

## RAW - How This Embedded Board and Systems Business Works

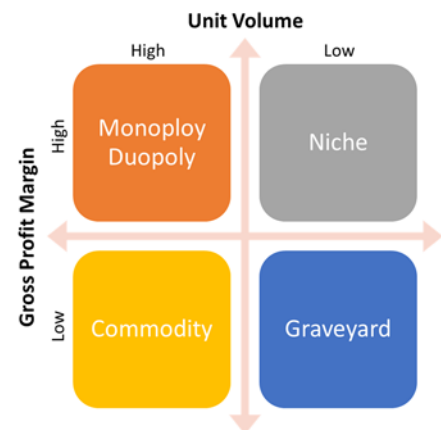
Ray Alderman  
Chairman of the Board  
VITA

*[This is an expanded version of a presentation given at the VSO meeting in November 2015. This whitepaper is in raw, unabridged format.]*

### Introduction

To understand our industry, let's start at the macro level. Using a modified Boston Consulting Group matrix, we can see the two basic variables defining our markets: volume (of product shipments) and margin (gross profit margin or GPM: sales minus the direct manufacturing costs, divided by sales). Typically, as volumes go up, margins come down. That gives us only four quadrants or markets in this embedded industry: HV/HM- high volume/high margin (monopoly or duopoly markets), HV/LM- high-volume/low margin (commodity markets), LV/HM- low volume/high margin (niche markets), and LV/LM- low volume/low margin (the Graveyard).

- HV/HM (Monopoly/Duopoly): This market is not available to us. Only Intel and Microsoft have enjoyed this position.
- HV/LM (Commodity): We have experienced this market characteristic for some embedded boards and systems, back when many board companies were infected with the froth of feral hydrophobic financial horrors by the telecom equipment makers. The potential volumes were high for our industry, but the margins were very low. The companies in this segment, over time, either sold out, merged, or went out of business. Volume is a great mistress but a terrible wife. Once you are married to volume, your life will be tenuous and miserable unless you sell-out or go elsewhere. The basic strategy for companies in this segment is to trade margin for market share, and that killed them.
- LV/HM (Niche): This is the primary stable market for embedded boards and systems. The military market is the best example of high margins and low volumes. The strategy for companies in this segment is to trade market share for margin, a healthy trade-off.
- LV/LM (Graveyard): We have seen numerous board-level technologies enter this segment as they are replaced by newer technologies. In particular, all the PC-based technologies end up here (ISA, EISA, PCI, PCIe Gen-1, PCIe Gen-2, etc.) as the next generation chips come to market. Others, like S-100, Miltibus-1, Multibus-2, STD, G-64/96, etc. encountered the same fate. In this segment, both volumes and margins are declining over time. Neither margin nor volume are important in this segment.



In the history of this industry, no company has ever reached \$1 billion in sales, and there's a good reason for that: once the volumes get high enough, customers move away from board vendors, to the contract manufacturers. Only three companies have achieved above \$500 million in sales in our industry, before they fell back to much lower levels, broke up, or sold out. That tells us that the HV/LM commodity market is a temporary market segment for embedded suppliers.

Since about 1989, the companies that had to sell out or went out of business made one of two basic mistakes: (1) they entered a commodity market with a niche strategy and could not drive their costs down fast enough to remain competitive (telecom boards), or (2) they entered a niche market with a commodity

## RAW - How This Embedded Board and Systems Business Works

strategy and the volumes never materialized (networking boards). The most glaring examples for (1) are when Solectron bought Force Computers, and then Intel bought Ziatech: in both cases, they couldn't drive their cost down fast enough (Solectron sold Force to Motorola Computer Group, and Intel sold Ziatech to Performance Technologies, both for much less than they paid for them). For case (2), a board company making Ethernet boards disappeared when board vendors put the Ethernet chips on their SBCs, and other companies making add-in local-bus cards for Sun workstations went south when the Ethernet interface was integrated onto the motherboard (the volumes never materialized). Telecom and networking has killed more engineering, marketing, and sales careers than the Bubonic plague killed Europeans. Some telecom-focused board companies were astute enough to sell out to larger companies on the upside before reality took hold.

There's much more to the BCG matrix used here. I recommend that you get a copy of the book, "Perspectives on Experience" by the Boston Consulting Group (1968). This book was only published for BCGs customers, not for the public. You can find used copies on Amazon, if you are lucky.

