

# Generic Macro Library Reference Manual



March 2018

---

## Copyright

Copyright © 2018 Lattice Semiconductor Corporation. All rights reserved. This document may not, in whole or part, be reproduced, modified, distributed, or publicly displayed without prior written consent from Lattice Semiconductor Corporation (“Lattice”).

## Trademarks

Lattice Semiconductor Corporation, L Lattice Semiconductor Corporation (logo), L (stylized), L (design), Lattice (design), LSC, CleanClock, Custom Mobile Device, DiePlus, E<sup>2</sup>CMOS, ECP5, Extreme Performance, FlashBAK, FlexiClock, flexiFLASH, flexiMAC, flexiPCS, FreedomChip, GAL, GDX, Generic Array Logic, HDL Explorer, iCE Dice, iCE40, iCE65, iCEblink, iCEcable, iCEchip, iCEcube, iCEcube2, iCEman, iCEprog, iCESab, iCESocket, IPexpress, ISP, ispATE, ispClock, ispDOWNLOAD, ispGAL, ispGDS, ispGDX, ispGDX2, ispGDXV, ispGENERATOR, ispJTAG, ispLEVER, ispLeverCORE, ispLSI, ispMACH, ispPAC, ispTRACY, ispTURBO, ispVIRTUAL MACHINE, ispVM, ispXP, ispXPGA, ispXPLD, Lattice Diamond, LatticeCORE, LatticeEC, LatticeECP, LatticeECP-DSP, LatticeECP2, LatticeECP2M, LatticeECP3, LatticeECP4, LatticeMico, LatticeMico8, LatticeMico32, LatticeSC, LatticeSCM, LatticeXP, LatticeXP2, MACH, MachXO, MachXO2, MachXO3, MACO, mobileFPGA, ORCA, PAC, PAC-Designer, PAL, Performance Analyst, Platform Manager, ProcessorPM, PURESPEED, Reveal, SensorExtender, SiliconBlue, Silicon Forest, Speedlocked, Speed Locking, SuperBIG, SuperCOOL, SuperFAST, SuperWIDE, sysCLOCK, sysCONFIG, sysDSP, sysHSI, sysI/O, sysMEM, The Simple Machine for Complex Design, TracelD, TransFR, UltraMOS, and specific product designations are either registered trademarks or trademarks of Lattice Semiconductor Corporation or its subsidiaries in the United States and/or other countries. ISP, Bringing the Best Together, and More of the Best are service marks of Lattice Semiconductor Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

## Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL LATTICE SEMICONDUCTOR CORPORATION (LSC) OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF LSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

LSC may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. LSC makes no commitment to update this documentation. LSC reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. LSC recommends its customers obtain the latest version of the relevant information to establish, before ordering, that the information being relied upon is current.

---

## Type Conventions Used in This Document

Convention	Meaning or Use
<b>Bold</b>	Items in the user interface that you select or click. Text that you type into the user interface.
<i>&lt;Italic&gt;</i>	Variables in commands, code syntax, and path names.
<b>Ctrl+L</b>	Press the two keys at the same time.
<code>Courier</code>	Code examples. Messages, reports, and prompts from the software.
<code>...</code>	Omitted material in a line of code.
<code>.</code> <code>.</code> <code>.</code>	Omitted lines in code and report examples.
[ ]	Optional items in syntax descriptions. In bus specifications, the brackets are required.
( )	Grouped items in syntax descriptions.
{ }	Repeatable items in syntax descriptions.
	A choice between items in syntax descriptions.

---

# Contents

<b>Arithmetic Functions</b>	<b>1</b>
Adders	2
G_FADD	2
G_HADD	3
<b>I/O Pins</b>	<b>5</b>
Bidirectional Pin	6
G_BIDIR	6
Input Pin	7
G_INPUT	7
Output Pin	8
G_OUTPUT	8
Tri-state Output Pin	9
G_TRI	9
<b>Logic Gates</b>	<b>11</b>
AND Gates	12
2AND, 2AND1, 2AND2	12
3AND, 3AND1, 3AND2, 3AND3	13
4AND, 4AND1, 4AND2, 4AND3, 4AND4	14
OR Gates	15
2OR, 2OR1, 2OR2	15
3OR, 3OR1, 3OR2, 3OR3	16
4OR, 4OR1, 4OR2, 4OR3, 4OR4	17
NAND Gates	18
2NAND, 2NAND1, 2NAND2	18
3NAND, 3NAND1, 3NAND2, 3NAND3	19
4NAND, 4NAND1, 4NAND2, 4NAND3, 4NAND4	20
NOR Gates	21
2NOR, 2NOR1, 2NOR2	21

