DECwest Operating System Group

17 February 1988

Digital Equipment Corporation
Confidential and Proprietary
Internal Use Only

Presentations

- **Overview - John Gilbert**
- **Glacier and Cheyenne Product Management - Terry Morris**
- **Glacier and Cheyenne Development - Benn Schreiber**
- **PRISM ULTRIX Program - Robert Bismuth**
- **Flint Development - David Ballenger**
- **PRISM ULTRIX Product Management - Rockie Morgan**

DECwest
Quarterly Review

PRISM Operating Systems Group

Digital Equipment Corporation
Confidential and Proprietary

John Gilbert
17th February 1988

PRISM Operating Systems Group

Mission

Provide all operating system support for PRISM-based systems

- Mica:
  - Modern operating system
  - Strong VMS compatibility
  - Flexibility, extensibility
- PRISM ULTRIX:
  - Industry-standard operating system
  - "World-class" through layered products, strong service offering, multiprocessing/multithreading

PRISM Operating Systems Group

Project Structure

- **Product Project Leaders:**
  Responsible for functionality, quality, integrity, and product schedule

- **Project Leaders:**
  Responsible for definition, design, specification, development, testing, and integration of functional elements

- **Architects:**
  Responsible for completeness, consistency, and integrity of design

- **Other Key Project Functions:**
  - Testing strategy
  - Performance measurement and analysis
  - Applications engineering

PRISM Operating Systems Group

Accomplishments This Year

Staffing:

- 36 team members now. Four more to join in March.

Mica Design:

- Overviews complete. Half of WDD chapters submitted to architects

Mica Project:

- Base level plan complete
- Project planning documents complete by the end of ~~July~~ June

PRISM ULTRIX Project:

- Aggressive staffing in progress
- Backbone schedule to hardware groups
- Detailed schedule in March

Problems:

- Mica staffing plan cut
- Debugger
- Testing strategist

Presentations

- Overview - John Gilbert
- Glacier and Cheyenne Product Management - Terry Morris
- Glacier and Cheyenne Development - Benn Schreiber
- PRISM ULTRIX Program - Robert Bismuth
- PRISM ULTRIX Product Management - Rockie Morgan
- Flint Development - David Ballenger

Glacier and Cheyenne Summary

Product Management

Digital Equipment Corporation
Confidential and Proprietary

Terry Morris
DECwest Product Management

Glacier

Phase Review Schedule

- **Phase 0 exit - December 1987**
- **Phase 1 exit - ???**
- **Phase 2 exit - September 1989**
- **Phase 3 exit - March 1990**

Glacier

FRS Proposed Software Tools Suite

- FORTRAN
- C
- PASCAL
- Ada
- Multithreaded debugger
- Language source code conversion aid
- Language Sensitive Editor (LSE)
(for COMPILE command hooks)
- Application tuning aid (like VAX PCA)
- AIA services (DECwindows, ARUS, CMA, PSM)
- IPSE and/or CASE tools

Glacier

Competitive Analysis

- Size: $176M (1986)
- 70% Servers to networked workstations
- 1986 Market shares (Dataquest):
  - Convex 23%
  - FPS 19%
  - Alliant 15%
  - Elxsi 7%
  - SCS 4%
  - Others 22%

Glacier

Features Required to Gain Market Share

- Parallel processing
- Vectors
- Large real memory Sizes
- Large virtual memory sizes
- Third-party applications
- VAX/VMS compatibility
- High memory and I/O bandwidth
- Large storage connectivity
- Cost-effective performance
- Reliability
- Common software architecture

Cheyenne

Phase Review Schedule

- Phase 0 exit - December 1987
- Phase 1 exit - July 1988
- Phase 2 exit - June 1990
- Phase 3 exit - December 1990

Cheyenne

DDA-Compliant Client Software

- Required at Cheyenne FRS
  - Rdb Star
  - SQL
  - Data dictionary
  - Intact/ACMS
  - DDTA front end and back end
- Software that will enhance competitive position
  - TEAMDATA
  - Rally
  - Hierarchical storage management

Cheyenne

Competitive Analysis

- OLTP
  - Market Leader: IBM
  - Technology Leader: Tandem
  - Others: Unisys, NCR
- EUC (End User Computing)
  - Investigating this is a high-priority task

Cheyenne

Features Req'd to Gain Market Share

- Data integrity
- Availability
- Throughput
- Response time
- Architecture compliance
  - DDTA
  - DDA
- Database size
- Price/performance

Glacier and Cheyenne

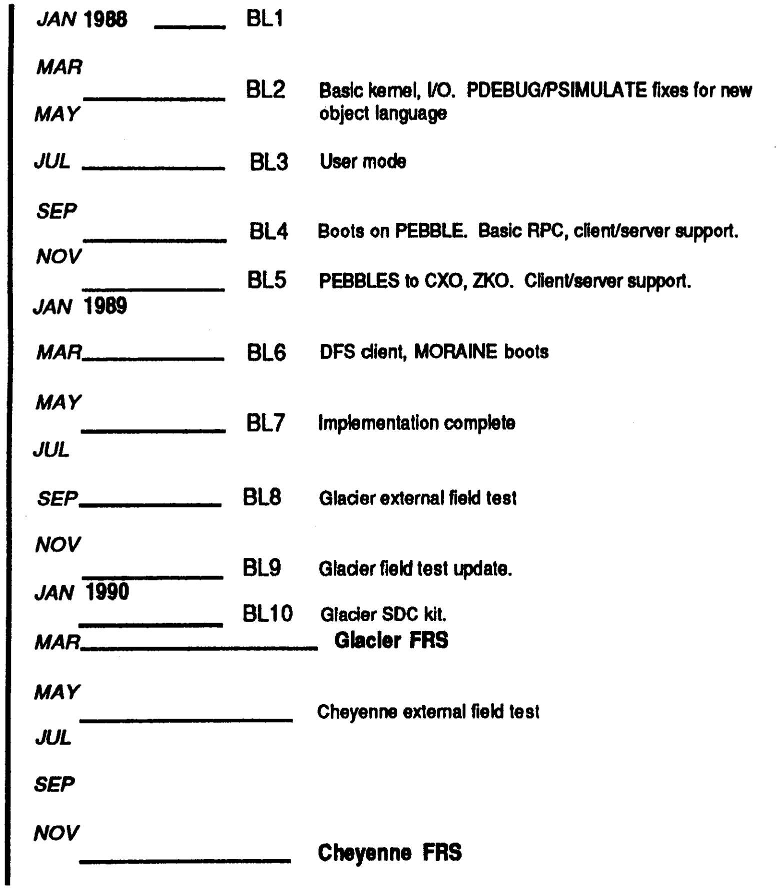
Development Status
17 February 1988

Digital Equipment Corporation
Confidential and Proprietary
Internal Use Only

WDD

- 60 overviews completed on schedule (1-Jan-1988)
- WDD chapters
  - 10 complete
  - 20 in various stages of review process
  - 28 chapters in progress
  - 1 chapter deferred, 1 chapter merged
- Presentations scheduled Mondays and Fridays into June
- Overview distribution March 1988

Figure 1: Glacier/Cheyenne Schedule



JAN 1988 BL1

MAR

BL2 Basic kernel, I/O. PDEBUG/PSIMULATE fixes for new object language

MAY

JUL BL3 User mode

SEP

BL4 Boots on PEBBLE. Basic RPC, client/server support.

NOV

BL5 PEBBLES to CXO, ZKO. Client/server support.

JAN 1989

MAR BL6 DFS client, MORAINE boots

MAY

BL7 Implementation complete

JUL

SEP BL8 Glacier external field test

NOV

BL9 Glacier field test update.

JAN 1990

BL10 Glacier SDC kit.

MAR **Glacier FRS**

MAY

Cheyenne external field test

JUL

SEP

NOV

Cheyenne FRS

External Dependencies

- **VAX RPC**
- **VAX DFS**
- **SDT**
  - **FORTRAN compiler and RTL**
  - **Language-independent RTL (AIA)**
  - **BLISS**
  - **DEBUG**
  - **PCA**
  - **PASCAL compiler and RTL**
  - **ADA**
- **SPM**
- **Product Marketing Groups for CMP applications**

Development Strategy

- Current development work on emulator and simulator
- Low-level kernel and exec first
- Booting on emulator
- Initial development in SIL, conversion to PILLAR this summer
- PEBBLE (PRISM-based Calypso box) "emulator"
- Second Ethernet through ZSO for DECnet Phase 5 development
- User-mode work on VAX/VMS
  - MICA/RMS to VMS/RMS interface routines

PRISM ULTRIX Program

17 February 1988

Digital Equipment Corporation
Confidential and Proprietary
Internal Use Only

Program Goals

- Provide early PRISM-based workstations for DIGITAL to compete in the market.
- Provide a high quality PRISM based leadership UNIX operating system providing:
  - Support for future PRISM systems
  - Compatibility with market demands (POSIX, SVID, NFS, etc.)
  - Compatibility with existing VAX ULTRIX
  - Compatibility with DIGITAL's Application Integration Architecture
  - Common layered product and tool environment with Mica

Products

- Software
  - Flint - PRISM ULTRIX on Shike and Osprey
  - World Class UNIX - evolutionary target for PRISM ULTRIX
- Workstations
  - Shrike - desktop
  - Osprey - deskside
- Servers
  - Moraine
  - Stone

Overall Schedule Dates

Flint on emulators - August 1988 - October 88

- Shrike - FRS June 1989
- Osprey - FRS September 1989
- Moraine - FRS March 1990
- Stone - FRS FY91
  Stone is the first system with world-class UNIX

Dependencies

- SDT
  - Language compilers (apart from C)
  - Application Integration Architecture
- ULTRIX Engineering Group
  - Initial source pool (Version 2.4)
  - Initial ports of:

    Networking software - DECnet, TCP/IP, NFS,
    LAT support
    Utilities
    Generic file system interface
    X and DECwindows

Vision of a World-Class UNIX

- Reached by evolution from Flint
- Shares common components with Mica:
  - Mica kernel
  - I/O architecture and drivers
  - Memory management
  - Mica interfaces: AIA, RMS, RPC, etc.
  - PRISM Calling Standard
- Conforms with industry standards

Goals of a World-Class UNIX

- To be a major force in the industry by which other UNIX implementations are measured.

- To place DIGITAL in a leadership position in the UNIX market.

PRISM ULTRIX

Flint

Product Management Summary

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

Rockie Morgan
DECwest Engineering Product Management

Preliminary Phase Schedule Overview

- **Phase 0 exit - December 1987**
  Phase 0 documents available March 1988
- **Phase 1 exit - August 1988**
- **Phase 2 exit - February 1989**
- **Phase 3 exit - June 1989**

Proposed Layered Products

Layered products required at FRS

- FORTRAN
- Ada
- GKS
- PHIGS
- EPIC Office suite
- PASCAL

Layered products required after FRS

- Common LISP
- IPSE architecture supporting UNIX tools
- AIA

Competitive Analysis Workstations 1987

- Market size: $2,200 million (Dataquest)
- Market share:
  - 28%+ Sun
  - 20% Apollo
  - 10% DIGITAL
  - 4% IBM
  - 38% other (Silicon Graphics, Masscomp)

Competitive Analysis Workstations - Sun 4/260

Processor Type:	MB86900 (Sun SPARC RISC Chip)
Clock Speed:	16.67 MHz
Floating Point:	Weitek 1164/1165
MIPS:	10 (5-7 VUPS)
Memory:	8 to 128 Mbytes
Cache:	128-Kbyte write-back
Bus:	VME (DMA)
Price:	$39,900 (8-Mbyte, monochrome, diskless)
	$68,400 (8-Mbyte, color, 280-Mbyte disk and tape)
Software:	Sun OS (4.3 BSD UNIX)
	NFS, NeWS and SunView window manager

Competitive Analysis Workstations - Sun 4/110

Processor Type:	MB86900 (Sun SPARC RISC Chip)
Clock Speed:	16.67 Mhz
Floating Point:	Weitek 1164/1165
MIPS:	7 (3-6 VUPS)
Memory:	8-Mbytes (later to 32-Mbytes)
Cache:	none
Bus:	VME (DMA)
Price:	$18,900 (8-Mbyte, monochrome, 140-Mbyte disk)
Software:	Sun OS (4.3 BSD UNIX)
	NFS, NeWS and SunView window manager

