

# Trends in Reconfigurable Computing: Applications and Architectures

Removed for Blind Review

**Abstract**— Since the release of the first commercial field programmable gate array (FPGA) in 1985, devices have enjoyed continuous improvements in all metrics due to technology scaling, architectural advances and the addition of features. In this paper, we explore performance and utilization trends associated with research designs as a function of FPGA technology progression. The data used is a subset of designs presented at the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM) over the past 20 years. These are compared to trends from theoretical and vendor sources, models generated and comparisons made. Finally, we compare operating frequency trends from our analysis to the trends exhibited by a set of vendor IP cores mapped to four generations of devices. The results of this investigation suggest that design implementations are generally following the theoretical trends and that the inclusion of embedded hard IP blocks has provided designers with additional performance benefits.

**Keywords**- Moore's Law; FPGA; trends; scaling;

## I. INTRODUCTION

As evidenced in references [1] and [2], Field-Programmable Gate Arrays (FPGAs) have come a long way since their first introduction. While the concept of reconfigurable computing was proposed by Estrin in the 1960s [3], practical applications only became possible after commercial field programmable gate arrays (FPGAs) were released in 1985. Since then, devices have enjoyed continuous improvements in all metrics including operating frequency, achievable density, and most recently, power. The primary driver of these improvements has been the dramatic improvement in technology over the years; FPGAs have been constructed in technologies ranging from 2.0 microns in 1985 down to 20 nanometers today. Also important, however, has been advances in FPGA Computer Aided Design (CAD) flow algorithms and architectures, including improved switch block and lookup table (LUT) architectures, embedded blocks, and novel optimization techniques. Today, the latest high-end devices have over a million LUTs, megabytes of embedded memory, thousands of digital signal processing (DSP) blocks and almost one hundred high speed transceivers.

To chart the course for the future of the FPGA industry, it is important to understand how these devices have been used throughout the years. In this paper, we explore the performance and utilization trends associated with application implementations over time as the underlying technology of FPGAs has improved. Although there have been several

retrospectives and surveys [1, 2, 4, 5], our motivations are different. Rather than focus on the architectures, CAD algorithms, or applications themselves, this work investigates how applications have been able to *use* the evolving technology over the years. We are not aware of any publications to date that compare third-party designs across different device generations and process technologies. Yet this is essential; understanding exactly what designers have been able to achieve with this technology, and how this changes over time, will provide insight into future FPGA trends. More specifically, our contributions include:

- models of application performance and resource utilization across various generations of devices,
- an analysis of how a large number of published research implementations compare with the theoretical maximum performance and resource usage on the devices,
- a comparison of the operating frequency trends from the research designs versus our benchmark designs and a discussion of how these trends can be extrapolated to future designs and devices.

We use a subset of all research application designs presented at the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM) between 1995 and 2014 to ensure that we have sufficient data for this analysis. For each of these, we relate key performance metrics (i.e resource usage, operating frequency) to improvements in technology. We also compare the long term trends in operating frequency found in this initial study to those exhibited by mapping a set of vendor IP cores (used as benchmarks) to four different generations of devices. We then analyze how the operating frequency trends of the benchmarks compare to the long term trends demonstrated by the larger data set of research designs.

This study is restricted to FPGAs based on static random access memory (SRAM) technology due to limited data being available for FPGAs based on other technology (e.g. antifuse, flash). To support reproducible research, the designs and data used to produce our results will be made publically available.

The remainder of the paper is organized as follows. Section II summarizes background on scaling of integrated circuit technology and FPGA architectures. We review the metrics we evaluated in this paper and recommended normalization techniques to account for FPGA architectural changes in Section III. The methodology used to extrapolate

the data trends for analysis and the results of this analysis are presented in Section IV. Finally, Section V summarizes our conclusions and future work.

## II. BACKGROUND

In this section, we review the trends that have impacted advancements in process technology and the evolution of FPGA architectures and design tools. We also summarize previously published surveys.

### A. Process Technology Trends

The International Technology Roadmap for Semiconductors (ITRS) is a organization sponsored by the semiconductor industry leaders to track the progress of the technology and set goals and predictions for the future [6]. Since the early days of VLSI design, the following key trends in technology advancements have been observed.

1) *Moore's Law*: In 1965, Gordon E. Moore published his now-famous observation that the number of transistors in an integrated circuit will double roughly every year [7], based on his observing a trend in the increased number of transistors per area over the preceding 5 years and theorizing how the industry would continue to achieve this growth. This was soon updated to a doubling of the number of transistors every two years, and became known as "Moore's Law" [7]. This trend has held for five decades and is likely to continue for the near future.

2) *Dennard's Law*: "Dennard's Law", also referred to as "Dennard scaling," is the observation that as transistor feature sizes ( $\lambda$  or  $\lambda$ ) decrease, their power density (power/area) remains constant [8]. For this to occur, delay ( $\tau$ ) scales in proportion to the inverse of feature size ( $\tau \propto 1/\lambda$ ). Furthermore, total chip power,  $P$ , is inversely proportional to the inverse of feature size squared ( $P \propto 1/\lambda^2$ ).

