Name:Kanad Gaikwad
Date:28/05/19

Integer palette

BITCLEAR

Description:

This block sets to 0 the specified bit of its input. The user gives the bit index in the field Index of Bit with index 0, as that of the least significant bit.The operation can be summarized by this expression:

Output = Input ^{mask}

i.e. a bitwise AND between Input and Mask, an integer of the same length as the input with a value 0 for the bit to clear and a value 1 for the other bits.

Datatypes:

The block supports the following types :

1) Input : scalar. All Scilab's integer type (Data type parameter).
2) Output : same type and dimensions than input.

Parameters:

Name	Description	Value
Data Type	It indicates the integer type of input: between 3 & 8	3:int32,4:int16,5:int8
Index of Bit	Index of bit to clear. 1. int32 or uint32: positive and less than 32. 2. int16 or uint16: positive and less than 16. 3. int8 or uint8: positive and less than 8.	0

Name:Kanad Gaikwad
Date:28/05/19

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[1/1]/type3
regular outputs		port1:size[3/3/type3
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		bit_clear_32

Name:Kanad Gaikwad
Date:28/05/19

BITSET

Description:

This block sets to 1 the specified bit of its input. The user gives the bit index in the field Index of Bit with index 0, as that of the least significant bit.The operation can be summarized by this expression:

Output = Input ^{mask}

i.e. a bitwise AND between Input and Mask, an integer of the same length as the input with a value 1 for the bit to clear and a value 0 for the other bits.

Datatypes:

The block supports the following types :

Input : scalar. All Scilab's integer type (Data type parameter).

Output : same type and dimensions than input.

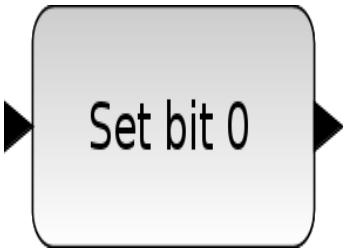
Parameters:

Name	Description	Value
Data Type	It indicates the integer type of input: between 3 & 8	3:int32,4:int16,5:int8
Index of Bit	Index of bit to clear. 1. int32 or uint32: positive and less than 32. 2. int16 or uint16: positive and less than 16. 3. int8 or uint8: positive and less than 8.	

Name:Kanad Gaikwad
Date:28/05/19

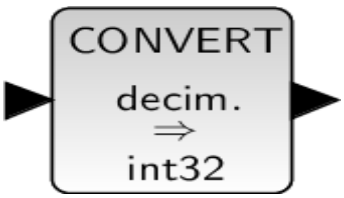
Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[1/1]/type3
regular outputs		port1:size[3/3/type3
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		bit_clear_32



Name:Kanad Gaikwad
Date:28/05/19

CONVERT



Description:

This block converts an input signal of real double or integer data type to an integer or real double data type.

Datatypes:

The block supports the following types :

Input : Scilab's integer and real double data types(Input Type parameter)

Output : Scilab's integer and real double data types(Input Type parameter)

Parameters:

Name	Description	Value
Input Type(a)	It indicates the input data type that it can be a double or an integer;between 1 and 8	1|2:double,3:int32,4:int16, 5:int8,6:unit32,7:unit16,8:unit8
Output Type(a)	It indicates the output data type that it can be a double or an integer;between 1 and 8	1|2:double,3:int32,4:int16, 5:int8,6:unit32,7:unit16,8:unit8
Do on Overflow(b)	When there is over flowing	0: result is not saturated 1:result is saturated 2:error message

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[-1/-2]/type1
regular outputs		port1:size[-1,-2]type3
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no

Name:Kanad Gaikwad
Date:28/05/19

name of computational function		convert

DFLIPFLOP

Description:

That block outputs on Q its input state (D) when the enable input (en) is set and on the rising edge of the clock input (clk). The !Q output is the logical negation of Q. This flip-flop is also known as a delay flip-flop because the state of the input is modified only at the next rising edge of the clock. The D flip-flop is used as a basic cell of shift-registers.

clk	**en**	**D**	**Q_n**	**$!Q_n$**
↑	0	X	Q_{n-1}	$!Q_{n-1}$
↑	1	0	0	1
↑	1	1	1	0
Non-Rising	X	X	Q_{n-1}	$!Q_{n-1}$

where X stands for "indifferent" and Qn-1 is the previous output state of Qn .

Datatypes:

The block supports the following types :

- Inputs:

  D: scalar.Scilab’s int8 data type only.

  en:scalar.Scilab’s real double

  clk:scalar.Scilab’s real double

A positive input is considered as logical 1, a negative or a null input as logical 0.

- Outputs : scalar. Scilab's int8 data type.