Competitive Analysis Workstations - Sun 5/260

Processor Type:	**Sun SPARC RISC Chip version 2**
Floating Point:	**Weitek 1164/1165**
MIPS:	**20 (10-14 VUPS)**
Memory:	**32-Mbytes**
Bus:	**VME (DMA)**
Price:	**$42,800 (32-Mbyte, color, 400-Mbyte disk)**
Software:	**Sun OS (4.3 BSD UNIX) NFS, NeWS and SunView window manager**

Features Required to Gain Market Share

- Industry Standards
  - 4.3BSD
  - SVID (volumes 1 and 2)
  - POSIX
  - TCP/IP
  - NFS
  - UUCP
  - DECnet
- Security Enhancements
- Vectors
- Symmetric multiprocessing and multithread support
- Vectorizing/decomposing FORTRAN and C
- DIGITAL's full compiler set
- IPSE architecture
- Common Software Architecture - CSA

Flint Development

17 February 1988

Digital Equipment Corporation
Confidential and Proprietary
Internal Use Only

Flint Schedule

- **BL1 - June 1988**

  Initial design and coding, code "ported" to VAX C

- **BL2 - August 1988**

  Convert to PRISM C and run on emulators

- **BL3 - September 1988** - now November! Shrike prototypes delayed

  Boot on Shrike, begin integration of UEG-supplied components, ULTRIX RMS implementation

- **BL4 - October 1988**

  Full Shrike graphics and device support, continue integration and test of UEG-supplied code

- **BL5 - November 1988**

  Running on Pebble, begin Osprey support, compilers, linkers, and librarians ported to PRISM ULTRIX

- **BL6 - January 1989**

  Complete integration and test of UEG- and SDT-supplied components, full system builds on Shrike

Flint Schedule

(continued)

- BL7 - February 1989
  Field test of PRISM ULTRIX on Shrike, complete support for Osprey
- BL8 - March 1989
  Field test of PRISM ULTRIX on Osprey
- BL9 - June 1989
  FRS of PRISM ULTRIX on Shrike
- BL8 - September 1989
  FRS of PRISM ULTRIX on Osprey

Cooperative Agreement with UEG

- UEG deliverables
  - Most ULTRIX utilities and RTLs
  - ULTRIX file systems and network support
  - X and DECwindows support
- DECwest responsibilities
  - Base ULTRIX kernel
  - Drivers and device support
  - Compilers, linker, librarian
  - System-dependent tools and RTL components
  - Common software architecture components
  - System management and installation
  - Software integration and build
  - Overall product integration
- Source and version synchronization
- Project tracking and schedule

Progress to Date

- **Initial cooperative agreement circulated and reviewed**
  Final agreement to be reached by the end of the month

- **Initial project scheduling**
  More detailed baselevel schedule available by March

- **Development begins in March**

Development Strategy

- **Initial design and coding, code "ported" to VAX C**
- **Convert to PRISM C and boot on emulators**
  Cross-development system VAX to PRISM
- **Basic user environment on emulators**
- **Provide support for Shrike**
- **Begin integration of UEG-supplied components**
- **Provide support for Pebble**
- **Compilers, linker, and librarian ported to Flint**
- **Full development environment and build on Flint**
- **Complete integration of UEG- and SDT-supplied components**
- **Provide support for Osprey**

PRISM Layered Products Overview

Software Development Technologies
Spit Brook Road

17 February 1988

Speakers

Chip Nylander

Overview

Rich Grove, GEM Project

FORTRAN
GEM Common Back End
Vectorization
Decomposition

Chip Nylander

BLISS
PASCAL
Ada

Al Simons, Run Time Library

Language RTLs
Math RTL
RTL Vectorization
AIA and Utility RTL
Common Multithread Architecture
SORT/MERGE

Chip Nylander

DEBUG
Performance and Coverage Analyzer
VAXSET and Tools
Integrated Programming Support Environment
Graphics
Compound Document Architecture
Information Architecture

End User Information Management

Overview

Targets for SDT Layered Products

- PRISM Ultrix V1.0
  SHRIKE and OSPREY desktop/deskside
- GLACIER
  MORAINE and STONE MICA compute server
- PRISM Ultrix V2.0
  Desktop/deskside and MORAINE and STONE
- ROCK
  64-bit PRISM

Seamless VAX <-> PRISM Layered Products

SDT VAX-PRISM Compatibility Working Group
Managing this within SDT

PRISM Common Software Architecture

- Develop most PRISM layered products just once
- Deploy on all targets (as appropriate)
- Some limitations—DEBUG, etc.

PRISM Common Software Architecture

Joint Effort between DECwest and SDT

Compatible superset of AIA

Planning delivery in two main waves

- PRISM Ultrix V1.0 (SHRIKE/OSPREY)
- MICA V1.0 and ULTRIX V2.0 (MORAINE)

PRISM Common Software Architecture

Components currently planned

- Extended Calling Standard
- Condition Handling
- Naming Conventions and Name Space Management
- Object Language
- Librarian Interfaces
- Status codes, message files, message reporting
- Integrated Programming Support Environment
- Record Management and File Specification Parsing
- Product Invocation (command parsing, etc.)
- Remote Procedure Calls
- Common Multithread Architecture
- Date/Time Format and Manipulation
- AIA Utility RTL (system interfaces)
- Language RTLs
- Math RTL
- DECwindows
- Modular Memory Management
- String Manipulation
- Compound Document Architecture

To reiterate—this is a joint DECwest and SDT effort

Summary of Layered Product Staging

** Preliminary **

SHRIKE and OSPREY FRS

FORTRAN (scalar uniprocessor) compiler and RTL
BLISS cross-compiler (internal use only)
Scalar Math Run Time Library
First wave of AIA

Still Being Evaluated For SHRIKE/OSPREY

PASCAL compiler and run time library
Graphics—GKS, PHIGS, EPIC, etc.

MORAINE FRS

FORTRAN (vectorizing decomposing) compiler and RTL
BLISS native compiler (internal use only)
Vector Math Run Time Library
Next wave of AIA
SORT/MERGE
DEBUG and PCA run time kernels for MICA
VAXSET and NOTES for PRISM ULTRIX
IPSE Junior
Access to Common Data Dictionary from server???

Staging and Strategy Still Being Evaluated

Compound Document Architecture

- Digital Data Interchange Syntax
- Digital Document Interchange Format
- Table Data Interchange Format
- Toolkits, converters, viewers, etc.
- Desktop Publishing

Information Architecture

- Common Data Dictionary
- Etc.

End User Information Management

NOT SDT

- C Compiler
- LISP and other AI Products

Summary of Layered Product Staging

** Preliminary **

STONE FRS

Full IPSE
Earliest possible Ada
Earliest possible DEBUG and PCA for ULTRIX
Ongoing global optimization improvements
Ongoing decomposition and vectorization

ROCK FRS

Retarget products to 64-bit environment

Dependencies

- Stable PRISM Program definition.
- PRISM hardware architecture specification.
- Availability of adequate prototype hardware early enough.
- ULTRIX Operating System base levels for development and testing.
- ULTRIX development tools, debugger, and performance profiler.
- MICA Operating System specifications and design.
- MICA base levels for development and testing.
- PRISM Common Software Architecture.
- PRISM C compiler
- PILLAR compiler

- Good market analysis on which to base requirements and set priorities.

Risks and Issues

- There are PLENTY of unknowns and TBDs—this is an aggressive program of development.

- Bi-coastal development program.

- SDT is NOT satisfied with current SDT commitments; layered product set looks thin, especially for PRISM ULTRIX V1.0. VAXSET, languages, etc. This is best we can commit to, however.

PRISM Layered Products Details

Software Development Technologies
Spit Brook Road

17 February 1988

BLISS

Requirements

- BLISS for DEC internal development purposes.

  Compiles Common BLISS ("Common" slightly redefined)
  May be made available to selected customers if needed.

- Portability flagging in compiler

- Documentation on transportable programming

  VAX <-> PRISM
  32-bit <-> 64-bit

- SHRIKE/OSPREY Compiler

  VAX -> PRISM cross-compiler
  Adequate, correct code.

- MORAINE Compiler

  Native PRISM compiler
  Highly optimized, correct code

BLISS

Strategy

- Use VAX BLISS as cross-compiler base; modify back end for PRISM instruction set and object code.
- Interface BLISS front end to GEM compiler shell and back end for native PRISM BLISS compiler.

Progress and Current Status

- Portability flagging in compiler—complete
- Cross-compiler for PRISM emulator—complete
- Guide to Transportability—complete, ongoing
- Cross-compiler support for new object language—in progress
- Native compiler—planning in progress

Future Plans

- Q1FY89—New cross-compiler for internal use
- Q1FY90—Native Compiler

PASCAL

Requirements

- PASCAL compiler for scalar, non-decomposed applications
- VAX PASCAL compatibility
- Handle unformatted D-floating data in external files
- Handle both VAX and PRISM record alignment
- Available with PRISM ULTRIX V1.0 and MICA V1.0

Commitments

- PASCAL available with PRISM ULTRIX V2.0 and MICA V1.0

Strategy

- Retarget VAX PASCAL front end to GEM compiler shell and back end
- Preserve existing code -> RTL interface design
- VAX PASCAL and PRISM PASCAL built from as much shared source code as possible.

PASCAL

Progress and Current Status

- Front end retargeting in progress
- Compiler has been compiled with PRISM BLISS
- Some retargeting to GEM complete

Future Plans

- Finish compiler shell and intermediate language retargeting
- Plans will be further detailed during Phase 1

Risks, Issues, Problems

- Current schedule commitments do not match currently stated schedule requirements. PASCAL is resource-constrained.
- We hear the current requirements, and are investigating what we can do.
- Will respond to schedule requirements by end of Phase 0 (targeting 3/88)
- Vectorizing, decomposition requirements for PASCAL?

  Answer will apply to VAX PASCAL as well.

Ada

Requirements

- Ada compiler for scalar, non-decomposing applications
- VAX Ada compatibility
- Handle unformatted D-floating data in external files
- Handle both VAX and PRISM record alignment

Commitments

- Ada team will begin looking seriously at PRISM Ada in Q3FY89.
- Compiler not expected before Q2FY91 (earliest)

Strategy

- Retarget VAX Ada front end to GEM compiler shell and back end
- VAX Ada and PRISM Ada built from as much shared source code as possible.
- Will investigate basing VAX Ada V3.0 on VAX version of GEM.

Ada

Progress and Current Status

- Ada team has been tracking and getting their requirements into PRISM architecture, calling standard, MICA operating system design.
- Ada team initiated and provided primary leadership in design of Common Multithread Architecture.
- Currently dedicated to VAX Ada V2.0

Future Plans

- Begin looking at PRISM Ada and GEM in Q3FY89.

Dependencies

- Ada is more dependent than other languages on special support in MICA operating system (for tasking).

Ada

Risks, Issues, Problems

- Ada is both resource-constrained (Ada front end) and technology-constrained (code generation for Ada on PRISM)
- Vectorizing, decomposition requirements for Ada?

  Answer will apply to VAX Ada as well.

PRISM DEBUG

Requirements

- Debug run-time kernel hosted on MICA V1.0
- User interface (DECwindows) and debugging hosted on VAX client—"Remote Debugging".
- Support for vectorized, multithreaded and decomposed applications.
- Eventually, debugging on PRISM ULTRIX.

Commitments

- Same as requirements.

PRISM DEBUG

Strategy

- Redesign the Debug Symbol Table (DST) for general retargetting.
- For ULTRIX V1.0, dbx (not done in SDT) is expected to support this Debug Symbol Table.
- Split debugger into target specific and target independent pieces for general retargetting.
- Engineer for remote debugging on MICA V1.0.
- Strategy summary

  DECwindows user interface
  General retargetting
  Remote debugging for MICA
  Whole of DEBUG to PRISM ULTRIX (someday)

PRISM DEBUG

Progress and Current Status

- DECwindows user interface in progress.
- Redesign of eventpoint handing underway for general retargetting.
- Redesign of Debug Symbol Table in progress.
- Tracking MICA development.

Dependencies

- MICA Operating System design.
- MICA Calling Standard.

Short Term Risks

- Resources

Performance and Coverage Analyzer

Requirements

- Run time Performance Collector hosted on MICA V1.0
- User interface (DECwindows) and performance analysis hosted on VAX client—"Remote Performance Analysis".
- Eventually, performance collection and analysis on PRISM ULTRIX.

Commitments

- Same as requirements.