### B. Evolution of Programmable Architectures

The programmable logic industry has made significant improvements in FPGA products with the release of each new technology node. FPGA architectures have evolved from an array of homogeneous *logic cells*, comprising a Look-up Table (LUT) and flipflop, to arrays of more sophisticated fracturable LUTs with hard blocks, including block RAM (BRAM), increasingly sophisticated multipliers and DSP blocks, networking components, phase-locked loops etc. FPGAs have also continued to grow in size; each new generation of devices provides an increasing number of logic cells, I/Os, BRAMs, DSP blocks, etc. as well as new/improved features (embedded processors, high speed transceivers, etc). The first FPGA, Xilinx's XC2064 had 1200 logic gates, 64 logic cells and 58 I/O pins [9]. Today's FPGAs can have upwards of 1.2 million logic cells, 1200 user I/O pins, 67 Mb of block RAM (BRAM), 3,600 DSP blocks, etc. [10].

Despite these advances, ASIC designs still outperform FPGA designs rather significantly in terms of area, delay and

dynamic power consumption. Specifically, research suggests that FPGAs designs are on average 35 times larger than ASIC designs, 3.5 times slower (4.6 times slower if using the lower end speed grade FPGAs), and that they consume on average 14 times more power [11]

### C. Previous Surveys

There has been significant previous work that surveys the underlying technology and software used to create FPGA designs [1, 2, 4, 5]. However, our focus is on how designs are able to use the evolving technology and to highlight trends in the design data itself. In order to highlight trends over different design data, product families or technology generations, a meaningful way to compare them is needed, e.g. equivalencies in FPGA logic usage over different generations of products with different architectures. In Kuon et al. [12], key components of FPGAs were surveyed and the effect of various design parameters on speed, critical path, area, etc. was studied. For example, they found that increasing LUT sizes up to 6-inputs (6-LUTs) significantly decreased the critical path, and that FPGAs with different LUT sizes were able to implement designs more efficiently than FPGAs with fixed LUT sizes. Similar tests were done in simulation using the VPR toolset looking at how varying LUT sizes affect area and delay [13]. Hard blocks have also been extensively studied, one general framework being the virtual embedded block approach which facilitates estimation of performance improvement using dummy blocks [14].

## III. METRICS AND EVALUATION CRITERIA

Typically, the quality of design implementations are evaluated based on quantitative metrics such as operating frequency, latency and resource usage. Power has also become an increasingly important metric as the complexity of designs increases in parallel with the number of programmable resources. Other metrics, such as the approximation of LUTs as a set number of transistors (to provide some baseline as to the overhead of using FPGAs to implement a design instead of ASICs) are not relevant to this study.

Due to the nature of applications typically implemented on FPGAs, e.g. streaming DSP applications and networking, certain designs (e.g. FFTs) have been the subject of numerous publications as researchers tried to find the best architectural mapping to the device [15]. Furthermore, vendors provide automated tools for generating circuits for commonly used Intellectual Property (IP) cores (FFT, FIR filters, etc). Such cores have been tuned to efficiently utilize improvements in device architecture (e.g. 4-LUTs to 6-LUTs/ALMs, the inclusion of embedded memory blocks and DSPs, etc.), making comparisons challenging. The relationship between design implementation and architectures becomes somewhat muddled as it is difficult to separate the effects of process technology, circuit design and CAD

algorithms. In fact, an increase in a design’s clock frequency could be due in part to improvements in all three factors.

#### A. Normalized Resource Usage

The challenge when comparing design resource usage across various device architectures is the significant changes in architecture, ranging from the inclusion of embedded blocks to the change in LUT architecture. The three metrics we consider are LUTs, BRAMs, and Multiplier DSP Blocks<sup>1</sup>. This is discussed in detail below.

1) *Look-up-Tables (LUTs)*: The FPGA’s main reconfigurable logic component, the LUT, has changed significantly from the earliest devices: from the initial standard 4-input LUT to 1) Xilinx’s 6-input fracturable LUTs, and 2) Altera’s 8-input Adaptive Logic Module (ALMs) used in modern FGPAs. The 6-LUT was introduced by Xilinx in 2006 with the release of their Virtex 5 products [16], whereas the 8-input ALM was introduced by Altera in 2005 with their Stratix II devices [17]. This presents the challenge of determining appropriate conversions when trying to compare design resource usage across technologies and between vendors. We use vendor recommended normalization values in this study: Xilinx’s 6-LUT translates to 1.6 4-LUTs [18]; and Altera’s ALM translates to 2.5 4-input logic elements [19].

We note that the vendors themselves have differing opinions as to how to normalize comparisons between their technology. Altera claims that their 8-input ALMs are equivalent to 1.8 Xilinx 6-LUTs, and Xilinx refutes this by saying one 8-input ALM is equivalent to 1.2 of their 6-LUTs [20] [21]. Given this dispute and the apparent acceptance of the 4-LUT equivalencies, we have normalized all of our results to 4-LUT technology and do not attempt to normalize ALMs to fracturable 6-LUTs or vice versa.

2) *BRAMs*: BRAM sizes have varied greatly over time from the early 4096-bit blocks [22] to the 36 Kbit blocks available on today’s devices [10]. Furthermore, some FPGAs offer flexible BRAM block sizes (including BRAMs that can be split into smaller, independent BRAMs) [10]. Additional features have also been included in later devices, e.g. error correction [23]. Since we consider only the amount of memory used by designs, we report the number of Kbits used in a design in lieu of the number/types of BRAMs.

