11 11 TEX LATEX

Brief Introduction to Alpha Systems and Processors

Neal Crook, Digital Equipment (Editor: *David Mosberger* <mailto:davidm@azstarnet.com>) V0.11, 6 June 1997

This document is a brief overview of existing Alpha CPUs, chipsets and systems. It has something of a hardware bias, reflecting my own area of expertese. Although I am an employee of Digital Equipment Corporation, this is not an official statement by Digital and any opinions expressed are mine and not Digital's.

Contents

1 What is Alpha

"Alpha" is the name given to Digital's 64-bit RISC architecture. The Alpha project in Digital began in mid-1989, with the goal of providing a high-performance migration path for VAX customers. This was not the first RISC architecture to be produced by Digital, but it was the first to reach the market. When Digital announced Alpha, in March 1992, it made the decision to enter the merchant semicondutor market by selling Alpha microprocessors.

Alpha is also sometimes referred to as Alpha AXP, for obscure and arcane reasons that aren't worth persuing. Suffice it to say that they are one and the same.

2 What is Digital Semiconductor

Digital Semiconductor <http://www.digital.com/info/semiconductor/> (DS) is the business unit within Digital Equipment Corporation (Digital - we don't like the name DEC) that sells semiconductors on the merchant market. Digital's products include CPUs, support chipsets, PCI-PCI bridges and PCI peripheral chips for comms and multimedia.

3 Alpha CPUs

There are currently 2 generations of CPU core that implement the Alpha architecture:

- EV4
- EV5

Opinions differ as to what "EV" stands for (Editor's note: the true answer is of course "Electro Vlassic" 12 ([1])), but the number represents the first generation of Digital's CMOS technology that the core was implemented in. So, the EV4 was originally implemented in CMOS4. As time goes by, a CPU tends to get a mid-life performance kick by being optically shrunk into the next generation of CMOS process. EV45, then, is the EV4 core implemented in CMOS5 process. There is a big difference between shrinking a design into a particular technology and implementing it from scratch in that technology (but I don't want to go into that now). There are a few other wildcards in here: there is also a CMOS4S (optical shrink in CMOS4) and a CMOS5L.

True technophiles will be interested to know that CMOS4 is a 0.75 micron process, CMOS5 is a 0.5 micron process and CMOS6 is a 0.35 micron process.

To map these CPU cores to *chips* we get:

21064-150,166

EV4 (originally), EV4S (now)

21064-200

EV4S

21064A-233,275,300

EV45

21066

LCA4S (EV4 core, with EV4 FPU)

21066A-233

LCA45 (EV4 core, but with EV45 FPU)

21164-233,300,333

EV5

21164A-417

EV56

21264

EV6 `<http://www.mdronline.com/report/articles/21264/21264.html>`

The EV4 core is a dual-issue (it can issue 2 instructions per CPU clock) superpipelined core with integer unit, floating point unit and branch prediction. It is fully bypassed and has 64-bit internal data paths and tightly coupled 8Kbyte caches, one each for Instruction and Data. The caches are write-through (they never get dirty).

The EV45 core has a couple of tweaks to the EV4 core: it has a slightly improved floating point unit, and 16KB caches, one each for Instruction and Data (it also has cache parity). (Editor's note: Neal Crook indicated in a separate mail that the changes to the floating point unit (FPU) improve the performance of the divider. The EV4 FPU divider takes 34 cycles for a single-precision divide and 63 cycles for a double-precision divide (non data-dependent). In constrast, the EV45 divider takes typically 19 cycles (34 cycles max) for single-precision and typically 29 cycles (63 cycles max) for a double-precision division (data-dependent).)

The EV5 core is a quad-issue core, also superpipelined, fully bypassed etc etc. It has tightly-coupled 8Kbyte caches, one each for I and D. These caches are write-through. It also has a tightly-coupled 96Kbyte on-chip second-level cache (the Scache) which is 3-way set associative and write-back (it can be dirty). The EV4->EV5 performance increase is better than just the increase achieved by clock speed improvements. As well as the bigger caches and quad issue, there are microarchitectural improvements to reduce producer/consumer latencies in some paths.

