APZ 212 30—Ericsson's new high-capacity AXE central processor

Per Holmberg and Nils Isaksson

The APZ 212 30 is a completely new central processor for the AXE system with three to four times the execution capacity of its predecessors. The high capacity is achieved by combining unique APZ performance features with state-of-the-art processor architecture and innovative design.

Other features in the new processor are improved memory capacity and a new ring network for external communication interfaces that greatly extends the flexibility of regional processor bus configurations and enables the APZ processors to make use of new, high-speed communication interfaces. The implementation of highly integrated CMOS circuits gives low power dissipation and improves reliability.

Being fully compatible with existing applications and installed hardware, the new processor can be used in new installations and for upgrading the system execution and memory capacity of previously installed systems. With this new processor, the AXE 10 system satisfies the demands for greater capacity created by the increasing number of subscribers in mobile networks and by new, revenue-generating service offerings.

The authors describe the architecture and implementation of the new APZ 212 30 processor, paying special regard to its advanced execution and communication mechanisms.

APZ 212 30 architecture

The APZ 212 30 is a completely new design (Figure 1). It retains and further enhances the unique high-performance architecture of the APZ 212 series of processors, implements a state-of-the-art execution pipeline, and introduces several new performance features. Its entire architecture has been optimized for the characteristics of telecommunications—efficient context switching, memory access and communication enable the processor to execute thousands of tasks in parallel. Instead of relying on a single processor unit to do all the work, the task of execution has been divided between two dedicated processors:

- the instruction processor unit (IPU), which executes application code; and
- the signal processor unit (SPU), which terminates protocols and schedules jobs (in conventional computers, these functions are usually associated with the operating system).

Another feature retained from previous APZ 212 processors is the pure Harvard architecture, in which the IPU has separate instruction and data caches and separate memory for instructions (program store, PS) and data (data store, DS). This design permits parallel access to instructions and data even at cache misses.

Program execution in the IPU is very advanced: instructions are decoded and executed in parallel (superscalar execution); the instructions are also dynamically reordered ("out-of-order" execution) for optimum performance. Instructions from application programs are decoded into internal RISC-style (reduced instruction-set computer) instructions. To handle jumps in the code, the processor employs dynamic branch prediction, executing on the predicted path (speculative execution). Innovative features include the pre-decoding of instructions as they are loaded into the program store, and a high-performance data store architecture.

The APZ 212 30 communicates with the AXE system via the regional processor bus handler (RPH), which implements a new ring network that allocates communication bandwidth to serial and parallel regional processor (RP) bus interfaces and high-capacity networks.

Job signal flow

The regional processor bus handler connects directly to up to 32 regional processor bus branches. External job signals (messages) that arrive over the RP bus are forwarded to the signal processor unit, which analyzes the signal, assigns a priority to it, and queues it in the job buffer where it awaits execution in the instruction processor unit. The SPU loads one job signal at a time into the IPU. When a job signal arrives, the IPU identifies it, looks up the start address of related program code in the program and reference store (PRS) table, and then begins executing the program. Programs that execute in

BOX A, ABBREVIATIONS

ALU	Arithmetic logic unit
ASIC	Application-specific integrated circuit
BGA	Ball grid array
CMOS	Complementary metal-oxide semiconductors
CP	Central processor
CPS	Central processor operating system
DMA	Direct memory access
DRAM	Dynamic random-access memory
DS	Data store
ECC	Error-correcting code
I/O	Input-output
IPU	Instruction processor unit
ISP	In-service performance
MAS	Maintenance system
MAU	Maintenance unit
MTBSF	Mean time between system failures
POWC	Power controller
PRS	Program and reference store
PS	Program store
RISC	Reduced instruction set computer
RP	Regional processor
RPH	Regional processor handler
RS	Reference store
SDRAM	Synchronous DRAM
SPU	Signal processor unit
SRAM	Static RAM
SSRAM	Synchronous, static RAM
UMB	Update and match bus

