Factors for DSP Success

Bob Caspe

March 19, 2004

Table of Contents

Introduction

This article is intended for potential investors who are considering investment in a DSP company. A general overview of the DSP market and history of designs is offered in order to provide some perspective on the industry and to provide a general understanding of the factors required for success.

About the Author

Bob has designed DSPs, written software for DSPs and run companies that designed, programmed and used embedded DSPs from 1969 to 2003. Currently, Bob is offering his services as a consultant to assist investors with the evaluation of investment strategies including, but not limited to, the evaluation of DSP market strategies and architectures.

Definition – What is a DSP?

DSP stands for Digital Signal Processor, which has become synonymous with prior category names: Signal Processor and Array Processor. Today, typically DSP designs are single custom integrated circuits that contain either multiple processors or processors with architectures that make them especially good at math. Historically, Array Processors were complete, stand-alone, racks of equipment that worked in tandem, often, with more general purpose computers or they were circuit board sized processors that plugged in to general purpose computers.

In all cases, their purpose was to perform mathematical equations, at high speed, redundantly on large amounts of data – at performance levels that exceeded that of general purpose machines of comparable cost.

(The original term for this architecture, Array Processor, was meant to convey that the data set that was being operated upon was an array or vector, and that the processor was proficient at vector math. Confusion with the alternative concept of “an array of processors” which might, in fact, be used for this purpose, probably was responsible for the loss of popularity for this name. Although even early DSP designs had multiple multipliers and adders, contemporary architectures often do have totally redundant processors arranged in some sort of array.)

History of the DSP

In the late 1960s and early 1970s Array Processors could be found primarily in military systems. Specifically, radar, sonar and communications systems served as the beginning of the use of custom architectures for cost/performance advantages when solving the numerical equations used to emulate prior analog systems. In, for example, a typical Doppler sonar or radar system, Fourier transforms would be used to convert the time domain data to frequency domain data and thus extract the Doppler shift and calculate the speed of the target.

In the late 1970s, medical imaging systems like the CAT scanner were the first use of Array Processor architectures outside of military applications. (Note that the use of continuous tone displays on computers followed a similar migration from military systems to medical systems over the same time period and for the same reasons). In a CAT scanner, the spatial convolution (done through Fourier analysis) and back projection were the mathematical processes that required the speed of an Array Processor. At the time that CAT scanners were first being developed, mini-computers like the PDP-11, Data General Nova and the Interdata-32 were contemporary general purpose architectures and therefore, the Array Processors were often circuit boards that plugged in to these machines.

In the 1980s the same mathematical equations, combined with a further decrease in cost/performance moved the Array Processor from the medical industry into the professional graphics and audio markets. For example, JPEG compression was originally designed for use with medical images, but found its real home in graphic arts applications. (It's interesting to note that JPEG compression never was accepted for use with medical images because of the compression artifacts that could be introduced.)

Then, in the 1990s, with even more reductions in cost/performance, Array Processors or as they were now known, DSPs, found their way into consumer products including digital cameras, DVD and MP3 players.

An interesting example of this migration downward in cost, for me, is that back in the early 1970s, I worked on a rack of equipment that we designed which mounted into a military jet. It would perform adaptive noise cancellation for the pilot's headphones reducing ambient engine and wind noise. Today, the same algorithms are built into an inexpensive headphone available on a business-class seat on a consumer aircraft.

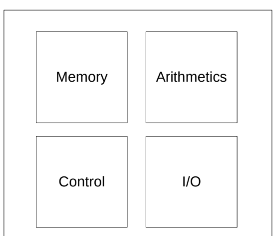
Speed Improvements

Back in 1971 we advertised our Array Processor as having an extraordinary 6 million multiplies per second. Of course, that was a rack full of equipment at a cost of $100,000. The architecture actually included 4 multiply-accumulate sections that acted in parallel. Today, in comparison, Analog Devices offers the TigerSHARC Processor which can perform 4 billion multiply-accumulates (4000MMACs) per second in a sub-$100 25$mm^2$ plastic package. That's roughly a 1,000,000 to 1 performance improvement over 33 years.

