The Thesis Committee for Venkatramanan Jayatheerthan
Certifies that this is the approved version of the following thesis:

Software for Chip Companies: An Analysis and Strategies to Build Software IP

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: ______________________________
Steven Nichols

____
Anthony P Ambler

Software for Chip Companies: An Analysis and Strategies to Build Software IP

by

Venkatramanan Jayatheerthan B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2011

Abstract

Software for Chip Companies: An Analysis and Strategies to Build Software IP

Venkatramanan Jayatheerthan, M.S.E

The University of Texas at Austin, 2011

Supervisor: Steven Nichols

Software plays an important role in making products usable. We couldn't imagine a laptop without software that run it making the things it does possible with the laptop hardware. Software has penetrated into several industries making significant contribution in how the products are designed and to make them more usable. This thesis focuses on semiconductor industry and analyzes the role played by software to enhance their products and differentiate them from competition. In this context, the thesis looks at acquisition of software companies by chip companies and analyzes them to determine the benefits and how it changed the market space.

In a semiconductor company, the focus is predominantly on hardware. Although software is equally crucial to the success of the product, not much focus is placed on it in terms of innovation and building sustained software IP portfolio. One of the questions that this thesis tries to answer is how to build a robust software IP portfolio in a chip company. Case studies of different products were conducted to analyze their IP building strategies in general and focusing specifically on software patenting in terms number of patents filed and procedures adopted to encourage it. It looks closely at the best and not-so-best practices adopted by the teams and analyzing them to determine why certain initiatives succeeded while others failed.

A crucial aspect of building software IP pipeline is to involve junior level engineers in this process. The thesis looks at some of the strategies companies could use to bring the culture of patents to the lowest levels of engineers. Typically the senior engineers are well tuned in to the process and regularly file patents while the junior engineers don't. This is crucial to the company since today's junior engineer is tomorrow's senior engineer leading technology initiatives. The thesis concludes by putting forward recommendations to encourage software patenting.

Table of Contents

List of Figures

Chapter 1: Introduction

From the days of MS-DOS in late 1970's to Windows 7 and MAC OS X Snow Leopard, software has made tremendous strides in bringing out the real potential of products. Software use in products is widely prevalent these days from vending machines to TVs and to medical devices. Even small devices like an Infrared to USB dongle runs software in it. It's hard to imagine products without software. It has become an integral part of the product improving its usability while bringing out its full potential.

Over the past decades, the capabilities of both hardware and software have improved so much to the extent that things that were thought to be not feasible even 10 years back are feasible now and better than what had been thought. The devices have become faster, smaller and more capable. For example, today's cell phone has better hardware specs than a PC 6 or 7 years ago. Yet, these devices need software more than ever to enable wider adoption of devices.

For instance, software played a crucial role in adoption of Internet as a dominant communication platform that it is today. Although hardware is needed to send and receive raw data between remote computers, it's the software's ability to make information out of it that led to the explosive growth of Internet bringing out its real potential. With applications like email, web servers, secure transactions, access to remote databases and VoIP, to name a few, software has created several uses of the basic communication infrastructure that was made possible by advancement in hardware technologies.

There are several instances where expensive labs, costing several million dollars, are being utilized 24 hours a day. It's used by US employees from multiple US sites during the US daytime. During US night time, it's used by India employees. Anything that can be done locally in the lab is doable from remote access. This is made possible by the communications infrastructure and the equipment made specifically to be usable from remote using software. This doubles the productivity of resources. Internet technology has allowed such effective usage of resources.

The software allows easy upgrade of products. Given a product with a certain set of hardware features, the usability of it can be improved by upgrading the software running in it. In a real life example, Bose Corporation upgraded the software that run in their Lifestyle home theater systems couple of years back. Being a Bose product, the quality of audio is exceptional, before and after the upgrade. However, the ease of use of the system has increased dramatically after the upgrade. This system has a feature to store music discs. Before the upgrade, only regular audio disc, those that would be playable in a CD player, could be stored although MP3 format songs could be played. This made storing discs a cumbersome process as only less number of songs fit in a regular format disc requiring more discs to be made. After the upgrade, the system supports storing MP3 discs. The user interface was also re-designed to organize songs based on albums. This made it lot easier to retrieve and play songs of an album. In this case, the underlying hardware – the speakers, sub-woofer and the DVD player remained the same. But a software update immensely helped enjoy the system more.

These examples show that software is an integral part of a system. It is an essential part that increases the usability and configurability bringing out the full potential of the system.

Chapter 2: Products of Chip Companies

Chip companies make different types of processors and Integrated Circuits that are used as building blocks of a product. These chips vary widely in complexity and usage. It can be as simple as a chip that controls an LED bulb or as complex as a processor in high availability server. Depending upon the complexity of the chip the software support needed to integrate it to the rest of the system varies. Software support for a simple chip would be straight forward. On the contrary, handling an ADSL communication chip that does the low level Internet communication or a general purpose processor like Intel Core i7 would be hugely complex requiring different sets of software tools and driver support to use it effectively. These processors and chips have become an integral part of people's lives. They offer great deal of advantages over traditional techniques which they replaced. Managing a resource using chips allows finer control and accurate management. It also allows quicker response time to take corrective actions.

There are several examples where hardware electronics in combination with software are replacing conventional system for better. From agriculture to defense to medical equipment to entertainment to telecom, electronic systems are being used more than ever. The complexity of these chips has been increasing over the years as they become smaller, faster and better.

AUTOMOBILES

A good example would be the use of electronics to control different aspects in a car. Even 20 years back, electronics was used mainly for in-car entertainment like radio, cassette and CD player. Today, electronics are being used extensively in what is called "drive-by-wire" design. Fuller says a car with this system relies mainly on electronics to control wide range of vehicular operations like steering, acceleration and braking, to name a few. Conventional cars mainly use Mechanical and Hydraulic systems to achieve these operations. Several components like brake booster, master cylinder, steering column, steering shaft, etc form components of the conventional system. These occupy more space in the car, are inefficient and are prone to wear and tear over the years. (Fuller 2009)

The car manufacturers have been integrating computers and electronics into cars. Fuller also says, "If the drivers get accustomed to the idea of drive-by-wire, it can increase comfort, functionality and safety" (Fuller 2009). Throttle-by-wire is the type of drive-by-wire system where the acceleration is managed by electronics. These systems use a pedal unit and an engine management system. In cars with these systems, the pressing of accelerator is just a way to tell the electronics how fast to accelerate. The pedal uses sensors that measure how much or how little the driver moves the accelerator, and the sensors send that information to the engine management system. The engine management system is a computer that, among other tasks, determines how much fuel is required, and it provides this input to an actuator - a device that converts energy into mechanical motion converting the action of pressing the accelerator pedal to wheels rotating fast. In this process, software receives information from different sensors, determines what to do and sends commands to the respective units to take action.

TELEVISIONS

The use of electronics, with software managing them, has provided significant advantage over how televisions are designed and manufactured these days. With the Digital televisions, the inside of the television has completely changed over last decade or so. What used to be a bunch of electrical and electronics components that decode and display the analog signal has changed to running a modern Open Source Operating System like Linux. In a real life example, the author was surprised to find Linux GPL license agreement in the manual when buying a Sony LCD HDTV. This shows how far software has penetrated into a traditionally hardware dominated system.

