. In this case "1" if the parameters are different or "0" if they are equal.

The documentation for this class was generated from the following file:

- add/dataflow/sync/BNE.java

3.16 add.dataflow.sync.BNEI Class Reference

Inheritance diagram for add.dataflow.sync.BNEI:



Public Member Functions

- BNEI ()
- int Compute (int data)

Additional Inherited Members

3.16.1 Detailed Description

BEQI component for the UFV ADD synchronous data flow simulator.

The component is responsible for comparing inequality between the input and a constant (Immediate). Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.16.2 Constructor & Destructor Documentation

3.16.2.1 add.dataflow.sync.BNEI.BNEI ( )

Object Constructor.

3.16.3 Member Function Documentation

3.16.3.1 int add.dataflow.sync.BNEI.Compute (int *data*)

Method responsible for component computing: in this case performs a comparison of inequality between the input and a constant. Depending on the result of the comparison, the "IF" output or the "ELSE" output will receive the value "1" while the other will receive the value "0".

Parameters

data	- Value to be used for computing.

Returns

- Returns the result of the computation. In this case "0" if the parameter is equal to the constraint or "1" if they are different.

The documentation for this class was generated from the following file:

- add/dataflow/sync/BNEI.java

3.17 add.examples.dataflow_sync.BRANCH_TEST_DATAFLOW_FPGA Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.17.1 Detailed Description

Branch_Test data flow example in FPGA Board in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/BRANCH_TEST_DATAFLOW_FPGA.java

3.18 add.examples.dataflow_sync.BRANCH_TEST_DATAFLOW_HADES Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.18.1 Detailed Description

Branch_Test data flow example in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/BRANCH_TEST_DATAFLOW_HADES.java

3.19 add.examples.dataflow_sync.BRANCH_TEST_DATAFLOW_HADES_WITH_GENERATED_HDS Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.19.1 Detailed Description

Branch_Test data flow (Automatically generated) example in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/BRANCH_TEST_DATAFLOW_HADES_WITH_GENERATED_HDS.java

3.20 add.util.CONF_READER Class Reference

Public Member Functions

- int[ ] ReadConfig (File file)

3.20.1 Detailed Description

Class responsible for providing useful routines for the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.20.2 Member Function Documentation

3.20.2.1 int [ ] add.util.CONF_READER.ReadConfig (File *file*)

Method responsible for reading a configuration file and returning a vector with the values read.

Parameters

file	- File to read

Returns

- Returns a vector containing the values read in the file.

The documentation for this class was generated from the following file:

- add/util/CONF_READER.java

3.21 add.dataflow.DATAFLOW_SYNC_BASE_SIMUL Class Reference

Public Member Functions

- int[] start_HADES (int[] CONF, String DESIGN)
- void start_HADES (String CONF, String DESIGN, String DESIRED_RETURN)
- int[] start_FPGA (int[] CONF, String QUARTUS)
- void start_FPGA (String CONF, String QUARTUS, String DESIRED_RETURN)
- int[] exec_HADES (int[] raw_data, String DESIGN)
- int[] exec_FPGA (int[] raw_data, String QUARTUS)

3.21.1 Detailed Description

Base class for executing algorithms in the UFV ADD synchronous data flow simulator orin the bundle with FPGA.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.21.2 Member Function Documentation

3.21.2.1 int [ ] add.dataflow.DATAFLOW_SYNC_BASE_SIMUL.exec_FPGA (int[ ] *raw_data,* String *QUARTUS*)

Execution in FPGA Boards

Parameters

raw_data	- Vector of data to be processed
QUARTUS	- Path to the quartus_stp application.

Returns

- Returns a vector with the processing results.

3.21.2.2 int [ ] add.dataflow.DATAFLOW_SYNC_BASE_SIMUL.exec_HADES (int[ ] *raw_data,* String *DESIGN*)

Execution in HADES Simulator

Parameters

raw_data	- Vector of data to be processed
DESIGN	- Design to be used to run the simulator.

Returns

- Returns a vector with the processing results.

3.21.2.3 int [ ] add.dataflow.DATAFLOW_SYNC_BASE_SIMUL.start_FPGA (int[ ] *CONF,* String *QUARTUS*)

Method responsible for running the algorithm on the FPGA board.

Parameters

CONF	- Configuration vector and data to be executed.
QUARTUS	- Path to the quartus_stp application.

Returns

- Returns a vector with the processing results.

3.21.2.4 void add.dataflow.DATAFLOW_SYNC_BASE_SIMUL.start_FPGA (String *CONF*, String *QUARTUS*, String *DESIRED_RETURN*)

Method responsible for running the algorithm on the FPGA boardand display the output in the system default output.

Parameters

CONF	- File containing the configuration and data to be processed.
QUARTUS	- Path to the quartus_stp application.
DESIRED_RET-URN	- Expected outcome.

3.21.2.5 int [ ] add.dataflow.DATAFLOW_SYNC_BASE_SIMUL.start_HADES (int[ ] *CONF*, String *DESIGN*)

Method responsible for executing the algorithm in the simulator.

Parameters

CONF	- Configuration vector and data to be executed.
DESIGN	- Design to be used to run the simulator.

Returns

- Returns a vector with the processing results.

3.21.2.6 void add.dataflow.DATAFLOW_SYNC_BASE_SIMUL.start_HADES (String *CONF*, String *DESIGN*, String *DESIRED_RETURN*)

Method responsible for executing the algorithm in the simulator and display the output in the system default output.

Parameters

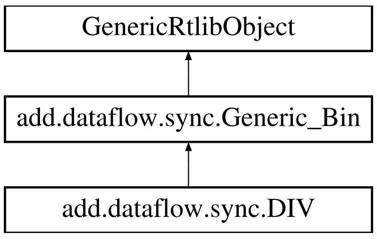
CONF	- File containing the configuration and data to be processed.
DESIGN	- Design to be used to run the simulator.
DESIRED_RET-URN	- Expected outcome.

The documentation for this class was generated from the following file:

- add/dataflow/DATAFLOW_SYNC_BASE_SIMUL.java

3.22 add.dataflow.sync.DIV Class Reference

Inheritance diagram for add.dataflow.sync.DIV:



Public Member Functions

- DIV ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.22.1 Detailed Description

DIV component for the UFV ADD synchronous data flow simulator.

The component is responsible for dividing the inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.22.2 Constructor & Destructor Documentation

3.22.2.1 add.dataflow.sync.DIV.DIV ( )

Object Constructor.

3.22.3 Member Function Documentation

3.22.3.1 int add.dataflow.sync.DIV.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case performs a division of the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

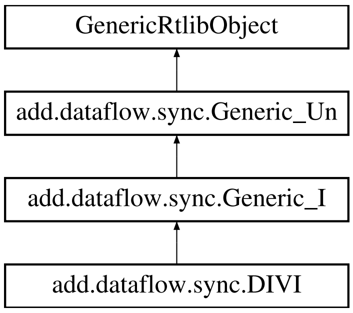
- Returns the result of the computation. In this case the value of the division of the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/DIV.java

3.23 add.dataflow.sync.DIVI Class Reference

Inheritance diagram for add.dataflow.sync.DIVI:



Public Member Functions

- DIVI ()
- int Compute (int data)

Additional Inherited Members

3.23.1 Detailed Description

DIVI component for the UFV ADD synchronous data flow simulator.

The component is responsible for dividing the input by a (immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.23.2 Constructor & Destructor Documentation

3.23.2.1 add.dataflow.sync.DIVI.DIVI ( )

Object Constructor.

3.23.3 Member Function Documentation

3.23.3.1 int add.dataflow.sync.DIVI.Compute (int *data*)

Method responsible for the component computation: in this case performs a division of the parameter by an (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

- Returns the result of the computation. In this case the value of the division of the parameter by the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/DIVI.java

3.24 add.examples.dataflow_sync.FIR_4_DATAFLOW_FPGA Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.24.1 Detailed Description

FIR_4 data flow example in FPGA Board in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/FIR_4_DATAFLOW_FPGA.java

3.25 add.examples.dataflow_sync.FIR_4_DATAFLOW_HADES Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.25.1 Detailed Description

FIR_4 data flow example in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

∗ 1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/FIR_4_DATAFLOW_HADES.java

3.26 add.examples.dataflow_sync.FIR_4_DATAFLOW_HADES_WITH_GENERATED_HD-S Class Reference

Static Public Member Functions

- static void **main** (String argv[])

3.26.1 Detailed Description

FIR_4 data flow (Automatically generated) example in the UFV ADD synchronous data flow simulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

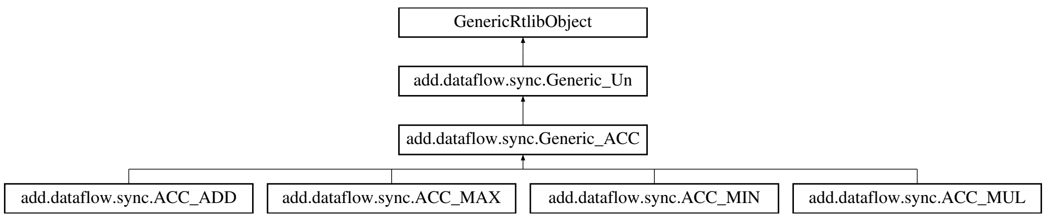
1.0

The documentation for this class was generated from the following file:

- add/examples/dataflow_sync/FIR_4_DATAFLOW_HADES_WITH_GENERATED_HDS.java

3.27 add.dataflow.sync.Generic_ACC Class Reference

Inheritance diagram for add.dataflow.sync.Generic_ACC:


GenericRtlibObject
add.dataflow.sync.Generic_Un
add.dataflow.sync.Generic_ACC
add.dataflow.sync.ACC_ADD
add.dataflow.sync.ACC_MAX
add.dataflow.sync.ACC_MIN
add.dataflow.sync.ACC_MUL

Public Member Functions

- Generic_ACC ()
- void Reseted ()
- void evaluate (Object arg)

Protected Member Functions

- void Accumulate (int data)

Protected Attributes

- int **acc**
- boolean **start**

3.27.1 Detailed Description

Generic_ACC component for the UFV ADD synchronous data flow simulator.

