

ARMv2 vs SLC-500 Comparison

PKI Stud. Ugnius Vizbaras

Elementary Base and Physical Characteristics

The elementary base of the ARMv2 architecture was Very Large Scale Integration (VLSI), specifically realized in the VL86C010 microprocessor. Designed by Acorn Computers and manufactured by VLSI Technology Inc., this 32-bit Reduced Instruction Set Computer (RISC) chip was fabricated using a 2.0 μm CMOS process (VLSI Technology Inc., 1987). The processor contained 27,000 transistors and featured a die size of 34.13 mm^2 (5.842 mm \times 5.842 mm) (Furber, 2000). It was packaged in an 84-pin Plastic Leaded Chip Carrier (PLCC-84) measuring approximately 29.41 mm by 30.35 mm. While the chip was rated for clock frequencies of 10 MHz or 12 MHz, early implementations often utilized it at 8 MHz. Regarding power consumption, the manufacturer's datasheet specified a maximum power dissipation of 2 Watts (VLSI Technology Inc., 1987). This figure represented the thermal limit of the package rather than typical usage. In practice, when operating within standard parameters, the chip was highly efficient, drawing approximately 0.1 Watts (100 mW) (Furber, 2000). This low power draw allowed the processor to operate without a heatsink. The equipment built around this processor, such as the Acorn Archimedes A300 series, typically followed a desktop form factor with separate peripherals. The main unit of the A310 measured 362mm in width, 406mm in depth, and 97mm in height, and it was powered by a 50-watt power supply unit. Later models like the Acorn Archimedes A3000 integrated the motherboard and keyboard into a single "all-in-one" chassis. This unit measured 470mm by 380mm and ran on an internal power supply rated at 22 Watts.

In contrast, the elementary base of the SLC 500 is a modular, chassis-based programmable controller system rather than a single-chip implementation. The system is composed of a processor module, power supply, and I/O modules housed in a 4, 7, 10, or 13-slot chassis (Rockwell Automation, 2013). The processor modules, such as the SLC 5/03, 5/04, and 5/05, utilize modern solid-state microprocessors and Large Scale Integration (LSI) components designed to withstand industrial vibration, thermal extremes, and electrical noise (Rockwell Automation, 2013). Physically, the equipment is ruggedized for back-panel mounting; for example, a 4-slot modular chassis measures approximately 158mm in height, 171mm in width, and 145mm in depth (Rockwell Automation, 2013). Power consumption is dependent on the specific modules used; for instance, an SLC 5/03 processor draws 500 mA at 5V DC from the backplane, while an SLC 5/05 draws 1000 mA at 5V DC (Rockwell Automation, 2013). The system requires a dedicated power supply (e.g., 1746-P1) capable of handling high inrush currents up to 20 A (Rockwell Automation, 2013).

Architecture Type

The ARMv2 architecture, utilized in the VL8C010 processor, was a General Purpose Register (GPR) architecture (Seal, 200). This design was different from accumulator-based architectures or stack-based architecture. In the ARMv2 design, the processor could not perform arithmetic or logical operations directly on memory addresses. Instead, all processing happened within the internal register file. Data had to be explicitly loaded from memory into a register before it could be manipulated, and the results had to be stored back to memory using separate instructions (Furber, 2000). This architecture relied on a bank of 27 physical 32 bit registers, but the programmer generally accessed a set of 16 logical registers (R0 through R15) in any given processor mode. A specific characteristic of

the ARMv2 implementation was the usage of register R15. This register functioned as the Program Counter (PC), but also contained the Processor Status Register flags in the upper bits. This dual usage of R15 restricted the addressable memory space to 26 bits or 64 MB (Seal, 2000).

