

Architecting Processing System Applications Past Moore's Law

Jeremy W. Horner
Northrop Grumman Corporation
Linthicum, MD, U.S.A.
Jeremy.Horner@ngc.com

Eliot Glaser
Northrop Grumman Corporation
Linthicum, MD, U.S.A.
Eliot.Glaser@ngc.com

John Holland
Northrop Grumman Corporation
Linthicum, MD, U.S.A.
John.Holland@ngc.com

Gary Petrosky
Northrop Grumman Corporation
Linthicum, MD, U.S.A.
Gary.Petrosky@ngc.com

Abstract— The performance of General Purpose Computing applications has followed Moore's Law [1, 2] for decades, but there is a class of high-performance computing applications where the influence of Moore's Law is more subtle and more complex. In these applications, Moore's Law primarily impacts the performance of the devices, but the application can span multiple devices. So, the overall performance is influenced not only by Moore's Law, but it also involves the interaction of the devices and how well the application maps to the devices.

Keywords—Front end processing, general purpose computing, signal processing, embedded computing, throughput, data bandwidth, multi-rate processing, digital beamforming, analog to digital converters

I. INTRODUCTION

High Performance Embedded Computer Applications such as digital front end processors implement highly complex algorithms with strict real-time scheduling requirements, and the stakes are high. They must perform with a high degree of reliability under extreme circumstances because of critical continued operation. The aggregate performance requirements for these applications far exceed what one might expect as compared to typical general purpose computing applications. The processor must achieve the required results while meeting challenging form factors because these high performance embedded computing systems often are used on size, weight and power (SWaP)-challenged systems such as airborne and space platforms.

This paper discusses factors, other than device density, that influence high-performance embedded front end processing. Section II of this paper describes the characteristics of high performance front end processing. Section III discusses the effect of Moore's Law on high-performance applications. Section IV provides examples of algorithm implementation options that decrease the impact of device density and performance. Section V discusses a process of developing signal processing architectures within these trade spaces.

Section VI notes the challenges of data bandwidth between processing nodes.

II. CHARACTERISTICS OF HIGH PERFORMANCE COMPUTING APPLICATIONS THAT RESIST THE LIMITATIONS OF MOORE'S LAW

This class of high-performance computing applications is defined by a number of distinct characteristics including:

- Data-intensive processing
- Emphasis on throughput requirement and latency requirements
- Flexible architecture that is amenable to high degree of parallelism
- Suitability of algorithms to fixed-point implementation

Data intensive processing requires real-time processing of wide data bandwidths. With commercially available analog-to-digital converters (ADC) at 16-bits of resolution for high signal-to-noise ratio applications and operating at 1 Giga-sample per second (Gsps) or greater sample rates, a front end processor must be able to handle at least 16 Gigabits per second (Gbps) of data bandwidth per data channel. This per-channel performance emphasizes throughput and demands minimal latencies for real-time processing. These high sample rates (and thus high data processing clock rates) often dictate parallel multi-rate processing architectures rather than the iterative loops found in general purpose processing and need processing architectures that support signal and data processing across parallel paths at high clock rates. Data throughput and latency also can drive the choice of fixed-point processing in order to achieve results in fewer clocks and avoid multi-clock loops and scaling. In these ways front end signal processing presents a different requirement set than general purpose processing.

As an example, consider the digital beamformer shown in Figure 1. Digital beamforming forms one or more beams from

sub-apertures of an antenna. In the figure the incoming wave front is received by each of the channels of the antenna, down-converted and A/D sampled. Digital beamforming uses time delays and/or frequency-specific phase shifts to steer the input channels so that each senses the same frequency and phase of the wave front. Key parameters for digital beamforming architectures are: input channels and sample rates, output beams, data bandwidth and pre-beamforming tuning and/or filtering requirements [3].

