

CP-6 PRELIMINARY DESIGN REVIEW

HONEYWELL

AGENDA

MONDAY, SEPTEMBER 19, 1977

8:00 COFFEE AND ROLLS

8:30 WELCOME

SHEL KLEE

9:00 INTRODUCTION AND INSTRUCTION OF THE BOARD

BOB SMITH

9:30 HISTORY AND BUSINESS OVERVIEW OF THE XEROX PROGRAM

HANK HAUGLAND

10:45 LADC ORGANIZATION AND RESPONSIBILITIES

SHEL KLEE

12:00 LUNCH

1:00 INTRODUCTION TO THE PDR

DICK LITSCHGI

CP-6 SOFTWARE FACTORY

JOHN CATOZZI

CP-6 PRELIMINARY DESIGN REVIEW

HONEYWELL

AGENDA

TUESDAY, SEPTEMBER 20, 1977

8:30	CP-6 SYSTEM STRUCTURE OVERVIEW	DOUG HEYING
9:45	PROGRAM-TO-PROGRAM INTERFACE PROGRAM-TO-SYSTEM INTERFACE	DOUG HEYING
10:00	MEMORY MANAGEMENT AND JOB STEP CONTROL	JOHN COLLINS
11:00	CLIMB-PMME, DISPATCHING, FAULTS, EXCEPTIONS	LINDA AVERY
12:15	LUNCH	
1:00	CONNECTIONS	SAM KEYS
2:00	FILE MANAGEMENT	PAT CRISMAN
3:30	USER-TO-SYSTEM INTERFACE	LARRY FELDMAN
4:30	T & D INTERFACE	DAVE MORGAN
4:50	MISCELLANEOUS INTERFACES	DAVE YOX
5:30	SUMMARY	DOUG HEYING

AGENDA

WEDNESDAY, SEPTEMBER 21, 1977

- | | | |
|-------|---|--|
| 8:30 | COMMUNICATIONS – INTRODUCTION
L6 FACTORY
COMMUNICATION SOFTWARE | CHUCK MARTIN
TERRY COX
TOM MELTON |
| 10:30 | LOS ANGELES DEVELOPED PROCESSORS
PL-6
BASIC
FORTRAN
APL
TEXT AND SUMMARY | WING WONG
RICH HANSON
JIM GRIFFIN
JOHN FLINT
TOM MARTIN
WING WONG |
| 12:30 | LUNCH | |
| 1:15 | PHOENIX DEVELOPED PROCESSORS OVERVIEW
COBOL 74; PL/I
SORT MERGE
ASSEMBLER | GEORGE MANN
DUANE DAVIS
JOHN WERTZ
GEORGE MANN |
| 3:15 | LOS ANGELES DEVELOPED PROCESSORS
I-D-S/II, IDP/MANAGE
RPG-II | JOHN ROBERTSON
DOUG CHESTER |
| 4:00 | TEST PLAN AND CONVERSION PLAN | HERB GESHWIND |
| 4:30 | STAGING AND SCHEDULING – – THE WORK PLAN | DICK LITSCHGI |

CP-6 PRELIMINARY DESIGN REVIEW

HONEYWELL

AGENDA

THURSDAY, SEPTEMBER 22, 1977

8:30 LADC SUMMARY – RISKS AND ISSUES

DICK LITSCHGI

9:30 GENERAL QUESTION PERIOD

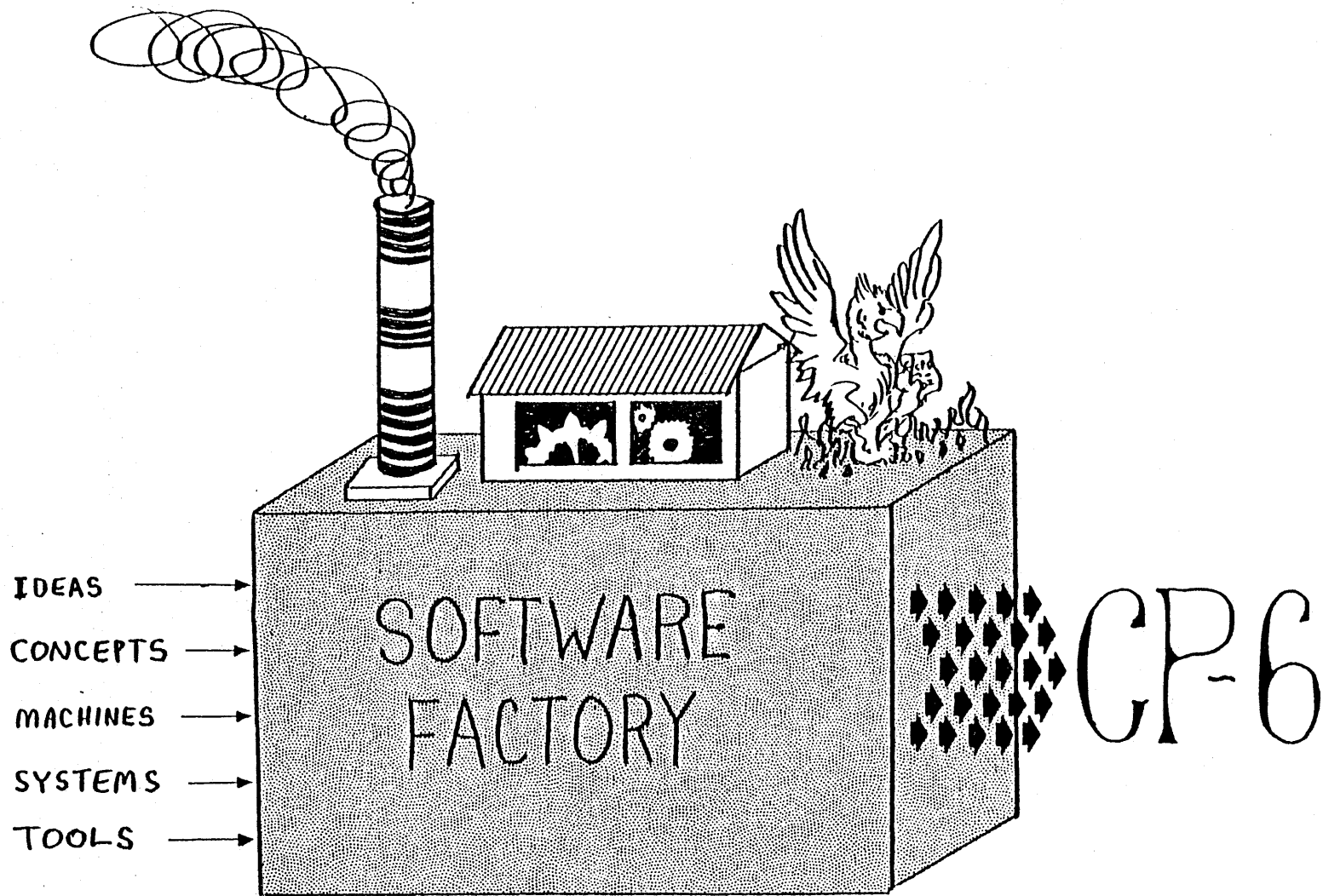
– BOARD DISCUSSION AND RISK DETERMINATION

12:00 LUNCH

– MORE BOARD DISCUSSION

9/19/77 EB

CP-66



SOFTWARE FACTORY

- ▶ REQUIREMENTS - WHAT SHOULD IT PROVIDE ?
- ▶ HISTORY - HOW WE GOT TO WHERE WE ARE
- ▶ SYSTEM IMPLEMENTATION LANGUAGE - WHAT IS PL-6 ?
- ▶ COMPONENTS OF THE FACTORY - SOFTWARE TO MAKE SOFTWARE
- ▶ USE OF THE FACTORY - HOW IT ALL WORKS TOGETHER
- ▶ TRANSPORT OF THE FACTORY TO CP-6 BASE SYSTEM
- ▶ EXPERIENCE TO DATE - HOW WELL DOES IT WORK ?

REQUIREMENTS OF SOFTWARE FACTORY

- ▶ LANGUAGES TO SUIT NEEDS OF SYSTEM IMPLEMENTATION
- ▶ TOOLS TO PUT PROGRAMS TOGETHER AND ONTO TARGET MACHINE
- ▶ DEBUGGING AIDS TO HELP GET CODE WORKING SOONER
- ▶ BASE SYSTEM WITH CONVENIENT, CONTROLLED ACCESS and FILE SYSTEM
- ▶ SUFFICIENT HARDWARE TO PROVIDE HOME FOR SOFTWARE FACTORY
- ▶ EFFICIENT ENVIRONMENT TO PROMOTE PRODUCTIVITY and HELP RETAIN PROGRAMMERS' SANITY

A LITTLE HISTORY

- ▶ PL/1 – TOO MUCH OF A GOOD THING
- ▶ MULTICS – NICE TOOLS, WRONG MACHINE
- ▶ GCOS – ASSEMBLER, RIGHT MACHINE
- ▶ PL-6 – SMALL, CONTROLLABLE
- ▶ CP-V – FAMILIAR, AVAILABLE
- ▶ BMAP – GMAP ON CP-V

PL-6 THE SYSTEM IMPLEMENTATION LANGUAGE

- ▶ PL/1 - LIKE SYNTAX
- ▶ BLOCK STRUCTURED
- ▶ SIMPLE DATA TYPES
- ▶ MINIMAL RUN-TIME ROUTINES
- ▶ NO HIDDEN OVERHEAD
- ▶ INTERFACES TO SYSTEM SERVICES
- ▶ FACILITATES CODING IN NSA ENVIRONMENT
- ▶ USES CAPABILITIES OF L66 INSTRUCTION SET

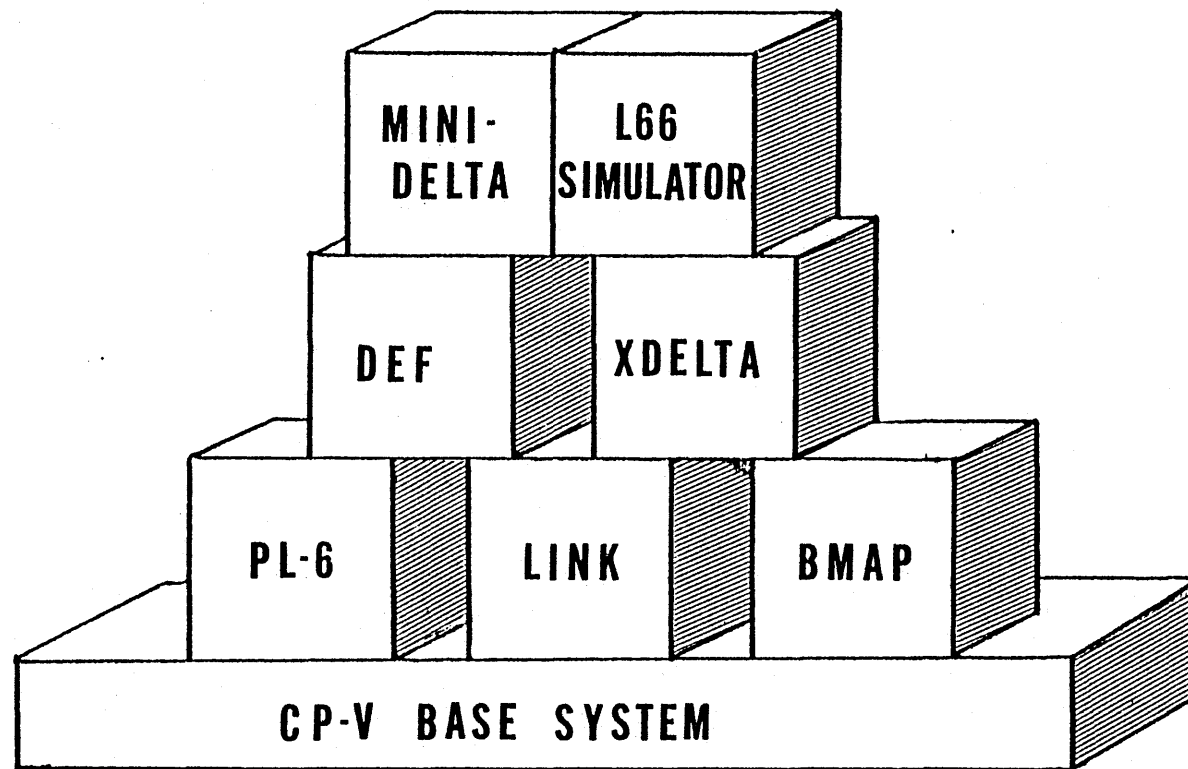
H A R D W A R E

- ▶ DUAL 560 - PRIMARY FACTORY SYSTEM THROUGH 1Q79
- ▶ DUAL SIGMA6 - OVERFLOW/BAC KUP SYSTEM FOR DUAL 560
- ▶ TWO L66's - FOR HANDS ON DEBUGGING THROUGH 3Q78
- ONE BECOMES FACTORY SYSTEM 4Q78
- ▶ TERMINALS - IN OFFICES OF ALL PROGRAMMERS and A FEW
TERMINAL ROOMS
- ▶ PHOENIX FACTORY SYSTEM - SIGMA6 and L66 AVAILABLE

CP-V BASE SYSTEM

- ▶ CONTROLLED DEPENDABLE FILE SYSTEM
- ▶ EDIT - KNOWN ENTITY
- ▶ DEBUGGER - ALLOWS FOR FAST DEVELOPMENT OF BUILDING BLOCKS
- ▶ FORTRAN - MOST FACTORY SOFTWARE CODED IN FORTRAN

**BUILDING BLOCKS OF THE FACTORY
(SOFTWARE TO BUILD SOFTWARE)**



CP-6 OBJECT LANGUAGE

- ▶ DEVELOPED TO SPECIFICALLY FIT CP-6 ENVIRONMENT
- ▶ PROVIDES FOR STANDARD CALLING SEQUENCES
- ▶ ALLOWS IMPLEMENTATION OF A COMMON DEBUGGER FOR ALL LANGUAGES
- ▶ FACTORY FORMAT ACCOMODATES 36-BIT WORD SIZE WITHOUT ALTERING
FILE RECORD LAYOUT (2 WORDS = 1 WORD)

PL-6

- ▶ **BASED ON PL/H (BASED ON CONTROL FORTRAN)**
- ▶ **CODED IN FORTRAN**
- ▶ **LIVES ON CP-V SYSTEM**
- ▶ **PRODUCES CP-6 OBJECT UNITS WITH DEBUG SCHEMA**

BMAP

- ▶ L66 ASSEMBLER (WITH NSA, EIS)
- ▶ CODED IN FORTRAN
- ▶ LIVES ON CP-V
- ▶ PRODUCES CP-6 OBJECT UNITS
- ▶ GMAP EQUIVALENT WITH EXTENSIONS

LINK

- ▶ LINKER WHICH CREATES PROGRAM RUN UNITS FROM OBJECT UNITS
- ▶ CREATES ALL EXECUTABLE PROGRAMS (BOOTSTRAP, MONITOR, USER PROGRAMS)
- ▶ SUPPORTS DEBUG SCHEMA
- ▶ CREATES LOAD MAP OF THE RUN UNIT
- ▶ CODED IN FORTRAN
- ▶ LIVES ON CP-V SYSTEM
- ▶ SUBSET OF CP-6 LINKER FUNCTIONALITY

L66 SIMULATOR

- ▶ PROVIDES CP-6 / L66 SIMULATION ON CP-V
- ▶ USES LINK BUILT RUN UNITS FOR INPUT
- ▶ INCLUDES
 - ▶ L66 INSTRUCTION SIMULATION (INCLUDING EIS & NSA)
 - ▶ INTERACTIVE SYMBOLIC DEBUGGER
 - ▶ MINIMAL SYSTEM SERVICES FOR I / O
- ▶ ALLOWS CHECKOUT OF CODE BEFORE SYSTEM INTEGRATION
- ▶ CODED IN METASYMBOL (CP-V ASSEMBLER)

DEF

- ▶ CREATES THE SYSTEM BOOT TAPE (PO TAPE)
- ▶ BOOTSTRAP, MINI-DELTA, MONITOR, XDELTA, GHOST1 PROGRAM IMAGES
EXTRACTED FROM LINK BUILT RUN UNIT FILES
- ▶ MONITOR DEBUG SCHEMA SORTED BY FCG AND WRITTEN TO TAPE
- ▶ SYSTEM FILES WRITTEN TO TAPE IN CP-6 TAPE FORMAT (RUN UNIT FILES & CP-V FILES)
- ▶ CONVERTS 32-BIT WORD FORMATS TO 36-BIT FORMAT
- ▶ CODED IN METASYMBOL

MINI - DELTA

- ▶ RUNS ON L66 AS STANDALONE HOST DEBUGGING SYSTEM
- ▶ PROVIDES INTERACTIVE DEBUGGING OF CP-6 MONITOR AT SYSTEM CONSOLE
- ▶ READS PO TAPE, INITIALIZES MPC'S, SETS UP MONITOR'S RUNNING ENVIRONMENT
- ▶ DEBUGGER PROVIDES
 - ▶ SYMBOLIC SEGMENT SPECIFICATION
 - ▶ ABSOLUTE ADDRESSING WITHIN SEGMENT (ALLOWS SYMBOL DEFINITION)
 - ▶ INSTRUCTION BREAKPOINTS (UP TO SEVEN)
 - ▶ MEMORY DUMPS TO CONSOLE OR PRINTER
 - ▶ DISPLAY AND MODIFICATION OF MEMORY AND REGISTERS
- ▶ READS INPUT COMMANDS FROM SYSTEM CONSOLE, PO TAPE, OR CARD READER
- ▶ CODED IN BMAP

