

3 Microarchitecture

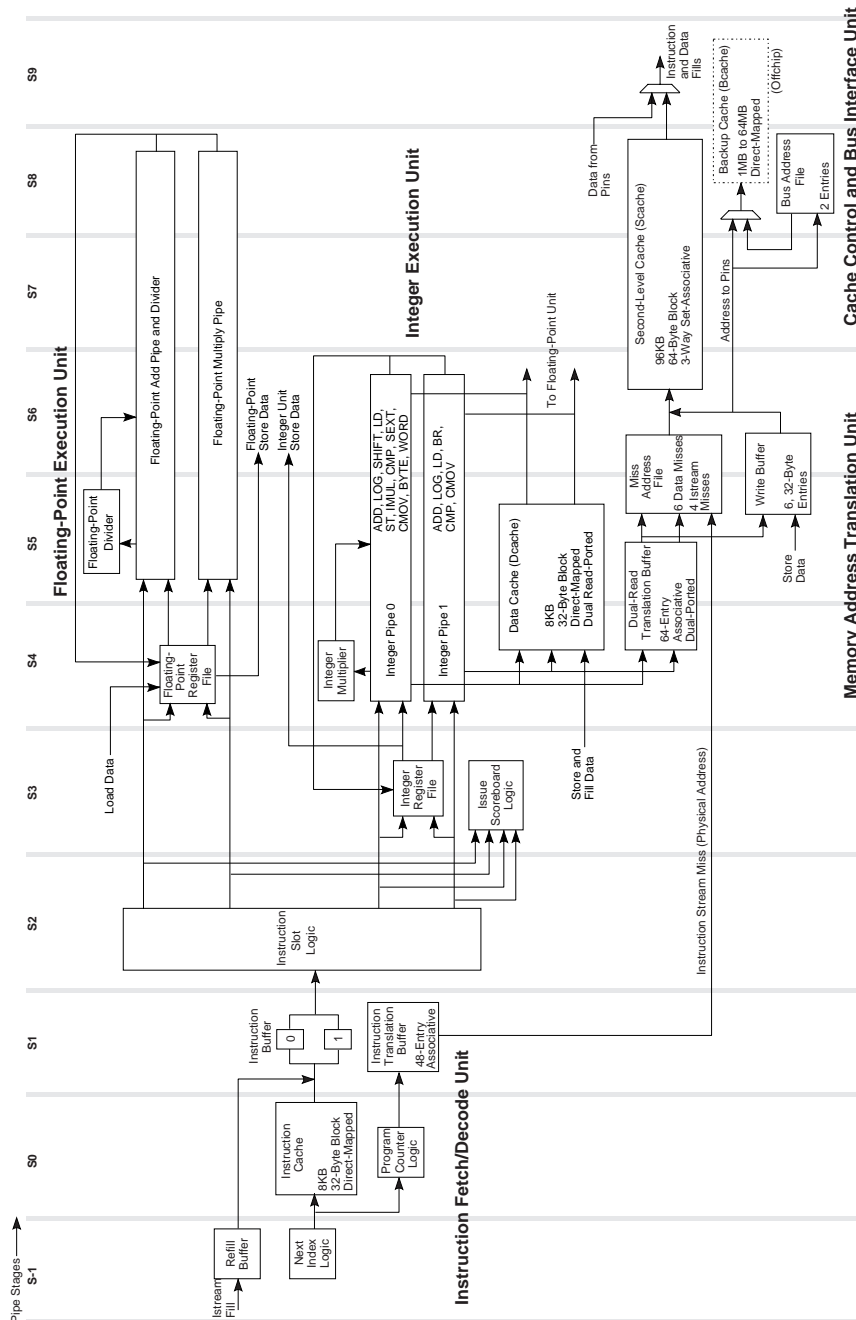
The 21164 microprocessor is a high-performance implementation of Samsung's Alpha architecture. The following sections provide an overview of the chip's architecture and major functional units.

Figure 1 is a block diagram of the 21164.

