Macro Library User Guide for SmartFusion 2 and IGLOO 2 FPGAs

Libero SoC v2025.1

Introduction (Ask a Question)

This macro library guide supports the SmartFusion® 2 FPGA and IGLOO® 2 FPGA families. 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
- DL—D-type latch

The next mandatory character indicates the output polarity:

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

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

- E0—Active-Low enable (bubbled)
- E1—Active-high enable (non-bubbled)

The next 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)

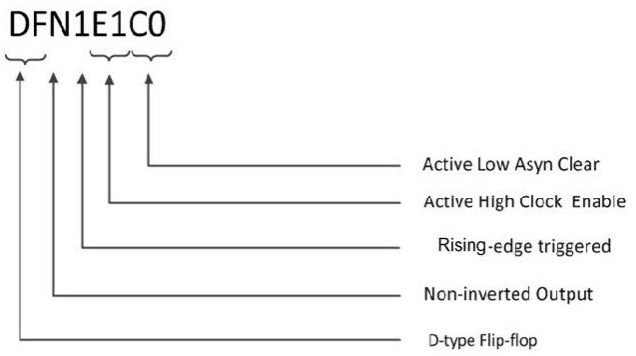
The next two optional characters indicate the polarity of the asynchronous Clear pin, if present:

- 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 PolarFire® FPGA 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 table.

Figure 1. Naming Convention



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

State	Meaning
0	Logic 0
1	Logic 1
X	Do not Care (for Inputs), Unknown (for Outputs)
Z	High Impedance

User Parameter/Generics

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. Combinatorial Logic (Ask a Question)

1.1. AND2 (Ask a Question)

2-Input AND.

Figure 1-1. AND2

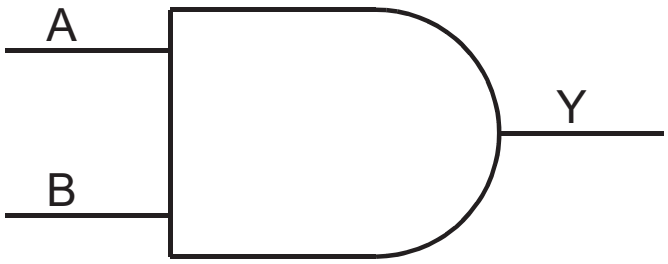


Table 1-1. AND2 I/O

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

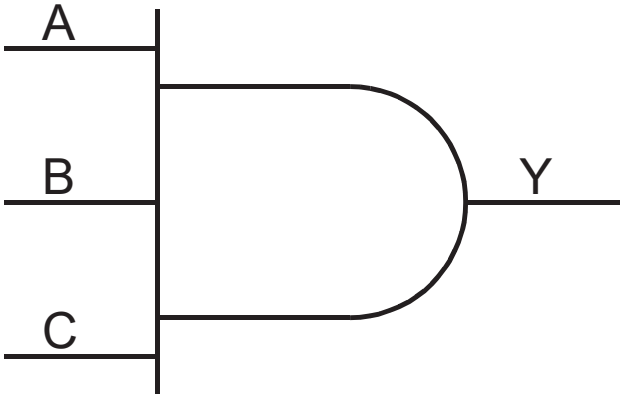


Table 1-3. AND3 I/O

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

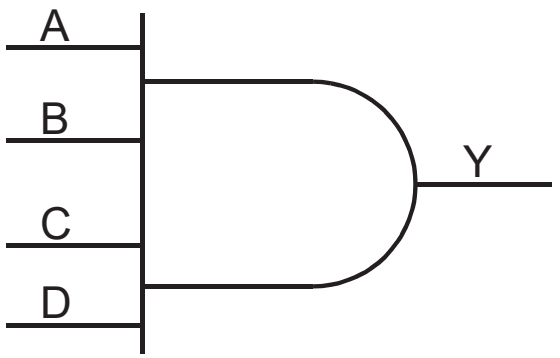


Table 1-5. AND4 I/O

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. ARI1 (Ask a Question)

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

Figure 1-4. ARI1

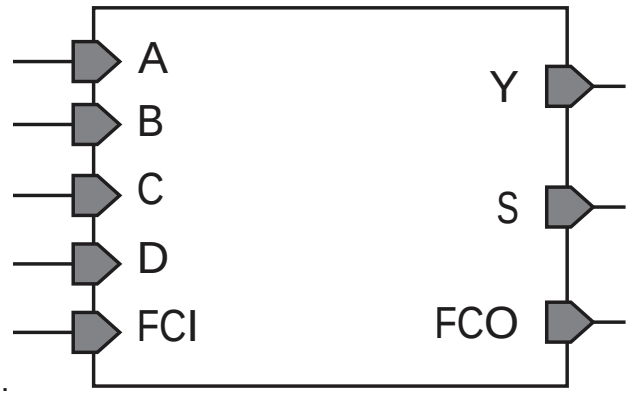


Table 1-7. ARI1 I/O

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 1-8. 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 1-9. ARI1 Truth Table for S

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

ARI1 LOGIC

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 1-10. ARI1 INIT[17:16] String Interpretation

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

Table 1-11. ARI1 INIT[19:18] String Interpretation

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

Table 1-12. FCO Truth Table

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

1.5. 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 maco except that it does not generate the Final Carry Out (FCO).

Note: FC1 and FC0 do not perform any functions.

Figure 1-5. ARI1_CC

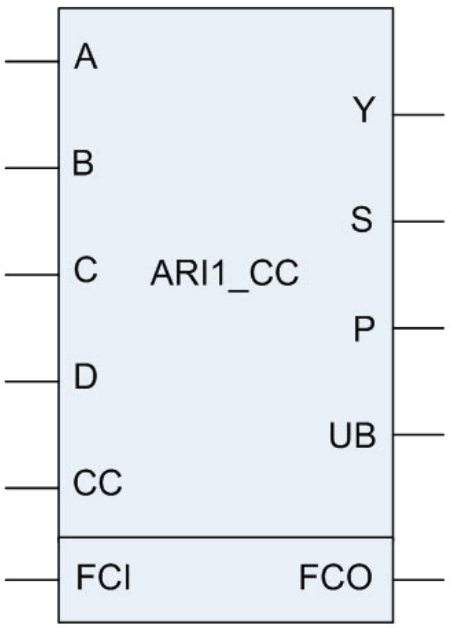


Table 1-13. ARI1_CC I/O

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.

Table 1-14. INTERPRETATION OF 16 LSB OF THE INIT STRING FOR ARI1_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]	
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 1-15. 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 1-16. 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

1.6. BUFD (Ask a Question)

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

Figure 1-6. BUFD

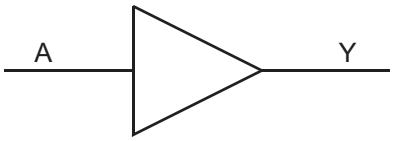


Table 1-17. BUFD I/O

Input	Output
A	Y

Table 1-18. BUFD Truth Table

A	Y
0	0
1	1

1.7. BUFF (Ask a Question)

Buffer.

Figure 1-7. BUFF

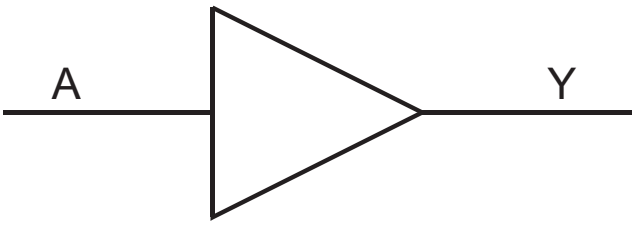


Table 1-19. BUFF I/O

Input	Output
A	Y

Table 1-20. BUFF Truth Table

A	Y
0	0
1	1

1.8. 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.

1.9. 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 1-8. CFG2

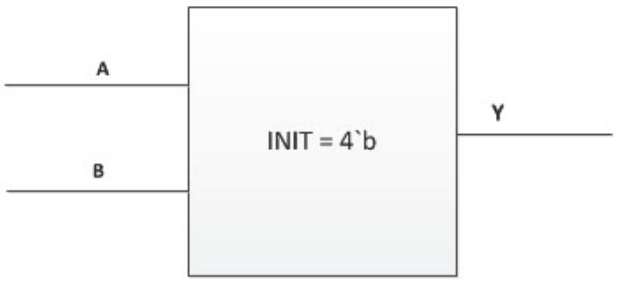


Table 1-21. CFG2 I/O

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

Table 1-22. CFG2 INIT String Interpretation

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

1.10. 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 1-9. CFG3

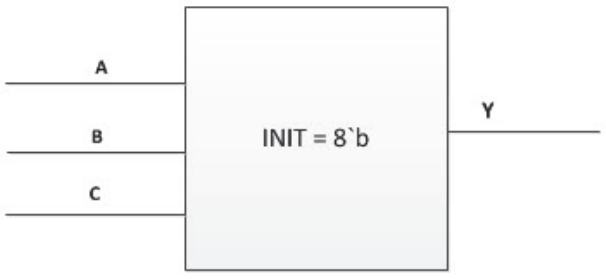


Table 1-23. CFG3 I/O

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

Table 1-24. CFG3 INIT String Interpretation

CBA	Y
000	INIT[0]

Table 1-24. 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]

1.11. 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 1-10. CFG4

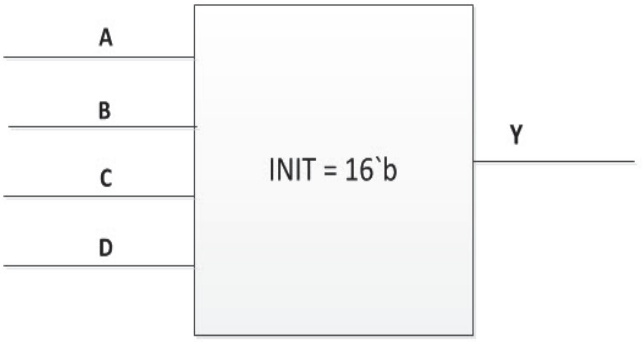


Table 1-25. CFG4 I/O

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

Table 1-26. 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 1-26. CFG4 Truth Table (continued)

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

1.12. INV (Ask a Question)

Inverter.

Figure 1-11. INV

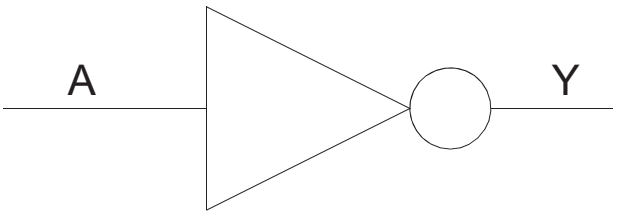


Table 1-27. INV I/O

Input	Output
A	Y

Table 1-28. INV Truth Table

A	Y
0	1
1	0

1.13. INVD (Ask a Question)

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

Figure 1-12. INVD

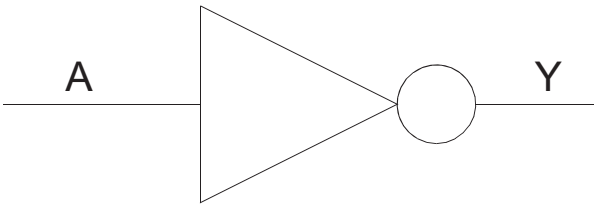


Table 1-29. INVD I/O

Input	Output
A	Y

Table 1-30. INVD Truth Table

A	Y
0	1
1	0

1.14. MX2 (Ask a Question)

2 to 1 Multiplexer.

Figure 1-13. MX2

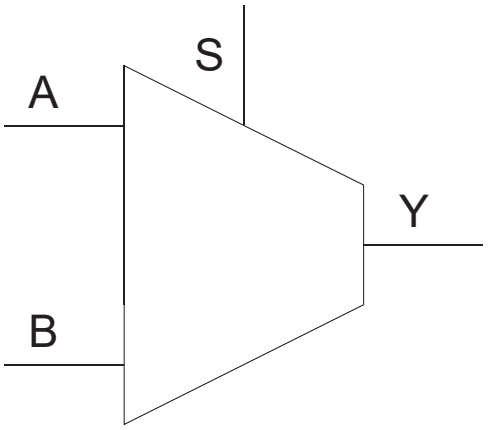


Table 1-31. MX2 I/O

Input	Output
A, B, S	Y

Table 1-32. MX2 Truth Table

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

1.15. MX4 (Ask a Question)

4 to 1 Multiplexer.

This macro uses two logic modules.

Figure 1-14. MX4

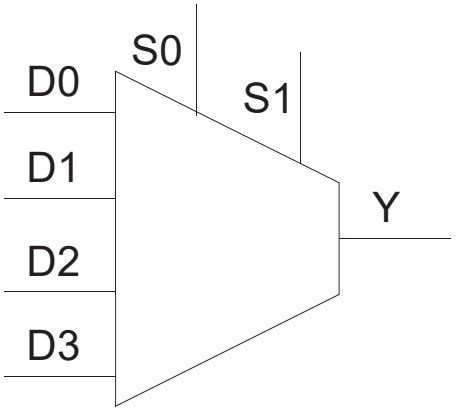


Table 1-33. MX4 I/O

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

Table 1-34. 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

1.16. NAND2 (Ask a Question)

2-Input NAND.

Figure 1-15. NAND2

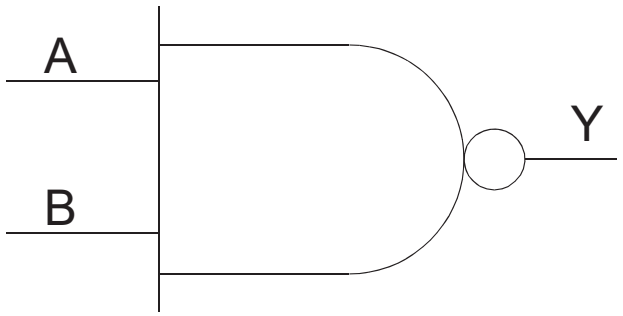


Table 1-35. NAND2 I/O

Input	Output
A, B	Y

Table 1-36. NAND2 Truth Table

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

1.17. NAND3 (Ask a Question)

3-Input NAND.

