

SSEC No76.07.II

THE SCHWERTFEGER LIBRARY
1225 W. Dayton Street
Madison, WI 53708

INNOVATIVE VIDEO APPLICATIONS

IN METEOROLOGY (IVAM)

ANNUAL REPORT JULY 1976

A REPORT

from the space science and engineering center
the university of wisconsin-madison
madison, wisconsin

INNOVATIVE VIDEO APPLICATIONS

I. INTRODUCTION

IN METEOROLOGY (IVAM)

ANNUAL REPORT JULY 1976

A. Scope and Objectives	I-1
B. Background and Objectives	I-1
C. Delay of Fiscal Year 1976 Report	I-5
D. The Need for IVAM	I-8

An Annual Report

II. PRESENTATION CONCEPT STUDY	II-1
--------------------------------	------

III. SOFTWARE DEVELOPMENT Contract 3-35156

A. Introduction	III-1
B. IVAM Operating System	III-10
C. Needs For Period 1 August 74 - 31 July 1976	III-20
D. Conclusion	III-30

IV. HARDWARE CONCEPT STUDIES

A. The IVAM System	IV-1
B. System Timing and Sizing	IV-3
C. AFOS Interface	IV-8
D. Development System Program	IV-13
E. Prototype Element Configuration	IV-14

Submitted to:

U.S. Dept. of Commerce
National Oceanic and Atmospheric Administration
6010 Executive Blvd.
Rockville, Maryland 20852

V. OUTPUT DISTRIBUTION STUDY

A. Broadcast Network Distribution	V-1
B. Distribution to Broadcast TV Stations	V-2

Prepared by:

Space Science and Engineering Center
University of Wisconsin-Madison
1225 West Dayton Street
Madison, Wisconsin 53706

TABLE OF CONTENTS

1 May 75 - 1 May 76

I.	INTRODUCTION	
A.	Scope and Organization of This Report	I-1
B.	Background and Objectives	I-1
C.	Delay of Fiscal Year 1976 Effort	I-6
D.	The Need for IVAM	I-8
II.	PRESENTATION CONTENT STUDY	II-1
III.	SOFTWARE DEVELOPMENT	
A.	Introduction	III-1
B.	IVAM Operating System	III-10
C.	Modules	III-20
D.	Conclusion	III-38
IV.	HARDWARE CONCEPT STUDIES	
A.	The IVAM System	IV-1
B.	System Timing and Sizing	IV-5
C.	AFOS Interface	IV-8
D.	Development System Procurement	IV-15
E.	Prototype Element Generator Subsystem Configuration	IV-16
F.	Comparison of the Two Approaches: Communication Bus vs. Crosspoint	IV-28
G.	Arguments by Configuration	IV-30
H.	Hardware Procurement	IV-32
I.	IVAM Development System	IV-39
V.	OUTPUT DISTRIBUTION STUDY	
A.	Broadcast Network Distribution	V-1
B.	Distribution to Cable TV and Independent Broadcast TV Stations	V-3
C.	Low Bandwidth Transmission Capacities	V-4
D.	Low Bandwidth Transmission Cehemes	V-5

VI. PRESENTATION TEST AND EVALUATION

A. Software Module Demonstration Tapes	VI-1
B. Picture Quality Studies	VI-1
C. Color Evaluation Tests	VI-5
D. Other Picture Quality Considerations	VI-6
E. Summary Color Rules for IVAM	VI-7
F. AMS Advisory Panel	VI-8

VII. FUTURE IVAM EFFORT

A. Introduction	VII-1
B. Software Development	VII-2
C. Image Generation Module Development	VII-2
D. Physical Operating System	VII-5
E. Hardware Development	VII-7
F. Output Distribution Study	VII-17
G. Presentation Content Studies	VII-18
H. Presentation Content Test and Evaluation	VII-18
I. Final Program Development	VII-19

APPENDIX A

IVAM Computer Bid Request	A-1
---------------------------	-----

APPENDIX B

Color Evaluation Results	B-1
--------------------------	-----

I. INTRODUCTION

A. SCOPE AND ORGANIZATION OF THIS REPORT

This is the second annual progress report on the development program, "Innovative Video Applications in Meteorology (IVAM)". This report is submitted to the National Oceanic and Atmospheric Administration (NOAA) in partial fulfillment of Task H, Article I of Contract No. 5-35156. The report presents a comprehensive review of the work performed under the contract from 1 May 1975 to 1 May 1976 and describes the work to be undertaken to complete the IVAM program objectives. The effort reported upon was performed by members of the staff of the Space Science and Engineering Center and other members of the University of Wisconsin-Madison campus.

A brief statement of IVAM program objectives and background material, including a summation of the first year's effort, is included in this introduction. Readers not familiar with IVAM should read this section to provide the context necessary to understand the rest of this report.

This report is assembled in the same order as the previous progress report to make it easier for those interested in particular aspects of the program to locate pertinent materials. The five major tasks of the program determine the major sections of the report. The objectives of each of these tasks are reviewed in the background summary.

B. BACKGROUND AND OBJECTIVES

The purpose of the IVAM program is stated succinctly in the contract statement of work and is quoted here:

The Problem

The National Oceanic and Atmospheric Administration would achieve a significant gain in keeping the public advised of changing weather conditions if effective, automated methods of delivering video segments of weather information to TV outlets could be devised. Although this is known to be a technologically possible accomplishment, the methods needed to make it economically feasible have not been developed.

Solution

The National Weather Service and the National Environmental Satellite Service jointly undertake the funding of a study to develop a state-of-the-art capability to efficiently deliver quality video presentations of weather information to the public at acceptable cost.

Study Objective

This study will be directed toward the development of formats and techniques designed to maximize the effectiveness of the TV presentation of weather information to the public. Program content and organization will be addressed, as well as the conceptual design of the communications methods and systems that would allow the presentations to be economically delivered from NOAA sources to various redistribution terminals. The objectives of the study are to develop techniques and formats that are characterized by:

- High information content in an interesting and understandable presentation.
- A maximum employment of automated presentation formatting, in order to keep staffing levels reasonable.
- The maximum utilization of existing or planned NOAA facilities, methodologies, and communications systems.
- Modest NOAA implementation costs and modest-to-low cost for media, public, or private acquisition.
- Quality and utility coefficients that generate media and public enthusiasm and demand.

The IVAM program is organized in five major tasks described below. The correspondence of the five major tasks to the ten tasks specified in the contract work statement is indicated by the lettered task designation in parenthesis which refer to the work statement.

Task 1 - Software Development (Tasks E, J)

This is the major development effort in the IVAM program. If the objectives as described by NOAA above are to be achieved, the software system must be as efficient and as flexible as the state-of-the-art allows. Early in the program we conducted a series of very thorough, and very critical reviews of several possible routes we could have pursued. We chose to start at the very beginning, and to defer any actual software development until we

had produced a sound software system concept which we were confident could support a development program capable of meeting the NOAA objectives. During the first nine months we achieved our initial goal and the software system concept document was submitted with the 1 May 1976 report. The software development task was about 30% completed a year ago with the completion of the concept document and the existing McIDAS software which we have drawn upon for specific purposes.

The software design philosophy which has been adopted is based on the following considerations and decisions:

- 1) The system to be developed is a production system. The output is a large number of graphic images, properly sequenced, meeting high aesthetic standards, complying with NTSC standards, and coming off the line at 30 frames per second. This is the first time such a system has been attempted.

- 2) The system must be automatic. To meet output requirements it must work at high speed and must not depend upon human interaction. At the same time, full system control must be available to the WSFO forecaster.

- 3) Software will be organized to operate in a basic net structure to provide an efficient, fast response operating system.

- 4) Software will be completely modular with standard module to module and module to net interfaces to decrease development time and to permit modification at low cost. This approach will trade memory size for software and maintenance economy.

- 5) A multi-processor, multi-memory hardware design will be used to implement the system to obtain least cost in hardware, low maintenance costs, and high processing rates.

- 6) Maximum integration with AFOS will be a principal objective to eliminate function redundancy and hold implementation and operation costs to the minimum.

Task 2 - Presentation Content Studies (Tasks A, B)

Starting with an extensive review of previous work at SSEC and in the literature, users' needs for weather information were assembled and tabulated. This voluminous data set was consolidated through a series of carefully defined steps to a tabulation of user's needs listed by parameters which can be used as decision bases by the IVAM control processor. These parameters are season, time of day, weather situation, TV medium, emergency status, past and present or predicted, and spacial scale. The next step, which was practically completed the past year, has been to generate the minimum set of presentation segments which are uniquely identified by the seven parameters and which meet all important users' needs. These segment specifications are being translated into video storyboards to define system performance requirements.

At present there are 240 segments in the set and this number is not expected to change greatly.

Task 3 - Hardware Concept Studies (Tasks E, F)

Hardware definition was purposely deferred until the system and software concepts were well developed. We view hardware as just the means of implementing IVAM and have tried to avoid having hardware decisions determine system performance. As we proceeded to purchase the equipment for the prototype of the first half of the system the validity of this approach was fully confirmed. The problem we faced was not to find a set of equipment capable of the IVAM task but to select from among many alternatives. Thus we were able to optimize the equipment selection for low cost, maintainability, flexibility, suitability of system software, etc.

To date only the hardware for the front end of the IVAM prototype has been procured. Specification and procurement of the remainder of the system is the major task for the last phase of the program in FY-77.

Concepts for distribution of the IVAM products to broadcast and cable TV stations were developed during the first phase of the program and were reported last year. Briefly, it appears feasible to distribute presentation segments via network facilities during the half-hourly station break periods when the network is normally "black". Further inquiries during the past year have confirmed the feasibility of this concept. For cable distributors the plan is not as easy to define since each cable company presents a different set of circumstances. During the past year we have confirmed the unanimous enthusiasm of cable system operators for the IVAM product. Operators of systems located near WSFO's to be equipped with IVAM foresaw not significant difficulty in obtaining the output. Those located at greater distances expressed interest in the possibility of obtaining an automatically updated presentation via telephone lines. Possible designs for such a capability are discussed in this report.

Task 4 - Program Test and Evaluation (Task C)

Last year we produced two test video tapes, one of a possible IVAM presentation for broadcast stations and one for cable outlets. The tapes were made from films of graphics designed to simulate IVAM images. Technically the tapes differed from and were inferior to the expected IVAM product because of the limitations of filmed graphics. Nevertheless, the tapes have proven to be of great value in demonstrating IVAM and in obtaining answers to important system design questions. Last year we anticipated making several more tapes to be used to test specific IVAM system design alternatives. However, we have found that at this stage in the program it is more productive to consult experts than to test the public.

The American Meteorological Society IVAM Advisory Panel was organized and met in February, 1976. Five of the nation's top TV weathercasters make

up the panel. This group reinforced our previous finding of enthusiastic acceptance of IVAM by TV weathercasters, the station management and the networks. While much of the first meeting was spent in introducing IVAM and discussing program details, the panel members recommended several modifications to improve the IVAM product and to simplify routine operations. These are discussed in detail in this report.

Test films have been made during the past year to investigate difficult graphics transitions and timing problems for which we were unable to find expert advice. One series is intended to explore the effectiveness of different special effects in alerting viewers to emergency situations. This problem is important at this time since the equipment and software required for special effects must be specified soon. The second film explores the method of transitioning from a series of satellite images into graphics depicting the future. Again, equipment designs depend upon the method chosen to make this transition.

Task 5 - Final Program Development (Tasks D, I)

This task will start only after NOAA is satisfied that the implementation of IVAM is feasible, and that when implemented, IVAM will meet the NOAA objectives. This is the wrap-up effort when software development is stopped, configuration control is imposed, detailed interfacing with AFOS is accomplished, hardware specifications are completed, final documentation is prepared, etc.

C. DELAY OF FISCAL YEAR 1976 EFFORT

The initial funding increment for the IVAM program covered the period 1 August 1974 to 31 July 1975. At the end of that period approximately \$30,000 remained unspent and was reserved to cover part of the costs of the first hardware buy. The second increment of funding was to cover the period

1 August 1975 to 31 July 1976, but the contract amendment was not received by the University until 17 November 1975. Also, authority to use the remaining \$30,000 for capital equipment purchases was not received until 17 November 1975. During this period from 1 August 1975 to 17 November 1975, while the contract was actually unfunded, we continued work on IVAM but at a reduced level, and we were unable to proceed with hardware procurement until we received approval to spend the \$30,000 on capital equipment or until additional funds were actually in hand. During this period we obtained and evaluated bids, but did not release purchase orders until 17 November 1975. This date represented an actual delay of about five months compared to the program schedule presented in May 1975, because we had anticipated proceeding in June 1975 to purchase the PDP-11/40 computer with the \$30,000 on hand at that time. The consequences of this delay have been as follows:

- 1) We had a much longer time to evaluate alternative hardware configurations. The basic multi-processor approach was reaffirmed and developed in greater detail. Some significant improvements in the hardware design resulted. The superiority of the PDP system for this application at this time was firmly established.

- 2) We were forced to restructure our software development program to minimize overall program delay. We decided to adopt the RSX-11M system software as the starting point for the prototype IVAM system rather than to proceed with development of the specialized net structure originally planned. This decision allowed us to proceed with development of applications modules on McIDAS in FORTRAN using the specification of the RSX-11M system for interface definition. The downstream consequences will be a delay in implementing the optimum system software design for IVAM.

- 3) We missed some short delivery schedule opportunities, and have had actual hardware deliveries delayed well beyond the 107 delays in fund-

ing authority. Most of the computer hardware was delivered in March 1976. Consequently transfer of software modules to the IVAM equipment from McIDAS is just starting.

4) We have been unable to generate or display graphics on IVAM equipment to IVAM standards. Testing of graphics modules has been on McIDAS equipment in which aesthetics considerations have been secondary. The video tape which accompanies this report is a demonstration of the graphics generation software modules developed to date, but the aesthetic qualities of the images are determined, and severely limited, by the McIDAS equipment.

5) The overall program has probably slipped four to six months, but may have benefitted from the more extensive hardware design review.

D. THE NEED FOR IVAM

IVAM grew out of our earlier studies of people's needs for weather information and how well these needs are met which led directly to the concept of NOWCASTING. Five years ago we were persuaded that people wanted and needed accurate detailed information about the weather in their locality right now and what it would be during the next six hours. Today we are more convinced than ever that that need is real and, despite some encouraging signs of increasing official awareness, that it is still underappreciated.

We were happy to read a paper which reported the endorsement of the NOWCASTING concept by the NWS Southern Region WSFO's. The paper was doubly refreshing in its clear presentation of what NOWCASTING is and why it is important to the National Weather Service. It said:

"What is it? -- Nowcasting is a combination of real time weather reporting with short term forecasting -- from time now out to about six hours in advance -- with emphasis on weather that is likely to affect the public both in terms of unusual and critical conditions and in terms of immediate occurrence.

"Should it be done? -- WSFO analyses submitted late this spring provided an interesting case of Management by Objective. The consensus of these reports clearly identified 'the communication

of useful weather information on a current and immediate basis' as a primary objective of our organization. This is nowcasting. Yes, we should go ahead with this concept.

"How can it be done? -- Although most WSFOs are now using some nowcasting features and the consensus of WSFOs was to move on, there was a diversity of problems outlined and a limited experiment is proposed rather than proceeding at all offices at this time. Effectiveness of a nowcasting program hangs on all the links of a chain of facilities -- a means of gathering pertinent information quickly, techniques for translating this into a release, and a means for disseminating the release to the public quickly...."

The paper goes on to outline an experimental implementation of NOWCASTING via the National Weather Radio facilities - an excellent plan in our judgement. We would like to repeat one sentence from the paper:

"Effectiveness of a Nowcasting program hangs on all the links of a chain of facilities -- a means of gathering pertinent information quickly, techniques for translating this into a release, and a means for disseminating the release to the public quickly."

And that is what IVAM is all about.

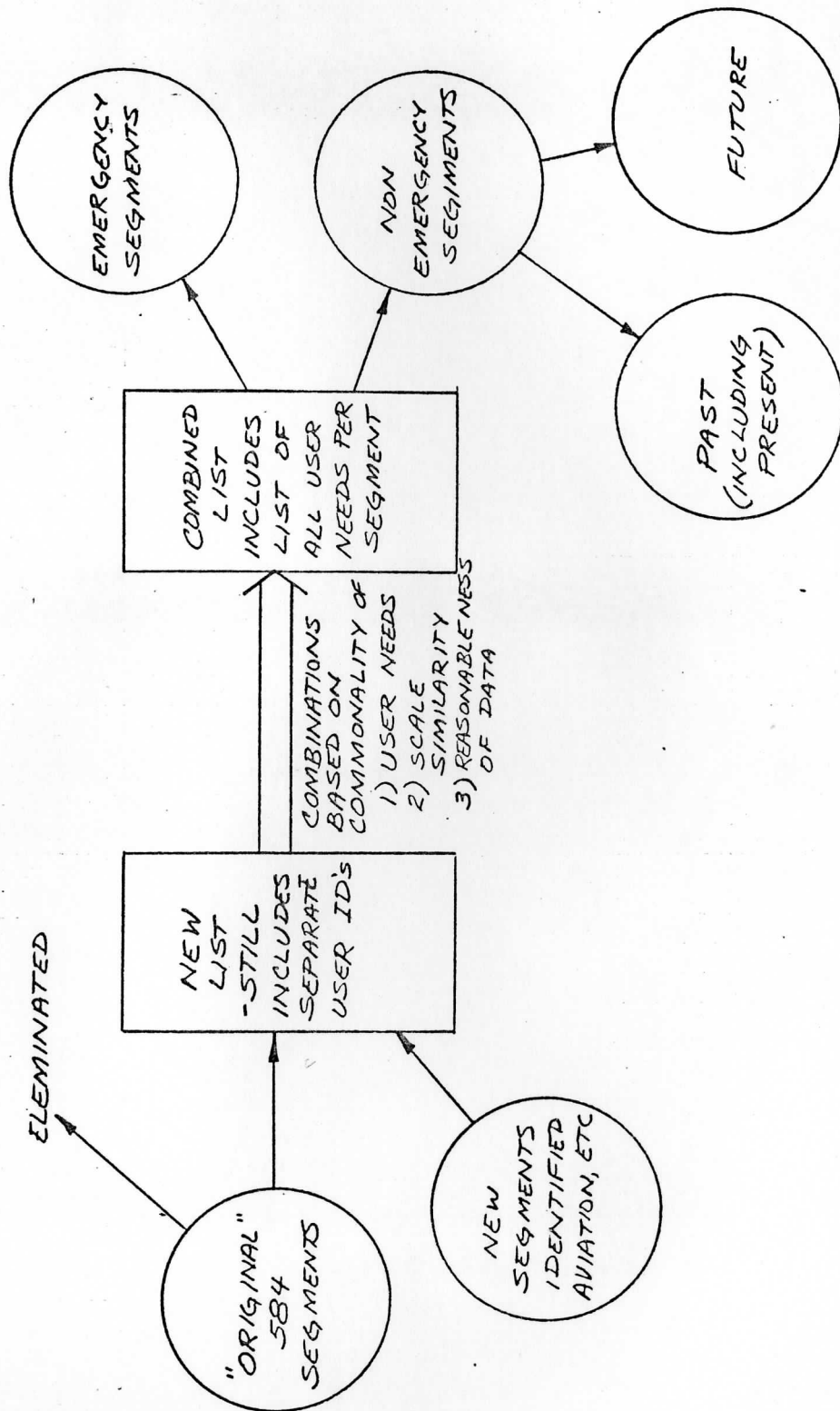
II. PRESENTATION CONTENT STUDY

This year the presentation content study has refined and reduced the segment list from the 584 listed a year ago to 240. The needs of the weather user, identified in the 584 original segments, have been carried forth through the reduction steps to identify the contents of the final segments. The segments are now identified by the following parameters which are decision criteria for the IVAM control processor:

- 1) Emergency status
- 2) Season
- 3) Time of day of presentation
- 4) Weather situation
- 5) TV medium (broadcast TV or cable TV)
- 6) Weather parameter
- 7) Time/space scale
- 8) Past/present or predicted information

Reduction of the segments was performed by holding four of five parameters (numbers 2, 3, 5, 6, and 7 above) constant and combining within the fifth. This was done for each parameter in turn, until a minimum set of segments was reached. The practical limit of time allowed on broadcast TV was also considered in limiting the time and range of subjects which can be addressed on that medium. A set of about 100 segments was reached, but did not distinguish between past and future, or emergency and non-emergency. The user needs identified in the previous year's work were carried through the segment reductions and assigned to the new combined segments. A diagram of the steps involved is shown in Figure II-1.

The next step was to expand the segment list into two categories, emergency and non-emergency. The assignment of emergency status is based on current practice for watch and warning messages, and storyboarding of those



FLOW DIAGRAM OF SEGMENT COMBINING STEPS
FIGURE II - 1

segments will include recommendations from various disaster preparedness and disaster survey reports. Storyboarding of the emergency segments will be done in close cooperation with the NWS to insure that the IVAM warning messages will be most effective and consistent with current practice.

The non-emergency segments were further split into past/present segments and predictive (or future) segments. The data bases used for past and present weather are different than the bases available for prediction. Present is considered just the most recent past data. The past/present data are based on actual measurement, whereas the future data must be based on extrapolated or numerically predicted values or graphics generated by NPC. These different bases have the result of producing data to IVAM through the AFOS system that are much different in timing, time interval, and format. These basic differences in source data necessitate a different storyboard for the past/present and the future segments.

Since many weather shows combine past/present and future information, the past/present and future segments for IVAM will be storyboarded with start and end transitions that will allow continuous past-to-forecast presentation for a particular weather parameter.

Current activity is that of assembling the header sheet for each segment. Four examples of header sheets are shown in Figures II-2 to II-5. The header sheet identifies all of the segment parameters, references the source of input data, and references the user needs for storyboarding.

The header sheet provides the requirements for storyboarding. Each segment has its own storyboard which in turn becomes the production specification for the software coding. The storyboard is a complete specification of the visuals, which includes: image styles, transition between images, order of sequence, animation or still frames, annotation style and location, color constraints, special effects (scintillation, etc.), overall timing, etc.

SEGMENT HEADER

SEGMENT NUMBER _____

EMERGENCY _____ NON EMERGENCY X

MEDIUM Cable _____ Broadcast _____ Both X

SEASON Winter X Spring X Summer X Fall X

TIME OF DAY Midnight - 6am _____ 6am - 4pm X 4pm - Midnight X All _____

WEATHER SITUATIONS 1) After widespread storm in U.S. _____

2) After excessive precipitation _____

or during major flooding episodes _____

WEATHER PARAMETER(S) Precipitation _____

TIME REFERENCE Past (incl. present) X 12-72 Prediction _____

SCALE National X Regional _____ State _____ Local _____

COMMENTS ON DESIGNATORS (i.e., use in _____ area only, etc) _____

MAJOR USER OF SEGMENT: (reference previous IVAM study) General Public (GPA)

WHY CAN'T SEGMENT BE COMBINED WITH ANOTHER These are past weather "results," not combinable with other weather parameters - unique emphasis.

PURPOSE OF SEGMENT (WHAT'S TO BE CONVEYED) Areas where precipitation exceeded a boundary threshold in last 72 hours. (See below.)

FORMAT OF SEGMENT (scale, background, foreground, duration, etc) National map with state outlines. Contours of precipitation (not to exceed 4 contours) beginning at .25", then .50", 1.0", then year contour for maximum received, with annotation of that total within contour. Within contour to be shaded - lightest shading in .25 to .50 contour etc. Probably add areas one at a time at 5 sec. intervals - start with least.

Should be provision to overlay hatching in red to indicate flooding areas or states.

Duration: 20 sec. plus 10 sec. if flooding areas are overlayed.

DATA SOURCE (if non-existent, describe probable source) Fax precip. maps, teletype data; flood forecast and report on rawarc circuit and weather wire.

FIGURE II-2

SEGMENT HEADER

SEGMENT NUMBER _____

EMERGENCY _____ NON EMERGENCY X

MEDIUM Cable _____ Broadcast _____ Both X

SEASON Winter X Spring X Summer _____ Fall X

TIME OF DAY Midnight - 6am _____ 6am - 4pm X 4pm - Midnight X All _____

WEATHER SITUATIONS Days when mean temperature
falls below 65° - optional use
segment (not used every day)

WEATHER PARAMETER(S) Degree Days (Heating)

TIME REFERENCE Past (incl. present) _____ Prediction 12-72 HRS

SCALE National _____ Regional _____ State _____ Local X

COMMENTS ON DESIGNATORS (i.e., use in _____ area only, etc) _____
Probably not fed in S.E. or S.W. states

MAJOR USER OF SEGMENT: (reference previous IVAM study) General public, and/or
dealers.

WHY CAN'T SEGMENT BE COMBINED WITH ANOTHER Unique parameter, and too complex for
combination.

PURPOSE OF SEGMENT (WHAT'S TO BE CONVEYED) Expected degree day totals next 1, 2, 3
days.

FORMAT OF SEGMENT (scale, background, foreground, duration, etc) Bar graph - days
at bottom, degree days vertical. Space, bars and shade in color. Overlay each bar
with large numerals indicating number of D-Days on that day. Perhaps dissolve in
each day separately (1 + 2 + 3)

DATA SOURCE (if non-existent, describe probable source) fax forecast of temperatures;
NWS zones forecasts and extended outlooks. Computer to perform calculation required.

SEGMENT HEADER

SEGMENT NUMBER _____

EMERGENCY X NON EMERGENCY _____

MEDIUM Cable _____ Broadcast X Both _____

SEASON Winter _____ Spring X Summer X Fall X

TIME OF DAY Midnight - 6am _____ 6am - 4pm _____ 4pm - Midnight _____ All X

WEATHER SITUATIONS Expected or actual temps. below 32° - all such days between
15 days before normal last frost and until hard freeze occurs in
the fall.

WEATHER PARAMETER(S) Frost and Freeze (28° and 32° contours)

TIME REFERENCE Past (incl. present) _____ Prediction 1-6 HRS

SCALE National _____ Regional _____ State X Local _____

COMMENTS ON DESIGNATORS (i.e., use in _____ area only, etc) _____

MAJOR USER OF SEGMENT: (reference previous IVAM study) Agriculture

WHY CAN'T SEGMENT BE COMBINED WITH ANOTHER Emergency segment - impact required. Large
dollar losses possible from frost.

PURPOSE OF SEGMENT (WHAT'S TO BE CONVEYED) The areas of state or region which are
currently, or will be within time frame, below freezing.

FORMAT OF SEGMENT (scale, background, foreground, duration, etc) State map with tempera-
ture contours every 4° beginning at 32° and extending lower. Higher temperatures are
extraneous and should not be recorded. Area between 32° and 28° contour to be
colored yellow; area less than 28° to be colored red. Possibly overlay wind arrow
(with speed included) but detailed wind information is not desirable - too complex.

DATA SOURCE (if non-existent, describe probable source) Local and zone forecasts on
weather wire; fax computer chart of min. temps. is secondary (not as good) info.

FIGURE II-4

SEGMENT HEADER

SEGMENT NUMBER _____

EMERGENCY X NON EMERGENCY _____

MEDIUM Cable _____ Broadcast _____ Both X

SEASON Winter X Spring X Summer X Fall X

TIME OF DAY Midnight - 6am _____ 6am - 4pm _____ 4pm - Midnight _____ All X

WEATHER SITUATIONS Pollution concentrations exceed _____
 or are forecast to exceed alert _____
 standards. _____

WEATHER PARAMETER(S) _____

TIME REFERENCE Past (incl. present) _____ Prediction 1-6 HRS

SCALE National _____ Regional _____ State _____ Local X

COMMENTS ON DESIGNATORS (i.e., use in _____ area only, etc) _____
Primarily major urban use.

MAJOR USER OF SEGMENT: (reference previous IVAM study) GP 3; utilities

WHY CAN'T SEGMENT BE COMBINED WITH ANOTHER Warning segments must stand alone for
impact and for individual use.

PURPOSE OF SEGMENT (WHAT'S TO BE CONVEYED) _____

FORMAT OF SEGMENT (scale, background, foreground, duration, etc) City name predominantly
at bottom. Horizontal cross-section of city skyline with smoke curling in air

- Annotation: time of segment - time alert is valid 'til - phrases "avoid unnecessary activity or overexertion, especially elderly and small children"
- Annotate with outlook: "Improving by _____" or "little change"

DATA SOURCE (if non-existent, describe probable source) NOAA wire, ?

The storyboarding is the step that defines the communication between the technical field of meteorology and the non-technical public. The storyboarding must be done by personnel trained in meteorology, TV media, and communication. Since the storyboarding unavoidably involves the style of the person preparing the storyboard, the Test and Evaluation part of the program will be used to "depersonalize" the segment and to insure that the storyboards produce effective communication.

Since IVAM is to be an automatic system, the selection of segments to be used for a particular day must be done on an automatic basis as well. Most of the 240 segments available on IVAM do not apply on any particular day because of time-of-day, season, or weather situation. In other words, only a subset of the 240 segments are topical at a particular time in terms of the eight parameters listed earlier. The principal basis for deciding which segments are to be prepared is a determination of the weather situation. Each header sheet lists the weather situation for which the segment applies. To decide which segments to prepare, the control processor will reduce the total set by comparing the current situation with the segment designators, proceeding through the parameter list in the order shown. Of course, the weather situation will change during the day. The control processor will proceed through the decision tree once each hour to determine which segments to produce during the next hour. In most cases this procedure will work very well. However, the time required to transmit all of the segments needed for the network weathercasters is several hours. Care must be exercised to produce first those segments least affected by local weather change.

The set of weather situations will differ for different WSFO locations, depending upon whether flood conditions, hurricane situations, drought, high tides, etc., apply.

Weather situation selection can be made either by the forecaster keying into the system or by the automatic comparison of specified weather parameters against preset limits.

III. SOFTWARE DEVELOPMENT

A. INTRODUCTION

The first major software effort during the past year was a thorough review of the IVAM hardware and software needs. The result of this review was that the arguments for a parallel processing approach presented at the last Annual Briefing were resubstantiated. An in-depth survey was made of existing methods of configuring and controlling parallel processing systems and on that basis a development system was purchased and an implementation strategy designed. Software conventions were established to guide the interfacing of software components. Subsequently, the implementation has proceeded on those parts of the system that are machine independent while the machine dependent parts have been slowed by delays in funding and equipment delivery.

The year has been highlighted by several decisions that have important implications for the development of the project. The first of these was the decision to develop the software in a parallel processing environment, rather than to use a single processor and to assume the code could be easily transferred. The next decision was to purchase a sophisticated software package with the development hardware to provide software support during development and if possible to provide a basis for the IVAM Operating System. This decision was influenced by the identification of several radically different approaches to the Operating System. The choice will be dictated by the method of distributing the IVAM output rather than by hardware or software requirements. Therefore, the decision to purchase the RSX-11M Operating System had the further implication that the full implementation of the IVAM Operating System would be deferred until the operational environment is better defined.

We learned that a number of system structures could be merged into a single construct. The data set descriptors specified in last year's software document have been expanded in function so that they now encompass the specification of image elements, image formats, and image sequences. The descriptor structure is collapsed step by step during image generation so that at each point the descriptor is a specification of what remains to be done to complete the image - all this in a form which permits independent subtasks to be executed by separate processors.

In the balance of this report we will review the characteristics of the IVAM task and the assumptions on which our approach is based. Next we will describe the parallel processing studies that led to our choice of development system. Then the progress in three areas of implementation will be described: operating system, modules, and image formats. In each case the plan of future effort will be outlined. Finally, the interface between the software and other aspects of the system will be described. In particular, the implications of what we now know about the software for hardware, distribution, testing, and interaction with AFOS will be discussed.

1. Software Task Review

The IVAM software must produce up to 20 minutes per hour of broadcast quality video images. These images must be suitable for easy viewing by the broadcast public. The lettering must be easily seen by the whole range of viewers sitting at normal viewing distance. All image information must be equally clear on black and white as well as color sets. The IVAM images must convey weather information to a lay public rather than meteorological data to professional meteorologists.

The information must be tailored to the viewer's locale and must provide him with the most up-to-date information available. This means that the

available data must be scaled and translated to fit the range of the stations being served. Since part of the data is in the form of satellite images it is necessary to transform other images so that they can be shown in satellite projection and overlaid on the satellite image.

The IVAM software task is different from that of AFOS in several respects. First, IVAM is not concerned with the transmission of data over a limited bandwidth channel; therefore, no compaction is required. Also, IVAM is not concerned with the text of messages--only with its content.

Since the IVAM processors must produce up to 20 minutes of video per hour the amount of data to be processed would appear to be astronomical. In fact it is reduced somewhat by the fact that most frames are repeated from 6 to 60 times. Nevertheless our analyses have shown that no existing or immediately foreseen single processor will be capable of producing the amount of image data required. Thus some form of parallel processing is absolutely necessary.

