International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-2, July 2013

Comparative Study of the 64-bit and Apple families of Microprocessors

Nitin Kumar Sharma, K.P.Yadav,B.K.Sharma

Abstract- We live in an era of computers. From the smallest embedded system to the complex servers that take care of the world economy, we need microprocessors to run them. As time passed by, applications needed more processing power and this lead to an explosive era of research on the architecture of microprocessors. As part of our project we are going to present a technical & comparative study of these smart microprocessors. It will include the different software & hardware technical aspects of such devices for instance OS, applications used, hardware study etc. In this report, we study and compare two microprocessor families that have been at the core of the world's most popular microprocessors of today – 64 bit microprocessor & Apple microprocessor.

Keywords CPU, ALU, AMD, RISC, SIMD etc.

I. INTRODUCTION

The scope of the project is to explore the different technical aspects of present world's hottest technology i.e. "Tablet PCs". It includes the technical findings, explanations for different technologies getting used in modern world tablet pcs, limitations, advantages & comparison between different technologies which are getting used for the development of such devices. In computer architecture, **64-bit computing** is the use of processors that have data path widths, integer size, and memory addresses of 64 bits (eight octets) wide. Also, 64-bit CPU and ALU architectures are those that are based on registers, address buses, or data buses of that size. From the software perspective, 64-bit computing means the use of code with 64-bit virtual memory addresses.

The term *64-bit* is a descriptor given to a generation of computers in which 64-bit processors are the norm. 64 bits is a word size that defines certain classes of computer architecture, buses, memory and CPUs, and by extension the software that runs on them. 64-bit CPUs have existed in supercomputers since the 1970s (Cray-1, 1975) and in RISC-based workstations and servers since the early 1990s, notably the DEC Alpha, Sun UltraSPARC, FujitsuSPARC64 and IB M PowerPC-AS. In 2003 they were introduced to the (previously 32-bit) mainstream personal computer arena in the form of the x86-64 and 64-bit PowerPC processor architectures and later in 2012[1] even in processors that were before mainly considered only as part of embedded systems with the introduction of the AArch64 processor architectures in ARMv8.

Manuscript published on 30 July 2013.
*****Correspondence Author(s)
Nitin Kumar Sharma, Research Scholar, Department of Computer Science & Engineering, Singhania University, Rajasthan.India.
Prof.(Dr.) K.P.Yadav, Director, Mangalmay Institute of Engineering & Technology, Greater Noida.India.
Dr.B.K.Sharma, Head, Research Centre, NITRA, Ghaziabad, (U.P)India.

A 64-bit register can store 2^{64} (over 18 quintillion) different values. Hence, a processor with 64-bit memory addresses can directly access 2^{64} bytes (=16 exabytes) of byte-addressable memory.

Without further qualification, a*64-bit computer architecture* generally has integer and addressing registers that are 64 bits wide, allowing direct support for 64-bit data types and addresses. However, a CPU might have external data buses or address buses with different sizes from the registers, even larger (the 32-bit Pentium had a 64-bit data bus, for instance). The term may also refer to the size of low-level data types, such as 64-bit floating-point numbers.

Most CPUs are designed so that the contents of a single integer register can store the address (location) of any data in the computer's virtual memory. Therefore, the total number of addresses in the virtual memory – the total amount of data the computer can keep in its working area – is determined by the width of these registers. Beginning in the 1960s with the IBM System/360 (which was an exception, in that it used the low order 24 bits of a word for addresses, resulting in a 16 MB [16 × 1024^2 bytes] address space size), then (amongst many others) the DEC VAX minicomputer in the 1970s, and then with the Intel 80386 in the mid-1980s, a *de facto*consensus developed that 32 bits was a convenient register size. A 32-bit address register meant that 2^{32} addresses, or 4 GB of RAM, could be referenced. At the time these architectures were devised, 4 GB of memory was so far beyond the typical quantities (4 MB) in installations that this was considered to be enough "headroom" for addressing. 4.29 billion addresses were considered an appropriate size to work with for another important reason: 4.29 billion integers are enough to assign unique references to most entities in applications like databases.

