

# Proposing the Architecture for a High-performance GPS-enabled Handheld Cartographic Display System

Igor Ikodinović, Veljko Milutinović

Zoran Dimitrijević

Antonio Cosimo Prete

Department of Computer Engineering  
School of Electrical Engineering  
University of Belgrade  
P.O. BOX 35-54  
11120 Belgrade, Serbia, Yugoslavia

Department of Computer Science  
University of California at  
Santa Barbara  
Santa Barbara, CA 93106

Department of Electronics,  
Informatics and Telecommunications  
University of Pisa  
Via Diotisalbi 2, 56100 Pisa, Italy

igi@eunet.yu, vm@etf.bg.ac.yu

zoran@cs.ucsb.edu

prete@iet.unipi.it

## Abstract

*A high-performance handheld system needs fast graphics, good communication capabilities, and outstanding computational performances. In addition, low power consumption, small size, and low price are desirable. The existing high-end GPS-based cartographic display systems can not fully meet all these requirements mainly for the following two reasons: 1) Systems are typically based on standard uniprocessor architectures or its versions, which are not well suited for applications involving real-time processing of multiple data streams (a predominant type of processing in such systems), and 2) The existing systems are mostly built around several separate microprocessor and companion chips, which limits the level of system integration (and its benefits).*

*As an alternative to the existing solutions we proposed a system with a bus-based symmetrical multiprocessor (SMP) architecture and investigated its two main versions: a cacheless and a cache-coherent system. As expected, cacheless system exhibited slightly lower performances than its cache-coherent peer. However, the gain in performances obtained by adding processor caches to the cacheless system can not justify a significant increase of the chip complexity. To address the problem of system integration, the design was based on ready-made proprietary microprocessor cores and logic blocks, and the whole system is incorporated on a single chip.*

## 1. Introduction

Handheld cartographic display systems for precise positioning and navigation via GPS are basically used to show the current user position on the screen where a cartographic map of the area is displayed. However, their use is not limited to that basic function. They often include various navigational tools, databases, communication services, and various software applications. It requires substantial processing power to perform all these tasks in a satisfactory manner.

While fixed systems can be built around a standard personal computer platform, handheld systems are designed with completely different requirements. Limited dimensions and the need for low power consumption make standard digital components inadequate for use in such systems. Instead, either commercial low-power microprocessors, or custom designed chips and components are used. Designing systems with off-the-shelf microprocessors and components reduces design time, but limits the possibility of system integration and benefits that it brings, like reduced power consumption, reduced size of the system, and increased processing power. Custom chip design, on the other hand, exacts additional design effort, but enables increased level of system integration. Such approach is especially beneficiary if based on ready-made (usually proprietary) microprocessor cores and logic blocks. It significantly reduces design time and allows designers to focus on the matter of system architecture rather than component design.

Conventional uniprocessor architectures, often used in commercial systems, are not well suited for real-time processing of data streams, which is a predominant type of processing in advanced GPS-based cartographic display systems [1]. To achieve higher performances some manufacturers have designed systems with unorthodox architectures. These solutions include on-chip caches, SRAM, additional processors, DSPs (Digital Signal Processor), graphics accelerators, or other resources.

Even simple handheld systems need some sort of embedded operating system to drive them. Developing a new operating system for a specific platform is difficult and inefficient. A widely adopted solution to this problem is to adapt an existing operating system to work on a specific platform. Nevertheless, setting up the embedded OS is a time-consuming task. Since technology improves at a fast rate and the competition on the market is sharp, systems quickly become obsolete. Manufacturers must periodically improve system design or even introduce completely new architectures. Each such change requires additional time to adapt the OS to the new platform.

Having all this in mind, we tried to propose the architecture that can conform to the following requirements: 1. utilizes the benefits of integration and design based on ready-made microprocessor cores and logic blocks, 2. enables efficient execution of various types of applications including processing of multiple data streams, and 3. requires only minimum or no effort to adapt the OS as the system improves. A solution was found in a bus-based symmetrical multiprocessor architecture (SMP). Its two main versions, a cacheless SMP and a cache-coherent SMP, were evaluated by running a number of simulations using an adequate workload, and then were compared from the performance/complexity point of view.

The paper is organized as follows: In Section 2 we discuss the requirements that a high-performance GPS-enabled cartographic display system should fulfill and then investigate the existing solutions and their deficiencies, including the best known solution. In Section 3 we describe the bus-based SMP architecture and explain the reasons that make it suitable for a highly integrated system design, with all components incorporated on a single chip. This architecture is evaluated with a series of simulations, whose results are presented and discussed in Section 4. The conclusions are given in Section 5.

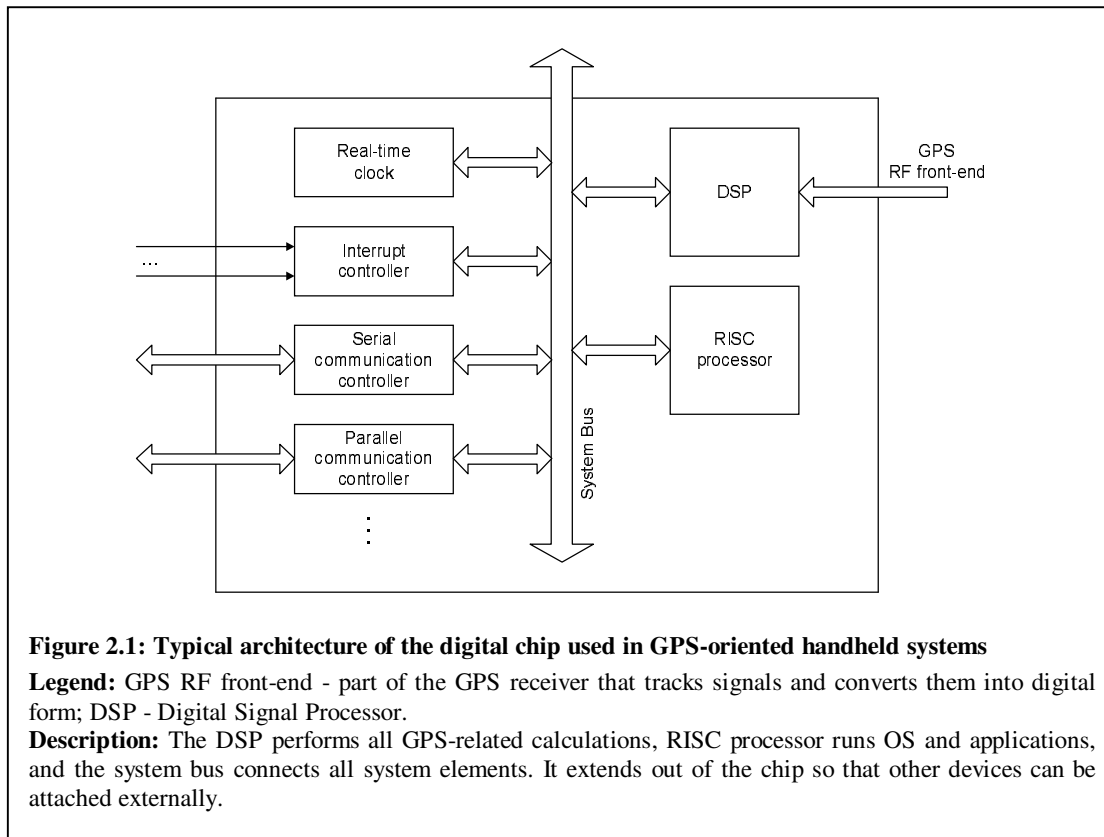
## **2. Searching for the Right Architecture: Evaluation of the Existing Solutions**

A high-performance GPS-enabled handheld cartographic system needs good quality graphics, must be able to support intensive communication using a number of standard interfaces, and possess excellent computational capabilities for fast and accurate calculation of position via GPS. In addition, it should have low power consumption for long autonomy of use, small size for easier handling, and an affordable price to attract enough consumers. Some of these requirements are contradictory and ask for a compromise.

