



***Generic Macro Library***  
**Reference Manual**

---

# Table of Contents

---

<b>Arithmetic Functions</b> .....	<b>3</b>
Adders .....	4
G_FADD .....	4
G_HADD .....	5
<b>I/O Pins</b> .....	<b>6</b>
Bidirectional Pin – G_BIDIR .....	7
Input Pin – G_INPUT .....	8
Output Pin – G_OUTPUT .....	9
G_TRI .....	10
<b>Logic Gates</b> .....	<b>11</b>
AND gates .....	12
2 input AND gates .....	12
3 input AND gates .....	13
4 input AND gates .....	14
OR gates .....	15
2 input OR gates .....	15
3 input OR gates .....	16
4 input OR gates .....	17
NAND gates .....	18
2 input NAND gates .....	18
3 input NAND gates .....	19
4 input NAND gates .....	20
NOR gates .....	21
2 input NOR gates .....	21
3 input NOR gates .....	22
4 input NOR gates .....	23
BUF, INV, XOR, XNOR .....	24

<b>Multiplexers</b> .....	<b>25</b>
G_MUX21 .....	26
G_MUX41 .....	27
G_MUX81 .....	28
<b>Registers</b> .....	<b>29</b>
D Flip-flops .....	30
G_D, G_DC, G_DP .....	30
G_DEC, G_DEP .....	31
JK Flip-flops .....	32
G_JK, G_JKC, G_JKP .....	32
Toggle Flip-flops .....	33
G_T, G_TC, G_TP .....	33
D Latch .....	34
G_LATCH .....	34
SR Latches .....	35
G_RS, G_RSC, G_RSP .....	35

# Arithmetic Functions

---

This chapter contains information on the following macros:

- Adders

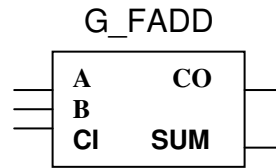
## Adders

### G\_FADD

**Function:**

G\_FADD : 1-bit full adder.

**Type:** Soft.



**Logic Resources:**

Macro	PT	Macrocell	Output	Level
G_FADD	*	2	2	1

\* Z0 : 3 PT      C0 : 3 PT

**Truth Table:**

Input			Output	
A0	B0	CI	C0	SUM
data	data	0	*	A + B
data	data	1	**	A + B + 1

\* If  $A+B < 2^n$ , C0 = 0. If  $A+B \geq 2^n$ , C0 = 1.

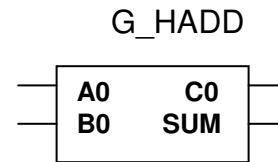
\*\* If  $A+B+1 < 2^n$ , C0 = 0. If  $A+B+1 \geq 2^n$ , C0 = 1.

## G\_HADD

### Function:

G\_HADD : 1-bit half adder.

Schematic appear on the following page.



### Type:

Soft : G\_HADD.

Logic Resources:

Macro	PT	Macrocell	Output	Level
G_FADD	*	2	2	1

\* Z0 : 2 PT      C0 : 1 PT

### Truth Table:

Input			Output	
A0	B0	CI	C0	SUM
data	data	0	*	A + B

\* If  $A+B < 2^n$ ,  $C0 = 0$ . If  $A+B \geq 2^n$ ,  $C0 = 1$ .

# I/O Pins

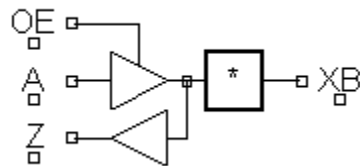
---

This chapter contains information on the following macros:

- Bidirectional Pin
- Input Pin
- Output Pins

## Bidirectional Pin

G\_BIDIR



### Function:

G\_BIDIR : 1-bit bidirectional pin.

**Type:** Hard

### Truth Table:

Do not drive XB when OE =1.

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

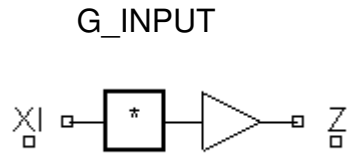
D = any pattern of 1s and 0s on an input or set of inputs, x = don't care,

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

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



## Input Pin



### Function:

G\_INPUT : 1-bit input pin.

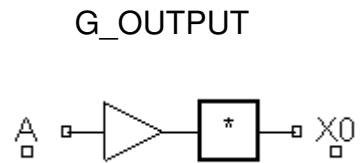
**Type:** Hard

### Truth Table:

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 Pins



**Function:**

G\_OUTPUT : 1-bit output pin.