The component implements a generic accumulator.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.27.2 Constructor & Destructor Documentation

3.27.2.1 add.dataflow.sync.Generic_ACC.Generic_ACC ( )

Object Constructor.

3.27.3 Member Function Documentation

3.27.3.1 void add.dataflow.sync.Generic_ACC.Accumulate (int *data*) [protected]

Method responsible for performing the accumulation or not.

Parameters

data	- Value to be used for the computation.

3.27.3.2 void add.dataflow.sync.Generic_ACC.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked whether the ports are connected and will execute the Compute (int data) method if the R_IN input is high level. It will execute the Reseted(), Tick_Up(), and Tick_Down() methods if their respective entries order it. It will update the output with the ACC value when the computation finishes.

Parameters

arg	an arbitrary object argument

3.27.3.3 void add.dataflow.sync.Generic_ACC.Reseted ( )

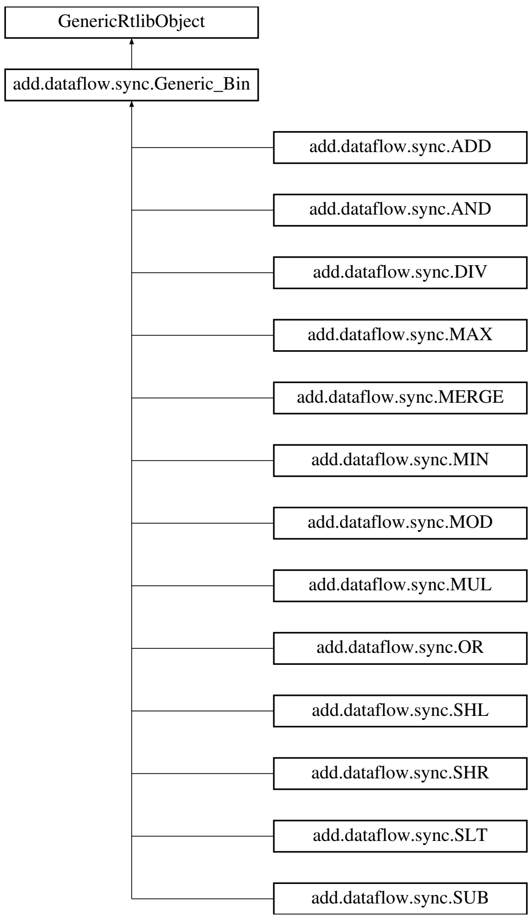
Method executed when the signal from the reset input goes to high logic level.In this case it clears the text displayed by the component and de accumulator.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_ACC.java

3.28 add.dataflow.sync.Generic_Bin Class Reference

Inheritance diagram for add.dataflow.sync.Generic_Bin:



Public Member Functions

- Generic_Bin ()
- void constructPorts ()
- void setString (String s)
- void setSymbol (Symbol s)
- int Compute (int data1, int data2)
- void NotCompute ()

- void Reseted ()
- void Tick_Up ()
- void Tick_Down ()
- void Set_Name (String l)
- void evaluate (Object arg)
- boolean needsDynamicSymbol ()
- void constructDynamicSymbol ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)

Protected Attributes

- Label **stringLabel**
- String **l**
- PortStdLogic1164 **port_CLK**
- PortStdLogicVector **port_DATA_IN1**

3.28.1 Detailed Description

Generic_Bin component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components with two inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.28.2 Constructor & Destructor Documentation

3.28.2.1 add.dataflow.sync.Generic_Bin.Generic_Bin ( )

Object Constructor.

3.28.3 Member Function Documentation

3.28.3.1 int add.dataflow.sync.Generic_Bin.Compute (int *data1*, int *data2*)

Method responsible for the computation of the output.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 1.

Returns

- Return of computation

3.28.3.2 void add.dataflow.sync.Generic_Bin.constructDynamicSymbol ( )

Method responsible for dynamically constructing the component symbol.

3.28.3.3 void add.dataflow.sync.Generic_Bin.constructPorts ( )

Method responsible for initializing the component input and output ports.

3.28.3.4 void add.dataflow.sync.Generic_Bin.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked whether the ports are connected and will execute the Compute (int data) method if the R_IN (1 and 2) inputs are high level. It will execute the Reseted(), Tick_Up(), and Tick_Down() methods if their respective entries order it. It will update the output with the Compute(int data) method result.

Parameters

arg	an arbitrary object argument

3.28.3.5 boolean add.dataflow.sync.Generic_Bin.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.28.3.6 boolean add.dataflow.sync.Generic_Bin.needsDynamicSymbol ( )

Method responsible for indicating to the simulator that the component's symbol will be constructed dynamically by the constructDynamicSymbol() method, or will be read from a file of the same name as the ".sym" extension.

3.28.3.7 void add.dataflow.sync.Generic_Bin.NotCompute ( )

Method executed when computing is not performed. In this case it clears the text displayed by the component.

3.28.3.8 void add.dataflow.sync.Generic_Bin.Reseted ( )

Method executed when the signal from the reset input goes to high logic level.In this case it clears the text displayed by the component.

3.28.3.9 void add.dataflow.sync.Generic_Bin.Set_Name (String *l*)

Method responsible for changing the label that displays the name of the component.

3.28.3.10 void add.dataflow.sync.Generic_Bin.setString (String *s*)

Method responsible for updating the text displayed by the component.

Parameters

s	- Text to be updated.

3.28.3.11 void add.dataflow.sync.Generic_Bin.setSymbol (Symbol *s*)

Method responsible for updating the component symbol.

3.28.3.12 void add.dataflow.sync.Generic_Bin.Tick_Down ( )

Method executed when the clock signal goes to low logic level.

3.28.3.13 void add.dataflow.sync.Generic_Bin.Tick_Up ( )

Method executed when the clock signal goes to high logic level.

3.28.3.14 void add.dataflow.sync.Generic_Bin.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

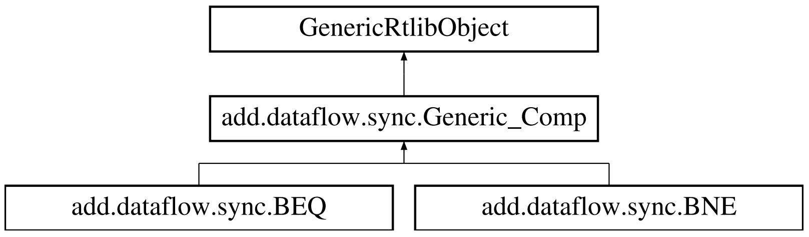
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_Bin.java

3.29 add.dataflow.sync.Generic_Comp Class Reference

Inheritance diagram for add.dataflow.sync.Generic_Comp:



Public Member Functions

- Generic_Comp ()
- void constructPorts ()
- void setString (String s)
- void setSymbol (Symbol s)
- int Compute (int data1, int data2)
- void Reseted ()
- void Tick_Up ()
- void Tick_Down ()
- void Set_Name (String l)
- void evaluate (Object arg)

- boolean needsDynamicSymbol ()
- void constructDynamicSymbol ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)

Protected Attributes

- Label **stringLabel**
- String **l**
- PortStdLogic1164 **port_CLK**
- PortStdLogicVector **port_DATA_IN1**
- PortStdLogic1164 **port_IF**

3.29.1 Detailed Description

Generic_Comp component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components with an input and that make a comparison between the inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.29.2 Constructor & Destructor Documentation

3.29.2.1 add.dataflow.sync.Generic_Comp.Generic_Comp ( )

Object Constructor.

3.29.3 Member Function Documentation

3.29.3.1 int add.dataflow.sync.Generic_Comp.Compute (int *data1*, int *data2*)

Method responsible for the computation of the output.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 1.

Returns

- Return of computation

3.29.3.2 void add.dataflow.sync.Generic_Comp.constructDynamicSymbol ( )

Method responsible for dynamically constructing the component symbol.

3.29.3.3 void add.dataflow.sync.Generic_Comp.constructPorts ( )

Method responsible for initializing the component input and output ports.

3.29.3.4 void add.dataflow.sync.Generic_Comp.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events.In this case, it will be checked whether the ports are connected and will execute the Compute (int data) method if the R_IN (1 and 2) inputs are high level. It will execute the Reseted(), Tick_Up(), and Tick_Down() methods if their respective entries order it. It will update the output with the Compute(int data) method result.

Parameters

arg	an arbitrary object argument

3.29.3.5 boolean add.dataflow.sync.Generic_Comp.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.29.3.6 boolean add.dataflow.sync.Generic_Comp.needsDynamicSymbol ( )

Method responsible for indicating to the simulator that the component's symbol will be constructed dynamically by the constructDynamicSymbol() method, or will be read from a file of the same name as the ".sym" extension.

3.29.3.7 void add.dataflow.sync.Generic_Comp.Reseted ( )

Method executed when the signal from the reset input goes to high logic level.In this case it clears the text displayed by the component.

3.29.3.8 void add.dataflow.sync.Generic_Comp.Set_Name (String *l*)

Method responsible for changing the label that displays the name of the component.

3.29.3.9 void add.dataflow.sync.Generic_Comp.setString (String *s*)

Method responsible for updating the text displayed by the component.

Parameters

s	- Text to be updated.

3.29.3.10 void add.dataflow.sync.Generic_Comp.setSymbol (Symbol *s*)

Method responsible for updating the component symbol.

3.29.3.11 void add.dataflow.sync.Generic_Comp.Tick_Down ( )

Method executed when the clock signal goes to low logic level.

