Comparative study on Hardwired and Microprogrammed control logic

PROJECT REPORT

submitted by

DEVIKA P SAJITH
ASI23CA028
Fathima P Ajvad
ASI23CA029
Fayiza K H
ASI23CA030
Gokul G Nair
ASI23CA031
Gokul P
ASI23CA032
Govind Lal TL
ASI23CA033

Under the guidance of

PROF. ASHA ROSE THOMAS

To

APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree
of
Bachelor of Technology In Computer Science and Engineering(Artificial Intelligence)

Department of Computer Science and Engineering(AI)

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY
KALADY

APRIL 2025

Comparative study on Hardwired and Microprogrammed control logic

PROJECT REPORT

submitted by

DEVIKA P SAJITH
ASI23CA028
Fathima P Ajvad
ASI23CA029
Fayiza K H
ASI23CA030
Gokul G Nair
ASI23CA031
Gokul P
ASI23CA032
Govind Lal TL
ASI23CA033

Under the guidance of

PROF. ASHA ROSE THOMAS

To

APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree
of
Bachelor of Technology In Computer Science and Engineering(Artificial Intelligence)

Department of Computer Science and Engineering(Artificial Intelligence))

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY
KALADY
APRIL 2025

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY, KALADY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)

CERTIFICATE

Certified that this is a bonafide record of the project entitled

Comparative study on Hardwired and Microprogrammed control logic

Submitted by

DEVIKA P SAJITH
ASI23CA028
Fathima P Ajvad
ASI23CA029
Fayiza K H
ASI23CA030
Gokul G Nair
ASI23CA031
Gokul P
ASI23CA032
Govind Lal TL
ASI23CA033

during the year 2024-25 in partial fulfillment of the requirement for the award of the degree of
Bachelor of Technology in Computer Science and Engineering(AI)

ACKNOWLEDGMENT

At the very outset, we would like to give the first honours to God, who gave us the wisdom andknowledge to complete this project.

Our extreme thanks to **Dr M S Murali.**, Principal for providing the necessary facilities for the completion of this project in our college.

We sincerely extend our thanks to **Prof. P V Rajaraman**, Chief Technology Officer for allthe help, motivation and guidance throughout the completion of this project.

We also like to extend our gratitude to **Dr.Sarika S .** AI&DS for all the help, motivationand guidance throughout the project.

We wish to extend our sincere thanks to the course coordinator **Prof. Asha Rose Thomas** for her valuable guidance and support throughout the project.

.

We also thank our Parents, friends and all well-wishers who supported us directly or indirectlyduring this project.

VISION AND MISSION OF THE DEPARTMENT

VISION

To be in the frontier of AI technology through quality of education, collaborative research and produce globally competitive, industry ready engineers with social commitment.

MISSION

- M1: Achieve excellence in the educational experience, fostering collaborative research through state-of-the-art infrastructure and innovative elements.
- M2: Establish industry collaboration to address interdisciplinary challenges across diverse applications.
- M3: Inspire students to develop into ethical, Innovative and entrepreneurial leaders through a socially-centered program.

ABSTRACT

This project gives a comparative analysis of Hardwired and Microprogrammed Control Logic, comparing their architectural aspects, strengths, and implications on real-world performance. Hardwired control, a characteristic feature of RISC architectures, uses specialized circuits to execute instructions efficiently and quickly, while microprogrammed control, used in CISC architectures, uses a control memory to provide more flexibility and ease of change.

For complementing theoretical analysis, benchmark tests were performed on two real-world processors: the Intel Core i7-9700F (microprogrammed) and the Snapdragon 865 (hardwired). The outcome reveals that although the i7-9700F produced greater absolute performance, this benefit was partly a result of its much higher clock speed and power utilization. Normalized for performance per watt, the Snapdragon 865 also exhibited better energy efficiency, affirming the pragmatic virtues of hardwired control in mobile and low-power applications.

This work emphasizes the inherent trade-offs among speed, efficiency, and flexibility in control unit design. Hardwired control is power-efficient and real-time optimized, thus well-suited for mobile and embedded systems, whereas microprogrammed control is more adaptable and software-compatible, and hence suitable for general-purpose and high-performance computing. The results emphasize the need to choose the right control logic depending on target applications, considering performance, power efficiency, and architectural requirements.

