A Power Efficient Image Convolution Engine for Field Programmable Gate Arrays

Matthew French
mfrench@isi.edu
University of Southern California, Information Sciences Institute
3811 North Fairfax Dr, Suite 200
Arlington, VA 22203

I. Introduction

Field Programmable Gate Arrays (FPGAs) have become increasingly popular in high-speed front-end digital signal processing (DSP) applications. Many of the recent advances in FPGA device technology have directly impacted DSP performance, most notably with the inclusion of dedicated multiplier resources. With the availability of more multipliers with better precision, increasingly more sophisticated DSP kernels such as Finite Impulse Response (FIR) filters have grown from a few taps implemented using Lookup-Table (LUT)-based CORDICs[1] to hundreds of taps utilizing embedded multipliers. This particularly benefits two-dimensional filters such as those used in image processing, which are bound by multiplier performance[2,3].

While Moore's Law has provided more DSP processing power for an FPGA, there is a tradeoff of in power consumption. Dynamic power consumption is described as:

$$Power = \sum_i F_i * T_i * C_i * V_i^2 \qquad [4]$$

where F is the frequency, T is the toggling rate, C is the capacitance and V is the voltage of the ith component. Figure 1 shows the general trends of these variables over each successive family of Xilinx FPGAs. The voltage of a device decreases roughly every other generation, decreasing the power consumed. However the logic density and clock frequency gains far outpace the power savings contributed by the lower voltages, resulting in an exponentially increasing maximum power per device.

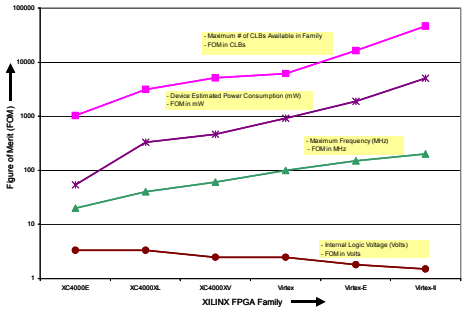

Figure 1. Xilinx FPGA Power Consumption Trends

This power trend shows that an FPGA user can no longer merely rely on the inherent power savings of a new architecture with a lower voltage to satisfy power budgets. Power must be considered as first-class design parameter on par with DSP functionality and maximum throughput. This is especially true for power critical domains such as space-based applications, handheld devices, and remote sensors. In this paper, we describe an image convolution kernel specifically designed for power savings. Critical to the success of this approach is considering how the design is mapped to the FPGA micro-architecture, and how to optimize for total switching capacitance as well as throughput. Routing is also considered as we eliminate monolithic global state machines in favor of distributed control flow. This, along with relative placement macros (RPMs), help deter the place and route engine from using long routing wires with high capacitance. In this paper we will outline the design for a 3x3 image convolution kernel, however the techniques are extensible for any size convolution. This design yields up to an 8.9x increase in power efficiency over a traditional design.

II. Key Findings

The first step of our approach was to examine the FPGA architecture itself and determine the capacitance of the various micro-architecture components and routing lines. The capacitance of the micro-architecture features were compiled from [5], Xilinx's power estimation spreadsheets[6], and our own test results using XPower and lab experiments. While capacitance estimates varied among the sources, the general trends and relative capacitance values remained consistent. Table 1 shows the capacitance values we used for the micro-architecture features for the Xilinx Virtex-II architecture.

Resource	Capacitance (pF)
Embedded Multiplier	1,196
Block Select RAM	880
CLB	26
Long-line Route	23
Hex-line Route	18
Double-line Route	13
Direct-Connect Route	5

Table 1. Virtex-II Resource Capacitance

From this table we can derive a priority list of optimization for a given kernel. For example, even though embedded multipliers are more efficient than implementing a multiplier in the equivalent number of CLBs, they still consume the most power, and a design minimizing the number of multipliers used, would be desirable. Overall dynamic power is dominated by routing; therefore,

minimizing the number and length of nets yields further power reduction.

The next step in our design process is to consider the DSP kernel itself. An important insight in how to optimize an image convolution kernel can be gained by considering its specific tap values. Example tap masks for blurring, sharpening, edge detection, derivative filtering, and template matching are shown in Figure 2.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

a) Blurring

-1	-1	-1
-1	8	-1
-1	-1	-1

b) Sharpening

-1	-1	+1
-1	+1	-1
+1	-1	-1

c) Edge-detection

-1	-1	-1
0	0	0
+1	+1	+1

d) Derivative (Prewitt)

-1	0	+1
-1	0	+1
-1	0	+1

-6	4	11
-1	9	1
5	7	-3

e) Template

Figure 2. Convolution Tap Masks

In these examples, it is evident that there is typically symmetry within a tap mask and tap values are often reused. Blurring masks use one unique filter tap value, sharpening and edge-detection use two, derivative filters use three, and template filters can use from one to nine unique filter tap values depending on the exact template.

