EN2031 - Fundamentals of Computer Organization and Design

TEAM NITRO S

Processor Dissection

Prepared by:

DULNATH W.H.R :210152E

U.M.Y.B. ALAHAKOON :210027C

SHAMIKA K.A.M. :210600D

Abstract

The processor is considered as the brain of a computer as it performs computational and logical functions together with control instructions. This report conducts an in-depth discussion and comparison of two distinct processors, each representing a different instruction set architecture. The AMD Phenom X4, which adheres to the Complex Instruction Set Computer (CISC) architecture, and the Cortex A55, which represent the Reduced Instruction Set Computer (RISC) architecture. The examination focuses on key characteristics of their instruction set architectures, such as instruction sets, classes, and formats, micro-architectural aspects like data paths and controllers, ALU functions, cache memory utilization, memory interfacing, and timing-related characteristics associated with memory access. The choice between these processors ultimately depends on specific application requirements and power constraints, and this report offers a detailed comparison of their distinctions to facilitate well-informed decisions.

Table of Contents

AMD Phenom X4 Processor

1.1. Introduction

The AMD Phenom X4, a representative of the CISC architecture, stands out with its comprehensive instruction set designed for desktop computing, which includes arithmetic, logic, control, and data manipulation operations. Its micro-architecture features advanced techniques such as out-of-order execution and branch prediction, contributing to enhanced performance. The memory subsystem of the Phenom X4 is also equipped with several cache levels, which reduces memory latency and improves system performance. The optimization of the processor's total performance, however, depends significantly on the subtle connections between the processor's performance and memory access challenges, such as cache efficiency and data transfer complications.

1.2. Instruction Set Architecture

The ISA of the AMD Phenom X4 processor, designed by Advanced Micro Devices (AMD), is an amazing representation of the Complex Instruction Set Computer (CISC) concept. It has a broad and extensive collection of instructions that enable the processor to carry out a variety of complex operations. It is based on the x86 instruction set architecture.

1.2.1. Instruction Formats and Classes

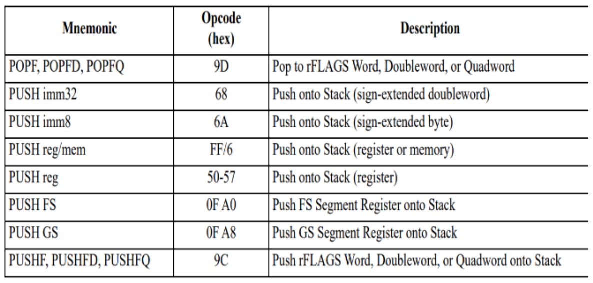
AMD Phenom X4 processors support the AMD64 instruction set, which is an extension of the x86 instruction set. This instruction set allows for 64-bit computing, providing compatibility with both 32-bit and 64-bit software. Its ISA represents these features by providing a broad range of instructions that can carry out operations in the areas of logic, control, and data manipulation. The instructions are described in the following subsets are used in AMD64 ISA.

- **General-Purpose Instructions:** The basic x86 integer instructions known as general-purpose instructions are those that are used in almost all programs. The majority of these instructions load, save, or perform operations on data that is present in memory or the general-purpose registers.
- **Branching Instructions:** The AMD64 architecture expands two branching mechanisms to accommodate branches in the full 64-bit virtual-address space. Near-branch semantics has been redefined in 64-bit mode. A 64-bit call-gate descriptor is defined for far calls in both compatibility mode and 64-bit mode. The SYSCALL and SYSRET directives from the past have also improved.
- **Reference RSP Instructions:** All instructions that implicitly refer to the 64-bit stack pointer, RSP, other than separated branches, default to a 64-bit operand size in 64-bit mode. With these instructions, 32-bit stack values cannot be pushed or popped in 64-bit mode, but they can be overridden to 16 bits.
- **SWAPGS Instruction:** The AMD64 ISA provides the SWAPGS instruction as a fast method for system software to load a pointer to system data-structures. SWAPGS is valid only in 64-bit mode.