3NOR, 3NOR1, 3NOR2, 3NOR3 22  
4NOR, 4NOR1, 4NOR2, 4NOR3, 4NOR4 23  
BUF, INV, XOR, XNOR 24

**Multiplexers 25**

MUX21 26

MUX41 27

MUX81 28

**Registers 29**

D Flip-Flops 30

    G\_D, G\_DC, G\_DP 30

G\_DEC, G\_DEP 32

JK Flip-Flops 33

    G\_JK, G\_JKC, G\_JKP 33

Toggle Flip-flop 35

D Latches 37

    G\_LATCH 37

SR Latches 38

    G\_RS, G\_RSC, G\_RSP 38

# Arithmetic Functions

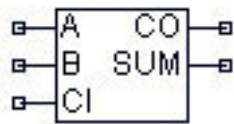
This chapter contains information on the following macros:

- ▶ Adders

# Adders

## G\_FADD

G\_FADD



### Function

G\_FADD: 1-bit full adder.

### Type

Soft

### Logic Resources

**Table 1:**

Macro	PT	Macrocell	Output	Level
G_FADD	*	2	2	1

\* Z0: 3 PTC0: 3PT

### Truth Table

**Table 2:**

Input			Output	
A	B	CI	CO	SUM
data	data	0	*	A+B
data	data	1	**	A+B+1

\* If  $A + B < 2^n$ , CO=0. If  $A + B \geq 2^n$ , CO=1.\*\* If  $A + B + 1 < 2^n$ , CO=0. If  $A + B + 1 \geq 2^n$ , CO=1.



## G\_HADD

G\_HADD



### Function

G\_HADD: 1-bit half adder.

### Type

Soft

### Logic Resources

**Table 3:**

Macro	PT	Macrocell	Output	Level
G_FADD	*	2	2	1

\* Z0: 2 PTC0: 1PT

### Truth Table

**Table 4:**

Input			Output	
A	B	CI	CO	SUM
data	data	1	*	A+B

\* If  $A + B < 2^n$ , CO=0. If  $A + B \geq 2^n$ , CO=1.



## I/O Pins

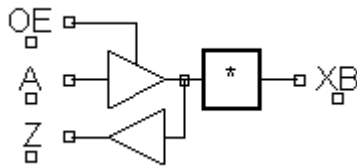
This chapter contains information on the following macros:

- ▶ Bidirectional Pin
- ▶ Input Pin
- ▶ Output Pins

# Bidirectional Pin

## G\_BIDIR

G\_BIDIR



### Function

G\_BIDIR: 1-bit bidirectional pin.

### Type

Hard

### Truth Table

Do not drive XB when OE=1.

**Table 5:**

Input			Output	
OE	A	B	XB	Z
0	x	d	-	d
0	x	Z	-	X
1	d	Z	d	d

D = any pattern of 1s and 0s on an input or set of inputs, x = don't care,

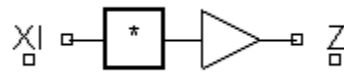
X = X (unknown) state, Z = high impedance state,

- = appears in output column if a bidirectional pin acts as an input pin.

# Input Pin

## G\_INPUT

G\_INPUT



### Function

G\_INPUT: 1-bit input pin.

### Type

Hard

### Truth Table

**Table 6:**

Input	Output
XI	Z
d	d

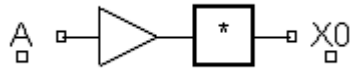
d = any pattern of 1s and 0s on an input or set of inputs,

XI = external input pin, Z = output.

# Output Pin

## G\_OUTPUT

G\_OUTPUT



### Function

G\_OUTPUT: 1-bit output pin.

### Type

Hard

### Truth Table

Table 7:

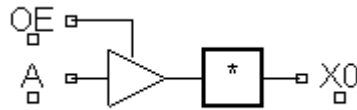
Input	Output
A	X0
d	d

d = any pattern of 1s and 0s on an input or set of inputs.

