SmartFusion2 and IGLOO2 Macro Library Guide
For Libero SoC v11.7 SP2

Power Matters.™

Table of Contents - All Macros

Table of Contents - Combinatorial Logic

Table of Contents - Sequential Logic

Table of Contents - RAM Blocks

Table of Contents - Math Blocks

Table of Contents - I/Os

Table of Contents - Clocking

Table of Contents - Special

Introduction

This macro library guide supports the SmartFusion2 and IGLOO2 families. See the Microsemi website for macro library 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 will indicate 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 SmartFusion2 and IGLOO2 families.

As an 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 Figure 1.

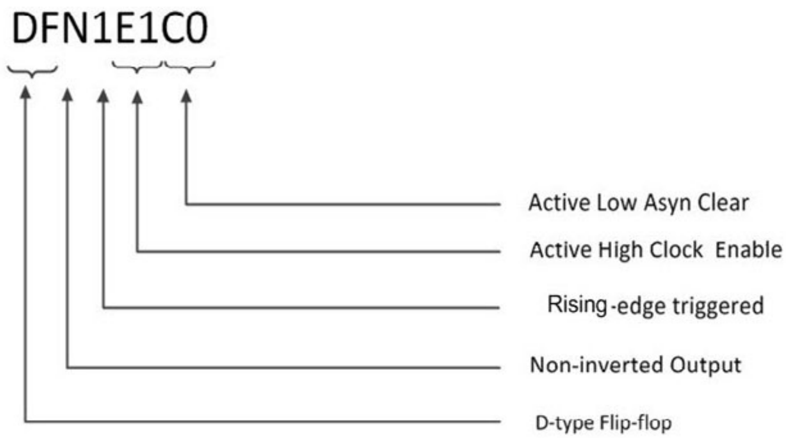

Figure 1 **• Naming Convention**

Truth Table Notation

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

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

AND2

2-Input AND

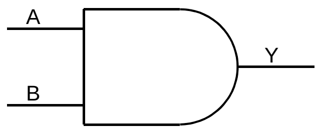

Figure 2 • **AND2**

Inputs	Output
A, B	Y

Truth Table

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

AND3

3-Input AND

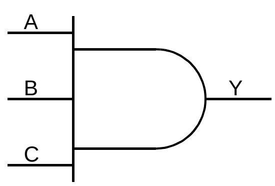

Figure 3 • **AND3**

Input	Output
A, B, C	Y

Truth Table

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

AND4

4-Input AND

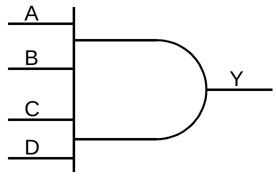

***Figure 4* • AND4**

Input	Output
A, B, C, D	Y

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

CFG1/2/3/4 and LUTs (Look-Up Tables)

CFG1, CFG2, CFG3, and CFG4 are post-layout LUTs (Look-up table) used to 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.

CFG2

Post-layout macro used to implement 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.

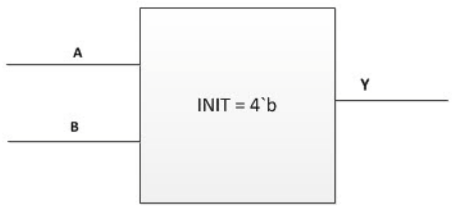

***Figure 5* • CFG2**

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

***Table 1* • CFG2 INIT String InterpretationI**

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

CFG3

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.

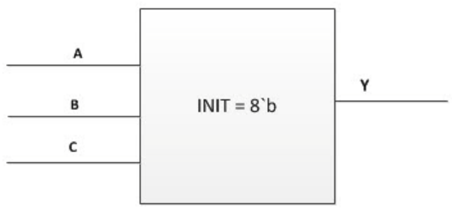

***Figure 6* • CFG3**

.

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

***Table 2* • CFG3 INIT String Interpretation**

CBA	Y
000	INIT[0]
001	INIT[1]
010	INIT[2]
011	INIT[3]
100	INIT[4]
101	INIT[5]
110	INIT[6]
111	INIT[7]

CFG4

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.

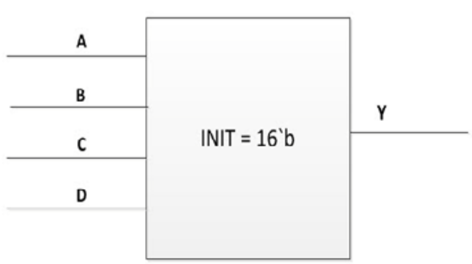

***Figure 7* • CFG4**

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

Table 3 • **CFG4 INIT String Interpretation**

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

BUFF

Buffer

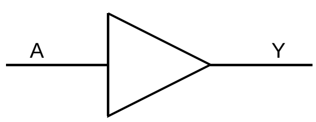

Figure 8 • **BUFF**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

BUFD

Buffer. Note that Compile optimization will not remove this macro.

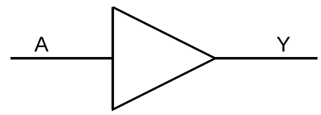

Figure 9 **• BUFD**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

CLKINT

Macro used to route an internal fabric signal to global network.