As an example of what you will learn in this book, every time your shipments of a product doubles, your costs will automatically go down by 20-30%. Every time the volume of products your market segment ships doubles, the costs of all competing vendors in that industry automatically go down by 25-30%. I'll get more into this phenomenon later, when I talk about the Lanchester Laws, so save your snarky comments, that this rule only applies to markets like toothpaste, laundry detergent, and deodorant until you read the Market Share Strategies section. And be prepared to apologize for your thoughts.

### Market Numbers

Let's look at our markets today. Our military embedded board and system merchant market (all form factors) is about \$1.2 Billion in sales. Let's wrap our heads around this number by comparison: the cat litter market in the U.S. alone is about \$1.8 billion, and their product is infinitely less complicated. Their margins are low, their volumes are high, and they use a complex distribution system to get the litter to the cats. Our margins are high, our volumes are low, and we deal direct with the users.

If you add-up the sales of the top three companies in the MIL segment, you get about \$750 million. That leaves about \$450 million for all the other players. While that looks high, consider mezzanine cards, packaging, backplanes, power supplies, and motherboards. It's a conservative number (the merchant market does not count boards and systems made by contract manufacturers for specific customers or sold through distribution). The three major companies in this segment each do \$200 million or more. The primary value added in this segment is intellectual value, which is why the vendors enjoy high GPM.

The merchant commercial market for embedded boards and systems (telecom, industrial, commercial, transportation, etc.) is about \$2.0 billion. By comparison, the dry cat food market in the U.S. alone is about \$3.8 billion. The cat food supplier's margins are low, their volumes are high, and they use a complex distribution system to get the food to the cats. The players in this commercial board market show low margins, low volumes (compared to other segments), and they sell directly to the end users.

When you add-up the sales of the top four companies in this commercial board segment, you come up with about \$1.5 billion, leaving \$500 million for the rest of the players. This market is much harder to quantify since the server and PC manufacturers sell motherboards for embedded applications. The volumes are higher here, and the selling prices (and margins) are very low. Two of the top four companies in this segment do more than \$500 million each. The primary value-added in this segment is manufacturing value, which is why these vendors are plagued with low GPM.

This all says that the total embedded board market, that we can participate in, is about \$3.2 billion. You could stretch this number to maybe \$3.5 billion, considering that my estimates may have missed something here and there. But, let's use \$3.2 billion for this discussion and the math. Also, understand that these numbers are worldwide market numbers.

If you look at the numbers again, you can see that 61% of the military embedded board/system market is controlled by three vendors, \$750 million. The remaining \$450 million (39%) is shared by many small vendors, which says there are some nice profitable niches out there. Several smaller companies do \$30 million to \$50 million in this segment.

## RAW - How This Embedded Board and Systems Business Works

Four companies control 75% of the commercial embedded market segment, or \$1.5 billion. That leaves \$500 million (25%) for the smaller companies. You can see that there are not many fairly large or profitable niches for the smaller guys. Not many of the smaller companies do more than \$10-15 million in sales here.

### Business Models and Gross Profit Margins

There are only three values you can add to electronic components to make money in our markets:

1. **Manufacturing value:** Adding only manufacturing value will get you about 8-10% gross profit margin. If you don't believe me, go look at the financial reports of the major contract manufacturers. The CEMs make money on their supply chain. They don't buy ten thousand 0.1  $\mu$ f capacitors for one product run: they buy ten million .1  $\mu$ f capacitors for a hundred different products coming through their machines and charge customer more than they cost. You can only make money with significant volume when adding manufacturing value. And it only works if volumes can scale. This is where the telecom and some commercial board makers reside.
2. **Service value:** In our industry, this means doing integration: assembling the chassis, wiring-up the power supplies, acquiring the boards and inserting them into the chassis, adding some software, engineering the cooling, testing the system, and shipping it. Adding service value will get you about 20-25% gross profit margin. The problem with integration is that it doesn't scale well. Integration is labor-intensive, and the labor required is higher skilled and more expensive than traditional manufacturing workers. Because of complexity, one engineer can handle only a few major integration projects at one time. Add more engineers, and your margin falls unless you have more volume. There are a few board/system companies who reside in this segment.
3. **Intellectual value:** Adding Intellectual value is applying high-level technical skills that other companies do not have. Examples are companies who can design high-performance GPU boards, high speed A-to-D converter boards, and FPGAs with DSP cores inside. Adding intellectual value will get you 50-70% gross profit margin. If you want to raise your GPM, add more intellectual value. This is the segment where most of the military suppliers reside.

