Department of Electronic and Telecommunication Engineering

University of Moratuwa

Report on Processor Dissection

AMD Athlon XP (CISC)
ARM Cortex-A710 (RISC)

EN2031: Fundamentals of Computer Organization and Design

Group: Neumann++

Index Number	Name
210069F	B.Y.N. Basnayake
210258J	R.M.K.C. Jayathissa
210542B	R.M.L.H. Ratnayake

Task		AMD Athlon XP	ARM Cortex A710
ISA	Instruction set	210542B	210069F
	Instruction classes and instruction formats		
Microarchitecture	Datapath	210258J	210542B
	Controller		
ALU functions		210069F	210258J
Memory	Cache memory	210258J	210542B
	Memory interfacing		
	Timing related to memory		
Critical comparison of the two processors		210069F	

Contents

Chapter 1

AMD Athlon XP

1.1 Overview

AMD Athlon XP is a 32-bit processor released by AMD in 2001. It mainly targeted the desktop PCs and was originally developed by AMD to compete with Intel Pentium processors. AMD Athlon XP is a Complex Instruction Set Computing (CISC) processor running the x86 instruction set architecture.

1.2 Instruction Set Architecture (ISA): AMD Athlon XP

1.2.1 Instruction Set

AMD Athlon XP is a microprocessor designed and released by AMD around 2001, that uses the x86 32-bit architecture originally designed by Intel. Other than that, the processor makes use of the x87 which is a floating-point subset of x86 for enhanced floating-point calculations. The model 10 of AMD Athlon XP is binary compatible with the existing x86 architecture while it is backward compatible with the MMX instruction model which enables the processor to perform 4 32-bit single-precision floating-point calculations per one clock cycle.

x86 is a CISC architecture which uses complex instructions that may take multiple clock cycles to run. On contrast to RISC-V, the x86 Instruction Set Architecture (ISA) uses instructions varying from 1 byre to 15 bytes. This makes it harder to implement at the microarchitecture level, however it has its own advantages as well such as having more complex instructions and formats that the programmers find easy to work with. 1 byte is used for the opcode field, of which 1 bit specified whether the operands are 8-bit or 32-bit long. The opcode field also specifies the addressing mode of that instruction. There is a field called the "post-byte" field in the instruction which is an extra opcode field giving some more information about the nature and the behavior of the instruction.

1.2.2 Instruction Classes and Instruction Formats

The integer operations of the x86 ISA used in AMD Athlon XP can be divided into four categories.

1. Arithmetic and logic instructions
2. Data movement instructions
3. Control flow/ Branching instructions
4. String instructions

For the first two types of instructions, it can be observed that the ISA uses five instruction types. Table 1.1 shows the instruction types of the arithmetic, logical and data transfer instructions.

It is noteworthy that x86 uses a Base plus Scaled Index addressing mode unlike in RISC-V. Here, a scale factor is used to scale the index present in a register to a byte address. A scale factor of 0 is used for no scaling, 1 for 16-bit addresses, and 2 for 32-bit addresses.

Source/ Destination Operand Type	Second Source Operand
Register	Register
Register	Immediate
Register	Memory
Memory	Register
Memory	Immediate

Table 1.1: Arithmetic, Logical and Data Flow Instruction Types of x86

$$Address = Base + (2^{ScaleFactor} * Index)$$

The most common type of register-memory type, where either source of destination address is a memory address rather than a register. The other operand can be a register or an immediate value.

1.3 Microarchitecture: AMD Athlon XP K7 Processor

The AMD Athlon XP processors were based on the AMD K7 microarchitecture. The K7 microarchitecture, also known as the "Athlon" architecture, was a significant advancement for AMD and marked their entry into high-performance desktop computing.