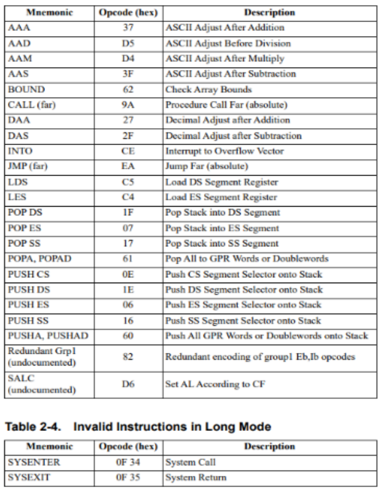
1.2.2. Types of common Instructions Use in AMD Phenom X4 Processor

- **Arithmetic Instructions:** ADD, SUB
- **Logical Instructions:** Bitwise AND Bitwise OR
- **Data Transfer Instructions:** LOAD, STORE
- **Stack Operations:** PUSH, POP
- **Control Signals:** JUMP, BEQ
- **Comparison and Conditional Branching Instructions:** BGT, BLT
- **8 Bit LOAD Immediate Instructions:** LDI

Table01: Reference (RSP) instructions



Mnemonic	Opcode (hex)	Description
POPF, POPFD, POPFQ	9D	Pop to rFLAGS Word, Doubleword, or Quadword
PUSH imm32	68	Push onto Stack (sign-extended doubleword)
PUSH imm8	6A	Push onto Stack (sign-extended byte)
PUSH reg/mem	FF/6	Push onto Stack (register or memory)
PUSH reg	50-57	Push onto Stack (register)
PUSH FS	0F A0	Push FS Segment Register onto Stack
PUSH GS	0F A8	Push GS Segment Register onto Stack
PUSHF, PUSHFD, PUSHFQ	9C	Push rFLAGS Word, Doubleword, or Quadword onto Stack

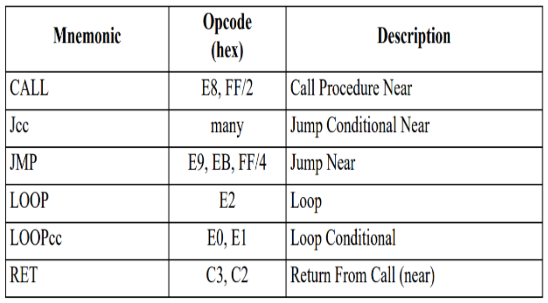


Mnemonic	Opcode (hex)	Description
AAA	37	ASCII Adjust After Addition
AAD	D5	ASCII Adjust Before Division
AAM	D4	ASCII Adjust After Multiply
AAS	3F	ASCII Adjust After Subtraction
BOUND	62	Check Array Bounds
CALL (far)	9A	Procedure Call Far (absolute)
DAA	27	Decimal Adjust after Addition
DAS	2F	Decimal Adjust after Subtraction
INTO	CE	Interrupt to Overflow Vector
JMP (far)	EA	Jump Far (absolute)
LDS	C5	Load DS Segment Register
LES	C4	Load ES Segment Register
POP DS	1F	Pop Stack into DS Segment
POP ES	07	Pop Stack into ES Segment
POP SS	17	Pop Stack into SS Segment
POPA, POPAD	61	Pop All to GPR Words or Doublewords
PUSH CS	0E	Push CS Segment Selector onto Stack
PUSH DS	1E	Push DS Segment Selector onto Stack
PUSH ES	06	Push ES Segment Selector onto Stack
PUSH SS	16	Push SS Segment Selector onto Stack
PUSHA, PUSHAD	60	Push All GPR Words or Doublewords onto Stack
Redundant Grp1 (undocumented)	82	Redundant encoding of group1 Eb,Ib opcodes
SALC (undocumented)	D6	Set AL According to CF

Table 2-4. Invalid Instructions in Long Mode

Mnemonic	Opcode (hex)	Description
SYSENTER	0F 34	System Call
SYSEXIT	0F 35	System Return

