A Guide to ACTgen Macros

Actel

Actel Corporation, Sunnyvale, CA 94086

Part Number: 5029108-1

Release: May 1999

Trademarks

Table of Contents

Introduction

This Guide provides descriptions of macros that can be generated using the Actel ACTgen Macro Builder Software. For further information about instantiating macros, refer to *Designing with Actel* or the *Actel HDL Coding Style Guide*.

The Actel ACTgen Macro Builder generates a large variety of commonly used functions in seconds. Structural netlists can be generated in EDIF, VHDL, Verilog, Viewlogic, Mentor Graphics, and Cadence. Furthermore, VHDL and Verilog behavioral models can be generated for most parameterized functions (the behavioral models can be used in a simulation environment).

Actel's parameterized macros:

- reduce the development time of complex functions.
- offer a large set of implementations for each type of function.
- offer a wide range of bit widths that provides a quick change of design definitions.

Additional information is available in Actel's *Designing with Actel*, and *HDL Coding Style Guide*.

Document Conventions

The following table describes the conventions that are used throughout this manual.

Table 1. Functional Description of Table Nomenclature

Symbol	Definition
X	Don't care
1	Logical 1 or high
0	Logical 0 or low
↑	Rising edge
↓	Falling edge

Table 1. Functional Description of Table Nomenclature

Symbol	**Definition**
Q_n	Value of the signal Q before the active edge of the clock
Q_{n+1}	Value of the signal Q after the active edge of the clock
m, n	Binary pattern with width of function

Note: Default values in tables are displayed in **bold** type.

Symbols

Each macro symbol shows the input and output ports. Busses are highlighted with a bold line; scalar signals with a thin line. The actual symbols generated by ACTgen could look slightly different determined by the particular CAE tool used. Some ports shown could be optional, as described in the port description tables. Default polarities are shown on the symbols.

Actel Manuals

The Designer Series software includes printed and on-line manuals. The on-line manuals are in PDF format on the CD-ROM in the "/manuals" directory. These manuals are also installed onto your system when you install the Designer software. To view the on-line manuals, you must install Adobe® Acrobat Reader® from the CD-ROM.

The Designer Series includes the following manuals, which provide additional information on designing Actel FPGAs:

Designing with Actel. This manual describes the design flow and user interface for the Actel Designer Series software, including information about using the ACTgen Macro Builder and ACTmap VHDL Synthesis software.

Actel HDL Coding Style Guide. This guide provides preferred coding styles for the Actel architecture and information about optimizing your HDL code for Actel devices.

ACTmap VHDL Synthesis Methodology Guide. This guide contains information, optimization techniques, and procedures to assist designers in the design of Actel devices using ACTmap VHDL.

Silicon Expert User's Guide. This guide contains information and procedures to assist designers in the use of Actel's Silicon Expert tool.

DeskTOP Interface Guide. This guide contains information about using the integrated VeriBest® and Synplicity® CAE software tools with the Actel Designer Series FPGA development tools to create designs for Actel Devices.

Cadence® Interface Guide. This guide contains information and procedures to assist designers in the design of Actel devices using Cadence CAE software and the Designer Series software.

Mentor Graphics® Interface Guide. This guide contains information and procedures to assist designers in the design of Actel devices using Mentor Graphics CAE software and the Designer Series software.

MOTIVE™ Static Timing Analysis Interface Guide. This guide contains information and procedures to assist designers in the use of the MOTIVE software to perform static timing analysis on Actel designs.

Synopsys® Synthesis Methodology Guide. This guide contains preferred HDL coding styles and information and procedures to assist designers in the design of Actel devices using Synopsys CAE software and the Designer Series software.

Viewlogic Powerview® Interface Guide. This guide contains information and procedures to assist designers in the design of Actel devices using Powerview CAE software and the Designer Series software.

Viewlogic Workview Office Interface Guide. This guide contains information and procedures to assist designers in the design of Actel devices using Workview Office CAE software and the Designer Series software.

VHDL Vital Simulation Guide. This guide contains information and procedures to assist designers in simulating Actel designs using a Vital compliant VHDL simulator.

Verilog Simulation Guide. This guide contains information and procedures to assist designers in simulating Actel designs using a Verilog simulator.

Activator and APS Programming System Installation and User's Guide. This guide contains information about how to program and debug Actel devices, including information about using the Silicon Explorer diagnostic tool for system verification.

Silicon Sculptor User's Guide. This guide contains information about how to program Actel devices using the Silicon Sculptor software and device programmer.

Silicon Explorer Quick Start. This guide contains information about connecting the Silicon Explorer diagnostic tool and using it to perform system verification.

Designer Series Development System Conversion Guide UNIX® Environments. This guide describes how to convert designs created in Designer Series versions 3.0 and 3.1 for UNIX to be compatible with later versions of Designer Series.

Designer Series Development System Conversion Guide Windows Environments. This guide describes how to convert designs created in Designer Series versions 3.0 and 3.1 for Windows to be compatible with later versions of Designer Series.

Actel FPGA Data Book. This guide contains detailed specifications on Actel device families. Information such as propagation delays, device package pinout, derating factors, and power calculations are found in this guide.

Macro Library Guide. This guide provides descriptions of Actel library elements for Actel device families. Symbols, truth tables, and module count are included for all macros.

A Guide to ACTgen Macros. This Guide provides descriptions of macros that can be generated using the Actel ACTgen Macro Builder software.

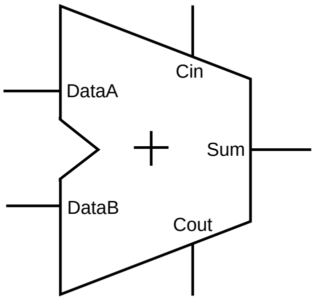
On-Line Help

The Designer Series software comes with on-line help. On-line help specific to each software tool is available in Designer, ACTgen, ACTmap, Silicon Expert, Silicon Explorer, Silicon Sculptor, and APSW.

Adder

Features

- Parameterized word length
- Optional Carry-in and Carry-out signals
- Three gate level implementations (speed/area tradeoffs)
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
DataB	WIDTH	Input	Req.	Input Data
Cin	1	Input	Opt.	Carry-in
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA, DataB and Sum
CI_POLARITY	0 1 **2**	Carry-in polarity (active high, active low and not used)
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low and not used)

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Adder category
LPM_HINT	FADD	Very fast carry select model
	MFADD	Fast carry select model
	RIPADD	Ripple carry model

Functional Description

DataA	DataB	Sum	Cout [a]
m	n	$(m + n + Cin^{b}) \bmod 2^{width}$	$(m + n + Cin) = 2^{width}$

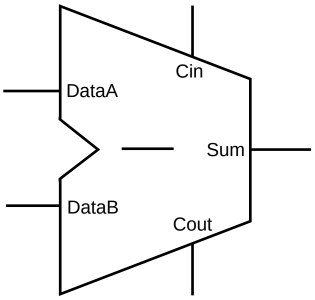
a. Cout is active high

b. Cin is active high

Subtracter

Features

- Parameterized word length
- Optional Carry-in and Carry-out signals
- Three gate level implementations (speed/area tradeoffs)
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
DataB	WIDTH	Input	Req.	Input Data
Cin	1	Input	Opt.	Carry-in
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA, DataB and Sum
CI_POLARITY	0 1 **2**	Carry-in polarity (active high, active low and not used)
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low and not used)

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Subtracter category
LPM_HINT	FSUB	Very fast carry select model
	MFSUB	Fast carry select model
	RIPSUB	Ripple carry model

Functional Description

DataA	DataB	Sum	Cout[a]
m	n	(m - n - Cin[b]) mod 2^{width}	(m - n - Cin) < 2^{width}

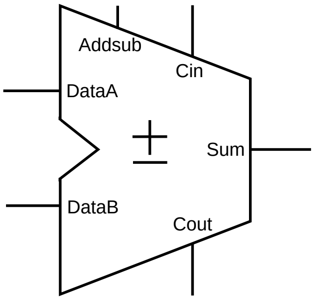
a. Cout is active high

b. Cin is active high

Adder/Subtracter

Features

- Parameterized word length
- Optional Carry-in and Carry-out signals
- Three gate level implementations (speed/area tradeoffs)
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
DataB	WIDTH	Input	Req.	Input Data
Cin	1	Input	Opt.	Carry-in
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out
Addsub	1	Input	Req.	Addition (AddSub = 1) or subtraction (Addsub = 0)

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA, DataB and Sum
CI_POLARITY	0 1 **2**	Carry-in polarity (active high, active low and not used)

Parameter Description

Parameter	Value	Function
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low and not used)

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Adder/Subtracter category
LPM_HINT	FADDSUB	Very fast carry select model
	MFADDSUB	Fast carry select model
	RIPADDSUB	Ripple carry model

Functional Description

DataA	DataB	Addsub	Sum	Cout[a]
m	n	1	(m + n + Cin[b]) mod $2^{\mathbf{width}}$	(m + n + Cin) ≥ $2^{\mathbf{width}}$
m	n	0	(m - n - Cin) mod $2^{\mathbf{width}}$	(m - n - Cin) < $2^{\mathbf{width}}$

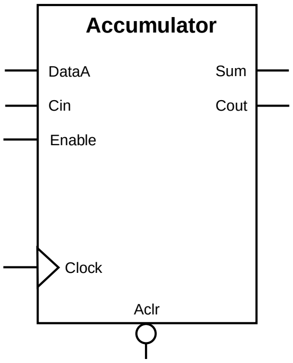
a. Cout is active high

b. Cin is active high

Accumulator

Features

- Parameterized word length
- Optional Carry-in and Carry-out signals
- Asynchronous reset
- Accumulator Enable
- Three gate level implementations (speed/area tradeoffs)
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
Cin	1	Input	Opt.	Carry-in
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA and Sum
CI_POLARITY	0 1 **2**	Carry-in polarity (active high, active low and not used)
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low and not used)

Parameter Description (Continued)

Parameter	Value	Function
CLR_POLARITY	**0** 1 2	Asynchronous reset (active high, active low and not used)
EN_POLARITY	0 **1** 2	Accumulator enable (active high, active low and not used)
CLK_EDGE	**RISE** FALL	

Fanin Control Parameters

Parameter	Value
CLR_FANIN	**AUTO** MANUAL
CLR_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]
EN_FANIN	**AUTO** MANUAL
EN_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
CLK_FANIN	AUTO **MANUAL**
CLK_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Accumulator category
LPM_HINT	FACC	Very fast carry select model
	MFACC	Fast carry select model
	RIPACC	Ripple carry model