# Tri-state Output Pin

## G\_TRI

G\_TRI



### Function

G\_TRI: 1-bit tri-state output pin.

### Type

Hard

### Truth Table

**Table 8:**

Input		Output
OE	A	X0
0	x	Z
1	D	d

d = any pattern of 1s and 0s on an input or set of inputs,

x = don't care, Z = high impedance state.





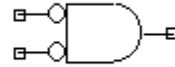
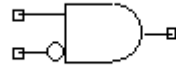
## Logic Gates

This chapter contains information on logic gate macros.

# AND Gates

## 2AND, 2AND1, 2AND2

G\_2ANDG\_2AND1G\_2AND2



### Function

G\_2AND, G\_2AND1, G\_2AND2: 2 input AND gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic primitive.

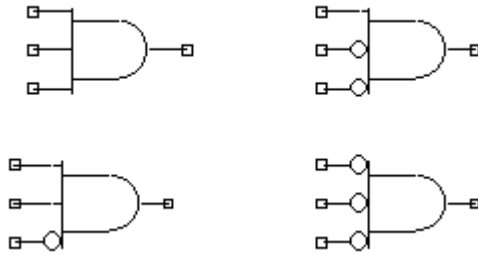
### Truth Table

**Table 9:**

Inputs		2AND	2AND1	2AND2
0	0	0	0	1
0	1	0	0	0
1	0	0	1	0
1	1	1	0	0

## 3AND, 3AND1, 3AND2, 3AND3

G\_3ANDG\_3AND2



G\_3AND1G\_3AND3

### Function

G\_3AND, G\_3AND1, G\_3AND2: 3 input AND gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic primitive.

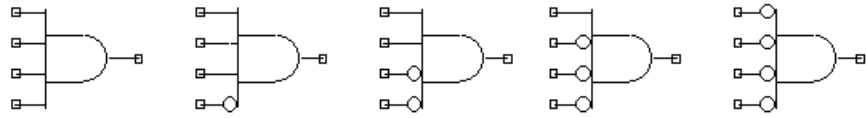
### Truth Table

Table 10:

Inputs			3AND	3AND1	3AND2	3AND3
0	0	0	0	0	0	1
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	1	0
1	0	1	0	0	0	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

## 4AND, 4AND1, 4AND2, 4AND3, 4AND4

G\_4ANDG\_4AND1G\_4AND2G\_4AND3G\_4AND4



### Function

G\_4AND, G\_4AND1, G\_4AND2, G\_4AND3, G\_4AND4: 4 input AND gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic primitive.

### Truth Table

Table 11:

Inputs				4AND	4AND1	4AND2	4AND3	4AND4
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0
1	1	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0
1	1	1	1	1	0	0	0	0

# OR Gates

## 2OR, 2OR1, 2OR2

G\_2ORG\_2OR1G\_2OR2



### Function

G\_2OR, G\_2OR1, G\_2OR2: 2 input OR gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic primitive.

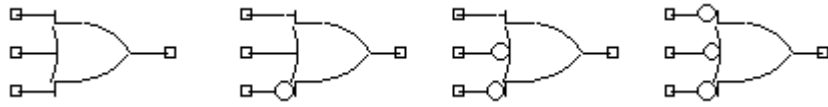
### Truth Table

**Table 12:**

Inputs		2OR	2OR1	2OR2
0	0	0	0	1
0	1	1	0	1
1	0	1	1	1
1	1	1	1	0

## 3OR, 3OR1, 3OR2, 3OR3

G\_3OR G\_3OR1 G\_3OR2 G\_3OR3



### Function

G\_3OR, G\_3OR1, G\_3OR2, G\_3OR3: 3 input OR gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic primitive.

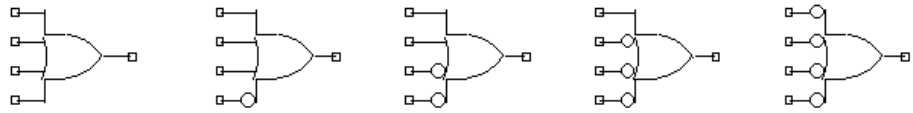
### Truth Table

Table 13:

Inputs			3OR	3OR1	3OR2	3OR3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	0	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	0

## 4OR, 4OR1, 4OR2, 4OR3, 4OR4

G\_4ORG\_4OR1G\_4OR2G\_4OR3G\_4OR4



### Function

G\_4OR, G\_4OR1, G\_4OR2, G\_4OR4: 4 input OR gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic primitive.