The EV56 core is fundamentally the same microarchitecture as the EV5, but it adds some new instructions for 8 and 16-bit loads and stores (see Section 8 (Bytes and all that stuff)). These are primarily intended for use by device drivers. The EV56 core is implemented in CMOS6, which is a 2.0V process.

The 21064 was anounced in March 1992. It uses the EV4 core, with a 128-bit bus interface. The bus interface supports the 'easy' connection of an external second-level cache, with a block size of 256-bits (2 data beats on the bus). The Bcache timing is completely software configurable. The 21064 can also be configured to use a 64-bit external bus, (but I'm not sure if any shipping system uses this mode). The 21064 does not impose any policy on the Bcache, but it is usually configured as a write-back cache. The 21064 does contain hooks to allow external hardware to maintain cache coherence with the Bcache and internal caches, but this is hairy.

The 21066 uses the EV4 core and integrates a memory controller and PCI host bridge. To save pins, the memory controller has a 64-bit data bus (but the internal caches have a block size of 256 bits, just like the 21064, therefore a block fill takes 4 beats on the bus). The memory controller supports an external Bcache and external DRAMs. The timing of the Bcache and DRAMs is completely software configurable, and can be controlled to the resolution of the CPU clock period. Having a 4-beat process to fill a cache block isn't as bad as it sounds because the DRAM access is done in page mode. Unfortunately, the memory controller doesn't support any of the new esoteric DRAMs (SDRAM, EDO or BEDO) or synchronous cache RAMs. The PCI bus interface is fully rev2.0 compliant and runs at upto 33MHz.

The 21164 has a 128-bit data bus and supports split reads, with upto 2 reads outstanding at any time (this allows 100% data bus utilisation under best-case dream-on conditions, i.e., you can theoretically transfer 128-bits of data on every bus clock). The 21164 supports easy connection of an external 3-rd level cache (Bcache) and has all the hooks to allow external systems to maintain full cache coherence with all caches. Therefore, symmetric multiprocessor designs are 'easy'.

The 21164A was announced in October, 1995. It uses the EV56 core. It is nominally pin-compatible with the 21164, but requires split power rails; all of the power pins that were +3.3V power on the 21164 have now been split into two groups; one group provided 2.0V power to the CPU core, the other group supplies 3.3V to the I/O cells. Unlike older implementations, the 21164 pins are not 5V-tolerant. The end result of this change is that 21164 systems are, in general, not upgradeable to the 21164A (though note that it would be relatively straightforward to design a 21164A system that could also accommodate a 21164). The 21164A also has a couple of new pins to support the new 8 and 16-bit loads and stores. It also improves the 21164 support for using synchronus SRAMs to implement the external Bcache.

4 21064 performance vs 21066 performance

The 21064 and the 21066 have the same (EV4) CPU core. If the same program is run on a 21064 and a 21066, at the same CPU speed, then the difference in performance comes only as a result of system Bcache/memory bandwidth. Any code thread that has a high hit-rate on the *internal* caches will perform the same. There are 2 big performance killers:

1. Code that is write-intensive. Even though the 21064 and the 21066 have write buffers to swallow some of the delays, code that is write-intensive will be throttled by write bandwidth at the system bus. This arises because the on-chip caches are write-through.

2. Code that wants to treat floats as integers. The Alpha architecture does not allow register-register transfers from integer registers to floating point registers. Such a conversion has to be done via memory (And therefore, because the on-chip caches are write-through, via the Bcache). (Editor's note: it seems that both the EV4 and EV45 can perform the conversion through the primary data cache (Dcache), provided that the memory is cached already. In such a case, the store in the conversion sequence will update the Dcache and the subsequent load is, under certain circumstances, able to read the updated d-cache value, thus avoiding a costly roundtrip to the Bcache. In particular, it seems best to execute the stq/ldt or stt/ldq instructions back-to-back, which is somewhat counter-intuitive.)