Performance and Coverage Analyzer

Strategy

- PCA uses many of DEBUG interfaces for collection and analysis; tends to follow DEBUG development.
- Additional, new PCA-specific interfaces and techniques also planned for PRISM.
- PRISM Compiler, DEBUG, ULTRIX groups must design new interfaces for support of PRISM performance analysis before ULTRIX V1.0
- For ULTRIX V1.0, prof and gprof will be retargeted to those interfaces
- For MICA V1.0, PCA will be engineered for remote performance analysis and collector hosted on MICA.
- Eventually, plan to have full PCA for PRISM ULTRIX.
- Strategy summary—One Step At A Time

  DECwindows user interface
  Remote performance analysis
  Collector to PRISM MICA
  Collector to PRISM ULTRIX
  User interface and analysis to PRISM ULTRIX

Performance and Coverage Analyzer

Progress and Current Status

- DECwindows user interface in progress.
- Tracking DEBUG and MICA development.

Future Plans

- When DECwindows user interface done, begin addressing MICA work.
- Must participate in design of new performance collection interfaces for PRISM.
- ULTRIX PCA not before Q2FY91.

Dependencies

- PCA dependent on DECwest languages, DECwest ULTRIX, SDT languages, and SDT DEBUG getting together to define compiler -> performance collection and analysis interfaces.

VAXSET

Requirements

- VAXSET tools requested for PRISM ULTRIX V1.0

  Text Processing Utility
  Language Sensitive Editor
  Source Code Analyzer
  Code Management System
  DEC Test Manager
  etc.

- Seamless VAX-hosted tool support for compilers and applications hosted on MICA V1.0.

Commitments

- Planning seamless support for MICA at MICA V1.0.
- Targeting PRISM ULTRIX V2.0, but firm plans not in place yet; this is a ** preliminary ** target.

VAXSET

Strategy

- For MICA, incremental enhancements to existing VAX-based tools.
- For ULTRIX, To Be Determined.

  Depends on larger corporate UNIX strategy, and what is done for VAX ULTRIX.

Progress and Current Status

- Developing DECwindows user interfaces and new required tool functionality.

  Resulting products should be basis for PRISM products.

Risks, Issues, Problems

- Mismatch of requirements and current commitments.

  VAXSET work is resource-constrained.
- Uncertainty around corporate UNIX strategy (VAX)

Integrated Programming Support Environment

Requirements

- IPSE callable interfaces on MICA V1.0 for seamless support of MICA compilers and applications.
- Full IPSE for PRISM ULTRIX as possible.

Commitments

- Same as requirements.

Strategy

- To Be Determined—VAX IPSE in very early V1.0 development.

Future Plans

- Currently planning VAX IPSE in two waves:

  IPSE Junior
  Full IPSE

  PRISM ULTRIX IPSE likely to be staged in same manner.

Graphics and Forms

Requirements

- GKS, PHIGS, EPIC, etc. requested for PRISM ULTRIX V1.0
- Investigating what requirements might be for callable application interfaces on MICA compute server, driving remote VAX client user interfaces.

Commitments

- None yet. This area still being evaluated.
- GKS and PHIGS are currently targeted for PRISM ULTRIX work in next fiscal year.
- Forms products are not currently planned for any PRISM system.

Risks, Issues, Problems

- What ARE the requirements (if any) for callable application interfaces on MICA?

Compound Document Architecture

CDA includes

Digital Data Interchange Syntax (DDIS)
Digital Document Interchange Format (DDIF)
Table Data Interchange Format (TDIF)
Toolkits, converters, viewers, etc.
Desktop Publishing

Requirements

- Not well understood.
- PRISM applications need to be able to exchange data with VAX applications.
- Part of CDA likely to be in AIA.

Compound Document Architecture

Commitments

- None yet. This area still being evaluated.

Progress and Current Status

- Core Application Group has not been presented with a product strategy that they can respond to.

Risks, Issues, Problems

- DDIF requires new support in operating system supplied Record Management System.
- What are the requirements for Desktop Publishing on PRISM ULTRIX?

Information Architecture

Requirements

- Not well understood.
- Seamless VMS <-> MICA environment probably requires compiler and application access to Common Data Dictionary.

Commitments

- None yet. This area still being evaluated.

Strategy

- To Be Determined.

Progress and Current Status

- Dictionary group has started investigating making Dictionary accessable from MICA server.

Risks, Issues, Problems

- What are requirements for IA products on PRISM?

End User Information Management

Requirements

- Not well understood.

Commitments

- None yet.

Strategy

- To Be Determined.

Risks, Issues, Problems

- What are requirements, if any, for End User Information Management products for PRISM systems?

FORTRAN Review

S. Whitlock
R. Grove

17 February 1988

Review Agenda

PRISM Fortran

- Product Description
- Development Strategy
- Dependencies
- Base Level Progress

Product Description

PRISM Fortran

- Accepts any standard VAX FORTRAN program
  - will generate multi-thread reentrant code
  - VAX and PRISM record alignment
  - D_FLOAT and H_FLOAT not on PRISM
  - supports D_FLOAT in files
  - remove PDP-11 FORTRAN compatibility features
- will remain compatible with future VAX FORTRAN versions
- today's target is PRISM-32 under MICA and ULTRIX
- planning for PRISM-64 in the future
  - only known language difference is INTEGER*8

Staffing

Denise Lagassé	- supervisor
Stan Whitlock	- project leader
Mike Anderson	
Sid Maxwell	- handles VAX FORTRAN/ULTRIX

Development Strategy

- start with VAX FORTRAN v5 front end
- track any VAX FORTRAN v5 and v6 additions
- use GEM back end:

  code generation
  intermediate language optimizations
  vectorization
  any decomposition for parallelism
- use GEM shell:

  command line and file name processing
  file I/O, INCLUDE file, LIBRARY file handling
  LSE diagnostic file handling
  generate .LIS, .OBJ, DEBUG DSTs
- generate calls to PRISM RTL
- host-target development timeline

  VAX hosted -> emulated PRISM-32 target
  VAX hosted -> PRISM-32 hardware
  PRISM-32 native compiler under ULTRIX
  PRISM-32 native compiler under MICA

Dependencies

PRISM Fortran depends on

- PRISM emulators, simulators, and hardware
- ULTRIX and MICA
- PRISM Calling Standard
- PRISM RTL - MTH$, OTS$, LIB$, vector MTH$
- PRISM RTL - FOR$ (changing some interfaces)
- GEM back end and shell
- BLISS on PRISM
- PRISM DEBUG to debug compiler and resulting code
- PRISM Layered Product documentation strategy
- VAX FORTRAN and VAX Vector FORTRAN

no known project depends on PRISM FORTRAN

Base Level Progress

PRISM Fortran Base Levels

1. using GEM interfaces, copying VAX Fortran
2. compiler prototype using GEM shell for command line, source input, and listing output

    continuation lines
    comment processing
    right margin
    statement classification

3. process program units, classify all lexemes
4. create symbol table, handle minimal program, design expression handling

PROGRAM	
SUBROUTINE	minimal support
FUNCTION	minimal support
BLOCK DATA	minimal support
END	
labels	

5. type declarations, branching, assignment without operations

    unconditional GOTO
    INTEGER, REAL, etc
    assignment

Base Level Progress

PRISM Fortran Base Levels - continued

6. arithmetic expressions, array declarations
   - DIMENSION
   - all operators and precedence
   - arithmetic IF
   - data type conversion
7. HANOI.FOR without I/O compiled on 15-Dec-87
   - array references
   - array assignment
   - CONTINUE
8. block constructs, side effect processing, record handling - current base level
   - iterative DO
   - DO WHILE - ENDDO
   - IF - THEN - ELSEIF - ELSE - ENDIF
   - logical IF
   - COMMON
   - EQUIVALENCE
   - STRUCTURE - RECORD
9. routine calls, data initialization
   - CALL
   - function expressions
   - argument lists
   - formal parameters
   - DATA

Base Level Progress

PRISM Fortran Base Levels - continued

10. RTL interface, intrinsic functions

    OPEN - CLOSE
    READ - WRITE
    FORMAT
    SIN - COS
    implied DO lists

11. Miscellaneous

    integrate scalar globally optimizing compiler
    executable testing of cross-compiler on target
    October 1988

12. ULTRIX port

    target: SHRIKE FRS Q4FY89

13. MICA port

    target: MORAINE FRS Q3FY90 includes vectors

GEM Review

R. Grove
17 February 1988

Review Agenda

Project goals
Development Strategy and Methods
Current Status
Base Levels
Schedule
Dependencies
Issues

GEM Deliverables

GEM Version 1.0

- Supports PRISM FORTRAN V1.0
- Scalar globally optimizing compiler
- Ships with PRISM ULTRIX V1.0
- FRS June 89

GEM Version 2.0

- Supports FORTRAN, Pascal, Bliss
- State-of-the-art vectorization
- Ships with PRISM ULTRIX 2.0, MICA 1.0
- FRS March 90

Project Goals

GOALS (priority-ordered)

- Quality
- Schedule

  6/89 PRISM ULTRIX FRS
  3/90 MICA FRS
- Run-time performance
- Multi-language

  V1: FORTRAN
  V2: Pascal, Bliss
  Ada
  Perhaps COBOL and others
- Future 64-bit PRISM version
- Retargetable and rehostable: RISC and CISC

Optimization Goals

V1 Commitments (ULTRIX)

- State-of-the-art scalar global optimization
  Equivalent to VAX FORTRAN V5
- Procedure inlining
- Target-specific code scheduling

V2 Commitments (MICA)

- Loop unrolling
- Automatic vectorization
- Manual decomposition (VAX FORTRAN V5)

Under Study

- Automatic decomposition, Alliant-style fine-grain

Future

- Additional scalar optimization
- Interprocedural optimization

Development Strategy

GEM foundations, BL1-3

- Basic optimizer structure
- Code for assignments, conditionals
- OPAL front end only

Front end integration, BL4-5

- Integrate with FORTRAN front end
- Additional optimization, code generation

Globally optimizing scalar compiler, BL6-7

Rehost/Retarget PRISM ULTRIX, BL8

Advanced Optimization

- Vectorization
- Decomposition

Current Status

Base Level 5 completed this month

- Split lifetime optimization
- IL design for procedures and parameters
- Code for unaligned memory refs
- Code for byte/word arithmetic
- Peephole and branch optimizer
- PRISM native .OBJ file (partial)

Generates "reasonably optimized" code for many small benchmark programs

Lots of room for improvement in local code quality. That is, the global optimizer is more nearly complete than the local code generator.

Current Status, continued

FORTRAN successfully interfaced with GEM

Use of GEM shell components
- Driver and command parsing
- Source input
- Listing output package
- Source locators and error messages
- LSE diagnostic file output
- GEM integrity checker and dumper

Compiled HANOI.FOR 15 December

Now compiles many small tests and benchmarks

GEM/FORTRAN integration has proceeded smoothly and quickly

Base Levels

Base Levels 1-3

- IL and symbol table definition
- OPAL
- Flow graph construction
- Shell design and components

Base Level 4

- Global optimization
- Motions and profitability
- Machine code listing

Base Level 5, Feb 88

- Split lifetime optimization
- IL expansion for procedure calls
- Peephole optimizer
- Object file generation (partial)

Base Levels, continued

Base Level 6, May 88

- Optimization for procedures
- Code for procedure calls
- Code scheduling
- Finish object file construction

Base Level 7, Sep 88

- Code for strings (character and bit)
- DST generation

Base Level 8

- Rehost on PRISM ULTRIX

Base Levels 9...

