E82-10015

CR-161039

MU-I1-00300
JSC-16918

MAR 2 6 1981

AgRISTARS

Foreign Commodity Production Forecasting and Supporting Research

A Joint Program for Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing

September 1980

ERSYS—SPP ACCESS METHOD SUBSYSTEM DESIGN SPECIFICATION

NASA CR-161039

R. C. Weise

(E82-10015) ERSYS-SPP ACCESS METHOD SUBSYSTEM DESIGN SPECIFICATION (IBM Federal Systems Div.) 74 p HC A04/MF A01 CSCL 02C

N82-15493

Unclas
G3/43 00015

International Business Machine Corporation
Federal Systems Division
1322 Space Park Drive
Houston, Texas 77058

Lyndon B. Johnson Space Center
Houston, Texas 77058

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
7. JSC-16918; MU-I7-00300		

4. Title and Subtitle	5. Report Date
AGRISTARS: Foreign Commodity Production Forecasting + Supporting Research, 2. ERSYS - SPP Access Method Subsystem Design Specification	1. September 26, 1980
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
3. R. C. Weise	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
4. International Business Machines Corporation Federal Systems Division 1322 Space Park Drive Houston, Texas 77058	5. NAS9-14350
	13. Type of Report and Period Covered
	Task Description

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 Contract Monitor: ~~B. Hay SG2~~ ROY EASON	

15. Supplementary Notes

16. Abstract

The STARAN Special Purpose Processor (SPP) is a machine allowing the same operation to be performed on up to 512 different data elements simultaneously. In the ERSYS system, it will be attached to a 4341 Plug Compatible Machine (PCM) to do certain existing algorithms and, at a later date, to perform other to be specified algorithms.

That part of the interface between the 4341 PCM and the SPP located in the 4341 PCM will be known as the SPP Access Method (SPPAM). Access to the SPPAM will be obtained by use of the NQUEUE and DQUEUE commands. The Subsystem Design Specification is to incorporate all applicable design considerations from the ERSYS System Design Specification and the Level B Requirements documents relating to the SPPAM. It is intended as a basis for the Preliminary Design Review (PDR) and will expand into the Subsystem Detailed Design Specification.

17. Key Words (Suggested by Author(s))	18. Distribution Statement

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	71	

*For sale by the National Technical Information Service, Springfield, Virginia 22161

Contract Number NAS9-14350
JSC No. 16918
DRL Line Item No. 15

APPROVAL SHEET

B. B. Duprey
ERSYS Application Systems

E. W. Poole
Earth Resources Systems

L. S. Wright
Support Software

S. A. Hudson
Ground Based Space Systems

A. A. Aldrich — 10/17/80
Quality Assurance

NASA APPROVAL RECEIVED
ON TIRF 22118 11/22/80

ERSYS

Special Purpose Processor (SPP)

Access Method Subsystem

Design Specification

HR4

9/26/80

TABLE OF CONTENTS

1.0 INTRODUCTION

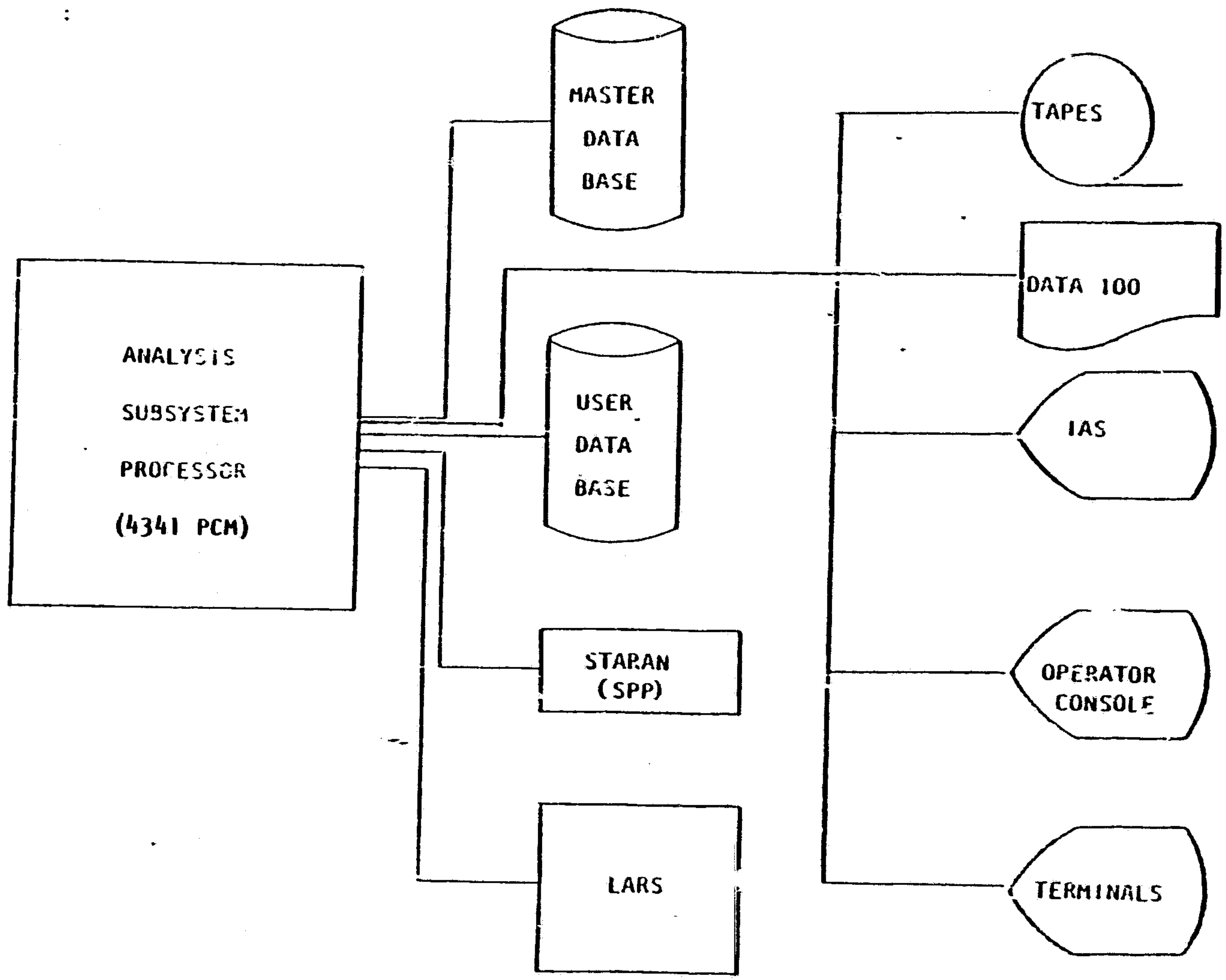
1.1 DOCUMENT PURPOSE AND SCOPE

The STARAN Special Purpose Processor (SPP) is a machine allowing the same operation to be performed on up to 512 different data elements simultaneously. In the ERSYS system, it will be attached to a 4341 Plug Compatible Machine (PCM) to do certain existing algorithms and, at a later date, to perform other to be specified algorithms. Figure 1.1-1 shows how the SPP will fit into the ERSYS hardware environment.

That part of the interface between the 4341 PCM and the SPP located in the 4341 PCM will be known as the SPP Access Method (SPPAM). Access to the SPPAM will be obtained by use of the NQUEUE and DQUEUE commands. The Subsystem Design Specification is to incorporate all applicable design considerations from the ERSYS System Design Specification and the Level B Requirements documents relating to the SPPAM. It is intended as a basis for the Preliminary Design Review (PDR) and will expand into the Subsystem Detailed Design Specification.

1.2 ORGANIZATION

Section 2 presents an overview of the design, along with the performance characteristics, and a description of situations causing abnormal task termination. Section 3 gives the module specifications for the SPPAM functions, with an explanation of the NQUEUE and DQUEUE functions given in Appendix A. Section 4 presents the module specification for a status command called SPPSTAT. Section 5 treats the topics of testing tools and schedules. A glossary is given in Appendix B.



Configuration Showing Place of SPP in EODLS

Figure 1.1-1

2.0 SUBSYSTEM DESIGN

This section contains the high level design and design considerations for the ERSYS SPPAM. It includes an overview of the subsystem design, assumptions, constraints, design objectives, major design decisions, and a description of the subsystem architecture and major functions.

2.1 DESIGN OVERVIEW

The ERSYS SPPAM Subsystem consists of 3 functions:

SPPRES
SPPRDWRT
SPPREL

The SPPRES function requests the use of the SPP via the NQUEUE command and initializes the task once the user machine has the use of the SPP. The reader is directed to Appendix A for a description of the NQUEUE command. When given access to the SPP, a time-out clock is initialized and the Reserve command and Interface Control Record sent to the SPP.

The SPPRDWRT function has the capability to read results from the SPP and write data to be processed, including the constants record, to the SPP and reset the clock whenever data is sent to the SPP.

The SPPREL function issues the Release command to the SPP and detachs the SPP from the user machine by use of the DQUEUE command (Appendix A). In addition, this command turns off the time-out clock.

In the event of an uncorrectable I/O error during the execution of any one of these functions, each function has the capability to end the application and detach the SPP by use of the DQUEUE function. The user has the ability to cancel. An example of how these functions could be used is given in Example 2-1: the two functions FRMTBUF and FRMTICR are non-SPPAM application dependent formatting functions. In addition there will be a command, SPPSTAT, allowing the user, at any time while under ERSYS, to receive a message giving an upper limit on the time necessary to wait before obtaining the SPP.

2.2 ASSUMPTIONS AND CONSTRAINTS

1. Optimize for LACIE segment size; allow capability for other sizes.

2. The SPP processes one user at a time.

3. The SPPAM conforms to the present specifications given in the Interface Control Document NASA/STARAN SPP (GER 16224).

SPP APPLICATION

** RESERVE SPP AND SEND ICR.

FRMTICR
SPPRES

** PROCESS DATA UNTIL ALL RESULTS HAVE COME BACK FROM THE SPP.

WHILE

NOT ALL RESULTS BACK FROM THE SPP = TRUE

DO

FRMTBUF
SPPRDWRT

OD

SPPREL

EXAMPLE 2-1

2.3 DESIGN OBJECTIVES

1. The SPPAM should be independent of the particular application executed on the SPP.

2. The SPP Access Method Design should make efficient use of the SPP.

3. User interface with the SPPAM should be simple and permit the authorized user to execute already written macros or extract the SPPAM functions for use in his or her own application.

4. Error messages will be sufficient to tell the terminal user the problem and, where appropriate, recommend a solution.

5. The design should support the existing SPP applications without modification of SPP software. However, it must be general enough to accomodate new approaches to programming the SPP.

2.4 MAJOR DESIGN DECISIONS

1. Any user machine wishing to use the SPP will do so through a resource allocator residing in a separate virtual machine. This resource allocator will attach the SPP to the given user machine for processing of an application by that user machine and detach it at the end of the SPP application. A description of this resource allocator is given in Appendix A.

2. Use the DIAG x '20' instruction instead of the EXCP instruction to execute CCW programs performing SPP reads and writes. The DIAG x '20' requires less coding of tables and has sufficient return codes to do error handling.

3. Requests to reserve and release the SPP will be retried once; all other requests will be tried once.

2.5 ARCHITECTURE

2.5.1 Environment to be Supported

All user machines communicating with the SPP under ERSYS will operate in a VM/CMS environment. Figure 2.5.1-1 shows the environment in the ERSYS system.

2.5.2 Abnormal Task Termination

These situations will cause a task to be abnormally terminated:

1. A given user machine which has access to the SPP has not sent data to the SPP within (TBD) time.

2. The SPP has not responded to commands issued by the SPPAM within (TBD) time.

ANALYSIS SUBSYSTEM CONFIGURATION SUMMARY

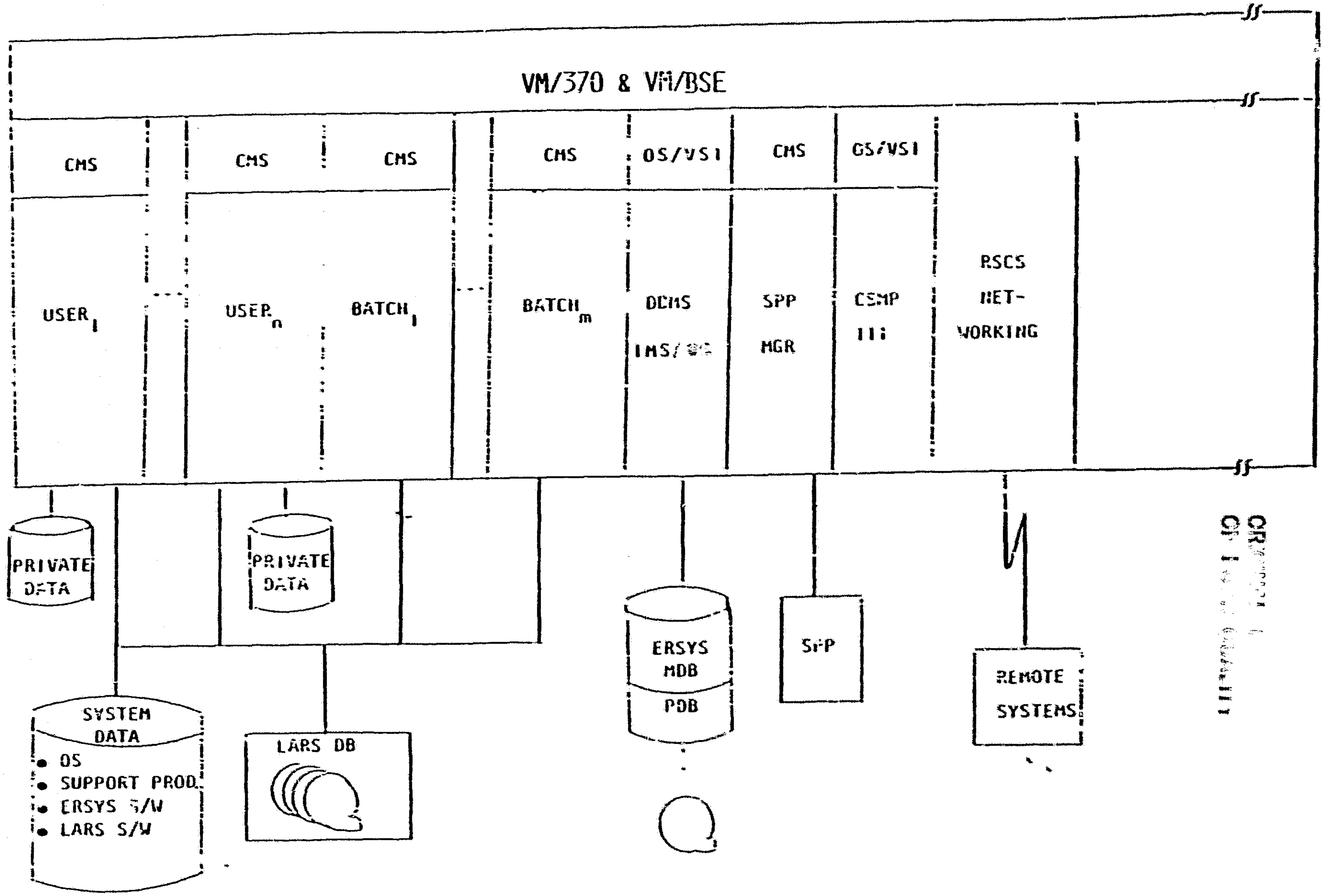


Figure 2.5.1-1

3. The user machine receives a program check or other type of interrupt which means that the user machine will not be able to finish processing of the task.

4. A data transfer to the SPP has resulted in an uncorrectable I/O error.

In the first three cases, there are programs provided within the ERSYS User Interface which will terminate the task and detach the SPP. In the last case, similar capabilities within each SPPAM function are invoked, using the DQUEUE function.

2.5.3 Performance Characteristics of Architecture

The functions of the SPPAM are reentrant. The option to either have one copy which is shared by all user machines or have a copy in each user machine will be a system build option. An application using the SPPAM can be coded in either assembler language or FORTRAN.

2.6 SPP/DATA BASE INTERFACE

All interfacing with the data base is done by non-SPPAM application functions.

2.7 REQUIREMENTS TRACEABILITY

Tables 2.7-1 and 2.7-2 show the requirements resulting from the Level B Requirements and the System Design Specification. The two tables together give the requirements on the subsystem. The appropriate documents will be amended so that all requirements will be cross-referenced.

	SPPRES	SPPRDWRT	SPPREL	SPPSTAT
All users dynamically share SPP.	X		X	
Support ERSYS configurations.	X	X	X	X
There is a simple interface to the SPP Management Facility.	X	X	X	X
Load into ERSYS supervisor area of user's space by initialization function.				
Perform error recovery.	X	X	X	
Have access to VMCF.	X	X	X	X

CORRELATION MATRIX FOR SPP INTERFACE REQUIREMENTS

Table 2.7-1

	SPPRES	SPPRDWRT	SPPREL	SPPSTAT
Run in a separate CMS virtual machine.				
Attach SPP Hardware to SPP Manager only.				
Priority queueing possible.				
Process data.	X	X	X	
Handle all communication with SPP Hardware.	X	X	X	
User ERSYS Supervisor to access input data from system or user data bases.				
Format input data for transmission to and use by the SPP Software.				
Provide interface functions for interfacing with the SPP.	X	X	X	X
Send results to user machine.	X	X	X	
Queue requests to use the SPP on a first in, first out basis.	X	X	X	

CORRELATION MATRIX FOR SPP MANAGEMENT FACILITY REQUIREMENTS

Table 2.7-2

3.0 SPPAM

The SPPAM module is presented in this section. The functions of this module cannot, in themselves, form an SPP application. They are intended to be interleaved with non-SPPAM functions to create an SPP application. The interleaving will be application dependent. Certain users will be able to access these functions to form their own application. An example of one such interleaving is given in Example 2-1.

