Date: February 1979

Revision of
Diagnostic Engineering Project Plan for

DOLPHIN

Issued by:

____
Dale C. Cook -- Supervisor

____
Jim Jones -- Project Leader

Approved by:

____
Vic Ku -- Funding Manager

____
Richard P. Maliska -- Development Cost Center Manager

Reviewed by:

Tom Eggers	Jud Leonard
John Grose	Don Lewine
Paul Guglielmi	Mike Robey
Peter Hurley	Pat Sullivan
John Kirchoff	Phil Wilson R.I.P.
Alan Kotok	Sultan Zia

and other members of the Dolphin team

Revision History:

Rev 1 issued 27-FEB-78 by Jim Jones

Contents

Section	Topic	Page
1	Executive Summary	1
1.1	The Dolphin System	1
1.2	The Dolphin Diagnostic Product	1
1.3	Product Audience	2
1.4	Success Criteria	2
1.5	Risks	3
1.6	Scope of the Project and this Document	3
2	Description of Diagnostic Product	4
3	Goals and Non-goals	5
3.1	Functionality Goals	5
3.1.1	MCA Support	5
3.1.2	Console Software	5
3.1.3	Console-based Diagnostics	6
3.1.4	Dolphin-based CPU Diagnostics	6
3.1.5	Dolphin-based Peripheral Diags.	7
3.1.6	System Exerciser	7
3.1.7	Quick-verify Support	7
3.2	Performance Goals	8
3.2.1	Fault Detection	8
3.2.2	Fault Resolution	8
3.2.3	Runtimes	8
3.3	Non-goals	9
3.3.1	32-bit Support	9
3.3.2	Additional Console Peripherals	9
3.3.3	MCA Incoming Inspection Tests	9
3.3.4	CPU Board Tests	9
3.3.5	Systems with Bus Repeaters	9
4	Operating Environment	10
4.1	Hardware	10
4.2	Software	10
5	Interdependencies with Other Projects	11
5.1	Hardware	11
5.2	Software	11
6	Compatability and Standards	12
7	Product Review Method	12
8	Component Plans	12
9	Development Staffing and Funding	13
9.1	Personae	13
9.2	Manpower Loading	14
9.3	Funding Required	15
9.4	Hardware Requirements	15
10	Implementation Plan	16
11	Documentation Staffing and Work Plan	17
12	Quality Assurance Plan	17
13	Distribution Plan	17
14	Training Plan	17
15	Maintenance Plan	17

1. Executive Summary

1.1. The Dolphin System

Dolphin is an advanced computer system based on a new generation CPU incorporating Macro Cell Array technology. It is the logical successor to the KL-10 based PDP-10/20 system, and may well become the high-end VAX system. From a diagnostic point of view, the Dolphin system will be capable of more complete detection and resolution of faults than any previous DEC computer system. From an availability point of view, Dolphin has more built-in error correcting and recovery hardware than any previous system. These two aspects of the system should make it possible for DEC to penetrate new areas of the commercial market place and set new standards of system reliability, availability and maintainability.

1.2. The Dolphin Diagnostic Product

The Dolphin Diagnostic Product consists of a comprehensive set of tests and support software to allow complete diagnoses of faults anywhere in the Dolphin system. The package includes software which runs in the PDP-11 based console processor and software which is executed by the Dolphin processor. Part of the product is intended to run in an operating time sharing system environment (the On-line component). Part is used to find problems when the system is "down" (the Stand-alone component). Some of the testing is aimed at resolving faults to the failing MCA chip, other tests the system functionally, and, at the highest level, some testing is aimed at forcing system interaction problems by exercising the system busses.

An important part of the Diagnostic Product for Dolphin is the console support software. This software will provide for system start-up (loading of all relevant microcode and bootstrap code, configuration of MCA's, busses, controllers, and memories), system power control, operator interface and control, run-time support and display, as well as complete support for the the diagnostics which execute out of the console processor and centralized error handling for diagnostics which run on the 10 side.