CONTENT

Introduction

Control unit is one of the most basic parts of a processor that guides the data flow and instructions within a computer system. It produces control signals that determine how various components of the CPU communicate to carry out instructions. The design and efficiency of the control unit play a major role in how efficiently a processor functions, consumes power, and is flexible.

There are two main approaches to designing a control unit: hardwired control and microprogrammed control. Both methods have unique advantages and trade-offs, and each is appropriate for a different kind of computing system.

Hardwired Control Logic

Fixed electronic circuits, usually constructed using combinational logic and finite state machines, are employed in hardwired control units to produce control signals. The control signals are based on the instruction opcode and are directly executed by hardware. This method is very fast and efficient since it does not require extra memory lookups. It has a significant drawback, though: inflexibility. Once a hardwired control unit is implemented, adding new instructions or optimizing performance necessitates redesigning the entire circuit.

Microprogrammed Control Logic

Microprogrammed control logic, on the other hand, employs a control memory where a series of microinstructions dictates how each machine instruction is to be carried out. This approach to design adds a level of abstraction so that more flexibility and easier modification of instruction sets are possible without changing the physical hardware. Microprogrammed control is extensively employed in complex instruction set computers (CISC) because it can efficiently manage complex instruction decoding. Yet this flexibility is paid for in terms of performance since fetching and decoding microinstructions adds extra execution time.

Objective of the Study

This comparative study seeks to compare the pros and cons and real-world uses of both control logic methods. Through the examination of performance, design complexity, flexibility, cost, and power efficiency, we are able to look into how they affect processor design. Furthermore, this research delves into the historical development of control logic, examples from the real world of processors applying each approach, and how modern developments influence determining the applicability of hardwired and microprogrammed control in contemporary computing.

Knowledge of these control mechanisms is necessary to understand CPU architecture and maximize processor performance for a particular application. Though hardwired control is preferred in high-speed and low-energy designs (e.g., RISC processors), microprogrammed control is still beneficial in systems where frequent updating and large instruction sets are necessary. From this analysis, we will emphasize how each method aids contemporary computer architecture and identify which technique is more common in current processors.

Hardwired Control Unit

Introduction

A CPU's control unit controls everything through generating control signals that regulate data flow and carrying out instructions. Hardwired control unit is a control unit that is implemented using fixed electronic circuits, typically described as combinational and sequential logic. Compared with microprogrammed control units using a control memory to store microinstructions, hardwired control units calculate control signals directly using fixed logic circuits.

This method is typically used in *Reduced Instruction Set Computer (RISC)* processors, where instruction sets are more straightforward, using fewer control signals. Although hardwired control is very fast and efficient, it is not very flexible, and it is hard to modify and change.

2. **Structuring of a Hardwired Control Unit**

A hardwired control unit involves a number of interconnected units, and each one has a vital function in decoding instruction and its execution.

A. Core ingredients

Instruction Register (IR)	Holds the instruction being executed. The opcode is read out and sent to the control logic.
Decoder	Translates the opcode into a binary control signal to call particular execution paths.

Finite State Machine (FSM)	Manages the order of instructions so that every instruction is executed in the proper execution cycle.
Combinational Logic Circuits	Derive control signals from execution state and opcode using logic gates
Clock Generator	Produces timing signals to synchronize code generation and instruction execution.
Control Signal Generator	Produces actual signals utilized to drive the CPU blocks such as ALU, registers, and memory.

3. **Operation Principle of a Hardwired Control Unit**

Step 1: Extracting the Instruction
The Program Counter (PC) holds the address of the next instruction to be executed.
The Instruction Register (IR) loads the instruction from memory.
The instruction opcode (operation code) is read and forwarded to the decoder to be interpreted.

Step 2: Decoding the Instruction
The Decoder takes the opcode and identifies the instruction type as arithmetic, logical, or load/store.
Based on the decoded instruction, the finite state machine (FSM) determines the sequence of operations required.

Step 3: Generation of Control Signal
The Combinational Logic Circuits produce control signals as per the instruction decoded.
These control signals energize different CPU elements, such as:
The ALU for arithmetic and logic operations.
The Register File for reading and writing data.
The Memory for load and store operations.

Step 4: Executing the Instruction

The information required is retrieved from registers or memory.
The operation is performed by the ALU or the relevant CPU unit.
The result is stored back in a register or memory when needed.

Step 5: Proceed to the Next Instruction
Program Counter is incremented to point to the next instruction.
The procedure is repeated for the next instruction.

