========================
Semi-Conductor Landscape 2022
========================

The current execution and strategies of the players in the global silicon market has been undergoing drastic change in recent years. The rise of players such as Nvidia and Qualcomm, who at one time specialized solely in purpose built silicon, into complete players in multiple areas such as core processors, machine learning, robotics, and automation is nothing short of remarkable. The burgeoning chip market has opened opportunities for the entire computing industry from start-ups to established players. It is truly an exciting time to be part of the semi-conductor industry.

The seismic shift in mode of execution of global silicon players at this point in time largely revolves around the increased capabilities in silicon customization and the ability of a larger array of players to provide solutions that address a growing variety of customer needs. Often these needs are outside the bounds of traditional PC, server, and laptop implementations where Intel and AMD still tend to dominate.

One of the key initial drivers of today’s altered landscape was mobile computing and the smart phone revolution and along with this a shift to more power thrifty RISC based processors (battery life becomes a key feature in mobile). This left a large opportunity for traditional embedded players like Arm (and others) to fill the void in a burgeoning market. Not only are the approaches to chip architecture quite different, but the business approaches of companies like Arm and more traditional silicon players like Intel are quite different as well.

While Intel actually produces silicon in the form of chips and cores as a sort of vertical, Arm does not produce chips at all. But rather, it produces and licenses the core Intellectual Property (IP) that other organizations, partners, and even competitors can modify/specialize and then typically have fabricated (fabbed) by yet another party (a good example of this is the Qualcomm Snapdragon).

These specialized designs span the market from small IoT devices all the way to servers in cloud providers(e.g. Amazon Graviton). Arm certainly was not the only player to produce their own RISC instruction set but as it turned out its solutions, business model, and technology areas of expertise were certainly right place - right time.

Furthermore, today even traditional software companies such as Google, Amazon, and Facebook are now engaging in efforts to design their own chips following paths similar to the Apple M1 model. In most of these cases the flexibility of the Arm business and architecture models allow for varying degrees of customization. This also provides a growing market for software companies and support services (from CAD tools, to operating systems, to applications). For some markets the older model of a complete vertical solution provided by companies like Intel serves them well and for others the more flexible model is required.

Increased flexibility and cost in the IoT space has brought even more change as a legitimate force in Open Source IP has emerged in RISC-V. While Open Source is not really a new concept in the semi-conductor IP space --Opencores and other architectures have provided some degree of Open Source over the years-- RISC-V may have a market opportunity today in IoT and low cost devices similar to Arm’s earlier opportunity in mobile.

Generally like Open Source software Open Source IP will provide some of the benefits for customers with increased levels of customization, potentially lower cost, and increased reliability due to a larger transparent community of users sharing ideas and solutions. On the flip side for silicon providers this may mean an even greater focus on value-added services, solutions, and customizations as the changes in licensing may impact the older business model.

Of course building great silicon is one thing but being able to provide the features and services the chips are capable of is another. This is where the operating system comes in and, generally speaking, it is the heart of the hardware-software interface. In the abstract the operating system can be seen in two pieces one which has direct access to the hardware and its facilities through registers, drivers, mapping, interrupts and the like and the other which provides the services of the hardware and the system to applications via software interfaces, libraries, and other system specific software.

The relationship of the operating system and the silicon platforms is crucial and symbiotic without it the capabilities and features of the overall system cannot be provided to users. At the very core of the kernel are platform specific directories that contain specific code for each silicon family often times written in assembly –in effect the operating system kernel intimately knows the instruction set and the internals of the hardware.

Having a solid and reliable operating system is a key to success in the computing world today. Linux is part of the larger UNIX family and its design has been validated through the test of time. A large part of the strength of Linux and its family has been the Open Source model. Having worked in the world of embedded proprietary RTOSs earlier in my career I saw first hand the drawbacks to being reliant on a single vendor for support. And even though it seemed counter intuitive at the time, having a large development community of support on the internet proved far more capable of answering questions and helping to solve problems than phone tag with a call center. At the time I was working in telecom and often advocated for Linux, but just bringing up Linux as a replacement for an RTOS was difficult then. Sometimes you have to persist.

My experience with Linux distributions goes far back, in the mid 1990s I began as a Slackware user and then later moved to Debian. Those earlier distributions required a large amount of user effort and were often accompanied by problematic upgrades, hardware anomalies, and etc. One of the key reasons for my switch to Ubuntu was I no longer had the time for all-nighters working on getting the latest release of Debian to work. Ubuntu brought most of the problems to an end and for me this is a lasting lesson in making things easier and as seamless as possible for the customer/end user. And I continue to be an Ubuntu user to this day (this test is being written on my Ubuntu machine). At least in my circles Ubuntu is well regarded as a great and easy to use desktop and server system.

One arena that I have seen less Ubuntu presence is in the embedded space along with specialized silicon. For a long time Red Hat and other Linux implementations seemed to have had a larger piece in that domain. The embedded space is often where the first line of development occurs for some of the silicon providers and development of Ubuntu in that space ported early to the platforms in conjunction with the silicon providers might drive greater presence in embedded --this would seem important especially for IoT, automation, and machine learning in my view. Moreover, getting buy in at this level should provide opportunities for further growth as chip families tend to spread out and users find multiple ways to apply technologies (as mentioned earlier). This is an area where my experience runs deep and leading Canonical in an area like this would be a natural fit.

My experience in technology is broad across many domains from the chip level to top of the stack. I have been able to smoothly transition from a company like Akamai working on CDN (SAAS) for mobile, to Arm at the semi-conductor level. Product leadership has been oriented around delivering the right mix of products, ensuring that development teams get the resources they need, and priorities are right-sized to deliver what is expected.

I consider partnerships and alliances as keys to business development particularly when focusing on silicon and the semi-conductor ecosystems. Working hand in glove with the silicon players is crucial in delivering the power of the silicon to users and their applications. Getting in early and developing the mutually beneficial relationships and partnerships is extremely important. When I was in telecom I was on the customer side working with silicon vendors and software and operating system vendors. Most of the projects were developed together, delivering those products would have been nearly impossible without strong partnerships, good working relationships, and shared knowledge.

I have been very fortunate in my career so far to have been afforded the opportunities to work in many different domains from startups to large corporations in a wide variety of different roles. And as mentioned earlier this has included a range of technical domains from the semi-conductor level to SAAS along with roles from contributor to management. Along the way I have been able to work with both proprietary and open systems and engage with different parts of the business with customers, partners, and vendors. For me personally this has required the ability to ramp up fast, shift gears rapidly, and be able to deal with change. And that has allowed me to have a career that has so far provided me with continuous learning, interesting challenges, and never a boring day.

(Reader Note: This document is strictly my own work and I am solely responsible for its content. And it was done without references to any other material. Any similarity to other ideas or texts is coincidental)