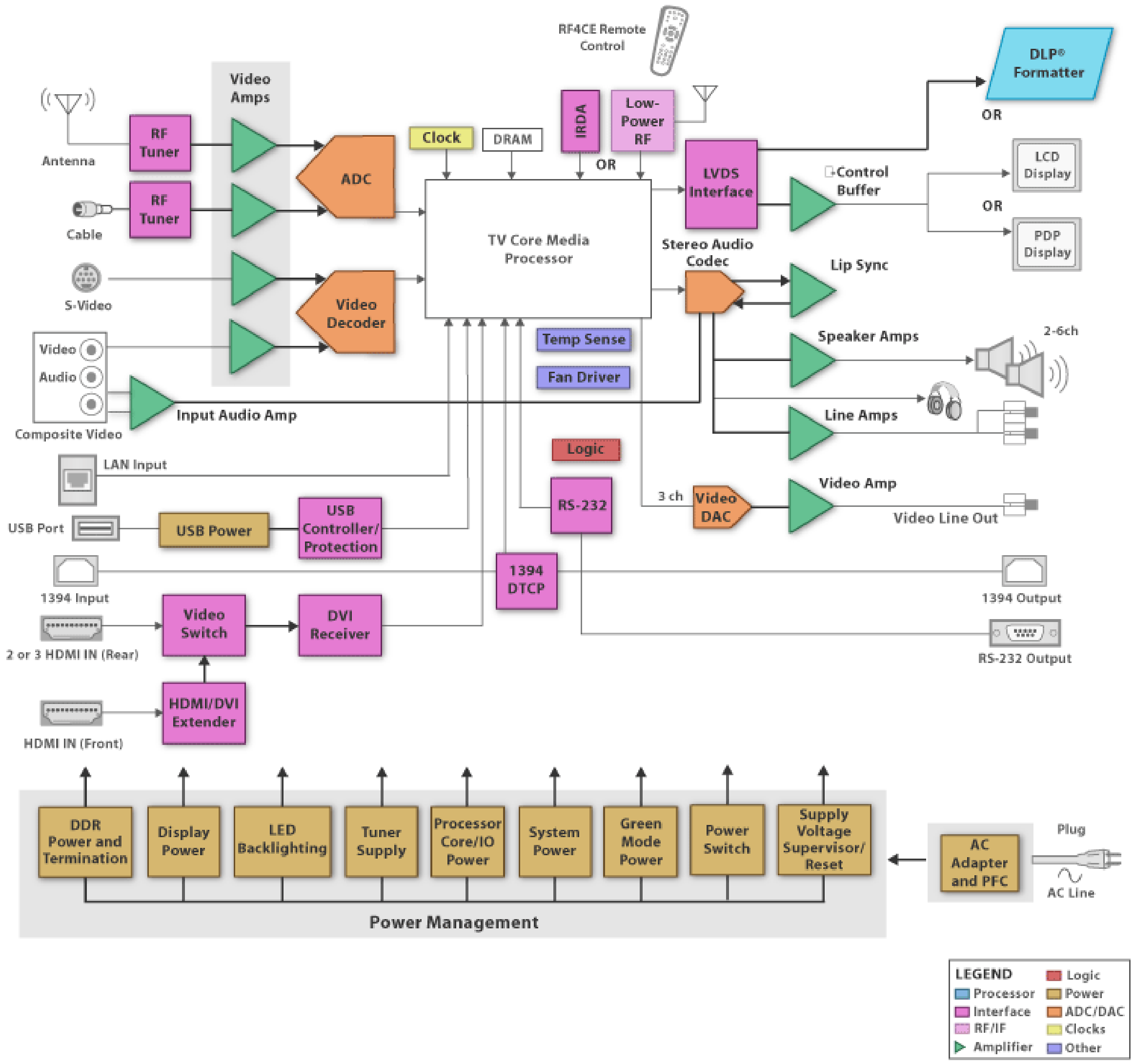

Figure 1: Block Diagram of a Digital Television. © ti.com

A general block diagram of a digital television is given above. It contains a Core Media Processor (ti.com n.d.) at the heart of the system handling configuration and management functions. This processor runs a standard operating system like Linux. This processor is connected to different components that make up the television like the LCD/LED display that actually shows the picture, the audio management components that reproduces the sound, the remote control management component and several such components. Driver software is written in Linux to configure and manage these devices. Whenever the user changes a TV setting, the Linux operating system using the driver software "implements" the change by configuring the device with the new setting. This makes the device to operate as per the new configuration.

Chapter 3: Advantages of Software based Solutions

There are several advantages in using software to manage the hardware instead of just conventional hardware based solution.

DEVELOPMENT OF COMMON PLATFORMS

In case of televisions, when the first version of the product line is being developed software is written to work in that platform. If a different model of the television is to be developed, the base platform can be modified to include the differentiating features like LED instead of LCD display management components. In this case the software needs to be changed to support the difference of features. The rest of the features are already supported. This incremental change allows faster product development cycle reducing the time to market.

BETTER USER INTERFACE

Using software provides the system with better user interfaces. Comparing the televisions 15 years ago with ones we have now clearly shows that using software based interfaces bring about positive changes to interfaces. They are much cleaner and easier to use. Another example is provided earlier with home theater system where a software upgrade made the system more useful than before although the hardware remained the same.

LEVERAGE VENDOR SOFTWARE

Using software based solutions also allows software developed by the vendor to be used into the product development. For instance, the component vendor provides the driver software for that component to integrate into the system. The customer can use this software in their product allowing faster development of the product. Since the driver software is developed by the vendor, it is more likely to be stable as it would have been verified before releasing and multiple customers would be using them. Also, if the operating system software is standard like Linux, it enhances reusability.

EASY UPGRADABILITY

Software is inherently easier to upgrade compared to hardware. Most of the time, it is just a matter of sending a CD or downloading latest version from web and upgrading it. On the contrary upgrading hardware is tougher and costs more. When developing a device, the development team can put together all the hardware components needed. The first release of the software may not make use of all the hardware components available. The software can then be upgraded incrementally to effectively make use of all the components. This provides a stable hardware platform and a software platform that gets better and better over time. This also future proof's the customer's investment. They know that the hardware they buy today is not going to be out dated. This strategy is used in premium audio systems that are expensive to replace every few years.

Chapter 4: Software for Chip Products

In general chip products are not the end products by themselves. They are used to build other products that eventually end up with the customer. For example, MP3 players with storage use NAND flash chip. The flash chip is a product provided by several vendors. But it is not used by the consumers directly. It is practically useless to the end user. In turn, it is used by MP3 player makers to store music. The final product is used by their customers.

In such chip products, there is at least one vendor between the chip and the “real” customer. This means that somebody has to use this chip to build a product and then sell it to make money. Only when the first level vendor makes money from “real” customers do the second level vendor makes money. This also means that the company using the chip should be able to integrate the chip into their product easily. The basic idea is to make sure the first level customer makes money. Then the chip vendor makes money automatically. Anything that helps achieve this objective needs to be done by the chip vendor.

