Macro Library User Guide for RTG4 FPGA

Libero SoC v2025.1

Introduction (Ask a Question)

This macro library guide supports the RTG4™ FPGA family. See the Microchip website for macro guides for other families. This guide follows a naming convention for sequential macros that is unambiguous and extensible, making it possible to understand the function of the macros by their name alone.

The first two mandatory characters of the macro name indicates the basic macro function:

- DF—D-type flip-flop

The following mandatory character indicates the output polarity:

- I—output inverted (QN with bubble)
- N—output non-inverted (Q without bubble)

The following mandatory number indicates the polarity of the clock or gate:

- 1—rising edge-triggered flip-flop or transparent high latch (non-bubbled)
- 0—falling edge-triggered flip-flop or transparent low latch (bubbled)

The following two optional characters indicate the polarity of the Enable pin, if present:

- E0—active-low enable (bubbled)
- E1—active-high enable (non-bubbled)

The following two optional characters indicate the polarity of the asynchronous Preset pin, if present:

- P0—active-low asynchronous preset (bubbled)
- P1—active-high asynchronous preset (non-bubbled)

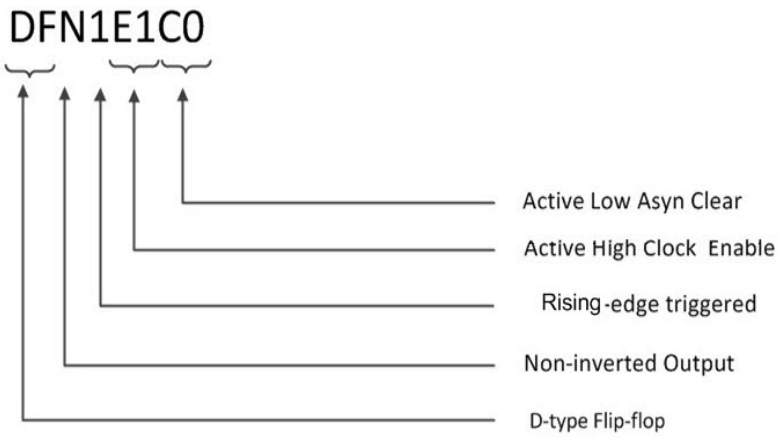
The following two optional characters indicate the polarity of the asynchronous Clear pin, if present:

- C0—active-low asynchronous clear (bubbled)
- C1—active-high asynchronous clear (non-bubbled)

All sequential and combinatorial macros (except MX4 and XOR8) use one logic element in the RTG4 family.

For example, the macro DFN1E1C0 indicates a D-type flip-flop (DF) with a non-inverted (N) Q output, positive-edge-triggered (1), with Active-High Clock Enable (E1) and Active-Low Asychronous Clear (C0). See the following figure.

Figure 1. Naming Convention



Truth Table Notation (Ask a Question)

The truth table states in this User Guide are defined as follows.

Table 1. Truth Table

State	Meaning
0	Logic “0”
1	Logic “1”
X	Don't Care (for Inputs), Unknown (for Outputs)
Z	High Impedance

User Parameter/Generics (Ask a Question)

WARNING_MSGS_ON

This feature enables you to disable the warning messages display. Default is ON ('True' in VHDL and '1' in Verilog).

Table of Contents

1. All Macros (Ask a Question)

1.1. AND2 (Ask a Question)

2-Input AND.

Figure 1-1. AND2

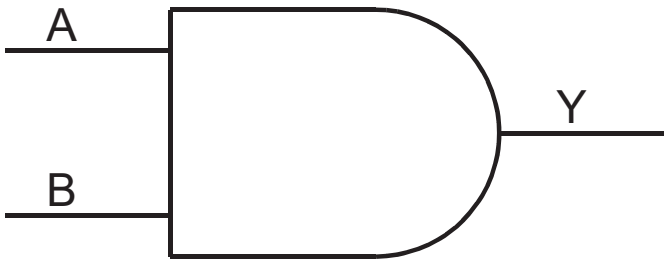


Table 1-1. AND2 I/O

Inputs	Output
A, B	Y

Table 1-2. AND2 Truth Table

A	B	Y
X	0	0
0	X	0
1	1	1

1.2. AND3 (Ask a Question)

3-Input AND.

Figure 1-2. AND3

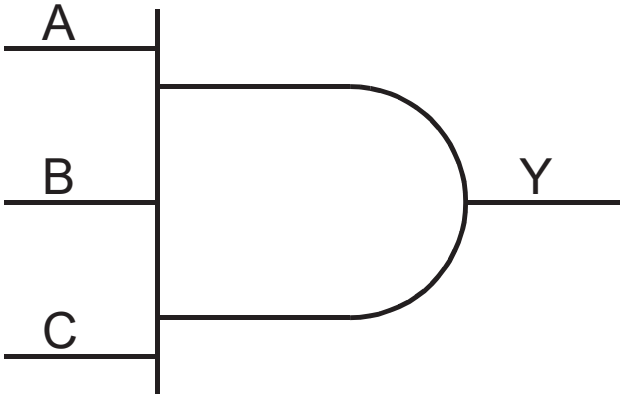


Table 1-3. AND3 I/O

Input	Output
A, B, C	Y

Table 1-4. AND3 Truth Table

A	B	C	Y
X	X	0	0
X	0	X	0
0	X	X	0
1	1	1	1

1.3. AND4 (Ask a Question)

4-Input AND.

Figure 1-3. AND4

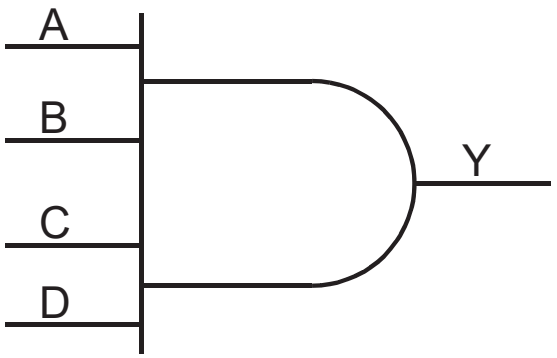


Table 1-5. AND4 I/O

Input	Output
A, B, C, D	Y

Table 1-6. AND4 Truth Table

A	B	C	D	Y
X	X	X	0	0
X	X	0	X	0
X	0	X	X	0
0	X	X	X	0
1	1	1	1	1

1.4. CFG1/2/3/4 and Look-Up Tables (LUTs) (Ask a Question)

CFG1, CFG2, CFG3, and CFG4 are post-layout LUTs that implement any 1-input, 2-input, 3-input, and 4-input combinational logic functions respectively. Each of the CFG1/2/3/4 macros has an INIT string parameter that determines the logic functions of the macro. The output Y is dependent on the INIT string parameter and the values of the inputs.

1.5. CFG2 (Ask a Question)

Post-layout macro implements any 2-input combinational logic function. Output Y is dependent on the INIT string parameter and the value of A and B. The INIT string parameter is 4 bits wide.

Figure 1-4. CFG2

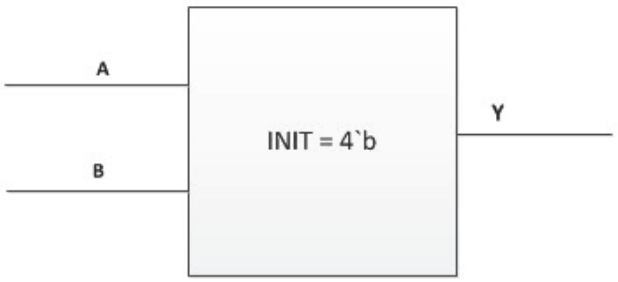


Table 1-7. CFG2 I/O

Input	Output
A,B	Y = f (INIT, A, B)

Table 1-8. CFG2 INIT String Interpretation

BA	Y
00	INIT[0]
01	INIT[1]
10	INIT[2]
11	INIT[3]

1.6. CFG3 (Ask a Question)

Post-layout macro used to implement any 3-input combinational logic function. Output Y is dependent on the INIT string parameter and the value of A, B, and C. The INIT string parameter is 8 bits wide.

Figure 1-5. CFG3

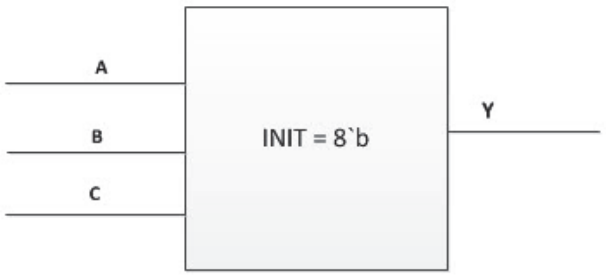


Table 1-9. CFG3 I/O

Input	Output
A, B, C	Y = f (INIT, A,B, C)

Table 1-10. CFG3 INIT String Interpretation

CBA	Y
000	INIT[0]

Table 1-10. CFG3 INIT String Interpretation (continued)

CBA	Y
001	INIT[1]
010	INIT[2]
011	INIT[3]
100	INIT[4]
101	INIT[5]
110	INIT[6]
111	INIT[7]

1.7. CFG4 (Ask a Question)

Post-layout macro used to implement any 4-input combinational logic function. Output Y is dependent on the INIT string parameter and the value of A, B, C, and D. The INIT string parameter is 16 bits wide.

Figure 1-6. CFG4

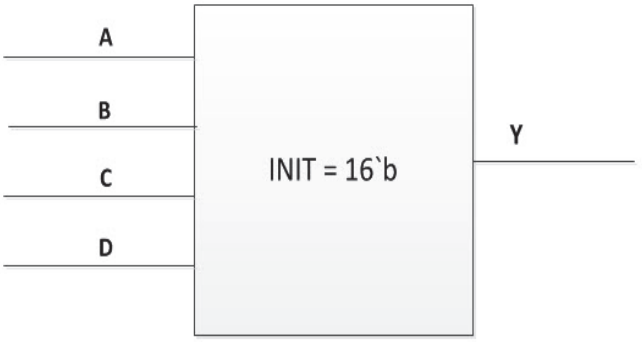


Table 1-11. CFG4 I/O

Input	Output
A, B, C, D	Y = f (INIT, A,B, C, D)

Table 1-12. CFG4 Truth Table

DCBA	Y
0000	INIT[0]
0001	INIT[1]
0010	INIT[2]
0011	INIT[3]
0100	INIT[4]
0101	INIT[5]
0110	INIT[6]
0111	INIT[7]
1000	INIT[8]
1001	INIT[9]
1010	INIT[10]
1011	INIT[11]

Table 1-12. CFG4 Truth Table (continued)

DCBA	Y
1100	INIT[12]
1101	INIT[13]
1110	INIT[14]
1111	INIT[15]

1.8. BUFF (Ask a Question)

Buffer.

Figure 1-7. BUFF

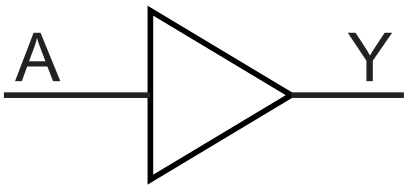


Table 1-13. BUFF

Input	Output
A	Y

Table 1-14. Truth Table

A	Y
0	0
1	1

1.9. BUFD (Ask a Question)

Buffer.
Note: The compile optimization does not remove this macro.

Figure 1-8. BUFD

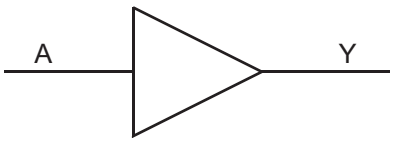


Table 1-15. BUFD I/O

Input	Output
A	Y

Table 1-16. BUFD Truth Table

A	Y
0	0
1	1

1.10. BUFD_DELAY (Ask a Question)

Buffer. The cell delay of BUFD_DELAY is about 0.4 ns at Military operating conditions. Its delay is longer than that of BUFD.
Note: Compile optimization will not remove this macro.

Figure 1-9. BUFD_DELAY

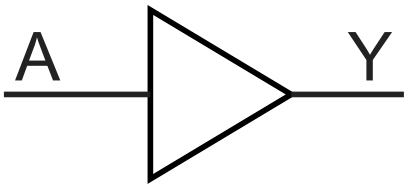


Table 1-17. BUFD_DELAY

Input	Output
A	Y

Table 1-18. Truth Table

A	Y
0	0
1	1

1.11. CLKINT (Ask a Question)

This macro routes an internal fabric signal to the global network.

Figure 1-10. CLKINT

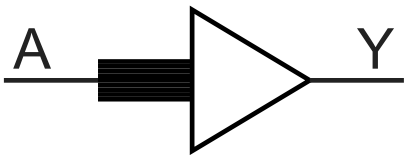


Table 1-19. CLKINT I/O

Input	Output
A	Y

Table 1-20. CLKINT Truth Table

A	Y
0	0
1	1

1.12. GBR (Ask a Question)

Back-annotated macro that routes an internal fabric signal to global network.

Figure 1-11. GBR

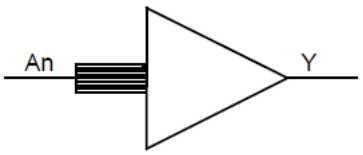


Table 1-21. GBR

Input	Output
An	Y

Table 1-22. Truth Table

An	Y
0	0
1	1

1.13. CLKINT_PRESERVE (Ask a Question)

This Macro routes an internal fabric signal to the global network. It has the same functionality as CLKINT except that this clock always stays on the global clock network and will not be demoted during design implementation.

Figure 1-12. CLKINT_PRESERVE

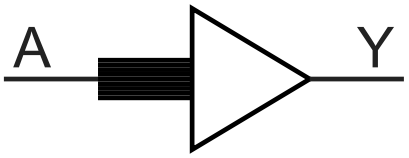


Table 1-23. CLKINT_PRESERVE I/O

Input	Output
A	Y

Table 1-24. CLKINT_PRESERVE Truth Table

A	Y
0	0
1	1

1.14. GRESET (Ask a Question)

This Macro connects an I/O or route an internal fabric signal to the global reset network. The connection to the GRESET is hardened for radiation only if the driver is an I/O fixed at a package pin with GRESET in its function name. Routing an internal fabric signal through the GRESET macro is hardened by the radiation.

Figure 1-13. GRESET

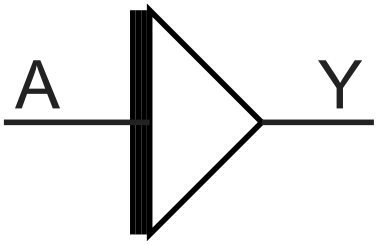


Table 1-25. Truth Table

A	Y
0	0
1	1

1.15. RCLKINT (Ask a Question)

This Macro routes an internal fabric signal to a row global buffer, thus creating a local clock.

Figure 1-14. RCLKINT

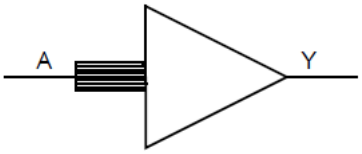


Table 1-26. RCLKINT

Input	Output
A	Y

Table 1-27. Truth Table

A	Y
0	0
1	1

1.16. RGB (Ask a Question)

Back-annotated macro that routes an internal fabric signal to a row global buffer.

Figure 1-15. RGB

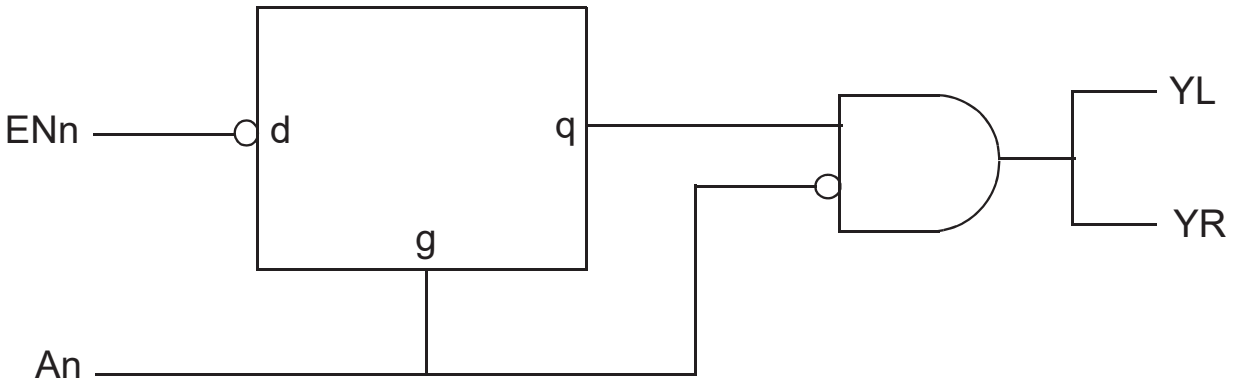


Table 1-28. RGB

Input	Output
An	YL, YR

Table 1-29. Truth Table

An	YL	YR
0	0	0
1	1	1

1.17. RGRESET (Ask a Question)

Macro used to route a triplicated fabric signal to a row global buffer and create a local reset. The three input bits must be driven by three separate logic cones replicating the paths from the source registers.

Figure 1-16. RGRESET

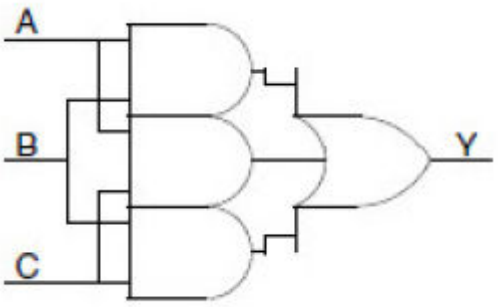


Table 1-30. Truth Table

A[2]	A[1]	A[0]	Y
X	0	0	0
0	X	0	0
0	0	X	0
X	1	1	1
1	X	1	1
1	1	X	1

1.18. SLE (Ask a Question)

Sequential Logic Element.

Figure 1-17. Sequential Logic Element (SLE)

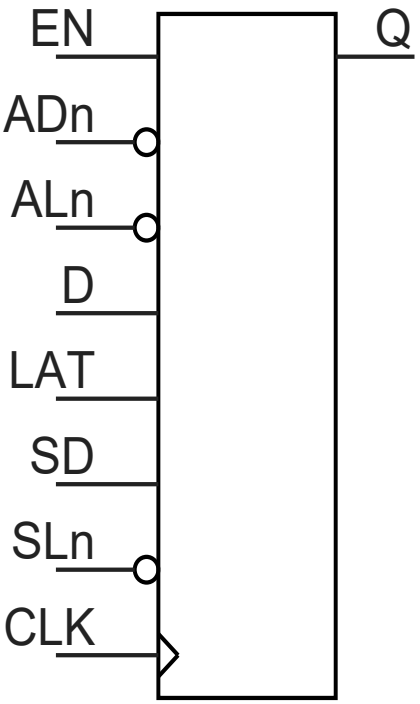


Table 1-31. Sequential Logic Element

Input		Output
Name	Function	Name
D	Data input	Q
CLK	Clock input	
EN	Active-High CLK enable	
ALn	Asynchronous Load. This active-low signal either sets the register or clears the register depending on the value of ADn.	
ADn[1]	Static asynchronous load data. When ALn is active, Q goes to the complement of ADn.	
SLn	Synchronous load. This active-low signal either sets the register or clears the register depending on the value of SD, at the rising edge of the clock.	
SD[1]	Static synchronous load data. When SLn is active (that is, low), Q goes to the value of SD at the rising edge of CLK.	
LAT[1]	LAT is always tied to low. Q output is invalid when LAT=1.	