1. **Superscalar Execution**: The AMD-K7 processor supports superscalar execution, meaning it can execute multiple instructions in parallel, decoding complex x86 instructions into simpler fixed-length Macro-Ops.
2. **Nine-Stage Pipeline**: The processor operates using a nine-stage processing pipeline, which allows for efficient processing of instructions.
3. **Macro-Ops**: Instead of executing complex x86 instructions directly, the processor translates them into Macro-Ops, which are simpler fixed-length operations. These Macro-Ops are then executed independently in the integer or multimedia pipelines.
4. **Instruction Decoding**: X86 instructions are categorized into DirectPath and VectorPath decodes. DirectPath decodes common instructions, delivering up to three Macro-Ops per cycle, while VectorPath decodes complex or less common instructions.
5. **Branch Prediction**: The processor employs sophisticated dynamic branch prediction logic to minimize delays caused by branch instructions. This includes a branch prediction table, a branch target address table, and a return address stack.
6. **Cache Architecture**: The processor features a split 128-Kbyte writeback level-one cache (64 Kbytes instruction cache and 64 Kbytes data cache) and supports up to 512 Kbytes of L2 cache. It uses the S2K interface technology for system memory access, providing high bandwidth.
7. **Execution Units**: The processor has multiple integer and floating-point schedulers for executing Macro-Ops concurrently.
8. **Instruction Control Unit (ICU)**: The ICU manages instruction dispatch, exception handling, and instruction retirement operations. It can handle up to 72 Macro-Ops simultaneously.
9. **Affordable Performance**: The AMD-K7 processor delivers high x86 performance and full x86 binary software compatibility at a reasonable cost.

In summary, the AMD-K7 processor's microarchitecture utilizes various techniques such as superscalar execution, efficient instruction decoding, advanced branch prediction, and cache architecture to achieve high-performance computing. This technology allows the processor to execute complex tasks efficiently, making it a significant advancement in the field of computer processing.

1.4 ALU Functions: AMD Athlon XP

The AMD Athlon XP processors, released in the early 2000s, featured a robust Arithmetic Logic Unit (ALU) responsible for a wide range of essential functions. These functions enabled the Athlon XP to perform a variety of computations and data manipulations, though they have since been surpassed by more powerful and efficient processor architectures.

1. Arithmetic Operations

The Athlon XP's ALU executed fundamental arithmetic operations, including addition, subtraction, multiplication, and division. These operations formed the foundation for various mathematical computations.

2. Logical Operations

The ALU managed logical functions, such as bitwise AND, OR, XOR, and NOT. These operations were critical for data manipulation and control flow in software programs.

3. Shift and Rotate Operations

The processor performed bit-shift and rotate operations, allowing for binary data manipulation and bit-level processing, which was essential for various applications.

4. Comparison and Conditional Operations

The ALU compared values to determine relationships such as equality, greater or lesser, which were vital for decision-making in conditional statements and branching within programs.

5. Data Movement

It facilitated the transfer of data between registers and memory locations, enabling efficient data handling and processing.

6. Address Calculation

The ALU played a key role in calculating memory addresses, which was crucial for data and instruction retrieval, ensuring the smooth execution of programs.

7. Floating-Point Operations

In addition to integer operations, Athlon XP processors integrated Floating-Point Units (FPUs) to handle scientific and multimedia computations, further expanding their capabilities.

In conclusion, the AMD Athlon XP processors were a significant part of computing history, offering a versatile ALU that executed diverse instructions and complex calculations. While they have been surpassed by more modern and efficient processor architectures, their legacy in the evolution of microprocessors remains noteworthy.

1.5 Memory: AMD Athlon XP

1.5.1 Memory Hierarchy

The AMD Athlon XP processors employed a hierarchical memory system, with cache memory at different levels to improve performance.

- **L1 cache**: The L1 cache is the fastest cache memory, located on the processor die itself. It stores the most frequently accessed instructions and data, allowing the processor to access them quickly without having to go to main memory.
- **L2 cache**: The L2 cache is slower than the L1 cache, but it is also larger. It stores less frequently accessed instructions and data, and it can serve as a buffer between the L1 cache and main memory.
- **Main memory (RAM)**: Main memory is the slowest type of memory, but it is also the largest. It stores all of the instructions and data that the processor needs, but it can take longer for the processor to access them than it would for the L1 or L2 cache.

The Athlon XP processors had separate L1 instruction and data caches, each with a capacity of 64 KB. This design allowed for swift access to frequently used instructions and data, enhancing overall processing speed. The L2 cache was unified, with a capacity of 256 KB or 512 KB, depending on the processor model.

