The Protocol Engine® Chipset*

Greg Chesson, Silicon Graphics
Des Young, Protocol Engines

*Protocol Engine is a registered trademark of Protocol Engines, Inc.

Technical Focus:

VLSI for Networking

"any protocol, any media, any bus ... "

Project History

Datakit work by Chesson at Bell Labs in 1980-81

R+D study within Silicon Graphics during '88-'89

Protocol Engines venture-funded in late '89

Joint SGI/PEI development program 1990 to present

First silicon: 4Q91

Commercial availability expected March '92

Engineering Team

PEI

Hwa-Jin Bae
Hon Wah Chin
Inwah Choi
Alan Hewer
Vasan Karighattam
Wesley Lee
Rajiv Limaye
Eric Lin
Mike Murray
Jeannine Smith
Vinod Sutravez
Daniel Yau
Des Young

SGI

Bill Calvert
Greg Chesson
Way Chi Doo
Jan Gronski
Roger Hu
Kianoosh Naghshineh
Carlin Otto
Jim Roark
Rob Warnock
Tina Yang

Performance Notes

1. Network traffic is 60% to 80% small (60-200 bytes) packets. The chip set is designed to handle continuous streams of short back-to-back packets at full network speed (up to 200 Mbits/sec) for an arbitrary number of sources or sinks.

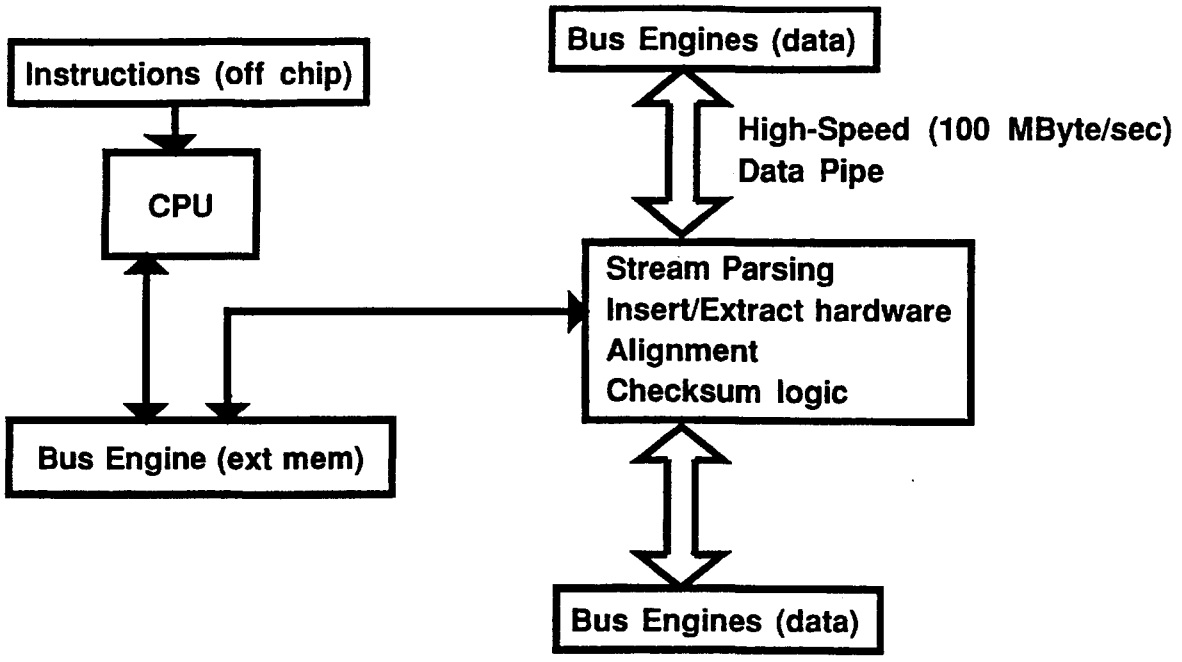
2. Simulations predict that the overhead for a host network software can be reduced by 50% with fastpath only firmware in the chips. Second release firmware will improve by removing overheads associated with ACK traffic and RPC/XDR operations.