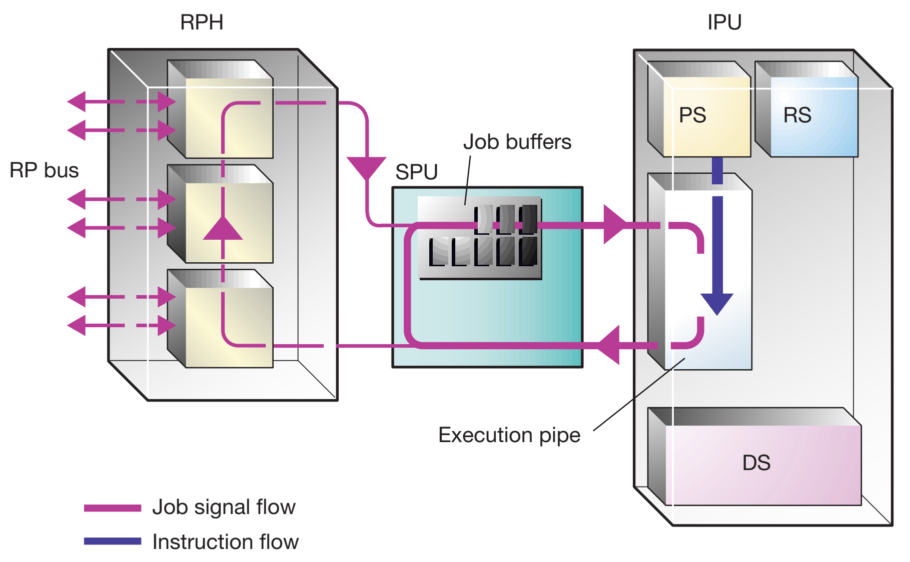


Figure 1
The APZ 212 30 architecture.

the IPU can send new job signals to the SPU. Signals that are designated to other program blocks are queued in job buffers; signals designated to other processors in the system are routed for transmission over the RP buses.

Job signal pipeline

The IPU-SPU interface has been optimized to support high throughput of job signals, which are transported through a pipeline between the SPU and IPU. While the IPU executes one job, the SPU preloads the next job signal directly into an extra bank of processor registers in the IPU. Thus, when the IPU finishes the first job, it swaps register banks and immediately begins executing the preloaded job without first having to copy registers (Figure 2).

Job signals to the SPU are also transported through a pipeline. The processor registers included in the signals are copied to a send buffer using the full internal band-

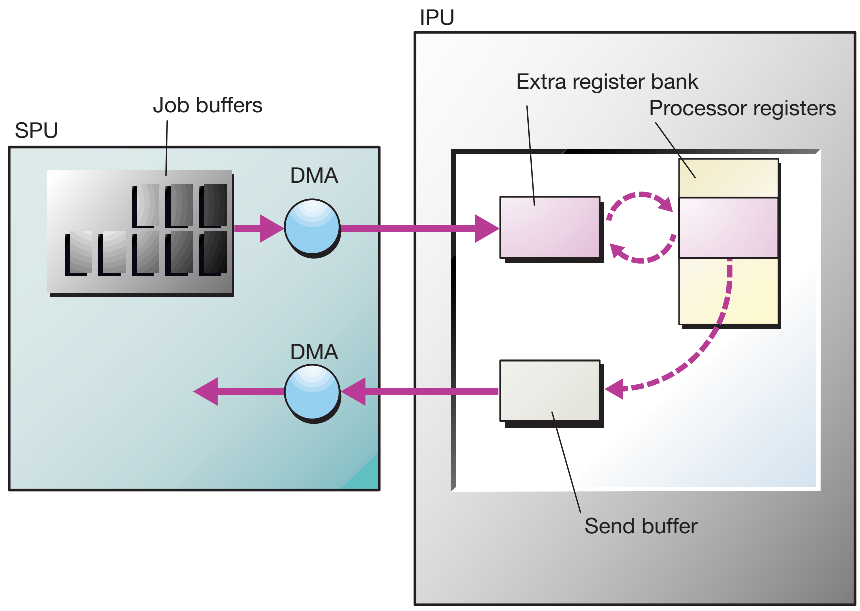


Figure 2
The IPU-SPU interface: the jobs signal pipeline.

width on the IPU processor chip. The SPU fetches signals from the send buffer while the IPU continues executing the same job or switches to the next. The SPU uses autonomous direct-memory-access (DMA) engines to transport job signals.

