

Logos2 系列 FPGA 算术处理模块(APM) 用户指南

(UG040003, V1.3)

(2023.7.17)

版权所有 侵权必究

文档版本修订记录

版本号	发布日期	修订记录
V0.1	2019/12/03	初始 Alpha 版本
V1.0	2020/9/27	<ol style="list-style-type: none"> 更新了表 2-1 GTP_APM_E2 端口说明列表、表 2-5 典型 FIR 滤波器实现、表 2-8 时分复用详细表格的部分配置位设置 对 3 附录部分示例增加说明
V1.1	2021/3/29	<ol style="list-style-type: none"> 完善 3.1 级联可实现功能，增加级联端口说明 更新 2.1 APM 结构说明中图 2-1、图 2-2、图 2-3 功能框图，更新 2.3 APM 工作模式中图 2-5、图 2-6、图 2-8、图 2-10、图 2-11、图 2-13 的应用示意图 2.1 APM 结构说明中补充功能说明，增加各结构单元功能特性描述、资源描述 修正补充 2.2.2 端口说明、2.2.3 参数说明中端口及参数描述 补充 2.2.4 模式说明中描述 增加 3.2 设计建议说明
V1.2	2021/10/24	<ol style="list-style-type: none"> 表 2-3 MODE 端口功能说明中取负功能修正为取反 更新表 2-7 Rounding 各功能配置列表中参数名称及补充 Rounding 功能说明 表 2-1 GTP_APM_E2 端口说明列表中增加链接，修改级联端口描述 2.2.4 模式说明中删除补偿延迟的描述 补充 2.3 APM 工作模式中可实现表达式，修正时序图描述 2.3.2 通用乘加模式增加 Ymux 与 Zmux 取反的说明 修正表 2-5 典型 FIR 滤波器实现中参数名称 完善 2.4.2 时分复用中表述 修正 3 附录例化说明中的取反说明，删除端口注释，补充参数位宽 修正表 3-1 APM 级联可实现功能描述中 CPI 端口名称 2.1 APM 结构说明中 APM 资源描述更新 2.2 GTP 说明增加默认值描述 更新图 2-2 X、XB 及 Pre-adder 功能框图 更新表 2-1 GTP_APM_E2 端口说明列表中寄存器名称 更新表 2-2 GTP_APM_E2 参数说明列表中 XB_SEL、P_INIT1、USE_ACCLW、CIN_SEL 参数说明 1 总体介绍增加 IP 用户指南描述 更新表 2-2 GTP_APM_E2 参数说明列表中 P_INIT0、P_INIT1、ROUNDMODE_SEL 参数说明 更新图 2-12 乘累加模式的典型时序
V1.3	2023/7/17	<ol style="list-style-type: none"> 更新 1 总体介绍中 IP 用户指南说明 更新图 2-1、图 2-5、图 2-6、图 2-8、图 2-10、图 2-11、图 2-13，补充 MUX 及 CIN_i 信号名称 删除 2.1 APM 结构说明中 APM TILE 说明 更新文中“PREADD”参数名为“USE_PREADD” 更新表 2-2 GTP_APM_E2 参数说明列表中 CXO_REG 及 CIN_SEL 参数说明 更新表 2-3 MODE 端口功能说明中取负说明，新增注释说明 更新 2.3.3 乘累加模式中特性说明 更新表 2-5 典型 FIR 滤波器实现中 XB_SEL 参数值 更新 2.4.1 Rounding 功能说明 更新 3 附录中例化模板注释说明

术语与缩略语

Terms and Abbreviations 术语与缩略语	Full Spelling 英文全拼	Chinese Explanation 中文解释
APM	Arithmetic Process Module	算术处理单元
MAC	Multiply Accumulate	乘累加
SIMD	Single-instruction-multiply-data	单指令多数据

目 录

文档版本修订记录	1
术语与缩略语	2
目 录	3
表目录	5
图目录	6
1 总体介绍	7
2 功能描述	8
2.1 APM 结构说明	8
2.1.1 I/O 单元	10
2.1.2 预加单元	10
2.1.3 乘法单元	10
2.1.4 累加单元	10
2.2 GTP 说明	11
2.2.1 GTP 结构框图	11
2.2.2 端口说明	12
2.2.3 参数说明	13
2.2.4 模式说明	14
2.3 APM 工作模式	16
2.3.1 乘法模式	16
2.3.2 通用乘加模式	18
2.3.3 乘累加模式	19
2.3.4 FIR 应用	21
2.4 具体功能描述	22
2.4.1 Rounding 功能	22
2.4.2 时分复用	23
3 附录	24
3.1 级联可实现功能	24
3.2 设计建议	25

3.3 使用 GTP 例化为乘法模式.....	25
3.4 使用 GTP 例化为乘加模式.....	27
3.5 使用 GTP 例化为乘累加模式.....	29
免责声明.....	32

表目录

表 2-1 GTP_APM_E2 端口说明列表	12
表 2-2 GTP_APM_E2 参数说明列表	13
表 2-3 MODE 端口功能说明	15
表 2-4 MODEIN 功能列表	15
表 2-5 典型 FIR 滤波器实现	21
表 2-6 Rounding 详细功能列表	22
表 2-7 Rounding 各功能配置列表	22
表 2-8 时分复用详细表格	23
表 3-1 APM 级联可实现功能描述	24

图目录

图 2-1 APM 整体功能框图.....	8
图 2-2 X, XB 及 Pre-adder 功能框图	9
图 2-3 Y 端口功能框图	9
图 2-4 GTP_APM_E2 结构图	11
图 2-5 乘法模式应用示意图	16
图 2-6 带预加的乘法模式应用示意图	17
图 2-7 乘法模式的典型时序图	17
图 2-8 通用乘加模式应用示意图	18
图 2-9 通用乘加模式的典型时序	19
图 2-10 乘累加模式应用示意图	19
图 2-11 带预加的乘累加应用示意图	20
图 2-12 乘累加模式的典型时序	20
图 2-13 Symmetric systolic FIR 功能示意图	21

1 总体介绍

Logos2 系列产品提供了 APM（Arithmetic Process Module，算术处理单元）模块，为 Logos2 系列产品提供了高效的数字信号处理能力。

APM 在 Logos2 系列产品中按列分布，其主要特性有：

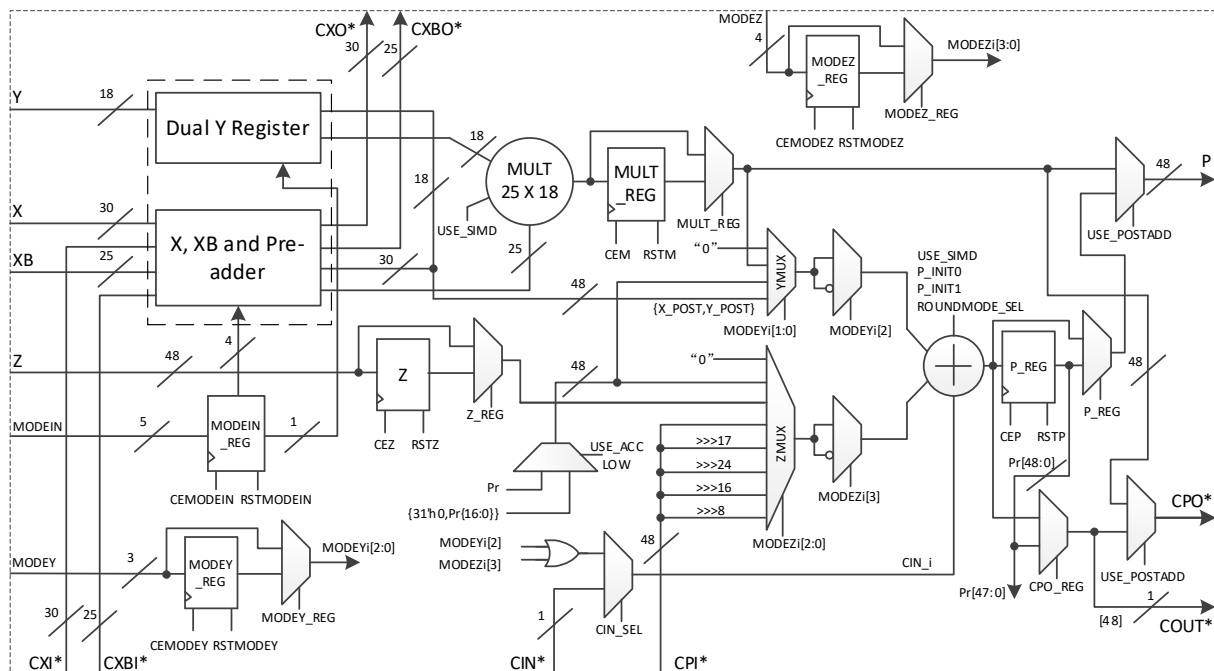
- 有符号数乘法器 25×18 ，无符号乘通过高位赋 0 实现
- 所有的计算及输出结果均为有符号数，已包含符号位
- 支持 1 个 48bits 加/减/累加运算或 2 个 24bits 运算
- Pre-add 为 25bits
- 独立可选的 CE 和 RST
- 支持输入级联
- 支持输出级联
- 控制/数据信号流水线
- 支持动态模式切换
- 支持 Rounding（取整）功能