3) *Multiplier/DSP Blocks*: Embedded multiplier blocks have changed dramatically, increasing in bit-width and functionality. They were first introduced to FPGA architecture in Xilinx’s Virtex II and Altera’s Stratix, which had 18x18 bit [24] and 36x36 bit multipliers (or 4 18x18 bit multipliers) [25], respectively. Both vendors evolved their multipliers into DSP blocks that incorporated additional functionality

(e.g. adders and accumulators [26], the ability to use a large multiplier to implement multiple independent smaller multipliers [27]).

Since Xilinx architectures have one 18×18 or one 25×18 multiplier per DSP block and Altera’s DSP blocks have 2 (Stratix V), 4 (Stratix and Stratix II), or 8 (Stratix III and Stratix IV) 18×18 multipliers per DSP block, we have scaled all of the Altera DSP block numbers with the appropriate factors to present a normalized comparison.

#### B. Operating Frequency

Operating frequency is directly reported in this work and we attempt to relate this to the feature size. Thus, the main questions are to determine if:

- 1) the operating frequency trends of research designs matches the trend in increased maximum operating frequency for devices,
- 2) the operating frequency trend of a given benchmark synthesized with the same CAD flow, assuming that the design’s functionality is mapped to the same types of resources (LUTs, memory, DSPs, etc.), matches what is theoretically expected from the feature size reduction alone or is more comparable to the maximum operating frequency trend for devices.

We also note that the critical path in an FPGA is usually due to routing rather than logic delay so for practical designs, this does not directly relate to transistor speed. In an FPGA, maximum achievable frequency is normally limited by the speed of the global clock buffers, which this does not scale with feature size. For research designs, the operating frequency achieved is normally limited by routing rather than logic delays. In both cases, frequency is generally not directly related to feature size.

## IV. METHODOLOGY AND RESULTS

In this section, we present our methodology and results for analyzing design performance and resource usage across various device families. Table I summarizes the initial release dates of select Altera and Xilinx FPGA families. For both vendors, year and feature size is included along with the model names for their largest devices. For each, we report the number of 4-LUTs/6-LUTs for Xilinx devices and the number of Logic Elements<sup>2</sup> (LEs)/ALMs for Altera devices, together with the number of DSP/Multiplier blocks, and Kbits of embedded BRAM<sup>3</sup>. Since the ratio of flipflops to LUTs is fixed for Xilinx, and Altera has similarly fixed this ratio for their LEs and ALMs, the number of flipflops are not reported.

<sup>1</sup>We recognize that flipflop utilization is an important resource usage metric for a design. However, flipflop architectures have remained relatively unchanged, except for the recent support for latch implementations in the latest FPGA generations. As such, flipflop usage does not require normalization across architectures, and simply trends with design complexity, so we do not consider it as a key metric in this analysis.

<sup>2</sup>Each Logic Element has one 4-LUT and is similar to Xilinx’s 4-LUT devices.

<sup>3</sup>Since this paper focuses on trends in reconfigurable resource usage, we have also excluded the embedded processors found on the Excalibur, Virtex II Pro, and the Virtex 4 & 5 FX devices

Table I: Select Altera and Xilinx devices with: feature size, embedded block data and release date

Year	Feature Size	Xilinx FPGA family	Device	LUTs	DSP/Mult blocks	BRAM Kbits	LUTs /DSP	LUTs /BRAM	Altera FPGA family	Device	ALMs (LEs)	DSP /Mult blocks	BRAM Kbits	LEs/ DSP	LEs/ BRAM	
2011	28 nm	Virtex 7	V	XC7V2000T	1,221,600	2,160	46,512	566	26							
			VX	XC7VX1140T	712,000	3,600	67,680	198	11							
			VH	XC7VH870T	547,600	2,520	50,760	217	11							
2010	28 nm								Stratix V	GT	5SGTC7	622,000	512	50,000	1,215	12
										GX	5SGXBB	952,000	704	52,000	1,352	18
										GS	5SGSD8	695,000	3,926	50,000	177	14
										E	5SEEB	952,000	704	52,000	1,352	18
2009	40 nm	Virtex 6	LX	XC6VLX760	474,240	864	25,920	549	18							
			SX	XC6V SX475T	297,600	2,016	38,304	148	8							
			HX	XC6VHX565T	354,240	864	32,832	410	11							
2008	40 nm								Stratix IV	GT	EP4S100G5	531,200	1,024	27,376	519	19
										GX	EP4SGX530	531,200	1,024	27,376	519	19
										E	EP4SE820	813,050	960	33,294	847	24
2006	65 nm	Virtex 5	LX	XC5VLX330	207,360	192	10,368	1,080	20							
			SX	XC5V SX240T	149,760	1,056	18,576	142	8							
			FX	XC5VFX200T	122,880	384	16,416	320	7							
2005	90 nm 130 nm								Stratix III	L	EP3SL340	337,500	576	16,272	586	21
										E	EP3SE260	255,000	768	14,688	332	17
										GX	EP2SGX130/G	132,540	252	6,747	526	20
2004	90 nm	Virtex 4	LX	XC4VLX200	178,176	96	6,048	1,856	29							
			SX	XC4V SX55	49,152	512	5,760	96	9							
			FX	XC4VFX140	126,336	192	9,936	658	13							
2002	130 nm								Stratix	GX	EP1SGX40D	41,250	56	3,423	737	12
											EP1S80	79,040	88	7,428	898	11
2001	130 nm 0.15 um	Virtex II	Pro	XC2VP100	88,192	444	7,992	199	11							
			Pro X	XC2VPX70	66,176	308	5,544	215	12							
			V	XC2V8000	93,184	168	3,024	555	31							
2000	0.18 um								Mercury		EP1M350	14,400	0	115	-	125
										Excalibur		EPXA10	38,400	0	3,146	-
1999	0.18 um	Virtex E		XCV3200E	64,896	0	851	-	76							
1998	0.22 um	Virtex		XC1000	24,576	0	131	-	188							
			4000 E/XL		XC4085XL	12,544	0	0	-	-						
1996	0.3 um								Flex 10KA	EPF10K250A	12,160	0	41	-	297	
1995	0.42 um								Flex 10K	EPF10K100	4,992	0	25	-	200	
1992	0.6 um								Flex 8000	EPF81500A	1,296	0	0	-	-	
1991	0.8 um	4000 series		XC4025	2,048	0	0	-	-							
1985	2 um	2000 series		XC2018	400	0	0	-	-							

