**Gameboy Advance and Nintendo DS Comparison**

**Coleman Gibson and Katrina Kerrick**

**ECE 332**

**Introduction**

Nintendo started the console craze with the Game Boy, launched in 1989. After its success, they started building improvements. In 2001, Nintendo released the Game Boy Advance (GBA), and shortly afterward in 2004, it released the Nintendo Dual Screen (DS). Both of these consoles are rather old now, but they helped pave the way for handheld consoles of all kinds.

Both of these consoles feature cutting edge hardware for their time. The GBA was the first handheld to feature two processors, each with a separate task, and the DS took a gamble with the use of two screens instead of one. Both use their specializations to the fullest, with a huge amount of games written to take advantage of these. This paper focuses on the CPU design and how that affected gameplay, and the improvements made in the DS.

Nintendo is very private with their hardware specifications, and still has not released any specialized hardware information about its newer consoles. Even gleaning this information personally is difficult, because Nintendo uses triangular screws to keep nosy consumers from opening the cases of their consoles. However, because of the age of the GBA and the DS, they are public online down to their datapaths. In fact, emulators of these devices are both common and free online, which makes a case study of their CPUs much easier.

**Gameboy Advance**

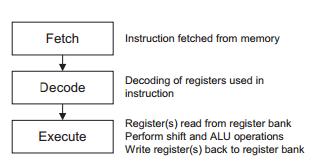
**Background**

In the first half of 2001, Nintendo continued its series of handheld video game consoles with the arrival of the Gameboy Advance. This device turned out to be extremely popular, selling over 81 million units worldwide, and having 576 games released for the platform in Europe and America. Intended to be a successor to the Gameboy Color, it is backwards compatible with all Gameboy Color and standard Gameboy games. In order to play past games, the Advance has two processors--an ARM7 processor for the newest games, and a Z80 processor identical to the Gameboy Color’s. The GBA selects the appropriate processor at bootup, based on which type of game pack was inserted.

**CPU**

The Gameboy Advance uses a heavily customized Zilog Z80 processor for backwards compatibility. This processor is extremely similar to an Intel 8080, except it does not have any of the exchange instructions which were used to swap floating point numbers from one sixteen bit register to another. [1] There are also a handful of other minor instructions that were added to the processor.

The other processor in the GBA is the ARM7-TDMI, which is a standard ARM7 processor used in smaller electronics. ARM processors are specifically designed to take as little power as possible, and this allows the GBA to run on two AA batteries and still get about fifteen hours of battery life.

The ARM7-TDMI processor supports two instructions types--ARM4, a verbose 32-bit instruction set, and THUMB, which is a 16-bit, simplified version of ARM. Though THUMB has less capabilities than ARM it can be loaded faster from ROM because the ROM bus width in the GBA is only 16 bits. THUMB instructions can be decoded in constant, negatable time because they use 32-bit registers. This processor uses only a three stage pipeline--Fetch, Decode, and Execute. Writeback occurs during the execute stage. [2] This makes flushing the pipeline relatively inexpensive.

The THUMB instruction set is a subset of ARM instructions, simplified to fit in a 16-bit bus. To accommodate that smaller bus, there are fewer opcodes, only seven registers available, no conditional bits, and only two registers can be specified in each instruction. [3] For example, in ARM the instruction “x = y + z << 4” that also doesn’t set the comparison registers would be simplified to the format “x = x + y”, where shifts aren’t available and the comparison registers are always set. Comparison bits are carry, overflow, and sign flags set by the ALU, and are used in branches. While the simplified instructions are 16-bit, they directly translate to 32-bit instructions, which is why they have no extra latency.

ARM, on the other hand, is a much more robust assembly language. It makes full use of 32-bit instructions by including multifunctional instructions, like shifting inside an add. This is extremely similar to MIPS’s shamt section of the instruction code. In addition, there’s a bit switch for choosing whether or not to update condition codes. With these two instruction sets the ARM7-TDMI is an extremely useful processor for something as small as a GBA. When writing code, the programmer must give the compiler a signal to show which instruction set to use. This allows the programmer to use both types in a single program. The instruction set can be set using a special instruction, or it can be specified within a branch instruction which assembly language is being used on the other side.