### **2.1 Processing Power: Limits of the Existing Architectures**

A typical GPS-enabled handheld system [2] includes only few chips: RF front-end, digital chip (that performs all data processing), various device controllers, and external RAM and ROM. The RF front-end is important for GPS-related performances, but the digital chip and components are crucial for the overall performance of the system.

The digital chip in a typical commercial system (Figure 2.1) includes a single bus, general-purpose RISC microprocessor, an attached DSP, and accompanying elements such as serial and parallel communication controllers, interrupt controller, real-time clock, possibly on-chip RAM, etc. Since low power consumption is a necessity in handheld devices, most of the existing GPS-oriented chips have power-saving features. Processor cores with low power consumption are used in the system design, and latest chip fabrication technologies are used in an attempt to reduce power consumption, increase speed and decrease die area. Sometimes even a special low-power controller is included on the chip.

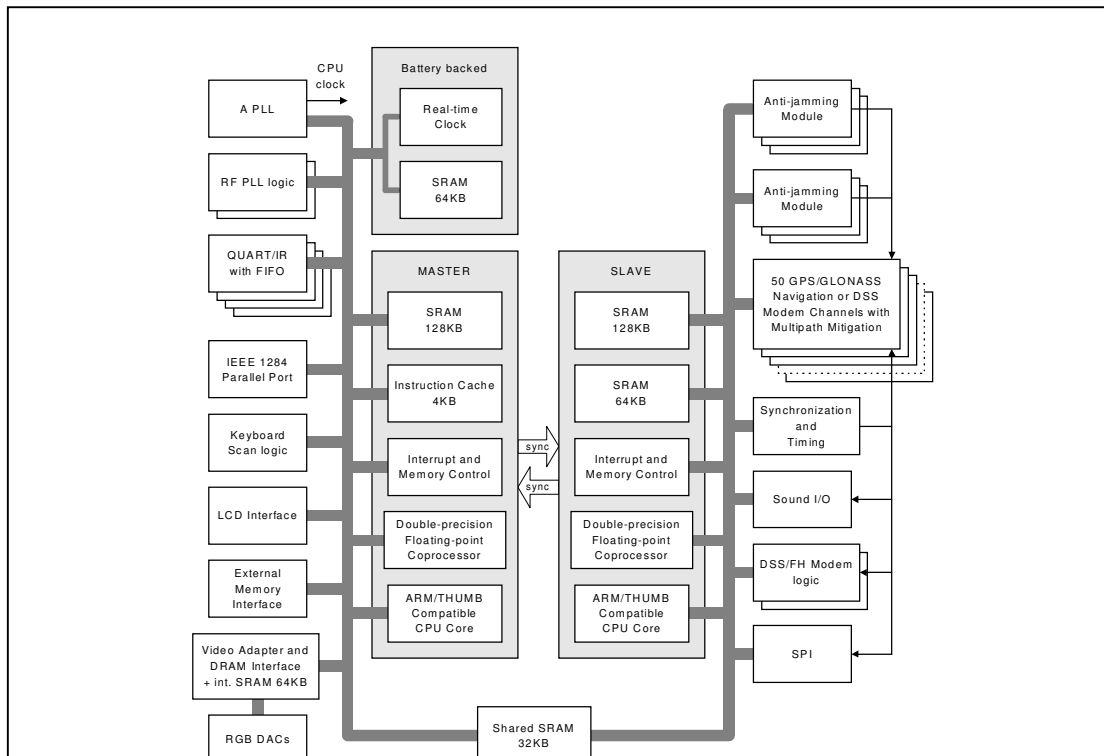


The architecture in Figure 2.1 is an extension of a classical uniprocessor architecture. The DSP takes the load off the general-purpose microprocessor and performs computationally intensive calculations of position. However, if the system needs to handle tasks like intensive communication (possibly a large number of communication controllers can be in the system), or if it needs to drive a large display, manage multimedia contents, or work with larger databases, it can be difficult to process such a multitude of data. In a typical configuration only the most critical task of calculating GPS position is isolated and handled by a specific hardware (the DSP), but in advanced systems there is, in many ways critical, issue of capability of the microprocessor to deliver sufficient processing power for all other tasks.

To increase processing power, some designers use state-of-the-art microprocessors (typically a multistage pipelined RISC core with instruction and/or data cache). The most advanced solutions include a second general-purpose microprocessor and additional floating-point coprocessor units instead of the traditional single-purpose DSP [1,3] (see Figure 2.2). Including more than one processing unit in the system is a possible solution to the problem of insufficient processing power. In the coming era of broadband mobile communications and multimedia, the issue of processing power will only become more critical [1,3,4].

## 2.2 Embedded OS: Custom Software Still Used Instead of Standard OS

In the past, handheld systems predominantly featured simple uniprocessor architectures. They were relatively easy to program. It was a common practice to develop system software and define custom user interface from the scratch, especially because there were no standard embedded operating systems available. However, things have changed with time. The architectures of handheld systems have become more complex (see Figure 2.2) and thus harder to program directly. At the same time sharp competition on the market leaves little time to develop system software and applications from the scratch. A different approach is used instead: an existing OS is adapted to run on the specific



**Figure 2.2: Best of the existing solutions - JPSScore<sup>®</sup> chip architecture**

**Legend:** JPSScore<sup>®</sup> is a trademark of Javad Positioning Systems, recently acquired by Topcon Positioning Systems (<http://www.topconps.com>)

