Logos2 Family FPGA Arithmetic Process Module (APM) User Guide

(UG040003, V1.3)

(17.07.2023)

Shenzhen Pango Microsystems Co., Ltd.

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.3	17.07.2023	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
APM	Arithmetic Process Module
MAC	Multiply Accumulate
SIMD	Single-instruction-multiply-data

Table of Contents

Tables

Figures

Chapter 1 General Introduction

Logos2 Family products are equipped with APM (Arithmetic Process Module), which provides efficient digital signal processing capabilities.

Distributed by column in Logos2 products, APM has the following main features:

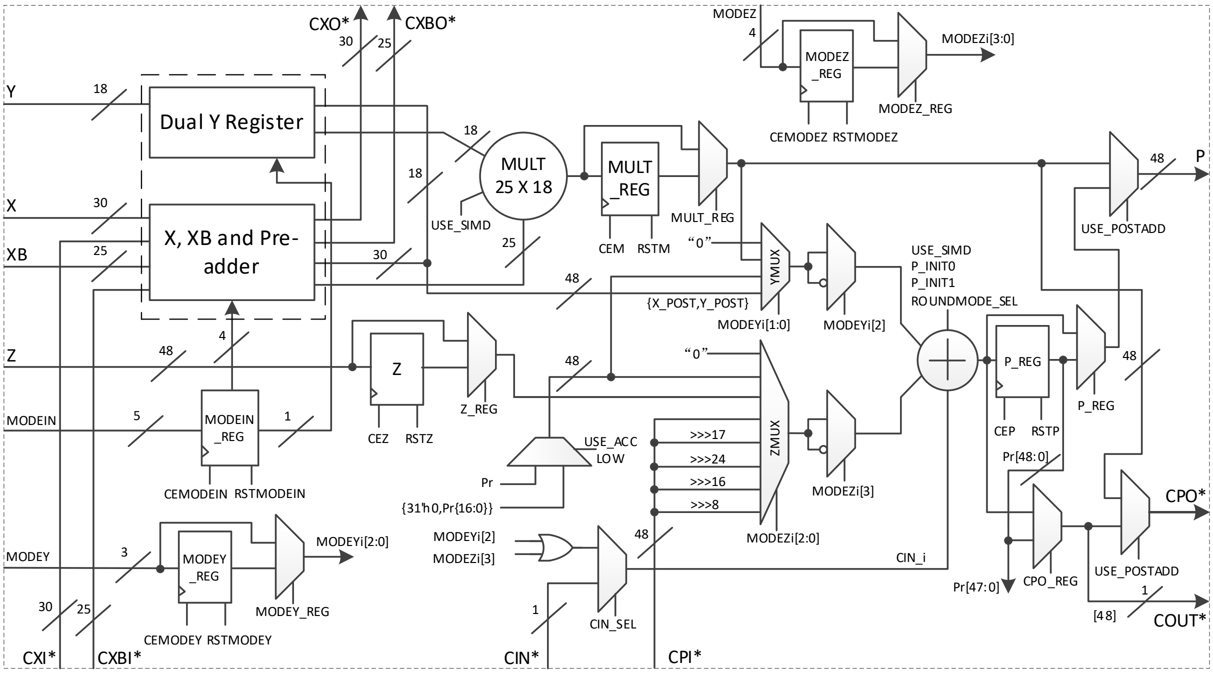
- Signed multiplier 25×18, unsigned multiplication is achieved by assigning 0 to the high bits
- All calculations and output results are signed, including the sign bit
- One 48-bit addition/subtraction/accumulation operation or two 24-bit operations supported
- With a 25-bit Preadder
- Independent optional CE and RST
- Input cascade chains supported
- Output cascade chains supported
- Control/data signal pipeline
- Dynamic mode switching supported
- Rounding function

APM can be implemented with the Pango Design Suite software (hereinafter referred to as PDS). For user convenience, IP is generated by the IP Compiler tool embedded in PDS. The PDS software integrates the IP User Guide. For the specific usage of IPs, refer to the user guide after selecting the APM usage mode.

Chapter 2 Function Description

2.1 APM Structure

APM primarily consists of four functional units: I/O Unit, Preadd Unit, Mult Unit, and Postadd Unit. Figure 2-1 shows the block diagram of overall APM functions.



Note: Signals marked with * are internal cascade signals.

Figure 2-1 Block Diagram of Overall APM Functions

The input structures of X, XB, Pre-adder and dual-Y port register in the dashed boxes in Figure 2-1 are shown in Figure 2-2 Table 2-2and Figure 2-3.

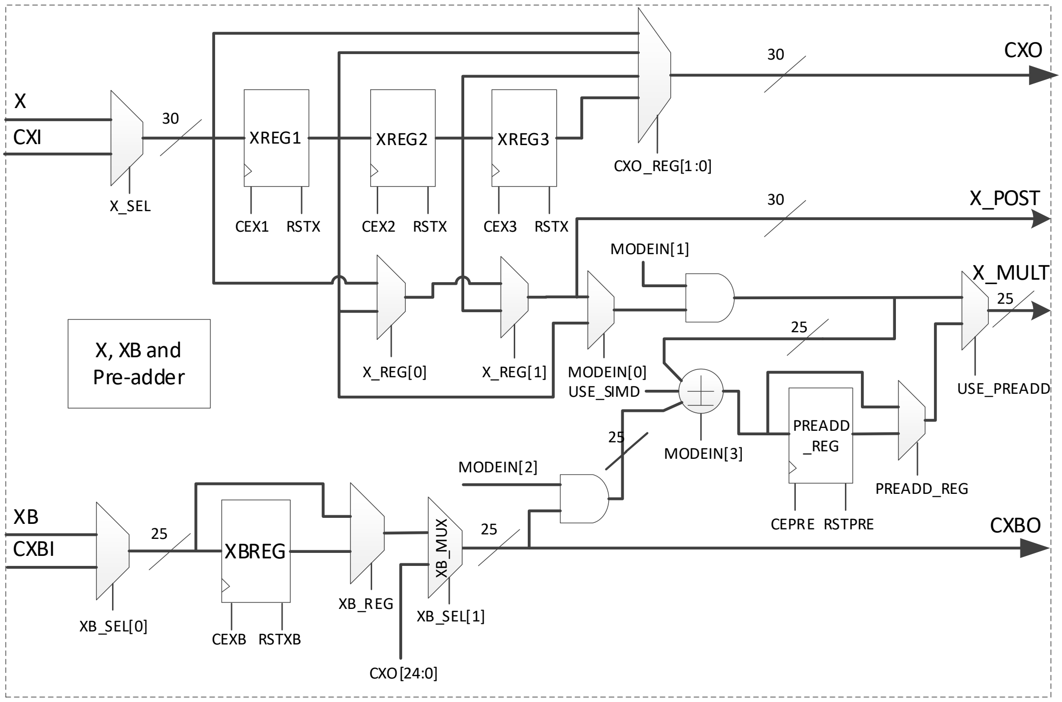


Figure 2-2 Block Diagram of X, XB, and Pre-adder Functions

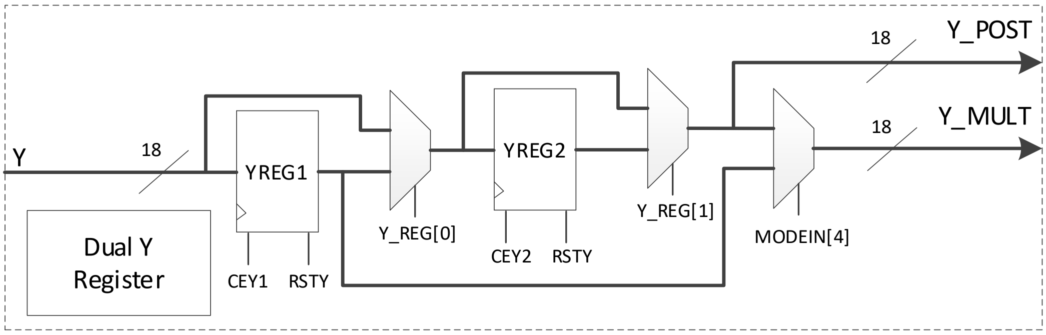


Figure 2-3 Block Diagram of Port Y Functions

APM includes a multiplier followed by an adder. APM requires the use of all pipeline registers when operating at full speed.

- APM supports cascade mode, with common applications including FFT, FIR, multiplication, addition or subtraction operations, and large bit-width multiplication.
- APM also provides functions such as synchronous reset, clock enable, dual input registers for the X port, SIMD, Direct Rounding, and Round-toward-nearest.
- 8/16/17/24-bit right shift functions can achieve larger bit-width multiplication.
- Optional input, pipeline, pre-add, output, multiplication, and postadd registers.
- Optional control signal (MODEY, MODEZ, MODEIN) registers.