3. Destination address processing in the MPort receive channel has dedicated hardware for the lookup and decision logic. On chip fifo depth is sufficient to make a decision to not forward an incoming packet before the packet has crossed the chip. This improves data bus statistics in bridge/router applications.

Chipset Summary

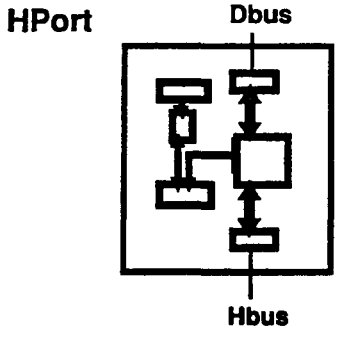
delivers 100 MIPS or more
programmable for different applications
initial firmware for IP/TCP/UDP/NFS/XTP protocols
5 - 10 microseconds packet processing time
up to 200K packets/second per port
up to 200 Mbit/sec media port
supports several different MAC interfaces
handles all checksum, addressing, segmentation, fragmentation
buffer management, DMA, alignment, multiplexing,
and data transfer functions.
supports bridging/routing applications

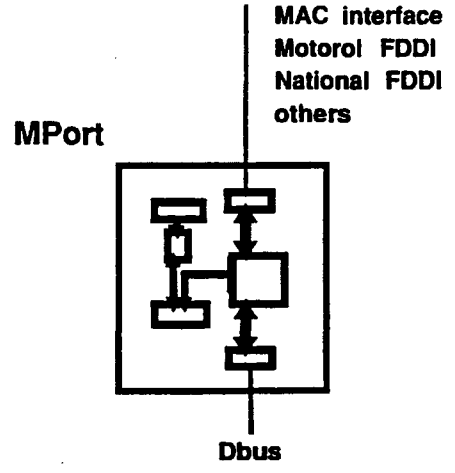
network management, SMT, connection management supplied
by ancillary cpu or host.



Chip Architecture

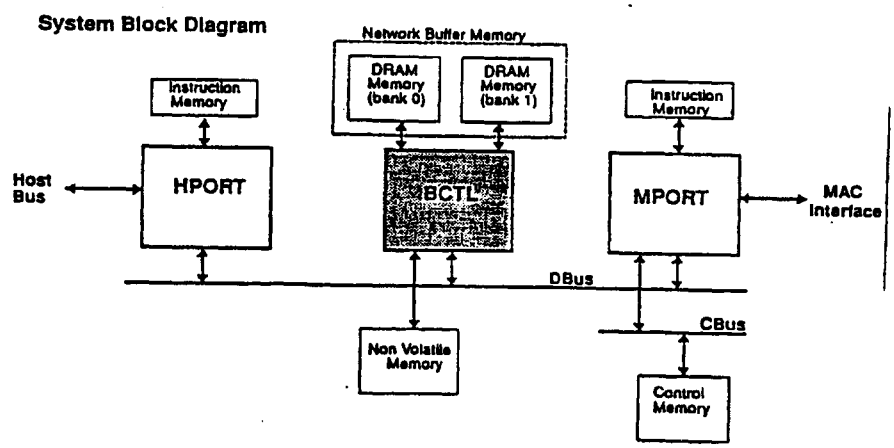
Four bus interfaces: two for data path across chip, one for cpu instruction fetch, one for off chip data. Extract logic is driven by the stream parser at Data Pipe speed for automatic extraction of header information for cpu to look at. This happens in parallel with data flowing through the pipe.