Figure 1-16. NAND3

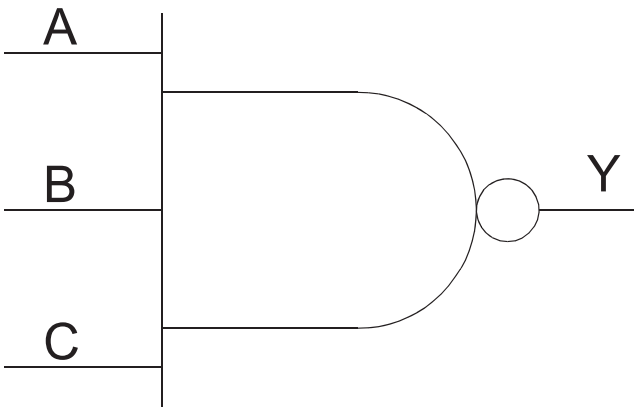


Table 1-37. NAND3 I/O

Input	Output
A, B, C	Y

Table 1-38. NAND3 Truth Table

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

1.18. NAND4 (Ask a Question)

4-input NAND.

Figure 1-17. NAND4

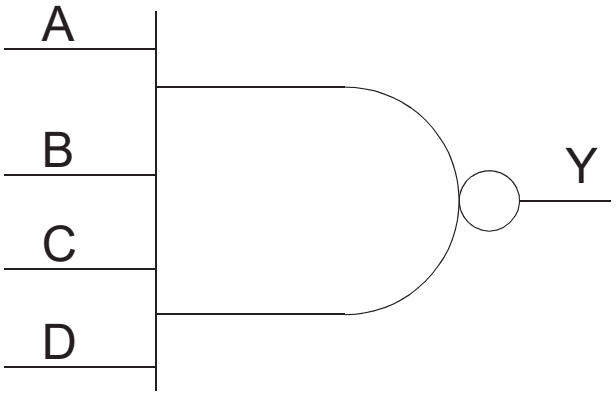


Table 1-39. NAND4 I/O

Input	Output
A, B, C, D	Y

Table 1-40. 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

1.19. NOR2 (Ask a Question)

2-input NOR.

Figure 1-18. NOR2

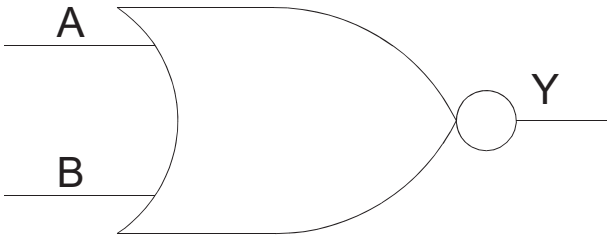


Table 1-41. NOR2 I/O

Input	Output
A, B	Y

Table 1-42. NOR2 Truth Table

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

1.20. NOR3 (Ask a Question)

3-input NOR.

Figure 1-19. NOR3

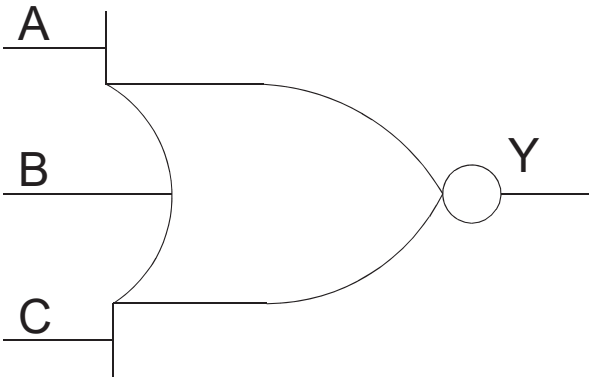


Table 1-43. NOR3 I/O

Input	Output
A, B, C	Y

Table 1-44. NOR3 Truth Table

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

1.21. NOR4 (Ask a Question)

4-input NOR.

Figure 1-20. NOR3

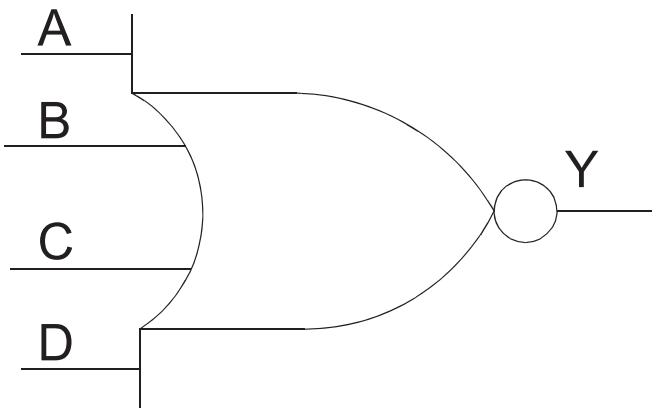


Table 1-45. NOR4 I/O

Input	Output
A, B, C, D	Y

Table 1-46. 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

1.22. OR2 (Ask a Question)

2-input OR.

Figure 1-21. OR2

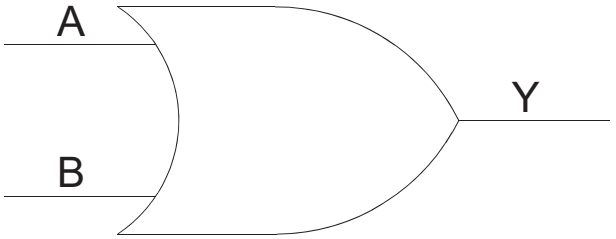


Table 1-47. OR2 I/O

Input	Output
A, B	Y

Table 1-48. OR2 Truth Table

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

1.23. OR3 (Ask a Question)

3-input OR.

Figure 1-22. OR3

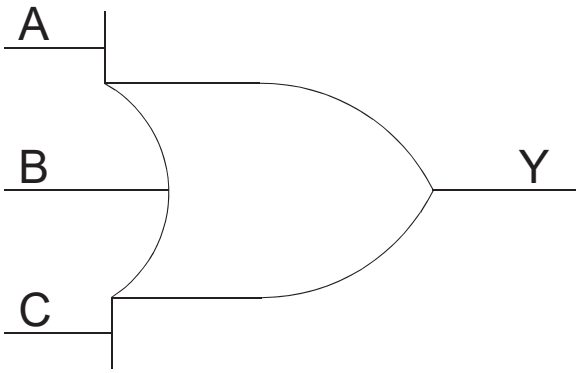


Table 1-49. OR3 I/O

Input	Output
A, B, C	Y

Table 1-50. OR3 Truth Table

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

1.24. OR4 (Ask a Question)

4-input OR.

Figure 1-23. OR4

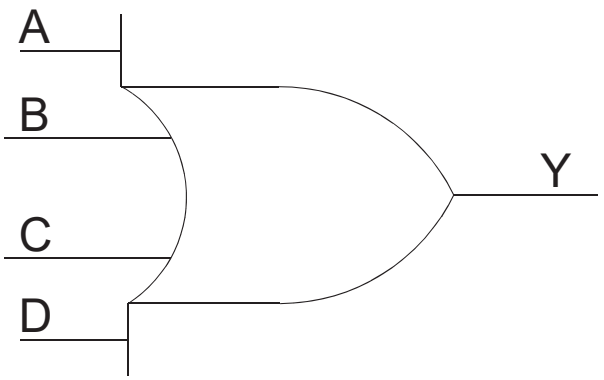


Table 1-51. OR4 I/O

Input	Output
A, B, C, D	Y

Table 1-52. 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

1.25. XOR2 (Ask a Question)

2-input XOR.

Figure 1-24. XOR2

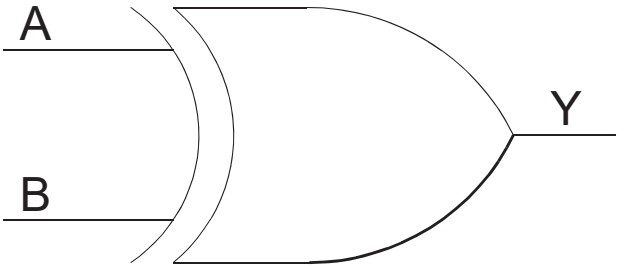


Table 1-53. XOR2 I/O

Input	Output
A, B	Y

Table 1-54. XOR2 Truth Table

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

1.26. XOR3 (Ask a Question)

3-input XOR.

Figure 1-25. XOR3

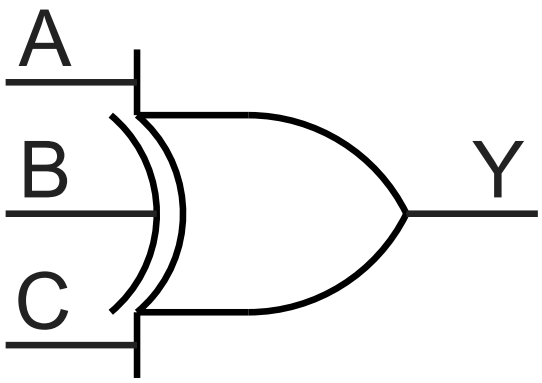


Table 1-55. XOR3 I/O

Input	Output
A, B, C	Y

Table 1-56. 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

1.27. XOR4 (Ask a Question)

4-input XOR.

Figure 1-26. XOR4

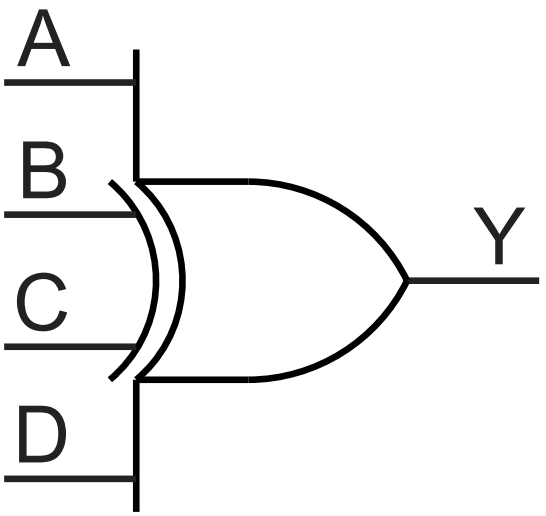


Table 1-57. XOR4 I/O

Input	Output
A, B, C, D	Y

Table 1-58. 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

1.28. XOR8 (Ask a Question)

8-input XOR.

This macro uses two logic modules.

Figure 1-27. XOR8

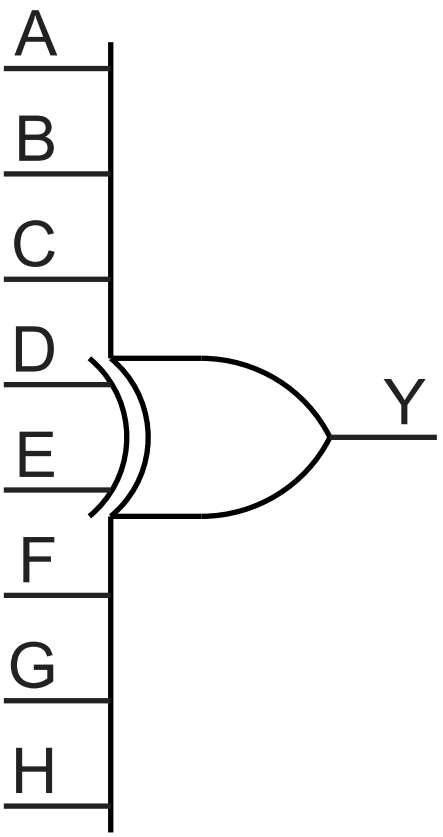


Table 1-59. XOR8 I/O

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 1-60. 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

1.29. LIVE_PROBE_FB (Ask a Question)

This is a special-purpose macro that feeds the live probe signals back to the fabric. You can connect the PROBE_A/PROBE_B signals to any unused I/O during design generation. This is useful if PROBE_A/PROBE_B cannot be brought out for debugging due to board limitations. PROBE_A and PROBE_B pins must be reserved, if LIVE_PROBE_FB macro is used. This macro is not supported in simulation.

Figure 1-28. LIVE_PROBE_FB

2. Sequential Logic (Ask a Question)

Important: In SmartFusion 2 and IGLOO 2 devices, flip-flops do not power up in a known state. If no effective reset or set signal is applied to a flip-flop, the output state is considered indeterminate. This means the flip-flop could power up in either a '0' or '1' state, or even in a metastable state. It is recommended to initialize flip-flops to a known state using a reset signal at power-up.

2.1. DFN1 (Ask a Question)

D-Type Flip-Flop.

Figure 2-1. DFN1

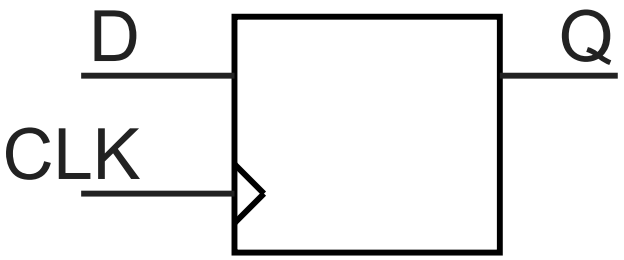


Table 2-1. DFN1 I/O

Input	Output
D, CLK	Q

Table 2-2. DFN1 Truth Table

CLK	D	Q_{n+1}
not Rising	X	Q_n
—	D	D

2.2. DFN1C0 (Ask a Question)

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

Figure 2-2. DFN1C0

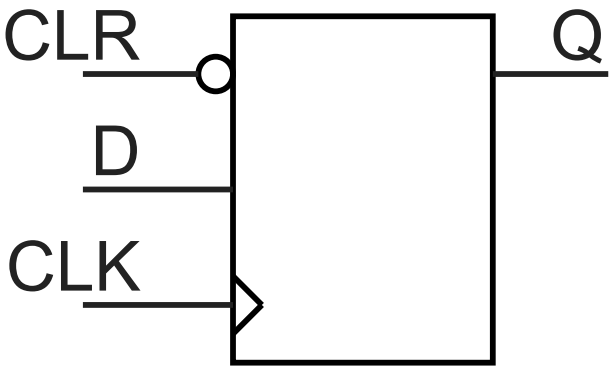


Table 2-3. DFN1C0 I/O

Input	Output
D, CLK, CLR	Q

Table 2-4. DFN1C0 Truth Table

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

2.3. DFN1E1 (Ask a Question)

D-Type Flip-Flop with active high Enable.