4. Hardwired Control Units Benefits

High Speed
Hardwired control units produce control signals in a direct manner by logic circuits without the use of microinstruction lookups.
This renders them far quicker than microprogrammed control units.

Power Efficiency
Since hardwired logic does not involve memory access to generate control signals, it consumes less power and is therefore employed in embedded systems and real-time systems.

Streamlined for Simplicity
Hardwired control is especially well adapted to RISC architectures, which have simple and fixed-length instruction formats, reducing the complexity of the control logic.

5. **Disadvantages of Hardwired Control Unit**

Hard to Change Any modification to the instruction set necessitates redesign of the whole control circuit, and therefore upgrades are costly and time-consuming.

Complexity for Large Instruction Sets

In Complex Instruction Set Computers (CISC), the large number of instructions necessitates more complicated control logic, and hence hardwired control becomes uneconomical.

Limited Scalability

With increasing complexity in CPUs, adding new features like virtualization, parallelism, or security features is not easily possible with hardwired control.

6. **Comparison to Microprogrammed Control Units**

Speed - *Hardwired control is faster because it produces control signals directly, while microprogrammed control is slower because it requires memory access for microinstructions.*

Flexibility - *Hardwired control is not flexible as modifications entail redesign of hardware, while microprogrammed control is flexible and modifications can be made by rewriting microcode.*

Complexity - *Hardwired control is complicated for big instruction sets, while microprogrammed control is simpler to design and to alter.*

Application - *Hardwired control is applied in RISC processors and high-speed calculation, while microprogrammed control is applied in CISC processors and systems that require frequent updating.*

Power Consumption - Hardwired control uses less power than microprogrammed control since it has fewer memory accesses.

7. **Real-World Applications**

A. **Hardwired Control** in RISC Processors MIPS Processors use a hardwired control unit to allow quick execution of simple instructions. ARM processors used in smartphones and embedded systems are based on hardwired control for efficiency.

B. **Special Purpose Hardware-**Initiated Hardwired Control Digital Signal Processors (DSPs) are used in audio processing, telecommunications, and radar systems. Supercomputers employ hardwired control for particular operations where maximum speed is needed.

C. **Hardwired Control in Early Computers IBM System/360** initially applied hardwiring control before the transition into microprogramming by later models. 8. Conclusion The hardwired control unit is an extremely efficient method to the implementation of CPU control logic, offering speed, power efficiency, and simplicity in RISC-based and real-time systems. However, its lack of flexibility and difficulty of modification render it less attractive to CISC processors and systems that require frequent updates. With enhanced processor design, modern high-speed CPUs employ a hybrid scheme, which combines the speed of hardwired control for simple instructions and the flexibility of microprogrammed control for complicated operations.

Microprogrammed Control Unit

1. Introduction

The control unit of a CPU is responsible for generating control signals that control instruction execution. A microprogrammed control unit uses a control memory and a sequence of microinstructions that define the control signals for each machine instruction. Unlike hardwired control units generating signals from static logic circuits, microprogrammed control units read and execute microinstructions from control memory with the benefit of increased flexibility.

Microprogrammed control is used in ***Complex Instruction Set Computer (CISC)*** processors whose instructions are longer than one cycle and have complex control sequences. Microprogrammed control is less rigid and more easily changed but slower compared to hardwired control due to memory access overhead.

2. **Structure of a Microprogrammed Control Unit**

A control unit that is microprogrammed contains a number of parts that cooperate to interpret instructions and create control signals.

A. Principal Components

Instruction Register (IR)	Holds the instruction being executed. The opcode is read out and sent to the control logic.
Control Memory	Holds microinstructions that specify the sequence of operations for every machine instruction
Microinstruction Decoder	Interprets the microinstruction from control memory and outputs control signals according to it.

Address Sequencer	Determines the address of the next microinstruction to be run, depending on the type of instruction and the status of execution
Clock Generator	Produces timing signals to synchronize code generation and instruction execution.
Control Signal Generator	Produces actual signals utilized to drive the CPU blocks such as ALU, registers, and memory.

3. Working Mechanism of a Microprogrammed Control Unit

Step 1: Retrieving the Instruction

The Program Counter (PC) contains the address of the next instruction.
The Instruction Register (IR) gets the instruction from memory.
The opcode is fetched and transmitted to the control memory.

Step 2: Retrieving Microinstructions

The opcode is an address used to read the associated microinstruction from control memory.
The first microinstruction of the stream is found in the Microinstruction Decoder.