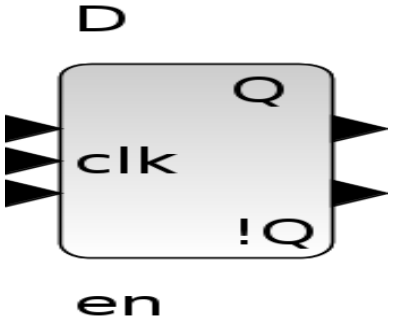
Parameters:

Nil

Name:Kanad Gaikwad
Date:28/05/19

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[1/1]/type5 port2:size[1/1]/type1 port3:size[1/1]/type1
regular outputs		port1:size[1/1]type5 port2:size[1/1]/type5
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		csuper



Name:Kanad Gaikwad
Date:28/05/19

DLATCH

Description:

This block copies its input state (D) on the output (Q) when the enable input (C) is high and in this configuration it appears as transparent. The !Q output is the logical negation of Q
When the enable input goes low, the output keeps its previous state and acts like a memory.
This block can be typically used in I/O ports.

C	**D**	**Q_n**	**$!Q_n$**	
0	x	Q_{n-1}	$!Q_{n-1}$	No changes
1	0	0	1	Reset
1	1	1	0	Set

where x means "indifferent" and Qn-1 the previous state of Qn.

Datatype:

The block supports the following types :

- Inputs:

  D: scalar.Scilab’s int8 data type only.

  C:scalar.Scilab’s real double

  A positive input is considered as logical 1, a negative or a null input as logical 0.

- Outputs : scalar. Scilab's int8 data type.

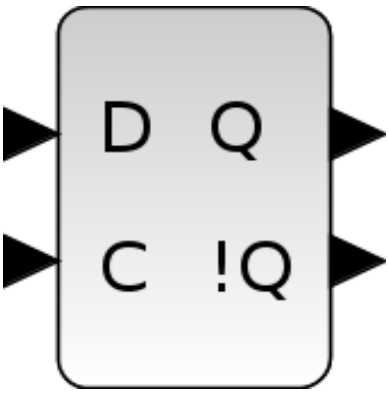
Parameters:

Nil

Name:Kanad Gaikwad
Date:28/05/19

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[1/1]/type5 port2:size[1/1]/type-1
regular outputs		port1:size[1/1]type5 port2:size[1/1]/type5
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		csuper



Name:Kanad Gaikwad
Date:28/05/19

EXTRACTBITS

Description:

For an integer input this block outputs a contiguous selection of bits. The operation can be summarized by this expression:

Output = Input ^{mask}
i.e. a bitwise AND between Input and Mask an integer of the same length as input with the bits to extract set to 1 and the other bits set to 0. Bits to Extract parameter defines the method by which the user selects the output bits as summarized by the following table with index 0, as that of the least significant bit :

Bits to Extract	**Output**	**Number of Bits or Index of Bit**
Upper Half	Half of the input that contain the most significant bit	Ignored
Lower Half	Half of the input that contain the least significant bit	Ignored
Range from MSB	**Number of Bits or Index of Bit** bits of the input that contain the most significant bit (MSB)	Number of bits to extract.
Range to LSB	**Number of Bits or Index of Bit** bits of the input that contain the least significant bit (LSB)	Number of bits to extract.
Range of bits	Range of bits of the input between the indexes of the bits `Start` and `End` of **Number of Bits or Index of Bit**parameter.	Vector with the format `[Start, End]` where `start`the first bit index and `end` the last bit index.

The output value depends also on the forth parameter Treat Bit Field as an Integer :

0 : the output is directly the result of extraction.

1 : the output is the integer conversion of the extraction result according to the signed or non-signed status of the input.

Datatypes:

The block supports the following types :

Input : scalar. All Scilab's integer type (Data type parameter).

Output : same type and dimensions than input.

Name:Kanad Gaikwad
Date:28/05/19

Parameters:

Name	Description	Value
Data Type	It indicates the integer type of input: between 3 & 8	3:int32,4:int16,5:int8
Bits to extract	It indicates the mode used to extract bits from the input data : Between 1 and 5	1:Upper Half, 2:Lower Half, 3:Range from MSB, 4:Range to LSB, 5:Range of bits
Number of Bits or Index of Bit	Dependent upon Bits to Extract	Dependent upon Bits to Extract
Treat Bit Field as an Integer	It indicates the scaling mode to use on the output bits selection	0:No 1:Yes

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[1/1]/type3
regular outputs		port1:size[3/3/type3
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		extract_bit_32_UHO

Name:Kanad Gaikwad
Date:28/05/19

INTMUL

Description:

That block computes the matrix multiplication of two integer input matrices. The number of rows of the second matrix must be equal to the number of columns of the first matrix.
The output is a matrix where the number of rows is equal to the number of rows of the first input matrix and the number of columns is equal to the number of columns of the second input matrix.
On overflow, the result can take different forms:
A normal non saturated result. By example, if type is int8 and the result is 128, the block output value will be -128.
A saturated result. For the previous example the block output value will be 127
An error message warning the user about the overflow.
The user can select one of these three forms by setting the Do on Overflow field to 0, 1 or 2.