Our primary optimization is to leverage tap symmetry to minimize the number of high capacitance multipliers per pixel. The approach is to sum all samples at all matching taps before multiplying by the tap value. This is similar to the traditional optimization used in one-dimensional symmetric FIR filters, where samples are summed at symmetric taps to reduce the number of multipliers by 2x [7]. In image convolution, a similar technique can be extended to the 2-dimensions, achieving a reduction of up to 9 multipliers per tap mask. In the full paper, the details of the necessary overhead logic to implement this technique will be explained. Figure 3 depicts the top-level implementation of the design, which relies on multiplexers and adjustable adder trees to dispatch the proper data to the correct filter tap multipliers. The design then uses a fixed number of multipliers, with the number of pixels we are able to process in parallel dependent upon the number of unique taps in the filter mask.

Exploiting filter tap re-use is the most significant contributor to power savings in this design; however other low-power design techniques such as distributed flow control and RPMs help minimize the impact of the overhead logic by reducing the total routing resource requirements.

In our testing, we ran a 512 x 512 8-bit image using 8-bit filter taps through our kernel for each case of unique filter taps. Preliminary results showing the total non-quiescent energy consumed and the improvement over a standard convolution kernel implementation are shown in Table 2. Here we can see that for small numbers of filter tap re-use, we lose 1% of power efficiency due to the extra overhead logic of our design. However for more common cases where there is more symmetry to exploit, we achieve up to almost 9x reduction in power. The step function in energy used is due to the quantization effects of distributing the number of unique filter taps over a finite number of multipliers.

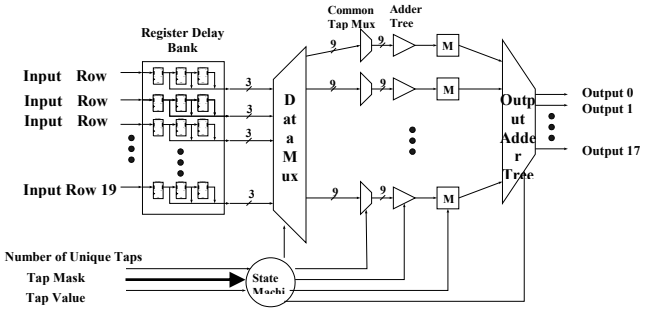

Figure 3. Top Level Convolution Architecture

Number of Unique Filter Taps	Energy (Joules)	Power Efficiency Increase Factor
Benchmark (9)	1.15×10^{-6}	NA
9	1.16×10^{-6}	0.99
8	1.16×10^{-6}	0.99
7	1.16×10^{-6}	0.99
6	7.74×10^{-7}	1.49
5	7.74×10^{-7}	1.49
4	5.81×10^{-7}	1.99
3	3.87×10^{-7}	2.99
2	2.58×10^{-7}	4.49
1	1.29×10^{-7}	8.99

III. Conclusions

Considering power as a first class FPGA design constraint early in the design stage allows us to make optimizations that have a considerable impact on the power used. These techniques are often complimentary to optimizations necessary to achieve higher clocking frequencies.

REFERENCES

[1] B. Nelson, "Configurable Computing and Sonar Processing - Architectures and Implementations," *ASILOMAR 2001*, November 2001.

[2] A. Benedetti, "Image Convolution on FPGAs: the implementation of a Multi-FPGA FIFO structure," 24th. EUROMICRO Conference Volume 1 (EUROMICRO'98), August 25 - 27, 1998,Sweden.

[3] S. Perri, et. al., "SIMD 2-D Convolver for Fast FPGA-based Image and Video Processors," MAPLD, September 2003.

[4] Xilinx XPower Manual, www.xilinx.com

[5] L. Shang, A. Kaviani, and K. Bathala, "Dynamic Power Consumption in Virtex-II FPGA Family," FPGA '02, Monterey, California, February, 2002.

[6] www.xilinx.com/cgi-bin/power_tool/web_power_tool.pl?power_Virtex2

[7] R. Andraka, A. Berkin, "FPGAs Make a Radar Processor on a Chip a Reality," ASILOMAR 1999, October, 1999.