The SLC 500 architecture operates primarily as a memory-to-memory machine from the programmers perspective, utilizing a proprietary scanning architecture optimized for logic solving. The processor provides control through a processor file composed of Program Files and Data Files such as Output, Input, Status, Binary, Timer, Counter, Control, Integer, and Float (Rockwell Automation, 2008). Instructions typically operate directly on these data files rather than loading data into general-purpose registers first. For example, mathematical instructions take input values from specific memory addresses, perform the operation, and output the result to a destination memory address (Rockwell Automation, 2008).

Machine Address Format

The ARMv2 architecture is strictly classified as a three-address machine (Seal, 2000). Data processing instructions explicitly specify three distinct operands: a destination register (Rd) and two source operands (Rn and Operand2). For example, the instruction ADD R0, R1, R2 calculates the sum of R1 and R2 and places the result into R0.

The SLC 500 also functions largely as a three-address machine for its arithmetic and logical operations. Majority of the instructions, such as ADD and SUB, take two input values (Source A and Source B) and place the result in a specified Destination address (Rockwell Automation, 2008). Some instructions, however, utilize a two-address format, such as the MOV instruction, which moves a Source value to a Destination location (Rockwell Automation, 2008a), or the NEG instruction, which changes the sign of a Source and places it in a Destination (Rockwell Automation, 2008).

Registers

The ARMv2 architecture was a Load/Store machine utilizing a set of 32-bit General Purpose Registers (GPR), it had 27 physical registers. Using called register banking, the processor exposed only 16 logical registers (R0 through R15) to the programmer at any one time, swapping specific physical registers automatically during mode switches to isolate interruption contexts (Furber, 2000). While for the most part it was general-purpose, specific registers held specialized roles. For example R13 served by convention as a Stack Pointer, R14 acted as a Link Register for return addresses, and R15 uniquely combined the Program Counter and Processor Status Register into a single 32-bit value (Seal, 2000).

The SLC 500 does not utilize General Purpose Registers in the traditional sense. Instead, it uses a Data Table organized into specific file types that act as memory-mapped registers (Rockwell Automation, 2008a). For example, the Status File (S2) stores controller operation information, and the Integer File (N7) is used to store numeric values (Rockwell Automation, 2008a). The system does utilize specific internal registers for math operations; Status word S:13 and S:14 act as a "Math Register" to store the 32 bit results of multiply instructions or the remainder/quotient of divide instructions (Rockwell Automation, 2008a). Additionally, an Index Register (S:24) is provided to store offset values for indexed addressing operations (Rockwell Automation, 2008a).

Flags

The ARMv2 architecture used status flags to indicate arithmetic results and control program flow. These flags resided in the upper six bits of register R15, with the Program Counter. The four condition flags were N (Negative), Z (Zero), C (Carry), and V (Overflow) (Seal, 2000). Instructions updated these flags only when the "S" bit was set in the opcode. R15 also held two interrupt disable flags, I (IRQ Disable) and F (FIQ Disable). Placing these flags within the PC register made sure that saving R15 automatically preserved the processor status during interrupts or other subroutine calls.

The SLC 500 utilizes arithmetic status flags located in Word 0 of the Status File (S:0) (Rockwell Automation, 2008a). These flags include the Carry (C) bit (S:0/0), which is set if a carry is generated; the Overflow (V) bit (S:0/1), which sets if a result does not fit in the destination; the Zero (Z) bit (S:0/2), which sets if a result is zero; and the Sign (S) bit (S:0/3), which sets if a result is negative (Rockwell Automation, 2008a). Additionally, a Minor Error bit (S:5/0) serves as an overflow trap bit to detect mathematical overflows (Rockwell Automation, 2008a).

Machine Word Width

The ARMv2 was a 32-bit architecture with a native machine word of 32 bits. All internal registers, the Arithmetic Logic Unit (ALU), and the external data bus measured 32 bits (Furber, 2000). This design enabled the manipulation of 32-bit integers in a single instruction cycle. The architecture supported 8-bit byte and 32-bit word data transfers but lacked native support for 16-bit half-word transfers. While the data path was 32 bits wide, the addressable memory space was restricted to 26 bits (64MB) because the upper and lower bits of the Program Counter were used for the status flags.