3.29.3.12 void add.dataflow.sync.Generic_Comp.Tick_Up ( )

Method executed when the clock signal goes to high logic level.

3.29.3.13 void add.dataflow.sync.Generic_Comp.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

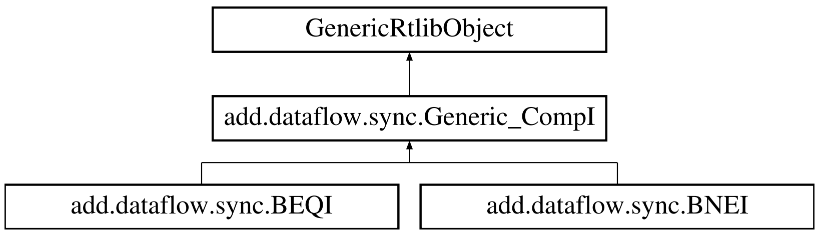
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_Comp.java

3.30 add.dataflow.sync.Generic_CompI Class Reference

Inheritance diagram for add.dataflow.sync.Generic_CompI:



Public Member Functions

- Generic_CompI ()
- void constructPorts ()
- void setString (String s)
- void setSymbol (Symbol s)
- int Compute (int data)
- void Reseted ()
- void Tick_Up ()
- void Tick_Down ()
- void Set_Name (String l)
- void evaluate (Object arg)
- boolean needsDynamicSymbol ()
- void constructDynamicSymbol ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)
- void mousePressed (java.awt.event.MouseEvent me)

Protected Attributes

- Label **stringLabel**
- String **s**
- Rectangle **background**
- PortStdLogic1164 **port_CLK**
- PortStdLogicVector **port_DATA_IN**
- int **constante**

3.30.1 Detailed Description

Generic_Compl component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components with an input and that make a comparison with a (immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.30.2 Constructor & Destructor Documentation

3.30.2.1 add.dataflow.sync.Generic_Compl.Generic_Compl ( )

Object Constructor.

3.30.3 Member Function Documentation

3.30.3.1 int add.dataflow.sync.Generic_Compl.Compute (int *data*)

Method responsible for the computation of the output.

Parameters

data	- Value to be used for the computation.

Returns

- Return of computation

3.30.3.2 void add.dataflow.sync.Generic_Compl.constructDynamicSymbol ( )

Method responsible for dynamically constructing the component symbol.

3.30.3.3 void add.dataflow.sync.Generic_Compl.constructPorts ( )

Method responsible for initializing the component input and output ports.

3.30.3.4 void add.dataflow.sync.Generic_Compl.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked whether the ports are connected and will execute the Compute (int data) method if the R_IN input is high level. It will execute the Reseted(), Tick_Up(), and Tick_Down() methods if their respective entries order it. It will update the output with the Compute(int data) method result.

Parameters

arg	an arbitrary object argument

3.30.3.5 boolean add.dataflow.sync.Generic_Compl.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.30.3.6 void add.dataflow.sync.Generic_Compl.mousePressed (java.awt.event.MouseEvent *me*)

Method responsible for changing the value of the constant for more or less, depending on whether the mouse click is done by the right or left button respectively.

Parameters

me	- Object where the event occurred.

3.30.3.7 boolean add.dataflow.sync.Generic_Compl.needsDynamicSymbol ( )

Method responsible for indicating to the simulator that the component's symbol will be constructed dynamically by the constructDynamicSymbol() method, or will be read from a file of the same name as the ".sym" extension.

3.30.3.8 void add.dataflow.sync.Generic_Compl.Reseted ( )

Method executed when the signal from the reset input goes to high logic level.In this case it clears the text displayed by the component.

3.30.3.9 void add.dataflow.sync.Generic_Compl.Set_Name (String *l*)

Method responsible for changing the label that displays the name of the component.

3.30.3.10 void add.dataflow.sync.Generic_Compl.setString (String *s*)

Method responsible for updating the text displayed by the component.

Parameters

s	- Text to be updated.

3.30.3.11 void add.dataflow.sync.Generic_Compl.setSymbol (Symbol *s*)

Method responsible for updating the component symbol.

3.30.3.12 void add.dataflow.sync.Generic_Compl.Tick_Down ( )

Method executed when the clock signal goes to low logic level.

3.30.3.13 void add.dataflow.sync.Generic_Compl.Tick_Up ( )

Method executed when the clock signal goes to high logic level.

3.30.3.14 void add.dataflow.sync.Generic_Compl.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

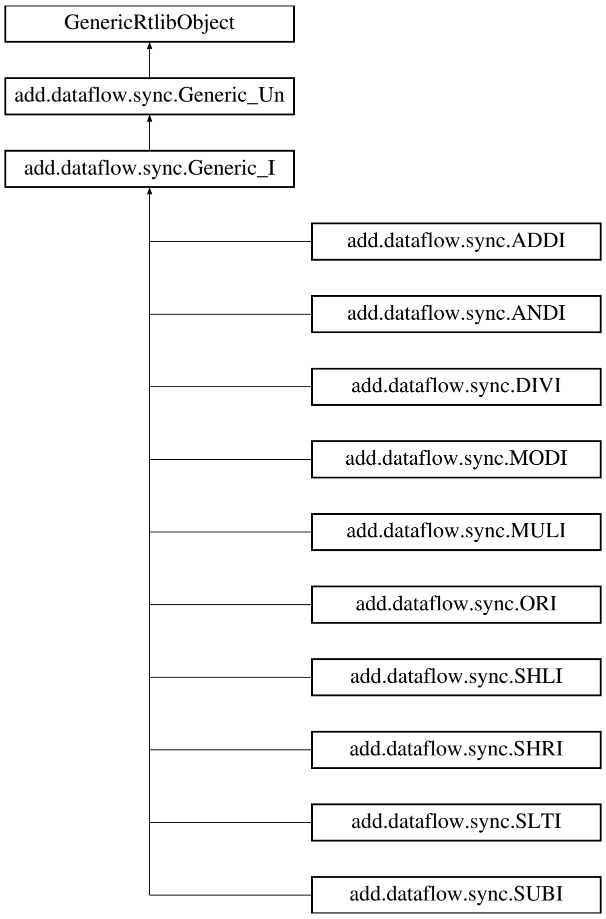
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_Compl.java

3.31 add.dataflow.sync.Generic_I Class Reference

Inheritance diagram for add.dataflow.sync.Generic_I:



Public Member Functions

- Generic_I ()
- int Compute (int data)
- void NotCompute ()
- void Reseted ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)
- void mousePressed (java.awt.event.MouseEvent me)

Protected Attributes

- int **constante**

3.31.1 Detailed Description

Generic_I component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components with an input and that perform the computation with an (immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.31.2 Constructor & Destructor Documentation

3.31.2.1 add.dataflow.sync.Generic_I.Generic_I ( )

Object Constructor.

3.31.3 Member Function Documentation

3.31.3.1 int add.dataflow.sync.Generic_I.Compute (int *data*)

Method responsible for the computation of the output and set the new text to be shown by the component. In this case the constant.

Parameters

data	- Value to be used for the computation.

Returns

- Return of computation

3.31.3.2 boolean add.dataflow.sync.Generic_I.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.31.3.3 void add.dataflow.sync.Generic_I.mousePressed (java.awt.event.MouseEvent *me*)

Method responsible for changing the value of the constant for more or less, depending on whether the mouse click is done by the right or left button respectively.

Parameters

me	- Object where the event occurred.

3.31.3.4 void add.dataflow.sync.Generic_I.NotCompute ( )

Method executed when computing is not performed.

3.31.3.5 void add.dataflow.sync.Generic_I.Reseted ( )

Method executed when the signal from the reset input goes to high logic level. It sets the new text to be shown by the component. In this case the constant.

3.31.3.6 void add.dataflow.sync.Generic_I.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

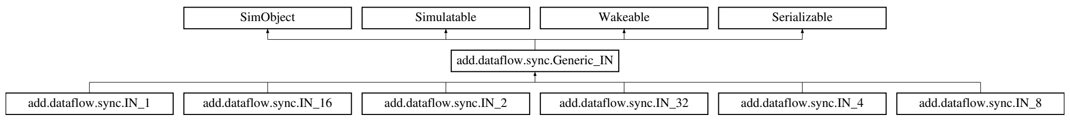
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_I.java

3.32 add.dataflow.sync.Generic_IN Class Reference

Inheritance diagram for add.dataflow.sync.Generic_IN:



Public Member Functions

- Generic_IN ()
- Generic_IN (int QTDE_PORTS)
- void constructPorts ()
- void setVectorIn (int[] vector)
- void Set_Name (String l)
- void evaluate (Object arg)
- boolean needsDynamicSymbol ()
- void constructDynamicSymbol ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)
- double getDelay ()
- void setDelay (double _delay)
- void setDelay (String s)
- void wakeup (Object arg)
- void updateSymbol ()

Protected Member Functions

- void constructStandardValues ()

Protected Attributes

- int **n_bits** = 16
- StdLogicVector **vector**
- PortStdLogicVector **vectorOutputPort**
- double **delay**
- double **defaultdelay** = 10E-9
- boolean **enableAnimationFlag**
- ColoredValueLabel **valueLabel**
- FlexibleLabelFormatter **labelFormatter**
- PortStdLogic1164 **port_CLK**
- PortStdLogicVector[] **port_DATA_OUT**
- PortStdLogic1164[] **port_R_OUT**
- int[] **vector_in**
- int **idx_data_in**

3.32.1 Detailed Description

Generic_IN component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components that implement input queues with 1, 2, 4, 8, 16, or 32 outputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.32.2 Constructor & Destructor Documentation

3.32.2.1 add.dataflow.sync.Generic_IN.Generic_IN ( )

Object Constructor. By default, an input queue of an output is created.