Step 3: Generation of Control Signal

The Microinstruction Decoder translates the microinstruction into control signals.
These signals drive various components of the CPU, including:
The ALU that processes arithmetic and logical operations.
The Register File for data transfer.
The Memory for load and store operations.

Step 4: Executing the Microinstructions

All the microinstructions execute sequentially, utilizing different units.

The Address Sequencer chooses the next microinstruction address. Should the instruction need extra microinstructions, this is done until the end of the instruction.

Step 5: Proceeding to the Next Instruction
After all the microinstructions of the current instruction are completed, the Program Counter (PC) is incremented.
The procedure is repeated for the next instruction.

4. Microprogrammed Control Unit Advantages

Adaptive Design
Microprogrammed control units permit adjustments through reprogramming control memory rather than redefining hardware.

Reduced Control Logic
Use of microinstructions eliminates the complexity involved in designing the control logic and complicated instruction sets are easier to implement.

Easier Support for Complex Instructions
Microprogramming is beneficial to CISC architectures because they possess multi-step execution sequences.

5. Microprogrammed Control Unit Disadvantages

Slower Execution

Microprogrammed control introduces delays in memory access because control signals need to be read from control memory.

Increased Power Consumption

More power is needed to retrieve and interpret microinstructions than for hardwired control.

Needs More Storage

Control memory takes up extra space, adding to CPU cost and complexity.

6. Comparison of Hardwired Control Units

Speed - *Microprogrammed control is slower due to memory access, whereas hardwired control is faster with direct logic circuits.*

Flexibility - *Microprogrammed control is flexible and can be changed, while hardwired control requires redesign of hardware.*

Complexity - *Microprogrammed control is less complicated in design, whereas hardwired control is complex for large instruction sets.*

Application - *Microprogrammed control is applied in CISC processors, whereas hardwired control is the choice for RISC processors and real-time systems.*

Power Consumption - *Microprogrammed control is more power-hungry because of extra memory accesses, whereas hardwired control is less power-hungry*.
7. **Real-World Applications**
A**. Microprogrammed Control in CISC Processors Intel x86 CPUs** use microprogramming to enable intricate instructions. IBM Mainframes utilize microprogrammed control in large computing.

B**. Microprogrammed Control in Programmable Architectures Digital Signal Processors (DSPs)** can implement programmable instruction execution using microprogramming.
C**. Microprogrammed Control in Early Computers IBM System/360** utilized microprogramming to accommodate a variety of instruction formats. 8. Conclusion The microprogrammed

control unit is a flexible and modular means of controlling the CPU logic, which should be implemented in CISC architectures and systems with high update frequencies. Due to its slower rate of execution and higher power consumption, though, it cannot be used for high-performance purposes. Modern processors are usually a combination of hardwired and microprogrammed control, using microprogramming for complex operations and hardwired control for performance-critical instructions.

Theoretical Comparison of Hardwired vs. Microprogrammed Control

1. Control Unit Overview

The **control unit** of a processor is responsible for generating signals that control data flow, instruction execution, and memory operations. There are two primary types of control units:

- **Hardwired Control Unit** – Uses fixed logic circuits to generate control signals.
- **Microprogrammed Control Unit** – Uses a control memory (microcode) to store predefined sequences of control signals.

2. Key Differences

Feature	Hardwired Control	Microprogrammed Control
Instruction Handling	Uses combinational logic circuits to generate control signals directly	Uses microinstructions stored in ROM or RAM to generate control signals
Execution Speed	Faster because control signals are generated instantly	Slightly slower due to microinstruction decoding overhead
Flexibility	Limited – Changes require hardware modifications	Highly flexible – New instructions can be added via microcode updates
Complexity	Simple for RISC architectures but complex for CISC	Easier to implement for CISC architectures
Power Efficiency	More efficient due to direct	Less efficient due to additional memory

	execution	access for microinstructions
Best Suited For	RISC processors, mobile devices, real-time systems	CISC processors, general-purpose computing, servers

3. Theoretical Performance Expectations

- **Hardwired control is expected to be faster** because it directly generates control signals without an extra decoding step.
- **Microprogrammed control is expected to be more flexible** because it allows updates and complex instruction handling.
- **Hardwired control should be more power-efficient**, making it better suited for mobile and embedded applications.
- **Microprogrammed control should perform better on complex workloads**, making it ideal for general-purpose computing.