For APM resources and performance, refer to the APM section of the DS04001_Logos2 Family FPGA Device Datasheet.

2.1.1 I/O Unit

- Implements data in/out registration.
- Each register's CE signals (active-high) and RST signals (active-high) are controlled by their respective independent CE and RST inputs. The asynchronous/synchronous function of RST is determined by the shared parameter ASYNC.
- Mode ports are divided into three groups: MODEIN[4:0], MODEY[2:0], and MODEZ[3:0] according to the control function. Each group of mode ports is controlled by its own independent REG, RST, and CE parameters.
- APM cascading is used to implement FIR and high-bit-width multipliers such as 49*35.
- Simplifies routing for constant input and repeated sign bits.
- Port X supports 30-bit binary operations, and Port XB[24:0] supports PREADD_MULTADD operations. The MODEIN control terminal supports time-division multiplexing functions.
- For the 30-bit data of Port X, the lower 25 bits are used for the multiplier, and the entire 30 bits form the higher 30 bits of the 48-bit X:Y bit-width data.
- The multi-stage registers for X and Y inputs can be controlled independently.
- Independent Z input and input register.
- CIN and COUT are internal cascade signals that support 96-bit multiplication and addition/subtraction between two APMs.

2.1.2 Preadd Unit

- The hardware implementation is 25±25, producing a 25-bit result; or two 12±12, each producing a 12-bit result (when USE_SIMD=1).
- The pre-add function can be bypassed (USE_PREADD=0).
- With output register

2.1.3 Mult Unit

- Implements a 25×18 with the result sign-extended to 48 bits; or two 12×9 (USE_SIMD=1), with each 12×9 result sign-extended to 24 bits.
- All operands are signed numbers.

2.1.4 Postadd Unit

- Implements a 48-bit addition or two 24-bit additions. Subtraction operation P=X-Y is converted to addition P=X+~Y+1 for computation.
- POSTADD supports the rounding function, with ROUNDMODE, PINIT0, and PINIT1 parameters set. Refer to 2.4.1 for details.
- POSTADD can be bypassed (when POSTADD=0), at which time the APM outputs the result of the multiplier.

2.2 GTP Description

The GTP_APM_E2 unit integrates various working modes of APM, allowing users to select different modes and implement APM functions by configuring interfaces and parameters. For the default values of each register, refer to UG050007_Logos2 Family Product GTP User Guide.

2.2.1 Block Diagram of GTP Structure

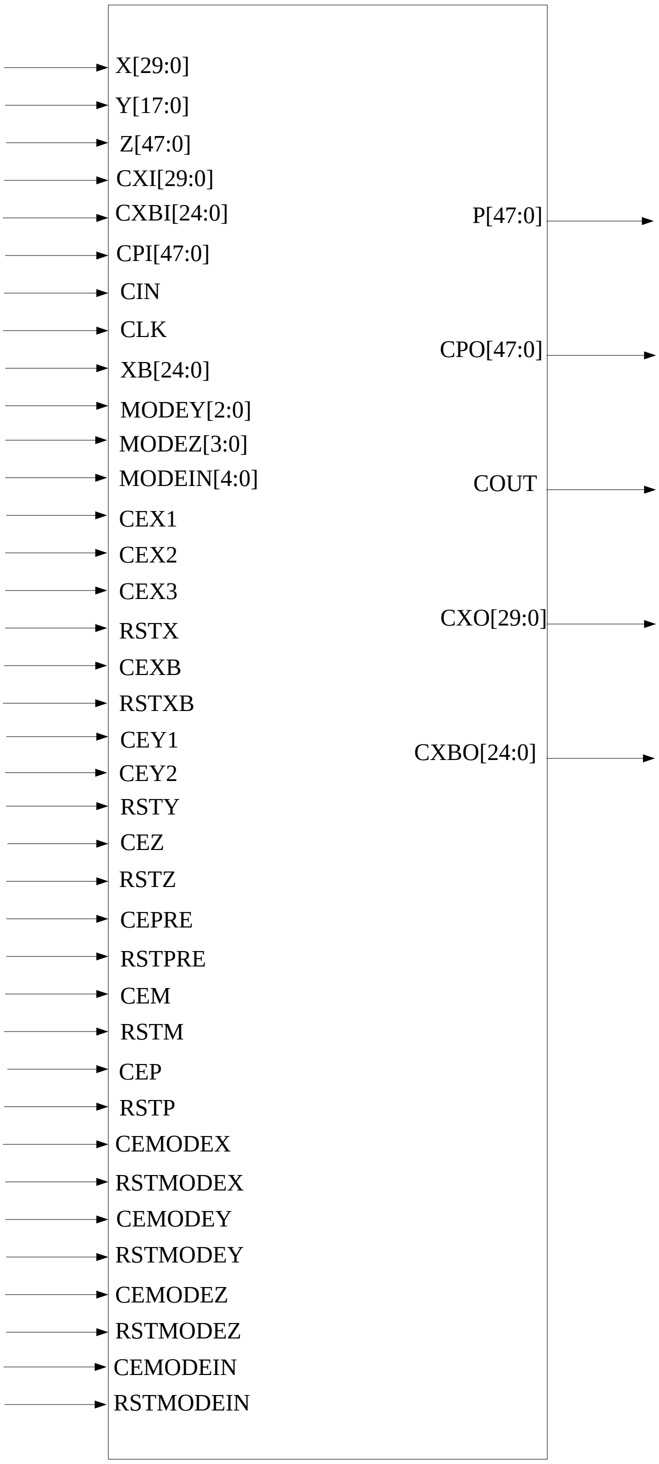


Figure 2-4 GTP_APM_E2 Structure Diagram

2.2.2 Port Description

Table 2-1 GTP_APM_E2 Port Description List

Port Name	Input/Output	Description
X[29:0]	Input	Parallel data input; X[24:0] serves as the X input of the multiplier or pre-adder, and X[29:0] serves as the higher 30 bits of X:Y, which can be used as the input data for the Postadder.
CXI[29:0]	Input	Cascade X input, from the CXO ports of other APM modules
CXBI[24:0]	Input	Cascade XB input, from the CXBO ports of other APM modules
XB[24:0]	Input	Parallel data input XB, as the input of the preadder, X±XB serves as the output of the preadder, controlled by MODEIN[3]
Y[17:0]	Input	Parallel data input Y, Y input of the multiplier, Y[17:0] serves as the lower 18 bits of X:Y, which can be used as the input data for the postadder
Z[47:0]	Input	Parallel data input Z, input data for the Postadder
CPI[47:0]	Input	Cascade P input, from the CPO port of the previous APM module
CIN	Input	Cascade CIN carry input, from the COUT port of the previous APM module
MODEY[2:0]	Input	APM dynamic Y-port control operator. Refer to 2.2.4 Mode Descriptions for function details.
MODEZ[3:0]	Input	APM dynamic Z-port control operator. Refer to 2.2.4 Mode Descriptions for function details.
MODEIN[4:0]	Input	APM dynamic input control operator, refer to 2.2.4 Mode Descriptions for function details.
CLK	Input	Clock input, used for all internal registers
CEX1	Input	Clock enable signal for XREG1 (the first X-port input register), used when X_REG=1 or 3
CEX2	Input	Clock enable signal for XREG2 (the second X-port input register), used when X_REG=2 or 3 and MODEIN[0]=0
CEX3	Input	Clock enable signal for XREG3 (the third X-port input register), used when CXO_REG=3
RSTX	Input	Active-high X reset signal to reset all X-port input registers
CEXB	Input	Clock enable signal for XB-port registers
RSTXB	Input	Active-high XB reset signal to reset XB input registers
CEY1	Input	Clock enable signal for YREG1 (the first Y-port input register), used when Y_REG=1 or 3
CEY2	Input	Clock enable signal for YREG2 (the second Y-port input register), used when Y_REG=2 or 3 and MODEIN[4]=0
RSTY	Input	Active-high Y reset signal to reset all Y-port input registers
CEZ	Input	Clock enable signal for Z-port input registers
RSTZ	Input	Active-high Z reset signal to reset Z-port input registers
CEPRE	Input	Clock enable signal for PREADD registers
RSTPRE	Input	Active-high PREADD reset signal to reset PREADD registers
CEM	Input	Clock enable signal for MULT registers
RSTM	Input	Active-high MULT register reset signal to reset MULT registers
CEP	Input	Clock enable signal for Postadd registers
RSTP	Input	Active-high P reset signal to reset Postadd registers
CEMODEIN	Input	Clock enable signal for MODEIN registers
RSTMODEIN	Input	Active-high MODEIN reset signal to reset MODEIN registers