Note:

1. ADn, SD, and LAT are static signals defined at design time and need to be tied to 0 or 1.

Table 1-32. Truth Table

ALn	ADn	LAT	CLK	EN	SLn	SD	D	Qn+1
0	ADn	X	X	X	X	X	X	!ADn
1	X	0	Not rising	X	X	X	X	Qn
1	X	0	↑	0	X	X	X	Qn
1	X	0	↑	1	0	SD	X	SD
1	X	0	↑	1	1	X	D	D
X	X	1	X	X	X	X	X	Invalid

1.19. SLE_RT (Ask a Question)

Sequential Logic Element macro available only in post-layout netlist.

Figure 1-18. Sequential Logic Element (SLE)

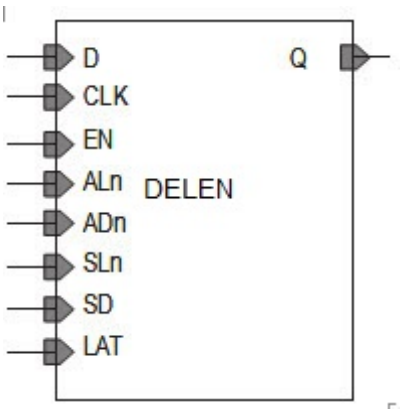


Table 1-33. Sequential Logic Element

Input		Output
Name	Function	Q
D	Data input	
CLK	Clock input	
EN	Active High CLK enable	
ALn	Asynchronous Load. This active low signal either sets the register or clears the register depending on the value of ADn.	
ADn[1]	Static asynchronous load data. When ALn is active, Q goes to the complement of ADn.	
SLn	Synchronous load. This active low signal either sets the register or clears the register depending on the value of SD, at the rising edge of the clock.	
SD[1]	Static synchronous load data. When SLn is active (that is, low), Q goes to the value of SD at the rising edge of CLK.	
DELEN[1]	Enable Single-event Transient mitigation.	

Note:

1. ADn, SD, and DELEN are static signals defined at design time and need to be tied to 0 or 1.

Table 1-34. Truth Table

ALn	ADn	CLK	EN	SLn	SD	D	Qn+1
0	ADn	X	X	X	X	X	!ADn
1	X	Not rising	X	X	X	X	Qn
1	X	↑	0	X	X	X	Qn
1	X	↑	1	0	SD	X	SD
1	X	↑	1	1	X	D	D

1.20. ARI1 (Ask a Question)

The ARI1 macro is responsible for representing all arithmetic operations in the pre-layout phase.

Figure 1-19. ARI1

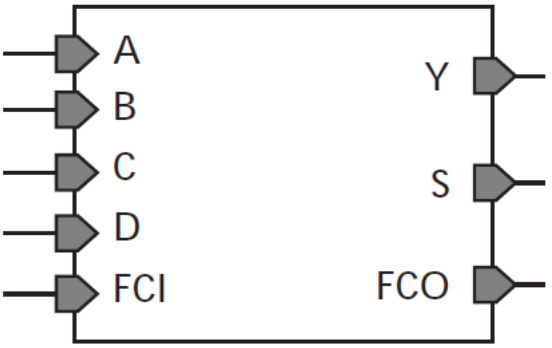


Table 1-35. ARI1

Input	Output
A, B, C, D, FCI	Y, S, FCO

The ARI1 cell has a 20 bit INIT string parameter that is used to configure its functionality. The interpretation of the 16 LSB of the INIT string is shown in the following table. F0 is the value of Y when A = 0 and F1 is the value of Y when A = 1.

Table 1-36. Interpretation of 16 LSB of the INIT String for ARI1

ADCB	Y	
0000	INIT[0]	F0
0001	INIT[1]	
0010	INIT[2]	
0011	INIT[3]	
0100	INIT[4]	
0101	INIT[5]	
0110	INIT[6]	
0111	INIT[7]	
1000	INIT[8]	F1
1001	INIT[9]	
1010	INIT[10]	
1011	INIT[11]	
1100	INIT[12]	
1101	INIT[13]	
1110	INIT[14]	
1111	INIT[15]	

Table 1-37. Truth Table for S

Y	FCI	S
0	0	0
0	1	1
1	0	1
1	1	0

ARI1 LOGIC

[Required-Cleanup]:

The 4 MSB of the INIT string controls the output of the carry bits. The carry is generated using carry propagation and generation bits, which are evaluated according to the following tables.

Table 1-38. ARI1 INIT[17:16] String Interpretation

INIT[17]	INIT[16]	G
0	0	0
0	1	F0
1	0	1
1	1	F1

Table 1-39. ARI1 INIT[19:18] String Interpretation

INIT[19]	INIT[18]	P
0	0	0
0	1	Y
1	X	1

Table 1-40. FCO Truth Table

P	G	FCI	FCO
0	G	X	G
1	X	FCI	FCI

1.21. ARI1_CC (Ask a Question)

The ARI1_CC macro is responsible for representing all arithmetic operations in the post-layout phase. It performs all the functions of the ARI1 macro except that it does not generate the Final Carry Out (FCO).

Note: FC1 and FC0 do not perform any functionalities.

Figure 1-20. ARI1_CC

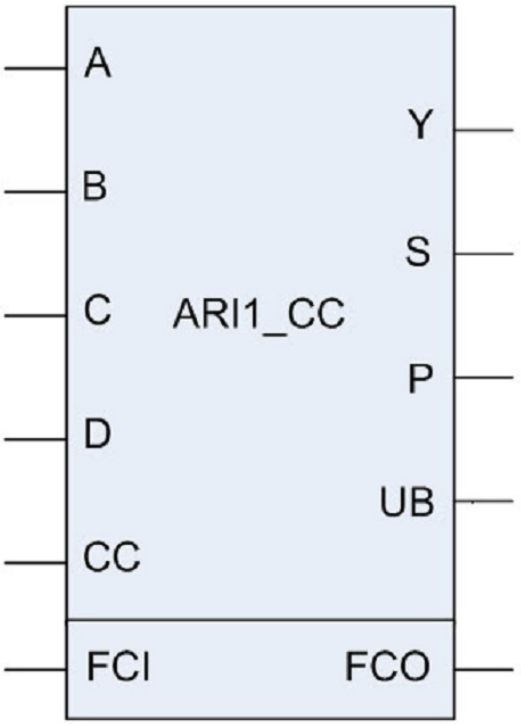


Table 1-41. ARI1_CC

Input	Output
A, B, C, D, CC	Y, S, P, UB

The ARI1_CC cell has a 20-bit INIT string parameter that is used to configure its functionality. The interpretation of the 16 LSB of the INIT string is shown in the following table. F0 is the value of Y when A = 0 and F1 is the value of Y when A = 1. The following table shows the interpretation of 16 LSB of the INIT string for AR1_CC.

Table 1-42. Interpretation of 16 LSB of the INIT String for AR1_CC

ADCB	Y	
0000	INIT[0]	F0
0001	INIT[1]	
0010	INIT[2]	
0011	INIT[3]	
0100	INIT[4]	
0101	INIT[5]	
0110	INIT[6]	
0111	INIT[7]	

Table 1-42. Interpretation of 16 LSB of the INIT String for AR1_CC (continued)		
ADCB	Y	
1000	INIT[8]	F1
1001	INIT[9]	
1010	INIT[10]	
1011	INIT[11]	
1100	INIT[12]	
1101	INIT[13]	
1110	INIT[14]	
1111	INIT[15]	

The 4 MSB of the INIT string controls the output of the carry bits. The carry is generated using carry propagation and generation bits, which are evaluated according to the following tables.

Table 1-43. ARI1_CC INIT[17:16] String Interpretation

INIT[17]	INIT[16]	UB
0	0	1
0	1	!F0
1	0	0
1	1	!F1

Table 1-44. ARI1_CC INIT[19:18] String Interpretation

INIT[19]	INIT[18]	P
0	0	0
0	1	Y
1	X	1

The equation of S is given by:

S=Y^CC

1.22. CC_CONFIG (Ask a Question)

The CC_CONFIG macro is responsible for generating the Carry bit for each ARI1_CC cell in the cluster.

Figure 1-21. CC_CONFIG

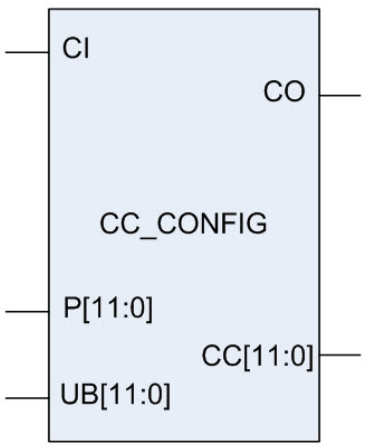


Table 1-45. CC_CONFIG

Input	Output
CI, P, UB	CO, CC

CI and CO are the carry-in and carry-out, respectively, to the cell. The intermediate carry-bits are given by CC[11:0]. The functionality of the CC_CONFIG is basically evaluating CC using

CC[n] = !Px!UB+PxCC[n-1]

where CC[-1] is CI and CC[12] is CO.

Inside every cluster, there are 12 ARI1_CC cells and only one CC_CONFIG cell. The CC_CONFIG takes as input the P and UB outputs of each ARI1_CC cell in the cluster and generated the CC (carry-out bit), which is then fed to the next ARI1_CC cell in the cluster as the carry-in.

[Required-Cleanup]:

1.23. FCEND_BUFF (Ask a Question)

Buffer, driven by the FCO pin of the last macro in the Carry-Chain.

Figure 1-22. FCEND_BUFF

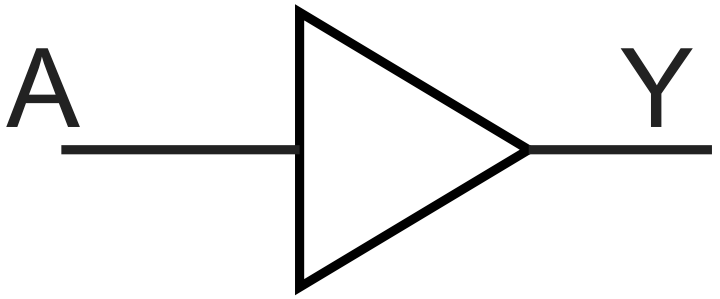


Table 1-46. FCEND_BUFF

Input	Output
A	Y

Table 1-47. Truth Table

A	Y
0	0
1	1

1.23.1. FCINT_BUFF (Ask a Question)

Name special Buffer, used to initialize the FCI pin of the first macro in the Carry-Chain.

Figure 1-23. FCINT_BUFF

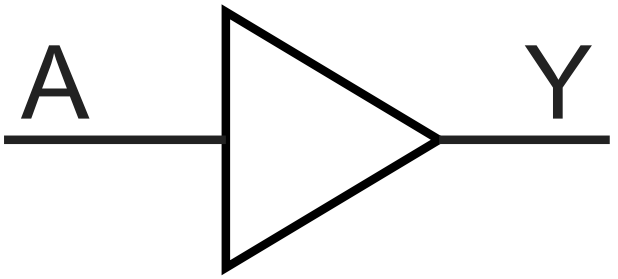


Table 1-48. FCINT_BUFF

Input	Output
A	Y

Table 1-49. Truth Table

A	Y
0	0
1	1

1.24. RCOSC_50MHZ (Ask a Question)

The RCOSC_50MHZ oscillator is an RC oscillator that provides a free-running clock of 50 MHz at CLKOUT.

Figure 1-24. RCOSC_50MHZ

1.25. SYSRESET (Ask a Question)

It is a special-purpose macro. The Output POWER_ON_RESET_N goes low at power-up and when DEVRST_N goes low.

Figure 1-25. SYSRESET

Table 1-50. SYSRESET

Input	Output
DEVRST_N	POWER_ON_RESET_N

Table 1-51. Truth Table

DEVRST_N	POWER_ON_RESET_N
0	0
1	1

1.26. SYSCTRL_RESET_STATUS (Ask a Question)

This special-purpose macro checks the status of the System Controller. The output port RESET_STATUS goes high when the System Controller is in reset. To enable this macro, select the "**Enable System Controller Suspend Mode**" option in the "**Configure Programming Bitstream Settings**" tool in Libero. After programming, the device enters System Controller Suspend Mode if TRSTB is tied low during power-up.

Important: This macro does not support simulation. To simulate the System Controller suspend mode, add the following pseudo-code to the simulation testbench:

- At simulation time t = 0, set `RESET_STATUS` = 0.
- 1.46 µs after observing `POWER_ON_RESET_N` = 1, set `RESET_STATUS` = 1 to indicate that the system controller has entered suspend mode.

Figure 1-26. SYSCTRL_RESET_STATUS

1.27. DFN1 (Ask a Question)

D-Type Flip-Flop.

Figure 1-27. DFN1

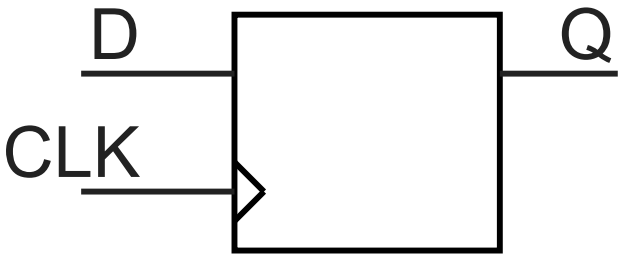


Table 1-52. DFN1 I/O

Input	Output
D, CLK	Q

Table 1-53. DFN1 Truth Table

CLK	D	Q_{n+1}
not Rising	X	Q_n
—	D	D

1.28. DFN1C0 (Ask a Question)

D-Type Flip-Flop with active-low Clear.

Figure 1-28. DFN1C0

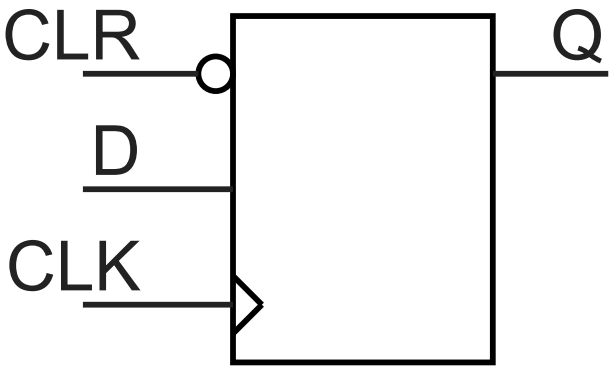


Table 1-54. DFN1C0 I/O

Input	Output
D, CLK, CLR	Q

Table 1-55. DFN1C0 Truth Table

CLR	CLK	D	Q_{n+1}
0	X	X	0
1	not Rising	X	Q_n
1	—	D	D

1.29. DFN1E1 (Ask a Question)

D-Type Flip-Flop with active high Enable.

Figure 1-29. DFN1E1

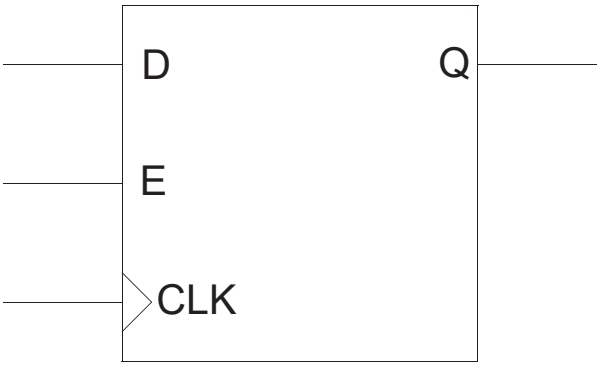


Table 1-56. DFN1E1 I/O

Input	Output
D, E, CLK	Q

Table 1-57. DFN1E1 Truth Table

E	CLK	D	Q_{n+1}
0	X	X	Q_n
1	not Rising	X	Q_n

Table 1-57. DFN1E1 Truth Table (continued)			
E	CLK	D	Q_{n+1}
1	—	D	D

1.30. DFN1E1C0 (Ask a Question)

D-Type Flip-Flop, with active-high Enable and active-low Clear.

Figure 1-30. DFN1E1C0

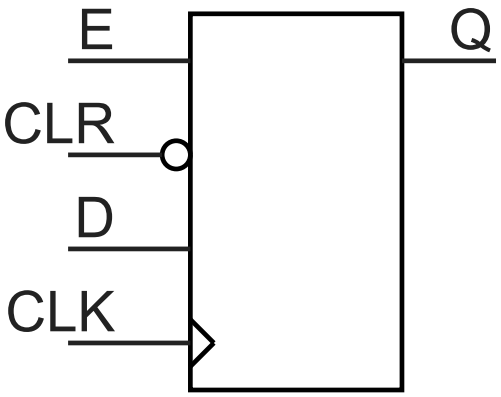


Table 1-58. DFN1E1C0 I/O

Input	Output
CLR, D, E, CLK	Q

Table 1-59. DFN1E1C0 Truth Table

CLR	E	CLK	D	Q_{n+1}
0	X	X	X	0
1	0	X	X	Q_n
1	1	not Rising	X	Q_n
1	1	—	D	D

1.31. DFN1E1P0 (Ask a Question)

D-Type Flip-Flop with active-high Enable and active-low Preset.

Figure 1-31. DFN1E1P0

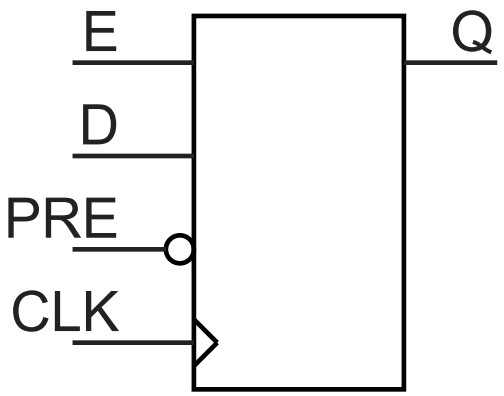


Table 1-60. DFN1E1P0

Input	Output
D, E, PRE, CLK	Q

Table 1-61. Truth Table

PRE	E	CLK	D	Q_{n+1}
0	X	X	X	1
1	0	X	X	Qn
1	1	not Rising	X	Qn
1	1	↑	D	D

1.32. DFN1P0 (Ask a Question)

D-Type Flip-Flop with active-Low Preset.

Figure 1-32. DFN1P0

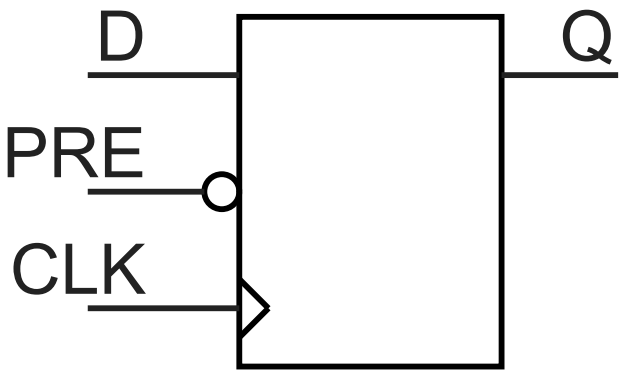


Table 1-62. DFN1P0 I/O

Input	Output
D, PRE, CLK	Q

Table 1-63. DFN1P0 Truth Table

PRE	CLK	D	Q_{n+1}
0	X	X	1
1	not Rising	X	Qn
1	—	D	D

2. hm4 (Ask a Question)

2.1. INV (Ask a Question)

Inverter.

