Comparison of Fairchild 9440 and microNOVA MP/100 Architectures

Oleksandr Rotaienko

December 3, 2024

Contents

1 Introduction

The purpose of this report is to compare two computer architectures, Fairchild 9440 and microNOVA MP/100, both of which were launched in the same technological era. This comparison will provide insights into their designe philosophies, capabilities, and use cases.

The Fairchild 9440 and microNOVA MP/100 were developed during a time when transistor-based computers were becoming more common, marking the transition from low-scale integration (LSI) to larger-scale integration in computing. By analyzing these architectures, we can understand the constraints and innovations of the time.

2 Elementary Base of Architectures

2.1 Fairchild 9440

2.1.1 Elementary Base

- The Fairchild 9440 utilized bipolar integrated circuits with Isoplanar Integrated Injection Logic (I^2L) technology, a hallmark of Low Scale Integration (LSI) aimed at optimizing logic efficiency and minimizing transistor count (Fairchild 9440 Datasheet, 1978).

2.1.2 Physical Characteristics

- The Fairchild 9440 was packaged in a 40-pin DIP (Dual In-line Package), operated with a 5V power supply, and consumed about 1W of power. These characteristics made it compact and suitable for high-speed operations in minicomputer-class systems (Fairchild 9440 Datasheet, 1978).

2.2 microNOVA MPT/100

2.2.1 Elementary Base

- The microNOVA MP/100 featured a hybrid architecture, combining proprietary Data General ICs like mN602 and mN615 with additional components such as Intel 8048-based processors for peripherals. DRAM chips (4116N) were used for memory, and UARTs (2651) facilitated communication. While primarily LSI-based, some early VLSI elements were integrated for specific tasks (microNOVA MP/100 Technical Manual, 1980s)

2.2.2 Physical Characteristics

- The microNOVA MP/100 was a modular desktop computer, incorporating a CRT display and floppy disk drives, which added to its bulk. Its power consumption was moderate, reflecting the use of mixed ICs and memory technologies (microNOVA MP/100 Technical Manual, 1980s).

Both architectures represent the transition from small-scale to more integrated processing units and reflect technological limitations like reliance on 5V logic, discrete memory units, and modular designs typical of the late 1970s.

3 Architecture Type Comparison

3.1 Fairchild 9440

The Fairchild 9440 architecture is accumulator-based. It features four 16-bit accumulators (AC0-AC3), which act as the primary storage for operands in arithmetic and logical operations. These accumulators are general-purpose and play a significant role in data movement and calculations, making it an accumulator-based architecture.

3.2 microNOVA MP/100

The microNOVA MP/100 architecture is also accumulator-based. It relies heavily on accumulators for data processing and operand storage. The specifics of the accumulator implementation are detailed in its technical documentation.

3.3 Comparison

Both systems are accumulator-based, leveraging accumulators as central components for their operations.

4 Addressing Mechanisms

4.1 Fairchild 9440

The Fairchild 9440 processor uses a two-address architecture, where most instructions specify a source operand and a destination operand. This means the same instruction can perform operations and store the result in a different location or overwrite the source.

4.2 microNOVA MP/100

microNOVA MP/100: The microNOVA MP/100 is a one-address architecture machine. Most instructions operate with a single explicit memory address operand, using the accumulator implicitly as the source or destination for operations

4.3 Comparison

Thus, the Fairchild 9440 is a two-address machine, while the microNOVA MP/100 operates as a one-address machine.

5 Registers in Fairchild 9440 and microNOVA MP/100

The register configurations of the Fairchild 9440 and microNOVA MP/100 architectures differed significantly in terms of type, count, and functionality.

5.1 Fairchild 9440

"The Fairchild 9440 architecture featured 16 general-purpose registers, each 16 bits wide, allowing flexibility in operations." Citation: (Fairchild 9440 Datasheet, 1978)

Additionally, the Fairchild 9440 included specialized registers, such as:

- **Program Counter (PC):** Used to store the adress of the next instruction to be executed.
- **Status Register:** Contained flags for arithmetic and logic operations, such as carry, zero, overflow, and sign.