### Gross Profit Margin

Let's explore gross profit margin (GPM). No company in our industry has ever achieved 75% GPM. That's because, since we are a low volume niche market, we cannot get our component costs below 25% of the selling price. However, several companies, by adding intellectual value, have enjoyed near 70% margins. When you look at all the exit multiples (selling price divided by sales) of the many M&A transactions in this industry's history, and you can see the GPM of the companies being sold from their financial statements, you will find that the selling price works out to be 5 times their GPM: if a company was making 50% GPM, they were sold for about 2.5 times sales. This formula works perfectly, within a percentage point or two. The highest multiple ever paid for a board company was about 3.4 times sales (the company had a GPM of about 67-68%). If a company was making 75% GPM, they would sell for 3.75 times sales, and that has never happened because board companies are restricted by the 25% direct manufacturing cost rule. The lowest multiple paid for a viable board company was about 0.7 times sales (a telecom board company making about 14% GPM, but with very heavy administrative costs making them marginally profitable). Software and services companies sell for 5-10 times sales because their direct cost of sales is near zero (but their administrative costs are 30-40% of sales, and can be significantly reduced through massive layoffs). A complete list of historical M&A and the exit numbers for our industry are a topic for another paper in the future.

Another word about margin: the margin on boards is much higher than the margin on systems. Systems require more skilled labor content. It has the same problems as integration and it doesn't scale well. The margin on the packaging and backplane elements is notoriously lower than boards. The margin on power supplies is abysmal. If you are building systems, your GPM will fall to about 20-30% if you do it right. And down to 10-14% if you sell to the wrong customers like telecom.

## RAW - How This Embedded Board and Systems Business Works

### How to Be Successful in This Embedded Board and System Business

There are only three ways to be successful in the embedded board and systems business: Be first, be smarter, or cheat (taken from the movie "Margin Call", and applied to our business):

1. Be first: Be the leader in market share, or be first to market with your product. The advantage always goes to the market leader or the first supplier of a new product. There's a business model and strategy based on market share (The New Lanchester Strategy), and that's covered below.
2. Be smarter: If you are not the market share leader or first with new products, you have to niche-out and avoid competition with the big guys. They will beat you every time. So, you have to design niche products, for niche applications and niche market segments. Design products targeted at smaller application segments that the big guys ignore. Being smarter means defining and filling the needs of those applications. As an example, design and sell board products to industrial users instead of big powerful 6U processor cards to military or telecom customers.
3. Cheat: It's hard to cheat in the open market, against big entrenched players. The best way to cheat is to fragment an existing market, by creating a niche, by doing an open standard for your version of a board or system implementation. Creating a niche is like being smarter, but is marketing-based instead of engineering-based. At VITA/VSO, the policies and procedures allow any company, along with two other sponsors, to write a new standard without interference. Nobody can stop them. The VITA philosophy is that no committee or group of people has the right or the authority to decide which technologies can or cannot be standardized. All power in the VSO is vested in the working group that is writing the new standard. The market will decide which standards are successful and which one's fail.

Profiles in VPX are another acceptable way to create a niche, which is why we have so many of them. I was not fond of profiles in VPX, but my opinion is probably associated with being bitten by the profile dog during the Futurebus days. The VSO policies and procedures were intentionally written in such a manner that all companies can take their ideas, and create a new niche fair and square. John Rynearson and I became "standards engineers" when VITA became an ANSI accredited standards developer back in 1993. VITA's motto is "Open Systems, Open Markets", and we applied our skills to design a standards architecture that works most efficiently to reach that goal. VSO policies have replaced Draconian rules, found in other standards developers and consortia, with market forces. These procedures reduce barriers to entry into the markets. DO NOT change these existing procedures or add committees that have the power to accept or reject ideas for new standards. Politics and competitive agendas will enter the process, and you'll be in big trouble. Standards are the fulcrum of our industry. They give us mechanical advantage, to fragment markets and create new market niches for growth.