APM的实现可通过软件Pango Design Suite（简称为PDS）来完成。为方便用户使用，Pango Design Suite内嵌的IP Compiler工具提供相关功能IP的生成。PDS软件中集成IP用户指南，具体IP的使用可在选择APM的使用模式后查阅相应的IP用户指南。

2 功能描述

2.1 APM 结构说明

APM 主要由 I/O Unit (I/O 单元)、Preadd Unit (预加单元)、Mult Unit (乘法单元) 和 Postadd Unit (累加单元) 四个功能单元组成。APM 的整体功能框图如图 2-1 所示。



注：*这些信号为内部级联信号

图 2-1 APM 整体功能框图

图 2-1 中虚线框内 X, XB, Pre-adder 与 Y 端口双寄存器输入结构如图 2-2, 图 2-3 所示。

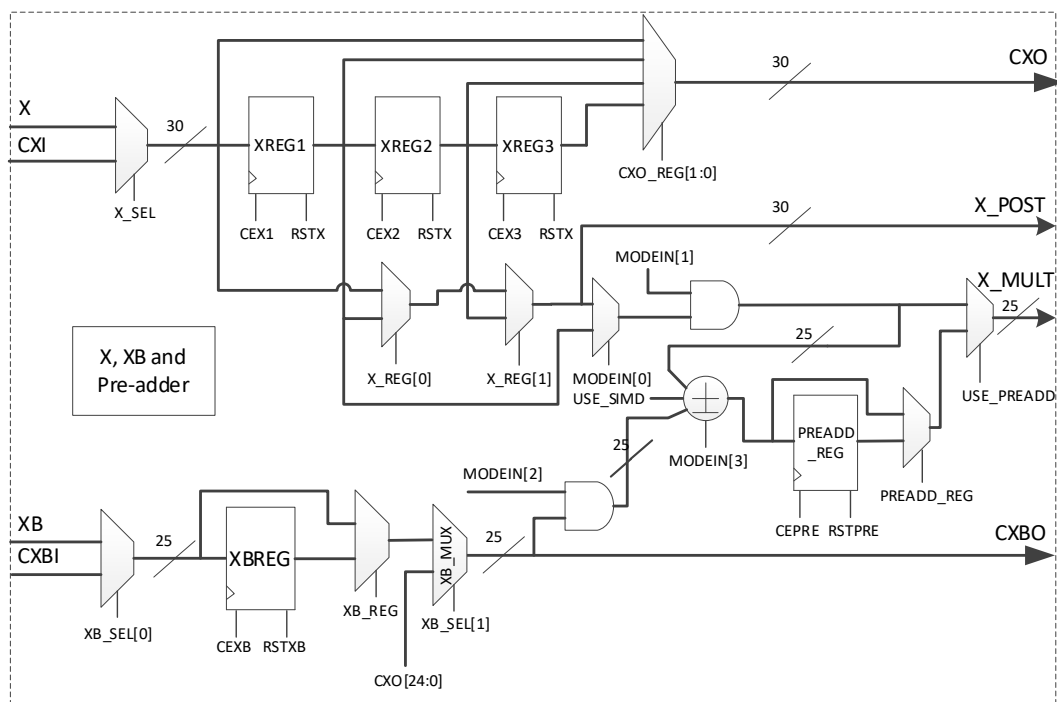


图 2-2 X, XB 及 Pre-adder 功能框图

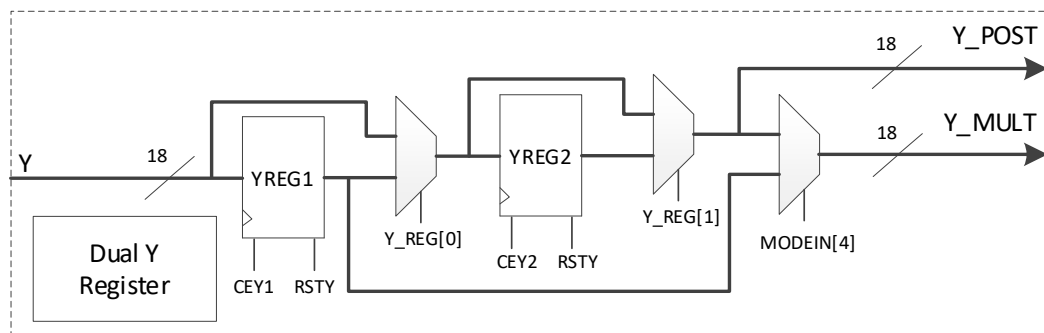


图 2-3 Y 端口功能框图

APM 模块包括一个乘法器、其后与一个加法器连接。APM 满速度工作时需要使用全部流水寄存器。

- APM 支持级联模式，常用的应用有 FFT、FIR、乘法/加减运算、大位宽乘法等。
- APM 还包括同步复位、时钟使能、X 端口双输入寄存器、SIMD 功能、Direct rounding 和 Round-toward-nearest 功能。
- 8/16/17/24 位右移功能可实现更大位宽的乘法。
- 可选的输入、流水、预加、输出、乘法、累加寄存器。
- 可选的控制信号（MODEY、MODEZ、MODEIN）寄存器。

APM 资源情况及性能参见《DS04001_Logos2 系列 FPGA 器件数据手册》APM 部分。

2.1.1 I/O 单元

- 实现数据输入/输出寄存。
- 各个寄存器的 CE 信号（高有效）和 RST 信号（高有效）都有其各自独立的 CE 和 RST 输入来控制。RST 异步/同步性质由共享参数 ASYNC 决定。
- 模式端口按控制功能归属分成 MODEIN[4:0], MODEY[2:0], MODEZ[3:0]。每组模式端口控制由各自的独立的 REG, RST, CE 参数控制。
- APM 级联用于实现 FIR 以及高位宽乘法器如 49*35。
- 实现对常数输入和重复的符号位简化接线。
- X 端口支持 30 位二则运算，端口 XB[24:0]可支持 PREADD_MULTADD 运算，MODEIN 控制端用于支持时分复用的功能。
- X 端口的 30 位数据中低 25 位用于乘法器，整个 30bit 组成了 48bit 的 X:Y 宽位宽数据的高 30 位。
- X、Y 输入的多级寄存器可独立控制。
- 独立的 Z 输入与输入寄存器。
- CIN, COUT 为内部级联信号，支持两个 APM 间的 96 位乘法、加减法。

2.1.2 预加单元

- 硬件实现为 25 ± 25 ，输出 25 位结果；或两个 12 ± 12 ，各输出 12 位结果(当 USE_SIMD=1)。
- 预加功能可被旁路 (USE_PREADD=0)。
- 带有输出寄存器

2.1.3 乘法单元

- 实现一个 25×18 ，结果带符号扩展为 48 位；或两个 12×9 （当 USE_SIMD=1），每个 12×9 结果带符号扩展为 24 位。
- 操作数均为有符号数。

2.1.4 累加单元

- 累加单元支持实现一个 48 位加法；或两个 24 位加法，减法运算 $P=X-Y$ 转换成加法 $P=X+\sim Y+1$ 来运算。
- POSTADD 部分支持 rounding 功能，设置有 ROUNDMODE, PINIT0 和 PINIT1 参数，具体可参考 [2.4.1 Rounding 功能描述](#) 章节
- POSTADD 可被旁路（当 POSTADD=0），此时 APM 输出乘法器结果。