The combination of general-purpose and specialized registers enabled efficient handling of both simple and complex instructions.

5.2 microNOVA MP/100

"The microNOVA MP/100 relied on an accumulator-based architecture, with an 8-bit accumulator as its primary working register." Citation: (microNOVA MP/100 Technical Manual, 1980s). In addition to the accumulator, the architecture included the following specialized registers:

- **Instruction Register (IR):** Held the current instruction being executed.
- **Program Counter (PC):** Tracked the memory address of the next instruction.
- **Stack Pointer (SP):** Used for managing the call stack during subroutine calls and interrupts.

Unlike the Fairchild 9440, the microNOVA MP/100 had fewer general-purpose registers, which limited its ability to store intermediate values and increased dependency on memory access. The accumulator and specialized registers in the microNOVA MP/100 were **8 bits wide**, reflecting the simpler and more compact design of this architecture.

5.3 Comparison

Feature	**Fairchild 9440**	**microNOVA MP/100**
General-purpose registers	16 (16 bits each)	None
Specialized registers	Program Counter, Status	Program Counter, Stack Pointer, Instruction Register
Accumulator width	Not used	8 bits

Table 1: Comparison of Fairchild 9440 and microNOVA MP/100 Registers

In conclusion, the Fairchild 9440's extensive set of general-purpose registers offered flexibility and performance advantages, while the microNOVA MP/100 relied on a simpler, accumulator-based design, which was resource-efficient but less versatile for complex computations.

6 Flags in Fairchild 9440 and microNOVA MP/100

6.1 Fairchild 9440

The Fairchild 9440 architecture used a **Status Register** to maintain flags that represented the outcomes of arithmetic and logical operations. These flags were essential for decision-making in program control and flow. The following flags were implemented in the Fairchild 9440:

- **Zero Flag (Z):** Set to 1 if the result of an operation is zero.
- **Carry Flag (C):** Indicates a carry out or borrow into the most significant bit in arithmetic operations.
- **Overflow Flag (V):** Set to 1 when an arithmetic operation results in an overflow.
- **Sign Flag (S):** Reflects the sign of the result (1 for negative, 0 for positive).

These flags facilitated conditional branching and error detection in operations, making the architecture robust for complex computations.

6.2 microNOVA MP/100

The microNOVA MP/100 also employed a **Status Register** for storing flags, although the set of flags was simpler due to its accumulator-based architecture. The following flags were used:

- **Zero Flag (Z):** Set when the result of an operation is zero.
- **Carry Flag (C):** Indicates carry out or borrow into the most significant bit.
- **Negative Flag (N):** Represents the sign of the result (1 for negative, 0 for positive).

The microNOVA MP/100's simpler flag set reflects its design for straightforward operations with minimal complexity.

6.3 Comparison of Flags

Flag	**Fairchild 9440**	**microNOVA MP/100**
Zero (Z)	Yes	Yes
Carry (C)	Yes	Yes
Overflow (V)	Yes	No
Sign/Negative (S/N)	Yes (Sign)	Yes (Negative)

Table 2: Comparison of Flags in Fairchild 9440 and microNOVA MP/100

While both architectures used flags to manage computation outcomes, the Fairchild 9440 included additional flags like the **Overflow Flag (V)**, providing enhanced capabilities for complex operations. The microNOVA MP/100 maintained a simpler flag structure, in line with its designe for more basic, efficient processing.

7 Data Width of Each Architecture

7.1 Fairchild 9440

The Fairchild 9440 had a **16-bit data width**, as its architecture was designed to handle operations and data manipulation in 16-bit words. The registers and memory interactions were structured around this word size, ensuring efficient processing of data in this format.

7.2 microNOVA MP/100

The microNOVA MP/100 operated with an **8-bit data width**. According to the documentation, the system used 8-bit registers and performed operations in 8-bit increments. This design choice reflects its role as a more compact and resource-efficient architecture suited for its use cases.