### Marketing Strategies

There are really only two strategies for us in the embedded space: technology-push or customer-pull:

1. Technology Push is taking a new idea or product to a customer and convincing him to adopt it. What we are really doing is "Technology-Pull". The semiconductor makers actually do the technology-push, and we come along for the ride. We don't have to push PCIe Gen-4 or the next faster version of a processor chip. We just ride the adoption curve they create. Since all the board and system makers ride this curve at the same time, it is critical to "be first" with your new products using these technologies, by being in the semiconductor maker's "partner" program, to get advanced information and a time-to-market advantage over competitors. Pure technology-push in our industry is problematic. Just look at the adoption level of MicroTCA<sup>®</sup> or PICMG 2.16 as examples.
2. Customer Pull is asking a customer what he wants, designing it for him, building it, and delivering it to him. This strategy is more prevalent in the high-performance segments like Military applications, where the customers have an insatiable need for performance.

In commodity segments like telecom and industrial, the customers will use the cheapest generic version of a PC motherboard since their applications are not particularly unique or demanding. Industrial applications typically use sequencers with no real-time or significant performance requirements. Telecom applications

## RAW - How This Embedded Board and Systems Business Works

are just moving bits from one place to another, where everything is buffered anyway, so there are no real-time demands. Using customer-pull in this segment means designing a product to reduce a customer's cost.

Very seldom do we see "Customer Push", but it does happen now and again. The best example is when SecDef Perry issued his COTS Initiative (Commercial Off The Shelf) for military electronics in 1994.

This information leads us to some conclusions: Niche customers want increases in performance and capability and they will pay 50-70% GPM for it. Commodity customers only want reductions in their cost and they will pay 8-10% GPM for it.

### Market Share Strategies

This train of thought takes us into a discussion about market share strategies. To be a player in a market, you must have 26.1% market share. To be the market leader and drive the market, you must have 41.7% market share. To have a monopoly, you must have 73.9% market share. Between 26.1% and 73.9% market share lies the most profitable phase in a company's life. Above 73.9%, the correlation between market share and profitability declines. Just look at Microsoft's financials for the past ten years to prove this axiom.

From the market numbers stated above, the top three vendors in the embedded military segment have 26%, 20%, and 15% market share (rough estimates). The remaining 39% of this market is spread out among many small vendors. The top three are fighting it out to see which one can get to 41.7% and become the market leader. So far, none have reached that critical mass. If you agree with the \$1.2 billion valuation of this market segment, then a company needs \$313 million in sales (26.1%) to be a player, and \$500 million in sales (41.7%) to be the market leader. We have one player in this market segment.

In the commodity commercial embedded segment, the top four vendors have 35%, 25%, 10%, and 5% market share (rough estimates). Two of these companies are players, if you give the 25% market share holder the benefit of the doubt. Again, the top two are fighting to be the market leader and achieve 41.7% market share. The remaining 25% of this market is spread out among the smaller vendors. If you agree with the \$2.0 billion market estimate, then a player needs \$522 million in sales (26.1%). The market leader would need \$834 million in sales (41.7%). The two top players in this segment have been close to this number, but fell back due to deteriorating economic conditions or primary customers moving to contract manufacturers. This segment suggests the inherent limit to the size of embedded board and systems companies, and why no company has ever achieved \$1 billion in sales.

And, there's more. In the history of our niche embedded board and system market, no company has ever achieved 41.7% market share in the embedded markets. In the 1980's, Motorola Computer Group did achieve about 50% market share in 6U air-cooled 68000 and PowerPC VME processor cards, but had no more than about 25% of the total VME market. They were the market leader and driver in processor cards, but they could not drive the total VME market. Evidence of this statement is the number of smaller VME vendors that thrived, over 100 companies, back in the 80's and 90's. At 41.7% market share, the market leader can drive the smaller companies out of the market, which Motorola did in 6U CPU cards. The second company in VME market share began to make SPARC and Intel-based processor cards to survive and grow. Other VME vendors went into smaller 3U board sizes and mezzanine cards. As the VME CPU market fragmented into other processor technology niches, and with the addition of conduction-cooled 6U CPU cards for the military, MCG's market share in CPU's fell below the 41.7% required to be the market leader and they could no longer drive the market and grow. So they had to enter other niches, like CompactPCI, 2.16, and eventually ATCA.