1.5.2 Memory Interfacing

The Athlon XP processors used a front-side bus (FSB) to communicate with the system's main memory (RAM) and other essential components. The FSB speed varied among different Athlon XP models, impacEng the data transfer rate between the processor and memory. The memory controller was integrated into the Northbridge chipset on the motherboard, not directly on the processor die. This architectural approach was common during the Athlon XP era and facilitated efficient communication between the processor and system memory.

1.5.3 Timing Related to Memory

The Athlon XP processors adhered to specific memory timings, including parameters like CAS latency, RAS to CAS delay, and RAS precharge time. These timings dictated how swiftly the processor could read from or write data to the memory modules, ensuring efficient memory operations.

1.5.4 Memory Speed Support and Bandwidth

The Athlon XP processors supported various memory speeds, including DDR (Double Data Rate) memory types such as PC2100 and PC2700. The memory speed, expressed in MHz, denoted the data transfer rate between the processor and memory.
The combination of the FSB and memory speed directly influenced memory bandwidth. Higher memory bandwidth facilitated faster data access, significantly enhancing overall system performance, especially during memory-intensive tasks.

1.5.5 Conclusion on Memory

The AMD Athlon XP processors featured a robust memory hierarchy and efficient memory management capabilities. This allowed them to deliver high performance for a wide range of computing tasks.
It is important to note that the specific cache sizes, memory support, and timings varied among different Athlon XP models. This was done to meet the performance requirements of diverse computing needs.

Chapter 2

ARM Cortex-A710

2.1 Overview

ARM Cortex A710 is a processor core based on Reduced Instruction Set Computing (RISC) developed by ARM and released in 2021. The processor is based on the ARMv9 Harvard architecture. It is mainly used as a mobile processor and can be found in Qualcomm Snapdragon 7/7+/8/8+ and Samsung Exynos 2200 series mobile processors.

2.2 Instruction Set Architecture (ISA): Cortex-A710

The ARM Cortex-A710 processor is built upon the ARMv9.0-A Instruction Set Architecture (ISA) and incorporates advanced features and capabilities.

Key Features of the ARMv9.0-A ISA:

1. **Security Enhancement:** The ARMv9.0-A ISA prioritizes security, introducing features like Pointer Authentication to defend against memory corruption attacks. This ensures data integrity and confidentiality in both user and kernel modes, making the Cortex-A710 suitable for secure computing environments.

2. **Scalability:** The ISA is inherently scalable and supports a wide range of configurations, from single-core embedded devices to multi-core servers, catering to the diverse needs of IoT and high-performance computing applications.

3. **Virtualization Support:** ARMv9.0-A provides enhanced support for virtualization, simplifying and securing the management of virtual machines in data centers and cloud computing environments, optimizing resource usage.

4. **Advanced SIMD:** The ISA continues to support NEON SIMD technology, significantly accelerating multimedia and signal processing workloads, making the Cortex-A710 ideal for multimedia and digital signal processing applications.

5. **Extended Vector Processing:** ARMv9.0-A introduces Scalable Vector Extensions (SVE2), offering advanced support for vector processing tasks with applications in scientific computing and artificial intelligence.

6. **Ample Address Space:** The architecture extends the virtual address space to 64 bits, supporting extensive memory configurations, addressing the needs of memory-intensive applications like databases and scientific computing.

The utilization of the ARMv9.0-A ISA in the Cortex-A710 positions it as a highly capable and adaptable solution for various computing applications. It forms the foundation for the Cortex-A710's advanced security, virtualization support, and performance capabilities.

In conclusion, the integration of the ARMv9.0-A ISA within the Cortex-A710 demonstrates its commitment to advancing computing performance, security, and versatility.

2.3 Microarchitecture: ARM Cortex-A710

ARM Cortex A710 is designed for the ARMv9.0-A, or more formally AArch64 architecture. Moreover, it is one of a few processor cores that support both AArch32 and AArch64. It is noteworthy that this ISA uses fixed length instructions, and the integers are little-endian.

2.3.1 Datapath

The ARM Cortex A710 uses an advanced superscalar model. The processor has separate registers for integer and floating-point values to be stored. Moreover, the datapath has multiple execution units for arithmetic and logic, integers and floating-point values, and memory operations. On the other hand, the processor has a Load-Store Unit which manages the efficiency of flow of data in the datapath for the memory related instructions.