Some supercomputer architectures of the 1970s and 1980s used registers up to 64 bits wide. In the mid-1980s, Intel i860 development began culminating in a (too late for Windows NT) 1989 release. However, 32 bits remained the norm until the early 1990s, when the continual reductions in the cost of memory led to installations with quantities of RAM approaching 4 GB, and the use of virtual memory spaces exceeding the 4 GB ceiling became desirable for handling certain types of problems. In response, MIPS and DEC developed 64-bit microprocessor architectures, initially for high-end workstation and server machines. By the mid-1990s, HAL Computer Systems, Sun Microsystems, IBM, Silicon Graphics, and Hewlett Packard had developed 64-bit architectures for their workstation and server systems. A notable exception to this trend were mainframes from IBM, which then used 32-bit data and 31-bit address sizes; the IBM mainframes did not include 64-bit processors until 2000.

During the 1990s, several low-cost 64-bit microprocessors were used in consumer electronics and embedded applications. Notably, the Nintendo 64[4] and the PlayStation 2 had 64-bit microprocessors before their introduction in personal computers. High-end printers and network equipment, as well as industrial computers, also used 64-bit microprocessors, such as the Quantum Effect Devices R5000. 64-bit computing started to drift down to the personal computer desktop from 2003 onwards, when some models in Apple's Macintosh lines switched to PowerPC 970 processors (termed "G5" by Apple), and AMD released its first 64-bit x86-64 processor.

II. 64 BIT MICROPROCESSOR

In computer architecture, **64-bit computing** is the use of processors that have datapath widths, integer size, and memory addresses of 64 bits (eight octets) wide. Also, 64-bit CPU and ALU architectures are those that are based on registers, address buses, or data buses of that size. From the software perspective, 64-bit computing means the use of code with 64-bit virtual memory addresses.

The term *64-bit* is a descriptor given to a generation of computers in which 64-bit processors are the norm. 64 bits is a word size that defines certain classes of computer architecture, buses, memory and CPUs, and by extension the software that runs on them. 64-bit CPUs have existed in supercomputers since the 1970s (Cray-1, 1975) and in RISC-based workstations and servers since the early 1990s, notably

the DEC Alpha, Sun UltraSPARC, FujitsuSPARC64 and IB M PowerPC-AS. In 2003 they were introduced to the (previously 32-bit) mainstream personal computer arena in the form of the x86-64 and 64-bit PowerPC processor architectures and later in 2012[1] even in processors that were before mainly considered only as part of embedded systems with the introduction of the AArch64 processor architectures in ARMv8.

A 64-bit register can store 2^{64} (over 18 quintillion) different values. Hence, a processor with 64-bit memory addresses can directly access 2^{64} bytes (=16 exbibytes) of byte-addressable memory.

Without further qualification, a *64-bit computer architecture* generally has integer and addressing registers that are 64 bits wide, allowing direct support for 64-bit data types and addresses. However, a CPU might have external data buses or address buses with different sizes from the registers, even larger (the 32-bit Pentium had a 64-bit data bus, for instance). The term may also refer to the size of low-level data types, such as 64-bit floating-point numbers.

III. ARCHITECTURAL IMPLICATIONS

Processor registers are typically divided into several groups: *integer*, *floating-point*, *SIMD*, *control*, and often special registers for address arithmetic which may have various uses and names such as *address*, *index* or *base registers*. However, in modern designs, these functions are often performed by more general purpose *integer* registers. In most processors, only integer and/or address-registers can be used to address data in memory; the other types of registers cannot. The size of these registers therefore normally limits the amount of directly addressable memory, even if there are registers, such as floating-point registers, that are wider.

Most high performance 32-bit and 64-bit processors (some notable exceptions are older or embedded ARM and 32-bit MIPS CPUs) have integrated floating point hardware, which is often, but not always, based on 64-bit units of data. For example, although the x86/x87architecture has instructions capable of loading and storing 64-bit (and 32-bit) floating-point values in memory, the internal floating point data and register format is 80 bits wide, while the general-purpose registers are 32 bits wide. In contrast, the 64-bit Alpha family uses a 64-bit floating-point data and register format (as well as 64-bit integer registers).

IV. FEATURES

A. True 64-bit microprocessor
   a. 64-bit integer operations
   b. 64-bit floating-point operations
   c. 64-bit registers
   d. 64-bit virtual address space

B. High-performance microprocessor
   a. 150 peak MIPS at 150MHz
   b. 50peak MFLOP/s at 150MHz
   c. 96 SPECint92 at 150Mz
   d. Two-way set associative caches

• High level of integration
   - 64-bit integer CPU
   - 64-bit floating-point unit
   - 16KB instruction cache; 16KB data cache
   - Flexible MMU with large TLB

• Low-power operation
   - 3.3V or 5V power supply options
   - 20mW/MHZ typical internal power dissipation (2.0W @ 100MHz, 3.3V)
   - Standby mode reduces internal power to 90mA @ 5V and 60mA @ 3.3V (450mW @ 5V and 400mW @ 3.3V).