2.2 GTP 说明

GTP_APM_E2 单元集成了 APM 的各种工作模式，用户可通过对接口和参数的配置，实现各种模式的选择和 APM 功能实现。各寄存器默认值以《UG050007_Logos2 系列产品 GTP 用户指南》为准。

2.2.1 GTP 结构框图

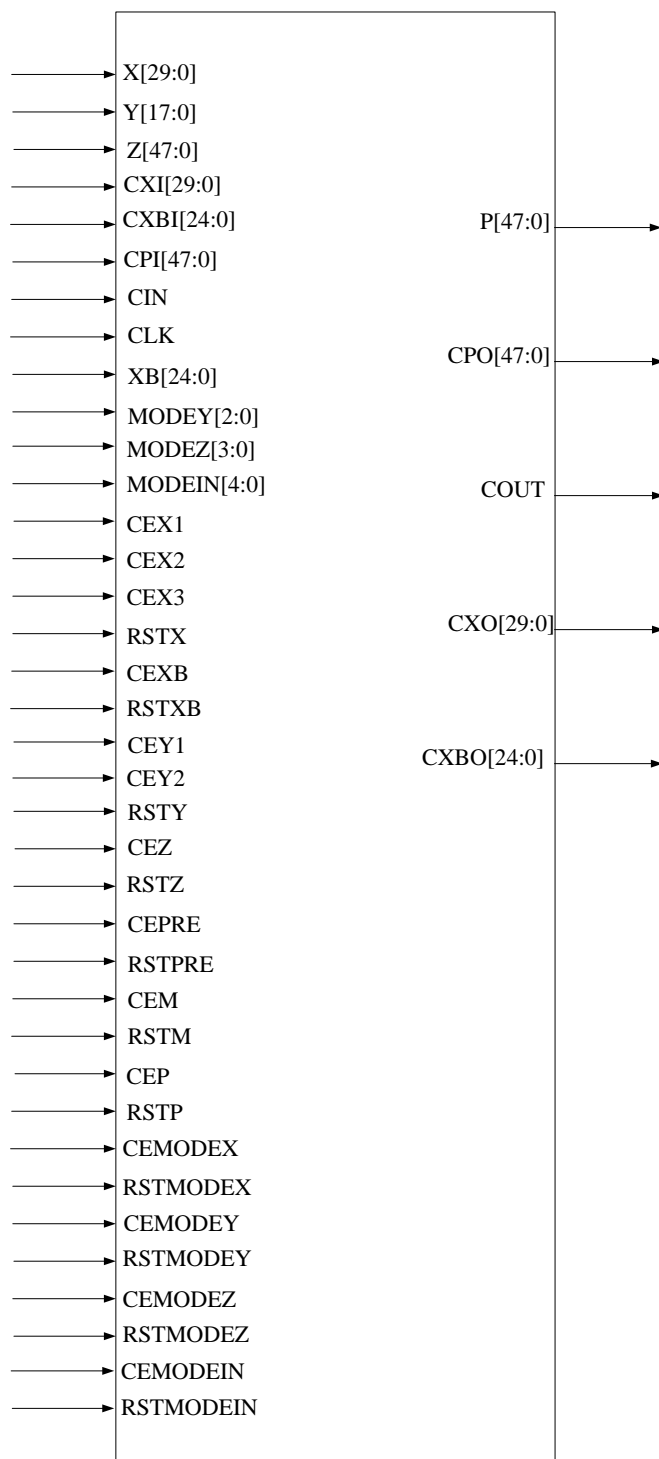


图 2-4 GTP_APM_E2 结构图

2.2.2 端口说明

表 2-1 GTP_APM_E2 端口说明列表

端口名	输入/输出	描述
X[29:0]	输入	并行数据输入，X[24:0]作为乘法器或预加器的 X 输入，X[29:0]作为 X:Y 的高 30 位，可作为累加器输入数据
CXI[29:0]	输入	级联 X 输入，来自其他 APM 模块的 CXO 端口
CXBI[24:0]	输入	级联 XB 输入，来自其他 APM 模块的 CXBO 端口
XB[24:0]	输入	并行数据输入 XB，作为预加器输入，预加器输出为 $X \pm XB$ ，由 MODEIN[3]控制
Y[17:0]	输入	并行数据输入 Y，乘法器的 Y 输入，Y[17:0]作为 X:Y 的低 18 位，可作为累加器输入数据
Z[47:0]	输入	并行数据输入 Z，累加器输入数据
CPI[47:0]	输入	级联 P 输入，来自上一个 APM 模块的 CPO 端口
CIN	输入	级联 CIN 进位输入，来自上一个 APM 模块的 COUT 端口
MODEY[2:0]	输入	APM 动态 Y 端控制操作符，具体功能见 模式说明
MODEZ[3:0]	输入	APM 动态 Z 端控制操作符，具体功能见 模式说明
MODEIN[4:0]	输入	APM 动态输入控制操作符，具体功能见 模式说明
CLK	输入	时钟输入，用于所有内部寄存器
CEX1	输入	XREG1（第一个 X 端口输入寄存器）的时钟使能信号，XREG1 在 X_REG=1 或 3 时使用
CEX2	输入	XREG2（第二个 X 端口输入寄存器）的时钟使能信号，XREG2 在 X_REG=2 或 3 及 MODEIN[0]=0 时使用
CEX3	输入	XREG3（第三个 X 端口输入寄存器）的时钟使能信号，X_REG3 在 CXO_REG=3 时使用
RSTX	输入	高有效的 X 复位信号，复位所有 X 端口输入寄存器
CEXB	输入	XB 端口寄存器的时钟使能信号
RSTXB	输入	高有效的 XB 复位信号，复位 XB 输入寄存器
CEY1	输入	YREG1（第一个 Y 端口输入寄存器）的时钟使能信号，YREG1 在 Y_REG=1 或 3 时使用
CEY2	输入	YREG2（第二个 Y 端口输入寄存器）的时钟使能信号，YREG2 在 Y_REG=2 或 3 且 MODEIN[4]=0 时使用
RSTY	输入	高有效的 Y 复位信号，复位所有 Y 端口输入寄存器
CEZ	输入	Z 端口输入寄存器的时钟使能信号
RSTZ	输入	高有效的 Z 复位信号，复位 Z 端口输入寄存器
CEPRE	输入	PREADD 寄存器的时钟使能信号
RSTPRE	输入	高有效的 PREADD 复位信号，复位 PREADD 寄存器
CEM	输入	MULT 寄存器的时钟使能信号
RSTM	输入	高有效的 MULT 寄存器复位信号，复位 MULT 寄存器
CEP	输入	Postadd 寄存器的时钟使能信号
RSTP	输入	高有效的 P 复位信号，复位 Postadd 寄存器
CEMODEIN	输入	MODEIN 寄存器的时钟使能信号
RSTMODEIN	输入	高有效的 MODEIN 复位信号，复位 MODEIN 寄存器
CEMODEY	输入	MODEY 寄存器的时钟使能信号

端口名	输入/输出	描述
RSTMODEY	输入	高有效的 MODEY 复位信号，复位 MODEY 寄存器
CEMODEZ	输入	MODEZ 寄存器的时钟使能信号
RSTMODEZ	输入	高有效的 MODEZ 复位信号，复位 MODEZ 寄存器
P[47:0]	输出	APM 的 48 位并行数据输出
CPO[47:0]	输出	级联 P 输出，连接其他 APM 的 CPI
COUT	输出	级联 CIN 输出，连接其他 APM 的 CIN
CXO[29:0]	输出	级联 X 输出，连接其他 APM 的 CXI
CXBO[24:0]	输出	级联 XB 输出，连接其他 APM 的 CXBI