Table03: Other Instructions



Mnemonic	Opcode (hex)	Description
CALL	E8, FF/2	Call Procedure Near
Jcc	many	Jump Conditional Near
JMP	E9, EB, FF/4	Jump Near
LOOP	E2	Loop
LOOPcc	E0, E1	Loop Conditional
RET	C3, C2	Return From Call (near)

Table02: Branching instructions

1.3 Micro Architecture

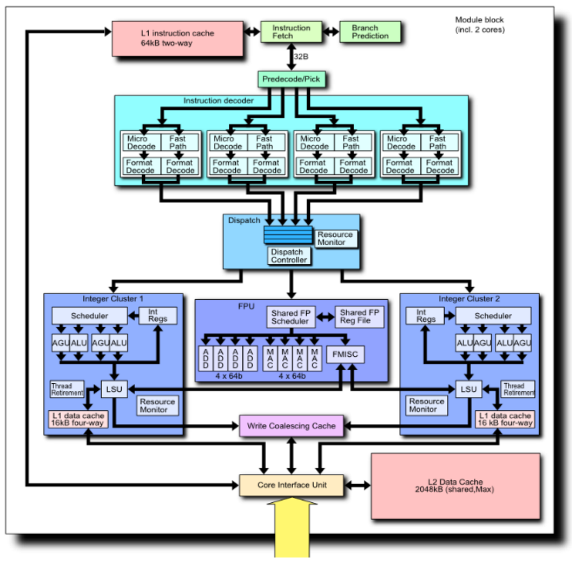
The Advanced Micro Devices (AMD) Phenom X4 is a quad-core processor that makes use of the K10 microarchitecture. The Phenom X4 processor, which was released in the late 2000s, represented a significant advancement in AMD's processor technology. A unique characteristic of this quad-core architecture was its capacity to handle numerous jobs concurrently, which improved system performance. Because of the processor's 64-bit architecture, it was possible to access bigger amounts of memory and do calculations with greater accuracy. Especially notable were AMD's Hyper Transport technology for high-speed networking and an integrated memory controller that improved memory access speeds. It featured a common L3 cache for all cores and a specialized L2 cache for each core. AMD's Cool ‘Quiet technology dynamically changed voltages and clock rates to conserve energy and lower heat. The Direct Connect Architecture was designed to reduce memory latency and boost system efficiency. Phenom X4 processors, in contrast to several of their rivals, did not support simultaneous multithreading (SMT), handling just one thread at a time per core. Even though they had good multi-core performance, Intel's Core 2 Quad CPUs posed a serious threat, and later AMD microarchitectures continued to advance.

1.3.1 Main Components of K10 Architecture

The AMD Phenom X4 CPU, based on the K10 microarchitecture, is designed with a number of essential elements. Its quad-core architecture allows for effective parallel processing and multitasking. Data access performance is improved by a cache structure that consists of shared Level 2 (L2) and Level 1 (L1) caches. Reduced memory latency and improved system performance are also benefits of the integrated memory controller. The use of Hyper Transport technology enables quick communication with other system parts.

Complex mathematical calculations are handled by floating-point units, whereas arithmetic and logical processes are handled by integer execution units.

Branch prediction reduces delays caused by incorrectly predicted branches and efficient instruction fetch and decode units prepare instructions for execution. Memory activities are managed by load/store units, and Cool ‘Quiet technology further lowers power usage. Power management features dynamically alter voltages and clock speeds for maximum energy efficiency. Together, these parts provide the Phenom X4 processor the ability to handle a variety of computer tasks, including gaming and content creation, making it a flexible and competent CPU.



Microarchitecture

1.3.2 Registers

A wide variety of registers that play different roles in the AMD Phenom X4 processor's microarchitecture are included in the processor's hardware. 32-bit registers are necessary for data manipulation and computation, and general-purpose registers (GPRs) are useful for carrying out arithmetic and logical operations. On the other hand, floating-point registers are adept at dealing with real-number arithmetic. While segment registers used to govern memory segmentation, they are now less important in 64-bit current processors like the Phenom X4. govern registers manage the behavior and system settings of the processor.