The ratios of 1) LUTs to Kbits of BRAM, and 2) LUTs to DSP/Multiplier blocks are also included for both vendor's devices. Lower values indicate more DSP blocks/Kbits of BRAM memory relative to the devices reconfigurable logic (LUTs). Although the 4000 series did not provide embedded BRAMs, users could configure the logic cells in this series as distributed RAM. Altera and Xilinx introduced embedded BRAM blocks in the 10K series and Virtex series, respectively. Both vendors increased the ratio of embedded memory in their parts until the 130nm fabricated devices (Stratix and Virtex II Pro). Since then, the ratio has remained relatively constant (7-29 LUTs/BRAM Kbits), although varied, over the different functional families within the device family (e.g. LX, SX, FX, GX, GT, etc.). Conversely, Xilinx's ratio of multipliers/DSP blocks to logic has increased slightly (except in LX devices, which have seen a notable increase), whereas Altera's ratio has fluctuated within a relatively fixed range and decreased somewhat for the Stratix V. This is likely because the changes in the bit width (18x18 to 25x18 for Xilinx and 36x36 to 27x27 for Altera) and functionality of these blocks (e.g. simple multipliers to DSP blocks, the number of independent 18x18 multipliers that can be implemented within a single DSP block) has allowed designers to encapsulate more functionality in a single block.

Obviously, both the absolute numbers of DSP blocks and BRAMs are likely to increase as device size increases. However, the ratios are likely to continue to fluctuate as

both vendors respond to customer needs. This has also lead to multiple families for each device generation: 1) those that have a larger ratio of DSP blocks and embedded memory (e.g. for DSP streaming applications), 2) those with a decreased ratio of DSP to logic but a similarly larger ratio of embedded memory to logic, and 3) devices that are primarily logic with lesser numbers of DSP blocks and embedded memory.

#### A. Moore's Law and Dennard's Law

Figure 1 plots the reciprocal of feature size squared ( $1/\lambda^2$ ) (log y-axis) versus the year it was released (linear x-axis) in green crosses. This graph is generated using the data from Table I. Superimposed on the same figure is a linear regression fit of the exponential growth equation  $y = 2^{k+ax}$ . This results in a doubling of  $y$  every  $1/a$  years, and for transistor area, our fit yields  $1/a = 2.04$  years, which is consistent with Moore's Law.

A similar regression of  $1/\lambda$  vs year gives  $y = 2^{(500.39004+0.24652x)}$ . According to Dennard's Law, this suggests that technology would contribute a halving of transistor delay every 4 years.

#### B. Research Designs

In this section, we report the methodology and results for our analysis of the sample set of research designs from the past 20 years of IEEE's FCCM publications.

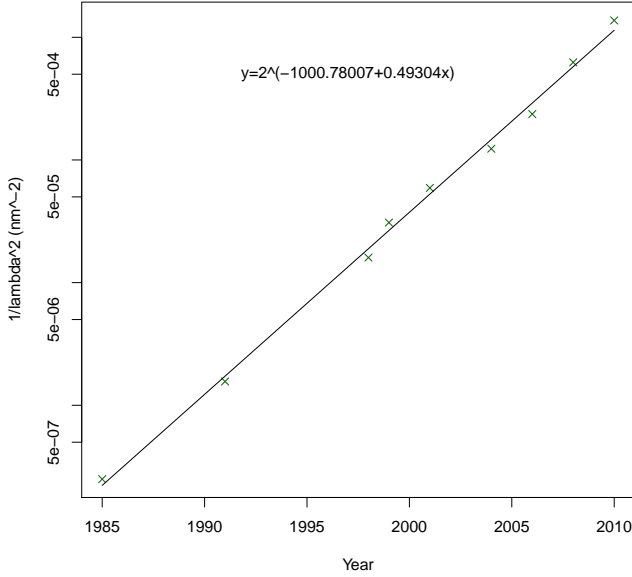


Figure 1: Plot of  $1/\lambda^2$  vs year and associated regression fit.