Figure 2-3. DFN1E1

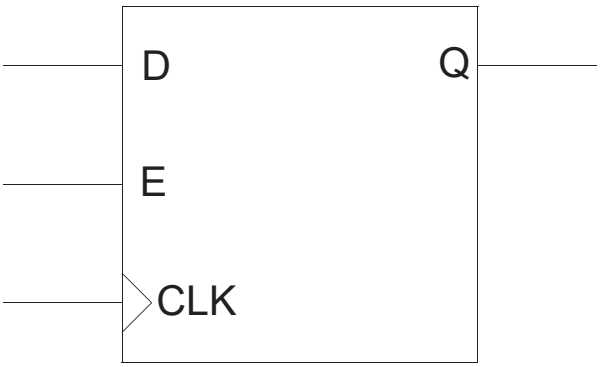


Table 2-5. DFN1E1 I/O

Input	Output
D, E, CLK	Q

Table 2-6. DFN1E1 Truth Table

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

2.4. DFN1E1C0 (Ask a Question)

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

Figure 2-4. DFN1E1C0

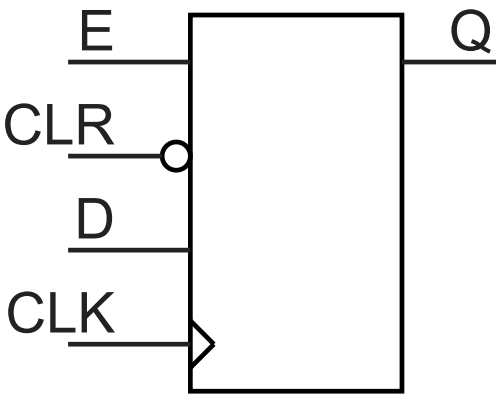


Table 2-7. DFN1E1C0 I/O

Input	Output
CLR, D, E, CLK	Q

Table 2-8. 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

2.5. DFN1E1P0 (Ask a Question)

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

Figure 2-5. DFN1E1P0

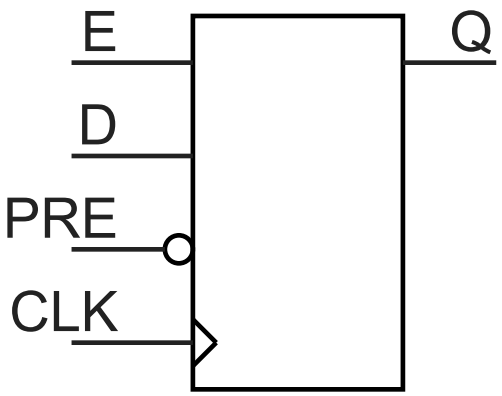


Table 2-9. DFN1E1P0 I/O

Input	Output
D, E, PRE, CLK	Q

Table 2-10. DFN1E1P0 Truth Table

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

2.6. DLN1 (Ask a Question)

Data Latch.

Figure 2-6. DLN1

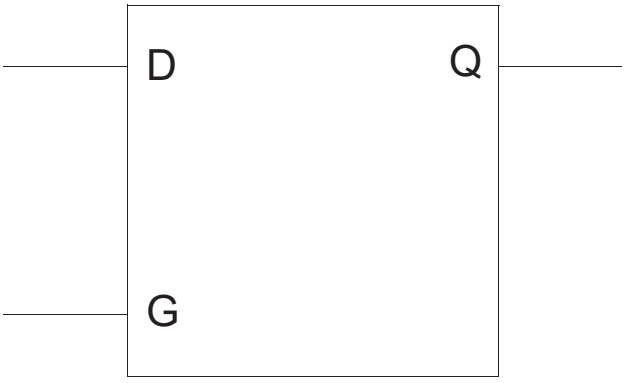


Table 2-11. DLN1 I/O

Input	Output
D, G	Q

Table 2-12. DLN1 Truth Table

G	D	Q
0	X	Q
1	D	D

2.7. DLN1C0 (Ask a Question)

Data Latch with active-low Clear.

Figure 2-7. DLN1C0

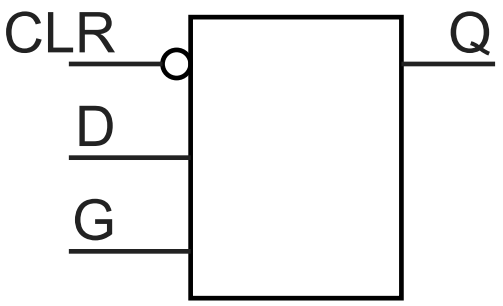


Table 2-13. I/O

Input	Output
CLR, D, G	Q

Table 2-14. Truth Table

CLR	G	D	Q
0	X	X	0
1	0	X	Q
1	1	D	D

2.8. DLN1P0 (Ask a Question)

Data Latch with active-low Preset.

Figure 2-8. DLN1P0

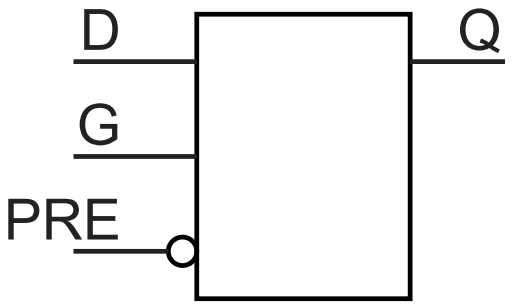


Table 2-15. DLN1C0 I/O

Input	Output
D, G, PRE	Q

Table 2-16. DLN1C0 Truth Table

PRE	G	D	Q
0	X	X	1
1	0	X	Q
1	1	D	D

2.9. SLE (Ask a Question)

Sequential Logic Element.

Figure 2-9. SLE

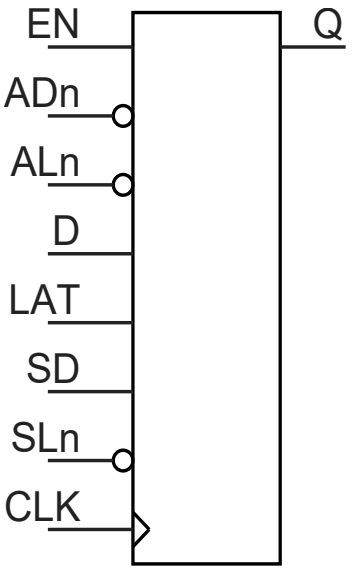


Table 2-17. SLE I/O

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 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]	Active-High Latch Enable. This signal enables latch mode when high and register mode when low.	

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

Table 2-18. SLE Truth Table

ALn	ADn	LAT	CLK	EN	SLn	SD	D	Q_{n+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
1	X	1	0	X	X	X	X	Qn
1	X	1	1	0	X	X	X	Qn
1	X	1	1	1	0	SD	X	SD
1	X	1	1	1	1	X	D	D

3. I/O (Ask a Question)

3.1. BIBUF (Ask a Question)

Bidirectional Buffer.

Figure 3-1. BIBUF

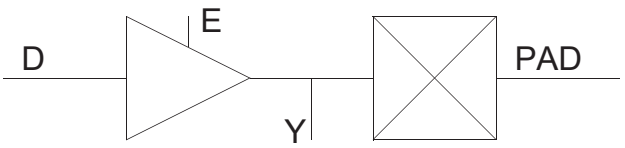


Table 3-1. BIBUF I/O

Input	Output
D, E, PAD	PAD, Y

Table 3-2. BIBUF 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

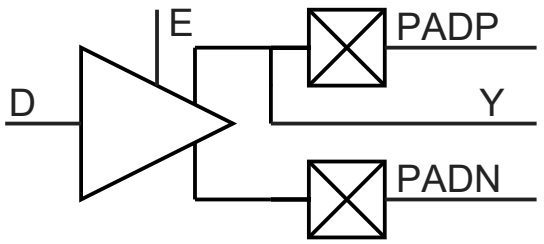


Table 3-3. BIBUF_DIFF I/O

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

Table 3-4. BIBUF_DIFF 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. BIBUF_DIFF 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 the global network.

Figure 3-3. CLKBIBUF

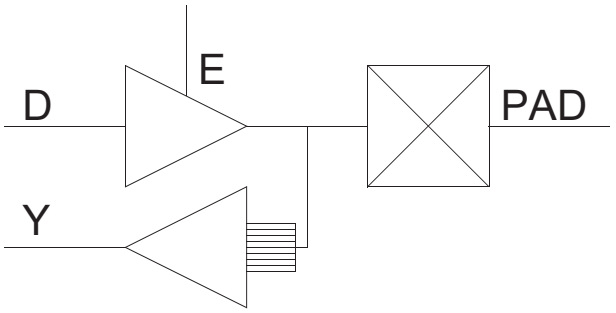


Table 3-5. CLKBIBUF I/O

Input	Output
D, E, PAD	PAD, Y

Table 3-6. CLKBIBUF 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 the global network.

Figure 3-4. CLKBUF

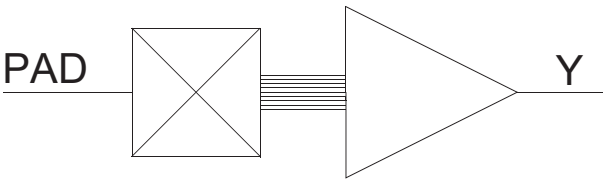


Table 3-7. CLKBUF I/O

Input	Output
PAD	Y

Table 3-8. CLKBUF Truth Table

PAD	Y
0	0
1	1

3.5. CLKBUF_DIFF (Ask a Question)

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

Figure 3-5. CLKBUF_DIFF

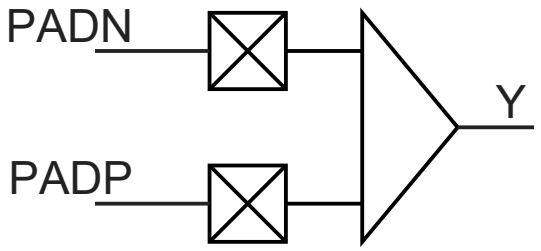


Table 3-9. INBUF_DIFF I/O

Input	Output
PADP, PADN	Y

Table 3-10. INBUF_DIFF Truth Table

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

3.6. 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 3-6. DDR_IN

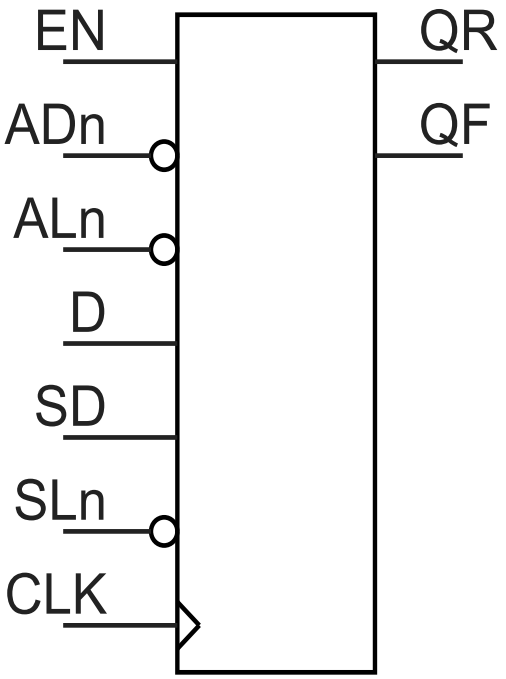


Table 3-11. DDR_IN I/O

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 (low), QR and QF go to the value of SD at the rising edge of CLK.	

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

Table 3-12. DDR_IN TRUTH TABLE

ALn	CLK	EN	SLn	df_{n+1} (Internal Signal)	QR_{n+1}	QF_{n+1}
0	X	X	X	$!AD_n$	$!AD_n$	$!AD_n$
1	Not rising	X	X	df_n	QR_n	QF_n
1	—	0	X	df_n	QR_n	QF_n
1	—	1	0	df_n	SD	SD
1	—	1	1	df_n	D	df_n
1	↓	X	X	D	QR_n	QF_n

3.7. 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 3-7. DDR_OE_UNIT

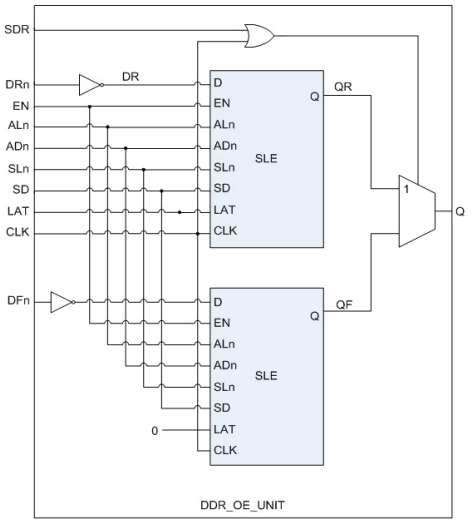


Table 3-13. DDR_OE_UNIT I/O

Input		Output
Name	Function	
DRn	Data input (Rising Edge)	Q
DFn	Data input (Falling Edge)	
CLK	Clock input	
EN	Active-High CLK enable	
AL_n	Asynchronous load. This active-low signal either sets the register or clears the register depending on the value of ADn.	
AD_n	Static asynchronous load data. When ALn is active, Q goes to the complement of ADn.	
SL_n	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 (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 3-14. DDR_OE_UNIT TRUTH TABLE

SDR	ALn	CLK	EN	SLn	QR_{n+1}	QF_{n+1}	Q_{n+1}
0	0	X	X	X	$!AD_n$	$!AD_n$	$!AD_n$
0	1	1	X	X	QR_n	QF_n	QR_n
0	1	—	0	X	QR_n	QF_n	QR_{n+1}

Table 3-14. DDR_OE_UNIT TRUTH TABLE (continued)

SDR	ALn	CLK	EN	SLn	QR_{n+1}	QF_{n+1}	Q_{n+1}
0	1	—	1	0	SD	SD	QR_{n+1}
0	1	—	1	1	!DRn	!DFn	QR_{n+1}
0	1	0	X	X	QR_n	QF_n	QF_n

3.8. 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 3-8. DDR_OUT

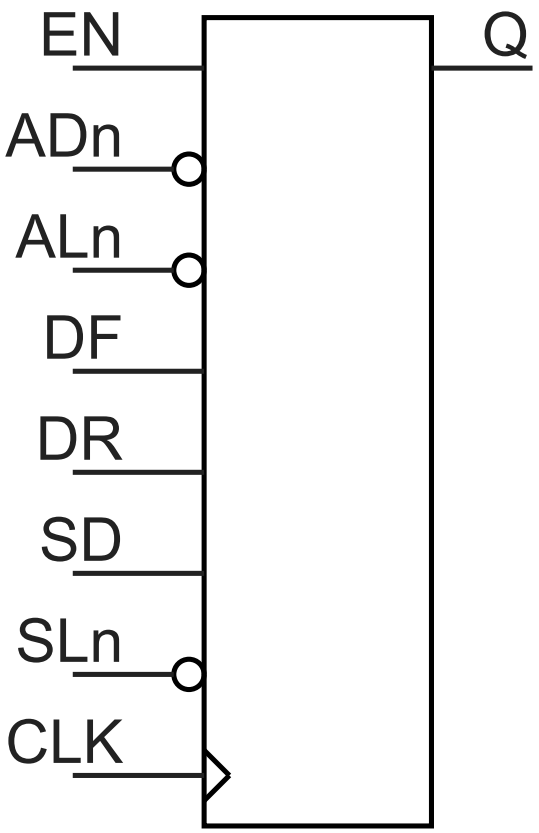


Table 3-15. DDR_OUT I/O

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

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

Table 3-16. DDR_OUT TRUTH TABLE

ALn	CLK	EN	SLn	QR_{n+1}	QF_{n+1}	Q_{n+1}
0	X	X	X	$!AD_n$	$!AD_n$	$!AD_n$
1	1	X	X	QR_n	QF_n	QR_n
1	—	0	X	QR_n	QF_n	QR_{n+1}
1	—	1	0	SD	SD	QR_{n+1}
1	—	1	1	DR	DF	QR_{n+1}
1	0	X	X	QR_n	QF_n	QF_n

3.9. GCLKBIBUF (Ask a Question)

Bidirectional I/O macro with gated input to the global network; the Enable signal can be used to turn off the global network to save power.

