1. PowerPC 970MP Overview

The IBM PowerPC 970MP reduced instruction set computer (RISC) microprocessor is an implementation of the PowerPC Architecture. This chapter provides an overview of the features of the 970MP microprocessor and includes two block diagrams showing the major functional components.

Note: The 970MP microprocessor incorporates two complete microprocessors on a single chip, along with some common logic to connect these microprocessors to a system. The terms microprocessor, processor, and processing unit are used interchangeably to describe each of the two individual processors. The term core refers to the instruction fetch and execution logic, including the L1 cache, but excluding the storage subsystem, of each processor. The term PowerPC 970MP or 970MP refers to the single chip module comprising the two processing units and the common logic.

1.1 PowerPC 970MP Microprocessor Overview

The 970MP microprocessor is a dual core, 64-bit PowerPC RISC microprocessor with vector processing unit (VPU) extensions—the single-instruction, multiple-data (SIMD) operations that accelerate data intensive processing tasks. This processor is designed to support multiple system configurations ranging from desktop and low-end server applications, up through 4-way symmetric multiprocessor (SMP) configurations.

Each processing unit of the IBM PowerPC 970MP RISC Microprocessor consists of three main components:

- The core, which includes the VPUs
- The storage subsystem (STS), which includes the core interface logic, noncacheable unit, L2 cache and controls, and bus interface unit
- Pervasive functions

The block diagram in *Figure 1-1* on page 34 shows the major functional units comprising the core and storage subsystem. In the core, these units include instruction fetch, decode and dispatch units, plus the register files and execution units. The storage subsystem includes the second level (L2) cache and interface units.

The block diagram in *Figure 1-2* on page 35 shows how the two processing units (PU0 and PU1) are connected through the common logic to the processor interface.

