Guidebook to Partitioning and Virtualization in an Embedded Environment

by Josh Whitehead

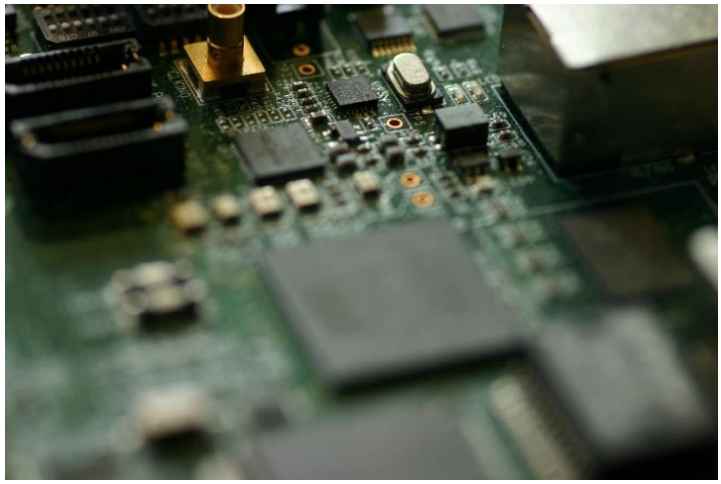
DornerWorks, Ltd.
www.dornerworks.com

Technology engineering so you can focus.

Partitioning and Virtualization in an Embedded Environment

What is Driving the Need for Virtualization?

In the past few years, the embedded computing space has seen a similar rise in demand and need for virtualization and partitioning that the enterprise IT/Server world saw nearly two decades ago. Some of the factors driving this demand are similar to the server world, the primary one being that modern embedded System On a Chip (SoC) modules often have substantially more processing, memory, and I/O resources than can be optimally used by a single application or operating system.



Virtualization and partitioning provide a method for efficiently using this abundance of hardware resources by allowing multiple independent software payloads to be run concurrently on the same hardware. This capability makes virtualization attractive in embedded computing as many of the target industries (aerospace, automotive, medical, etc) put great value on size, weight, and power (SWaP) reductions, so combining many applications onto a single computing platform is highly desirable.

However, these industries have an extra requirement beyond that of the enterprise IT world: their software must pass various levels of stringent certification (DO-178, ISO26262, IEC 62304, etc) in order to be used on their target products.

Generally speaking, the smaller and more specialized the piece of software, the less effort is required to pass these certifications. The traditional "federated system" approach facilitated this certification by building many small applications that were then deployed across a large system of independent and isolated processors (Fig 1). So while virtualization can provide a means to run multiple applications on the same platform and thereby reduce SWaP, it is still a requirement that the virtualized hardware provides a similar level of isolation between the software applications as achieved by the federated hardware. Meeting this requirement for isolation is the only way in which the software running in a consolidated system will be able to achieve its requisite certification. It is with that goal in mind that we introduce a hypervisor into the system.