Port Name	Input/Output	Description
CEMODEY	Input	Clock enable signal for MODEY registers
RSTMODEY	Input	Active-high MODEY reset signal to reset MODEY registers
CEMODEZ	Input	Clock enable signal for MODEZ registers
RSTMODEZ	Input	Active-high MODEZ reset signal to reset MODEZ registers
P[47:0]	Output	48-bit parallel data output for APM
CPO[47:0]	Output	Cascaded P output to connect to other APMs' CPI
COUT	Output	Cascaded CIN output to connect to other APMs' CIN
CXO[29:0]	Output	Cascaded X output to connect to other APMs' CXI
CXBO[24:0]	Output	Cascaded XB output to connect to the other APMs' CXBI

2.2.3 Parameter Description

Table 2-2 GTP_APM_E2 Parameter Description List

Parameter	Valid Values	Defaults	Function Description
USE_POSTADD	0,1	0	Postadd function enable 0: Not enabled 1: Enabled
USE_PREADD	0,1	0	Preadd function enable 0: Not enabled 1: Enabled
USE_MULT	0,1	1	Multiplier enable 0: Not enabled 1: Enabled
CXO_REG	0,1,2,3	0	Cascaded X output register delay selection; 0/1/2/3 corresponds to 0/1/2/3-stage output register, and the corresponding register needs to be enabled. Refer to Table 2-2Table 2-2Table 2-2 for detailed structure.
X_REG	0,1,2,3	0	X-port register enable; when it is 0 and MODEIN[0]=0, X_REG is not used; when it is 1, XREG1 is used; when it is 2 or 3 and MODEIN[0]=0, XREG1 and XREG2 are used; the corresponding register needs to be enabled. When MODEIN[0]=1. XREG1 is used, independent of this parameter; the corresponding register needs to be enabled.
XB_REG	0,1	0	XB-port register enable 0: Not enabled 1: Enabled
Y_REG	0,1,2,3	0	Y-port register enable; when it is 0 and MODEIN[4]=0, YREG is not used; when it is 1, YREG1 is used; when it is 2 and MODEIN[4]=0, YREG2 is used; when it is 3 and MODEIN[4]=0, YREG1 and YREG2 are used; corresponding registers need to be enabled. When MODEIN[4]=1, Y_REG1 is used, independent of this parameter; the corresponding register needs to be enabled.
Z_REG	0,1	0	Z-port register enable 0: Not enabled 1: Enabled
PREADD_REG	0,1	0	PREADD register enable 0: Not enabled 1: Enabled
MULT_REG	0,1	0	Mult register enable

Parameter	Valid Values	Defaults	Function Description
			0: Not enabled 1: Enabled
P_REG	0,1	0	P-port output register enable 0: Not enabled 1: Enabled
MODEIN_REG	0,1	0	MODEIN register enable 0: Not enabled 1: Enabled
MODEY_REG	0,1	0	MODEY register enable 0: Not enabled 1: Enabled
MODEZ_REG	0,1	0	MODEZ register enable 0: Not enabled 1: Enabled
X_SEL	0,1	0	X input selection: 0: Select X input 1: Select CXI cascade input
XB_SEL	0,1,2,3	0	XB_MUX backward transfer selection: 0: XB direct input 1: CXBI cascade input 2/3: Select CXO[24:0]
ASYNC_RST	0,1	0	Asynchronous reset enable 0: Select synchronous reset 1: Select asynchronous reset
USE_SIMD	0,1	0	SIMD mode selection; SIMD mode is enabled when it is 1
P_INIT0	48'h0-48'hffffffffffff	48'h0	When the Rounding function is not used, set "P_INIT0" to 48'h0; To use the Rounding function, set it up in accordance with 2.4.1 Rounding Function.
P_INIT1	48'h0-48'hffffffffffff	48'h0	When the Rounding function is not used: if the post-add function is also not used, set "P_INIT1" to 48'h0; if the post-add function is used, set "P_INIT1" as the initial value for the post-add operation; To use the Rounding function, set it up in accordance with 2.4.1 Rounding Function.
ROUNDMODE_SEL	0,1	0	Round mode parameters, used in conjunction with "P_INIT0" and "P_INIT1"; When the Rounding function is not used, set "ROUNDMODE_SEL" to 0; To use the Rounding function, set it up in accordance with 2.4.1 Rounding Function.
CPO_REG	0,1	0	CPO, COUT output register enable
USE_ACCLOW	0,1	0	Configure Postadder feedback 0: Use all 48 bits 1: Use only the lower 17 bits
CIN_SEL	0,1	0	Select carry CIN_i input source 0: MODEY[2] | MODEZ[3] 1: CIN
GRS_EN	"TRUE", "FALSE"	"TRUE"	Global reset enable "FALSE": Not enabled "TRUE": Enabled

2.2.4 Mode Descriptions

APM includes a 25-bit preadder and a 25-bit * 18-bit binary complement multiplier. Their data paths are selected by MODEIN, MODEY, and MODEZ, and then connect to the postadd unit. APM inputs are connected to each arithmetic unit, and users can manually select X- and Y-port inputs for a single or two register stages to meet different APM applications. Users can select XB- and Z-port inputs, and control signal inputs for a single register stage. To achieve the maximum speed specified in the Data Sheet, enable all pipeline registers.

The operating mode of the APM is controlled by three sets of signals: MODEIN[4:0], MODEY[2:0], and MODEZ[3:0]. Table 2-3 shows their port descriptions, and Table 2-4 lists the MODEIN functions (control the preadder to select the XB data). Two-stage X1, X2 and Y1, Y2 registers can achieve consistent functional data delays under time-division multiplexing.

Table 2-3 MODE Port Function Description

Signal	**Description**
MODEIN[4]	0: Y2 1: Y1
MODEIN[3]	Preadder add/subtract selection 0: Add 1: Subtract
MODEZ[3]	Bitwise inversion of ZMUX output, 1 is valid (then adding 1 through CIN_i in the postadder to achieve negation)
MODEZ[2:0]	ZMUX output selection 0: Zero 1: Postadder feedback 2: Z input 3: Cascaded output of the previous-stage APM 4: Cascaded output of the previous-stage APM, which shifts right by 17 bits 5: Cascaded output of the previous-stage APM, , which shifts right by 24 bits 6: Cascaded output of the previous-stage APM, which shifts right by 16 bits 7: Cascaded output of the previous-stage APM, which shifts right by 8 bits
MODEY[2]	Bitwise inversion of YMUX output, 1 is valid (then adding 1 through CIN_i in the postadder to achieve negation)
MODEY[1:0]	YMUX output selection 0: Zero 1: Multiplier output 2: Postadder feedback 3: Large bit-width data {X[29:0], Y[17:0]}

Note: For detailed structures of the functions in the table, refer to Figure 2-1.

Table 2-4 MODEIN Function List

MODEIN [3]	MODEIN [2]	MODEIN [1]	MODEIN [0]	"PREADD"	X_MULT
x	x	0	0	0	zero
x	x	0	1	0	zero
x	x	1	0	0	X2
x	x	1	1	0	X1
0	0	0	0	1	zero
0	0	0	1	1	zero
0	0	1	0	1	X2
0	0	1	1	1	X1
0	1	0	0	1	XB
0	1	0	1	1	XB
0	1	1	0	1	X2+XB
0	1	1	1	1	X1+XB
1	0	0	0	1	zero
1	0	0	1	1	zero
1	0	1	0	1	X2
1	0	1	1	1	X1
1	1	0	0	1	-XB
1	1	0	1	1	-XB
1	1	1	0	1	X2-XB
1	1	1	1	1	X1-XB