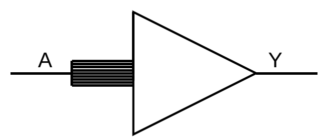

Figure 10 **• CLKINT**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

CLKINT_PRESERVE

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

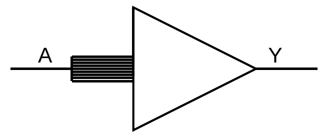

***Figure 11* • CLKINT_PRESERVE**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

GB

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

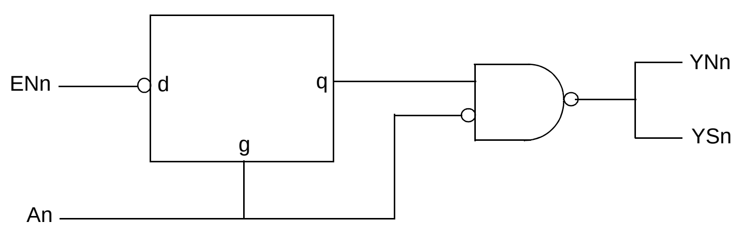

***Figure 12* • GB**

Input	Output
An, ENn	YNn, YSn

Truth Table

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

GBM

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

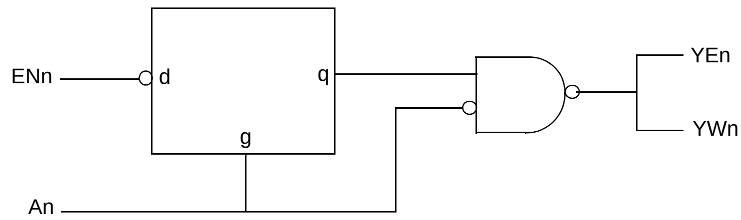

***Figure 13* • GBM**

Input	Output
An, ENn	YEn, YWn

Truth Table

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

GB_NG

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

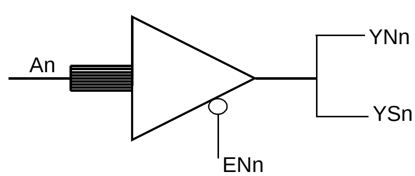

***Figure 14* • GB_NG**

Input	Output
An, ENn	YNn, YSn

Truth Table

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

GBM_NG

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

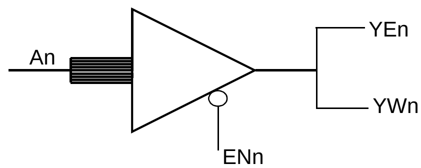

Figure 15 **• GBM_NG**

Input	Output
An, ENn	YEn, YWn

Truth Table

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

GCLKINT

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

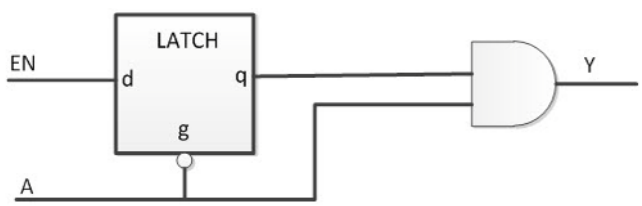

Figure 16 **• GCLKINT**

Input	Output
A, EN	Y

Truth Table

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

RCLKINT

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

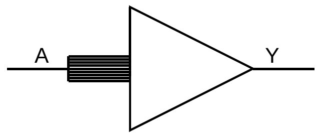

Figure 17 **• RCLKINT**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

RGB

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

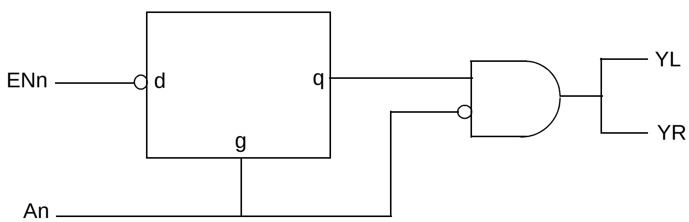

Figure 18 **• RGB**

Input	Output
An, ENn	YL, YR

Truth Table

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

RGB_NG

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

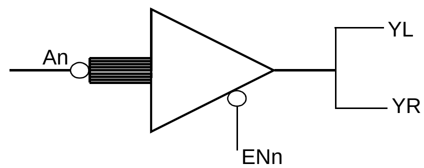

Figure 19 **• RGB_NG**

Input	Output
An, ENn	YL, YR

Truth Table

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

RGCLKINT

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.

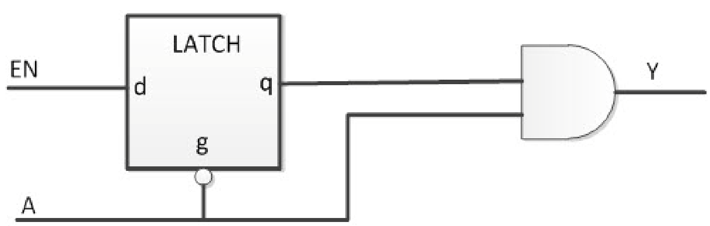

Figure 20 **• RGCLKINT**

Input	Output
A, EN	Y

Truth Table

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

SLE

Sequential Logic Element

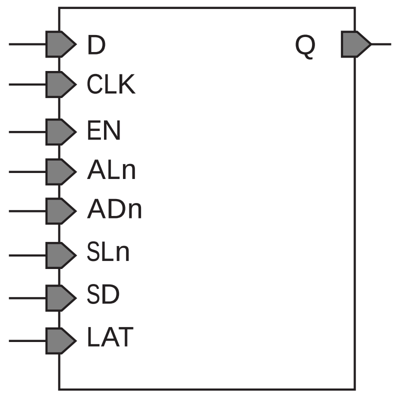

***Figure 21* • SLE**

Input		Output
Name	**Function**	
D	Data	Q
CLK	Clock	
EN	Enable	
ALn	Asynchronous Load (Active Low)	
ADn*	Asynchronous Data (Active Low)	
SLn	Synchronous Load (Active Low)	
SD*	Synchronous Data	
LAT*	Latch Enable	

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

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

ARI1

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

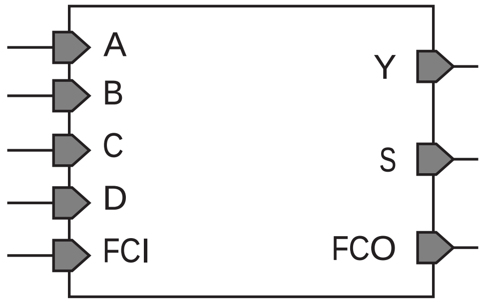

Figure 22 • **ARI1**

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

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

Table 4 **• 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 5 • **Truth Table for S**

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

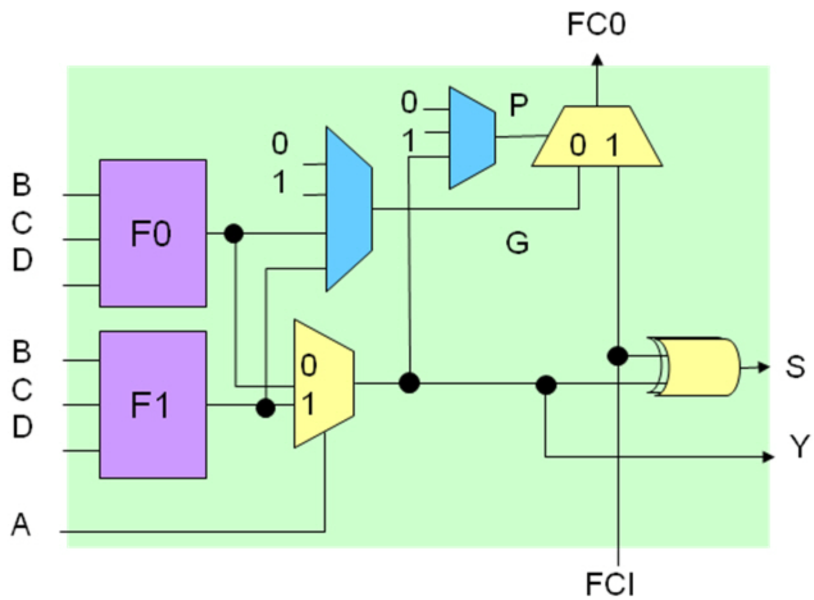

***Figure 23* • ARI1 Logic**

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 tables below.

Table 6 • **ARI1 INIT[17:16] String Interpretation**

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

Table 7 • **ARI1 INIT[19:18] String Interpretation**

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

Table 8 • **FCO Truth Table**

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

ARI1_CC

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 that FC1 and FC0 do not perform any functionalities.

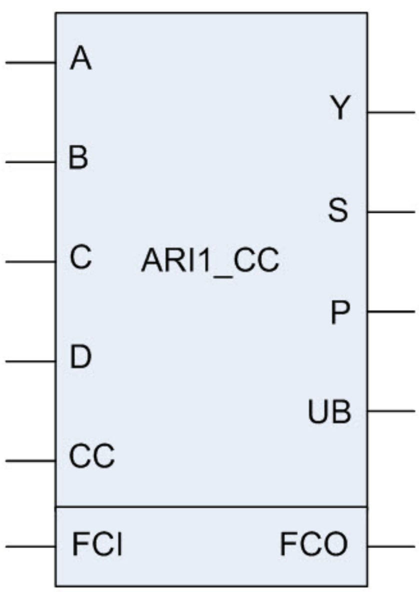

***Figure 24* • ARI1_CC**

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

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

***Table 9* • 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 tables below.

***Table 10* • 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 11* • 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

CC_CONFIG

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

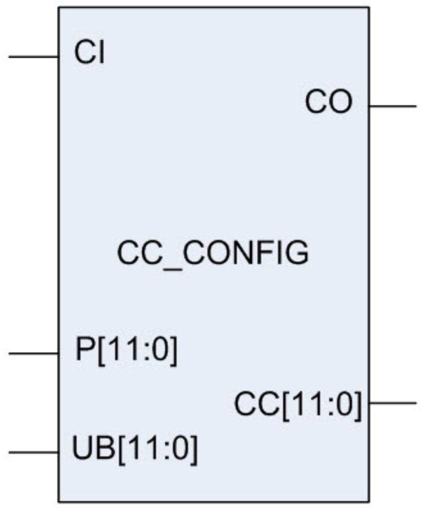

Figure 25 • **CC_Config**

Input	Output
CI, P, UB	CO, CC

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

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

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

Inside every cluster, there are 12 ARI1_CC cells and only one CC_CONFIG cell. The CC_CONFIG takes as input the P and UB outputs of each ARI1_CC cell in the cluster and generates the CC (carry-out bit), which is then fed to the next ARI1_CC cell in the cluster as the carry-in. The connection between the ARI1_CC cells inside the cluster and the CC_CONFIG cell is shown in Figure 26.