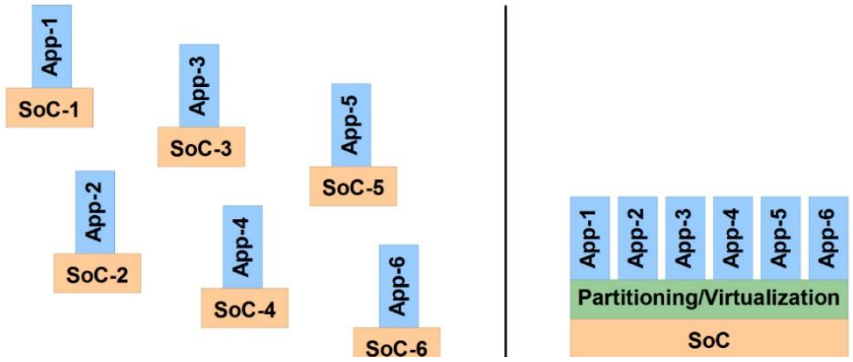


Figure 1 – Federated vs Consolidated System Architecture

Software Virtualization

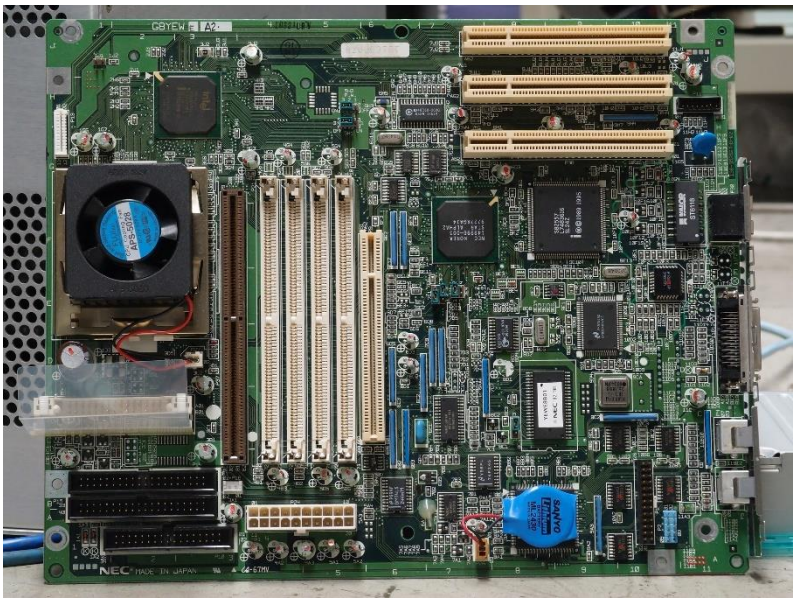
A hypervisor, or more generically, "virtual machine monitor," is a piece of software that allows for the creation and management of virtualized hardware partitions in the form of virtual machines. If implemented properly, the hypervisor is capable of isolating its virtual machines from one another in such a way that any one virtual machine is incapable of affecting the behavior of another. In this way, the isolation characteristic of a federated system is achieved via software.

DornerWorks has been working with several hypervisors and related separation/isolation technologies and applying them to the embedded computing arena for many years. Included

in this list are commercial embedded partitioning environments like Wind River VxWorks 653 and Green Hills INTEGRITY Multivisor, as well as open source environments like The Xen Project Hypervisor, OpenXT, and the seL4 microkernel, with our primary focus being on embedded Xen. All of these are software-based virtualization and partitioning technologies that take different approaches to the same goal mentioned above: allowing multiple virtual machines to run on single piece of hardware without causing negative interference to one another (isolation) while providing the prescribed level of performance and/or real-time timing.

Hardware Partitioning

In the past year or so, a new technology has appeared that attempts to tackle this same goal, which I will broadly refer to as “hardware partitioning.” Hardware partitioning is not entirely new, it’s been happening on a small scale on both Intel and ARM with the introduction of their respective “virtualization extensions” technologies, which allow things like segments of memory and I/O registers to be constrained, at the hardware level, to a particular virtual machine being operated by a compatible hypervisor.



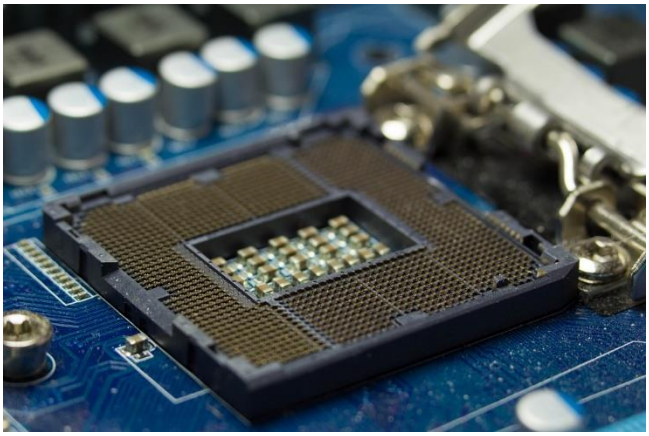
Recently, a few SoC families have been released or announced that expand that capability, and are able to partition all elements of the SoC, including the processor cores themselves, at the hardware level without requiring hypervisor. An example of this is the i.MX8 from NXP which uses a System Controller interface, programmed by the bootloader via special registers, in order to configure the state of the hardware on the SoC. It separates the available processor cores, memory, and I/O into predefined partitions that can be used and accessed as though they were completely independent SoC’s.

Let’s take a look at some of the advantages and implications of this type of partitioning and how it compares to software-based partitioning.