To preserve this option and to ease the implementation and modification of the IVAM System, the IVAM software is being developed according to the following criteria: First, the software design will be completely modular. The image specification will allow separable, i.e. independent, processes to be executed on separate processors. To the extent possible the software will be machine, configuration, and format independent. Also the interdependencies between different parts of the system will be held to a minimum.

2. Multiple Processing

Given the fact that some form of multi-processing seemed necessary and given the acceptance of our parallel processing proposal last April, we decided to investigate the methods of multiple processing that were available and to determine what their implications would be for the IVAM software. We also

analyzed the IVAM tasks to see where these methods could be applied and what kinds of processors would be appropriate.

A number of computer configurations were examined as well as the combination of hardwired processors. The first means of classifying existing systems was as parallel processors or pipeline processors. In the first case each processor works on the same step in a procedure while in the second each processor works on a different step in the procedure while the data passes from one processor to another. The data that passes from the last processor is completed while that in each of the earlier processors is less and less complete. In the IVAM processing both parallel and serial processing are needed. An IVAM image is made up of a number of image elements. Each of these is logically independent of the others and can be processed separately, independent of the rest.

It was felt that IVAM could not commit completely to either the parallel or the serial approach. The pipeline aspects of the IVAM processing, for instance, would become less significant if hardwired processors were used. Therefore, the decision was made to consider as flexible a multiple processing configuration as possible. Thus we chose to consider only those multiple processing configurations where any task could be run on any processor. With such a configuration pipeline processes can be implemented and special purpose processors simulated.

In studying existing parallel processing systems we identified several different approaches which had clearly identifiable characteristics in both hardware and software. The first and most common would be independent processors that are simply interconnected and each have their own fully independent operating systems. These systems are sometimes motivated by a desire to net already existing processors, but perhaps more often by an unplanned

overflow process whereby the task becomes too large for one processor and another is added to take up the extra work. The additional processors are often connected by DMA channels which usually become bottlenecks even though they are often thought of as high speed channels. In these systems the software is much like standard systems and so is the performance. In fact there is usually a degradation in performance over what it would be if the whole process could be handled by one machine.

The next class of systems are those that are conceived with parallel processing in mind from the beginning. In these systems the basic problems of parallel processing such as interprocessor transfer and control are the paramount concerns of systems design. The bandwidth of the interprocessor channels determines how long one processor will have to wait for data to be transmitted from another machine. Also important is whether interprocessor transfer stops execution on both machines, whether it degrades execution by significant cycle stealing, or whether it reduces the disjoint address space of both machines.

An important design trade-off involves the hardware and software overhead incurred from interprocessor transfer versus the characteristics of the task. In some tasks where the ratio of processing to transfer is very high, i.e. where a small amount of data is processed for a long time, a slow interprocessor transfer system may be adequate. In other tasks large data sets may undergo very little processing so the input and output transfers will take much longer than the processing. A search for a single entry in a large table is an example of this kind of process. In this case either a high speed transfer or one that is transparent, i.e. does not degrade the processor, is required. In general, the higher the speed of transfer the more expensive the hardware and the greater the number of transfers the greater the software overhead.

IVAM appears to have both kinds of tasks--those in which the ratio of input to processing is very low, such as contouring or map transformations, and those where it is relatively high, such as the merging of image elements. However, the nature of the high ratio tasks is such that they can be broken into shorter steps and can be interrupted mid-process to give the system more flexibility in distributing the load they represent. This factor was important to a decision affecting module design which will be described later. Its significance is that we decided to break up the execution of large modules into smaller steps to allow the system to sense and reevaluate its needs more often.

The existing parallel processing systems have several types of operating systems which have a tremendous effect on how well the multiple processors can be utilized. In the most common case where the machines and their software are completely independent the system seeking to initiate a transfer must idle as it waits for the other processor to acknowledge its request. In some of the other systems the control structure is hierarchical with one of the processors being designated the control processor. The control processor then has the responsibility for assigning tasks to the subordinate processors which simply do what they are told. The subordinate processors will have the abbreviated operating systems required to control their local environments but make no decisions for the system as a whole.

Another alternative will be described in some detail because of its extraordinary departure from computing traditions. This is the system designed by Bolt, Barenak, and Newman to perform the packet switching function at the nodes of the ARPA Net. In hardware terms it contains three kinds of buses: processor buses, memory buses, and I/O buses (Figure III-1). A processor bus typically has two processors each with its own dedicated memory. The control

Bolt Beranek and Newman

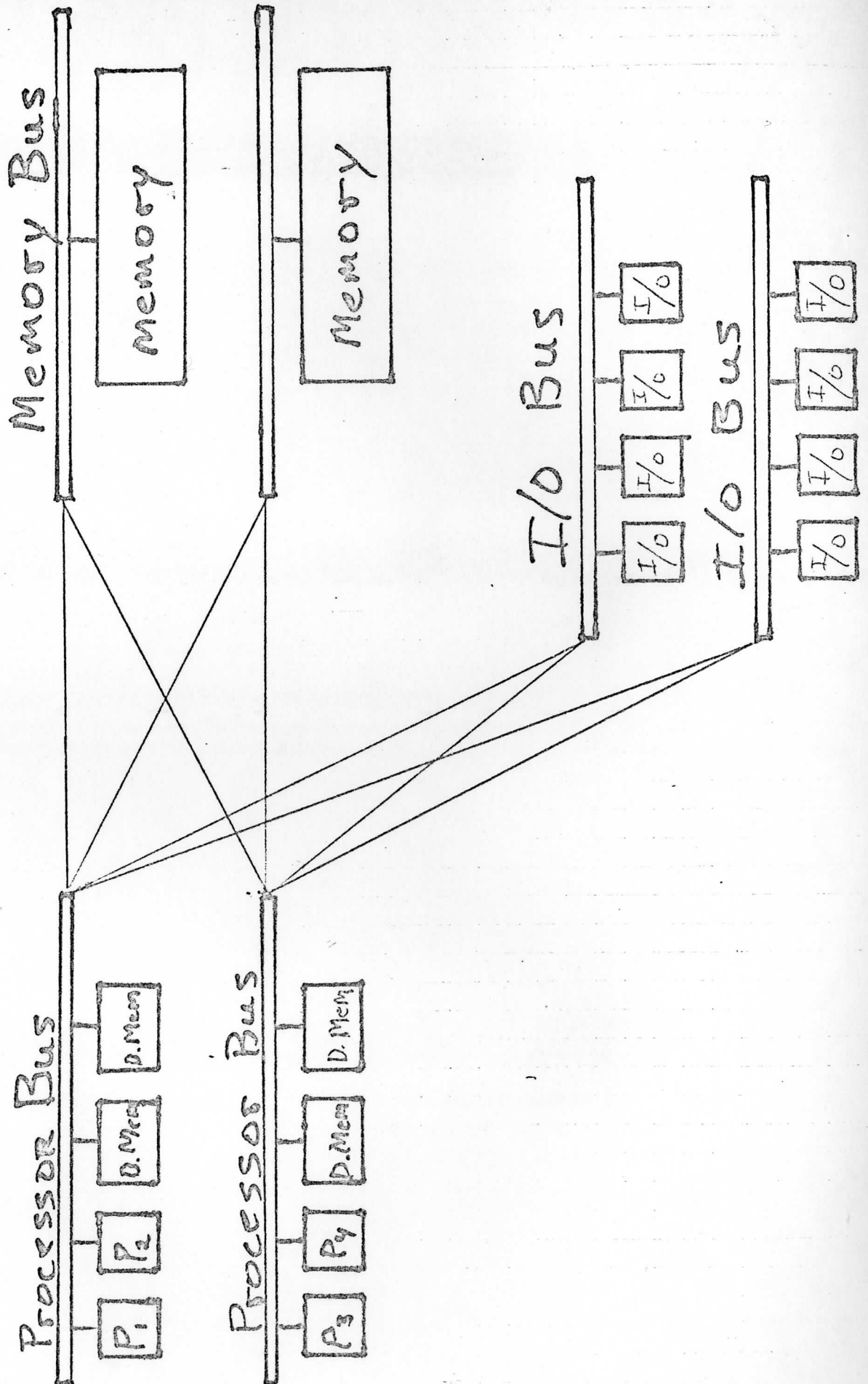


Figure III-1

of the processor is exercised by a special piece of hardware called a Bus Arbiter which decides on a first-come-first-serve basis who gets control of the bus. This function is usually part of the processor hardware but here the bus is a separate entity which views the processors as users. Any processor bus can connect to any memory bus or any I/O bus, thus giving each processor the potential of controlling any device in the system. However, the attachment of any bus to any other bus is made only for the duration of one transaction. There is no concept of assignment of resources in the usual sense. At the hardware level this system has the advantage that any processor can accomplish any function since all processors are identical and capable of controlling the system. This means that the function of any one processor can be assumed by the others in the event of its failure.

It is at the software level, however, that this system is truly unique. Not only are the processors all identical from a hardware point of view, they also contain identical software in their dedicated memories. All input, in-process, and output data are contained in the shared memory on the memory buses. Therefore at any point in time each processor is a completely general entity from both a hardware and a software point of view. It can be assigned to any task.

It is natural to wonder here who is in charge and how tasks get assigned. The operating system is never assigned permanently to a processor. Instead, it consists of a series of tasks which are mixed in with the other tasks in the task list. When a processor becomes free it consults the task queue and adopts the highest priority task which may or may not be part of the operating system. Note that there is no operating system in the normal sense. Neither are there any interrupts. The assignment of a task is accomplished when a processor consults the PID, a pseudo-interrupt device, which maintains the task queues.

In the course of a single machine cycle the processor has requested and been assigned a task and the task list has been updated.

To make this system work, an extremely severe programming convention has been adopted. All programming is done in units of code called "strips" which are coded in machine language and take no more than 350 microseconds to execute. This means no processor is ever assigned to any task for more than 350 microseconds. It also means that the strips are coded in such a way that they are restored to a processor independent form every 350 microseconds.

The operation of this system has another interesting feature. Maintenance programs are run as a regular part of the system load. All processors are constantly being checked by the others. When the reliability of any of the processors or other devices becomes suspect it can be switched off by any of the other processors.

The ARPA Net system has several characteristics. First it is the solution to a well-understood problem which had several generations of solution in the field. Second it is a homogeneous problem in that all tasks are of the data shuffling rather than computation variety. Therefore, it is easy to make every processor identical. In a mathematical system the addition of floating point hardware to each processor would up the cost of the processors by a large factor. Third, while the task mix is completely homogeneous and the system thinks only in terms of 350 microseconds strips, the arrival of requests is completely unpredictable. It depends upon the arrival of packets from other locations. Therefore the system can only decide what to do next. It cannot lay out its work ahead of time.

A great deal was learned from the Bolt, Barenak, and Newman system. The idea of a resource pool and identical processors was considered very interesting as was the demonstrated ability of slow cheap processors to achieve high

throughput. Also the assertion that successful parallel processing requires a restructuring of the programs appears to be a general truth. At the same time constraining all code to 350 microseconds strips seems to be an unreasonable burden during the IVAM development processor, for the first task is to demonstrate a satisfactory video product and then to specify the production system. Also it appears that the IVAM task may be fundamentally different from packet switching in that the order of requests can be known ahead of time.

3. Software Breakdown

Armed with the insights gained from others' experience we refined our concepts of the IVAM software and identified the tools required to develop it so that our software would be consistent with any parallel processing implementation. To this end it was decided that the IVAM software should be divided into three types: the Semantic System, the Physical Operating System, and the Modules. The Semantic System is concerned with representing the logic of image specification and generation. It is machine independent in that it can be bootstrapped from a small amount of machine code. The Physical Operating System is concerned with access to data sets and allocation of resources. While the Semantic System knows that certain data sets exist, it is the responsibility of the Physical Operating System to keep track of where they are stored. Since the Physical Operating System deals with actual devices, much of it is machine and configuration dependent. However, since its design is modular, only certain modules would have to be changed if the configuration were changed. Finally, the modules themselves do the work of image generation. While many of these modules can be coded in machine independent FORTRAN, it was recognized that some of them would be much more efficient if written in machine code or otherwise made machine dependent. Therefore we decided that machine dependence should be localized but not avoided completely.

B. IVAM OPERATING SYSTEM

Much of the foregoing discussion has assumed the need for an IVAM Operating System. We verified this need by identifying systems level tasks which have to be performed under explicit IVAM control. Only some of these are peculiar to the multiple processor environment; most are required by the nature of the IVAM task.

1. Why Needed

The multiple processor environment requires the assignment of a task to a processor, the transfer of code and data sets among processors, and the communication of completion and error conditions so that the system can allocate and deallocate resources.

The IVAM task itself requires the maintenance of partially processed data store which, while it is derived from the AFOS data base, it is very different in format and purpose. IVAM must constantly update this data store as products arrive; alter its processing if a product has not yet arrived; and revise its output if the product is definitely not available. It must schedule its processing according to real world clock time when TV broadcasts occur and when scheduled inputs arrive. It must also work in video time to assure that the video chain is kept full and output is maintained. Simultaneously, the system must control external devices such as disks, video tape recorders, and color enhancements. The system must constantly refer to the dependency queue that specifies the output to determine what tasks should be executed next. Finally, the system must be capable of dynamic memory allocation; fixed partition operation is not enough.

Scheduling, device control, resource allocation, data management, and device handling are the functions of an Operating System. All must be under IVAM control. Therefore IVAM must have an operating system.

2. Specification for Development System

As the foregoing discussion suggests, the IVAM software designers felt that multiple processing was a significant aspect of the IVAM software problem and much more than an implementation detail. Therefore the decision was made to implement IVAM in a multiple processing environment from the start. Again this was for software more than hardware reasons. While one can try to test multiple processing software in a single processor environment, its ability to run on multiple processors will be suspect because the multiple processing compatibility has not been proved. While we decided that we should develop the software in a multiple processor system we did not necessarily feel that a great deal of effort or money should go into optimum hardware design or execution performance until the basic system was running. It was felt that in general it is an error to try to optimize the performance of a system until it has been verified that the task it performs is the one that actually exists in the field. Also the costs of hardware are in such a state of flux that to optimize a system that is going to be produced several years hence in terms of today's economics represents a false economy that ultimately wastes money.

For the purposes of the bid specification a configuration was chosen that could be implemented with off-the-shelf devices and to some extent governed by a standard operating system. While not optimum, this configuration was conceptually simple. Its main drawback was the existence of a single limited bandwidth channel joining all the processors. It seemed that this channel might saturate and would definitely degrade as contention increased. In fact contention would degrade performance even when the communication channel was only 30% loaded. Nevertheless it was felt that such a system was a good place to start. We also invited the bidders to propose alternative configurations and compatible software as they were able.

In the specification the following criteria were established: the processors were required to have the instructions and addressing modes desirable for systems programming. Also since the current IVAM effort is in software development rather than performance optimization this system was required to have at least one processor equipped with the software tools needed for that type of effort. It was desirable but not absolutely required that the system come with a multiple processor operating system.

Additional hardware devices were specified as aids in software development. An alphanumeric terminal was required for interactive debugging and program editing. A dual floppy disk system was included as the primary method of off-line program input and storage to serve much the same function as a card reader. Finally, a data tablet was identified as a necessary tool for graphic input and generation of complex sets of test data.

The bids received contained a rather limited set of software options. All included single processor operating systems. DEC, MODCOMP, and VARIAN offered some types of multiple processor operating systems. In each case these processors were netted rather than truly integrated and medium speed interfaces were assumed to be adequate for interprocessor communication. In each case the operating system on one machine could initiate the execution of a task on another processor. In the MODCOMP and VARIAN systems the configurations were so awkward and the software protocols so tortured that it was difficult to see how any performance advantage could be realized from the systems proposed. The DECNET software proposed by DEC had some of the same failings but to a lesser degree. DEC did, however, present a wide variety of interface options that presented tantalizing sets of software possibilities including bus links, bus windows, dual port memory, and a DMA bus. The identical Bolt, Barenak, and Newman and Lockheed bids proposed a running multiple processor system

which unfortunately was almost completely unprogrammable. There was no FORTRAN compiler, no editor, and no debugging system, just an assembler. There also appeared to be no large assortment of device and interface options much as would be needed to outfit the development system. Therefore it was felt that while BBN bid a conceptually provocative configuration for a production system it was not appropriate as a development system both because of the lack of software and the lack of protection features.

None of the systems bid were completely satisfactory as proposed. Therefore the decision was made to choose a processor and a configuration and then build or buy the interfacing required to tie several such processors together. The PDP-11/40 computer was chosen for the reasons stated earlier and because it allowed the purchase of compatible subordinate processors at very low prices. The PDP-11/40 was chosen to be the program development machine and was then augmented with memory and support devices. Nova and Keronix were not chosen primarily because they did not bid even though they received the RFP, and our evaluations did not rank the NOVA high in programmability. With very limited funds available we were extremely sensitive to prices, and the DEC bid offered the best bargain by a large margin.

The configuration chosen was a crosspoint memory system which will be described in hardware terms in a later section. Its significance lies mainly in its software implications. First, the crosspoint scheme provides an extremely high rate of interprocessor transfer. Second, while it is not as general or elegant as the Bolt, Barenak, and Newman system, it does have the advantage that the crosspoint memory modules provide a form of memory protection. Since a given crosspoint module can be addressed by only one processor at a time, there is no way that a program running on one machine can destroy the data or code being used by another machine. Not only are all transactions

between processors initiated and controlled by the Operating System on the Control Processor, they are only possible, for hardware reasons, from the Control Processor. This feature eliminates a whole class of bugs that could exist in many of the other configurations.

3. Purchase of RSX-11M

The decision to also purchase the RSX-11M Operating System from DEC was a very important one and it was based on several significant considerations. The earlier model of the IVAM development was to first create a series of powerful systems and then to use their power and flexibility to develop a variety of video products. However, the long delay in funding approval and a realistic appraisal of expected versus promised delivery suggested that we might have to compromise our original plan. Originally we were confident that we could produce a truly superior product whose performance exceeded every requirement and whose flexibility anticipated many of the problems of field implementation and growth that are usually overlooked during the development phase. However, as we saw time elapse and remaining time in the program dwindle we came to feel that we should press to implement a solid video result as soon as possible, postponing some of the system flexibility and performance until the first goal was accomplished. If the preliminary implementation is done in a way that is consistent with the later development this approach does not have to prejudice the final result. It does make the video solution harder to achieve and means that some of the features that are desirable but not required may be cut from the end of the development period. It was therefore decided that some of the operating system development should be postponed and that we should purchase an operating system under which to run during early development and then see if that software could be used in whole or in part to do the final IVAM job.

We therefore decided to purchase the RSX-11M Operating System from DEC. This is a fixed partition multitasking operating system that operates from disk and comes complete with an assortment of program development tools. While it definitely does not do the IVAM production job as currently defined, it is a very powerful development tool and perhaps later can be modified to serve that function or some of its code cannibalized for use in the final IVAM system.

4. Development Process

Given the procurement decisions and the software reasoning that led to them, we then altered our implementation plan to reflect the new priorities. These decisions and activities were again divided into the three major areas of Physical Operating System, Modules, and Semantic System. We had to determine just how we would use the RSX-11M software to do what we wanted and then define our interface to it. Then we had to define the interface between the IVAM Operating System and the IVAM modules. Finally, we had to define the exact interface between the Semantic System and the Physical Operating System. The order of decisions and the order of implementation are the same except that the full implementation of the IVAM Physical Operating System will be addressed only after the modules and Semantic System exist.

a. Operating System

The first task in the operating system area was to learn as much about the RSX-11M as possible before the PDP-11/40 arrived. This has been a very time-consuming process and has been hampered by the fact that the ultimate authority on an operating system is not its documentation but its actual performance. There is no way for manuals and listings to communicate the full subtleties of complex software. Since the hardware has arrived only recently, we are only now able to test our understanding of its capability and amenability to our control.

Our investigation has produced a number of important observations but has not yet resolved the basic question about whether RSX-11M can do our whole job. That determination is about two or three months off. However, we can see what has to be done to make it serve our immediate purpose and outline a procedure whereby our interface to RSX-11M is the same as that which will be required to our own operating system should RSX-11M prove to be inadequate.

The most obvious fault in RSX-11M, and in most commercially available operating systems, is that while the machine has an operating system, the user does not. There are many functions which the operating system provides itself but which it does not gracefully provide the user's programs. Since the goal of the IVAM/RSX-11M interface is to give IVAM complete knowledge and control of the machine and its operating system, and RSX-11M has no concept of a task superior to itself, this goal may be difficult to attain. For example, RSX-11M allocates space within partitions but does not provide a task with the means to do its own suballocation. It provides file structures but these are biased in favor of sequential files organized in records. There is no substructure definable through the operating system. IVAM must think in terms of data sets and module code, not files; RSX-11M wants to execute programs, not act as an executive for a user system. IVAM is a system and its linkage between modules is not fixed by a compilation, link edit, or collection process. None of these problems is intractable but some require a certain amount of testing before we can confirm that our proposed solutions will work.

IVAM adds a few extra requirements to those that RSX-11M was designed to handle. IVAM must be capable of coordinating the scheduling and execution of several processors. This means that it must be able to send control messages between the processors. It must be able to transfer data from one machine to

another, and it must be able to set up and initiate a process on any of the processors. It must also do allocation and scheduling for the system as a whole. The IVAM system must also be able to reference a data set in a way that is consistent regardless of whether the data set is in one of the processor memories or on the disk. The references to these data sets must also allow the creation of complex structures in addition to simple files. It must also be possible for the development programmer to deal with the file structure in terms that are convenient for him rather than using abbreviated file identifiers.

Another need arises from the nature of the IVAM task. Most operating systems are designed to regulate the allocation of resources in what is conceived to be an unpredictable environment. In batch processing and time-sharing environments, it is usually assumed that the system knows very little about its load. It assigns resources as requested or as available but not according to a plan. Since the machine is not doing its work while it is performing its overhead functions, there is little incentive to have the operating system try to make the best decision about what to do next. The time spent deciding quickly exceeds that which might be saved by a better decision. So most operating systems use only the simplest algorithms for deciding what to do.

In IVAM the situation is different for two reasons. First, because of the multiple processors, operating system decisions need not subtract from system throughput. A processor dedicated to the overhead function can be looking ahead to see what should be done next. Therefore, the overhead can be looked upon as a one-time dollar cost rather than a constant performance loss. The reason that this is possible is that IVAM, unlike most systems, can know what it is doing. To a large degree IVAM can know ahead of time how long each step will take and what resources it will require. Also, since the IVAM output

specification is compiled into a dependency structure rather than a fixed sequence of steps, the run time operating system has some leeway in the actual order of execution. It also allows different processors to be operating on different branches of the dependency tree at the same time.

This look-ahead feature opens another option that does not normally exist. In the case where IVAM knows exactly what its processing load will be, the operating system need not be run at the same time as image generation at all. Rather the scheduling can be done ahead of time according to a sophisticated optimization scheme that will maximize throughput. The execution speed of the scheduling program itself would be unimportant as it would not interfere with throughput. Its output would be a long sequence of explicit utility calls. There would be no run time decision making or resource allocation. All that would have been done ahead of time. The IVAM system would operate much like a numerically controlled machine tool whose program has been optimized to maximize performance. Whether this option is real depends more on distribution policy than on software constraints.

The IVAM operating system also has to perform all of the overhead associated with the execution of a module. Since the IVAM module is purposely made as helpless as possible, much is required of the operating system to effect its setup, linkage to input and output arrays, and calling and return messages.

b. Operating System Conventions

In the IVAM Operating System conventions were established to guide development. Machine dependent functions would be isolated into separate modules, and RSX-11M utilities would be used wherever possible. The interface to RSX-11M would be a two-step interface. One step would be a PDP-11 macro to provide the IVAM function through RSX-11M. The second step would be a general machine independent call which would reference the macro. This means that at

a certain level the IVAM Control structures are machine independent. The distinction between the Semantic System and the Physical Operating System was made for that reason. The Semantic System is designed to be machine independent except for its reliance on a small amount of machine code to boot-strap the system.

The interface between the operating system and the modules was defined in such a way as to minimize the dependency of one part of the code on another, to minimize the total amount of space used, and to maximize flexibility. Modules are always called by the system and always return to the system. Thus there is only one call to a given module. This makes it very easy to modify a module because there are not a hundred points in the system that depend upon its previous form.

c. Primitives

With the above needs and conventions laid out we were able to specify a number of operating system primitive. (Primitives are small units of function which are fundamental to the IVAM system. All larger functions are defined in terms of these primitives). The first primitives create standard calls of RSX-11M functions. The next set are those associated with the IVAM trap handler. Next are those associated with dynamic memory allocation within a single processor. This system has already been designed and tested outside of RSX-11M. Then there are those functions associated with communication of messages between processes and between processors. These are also well-defined and ready to implement. The next primitives will be those associated with the control of the file functions and the creation of the IVAM file structures and in particular the implementation of the data set directory which is one of the points of interface between the Semantic System and the Physical Operating System. There are still uncertainties in the design of these primitives,

based on the current state of our experience with RSX-11M. The next set of primitives will be those associated with control of the crosspoint system. These are completely defined. Completion of these primitives will allow simple multiple processing operation. A more finely tuned operating system will await some decisions about the operational environment and manner of distribution as mentioned earlier.

The level of system function will depend on the implementation of the primitives mentioned above. The initial primitives give us a software way of controlling RSX-11M. The next are required for module support. Then those needed to sequence modules will be implemented. The following step is image assembly from specific data sets. The data set directory will then allow the implementation of the Semantic System and a general scheme of image generation. Up to this point the system will be running mainly on the Control Processor. The addition of the crosspoint primitives and a simple system to control them will give us multiple processing operation. The final step again is the optimization of the production environment.

C. MODULES

Once the Operating System decisions were made and the initial studies of RSX-11M completed, it was possible to refine our preliminary specification of the IVAM modules. The general specification established conventions that will be observed throughout the implementation. IVAM modules will be called as FORTRAN subroutines even if they are in fact implemented in machine language. They will return through special IVAM traps. The normal FORTRAN RETURN will never be used. They will be stored as unlinked relocatable routines. They will not go through a Task Builder or collection step. This means that we will have to fool the system into calling a subroutine. It does not want to do that. It wants to give control to load modules with no external references.

However, that would require each module to be collected with a dummy main program which when called would trap back to the system to get the subroutine parameters and then generate the actual subroutine call. While this might appear easier to implement, it is not, because another trap process has been created which must be coordinated with the main program call. It is simpler if the number of system interfaces and the amount of dependent code are minimized. Also there is the obvious fact that the call to the main program and immediate trap for parameters are superfluous if the system can call the subroutine directly.

It was decided that the modules would not do any input or output. The system loads the module code and the input data sets and sets aside space for the output data sets. It then generates the FORTRAN call that passes the bases of the input and output data sets as array parameters. This decision is important because it means that it is not necessary for each module to have a copy of the FORTRAN I/O routines. In some systems a significant percentage of the disk space is devoted to multiple copies of these routines. Thus the modules are smaller. They are also not dependent on the system's I/O structure.

Modules are usually conceived as generating output data sets from input data sets without storing along the way. This means that the module itself contains no dimensioned arrays. If it did, that array would have to be dimensioned to accommodate the largest data sets the module would ever handle. This space would have to be set aside even when the module was working on much smaller data sets. For example, the mapping module will be asked to transform the thousands of points in the U.S. grid one minute and then be working on the two or three points corresponding to the locations of as many cities which are to be inserted on that map. It seems unreasonable to set aside the same amount of space for both problems.

Similarly the sizes of the input and output data buffers are not fixed. One of the parameters to the module is the amount of output to generate. If it runs out of space before the data is exhausted it traps to the local Operating System which sets aside additional space and restarts it. In this way the module can handle very large and very small data sets. This problem is solved once for the system, not once for each module.

The intent of the IVAM modules is to provide a very general interface between units of code which do very specific things. Each IVAM module performs a very specific function in a very specific manner. Where another system might have a general routine capable of handling many different cases, IVAM would have a number of modules, one for each case. The argument for this is that general routines spend much of their code being general, testing the input parameters to see what options are intended. Also in any single run only some of the options are being used and the rest of the code, and therefore the rest of the memory space, is being wasted. IVAM modules require space only for the task at hand.

Even for a single task, the IVAM approach is to make the code extremely modular. While the code to perform contouring could be considered as a module, we decided that each of its processing steps should become a module. This is because these steps are common to a great number of other processes. Because of this decision it becomes easier and easier to implement new processes because existing module steps are likely to be applicable. Therefore a new process may just require the concatenation of existing modules with one or two that are specific to the process. This is much as one can define functions and concatenate them with the arithmetic operators in FORTRAN.

Another decision that makes the modules easy to use and the system easy to debug is that modules do not call other modules, nor do they use the system to call other modules which ultimately return control back to them. Thus

there is no concept of resting of modules. Each module takes all of its data and processes it to the completion of that step. It then returns to the system which applies the next module to the data. There are some exceptions to these rules but even they maintain the important aspects of the convention. For instance, some modules will generate a number of data sets one at a time, each of which can be processed independently in succeeding steps. As each of these data sets is created, the module returns to the system which takes responsibility for the new data set which may be passed to another module in another processor. The system then returns control back to the original module. Notice that while control is passed back to the module no results are returned. Once a module has been started it receives no further data input from the system or any other source. This is another convention which serves to minimize the interdependence of modules.

To make the modules work as described in the last section, it was desirable to create a number of standard data formats so that a module is characterized by its input and output data formats. If it becomes desirable to apply a module to a data set that is in a different format than that which the module expects, a special format translation module is automatically invoked by the system which knows what formats are associated with what modules. Thus the modular concept has been carried several steps farther than it is in most systems. The result is that it is easy to augment the system and to take advantage of all existing code. Modules are also easy to rewrite or replace. In almost every case the module conventions lead to efficient code. In the few cases where a slightly more optimum approach exists we were quite willing to trade off development efficiency and overall system efficiency against the local optimization of code.

1. Module Implementation

The module conventions just described were actually defined in the context of the kinds of functions we know IVAM has to perform. Quite a number of these modules have been developed. These are divided into several categories: image element generation, image element manipulation, and image assembly. The first two are mainly mathematical and written in FORTRAN, while the third deals with the shuffling of large data sets in ways that are not done conveniently in FORTRAN. These were written in machine language. Many of them are candidates for hardware or firmware implementation.

In the mathematical modules the first need that was recognized was the one for a standardized data space in which the calculations could take place. Thus, before computations are performed the data must be normalized so that data from different origins can ultimately be merged into a common image space.

a. Mapping

Once these conventions were established we decided to implement some of the machine independent tasks as modules to test the adequacy of our concepts. We chose the mapping module as a starting point.

The first step in designing the IVAM mapping modules was the investigation of two existing general mapping systems: C.A.M. (Cartographic Automatic Mapping) which was originally developed for the CIA, and SUPERMAP, which was originally developed by R. L. Parker of the University of California at San Diego and is currently in use in the Center. It was decided that C.A.M. would be easier to work with because of its superior documentation.

Given this working mapping system, one might wonder what remained to be done to create an IVAM mapping module. The difference between the system and the module is one of intent and application. C.A.M. solves all the problems associated with computing, displaying, and labeling a wide variety of map projections. Much of the code is used to support this generality. On any

given run most of the code will not be executed. In the IVAM system only two mapping requirements exist for certain: from earth coordinates to satellite projection and from earth coordinates to a stereographic projection. Therefore most of the C.A.M.'s code is forever irrelevant. Also generality, where it exists in the IVAM system, is at the system level which is designed to support it and not at the module level where it must be accomplished in FORTRAN which is not. Also, the mapping function is only one of many which must be performed to generate an output image. The mapping module itself does not produce visual output. It produces an image element, not an image. It does not label or plot; these are IVAM system functions which must be guided from the top level of the system, not the bottom. What remains in the module then is pure algorithm - just the mathematical code for mapping input data sets into output data sets. This is as it should be. FORTRAN is good at mathematics, poor at logical or relational expression.

Also the resulting module is small; it requires little core memory space. This is important because the system performance will be a direct function of the percentage of the code which can be core resident throughout the execution of a scenario.

For the satellite projection module it was necessary to modify C.A.M.'s perspective projection section so the algorithm accomplished the desired function. In the perspective projection the viewer is assumed to be directly above the center of the map whereas the satellite, in general, is not. Therefore the algorithm had to be changed to compute a general satellite perspective which allowed for movement of the satellite as opposed to a perpendicular perspective which fixed the viewer above the center of the area to be mapped.