3.32.2.2 add.dataflow.sync.Generic_IN.Generic_IN (int *QTDE_PORTS*)

Object Constructor. An input queue of N outputs is created.

Parameters

QTDE_PORTS	- Number of queue outputs to be created

3.32.3 Member Function Documentation

3.32.3.1 void add.dataflow.sync.Generic_IN.constructDynamicSymbol ( )

Method responsible for dynamically constructing the component symbol.

3.32.3.2 void add.dataflow.sync.Generic_IN.constructPorts ( )

Method responsible for initializing the component input and output ports.

3.32.3.3 void add.dataflow.sync.Generic_IN.constructStandardValues ( ) [protected]

Method responsible for creating some auxiliary variables for working with bit vectors.

3.32.3.4 void add.dataflow.sync.Generic_IN.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked whether the ports are connected. It Will pass the vector data to the outputs.

Parameters

arg	an arbitrary object argument

3.32.3.5 double add.dataflow.sync.Generic_IN.getDelay ( )

Method responsible for returning the value of the delay variable that contains the response delay time of the component.

3.32.3.6 boolean add.dataflow.sync.Generic_IN.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.32.3.7 boolean add.dataflow.sync.Generic_IN.needsDynamicSymbol ( )

Method responsible for indicating to the simulator that the component's symbol will be constructed dynamically by the constructDynamicSymbol() method, or will be read from a file of the same name as the ".sym" extension.

3.32.3.8 void add.dataflow.sync.Generic_IN.Set_Name (String *l*)

Method responsible for changing the label that displays the name of the component.

3.32.3.9 void add.dataflow.sync.Generic_IN.setDelay (double *_delay*)

Method responsible for changing the value of the delay variable that contains the response delay time of the component.

Parameters

_delay	

3.32.3.10 void add.dataflow.sync.Generic_IN.setDelay (String *s*)

Method responsible for changing the value of the delay variable that contains the response delay time of the component.

Parameters

s	

3.32.3.11 void add.dataflow.sync.Generic_IN.setVectorIn (int[ ] *vector*)

Method responsible to set the data vector to be delivered to the outputs.

Parameters

vector	- Vector that will be delivered to the outputs

3.32.3.12 void add.dataflow.sync.Generic_IN.updateSymbol ( )

Method responsible for updating the component symbol.

3.32.3.13 void add.dataflow.sync.Generic_IN.wakeup (Object *arg*)

wakeup(): Called by the simulator as a reaction to our own scheduleWakeup()-calls. For RTLIB components, a wakeup() is normally used to update the value label on its graphical symbol. A WakeupEvent for this purpose should have either 'null' or the current 'this' object as its payload.

A second use is to update our internal 'vector' variable at a specified simulation time, which is needed to implement the assign() method from interface hades.simulator.Assignable. A WakeupEvent for this purpose is expected to hold a StdLogicVector object (with the 'value' from the assign call) as its payload.

3.32.3.14 void add.dataflow.sync.Generic_IN.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

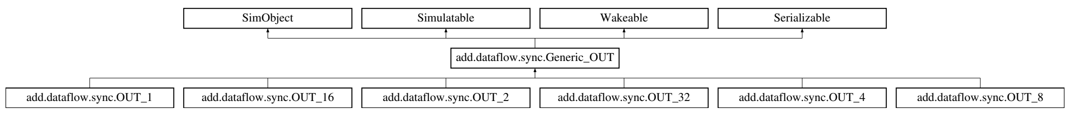
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_IN.java

3.33 add.dataflow.sync.Generic_OUT Class Reference

Inheritance diagram for add.dataflow.sync.Generic_OUT:



Public Member Functions

- Generic_OUT ()
- Generic_OUT (int QTDE_PORTS)
- void constructPorts ()
- boolean getDoneSignal ()
- void setVetor (int k)
- int[] getVectorOut ()
- void Set_Name (String l)
- void evaluate (Object arg)
- boolean needsDynamicSymbol ()
- void constructDynamicSymbol ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)
- double getDelay ()
- void setDelay (double _delay)
- void setDelay (String s)
- void wakeup (Object arg)
- void updateSymbol ()

Protected Member Functions

- void constructStandardValues ()

Protected Attributes

- int **n_bits** = 16
- StdLogicVector **vector**
- PortStdLogicVector **vectorOutputPort**
- double **delay**
- double **defaultdelay** = 10E-9
- boolean **enableAnimationFlag**
- ColoredValueLabel **valueLabel**
- FlexibleLabelFormatter **labelFormatter**
- PortStdLogic1164 **port_CLK**
- PortStdLogicVector[] **port_DATA_IN**
- PortStdLogic1164[] **port_R_IN**
- int[] **vector_out**
- int **idx_data_out**
- int **tamanho_vetor_saida**
- boolean **done** = false

3.33.1 Detailed Description

Generic_OUT component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components that implement output queues with 1, 2, 4, 8, 16, or 32 inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.33.2 Constructor & Destructor Documentation

3.33.2.1 add.dataflow.sync.Generic_OUT.Generic_OUT ( )

Object Constructor. By default, an input queue of an output is created.

3.33.2.2 add.dataflow.sync.Generic_OUT.Generic_OUT (int *QTDE_PORTS*)

Object Constructor. An output queue of N inputs is created.

Parameters

QTDE_PORTS	- Number of queue inputs to be created

3.33.3 Member Function Documentation

3.33.3.1 void add.dataflow.sync.Generic_OUT.constructDynamicSymbol ( )

Method responsible for dynamically constructing the component symbol.

3.33.3.2 void add.dataflow.sync.Generic_OUT.constructPorts ( )

Method responsible for initializing the component input and output ports.

3.33.3.3 void add.dataflow.sync.Generic_OUT.constructStandardValues ( ) [protected]

Method responsible for creating some auxiliary variables for working with bit vectors.

3.33.3.4 void add.dataflow.sync.Generic_OUT.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked whether the ports are connected and if the R_IN inputs are high level. It Will pass the data from the inputs to the vector.

Parameters

arg	an arbitrary object argument

3.33.3.5 double add.dataflow.sync.Generic_OUT.getDelay ( )

Method responsible for returning the value of the delay variable that contains the response delay time of the component.

3.33.3.6 boolean add.dataflow.sync.Generic_OUT.getDoneSignal ( )

Method responsible for returning end of data entry.

3.33.3.7 int [ ] add.dataflow.sync.Generic_OUT.getVectorOut ( )

Method responsible for returning the data vector received by the queue entries.

3.33.3.8 boolean add.dataflow.sync.Generic_OUT.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.33.3.9 boolean add.dataflow.sync.Generic_OUT.needsDynamicSymbol ( )

Method responsible for indicating to the simulator that the component's symbol will be constructed dynamically by the constructDynamicSymbol() method, or will be read from a file of the same name as the ".sym" extension.

3.33.3.10 void add.dataflow.sync.Generic_OUT.Set_Name (String *l*)

Method responsible for changing the label that displays the name of the component.

3.33.3.11 void add.dataflow.sync.Generic_OUT.setDelay (double *_delay*)

Method responsible for changing the value of the delay variable that contains the response delay time of the component.

Parameters

_delay	

3.33.3.12 void add.dataflow.sync.Generic_OUT.setDelay (String *s*)

Method responsible for changing the value of the delay variable that contains the response delay time of the component.

Parameters

s	

3.33.3.13 void add.dataflow.sync.Generic_OUT.setVetor (int *k*)

Method responsible for inserting elements into the vector.

3.33.3.14 void add.dataflow.sync.Generic_OUT.updateSymbol ( )

Method responsible for updating the component symbol.

3.33.3.15 void add.dataflow.sync.Generic_OUT.wakeup (Object *arg*)

wakeup(): Called by the simulator as a reaction to our own scheduleWakeup()-calls. For RTLIB components, a wakeup() is normally used to update the value label on its graphical symbol. A WakeupEvent for this purpose should have either 'null' or the current 'this' object as its payload.

A second use is to update our internal 'vector' variable at a specified simulation time, which is needed to implement the assign() method from interface hades.simulator.Assignable. A WakeupEvent for this purpose is expected to hold a StdLogicVector object (with the 'value' from the assign call) as its payload.

3.33.3.16 void add.dataflow.sync.Generic_OUT.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

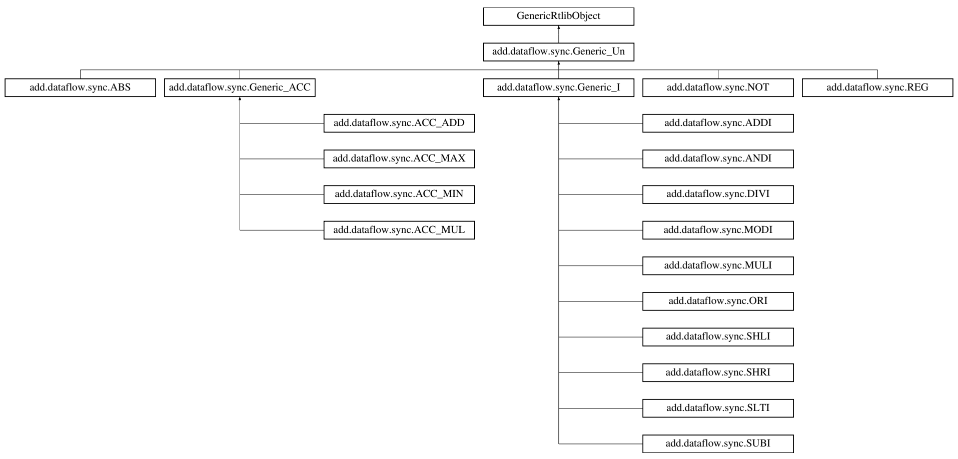
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_OUT.java

3.34 add.dataflow.sync.Generic_Un Class Reference

Inheritance diagram for add.dataflow.sync.Generic_Un:



Public Member Functions

- Generic_Un ()
- void constructPorts ()
- void setString (String s)
- void setSymbol (Symbol s)
- int Compute (int data)
- void NotCompute ()
- void Reseted ()
- void Tick_Up ()
- void Tick_Down ()
- void Set_Name (String l)
- void evaluate (Object arg)
- boolean needsDynamicSymbol ()

- void constructDynamicSymbol ()
- void write (java.io.PrintWriter ps)
- boolean initialize (String s)

Protected Attributes

- Label **stringLabel**
- String **s**
- Rectangle **background**
- PortStdLogic1164 **port_CLK**
- PortStdLogicVector **port_DATA_IN**

3.34.1 Detailed Description

Generic_Un component for the UFV ADD synchronous data flow simulator.