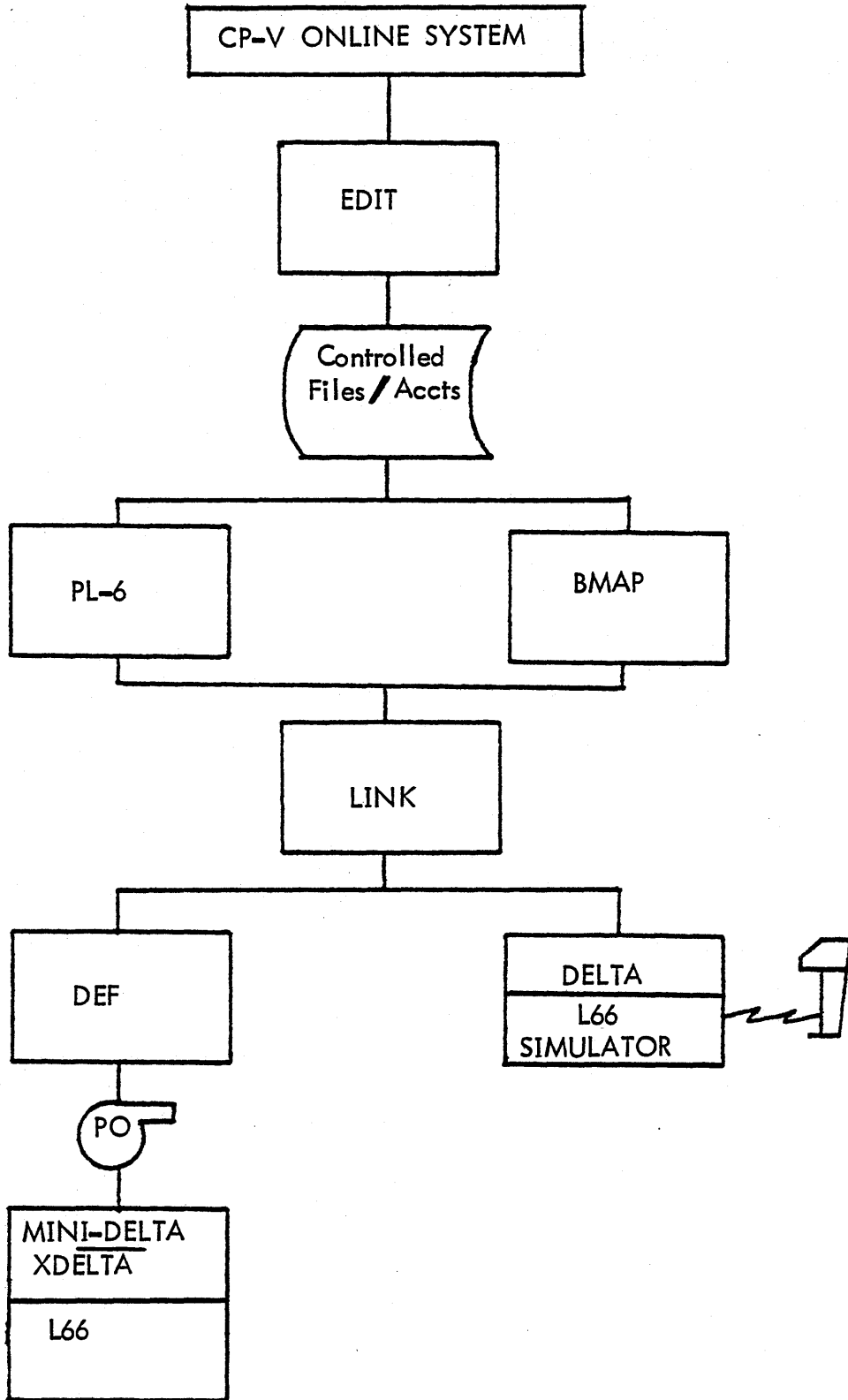
X DELTA

- ▶ RUNS ON L66 AS STANDALONE HOST DEBUGGING SYSTEM
- ▶ POWERFUL INTERACTIVE DEBUGGER - REPLACES MINI-DELTA
- ▶ UTILIZES PL-6 GENERATED PROGRAM SCHEMA
- ▶ INITIAL VERSION INTERFACES WITH MINI-DELTA
- ▶ PROVIDES ▶ STATEMENT OR INSTRUCTION MODE SINGLE STEP EXECUTION
 - ▶ UNLIMITED BREAKPOINTS (WITH CONDITIONS, ATTACHED COMMANDS)
 - ▶ DISPLAYS DATA, INSTRUCTIONS IN SPECIFIABLE FORMAT
 - ▶ PERFORMS DISK, TAPE I/O TO FACILITATE FILE SYSTEM CHECKOUT
 - ▶ ALLOWS DISPLAY, MODIFICATION OF ALL MEMORY AND REGISTERS
- ▶ CODED IN PL-6 (SOME BMAP)
- ▶ AVAILABILITY SCHEDULED FOR NOVEMBER 1977

MISCELLANEOUS TOOLS

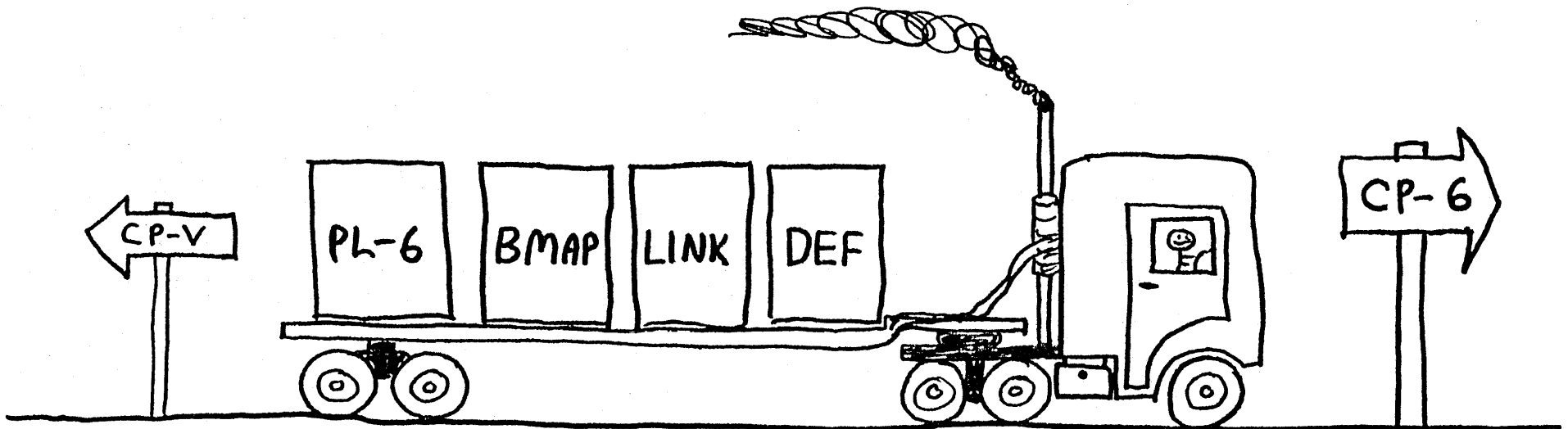
- ▶ EXTRACT
- ▶ DECOMMENT
- ▶ DRAW
- ▶ XREF
-
-
-
- ▶ (LIST GROWS AS NEEDS ARISE)

THE FACTORY ON CP-V



TRANSPORTING FACTORY TO CP-6

- ▶ ALL CP-6 FACTORY BUILDING BLOCKS CODED IN EITHER PL-6 OR FORTRAN
- ▶ PL-6 BLOCKS ARE CREATED ON CP-V FACTORY
- ▶ CP-6 CODEGEN FOR CP-V FORTRAN AVAILABLE JUNE '78
 - ▶ RUNS ON CP-V
 - ▶ PRODUCES CP-6 OBJECT UNITS
- ▶ FORTRAN BLOCKS ARE COMPILED WITH THIS MODIFIED FORTRAN THEN LINKED AND TRANSPORTED JUST AS THE PL-6 BLOCKS



BLOCKS TO BE RE-CODED IN PL-6

- ▶ LINK (FULL IMPLEMENTATION)
- ▶ DEF

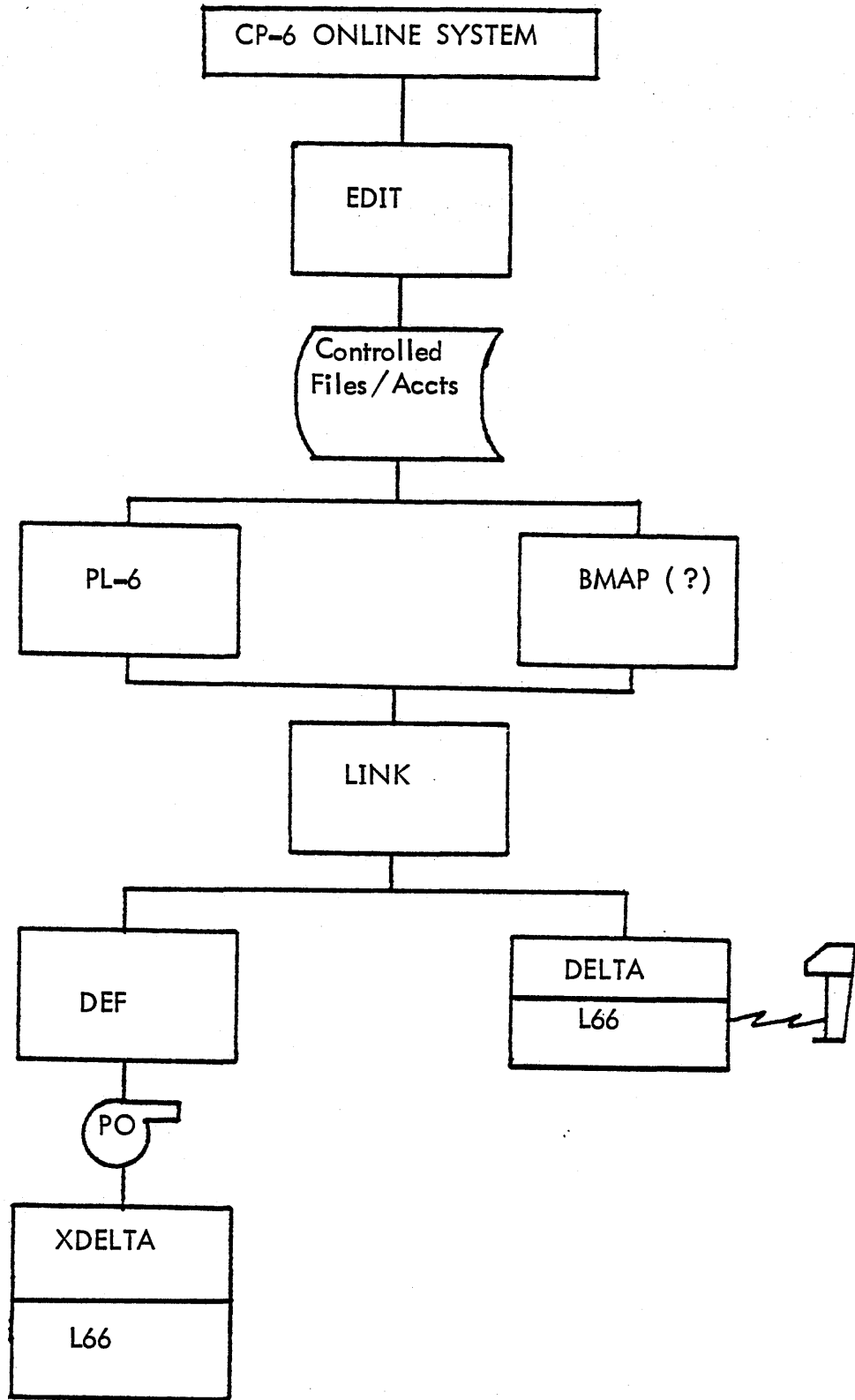
BLOCKS TO BE CODED IN PL-6

- ▶ EDIT
- ▶ XDELTA
- ▶ DELTA

BLOCKS TO BE MOVED VIA FORTRAN VEHICLE

- ▶ PL-6 (MAIN REASON FOR CREATING VEHICLE)
- ▶ BMAP (???)

THE FACTORY ON CP-6



EXPERIENCE TO DATE

(Or How To Lick a Camel)

TWO MAJOR HUMPS -

- ① LEARNING INTIMACIES OF NSA ADDRESSING
- ② LEARNING TO CODE IN NEW (HIGHER-LEVEL) LANGUAGE

FACTORY TOOLS AIDED IMMENSELY

- ▶ MINI-DELTA – HANDS ON WITH L66 EASY TO SEE WHATS HAPPENING
- ▶ SIMULATOR – ABILITY TO SEE CODE WORK

WHERE ITS ALL GOTTEN US

- ▶ MINI-DELTA PATH (WITH L66)
 - BOOT PROCESS
 - MEMORY PROCESS
 - SCHEDULER
 - FAULT HANDLER
 - PMME HANDLER (SERVICE DECODER)
 - IOQ-IOS
- ▶ SIMULATOR PATH
 - FILE MANAGEMENT
 - SERVICE PROCESSORS
- ▶ PL-6 – HOW HAS IT HELPED

ON SCHEDULE AND MOVING FORWARD

CP - 6

- SYSTEM STRUCTURE
- PROGRAM to PROGRAM INTERFACE
- USER to SYSTEM INTERFACE
- PROGRAM to SYSTEM INTERFACE
- INITIALIZATION, RMA, PERFORMANCE

CP-6 IS A SYSTEM

- STRONG INTERFACES
- MONITOR, PROCESSORS WORK TOGETHER
- FUNCTIONS NOT DUPLICATED (EFFORT or STORAGE)
- ONENESS OF THE SYSTEM
- FULL PROTECTION
- PL-6 RELATIONSHIP TO SYSTEM

STRENGTH OF CP-6 IS IN ITS INTERFACES

- USER STRUCTURE
- SYSTEM SERVICE INTERFACE
- PROGRAM BINDING INTERFACE
- PROGRAM CALLING INTERFACE
- USER to SYSTEM INTERFACE

SYSTEM SERVICE INTERFACE

- UNIFORM ABSTRACT INTERFACE
- ISOLATES USER PROGRAMS FROM HARDWARE/SYSTEM SOFTWARE
- ENABLES COMPLETE DEVICE INDEPENDENCE
- MAKES FILES INTERCHANGEABLE AMONG LANGUAGES
- CLIMB is EXCELLENT VEHICLE

USER STRUCTURE (DOMAINS)

- USER DOMAIN
- ASL DOMAIN
- COMMAND PROCESSOR DOMAIN
- DEBUGGER DOMAIN
- PLUS MONITOR DOMAIN via PMME

USER DOMAIN

- JOB INFORMATION TABLE (JIT)
- READ ONLY SEGMENT (ROS)
 - Data Control Blocks (DCB)
 - Task Control Block (TCB)
- INSTRUCTION SEGMENT (IS)
- DYNAMIC SEGMENT (DS1 - DS8)

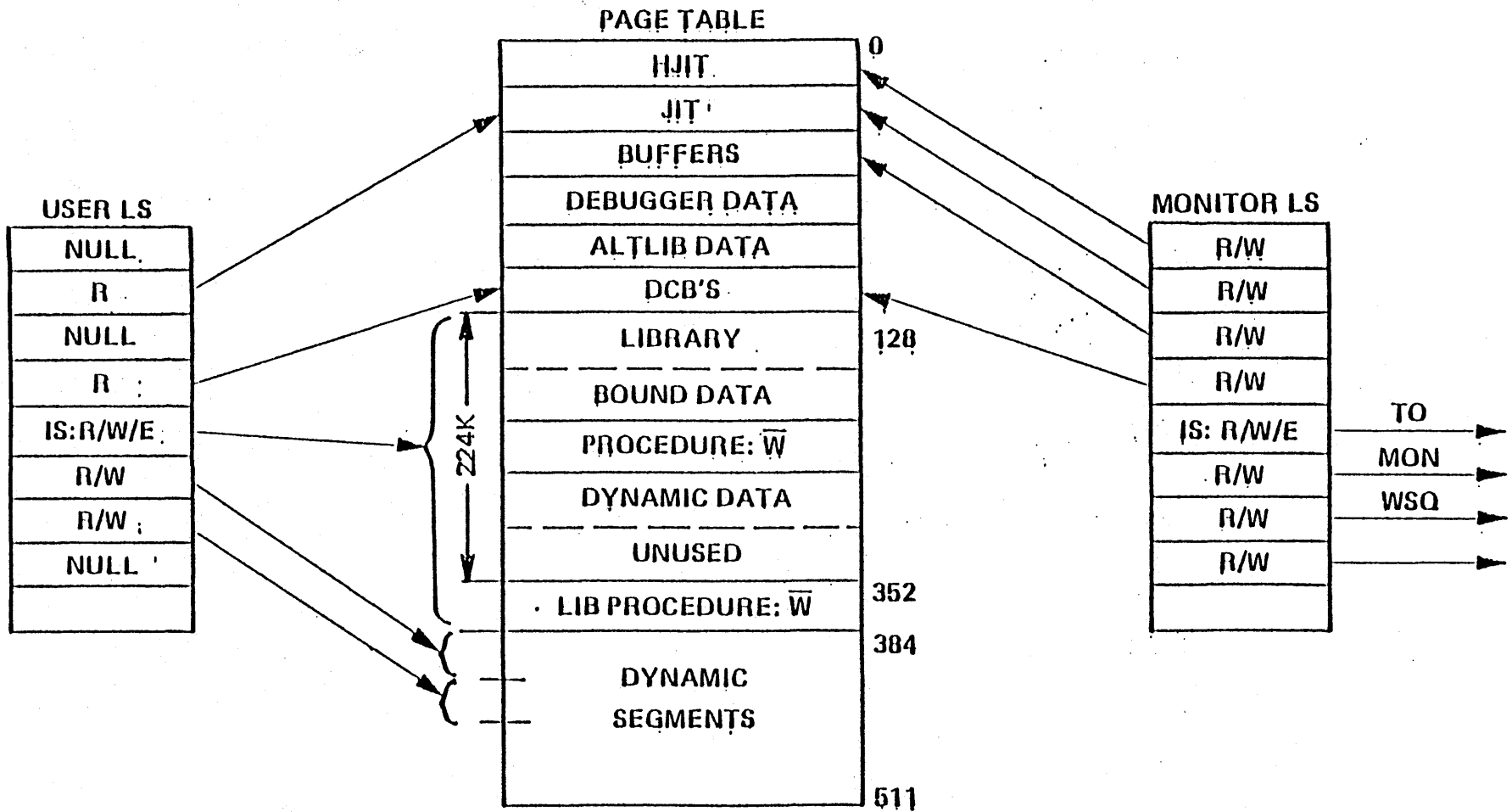
INSTRUCTION SEGMENT

- STATIC DATA
- PROCEDURE and CONSTANTS
- DYNAMIC DATA
- SHARED RUNTIME LIBRARY

USER VIRTUAL SPACE

- 'THE USER' FROM STANDPOINT of MONITOR CONTROL
- CONTAINS ALL 'GLUE' HOLDING IT TOGETHER
 - Page Table
 - Linkage Segments
 - Safe Store Stack
 - Argument/Parameter Segments
- COLLECTS ALL PHYSICAL MEMORY ALLOCATED TO THIS USER
- LOCATES BUT DOES NOT CONTAIN SHARED PROCEDURE
 - Shared Processor
 - Runtime Library