• Standard operating system support includes:
   - Microsoft Windows NT
   - UNISOFT Unix System V.4

• Fully software compatible with R4000 RISC Processor Family
• Available in R4000PC/R4000PC pin-compatible 179-pin PGA or 208-pin MQUAD
• 50MHz, 67MHz, 75MHz input frequencies with mode bit dependent output clock frequencies
   - On-chip clock doubler for 150MHz pipeline
• 64GB physical address space
• Processor family for a wide variety of applications
   - Desktop workstations and PCs
   - Deskside or departmental servers

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

- High-performance embedded applications (e.g. Color printers, multi-media and internetworking.)
- Notebooks

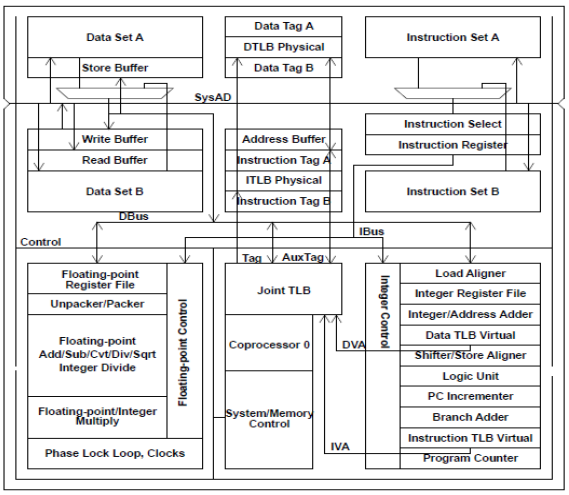

Figure 1: 64 Bit Microprocessor Architecture

V. CURRENT 64-BIT MICROPROCESSOR ARCHITECTURES

64-bit microprocessor architectures for which processors are currently being manufactured (as of January 2011) include:

- The 64-bit extension created by AMD to Intel's x86 architecture (later licensed by Intel); commonly known as "x86-64", "AMD64", or "x64":
- AMD's AMD64 extensions (used in Athlon 64, Opteron, Sempron, Turion 64, Phenom, Athlon II and Phenom II processors)
- Intel's Intel 64 extensions (used in newer Celeron, Pentium, and Xeon processors, in Intel Core 2/i3/i5/i7 processors, and in some Atom processors)
- VIA Technologies' 64-bit extensions, used in the VIA Nano processors
- The 64-bit version of the Power Architecture:
- IBM's POWER6 and POWER7 processors
- IBM's PowerPC 970 processor
- The Cell Broadband Engine used in the PlayStation 3, designed by IBM, Toshiba and Sony, combines a 64-bit Power architecture processor with seven or eight Synergistic Processing Elements.
- IBM's "Xenon" processor used in the Microsoft Xbox 360 comprises three 64-bit PowerPC cores.
- SPARC V9 architectureSun's UltraSPARC processors
- Fujitsu's SPARC64 processors
- IBM's z/Architecture, a 64-bit version of the ESA/390 architecture, used in IBM's eServer zSeries and System z mainframes
- Intel's IA-64 architecture(used in Itanium processors)
- MIPS Technologies' MIPS64 architecture
- ARM Holdings' AArch64 architecture

Most 64-bit processor architectures that are derived from 32-bit processor architectures can execute code for the 32-bit version of the architecture natively without any performance penalty. This kind of support is commonly called *bi-arch support* or more generally *multi-arch support*.

VI. LIMITATIONS OF PRACTICAL PROCESSORS

In principle, a 64-bit microprocessor can address 16 exabytes of memory. In practice, it is less than that. For example, the AMD64 architecture as of 2011 allows 52 bits for physical memory and 48 bits for virtual memory. These limits allow memory sizes of 4 PB (4×1024^5 bytes) and 256 TB (256×1024^4 bytes), respectively. A PC cannot contain 4 petabytes of memory (due to the physical size of the memory chips, if nothing else) but AMD envisioned large servers, shared memory clusters, and other uses of physical address space that might approach this in the foreseeable future, and the 52-bit physical address provides ample room for expansion while not incurring the cost of implementing 64-bit physical addresses. Similarly, the 48-bit virtual address space was designed to provide more than 65,000 times the 32-bit limit of 4 GB (4×1024^3 bytes), allowing room for later expansion without incurring the overhead of translating full 64-bit addresses.