For the more conventional flat maps several transformations were considered, including the Mercator and the stereographic. The stereographic was chosen

because most of the people consulted seemed to think it produced a more pleasing result and is the projection most used for current weather maps. These modules are coded and tested, running according to IVAM module conventions.

b. Contouring

The next set of processing modules to be coded were those associated with the contouring process. In most systems a function like contouring would be accomplished by a single module but in IVAM we have broken it into a number of module steps so that these steps may be used elsewhere. The first module step is to select the input data. On each of the systems where we run this module the data is acquired somewhat differently. Therefore this module tends to be system specific. The IVAM module here is tentative. It is subject to change depending on our relationship to the AFOS data base. The next step is the normalization of the coordinates of the input data set. Next is the module that computes a Uniform Grid of values based on weighted interpolation of the input points. The next module contours the values in the Uniform Grid. It searches the grid until it finds a curve which it then traces completely. After it has the complete curve it returns to the system which records the value of the curve, its location in the system, and the number of data points it contains. The system then has complete responsibility for this curve, and the module knows nothing about it. The system can then return to the contouring module for another curve or it can take each curve through all the processing steps before it gets the next. This decision is at the systems level. It has no effect on the modules. Assuming the system returns to the contouring module, it will keep producing curves until the Uniform Grid is exhausted.

Actually the curves produced by the contouring module are not curves at all. They are connected line segments. The vertices of these line segments

are in standard coordinate space. Therefore if the contours are to be overlaid on a satellite image, these lines must be passed through the satellite projection mapping module. To make the contours appear as pleasing smooth curves instead of zig-zags, the lines are passed to a smoothing module. This smoothing function is of general utility throughout the system and therefore was written as a separate module. At this point the mathematical part of the contouring process is complete. Each line exists as a smooth curve in its own copy of the standard space. It was decided to keep the lines separate until the last possible step to provide the maximum flexibility during image assembly and fill-in. The modules that take the smoothed curve through the steps which remain before an image is produced belong to a different category and will be discussed separately. The same is true of notation. The labeling of curves is an image function and this has nothing to do with the module that generates the curve.

As an example of the benefit of the module design, the streamlining of winds requires only one module to be replaced in the contouring chain. This means that not only do the functions have to be coded only once, but much more importantly there is only one copy of each module on the disk. The only time a module is duplicated is when it is desirable to have it running on two machines simultaneously.

c. Image Element Manipulation

The next class of modules that have been coded contains those that take an image element and alter it in some way before combining it with other image elements to form a completed image. Since each of the elements is stored separately at this point it is easy to manipulate part of the image while the rest remains fixed. In this way we can perform animation of image elements without recomputing the whole image. This approach was designed to

duplicate the layering techniques used by animators. It was chosen over some of the other possibilities because it yields a much greater flexibility of image formats. There is a price paid in the amount of storage required, but this price is cheerfully paid during the development phase because the optimum solution will probably be implemented in hardware--not software--and cannot be identified until the result it produces has been accepted by the media.

These element modules were defined to accept data elements defined in the standard coordinates and to produce results also defined in that space. In that way they can be inserted anywhere in the processing chain where data is in that format.

The first module to be implemented performs a simple translation of an image element on the screen. Thus an image element can be defined around zero in the standard space and then moved anywhere on the screen. This is of use not only for animation but also for the placement of alphanumerics and weather symbols.

The next set of modules are those required to scale an image element. The scaling can either be symmetric in the X and Y directions or can be uneven so that an element is squeezed either horizontally or vertically. When applied asymmetrically the result is apparent three dimensional rotation of the planar element. The symmetric scaling functions are useful for zooming. They also allow the system to define a useful object at a standard size and then scale it down to whatever size is desired and to use the translation modules to place it anywhere on the screen. Note that an image element need be defined only once even if it occurs several places in an image at different scales. This feature is very important for alphanumerics. Also the ability to squeeze an image element horizontally or vertically was included because it allows a variety of alphanumeric titling techniques.

A separate windowing module was created because it allows IVAM to pass through a larger image space or to define split screen and insert images.

A two-dimensional rotation module has also been coded. This will be used after the satellite mapping module when we have the rotational information about the orbit. The rotation module will also be used for animation of weather symbols and objects like the football or the tractor in the video tape made last year. The module was defined so that rotation can be around any point on the screen. In the tractor sequence there are two simultaneous rotations. The first is the slow counterclockwise rotation of the whole tractor about its front end which produces the sinking of the rear end. Second is the independent rotation of the back wheel that shows the lack of traction. Another example of the need for the rotation module is the rotating wind arrow in the cable sequence.

An additional module has been defined but not yet coded. This is a module that transforms one image element into another. Actually the module only computes one of the steps in the transformation. It must be called successively from one weather situation to another or from one cloud pattern to another or from one radar sample to another. It can be used as an alternative to the cross fade transitions used in parts of the film.

d. Video Chain Modules

The last class of modules defined and now in the process of implementation are those associated with translating an image element defined in a continuous mathematical space into a line and pixel video format and then assembling these elements into a final video image. The steps in general will run as follows: computing the intersections of the mathematically defined elements with scan lines, ordering these intersections by line and pixel number, packing this ordered format, merging packed elements into a Scan Topology, and then

filling in the pixel by pixel detail of the image. These modules require a great deal of thought, suggest myriad options, and represent the rate limiting steps in the system. Modules have been designed, coded and implemented for each of these steps. In each case several approaches will be tried and the final implementation versions are still some time off; however, what exists provides a test bed in which to develop the other parts of the system. The current approach is based upon some very sound thinking and will be described below.

The first step in the realization of an image element is its translation from a continuous mathematical space into a discrete video space. The module that performs this function is called the Line Intersect Module. It was written in FORTRAN because its function is basically mathematical. The routine traces the curve that defines an image element and computes the intersections of the curve with TV scan lines as they occur. These intersections are often a single output array of Y-X pairs defined in our standard video coordinates.

We have defined the upper left corner of the screen to be line 1, pixel 1. There is no line = 0 or pixel = 0. This means that zero can be used as a flag value in any of the video modules. Also note that Y values increase as you move down the screen. This is in contrast with mathematical convention that usually makes the lower left or the center the origin. This module also keeps track of the number of crossings and the greatest and least line number crossed by the curve. This information is passed back to the system where it can be used for look-ahead purposes by the system and passed on to other modules for an aid in using memory optimally.

The scan line intersections are computed in the order of their connectedness on the curve. However, they appear in the output according to their line and pixel values. Next to the fill-in process this ordering step may

be the most time-consuming one in the IVAM system. A number of approaches were considered. Each had its own advantages in terms of trading off speed against space. The problem is that the fastest methods use a great deal of space with complex images. With simpler images or image elements, however, these methods work quite well. Therefore the decision was made to order each image element separately. When this is done the only elements that are large enough to present a problem are the U.S. grid and very large contours. If an element exceeds the maximum available space the system simply truncates the element when it runs out of space and creates a new element that contains the overflow. Later during the merge step the two parts of element are reunited into a single curve.

The method of sorting currently being implemented is the Linked Line List Module. This module uses a combination of indexing and list processing to accomplish its function. While it uses two words per crossing, it only requires as many word pairs as there are crossings. This means that the amount of space it requires for the output data set is exactly the same as that required by the input data set and varies with the size of the image element. The method used takes each line-pixel pair and indexes into an array using the line number. The line entry in this array is the first crossing on this line that has been found so far. The second word in the entry is a pointer to the entry for the next crossing which has the same format. The last entry on a line is a zero pointer field. By allocating the linked crossings as they are needed after the indexed line array, a minimum of storage is used. The indexed line entry makes finding the right line very fast and the ordered linked list of crossings for each line makes it easy to enter a new crossing into the list by changing pointers rather than rewriting the list. The combination of the two techniques in the same algorithm provides for a very fast module. Many of the other

techniques considered were slower and required setting aside the maximum amount of space the process will ever need rather than only using as much as needed for the curve currently being sorted. Also the decision to sort the elements separately means that at this point in the processing there need be no curve identifier with each line crossing which results in a space savings.

As mentioned above, the Linked Line List Sort Module uses two words for every crossing, but this is a working construct. Once the sorting process is complete the link fields can be squeezed out and the storage required cut in half. This is useful where a sorted element is going to be stored on disk for awhile. Or during the image process, use of the packed format may be the only way it is possible to get all the elements into memory at once. The module that accomplishes this function has been coded and is called the Pack Module. The next step in the software video chain is the Element Merge which assembles the separate image elements into a composite outline image. The module code takes an arbitrary number of ordered elements in packed format and produces another ordered list of crossings where each crossing is accompanied by a curve identifier. Currently the crossing value and the curve identifier are stored in separate words even when they could both be packed into one word. The decision not to pack at this time is based on the fact that the fill-in process is currently being done in software. Therefore, the computer would have to spend time in both the packing and the unpacking steps. Thus, if the outline image is consumed immediately after or even as it is generated, there is no reason to be conserving the storage. If an image is to be stored on a disk then there is a reason to pack it. Or, if the fill-in process is to be accomplished in hardware, then the unpacking step can easily be accomplished in the hardware.

The last step in the software video chain is the fill-in process. This module is written in machine language for three reasons: first it is small;

second it may be reduced to hardware; and third it is the most heavily used code in the system. The fill-in process steps through the curve intersections listed in the outline image, resolves priorities, i.e. which domains are in front of others, and then assigns a value to each pixel between the crossings.

After each pixel is generated it can either be output immediately or accumulated into a line buffer. This decision depends on the characteristics of the data path to the display refresh. If there is a random access refresh buffer, the pixel values can be transmitted as generated. If, on the other hand, they are sent down a limited bandwidth serial path, as will be the case with the disk refresh on the Remote Terminal, then the output process requires a fair amount of handshaking and buffering.

The modules described above provide a complete video chain. Examples of each have been coded. They allow the IVAM team to start addressing some of the peculiarities of digital video. New or alternative modules can be fitted into the existing chain. Because of the modularity, it is easy to work with any step in the chain. The reason for establishing this point as a goal was that now another part of the IVAM effort has become independent of the others. The support now exists for image quality studies to proceed without assistance from the systems people who will be working on the image formats and the operating system.

e. Image Formats and Control

The modules described in the last section perform the computations that generate and manipulate image elements. The constructs to be described now are used to specify how these image elements will be assembled into complete images. Also of interest is the software designed to make use of two significant devices in the development system: the data tablet and the video tape recorder.

During the past year the semantic structures have undergone considerable conceptual maturation. A year ago there were several data structures that were of fundamental significance to the IVAM system. These were Data Sets, Descriptor Nets, Operation Nets, Structure Nets, Image Formats, and Semantic Nets. All of these entities still exist but their functions have been altered and refined considerably. Before, each data set was qualified by a descriptor which would indicate its parameters, the Structure Net required to navigate it, and its semantic significance. Now, the descriptors and the data sets still exist and are still related except that the descriptor has assumed a much broader function while the data set is the very lowest construct having no other function than the storage of data. The format of the data set used to be called out by the descriptor, which indicated a particular Structure Net that was to be used to navigate the data set. The Structure Net was a specification for the algorithm to be used to access the next datum. However, as the authorization for and the delivery of the IVAM hardware was delayed further and further, the generality of the Structure Nets was constrained in order to speed implementation. In the past, modules were absolutely independent of the format of the input data sets. Now a number of standard formats have been specified. Each module is specific to these data formats. Format independence exists, but at the system level, and now it exacts a performance price when used. Now, when a data set is not in the format expected by the next operation, a format translation module is invoked which makes the data compatible. The Structure Net, when it exists, refers to disk files and is pointed to by the Data Set Directory.

The Data Set Directory is a new construct which is shared between the Physical Operating System and the Semantic System. The Semantic System deals only with the existence and directory identifier of a data set while the

Physical Operating System is concerned with its length, format, location, etc. A data set entry may or may not have a descriptor associated with it. System data sets which exist apart from any particular image do have their own descriptors, while in-process data sets, i.e. those intermediate products generated during the production of an image, are referred to by terminal nodes in the descriptor for the image, but have no descriptor of their own.

The Descriptor then has changed a great deal. Its physical structure has changed little, but that structure has been found rich enough to accomodate additional function. This development is less the result of a single dramatic decision than the outcome of an evolution that occurred as we designed these structures and then redesigned them around RSX-11M.

The current descriptors combine the function of the scenario, presentation segment, image format, dependency net, and descriptor into a single construct. The origin of this generalization was in the design of the Image Format. The Image Format is a very general specification for an image or an image element. It is not associated with any particular time, location, weather parameter, or color assignment. It simply divides the image into a number of domains which can in turn be divided into subdomains. At this point it was already apparent that there would have to be an element of recursion in the definition of an Image Format. This became unavoidable when we considered the case where we wanted to include a complete image as an insert in another image. In such an image the Format contains a complete image as a subset and so the definition of an Image Format has to be recursive, i.e. an Image Format is defined in terms of image formats.

Once this step was taken it was easy to think of an image format as including more than one image or a sequence of images. To accomplish this change several new node types were required. The Image Format itself requires

the definition of a variable node which is looking to be bound to a particular parameter. Then there was the image node which indicates that all subordinate nodes are to be assembled into a complete image to be shown for a given amount of time. If one image node is subordinate to another, it will be surrounded by manipulation nodes which indicate how this image is to be scaled and translated before being inserted into the controlling image. A repeat node indicates that the image subordinate to it is to be repeated so many frames. However, the repeat node branches if there is partial animation. The animation branch indicates what parameters and what manipulation modules are to be applied to the subordinate image elements. The animation of completely different images would appear as simply a sequence of image formats in the descriptor. When the repeat and animation nodes are applied to whole images rather than to image elements the descriptor is being used as the specification of a segment and then a scenario without any change in its structure.

In use, the descriptor is not a static specification. On the contrary, the descriptor is continually being altered as the image generation proceeds. At any moment its state is a complete description of what remains to be done given the results attained so far. Each of the nodes in the descriptor corresponds to a module and its subordinate nodes are the modules that produce its input. When the subordinate nodes have been processed, they collapse into data sets. When all of the nodes subordinate to a given node are also collapsed, the system seeks to apply the module corresponding to that node to the data sets subordinate to it. The system consults the operation net associated with the module to verify that the subordinate data sets are indeed suitable as the module parameters.

At any point in time the descriptor may not exhaustively list every module and every data set required to achieve the desired output. Instead the

descriptor may contain macro nodes which represent a group of nodes. The system then expands these macro nodes only when it is actually about to process that part of the image. This allows a shorthand form of segment specification that gets specific only when the system is ready for specific instruction. These expansions require a series of very simple structures at the systems level which constitute a transition net grammar.

In addition to these changes in the IVAM Semantic System we have purchased two pieces of equipment which are primarily intended to assist in image generation. The first of these is the data tablet which is a very important part of any computer graphics development program. It provides an easy way to enter graphic information into the system. The tablet will be used to generate domain-defining curves that will be used as test data for the plotting software and the color assignment algorithms. The tablet will also be used to define complicated control patterns such as those used to control enhancement, animation, and those defining texture. The basic mechanisms associated with the tablet will allow the developer to define, name, store, and edit curves. Then these curves can be applied anywhere in the system where such a data set is expected as an input. The tablet software is specified and coding will begin in late April.

The video tape recorder is the second significant tool for image development. The software being designed for it will allow IVAM to create demonstration tapes of professional quality. Since this recorder will be under computer control it will be able to do frame edits which will overcome some of the throughput limitations of the development system which uses a display, which must be loaded through a low bandwidth channel. Without this recorder we would be able to record only the shortest sequences before there would have to be a very obtrusive loss of synch due to a bad edit.

D. CONCLUSION

The overall result of the year's effort was impaired by the delays in the authorization and delivery of the hardware. Nevertheless considerable progress was made in the finished specification and coding of machine independent IVAM modules which are running on both McIDAS and the Univac 1110. A very mature system concept has been defined and is now being implemented. Also there has been significant experience with the digital video display on McIDAS. The support systems are fully defined and ready to be implemented as soon as the rest of the interfacing is completed. The current state of the software effort can be broken down as follows:

<u>Areas of Effort</u>	<u>% Completed</u>
Operating System	20%
Image Formatting	30%
Modules	50%
Testing Results	30%

IV. HARDWARE CONCEPT STUDIES

A. THE IVAM SYSTEM

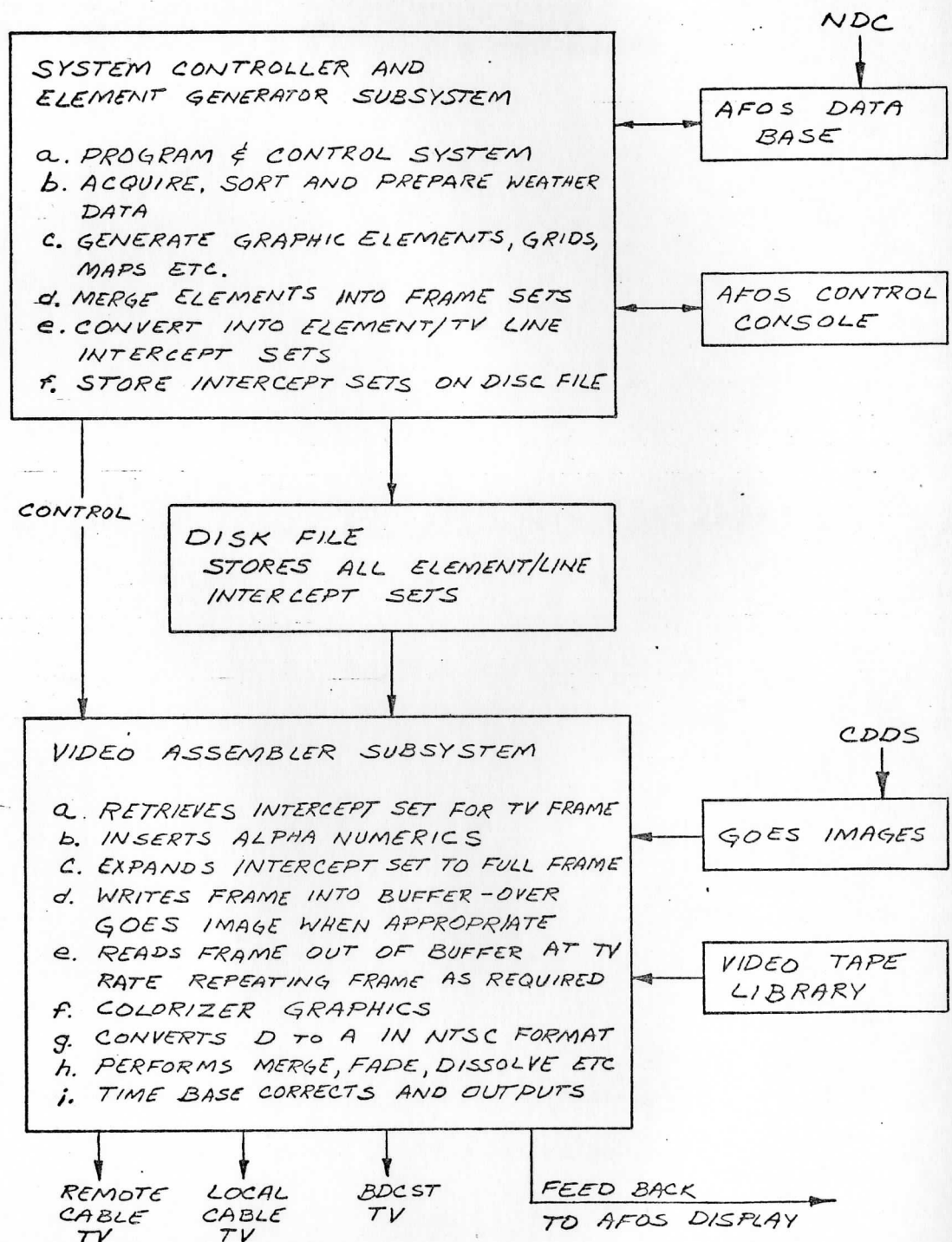
The job the IVAM system is to perform is a straight-forward one, although it takes a relatively complex set of equipment and software to accomplish it. The IVAM task is to convert weather information from the several formats in which it exists to high quality television presentations automatically and without delay. The task does not include interpretation or analysis functions. Both the input data formats and rates, and the output video sequences are known completely by the system before it starts to work. The IVAM task is a production job, and the system is being designed to do the production job and nothing else.

Figure IV-1 presents a much simplified block diagram of the IVAM system. Several aspects are worth noting:

1) There are three input storage devices: the AFOS data base which receives and stores selected information from the National Distribution Circuit; the GOES image digital disk (similar to the NESS Satellite Image Sequencer); and the video tape library which holds several hundred local area weather examples. Each of these devices is tailored to the type of data it stores and the input-output requirements for that particular data set.

2) The IVAM system is divided into two major subsystems connected by a medium-sized digital storage disk. In the Controller and Element Generator subsystem the jobs are to:

- a) Program and control the entire IVAM system
- b) Acquire the meteorological data from the AFOS data base and prepare it for IVAM processing
- c) Generate graphical image elements required to describe meteorological parameters, maps, contours, etc.



SIMPLIFIED IVAM SYSTEM PLAN
FIGURE II-1

- d) Merge the image elements into sets corresponding to final TV frames
- e) Convert the information from image element format to sets of image element/TV raster line intercept designators
- f) Output the element/line intercept sets to the Disk File

The control task is continuous but it never becomes a large-scale job in comparison to processor load. It is the image element production that determines the size of this part of the system. Weather information enters the AFOS data base continuously with specific message formats being repeated at periods of one, three, six, twelve, and 24 hours. Some information, such as emergency notices, will be received on a non-scheduled basis, but these are exceptions. Since the IVAM controller anticipates these scheduled input times, and because it knows what the system must produce during the next several hours, it schedules the production of image elements at a steady, nearly uniform rate. In this way the basic image parts are produced most economically and then stored in a form which takes minimum space until needed by the video assembler and producer.

3) The Disk File holds only element/line intercept sets and their labels. Storage of inactive software modules, segment descriptor files, grid point data sets, and other basic production tools is in smaller disks which are part of the Control and Element Generator subsystem. Since most image elements will be used more than once, the Disk File reduces the production job of the element generator by a large factor as described below.

4) The Video Assembler subsystem is a hardwired special purpose processor. It performs a number of specific processing tasks in a tightly time-controlled serial assembly line fashion. As directed by the IVAM controller it:

- a) Retrieves from the disk store the element/line intercept block required to make up a specified TV frame

- b) Inserts alphanumerics
- c) Fills in the space between the raster line intersects to make up a complete line
- d) Places GOES image in one frame of a three-frame digital buffer when required
- e) Writes graphic image into buffer overlaying GOES image if present
- f) Reads the completed frame out of the buffer at full video rate (30 frames/second) as many times as required to meet segment specification and builds up next image in second buffer at the same time
- g) Converts image from digital to analog signal in NTSC format and performs time base correction. Product is completed.
- h) For narrow-band transmission to remote cable outlets via conditioned telephone lines, the appropriate finished images are transferred to a low output rate buffer (Note: the need for this service is not firm and the buffer is a design alternative not yet incorporated in the IVAM program)
- i) As appropriate, sequences from the video tape library will be fed directly into the time base corrector

The satellite image, when required, will be written into the buffer at full TV rate and then the graphics will be over-written at 1/6 TV rate. The graphics pixels will be flagged with an extra bit so that only these pixels will be color-converted.

A three frame buffer is required in the Video Assembler so that fade dissolves between two images can be produced without losing output continuity. Dual channels must be provided from the buffers through the digital-to-analog converters because the images cannot both be colorized and fade-merged in digital form - the fade dissolve merging must be done using the analog signals.

The dual channels also permit a large variety of color, texture, window and other special effects to be created without adding to the load on the Element Generator subsystem.

B. SYSTEM TIMING AND SIZING

It is easier to work backward through the IVAM system since the output rate is firmly determined at 30 frames per second and the output volume is relatively well-fixed at 20 minutes of TV segments per hour. The 20 minutes is made up of one full update of cable TV presentation (5 minutes) per half-hour and up to ten minutes of segments to be sent to broadcast stations per hour. The total output of frames is:

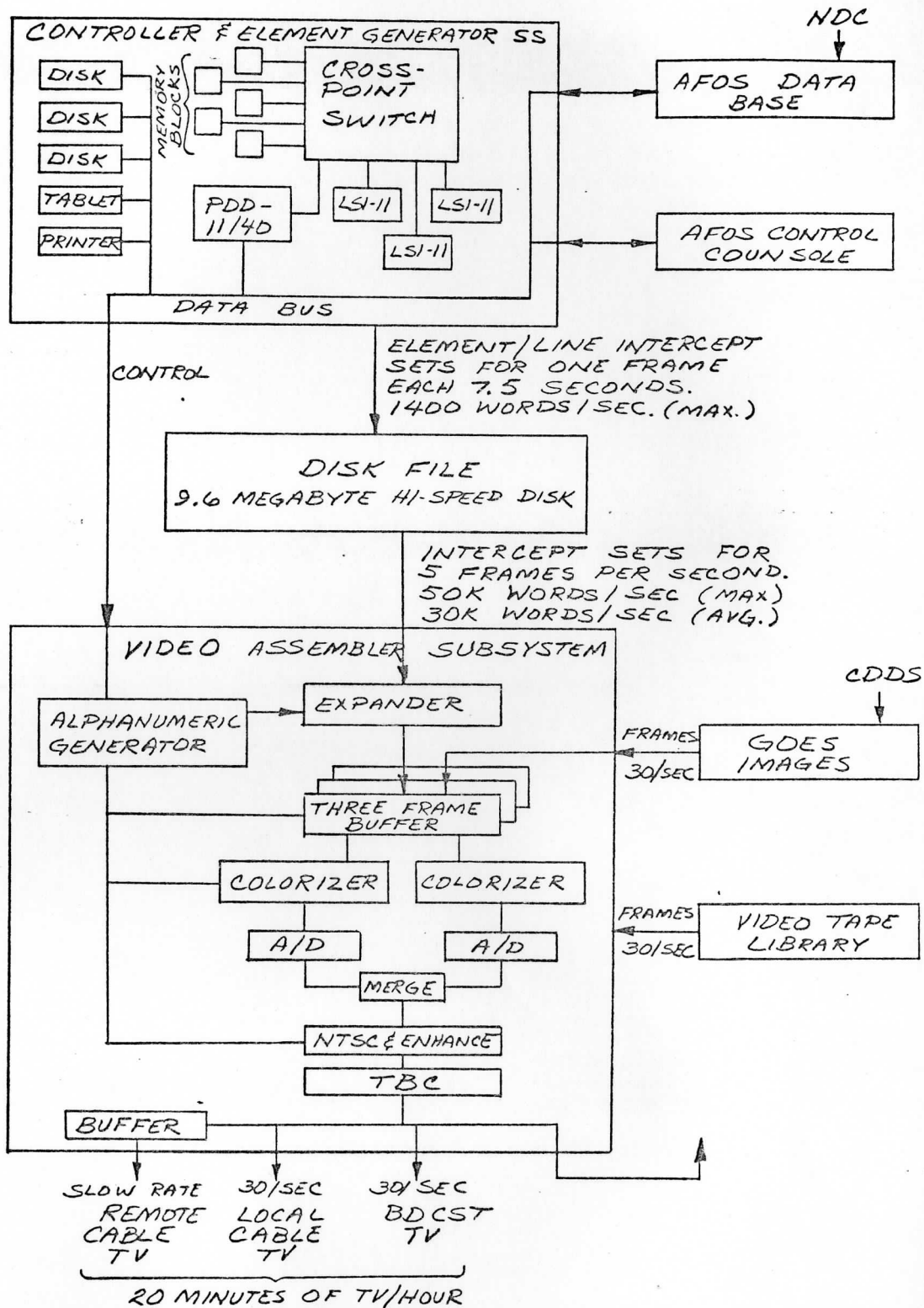
$$20 \text{ minutes} \times 60 \text{ seconds/minute} \times 30 \text{ frames/second} = 36,000 \text{ frames}$$

Previous studies of simulated IVAM presentations and similar TV programs show that each TV frame is repeated from 6 to as many as 120 times. At six repetitions per frame the Video Assembler subsystem must be capable of accepting a peak rate of 5 new frames per second. The average new frame rate will be much lower.

If we examine TV graphic images, we find that the most complex average fewer than 20 image element intersections per TV line. Using one word for each element/line intersection, the maximum number of words required to describe a set of image elements for one TV frame is

$$485 \text{ lines} \times 20 \text{ crossings/line} = 9700 \text{ words/frame}$$

We will probably store the image elements on the Disk File in this line intersect format. Therefore, at five new frames per second we would have the maximum transfer rate from the Disk File of 50K words per second - a rate which can be met by available disks even after making generous allowance for disk access and head movement delays.



IVAM SYSTEM BLOCK DIAGRAM
FIGURE II-2

The output of the Element Generator subsystem can be at a much lower rate. The rate at which new sets of image element/raster line intersects must be generated is the product of the following factors:

1) Element generation can be going on continually whereas the Video Assembler will operate only 20 minutes per hour. This fact lowers the data rate of the output of the Element Generator by a factor of $1/3$.

2) The rate of element generation is determined by the average number of times a frame is repeated, not the minimum number. Experience with the IVAM films has shown that on the average each frame is repeated 25 times reducing the element generation rate to $1/25$ that of full video.

3) While the TV images delivered to the broadcast nets will appear to be quite different from those sent to the cable TV outlets, many of the images in both sets will be generated from the same element/line intercept sets. Also, each hour's presentation will include images from data more than one hour old. We estimate that these two factors reduce the new frame production rate by at least $1/3$.

Therefore, the total number of new output frames to be computed each hour is:

$36000 \text{ frames/hour} \times 1/25 \text{ (average repetitions per frame)} \times$

$2/3 \text{ (previously generated frames)} = 480 \text{ new frames per hour}$

or a new frame must be computed every:

$480 \text{ new frames/hour} \div 60 \text{ min./hour} \div 60 \text{ sec./min.} = 1 \text{ new frame/7.5 seconds}$

The Controller and Element Generator subsystem is being developed to operate at this rate.

The Disk File must have the capacity to store all of the finished sets of image element/raster line intercepts which the Video Assembler subsystem will need. We believe the total of these sets will be very close to the

following:

480	frames one hour old or less
120	frames one to six hours old
60	frames six to 48 hours old
<u>20</u>	frames of "permanent" images
680	total TV frames of intercept sets

The maximum number of words per set is 10,000 and the minimum is about 2,000 with the average being close to 6,000 words. Therefore the Disk File must have capacity for:

680 sets x 6,000 words x 2 bytes/word = 8.16 megabytes

High speed disks which meet our delay time and transfer rate requirements, and which have a capacity of 9.6 megabytes are available as standard equipment at moderate cost.

C. AFOS INTERFACE

The AFOS/IVAM interface cannot be specified completely until some time in the future; however, it is not too soon to start formalizing the interface requirements. What follows is an initial specification of IVAM requirements and a discussion of possible alternative interface approaches. This is a first step in the interface definition process. It is hoped that this first cut from the IVAM point of view will assist the AFOS Program Office so that what now appear as distinct systems can be integrated into an operational whole.

IVAM is predicated upon a successful AFOS system. AFOS gathers, distributes, and stores the nation's weather data. IVAM must make use of the AFOS data base. IVAM will be particularly interested in Service A, Service C, and all graphic products and forecasts. Local radar will be of great interest if it is incorporated into the AFOS data base, or if it is available from some other source in a format which IVAM can accept.

In many cases IVAM can use the AFOS data as it is currently planned to

be stored; however, there are three categories where changes in the AFOS data format would make our task easier. The most important is graphics. AFOS graphic products are planned to be transmitted in a form that is directly compatible with the AFOS display devices; however, the significance of the parts of the image is lost in the transmission format. The alphanumeric information labeling the contour curves is transmitted separately from the curve itself. This works fine for AFOS because the label is placed near the curve by the display and the operator makes the association. But for IVAM to try to determine which label goes with which curve on the basis of position is a very elaborate process which is not guaranteed to succeed and could be easily avoided. This information is available in the programs at NMC and is not planned to be transmitted because there is no need in AFOS. It would be easy to include the labels with the curves. We request that there be agreement in principle that this change will be made. The exact format can be decided later.

In informal discussions with members of the AFOS Program Office, the advantages and disadvantages of transmitting graphic data in the variable exclusion vector code were discussed. We would like to suggest that sending the basic uniform grid point values in an optimized (i.e., redundancy removed) code be considered as an alternative. Based on the limited information available to us, the advantages of this approach could be:

- 1) Greater efficiency because the message length might be reduced by a factor of two to four
- 2) Greater commonality because uniform grid values can be used by any graphics generation process while the variable exclusion vector code is peculiar to the projected AFOS hardware
- 3) Much more convenient to IVAM which must now add software to convert the vector code whereas we could use grid point values directly
- 4) Reduced effort at NMC since it would not have to convert grid

point values to vector code

The only disadvantage would appear to be the need to augment the AFOS display software to generate displays from grid point data. We are not able to assess the impact on AFOS, but it might even be an advantage since the necessary software is available from several sources.