3.1 SPPAM

3.1.1 Overview

There are three functions in this module: SPPRES, SPPRDWRT, and SPPREL. The functional capabilities of each are given in Section 2.1.

3.1.2 Interfaces

The input specified as time-out is an interrupt; it is generated by the SPP resource allocator. The inputs specified as abnormal end and command cancellation are generated when one of the conditions given in Section 2.5.2 arises. Inputs and outputs are given in Figure 3.1.5-1.

3.1.3 Required Resources

This module is written in assembler language. The local Houston version of the PDL processor is used. Other resources are specified in Section A.3.

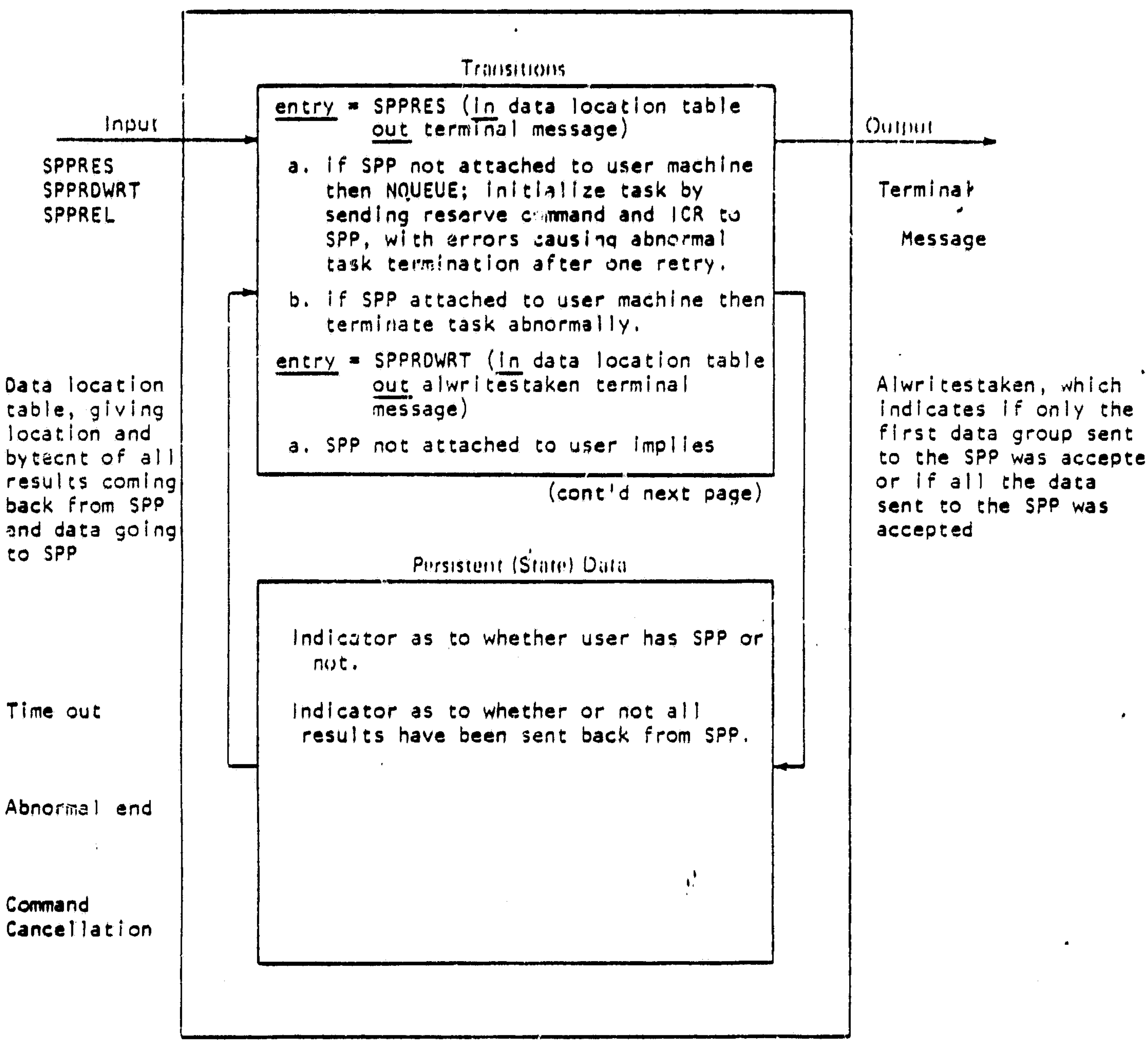
3.1.4 Performance Characteristics

See Section 2.5.3.

3.1.5 Description

The module specification is given in Figure 3.1.5-1 and continuation pages.

SPPAM



Initial State: Indicator as to whether user has SPP or not indicates that the user does not. Indicator as to whether or not all results have come back indicates they have not.

Figure 3.1.5-1

ORIGINAL PAGE IS
OF POOR QUALITY

SPPAM (Cont'd)

Transitions

Input

terminate task abnormally.

b. SPP attached to user machine and all results back from SPP implies terminate task abnormally.

c. SPP attached to user machine and not all results back from SPP implies build and issue CCW chain using data location table, setting indicator that all results have come back from the SPP when this happens; do error handling.

entry = SPPREL (out terminal message)

a. SPP not attached to user implies

(cont'd next page)

Output

Persistent (State) Data

Initial State:

Figure 3.1.5-1
(cont'd)

SPPAM (Cont'd)

Input

Transitions

terminate task abnormally.

b. SPP attached to user and all results not back from SPP implies terminate task abnormally.

c. SPP attached to user and all results back from SPP implies release SPP and detach from user machine, with errors after one retry, causing abnormal task termination.

<u>entry</u> = Command Cancellation (<u>out</u> terminal message)

a. SPP not attached to user implies end wait; terminate task abnormally.

(Cont'd next page)

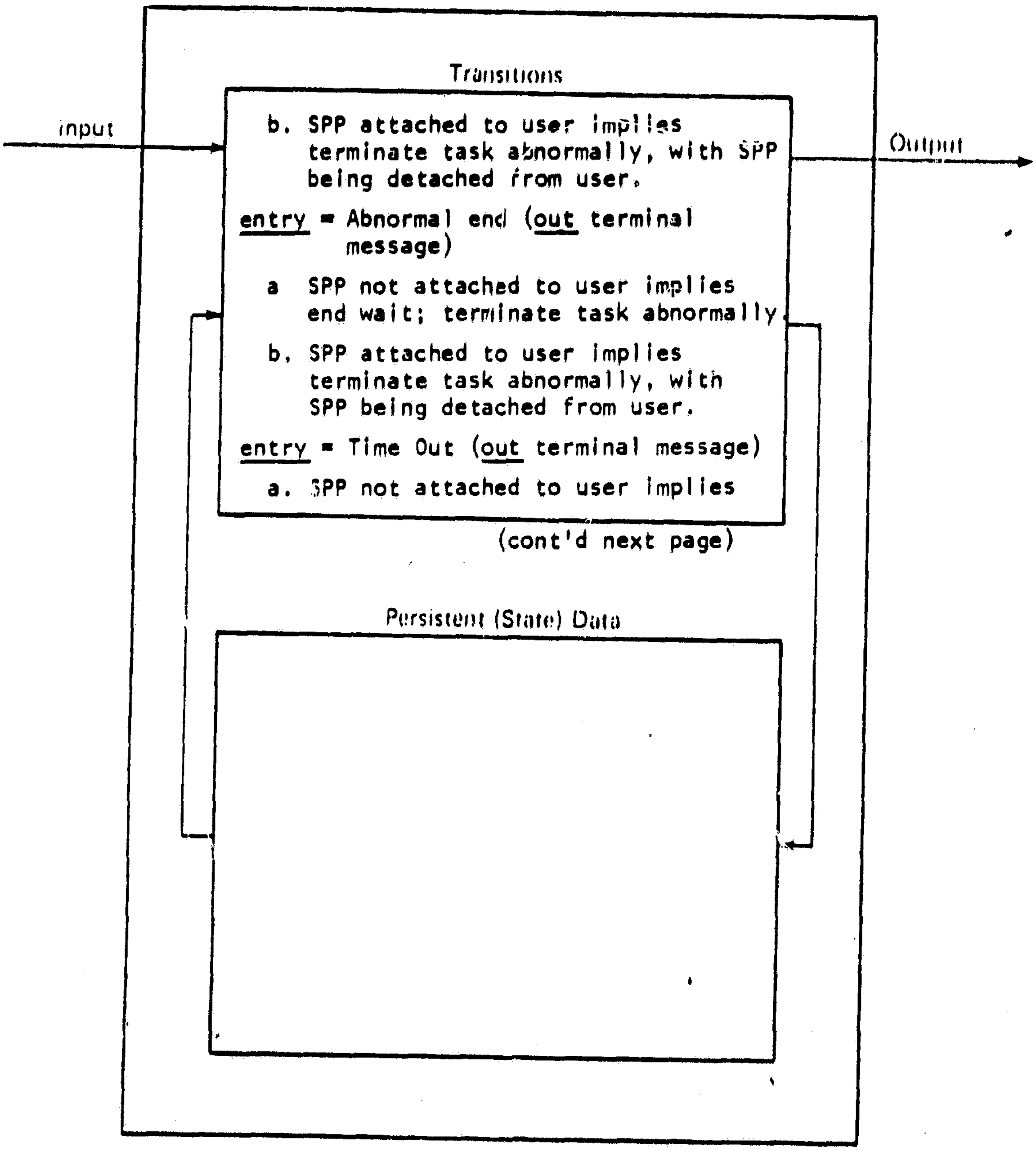
Output

Persistent (State) Data

Initial State:

Figure 3.1.5-1
(cont'd)

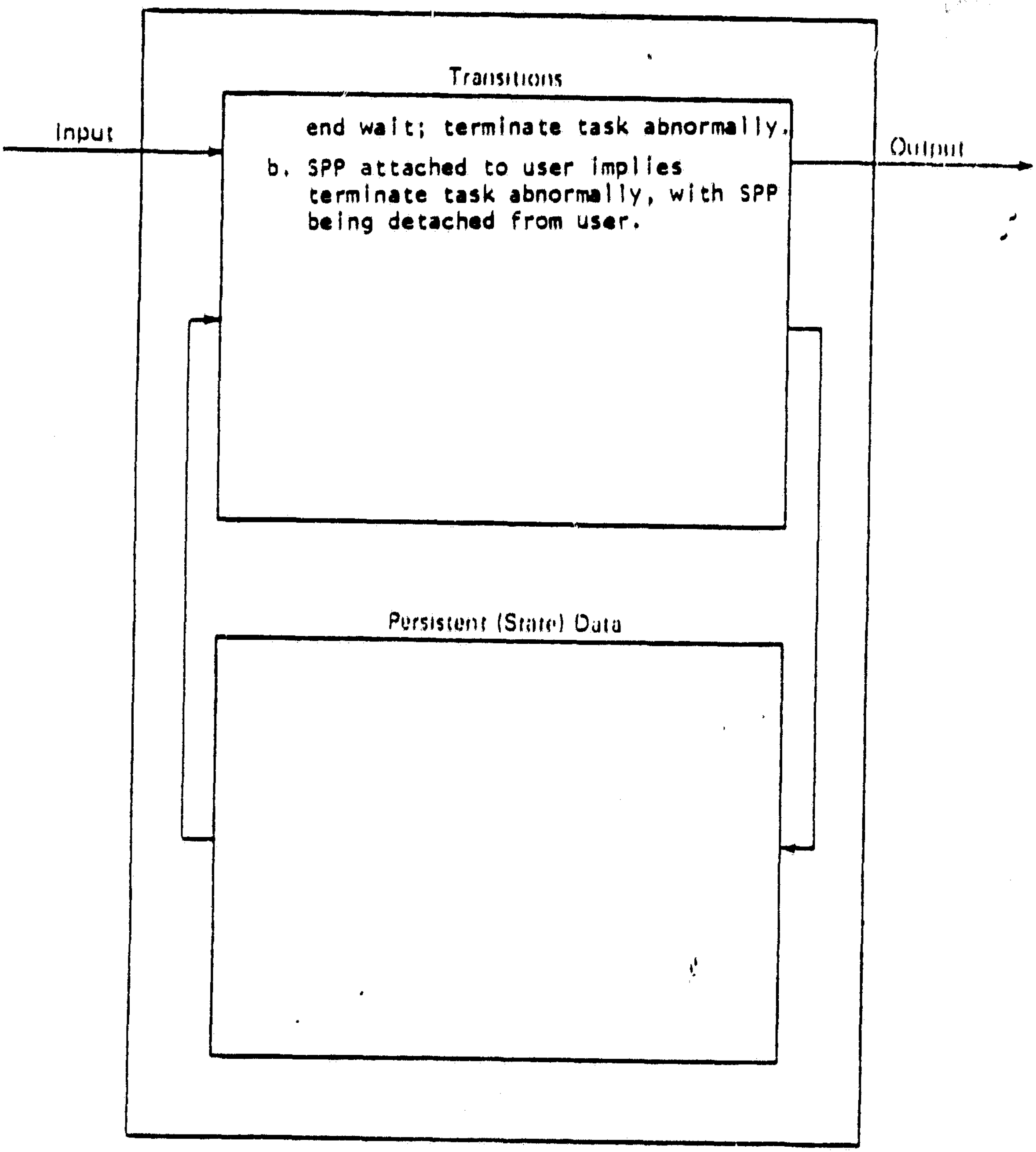
SPPAM (Cont'd)



Initial State:

Figure 3.1.5-1
(cont'd)

SPPAM (Cont'd)



Initial State:

Figure 3.1.5-1
(cont'd)

4.0 SPPSTAT

4.0.1 Overview

There is one routine in this module: SPPSTAT. This is a command used to receive an upper limit on the amount of time before the requests presently quoued up for the SPP will be processed.

4.0.2 Inputs/Outputs

Inputs and outputs are given in Figure 4.0.5-1.

4.0.3 Required Resources

This module is written in assembler language. The local Houston version of the PDL processor is used. Other resources are given in A.3.

4.0.4 Performance Characteristics

See Section 2.5.3.

4.0.5 Description

The module specification is given in Figure 4.0.5-1.

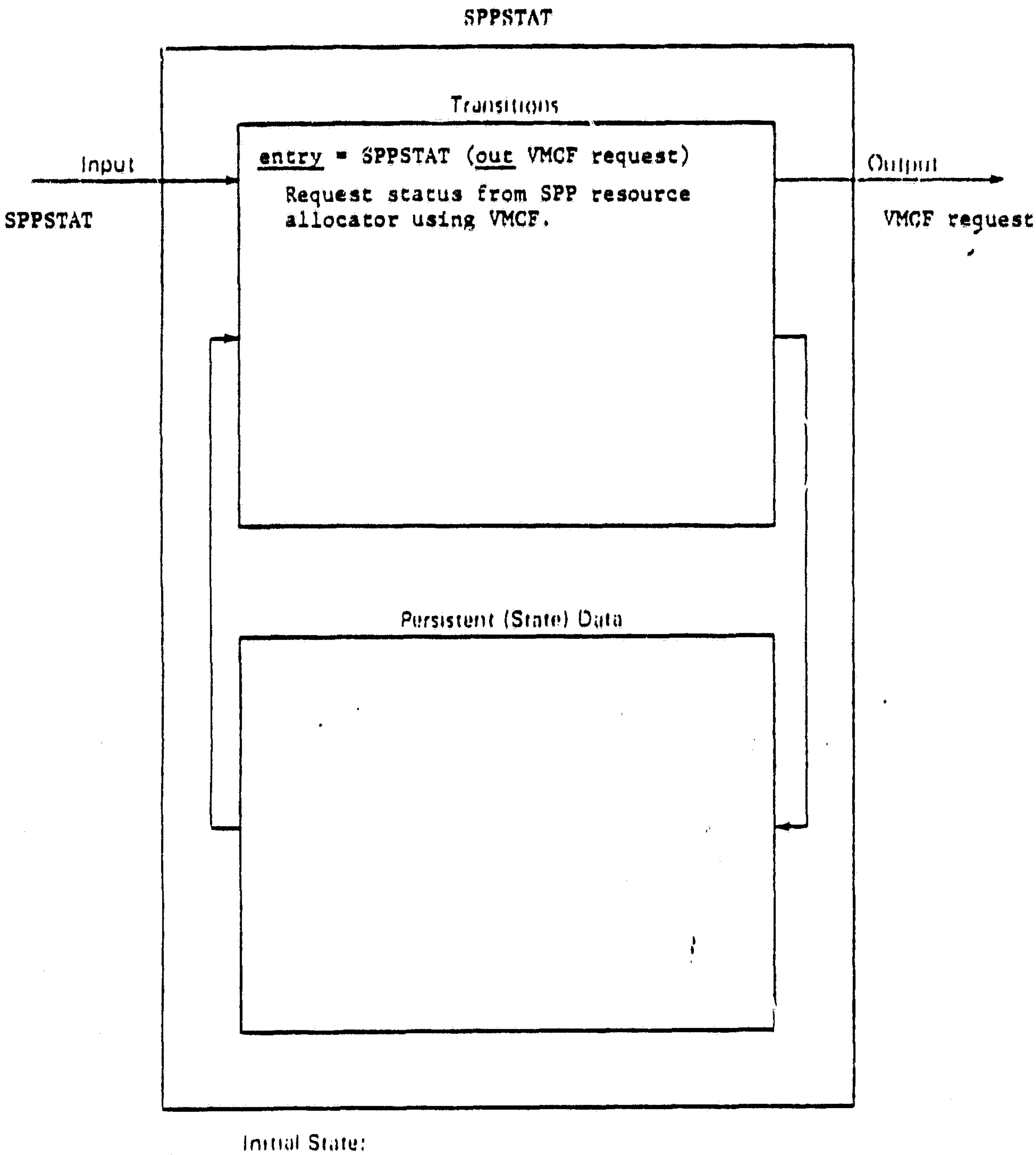

SPPSTAT
Transitions
Input
SPPSTAT
entry = SPPSTAT (out VMCF request)
Request status from SPP resource allocator using VMCF.
Output
VMCF request
Persistent (State) Data
Initial State:

Figure 4.0.5-1

5.0 SUBSYSTEM IMPLEMENTATION

This section describes how the subsystem will be implemented. The first subsection specifies implementation tools; the second gives a release schedule.