**Memory**

The Gameboy Advance has 32KB of fast, on-chip memory, and 256KB of slower, on-board memory. This memory is used for storing instructions and data for the game, as the rest of memory is either related to the other processor (VRAM), or is much slower. There is a 16KB section of memory called BIOS ROM, for instructions and the basic input/output system. BIOS ROM uses only a 16-bit data bus, but because of the bus size and it not being on chip it’s much slower than standard memory for 32-bit instructions. In addition, there are are two 1KB memories for color palettes and objects to be displayed on screen. Though it’s often confusing for programmers to keep track of these different kinds of memory, the GBA is much more efficient when all the memory works together.

**Nintendo DS**

**Background**

The Nintendo DS is a handheld gaming device released in November of 2004. While not intended to replace the GBA it featured better hardware and was backwards compatible with GBA gaming cartridges. It could not, however, run cartridges from the Gameboy Color device which the GBA was capable of. Nintendo subsequently released the Nintendo DS Lite in 2006 and the Nintendo DSi in 2008. All together, these consoles sold a total of 154 million units. The DS was intended to help the company’s image, which had been moving towards a fairly conservative one as the Gameboy Color, Gameboy Advance, Gameboy Advance SP, and others were all very similar plays on the same idea for a handheld device. With two screens and touchscreen capabilities, the DS took a step forward without leaving behind the feel of the previous devices. 

**CPU**

The DS contains two processors. The main processor is an ARM946E-S and the second is a ARM7-TDMI. The ARM9 was included because Nintendo did not believe the ARM7 processor would be able to keep up with advancing computation requirements. Interestingly, the ARM9 processor runs at 67.028 MHz and the ARM7 runs at 33.514 MHz when running Nintendo DS games and ~16 MHz when running GBA games. A fairly obvious observation is that the ARM7 processor runs at exactly half of the speed of the ARM9 processor when both are operating. This is because of communication timing. If the slower clock time did not divide the faster one they would quickly become out of sync. Messages sent back and forth could then arrive either too early or too late and be overwritten before they were used or overwrite something important themselves. Because of its lesser computing power the ARM7 processor is not intended to assist in serious computations, but was included to carry out simple tasks like taking input and controlling the wifi unit. The ARM7 can set values in shared memory or use the communication described in the next section to inform the ARM9 when buttons are pressed, what data is being received by the wifi unit, and other simple functions. The ARM9 is actually incapable of accessing most of the hardware directly, giving it the hardware enforced purpose of running game logic [5]. Another reason that Nintendo included the ARM7 processor was to make backwards compatibility with the GBA simple. Both the DS and the GBA contain the same ARM7-TDMI chip, the only difference being the clock speed. However, when the DS is being run in GBA mode the clock speed in slowed to match the GBA’s 16 MHz. When only one of the two screens is being used, the hardware in use is extremely similar to the GBA. Running games on the second processor also allows for lower power requirements when GBA games are being played. The ARM9 core can be shut off completely leaving only the more energy efficient ARM7-TDMI. 

**Memory**

The usage of memory in the DS is less unusual than its processor setup. The ARM9 features a 4KB data cache and an 8KB instruction cache. Both are four way set associative [5]. The processor also features two sections of something called tightly coupled memory for both instructions and data. This is similar to a cache in that it is fast and rarely requires delays to access, however it is controlled by the programmer and is physically separate from the other caches. It is also located outside of the processor. Locating the memory outside of the processor was a decision made by ARM to allow for some flexibility in the implementation of the tightly coupled memory [6]. The ARM7, on the other hand, has no cache memory at all [5]. Instead it was connected to a fast 64KB memory. Most of the procedures the ARM7 uses are I/O based, which is not a memory intensive task. Because of this extremely fast memory access is not a priority.

**CPU Communication**

The problem of inter-CPU communication is solved using a shared queue implemented with registers. Using this queue, either CPU can send instructions or data back and forth although it must be emptied for the direction to switch. The processors also share a chunk of memory in the main memory bank designed to allow for safe asynchronous reads and writes. The rest of the main memory can still be used by both processors, but safety is not guaranteed. Great care must be taken when using memory not in this small shared block. Because the ARM9 has a data cache and the ARM7 does not, data inconsistencies are easy to create. Modifying a value in the ARM9’s cache renders the value in main memory incorrect, but the ARM7 still has full access to it. This can happen the other way as well. The ARM9 will not see updates made to values in main memory if it holds those values in its cache. While there are a number of solutions to this problem, these were not implemented in the Nintendo DS. It is possible that coherency issues were left to the programmer to save power in the small handheld device. It is also possible, although less likely, that the release date was given priority over ease of programming. This communication is very different from the traditional classroom concepts of parallelism. Two processors were not included to make a single process run faster, but to allow for two different processes to be run at the same time. Neither processor will rely on the results from the other (although this could be enforced by the programmer), so the complexities of instruction level parallelism do not apply.