External View of port chips:

24-bit (+parity) instruction memory
32-bit Cbus interface (optional)
32-bit Dbus interface (all parts)
either (a) interface to a MAC chip, or
(b) 32-bit HBus (host bus) interface



Host Interface: HPort+MPort+BCTL+MAC

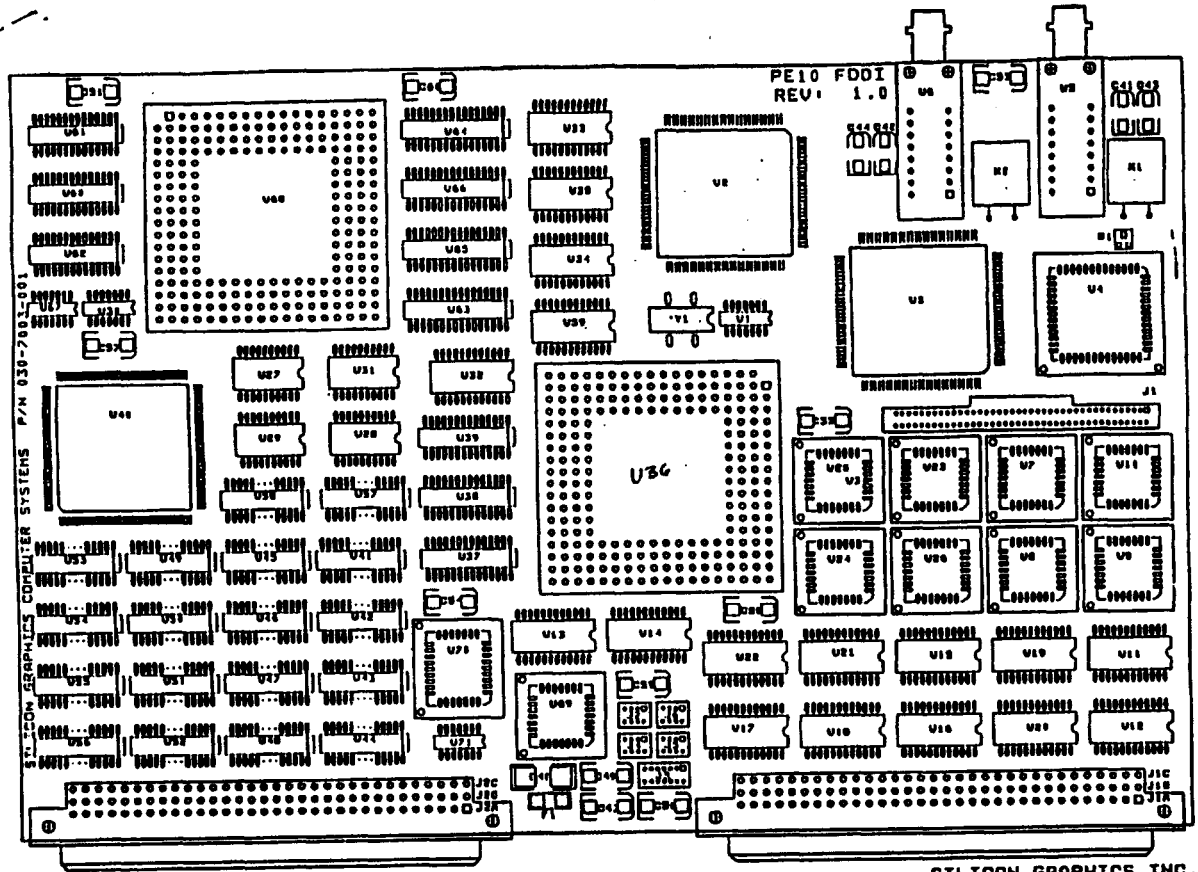
PE-10 Overview

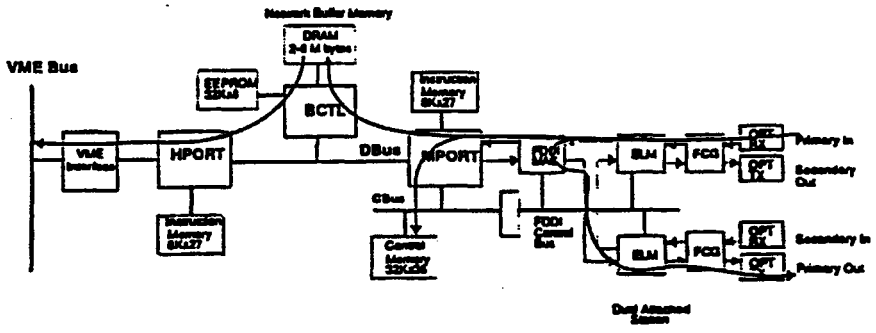
3 components:

MPORT - a single chip controlling a MAC and input fastpath processing

HPORT - a single chip controlling a host buss and output fastpath

BCTL - an ASIC providing dram memory control, timing and arbitration.