1.3.3 Addressing Modes

The Phenom X4 CPU provides a variety of options for effective memory access and operand manipulation in terms of addressing modes. Register Direct, Immediate, Register Indirect, Indexed, and Base-Indexed with Displacement are some of these modes. These addressing modes give instructions flexibility in how they can access and interact with data and memory locations, providing software developers and assembly language programmers with a useful set of tools.

1.3.4 Program Counters

In the x86 architecture, the Program Counter (PC), also known as the Instruction Pointer (IP), is a key component in managing the program flow inside the CPU. It keeps track of the memory address of the subsequent instruction that needs to be fetched and performed, ensuring that the program's instructions are executed in the proper order. The PC is increased to point to the following instruction in memory when the CPU performs instructions, supporting the smooth flow of a program's operations. These components—registers, addressing modes, and the program counter—all work together to enhance the Phenom X4's functionality and let it to carry out diverse computing tasks quickly and effectively.

1.4 ALU Functions

The AMD Phenom X4 processor featured a quad-core design. The Arithmetic Logic Unit (ALU) in this processor performed various essential functions.

- **Arithmetic Operations:** It executed addition, subtraction, multiplication, and division for both integer and floating-point numbers.
- **Logical Operations:** It handled logical operations like AND, OR, XOR and NOT, crucial for conditional statements, bit manipulation and Boolean algebra.

- **Shift and Rotate Operations:** The ALU could shift and rotate bits within data registers, useful for tasks like data manipulation and bit shifting.
- **Comparison Operations:** It compared values to determine relationships (e.g., equality, inequality, greater than, less than), critical for branching and decision-making in programs.
- **Data Conversion**: The ALU converted data between different types or representations (e.g: integer to floating-point).
- **Bitwise Operations:** It manipulated individual bits within data through operations like AND, OR, XOR and NOT, often used in low-level programming and data processing.
- **Conditional Operations:** The ALU could set or clear specific bits based on conditions.
- **Control Operations:** It played a role in control flow by executing branch and jump instructions that determined the program's order of execution.

1.5 Memory Interfacing and Cache Memory

The AMD Phenom X4 processor, introduced in the mid-2000s, is a quad-core microprocessor. It incorporates memory interfacing and cache memory for efficient performance.

Memory Interfacing: Memory interfacing is how the CPU communicates with system memory, such as RAM. For AMD Phenom X4 processors:

- They support dual-channel memory architecture for improved memory bandwidth.
- These processors typically work with DDR2 or DDR3 memory modules, affecting memory type and speed.
- An integrated memory controller efficiently manages data transfers between the CPU and RAM, reducing latency.

Cache Memory: Cache memory, a high-speed memory within the processor, stores frequently accessed data and instructions to reduce access time to slower main memory.

AMD Phenom X4 processors include:

- L1 Cache, dedicated to each core and divided into instruction and data caches. Small but very fast.
- Shared L2 Cache that all cores can access, with size variations among models.
- Some models may have a shared L3 cache among all cores, which enhances system performance by reducing memory latency for frequently used data.

Cache memory significantly enhances a processor's efficiency and performance by minimizing wait times for data retrieval from main memory.

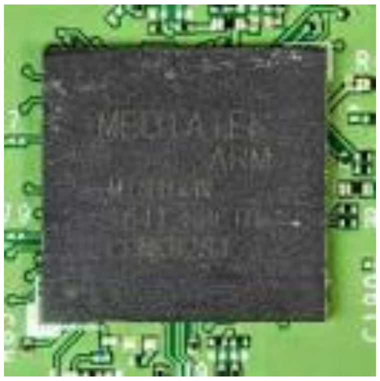
1.6. Timing Related to Memory

The AMD Phenom X4 processor has various memory-related factors to consider:

- Memory frequency (speed) depends on the motherboard and modules used.
- Lower CAS latency is better for memory performance, but it's influenced by the modules and motherboard.
- It uses a dual-channel memory controller, so installing memory in pairs is beneficial.
- The shared L3 cache can improve memory access times.
- The processor supports DDR2 and DDR3 memory types, impacting memory performance.
- An integrated memory controller in the processor manages memory access and affects performance.