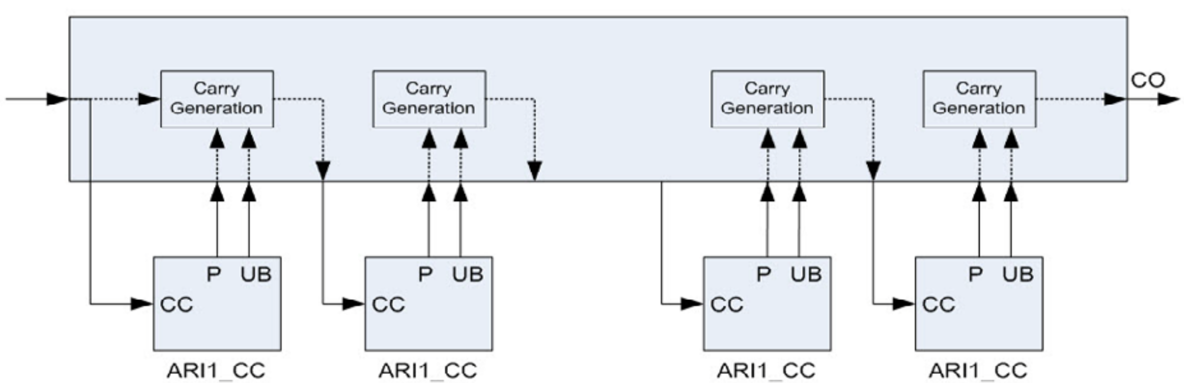

Figure 26 • **CC_CONFIG Connections to ARI1_CC Cells**

FCEND_BUFF

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

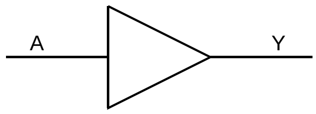

Figure 27 • **FCEND_BUFF**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

FCINIT_BUFF

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

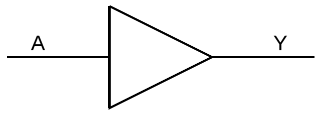

Figure 28 **• FCINIT_BUFF**

Input	Output
A	Y

Truth Table

A	Y
0	0
1	1

FLASH_FREEZE

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 2 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.

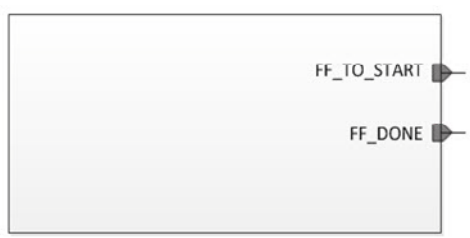

Figure 29 **• FLASH_FREEZE**

For more information about this macro, refer to the System Controller User Guide and the SmartFusion2 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.

OSCILLATOR

The OSC macro is a special-purpose macro. It can be configured as a Crystal Oscillator (XTLOSC), a 25/50 MHz RC Oscillator or a 1MHz RC Oscillator. All three configurations are supported by simulation models.

XTLOSC

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. If the DISABLE input is high, the output is low.

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

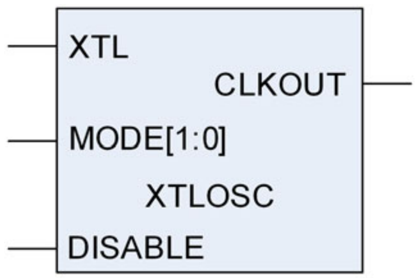

***Figure 30* • XTLOSC**

RCOSC_1MHZ

The RCOSC_1MHz oscillator is an RC oscillator that provides a free running clock of 1MHz frequency. The DISABLE pin is active high and when asserted, it turns off the oscillator output.

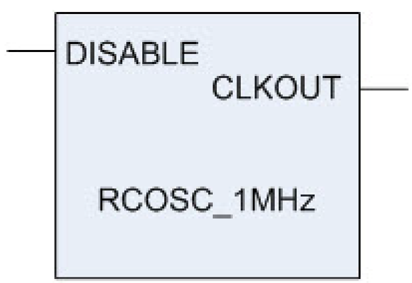

***Figure 31* • RCOSC_1MHz**

RCOSC_25_50MHz

The RCOSC_25_50MHz oscillator is an RC oscillator that provides a free running clock of 25 MHz (at 1.0V supply voltage) or 50MHz (at 1.2V supply voltage). The DISABLE pin is active high and when asserted, it turns off the oscillator output.

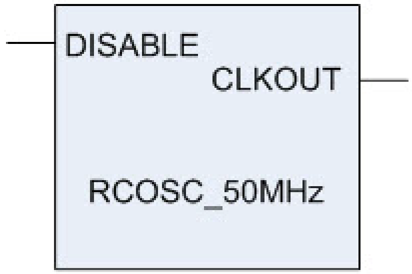

Figure 32 **• RCOSC_25_50MHz**

SYSRESET

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

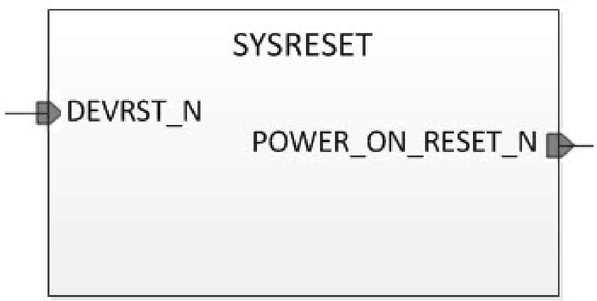

Figure 33 **• SYSRESET**

Input	Output
DEVRST_N	POWER_ON_RESET_N

Truth Table

DEVRST_N	POWER_ON_RESET_N
0	0
1	1

SYSCTRL_RESET_STATUS

This is a special-purpose macro to check the status of the System Controller. The output port RESET_STATUS goes high if the System Controller is in reset ("System Controller Suspend Mode" option is checked in Device Settings under Libero's Project Settings).

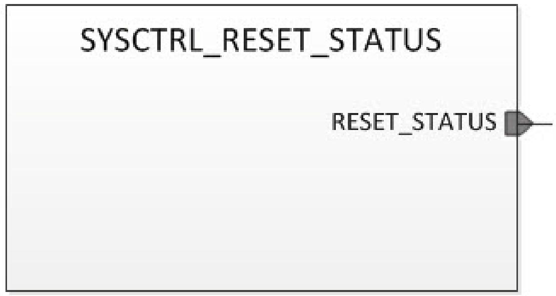

***Figure 34* • SYSCTRL_RESET_STATUS**

This macro is not supported in simulation.

LIVE_PROBE_FB

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 debug due to board limitations.

Note: PROBE_A and PROBE_B pins must be reserved if LIVE_PROBE_FB macro is used.

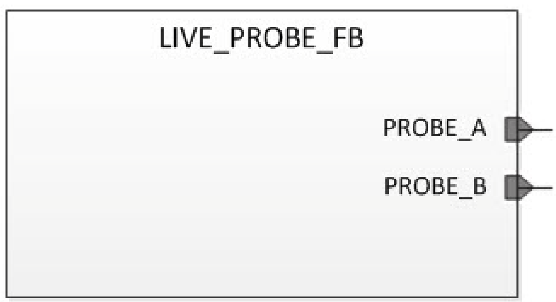

***Figure 35* • LIVE_PROBE_FB**

This macro is not supported in simulation.