A. 64- bit application

32-bit vs 64-bit

A change from a 32-bit to a 64-bit architecture is a fundamental alteration, as most operating systems must be extensively modified to take advantage of the new architecture, because that software has to manage the actual memory addressing hardware. Other software must also be ported to use the new capabilities; older 32-bit software may be supported through either a *hardware compatibility mode* in which the new processors support the older 32-bit version of the instruction set as well as the 64-bit version, through software emulation, or by the actual implementation of a 32-bit processor core within the 64-bit processor, as with the Itanium processors from Intel, which include an IA-32 processor core to run 32-bit x86 applications. The operating systems for those 64-bit architectures generally support both 32-bit and 64-bit applications.

One significant exception to this is the AS/400, whose software runs on a virtual Instruction Set Architecture (ISA) called TIMI (Technology Independent Machine Interface), which is translated to native machine code by low-level software before being executed. The translation software is all that has to be rewritten to move the entire OS and all software to a new platform, such as when IBM transitioned their line from the older 32/48-bit "IMPI" instruction set to the 64-bit PowerPC-AS instruction set, codenamed "Amazon" (the IMPI instruction set was quite different from the 32-bit PowerPC instruction set, so this was an even bigger transition than from a 32-bit version of an instruction set to a 64-bit version of the same instruction set).

On 64-bit hardware with x86-64 architecture (AMD64), most 32-bit operating systems and applications can run without compatibility issues. While the larger address space of 64-bit architectures makes working with large data sets in applications such as digital video, scientific computing, and large databases easier, there has been considerable debate on whether they or their 32-bit compatibility modes will be faster than comparably priced 32-bit systems for other tasks.

A compiled Java program can run on a 32- or 64-bit Java virtual machine without modification. The lengths and precision of all the built-in types are specified by the standard and are not dependent on the underlying architecture. Java programs that run on a 64-bit Java virtual machine have access to a larger address space.

Speed is not the only factor to consider in a comparison of 32-bit and 64-bit processors. Applications such as multi-tasking, stress testing, and clustering for high-performance

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

computing (HPC)—may be more suited to a 64-bit architecture when deployed appropriately. 64-bit clusters have been widely deployed in large organizations, such as IBM, HP, and Microsoft, for this reason.

Summary:

- A 64-bit processor performs best with 64-bit software.
- A 64-bit processor has backward compatibility and will handle most 32-bit software.
- A 32-bit processor is not compatible with 64-bit software.

Pros and Cons

A common misconception is that 64-bit architectures are no better than 32-bit architectures unless the computer has more than 4 GB of random access memory. This is not entirely true:

- Some operating systems and certain hardware configurations limit the physical memory space to 3 GB on IA-32 systems, due to much of the 3–4 GB region being reserved for hardware addressing; see 3 GB barrier; 64-bit architectures can address far more than 4 GB. However, IA-
- 32 processors from the Pentium II onwards allow for a 36-bit *physical* memory address space, using Physical Address Extension (PAE), which gives a 64 GB physical address range, of which up to 62 GB may be used by main memory; operating systems that support PAE may not be limited to 4GB of physical memory, even on IA-32 processors. However, drivers and other kernel mode software, particularly older versions, may not be compatible with PAE.
- Some operating systems reserve portions of process address space for OS use, effectively reducing the total address space available for mapping memory for user programs. For instance, 32-bit Windows reserves 1 or 2 GB (depending on the settings) of the total address space for the kernel, which leaves only 3 or 2 GB (respectively) of the address space available for user mode. This limit is much higher on 64-bit operating systems.
- Memory-mapped files are becoming more difficult to implement in 32-bit architectures as files of over 4 GB become more common; such large files cannot be memory-mapped easily to 32-bit architectures—only part of the file can be mapped into the address space at a time, and to access such a file by memory mapping, the parts mapped must be swapped into and out of the address space as needed. This is a problem, as memory mapping, if properly implemented by the OS, is one of the most efficient disk-to-memory methods.
- Some 64-bit programs, such as encoders, decoders and encryption software, can benefit greatly from 64-bit registers, while the performance of other programs, such as 3D graphics-oriented ones, remains unaffected when switching from a 32-bit to a 64-bit environment.
- Some 64-bit architectures, such as x86-64, support more general-purpose registers than their 32-bit counterparts (although this is not due specifically to the word length). This leads to a significant speed increase for tight loops since the processor does not have to fetch data from the cache or main memory if the data can fit in the available registers.