The 21164 consists of the following sections (see Figure 1):

- Instruction fetch/decode and branch unit (IDU)
- Integer execution unit (IEU)
- Memory address translation unit (MTU)
- Cache control and bus interface unit (CBU)
- Floating-point execution unit (FEU)
- Data cache (Dcache)
- Instruction cache (Icache)
- Secondary cache (Scache)
- Serial read-only memory (SRAM) interface

Figure 1 21164 Microprocessor Block/Pipe Flow Diagram



LJ-05383.A14

Instruction Fetch/Decode and Branch Unit

3.1 Instruction Fetch/Decode and Branch Unit

The primary function of the instruction fetch/decode and branch unit (IDU) is to manage and issue instructions to the IEU, MTU, and FEU. It also manages the instruction cache. The IDU contains:

- Prefetcher and instruction buffer
- Instruction slot and issue logic
- Program counter (PC) and branch prediction logic
- 48-entry instruction translation buffers (ITBs)
- Abort logic
- Register conflict logic
- Interrupt and exception logic

3.1.1 Instruction Prefetch and Decode

The IDU handles only NATURALLY ALIGNED groups of four instructions (INT16). The IDU does not advance to a new group of four instructions until all instructions in a group are issued. If a branch to the middle of an INT16 group occurs, then the IDU attempts to issue the instructions from the branch target to the end of the current INT16; the IDU then proceeds to the next INT16 of instructions after all the instructions in the target INT16 are issued. Thus, proper code scheduling is required to achieve optimal performance.

3.1.2 Branch Prediction

The branch unit, or prediction logic, is also part of the IDU. Branch and PC prediction are necessary to predict and begin fetching the target instruction stream before the branch or jump instruction is issued. Each instruction location in the instruction cache (Icache) contains a 2-bit history state to record the outcome of branch instructions.

3.1.3 Instruction Translation Buffer

The IDU includes a 48-entry, fully associative instruction translation buffer (ITB). The buffer stores recently used instruction stream (Istream) address translations and protection information for pages ranging from 8KB to 512KB and uses a not-last-used replacement algorithm.

Integer Execution Unit

The 21164 provides two optional translation extensions called superpages. Access to superpages is allowed only while executing in privileged mode.

- One superpage maps virtual address bits <39:13> to physical address bits <39:13>, on a one-to-one basis, when virtual address bits <42:41> equal 2.
- The other superpage maps virtual address bits <29:13> to physical address bits <29:13>, on a one-to-one basis, and forces physical address bits <39:30> to 0 when virtual address bits <42:30> equal 1FFE(hex).

3.1.4 Interrupts

The IDU exception logic supports three sources of interrupts:

- Hardware interrupts

There are seven level-sensitive hardware interrupt sources supplied by the following signals:

irq_h<3:0>
sys_mch_chk_irq_h
pwr_fail_irq_h
mch_halt_irq_h

- Software interrupts

There are 15 prioritized software interrupts sourced by an onchip internal processor register (IPR).

- Asynchronous system traps

There are four asynchronous system traps (ASTs) controlled by onchip IPRs.

Most interrupts can be independently masked in onchip enable registers. In addition, AST interrupts are qualified by the current processor mode. All interrupts are disabled when the processor is executing PALcode.

3.2 Integer Execution Unit

The integer execution unit (IEU) contains two 64-bit integer execution pipelines—E0 and E1, which include the following:

- Two adders
- Two logic boxes
- A barrel shifter
- Byte-manipulation logic

Floating-Point Execution Unit

- An integer multiplier

The IEU also includes the 40-entry, 64-bit integer register file (IRF) that contains the 32 integer registers defined by the Alpha architecture and 8 PALshadow registers. The register file has four read ports and two write ports, which provide operands to both integer execution pipelines and accept results from both pipes. The register file also accepts load instruction results (memory data) on the same two write ports.

3.3 Floating-Point Execution Unit

The onchip, pipelined floating-point unit (FPU) can execute both IEEE and VAX floating-point instructions. The 21164 supports IEEE S_floating and T_floating data types, and all rounding modes. It also supports VAX F_floating and G_floating data types, and provides limited support for the D_floating format. The FPU contains:

- A 32-entry, 64-bit floating-point register file (FRF).
- A user-accessible control register.
- A floating-point multiply pipeline.
- A floating-point add pipeline—The floating-point divide unit is associated with the floating-point add pipeline but is not pipelined.

The FPU can accept two instructions every cycle, with the exception of floating-point divide instructions. The result latency for nondivide, floating-point instructions is four cycles.

3.4 Memory Address Translation Unit

The memory address translation unit (MTU) contains three major sections:

- Data translation buffer (dual ported)
- Miss address file (MAF)
- Write buffer address file

The MTU receives up to two virtual addresses every cycle from the IEU. The translation buffer generates the corresponding physical addresses and access control information for each virtual address. The 21164 implements a 43-bit virtual address and a 40-bit physical address.

The MTU can perform read and write operations to/from memory on byte, word, longword, and quadword boundaries.