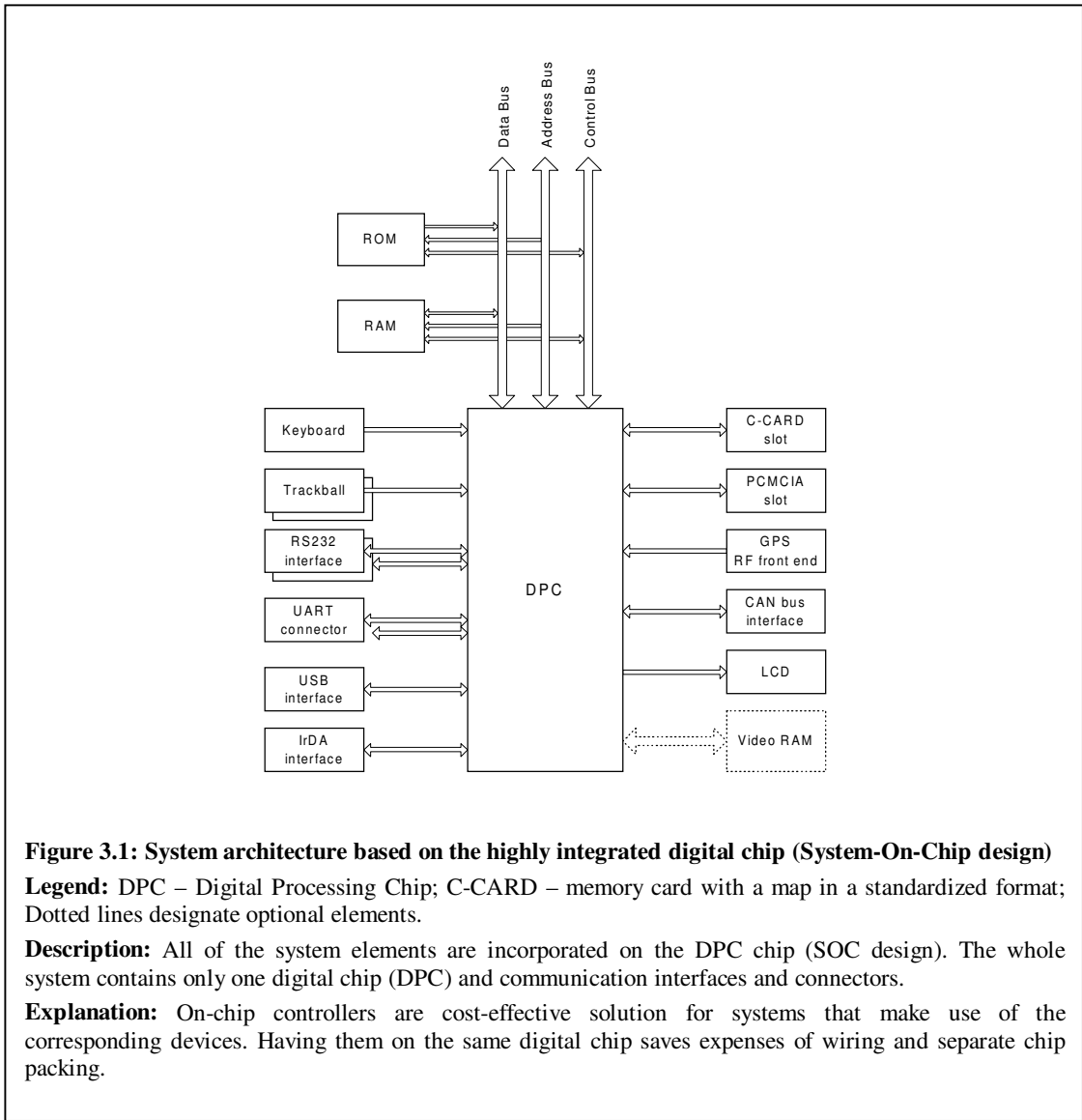
**Description:** JPSScore<sup>®</sup> is one of the most advanced and complete digital GPS/GLONASS chips for general use. Its design extends the conventional bus-based architecture with another general-purpose microprocessor (ARM7TDMI microprocessor core instead of the DSP), on-chip SRAM, FPUs for both processors, and instruction cache for the master processor. In addition, a multitude of device controllers is placed on the chip.

**Explanation:** Assuming that device controllers occupy relatively small die area, compared to the area occupied by the two microprocessor cores, cache, and SRAM, they are inexpensive to incorporate on the chip. This is a good example of the System-On-Chip design.

platform [5], so all applications can be developed using the existing tools and libraries. Moreover, a multitude of existing and legacy applications can be immediately run on the new platform. A cross-platform compatibility is obtained by default. However, there are still few GPS-oriented handheld systems driven by a standard OS.

### 2.3 Integration: System-On-Chip Design Still Rare

Vendors of commercial GPS-dedicated chips tend to include in their design only the resources that will be needed by the majority of GPS systems on the market. For example, since many devices that make use of the GPS have only rudimentary display capabilities, like wrist watches, mobile phones, and simple outdoor GPS receivers, it would be costly to have a full graphics controller incorporated on the same digital chip with the rest of the system. A typical system thus employs a graphics controller as a separate chip. Another reason to have graphics controller as a separate chip is that it is device-dependent, i.e. it is designed according to the type of the display it drives. For similar reasons most other device controllers are often not incorporated on the same chip with the rest of the system. Only the most common controllers are included. This largely reduces possibility of system integration.