There is also a class of products which are transmitted as unformatted English text. While these items are conceived as communication between human operators they present a problem to an automated system which must substitute software for intelligence. The more of this information that can be constrained to fixed formats the more will be usable by the IVAM system.

We would also like to suggest that significant system advantages could be realized if the use of ASCII text as the major storage and transmission format were changed to binary or some related abbreviated format. This would especially apply to data which is mainly numeric, such as the Service A and Service C data. The major advantages of this suggestion are:

- 1) The transmission time of the messages over the NDC would be greatly reduced. We estimate reduction by a factor of from two to four.
- 2) It would reduce the AFOS disk storage requirements by the same factor.
- 3) In the event that AFOS will ever do any analysis of the data, it is much more convenient to retrieve the raw data directly, rather than to have to scan the ASCII text for the selected parameters.
- 4) It would reduce the IVAM load of retrieving weather data for the same reasons as 3) above.

Of course, it would be necessary to use some type of abbreviated format to avoid the potential conflict between pure binary and the ASCII control characters used by ADCCP (ETX, DEC, etc.). However, such formats exist (and

others can easily be developed) to convert an 8-bit integer number into no more than two 8-bit characters, which in no way conflict with any of the ASCII control characters. This format would only be needed for transmission of the data. It could be stored on the disk as pure binary.

The major disadvantage of this scheme is that in displaying the data, a binary-to-ASCII conversion is required for each data-value. However, this is a simple step and can be done without slowing down the display process.

The data transfer relationship between IVAM and AFOS becomes very important because on the one hand IVAM represents a potential load on AFOS and on the other hand AFOS represents a potential bottleneck to IVAM. It therefore seems prudent to plan for peaks but to operate in such a way as to avoid them. One way of doing this is for IVAM to take relevant AFOS data as it comes in and not to wait until it has a real-time requirement for a given product. On the other hand it should be recognized that a certain amount of the data will be needed quickly, for it is the nature of both IVAM and AFOS to be interested in the most up-to-date information. Therefore a high bandwidth channel should be provided even if it is not fully utilized.

In software terms there are several possible relationships between IVAM and AFOS. At one extreme IVAM is passive and AFOS notifies IVAM when new data arrives and transmits it as soon as IVAM is ready. At the other extreme AFOS is passive and IVAM continually checks the AFOS data base to see if new products have arrived. A number of such possibilities will be cited below.

1) AFOS could simply let IVAM tap off the NDC or the WSO line directly. This is not desirable because IVAM would have to monitor all transmissions to determine which were of interest. It would also have to deal with the ADCCP protocols which have already been observed by AFOS. IVAM would rather not duplicate any part of AFOS.

2) AFOS could have a list of products of interest to IVAM and send them automatically in accordance with a fixed input schedule.

3) AFOS could send IVAM the PIL entry for each new product as it arrives and let IVAM then indicate whether it wants the message data.

4) IVAM could continually check the PIL for new products. However, while it is easy to find the latest product from the PIL it is more difficult to determine whether the "latest" entry is also a "new" entry unknown to the IVAM system as yet. To do this IVAM would have to save the last version of the PIL and compare it with the new version. Those entries in the new PIL which were not in the old PIL would, of course, be the new products.

The best IVAM/AFOS relationship probably is a combination of those listed above. Most of the traffic between the systems will be the routine automatic transmission of relevant products originated by AFOS as per option 2. The software constructs necessary to provide this service may not exist nor be natural within AFOS, in which case it may be necessary to use option 3. IVAM would maintain the list of products in which it was interested and check each new PIL entry against that list. As the deadline approached it would check to see which products had arrived and keep requesting those either until they came in or until it was too late. Thus there should be both a notification and a request capability. We prefer option 2 with a request capability to be used only for exceptions.

1. Software

Once the products have been transmitted to IVAM, additional software is needed to translate them into a form suitable for IVAM. This is necessary because of the difference between the AFOS data base and the IVAM data store. AFOS currently thinks in terms of messages designed to communicate directly with a human operator through a high resolution CRT display. IVAM, on the other hand, must automatically take data designed for professional meteorolo-

gists and turn it into displays which communicate to the lay public over standard TV. This means that IVAM must take data out of its message format and disassociate it from its source, such as Service A or Service C, and just present the weather quantities the public understands.

Also, graphics products which are encoded in terms of the raster characteristics must be decoded from their variable exclusion vector format, translated into the IVAM standard coordinate space, and rescanned by a 525 x 672 raster. Some sort of smoothing function may be required if the two digitizing steps interact in an unfavorable way.

Computer worded forecasts and verbal texts present a more difficult problem. While keyword analysis is possible, comprehension of natural language is a difficult problem and beyond the scope of the IVAM effort. Therefore it would be simpler if IVAM could receive the data underlying the computer worded forecasts. Where a format is filled in it would be easiest to receive the data in a fixed format where the significance of each field is already known.

2. AFOS Control

In addition to the software required for IVAM to make use of the AFOS data base there must be additional software to allow the AFOS operator to control IVAM. While IVAM is honestly presented as an automated system, it is but part of the AFOS system, an AFOS output processor which should be controllable by the AFOS operator. There are several reasons this feature is desirable:

- 1) In extreme weather situations the AFOS operator may want to take control of IVAM and communicate directly to the public.
- 2) He may want to assemble his own scenario for special situations.
- 3) IVAM will be sized to handle its peak loads which will tend to occur around the times of network broadcasts. At other times IVAM may have extra computing capability which would augment that available to the AFOS system.

4) During the field installation process each site will have its own peculiar needs which are best met locally. Therefore, the installation operator will have to interact with the system through the AFOS terminal because IVAM will not have its own terminal. The alternative is that he bring a portable control device to address IVAM.

In all of the cases described above, systems software is required to effect the communication and control, and decisions as to where the software is to reside must be made during the next several months.

3. Hardware

There are a number of points where IVAM could physically interface to the AFOS system. The trade-offs here are speed versus the existence of software support for certain interfaces. To some extent the hardware options parallel the software options mentioned earlier.

1) IVAM could interface directly to the NDC. To interface directly would be expensive because of the cost of a modem. Even after the modem IVAM could only eavesdrop, it could not be an active participant in the NDC. This means that not only would IVAM have to deal with all of the line protocol, it would have to recognize situations where a station did not acknowledge a transmission and a block was sent again. We oppose this alternative.

2) IVAM could interface through the EDS-8 interface and maintain the ADCCP line protocols. A modem would not be necessary. If this line is run at 2400 baud or 9600 baud it is too slow to be convenient. However, if it can be run much faster it may be adequate if not ideal. The advantage of this entry is that the software hooks already exist.

3) IVAM could interface directly to the AFOS graphics bus and appear to be another operators display. This seems to be a relatively natural interface because IVAM could simulate an AFOS operator and would get data without maintaining any communication protocol.

4) IVAM could interface to AFOS through a DMA bus similar to the one joining the AFOS Communication and Display processors. This would provide the needed bandwidth and may have some of the software support needed already. More software would have to be added to AFOS to allow data transmission across such a line. This method seems appropriate in that it is the AFOS interprocessor communication already.

4. Summary

The AFOS/IVAM interface must meet the following criteria:

- 1) It must provide IVAM complete "read only" access to the AFOS data base.
- 2) The AFOS data must be in a form which is usable or translatable by a computer program.
- 3) Annotation should be available with graphic products so IVAM can determine the significance of each of the graphic elements.
- 4) A control interface between AFOS and IVAM needs to be defined.
- 5) The data interface should be a high bandwidth DMA channel such as the interprocessor or graphics channels.

D. DEVELOPMENT SYSTEM PROCUREMENT

A year ago we stated that the development of the IVAM system must provide:

- 1) Flexibility for an optimum interface of AFOS and IVAM to be defined at a later date.
- 2) Flexibility for a good interface of IVAM to the media when the specifics of distribution are confirmed.
- 3) Maximum advantage of latest developments in new hardware technology.

A further consideration stated was that the software developed must not be limited to running on the machine of a single manufacturer, but the hardware choice must also allow development of general software which runs (or can be

easily adapted to run) on any machines of appropriate power. Our activity in hardware evaluation and in the bid evaluation process during the past year has been consistent with those goals expressed a year ago.

During the past year we have purchased and received delivery, or are fabricating the equipment required to implement the Control and Element Generator subsystem of IVAM. This is the "front" part of the final system and it contains nearly all of the software and performs nearly all of the data manipulations up to formation of TV images. We intend to use this equipment to develop and test software while we specify and procure or develop the equipment for the "back" end of the system. We intend to have the entire prototype system operating and ready to interface with AFOS in the fall of 1977.

The prototype equipment for the Control and Element Generator subsystem cannot produce a completed TV image by itself; therefore, until the rest of the system is in hand it will make use of the existing McIDAS data display and analysis system. The McIDAS capabilities are not optimum for IVAM but they will be adequate for test purposes. As the Video Assembler and Producer hardware becomes available functions provided by McIDAS will be replaced with IVAM dedicated hardware.

E. PROTOTYPE ELEMENT GENERATOR SUBSYSTEM CONFIGURATION

Several hardware configurations were studied for use as the development system. Since the exact size of the system needed cannot be determined until the software is largely developed, the subject of expansion capability was carefully considered. After all, ending up with a system capable of doing 90 per cent of the required task is a failure.

The single processor approach was abandoned early in the study, for the only guarantee of having adequate system capacity with a single computer was the choice of computer far exceeding the budget. Further arguments against

use of a large single computer are: (a) the inability to implement the final hardware in incremental fashion, and (b) far less likelihood of rapid decline of the purchase price in the next several years.

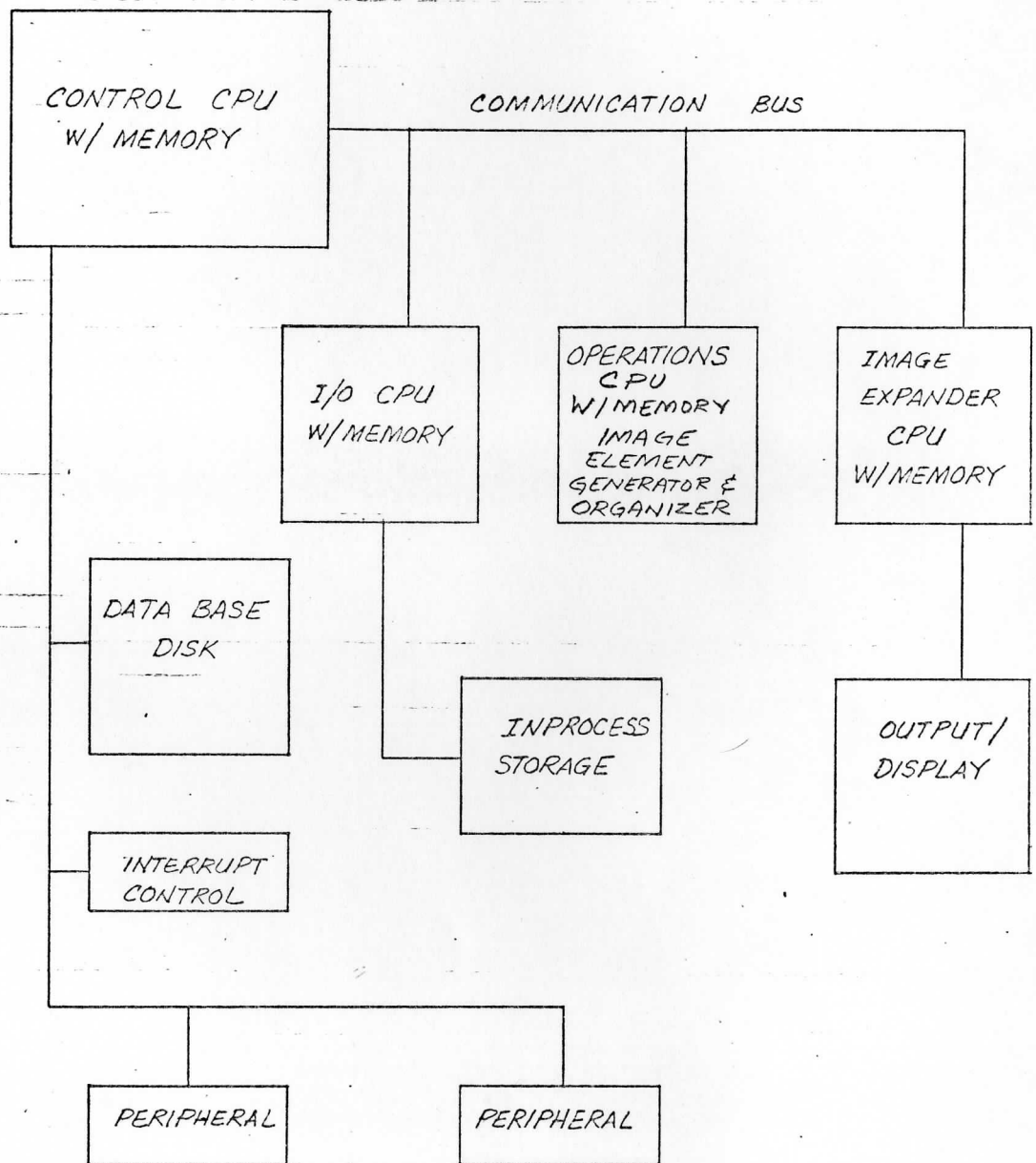
Only the hardware required for the prototype Controller and Element Generator subsystem was considered for procurement last year. Two basic configurations discussed below, and variations of them, were analyzed in detail.

1. Data Communication Bus

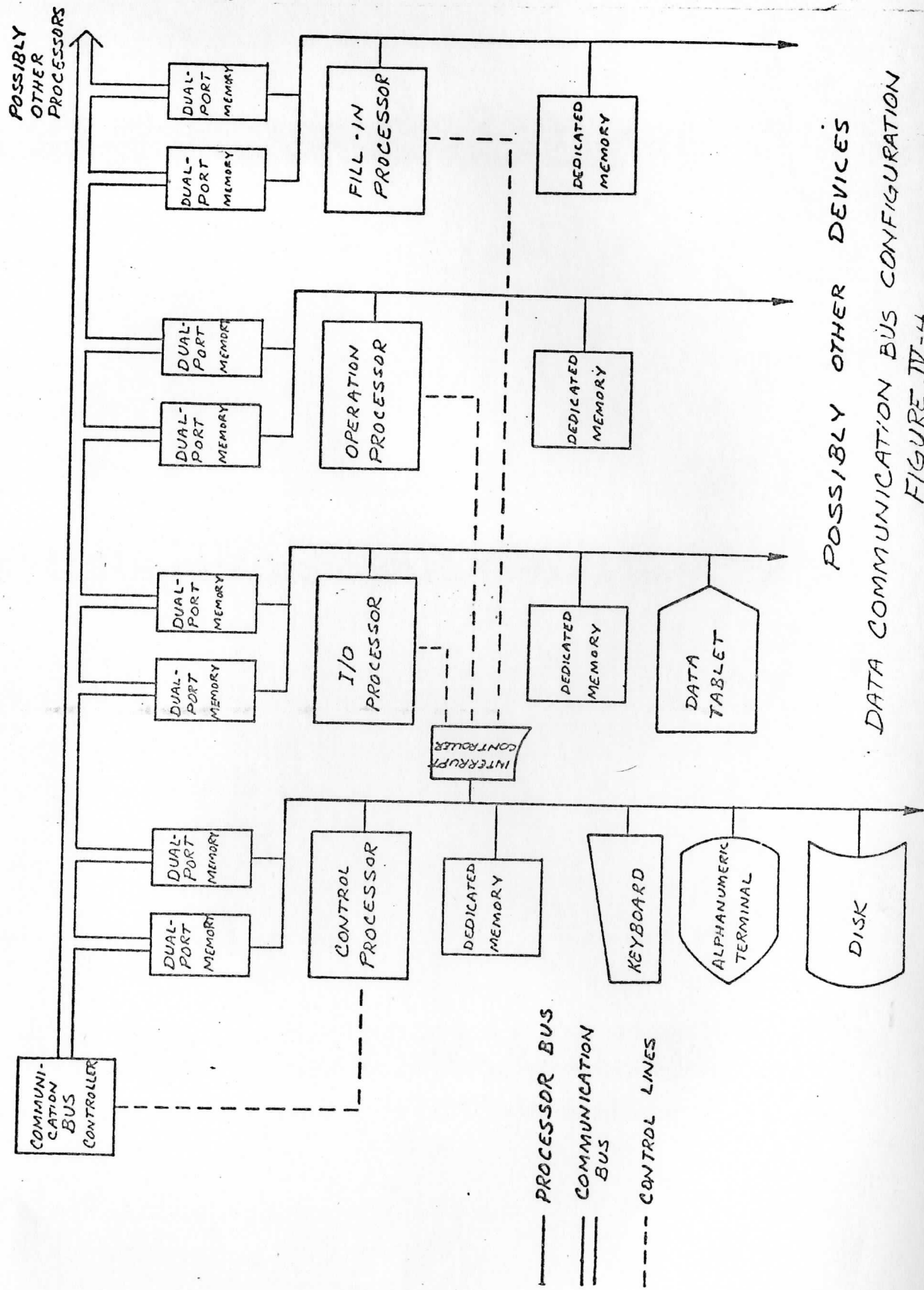
The basic Communication Bus configuration, shown in Figure IV-3, is designed to have at least four processors configured to allow transfer of data between any two of them. The system is expandable to accept additional processors, memory, or hardware devices as needed.

Since the ratio of data transfer to processing is expected to be high, the design of the interprocessor communications is of crucial importance to the performance of the system. Such transfers will consist mainly of block data rather than individual words so it is not necessary for all processors to share a common address space. Data transfer will be initiated by the control computer but may actually be effected by a microprocessor or a hardwired bus controller. Since this process may involve more than one computer it is imperative that the transfer process not interrupt or degrade the processing of any machine more than necessary. Several alternative approaches, all based on the Communication Bus concept, were considered:

a. Approach #1: The first approach, shown in Figure IV-4, furnishes each machine with at least two dual port memories as well as a small amount of dedicated storage. Each dual port memory would be accessible to both its CPU and to the communications bus. Since the dual port memories on each processor are independent, the CPU could be working in one memory while the communication bus is filling the other. When the CPU has done its task in one memory it would signal the controlling processor that it was done and start



DATA COMMUNICATION BUS BASIC HARDWARE CONFIGURATION
Figure II-3



DATA COMMUNICATION BUS CONFIGURATION
FIGURE IV-4

working in the other memory. Note that the transfer in no way impedes or degrades the performance of the CPU.

In this configuration it is desirable that the amount of memory accessible by the communication bus not be constrained by an arbitrary limit. It is worth noting that "dual port" here refers only to the ability to reference the memory from two different buses. This can be accomplished by simple electronic switching between the two buses or by true dual porting where the memory has two independent sets of addressing hardware.

b. Approach #2: The communications computer could have the pair of dual port memories associated with each of the subordinate processors within its address space as shown in Figure IV-5. In this case the transfer of data from one machine to another would be seen as a block transfer within the communications machine. Note that the transmission of data from machine B to machine C operates at program rates and may "steal" machine A cycles of up to 50%. However, if machines B and C are operating in their other dual port memories they are unaffected. Since a minicomputer having a large address space is likely to be more expensive, this alternative seems less desirable than the first approach.

c. Approach #3: In using a machine with an extended address bus as the communication controller, at least that processor is lost to the system during transfer. Therefore the question arises whether that machine can still accomplish functions other than data transfers. One way of relieving the communication burden on the computer whose bus has access to every machine's memory is to create an additional interprocessor path that is dedicated to transfers between two of the machines, as shown in Figure IV-6. This lowers the volume of traffic along the main bus. However, the software in those machines would have to be more sophisticated.

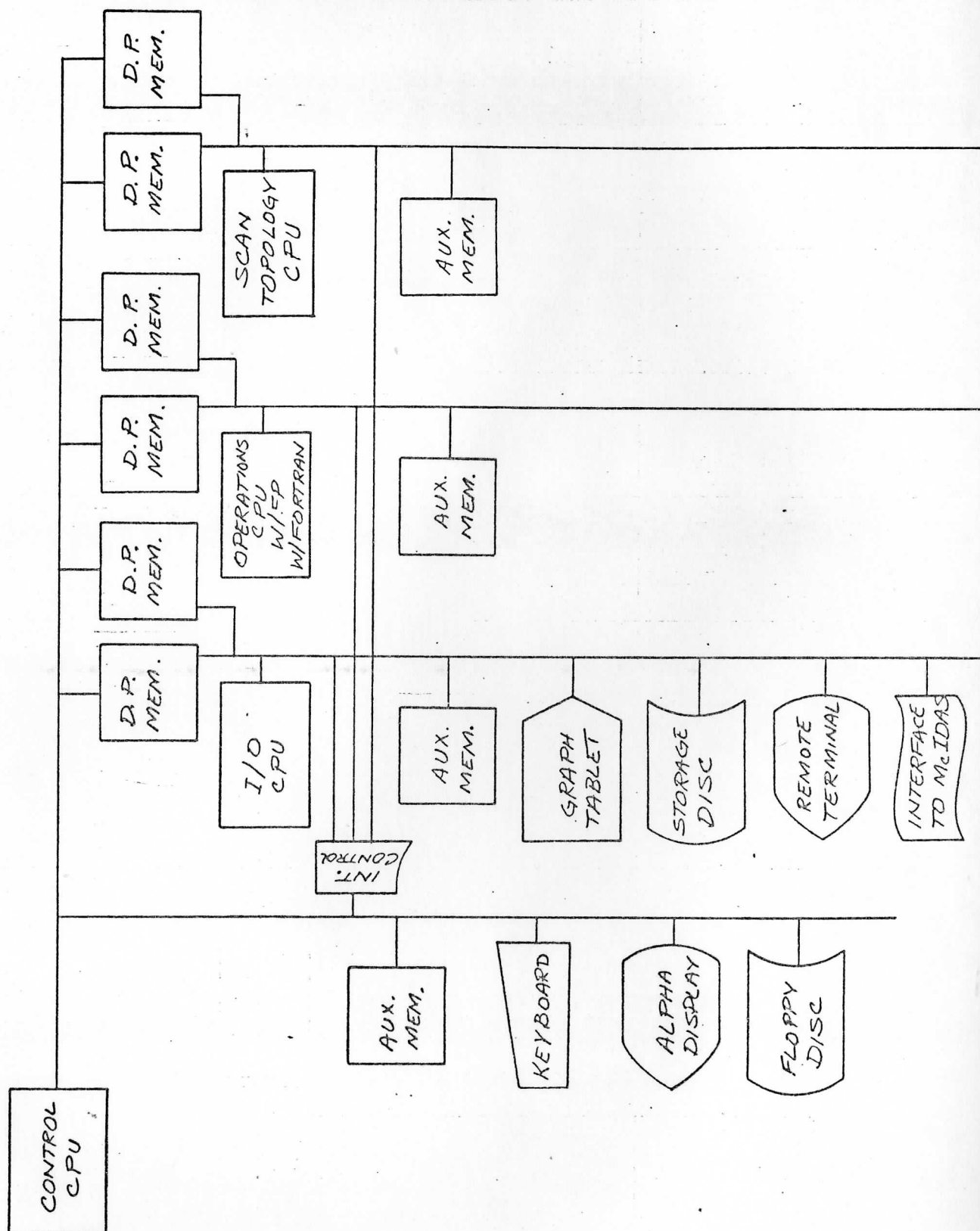
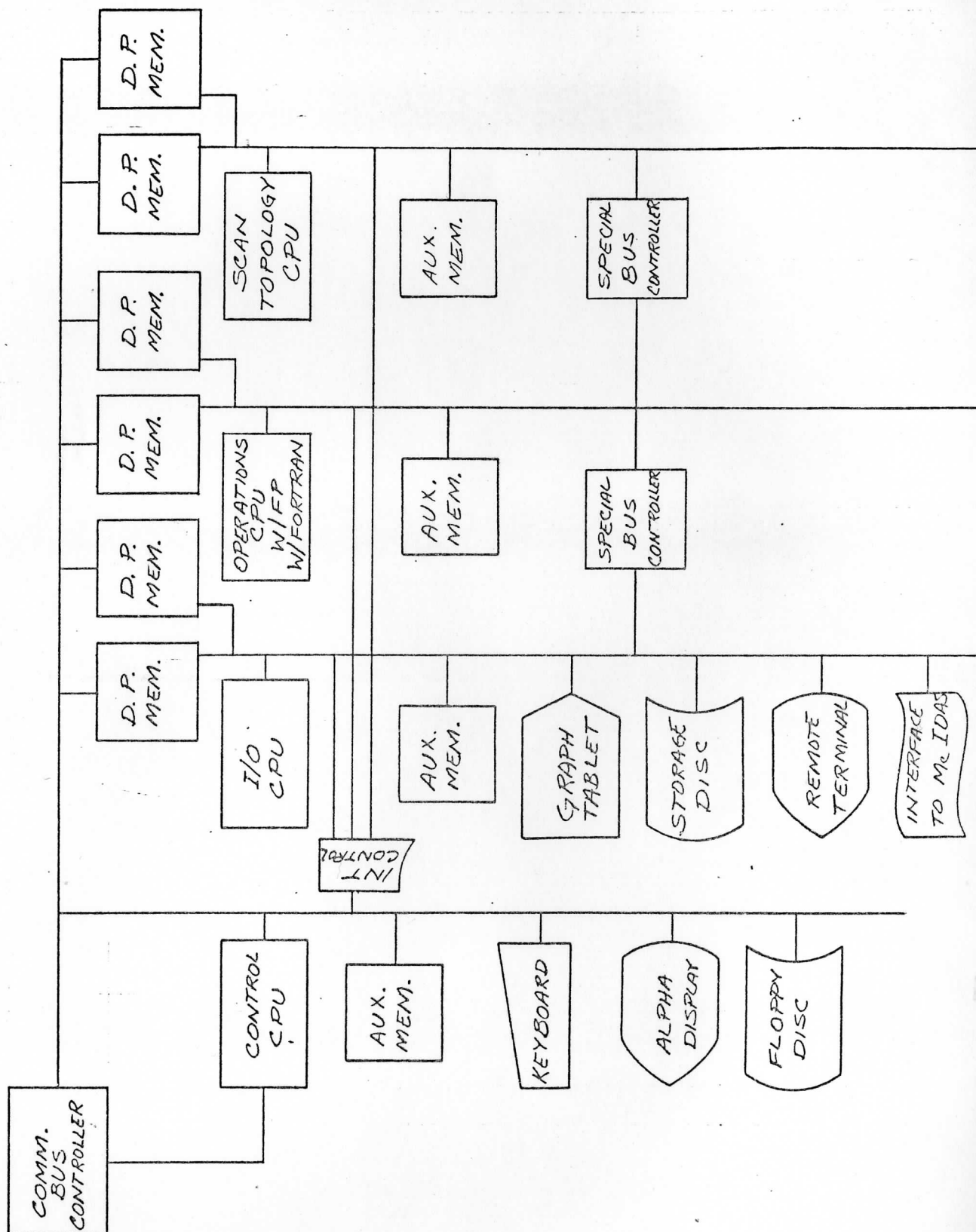


FIGURE II-5



d. Approach #4: Approach #2 could be modified by using only one dual port memory per subordinate processors as in Figure IV-7. In this case the interprocessor transfer further degrades the system in one of two ways. If the subordinate processor is operating within its dual port memory, its processing is degraded and perhaps precluded by the cycle stealing of the transfer. On the other hand, if the subordinate processor is executing within a dedicated memory, some overhead would be incurred when it moved its data to that memory.

e. Approach #5: Each of the computers could be fitted with conventional single port memory and a standard DMA channel as in Figure IV-8. The communications processor would then have a second back-to-back with each of the other processors. Interprocessor transfer from B to C would then be a two-step process. The first step would be a DMA transfer from machine B to the communications processor which would degrade that machine's performance.

f. Approach #6: A special DMA processor could be built as shown in Figure IV-9. This option would again assume that each of the subordinate processors has its own DMA. Then a path would be established tying the output of the DMA from one machine into the input to the DMA for another. In this case a transfer from machine B to machine C would be a one-step process which would degrade both machines simultaneously but would not require the communication processor to have its own storage.