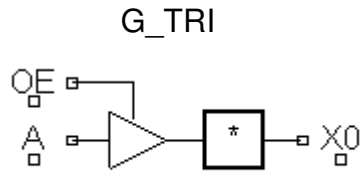
**Type:** Hard

**Truth Table:**

Input	Output
<b>A</b>	<b>X0</b>
d	d

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

## Tri-state Output Pin



**Function:**

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

**Type:** Hard

**Truth Table:**

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.

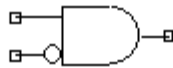
## AND gates

2AND, 2AND1, 2AND2

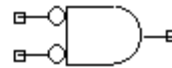
G\_2AND



G\_2AND1



G\_2AND2



### Function:

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

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

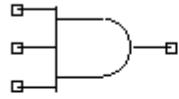
**Type:** Logic Primitive.

### Truth Table:

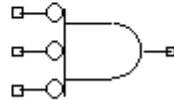
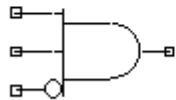
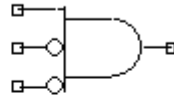
Inputs		2AND	2AND1	2AND2
0	0	0	0	1
0	1	0	0	0
1	0	0	1	0
1	1	1	0	0

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

G\_3AND



G\_3AND2



G\_3AND1

G\_3AND3

### Function:

G\_3AND, G\_3AND1, G\_3AND2, G\_3AND3: 3 input AND gates.

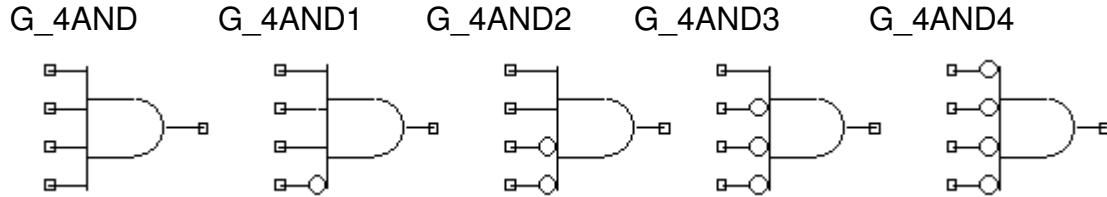
Note: Bubble(s) on the input(s) mean inverted input(s).

**Type:** Logic Primitive.

### Truth Table:

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



### 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).

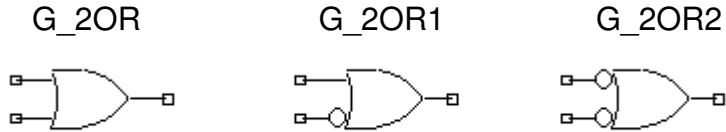
**Type:** Logic Primitive.

### Truth Table:

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



#### Function:

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

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

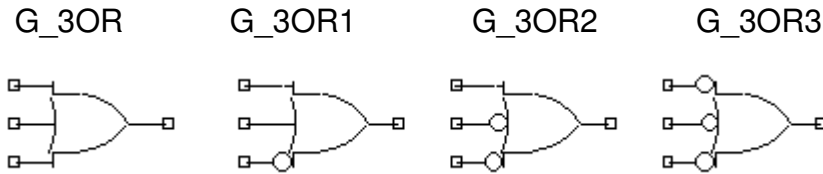
**Type:** Logic Primitive

#### Truth Table:

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



### Function:

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

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

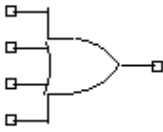
**Type:** Logic Primitive

### Truth Table:

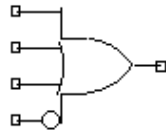
Inputs			3OR	3OR1	3OR2	3OR3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	0	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	0

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

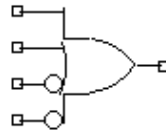
G\_4OR



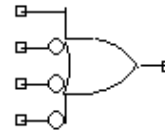
G\_4OR1



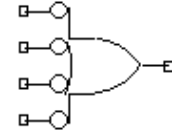
G\_4OR2



G\_4OR3



G\_4OR4



### Function:

G\_4OR, G\_4OR1, G\_4OR2, G\_4OR3, G\_4OR4: 4 input OR gates.

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

**Type:** Logic Primitive

### Truth Table:

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).