USER VIRTUAL ADDRESS SPACE

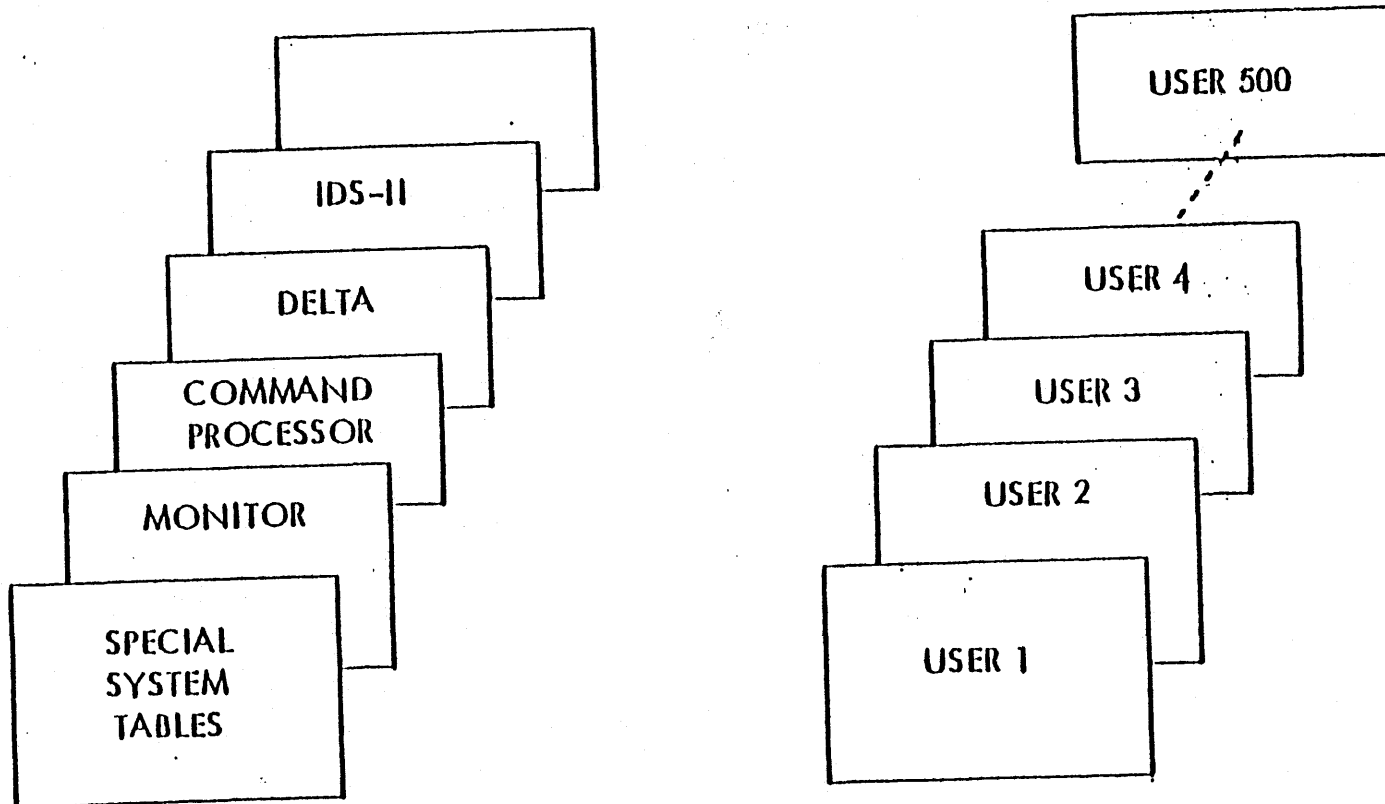


CP-6 MEMORY UTILIZATION

- CP-V BASED ON HIGH SPEED SWAPPER AND NEW INVENTIONS NOT TOLERABLE
- ROTATING MEMORY AND CCD'S CONSIDERED
- EVALUATION OF RAM COST TREND LED TO NOVEMBER '76 DECISION
 - USE 16K RAM IN LIEU OF SWAPPER
 - DECISION IRREVOCABLE WITHOUT SCHEDULE IMPACT
- CURRENT STATUS SHOWS NO SIGNIFICANT CHANGES
 - MUCH TOO LATE FOR ANY SWAPPER DEVELOPMENT
 - PROJECTED MEMORY REQUIREMENTS REMAIN THE SAME
 - 4K CHIP COST TREND BETTER THAN TARGET (BUT NOT A SOLUTION)
 - BUT WILL VOLUMES DEVELOP FOR 16K CHIPS

9/19/77

SYSTEM VIRTUAL ADDRESS SPACE



MONITOR DOMAIN

- USER JIT, ROS, HJIT, FILE BUFFERS, T STACK, PAGE TABLE
- USER PARAMETER SEGMENT
- MONITOR IS
- MONITOR DS's
- REAL MEMORY

MULTIPROCESSING

- USERS RUN STRICTLY in USER CONTEXT - ANY CPU
- MOST MONITOR SERVICES RUN in USER CONTEXT - ANY CPU
- FAULT HANDLER RUNS in USER/CPU CONTEXT - ANY CPU
- SCHEDULER/PHYSICAL I/O+MONITOR SERVICES WHICH USE CPU GLOBAL DATA RUN ON MASTER ONLY

MONITOR MEMORY USAGE

- CPU SPECIFIC (UNIQUE COPIES PER CPU)
 - MONITOR JIT and HJIT
 - Page Table Directory, Page Table, TSTACK
 - STATIC DATA Protected by SOFT DISABLE
- CPU GLOBAL (SAME COPY USED BY ALL CPU's)
 - STATIC DATA
 - REAL (Allocated At Boot Time)
 - DYNAMIC SEGMENTS
 - DYNAMIC REAL (Will be in some Users PT for REF)
 - PROCEDURE and CONSTANTS

ONENESS OF THE SYSTEM

- ONE KIND of JOB
- ONE CPU SCHEDULER
- ONE COMMAND LANGUAGE, DEBUGGER
- ONE FILE MANAGEMENT SYSTEM

PL-6 RELATIONSHIP TO SYSTEM

- LANGUAGE BUILT TO FIT SYSTEM, NOT VICE VERSA
 - Operating System is the Runtime 'Library'
- FACILITIES INCLUDED TO FACILITATE
 - Building the system
 - Using the system
- DOES NOT PROVIDE HIDDEN CONTROL MECHANISMS
- SYSTEM IS NOT PREJUDICED TO ANY LANGUAGE
 - But Provides Facilities Necessary to Implement All

CP-6 IS A SYSTEM

- STRONG INTERFACES
- MONITOR, PROCESSORS WORK TOGETHER
- FUNCTIONS NOT DUPLICATED (EFFORT or STORAGE)
- ONENESS OF THE SYSTEM
- FULL PROTECTION
- PL-6 RELATIONSHIP TO SYSTEM

STRENGTH OF CP-6 IS IN ITS INTERFACES

- USER STRUCTURE
- SYSTEM SERVICE INTERFACE
- PROGRAM BINDING INTERFACE
- PROGRAM CALLING INTERFACE
- USER to SYSTEM INTERFACE

PROGRAM BINDING INTERFACE

- OBJECT LANGUAGE COMMON FOR ALL LANGUAGES
- DEBUG SCHEMA INCLUDED PERMITTING COMMON DEBUGGER
- ORIENTED TO SHARED PROCEDURE ENVIRONMENT
- PERMITS GENERAL LINK TIME BINDING

FEATURES OF OBJECT LANGUAGE/LINKER

- GENERAL RELOCATION of FIELDS
- COMPLETE DESCRIPTION of VARIABLES/PROCEDURES
- DETECTION of PROCEDURE DEFINITION/CALL MISMATCH
- SYSTEMIC DEFINITIONS SUPPLIED by LINKER
- PERMITS GENERAL LINK TIME BINDING to PROMOTE MODULAR PROGRAMMING
- DESIGNED FOR EASE of GENERATION AND SPEED of LINKING
- LINKER PRODUCES RUN UNIT

FEATURES OF RUN UNIT

- SAME FORMAT USED FOR ALL TYPES PROGRAM
- EXECUTABLE FORM CONTAINING STATIC,
PROCEDURE, DCB's and TCB
- IDENTIFIES REQUIRED LIBRARY and ASL
- STANDARD FILE, of course

PROGRAM CALLING INTERFACE

- STANDARD SYSTEM CALLING SEQUENCE
- ACCOMODATES NEEDS of ALL LANGUAGES
- FACILITATES MIXED LANGUAGE PROGRAMS
- DESIGNED FOR EFFICIENT FORMAL INTERFACE
Among PROGRAMS In a RUN UNIT
- PROMOTES COMMON LIBRARY ROUTINES

CALLING SEQUENCE ATTRIBUTES

- DESIGNED For NSA ENVIRONMENT
- ORIENTED TO PURE PROCEDURE ENVIRONMENT
- CONTAINS INFORMATION USEFUL to DEBUGGER
- INTEGRATED WITH PL/1, PL-6 STACK FRAME MANAGEMENT
- ENCOMPASSES LIBRARY FUNCTION CALL FORMAT

CP-6 MEMORY MANAGEMENT

- PURPOSE
- ORGANIZATION
- OVERVIEW OF VIRTUAL MEMORY STRUCTURE
- DOMAINS – INTERRELATIONSHIP OF ADDRESS SPACES
- INTERNAL FUNCTIONS
- PROGRAM INTERFACE

MEMORY MANAGEMENT IS BUILT IN LEVELS

- PHYSICAL PAGE ALLOCATION
- PAGE TABLE MANIPULATION
- WSQ VIRTUAL PAGE ALLOCATION
- LINKAGE SEGMENT MANIPULATION
- SEGMENT – RELATIVE PAGE ALLOCATION
- ABSTRACTIONS BUILT ON ABOVE CAPABILITIES:
 - DATA SEGMENT ALLOCATION
 - "DYNAMIC" DATA ALLOCATION
 - FILE MANAGEMENT BUFFER ALLOCATION

USER WSQ

JIT, TSTACK, ACCOUNTING	
HJIT, LINKAGE SEGMENTS, SAFESTORE STACK, PARAMETER STACK	
FILE (& COOP) BUFFERS	
TCB, ECCB, TREE, DCBs	
LIBRARY AND USER DATA	
USER PROCEDURE	INSTRUCTION SEGMENT (256K)
DYNAMIC DATA	

SHARED RUN-TIME LIBRARY	
AUTOMATIC DATA, COMMON DATA, USER DATA SEGMENTS	
SPECIAL SHARED PROCESSOR DATA SEGMENTS	

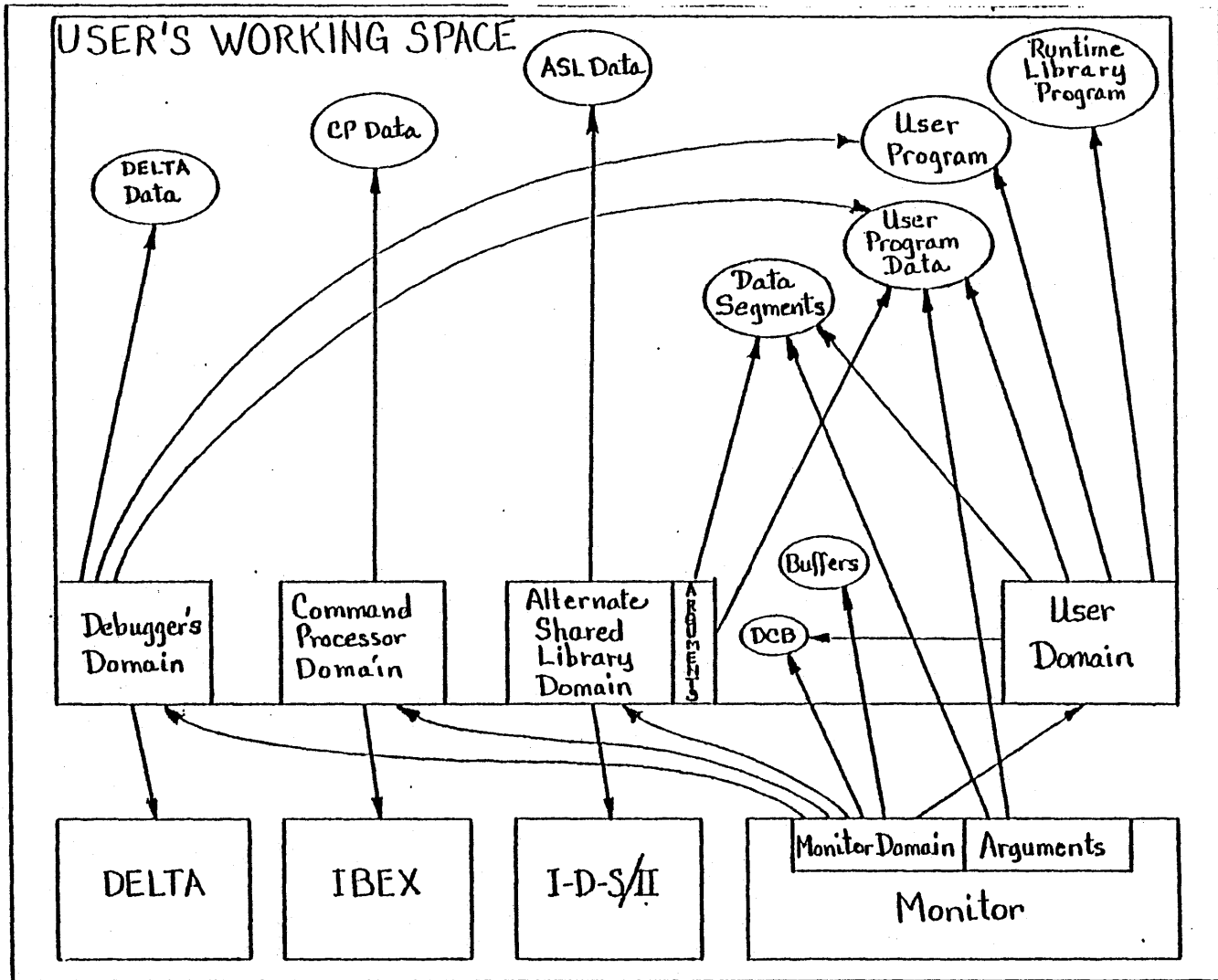
USER AND MONITOR VIRTUAL MEMORY LAYOUT

USER WSQ

MONITOR WSQ

JIT, TSTACK, ACCOUNTING	JIT, TSTACK, ACCOUNTING
HJIT, LINKAGE SEGMENTS, SAFESTORE STACK, PARAMETER STACK	HJIT, LINKAGE SEGMENT , SAFESTORE STACK, PARAMETER STACK
FILE (& COOP) BUFFERS	MONITOR "WINDOW" AREA
TCB, ECCB, TREE, DCBs	MONITOR INITIALIZATION - TIME TABLES
LIBRARY AND USER DATA	MONITOR DATA
USER PROCEDURE	INSTRUCTION SEGMENT (256K)
DYNAMIC DATA	MONITOR PROCEDURE (All pages marked housekeeping)
SHARED RUN-TIME LIBRARY	MONITOR DATA SEGMENTS
AUTOMATIC DATA, COMMON DATA, USER DATA SEGMENTS	RESERVED SPACE (64 pages)
SPECIAL SHARED PROCESSOR DATA SEGMENTS	