As you can see today, no company in our business has the 41.7% market share to be the market leader, in either the CPU market or the overall embedded board and systems market. Only three companies have 26.1% or more to be players. In the history of our industry, we have seen more than a few billion dollars spent on acquiring board vendors. Even with the additions of these company's sales and products, none of the acquirers have achieved 41.7% market share. Our markets live on very short technology S-curves, that last for about 7 to 10 years. Some are shorter like Multibus I, Multibus II, ISA, EISA, CompactPCI® and 2.16. And some are longer like VME, at over 30 years and still going.

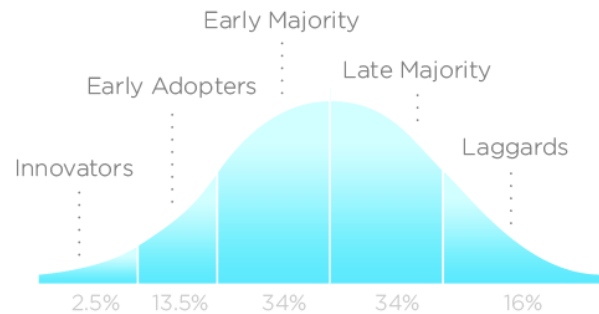
## RAW - How This Embedded Board and Systems Business Works

Finally, our markets are inelastic. If you drop your price by 50% (and ignore your margin), your sales will not double. By definition, we are in a niche market with a small finite volume. Our market S-curves are short. Add those facts to the reasons why no company has ever achieved 41.7% market share or \$1 billion in sales.

For those of you who are unfamiliar with S-curves, they are the first half of the technology lifecycle curve. In the total lifecycle curve, we have innovators, early adopters, early majority, late majority, and laggards.

The S-curve has the infancy, expansion, and maturity phases. It's obvious that the majority of the growth is in the first half, and also the majority of the profitability, from innovators to maturity.

That's why we live and die on the S-curve, where the growth and profitability exist.



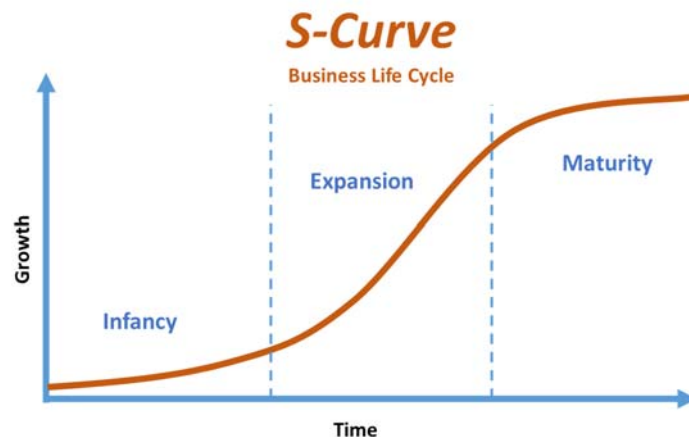
INNOVATION ADOPTION LIFECYCLE

The 26.1%, 41.7%, and 73.9% market share trigger-points originally come from Frederick William Lanchester's book, "Aircraft in Warfare" (1916), revealing his first and second stochastic laws. His work was expanded by Dr. B. O. Koopman, a Columbia University mathematics professor drafted into the U.S. Navy, and Lanchester's laws were used during the battles of WWII. If you want to understand his formulas and laws better, Shinichi Yano has written three small books that explain it very clearly, and applies them to sales and marketing strategies:

- Volume 1 – "The New Lanchester Strategy"
- Volume 2 – "Sales and Marketing Strategy for the Weak"
- Volume 3 – "Sales and Marketing Strategy for the Strong"

These books were written in 1990 (1995 in English). I highly recommend, if you want to understand how markets really work, that you buy these books (available at Amazon) and read them instead of screwing around with irritating social media advertising programs.

Now, I am sure someone will say that the Lanchester market share strategy is only appropriate for long lifecycle markets like toothpaste, laundry soap, or deodorant. But, that is an uninformed statement made by marketing novices. Lanchester's Laws apply to all markets, even markets like ours with short S-curves. Companies in our industry must push towards 26.1% market share in their niche as rapidly as possible, enjoy the profitability in that position for 7-10 years, and move to the next technology S-curve. We have seen this happen since the 1980's with great success, and no company ever achieved 41.7% market share (except MCG in VME processor cards, and maybe Greenspring in IndustryPacks mezzanine cards). So, the goal on short S-curves is 26.1% market share quickly, before the curve rolls over.