Software support for the chip helps its user in several ways. It enables easy integration of the chip into the product thereby reducing the cost involved in using the chip. The level of software support depends on the type of the chip. For example, the NAND chip mentioned earlier for the MP3 player requires driver software for standard operating systems. This software if provided by the chip maker, allows the MP3 player maker to readily use the chip in their hardware and manage the chip from the operating system software like Linux.

In case of a general purpose processor, the software and tools required is entirely different. The processor needs operating system software support. The processor vendor needs to provide all software and tools needed to achieve that. To be able to do that, several different kinds of tools are needed. First the standard operating system software needs to be extended to support this processor so the customers can use that operating system to run their application. Tools will be needed to build and debug applications that can run on the processor. This includes compilers and debuggers. There are standard compilers and debuggers available from GNU. The vendor of the processor needs to provide extensions to these tools. These tools allow the user of the processor to build the applications and verify them. There is no clear division of labor between the chip company and the operating system software company. Sometimes, the processor support is provided by the operating system software company. It just depends on how popular the processor is and the needs of the market.

Most of the times, the processor is marketed even before it is available for delivery. If there is a software tool that can engage the customer before the chip is available, it is a highly useful approach. The vendors of processor often provide software called Simulator that behaves like the real processor only it "simulates" the behavior in software running on a Linux/Windows/Solaris host machine. This allows the customer to work on their applications and verify it while the processor is being built. When the processor is available, it will be delivered to the customer at which point they will build the hardware using the processor and verify the applications on the real hardware. It's a win-win situation for both the vendor and the customer.

As it is evident here, the software support needed to "push" a chip varies widely depending on what the chip does. The chip vendor needs to take different approach depending upon the chip being developed. Let's take an example of LSI's Network Processor and look at the suite of tools provided with the chip.

LSI'S NETWORK PROCESSOR

LSI designs and markets a network processor targeted at wireless and wire line markets. It is used by communication equipment makers to build equipments used for Internet, Cell Phone towers and Ethernet switches to name a few. The network processor is used to do packet processing at real-time speeds allowing sending and receiving packets at high speeds. It has hardware accelerators that can inspect the packet data, match patterns and forward them, do traffic management and security. The chip also has third party general purpose processor that can be used to run applications.

This product line is available for several years. Currently there are several customers that use this chip in their products. Apart from the physical chip, the customer is provided with detailed set of documentation that gives all the information needed to configure and use the chip and build their hardware board. The customer also has access to the support team that helps them solve issues that they run into when they build their system. Any query regarding the product is fielded by the support team. The support team contacts the hardware and/or software team if more information is needed or the query is specific and needs expert guidelines.

Software and Tools Provided

The chip is not complete without the software and tools that are provided with it. There is a rich suite of tools that is provided with the network processor that makes using the chip a lot easier. In fact, these tools are a necessity since such a complex chip cannot be used effectively with just documentation and support team.

The tools address different aspects of the software support. The Hardware Configuration GUI shows different features of the chip and helps the user to configure them. The runtime software loads the chip configuration at start up and configures and queries the chip at runtime. The Advanced Simulation environment simulates the chip which allows application development when the customer's hardware platform is not ready. The support is provided for leading operating system software's in the market, like Linux and VxWorks. This allows the customer to re-use this so they can port it to their platform easily. LSI also provides Application Development Kit that targets software for a specific market segment – like wireless, DSLAM, etc. It implements protocol specific features using the features available in the chip.

Advanced Simulation Environment

This is used to simulate the features of the chip on a standard Linux, Solaris or Windows machine. Once this software is installed, the user can configure the different features available in the chip and simulate them. This tool allows the customer's software team to continue developing their application while the hardware team is building a board with this chip in it. This enables parallel development for the customer thereby reducing the time taken to build the end product.

When running applications, the simulator provides a detailed set of data that helps the user debug their application. It simulates all the modules available in the hardware and provides a view of the chip that is consistent with the hardware. The customer can write their applications for all of the modules and verify them. Once the applications are developed and verified, they can be easily used on the real chip when it is available.

Hardware Configuration GUI

The Hardware Configuration GUI is the primary tool used by the customer to view the different features of the chip and configure them as needed by the customer. It brings out all the features supported by the chip and lets the user enable and configure the features needed by the customer and disable the ones not needed by the customer. A configuration file is generated which is then loaded using the runtime software on to the chip or the simulator.

The hardware configuration also verifies that all the features are configured properly and no information is missing. Also, if there is any interdependency of a feature on another feature, it is detected and prompts the user to configure the dependent feature. The chip has several modules that can run scripts to do their job. This GUI provides a single interface to write and build those scripts. These scripts if defined are packaged into the configuration file. Once a configuration file is generated from this GUI, it is assured that the features needed are rightly configured and will work as expected when loaded on the chip.

Runtime Software

The runtime software manages and configures the chip at runtime. The initial configuration of the chip is received as a configuration file generated by the hardware configuration GUI. When it is loaded, the runtime software configures the different blocks of the chip with the features enabled by the user and disables the ones not needed. This software knows the internal details on how to configure the features in the hardware.

The runtime software also provides programming interface that the customer's application can use to change the configuration of the chip at runtime. As the initial configuration is only the starting point, the configuration of the chip changes over the period of time as the product is used in the field. This software is used to accomplish the runtime changes needed by the application. Needless to say, it is used extensively by the customer. This software is developed to allow support for multiple operating systems of customer's choice. Another feature of the runtime software is it can be used for both hardware systems and simulator systems. The interface used by the application is same while it is built for different platforms. This allows fully integrated testing of the application on the simulator. When the hardware is ready the same application can be built for the hardware platform and used.

Operating System Support

Apart from the software and tools mentioned above, support is provided for operating systems that are targeted for this chip keeping in mind the customers being pursued. This serves two purposes. First, it helps in-house validation of the chip's features which is a necessary step before it is sold to customers. Secondly, it serves as a reference implementation that the customer can use for their system. The customer can compare their system with this reference system and get information on how the board needs to be designed and how the driver software needs to be written. This helps cut down development time for the customer.

Application Development Kit

This software provides industry and network protocol specific programming interfaces to the customer. This piece of software is built upon the runtime software to configure the chip while exposing programming interfaces that are specific to a network protocol or a specific market segment. For example, there are interfaces for wireless protocols. While the runtime software is based on the chip features, this software provides full abstraction of chip related details and exposes industry standard objects and interfaces. This makes it easier for the customer to develop the application product.

This layer hides the chip specific details and complexity so the customer doesn't have to fully understand the chip implementation to use it. The customer can just start using the software and the chip based on their understanding of the market segment and technology they are operating on.

Demos and Samples

The samples and demos for most widely used scenarios are also provided. It illustrates frequently used features of the chip how it needs to be configured and used. This provides the customer with a ready-made reference for using a feature. These demos are available both for hardware platform and simulator. So when the hardware is being built, the demo can be used on the simulator to better understand the feature.