Figure 2-1. INV

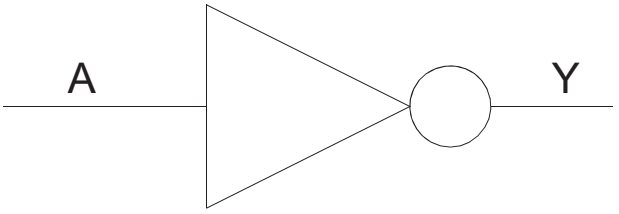


Table 2-1. INV I/O

Input	Output
A	Y

Table 2-2. INV Truth Table

A	Y
0	1
1	0

2.2. INVD (Ask a Question)

Inverter.
Note: Compile optimization does not remove this macro.

Figure 2-2. INVD

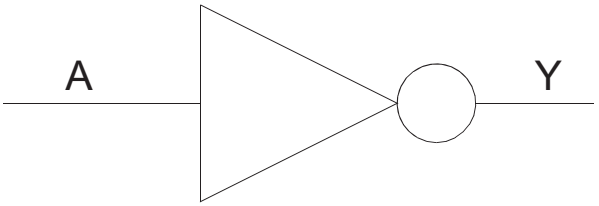


Table 2-3. INVD I/O

Input	Output
A	Y

Table 2-4. INVD Truth Table

A	Y
0	1
1	0

2.3. MX2 (Ask a Question)

2 to 1 Multiplexer.

Figure 2-3. MX2

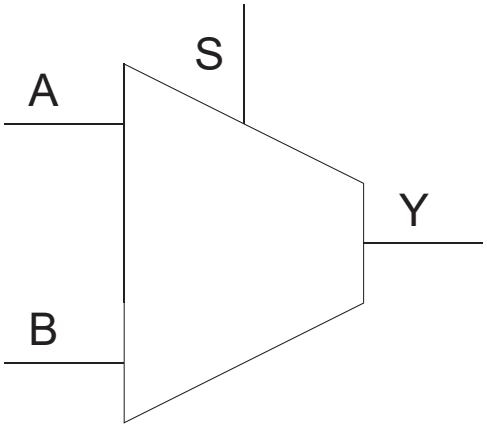


Table 2-5. MX2 I/O

Input	Output
A, B, S	Y

Table 2-6. MX2 Truth Table

A	B	S	Y
A	X	0	A
X	B	1	B

2.4. MX4 (Ask a Question)

4 to 1 Multiplexer.

This macro uses two logic modules.

Figure 2-4. MX4

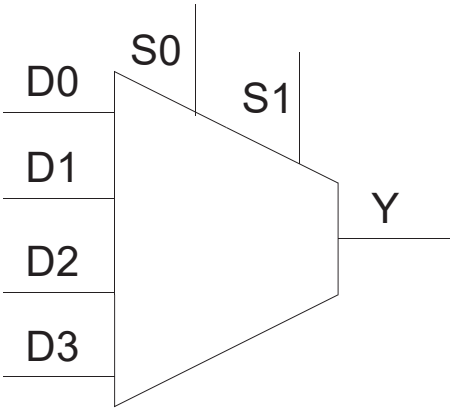


Table 2-7. MX4 I/O

Input	Output
D0, D1, D2, D3, S0, S1	Y

Table 2-8. MX4 Truth Table

D3	D2	D1	D0	S1	S0	Y
X	X	X	D0	0	0	D0
X	X	D1	X	0	1	D1
X	D2	X	X	1	0	D2
D3	X	X	X	1	1	D3

2.5. NAND2 (Ask a Question)

2-Input NAND.

Figure 2-5. NAND2

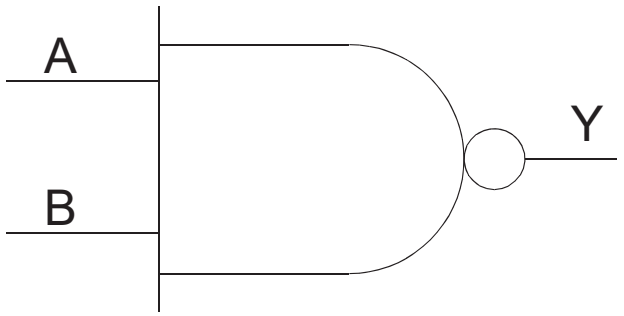


Table 2-9. NAND2 I/O

Input	Output
A, B	Y

Table 2-10. NAND2 Truth Table

A	B	Y
X	0	1
0	X	1
1	1	0

2.6. NAND3 (Ask a Question)

3-Input NAND.

Figure 2-6. NAND3

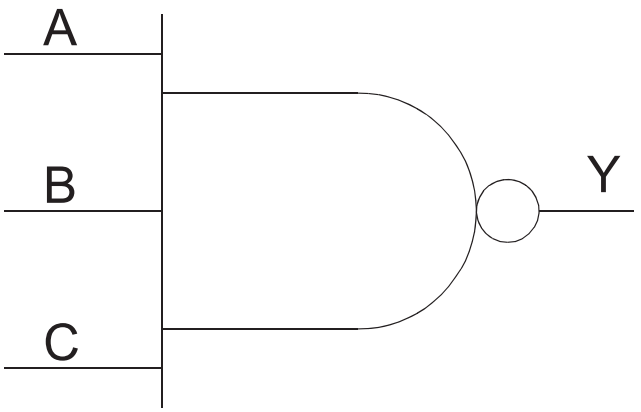


Table 2-11. NAND3 I/O

Input	Output
A, B, C	Y

Table 2-12. NAND3 Truth Table

A	B	C	Y
X	X	0	1
X	0	X	1
0	X	X	1
1	1	1	0

2.7. NAND4 (Ask a Question)

4-input NAND.

Figure 2-7. NAND4

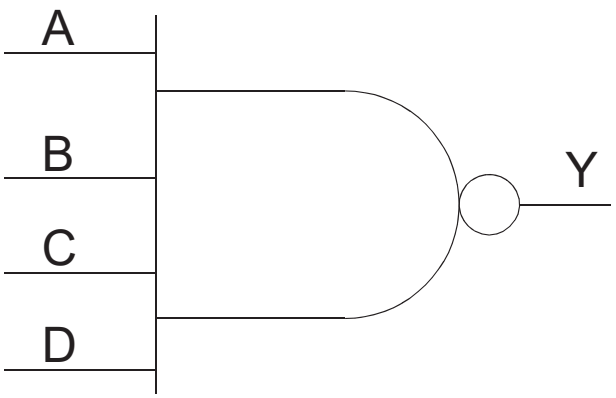


Table 2-13. NAND4 I/O

Input	Output
A, B, C, D	Y

Table 2-14. NAND4 Truth Table

A	B	C	D	Y
X	X	X	0	1
X	X	0	X	1
X	0	X	X	1
0	X	X	X	1
1	1	1	1	0

2.8. NOR2 (Ask a Question)

2-input NOR.

Figure 2-8. NOR2

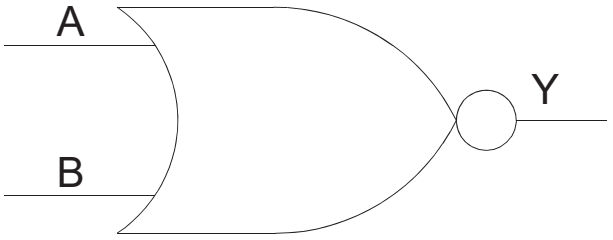


Table 2-15. NOR2 I/O

Input	Output
A, B	Y

Table 2-16. NOR2 Truth Table

A	B	Y
0	0	1
X	1	0
1	X	0

2.9. NOR3 (Ask a Question)

3-input NOR.

Figure 2-9. NOR3

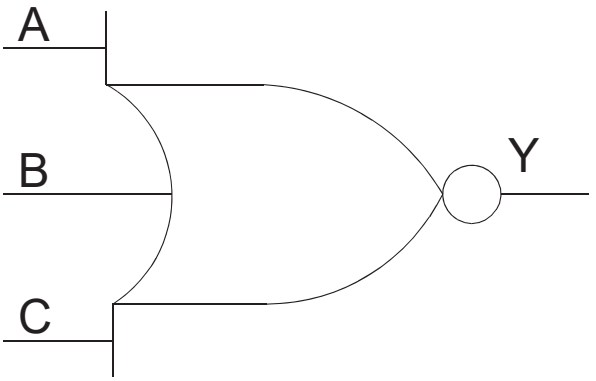


Table 2-17. NOR3 I/O

Input	Output
A, B, C	Y

Table 2-18. NOR3 Truth Table

A	B	C	Y
0	0	0	1
X	X	1	0
X	1	X	0
1	X	X	0

2.10. NOR4 (Ask a Question)

4-input NOR.

Figure 2-10. NOR3

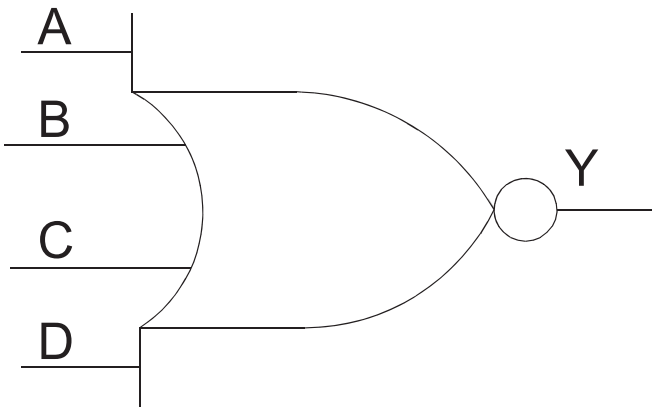


Table 2-19. NOR4 I/O

Input	Output
A, B, C, D	Y

Table 2-20. NOR4 Truth Table

A	B	C	D	Y
0	0	0	0	1
1	X	X	X	0
X	1	X	X	0
X	X	1	X	0
X	X	X	1	0

2.11. OR2 (Ask a Question)

2-input OR.

Figure 2-11. OR2

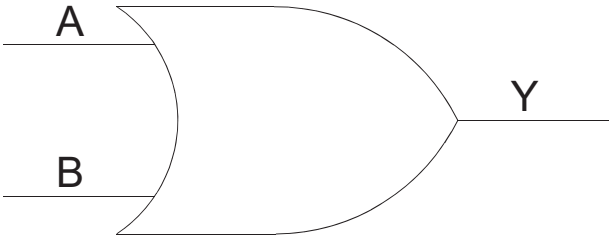


Table 2-21. OR2 I/O

Input	Output
A, B	Y

Table 2-22. OR2 Truth Table

A	B	Y
0	0	0
X	1	1

Table 2-22. OR2 Truth Table (continued)

A	B	Y
1	X	1

2.12. OR3 (Ask a Question)

3-input OR.

Figure 2-12. OR3

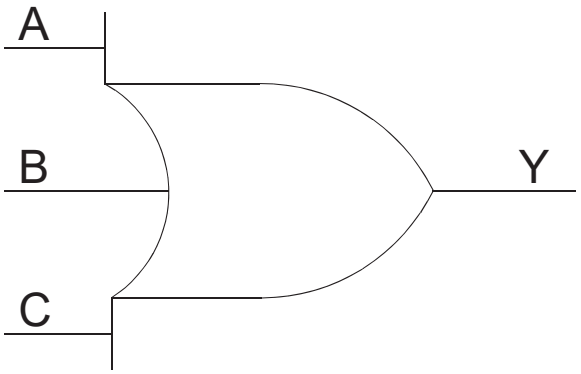


Table 2-23. OR3 I/O

Input	Output
A, B, C	Y

Table 2-24. OR3 Truth Table

A	B	C	Y
0	0	0	0
X	X	1	1
X	1	X	1
1	X	X	1

2.13. OR4 (Ask a Question)

4-input OR.

Figure 2-13. OR4

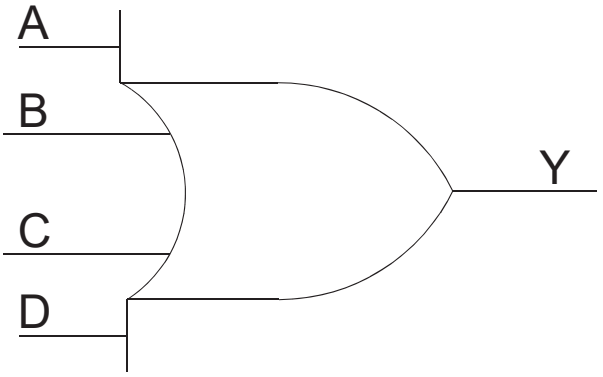


Table 2-25. OR4 I/O

Input	Output
A, B, C, D	Y

Table 2-26. OR4 Truth Table

A	B	C	D	Y
0	0	0	0	0
1	X	X	X	1
X	1	X	X	1
X	X	1	X	1
X	X	X	1	1

2.14. XOR2 (Ask a Question)

2-input XOR.

Figure 2-14. XOR2

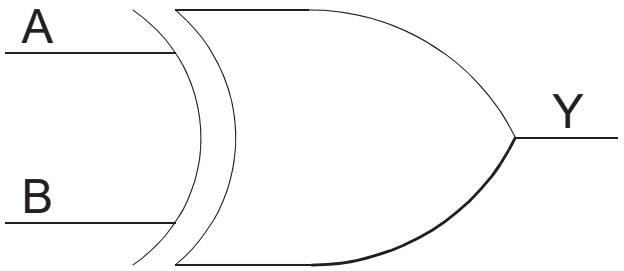


Table 2-27. XOR2 I/O

Input	Output
A, B	Y

Table 2-28. XOR2 Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2.15. XOR3 (Ask a Question)

3-input XOR.

Figure 2-15. XOR3

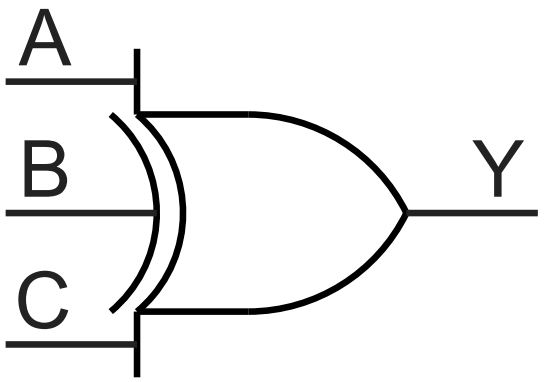


Table 2-29. XOR3 I/O

Input	Output
A, B, C	Y

Table 2-30. XOR3 Truth Table

A	B	C	Y
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	0
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	1

2.16. XOR4 (Ask a Question)

4-input XOR.

Figure 2-16. XOR4

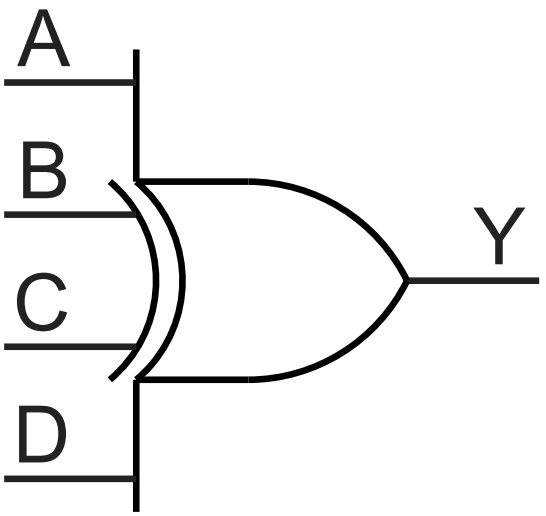


Table 2-31. XOR4 I/O

Input	Output
A, B, C, D	Y

Table 2-32. XOR4 Truth Table

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

2.17. XOR8 (Ask a Question)

8-input XOR.

This macro uses two logic modules.

Figure 2-17. XOR8

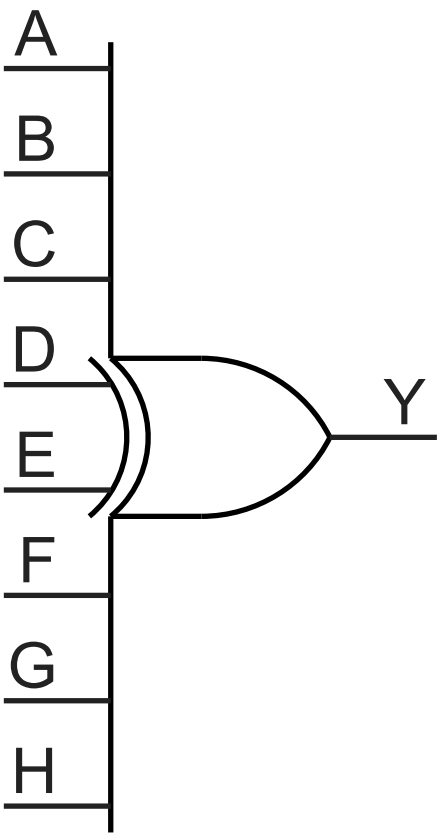


Table 2-33. XOR8 I/O

Input	Output
A, B, C, D, E, F, G, H	Y

If you have an odd number of inputs that are High, the output is High (1).

If you have an even number of inputs that are High, the output is Low (0).

For example:

Table 2-34. XOR8 Truth Table

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

2.18. UJTAG (Ask a Question)

The UJTAG macro is a special purpose macro. It allows access to the user JTAG circuitry on board the chip. You must instantiate a UJTAG macro in your design if you plan to make use of the user JTAG feature. The TMS, TDI, TCK, TRSTB, and TDO pins of the macro must be connected to top level ports of the design.

