University of Moratuwa

Department of Electronic Telecommunication Engineering

Processor Dissection

Team Jarvis

230585C Saruka U.

EN2031 – Fundamentals of Computer Organization and Design

Contents

1 Intel Core i5-6300U Processor 2

1.1 Overview . . . 2

1.2 Instruction Set Architecture . . . 2

1.3 Instruction Format . . . 2

1.4 Basic Instruction Classes Available . . . 3

1.4.1 Arithmetic and Logic Instructions . . . 3

1.4.2 Data Transfer Instructions . . . 3

1.4.3 Control Transfer Instructions . . . 4

1.4.4 Bit Manipulation Instructions . . . 4

1.4.5 String Instructions . . . 4

1.4.6 Floating-Point Instructions . . . 5

1.5 Micro-Architecture and Memory Organization . . . 5

1.5.1 Micro-Architecture . . . 5

1.5.2 ALU Operations . . . 5

1.5.3 Cache Memory and Memory Interfacing . . . 6

1.5.4 Timing Related to Memory . . . 6

2 ARM Cortex-M23 Processor 7

2.1 Overview . . . 7

2.2 Instruction Set Architecture . . . 7

2.3 Instruction Format . . . 7

2.4 Basic Instruction Classes Available . . . 7

2.5 Micro-Architecture . . . 8

2.6 ALU Operations . . . 8

2.7 Cache Memory and Memory Interfacing . . . 8

2.8 Timing Related to Memory . . . 8

3 Comparison of Intel Core i5-6300U and ARM Cortex-M23 9

Abstract

This report presents a detailed analysis and comparison of two processors belonging to different architectural families: the Intel Core i5-6300U processor used in the HP EliteBook 840 G3 laptop, and the ARM Cortex-M23 processor. The study examines their instruction set architecture, micro-architecture, ALU functionality, cache and memory systems, power consumption, and application domains. The objective of this report is to understand how processor design varies based on intended use cases such as high-performance computing and low-power embedded systems.

1 Intel Core i5-6300U Processor

1.1 Overview

The Intel Core i5-6300U is a dual-core, 64-bit processor based on Intel's Skylake microarchitecture and is manufactured using a 14 nm process technology. It is designed specifically for ultrabooks and business-class laptops, such as the HP EliteBook 840 G3. The processor supports simultaneous multithreading through Intel Hyper-Threading technology, allowing it to execute up to four threads concurrently. With a focus on balancing computational performance and energy efficiency, the Core i5-6300U provides reliable performance for everyday computing tasks while maintaining low power consumption, making it suitable for portable computing environments.

1.2 Instruction Set Architecture

The Intel Core i5-6300U processor used in the HP EliteBook 840 G3 is based on the Intel64 instruction set architecture, which is an extension of the IA-32 architecture and follows the Complex Instruction Set Computing (CISC) philosophy. Unlike RISC architectures, CISC architectures provide a large and diverse set of instructions, many of which are capable of performing complex operations in a single instruction. This design approach reduces the number of instructions required to execute a program, thereby improving code density and software compatibility.

The Intel64 architecture is a 64-bit ISA, enabling the processor to efficiently handle large memory address spaces and perform 64-bit arithmetic operations. A key feature of the Intel architecture is its backward compatibility with legacy 32-bit and 16-bit software, allowing applications developed for earlier x86 processors to execute without modification. The Intel Core i5-6300U also supports several instruction set extensions, such as Streaming SIMD Extensions (SSE) and Advanced Vector Extensions (AVX and AVX2), which enhance performance in multimedia, scientific, and parallel processing applications.

1.3 Instruction Format

The instruction format of the Intel64 architecture is variable in length, which is a defining characteristic of CISC-based processors. Instructions can range from one to fifteen bytes in length, depending on the operation and the addressing modes used. This variable-length format allows frequently used instructions to be encoded compactly while still supporting complex instructions with multiple operands and addressing modes.

A typical Intel64 instruction consists of optional prefix bytes, an opcode field, operand specifiers, displacement fields, and immediate data fields. Prefix bytes modify instruction behavior, such as operand size or addressing mode, while the opcode specifies the operation to be performed. Although variable-length instructions increase decoding complexity, modern Intel processors internally translate these instructions into simpler micro-operations, allowing efficient execution within the processor pipeline.