- MICA development
- Vectorization
- Decomposition: manual, auto

Schedule

GEM schedule is currently being reworked to address all the changes in the PRISM program of past 6 months

- BL5 just completed
- Sharper dates and detailed workplans in March

Table 1: GEM Current Estimates

Date	Event
May 88	GEM Base Level 6
Sep 88	GEM Base Level 7, scalar optimizing cross-compiler
Sep 88	Start ULTRIX bootstrap
Oct 88	Executable FORTRAN tests, cross-compiler
Fall 88	Start GEM vectorization design
TBS	Start ULTRIX field test
Jun 89	PRISM ULTRIX FRS
Sep 89	MICA Field Test
Mar 90	MICA FRS, PRISM ULTRIX 2.0

Dependencies

GEM depends on	For
VAX FORTRAN	Vectorization technology
SDT Parallel AD	Auto Decomp technology
Front ends	GEM specification review
SDT DEBUG	PRISM DST specification
DECwest,PCA,ULTRIX	PRISM profiling interface
BLISS	BLISS32P cross-compiler
DECwest	Common Software Architecture: MICA and ULTRIX
DECwest	MICA specs, WDD
DECwest	Object language, Linker
DECwest	Hardware timing model
DECwest	PILLAR def module interface
DECwest	Universal optimization database

Front End Dependencies

Front end projects depend on GEM for:

- Specifications
- System interfaces
- Correct and optimized code generation
- Help in interfacing to GEM

GEM depends on front ends for:

- Review and feedback
- Conformance to interfaces
- Language-specific symbol tables and DST
- Language-sepcific optimization rules

Front end groups are responsible for:

- PRISM product definition
- Ultimate product quality
- RTL interface issues
- RTL semantic issues

GEM Vectorization Development Strategy

FORTRAN RAD Vectorization project (86-87) develops vectorization technology for VAX and PRISM

VAX FORTRAN V6 project (87-88) develops vectorizing FORTRAN compiler for FRS of VAX vector hardware

GEM project (88-89) transfers technology from VAX to PRISM.

Common vectorization algorithms and data structures, but 2 complete implementations.

FORTRAN RAD Project

Participants

Kevin Harris
Brian Koblenz
Steve Hobbs

Contributions

- Literature survey, analysis, assimilation
- Analysis of vector benchmarks
- PRISM and VAX vector architecture review
- Detailed design of vectorization in the context of the FORTRAN V4 global optimizer (similar to GEM)
- New dependency model
    - Tuples instead of statements
    - Refined dependency definitions
    - Added concepts of reversibility and distance

VAX FORTRAN Vectorization Base Levels

1. Compiler restructuring
2. Rough dependency analysis (Feb 88)
   Simple linear subscripts, loop control variable
3. Vectorize innermost loop (Spring 88)
   Generate vector IL and instructions
   Executable (and testable) code
4. Induction variable recognition
   Scalar expansion, assertions
5. Cost-based analysis
   Finding recurrences
   Fancier dependency analysis
6. Refined machine model
   Dependency analysis for aliases
7. Store motions
   Moving sectioning loops

Vectorization Schedules

Although VAX and PRISM vectorization are separate projects, PRISM builds on previous work for VAX

Table 2: PRISM and VAX vectorization

Date	Event
1986-1987	FORTRAN RAD project
Feb 88	VAX FORTRAN BL2
Spring 88	VAX FORTRAN BL3, first vector code
Fall 88	Start GEM vector design and implementation
Feb 89	VAX FORTRAN V6 Field Test
Sep 89	GEM V2.0 MICA Field Test

Vectorizing Compiler Structure

Front End

Local optimization

Global flow analysis
- Identify CSEs
- Constant propagation

GRAPH: Build dependency graph for each loop nest and do dependency analysis

SCR: Find strongly connected regions, cycles in dependency graph

GAUGE: Determine vectorization possibilities, costs

STRUCT: Choose best implementation, vector or scalar

SCHED: Reorder and schedule vector IL

Recreate flow graph and reapply (scalar) global flow analysis and optimizations:
- CSEs and code motion
- Value propagation

Code generation, FINAL, ...

GEM Decomposition Development Strategy

Similar sequence of steps as vectorization:

1. AD project develops technology
2. Development group assimilates AD results
3. GEM auto-decomp product development

SDT Parallel Processing AD project ongoing

Bill Noyce, project leader
Steve Hobbs, consultant

AD final reports, Spring 1988

GEM Decomposition
AD Project

AD project results from Bill Noyce:

- Know how to do "easy" things, where there is no dependency.
- The same dependency analysis techniques being used in FORTRAN V6 and GEM V2 are sufficient for automatic decomposition.
- Automatic decomposition will fit into the same overall compiler phase structure as is being used for vectorization.
- Considering two execution environments
  Multi-thread (MICA)
  Multiple processes (VMS, FORTRAN V5)

Currently working on heuristics for code replication

Detailed design notes, Spring 1988

GEM Decomposition Development Plans

The Parallel AD project is making good progress, and the results to date look very encouraging. There is a good fit between vectorization and decomposition.

Timeline

- Spring 1988, AD reports
- Fall 1988, GEM detailed design for vectorization and decomposition

Issues

PRISM ULTRIX schedule and requirements
Less than 12 months until Field Test
VERY FEW details known to us

Pascal required for PRISM ULTRIX V1.0, but planned later

Ada required for FRS, but not possible

Advanced optimizations not yet fully planned

- Vectorization: AD complete, commitments made, detailed design and implementation to be done.
- Decomposition: We need to assimilate results of Parallel AD, then estimate what can be committed for GEM V2.0 (MICA)

GEM is a very complex project, with many closely spaced and interdependent deliverables:
Scalar optimizing compiler
Vectorization
Parallel decomposition
More languages
Future VAX target

Run Time Libraries

- ARUS – The utility RTL
- FORTRAN Language support
- Pascal Language Support
- Math Library and vectorization
- SORT/MERGE
- Common Multithread Architecture

Key Players

Jim Totton	Development manager
Ken Hobday	Supervisor
Al Simons	ARUS Project Leader, Primary contact
Matt Lapine	Language support Project Leader
Jeff Wiener	Math Project Leader, Primary Math contact
Doug Ray	CMA Architecture contact

ARUS – The utility RTL

Requirements

- Define a set of portable operating system and utility routine interfaces

  "Raise the level of abstraction of the operating systems."

- Insulate LP and application programmers from OS and hardware specifics

- The first set of the routines must support the needs of FRS utilities and languages

- Provide implementations of the routines for PRISM ULTRIX and Glacier

- Provide support for application's direct use of VAX/VMS RTL routines, where feasible

ARUS – The utility RTL

Committments

- Be the driving force for AIA RTL–level architectures over time

  We are the run–time interfaces representatives on Scott Davis' AIA committees

- Interfaces and routines in the following areas for PRISM ULTRIX and Glacier FRS:
  - Memory allocation and deallocation
  - Condition handling
  - Date and time manipulation and formatting
  - Numeric conversions
  - String copying
  - String mapping
  - String formatting
  - Process information

ARUS – The utility RTL

Strategy

- A staged process
- Interface Design
  - Learn from 12 years of VAX/VMS RTL interface designs
  - Aim for robust, powerful capabilities for *our* products to layer on
  - Exploit Digital hardware / OS advantages at cost of not being able to port everywhere with no changes to underlying OS
  - Allow industry standards (as they firm up) to co-exist with or layer on the routines we provide
  - Exclude OS / Hardware specific concepts from interfaces
- Routine development
  - A completely new implementation
  - PILLAR is primary development language
  - Instrument existing RTLs to study usage patterns
  - For heavily used routines, simulate effects of different algorithms
  - Use VAX/VMS RTL algorithms where appropriate
  - Test on VAX/VMS as thoroughly as possible

ARUS – The utility RTL

Progress to date

- First of four interface design documents out for review
- Overall VMS RTL usage instrumented for several months at customer sites selected for approximate match to "typical" Glacier customer
- Memory management routines instrumented in house to allow algorithm simulation

Current status and activities

- Working on interface design
- Staffed at two engineers

ARUS – The utility RTL

Future plans

- Complete interface design early April
- Add one engineer early April, one more in mid–May

The remaining dates are for coded routines tested to the fullest extent possible on VMS and PRISM emulators. Full testing requires operating system support and hardware prototypes.

- July '88 (Mica BL3)
  - Memory allocation / deallocation routines
  - Some numeric conversion routines
- October '88 (Mica BL4)
  - Condition raising and handling (partial)
  - String copying routines
  - String mapping routines
  - Date and time manipulation
- December '88 (Mica BL5)
  - Remaining numeric conversion routines
- March '89 (Mica BL6)
  - Remainder of condition handling code

ARUS – The utility RTL

Dependencies

- Others' dependencies on this project
  - Virtually all layered products
  - Many bundled utilities

- This project's dependencies on others
  - Mica operating system development for stable interfaces
  - Following interfaces common between Mica and PRISM ULTRIX:
    - Memory management
    - Condition handling
    - Internal time format
    - RMS
  - Pillar/SIL compiler development and support

ARUS – The utility RTL

Problems, Risks, Unresolved Issues

- Doubly staged program:
  - Architecture and capabilities are staged
  - Implementation is staged across operating systems and hardware architectures

  Routines may not be available on VAX/VMS at PRISM ULTRIX or Glacier FRS

FORTRAN Language Support

Requirements

- Deliver run-time support for FORTRAN consistent with VAX/VMS and VAX/Ultrix FORTRAN implementations
- Support multithreading

Committments

- Full sequential I/O support
- Multithreaded I/O capability
- Support for some multithreading
- Miscellaneous built in functions

The following "don't gets" are driven by compiler and operating system limitations.

- *NO* Relative, indexed files (at Glacier FRS)
- *NO* Multithreading support (at PRISM ULTRIX FRS)
- *NO* Support for user program access to rms services
- *NO* PDP-11 compatibility

FORTRAN Language Support

Strategy

- Totally new implementation
- SIL/Pillar is development language of choice
- Use existing algorithms where appropriate
- Layer exclusively on AIA services
- Remove OS dependencies
- Provide multithread capability
- Test on VAX/VMS as much as possible

Progress to date

- I/O design nearly complete
- Several review cycles completed
- Error handling design nearly complete
- Remaining components understood

FORTRAN Language Support

Current status and activities

- Attempting to complete design documents
- Current staff: 1 (Matt LaPine)

Future plans

- April '88 (Mica BL2):
  - start coding
  - add second engineer
- July '88 (Mica BL3):
  - unformatted I/O working
  - add third engineer
- October '88 (Mica BL4):
  - formatted I/O working
  - error handling mechanisms supported
- December '88 (Mica BL5):
  - ready for PRISM ULTRIX
  - start on multithreading support
- June '89 (Mica BL7):
  - ready for Glacier

FORTRAN Language Support

Dependencies

- Others' dependencies on this project
    - FORTRAN compiler
    - Existing user applications (implicit and explicit uses)

- This project's dependencies on others
    - RMS services
    - Application Runtime Utility services
    - Common Multithread Architecture services

Problems, Risks, Unresolved Issues

- Availability of RMS at PRISM ULTRIX FRS uncertain
- Multithreading support at PRISM ULTRIX FRS unlikely, support at Glacier FRS not yet committed (consistent with FORTRAN compiler)

Pascal Language Support

Requirements

- Deliver run-time support for Pascal consistent with VAX/VMS Pascal implementation
- Support multithreading

Committments

- Provide support at Glacier FRS
- Studying PRISM ULTRIX support

Pascal Language Support

Strategy

- Totally new implementation
- SIL/Pillar is development language of choice
- Use existing algorithms where appropriate
- Layer exclusively on AIA services
- Remove OS dependencies
- Provide multithread capability
- Test on VAX/VMS as much as possible

Progress to date

- I/O design discussed, general model understood
- Error handling design partially understood

Current status and activities

- Project unstaffed

Future plans

- Plan in place to staff to meet requirements

Pascal Language Support

Dependencies

- Others' dependencies on this project
  - Pascal compiler
  - Existing user applications (implicit and explicit uses)

- This project's dependencies on others
  - RMS services
  - Application Runtime Utility services
  - Common Multithread Architecture services

Problems, Risks, Unresolved Issues

- PRISM ULTRIX support schedule uncertain

Math Library and Vectorization

Requirements

- Support scalar and vector math requirements of FORTRAN, Pascal and C
- Provide AIA compliant math routines

Committments

- Support FORTRAN
- Provide the VAX/VMS math run time library
- Vectorized BLAS routines
- Public domain BLAS scalar routines

- *NO* LINPACK
- *NO* EISPACK

Math Library and Vectorization

Strategy

- Staged process
  - Design new algorithms, both vector and scalar
  - Implement on VAX/VMS in assembler for performance
  - The VMS implementation tests the algorithm's performance and accuracy
  - Re–implement for PRISM systems–the algorithm is portable
- Development language is TBD, but leaning toward Pillar
- Support of FORTRAN requirements is highest priority
- AIA math routines are secondary in importance

Progress to date

- Algorithm design approximately 50% complete, including VAX/VMS implementation and scalar testing