5.1 TOOLS

Control and debugging of the SPP Access Method will be done using standard CMS and CP functions. There will be a diagnostic function; it will reside in the machine specified as the resource allocator by the NQUEUE function and will be invoked at system initialization time.

5.2 PHASING PLANS AND RATIONALE

Basic release - SPP Manager function and SPP User Machine functions will work on the development testing test case.

Application releases - as each SPP application on the SPP is finished, a macro to run that application will be released.

Detailed schedules will be maintained in the ERSYS development plan.

5.3 DEVELOPMENT TESTING

There will be development testing of the interface. It will test all major components of the interface in a single and multiple user environment.

APPENDIX A

Description of NQUEUE and DQUEUE

A.1 Overview

The queue manager controls access to system-wide resources. For example, if virtual machine #1 requires the SPP and the SPP is represented by queue #1, virtual machine #1 would issue a NQUEUE specifying queue #1. The queue manager would service the request and ATTACH the SPP to that virtual machine. Future requests for the SPP would be enqueued until virtual machine #1 DQUEUES on the resource. That part of the queue handler which manages the SPP is called the SPP Manager.

The queue manager contains two functions: a NQUEUE function whereby a user virtual machine requests a system resource; and a DQUEUE function, whereby a user virtual machine relinquishes a system resource, making it available to other virtual machines. The queue manager never permits more than one virtual machine to have a resource.

Figure A-1 shows a simplified view of a NQUEUE request, a DQUEUE request follows a similar path except no ECBs are involved. In Figure A-1 the user program issues a NQUEUE for queue #1. As a result, control is passed to an interface routine which passes the request to the queue manager's virtual machine via VMCF. The user virtual machine then issues a WAIT for the ECB corresponding to queue #1. In this example, since no other virtual machine is using the resource, the queue manager, on receipt of the user's request, immediately gives the resource to the requesting virtual machine via VMCF. As a result of the VMCF request, the user's External Interrupt Handler receives control and posts the ECB for queue #1. The interface returns to its caller and the user program continues execution.

The queue manager contains logic to release the SPP resource if the resource is held for an excessively long time by a virtual machine. Further, if the queue manager abends, the queue manager STAE routine will flush all queues and send abend VMCF messages to all queued virtual machines.

A.2 Interfaces

Inputs and outputs are given in Figures A.5-1 and A.5-2.

A.3 Required Resources

This module uses VMCF; it will be coded in assembler language.

A.4 Performance Characteristics

See Section 2.5.3.

A.5 Description

The module specification is given for the queue manager machine in Figure A.5-1 and, for the user's machine, in Figure A.5-2.

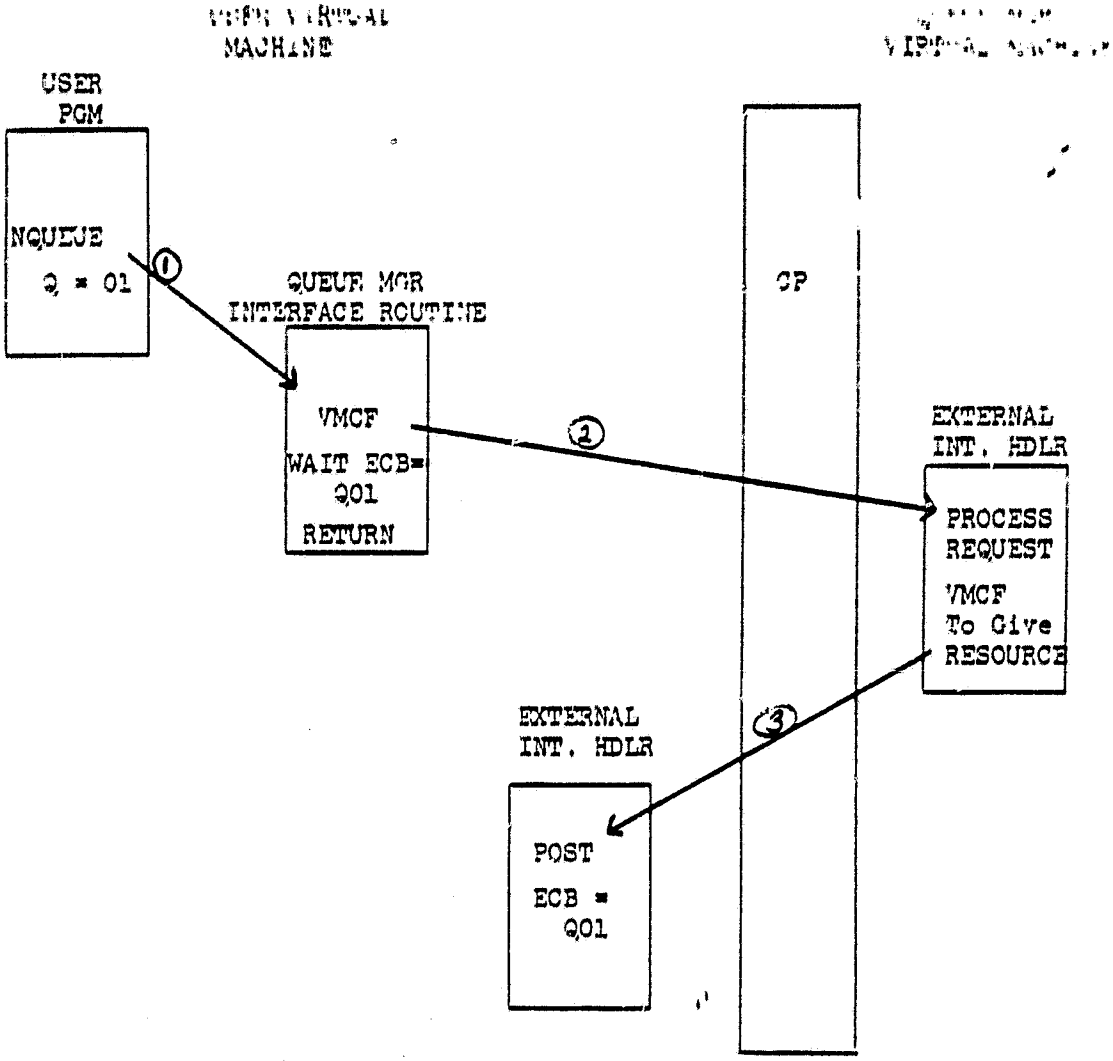
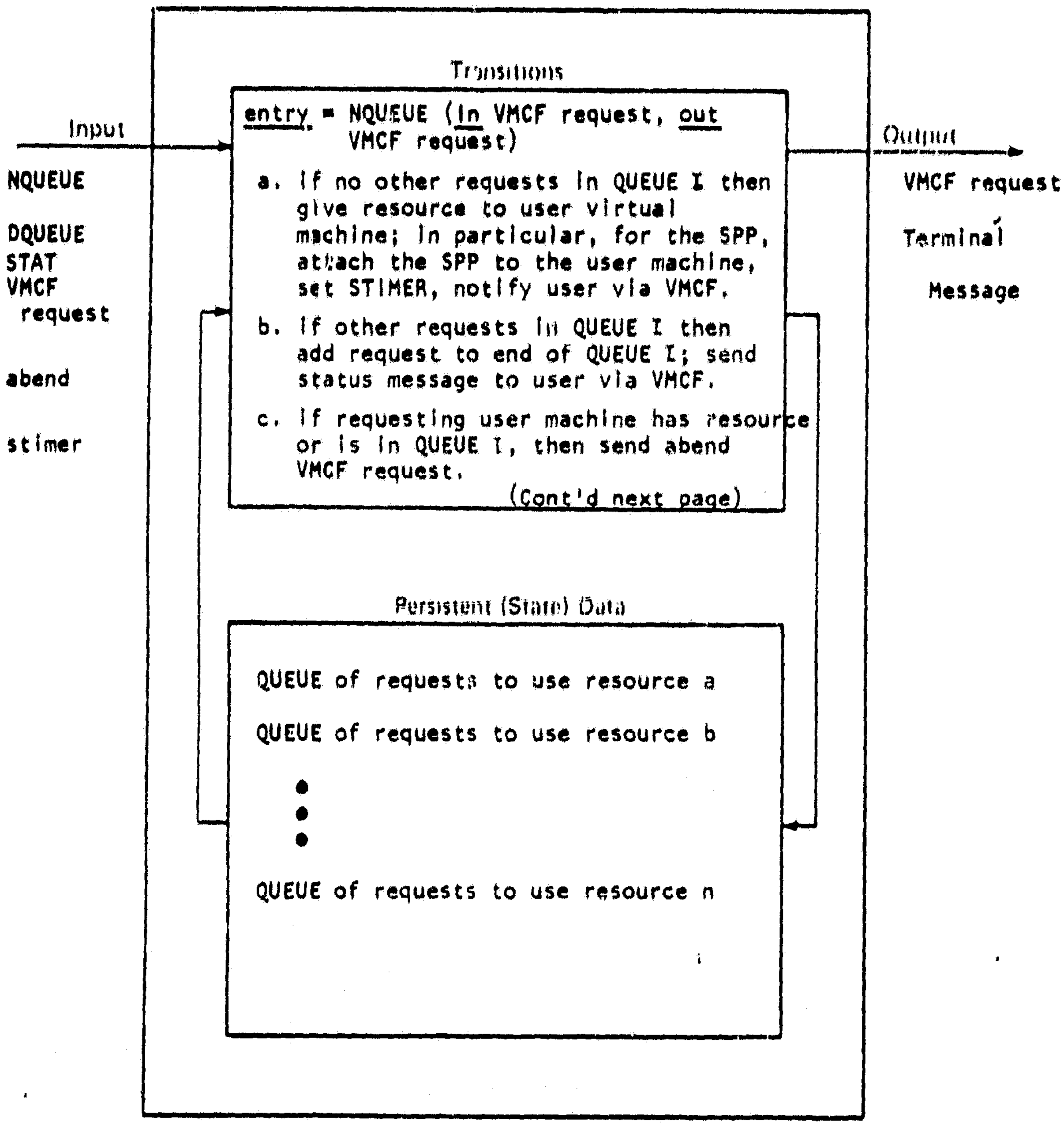


Figure A-1 ENQUEUE Request Processing (Simplified)

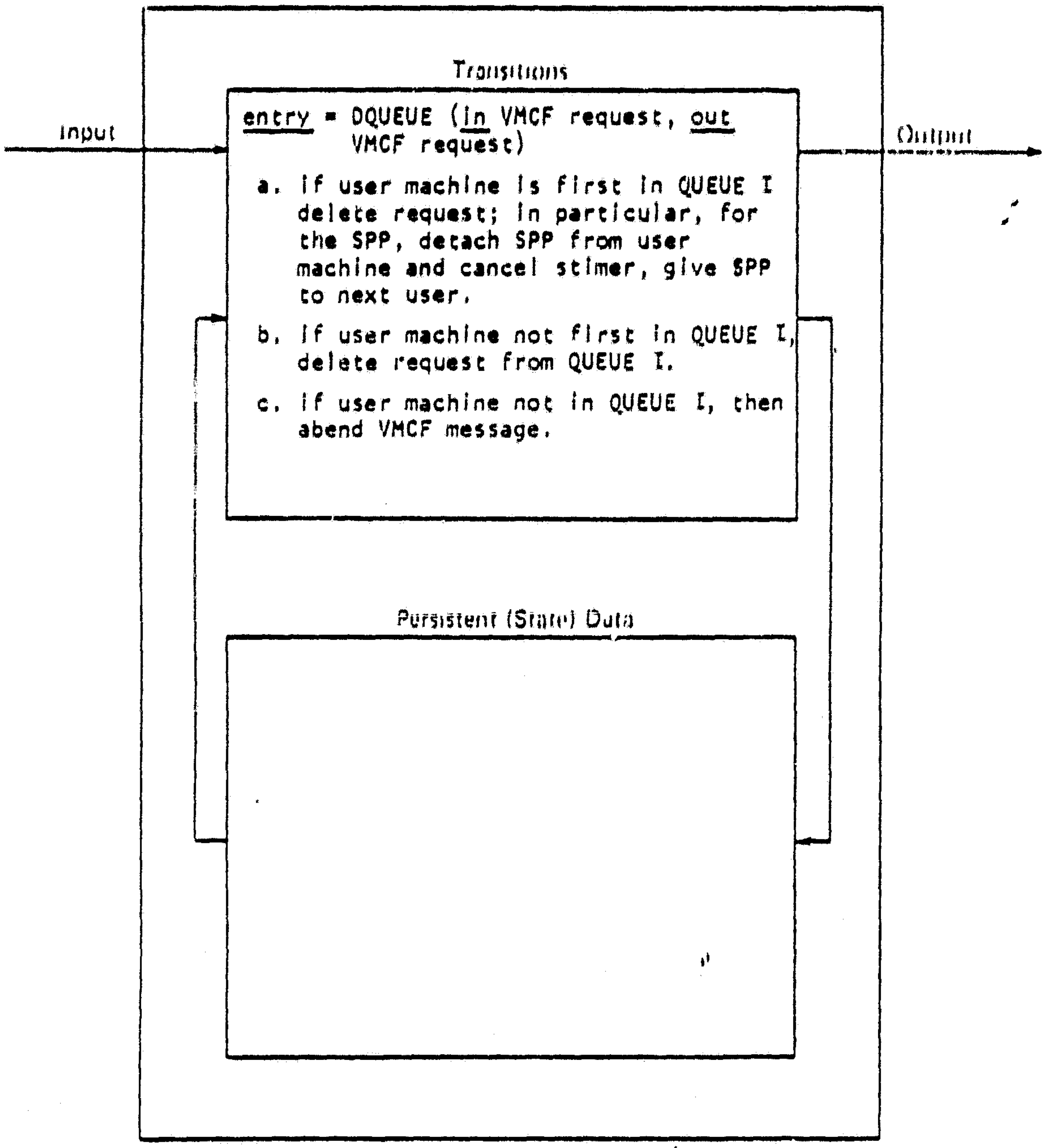
QUEUE MANAGER'S VIRTUAL MACHINE



Initial State: QUEUE 1, ..., QUEUE N: = nil, ... nil

FIGURE A.5-1

QUEUE MANAGER'S VIRTUAL MACHINE (CONT'D)



Initial State:

FIGURE A.5-1
(Cont'd)

QUEUE MANAGER'S VIRTUAL MACHINE (CONT'D)