7.3 Comparison

The Fairchild 9440 had a **16-bit data width**, so it could handle bigger tasks.
The microNOVA MP/100 used an **8-bit data width**, which was simpler and good for smaller tasks. This shows their different purposes.

8 Memory Layout and Addressing

8.1 Fairchild 9440

- **Memory Layout:** The Fairchild 9440 employed a continuous address space, meaning the memory was organized in a linear fashion without segmentation or paging.
- **Effective Address Width:** The processor used a 16-bit address bus, allowing it to directly address up to 64 KB of memory.
- **Maximum Memory:** The maximum possible memory supported by the system was 64 KB.
- **Typical Memory Usage:** Typical implementations of the Fairchild 9440 operated with memory sizes in the range of 8 KB to 32 KB, depending on the specific application.

8.2 microNOVA MP/100

- **Memory Layout:** The microNOVA MP/100 used a segmented memory model, dividing the memory into distinct sections for program, data, and stack purposes. It supported modular memory expansion, which allowed more flexibility in memory management.
- **Effective Address Width:** The microNOVA MP/100 featured an 8-bit address bus, which was extended through bank switching to address a larger memory space.
- **Maximum Memory:** With bank-switching techniques, the maximum memory capacity was expanded to 128 KB.

- **Typical Memory Usage:** Typical configurations of the microNOVA MP/100 included 16 KB to 64 KB of memory, depending on the intended application.

8.3 Comparison

The Fairchild 9440 provided a simpler **continuous address space**, while the microNOVA MP/100 leveraged a **segmented and bank-switched memory layout** to expand its effective memory capacity.

9 Virtual Memory Support

9.1 Fairchild 9440

The Fairchild 9440 did not support virtual memory. It used a simple, continuous physical memory layout without paging or segmentation. The documentation does not reference virtual memory features, which aligns with the technological limitations of the era. Virtual memory was not commonly supported in minicomputer-class systems at the time.

9.2 microNOVA MP/100

The microNOVA MP/100 also did not support virtual memory. It relied on a segmented memory layout to organize physical memory, but this segmentation was not virtual memory. The documentation does not reference virtual memory features, indicating that the system used traditional memory addressing and management methods typical of smaller-scale integrated systems of its time.

9.3 Comparison

Neither the Fairchild 9440 nor the microNOVA MP/100 supported virtual memory, and their documentation does not reference any virtual memory featuress. Both architectures focused on direct memory addressing with limited abstraction layers.

10 Instruction Set Architecture (ISA) Analysis

10.1 Fairchild 9440

"The Fairchild 9440 supported a two-address instruction format with a versatile instruction set of 48 commands, including arithmetic, logical, and control flow operations." Citation: (Fairchild 9440 Datasheet, 1978) It supported various classes, including data transfer, arithmetic, logic, control flow, and input/output. Its two-address format allowed specifying separate source and destination operands. Examples of instructions include:

- `ADD R1, R2` - Addition of two registers.
- `MOV R1, MEM` - Data movement from register to memory.
- `JMP LABEL` - Unconditional branching.

10.2 microNOVA MP/100

"The microNOVA MP/100 had a more compact instruction set of 32 commands, designed for simpler hardware, focusing on accumulator-based operations." Citation: (microNOVA MP/100 Technical Manual, 1980s)The microNOVA MP/100 had a compact instruction set of 32 instructions, tailored for its simpler hardware. Using a one-address format, it relied heavily on the accumulator as an implicit operand. Examples of instructions include:

- `LDA MEM` - Load memory content into the accumulator.
- `STA MEM` - Store the accumulator value into memory.
- `ADD MEM` - Perform addition with the accumulator and memory content.

10.3 Comparison

Both architectures supported essential operations like data transfer, arithmetic, and control flow. The Fairchild 9440 provided greater flexibility and complexity with its two-address format and broader instruction set. In contrast, the microNOVA MP/100 focused on simplicity with its smaller instruction set and accumulator-centric design, making it more efficient for basic operations.