CP-6 DOMAINS OF REFERENCE



MEMORY MANAGEMENT INTERNAL FUNCTIONS

- GET and RELEASE PHYSICAL PAGE
- GET and RELEASE STOLEN PHYSICAL PAGE
- GET and RELEASE I/O CACHE PAGE

- MAP PHYSICAL PAGE TO VIRTUAL PAGE
- SET ACCESS CONTROL FLAGS FOR VIRTUAL PAGE
- SET SOFTWARE CONTROL FLAGS FOR VIRTUAL PAGE

- GET and RELEASE VIRTUAL PAGE
- CHANGE VIRTUAL MAP
- SET MEMORY PROTECTION
- EXCHANGE I/O CACHE PAGE FOR ONE CURRENTLY IN PAGE TABLE

MORE MM INTERNAL FUNCTIONS

- READ and MODIFY LINKAGE SEGMENT DESCRIPTORS
- GET and RELEASE SEGMENT - RELATIVE PAGES
- GET and RELEASE DATA SEGMENT SPACE
- GET and RELEASE DYNAMIC DATA PAGES
- GET DYNAMIC DATA LIMITS
- GET A FILE MANAGEMENT BUFFER

PROGRAM INTERFACE TO MEMORY MANAGEMENT

- M\$GDS - GET DATA SEGMENT SPACE
- M\$FDS - FREE DATA SEGMENT SPACE
- M\$GDP - GET DYNAMIC DATA PAGES
- M\$FDP - FREE DYNAMIC DATA PAGES
- M\$GDDL - GET DYNAMIC DATA LIMITS
- M\$GVP - GET VIRTUAL PAGE
- M\$FVP - FREE VIRTUAL PAGE
- M\$STLPP - STEAL PHYSICAL PAGE
- M\$RSPP - RELEASE STOLEN PAGE
- M\$CVM - CHANGE VIRTUAL MAP
- M\$SMPT - SET MEMORY PROTECTION
- M\$SSC - SET SOFTWARE CONTROL FLAGS

CP-6 EXECUTION CONTROL

- CONCEPTS - USERS, PROGRAMS, JOB STEPS, COMMAND PROCESSORS
- PURPOSE
- BASIC FUNCTIONS
- PROGRAM INTERFACE
- EXAMPLE OF JOB STEP CYCLE

CP-6 JOB STEP CONCEPTS

- A USER IS NOT A PROGRAM
- THE INITIAL USER CONSISTS OF A
 - JIT
 - HJIT
 - INITIAL READ-ONLY SEGMENT
- JOB STEP EXTENDS FROM PROGRAM INITIATION TO RUNDOWN
- COMMAND PROCESSOR ALLOWS SPECIFICATION OF JOB STEPS
- ALL USERS HANDLED BY SAME EXECUTION CONTROL ROUTINES

PURPOSE OF EXECUTION CONTROL

- JOB STEP INITIATION
- JOB STEP TERMINATION
- INTRA-JOB STEP FUNCTIONS
 - OVERLAYS
 - LOAD-AND-LINK
 - ASSOCIATE SHARED LIBRARIES, ASL, OR DEBUGGER

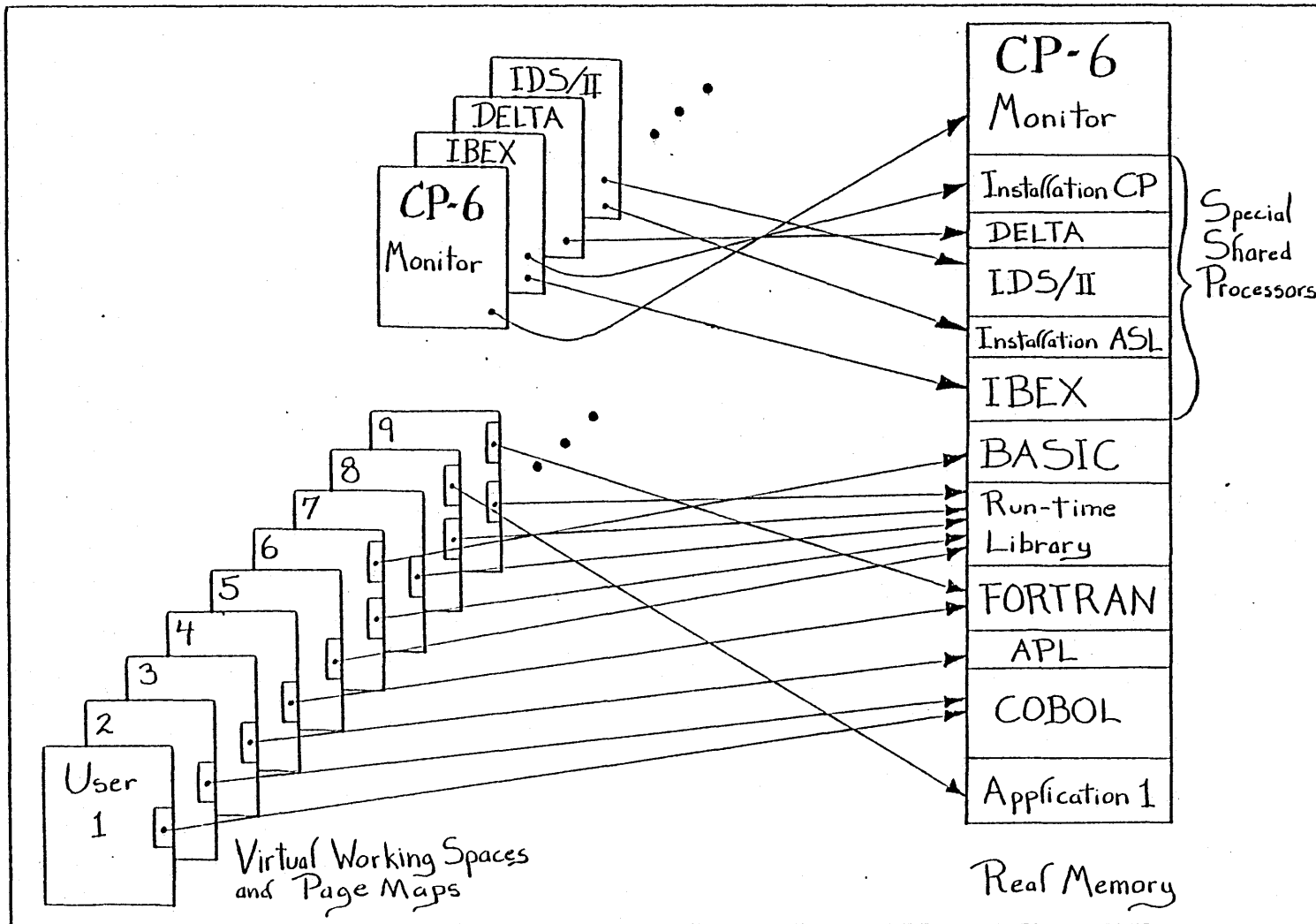
EXECUTION CONTROL BASIC FUNCTIONS

- INITIATE OR CONTINUE A JOB STEP
- ASSOCIATE COMMAND PROCESSOR
- ASSOCIATE A SPECIFIED PROGRAM/PROCESSOR
- TERMINATE A PROGRAM-EXIT, ERROR, ABORT
- RUNDOWN A USER - TERMINATE A JOB STEP
- LOAD-AND-LINK TO ANOTHER PROGRAM
- LOAD AN OVERLAY FROM A RUN UNIT
- PROVIDE EXIT CONTROL DISPATCHING

SHARING PROCEDURE

- SHARABILITY DETERMINED AT JOB STEP INITIATION
- SHARED PROCESSOR INITIAL DATA, READ ONLY SEGMENT
ALLOCATED TO INDIVIDUAL USER's WSQ
- SHARED PROCESSOR PROCEDURE MAPPED INTO EVERY USER's WSQ
- UNSHARED PROGRAMS READ ENTIRELY INTO MEMORY ALLOCATED
TO INDIVIDUAL WSQ's

CP-6 SHARED PROCESSORS



PROGRAM INTERFACE

- M\$CPEXIT - INITIATES NEW JOB STEPS
- M\$RUND - TERMINATES A JOB STEP - RUNS DOWN A USER

- M\$EXIT - PROGRAM NORMAL EXIT
- M\$ERR - PROGRAM ERROR EXIT
- M\$XXX - PROGRAM ABORT EXIT

- M\$OLAY - BRINGS IN A PROGRAM OVERLAY
- M\$LDTRC - TRANSFERS CONTROL TO ANOTHER PROGRAM
- M\$LINK - TRANSFERS CONTROL TO ANOTHER PROGRAM, SAVING
THE CURRENT PROGRAM STATE FOR LATER RESUMPTION

SAMPLE JOB STEP CYCLE

- NEW USER CREATED and LOGGED ON
- SCHEDULER CALLS CPEXIT TO ASSOCIATE COMMAND PROCESSOR
- COMMAND PROCESSOR ENTERED
- CP is REQUESTED TO RUN A PROGRAM
- CP ISSUES M\$CPEXIT TO INITIATE PROGRAM
- EXECUTION CONTROL FETCHES PROGRAM INTO WSQ, CALLS SCHEDULER
- PROGRAM EXECUTES and EVENTUALLY EXITS
- CP IS REENTERED
- CP IS REQUESTED TO RUN ANOTHER PROGRAM
- CP TERMINATES CURRENT JOB STEP VIA M\$RUND, INITIATES ANOTHER VIA M\$CPEXIT.

PROGRAM TO SYSTEM INTERFACE

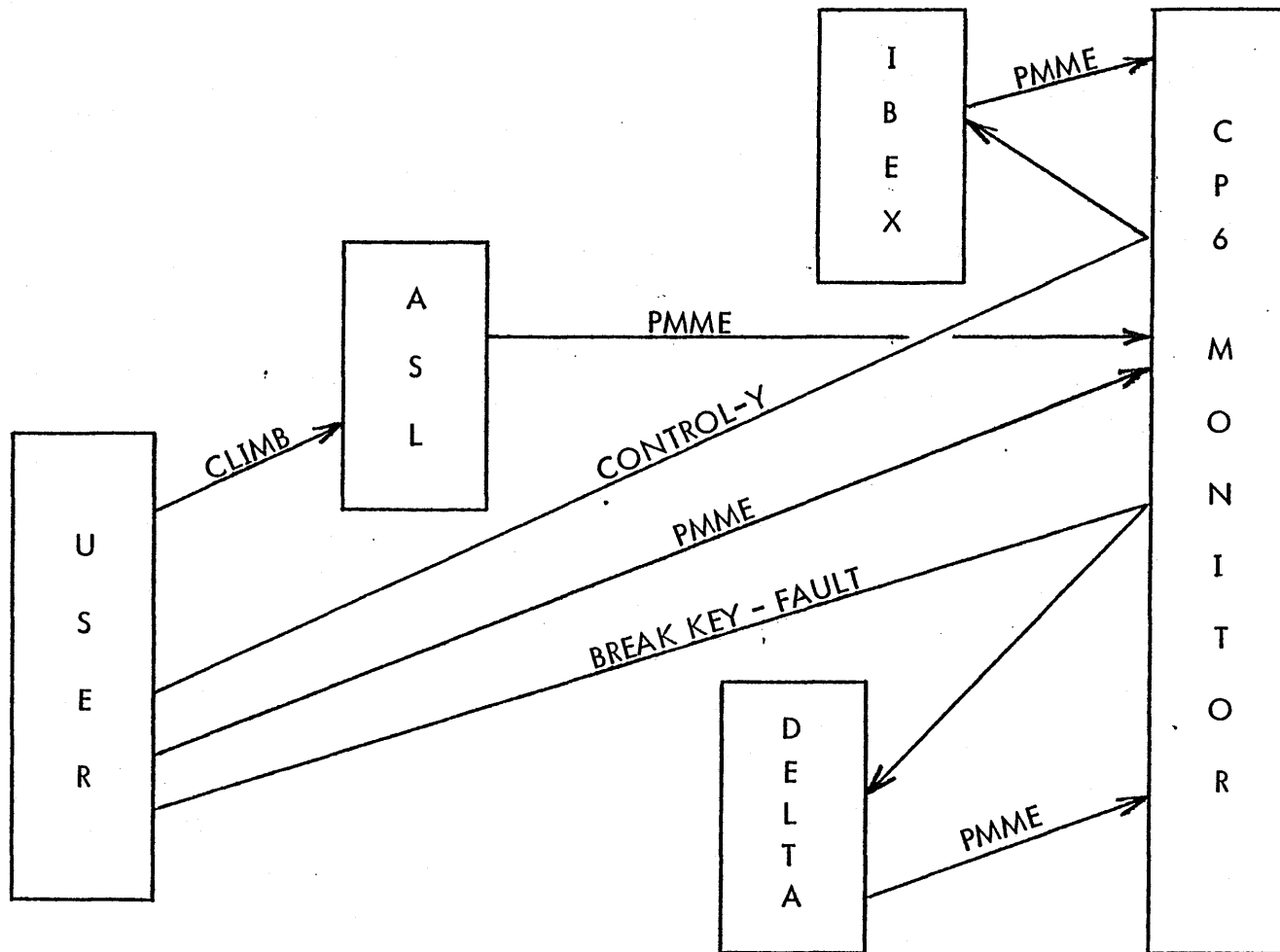
- CONTROL PATHS BETWEEN CP-6 DOMAINS
- MONITOR SERVICES INTERFACE
- SCHEDULING (DISPATCHING)
- FAULT and EXCEPTIONAL CONDITION HANDLING

INTER-DOMAIN INTERFACE

- DOMAINS HAVE ESTABLISHED LEVEL OF PRIORITY
 - USER
 - ALTERNATE SHARED LIBRARY
 - DEBUGGER
 - COMMAND PROCESSOR
 - MONITOR

- INWARD CLIMB TO DOMAIN OF HIGHER PRIORITY
 - SAVES ENVIRONMENT
 - OPTIONALLY PREPARES NEW PARAMETERS
 - ESTABLISHES NEW ENVIRONMENT

- OUTWARD CLIMB TO RETURN
 - RESTORES ENVIRONMENT
 - RELINQUISHES CONTROL



CONTROL PATHS BETWEEN CP-6 DOMAINS

CP-6 MONITOR SERVICES INTERFACE

- SERVICES AVAILABLE IN ALL LANGUAGES
- SERVICES AVAILABLE TO ALL DOMAINS
- OPERATION IS IDENTICAL FOR BATCH and ON-LINE
- BROAD SPECTRUM of SERVICES AVAILABLE

CATAGORIES OF CP-6 MONITOR SERVICES

•	JOB STEP CONTROL	15
•	RUN-TIME SERVICES	25
•	MEMORY MANAGEMENT	15
•	FILE MANAGEMENT/DEVICE I/O	25
•	ON-LINE TERMINAL CONTROL	10
•	DIAGNOSTIC SERVICES	10
•	SPECIAL SHARED PROCESSOR SERVICES	5

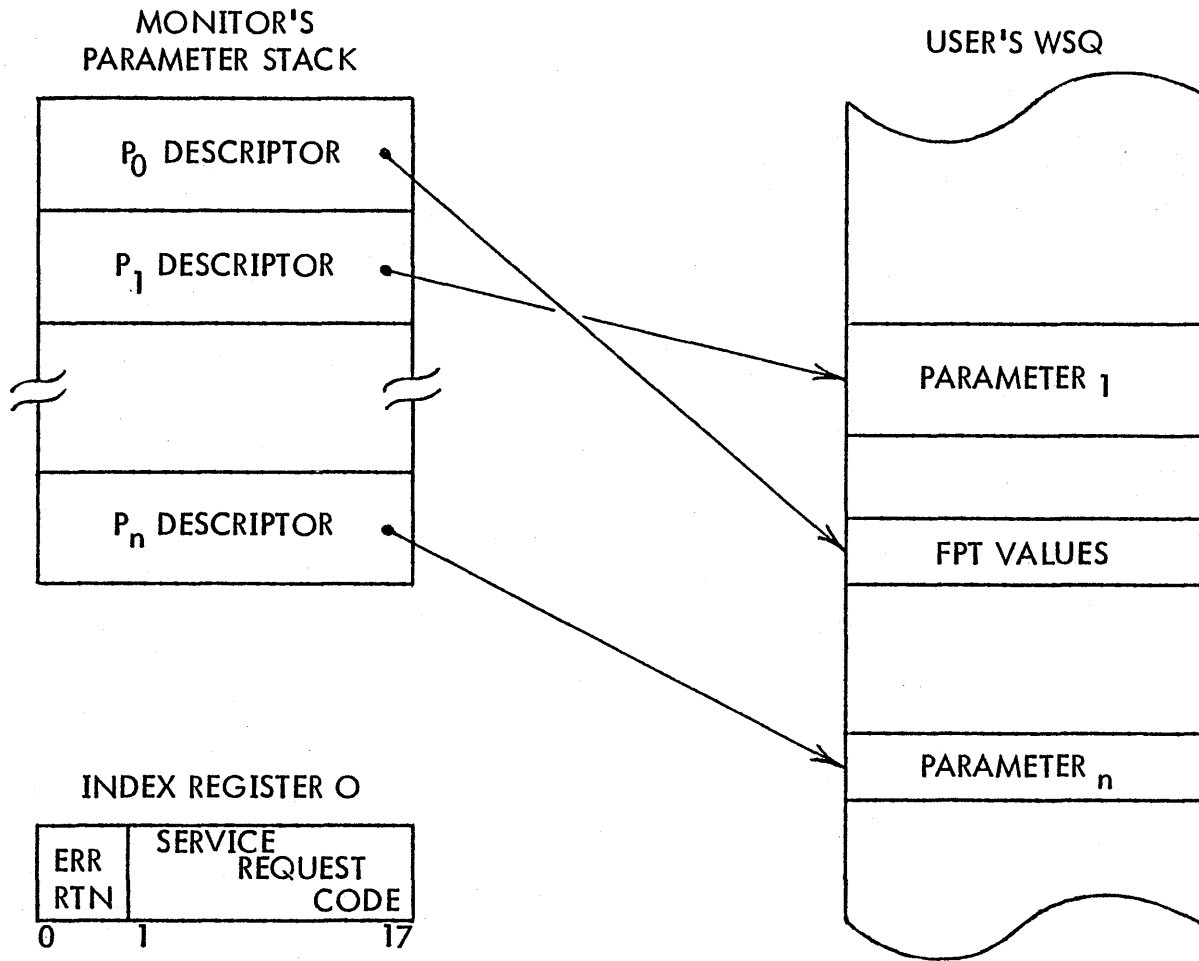
CP-6 MONITOR SERVICES REQUEST

- INVOKED VIA PMME FORM of CLIMB
 - SAVE ENVIRONMENT
 - VALIDATES PARAMETER ADDRESSES/SIZE
 - ESTABLISHES MONITOR ENVIRONMENT

