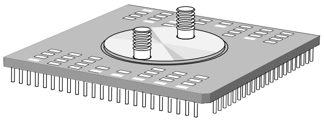
Digital 21064-AA Microprocessor

Product Brief

Preliminary, February 1992

digital™



Features

- Full 64-bit Alpha architecture:
  - Advanced RISC architecture
  - Optimized for high performance implementations
  - Multiprocessor support
  - IEEE single and double precision, VAX F_ floating and G_floating, longword and quadword data types
  - Cycle counter for code optimization
- Privileged Architecture Library Code (PALcode) supports:
  - Optimization for multiple operating systems
  - Flexible memory management implementations
  - Multi-instruction atomic sequences
- Ultra-high performance Alpha implementation:
  - Dual-pipelined architecture
  - 150 MHz cycle time
  - Peak instruction execution of 300 million operations per second
- On-chip write buffer with four 32-byte entries
- Selectable data bus width and speed:
  - 64 or 128 bit data width
  - 75 MHz to 18.75 MHz bus speed
- On-chip pipelined floating point unit
- 8K byte data cache
- 8K byte instruction cache
- External cache memory support:
  - On-chip external secondary cache control
  - Programmable cache size and speed
- On-chip demand paged memory management unit:
  - 12 entry I-stream TB with 8 entries for 8K byte pages and 4 entries for 4M byte pages
  - 32 entry D-stream TB with each entry able to map 8K, 64K, 512K, or 4M byte pages
- On-chip parity and ECC generators and checkers
- Internal clock generator provides:
  - High-speed chip clock
  - Pair of programmable system clocks (CPU/2 to CPU/8)
- Programmable on-chip performance counters measure CPU and system performance
- Chip and module level test support
- 3.3-volt supply voltage
  - Lower power
  - Higher reliability
  - Interface to 5-volt logic

Description

Digital's 21064-AA microprocessor is the first in a family of chips to implement Digital's Alpha architecture. The 21064-AA microprocessor is a .75 micron CMOS based super-scalar super-pipelined processor using dual instruction issue and a 150 MHz cycle time. The Alpha architecture is a 64-bit RISC architecture designed with particular emphasis on speed, multiple instruction issue, multiple processors, and software migration from VAX/VMS and MIPS/ULTRIX.

21064-AA MicroArchitecture

Digital's 21064-AA microprocessor consists of four independent functional units: the integer execution unit (Ebox), floating point unit (Fbox), the load/store or address unit (Abox) and the branch unit. Other sections include the central control unit (Ibox) and the I and D cache.

Ebox - Contains a 64-bit fully pipelined integer execution data path including: adder, logic box, barrel shifter, byte extract and mask, and independent integer multiplier. The Ebox also contains a 32-entry 64-bit integer register file.

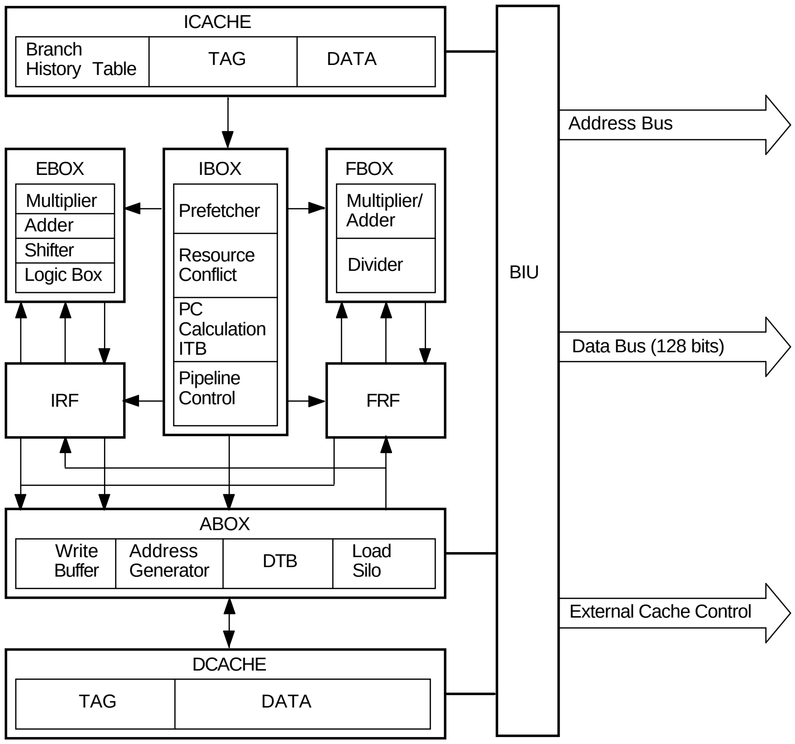
Fbox - Contains a fully pipelined floating point unit and independent divider, supporting both IEEE and VAX floating point data types.