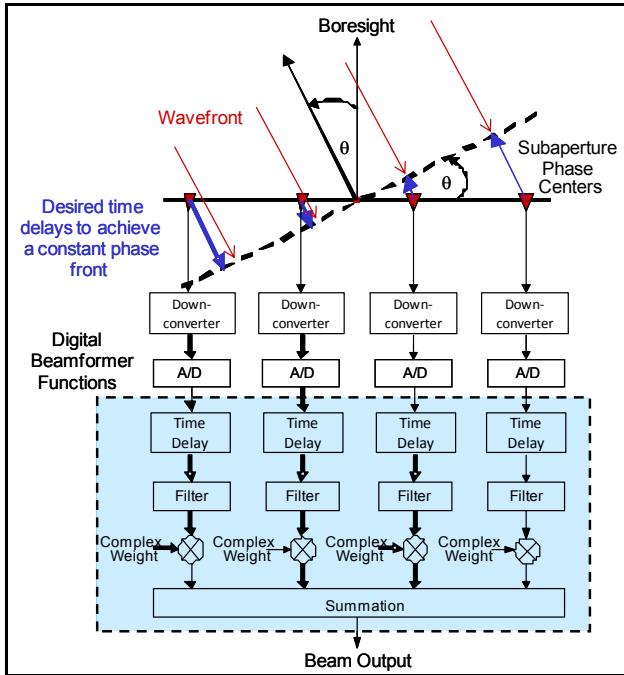


Fig. 1. Notional digital beamforming front end processing architecture

Figure 2 shows an example implementation of the digital beamformer. Using the example sample rates above, each ADC channel produces 16 Gbps of data. If the system is 16 channels, then total data bandwidth is 256 Gbps. The front end processing requires high data throughput and a high degree of predictability in data arrival at the beamforming stage in order to avoid large memories for data alignment.

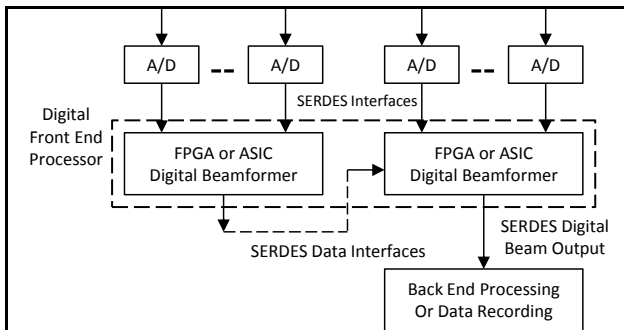


Fig. 2. Example implementation of digital beamforming front end processing architecture

Predictable and consistent throughput performance (for both data and operations) must be maintained across all of the channels through any required channel processing before the

beamformer. This per-channel processing could include large polyphase filter banks to reduce beamforming bandwidth. The output beams represent another parallel data stream, which may require additional processing (pulse compression, etc.) before transmission to the back-end system.

III. EFFECT OF MOORE'S LAW ON THESE HIGH PERFORMANCE APPLICATIONS

The most significant ways that Moore's Law impacts the high-performance computing application class are:

- The efficiency of each processing node in the system
- The ever-increasing amount of sensor data that needs to be processed
- The need for faster data interconnects to support the required sensor and processing rates, leading to multi-gigabit serial interfaces

The most visible effect of Moore's Law has been the exponential increase in memory density over the past 40 years. However, expanded memory capacity has little benefit to such high performance applications; the high processing rates and low latency requirements do not allow for storage and retrieval of any appreciable amount of data. Buffering intermediate results is needed for alignment and synchronization across the processor, but these memories are small and distributed amongst the processing nodes.

Moore's Law has had a profound effect on processor node efficiency. As processor nodes have gotten smaller and faster, the efficiency has increased – more operations done faster and with wider data widths. This has served as an enabler for increasingly complex processing and larger system front ends. Can the analog domain keep up?

While the digital domain has gotten all of the attention from Moore's Law, the Analog to Digital Converter (ADC) has hardly stood still. Smaller scaling of ADC device feature size is a partial contributor [5], but other factors responsible for much of the ADC performance increase include advanced converter topologies, lower operating voltages, reduced noise, improved linearity, and reduced clock jitter. Studies use several different "Figures of Merit" (FoM) to evaluate the increased performance of ADCs, due to the differences in converter types and associated performance specifications. FoMs evaluate conversion speed, resolution, power dissipation, effective number of bits (ENOB), dynamic range, bandwidth, and signal to noise (SN) ratio. Depending on which FoM is employed, ADC performance over the past 20 years has doubled between every 3 years to every 1.8 years [6]. It would seem that Moore's Law does extend to the ADC domain.