### Truth Table

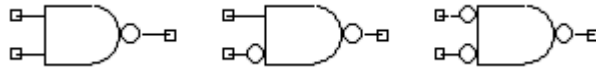
**Table 14:**

Inputs				4OR	4OR1	4OR2	4OR3	4OR4
0	0	0	0	0	1	1	1	1
0	0	0	1	1	0	1	1	1
0	0	1	0	1	1	1	1	1
0	0	1	1	1	1	0	1	1
0	1	0	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	1
1	0	0	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1
1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	0

# NAND Gates

## 2NAND, 2NAND1, 2NAND2

G\_2NAND G\_2NAND1 G\_2NAND2



### Function

G\_2NAND, G\_2NAND1, G\_2NAND2: 2 input NAND gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic Primitive.

### Truth Table

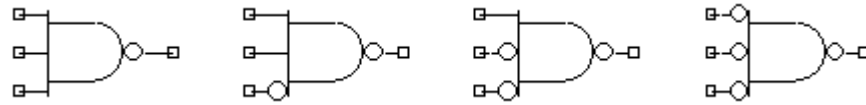
Table 15:

Inputs		2NAND	2NAND1	2NAND2
0	0	1	1	0
0	1	1	1	1
1	0	1	0	1
1	1	0	1	1



## 3NAND, 3NAND1, 3NAND2, 3NAND3

G\_3NANDG\_3NAND1G\_3NAND2G\_3NAND3



### Function

G\_3NAND, G\_3NAND1, G\_3NAND2, G\_3NAND3: 3 input NAND gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

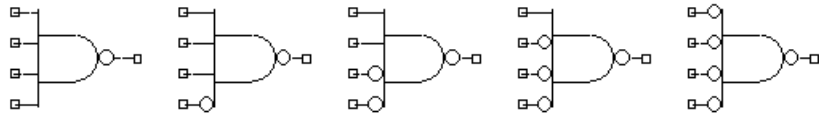
Logic Primitive.

### Truth Table

Inputs			3NAND	3NAND1	3NAND2	3NAND3
0	0	0	1	1	1	0
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	1	0	1
1	0	1	1	1	1	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1

## 4NAND, 4NAND1, 4NAND2, 4NAND3, 4NAND4

G\_4NANDG\_4NAND1G\_4NAND2G\_4NAND3G\_4NAND4



### Function

G\_4NAND, G\_4NAND1, G\_4NAND2, G\_4NAND3, G\_4NAND4: 4 input NAND gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

Logic Primitive.

### Truth Table

Inputs				4NAND	4NAND1	4NAND2	4NAND3	4NAND4
0	0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0	1
1	0	0	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	1	1	0	1	1
1	1	0	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1
1	1	1	1	0	1	1	1	1

## NOR Gates

### 2NOR, 2NOR1, 2NOR2

G\_2NORG\_2NOR1G\_2NOR2



#### Function

G\_2NOR, G\_2NOR1, G\_2NOR2: 2 input NOR gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

#### Type

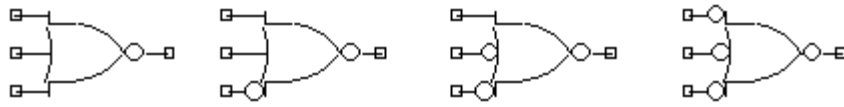
Logic Primitive.

#### Truth Table

Inputs		2NOR	2NOR1	2NOR2
0	0	1	0	0
0	1	0	1	0
1	0	0	0	0
1	1	0	0	1

## 3NOR, 3NOR1, 3NOR2, 3NOR3

G\_3NOR G\_3NOR1 G\_3NOR2 G\_3NOR3



### Function

G\_3NOR, G\_3NOR1, G\_3NOR2, G\_3NOR3: 3 input NOR gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

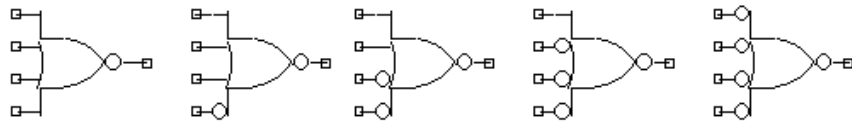
Logic Primitive.