Datatypes:

The block supports the following types :

nput : scalar,vectors or scalars. All Scilab's integer type (Data type parameter).

Output : same type and dimensions than input.

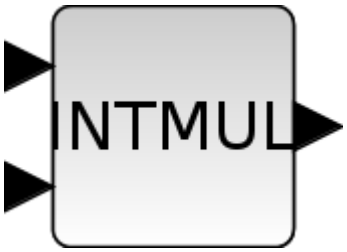
Parameters:

Name	Description	Value
Data Type	It indicates the integer type of input: between 3 & 8	3:int32,4:int16,5:int8
Do on Overflow	Between 0 and 2.	0:non saturated result 1:saturated solution 2:error message

Name:Kanad Gaikwad
Date:28/05/19

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port1:size[-1/-2]/type3
regular outputs		port1:size[-2/-3/type3
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		matmul_32



Name:Kanad Gaikwad
Date:28/05/19

JKFLIPFLOP

Description:

The JK flip flop is the most versatile of the basic flip-flops. It has two inputs traditionally labeled J (Set) and K (Reset).

- When the inputs J and K are different, the output Q takes the value of J at the next falling edge.
- When the inputs J and K are both low, no change occurs in the output state.
- When the inputs are both high the output Q will toggle from one state to other. It can perform the functions of the set/reset (SR) flip-flop and has the advantage that there are no ambiguous states.

The !Q output is the logical negation of Q
It can also act as a T flip-flop to accomplish toggling action if J and K are tied together. This toggle application finds extensive use in binary counters.
The user can set the initial output state with Initial Value parameter.

J	***K***	Q_n	$!Q_n$	
0	*0*	Q_{n-1}	$!Q_{n-1}$	Hold
0	*1*	*0*	*1*	Reset
1	*0*	*1*	*0*	Set
1	*1*	$!Q_{n-1}$	Q_{n-1}	Toggle

where Qn-1 is the previous state of Qn

Datatypes:

The block supports the following types :

- Inputs:

  J: scalar.Scilab’s int8 data type only.

  clk:scalar.Scilab’s real double

  K: scalar.Scilab’s int8 data type only.

  A positive input is considered as logical 1, a negative or a null input as logical 0.

- Outputs : scalar. Scilab's int8 data type.

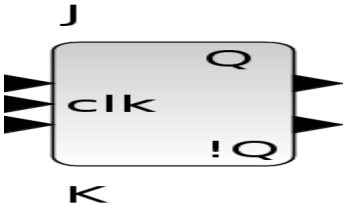
Name:Kanad Gaikwad
Date:28/05/19

Parameters:

Name	Description	Value
Initial Value	Initial state of Q output	Initial value must be 0 or 1 Negative values are int8(0) Positive values are intt8(1)

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port 1 : size [1,1] / type 5 port 2 : size [1,1] / type 1 port 3 : size [1,1] / type 5
regular outputs		port 1 : size [1,1] / type 5 port 2 : size [1,1] / type 1
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		csuper



Name:Kanad Gaikwad
Date:28/05/19

LOGIC

Description:

This block implements a standard truth table for modeling programming array, digital circuit and any other boolean expressions.
The user must specify a matrix that defines all the possible block outputs in the Truth Table field. In consequence, the number of rows must be a power of two. Each row of the matrix contains a logic combination of input elements.
Setting the parameter Truth Table defines the number of inputs and outputs in the following way :
The number of inputs is defined by the equation:

$$Number\_of\_lines = 2^{(Number\ of\ inputs)}$$

The number of outputs is equal to the number of columns of the matrix.

Datatypes:

The block supports the following types :

Input : scalar,vectors or scalars. All Scilab's integer type (Data type parameter).

Output : same type and dimensions than input.

Parameters:

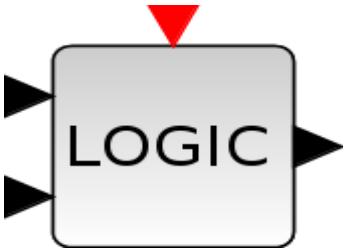
Name	Description	Value
Truth Table(matrix of outputs)	The matrix of outputs. The elements must be 0 or 1. For more information see the description part.	[0;0;0;1]
Accepts Inherited Event	Specifies if the clock is inherit or not.	0:no 1:Yes

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by	yes

Name:Kanad Gaikwad
Date:28/05/19

	input	
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port 1 : size [1,1] / type 5 port 2 : size [1,1] / type 5
regular outputs		port 1 : size [1,1] / type 5
number/size of activation input		1
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		logic



Name:Kanad Gaikwad
Date:28/05/19

SHIFT

Description:

This block shifts the bits of the input signal. In this operation the digits are moved to the right or to the left. The user can choose the rule to shifts the bits that can be normal or cycle by setting the **Shift Type**parameter to 0 or 1. The number and the direction of the shifts are set with the **Number of Bits to Shift Left**. If this number is positive the input is shifted to the left, otherwise it is shifted to the right.