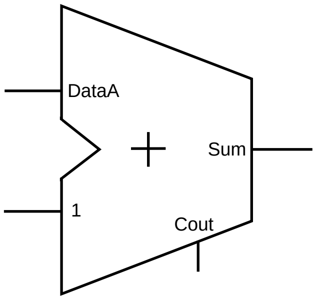
Functional Description

DataA	**Sum^{n+1}**	**Cout**[a]
m	$(m + Sum_n + Cin^{b})$ mod $2^{\mathbf{width}}$	$(m + Sum_n + Cin) \geq 2^{\mathbf{width}}$

a. Cout is actuve high

b. Cin is active high

Incrementer



Features

- Parameterized word length
- Optional Carry-out signals
- One very fast gate level implementation
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 2/1200XL, ACT 3, 3200DX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA and Sum
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low and not used)

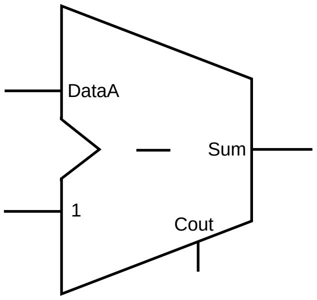
Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Incrementer category
LPM_HINT	FINC	Very fast carry look ahead

Functional Description

DataA	Sum	Cout
m	m + 1	$(m + 1) \geq 2^{\mathbf{width}}$

Decrementer



Features

- Parameterized word length
- Optional Carry-out signals
- One very fast gate level implementation
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 2/1200XL, ACT 3, 3200DX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA and Sum
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low, and not used)

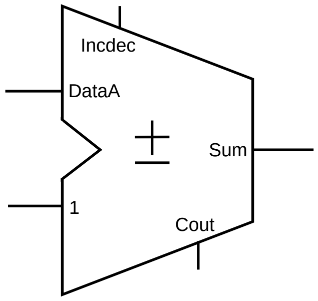
Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Decrementer category
LPM_HINT	FDEC	Very fast carry look ahead

Functional Description

DataA	DataB	Sum	Cout
m	n	m - 1	(m-1) <0

Incrementer/Decrementer



Features

- Parameterized word length
- Optional Carry-out signals
- One very fast gate level implementation
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 2/1200XL, ACT 3, 3200DX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
Sum	WIDTH	Output	Req.	Sum
Cout	1	Output	Opt.	Carry-out
Incdec	1	Input	Req.	Increment (Incdec = 1) or decrement (Incdec = 0)

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA and Sum
CO_POLARITY	0 1 **2**	Carry-out polarity (active high, active low and not used)

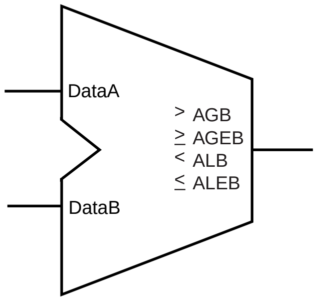
Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_ADD_SUB	Incrementer/Decrementer category
LPM_HINT	FINCDEC	Very fast carry look ahead

Functional Description

DataA	Incdec	Sum	Cout
m	1	m + 1	$(m + 1) \geq 2^{\mathbf{width}}$
m	0	m - 1	$(m - 1) < 0$

Comparator



Features

- Parameterized word length
- Unsigned and signed (two's complement) data comparison
- One very fast gate level implementation
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTH	Input	Req.	Input Data
DataB	WIDTH	Input	Req.	Input Data
AGB	1	Output	Opt.	Output comparison
AGEB	1	Output	Opt.	Output comparison
ALB	1	Output	Opt.	Output comparison
ALEB	1	Output	Opt.	Output comparison
AEB	1	Output	Opt.	Output comparison
ANEB	1	Output	Opt.	Output comparison

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of DataA and DataB

Parameter Description

Parameter	Value	Function
REPRESENTATION	**UNSIGNED** SIGNED	
AGB_POLARITY	0 1 **2**	AGB polarity (active high, active low and not used)
AGEB_POLARITY	0 1 **2**	AGEB polarity (active high, active low and not used)
ALB_POLARITY	0 1 **2**	ALB polarity (active high, active low and not used)
ALEB_POLARITY	0 1 **2**	ALEB polarity (active high, active low and not used)
AEB_POLARITY	0 1 **2**	AEB polarity (active high, active low and not used)
ANEB_POLARITY	0 1 **2**	ANEB polarity (active high, active low and not used)

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_COMPARE	Comparator category
LPM_HINT	COMPARE	Very fast carry select

Parameter Rules

Parameter Rules
At lease one of the comparisons (AGB, AGEB, ALB, ALEB, AEB or ANEB) must be selected
Only one of the magnitude comparisons (AGB, AGEB, ALB or ALEB) can be selected at the same time

Parameter Rules

Parameter Rules
Only one of the equality comparisons (AEB or ANEB) can be selected at the same time

Functional Description

DataA	DataB	AGB	AGEB	ALB	ALEB	AEB	ANEB
m	n	$m > n$	$m \geq n$	$m < n$	$m \leq n$	$m = n$	$m \neq n$

Implementation Parameters

Implementation (LPM_HINT)	Description
COMPARE	Very fast carry select model

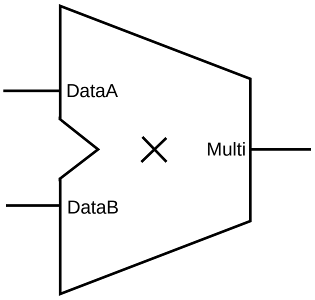
Parameter Rules

Parameter rules
At least one of the comparisons (AGB, AGEB, ALB, ALEB, AEB or ANEB) must be selected
Only one of the magnitude comparisons (AGB, AGEB, ALB or ALEB) can be selected at the same time
Only one of the equality comparisons (AEB or ANEB) can be selected at the same time

Multiplier

Features

- Parameterized word lengths
- Unsigned and signed (two's complement) data representation
- Booth implementation (pipelined or not pipelined)
- Behavioral simulation model (for non-pipelined multiplier only) in VHDL and Verilog



Family Support

ACT 2/1200XL, ACT 3, 3200DX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
DataA	WIDTHA	Input	Req.	Input Data
DataB	WIDTHB	Input	Req.	Input Data
Clock	1	Input	Opt.	Clock
Product	WIDTHA+WIDTHB	Output	Req.	Product DataA*DataB

Parameter Description

Parameter	Value	Function
WIDTHA	2-29	Word length of DataA
WIDTHB	2-29	Word length of DataB
REPRESENTATION	**UNSIGNED** SIGNED	DataA and DataB data representation
CLK_EDGE	**RISE** FALL	Clock (if pipelined)

Functional Description

DataA	DataB	Sum[a]
m	n	m * n

a. If pipelined, the sum is correct (available) after <latency> cycles. Latency is a function of WIDTHA and WIDTHB.

Parameter Rules

Parameter rules
WIDTHA ≥ WIDTHB
WIDTHA + WIDTHB <=32

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_MULT	Multiplier category
LPM_HINT	BOOTHMULT	Booth model
	BOOTHMULTP	Pipelined booth model

Implementations

Actel’s multiplexer-based architecture allows efficient implementation of the multiplier’s high performance. The function of a binary unsigned multiplier, like its decimal counterpart, consists of a multiplicand (X), a multiplier (Y), and a product (P). The result is the product of the multiplier and the multiplicand (P=X*Y).

As in decimal multiplication, the least significant digit of the multiplier combines with each digit of the multiplicand, forming a partial product (Y0X3, Y0X2, Y0X1, Y0X0). For a 4-bit multiplier, three other partial products are similarly formed. To arrive at the final result, all four of the partial products are added.

The conventional approach to implementing a multiplier in digital logic is to AND individual multiplier and multiplicand bits to generate the partial products (PP1, PP2, PP3, PP4). Using this approach, a four-bit multiplier would consist of 16 dual-input AND gates and three adders. This implementation is resource intensive and does not produce optimal performance.

ACTgen uses the Booth algorithm, an alternative technique based on multiplexers that implement multipliers more efficiently fit for the Actel architecture. For a four-bit multiplier, the two least significant bits are handled separately from the two most significant bits. A multiplexer replaces the first stage of partial sum generation. The four possible combinations of the multiplier bits are covered with the multiplexers. Specifically, the four combinations are zero (the trivial case), X (when Y1=0 and Y0=1), X shifted left or 2X (when Y1=1 and Y0=0), and 3X (when Y1=Y0=1). The multiplexers are eight bits deep to accommodate all eight possible inputs for the adders. To obtain the final product, the two partial sums are added with an eight-bit adder. High-speed adders are used in this implementation because the shortest delay from input to output is the primary design constraint.

Figure 2 shows the booth multiplier schematic.

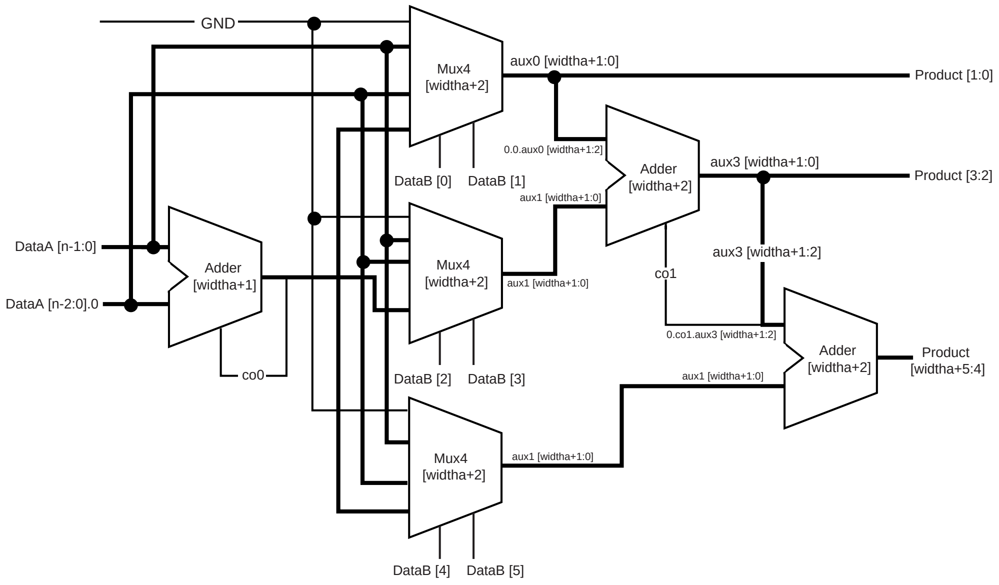


Figure 2. Booth Multiplier Schematic

Booth (pipelined)

The current booth multiplier pipeline architecture does not allow you to select the latency of the pipeline multiplier nor the number of logic levels between the pipeline stages. Registers are automatically inserted between the major components of the architecture, primarily the multiplexer and adder macros, as shown in Figure 3.