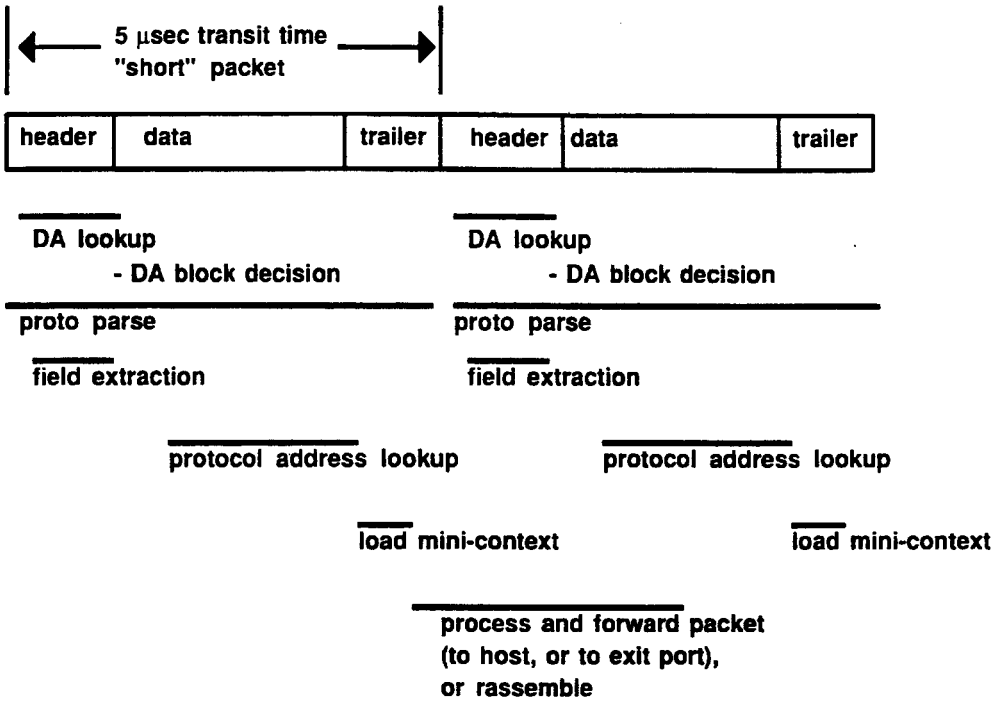


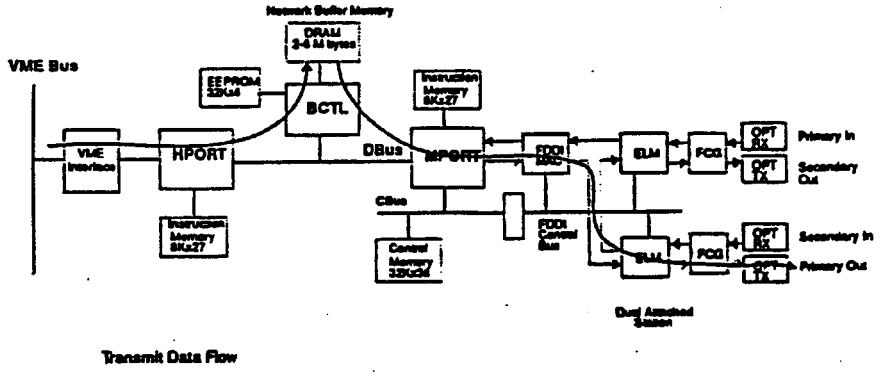
Receive Data Flow

Receive Data Flow

Hardware state machines buffer incoming packets onto temporary linked lists in the BCTL's buffer memory. At the same time the proto parser recognizes the frame and extracts information for the on-chip Kosy. Firmware then either puts the received packet back on the freelist, or appends it to the chosen queue entering the host, or keeps it on a reassembly queue until any missing pieces arrive or the reassembly process times out.