1.4 Basic Instruction Classes Available

The Intel Core i5-6300U supports a wide range of instruction classes that enable general-purpose computing. Arithmetic and logic instructions perform integer and logical operations such as addition, subtraction, multiplication, division, and bitwise processing. Data transfer instructions move data between registers, memory, and input/output interfaces, supporting various addressing modes.

Control transfer instructions are used to modify program execution flow through conditional and unconditional jumps, procedure calls, and returns. In addition, the processor supports bit manipulation instructions, string instructions for block data operations, floating-point instructions executed by the floating-point unit, and SIMD instructions that allow parallel data processing. These instruction classes collectively provide the flexibility and performance required for multitasking and computation-intensive applications.

1.4.1 Arithmetic and Logic Instructions

- Arithmetic and logic instructions form the core computational capability of the Intel Core i5-6300U processor and are executed by the Arithmetic Logic Unit (ALU).
- These instructions perform basic mathematical operations such as addition, subtraction, multiplication, and division on integer data.
- Logical operations including AND, OR, XOR, and NOT are also supported and are used for decision making and bit-level data processing.
- Arithmetic and logic instructions operate primarily on data stored in registers, though some instructions support direct memory operands.
- These instructions update processor status flags, enabling conditional execution and control flow decisions.

1.4.2 Data Transfer Instructions

- Data transfer instructions are responsible for moving data between processor registers, memory locations, and input/output interfaces.
- These instructions support a wide range of addressing modes, providing flexible access to operands.

- Common operations include loading data from memory into registers and storing register contents back into memory.
- Efficient data transfer is essential for maintaining high execution performance and ensuring that execution units receive operands without delay.
- These instructions do not typically modify arithmetic flags.

1.4.3 Control Transfer Instructions

- Control transfer instructions alter the normal sequential flow of program execution in the Intel Core i5-6300U processor.
- These instructions include conditional and unconditional jumps, procedure calls, and return instructions.
- Control transfer instructions enable the implementation of loops, conditional branches, and function calls in software programs.
- Modern Intel processors employ branch prediction mechanisms to reduce the performance impact of control flow changes.
- These instructions rely heavily on condition flags to determine execution paths.

1.4.4 Bit Manipulation Instructions

- Bit manipulation instructions allow direct access to individual bits or groups of bits within registers or memory locations.
- Operations such as bit testing, setting, clearing, and complementing are supported without affecting unrelated bits.
- These instructions are widely used in low-level system software, cryptographic algorithms, and hardware control routines.
- Bit manipulation instructions often update status flags, enabling efficient bit-based decision making.
- They contribute to compact and optimized low-level code.

1.4.5 String Instructions

- String instructions are designed to operate on sequences of data stored in contiguous memory locations.
- These instructions support operations such as moving, comparing, scanning, and initializing blocks of memory.

- The Intel Core i5-6300U provides hardware support for automatic pointer updates during string instruction execution.
- String instructions reduce software overhead by minimizing the need for explicit loop constructs.
- They are commonly used in memory management and system-level software.

1.4.6 Floating-Point Instructions

- Floating-point instructions enable arithmetic operations on real numbers with fractional components.
- These instructions are executed using the processor's Floating Point Unit (FPU), which is optimized for high-precision computation.
- The Intel64 / IA-32 architecture supports operations such as floating-point addition, subtraction, multiplication, and division.
- Floating-point instructions are essential for scientific, engineering, and multimedia applications.
- The Intel Core i5-6300U also supports advanced floating-point extensions that improve performance in computation-intensive workloads.

1.5 Micro-Architecture and Memory Organization

1.5.1 Micro-Architecture

The Intel Core i5-6300U processor used in the HP EliteBook 840 G3 is based on Intel's Skylake micro-architecture. It employs an out-of-order execution model, allowing instructions to be executed in a different order than they appear in the program to improve performance. The processor features a deep instruction pipeline, multiple execution units, and advanced branch prediction mechanisms. These design choices enable efficient utilization of hardware resources while maintaining high instruction throughput.