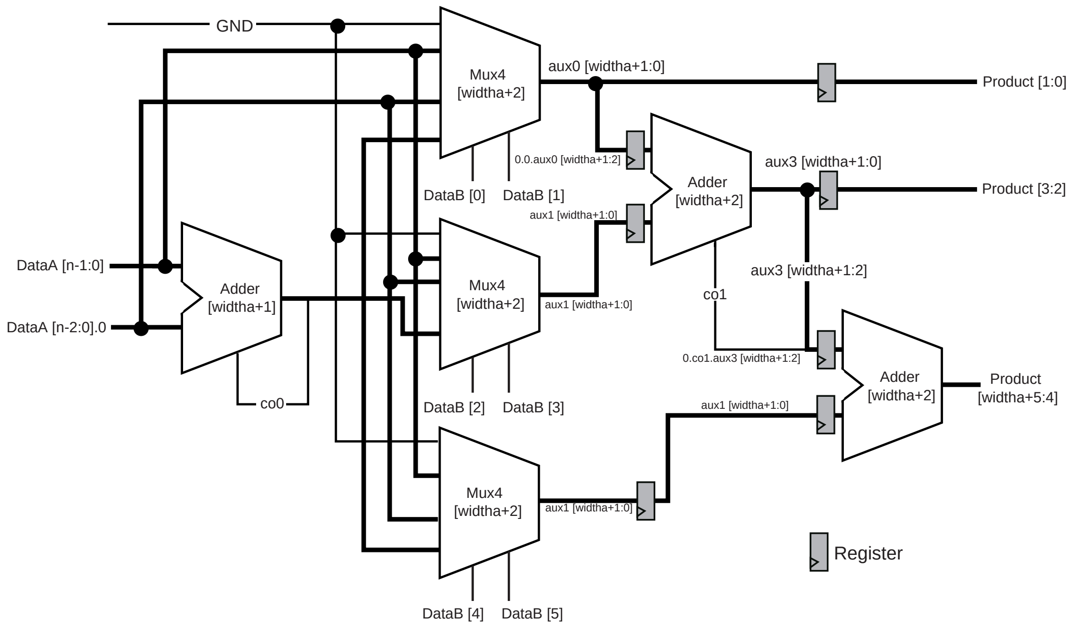


Figure 3. Booth Multiplier Architecture (Pipeline)

The number of pipeline stages is a function of the width of the DataB input. The number of logic levels per pipeline stage is a function of the width of the DataA input. Therefore, the number of logic levels per pipeline stage is equal to the number of logic levels of the first adder (WIDTHA + 1) plus 1 for the 4 to 1 multiplexer, as shown in Figure 3.

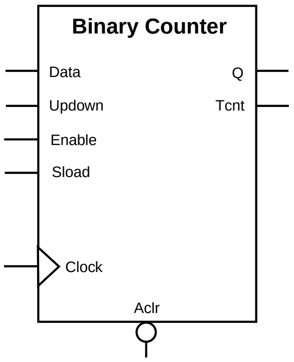
Pipeline Stages/Input Widths

WidthB Range	Pipeline Stages	WidthA Range	Logic Levels
2	0	2-5	3
3-4	1	6-17	4
5-8	2	18-29	5
9-16	3		

Binary Counter

Features

- Parameterized word length
- Up, Down and Up/Down architectures
- Asynchronous clear
- Synchronous counter load
- Synchronous count enable
- Terminal count flag
- Multiple gate level implementations (area/speed tradeoffs)
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 2/1200XL, ACT 3, 3200DX, 42MX, 54SX

Description

The ACTgen binary counters are general purpose UP, DOWN, or UP/ DOWN (direction) counters.

When the count value equals $2^{\mathbf{width}}$-1, the signal *Tcnt* (terminal count), if used, is asserted high. For specific “count to” counters with synchronous clear, see “Modulo Counter” on page 41.

The counters are WIDTH bits wide and have $2^{\mathbf{width}}$ states from “000…0” to “111…1”. The counters are clocked on the rising (RISE) or falling (FALL) edge of the clock signal *Clock* (CLK_EDGE).

The *Clear* signal (CLR_POLARITY), active low or high, provides an asynchronous reset of the counter to “000…0”. You may choose to not implement the reset function.

In the case of an Up/Down counter, the *Updown* signal controls whether the counter counts up (Updown = 1) or down (Updown = 0).

The counter could be loaded with *Data*. The *Sload* signal (LD_POLARITY), active high or low, provides a synchronous load operation with respect to the clock signal *Clock*. You can choose to not implement this function.

The ACTgen counters have a count enable signal *Enable* (EN_POLARITY). *Enable* can be active high or low. When *Enable* is not active, the counter is disabled and the internal state is unchanged.

Port Description

Port Name	Size	Type	Req/ Opt	Function
Data	WIDTH	input	Opt.	Counter load input
Aclr	1	input	Opt.	Asynchronous counter reset
Enable	1	input	Req.	Counter enable
Sload	1	input	Opt.	Synchronous counter load
Clock	1	input	Req.	Clock
Updown	1	input	Opt.	UP (Updown = 1), DOWN (Updown = 0)
Q	WIDTH	out-put	Req.	Counter output bus
Tcnt	1	out-put	Opt.	Terminal count (active high)

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of Data and Q
DIRECTION	**UP** DOWN UPDOWN	Counter direction
CLR_POLARITY	**0** 1 2	Aclr can be active low, active high or not used
EN_POLARITY	0 **1**	Enable can be active low, active high
LD_POLARITY	0 **1** 2	Sload can be active low, active high or not used
CLK_EDGE	**RISE** FALL	

Parameter Description (Continued)

Parameter	Value	Function
TCNT_POLARITY	1 **2**	Tcnt can be active high or not used

Fanin Control Parameters

Parameter	Value
CLR_FANIN	**AUTO** MANUAL
CLR_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]
LD_FANIN	**AUTO** MANUAL
LD_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
CLK_FANIN	AUTO **MANUAL**
CLK_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]

Implementation Parameters

Parameter	Value	Description	Family
LPMTYPE	LPM_COUNTER	Counter category	
LPM_HINT	LLCNT	Prescaled model	All
	TLACNT	Register look ahead model	All
	FBCNT	Fast Balanced model	54SX only
	BCNT	Balanced model	All
	LECNT	Fast Enable Balanced	All
	COMPCNT	Compact model	All
	RIPPLE	Ripple model	All

Functional Description[a]

Data	Aclr	Enable	Sload	Clock	Up down	Qn+1	Tcnt n+1
X	0	X	X	X	X	0's	0
X	1	X	X	-	X	Qn	Qn+1== 2^{width}-1
X	1	0	0	-	X	Qn	Qn+1== 2^{width}-1
m	1	X	1	-	X	m	Qn+1== 2^{width}-1
X	1	1	0	-	1	Qn + 1	Qn+1== 2^{width}-1
X	1	1	0	-	0	Qn - 1	Qn+1== 2^{width}-1

a. Aclr is active low, Enable is active high, Sload is active high, Clock is rising, Tcnt is active high

Implementations

Pre-Scaled Counter

The pre-scaled counter achieves absolute maximum count and count enable performance by sacrificing synchronous load performance. This counter registers the two least significant bits and uses them as an enable for the upper bits. Count performance is limited only by the delay in the lower two bits and the enable path for the upper bits. Because the upper bits are only updated (enabled) every fourth cycle, they can accommodate more delay (up to 4 times the clock frequency).

There are two limitations related to the use of the pre-scaled counter. The first is in analyzing the actual performance of the counter. The second is correctly performing data load functions; these two limitations are related. Two parameters must be measured to overcome these two limitations. The first parameter that must be measured is the worst internal delay inside the counter. The second parameter is the worst delay from Q0/Q1 to any upper bit. The minimum count period is then defined by the greater value of these two parameters.

The load function is a slave of the maximum internal path delay in the pre-scaled counter. The load function must be held for as many clock periods as required to exceed the maximum internal delay; this ensures that all internal nodes are settled and that correct count operation can be performed. This requirement can be waived if you can guarantee that 0’s will always be loaded in Q0 and Q1 (resulting in only a single load cycle).

The count path in pre-scaled counters without Sload or Enable functions only have a single logic level for ACT 2/1200XX, ACT 3, 3200DX, 42MX and 54SX. All other combinations of pre-scaled counters have two logic levels in their count path. In these cases, given the two limitations mentioned previously related to the pre-scaled counter, you should use the Register Look Ahead or Fast Balanced counters.

Register Look Ahead Counter

This counter achieves the absolute maximum performance for the count, count enable, and synchronous load functions. The counter operates by registering intermediate count values providing “look-ahead” carry circuitry. As a result, this counter variation requires more flip-flops (sequential modules) than other counters.

Fast Balanced Counter

This counter is only available for the 54SX family. It takes advantage of the architectural features of the 54SX family, including flip-flops with built-in enable and more powerful combinatorial cells. Using these two features, it is possible to build a very fast and compact binary counter without using “look-ahead” carry circuitry. This counter should be preferred over all the others available for this family.

Balanced Counter

This counter achieves high performance for both the count and enable functions using standard design approaches. Module count performance is sacrificed to maintain high speed. This counter is the result of the performance balance between the count/enable functions and the balance between the performance/cost in building this architecture. This counter should address most counter needs for the ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX and 42MX families.

Fast Enable Counter

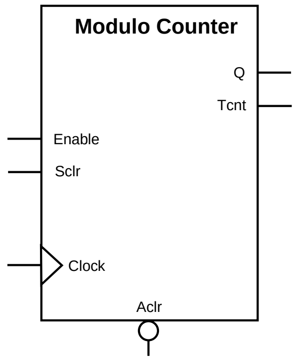
This compact counter is fully synchronous and has higher performance than the ripple counter. However, this counter should only be used in moderate performance applications, especially for large widths.

Ripple Counter

The ripple counter is an asynchronous counter where the Q of each bit feeds the clock of the next bit; performance is sacrificed to build this variation. However, the ripple counter uses the least amount of logic resources. This counter should only be used in very low-performance applications or for very small counters.

Because of the asynchronous nature of the count function, this counter does not have a synchronous load function.

