RUNNING HEAD: Mobile Computing Innovations

Mobile Computing Innovations

Aidan Hopper

ID: 42236214

CS 312

Dr. Donald Davendra

I pledge that this submission is solely my work, and that I have neither given,

nor received help from anyone.

**Abstract**

Mobile computing is an ever-advancing field of research that has had a multitude of advancements in recent history. One of the first advancements to happen to mobile computing is ARM architecture. ARM architecture differs from x86-64 architecture in that it only includes fundamental instructions that are needed called RISC architecture. This allows ARM processors to use fewer transistors for storing instructions reducing power consumption. ARM processors further increase their efficiency through integration in an SoC. This allows ARM processors to be further optimized for all the components included in an SoC. The other innovation that has greatly benefited mobile computing is the upgrade from 32-bit ARM architecture to 64-bit ARM architecture. This upgrade allows ARM processors to utilize more than 4GB of RAM per application and increases efficiency because of the increased register size. These innovations have specifically benefitted the mobile form factor due to their power-efficient, small size nature.

Table of Contents

[Abstract 2](#1%19Abstract%19C)

[Introduction 4](#1%19Introduction%19C)

[32-bit Architecture Limitations 4](#1%1932-bit_Architecture_Limitations%19C)

[ARM Processors 6](#1%19ARM_Processors%19C)

[ARM Power Efficiency 6](#1%19ARM_Power_Efficiency%19C)

[SoC Architecture 7](#1%19SoC_Architecture%19C)

[64-bit ARM 8](#1%1964-bit_ARM%19C)

[Conclusion 9](#1%19Conclusion%19C)

[References 10](#1%19References%19C)

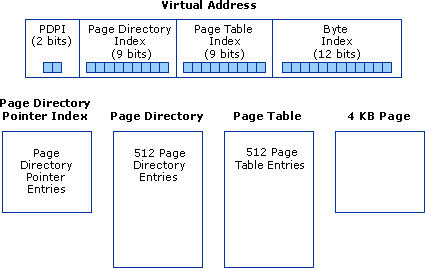
**Introduction**

Mobile computing has seen many advancements throughout recent history. From the release of the first iPhone to now, there has been a multitude of improvements to processing power, battery life, and new features such as fingerprint sensing. These improvements are enabled by innovations in mobile computing and broader computer processing. This paper will analyze mobile computing's most influential advancements in processing power, power consumption, and techniques that were previously used to circumvent architectural issues.

**32-bit Architecture Limitations**

In computer processors, there is a type of memory used when directly manipulating data. This is referred to as a register. There are multiple registers in a computer processor with set sizes with the most common ranging from 8-bit, 16-bit, 32-bit, and 64-bit. Registers can store instructions, memory addresses, or data types that are being worked with. When referencing 32-bit in terms of computer processors/architecture, it is referencing the size in bits of the processor's registers. 32-bit registers were the most common size before the mass upgrade to 64-bit registers in computers. Processors have multiple registers, each with an assigned or general-purpose. The register where memory addresses are stored is one of 32-bit architecture's major limitations. The memory address register is used to point to a location in Random Access Memory (RAM) which facilitates access to that point in RAM. A 32-bit register means that there are 232 unique memory addresses that can be accessed in RAM, which is equivalent to 4GiB of addresses. This means that the size of RAM had a hardware limitation of 4GB on 32-bit architecture. With the onset of 64-bit architecture, the limitation of 4GB had been greatly exceeded with the new theoretical limit on RAM size being put at 264 bytes, exponentially larger than on 32-bit architecture.

There is a method to increase the total amount of RAM a computer can use on 32-bit architecture through Physical Address Extension (PAE). PAE operates through virtualizing memory that can be translated into a physical address in memory. PAE uses the virtual address to add an extra layer of translation called the Page Directory Pointer Table. This extra layer of translation allows a processor to utilize more than 4GB of RAM. The limitation of PAE is that individual applications are still limited to 4GB of RAM (*How PAE X86 Works: System Reliability*, 2009).

Figure 1: Virtual Address Diagram

Source: How PAE X86 Works: System Reliability, 2009

Figure 1 showcases the design of the virtual address which was used to partially circumvent the 4GB RAM limit on 32-bit machines. In the diagram, the Page Directory Pointer points to a Page Directory, which then points to a Page Table, which then points to the index of the desired byte. Using the extra translation layer that the Page Directory Ponter Table provides enables individual applications to address in 32-bits that are mapped to different areas in RAM per application.

**ARM Processors**

Computers have two main types of instruction sets that can be used depending on a specific use case. The two types are the x86 architecture and ARM architecture. Each of the two processors has its costs and benefits which make them stronger in some sectors than others. ARM processors are Restricted Instruction Set Computers (RISC), which are designed to work with simple instructions where all instructions are the same size in memory. X86 processors are Complex Instruction Set Computers (CISC) where the instructions are a variety of sizes in memory and complexity. An example of a complex instruction in CISC is the MULT instruction, which can be replicated in RISC with the use of three separate instructions (*RISC vs. CISC*, n.d.). CISC processors include many instructions in their architecture that go widely unused. Since RISC instructions are much simpler than CISC instructions, only fundamentally necessary instructions are included in the architecture. The simplicity of RISC instructions causes programs built on top of RISC to be larger than an equivalent CISC program. This is because multiple RISC instructions are needed to perform the same function as one CISC instruction (Aroca & Gonçalves, 2012).