Memory Address Translation Unit

3.4.1 Data Translation Buffer

The 64-entry, fully associative, dual-read-ported data translation buffer (DTB) stores recently used data stream (Dstream) page table entries (PTEs). Each entry supports all four granularity hint-bit combinations, so that a single DTB entry can provide translation for up to 512 contiguously mapped, 8KB pages.

The DTB also supports the register-enabled superpage extension. The DTB superpage maps provide virtual-to-physical address translation for two regions of the virtual address space.

3.4.2 Miss Address File

The MTU begins the execution of each load instruction by translating the virtual address and by accessing the data cache (Dcache). Translation and Dcache tag read operations occur in parallel. If the addressed location is found in the Dcache (a hit), then the data from the Dcache is formatted and written to either the integer register file (IRF) or floating-point register file (FRF). The formatting required depends on the particular load instruction executed. If the data is not found in the Dcache (a miss), then the address, target register number, and formatting information are entered in the miss address file (MAF).

The MAF performs a load-merging function. When a load miss occurs, each MAF entry is checked to see if it contains a load miss that addresses the same Dcache (32-byte) block. If it does, and certain merging rules are satisfied, then the new load miss is merged with an existing MAF entry. This allows the MTU to service two or more load misses with one data fill from the CBU.

There are six MAF entries for load misses and four more for IDU instruction fetches and prefetches. Load misses are usually the highest MTU priority.

3.4.3 Store Execution

The Dcache follows a write-through protocol. During the execution of a store instruction, the MTU probes the Dcache to determine whether the location to be overwritten is currently cached. If so (a Dcache hit), the Dcache is updated. Regardless of the Dcache state, the MTU forwards the data to the CBU.

A load instruction that is issued one cycle after a store instruction in the pipeline creates a conflict if both the load and store operations access the same memory location. (The store instruction has not yet updated the location when the load instruction reads it.) This conflict is handled by forcing the load instruction to take a replay trap; that is, the IDU flushes the pipeline and restarts execution from the load instruction.

Cache Control and Bus Interface Unit

By the time the load instruction arrives at the Dcache the second time, the conflicting store instruction has written the Dcache and the load instruction is executed normally.

Replay traps can be avoided by scheduling the load instruction to issue three cycles after the store instruction. If the load instruction is scheduled to issue two cycles after the store instruction, then it will be issue-stalled for one cycle.

3.4.4 Write Buffer

The MTU also contains a write buffer that has six 32-byte entries. The write buffer provides a finite, high-bandwidth resource for receiving store data to minimize the number of CPU stall cycles.

3.5 Cache Control and Bus Interface Unit

The cache control and bus interface unit (CBU) processes all accesses sent by the MTU and implements all memory-related external interface functions, particularly the coherence protocol functions for write-back caching. It controls the second-level cache (Scache) and the optional board-level backup cache (Bcache). The CBU handles all instruction and primary Dcache read misses, performs the function of writing data from the write buffer into the shared coherent memory subsystem, and has a major role in executing the Alpha memory barrier (MB) instruction. The CBU also controls the 128-bit bidirectional data bus, address bus, and I/O control.

3.6 Cache Organization

The 21164 has three onchip caches—a primary L1 data cache, a primary L1 instruction cache, and a second-level L2 combined data and instruction cache. All memory cells in the onchip caches are fully static, 6-transistor, CMOS structures.

The 21164 also provides control for an optional board-level, external L3 cache.

3.6.1 Data Cache

The data cache (Dcache) is a dual-read-ported, single-write-ported, 8KB cache. It is a write-through, read-allocate, direct-mapped, physical cache with 32-byte blocks.

3.6.2 Instruction Cache

The instruction cache (Icache) is an 8KB, virtual, direct-mapped cache with 32-byte blocks. Each block tag contains:

Serial Read-Only Memory Interface

- A 7-bit address space number (ASN) field as defined by the Alpha architecture
- A 1-bit address space match (ASM) field as defined by the Alpha architecture
- A 1-bit PALcode (physically addressed) indicator

Software, rather than Icache hardware, maintains Icache coherence with memory.

3.6.3 Second-Level Cache

The second-level cache (Scache) is a 96KB, 3-way, set-associative, physical, write-back, write-allocate cache with 32-byte or 64-byte blocks. It is a mixed data and instruction cache. The Scache is fully pipelined; it processes read and write operations at the rate of one INT16 per CPU cycle and can alternate between read and write accesses without bubble cycles.

When operating in 32-byte block mode, the Scache has 64-byte blocks with 32-byte subblocks, one tag per block. If configured to 32 bytes, the Scache is organized as three sets of 512 blocks, with each block divided into two 32-byte subblocks. If configured to 64 bytes, the Scache is three sets of 512 64-byte blocks.