Chapter 5: Semiconductor Industry Trends

Given the importance of software for chip companies, it would be interesting to see if this translates into note worthy events in the semiconductor industry. There has been a lot of activity in past few years that point to gaining realization of the importance of software. The chip companies have been actively seeking to fill the gaps in product offerings to customers. There are trends that suggest chip companies are ramping up on software support they provide for their products. This trend has been gaining momentum in non-PC space like mobile computing and tablets. This ramp up by hardware companies blurs the demarcation between hardware and software companies. This allows the companies to play both the hardware and software side of the end product there by providing more of a solution to the customer as opposed to a piece of the puzzle.

One of the means used by chip companies to acquire software expertise is through acquisitions. Past few years has seen hardware companies acquiring software companies. Some examples include acquisition of leading software maker for embedded devices, Wind River Systems by Intel in 2009. This deal pushes Intel deeper into the market for wide range of non-PC machines. (Kharif 2009). Another example is the acquisition of embedded Linux software vendor MontaVista Software by Cavium Networks, a leading provider of processors for communication equipments.

ACQUISITION OF WIND RIVER SYSTEMS BY INTEL

Intel entered into a definitive agreement to acquire Wind River Systems in June 2009 paying $884 million, in cash, to buy all of the outstanding shares. (intel.com 2009) Wind River is a leader in providing software for embedded devices, and will become part of Intel's strategy to grow its processor and software presence outside the traditional PC and server market segments into embedded systems and mobile handheld devices. Wind River has become a wholly owned subsidiary of Intel continuing with its current business model of supplying leading-edge products and services to its customers worldwide. (intel.com 2009).

"This acquisition will bring us complementary, market-leading software assets and an incredibly talented group of people to help us continue to grow our embedded systems and mobile device capabilities," said Renee James, Intel vice president and general manager of the company’s Software and Services Group. This acquisition will boost Wind River’s sales for Intel based solutions. (Thomas, Intel to Acquire Wind River Systems for Approximately $884 Million 2009).

Wind River Facts

Wind River is a leading provider of software for embedded and mobile devices. It has been pioneering computing in embedded devices since 1981. Its technology is used in more than 500 million non-PC devices. (windriver.com n.d.). It develops operating systems, middleware (software between an OS and software application), and software design tools for a variety of embedded computing systems. Its main products include

VxWorks, the market-leading proprietary and multi core ready real-time operating system, and commercial-grade Linux software platforms. Wind River has more than 1,600 employees and operations in more than 15 countries. During its fiscal year ended Jan. 31, 2009, Wind River reported annual revenues of $359.7 million. (Thomas, Intel to Acquire Wind River Systems for Approximately $884 Million 2009).

Intel Facts

Intel designs, manufactures, and sells integrated circuits for computing and communications industries worldwide. It offers microprocessor products used in laptops, desktops, servers, storage products, embedded applications, communications products, consumer electronics devices, and handhelds. The company also offers system on chip products that integrate its core processing functionalities with other system components, such as graphics, audio, and video, onto a single chip. Intel also provides software products, including operating systems, middleware, and tools used to develop, run, and manage applications in their processors. The company was founded in 1968 and is based in Santa Clara, California. (finance.yahoo.com n.d.)

Intel is the largest maker of computer chips. In Q4 2009, Intel controlled 80.6% of the market share for microprocessors (Sherr 2010). It made yearly revenue of $37.6 billion last year out of which a small fraction of $1 billion came from embedded devices. (Kharif 2009).

Analysis of Acquisition

New Markets

The sales of non-PC computing—a category that excludes smart phones, netbooks, and traditional computers—generated $900 billion in revenues in 2007, according to consulting firm IDC. That makes it four times larger than the market for PCs and it's growing almost twice as fast. (Kharif 2009). The PC markets have saturated and Intel already has dominant market share in it. Intel has very little penetration in non-PC segments where computing processors go into car entertainment systems, TVs, mobile phones, defense, aerospace and other sectors. These sectors largely bypass Intel products and use ARM, Freescale and other processors that are more suited for embedded systems. As computer technology migrates into cell phones, cars and other places you don't often find Intel chips, Intel feels more pressure than ever to diversify. (Rogoway 2009)

Looking at the size of the non-PC market and the amount Intel is making from it, there is a huge upside potential for penetration. In these markets, the hardware and software are tightly integrated and having a partner who is entrenched extensively in that market is certainly going to help Intel. While Wind River's annual revenue of $360 million is small compared with Intel's $38 billion, Intel plays in a much narrower market segment than Wind River. Wind River serves customers in aviation, automotive, consumer electronics, industrial, communications, medical and defense. Among its customers are the biggest OEMs in these markets, including Alcatel-Lucent, Boeing, LM

Ericsson, Lockheed Martin, Motorola, Northrop Grumman and Raytheon. These are companies that Intel will now have a strong foothold to push its products. (Ojo 2009).

Intel and Wind River are taking steps to penetrate into non-PC markets. Wind River updated its flagship OS VxWorks 6.8 to allow increased multi-core processor support for Intel® Core™ i7 processor. (Thomas, Wind River VxWorks 6.8 Broadens Multicore Capabilities and Processor Support 2010). Wind River updated its Hypervisor product to support latest Intel® micro-architecture codename Nehalem-based processors, such as the Intel® Xeon® processor 5500 series as well as Intel® Core™ i5 processor and Intel Core™ i7 processor utilizing advanced virtualization hardware assist capabilities. (Thomas, Wind River Extends Virtualization Support With New Release of Wind River Hypervisor 2010). These and similar support that Wind River is adding to their software is taking Intel deeper into non-PC market.

Monetizing Software

Although Intel is world leader in making microprocessors, it has dabbled in software through its software-and-services division for the past 15 years. The unit has never made money and exists mainly to support sales of related chips (Kharif 2009). During the 19 years that Grove was president, then chief executive, Intel established itself as the world's preeminent computer hardware company. Even so, Intel has repeatedly sought to branch out in software as a complement to its chips and as a tool for breaking into new technologies. "The results have been very consistent," Grove says now 73 and retired. "They amounted to nothing." Renee James, vice president leading the software group of over 5000, is trying to break that losing streak. To do that, she's taking Intel into uncomfortable territory. The new mobile operating system puts Intel in competition with Microsoft, its oldest and most dependable partner. Intel's software group has taken on a support role at other companies, like helping DreamWorks, transfer its animation software to Intel technology (Rogoway 2009).

"We know how to build and deliver software profitably," says John Bruggeman, chief marketing officer at Wind River, the world's largest maker of non-PC computing software. (Kharif 2009). Wind River has been monetizing software products and services by selling to several top tier customers. Acquiring them was a natural choice for Intel to pursue their software strategy.

Partners become Competitors

Wind River has big presence in mobile operating systems especially in Andriod, the mobile operating system promoted by Google. Both Intel and Wind River is part of the alliance and Wind River is closely involved in designing about one-third of the 18 Android based mobile phones expected to be released. With this deal, this engagement will only grow. (Kharif 2009).

This is in direct competition with Microsoft which has been a most trusted and dependable partner for Intel for more than two decades in PC market. Microsoft offers its own operating system for mobile devices called Windows Phone 7. It has not had much success losing market share to Android and Apple. The "Wintel" partnership that dominated PCs doesn’t seem like it will continue in mobile platforms. With Intel moving deeper into software, it is directly in competition with Microsoft.

