

ALPHA AXP OVERVIEW

Jurij Šilc[†], Borut Robič[†] and Jože Buh[‡]

[†] Jožef Stefan Institute

Laboratory for Computer Architectures, Jamova 39, Ljubljana, Slovenia

Phone: +38 61 159 199, Fax: +38 61 161 029

E-mail: jurij.silc@ijs.si or borut.robic@ijs.si

[‡] EuroComputer Systems

Digital Authorized Representative, Vojkova 50, Ljubljana, Slovenia

Phone: +38 61 182 500, Fax: +38 61 181 005

Keywords: Alpha AXP architecture, RISC architecture, multiple instruction issue, shared-memory multiprocessing, overview

Edited by: Marco Botta

Received: February 22, 1993

Revised: March 15, 1993

Accepted: March 22, 1993

The paper describes the Alpha AXP architecture and some existing implementations. It is a true 64-bit RISC architecture supporting multiple instruction issue, shared-memory multiprocessing, and several today's leading operating system environments. The first Alpha AXP microprocessor DECchip 21064 and several hardware products using it are also briefly described.

1 Introduction

As the 20th Century draws to a close, more and more computer power is being needed to drive our extremely complex applications. The application of computing took us from mainframes and 16-bit minicomputers of the early 1970s to the 32-bit microprocessors of the mid-1980s, in the age of desktop computing with windowing user interfaces. What will stimulate the next technological leap that brings us applications of the future? One of the answers might be an advanced 64-bit RISC architecture.

In late 1970s DEC introduced the VAX architecture with one hardware product (the VAX 11/780), one operating system (VAX VMS), one network (DECnet) and one high-level language (Fortran).

In early 1990s it introduced Alpha AXP architecture with several hardware products, three operating systems, multiple networking protocols, multiple languages, and so forth.

In this paper, we shall present a short overview of the Alpha project. We will start with description of main project goals and proceed to basic architectural features such as multiple instruc-

tion issuing and the possibility of multiprocessing. The first implementation of the Alpha AXP architecture is DECchip 21064. The chip and several hardware products using it are briefly described. Finally, we give a short overview of the operating systems supported by Alpha AXP architecture.

2 Project overview

Alpha was the largest engineering project in Digital's history, spanning more than 30 engineering groups in ten countries [8]. It started with a task force chartered to define a high-performance RISC architecture for the 1990s and beyond. Even before the architecture definition was complete, work began on implementing a high-performance microprocessor. The work was done in the summer 1991 when a product-level chip DECchip 21064 was fabricated [4]. However, a prototype chip was fabricated in late 1990 and was used in an experimental multiprocessor system called ADU (Alpha demonstration unit) [9]. This system was of great benefit to software developers since it allowed them to boot the first Alpha AXP operating systems early in 1991.

3 Project goals

The Alpha AXP architecture¹ project started with a small list of goals:

- **High performance and longevity.** In current architectures, a primary limitation is the 32-bit memory address. Therefore, the project adopted a full 64-bit architecture (with a minimal number of 32-bit operations for backward compatibility). It was estimated that it would be reasonable for raw clock rates to improve only by a factor of 10 over the coming 25 years. If the clock cannot be made faster, more work should be done per clock tick to obtain increase in performance. Alpha AXP architecture was therefore designed to encourage *multiple instruction issue* implementations eventually sustaining about 10 new instructions starting every clock cycle. Additional performance improvements are to be expected from multiple processors. Hence, Alpha AXP architecture project early focused on multiple processors, and designed a multiprocessor memory model and matching instructions from the very beginning.

- **Run several operating systems.** Underpinnings were placed for interrupt delivery and return, exceptions, context switching, memory management, and error handling, all in a set of privileged software subroutines called PALcodes. By having different sets of PALcode for different operating systems, neither the hardware nor the operating system is burdened with a bad interface match, and the architecture is not biased toward a particular computing style.

- **Easy migration from other architecture customer bases.** To run an existing (old architecture, such as VAX and MIPS) binary version of a complex application, the idea of *binary trans-*

¹ *Computer architecture* is defined as the attributes and behavior of a computer as seen by a machine language programmer, while *implementation* is defined as the actual hardware structure, logic design, and data-path organization of a particular embodiment of the architecture. The architecture therefore carefully describes the behavior that a machine language programmer sees, but does not describe the means by which a particular implementation achieves that behavior.

lation was adopted [7]. It allows a user to get applications up and running immediately, with minimal porting effort.

4 Alpha AXP architecture

4.1 Its approach to RISC architecture

Alpha is a 64-bit load/store RISC architecture designed with particular emphasis on clock speed, multiple instruction issue, and multiple processors [1]. Its architects examined and analyzed current and theoretical RISC architecture design elements and developed high-performance alternatives for the Alpha architecture.