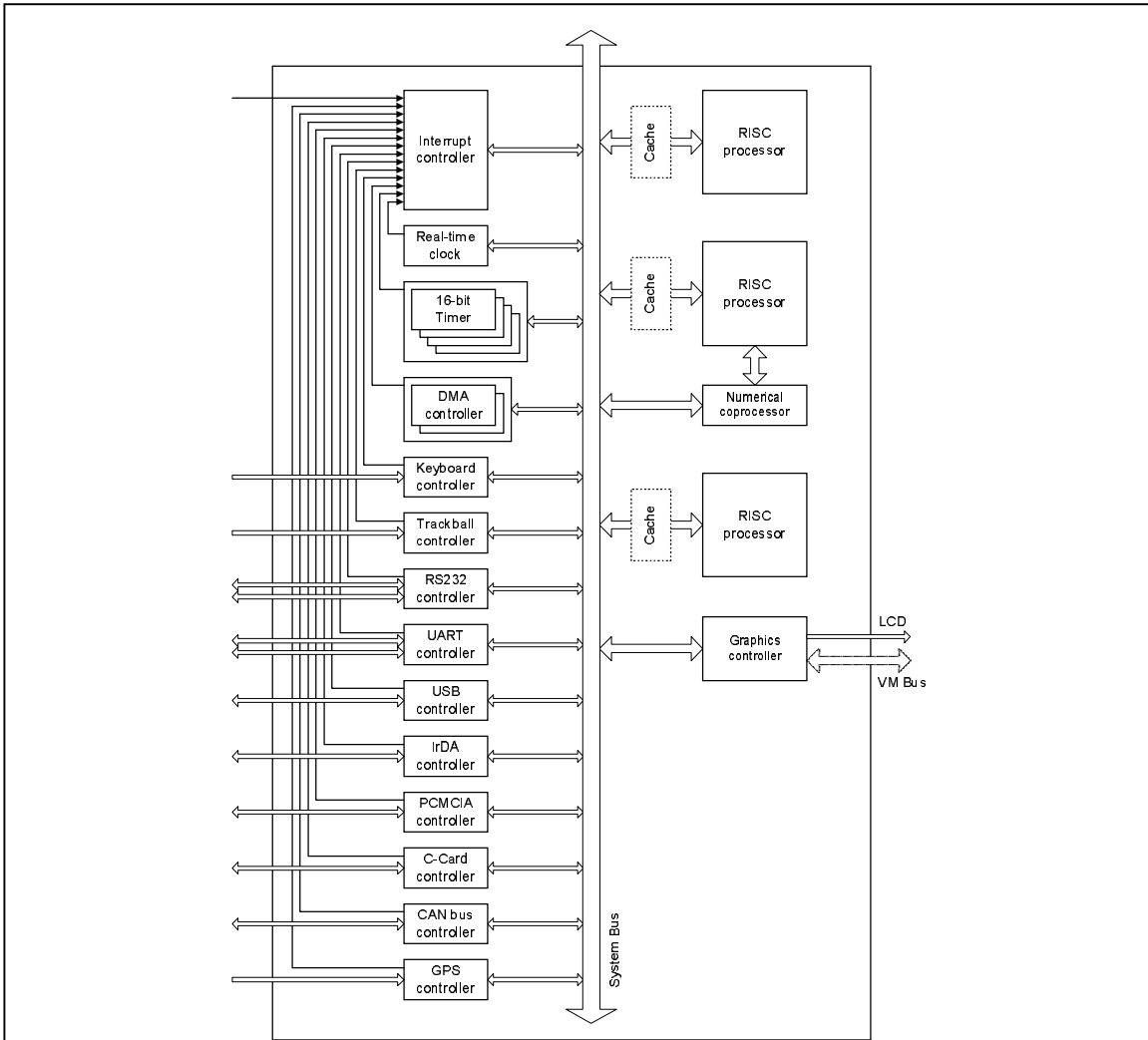


Since device controllers usually do not occupy large die area, for the benefit of system integration it can be justified to incorporate them on a single digital chip with the rest of the system, as an additional functionality (see Figure 2.2). The cost of occupying additional die area is often by far exceeded by the gain in performances, power consumption reduction, and reduction of the system size.

Most of the existing systems are not designed as System-On-Chip (SOC). Designing a SOC requires transition from a level where chips are used as building blocks to design System-On-Board to a level where logic blocks are used as building blocks to design System-On-Chip.

### 3. A Novel Approach to System Architecture: SMP on a Chip

Integrated circuit processing technology keeps improving at a fast rate. At the current integration level there are plenty of transistors on the chip for SOC design. The main question is how to use the chip resources in the most effective way, having in mind the type of workload in a GPS-oriented handheld system. Researches in this field [3,6] show that a multiprocessor chip architecture has better performance, complexity, and power consumption indicators when compared to most other



**Figure 3.2: The SMP architecture of the digital chip**

**Legend:** Cache – Processor cache; VM Bus – Bus that connects the graphics controller to the video memory; Dotted lines designate optional elements.

**Description:** Numeric coprocessor is added to the GPS dedicated processor. VM bus can be included for communication with external video memory. Caches can be added for better processor performance.

**Explanation:** Calculation of position via GPS is a computationally intensive task, so a FP coprocessor is added. Graphics-related activities induce significant bus traffic, which can impair system performance, especially if memory access is slow. Problems with bus traffic can be avoided if video memory is separated from the main memory. Further improvement in graphics processing can be accomplished by adding the accelerator capabilities to the graphics controller. Simulations were performed to determine if processor caches are needed. Introducing caches arises the question of cache coherence. It is necessary to add tags to the cache lines to support a cache coherence protocol.

types of architectures, including the advanced uniprocessors<sup>1</sup>. Thus an advanced GPS-oriented handheld cartographic display system should be designed in a SOC manner with a multiprocessor chip architecture (see Figure 3.1).

<sup>1</sup> All architectures that were compared in the researches have similar complexity, so that comparison is fair. The Vector IRAM chip architecture [3] performs better than a multiprocessor in some aspects (like power consumption); however, this architecture is still in a phase of scientific evaluation, and not a commercially viable solution.

A bus-based SMP (Symmetrical Multi-Processor) is a natural extension of the bus-based uniprocessor architecture. SMP can be designed with commercial low-power low-complexity processor cores, which can significantly reduce the effort needed to design the system. It is also a scalable architecture, which allows that applications that are once written in a multithreaded fashion can be used in systems with any number of processors. The most widely used embedded operating systems support multithreading on SMP platforms, so that a large base of the existing development tools and application programs is already available.

### 3.1 Complexity

As an example, we considered the ARM7 family of widely used commercial microprocessor cores and accompanying logic cells. The core itself counts around 50k gates. As can be noted from Table 1, cached versions have significantly higher complexity (occupy approx. 5 times bigger die area) and power consumption (approx. 3 times bigger power consumption). This raises the question whether caches can increase performance in accordance with the extreme increase of the complexity and power consumption they require.

CPU Core	Die Area	Power (mW/MHz)	Clock Frequency and Performance		CPU Core	Cache	Memory Management
ARM7TDMI ARM RISC Core with Thumb and EmbeddedICE	1.0mm <sup>2</sup> on 0.25µm 2.1mm <sup>2</sup> on 0.35µm 4.8mm <sup>2</sup> on 0.6µm	Peak: 1.2mW/MHz Ave: 0.6mW/MHz Idle: <100µw @ 3.3V, 0.35µm CMOS	66MHz on commodity 0.35µm CMOS	0.9 MIPS/MHz 59MIPS @ 66MHz	N/A	N/A	N/A
ARM710T Cached Processor Macrocell	5.8mm <sup>2</sup> on 0.25µm 11.7mm <sup>2</sup> on 0.35µm	Peak: 3.6mW/MHz Ave: 1.8mW/MHz Idle: <100µw @ 3.3V and with Cache ON. 0.35µm CMOS	59MHz on commodity 0.35µm CMOS	53MIPS @ 59MHz	ARM7TDMI	8K Unified Cache	MMU giving full virtual memory support
ARM720T Cached Processor Macrocell with MMU for WindowsCE	5.8mm <sup>2</sup> on 0.25µm 11.7mm <sup>2</sup> on 0.35µm	Peak: 3.6mW/MHz Ave: 1.8mW/MHz Idle: <100µw @ 3.3V and with Cache ON. 0.35µm CMOS	59MHz on commodity 0.35µm CMOS	53MIPS @ 59MHz	ARM7TDMI	8K Unified Cache	MMU giving full virtual memory and fast context switching support
ARM740T Cached Processor Macrocell	4.9mm <sup>2</sup> on 0.25µm 9.8mm <sup>2</sup> on 0.35µm	Peak: 3.5mW/MHz Ave: 1.6mW/MHz Idle: <100µw @ 3.3V and with Cache ON. 0.35µm CMOS	59MHz on commodity 0.35µm CMOS	53MIPS @ 59MHz	ARM7TDMI	8K Unified Cache	Simple Memory Configuration and Protection

**Table 1: ARM CPU core complexity review (Data from ARM web site <http://www.arm.com>)**

Complexity of the standard device controller cells (given in Table 2) indicate that incorporating device controllers increases overall chip complexity by a factor that is almost an order of magnitude lower than the chip's original complexity (especially if cached cores are used). It is thus justified to incorporate the device controllers in the chip design.