The combination of off-loading job scheduling and protocol termination from the main CPU to the SPU and of job signal pipelines enables the APZ 212 30

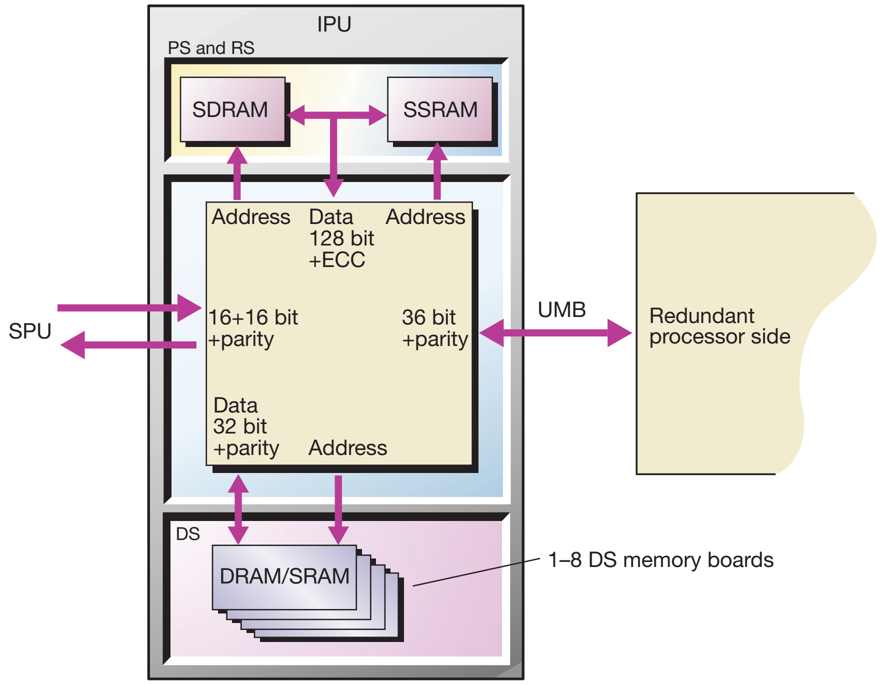
- to switch contexts and start executing a new job in as few as 30 clock cycles; and
- to send a signal in just 15 clock cycles.

Compare this to the hundreds or thousands of clock cycles that a standard microprocessor needs to do the same task. The APZ 212 30 can thus efficiently switch context 300,000 times per second and still devote most of its time to executing application code.

IPU structure

Compared to ordinary microprocessors, the APZ 212 30's instruction processor circuit is not limited to a single processor bus for communicating with main memory and other processors and networks. Instead, separate high-capacity buses connect to the program and reference store memory, the data store memory boards, and the SPU. Each bus operates at full processor frequency (Figure 3).

Figure 3
The IPU block diagram.



The IPU memory interfaces have been optimized to suit the characteristics of telecommunications applications. Access to the program and reference store is often sequential and concentrated to a narrow range of addresses. To support this kind of access, the memory system implements a wide bus and uses "page mode" in modern, synchronous, dynamic, random-access memory (SDRAM). This combination gives almost instantaneous access (three clock cycles) to data within an 8 Kword (16 Kbyte) range of addresses. Similarly, frequently used tables and program blocks are copied to synchronous, static RAM (SSRAM) for access in just two clock cycles.

By contrast, access to the data store is usually non-sequential and distributed between several different memory addresses. The IPU supports this kind of access by dividing the data store into banks and allowing up to eight parallel access attempts—provided no two attempts address the same bank simultaneously. The memory area on each data store memory board is divided into 16 banks, which means that one board can give full memory bandwidth in the system. The data store, which is highly configurable, can hold any combination of from one to eight memory boards. Two types of memory board are currently available: a 512 MWord (1 GByte) DRAM board, and a 32 MWord (64 MByte) high-speed SRAM board.

Execution of instructions

Instructions are executed in the instruction processor unit, which has a short, six-stage instruction-execution pipeline (Figure 4) in which each stage corresponds to one clock cycle. Instructions are shown flowing from top to bottom, with a new pair of instructions starting each clock cycle. In telecommunication applications, which are characterized by many changes in control flow (short jobs and frequent jumps and calls to other software blocks), the pipeline must be short.

Internally, the processor uses the equivalent of internal RISC microinstructions. Application program instructions are decoded into these microinstructions before they are executed. Complex instructions are decoded into a stream of microinstructions.