2.2.3 参数说明

表 2-2 GTP_APM_E2 参数说明列表

参数名	有效值	默认值	功能描述
USE_POSTADD	0,1	0	Postadd 功能使能 0:不使能累加单元 1:使能累加单元
USE_PREADD	0,1	0	Preadd 功能使能 0:不使能预加单元 1:使能预加单元
USE_MULT	0,1	1	乘法器使能 0:不使用乘法器 1:使用乘法器
CXO_REG	0,1,2,3	0	X 级联输出寄存器延迟选择， 0/1/2/3 分别对应 0/1/2/3 级输出寄存器，需要使能对应寄存器，具体结构见图 2-2
X_REG	0,1,2,3	0	X 端口寄存器使能，为 0 且 MODEIN[0]=0 时，不使用 X_REG；为 1 时使用 XREG1；为 2 或 3 且 MODEIN[0]=0 时使用 X_REG1 和 XREG2；需要使能对应寄存器。 MODEIN[0]=1 使用 XREG1，与此参数无关，需要使能对应寄存器。
XB_REG	0,1	0	XB 端口寄存器使能 0:不使能 1:使能
Y_REG	0,1,2,3	0	Y 端口寄存器使能，为 0 且 MODEIN[4]=0 时，不使用 Y_REG；为 1 时使用 YREG1；为 2 且 MODEIN[4]=0 时使用 YREG2；或 3 且 MODEIN[4]=0 时使用 YREG1 和 YREG2；需要使能对应寄存器。 MODEIN[4]=1 使用 Y_REG1，与此参数无关，需要使能对应寄存器。
Z_REG	0,1	0	Z 端口寄存器使能 0:不使能 1:使能
PREADD_REG	0,1	0	预加寄存器使能 0:不使能 1:使能
MULT_REG	0,1	0	Mult 寄存器使能 0:不使能 1:使能

参数名	有效值	默认值	功能描述
P_REG	0,1	0	P 端口输出寄存器使能 0:不使能 1:使能
MODEIN_REG	0,1	0	MODEIN 寄存器使能 0:不使能 1:使能
MODEY_REG	0,1	0	MODEY 寄存器使能 0:不使能 1:使能
MODEZ_REG	0,1	0	MODEZ 寄存器使能 0:不使能 1:使能
X_SEL	0,1	0	X 输入选择: 0: 选择 X 输入, 1: 选择 CXI 级联输入
XB_SEL	0,1,2,3	0	XB_MUX 向后传递选择: 0: XB 直接输入 1: CXBI 级联输入 2/3: 选择 CXO[24:0]
ASYNC_RST	0,1	0	异步复位使能 0: 选择同步复位 1: 选择异步复位
USE_SIMD	0,1	0	SIMD 模式选择, 为 1 时使能 SIMD 模式
P_INIT0	48'h0-48'hffffffff	48'h0	不使用 Rounding 功能时, 将"P_INIT0"设置为 48'h0; 使用 Rounding 功能时, 参照 2.4.1Rounding 功能 说明设置
P_INIT1	48'h0-48'hffffffff	48'h0	不使用 Rounding 功能时, 若不使用累加功能将"P_INIT1"设置为 48'h0, 使用累加功能时"P_INIT1"可作为做累加运算的初值; 使用 Rounding 功能时, 参照 2.4.1Rounding 功能 说明设置
ROUNDMODE_SEL	0,1	0	Round 模式参数, 配合"P_INIT0"、"P_INIT1"使用; 不使用 Rounding 功能时, 将"ROUNDMODE_SEL"设置为 0; 使用 Rounding 功能时, 参照 2.4.1Rounding 功能 说明设置
CPO_REG	0,1	0	CPO, COUT 输出寄存器使能
USE_ACCLOW	0,1	0	配置累加器反馈 0: 使用全部 48bit 1: 仅使用低 17bit
CIN_SEL	0,1	0	选择进位 CIN_i 输入来源 0: MODEY[2] MODEZ[3] 1: CIN
GRS_EN	"TRUE", "FALSE"	"TRUE"	全局复位使能 "FALSE":不使能 "TRUE":使能

2.2.4 模式说明

APM 包含一个 25bit 预加器, 一个 25bit*18bit 的二进制补码乘法器, 其中的数据通路选择由 MODEIN、MODEY、MODEZ 控制, 之后连接到累加器单元。APM 的输入连接到

各算数单元，X、Y 输入都可以手动选择寄存 1 或 2 次以满足不同 APM 应用。XB 与 Z 输入均可以进行一次寄存，控制信号输入也可以选择进行一次寄存。要达到 Data Sheet 最大速度需要使能全部流水寄存器。

控制 APM 工作模式的为 MODEIN[4:0]，MODEY[2:0]，MODEZ[3:0]三组信号，它们的端口说明如表 2-3 所示，其中 MODEIN 详细的列表请见表 2-4(可控制预加器选择 XB 端数据)。两级的 X1,X2 和 Y1, Y2 两级寄存器可实现时分复用的功能下各个功能数据延迟的一致性。

表 2-3 MODE 端口功能说明

信号	说明
MODEIN[4]	0: Y2 1: Y1
MODEIN[3]	Preadder 加减选择 0: 加 1: 减
MODEZ[3]	对 ZMUX 输出按 bit 取反, 1 有效（之后在累加器处通过 CIN_i 加 1 实现取负）
MODEZ[2:0]	ZMUX 输出选择 0: 零 1: 累加器反馈 2: Z 输入端 3: 上一级 APM 结果级联输出 4: 上一级 APM 结果级联输出右移 17 位 5: 上一级 APM 结果级联输出右移 24 位 6: 上一级 APM 结果级联输出右移 16 位 7: 上一级 APM 结果级联输出右移 8 位
MODEY[2]	对 YMUX 输出按 bit 取反, 1 有效（之后在累加器处通过 CIN_i 加 1 实现取负）
MODEY[1: 0]	YMUX 输出选择 0: 零 1: 乘法器输出 2: 累加器反馈 3: 宽位宽数据{X[29: 0],Y[17: 0]}

注：表中功能具体结构见图 2-1

表 2-4 MODEIN 功能列表

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

MODEIN [3]	MODEIN [2]	MODEIN [1]	MODEIN [0]	"PREADD"	X_MULT
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 工作模式

2.3.1 乘法模式

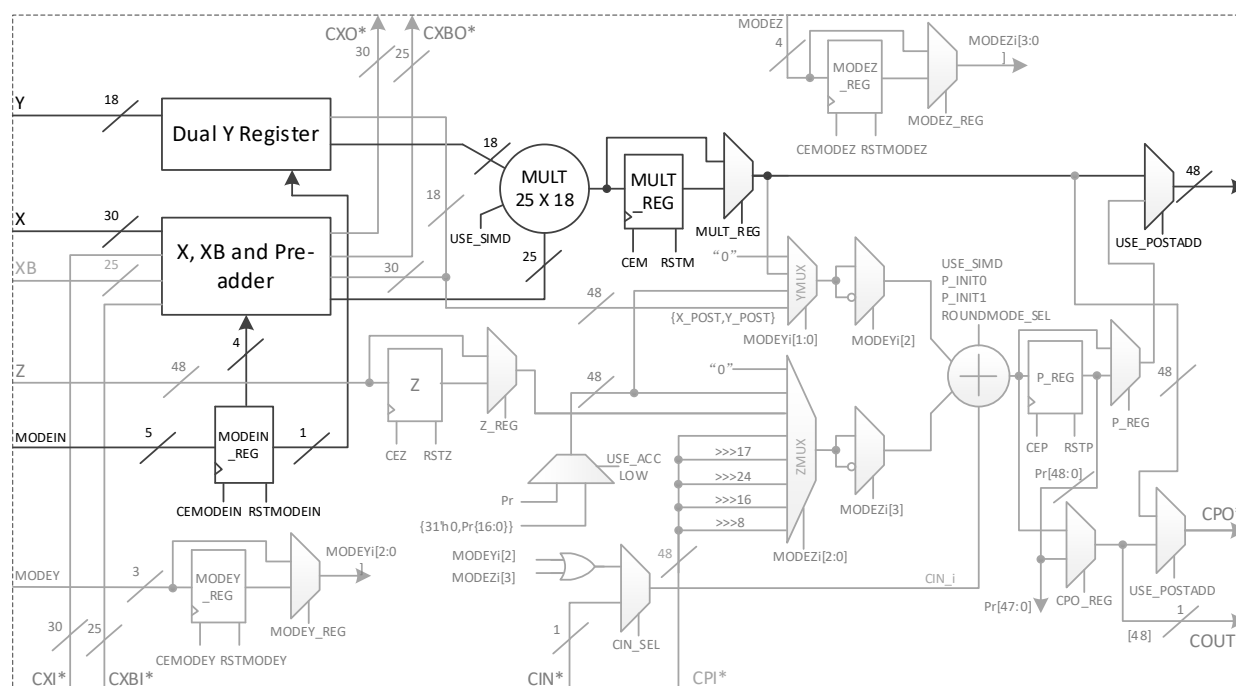


图 2-5 乘法模式应用示意图

如图 2-5 所示，APM 配置成乘法模式时，可实现表达式为：

$$P=X \times Y$$

APM 配置成乘法模式时主要特性有：

- 每个 APM 可支持一个 25×18 ($USE_SIMD=0$) 或两个 12×9 运算 ($USE_SIMD=1$)