1.3. Product Audience

The Dolphin Diagnostic Product has really three audiences. Initially, the tests will be used by Large System Hardware Engineering to establish the working design of the system. It is anticipated that this will require considerable interaction and flexibility in using preliminary versions of the software. Next the product will be used by Large System Manufacturing to bring up the first Dolphins for internal use in Design Maturity Testing and for use as Quick Verify stations - also a process involving considerable interaction and support between the two groups. Then the Diagnostic Product must provide for volume production of Dolphins in the most cost effective manner possible. As systems are shipped to the field, Field Service becomes the key audience - both as on-site users and via the Remote Diagnostic link built-in to the Dolphin system. It is always a goal of Large Systems Diagnostics to support these three audiences and make intelligent compromises which do not facilitate one user at the expense of the others.

1.4. Success Criteria

It is a primary goal of the Dolphin Diagnostic Project to make full use of the advanced diagnostic hardware features of Dolphin to achieve the highest RAMP perforance possible. We will know we have succeeded if: 1) the system comes up smoothly in the Engineering environment, 2) systems can be built to meet Manufacturing standard cost goals, and 3) systems are maintainable in the field at realistic warranty costs. To aid in early evaluation of our success, we intend to make use of hardware and software fault simulation to the maximum extent possible. Unlike the KL-10 environment, much of the Dolphin logic is buried in the MCA chips, making physical fault insertion of little use in the CPU area. We intend to use VOTE (Verification Of Test Effectiveness), a state-of-the-art software fault simulation system being developed by Diagnostic Support Engineering in Maynard, to verify test coverage and resolution when the simulator software is available. In addition, we will work to maintain the tightest possible feedback from our customers to assure that problems are solved and corrections implemented in the minimum possible time.

1.5. Risks

There are several factors that are key to the success of the Dolphin Diagnostic effort. They are (in no particular order):

a. The IIL diagnostic logic on the MCA chips. If this logic is too slow or too difficult to use, the execution times of the high-resolution diagnostics will become unacceptable (in terms of the Manufacturing QV times and Field Service MTTD goals).

b. The F-11 console. In deciding to do our own console design without Corporate bus, we have made the successful operation of the console a prerequisite to any further system debug. Should the console not work or come up late, the whole project must wait.

c. The Mercury subsystem. We are placing a lot of faith in the success of the Tewksbury design - both hardware and diagnostic software. Whereas we are doing a UBA as "backup", realistic schedules do not permit us to go very far down the two pronged path.

d. The DIAG JSYS. We are counting heavily on the Monitor people to give us a Dolphin version of this call. Without this "hook", the USER mode peripheral diagnostics will not work properly.

e. VOTE. We are planning on using the Diagnostic fault simulator to build "known good machine" data, as well as verifying test coverage. If this project slips much more, it may not be available in time for Dolphin.

f. GR test. There are no plans to do any GR testing of modules for Dolphin. Manufacturing has pointed out that, should the high-resolution diagnostics fall behind schedule or fail to meet their isolation goals, there would be over a year necessary to start up a backup GR strategy. Diagnostics feels that this is an extremely low risk, as the "goodness" of the diagnostics depends solely on the new MCA technology and not - as was the case for KL - on programmer talent.

1.6. Scope of the Project and of this Document

The Dolphin Diagnostic effort began over a year ago and will continue well into the 1980's. This document focuses on the period extending from the present to about mid 1981 (all references to dates in this document are calendar not fiscal!)

2. Description of Diagnostic Product

The diagnostic product for Dolphin may be broken into three major areas: console subsystem software, CPU cluster diagnostics and peripheral subsystem diagnostics.

The console software resides in the front-end processor's memory. The portion of the code needed for dealing with an absolutely "dead" system will be cast in ROM and the balance (majority) will be in RAM. This software provides for bootstrapping the system, configuring memory, real-time display, bus monitoring, KLINIK remote diagnostic support, manufacturing APT system support, standardized diagnostic error reporting, system power control, and support for the diagnostics which execute in the front-end processor.