Modulo Counter



Features

- Parameterized modulus value
- Asynchronous clear
- Synchronous clear
- Synchronous count enable
- Terminal count flag

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

The modulo counter counts to the value specified by modulus. When the modulo counter has counted modulus clock cycles from the time it's been reset (asynchronously or synchronously), signal *Tcnt* (terminal count) is asserted high and the counter is synchronously reset to zero. The modulo counter uses an LFSR (linear feedback shift register) implementation. The internal states of the counter do not behave as a binary counter.

The width of the modulo counters is a function of the modulus value and is transparent. The counters are clocked on the rising (RISE) or falling (FALL) edge of the clock *Clock* (CLK_EDGE).

The *Clear* signal (CLR_POLARITY), active low or high, provides an asynchronous reset of the counter to “000…0”. You may choose to not implement this function.

The counter may be synchronously reset to “000…0”. The *Sclr* signal (SCLR_POLARITY), active high or low, provides a synchronous reset operation with respect to the clock signal *Clock*. You may choose to not implement this function.

The ACTgen modulo counters have a count enable signal *Enable* (EN_POLARITY*)* that can be active high or low. When *Enable* is not active, the counter is disabled and the internal state is unchanged. You may choose to not implement this function.

In the current implementations, *Sclr* has priority over *Enable*.

Port Description

Port Name	Size	Type	Req/Opt	Function
Aclr	1	input	Opt.	Asynchronous counter reset
Enable	1	input	Opt.	Counter enable
Sclr	1	input	Opt.	Synchronous counter reset
Clock	1	input	Req.	Clock
Tcnt	1	output	Req.	Terminal count (active high)

Parameter Description

Parameter	Value	Function
MODULUS	5-2000626046	“count to” value
CLR_POLARITY	**0** 1 2	Aclr can be active low, active high or not used
EN_POLARITY	0 **1** 2	Enable can be active low, active high
SCLR_POLARITY	0 **1** 2	Sclr can be active low, active high or not used
CLK_EDGE	**RISE** FALL	Clock
TCNT_POLARITY	1	Tcnt is active high

Fanin Control Parameters

Parameter	Value
CLR_FANIN	**AUTO** MANUAL
CLR_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]
EN_FANIN	**AUTO** MANUAL
EN_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
SCLR_FANIN	**AUTO** MANUAL
SCLR_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
CLK_FANIN	AUTO **MANUAL**
CLK_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]

Implementation Parameters

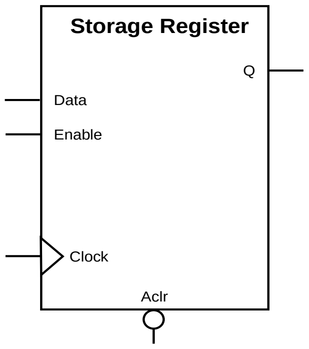
Parameter	Value	Description	Family
LPMTYPE	LPM_COUNTER	Counter category	
LPM_HINT	FASTMOD	Fast LFSR Model	All

Functional Description[a]

Aclr	Enable	Sclr	Clock	Tcnt_{n+1}
0	X	X	X	0
1	X	X	↓	Tcnt_n
1	0	0	↑	Tcnt_n
1	X	1	↑	0
1	1	0	↑	Q_{n+1} == modulus

a. Q represents the symbolic internal state of the modulo counter.

Storage Register



Features

- Parameterized word length
- Asynchronous clear
- Synchronous register parallel load
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Storage registers have a parallel-in/parallel-out (PIPO) architecture. The registers are WIDTH bits. They are clocked on the rising (RISE) or falling (FALL) edge of the clock *Clock* (CLK_EDGE).

The *Clear* signal (CLR_POLARITY), active high or low, provides an asynchronous reset of the registers to “000…0”. You may choose to not implement the reset function.

The *Enable* signal (EN_POLARITY), active high or low, provides a synchronous load enable operation with respect to the *Clock* signal. You can choose to not implement this function. Storage registers are then loaded with a new value every clock cycle.

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	WIDTH	input	Req.	Register load input
Aclr	1	input	Opt.	Asynchronous register reset
Enable	1	input	Opt.	Synchronous Parallel load enable
Clock	1	input	Req.	Clock
Q	WIDTH	output	Req.	Register output bus

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of Data and Q
CLR_POLARITY	**0** 1 2	Aclr can be active low, active high or not used
EN_POLARITY	0 **1** 2	Enable can be active low, active high
CLK_EDGE	**RISE** FALL	Clock can be rising or falling

Fanin Control Parameters

Parameter	Value
CLR_FANIN	**AUTO** MANUAL
CLR_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]
EN_FANIN	**AUTO** MANUAL
EN_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
CLK_FANIN	AUTO **MANUAL**
CLK_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]

Implementation Parameters

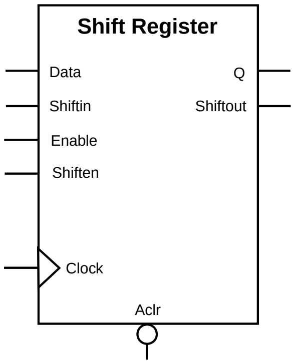
Parameter	Value	Description
LPMTYPE	LPM_DFF	Register category
LPM_HINT	PIPO	Parallel-in/Parallel-out

Functional Description[a]

Data	**Aclr**	**Enable**	**Clock**	$\mathbf{Q_{n+1}}$
X	0	X	X	0’s
X	1	X	$\downarrow$	Q_n
X	1	0	$\uparrow$	Q_n
m	1	1	$\uparrow$	m

a. Aclr is active low, Enable is active high, Clock is rising

Shift Register



Features

- Parameterized word length
- Asynchronous clear
- Synchronous parallel load
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Shift registers have parallel-in/parallel-out (PIPOS), parallel-in/serial-out (PISO), serial-in/parallel-out (SIPO) and serial-in/serial-out (SISO) architecture. The registers are WIDTH bits. They are clocked on the rising (RISE) or falling (FALL) edge of the clock *Clock* signal (CLK_EDGE).

The *Clear* signal (CLR_POLARITY), active high or low, provides an asynchronous reset of the registers to “000…0”. You may choose to not implement the reset function.

Shift registers can be loaded with *Data*. The *Enable* signal (EN_POLARITY), active high or low, provides a synchronous load enable operation with respect to the clock signal *Clock*. You may choose to not implement this function. Shift registers are then implemented in a serial-in mode (SIPO or SISO).

Shift registers have a shift enable signal *Shiften* (SHEN_POLARITY) that can be active high or low. When *Shiften* is active, the register is shifted internally. The LSB is loaded with *Shiftin*.

In the current implementation, *Enable* has priority over *Shiften*.

Port Description

Port Name	**Size**	**Type**	**Req/Opt**	**Function**
Data	WIDTH	input	Opt.	Register load input data
Shiftin	1	Input	Opt.	Shift in signal
Aclr	1	input	Opt.	Asynchronous register reset

Port Description

Port Name	Size	Type	Req/Opt	Function
Enable	1	input	Opt.	Synchronous Parallel load enable
Shiften	1	input	Req.	Synchronous register shift enable
Clock	1	input	Req.	Clock
Q	WIDTH	output	Opt.	Register output bus
Shiftout	1	output	Opt.	Shift out signal

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of Data and Q
CLR_POLARITY	**0** 1 2	Aclr can be active low, active high or not used
EN_POLARITY	0 **1** 2	Enable can be active low, active high
SHEN_POLARITY	0 **1**	Shiften can be active low, active high or not used
CLK_EDGE	**RISE** FALL	Clock can be rising or falling

Fanin Control Parameters

Parameter	Value
CLR_FANIN	**AUTO** MANUAL
CLR_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]
EN_FANIN	**AUTO** MANUAL
EN_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
SHEN_FANIN	**AUTO** MANUAL
SHEN_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
CLK_FANIN	AUTO **MANUAL**

Fanin Control Parameters

Parameter	Value
CLK_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_DFF	Register category
LPM_HINT	PIPOS	Parallel-in/Parallel-out shift register
	PISO	Parallel-in/Serial-out shift register
	SIPO	Serial-in/Parallel-out shift register
	SISO	Serial-in/Serial-out shift register

Functional Description[a]

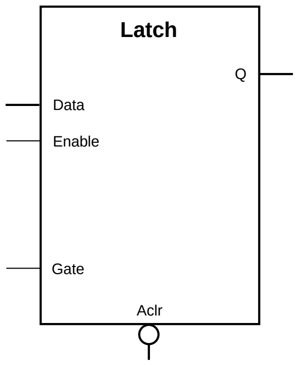
Data	Aclr	Enable	Shiften	Clock	Q_{n+1}[b]	Shiftout[c]
X	0	X	X	X	0's	Q_{n+1} [WIDTH-1]
X	1	X	X	↓	Q_n	Q_{n+1}[WIDTH-1]
X	1	0	0	↑	Q_n	Q_{n+1}[WIDTH-1]
X	1	0	1	↑	Q_n[WIDTH-2:0] && Shiftin	Q_{n+1}[WIDTH-1]
m	1	1	X	↑	m	Q_{n+1}[WIDTH-1]

a. Aclr is active low, Enable is active high, Shiften is active high, Clock is rising.

b. For the PISO and SISO implementations, Q is an internal register.

c. For the PIPO and SIPO implementations, Shiftout is not present.

Latch



Features

- Parameterized word length
- Asynchronous clear
- Synchronous latch enable
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 54SX

Description

Latches have a parallel-in/parallel-out architecture (PIPO). The latches are WIDTH bits. The latches are gated on the active high (HIGH) or low (LOW) state of the gate *Gate* (GATE_POLARITY).

The *Clear* signal (CLR_POLARITY), when active high or low, provides an asynchronous reset of the latch to "000...0". You may choose to not implement this function.

The *Enable* signal (EN_POLARITY), when active high or low, provides a synchronous latch enable operation with respect to the gate *Gate*. You may choose to not implement this function. Latches are then loaded with a new value when both *Enable* and *Gate* are active.