Concurrent Processing *for both low latency and high bandwidth*





Transmit Data Flow

Transmit Data Flow

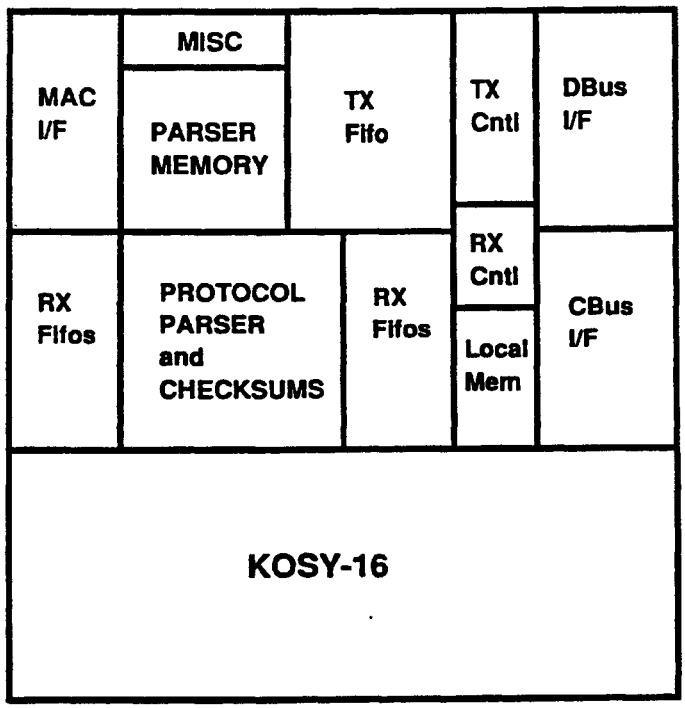
HPort reads commands and data from lists in host memory. As data moves across HPort it is checksummed and attached to protocol headers that are generated as needed. Completed frames are placed on MPort output queues for transmission.

Chipset Configurations

simple host interface:	1 Hport, 1 Mport, 1 BCTL
simple bridge:	2 Mports, 1 BCTL
multi-media host:	2 Hports, 1 Mport, 1 BCTL
bigger bridge/router:	4 Mports, 1 Hport, 1 BCTL

Up to five Port chips can be on the Dbus operated by a buffer control (BCTL) chip.