The SLC 500 architecture is based on a 16-bit machine word. Input, Output, Status, and Integer files are all composed of 16-bit elements (Rockwell Automation, 2008a). However, the architecture supports 32-bit operations synthetically. For example, Floating Point (F8) data files use 2-word elements (32 bits per element) to store single-precision numbers (Rockwell Automation, 2008a). Furthermore, the Math Overflow Selection Bit (S:2/14) allows the processor to perform 32-bit signed math by cascading calculations across 16-bit words (Rockwell Automation, 2008a).

Memory Layout

The memory layout of the ARMv2 systems, specifically the Acorn Archimedes, was defined by the MEMC chip. The architecture utilized a logical address space of 64 MB, even though the physical data bus was 32 bits wide. This 26-bit limitation arose because the upper 6 bits and lower 2 bits of the Program Counter were reserved for status flags, leaving only bits 2 to 25 for addressing (Seal, 2000). The address space was segmented in a linear space divided into logical pages. The physical memory was subdivided into banks controlled by the MEMC. This controller managed the mapping between the logical address space seen by the software and the physical RAM chips. The maximum possible physical RAM supported by the MEMC was 4 MB. Typical system configurations for the Acorn Archimedes ranged from 1 MB in the base A310 or A3000 models up to 4 MB in expanded machines. While the theoretical address space was 64 MB, the actual installed memory was significantly smaller due to the cost of RAM at the time and the limitations of the memory controller.

The SLC 500 memory is segmented into Program Files and Data Files (Rockwell Automation, 2008a). Program files (ID 0-255) contain system information and ladder logic, while Data Files (ID 0-255) store status information and data variables (Rockwell Automation, 2008a). The addressing space is not continuous but is file-based; for example, Output files are file 0, Input files are file 1, and Integer files are file 7 (Rockwell Automation, 2008a). The maximum possible memory varies by processor model: the SLC 5/01 supports 1K or 4K instruction words, whereas the SLC 5/03, 5/04, and 5/05 support up to 64K words (Rockwell Automation, 2013).

Virtual Memory

Virtual memory support in ARMv2 systems was implemented via the MEMC chip, which acted as a Memory Management Unit. It utilized a paging mechanism rather than segmentation. The MEMC translated logical addresses into physical addresses using a single-level page table stored in physical memory. The page size was programmable, typically ranging from 4 KB to 32 KB depending on the amount of installed RAM (Furber, 2000). But it did not support swapping pages to disk.

The SLC 500 architecture does not support virtual memory. It utilizes a static memory system where user memory is comprised of specific Data Files and Program Files. Data is stored in pre-allocated tables specific to its function—such as File 0 for Outputs, File 1 for Inputs, File 4 for Timers, and File 7 for Integers—while logic is stored separately in Program files. To maintain this static map, the system relies on volatile RAM that is kept alive during power outages by a lithium battery, with the option of using a non-volatile Flash EPROM module to permanently burn the program and data.(Rockwell Automation, 2008a; Rockwell Automation, 2013).

Instruction Set Architecture

The Instruction Set Architecture of the ARMv2 was a Reduced Instruction Set Computer design containing a small set of approximately 44 to 46 instructions. There were five primary classes which are data processing instructions, data transfer instructions, control flow instructions, coprocessor instructions, and the Software Interrupt (Seal, 2000). All ARM instructions are 32 bits wide except the compressed 16-bit Thumb instructions.

The SLC 500 uses a proprietary Instruction Set Architecture represented by Ladder Logic (LAD). The instruction set varies by processor model; the SLC 5/01 supports a basic set of 52 instructions, while the SLC 5/03 and higher support over 100 instructions (Rockwell Automation, 2013). The instructions are categorized into functional groups including Bit, Timer and Counter, Compare, Compute/Math, Logical, Move, and Program Control (Rockwell Automation, 2013). Examples include XIC (Examine if Closed), TON (Timer On-Delay), MOV (Move), and PID (Proportional Integral Derivative) (Rockwell Automation, 2013).