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	WIDTH	input	Req.	Latch load input
Aclr	1	input	Opt.	Asynchronous latch reset
Enable	1	input	Opt.	Synchronous parallel latch enable
Gate	1	input	Req.	Gate
Q	WIDTH	output	Req.	Latchoutput bus

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of Data and Q
CLR_POLARITY	**0** 1 2	Aclr can be active low, active high or not used
EN_POLARITY	0 **1** 2	Enable can be active low, active high
GATE_POLARITY	0 **1**	Gate can be active low or active high

Fanin Control Parameters

Parameter	Value
CLR_FANIN	**AUTO** MANUAL
CLR_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]
EN_FANIN	**AUTO** MANUAL
EN_VAL	<val> [default value for AUTO is 6, 1 for MANUAL]
GATE_FANIN	AUTO **MANUAL**
GATE_VAL	<val> [default value for AUTO is 8, 1 for MANUAL]

Implementation Parameters

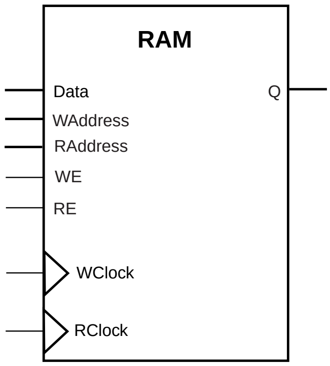
Parameter	Value	Description
LPMTYPE	LPM_LATCH	Latch category
LPM_HINT	N/A	Not needed

Functional Description[a]

Data	Aclr	Enable	Gate	Q_{n+1}
X	0	X	X	0’s
X	1	X	0	Q_n
X	1	0	Ξ	Q_n
m	1	0	Ξ	Q_n

a. Aclr is active low, Enable is active high, Gate is active high

Synchronous/Asynchronous Dual Port RAM



Features

- Parameterized word length and depth
- Dual port synchronous RAM architecture
- Dual port synchronous write, asynchronous read RAM architecture

Family Support

3200DX, 42MX

Description

The RAM macros use 3200DX and 42MX, 32x8 or 64x4, dual port RAM cells.

In the synchronous mode, the read and write operations are totally independent and can be performed simultaneously. The operation of the RAM is fully synchronous with respect to the clock signals, *WClock* and *RClock*. Data of value *Data* are written to the *WAddress* of the RAM memory space on the rising (RISE) or falling (FALL) edge of the clock *WClock* (WCLK_EDGE). Data are read from the RAM memory space at *RAddress* into Q on the rising (RISE) or falling (FALL) edge of the clock signal *RClock* (RCLK_EDGE).

Note: The behavior of the RAM is unknown if you write and read at the same address and signals *WClock* and *RClock* are not the same. The output Q of the RAM depends on the time relationship between the write and the read clock.

In the asynchronous mode, the operation of the RAM is only synchronous with respect to the clock signal *WClock*. Data of value *Data* are written to the *WAddress* of the RAM memory space on the rising (RISE) or falling (FALL) edge of the clock signal *WClock* (WCLK_EDGE). Data are read from the RAM memory space at *RAddress* into Q after some delay when *RAddress* has changed.

Note: The behavior of the RAM is unknown if you write and read at the same address. The output Q depends on the time relationship between the write clock and the read address signal.

The WIDTH (word length) and DEPTH (number of words) have continuous values but the choice of WIDTH limits the choice of DEPTH and vice versa.

The write enable (*WE*) and read enable (*RE*) signals are active high request signals for writing and reading, respectively; you may choose not to use them.

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	WIDTH	input	Req.	Input Data
WE	1	input	Opt.	Write Enable
RE	1	input	Opt.	Read Enable
WClock	1	input	Req.	Write clock
RClock	1	input	Opt.	Read clock
Q	WIDTH	output	Req.	Output Data

Parameter Description

Parameter	Value	Function
WIDTH	width	Word length of Data and Q
Depth	depth	Number of RAM words
WE_POLARITY	**1** 2	WE can be active high or not used
RE_POLARITY	**1** 2	RE can be active high or not used
WCLK_EDGE	**RISE** FALL	WClock can be rising or falling
RCLK_EDGE	**RISE** FALL NONE	RClock can be rising, falling or not used

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_RAM_DQ	Generic Dual Port RAM category

Fanin Parameters

Parameter	Value	Description
RAMFANIN	AUTO MANUAL	See Fanin Control section below

Parameter Rules

Parameter Rules
If RCLK_EDGE is NONE (Asynchronous mode), then RE_POLARITY must be 2 (note used)

Fan-in Control

One of the key issues when building RAM macros is to control the routing congestion near the RAM cells. The problem becomes more critical when deep RAM macros are built. You need to broadcast signals throughout the height of the chip. The place & route algorithm could have difficulties satisfying all routing constraints. As a result, much slower routing resources could be allocated to satisfy all constraints. To make this problem less likely, a special buffering scheme has been implemented to relieve the congestion near the RAM cells. However, you may choose to control the buffering yourself to improve performances when needed. The RAM macro can be built using either the automatic buffering architecture or the manual buffering architecture.

Automatic Buffering

In this mode (default), a buffering scheme is automatically built into the RAM macro architecture (see Figure 4); this mode should always be considered first. However, if the performance is not met, it may be better to use the manual buffering option.

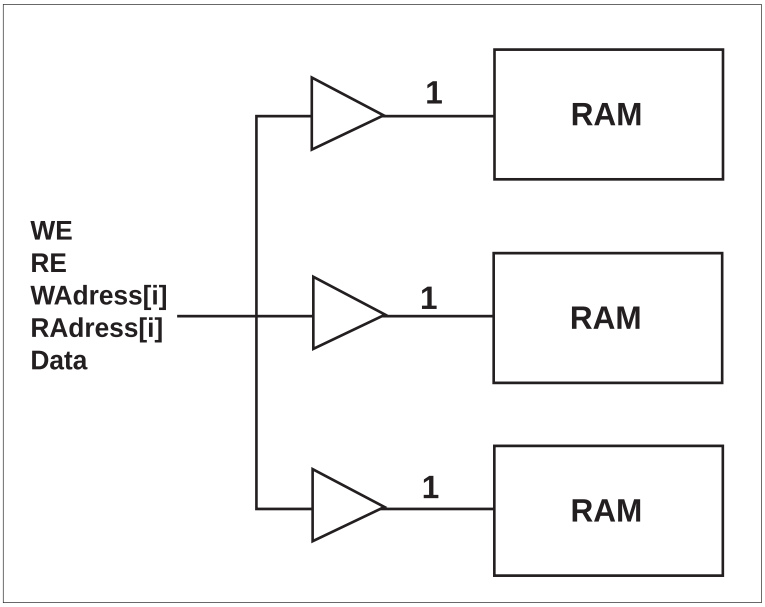


Figure 4. Automatic Buffering for RAM Macros

Manual Buffering

Figure 5 shows how manual buffering is done. A fanin of one (1) is enforced on all signals fanning out to more than one RAM cell. If these signals were broadcasted to all RAM cells, very slow routing resources (long freeways) would be required to route the signals impacting the RAM performance.

Manual buffering should only be used if the expected performance is not realized using the automatic buffering scheme, or if you know ahead of time that you need to use this scheme to meet your timing goals. In this architecture, the idea is to not buffer the signals internally but rather give some kind of access to the RAM macro internal signals. Then, you must buffer the signals outside the macro and either use traditional buffers or duplicate the logic that drives these signals externally. If manual buffering is chosen, the *WE*, *RE*, *Waddress(i)*, *RAddress(i)* and *Data[i]* signals become busses external to the macro. For all these signals, the bus width is equal to the number of RAM cells (used to build a given configuration) driven by each signal. Figure 5 illustrates the manual buffering architecture for a 96x8 RAM configuration, built of three 32x8 configured RAM cells. In this configuration, the *WE*, *RE*, *WAddress* and *RAddress* signals drive all RAM cells simultaneously. Figure 6 shows a 128x8 RAM configuration, built using four 64x4 configured RAM cells. In that configuration, the 8-bit data bus is split into two completely independent 4-bit data busses.

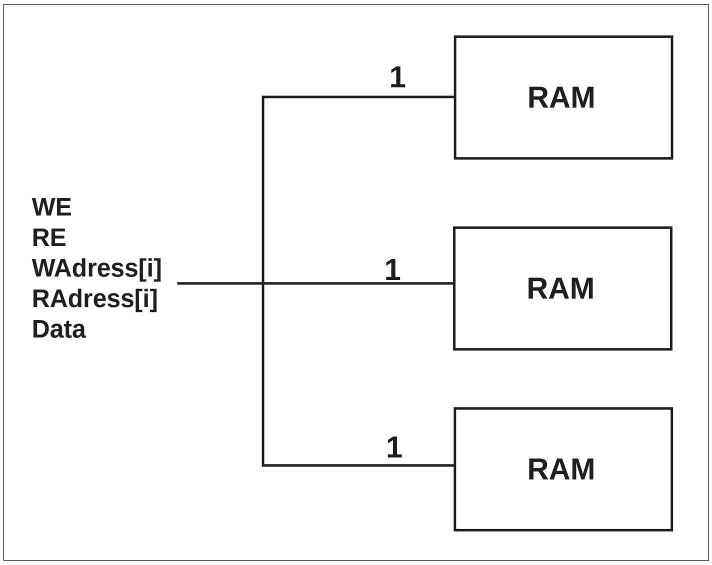


Figure 5. Manual Buffering (96x8 RAM Configuration)

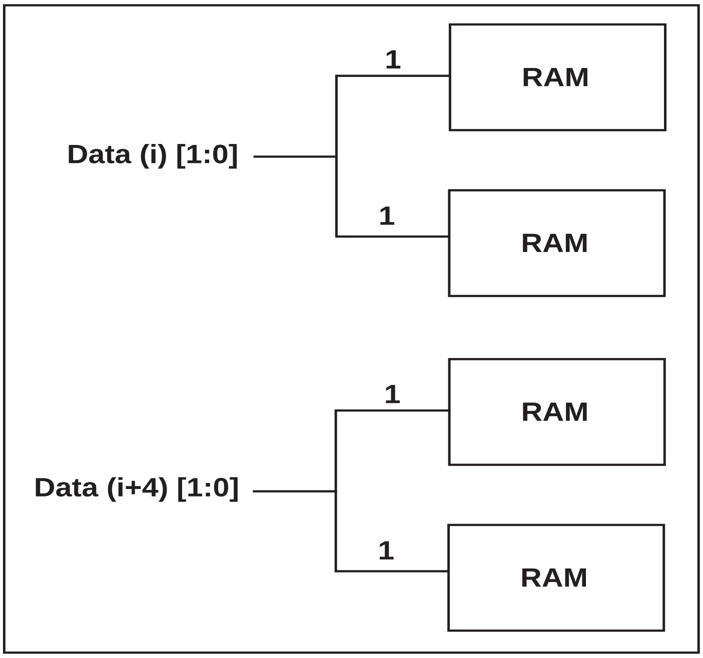


Figure 6. Manual Buffering for the Data Bus (128x8 RAM Configuration)