Description	Approximate gate count
UART w/ IrDA SIR: Similar to 16C550, 16Byte FIFO, up to 115K2 bits/s	7.7k
Synchronous Serial I/F: Supports Motorola SPI, TI SSI, Microwire	7.4k
Real Time Clock: 32 Bit Counter, Match Reg, Requires 1Hz Ck	2.8k
Audio Codec I/F: 8-bit, 16byte FIFO, Prog Data Rate	4.5k
Keyboard/mouse I/F: PS/2 compatible	1.9k
General Purpose IO: 2x8bit	0.8k
DC to DC Converter Interface: 1.8MHz, 900, 225, 96 kHz Prog O/P	1.4k

Smartcard Interface: Compliant with the EMV Standard and ISO 7816-3	12.3k
Generic IR Interface: Capable Tx or Rx a modulated carrier, or direct digital signal	10.5k

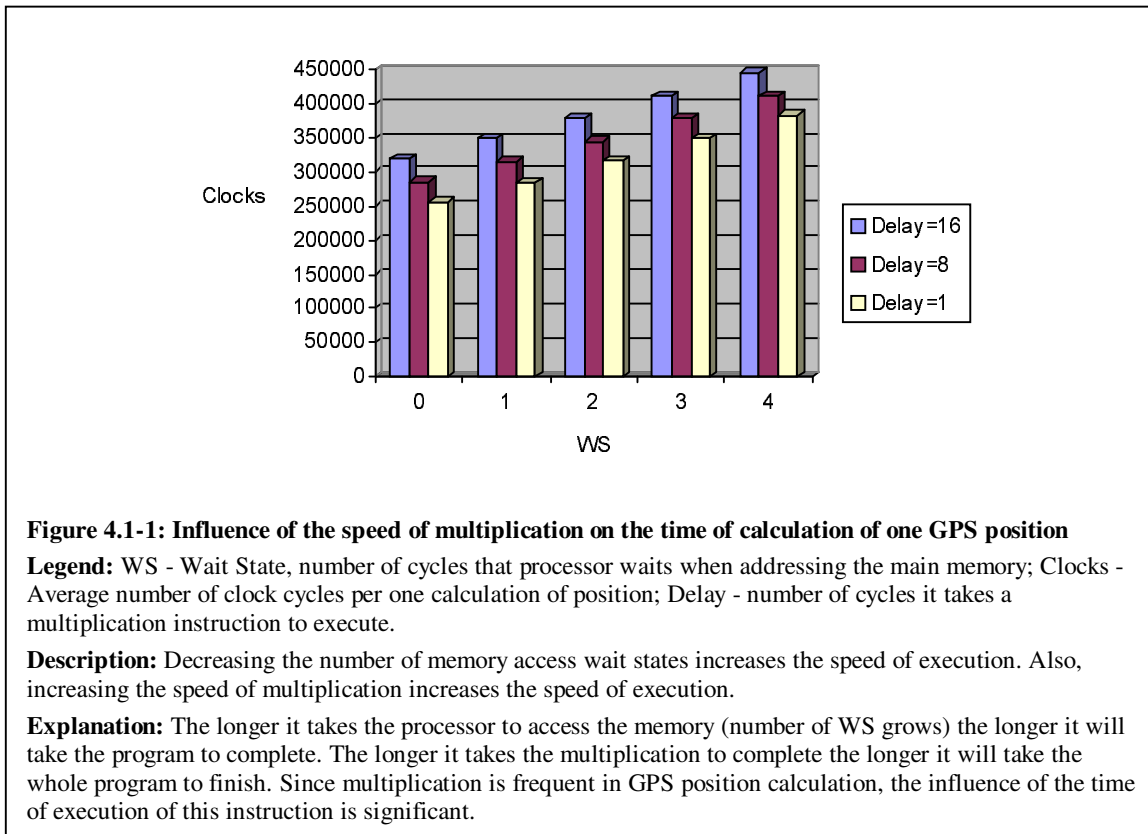
**Table 2: Approximate gate count for standard device controller cells (Data from ARM web site <http://www.arm.com>)**

## 4. Evaluation of the Proposed Solution: SMP Architecture in a GPS World

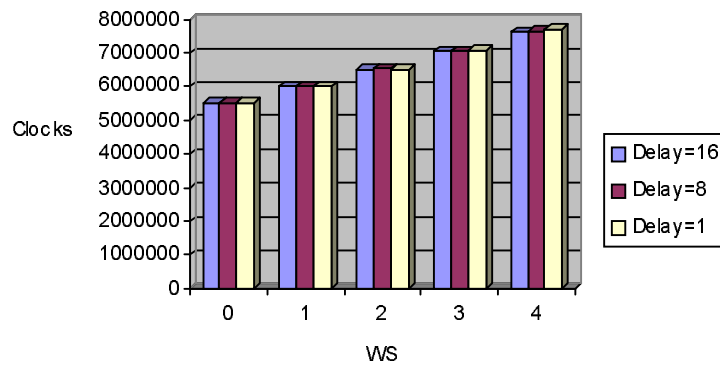
Simulations were performed using the Limes multiprocessor simulation environment [7] to simulate the execution of a custom designed test workload on a particular SMP architecture. The test workload is designed to includes three general areas of importance: graphics, communication, and calculations of position via GPS.

Three parameters were measured for both cacheless and cache-coherent systems: execution time of of a single calculation of position via GPS, execution time of a graphically intensive benchmark, and bus traffic. These three parameters show how the three most important beforementioned tasks influence the system. Since multiplication is by far the most time consuming FP operation in a GPS system, we measured the three parameters for various multiplication execution times. It was a good indication of required FP performances.

### 4.1 Simulation Analysis of a Cacheless SMP





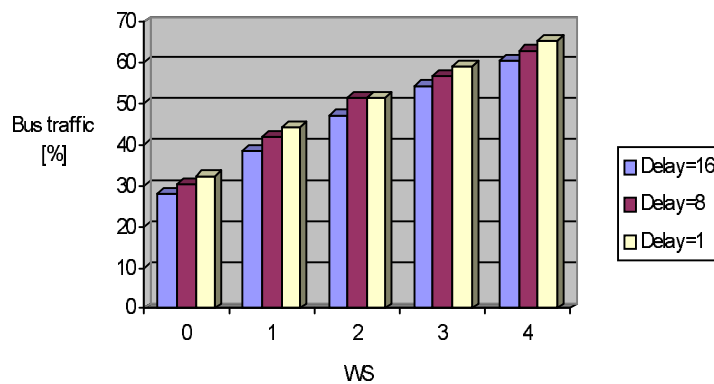


**Figure 4.1-2: Influence of the speed of multiplication on average time of execution of a graphically intensive benchmark program**

**Legend:** WS - Wait State, number of cycles that processor waits when addressing the main memory; Clocks - Average number of clock cycles per execution of benchmark program; Delay - number of cycles it takes a multiplication instruction to execute.