2.3.2 Controller

The ARM Cortex A710 utilizes a 10-stage pipeline with out-of-order execution of instructions enhancing the efficiency of the processor. Since the ISA uses fixed-length instructions compared to some other ISAs like x86, the design of the controller is somewhat simpler.

There is a branch predictor which predicts what instruction should be fetched next from the L1 cache. Then, those instructions are given to the decoder. After decoding, the instructions are converted to micro-op codes which are essentially more simplified instructions for the hardware. Then, the micro-op codes are used to map the architectural registers to physical registers in the register file and this is done by a unit called “Renamer”.

There is a unit called the “Scheduler” which facilitates the out-of-order execution of instructions. It lets the ALU, or any other execution unit execute the instructions whenever all the inputs/ operands are ready. For this store-forwarding concept is used where the operands are given to the instructions on-flight.

2.4 ALU Functions: ARM Cortex-A710

The Cortex-A710 processor features a range of Arithmetic and Logical Unit (ALU) functions in both AArch64 and AArch32 instruction sets. These functions are designed for executing arithmetic and logical operations.

The Cortex-A710 processor features a range of Arithmetic and Logical Unit (ALU) functions in both AArch64 and AArch32 instruction sets. These functions are designed for executing arithmetic and logical operations.

The AArch64 ALU functions include basic ALU operations, extend and shift operations, arithmetic operations, conditional compare and select operations, convert floating-point condition flags operations, flag manipulation instructions, insert random tags instructions, insert tag mask instructions, logical operations, subtract pointer operations, and insert random tags instructions.

The AArch32 ALU functions include basic ALU operations, arithmetic operations, conditional operations, shift-by-register operations, logical operations, and shift-by-immediate operations.

The ALU functions in the Cortex-A710 processor are designed to provide high performance and efficiency for a wide range of applications.

- ALU basic - ADD, ADC, AND, BIC, EON, EOR, ORN, ORR, SUB, SBC
- ALU basic, flagset - ADDS, ADCS, ANDS, BICS, SUBS, SBCS
- ALU extend and shift - ADD{S}, SUB{S}
- Conditional compare - CCMN, CCMP
- Conditional select - CSEL, CSINC, CSINV, CSNEG

2.5 Memory: ARM Cortex-A710

2.5.1 Cache Memory and Timing Related to Memory

The ARM Cortex A710 processor core comes with L1, L2, and L3 caches from varying sizes and speeds.

Cache Memory		**Size**	**Speed**
L1	Instruction cache	32KB or 64KB	48 bytes / cycle
	Data cache	32KB or 64KB	
L2 (Unified cache)		256KB or 512KB	20 bytes / cycle
L3		Up to 16MB	

Table 2.1: L1, L2, and L3 Cache Size and Speed

The L1 cache consists of two separate caches: Instruction cache and Data cache, while the L2 cache is a unified private cache. The L1 cache memories are virtually indexed and physically tagged while the L2 cache is physically indexed and physically tagged.

The L1 cache's main duty is to provide an instruction stream to the decoder. It also works in the fetch stage of the instructions and helps with dynamic branch prediction. The L1 Instruction cache consists of three main components the instruction cache itself, Instruction Translation and Lookaside Buffer (TLB), Macro-Operation (MOP) cache. The L1 Data cache and the L2 cache have their respective TLB units as well.

2.5.2 Memory Interfacing

The ARM Cortex A710 is equipped with a Memory Management Unit (MMU) which basically translates input addresses to output addresses. This is done by address mapping through information available in the internal registers and translation tables (TLBs). This process has multiple stages: translating from input virtual address to output physical address or intermediate physical address directly or translating from input virtual address to intermediate physical address and then intermediate physical address to output physical address.

The L1 data cache system interfaces with the integer or vector pipelines in the following ways.

- 3x64-bit read paths and 4x64-bit write paths for the integer execute pipeline.
- 3x128-bit read paths and 2x128-bit write paths for the vector execute pipeline.

If multiple cores of Cortex A710 are used to create DynamIQ Shared Unit (DSU), which is an advanced application of the processor cores, the L2 cache a CHI Issue E compliant interface with 256-bit read and write DAT channel widths.

Other than that, the processor can access the internal memory directly when necessary.