➤ 输入/输出寄存器可选

使能APM中的Preadd Unit后，APM可通过XB端口配置成带预加的乘法模式，其算术表达式为：

$$P=Y \times (X \pm XB)$$

该模式下，每个APM可实现两个 $(12 \pm 12) \times 9$ 运算，或1个 $(25 \pm 25) \times 18$ 运算，支持动/静态预加/减控制。除了可选的输入/输出寄存器，带预加的乘法模式还可选择使能内部流水寄存器，其应用示意图如图 2-6所示：

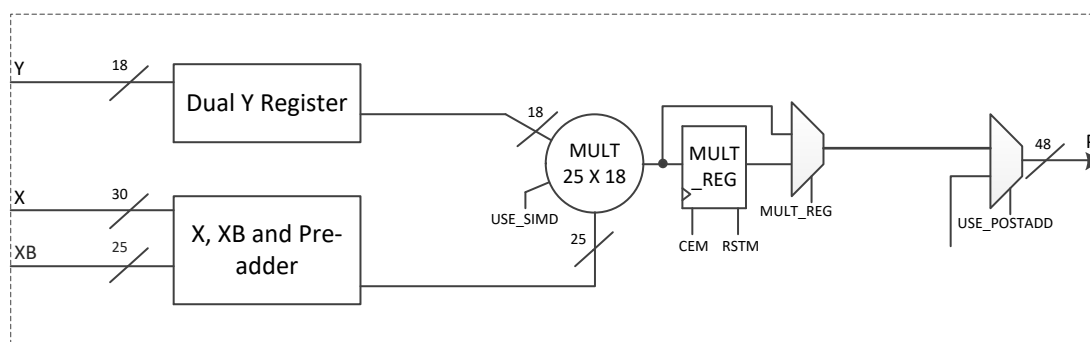


图 2-6 带预加的乘法模式应用示意图

乘法模式的典型时序如图 2-7，其中 $P=Y \times (X+XB)$ 。只使能一级寄存器时，P端对应X,Y端输入数据的计算结果在下一个时钟上升沿输出。

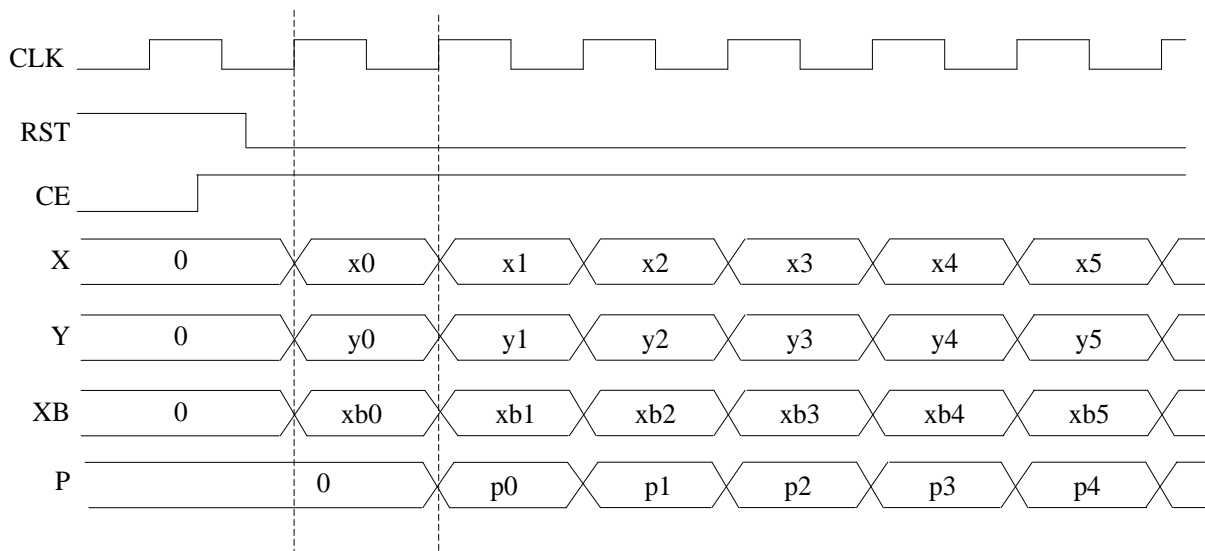


图 2-7 乘法模式的典型时序图

2.3.2 通用乘加模式

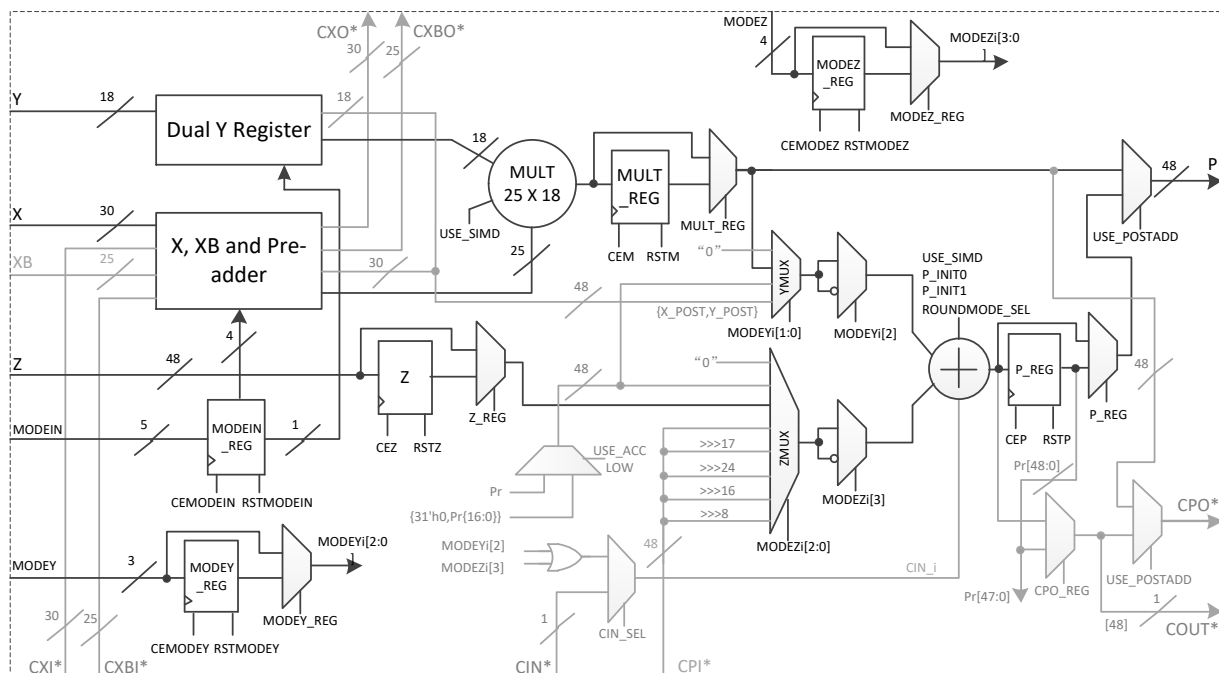


图 2-8 通用乘加模式应用示意图

如图 2-8所示，APM配置成通用乘加模式时，可实现表达式为：

$$P=X \times Y \pm Z \text{ 或 } P=-X \times Y+Z \text{ 或 } P=-X \times Y-Z-1$$

通用乘加模式的主要特性有：

- 每个 APM 可实现一个 $25 \times 18 \pm 48$ ($USE_SIMD=0$) 或两个 $12 \times 9 \pm 24$ 运算 ($USE_SIMD=1$)
- 支持有符号数，加减都可动/静态控制
- 可选的输入/输出寄存器

使用GTP_APM_E2单元进行例化。通用乘加模式的典型时序如图 2-9，其中 $P=X \times Y+Z$ ，使能一级寄存器后，P端对应X,Y端输入数据的计算结果在下一个时钟上升沿输出。