### Truth Table

Inputs			3NOR	3NOR1	3NOR2	3NOR3
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

## 4NOR, 4NOR1, 4NOR2, 4NOR3, 4NOR4

G\_4NOR G\_4NOR1 G\_4NOR2 G\_4NOR3 G\_4NOR4



### Function

G\_4NOR, G\_4NOR1, G\_4NOR2, G\_4NOR3, G\_4NOR4: 4 input NOR gates.

Note: Bubble(s) on the input(s) mean inverted input(s).

### Type

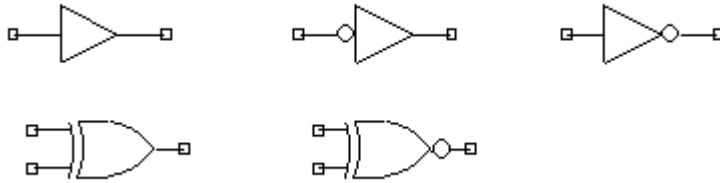
Logic Primitive.

### Truth Table

Inputs				4NOR	4NOR1	4NOR2	4NOR3	4NOR4
0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1

## BUF, INV, XOR, XNOR

G\_BUF, G\_BUFB, G\_INV



G\_XOR, G\_XNOR

### Function

G\_BUF: single input buffer.

G\_BUFB: single inverted input buffer.

G\_INV: single input inverter.

G\_XOR: 2 input exclusive OR gate.

G\_XNOR: 2 input exclusive NOR gate.

### Type

Logic Primitive.

### Truth Table

Input	G_BUF	G_BUFB	G_INV
0	0	1	1
1	1	0	0

Inputs		XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

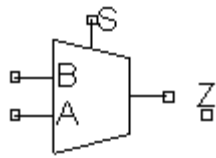
# Multiplexers

This chapter contains information on the following macros:

- ▶ 2-to-1 multiplexer
- ▶ 4-to-1 multiplexer
- ▶ 8-to-1 multiplexer

## MUX21

G\_MUX21



### Function

G\_MUX21: 2 input multiplexer with 1 select line.

### Type

Soft

### Logic Resources

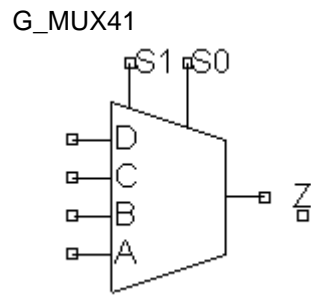
Macro	PT	Macrocell	Output	Level
G_MUX21	2	1	1	1

### Truth Table

Input	Output
S	Z
0	A
1	B



# MUX41



## Function

G\_MUX41: 4 input multiplexer with 2 select line.

## Type

Soft

## Logic Resources

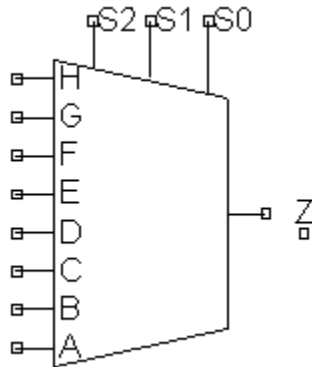
Macro	PT	Macrocell	Output	Level
G_MUX41	4	1	1	1

## Truth Table

Input		Output
S1	S0	Z
0	0	A
0	1	B
1	0	C
1	1	D

# MUX81

G\_MUX81



### Function

G\_MUX81: 8 input multiplexer with 3 select line.

### Type

Soft

### Logic Resources

Macro	PT	Macrocell	Output	Level
G_MUX81	8	2	1	1

### Truth Table

Input			Output
S2	S1	S0	Z
0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E
1	0	1	F
1	1	0	G
1	1	1	H

# Registers

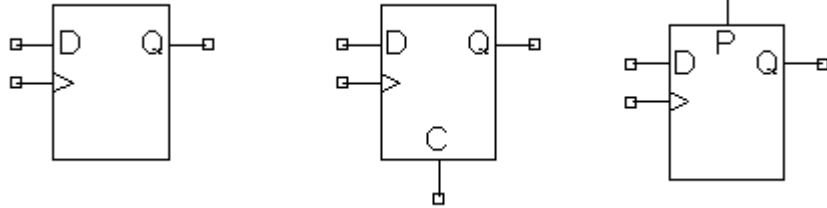
This chapter contains information on the following macros:

- ▶ D Flip-Flops
- ▶ JK Flip-Flops
- ▶ Toggle Flip-Flops
- ▶ D Latches
- ▶ SR Latches

## D Flip-Flops

### G\_D, G\_DC, G\_DP

G\_DG\_DCG\_DP



#### Function

G\_D: 1-bit D flip-flop.