Figure 3-9. GCLKBIBUF

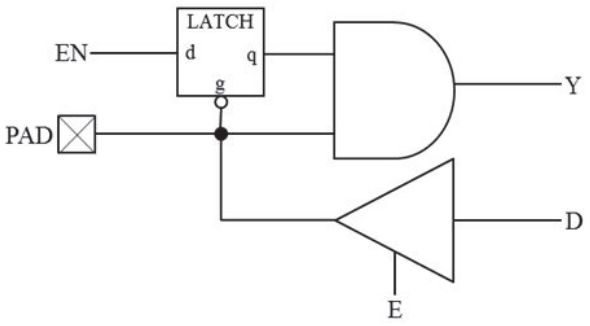


Table 3-17. GCLKBIBUF I/O

Input	Output
D, E, EN, PAD	Y, PAD

Table 3-18. GCLKBIBUF TRUTH TABLE

D	E	EN	PAD	q	Y
X	0	0	0	0	0

Table 3-18. GCLKBIBUF TRUTH TABLE (continued)

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

3.10. GCLKBUF (Ask a Question)

Gated input I/O macro to the global network. The Enable signal can turn off the global network to save power.

Figure 3-10. GCLKBUF

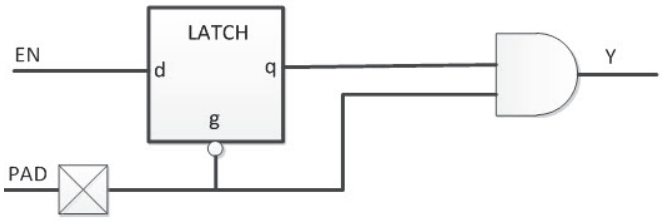


Table 3-19. GCLKBUF I/O

Input	Output
PAD, EN	Y

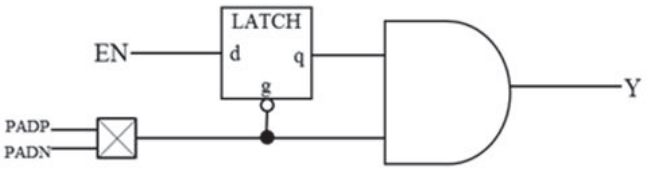
Table 3-20. GCLKBUF Truth Table

PAD	EN	q	Y
0	0	0	0
0	1	1	0
1	X	q	q
Z	X	X	X

3.11. GCLKBUF_DIFF (Ask a Question)

Gated differential I/O macro to global network; the Enable signal can be used to turn off the global network.

Figure 3-11. GCLKBUF_DIFF



Differential

Table 3-21. GCLKBUF_DIFF I/O

Input	Output
PADP, PADN, EN	Y

Table 3-22. GCLKBUF_DIFF Truth Table

PADP	PADN	EN	q	Y
0	1	0	0	0
0	1	1	1	0
1	0	X	q	q
0	0	X	X	X
1	1	X	X	X
Z	Z	X	X	X

3.12. INBUF (Ask a Question)

Input Buffer.

Figure 3-12. INBUF

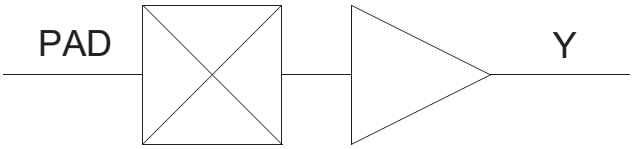


Table 3-23. INBUF I/O

Input	Output
PAD	Y

Table 3-24. INBUF Truth Table

PAD	Y
Z	X
0	0
1	1

3.13. INBUF_DIFF (Ask a Question)

Input Buffer, Differential I/O.

Figure 3-13. INBUF_DIFF

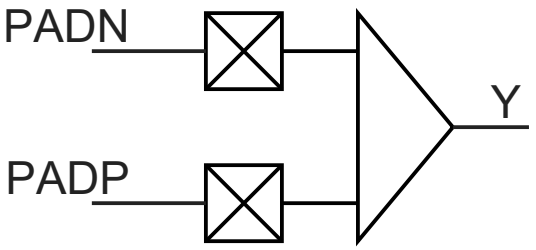


Table 3-25. INBUF_DIFF I/O

Input	Output
PADP, PADN	Y

Table 3-26. INBUF_DIFF Truth Table

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

3.14. IOIN_IB (Ask a Question)

Buffer macro available in post-layout netlist only.

Figure 3-14. IOIN_IB

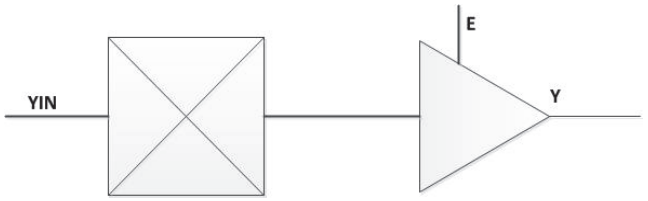


Table 3-27. IOIN_IB I/O

Input	Output
YIN, E	Y

E input is not used.

Table 3-28. IOIN_IB TRUTH TABLE

YIN	Y
Z	X
0	0
1	1

3.15. IOINFF (Ask a Question)

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

Figure 3-15. IOINFF

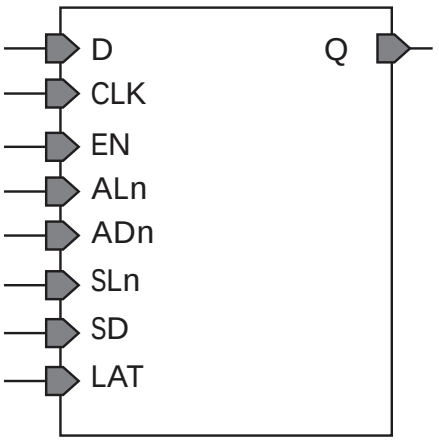


Table 3-29. IOINFF I/O

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	
LAT[1]	Latch Enable	

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

Table 3-30. IOINFF TRUTH TABLE

ALn	ADn	LAT	CLK	EN	SLn	SD	D	Q_{n+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
1	X	1	0	X	X	X	X	Qn
1	X	1	1	0	X	X	X	Qn
1	X	1	1	1	0	SD	X	SD
1	X	1	1	1	1	X	D	D

3.16. IOOEFF (Ask a Question)

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

Figure 3-16. IOOEFF

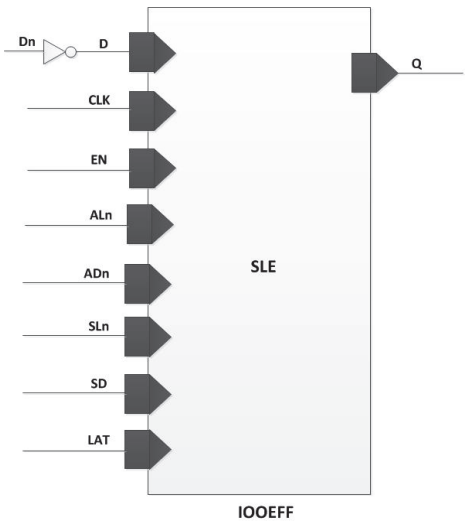


Table 3-31. IOOEFF I/O

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	
LAT[1]	Latch Enable	

Note:

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

Table 3-32. IOOEFF TRUTH TABLE

ALn	LAT	CLK	EN	SLn	Q
1	0	not rising	X	X	Q
1	0	rising	0	X	Q
1	0	rising	1	1	!Dn
1	0	rising	1	0	SD
0	0	X	X	X	!ADn
1	1	0	X	X	Q
1	1	1	0	X	Q
1	1	1	1	1	!Dn
1	1	1	1	0	SD
0	1	X	X	X	!ADn

3.17. IOPAD_IN (Ask a Question)

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

Figure 3-17. IOPAD_IN

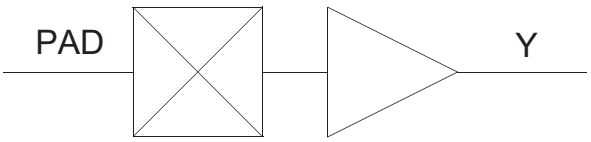


Table 3-33. IOPAD_IN I/O

Input	Output
PAD	Y

Table 3-34. IOPAD_IN TRUTH TABLE

PAD	Y
Z	X
0	0
1	1

3.18. IOPAD_TRI (Ask a Question)

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

Figure 3-18. IOPAD_TRI

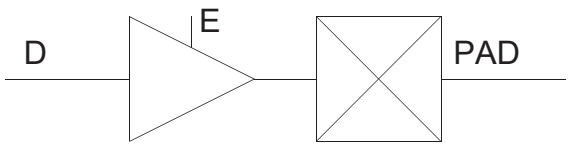


Table 3-35. IOPAD_TRI I/O

Input	Output
D, E	PAD

Table 3-36. IOPAD_TRI TRUTH TABLE

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

3.19. OUTBUF (Ask a Question)

Output buffer.

Figure 3-19. OUTBUF

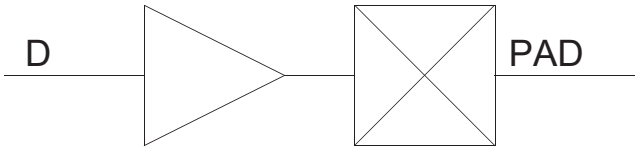


Table 3-37. OUTBUF I/O

Input	Output
D	PAD

Table 3-38. OUTBUF Truth Table

D	PAD
0	0
1	1

3.20. OUTBUF_DIFF (Ask a Question)

Output buffer, Differential I/O.

Figure 3-20. OUTBUF_DIFF

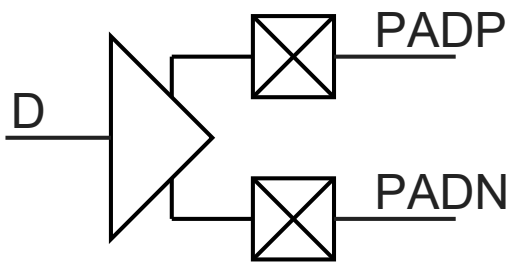


Table 3-39. OUTBUF_DIFF I/O

Input	Output
D	PADP, PADN

Table 3-40. OUTBUF_DIFF Truth Table

D	PADP	PADN
0	0	1
1	1	0

3.21. TRIBUFF (Ask a Question)

Tristate output buffer.

Figure 3-21. TRIBUFF

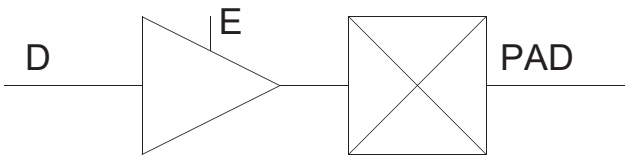


Table 3-41. TRIBUFF I/O

Input	Output
D, E	PAD

Table 3-42. TRIBUFF Truth Table

D	E	PAD
X	0	Z
D	1	D

3.22. TRIBUFF_DIFF (Ask a Question)

Tristate output buffer, Differential I/O.

Figure 3-22. TRIBUFF_DIFF

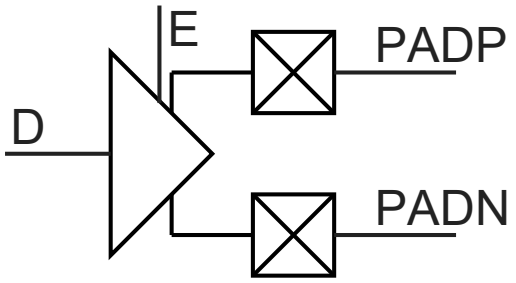


Table 3-43. TRIBUFF_DIFF I/O

Input	Output
D, E	PADP, PADN

Table 3-44. Truth Table

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

3.23. 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 3-23. UJTAG

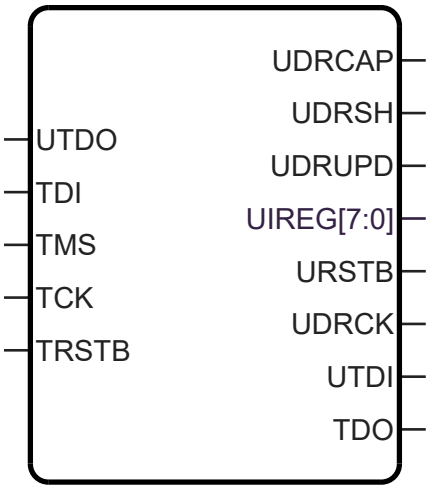


Table 3-45. Ports and Descriptions

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. **Note:** UDRCK must be connected to a global macro such as CLKINT. If this is not done, Synthesis/Compile will add it to the netlist to legalize it.
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.
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).