2. Crosspoint Memory Approach

Since the nature of the data processing is more one of manipulation and formatting rather than actual processing, a method of memory pooling was investigated, as shown in Figure IV-10. Each processor would have its own dedicated memory in addition to the access to the common memory. Address space and access are potential items of contention and were studied in detail. A configuration which evolved is a CROSSPOINT System very similar to the HYDRA

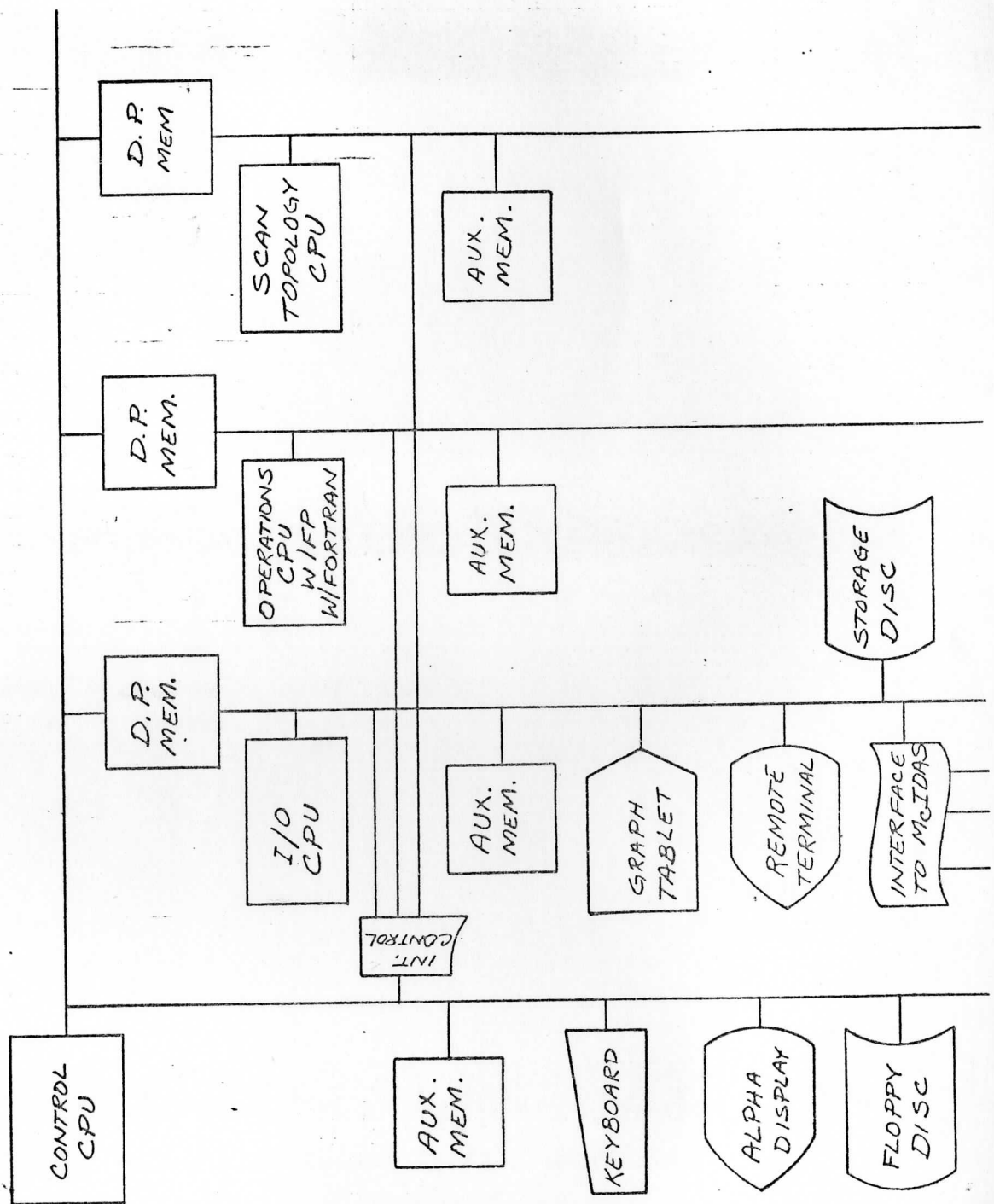
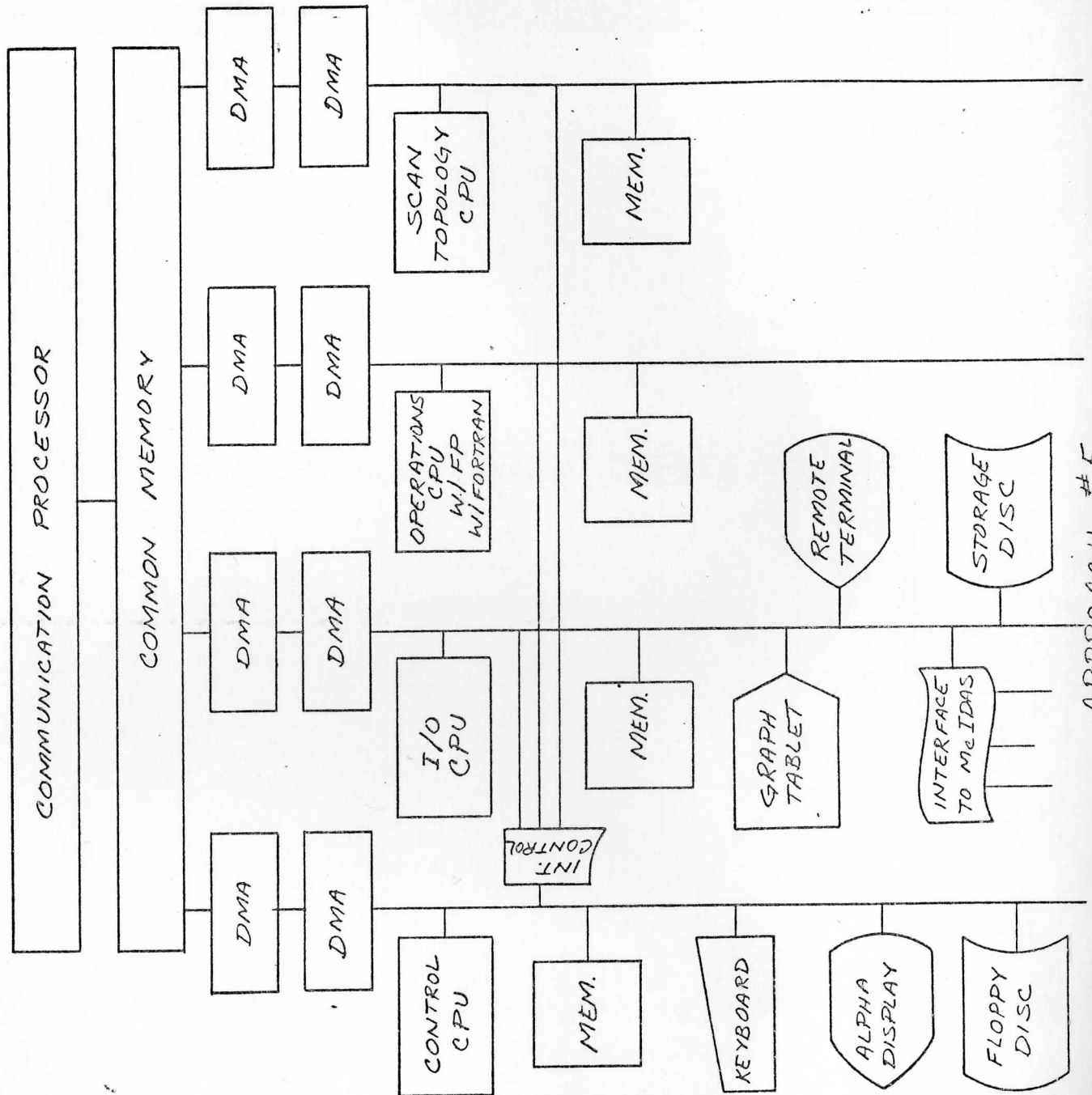
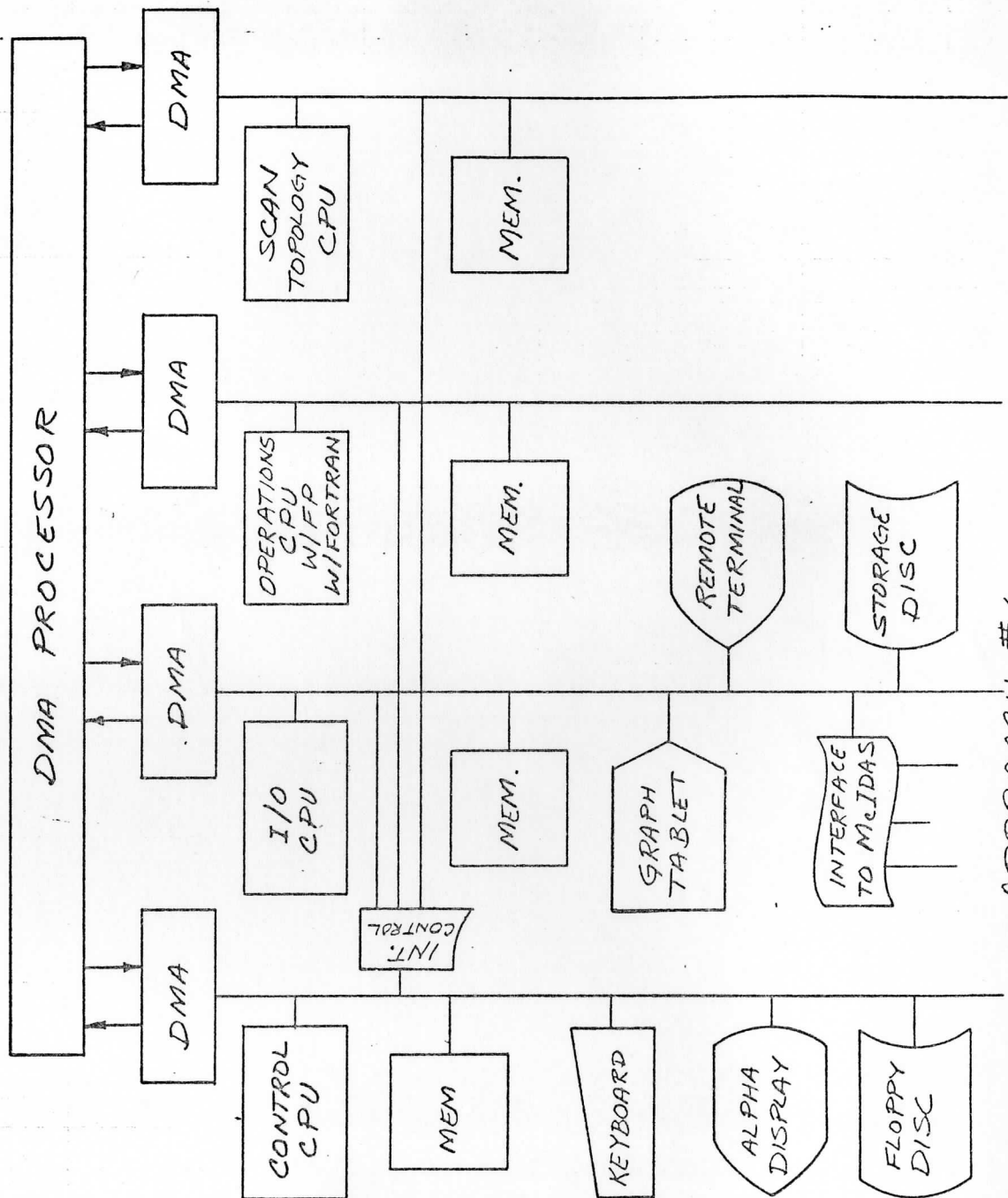


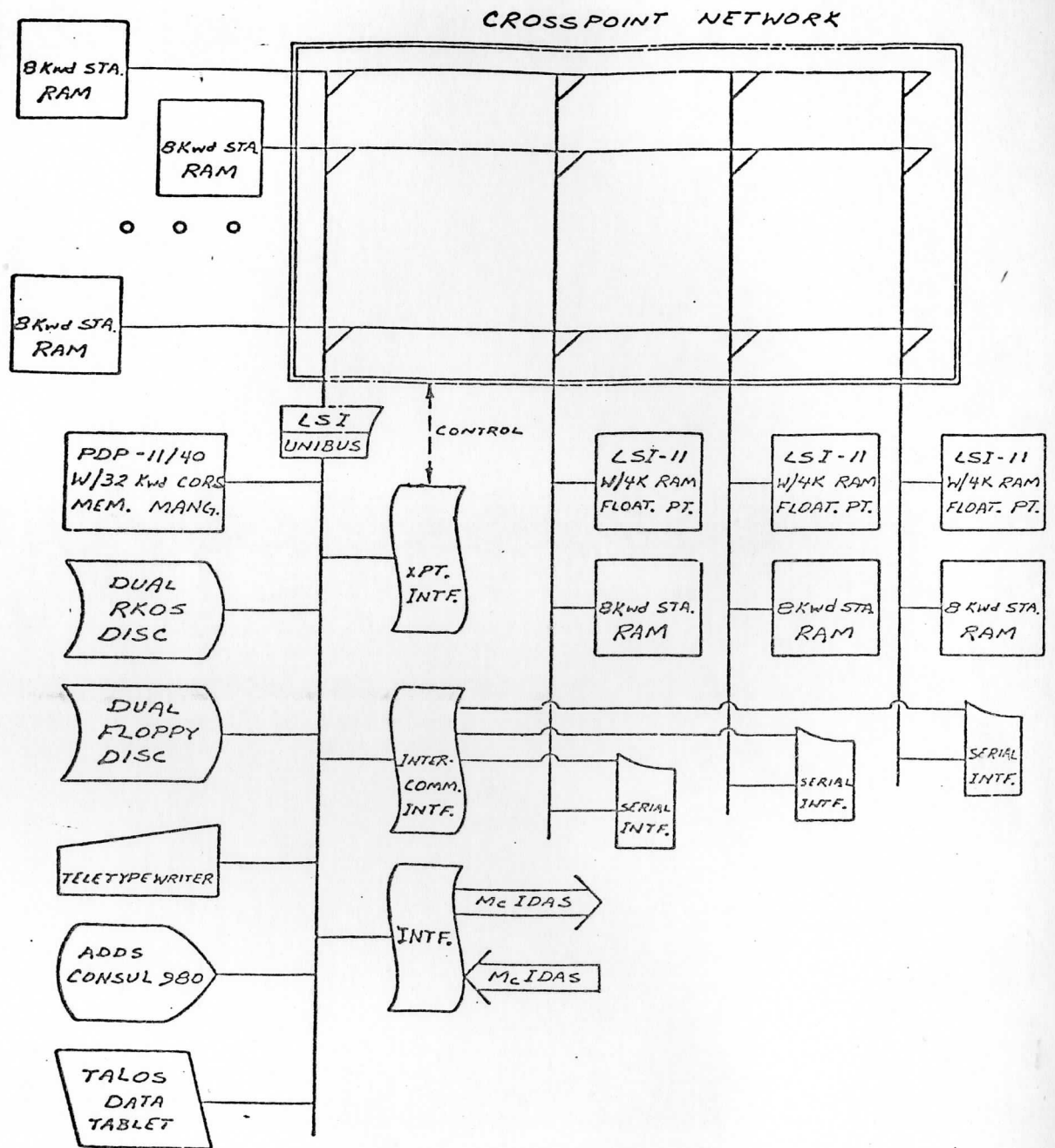
FIGURE IV-7



APPROACH #5



APPROACH #6
FIGURE IV-9



IVAM CROSSPOINT SYSTEM
FIGURE IV-10

system used at Carnegie-Mellon University. The key element in the system is the crosspoint switch, a multi-element switch which has the capability to connect any processor to any memory block. The protocol of switching would allow any number of memory blocks to be connected to any one processor, within the address space of the processor, but only one processor can be attached to any memory at any point in time.

F. COMPARISON OF THE TWO APPROACHES: COMMUNICATION BUS VS. CROSSPOINT

1. Communication Bus

With the Communication Bus there is a single limited bandwidth communication channel through which all interprocessor data communication is routed. This communication consists of physically moving data from one computer to another. This transfer:

- 1) takes time
- 2) ties up memory on both processors from the moment the transfer is requested until it is completed
- 3) may require a dedicated processor
- 4) may become a bottleneck as the communication path saturates and processors have to wait for their data

The Communication Bus can be implemented in several different ways depending upon what is bid. The difference is the bandwidth of the Communication Bus and the amount of processor power required to drive it.

a. Options

1) **Software Control of Transfer:** The data is transferred a word at a time under program control requiring 6.2 microseconds per word, yielding a bandwidth of 161 kilowords/second. Assuming 5000 words of data and/or program to be transferred per program step, the system can handle 32 transactions per second.

2) Firmware: This option is available only with certain manufacturers and uses microcode instructions to control the transfer. The speed of the microcode can be estimated at 2 microseconds per word or a bandwidth of 500 kilowords per second. Assuming 5000 words to be transferred per program step, this system could handle 100 steps per second.

3) Hardware: If a hardwired block transfer controller is made for the bus, the communication bus becomes completely independent of the processor bus and thus frees a processor to do useful work. The speed of the transfer is limited only by the speed of the memory and should be better than 1.4 microseconds per word or 714 kilowords per second. The performance of this alternative is independent of the processor speed. Assuming the need to transfer 5000 words per program step, this alternative could handle 143 transactions per second. The Communication Bus slows down the system as soon as there is a processor which is held up because another processor is using the Communication Bus. This situation will occur long before the Communication Bus itself is in use 100 per cent of the time. In fact, as soon as bus utilization reaches 20 per cent, some contention is to be expected. The bus can be considered saturated as soon as its utilization climbs above 50 per cent. The problem is alleviated somewhat because only three processors would be doing work and thus generating request for transfers.

No one can say for sure that the Communication Bus will saturate at any particular load level because software could be designed to minimize that danger. The answer hinges on the ratio of transfer and processing. If the amount of time required to input to and output from a module is small relative to the time spent processing within the module, then communication would not be too great a problem. For some modules already coded, the ratio of processing to transfer is about 25 to 1.

Use of microcode instructions involves a considerable amount of additional programmer time and the additional processor speed gained must be weighed against the development time of the project.

2. Crosspoint

With the crosspoint system data is not physically moved from one memory to another; instead the memory containing the data is reassigned to a different processor, an action requiring only a few microseconds. Obviously switching an entire memory is faster than moving its contents a word at a time. With 10 memory modules of 8K each which can be completely reassigned in less than 50 microseconds the theoretical bandwidth of the system is:

$$\frac{1,000,000}{50} \times 10 \times 8000 = 160,000,000 \text{ words/second or } 32,000 \text{ transactions/second}$$

While it is unlikely that the processors could make meaningful use of this capability, it is clear that system performance will not be limited by the interprocessor communication paths.

G. ARGUMENTS BY CONFIGURATION

1. Communication Bus

a. Advantages

Off-the-shelf components

Likely to be standard product in the future

Likely to decline in costs

Simple and reliable

Easily expandable until the bus saturates

Software, firmware and hardware are not mutually exclusive; the optimum mix of the three can be changed later if desired

b. Disadvantages

Transfer data slowly

Limited bandwidth transfer channel may saturate and limit performance

c. Unknown

Cost: Not currently available as a system; prototype copy must incorporate one-time costs, future copies' cost unknown

Programming: The protocols for communication bus control must be carefully worked out to allow optimum use of bus transfer.

Avoidance of bus saturation would require careful vigilance on the part of the implementers. It will always be an important constraint which could kill the system. With the crosspoint this issue is avoided completely; the designers will spend their time working on problems that are more uniquely IVAM.

2. Crosspoint System

a. Advantages

Extremely fast, transfer time transparent to the system

Processors not involved in transfer are available for processing while other processors are transferring

Control of the system will not require a dedicated processor

A working example (at Carnegie-Mellon) exists

Development prototype will perform much like a production model

Software development can progress rapidly because contention for priority on the data channel does not exist

b. Disadvantages

The multitude of cables needed for switching may cause reliability problems

Not likely to decline in cost because major cost is mechanical--cables, connectors, etc.

Expansion difficult because number of switches grows as the product of the processors and memory blocks

c. Unknown

Large blocks of switch hardware may be expensive, depending on hardware vendor choice

3. Summary

There is clear performance advantage to the crosspoint system. Earlier concern that costs might be prohibitively high has been dispelled.

H. HARDWARE PROCUREMENT

A hardware specification was prepared for the multi-processor system, outlining the requirements for system components separately. The Request for Bid asked for bids on any single item or combination of the following:

- 1) Entire system with dual port memory
- 2) Entire system with single port memory
- 3) Individual processors or peripherals within the system
- 4) Any other option which the bidder felt might meet the requirements

Also required in the bids were inclusion of any additional cost associated with use of the equipment, such as documentation, software, software licenses, maintenance, special interfaces and other support.

Requests for bids were sent to over 50 manufacturers of computers, peripherals, and systems. A copy of the RFP is attached as Appendix A.

1. Bid Evaluation Process

Bids were received from 20 companies, covering complete systems, peripherals and components only. The bid evaluation was done in two steps, technical evaluation with no knowledge of cost, then a second step with cost and performance trade-offs considered.

a. First Step - Technical Evaluation

A technical evaluation committee was named, made up of two system

programmers, one applications programmer, one electronic engineer, and one systems expert from the Computer Science Department. Technical evaluation was based on the computer system in the proposed configuration or application of individual computers proposed to specific configurations of crosspoint or communication bus. Seven manufacturers of processors bid and the bids were separated into two groups: bids generally meeting requirements and bids not technically acceptable. Several of the bidders proposed more than one approach and each of those bids were evaluated separately. An evaluation form was used for numerical scoring of the bids and is attached in the following pages.

b. Second Step - Cost versus Performance Evaluation

Of the top four configurations bid, three were from Digital Equipment Corporation (DEC), using PDP-11 type machines. The Bus Window is a new system development from DEC, not yet released, but being developed for the telephone company in Canada. It is a data communication bus similar to our design but uses "time slices" on the bus for the transfer of data. The bus has a rigid use structure in terms of time multiplexing and would not allow optimum use of the bus for data transfer as needed for IVAM. In addition, delivery of a Bus Window system from DEC in the near future seemed doubtful and could delay the development of IVAM software even further.

The real trade-off was between the Data Communication Bus system using DEC computers and the Crosspoint Switch System using DEC computers. Using the standard PDP-11's, the cost of the crosspoint is high because the cost of hardware component compatible with the DEC UNIBUS is high. However, using the new DEC microprocessor board, LSI-11, the cost drops dramatically. The basic processor costs less than \$1,000 each, low-cost memories are available, and the mechanical hardware associated with the LSI-11 is low cost. The

I. PROCESSOR EVALUATION (40)

WEIGHT

A. Processor (30)

		Data 100	Computer Automation	Harris Corporation	Digital Equipment	Bolt, Beranek & Newman	Modular Computer System	Varian Data Machines
Registers	1.5							
Addressing Modes	2.1							
Address Space	1.5							
Hardware Stacks	1.7							
Byte Manipulation	1.8							
Auto Increment/Decrement Registers	1.5							
Memory to Memory Instructions	1.5							
Hardware Multiply/Divide	.5							
Floating Point	.5							
Reentrant Code	1.7							
Relocatable Code	1.7							
Memory Management	1.8							
Microprogramability	1.3							
User/System Mode	1.2							
Stack Overflow/Underflow	1.4							
Illegal Instruction Trap	1.2							
Memory Protection	1.9							
Start-up	.7							
Throughput	1.5							
Consistency	1.8							
Instruction Speed	1.2							

30

B. Software

General Category - Including

Real-time

Multiprogramming

Multiprocessor Operating System with:

File System

Editor

Assemblers

Compilers

Debugging Package

WEIGHT
10

Data 100	Computer Automation	Harris Corporation	Digital Equipment	Bolt, Beranek & Newman	Modular Computer Systems	Varian Data Machines

II. CONFIGURATION EVALUATION (60)

			Quad Port Memory	Shared Mem. Pairs						
			WEIGHT							
A. Ease of Development (20)										
1.	Amount of in-house effort to complete configuration	(Policy)								
2.	Ease of start-up (4)									
	Multiprocessor systems software	2.0								
	Hardware - time availability	2.0								
3.	Complexity	4.0								
	Components in system can be treated in consistent manner									
4.	Protection & Vulnerability (6)									
	Standard Memory Protection	2.3								
	Configuration established protection	3.7								
5.	Support (6)									
	Listings	1.3								
	Hardware guarantees for completeness	2.1								
	Software updates	.9								
	Maintainability	1.7								
		20								
B. Product (Production) (20)										
1.	Reliability (3)									
	Fail soft	1.5								
	MTBF	.9								
	MTTR	.6								
2.	Flexibility (4)									
	Reassign functions	1.8								
	Reassign peripherals	.9								
	Configuration Change	1.3								
3.	Ability to implement production Model in increments	5								
4.	Expansion Capability (3)									
	More Processors	1.2								
	More Memory	0.9								
	More Communications Paths	0.9								
5.	Total Address Space	2								
6.	Adapatability to Expected Hardware cost	3								

20

C. Efficiency (20) WEIGHT

1. Throughput (20)

Transparency of Communication 3.5

CPU Degradation 5.0

Processor Bus Degradation 2.5

Communication Path saturation 5.0

Memory space tied up during communication 2.0

Systems setup overhead for a communication 2.0

20

(see Below)

Quad Port Memory

Shared Memory Pairs

combination of a PDP-11/40 for system control and main processing with LSI-11's for subordinate processors is by-far the most cost effective purchase of processing power. Two very powerful software operating systems, RSX-11M and RSX-11S, are available on the processors so that the crosspoint operating system primitives can be developed and implemented later as the system progresses.

TABLE IV-1
PROCESSOR/CONFIGURATION FINAL RANKINGS

	<u>Processor</u>	<u>Configuration</u>	<u>Average</u>
1.	DEC	Crosspoint	85.93
2.	Data-100	Crosspoint	84.08
3.	DEC	Comm. Bus	80.63
4.	DEC	Bus Window	80.02
5.	BBN	Multiple Mem. Bus	78.34
6.	DEC	Multiplexed Mem. Bus	77.85
7.	Data-100	Multiplexed Mem. Bus	76.20
8.	DEC	DMA Approach	73.63
9.	DEC	Shared Mem. Pairs	73.33
10.	Data-100	DMA Approach	71.98
11.	Data-100	Shared Mem. Pairs	71.68
12.	Harris	Single Processor	67.225
13.	Mod Comp	Quad Port Mem.	64.445
14.	Varian	Shared Mem. Pairs	62.885
15.	Varian	Quad Port Mem.	62.175
16.	Computer Automation	Single Processor	58.485

I. IVAM DEVELOPMENT SYSTEM

The Space Science and Engineering Center is currently putting together the prototype system on which the IVAM capability will be developed and demonstrated. This equipment falls into three categories:

1. The Multiprocessor Assembly which is the Control and Image Generator Subsystem of the prototype system and which was purchased under the IVAM contract.
2. McIDAS and other equipment which is being used temporarily to support IVAM development (but which is not part of the deliverable system).
3. Development tools purchased by IVAM which support the software development but will not be part of an operational system.

In addition there are a number of interfaces fabricated in-house which tie the parts of this system together.

The major elements of the system are shown in Figure IV-11 and will be described separately below.

1. Control and Element Generator Subsystem

The Control and Element Generator Subsystem consists of the following components:

PDP-11/40

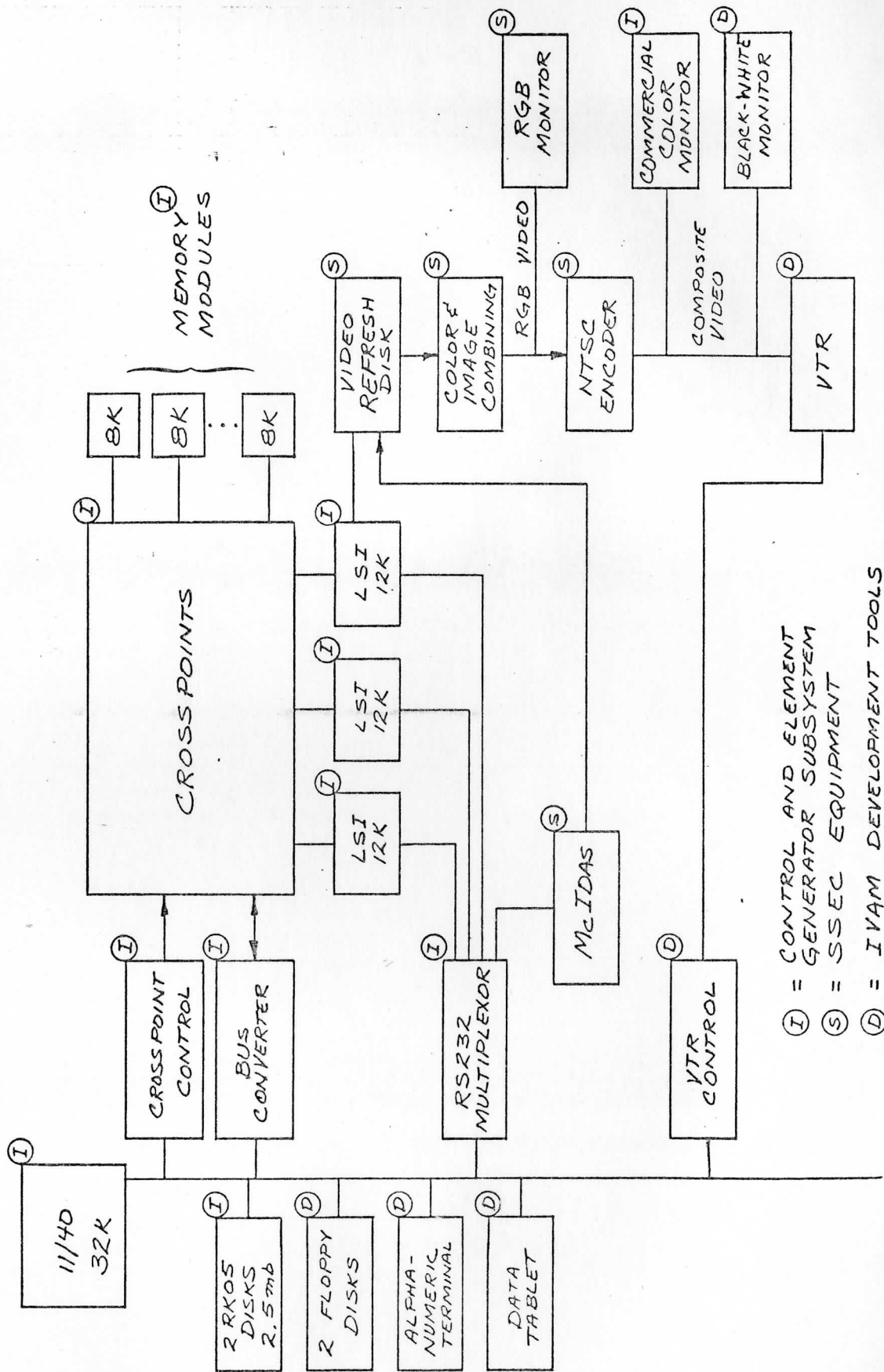
Three LSI/11 microcomputers

Interprocessor Communication System

Crosspoint Memory System

a. PDP-11/40

The Digital Equipment PDP-11/40 minicomputer is equipped with 32K core, memory management, two 2.5 megabyte disks, a sophisticated operating system



- (I) = CONTROL AND ELEMENT GENERATOR SUBSYSTEM
- (S) = SSEC EQUIPMENT
- (D) = IVAM DEVELOPMENT TOOLS

IVAM DEVELOPMENT SYSTEM
FIGURE IV-11

and software development tools. Its function is to support software development and to serve as the Control Processor for the production system. The 11/40 will run the IVAM operating system which will assign tasks and allocate resources for the LSI/11's as well as the 11/40.

b. LSI/11's

The three LSI/11 microcomputers each have 12K of dedicated memory and are code compatible with the 11/40. The main difference is the number of interrupt priority levels and the fact that the address and data bits are multiplexed on the same bus. Each LSI can access one of several additional 16K words of memory at any point in time through the crosspoint system. The LSI's will be used to execute modules under the direction of the Control Processor. Until special hardware is built one of the LSI's will be dedicated to the Fill-in process and will be interfaced to the video refresh disk.

c. Interprocessor Communication System

All interprocessor command and status information is communicated between each LSI and the 11/40 over RS232 lines at 9600 baud on an interrupt basis. Each of the LSI's has a single serial channel while the 11/40 has a 16 channel multiplexor board which it also uses to communicate with McIDAS and to control the alphanumeric terminal.

d. Crosspoint System

The passing of data and code between processors is accomplished not by block transfer but by switching the memory so that the receiving processor can access it. Since the entire contents of the memory are switched simultaneously, the apparent transfer rate is very high.

The IVAM crosspoint system is designed to allow each of up to 8 processors to access any of 16 memory modules. Currently the IVAM system has four pro-

cessors and ten 8K memory modules. At any one time a given memory module can be accessed by only one processor. However, a processor can simultaneously access as many memory modules as its address space allows, two for each LSI and four for the 11/40. The crosspoint system is under the complete control of the 11/40. It allocates crosspoints to processors and determines when to switch them to other processors. It determines whether a memory is read-only and what addresses it will occupy on the processor it is connected to.

During initial development, the multiprocessing system will perform every step required to generate full video images. It will generate complete 485 line x 672 pixel image a line-at-a-time in software. Each line will be transmitted to the digital refresh disk as it is generated. In the finished prototype the fill-in process will be hardwired. The output of the Multiprocessing system will be graphical image element - TV raster line intercept sets which provide in compacted form all the information required to generate the full image.

2. Center Equipment

Currently, part of the final IVAM system is being simulated by the following Center equipment:

- 1) The McIDAS system is simulating both AFOS and CDDS. It acts as a data source for IVAM sending it Service A, Service C, and radar data. It also sends the satellite image to the digital refresh disk.
- 2) The digital refresh disk is capable of storing twelve full raster, 5 bit video images in digital form. It also contains a two bit semiconductor video refresh buffer. This terminal can simultaneously readout, colorize, and to some extent combine the two 5-bit disk images with the 2-bit RAM image.

- 3) Black and White TV monitor: During development it will be necessary to view the IVAM output on both Black and White and Color commercial TV monitors to insure that the IVAM output is compatible with both displays.

3. Development Tools

The following devices are being used during the IVAM development effort and will not be part of the final system, except for item 4 :

- 1) Data Tablet: This is a 22" x 22" tablet with 100 points per inch resolution. It will be used to generate graphic input data, weather symbols, and complex parameters.
- 2) Dual Floppy Disks: These disks are being used for offline storage of programs and test data.
- 3) NTSC Encoder: The output of the refresh disk is converted to RGB video which must be passed through the Center's NTSC encoder which produces a broadcast compatible color video signal which can be viewed on a standard color monitor. This device is important because its bandwidth characteristics strongly affect IVAM color relationships.
- 4) Video Tape Recorder: This is a Sony cassette recorder with full edit capability. It has the ability to synchronize its movement with an external video source. A controller is being built to allow the computer to number every frame randomly on the tape. The computer will have full control of the recorder's tape movement and read/write controls. When the Video Assembler subsystem is completed the recorder and controller will become the video tape library for the prototype system.

4. Performance

The McIDAS digital refresh disk represents a major throughput bottleneck during development because it can input only one digital scan line every 1/30 second. This means that it takes 16 seconds to update a frame. Since the disk holds only 12 frames, that is the greatest number of different frames that can be shown in a single continuous output. Since each of these frames can be repeated as many times as desired, the disk can maintain a continuous sequence of video still frames for a minute or more. However, if the images are changing at the peak rate of 5 new frames per second, one loading of the disk provides only 2.4 seconds of output.

Because of this temporary limitation, the Sony cassette recorder is needed to produce the 5 minute test tapes that will be required for evaluating the IVAM product. The video tape recorder allows IVAM to generate short sequences at the rate the disk allows and then to assemble them into longer presentations on the tape.

As a further consequence of using the refresh disk, one of the LSI processors is temporarily dedicated to the fill-in process and to transmitting the scan lines to the disk. Later in the program, both the fill-in process and the output staging will be hardwired, freeing that processor for other work. At that point the multiprocessor system will be able to generate outline messages at the rate required to sustain 20 minutes of video output out of every hour as required.

V. OUTPUT DISTRIBUTION STUDY

In the previous report, the major emphasis on output was to provide full video, ready-to-use, to both the broadcast TV and the cable TV operators. The distribution concept for cable companies, using 5 minutes per hour or 1/2 hour for transmission, may not be optimum for some companies. Several other possibilities are being studied, such as transmitting update information only, at lower bandwidth, and then inserting the new information into the output system at the proper time.

A summary of distribution to network broadcast stations is given below, followed by discussion of several methods for low bandwidth transmission to cable companies.

A. BROADCAST NETWORK DISTRIBUTION

Twenty-four WSFO's were identified previously as candidate locations for installation of the IVAM system. Those suggested locations were based on:

- 1) Distribution routing of the major TV networks
- 2) Geographical area of service for local WSFO's
- 3) Area known served by local TV stations

There has been no effort during the past year to refine the selection of WSFO's, since such an effort must be done in conjunction with the NWS personnel in terms of long range planning for use of the WSFO's and addressing the specific problems for each location in terms of operation requirements and local routing of video feeds to the local TV stations.