G\_DC: 1-bit D flip-flop with synchronous clear.

G\_DP: 1-bit D flip-flop with synchronous preset.

#### Type

Logic Primitive.

#### Logic Resources

Macro	PT	Macrocell	Output	Level
G_D	1*	1	1	1
G_DC	1*	1	1	1
G_DP	2*	1	1	1

\* Add 1 PT per GLB if Product Term Clock is used.

### Truth Table

Input				Output (Q0~Q <sub>n-1</sub> )		
D0~D <sub>n-1</sub>	CLK	C	P	G_D	G_DC	G_DP
x	↑	1	1	-	0	1
d	↑	0	0	d	d	d
x	0	0	x	Q0'~Qn'	Q0'~Qn'	Q0'~Qn'
x	1	0	x	Q0'~Qn'	Q0'~Qn'	Q0'~Qn'

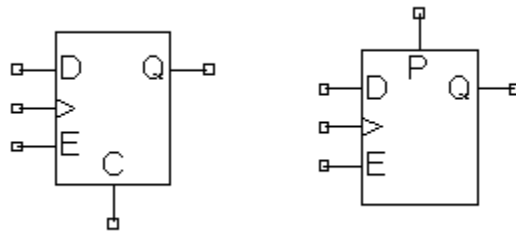
d = any pattern of 1s and 0s in an input or set of inputs,

Q0'~Qn' = previous output of flip-flop or latch, - = not apply,

x = don't care, ↑ = rising clock edge.

## G\_DEC, G\_DEP

### G\_DECG\_DEP



### Function

G\_DEC: 1-bit D flip-flop with synchronous clear and enable.

G\_DEP: 1-bit D flip-flop with synchronous preset and enable.

### Type

Soft

### Logic Resources

Macro	PT	Macrocell	Output	Level
G_DEC	1*	1	1	1
G_DEP	2*	1	1	1

\* Add 1 PT per GLB if Product Term Clock is used.

### Truth Table

Input				Output (Q0~Q <sub>n-1</sub> )	
D0~D <sub>n-1</sub>	CLK	C	P	G_DEC	G_DEP
x	↑	1	1	0	1
d	↑	0	0	d	d
x	0	0	x	Q0'~Qn'	Q0'~Qn'
x	1	0	x	Q0'~Qn'	Q0'~Qn'

d = any pattern of 1s and 0s in an input or set of inputs,

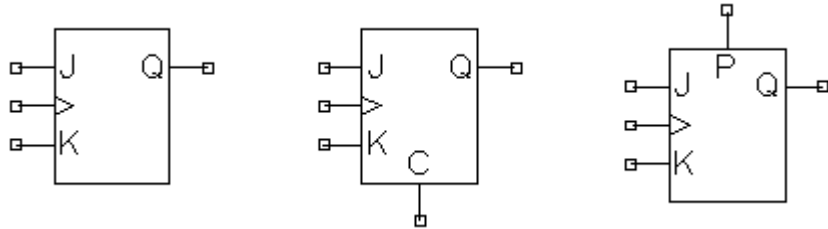
Q0'~Qn' = previous output of flip-flop or latch,

x = don't care, ↑ = rising clock edge.

# JK Flip-Flops

## G\_JK, G\_JKC, G\_JKP

G\_JKG\_JKCG\_JKP



### Function

G\_JK: JK flip-flop.

G\_JKC: JK flip-flop with synchronous clear.

G\_JKP: JK flip-flop with synchronous preset.

### Type

Soft

### Logic Resources

Macro	PT	Macrocell	Output	Level
G_JK	2*	1	1	1
G_JKC	2*	1	1	1
G_JKP	2*	1	1	1

\* Add 1 PT per GLB if Product Term Clock is used.