Addressing Modes

When accessing an operand for a data processing or movement instruction, there are several standard techniques used to specify the desired location. Most ARM processors support several of these

addressing modes though few support all of them. ARMv2 supports immediate, register, register indirect, pre-indexed, post-indexed, PC relative and multiple register (Seal, 2000).

The SLC 500 supports several addressing modes including Direct Addressing, where data is stored in a specific address (e.g., N7:0); Indexed Addressing, indicated by a # symbol (e.g., #N7:10), where the processor adds the value in the Index Register (S:24) to the base address; and Indirect Addressing, available on SLC 5/03 and higher, where a substitute address is enclosed in brackets (e.g., N7:[N10:3]) (Rockwell Automation, 2008a).

I/O Capabilities

The ARMv2 architecture utilized Memory-Mapped I/O. It did not possess separate I/O instructions or a distinct I/O address space. Peripheral devices appeared to the processor as standard memory addresses. In the Acorn Archimedes implementation the IOC handled interrupts and peripherals like timers and the keyboard (Furber, 2000).

The SLC 500 is optimized for I/O control, utilizing a modular chassis system where I/O modules communicate across a backplane (Rockwell Automation, 2013). The processor updates Input (I1) and Output (O0) data files synchronously with the program scan (Rockwell Automation, 2008a). Additionally, SLC 5/02 and higher processors support Block Transfer Read (BTR) and Block Transfer Write (BTW) instructions for transferring larger blocks of data to remote devices over a Remote I/O link (Rockwell Automation, 2008a).

Interrupts

The architecture supported two primary types of hardware interrupts: IRQ and FIQ. The interrupt vector table was located at the bottom of memory, starting at address 0x00000000. When an exception occurred, the processor jumped to a specific vector address (Seal, 2000).

The SLC 500 supports user-configurable interrupt routines. These include the Selectable Timed Interrupt (STI), which interrupts the scan on a periodic basis to scan a specified subroutine (Rockwell Automation, 2008a), and the Discrete Input Interrupt (DII), which executes a subroutine when a specific input bit pattern matches a programmed compare value (Rockwell Automation, 2008a). The SLC 500 also includes a User Fault Routine to handle system faults without shutting down the processor (Rockwell Automation, 2008a).

Data Types

At the hardware instruction level, the ARMv2 supported 32-bit signed and unsigned words and 8-bit signed and unsigned bytes (Furber, 2000). Integers were processed as Two's Complement signed values or unsigned values depending on the instruction interpretation. Floating-point arithmetic was not supported natively by the ARM2 hardware. Floating-point operations were typically handled via software emulation. While the instruction set reserved opcodes for coprocessors. Consequently, complex data types like decimal numbers or complex numbers had to be implemented entirely in software libraries.

The SLC 500 supports specific data types defined by its file structure. These include Bit , Integer which stores 16-bit signed values, and Floating Point which stores 32-bit IEEE single-precision numbers (available on SLC 5/03 OS301 and higher). It also supports ASCII and String file types for text manipulation, as well as structured data types for Timers, Counters, and Control (Rockwell Automation, 2008a).

System Speed

The ARM2 processor in the Acorn Archimedes typically operated at a clock frequency of 8 MHz, with later variants running at 12 MHz. Despite this relatively low clock speed, the RISC architecture allowed it to achieve approximately 4 to 5 MIPS. Most instructions executed in a single cycle, except memory accesses and complex instructions took longer (Furber, 2000).