1) *Methodology*: As mentioned in Section III, we wish to analyze the resource usage (i.e. LUTs, DSPs, and embedded memory usage) and operating frequency trends exhibited by research designs. Our sample set of research designs comprises all application papers using SRAM-based FPGAs that were published in every *odd* year of the IEEE’s FCCM publications from 1995-2013, to reflect the expected changes in device fabrication from Moore’s Law (i.e. every two years). For each of the metrics we wish to evaluate, we plot the metric data on a log scale versus time. We then apply curve fitting for the data from our sample set of research designs. We also plot the theoretical FPGA technology maximums (obtained from vendor data sheets) for each metric on the same graph and curve fit this data as well to see how it trends relative to Moore’s law.

2) *Results and Analysis*: Improvements in FPGA technology can be linked to Moore’s Law, architectural and circuit improvements. The top green crosses in Figure 2 show how the maximum number of logic cells in the largest Xilinx Virtex FPGA has evolved. The top green crosses are used similarly in Figures 3-5; they show: the maximum clock frequency vs year in Figure 3; the maximum amount of RAM vs year in Figure 4; and the maximum number of DSP blocks vs year in Figure 5. For each figure, the top black lines depict linear regression fits of the data to the base 2 logarithm of the y-axis, with the resulting equations being shown in black text. Regression coefficients are expressed to 5 decimal places as the exponentiation in the  $2^{(k+ax)}$  expression of the model fit makes the output very sensitive to perturbations of the input.

*LUTs*: Figure 2 shows a scatter plot of the number of normalized LUTs (as explained in Section III-A1) vs year for all the designs studied. All research design data points

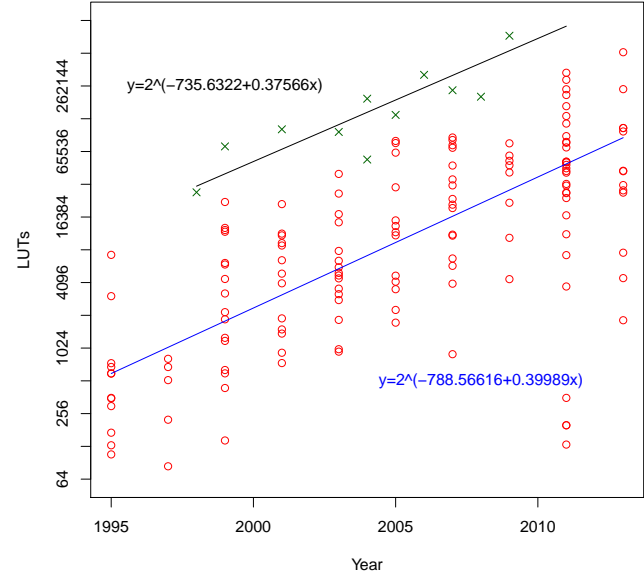


Figure 2: Scatterplot and regression fit showing maximum number of normalized 4-LUTs in the largest FPGA and normalized 4-LUTs used in research designs over time.

are shown as red circles, with a linear regression fit to the median value and corresponding equation shown in blue. A linear trend in the plot with logarithmic y-axis is apparent, corresponding to an exponentially increasing relationship with year. As can be seen in the graph, the slope of the fit is also consistent with the maximum values.

From Figure 2, it can be seen that the maximum number of lookup tables available in an FPGA and the number of lookup tables in the surveyed research designs increase at a similar rate ( $a=0.38$  vs  $0.40$ ). For  $a = 0.38$ , this corresponds to a doubling roughly every 2.6 years, higher than the value of 2.0 expected from Moore’s Law. We believe that this is because FPGA clock trees and hard blocks account for a significant proportion of the total silicon compared with LUTs, but these are not considered in our accounting.

*Operating Frequency*: Figure 3 shows a scatter plot of research design operating frequencies vs year, which obviously should increase with time. A doubling of research design frequencies occurs every 5 years (blue regression line), and this is contrasted with a doubling in maximum FPGA frequency every 8 years (black regression line). We believe one of the factors is the inclusion of hard blocks in the FPGA that allows increased operating frequency and reduced critical paths through lookup tables. Another factor is that the introduction of fracturable LUTs with more inputs can potentially reduce the number of LUTs between registers, thus increasing frequency. As discussed in Section III-B, these values are not directly related to improvements in delay expected by Dennard’s Law, from which we expect a halving in delay every 4 years.

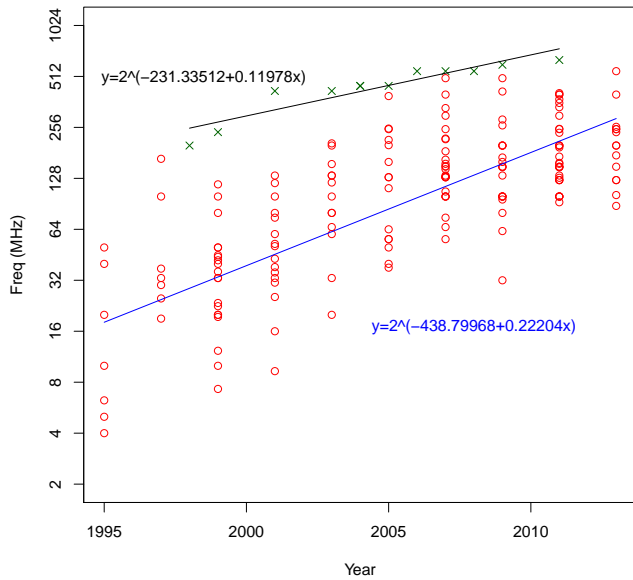


Figure 3: Scatterplot and regression fit showing maximum frequency in largest FPGA and frequency used in research designs over time.

