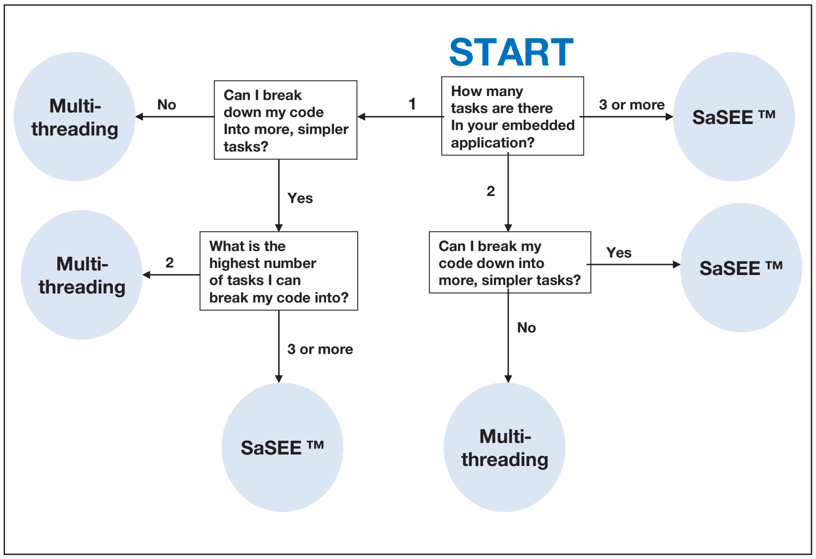
WHITE PAPER

PROCESSOR AND SYSTEM EFFICIENCIES: XYRON’S SaSEE™ TECHNOLOGY VS. MULTITHREADING

Of the issues facing hardware and software engineers in designing embedded applications, processor performance and total system development cost rank highest. With multimedia, networking devices and other system-on-chip products increasing in functionality and complexity, the ability to wring the most out of a processor is, perhaps, a more pragmatic measure of its value than clock speed. Consequently, industry giants such as Intel and upstarts such as Xyron Semiconductor have developed new approaches to increase overall processor and system efficiency while driving down overall system cost.

The focus on processor efficiency stems from the fact that, because of traditional system design, microprocessors spend more time taking care of administrative tasks (such as context switching and task scheduling) than they do on processing actual applications. Depending upon the number of tasks in an application and the number of interrupts it receives, the efficiency of an embedded processor can go below 40%. This means that the processor spends less than 40 of every 100 clock cycles actually processing data, and over 60 clock cycles determining which task it should work on next and switching among tasks - i.e., the microprocessor is operating at less than 40% of its advertised speed.

In addition, because so much software is needed to switch among and schedule tasks, dedicated processors have to be added to a system to handle functionality like video and audio processing, significantly increasing the cost of an application. “Standard” techniques used to address processor inefficiencies include increasing clock speed, dedicating silicon for software tasks, and creating large complex software to avoid task switching. However, these solutions significantly impact development costs, power requirements, and memory requirements.



Deciding Whether Multithreading or SaSEE Technology is Optimal
To determine how to create optimal efficiency in an embedded application, designers have to examine the number of tasks in an application as well as their complexity. The decision tree in Figure One gives a rough summary of how to do this. Because of the increasing complexity of embedded designs, more often than not, SASEE™ technology will be preferred over multithreading. In addition to improving microprocessor performance, SaSEE technology greatly simplifies software design and upgradeability. Software engineers can now write much simpler tasks that are easier to understand, develop and field upgrade. Processing cycles originally used to switch and schedule among tasks can now handle additional functionality in software and can replace other peripheral processors, reducing complexity and saving costs.

Because of this, Xyron has developed innovative techniques to cope with the increasing complexity of embedded designs with its Scheduling and Switching Efficiency Engine (SaSEE™) technology.

SaSEE™ Technology
Simply put, SaSEE Technology moves the task scheduling and switching functionality of an embedded microprocessor from software to hardware so that tasks are managed in the background while the processor spends its cycles solely on the current task. A hardware based priority mechanism determines the highest priority task in the system.

If a lower priority task is currently executing and a higher priority task becomes active, then the processor switches from the lower priority task to the higher priority task. SaSEE technology uses memory termed ‘task RAM’ to store the states of tasks which are waiting to resume execution. The state data of the higher priority task is loaded into shadow registers in the background, and then the processor switches to the new task with zero overhead. The combination of background task scheduling and zero overhead task switching make the microprocessor virtually 100% efficient, focusing only on processing tasks.

When designing applications with SaSEE technology, the embedded designer no longer has to take into account the time an embedded microprocessor spends in task administration. The net result of the time saved? Either more throughput at the same clock rate or reduced clock speed and, thus, lower power. In addition, since switching is free, the designer may be able to divide previously complex code into smaller, highly focused tasks that are much easier to understand, develop, maintain, and field upgrade as products advance from one generation to the next.