Competitors become Partners

Wind River software products provide support for wide ranging hardware platforms. The solutions are hardware-agnostic and support Intel, Freescale, PowerPC, MIPS, ARM, Cavium and other platforms. (windriver.com 2010). Being a wholly owned subsidiary of Intel, it will be interesting to see how much resources are being spent on non-Intel platforms. Ken Klein, President of Wind River, adds in his blog: Since the close of the acquisition, we’ve publicly committed to future hardware roadmaps with long-standing partners Cavium and Freescale. We’ve also received public validation from Cavium, MIPS Technologies, RMI, and customers such as Huawei as part of our Carrier Grade Linux leadership. (Klien 2009).

If Intel can offer Wind River software with its chips at a reasonable price, "there'll be few companies that won't buy it [from Intel]," says Richard Williams, an analyst at Cross Research. That poses a potential threat to rivals such as Freescale Semiconductor, Renesas Technology, and Texas Instruments (Kharif 2009). While it has to promote its products with Wind River, Intel also needs to assure Wind River’s customers of the continued support for other platforms. The companies that were competitors now become partners in a way.

ACQUISITION OF MONTAVISTA SOFTWARE BY CAVIUM NETWORKS

Cavium Networks, a leading provider of highly integrated semiconductor products that enable intelligent processing for networking, wireless, storage and video applications, announced in November 2009 that it is has signed a definitive agreement to acquire MontaVista Software for $50 million. This acquisition will significantly increase Cavium's software and services revenue. (caviumnetworks.com 2009)

MontaVista Software Facts

MontaVista Software is a leader in multi-core embedded Linux operating systems, virtualization, development tools and professional services. It helps customers get the most out of open source by providing commercial quality Linux and developer tools that significantly reduce the time and effort required to deliver commercial products to market. There are over 50 million devices in the market today running on MontaVista Linux. (mvista.com n.d.). MontaVista provides Carrier Grade Linux that has been widely adopted by leading companies networking and consumer electronics companies like Alcatel-Lucent, Cisco and Ericsson, Sony, Samsung and Philips. (mvista.com 2009)

Cavium Networks Facts

Cavium Networks is a leading provider of highly integrated semiconductor products that enable intelligent processing in storage, networking, communications and security applications. Cavium Networks offers a broad portfolio of integrated, software compatible processors ranging in performance from 10 Megabits to 20 Gigabits per second that enable secure, intelligent functionality in enterprise, data-center, broadband/consumer and access & service provider equipment. Cavium Networks processors are supported by ecosystem partners that provide operating systems, tool support, reference designs and other services. (caviumnetworks.com 2009).

Analysis of Acquisition

Complete Solution

Cavium has been partnering with software companies to provide operating system and tools support to their processors. It has been working with ENEA, Wind River, MontaVista, etc to provide support in multiple different operating systems (caviumnetworks.com n.d.). This allowed their customers to get both software and hardware support. However, with this acquisition, Cavium can provide complete solutions to embedded device manufacturers (caviumnetworks.com 2009).

"Software is becoming an increasingly important part of the total solution with the rapidly increasing adoption of multi-core processors," said Syed Ali, President and CEO of Cavium Networks. "This acquisition will complement Cavium's market leading processor portfolio with world-class software expertise and will enable us to deliver highly compelling and differentiated solutions to the market." "Embedded Linux is poised for rapid growth," said Rusty Harris, President and CEO, MontaVista Software. "By becoming part of Cavium Networks, MontaVista can confidently continue to offer industry leading commercial grade embedded Linux, support and services to our customers and partners.” (caviumnetworks.com 2009). This allows Cavium to leverage the software support and claim a bigger pie in the stake.

Software Services

Apart from supporting Cavium products, MontaVista also supports other hardware platforms like ARM, Freescale, IBM, MIPS and AMCC. (caviumnetworks.com n.d.). MontaVista will operate as a wholly owned subsidiary under Cavium. This allows independence, to some extent, that MontaVista needs to support competing hardware platforms. One of the reasons why Cavium bought MontaVista is to build a profitable software and services business (caviumnetworks.com 2009). This will allow MontaVista to continue supporting other hardware platforms while working in tandem with Cavium products.

Early Revenue from Software

The typical cycle for revenue of a chip company is long. The company takes 2-3 years to build a chip. It markets the chip and makes a design win with the customer. There is no revenue yet except for prototypes which are typically factored into the deal. The customer takes another 2-3 years to build, test and sell the product. So to see a real revenue from a chip is anywhere between 3-6 years. Any chance of making money early in this cycle is a better option.

This deal with MontaVista allows Cavium to monetize design wins early in the design cycle before products go to mass production by selling operating system, productivity tools and professional services. (caviumnetworks.com 2009).

Chapter 6: Building Software IP portfolio

So far we have seen how software is important for chip companies and how it enhances their products from the customers' perspective. Different trends happening in the industry also demonstrates the role of software for chip companies. With software being a necessity for a chip product, the thesis now explores how software expertise can be built in a software team that is involved in supporting a chip product. Specifically, it looks at how a strong software patent portfolio can be built in this setup.

Bill Gates has said that Microsoft, at any given time, is only two years away from failure. His basic message is that all companies have to innovate to survive. (Teska 2007). Companies that don't innovate on a regular basis perish at their own risk. For instance, photography major Kodak Inc., a leader in film photography, completely missed the digital photography innovation and is now struggling to catch up with rest of the industry. In today's technology-driven markets, most companies compete through ideas and relationships. Their most important assets are their intellectual property, knowledge and people. Patents are the most quantifiable of intellectual property assets. (Cantrell 2009)

Software tools provided with the chip are an integral part of the value proposition. These tools help the chip to be integrated easily into the customer system. From a customer perspective, they see the chip and the software tools as part of the solution. With software tools so closely tied to the success of the chip, it is important to innovate and protect it by filing patents. The company needs to protect all aspects of the product – both hardware and software. This will protect the whole ecosystem of the chip.

CAPTURING PRODUCT INNOVATION

Towards protecting the intellectual property, companies usually have an elaborate process to capture the inventions during product development. Generally, there is a product specific IP workshop at regular intervals to identify patentable inventions. A company also has an invention disclosure program where the engineers document inventions. These inventions are reviewed by a panel of experts on patentability and decided how to proceed on an invention. Once an invention is determined to be patentable, the IP team does research on prior-art and if the invention is unique enough, the invention is filed for a patent at the patent office. Financial incentives are provided to the inventors to motivate invention and disclosure. These processes and procedures are applicable to both hardware and software teams involved in developing the product.

IP Workshop

The IP workshop is generally the starting point to capture innovations in products. These workshops are conducted periodically at every product development site. The representatives from the IP team meet with the engineers who actually develop the product and hence are more likely to innovate. The engineers put forward different ideas and implementations that they have been working with. The IP team make note of the interesting ones and follow up with the inventors after the workshop on next steps.