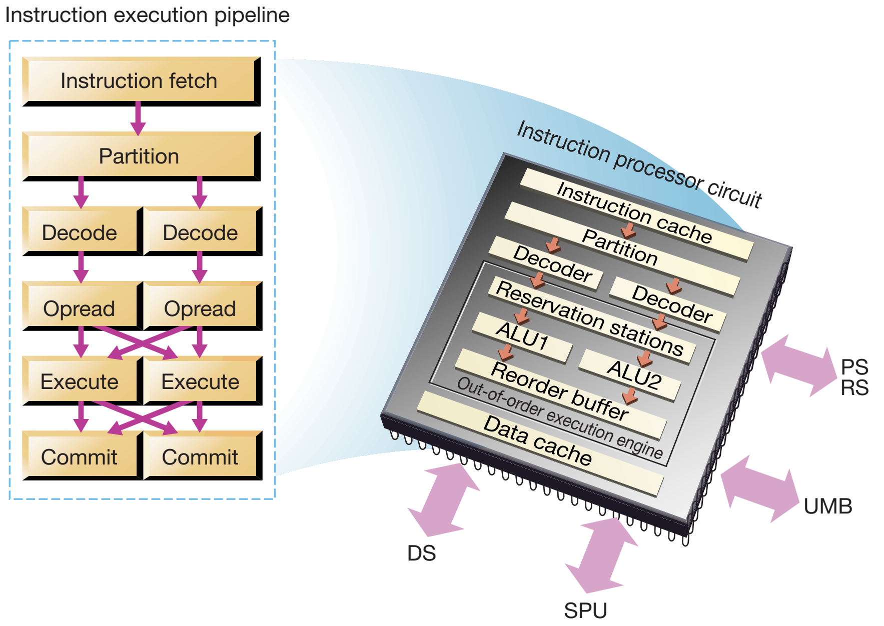


Figure 4
Instruction pipeline.

The first stage, called *Instruction fetch*, fetches instructions in 128 bit memory words (eight 16 bit words) from the on-chip program cache, the external second-level cache, or from the program and reference store. Given an average instruction length of 1.5 words, there are approximately five instructions in each memory word. Thus, five new instructions are loaded every clock cycle.

In the second stage, called *Partition*, up to two instructions are extracted from the memory word. These are decoded in the third stage, *Decode*. Instructions that perform simple operations—such as an ADD instruction, which adds the context of two processor registers—are directly decoded into a single microinstruction. Instructions that perform complex operations—for instance, an end program (EP) instruction, which ends the execution of the current job and switches context—are decoded into a stream of microinstructions.

The forth stage is called opread. Depending on the type of *Operand*, the microinstructions are written to one of five queues, called reservation stations. Here, the instructions wait for their operands to be fetched from the register file or from memory, or they wait for the results of earlier instructions. Up to eight instructions can be active in this stage simultaneously.

When an instruction has received all its operands, it is passed to the fifth stage, *Execute*. In this stage, up to two instructions can be executed in parallel, in separate arithmetic logic units (ALU).

The final stage, *Commit*, writes the results of the instructions to a register or memory.

The ultramodern design of the execution pipeline features the following characteristics:

- superscalar execution—two instructions can be decoded, executed and committed in the same clock cycle.
- branch prediction—when the processor performs a conditional jump, it does not wait until the branch condition is known; instead, it predicts the most probable branch and continues execution on that branch. Branch prediction is based on a very large 64 K entry prediction table used to achieve high prediction accuracy in the telecom application;
- speculative execution—execution on a conditional branch is speculative until the branch condition is known. The results from executed instructions are stored in temporary registers. If the processor failed to predict the correct branch, then the registers are cleared and execution is restarted. Otherwise, the results are committed in the last stage of the pipeline (Commit).
- Out-of-order (non-sequential) execution—if one instruction is delayed (for example, from waiting for data to arrive from memory), then subsequent instruc-

BOX B, MAIN FEATURES OF THE APZ 212 30 PROCESSOR

The main features of the APZ 212-30 processor are:

- Large processing capacity—three to four times more capacity than the APZ 212 20 (depending on application characteristics). To expand capacity, designers combined a new, advanced processor architecture, a higher clock frequency and high-speed static RAM data-storage boards;
- Large memory capacity—4 GWord (8 GByte) of data storage (up from 1.5 G Word), 96 MWord (192 MByte) of program storage (up from 64 MWord), and 32 MWord (64 MByte) of reference storage (up from 4 MWord);
- New functions—a generic communication bus interface based on a ring network that allows for adaptation to high-speed networks. A new communication-buffer mechanism allows program blocks to share
  - data in a buffer; and
  - access, using new, rapid-communication buffer read-and-write instructions.
- Optional configurations with high-speed SRAM boards and standard dynamic RAM boards for data storage. By using additional SRAM boards, the system can be configured from optimal price/capacity to best capacity.
- In-service performance (ISP)—integration into custom complementary metal-oxide semiconductors (CMOS) improves system reliability. In most configurations, mean time between system failures (MTBSF) is now more than 10,000 years. Well-defined interfaces and fewer boards improve diagnostics when hardware faults occur.
- Improved hardware maintenance—board number, revision.
- Reduced size—the processor fits into a single 600 mm, double-sided cabinet (half the size of its predecessor).