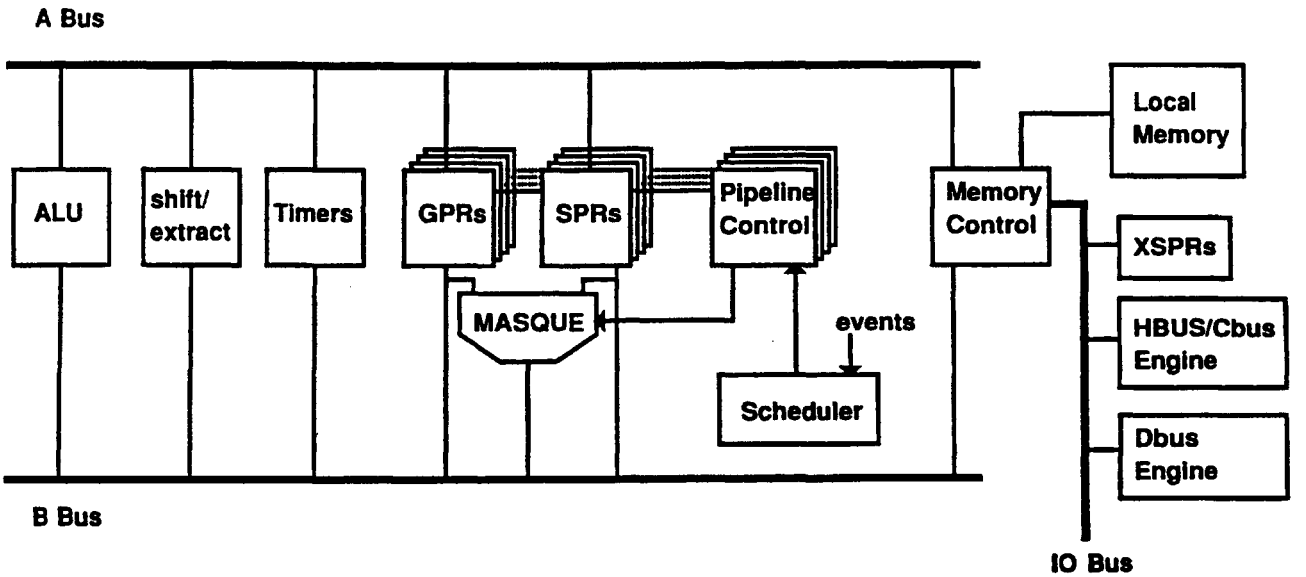
MPort Floorplan



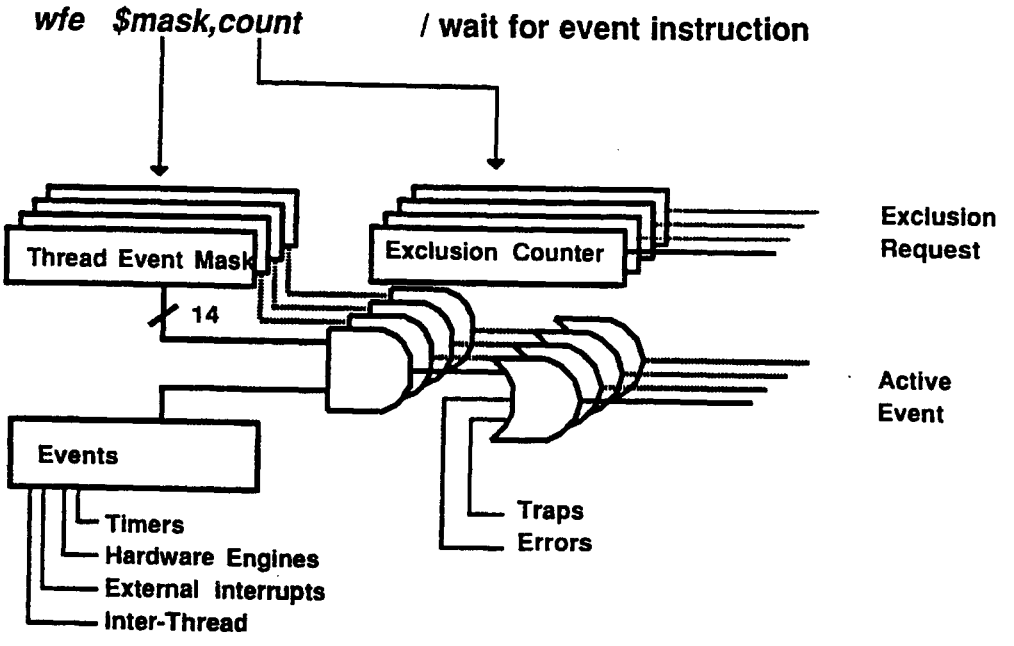
Real Time Core (Kosciusko) features

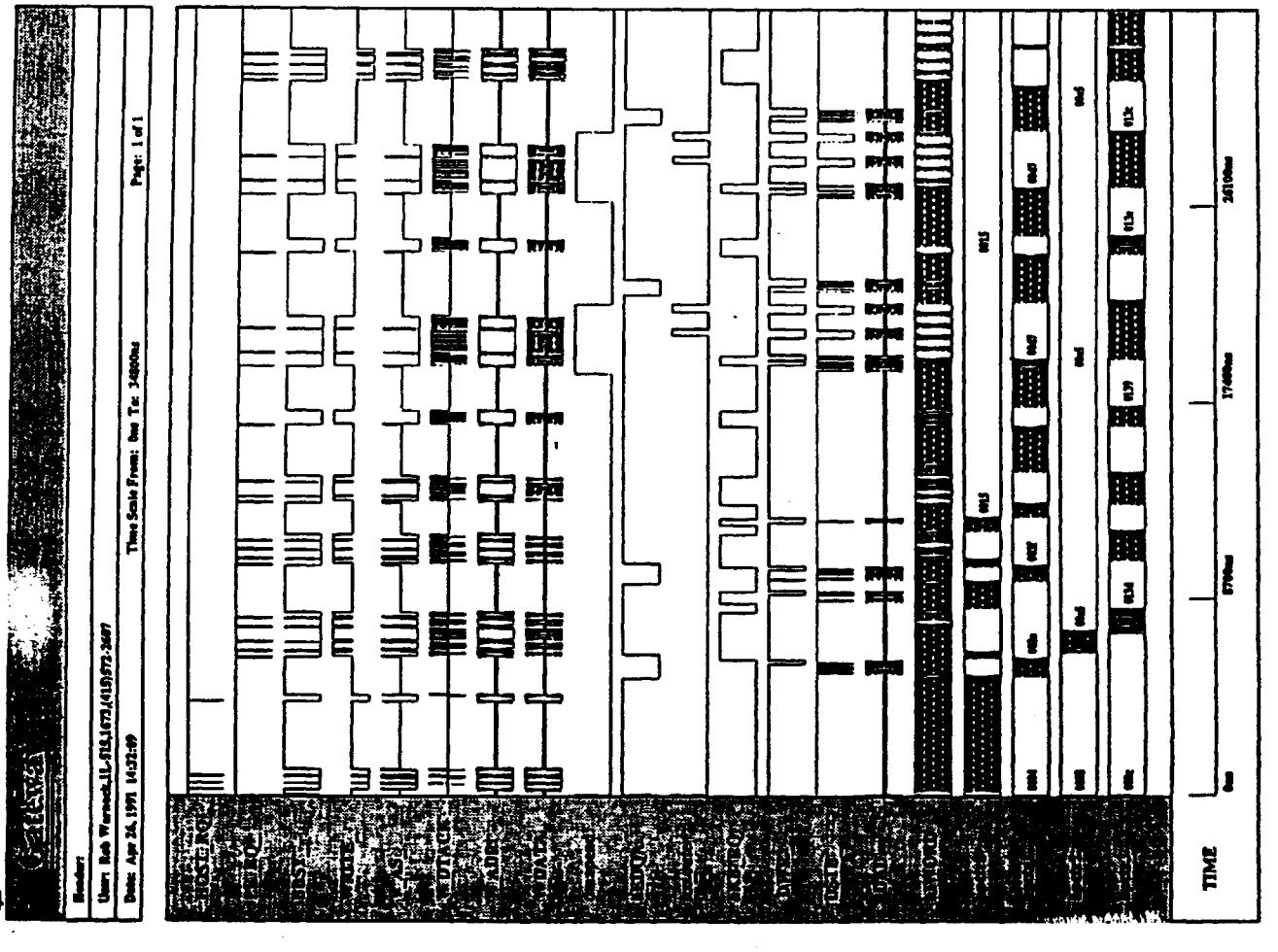
- four process threads each with its own GPR register set and state.
- 32 registers per set.
- an additional 32 register SPR set for communicating with onchip machines.
- single delay slot after loads and branches, but not stores.
- both 16-bit and 32-bit core standard cell designs (Kosy-16 and Kosy-32).
- an active thread can execute in parallel with bus engine activity or load/store multiple activity of another thread.
- threads are scheduled from hardware events.
- scheduling latency is one or two clock ticks with no pipe stalls. This means interrupt/event response is about 80 ns with no overhead for register save/restore.
- the *Masque* logic lets one thread masquerade as another. This allows a debugger to execute in the chip and access/manipulate the state of another thread.
- internal pipeline state is brought out of the core to external pins. This allows a programmed logic analyser to be a poor man's ICE.

Kosciusko Block Diagram



Thread Scheduling





Protocol Engine Futures

- support for multiple 8-12 ethernets on an MPort
- support for gigabit interfaces
- eventually a single-chip implementation
- standalone version of Kosy, if sufficient demand
- interfaces to compression and encryption devices
- applications of Kosy and pipeline processing to i/o applications other than networking.