The CPU cluster diagnostics detect and isolate faults in the CPU itself, the console processor and the various bus interfaces associated with the CPU. They consist of the high-resolution front-end resident diagnostics which make use of the IIL logic designed into the Macro Cell Arrays and a stored data base of "known good" machine states to diagnose to a faulty MCA chip, the 11-based console diagnostics, the functional tests which verify overall CPU and bus adapter operation, and exerciser programs which attempt to provoke bus conflicts to resolve system interaction problems.

The peripheral subsystem diagnostics detect and isolate faults in parts of the bus adapters, the peripheral controllers, and the peripheral devices themselves. As in the past, they will have two modes of operation: "reliability" and "device". Reliability mode is aimed at exercising the subsystem thoroughly and detecting any failures; device mode provides higher resolution with trouble-shooting aides such as "scope loops" and operator-controlled special testing. To the maximum extent possible within hardware and monitor software constraints, these diagnostics will run as USER mode programs under timesharing.

3. Goals and Non-goals

3.1. Functionality Goals

3.1.1. MCA Support

To assist Hardware Engineering in verifying MCA designs by providing functional test patterns for use with the SAGE chip simulations.

3.1.2. Console Processor Bootstraps

To supply console processor bootstraps which allow the console to be loaded from the following media:

a. TU-58
b. System disk
c. Magtape
d. Communications line

3.1.3. Console Software

To supply console software to do the following:

a. Control sequencing of system power
b. Monitor the hardware environment and protect the system from cooling or power supply malfunctions
c. Bootstrap the system from a "cold" start, including loading necessary microcodes, configuring MCA's and busses, configuring memory, and loading and starting the appropriate monitor bootstrap
d. Monitor the Dolphin bus
e. Provide realtime displays (if required)
f. Support remote diagnosis via the KLINIK port
g. Support APT
h. Handle error reporting for all diagnostics
i. Serve as the common interface to the hardware for those diagnostics which run in the console
j. Support diagnostic and hardware development by providing debugging aides such as breakpointing, and program and machine state tracing
k. Provide for operator control and visibility of the hardware
l. Support use of the CTY as the operating system's controlling terminal during timesharing.

3.1.4. Console-based Diagnostics

To provide high-resolution diagnostics which execute in the PDP-11 console to check out the following:

a. The console subsystem with its associated RAM memory and Dolphin bus interface
b. The system clock control logic
c. The system bus and bus monitoring logic
d. The Diagnostic memory buffer
e. As much as possible of the MBA (that is, the portion of the Mass Bus Adapter visible from the 11 side)
f. As much as possible of the IBA (the ICCS port adapter)
g. As much as possible of the UBA. (Since current plans are to do a UBA only as a back-up to the ICCS-based Communications and Unit Record subsystem, these diagnostics may be preliminary and not up to usual release standards.)
h. The Dolphin CPU
i. The Dolphin memory.

3.1.5. Dolphin-based CPU Diagnostics.

To provide functional tests which execute in the Dolphin CPU to verify correct operation of the following:

a. The Dolphin CPU

(This effort includes conversion - where necessary - of the famous "DF" series, conversion of any relavent "KL instruction" tests, and the creation of a new test to verify the new COBOL-specific instructions being implemented for Dolphin.)

b. Dolphin memory
c. MBA
d. IBA
e. UBA (See note under point g of 3.1.4).

3.1.6. Dolphin-based Peripheral Diagnostics.

To provide reliability and device diagnostics which execute in the Dolphin CPU in both EXEC and USER mode to check out the following devices:

a. RP06
b. RP07
c. RP07+
d. TM03/TU77
e. TM78/TU78
f. DX20/TU70
g. HSC50
h. The Mercury communications and unit record subsystem. (It is anticipated that much of this work will consist of adapting tests written for Mercury VAX in Tewksbury.)

3.1.7. System Exerciser

To supply a system exerciser package to test all hardware essential to basic monitor operation in a fashion which provides a high confidence of system operability.

3.1.8. Quick-verify Support

To provide a capability whereby board-level diagnostics may be easily supported as a subset of the system-level tests in order to provide excellent fault coverage for Manufacturing quick-verify in the minimum time.