2.3 APM Operating Modes

2.3.1 Multiplication Mode

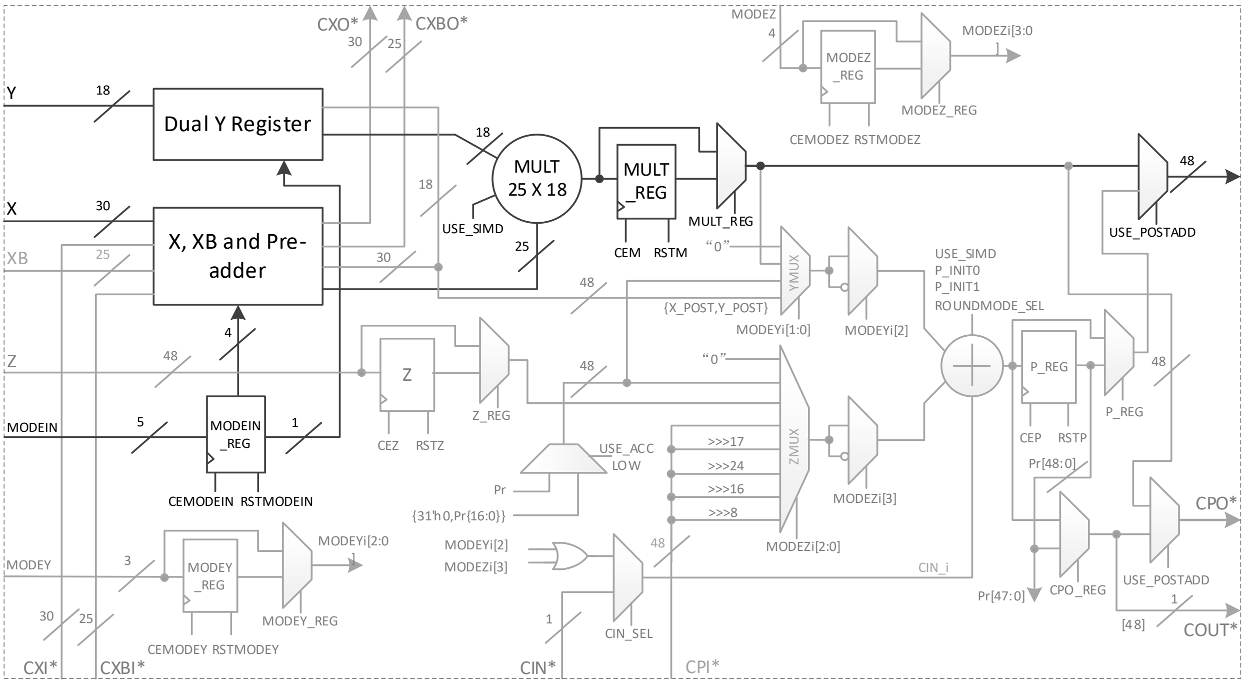


Figure 2-5 Multiplication Mode Application Diagram

As shown in Figure 2-5, when APM is configured in multiplication mode, its implementation expression is:

P=X×Y

The main features of multiplication mode include:

- Each APM can perform two 12×9 operations (USE_SIMD=1) or one 25×18 operation (USE_SIMD=0).
- Optional input/output register

After enabling the Preadd Unit in APM, APM can be configured into a pre-addition multiplication mode through Port XB, with the arithmetic expression as follows:

P=Y×(X±XB)

In this mode, each APM can perform two (12±12)×9 operations or one (25±25)×18 operation, supporting dynamic/static pre-add/subtract control. In addition to the optional input/output registers, users can also choose to enable the internal pipeline registers in the pre-addition multiplication mode, as illustrated in Figure 2-6:

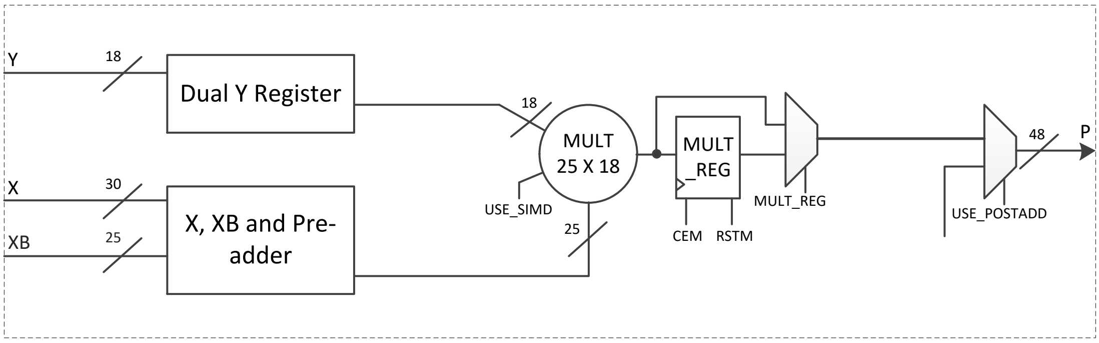


Figure 2-6 Application Diagram of Pre-addition Multiplication Mode

Figure 2-7 shows the typical timing of multiplication mode, where P=Y×(X+XB). When only the one-stage register is enabled, the P-port calculation result corresponding to the X-port and Y-port input data will be output on the next rising edge of the clock.

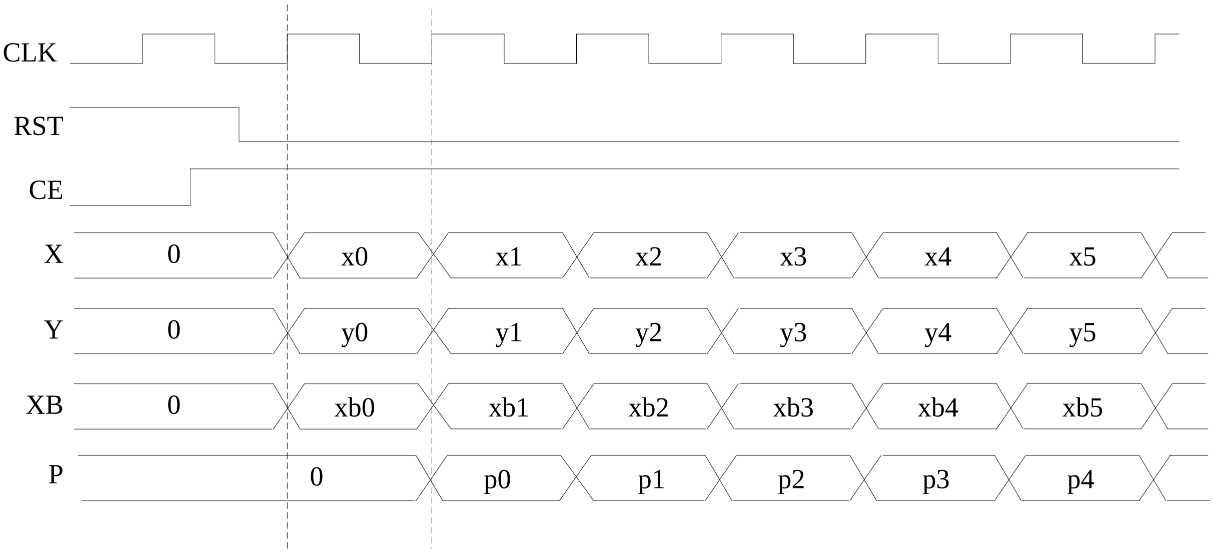


Figure 2-7 Typical Timing Diagram for Multiplication Mode

2.3.2 General Multiply-Add Mode

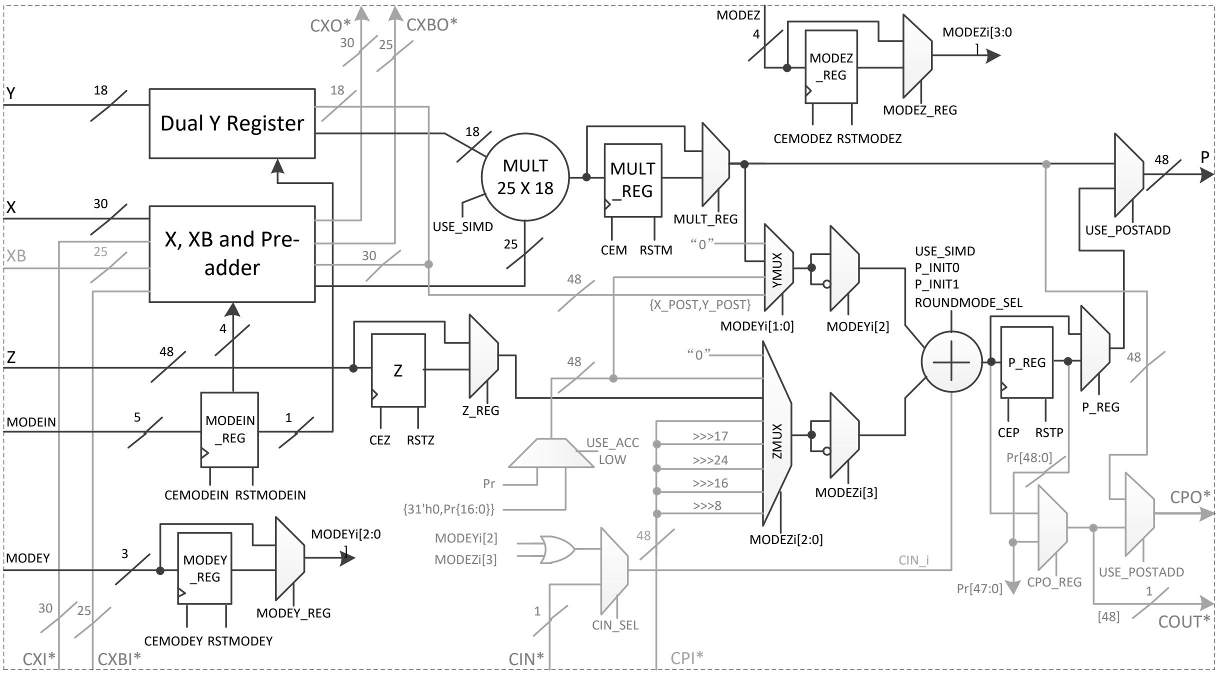


Figure 2-8 Application Diagram of General Multiply-Add Mode

As shown in Figure 2-8, when APM is configured in general multiply-add mode, its implementation expression is:

P=X ×Y ±Z, -X ×Y+Z or -X ×Y-Z-1

The main features of the general multiply-add mode include:

- Each APM can perform two 12 ×9 ±24 operations (USE_SIMD=1) or one 25 ×18 ±48 operation (USE_SIMD=0).
- Signed numbers are supported, and addition and subtraction can be dynamically/statistically controlled.
- Optional input/output registers

Instantiate using GTP_APM_E2 unit. Figure 2-9 shows the typical timing for general multiply-add mode, where P=X ×Y+Z. After the one-stage register is enabled, the P-port calculation result corresponding to the X-port and Y-port input data is output on the next clock rising edge.