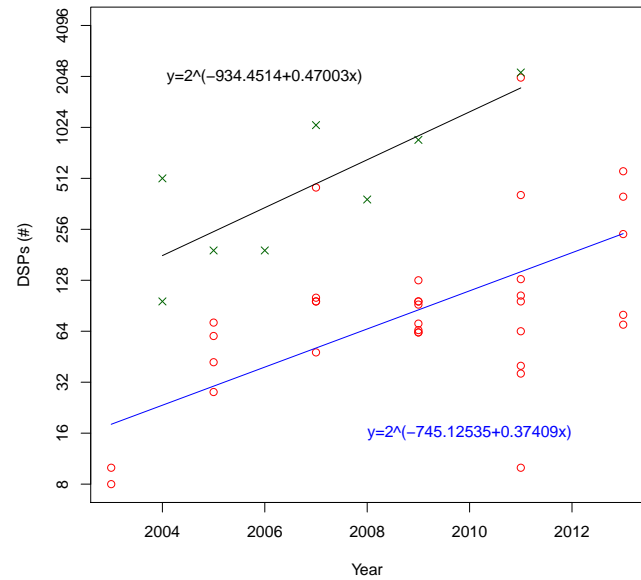


Figure 5: Scatterplot and regression fit showing maximum DSPs in largest FPGA and number of DSPs used in research designs over time.

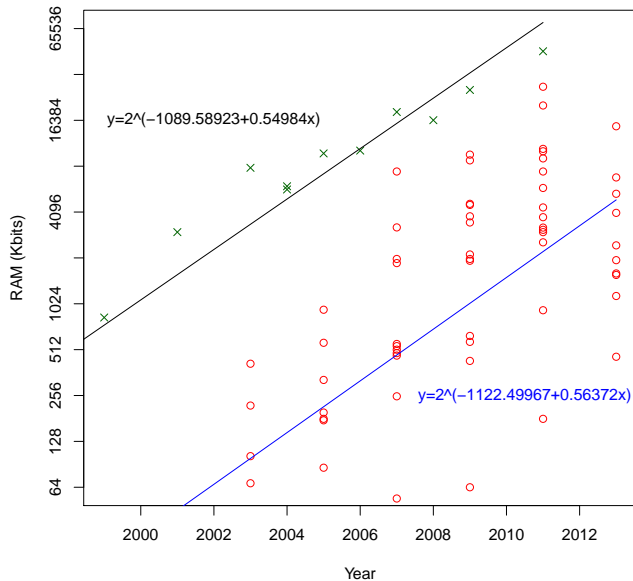


Figure 4: Scatterplot and regression fit showing maximum Kbits of BRAM in the largest FPGAs and the Kbits of BRAM used in research designs over time.

**BRAM:** Figure 4 shows a scatter plot of research design RAM usage vs year<sup>4</sup>. Comparing the slopes of the two regression lines, it can be seen that the usage of BRAM in research designs has tracked improvements in technology, with each doubling approximately every two years.

**DSPs:** Figure 5 shows the maximum number of DSP blocks available in the largest FPGA vs year. The green

crosses show a clear trend of this resource doubling every 2.12 years, consistent with that expected from Moore’s Law. The bottom blue regression has a smaller slope, indicating that designs which employ DSP blocks often use a smaller percentage of those available. This is due to: 1) small chips being used, 2) other resources such as LUTs or RAM being the limiting resource for a design, and/or 3) that multiplication only forms a subset of the design. The slope of the maximum values of red circles in the scatter plot is similar to that of the green crosses, meaning that there are always some research designs that utilize all DSP resources available on the largest devices.

### C. Benchmark designs

In this section, we report the methodology and results for our analysis of the benchmark designs generated using various vendor IP cores.

1) *Methodology:* For this set of experiments, we wanted to use the same CAD flow to map the same IP cores to various generations of FPGAs. We were able to use the same version of one vendor’s CAD flow to map to four generations of devices<sup>5</sup>. These device families were introduced between 2004 and 2011 and had minimum feature sizes of 90nm, 65nm, 40nm, and 28nm, respectively. In an attempt to minimize the size and resource variation between device generations, we used the largest device from the oldest device family (fabricated with 90nm technology) and then

<sup>4</sup>FPGAs with embedded memories were not available from Altera and Xilinx before 1995 and 1998, respectively.

<sup>5</sup>As we do not wish to benchmark the vendor’s IP cores, CAD flow, or devices, their name has been anonymized for this discussion.