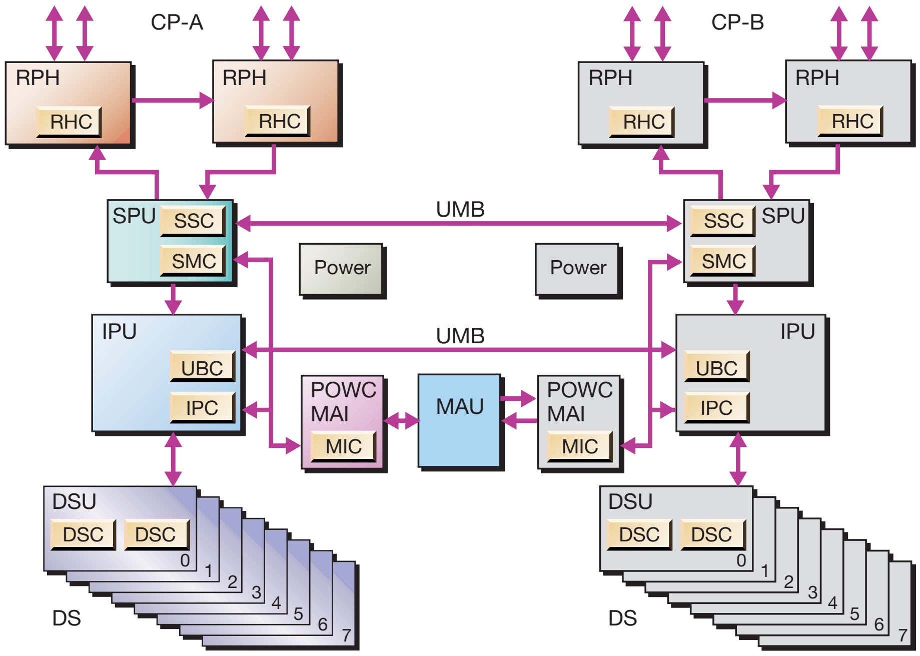


Figure 5
Duplicated hardware structure of the APZ 212 30 central processor.

DS	Data store
DSC	Data store circuit
DSU	Data store unit
IPC	Instruction procesor circuit
IPU	Instruction processor unit
MAI	Maintenance unit interface
MAU	Maintenance unit
MIC	Maintenance interface circuit
POU	Power unit
POWC	Power control unit
PRS	Program and reference store
RHC	RPH circuit
RPH	Regional processor handler
SMC	SPU master circuit
SPU	Signal processor unit
SSC	SPU slave circuit
UBC	Update bus circuit

tions are allowed to bypass the waiting instruction. The hardware first confirms inter-instruction dependency to ensure that an instruction does not bypass any instructions on which it depends for data.

- register renaming—the processor has access to more physical registers than the programmer can see. Dependencies are avoided by assigning a new, temporary register to the results of each instruction. This further enhances the capacity of out-of-order-execution;
- multilevel instruction cache system—support for fetching instructions is provided by a small, single-clock-cycle access, on-chip, level-one cache and a larger SRAM-based level-two cache; and
- data cache—support for fetching data is provided by a small, on-chip data cache together with optional, low-latency SRAM boards.

In addition to these features, the APZ instruction processor implements the following unique features to further improve capacity:

- Harvard architecture—the design of separate instruction and data memory allows simultaneous access to instructions and data.
- Load-time pre-decode—instructions are mapped to a new optimized format when loaded into the program memory. This action, which is performed by the loader in the operating system, is not visible to the user. Thanks to the new format, the IPU is able to extract two instructions from a memory word in just one clock cycle.
- Early jump extraction—jump instructions and their target addresses are identified in the partition stage before the instructions have been decoded. This enables the IPU to fetch instructions from the new path earlier and minimizes the penalty for taking the jump. This feature is especially important in processors for telecom applications, since the associated code has a high frequency of jump instructions.
- Early load-extraction—the one factor that most affects capacity in modern processors that run applications, such as a telecommunications control system (which uses a large amount of data storage) is access time for reading data from

DRAM. The new, optimized instruction format enables the IPU to identify and extract the variable address early on in the pipeline (during the partition stage), which decreases the access time for reading data.