When MODEY[2]==1, the output of the multiplier is inverted at Ymux, and then plus 1 at the postadder, which is equivalent to negative the multiplier output.

When MODEZ[3]==1, the output of Zmux is inverted, and then plus 1 at the postadder, which is equivalent to negative the Zmux output.

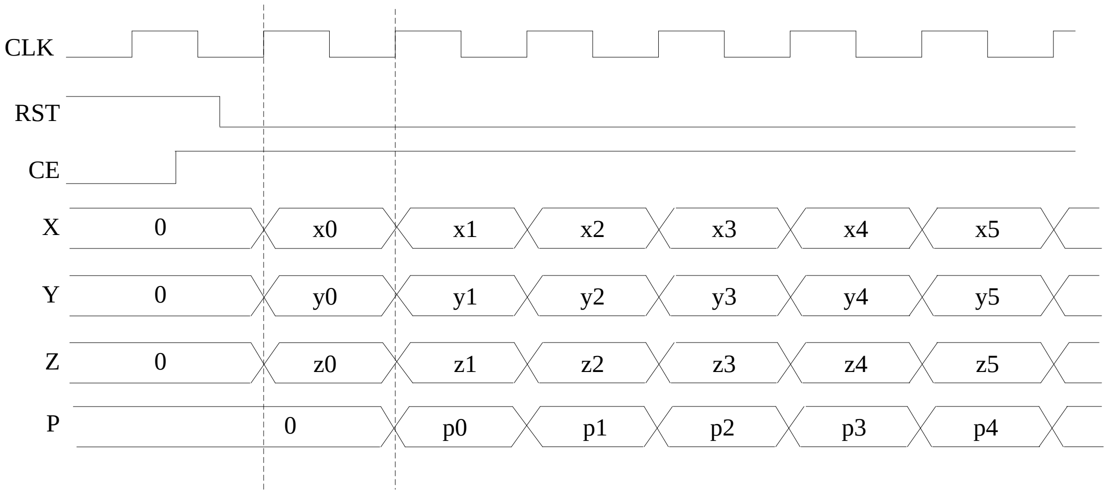


Figure 2-9 Typical Timing for General Multiply-Add Mode

2.3.3 Multiply-Accumulate Mode

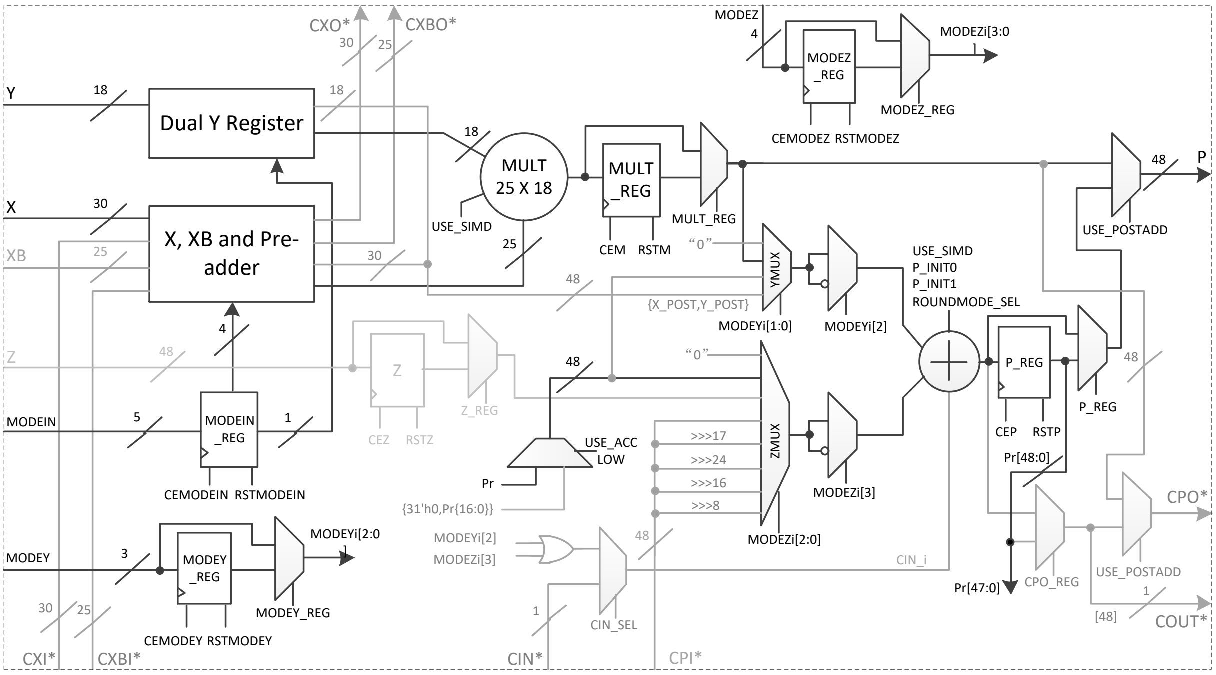


Figure 2-10 Application Diagram of Multiply-Accumulate Mode

As shown in Figure 2-10,when APM is configured in multiply-accumulate mode, its implementation expression is:

P= P ± X ×Y or P=- P ± X ×Y or P= - P - X ×Y-1

The main features of the multiply-accumulate mode include:

- A single APM can perform one 25*18 multiply-accumulate operation (P is 48bit) or two 12*9 multiply-accumulate operations (P is two 24bits)

- Signed numbers, and dynamic/static post-add/subtract control supported
- Optional input/output registers
- P value can be preset (via P_INIT1)
- RELOAD function supported

After enabling the Preadd Unit in APM, APM can be configured into a pre-addition multiply-accumulate mode, with its implementation expression as follows:

P=P ±Y ×(X ±XB) or P=-P+Y ×(X ±XB) or P=-P-Y ×(X ±XB)-1

The pre-addition multiply-accumulate mode has two levels of internal pipeline registers to choose from, and the application diagram is shown in Figure 2-11:

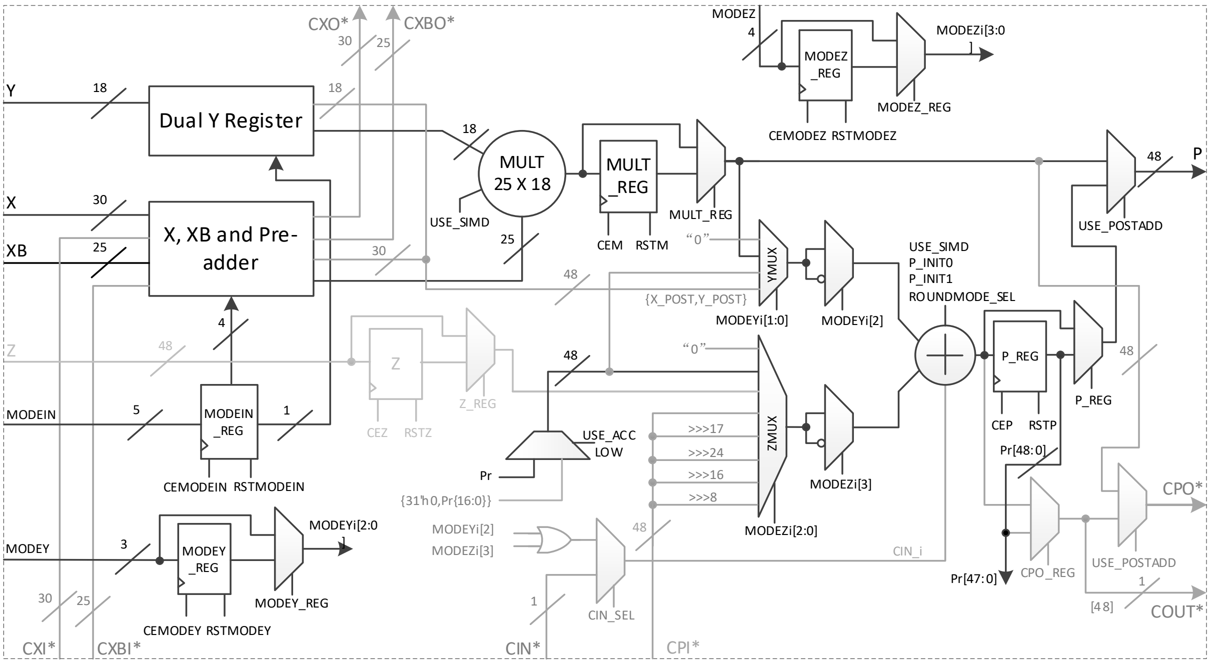


Figure 2-11 Application Diagram of Pre-addition Multiply-Accumulate Mode

Figure 2-12 shows the typical timing for multiply-accumulate mode, where X=1, Y=1, P=P+X*Y, and P is output only when CEP is active. After P_REG is enabled, when there is only the one-stage register, the P-port calculation result corresponding to the X-port and Y-port input data will be output on the next rising edge of the clock.