4.2 True 64-bit architecture

All registers are 64 bit in length and all operations are performed between 64-bit registers. Hence, it is not a 32-bit architecture that was later expanded to 64 bits. There are 32 integer registers R0..R31 and 32 floating-point registers F0..F31.

The basic unit of data is 64-bit *quadword*. There are three fundamental datatypes: integer (32-bit longword, 64-bit quadword), IEEE floating-point (32-bit S-floating point, 64-bit T-floating point), and VAX floating-point (32-bit F-floating point, 64-bit G-floating point).

Table 1: MIN and MAX Values for the Floating-point Data Formats

Data Format	MIN	MAX
F-floating	0.294e-38	1.70e38
G-floating	0.56e-308	0.899e308
S-floating	1.175e-38	3.40e38
T-floating	2.225e-308	1.798e308

Each of 168 Alpha instructions is 32 bits in length. There are four major instruction formats (PALcode, Branch, Memory, Operate) and all have 6-bit opcode.

- **PALcode instructions.** These instructions specify one of few dozen complex operations from Privileged Architecture Library² to be performed.

²A Privileged Architecture Library is a set of subroutines that is specific to a particular Alpha operating-system implementation.

- **Branch instructions.** Conditional branch instructions can test a register for positive/negative or for zero/nonzero. They can also test integer registers for even/odd. Unconditional branch instructions can write a return address into a register. There is also a calculated jump instruction that branches to an arbitrary 64-bit address in a register.
- **Memory instructions.** Memory instructions are used for loads, stores, and a few miscellaneous operations.
- **Operate instructions.** There are five groups of register-to-register operate instructions: integer arithmetic, logical, byte-manipulation, floating-point, and miscellaneous.

4.3 Multiple instruction issue

Alpha implementation will issue multiple instructions in a single cycle. To improve the odds of multiple-issue, compilers should choose pairs of instructions to put in aligned quadwords. Pick one from type A and one from type B but only a total of one Load/Store and Branch per pair:

Type A	Type B
Integer Operate	Floating Operate
Floating Load/Store	Integer Load/Store
Floating Branch	Integer Branch
Integer Operate	Floating Operate
	Unconditional Branch
	Branch to Subroutine
	Jump to Subroutine

To avoid any mechanism that would hinder such implementations, all special or hidden processor resources were avoided [6]. Therefore, there are:

- **No branch delay slots.** Branch delay slots require exactly one following instruction to be executed after a conditional branch. This, however, does not scale well to a multiple-way issue chip with a multiple-cycle instruction cache where several instructions will be needed in the delay slot.
- **No suppressed instructions or skips.** When execution of one instruction conditionally suppresses or skips a following one (found in some other RISC architectures) the suppression bits represent a nonreplicated hidden state. Hence,

it is difficult to multi-use more than one potential suppressor.

- **No precise arithmetic exceptions.** Reporting an arithmetic exception (such as overflow and underflow) means that instructions subsequent to the one causing the exception must not be executed. This, however, becomes difficult in a pipelined multiple issue implementation. Alpha architecture uses the Trap Barrier instruction which stalls instruction issuing until all prior instructions are guaranteed to complete without incurring arithmetic traps. A code-generation design was documented by Alpha project which needs one trap barrier per branch to give precise reporting.

- **No single-byte writes to memory.** The byte load/store instructions found in some other RISC architectures can be a performance bottleneck because they require an extra byte shifter in the speed-critical load and store paths, and they force a hard choice in fast cache design. Therefore, in the Alpha AXP architecture, a byte load is done as an explicit load/shift sequence; a byte store as an explicit load/modify/store sequence. Instructions in these sequences can be multi-issued with other computation.

Moreover, there are no condition codes, no global exception enables, and no multiplier-quotient or string registers.

4.4 Shared-memory multiprocessing

An Alpha system consists of a collection of processors and shared coherent memories that are accessible by all processors. (There may also be unshared memories.) There are several types of accesses³ that a processor may generate to shared memory locations (I-stream access, D-stream accesses, and barriers).

Writes to shared data must be synchronized by the programmer.

The basic multiprocessor interlocking primitive

³Instruction fetch by processor *i* to location *x*, returning value *a*.

Data read by processor *i* to location *x*, returning value *a*.

Data write by processor *i* to location *x*, returning value *a*.

Memory barrier instruction issued by processor *i*.

I-stream memory barrier instruction issued by processor *i*.