当 $MODEY[2]==1$ ，乘法器输出取反（在Ymux处取反）后，在累加器处加1，等价于乘法器取负。

当 $MODEZ[3] == 1$ ，Zmux输出取反后，在累加器处加1，等价于Zmux取负。

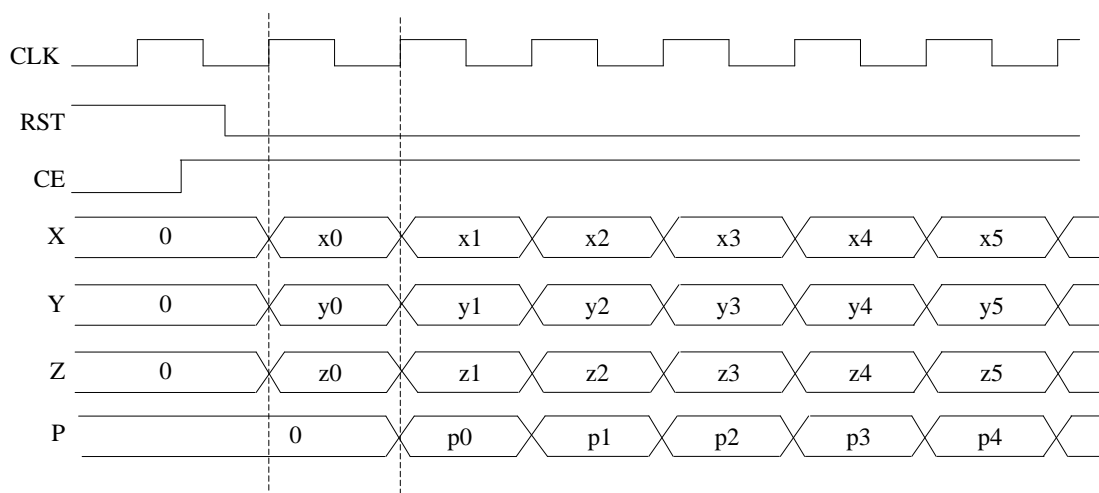


图 2-9 通用乘加模式的典型时序

2.3.3 乘累加模式

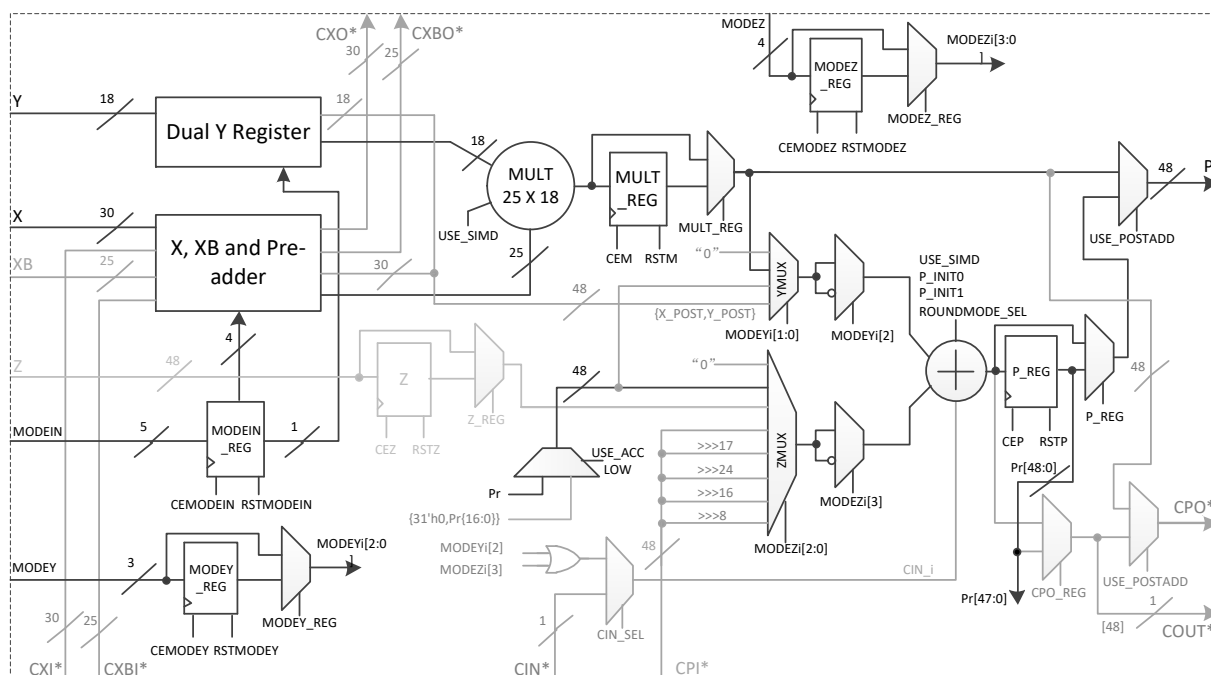


图 2-10 乘累加模式应用示意图

如图 2-10所示，APM配置成乘累加模式时，可实现表达式为：

$$P = P \pm X \times Y \text{ 或 } P = -P \pm X \times Y \text{ 或 } P = -P - X \times Y - 1$$

乘累加模式的主要特性有：

- 一个 APM 可实现一个 25×18 的乘累加运算(P 为 48bit)或两个 12×9 的乘累加运算 (P 为两个 24bit)
- 支持有符号数，支持动/静态累加/减控制
- 可选的输入/输出寄存器

- 使能 APM 中的 Preadd Unit 后, APM 可配置成带预加的乘累加模式, 可实现表达式为:

$$P=P_{\pm}Y \times (X_{\pm}XB) \text{ 或 } P=-P_{+}Y \times (X_{\pm}XB) \text{ 或 } P=-P_{-}Y \times (X_{\pm}XB)-1$$

带预加的乘累加模式有两级内部流水寄存器可供选择，其应用示意图如图 2-11 所示：

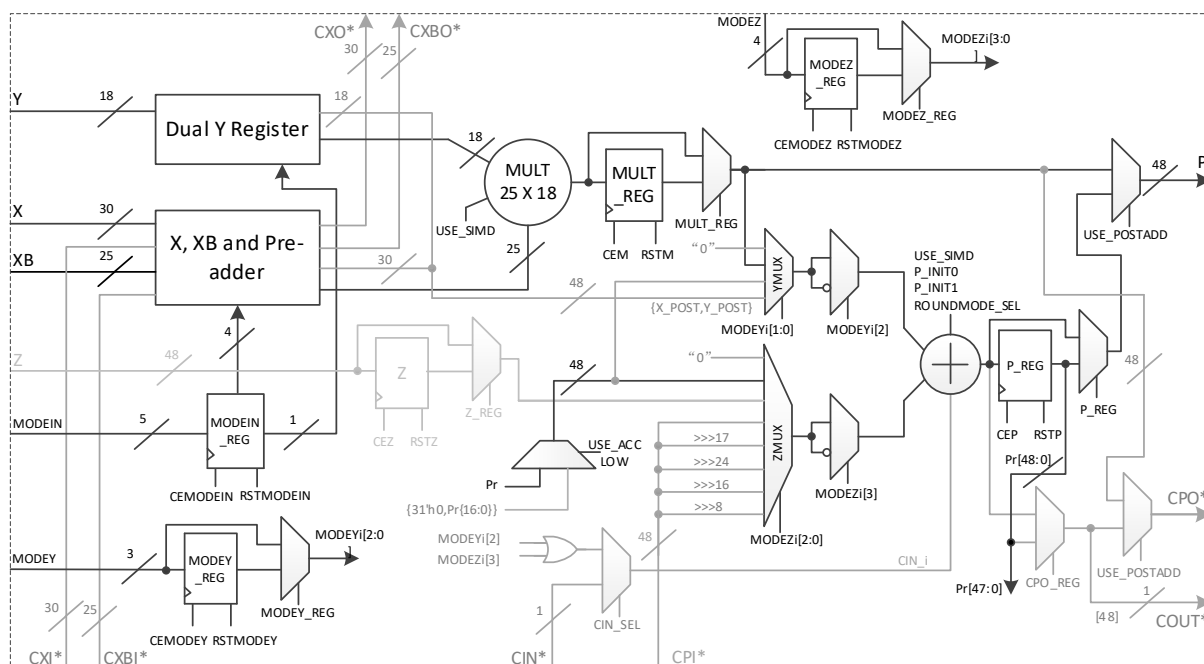


图 2-11 带预加的乘累加应用示意图

乘累加模式的典型时序如图 2-12，其中 $X=1$ ， $Y=1$ ， $P=P+X \times Y$ ，P在CEP有效时才能输出，使能P_REG后，只有一级寄存器时，P端对应X,Y端输入数据的计算结果在下一个时钟上升沿输出。