GCLKBUF

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

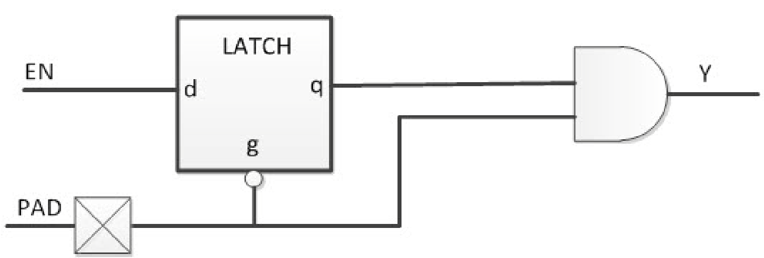

***Figure 36* • GCLKBUF**

Input	Output
PAD, EN	Y

Truth Table

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

GCLKBUF_DIFF

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

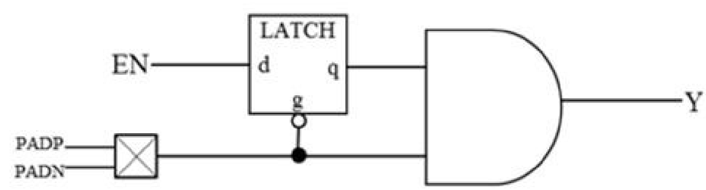

Figure 37 **• GCLKBUF_DIFF**

Differential

Input	Output
PADP, PADN, EN	Y

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

GCLKBIBUF

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

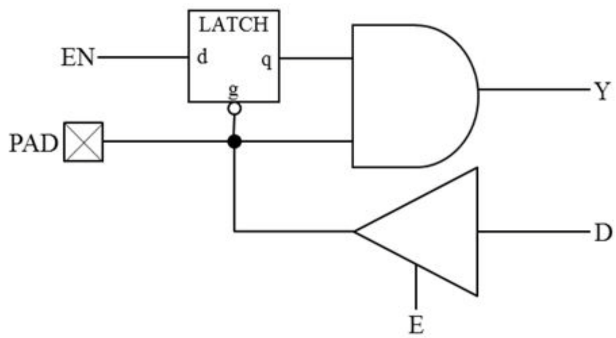

Figure 38 • **GCLKBIBUF**

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

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

DFN1

D-Type Flip-Flop

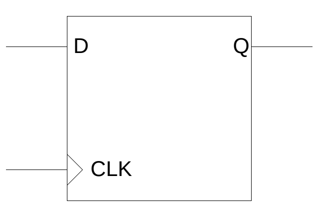

Figure 39 • **DFN1**

Input	Output
D, CLK	Q

Truth Table

CLK	D	Q_{n+1}
not Rising	X	Q_n
↑	D	D

DFN1C0

D-Type Flip-Flop with active low Clear

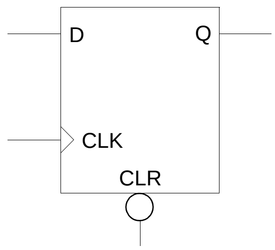

Figure 40 **• DFN1C0**

Input	Output
D, CLK, CLR	Q

Truth Table

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

DFN1E1

D-Type Flip-Flop with active high Enable

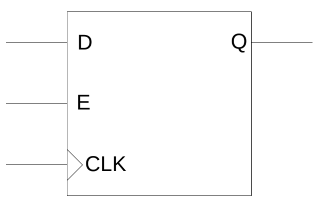

Figure 41 **• DFN1E1**

Input	Output
D, E, CLK	Q

Truth Table

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

DFN1E1C0

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

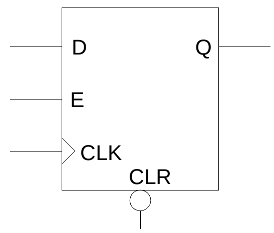

Figure 42 • **DFN1E1C0**

Input	Output
CLR, D, E, CLK	Q

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

DFN1E1P0

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

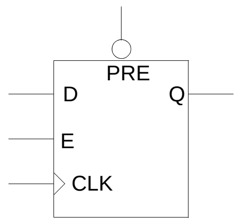

Figure 43 **• DFN1E1P0**

Input	Output
D, E, PRE, CLK	Q

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

DFN1P0

D-Type Flip-Flop with active low Preset.

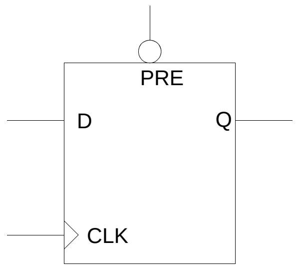

Figure 44 **• DFN1P0**

Input	Output
D, PRE, CLK	Q

Truth Table

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

DLN1

Data Latch

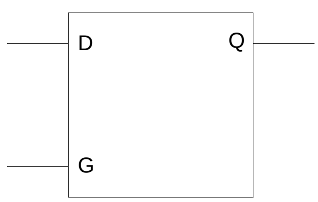

Figure 45 **• DLN1**

Input	Output
D, G	Q

Truth Table

G	D	Q
0	X	Q
1	D	D

DLN1C0

Data Latch with active low Clear

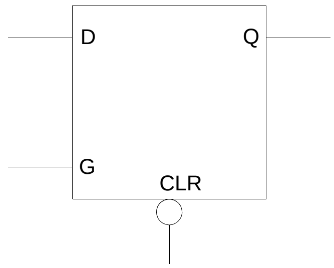

***Figure 46* • DLN1C0**

Input	Output
CLR, D, G	Q

Truth Table

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

DLN1P0

Data Latch with active low Preset

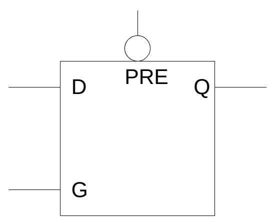

***Figure 47* • DLN1P0**

Input	Output
D, G, PRE	Q

Truth Table

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

INV

Inverter

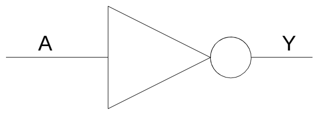

***Figure 48* • INV**

Input	Output
A	Y

Truth Table

A	Y
0	1
1	0

INVD

Inverter; note that Compile optimization will not remove this macro.

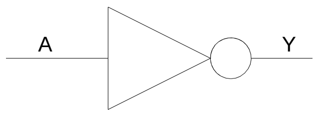

***Figure 49* • INVD**

Input	Output
A	Y

Truth Table

A	Y
0	1
1	0

MX2

2 to 1 Multiplexer

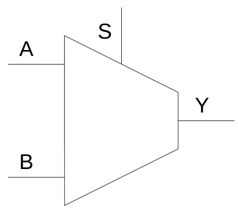

Figure 50 **• MX2**

Input	Output
A, B, S	Y

Truth Table

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

MX4

4 to 1 Multiplexer

This macro uses two logic modules.