Table 2: Competitive Position

Vendor	Digital	MIPS	Sun/TI	IBM	HP	Intel	Motorola
Device	21064	R4000	Viking	RIOS	PA-4	i860XP	88110
Max freq. (internal)	200MHz	100MHz	50MHz	50MHz	66MHz	50MHz	50MHz
No. chips required	1	1	1	7-9	2	1	1
Peak MIPS	400	100	150	200	132	150	150
Peak MFLOPS	200	50	50	100	132	100	100
Base arch. design	64-bit	64-bit	32-bit	32-bit	32-bit	32-bit	32-bit

Table 3: CMOS Technology Roadmap

CMOS	1	2	3	4
Mfg Year	1985	1987	1989	1991
Min features size	2.0 μ m	1.5 μ m	1.0 μ m	0.75 μ m
Chip power supply	5.0V	5.0V	3.3V	3.3V
Max μ P chip size	0.9cm ²	1.2cm ²	1.5cm ²	2.2cm ²
Gate oxide	30nm	22nm	15nm	10nm
Effective L	1.3 μ m	1.0 μ m	0.7 μ m	0.5 μ m
# of wiring levels	2	2	3	3
# of transistors	200,000	400,000	800,000	1,680,000

is a RISC-style load-locked, in-register modify, store-conditional sequence of instructions.

If the sequence runs without interrupt, exception, an interfacing write from another processor, or a PALcode instruction, then the conditional store succeeds – an atomic update was in fact performed. Otherwise, the store fails and the program eventually must branch back and retry the sequence until it succeeds.

This style of interlocking scales well with very fast caches, and makes Alpha a suitable architecture for building multiprocessor systems. There is no strict multiprocessor read/write ordering, whereby the sequence of reads and writes issued by one processor is delivered to all other processors in exactly the order issued. The strict ordering can be specified when needed by insertion of Memory Barrier instruction. This instruction guarantees that all subsequent loads or stores will not access memory until after all previous loads and stores have accessed memory, as observed by other processors.

5 An implementation: DECchip 21064

DECchip 21064 microprocessor represents the first implementation of the Alpha AXP architecture [4]. It is a super-scalar super-pipelined processor, using dual instruction issue, that has sampled up to 200MHz cycle time. Super-pipelined means that an instruction is issued to the functional unit at every clock tick and the results are pipelined. The integer pipeline is seven stages deep, where each stage is a 5 ns clock cycle. The first four stages are associated with instruction fetching, decoding, and scoreboard checking of operands. Pipeline stages 0 through 3 can be stalled. Beyond 3, however, all pipeline stages advance every cycle. Most ALU operations complete in cycle 4 allowing single-cycle latency, with the shifter being the exception. Primary cache accesses complete in cycle 6, so cache latency is three cycles. The floating-point pipeline is identical and mostly shared with the integer pipeline in stages 0 through 3, however, the execution phase is three cycles longer.

Table 2 shows competitive position of DECchip 21064 microprocessor.

Table 4: Alpha AXP System Comparison Chart

System	3000/400	3000/500	4000	7000	10000
No. of processors	1	1	1 or 2	up to 6	up to 6
CPU	DECchip 21064	DECchip 21064	DECchip 21064	DECchip 21064	DECchip 21064
Clock speed	133 MHz	150 MHz	160 MHz	182 MHz	200 MHz
Max memory capacity	512 MB	1 GB	2 GB	14 GB	14 GB
Max disk capacity	9.5 GB	11.6 GB	56 GB	over 10 TB	over 10 TB
Max I/O throughput	90 MB/s	100 MB/s	160 MB/s	400 MB/s	400 MB/s

The CMOS process is a .75 micron and 1.4- by 1.7-cm chip incorporating 1.68 million transistors. DECchip 21064 include 8KB instruction cache, 8KB data cache and two associated translation buffers, a four-entry 32B/entry write buffer, a pipelined 64B integer execution unit with 32-entry register file, and a pipelined floating-point unit with an additional 32 registers. The bus interface unit handles all communication between the chip and environment. The CMOS technology used to manufacture the DECchip 21064 evolved from three previous generation used to produce very high-performance microprocessors (Table 3).

6 Hardware products

At the time of writing, there are several hardware products available spanning desktop through data center: DEC 3000/400 AXP (Desktop Workstation or System), DEC 3000/500 AXP (Deskside Workstation or System), DEC 4000 AXP (Distributed/Departmental System), DEC 7000 AXP (Data Center System), and DEC 10000 AXP (Mainframe-Class System). Three more products are due in the next few months [3]. See Table 4.

7 Operating systems

Alpha AXP supports today's leading operating system environments: OpenVMS, UNIX, and Windows NT (to be announced soon).

• OpenVMS AXP.