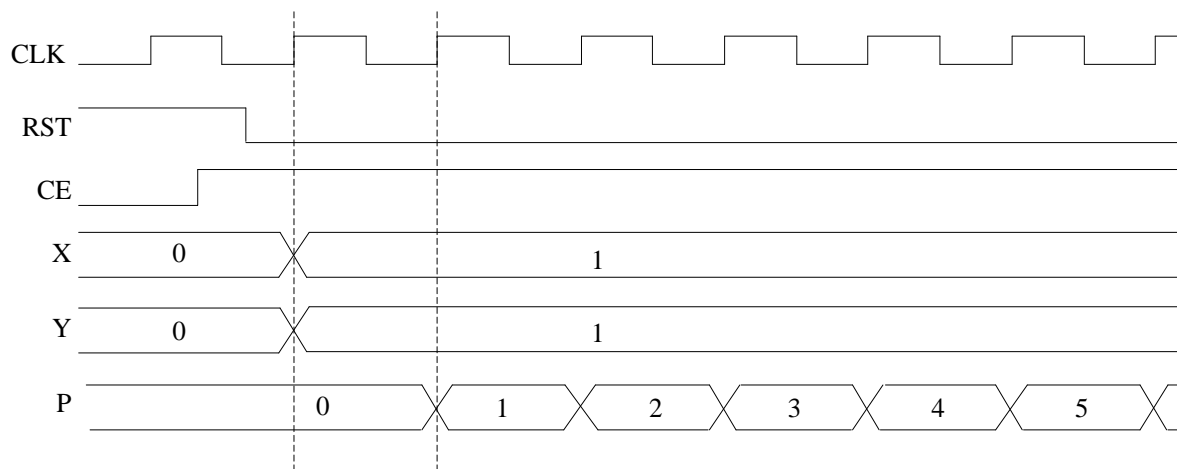


图 2-12 乘累加模式的典型时序

2.3.4 FIR 应用

通过APM级联也可实现FIR运算，典型的FIR filter可以描述为以下表达式：

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

x为输入的数据流，y为输出的数据流，h为系数。图 2-13为采用4个APM级联实现的8阶 Symmetric systolic FIR运算的功能实现图。

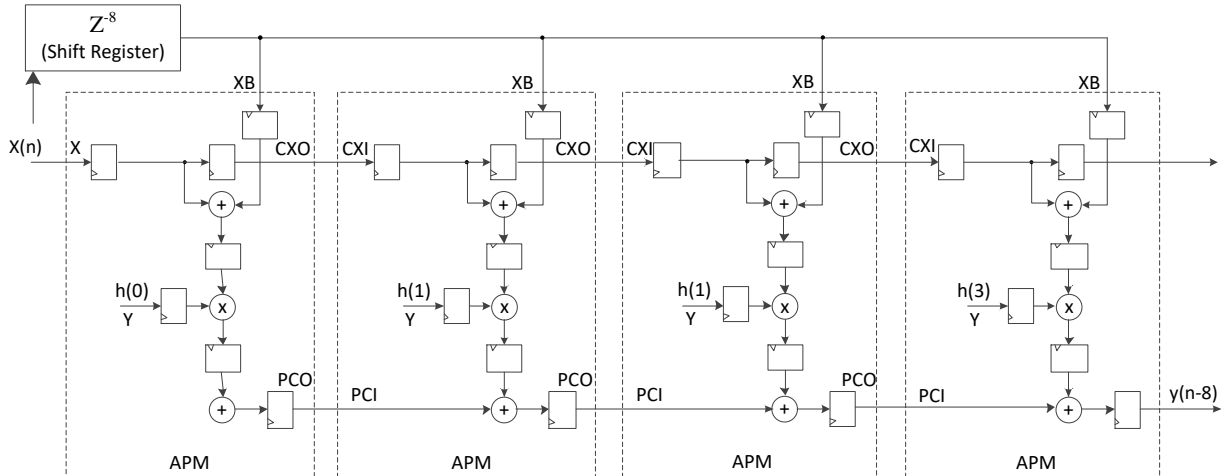


图 2-13 Symmetric systolic FIR 功能示意图

通过对GTP_APM_E2单元的配置，可以实现多种形式的FIR运算，具体的实现类型及参数配置建议如表 2-5所示（所有FIR配置里都有CPO_REG = 1）。

表 2-5 典型 FIR 滤波器实现

滤波器类型	第一级 APM	中间级 APM	最后一级 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 (偶数 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 (奇数 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

滤波器类型	第一级 APM	中间级 APM	最后一级 APM
2-Channel Symmetrical FIR (偶数 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 (奇数 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 具体功能描述

2.4.1 Rounding 功能

Logos2 APM 的累加单元支持乘法运算和乘加运算的 Direct rounding 和 Round-toward-nearest。Rounding功能是指将累加器输出的低Nbit的信号认为是小数位，然后通过舍弃小数位得到最近的整数值。实际使用时，Postadder会根据不同的Rounding模式计算得到48bit的结果，用户需要自行截取P[47:N]作为取整， $N \in [1, 45]$ ， $N=0$ 即没有舍入。其详细功能如表 2-6所示。

表 2-6 Rounding 详细功能列表

Rounding 功能		功能举例说明	乘法功能	乘加减功能	实现方式 (N 为浮点位置)
Direct rounding	Round-ceiling	3.1,3.5,3.9>4 -3.1,-3.5,-3.9>-3	支持	支持	$+2^N-1$,然后取整
	Round-floor	3.1,3.5,3.9>3 -3.1,-3.5,-3.9>-4	支持	支持	直接取整
	Round-toward zero	3.1,3.5,3.9>3 -3.1,-3.5,-3.9>-3	支持	支持	对负数 $+2^N-1$,正数直接取整
	Round-away from zero	3.1,3.5,3.9>4 -3.1,-3.5,-3.9>-4	支持	支持	对正数 $+2^N-1$,负数直接取整
Round-toward-nearest	Symmetric Round-half-up	3.1>3; 3.5,3.9>4 -3.1>-3; -3.5,-3.9>-4	支持	支持	正数 $+2^{(N-1)}$,负数 $+2^{(N-1)}-1$,然后取整
	Asymmetric Round-half-up	3.1>3; 3.5,3.9>4 -3.1,-3.5>-3; -3.9>-4	支持	支持	$+2^{(N-1)}$,然后取整
	Symmetric Round-half-down	3.1,3.5>3; 3.9>4 -3.1,-3.5>-3; -3.9>-4	支持	支持	正数 $+2^{(N-1)}-1$,负数 $+2^{(N-1)}$,然后取整
	Asymmetric Round-half-down	3.1,3.5>3; 3.9>4 -3.1>-3; -3.5,-3.9>-4	支持	支持	$+2^{(N-1)}-1$,然后取整

Logos2 APM实现不同Rounding功能的参数配置如表 2-7:

表 2-7 Rounding 各功能配置列表

Rounding 功能		ROUNDMODE_SEL	P_INIT0	P_INIT1
Direct rounding	Round-ceiling	0	2^N-1	0

Rounding 功能		ROUNDMODE_SEL	P_INIT0	P_INIT1
	Round-floor	0	0	0
	Round-toward zero	1	0	2^N-1
	Round-away from zero	1	2^N-1	0
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 时分复用

Logos2的APM支持时分复用功能，表 2-8给出了时分复用功能支持的运算功能及其参数配置。

表 2-8 时分复用详细表格

配置 功能	寄存器情况			动态控制配置			
	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
宽位 ADD/SUB	2	0	1	0	0	2'b11	3'b010

1. 做时分复用功能时，APM可支持功能都配置成三级寄存器。
2. Input_reg包括所有数据通路和控制信号的输入寄存器，X和Y两路做时分复用时，需要有两个输入寄存器延迟，即X/Y_REG[1:0]要设置为2'b11。时分复用功能中Pipe_reg即为MULT_REG，Output_reg即为P_REG。
3. 到达POSTADD的Ymux数据延迟是2个CLK延时，MODEY[1:0]在APM中只有1个输入寄存器。需错开一个CLK延时控制或者MODEY[1:0]在APM外部补偿一个CLK延时。
4. Z路径和MODEZ的延迟都为1个CLK延时，需与X,Y路径上的数据错开一个延时控制，或者在APM外部补充一个CLK延时。
5. MODEIN [0], MODEIN [4]配置固定为1，POSTADD固定使能，时分复用中若有PREADD功能，PREADD固定使能，并且PREADD_REG不使能，若没有PREADD可关闭。
6. PREADD_MULT25和PRE_MULTADD25没有PREADD_REG，无法达到最高的频率。