Table II: Benchmark Details

Benchmarks	Data word width	Norm. LUTs	Flipflops	DSPs (DSP48E1)	BRAM (KBits)
FFT (1k Fixed Point)	16	2,362	2,066	9	126
FPU	64	4,900	5,950	0	0
Square Root	48	3,740	2,530	0	0
ArcTanh	48	10,325	7,250	0	0
Trimode Ethernet	32	2,348	1,352	0	0
FFT Large (Dual Channel, 32k Fixed Point)	34	12,405	10,026	75	5,382

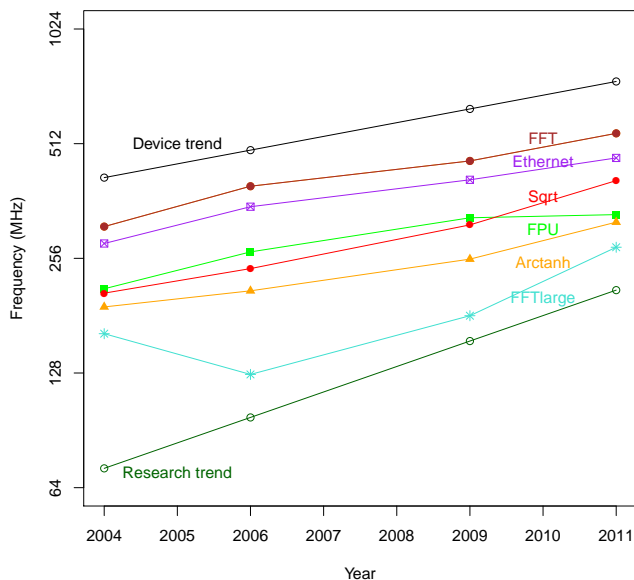


Figure 6: Maximum frequency of benchmarks.

selected devices from each of the newer families that had a comparable number of logic resources.

We then found a series of IP cores that the CAD flow would allow us to map to all of these devices and evaluated our benchmark designs using the same metrics as the research designs. Table II summarizes the benchmarks and data width used. They include a range of mathematical functions, two FFTs of varied size, and a Trimode Ethernet core. Since our design parameters remained constant for all four generations of devices, Table II also lists the average number of normalized LUTs, flipflops and DSP blocks used (which varied), as well as the amount of BRAM used (which remained constant).

2) *Results and Analysis:* In this section, we discuss the resource usage and operating frequency trends for our benchmark designs.

*Resource Usage:* Unlike the BRAM usage, which remained constant for all of our benchmarks, the normalized LUT usage varied noticeably and the flipflop and DSP block usage varied slightly from device generation to generation. This is likely due to factors such as: 1) how the core was mapped to the fracturable LUTs (packing of smaller LUTs into the larger fracturable LUTs was achieved with varying efficiency); 2) changes in the underlying architecture of the embedded blocks (e.g. DSP blocks) changing what could

be implemented using them; 3) how many LUTs were used to route signals (instead of mapping logic) which tends to improve operating frequency.

*Operating Frequency:* Figure 6 maps the operating frequency of the benchmark designs on a log graph as a set of connected piece-wise linear plots over time. This figure includes the maximum device frequency trend line (top solid black line) and the research design frequency trend line (bottom solid dark green line) from Figure 3 for reference. It is apparent that all of the designs’ operating frequencies are higher than the research trend line, implying that they are mostly faster than the research designs. This is expected as all of these IP cores were provided by the vendor and are highly optimized for at least one generation of their devices. Additionally, all of the designs’ operating frequencies increase almost linearly (excluding the FFT large design) and trend quite closely with that of the device trend line. The decrease in frequency for the FFT large design in 2006 is likely due to the relatively large number of BRAM and DSP block resources required. Although embedded blocks improve the overall operating frequency of a design, routing between these blocks and logic/other embedded blocks can often be circuitous, causing significant routing delays.

## V. CONCLUSION AND FUTURE WORK

A study of quantitative information from designs presented over the last 20 years of the FCCM conference was undertaken. From this work, we conclude that: 1) FPGA feature size has been closely following Moore’s Law; 2) the number of lookup tables for designs and devices are closely related and doubling every 2.5 years; 3) operating frequency of research designs and FPGA maximum operating frequency are increasing at different rates (a doubling every 8 and 5 years respectively), this rate being slower than that offered by technology scaling; 4) memory utilization of designs and devices are doubling every 1.8 years; 5) the number of DSPs in FPGAs is increasing at a faster rate than designs (doubling every 2 years vs 3 years); 6) the abovementioned trends can be modeled using the equations introduced in the paper. We further suggest that research in reconfigurable computing is able to exploit new FPGA features as they are introduced, and that the future trajectory of research designs will follow our models.

By analyzing our benchmark designs, we noted that the fixed scaling factors provided by vendors provide an approximation for LUT usage comparisons between device generations with different architectures. However, since the designs mapped to 4-LUT architectures obviously cannot always be packed into 6-LUT/ALM architectures efficiently and since some 6-LUTs/ALMs may be used to route signals on different devices, these scaling numbers are approximate. However, our results from comparing operating frequency trends of our benchmark designs against maximum device



operating frequency suggest that it may be possible to determine normalization factors for comparisons across device generations with further study. Obviously, factors such as the portion of the chip resources used by a design will also be a factor.

We hypothesized that the number of lookup tables did not match Moore's Law and that maximum frequency of designs did not track technology due to the inclusion of hard blocks, which require chip area and can significantly impact a design's operating frequency. This issue requires further study. Moreover, since FPGA architectures are interconnect-dominated, it would be interesting to explore trends associated with Rent coefficients [29]. Finally, many other interesting designs, parameters (particularly power) and hard blocks can be studied in a similar fashion. We believe that instead of looking at how single designs utilize architectural features, studies of how groups of designs do so can enable new information to be gained regarding the converse, i.e. how to design FPGA architectures better suited for applications.