The component creates the basis for other components with one input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.34.2 Constructor & Destructor Documentation

3.34.2.1 add.dataflow.sync.Generic_Un.Generic_Un ( )

Object Constructor.

3.34.3 Member Function Documentation

3.34.3.1 int add.dataflow.sync.Generic_Un.Compute (int *data*)

Method responsible for the computation of the output.

Parameters

data	- Value to be used for the computation.

Returns

- Return of computation

3.34.3.2 void add.dataflow.sync.Generic_Un.constructDynamicSymbol ( )

Method responsible for dynamically constructing the component symbol.

3.34.3.3 void add.dataflow.sync.Generic_Un.constructPorts ( )

Method responsible for initializing the component input and output ports.

3.34.3.4 void add.dataflow.sync.Generic_Un.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked whether the ports are connected and will execute the Compute (int data) method if the R_IN input is high level. It will execute the Reseted(), Tick_Up(), and Tick_Down() methods if their respective entries order it. It will update the output with the Compute(int data) method result.

Parameters

arg	an arbitrary object argument

3.34.3.5 boolean add.dataflow.sync.Generic_Un.initialize (String *s*)

Method responsible for reading the component settings in the file saved by the simulator.

Parameters

s	- Settings for the component read from the file saved by the simulator.

Returns

- Returns true if the settings are read successfully.

3.34.3.6 boolean add.dataflow.sync.Generic_Un.needsDynamicSymbol ( )

Method responsible for indicating to the simulator that the component's symbol will be constructed dynamically by the constructDynamicSymbol() method, or will be read from a file of the same name as the ".sym" extension.

3.34.3.7 void add.dataflow.sync.Generic_Un.NotCompute ( )

Method executed when computing is not performed. In this case it clears the text displayed by the component.

3.34.3.8 void add.dataflow.sync.Generic_Un.Reseted ( )

Method executed when the signal from the reset input goes to high logic level.In this case it clears the text displayed by the component.

3.34.3.9 void add.dataflow.sync.Generic_Un.Set_Name (String *l*)

Method responsible for changing the label that displays the name of the component.

3.34.3.10 void add.dataflow.sync.Generic_Un.setString (String *s*)

Method responsible for updating the text displayed by the component.

Parameters

s	- Text to be updated.

3.34.3.11 void add.dataflow.sync.Generic_Un.setSymbol (Symbol *s*)

Method responsible for updating the component symbol.

3.34.3.12 void add.dataflow.sync.Generic_Un.Tick_Down ( )

Method executed when the clock signal goes to low logic level.

3.34.3.13 void add.dataflow.sync.Generic_Un.Tick_Up ( )

Method executed when the clock signal goes to high logic level.

3.34.3.14 void add.dataflow.sync.Generic_Un.write (java.io.PrintWriter *ps*)

Method responsible for writing component settings to the file saved by the simulator.

Parameters

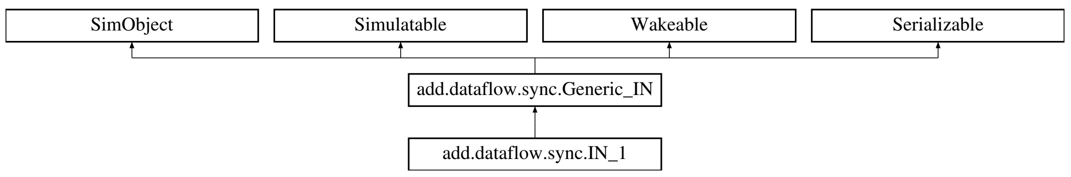
ps	-Simulator writing object.

The documentation for this class was generated from the following file:

- add/dataflow/sync/Generic_Un.java

3.35 add.dataflow.sync.IN_1 Class Reference

Inheritance diagram for add.dataflow.sync.IN_1:



Public Member Functions

- IN_1 ()

Additional Inherited Members

3.35.1 Detailed Description

IN_1 component for the UFV ADD synchronous data flow simulator.

The component implements an input queue with 1 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.35.2 Constructor & Destructor Documentation

3.35.2.1 add.dataflow.sync.IN_1.IN_1 ( )

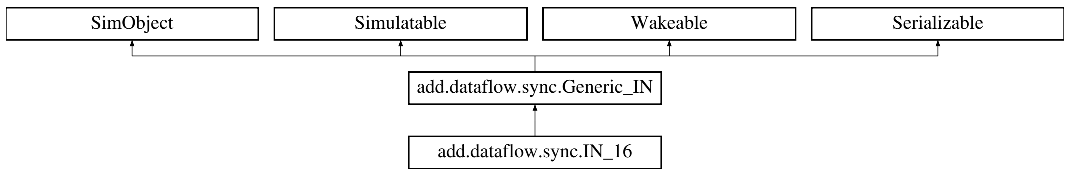
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/IN_1.java

3.36 add.dataflow.sync.IN_16 Class Reference

Inheritance diagram for add.dataflow.sync.IN_16:



Public Member Functions

- IN_16 ()

Additional Inherited Members

3.36.1 Detailed Description

IN_16 component for the UFV ADD synchronous data flow simulator.

The component implements an input queue with 16 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.36.2 Constructor & Destructor Documentation

3.36.2.1 add.dataflow.sync.IN_16.IN_16 ( )

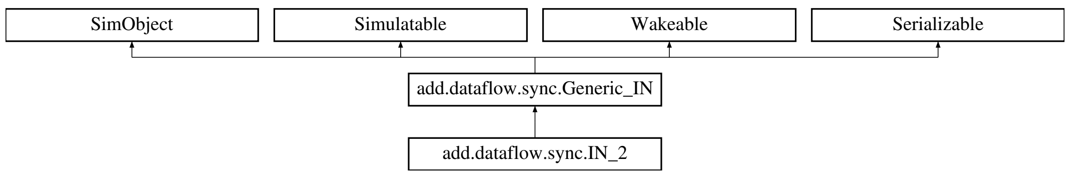
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/IN_16.java

3.37 add.dataflow.sync.IN_2 Class Reference

Inheritance diagram for add.dataflow.sync.IN_2:



Public Member Functions

- IN_2 ()

Additional Inherited Members

3.37.1 Detailed Description

IN_2 component for the UFV ADD synchronous data flow simulator.

The component implements an input queue with 2 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.37.2 Constructor & Destructor Documentation

3.37.2.1 add.dataflow.sync.IN_2.IN_2 ( )

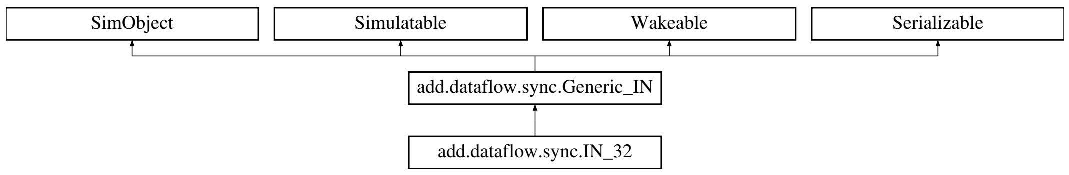
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/IN_2.java

3.38 add.dataflow.sync.IN_32 Class Reference

Inheritance diagram for add.dataflow.sync.IN_32:



Public Member Functions

- IN_32 ()

Additional Inherited Members

3.38.1 Detailed Description

IN_32 component for the UFV ADD synchronous data flow simulator.

The component implements an input queue with 32 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.38.2 Constructor & Destructor Documentation

3.38.2.1 add.dataflow.sync.IN_32.IN_32 ( )

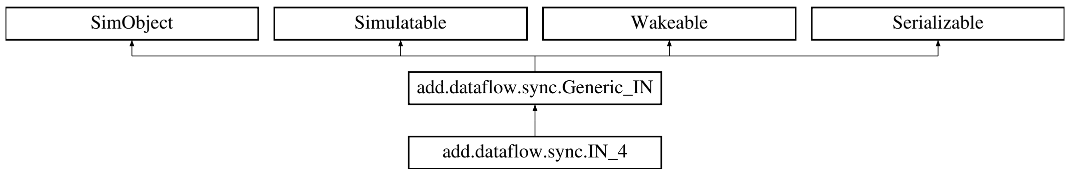
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/IN_32.java

3.39 add.dataflow.sync.IN_4 Class Reference

Inheritance diagram for add.dataflow.sync.IN_4:



Public Member Functions

- IN_4 ()

Additional Inherited Members

3.39.1 Detailed Description

IN_4 component for the UFV ADD synchronous data flow simulator.

The component implements an input queue with 4 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.39.2 Constructor & Destructor Documentation

3.39.2.1 add.dataflow.sync.IN_4.IN_4 ( )

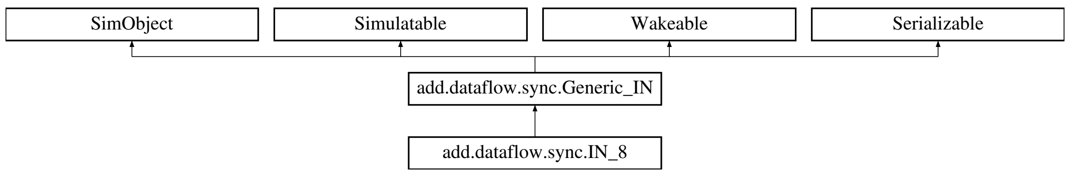
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/IN_4.java

3.40 add.dataflow.sync.IN_8 Class Reference

Inheritance diagram for add.dataflow.sync.IN_8:



Public Member Functions

- IN_8 ()

Additional Inherited Members

3.40.1 Detailed Description

IN_8 component for the UFV ADD synchronous data flow simulator.

