EXHIBIT X

XP 000354320

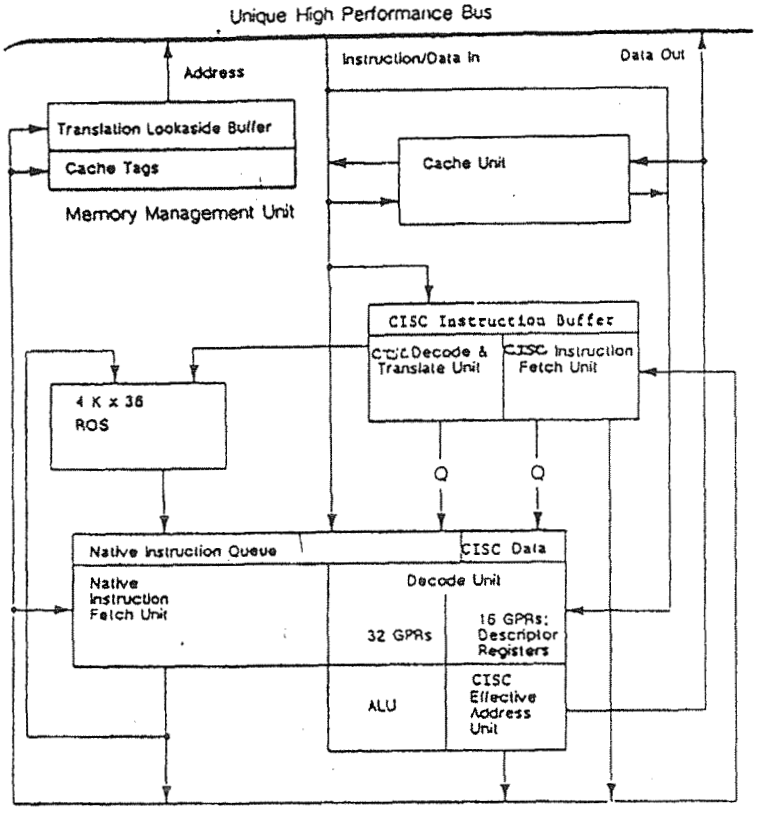
losure Bulletin Vol. 36 No. 02 February 1993

p.231-234

700
Armonk, NY, US

G06F9/30E

High Performance Dual Architecture Processor



RISC/CISC Block Diagram

Disclosed is the IBM RISC/CISC, a unique processor that exhibits exter- nally two major architectures. It can appear as:

- A high performance Reduced Instruction Set Computer (RISC) This Native manifestation of the RISC/CISC provides the typical RISC concepts of a User Mode and Privileged Mode.
- The CISC manifestation is a 100% compatible emulation of an industry standard CISC processor provided by a combination of hardware and microcode; the microcode uses the Native architecture, plus extensions, to emulate the CISC. The RISC/CISC emulation performs at least as well as the real industry standard processor.

The RISC/CISC processor provides the ability to run software compatible with both aspects of the processor. Thus, it provides a migration path from the CISC environment to the

emulation

RISC environment. It does so without software emulation of a CISC processor by a RISC proces- sor, a solution with severe performance problems.

The important aspects of the RISC/CISC are:

- Full 32-bit architecture
- 64, 32-bit General Purpose Registers (architecturally)
- Several Special Purpose Registers
- Pipelined instruction processing
- Internal instruction and data cache
- Virtual memory support
- Demand paging support
- Translator for CISC emulation
- Internal ROS for CISC emulation

The figure shows a block diagram of the RISC/CISC. It contains the following major components:

- MMU - The Memory Management Unit contains various registers and control logic for virtual to real address translation and cache management.
- Cache Unit - The cache unit contains a small on-chip cache, cache tags for a larger, off-chip cache and cache management logic.
- CISC Instruction Buffer - The CISC Instruction Buffer accepts instruction fetch results, and allows sophisticated manipulation of the information in the buffer to facilitate CISC instruction decoding.
- CISC Decode and Translate Unit - The CISC Decode and Translate Unit decodes the data in the CISC Instruction Buffer into complete CISC instructions, and then translates the instruction into:

  a Native instruction - the only instruction, or the first instruction of a sequence required to implement the CISC instruction
  a ROS address - for the rest of an instruction sequence, if any
  CISC data - from various CISC instruction fields, and state information kept by the Translation Unit, both used by subse- quent units to implement CISC instructions

- CISC Instruction Fetch Unit - The CISC Instruction Fetch Unit is responsible for calculating CISC instruction addresses and fetching CISC instructions.
- ROS - The ROS contains instructions for implementing complex CISC instructions and for implementing some microcoded Native instructions.
- RISC/CISC Instruction Queue - The RISC/CISC Instruction Queue accepts instructions from the bus, the cache, ROS and the Translator. It feeds the instructions to the Decode Unit.

- Decode Unit - The Decode Unit decodes Native instructions, addressing the registers as required, to feed control and data to the ALU and CISC Effective Address Unit.
- ALU - The ALU contains a fixed point arithmetic unit, and a sophisticated logical unit. These work on bytes, half words and words.
- CISC Effective Address Unit - The CISC Effective Address Unit calculates CISC addresses like the CISC, with a based, index and displacement, and then produces a CISC Linear Address (the EA plus a segment base).

To exhibit the different external architectures, the RISC/CISC runs in several modes that present different environments by allowing:

- execution of some (proper) subset of the RISC/CISC instructions
- access to some (proper) subset of the RISC/CISC registers.

The modes are:

- Native User Mode - provides the Native User environment, for running Native applications
- Privileged Mode - provides the Native Privileged environment (Native User environment plus Privileged extensions), for implementing operating systems that support Native applications
- Emulator Mode - provides an environment for CISC instruction emulation (Native Privileged environment plus Emulator Mode extensions)

Native Mode exhibits the following 32-bit RISC processor character- istics:

32, 32-bit General Purpose Registers (GPRs)
Reference to storage only through load and store instructions
Three operand instructions
Interrupt handling
Storage control

In addition to the "standard" instructions in a RISC processor for arithmetic, shifting and rotating, logical operations and branching, the RISC/CISC Native Mode exhibits some discriminating features:

Powerful rotate and merge instructions
Multiply and divide instructions
Support for branch prefetch optimization
Support for branch "scheduling" optimization

Emulator Mode extends Native Mode in several areas; the extensions allow high performance CISC emulation.

- Additional registers - The additional registers provide additional high speed general purpose registers, CISC-like status and storage control registers and various special purpose registers.
- Additional register addressing capability - This capability permits addressing of some special purpose registers as general purpose registers and indirect addressing of general purpose and some special purpose registers.
- Additional operand sizes - This capability allows the RISC/CISC to work efficiently with 8- and 16-bit operands as well as 32-bit operands.

- Additional instructions - Emulation requires various new instructions for accessing non-Native registers, and to implement other functions not possible or that perform poorly using only Native instructions. An example of the latter category is the set of descriptor based storage access instructions. These form addresses for storage accesses like the CISC, with base and index registers, a displacement and a segment base; Native instructions cannot access the registers that contain descriptors, and implementation would require many Native instructions to do the same task that a single cycle descriptor based instruction can accomplish.
- CISC condition code setting - This capability allows the RISC/CISC to set CISC condition codes after various operations.
- Hidden Address Space Access - Hidden Address Space Access is available in Emulator Mode. Instruction fetches and Native loads and stores access Hidden Address Space; this leaves all of "normal" storage for external programs written with either Native or CISC instructions.
- Interrupt and Exception Processing - CISC emulation requires additional interrupts and exceptions that the RISC/CISC must handle. For example, descriptor based storage accesses require segment protection violation exceptions.

In summary, the RISC/CISC dual architecture processor thus exhibits:

- the ability to operate efficiently as a RISC and a CISC
- emulation of a CISC via a Native architecture plus extensions.