ARM Cortex 55 Processor

2.1. Introduction



The Cortex A55, a common instance of the Reduced Instruction Set Computer architecture, is showing of effective and environmentally friendly designs. The Cortex A55, created by ARM Holdings, is a flexible processor designed to support a wide range of applications, including those in the mobile and embedded systems domains. This processor follows the ARMv8-A architecture and emphasizes energy economy while providing a productive 64-bit computing environment. In this section we delve into the various facets of the Cortex A55 processor, from its instruction set to microarchitecture, facilitating a comprehensive understanding for potential design considerations.

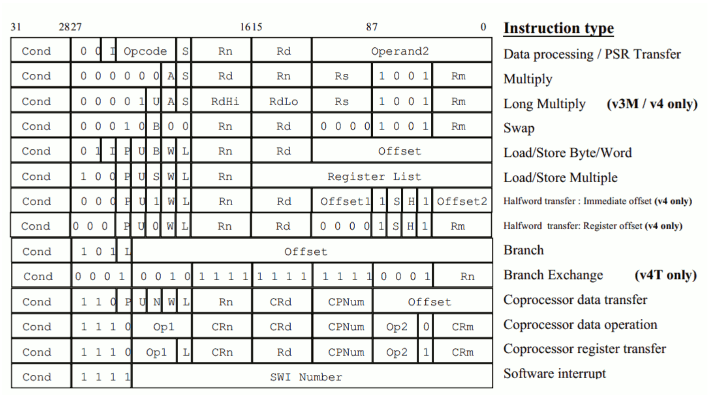
2.2. Instruction Set Architecture

The ARM Cortex-A55, a product of ARM Holdings' Cambridge design center, is a central processing unit that brings the power of the ARMv8.2-A 64-bit instruction set to the forefront of modern computing. The Cortex-A55 was designed to be flexible and efficient, and it has a 2-wide decode in-order superscalar pipeline that shows an innovative method of executing instructions.

2.2.1. Instruction Formats and Classes

The Cortex-A55's ISA is notable for its 64-bit architecture, which improves the processor's capacity for handling bigger data sets and memory addressing while keeping an eye on energy efficiency. Following instruction classes, which meet a variety of computing needs, serve as an example of the ISA's flexibility and capability.

- **Data processing Instructions**: These instructions manipulate information stored in registers. They include foundational operations in arithmetic and logic. Ex: ADD, SUB
- **Storage and Loading Instructions**: These instructions make it easier to transfer data between memory and registers. Ex: LDR (load register), STR (store register).
- **Control Flow Instructions**: Control flow instructions manage program execution, including branching and conditional execution, Ex: BEQ, BGT, BLT
- **Bit Manipulation Instructions**: Bit-level operations like AND, OR, and XOR are made possible by these instructions. For instance, XOR is performed between registers by EOR.
- **System Instructions:** System instructions have a high level of responsibility and are used to manage and control system resources. Ex: HLT (stop) and SVC (supervisor call).
- **Data Transfer Instructions**: These instructions handle data transfer between general-purpose registers and coprocessor registers, typical in system control tasks. Ex: MCR (move to coprocessor from register) and MRC (move to register from coprocessor).
- **Floating-Point and NEON Instructions**: Floating-point instructions are essential for scientific and engineering applications. NEON instructions support Single Instruction, Ex: FADD (floating-point addition) and VFMA (vector fused multiply-add).
- **64 bits Multiply and Divide Instructions**: Ex: UMULL (unsigned multiply long), SDIV (signed divide).