Current status and activities

- Continuing on algorithm design and VAX/VMS implementation

Future plans

- Staffing plans are in place to address PRISM requirements

Math Library and Vectorization

Dependencies

- Others' dependencies on this project
  - FORTRAN compiler

- This project's dependencies on others
  - Pillar compiler

Problems, Risks, Unresolved Issues

- New algorithms
- Not necessarily bit for bit compatible results across hardware
- The packaging of EISPACK and LINPACK is not determined
- What is an AIA Math routine?
- Some ULTRIX math functions not part of the VAX/VMS math library

Common Multithread Architecture

Requirements

- Define an Architecture for VAX and PRISM systems to support multithreading within a process context
- Provide guidelines in adherence to this architecture

Commitments

- Support a full range of parallel processing applications and approaches
- Finalize Architecture soon to enable development and delivery for each system
  - required for Glacier FRS
  - not required for PRISM ULTRIX FRS

Strategy

- Eliminate all OS dependencies (this is an AIA component)
- Assure implementability and performance

Common Multithread Architecture

Progress to date

- Preliminary Requirements are completed
- Two functional review cycles are completed
- Staffing is complete

Current status and activities

- Review and final resolution of technical issues is in progress

Future plans

- Requirements Phase ends February, 1988
- Overall Architecture is nearing completion – April, 1988

Common Multithread Architecture

Dependencies

- Others' dependencies on this project
  - Architecture must be completed in time for RTL/OS implementations
  - Implementations must be available in time for their clients, such as FORTRAN, C, and Ada

- This project's dependencies on others
  - Finalization of Architecture requires review by OS people
  - Architecture depends on some other AIA components

SORT/MERGE

Requirements

- No stated requirements at Glacier Phase 0
- Vague requirements last time

Committments

- We will provide VAX/VMS SORT/MERGE on PRISM as required

Strategy

- SORT/MERGE is a port, not a re-implementation

Current status and activities

- Engineer assigned to VAX/VMS SORT/MERGE, will also do PRISM work

Questions

and

Answers

TALK OVERVIEW

- Quartz Goal Refinement
- Quartz Definition
- Quartz Schedule and Status
- Quartz Design
- Quartz Dependencies
- Quartz Challenges
- Your Questions

Cheyenne

Quartz

Digital Equipment Corporation
Confidential and Proprietary
Internal Use Only

Dave Schrader/Bob Gerber
Quartz
February 17, 1988
DECwest Quarterly Review

QUARTZ GOAL REFINEMENT

V1.0 Deliverables:

- Schedule: for Cheyenne V1.0 to FRS in Q2FY91, Quartz V1.0 must complete in Q2FY90.
- Data Integrity: Quartz must not corrupt data, must provide utilities to restore database integrity.
- Availability: requires Quartz configuration management and fast (parallel) recovery.
- Performance: To achieve TPS goals, Quartz/Mica must provide 2 D/C TPS/eVUP, use partitioned databases.
- Database Size: Quartz must effectively manage data placement, overheads, avoid hotspots.
- Extensibility: Quartz must manage 1-n Stones; must interface well to Rdb/Star
- Architecture: Quartz must fit seamlessly into DDTA and DDA.

ARCHITECTURE COMPLIANCE

DDTA - the DIGITAL OLTP Architecture

- Developed by OLTP Architecture Committee
- First cut published in January 1988
- First review held in January 1988
- Refinements underway; May 1988 target for completing first design

DDA - the DIGITAL Database Architecture

- Phase II: an opportunity to put the database platform in place for the 1990s
- DDA is the "glue" for many products
- Changes in DDA Phase II:
  - Support for client concurrency
  - Client/server model
  - Support for SQL2, ESQL
  - Tighter integration with CDD, utilities, DBA
- First cut definition 4/88; full definition 12/88

QUARTZ DEFINITION

- HOST INTERFACE: VAX and PRISM Quartz-specific communication functions as well as session management. Transfers requests for compilation (from application developers) and requests for transaction execution (from end users).
- QUERY PREPARATION: transforms an encoded form of a query into a request graph that can be executed by QUERY EXECUTION. Stores and retrieves precompiled queries. Interacts with Data Dictionary. Provides schema management.
- QUERY EXECUTION: executes request graphs. Includes transaction management, concurrency control, recovery, algebraic operators, indexing.
- OS INTERFACE: thin layer that provides relation and collection interfaces to QUERY EXECUTION. Handles faults, logical and physical configuration management.
- UTILITIES: several Quartz-specific tools available to DBAs. Console interface, installation, backup and restore, archiving, performance monitoring and tuning, error logging.
- TESTING: module and subsystem testing for Quartz; interface testing for Mica and Stone; interface testing for OLTP.

QUARTZ X0 .. X4

X0: 1/88

- High level cut at V1 features
- Conceptual design for Query Execution
- Initial design for Query Preparation, OS Interface
- Understand requirements for Host Interface, Utilities

X1: 7/88

- Run hand-compiled Debit/Credit through Query Execution components in a Mica/VMS/VAX environment.
- Complete conceptual design for Query Preparation and OS Interface.
- Initial conceptual design for Host Interface, Utilities.

X2: 1/89

- Run SQL D/C and DeWitt benchmarks on Pebble and Mica/VMS/VAX environments. Single box emphasis. Refine interfaces to Mica.
- Emphasis on Host Interface and Utility designs.

X3: 7/89

- Run both benchmarks on Pebble and Mica/VMS/VAX environments. Single box emphasis. Refine interfaces to Mica.
- Start tuning, optimizations.
- Design for multi-box Stone, configuration management.

X4: 12/89

- Production quality Quartz on multi-Stone, including performance monitoring, internationalization.
- Final check for architectural compliance with DDTA, DDA.
- Integration testing.

Refinements of X1 .. X4 features and schedules appear in the Quartz Baselevel Plan.

QUARTZ TOP-LEVEL SCHEDULE

- 01/88: Quartz X0 designs published
- 02/88: Quartz X0 external review
- 07/88: Quartz X1 + Mica/VMS stub + VAX
- 01/89: Quartz X2 + Mica/VMS stub + VAX
- 07/89: Quartz X3 + Mica BL6 + Pebble
- 12/89: Quartz X4 + Mica BL8 + Pebble
- 01/90: IFT starts (on Mica BL8 + Pebble)
- 04/90: Quartz X4 + Mica BL9 + Stone
- 06/90: EFT starts (on Mica BL10 + Stone)
- 12/90: FRS

Assumptions about Seattle deliveries:

- 12/88: Mica BL5/Pebble baselevel available to CXO
- 03/89: Mica BL6/Pebble baselevel available to CXO
- 06/89: Mica BL7/Pebble baselevel available to CXO
- 09/89: Mica BL8/Pebble baselevel available to CXO
- 12/89: Mica BL9/Pebble baselevel available to CXO
- 03/90: Mica BL10/Stone baselevel available to CXO

QUARTZ STATUS

- PROGRAM:
  - Discussions with OLTP DDTA architects
  - Phase 1 planning underway
  - Configurations, I/O subsystem study
  - Test and integration discussions started
- QUARTZ:
  - 21 Design documents (700 pages) released for review 1/88. Focus on Query Execution.
  - External reviews within DBS in 2/88.
  - Using internal and external consultants (DeWitt, Maier, Lomet, Bernstein, Brown.)
  - Implementation staged to reduce risk: X1, X2, X3, X4.
  - Coding of X1 has started.
  - Design ongoing in Query Preparation and OS Interface.
  - Mica interface work: transactions, host comm.
  - Early work on Host Communications and Utilities.

QUARTZ X0 DOCUMENTATION

Overview Documents

- □ Quartz Project Overview

Query Execution Subsystem Documents

- □ Query Execution Overview
- □ Partitioned Databases and Cheyenne Servers
- □ Database Cache Management
- □ Data Access Subsystem
- □ Query Execution and Communications Operators
- □ Transaction Management
- □ Database Scheduler

Query Preparation Subsystem Documents

- □ Query Preparation Overview
- □ Query Optimization Overview
- □ Query Preparation/Query Execution Interface
- □ Quartz View Processor Proposal
- □ Query Graphs: Internal Query Representation

Operating System Interface Documents

□ Operating System Interface Overview

□ Message Queues

□ Cheyenne Configurations

□ Backup/Restore System

□ Relation and Collection Managers Working Paper

Host Interface Documents

□ Host Communications Overview

Utilities Documents

□ Quartz Utilities Plan

QUARTZ X1 FOCUS: Query Execution Subsystem

- Contributes to achieving Cheyenne high availability by supporting partitioned databases that can be recovered quickly (<1 minute)
- Provides extendible, high performance via parallel, dataflow operations on partitioned databases. Cheyenne simulation results:
  1. 85% DC performance improvements per additional Stone
  2. 2 TPS/VUP
  3. 3 Stones yield 500 TPS easily
- Improves the reliability of Cheyenne via self-checking data structures that are fire-walled in a number of distinct protection spheres.
- Ensures the logical integrity of databases.
- X1 Query Execution Prototype runs Debit Credit Workload (7/88)

QUARTZ Query Execution

Table 1: Design Considerations

Design Constraint	Cheyenne Benefit Provided
Partitioned Databases	High Availability, Extendible Performance
Fast, Parallel Database Recovery (< 1 Minute)	High Availability
Fine Granularity Synchronization	High Performance OLTP Workloads
Protected Data Structures	Enhanced Data Integrity
Hash-Partitioned Algorithms	Make Partitioned Databases Feasible
Effective MICA Usage	Reliability and Performance
Highly Parallel Query Executions	High Performance

QUARTZ Query Execution

Database Cache Features

- I/O reduction via domain-based page replacement policies
- Support for large database caches (> 128 MB)
- Highly concurrent access to data structures
- Variable size database pages
- Transparent support for long-running transactions
- Frequent, fuzzy checkpoints

Data Access Subsystem Features

- Dynamic Indices: B+ trees & extendible hashing
- Highly concurrent index reorganizations
- Partitioned database support
- Clustered relations
- Phantom tuple prevention
- Multiversioned tuples
- Reduced pathlength operators

QUARTZ Query Execution

Database Operator Features

- Transparent support for partitioned databases
- Hash-partitioned operators
- Highly parallel operators
- Dataflow pipelines for operators
- Low activation and control costs
- Extended relational functionality

Database Scheduler Features

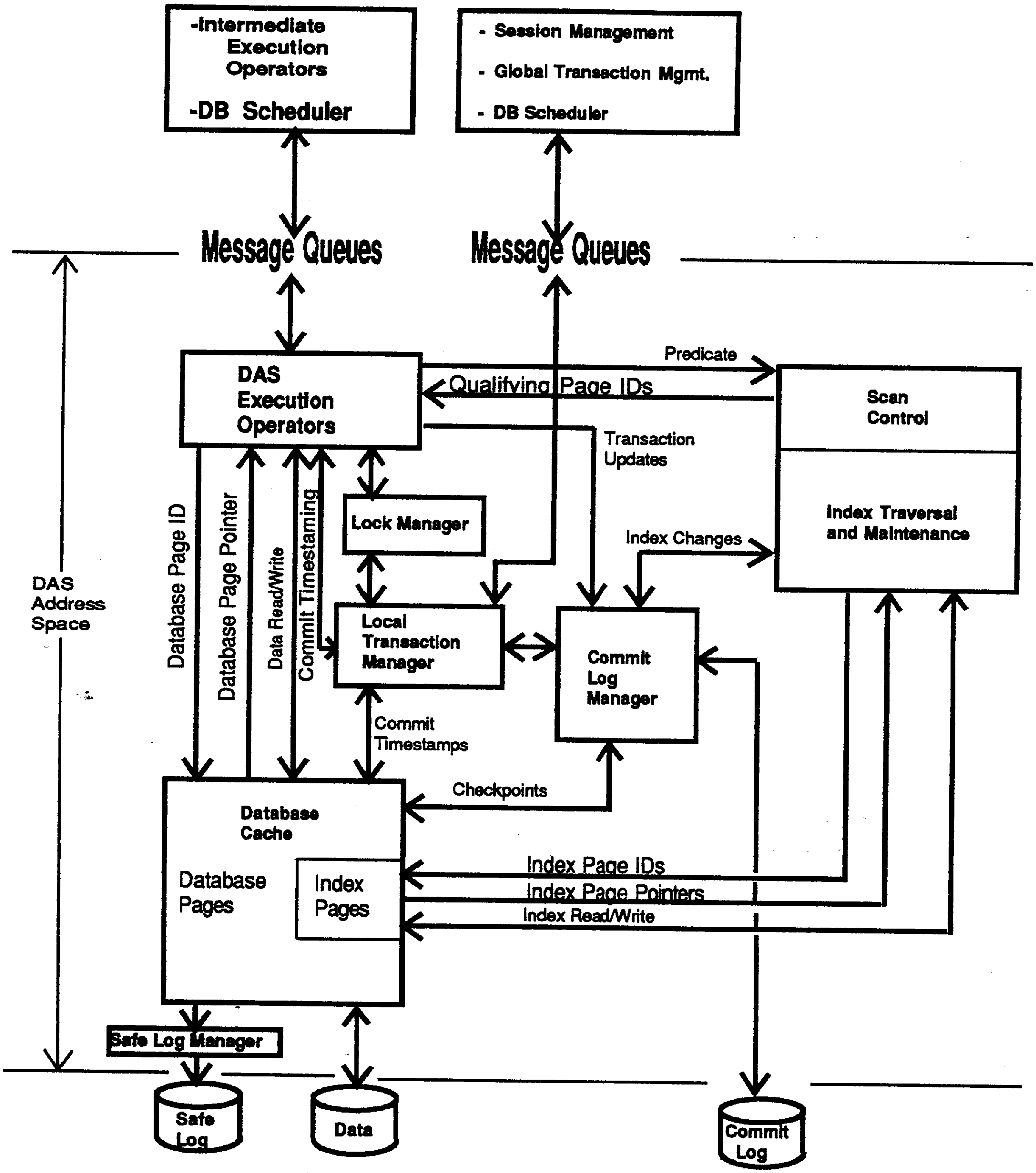
- Low cost dataflow activations
- System level load control
- Parallel execution of requests
- As-needed, dataflow operator activations
- Provides partitioned database transparency to operators
- Operator placements
- Execution time optimizations