The component implements an input queue with 8 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.40.2 Constructor & Destructor Documentation

3.40.2.1 add.dataflow.sync.IN_8.IN_8 ( )

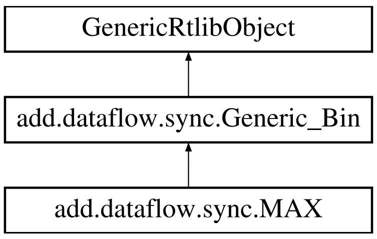
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/IN_8.java

3.41 add.dataflow.sync.MAX Class Reference

Inheritance diagram for add.dataflow.sync.MAX:



Public Member Functions

- MAX ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.41.1 Detailed Description

MAX component for the UFV ADD synchronous data flow simulator.

The component is responsible for passing the output to the largest value input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.41.2 Constructor & Destructor Documentation

3.41.2.1 add.dataflow.sync.MAX.MAX ( )

Object Constructor.

3.41.3 Member Function Documentation

3.41.3.1 int add.dataflow.sync.MAX.Compute (int *data1*, int *data2*)

Method responsible for the computation of components: in this case, it performs a comparison between the parameters and returns the largest between the two.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

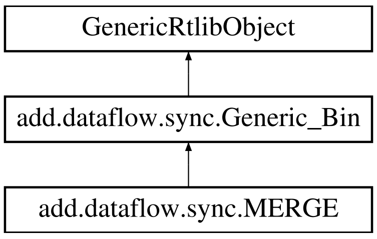
- Returns the result of the computation. In this case, the largest of the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/MAX.java

3.42 add.dataflow.sync.MERGE Class Reference

Inheritance diagram for add.dataflow.sync.MERGE:



Public Member Functions

- MERGE ()
- void evaluate (Object arg)

Additional Inherited Members

3.42.1 Detailed Description

MERGE component for the UFV ADD synchronous data flow simulator.

The component is responsible for choosing which of the inputs to pass to the output depending on the value of R_IN1 and R_IN2.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.42.2 Constructor & Destructor Documentation

3.42.2.1 add.dataflow.sync.MERGE.MERGE ( )

Object Constructor.

3.42.3 Member Function Documentation

3.42.3.1 void add.dataflow.sync.MERGE.evaluate (Object *arg*)

evaluate(): called by the simulation engine on all events that concern this object. The object is responsible for updating its internal state and for scheduling all pending output events. In this case, it will be checked if any of the R_IN (1 or 2) inputs is at high level and put the respective input value in the output. If the two R_IN signals are at high level, the value of input 1 will be set to the output. If both are 0, nothing will be done.

Parameters

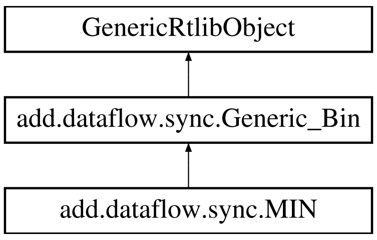
arg	an arbitrary object argument

The documentation for this class was generated from the following file:

- add/dataflow/sync/MERGE.java

3.43 add.dataflow.sync.MIN Class Reference

Inheritance diagram for add.dataflow.sync.MIN:



Public Member Functions

- MIN ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.43.1 Detailed Description

MIN component for the UFV ADD synchronous data flow simulator.

The component is responsible for passing the output to the lowest value input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.43.2 Constructor & Destructor Documentation

3.43.2.1 add.dataflow.sync.MIN.MIN ( )

Object Constructor.

3.43.3 Member Function Documentation

3.43.3.1 int add.dataflow.sync.MIN.Compute (int *data1,* int *data2*)

Method responsible for the computation of components: in this case, it performs a comparison between the parameters and returns the smaller between the two.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

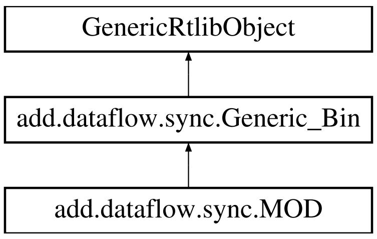
- Returns the result of the computation. In this case, the smallest of the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/MIN.java

3.44 add.dataflow.sync.MOD Class Reference

Inheritance diagram for add.dataflow.sync.MOD:



Public Member Functions

- MOD ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.44.1 Detailed Description

MOD component for the UFV ADD synchronous data flow simulator.

The component is responsible for calculating the rest of the integer division of the first input by the second one.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.44.2 Constructor & Destructor Documentation

3.44.2.1 add.dataflow.sync.MOD.MOD ( )

Object Constructor.

3.44.3 Member Function Documentation

3.44.3.1 int add.dataflow.sync.MOD.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case, it returns the rest of the division between the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

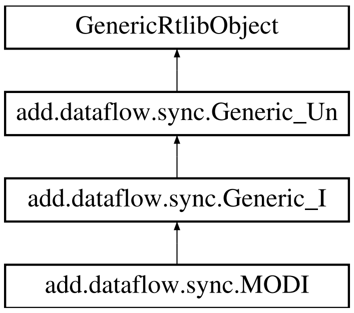
- Returns the result of the computation. In this case, it returns the rest of the division between the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/MOD.java

3.45 add.dataflow.sync.MODI Class Reference

Inheritance diagram for add.dataflow.sync.MODI:



Public Member Functions

- MODI ()
- int Compute (int data)

Additional Inherited Members

3.45.1 Detailed Description

MODI component for the UFV ADD synchronous data flow simulator.

The component is responsible for calculating the rest of the integer division of the input by a constant (Immediate).

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.45.2 Constructor & Destructor Documentation

3.45.2.1 add.dataflow.sync.MODI.MODI ( )

Object Constructor.

3.45.3 Member Function Documentation

3.45.3.1 int add.dataflow.sync.MODI.Compute (int *data*)

Method responsible for the component computation: in this case, it returns the rest of the division of the parameter by the constant.

Parameters

data	- Value to be used for computing.

Returns

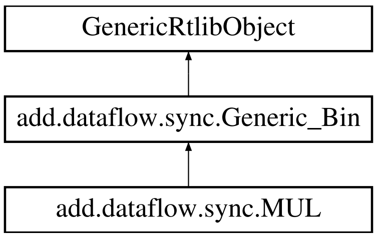
- Returns the result of the computation. In this case, it returns the rest of the division of the parameter by the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/MODI.java

3.46 add.dataflow.sync.MUL Class Reference

Inheritance diagram for add.dataflow.sync.MUL:



Public Member Functions

- MUL ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.46.1 Detailed Description

MUL component for the UFV ADD synchronous data flow simulator.

The component is responsible for multiplying the inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.46.2 Constructor & Destructor Documentation

3.46.2.1 add.dataflow.sync.MUL.MUL ( )

Object Constructor.

3.46.3 Member Function Documentation

3.46.3.1 int add.dataflow.sync.MUL.Compute (int *data1,* int *data2*)

Method responsible for the component computation: in this case performs a multiplication of the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

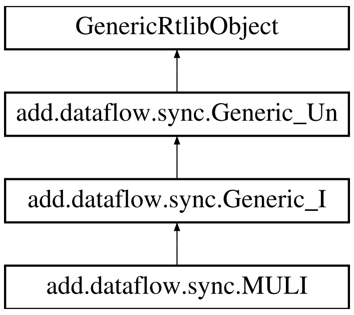
- Returns the result of the computation. In this case the value of the multiplication of the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/MUL.java

3.47 add.dataflow.sync.MULI Class Reference

Inheritance diagram for add.dataflow.sync.MULI:



Public Member Functions

- MULI ()
- int Compute (int data)

Additional Inherited Members

3.47.1 Detailed Description

MULI component for the UFV ADD synchronous data flow simulator.

The component is responsible for multiplying the input by a (Immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.47.2 Constructor & Destructor Documentation

3.47.2.1 add.dataflow.sync.MULI.MULI ( )

Object Constructor.

3.47.3 Member Function Documentation

3.47.3.1 int add.dataflow.sync.MULI.Compute (int *data*)

Method responsible for the component computation: in this case performs a multiplying of the parameter by an (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

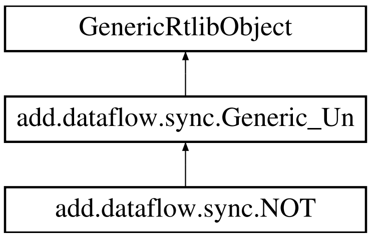
- Returns the result of the computation. In this case the value of the multiplication of the parameter by the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/MULI.java

3.48 add.dataflow.sync.NOT Class Reference

Inheritance diagram for add.dataflow.sync.NOT:



Public Member Functions

- NOT ()
- int Compute (int data)

Additional Inherited Members

3.48.1 Detailed Description

NOT component for the UFV ADD synchronous data flow simulator.

The component is responsible for the bitwise inversion of the input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.48.2 Constructor & Destructor Documentation

3.48.2.1 add.dataflow.sync.NOT.NOT ( )

Object Constructor.

3.48.3 Member Function Documentation

3.48.3.1 int add.dataflow.sync.NOT.Compute (int *data*)

Method responsible for the component computation: in this case performs a bitwise inversion of the parameter.

Parameters

data	- Value to be used for computing.