The long-line system which serves the four TV networks is adequately described in last year's report. Summary characteristics of those lines are:

- 1) Leased by each network annually for 24 hour per day use
- 2) Dedicated lines for national coverage

- 3) Generally fed in a single direction but many links could be reversed if necessary
- 4) Back-up channel is available
- 5) Each of the four networks operate independently but all follow the general rule of:
 - a) periodic blank times on the feed of 15, 30, 60, 72, or 96 seconds, at least every 1/2 hour
 - b) local and regional feeds of commercials
 - c) open-form news feed to affiliated stations during late-afternoon daily
 - d) generally blank time between midnight and 6 a.m. every day

The method of distribution for network stations described in detail in the previous annual report has met with wide acceptance. The feed of individual segments at standard TV rates on the network during the periodic blank times has been discussed with station operators and the interest is high. One of the problems cited in the previous report was that small local links were needed in some cities where the WSFO and the local network terminal were not proximate. In San Francisco, one of the problem cities, the situation was discussed in detail with personnel from the WSFO and from the NBC and CBS TV stations. The IVAM program was described and the sample video tape demonstrated. Mr. Berryhill, Chief Engineer of KRON-TV, summarized the attitude of all when he said, "tell us when the IVAM output will be available and we will see that the local inter-connection from the WSFO is made two weeks before you are ready."

In all discussions with network affiliate stations, the concept of feeding individual segments during the network blank times was positively received. Insertion of regional or local weather segments into the network for use by "downstream" stations was also discussed and found to be simply a matter of working out the details as is done regularly for regional commercials which use the same scheme.

B. DISTRIBUTION TO CABLE TV AND INDEPENDENT BROADCAST TV STATIONS

While network broadcast TV reaches 95 per cent of the U.S. population, cable TV is a very important medium because it has the capacity for supplying a much greater volume of information. Even considering the growth that has occurred in the past few years, cable TV is still in its infancy. The major obstacles of licensing and other legally-oriented restrictions have been passed and cable TV has before it a major growth period. Cable TV now serves almost eleven million homes in the U.S. today or 15.3 per cent of the total U.S. population, and its cables pass nearly 30 per cent of the homes with TV sets in the country. This means that one of the major costs for cable companies, laying cables, is behind them for a large market and increasing subscribership is less costly for the future. As reported in the 15 March 1976 issue of New Yorker, Mr. William Donnelley, a Young and Rubicam executive, recently pointed to what he calls, "a magical and critical number: thirty per cent penetration for cable TV." By 1981, he estimates, thirty-three per cent of all TV sets will be wired for cable. And then he pointed out that major advertisers left radio for TV as soon as thirty per cent of the nation's households had TV's, and that advertisers switched to color commercials as soon as thirty per cent of the Nation's TV owners bought color sets.

According to Mr. Donnelly, "In 1980 cable TV will have the bone structure for a quantum leap followed by an effectively wired nation a decade later."

Cable TV is an ideal medium for the dissemination of IVAM data since the number of channels provided by cable TV allows dedication of one channel to full-time weather advisory service. Many municipalities require the local cable TV to provide continuous weather coverage 24 hours a day on a dedicated channel.

A general plan for distribution to cable companies, like the general plan for network broadcast TV, is not possible. Instead, distribution to cable companies must be addressed on an individual basis. Some cable companies already service the area in which a WSFO is located. Connection for them is a simple feed to the already-existing cable. Theta Cable of California serves over 80,000 subscribers in the Santa Monica area surrounding the Los Angeles WSFO and is ready to "hook-on" whenever IVAM is available. Theta Cable is a relatively large company and would be willing to invest in the special equipment required for a continuous IVAM channel. Maximum dollar figures were not named during our interview with Theta executives, but the mention of as much as \$20,000 to \$30,000 was received calmly.

However, not all cable companies are located so conveniently and other methods of distribution must be found.

C. LOW BANDWIDTH TRANSMISSION CAPACITIES

When considering data transmission at less than video rate, there are four standard options available:

- 1) 4.8Kbps switched phone line
- 2) 9.6Kbps dedicated phone line
- 3) 19.2Kbps conditioned phone line
- 4) 56Kbps conditioned phone line

The standard "dial-up" phone line is sufficient for transmission rates up to 4.8Kbps. The dial-up line has been used up to 7.2Kbps but the reliability is low for such use on a regular basis. Conditioned lines are available up to the 56Kbps rate, guaranteed bandwidth, with the appropriate conditioners at each end.

The use of satellite for relay of video data has been growing steadily, particularly for providing source programming to cable TV stations. There are

two satellite systems used, the Western Union with 12 channels, and the RCA satellite with 24 channels. There are currently about 90 earth stations receiving satellite relayed video. Each ground station feeds 3 or 4 cable TV systems, and an estimated 200 receiving stations are expected to be operating by late 1977.

There is a tendency for the cable companies to interconnect in areas surrounding the receiving stations with the effect of an informal sort of networking, or "netting". The likelihood of transmitting IVAM data via satellite is low, since IVAM's weather for cable is emphasis on the local, and satellite transmission is suited for cross-country communication. However, the netting being developed at satellite terminals provides a strong cable distribution system at the local and regional level. In fact, the local netting of cable coincides with the location of many of the WSFO's identified as IVAM sources.

D. LOW BANDWIDTH TRANSMISSION SCHEMES

A full video image in digital format takes 10 minutes to transmit on a conventional phone line at 4.8kbps. For a 1/2 hour update period, this allows an update of 3 pictures. Assuming an update requirement of 20% of information per hour, this would allow change of 6 out of 30 TV frames. While this seems hopelessly slow compared to the video rates discussed in other parts of IVAM, using good quality graphics to present past and predicted information is a great improvement over the alphanumerics currently in use on cable TV. The display of current data in color on a map makes the "picture's worth 1,000 words" phrase come true.

At little increase in cost, over conventional switched lines, a 9.6kbps phone line is available. The greater bandwidth doubles the possibilities described in the previous paragraph; up to 12 complete frames can be trans-

mitted every hour. The larger number of available frames would allow slow step, animated sequences, as well as static graphic presentations.

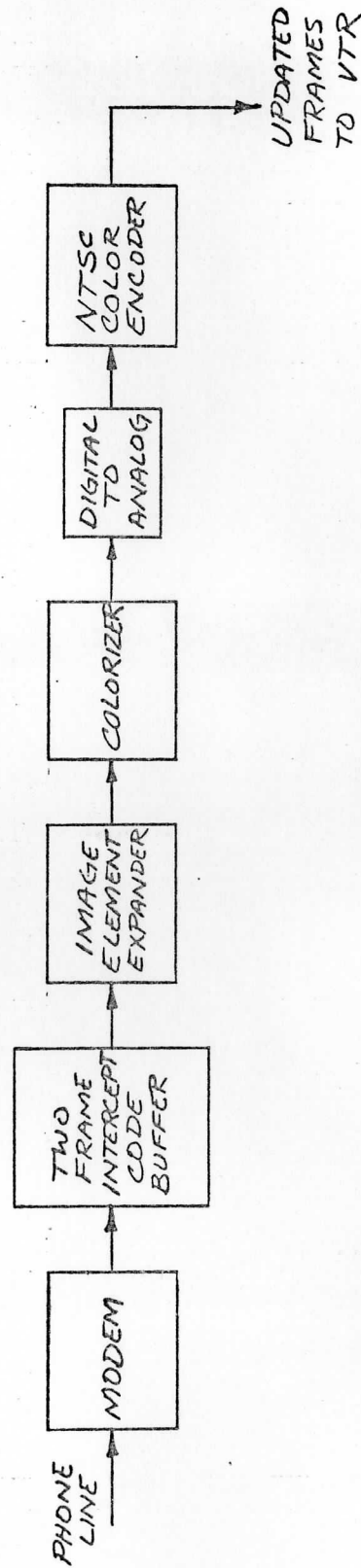
The next option, at a significant increase in cost, is to install a 19.2kbps phone line. This line would transmit 24 frames per hour with a greater range of animation available as well as more rapid response to changing weather situations.

Equipment at the receiver end must be capable of accumulating the digital data, storing it until the proper frame on the looping program tape reaches the record head, then writing the accumulated data into the proper frame on the tape, repeated as necessary. The digital data would output from the accumulator at video rate, be converted to RGB by a prewired colorizer, then converted from digital to analog and color encoded to NTSC standards before being recorded.

The equipment needed consists of storage space for two complete frames of digital data (the most expensive part of the system, but within reason since they are first-in, first-out configured), a simple colorizer with prewired color assignment, digital-to-analog converter, and an NTSC color-phase modulator. Estimated cost for such equipment is about \$30-\$40,000, the major cost being the two full frame storage buffers.

An alternative scheme for low bandwidth transmission is to send data in the image element-TV line intersect format, prior to being expanded into full TV frames. The intersect format requires 1/10 the data volume of fully expanded scan lines, which, for the same transmission line, increases the information available to a cable station by a factor of ten. This system is shown in Figure V-1. The two major advantages this system has over sending fully expanded images are:

- 1) Increased capacity for updating segments
- 2) Decreased capacity required for buffer storage



ALTERNATE SCHEME FOR RECEPTION OF LOW BANDWIDTH IVAM TRANSMISSION

FIGURE V-1

This system does not include a temporary storage buffer which can hold frames until the looping tape player (assuming about a five-minute cycle time) reaches the frame to be replaced. It would appear that it should be possible to time the transmission of new frames to coincide with the position of the looping tape, but we have not yet analyzed this distribution mode completely. The use of an analog video recording disk instead of a looping tape recorder must be studied carefully.

If we take the 9.6kbps line as the basis for a low bandwidth distribution system and use the intercept coding scheme, the average frame rate will be about 120 frames per hour. The characteristics of the presentation which this capability can support will be explored in greater detail during the next year. The possibilities are very attractive, and receiving end equipment costs probably can be kept quite low.

Equipment needed for this system is basically the same as for the fully-expanded frame data, but with the requirement for buffers reduced to 1/10 and the addition of an Expander. Cost of the buffers will drop from \$20,000 for the full frame buffers to about \$5,000 for the 1/10 frame buffers. Additional cost of the Expander amounts to about \$5,000, resulting in a projected cost for this configuration around \$25,000 total. The control functions necessary for this receiving system are straight-forward and may be simply prewired. However, it may be less expensive to use a micro-processor and our cost estimate is based on that alternative.

The extent to which low bandwidth transmission using intercept coding will be used depends to some extent on the acceptability of presentations which do not include unmodified satellite images. It is possible to process satellite pictures to produce smoothed contours of cloud brightness and to reconstruct them at the receiving end. The result is a simplified cloud

cover graphic which includes most of the large-scale information present in the original image, but small-scale details are eliminated and it does not "look like a picture". The acceptability of these cloud graphics will be evaluated during the next year. The techniques and software necessary to produce the cloud graphics automatically are nearly all in hand at SSEC but need to be assembled and tested. If cloud graphics prove to be acceptable or even preferred to the original pictures for TV presentations, then the designs for the IVAM Video Assembler subsystem will also be changed and simplified. This is discussed further in Section VII.

VI. PRESENTATION TEST AND EVALUATION

With the very recent delivery of the development hardware, there has been no opportunity to test actual IVAM output. Instead, we have concentrated on two subsets of the output: (1) a demonstration of some of the software modules which have been coded and tested on the computer, and (2) detailing some of the specifications for picture quality.

A. SOFTWARE MODULE DEMONSTRATION TAPES

Software modules have been coded which are used internally in the IVAM system as part of the video output image. These modules produce maps at different scales and projections, produce contour lines from point data on the map, and produce several other basic processes used in the preparation of the image output. These images are not visually suitable for final IVAM output because we are using McIDAS display equipment which is not capable of being adjusted for color compatibility, optimum labeling, optimum contour step size, etc. However, the output of these modules does demonstrate the ability to: create contour images from point data, produce maps for any part of the U.S. at any scale or projection desired, apply contours to any map projection, and produce accurate overlay of surface observation data on satellite data. The modules and their functions are listed in Table VI-1.

B. PICTURE QUALITY STUDIES

The ultimate test of the IVAM system is the public's reaction to the pictorial output. To establish the levels of excellence which the IVAM system must meet, we are conducting a picture quality study. This study is on-going and probably will continue long after IVAM is in full operation. Until IVAM can address the public on a regular basis, we must depend upon the advice and criticism of experts. The AMS Advisory Panel, all meteorolo-

TABLE VI-1

MODULE LIST

<u>Module Name</u>	<u>Function</u>
RTSMS	Ingest GOES-A satellite data
RTSUCA	Ingest Service A data to the data base
LDCNTV	Load a SMS satellite picture onto the video display
PLOT	General purpose 4-bit WRRRM refresh plot package
SATEAR	Satellite Navigation Transform Routines
GRDFILE	Continental and State Boundary File
SELGRD	Select a set of points from GRDFILE to generate a base-map (in either satellite or Mercator projection)
GETCUR	Reads the cursor position and size
SELDATA	Selects a Service A Data Set
GENDATA	Generates a derived Data Set (Equip. Potential Temperature, etc.)
CTOUR	Generates Isopleths from a Data Set
FILLIN	Contours (fills in) an image from a Data Set (from CTOUR <u>or</u> GETCUR)
ANNOT	Generates Alphanumerics on the 4-bit WRRRM
ENHOOL	Generates a color set

logists expert in the communication of weather information, monitors progress of the study and advises us at semi-annual meetings.

IVAM must operate under constraints inherent in the U.S. television systems and those imposed by the necessity to limit cost in the IVAM system itself.

1. Compatibility with Commercial TV

Color information in the IVAM system is created and processed in the RGB convention because it is cheaper and easier to do so. A greater range of color and brightness possibilities exist in the RGB system than are allowed by the NTSC system used by broadcast and cable TV in the United States.

a. Bandwidth Limits of the NTSC

The phase modulation scheme of NTSC encoding exerts two limits on the system capability to produce color:

- 1) Many colors attainable in RGB are not attainable in the NTSC system
- 2) Adjacency of certain colors causes "color crawl"

The RGB system makes use of a large bandwidth by virtue of its three separate channels, which allows a broad range of combinations of full range red, full range green, and full range blue. With the NTSC encoding system, the full information must be "squeezed" through a bandwidth which does not pass the full range of R, G, and B combinations. This slew-limiting generally pre-empts encoding of colors beyond a point determined by the combined values of luminance and saturation. The large choice of colors attainable by combinations of R, G, and B must be limited in the selected color sets to those which can be transmitted on standard TV.

The same bandwidth limit causes "color crawl". When two adjacent colors within a TV scan line require a significant difference in phase angle representation by the color subcarrier, the system does not allow phase shift

quickly enough and the first color "smears" into the adjacent color. The two fields of the TV frame are produced at 180° polarity (for electronic convenience) so adjacent scan lines in the picture smear differently and a scintillation effect occurs.

Hence, in addition to concern about the reproduction of a certain color, the issue of adjacent colors must be addressed. Color crawl increases when the colors used are farther apart on the color wheel, i.e., colors with greatly different phase angles.

b. Black and White Compatibility

When a color signal is shown on monochrome TV, the only distinction between colors (which show as gray levels) is the luminance value. Two colors may have sufficient color contrast but still have the same luminance, and when shown on monochrome TV they will be indistinguishable. To be acceptable for monochrome reception, adjacent colors of a colored image must have significantly different luminance values. The practical limit of number of gray levels discernible on a monochrome TV receiver is ten, but industry guidelines suggest no more than five. These numbers are very restrictive in terms of colors available, unless one adds another level of complexity, the consideration of non-adjacent use of different colors with the same luminance.

c. Line Quality Considerations

The present system on McIDAS draws map and contour lines one pixel (picture element) wide. These lines are acceptable within the wide bandwidth of the RGB system, but the one pixel vertical lines disappear when displayed on the NTSC monitor. Widening the line to 2, 3, or 4 pixels helps, and this capability will be present in the Video Assembler subsystem. Horizontal lines one pixel wide tend to "blink" because they reside in only one field of each frame and are refreshed only 30 times per second. Widening these lines helps,

and will be done, but the optimum line width in TV lines and pixels depends upon the complete graphic. The digital "stair-step" appearance of thin lines can be decreased by "shading" the lines on the outside pixels. Tests of shading and smoothing are planned for next year to determine if needed in IVAM.

d. Other Limitations Due to Use of Conventional TV

Some TV broadcast stations use a "hard AGC" (Automatic Gain Control) which forces all frames to an average brightness. If color choice (and corresponding gray levels) are not properly chosen, color balances and adjacent gray values are shifted up or down in brightness and the picture quality is lost. This effect can be noticed on some local TV stations when playing old films which use subtle lighting effects.

Picture brightness must follow the rules of:

- 1) Picture is viewable with normal ambient light conditions
- 2) Picture does not make large changes in average brightness
from frame to frame, or segment to segment

C. COLOR EVALUATION TESTS

Tests were performed on a standard set of color swatches, COLOR-AID PACK, manufactured by Geller Artist Materials, Inc., to evaluate the RGB values and suitability for TV use. The tests were performed using a calibrated TV camera set-up, with color vector scope and oscilloscope for quantitative readout. AGC variation was balanced out by adjusting each TV picture to normalized gray scale and color balance. Measurements were made of R, G, B values, as well as hue, saturation, and luminance for each color. Many colors were too intense to be used for TV, either in luminance or in the combination of luminance and saturation. The results of the test are attached as Appendix B. Note that many of the yellows and oranges of this set of colors are too intense for TV.

While the color swatch tests provide useful information on the suitability of a particular color for TV, some difference in the color reproduced using the RGB values is likely. The printer's ink used for producing such samples generally provides narrow spectra of wavelength, whereas the TV camera has a wider three color responsivity peaking at different frequencies. In addition some components of ink on a swatch contain components of color outside the wavelength of the camera responsivity.

The establishment of color sets for selection at IVAM output may be created by generating color combinations of the color bar test pattern which are within known allowable saturation levels. By varying the luminance of the color-bar colors and combining them, a complete palette can be generated, and the resultant combinations will be reproducible within the NTSC system.

D. OTHER PICTURE QUALITY CONSIDERATIONS

1. Number of Colors in the Image

The literature generally recommends a minimum number of colors be used in graphics. The number of colors, choice of colors for color compatibility and black and white compatibility are closely related to the issues of image "clutter" and the length of time required for a segment to convey information. Here engineering begins to get close to subjective evaluation and timing and image clutter will be the subject for testing with the AMS Advisory Panel during the next year as complete segments are produced by the IVAM output.

The purpose of addressing color issues is to define a collection of "color sets" which can be selected automatically by the Controller, to provide a pleasing image and satisfy the requirements of NTSC and black and white compatibility. A segment will have a basic requirement for a number of colors--two, three, four, up to eight. There will be several sets of three colors which could be used for any segment requiring three colors. However,

to be suitable for the content of the segment, some of the sets may be excluded to avoid showing water on the map image as red or green, etc. The color set(s) which apply to certain segments will be identified in the segment descriptor. Several different color combinations might be available for the same segment, but fed at different times. This would allow different local TV stations to maintain a certain amount of individuality.

2. Timing

The amount of time required for effective communication of a particular message must be determined for each segment. Some brief tests of the time required to convey warning messages have been performed, but only tentative conclusions have been reached. The timing consideration is dependent on context, color choice, picture complexity, and special effects, such as scintillation and animation. Storyboarding will determine the explicit "style" of each segment and the timing will be determined by testing the segments with the AMS Advisory Panel during the next year. Since transmission times are limited, optimum use of segment time is paramount.

3. Alphanumerics Placement

When images are produced on an automated basis, the location of labeling in the picture must also be determined automatically. This task is easily done by hand, where one looks at the picture and can subjectively determine the size and location of alphanumerics. However, the algorithms for doing it automatically are not obvious. As the segments are storyboarded, a pattern of space use within the picture format will allow the algorithms to be developed so that important picture information is not covered with alphanumerics.

E. SUMMARY COLOR RULES FOR IVAM

The choice of color sets which the IVAM system uses will be determined during the development phase of the program. One or more color sets may

apply to a particular segment and will be stored in the Controller to be called up as part of the automatic system operation. Each color set must satisfy the following:

- 1) ADJACENT COLORS should be at least two gray levels apart in the standard 10 level gray scale. This is to insure black and white compatible color.
- 2) ADJACENT COLORS must not be greater than 90° apart on the color wheel, to avoid color "crawl" at the edge between colors.
- 3) THE SUM OF LUMINANCE VALUE AND 1/2 THE SATURATION VALUE for any single color must be less than 1.1, to be reproducible through the NTSC encoding system.
- 4) COLOR CHOICE should represent mid-range values of average picture luminance for best picture balance and brightness.
- 5) NUMBER OF COLORS should be minimized for best information transfer.

The apparent conflict between "rules" 1 and 4 is resolved by compromise, depending on the information content of the picture. One can make some adjacent colors only one gray level apart if the gray values are near mid-scale and the overall picture is uncluttered and well-balanced.

F. AMS ADVISORY PANEL

The first meeting of the AMS Advisory Panel was held on 15 February 1976. Attending were: Conrad Johnson (Chairman), WMT-TV, Cedar Rapids, Iowa; Mark Eubank, KUTV, Salt Lake City, Utah; Fred Norman, KOCO-TV, Oklahoma City, Oklahoma; Elliot Abrams, WPSX-TV, State College, Pennsylvania; George Winterling, WNIV-TV, DeKalb, Illinois; and University of Wisconsin personnel. IVAM concepts and status were discussed and the panel made several suggestions for improving the IVAM output.

The panel emphasized:

- 1) Animation is very important for conveying certain weather information,

especially precipitation patterns and history.

- 2) All of the panel members urged IVAM to provide, as an additional service, a special series of TV frames with the "raw" weather data in the AFOS manner for weathercaster use only.
- 3) Panel members felt that, if unavoidable, they could afford hardware for recording image segments and perhaps also to provide data to remote locations.
- 4) IVAM must be capable of providing warnings.

Several of the weathercasters use their own animated artwork, using various schemes to produce it. The production of it is tedious and expensive. IVAM animated segments were discussed with great enthusiasm. Mark Eubank remarked, "one of the most difficult things is to stand in front of a still map and try to describe how the weather is moving". Conrad Johnson uses an elaborate system of mirrors, rotating disks, and a viticon camera with chroma-keyed overlays to simulate radar motion.

All of the weathercasters expressed interest in having the raw data available from which the IVAM segments are produced. There were three major reasons: (a) to be able to adjust, or at least comment on, situations which may be modified by specific local effects; (b) to check on the accuracy or timeliness of certain data; and (c) to be able to produce special user programs or forecasts for special audiences or private meteorologist service. A special series of TV frames, not to be broadcast but for weathercaster use only, could transmit Service A, Service C, and forecasts in a single half-minute segment. These could be recorded by the weathercaster and reviewed one frame at a time. The data is readily available in the data base and could be simply defined as another segment.

Transmission of picture information on phone line was discussed. The cost of \$50,000 to \$100,000 for a piece of equipment was considered a reasonable

amount for a TV station to buy. That would make possible the production of data for phone line transmission locally, and the TV station could receive image elements by telephone for updating the data base. This subject is discussed more fully in Section VII.

Elliot Abrams said that once the idea of "NOWCASTING" is established, there will be pressure "to be right". The ability to adjust the forecast where one knows of very local effects is important. For instance, if the segment shows a snow cover of 2-4 inches, but locally there is ten inches, you've got to be able to show ten inches or you're in trouble.

Fred Norman said, "you've got to be able to do warnings! That's what it's all about in Oklahoma. If you are not warning about tornados you might as well not be in business!" Where warnings cannot pre-empt a current TV program, IVAM could transmit a part of a frame each frame during the retrace period without disrupting the program. Special equipment would be required at the receiving station to strip out the part images and to reassemble them into complete frames. A new frame could be transmitted every three seconds by this method.

1. Summary of AMS Advisory Group Panel Reviews

The overall concept of IVAM was wholeheartedly endorsed. Presentation formats, timeliness, and distribution were enthusiastically received. Suggestions made by the group being studied for incorporation into the IVAM system are:

- 1) Addition of segment(s) for distribution of raw data to the weather-caster. This is necessary for adjustments due to local effects, as well as allowing the weathercaster to enhance the pictorial data received.
- 2) Transmission of specific information via low bandwidth channels, such as phone lines, so that data could be relayed to consumer via private meteorology consultants.

VII. FUTURE IVAM EFFORT

A. INTRODUCTION

A major effort for the next year will continue to be software development. However, as the software system becomes more complete, more emphasis must be placed on completing the prototype IVAM hardware. The hardware at present is a development tool which borrows many functions from McIDAS. As the software modules begin to operate together, the system flow will be impeded by slow software and hardware not designed for a production system. We plan to schedule development of the prototype Video Assembler subsystem so that a serious bottleneck in software development will not occur.

The second major task will be to define, develop and assemble the Video Assembler subsystem.

A third major effort will be delivery and installation of the IVAM prototype to the Washington, D.C. area to interface with AFOS. We must begin soon to make detailed plans for the installation and test of the system, which is targeted for late calendar 1977.

Effort on the definition of segments will continue with storyboarding and standardizing of video images. There will be more emphasis on the meteorology inputs, too, as far as choosing most effective data source for determining specific weather information.

The Test and Evaluation effort will begin a second phase, where actual IVAM output may be evaluated. Current studies, dealing with general issues such as satellite image graphics, color and transitions, will give way to testing segment effectiveness, communication quality, and usefulness by the weathercaster and TV station. Coordination will continue with the AMS Advisory Panel.

Distribution to broadcast network stations is no longer a problem, but

detailing the specific interface from the NWS to many cable stations will require considerable effort during the next year.

B. SOFTWARE DEVELOPMENT

IVAM software is to be developed in two steps. In the first step the goal is to complete a flexible tool for generating a variety of video images. The throughput of the system is not of paramount concern. More compelling is the optimization of the development process itself. During this phase it is necessary to demonstrate the feasibility of parallel processing and the compatibility of the IVAM software with such a solution. The second step, in which the parallel processing operation achieves production efficiency, will start about December 1976, which fits with the hardware development schedule since the Video Assembler subsystem equipment will be available then.

During the development phase the first priority is to take control of the Control Processor environment. This means first of all, implementing drivers for the data tablet, the Adds terminal, the video tape recorder, the digital refresh disk, the semiconductor refresh buffer, the color enhancements, and the McIDAS communication. At the same time, the rest of the interface to the RSX-11M operating system has to be resolved so that IVAM modules can be run under RSX-11M.

When these two tasks are completed, and we expect them to be finished by June 1976, two major efforts can proceed independently. The first is to implement the modules for specifying and generating video images, and the second is to develop the system which schedules the processes and allocates the resources required to produce them.

C. IMAGE GENERATION MODULE DEVELOPMENT

1. Video Chain

The first goal of the module development immediate effort will be to

produce a basic sequence of modules which constitute a software video chain. Once a complete chain has been established, it will provide a context into which new modules can be added or substituted easily.

The video chain will be used as a test bed to generate images which will be evaluated to provide feedback for further module development and refinement. Early feedback on image quality, aesthetics, and information transfer effectiveness is very important because these findings may impact the design of the Video Assembler subsystem equipment.

The modules required to provide the basic video chain are scan conversion, linked line sort, element pack, element merge, and fill-in. For each of these modules a number of alternatives are possible. Each involves trade-offs of processing time, storage requirements, and ease of use. Rather than try to resolve these issues entirely by analysis, we have chosen to implement the most general solutions first. These modules are now coded and will be implemented in such a way that the preceding and following steps do not care how that step is done so that the change to one module can be implemented and substituted without affecting the other modules.

2. Semantic System

As the video chain is developed, the implementation of the Semantic System can begin. This first entails the building of a data set directory system which is one of the two points of interface between the Semantic System and the Physical Operating System. It allows the Semantic System to refer to a data set by its directory entry and to be ignorant of its physical location and characteristics.

At this point the basic net mechanisms must be implemented. These will be used to build the descriptors which are used to specify image formats, image sequences, and processing dependencies. They will also be used

to define the Operation Nets that control the execution of the modules. The color sets, the color topology and the domain topologies will also use these mechanisms.

3. Image Formats

As the Semantic System takes form it will become possible to build a library of very general image formats which are not specific to time, location, or weather data. These formats will be used as templates to guide the modules in producing all or part of a particular graphic image.

A dynamic capability will be added to these formats so that all of the IVAM segments can be created. Many alternative segments will have to be tested to determine the best methods of animated presentation.

4. Presentations

As all of the image types are identified and reduced to Image Formats, the methods for specifying sequences of images will be implemented. Special graphic sequences like the tractor and football sequences demonstrated in last year's test tape will be included. Throughout this process a number of test tapes will be generated for the AMS Weathercaster IVAM Advisory Panel to review and criticize.

The decision tree to determine which IVAM segments are to be shown at a given time will be addressed. The decision process will be developed to be run automatically by IVAM based on what it knows about the weather situations, but provision will be made for the AFOS operator to monitor and to override the automatic processor.

5. Sysgen

Finally, a series of procedures will be developed which will aid AFOS in tailoring the IVAM system to a particular WSFO location. These procedures will allow the identification of local scales, cities of interest, and local

landmarks. It will also provide means for later adding new formats and data sources as they become available.

D. Physical Operating System

The systems programming effort will proceed in parallel with the development of the video processing and Semantic systems. This part of IVAM software is scheduled to be completed later because the video software chain can operate initially under RSX-11M. Using RSX-11M it will be possible to produce any static IVAM image as soon as the necessary modules have been implemented. As the IVAM operating system is brought up on the 11/40 and tied to the Semantic System, it will start to provide its flexibility before the multiprocessing version is implemented. The multiprocessing capability will greatly increase the throughput of IVAM, but will have little effect on its capability in strictly video or programming terms. This means that much of the operating system development task can proceed independently of the video effort. In fact the video modules will be used to test the operating system without delaying video development.

1. Operating System Primitives

As the understanding of RSX-11M deepens, more and more of the basic IVAM operating functions can be implemented initially through it. First, we will implement the IVAM trap processor so that IVAM can take control of the RSX-11M interprocess message system, the scheduler, and the file handler. Next, a dynamic memory allocation system will be inserted to suballocate within partitions.

At this point, the series of mechanisms required to support the execution of IVAM modules will be brought up. These will allocate space, stage the input data, load the module code, create the module call, monitor module progress, receive the module return, and accept responsibility for

the module output.

2. IVAM Operation

When the module support functions are in place, it will be possible to execute sequences of modules under IVAM as opposed to RSX-11M control. These functions will be simple sequenced at first but later will be brought under the control of the Semantic System which will use an image descriptor to determine what modules should be executed next.

3. Crosspoint Primitives

As the IVAM system is brought up on the 11/40, the crosspoint primitives can be implemented. These are the simply functions required to assign and free crosspoints and to effect interprocessor communication. The interprocessor communication will operate through the system of interprocess messages mentioned earlier.

When all of the crosspoint primitives are completed, it will be possible to do simple multiprocessing under explicit programmer control.

4. Automated Processing

At this point the nature of the parallel processing environment will have been determined. So long as IVAM's load is completely predictable and known in advance, then a special scheduling program can compute the optimum order of processing and output a stream of very low-level commands which will drive the operating system functions. In this case scheduling and resource allocation will not have to be done in real-time. A simple sequencer will step through the system command list.

On the other hand, the actual operational environment can become more complex with new requests being made at arbitrary times during periods of severe weather. At these times the operating system will have to make deci-

sions in near real-time, but the range of decisions is likely to be much smaller.

5. Integration and Optimization

Once IVAM is a multiprocessor operation it will be possible to tune its performance to optimize throughput. It will also be necessary to integrate the final output hardware which will change the system timing and throughput dramatically. The integration of the system into AFOS can also begin during this phase; hopefully with a direct telephone tie into the experimental facility.

E. HARDWARE DEVELOPMENT

Earlier, in Section IV, the IVAM system was described as consisting of two major subsystems and a Disk File. Hardware for most of the Controller and Element Generator Subsystem was purchased during the past year. The Disk File must be purchased, and the hardware for the Video Assembler subsystem must be purchased or fabricated during the next year. The hardware development and purchase program is presented below by subsystem.

1. Controller and Element Generator Subsystem

This subsystem is complete except that we purposely bought only half the disk space which the final prototype configuration will require. We now have two model RK05 disk drives operating through a single controller. These disks provide a total of five megabytes capacity for inactive modules, input data transient store, grid point value files, etc. We know that we must double this capacity for the prototype system and we believe that doubling it will be enough. During the next year we plan to buy two more RK05 disk drives and a controller.