The SLC 500 speed is characterized by "Typical Scan Time," which is the time to execute 1K of ladder logic. The SLC 5/01 has a typical scan time of 8.0 ms/K, while the SLC 5/03 is significantly faster at 1.0 ms/K, and the SLC 5/05 is 0.9 ms/K. Bit execution times are 4 μ s for the SLC 5/01 and 0.37 μ s for the SLC 5/05 (Rockwell Automation, 2013).

Cache Memory

The ARMv2 processor did not use on-chip cache memory. It relied on the speed of Fast Page Mode DRAM and the MEMC to feed instructions to the processor. The lack of cache was a design choice to keep the transistor count and cost low (Furber, 2000).

The SLC 500 processors do not utilize user-accessible cache memory. They rely on battery-backed RAM or Flash EEPROM modules for program execution and data storage (Rockwell Automation, 2013).

Typical Application Areas

The primary application area was the educational market in the United Kingdom. It was the standard computer in many British schools during the late 1980s and early 1990s. Beyond education, it was used in desktop publishing and television production due to its graphical capabilities.

The SLC 500 is designed for a broad range of industrial automation applications, including material handling, HVAC control, high-speed assembly operations, small process control, and SCADA applications (Rockwell Automation, 2013). A typical installation might involve an SLC 5/05 processor controlling a bottling line via Ethernet, while monitoring sensors and actuators through local I/O modules (Rockwell Automation, 2008a).

Software and Tools

A moderate library of software was produced for the ARMv2, specifically targeting the RISC OS operating system. The software was mainly targeted towards education, productivity and utilities. Game library also existed, including the seminal game Zarch mainly used to showcase the machine

speed. When it comes to tools it supported "BBC BASIC V" and also it had Acorn C/C++ compiler suite.

The SLC 500 is programmed using RSLogix 500 software, a Windows-based ladder logic programming package that provides editing, I/O configuration, and diagnostic tools (Rockwell Automation, 2013). Communication with the processor is handled by RSLinx software (Rockwell Automation, 2013), and network configuration for ControlNet and DeviceNet is managed via RSNetWorx (Rockwell Automation, 2013).

Emulation

For the ARMv2, emulators such as Arculator and Archie exist to replicate the Acorn Archimedes environment. For the SLC 500, Rockwell Automation offers RSLogix Emulate 500 software, which emulates one or more SLC 500 processors to test and debug ladder logic programs in a Windows environment without physical hardware (Rockwell Automation, 2013).

Conclusion

The comparison establishes that the ARMv2 and SLC 500 utilize opposing architectural strategies suited to their distinct environments. The ARMv2 focuses on VLSI integration and a register-based RISC design to maximize computational efficiency on a single low-power chip. In contrast, the SLC 500 prioritizes physical ruggedness and modularity, employing a memory-to-memory scanning architecture for deterministic machine control. This distinction confirms that the ARMv2 was engineered for data processing, while the SLC 500 was designed for reliability and I/O actuation.

References:

Furber, S. (2000). ARM System-on-Chip Architecture (2nd Ed.). Addison-Wesley.

Rockwell Automation (2005). SLC 500 BASIC and BASIC-T Modules User Manual. (Publication 1746-UM004B-EN-P). Rockwell Automation, Inc.

Rockwell Automation (2008a). SLC 500 Instruction Set Reference Manual. (Publication 1747-RM001G-EN-P). Rockwell Automation, Inc.

Rockwell Automation (2008b). Installation Instructions SLC 5/03, SLC 5/04, and SLC 5/05 Modular Processors. (Publication 1747-IN009B-EN-P). Rockwell Automation, Inc.

Rockwell Automation (2013). SLC 500 Systems Selection Guide. (Publication 1747-SG001E-EN-E). Rockwell Automation, Inc.

Seal, D. (Ed.). (2000). ARM Architecture Reference Manual. Addison-Wesley.

VLSI Technology Inc. (1987). VL86C010 32-Bit RISC Microprocessor Data Manual. VLSI Technology Inc.