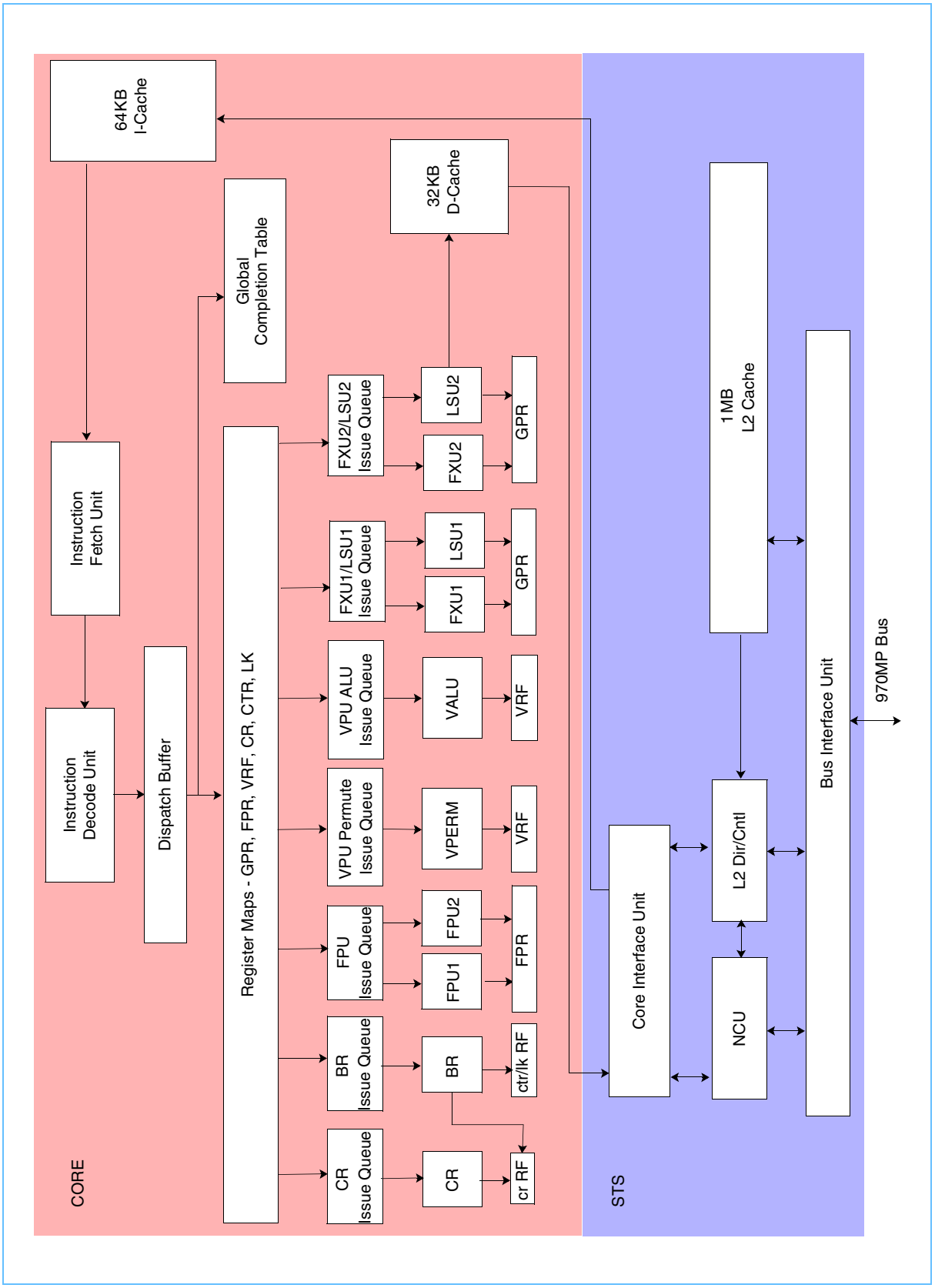
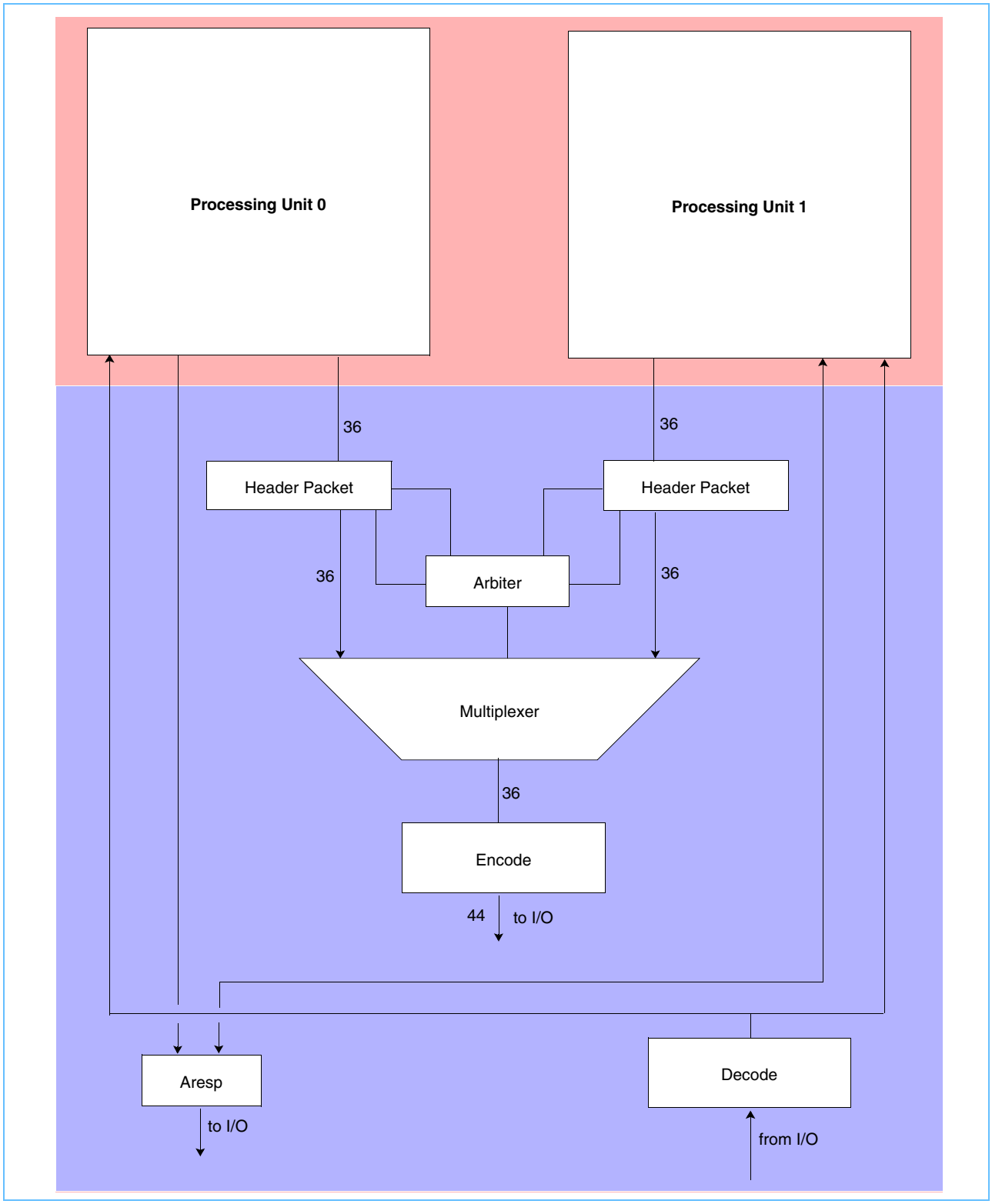