Returns

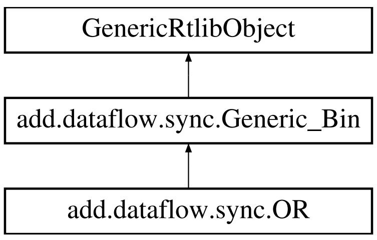
- Returns the result of the computation. In this case the value of the bitwise inversion of the parameter.

The documentation for this class was generated from the following file:

- add/dataflow/sync/NOT.java

3.49 add.dataflow.sync.OR Class Reference

Inheritance diagram for add.dataflow.sync.OR:



Public Member Functions

- OR ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.49.1 Detailed Description

OR component for the UFV ADD synchronous data flow simulator.

The component is responsible for the logical operation "OR" between the input

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.49.2 Constructor & Destructor Documentation

3.49.2.1 add.dataflow.sync.OR.OR ( )

Object Constructor.

3.49.3 Member Function Documentation

3.49.3.1 int add.dataflow.sync.OR.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case it performs the logical operation "OR" between the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

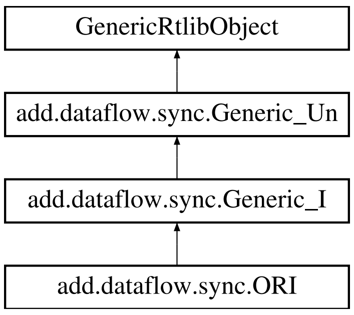
- Returns the result of the computation. In this case the result of the logical operation "OR" between the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OR.java

3.50 add.dataflow.sync.ORI Class Reference

Inheritance diagram for add.dataflow.sync.ORI:



Public Member Functions

- ORI ()
- int Compute (int data)

Additional Inherited Members

3.50.1 Detailed Description

ORI component for the UFV ADD synchronous data flow simulator.

The component is responsible for the logical operation "OR" between the input and a constant (Immediate)

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.50.2 Constructor & Destructor Documentation

3.50.2.1 add.dataflow.sync.ORI.ORI ( )

Object Constructor.

3.50.3 Member Function Documentation

3.50.3.1 int add.dataflow.sync.ORI.Compute (int *data*)

Method responsible for the component computation: in this case it performs the logical operation "OR" between the parameter and the (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

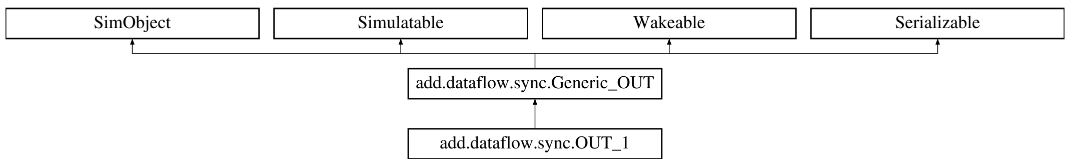
- Returns the result of the computation. In this case the result of the logical operation "OR" between the parameter and the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/ORI.java

3.51 add.dataflow.sync.OUT_1 Class Reference

Inheritance diagram for add.dataflow.sync.OUT_1:



Public Member Functions

- OUT_1 ()

Additional Inherited Members

3.51.1 Detailed Description

OUT_1 component for the UFV ADD synchronous data flow simulator.

The component implements an output queue with 1 input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.51.2 Constructor & Destructor Documentation

3.51.2.1 add.dataflow.sync.OUT_1.OUT_1 ( )

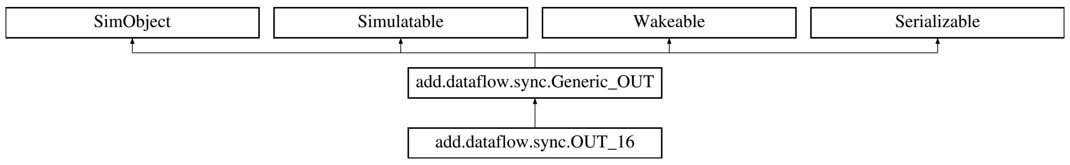
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OUT_1.java

3.52 add.dataflow.sync.OUT_16 Class Reference

Inheritance diagram for add.dataflow.sync.OUT_16:



Public Member Functions

- OUT_16 ()

Additional Inherited Members

3.52.1 Detailed Description

OUT_16 component for the UFV ADD synchronous data flow simulator.

O implements an input queue with 16 output.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.52.2 Constructor & Destructor Documentation

3.52.2.1 add.dataflow.sync.OUT_16.OUT_16 ( )

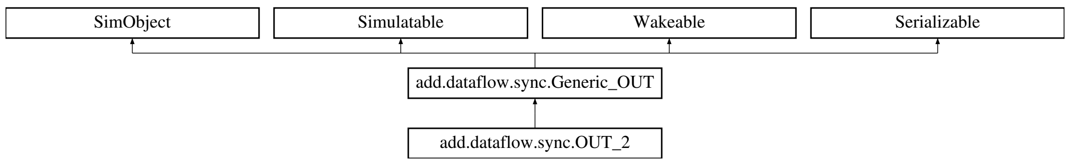
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OUT_16.java

3.53 add.dataflow.sync.OUT_2 Class Reference

Inheritance diagram for add.dataflow.sync.OUT_2:



Public Member Functions

- OUT_2 ()

Additional Inherited Members

3.53.1 Detailed Description

OUT_2 component for the UFV ADD synchronous data flow simulator.

The component implements an output queue with 2 input..

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.53.2 Constructor & Destructor Documentation

3.53.2.1 add.dataflow.sync.OUT_2.OUT_2 ( )

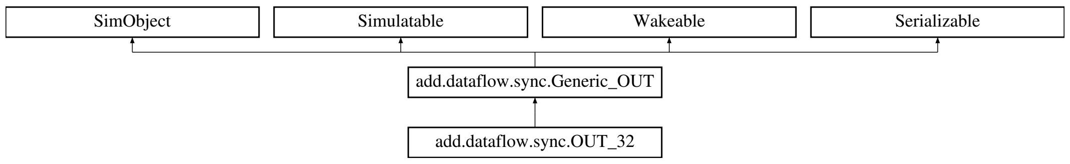
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OUT_2.java

3.54 add.dataflow.sync.OUT_32 Class Reference

Inheritance diagram for add.dataflow.sync.OUT_32:



Public Member Functions

- OUT_32 ()

Additional Inherited Members

3.54.1 Detailed Description

OUT_32 component for the UFV ADD synchronous data flow simulator.

The component implements an output queue with 32 input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.54.2 Constructor & Destructor Documentation

3.54.2.1 add.dataflow.sync.OUT_32.OUT_32 ( )

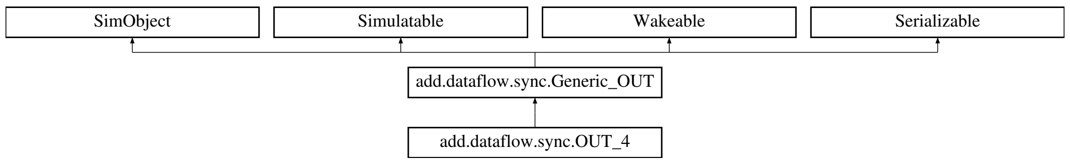
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OUT_32.java

3.55 add.dataflow.sync.OUT_4 Class Reference

Inheritance diagram for add.dataflow.sync.OUT_4:



Public Member Functions

- OUT_4 ()

Additional Inherited Members

3.55.1 Detailed Description

OUT_4 component for the UFV ADD synchronous data flow simulator.

The component implements an output queue with 4 input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.55.2 Constructor & Destructor Documentation

3.55.2.1 add.dataflow.sync.OUT_4.OUT_4 ( )

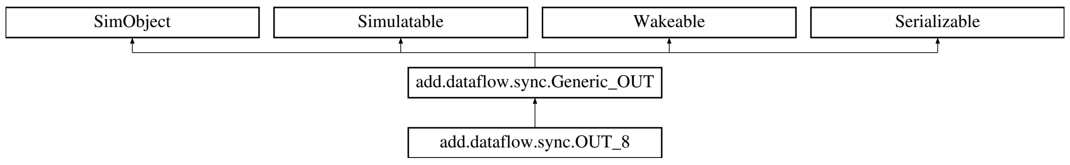
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OUT_4.java

3.56 add.dataflow.sync.OUT_8 Class Reference

Inheritance diagram for add.dataflow.sync.OUT_8:



Public Member Functions

- OUT_8 ()

Additional Inherited Members

3.56.1 Detailed Description

OUT_8 component for the UFV ADD synchronous data flow simulator.

The component implements an output queue with 8 input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.56.2 Constructor & Destructor Documentation

3.56.2.1 add.dataflow.sync.OUT_8.OUT_8 ( )

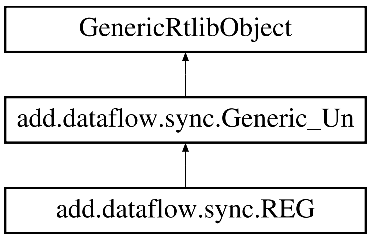
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/OUT_8.java

3.57 add.dataflow.sync.REG Class Reference

Inheritance diagram for add.dataflow.sync.REG:



Public Member Functions

- REG ()

Additional Inherited Members

3.57.1 Detailed Description

REG component for the UFV ADD synchronous data flow simulator.

The component is responsible for pass the input to the output when a clock pulse occurs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.57.2 Constructor & Destructor Documentation

3.57.2.1 add.dataflow.sync.REG.REG ( )

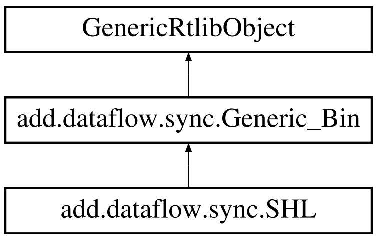
Object Constructor.