IEEE single precision and double precision floating point data types are supported. VAX F_floating and G_floating data types are fully supported with limited support for the D_floating data type.

Abox - Contains five major sections: address translation data path, load silo, write buffer, data cache (Dcache) interface, and the external bus interface unit (BIU).

The Abox supports all integer and floating point load and store instructions, including address calculation and translation, and cache control logic.

Ibox - Performs instruction fetch, resource checks, and dual instruction issue to the Ebox, Abox, Fbox, or branch unit. In addition, the Ibox controls pipeline stalls, aborts and restarts.



PRELIMINARY

Pipeline Organization

The 21064-AA microprocessor uses a seven stage pipeline for integer operate and memory reference instructions, and a ten stage pipeline for floating point operate instructions. The Ibox maintains state for all pipeline stages to track outstanding register writes.

Cache Organization

The 21064-AA microprocessor contains two on-chip caches, data cache (Dcache) and instruction cache (Icache). The chip also supports an external cache.

Dcache - Contains 8K bytes and is a write through, direct mapped, read-allocate physical cache with 32-byte blocks.

Icache - Contains 8K bytes and is a physical direct-mapped cache with 32-byte blocks.

External Cache - The 21064-AA chip supports external cache built from off-the-shelf static RAMs. The 21064-AA chip directly controls the RAMs using its programmable external cache interface, allowing each implementation to make its own external cache speed and configuration trade-offs.

The external cache interface supports cache sizes from 0 to 8M bytes and a range of operating speeds which are sub-multiples of the chip clock.

Virtual Address Space

The virtual address is a 64-bit unsigned integer that specifies a byte location within the virtual address space. The 21064-AA microprocessor checks all 64-bits of a virtual address and implements a 43-bit subset of the address space. The 21064-AA supports a physical address space of 16G bytes.

Characteristics

Power Supply	Vss 0.0 V, Vdd 3.3 V ±5%
Operating Temperature	T_j max = 85°C
Storage Temperature Range	-55°C to 125°C
Power Dissipation @Vdd = 3.45V Speed = 6.6 ns	23 W typical, 27.5 W maximum

Alpha Architecture Summary

The 21064-AA microprocessor implements the Alpha architecture. The Alpha architecture supports:

- A fixed 32-bit instruction size
- Separate integer and floating point registers
  - 32 64-bit integer registers
  - 32 64-bit floating point registers
- 32-bit (longword) and 64-bit (quadword) integer along with 32-bit and 64-bit IEEE and VAX floating-point data types
- Memory access using a 64-bit virtual byte address
- Privileged Architecture Library Code (PALcode)

Instruction Set

Alpha instructions are all 32 bits in length using four different instruction formats specifying 0, 1, 2, or 3 5-bit register fields. Each format uses a 6-bit opcode.

OP	Number				CALL_PAL
OP	RA	Displacement			Branch
OP	RA	RB	Displacement		Memory
OP	RA	RB	Function	RC	Operate

CALL_PAL Instructions - vector to a privileged library of software that atomically performs both privileged and unprivileged functions.

Branch Instructions - Conditional branch instructions test a register for positive/negative, zero/nonzero, or even/odd, and perform a PC relative branch. Unconditional branch instructions perform either a PC relative or absolute jump using an arbitrary 64-bit register value. They can update a destination register with a return address.

Load/Store Instructions - can move either 32-bit or 64-bit quantities. 8-bit and 16-bit load/store operations are supported through an extensive set of in-register byte manipulations.

Integer Operate Instructions - manipulate full 64-bit values, and include a full complement of arithmetic, compare, logical, and shift instructions. In addition there are three 32-bit integer operates: add, subtract, and multiply.

In addition to the operation of conventional RISC architectures, the Alpha architecture provides scaled add/subtract for quick subscript calculation, 128-bit multiply for division by a constant and multi-precision arithmetic, conditional moves for avoiding branches, and an extensive set of in-register byte manipulation instructions.

Floating-Point Operate Instructions - include four complete sets of instructions for IEEE single, IEEE double, VAX F_floating and VAX G_floating arithmetic. In addition to arithmetic instructions there are also instructions for conversions between floating and integer values including the VAX D_floating data type.

