APPLICATION NOTE

Mentor Synthesis Group

Pipelining Multipliers in Precision

Tom Hill – Synthesis Product Specialist

Mentor Graphics®

Introduction

Precision RTL Synthesis automatically performs an optimization technique called "pipelining' on multipliers for certain technologies during optimization. The pipelining process takes advantage of natural breaking points in the multiplier architecture to move banks of registers backwards into the multiplier logic to improve performance. Users can place several banks of registers at the multiplier outputs, which will be automatically pipelined during optimization.

Precision will pipeline multipliers with registers at the output during the mapping phase of optimization. This is not timing driven optimization process - pipelining will occur regardless of whether a design is meeting timing. Both Xilinx Virtex II and Altera Stratix include dedicated registered multipliers, which makes pipelining optimal implementation for both area and performance.

How Pipelining Works

Pipelining will 'absorb" registers at the output of a multiplier to improve performance. Simple enough, however, this statement generally results in 2 questions:

1. How many pipeline stages can the multiplier "absorb"?
2. What type of performance improvements can be obtained?

How many pipeline stages can be "absorbed"?

There is no limit to the number of pipeline stages that a user may place at the outputs of a multiplier when pipelining. There is, however, an optimal number that will deliver the maximum performance benefits. If fewer than the optimal number are used then the multiplier will see some performance benefits but not the maximum possible through pipelining. If the user places more than the optimal number then the "extra" pipeline stages will remain at the outputs of the multiplier and provide no additional performance gains. The optimal number of pipeline stages is calculated through the following simple equation.

Optimal # of pipeline stages = Log10 (input data bus width)

Although the answer to this question is somewhat mathematical it is not as confusing as it appears. For example if the input databus is 16 bits than the optimal number of pipeline stages is 4. The table below provides some common values

Input data bus width	Pipeline stages
8	3
16	4
32	5
64	6
128	7

Table 1 – Optimal number of pipeline stages

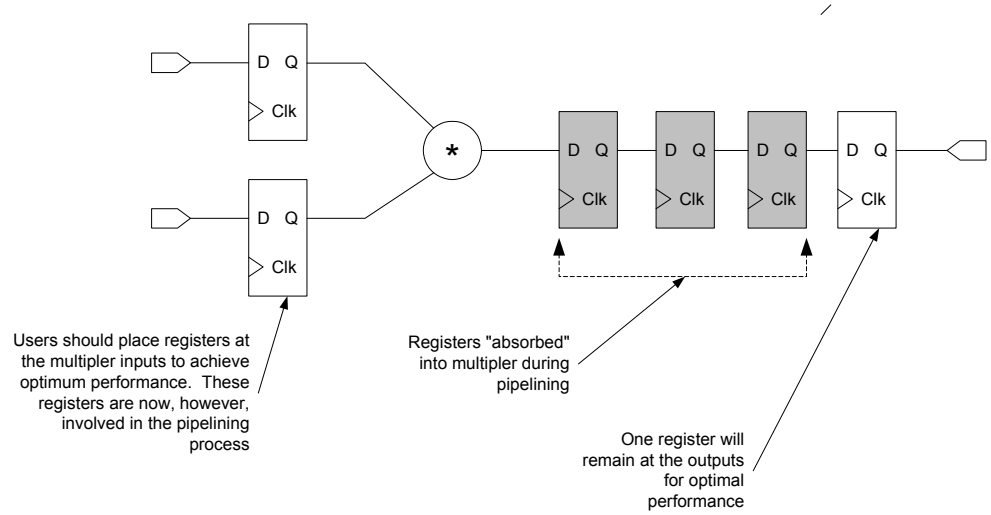

Figure 1 – Pipelined multiplier – before pipelining

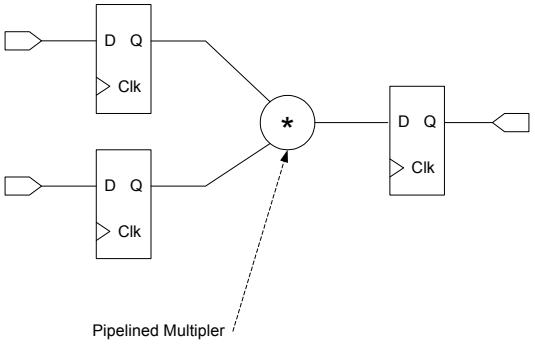

Figure 2 – Pipelined multiplier – after retiming

Performance Improvements achieved through Pipelining

Pipelining can improve the performance of a multiplier by 2X when the optimal number of pipeline stages is used. Using fewer pipeline stages will result in some improvement less that 2X. When more than the optimal number pipeline stages are used not additional performance gains are realized.

Supported Technologies

Pipelining is supported for the following technologies

Vendor	Device Family
Xilinx	Virtex
	Virtex E
	Virtex II
	Virtex II Pro
	Spartan II
	Spartan IIE
	Spartan III
Altera	APEX 20K
	APEX 20KC
	APEX 20KE
	APEX II
	Stratix
	Stratix GX