**Description:** Decreasing the number of memory access wait states increases the speed of execution. Decreasing the speed of multiplication increases the speed of execution of benchmark program.

**Explanation:** The longer it takes the processor to access the memory (number of WS grows) the longer it will take the program to complete. The shorter it takes the multiplication to complete the faster the GPS positions are calculated, which increases the bus traffic. Since multiplication is frequent in GPS position calculation and not frequent in the graphically intensive code, the influence of the time of execution of multiplication will negatively impact the execution of benchmark program. This is due to the increased buss traffic that causes the benchmark program to stall more than the faster multiplication accelerates it.



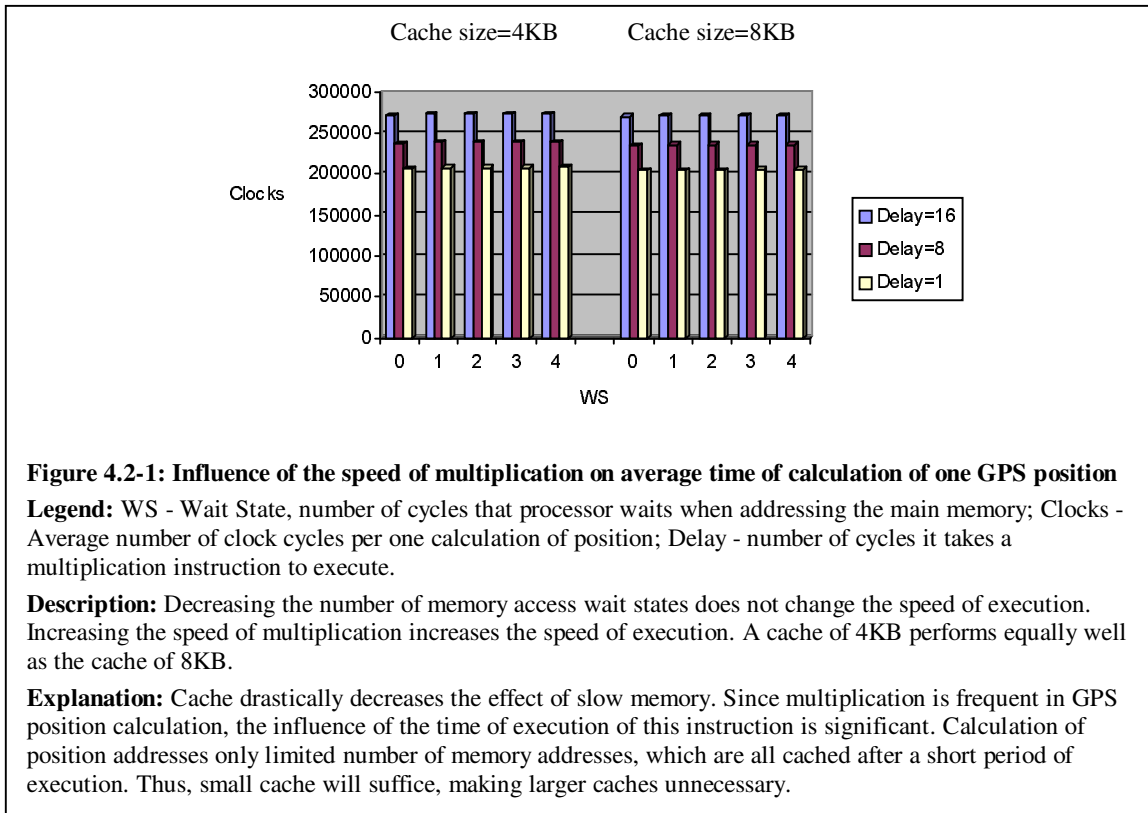
**Figure 4.1-3: Influence of the speed of multiplication on bus traffic**

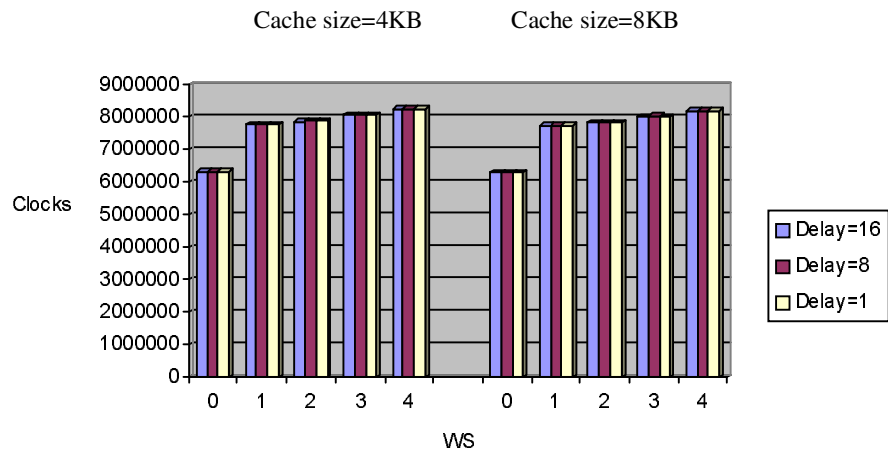
**Legend:** WS - Wait State, number of cycles that processor waits when addressing the main memory; Delay - number of cycles it takes a multiplication instruction to execute.

**Description:** Increasing the number of memory access wait states increases the bus traffic. Also, increasing the speed of multiplication increases the bus traffic.

**Explanation:** The longer it takes the processor to access the memory (number of WS grows) the greater the bus traffic. The shorter it takes the multiplication to complete the GPS calculations are more frequent, which increases the bus traffic.

## 4.2 Simulation Analysis of a Cache-coherent SMP



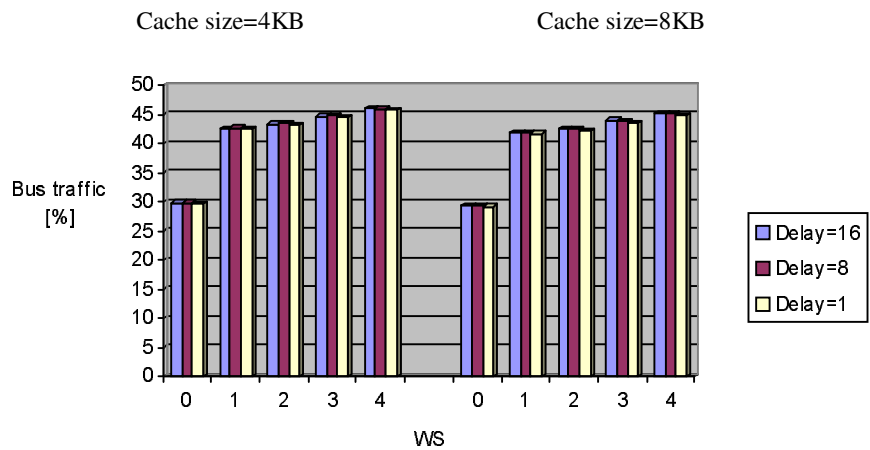


**Figure 4.2-2: Influence of the speed of multiplication on average time of execution of a graphically intensive benchmark program**

**Legend:** WS - Wait State, number of cycles that processor waits when addressing the main memory; Clocks - Average number of clock cycles per execution of benchmark program; Delay - number of cycles it takes a multiplication instruction to execute.