Privileged Architecture Library Code

PALcode is a privileged library of software that atomically performs such functions as the dispatching and servicing of interrupts, exceptions, task switching, and additional privileged and unprivileged user instructions as specified by operating systems using the CALL_PAL instruction.

PALcode is the only method of performing some operations on the hardware. In addition to the entire Alpha instruction set, a set of implementation specific instructions is provided.

PALcode runs in an environment with privileges enabled, instruction stream mapping disabled, and interrupts disabled. Disabling memory mapping allows PALcode to support functions such as TB miss routines. Disabling interrupts allows the instruction stream to provide multi-instruction sequences as atomic operations.

Memory Management

The Alpha memory management architecture is designed to meet several goals:

- Provide a large address space for instructions and data
- Provide convenient and efficient sharing of instructions and data.
- Provide independent read and write access protection
- Provide flexibility through programmable PALcode support

Alpha Architecture Compared to Conventional RISC Architecture

The Alpha architecture is different from conventional RISC architectures in a number of ways:

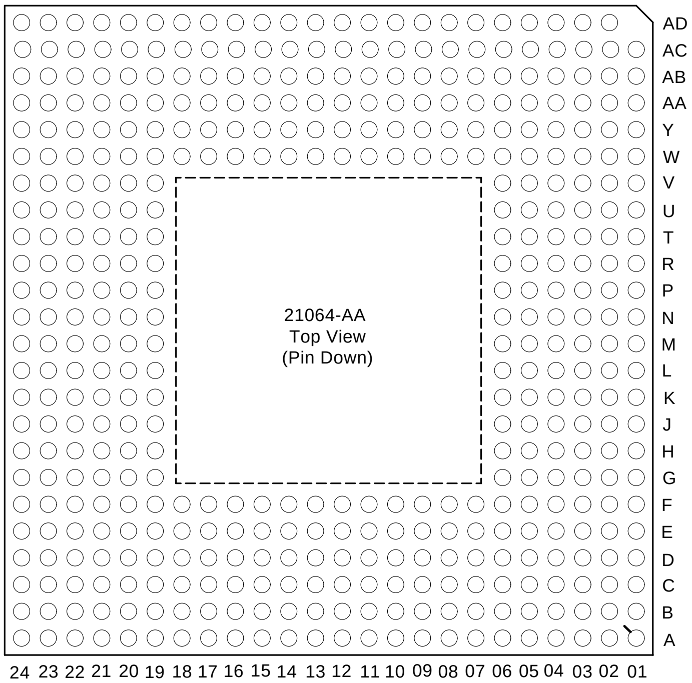
Feature	Difference
64-Bit Architecture	True 64-bit architecture with 64-bit data and address. Not a 32-bit architecture that was later expanded to 64 bits.
High Speed	The Alpha architecture was designed to allow very high-speed implementations. Simple instructions make it particularly easy to build implementations that issue multiple instructions every CPU cycle. There are no implementation specific pipeline timing hazards, no load delay slots, and no branch delay slots.
Multiprocessor Support	The Alpha architecture does not enforce strict read/write ordering between multiple processors. This allows multiprocessor implementations to easily use features such as: multi-bank caches, bypassed write buffers, write merging, and pipelined writes with retry on error. To maintain strict ordering between accesses as seen by a second processor, memory barrier instructions can be explicitly inserted in the program. The basic multi-processor interlocking primitive is a RISC style load_locked, modify, store_conditional sequence. If the sequence runs without interrupt, exception, or an interfering write from another processor, the store succeeds. Otherwise, the store fails and the program eventually must branch back and retry the sequence.
Multiple Operating Systems	The Alpha architecture provides flexibility by allowing the user to implement a privileged library of software for operating system specific operations. This allows Alpha to run full VMS using one version of this software library that mirrors many of the VAX operating system features, and to run OSF/1 using a different version that mirrors many of the MIPS operating system features. Additional operating system implementations can be efficiently supported.
Byte Manipulation	The Alpha architecture is unconventional in the approach to byte manipulation. Byte loads, stores, and operations are done with normal 64-bit instructions, crafted to keep the sequences short. Single-byte stores found in conventional RISC architectures force cache and memory implementations to include hardware byte operations and implement read-modify-write cycles which can complicate system design and reduce performance.
Arithmetic Traps	In contrast to conventional RISC architectures, the reporting of Alpha arithmetic traps (overflow, underflow, and others) are imprecise. This removes architectural bottlenecks that affect performance. If precise arithmetic exceptions are desired, trap barrier instructions can be explicitly inserted in the program to force traps to be delivered at specific points.
HINTS	Alpha includes a number of implementation specific HINTS aimed at allowing higher performance. Software is able to provide HINTS to the hardware that enable the hardware to optimize its operation. HINTS can help improve the utilization of the pipeline, cache memory, and translation lookaside buffers.