(Note that often the number of multipliers or adders in a particular DSP design relate to the specific algorithm for which the DSP has been designed. For example, Fourier transforms are often calculated using an FFT, Fast Fourier Transform. A Radix-2 version of this algorithm has, what is called, a butterfly calculation that involves 4 multiplies and 6 add/subtracts. Thus, a DSP designed specifically for this calculation might have this number of independent multipliers and adders, pre-wired in a butterfly arrangement)

The speed of DSPs is classically presented in terms of their mathematical capability, often quoted in MACs or MMACs standing for Multiply Accumulate (add) or Million Multiply Accumulate operations. (The reason that multiply and add are joined together is that most signal processing algorithms use this basic construct to perform a filter or transform.) It should be noted that some DSPs use fixed point notation while others use floating point, the latter being more expensive to implement but often easier to program.

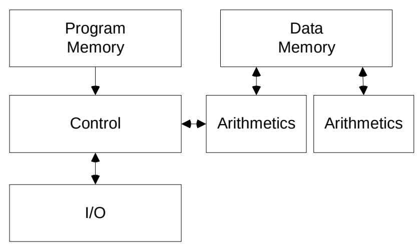
Architecture



Von Neumann Block Diagram

All computers today have architectures which were first defined by Von Neumann in 1946 as having 4 basic building blocks: memory, control, arithmetic and I/O. An Array Processor is simply a computer for which the architecture has been modified so as to amplify the arithmetic capability usually at a sacrifice of simplicity or ease of programming.

There are many simple, but illustrative examples.



Early DSP Block Diagram

Example 1: Separate the program and data memories. Prior art had both programs and data stored in a common memory. By separating these two entities, one could have the machine read data and the next program step simultaneously thus increasing the performance of the machine. Unfortunately, the programmer would now need to keep track of where and how much data and program were stored.

Example 2: Pipelining. A common technique for increasing machine speed is to partition instruction processes into "baby steps" and to separate the baby steps into separate portions of hardware and pass the intermediate results from portion to portion in a bucket brigade fashion. For example, while portion 1 is fetching the next instruction from program memory, portion 2 is executing the prior instruction, thus allowing the fetch and execution to be occurring simultaneously, albeit not for the same instruction. The problem introduced occurs when either intermediate results are needed for the next instruction (they might not be ready) or a conditional branch is taken. Again the burden typically falls upon the programmer who must keep these considerations in mind. A simple description of pipelining for the novice can be found at Stanford University's site.

Example 3: Multi Processors. Another popular technique is to simply have multiple – often identical, processors or data paths each sharing memory or results that are passed between them. The additional complication should be obvious.

Why Bother?

Today, your common experience is that you can simply pay more money for a PC that is faster. So why bother with modifying the architecture in order to increase the speed? The answer is simple, it's cheaper. And, this is the key to the DSP market. DSP designers are accomplishing a simple overall objective:

For a given algorithm, there is always a general purpose computer that can perform the equations at some performance level. Often, by simply using a faster general purpose machine, one can achieve performance levels that are consistent with a particular real time requirement (see Defining Real Time Requirements). Keep in mind that general purpose computers extend up to Cray or other super computers.

Thus, the DSP, or using a variance of architecture at the cost of complexity of use, is a method for achieving a particular performance level at a substantially lower cost than offered by general purpose architectures. And, here's the rub: General purpose machines are constantly getting cheaper and faster, and thus are constantly obsolescing prior DSP designs.

To make things worse, general purpose architectures, for which there exist advanced development tools, are incorporating techniques heretofore only found in Array Processors. For example, instruction pre-fetch, independent data and program memory caches, and pipelining are all standard elements of high performance RISC architectures today. Thus, the gap between "DSP only" applications and those that can be done by conventionally architected machines is narrowing.

Defining Real Time Requirements

Unlike when one purchases a PC and considers the 2.5GHz machine as being faster and a bit more convenient than the 1GHz machine, often, the desire to include a DSP is driven by a need to perform a particular mathematical equation at a predefined speed. For example, if one is trying to decode an MP3 compressed audio stream, one needs to perform the equations fast enough to play the resulting audio data stream in "real time" so that it sounds correct. (Note that alternatively, one could decompress it in non-real time and spool the results to a disk from which it would then be played in real time later.)