Figure 2-18. UJTAG

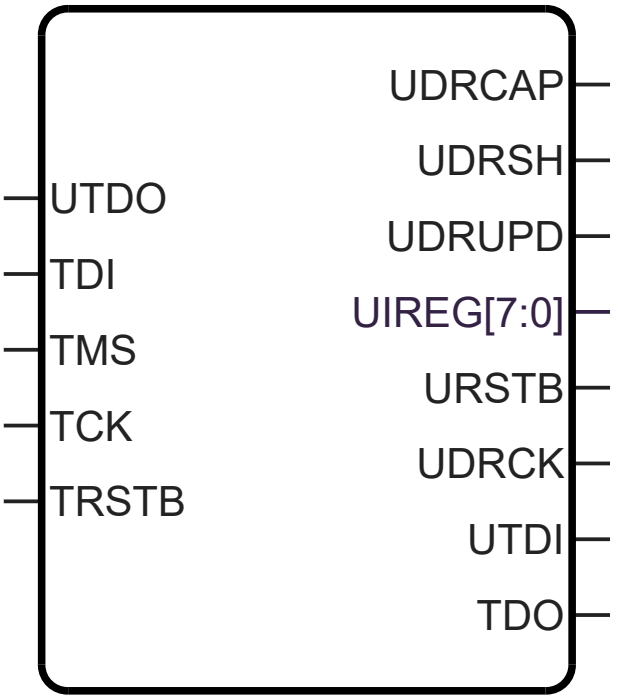


Table 2-35. Ports and Descriptions

Port	Direction	Polarity	Description
UIREG[7:0]	Output	—	This 8-bit bus carries the contents of the JTAG instruction register of each device. Instruction values 16 to 127 are not reserved and can be employed as user-defined instructions.
URSTB	Output	Low	URSTB is an Active-Low signal and is asserted when the TAP controller is in Test-Logic-Reset mode. URSTB is asserted at power-up, and a Power-on Reset signal resets the TAP controller state.
UTDI	Output	—	This port is directly connected to the TAP's TDI signal.
UTDO	Input	—	This port is the user TDO output. Inputs to the UTDO port are sent to the TAP TDO output MUX when the IR addess is in user range.
UDRSH	Output	High	Active-High signal enabled in the Shift_DR TAP state.
UDRCAP	Output	High	Active-High signal enabled in the Capture_DR_TAP state.
UDRCK	Output	—	This port is directly connected to the TAP's TCK signal.
UDRUPD	Output	High	Active-High signal enabled in the Update_DR_TAP state.
TCK	Input	—	Test Clock Serial input for JTAG boundary scan, ISP, and UJTAG. The TCK pin does not have an internal pull-up/pull-down resistor. Connect TCK to GND or 3.3V through a resistor (500–1 KΩ) placed closed to the FPGA pin to prevent totem-pole current on the input buffer and TMS from entering into an undesired state. If JTAG is not used, connect it to GND.
TDI	Input	—	Test Data in. Serial input for JTAG boundary scan. There is an internal weak pull-up resistor on the TDI pin.
TDO	Output	—	Test Data Out. Serial output for JTAG boundary scan. The TDO pin does not have an internal pull-up/pull-down resistor.
TMS	Input	—	Test mode select. The TMS pin controls the use of the IEEE® 1532 boundary scan pins (TCK, TDI, TDO, and TRST). There is an internal weak pull-up resistor on the TMS pin.

Table 2-35. Ports and Descriptions (continued)

Port	Direction	Polarity	Description
TRSTB	Input	Low	Test reset. The TRSTB pin is an active-low input. It synchronously initializes (or resets) the boundary scan circuitry. There is an internal weak pull-up resistor on the TRSTB pin. To hold the JTAG in reset mode and prevent it from entering into undesired states in critical applications, connect TRSTB to GND through a 1 KΩ resistor (placed close to the FPGA pin).

3. IO1 (Ask a Question)

3.1. BIBUF (Ask a Question)

Bidirectional Buffer.

Figure 3-1. BIBUF

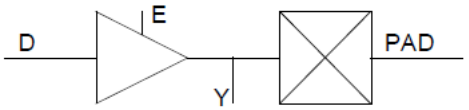


Table 3-1. BIBUF

Input	Output
D, E, PAD	PAD, Y

Table 3-2. Truth Table

MODE	E	D	PAD	Y
OUTPUT	1	D	D	D
INPUT	0	X	Z	X
INPUT	0	X	PAD	PAD

3.2. BIBUF_DIFF (Ask a Question)

Bidirectional Buffer, Differential I/O.

Figure 3-2. BIBUF_DIFF

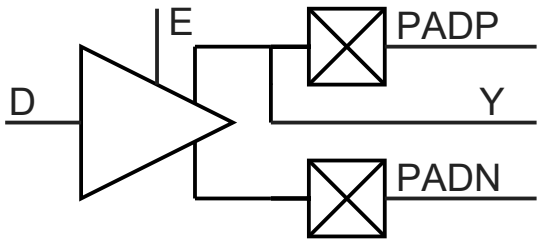


Table 3-3. BIBUF_DIFF

Input	Output
D, E, PADP, PADN	PADP, PADN, Y

Table 3-4. Truth Table

MODE	E	D	PADP	PADN	Y
OUTPUT	1	0	0	1	0
OUTPUT	1	1	1	0	1
INPUT	0	X	Z	Z	X
INPUT	0	X	0	0	X

Table 3-4. Truth Table (continued)

MODE	E	D	PADP	PADN	Y
INPUT	0	X	1	1	X
INPUT	0	X	0	1	0
INPUT	0	X	1	0	1

3.3. CLKBIBUF (Ask a Question)

Bidirectional Buffer with Input to global network.

Figure 3-3. CLKBIBUF

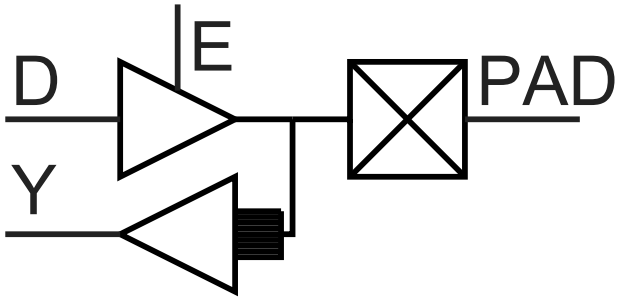


Table 3-5. CLKBIBUF

Input	Output
D, E, PAD	PAD, Y

Table 3-6. Truth Table

D	E	PAD	Y
X	0	Z	X
X	0	0	0
X	0	1	1
0	1	0	0
1	1	1	1

3.4. CLKBUF (Ask a Question)

Input Buffer to global network.

Figure 3-4. CLKBUF

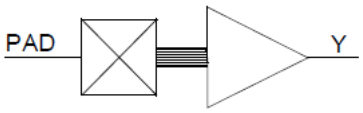


Table 3-7. CLKBUF

Input	Output
PAD	Y

Table 3-8. Truth Table

PAD	Y
0	0
1	1

3.5. CLKBUF_DIFF (Ask a Question)

Differential I/O macro to global network, Differential I/O.

Figure 3-5. INBUF_DIFF

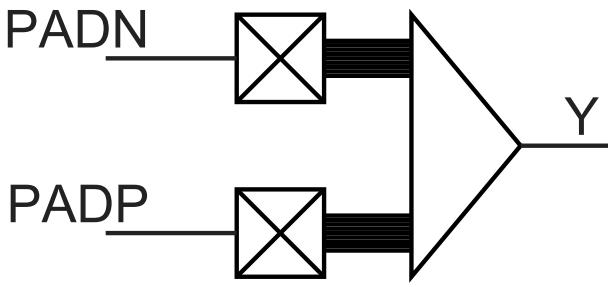


Table 3-9. INBUF_DIFF

Input	Output
PADP, PADN	Y

Table 3-10. Truth Table

PADP	PADN	Y
Z	Z	Y
0	0	X
1	1	X
0	1	0
1	0	1

3.6. INBUF (Ask a Question)

Input Buffer.

Figure 3-6. INBUF

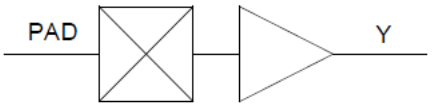


Table 3-11. INBUF

Input	Output
PAD	Y

Table 3-12. Truth Table

PAD	Y
Z	X
0	0
1	1

3.7. INBUF_DIFF (Ask a Question)

Input Buffer, Differential I/O.

Figure 3-7. INBUF_DIFF

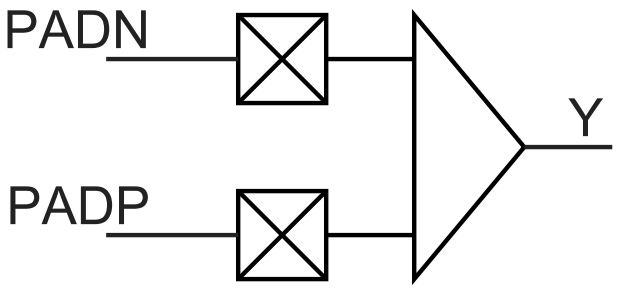


Table 3-13. INBUF_DIFF

Input	Output
PADP, PADN	Y

Table 3-14. Truth Table

PADP	PADN	Y
Z	Z	X
0	0	X
1	1	X
0	1	0
1	0	1

3.8. IOINFF_BYPASS (Ask a Question)

The I/O input bypass macro is available in post-layout netlist only.

Figure 3-8. IOINFF_BYPASS

Table 3-15. IOINFF_BYPASS

Input	Output
A	Y

Table 3-16. Truth Table

A	Y
0	0
1	1

3.9. IOENFF_BYPASS (Ask a Question)

The I/O enable bypass macro is available in post-layout netlist only.

Figure 3-9. IOENFF_BYPASS

Table 3-17. IOENFF_BYPASS

Input	Output
A	Y

Table 3-18. Truth Table

A	Y
0	0
1	1

3.10. IOOUTFF_BYPASS (Ask a Question)

The I/O output bypass macro is available in post-layout netlist only.

Figure 3-10. IOOUTFF_BYPASS

Table 3-19. IOOUTFF_BYPASS

Input	Output
A	Y

Table 3-20. Truth Table

A	Y
0	0
1	1

3.11. IOPAD_BI (Ask a Question)

The I/O output bypass macro is available in post-layout netlist only.

Figure 3-11. IOPAD_BI

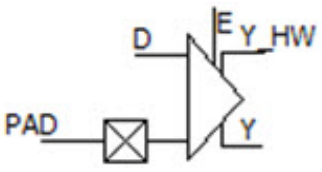


Table 3-21. IOPAD_BI

Input	Output
D, E, PAD	PAD, Y, Y_HW

Table 3-22. Truth Table

MODE	E	D	PAD	Y	Y_HW
OUTPUT	1	D	D	D	D
INPUT	0	X	Z	X	X
INPUT	0	X	PAD	PAD	PAD

3.12. IOPADP_BI (Ask a Question)

The I/O PAD bi-directional macro is available in post-layout netlist only.

Figure 3-12. IOPADP_BI

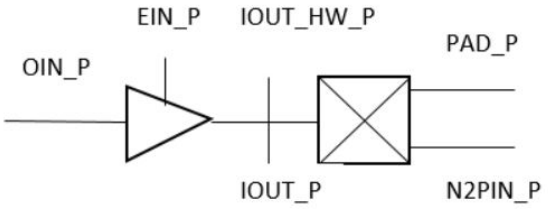


Table 3-23. IOPADP_BI

Input	Output
N2PIN_P, OIN_P, EIN_P, PAD_P	PAD_P, IOUT_P, IOUT_HW_P

Table 3-24. Truth Table

MODE	EIN_P	OIN_P	PAD_P	N2PIN_P	IOUT_P	OUT_HW_P
OUTPUT	1	0	0	1	0	0
OUTPUT	1	1	1	0	1	1
INPUT	0	X	Z	Z	X	X
INPUT	0	X	0	0	X	X
INPUT	0	X	1	1	X	X
INPUT	0	X	0	1	0	0
INPUT	0	X	1	0	1	1

3.13. IOPADN_BI (Ask a Question)

The I/O PAD bi-directional macro is available in post-layout netlist only.

Figure 3-13. IOPADN_BI

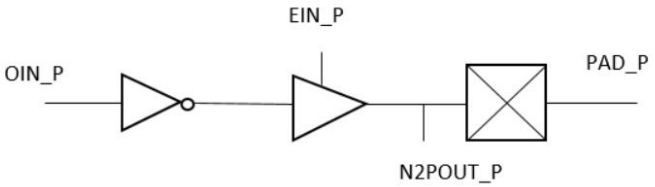


Table 3-25. IOPADN_BI

Input	Output
OIN_P, EIN_P, PAD_P	PAD_P, N2POUT_P

Table 3-26. Truth Table

MODE	EIN_P	OIN_P	PAD_P	N2POUT_P
OUTPUT	1	1	0	0
OUTPUT	1	0	1	1
INPUT	0	X	Z	X
INPUT	0	X	0	X
INPUT	0	X	1	X
INPUT	0	X	0	0
INPUT	0	X	1	1

3.14. IOPADP_IN (Ask a Question)

The I/O PAD input macro is available in post-layout netlist only.

Figure 3-14. IOPADP_IN

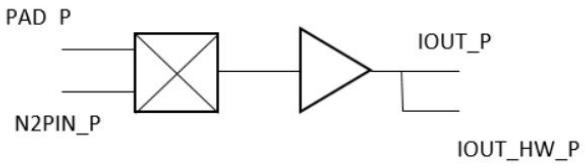


Table 3-27. IOPADP_IN

Input	Output
PAD_P, N2PIN_P	IOUT_P, IOUT_HW_P

Table 3-28. Truth Table

PAD_P	N2PIN_P	IOUT_P	IOUT_HW_P
Z	X	X	X
0	X	0	0
1	X	1	1

3.15. IOPADN_IN (Ask a Question)

The I/O PAD input macro is available in post-layout netlist only.

Figure 3-15. IOPADN_IN

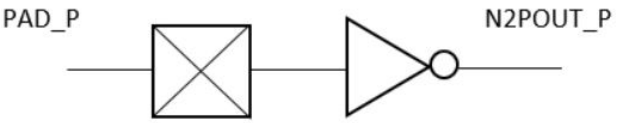


Table 3-29. IOPADN_IN

Input	Output
PAD_P	N2POUT_P

Table 3-30. Truth Table

PAD_P	N2POUT_P
0	1

Table 3-30. Truth Table (continued)	
PAD_P	N2POUT_P
1	0

3.16. IOPADP_TRI (Ask a Question)

The I/O PAD tristate output macro is available in post-layout netlist only.

Figure 3-16. IOPADP_TRI

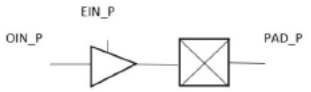


Table 3-31. IOPADP_TRI

Input	Output
OIN_P, EIN_P	PAD_P

Table 3-32. Truth Table

OIN_P	EIN_P	PAD_P
X	0	Z
OIN_P	1	OIN_P

3.17. IOPADN_TRI (Ask a Question)

The I/O PAD tristate output macro is available in post-layout netlist only.

Figure 3-17. IOPADN_TRI

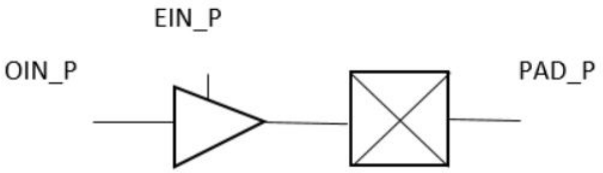


Table 3-33. IOPADN_TRI

Input	Output
OIN_P, EIN_P	PAD_P

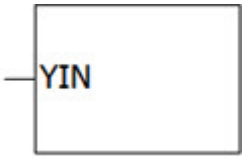
Table 3-34. Truth Table

OIN_P	EIN_P	PAD_P
X	0	Z
0	1	1
1	1	0

3.18. IO_DIFF (Ask a Question)

The I/O Differential macro is available only in post-layout netlist (place holder to reserve the N location).

Figure 3-18. IO_DIFF



Input = YIN

3.19. IOTRI_OB_EB (Ask a Question)

The I/O feed through macro is available in post-layout netlist only.

Figure 3-19. IOTRI_OB_EB

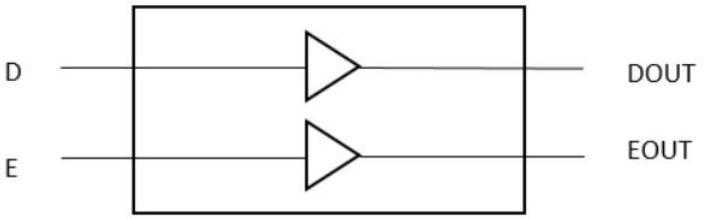


Table 3-35. IOTRI_OB_EB

Input	Output
D, E	DOUT, EOUT

Table 3-36. Truth Table

D	DOUT
0	0
1	1

Table 3-37. Truth Table

E	EOUT
0	0
1	1

3.20. IOBI_IB_OB_EB (Ask a Question)

The I/O feed through macro is available in post-layout netlist only.

Figure 3-20. IOBI_IB_OB_EB

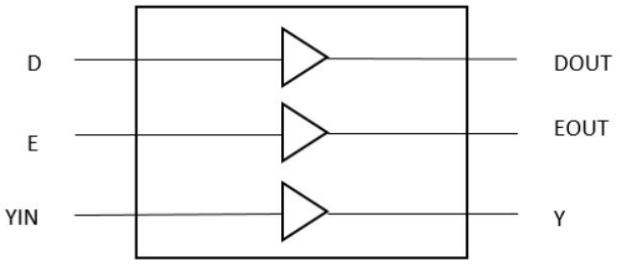


Table 3-38. IOBI_IB_OB_EB

Input	Output
D, E, YIN	DOUT, EOUT, Y

Table 3-39. Truth Table

D	DOUT
0	0
1	1

Table 3-40. Truth Table

E	EOUT
0	0
1	1

Table 3-41. Truth Table

YIN	Y
0	0
1	1

3.21. OUTBUF (Ask a Question)

Output buffer.

Figure 3-21. OUTBUF

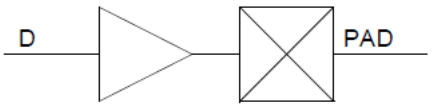


Table 3-42. OUTBUF

Input	Output
D	PAD

Table 3-43. Truth Table

D	PAD
0	0
1	1

3.22. OUTBUF_DIFF (Ask a Question)

Output buffer, Differential I/O.

Figure 3-22. OUTBUF_DIFF

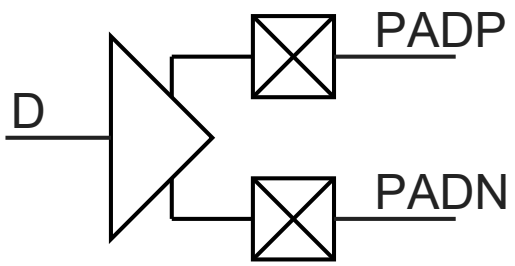


Table 3-44. OUTBUF_DIFF

Input	Output
D	PADP, PADN

Table 3-45. Truth Table

D	PADP	PADN
0	0	1
1	1	0

3.23. TRIBUFF (Ask a Question)

Tristate output buffer.

Figure 3-23. TRIBUFF

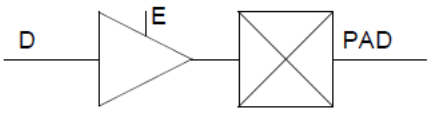


Table 3-46. TRIBUFF

Input	Output
D, E	PAD

Table 3-47. Truth Table

D	E	PAD
X	0	Z
D	1	D

3.24. TRIBUFF_DIFF (Ask a Question)

Tristate output buffer, Differential I/O.

Figure 3-24. TRIBUFF_DIFF

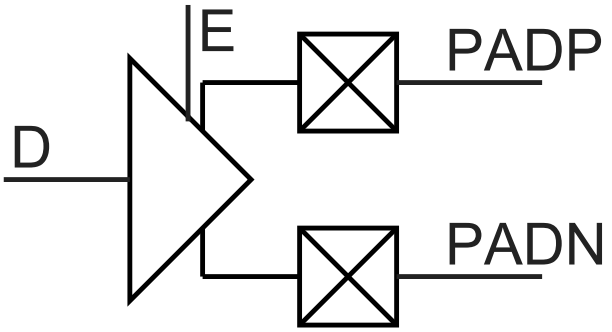


Table 3-48. TRIBUFF_DIFF

Input	Output
D, E	PADP, PADN

Table 3-49. Truth Table

D	E	PADP	PADN
X	0	Z	Z
0	1	0	1
1	1	1	0

3.25. DDR_IN (Ask a Question)

The DDR_IN macro is available for both pre-layout and post-layout simulation flows. It consists of two SLE macros and a latch. The input D must be connected to an I/O.