Typically the frequency of these workshops depends on the complexity of the product and potential for new inventions in the product. It is also determined by whether the product is path breaking next generation chip or an incremental chip based on existing design. There is lot more scope for inventions in a next generation chip compared to the incremental chip. The frequency of the workshops can be anywhere from once every quarter to once every two years.

Invention Disclosure Procedure

The inventions identified in IP workshop as patent-worthy are documented in Invention disclosure process. This captures the inventors, the description, a diagram depicting the invention and other details of it. Care is taken here to include all of the inventors. Once all the details of the invention are captured, it is then submitted to the review committee. The review committee looks at it and determines patent worthiness. For those inventions that are patent-worthy, they also research for prior art. If something similar is already patented, then it's likely that this invention is not patented for there is a good chance that the invention is rejected by USPTO.

In the patent world, the cost even to be in the game is high. The cost to file for a patent in US on anything but simplest of inventions will run over $10,000 and quite possibly over $25,000 per patent, considering filing and prosecution fees (Quinn 2011). The patent filing cost is another factor that is considered before filing for a patent. If the invention is outside of core IP of the product and if the revenue and/or product defense potential is not high, it's possible the invention is not recommended for patenting.

Software Patents

With all the processes and procedures to capture the product innovation, the number of patents filed in hardware outnumbers the patents filed for the software tools in the product that the author is involved on. This is an interesting metric. The procedures and the financial incentives provided are identical for both hardware and software teams. Yet, the numbers of patents filed by the teams were not comparable. This is the basic question of the thesis: Given a product, why does number of patents filed in software is less than that of hardware and whether this is prevalent in other products?

Is it possible that there is not enough scope in software to file that many patents? Or is it that there was not enough focus on software patents for engineers to think about coming up with patentable ideas? Or is it possible that there were patentable ideas but they were not just followed up with the invention disclosure and rest of the patent filing procedures. This thesis hopes to find answers to these questions and put forward recommendations on how this can be avoided.

Patent Portfolio Case Studies

The Method

Products

As part of this process, multiple products were identified to analyze the strategies employed to build patent portfolio. Special attention was paid to the strategies used for software patents. The products identified are from networking and storage industries. Although, the industries are different, the type of products is the same – they are all chip products that are used to develop end products by the customers.

People

With the products identified, next step was to identify whom to be interviewed to gather the necessary information. With the kind of information that is needed for the analysis, the person needs to have a high level view of the product itself like a technical architect or some other person who is involved in managing the patenting process for this product. It was also important for the person to have firsthand knowledge of the product in terms of how it was developed, the inventions that were patented for this product – both hardware and software related and be familiar with the patent process.

Information

A questionnaire was prepared with information that needed to be captured for different products. Some of the information that was captured involved the number of patents filed, whether they were hardware or software related (if it is possible to categorize them that way) and typical uses of these patents. The author also tried to capture patent revenue information for these products. However, that information was not available on a per-product basis. The interview then focused on the patent portfolio building strategy used by the team. It gathers information about different aspects of it. This information is used later to analyze the strategies and provide recommendations. For reference purposes, the questionnaire used for case study is included in the Appendix section.

Due the nature of information that is captured, the products are not being identified and proprietary information regarding them is not shared. However, the data is used make recommendations on best and worst practices.

Case Studies

Storage Controller Chip

This chip is used as a storage controller for server systems. As part of the product development, well defined patent development strategy was used. Patents were filed both in hardware and software areas. This is one of the products where the number of hardware and software patents was comparable. Regular patent workshops were conducted to capture product innovation.

Networking Chip

This chip is used in communication equipments to process packets. Patent portfolio building was part of the development strategy but on a background priority. The development of the product was given more importance. Patents were filed in both hardware and software areas. However, the software patents lagged the hardware ones. This was one of the first products developed for this market by the company and had an opportunity innovate both in hardware and software.

Processor Subsystem Product 1

This IP is used as processor subsystem in System-on-Chip products. IP portfolio building was part of the plan although it was mainly self driven. IP workshops were conducted; inventions were identified and filed for protection. These workshops were not held on a regular basis.

Processor Subsystem Product 2

This IP is also used as processor subsystem in System-on-chip products. However this IP was handled by another team that had a detailed plan to build IP portfolio. This plan was made an integral part of the product development when frequent workshops and measurement of metrics that encourage disclosure of inventions. These procedures were designed to make the filing easier on the inventors.

Analysis of Case Studies

Comparing the above products and their software patent building strategies, it's a mixed bag of best practices and worst practices. Some of the products seem to have got the act right in software innovation. While others' model largely relies on personal interest.

One of the main factors, during the case study, that was found to have direct impact on software patents is the management focus. The management needs to give sufficient importance to software patenting so that the software team can work on it and come up with creative ideas that can be patented. During the case studies two of the products, Storage Controller Chip and Processor Subsystem Product 2, which had full management focus on software patents filed more number of software patents. When the management didn't give much importance, the numbers showed in them – they filed less number of software patents like in other two case studies.

The other interesting aspect that played a role in motivating engineers is setting goals and rewarding them around software patents. Storage Controller product used personal and group objectives to set goals to engineers and teams. The performance against these goals was used during review process to reward the team and individuals who did well. Program-level incentives were also put in place around software patents to encourage engineers to innovate in software. Going by motivation theory, this strategy made sure that there were no conflicts between the behavior that is expected and the behavior that is rewarded. The result of such consistent strategy is that the product produced more number of software patents.

Processor Subsystem Product 2 used frequent workshops and surveys with the engineers to improve the patent filing procedure. They monitored several metrics that measured time between an idea, its implementation and its disclosure. This allowed the management to fine tune the procedures so invention disclosure doesn't become a hassle. The engineers readily disclose inventions benefitting both themselves in terms of incentives and also the product now that it has more innovations protected.

Some of the products didn't fare well in software patents department because either a structured plan for building software patents was not put together or schedules and customer milestones were focused more. This again showed in the number of software patents filed - they were less as the focus was not on innovation.

Chapter 7: Recommendations

Personal and Group Objectives

Setting personal and group objectives around software invention disclosures was found to be very effective in guiding the individual or a team in the right direction. These objectives set the baseline tasks that need to be achieved for the group and the engineer. It also indicates the importance of invention disclosures to the management team making the engineers to put more effort on patenting ideas. The performance on these objectives could be taken as input to the review process and used to reward the employees rightly through career advancements and other incentives. This motivates everybody to a common goal of software innovation.

Program-level Incentives

Program level incentive is another tool that could be used to motivate software teams and push them for innovation. Generally companies have an incentive program around invention disclosure, filing and approval. The program incentive in conjunction with the company-wide incentive creates an edge making the engineers go the extra mile to innovate and come up with creative solutions to problems.

Program level incentives typically use monetary benefits to motivate people. People are motivated by money more than anything else. However, monetary motivation has its limits. Numerous studies have concluded that for people with satisfactory salaries, some non-financial incentives are more effective than extra cash in motivating employees (Gibbons 2006). A recent *McKinsey Quarterly* survey underscores the opportunity. The respondents view three noncash motivators—praise from immediate managers, leadership attention (for example, one-on-one conversations), and a chance to lead projects or task forces—as no less or even more effective motivators than the three highest-rated financial incentives: cash bonuses, increased base pay, and stock or stock options (Dewhurst, Guthridge and Mohr 2009). A company needs to explore both these aspects and come up with a right mix of them. Some examples of non-monetary incentives that could be used are given below.