Comparative Study of Hardwired vs. Microprogrammed Control Using Real-World Benchmarks

To contrast these two control logic methods, we examine two actual processors:

- Intel Core i7-9700F (CISC, Microprogrammed Control, Desktop-class CPU)
- Qualcomm Snapdragon 865 (RISC, Hardwired Control, Mobile-class CPU)

We will evaluate them on the basis of performance (**Geekbench** scores), power usage, and efficiency to find out the trade-offs between microprogrammed and hardwired control in practical computing scenarios.

2. Processor Specifications

CPU	***Intel Core i7-9700F***	***Snapdragon 865***
Control Unit Type	***Microprogrammed***	***Hardwired***
Architecture	***x86 (CISC)***	***ARM (RISC)***
Cores/Threads	***8C/8T***	***8C (1 Prime + 3 Performance + 4 Efficiency)***
Base/Turbo Clock	***3.0 GHz / 4.7 GHz***	***2.84 GHz (Prime Core)***
Manufacturing Process	***14nm***	***7nm***
TDP (Power Consumption)	***65W***	***7W***

3. Benchmark Performance (Geekbench Scores)

We tested both processors with **Geekbench** to assess their single-core and multi-core performance under actual circumstances.

Processor	**Control Unit Type**	**Single-Core Score**	**Multi-Core Score**
Intel Core i7-9700F	Microprogrammed	1477	5191
Snapdragon 865	Hardwired	1150	3035

Key Takeaways
- Intel Core i7-9700F (CISC, Microprogrammed) is faster than Snapdragon 865 in single-core as well as multi-core performance. (I7 Has a Higher clock speed, Better cooling and uses way more power)

- Snapdragon 865 (RISC, Hardwired) is behind in raw performance but consumes a fraction of the power.

4. Power Efficiency: Performance per Watt

Efficiency is measured by comparing performance in relation to power consumption through performance-per-watt measurements.

Metric	**Intel Core i7-9700F (Microprogrammed)**	**Snapdragon 865 (Hardwired)**
Single-Core Performance per Watt (SC/TDP)	22.7 (1477 ÷ 65W)	164.3 (1150 ÷ 7W)
Multi-Core Performance per Watt (MC/TDP)	79.8 (5191 ÷ 65W)	433.6 (3035 ÷ 7W)

Key Takeaways
- Hardwired control (Snapdragon 865) is 7.2× more power-efficient in single-core performance per watt.
- Multi-core efficiency is even more extreme, with Snapdragon 865 being 5.4× more efficient.
- This highlights why RISC-based processors with hardwired control are dominant in mobile and battery-powered environments.

5. Analysis of Control Unit Design and Performance

Feature	**Hardwired Control (Snapdragon 865, RISC)**	**Microprogrammed Control (Intel i7-9700F, CISC)**
Speed	Fast execution for simple instructions	Extra decoding step adds latency
Flexibility	Fixed logic; changes require redesign	Microcode updates allow easier modification
Power Efficiency	High – Optimized for mobile efficiency	Lower – Consumes more power for complex tasks
Execution Model	Instructions controlled by dedicated logic circuits	Instructions controlled via microcode lookups
Instruction Complexity	Simple, fixed-length instructions	Variable-length, complex instructions
Best Use	Mobile devices,	Desktops, servers,

Cases	embedded systems, real-time applications	high-performance computing

Why RISC Uses Hardwired Control
- As RISC processors contain fewer and simpler instructions, they do not need microcode and can effectively utilize hardwired logic.
- This allows for quicker instruction execution and reduced power consumption, which is essential for battery-powered devices.

Why CISC Uses Microprogrammed Control
- CISC designs contain big, complicated instruction sets, and thus microcode is needed for effective instruction processing.
- Microprogramming simplifies the addition of new instructions, which is useful for general-purpose computing and software compatibility.

6. Real-World Impact on Performance and Use Cases

Scenario 1: General-Purpose Computing (Desktops, Workstations, Laptops)
- The Intel Core i7-9700F is optimized for high-end workloads, such as gaming, video editing, and multi-threaded loads.
- Its flexibility and capability to process complex instructions make it better for desktop workloads despite greater power usage.

Scenario 2: Mobile and Embedded Computing (Smartphones, IoT, Real-Time Processing)
- Snapdragon 865 is designed for battery-constrained devices, optimizing between performance and energy efficiency using hardwired control logic.
- It is suited for smartphones, tablets, and embedded systems where power limitations are of utmost importance.