Figure 1-1. 970MP Block Diagram

Figure 1-2. 970MP Dual Core with Common Arbitration Logic



1.2 PowerPC 970MP Functional Units

1.2.1 Introduction

This section provides an overview of the 970MP microprocessor core, VPU, storage, and bus interface units of each processing unit. It includes a summary and details of key design fundamentals.

1.2.1.1 Key Design Fundamentals of the Microprocessor Core

- 64-bit implementation of the PowerPC Architecture (version 2.01)
  - Binary compatibility with all PowerPC Architecture application level code (user [problem] state)
  - Support for the 32-bit operating system bridge facility
  - Vector/SIMD multimedia extension
- Layered implementation strategy for very high-frequency operation
  - Deeply pipelined design
    - 16 stages for most fixed-point register-to-register operations
    - 18 stages for most load-and-store operations (assuming an L1 D-cache hit)
    - 21 stages for most floating-point operations
    - 19 stages for fixed-point, 22 stages for complex-fixed, and 25 stages for floating-point operations in the vector arithmetic logic unit (VALU)
    - 19 stages for VPU permute operations
  - Dynamic instruction cracking[1] for some instructions allows for simpler inner core dataflow
    - Dedicated dataflow for cracking one instruction into two internal operations
    - Microcoded templates for longer emulation sequences
- Speculative superscalar inner core organization
  - Aggressive branch prediction
    - Prediction for up to two branches per cycle
    - Support for up to 16 predicted branches in flight
    - Prediction support for branch direction and branch addresses
  - In-order dispatch of up to five operations into the distributed issue queue structure
  - Out-of-order issue of up to 10 operations into 10 execution pipelines
    - Two load or store operations
    - Two fixed-point register-to-register operations
    - Two floating-point operations
    - One branch operation
    - One Condition Register operation
    - One VPU permute operation
    - One VPU arithmetic logic unit (ALU) operation
  - Register renaming on General Purpose Registers (GPRs), Floating-Point Registers (FPRs), Vector Registers (VRs), Condition Register (CR) fields, two bits of the Integer Exception Register (XER), Floating-Point Status and Control Register (FPSCR), the Vector Save/Restore Register (VRSAVE), Vector Status and Control Register (VSCR), Link Register (LR), and Count Register (CTR)

1. Process by which some complex instructions are broken into two simpler, more RISC-like instructions.

- Large number of instructions in flight (theoretical maximum of 215 instructions)
  - Up to 16 instructions in the instruction fetch unit (fetch buffer and overflow buffer)
  - Up to 32 instructions in the instruction fetch buffer in the instruction decode unit
  - Up to 35 instructions in three decode pipe stages and four dispatch buffers
  - Up to 100 instructions in the inner core (after dispatch)
  - Up to 32 stores queued in the store queue (STQ) (available for forwarding)
  - Fast, selective flush of incorrect speculative instructions and results
- Specific focus on storage latency management
  - Out-of-order and speculative issue of load operations
  - Support for up to eight outstanding L1 cache line misses
  - Hardware-initiated instruction prefetching from the L2 cache
  - Software-initiated data stream prefetching with support for up to eight active streams
  - Critical word forwarding—critical sector first
  - New branch processing—Prediction hints on branch instructions
- Power management
  - Static power management
    - Software initiated Doze, Nap, and Deep Nap low-power modes
  - Dynamic power management
    - Parts of the design stop their clocks when not in use under hardware control
  - Power tuning through frequency scaling
    - Software initiated slow down of the processor; selectable to a half or quarter of the nominal operating frequency
    - Programmable latency for power mode transitions to control current spikes