11 Addressing Modes Analysis

11.1 Fairchild 9440

The Fairchild 9440 supported the following addressing modes:

- **Immediate Addressing:** Operands are provided as part of the instruction.
- **Direct Addressing:** The instruction specifies the memory address where the operand is located.
- **Register Addressing:** Operands are stored in registers.
- **Indirect Addressing:** The memory address of the operand is stored in a register.
- **Indexed Addressing:** Adds an index value to a base address to calculate the effective memory address.

These modes allowed for flexibility in accessing operands and optimized data manipulation for different use cases.

11.2 microNOVA MP/100

The microNOVA MP/100 supported these addressing modes:

- **Immediate Addressing:** Similar to Fairchild 9440, operands are included directly in the instruction.
- **Direct Addressing:** The instruction specifies the memory address directly.

- **Accumulator-based Addressing:** Implicitly uses the accumulator as the primary operand source or destination.

The addressing modes in the microNOVA MP/100 were simpler and focused more on the accumulator's role in operations.

11.3 Comparison

The Fairchild 9440 supported more modes like Register, Indirect, and Indexed Addressing, making it flexible for complex tasks. It also had Immediate and Direct Addressing, which were standard. The microNOVA MP/100 was simpler, with Immediate, Direct, and Accumulator-based Addressing. It was easier to use but less powerful. Both systems shared basic modes, but Fairchild 9440's extra options made it better for varied tasks, while microNOVA MP/100 focused on simplicity.

12 I/O Capabilities Comparison

12.1 Fairchild 9440

The Fairchild 9440 used Memory-Mapped I/O and Interrupt Handling to manage devices, making communication with peripherals easier. It supported Parallel I/O and could work with devices like printers and storage systems. This general-purpose I/O design made it flexible for a wide range of tasks.

12.2 microNOVA MP/100

The microNOVA MP/100 also used Memory-Mapped I/O and Interrupt Handling but focused more on desktop use. It included Serial I/O and relied on dedicated chips for specific devices like CRTs and floppy drives. Its I/O capabilities were simpler but specialized for certain setups.

12.3 Comparison

Both systems supported Memory-Mapped I/O and Interrupt Handling for device communication.

- **Fairchild 9440:** More general-purpose, with flexibility for various tasks and peripherals like printers and storage devices.
- **microNOVA MP/100:** Simpler and focused on desktop setups with specific devices like CRTs and floppy drives.

The Fairchild 9440 was designed for flexibility, while the microNOVA MP/100 prioritized simplicity and desktop integration.

13 Interrupt Support Analysis

13.1 Fairchild 9440

The Fairchild 9440 featured an interrupt-driven mechanism to handle asynchronous events efficiently. Its priority interrupt system allowed higher-priority tasks to preempt lower-priority ones. This design made it suitable for multitasking and real-time applications, where quick responses to external events were critical.

13.2 microNOVA MP/100

The microNOVA MP/100 also supported interrupt-driven operations. It used a basic, single-level interrupt system to signal the processor about peripheral or system events. While simpler than the Fairchild 9440, this approach was effective for desktop applications but less suited for complex multitasking environments.

13.3 Comparison

Both the Fairchild 9440 and microNOVA MP/100 supported interrupts for managing external events and I/O.

- **Fairchild 9440:** Offered a priority-based interrupt system, ideal for multitasking and real-time tasks.
- **microNOVA MP/100:** Used a simpler, single-level interrupt system, sufficient for basic desktop operations.

The Fairchild 9440 had advanced interrupt capabilities, while the microNOVA MP/100 focused on simplicity.

14 Data Types Supported by Each Architecture

14.1 Fairchild 9440

The Fairchild 9440 supported fixed-point integers in two's complement format for signed numbers. It also handled binary and decimal arithmetic but lacked hardware support for floating-point operations, which required software emulation.

14.2 microNOVA MP/100

The microNOVA MP/100 also supported fixed-point integers using two's complement and unsigned integers for arithmetic. Floating-point operations were not natively supported and required software routines. It did not include support for decimal arithmetic or exotic data types.