#### REFERENCES

- [1] K. Pocek, R. Tessier, and A. DeHon, "Birth and adolescence of reconfigurable computing: A survey of the first 20 years of Field-programmable Custom Computing Machines," in *Highlights of the First Twenty Years of the IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2013, pp. 3–19.
- [2] S. Trimberger, "Three ages of FPGAs: A retrospective on the first 25 years of FPGA technology," in *FPGA 20: Highlights of the International Symposium on Field-Programmable Gate Arrays*, 2012.
- [3] G. Estrin and C. R. Viswanathan, "Organization of a "fixed-plus-variable" structure computer for computation of eigenvalues and eigenvectors of real symmetric matrices," *J. ACM*, vol. 9, no. 1, pp. 41–60, Jan. 1962. [Online]. Available: <http://doi.acm.org/10.1145/321105.321110>
- [4] K. Compton and S. Hauck, "Reconfigurable computing: A survey of systems and software," *ACM Comput. Surv.*, vol. 34, no. 2, pp. 171–210, Jun. 2002. [Online]. Available: <http://doi.acm.org/10.1145/508352.508353>
- [5] T. J. Todman, G. A. Constantinides, S. J. Wilton, O. Mencer, W. Luk, and P. Y. Cheung, "Reconfigurable computing: architectures and design methods," *IEE Proceedings: Computers and Digital Techniques*, vol. 152, no. 2, pp. 193–207, 2005.
- [6] (2014, Mar.) International technology roadmap for semiconductors. [Online]. Available: <http://www.itrs.net/>
- [7] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, April 1965.
- [8] R. Dennard, F. Gaensslen, V. Rideout, E. Bassous, and A. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *Solid-State Circuits, IEEE Journal of*, vol. 9, no. 5, pp. 256–268, Oct 1974.
- [9] Xilinx, "XC2064/XC2018 logic cell array product specification."
- [10] —, "7 series FPGAs overview," Oct 2014. [Online]. Available: [www.xilinx.com](http://www.xilinx.com)
- [11] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 203–215, Feb 2007.
- [12] I. Kuon, R. Tessier, and J. Rose, "FPGA architecture: Survey and challenges," *Found. Trends Electron. Des. Autom.*, vol. 2, no. 2, pp. 135–253, Feb. 2008. [Online]. Available: <http://dx.doi.org/10.1561/1000000005>
- [13] J. Luu, I. Kuon, P. Jamieson, T. Campbell, A. Ye, W. M. Fang, and J. Rose, "VPR 5.0: FPGA CAD and architecture exploration tools with single-driver routing, heterogeneity and process scaling," 2009.
- [14] C. Ho, P. Leong, W. Luk, S. Wilton, and S. Lopez-Buedo, "Virtual embedded blocks: A methodology for evaluating embedded elements in FPGAs," in *Field-Programmable Custom Computing Machines, 2006. FCCM '06. 14th Annual IEEE Symposium on*, April 2006, pp. 35–44.
- [15] S. Sukhsawas and K. Benkrid, "A high-level implementation of a high performance pipeline FFT on Virtex-E FPGAs," in *VLSI, 2004. Proceedings. IEEE Computer Society Annual Symposium on*, Feb 2004, pp. 229–232.
- [16] Xilinx, "Virtex 5 family overview," Feb 2009. [Online]. Available: [www.xilinx.com](http://www.xilinx.com)
- [17] Altera, "Introduction," *Stratix II Device Handbook*, vol. 1, May 2007. [Online]. Available: [www.altera.com](http://www.altera.com)
- [18] C. Davies and J. Bateman, "Using Virtex-5 FPGAs in cots board-level products," *Xcell Journal*, pp. 90–95, 2006.
- [19] Altera, "Stratix II performance and logic efficiency analysis," *White Paper*, Sept 2006. [Online]. Available: [www.altera.com](http://www.altera.com)
- [20] —, "FPGA architecture," *White Paper*, July 2006. [Online]. Available: [www.altera.com](http://www.altera.com)
- [21] Xilinx, "Advantages of the Virtex-5 FPGA 6-input LUT architecture," *White Paper*, Dec 2007. [Online]. Available: [www.xilinx.com](http://www.xilinx.com)
- [22] —, "Virtex E 1.8 v field programmable gate array," March 2014. [Online]. Available: [www.xilinx.com](http://www.xilinx.com)
- [23] Altera, "Arria 10 device overview," Sept 2014. [Online]. Available: [www.altera.com](http://www.altera.com)
- [24] Xilinx, "Virtex II platform FPGAs: Complete datasheet," April 2014. [Online]. Available: [www.xilinx.com](http://www.xilinx.com)
- [25] Altera, "Dsp blocks in Stratix and Stratix GX devices," *Stratix Device Handbook*, vol. 2, July 2005. [Online]. Available: [www.altera.com](http://www.altera.com)
- [26] Xilinx, "Virtex IV family overview," August 2010. [Online]. Available: [www.xilinx.com](http://www.xilinx.com)
- [27] Altera, "Dsp blocks in Stratix III devices," *Stratix III Device Handbook*, vol. 1, March 2010. [Online]. Available: [www.altera.com](http://www.altera.com)
- [28] P. Jamieson and J. Rose, "Mapping multiplexers onto hard multipliers in FPGAs," in *IEEE-NEWCAS Conference, 2005. The 3rd International*, June 2005, pp. 323–326.
- [29] P. Christie and D. Stroobandt, "The interpretation and application of Rent's rule," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 639–648, Dec 2000.