Timing Waveforms

Timing Waveform Terminology

Term	Description	Term	Description
t_{ckhl}	Clock high/low period	t_{dsu}	Data setup time
t_{rp}	Reset pulse width	t_{rco}	Data valid after clock high/low
t_{wesu}	Write enable setup time	t_{rao}	Data valid after read address has changed
t_{resu}	Read enable setup time	t_{co}	Flip-flop clock to output

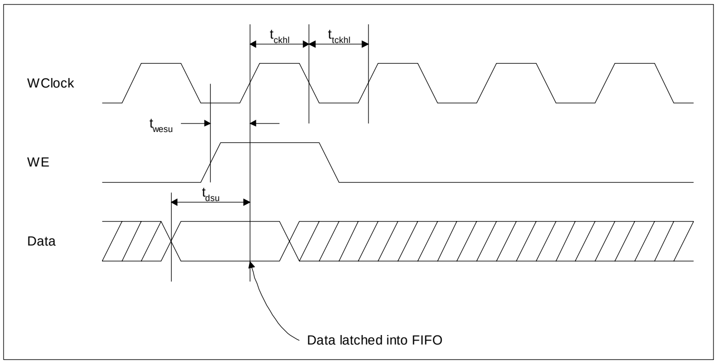


Figure 7. RAM Write Cycle

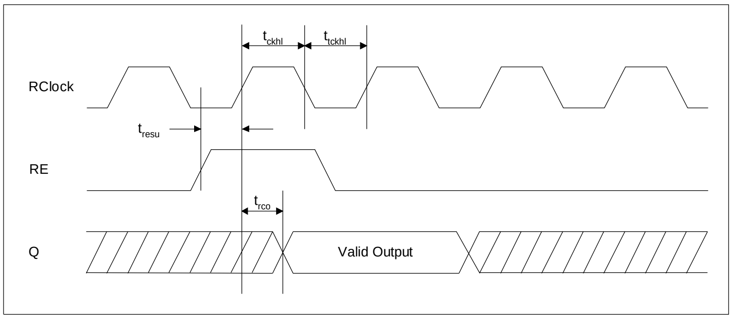


Figure 8. RAM Synchronous Read Cycle

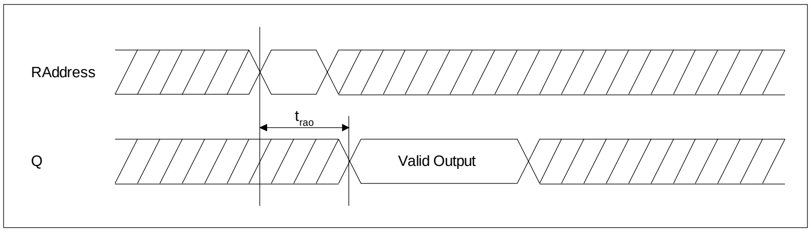
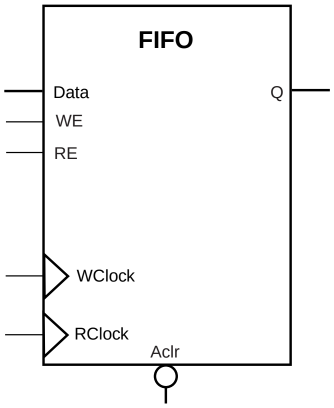


Figure 9. RAM Asynchronous Read Cycle

Synchronous FIFO with Independent Read and Write Functions

Features

- Parameterized word length and depth
- Dual port synchronous FIFO (write and read clocks are separated) with no static flag logic
- Global reset of FIFO address pointers



Family Support

3200DX, 42MX

Description

The ACTgen FIFO macros use the 3200DX and 42MX 32x8 or 64x4 dual-port RAM cells available on the chip. Addresses are generated internally using counters and token chains to address the RAM (this is transparent to the user). Dedicated read and write address data paths are used in the FIFO architecture. The read and write operations are totally independent and can be performed simultaneously.

The WIDTH (word length) and DEPTH (number of words) have continuous values but the choice of WIDTH limits the choice of DEPTH and vice versa.

The asynchronous clear signal, *Aclr*, can be active low or active high (low is the default option and is the preferred use for all synchronous elements in the two supported families). When the asynchronous clear is active, all internal registers used to determine the current FIFO read and write addresses (counters and token chains) are reset to “0.” The FIFO is now in an empty state; the RAM content is not affected. When power is first applied to the FIFO, the FIFO must be initialized with an asynchronous clear cycle to reset the internal address pointers.

The write enable *WE* and read enable *RE* signals are active high request signals for writing into and reading out of the FIFO respectively. The *WE* and *RE* signals only control the logic associated with the FIFO write and read address pointers.

When *WE* is asserted high, the write cycle is initiated, and Data are written into the FIFO. The design using the FIFO is responsible for handling the full and empty states of the FIFO macro.

When *RE* is asserted high, the read cycle is initiated, and Q is read from the FIFO. The design using the FIFO is responsible for handling the full and empty states of the FIFO macro.

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	WIDTH	input	Req.	Input Data
WE	1	input	Req.	Write Enable
RE	1	input	Req.	Read Enable
WClock	1	input	Req.	Write clock
RClock	1	input	Req.	Read clock
Q	WIDTH	output	Req.	Output Data

Parameter Description

Parameter	Value	Function
WIDTH	width	Word length of Data and Q
DEPTH	depth	Number of FIFO words
WCLK_EDGE	**RISE** FALL	WClock can be rising or falling
RCLK_EDGE	**RISE** FALL	RClock can be rising falling

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_FIFO_DQ	Generic FIFO category
LPM_HINT	SFIFO	Synchronous FIFO with no flags

Fanin Parameters

Parameter	Value	Description
RAMFANIN	**AUTO** MANUAL	See "Fan-in Control" on page 59

Timing Waveforms

Timing Waveform Terminology

Term	Description	Term	Description
$\mathbf{t}_{ckhl}$	Clock high/low period	$\mathbf{t}_{dsu}$	Data setup time
$\mathbf{t}_{rp}$	Reset pulse width	$\mathbf{t}_{rco}$	Data valid after clock high/low
$\mathbf{t}_{wesu}$	Write enable setup time	$\mathbf{t}_{co}$	Flip-flop clock to output
$\mathbf{t}_{resu}$	Read enable setup time		

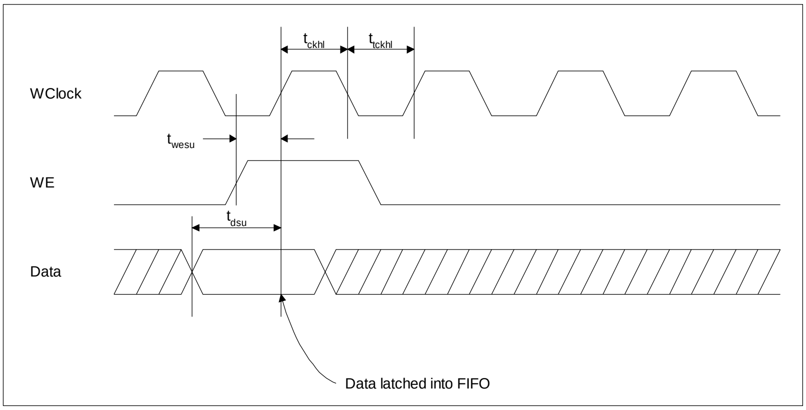


Figure 10. FIFO Write Cycle

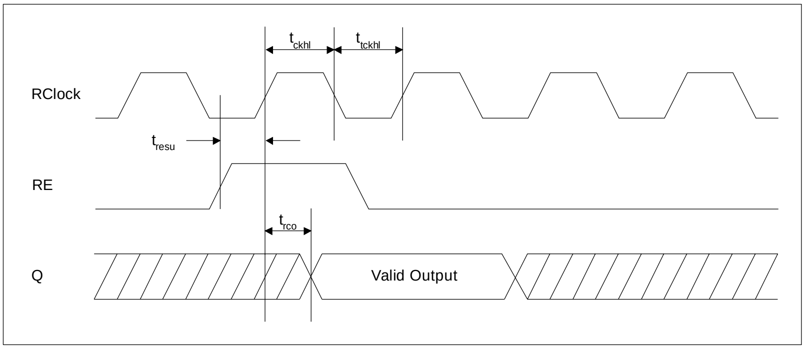
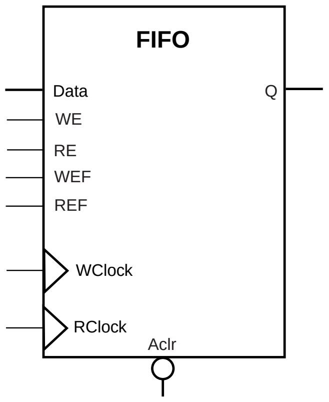


Figure 11. FIFO Read Cycle

Synchronous FIFO With Static Flag Logic

Features



- Parameterized word length and depth
- FIFO full and empty flags
- Statically programmable almost-full flag to indicate when the FIFO macro reaches a specific level, usually when writing into the FIFO
- Statically programmable almost-empty flag to indicate when the FIFO macro reaches a specific level, usually when reading from the FIFO
- Separate read and write data paths (independent read and write clocks)
- Global reset of the FIFO address pointers and flag logic
- Dual port synchronous FIFO (write and read clocks are separated) with no static flag logic

Family support

3200DX, 42MX

Description

The ACTgen FIFO macros use the 3200DX and 42MX 32x8 or 64x4 dual-port RAM cells. Addresses are generated internally using counters and token chains to address the RAM (this is transparent to the user). Dedicated read and write address data paths are used in the FIFO architecture. The read and write operations are totally independent and can be performed simultaneously.

The WIDTH (word length) and DEPTH (number of words) have continuous values but the choice of WIDTH limits the choice of DEPTH and vice versa.

The asynchronous clear signal, *Aclr*, can be active low or active high (low is the default option and should be used for all synchronous elements in the two supported families). When the asynchronous clear is active, all internal registers used to determine the current FIFO read and write addresses (counters and token chains) are reset to "0." The FIFO is now in an empty state; the RAM content is not affected. When power is first applied to the FIFO, the FIFO must be initialized with an asynchronous clear cycle to reset the internal address pointers.

The full flag signal, *FF*, is optional and is available only for the High Speed Flag (FFIFO) and the Medium Speed Flag (MFFIFO) variations. The *FF* signal is active high only (if selected) and indicates when the FIFO is full. The signal is asserted high on the rising (RISE) or falling (FALL) edge of the clock signal *Clock* with no delay.