Example in C:

```
int a, b, c, d, e;
for (a=0; a<100; a++)
{
b = a;
        c = b;
       d = c;
        e = d;
       }
```

If a processor only has the ability to keep two or three values or variables in registers it would need to move some values between memory and registers to be able to process variables d and e as well; this is a process that takes many CPU cycles. A processor that is capable of holding all values and variables in registers can loop through them without needing to move data between registers and memory for each iteration. This behavior can easily be compared with virtual memory, although any effects are contingent upon the compiler.

The main disadvantage of 64-bit architectures is that, relative to 32-bit architectures, the same data occupies more space in memory (due to longer pointers and possibly other types, and alignment padding). This increases the memory requirements of a given process and can have implications for efficient processor cache utilization. Maintaining a partial 32-bit model is one way to handle this, and is in general reasonably effective. For example, the z/OS operating system takes this approach, requiring program code to reside in 31-bit address spaces (the high order bit is not used in address calculation on the underlying hardware platform) while data objects can optionally reside in 64-bit regions.

As of June 2011, most proprietary x86 software is compiled into 32-bit code, with less being also compiled into 64-bit code (although the trend is rapidly equalizing[*citation needed*]), so most of that software does not take advantage of the larger 64-bit address space or wider 64-bit registers and data paths on x64 processors, or the additional general-purpose registers. However, users of most RISC platforms, and users of free or open source operating systems (where the source code is available for recompiling with a 64-bit compiler) have been able to use exclusive 64-bit computing environments for years. Not all such applications require a large address space or manipulate 64-bit data items, so these applications do not benefit from these features. The main advantage of 64-bit versions of such applications is the ability to access more registers in the x86-64 architecture.

VII. APPLE MICROPROCESSOR

Figure 2: Apple microprocessor architecture

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Apple is using its own A4 microprocessor in a next-generation iPhone, according to an analysis by Vietnamese bloggers who obtained a prototype phone.

The A4, which was evidently designed at Apple by the engineers it acquired from PA Semi and Intrinsity, is a 1-gigahertz ARM-based chip that serves as the brain of the iPad. So it's not surprising that Apple would repurpose the chip for future iPhones.

The A4 combines an ARM processor with graphics on a single chip. It uses graphics licensed from Imagination Technologies, a U.K. company that created the PowerVR graphics architecture. Apple owns 9 percent of Imagination, while Intel owns 16 percent.

Apple's wider use of the A4 in multiple product lines will give it the unit sales volume necessary to reach economies of scale for its chip. The more devices it sells, the more Samsung can manufacture, the faster the companies can reduce the costs of the chips, and the more profit falls into Apple's pockets.

It's a smart strategy, as long as Apple can keep up with or surpass the broad merchant semiconductor makers with performance. That means Apple will have to continuously invest in new chip designs. For other companies, that's not an option and its much easier to buy chips from the likes of Broadcom, Intel, Marvell, Nvidia, Texas Instruments and Samsung.

But Apple has deep enough pockets to pull this off. It also has a vested interest in differentiation. Apple has never wanted to have exactly what other companies have. It wants faster chips than its rivals have access to so that it can set its own products apart from the pack. Other companies don't seem to mind being part of the pack if it means their costs are lower. Apple also wants to capture more margin for itself. This way, it doesn't have to pay another chip design company, which might have a chip fabricated for $10 and then sell it for $30.

But as we pointed out in a recent story on vertical vs. horizontal strategy, the thinking has been changing in Silicon Valley. Google acquired its own chip talent with its acquisition of Agnilux. It thus joins the likes of IBM, Oracle, and Microsoft in having chip design capability in addition to its other software businesses.

I asked Paul Otellini, chief executive of Intel, about why so many tech companies were moving to vertical strategies where they owned their own chips. He said, "Apple envy." Then he added, "They should remember that in our industry, open wins. You have to be open."

I suspect that Otellini is right, though we can argue about exactly where and how companies need to be open and where they need to own their own technology. Apple has decided that it should own the designs for at least two of its major product lines. We'll see if it ever moves off Intel chips for its Mac computers. That may sound crazy, but Apple has already pulled off a switch to its own chip that has surprised everyone.