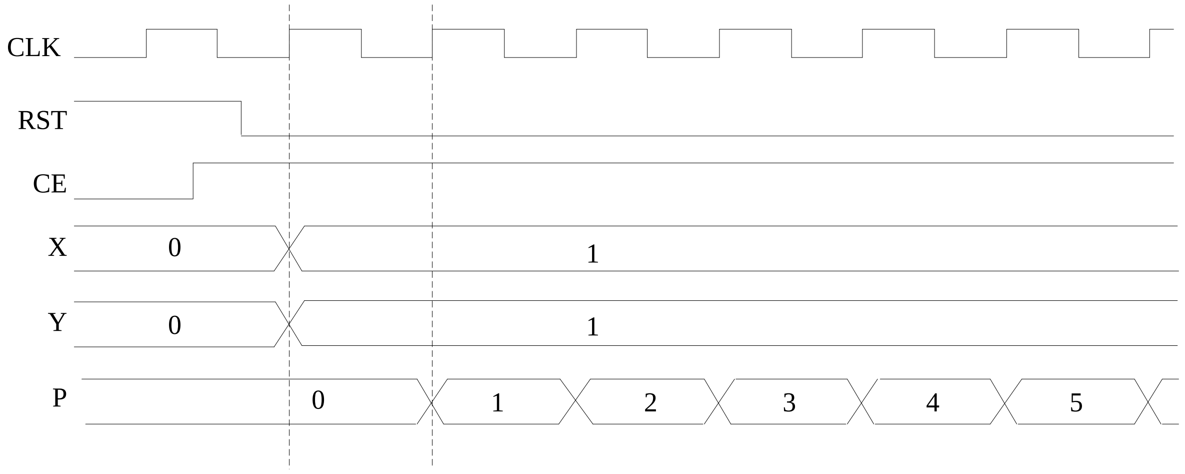


Figure 2-12 Typical Timing of Multiply-Accumulate Mode

2.3.4 FIR Application

FIR operations can also be achieved through APM cascading, and a typical FIR filter can be described by the following expression:

$$y_n = \sum_{i=0}^{N-1} x_{n-i} h_i = x_n h_0 + x_{n-1} h_1 + \ldots + x_{n-N+1} h_{N-1}$$

x is the input data stream, y is the output data stream, and h is the coefficient. Figure 2-13 shows a 8-stage Symmetric systolic FIR operation by cascading 4 APMs.

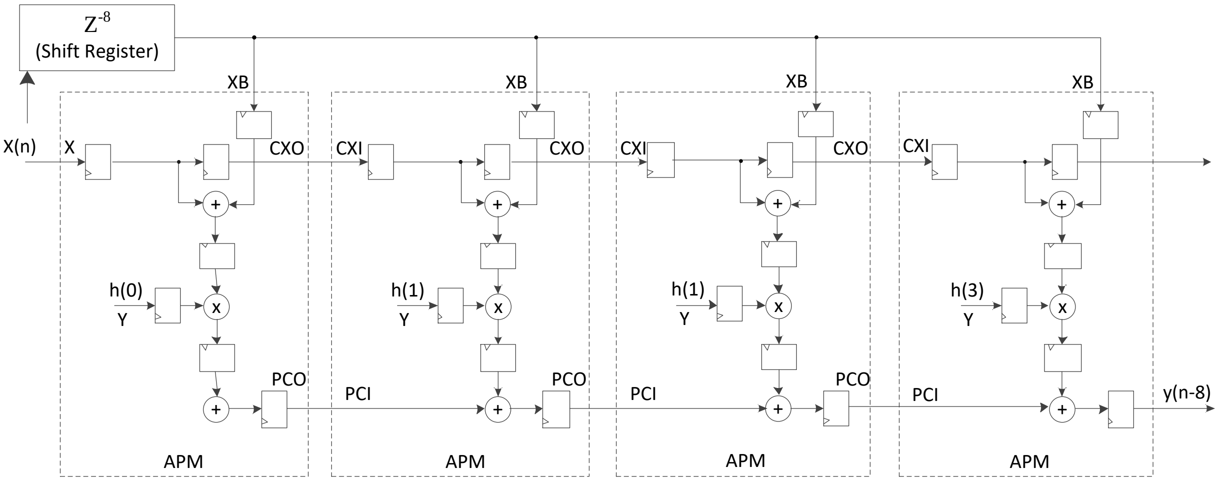


Figure 2-13 Symmetric Systolic FIR Functional Diagram

By configuring GTP_APM_E2 units, various forms of FIR computations can be implemented, with specific implementation types and parameter configuration recommendations shown in Table 2-5 (CPO_REG = 1 in all FIR configurations).

Table 2-5 Typical FIR Filter Implementation

Filter Type	First-Stage APM	Intermediate-Stage APM	Last-Stage APM

Filter Type	First-Stage APM	Intermediate-Stage APM	Last-Stage APM
Common setting:	CPO_REG=1		

Systolic FIR	CXO_REG=2
	X_REG=1
	X_SEL=0
		CXO_REG=2
		X_REG=1
		X_SEL=1
			X_REG=1
			X_SEL=1

2-Channel Systolic FIR	CXO_REG=3
	X_REG=1
	X_SEL=0
		CXO_REG=3
		X_REG=1
		X_SEL=1
			X_REG=1
			X_SEL=1

Transposed FIR	CXO_REG=1
	X_REG=1
	X_SEL=0
		CXO_REG=0
		X_REG=0
		X_SEL=1
			X_REG=0
			X_SEL=1

Symmetrical FIR
(even tap)
	USE_PREADD=1
	CXO_REG=2
	X_REG=1
	X_SEL=0
	XB_SEL=3
	XB_REG=0
		USE_PREADD=1
		CXO_REG=2
		X_REG=1
		X_SEL=1
		XB_SEL=3
		XB_REG=0
			USE_PREADD=1
			CXO_REG=2
			X_REG=1
			X_SEL=1
			XB_SEL=2/3

Symmetrical FIR
(odd tap)
	USE_PREADD=1
	CXO_REG=2
	X_REG=1
	X_SEL=0
	XB_SEL=3
	XB_REG=0
		USE_PREADD=1
		CXO_REG=2
		X_REG=1
		X_SEL=1
		XB_SEL=3
		XB_REG=0
			USE_PREADD=0
			CXO_REG=1
			X_REG=1
			X_SEL=1
			XB_SEL=2/3

2-Channel Symmetrical FIR
(even tap)
	USE_PREADD=1
	CXO_REG=3
	X_REG=1
	X_SEL=0
	XB_SEL=3
	XB_REG=1
		USE_PREADD=1
		CXO_REG=3
		X_REG=1
		X_SEL=1
		XB_SEL=3
		XB_REG=1
			USE_PREADD=1
			CXO_REG=3
			X_REG=1
			X_SEL=1
			XB_SEL=2/3

2-Channel Symmetrical FIR
(odd tap)
	USE_PREADD=1
	CXO_REG=3
	X_REG=1
	X_SEL=0
	XB_SEL=3
	XB_REG=1
		USE_PREADD=1
		CXO_REG=3
		X_REG=1
		X_SEL=1
		XB_SEL=3
		XB_REG=1
			USE_PREADD=0
			CXO_REG=1
			X_REG=1
			X_SEL=1
			XB_SEL=2/3

2.4 Function Descriptions

2.4.1 Rounding Function

The postadd unit of Logos2 APM supports Direct rounding and Round-toward-nearest for multiplication and multiply-add operations. The Rounding function means regarding the low N bits output by the postadder as decimal places, and then discarding the decimal places to obtain the nearest integer value. In practical use, the Postadder calculates a 48-bit result according to different Rounding modes. Users need to manually extract P[47:N] for rounding, where N∈[1,45], and N=0 (no rounding). Table 2-6 lists the detailed functions.