Often, for audio and imaging applications, there exists a specific real time requirement. Just as often, there exists a performance of "convenience" which, while less critical, often defines the potential marketability of the design. In some cases, digital designs based upon DSPs are replacing analog or mechanical systems that create specific cost and performance targets that while arbitrary, are still real market requirements. An example would be image stabilization through alternative methods.

DSP Market Opportunities

I have a preferred way of thinking about market opportunities for DSPs. First, one must define if a mathematical algorithm exists for which there is unsatisfied market demand. This could exist in any of three possible forms:

Form 1: There exists a mathematical algorithm which is a replacement for an existing mechanical or analog system which has a defined cost/performance and a defined market penetration that can either be replaced or caused to grow through achieving a new cost/performance level.

> An example might include: Make a “digital zoom lens” for a digital video camera by having a super high resolution sensor and a DSP that processes the image in real time giving the user the illusion of continuous zoom through sampling and interpolation algorithms.

Form 2: There exists a mathematical algorithm that is already being performed by a DSP but for which a modification of architecture or technology can offer a cost/performance improvement that can either replace or cause to grow the existing market.

> An example might include creating a higher speed DSP that can combine the video and audio processing found within a DVD player, but separated into independent processors today.

Form 3: There exists a mathematical algorithm for an application that has heretofore not been commercialized possibly because of the cost/performance limitations of prior architectures or possibly because there is no need.

> An example might include: Add voice recognition to a simple $10 light switch.

How Do New DSP Markets Emerge?

DSP markets emerge through all three, above listed, mechanisms. The first two forms listed imply an existing market for a “prior art.” The third form, however, is often the most risky in that it requires the creation of a new vertical market value proposition that may not be valid. In the above example, users may not like to control their lights through voice commands, thus invalidating the premise. Form 3 therefore represents the highest risk for business success.

Form 3 has one additional problem. Keep in mind that the simpler solution using a general purpose processor is always creeping up behind any DSP application. Thus, doing market development of a new application with a DSP always runs the risk that the bleeding edge market development costs are never recovered because by the time the market develops; one can accomplish the task with even simpler and less expensive general purpose architecture.

This problem is not a problem just experienced in the DSP market, but is one that affects most of High Technology. As general purpose computers get faster and more powerful, and their applications get more powerful, the need for specialized custom solutions is forced up-stream to ever more complicated problems.

The Concept of the General Purpose DSP

There are a few companies who currently sell DSP development kits and try to target their devices at new and emerging markets. Of course, once a market is well defined, it's often possible to achieve further improvements in cost/performance through a new and custom design that is perfectly tailored to the algorithm, memory size and I/O requirements. However, it is clear that many new applications are started with existing DSPs that still offer cost/performance improvements over more conventional architectures. In today's market, where video and audio compression is being integrated into other systems, there are developers who can use existing DSPs for these new projects.

It should be noted that even the successful companies today that offer general purpose DSP development kits, started themselves in specific vertical markets trying to solve single problems. It should also be noted that the development support costs for general purpose architectures has often proved to be overwhelming to these companies.

Development Tools

One of the most difficult aspects of providing DSPs for general development has been the creation of tools for compiling, debugging and optimizing code. Some companies have even elected to forego this and simply limit software development to their own internal developers who become intimately involved in the coding and tolerate the inadequacies of the toolset.

Compilers

In an ideal world, users would like to simply write C or C++ code and have the compiler partition the algorithms to the multitude of processors and deal with the pipeline delays.

Unfortunately, while this is ideal, it is seldom if ever reached. Optimization often requires a bird's-eye view of the data flow that eludes the compiler. For the most part, programmers are still required to partition and organize the code for optimization.