The empty flag signal, *EF*, is optional and is available only for the High Speed Flag (FFIFO) and the Medium Speed Flag (MFFIFO) variations. The *EF* signal is active low only (if selected) and indicates when the FIFO is empty. The signal is asserted low on the rising (RISE) or falling (FALL) edge of the clock signal *Clock* with no delay.

The write enable signals, *WE* and *WEF*, and read enable signals, *RE* and *REF*, are active high requests for writing into and reading out of the FIFO respectively. The *WE* and *RE* signals only control the logic associated with the FIFO write and read address pointers. The *WEF* and *REF* signals control the logic implementing the different flags. The *WE* and *WEF* signals should be logically driven by the same logic outside the FIFO macro. The same behavior applies to the *RE* and *REF* signals as well.

When *WE* is asserted high and *FF* is asserted low (not full), the write cycle is initiated and Data are written into the FIFO. When *WE* is asserted high and *FF* is asserted high (full), the FIFO behavior is undefined. When *RE* is asserted high and *EF* is asserted high (empty), the read cycle is initiated and Q is read from the FIFO. When *RE* is asserted high and *EF* is asserted low (empty), the FIFO behavior is undefined. When *RE* and *WE* are asserted high at the same time, Data are written into the FIFO and Q is read from the FIFO simultaneously. The read and write operations are fully synchronous with respect to the clock signal *Clock*.

The FIFO function offers a parameterizable almost-full flag, *AFF*. The *AFF* flag is asserted high when the FIFO contains aff_val words or more as defined by the parameter AFF_VAL. Otherwise, *AFF* is asserted low. The aff_val value is a parameter to the macro, and thus logic is built at generation time to realize the almost-full flag function.

The FIFO function offers a parameterizable almost-empty flag, *AEF*. The *AEF* flag is asserted low when the FIFO contains aef_val words or less as defined by the parameter AEF_VAL. Otherwise, *AEF* is asserted low. The aef_val value is a parameter to the macro, and thus logic is built at generation time to realize the almost-empty flag function.

Port Description

Port Name	**Size**	**Type**	**Req/Opt**	**Function**
Data	WIDTH	input	Req.	Input Data
WE	1	input	Req.	Write Enable with the FIFO only (noflag)
RE	1	input	Req.	Read Enable with the FIFO only (no flag)
WEF	1	input	Req.	Write enable associated with the flag logic only
REF	1	input	Req.	Read enable associated with the flag logic only
Clock	1	input	Req.	Write and read clock
Q	WIDTH	output	Req.	Output Data

Parameter Description

Parameter	**Value**	**Function**
WIDTH	width	Word length of Data and Q
DEPTH	depth	Number of FIFO words
FF_POLOARITY	**1** 2	FF can be active high or not
EF_POLARITY	**0** 2	EF can be active low or not used

Parameter Description (Continued)

Parameter	Value	Function
AFF_VAL	aff_val (see parameter rules)	AFF value (not used if aff_val is 0
AEF_VAL	aef_val (see parameter rules	AEF value (not used if aef_val is 0
CLK_EDGE	**RISE** FALL	Clock can be rising or falling

Implementation Parameters

Parameter	Value	Description
LPMTYPE	LPM_FIFO_DQ	Generic FIFO category
LPM_HINT	SFIFO	Synchronous FIFO with no flags
	FFIFO	High skpeed FIFO with flags
	MFFIFO	Medium speed FIFO with flags

Fanin Parameters

Parameter	Value	Description
RAMFANIN	**AUTO** MANUAL	See Fanin Control section below

Parameter Rules

Parameter Rules
If RCLK_EDGE is NONE (Asynchronous mode), then RE_POLARITY must be 2 (not used)

Timing Waveforms

Timing Waveform Terminology

Term	Description
t_{ckhl}	Clock high/low period
t_{rp}	Reset pulse width
t_{wesu}	Write enable setup time
t_{resu}	Read enable setup time
t_{adsu}	Data setup time
t_{rco}	Data valid after lock high/low
t_{rao}	Data valid after read address has changed
t_{co}	Flip-flop clock to output

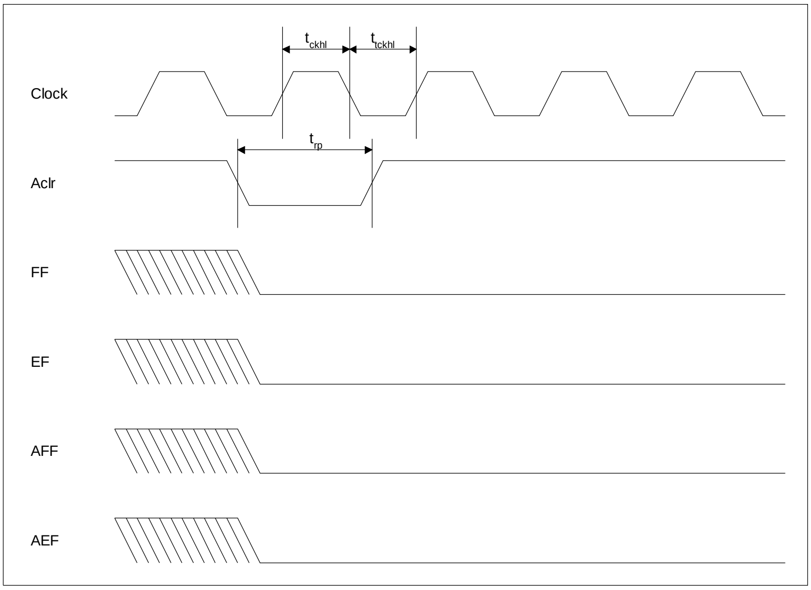


Figure 12. Reset Cycle

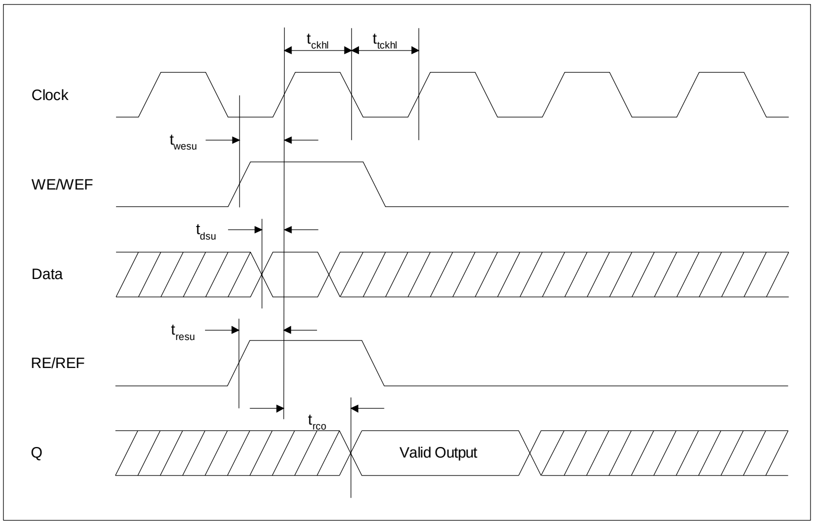


Figure 13. Write and Read Cycle

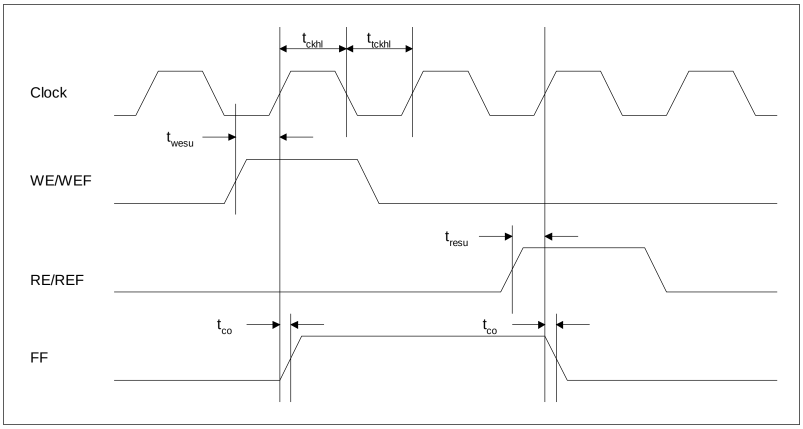


Figure 14. Full FIFO Timing Diagram

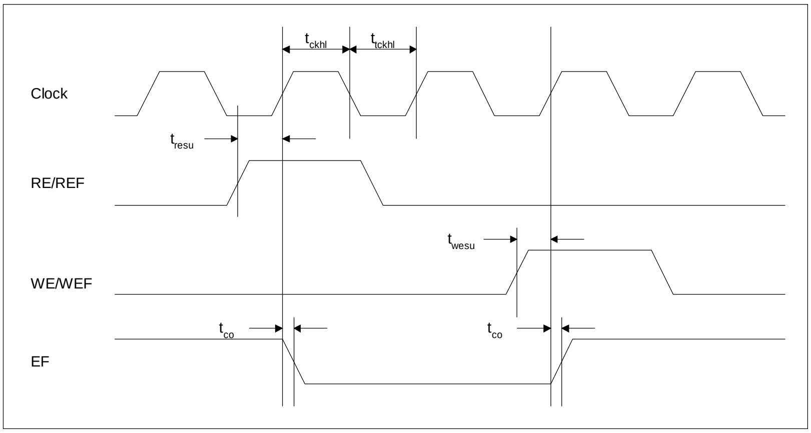


Figure 15. Empty FIFO Timing Diagram

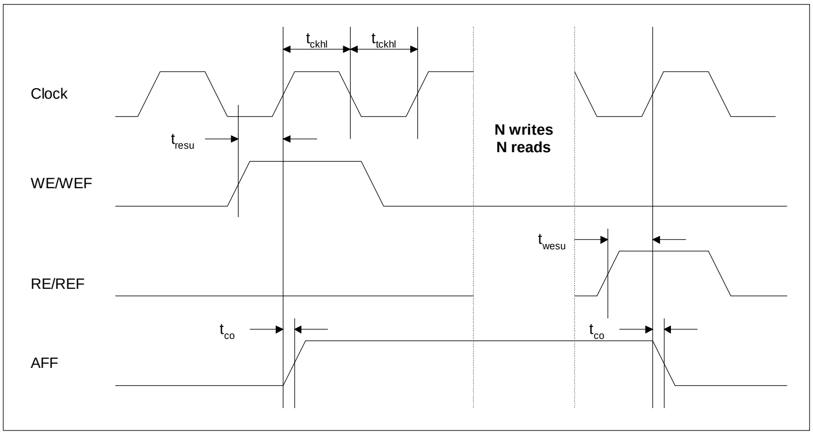
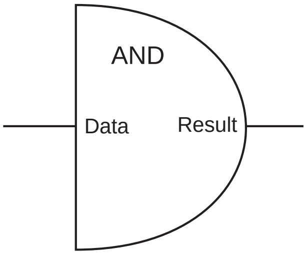


Figure 16. Almost Full FIFO Timing Diagram

Logic (AND)



Features

- Parameterized AND size
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	SIZE	input	Req.	Input data
Result	1	output	Req.	output

Parameter Description

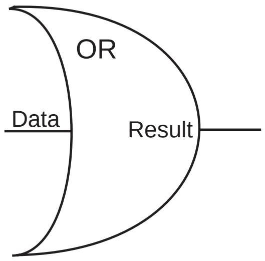
Parameter	Value	Function
SIZE	2-64	Word length of Data
RESULT_POLARITY	0 **1**	Output polarity (active low or active high)

Functional Description[a]

Data	Result
m	m[0] and m[1] and … and m[SIZE-1]

a. Result is active high.