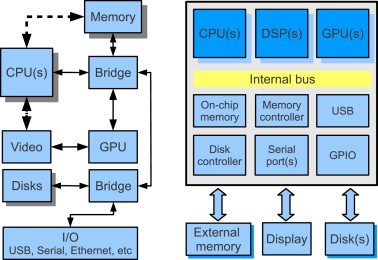
**ARM Power Efficiency**

One advantage that ARM processors have over x86 is power efficiency. Multiple design features allow this. The first and most important one is ARM architecture's integration with its internal components. To illustrate this it is best to understand x86 bridges. An x86 processor is connected to outside chips like the GPU, I/O, etc. through bridges. This is in contrast to ARM processors which are deeply integrated under one chip called an SoC (System on a Chip). This is one of the x86 architectures trade-offs with the ARM architecture, where x86 gains in modularity it loses in integrability. This difference among others allows ARM processors to be optimized for the components included in the chip, increasing power efficiency over its x86 counterpart (Aroca & Gonçalves, 2012). This is why ARM Processors are heavily used in mobile phones and laptops.

Another reason ARM architecture is more efficient is its use of the RISC instruction set. This is because RISC includes fewer instructions, that are more efficient because they are designed to ideally execute in a single clock cycle. RISC processors require fewer transistors because of the lack of legacy instructions, which by logical extension means lesser power requirements (*RISC vs. CISC*, n.d.).

**SoC** **Architecture**

SoC architectures are the backbone of mobile devices. As said previously, they leverage their ARM architecture, along with complete system integration, to produce high-performance, low-power designs for mobile computing. An SoC will typically integrate a GPU, RAM, CPU, and I/O under one chip. This requires less space than a design with separate, GPU, RAM, CPU, and I/O chips, and is more reliable due to full system integration (*Chipset and System-on-a-Chip (SoC) Reference for Intel® NUC. . .*, n.d.).

Figure 2: Comparison of x86 system (left) and SoC (right)

Source: Aroca & Gonçalves, 2012

Figure 2 showcases the differences between a typical SoC and an x86 system. With the x86 system, the individual parts are connected through multiple bridges, while the SoC design is integrated under one internal bridge. This integration allows for data transfer between each component in the system to consume less power than the x86 system which has to transfer data between many bridges to communicate with other components. This integration allows SoCs to be made smaller than an x86 system.

**64-bit ARM**

One innovation that has occurred in recent years is 64-bit ARM processors (“ARM Targets Server Designs with ‘first’ 64- Bit Processor.,” 2011). In 2011 ARM released details of its new ARMv8 architecture, which included a set of 64-bit instructions, along with a set of 32-bit instructions used previously. At the most basic level, the 64-bit architecture allows for the processing of larger chunks of data, due to the doubling in register size from 32-bit architecture. This combined with ARM architecture's high efficiency compared to x86-64 architecture and the exponential increase to the RAM size limit creates the perfect environment for advancements in mobile computing. This is why all modern mobile processors are based on ARM architecture (PratapSingh & Kumar Jain, 2014).

**Conclusion**

Mobile computing has gone through many innovations. One notable innovation is the usage of ARM architecture. ARM processors use RISC, a more simple, optimized instruction set to achieve high-efficiency computing. This allows RISC processors to use be less power-hungry than a CISC processor, making RISC architecture the perfect choice for mobile devices. Another notable innovation that occurred in the mobile phone market, and broadly computing in general, was the upgrade from 32-bit to 64-bit architecture, referring to the size of the registers. This allowed for more efficient processing of larger data types, such as longs. This also allowed computer programs to use more than 4GB of RAM, which was caused to a 32-bit hardware limitation. When ARM released a 64-bit instruction set, it allowed mobile processors to take advantage of the benefits of 64-bit computing.

# References

ARM targets server designs with “first” 64- bit processor. (2011). *Electronics Weekly*, *2486*(4).

Aroca, R. V., & Gonçalves, L. M. G. (2012). Towards green data centers: A comparison of x86 and ARM architectures power efficiency. *Journal of Parallel and Distributed Computing*, *72*(12), 1770–1780. https://doi.org/10.1016/j.jpdc.2012.08.005

*Chipset and System-on-a-Chip (SoC) Reference for Intel® NUC. . .* (n.d.). Intel. https://www.intel.com/content/www/us/en/support/articles/000056878/intel-nuc/intel-nuc-laptop-kits.html

*How PAE X86 Works: System Reliability*. (2009, October 8). Microsoft Docs. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc736309(v=ws.10)?redirectedfrom=MSDN

PratapSingh, M., & Kumar Jain, M. (2014). Evolution of Processor Architecture in Mobile Phones. *International Journal of Computer Applications*, *90*(4), 34–39. https://doi.org/10.5120/15564-4339

*RISC vs. CISC*. (n.d.). Stanford. https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/