Apple to Use Intel Microprocessors Beginning in 2006 WWDC 2005, SAN FRANCISCO—June 6, 2005—At its Worldwide Developer Conference today, Apple® announced plans to deliver models of its Macintosh® computers using Intel® microprocessors by this time next year, and to transition all of its Macs to using Intel microprocessors by the end of 2007. Apple previewed a version of its critically acclaimed operating system, Mac OS® X Tiger, running on an Intel-based Mac® to the over 3,800 developers attending CEO Steve Jobs' keynote address. Apple also announced the availability of a Developer Transition Kit, consisting of an Intel-based Mac development system along with preview versions of Apple's software, which will allow developers to prepare versions of their applications which will run on both PowerPC and Intel-based Macs.

"Our goal is to provide our customers with the best personal computers in the world, and looking ahead Intel has the strongest processor roadmap by far," said Steve Jobs, Apple's CEO. "It's been ten years since our transition to the PowerPC, and we think Intel's technology will help us create the best personal computers for the next ten years."

"We are thrilled to have the world's most innovative personal computer company as a customer," said Paul Otellini, president and CEO of Intel. "Apple helped found the PC industry and throughout the years has been known for fresh ideas and new approaches. We look forward to providing advanced chip technologies, and to collaborating on new initiatives, to help Apple continue to deliver innovative products for years to come."

"We plan to create future versions of Microsoft Office for the Mac that support both PowerPC and Intel processors," said Roz Ho, general manager of Microsoft's Macintosh Business Unit. "We have a strong relationship with Apple and will work closely with them to continue our long tradition of making great applications for a great platform."

"We think this is a really smart move on Apple's part and plan to create future versions of our Creative Suite for Macintosh that support both PowerPC and Intel processors," said Bruce Chizen, CEO of Adobe.

The Developer Transition Kit is available starting today for $999 to all Apple Developer Connection Select and Premier members. Further information for Apple Developer Connection members is available at developer.apple.com. Intel plans to provide industry leading development tools support for Apple later this year, including the Intel C/C++ Compiler for Apple, Intel Fortran Compiler for Apple, Intel Math Kernel Libraries for Apple and Intel Integrated Performance Primitives for Apple.

VIII. APPLE BETS ITS CHIPS ON IN-HOUSE MICROPROCESSOR DESIGN

The microprocessors that will go into future Apple products -- especially the handheld variety -- will need to be smaller, cheaper and more powerful while using less power. Those chips will be the result of Apple designer brainpower, not third-party suppliers, according to a recent report.

Steve Jobs' company is in the middle of a hiring spree, *The Wall Street Journal* reported Thursday, recently posting openings for microprocessor designers and putting new high-level hires from Advanced Micro Devices to work on next-generation chips.

The *Journal* calls it a big shift in strategy that began a year ago, when Apple bought low-power processor design company PA Semi.

The company will likely have to outsource the actual manufacturing of processors to a foundry, but bringing chip design within its Cupertino campus not only might prove more efficient, but also could help protect vital intellectual property.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Yet it will also mean that Apple will have to spend enough money to retain those qualified chip designers and to maintain a product pipeline that can squeeze enough profit out of the new chips.

IX. THE NEED FOR SOCS

At the core of Apple's processor issue is the need to put more features and computing power onto a single chip. Standard desktops and notebooks have enough elbow room on their motherboards for all the processors that work with a central processing unit -- graphics, memory, etc. However, an iPhone or iPod touch has limited space available; hence the need for more robust systems-on-a-chip (SOCs) that contain all those elements on a single processor. However SOCs are known to suck up battery juice.

"The more specific your SOC is to the product design, the more optimized it is, and the better fit you have" Tom Halfhill, senior analyst with In-Stat's Microprocessor Report, told MacNewsWorld. "If you take all the features you want in a portable product -- smartphone, laptop, netbook, whatever -- and you integrate more of those functions in one chip, you can now make a very optimized product that burns less power, can be smaller [and] less costly to make -- and you can improve your profit margins."

Power management will be key, said 451 Group Research Director Chris Hazelton, since some of Apple's smartphone competitors boast longer battery life between recharges.

"The iPhone is such a great device that you have people using it for longer periods of time during the day than they would use any other smartphones," Hazelton told MacNewsWorld. "You have video, high-resolution touchscreens, excellent browsing capabilities. You have to have battery management."