- Opportunity to play expanded roles like representing the company in professional bodies like technology standards committee, ITU, IEEE, IETF, etc.
- Publicly recognizing the contributions of employee by giving honors.

MENTORING JUNIOR ENGINEERS

Innovation is a continuous process that needs to be nurtured and cultivated over extended periods of time. The benefits of such sustained innovation are enormous - ranging from competitive advantage and leadership position to entering new markets to the ability to hire and retain best talent. The innovation pipeline needs to be a well oiled machine working continuously. This goal needs persistent efforts in grooming junior engineers to be innovators of the future. The senior engineers can play a big role in this process by mentoring junior engineers. This provides an opportunity for junior engineers to learn the tricks from the pros and lead the way in future.

Multiple products in the case study used this approach to enhance the skills of junior engineers and groom them for future. The senior engineers can put forth the ideas and the details while the junior engineers can support the senior engineers by following up on the invention disclosure and other necessary tasks. This ensures junior engineers are involved in the innovation right from the early career and are more tuned to it.

This strategy didn't work in some cases. Most common reason being the senior engineers were reluctant to share their ideas with junior engineers and include them as co-inventors. If this trust gap is overcome, the benefits are enormous in the long term.

MAKING PATENT DISCUSSIONS PART OF DAILY ROUTINE

It is not enough to have invention disclosure procedures, IP review board or incentives to encourage innovation. The innovation culture needs to be brought to the lowest of the team members and built ground up. Frequent brain storming sessions ought to be scheduled where IP is the only topic. This generates enough critical mass of the IP thought process motivating engineers from each other. These sessions should involve both senior and junior engineers bouncing ideas off of each other. These types of sessions also allow the junior engineer to pick up the thought process from seniors helping them in future.

HASSLE-FREE INVENTION DISCLOSURE PROCEDURE

This deals with procedural aspect of the invention. Once the invention is conceived and implemented, it needs to be disclosed and evaluated whether it is worthy to protect. If it is, it needs to be decided whether to go trade secret route or patent route. Sometimes, this procedure could be long and hassle-prone from inventor's perspective. Such long procedures could end up being an impediment to declare inventions resulting in not knowing about them.

One of the products in the case study conducted frequent IP workshops collecting detailed information about the invention. One of the main objectives of this workshop is to reduce the barrier in filing invention disclosures and making them hassle free for engineers. Different metrics like time taken to file a disclosure, time taken to initiate invention disclosure since the time the idea was invented and many such useful metrics were gathered to help manage the process better. These were used to identify and streamline bottlenecks in the system. This helped a lot in increasing the comfort level of the procedure with the engineers. Once they got used to the procedure, the invention disclosure became less of a hassle and more of a part of job description. This turnaround only helped the product for better.

NO PLAN IS THE PLAN

In certain cases, although all procedures and incentives are in place to encourage software patents, only few patents are filed. Processor Subsystem Product 1 case study is an example of this. The basic reason for this is rooted in the fact that there was no implementation plan in the team level. There were no periodic brainstorming sessions and no frequent IP workshops. Any patent that was filed was mainly driven by self motivation. Had a plan been in place, it would have created a more formal structure around the efforts. This could have yielded better results and more number of inventions being disclosed. The software patent building strategy needs to be made a part of product development.

EXCESSIVE FOCUS ON SCHEDULES

Often times, in an effort to generate customer traction, teams are driven by customer commitments and schedules. This becomes a large part of product development pushing the management to assign higher priority to these tasks. This causes inventions and their disclosures to take a lover priority. This is a short sighted approach causing long term issues with the product.

There needs to be a balancing act that the managers need to do selectively prioritize the tasks. Effort must be taken to de-prioritize tasks for customers that can wait and push IP related tasks so they get worked on. This day-to-day prioritization is a necessity that the manager has to deal with. Alternatively, an initial scoping effort can be done to identify potential inventions. These ideas can be worked on later scheduling a chunk of time for it in the project schedule.

MANAGEMENT FOCUS

Management focus or lack thereof, is the single biggest reason for software patents to lag behind hardware patents. In cases where the management was equally committed to innovation in software and hardware, the results are apparent in the number of patents filed by the software teams. When the management focus is lacking in software innovation, it shows in the patents filed.

Some of the examples of good management focus would be keeping track of metrics around invention disclosures, allowing creative space to software team to do their own thing once they are set in right direction. It could also involve fine tuning the procedures to make it easy to file software patents. Being a chip company, the management in hardware side always pushes for IP generation. If similar tactics are not used in software side, the results are clear – less software patents and those are only driven by personal interest.

Chapter 8: Conclusion

In conclusion, the software and tools that are provided with chip products are crucial to make them usable to its customers. It helps the customer to integrate the chip easily into their system. It also reduces the time to market as the software provided with the chip already knows how to configure it. Different chips need different levels of software support. A flash storage chip needs only driver software while a network processor needs a suite of software tools to make the initial configuration, change configuration at runtime, simulate it and provide application development kit that is closer to industry and technology making end-system development easier.

The trends in semiconductor industry confirm the role of software to chip. There have been multiple acquisitions of software companies by chip makers. Intel bought Wind River Systems in June 2009 and Cavium Networks bought Montavista in November 2009. Both these acquisitions help the buyer to better position themselves in their existing market or to go after new markets increasing their top line. It also helps them to bid for bigger piece and come up with a full solution.

Since software is a very crucial to the success of the chip, it’s very important to innovate and protect it by filing patents. Most of the companies have well defined procedures to capture product innovation and file invention disclosures. They conduct IP workshops periodically to identify inventions and file them. They also have incentive programs to encourage inventions and their disclosures.

In spite of all these procedures and incentives, in some cases less number of software patents are filed when compared to hardware numbers. Multiple case studies were performed on multiple products to analyze the root cause of this and propose recommendations. Four products were studied for their IP building strategies, the best practices that encouraged software patenting and the ones that didn't help.

On the products where software patents didn't fare well, several things were identified to be lacking. First and most important was lack of management focus on patenting in software. This resulted in other things being prioritized over patenting resulting in low numbers. Another aspect that came to light was not having a detailed plan on the patent strategy. This resulted in best effort strategies motivated by personal interest. This didn't work well in fully utilizing the opportunity to file patents in software. Making the disclosure procedures hassle-free also helps as the engineers are more comfortable with it and use it more often.

On products where the numbers are comparable multiple strategies were being adopted to motivate inventions and their disclosures. Personal and group objectives were assigned to teams and career development was linked to this performance. Program level incentives were also initiated providing more incentives for inventions and their disclosures. Brainstorming sessions were conducted regularly to make software patenting part of daily routine in addition to schedules and deadlines. Junior engineers were also involved in the process guided by senior engineers. This ensured the innovation culture is passed on to the next generation of engineers that would take lead in future.