1.5.2 ALU Operations

The Arithmetic Logic Unit (ALU) in the Intel Core i5-6300U performs a wide range of integer arithmetic and logical operations required by the instruction set architecture. These operations include addition, subtraction, multiplication, division, and logical functions such as AND, OR, XOR, and shifting. The processor also supports parallel execution of operations through multiple execution units, improving overall computational performance.

1.5.3 Cache Memory and Memory Interfacing

The Intel Core i5-6300U incorporates a multi-level cache hierarchy to reduce memory access latency and improve system performance. It includes separate Level 1 instruction and data caches, a Level 2 cache for each core, and a shared Level 3 cache accessible by all cores. The processor interfaces with external main memory using high-speed DDR3 or DDR4 memory channels, enabling efficient data transfer between the processor and system memory.

1.5.4 Timing Related to Memory

Memory access timing in the Intel Core i5-6300U is optimized using its hierarchical cache structure. Frequently accessed data is stored in faster cache levels, significantly reducing access time compared to main memory. The processor also employs techniques such as prefetching and cache coherence mechanisms to minimize memory latency and ensure data consistency. These optimizations contribute to improved overall system responsiveness and performance.

2 ARM Cortex-M23 Processor

2.1 Overview

The ARM Cortex-M23 is a 32-bit microcontroller-class processor based on the ARMv8-M baseline architecture. It is specifically designed for ultra-low-power and cost-sensitive embedded applications such as Internet of Things (IoT) devices, sensors, and control systems. The processor employs a simple in-order execution model with a short pipeline, enabling predictable timing behavior and efficient energy usage. With optional security features such as ARM TrustZone technology, the Cortex-M23 provides hardware-based isolation between secure and non-secure execution environments, making it suitable for embedded systems that require both low power consumption and enhanced security.

2.2 Instruction Set Architecture

The ARM Cortex-M23 processor is based on the ARMv8-M baseline instruction set architecture, which follows the Reduced Instruction Set Computing (RISC) philosophy. This architecture is designed to achieve simplicity, energy efficiency, and predictable execution behavior. The Cortex-M23 is a 32-bit processor and is primarily targeted at low-power embedded and Internet of Things (IoT) applications. Unlike CISC architectures, ARMv8-M emphasizes a smaller and more efficient instruction set, enabling faster instruction decoding and reduced hardware complexity.

2.3 Instruction Format

The instruction format used by the Cortex-M23 is simple and largely fixed in length. It supports the Thumb-2 instruction set, which consists of both 16-bit and 32-bit instruction encodings. This mixed-length format improves code density while maintaining reasonable performance. Each instruction typically contains fields for the opcode, register operands, and immediate values. The simplified instruction format reduces decoding overhead and contributes to lower power consumption and deterministic execution timing.

2.4 Basic Instruction Classes Available

The Cortex-M23 supports essential instruction classes required for embedded system operation. These include arithmetic and logical instructions for basic computation, load and store instructions for memory access, and branch instructions for control flow. Since ARM follows a load-store architecture, only load and store instructions are allowed to access memory, while all other operations are performed using registers. In addition, the processor supports system control instructions for exception handling and low-power operation, which are critical for real-time embedded applications.

2.5 Micro-Architecture

The micro-architecture of the ARM Cortex-M23 is optimized for simplicity and energy efficiency. It uses an in-order execution model with a two-stage pipeline consisting of instruction fetch and execution stages. This straightforward pipeline design reduces hardware complexity and ensures predictable instruction timing. The Cortex-M23 is designed to deliver deterministic performance, which is essential for real-time and safety-critical systems.

2.6 ALU Operations

The Arithmetic Logic Unit (ALU) in the Cortex-M23 performs basic integer arithmetic and logical operations such as addition, subtraction, bitwise AND, OR, XOR, and shift operations. These operations are optimized for low power consumption rather than high computational throughput. The ALU supports single-cycle execution for most operations, enabling efficient processing in resource-constrained environments.