3.2. Performance Goals

3.2.1. Fault Detection

To provide hardware diagnostic programs to detect in excess of 98% of solid hardware faults in the following logic:

- a. Console subsystem
- b. System clock control
- c. System bus and bus monitoring logic
- d. Diagnostic memory buffer
- e. Bus adapters
- f. Dolphin CPU
- g. Dolphin memory

To provide peripheral device diagnostics to detect the highest percentage of solid hardware faults permitted by the vendor hardware. (Current practice is in the 80% range).

3.2.2. Fault Resolution

The overall goal is to isolate any solid system fault to the field replaceable unit for better than 90% of all detectable faults.

In the CPU where MCA's are used, the goal is to isolate to a single MCA for better than 95% of all detectable faults.

For other lower density logic, the goal is to isolate to the board level.

3.2.3. Runtimes

It is an overall goal to minimize the time required to diagnose and resolve a fault anywhere in the Dolphin System. This goal will be met by following good diagnostic practice with regard to test structuring and partitioning, efficient use of hardware visibility, and store data base.

3.3. Non-goals:

3.3.1. 32-bit Support

The Dolphin Diagnostic Project does not include any funding for support of a 32-bit Dolphin. This is not to say that anything will intentionally be done to preclude such work being done at a future time (in fact, a reasonable effort is being made to be VAX compatible where possible). It does mean that additional funding and planning will be required should support of a VAX Dolphin become a goal.

3.3.2. Additional Console Peripherals

The only peripherals supported on the console processor will be the video terminal, TU-58, hard copy terminal, and APT/KLINIK ports. The console software will not be required to have any knowledge of the operating system's file structure beyond the ability to start the basic boot code.

3.3.3. MCA Incoming Inspection Tests

No integrated circuit inspection tests are included in this project - other than the design verification patterns of 3.1.1.

3.3.4. CPU Board Tests

No CPU board tests other than those provided for QV are planned. It is assumed that boards composed of macro-cell arrays will require no testing other than the appropriate high-resolution diagnostics, since chip isolation will be provided.

3.3.5. Systems with Bus Repeaters

There are no plans to support "expanded" systems which do not have a complete console subsystem - including TU-58 and console terminal - on each segment of the bus.

4. Operating Environment

4.1. Hardware

The diagnostics will run with any of the system configurations described in the Dolphin Implementation Plan (issued 5 January 1979). Since many alternate paths for loading diagnostics and for booting the system are provided, it is unlikely that the diagnostics would limit other possible system configurations (e.g. so-called "mini" Dolphins). The following components are required in any supported system:

a. Complete Console Subsystem including:
   F-11
   80K of Console Memory
   Console Terminal
   TU-58
   Dolphin Bus Monitoring Logic
   Diagnostic Memory
b. Dolphin CPU
c. 128K of Dolphin Memory
d. System Disk
e. System Magtape (standard diagnostic distribution medium)

4.2. Software

The USER mode peripheral diagnostics will run under both the TOPS-10 and TOPS-20 Operating Systems done for Dolphin, provided only that the equivalent functionality of the "DIAG JSYS" is implemented in both monitors.

The diagnostic package for the Dolphin project will include stand-alone monitors for tape and disk and all necessary utilities for building, loading, running, and maintaining the diagnostics.

5. Interdependencies With Other Projects

5.1. Hardware

The Dolphin diagnostic project is critically dependent on the RAMP hardware features being designed into the system. Diagnostic visibility and execution speed, for example, is almost entirely a function of the - currently unspecified (by Motorola) - IIL logic allocated to each MCA. Another key hardware area is the console and its associated subsystem (including the bus monitor memory). There is also a strong dependency on the "MAD-30" memory being designed for the breadboard system. It must function correctly to be able to load and run the functional tests needed to even start CPU testing.

The schedules and manpower loading given in this plan are based on the master schedule published in the Dolphin System Proposal and Implementation Plan of 5 January 1979. Before a diagnostic can be designed, a certain amount of the hardware design must be complete. Before a diagnostic can be debugged, an at least "semi-working" hardware breadboard must exist. Therefore, a lag in any phase of the hardware development will almost surely invalidate the estimates contained in this plan.