31 28	27	16 15		8 7		0	Instruction type
Cond	0 0 I Opcode S	Rn	Rd	Operand2			Data processing / PSR Transfer
Cond	0 0 0 0 0 0 A S	Rd	Rn	Rs	1 0 0 1	Rm	Multiply
Cond	0 0 0 0 1 U A S	RdHi	RdLo	Rs	1 0 0 1	Rm	Long Multiply **(v3M / v4 only)**
Cond	0 0 0 1 0 B 0 0	Rn	Rd	0 0 0 0	1 0 0 1	Rm	Swap
Cond	0 1 I P U B W L	Rn	Rd	Offset			Load/Store Byte/Word
Cond	1 0 0 P U S W L	Rn	Register List				Load/Store Multiple
Cond	0 0 0 P U 1 W L	Rn	Rd	Offset1	1 S H 1	Offset2	Halfword transfer : Immediate offset **(v4 only)**
Cond	0 0 0 P U 0 W L	Rn	Rd	0 0 0 0	1 S H 1	Rm	Halfword transfer: Register offset **(v4 only)**
Cond	1 0 1 L	Offset					Branch
Cond	0 0 0 1 0 0 1 0	1 1 1 1	1 1 1 1	1 1 1 1	0 0 0 1	Rn	Branch Exchange **(v4T only)**
Cond	1 1 0 P U N W L	Rn	CRd	CPNum	Offset		Coprocessor data transfer
Cond	1 1 1 0 Op1	CRn	CRd	CPNum	Op2 0	CRm	Coprocessor data operation
Cond	1 1 1 0 Op1 L	CRn	Rd	CPNum	Op2 1	CRm	Coprocessor register transfer
Cond	1 1 1 1	SWI Number					Software interrupt

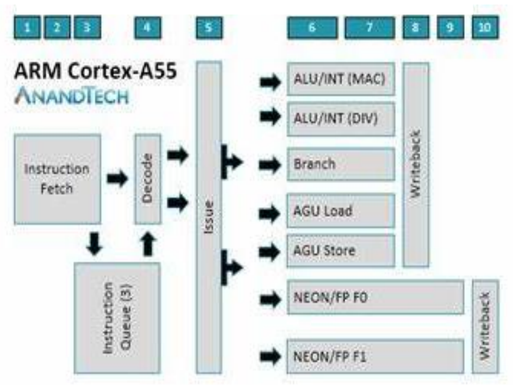
Table 04: Arm 8A 64bit Instruction set format.

- A fixed-length 32-bit instruction format is used in the ARM 64-bit (ARMv8-A) instruction formats, which are created for simplified and effective processing. These instructions enable high-performance and energy-efficient computing on ARM-based systems by covering a wide range of functions, including as data processing, control flow, load, and store, and more.

2.3. Microarchitecture

Leading semiconductor and software design company ARM Holdings created the Cortex-A55 microarchitecture. It is a component of the ARMv8-A architecture and is recognized for its adaptable and energy-efficient design. In many different types of computing devices, especially in the embedded systems and mobile technology industries, the Cortex-A55 microarchitecture is frequently employed. It has become well-liked for its capacity to provide a balance between performance and power efficiency, making it an appropriate option for uses where low power consumption and extended battery life are required. We shall examine the main characteristics and importance of the Cortex-A55 microarchitecture in the world of central processing units (CPUs) in this introduction.

2.3.1 Main Components of ARMv8-A architecture



ARM Cortex 55 Microarchitecture

The microarchitecture of the Cortex-A55 CPU is made up of numerous essential elements. Program instructions are retrieved from memory by the "Instruction Fetch" unit, which is then translated into micro-operations by the "Instruction Decode" step. The "Instruction Dispatch" unit then effectively plans and distributes instructions to the right execution units for expedited processing. A large variety of arithmetic and logical operations are handled by the "Integer Execute" component, and memory activities are managed by the "Load/Store Unit" by obtaining and storing data. A "Advanced SIMD and Floating-Point Unit" is also included in the processor to handle challenging mathematical calculations, which is especially useful for multimedia and scientific computations. Together, these parts give the Cortex-A55 the ability to efficiently handle a variety of tasks while putting an emphasis on energy saving.

2.3.2 Registers

Registers are a key component of the Cortex-A55 microarchitecture, enabling effective data handling and computing. These registers include floating-point registers designed for real-number calculations as well as general-purpose registers (GPRs) used for arithmetic and logical operations. While segment registers might aid with memory segmentation, control registers oversee the processor's behavior and system settings. The Cortex-A55's total computing capabilities are improved by the efficient processing of data and instructions made possible by these registers.