Figure 3-25. DDR_IN

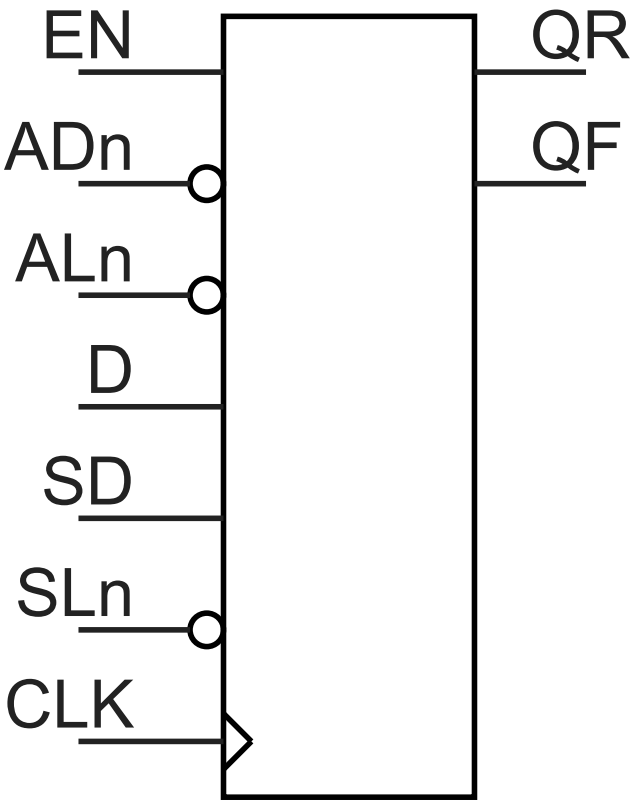


Table 3-50. DDR_IN

Input		Output
Name	Function	Name
D	Data input	QR QF
CLK	Clock input	
EN	Active High CLK enable	
ALn	Asynchronous load. This active low signal either sets the register or clears the register depending on the value of ADn.	
ADn[1]	Static asynchronous load data. When ALn is active, QR and QF go to the complement of ADn.	
SLn	Synchronous load. This active low signal either sets the register or clears the register depending on the value of SD, at the rising edge of CLK.	
SD[1]	Static synchronous load data. When SLn is active (that is, low), QR and QF go to the value of SD at the rising edge of CLK.	

Note:

1. ADn and SD are static inputs defined at design time and need to be tied to 0 or 1.

Table 3-51. Truth Table

ALn	CLK	EN	SLn	dfn+1 (Internal Signal)	QRn+1	QFn+1
0	X	X	X	!ADn	!ADn	!ADn
1	Not rising	X	X	df_n	QRn	QFn
1	↑	0	X	df_n	QRn	QFn
1	↑	1	0	df_n	SD	SD
1	↑	1	1	df_n	D	dfn
1	↓	X	X	D	QRn	QFn

3.26. DDR_OUT (Ask a Question)

The DDR_OUT macro is an output DDR cell and is available for pre-layout simulation. It consists of two SLE macros. The output Q must be connected to an I/O.

Figure 3-26. DDR_OUT

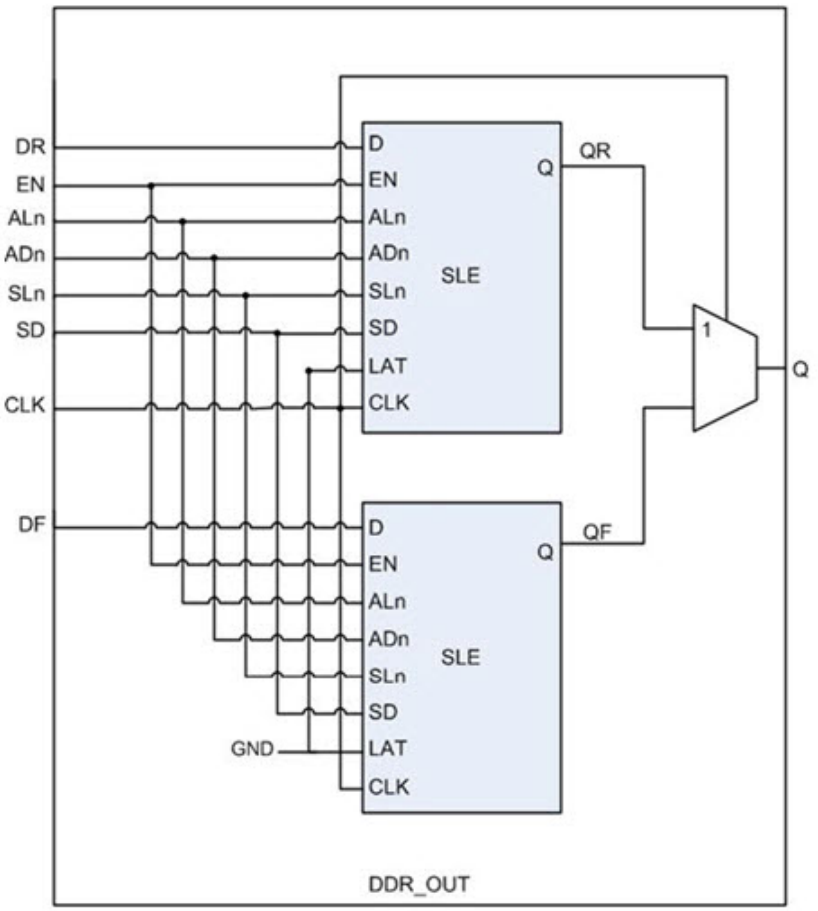


Table 3-52. DDR_OUT

Input		Output
Name	Function	
DR	Data input (Rising Edge)	Q
DF	Data input (Falling Edge)	
CLK	Clock input	
EN	Active High CLK enable	
ALn	Asynchronous load. This active low signal either sets the register or clears the register depending on the value of ADn.	
ADn[1]	Static asynchronous load data. When ALn is active, Q goes to the complement of ADn.	
SLn	Synchronous load. This active low signal either sets the register or clears the register depending on the value of SD, at the rising edge of CLK.	
SD[1]	Static synchronous load data. When SLn is active (that is, low), Q goes to the value of SD at the rising edge of CLK.	

Note:

1. ADn and SD are static inputs defined at design time and need to be tied to 0 or 1.

Table 3-53. Truth Table

ALn	CLK	EN	SLn	QR_{n+1}	QF_{n+1}	Qn+1
0	X	X	X	!ADn	!ADn	!ADn
1	1	X	X	QR_n	QF_n	QRn
1	↑	0	X	QR_n	QF_n	QRn+1
1	↑	1	0	SD	SD	QRn+1
1	↑	1	1	DR	DF	QRn+1
1	0	X	X	QR_n	QF_n	QF_n

3.27. DDR_OE_UNIT (Ask a Question)

The DDR_OE_UNIT macro is an output DDR cell that is only available for post-layout simulations. Every DDR_OUT instance is replaced by DDR_OE_UNIT during compile. The DDR_OE_UNIT macro consists of a DDR_OUT macro with inverted data inputs and SDR control.

Figure 3-27. DDR_OE_UNIT

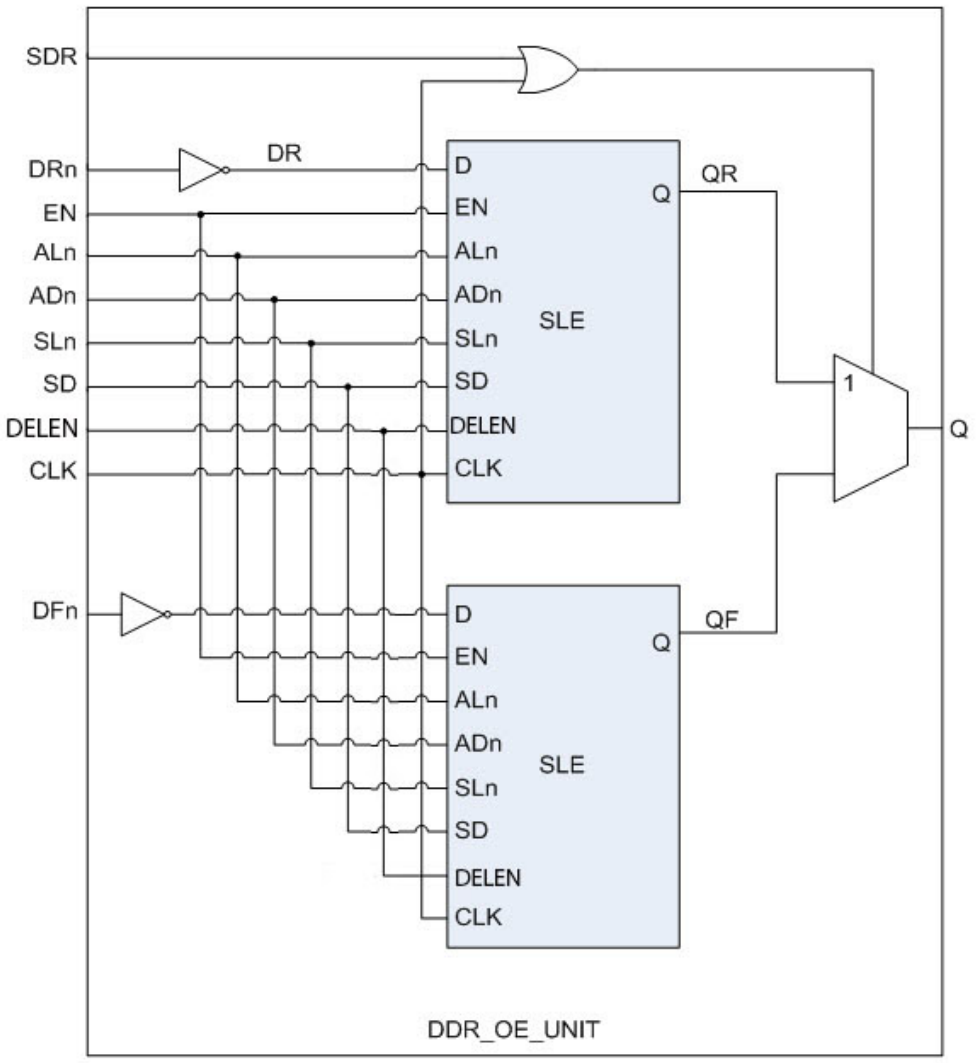


Table 3-54. DDR_OE_UNIT

Input		Output
Name	Function	
DRn	Data input (Rising Edge)	Q
DFn	Data input (Falling Edge)	
CLK	Clock input	
EN	Active High CLK enable	
ALn	Asynchronous load. This active low signal either sets the register or clears the register depending on the value of ADn.	
ADn	Static asynchronous load data. When ALn is active, Q goes to the complement of ADn.	
SLn	Synchronous load. This active low signal either sets the register or clears the register depending on the value of SD, at the rising edge of CLK.	
SD	Static synchronous load data. When SLn is active (that is, low), Q goes to the value of SD at the rising edge of CLK.	
SDR	Controls whether the cell operates in DDR (SDR = 0) or SDR (SDR = 1) modes.	

Table 3-55. Truth Table

SDR	ALn	CLK	EN	SLn	QR_{n+1}	QF_{n+1}	Qn+1
0	0	X	X	X	!ADn	!ADn	!ADn
0	1	1	X	X	QR_n	QF_n	QRn
0	1	↑	0	X	QR_n	QF_n	QRn+1
0	1	↑	1	0	SD	SD	QRn+1
0	1	↑	1	1	!DRn	!DFn	QRn+1
0	1	0	X	X	QR_n	QF_n	QF_n

3.28. IOIN_IB (Ask a Question)

Buffer macro available in post-layout netlist only.

Figure 3-28. IOIN_IB

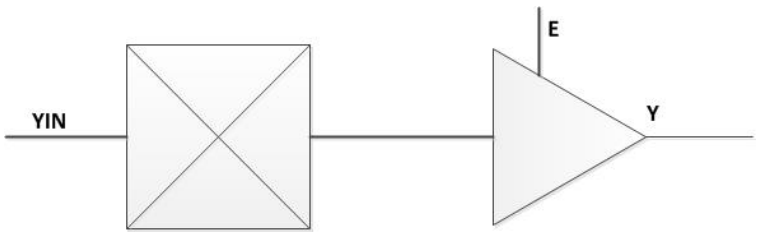


Table 3-56. IOIN_IB

Input	Output
YIN, E	Y

Note: E input is not used.

Table 3-57. Truth Table

YIN	Y
Z	X
0	0
1	1

3.29. IOPAD_IN (Ask a Question)

Input I/O macro available in post-layout netlist only.

Figure 3-29. IOPAD_IN

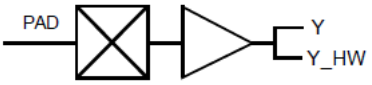


Table 3-58. IOPAD_IN

Input	Output
PAD	Y, Y_HW

Table 3-59. Truth Table

PAD	Y, Y_HW
Z	X
0	0
1	1

3.30. IOPAD_TRI (Ask a Question)

Tri-state output buffer available in post-layout netlist only.

Figure 3-30. IOPAD_TRI

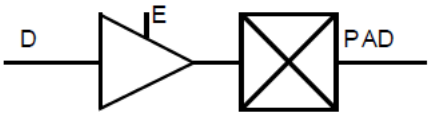


Table 3-60. IOPAD_TRI

Input	Output
D, E	PAD

Table 3-61. Truth Table

D	E	PAD
X	0	Z
0	1	0
1	1	1

3.31. IOINFF (Ask a Question)

Registered input I/O macro available only in post-layout netlist.

Figure 3-31. IOINFF

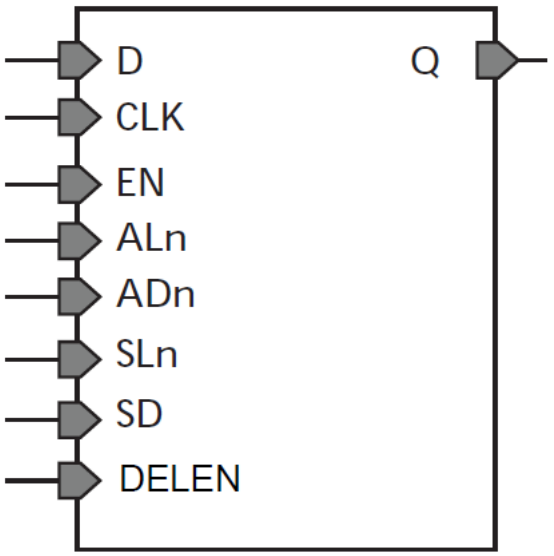


Table 3-62. IOINFF

Input		Output
Name	Function	Q
D	Data	
CLK	Clock	
EN	Enable	
ALn	Asynchronous Load (Active-Low)	
ADn[1]	Asynchronous Data (Active-Low)	
SLn	Synchronous Load (Active-Low)	
SD[1]	Synchronous Data	
DELEN[1]	Enable Single-event Transient mitigation	

Note:

1. ADn, SD, and DELEN are static signals defined at design time and need to be tied to 0 or 1.

Table 3-63. Truth Table

ALn	ADn	CLK	EN	SLn	SD	D	Qn+1
0	ADn	X	X	X	X	X	!ADn
1	X	Not rising	X	X	X	X	Qn
1	X	↑	0	X	X	X	Qn
1	X	↑	1	0	SD	X	SD
1	X	↑	1	1	X	D	D

3.32. IOOEFF (Ask a Question)

Registered output I/O macro available only in post-layout netlist. The IOOEFF is an SLE_RT with an inverted data input.

Figure 3-32. IOOEFF

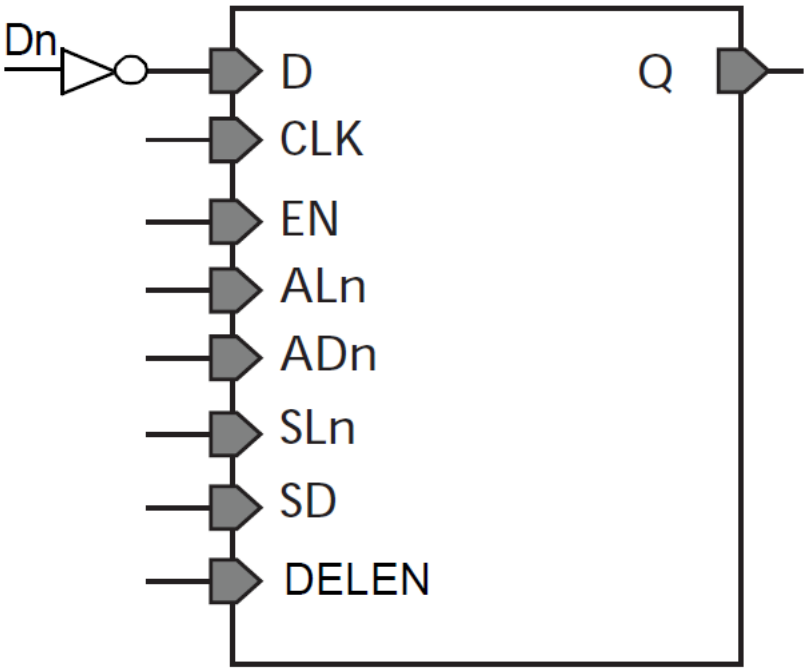


Table 3-64. IOOEFF

Input		Output
Name	Function	Q
D	Data	
CLK	Clock	
EN	Enable	
ALn	Asynchronous Load (Active Low)	
ADn[1]	Asynchronous Data (Active Low)	
SLn	Synchronous Load (Active Low)	
SD[1]	Synchronous Data	
DELEN[1]	Enable Single-event Transient mitigation	

Note:

1. ADn, SD, and DELEN are static signals defined at design time and need to be tied to 0 or 1.

Table 3-65. Truth Table

ALn	ADn	CLK	EN	SLn	SD	D	Qn+1
0	ADn	X	X	X	X	X	!ADn
1	X	Not rising	X	X	X	X	Qn
1	X	↑	0	X	X	X	Qn
1	X	↑	1	0	SD	X	SD
1	X	↑	1	1	X	D	!D

4. SRAM (Ask a Question)

4.1. RAM1K18_RT (Ask a Question)

The RAM1K18_RT block contains 24,576 (18,432 with ECC) memory bits and is a true dual-port memory with two independent data ports A and B. The RAM1K18_RT memory can also be configured in two-port mode. All read/write operations to the RAM1K18_RT memory are synchronous. To improve the read-data delay, an optional pipeline register at the output is available. RAM1K18_RT also adds a Read-enable control to both dual-port and two-port modes. The RAM1K18_RT memory has two data ports which can be independently configured in any combination as follows.