Table 2-6 Detailed Rounding Function List

Rounding Function		Function Example	Multiply Function	Multiply-Add-Subtract Function	Implementation Method (N is the float number position)
Direct rounding	Round-ceiling	3.1,3.5,3.9>4 -3.1,-3.5,-3.9>-3	Supported	Supported	+2^N-1, and then rounding
	Round-floor	3.1,3.5,3.9>3 -3.1,-3.5,-3.9>-4	Supported	Supported	Direct rounding
	Round-toward zero	3.1,3.5,3.9>3 -3.1,-3.5,-3.9>-3	Supported	Supported	+2^N-1 for negative numbers, direct rounding for positive numbers
	Round-away from zero	3.1,3.5,3.9>4 -3.1,-3.5,-3.9>-4	Supported	Supported	+2^N-1 for positive numbers, direct rounding for negative numbers
Round-toward-nearest	Symmetric Round-half-up	3.1>3; 3.5,3.9>4 -3.1>-3; -3.5,-3.9>-4	Supported	Supported	+2^(N-1) for positive numbers, +2^(N-1)-1 for negative numbers, and then rounding
	Asymmetric Round-half-up	3.1>3; 3.5,3.9>4 -3.1,-3.5>-3; -3.9>-4	Supported	Supported	+2^(N-1), and then rounding
	Symmetric Round-half-down	3.1,3.5>3; 3.9>4 -3.1,-3.5>-3; -3.9>-4	Supported	Supported	+2^(N-1)-1 for positive numbers, +2^(N-1) for negative numbers, and then rounding
	Asymmetric Round-half-down	3.1,3.5>3; 3.9>4 -3.1>-3; -3.5,-3.9>-4	Supported	Supported	+2^(N-1)-1, and then rounding

Table 2-7 shows the parameter configurations for Logos2 APM to implement different rounding functions:

Table 2-7 List of Rounding Function Configurations

Rounding Function		ROUNDMODE_SEL	P_INIT0	P_INIT1
Direct rounding	Round-ceiling	0	2^N-1	0
	Round-floor	0	0	0
	Round-toward zero	1	0	2^N-1
	Round-away from zero	1	2^N-1	0

Rounding Function		ROUNDMODE_SEL	P_INIT0	P_INIT1
Round-toward-nearest	Symmetric Round-half-up	1	2^(N-1)	2^(N-1)-1
	Asymmetric Round-half-up	0	2^(N-1)	0
	Symmetric Round-half-down	1	2^(N-1)-1	2^(N-1)
	Asymmetric Round-half-down	0	2^(N-1)-1	0

2.4.2 Time Division Multiplexing

Logos2 APM supports time-division multiplexing. Table 2-8 shows the time-division multiplexing operations and their parameter configurations.

Table 2-8 Time-Division Multiplexing Details

Configuration Function	Register			Dynamic Control Configuration			
	Input_reg	Pipe_reg	Output_reg	MODEIN [1]	MODEIN [2]	MODEY [1:0]	MODEZ [2:0]
MULT25	1	1	1	1	0	2’b01	3’b000
PREADD_MULT25	1	1	1	1	1	2’b01	3’b000
MULTADD25	1	1	1	1	0	2’b01	3’b010
PRE_MULTADD25	1	1	1	1	1	2’b01	3’b010
Wide bit ADD/SUB	2	0	1	0	0	2’b11	3’b010

1. When implementing time-division multiplexing, APM-supported functions can be configured into three-stage registers.
2. Input_reg includes input registers for all data paths and control signals. When X and Y paths are time-division multiplexed, two input register delays are needed, i.e., X/Y_REG[1:0] should be set to 2’b11. In the time-division multiplexing function, Pipe_reg is MULT_REG and Output_reg is P_REG.
3. When the Ymux data reaching POSTADD has 2 CLK delays, MODEY[1:0] has only 1 input register in the APM. It needs one staggered CLK delay, or compensate for one CLK delay outside the APM through MODEY[1:0].
4. The Z path and MODEZ both have 1 CLK delay. Control them with one staggered delay compared with data on the X and Y paths, or compensate for one CLK delay outside the APM.
5. MODEIN [0], MODEIN [4] are fixed at 1, POSTADD is always enabled, and if PREADD function is used in time-division multiplexing, PREADD is always enabled, PREADD_REG is not enabled (deactivated if PREADD is unused).

6. PREADD_MULT25 and PRE_MULTADD25 do not have PREADD_REG, making it impossible to achieve the maximum frequency.

Chapter 3 Appendix

3.1 Cascading Capabilities

Cascading APMs can achieve higher bit-width operations. Table 3-1 describes the functions achieved by APM cascading. Table 3-1 show examples for reference only. Cascading input ports can only be connected to the cascading output ports of the previous stage, and cascading output ports can only be connected to the cascading input ports of the next stage. Leave them floating when unused.

Table 3-1 Descriptions of Functions Achieved by APM Cascading

APM Cascading	Capabilities
Cascading 2 APMs	Two (12×9+12×9)
	Two [(12+12)×9+(12+12)×9]
	Output 65-bit MULTACC25
	Output 96-bit MULTACC25
	Output 96-bit ADD/ACC (addition)
	25×18+25×18
	(25+25)×18+(25+25)×18
	Output 65-bit PREADD_MULTACC25
	Output 96-bit PREADD_MULTACC25
	49×18 (only use CPI 24-bit right shift function)
	42×18 (only use CPI 17-bit right shift function)
	25×35 (only use CPI 17-bit right shift function)
	25×34 (only use CPI 16-bit right shift function)
	(25+25)×35
Cascading 3 APMs	Two 24-bit [MAC+(12×9+12×9)]
	Two 24-bit [MAC+((12+12)×9+(12+12)×9)]
	Output 48-bit MAC+(25×18+25×18)
	Output 48-bit MAC+((25+25)×18+(25+25)×18)
	59×18 (only use CPI 17-bit right shift function)
	25×52 (only use CPI 17-bit right shift function)
	25×50 (only use CPI 16-bit right shift function)
	73×18 (only use CPI 24-bit right shift function)
	MAC+42×18
	MAC+25×35
	MAC+(25+25)×35
Cascading 4 APMs	Two [(12×9+12×9)+(12×9+12×9)]
	Two [((12+12)×9+(12+12)×9)+((12+12)×9+(12+12)×9)]

	Output 65-bit MAC+(25×18+25×18)
	Output 65-bit MAC+((25+25)×18+(25+25)×18)
	(25×18+25×18)+(25×18+25×18)
	((25+25)×18+(25+25)×18)+((25+25)×18+(25+25)×18)
	35×25+35×25 (only use CPI 17-bit right shift function)
	42×18+42×18 (only use CPI 17-bit right shift function)
	25×34+25×34 (only use CPI 16-bit right shift function)
	49×18+49×18 (only use CPI 24-bit right shift function)
	42×35 (only use CPI 17-bit right shift function)
	25×69 (only use CPI 17-bit right shift function)
	76×18 (only use CPI 17-bit right shift function)
	49×34 (use CPI 16- and 8-bit right shift functions)
	36×(25+25)+36×(25+25)

3.2 Design Recommendations

- For high-speed filtering pipeline applications, it is recommended to use APM cascading, including adder cascading.
- The multiplier is dynamically controlled by signals such as MODEY, MODEZ, and MODEIN, offering high flexibility. Using registers and dynamic configuration modes in the design can make fuller use of APM performance compared to multiplier combinations.
- To achieve the highest performance of APM, use all pipeline registers. If not all registers can be used due to requirements for delay, please enable MREG as possible.
- Using internal cascade ports instead of fabric routing connections can save power consumption.
- Perform sign bit extension when performing operations with smaller bit widths.
- When multiple APMs are cascaded, the pipeline stages of different signal channels should match.

3.3 Instantiate GTP for multiplication mode

The instantiation template of GTP for pre-addition multiplier is as follows:

- Enable the pre-add function with USE_PREADD=1; not enable POSTADD
- Not use X port, Y port input registers; use MULT_REG
- When MODEIN=5'b00110, X_MULT=X2+XB, pre-addition is addition, select Y2. As X/Y port registers are not enabled, X1/X2 are equivalent to X, and Y1/Y2 are equivalent to Y.
- When MODEY=3'b000, MODEY controls the POSTADD function, with POSTADD not enabled.