Signals

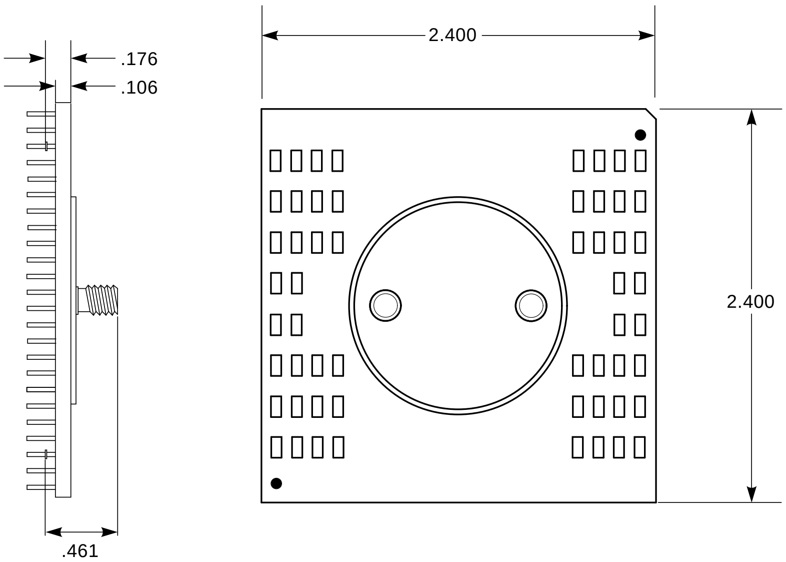
Name	Type	Function
adr_h 33:5	Input/Output	Address bus
data_h 127:0	Input/Output	Data bus
check_h 27:0	Input/Output	Check bit bus
dOE_l	Input	Data bus output enable
dWSel_h 1:0	Input	Data bus write data select
dRAck_h 2:0	Input	Data bus data acknowledge
tagCEOE_h	Output	External cache RAM tagCtl, tagAdr CE/OE
tagCtlWE_h	Output	External cache RAM tagCtl WE
tagCtlV_h	Input/Output	Tag valid
tagCtlS_h	Input/Output	Tag shared
tagCtlD_h	Input/Output	Tag dirty
tagCtlP_h	Input/Output	Tag V/S/D parity
tagAdr_h 33:17	Input	Tag address
tagAdrP_h	Input	Tag address parity
tagOK_h,_l	Input	Tag access from CPU is ok
tagEq_l	Output	Tag compare output
dataCEOE_h 3:0	Output	External cache RAM data CE/OE, longword
dataWE_h 3:0	Output	External cache RAM data WE, longword
dataA_h 4:3	Output	External cache RAM data A 4:3
holdReq_h	Input	Hold request
holdAck_h	Output	Hold acknowledge
cReq_h 2:0	Output	Cycle request
cWMask_h 7:0	Output	Cycle write mask
cAck_h 2:0	Input	Cycle acknowledge
iAdr_h 12:5	Input	Invalidate address, Dcache
dInvReq_h	Input	Invalidate request, Dcache
dMapWE_h	Output	External Dcache duplicate tag RAM WE
irq_h 5:0	Input	Interrupt request
sRomOE_l	Output	Serial ROM output enable
sRomD_h	Input	Serial ROM data/Rx data
sRomclk_h	Output	Serial ROM clock/Tx data
vRef	Input	Input reference
eclOut_h	Input	Output mode selection
perf_cnt_h 1:0	Input	Performance counter inputs
threestate_l	Input	Three state for testing
icMode_h 1:0	Input	Icache Test Mode Selection
cont_l	Input	Continuity for testing
clkIn_h,_l	Input	Clock input
testClkIn_h,_l	Input	Clock input for testing
cpuClkOut_h	Output	CPU clock output
sysClkOut1_h,_l	Output	System clock output, normal
sysClkOut2_h,_l	Output	System clock output, delayed
dcOk_h	Input	Power and clocks ok
reset_l	Input	Reset

Packaging

431 Pin Grid Array



Package Dimensions



PRELIMINARY

Information

For more information on Digital's 21064-AA Microprocessor call:

Voice: **1-800-DEC-2717**

TDD: **1-800-DEC-2515**

Orders may be placed through Digital's Technical OEM (TOEM) Sales Representatives. Call your local Digital Sales Office for details.

digital™

PRELIMINARY