Debugging

Debugging multiprocessor code is difficult at best. It often requires running debug environments for different processors that run different parts of the data flow, in different languages simultaneously. Enormous costs are associated with code debugging and the quality of the tool set can be critical. While industry averages for RISC processors is 15 lines of code per day per programmer, only one tenth of that is typical for DSP code development.

Optimization

Optimization often requires real time analysis of data flow and is not simply a matter of timing different routines. Specialized tools are often used for this aspect of code development.

ASIC Configuration for a Custom IC

Often, the goal of a DSP customer is to enter the market quickly with a selected off-the-shelf DSP, but then, if the market grows, to achieve further cost/performance improvements by creating a custom ASIC with a DSP core included. Various "open platform" DSPs like the OAK core have been popular because of this.

Follow-on Competition for Successful DSP Designs

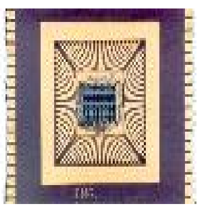
When there is a successful DSP implementation, there are three forms of competition that are constantly applying margin pressure:

1. Alternative general purpose DSPs that can be programmed to meet the specific requirements of a vertical market. A good example can be found in the digital camera market where there are a number of DSPs that compete for the same market space.

2. Ever increasing performance standard microprocessors making a simpler design possible. The best example is the modem market where eventually, the CPU of a PC was capable of replacing the DSP found in the modem.
3. Custom ASICs that implement the algorithm in a hard-wired way. A good example is that one can purchase a core JPEG compressor and de-compressor which take roughly 40,000 gates and can run at video rates. One can license this core and place it within a new ASIC. This core is hard wired and cannot do any other algorithm.

Protection from margin pressure can come from the development of proprietary algorithms that are difficult to copy. Some algorithms have been protected through controlling the vertical market through brand recognition as has been done by Dolby.

Re-Configurable DSPs



In the last few years, several University spin-offs have been touting the idea of DSPs that have re-configurable internal paths and processors so that the actual architecture of the chip can be modified to better fit the specific algorithm that is being solved. This is quite attractive if one really has a problem that requires re-configuration. However, these types of applications are few and the additional cost for this capability runs contrary to the basic goal for selecting a DSP in the first place as a route to cost reduction. However, these devices could be ideal for prototyping a new design and then, if commercially successful, moving to a custom ASIC that embodies the data flow.

Current Markets

There are a wide range of uses for DSPs currently. I present a sampling of these uses to give the reader a better understanding of where they are found.

Modems

Modems were the birthplace of the DSP. They also had the quality that their architecture was the digital solution of a set of equations that emulated the analog processes found in an analog modem. The implementation of FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters through simple linear equations that were executed on data derived from an A to D converter were some of the very first demonstrations of how a digital processor could emulate, and eventually replace, an analog system.

If one examines a contemporary modem design, it typically contains two components, a general purpose DSP of moderate speed and low cost, and a DAA which interfaces to the telephone line. Several years ago, adding a modem to a PC involved either an external box or insertion of a custom card that contained both the modem and the DAA. It is interesting to note that currently, PCs are fast enough that “soft-modems” or simply stated, modem software running on the PC itself has replaced much of the modem market and the only thing left on the card is the DAA. This serves as a good example of the niche that DSPs fill and how they are eventually replaced with conventionally architected machines.

Texas Instruments early success in DSPs was a direct result of the emergence of the modem market and its need for math processing.

Digital Cameras and Video Cameras

Most recently, DSPs have emerged within both the digital camera market and digital video camera market. The explosive growth of these markets over the last 10 years has resulted in a number of DSP designs that have been customized for this application. The extraordinary price pressure on both digital cameras and video cameras has forced the manufacturers to attempt to integrate as many disparate functions into single ASICS as possible including the mathematical engines for JPEG compression and decompression.

The mathematical processes most likely relegated to the DSP in these designs include JPEG compression, Gamma mapping, Interpolation, uniformity correction, white balance and sharpening.

Video and Audio Players (DVDs)

The DVD market combined with other media players represents an interesting case study in market growth through a single vertical market. Most of Zoran’s success has been through providing the audio decoding DSP found in DVD players. Their devices are not generally programmable by customers but rather, Zoran provides the custom software necessary for a particular vertical market.