Since 1981, there have been about forty S-curves in the embedded board business, by my count. Some were great, some were good, and some were a waste of financial and engineering resources. The outcome of the embedded telecom board and systems markets shows how fast an S-curve can roll-over. No company ever achieved 41.7% market share before the business went to contract manufacturers. One of

## RAW - How This Embedded Board and Systems Business Works

them may have achieved 26.1% market share, but the cost-driven nature of the telecom board customers made their position unstable and they sold-out.

And, if you want to reduce your ignorance about how technologies move through markets, go buy the book "Diffusion of Innovations" by Everett Rogers and understand the Rogers model. This book will change your perspective on how technology markets work.

### Technology

Let's get right to the point here: what are the primary problems we face, technically speaking, in this embedded board and systems business? One is the tremendous disparity in the performance improvements between CPUs, memory, I/O, graphics, and communication chips. They all improve at different rates, at different times, out of phase, and that creates severe bottlenecks that add significant latencies in the architecture of the embedded computers we design. Out of those, the biggest problem we have is the I/O links. The semiconductor engineers who design this stuff have no clue about computer architectures, which is why we have so many bottlenecks. They are stuck in the von Neumann architecture model of memory, I/O, graphics, and CPU subsystems, hooked together by some communications interconnect. And each one of these connections robs us of performance.

How did we get here? From the 1940's to about 1993, all computers were CPU-bound: The I/O links could deliver more data than the processor could handle. Starting in the early 1990's, with the advent of GHz processors, all computers became I/O-bound: today's processor can now handle more data than the I/O links can deliver. While the semiconductor engineers have given us faster links, they have introduced horrendous electrical protocol latencies (i.e., 8b/10b), horrible connection and disconnection latencies, massive packet headers, and useless packet post-ambles.

If you map the connection latency, transmission latency, and disconnection latency of the fabrics, Ethernet, fiber channel, PCIe, and some of the FPGA interconnects in a 3-dimensional graph, you will immediately see that there are three distinct protocol categories: intimate, friend, and stranger. The low-overhead protocols are between the CPU and memory, and some of the FPGA links. These are "intimate" informal protocols for chips close together. The next level is InfiniBand and Ethernet with RDMA. This is a more formal protocol with some added latencies, but fairly efficient. Finally, we have the fabrics like PCIe, with huge protocol latencies. What we have here is an interconnect hierarchy.

When you look at this proximity-based interconnect hierarchy, the lowest protocol-latency interface should be between the CPU and memory, or between FPGAs and their resources. The next lowest-latency interface should be between the I/O chips and memory, and between two boards in a backplane. Then comes the high protocol-stack latency interfaces between computers on a network like Ethernet. Finally, the king of high protocol-stack latencies is the TCP/IP connections to the internet. Today we see a mix of these interfaces on the backplane: technological promiscuity if you will. The fabrics and the networking chips are problematic for performance-driven multiprocessor architectures.

Even the data center guys are recognizing this atrocious mismatch of I/O links to processing power. In January 2015, Amazon bought semi-maker Annapurna.<sup>1</sup> Annapurna is supposedly working on very efficient networking chips, to reduce the latencies between routers and servers on the network in the data center. They are moving up the interconnect hierarchy, to low-latency connections, exactly what we need to do with our board-to-board connections. And, you can be sure that these connections won't be copper-based. They have to be optical to eliminate all the hideous transmission line signal integrity problems that come with copper.

We are slaves to whatever these semiconductor guys give us, with the possible exception of the board vendors using FPGAs. Our present board-to-board connections are atrocious from a performance standpoint. With architectures like VPX, we are now deeply involved with heterogeneous multi-processor architectures, and the high-latency fabric links are killing us. Multiprocessor environments require very fast

---

<sup>1</sup> <http://www.extremetech.com/computing/198140-amazon-buys-secretive-chip-maker-annapurna-labs-for-350-million>

## RAW - How This Embedded Board and Systems Business Works

and efficient interprocessor communications mechanisms (IPC). All the present fabric and communications chips suck at this chore.

The reason we are I/O-bound is that we move the data around too much. Whenever data is moved, it has to go through the bottlenecks, the high latency paths in the system. There are efforts underway to enable the memory itself to do the processing, and that will eliminate moving the data constantly: Neuromorphic chips.<sup>2</sup> But, these solutions will take some time to develop and become accepted. Meanwhile, we are stuck with the chips coming from the semiconductor engineers, who only understand von Neumann architectures and continue to make the bottlenecks worse.