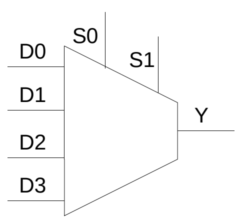

Figure 51 **• MX4**

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

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

NAND2

2-Input NAND

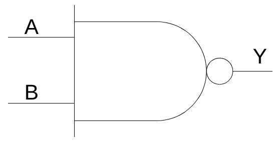

***Figure 52* • NAND2**

Input	Output
A, B	Y

Truth Table

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

NAND3

3-Input NAND

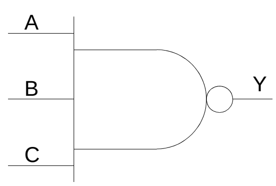

***Figure 53* • NAND3**

Input	Output
A, B, C	Y

Truth Table

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

NAND4

4-input NAND

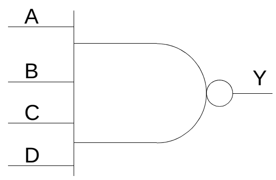

Figure 54 • **NAND4**

Input	Output
A, B, C, D	Y

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

NOR2

2-input NOR

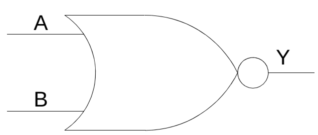

Figure 55 • **NOR2**

Input	Output
A, B	Y

Truth Table

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

NOR3

3-input NOR

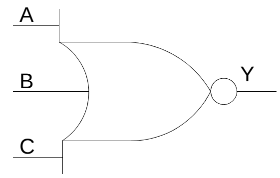

***Figure 56* • NOR3**

Input	Output
A, B, C	Y

Truth Table

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

NOR4

4-input NOR

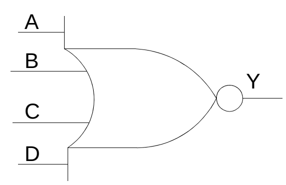

***Figure 57* • NOR4**

Input	Output
A, B, C, D	Y

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

OR2

2-input OR

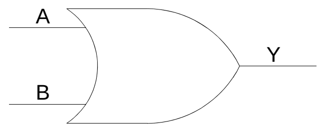

Figure 58 • **OR2**

Input	Output
A, B	Y

Truth Table

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

OR3

3-input OR

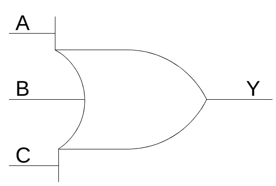

Figure 59 • **OR3**

Input	Output
A, B, C	Y

Truth Table

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

OR4

4-input OR

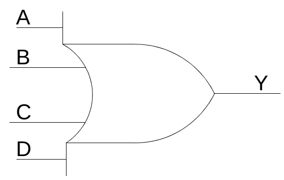

***Figure 60* • OR4**

Input	Output
A, B, C, D	Y

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

XOR2

2-input XOR

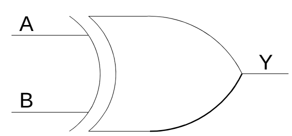

***Figure 61* • XOR2**

Input	Output
A, B	Y

Truth Table

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

XOR3

3-input XOR

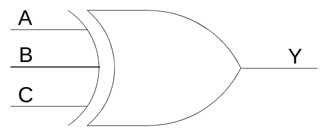

Figure 62 • **XOR3**

Input	Output
A, B, C	Y

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

XOR4

4-input XOR

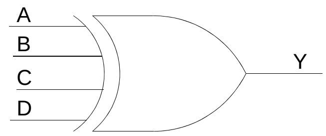

Figure 63 • **XOR4**

Input	Output
A, B, C, D	Y

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

XOR8

8-input XOR

This macro uses two logic modules.

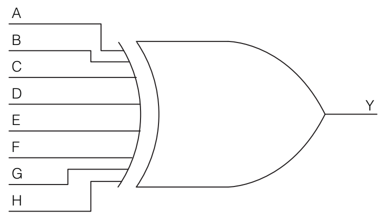

***Figure 64* • XOR8**

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

Truth Table

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:

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

UJTAG

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.

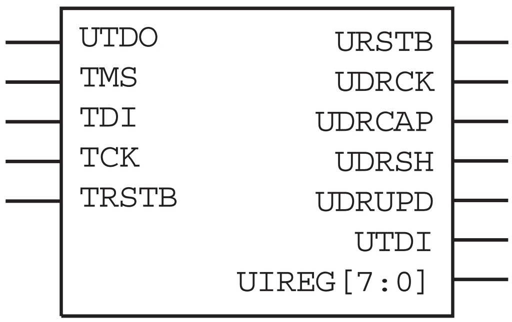

Figure 65 • **UJTAG**

Table 12: 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 address is in user range.
UDRSH	Output	High	Active High signal enabled in the Shift_DR TAP state.
UDRCAP	Output	High	Active High signal enabled in the Capture_DR_TAP state.
UDRCK	Output	—	This port is directly connected to the TAP's TCK signal.
UDRUPD	Output	High	Active High signal enabled in the Update_DR_TAP state.

Table 12: Ports and Descriptions (Continued)

Port	Direction	Polarity	Description
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 +3.3 V 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 IEEE1532 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).

BIBUF

Bidirectional Buffer

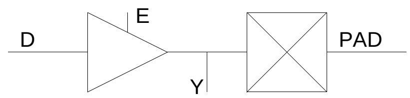

Figure 66 **• BIBUF**

Input	Output
D, E, PAD	PAD, Y

Truth Table

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

BIBUF_DIFF

Bidirectional Buffer, Differential I/O

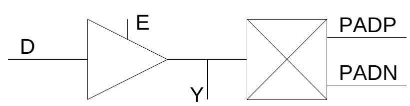

Figure 67 **• BIBUF_DIFF**

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

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
INPUT	0	X	1	1	X
INPUT	0	X	0	1	0
INPUT	0	X	1	0	1

CLKBIBUF

Bidirectional Buffer with Input to global network

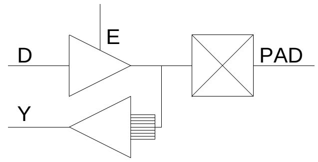

Figure 68 • **CLKBIBUF**

Input	Output
D, E, PAD	PAD, Y

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

CLKBUF

Input Buffer to global network

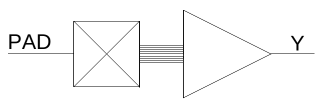

Figure 69 • **CLKBUF**

Input	Output
PAD	Y

Truth Table

PAD	Y
0	0
1	1

CLKBUF_DIFF

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

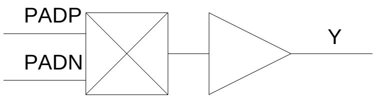

Figure 70 **• INBUF_DIFF**

Input	Output
PADP, PADN	Y

Truth Table

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

INBUF

Input Buffer

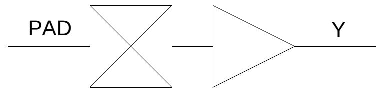

Figure 71 • **INBUF**

Input	Output
PAD	Y

Truth Table

PAD	Y
Z	X
0	0
1	1

INBUF_DIFF

Input Buffer, Differential I/O

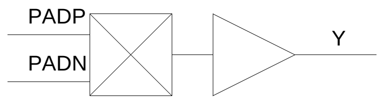

Figure 72 • **INBUF_DIFF**

Input	Output
PADP, PADN	Y

Truth Table

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

OUTBUF

Output buffer

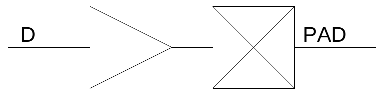

***Figure 73* • OUTBUF**

Input	Output
D	PAD

Truth Table

D	PAD
0	0
1	1

OUTBUF_DIFF

Output buffer, Differential I/O

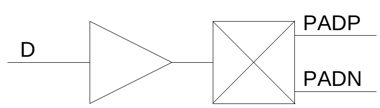

***Figure 74* • OUTBUF_DIFF**

Input	Output
D	PADP, PADN

Truth Table

D	PADP	PADN
0	0	1
1	1	0

TRIBUFF

Tristate output buffer

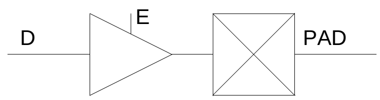

Figure 75 **• TRIBUFF**

Input	Output
D, E	PAD

Truth Table

D	E	PAD
X	0	Z
D	1	D

TRIBUFF_DIFF

Tristate output buffer, Differential I/O

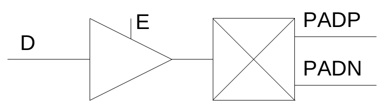

Figure 76 **• TRIBUFF_DIFF**

Input	Output
D, E	PADP, PADN

Truth Table

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

DDR_IN

Input DDR Register; input D must be connected to an I/O.