- INPUTS ASSOCIATED with EACH REQUEST
 - SERVICE REQUEST CODE
 - ERROR RETURN SPECIFICATION
 - WHERE REQUIRED - USER PARAMETERS

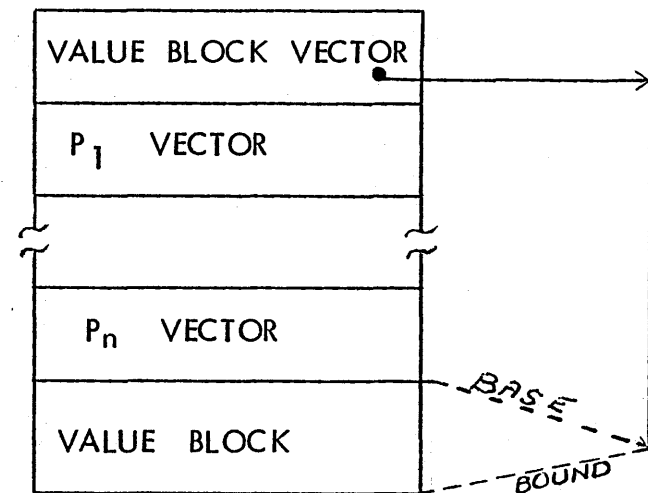
- OUTPUTS SUPPLIED for ERROR ROUTINES
 - COPY of SAFE STORE FRAME
 - SERVICE REQUEST CODE
 - DCB# (IF APPLICABLE)
 - ERROR CODE

MONITOR SIDE OF PMME



FUNCTIONAL PARAMETER TABLE

- FIXED FORMAT for EACH REQUEST
- SUPPLIES MONITOR with USER SPECIFIC INFORMATION
 - AREAS in USER's MEMORY
 - PRESENCE/ABSENCE of OPTIONAL PARAMETERS
 - VALUES to OVER-RIDE DEFAULTS
- AREAS IN USER MEMORY SPECIFIED AS VECTORS
- ALL VALUES PASSED AS ONE PARAMETER



LDPO FPT

CLIMB EA - BIT 0 = ERR RETURN SPECIFICATION
 BITS 0-17 = SERVICE REQUEST CODE
 SEGID = PMME
 C = INWARD CLIMB
 LOAD XO WITH EA
 E = 1 - PREPARE PARAMETERS (OR ZERO)
 P = # of PARAMETERS - 1

TRA ERROR (if EA BIT 0 = 1)

USER SIDE OF PMME

PL-6 EXAMPLE

```
DCL INBUF CHAR (80) STATIC;
```

```
% INCLUDE CP-6;
```

```
% FPT - READ (FPTN = SIFPT,
```

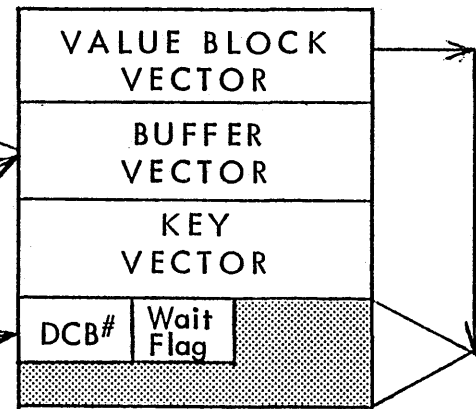
```
    BUF = INBUF;
```

```
    DCB = M$SI);
```

```
CALL M$READ (SIFPT) ALTRET (READERR);
```

```
  .  
  .  
  .
```

```
READERR:      /*ERROR HANDLER*/
```



CP-6 SCHEDULER

- STATE QUEUE MAINTENANCE
- QUANTUM TIMING and CONTROL
- EXECUTION SCHEDULING
- ASYNCHRONOUS EVENT PROCESSING

STATE QUEUES

- EXECUTING
- EXECUTABLE
 - PRIORITY 1
 - PRIORITY 2
 - PRIORITY 3
 - .
 - .
 - .
 - PRIORITY n
- NON-EXECUTABLE
 - I/O WAIT
 - SLEEPING
 - QUEUED FOR RESOURCE
 - TERMINAL INPUTTING
 - TERMINAL OUTPUTTING

STATE QUEUE MAINTENANCE

- EVERY JOB in SYSTEM HAS ASSIGNED PRIORITY
- EVERY JOB HAS SINGLE ENTRY in STATE QUEUES
- EXECUTABLE STATE QUEUES HAVE a PRIORITY
- 'EVENTS' CAUSE CHANGE in STATE
 - BLOCKING EVENT MOVES USER to NON-EXECUTABLE STATE
 - UNBLOCKING EVENT MOVES USER TO EXECUTABLE STATE BASED ON PRIORITY ASSOCIATED WITH THE EVENT.
 - EXECUTION SCHEDULER MOVES HIGHEST PRIORITY EXECUTABLE USER TO EXECUTING STATE

EVENTS

I/O IN PROGRESS
I/O COMPLETE

SLEEP
WAKE - UP

NEED MEMORY PAGE
PAGE AVAILABLE

NO DISK SPACE
DISC SPACE AVAILABLE

COMMUNICATIONS INPUT IN PROCESS
COMMUNICATIONS INPUT COMPLETE

BREAK CHARACTER RECEIVED
CONTROL Y RECEIVED
USER ABORTED BY OPERATOR
I/O COMPLETE with EVENT INFO

QUANTUM TIMING

- SYSTEM PARAMETERS

QUAN - MAXIMUM TIME ALLOWED COMPUTE - BOUND USER

- USED TO CAUSE TIMER RUNOUT FAULT at QUANTUM END

QMIN - AMOUNT OF TIME GUARANTEED TO USER

- USER MAY BE FORCED to RELINQUISH CONTROL to
HIGHER PRIORITY USER ONCE QMIN IS REACHED

EXECUTION SCHEDULER

- ENTERED FOLLOWING CLIMB FOR: FAULT
INTERRUPT
PMME
- DETERMINE IF JOB ALLOWED TO CONTINUE
 - SUSPEND USER WITH ENVIRONMENT
SAVED in SAFE-STORE
 - RELOAD SSR FOR NEW USER
- PROCESS ASYNCHRONOUS EVENTS
 - ENTER COMMAND PROCESSOR
 - ENTER DEBUGGER
 - ENTER USER ASYNCHRONOUS EVENT HANDLER
- RET THROUGH FRAME LAID DOWN BY: FAULT
INTERRUPT
PMME

CP-6 FAULT PROCESSING

ACTION TAKEN DEPENDS ON:

- TYPE OF FAULT
 - ARITHMETIC
 - PROGRAMMED
 - PROCEDURE ERRORS
 - SYSTEM
- DOMAIN IN CONTROL
- REQUEST FOR CONTROL

CLASSES OF FAULTS

ARITHMETIC

OVERFLOW
DIVIDE CHECK

PROGRAMMED

MASTER MODE ENTRY
DERAIL
FAULT TAG

PROCEDURE ERRORS

MEMORY
COMMAND
LOCKUP
ILLEGAL PROCEDURE
MISSING SEGMENT
MISSING PAGE
SECURITY 2
SAFE STORE
SECURITY 1

SYSTEM

TIMER, RUN-OUT
PARITY
OP NOT COMPLETE
CONNECT
DYNAMIC LINK
MISSING WORKING SPACE
EXECUTE
START UP
SHUT DOWN

SYSTEM FAULTS

TIME RUN-OUT	-	CALL SCHEDULE
PARITY		
OP NOT COMPLETE	-	} CALL T&D
CONNECT	-	PERFORM COMMANDS IN MAILBOX
DYNAMIC LINK		
MISSING WORKING SPACE		
EXECUTE		
STARTUP		
SHUTDOWN		} CALL RECOVERY
 <u>ALL OTHERS</u>		
USER IN CONTROL	-	ABORT OR ENTER USER TRAP HANDLER
MONITOR IN CONTROL-		CALL RECOVERY

EXCEPTIONAL CONDITION PROCESSING

- TYPES OF EXCEPTIONAL CONDITIONS
- ESTABLISHING CONTROL
- EXCEPTIONAL CONDITION ENVIRONMENT
- ENTERING EXCEPTIONAL CONDITION ROUTINES
- EXIT FROM EXCEPTIONAL CONDITION ROUTINES

TYPES OF EXCEPTIONAL CONDITIONS

- FAULTS
- PMME ERRORS
- OPERATOR ABORTS
- LIMITS EXCEEDED
- PROGRAM EXITS AND ABORTS
- TIMER RUN-OUT
- EVENT COMPLETION
- BREAK KEY INTERRUPT

ESTABLISHING CONTROL

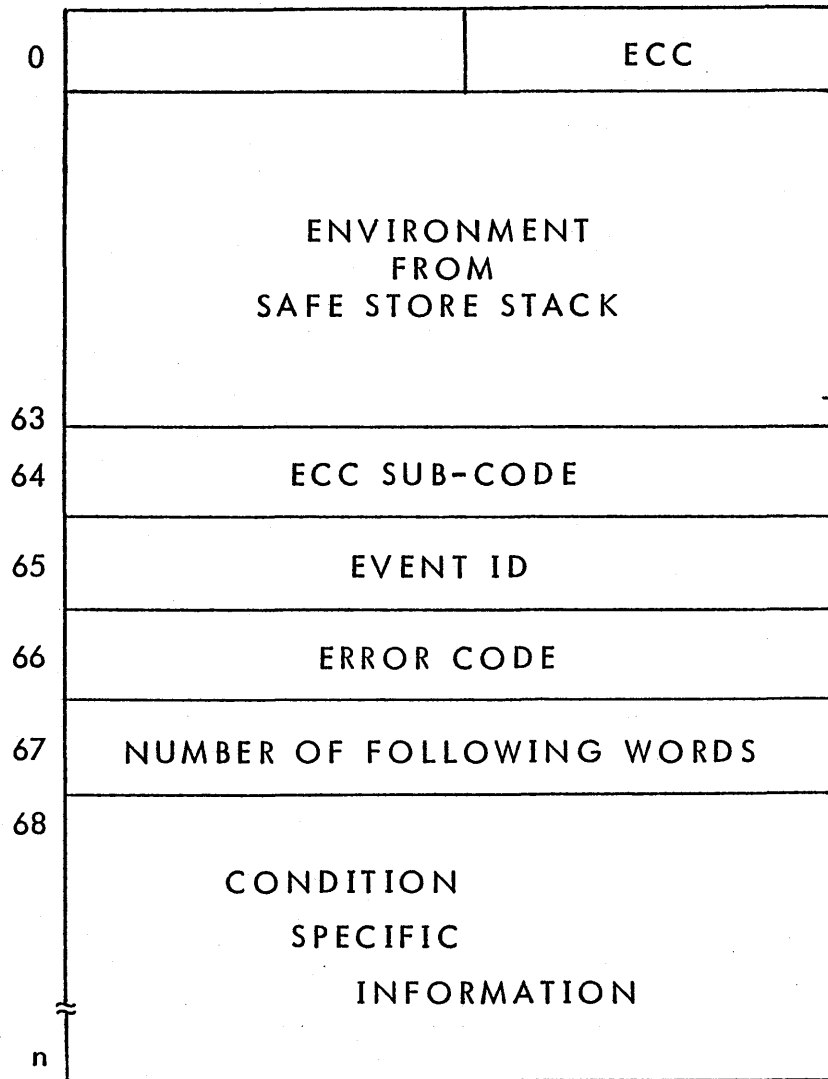
- PROGRAM MAY REQUEST CONTROL FOR ANY OR ALL
 - M\$TIMER TIMER RUN-OUT
 - M\$INT CONSOLE INTERRUPT
 - M\$EVENT COMPLETION OF SPECIFIED EVENT
 - M\$TRAP MACHINE TRAPS - 3 CLASSES
 PMME ERRORS/NO ERR RETURN SPECIFIED
 - M\$XCON NORMAL OR ABNORMAL EXIT OF CURRENT PROGRAM
 - MONITOR SERVICES ERROR RETURN SPECIFICATION

- SEPERATE LEVEL of CONTROL for EACH DOMAIN

EXCEPTIONAL CONDITION CONTROL BLOCK (ECCB)

- CONTAINS USER SPECIFIED HANDLER ADDRESSES
- SEPARATE ECCB FOR EACH DOMAIN

STIMER	EVENT
BREAK	XCON
PMME	ARITHMETIC
PROGRAMMED	ERROR
CONTROL FLAGS	



ECC:

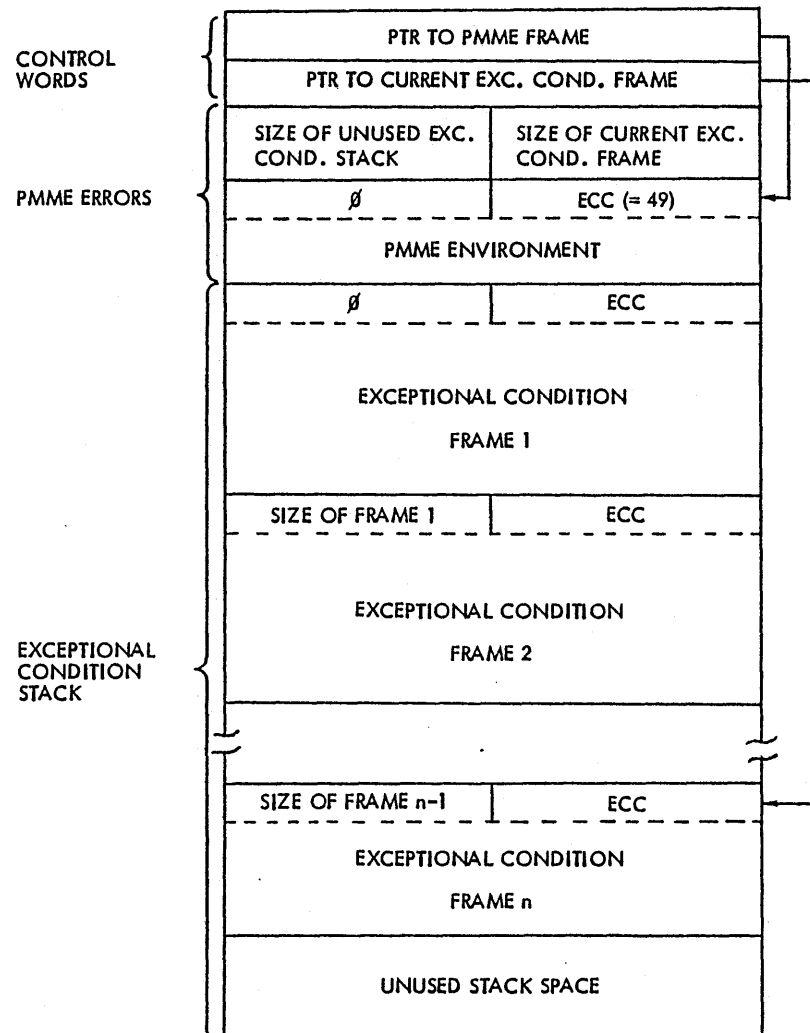
- 0 - TIMER RUNOUT
- 1 - EVENT COMPLETION
- 2 - BREAK
- 3 - XCON
- 4 - BAD PMME/NO ERRΔRTN
- 5 - ARITHMETIC FAULT
- 6 - PROGRAMMED FAULT
- 7 - PROCEDURE FAULT

99 - PMME ERROR RETURN

EXCEPTIONAL CONDITION ENVIRONMENT

TASK CONTROL BLOCK

- STORAGE FOR RELEVANT INFORMATION
- SINGLE FRAME FOR PMME ERROR RETURNS
- STACKED FRAMES FOR ALL OTHER CONDITIONS
- SEPARATE TCB FOR EACH DOMAIN