New Capabilities - With Some Caveats

Claiming Simpler Configuration, Hardware Partitioning Comes with its Own Complications

One common criticism of hypervisors and software partitioning is that it can be complicated to properly setup and configure. This likely stems from the high level of customizability and large number of options for accomplishing the same task that are often present in a typical hypervisor solution. A benefit hardware vendors claim is that hardware partitioning is easier to use and configure than equivalent software options. This is likely true as the number of available options for partitioning would be much smaller. Further, because it is tied to a specific processor and architecture, there are no extraneous features as are typically included in a software solution that is written to support multiple hardware architectures. However, this type of partitioning could be problematic for an embedded developer as it relies heavily on the hardware vendor to provide appropriate tools for creating the configuration and for providing thorough and accurate documentation so that all of the side effects of a particular setting can be understood correctly. The methods for creating these configurations will also be hardware vendor specific, and possibly even processor family specific, such that moving to another vendor or even a new processor family from the same vendor may require retraining and reworking the configuration and interfaces for the partitioned software.



In contrast, a software based solution like Xen may initially be more complex to configure, but the majority of the knowledge gained and the work performed on one processor will apply to all others, with little or no re-porting effort.

While Potentially Robust, Hardware Partitioning Does Not Eliminate the Need for Good Software

Hardware partitioning can allow for the complete elimination of software-based partitioning, virtualization, and emulation. This may be a highly desirable feature in secure systems as it

can lower both the number of attack vectors (the software is no longer there to be compromised) and reduce the size of the attack surface of existing vectors. This leads to a benefit often claimed by hardware partitioning methods: hardware partitioning cannot be compromised due to faulty software.

As mentioned previously though, realizing this benefit would still require some expert knowledge of configuration or startup code, and it also makes a few assumptions. First, it assumes that the hardware itself has been configured correctly, which is reliant on that expert knowledge of the system configuration and setup. Second, it assumes that whatever configuration interface is being used, such as the previously mentioned i.MX8 System Controller, and its associated software, doesn't have a vulnerability that could be exploited to break the hardware partitioning. Lastly, this benefit also makes the assumption that the partitioning hardware itself does not have a vulnerability in its structure which, with the recent rise of attacks like RowHammer(1), is becoming a more and more problematic assumption.

Hardware Partitioning May or May Not Ease Safety and Security Certification

One final benefit offered by hardware partitioning is that it can lower the complexity of combining high criticality certified software, like ISO 26262 or DO-178, with non-critical software. Because the partition is created and enforced at the hardware level, the certification needs of one set of software will not bleed over to other software on the system forcing previously uncertified software to require certification (just to be sure that it does not impact the critical functions). Driving home this benefit, some hardware vendors claim that their hardware configuration is mathematically provable. It should be noted though that this benefit assumes that it is easier to prove hardware partitioning is correct (and therefore sufficient to not require recertification or additional certification of application software) than it is to prove that software partitioning is correct.

In the arena of certification, there is also the question of whether or not the hardware vendor will cooperate with the certification authority in question by providing its hardware design and documentation, as well any designs and documentation for their configuration tool/boot ROMs used to setup the hardware in the first place. It also doesn't eliminate the

need for software certification: some portion of the software must still pass the necessary certification process. In some ways, hardware partitioning could potentially complicate certification efforts, as having a modifiable hardware configuration has often been a sore spot with certification authorities, especially in the aviation industry. This means a mechanism like NXP's System Controller, that might allow an end user to modify the hardware's configuration, could compound many issues dealing with the certification of software on that system.

Limitations of Hardware Partitioning and Missing Features

Resource Sharing is Only Made Possible Through Virtualization

One of the biggest limitations of hardware partitioning is that it's just that, partitioning, and partitioning only. It cannot perform any form of virtualization of hardware resources. This means that it is always acting on a fixed limit of literal hardware resources, and cannot provide more virtualized copies of a device than are built into the SoC when it is manufactured.