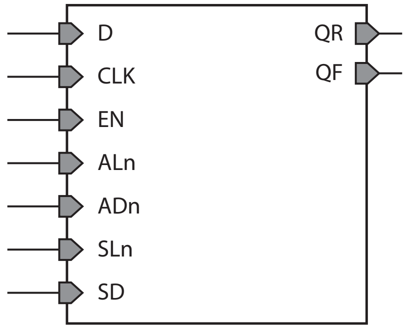

***Figure 77* • DDR_IN**

Input	Output
D, CLK, EN, ALn, ADn, SLn, SD	QR, QF

Input		Output	
Name	**Function**	**Name**	**Function**
D	Data	QR	Q (Rising Edge)
CLK	Clock	QF	Q (Falling Edge)
EN	Enable		
ALn	Asynchronous Load (Active Low)		
ADn*	Asynchronous Data (Active Low)		
SLn	Synchronous Load (Active Low)		
SD*	Synchronous Data		

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

Truth Table

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

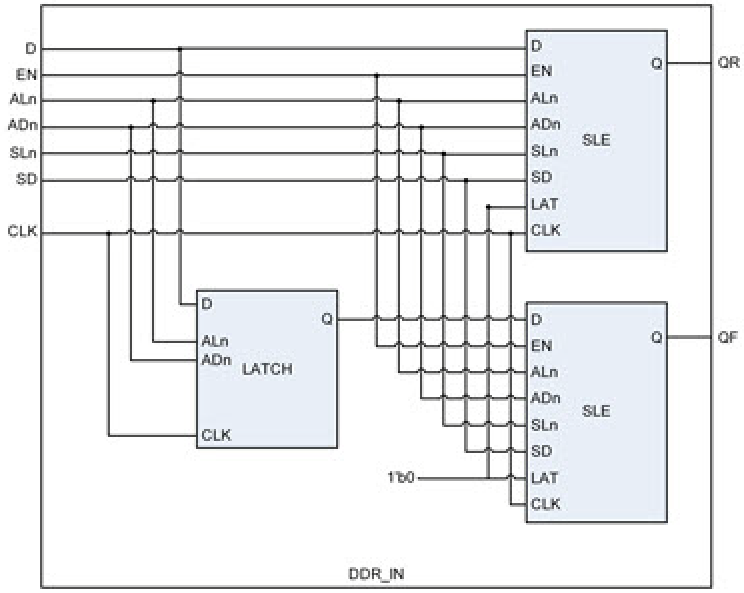

Figure 78 • **DDR_IN**

DDR_OUT

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.

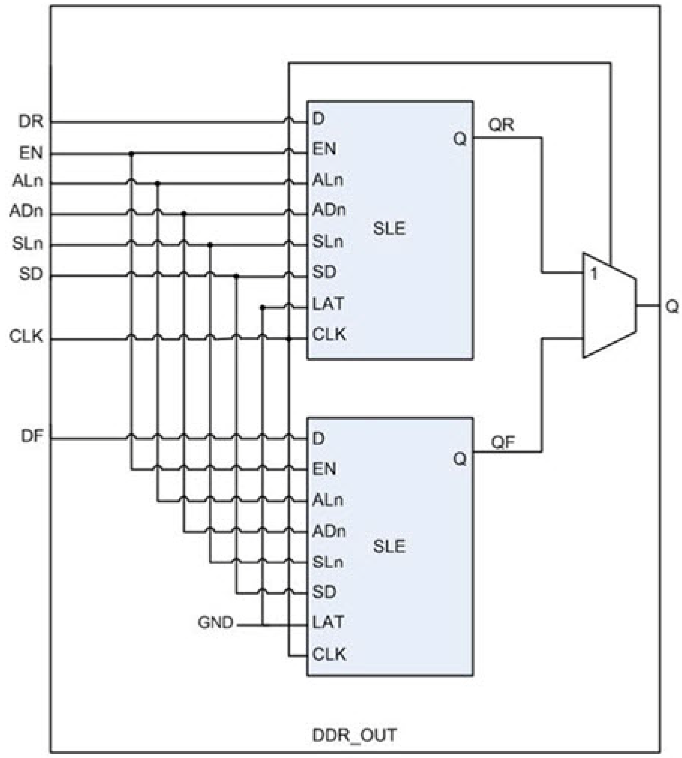

***Figure 79* • DDR_OUT**

Input		Output
Name	**Function**	
DR	Data (Rising Edge)	Q
DF	Data (Falling Edge)	
CLK	Clock	
EN	Enable	
AL_n	Asynchronous Load (Active Low)	
AD_n*	Asynchronous Data (Active Low)	
SL_n	Synchronous Load (Active Low)	
SD*	Synchronous Data	

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

Truth Table

ALn	CLK	EN	SLn	Q_{n+1}
1	not rising	X	X	Q_n
1	↑	0	X	Q_n
1	↑	1	1	DR_n

ALn	CLK	EN	SLn	Q_{n+1}
1	↓	1	1	DF_n
1	↑	1	0	SD
0	X	X	X	$!AD_n$

DDR_OE_UNIT

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 (“DDR_OE_UNIT” on page 63). SDR controls whether the cell operates in DDR (SDR = 0) or SDR (SDR = 1) modes.

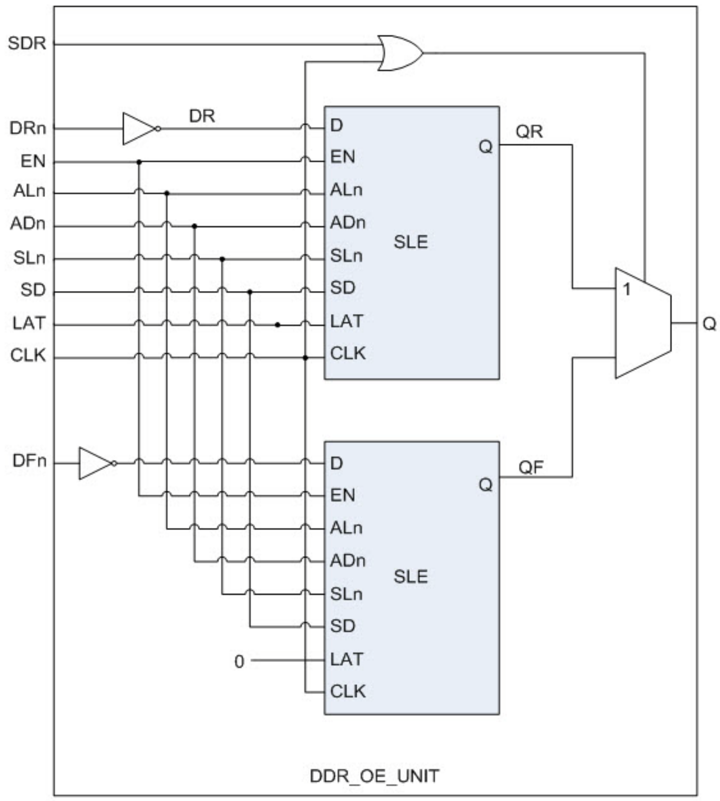

Figure 80 **• DDR_OE_UNIT**

IOIN_IB

Buffer macro available in post-layout netlist only.

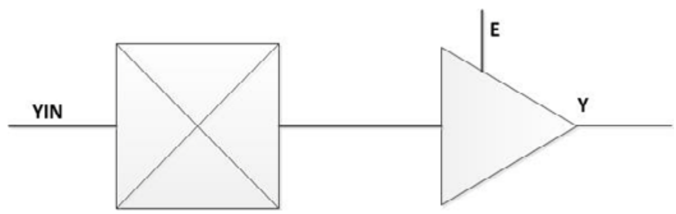

Figure 81 • **IOIN_IB**

Input	Output
YIN, E	Y

Note: E input is not used.

Truth Table

YIN	Y
Z	X
0	0
1	1

IOPAD_IN

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

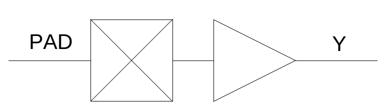

Figure 82 • **IOPAD_IN**

Input	Output
PAD	Y

Truth Table

PAD	Y
Z	X
0	0
1	1

IOPAD_TRI

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

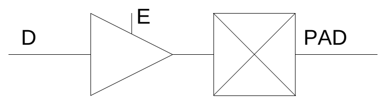

Figure 83 **• IOPAD_TRI**

Input	Output
D, E	PAD

Truth Table

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

IOINFF

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

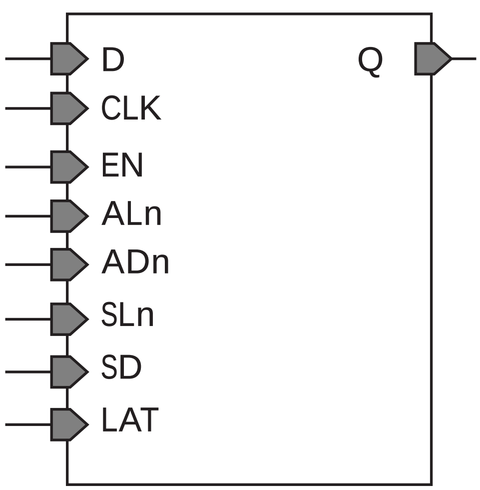

Figure 84 **• IOINFF**

Input		Output
Name	**Function**	Q
D	Data	
CLK	Clock	
EN	Enable	
ALn	Asynchronous Load (Active Low)	
ADn*	Asynchronous Data (Active Low)	
SLn	Synchronous Load (Active Low)	
SD*	Synchronous Data	
LAT*	Latch Enable	