ENTERING EXCEPTIONAL CONDITION HANDLERS

- SYNCHRONOUS EVENTS

DOES NOT CAUSE CHANGE OF DOMAIN

SAFE - STORE COPIED TO TCB

IC IN SAFE-STORE SET TO CONDITION HANDLER

RET CAUSES HANDLER TO BE ENTERED

- ASYNCHRONOUS EVENTS

MAY CAUSE CHANGE OF DOMAIN

— CONDITION FOR EXECUTING DOMAIN

SAME AS FOR SYNCHRONOUS EVENTS

— CONDITION FOR DOMAIN WITH HIGHER PRIORITY

LTRAD TO CONDITION HANDLER

— CONDITION FOR DOMAIN OF LOWER PRIORITY

DEFER PROCESSING

EXITING EXCEPTIONAL CONDITION ROUTINE

- MONITOR SERVICE ERROR RETURNS
 - NO ACTION is REQUIRED
 - M\$MERC MONITOR ERROR HANDLING
 - M\$RETRY MODIFY ENVIRONMENT and RETRY PMME

- STACK CONDITION ROUTINES
 - M\$TRTN MODIFY ENVIRONMENT and RETURN
 - M\$CLRSTK POP STACK FRAME and CONTINUE

CONNECTING USERS TO DEVICES AND FILES

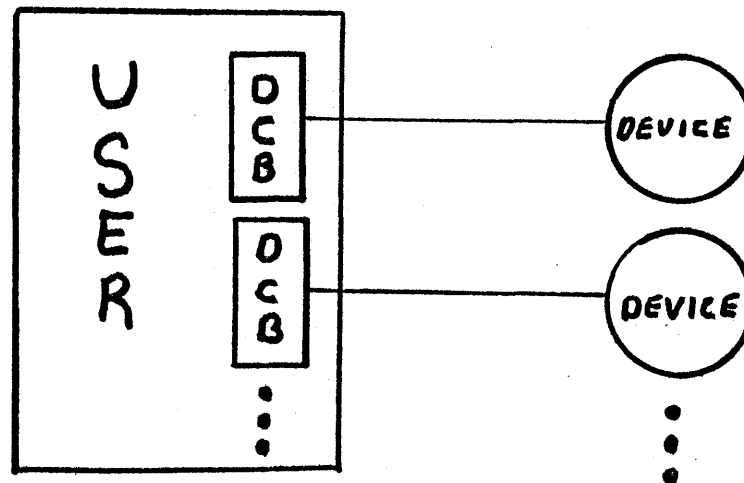
- TO CENTRAL SITE PERIPHERALS
- TO DISK FILES
- TO PVT. VOL. FILES
 - BOTH DEVICE AND FILE
- TO COMMUNICATIONS DEVICES
 - THESE MUST ALSO BE CONNECTED THEMSELVES

9/19/77

SK

THE DATA CONTROL BLOCK (DCB)

- READ ONLY USER CONTEXT
- EACH USER HAS HIS OWN
- CONNECTION POINT FOR USERS TO DEVICE
- BASIC OPERATIONS: OPEN, READ, WRITE, CONTROL, CLOSE
- OPEN IS THE CONNECTING PROCESS



TYPES OF ACCESS

- PUBLIC FILE
- PRIVATE DISK FILE
- MANAGED TAPE FILE
- VIRTUAL DEVICE
- WSN/DEVICE
- MASTER T.S. TERMINAL
- COMGROUP

OPEN

- BASIC ELEMENTS SPECIFIED
- ASN: FILE, MANAGED TAPE, DEVICE, COMGROUP, ME
- WSN: SUBSET OF DEVICES
- RN: TYPE OF DEVICE/VIRTUAL DEVICE/OPLABEL
- SER: TAPE OR PRIVATE DISK VOLUME
- N.A.P: IDENTIFIES FILE
- SOME OR ALL MAY BE SPECIFIED ON DCB CREATION
OR ISET

STANDARD FILE IDENTIFIER (FID)

RN#SER/N.A.P. @WSN

- ALL FIELDS OPTIONAL
- # SAYS DEVICE
- @ SAYS DEVICE
- / SAYS FILE
- . SAYS FILE
- RESERVED RN'S : MT, LT, DP,
DC, CG, OPLABELS

FID EXAMPLES

PUBLIC FILE : A.B.C
PRIVATE DISK FILE : DP#1234/A.B.C
MANAGED TAPE FILE LT#1234/A.B.C
VIRTUAL DEVICE : MTO1#123
WSN/DEV : LP@BOSTON
OPLABEL : LP
COMGROUP : CG#OSNET
MASTER TERMINAL : ME

HISTORICAL PROBLEM :
IS 'ZZ' DEVICE OR FILE?

9/19/77

SK

FUNCTIONS BETWEEN USER AND DEVICES

- OPLABELS
- WORKSTATIONS
- SYMBIONTS (SPOOLING)
- STREAMS
- RESOURCE MANAGEMENT
- FILE MANAGEMENT
 - CONVERTS FILE COMMANDS TO SENSIBLE I/O TO REAL DEVICE

OPLABELS

- BATCH, ONLINE, GHOST

<u>OPLABEL</u>	<u>BATCH</u>	<u>ONLINE</u>	<u>GHOST</u>
ME	LP01,CR01	UC	OC
SI	CR01	UC	OC
LO	LP01	UC	OC
LP	LP01	LP01	LP01

●

●

- ALLOW STANDARD ASSIGNMENT TO "NATURAL" DEVICE FOR MODE
- HEAVILY USED FOR LP, CP, UC IN CP-V AND CP-6 BY PROCESSORS

9/19/77

SK

WORKSTATIONS

- EVERY USER HAS WORKSTATION OF ORIGIN
- WSN IS SUBGROUP OF ALL DEVICES
- DEVICE TYPE TRANSLATED THROUGH WSN DEFINITION TO PRODUCE REAL DEVICE AND "SYMBIONTNESS"

WSN DEFINITION

LP: SYMBIONT , OTHER ATTRIBUTES
LP01 ON TERMINAL BOSTON
LP02 ON TERMINAL BOSTON

QQ: NON-SYMBIONT, OTHER ATTRIBUTES
ZZ01 ON TERMINAL BIGBOSS

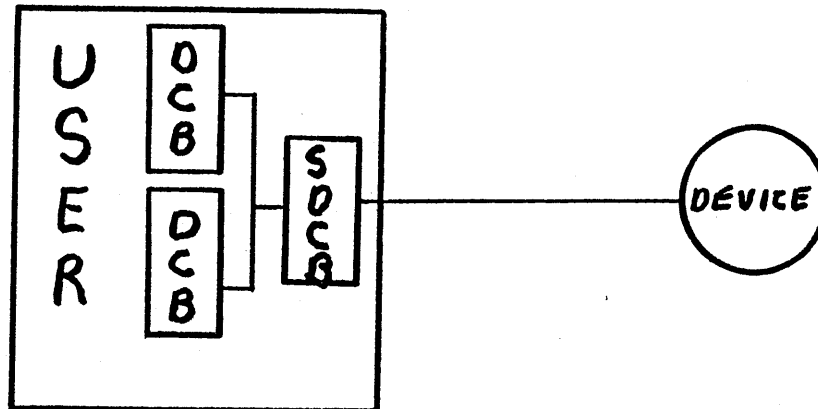
MT: USE CENTRAL SITE DEVICES

9/19/77 SK

SYMBIONTS

- PROVIDES FILE BUFFERING FOR SLOW U.R. DEVICES
- MANY USERS USING FEW DEVICES
- TALK TO COMMUNICATION DEVICES NOT CONNECTED
- SENSIBLE OUTPUT SCHEDULING
 - FORMS
 - SIZE
 - PRIORITY
- BATCH QUEUE
- NON-CONTROL INPUT FILES

STREAMS



- ALLOW SEVERAL DCBS TO MIX OUTPUT
 - DIFFERENT OR SAME M\$LO, M\$DO
- STREAM IS VIRTUAL DEVICE NAMED VIA LDEV
- ! LDEV LP02, LPC, ATTRIBUTES
- STREAM DCB IS ASSIGNED TO WSN/DEV
- LP01 IS ALWAYS AUTOMATICALLY THERE - POINTS TO LP@ WSN OF ORIGIN
- ALWAYS SYMB IONT IN 1ST RLSE

RESOURCE MANAGEMENT

- RESOURCE IS VIRTUAL DEVICE NAMED VIA LIMIT
| LIMIT MT01, MT, OTHER ATTRIBUTES
- ACTUAL DEVICE IS IN WSN/DEV FORM
- ALWAYS NON-SYMBIONT
- LIMIT PREVENTS CONTINUING UNTIL DESCRIBED
REAL DEVICE IS ACQUIRED
- EXCEPTIONS:
 - ONLINE | LIMIT
 - ONLINE TAPES: MT01, MT02

OPENING TO PUBLIC FILE

- ASN = FILE
- RN = DC
- NAME (.ACCOUNT. PASSWORD)
- SER: USUALLY NOT PRESENT

GOES DIRECTLY TO FILE MANAGEMENT

9/19/77

SK

OPENING TO PRIVATE VOLUME FILE

- ASN = FILE (DISK) OR MANAGED TAPE (TAPE)
 - RN = DPXX(DISK), MTXX, LTXX (TAPE)
 - IF 'XX' = '00 - '99' RN IS VIRTUAL DEVICE
 - IF 'XX' = ' ' ' RESOURCE MANAGEMENT PICKS VIRTUAL DEVICE
 - SER: USUALLY PRESENT
 - NAME(.ACCOUNT. PASSWORD)
- GOES TO RESOURCE MANAGEMENT TO GET DEVICE THEN TO FILE MANAGEMENT

9/19/77 SK

OPENING TO VIRTUAL DEVICE

ASN = DEVICE

RN = VIRTUAL DEVICE (MT_{xx}, LP02)

SER = IS USED FOR TAPES

- THE OPEN IS TO A STREAM IF ONE EXISTS BY THE RN NAME - OTHERWISE TO RESOURCE VIA RESOURCE MANAGEMENT

9/19/77

OPENING TO WSN/DEVICE

ASN = DEVICE

RN = OPLABEL or DEVICE TYPE ('LP', 'QQ')

SER: IS NOT USED

WSN: MAY BE PRESENT

- IF WSN IS NULL, WORK STATION OF ORIGIN IS APPLIED
- IF OPLABEL, TRANSLATION PRODUCES STREAM
- ASSIGNMENT IS TO SYMBIONT FILE OR RESOURCE
BASED ON WSN DEFINITION
- REMEMBER VIRTUAL DEVICES USE THIS METHOD
- SPECIAL DEVICE TYPES: JE, JF

OPENING TO MASTER T.S. TERMINAL

- SPECIAL UC STREAM POINTS TO TERMINAL

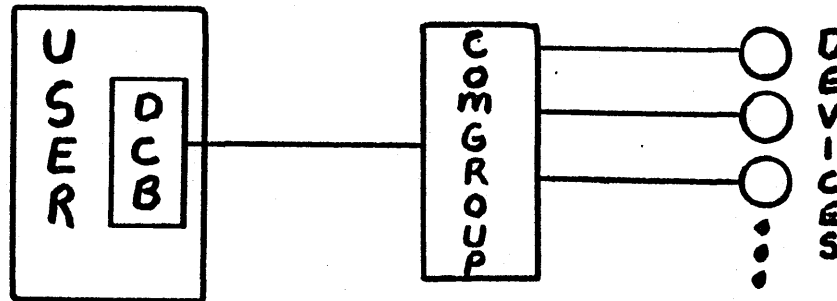
ASN = DEVICE

RN = 'ME' OR OPLABEL POINTING TO
UC STREAM

9/19/77 SK

OPENING TO COMGROUP

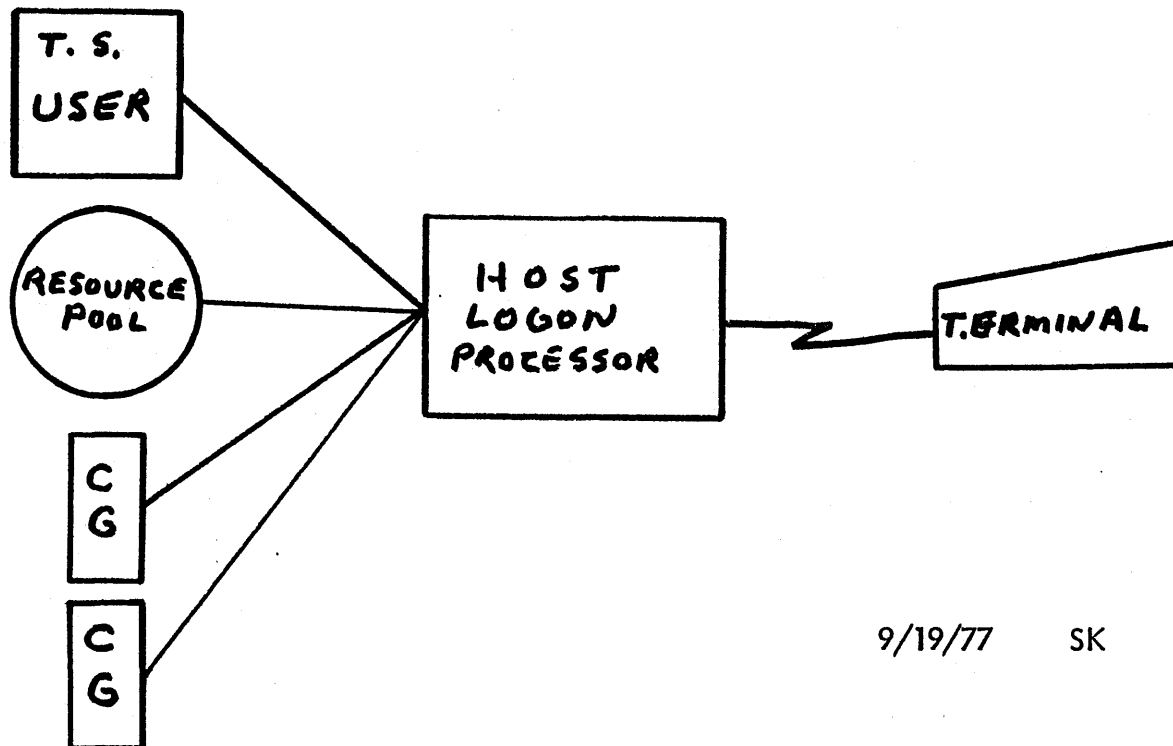
- ASN = COMGROUP
- SER = COMGROUP NAME



- WAY TO ASSIGN ONE DCB TO MANY DEVICES
- USUALLY COMMUNICATIONS DEVICES
- NEW CONCEPT IN CP-6
 - USED ONLY BY MONITOR IN 1ST RLSE
 - WILL BE USED EVENTUALLY FOR T.P.
- SYMBIONTS and KEYIN USE THEM
- ALLOW RAT, RNT, WLS, WNS

CONNECTING COMMUNICATIONS DEVICES

- ALL CONNECT AND LOGON TO HOST LOGON PROCESSOR
- HLP CONNECTS THEM TO
 - 1) NEW T.S. USER (T.S. TERMS)
 - 2) RESOURCE (ODD PERIPHERALS)
 - 3) COMGROUP (LP, CP, CR, OC, TP)



9/19/77

SK

CONNECTING USERS TO DEVICES AND FILES

- PUBLIC FILE
- PRIVATE DISK FILE
- MANAGED TAPE FILE
- VIRTUAL DEVICE
- WSN/DEVICE
- MASTER T.S. TERMINAL
- COMGROUP

9/19/77 SK

FILE MANAGEMENT DEFINITIONS

- GRANULE UNIT OF ALLOCATION OF DISK STORAGE
= 1024 WORDS = 1 MEMORY PAGE
- ACCOUNT 1) LOGICAL GROUPING OF FILES, 2) IDENTIFICATION
OF USER FOR LOGON AND FILE ACCESS
- ACCOUNT DIRECTORY CATALOG OF ACCOUNTS WITHIN THE SYSTEM OR ON
A PACK SET
- FILE DIRECTORY CATALOG OF FILES WITHIN AN ACCOUNT
- FIT FILE INFORMATION TABLE
- PACK SET LOGICAL GROUP OF DISK PACKS MAY CONSIST OF
ONE OR MORE PACKS. MAY CONTAIN FILES FOR
ONE OR MORE ACCOUNTS

FILE ORGANIZATIONS

● SIX DISK FILE ORGANIZATIONS

- KEYED – EACH RECORD IDENTIFIED BY UNIQUE NAME (KEY)
RECORDS STORED IN SORTED ORDER
VARIABLE LENGTH RECORDS AND KEYS
KEY MAXIMUM LENGTH = 31 CHARACTERS

- CONSECUTIVE – NO UNIQUE RECORD IDENTIFIERS
RECORDS WRITTEN AND READ IN SEQUENTIAL ORDER
VARIABLE LENGTH RECORDS

- RANDOM – NO KNOWN STRUCTURE

- RELATIVE – NO UNIQUE RECORD IDENTIFIERS OTHER THAN RECORD NUMBER
FIXED LENGTH RECORDS

- INDEXED – RECORD IDENTIFIER IS WITHIN DATA
~~VARIABLE~~ ~~FIXED~~ ~~LENGTH~~ ~~RECORD~~ ~~AND~~ ~~IDENTIFIER~~ ~~(KEY)~~
KEY LENGTH MAXIMUM = 255 CHARACTERS
FIXEDS

- INTEGRATED – RANDOM FILE STRUCTURED BY IDS

● EIGHT ANS TAPE FORMATS

- ANS F – FIXED LENGTH RECORDS
- D – VARIABLE LENGTH WITH DECIMAL CONTROLS
- V – VARIABLE LENGTH WITH ASCII CONTROLS
- U – UNDEFINED

- CP-6 K – KEYED (INDEXED)
- C – CONSECUTIVE (RELATIVE)
- R – RANDOM
- B – BLOCKED

FILE MANAGEMENT SERVICES

- NORMAL SERVICES – OPEN, CLOSE, READ, WRITE, POSITION, DELETE RECORD, DELETE FILE
AUTOMATIC: BLOCK, DEBLOCK, GRANULE ALLOCATION