Chapter 3

Critical Comparison of the AMD Athlon XP and ARM Cortex-A710

Instruction Set Architecture (ISA)

AMD Athlon XP The AMD Athlon XP utilizes a 32-bit x86 CISC architecture, originally designed by Intel. It is also backward compatible with MMX instructions, which enhance its multimedia capabilities.

ARM Cortex-A710 The ARM Cortex-A710 is built on the ARMv9 RISC architecture. This processor supports both AArch64 and AArch32 instruction sets. It features advanced security enhancements, scalability, virtualization support, and advanced SIMD and vector processing.

Microarchitecture

AMD Athlon XP The AMD Athlon XP is based on the K7 microarchitecture, featuring a superscalar execution model. It operates with a nine-stage pipeline, Macro-Ops translation, advanced branch prediction mechanisms, and a split L1 cache.

ARM Cortex-A710 The ARM Cortex-A710 employs a 10-stage pipeline with out-of-order execution. It also incorporates branch prediction, fixed-length instructions, and register mapping.

ALU Functions

AMD Athlon XP The AMD Athlon XP's ALU functions encompass arithmetic, logic, data movement, control flow, and string operations, making it suitable for various desktop computing tasks.

ARM Cortex-A710 The ARM Cortex-A710 provides ALU functions for arithmetic, logical operations, shift operations, conditional comparisons and selections, conversion of floating-point condition flags, flag manipulation, and more.

Memory

AMD Athlon XP The AMD Athlon XP employs a hierarchical memory system, comprising L1 and L2 caches, and main memory (RAM) with a front-side bus (FSB) interface. This design facilitates efficient data access for desktop computing.

ARM Cortex-A710 The ARM Cortex-A710 utilizes L1 (instruction and data), L2, and L3 caches with varying sizes and speeds. It also incorporates a Memory Management Unit (MMU) for address translation and memory interfacing. This memory hierarchy optimizes performance for mobile applications.

In short, the AMD Athlon XP and ARM Cortex-A710 processors are designed for different types of computing tasks. The AMD Athlon XP is a 32-bit processor meant for desktop computers, while the ARM Cortex-A710 is a 64-bit processor designed for use in mobile devices. The unique designs of their internal structures, arithmetic and logic capabilities, and memory systems are optimized for their specific intended uses.

Bibliography

[1] "AMD Athlon XP Processor Model 10 Data Sheet," [Online]. Available: https://en.wikichip.org/w/images/3/38/AMD_Athlon_XP_Processor_Model_10_Data_Sheet.pdf.

[2] "AMD-K7 Technical Brief," [Online]. Available: https://www.epos.ua/site/file_uploads/ru/about_publication/amdk7tb.pdf.

[3] "X87," [Online]. Available: https://en.wikipedia.org/wiki/X87.

[4] D. A. Patterson and J. L. Henessey, "Computer Organization and Design – RISC-V Edition."

[5] "ARM's Cortex-A710: Winning by Default?," [Online]. Available: https://chipsandcheese.com/2023/08/11/arms-cortex-a710-winning-by-default/.

[6] "ARM Cortex-A710," [Online]. Available: https://en.wikipedia.org/wiki/ARM_Cortex-A710.

[7] "AArch64 - ARMv9-A," [Online]. Available: https://en.wikipedia.org/wiki/AArch64#ARMv9-A.

[8] "ARM Cortex-X2, A710, and A510: A look at ARM's latest CPUs," [Online]. Available: https://www.androidauthority.com/arm-cortex-x2-a710-a510-1225203/.

[9] "ARM Cortex-A710," [Online]. Available: https://www.wikiwand.com/en/ARM_Cortex-A710.

[10] "ARM Documentation," [Online]. Available: https://developer.arm.com/documentation/101800/0201/.

[11] "AMD Athlon XP Processors," [Online]. Available: https://www.cpu-world.com/CPUs/K7/TYPE-Athlon%20XP.html.

[12] "AMD K7 Microarchitecture," [Online]. Available: https://www.epos.ua/site/file_uploads/ru/about_publication/amdk7tb.pdf.

[13] "AMD K7 Microarchitecture - Memory Hierarchy," [Online]. Available: https://en.wikichip.org/wiki/amd/microarchitectures/k7#Memory_Hierarchy.