2.3.3 Addressing Modes

The Cortex-A55 microarchitecture's addressing modes offer adaptable ways to access memory and operands. A register can be used as a memory pointer in the Register Indirect mode to allow indirect access to data stored there, the Register Direct mode to allow direct operation on data stored in specific registers, the Immediate mode to include data as part of the instruction itself, the Indexed mode to calculate effective memory addresses by adding offsets to a base address in a register, and the Base-Indexed with Displacement mode to combine a base register, index register, and constant offset for memory addressing. This variety of addressing modes equips assembly language programmers and software developers with adaptable tools for efficient data access and manipulation.

2.3.4 Programing Counters

A crucial part of the Cortex-A55 microarchitecture is the Program Counter (PC), which controls how programs are executed. The Program Counter (PC), also referred to as the Instruction Pointer (IP) in some architectures, keeps track of the memory address of the following instruction that needs to be fetched and executed in order to preserve the orderly flow of program instructions. The PC is increased to point to the following instruction in memory as instructions are performed. It is crucial for regulating program flow and sequence, which enables the Cortex-A55 to effectively carry out commands and govern the logical flow of software processes.

2.4 ALU Functions

The ALU (Arithmetic Logic Unit) in ARM Cortex processors offers various essential functions.

- **Arithmetic Operations:** It can perform a broad range of arithmetic operations, handling both integer and floating-point numbers, including addition, subtraction, multiplication, and division.
- **Logical Operations:** It supports logical operations like AND, OR XOR and NOT, critical for tasks involving conditional statements, bitwise operations and Boolean algebra.
- **Shift and Rotate Operations:** ARM processors enable bit shifting and rotation within data registers, useful for data manipulation, scaling, and rotation.
- **Comparison Operations:** The ALU can compare values to determine relationships, such as equality, inequality, greater than, or less than, which is essential for branching and decision-making in software.
- **Data Conversion:** ARM processors can convert data between different types or representations, particularly crucial in data manipulation tasks.
- **Bitwise Operations:** Bitwise operations, including AND, OR XOR and NOT, facilitate manipulation of individual bits within data, beneficial for low-level programming and data processing.
- **Conditional Operations:** The ALU can handle conditional operations, enabling the setting or clearing of specific bits and actions based on specific conditions.

2.5 Memory Interfacing and Cache Memory

Memory Interfacing: Memory interfacing in the ARM Cortex-A55 processor relates to how the CPU connects with the system's memory, which includes RAM and other storage devices.

- **Memory Channels:** The processor can be part of a system-on-chip (SoC), and its memory interface depends on the SoC design. It can support different memory channels (e.g., LPDDR4, LPDDR4X) impacting memory bandwidth and overall system performance.
- **Memory Type:** Memory support is determined by the SoC and manufacturer's choices. It may work with various memory types (e.g., LPDDR4, LPDDR4X) that influence memory type and speed.
- **Memory Controller:** The Cortex-A55 processor itself lacks an integrated memory controller. Instead, it relies on the SoC's memory controller to efficiently manage data transfers between the CPU and RAM, reducing latency and enhancing system performance.

Cache Memory: Cache memory in the ARM Cortex-A55 processor is crucial for reducing memory access delays and enhancing system performance. It consists of three cache levels,

- **L1 Cache:** Each core has its dedicated L1 cache, split into separate instruction and data caches. These caches are small but very fast, minimizing latency.
- **L2 Cache:** Cortex-A55 cores generally share an L2 cache, which is larger than L1 and accessible by all cores. Its size can vary depending on the specific system-on-chip (SoC) design.
- **L3 Cache:** In some SoCs with Cortex-A55 cores, there may be a shared L3 cache. This cache improves overall system performance by reducing memory latency for frequently accessed data shared across multiple cores.

2.6 Timing Related to Memory

The Cortex-A55 processor incorporates a memory hierarchy with L1, L2, and potentially L3 caches, optimizing data access. It's adaptable to various memory types like LPDDR4, LPDDR4X, LPDDR5, and DDR4, impacting system performance through memory bandwidth. Specific memory timings, like CAS latency, are determined by memory modules and affect data access speed. In multi-core setups, memory channel sharing necessitates efficient access mechanisms to avoid latency. Cache coherency protocols, such as ARM's CCI, maintain consistency in multi-core systems, impacting data access timing. A memory controller coordinates memory access and can influence timings. The processor's clock speed also affects memory access, with faster cores requiring quicker memory access.