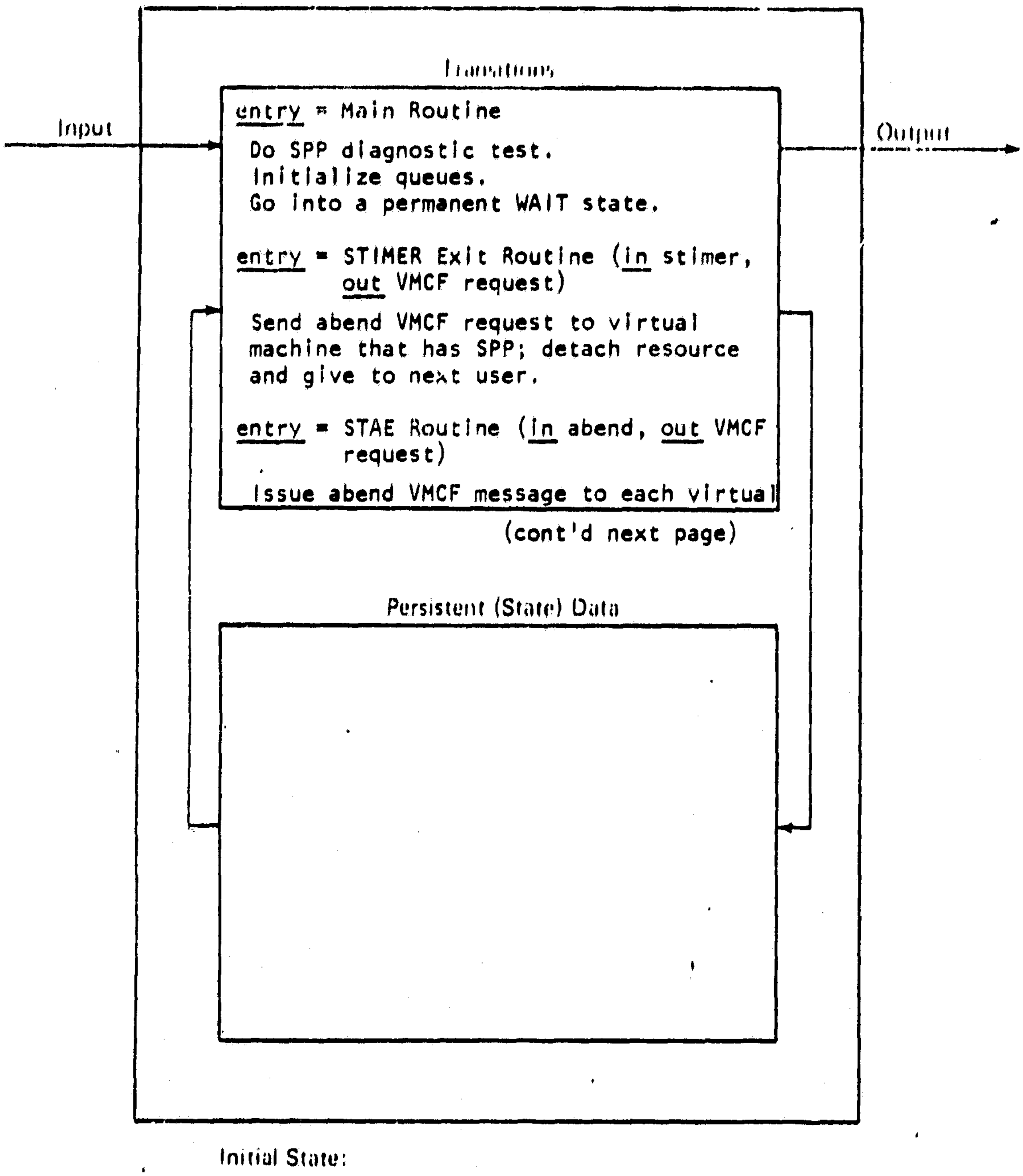


Figure A.5-1
(Cont'd)

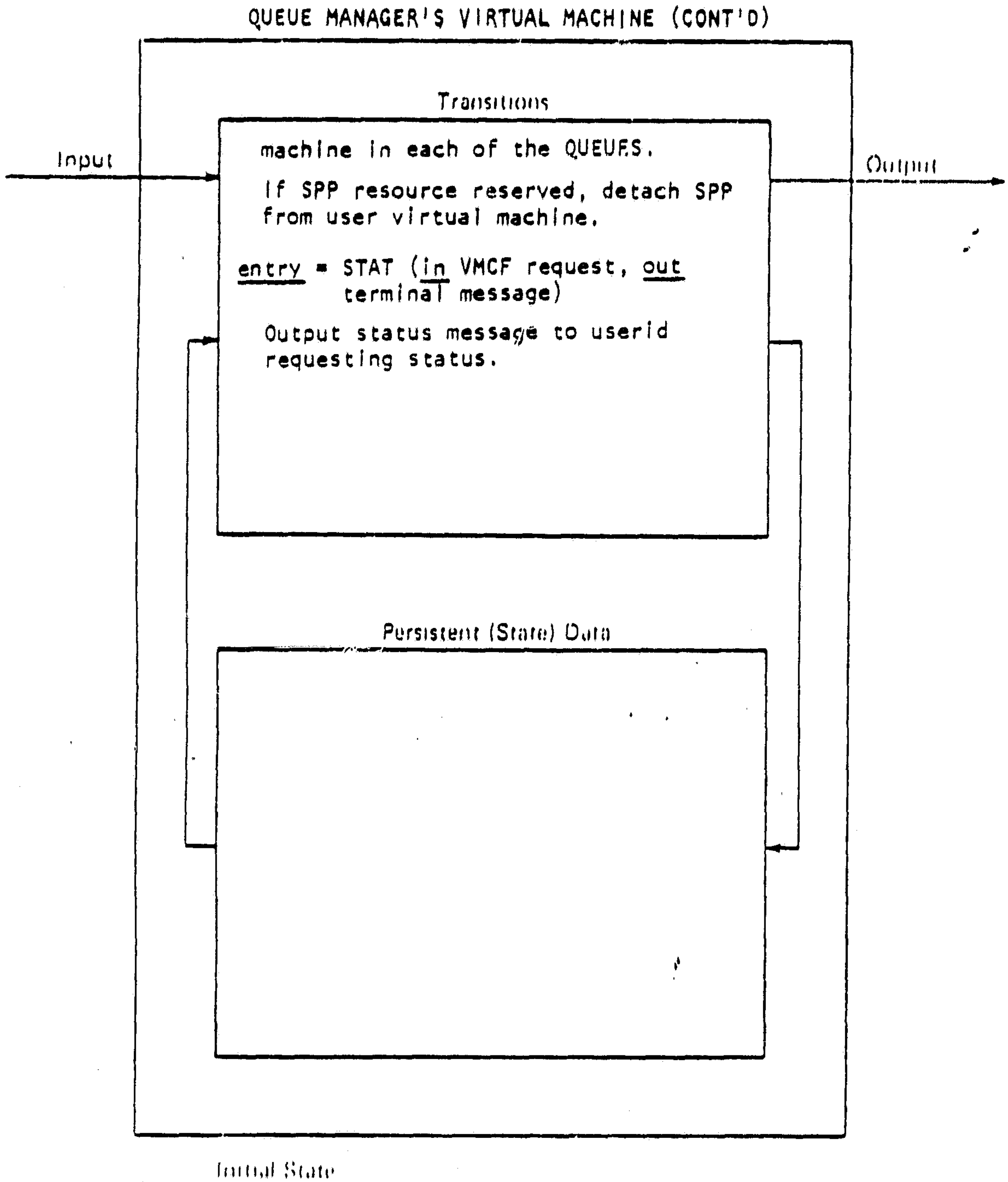


FIGURE A.5-1
(Cont'd)

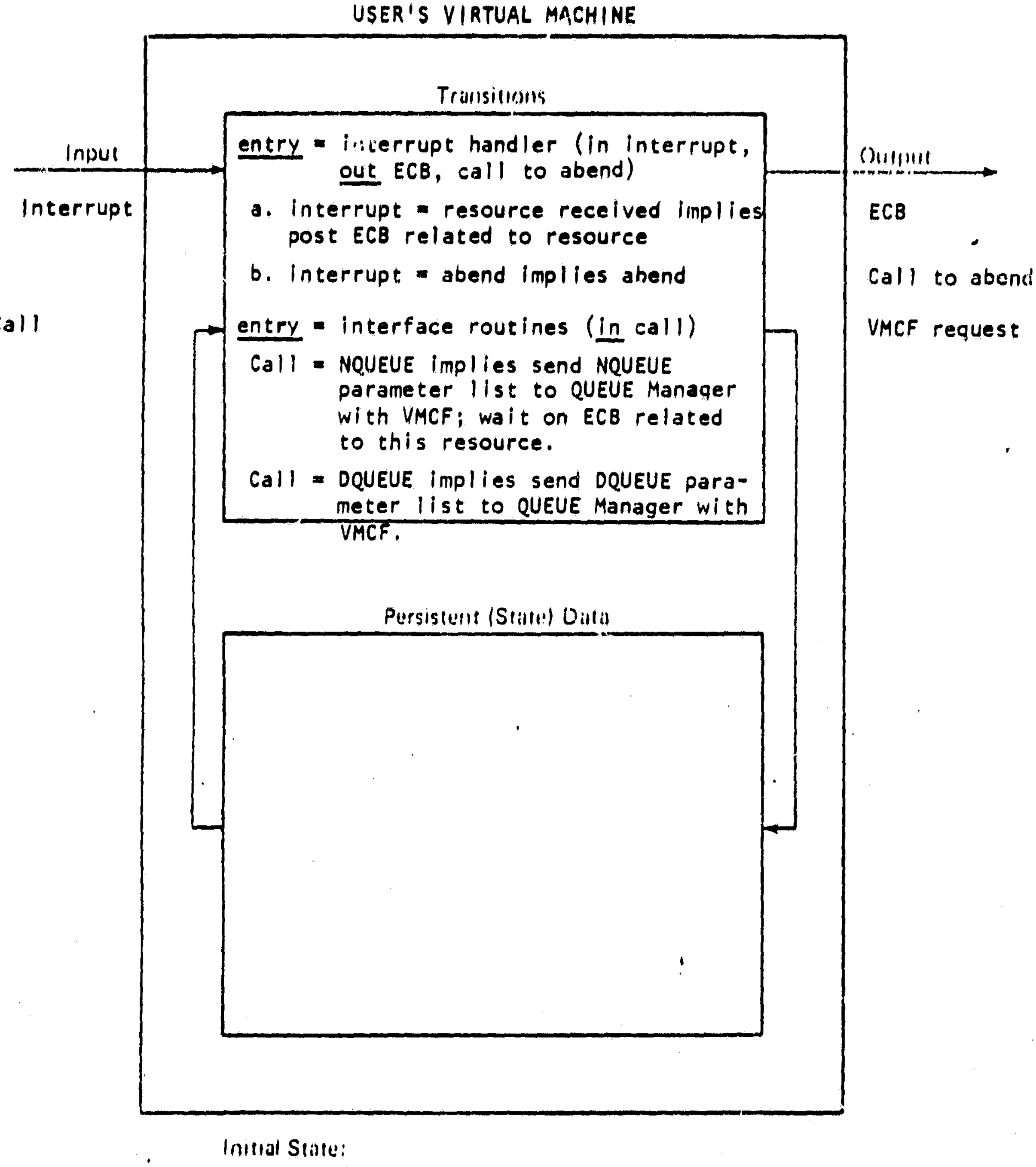

USER'S VIRTUAL MACHINE
Transitions
Input
Interrupt
Call
entry = interrupt handler (in interrupt, out ECB, call to abend)
a. interrupt = resource received implies post ECB related to resource
b. interrupt = abend implies abend
entry = interface routines (in call)
Call = NQUEUE implies send NQUEUE parameter list to QUEUE Manager with VMCF; wait on ECB related to this resource.
Call = DQUEUE implies send DQUEUE parameter list to QUEUE Manager with VMCF.
Output
ECB
Call to abend
VMCF request
Persistent (State) Data
Initial State:

Figure A.5-2

APPENDIX B

Glossary

Interface Control Record (ICR) - an application dependent record providing needed task management information to the SPP. A complete explanation is given in the Interface Control Document NASA/STARAN SPP (GER 16224).

Constants Record - an application dependent record providing needed application initialization information to the SPP. A complete explanation is given in the Interface Control Document NASA/STARAN SPP (GER 16224).

Contract NAS9-14350
JSC-16918
DRL Line Item No. 15

ERSYS SPP ACCESS METHOD SUBSYSTEM DESIGN

PDR MINUTES

APPROVAL SHEET

B. B. Duprey
ERSYS Application Systems

E. W. Poole
Earth Resources Systems

L. S. Wright
Support Software

S. A. Hudson
Ground Based Space Systems

10/13/80
A. A. Aldrich
Quality Assurance

NASA APPROVAL RECEIVED
ON TIRF 22113 10/28/80

APPENDIX C PDR MINUTES

ERSYS SPPAM Subsystem Design

PDR MINUTES

The preliminary design review for the ERSYS subsystem design was presented by P. Fong and R. Wiese, September 15, 1980, in Building 17 at NASA. The design presented included the following topics: Design Alternatives Considered, Design Overview, and Issues Awaiting Resolution. The presentation was intended to introduce the ERSYS SPPAM Subsystem Design Specification which incorporates design considerations in the ERSYS Level B Requirements and the ERSYS System Design Specification which pertain to the SPP under ERSYS and ERSYS SPP Applications.

SPECIAL PURPOSE PROCESSOR (SPP) INTERFACE

SPECIAL STUDY

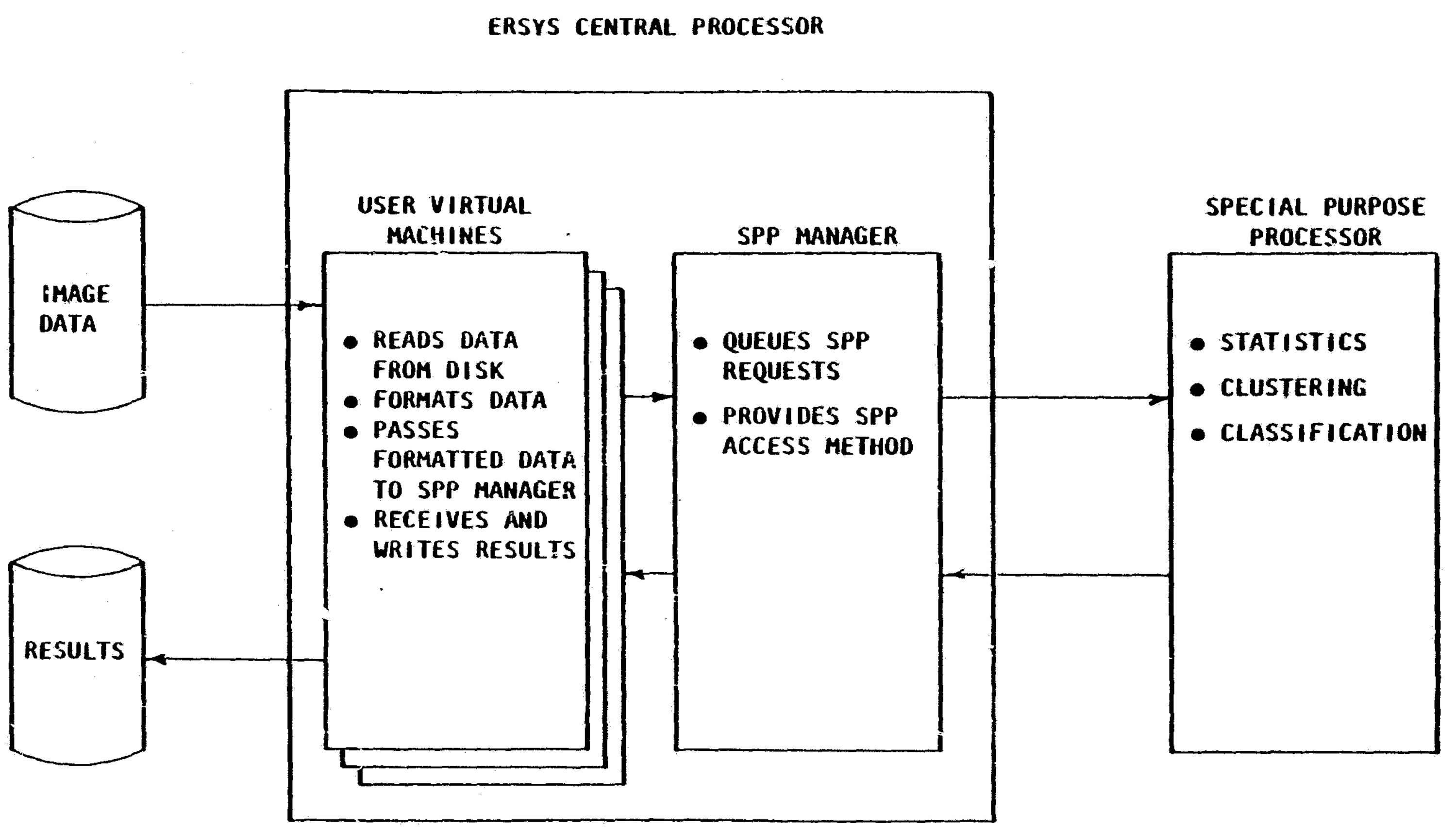
P. J. FONG

SEPTEMBER 16, 1980

PURPOSE

- PROJECT AND COMPARE PERFORMANCE OF ALTERNATE SPP INTERFACE CONFIGURATIONS.
- PROJECT AND COMPARE RESOURCE REQUIREMENTS OF ALTERNATE SPP INTERFACE CONFIGURATIONS.
- RECOMMEND A CONFIGURATION WHICH OFFERS BEST PERFORMANCE AND EFFICIENCY.