QUARTZ Query Execution

Transaction Management Features

- Two phase locking protocols (localized)
- In-cache workspace model
- REDO-only logging (primarily)
- Fuzzy database checkpoints
- Fast, parallel database recovery
- Atomic database page writes
- Nested transactions
- Deadlock detection via timeouts
- Escrow solution for database hotspots

QUARTZ X1 PROTOTYPE



Note: Labels on arrows denote predominant information flow only, and are not exhaustive.
Interface between storage devices and storage system is procedure calls into SHARD library.
Recovery System interfaces not shown.

QUARTZ DEPENDENCIES

- Internal Engineering
  - Mica
  - Pebble, Stone
  - DBA Tools
- External Engineering
  - OLTP Architecture
  - DDA Architecture
  - CDD
  - TP-Builder, ACMS, DEC/Intact
  - V2: DSA-2, HSM
- Product Management
  - Configurations
  - Benchmarks
  - Pricing, selling strategies

QUARTZ - MICA INTERDEPENDENCIES

- Transaction Services
- Configuration Options
- System Management
- Prototype Plans
- Testing and Integration Plans
- IPC
- Host Communication
- Configuration Management
- Tapes
- SIL, Pillar

QUARTZ CHALLENGES

- Building a highly integrated product
  - Close monitoring of OLTP program
  - Early OLTP testing/integration plan
- Achieving high availability goals
  - Fault probabilities
  - Quartz/Mica software fault modeling
  - OLTP integration testing
- Achieving performance goals
  - Pathlength budgets, 2 D/C TPS/VUP
  - Inter-Stone communications cost
  - Hardware modeling, 15-20 effective VUPs/CPU

PRISM
Documentation Status

Digital Equipment Corporation
Confidential and Proprietary

Ken Western
DECwest Publications

PRISM Documentation Status

- Overall documentation status — Ken Western, DECwest

- Flint documentation status — Michael Tardiff, DECwest

- Quartz documentation status — Rose Johnston, Colorado Springs

Glacier Documentation Status

- Preliminary Glacier Master Documentation Plan available
- Preliminary Glacier Customer and Support documentation plans in progress
- Individual Glacier manual documentation plans in progress. All due for completion by the end of April
- Some manuals being written. All writers will begin writing no later than the beginning of May.

PRISM Documentation Priorities

- Finalize Flint documentation planning
- Finalize Glacier Customer and Support documentation plans
- Ensure our plans are synchronized with our external partners: Colorado Springs (Cheyenne), SDT (Glacier and Flint), UEG (Flint)
- Ensure our plans are reviewed by the internationalization groups

Flint Documentation

17 February 1988

Michael Tardiff
Digital Equipment Corporation

Documentation Deliverables

- Reference information
  - "Manpages," quick reference
  - Error messages
  - Language references
  - Revised and improved, but compatible
- Utility descriptions
  - awk, dbx, mail
  - Borrow from Supplementary Documents, write anew
- Procedural descriptions
  - System setup, day-to-day care
  - Migration, porting
  - Tuning, diagnosing
  - Writing drivers

Documentation Deliverables

Continued

- Tools synthesis
  - Shell scripts, programming aids
  - Not simplistic
- Cross-referencing
  - Reader's guide, roadmap, packaging
  - Good indexes
- Conceptual information
  - UNIX system principles
  - Introductory material

Where We Are Today

- Revising documentation options document
- Analyzing ULTRIX-32 Version 2.2 docset
- Researching competitor's docsets
- Tracking development
- Focusing on requirements
- Ranking requirements by priority
- Choosing deliverables
- Investigating visual improvements to docset

What Comes Next

- Provide ULTRIX orientation for doc group
- Resolve relationship with CUP/ULTRIX
- Draft documentation plan
- Identify resources
- Choose production tools

Cheyenne

ESDP

Digital Equipment Corporation
Confidential and Proprietary
Internal Use Only

Rose Johnston
ESDP/CXO DB Systems
February 17, 1988
DECwest Quarterly Review

Short-Term Deliverables

- July 1988
  - Updated documentation plan
  - Competitive analysis
  - Quartz glossary

- December 1988
  - Outlines

Long-Term Deliverables
Cheyenne
FRS Documentation

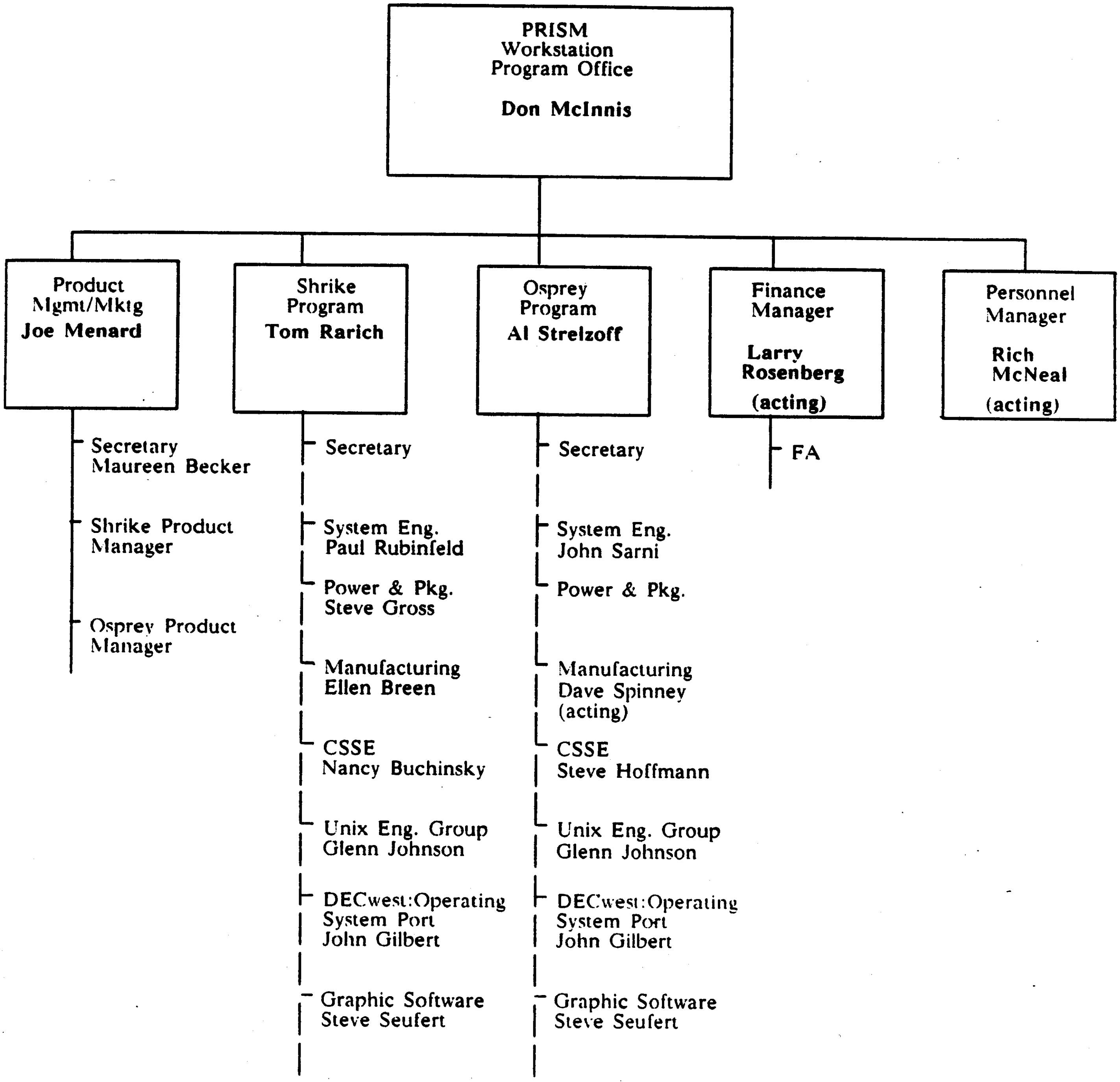
Deliverable	Information Available	First Draft Available
Summary Description	Continuous	Oct 89
System planning, configuring, installing, upgrades		
Quartz	Mar—Jun 89	Jun—Sep 89
System performance		
Quartz	Jun—Oct 89	Aug—Dec 89
System maintenance, operations		
Quartz	Mar—Jun 89	Sep 89—Dec 89
System security		
Quartz	Sep—Oct 88	Oct 88—Nov 88
Database design and creation	Jun—Aug 88	Aug 88—Oct 88
Database use	Jun—Aug 89	Dec 89—Feb 90

ESDP Dependencies

- Quartz/Mica/Stone development
- DDA Phase II completion
- Language interface: ESQL, SQL
- OLTP requirements
- DECwest documentation
- Tools
  - VAX DBA
  - Conceptual/Logical Database Designer

PROGRAM OFFICE ORGANIZATION CHART

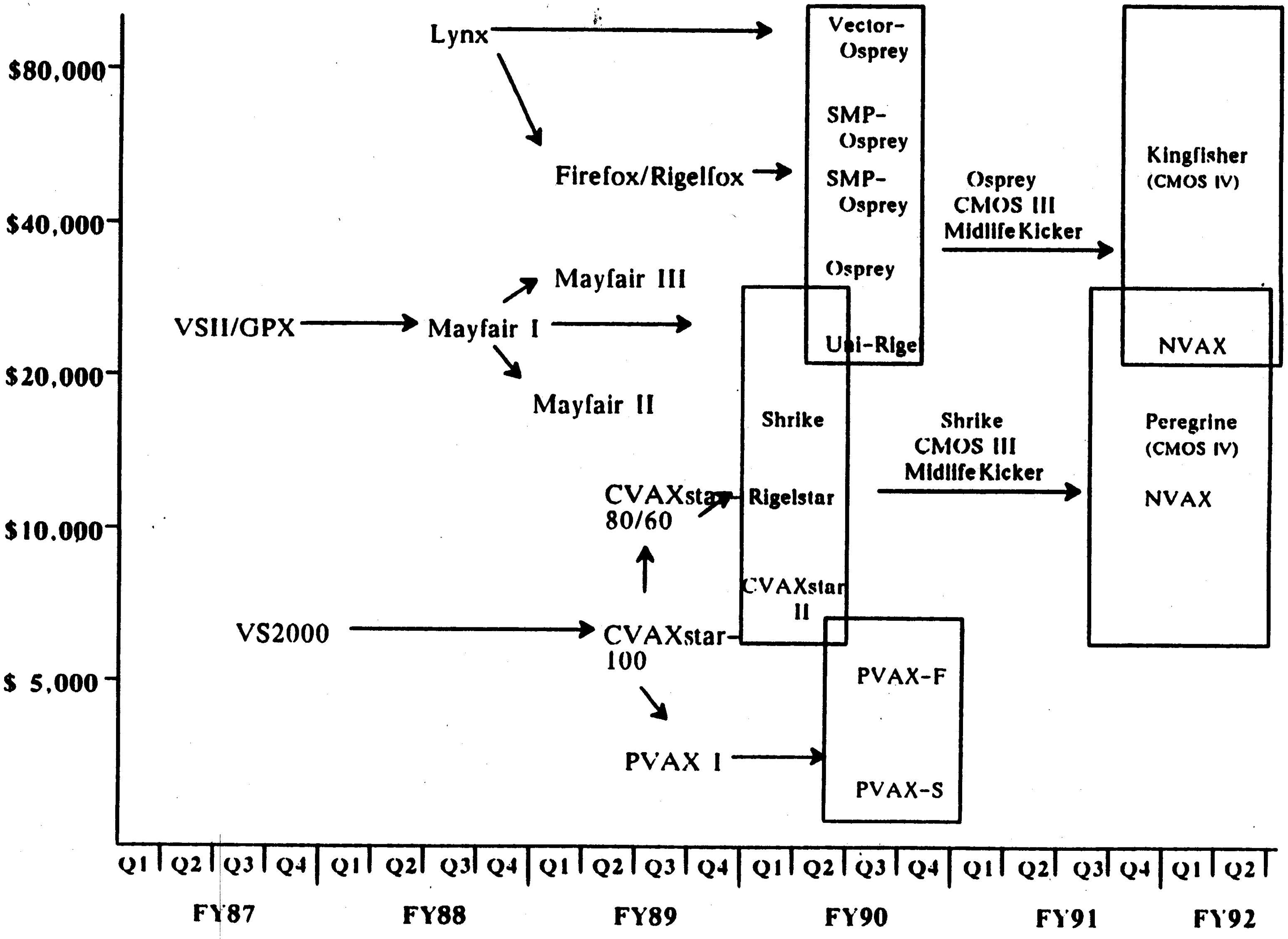
Proposed Program Office Organization Chart