Scenario 3: Cloud Computing and Servers
- Microprogrammed control is prevalent in x86 server processors because of its adaptability in supporting a large set of instructions.
- But RISC-based processors (such as ARM-based AWS Graviton chips) are coming into vogue in power-efficient cloud computing.

7. Conclusion

Hardwired Control (Snapdragon 865) - Pros and Cons
- Power-efficient – Suitable for mobile and embedded systems
- Simple instructions execute faster
- Reduced cost in power-limited environments
- Less adaptable – Changes need redesign in hardware
- Not the best for complex instruction sets

Microprogrammed Control (Intel Core i7-9700F) - Advantages and Disadvantages
- Greater raw performance – Better suited for high-complexity computing
- Flexible – Instruction set modifications possible due to microcode updates
- More apt for complex and variable-length instructions
- Greater power consumption
- Slightly slower owing to overhead of microinstruction decoding

Conclusion

This comparative study of **hardwired vs. microprogrammed control logic** provides key insights into the trade-offs between speed, flexibility, power efficiency, and complexity in modern processor design. By examining both **theoretical differences** and **real-world benchmarks**, we can make the following observations:

1. Theoretical Perspective

- **Hardwired control** is optimized for speed and power efficiency by using dedicated circuits to directly generate control signals. This approach works well for **RISC architectures**, where instructions are simple and uniform, leading to predictable execution times.
- **Microprogrammed control**, on the other hand, relies on a control memory that stores microinstructions, making it more **flexible and easier to modify**. This is advantageous for **CISC architectures**, where instructions are more complex and variable in length.

The key takeaway from a theoretical standpoint is that **hardwired control sacrifices flexibility for efficiency**, while **microprogrammed control sacrifices efficiency for flexibility**.

---

2. Real-World Performance and Efficiency

To validate these theoretical differences, we analyzed two real-world processors:

Processor	**Control Unit Type**	**Clock Speed (Max)**	Single-Core Score	**Multi-Core Score**
Intel Core i7-9700F	**Microprogrammed**	**4.7 GHz**	**1477**	**5191**
Snapdra	**Hardwired**	**2.8**	**11**	**30**

gon 865		**4 GHz**	**50**	**35**

The **Intel Core i7-9700F (Microprogrammed, CISC)** achieved higher **absolute performance** in both **single-core and multi-core benchmarks**. However, this raw performance advantage must be considered in context:

- The i7-9700F operates at **a much higher clock speed** (up to 4.7 GHz) compared to the Snapdragon 865 (2.84 GHz).
- When **normalized for clock speed**, the performance gap **shrinks significantly**, suggesting that the i7's advantage is **partially due to higher clock frequency** rather than just architectural superiority.
- The Snapdragon 865, despite its lower raw performance, is **far more power-efficient**, consuming only **7W** compared to **65W** for the i7-9700F.

To measure efficiency, we calculate **performance per watt** (Geekbench score per unit power consumption):

Metric	**Intel Core i7-9700F (Microprogrammed)**	**Snapdragon 865 (Hardwired)**
Single-Core Performance per Watt (SC ÷ TDP)	**22.7** (1477 ÷ 65W)	**164.3** (1150 ÷ 7W)
Multi-Core Performance per Watt (MC ÷ TDP)	**79.8** (5191 ÷ 65W)	**433.6** (3035 ÷ 7W)

Key Observations:

- The Snapdragon 865 is **over 7 times more power-efficient per watt** in single-core performance.
- In multi-core workloads, the Snapdragon 865 is still **5.4**

times more efficient than the i7-9700F.
- This highlights **why mobile processors favor hardwired control**—they deliver **more performance per watt**, making them ideal for battery-powered environments.

---

3. Trade-Offs and Design Considerations

Feature	***Hardwired Control (Snapdragon 865, RISC)***	***Microprogrammed Control (Intel i7-9700F, CISC)***
Speed	Faster execution for simple instructions	Additional decoding step adds latency
Flexibility	Fixed logic; changes require hardware redesign	Microcode updates allow easier modification
Power Efficiency	**Higher** – Optimized for mobile efficiency	**Lower** – Consumes more power for complex tasks
Execution Model	Instructions controlled by dedicated logic circuits	Instructions controlled via microcode lookups
Instruction Complexity	Simple, fixed-length instructions	Variable-length, complex instructions
Best Use Cases	**Mobile devices, embedded systems, real-time applications**	**Desktops, servers, high-performance computing**

Why RISC Uses Hardwired Control

- RISC processors have **simpler, fixed-length instructions**, which means **they don't need microcode**.
- This enables **fast, efficient execution**, which is **critical for battery-powered and low-thermal environments** like smartphones and embedded systems.