- NOT-SO-NORMAL – TEXT DATA COMPRESSION
WRITE-MORE/READ-LESS (SCATTER I/O)
SHARE (INPUT, NONE, ALL)
REWRITE (KEYED, CONSECUTIVE)
READ/WRITE SEQ OR RANDOM (KEYED)
AUTOMATIC FILE EXTENSION OR CONCATENATION
DCB CORRESPONDENCE

- FEATURES – STAR FILES – UNIQUE TEMP FILES
JOURNAL MODE – (OUTPUT CONSEC)

FILE SECURITY

- GRANULE ACCESS : STAMP
- ACCOUNT ACCESS : READ or CREATE
- FILE ACCESS : MULTIPLE TYPES
- DATA ACCESS : PASSWORD and ENCRYPTION

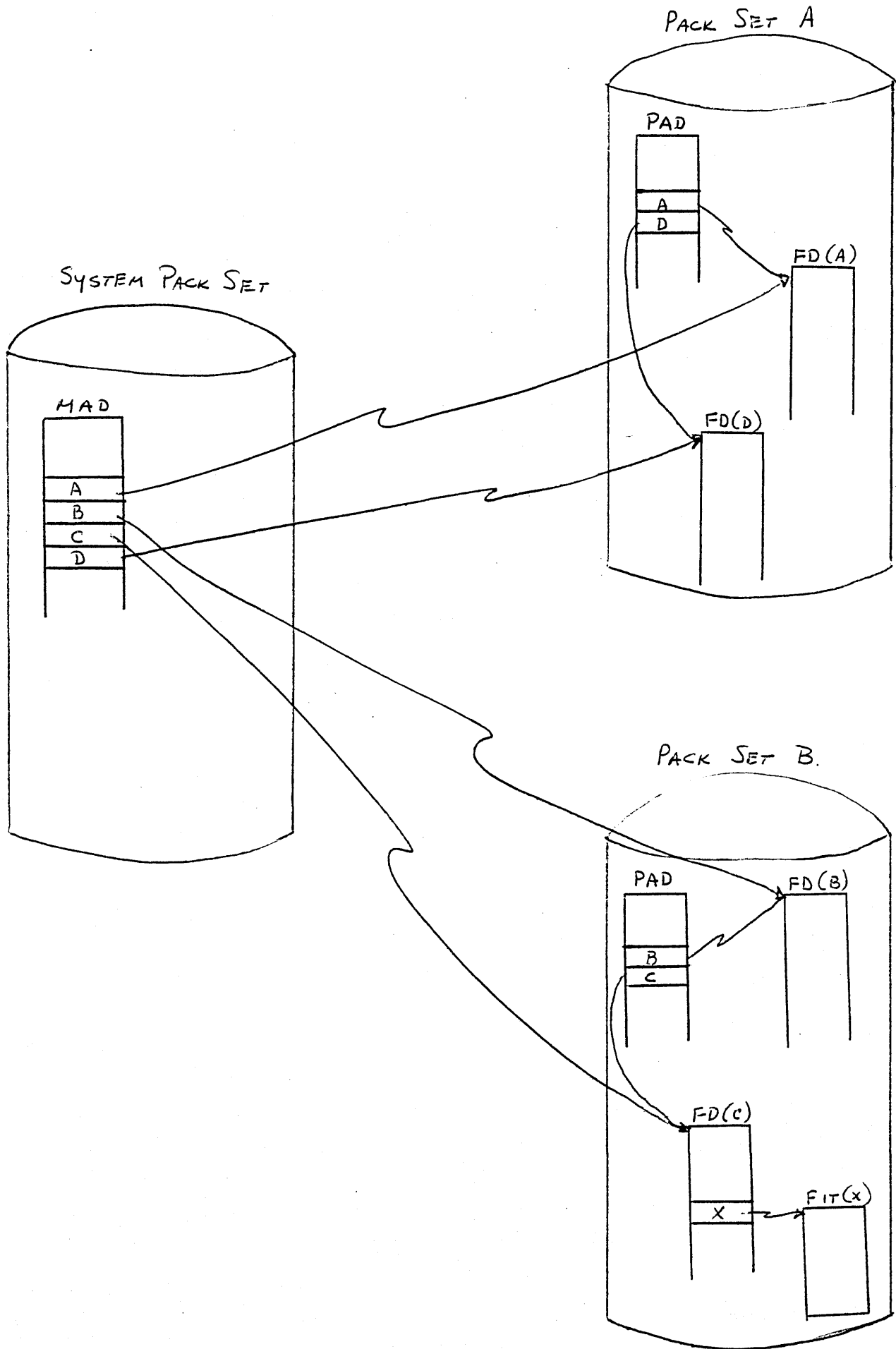
FILE ACCESS TYPES :
READ
UPDATE
WRITE NEW
DELETE RECORDS
DELETE FILE
LIST
FILE INFORMATION
ACCESS VEHICLE

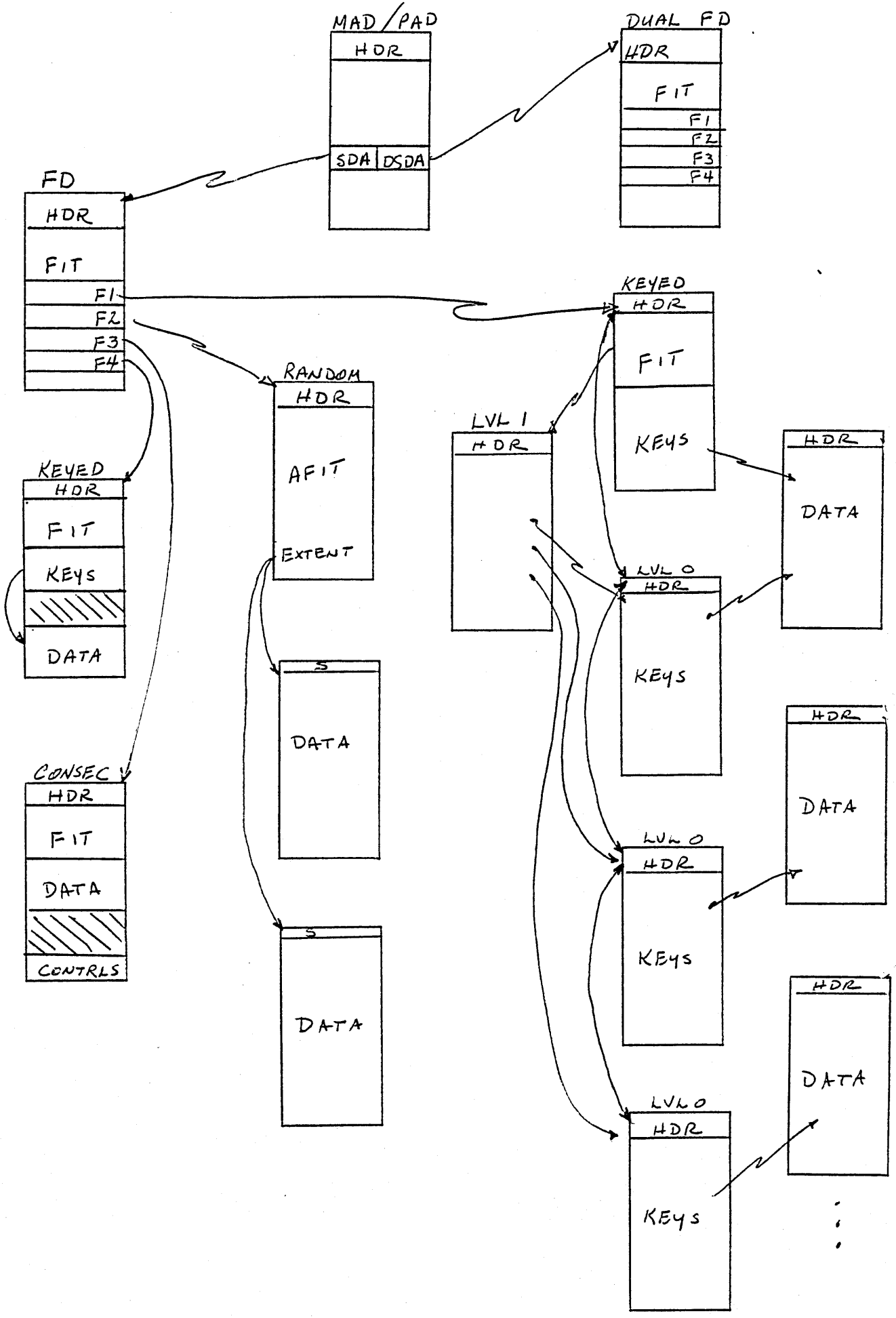
FILE MANAGEMENT PROCESSORS

- INITVOL INITIALIZE PACK SETS
- CAT CATALOG PACK SETS and THEIR
BACKUP MEDIA
- GAC GRANULE ACCOUNTING
- LABEL INITIALIZE ANS TAPES
- EFT BACKUP/RESTORE and STOW/FETCH
- HGPRECON GRANULE POOL RECONSTRUCTION

INTERNAL FEATURES

- CFU RETENTION : ACTIVE FILES DO NOT REQUIRE DIRECTORY SEARCH for M\$OPEN
- READ-AHEAD, WRITE-AHEAD : NO DELAY for PHYSICAL I/O
- I/O CACHE (STICKY PAGES) : SIGNIFICANT GRANULES REMAIN IN MEMORY
- RECOVERY and JOB STEP RUNDOWN : DEFAULT BUFFER TRUNCATION and FILE CLOSE
- GRANULE STAMP : NO SCRUB
- COMPRESSED KEYS : SPACE and ACCESS SAVING
- IDS : MODIFICATION DATE per GRANULE
- EFT : BLOCK ACCESS and FILE DESCRIPTORS





USER'S VIEW OF CP-6

- USER AUTHORIZATION
- COMMON COMMAND LANGUAGE
- IBEX (INTERACTIVE and BATCH EXECUTIVE)
- CP-6 PROCESSORS
- USER SERVICES

USER AUTHORIZATION

- CENTRALIZED CONTROL
- INDIVIDUAL AUTHORIZATION
- SYSTEM DEFAULTS and LIMITS
- BUDGET CHECK
- PRE-SCANNED BATCH JOBS

COMMON COMMAND LANGUAGE

- SIMPLE and COMMON SYNTAX RULES
- STANDARD CALLING SEQUENCE
- UNIVERSAL FILE IDENTIFIER (FID)

STANDARD CALLING SEQUENCE

- TRANSFER A OVER B, LP; LS, GO, SI
- IMPLICIT DCB ASSIGNMENT
- OPTIONS FOLLOW THE SEMICOLON

IMPLICIT DCB ASSIGNMENT

" ANSFORT A OVER B, LP; LS, GO, SI"

- LINK RELATES POSITION TO ACTUAL DCB
- IBEX RELATES POSITION TO FID
- STEP CONNECTS FID TO ACTUAL DCB
- IBEX PROVIDES POINTER TO OPTIONS

IBEX

- SINGLE REPLACEMENT FOR TEL/CCI
- EXECUTION CONTROL
- COMMAND FILES ("CATALOGUED PROCEDURES")
- FILE MANAGEMENT INTERFACE
- COMMUNICATIONS INTERFACE
- ACCOUNTING

EXECUTION CONTROL

- STEP CONDITION CODE
- COMMAND VARIABLES
- STEP LIMITS and RESOURCE RETURN
- INTERRUPT and CONTINUE
- BUDGET OVERRUN ABORT

EXAMPLE

⋮
!PROGRAMA.
!STEP IF STEPCC \approx 0, LET A = 2
!PROGRAMB.
!STEP IF A \approx 0, GOTO ABC
!PROGRAMC.
⋮
!STEP(ABC)
!LIMIT(STEP) LO = 99, TIME = 1
!PROGRAMI.
!STEP IF STEPCC \approx 0, GOTO STEP
⋮

COMMAND FILES

- AVAILABLE TO ALL MODES
- NESTING ALLOWED
- STRING AND FIELD SUBSTITUTION

FILE MANAGEMENT INTERFACE

- ASSOCIATE A DCB WITH A FID (ISET)
- ARCHIVE STORAGE QUALIFICATION (ISTOW)
- ARCHIVE RETRIEVAL REQUEST (IFETCH)
- STREAM DEFINITION and MANIPULATION

ILDEV LP07 LP, FORM = CHECKS, LINES = 9

⋮

IERASE LP07

COMMUNICATIONS INTERFACE

- ACCESS TO NETWORK
- TERMINAL PROFILE
- STATISTICS

ACCOUNTING

- BASIC ACCOUNTING
- BUDGET ACCOUNTING
- JOB-STEP ACCOUNTING
- PROPRIETARY PROCESSOR ACCOUNTING
- FORMS ACCOUNTING

CP-6 PROCESSORS

- DELTA
- UTILITY PROCESSORS
- SYSTEM MANAGEMENT PROCESSORS

DELTA

- SINGLE REPLACEMENT FOR 4 CP-V DEBUGGERS
- EXTERNAL TO THE DEBUGGED PROGRAM
- MULTILINGUAL
- EXECUTIVE VERSION IS MONITOR INDEPENDENT

DELTA

- PROGRAM INTERRUPT on SPECIFIED CONDITIONS
- PROCEDURE and DATA MODIFICATION and DISPLAY
- HISTORY TRACE of EXECUTION
- POST-ABORT ASSOCIATION and DEBUGGING

UTILITY PROCESSORS

- EDIT - GENERAL PURPOSE TEXT EDITOR
- PCL - GENERAL PURPOSE FILE MANIPULATOR
- LINK - LINK LOADER
- LEMUR - LIBRARY EDITOR and MAINTENANCE ROUTINE

SYSTEM MANAGEMENT PROCESSORS

- SUPER - INDIVIDUAL USER AUTHORIZATION
- RATES - CHARGE TABLE MAINTENANCE
- CONTROL- SYSTEM PERFORMANCE CONTROL
- DEF - SYSTEM BOOT TAPE CREATION
- ANLZ - SYSTEM CRASH ANALYZER

USER SERVICES

- HELP COMMAND
- STANDARD ERROR HANDLING
- FID DECODER (M\$FID)
- GENERAL PURPOSE OUTPUT FORMATTER (M\$FORMAT)
- CP-V TO CP-6 JCL CONVERTER
- GENERAL PURPOSE SYNTAX PARSER (M\$PARSE)

M\$PARSE

- SINGLE PARSING ROUTINE FOR ALL UTILITY PROCESSORS
- POWERFUL TABLE DRIVEN STRUCTURE
- EASILY EXTENDABLE

MISCELLANEOUS TOPICS

- SYSTEM INITIALIZATION and CONFIGURATION CONTROL
- SYSTEM RECOVERY
- PERFORMANCE MONITORING and CONTROL
- ERROR LOGGING and LISTING

SYSTEM INITIALIZATION AND CONFIGURATION CONTROL-I

- GOALS

- MINIMIZE CODE SPECIFIC TO SYSTEM BUILD
- SIMPLIFY SYSTEM BUILD PROCESS OVER CP-V
- BOOT-TIME RECONFIGURATION CAPABILITY
- STANDARD SYSTEM, READY TO RUN

- SYSTEM PACKAGING
 - THE MINIMUM SYSTEM
 - DEFAULT CONTROL PARAMETERS

- RELOAD SYSTEM WHEN ADDITIONAL FEATURES REQUIRED

- USES STANDARD PROCESSORS

SYSTEM INITIALIZATION AND CONFIGURATION CONTROL-II

- BOOTING THE SYSTEM

- BOOT ROUTINE (AARDVARK)

- .MINI - I/O

- .MINI - XDELTA (SUBSET OF XDELTA)

- .MPC INITIALIZATION (TAPE, DISK, UNIT RECORD)

- SYSTEM IMAGE

- TIGR

- XDELTA

- DEBUG SCHEMA

- GHOST 1

- PATCHES

- PROCESSOR PATCHES (GENMDS)

- CONFIGURATION CONTROL CARDS

- LABELLED TAPE PORTION

- PROCESSORS

SYSTEM INITIALIZATION AND CONFIGURATION CONTROL-III

- TIGR - TABLE INITIALIZATION and GENERATION ROUTINE
 - ESTABLISH HARDWARE CONFIGURATION TABLES
 - .DCT
 - .CIT
 - .PPUT
 - .ETC.
 - ESTABLISH DYNAMIC INSTALLATION DEPENDENT TABLES
 - .USER TABLES
 - .I/O QUEUE SPACE
 - .ETC.
- SYSCON
 - RUN TIME PARTITIONING PROCESSOR

SYSTEM INITIALIZATION AND CONFIGURATION CONTROL -IV

- CONTROL (STANDARD SYSTEM PROCESSOR)

- SETS INSTALLATION DEPENDENT CONTROL AND DEFAULT PARAMETERS

- USED TO CHANGE CONTROL/DEFAULT PARAMETERS AS WORK PROFILE CHANGES

- TYPES OF PARAMETERS

- . NUMBER OF USERS
- . CORE USAGE
- . MULTIPROCESSING CONTROL
- . SCHEDULER CONTROL
- . BATCH STREAM CONTROL
- . I/O ACCELERATOR CONTROL
- . JOB SERVICE LIMITS
- . JOB RESOURCE LIMITS
- . JOB DEFAULTS