- Loop unroller—instead of running loops in the microprogram, the loop unroller generates sequential instructions on the fly. This completely eliminates jumps in loops and improves capacity when data is copied to and from registers; when data is copied from memory to memory; and during linear search operations in memory.

Signal processor unit

The signal processor unit (SPU) is equipped with two specialized processors: the SPU master processor and the SPU slave processor, each of which is a micro-programmed RISC processor whose instruction set has been optimized for its specific duties.

The SPU master processor schedules jobs and preloads them for execution in the IPU. It also schedules periodic jobs in the system by scanning the job table and creating and scheduling the job signals to start them.

The SPU slave processor administers the RPH ring network and terminates the RP bus protocol (for example, retransmission). Outgoing messages are routed to the correct RPH. Incoming messages are forwarded to the SPU master processor.

Direct memory access engines serve as automatic data transports of signals between SPU buffer memory and the IPU as well as to the RPH.

Regional processor handler

The regional processor handler connects the central processor to the regional processors by providing interfaces to up to 32 RP bus branches. Internally, the regional processor handler implements a new ring network for communication between the SPU and interface boards. The ring network yields greater bandwidth and facilitates flexible configuration of interface boards and flexible allocation of communication bandwidth. There are currently two interface board types: one for connecting to two parallel RP buses; and one for connecting to four of the new, serial RP buses. The ring network in the RPH also supports the addition of new, high-speed data-communication interfaces (Box C).

In-service performance

With its fault-tolerant configuration, using two central-processor sides (CP-A and CP-B) that execute in parallel, the APZ 212 30 furthers the tradition of APZ robustness (Figure 5). A maintenance unit (MAU) supervises operation, selecting one side to execute and the other side to operate on standby. The standby side performs the same operations as the executing side, trailing it by 12 clock cycles. Having two sides guarantees tolerance against hardware faults and enables operators to conduct maintenance activities without loss of service. For example, one side can be extended with new hardware or software

BOX C, DESIGN AND POTENTIAL OF THE RPH/RING NETWORK

Logical design
Several point-to-point connections from the SPU to the RPH with two kinds of communication channel:

- a signaling channel
- a broadcast channel (from the SPU to every RPH)

Physical design

- Ring topology with a time-slot protocol that guarantees bandwidth per RPH.
- Synchronous clock operation supports fault tolerance in the central processor.
- 160 Mbit/s bandwidth
- Highly configurable
  - 1 to 16 interface boards of available types
  - free configuration of parallel and serial RP bus interfaces
  - (prepared for) dynamically allocable bandwidth per board
  - support for new high-speed interfaces
  - automatic configuration identifies new boards