3.6.4 External Cache

The CBU implements control for an optional, external, direct-mapped, physical, write-back, write-allocate cache with 32-byte or 64-byte blocks. The 21164 supports board-level cache sizes of 1MB, 2MB, 4MB, 8MB, 16MB, 32MB, and 64MB.

3.7 Serial Read-Only Memory Interface

The serial read-only memory (SROM) interface provides the initialization data load path from a system SROM to the instruction cache. Following initialization, this interface can function as a diagnostic port by using privileged architecture library code (PALcode).

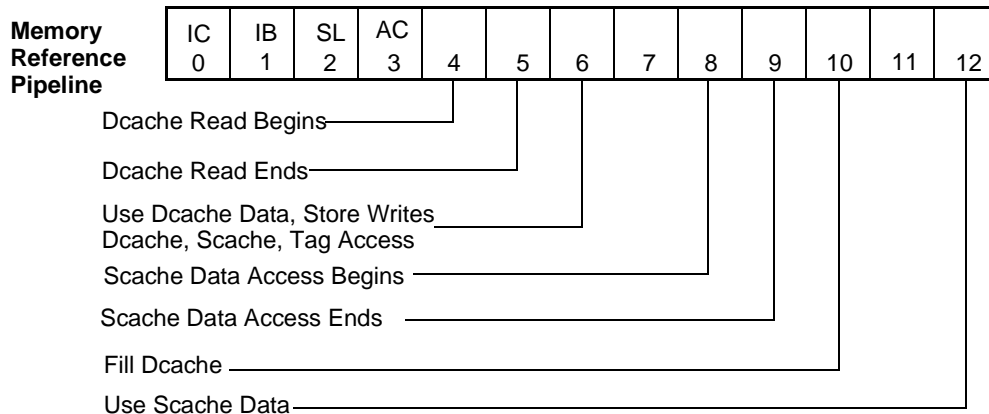
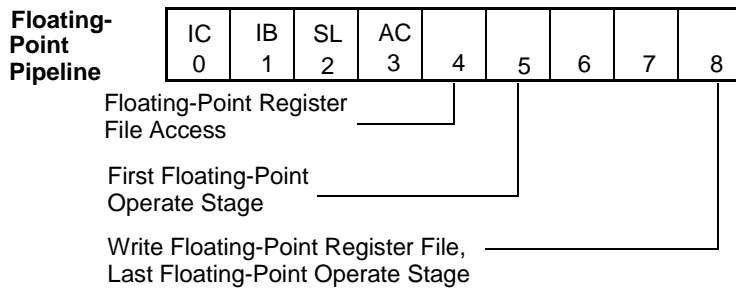
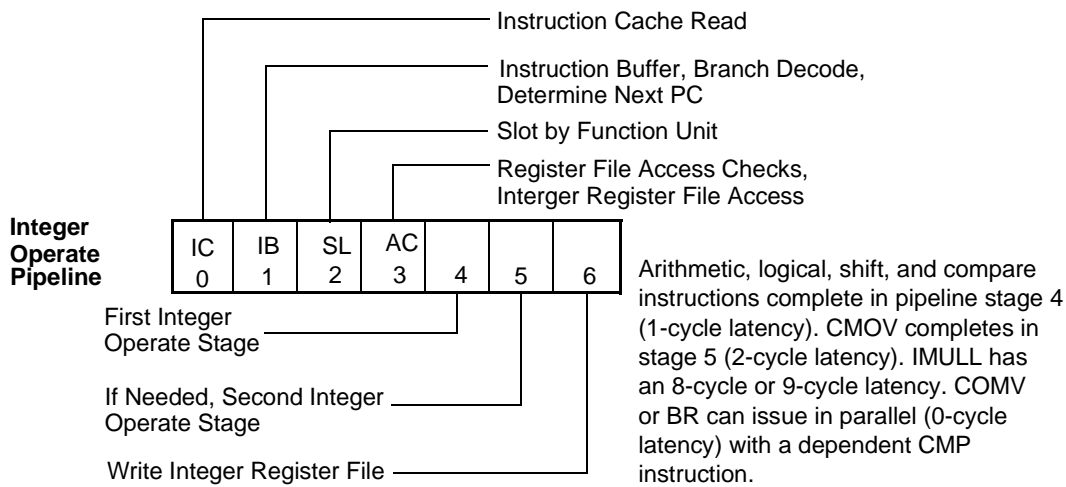
3.8 Pipeline Organization

The 21164 has a 7-stage (or 7-cycle) pipeline for integer operate and memory reference instructions, and a 9-stage pipeline for floating-point operate instructions. The IDU maintains state for all pipeline stages to track outstanding register write operations.