4. Clocking (Ask a Question)

4.1. CLKBIBUF (Ask a Question)

Bidirectional Buffer with Input to the global network.

Figure 4-1. CLKBIBUF

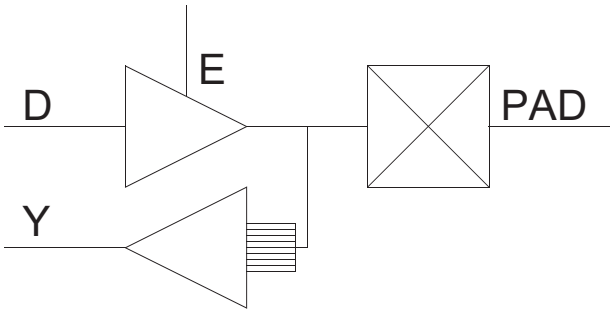


Table 4-1. CLKBIBUF I/O

Input	Output
D, E, PAD	PAD, Y

Table 4-2. CLKBIBUF 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

4.2. CLKBUF (Ask a Question)

Input Buffer to the global network.

Figure 4-2. CLKBUF

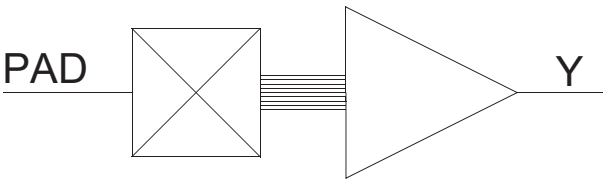


Table 4-3. CLKBUF I/O

Input	Output
PAD	Y

Table 4-4. CLKBUF Truth Table

PAD	Y
0	0
1	1

4.3. CLKBUF_DIFF (Ask a Question)

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

Figure 4-3. CLKBUF_DIFF

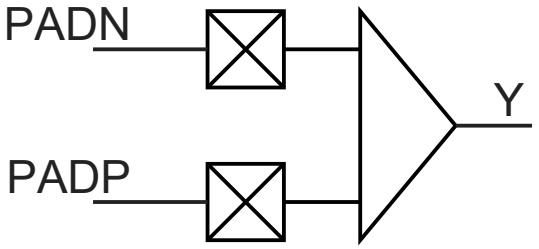


Table 4-5. INBUF_DIFF I/O

Input	Output
PADP, PADN	Y

Table 4-6. INBUF_DIFF Truth Table

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

4.4. CLKINT (Ask a Question)

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

Figure 4-4. CLKINT

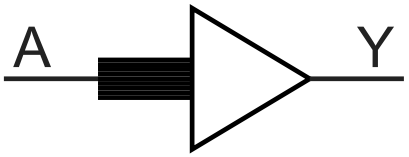


Table 4-7. CLKINT I/O

Input	Output
A	Y

Table 4-8. CLKINT Truth Table

A	Y
0	0
1	1

4.5. 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 4-5. CLKINT_PRESERVE

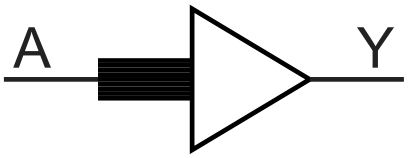


Table 4-9. CLKINT_PRESERVE I/O

Input	Output
A	Y

Table 4-10. CLKINT_PRESERVE Truth Table

A	Y
0	0
1	1

4.6. GB (Ask a Question)

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

Figure 4-6. GB

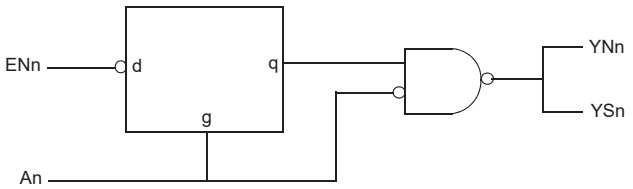


Table 4-11. GB I/O

Input	Output
An, ENn	YNn, YSn

Table 4-12. GB TRUTH TABLE

An	ENn	q (internal signal)	YNn	YSn
1	0	1	1	1
1	1	0	1	1
0	X	q	!q	!q

4.7. GB_NG (Ask a Question)

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

Figure 4-7. GB_NG

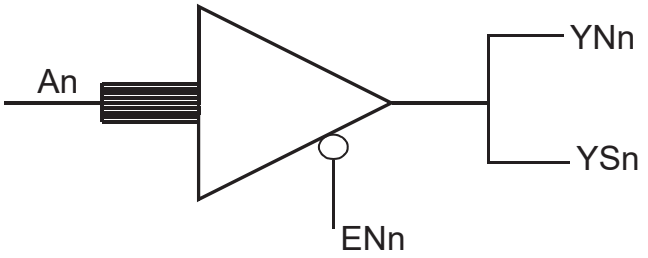


Table 4-13. GB_NG I/O

Input	Output
An, ENn	YNn, YSn

Table 4-14. TRUTH TABLE

An	ENn	YNn	YSn
0	0	0	0
1	0	1	1
X	1	X	X

4.8. GBM (Ask a Question)

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

Figure 4-8. GBM

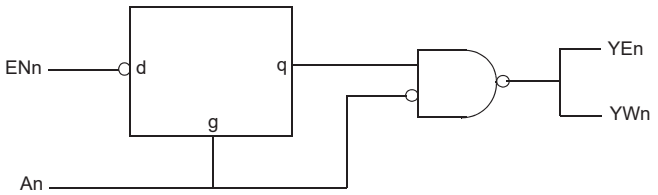


Table 4-15. GBM I/O

Input	Output
An, ENn	YEn, YWn

Table 4-16. GBM TRUTH TABLE

An	ENn	q (internal signal)	YEn	YWn
1	0	1	1	1
1	1	0	1	1
0	X	q	!q	!q

4.9. GBM_NG (Ask a Question)

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

Figure 4-9. GBM_NG

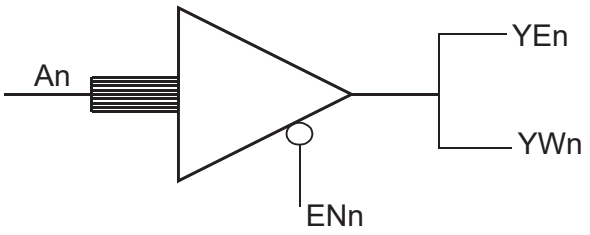


Table 4-17. GBM_NG I/O

Input	Output
An, ENn	YEn, YWn

Table 4-18. GBM_NG TRUTH TABLE

An	ENn	YEn	YWn
0	0	0	0
1	0	1	1
X	1	X	X

4.10. GCLKBIBUF (Ask a Question)

Bidirectional I/O macro with gated input to the global network; the Enable signal can be used to turn off the global network to save power.

Figure 4-10. GCLKBIBUF

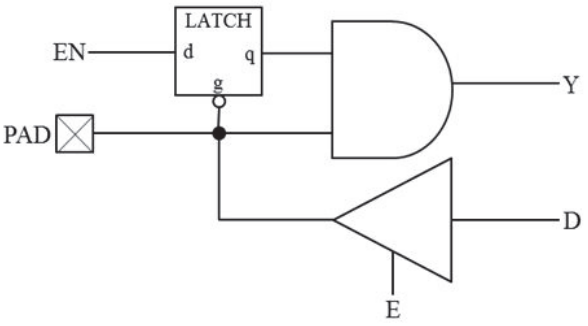


Table 4-19. GCLKBIBUF I/O

Input	Output
D, E, EN, PAD	Y, PAD

Table 4-20. GCLKBIBUF TRUTH TABLE

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

4.11. GCLKBUF (Ask a Question)

Gated input I/O macro to the global network. The Enable signal can turn off the global network to save power.

Figure 4-11. GCLKBUF

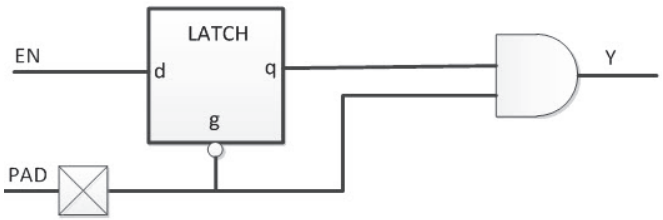


Table 4-21. GCLKBUF I/O

Input	Output
PAD, EN	Y

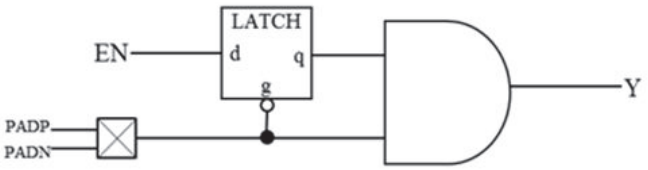
Table 4-22. GCLKBUF Truth Table

PAD	EN	q	Y
0	0	0	0
0	1	1	0
1	X	q	q
Z	X	X	X

4.12. GCLKBUF_DIFF (Ask a Question)

Gated differential I/O macro to global network; the Enable signal can be used to turn off the global network.

Figure 4-12. GCLKBUF_DIFF



Differential

Table 4-23. GCLKBUF_DIFF I/O

Input	Output
PADP, PADN, EN	Y

Table 4-24. GCLKBUF_DIFF Truth Table

PADP	PADN	EN	q	Y
0	1	0	0	0
0	1	1	1	0
1	0	X	q	q
0	0	X	X	X
1	1	X	X	X
Z	Z	X	X	X

4.13. GCLKINT (Ask a Question)

Gated macro used to route an internal fabric signal to the global network. The Enable signal can be used to turn off the global network to save power.

Figure 4-13. GCLKINT

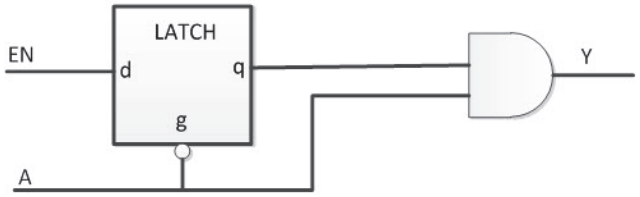


Table 4-25. GCLKINT I/O

Input	Output
A, EN	Y

Table 4-26. GCLKINT Truth Table

A	EN	q (Internal Signal)	Y
0	0	0	0
0	1	1	0
1	X	q	q

4.14. IOINFF (Ask a Question)

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

Figure 4-14. IOINFF

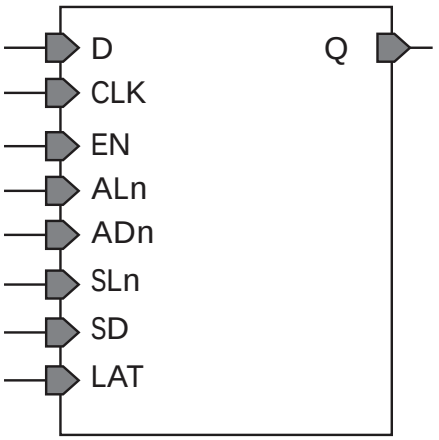


Table 4-27. IOINFF I/O

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	
LAT[1]	Latch Enable	

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

Table 4-28. IOINFF TRUTH TABLE

ALn	ADn	LAT	CLK	EN	SLn	SD	D	Q_{n+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
1	X	1	0	X	X	X	X	Qn
1	X	1	1	0	X	X	X	Qn
1	X	1	1	1	0	SD	X	SD
1	X	1	1	1	1	X	D	D

4.15. RCLKINT (Ask a Question)

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

Figure 4-15. RCLKINT

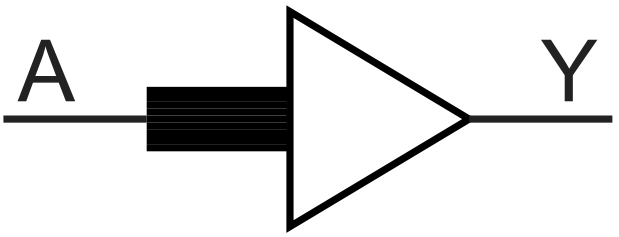


Table 4-29. RCLKINT I/O

Input	Output
A	Y

Table 4-30. RCLKINT Truth Table

A	Y
0	0
1	1

4.16. RGB (Ask a Question)

Back-annotated macro used to route an internal fabric signal to row the global network buffer.

Figure 4-16. RGB

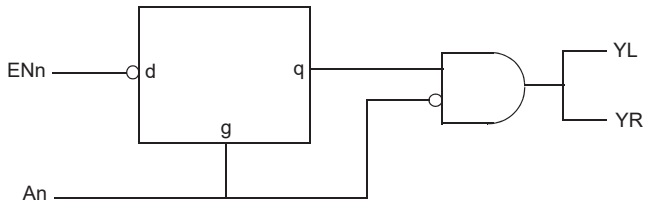


Table 4-31. RGB I/O

Input	Output
An, ENn	YL, YR

Table 4-32. RGB TRUTH TABLE

An	ENn	q (internal signal)	YL	YR
1	0	1	0	0
1	1	0	0	0
0	X	q	q	q

4.17. RGB_NG (Ask a Question)

Back-annotated macro used to route an internal fabric signal to a row global buffer.

Figure 4-17. RGB_NG

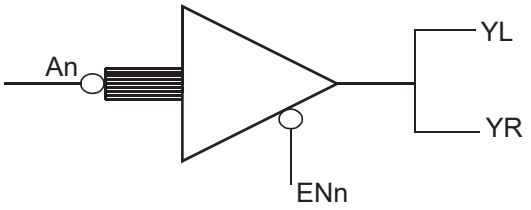


Table 4-33. RGB_NG I/O

Input	Output
An, ENn	YL, YR

Table 4-34. RGB_NG TRUTH TABLE

An	ENn	YL	YR
0	0	1	1
1	0	0	0
X	1	X	X

4.18. RGCLKINT (Ask a Question)

Gated macro used to route an internal fabric signal to a row global buffer, thus creating a local clock. The Enable signal can be used to turn off the local clock to save power.