BASIC SYSTEM DESIGN



PERFORMANCE CRITERIA

- SHORT RESPONSE TIMES TO SPP REQUESTS
- HIGH SPP REQUEST THROUGHPUT
- EFFICIENT USE OF SYSTEM RESOURCES
- HIGH DEGREE OF PARALLEL PROCESSING BETWEEN SPP AND CENTRAL PROCESSOR

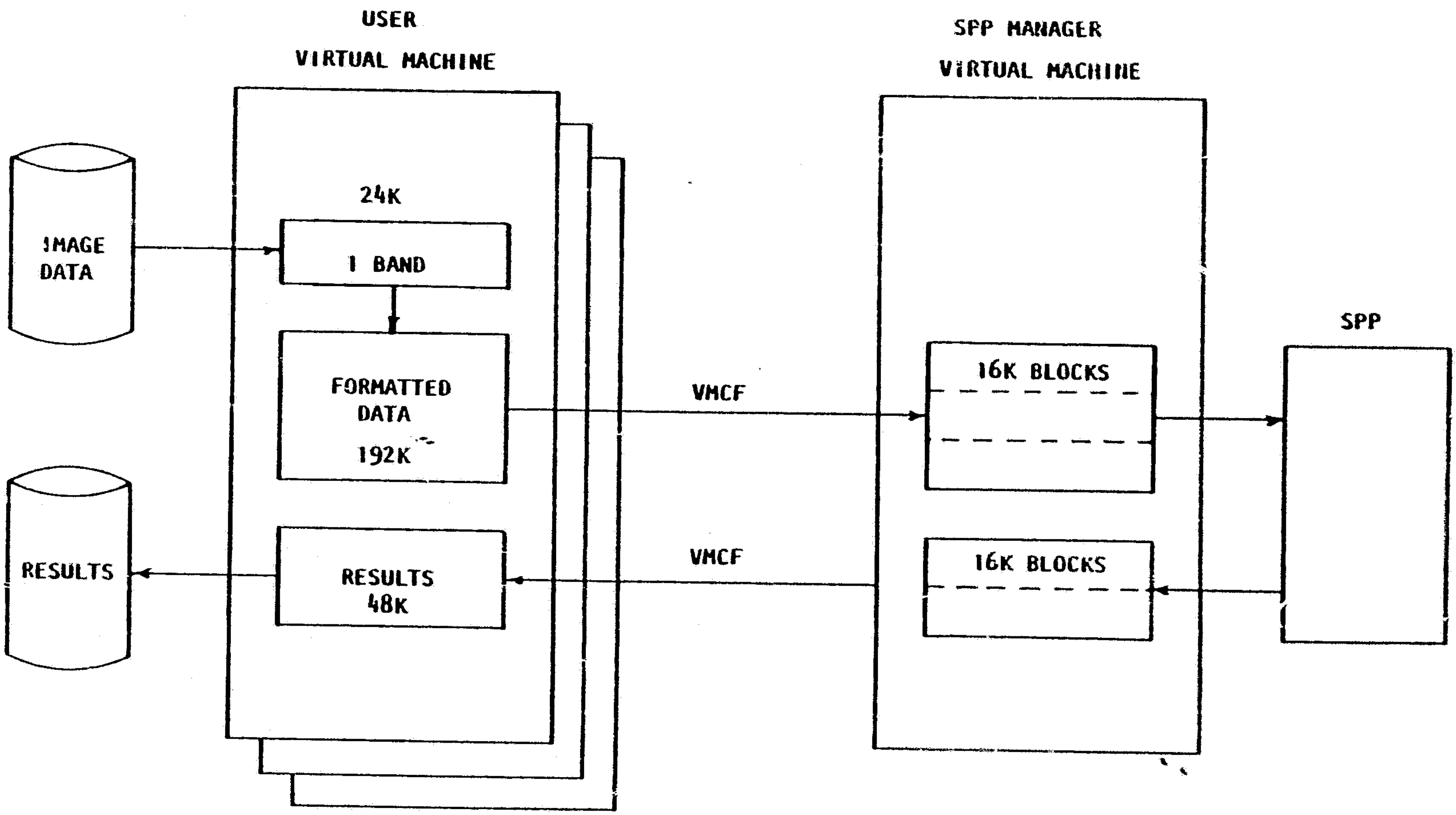
ASSUMPTIONS

- SPP PHYSICAL I/O:
  - THE SPP CAN SEND AND RECEIVE DATA BLOCKS NO LARGER THAN 20K.
  - THE SPP CAN RECEIVE DATA BLOCKS FROM ONLY ONE USER AT A TIME.
- THE SPP ACCESS METHOD (DIAGNOSE X'20') REQUIRES THE VIRTUAL MACHINE TO PERFORM "NO-WAIT" I/O.
- INPUT DATA:
  - IMAGE DATA WILL RESIDE ON CMS FILES IN EITHER BAND-INTERLEAVED OR BAND-SEQUENTIAL ORDER.
  - 2K OR 4K BLOCKSIZE.
  - DASD I/O IS NOT CHAINED.

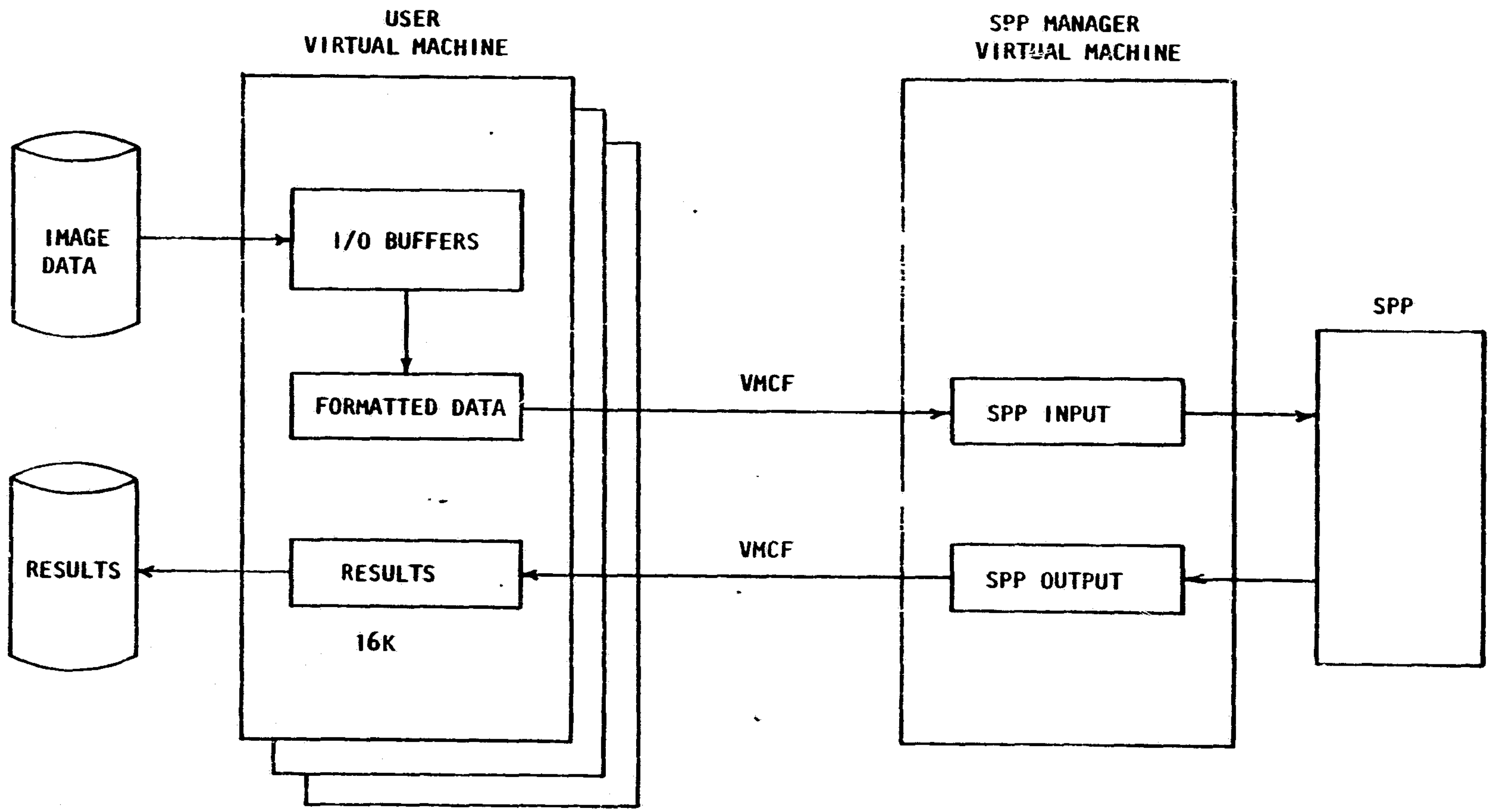
EXAMPLE SCENARIO

- EIGHT-CHANNEL LACIE-TYPE IMAGE TO BE PROCESSED BY SPP
- IMAGE DATA RESIDES ON 3350 DASD:
  - BAND-SEQUENTIAL (CHANNEL-SEQUENTIAL) ORDER
  - EACH BAND RESIDES IN A SEPARATE CMS FILE OR MINIDISK
  - 4K BLOCKSIZE
- 48K RESULT FROM SPP. LAST SEGMENT EXPECTED 200 MILLISECONDS AFTER ALL DATA RECEIVED BY SPP
- MINIMUM CONTENTION