1.2.1.2 Detailed Features of the Microprocessor Core

- Instruction fetching and branch prediction
  - 64KB, direct-mapped instruction cache (I-cache)
    - 128-byte lines (broken into four 32-byte sectors)
    - Dedicated 32-byte read/write interface from the L2 cache with a critical-sector-first reload policy
    - Effective-address index, real address tags
    - Cache supports one read or one write per cycle
    - Five additional predecode bits per word to aid in fast decoding and group formation
    - Parity protected with a force invalidate and reload on parity error
  - 128 total entries in the effective-to-real-address translation (ERAT) cache; 2-way, set-associative
    - Organization is 64 entries by two ways
    - Each entry translates 4 KB (no large page support; large pages take multiple entries)
  - 4-entry, 128-byte, instruction prefetch queue above the I-cache; hardware-initiated prefetches
  - Fetch a 32-byte aligned block of eight instructions per cycle
  - Branch prediction
    - Scan all eight fetched instructions for branches each cycle
    - Predict up to two branches per cycle
    - 3-table prediction structure: global, local, and selector (16 K entries x 1 bit each)
    - 16-entry link stack for address prediction (with stack recovery)
    - 32-entry count cache for address prediction (indexed by the address of the Branch Conditional to Count Register [**bcctr**] instructions)

- Instruction decode and preprocessing
  - 3-cycle pipeline to decode and preprocess instructions
    - Dedicated dataflow for cracking one instruction into two internal operations
    - Microcoded templates for longer emulation sequences of internal operations
    - All internal operations expanded into 86-bit internal form to simplify subsequent processing and explicitly expose register dependencies for all register pools
    - Dispatch groups (up to five instructions) formulated along with inter-instruction dependence masks
  - Cracked and microcoded instructions have access to four renamed emulation GPRs (eGPRs), one renamed emulation FPR (eFPR), and one renamed emulation CR (eCR) field (in addition to architected facilities)
  - 8-entry (16 bytes per entry) instruction fetch buffer (up to eight instructions in and five instructions out during each cycle)
  - Microcode patch facility allows most instructions other than branches to trap to microcode, which can be programmed to either emulate the effects of the instruction or cause an interrupt.
- Instruction dispatch, sequencing, and completion control
  - Four dispatch buffers, which can hold up to four dispatch groups when the global completion table (GCT) is full
  - 20-entry global completion table
    - Group-oriented tracking associates a 5-operation dispatch group with a single GCT entry
    - Tracks internal operations from dispatch to completion for up to 100 operations
    - Capable of restoring the machine state for any of the instructions in flight
      - — Very fast restoration for instructions on group boundaries (that is, branches)
      - — Slower for instructions contained within a group
  - Supports precise exceptions (including machine check interrupt)
  - Register renaming resources
    - 80-entry GPR rename mapper (32 architected GPRs plus four eGPRs and VRSAVE)
    - 80-entry FPR rename mapper (32 architected FPRs plus one eFPR)
    - 80-entry Vector Register file (VRF) rename mapper (32 architected VRFs)
    - 24-entry XER rename mapper (the XER is broken into mappable and nonmappable fields)
      - — Two mappable fields: ov and ca
      - — Nonmappable field: string-count
    - 16-entry LR/CTR rename mapper (one architected LR and one architected CTR)
    - 32-entry CR rename mapper (eight architected CR fields plus one eCR field)
    - 20-entry FPSCR rename mapper
    - VRSAVE
    - VSCR
  - Instruction queuing resources:
    - Two 18-entry issue queues for fixed-point and load/store instructions
    - Two 10-entry issue queues for floating-point instructions
    - 12-entry issue queue for branch instructions
    - 10-entry issue queue for CR-logical instructions
    - 16-entry issue queue for vector permute instructions
    - 20-entry issue queue for vector ALU instructions and VPU stores