2. Disk File

We have examined several alternatives for the storage buffer which we need between the two subsystems. The required capacity has been determined quite firmly, and the input data rate is known to be very low compared to current hardware capabilities. Output, however, could be a problem because required head movement and disk rotational latency might present unacceptable delays. Splitting the file into two disks accessed independently could relieve the problem given well-planned placement of data sets on the disk. However, this increases costs and we are now convinced that a single disk will suffice given the same ability to plan data placement and accepting the possibility of minor output timing constraints which might require repeating a few frames a fraction of a second longer than required.

During the next year we will confirm our present plan by further analysis and actual disk access tests using McIDAS disks. If our plan is confirmed we will purchase one 9.6 megabyte high speed access disk drive and controller with DMA option.

3. Video Assembler Subsystem

Studies to further refine and confirm the design of this subsystem will continue until about 1 August 1976. Then we will proceed to purchase or fabricate the components and to assemble the subsystem as rapidly as possible. Several alternative designs for this subsystem are being considered. Three which currently appear to offer the greatest promise are described and discussed below.

a. Alternative One

This alternative is the one described in Section IV. It is conservative, we are sure it will do the job, but another alternative may prove to be more economical or reliable after thorough analysis. This alternative is shown in

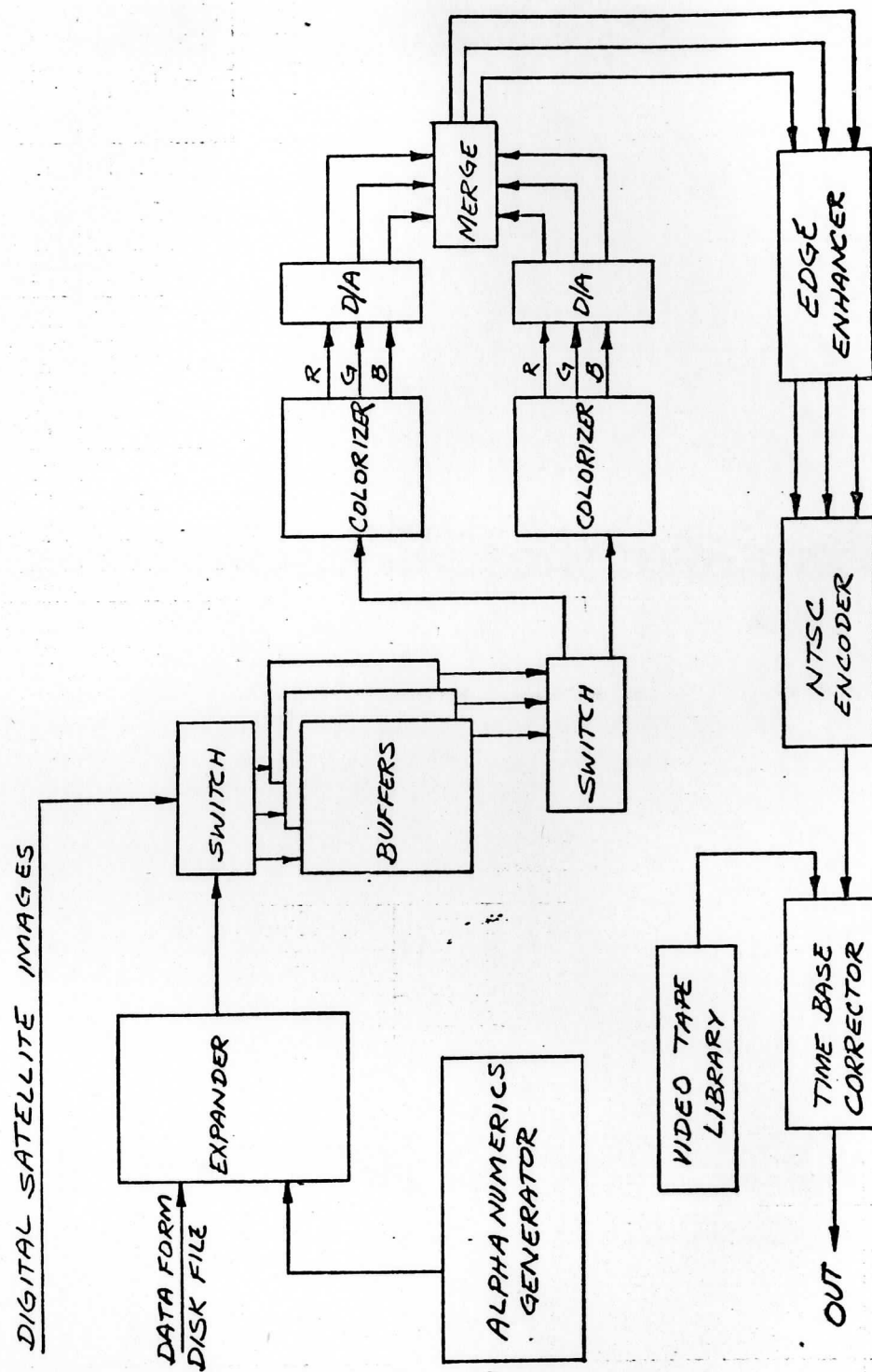
Figure VII-1.

Data from the disk file and from the Alphanumeric Generator, in 16 bit element/line intercept word format, is read into the Expander at the direction of the Controller. The Expander includes a small buffer pair where these intercept words are stored then read out in order of line position. The Expander repeats image element values to either widen graphic lines from one pixel to three or more for better appearance, or to fill the interval up to the next intercept. The output of the Expander is four bits of information plus a spare bit and a flag bit to be stored in one of the six bit buffers.

Three buffers are required so that two of them can be in the read-out mode simultaneously while the third is being loaded. Two buffers are needed for output so that two frames may be combined in a fade-in, fade-out mode. Before the graphics data are read into the buffer the satellite image is read in at TV rate from the Satellite Image Sequencer, then the graphics lines are overwritten. While graphics from the Expander are only four bits deep, satellite pictures are five bits. For overlays of graphics on satellite pictures, the graphics must be colored, but not the satellite picture. This is accomplished by using the sixth bit in the buffers as a flag bit to switch between "color" and "don't color".

The Colorizer is three 16 place look-up tables which are loaded by the Controller to convert the 4 bit graphic value to one of 15 colors (plus black). Since the computer can load the look-up tables at TV frame rate the Colorizers are also used to fade one image and increase the other for fade dissolves. The output of the Colorizer is a three wire R.G.B. digital signal which is converted to analog and then the two images are merged.

The output of the Merger is passed through an Edge Enhancer, a standard piece of TV equipment which sharpens the edges of lines and letters to provide



ALTERNATIVE ONE
VIDEO ASSEMBLER SUBSYSTEM
FIGURE III - 1

crispness to the final image. The final steps are to convert from R.G.B. to NTSC format and to correct the video time base to broadcast stability levels.

b. Alternative Two

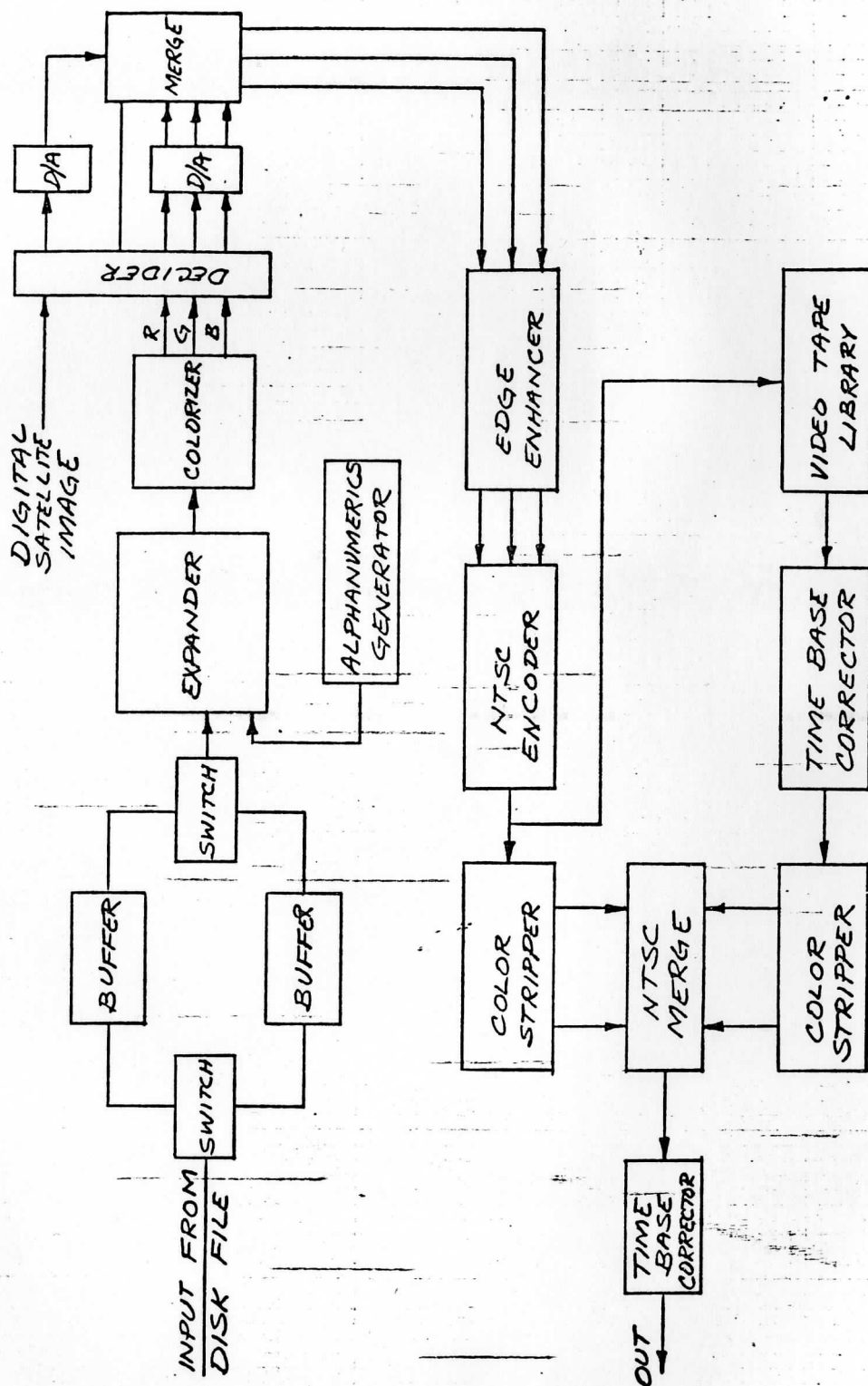
The cost of the three full frame buffers in Alternative One might prove to be excessive. This alternative eliminates the full frame buffers and replaces them with two smaller buffers before the Expander. Figure VII-2 shows Alternative Two in block diagram.

The input from the Disk File is switched to the empty buffer which need have a capacity of only 10% of the bits of the full frame buffer of Alternative One. The full buffer is read-out at 30 frames per second as many times as needed into the Expander synchronized with the input of the Alphanumeric Generator. The Expander overwrites one or the other as instructed by the Controller, widens lines, and fills areas as before. The expanded full TV frame is colorized as in Alternative One.

Without the full frame buffers a new technique must be employed to combine the graphics and satellite images. This is done by passing the synchronized digital satellite image and the colorized graphic through a "Decider" which determines which pixel has priority and signals the Merger through a delay line. After the signal is converted from digital to analog the Merger combines them properly.

The fully assembled RGB image is passed through an Edge Enhancer and an NTSC encoder as in Alternative One. But at this point some images will be switched to the recorder of the Video Tape Library where they are stored temporarily. This is done in advance to every second image in a series which is to have fade-dissolve transitions when shown in sequence.

The fade-dissolve is harder to do with NTSC encoded images but can be done by passing the signal through a "Color Stripper" which effectively



ALTERNATIVE TWO
VIDEO ASSEMBLER SUBSYSTEM
FIGURE VII - 2

separates the chroma and luminance information and forwards both on separate lines. In the NTSC Merger the luminance signal from the first image is reduced from full level at the same time as the luminance signal of the second image is increased from zero; at midpoint the chroma signal is switched from the first image to the second. The merged signal is passed through a Time Base Corrector and out to the media.

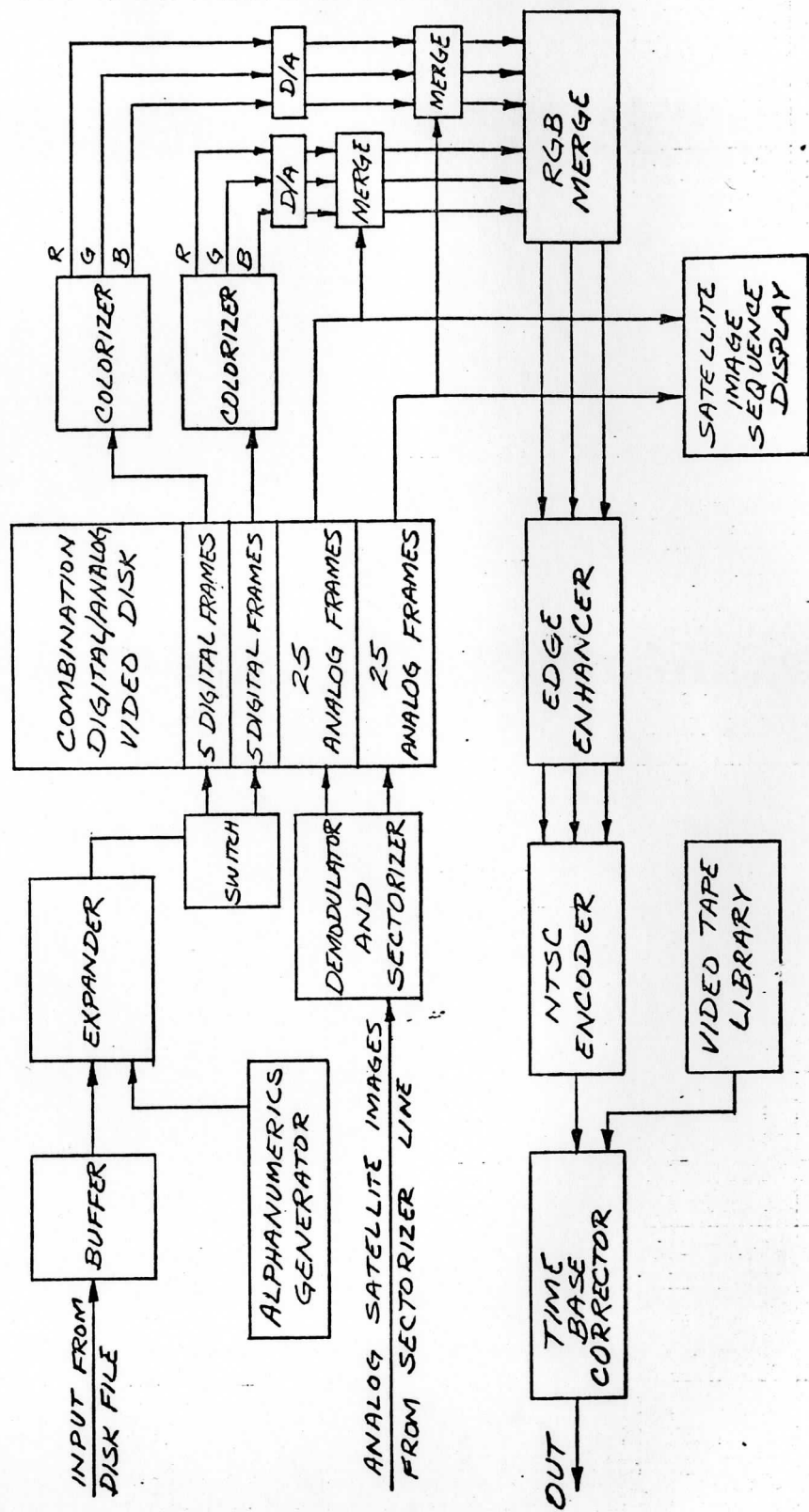
c. Alternative Three

Both Alternative One and Two presuppose the availability of a Satellite Image Sequencer or similar device capable of serving as a source of digitized satellite images. Since this may not be certain, a third alternative is being considered which can accept the unmodified output of the CDDS line from the NESS Sectorizers. In addition, it can provide the Satellite Image Sequencer display with greater capacity and flexibility "for free"! See Figure VII-3 for a block diagram of Alternative Three.

This alternative takes advantage of the flexibility provided by a disk recorder used to record TV frames in both analog and digital format. Disk drives with the required capabilities are available, and while they have been on the market less than a year they have demonstrated excellent performance and reliability.

The buffer between the Disk File and the Expander is the same size as the two in Alternative Two but only one is required. Disk File input and Alphanumerics data are fed into the Expander at full TV rate and the Expander performs the same functions as before. The full TV frames are switched alternately into two sections of the Video Disk, each capable of holding five full frames. These are repeated as required as they are read out from the disk, colorized and converted from digital to analog format.

Satellite images are received from the CDDS line, demodulated and resectorized to TV frame size and placed on the Video Disk in one of two



ALTERNATIVE THREE
VIDEO ASSEMBLER SUBSYSTEM
FIGURE III-3

sections each having space for at least 25 analog frames. The satellite images are read out as required to be merged with the graphics and the two completed image channels are in turn merged to provide fade-dissolve or other special effects. The merged signal is passed through an Edge Enhancer, NTSC Encoder and Time Base Correcter as before.

The block diagram also shows a Satellite Image Sequencer Display which could be incorporated with the AFOS display either as an additional CRT or as a call up display on existing consoles.

d. Comparison of Alternatives

We are not ready to recommend a design for the Video Assembler Subsystem, but it is worthwhile to discuss some aspects of the three alternatives described above.

- 1) Cost: Rough cost estimates indicate that costs would be approximately:

	<u>Devel. & Prototype Cost</u>	<u>Second Unit Cost</u>
Alternative One	\$111,000	\$79,000
Alternative Two	\$ 92,000	\$54,000
Alternative Three	\$115,000	\$66,000

- 2) Reliability: Alternative One has the highest reliability rating and lowest routine maintenance requirement because it has no rotating parts.

Alternatives One and Three include a 3/4 inch video cassette playback unit to serve as a source of time-lapse weather sequences, special prerecorded segments, etc. In Alternative Two this unit is upgraded to a full servo-controlled recorder and playback unit which is in-line for all satellite image sequences, and this fact reduces the reliability rating for Alternative Two.

The combination analog-digital disk in Alternative Three does increase the preventive maintenance requirement and probably would affect the reliability.

Overall we would rate the Alternatives One, Three, and Two for reliability.

3) Performance: Alternative Three will outperform either of the other two by a wide margin. One reason the cost of Alternative Three is highest is because it includes all of the hardware necessary to process satellite images from the CDDS line. To provide equivalent capability in the other two alternatives, their costs should be increased by about \$24,000.

In addition, Alternative Three offers a much larger store of satellite and other images which can be combined with graphics or with each other in many ways. In addition to the required fade-dissolve it would be easier to provide special effect transitions with Alternative Three than with the others.

Alternative Two requires preparation of parts of sequences involving satellite data in advance, but this should not prove to be a serious problem.

Alternative One offers the least flexibility - it will do the basic IVAM job but that is all.

e. Other Considerations

All of the IVAM designs discussed above assume that satellite images will be included in the final presentation in unmodified form, except for gridding or other overlays. As noted in Section V, this assumption needs to be examined carefully. We have produced examples of "cloud graphics" created by a multistep process which:

- 1) Normalizes satellite images to remove the effect of sun angle and earth curvature on cloud brightness.
- 2) Contours the images by reducing the number of brightness levels to a small number such as eight.
- 3) Generates a line around each of the contoured areas and smoothes the lines.
- 4) Reconstitutes the image by filling between lines with constant grey values.

At present we have nearly all of the software necessary to do this process, but it exists as parts of other software systems. We plan to reassemble the software to perform this process in an efficient manner, and to produce sequences of cloud graphics for a variety of weather and geographic situations. We will evaluate the resultant presentations using the AMS Weathercaster Advisory Panel and will submit video tapes for evaluation by NWS and NESS. Also, to a limited extent we will attempt to obtain the reaction of the general public.

We believe this study to be very important to the design of IVAM and, as discussed in Section V, to plan for distribution of IVAM presentations to cable TV outlets. If it should be decided that the cloud graphics are acceptable or even preferred for TV presentations in comparison to the original pictures, then the design of the Video Assembler Subsystem becomes much simpler. Further, the impact of such a finding on the general problem of distributing the output of the GOES satellites could be very important. If the cloud graphics should prove to be acceptable for TV presentation they might also be acceptable to other users of satellite pictures.

F. OUTPUT DISTRIBUTION STUDY

During the next year we will concentrate on the problems of distribution to cable TV outlets and non-network broadcast TV stations. The use of some form of narrow band distribution technique is becoming increasingly attractive. We will attempt to test one or more possible techniques using parts of the Video Assembler Subsystem hardware as it becomes available.

We have developed contacts in small cable companies, larger cable distributors and in the national associations. All of these contacts have volunteered to help us, and we will call upon them to help us in evaluating possible narrow band distribution systems. We plan to attend a national

cable TV conference in January or February 1977 with a demonstration and questionnaire to help define an acceptable system.

G. PRESENTATION CONTENT STUDIES

The study of informational content of the presentations is complete. It is described, a segment at a time, by the information on each segment header sheet. From the header sheet, a storyboard must now be developed for each segment. The storyboard determines the timing, style, limit of colors, and the communication effectiveness of the segment. The storyboard identifies every image in the segment, frame by frame, and describes all animation and transition effects. These storyboards, which are the output specification for the software, are being written first for the segments which use the software modules already working on IVAM. This allows the most immediate feedback on evaluation of the segments. It is expected that as the storyboarding proceeds, standard sequences of frames will appear and allow a somewhat modular approach to writing the storyboards themselves.

H. PRESENTATION CONTENT TEST AND EVALUATION

The concept of evaluating IVAM output has changed considerably since the organization of the AMS Advisory Panel. The previous plan involved a considerable amount of film and questionnaire preparation, coupled with the testing of the films with various groups, then evaluating questionnaires and gaining statistical results. We now realize that until IVAM is developed and able to provide a continuing series of presentations that it is not possible to obtain reliable public reactions. Instead, the AMS IVAM Advisory Panel is being consulted as a more effective approach at this time to the same questions. Each of these people is a meteorologist expert in the field of TV weather communication and has already studied many of the problems we

are now facing. The geographical locations of each of the people on the panel is an aid, too, for they represent a more diverse cross-section of audience.

Test tapes and films will be prepared on a regular basis and mailed to the members of the Advisory Panel. Tapes and questionnaires will be circulated every few months, as output development permits. Two Advisory Panel meetings are planned in the next 12 months, in August and in February.

We have discussed the possibility of performing IVAM program tests with either PBS or local cable stations. The National Cable Television Association monthly journal has offered us copy space for a questionnaire or series of questionnaires. We have no plans at present to use any of these media for tests in the next year, but may if some very specific questions may need a large audience response. An example is a questionnaire to cable operators regarding distribution techniques and issues.

As discussed in the previous hardware development plan, we intend to evaluate the acceptability of cloud graphics derived from satellite images as a possible replacement for the satellite pictures themselves. This study has high priority and will be pursued during the next few months.

I. FINAL PROGRAM DEVELOPMENT

This is the fifth and last task to be undertaken in this program. As originally proposed, this effort will start only after NOAA is satisfied that the implementation of IVAM is feasible, and that when implemented, IVAM will meet the NOAA objectives. This is the wrap-up effort in which software development is stopped, configuration control is imposed, detailed interfacing with AFOS is accomplished, hardware specifications are completed, final documentation is prepared, etc.

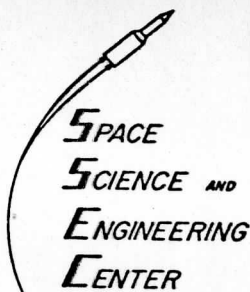
We mark the start of this task at the end of assembly of the complete

IVAM prototype system at SSEC and demonstration of the ability to produce broadcast quality presentations. Included in this task is the movement of the prototype equipment to the Washington, D.C. area, physical interface with AFOS and test operations in that location. We expect to start this task in September 1977 and to complete it by 31 December 1977.

IVAM ANNUAL REPORT, SPRING 1976

Appendix A

IVAM Computer Bid Request



1225 West Dayton Street
Madison, Wisconsin 53706

THE UNIVERSITY OF WISCONSIN

Gentlemen:

You are invited to submit bids for providing data processing equipment as described in the attached specification. The equipment is for a multiprocessor approach to data handling and is based on a configuration using dual-port memories for each processor. Attached to this letter is a figure which represent the approach. This figure is for information only; it is not part of the specification.

The system requires a variety of equipment. It is expected that no manufacturer produces all of the equipment requested, so bids are invited for separate items. Bidders are requested to organize their bids according to the following list. Bidders should address each subparagraph of the specification and describe how their proposed system will meet each requirement.

BID LIST

ITEM 1) SYSTEM

Bid a complete system using double dual-port memory for each processor.

- (a) cost dual-port memories separately such that they may be procured as single dual-port memories if so desired.
- (b) separate costs between the dual-port control and the memory itself if possible.
- (c) note all exceptions to the specification.
- (d) describe interprocessor data transfer subsystem and interrupt structure subsystem relative to requirements of Part VII of the specification.
- (e) bid peripherals separately.

PERIPHERALS

ITEM 2) KEYBOARD* - Bid separately the keyboard terminal per part VIII of the specifications.

ITEM 3) ALPHANUMERIC CRT* -
Bid separately the Alphanumeric CRT per part IX of the specification.

* The keyboard and Alphanumeric Terminal may be bid as one unit.

ITEM 4) DATA TABLET -
Bid separately the data tablet per part V of the specification.

ITEM 5) DISC - Bid separately the disc per part XI of the specification.

ADDITIONAL MEMORY

ITEM 6) Bid additional 4K, 8K, and 16K memory modules per part II.B.2.1 of the specification. Where applicable, bid both core and semi-conductor memory.

INTERFACE MODULES

ITEM 7) Bid interface modules per part XII of the specification.

ACCESSORIES

ITEM 8) Bid Accessories per part XIII of the specification.

DOCUMENTATION AND SUPPORT

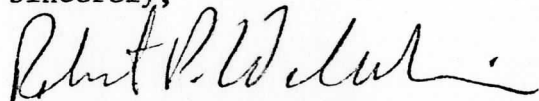
ITEM 9) Where costed additional to products, bid all documentation and/or support hardware or software per part XIV of the specification.

OPTIONAL SYSTEM

ITEM 10) Bidders are invited to propose any other options that they feel meets the stated requirements.

Please address all technical questions regarding this bid request to the undersigned at (608)262-5938.

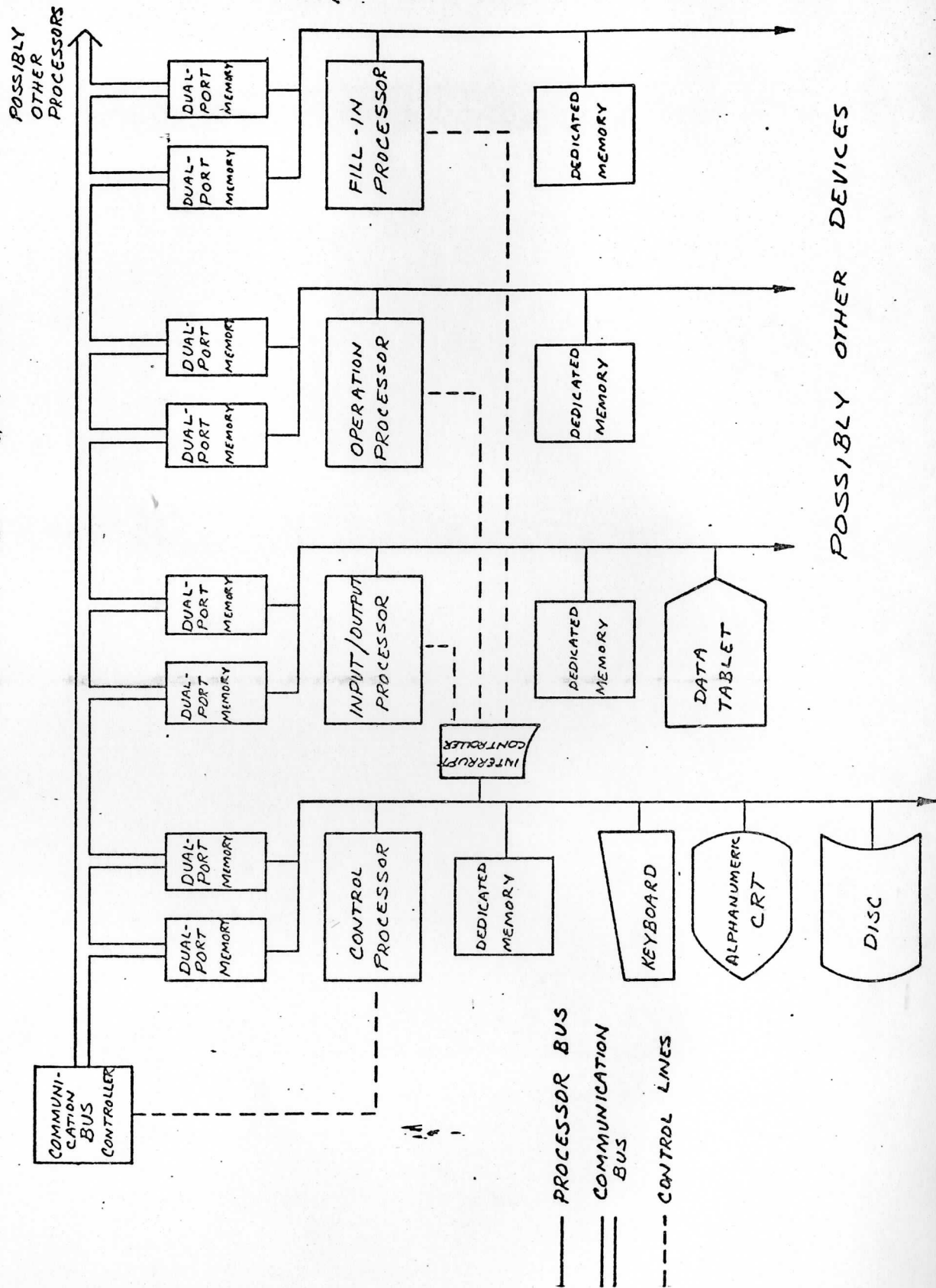
Sincerely,



Robert P. Wollersheim
Project Manager

RPW:mt
Enc.

FIGURE 1
DUAL-PORT MEMORY CONFIGURATION



SPECIFICATION

I. INTRODUCTION

This specification defines the requirements for equipment needed to build a multi-processor data handling system. The basic configuration identified for the system uses double dual-port memories for each processor.

II. SYSTEMS REQUIREMENTS

II.A. Systems Initialization

The system must contain the necessary hardware to facilitate complete initialization of the system in a turnkey mode. This must include:

- Hardware bootstrap on the Control Processor
- Necessary interprocessor protocol capability to initialize each of the subordinate processors.

II.B. Systems Support

The systems shall be provided with an operating system sufficient to facilitate software development and distribution of code to each of the processors. It must contain the following:

- Editor
- Relocatable Assembler and/or Macro Assembler
- Collector (Relocating loader)
- High-level Algorithmic Language
- Real-Time Debugging Software and/or Hardware
- Real-Time Multi-Level Task Operating System allowing access to Memory Allocation and Management and File Management.

II.C. Systems Integration

All processors and subsystems of the system shall be supplied with all enclosures, power supplies, cabling and interface modules to all relevant elements of the systems not covered specifically in this document.

III. PROCESSOR REQUIREMENTS

III.A. Word Byte Size

Each processor shall have a word size of 16 bits or greater and have a byte size of 8 bits.

III.B. Address Space

Each processor shall have an address space of 32K words or greater. Additional consideration shall be given where the address space allows expansion beyond the 32K word limit.

III.C. Registers

III.C.1 Each processor shall provide registers to perform arithmetic and logical computation and addressing and indexing. Additional consideration shall be given processors which provide hardware stack operations.

III.C.2 Additional consideration shall be given to processors which provide more than 4 general purpose registers which optimize their use of functions specified in III.C.1.

III.D. Classes of Instructions

III.D.1 Basic Instructions

- Each processor shall provide the following classes of instructions:

- Arithmetic
- Logic
- Indexing
- Input/Output
- Conditional and Unconditional Branching
- Shifting
- Halt
- Interrupt and/or Trap Instructions

III.D.2 Subroutine Protocol Instructions

Each processor shall provide instructions to facilitate subroutine linkage and return, recursive subroutines, and reentrant subroutines.

III.D.3 Other Requirements

Each processor shall provide instructions to facilitate relocatable, reentrant, reusable, and position independent code.