Why CISC Uses Microprogrammed Control

- CISC architectures have **large, complex instruction sets**, requiring **microcode for efficient instruction handling**.
- Microprogramming makes it **easier to add new instructions**, ensuring **backward compatibility and better software support**.

4. Real-World Applications

- **General-Purpose Computing (Desktops, Workstations, Laptops):**
  The **Intel Core i7-9700F** is optimized for **high-performance applications**, such as gaming, video editing, and multi-threaded workloads. The flexibility of **microprogrammed control** allows it to handle **a wide variety of software efficiently**.
- **Mobile and Embedded Computing (Smartphones, IoT, Real-Time Processing):**
  The **Snapdragon 865** is designed for **power efficiency** and **real-time responsiveness**. Its **hardwired control logic** ensures **low-latency execution** with **minimal power consumption**, making it ideal for **mobile devices and energy-efficient applications**.
- **Cloud Computing and Data Centers:**
  - Traditional **x86-based servers** often rely on **microprogrammed control** for maximum **compatibility and instruction set richness**.
  - However, **power-efficient ARM-based processors (like AWS Graviton) are gaining popularity**, leveraging **hardwired control for energy savings in**

large-scale cloud infrastructure.

5. Final Verdict

The **choice between hardwired and microprogrammed control logic** is not about which is strictly better—it depends on the **application and design goals**.

- **Hardwired control excels in power efficiency, low-latency execution, and high-speed instruction processing.** It is best suited for **mobile, embedded, and real-time computing environments** where energy constraints are critical.
- **Microprogrammed control is superior in flexibility, supporting complex and evolving instruction sets.** It is ideal for **general-purpose and high-performance computing**, where instruction handling and software compatibility are more important than raw efficiency.

While the **Intel Core i7-9700F achieved higher raw performance**, a significant portion of its advantage comes from **higher clock speeds and power consumption** rather than inherent superiority of microprogramming. When considering **performance-per-watt**, the **Snapdragon 865 demonstrates how hardwired control can be vastly more efficient**, making it the preferred choice for **power-constrained environments**.

As computing continues to evolve, both control logic approaches will remain **critical in different domains**, shaping the future of processors across desktops, mobile devices, and cloud computing.

References

1. Benchmark results: https://drive.google.com/drive/folders/10V_PVG09u6VtPDZkP3Dwu-Jm8XCowOpg?usp=sharing

2. **M. Mano and C. Kime, "Logic and Computer Design Fundamentals," Pearson, 5th Edition, 2015.**
   - Covers the theoretical background of hardwired and microprogrammed control.
3. **J. L. Hennessy and D. A. Patterson, "Computer Architecture: A Quantitative Approach," Morgan Kaufmann, 6th Edition, 2017.**
   - Discusses RISC vs. CISC architectures and control unit implementations.
4. **D. A. Patterson and J. L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface," Morgan Kaufmann, 5th Edition, 2013.**
   - Explains microprogramming, instruction execution, and performance trade-offs.
5. **Qualcomm Snapdragon 865 Technical Specification, Qualcomm Inc.**
   - https://www.qualcomm.com/products/snapdragon-865-5g-mobile-platform
6. **Intel Core i7-9700F Processor Specification, Intel Corporation.**
   - https://ark.intel.com/content/www/us/en/ark/products/186604/intel-core-i7-9700f-processor-12m-cache-up-to-4-70-ghz.html
7. **S. Borkar, "The Future of Microprocessors," Communications of the ACM, vol. 54, no. 5, pp. 67–77, 2011.**
   - Discusses power efficiency and performance trends in processor design.
8. **D. Kanter, "A Look at ARM’s Cortex-A77 CPU Core,"**

Real World Technologies, 2019.

   - https://www.realworldtech.com/arm-cortex-a77

9. **W. Stallings, "Computer Organization and Architecture: Designing for Performance," Pearson, 10th Edition, 2016.**
   - Covers the role of microprogrammed and hardwired control in modern processors.

10. **IEEE Transactions on Computers – Research on Hardwired vs. Microprogrammed Control.**

- https://ieeexplore.ieee.org