When the **Shift Type** parameter is :

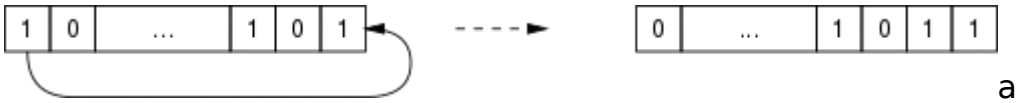
•**0** : an arithmetic shift is applied to the input signal. In this case, the bits that are shifted out of either end are discarded. Zeros are shifted in on the right, in the case of left shift; in the case of right shifts, copies of the sign bit is shifted in on the left.

By example, the one bit shift right gives:

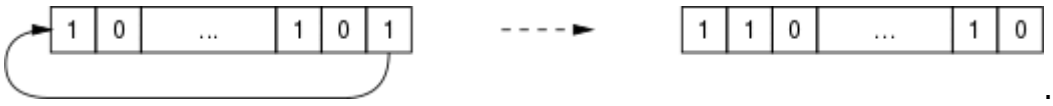
and the one bit shift left gives:

•**1** : a circular shift is applied to the input signal. In this case, the bits are rotated as if the left and right ends of the register are joined. The value that is shifted in on the right during a left-shift is whatever values was shifted out on the left, and vice versa.

By example, the one bit rotation right gives:



and the one bit rotation left gives:



.

The shift register makes a multiplication by 2^n(arithmetic left shift) or an integer division by 2^n(arithmetic right shift), where n is the number of bit shifts. It can also be used to serialize data or to create a memory buffer.

Name:Kanad Gaikwad
Date:28/05/19

Datatypes:

The block supports the following types :

Input : scalar,vectors or scalars. All Scilab's integer type (Data type parameter).

Output : same type and dimensions than input.

Parameters:

Name	Description	Value
Data Type	It indicates the integer type of input: between 3 & 8	3:int32,4:int16,5:int8
Number of bits to shift left (Negative number to shift right)	It indicates the number of bits the input signal is shifted/rotated. A positive value indicates a shift left, a negative value a shift right	int32 or uint32: positive and less than 32. int16 or uint16: positive and less than 16. int8 or uint8: positive and less than 8.
Shift Type	It indicates the rule used to shift the bits.	0:arithmetic 1:Circular

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port 1 : size [1,1] / type 5 port 2 : size [1,1] / type 5
regular outputs		port 1 : size [1,1] / type 5
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		logic

Name:Kanad Gaikwad
Date:28/05/19

SRFLIPFLOP

Description:

This block describes the simplest and the most fundamental latch the SR flip flop. The output **Q** depends of the state of the inputs **S** and **R**. The output **!Q** is the logical negation of **Q**

If S (Set) is pulsed high while R is held low, then the Q output is forced high, and stays high when S returns low.

•If **R** (Reset) is pulsed high while **S** is held low, then the **Q** output is forced low, and stays low when **R** returns low.

•When**S** and **R** are low,**Q(t)** takes the value of the previous output state **Q(t-1)**.

•When **S** and **R** are both high, both **Q** and **!Q**take the low or high values; the state is unstable. Practically this case is forbidden.

The user can set the initial output state with **Initial Value** parameter.

This block is almost used as a memory

The truth table of this block is:

S	**R**	**Q_n**	**$!Q_n$**	
0	0	Q_n	$!Q_n$	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	U	U	Forbidden state

Where U stands for "Unknown".

Datatypes:

The block supports the following types :

•Inputs:

•**R**: scalar. Scilab's int8 data type only.

•**S**: scalar. Scilab's int8 data type only.

A positive input is considered as logical 1, a negative or a null input as logical 0.

•Outputs: scalar. Scilab's int8 data type.

Name:Kanad Gaikwad
Date:28/05/19

Parameters:

Name	Description	Value
Initial Value	Initial value of state Q	Initial value must be 0 or 1 Negative values are int8(0) Positive values are intt8(1)

Properties:

Name	Description	Value
always active		no
direct-feedthrough	Output is controlled directly by input	yes
zero-crossing	Zero crossing detetction	no
mode		no
regular inputs		port 1 : size [1,1] / type 5 port 2 : size [1,1] / type 5
regular outputs		port 1 : size [1,1] / type 5
number/size of activation input		0
number/size of activation output		0
continous-time state		no
discrete-time state		no
object discrete time state		no
name of computational function		csuper