The documentation for this class was generated from the following file:

- add/dataflow/sync/REG.java

3.58 add.dataflow.sync.SHL Class Reference

Inheritance diagram for add.dataflow.sync.SHL:



Public Member Functions

- SHL ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.58.1 Detailed Description

SHL component for the UFV ADD synchronous data flow simulator.

The component is responsible for moving all the bits of the first input to the left N times, where N is equal to the value of the second input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.58.2 Constructor & Destructor Documentation

3.58.2.1 add.dataflow.sync.SHL.SHL ( )

Object Constructor.

3.58.3 Member Function Documentation

3.58.3.1 int add.dataflow.sync.SHL.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case, it moves all the bits of the first parameter to the left N times, where N is equal to the value of the second parameter.

Parameters

data	- Value to be used for computing.

Returns

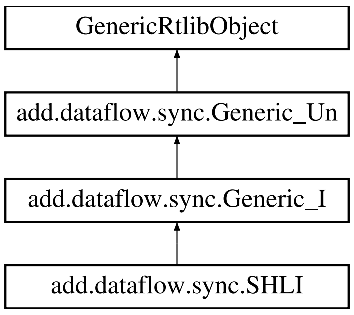
- Returns the result of the computation. In this case, it moves all the bits of the first parameter to the left N times, where N is equal to the value of second parameter.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SHL.java

3.59 add.dataflow.sync.SHLI Class Reference

Inheritance diagram for add.dataflow.sync.SHLI:



Public Member Functions

- SHLI ()
- int Compute (int data)

Additional Inherited Members

3.59.1 Detailed Description

SHLI component for the UFV ADD synchronous data flow simulator.

The component is responsible for moving all the bits of the input to the left N times, where N is equal to the value of a (immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.59.2 Constructor & Destructor Documentation

3.59.2.1 add.dataflow.sync.SHLI.SHLI ( )

Object Constructor.

3.59.3 Member Function Documentation

3.59.3.1 int add.dataflow.sync.SHLI.Compute (int *data*)

Method responsible for the component computation: in this case, it moves all the bits of the parameter to the left N times, where N is equal to the value of a (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

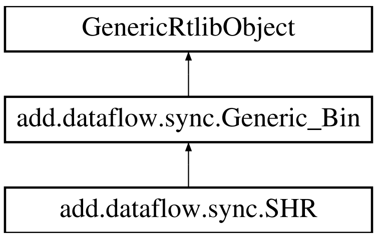
- Returns the result of the computation. In this case, it moves all the bits of the parameter to the left N times, where N is equal to the value of a (immediate) constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SHLI.java

3.60 add.dataflow.sync.SHR Class Reference

Inheritance diagram for add.dataflow.sync.SHR:



Public Member Functions

- SHR ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.60.1 Detailed Description

SHR component for the UFV ADD synchronous data flow simulator.

The component is responsible for moving all the bits of the first input to the right N times, where N is equal to the value of the value from the second input.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.60.2 Constructor & Destructor Documentation

3.60.2.1 add.dataflow.sync.SHR.SHR ( )

Object Constructor.

3.60.3 Member Function Documentation

3.60.3.1 int add.dataflow.sync.SHR.Compute (int *data1,* int *data2*)

Method responsible for the component computation: in this case, it moves all the bits of the first parameter to the right N times, where N is equal to the value of the second parameter.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

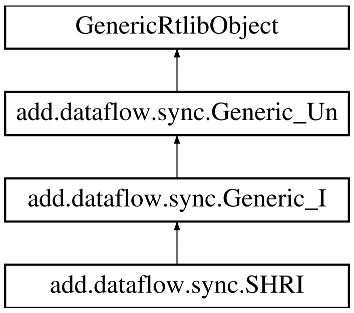
- Returns the result of the computation. In this case, it moves all the bits of the first parameter to the right N times, where N is equal to the value of the second parameter.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SHR.java

3.61 add.dataflow.sync.SHRI Class Reference

Inheritance diagram for add.dataflow.sync.SHRI:



Public Member Functions

- SHRI ()
- int Compute (int data)

Additional Inherited Members

3.61.1 Detailed Description

SHRI component for the UFV ADD synchronous data flow simulator.

The component is responsible for moving all the bits of the input to the right N times, where N is equal to the value of a (immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.61.2 Constructor & Destructor Documentation

3.61.2.1 add.dataflow.sync.SHRI.SHRI ( )

Object Constructor.

3.61.3 Member Function Documentation

3.61.3.1 int add.dataflow.sync.SHRI.Compute (int *data*)

Method responsible for the component computation: in this case, it moves all the bits of the parameter to the right N times, where N is equal to the value of a (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

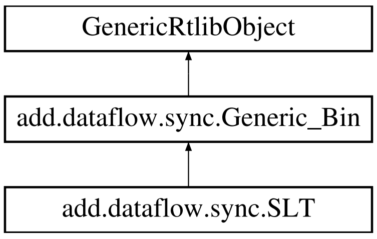
- Returns the result of the computation. In this case, it moves all the bits of the parameter to the right N times, where N is equal to the value of a (immediate) constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SHRI.java

3.62 add.dataflow.sync.SLT Class Reference

Inheritance diagram for add.dataflow.sync.SLT:



Public Member Functions

- SLT ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.62.1 Detailed Description

SLT component for the UFV ADD synchronous data flow simulator.

The component is responsible for returning the value 1 if the first input is less than the second one.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - jeronimopenha@gmail.com
Ricardo Santos Ferreira - cacauvicosa@gmail.com

Version

1.0

3.62.2 Constructor & Destructor Documentation

3.62.2.1 add.dataflow.sync.SLT.SLT ( )

Object Constructor.

3.62.3 Member Function Documentation

3.62.3.1 int add.dataflow.sync.SLT.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case performs a comparison if the first parameter is less than the other one.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

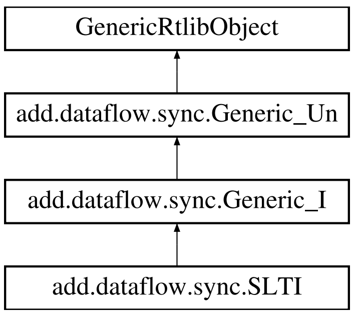
- Returns the result of the computation. In this case 1 or 0 depending on the comparison between the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SLT.java

3.63 add.dataflow.sync.SLTI Class Reference

Inheritance diagram for add.dataflow.sync.SLTI:



Public Member Functions

- SLTI ()
- int Compute (int data)

Additional Inherited Members

3.63.1 Detailed Description

SLTI component for the UFV ADD synchronous data flow simulator.

The component is responsible for returning the value 1 if the input is less than the (immediate) constant.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.63.2 Constructor & Destructor Documentation

3.63.2.1 add.dataflow.sync.SLTI.SLTI ( )

Object Constructor.

3.63.3 Member Function Documentation

3.63.3.1 int add.dataflow.sync.SLTI.Compute (int *data*)

Method responsible for the component computation: in this case performs a comparison if parameter is less than the (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

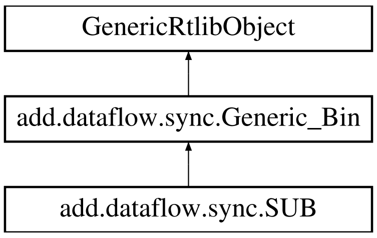
- Returns the result of the computation. In this case 1 or 0 depending on the comparison between the parameter and the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SLTI.java

3.64 add.dataflow.sync.SUB Class Reference

Inheritance diagram for add.dataflow.sync.SUB:



Public Member Functions

- SUB ()
- int Compute (int data1, int data2)

Additional Inherited Members

3.64.1 Detailed Description

SUB component for the UFV ADD synchronous data flow simulator.

The component is responsible for subtracting the inputs.

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.64.2 Constructor & Destructor Documentation

3.64.2.1 add.dataflow.sync.SUB.SUB ( )

Object Constructor.

3.64.3 Member Function Documentation

3.64.3.1 int add.dataflow.sync.SUB.Compute (int *data1*, int *data2*)

Method responsible for the component computation: in this case performs a subtraction of the parameters.

Parameters

data1	- Value to be used for the computation related to input 1.
data2	- Value to be used for the computation related to input 2.

Returns

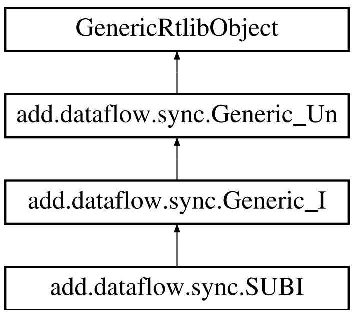
- Returns the result of the computation. In this case the value of the subtraction of the parameters.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SUB.java

3.65 add.dataflow.sync.SUBI Class Reference

Inheritance diagram for add.dataflow.sync.SUBI:



Public Member Functions

- SUBI ()
- int Compute (int data)

Additional Inherited Members

3.65.1 Detailed Description

SUBI component for the UFV ADD synchronous data flow simulator.

The component is responsible for subtracting the input by a constant (Immediate).

Universidade Federal de Viçosa - MG - Brasil.

Author

Jeronimo Costa Penha - `jeronimopenha@gmail.com`
Ricardo Santos Ferreira - `cacauvicosa@gmail.com`

Version

1.0

3.65.2 Constructor & Destructor Documentation

3.65.2.1 add.dataflow.sync.SUBI.SUBI ( )

Object Constructor.

3.65.3 Member Function Documentation

3.65.3.1 int add.dataflow.sync.SUBI.Compute (int *data*)

Method responsible for the component computation: in this case performs a subtraction of the parameter by an (immediate) constant.

Parameters

data	- Value to be used for computing.

Returns

- Returns the result of the computation. In this case the value of the subtraction of the parameter by the constant.

The documentation for this class was generated from the following file:

- add/dataflow/sync/SUBI.java

Index