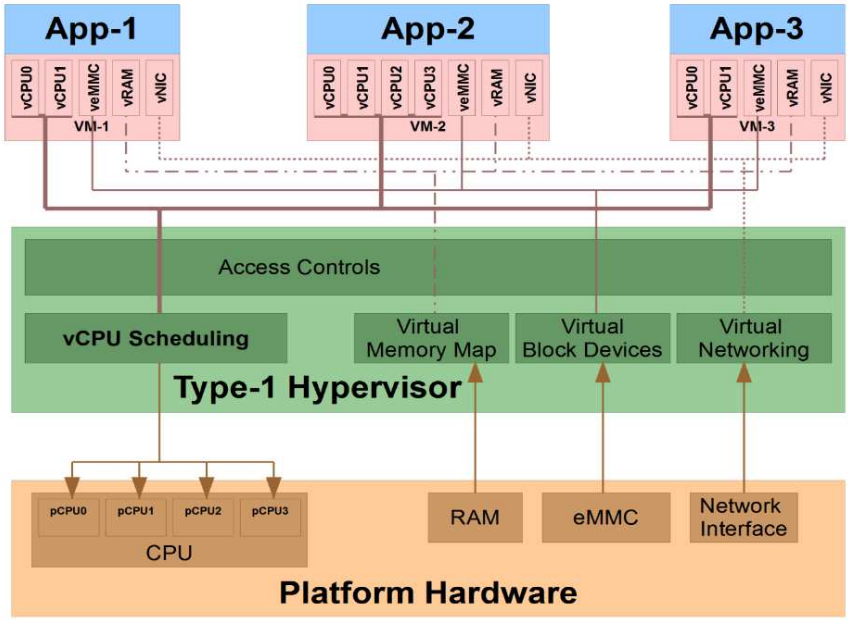


Figure 2 – Sharing Hardware Resources Using a Hypervisor

For example, if four virtual machines need access to a UART, but only two physical UARTs exist on the SoC, only two VMs will get access. If three virtual machines need access to an Ethernet device, but only one physical device is available, only one will get access. Thus, these types of resource-sharing use cases cannot be met by hardware partitioning alone. To address this limitation, software could be written to share these resources across virtual machines, which is precisely what existing software virtualization solutions, like the Xen hypervisor, have implemented (Fig 2).

In fact, it was this sharing of resources that was one of the most significant driving factors behind the adoption of virtualization in the enterprise server world – why have a multi-gigahertz multi-socket multi-core processor server with dozens of gigabytes of RAM and terabytes of storage hosting a single website? Similarly, in the embedded space, why have a multi-gigahertz, multi-core SoC with access to gigabytes of RAM running a single embedded application? The inability to share resources becomes particularly limiting in lower end chips in a processor family that still have multiple cores, yet fewer other physical resources (particularly I/O devices) to be spread around on a 1-to-1 basis via hardware partitioning.

Resource Sharing via Virtualization Includes Processing as Well

The examples of resource sharing above looked primarily at I/O devices, but this idea of resource sharing has a different set of implications when applied to processor resources (Fig 3). Hardware partitioning does not perform thread scheduling; therefore, if your number of software components that must be separated into partitions is equal to your number of processor cores, you are inherently limited to placing a single thread on each core. This is the case whether that thread is a tiny "bare metal" air speed monitor application, or a full Linux operating system with all the bells and whistles. In a situation that requires more partitions than there are available processor cores, with hardware partitioning alone you are at an impasse, and many of the use cases that we see at DornerWorks require far more than the 2-8 partitions that 2-8 cores would allow on today's typical SoC. Even in a case where there are enough cores available, you are very likely to be making inefficient use of the processors resources as hardware isolation is not work conserving and will frequently result in unsolvable "use it or lose it" situations.