2.7 Cache Memory and Memory Interfacing

The ARM Cortex-M23 typically does not include an internal cache memory, as it is intended for systems with small memory footprints. Instead, it relies on tightly coupled memory and direct access to on-chip Flash and SRAM. The processor interfaces with memory through a simple bus structure, allowing fast and predictable access to peripherals and memory-mapped I/O devices. Optional Memory Protection Units (MPUs) may be included to control memory access and enhance system security.

2.8 Timing Related to Memory

Memory access timing in the Cortex-M23 is highly deterministic, which is a key requirement for real-time embedded systems. Since the processor usually operates without cache memory, there are minimal variations in memory access latency. The architecture supports predictable interrupt response times and includes low-power instructions such as Wait For Interrupt (WFI) and Wait For Event (WFE). These features ensure reliable timing behavior and efficient power management in embedded applications.

3 Comparison of Intel Core i5-6300U and ARM Cortex-M23

Feature	**Intel Core i5-6300U**	**ARM Cortex-M23**
Architecture	CISC (x86-64)	RISC (ARMv8-M)
Instruction Set Width	64-bit	32-bit
Core Count	Dual Core	Single Core
Thread Support	Supports Hyper-Threading (4 threads)	No multithreading support
Clock Speed	Up to 3.0 GHz (Turbo Boost)	Typically tens of MHz
Pipeline Type	Deep, out-of-order pipeline	Simple in-order pipeline
Execution Model	Out-of-order execution	In-order execution
Instruction Format	Variable-length instructions	Fixed 16-bit and 32-bit instructions
Cache Memory	Multi-level cache (L1, L2, L3)	Typically no cache memory
Memory Interface	DDR3/DDR4 main memory interface	Direct access to on-chip Flash and SRAM
Memory Access Latency	Optimized using cache hierarchy	Deterministic and fixed latency
Floating Point Support	Dedicated FPU with SIMD support	Optional or limited floating-point support
SIMD / Vector Support	SSE, AVX, AVX2 supported	Not supported
Power Consumption	Approximately 15 W TDP	Micro- to milli-watt range
Thermal Design	Requires active cooling	Passive cooling sufficient
Operating System Support	Windows, Linux, Virtualization platforms	Bare-metal or RTOS (e.g., FreeRTOS)
Security Features	Intel hardware security features	TrustZone and MPU support

Typical Use Case	Laptops, desktops, general-purpose computing	Embedded systems and IoT devices

Conclusion

This report presented a detailed study and comparison of two processors with fundamentally different design goals: the Intel Core i5-6300U used in the HP EliteBook 840 G3 laptop and the ARM Cortex-M23 processor commonly employed in embedded systems. Through the analysis of instruction set architecture, instruction formats, basic instruction classes, micro-architecture, ALU operations, cache organization, memory interfacing, and memory timing behavior, it is evident that processor design is highly influenced by the intended application domain.

The Intel Core i5-6300U, based on the Intel64 / IA-32 CISC architecture, is optimized for high-performance, general-purpose computing. Its support for a rich and complex instruction set, out-of-order execution, multi-level cache hierarchy, and high clock frequencies enables efficient execution of demanding workloads such as multitasking, multimedia processing, and software development. These features, however, result in higher power consumption and greater hardware complexity, which are acceptable trade-offs in laptop and desktop environments where performance is a priority.

In contrast, the ARM Cortex-M23 processor follows the RISC-based ARMv8-M architecture and is specifically designed for low-power, cost-sensitive, and real-time embedded applications. Its simple instruction set, fixed instruction formats, in-order execution model, and deterministic memory access behavior allow predictable timing and extremely low power consumption. The absence of complex cache hierarchies and advanced execution mechanisms makes the Cortex-M23 ideal for battery-powered and safety-critical systems, where energy efficiency and reliability are more important than raw computational performance.

Overall, this comparison clearly demonstrates that there is no single processor architecture that is universally optimal for all applications. Instead, the choice of a processor depends on specific system requirements such as performance, power consumption, timing predictability, and application complexity. The Intel Core i5-6300U and ARM Cortex-M23 represent two distinct design philosophies, each excelling in its respective domain, and together they highlight the importance of architectural trade-offs in modern processor design.