**Comparison Between Devices**

**Differences in Graphics**

The DS and GBA have very different graphics systems. In the Advance, there are background and foreground layers to the display, and everything runs on a tile system. This means that in order to display anything to the screen the programmer must edit the tile’s value in a table to point to the sprite or graphic that should be displayed. The DS, on the other hand, has a rendering system that is capable of displaying 3-dimensional environments. Its ARM9 processor has its own rendering engine which, although not incredibly powerful, is a significant step up from purely 2-dimensional displays.

The Gameboy Advance only has 1 KB of sprite RAM. While a programmer can store as many sprites as they want in main memory, no more than 128 of them can be displayed to the screen at any one time. Most of the challenge of making a GBA game came from working around RAM limitations and the idiosyncrasies that came with working with so many varied technologies in one system [5]. For the Nintendo DS, memory is less of a problem. It has 8KB of memory in its instruction cache alone and has 4MB in its main RAM. It is because of this increase in memory, speed, and efficiency that two screens and a rendering engine can be included in a handheld device.

The biggest and most glaring difference between the two systems, however, is the screen types. As noted previously, Nintendo wished to fight its reputation as conservative after a number of similar releases in its Game Boy series [7]. While the GBA has one tile-based screen, Nintendo included two screens in the DS, one of them a touch screen. It also has improved resolution on the both of them. They did this without sacrificing backwards compatibility, as older games are run on a single one of the screens, which the player can choose.

**Cache**

Another major difference between the DS and the GBA is that the GBA has no cache. It has “fast memory” and “slow memory”, but none of it is actually a cache at all. The difference between these is that the device will not try to decide what will be placed in the fast memory. Instead, this is left up to the programmer. Despite this, both consoles are able to run games at a framerate of 60 frames per second.

The lack of a cache in the GBA may have been due to cost or power issues. The device runs on two double-A batteries and it would be extremely frustrating if the batteries ran out in just a few hours. The DS, on the other hand, is rechargeable. People are much more willing to plug their device in than continually buy batteries. Furthermore, the cost of adding a cache could have been substantially less after the separation in release dates.

**Dual Processors**

Both the DS and the GBA have two processors to use. For backwards compatibility, they both shut off the newer processor and exclusively use the older one. But when running newer games, they differ--the GBA doesn’t use the old processor at all and conserves power, while the DS uses its older processor to assist with basic calculations. Having two processors is a very simple solution to the potentially complex problem of backwards compatibility, as it’s much cheaper to buy old hardware than to try to adjust new processors to run old games.

**Conclusion**

Though the GBA and the DS came out more than ten years ago, they were fantastic systems for their time, paving the way for the future of gaming and spawning many knock-off systems. The two systems carry surprising similarities--their outward appearance is entirely different but they share very similar internal setups. This design has continued on and their most recent successor, the New 3DS, is still successful on the market with games continually coming out. Perhaps this is due to the performance limitations of a handheld device, or perhaps Nintendo sees no need to change a continually successful formula. Either way, Nintendo’s successful design remains an interesting study of a small and common system.

**Bibliography**

[1] Z*80 CPU User Manual*, Zilog, San Jose, CA., 2016

[2] *ARM7TDMI Technical Reference Manual*, ARM, 1994. Available: http://www.atmel.com/images/ddi0029g\_7tdmi\_r3\_trm.pdf

[3] Jaeden Amero. GBA ASM Tutorial [Online]. Available:

https://patater.com/gbaguy/gba/ch1.htm

[4]Martin Korth (2001-2014). *GBATEK Technical Info* [Online]. Available: http://problemkaputt.de/gbatek.htm

[5] Open Source Development Labs. (2008, 7, 17). *A guide to homebrew development for the Nintendo DS.* [Online] Available: http://osdl.sourceforge.net/main/documentation/misc/nintendo-DS/homebrew-guide/HomebrewForDS.html

[6] ARM Limited. (2001, 2, 16). *ARM946E-S Technical Reference Manual*. [Online] Available: http://pdf.datasheetarchive.com/datasheetsmain/Datasheets-10/DSA-180307.pdf

[7] N-Sider Media. (2004, 3, 1). *Various Satoru Iwata comments regarding the Nintendo DS.* [Online] Available: http://www.n-sider.com/contentview.php?contentid=515