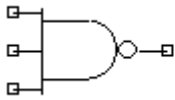
**Type:** Logic Primitive

#### Truth Table:

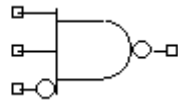
Inputs		2NAND	2NAND1	2NAND2
0	0	1	1	0
0	1	1	1	1
1	0	1	0	1
1	1	0	1	1

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

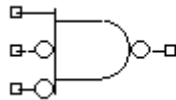
G\_3NAND



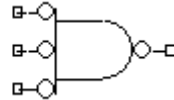
G\_3NAND1



G\_3NAND2



G\_3NAND3



### Function:

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

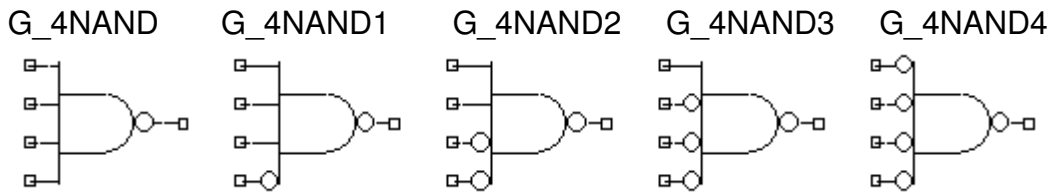
Note: Bubble(s) on the input(s) mean inverted input(s).

**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



### 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

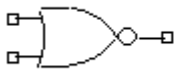
### Truth Table:

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\_2NOR



G\_2NOR1



G\_2NOR2



#### Function:

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

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

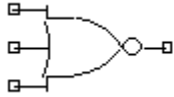
**Type:** Logic Primitive

#### Truth Table:

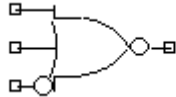
Inputs		2NOR	2NOR1	2NOR2
0	0	1	0	0
0	1	0	1	0
1	0	0	0	0
1	1	0	0	1

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

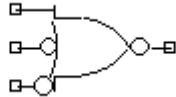
G\_3NOR



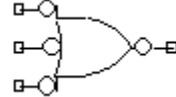
G\_3NOR1



G\_3NOR2



G\_3NOR3



### Function:

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

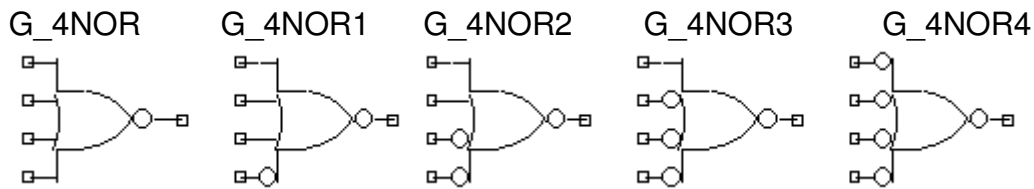
Note: Bubble(s) on the input(s) mean inverted input(s).

**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



### 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).

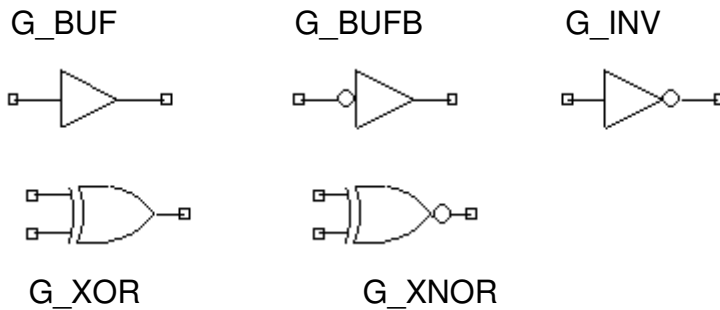
**Type:** Logic Primitive

### 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



### 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

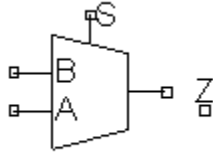
---

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:

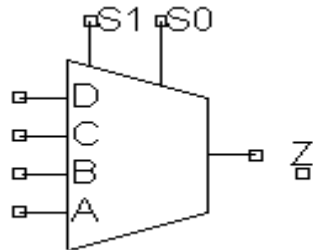
Macro	PT	Macrocell	Output	Level
G_MUX21	2	1	1	1

### Truth Table:

Input	Output
S	Z
0	B
1	A

## MUX41

G\_MUX41



### Function:

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

**Type:** Soft

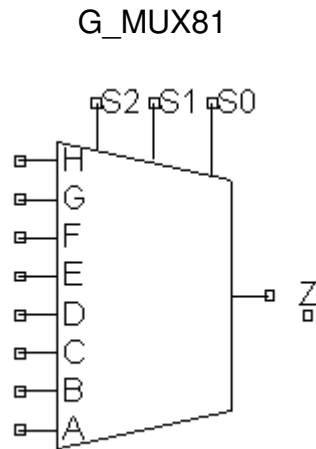
### Logic Resources:

Macro	PT	Macrocell	Output	Level
G_MUX41	4	1	1	1

### Truth Table:

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

# 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	H
0	0	1	G
0	1	0	F
0	1	1	E
1	0	0	D
1	0	1	C
1	1	0	B
1	1	1	A

# Registers

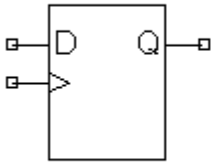
---

This chapter contains information on the following macros:

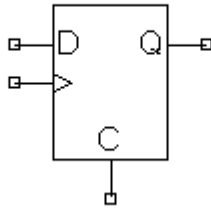
- D Flip-Flops
- JK Flip-Flops
- Toggle Flip-Flops
- D Latches
- SR Latches

## D Flip-Flops

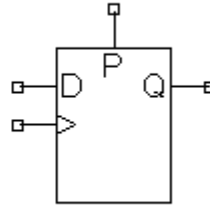
### G\_D, G\_DC, G\_DP



G\_D



G\_DC



G\_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

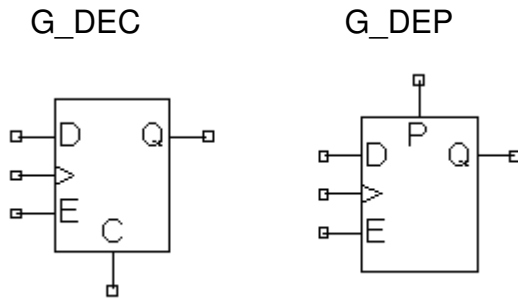
\* Add 1 PT per GLB if Product Term Clock is used.

#### 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



### 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:

D0~D <sub>n-1</sub>	Input			Output (Q0~Q <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,

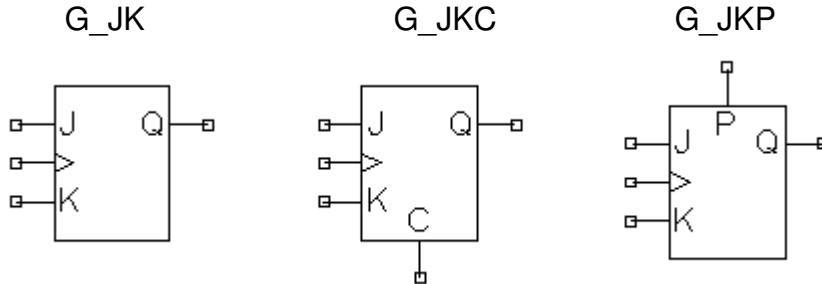
Q0'~Qn' = previous output of flip-flop or latch,

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



## JK Flip-Flops

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



#### 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:

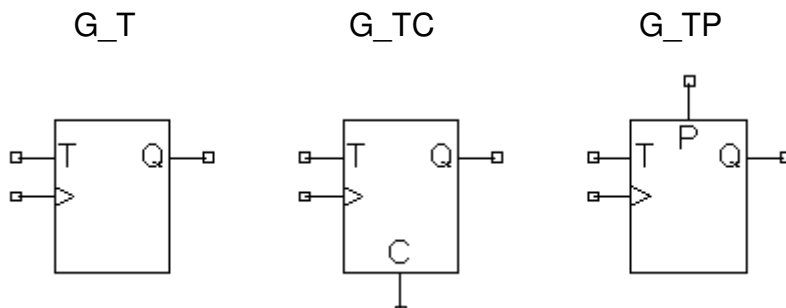
Input					Output
P	C	J	K	CLK	Q
x	1	x	x	x	0
0	0	0	0	↑	Q0'
0	0	0	1	↑	0
0	0	1	0	↑	1
0	0	1	1	↑	$\overline{Q0'}$
x	0	x	x	0	Q0'
x	0	x	x	1	Q0'
1	0	x	x	↑	1

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**



### **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:**

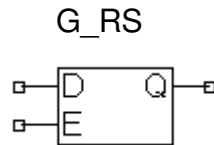
Input				Output
P	C	T	CLK	Q
1	x	x	↑	1
0	1	x	↑	0
0	0	0	↑	Q0'
0	0	1	↑	$\overline{Q0'}$
x	x	x	0	Q0'
x	x	x	1	Q0'

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

X=don't care, ↑ = rising clock edge

## D 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

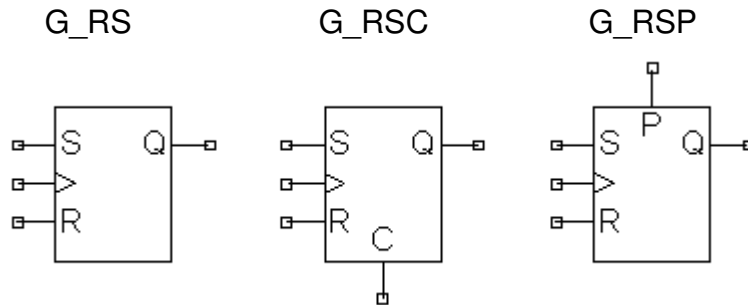
#### Truth Table:

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

## SR Latches

### G\_RS, G\_RSC, G\_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.