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

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

IOOEFF

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

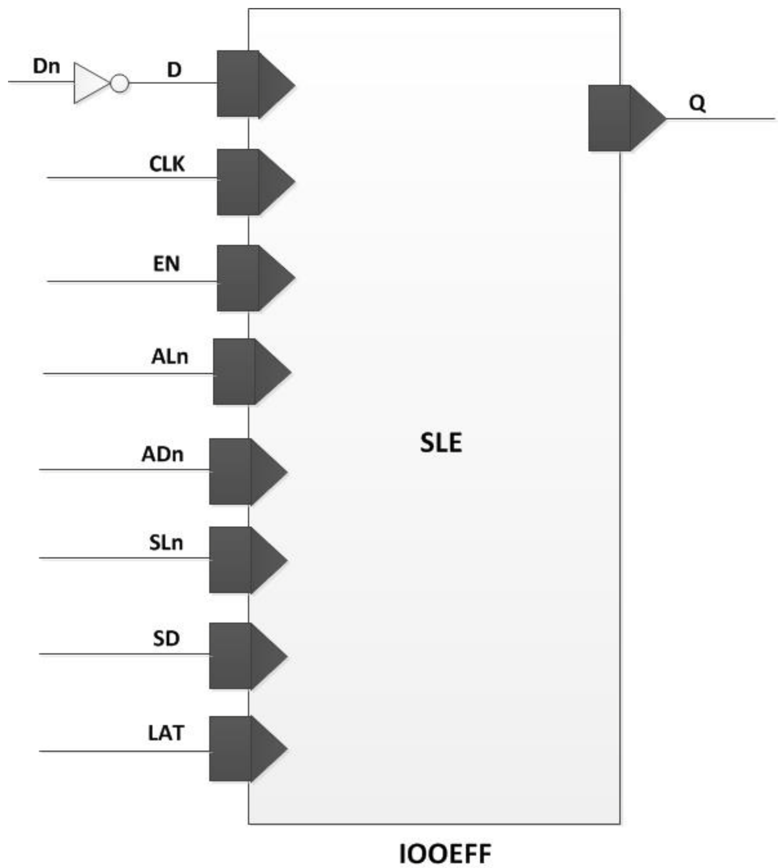

Figure 85 **• IOOEFF**

Input		Output
Name	**Function**	
D	Data	Q
CLK	Clock	
EN	Enable	
ALn	Asynchronous Load (Active Low)	
ADn*	Asynchronous Data (Active Low)	
SLn	Synchronous Load (Active Low)	
SD*	Synchronous Data	
LAT*	Latch Enable	

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

Truth Table

ALn	LAT	CLK	EN	SLn	Q
1	0	not rising	X	X	Q
1	0	rising	0	X	Q

ALn	LAT	CLK	EN	SLn	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

RAM1K18

The RAM1K18 block contains 18,432 memory bits and is a true dual-port memory. The RAM1K18 memory 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 shown below.

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:

- A RAM1K18 block has 18,432 bits.
- A RAM1K18 block provides two independent data ports A and B.
- RAM1K18 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 will become 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 will be 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 is expected to be written into the memory.

Figure 86 shows a simplified block diagram of the RAM1K18 memory block and Table 13 gives the port descriptions.

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

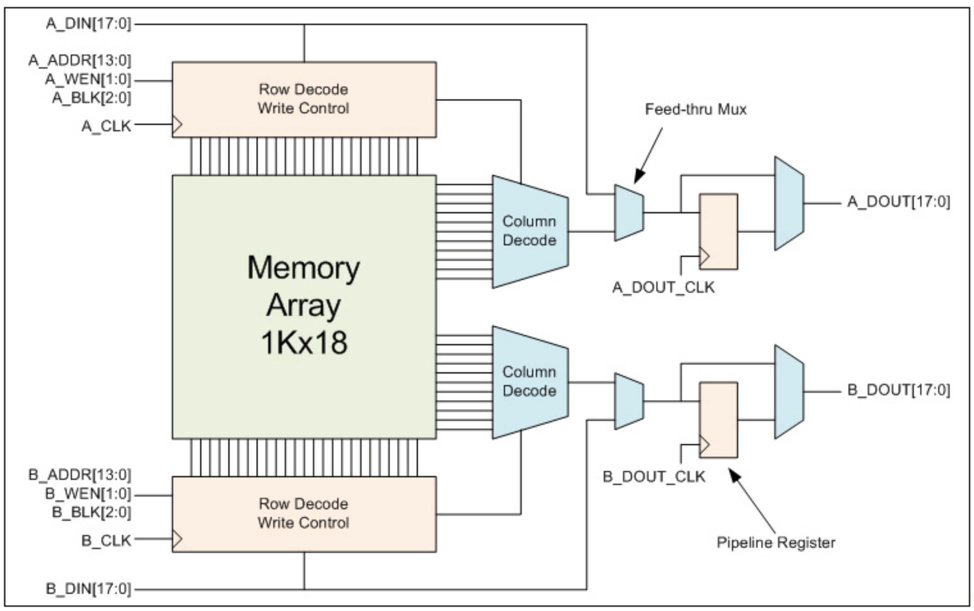

***Figure 86* • Simplified Block Diagram of RAM1K18**

***Table 13* • 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	

Table 13 • **Port RAM List for RAM1K18**

Pin Name	Pin Direction	Type	Description	Polarity
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

Note: 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

Table 14 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 14* • 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

Table 15 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 15* • 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]

Table 15 **• Write/Read Operation Select**

Depth x Width	A_WEN/B_WEN	Result
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

Table 16 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 16 **• 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]
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

Table 17 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 17 **• 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

Table 18 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 18 **• 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 18* • Data Output Buses Used and Unused Bits**

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

Table 19 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 19* • 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, i.e. 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 set up 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 should be tied to 1, which makes them transparent.

Table 20 describes the functionality of the control signals on the A_DOUT and B_DOUT pipeline registers.

***Table 20* • 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, 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 shown below.

- 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 will be triggered at the rising edge of the clock.
- Read from both ports A and B at the same location is allowed.
- Read and write on the same location at the same time results in unknown data to be read. **There is no collision prevention or detection**. However, correct data is expected to be written into the memory.

Figure 87 shows a simplified block diagram of the RAM64x18 memory block and Table 21 gives the port descriptions.

The simplified block diagram illustrates the three independent read/write ports and the pipeline registers on the read port.