Appendix

CASE STUDY QUESTIONNAIRE

Little bit of a background on what I am trying to do. I am focusing on software patents and innovation as pursued by chip companies. Filing, monetizing and defending hardware patents in a chip company is normal and given as that is their bread and butter. However, software plays a key role in making the chip easier to adopt by the customer. A chip cannot be sold without the tools and software that go along with it. I would like to analyze how the chip companies leverage and innovate in software in terms of filing patents, defending them and making revenue out of them. How do they compare with hardware patents in the above aspects? Finally I want to look at patent portfolio building strategies and any recommendations for chip companies to leverage software patents.

I have put together a set of questions that identify a product in your business unit and query related information about it.

I thank you very much for your time and assistance.

1. Please identify a product that you directly deal with and provide a short description on it.

2. Is the product in production currently? If not, when was it discontinued?

3. When and why was the product conceived?

4. When did the product development start and how long it took to complete the first version of the product?

5. How many total patents were filed for this product?

    a. Up to 10
    b. Up to 20
    c. Up to 40
    d. More than 40

6. Of these patents how many were hardware related?

    a. Up to 5
    b. Up to 10
    c. Up to 20
    d. More than 20

7. Of the total patents in Question 9, how many was software related?

    a. Up to 5
    b. Up to 10
    c. Up to 20
    d. More than 20

8. How are these patents being used? Select all of them that apply.

    a. Revenue generation through licensing.

    b. Use it in our products so others can't.

    c. Defending the IP against use by rivals.

    d. None of the above.

9. What percentage of the software patents were/are generating revenue?

    a. Up to 10%

    b. Up to 20%

    c. 20% - 50%

    d. More than 50%

    e. None of the above.

10. What percentage of the hardware patents were/are generating revenue?

    a. Up to 10%

    b. Up to 20%

    c. 20% - 50%

    d. More than 50%

    e. None of the above.

11. Was building patent portfolio part of the plan?

12. What was the strategy to build patent portfolio?

13. Were patent workshops held with engineers? How often were they held? Please provide a timeline if you can.

14. Was any special incentive program (other than companywide) implemented to encourage engineers to come up with patentable ideas? Were they effective?

15. Any other strategy that was used to encourage engineers to come up with patentable ideas?

16. Any recommendations from personal experience to bring patent culture down to every engineer contributing to the product and not just senior engineers?

Bibliography

Cantrell, Robert L. *Outpacing the Competition: Patent-based Business Strategy.* John Wiley & Sons, Inc., 2009.
caviumnetworks.com. *Cavium Networks Acquisition of MontaVista Software : FAQs.* November 10, 2009. http://www.caviumnetworks.com/pdfFiles/Cavium%20Acquires%20MontaVista%20FAQ.pdf (accessed June 23, 2010).
—. *Cavium Networks Signs Definitive Agreement to Acquire MontaVista Software.* November 10, 2009. http://www.caviumnetworks.com/newsevents_Caviumnetworks_MontaVista_Acquisition.html (accessed June 22, 2010).
—. *Ecosystem Partners.* http://www.caviumnetworks.com/Ecosystem_Partners.html (accessed May 23, 2010).
Dewhurst, Martin, Matthew Guthridge, and Elizabeth Mohr. *Motivating people: Getting beyond money* . November 2009. http://www.mckinseyquarterly.com/Motivating_people_Getting_beyond_money_2460 (accessed August 27, 2011).
finance.yahoo.com. *Intel Business Summary.* http://finance.yahoo.com/q/pr?s=INTC+Profile (accessed June 01, 2010).
Fuller, John. *How Drive-by-wire Technology Works.* April 28, 2009. http://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/drive-by-wire.htm (accessed June 26, 2010).
Gibbons, John. *Employee Engagement: A Review of Current Research and Its Implications.* Conference Board, 2006.
intel.com. *Intel to Acquire Wind River Systems for Approximately $884 Million.* June 04, 2009. http://www.intel.com/pressroom/archive/releases/2009/20090604corp.htm (accessed June 07, 2010).
Kharif, Olga. *Intel-WindRiver: What the Merger Means.* June 05, 2009. http://www.businessweek.com/technology/content/jun2009/tc2009065_713999.htm (accessed June 15, 2010).
Klien, Ken. *Wind River's Next Chapter.* October 21, 2009. http://blogs.windriver.com/klein/2009/10/wind-rivers-next-chapter-1.html (accessed June 17, 2010).
mvista.com. *About Us.* http://mvista.com/about.php (accessed May 18, 2010).
—. *Cavium Networks Signs Definitive Agreement to Acquire MontaVista Software.* November 10, 2009. http://mvista.com/Cavium/Cavium-acquires-MontaVista.php.
Ojo, Bolaji. *Intel eyes embedded apps with Wind River deal.* June 04, 2009. http://www.eetimes.com/electronics-news/4083260/Analysis-Intel-eyes-embedded-apps-with-Wind-River-deal (accessed May 15, 2010).

Quinn, Gene. *Cost of Obtaining Patent.* January 28, 2011.
http://ipwatchdog.com/2011/01/28/the-cost-of-obtaining-patent/id=14668/ (accessed August 21, 2011).
Rogoway, Mike. *Rewriting the rules: Intel's software chief challenges convention.* October 03, 2009.
http://www.oregonlive.com/business/index.ssf/2009/10/rewriting_the_rules_intels_sof.html (accessed May 25, 2010).
Sherr, Ian. *AMD chips away at Intel market share in Q4: iSuppli.* Edited by Richard Chang. March 31, 2010. http://www.reuters.com/article/idUSTRE62U4HA20100331 (accessed June 07, 2010).
Teska, Kirk. *Patent Savvy for Managers.* Nolo, 2007.
Thomas, Bryan. *Intel to Acquire Wind River Systems for Approximately $884 Million.* June 04, 2009. http://www.windriver.com/news/press/pr.html?ID=6921 (accessed June 17, 2010).
—. *Wind River Extends Virtualization Support With New Release of Wind River Hypervisor.* February 9, 2010. http://www.windriver.com/news/press/pr.html?ID=7861 (accessed August 28, 2011).
—. *Wind River VxWorks 6.8 Broadens Multicore Capabilities and Processor Support.* January 5, 2010. http://www.windriver.com/news/press/pr.html?ID=7701 (accessed August 28, 2011).
ti.com. *TV: High-Definition (HDTV).*
http://focus.ti.com/docs/solution/folders/print/387.html (accessed May 11, 2010).
windriver.com. *About Us.* http://www.windriver.com/company/ (accessed June 12, 2010).
—. *Wind River Corporate Profile.* 2010. http://www.windriver.com/company/Wind-River-Corporate-Profile.pdf (accessed June 07, 2010).

Vita

Venkatramanan Jayatheerthan has completed Bachelors of Engineering from P.S.G College of Technology, Bharathiar University in India. He is currently earning M.S in Engineering Management from University of Texas.

He worked for few years in India and continued his career in USA. He has 14 years of experience in software development, consulting and leading projects in embedded systems using multiple operating systems and processor platforms.

Email address: jjayathe@yahoo.com

This thesis was typed by Venkatramanan Jayatheerthan.