- ECC Dual-Port RAM with the following configuration:
  - 1Kx18 on both ports
- Non-ECC Dual-Port RAM with the following configurations:
  - 1Kx18 on both ports
  - 2Kx12 or 2Kx9 on both ports, but port B is read-only
  - 2Kx9 on port A, 1Kx18 on port B
- ECC Two-Port RAM with the following configurations:
  - Any of 512x36 or 1Kx18 on each port
- Non-ECC Two-Port RAM with port A write, port B read:
  - Any of 1Kx18 or 2Kx9 on each port
  - 2Kx12 on both ports
- Non-ECC Two-Port RAM with port A read, port B write:
  - Any of 512x36, 1Kx18, or 2Kx9 on each port
    FUNCTIONALITY

The main features of the RAM1K18_RT memory block are as follows:

- The address, data, block-port select, write enable and read-enable inputs are registered.
- An optional pipeline register with a separate enable and synchronous-reset is available at the read-data port to improve the clock-to-out delay.
- The registers in RAM1K18_RT block have an option to mitigate Single-event transients.
- There is an independent clock for each port. The memory will be triggered at the rising edge of the clock.
- Read from both ports at the same location is allowed.
- Read and write on the same location at the same time results in unknown data to be read. There is no collision prevention or detection. However, correct data are expected to be written into the memory.
- When ECC is enabled, each port of the RAM1K18_RT memory can raise flags to indicate single-bit-correct and double-bit-detect.

The following figure shows a simplified block diagram of the RAM1K18_RT memory block and the following table gives the port descriptions. The simplified block illustrates the two independent data ports and the read-data pipeline registers.

Figure 4-1. Simplified Block Diagram of RAM1K18_RT

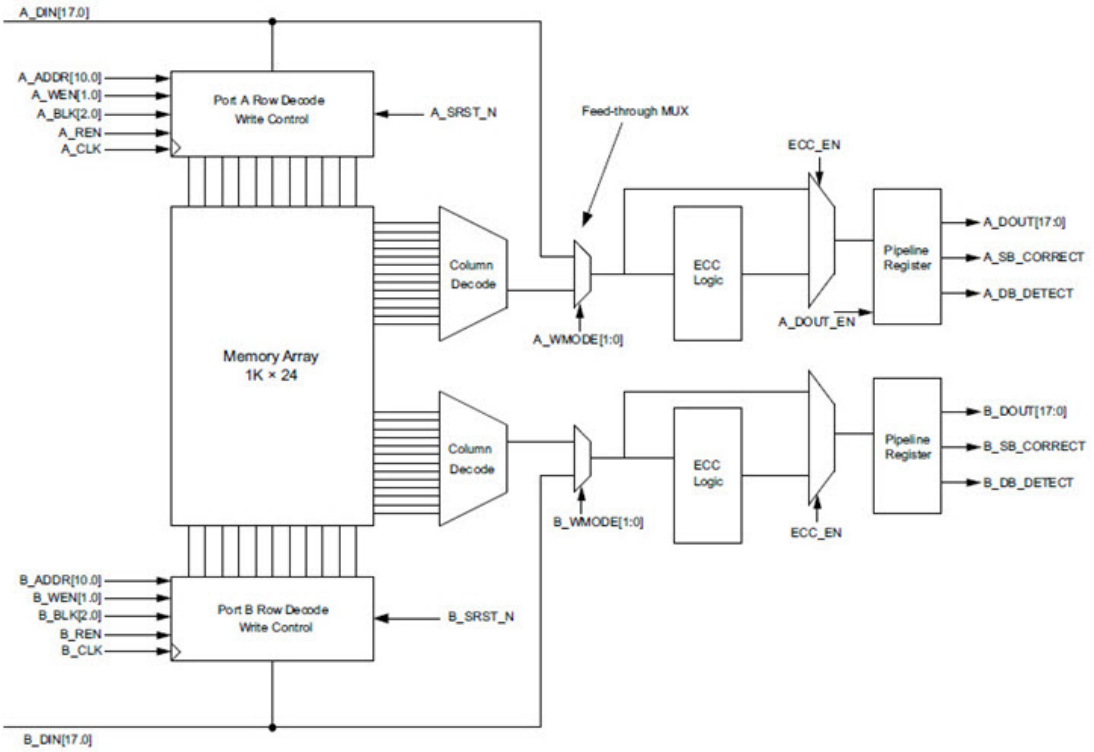


Table 4-1. Port List for RAM1K18_RT

Pin Name	Pin Direction	Type	Description	Polarity
A_ADDR[10:0]	Input	Dynamic	Port A address	—
A_BLK[2:0]	Input	Dynamic	Port A block selects	High
A_CLK	Input	Dynamic	Port A clock	Rising
A_DIN[17:0]	Input	Dynamic	Port A write-data	—
A_DOUT[17:0]	Output	Dynamic	Port A read-data	—
A_WEN[1:0]	Input	Dynamic	Port A write-enables (per byte)	High
A_REN	Input	Dynamic	Port A read-enable	High
A_WIDTH[1:0]	Input	Static	Port A width/depth mode select	—
A_DOUT_BYPASS	Input	Static	Port A pipeline register select	Low
A_WMODE[1:0]	Input	Static	Port A write mode	High
A_DOUT_EN	Input	Dynamic	Port A pipeline register enable	High
A_DOUT_SRST_N	Input	Dynamic	Port A pipeline register synchronous-reset	Low
B_ADDR[10:0]	Input	Dynamic	Port B address	—
B_BLK[2:0]	Input	Dynamic	Port B block selects	High
B_CLK	Input	Dynamic	Port B clock	Rising
B_DIN[17:0]	Input	Dynamic	Port B write-data	—
B_DOUT[17:0]	Output	Dynamic	Port B read-data	—
B_WEN[1:0]	Input	Dynamic	Port B write-enables (per byte)	High
B_REN	Input	Dynamic	Port B read-enable	High
B_WIDTH[1:0]	Input	Static	Port B width/depth mode select	—
B_WMODE[1:0]	Input	Static	Port B write mode	High
B_DOUT_BYPASS	Input	Static	Port B pipeline register select	Low

Table 4-1. Port List for RAM1K18_RT (continued)

Pin Name	Pin Direction	Type	Description	Polarity
B_DOUT_EN	Input	Dynamic	Port B pipeline register enable	High
B_DOUT_SRST_N	Input	Dynamic	Port B pipeline register synchronous-reset	Low
ARST_N	Input	Global	Pipeline registers asynchronous-reset	Low
ECC	Input	Static	Enable ECC	High
ECC_DOUT_BYPASS	Input	Static	ECC pipeline register select	Low
A_SB_CORRECT	Output	Dynamic	Port A single-bit correct flag	High
A_DB_DETECT	Output	Dynamic	Port A double-bit detect flag	High
B_SB_CORRECT	Output	Dynamic	Port B single-bit correct flag	High
B_DB_DETECT	Output	Dynamic	Port B double-bit detect flag	High
DELEN	Input	Static	Enable SET mitigation	High
SECURITY	Input	Static	Lock access to SII	High
BUSY	Output	Dynamic	Busy signal from SII	High

Note: Static inputs are defined at design time and need to be tied to 0 or 1.

Port Description

A_WIDTH AND B_WIDTH

The following table lists the width/depth mode selections for each port. Two-port mode is in effect when the width of at least one port is 36, and A_WIDTH indicates the read width while B_WIDTH indicates the write width.

Table 4-2. Width/Depth Mode Selection

Depth x Width	A_WIDTH/B_WIDTH
2Kx9, 2Kx12	00
1Kx18	01
512x36 (Two-port)	10

A_WEN AND B_WEN

The following table lists the write/read control signals for each port. Two-port mode is in effect when the width of at least one port is 36, and read operation is always enabled.

Table 4-3. Write/Read Operation Select

Depth x Width	A_WEN/B_WEN	Result
2Kx9, 2Kx12, 1Kx18	00	Perform a read operation
2Kx9, 2Kx12	11	Perform a write operation
1Kx18	01	Write [8:0]
	10	Write [17:9]
	11	Write [17:0]
512x36 (Two-port write)	B_WEN[0] = 1	Write B_DIN[8:0]
	B_WEN[1] = 1	Write B_DIN[17:9]
	A_WEN[0] = 1	Write A_DIN[8:0]
	A_WEN[1] = 1	Write A_DIN[17:9]

A_ADDR AND B_ADDR

The following table lists the address buses for the two ports. 11 bits are needed to address the 2K independent locations in x9 mode. In wider modes, fewer address bits are used. The required bits are MSB justified and unused LSB bits must be tied to 0. A_ADDR is synchronized by A_CLK while B_ADDR is synchronized to B_CLK. Two-port mode is in effect when the width of at least one port is 36, and A_ADDR provides the read-address while B_ADDR provides the write-address.

Table 4-4. Address Bus Used and Unused Bits

Depth x Width	A_ADDR/B_ADDR	
	Used Bits	Unused Bits (must be tied to 0)
2Kx9, 2Kx12	[10:0]	None
1Kx18	[10:1]	[0]
512x36 (Two-port)	[10:2]	[1:0]

A_DIN AND B_DIN

The following table lists the data input buses for the two ports. The required bits are LSB justified and unused MSB bits must be tied to 0. Two-port mode is in effect when the width of at least one port is 36, and A_DIN provides the MSB of the write-data while B_DIN provides the LSB of the write-data.

Table 4-5. Data Input Buses Used and Unused Bits

Depth x Width	A_DIN/B_DIN	
Used Bits	Unused Bits (must be tied to 0)	
2Kx9	[8:0]	[17:9]
2Kx12	[11:0]	[17:12]
1Kx18	[17:0]	None
512x36 (Two-port write)	A_DIN[17:0] is [35:18] B_DIN[17:0] is [17:0]	None

A_DOUT AND B_DOUT

The following table lists the data output buses for the two ports. The required bits are LSB justified. Two-port mode is in effect when the width of at least one port is 36, and A_DOUT provides the MSB of the read-data while B_DOUT provides the LSB of the read-data.

Table 4-6. Data Input Buses Used and Unused Bits

Depth x Width	A_DOUT/B_DOUT	
	Used Bits	Unused Bits
2Kx9	[8:0]	[17:9]
2Kx12	[11:0]	[17:12]
1Kx18	[17:0]	None
512x36 (Two-port read)	A_DOUT[17:0] is [35:18] B_DOUT[17:0] is [17:0]	None

A_BLK AND B_BLK

The following table lists the block-port select control signals for the two ports. A_BLK is synchronized by A_CLK while B_BLK is synchronized to B_CLK. Two-port mode is in effect when the width of at least one port is 36, and A_BLK controls the read operation while B_BLK controls the write operation.

Table 4-7. Block-Port Select

Block-port Select Signal	Value	Result
A_BLK[2:0]	111	Perform read or write operation on Port A. In 36 width mode, perform a read operation from both ports A and B.

Table 4-7. Block-Port Select (continued)

Block-port Select Signal	Value	Result
A_BLK[2:0]	Any one bit is 0	No operation in memory from Port A. Port A read-data will be forced to 0. In 36 width mode, the read-data from both ports A and B will be forced to 0.
B_BLK[2:0]	111	Perform read or write operation on Port B. In 36 width mode, perform a write operation to both ports A and B.
B_BLK[2:0]	Any one bit is 0	No operation in memory from Port B. Port B read-data will be forced to 0, unless it is a 36 width mode and write operation to both ports A and B is gated.

A_WMODE AND B_WMODE

Specifies the write mode for each port:

- Logic 00 = Read-data port holds the previous value.
- Logic X1 = This setting is invalid.
- Logic 10 = This setting is invalid.
  A_CLK AND B_CLK

All signals in ports A and B are synchronous to the corresponding port clock. All addresses, data, block-port select, write enable, and read-enable inputs must be setup before the rising edge of the clock. The read or write operation begins with the rising edge. Two-port mode is in effect when the width of at least one port is 36, and A_CLK provides the read clock while B_CLK provides the write clock.

A_REN AND B_REN

Enables read operation from the memory on the corresponding port.

Read-data Pipeline Register Control signals

- A_DOUT_BYPASS and B_DOUT_BYPASS
- A_DOUT_EN and B_DOUT_EN
- A_DOUT_SRST_N and B_DOUT_SRST_N

Two-port mode is in effect when the width of at least one port is 36, and the A_DOUT register signals control the MSB of the read-data while the B_DOUT register signals control the LSB of the read-data.

The following table describes the functionality of the control signals on the A_DOUT and B_DOUT pipeline registers.

Table 4-8. Truth Table for A_DOUT and B_DOUT Registers

ARST_N	_BYPASS	_CLK	_EN	_SRST_N	D	Qn+1
0	X	X	X	X	X	0
1	0	Not rising	X	X	X	Qn
1	0	↑	0	X	X	Qn
1	0	↑	1	0	X	0
1	0	↑	1	1	D	D
1	1	X	X	X	D	D

ARST_N

Connects the Read-data pipeline registers to the global Asynchronous-reset signal.

ECC AND ECC_DOUT_BYPASS

Controls ECC operation.

- ECC = 0: Disable ECC.
- ECC = 1, ECC_DOUT_BYPASS = 0: Enable ECC Pipelined.
- ECC Pipelined mode inserts an additional clock cycle to Read-data.
  - ECC = 1, ECC_DOUT_BYPASS = 1: Enable ECC Non-pipelined.
    A_SB_CORRECT AND B_SB_CORRECT

Output flag indicates single-bit correction was performed on the corresponding port.

A_DB_DETECT AND B_DB_DETECT

Output flag indicates double-bit detection was performed on the corresponding port.

DELEN

Enable Single-event Transient mitigation.

SECURITY

Controls signal, when 1 locks the entire RAM1K18_RT memory from being accessed by the SII.

BUSY

This output indicates that the RAM1K18_RT memory is being accessed by the SII.

4.2. RAM64x18_RT (Ask a Question)

The RAM64x18_RT block contains 1,536 (1,152 with ECC) memory bits and is a three-port memory providing one write port and two read ports. Write operations to the RAM64x18_RT memory are synchronous. Read operations can be asynchronous or synchronous for setting up the address and reading out the data. Enabling synchronous operation at the read-address port improves setup timing for the read-address and its enable signals. Enabling synchronous operation at the read-data port improves clock-to-out delay. Each data port on the RAM64x18_RT memory can be independently configured in any combination as follows.

- ECC Three-Port RAM with the following configuration:
  - 64x18 on all three ports
- Non-ECC Three-Port RAM with the following configurations:
  - Any of 64x18 or 128x9 on each port
  - 128x12 on all three ports
    FUNCTIONALITY

The main features of the RAM64x18_RT memory block are as follows.

- There are two independent read-data ports A and B, and one write-data port C.
- The write operation is always synchronous. The write-address, write-data, C block-port select and write enable inputs are registered.
- For both read-data ports, setting up the address can be synchronous or asynchronous.
- The two read-data ports have address registers with a separate enable and synchronous-reset for synchronous mode operation, which can also be bypassed for asynchronous mode operation.
- The two read-data ports have output registers with a separate enable and synchronous-reset for pipeline mode operation, which can also be bypassed for asynchronous mode operation.
- Therefore, there are four read operation modes for ports A and B:
  - Synchronous read-address without read-data pipeline registers (sync-async)
  - Synchronous read-address with read-data pipeline registers (sync-sync)
  - Asynchronous read-address without read-data pipeline registers (async-async)
  - Asynchronous read-address with read-data pipeline registers (async-sync)

- In ECC mode, all ports have word widths equal to 18 bits.
- In non-ECC mode, each port can be independently configured to any of the following depth/width: 64x18 or 128x9. In addition, all the ports can be configured to 128x12.
- The registers in RAM64x18_RT block have an option to mitigate Single-event transients.
- There is an independent clock for each port. The memory will be triggered at the rising edge of the clock.
- Read from both ports A and B at the same location is allowed.
- Read and write on the same location at the same time results in unknown data to be read.
- There is no collision prevention or detection. However, correct data are expected to be written into the memory.
- When ECC is enabled, each port of the RAM64x18_RT memory can raise flags to indicate single-bit-correct and double-bit-detect.

The following figure shows a simplified block diagram of the RAM64x18_RT memory block and the following table gives the port descriptions. The simplified block illustrates the three independent read/write ports and the pipeline registers on the read port.

Figure 4-2. Simplified Block Diagram of RAM64X18_RT

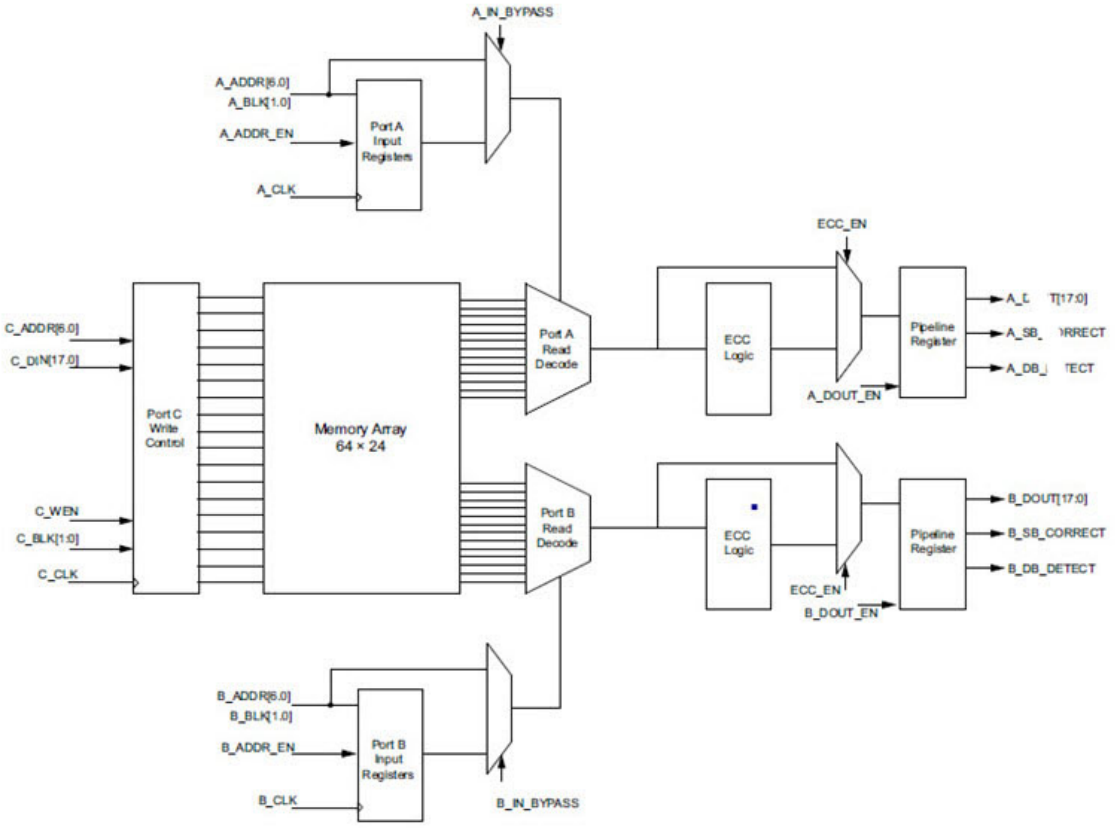


Table 4-9. Port List for RAM64X18_RT

Pin Name	Pin Direction	Type	Description	Polarity
A_ADDR[6:0]	Input	Dynamic	Port A read-address	—
A_BLK[1:0]	Input	Dynamic	Port A block selects	High
A_WIDTH	Input	Static	Port A width/depth mode selection	—
A_DOUT[17:0]	Output	Dynamic	Port A read-data	—

Table 4-9. Port List for RAM64X18_RT (continued)

Pin Name	Pin Direction	Type	Description	Polarity
A_DOUT_EN	Input	Dynamic	Port A read-data pipeline register enable	High
A_DOUT_BYPASS	Input	Static	Port A read-data pipeline register select	Low
A_DOUT_SRST_N	Input	Dynamic	Port A read-data pipeline register synchronous-reset	Low
A_CLK	Input	Dynamic	Port A registers clock	Rising
A_ADDR_EN	Input	Dynamic	Port A read-address register enable	High
A_ADDR_BYPASS	Input	Static	Port A read-address register select	Low
A_ADDR_SRST_N	Input	Dynamic	Port A read-address register synchronous-reset	Low
B_ADDR[6:0]	Input	Dynamic	Port B read-address	—
B_BLK[1:0]	Input	Dynamic	Port B block selects	High
B_WIDTH	Input	Static	Port B width/depth mode selection	—
B_DOUT[17:0]	Output	Dynamic	Port B read-data	—
B_DOUT_EN	Input	Dynamic	Port B read-data pipeline register enable	High
B_DOUT_BYPASS	Input	Static	Port B read-data pipeline register select	Low
B_DOUT_SRST_N	Input	Dynamic	Port B read-data pipeline register synchronous-reset	Low
B_CLK	Input	Dynamic	Port B registers clock	Rising
B_ADDR_EN	Input	Dynamic	Port B read-address register enable	High
B_ADDR_BYPASS	Input	Static	Port B read-address register select	Low
B_ADDR_SRST_N	Input	Dynamic	Port B read-address register synchronous-reset	Low
C_ADDR[6:0]	Input	Dynamic	Port C address	—
C_CLK	Input	Dynamic	Port C clock	Rising
C_DIN[17:0]	Input	Dynamic	Port C write-data	—
C_WEN	Input	Dynamic	Port C write enable	High
C_BLK[1:0]	Input	Dynamic	Port C block selects	High
C_WIDTH	Input	Static	Port C width/depth mode selection	—
ARST_N	Input	Global	Read-address and Read-data pipeline registers asynchronous-reset	Low
ECC	Input	Static	Enable ECC	High
ECC_DOUT_BYPASS	Input	Static	ECC pipeline register select	Low
A_SB_CORRECT	Output	Dynamic	Port A single-bit correct flag	High
A_DB_DETECT	Output	Dynamic	Port A double-bit detect flag	High
B_SB_CORRECT	Output	Dynamic	Port B single-bit correct flag	High
B_DB_DETECT	Output	Dynamic	Port B double-bit detect flag	High
DELEN	Input	Static	Enable SET mitigation	High
SECURITY	Input	Static	Lock access to SII	High
BUSY	Output	Dynamic	Busy signal from SII	High

Tip: Static inputs are defined at design time and need to be tied to 0 or 1.