**Description:** Decreasing the number of memory access wait states increases the speed of execution. Decreasing the speed of multiplication virtually does not influence the speed of execution of benchmark program. A cache of 4KB performs the same as the cache of 8KB.

**Explanation:** Cache drastically decreases the effect of slow memory in this case as well as for the GPS calculations. Since multiplication is not frequent in the graphically intensive benchmark program, the influence of the time of execution of the multiplication instruction is insignificant. Contrary to the situation with GPS calculations, here both 4KB and 8KB caches are too small to accept all the data. Thus, the caches will perform equally bad.



**Figure 4.2-3: Influence of the speed of multiplication on bus traffic**

**Legend:** WS - Wait State, number of cycles that processor waits when addressing the main memory; Delay - number of cycles it takes a multiplication instruction to execute.

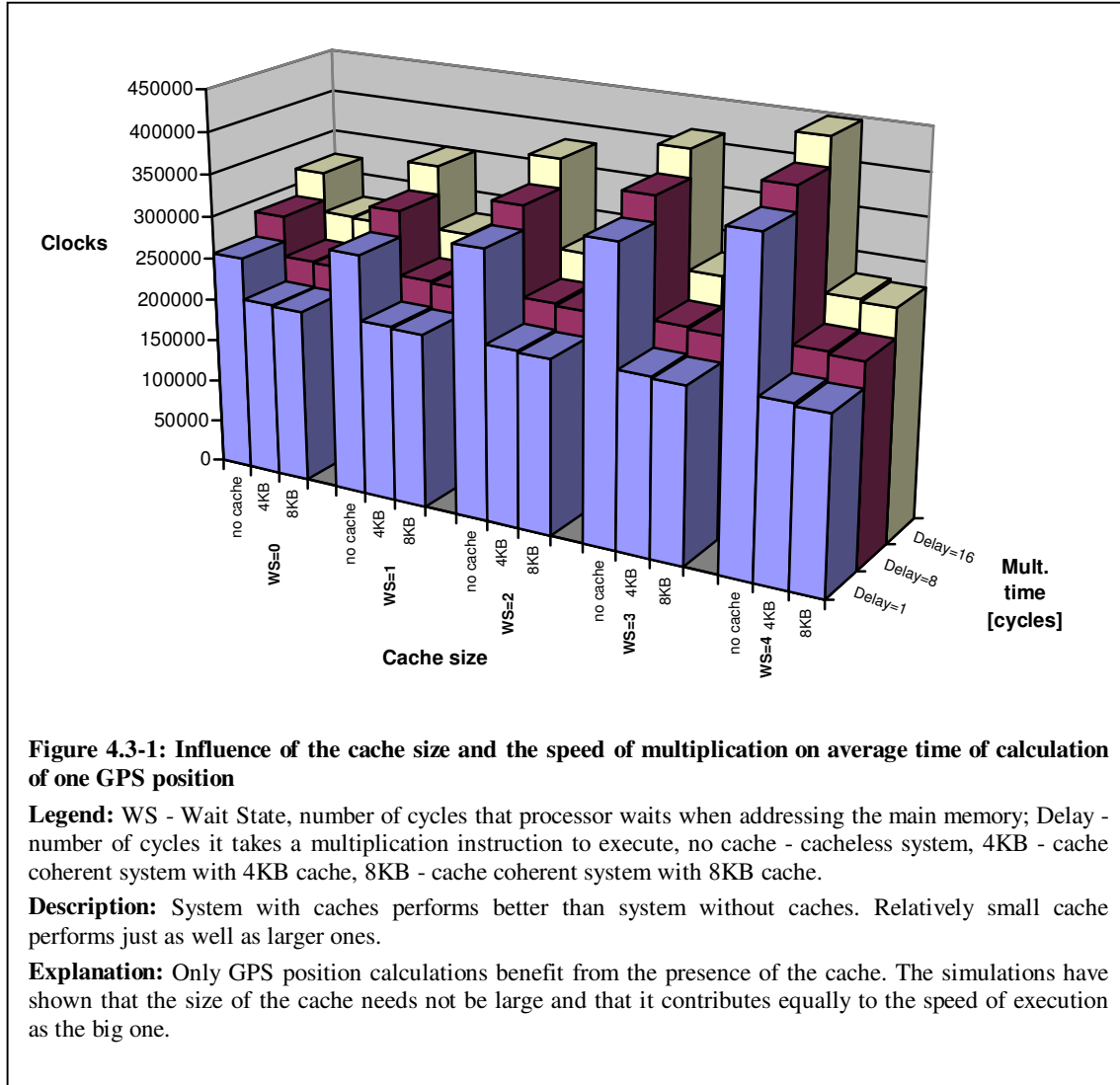
**Description:** Increasing the number of memory access wait states increases the bus traffic. Increasing the speed of multiplication virtually has no influence on the bus traffic.

**Explanation:** The longer it takes the processor to access the memory (number of WS grows) the greater the bus traffic. Since multiplication is not frequent in the graphically intensive benchmark program, the influence of the time of execution of the multiplication instruction is insignificant.

### 4.3 Conclusions: Cacheless SMP Has a Good Performance/Complexity Ratio

Figure 4.3.1 shows that (in case of a slow memory access) a small cache significantly improves the performance of GPS calculations. ARM7 RISC microprocessor cores considered for use in this project come only in versions with 8KB cache, or without cache, where cached versions occupy 5-6 times larger die area than non-cached ones. Since graphics is not improved if a small cache is used, but only by using a large one where all the data can fit, it is a matter of performance/complexity tradeoff whether only a cached processor core shall be used to improve GPS calculations, or a larger on-chip zero-WS SDRAM that improves both GPS calculations and graphically intensive tasks.

The simulations have shown that ARM processor cell with incorporated cache and multiplier is best suited for GPS calculation. Small cache is not well suited for graphic library functions, and integrated SDRAM or fast off chip RAM can improve drawing speed. In case that fast memory is used, cache is not needed. Depending on the GPS system complexity the third ARM processor cell is needed to increase the communication speed and OS performance.