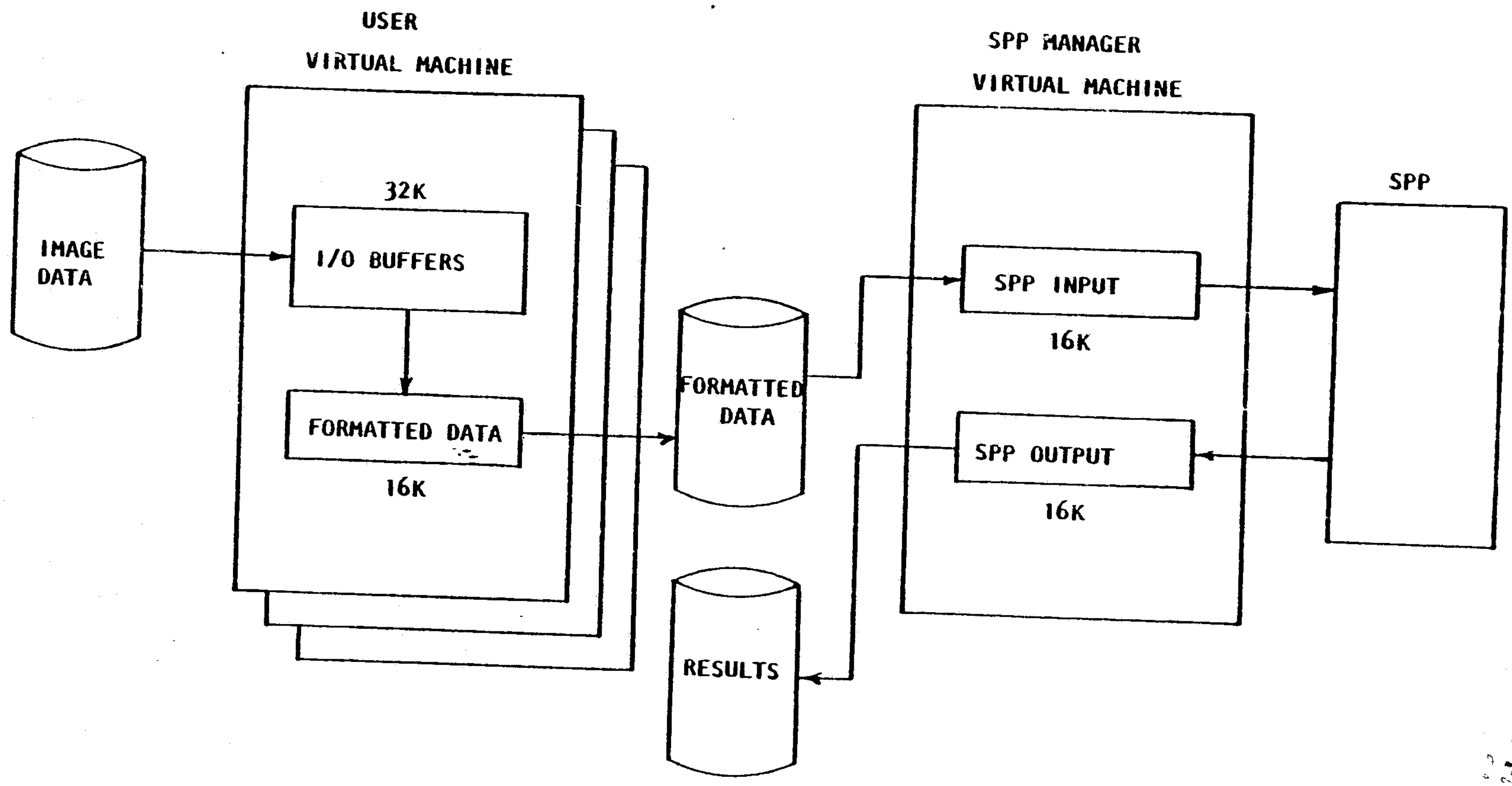
CONFIGURATION #1



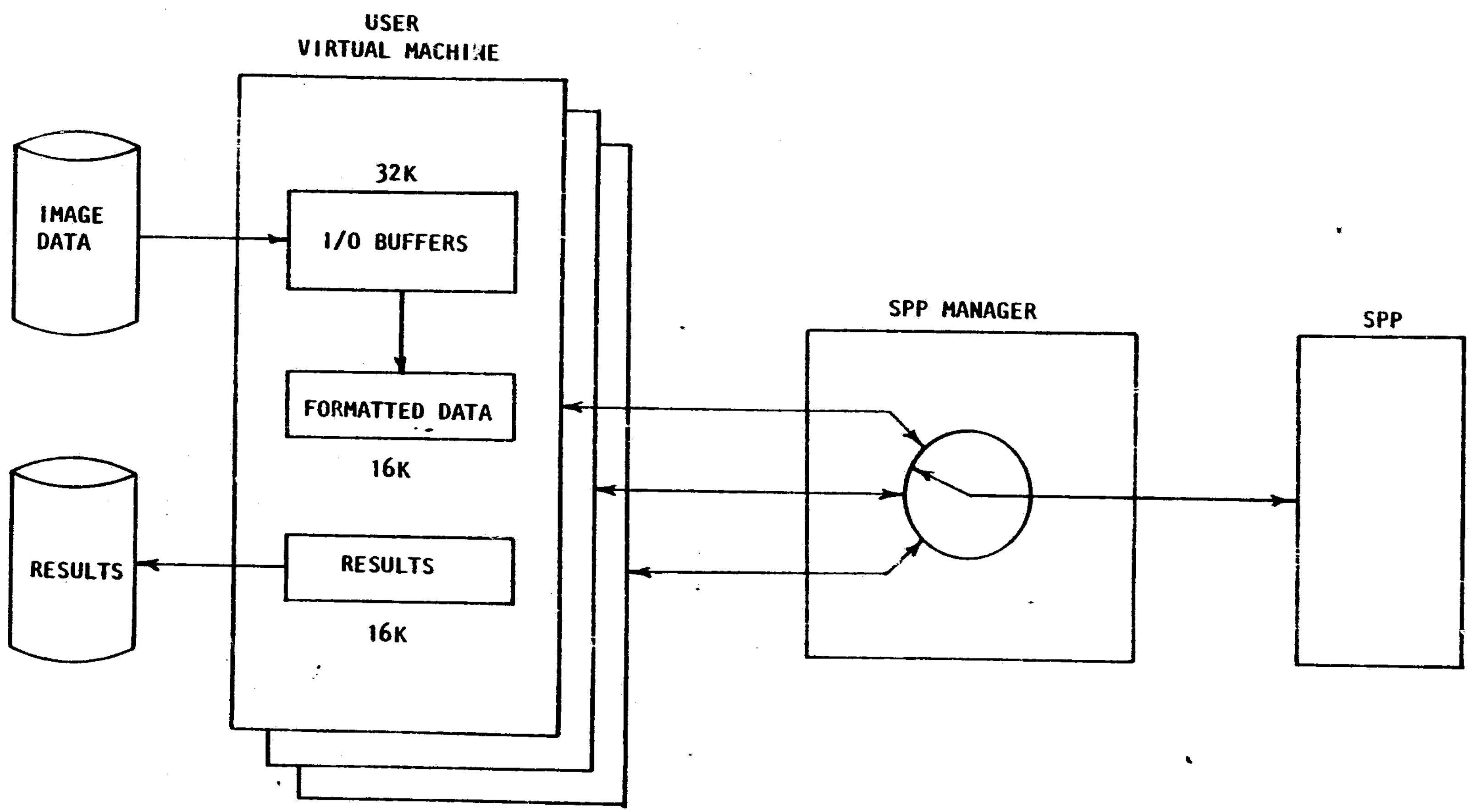
CONFIGURATION #2



CONFIGURATION #3



CONFIGURATION #4



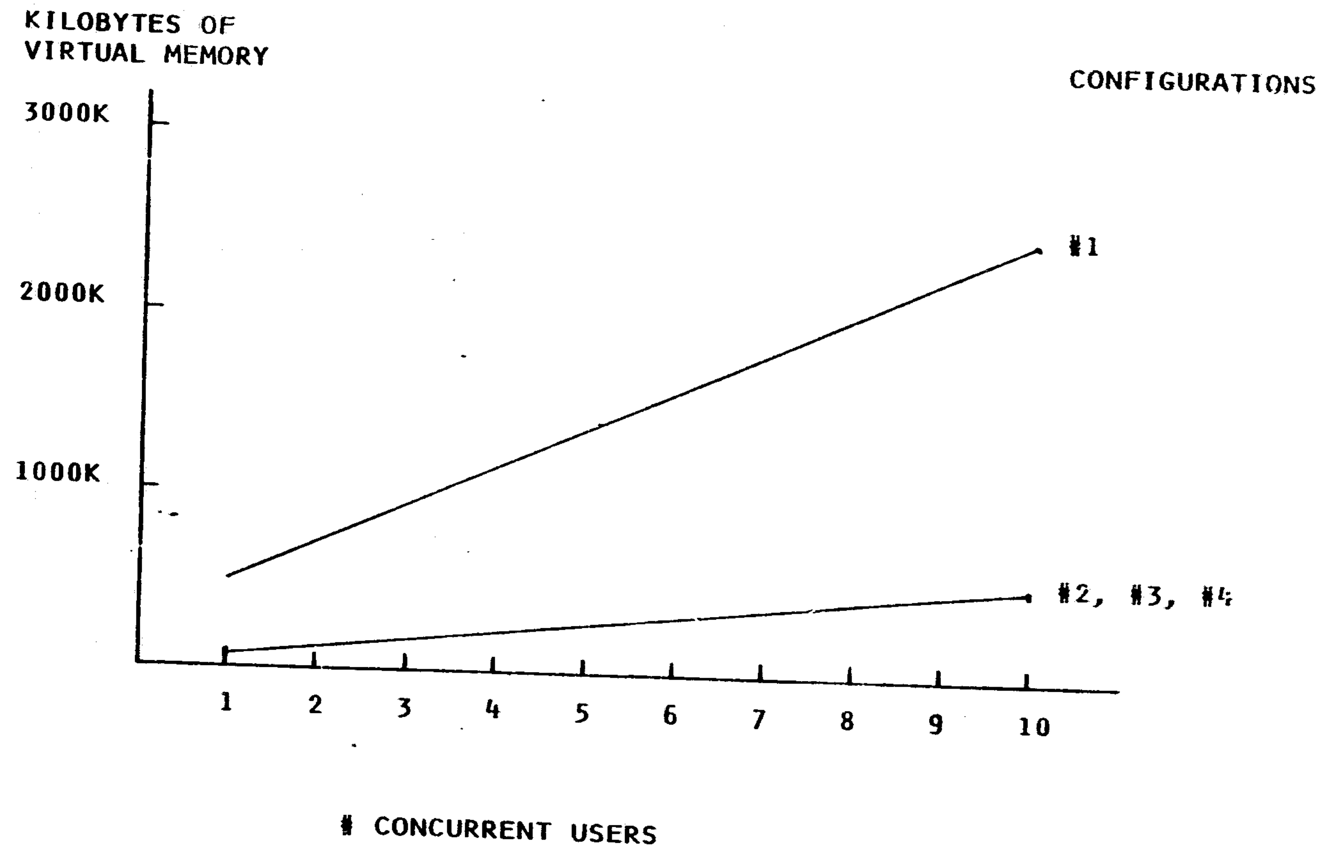
RESULTS

- NOMINAL SPP RESPONSE TIME BETWEEN DATA READS IS 100 TO 200 MILLISECONDS
- DASD I/O SERVICES COMPRISE MOST OF CPU EXPENDED
- CPU EXPENDED FOR FORMATTING IS INSIGNIFICANT. (~10 MS FOR 8-CHANNEL LACIE IMAGE)
- NOMINAL TIME REQUIRED TO READ 16K OF IMAGE DATA:

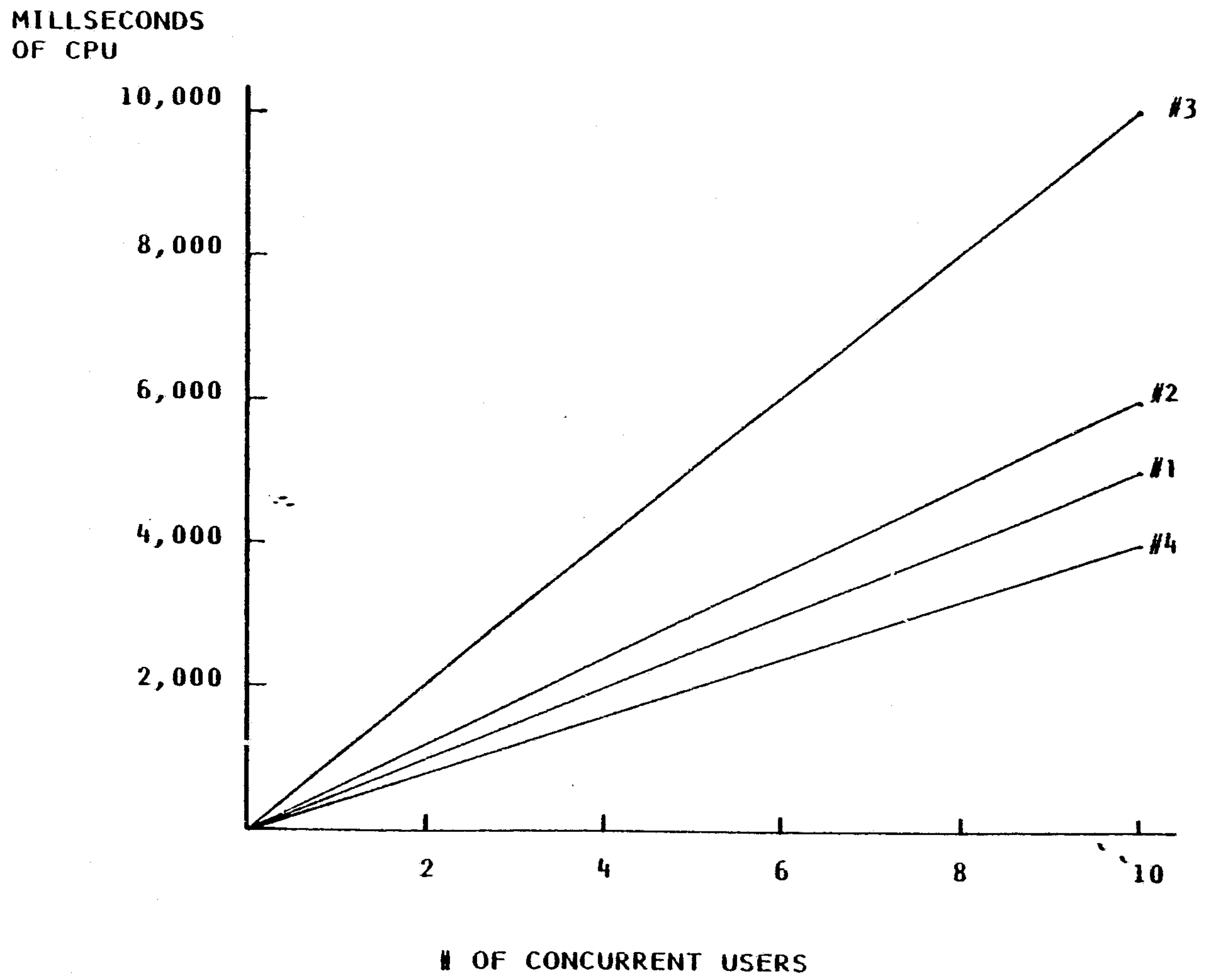
  ~100 MILLISECONDS FOR 4K BLOCKSIZE (APPENDIX I)

  ~200 MILLISECONDS FOR 2K BLOCKSIZE

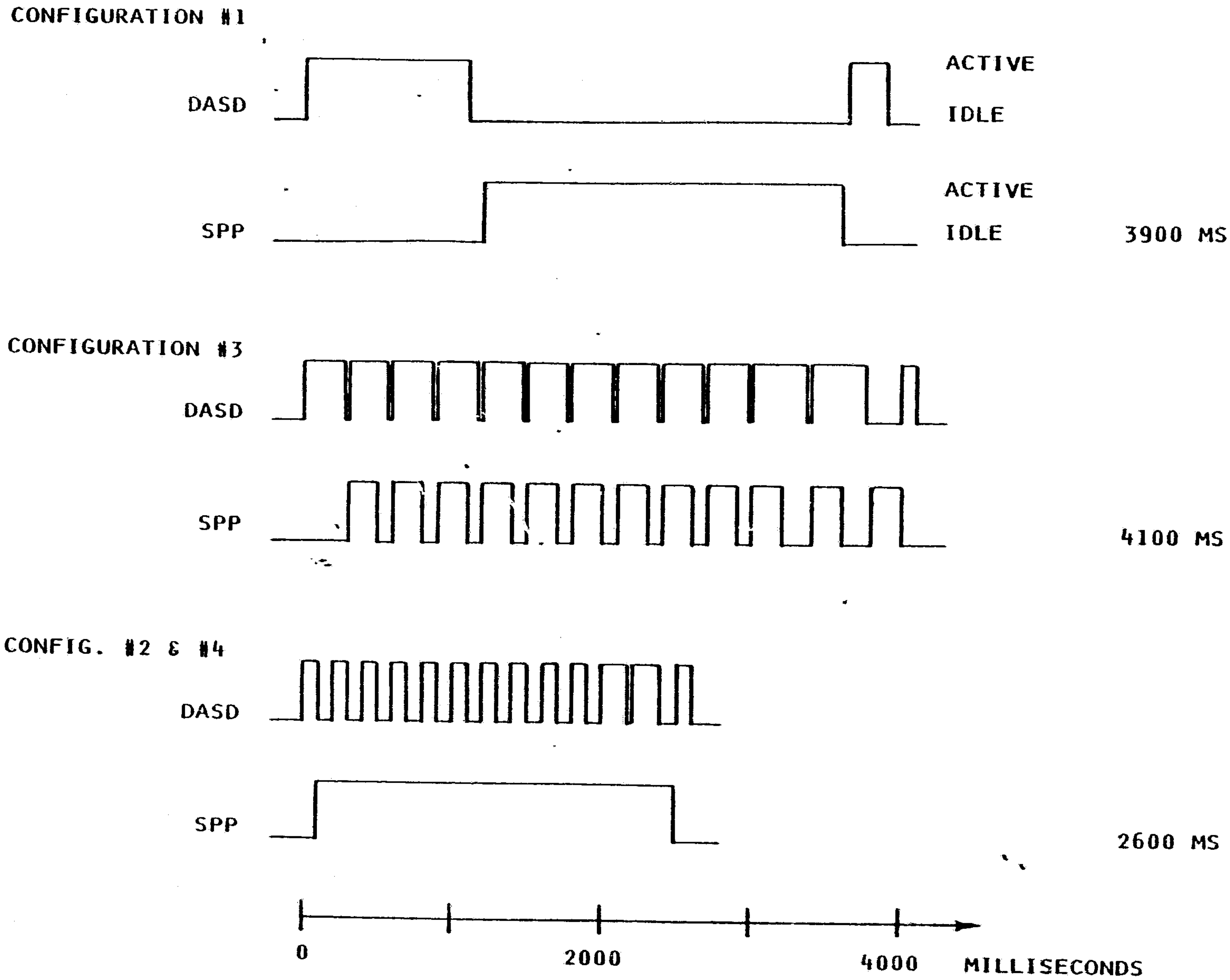
MULTIPLE USERS: VIRTUAL MEMORY



MULTIPLE USERS: CPU

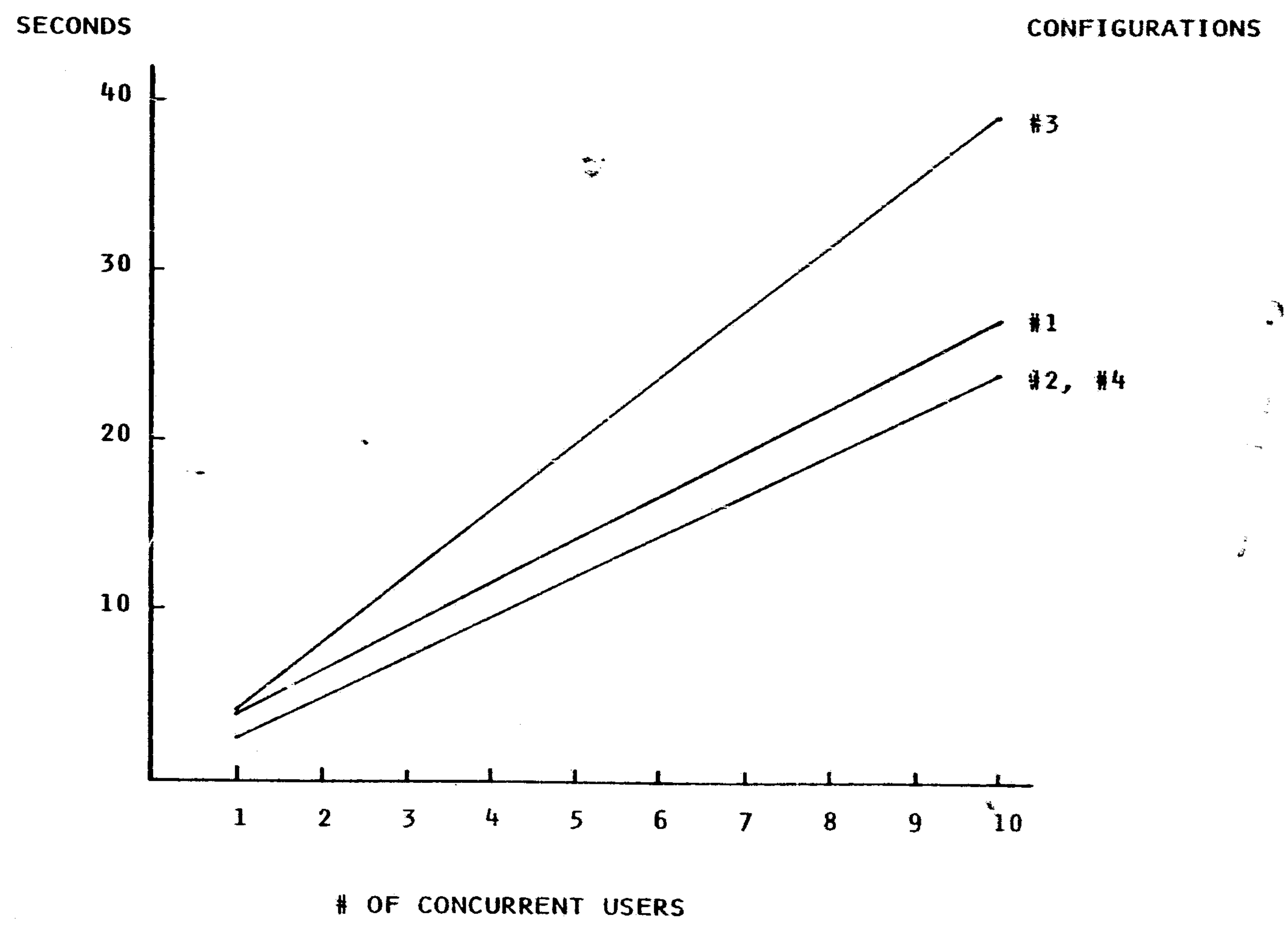


RESPONSE TIME: SINGLE USER



MULTIPLE USERS: TOTAL THROUGHPUT TIMES

(REF: APPENDIX II)



MULTIPLE USERS: SUMMARY

(10 USERS)

CONFIGURATION	VIRTUAL MEMORY (K)	CPU (MS)	THROUGHPUT (SEC.)
1	2450	5,000	27
2	530	6,000	24
3	510	10,000	39
4	500	4,000	24

(REF. APPENDIX III)

BULK DATA AND ITERATIVE OPERATIONS

- ERTS BULK DATA
  - 10.1 MEGABYTES PER CHANNEL, 53 MB/IMAGE
  - NOT ADDRESSED BY CONFIGURATION #1
- ITERATIVE OPERATIONS
  - FUNCTION MAY BE REQUIRED TO SAVE FORMATTED DATA

IMAGE DATA ORDERING CONSIDERATIONS

- BAND-SEQUENTIAL ORDER
  - REQUIRES AT LEAST ONE READ PER CHANNEL
  - MAY REQUIRE SEPARATE FILES PER CHANNEL TO MAINTAIN MORE CMS I/O BUFFERS
- BAND-INTERLEAVED ORDER
  - NO. READS DEPEND ONLY ON BUFFER SIZE
  - REQUIRES ONLY ONE I/O BUFFER

CONCLUSIONS

- CONFIGURATIONS #2 AND #4 OFFER HIGHEST POTENTIAL FOR HIGH THROUGHPUT WITH LOW RESOURCE USAGE.
- CONFIGURATION #1 HAS A HIGH PAGING REQUIREMENT AND DOES NOT ADEQUATELY ADDRESS PROCESSING OF BULK DATA.
- CONFIGURATION #3 WILL RESULT IN LOW THROUGHPUT AND HIGH CPU USAGE DUE TO LOW SPP UTILIZATION AND HIGH DASD I/O REQUIREMENTS (APPENDIX IV).
- CONFIGURATION #2 OFFERS SAME THROUGHPUT AS #4 BUT REQUIRES GREATER CPU.
- BAND-INTERLEAVED DATA REDUCES DASD I/O OVERHEAD AND BUFFER REQUIREMENTS.
- 4K BLOCKSIZE WILL HALVE CPU OVERHEAD AND DASD RESPONSE TIMES.