If you make the same comparison between a 21064A and a 21066A, there is an additional factor due to the different Icache and Dcache sizes between the two chips.

Now, the 21164 solves both these problems: it achieve *much* higher system bus bandwidths (despite having the same number of signal pins - yes, I *know* it's got about twice as many pins as a 21064, but all those extra ones are power and ground! (yes, really!!)) and it has write-back caches. The only remaining problem is the answer to the question "how much does it cost?"

5 A Few Notes On Clocking

All of the current Alpha CPUs use high-speed clocks, because their microarchitectures have been designed as so-called short-tick designs. None of the sytem busses have to run at horrendous speeds as a result though:

- on the 21066(A), 21064(A), 21164 the off-chip cache (Bcache) timing is completely programmable, to the resolution of the CPU clock. For example, on a 275MHz CPU, the Bcache read access time can be controller with a resolution of 3.6ns
- on the 21066(A), the DRAM timing is completely programmable, to the resolution of the CPU clock (*not* the PCI clock, the CPU clock).
- on the 21064(A), 21164(A), the system bus frequency is a sub-multiple of the CPU clock frequency. Most of the 21064 motherboards use a 33MHz system bus clock.
- Systems that use the 21066 can run the PCI at any frequency relative to the CPU. Generally, the PCI runs at 33MHz.
- Systems that use the APECs chipset (see Section 6 ()) always have their CPU system bus equal to their PCI bus frequency. This means that both busses tends to run at either 25MHz or 33MHz (since these are the frequencies that scale up to match the CPU frequencies). On APECs systems, the DRAM controller timings are software programmable in terms of the CPU system bus frequency

Aside: someone suggested that they were getting bad performance on a 21066 because the 21066 memory controller was only running at 33MHz. Actually, it's the superfast 21064A systems that have memory controllers that 'only' run at 33MHz.

6 The chip-sets

DS sells two CPU support chipsets. The 2107x chipset (aka APECS) is a 21064(A) support chiset. The 2117x chipset (aka ALCOR) is a 21164 support chipset. There will also be 2117xA chipset (aka ALCOR 2) as a 21164A support chipset.

Both chipsets provide memory controllers and PCI host bridges for their CPU. APECS provides a 32-bit PCI host bridge, ALCOR provides a 64-bit PCI host bridge which (in accordance with the requirements of the PCI spec) can support both 32-bit and 64-bit PCI devices.

APECS consists of 6, 208-pin chips (4, 32-bit data slices (DECADE), 1 system controller (COMANCHE), 1 PCI controller (EPIC)). It provides a DRAM controller (128-bit memory bus) and a PCI interface. It also does all the work to maintain memory coherence when a PCI device DMAs into (or out of) memory.

ALCOR consists of 5 chips (4, 64-bit data slices (Data Switch, DSW) - 208-pin PQFP and 1 control (Control, I/O Address, CIA) - a 383 pin plastic PGA). It provides a DRAM controller (256-bit memory bus) and a PCI interface. It also does all the work required to support an external Bcache and to maintain memory coherence when a PCI device DMAs into (or out of) memory.

There is no support chipset for the 21066, since the memory controller and PCI host bridge functionality are integrated onto the chip.

7 The Systems

The applications engineering group in DS produces example designs using the CPUs and support chipsets. These are typically PC-AT size motherboards, with all the functionality that you'd typically find on a high-end Pentium motherboard. Originally, these example designs were intended to be used as starting points for third-parties to produce motherboard designs from. These first-generation designs were called Evaluation Boards (EBs). As the amount of engineering required to build a motherboard has increased (due to higher-speed clocks and the need to meet RF emission and susceptibility regulations) the emphasis has shifted towards providing motherboards that are suitable for volume manufacture.

Digital's system groups have produced several generations of machines using Alpha processors. Some of these systems use support logic that is designed by the systems groups, and some use commodity chipsets from DS. In some cases, systems use a combination of both.