There are obviously dependencies the other way around. If the diagnostics are not ready on time, getting the breadboard checked out will be extremely difficult.

5.2. Software

The Dolphin USER mode diagnostics are dependent on the monitor group's implementation of the DIAG JSYS functionality. Also, there must be agreement on such items as error handling protocol, KLINIK support, CTY protocol, and "real-time" displays (if implemented).

The manpower estimates in this plan for the Mercury (Hydra) communications subsystem assume that F-11 based diagnostics for the F-11 itself, its links, line units and, possibly, the card reader and line printer unit record peripherals will be done by Tewskbury. We expect to be able to use them with only minor modification and the addition of a program to load the F-11 "down-line" from the Dolphin bus through the ICCS port. Also, in the communication area, we expect Hardware Engineering to make the "go no-go" decision on Mercury in time to allow us to hold schedule for first customer ship. There clearly is not enough manpower to go completely down both the UBA and IBA paths.

6. Compatibility and Standards

Diagnostic programs developed for Dolphin shall conform to the standards currently in force for KL10 and KS10 diagnostics. A DECDOC listing with a full cross-reference will be supplied on microfiche for all programs developed. Command parsing shall be consistent for all levels of command decoding. Standards developed for PDP11 diagnostics shall not apply to the console-based programs, as adherence to good programming practices is essential.

7. Product Review Method

The primary review body for this plan shall be the Dolphin Steering Committee. In addition, copies of this plan and the functional specifications which define the diagnostic product for Dolphin will be circulated for comment to those whose names appear on the front page. An effort will be made to circulate the diagnostic plans and functional specifications to all those on the project who have an interest in the product. The component functional specifications will be reviewed at Dolphin Committee meetings as appropriate.

8. Component Plans

This plan presents an overview of the diagnostic product for Dolphin. The details of each component will be found in the functional specification for the component. As functional specifications are written, they will be circulated to the people whose names appear on the front page of this document.

9. Development Staffing and Funding

9.1. Personae

Large Systems Diagnostics Cost Center Manager:	Dick Maliska
Dolphin Diagnostic Project Manager:	Dale Cook
Dolphin Diagnostic Project Leader:	Jim Jones
Dolphin Console Firmware:	Bob Petty
Dolphin Console Diagnostics:	Steve Beeman
Dolphin MCA Test Patterns:	Dave Tibbetts
Dolphin Simulation / Quality Assurance:	Ted Elkind
Dolphin Memory Diagnostics:	Dave Pease
Dolphin MBA Diagnostics:	Jim Jones
Dolphin UBA Diagnostics:	Tim Gawne
Dolphin Mercury Subsystem Diagnostics:	Tim Gawne

9.2. Manpower Loading

Task	Manpower	Begin	End
MCA Test Patterns	1	started	Dec 79
Basic Console	1	started	Dec 79
	1/2	Jan 80	Mar 80
Console Runtime Package	1/2	Jan 80	Dec 80
Console Utilities	1/2	started	Jun 79
Console Diagnostics	1	started	Mar 80
CPU Functional Tests	1	Apr 79	Mar 80
CPU Diagnostics	1	Jul 79	Dec 79
	2	Jan 80	Mar 80
	4	Apr 80	Dec 81
	2	Jan 82	Jun 82
Memory Configurator and Diagnostic	1/2	started	Dec 79
	1	Jan 80	Jun 80
	1 1/2	Jul 80	Jun 81
	1	Jul 81	Dec 81
MBA - Console-based Diagnostic	1/2	started	Sep 79
MBA - Dolphin-based	1	Oct 79	Jun 80
UBA - Console-based	1/2	started	Sep 79
UBA - Dolphin-based	1	Oct 79	Jun 80
IBA - Console-based	1/2	started	Dec 79
IBA - Dolphin-based	1	Jul 80	Mar 81
RP06 Basic + Reliability	1	Apr 80	Dec 80
TU77 Basic + Reliability	1	Apr 80	Sep 80
RP07 Basic + Reliability	1	Jul 80	Mar 81
TU78 Basic + Reliability	1	Jul 80	Mar 81
HSC50 Diagnostics	1	Jan 81	Mar 81
	2	Apr 81	Sep 81
Systems Exerciser	1	Jul 80	Dec 81
Q/A (Vote)	1	Jul 79	Dec 79
	1	Jan 81	Jun 81