RECOMMENDATIONS

- ADOPT CONFIGURATION #4 FOR SPP INTERFACE.
  - ADD A FUNCTION TO SAVE FORMATTED DATA FOR ITERATIVE OPERATIONS
  - ALLOW NUMBER OF FORMATTED BUFFERS QUEUED TO BE ADJUSTABLE
  - INVESTIGATE FAVOURED EXECUTION FOR USER VIRTUAL MACHINE
- USE 4K BLOCKSIZE FOR IMAGE DATA IN CMS FILES
- ADOPT BAND-INTERLEAVED ODERING OF IMAGE DATA
- IF BAND-SEQUENTIAL DATA IS ADOPTED:
  - PLACE EACH CHANNEL ON ITS OWN CMS FILE OR MINIDISK
  - ALLOW MULTIPLE DATA BUFFERS TO BE FORMATTED CONCURRENTLY
- AVOID FRAGMENTED FILES

APPENDIX I

3350 DASD I/O TIMING

1 ST BLOCK:	AVERAGE ACCESS MOTION	25.0 MILLISECONDS
	AVERAGE ROTATIONAL DELAY	8.4
	DATA TRANSFER (4K BLOCK)	3.4
		36.8 MS
NTH BLOCK: (SEQUENTIAL UNCHANGED)	ACCESS MOTION	0.0
	FULL ROTATION	16.4
	DATA TRANSFER + GAP	4.1
		20.5 MS
NTH BLOCK: (SAME CYL.)	ACCESS MOTION	0.0
	AVERAGE ROTATIONAL DELAY	8.4
	DATA TRANSFER (4K BLOCK)	3.4
		11.8 MS
NTH BLOCK: (DIFF. CYL.)	ACCESS MOTION (1-8 CYL.)	8.0
	AVERAGE ROTATIONAL DELAY	8.4
	DATA TRANSFER (4K BLOCK)	3.4
		19.8 MS

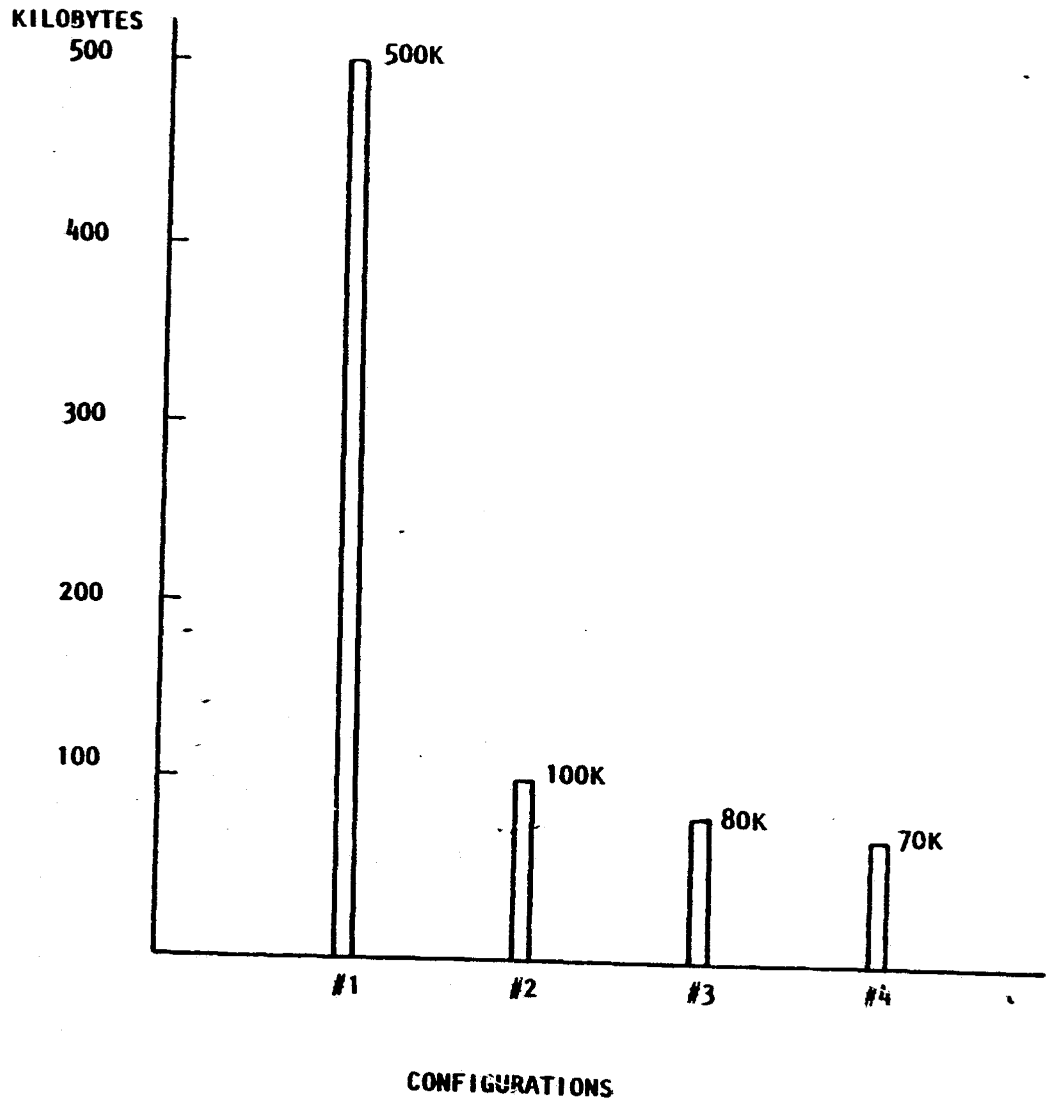
APPENDIX II

MULTIPLE USERS: THROUGHPUT TIME FORMULAS

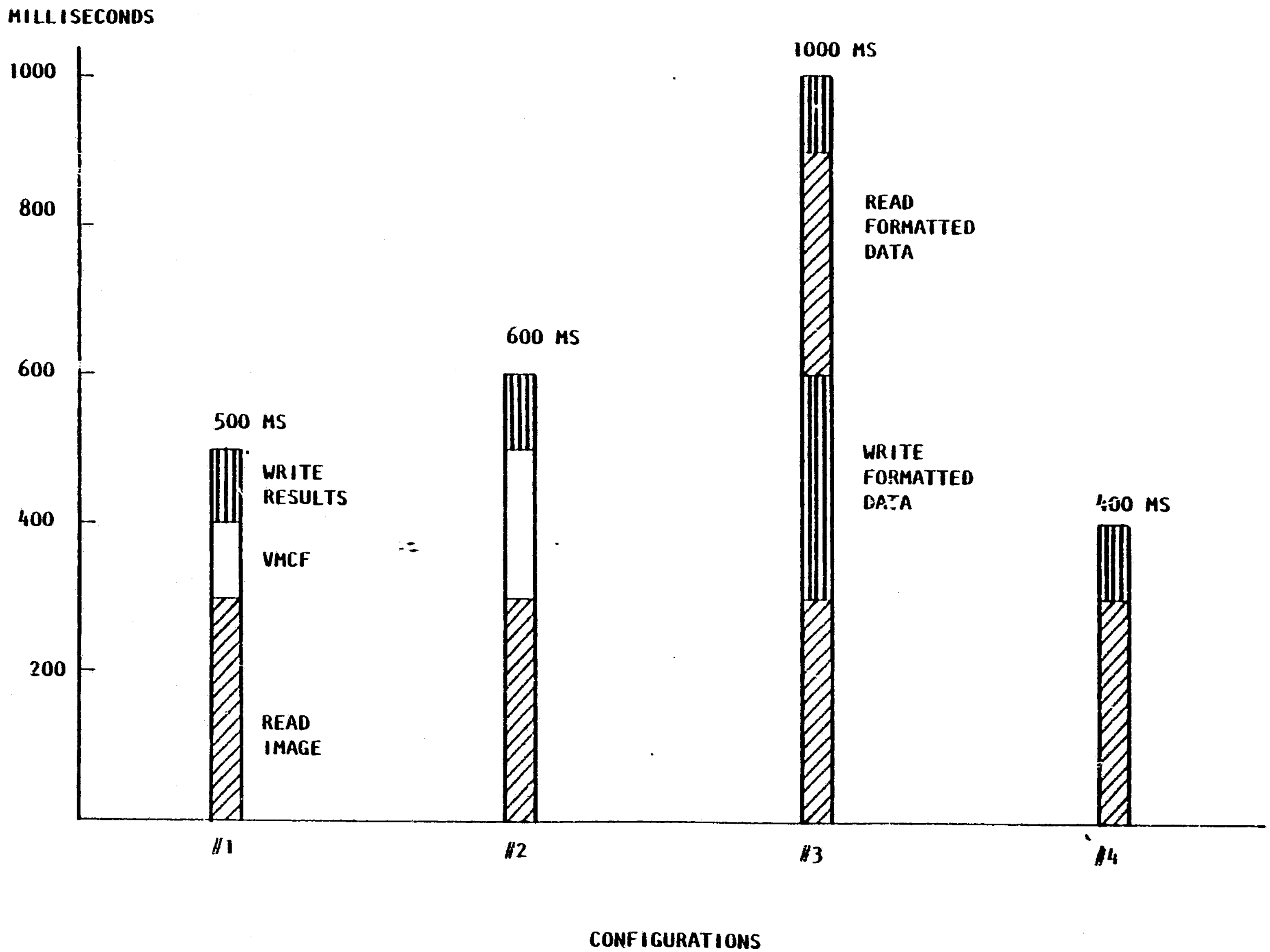
CONFIGURATION	TOTAL THROUGHPUT TIME
#1	192K READ + [(#USERS * (192K VMCF + SPP)] + 48K WRITE
#2 & #4	16K READ + [#USERS * (16K VMCF + SPP)] + 16K WRITE
#3	16K READ + 16K WRITE + [#USERS * (192K READ + SPP + 48K WRITE)]

APPENDIX III

SINGLE USER: VIRTUAL MEMORY



SINGLE USER: TOTAL CPU



APPENDIX IV

SPP UTILIZATION

CONFIGURATION	MAX. SPP UTILIZATION
#1	→ 100%
#2	→ 100%
#3	67%
#4	→ 100%

SINGLE USER ANALYSIS: SUMMARY

CONFIGURATION	VIRTUAL MEMORY(K)	CPU(MS)	RESPONSE(MS)
1	500	500	3900
2	100	600	2600
3	80	1000	4100
4	70	400	2600

APPENDIX V

DASD CONTENTION CONSIDERATIONS

CONFIGURATION	DASD: SSP OVERLAP RATIO	MARGIN OF SAFETY PER DASD-SPP I/O
#1	1:2	1200 MS
#2	1:2	100 MS
#3	3:2	-100 MS
#4	1:2	100 MS

APPENDIX VI

PAGING CONTENTION CONSIDERATIONS

CONFIGURATIONS	VIRTUAL REQUIRED (K) (10 USERS)	AVAILABLE REAL 1000K	500K	250K
#1	2450	59%	80%	90%
#2	530	0%	6%	53%
#3	510	0%	2%	51%
#4	500	0%	0%	50%

% PAGE FAULTS

ERSYS
SPECIAL PURPOSE PROCESSOR
ACCESS METHOD
DESIGN

ROBERT WIESE

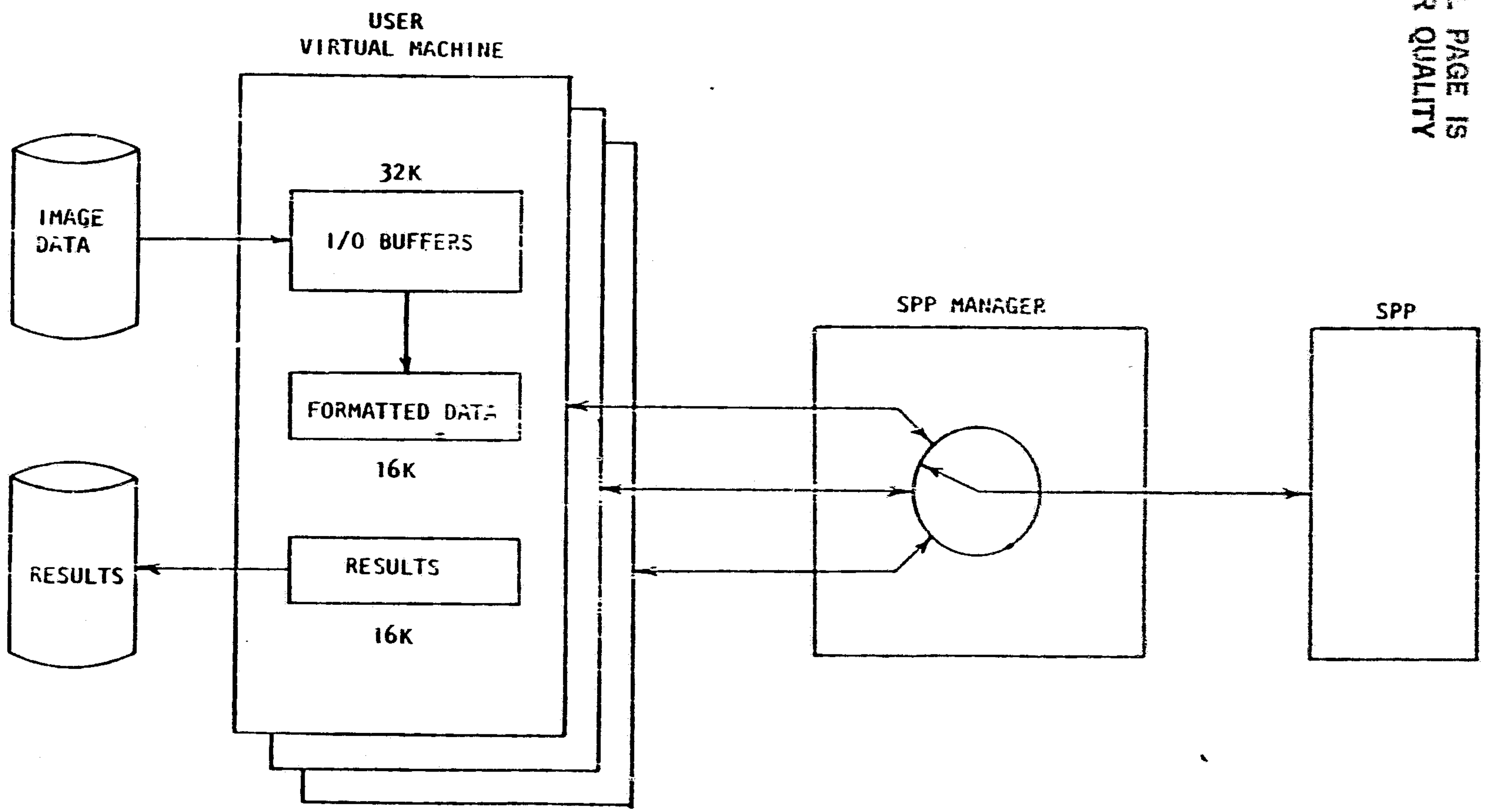
9/16/80

CONFIGURATION 2 VERSUS CONFIGURATION 4

- BOTH EQUAL VIS-A-VIS SYSTEM PARAMETERS
- LESS COMPLEXITY WITH 4
  - NO VIRTUAL MACHINE COMMUNICATION FACILITY REQUIREMENTS
  - SYSTEM PROVIDED QUEUE HANDLING MECHANISM

CONCLUSION: IMPLEMENT 4

CONFIGURATION #4



IBM

AgRISTARS

SPP APPLICATION

** RESERVE SPP AND SEND ICR.

FRMTICR

SPPRES

** PROCESS TASK

<u>WHILE</u>

(NOT ALL RESULTS ARE BACK) AND (THERE IS NO I/O ERROR)