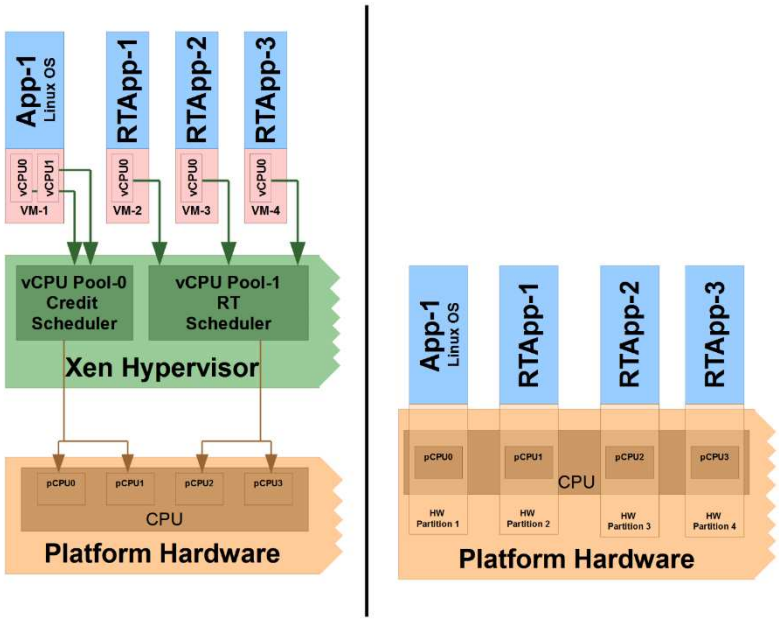


Figure 3 - Processor Allocation Strategies Using Xen vs Using Hardware Partitioning

Hardware Assisted Software Virtualization vs Hardware Partitioning Only

For example, a common use case might involve 3 small bare metal applications, that all require their own partition due to various certification and real-time requirements, and a single Linux operating system acting as a user interface and general purpose OS. On a quad-core SoC with hardware partitioning you would need to place each of these applications on a single processor core to maintain isolation.

Key		
Name	Description	CPU Needs:
App-1	Linux OS, 150%	150%
RT App-1	Periodic task, static exectuion time	30%
RT App-2	Periodic task, variable exectuion time	20-40%
RT App-3	Event driven task, static exectuion time	10% per call
	Available Idle CPU time	
	Unavailable Idle CPU Time	

The small bare metal apps, while still needing to meet their real-time requirements might only consume 20% to 40% of the processing time for the core on which they reside, meaning as much as 80% of each of those three cores processing capability is wasted (Fig 4).