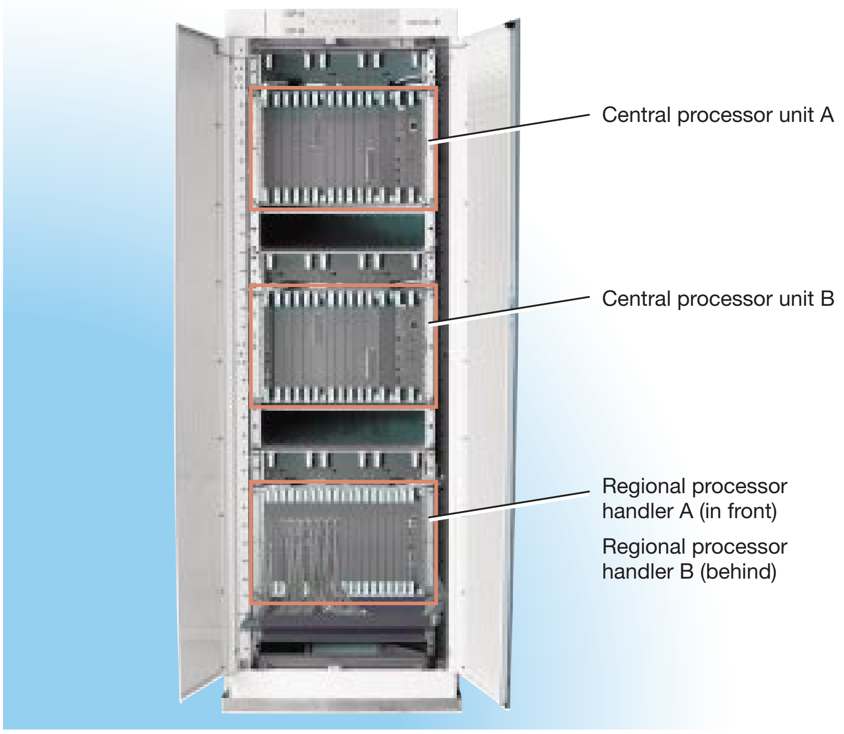


Figure 6
APZ 212 30 cabinet.

while the other side continues executing system operations.

In addition to the fault-tolerant configuration, each CP side has been designed to provide high availability. Memory (SRAM and SSRAM) is protected by error-correcting code (ECC), which corrects single bit errors. The EEC also corrects faults in whole circuits. The data store, for example, contains a large number of memory circuits, providing up to 8 GByte of memory. Consequently, if faults are detected in these circuits, the ECC corrects them.

The extensive use of application-specific integrated circuits (ASIC) makes for a very clean design (circuit boards solely contain custom circuits and memory) and reduces power dissipation. These features contribute toward exceptional mean time between system failures (MTBSF) for hardware faults—10,000 years in most configurations.

Technology

The APZ 212 30 is composed of eight separate circuit designs: one for the data store unit, two for the IPU boards, two for the SPU board, one for the RPH, one for the power controller (POWC) board, and one for the test unit (MIT trace equipment). All circuits have been implemented in high integration 0.35 micron CMOS. This circuit technology facilitated the advanced architecture of the APZ 212 30 that was needed to attain high capacity. To a large extent, the processor capacity of telecommunications applications is limited by the access time to large external data stores. At 80 MHz system frequency, the APZ 212 30 with its advanced, superscalar IPU, Harvard architecture, SRAM boards, and use of load-time pre-decode procedures, can easily keep up with memory access attempts.

The processor circuit of the IPU is housed in a 735-pin ball grid array (BGA) package. It is the largest circuit, with 2.8 million transistors in logic and 7.4 million transistors in memory.

The APZ 212 30 processor is housed in a single cabinet (600 mm wide) that holds four subracks, two for each CP side (Figure 6). The CPU subrack of each side holds the processor and memory boards; the RPH subrack holds the interface cards to the RP buses.

Software support and adaptation

Accompanying the new hardware are new releases of the operating system (CPS) and the maintenance system (MAS). These include support for the new processor hardware and functionality:

- Communication buffers—operating system support for allocating and deallocating communication buffers and for managing buffer pools.
- Data store SRAM boards—the operating system measures and allocates frequently used data to SRAM.
- Measurement functions—support for built-in performance counters and for measuring software behavior; capacity-enhancing mechanisms in the processor.
- RPH ring network—configuration of interface boards.

Software design support

While hardware designers developed the new APZ processor, other developers worked on an upgrade for the software design support:

- PLEX compiler—the new release offers better capacity and supports the new communication buffers.
- EMU CP emulator—a new version, based on new emulation technology, speeds up emulation and more exactly emulates the APZ processor.
- MIT trace equipment—this single-board trace device, which fits into a spare slot in the CPU subrack, can record every operation in the CP, including the execution of all application instructions, the execution of microinstructions in the IPU and the two SPU processors, and signals between units. This device is invaluable when debugging the new system. In real systems, collected traces give input for detailed analyses of application behavior.

Future directions

The APZ 212 30 is fully upgradable. Its advanced architecture will deliver substantially greater capacity when adapted to future silicon processes. Moreover, new features can be included for further decreasing system downtime and simplifying system handling.

Subsequent designs will make greater use of standard computer interfaces and components and support standard Internet protocols.

Conclusion

The APZ 212 30 is a completely new processor design that is now in operation in several markets around the world. It uses an advanced architecture for achieving high capacity in telecommunications applications. The processor yields three to four times greater execution capacity and extends data store capacity to a full 4 Gword (8 Gbyte).

A new, generic communication bus interface gives greater flexibility and opens the system to new types of communication buses.

The advanced processor architecture—implemented through standard CMOS technology—fulfills objectives for performance, integration, and power dissipation, and offers exceptionally high mean time between hardware-related system failures.