9.3. Funding Required

The following is derived from the above manpower loading and the assumption that labor costs $12,000 per person quarter (all figures in thousands of dollars):

Quarter Beginning	Total Manpower	Total Funding
Jan 79	5 1/2	66
Apr 79	6 1/2	78
Jul 79	8	96
Oct 79	9	108
Jan 80	8	96
Apr 80	9 1/2	114
Jul 80	12	144
Oct 80	11	132
Jan 81	11 1/2	138
Apr 81	9 1/2	114
Jul 81	8	96
Oct 81	6	72
totals ======>	104 1/2	1254

9.4. Hardware Requirements

The work described in this plan is totally dependent on hardware availability. It is assumed that the Diagnostic Timesharing System, 1029, will be maintained at its present level, or expanded if the load warrants doing so. It is also assumed that access to the Dolphin breadboard and prototypes will be allowed during all phases of the project. Since it is not possible to forecast the exact usage of Dolphin hardware at this time, Large Systems Diagnostics must assume that system time will be allocated by Hardware Engineering in an equitable and timely manner.

10. Implementation Plan

The sequence and rate of work on this project is largely gated by hardware design, specification and availability. The first efforts involve the MCA chip support, console firmware and utilities, and the various port adapters. As the CPU design becomes firm, work can proceed on the functional tests and, somewhat later, on the high resolution CPU diagnostics. Memory work can begin as that hardware design takes shape, and finally the peripheral and system exerciser work can start. Completion dates are gated by hardware power on dates and, ultimately, first customer ship date. A summary of the key schedule milestones of the project follows:

(Note: Functional specifications for a given component should be available approximately one month after the "begin" date given below.)

date	task
present	MCA support Begin console firmware, utilities and diagnostics Begin memory diagnostics Begin MBA, UBA and IBA console-based work
April 1979	Begin CPU functional tests
July 1979	Begin high-resolution CPU tests Begin VOTE simulator work
August 1979	Console breadboard available to start debug of preliminary console firmware & diagnostics
October 1979	Begin 10-side MBA and UBA diagnostics
November 1979	Breadboard available to start debug of preliminary functional tests
January 1980	MCA pattern support complete Console-based bus adapter tests complete
April 1980	Begin RP06 and TU77 diagnostics Basic console firmware and console utilities complete Console diagnostics complete CPU functional tests complete
July 1980	Begin RP07, TU78 and 10-side IBA diagnostics Begin systems exerciser 10-side MBA and UBA diagnostics complete
October 1980	TU77 diagnostics complete
January 1981	Begin HSC50 diagnostics RP06 diagnostics complete
April 1981	RP07, TU78 and 10-side UBA diagnostics complete
June 1981	Preliminary diagnostic package available to support first customer ship

11. Documentation Staffing and Work Plan

Documentation relating to the Dolphin Diagnostic Product will be covered as a component of the Dolphin Documentation Plan to be issued by Technical Documentation.

12. Quality Assurance Plan

The primary means of verifying diagnostic quality will be the VOTE simulator which allows for the simulated execution of a diagnostic against both fault-free and faulty hardware models. There is a risk that the VOTE project will fall further behind and not be available in the Dolphin timeframe.

13. Distribution Plan

All diagnostic software shall be distributed on magnetic tape. Diagnostic engineering shall supply software to transfer the tape to disk. All updates will be supplied on magnetic tape for local transfer to disk. The backup medium is expected to be in the form of downline loads from the DDC. A limited number of diagnostics will be supplied for use with a local load device. All listings shall be supplied on microfiche.

14. Training Plan

Training of Field Service and customer personnel to use the Dolphin Diagnostic Package will be covered in the Product Support Plan.

15. Maintenance Plan

Dolphin diagnostic maintenance will require 2 full-time employees for at least one year after first customer ship. In addition, some time from the developers who designed the programs will be required to cover Engineering Change Orders and AIDES reported bugs.