3 附录

3.1 级联可实现功能

通过对 APM 级联可以获得更高位宽运算，表 3-1 描述了 APM 级联可实现的功能。表 3-1 为举例，仅供参考。级联输入端口仅可连接上一级级联输出端口，级联输出端口仅可连接下一级级联输入端口，未使用时可悬空。

表 3-1 APM 级联可实现功能描述

APM 级联	可实现功能
2 个 apm 级联	2 个 $(12 \times 9 + 12 \times 9)$
	2 个 $[(12+12) \times 9 + (12+12) \times 9]$
	输出位宽为 65 位的 MULTACC25
	输出位宽为 96 位的 MULTACC25
	输出位宽为 96 位的 ADD/ACC(加法)
	$25 \times 18 + 25 \times 18$
	$(25+25) \times 18 + (25+25) \times 18$
	输出位宽为 65 位的 PREADD_MULTACC25
	输出位宽为 96 位的 PREADD_MULTACC25
	49×18 (只使用 CPI 右移 24 位功能)
	42×18 (只使用 CPI 右移 17 位功能)
	25×35 (只使用 CPI 右移 17 位功能)
	25×34 (只使用 CPI 右移 16 位功能)
	$(25+25) \times 35$
3 个 apm 级联	2 个输出位宽为 24 位的 $[MAC + (12 \times 9 + 12 \times 9)]$
	2 个输出位宽为 24 位的 $[MAC + ((12+12) \times 9 + (12+12) \times 9)]$
	输出为 48 位的 $MAC + (25 \times 18 + 25 \times 18)$
	输出为 48 位的 $MAC + ((25+25) \times 18 + (25+25) \times 18)$
	59×18 (只使用 CPI 右移 17 位功能)
	25×52 (只使用 CPI 右移 17 位功能)
	25×50 (只使用 CPI 右移 16 位功能)
	73×18 (只使用 CPI 右移 24 位功能)
	$MAC + 42 \times 18$
	$MAC + 25 \times 35$
	$MAC + (25+25) \times 35$
4 个 apm 级联	2 个 $[(12 \times 9 + 12 \times 9) + (12 \times 9 + 12 \times 9)]$
	2 个 $[(12+12) \times 9 + (12+12) \times 9 + ((12+12) \times 9 + (12+12) \times 9)]$
	输出为 65 位的 $MAC + (25 \times 18 + 25 \times 18)$
	输出为 65 位的 $MAC + ((25+25) \times 18 + (25+25) \times 18)$

	$(25 \times 18 + 25 \times 18) + (25 \times 18 + 25 \times 18)$
	$((25 + 25) \times 18 + (25 + 25) \times 18) + ((25 + 25) \times 18 + (25 + 25) \times 18)$
	$35 \times 25 + 35 \times 25$ (只使用 CPI 右移 17 位功能)
	$42 \times 18 + 42 \times 18$ (只使用 CPI 右移 17 位功能)
	$25 \times 34 + 25 \times 34$ (只使用 CPI 右移 16 位功能)
	$49 \times 18 + 49 \times 18$ (只使用 CPI 右移 24 位功能)
	42×35 (只使用 CPI 右移 17 位功能)
	25×69 (只使用 CPI 右移 17 位功能)
	76×18 (只使用 CPI 右移 17 位功能)
	49×34 (使用 CPI 右移 16 和 8 位功能)
	$36 \times (25 + 25) + 36 \times (25 + 25)$

3.2 设计建议

- 进行高速滤波流水线应用时，建议使用 APM 级联功能，包括加法器级联。
- 乘法器由 MODEY、MODEZ、MODEIN 等信号动态控制，具有很强的灵活性，设计中使用寄存器与动态配置模式，相较乘法器组合可更充分利用 APM 性能。
- 为实现 APM 的最高性能，需要使用全部流水寄存器，如果对延迟有要求，无法使用全部寄存器，请尽量使能 MREG。
- 使用内部级联端口而非通过 fabric routing 连接可以节省功耗。
- 进行较小位宽的运算时应进行符号位扩展。
- 多个 APM 进行级联时，不同信号通道的流水级数应匹配。

3.3 使用 GTP 例化为乘法模式

带预加的乘法器GTP例化模板如下：

- 通过 USE_PREADD=1 使能预加功能，不使能 POSTADD
- 不使用 X 端口、Y 端口等寄存器，使用 MULT_REG
- MODEIN=5'b00110, X_MULT=X2+XB, 预加器为加法，选择 Y2, 由于未使能 X/Y 端口寄存器，X1/X2 与 X 相当，Y1/Y2 与 Y 相当
- MODEY=3'b000, MODEY 控制 POSTADD 功能，此处没有使能 POSTADD
- MODEZ=4'b0000, Zmux 选择为 0, 不对 Zmux 输出取反，此处没有使能 POSTADD
- ROUNDMODE_SEL=0, P_INIT0 =48'd0, P_INIT1=48'd0, rounding 功能设置为 Round-floor

➤ 输出结果 $P=(X+XB)*Y$

```
GTP_APM_E2 #(
    .GRS_EN ("TRUE"),
    .USE_POSTADD (0),
    .USE_PREADD (1),    //使能预加功能
    .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输出为0
.MODEZ(4'b0000), //ZMUX输出为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 使用 GTP 例化为乘加模式

通用乘加的GTP例化模板如下：

- 通过 USE_POSTADD=1 使能累加功能，不使能 PREADD 功能
- 不使能 X 端口、Y 端口等寄存器，使用 P_REG
- MODEIN=5'b00010, X_MULT=X2, 预加器为加法（未使能 PREADD 功能），选择 Y2，由于未使能 X/Y 端口寄存器，X1/X2 与 X 相当，Y1/Y2 与 Y 相当
- MODEY=3'b001, MODEY 控制 POSTADD 功能，选择为乘法器输出，不对 Ymux 输出取反

- MODEZ=4'b0010, Zmux 选择为 Z 输入端, 不对 Zmux 输出取反
- ROUNDMODE_SEL=0, P_INIT0 =48'd0, P_INIT1=48'd0, rounding 功能设置为 Round-floor
- 输出结果为 $P=X*Y+Z$

```
GTP_APM_E2 #(
    .GRS_EN ("TRUE"),
    .USE_POSTADD (1), //使能累加功能
    .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输出乘法器结果
.MODEZ(4'b0010),    //ZMUX输出为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 使用 GTP 例化为乘累加模式

乘累加的GTP例化模板如下：通过USE_POSTADD=1使能累加功能，不使能PREADD功能

- 不使能 X 端口、Y 端口等寄存器，使用 P_REG
- MODEIN=5'b00010, X_MULT=X2, 预加器为加法（未使能 PREADD 功能），选择

Y2，由于未使能 X/Y 端口寄存器，X1/X2 与 X 相当，Y1/Y2 与 Y 相当

- MODEY=3'b001，MODEY 控制 POSTADD 功能，选择为乘法器输出，不对 Ymux 输出取反（取反结果变为 $P=P-X*Y$ ）
- MODEZ=4'b0001，Zmux 选择为累加器反馈，不对 Zmux 输出取反（取反结果变为 $P=-P+X*Y$ ）
- ROUNDMODE_SEL=0，P_INIT0 =48'd0，P_INIT1=48'd0，rounding 功能设置为 Round-floor
- 在乘累加模式中，需注意先将 MODEZ 设为 4'b0000，将 APM 内部的累加值初始为零，再将 MODEZ 值设为 4'b0001 进行乘累加计算
- 输出结果为 $P=P+X*Y$

```
GTP_APM_E2 #(
    .GRS_EN ("TRUE"),
    .USE_POSTADD (1), //使能累加功能
    .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输出乘法器结果  
    .MODEZ(4'b0001),    //ZMUX输出累加器反馈  
  
    .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)  
);
```


免责声明

版权声明

本文档版权归深圳市紫光同创电子有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此文档中的任何部分公开、转载或以其他方式披露、散发给第三方。否则，公司必将追究其法律责任。

免责声明

1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。

2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。