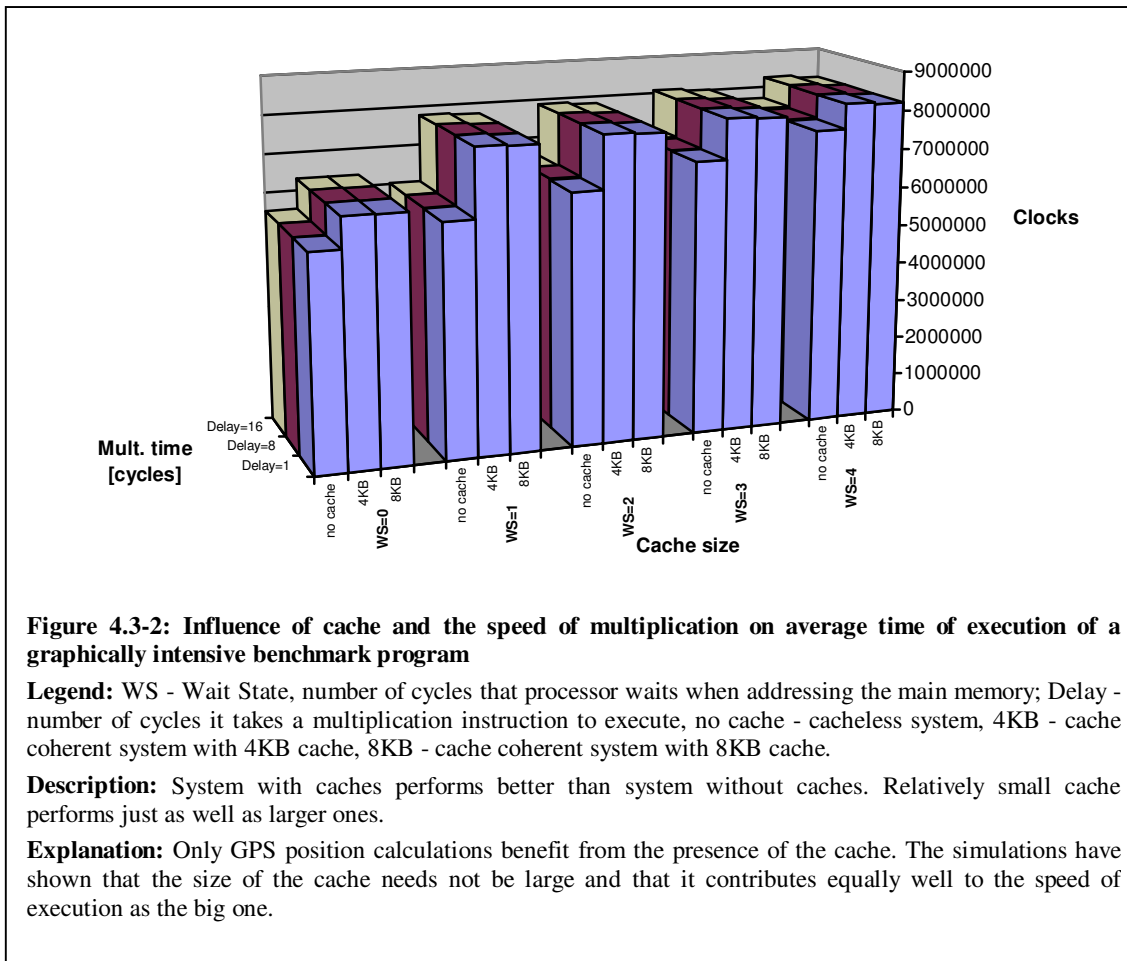


**Figure 4.3-1: Influence of the cache size and the speed of multiplication on average time of calculation of one GPS position**

**Legend:** WS - Wait State, number of cycles that processor waits when addressing the main memory; Delay - number of cycles it takes a multiplication instruction to execute, no cache - cacheless system, 4KB - cache coherent system with 4KB cache, 8KB - cache coherent system with 8KB cache.

**Description:** System with caches performs better than system without caches. Relatively small cache performs just as well as larger ones.

**Explanation:** Only GPS position calculations benefit from the presence of the cache. The simulations have shown that the size of the cache needs not be large and that it contributes equally to the speed of execution as the big one.



## 5. Conclusion: Architecture for the Future

Advances in fabrication technology inevitably drive handheld GPS-enabled cartographic display systems toward greater system integration. Integration reduces size, power consumption, and the cost of the system, and at the same time increases system performances and usability. The ultimate designers' goal is to get a System-On-Chip.

Commercial microprocessor cores and logic blocks are already implemented as part of many GPS-dedicated chips. In the future, the design will be probably heavily based on use of digital logic cores, whether they come from within the same company or from third-party vendors. Although multicore chip design rises the questions of system testability [8] and compatibility [9], using commercially available general-purpose microprocessor cells can significantly reduce the time of development of embedded systems. This can be an important advantage on the market, where products must appear in time, and possibly before the competition.

Increase in the chip complexity must be followed by the appropriate architectural solutions. It is evident that multiprocessing is entering the world of embedded applications. Even though it is still an advanced solution that can be found only in one or two products on the market, it may soon be widely adopted in powerful systems that offer high-precision positioning and navigation, and excellent graphics. SMP systems are naturally scalable, and once the OS and applications are written for it, it

will by default support any future improvements in hardware. It will be easily possible to add processors with only minimal changes to the system architecture. This, in turn, will save design time, and reduce the cost of development. Surprisingly, it may happen that a low complexity SMP on the chip first appears in embedded environment, and not in the server or supercomputing domain.

Embedded operating systems are already adapted to work on many handheld systems, enabling a multitude of application programs to run on these platforms. The trend of integration will eventually lead toward merging of GPS-based cartographic display systems with other handheld systems into a universal mobile platform that will incorporate the functionality of today separate systems.

## 6. References

- [1] K. Diefendorf, P. K. Dubey, "How Multimedia Workloads Will Change Processor Design," *Computer*, September 1997, pp. 43-45
- [2] I. Ikodinovic, Z. Dimitrijevic, V. Milutinovic, Antonio Cosimo Prete, "GPS-enabled Handheld Cartographic Display Systems: A Survey," Technical Report, 1999, [Online,] Available WWW: <http://galeb.etf.bg.ac.yu/~dsm/gps>
- [3] C. E. Kozyrakis, D. A. Patterson, "A New Direction for Computer Architecture Research," *Computer*, November 1998, pp. 24-32
- [4] D. Burger, J. R. Goodman, "Billion Transistor Architectures," *Computer*, September 1997, pp. 46-48
- [5] Joel R. Williams, "Embedding Linux in a Commercial Product," *Linux Journal*, October 1999, Issue No 66, Available WWW: <http://www2.linuxjournal.com/lj-issues/issue66/3587.html>
- [6] L. Hammond, B. A. Nayfeh, K. Olukotun, "A Single Chip Multiprocessor," *Computer*, September 1997, pp. 79-85
- [7] I. Ikodinovic, A. Milenkovic, V. Milutinovic, "Limes: A Multiprocessor Simulation Environment for PC Platforms," 3<sup>rd</sup> International Conference on Parallel Processing and Applied Mathematics, September 1999, Kazimierz Dolny, Poland, pp. 398-412
- [8] J. Lipman, "Add testability now to core-based chips, or pay later," *EDN*, February 16, 1998, Available WWW: [http://www.ednmag.com/ednmag/reg/1998/021698/04df\\_01.htm](http://www.ednmag.com/ednmag/reg/1998/021698/04df_01.htm)
- [9] J. Lipman, "Multicore chips challenge system-on-chip designers," *EDN*, June 18, 1998, Available WWW: <http://www.ednmag.com/reg/1998/061898/13cs.htm>