3. Comparison Between two Processors

In this section, we delve into a comprehensive comparison between two distinct but influential processor architectures: the AMD Phenom X4 and the ARM Cortex A55. These processors' important features, such as their ISAs, micro-architectures, memory hierarchies, and interfacing capabilities, will be explained by this comparison. We hope to offer helpful details to help decision-makers choose a processor for various computing needs by comprehending their advantages and trade-offs.

Feature	ARM Cortex A55	AMD Phenom X4
ISA Architecture	ARMv8.2-A 64Bit (RISC Architecture)	CISC Architecture
Instruction Width	64 Bits	32 Bits or 64 Bits
Supported Instruction Sets	A64	IA-32 (x86), IA-64 (x64), SSE4, SSE 4.1, SSE 4.2
Privilege Levels (ISA)	Provides privilege levels for secure execution	Usually runs in user mode and doesn't have many privilege levels.
Compatibility of ISA	Designed for power efficiency and compatibility with ARM-based systems	Developed for high-performance computing, often used in desktop systems.
Clock Speed	1.44 – 2.24 GHz	2.5 GHz
Feature Size	14 nm	16 nm
ALU	64-bit ALU	32-bit ALU
L1-I Cache (Per core)	32 KB, 4-way set associative	48 KB, 3-way set associative
L1-D Cache (Per core)	32 KB, 4-way set associative	32 KB, 4-way set associative
L2 Cache Size	512 KB, 16-way set associative	512 KB/1 MB/2 MB/4 MB, 16-way set associative or 2 x 1 MB 16-way shared caches
Memory Interface	Optimized for power efficiency and low latency in mobile and embedded applications.	Designed to handle high-performance computing tasks,

		catering to larger data sets and cache hierarchies.
Parallelism	In-order execution with moderate parallelism.	Out-of-order execution with higher parallelism.
Energy Efficiency	Designed for energy-efficient mobile and embedded applications.	Consumes more power but provides high processing power for desktop computing.
User Cases	Mobile devices, embedded systems, IoT, and energy-efficient applications	Desktop computers, gaming, multimedia processing, and high-performance computing.
Physical Size (Die Size)	Not specified	Approximately 258 mm²
Store Queue Size	42 entries	32 entries

The comparison of the ARM Cortex A55 and the AMD Phenom X4 processors reveals a stark contrast in design applications. The ARM Cortex A55 is the perfect choice for mobile, embedded, and IoT applications where saving battery life and controlling heat are important. Its energy-efficient architecture, compact form factor, and focus on power efficiency. The AMD Phenom X4 is a high-performance desktop processor, in contrast, with a rich but complex instruction set, a larger cache hierarchy, and a higher energy consumption profile, making it well-suited for tasks requiring for raw processing power in the areas of gaming, multimedia processing, and high-performance computing. The choice between these processors should align with the specific application requirements, where performance, power efficiency, and physical size play critical roles in the decision-making process.

4. References

- https://developer.arm.com/documentation/100442/latest/
- https://en.wikipedia.org/wiki/ARM_Cortex-A55#:~:text=The%20ARM%20Cortex%2DA55%20is,decode%20in%2Dorder%20superscalar%20pipeline.
- https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/24594.pdf
- https://www.scs.stanford.edu/05au-cs240c/lab/amd64/AMD64-1.pdf
- https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/Learn%20the%20Architecture/Armv8-A%20Instruction%20Set%20Architecture.pdf?revision=ebf53406-04fd-4c67-a485-1b329febfb3e

5. Task Allocation Among Group Members

DULNATH W.H.R	**Microarchitecture of the processors and comparison**
ALAHAKOON U.M.Y.B	**ISA's, ISA Formats, Classes of two processors and Comparison**
SHAMIKA K.A.M.	**ALU functions, Memory Interfacing of two processors and comparison**

****