III.E. Addressing Modes

Each processor shall provide the following address modes:

- relative
- index
- register
- indirect (deferred)
- absolute

Additional consideration shall be given processors which also provide the following address modes:

- immediate
- auto increment index registers
- auto decrement index registers
- linked list searches

III.F. Coding

III.F.1 Code Compatibility

- Each processor shall be code compatible with the other processors, or at least upward compatible to the Control Processor instruction set.

III.F.2 Instruction Usage

- Each processor shall optimize use of the instructing set per III.D and addressing modes per III.E on both word and byte data.

III.G. Error Detection and Protection

III.G.1 Each processor shall provide error detection and protection including stack underflow and overflow detection (if applicable), power fail/auto restart, and illegal instruction trap.

III.G.2 Additional consideration shall be given processors which also provide the following error detection and protection features:

- memory parity detection and correction

- privileged instruction protection
- memory protection including read only access, read/write access, and execute only access
- software settable stack boundaries (if applicable)
- software priority interrupt control

III.H. Instruction Execution Time

Each processor shall have a register-to-register add time of 1.5 microseconds or less.

IV. SPECIFIC PROCESSOR REQUIREMENTS

IV.A. Control Processor

In addition to meeting the basic requirements as stated in III., the Control Processor shall be provided with the following.

IV.A.1 Real-Time Clock

The Control Processor shall have a real-time clock. It may be either a programmable real-time clock or a line frequency clock.

IV.A.2 Alphanumeric CRT Terminal and Keyboard

The Control Processor shall be provided with an alphanumeric CRT terminal and keyboard per part IV.A and IV.B of this specification.

IV.A.3 Interrupt Capability

The Control Processor shall be capable of receiving vectored interrupts and responding to them as follows.

IV.A.3.1 Interprocessor Interrupt

The Control Processor shall be able to interrupt and be interrupted by the following processors per paragraph IV.E.

- Input/Output Processor
- Operations Processor
- Fill-in Processor

IV.A.3.2 Memory Transfer Controller

The Control Processor shall be capable of interrupting and being interrupted by the Communications Bus Controller per paragraph VII.

IV.A.3.3 Keyboard Interrupt

The Control Processor shall be interruptable by the Keyboard as specified in part VIII.

IV.A.3.4 Alphanumeric CRT Interrupt

The Control Processor shall be interruptable by the Alphanumeric CRT as specified in part IX.

IV.A.3.5 Disc Interrupt

The Control Processor shall be interruptable by the Disc as specified in part XI.

IV.A.3.6 Clock Interrupt

The Control Processor shall be interruptable by the real-time clock as specified in IV.A.1.

IV.A.4 Additional Consideration

Additional consideration shall be given when the Control Processor is configured to provide a user microprogramming capability as follows:

IV.A.4.1 Microprogrammability

The microprogramming capability shall include:

- a micro-assembler
- a micro-control debugging support, including hardware and/or software
- writeable control store

IV.A.4.2 Micro-code Insertion

The micro-code developed on the micro-processor of the Control Processor shall be capable of being inserted into the other processors, as specified in Sections IV.B., IV.C., and IV.D.

IV.B. Input/Output Processor

In addition to meeting the basic requirements as stated in III., the Input/Output Processor shall be provided with the following:

IV.B.1 Interrupt Capability

- The Input/Output Processor shall be capable of interrupting and being interrupted by the Control Processor as specified in IV.E.

IV.B.2 Disc Interface

The Input/Output Processor shall be capable of supporting a disc through a standard DMA interface.

IV.C. Operations Processor

In addition to meeting the basic requirements as stated in Part III, the Operations Processor shall be provided with the following:

IV.C.1 Language

The Operations Processor shall be supported with a high level algorithmic language.

IV.C.2 Additional Computational Capability

The Operations Processor shall support the following additional arithmetic functions:

- Multiply and divide
- Multiple-bit shift and double word shift
- Floating point operations
- Trigonometric functions and fast Fourier transformations.

Preference for the inclusion of these capabilities will be given in the following order:

hardware implementation

firmware (micro-code) implementation

software implementation

IV.C.3 Interrupt Capability

The Operations Processor shall be capable of interrupting and being interrupted by the Control Processor as specified in Part IV.E.

IV.C.4 Fill-in Processor

In addition to meeting the basic requirements as stated in III., the Fill-in Processor shall be capable of interrupting and being interrupted by the Control Processor as specified in Part IV.E.

IV.E. Interprocessing Interrupt Structure

The Interprocessor Interrupt Structure shall be configured to facilitate the following.

IV.E.1 Interrupt Control Processor

Each Subordinate (i.e. Input/Output, Operations, and Fill-in) Processor shall be capable of effecting a distinct vectored interrupt of the Control Processor.

IV.E.2 Interrupt Subordinate Processor

The Control Processor shall be capable of effecting a vectored interrupt of each of the Subordinate Processors.

IV.E.3 Additional Considerations

Additional consideration shall be given to an Interprocessor Interrupt Structure which facilitates as many of the following as possible.

- Passing of an interrupt vector address, used to give the address through which the processor is interrupted.
- Passing of simple messages or status information between

processors upon interrupt, from a byte up to several words in size.

V. DUAL-PORT MEMORY

V.A. Dual-Port Controller

A Dual-Port Controller shall allow access to a common memory module from either of two independent buses, one of which is accessed by the Communications Bus Controller (as specified in Part VII.) and the other being accessed by the individual processor bus (Control Processor, Input/Output Processor, Operations Processor, and Fill-in Processor) as specified in IV and VII.

V.B. Priority

The Dual-Port Controller shall give priority of access to the Communications Bus Controller.

V.C. Independent Address Space

The address space associated with the memory module of a Dual-Port Memory shall be independently assigned both to the Communications Bus Controller and the individual processor.

V.D. Additional Considerations

V.D.1 Simultaneous Access

Additional consideration shall be given to a Dual-Port Control which allows simultaneous and asynchronous access to the memory from each of the two buses.

V.D.2 Protection

Additional consideration shall be given to a Dual-Port Control which allows protection of the memory by allowing read-only or execute-only access to the memory module. This feature must be controllable from the Control Processor.

V.D.3 External Port Selection

Additional consideration shall be given to a Dual-Port Control which allows the port selection signals to be inserted externally by the Communications Bus Controller rather than through the address lines of the Communications Bus.

VI. MEMORY MODULES

Each of the processors shall be supplied with memory according to the following.

VI.A. Control Processor

- The Control Processor shall be supplied with at least two (2) independent Dual-Port Controllers (as specified in V.) each of which is supplied with at least 8K words of memory.
- The Control Processor shall be supplied with at least 16K words of additional memory which shall be non-volatile and read/write.
- The Control Processor shall allow expansion to the full address capability of the processor, as specified in Part III.B.

VI.B. Input/Output Processor

- The Input/Output Processor shall be supplied with at least two (2) independent Dual-Port Controllers (as specified in V.), each of which is supplied with at least 8K words of memory.
- The Input/Output Processor shall be supplied with at least 8K words of additional memory.
- The Input/Output Processor shall allow expansion to the full address capacity of the processor, as specified in III.B.

Operations Processor

- The Operations Processor shall be supplied with at least two (2) independent Dual-Port Controllers (as specified in V.), each of

which is supplied with at least 8K words of Memory.

- The Operations Processor shall be supplied with at least 8K words of additional memory.
- The Operations Processor shall allow expansion to the full address capacity of the processor as specified in Part III.B.

Fill-in Processor

- The Fill-in Processor shall be supplied with at least (2) independent Dual-Port Controllers (as specified in V.) each of which is supplied with at least 8K words of memory.
- The Fill-in Processor shall be supplied with at least 8K words of additional memory.
- The Fill-in Processor shall allow expansion to the full address capacity of the processor, as specified in III.B.

VII. COMMUNICATIONS BUS CONTROLLER

The Communications Bus Controller shall be configured to effect block transfers between any Dual-Ported Memories per part V, meeting the following additional requirements.

VII.A Control Processor Interface

- The Communication Bus Controller shall receive from the Control Processor set-up information, which is to include at least the following information (or equivalent):
 - Source Starting Address
 - Destination Starting Address
 - Number of words to Transfer
- The Communications Bus Controller shall transmit to the Control Processor, via an interrupt ~~line~~ per part IV.A.3.1 of the specification, a "Block Transfer Complete" signal.
- Additional consideration shall be given interfaces which facilitate the exchange of additional status information.

VII.B Priority

The Communications Bus Controller shall have priority of access over the processors to the Dual-Port Memory module.

VII.C Address Space

- The Communications Bus Controller shall have an Address Space sufficient to handle at least the configuration of Dual-Port Memory Modules as specified in VI.
- Additional consideration shall be given to a design which allows expansion capabilities to permit addressing or selection of up to sixty-four (64) 8K word Dual-Port Memory Modules.

VII.D Transfer Rate

The Communications Bus Controller shall effect the block transfer at maximum memory access rates.

VIII. KEYBOARD

The Control Processor shall be supplied with a keyboard which meets the following requirements.

VIII.A Character Set

The Keyboard shall have a standard ASCII character set.

VIII.B Interrupt Capability

The Keyboard shall have the necessary control lines to interrupt the Control Processor when a key is struck.

VIII.C Interface

The Keyboard shall be supplied with all enclosures, power supplies, cabling, and interference modules to the Control Processor.

IX. ALPHANUMERIC CRT

The Control Processor shall be supplied with an Alphanumeric CRT which meets the following requirements.

IX.A Character Set

The Alphanumeric CRT shall have a standard ASCII character set.

IX.B Number of Lines and Linewidth

- The Alphanumeric CRT shall have at least 24 lines visible on the CRT screen.
- Each line shall be capable of displaying at least 80 characters.

IX.C Transfer Rate

The Transfer Rate from Control Processor to the Alphanumeric CRT shall be at least 30 characters per second.

IX.D Cursor

The Alphanumeric CRT shall have a digitally-controlled cursor, to allow text editing and random screen writing.

IX.E Interrupt Capability

The Alphanumeric CRT shall have the necessary control lines to interrupt the Control Processors when a character has been received and written on the CRT.

IX.F Interface

The Alphanumeric CRT shall be supplied with all enclosures, power supplies, cabling, and interface modules to the Control Processor.

IX.G Additional Considerations

- Additional consideration shall be given if the Alphanumeric CRT allows simple vector-graphics capabilities.
- Additional consideration shall be given to CRT's which have both upper and lower case letters.

X. DATA TABLET

The Input/Out Processor shall be supplied with a Data Tablet which meets the following requirements.

X.A Working Area

The Data Tablet shall have a working area of at least 12" by 12".

X.B Resolution

The Data Tablet shall have a resolution of at least 1 part in 1000 in each axis.

X.C Modes of Operation

The Data Tablet shall have at least the following two modes of operation.

- Single point-activated by pressing a button or touching the point of the data pen to the data tablet.
- Continuous tracking - continuous monitoring of the point of the data pen at rates of at least 100 points per second.

X.D Interrupt

The Data Tablet shall have the necessary control lines to interrupt the Input/Output Processor while operating under both modes of operation, as specified in X.C.

X.E Interface

The Data Tablet shall be supplied with all enclosures, power supplies, cabling, and interface modules to the Input/Output Processor.

XI. DISC SYSTEM

The Control Processor shall be supplied with a Disc System, to support program development, meeting the following requirements.

XI.A Storage Capacity

- The Disc System shall contain at least 500K bytes of storage.
- The Disc System shall have a backup capability, through a removable disc pack (or diskettes) of virtually unlimited size.

XI.B Data Access/Transfer Rate

- The Disc System shall have an average access time of less than 470 milliseconds and a worst case access time of less than 820 milliseconds.

- The Disc System shall have a transfer rate of at least 15K words per second.
- The Disc System shall have an average latency of less than 80 milliseconds and a worst case latency of less than 170 milliseconds.

XI.C Interface

The Disc System shall be supplied with all enclosures, power supplies, cabling and DMA interfaces to the memory of the Control Processor.

XII. ADDITIONAL INTERFACE MODULES

Each of the processors shall allow addition of the following interface modules:

- Current Loop (20 mA) Serial Line Interface
- EIA RS 232 Serial Line Interface
- Direct Memory Access Interface
- Asynchronous Bi-directional Parallel Interfaces
- Synchronous Serial Interfaces

XIII. ACCESSORIES

The system shall allow addition of the following accessories:

- Additional Power supplies, as required
- Additional Enclosures, as required.
- Additional Connectors and Cabling, as required.
- Additional Mounting Units, as required.
- Additional Back-plane Modules, as required.

XIV. DOCUMENTATION AND SUPPORT

XIV.A Software Documentation

Documentation shall be provided for all software and shall include but not be limited to the following:

- Source listings of all software systems and modules.
- Systems conventions and functional and detail description of all software modules.

- User's manuals for all software.
- Operator's manuals for all software.

XIV.B Hardware Support & Maintenance

Documentation shall be provided for all hardware and shall include but not be limited to the following:

- Theory of operation, diagnostic tests, repair and preventive maintenance.
- Interconnection diagrams between units and associated equipment.
- Logic diagrams, schematics, assembly drawings, installation drawings, parts list, test-point list, and normal voltages, currents and waveforms.
- Table of Contents and a list of referenced illustrations.

RECOMMENDED BIDDERS LIST (50 Companies)

I. COMPUTER COMPANIES

1. California Data Processors, 2019 S. Ritchey Street, Santa Ana, California 92705. Telephone (714)558-8211.
 2. Computer Automation Incorporated. 18651 Van Karman Avenue, Irvine, California 92664. Telephone (714)833-8830.
 3. Datacraft Corporation, 1200 Gateway Drive, P.O. Box 23550. Fort Lauderdale, Florida 33309. Telephone (305)974-1700
 4. Data General Corporation, Route 9, Southboro, Massachusetts 01772. Telephone (617) 485-9100.
 5. Digital Equipment Corporation, 146 Main Street, Maynard, Massachusetts 01754. Telephone (617)897-5111.
 6. Fabri-Tek, Incorporated, 5901 South County Road 18, Minneapolis, Minnesota 55436
 7. General Automation, Inc., 1055 S. East Street, Anaheim, California 92805. Telephone (714)778-4800.
 8. Hewlett-Packard Company, Cupertino Division, 11000 Wolfe Road, Cupertino, California 95014. Telephone (213)877-1282.
 9. Interdata, Inc., 2 Crescent Place, Oceanport, New Jersey 07757. Telephone (201)229-4040.
 10. Lockheed Electronics Company (a subsidiary of Lockheed Aircraft Corporation (Data Products Division, 6201 E. Randolph Street, Los Angeles, California 90023. Telephone (213)722-6810.
 11. Microdata Corporation, 17481 Red Hill Avenue, Irvine, California 92705. Telephone (714)540-6730.
 12. Modular Computer Systems, Inc., 1650 West McNab Road, Fort Lauderdale, Florida 33309. Telephone (305)974-1380.
 13. Nanodata Corporation, 2457 Wehrle Drive, Williamsville, New York 14221. Telephone (716)631-5880.
 14. Prime Computer, Inc. 23 Strathmore Road, Natick, Massachusetts 01760. Telephone (617)655-6988
 15. Systems Engineering Laboratories, Inc., 6901 West Sunrise Boulevard, Fort Lauderdale, Florida 33313. Telephone (305) 587-2900.
- 15A
~~15B~~ Severo M. Ornstein, Bolt, Beranek, and Newman, 50 Moulton Street, Cambridge, MA 02138

16. Teledyne Systems Co., 19601 Nordoff Avenue, Northridge, California 91324
17. Varian Data Machines (a subsidiary of Varian Associates), 2722 Michelson Drive, Irvine, California 92664. Telephone (714) 833-2400.
18. Texas Instruments Inc. Digital Systems Division, P.O. Box 1444, Houston, Texas 77001. Telephone (713)777-1623.

II. MEMORY COMPANIES

1. Cambridge Memories. Inc., 12 Crosby Dr., Bedford, Massachusetts 01730. Telephone (617)271-6300
2. Advanced Memory Systems, 1275 Hammerwood Ave., Sunnyvale, California 94086 (408)734-4330
3. Advanced Micro Devices, 901 Thompson Pl, Sunnyvale, California 94086. Telephone (408)732-2400.
4. EMM Memory Products Group, c/o William F. Eversman, 1400 E. Touhy Ave., Suite 440, Des Plaines, Illinois 60018
5. Intel Corp., 3065 Bowers Ave., Santa Clara, California 95051. Telephone (408)246-7501.
6. Monolithic Memories, Inc., 1165 E. Arques Ave. Sunnyvale, California 94086. Telephone (408) 739-3535.
7. Monolithic Systems Corp., 14 Inverness Dr. E., Englewood, Colorado 80110. Telephone (303)770-7400.
8. Mostek Corp., 1215 W. Crosby Road, Carrollton, Texas, 75006. Telephone (214)242-0444
9. Motorola Semiconductor Products, Inc. 5005 E. McDowell Rd., Phoenix, Arizona 85036 Telephone (602)244-3465.
10. National Semiconductor Corp., 2900 Semiconductor Dr. Santa Clara, California 95051. Telephone (408)732-5000.
11. Plessey Semiconductors, 1674 McGaw Avenue, Santa Ana, California 92705 . Telephone (714)540-9979.
12. Synertek, 3050 Coronado Dr., Santa Clara, California 95051 Telephone (408)241-4300.
13. Teledyne Semiconductor, 1300 Terra Bella Avenue, Mountain View, California 94040 Telephone (415)968-9241.
14. Western Digital Corp., 19242 Red Hill Avenue, Box 2180, Newport Beach, California 92663 Telephone (714)557-3550.

III. CRT Terminals/Keyboards

1. Ann Arbor Terminals, Inc., 6107 Jackson Road, Ann Arbor, Michigan 48103
2. Applied Digital Systems, Inc., 100 Marcus Blvd. Hauppauge, N.Y. 11787
3. Beehive Terminals, Dept. G., 870 W. 2600 S., Salt Lake City, Utah 84119
4. Delta Data Systems Corporation, Woodhaven Industrial Park, Cornwells Heights, Pennsylvania 19020
5. Hazeltine Corporation, Pulaski, Road, Greenlawn, N.Y. 11740
6. Infoton, Inc., 2nd Avenue, Burlington, Massachusetts 01803
7. Lear Siegler, Inc., 714 North Brookhurst St., Anaheim, California 92803
8. Research, Inc., Box 24064, Minneapolis, Minnesota 55424
9. Sanders Data Systems, Ltd., Graphic Systems Marketing, Daniel Webster Highway-South, Nashua, N.H. 03060
10. Tektronix, Inc., Information Display D.V., P.O. Box 500, Beaverton, Oregon 97005

IV. DATA TABLETS

1. Electro Switch Corp, Dept. G., 167 King Ave., Weymouth, Maine 02188
2. Instronics, Inc., Dept. G. Bridge Plaza Suite, 204A, Ogdensburg, New York 13669
3. Summagraphics Corp., 398G Kings Hwy., Fairfield, Conn. 06430
4. Science Accessories Corporation, 970 Kings Highway West, Southport, Conn. 06490
5. Talas Sys. Inc., Dept. G., 7311 E. Evans Road, Scottsdale, Arizona 85260
6. Vector General, Inc., 8399G Topanga Canyon, Canoga Park, California 91304

V. FLOATING POINT SYSTEMS

1. Elsytec, Inc., 6150 Canoga Avenue, Woodland Hills, California
91364
2. Floating Point Systems, 10520 S.W. Carcade Blvd., Portland,
Oregon 97223

IVAM ANNUAL REPORT, SPRING 1976

Appendix B

Color Evaluation Results

Color Sample Evaluation Tables

for

COLOR-AID PACK

Geller Artist Materials, Inc.
116 East 27th Street
New York City 10016

Description of Test

Tests were performed on all 202 color swatches of the COLOR-AID package to determine R.G.B. values, as well as hue, saturation, and luminance values. The test set-up is shown in Figure A1-1. The view of the camera included a standard, calibrated gray scale and the color sample on a black velvet background. The electrical output of a scan line crossing the gray scale was displayed on the oscilloscope so that the gain of the system could be normalized for each sample as darker and lighter color samples were tested. Red, green, and blue values for each sample were read from the oscilloscope, as well as the luminance value. The vector scope provided the hues in angular degrees, and the saturation in vector magnitude.

The data listed in the following tables shows the R.G.B. values, hue, saturation, luminance, and an indicator for acceptability of use on standard TV. Luminance values specified are referenced to standard EIA Reflectance Chart with 10 = 100% white, 0 = absolute black, 1 = camera black, 2 = lowest picture black.

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
Y	115	108	36	40°	50	11	No
Y-T1	115	110	45	20°	40	10.5	No
Y-T2	115	112	53	18°	38	10.2	No
Y-T3	120	110	62	17°	35	10.5	No
Y-T4	115	110	68	16°	30	10.5	No
Y-S1	92	88	35	18°	38	9.0	No
Y-S2	65	63	30	18°	38	7.2	Yes
Y-S3	65	58	30	28°	26	6.8	No
YOY	110	90	33	48°	50	9.2	No
YOY-T1	112	90	53	50°	45	9.4	No
YOY-T2	112	94	50	50°	40	9.5	No
YOY-S1	68	65	40	20°	22	9.6	No
YOY-T3	113	95	54	50°	38	9.6	No
YOY-T4	115	100	60	34°	40	10.0	No
YOY-S2	78	75	52	19°	23	8.2	Yes
YOY-S3	83	83	60	16°	20	8.6	Yes
YO	110	76	30	50°	62	8.7	No
YO-T1	110	80	37	52°	57	8.9	No
YO-S1	70	60	28	32°	33	7.2	Yes
YO-T2	112	84	42	53°	50	9.0	No
YO-T3	112	88	50	53°	43	9.2	No
YO-T4	112	92	56	53°	38	9.5/9.4	No
YO-S2	60	58	30	24°	25	6.8	Yes

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
YO-S3	48	50	29	6°	18	5.2	Yes
OYO	108	62	26	59°	70	8.0	No
OYO-T1	104	62	30	60°	68	8.0	No
OYO-T2	108	67	38	60°	62	8.4	No
OYO-T3	112	78	49	60°	53	8.9	No
OYO-T4	115	84	53	59°	47	9.1	No
OYO-S1	70	58	42	48°	25	7.1	Yes
OYO-S2	80	68	51	46°	26	7.9	Yes
OYO-S3	85	74	53	41°	28	8.3	Yes
O	100	44	15	63°	78	7.0	No
O-T1	102	53	30	64°	70	7.6	No
O-T2	108	63	39	63°	64	8.2	No
O-T3	110	71	52	66°	57	8.6	No
O-T4	111	78	60	66°	57	9.0	No
O-S1	79	40	17	60°	58	5.5	Yes
O-S2	60	37	19	59°	38	4.7	Yes
O-S3	48	33	20	56°	27	4.0	Yes
ORO	94	36	17	78°	77	5.7	Yes
ORO-T1	101	49	37	70°	67	7.3	No
ORO-T2	106	57	44	70°	62	6.9	Yes
ORO-T3	110	66	43	70°	55	8.3	No
ORO-T4	111	70	63	72°	50	8.6	No
ORO-S1	70	50	46	71°	27	6.8	Yes

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
ORO-S2	70	55	50	68°	22	7.1	Yes
ORO-S3	86	70	65	76°	22	8.3	Yes
RO	93	43	19	71°	64	6.1	Yes
RO-T1	97	43	38	73°	64	7.0	Yes
RO-T2	101	55	48	73°	60	7.8	No
RO-T3	104	63	58	74°	53	8.2	No
RO-T4	108	66	62	74°	50	8.4	No
RO-S1	64	31	23	70°	42	5.3	Yes
RO-S2	43	27	20	67°	24	4.5	Yes
RO-S3	28	20	16	64°	14	3.4	Yes
ROR	86	30	23	74°	68	5.9	Yes
ROR-T1	95	45	44	76°	60	7.1	Yes
ROR-T2	96	49	49	77°	57	7.3	Yes
ROR-T3	100	56	57	78°	52	7.9	Yes
ROR-T4	102	63	64	78°	48	8.1	No
ROR-S1	69	42	53	100°	31	6.2	Yes
ROR-S2	74	50	56	86°	29	7.0	Yes
ROR-S3	81	57	66	90°	28	7.6	Yes
R	74	24	26	79°	58	5.1	Yes
R-T1	89	41	48	83°	56	6.8	Yes
R-T2	91	45	55	85°	52	7.1/7.2	Yes
R-T3	93	50	61	86°	50	7.4	Yes
R-T4	98	58	71	89°	43	8.0	No

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
R-S1	49	24	28	82°	21	4.1	Yes
R-S2	39	23	27	87°	20	4.0	Yes
R-S3	33	22	26	90°	16	3.8	Yes
RVR	60	23	42	96°	38	4.9	Yes
RVR-T1	77	34	52	91°	37	6.1	Yes
RVR-T2	83	42	60	94°	44	6.7	Yes
RVR-T3	88	52	73	100°	39	7.4	Yes
RVR-T4	91	57	75	96°	35	7.9	Yes
RVR-S1	65	40	55	100°	27	6.1	Yes
RVR-S2	78	47	60	94°	32	7.0	Yes
RVR-S3	82	58	74	102°	26	7.6	Yes
RV	53	18	46	110°	36	4.4	Yes
RV-T1	60	30	63	120°	35	5.6	Yes
RV-T2	73	40	74	119°	36	6.6	Yes
RV-T3	77	48	83	125°/124°	34	7.3	Yes
RV-T4	83	57	85	118°	30	7.7	Yes
RV-S1	47	20	48	118°	30	4.3	Yes
RV-S2	39	20	39	115°	23	4.2	Yes
RV-S3	33	23	40	134°	16	4.0	Yes
VRV	40	22	47	129°	23	4.3	Yes
VRV-T1	53	33	69	143°	28	5.5	Yes
VRV-T2	64	44	75	135°	28	6.7	Yes
VRV-T3	71	50	84	137°	28	7.2	Yes

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
VRV-T4	78	58	88	132°	17	7.7	Yes
VRV-S1	64	45	68	122°	23	6.6	Yes
VRV-S2	72	48	67	107°	26	7.0	Yes
VRV-S3	80	63	80	116°	21	8.2	Yes
V	33	19	47	148°	22	3.8	Yes
V-T1	50	35	70	152°	27	5.4	Yes
V-T2	58	46	80	156°	25	6.6	Yes
V-T3	67	54	88	155°	26	7.2	Yes
V-T4	70	58	91	158°	24	7.6	Yes
V-S1	40	22	45	126°	23	4.1	Yes
V-S2	44	28	33	88°	19	4.4	Yes
V-S3	54	38	34	70°	22	5.5	Yes
VBV	25	20	44	168°	18	3.7	Yes
VBV-T1	44	56	67	170°	22	5.5	Yes
VBV-T2	48	44	77	175°	23	6.0	Yes
VBV-T3	60	55	86	174°	23	7.1	Yes
VBV-T4	63	57	90	176°	24	7.3	Yes
VBV-S1	48	44	72	176°	22	6.0	Yes
VBV-S2	64	61	88	174°	21	7.5	Yes
VBV-S3	73	68	96	174°	21	8.0	Yes
BV	17	18	43	190°	20	3.5	Yes
BV-T1	40	39	76	186°	29	5.6	Yes
BV-T2	48	48	82	177°	28	6.3	Yes

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
BV-T3	57	55	88	174°	26	7.0	Yes
BV-T4	59	57	90	174°	26	7.3/7.2	Yes
BV-S1	19	23	42	200°	17	3.6	Yes
BV-S2	25	31	42	210°	15	4.0	Yes
BV-S3	27	34	42	235°	14	4.6	Yes
BVB	17	23	58	200°	32	3.7	Yes
BVB-T1	38	44	81	202°	33	5.8	Yes
BVB-T2	47	54	88	200°	32	6.8	Yes
BVB-T3	57	59	94	190°	28	7.3	Yes
BVB-T4	64	63	93	190°	24	7.7	Yes
BVB-S1	49	54	75	200°	20	6.7	Yes
BVB-S2	60	62	89	190°	22	7.3	Yes
BVB-S3	70	70	94	185°	18	8.2	Yes
B	18	27	57	205°	38	4.0	Yes
B-T1	32	46	87	212°	42	5.8	Yes
B-T2	42	56	93	213°	39	6.8	Yes
B-T3	47	63	96	213°	36	7.3	Yes
B-T4	53	67	103	211°	36	7.8	Yes
B-S1	18	29	60	200°	31	4.3	Yes
B-S2	16	23	51	210°	28	3.7	Yes
B-S3	17	23	40	206°	18	3.8	Yes
BGB	17	35	63	220°	35	4.7	Yes
BGB-T1	30	47	80	208°	40	6.0	Yes

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
BGB-T2	38	58	90	223°	41	7.0	Yes
BGB-T3	40	64	90	228°	40	7.3	Yes
BGB-T4	43	70	95	230°	41	7.6	Yes
BGB-S1	33	55	83	226°	40	6.5	Yes
BGB-S2	40	65	90	230°	40	7.3	Yes
BGB-S3	46	68	93	230°	37	7.6	Yes
BG	22	40	56	233°	29	5.0	Yes
BG-T1	26	53	77	230°	43	6.2	Yes
BG-T2	33	60	84	232°	42	6.8	Yes
BG-T3	38	66	90	233°	43	7.3	Yes
BG-T4	42	70	94	234°	43	7.6	Yes
BG-S1	20	40	57	233°	30	5.2	Yes
BF-S2	18	36	47	238°	25	4.6	Yes
BG-S3	17	32	46	232°	23	4.3	Yes
GBG	18	42	40	260°	26	4.8	Yes
GBG-T1	28	56	65	243°	33	6.3	Yes
GBG-T2	36	63	78	240°	38	6.9	Yes
GBG-T3	40	70	80	255°	38	7.4	Yes
GBG-T4	49	78	93	243°	38	8.2	Yes
GBG-S1	22	56	70	240°	35	6.3	Yes
GBG-S2	38	66	84	237°	39	7.3	Yes
GBG-S3	46	74	93	235°	39	7.8	Yes
G	18	48	32	278°	28	5.0	Yes

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
G-T1	27	63	54	267°	37	6.4	Yes
G-T2	33	70	62	267°	37	7.1	Yes
G-T3	44	80	74	265°	39	8.0	Yes
G-T4	54	88	83	265°	36	8.4	Yes
G-S1	22	45	37	276°	27	5.1	Yes
G-S2	20	39	36	267°	21	4.7	Yes
G-S3	18	32	32	267°	16	4.1	Yes
GYG	24	54	31	291°	31	5.7	Yes
GYG-T1	33	67	47	282°	36	6.7	Yes
GYG-T2	38	74	56	280°	35	7.3	Yes
GYG-T3	50	84	56	281°	35	8.2	Yes
GYG-T4	63	84	79	280°	32	8.9	Yes
GYG-S1	43	58	57	265°	16	6.6	Yes
GYG-S2	50	63	58	275°	15	7.1	Yes
GYG-S3	64	79	75	270°	16	8.3	Yes
YG	33	65	33	301°	34	6.6	Yes
YG-T1	40	73	48	290°	36	7.3	Yes
YG-T2	47	82	55	291°	34	7.8	Yes
YG-T3	58	90	63	297°	32	8.5	Yes
YG-T4	69	97	70	300°	29	9.0	Yes
YG-S1	43	60	42	303°	17	6.7	Yes
YG-S2	33	45	33	303°	14	5.5	Yes
YG-S3	30	34	33	No reading	No reading	4.6	No

IVAM COLOR CODING TABLE

Color Name	RGB VALUES			NTSC VALUES			Acceptable Color for TV
	Red	Green	Blue	Hue in Degrees	Saturation	Luminance (Gray value in %)	
YGY	50	78	34	320°	37	7.7	Yes
YGY-T1	60	88	43	323°	36	8.4	Yes
YGY-T2	66	93	52	322°	33	8.7	Yes
YGY-T3	74	96	58	325°	32	9.2	No
YGY-T4	80	102	59	334°	30	9.3	No
YGY-S1	60	72	40	346°	23	7.7	Yes
YGY-S2	66	73	48	355°	20	8.0	Yes
YGY-S3	80	89	57	351°	23	9.0	Yes
White	118	118	118	No reading	No reading	11.2	No
Gray-1	92	95	92	No reading	No reading	9.5	Yes
Gray-2	80	83	78	No reading	No reading	8.9	Yes
Gray-3	66	69	66	No reading	No reading	7.9	Yes
Gray-4	60	62	60	No reading	No reading	7.3	Yes
Gray-5	57	59	56	No reading	No reading	6.9	Yes
Gray-6	46	47	46	No reading	No reading	6.0	Yes
Gray-7	38	40	40	No reading	No reading	5.3	Yes
Gray-8	26	26	26	No reading	No reading	3.8	Yes
Black	14	14	14	No reading	No reading	1.7	Yes