Processing density increased to the point where traditional parallel device interconnects were not able to keep up with the data transport requirements, necessitating the migration to high speed serial interconnects between processing nodes. Holt showed [7] that the device interconnect bandwidths have doubled about every four years. The result is that the multi-gigabit (MGT) interfaces needed to transport data are now critical features of high speed processing devices. ADCs are not immune from this development; the JESD204 standard was

developed to provide MGT serial interface between ADCs and processing devices.

IV. THE INSULATING BARRIER BETWEEN MOORE’S LAW AND SYSTEM PERFORMANCE

The aggregate system performance required (Operations / Second) is based on the amount of data to be processed and the specific algorithms that must process the data. The ability for the algorithm to heavily influence the performance requirement coupled with the flexible architecture and the algorithm’s amenability to parallelism translates into a reasonably significant insulating barrier between Moore’s Law and the system performance especially when compared to the influence of Moore’s Law to general purpose computing.

Processing architecture options are many and varied. Example implementation options for front end processing algorithms include:

- Algorithm implementation at the ADC sample rate
- Multi-rate processing techniques to distribute processing over multiple parallel channels
- Signal processing solutions such as dividing the channel signal bandwidth into subbands to be processed at lower subband sample rates.

Algorithm implementation at high ADC sample rates potentially reduces device resources or logic, but also can increase registers due to the pipelining required for processing at high clock rates.

Multi-rate processing distributes algorithm execution over multiple parallel processing channels. The amount of logic increases with each parallel processing channel (for example, double with a 2nd channel, triples with a 3rd), but this technique results in a lower overall clock rate, fewer registers for pipelining, and smaller clock distribution trees.

Channel signal subbanding uses digital signal processing such as a polyphase filter bank to divide the channel signal spectrum into subbands. Each subband can move through a separate processing chain, often independently, allowing partitioning and distribution of processing resources.

Note that these signal processing implementation choices do not result in a lower total processing throughput or lower total data bandwidth. The processing throughput and data bandwidth requirements for each implementation match the system requirement. These implementation choices provide a signal processing trade space for total logic resources, processing clock rates, processing pipeline structures, numbers of registers, clock trees levels and loading, levels of logic between registers, independent or dependent resource distribution and other factors. This robust trade space provides insulation between Moore’s Law and system performance – successful implementation of a signal processing algorithm that meets system performance does not completely rely on the number of or density of transistors on a device.

V. ARCHITECTING THROUGH THE LIMITATIONS OF MOORE’S LAW AND THE COUNTING THE COST

The efficiency gains provided by Moore’s Law to each processing node in the system are important, but their influence is more subtle than is typically experienced by general-purpose computing. For these applications, the efficiency gains will primarily drive down the size, weight, and power (SWaP) requirements/characteristics of the system. This means that the raw aggregate performance limitations imposed by Moore’s Law on the application are only significant if the required SWaP efficiency also becomes a limiting factor on the aggregate performance.

When implementing a processing system belonging to this class of high performance computing applications, there are techniques that can be applied which provide flexibility in mapping the algorithm to available hardware.

A previous paper [4] discusses a process, shown in Figure 3, that can produce multiple candidate architectures for partitioning an algorithm into available technologies while considering the strengths and weaknesses of each technology and the algorithm’s requirements.

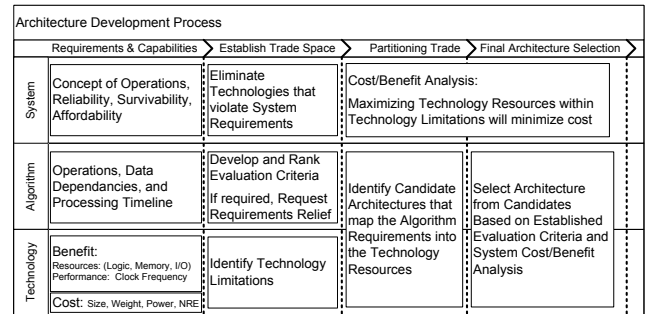


Figure 3: Architecture development process

The following example will use this process to develop candidate architectures for a Digital Beamforming Application in order to illustrate that the raw aggregate performance limitations of a single processing node are not the limiting factor in implementing this high performance computing application.