PORT DESCRIPTION

A_WIDTH, B_WIDTH AND C_WIDTH

The following table lists the width/depth mode selections for each port.

Table 4-10. Width/Depth Mode Selection

Depth x Width	A_WIDTH/B_WIDTH/C_WIDTH
128x9, 128x12	0
64x16, 64x18	1

C_WEN

This is the write enable signal for port C.

A_ADDR, B_ADDR AND C_ADDR

The following table lists the address buses for each port. 7 bits are required to address 128 independent locations in x9 mode. In wider modes, fewer address bits are used. The required bits are MSB justified and unused LSB bits must be tied to 0.

Table 4-11. Address Buses Used and Unused Bits

Depth x Width	A_ADDR/B_ADDR/C_ADDR	
	Used Bits	Unused Bits (must be tied to zero)
128x9, 128x12	[6:0]	None
64x18	[6:1]	[0]

C_DIN

The following table lists the write-data input for port C. The required bits are LSB justified and unused MSB bits must be tied to 0.

Table 4-12. Data Input Bus Used and Unused Bits

Depth x Width	C_DIN	
	Used Bits	Unused Bits (must be tied to 0)
128x9	[8:0]	[17:9]
128x12	[11:0]	[17:12]
64x18	[17:0]	None

A_DOUT AND B_DOUT

The following table lists the read-data output buses for ports A and B. The required bits are LSB justified.

Table 4-13. Data Output Used and Unused Bits

Depth x Width	A_DOUT/B_DOUT	
	Used Bits	Unused Bits
128x9	[8:0]	[17:9]
128x12	[11:0]	[17:12]
64x18	[17:0]	None

A_BLK, B_BLK AND C_BLK

The following table lists the block-port select control signals for the ports.

Table 4-14. Block-Port Select

Block-port Select Signal	Value	Result
A_BLK[1:0]	Any one bit is 0	Port A is not selected and its read-data will be forced to zero.
	11	Perform read operation from port A.

Table 4-14. Block-Port Select (continued)

Block-port Select Signal	Value	Result
B_BLK[1:0]	Any one bit is 0	Port B is not selected and its read-data will be forced to zero.
	11	Perform read operation from port B.
C_BLK[1:0]	Any one bit is 0	Port C is not selected.
	11	Perform write operation to port C.

C_CLK

All signals on port C are synchronous to this clock signal. All write-address, write-data, C block-port select and write-enable inputs must be set up before the rising edge of the clock. The write operation begins with the rising edge.

Read-address and Read-data Pipeline Register Control signals

- A_DOUT_BYPASS, A_ADDR_BYPASS, B_DOUT_BYPASS, and B_ADDR_BYPASS
- A_DOUT_EN, A_ADDR_EN, B_DOUT_EN, and B_ADDR_EN
- A_DOUT_SRST_N, A_ADDR_SRST_N, B_DOUT_SRST_N, and B_ADDR_SRST_N

The following table describes the functionality of the control signals on the A_ADDR, B_ADDR, A_DOUT, and B_DOUT registers.

Table 4-15. Truth Table for A_ADDR, B_ADDR, A_DOUT, and B_DOUT Registers

ARST_N	_BYPASS	_CLK	_EN	_SRST_N	D	Qn+1
0	X	X	X	X	X	0
1	0	Not rising	X	X	X	Qn
1	0	↑	0	X	X	Qn
1	0	↑	1	0	X	0
1	0	↑	1	1	D	D
1	1	X	X	X	D	D

ARST_N

Connects the read-address and read-data pipeline registers to the global Asynchronous-reset signal.

ECC and ECC_DOUT_BYPASS

Controls ECC operation.

- ECC = 0: Disable ECC.
- ECC = 1, ECC_DOUT_BYPASS = 0: Enable ECC Pipelined.
  - ECC Pipelined mode inserts an additional clock cycle to Read-data.
- ECC = 1, ECC_DOUT_BYPASS = 1: Enable ECC Non-pipelined.
  A_SB_CORRECT AND B_SB_CORRECT

Output flag indicates single-bit correction was performed on the corresponding port.

A_DB_DETECT and B_DB_DETECT

Output flag indicates double-bit detection was performed on the corresponding port.

DELEN

Enable Single-event Transient mitigation.

SECURITY

Control signal, when 1 locks the entire RAM64x18_RT memory from being accessed by the SII.

BUSY

This output indicates that the RAM64x18_RT memory is being accessed by the SII.

5. MACC (Ask a Question)

5.1. MACC (Ask a Question)

18-bit x 18-bit multiply-accumulate MACC block. The MACC block can accumulate the current multiplication product with a previous result, a constant, a dynamic value, or a result from another MACC block. Each MACC block can also be configured to perform a Dot-product operation. All the signals of the MACC block (except CDIN and CDOUT) have optional registers.

Figure 5-1. MACC Port

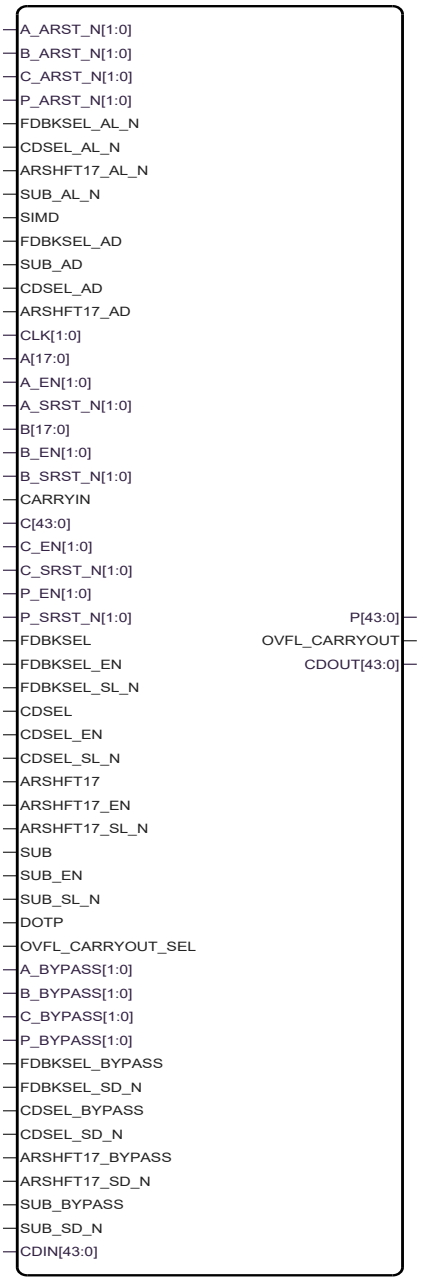


Table 5-1. Ports

Port Name	Direction	Type	Polarity	Description
DOTP	Input	Static	High	Dot-product mode. • When DOTP = 1, MACC block performs Dot-product of two pairs of 9-bit operands. • When DOTP = 0, it is called the normal mode.
SIMD	Input	Static	—	Reserved. Must be 0.
OVFL_CARRYOUT_SEL	Input	Static	High	Generate OVERFLOW or CARRYOUT with result P. OVERFLOW when OVFL_CARRYOUT_SEL = 0 CARRYOUT when OVFL_CARRYOUT_SEL = 1
CLK[1:0]	Input	Dynamic	Rising edge	Input clocks. CLK[1] is the clock for A[17:9], B[17:9], C[43:18], P[43:18], OVFL_CARRYOUT, ARSHFT17, CDSEL, FDBKSEL and SUB registers. CLK[0] is the clock for A[8:0], B[8:0], C[17:0], CARRYIN and P[17:0]. In normal mode, ensure CLK[1] = CLK[0].
A[17:0]	Input	Dynamic	High	Input data A.
A_BYPASS[1:0]	Input	Static	High	Bypass data A registers. A_BYPASS[1] is for A[17:9]. Connect to 1, if not registered. A_BYPASS[0] is for A[8:0]. Connect to 1, if not registered. In normal mode, ensure A_BYPASS[0] = A_BYPASS[1].
A_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for data A registers. Connect both A_ARST_N[1] and = A_ARST_N[0] to 1 or to the global Asynchronous reset of the design
A_SRST_N[1:0]	Input	Dynamic	Low	Synchronous reset for data A registers. A_SRST_N[1] is for A[17:9]. Connect to 1, if not registered. A_SRST_N[0] is for A[8:0]. Connect to 1, if not registered. In normal mode, ensure A_SRST_N[1] = A_SRST_N[0].
A_EN[1:0]	Input	Dynamic	High	Enable for data A registers. A_EN[1] is for A[17:9]. Connect to 1, if not registered. A_EN[0] is for A[8:0]. Connect to 1, if not registered. In normal mode, ensure A_EN[1] = A_EN[0].
B[17:0]	Input	Dynamic	High	Input data B.
B_BYPASS[1:0]	Input	Static	High	Bypass data B registers. B_BYPASS[1] is for B[17:9]. Connect to 1, if not registered. B_BYPASS[0] is for B[8:0]. Connect to 1, if not registered. In normal mode, ensure B_BYPASS[0] = B_BYPASS[1].
B_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for data B registers. In normal mode, ensure Connect both B_ARST_N[1] and B_ARST_N[0] to 1 or to the global Asynchronous reset of the design.

Table 5-1. Ports (continued)

Port Name	Direction	Type	Polarity	Description
B_SRST_N[1:0]	Input	Dynamic	Low	Synchronous reset for data B registers. B_SRST_N[1] is for B[17:9]. Connect to 1, if not registered. B_SRST_N[0] is for B[8:0]. Connect to 1, if not registered. In normal mode, ensure B_SRST_N[1] = B_SRST_N[0].
B_EN[1:0]	Input	Dynamic	High	Enable for data B registers. B_EN[1] is for B[17:9]. Connect to 1, if not registered. B_EN[0] is for B[8:0]. Connect to 1, if not registered. In normal mode, ensure B_EN[1] = B_EN[0].
P[43:0]	Output	—	High	Result data. Normal mode P = D + (CARRYIN + C) + (A * B), when SUB = 0 P = D + (CARRYIN + C) - (A * B), when SUB = 1 Dot-product mode P = D + (CARRYIN + C) + 512 * ((A_L * B_H) + (A_H * B_L)), when SUB = 0 P = D + (CARRYIN + C) - 512 * ((A_L * B_H) + (A_H * B_L)), when SUB = 1 Notation: A_L = A[8:0], A_H = A[17:9] B_L = B[8:0], B_H = B[17:9] Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
OVFL_CARRYOUT	Output	—	High	Overflow or CarryOut Refer to Table 5-5.
P_BYPASS[1:0]	Input	Static	High	Bypass result P registers. P_BYPASS[1] is for P[43:18] and OVFL_CARRYOUT. Connect to 1, if not registered. P_BYPASS[0] is for P[17:0]. Connect to 1, if not registered. In normal mode, ensure P_BYPASS[0] = P_BYPASS[1].
P_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for P and OVFL_CARRYOUT registers. Connect both P_ARST_N[1] and P_ARST_N[0] to 1 or to the global Asynchronous reset of the design.
P_SRST_N[1:0]	Input	Dynamic	Low	Synchronous reset for result P registers. P_SRST_N[1] is for P[43:18] and OVFL_CARRYOUT. Connect to 1, if not registered. P_SRST_N[0] is for P[17:0]. Connect to 1, if not registered. In normal mode, ensure P_SRST_N[1] = P_SRST_N[0].
P_EN[1:0]	Input	Dynamic	High	Enable for result P registers. P_EN[1] is for P[43:18] and OVFL_CARRYOUT. Connect to 1, if not registered. P_EN[0] is for P[17:0]. Connect to 1, if not registered. In normal mode, ensure P_EN[1] = P_EN[0].

Table 5-1. Ports (continued)

Port Name	Direction	Type	Polarity	Description
CDOUT[43:0]	Output	Cascade	High	Cascade output of result P. CDOUT is the same as P. The entire bus must either be dangling or drive an entire CDIN of another MACC block in cascaded mode.
CARRYIN	Input	Dynamic	High	CarryIn for operand C.
C[43:0]	Input	Dynamic	High	Routed input for operand C. In Dot-product mode, connect C[8:0] to the CARRYIN.
C_BYPASS[1:0]	Input	Static	High	Bypass data C registers. C_BYPASS[1] is for C[43:18]. Connect to 1, if not registered. C_BYPASS[0] is for C[17:0] and CARRYIN. Connect to 1, if not registered. In normal mode, ensure C_BYPASS[0] = C_BYPASS[1].
C_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for CARRYIN and C registers. Connect both C_ARST_N[1] and C_ARST_N[0] to 1 or to the global Asynchronous reset of the design.
C_SRST_N[1:0]	Input	Dynamic	Low	Synchronous reset for data C registers. C_SRST_N[1] is for C[43:18]. Connect to 1, if not registered. C_SRST_N[0] is for C[17:0] and CARRYIN. Connect to 1, if not registered. In normal mode, ensure C_SRST_N[1] = C_SRST_N[0].
C_EN[1:0]	Input	Dynamic	High	Enable for data C registers. C_EN[1] is for C[43:18]. Connect to 1, if not registered. C_EN[0] is for C[17:0] and CARRYIN. Connect to 1, if not registered. In normal mode, ensure C_EN[1] = C_EN[0].
CDIN[43:0]	Input	Cascade	High	Cascaded input for operand D. The entire bus must be driven by an entire CDOUT of another MACC block. In Dot-product mode the CDOUT must also be generated by a MACC block in Dot-product mode. Refer to Table 5-4 to see how CDIN is propagated to operand D.
ARSHFT17	Input	Dynamic	High	Arithmetic right-shift for operand D. When asserted, a 17-bit arithmetic right-shift is performed on operand D going into the accumulator. Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
ARSHFT17_BYPASS	Input	Static	High	Bypass ARSHFT17 register. Connect to 1, if not registered.
ARSHFT17_AL_N	Input	Dynamic	Low	Asynchronous load for ARSHFT17 register. Connect to 1 or to the global Asynchronous reset of the design. When asserted, ARSHFT17 register is loaded with ARSHFT17_AD.
ARSHFT17_AD	Input	Static	High	Asynchronous load data for ARSHFT17 register.
ARSHFT17_SL_N	Input	Dynamic	Low	Synchronous load for ARSHFT17 register. Connect to 1, if not registered. See Table 5-2.

Table 5-1. Ports (continued)

Port Name	Direction	Type	Polarity	Description
ARSHFT17_SD_N	Input	Static	Low	Synchronous load data for ARSHFT17 register. See Table 5-2.
ARSHFT17_EN	Input	Dynamic	High	Enable for ARSHFT17 register. Connect to 1, if not registered. See Table 5-2.
CDSEL	Input	Dynamic	High	Select CDIN for operand D. When CDSEL = 1, propagate CDIN. When CDSEL = 0, propagate 0 or P depending on FDBKSEL. Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
CDSEL_BYPASS	Input	Static	High	Bypass CDSEL register. Connect to 1, if not registered.
CDSEL_AL_N	Input	Dynamic	Low	Asynchronous load for CDSEL register. Connect to 1 or to the global Asynchronous reset of the design. When asserted, CDSEL register is loaded with CDSEL_AD.
CDSEL_AD	Input	Static	High	Asynchronous load data for CDSEL register.
CDSEL_SL_N	Input	Dynamic	Low	Synchronous load for CDSEL register. Connect to 1, if not registered. See Table 5-2.
CDSEL_SD_N	Input	Static	Low	Synchronous load data for CDSEL register. See Table 5-2.
CDSEL_EN	Input	Dynamic	High	Enable for CDSEL register. Connect to 1, if not registered. See Table 5-2.
FDBKSEL	Input	Dynamic	High	Select the feedback from P for operand D. When FDBKSEL = 1, propagate the current value of result P register. Ensure P_BYPASS[1] = 0 and CDSEL = 0. When FDBKSEL = 0, propagate 0. Ensure CDSEL = 0. Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
FDBKSEL_BYPASS	Input	Static	High	Bypass FDBKSEL register. Connect to 1, if not registered.
FDBKSEL_AL_N	Input	Dynamic	Low	Asynchronous load for FDBKSEL register. Connect to 1 or to the global Asynchronous reset of the design. When asserted, FDBKSEL register is loaded with FDBKSEL_AD.
FDBKSEL_AD	Input	Static	High	Asynchronous load data for FDBKSEL register.
FDBKSEL_SL_N	Input	Dynamic	Low	Synchronous load for FDBKSEL register. Connect to 1, if not registered. See Table 5-2.
FDBKSEL_SD_N	Input	Static	Low	Synchronous load data for FDBKSEL register. See Table 5-2.
FDBKSEL_EN	Input	Dynamic	High	Enable for FDBKSEL register. Connect to 1, if not registered. See Table 5-2.
SUB	Input	Dynamic	High	Subtract operation.
SUB_BYPASS	Input	Static	High	Bypass SUB register. Connect to 1, if not registered.
SUB_AL_N	Input	Dynamic	Low	Asynchronous load for SUB register. Connect to 1 or to the global Asynchronous reset of the design. When asserted, SUB register is loaded with SUB_AD.
SUB_AD	Input	Static	High	Asynchronous load data for SUB register.
SUB_SL_N	Input	Dynamic	Low	Synchronous load for SUB register. Connect to 1, if not registered. See Table 5-2.