PRISM Workstation Program Office
Don McInnis

- Product Mgmt/Mktg
  Joe Menard
  - Secretary
    Maureen Becker
  - Shrike Product Manager
  - Osprey Product Manager
- Shrike Program
  Tom Rarich
  - Secretary
  - System Eng.
    Paul Rubinfeld
  - Power & Pkg.
    Steve Gross
  - Manufacturing
    Ellen Breen
  - CSSE
    Nancy Buchinsky
  - Unix Eng. Group
    Glenn Johnson
  - DECwest:Operating System Port
    John Gilbert
  - Graphic Software
    Steve Seufert
- Osprey Program
  Al Strelzoff
  - Secretary
  - System Eng.
    John Sarni
  - Power & Pkg.
  - Manufacturing
    Dave Spinney (acting)
  - CSSE
    Steve Hoffmann
  - Unix Eng. Group
    Glenn Johnson
  - DECwest:Operating System Port
    John Gilbert
  - Graphic Software
    Steve Seufert
- Finance Manager
  Larry Rosenberg (acting)
  - FA
- Personnel Manager
  Rich McNeal (acting)

WORKSTATION STRATEGY


$80,000
$40,000
$20,000
$10,000
$ 5,000
Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2
FY87
FY88
FY89
FY90
FY91
FY92

Lynx

Firefox/Rigelfox

VSII/GPX

Mayfair I

Mayfair III

Mayfair II

VS2000

CVAXstar-100

CVAXstar 80/60

PVAX I

Vector-Osprey

SMP-Osprey

SMP-Osprey

Osprey

Uni-Rige

Shrike

Rigelstar

CVAXstar II

PVAX-F

PVAX-S

Osprey CMOS III Midlife Kicker

Shrike CMOS III Midlife Kicker

Kingfisher (CMOS IV)

NVAX

Peregrine (CMOS IV)

NVAX

Product Strategy

Product	Entry	Typical
Shrike	Diskless 15" Monochrome 1280 x 1024 16 Mbyte MLP: $19,900	400MB 19" 8 plane color 1280 x 1024 32 Mbyte MLP: $36,290
Osprey	Diskless 19" Monochrome 1536 x 1152 16 Mbyte MLP: $29,900	800MB 19" 8 plane color 1280 x 1024 32 Mbyte MLP: $48,290
Osprey Server	400MB Disk 64MB Memory MLP: $47,680	1200MB Disk 128MB Memory Vector Option MLP: $97,070

Prices include 1 year warranty

Calendar of Product Deliverables

	Target FRS	Package	CPU	Disk	Bus
Shrike	Q4 FY89	New	CMOS II PRISM	RF	NU
Osprey	Q1 FY90	New	Binned CMOS II PRISM	RF	XMI, NU
Shrike II	1H/FY91	Shrike	CMOS III PRISM	RF	NU
Osprey II	1H/FY91	Osprey	Binned CMOS III PRISM	RF	XMI, NU
Peregrine	H2/FY92	New	CMOS IV PRISM	TBD	TBD
Kingfisher	H2/FY92	New	Binned CMOS IV PRISM	TBD	TBD

Competitive Positioning

Feature	Shrike June 1989	SUN 4/110 February 1988
Memory	16MB	8MB
	64MB max	32MB max
Performance		
MIPS	12-15 VUPS	7 MIPS
DP.LINPACK	3 MFLOPS	.8 MFLOPS
2D VEC/SEC	100-250K	
3D VEC/SEC	100-200K	
POLY/SEC	20-60K	
Disk		
Entry	150MB	141MB
Best	400MB	327MB
Tape	1/2 in.	1/4 in.
Price		
B/W, No Disk	19.9	18.9
B/W, 150MB + Tape	26.4	27.9
Color, No Disk	23.5	23.9
Color, 150 MB	30.0	32.9

External Product Positioning

Projected SUN product line thru CY90

Product Class	CY87 (actuals)	CY88 (estimate)	CY89 (estimate)
Low End	SUN 3/60C 141 MB 60 MB Tape 8 MB (Parity) 2.5 VUPs $21,400	SUN 4/60C 280 MB 60 MB Tape 16 MB (ECC) 5-6 VUPs $28,000	SUN 5/60C 400 MB 60 MB Tape 16 MB (ECC) 8-10 VUPs $28,000
Midrange	SUN 3/260C 280 MB 60 MB Tape 16 MB (Parity) 3 VUPs $55,500	SUN 4/260C 400 MB 60 MB Tape 32 MB (ECC) 6-8 VUPs $55,500	SUN 5/260C 600-800 MB 100 MB Tape 32 MB (ECC) 12 VUPs $50,000
High End	SUN 4/260C 280 MB 60 MB Tape 32 MB (ECC) 6-7 VUPs $77,900	SUN 5/260C 600-800 MB 100 MB Tape 32 MB (ECC) 12 VUPs $75,000	SUN 6/260C 1.0-1.2 GB 300 MB Tape 64 MB (ECC) 20-24 VUPs $75,000

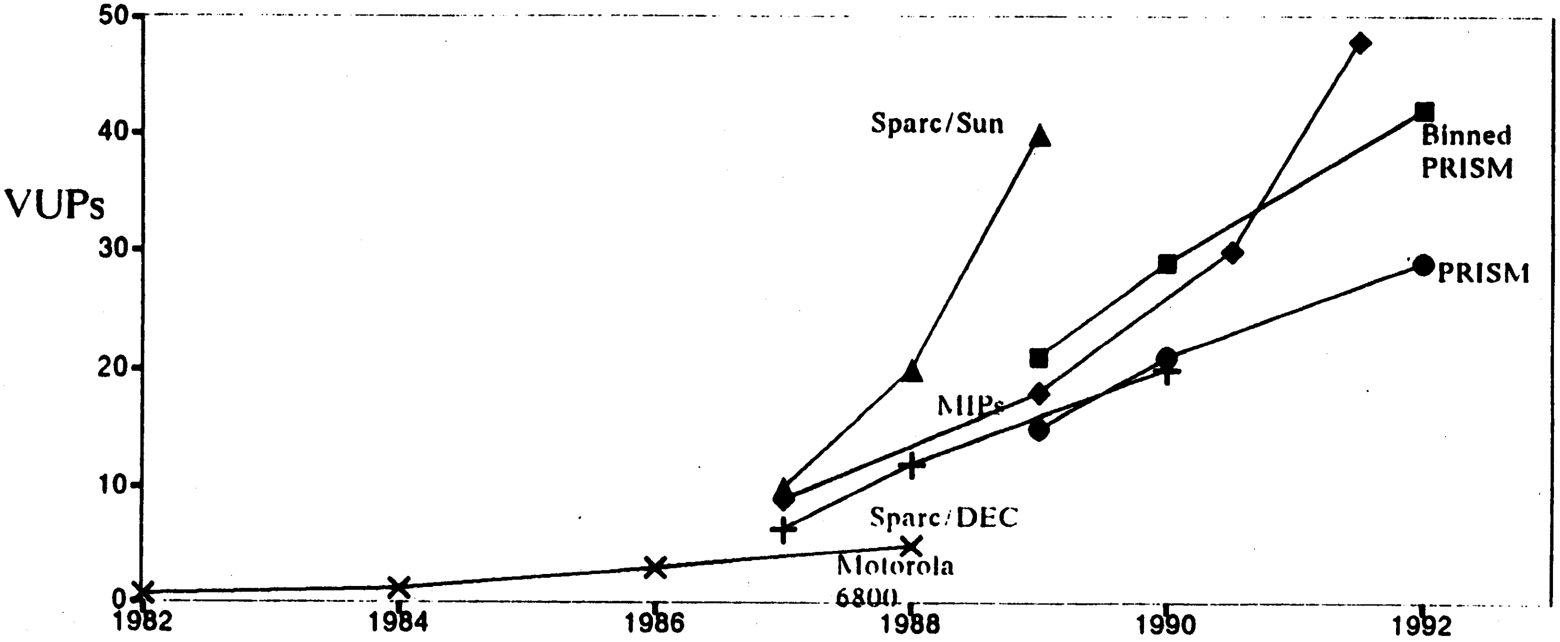
1. Systems all contain 8 plane color graphics.

2. Price rise in low end due to higher memory and use of ECC, and addition and enlarging of caches.

3. SUN 6/260C performance claims in FY89 were deemed too aggressive based on technology risks of CPU chip - 6-12 months at 20-24 VUPs seems more realistic.

SUN 5/260C performance may reach only 12 VUPs based on recent analysis of Cypress chips. Projections for SUN 6/260 do present DIGITAL with a marketing problem that we must resolve, even if they do not deliver on time.

Chip Performance Projections



Note 1: Sparc/Sun line represents Sun claims. Sparc/DEC line represents DEC estimate of what Sun will achieve.