<u>DO</u>

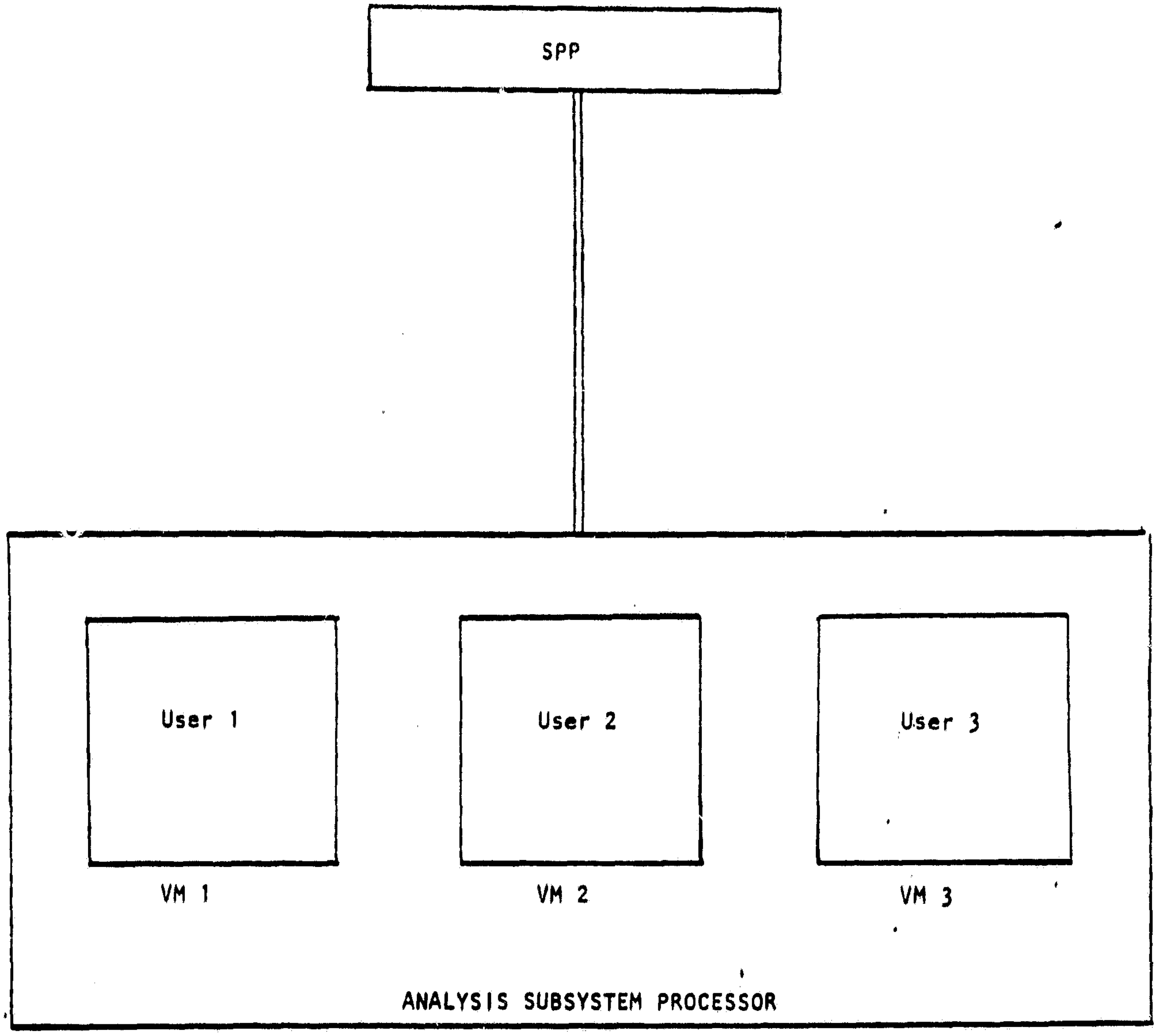
FRMTBUF

SPPRDWRT

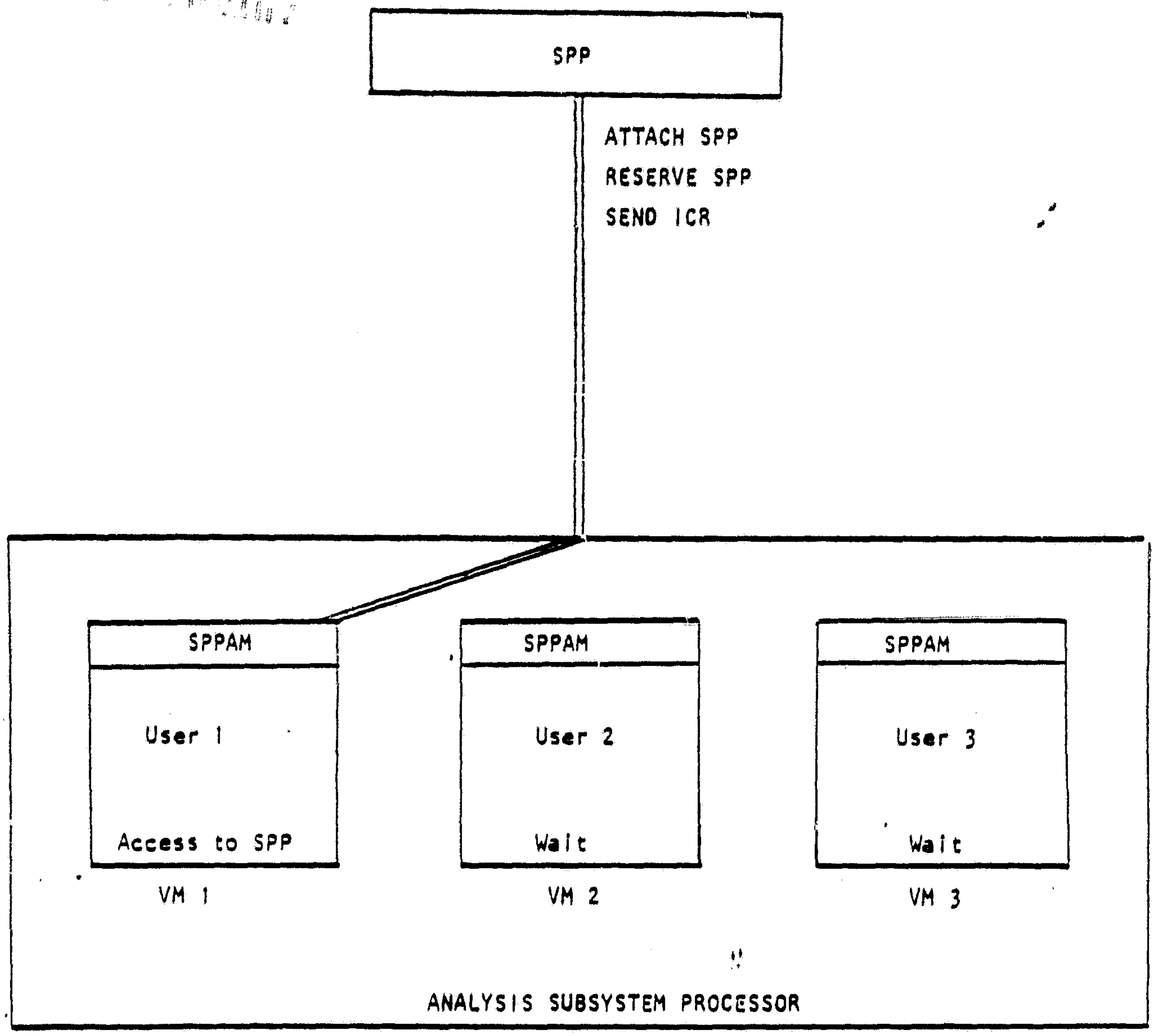
WRTRES

<u>OD</u>

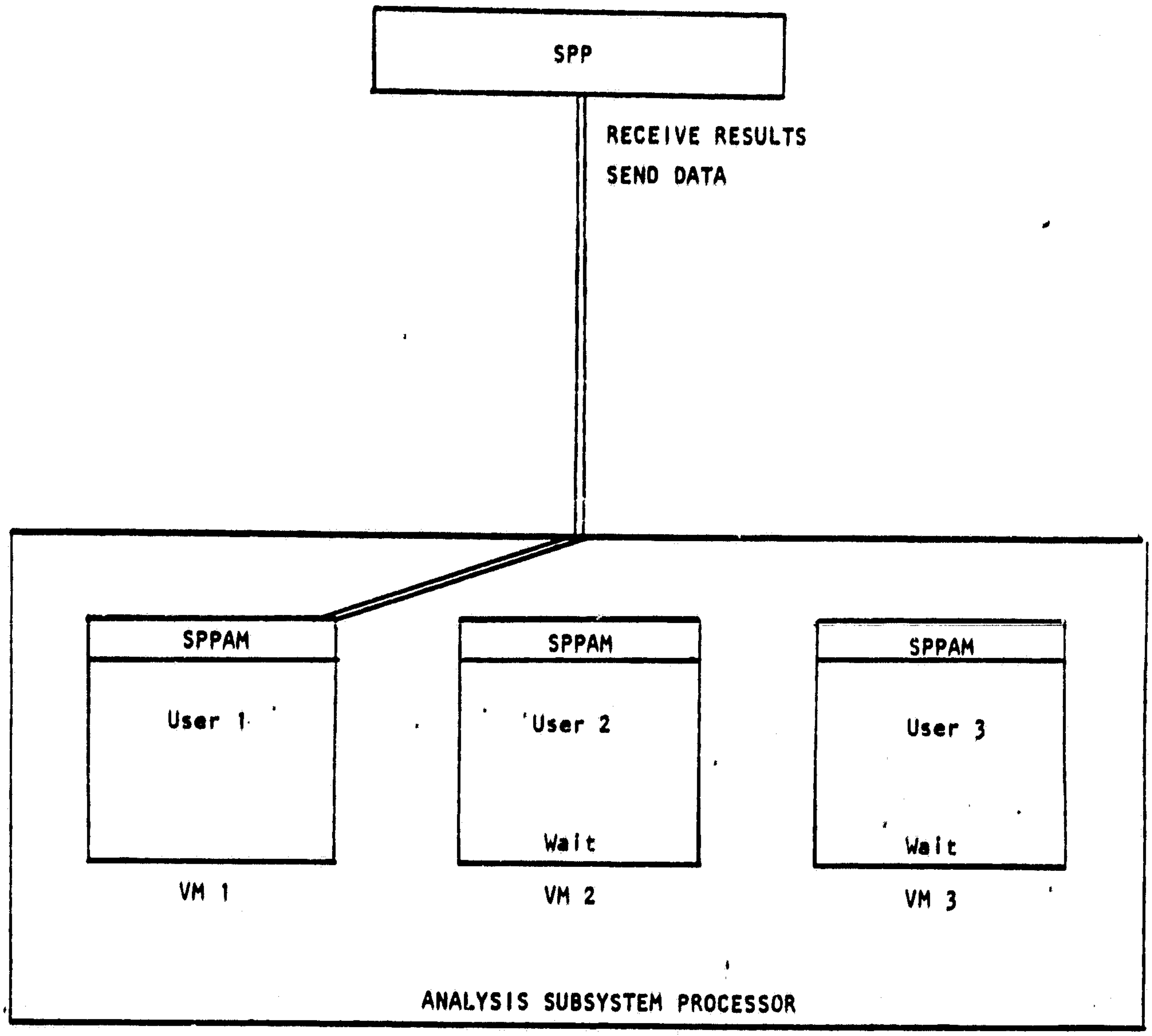
SPPREL


SPP
User 1
User 2
User 3
VM 1
VM 2
VM 3
ANALYSIS SUBSYSTEM PROCESSOR

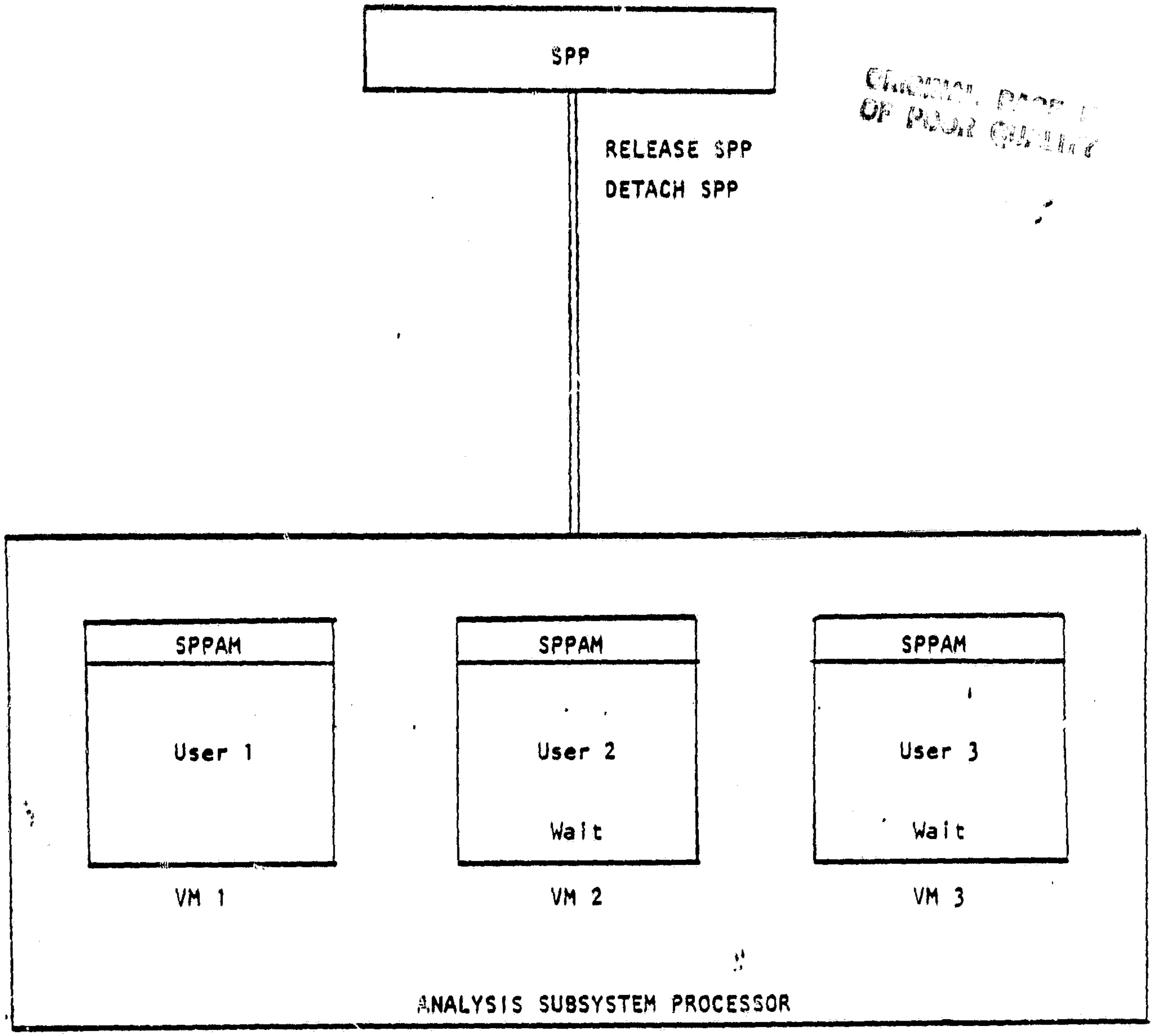
SPPAM IDLE



USERS INVOKE SPPAM
SPPRES



PROCESS REQUEST
SPPRDWRT



RELEASE SPP
SPP REL

ABORT TASK

- OCCURS WHEN:
  - USER TAKES TOO LONG TO SEND DATA TO THE SPP
  - SPP REFUSES TO RESPOND TO COMMANDS ISSUED BY HOST
  - I/O ERROR TAKES PLACE DURING A DATA TRANSFER
- RESULTS:
  - USER TASK IS ENDED
  - NEXT USER ALLOWED ACCESS TO SPPAM

OUTSTANDING ISSUES

- WHEN APPLICATIONS ARE DEVELOPED ROUTINES WILL BE WRITTEN WHICH PASS NECESSARY PARAMETERS TO THE SPPAM FUNCTIONS. THESE ROUTINES WILL BE AVAILABLE FOR ALL AUTHORIZED USERS.

- THE PRIORITY QUEUEING WILL BE THAT GENERATED BY THE COMMAND WHICH ALLOWS ONLY ONE USER TO ACCESS THE SPP AT A TIME.

- RELEVANT PARTS OF THE SYSTEM DESIGN SPECIFICATION AND THE LEVEL B REQUIREMENTS WILL BE AMENDED SO THAT CONFIGURATION 4 WILL BE CONSISTENT WITH THESE DOCUMENTS.

THE RESOLUTION OF THESE ISSUES WILL BE PRESENTED AT THE CDR.

ACTION ITEMS

The following matters were action items identified as a result of the PDR:

1. IBM - Amend the System Design Specification and the Level B requirements so that these documents and the Special Purpose Processor Access Method are consistent with each other.

This action item will have been resolved by CDR time.

The major questions raised, by presentation topic, are listed below with responses to those questions.

Design Alternatives Considered

Q. How are the design alternatives affected by contention?

A. Those alternatives which require more system resources are also more vulnerable to contention as a result. Hence, configurations #4 and #2 will remain our choices in a high contention environment.

Design Overview

Q. Will the user, before detaching, be able to manipulate results within SPPAM?

A. No.

Q. If a user cannot obtain the SPP immediately and does not want to wait, what can he do?

A. The user can cancel the command.

Q. Can the user receive status information on the queue?

A. No, with the LOCK macro this is not possible.

Issues Awaiting Resolution

Q. What is the priority queueing scheme provided by the LOCK macro?

A. The answer will be given at CDR.

PDR Attendees

NAME	ORGANIZATION
M. Alexander	NASA/SF6
K. Baker	NASA/SF3
K. Bulow	NASA/SF6
J. Carney	NASA/SF4
B. Duprey	IBM/HR4
J. Erickson	NASA/SF3
P. Fong	IBM/HJ5
F. Flowers	IBM/HM9
J. Gilbert	NASA/SF6
B. Goode	NASA/SF6
D. Hay	NASA/SF6
M. Heidt	NASA/SF6
M. Lopez	NASA/SF6
J. Mohon	IBM/HJ5
E. Poole	IBM/HN8
R. Sormani	IBM/HN8
D. Truitt	IBM/HM9
J. Ulrich	IBM/HK8
L. Webster	NASA/SF6
W. Weimer	NASA/SF6
R. Weise	IBM/HR4