Because the processor no longer has to consume overhead in switching from one task to another, CPU cycles originally used to run the scheduling and switching code can now be used to run other applications. These new applications can either provide new functionalities to enhance the end

-over-

product or functionalities originally run by other system processors. Since the need for other dedicated processors may be eliminated, the embedded application designer may realize significant cost savings through reduced hardware requirements, software simplicity and overall system power requirements.

Multithreading

Another, much publicized solution other semiconductor companies have developed to deal with processor inefficiencies is termed “multithreading”. Examples include Intel’s so-called “hyperthreading” and Infineon’s ‘My Virtual Processor (MyVP™). Multithreading implementations vary depending upon the particular solution, but the core features described below are the same.

Multithreading circuitry enables a processor to execute application software that can spawn two threads of code. These two threads of code are run simultaneously. To take optimal advantage of this technology, the two threads must be spawned in a way such that the threads utilize the same resources simultaneously as few times as possible. When this is unavoidable, the processor holds back execution of one of the threads until the resources in question are freed up for use by the other thread. Ideally, multithreading results in a doubling of efficiency. But since it is difficult to spawn two threads of code that don’t use the same resources simultaneously, such efficiency gains will rarely, if ever, be realized. Intel has shown a workstation with hyperthreaded Xeon chips running Alias-Wavefront, a graphics application, achieving a 30% improvement in performance. In general, Intel has said performance improvements from hyperthreading can run as high as 25%.

Comparing SaSEE™ Technology to Multithreading

Choosing the right technology for an embedded design entails two questions: how many tasks are in the embedded code and how complex are they? Task complexity is determined by the potential to divide the task into multiple, simpler tasks that, when run together in an application, perform the same functionality as the original complex task. The more the task can be divided, the more complex it is considered to be.

Take the example of an Ethernet videophone: you can run the application as one complex task combining Ethernet, audio, and video. But this task can be divided into six simple tasks: Ethernet in, Ethernet out, audio in, audio out, video in, and video out, which is the way it would run on SaSEE technology. In general, smaller, simpler tasks execute faster and are easier to code than large, complex tasks- hence, they enable greater overall efficiencies.

However, if an embedded application runs only one or two simple tasks, multithreading is probably a good option for increased performance. SaSEE technology provides little benefit to this type of application because there is very little switching or scheduling overhead that can be eliminated. Also, with only one or two simple tasks, it is unlikely that all the resources of a microprocessor are constantly being utilized. Therefore, the application designer will be able to design code spawning two threads with few circumstances where each thread requires the same resources simultaneously. However, most embedded, multimedia applications are not this simple.

As the number of complex tasks in an embedded application rise, so does the need for SaSEE Technology, starting with embedded applications having one or two complex tasks divisible into many more, simple tasks. Not only does “decomplexification” improve performance, it also makes the code easier and faster to develop as well as upgrade. In a SaSEE technology-enabled processor, the embedded designer is no longer concerned with the number of times scheduling and switching occurs because the SaSEE hardware manages the tasks when required. The designer only needs to set relative priorities and initial system constraints - which are upgradeable at any time.

When an embedded application code has three or more disparate tasks, SaSEE technology is the odds-on favorite. Why? For applications with this many tasks, multithreading will not dramatically improve the processor efficiency over a processor with only single thread capability - and software management becomes dramatically harder. With three or more tasks and SaSEE technology, the designer may be able to rewrite application code so that it is divided into more tasks that are easier to understand and upgrade. Again, because the designer no longer has to worry about the processor wasting valuable clock cycles switching among or scheduling tasks, the processor will efficiently process those tasks when necessary.

Increasingly, today’s embedded designs converge multiple applications into one ‘super-application’ - e.g., networking and multimedia - having larger numbers of complex tasks where SaSEE technology can increase processor efficiencies by as much as an order of magnitude. For super-applications multithreading provides negligible if any improvements. In addition, SaSEE technology gives the embedded application designer tremendous cost savings by eliminating specialized silicon; it also enables software flexibility and upgradeability, which are not possible with multithreading.

Conclusion

That Intel and Infineon developed multithreading to increase processor efficiency only validates the need to look at other, new methods for processor and system efficiencies for increasingly complex applications. Although both multithreading and SaSEE technology add circuitry to a microprocessor to improve efficiency, the former only addresses “simple” applications with two threads that can be processed simultaneously when they aren’t utilizing the same resources. SaSEE technology reexamines the basic way that processors handle multitasking and eliminates the time spent scheduling and switching among tasks. It can also increase the potential of the processor many multiples in highly interrupted, multitasking applications.

xyron semiconductor

201 NE Park Plaza Drive
Suite 165
Vancouver, WA 98684
360.449.8800
fax 360.449.8850

Mailing Address
PO Box 871808
Vancouver, WA 98687-1808

For more information,
visit xyronsemi.com