Note 2: MIPs line represents "binned MIPs parts"

Figure 1: Overall Electronics Block Diagram

```
+------------+            +-------------+                               +-----*------+
| Memory     |            |             |                         |     | SCSI       |     external
| Card  1    |<>-+        |  CPU Card   |<>-+                     |<--->| Interface  |<--> connector
| (standard) |   |        | (standard)  |   |                   C |     |            |
+------------+   |        +-------------+   |                   V | a   +------------+
                 |                          |                   A | k
+------------+   |            +-*-+         |         +-*-+     X | a   +------------+
| Memory     |   |   MD bus   | B |         |         | B |       |     | Color      |     external
| Card  2    |<>-+----------->| U |<--------+-------->| U |<------+---<>| Video Card |<--> RGB BNCs
| (optional) |   |            | F |                   | F |       |     | (Optional) |
+------------+   |            +---+                   +---+     B | I   +------------+
                 |                                              U | I          |
+------------+   |                                              S | 3   +-------------+
| Memory     |   |                                                | 2   | Graphics    |
| Card  3    |<>-+                                                +---<>| Accelerator |
| (optional) |                                                    |     | (optional)  |
+------------+                                                    |     +-------------+
                                  +---*--+   +-----*------+       |
                                  | VRAM |   | GSSC       |       |
                                  |      |<->|            |<----->|     B/W video and        external
                                  |      |   |            |<------|-------------------------> connectors
                                  +------+   +------------+       |       4 DZ lines
                                                  |               |
                                                  V               |     +------*-----+
                                             +-----*------+       |     | Ethernet   |-----> Thick wire
                                             | PROM       |       |<--->| Interface  |  or
                                             | Boot and   |------>|     | Logic      |-----> Thin wire
                                             | Diag. ROM  |       |     +------------+
                                             +------------+       |
                                                  |               |     +------*-----+
                                             +-----*------+       |     | DSSI       |     external
                                             | EEPROM     |       |<--->| Interface  |-----> connector
                                             |            |<----->|     | Logic      |
                                             |            |       |     +------------+
                                             +------------+       |
                                                                  |     +-------------+     Future
                                                                  |     | Public      |     public
                                                                  +---<>| Bus Adapter |<---> bus such
                                                                        | Card        |     as NU Bus
                                                                        | (optional)  |
                                                                        +-------------+
```

Figure 2: Physical Board Electronics

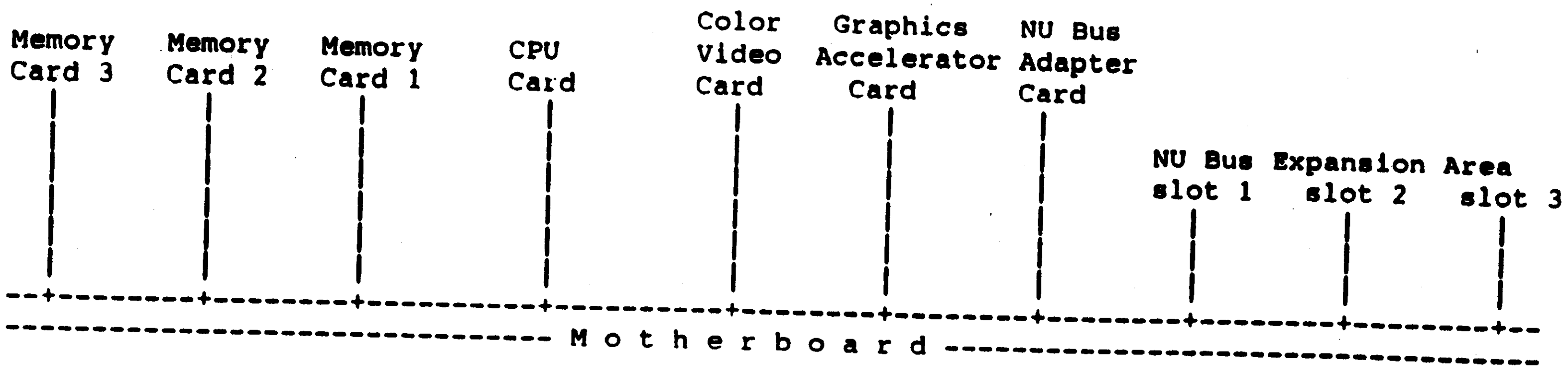


Figure 3: CPU Card Block Diagram

```
    +--------+
    | 160Mhz |
    |  Osc.  |
    +--------+
         |
         v
  +-------------+            +------------+
  | CPU Chip    |            | Floating   |
  |             |            | Point      |
  |             |            | Chip       |
  +-------------+            +------------+
        ^                          ^
        |        PDAL bus          |
        +--------------+-----------+
                       |
      +----------------+-------------+
      |                              |
      v                              v
+-------------+                +------------+           +------------+
| C_hi Chip   |                | C_lo Chip  |           | Cache      |
|             |<---------------|            |<--------->| Tags       |
|             |                |            |           | RAMs       |
+-------------+                +------------+           +------------+
      ^                            ^     ^
D_hi  |                        A   |     | D_lo
bus   |       +-------------+ bus  |     | bus
      |       | Cache       |      |     |
      |<----->| Data        |<-----+     | these three busses are
      |       | RAMs        |<-----|---->| collectively called the Cbus
      |       +-------------+      |     |
      |                            |     |
      +----------+   +-------------+     |
                 |   |                   |
                 |   |   +---------------+
                 |   |   |
                 v   v   v
              +------------+
              | Bus        |
              | Interface  |
              | Chip (BIC) |
              +------------+
                     ^
                     v
              To Mother Board
```

Digital Equipment Corporation - Confidential and Proprietary

Shrike Development Plan

Milestone	Date
Package Concept Selected	1-12-88
Phase 0 Exit	2-19-88
System Functional Spec Revision 1	2-25-88
Full Electronics Behavioral Model	3-7-88
Package Design Complete	6-15-88
Phase 1 Exit	8-88
Power On CPU, Memory, Motherboard	9-26-88
Power On 8 Plane Graphics	TBD
Power On NuBus Adapter	TBD
Power On GFPA	TBD
UNIX Boot	10-24-88
DVT Start	12-19-88
Field Test Start	3-13-89
RQT Start	3-13-89
FRS Goal	6-27-89

Prototypes

	Shrike	Osprey
Quantity Planned	130	130
Power On	9-26-88	12-88
Quantity Needed		
ZSO	4 in Aug	3 in Nov
CSSE	5 RQT Level	TBD
DVT/RQT/LSEE	94 Dec - Mar	TBD
Diagnostics	6	TBD
Packaging	10 each level	TBD
Manufacturing	TBD	TBD
Field Test	30 in Mar '89	TBD
Other	TBD	TBD
Proto Build Location	Alberquerque	TBD

Seed Units

CVAX/Firefox/Mayfair III Plans:

Base Prod Marketing	20
Technical Marketing	5
3rd Party Marketing	30
US area	10
Europe	70
GIA	5
Field Announcement Day	30
Application Centers for Technology	15
Misc.	10
	195

System Integration

System Elements	People	Location
Power & Packaging	Steve Gross Ron Piuze	ML
8 New VLSI Chips	Bob Supnik	HL
CPU Card/Motherboard/NuBus Adapter	Paul Rubinfeld	HL
Memory	John Sangermano	SHR
Graphics HW - FPA/8 Plane/28 Plane	Art Lim	LMO4
Kernel Package DVT	John Cyr	ML
Diagnostic SW	John Caporal	ML
PRISM ULTRIX	Dave Ballenger	ZSO
DECwindows/Utilities	Kent Ferson	ZKO
PEX, DDX Graphic SW	Steve Seufert	LMO4

Issues

- Systems integration responsibility
- Staffing
- Schedules
- Graphics board/Package design
- DMA on Shrike
- Diagnostic strategy
- DSSI peripherals
- Performance needs for I/O and bit map graphics
- Input device support
- SWS/Sales training on UNIX
- Product pricing for market

Overall hardware status

- System definition stable
- Single focus: - MORAINE
- Phase zero exit done
- Schedules almost complete
- Methodology document first pass distributed
- Electronics spec reviews Jan/Feb
- Some problems with staffing

rts 2/17/88

Moraine system features

- 1 to 4 processors, each with vector unit
- 1 to 4 memory modules, 64 Mbytes each
- 2 14 slot XMI card cages

Performance:

- 15 to 60 times 780 scalar performance
- 48 to 192 MFLOPS peak
- 7 to 30 MFLOPs LINPAC
- 64 to 256 Mbytes of memory, 1 Gbyte with 4Mbit chips
- 90 to 360 Mbytes/sec memory bandwidth
- 120 Mbytes/sec I/O bandwidth
- Greater than a year MTBF

rts 2/17/88

Moraine Technology

- PWBs
    - All polyimide
    - Buried but not blind VIAs
    - CPU, Memory:
        - Extended Hex
        - 16 layers
        - Double sided surface mount
        - Actives on side 2
        - Intrusive 536 pin Argonaut style connector

    - Backplane:
        - 18.2" by 19.4"
        - Not more than 20 layers (routing study)
        - SMT ASICs
        - SMT passives
        - Press pin Argonaut style connector
    - XMI modules

rts 2/17/88

Technology - contd

- SSI/MSI - SOIC and SOJ 50mil SMT
- Caps under SOJs on memory
- ASICs:
    - 12.5 mil HPTP
    - 224 pin 25mil SMT cerquad
    - 164 pin 25mil SMT cerquad
    - 132 pin 25mil SMT cerquad
    - 68 pin 25mil SMT cerquad
    - 44 pin 50mil SMT CLCC

rts 2/17/88

Major Milestones

Milestone	Plan	Actual
System Spec first pass	Nov 1987	Nov 1987
Module specs	Jan 88	Feb 88
Spec review	Jan 88	Feb 88
Four emulators to software	Jan 88	Jan 88
Moraine system spec complete	Mar 88	
Package mock up built	Apr 88	
First Pebble proto	Jul 88	
Proto package complete	Oct 88	
Pebble to Software group	Oct 88	
Moraine package,power,MEM,I/O	DEC 88	
First complete Moraine	Feb 89	
Moraine to software	May 89	
Moraine Field test	Sep 89	
Moraine FRS	Mar 90	

rts 2/17/88

CPU

- 1 module design
- 4 ASIC designs, All HPTP, SBG
- 7 engineers

Status:

- Module spec pass 1 complete
- Spec review done
- High level block diagram done
- Preliminary timing done
- ASIC block diagrams > 50% complete
- Module schematics started
- Beh model started

rts 2/17/88

Memory and crossbar

- 1 module design
- 5 ASIC designs, all cerquad, SOG
- 7 engineers

Status:

- Spec pass 1 done
- Spec review held for memory
- Block diagrams complete
- Preliminary module schematics done
- Preliminary layout done
- Preliminary dispersion pattern done
- 1st pass beh model done/ not physically accurate
- Crossbar
  - Block diagram done
  - Spec pass 1 done
  - Spec review Feb
  - 1st pass module schematics done, in CXO
  - 1st pass ASIC schematics done
  - Preliminary timing done

rts 2/17/88

I/O and console

- 2 modules, YIA, YIB
- 4 designs, 1 HPTP, 3 cerquad
- 7 engineers

Status:

- Specs pass 1 done
- SP design review held
- SP breadboard will be built in Feb
- Spec review YIA/YIB Feb
- YIA
  - Block diagrams complete
  - Beh model just started
- YIB
  - Block diagrams complete
  - Beh/struct model > 80% complete
  - Model testing started
- YII cable modelling started

rts 2/17/88

Package

- Cabinet, cooling, cardcages, etc,etc
- 6 Engineers
- Concepts defined for:
  - Cabinet structure
  - Cooling system
  - Power supply rack
  - MG set mechanical
  - Cable routing
- I/O Panel mock up complete
- Using Calypso XMI mounting
- Backplane cover done, sent to CXO
- Blower selected

rts 2/17/88

Power system

- Regulators
- Distribution/Bussing
- EMM
- AC input
- MG set
- EMI/RFI
- 4 engineers

Status:

- DECwest/CEAG Schedule complete
- Power system spec distributed
- Held spec review with CEAG
- EMM Spec distributed
- EMM schematics complete
- EMM spec review Feb
- MG vendor selected
- di/dt modelling ongoing

rts 2/17/88

Technology status

- Too much work to list
- 7 engineers
- Technology overviews done
- Set up contacts with external groups
    - HPTP
    - Semiconductor business group
    - ASIC center
    - Physical technologies
    - Backplane routing tests ongoing
    - Preliminary clock spec done
    - Testcase plan started
    - Spice modelling ongoing

rts 2/17/88

Diagnostics

- 13 tests and exercisers
- 4 people, and 3 on loan from MNFG

Status:

- Project plan distributed
- Instruction test spec distributed and reviewed
- Instruction test coded 90%
- Test strategy document distributed
- Modelling test plan distributed and reviewed
- Pebble diagnostic plan distributed
- Specs in progress for:
  - CPU test
  - MMI test
  - I/O test

rts 2/17/88

CAD

- Provide design environment
- Modelling and verification
  - Performance model
  - Behavioral model
  - Structural model
  - Timing verification
- Test pattern generation
- Produce manufacturing data bases
- Support
- 6 engineers

rts 2/17/88

Pebble

- Pebble is a product (Osprey CPU)
- 1 XMI module
- 2 ASIC designs, both cerquads
- 2.5 engineers

Status:

- Formal spec started
- Design changed to reduce design objects
- One LSI and one SBG ASIC
- Board schematic 95% complete
- XPPGA structural design complete
- Cache block diag complete
- Cache model complete
- Structural verification started
- Diagnostic plan written

rts 2/17/88

Risks

- Technical:
  - A huge amount of work
  - HPTP
  - Active backplane
  - CPU module real estate
  - Complexity - Vector unit, cache coherency
  - Architecture
- Business:
  - PNO stage 1 capability to build protos
  - Coordination with other programs
  - Staffing

rts 2/17/88

Goals for next quarter

- All specs complete and reviews held
- Moraine system spec distributed
- System behavioral model without vectors
- Package mockup complete
- All schedules in VUE
- Pebble design complete and release started

rts 2/17/88