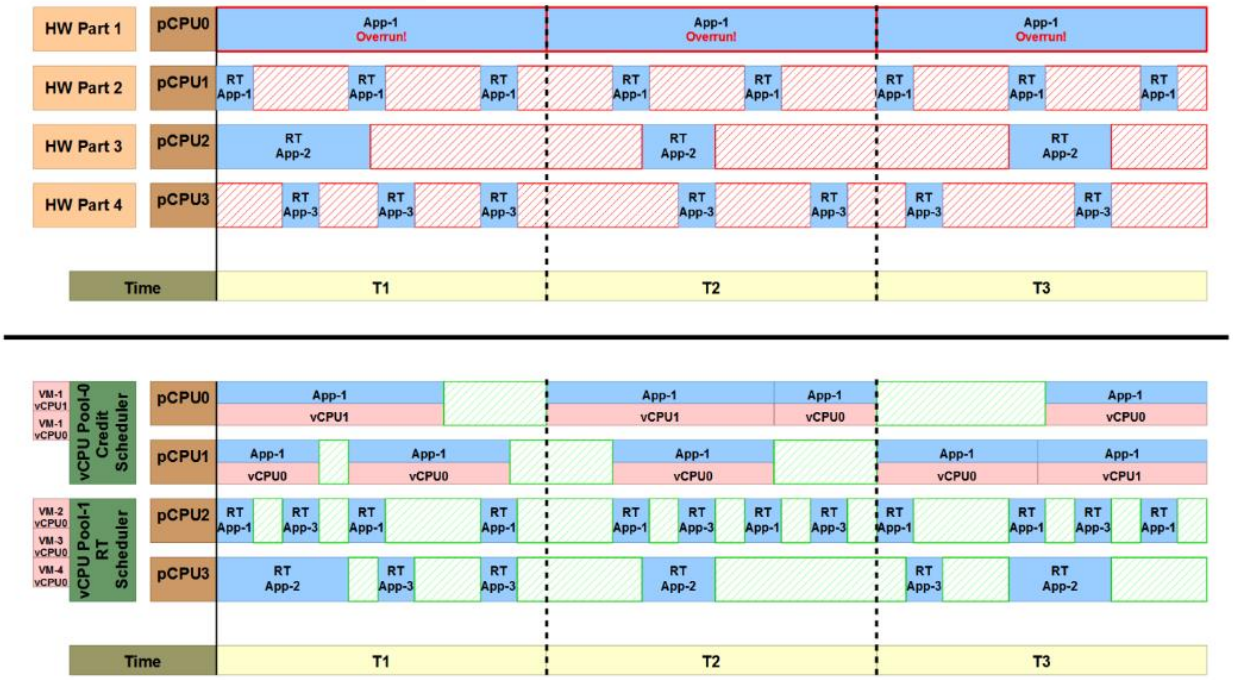


Figure 4 – Hardware Partitioned Processing vs Shared Processing Using Xen

While conversely you have a sluggish and underwhelming user GUI struggling to keep up because it needs more processing power than available on the single core to which it is restricted. Additionally, in this situation, there is no room for expansion or addition of new features or capabilities in the system as your hardware is already maxed out.

If we use Xen as a representative software partitioning solution, this situation looks very different. First, we create four partitions each containing the same apps as described above. We then use the Xen CPU Pool feature to create two CPU pools with two cores each. In the first pool, we place the Linux OS virtual machine and assign the Xen credit-scheduler to the pool, which will schedule the partition's vCPUs among the two available cores in much the same way the Linux "Completely Fair Scheduler" schedules Linux processes. We then place the other three domains into the second CPU pool and assign the Xen real-time scheduler to that pool, which when given parameters for each of the bare metals partitions, will multiplex them across the two available CPU cores in such a way that each can meet its real-time requirements. In this revised approach, each application is still isolated in its own partition, but the user GUI has enough processing resources to run very effectively, and the bare metal applications can still meet real-time and certification requirements, all while still leaving spare processing time for future expansions and additions. Because the Linux vCPUs are not required to run on a particular core (as shown in Fig 4), another non-real-time application or OS could be added to the Credit Scheduler pool, as long as it did not exceed the available processing time in that pool. Depending on timing requirements, it would even be possible to place one of the two real-time cores in a third CPU pool, and to run another type of scheduler in that third pool to meet the needs of any new applications.

System Wide Features, e.g. Inter-partition Communication and Health Monitoring

Strict isolation at the hardware level is not conducive to centralized health or security monitoring, and makes partition management cumbersome at best. While a hypervisor can be viewed as an additional point of attack in a system, it can also act as a buffer against many security and system management issues. It has been common practice amongst security minded businesses and agencies in the last few years to "assume the system is already compromised" [(2)] when dealing with external threats. This makes the role of a system security monitor critical in watching for and mitigating malicious behavior in a system, and it's something that's just not possible using hardware partitioning alone.

Outside the realm of security, many embedded applications would benefit from, and in some cases (such as the ARINC 653 standard) require, a system-wide health monitor that can detect when virtual machines or applications experience faults, performance problems, or other issues and can take automated steps to resolve or mitigate those issues.

Hardware partitioning methods also have no common standard for inter-partition communication. It is common to have a use-case where, even though applications are divided into separate partitions to avoid negative interactions, they still need to share data with one another. These types of inter-partition communication mechanisms could be implemented in a hardware partitioning system, but with no industry standards for reference it would be costly to develop and would have to be re-implemented, perhaps using a completely different method based on available features, on each individual hardware platform. This problem has already been solved by many of the existing software partitioning systems, and the solution is portable across all hardware platforms on which that software solution can run.

Customer Support and Vendor Dependency

A final important consideration when evaluating hardware partitioning is that it is a heavily vendor-dependent feature. It requires specialized training to produce the processor specific bootup and configuration code necessary to correctly implement a hardware partitioned environment. Support for this training and the related code is solely provided by the hardware vendor who may not be willing to support low volume customers, or the costs associated with that training and support might be prohibitively expensive for smaller businesses or projects with lean budgets. Conversely, software based solutions, like Xen, are supported by many companies like DornerWorks and others, meaning that any company, regardless of size or budget constraints can find the timely support that it needs to be successful. Furthermore, the software solution can be non-proprietary (if an open source product is selected) and is portable between many hardware platforms (true for open source as well many commercial products), as is the training and experience that a company has with that solution.

Two Peas in a Safe and Secure Virtualized Pod

Hardware partitioning introduces some great new capabilities and features, and allows for better, more efficient solutions for previously troublesome use cases. Though some have claimed hardware partitioning as a complete replacement for hypervisors and software

based virtualization, hardware partitioning technology is actually far more capable when combined with existing software technologies, and can act as something of a “force multiplier” in an embedded computing system.