Hazelton is expecting near-term iPhone variations that would include a hard keyboard to appease what he calls a "huge pool of users" unwilling to adapt to touchscreens. "What that will do is drive people to email and message more on the iPhone than they currently do. I imagine that [better SOCs] can provide some space-saving. They're not going to reduce the size of the screen -- it's the differentiator of the iPhone. What is lacking is a keyboard."

X. THE CHIP DESIGN PROCESS CALENDAR

iPhone variations are likely on the way, agreed Halfhill, as well as a netbook -- "it's just too popular a category to ignore" -- but that could just be the start of an incredibly demanding SOC product calendar.

"To design an SOC and get it out the door, you're talking a minimum of 12 to 18 months. That's assuming everything goes perfectly[11]. There's a long lag between hiring [designers] and getting a chip out the door," he said. "It also means in order to have a stream of products that are always improving, you have to have multiple SOC products in the pipeline -- almost a new SOC coming out every year. To do that, you need multiple projects on a staggered schedule, and you have to hire enough designers and engineers."

SOCs are expensive to develop, Halfhill said, and that will require plenty of refreshed iPhones, iPods, laptops and (potentially) netbooks to ensure that Apple gets its money's worth out of each new processor. Also, it's unknown whether Apple has plans to adopt ARM architecture for its new in-house designed SOCs -- the iPhone currently runs this processor architecture, which is known for better power management -- or if it will try to adopt the x86 architecture that is now found in Apple desktops and notebooks.

XI. APPLE 68000-8

8 MHz 64- pin side- brazed ceramic DIP This 68000 microprocessor was not manufactured by Apple, but it was put into separate "Apple" category due to very unusual Apple copyright on the CPU. It's likely the CPU was manufactured by Hitachi as their 68000 processors have similar packaging and chip markings.

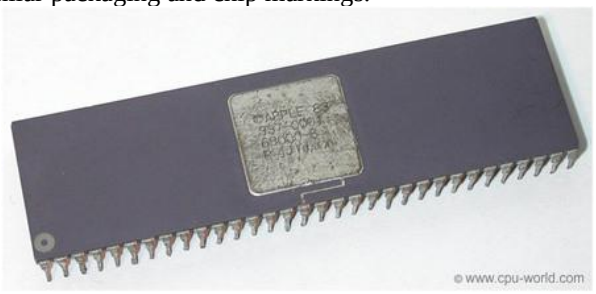

Figure 3: Hitachi Microprocessor

XII. CONCLUSION

The two micro-processor families described here really compete for the top spot in modern high performance microprocessors. The latest 64 bit microprocessor and the Apple microprocessor seem to be very close in terms of performance.

XIII. ACKNOWLEDGEMENT

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings. Especially, please allow me to dedicate my acknowledgment of gratitude toward the following significant advisors and contributors: First and foremost, I would like to thank Dr. B. K. Sharma, Principal Scientific Officer, NITRA (Govt. of India) for his most support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper. Second, I would like to thank Prof. Dr. K. P. Yadav to read my thesis and to provide valuable advices, as well as all the other professors who have taught me about microprocessors.

REFERENCES

1. Hadi. “Tablet Ward: Apple iPad vs. BlackBerry Playbook vs. Samsung Galaxy Tab” *September 28, 2010*
2. Josh Morse “Apple iPad vs. Tablet PC: A comparison” January 28th,2010
3. Apple Developer “Start Developing iPad Apps January 2010
4. Samsung Galaxy Tab Developer Forum "Developing applications for the Samsung Galaxy Tab
5. Techinsights “Apple-iPad- Redefining the table market”
6. Android Operating system a complete perspective
7. Nick Van Elslander, Mario Suppan, Thomas De Roy, Tuan Vu, iPhone Research Paper, MAD intensive programming 2009.
8. Windows 7 Architecture – Wikipedia,
9. iOS Developer Community , iOS – Features
10. Develop for iOS
11. Samsung Galaxy Tab vs. the iPad Compare for yourself
12. HTC Android Tablet appearing in CES 2010
13. HTC rumored to be readying an Android Tablet for Q1 2011

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Nitin Kumar Sharma, Research Scholar, Department of Computer Science & Engineering, Singhania University, Rajasthan.

Prof.(Dr.) K.P.Yadav, Director, Mangalmay Institute of Engineering & Technology, Greater Noida.

Dr.B.K.Sharma, Head, Research Centre, NITRA, Ghaziabad, (U.P)

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)