- *System Requirements:* Process 18 channels of sub-banded data to form 4 output beams.
- *Algorithm Requirements:* 138 Giga-Operations/sec
- *Algorithm Data-Dependencies:* None exists between subbands, but a significant data dependency between output beams.
- *Candidate Architectures:* The optimal architecture minimizes the number of devices used to process the output beams for a given subband and increases the number of subbands processed in a given device until one of the resources of that is consumed.

We will now follow the architecture development process using the above information for two different candidate FPGAs. For this example, the first FPGA candidate, “FPGA X”, will contain one half of the resources of the second FPGA candidate, “FPGA Y”.

The architecture in Figure 4 shows the result of the partitioning trade-off when using candidate “FPGA X”. It minimizes the number of FPGAs required by maximizing the resource utilization of each FPGA within the I/O constraints of the FPGA technology. All output beam processing for a given subband is performed in the same FPGA – giving priority to the data dependency between beams. The amount of resources available in candidate “FPGA X”, limit its processing ability to 34.5 Giga-Operations/sec. This candidate architecture, therefore, will require 4 FPGAs.

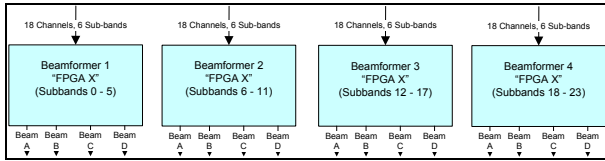


Figure 4: Block diagram of candidate architecture using “FPGA X”

The architecture in Figure 5 shows the result of the partitioning trade-off when using candidate “FPGA Y”. It also minimizes the number of FPGAs required by maximizing the resource utilization of each FPGA within the I/O constraints of the FPGA technology. All output beam processing for a given subband is performed in the same FPGA – giving priority to the data dependency between beams. The amount of resources available in candidate “FPGA Y”, limit its processing ability to 69 Giga-Operations/sec. This candidate architecture, therefore, will require 2 FPGAs.

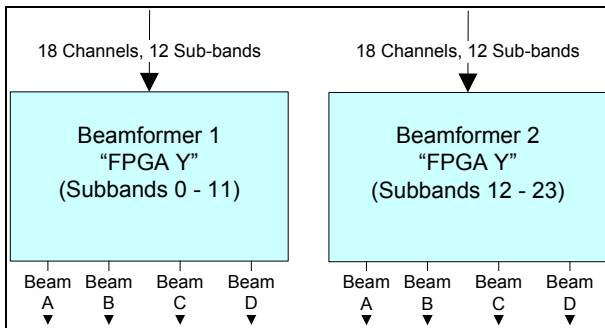


Figure 5: Block diagram of candidate architecture using “FPGA Y”

This example shows that although the performance of “FPGA Y” is double that of “FPGA X”, both technologies produced a candidate architecture that fully satisfied the system and algorithm requirements of high performance application. The only significant difference is that the use of “FPGA Y” resulted in the use of fewer components which would translate into a lower SWaP than could have been achieved when using “FPGA X”.

As long as the overall system requirements support the resulting SWaP for both candidate architectures, the performance gains achieved with “Moore’s Law” did not influence the ability to implement the required DBF application.

Moore’s Law will improve the performance of a single node in the processing system, but since the requirements can be spread across multiple nodes, the performance

improvements only cause the result to look different. It does not change the feasibility of implementing the capability. In this case, it would likely reduce the number of nodes in the processing system by increasing the processing rate in the multi-rate system. This will in turn drive down the size, weight, and power characteristics of the system.

The result is that a processing system implementation using newer processing nodes can meet more aggressive customer SWaP requirements when compared with implementations using older hardware, or the implementation can achieve the legacy SWaP requirements with lower risk.

Because architectural and algorithmic trade-offs can be applied to enable the use of a wide range of hardware devices, the relative importance of each device’s performance is de-emphasized, as long as the system requirements are met. This SWaP flexibility allows for performance to improve over time while maintaining the same SWaP, or it allows for the SWaP to be reduced over time while maintaining the same performance.

It should be noted, however, that some severely SWaP-challenged platforms do not have this same SWaP flexibility due to hard limits which bound the SWaP constraints that can be flowed down to sensors and applications. These hard limits act like a cliff function that excludes the deployment of certain applications because no practical implementation is viable until single-node performance achieves a critical level of capability. Reaching this critical level of capability, through the advancement that comes with Moore’s Law, will then enable applications that could not previously be considered.