This leads to another observation: we have severe thermal (cooling) problems with embedded computers today. While this problem is most often associated with the increasing number of transistors on the die, I wonder if it's actually being created by moving the data around so much. I suspect that 25% to 50% of the heat generated by today's chips is created by shuffling the data from I/O, to memory, to cache, to CPU.

### Multiprocessing

Semiconductor engineers have no clue about how to design heterogeneous multiprocessor architectures. There are three basic multiprocessor architectures and they relate back to the intimate-friend-stranger protocol examples above:

- Tightly-coupled/shared-everything (TCSE): This is where every processor has direct access to every resource in the system. It operates with an "intimate protocol" that allows peer-to-peer connections with very little overhead. Rapid IO is the only fabric out there that can implement a TCSE multiprocessing architecture with its peer-to-peer capability. TCSEs are very deterministic and real time, although we could encounter some "Blocking Latencies" depending on how many processors share the resource and how often they access it.
- Snuggly-coupled/shared-something (SCSS): All of the processors share a mechanism to exchange messages with each other and stay synchronized. Typically, that is a section of memory (though it could also be a segment of a disk). InfiniBand (and Ethernet with RDMA) are both examples of an SCSS multiprocessing architecture. And, this is where cache coherency resides. I would feel more comfortable explaining quantum physics to you than cache coherency, even with my experiences from being heavily involved in the Futurebus MOESI model (later reduced to the MOSI model for obvious reasons), so just read this epistle and spare me the pain.<sup>3</sup>
- Loosely-coupled/shared-nothing (LCSN): This is where the present fabrics come into play: they must ASK the other processors for the data (a READ transaction followed by a WRITE transaction). This is the dreaded high-latency "Split Transaction" we ran into with Futurebus, and uses the "stranger" protocol. Originally created by Intel as Multibus II using their proprietary message passing coprocessor chips (MPC), PCIe is just message passing with more boogers on it. It sucked then and it sucks now.

When we hook multiple processors together, with the chips the semiconductor engineers give us today, we get a mess. This goes all the way back to the work done by Sequent, with their NUMA-Q architecture (Non-Uniform Memory Architecture-Quad). Sequent hooked-together four CPU chips with a cache coherent memory block shared between them on a single card (a Quad). Then, they hooked the quad processor cards together with a high-speed serial fabric on a backplane (the serial fabric connections were tied to the cache-coherent memory blocks). Everything works fine, until you try to access some data that is not in the local cache-coherent memory. When that happens, we endure the dreaded "Locked Transaction" we found in Futurebus, and latencies skyrocket. As you can see, this I/O-bound problem has been with us for a long time. Sequent built the NUMA-Q architecture back in the 1990's. This same Locked Transaction happens when we get a "cache-miss" in our present-day embedded systems.

Teaching the semiconductor guys non-von Neumann architecture is like teaching a pig to sing: it takes a great deal of time, and it generally annoys the pig. Just look at the fabrics out there and you'll see that data gets molested at least three times before it is processed by the CPU. Freeman Dyson, physicist and

<sup>2</sup> <http://semiengineering.com/neuromorphic-chip-biz-heats-up/>

<sup>3</sup> <http://semiengineering.com/heterogeneous-multi-core-headaches/>



## RAW - How This Embedded Board and Systems Business Works

mathematician, once said: "A good scientist is a person with original ideas. A good engineer is a person who makes a design that works with as few original ideas as possible."

The wheel was invented in ancient Sumer about 8500BC. What did civilization use it for? Children's toys, to pull around in the sands of Mesopotamia. It took until 5500BC, three thousand years later, for the engineers to put wheels on carts drawn by people or animals. That created the transportation and distribution systems we have today. It took another 7500 years, until the 1980s, for some engineer to finally put wheels on luggage.

I think Dyson is correct about the creativity of engineers. There's an old adage: "You can't DO things differently until you SEE things differently". The semiconductor engineers are still screwing around with the von Neumann model. On top of that, the embedded board and systems engineers are still screwing-around with copper, and neither of them can SEE things differently. DNA tests will probably prove that the semiconductor and board engineers today are direct genetic descendants of Sumerian engineers who invented the wheel. The answer to our problem is simple, if we continue to tolerate von Neumann architectures and copper traces: dump the message passing craziness, eliminate all the heavy protocol stacks, minimize the transmission latencies, and move-up the proximity-based interconnect hierarchy.