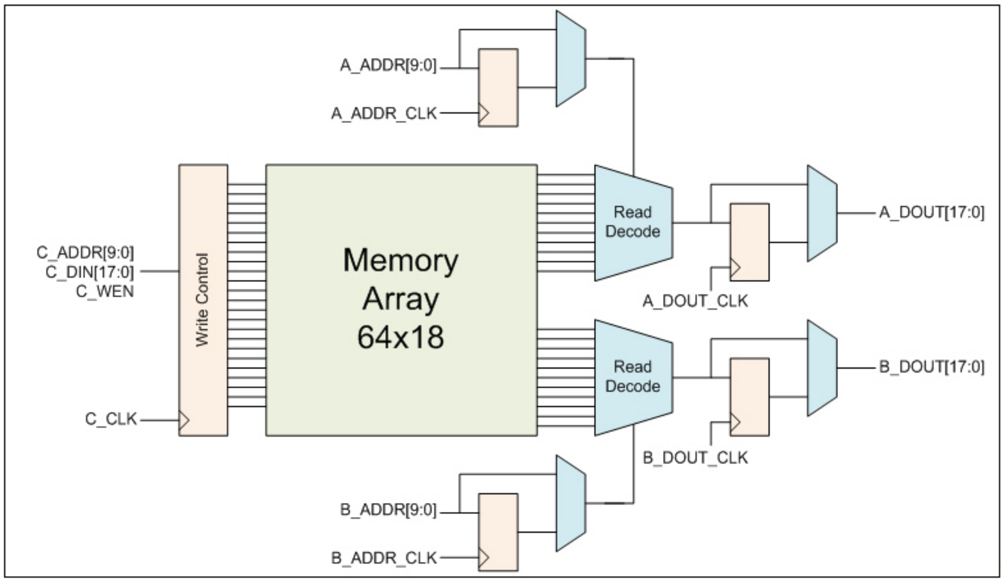

Figure 87 **• Simplified Block Diagram of RAM64x18**

Table 21 **• 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
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

***Table 21* • Port List for RAM64x18**

Pin Name	Pin Direction	Type	Description	Polarity
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

Note: 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

Table 22 lists the width/depth mode selections for each port.

***Table 22* • 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

Table 23 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 23 **• 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

Table 24 lists the write data input for port C. The required bits are LSB justified and unused MSB bits must be tied to 0.

Table 24 **• 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

Table 25 lists the read data output buses for ports A and B. The required bits are LSB justified.

***Table 25* • 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

Table 26 lists the block port select control signals for the ports.

***Table 26* • Block Port Select**

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

C_CLK

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

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 should be tied to 1, which makes them transparent.

Table 27 describes the functionality of the control signals on the A_ADDR, B_ADDR, A_DOUT and B_DOUT registers.

Table 27 **• 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.

MACC

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.

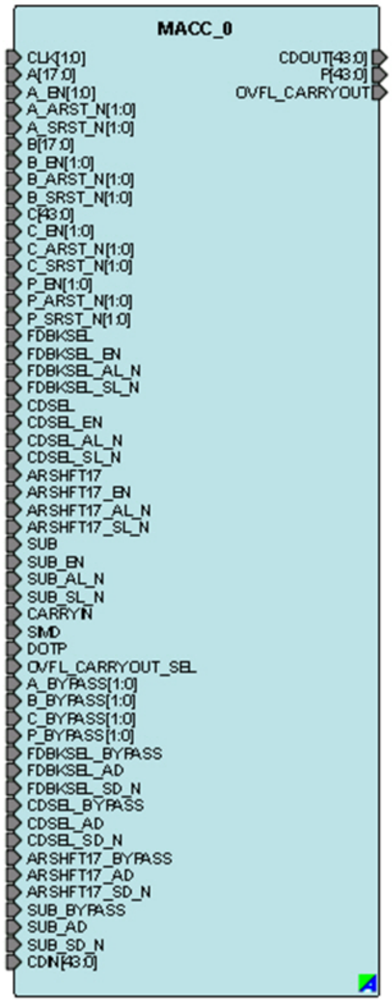

Figure 88 **• MACC Ports**

Table 28 • **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].

Table 28 • **Ports**

Port Name	Direction	Type	Polarity	Description
A_EN[1:0]	Input	Dynamic	High	Enable for data A registers. • A_EN[1] is for A[17:9]. Connect to 1, if not registered. • A_EN[0] is for A[8:0]. Connect to 1, if not registered. In normal mode, ensure A_EN[1] = A_EN[0].
B[17:0]	Input	Dynamic	High	Input data B.
B_BYPASS[1:0]	Input	Static	High	Bypass data B registers. • B_BYPASS[1] is for B[17:9]. Connect to 1, if not registered. • B_BYPASS[0] is for B[8:0]. Connect to 1, if not registered. In normal mode, ensure B_BYPASS[0] = B_BYPASS[1].
B_ARST_N[1:0]	Input	Dynamic	Low	Asynchronous reset for data B registers. • 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].

Table 28 **• Ports**

Port Name	Direction	Type	Polarity	Description
P[43:0]	Output		High	Result data. Normal mode • P = D + (CARRYIN + C) + (A * B), when SUB = 0 • P = D + (CARRYIN + C) - (A * B), when SUB = 1 Dot-product mode • P = D + (CARRYIN + C) + 512 * ((A_L * B_H) + (A_H * B_L)), when SUB = 0 • P = D + (CARRYIN + C) - 512 * ((A_L * B_H) + (A_H * B_L)), when SUB = 1 Notation: • A_L = A[8:0], A_H = A[17:9] • B_L = B[8:0], B_H = B[17:9] Refer to Table 31 on page 92 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].
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].

Table 28 • Ports

Port Name	Direction	Type	Polarity	Description
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].

***Table 28* • Ports**

Port Name	Direction	Type	Polarity	Description
C_SRST_N[1:0]	Input	Dynamic	Low	Synchronous reset for data C registers. • C_SRST_N[1] is for C[43:18]. Connect to 1, if not registered. • C_SRST_N[0] is for C[17:0] and CARRYIN. Connect to 1, if not registered. In normal mode, ensure C_SRST_N[1] = C_SRST_N[0].
C_EN[1:0]	Input	Dynamic	High	Enable for data C registers. • C_EN[1] is for C[43:18]. Connect to 1, if not registered. • C_EN[0] is for C[17:0] and CARRYIN. Connect to 1, if not registered. In normal mode, ensure C_EN[1] = C_EN[0].
CDIN[43:0]	Input	Cascade	High	Cascaded input for operand D. The entire bus must be driven by an entire CDOUT of another MACC block. In Dot-product mode the CDOUT must also be generated by a MACC block in Dot-product mode. Refer to Table 31 on page 92 to see how CDIN is propagated to operand D.
ARSHFT17	Input	Dynamic	High	Arithmetic right-shift for operand D. When asserted, a 17-bit arithmetic right-shift is performed on operand D going into the accumulator. Refer to Table 31 on page 92 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 29 on page 91.
ARSHFT17_SD_N	Input	Static	Low	Synchronous load data for ARSHFT17 register. See Table 29 on page 91.
ARSHFT17_EN	Input	Dynamic	High	Enable for ARSHFT17 register. Connect to 1, if not registered. See Table 29 on page 91.

Table 28 • **Ports**

Port Name	Direction	Type	Polarity	Description
CDSEL	Input	Dynamic	High	Select CDIN for operand D. When CDSEL = 1, propagate CDIN. When CDSEL = 0, propagate 0 or P depending on FDBKSEL. Refer to Table 29 on page 91 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 29 on page 91.
CDSEL_SD_N	Input	Static	Low	Synchronous load data for CDSEL register. See Table 29 on page 91.
CDSEL_EN	Input	Dynamic	High	Enable for CDSEL register. Connect to 1, if not registered. See Table 29 on page 91.
FDBKSEL	Input	Dynamic	High	Select the feedback from P for operand D. When FDBKSEL = 1, propagate the current value of result P register. Ensure P_BYPASS[1] = 0 and CDSEL = 0. When FDBKSEL = 0, propagate 0. Ensure CDSEL = 0. Refer to Table 31 on page 92 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 29 on page 91.
FDBKSEL_SD_N	Input	Static	Low	Synchronous load data for FDBKSEL register. See Table 29 on page 91.
FDBKSEL_EN	Input	Dynamic	High	Enable for FDBKSEL register. Connect to 1, if not registered. See Table 29 on page 91.

Table 28 **• Ports**

Port Name	Direction	Type	Polarity	Description
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 29.
SUB_SD_N	Input	Static	Low	Synchronous load data for SUB register. See Table 29.
SUB_EN	Input	Dynamic	High	Enable for SUB register. Connect to 1, if not registered. See Table 29.

Table 29 **• 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 30 **• 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
1	1	X	1	0	X	0
1	1	X	1	1	D	D

Table 31 **• 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]}

A – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

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**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office.

Visit About Us for sales office listings and corporate contacts.

Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; Enterprise Storage and Communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

©2016 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

5-02-00395-8/10.16