Within DVDs and other players, the DSPs are used to provide MPEG audio and video decoding.

Printers

A number of companies have found success in the narrow vertical niche of providing DSPs for printers. These DSPs perform RGB to CMYK color conversion, sharpening, and interpolation. If a scanner is included in the product, then white balance algorithms are also found.

Washing Machines and other devices

One of my favorite applications for a DSP is in newer vertical clothes washing machines. In these machines, the DSP is fed the "back EMF" (sensing motor load) of the motor circuit indicating the relative load position. Optimum cleaning is obtained by having the clothes spill from the top of the drum when washing. Thus, algorithms in the DSP calculate the best motor speed based upon this data. This example illustrates the development of a vertical application outside of traditional markets.

Automobiles

Over the last several years micro processors have found their way well into the automotive design market. Soon, we will start to see even more image processing and audio processing within these designs. For example, the airbag firing system, collision avoidance, voice recognition and synthesis are all places where DSPs can find usage within a car.

Surveillance and/or Smart Cameras

A number of companies have been pursuing the combination of DSPs with imaging sensors in order to make intelligent decisions at the camera and thus reduce the bandwidth of data coming out of the camera. Examples include surveillance cameras that do facial recognition or motion detection, and automotive applications where the cameras detect collision potential, road signs, curb position or even driver fatigue.

The most successful of these applications have been in manufacturing. Review of the products of companies like Cognex can be instructive about this industry.

Voice Synthesis and Recognition

Voice synthesis and recognition are illustrative examples of market applications where RISC architectures have become powerful enough to implement the vast majority of these algorithms, thus replacing DSPs that had been used.

Medical Devices

Historically, DSPs could be found in larger medical imaging systems ranging from CT, MRI, Nuclear Medicine and Doppler Ultrasound. In some cases, fairly large PCs eventually made their way into these designs and thus replaced some of the more custom architectures. However, in the future, with the advent of more portable low cost devices, DSPs will again find homes in medical systems.

Military Devices

The need for ever faster and cheaper DSPs has not ever left the military applications market. There are always more telephone calls to listen to, and ships to follow. Advanced targeting systems, night vision systems, communications systems and eavesdropping systems as well as the traditional radar and sonar applications continue to require improving cost/performance.

Conclusions

It should be apparent that one method for creating a successful DSP strategy involves identifying a vertical market segment that:

- requires a particular cost/performance,
- is currently not being adequately served and,
- offers explosive growth in a short period of time.

The DSP that is offered should have architectural features that make the particular vertical application even more cost/effective than can be achieved with other general purpose architectures.

Often, the most likely places one looks for this type of market opportunity is at the leading edge of speed development. The reason is simple; the presumption is that the vertical market is undeveloped primarily because the prior solution was prohibitively expensive at this performance level.

In some cases, the vertical market remains undeveloped because either the market need isn't really there, or the algorithm development isn't quite complete. These two cases represent extraordinary risk to the DSP company. Spending several years waiting for either a market or algorithm development to mature gives competing, simpler and more cost effective technologies the opportunity to mature and move in later.

Today's leading highest performance DSPs are well beyond the range of simple audio algorithms. Thus, the market opportunities that do exist can be found in imaging applications. These include: object recognition, person recognition, resolution scaling, compression, de-compression, robotics, machine vision, consumer videography, professional studio videography, security, safety systems and simulation to name a few.

Predictions

My DSP markets of choice for the next few years are three.

First, the HDTV market is rapidly expanding. It is being driven by the influence of both available source material as well as the emergence of the low cost DLP HDTV projectors. This includes the next generation DVD systems that will need to decompress HD streams as well scale lower resolution DVD signals up to HD format.

Second, I believe that it's time to re-cycle back into the military market. Military applications will be driven by the tactical problems created for fighting terrorism rather than a ground war. This might include robotic systems and surveillance systems.

Finally, the medical market is also primed for re-emergence. However, this time, it will most likely be driven by the demographic shift to an aging population. Applications might include automated pathology systems.