Just understand that there are billions of transistors in our embedded systems. They are all doing things right, but they are doing all the wrong things. They are molesting our data.

### Where do we go from here?

I've mentioned a few solutions to our problems above, that could mediate the von Neumann-based bottlenecks and eliminate a lot of the latencies associated with the present fabrics we are forced to use on copper. We must move up the interconnect hierarchy, to low-latency connections between the boards. My suggestions are based on Alderman's Laws of backplanes:

Law 1: when  $(f+P)$  is increasing, then  $(s/n)$  is decreasing

On copper traces, as the frequency ( $f$ ) of a link increases, and the electrical signaling protocol ( $P$ ) becomes more complex, the signal-to-noise ratio ( $s/n$ ) declines dramatically. The higher the frequency and the more complex the signaling protocol on copper, the more the signal looks like noise.

Law 2:  $2Bf=d/2$

Every time the frequency of the signals going through a copper connector and along the copper backplane traces doubles ( $2Bf$ ), the distance those signals can run intelligibly declines by 50% ( $d/2$ ). Otherwise, they look like noise again.

Law 3:  $2Nf=B/2$

Every time the network link frequency doubles ( $2Nf$ ), the demand for copper-based backplanes and their associated connectors declines by 50% ( $B/2$ ). When the network outperforms the backplane, there's no need for a backplane.

Law 4: if  $Nf > Bf$ , then  $2Nf = B/2$

If the network link frequency ( $Nf$ ) is greater than the backplane link frequency between the boards ( $Bf$ ), then backplanes lose their primary performance advantage. Today, we are running multiple links on copper traces to each board, to keep up with the network link bandwidth. When we run out of connector pins on the backplane for those additional copper links, Law 3 is invoked.

Understand that backplane-based computers are centralized systems. Each board in the rack is connected to the others through copper traces on a common backplane, and all the boards share a central power supply and chassis. Backplanes have two basic benefits: high performance data links between the boards, and modularity. Modularity, in turn, has two sub-benefits: maintainability and upgradeability. When the bandwidth of the network cable or optical fiber is greater than the bandwidth of the copper traces between the boards on the backplane, the boards in the rack will be broken-out into separate boxes, into a distributed system, and connected together with the network cable (Law 3). That's higher performance at a lower cost,

## RAW - How This Embedded Board and Systems Business Works

since the rack, the big power supply, and the backplane are more expensive than the smaller power supply and packaging for a single board (Law 4).

So, we can complacently sit here on this present von Neumann architecture and copper-trace technology curve, and wait for the next two or three copper-based evolutionary incremental improvements to occur, but our growth and profitability will suffer. Or we can move to low-latency optical interconnects on the backplane and avoid the effects of Alderman's Laws.

As Albert Einstein once said, "We cannot solve problems by using the same kind of thinking we used when we created them." If we accept Einstein as our guide, then we need to dump high-latency protocol stacks and move-up the interconnect hierarchy on copper. Then, we need to dump copper for optical. Finally, we need to dump von Neumann architectures for something like neuromorphic processing. Those are three S-curves that make sense when you consider the basic problems outlined here.

### Conclusions

1. Our overall market is not only a niche, but it is a collection of many smaller niches.
2. The only value worth adding in electronics is intellectual value. The other two are tenuous at best.
3. In niche markets, gross profit margin is much more important than volume.
4. No company in our overall market will ever reach 41.7% market share: The S-curves are too short and too many players will fragment the market. The best you can hope for is 41.7% market share in your smaller niche. All growth and profitability is in the niches.
5. von Neumann architectures worked when we were CPU-bound. They quit working when we became I/O-bound.
6. If we stay with the Von Neumann architecture and copper traces, we must move up the interconnect hierarchy to increase performance.
7. We are in the eye of the storm in heterogeneous multiprocessing. Our semiconductor suppliers have no clue how to build these kinds of systems, evident by all the latencies and bottlenecks they create in the interconnects.
8. When the last incremental improvement in copper bandwidth is attained, we must move to optical connections on backplanes. We are at that limit. Everything in the copper pipeline right now is a pig with lipstick.
9. At some point, we must process the data in place and quit moving it around. That means we must abandon von Neumann architectures and adopt neuromorphic chips or some other new computing architecture methodology.