Logic (OR)



Features

- Parameterized OR size
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	SIZE	input	Req.	input data
Result	1	output	Req.	output

Parameter Description

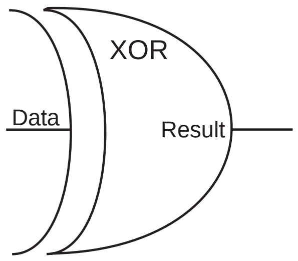
Parameter	Value	Function
SIZE	2-64	Word length of Data
RESULT_POLARITY	0 **1**	Output polarity (active low or active high)

Functional Description[a]

Data	Result
m	m[0] or m[1] or … or m[SIZE-1]

a. Result is active high.

Logic (XOR)



Features

- Parameterized XOR size
- Behavioral simulation model in VHDL and Verilog

Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	SIZE	input	Req.	input data
Result	1	output	Req.	output

Parameter Description

Parameter	Value	Function
SIZE	2-64	Word length of Data
RESULT_POLARITY	0 **1**	Output polarity (active low or active high)

Functional Description[a]

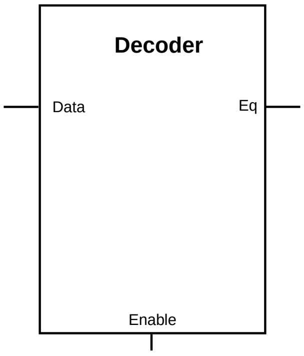
Data	Result
m	m[0] xor m[1] xor … xor m[SIZE-1]

a. Result is active high.

Decoder

Features

- Parameterized output size (DECODES)
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 42MX, 54SX

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
Data	decln[a]	input	Req.	Input data
Enable	1	input	Opt.	Enable
Eq	DECODES	output	Req.	output

a. decln is an integer and $\log_2$ (DECODES) = decln d<$\log_2$ (DECODES + 1. If decln is equal to 1, then Data is scalar, else Data is a bus.

Parameter Description

Parameter	Value	Function
DECODES	2-32	Word length of Eq
EN_POLARITY	0 1 **2**	Enable polarity (active high, active low or not used)
EQ_POLARITY	0 **1**	Eq polarity (active low or active high)

Functional Description[a]

Data	Enable	Eq
X	0	00…0
m	1	dec[b] (m)==decodes-1 &&[c] dec(m)==decodes-2 && … && dec(m)==0

a. Enable is active low and Eq is active high.

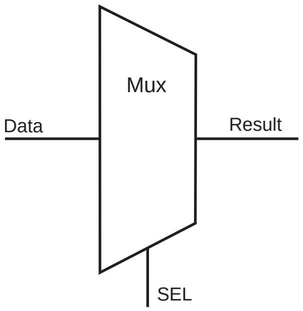
b. dec(m) defines the decimal value of m.

c. && indicates bity concatenation.

Multiplexer

Features

- Parameterized word length
- Parameterized multiplexer input number
- Behavioral simulation model in VHDL and Verilog



Family Support

ACT 1, ACT 2/1200XL, ACT 3, 3200DX, 40MX, 42MX, 54SX.

Description

Port Description

Port Name	Size	Type	Req/Opt	Function
Data_0	WIDTH	Input	Req.	Input data
Data_1	WIDTH	Input	Req.	Input data
…	…	…	…	…
$\text{Data}_{\text{SIZE-1}}$	WIDTH	Input	Req.	Input data
Sel_0	1	Input	Req.	Select line
Sel_1	1	Input	Req.	Select line
…	…	…	…	…
$\text{Sel}_{\text{SIZELN-1}}$	1	Input	Req.	Select line
Result	WIDTH	Output	Req.	output

Parameter Description

Parameter	Value	Function
WIDTH	2-32	Word length of Data
SIZE	1-32	Number of data inputs

Functional Description

$Data_0$	**$Data_1$**	**…**	**$Data_{SIZE-1}$**	**Sel_0**	**Sel_1**	**…**	**$Sel_{SIZELN-1}$**	**Result**
m_0	m_1	…	m_{SIZE-1}	0	0	…	0	m_0
m_0	m_1	…	m_{SIZE-1}	1	0	…	0	m_1
…	…	…	…	…	…	…	…	…
m_0	m_1	…	m_{SIZE-1}	1	1	…	1	m_{SIZE-1}

Product Support

Actel backs its products with various support services including Customer Service, a Customer Applications Center, a Web and FTP site, electronic mail, and worldwide sales offices. This appendix contains information about using these services and contacting Actel for service and support.

Actel U.S. Toll-Free Line

Use the Actel toll-free line to contact Actel for sales information, technical support, requests for literature about Actel and Actel products, Customer Service, investor information, and using the Action Facts service.

The Actel Toll-Free Line is (888) 99-ACTEL.

Customer Service

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

From Northeast and North Central U.S.A., call (408) 522-4480.
From Southeast and Southwest U.S.A., call (408) 522-4480.
From South Central U.S.A., call (408) 522-4434.
From Northwest U.S.A., call (408) 522-4434.
From Canada, call (408) 522-4480.
From Europe, call (408) 522-4252 or +44 (0) 1256 305600.
From Japan, call (408) 522-4743.
From the rest of the world, call (408) 522-4743.
Fax, from anywhere in the world (408) 522-8044.

Customer Applications Center

The Customer Applications Center is staffed by applications engineers who can answer your hardware, software, and design questions.

All calls are answered by our Technical Message Center. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:30 a.m. to 5 p.m., Pacific Standard Time, Monday through Friday.

The Customer Applications Center number is (800) 262-1060.

European customers can call +44 (0) 1256 305600.

Guru Automated Technical Support

Guru is a Web based automated technical support system accessible through the Actel home page (**http://www.actel.com/guru/**). Guru provides answers to technical questions about Actel products. Many answers include diagrams, illustrations and links to other resources on the Actel Web site. Guru is available 24 hours a day, seven days a week.

Web Site

Actel has a World Wide Web home page where you can browse a variety of technical and non-technical information. Use a Net browser (Netscape recommended) to access Actel’s home page.

The URL is **http://www.actel.com**. You are welcome to share the resources we have provided on the net.

Be sure to visit the “Actel User Area” on our Web site, which contains information regarding: products, technical services, current manuals, and release notes.

FTP Site

Actel has an anonymous FTP site located at **ftp://ftp.actel.com**. You can directly obtain library updates, software patches, design files, and data sheets.

Electronic Mail

You can communicate your technical questions to our e-mail address and receive answers back by e-mail, fax, or phone. Also, if you have design problems, you can e-mail your design files to receive assistance. The e-mail account is monitored several times per day.

The technical support e-mail address is **tech@actel.com**.

Worldwide Sales Offices

Headquarters

Actel Corporation
955 East Arques Avenue
Sunnyvale, California 94086
Toll Free: 888.99.ACTEL

Tel: 408.739.1010
Fax: 408.739.1540

US Sales Offices

California

Bay Area
Tel: 408.328.2200
Fax: 408.328.2358

Irvine
Tel: 949.727.0470
Fax: 949.727.0476

San Diego
Tel: 619.938.9860
Fax: 619.938.9887

Thousand Oaks
Tel: 805.375.5769
Fax: 805.375.5749

Colorado

Tel: 303.420.4335
Fax: 303.420.4336

Florida

Tel: 407.677.6661
Fax: 407.677.1030

Georgia

Tel: 770.831.9090
Fax: 770.831.0055

Illinois

Tel: 847.259.1501
Fax: 847.259.1572

Maryland

Tel: 410.381.3289
Fax: 410.290.3291

Massachusetts

Tel: 978.244.3800
Fax: 978.244.3820

Minnesota

Tel: 612.854.8162
Fax: 612.854.8120

North Carolina

Tel: 919.376.5419
Fax: 919.376.5421

Pennsylvania

Tel: 215.830.1458
Fax: 215.706.0680

Texas

Tel: 972.235.8944
Fax: 972.235.965

International Sales Offices

Canada
Suite 203
135 Michael Cowpland Dr,
Kanata, Ontario K2M 2E9

Tel: 613.591.2074
Fax: 613.591.0348

France
361 Avenue General de Gaulle
92147 Clamart Cedex

Tel: +33 (0)1.40.83.11.00
Fax: +33 (0)1.40.94.11.04

Germany
Bahnhofstrasse 15
85375 Neufahrn

Tel: +49 (0)8165.9584.0
Fax: +49 (0)8165.9584.1

Hong Kong
Suite 2206,
Parkside Pacific Place,
88 Queensway

Tel: +011.852.2877.6226
Fax: +011.852.2918.9693

Italy
Via Giovanni da Udine No. 34
20156 Milano

Tel: +39 (0)2.3809.3259
Fax: +39 (0)2.3809.3260

Japan
EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150

Tel: +81 (0)3.3445.7671
Fax: +81 (0)3.3445.7668

Korea
135-090, 18th Floor,
Kyoung Am Building
157-27 Samsung-dong
Kangnam-ku, Seoul

Tel: +82 (0)2.555.7425
Fax: +82 (0)2.555.5779

Taiwan
4F-3, No. 75, Sec. 1,
Hsin-Tai-Wu Road,
Hsi-chih, Taipei, 221

Tel: +886 (0)2.698.2525
Fax: +886 (0)2.698.2548

United Kingdom
Daneshill House,
Lutyens Close
Basingstoke,
Hampshire RG24 8AG

Tel: +44 (0)1256.305600
Fax: +44 (0)1256.355420