Figure 4-18. RGCLKINT

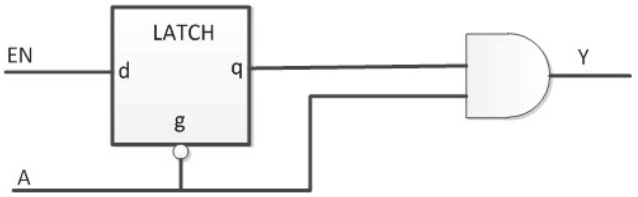


Table 4-35. RGCLKINT I/O

Input	Output
A, EN	Y

Table 4-36. RGCLKINT Truth Table

A	EN	q (Internal Signal)	Y
0	0	0	0
0	1	1	0
1	X	q	q

5. Special (Ask a Question)

5.1. FCEND_BUFF (Ask a Question)

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

Figure 5-1. FCEND_BUFF

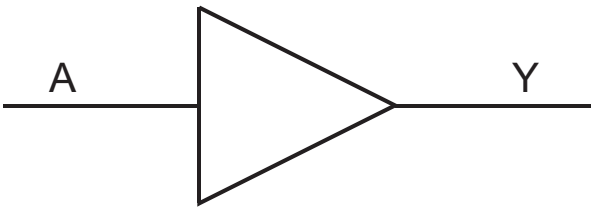


Table 5-1. FCEND_BUFF I/O

Input	Output
A	Y

Table 5-2. FCEND_BUFF Truth Table

A	Y
0	0
1	1

5.2. FCINIT_BUFF (Ask a Question)

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

Figure 5-2. FCINIT_BUFF

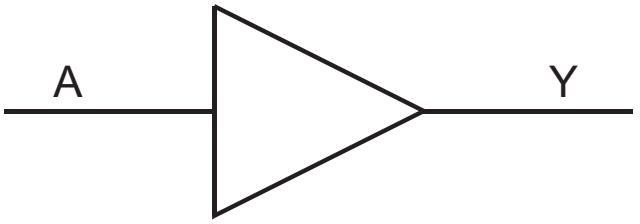


Table 5-3. FCINIT_BUFF I/O

Input	Output
A	Y

Table 5-4. FCINIT_BUFF Truth Table

A	Y
0	0
1	1

5.3. FLASH_FREEZE (Ask a Question)

The Flash_Freeze macro is a special-purpose macro that provides information on when the chip is about to go into Flash* Freeze mode to allow the user to perform any housekeeping needed before the device enters into Flash*Freeze mode. The macro has two outputs:

- FF_TO_START—This signal goes high when the FPGA is about to go into Flash*Freeze mode.

- FF_DONE—This signal goes high when the FPGA has successfully entered Flash*Freeze mode.

Figure 5-3. FLASH_FREEZE

For more information about this macro, see the UG0450: SmartFusion2 and IGLOO2 System Controller User Guide and the SmartFusion2 and IGLOO2 FPGA Low-Power Design User Guide.

There is no simulation model for this macro. The two outputs remain low during simulation because Flash-Freeze is not supported during simulation.

5.4. LIVE_PROBE_FB (Ask a Question)

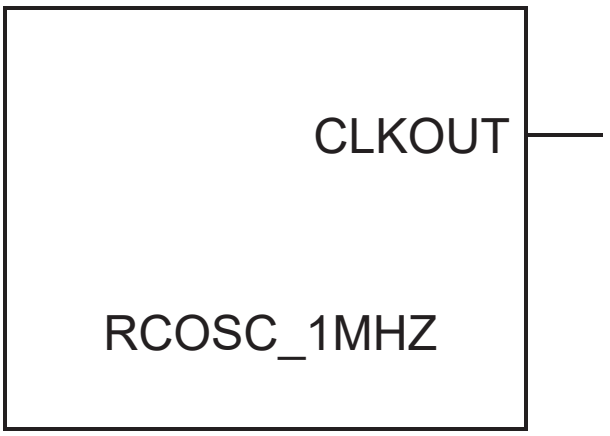
This is a special-purpose macro that feeds the live probe signals back to the fabric. You can connect the PROBE_A/PROBE_B signals to any unused I/O during design generation. This is useful if PROBE_A/PROBE_B cannot be brought out for debugging due to board limitations. PROBE_A and PROBE_B pins must be reserved, if LIVE_PROBE_FB macro is used. This macro is not supported in simulation.

Figure 5-4. LIVE_PROBE_FB

5.5. RCOSC_1MHZ (Ask a Question)

The RCOSC_1 MHZ oscillator is an RC oscillator that provides a free running clock of 1 MHz frequency.

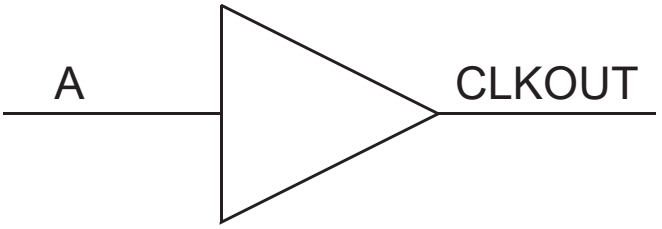
Figure 5-5. RCOSC_1 MHZ



5.6. RCOSC_1MHZ_FAB (Ask a Question)

The RCOSC_1 MHZ_FAB macro provides an interface from the RCOSC_1 MHZ oscillator to fabric logic.

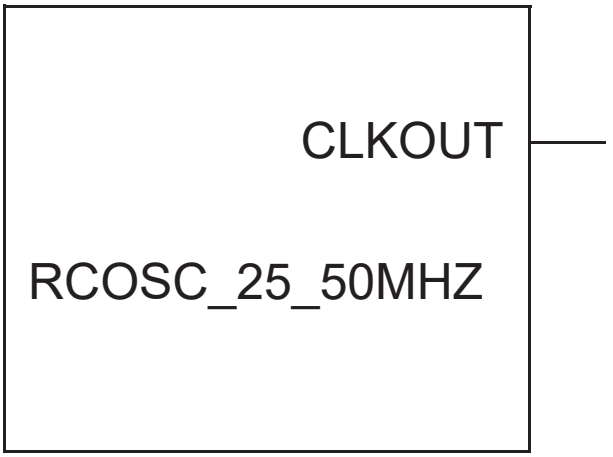
Figure 5-6. RCOSC_1 MHZ_FAB



5.7. RCOSC_25_50MHZ (Ask a Question)

The RCOSC_25_50 MHZ oscillator is an RC oscillator that provides a free running clock of 25 MHz (at 1.0V supply voltage) or 50 MHz (at 1.2V supply voltage).

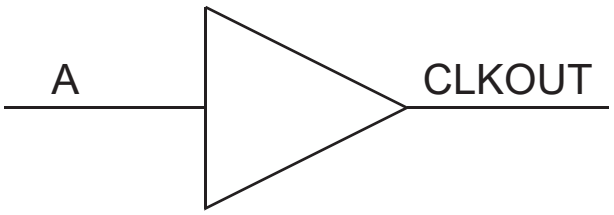
Figure 5-7. RCOSC_25_50MHZ



5.8. RCOSC_25_50 MHZ_FAB (Ask a Question)

The RCOSC_25_50MHZ_FAB macro provides an interface from the RCOSC_25_50MHZ oscillator to fabric logic.

Figure 5-8. RCOSC_25_50 MHZ_FAB



5.9. SYSCTRL_RESET_STATUS (Ask a Question)

This special-purpose macro checks the status of the System Controller. The RESET_STATUS output port goes high when the System Controller is in reset mode. To enable this, select the "System Controller Suspend Mode" option in the "Device Settings" under Libero's Project Settings.

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.
- X[1] µs after observing `POWER_ON_RESET_N` = 1, set `RESET_STATUS` = 1 to indicate that the system controller has entered suspend mode.

[1]The current supported simulation is a fast simulation. In this environment X = 12.12 µs. The current supported simulation is a fast simulation where X = 12.12 µs. For more information about the system-level simulations that include power supply ramp time and actual delay times, see power-up to functional times in the DS0128: IGLOO2 and SmartFusion2 Datasheet. In this environment, X = $T_{VDD2OUT}$ + 5 µs or X = $T_{DEVRST2OUT}$ + 5 µs.

Figure 5-9. SYSCTRL_RESET_STATUS

5.10. SYSRESET (Ask a Question)

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

Figure 5-10. SYSRESET

Table 5-5. SYSRESET I/O

Input	Output
DEVRST_N	POWER_ON_RESET_N

Table 5-6. SYSRESET TRUTH TABLE

DEVRST_N	POWER_ON_RESET_N
0	0
1	1

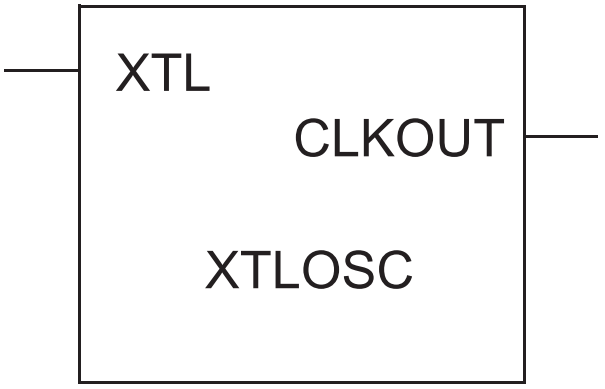
5.11. XTLOSC (Ask a Question)

The crystal oscillator provides up to a 20 MHz clock signal. Physically, it requires connection to an external crystal. However, for simulation purposes the XTL pin provides a clock signal running at the desired input frequency. MODE is a two-bit configuration parameter that specifies the frequency range, as shown in the following table.

Table 5-7. OPERATING MODES

MODE[1:0]	Frequency Range (MHz)
00	N/A
01	0.032 – 0.075
10	0.075 – 2.0
11	2.0 – 20.0

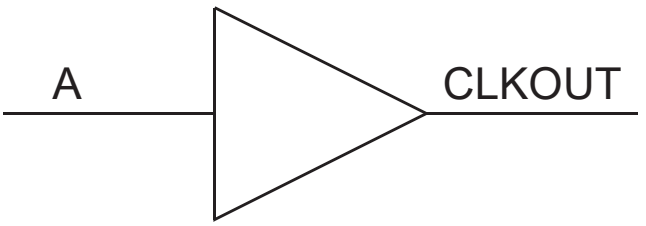
Figure 5-11. XTLOSC



5.12. XTLOSC_FAB (Ask a Question)

The XTL_OSC_FAB macro provides an interface from the crystal oscillator (XTLOSC) to fabric logic.

Figure 5-12. XTLOSC_FAB



6. RAM1K18 (Ask a Question)

The RAM1K18 block contains 18,432 memory bits and is a true dual-port memory. It can also be configured in two-port mode. All read/write operations to the RAM1K18 memory are synchronous. To improve the read data delay, an optional pipeline register at the output is available. A feed-through write mode is also available to enable immediate access to the write data. The RAM1K18 memory has two data ports, which can be independently configured in any combination as follows:

1. Dual-Port RAM with the following configurations:
   - 1Kx18, 1Kx16
   - 2Kx9, 2Kx8
   - 4Kx4
   - 8Kx2
   - 16Kx1
2. Two-Port RAM with the following configurations:
   - 512x36, 512x32
   - 1Kx18, 1Kx16
   - 2Kx9, 2Kx8
   - 4Kx4
   - 8Kx2
   - 16Kx1

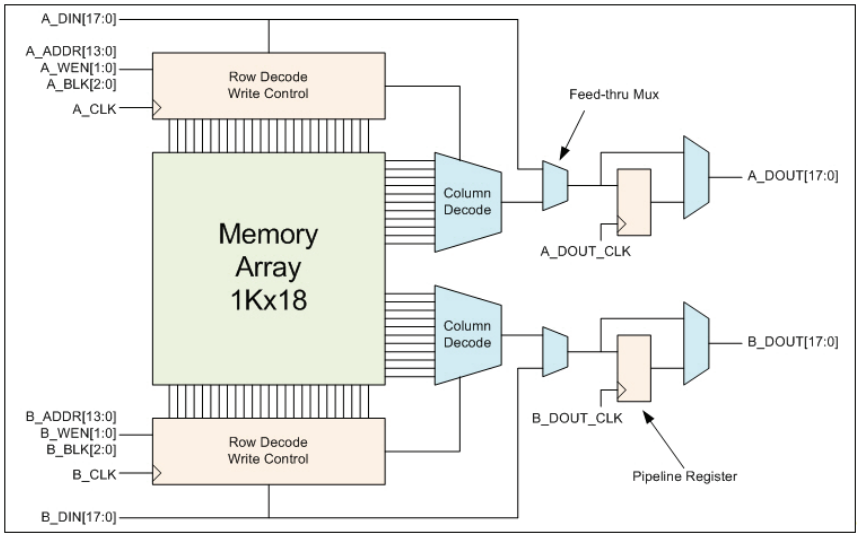
The main features of the RAM1K18 memory block are as follows:

- It has 18,432 bits.
- It provides two independent data ports A and B.
- It has a true dual-port mode, for which both ports have word widths less than or equal to 18 bits.
- In true dual-port mode, each port can be independently configured to any of the following depth/width: 1Kx18, 1Kx16, 2Kx9, 2Kx8, 4Kx4, 8Kx2, and 16Kx1.
- The widths of each port can be different, but one needs to be a multiple of the other. There are 29 unique combinations of true dual-port aspect ratios:
  - 1Kx18/1Kx18, 1Kx18/2Kx9
  - 1Kx16/1Kx16, 1Kx16/2Kx8, 1Kx16/4Kx4, 1Kx16/8Kx2, 1Kx16/16Kx1
  - 2Kx9/1Kx18, 2Kx9/2Kx9
  - 2Kx8/1Kx16, 2Kx8/2Kx8, 2Kx8/4Kx4, 2Kx8/8Kx2, 2Kx8/16Kx1
  - 4Kx4/1Kx16, 4Kx4/2Kx8, 4Kx4/4Kx4, 4Kx4/8Kx2, 4Kx4/16Kx1
  - 8Kx2/1Kx16, 8Kx2/2Kx8, 8Kx2/4Kx4, 8Kx2/8Kx2, 8Kx2/16Kx1
  - 16Kx1/1Kx16, 16Kx1/2Kx8, 16Kx1/4Kx4, 16Kx1/8Kx2, 16Kx1/16Kx1
- RAM1K18 also has a two-port mode. In this case, Port A becomes the read port and Port B becomes the write port.
- In two-port mode, each port can be independently configured to any of the following depth/width: 512x36, 512x32, 1Kx18, 1Kx16, 2Kx9, 2Kx8, 4Kx4, 8Kx2 and 16Kx1.
- The widths of each port can be different, but one needs to be a multiple of the other. There are 45 unique combinations of two-port aspect ratios:
  - 512x36/512x36, 512x36/1Kx18, 512x36/2Kx9
  - 512x32/512x32, 512x32/1Kx16, 512x32/2Kx8, 512x32/4Kx4, 512x32/8Kx2, 512x32/16Kx1

  - 1Kx18/512x36, 1Kx18/1Kx18, 1Kx18/2Kx9
  - 1Kx16/512x32, 1Kx16/1Kx16, 1Kx16/2Kx8, 1Kx16/4Kx4, 1Kx16/8Kx2, 1Kx16/16Kx1
  - 2Kx9/512x36, 2Kx9/1Kx18, 2Kx9/2Kx9
  - 2Kx8/512x32, 2Kx8/1Kx16, 2Kx8/2Kx8, 2Kx8/4Kx4, 2Kx8/8Kx2, 2Kx8/16Kx1
  - 4Kx4/512x32, 4Kx4/1Kx16, 4Kx4/2Kx8, 4Kx4/4Kx4, 4Kx4/8Kx2, 4Kx4/16Kx1
  - 8Kx2/512x32, 8Kx2/1Kx16, 8Kx2/2Kx8, 8Kx2/4Kx4, 8Kx2/8Kx2, 8Kx2/16Kx1
  - 16Kx1/512x32, 16Kx1/1Kx16, 16Kx1/2Kx8, 16Kx1/4Kx4, 16Kx1/8Kx2, 16Kx1/16Kx1
- RAM1K18 performs synchronous operation for setting up the address as well as writing and reading the data. The address, data, block port select, and write enable inputs are registered.
- An optional pipeline register with a separate enable, synchronous-reset, and asynchronous-reset is available at the read data port to improve the clock-to-out delay.
- There is an independent clock for each port. The memory is triggered at the rising edge of the clock.
- The true dual-port mode supports an optional feed-through write mode, where the write data also appears on the corresponding read data port.
- 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.