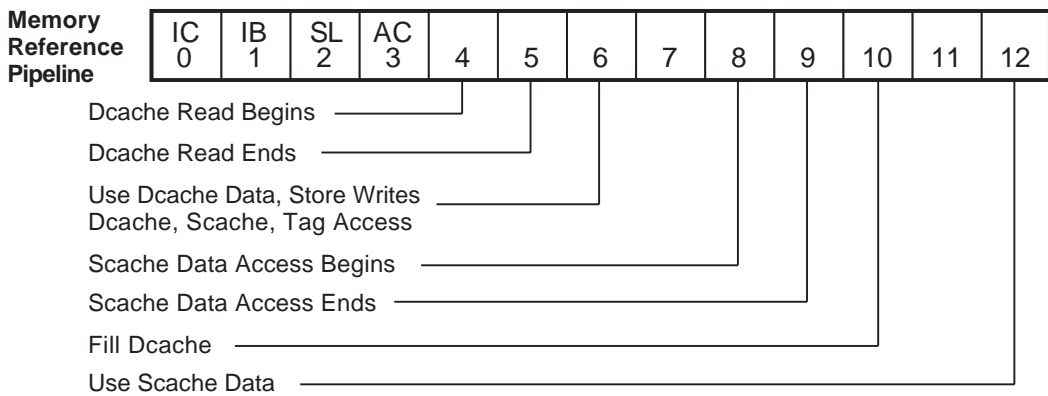
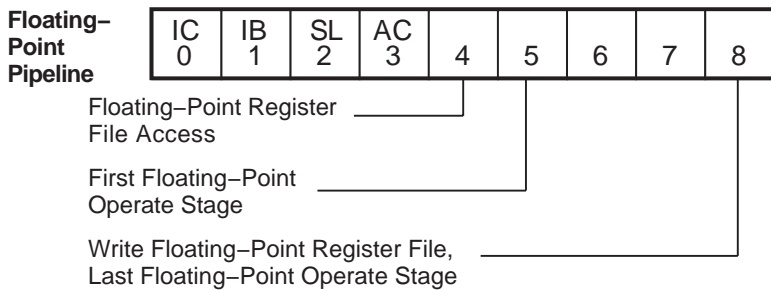
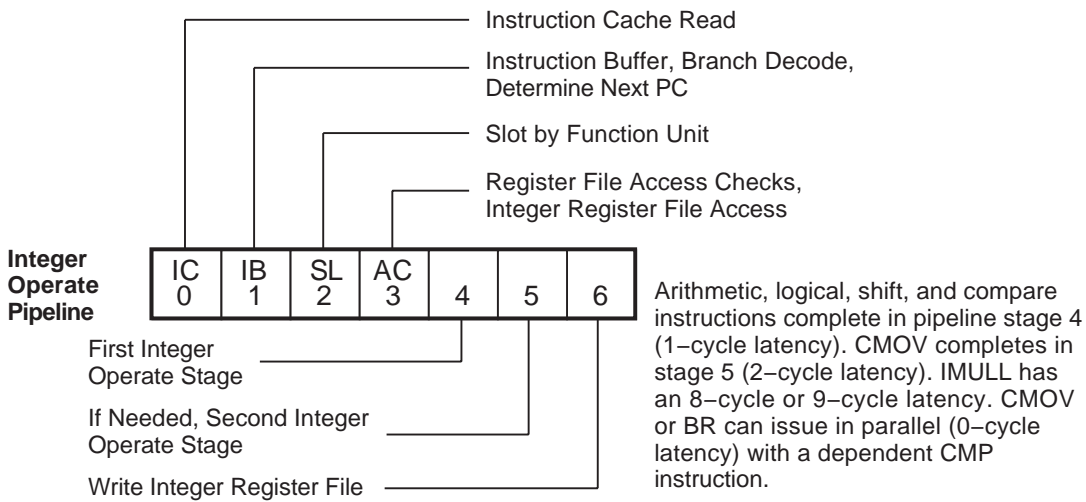
Figure 2 shows the integer operate, memory reference, and floating-point operate pipelines for the IDU, FPU, IEU, and MTU. The first four stages are executed in the IDU. Remaining stages are executed by the IEU, FEU, MTU, and CBU.

Pipeline Organization

Figure 2 Instruction Pipeline Stages



Pipeline Organization



HLO019