Various third-parties build systems using Alpha processors. Some of these companies design systems from scratch, and others use DS support chipsets, clone/modify DS example designs or simply package systems using build and tested boards from DS.

The EB64: Obsolete design using 21064 with memory controller implemented using programmable logic. I/O provided by using programmable logic to interface a 486<->ISA bridge chip. On-board Ethernet, SuperI/O (2S, 1P, FD), Ethernet and ISA. PC-AT size. Runs from standard PC power supply.

The EB64+: Uses 21064 or 21064A and APECs. Has ISA and PCI expansion (3 ISA, 2 PCI, one pair are on a shared slot). Supports 36-bit DRAM SIMs. ISA bus generated by Intel SaturnI/O PCI-ISA bridge. On-board SCSI (NCR 810 on PCI) Ethernet (Digital 21040), KBD, MOUSE (PS2 style), SuperI/O (2S, 1P, FD), RTC/NVRAM. Boot ROM is EPROM. PC-AT size. Runs from standard PC power supply.

The EB66: Uses 21066 or 21066A. I/O sub-system is identical to EB64+. Baby PC-AT size. Runs from standard PC power supply. The EB66 schematic was published as a marketing poster advertising the 21066 as "the first microprocessor in the world with embedded PCI" (for trivia fans: there are actually 2 versions of this poster - I drew the circuits and wrote the spiel for the first version, and some Americans mauled the spiel for the second version)

The EB164: Uses 21164 and ALCOR. Has ISA and PCI expansion (3 ISA slots, 2 64-bit PCI slots (one is shared with an ISA slot) and 2 32-bit PCI slots. Uses plus-in Bcache SIMMs. I/O sub-system provides SuperI/O (2S, 1P, FD), KBD, MOUSE (PS2 style), RTC/NVRAM. Boot ROM is Flash. PC-AT-sized motherboard. Requires power supply with 3.3V output.

The AlphaPC64 (aka Cabriolet): derived from EB64+ but now baby-AT with Flash boot ROM, no on-board SCSI or Ethernet. 3 ISA slots, 4 PCI slots (one pair are on a shared slot), uses plug-in Bcache SIMMs. Requires power supply with 3.3V output.

The AXPpci33 (aka NoName), is based on the EB66. This design is produced by Digital's Technical OEM (TOEM) group. It uses the 21066 processor running at 166MHz or 233MHz. It is a baby-AT size, and runs from a standard PC power supply. It has 5 ISA slots and 3 PCI slots (one pair are a shared slot). There are 2 versions, with either PS/2 or large DIN connectors for the keyboard.

Other 21066-based motherboards: most if not all other 21066-based motherboards on the market are also based on EB66 - there's really not many system options when designing a 21066 system, because all the control is done on-chip.

Multia (aka the Universal Desktop Box): This is a very compact pedestal desktop system based on the 21066. It includes 2 PCMCIA sockets, 21030 (TGA) graphics, 21040 Ethernet and NCR 810 SCSI disk along with floppy, 2 serial ports and a parallel port. It has limited expansion capability (one PCI slot) due to its compact size. (There is some restriction on when you can use the PCI slot, can't remember what) (Note that 21066A-based and Pentium-based Multia's are also available).

DEC PC 150 AXP (aka Jensen): This is a very old Digital system - one of the first-generation Alpha systems. It is only mentioned here because a number of these systems seem to be available on the second-hand market. The Jensen is a floor-standing tower system which used a 150MHz 21064 (later versions used faster CPUs but I'm not sure what speeds). It used programmable logic to interface a 486 EISA I/O bridge to the CPU.

Other 21064(A) systems: There are 3 or 4 motherboard designs around (I'm not including Digital *systems* here) and all the ones I know of are derived from the EB64+ design. These include:

- EB64+ (some vendors package the board and sell it unmodified); AT form-factor.
- Aspen Systems motherboard: EB64+ derivative; baby-AT form-factor.
- Aspen Systems server board: many PCI slots (includes PCI bridge).
- AlphaPC64 (aka Cabriolet), baby AT form-factor.