Table 5-1. Ports (continued)

Port Name	Direction	Type	Polarity	Description
SUB_SD_N	Input	Static	Low	Synchronous load data for SUB register. See Table 5-2.
SUB_EN	Input	Dynamic	High	Enable for SUB register. Connect to 1, if not registered. See Table 5-2.

Table 5-2. Truth Table for Control Registers ARSHFT17, CDSEL, FDBKSEL, and SUB

_AL_N	_AD	_BYPASS	_CLK	_EN	_SL_N	_SD_N	D	Qn+1
0	AD	X	X	X	X	X	X	AD
1	X	0	Not rising	X	X	X	X	Qn
1	X	0	—	0	X	X	X	Qn
1	X	0	—	1	0	SDn	X	!SDn
1	X	0	—	1	1	X	D	D
1	X	1	X	0	X	X	X	Qn
1	X	1	X	1	0	SDn	X	!SDn
1	X	1	X	1	1	X	D	D

Table 5-3. Truth Table—Data Registers A, B, C, CARRYIN, P, and OVFL_CARRYOUT

_ARST_N	_BYPASS	_CLK	_EN	_SRST_N	D	Qn+1
0	X	X	X	X	X	0
1	0	Not rising	X	X	X	Qn
1	0	—	0	X	X	Qn
1	0	—	1	0	X	0
1	0	—	1	1	D	D
1	1	X	0	X	X	Qn
1	1	X	1	0	X	0
1	1	X	1	1	D	D

Table 5-4. Truth Table—Propagating Data to Operand D

FDBKSEL	CDSEL	ARSHFT17	Operand D
0	0	x	44'b0
x	1	0	CDIN[43:0]
x	1	1	{{17{CDIN[43]}},CDIN[43:17]}
1	0	0	P[43:0]
1	0	1	{{17{P[43]}},P[43:17]}

Table 5-5. Truth Table—Computation of OVFL_CARRYOUT

OVFL_CARRYOUT_SEL	OVFL_CARRYOUT	Description
0	(SUM[45] ^ SUM[44) | (SUM[44] ^ SUM[43])	True if overflow or underflow occurred.
1	C[43] ^ D[43] ^ SUM[44]	A signal that can be used to extend the final adder in the fabric.

SUM[45:0] is defined similarly to P[43:0], except that SUM is a 46-bit quantity so that no overflow can occur. SUM[44] is the carry out bit of a 44-bit final adder producing P[43:0].

5.2. MACC_RT (Ask a Question)

18 bit x 18 bit multiply-accumulate MACC_RT block.

The MACC_RT block can accumulate the current multiplication product with a previous result, a constant, a dynamic value, or a result from another MACC_RT block. Each MACC_RT block can also

be configured to perform a Dot-product operation. All the signals of the MACC_RT block (except CDIN and CDOUT) have optional registers.

Figure 5-2. Ports

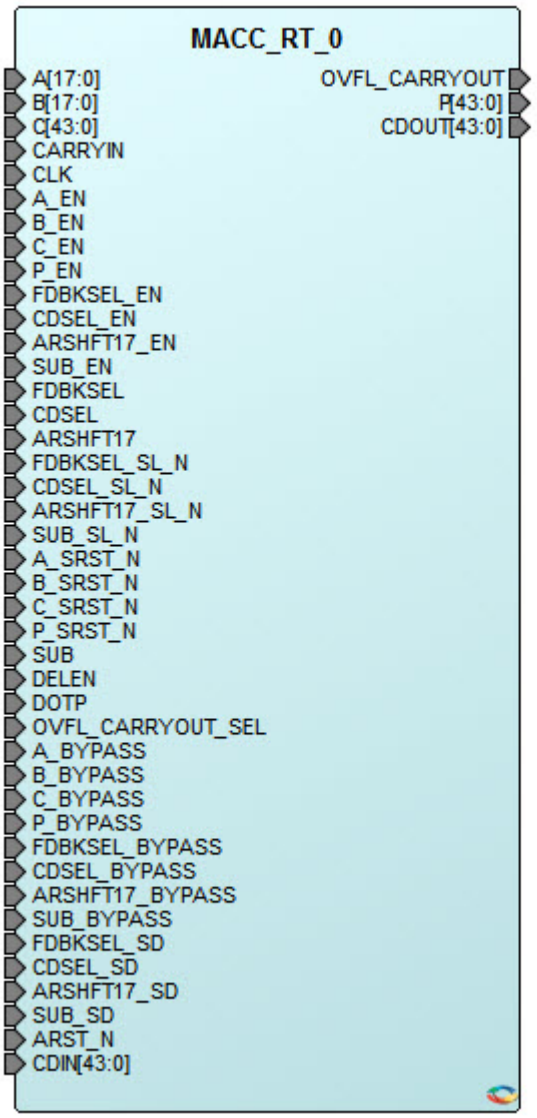


Table 5-6. Ports

Port Name	Direction	Type	Polarity	Description
DOTP	Input	Static	High	Dot-product mode. When DOTP = 1, MACC_RT block performs Dot-product of two pairs of 9-bit operands. When DOTP = 0, it is called the normal mode.
OVFL_CARRYOUT_SEL	Input	Static	High	Generate OVERFLOW or CARRYOUT with result P. OVERFLOW when OVFL_CARRYOUT_SEL = 0 CARRYOUT when OVFL_CARRYOUT_SEL = 1
DELEN	Input	Static	High	Enable Single-event Transient mitigation

Table 5-6. Ports (continued)

Port Name	Direction	Type	Polarity	Description
CLK	Input	Dynamic	Rising edge	Input clocks. CLK is the clock for A[17:0], B[17:0], C[43:0], P[43:0], OVFL_CARRYOUT, ARSHFT17, CDSEL, FDBKSEL and SUB registers.
ARST_N	Input	Dynamic	Low	Asynchronous reset for all registers
A[17:0]	Input	Dynamic	High	Input data A.
A_BYPASS	Input	Static	High	Bypass data A registers. Connect to 1, if not registered.
A_SRST_N	Input	Dynamic	Low	Synchronous reset for data A registers. Connect to 1, if not registered.
A_EN	Input	Dynamic	High	Enable for data A registers. Connect to 1, if not registered.
B[17:0]	Input	Dynamic	High	Input data B.
B_BYPASS	Input	Static	High	Bypass data B registers. Connect to 1, if not registered.
B_SRST_N	Input	Dynamic	Low	Synchronous reset for data B registers. Connect to 1, if not registered.
B_EN	Input	Dynamic	High	Enable for data B registers. Connect to 1, if not registered.
P[43:0]	Output	—	High	Result data. Normal mode P = D + (CARRYIN + C) + (A * B), when SUB = 0 P = D + (CARRYIN + C) - (A * B), when SUB = 1 Dot-product mode P = D + (CARRYIN + C) + 512 * ((A_L * B_H) + (A_H * B_L)), when SUB = 0 P = D + (CARRYIN + C) - 512 * ((A_L * B_H) + (A_H * B_L)), when SUB = 1 Notation: A_L = A[8:0], A_H = A[17:9] B_L = B[8:0], B_H = B[17:9] Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
OVFL_CARRYOUT	Output		High	Overflow or CarryOut Refer to Table 5-5.
P_BYPASS	Input	Static	High	Bypass P and OVFL_CARRYOUT registers. Connect to 1, if not registered.
P_SRST_N	Input	Dynamic	Low	Synchronous reset for P and OVFL_CARRYOUT registers. Connect to 1, if not registered.
P_EN	Input	Dynamic	High	Enable for P and OVFL_CARRYOUT registers. Connect to 1, if not registered.
CDOUT[43:0]	Output	Cascade	High	Cascade output of result P. CDOUT is the same as P. The entire bus must either be dangling or drive an entire CDIN of another MACC_RT block in cascaded mode.
CARRYIN	Input	Dynamic	High	CarryIn for operand C.
C[43:0]	Input	Dynamic	High	Routed input for operand C. In Dot-product mode, connect C[8:0] to the CARRYIN.

Table 5-6. Ports (continued)

Port Name	Direction	Type	Polarity	Description
C_BYPASS	Input	Static	High	Bypass CARRYIN and C registers. Connect to 1, if not registered.
C_SRST_N	Input	Dynamic	Low	Synchronous reset for CARRYIN and C registers. Connect to 1, if not registered.
C_EN	Input	Dynamic	High	Enable for CARRYIN and C registers. Connect to 1, if not registered.
CDIN[43:0]	Input	Cascade	High	Cascaded input for operand D. The entire bus must be driven by an entire CDOUT of another MACC_RT block. In Dot-product mode the CDOUT must also be generated by a MACC_RT block in Dot-product mode. Refer to Table 5-4 to see how CDIN is propagated to operand D.
ARSHFT17	Input	Dynamic	High	Arithmetic right-shift for operand D. When asserted, a 17-bit arithmetic right-shift is performed on operand D going into the accumulator. Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
ARSHFT17_BYPASS	Input	Static	High	Bypass ARSHFT17 register. Connect to 1, if not registered.
ARSHFT17_SL_N	Input	Dynamic	Low	Synchronous load for ARSHFT17 register. Connect to 1, if not registered. See Table 5-7.
ARSHFT17_SD	Input	Static	High	Synchronous load data for ARSHFT17 register. See Table 5-7.
ARSHFT17_EN	Input	Dynamic	High	Enable for ARSHFT17 register. Connect to 1, if not registered. See Table 5-7.
CDSEL	Input	Dynamic	High	Select CDIN for operand D. When CDSEL = 1, propagate CDIN. When CDSEL = 0, propagate 0 or P depending on FDBKSEL. Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
CDSEL_BYPASS	Input	Static	High	Bypass CDSEL register. Connect to 1, if not registered.
CDSEL_SL_N	Input	Dynamic	Low	Synchronous load for CDSEL register. Connect to 1, if not registered. See Table 5-7.
CDSEL_SD	Input	Static	High	Synchronous load data for CDSEL register. See Table 5-7.
CDSEL_EN	Input	Dynamic	High	Enable for CDSEL register. Connect to 1, if not registered. See Table 5-7.
FDBKSEL	Input	Dynamic	High	Select the feedback from P for operand D. When FDBKSEL = 1, propagate the current value of result P register. Ensure P_BYPASS = 0 and CDSEL = 0. When FDBKSEL = 0, propagate 0. Ensure CDSEL = 0. Refer to Table 5-4 to see how operand D is obtained from P, CDIN or 0.
FDBKSEL_BYPASS	Input	Static	High	Bypass FDBKSEL register. Connect to 1, if not registered.
FDBKSEL_SL_N	Input	Dynamic	Low	Synchronous load for FDBKSEL register. Connect to 1, if not registered. See Table 5-7.

Table 5-6. Ports (continued)

Port Name	Direction	Type	Polarity	Description
FDBKSEL_SD	Input	Static	High	Synchronous load data for FDBKSEL register. See Table 5-7.
FDBKSEL_EN	Input	Dynamic	High	Enable for FDBKSEL register. Connect to 1, if not registered. See Table 5-7.
SUB	Input	Dynamic	High	Subtract operation.
SUB_BYPASS	Input	Static	High	Bypass SUB register. Connect to 1, if not registered.
SUB_SL_N	Input	Dynamic	Low	Synchronous load for SUB register. Connect to 1, if not registered. See Table 5-7.
SUB_SD	Input	Static	High	Synchronous load data for SUB register. See Table 5-7.
SUB_EN	Input	Dynamic	High	Enable for SUB register. Connect to 1, if not registered. See Table 5-7.

Table 5-7. Truth Table for Control Registers ARSHFT17, CDSEL, FDBKSEL, and SUB

ARST_N	_BYPASS	_CLK	_EN	_SL_N	_SD	D	Qn+1
0	X	X	X	X	X	X	0
1	0	Not rising	X	X	X	X	Qn
1	0	—	0	X	X	X	Qn
1	0	—	1	0	SDn	X	SDn
1	0	—	1	1	X	D	D
1	1	X	0	X	X	X	Qn
1	1	X	1	0	SDn	X	SDn
1	1	X	1	1	X	D	D

6. Additional References (Ask a Question)

For more information on DDR and SerDes, see the following.

Table 6-1. Additional References

Core Name	Link
DDR	UG0573 RTG4 FPGA DDR Memory Controller User Guide RTG4 DDR Memory Controller Configuration User Guide RTG4 DDR Memory Controller with Initialization Configuration User Guide
SerDes	UG0567 RTG4 FPGA High-Speed Serial Interfaces User Guide RTG4 High Speed Serial Interface Configuration User Guide
AMBAIF_SRAM	RTG4_SRAM_AHBL_AXI User Guide
CCC CCCAPB	RTG4 FPGA Clock Conditioning Circuit with PLL Configuration User Guide
DPRAM	RTG4 FPGA Dual-Port Large SRAM Configuration User Guide
FDDR	RTG4 DDR Memory Controller Configuration User Guide
FDDR_INIT	RTG4 DDR Memory Controller with Initialization Configuration User Guide
NPSS_SERDES_IF NPSS_SERDES_IF_INIT PCIE_SERDES_IF PCIE_SERDES_IF_INIT	RTG4 High Speed Serial Interface Configuration User Guide
RTG4FCCC_ELOCK	RTG4 FCCC with Enhanced PLL Calibration Configurator User Guide
RTG4FCCC_ELOCK	RTG4 FCCC with Enhanced PLL Calibration Configurator Release Notes
TPRAM	RTG4 FPGA Two-Port Large SRAM Configuration User Guide
UPROM	RTG4 µPROM Configuration User Guide
URAM	Micro SRAM Configuration User Guide

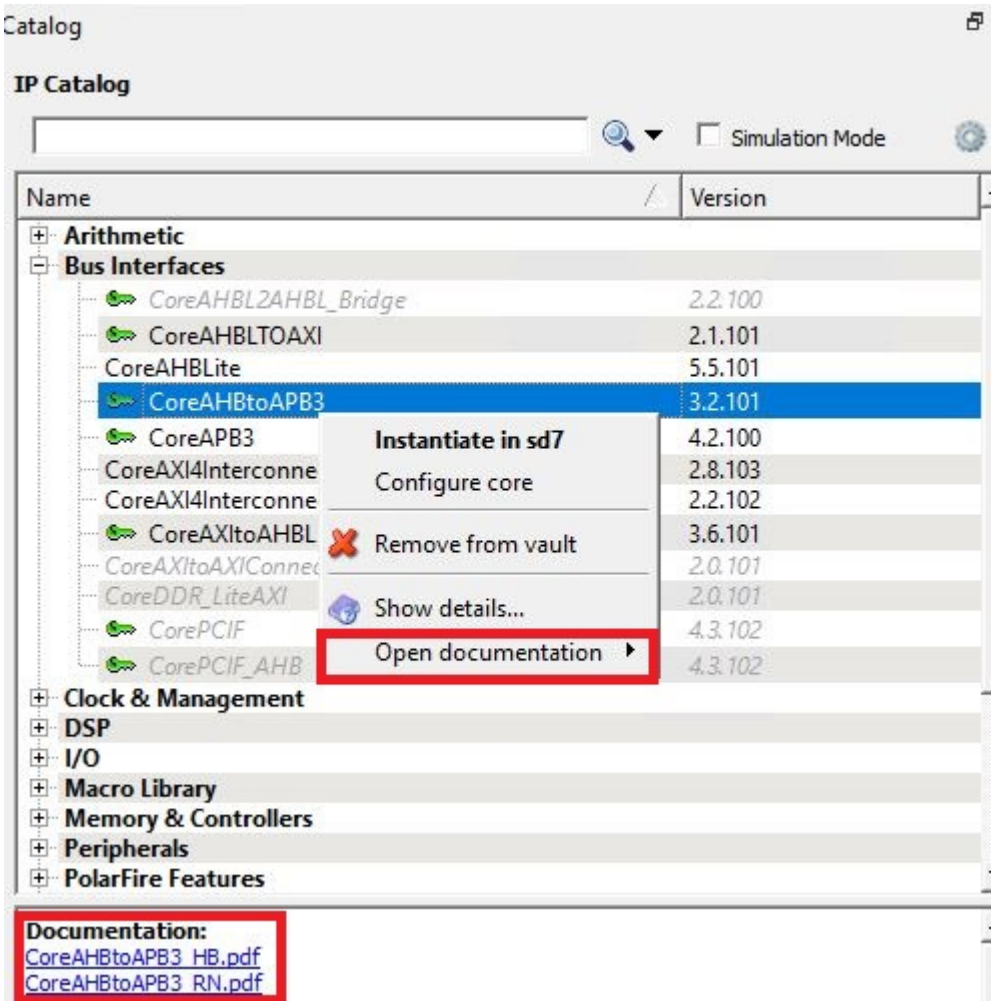
7. Appendix-Accessing IP Core Documentation (Ask a Question)

The IP cores are integrated with the Libero® SoC Design Suite. The use of IP cores not only shortens the design cycle time but also provides proven and reliable design components for re-use in multiple applications. The IP Catalog in Libero provides an interface to access and manage the entire set of IP cores and their documentation.

To access the IP Core documentation, follow the steps:

1. In Libero SoC, click the **Catalog** tab and select an IP core. The documentation associated with the IP core are listed under the **Documentation** view on the **Catalog** tab.
2. Alternatively, right click on any IP core in the IP **Catalog**.
3. Select **Open Documentation** from the context menu that appears. A list of the various documents available for the selected core appears.
4. Click on the required document to open it.

Figure 7-1. Accessing the IP Core Documentation



8. Revision History (Ask a Question)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
N	05/2025	This document is released with Libero SoC Design Suite v2025.1 without changes from v2024.2.
M	09/2024	The following change is made in this revision: • Updated SYSCTRL_RESET_STATUS.
L	08/2024	This document is released with Libero SoC Design Suite v2024.2 without changes from v2024.1.
K	02/2024	This document is released with Libero SoC Design Suite v2024.1 without changes from v2023.2.
J	08/2023	This document is released with Libero SoC Design Suite v2023.2 without changes from v2023.1.
H	04/2023	This document is released with Libero SoC Design Suite v2023.1 without changes from v2022.3.
G	12/2022	This document is released with Libero SoC Design Suite v2022.3 without changes from v2022.2.
F	08/2022	This document is released with Libero SoC Design Suite v2022.2 without changes from v2022.1.
E	04/2022	This document is released with Libero SoC Design Suite v2022.1 without changes from v2021.3.
D	12/2021	Fixed the IOINFF_BYPASS truth table.
C	06/2021	Editorial updates only. No technical content updates.
B	04/2021	Editorial updates only. No technical content updates.
A	11/2020	• Migrated the document from Microsemi to Microchip format. • Formatted this document per Microchip's standards.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-1096-7

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.