Hardware Assisted Virtualization for Difficult Cases

For example, hardware-based partitioning allows for partitioning of peripherals that are notoriously difficult to share between virtual machines. A great example of this is the split GPU implemented in the i.MX8 by NXP. The System Controller makes it possible to divide the GPU cores among two different virtual machines, giving them access to different video outputs and direct access to the computing power of the GPU. Because of the nature of a GPU, this capability is something rather difficult to implement via software partitioning alone.

Robust Partitioning for Safety and Security Critical Environments

Hardware partitioning can also be used to provide a second level of assurance for safety and security applications such that, even if the software partitioning mechanism were to be compromised, the partitioning between virtual machines would be maintained.

In many systems this means hardware partitioning could be used first to separate high criticality software into a single partition, while the rest of the hardware could be managed by a hypervisor in order to provide resource sharing and achieve maximum processing and I/O throughput (Fig 5).

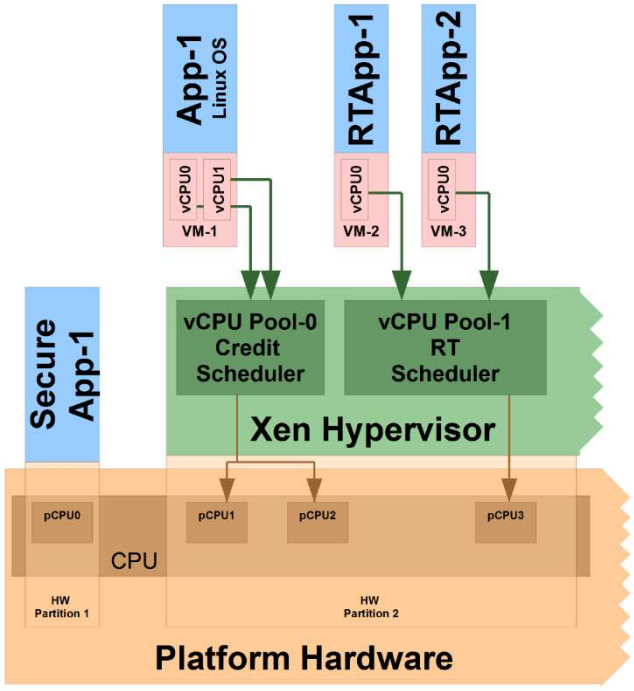


Figure 5 – Combined Hardware Partitioning and Virtualization

Both hardware and software partitioning have their own strengths and weaknesses, but it is the effective combination of these two technologies that can create new, efficient, and novel solutions for embedded computing problems.

Sources:

1. http://arstechnica.com/security/2016/10/using-rowhammer-bitflips-to-root-android-phones-is-now-a-thing/
2. http://www.dailytech.com/NSA+Switches+to+Assuming+Security+Has+Always+Been+Compromised/article20424.htm

By Josh Whitehead

What's Your Next Step?

Now that you are aware of the benefits of and differences in hardware partitioning, you are more prepared to implement the knowledge into your next project.

NXP, Microchip, and others endorse our software and firmware engineering team. We can help your team get to the finish line and work closely with the hardware team to ensure a cohesive solution.

When you need a partner that can provide safe and secure partitioning through software or hardware, DornerWorks can guide you to success. We've helped launch hundreds of virtualized and partition-related products for our customers, focusing on adding engineering expertise to their most innovative ideas, allowing them to focus on what matters most.

DornerWorks is just the kind of partner you're looking for. We would love to help you develop and launch your next project.

Let's Talk Today

About DornerWorks

Product developers know a lot about their products, but often are stressed about the technologies that make those products work. DornerWorks provides expert technology engineering to help them create standout products so that they are free to focus on what they do best.

We engineer technology solutions so that you can focus on your customers and your core expertise.

We help customers develop products:

- System design and consulting
- Product architecture and design
- Test systems engineering

We particularly focus on the platforms:

- Electronic hardware design
- Software and firmware design
- FPGA logic design

You shouldn't have to be an expert in everything.

DORNERWORKS
Technology engineering so you can focus