- Two fixed-point execution pipelines
  - Both capable of basic arithmetic, logical, and shifting operations
  - Both capable of multiplies
  - One capable of divides; the other capable of SPR operations
  - Out-of-order issue with bias towards oldest operations first
  - Symmetric forwarding between fixed-point and load/store execution pipelines
- Load/store execution pipelines
  - Two 6-stage load/store execution pipelines
  - Out-of-order issue with bias towards oldest operations first
    - Stores issue twice—an address generation operation (load/store), and a data steering operation (FXU/FPU/VPU)
  - 32KB, 2-way, set-associative D-cache
    - Triple ported to support two reads and one write every cycle (no banking)
    - 2-cycle load-use penalty for FXU loads
    - 4-cycle load-use penalty for FPU loads
    - 3-cycle load-use penalty for loads to vector permute unit (VPERM)
    - 4-cycle load-use penalty for loads to VALU
    - Store-through policy; no allocation on store misses
    - 128-byte cache line
    - Least recently used (LRU) replacement policy
    - Dedicated 32-byte reload interface from the L2 cache
    - Effective-address index, real address tags (hardware fix up on alias cases)
    - Parity protected; precise machine check interrupt on parity error; software fix if HID5[50] equals '1'. Otherwise, recovery is done by hardware (default).
  - 128-entry (total) ERAT cache, 2-way, set-associative
    - Organization is 64 entries by two ways
    - Each entry translates 4 KB (no large page support; large pages take multiple entries)
  - 32-entry store queue logically above the D-cache (real address based; content-addressable memory [CAM] structure)
    - Store addresses and store data can be supplied on different cycles
    - Stores wait in this queue until they are completed; then they write the cache
    - Supports store forwarding to inclusive subsequent loads (even if both are speculative)
  - 32-entry load reorder queue (real address based; CAM structure)
    - Keeps track of out-of-order loads and watches for hazards
      - Previous store to the same address that gets executed after the load causes a flush
      - Previous load from the same address when a cross-invalidate has occurred causes a flush
  - 8-entry load miss queue (LMQ) (real address based)
    - Keeps track of loads that have missed in the L1 D-cache
    - Allows a second load from the same cache line to merge onto a single entry

- Branch and Condition Register execution pipelines
  - One branch execution pipeline
    - Computes actual branch address and branch direction for comparison with prediction
    - Redirects instruction fetching if either prediction was incorrect
    - Assists in training and maintaining the branch table predictors, the link stack, and the count cache
  - One Condition Register logical pipeline
    - Executes CR logical instructions and the CR movement operations
    - Executes some Move To Special Purpose Register (**mtspr**) and Move From Special Purpose Register (**mfspr**) instructions also
  - Out-of-order issue with a bias towards oldest operations first
- Floating-point execution pipelines
  - Two 9-stage floating-point execution pipelines (6-stage execution)
    - Both capable of the full set of floating-point instructions
    - All data formats supported in hardware (no floating-point assist interrupts)
  - Out-of-order issue with bias towards oldest operations first
  - Symmetric forwarding between the floating-point pipelines
  - No support for the non-IEEE mode
- VPU execution pipelines
  - Two dispatchable units:
    - VALU contains three subunits:
      - Vector simple fixed: 1-stage execution
      - Vector complex fixed: 4-stage execution
      - Vector floating point: 7-stage execution
    - VPERM: 1-stage execution
  - Out-of-order issue with a bias towards oldest operations first
  - Symmetric forwarding between the permute and VALU pipelines
- Unified second-level memory management (address translation)
  - 1024-entry, 4-way, set-associative translation lookaside buffer (TLB)
    - Supports new large page architecture (16MB large pages supported)
    - Hardware-based reload (from the L2 cache interface; no L1 D-cache impact)
    - Hardware-based update of the referenced (R) and changed (C) bits in a page table entry (PTE)
    - Parity protected; precise machine check interrupt on parity error (software fix up)
  - 64-entry fully associative segment lookaside buffer (SLB)
    - SLB miss results in an interrupt; software reload of the SLB
    - SLB can also be loaded by the 32-bit PowerPC Segment Register instructions
  - Supports a 65-bit virtual address and a 42-bit real address
- Data stream prefetch
  - Eight data prefetch streams supported in hardware. Eight hardware streams are only available if VPU prefetch instructions are disabled.
  - Four VPU prefetch streams supported using four of the eight hardware streams. The VPU prefetch mapping algorithm supports most commonly used forms of vector prefetch instructions.