- When MODEZ=4'b0000, Zmux is selected as 0, with Zmux output not inverted and POSTADD not enabled.
- When ROUNDMODE_SEL=0, P_INIT0 =48'd0, and P_INIT1=48'd0, the rounding function is set to Round-floor.
- Output result is P=(X+XB)×Y.

```
GTP_APM_E2 #(
        .GRS_EN ("TRUE"),
        .USE_POSTADD (0),
        .USE_PREADD(1), //Enable pre-add function
        .PREADD_REG (0),

        .X_REG (0),
        .CXO_REG (0),
        .XB_REG (0),
        .Y_REG (0),
        .Z_REG (0),
        .MULT_REG (0),
        .P_REG (0),
        .MODEY_REG (0),
        .MODEZ_REG (0),
        .MODEIN_REG (0),

        .X_SEL (0),
        .XB_SEL (0),
        .ASYNC_RST (0),
        .USE_SIMD (0),
        .P_INIT0 (48'd0),
        .P_INIT1 (48'd0),
        .ROUNDMODE_SEL (0),

        .CPO_REG (0),
        .USE_ACCLOW (0),
        .CIN_SEL (0)
)
```

```
u0_GTP_APM_E2
(
        .P(P),
        .CPO(),
        .COUT(),
        .CXO(),
        .CXBO(),

        .X(X),
        .CXI(),
        .CXBI(),
        .Y(Y),
        .Z(),
        .CPI(),
        .CIN(),
        .XB (XB),
        .MODEIN(5'b00110),    //X_MULT=X2+XB, Y_MULT=Y2
        .MODEY(3'b000),        //YMUX output is 0
        .MODEZ(4'b0000),       //ZMUX output is 0

        .CLK(Clk),
        .RSTX(1'b0),
        .RSTXB(1'b0),
        .RSTY(1'b0),
        .RSTZ(1'b0),
        .RSTM(1'b0),
        .RSTP(1'b0),
        .RSTPRE(1'b0),
        .RSTMODEIN(1'b0),
        .RSTMODEY(1'b0),
        .RSTMODEZ(1'b0),

        .CEX1(1'b1),
        .CEX2(1'b1),
        .CEX3(1'b1),
        .CEXB(1'b1),
        .CEY1(1'b1),
        .CEY2(1'b1),
        .CEZ(1'b1),
        .CEM(1'b1),
        .CEP(1'b1),
        .CEPRE(1'b1),
        .CEMODEIN(1'b1),
        .CEMODEY(1'b1),
        .CEMODEZ(1'b1)
    );
```

3.4 Instantiate GTP for multiply-add mode

The instantiation template of GTP for general multiply-add is as follows:

- Enable the accumulation function with USE_POSTADD=1; not enable the PREADD function
- Not use X port, Y port input registers; use P_REG
- When MODEIN=5'b00010, X_MULT=X2, pre-addition applies addition (PREADD function not enabled). Select Y2. As X/Y port registers are not enabled, X1/X2 are equivalent to X, and Y1/Y2 are equivalent to Y.
- When MODEY=3'b001, MODEY controls the POSTADD function, select the multiplier output, with Ymux output not inverted.
- When MODEZ=4'b0010, Zmux is selected as Z input, with Zmux output not inverted.
- When ROUNDMODE_SEL=0, P_INIT0 =48'd0, and P_INIT1=48'd0, the rounding function is set to Round-floor.
- Output result is P=X*Y+Z

```
    GTP_APM_E2 #(
        .GRS_EN ("TRUE"),
        .USE_POSTADD(1),    //Enable postadd function
        .USE_PREADD (0),
        .PREADD_REG (0),

        .X_REG (0),
        .CXO_REG (0),
        .XB_REG (0),
        .Y_REG (0),
        .Z_REG (0),
        .MULT_REG (0),
        .P_REG (0),
        .MODEY_REG (0),
        .MODEZ_REG (0),
        .MODEIN_REG (0),

        .X_SEL (1'b0),
        .XB_SEL (0),
        .ASYNC_RST (0),
        .USE_SIMD (0),
        .P_INIT0 (48'd0),
        .P_INIT1 (48'd0),
        .ROUNDMODE_SEL (0),

        .CPO_REG (0),
        .USE_ACCLOW (0),
        .CIN_SEL (0)
    )
    u0_GTP_APM_E2
    (
        .P(P),
        .CPO(),
        .COUT(),
        .CXO(),
        .CXBO(),

        .X(X),
        .CXI(),
        .CXBI(),
        .Y(Y),
        .Z(),
        .CPI(),
        .CIN(),
        .XB (XB),
        .MODEIN(5'b00010),    //X_MULT=X2
        .MODEY(3'b001),         //YMUX outputs multiplier result
        .MODEZ(4'b0010),       //ZMUX output is Z

        .CLK(Clk),
        .RSTX(1'b0),
        .RSTXB(1'b0),
        .RSTY(1'b0),
        .RSTZ(1'b0),
        .RSTM(1'b0),
        .RSTP(1'b0),
        .RSTPRE(1'b0),
        .RSTMODEIN(1'b0),
        .RSTMODEY(1'b0),
        .RSTMODEZ(1'b0),

        .CEX1(1'b1),
        .CEX2(1'b1),
        .CEX3(1'b1),
        .CEXB(1'b1),
        .CEY1(1'b1),
        .CEY2(1'b1),
        .CEZ(1'b1),
        .CEM(1'b1),
        .CEP(1'b1),
        .CEPRE(1'b1),
        .CEMODEIN(1'b1),
        .CEMODEY(1'b1),
        .CEMODEZ(1'b1)
    );
```

3.5 Instantiate GTP for multiply-accumulate mode

The instantiation template of GTP for multiply-accumulate is as follows:

- Enable the accumulation function with USE_POSTADD=1; not enable the PREADD function
- Not enable X port, Y port input registers; use P_REG
- When MODEIN=5'b00010, X_MULT=X2, pre-addition applies addition (PREADD function not enabled). Select Y2. As X/Y port registers are not enabled, X1/X2 are equivalent to X, and Y1/Y2 are equivalent to Y.
- When MODEY=3'b001, MODEY controls the POSTADD function, select the multiplier output, with Ymux output not inverted (the result turns into P=P-X×Y if inverted).
- When MODEZ=4'b0001, Zmux is selected as the postadder feedback, with Zmux output not inverted (the result turns into P=-P+X×Y if inverted).
- When ROUNDMODE_SEL=0, P_INIT0 =48'd0, and P_INIT1=48'd0, the rounding function is set to Round-floor.
- In multiply-accumulate mode, it is important to first set MODEZ to 4'b0000, initializing the internal APM postadd value to zero, then set the MODEZ value to 4'b0001 to perform the multiply-accumulate calculation.
- Output result is P=P+X*Y

```
GTP_APM_E2 #(
        .GRS_EN ("TRUE"),
        .USE_POSTADD(1),    //Enable postadd function
        .USE_PREADD (0),
        .PREADD_REG (0),


        .X_REG (0),
        .CXO_REG (0),
        .XB_REG (0),
        .Y_REG (0),
        .Z_REG (0),
        .MULT_REG (0),
        .P_REG (0),
        .MODEY_REG (0),
        .MODEZ_REG (0),
        .MODEIN_REG (0),
        .X_SEL (0),
        .XB_SEL (0),
        .ASYNC_RST (0),
        .USE_SIMD (0),
        .P_INIT0 (48'd0),
        .P_INIT1 (48'd0),
        .ROUNDMODE_SEL (0),

        .CPO_REG (1),
        .USE_ACCLOW (0),
        .CIN_SEL (0)
    )
    u0_GTP_APM_E2
    (
        .P(P),
        .CPO(),
        .COUT(),
        .CXO(),
        .CXBO(),

        .X(X),
        .CXI(),
        .CXBI(),
        .Y(Y),
        .Z(),
        .CPI(),
        .CIN(),
        .XB (XB),
        .MODEIN(5'b00010),    //X_MULT=X2
        .MODEY(3'b001),       //YMUX outputs multiplier result
        .MODEZ(4'b0001),      //ZMUX output is postadder feedback

        .CLK(Clk),
        .RSTX(1'b0),
        .RSTXB(1'b0),
        .RSTY(1'b0),
        .RSTZ(1'b0),
        .RSTM(1'b0),
        .RSTP(1'b0),
        .RSTPRE(1'b0),
        .RSTMODEIN(1'b0),
        .RSTMODEY(1'b0),
        .RSTMODEZ(1'b0),

        .CEX1(1'b1),
        .CEX2(1'b1),
        .CEX3(1'b1),
        .CEXB(1'b1),
        .CEY1(1'b1),
        .CEY2(1'b1),
        .CEZ(1'b1),
        .CEM(1'b1),
        .CEP(1'b1),
        .CEPRE(1'b1),
        .CEMODEIN(1'b1),
        .CEMODEY(1'b1),
        .CEMODEZ(1'b1)
    );
```

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its family's products at any time without prior notice.