The following simplified block diagram illustrates the two independent data ports, the pipeline registers, and the feed-through multiplexors.

Figure 6-1. SIMPLIFIED BLOCK DIAGRAM OF RAM1K18



The following table shows a simplified block diagram of the RAM1K18 memory block and gives the port descriptions.

Table 6-1. PORT RAM LIST FOR RAM1K18

Pin Name	Pin Direction	Type	Description	Polarity
A_ADDR[13:0]	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_WIDTH[2:0]	Input	Static	Port A width/depth mode select	—
A_WMODE	Input	Static	Port A feed-through write select	High
A_ARST_N	Input	Dynamic	Port A reset (must be tied to 1)	Low
A_DOUT_LAT	Input	Static	Port A pipeline register select	Low
A_DOUT_ARST_N	Input	Dynamic	Port A pipeline register asynchronous reset	Low
A_DOUT_CLK	Input	Dynamic	Port A pipeline register clock (must be tied to A_CLK or 1)	Rising
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[13:0]	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_WIDTH[2:0]	Input	Static	Port B width/depth mode select	—
B_WMODE	Input	Static	Port B Feed-through write select	High
B_ARST_N	Input	Dynamic	Port B reset (must be tied to 1)	Low
B_DOUT_LAT	Input	Static	Port B pipeline register select	Low
B_DOUT_ARST_N	Input	Dynamic	Port B pipeline register asynchronous reset	Low
B_DOUT_CLK	Input	Dynamic	Port B pipeline register clock (must be tied to B_CLK or 1)	Rising
B_DOUT_EN	Input	Dynamic	Port B pipeline register enable	High
B_DOUT_SRST_N	Input	Dynamic	Port B pipeline register synchronous reset	Low
A_EN	Input	Static	Port A power-down (must be tied to 1)	Low
B_EN	Input	Static	Port B power-down (must be tied to 1)	Low
SII_LOCK	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.

Signal Descriptions for RAM1K18

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. Also, when the write width is 36, the read width must also be 36.

Table 6-2. WIDTH/DEPTH MODE SELECTION

Depth x Width	A_WIDTH/B_WIDTH
16Kx1	000
8Kx2	001
4Kx4	010
2Kx8, 2Kx9	011
1Kx16, 1Kx18	100
512x32, 512x36 (Two-port)	101 11x

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. Also, when the write width is 36, both A_WEN and B_WEN must be static.

Table 6-3. WRITE/READ OPERATION SELECT

Depth x Width	A_WEN/B_WEN	Result
16Kx1, 8Kx2, 4Kx4, 2Kx8, 2Kx9, 1Kx16, 1Kx18	00	Perform a read operation
16Kx1, 8Kx2, 4Kx4, 2Kx8, 2Kx9	01	Perform a write operation
1Kx16	01	Write [7:0]
	10	Write [16:9]
	11	Write [16:9], [7:0]
1Kx18	01	Write [8:0]
	10	Write [17:9]
	11	Write [17:0]
512x32 (Two-port write)	B_WEN[0] = 1	Write B_DIN[7:0]
	B_WEN[1] = 1	Write B_DIN[16:9]
	A_WEN[0] = 1	Write A_DIN[7:0]
	A_WEN[1] = 1	Write A_DIN[16:9]
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 address buses for the two ports. Fourteen bits are needed to address the 16K independent locations in x1 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 6-4. ADDRESS BUS USED AND UNUSED BITS

Depth x Width	A_ADDR/B_ADDR	
	Used Bits	Unused Bits **(must be tied to 0)**
16Kx1	[13:0]	None
8Kx2	[13:1]	[0]
4Kx4	[13:2]	[1:0]

Table 6-4. ADDRESS BUS USED AND UNUSED BITS (continued)

Depth x Width	A_ADDR/B_ADDR	
	Used Bits	Unused Bits (must be tied to 0)
2Kx8, 2Kx9	[13:3]	[2:0]
1Kx16, 1Kx18	[13:4]	[3:0]
512x32, 512x36 (Two-port)	[13:5]	[4:0]

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 6-5. DATA INPUT BUSES USED AND UNUSED BITS

Depth x Width	A_DIN/B_DIN	
	Used Bits	Unused Bits (must be tied to 0)
16Kx1	[0]	[17:1]
8Kx2	[1:0]	[17:2]
4Kx4	[3:0]	[17:4]
2Kx8	[7:0]	[17:8]
2Kx9	[8:0]	[17:9]
1Kx16	[16:9] is [15:8] [7:0] is [7:0]	[17] [8]
1Kx18	[17:0]	None
512x32 (Two-port write)	A_DIN[16:9] is [31:24] A_DIN[7:0] is [23:16] B_DIN[16:9] is [15:8] B_DIN[7:0] is [7:0]	A_DIN[17] A_DIN[8] B_DIN[17] B_DIN[8]
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 6-6. DATA OUTPUT BUSES USED AND UNUSED BITS

Depth x Width	A_DOUT/B_DOUT	
	Used Bits	Unused Bits
16Kx1	[0]	[17:1]
8Kx2	[1:0]	[17:2]
4Kx4	[3:0]	[17:4]
2Kx8	[7:0]	[17:8]
2Kx9	[8:0]	[17:9]
1Kx16	[16:9] is [15:8] [7:0] is [7:0]	[17] [8]
1Kx18	[17:0]	None

Table 6-6. DATA OUTPUT BUSES USED AND UNUSED BITS (continued)

Depth x Width	A_DOUT/B_DOUT	
	Used Bits	Unused Bits
512x32 (Two-port read)	A_DOUT[16:9] is [31:24] A_DOUT[7:0] is [23:16] B_DOUT[16:9] is [15:8] B_DOUT[7:0] is [7:0]	A_DOUT[17] A_DOUT[8] B_DOUT[17] B_DOUT[8]
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 6-7. 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.
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

In true dual-port write mode, each port has a feed-through write option:

- Logic 0 = Read data port holds the previous value.
- Logic 1 = Feed-through, that is, write data appears on the corresponding read data port. This setting is invalid when the width of at least one port is 36 and the two-port mode is in effect.

A_CLK AND B_CLK

All signals in ports A and B are synchronous to the corresponding port clock. All address, data, block port select and write 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_DOUT_LAT and B_DOUT_LAT
- A_DOUT_CLK and B_DOUT_CLK
- A_DOUT_ARST_N and B_DOUT_ARST_N
- A_DOUT_EN and B_DOUT_EN
- A_DOUT_SRST_N and B_DOUT_SRST_N

The A_DOUT_LAT and B_DOUT_LAT signals select the pipeline registers for the respective port. 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 pipeline registers have rising edge clock inputs for each port, which must be tied to the respective port clock when used. When the pipeline registers are not being used, they are forced into latch mode and the clock signals must be tied to 1, which makes them transparent.

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

Table 6-8. TRUTH TABLE FOR A_DOUT AND B_DOUT REGISTERS

_ARST_N	_LAT	_CLK	_EN	_SRST_N	D	Q_{n+1}
0	X	X	X	X	X	0
1	0	Not rising	X	X	X	Q_n
1	0	—	0	X	X	Q_n
1	0	—	1	0	X	0
1	0	—	1	1	D	D
1	1	0	X	X	X	Q_n
1	1	1	0	X	X	Q_n
1	1	1	1	0	X	0
1	1	1	1	1	D	D

A_EN AND B_EN

These are active-low, and power-down configuration bits for each port. They must be tied to 1.

A_ARST_N AND B_ARST_N

Always tie these signals to 1.

SII_LOCK

Control signal, when 1 locks the entire RAM1K18 memory from being accessed by the SII.

BUSY

This output indicates that the RAM1K18 memory is being accessed by the SII.

RAM64X18

The RAM64x18 block contains 1,152 memory bits and is a three-port memory providing one write port and two read ports. Write operations to the RAM64x18 memory are synchronous. Read operations can be asynchronous or synchronous for either setting up the address and/or 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 memory can be independently configured in any combination as shown in the following list:

- 64x18, 64x16
- 128x9, 128x8
- 256x4
- 512x2
- 1Kx1

The main features of the RAM64x18 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, synchronous-reset, and asynchronous-reset for synchronous mode operation, which can also be configured to be transparent latches for asynchronous mode operation.

- The two read data ports have output registers with a separate enable, synchronous-reset, and asynchronous-reset for pipeline mode operation, which can also be configured to be transparent latches for asynchronous mode operation. Therefore, there are four read operation modes for ports A and B:
  - Synchronous read address without pipeline registers (sync-async).
  - Synchronous read address with pipeline registers (sync-sync).
  - Asynchronous read address without pipeline registers (async-async).
  - Asynchronous read address with pipeline registers (async-sync).
- Each data port on the RAM64x18 memory can be independently configured in any of the following combinations: 64x18, 64x16, 128x9, 128x8, 256x4, 512x2, and 1Kx1.
- The widths of each port can be different, but they need to be multiples of one another.
- There is an independent clock for each port. The memory is 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.

The following figure shows a simplified block diagram of the RAM64x18 memory block and the following table gives the port descriptions. The simplified block diagram illustrates the three independent read/write ports and the pipeline registers on the read port.

Figure 6-2. SIMPLIFIED BLOCK DIAGRAM OF RAM64X18

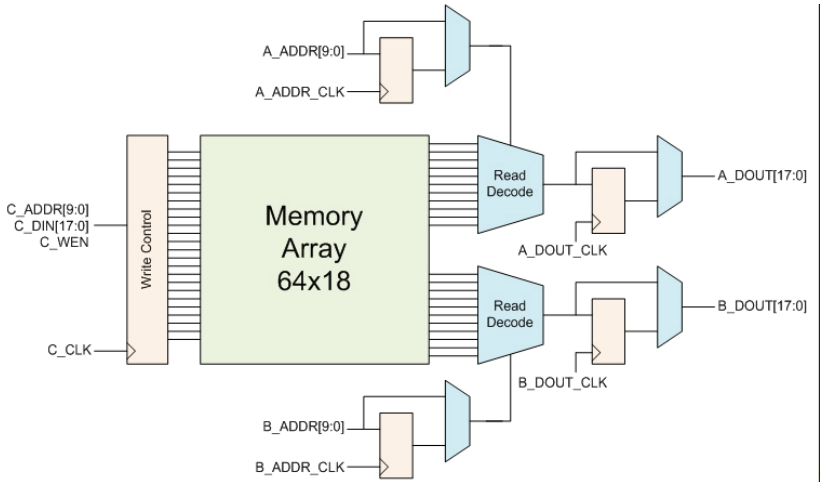


Table 6-9. PORT LIST FOR RAM64X18

Pin Name	Pin Direction	Type	Description	Polarity
A_ADDR[9:0]	Input	Dynamic	Port A address	—
A_BLK[1:0]	Input	Dynamic	Port A block selects	High
A_WIDTH[2:0]	Input	Static	Port A width/depth mode selection	—
A_DOUT[17:0]	Output	Dynamic	Port A read data	—
A_DOUT_ARST_N	Input	Dynamic	Port A pipeline register asynchronous reset	Low
A_DOUT_CLK	Input	Dynamic	Port A pipeline register clock	Rising
A_DOUT_EN	Input	Dynamic	Port A pipeline register enable	High
A_DOUT_LAT	Input	Static	Port A pipeline register select	Low
A_DOUT_SRST_N	Input	Dynamic	Port A pipeline register synchronous reset	Low

Table 6-9. PORT LIST FOR RAM64X18 (continued)

Pin Name	Pin Direction	Type	Description	Polarity
A_ADDR_CLK	Input	Dynamic	Port A address register clock	Rising
A_ADDR_EN	Input	Dynamic	Port A address register enable	High
A_ADDR_LAT	Input	Static	Port A address register select	Low
A_ADDR_SRST_N	Input	Dynamic	Port A address register synchronous reset	Low
A_ADDR_ARST_N	Input	Dynamic	Port A address register asynchronous reset	Low
B_ADDR[9:0]	Input	Dynamic	Port B address	—
B_BLK[1:0]	Input	Dynamic	Port B block selects	High
B_WIDTH[2:0]	Input	Static	Port B width/depth mode selection	—
B_DOUT[17:0]	Output	Dynamic	Port B read data	—
B_DOUT_ARST_N	Input	Dynamic	Port B pipeline register asynchronous reset	Low
B_DOUT_CLK	Input	Dynamic	Port B pipeline register clock	Rising
B_DOUT_EN	Input	Dynamic	Port B pipeline register enable	High
B_DOUT_LAT	Input	Static	Port B pipeline register select	Low
B_DOUT_SRST_N	Input	Dynamic	Port B pipeline register synchronous reset	Low
B_ADDR_CLK	Input	Dynamic	Port B address register clock	Rising
B_ADDR_EN	Input	Dynamic	Port B address register enable	High
B_ADDR_LAT	Input	Static	Port B address register select	Low
B_ADDR_SRST_N	Input	Dynamic	Port B address register synchronous reset	Low
B_ADDR_ARST_N	Input	Dynamic	Port B address register asynchronous reset	Low
C_ADDR[9:0]	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[2:0]	Input	Static	Port C width/depth mode selection	—
A_EN	Input	Static	Port A power-down (must be tied to 1)	Low
B_EN	Input	Static	Port B power-down (must be tied to 1)	Low
C_EN	Input	Static	Port C power-down (must be tied to 1)	Low
SII_LOCK	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.