Other 21164(A) systems: The only one I'm aware of that isn't simply an EB164 clone is a system made by DeskStation. That system is implemented using a memory and I/O controller proprietary to Desk Station. I don't know what their attitude towards Linux is.

8 Bytes and all that stuff

When the Alpha architecture was introduced, it was unique amongst RISC architectures for eschewing 8-bit and 16-bit loads and stores. It supported 32-bit and 64-bit loads and stores (longword and quadword, in Digital's nomenclature). The co-architects (Dick Sites, Rich Witek) justified this decision by citing the advantages:

1. Byte support in the cache and memory sub-system tends to slow down accesses for 32-bit and 64-bit quantities.

2. Byte support makes it hard to build high-speed error-correction circuitry into the cache/memory sub-system.

Alpha compensates by providing powerful instructions for manipulating bytes and byte groups within 64-bit registers. Standard benchmarks for string operations (e.g., some of the Byte benchmarks) show that Alpha performs very well on byte manipulation.

The absence of byte loads and stores impacts some software semaphores and impacts the design of I/O sub-systems. Digital's solution to the I/O problem is to use some low-order address lines to specify the data size during I/O transfers, and to decode these as byte enables. This so-called Sparse Addressing wastes address space and has the consequence that I/O space is non-contiguous (more on the intricacies of Sparse Addressing when I get around to writing it). Note that I/O space, in this context, refers to all system resources present on the PCI and therefore includes both PCI memory space and PCI I/O space.

With the 21164A introduction, the Alpha archtecture was ECO'd to include byte addressing. Executing these new instructions on an earlier CPU will cause an OPCDEC PALcode exception, so that the PALcode will handle the access. This will have a performance impact. The ramifications of this are that use of these new instructions (IMO) should be restricted to device drivers rather than applications code.

These new byte load and stores mean that future support chipsets will be able to support contiguous I/O space.

9 PALcode and all that stuff

This is a placeholder for a section explaining PALcode. I will write it if there is sufficient interest.

10 Porting

The ability of any Alpha-based machine to run Linux is really only limited by your ability to get information on the gory details of its innards. Since there are Linux ports for the E66, EB64+ and EB164 boards, all systems based on the 21066, 21064/APECS or 21164/ALCOR should run Linux with little or no modification. The major thing that is different between any of these motherboards is the way that they route interrupts. There are three sources of interrupts:

- on-board devices
- PCI devices
- ISA devices

All the systems use an Intel System I/O bridge (SIO) to act as a bridge between PCI and ISA (the main I/O bus is PCI, the ISA bus is a secondary bus used to support slow-speed and 'legacy' I/O devices). The SIO contains the traditional pair of daisy-chained 8259s.

Some systems (e.g., the Noname) route all of their interrupts through the SIO and thence to the CPU. Some systems have a separate interrupt controller and route all PCI interrupts plus the SIO interrupt (8259 output) through that, and all ISA interrupts through the SIO.

Other differences between the systems include:

- how many slots they have
- what on-board PCI devices they have
- whether they have Flash or EPROM

11 More Information

All of the DS evaluation boards and motherboard designs are license-free and the whole documentation kit for a design costs about \$50. That includes all the schematics, programmable parts sources, data sheets for CPU and support chipset. The doc kits are available from Digital Semiconductor distributors. I'm not suggesting that many people will want to rush out and buy this, but I do want to point out that the information is available.

Hope that was helpful. Comments/updates/suggestions for expansion to *Neal Crook* `<mailto:neal.crook@reo.mts.digital.com>`.

12 References

[1] `<http://www.research.digital.com/wrl/publications/abstracts/TN-13.html>` Bill Hamburgen, Jeff Mogul, Brian Reid, Alan Eustace, Richard Swan, Mary Jo Doherty, and Joel Bartlett. *Characterization of Organic Illumination Systems*. DEC WRL, Technical Note 13, April 1989.