VMS, now known as OpenVMS, supports open industry standard interfaces⁴, system purchasing

⁴The OpenVMS operating environment complies with IEEE POSIX and OSF/Motif standards and is X/Open XPG3 BASE Branded. Future plans also call for Open-

flexibility, and licensing. This combination is so new to the industry that the VMS operating system has been renamed to OpenVMS. It runs on both VAX and Alpha AXP hardware platforms. OpenVMS Alpha AXP provides the same features as OpenVMS VAX, enabling users and applications to easily move from one system to another. The benefits of VAXclusters will be made available on OpenVMS Alpha AXP, allowing OpenVMS VAX and OpenVMS Alpha AXP systems to co-exist in the same cluster. The main purpose of moving the OpenVMS system to the Alpha AXP architecture was to deliver the performance advantages of RISC to OpenVMS applications [5].

• DEC OSF/1 AXP.

DEC OSF/1 AXP is a true native UNIX. It implements the common definition agreed upon by UNIX Systems Labs (System V) and the Open Software Foundation (OSF/1):

- OSF Application Environment Specification;
- Systems V Interface Definition;
- OSF/Motif User interface;
- Distributed Computing Environment support; and
- Distributed Management Environment support committed.

DEC OSF/1 AXP supports several key standards in the area of the operating and window systems.⁵ It adds a range of enhanced program-

VMS compliance with OSF Distributed Computing Environment, and support for XPG4.

⁵IEEE POSIX 1003.1 (1990), 1003.2 (partial), 1003.4a (threads), and 1004.3 D11 (threads); FIPS 160 (ANSI C); X/Open Portability Guide 3; System V Interface Definition 2; System V Release compatibility (all SVID3 Base and Kernel Extensions with the exceptions of streams, signals, and counters); 4.3 BSD; Applications Environment specification (AES); MITs X Window System, X11 Release 5; Motif version 1.1.3; and ISO 9660 (CDROM f.s.).

ming tools available with DEC OSF/1 Developer Extension package. These tools provide a complete software development environment for programmers and application developers. It also provides ULTRIX compatibility, through standards conformance, development tools networking, user interfaces, data interoperability and compilation systems, allowing ULTRIX customers to move to DEC OSF operating system on the Alpha AXP architecture in one step.

To support realtime, DEC OSF/1 offers:

- *A pre-emptive kernel*, to ensure that external realtime events get immediate attention
- *Fixed priority scheduling*, to ensure that realtime applications aren't delayed by background activity;
- *Clocks and timers*, to provide the increased functionality and granularity needed for realtime applications;
- *Process memory locking*, to prevent system paging and swapping that could cause the system to respond unpredictably;
- *Asynchronous I/O*, that enables application between realtime processes;
- *Semaphores*, for fast, reliable communication between realtime processes;
- *Shared memory*, for fast data sharing between processes or applications.

• Windows NT AXP.

Alpha AXP systems will run Microsoft's Windows NT. Through Windows NT, users who use DOS and Window-based applications today can continue to use them tomorrow on Alpha AXP-based systems that run many times faster than today's fastest PCs.

For these operating systems, the November 1992 edition of the Alpha AXP Software Application Listing [2] provides a compendium of information on over 1500 software products.

References

- [1] Digital Equipment Corporation (1992) *Alpha Architecture Handbook*. EC-H1689-10.
- [2] Digital Equipment Corporation (1992) *Alpha AXP Software Application Listing*. First Edition, EC-J2088-10, November 1992.
- [3] Digital Equipment Corporation (1993) *Alpha AXP Systems Summary*. EC-F2183-46.
- [4] Dobberpuhl D. W. et al. (1992) A 200 MHz 61R Dual Issue CMOS Microprocessor. *IEEE Trans. Solid State Circuits*, 27, 11, p. 1555-1567.
- [5] Kronenberg N. et al. (1993) Porting Open-VMS from VAX to Alpha AXP. *Comm. ACM*, 36, 2, p. 45-53.
- [6] Sites R. L. (1993) Alpha AXP Architecture. *Comm. ACM*, 36, 2, p. 33-44.
- [7] Sites R. L. et al. (1993) Binary Translation. *Comm. ACM*, 36, 2, p. 69-81.
- [8] Supnik R. M. (1993) Digital's Alpha Project. *Comm. ACM*, 36, 2, p. 30-32.
- [9] Thacker C. P., Conroy D. G. & Stewart L. C. (1993) The Alpha Demonstration Unit: A High-Performance Multiprocessor. *Comm. ACM*, 36, 2, p. 55-67.

The following trademarks are used in this paper: AXP, Alpha AXP, DEC, DECchip, DECnet, Digital, OpenVMS, ULTRIX, VAX, VAXcluster, VMS are trademarks of Digital Equipment Corporation, OSF/1 and OSF/Motif are registered trademarks of the Open Software Foundation, Windows NT is a trademark of Microsoft Corporation, and UNIX is a registered trademark of UNIX System Laboratories, Inc.