**Truth Table**

Input					Output
P	C	J	K	CLK	Q
x	1	x	x	x	0
0	0	0	0	↑	Q0'
0	0	0	1	↑	0
0	0	1	0	↑	1
0	0	1	1	↑	$\overline{Q0'}$
x	0	x	x	0	Q0'
x	0	x	x	1	Q0'
1	0	x	x	↑	1

Q0' = previous output of flip flop,  $\overline{Q0'}$  = inverse of Q0',

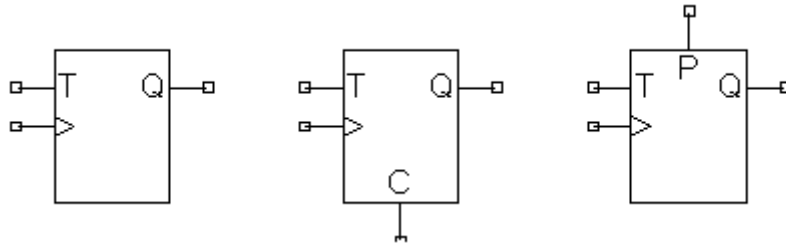
X=don't care, ↑ = rising clock edge



# Toggle Flip-flop

## G\_T, G\_TC, G\_TP

G\_TG\_TCG\_TP



### Function

G\_T: Toggle flip-flop.

G\_TC: Toggle flip-flop with synchronous clear.

G\_TP: Toggle flip-flop with synchronous preset.

### Type

Soft

### Logic Resources

Macro	PT	Macrocell	Output	Level
G_T	3*	1	1	1
G_TC	3*	1	1	1
G_TP	3*	1	1	1

\* Add 1 PT per GLB if Product Term Clock is used.

**Truth Table**

Input				Output
P	C	T	CLK	Q
1	x	x	↑	1
0	1	x	↑	0
0	0	0	↑	Q0'
0	0	1	↑	$\overline{Q0'}$
x	x	x	0	Q0'
x	x	x	1	Q0'

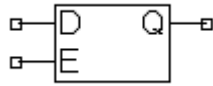
Q0' = previous output of flip flop,  $\overline{Q0'}$ =inverse of Q0',

X=don't care, ↑ = rising clock edge

## D Latches

### G\_LATCH

G\_LATCH



#### Function

G\_LATCH: 1-bit D latch with enable.

#### Type

Hard

#### Logic Resources

Macro	PT	Macrocell	Output	Level
G_LATCH	3	1	1	1

#### Truth Table

Input		Output
D0~D <sub>n-1</sub>	E	Q0~Q <sub>n-1</sub>
d	1	d
x	0	Q0'~Qn'

d = any pattern of 1s and 0s in an input or set of inputs,

Q0'~Qn' = previous output of flip-flop or latch,

x = don't care.

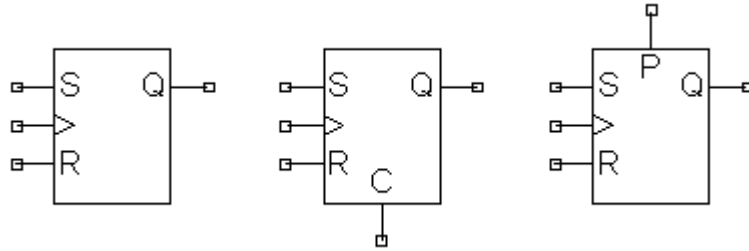
#### NOTE:

The Truth Table was changed in March, 2018. It previously had 1 in the x row. The 1 has been changed to 0.

## SR Latches

### G\_RS, G\_RSC, G\_RSP

G\_RSG\_RSCG\_RSP



#### Function

G\_RS: simple SR latch.

G\_RSC: simple SR latch with synchronous clear.

G\_RSP: simple SR latch with synchronous preset.

#### Type

Soft

## Logic Resources

Macro	PT	Macrocell	Output	Level
G_RS	2	1	1	1
G_RSC	2	1	1	1
G_RSP	2	1	1	1

## Truth Table

Input				Output
P	C	S	R	Q
0	0	0	0	Q
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1*
1	0	x	x	1
0	1	x	x	0

\* These outputs are not entirely stable. They may not remain when both S and R return to 1.