Signal Descriptions for RAM64x18

A_WIDTH, B_WIDTH AND C_WIDTH

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

Table 6-10. WIDTH/DEPTH MODE SELECTION

Depth x Width	A_WIDTH/B_WIDTH/C_WIDTH
1Kx1	000
512x2	001
256x4	010
128x8, 128x9	011
64x16, 64x18	1xx

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. 10 bits are required to address 1K independent locations in x1 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 6-11. ADDRESS BUSES USED AND UNUSED BITS

Depth x Width	A_ADDR/B_ADDR/C_ADDR	
	Used Bits	Unused Bits (must be tied to zero)
1Kx1	[9:0]	None
512x2	[9:1]	[0]
256x4	[9:2]	[1:0]
128x8, 128x9	[9:3]	[2:0]
64x16, 64x18	[9:4]	[3:0]

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 6-12. DATA INPUT BUS USED AND UNUSED BITS

Depth x Width	C_DIN	
	Used Bits	Unused Bits (must be tied to 0)
1Kx1	[0]	[17:1]
512x2	[1:0]	[17:2]
256x4	[3:0]	[17:4]
128x8	[7:0]	[17:8]
128x9	[8:0]	[17:9]
64x16	[16:9] [7:0]	[17] [8]
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 6-13. DATA OUTPUT USED AND UNUSED BITS

Depth x Width	A_DOUT/B_DOUT	
	Used Bits	Unused Bits
1Kx1	[0]	[17:1]
512x2	[1:0]	[17:2]
256x4	[3:0]	[17:4]
128x8	[7:0]	[17:8]
128x9	[8:0]	[17:9]
64x16	[16:9] [7:0]	[17] [8]
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 6-14. 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 are forced to zero.
	11	Perform read operation from port A.
B_BLK[1:0]	Any one bit is 0	Port B is not selected and its read data are 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 setup before the rising edge of the clock. The write operation begins with the rising edge:

- A_DOUT_LAT, A_ADDR_LAT, B_DOUT_LAT, and B_ADDR_LAT
- A_DOUT_CLK, A_ADDR_CLK, B_DOUT_CLK, and B_ADDR_CLK
- A_DOUT_ARST_N, A_ADDR_ARST_N, B_DOUT_ARST_N, and B_ADDR_ARST_N
- 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 _LAT signals select the registers for the respective port.

The address and pipeline registers have rising edge clock inputs for ports A and B. When both the address and pipeline registers for a port are in use, their clock signals must be tied together. When the registers are not being used, they are forced into latch mode and the clock signals must be tied to 1, which makes them transparent.

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

Table 6-15. TRUTH TABLE FOR A_ADDR, B_ADDR, A_DOUT, AND B_DOUT REGISTERS

_ARST_N	_LAT	_CLK	_EN	_SRST_N	D	Q_{n+1}
0	X	X	X	X	X	0
1	0	Not rising	X	X	X	Q_n
1	0	—	0	X	X	Q_n
1	0	—	1	0	X	0
1	0	—	1	1	D	D
1	1	0	X	X	X	Q_n
1	1	1	0	X	X	Q_n
1	1	1	1	0	X	0
1	1	1	1	1	D	D

A_EN, B_EN AND C_EN

Active-Low, power-down configuration bits for each port. They must be tied to 1.

SII_LOCK

Control signal, when 1 locks the entire RAM64X18 memory from being accessed by the SII.

BUSY

Output indicates that the RAM64X18 memory is being accessed by the SII.

7. 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 7-1. MACC PORTS

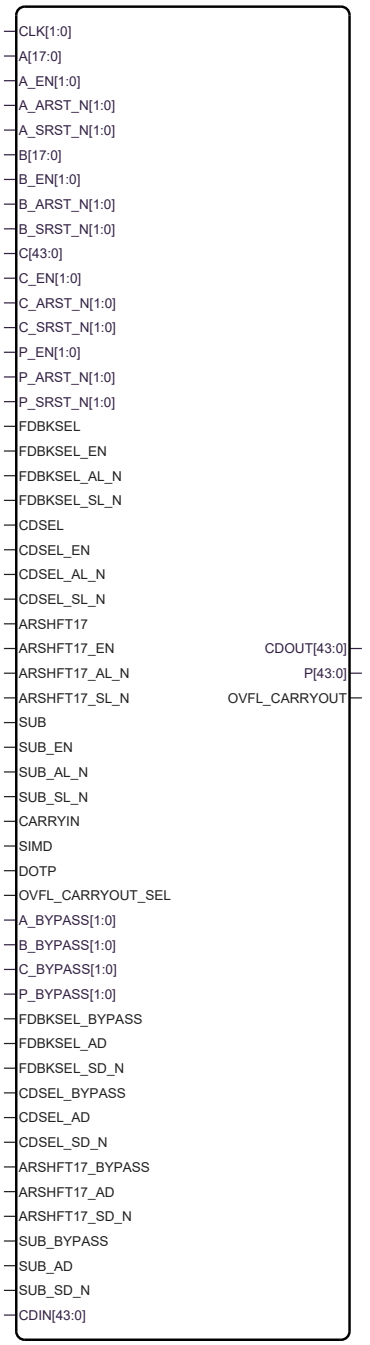


Table 7-1. PORTS

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.
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: • A_ARST_N[1] is for A[17:9]. Connect to 1, if not registered. • A_ARST_N[0] is for A[8:0]. Connect to 1, if not registered. In normal mode, ensure A_ARST_N[1] = A_ARST_N[0].
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].

Table 7-1. PORTS (continued)

Port Name	Direction	Type	Polarity	Description
B_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for data B registers: • B_ARST_N[1] is for B[17:9]. Connect to 1, if not registered. • B_ARST_N[0] is for B[8:0]. Connect to 1, if not registered. In normal mode, ensure B_ARST_N[1] = B_ARST_N[0].
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] See Table 7-4 to see how operand D is obtained from P, CDIN or 0.
OVFL_CARRYOUT	Output	—	High	Overflow or CarryOut: • Overflow when OVFL_CARRYOUT_SEL = 0 OVFL_CARRYOUT = (SUM[45] ^ SUM[44]) | (SUM[44] ^ SUM[43]) • CarryOut when OVFL_CARRYOUT_SEL = 1 OVFL_CARRYOUT = C[43] ^ D[43] ^ SUM[44]
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].

Table 7-1. PORTS (continued)

Port Name	Direction	Type	Polarity	Description
P_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for result P registers: • P_ARST_N[1] is for P[43:18] and OVFL_CARRYOUT. Connect to 1, if not registered. • P_ARST_N[0] is for P[17:0]. Connect to 1, if not registered. In normal mode, ensure P_ARST_N[1] = P_ARST_N[0].
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].
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 data C registers: • C_ARST_N[1] is for C[43:18] . Connect to 1, if not registered. • C_ARST_N[0] is for C[17:0] and CARRYIN. Connect to 1, if not registered. In normal mode, ensure C_ARST_N[1] = C_ARST_N[0].
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].

Table 7-1. PORTS (continued)

Port Name	Direction	Type	Polarity	Description
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. See Table 7-4 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. See Table 7-4 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, if not registered. 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 7-2.
ARSHFT17_SD_N	Input	Static	Low	Synchronous load data for ARSHFT17 register. See Table 7-2.
ARSHFT17_EN	Input	Dynamic	High	Enable for ARSHFT17 register. Connect to 1, if not registered. See Table 7-2.
CDSEL	Input	Dynamic	High	Select CDIN for operand D. When CDSEL = 1, propagate CDIN. When CDSEL = 0, propagate 0 or P depending on FDBKSEL. See Table 7-2 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, if not registered. 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 7-2.
CDSEL_SD_N	Input	Static	Low	Synchronous load data for CDSEL register. See Table 7-2.
CDSEL_EN	Input	Dynamic	High	Enable for CDSEL register. Connect to 1, if not registered. See Table 7-2.

Table 7-1. PORTS (continued)

Port Name	Direction	Type	Polarity	Description
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. See Table 7-4 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, if not registered. 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 7-2.
FDBKSEL_SD_N	Input	Static	Low	Synchronous load data for FDBKSEL register. See Table 7-2.
FDBKSEL_EN	Input	Dynamic	High	Enable for FDBKSEL register. Connect to 1, if not registered. See Table 7-2.
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, if not registered. 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 7-2.
SUB_SD_N	Input	Static	Low	Synchronous load data for SUB register. See Table 7-2.
SUB_EN	Input	Dynamic	High	Enable for SUB register. Connect to 1, if not registered. See Table 7-2.

Table 7-2. TRUTH TABLE FOR CONTROL REGISTERS ARSHFT17, CDSEL, FDBKSEL, AND SUB

_AL_N	_AD	_BYPASS	_CLK	_EN	_SL_N	_SD_N	D	Q_{n+1}
0	AD	X	X	X	X	X	X	AD
1	X	0	Not rising	X	X	X	X	Q_n
1	X	0	—	0	X	X	X	Q_n
1	X	0	—	1	0	SD_n	X	$!SD_n$
1	X	0	—	1	1	X	D	D
1	X	1	X	0	X	X	X	Q_n
1	X	1	X	1	0	SD_n	X	$!SD_n$
1	X	1	X	1	1	X	D	D

Table 7-3. TRUTH TABLE- DATA REGISTERS A, B, C, CARRYIN, P, AND OVFL_CARRYOUT

_ARST_N	_BYPASS	_CLK	_EN	_SRST_N	D	Q_{n+1}
0	X	X	X	X	X	0
1	0	Not rising	X	X	X	Q_n
1	0	—	0	X	X	Q_n
1	0	—	1	0	X	0
1	0	—	1	1	D	D
1	1	X	0	X	X	Q_n

Table 7-3. TRUTH TABLE- DATA REGISTERS A, B, C, CARRYIN, P, AND OVFL_CARRYOUT (continued)

_ARST_N	_BYPASS	_CLK	_EN	_SRST_N	D	Q_{n+1}
1	1	X	1	0	X	0
1	1	X	1	1	D	D

Table 7-4. 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]}

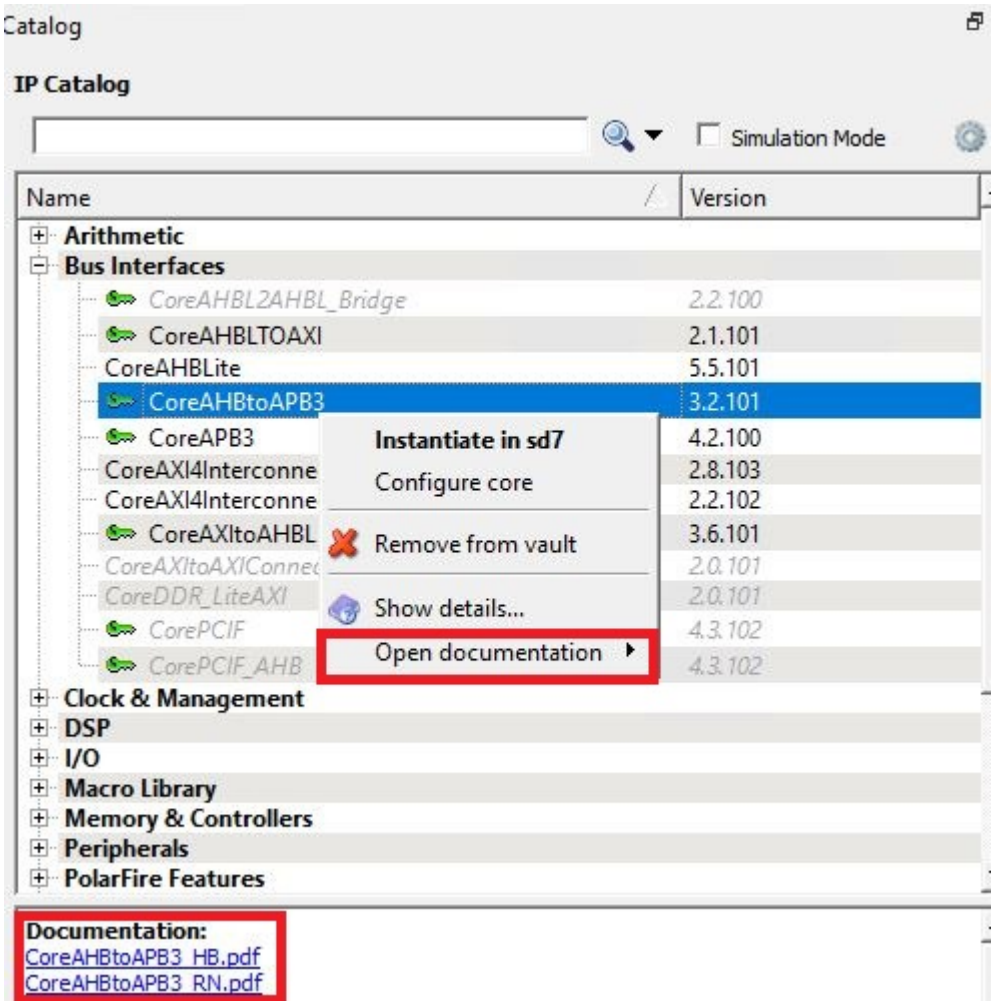
8. Appendix-Accessing IP Core Documentation (Ask a Question)

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 8-1. Accessing the IP Core Documentation



9. 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	The following change is made in this revision: • Updated Sequential Logic.
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	Added Appendix-Accessing IP Core Documentation.
C	08/2021	Editorial updates only. No technical content updates.
B	04/2021	Editorial updates only. No technical content updates.
A	11/2020	The following is a summary of changes made in this revision. • 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: 979-8-3371-1095-0

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.