VI. THE SHAPING INFLUENCE OF DATA MOBILITY IN SYSTEM DESIGN

Perhaps, the most significant influence to the system design is the ability to move data throughout the system as needed. It is possible, to a large degree, to add processing nodes to a system to meet performance requirements, but the limitations on data movement between the processing nodes can quickly undermine the value of each additional processing node.

Moving the increasing amount of data between processing nodes has continually presented challenges of increasing complexity. Whether communicating chip-to-chip, across a backplane, or through copper or fiber optic cables, reliability to meet the environmental conditions, added to the signal integrity requirements for high speed serial data transfers, offers the designers some unique challenges.

Available FPGAs and ASICs today provide large quantities of multi-gigabit serial data paths reaching speeds of 28 Gbits per second and above. Some new techniques embed two or more bits per symbol at these rates, to add to the complexity and importance of maintaining signal integrity. Designing transmission media to successfully pass this data by preserving its signal integrity requires an understanding of the environment, distance of travel, data rates, and materials involved.

For chip-to-chip and across backplane transmissions, important factors to manage are signal strength and integrity, line losses, noise immunity, reflections or ringing, and media

mode conversions. There are many techniques and guidelines to follow when considering each of these factors. For instance, signal integrity can be improved over longer lengths by taking advantage of pre-emphasis and equalization if available. Line losses can be improved by using better materials, increased and smoother copper surfaces [8], and lower and matched impedances. Some techniques may not be as realistic as others and are dependent on the architecture and environment in which the application is used. All these things should be considered when developing the optimum design.

For chassis to chassis communications, typically the preferred method has been a cable approach. Ruggedized cables and connectors are prevalent throughout industry today. Pushing the limit on data rates always seems to be an ongoing battle. Impedance matching over copper and through ruggedized connectors has become very important for digital signals running at high data rates. Line loss over distances at these speeds have also created challenges. Recently fiber optic cables have been making their way into the avionics and aerospace arena. Fiber optic cables offer an alternative that remedies some issues experienced with copper, but also create new ones. They offer much lighter weight and size option, can provide communication over much longer distances, and are immune to various noise and interferences such as crosstalk, RFI, and EMI. Some downside characteristics include coating CTE issues, outgassing, moisture retention, shrinking, and long term effects on performance due to exposure to radiation.

Maintaining a technical roadmap where the data link capacity continues to grow at least as fast as the performance growth rate will be vital going forward, particularly as Moore's Law is more difficult to maintain.

VII. CONCLUSION

High performance digital front-end processing applications present a unique set of challenges for FPGA and ASIC designs. Moore's Law has provided increased capability for processing nodes and this in turn has increased the need for architectures that can exploit this increased capability and continue to provide data at the high required data rates for efficient processing.

ACKNOWLEDGMENT

The authors acknowledge the management of the Digital Technology department at Northrop Grumman Electronic Systems in Baltimore, MD for their support of the work for this paper.

REFERENCES

- [1] G. Moore, "Cramming More Components onto Integrated Circuits", *Electronics*, pp 114-117, April 19, 1965
- [2] G. Moore, "Progress in Digital Integrated Electronics", *Technical Digest 1975, International Electron Devices Meeting, IEEE*, pp. 11-13, 1975
- [3] J. Holland, J. Horner, M. Corbin, E. Glaser, and G. Petrosky, "Processor Building blocks for Space Applications," *IEEE High Performance Extreme Computing (HPEC) Conference*, September 2014.
- [4] J. Holland, J. Horner, R. Kuning, and D. Oeffinger, "Implementation of Digital Front End Processing Algorithms with Portability across Multiple Processing Platforms," *High Performance Embedded Computing (HPEC) Workshop*, September 2011.
- [5] B. Jonsson, "A Survey of A/D-Converter Performance Evolution", *17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Athens, Greece, pp. 766-769, December 2010
- [6] B. Murmann, "The Race for the Extra Decibel: A Brief Review of Current ADC Performance Trajectories", *IEEE Solid-State Circuits Magazine*, vol. 7, no. 3, pp. 58-66, September 2015
- [7] W. Holt, "Moore's Law: A Path Going Forward", *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, pp. 8-13, February 2016
- [8] E. Bogatin, *Signal and Power Integrity – Simplified*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2010, pp. 381-382.