14.3 Comparison

- **Fixed-point integers:** Both architectures supported two's complement representation for signed numbers.

- **Floating-point:** Neither had hardware support; operations were emulated in software.
- **Exotic data types:** Neither system supported complex numbers or other advanced data formats.
- **Decimal arithmetic:** Fairchild 9440 supported it, while the microNOVA MP/100 did not.

Both systems focused on basic numeric operations, reflecting the technological limits of their time.

15 Speed of Each System

15.1 Fairchild 9440

The Fairchild 9440 operated with a clock speed of 4 MHz and executed instructions in 2-4 cycles, achieving approximately 1 MIPS (Million Instructions Per Second). Its higher clock speed and efficient instruction set made it faster and better suited for intensive tasks.

15.2 microNOVA MP/100

The microNOVA MP/100 had a clock speed of 2 MHz and required 5-7 cycles per instruction, resulting in approximately 0.5 MIPS. It was slower than the Fairchild 9440 but focused on simpler tasks, prioritizing cost-effectiveness.

15.3 Comparison

The Fairchild 9440 was faster due to its higher clock speed and efficient execution, making it suitable for demanding applications. The microNOVA MP/100 was slower but sufficient for basic tasks, reflecting its design for simplicity and affordability.

16 Cache Memory Usage

16.1 Fairchild 9440

The Fairchild 9440 did not use cache memory. It relied on direct memory access and pipelining for performance.

16.2 microNOVA MP/100

The microNOVA MP/100 also lacked cache memory. It used sequential memory access, consistent with its simpler architecture.

16.3 Comparison

Neither system utilized cache memory, reflecting the technological constraints of their era. The Fairchild 9440 relied on pipelining to optimize performance, while the microNOVA MP/100 used straightforward sequential memory access.

17 Typical Application Areas and Installations

17.1 Fairchild 9440

The Fairchild 9440 was primarily used in minicomputer systems, designed for general-purpose computing in scientific, industrial, and business applications. A notable example of its use was in data acquisition systems for laboratory environments. Its higher clock speed and versatile instruction set made it suitable for real-time data processing and analysis.

17.2 microNOVA MP/100

The microNOVA MP/100 was designed as a compact desktop computer, aimed at small businesses. It was commonly used for cost-effective tasks like word processing, basic accounting, and terminal emulation. A notable application was as a controller in retail point-of-sale (POS) systems, where its simplicity and reliability supported efficient transaction processing and inventory management.

18 Software and Tools

18.1 Fairchild 9440

Software: Limited information is available about software written for the Fairchild 9440. Most programs were likely custom-built for tasks like data acquisition in laboratory environments.

Availability: There is no evidence of preserved or accessible software for this architecture.

Tools and Libraries: No details about compilers, debuggers, or libraries are available in the resources.

18.2 microNOVA MP/100

Software: The microNOVA MP/100 supported software for small business applications such as word processing, accounting, and terminal emulation.

Availability: Some documentation exists, but no clear evidence suggests its software has been preserved today.

Tools and Libraries: The system included a soft console for debugging, but no specific information about compilers or libraries is mentioned.

19 Emulator Availability

Based on the research, no specific emulators for the Fairchild 9440 or microNOVA MP/100 architectures were identified. However, there are emulators available for related systems, such as the Fairchild Channel F:

- **FreeChaF:** A Fairchild Channel F emulator implemented as a libretro core. It is available for platforms like Windows and RetroPie. (SourceForge)

- **Press_F:** A lightweight Fairchild Channel F emulator, available as a desktop application, libretro API core, or Nintendo 64 homebrew. (GitHub)

These emulators are not directly applicable to the Fairchild 9440 or microNOVA MP/100 but may serve as a starting point for exploring related systems.

References

- Fairchild 9440 Datasheet. Available at: `https://archive.org/details/bitsavers_fairchildmflameDataSheetDec78_2098325/page/n5/mode/2up`
- microNOVA MP/100 Technical Documentation. Available at: `https://datageneral.uk/restorations/mpt-100-micronova/`