RECOVERY

- GOALS
 - CENTRALIZED SYSTEM ROUTINE (SCREECH)
 - AUTOMATIC, NO OPERATOR INTERVENTION
.INITIATED VIA FAULT OR LOGICAL INCONSISTENCY
 - FAST, 20 SEC TO 2 MINUTES
 - MINIMIZE FILE LOSS
- RECOVERY TYPES
 - ZAP/WARM BOOT
 - SCREECH
 - .SINGLE USER ABORT
 - .MULTI-USER ABORT
 - .FULL SYSTEM
- POWER FAIL SAFE RECOVERY
 - NON-VOLATILE MEMORY
 - MPC STATE
- CREATES DUMP WITH FORMATTED OUTPUT

PERFORMANCE AND CONTROL

- GOALS

 - LIKE CP-V ONLY BETTER

- TOOLS

 - INTERNAL

 - BUILT IN DATA COLLECTION

 - RESPONSE TIMES

 - DEVICE and CHANNEL BUSY TIMES

 - CPU UTILIZATION

 - PERFORMANCE TUNING VARIABLES

 - SCHEDULER CONTROLS

 - .QUANTUMS: QMIN,QUAN

 - .I/O BLOCK and UNBLOCK LIMITS

 - .BASE EXECUTION PRIORITIES:O,B,G

 - .I/O TIME ALLOWANCE

 - I/O ACCELERATOR CONTROL

 - .READ AHEAD

 - .DISASSOCIATED WRITE

 - .SEVERAL CACHE TYPES

 - EXTERNAL

 - STATS

 - CALMON

 - SUMMARY

CALMON OUTPUT

**** SUMMARY STATISTICS *

CAL1,1 REPORTS

	CAL COUNT	% OF CALCS	CPU TIME	% CAL TIME	% ALL TIME	AVG CPU PER CAL	I/O COUNT	% CAL I/O	AVG I/O PER CAL	AVG#BYTE PER CA
:REW	141	.05	158	.02	0	1.12	0	-	-	693.3
:WEOF	705	.25	976	.10	0	1.38	640	.60	.91	99.4
:DEVICE(PAGE)	572	.20	1,586	.16	.01	2.77	47	.04	.08	371.2
:DEVICE(VFC)	506	.18	248	.03	0	.49	0	-	-	257.3
:SETDCH	17,606	6.28	8,516	.87	.04	.48	0	-	-	917.1
:DEVICE(DRC)	9	0	6	0	0	.67	0	-	-	3,097.7
:DELREC	51	.02	352	.04	0	6.90	3	0	.06	117.6
:MOVE	652	.23	146,736	15.01	.74	225.06	16,065	15.02	24.64	1.19E
:TFILE	26	.01	408	.04	0	15.69	78	.07	3.00	9,377.6
:READ(FILE)	96,007	34.27	211,120	21.60	1.06	2.20	27,100	25.34	.28	519.0
:READ(LBL TAPE)	2,153	.77	4,252	.43	.02	1.97	95	.09	.04	80.6
:READ(DEVICE)	10	0	30	0	0	3.00	10	.01	1.00	-
:READ(COC)	3,700	1.32	14,432	1.48	.07	3.90	0	-	-	6.8
:READ(SYMBIONT)	791	.28	2,392	.24	.01	3.02	74	.07	.09	112.5
:WRITE(FILE)	26,706	9.53	55,614	5.69	.28	2.08	4,761	4.45	.18	233.2
:WRITE(LBL TAPE)	27,453	9.80	56,424	5.77	.28	2.06	6,383	5.97	.23	631.7
:WRITE(DEVICE)	10,057	3.59	17,210	1.76	.09	1.71	9,534	8.92	.95	332.5
:WRITE(COC)	8,937	3.19	43,330	4.43	.22	4.85	0	-	-	14.5
:WRITE(SYMBIONT)	30,970	11.05	60,294	6.17	.30	1.95	2,405	2.25	.08	89.9
:TRUNC	66	.02	190	.02	0	2.88	88	.08	1.33	84.0
:OPEN(FILE)	5,136	1.83	70,780	7.24	.36	13.78	5,183	4.85	1.01	4,016.9
:OPEN(LBL TAPE)	1,198	.43	11,904	1.22	.06	9.94	4,788	4.48	4.00	432.4
:OPEN(DEVICE)	59	.02	296	.03	0	5.02	96	.09	1.63	477.0
:OPEN(COC)	176	.06	390	.04	0	2.22	0	-	-	.8
:OPEN(SYMBIONT)	524	.19	1,236	.13	.01	2.36	0	-	-	.2
:OPEN(TSI-FIL)	8,096	2.89	69,976	7.16	.35	8.64	599	.56	.07	5,301.0
:CLOSE(FILE)	4,052	1.45	41,146	4.21	.21	10.15	9,970	9.32	2.46	6,108.5
:CLOSE(LBL TAPE)	1,171	.42	5,586	.57	.03	4.77	8,049	7.57	6.92	327.8
:CLOSE(DEVICE)	736	.26	842	.09	0	1.14	9	.01	.01	1.0
:CLOSE(COC)	232	.08	302	.03	0	1.30	0	-	-	17.4

STATS OUTPUT

STATISTICS ON-LINE

INTERVAL FROM 14:51 TO 14:55

MINS SINCE STARTUP= 992
 NUMBER OF USERS = 37
 NUMBER OF ONLINE = 26
 NUMBER OF BATCH = 5
 NUMBER OF GHOSTS = 6
 90% RESPONSE TIME = 1500

CPU %	ALL	SNAP	I/O PER MIN	ALL	SNAP
BATCH EXEC	42.6	94.3	SERVICE REQ	2897	9605
BATCH SERV	5.0	16.4	INTERACTIONS	19	29
ONLINE EXEC	14.4	34.2	CHAR IN	265	293
ONLINE SERV	5.2	14.9	CHAR OUT	2414	3572
GHOST EXEC	1.6	.7	TERM WRITES	98	150
GHOST SERV	1.3	.3	I/O ACCESSES	676	1269
MONITOR SERV	8.5	15.7	# TRUNCS	0	0
IDLE	48.3	.0	AIR ATTEMPTS	71	89
SWAP WAIT	.4	.7	AIR HITS	56	35
I/O WAIT	8.6	1.9	SYMBIONT	74	68
I/O&SWP WAIT	1.3	12.1	IN SWAPS	30	99
TOTAL	137.2	191.3	OUT SWAPS	21	67

SCPU USE %	ALL	SNAP	EVENT RATE/MIN	ALL	SNAP
SCPU #1 EXEC	37.3	91.0	MASTER CALS	2062	6224
SCPU #1 IDLE	62.1	9.2	SCPU #1 CALS	836	3381
SCPU #1 TOT	99.4	100.2	MASTER SCHEDS	2135	6417
			SCPU #1 SCHEDS	751	3176

HOW DO WE KNOW WE'LL BE FAST ENOUGH

- **ALGORITHMS PROVEN IN CP-V**
- **USAGE PATTERNS KNOWN**
- **INTEGRAL PERFORMANCE MONITOR**
- **HIGH LEVEL LANGUAGE PERMITS SYSTEM WIDE OPTIMIZATION**
- **USE OF STANDARD BENCHMARKS**
- **NSA HARDWARE USED FOR CONTEXT SWITCH**
- **OFFLOADING OF COMMUNICATIONS TO L6 FRONT ENDS**

ERROR LOGGING AND LISTING

- GOALS

- NEW TOOL COMMON WITH GCOS, TCOS
- ONLINE and BATCH OPERATION
- DESIGNED BY FED FOR FED

- ERROR TYPES

- 100 INFORMATIONAL
- 200 OPERATIONAL
- 300 HARDWARE ERRORS
- 400 SOFTWARE ERRORS

- REPORT TYPES

- MASS RAW DATA DUMPS
- CHRONOLOGICAL LISTING
- MEDIA ALERTS
- ERROR BY DEVICE/COMMAND
- ERROR BY MEDIA ID/UNIT
- SUMMARY REPORT
 - .RECOVERY
 - .MPC STATISTICS
 - .EDAC SYNDROME



STRENGTH OF CP-6 IS IN ITS INTERFACES

- USER STRUCTURE
- SYSTEM SERVICE INTERFACE
- PROGRAM BINDING INTERFACE
- PROGRAM CALLING INTERFACE
- USER to SYSTEM INTERFACE

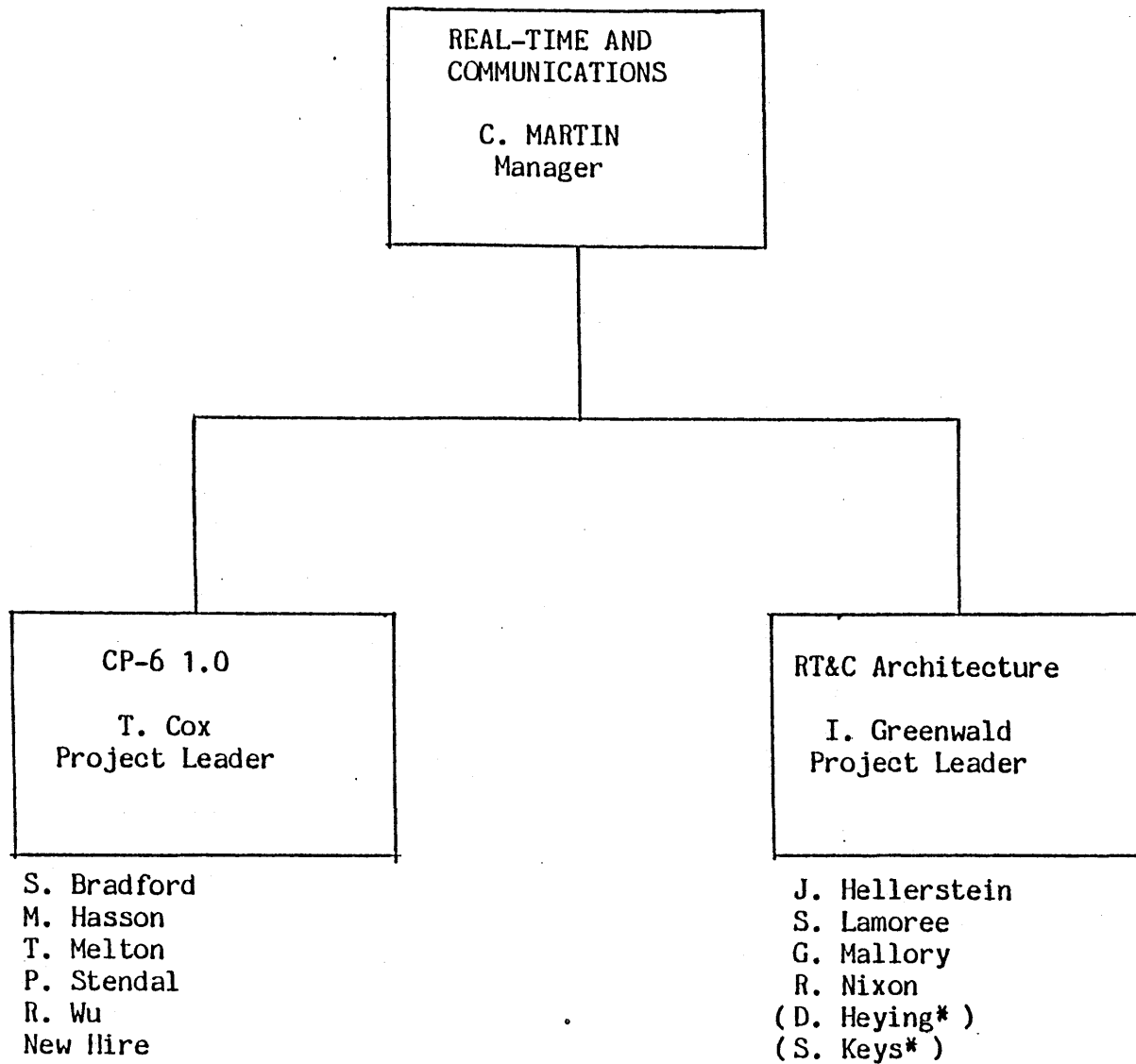
COMMON/SHARED PARTS OF CP-6

- USER STRUCTURE
- SCHEDULER
- ALL I/O INCLUDING FILE MANAGEMENT
- SHARED PROCESSORS
- SHARED LIBRARIES (RUNTIME AND ALTERNATE)
- COMMAND PROGRAM
- DEBUGGER
- SPECIAL SERVICES (M\$FID, M\$PARSE, M\$ERRPRT, HELP)

CP-66

CP-6/1.0 RT&C PROJECT

- PROJECT INTRODUCTION
 - ORGANIZATION
 - KEY EVENTS
 - PROJECT OVERVIEW
 - ISSUES
- SOFTWARE FACTORY
 - IMPLEMENTATION LANGUAGE
 - PROGRAM TRANSPORTATION
 - DEBUGGING
 - SUPPORT TOOLS
 - RELEASE SUPPORT
- RELEASE 1.0
 - TECHNICAL DESCRIPTION - OVERVIEW
 - TECHNICAL DESCRIPTION - HARDWARE
 - TECHNICAL DESCRIPTION - SOFTWARE



* Principal participants from CP-6 OS Section, as required.

RT&C PROJECT - KEY EVENTS

- o SCHEDULE CONSTRAINTS
- o COUPLER
- o CONTROL FORTRAN AND SIL-6

RT&C PROJECT - SCHEDULE CONSTRAINTS

- o LANGUAGE PROCESSOR CHECKOUT
 - APPROACH APRIL/MAY, 1978
 - PRODUCTIVE JULY, 1978
- o SERVICE PROCESSOR CHECKOUT
 - BEGIN T/S ENVIRONMENT CHECKOUT JAN/FEB, 1978
 - COMPLETE CHECKOUT MARCH/APRIL, 1978
- o BASIC T/S CAPABILITY REQUIRED BY JANUARY, 1978

9/19/77 PDR/CM

RT&C PROJECT - COUPLER

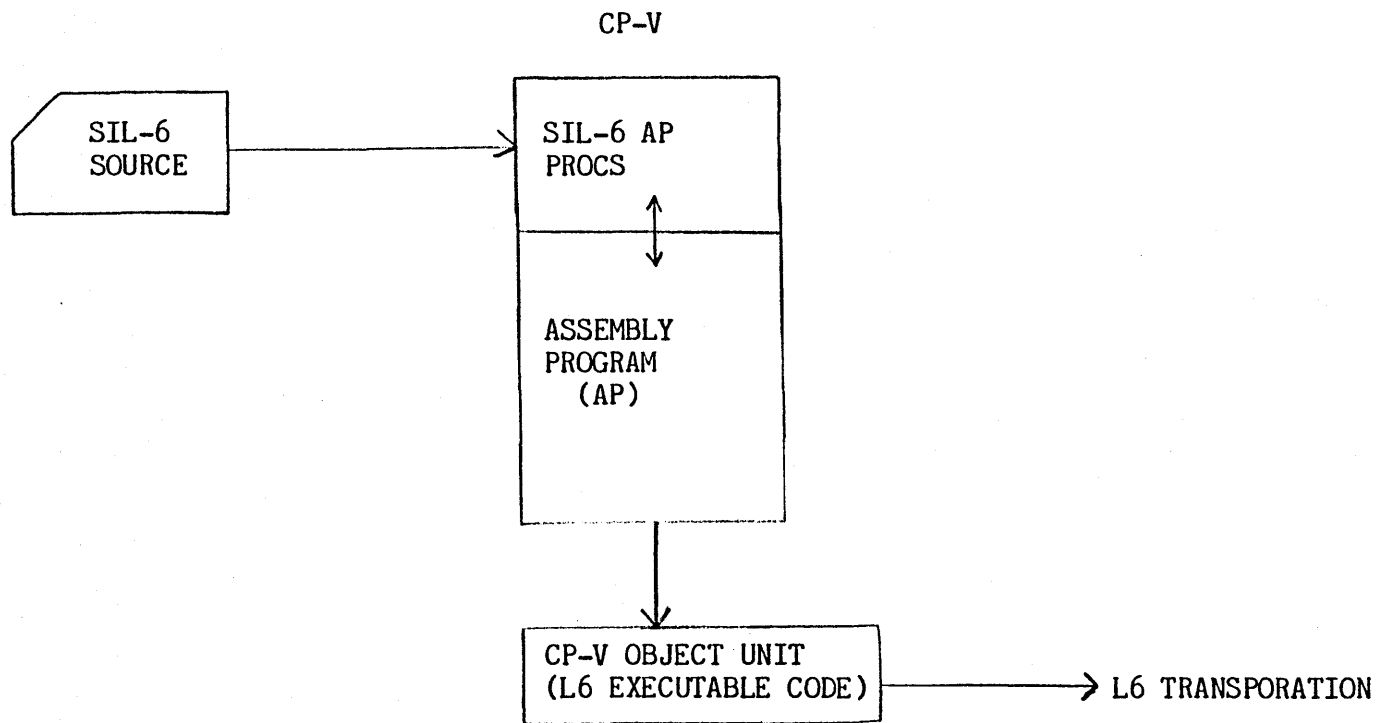
- o 1976 - ASSUMED A COUPLER AVAILABLE BY JULY, 1977.
- o MARCH 1977 - COUPLER (PROTOTYPE) COMMITTED 4Q77.
- o MAY 1977 - COUPLER COMMITTED TO 746.
- o JULY 1977 - COUPLER CDR HELD
- o NOVEMBER 1977 - INSTALL COUPLER AT LADC
- o DECEMBER 1977 - COUPLER OPERATIONAL AT LADC

RT&C PROJECT - CONTROL FORTRAN AND SIL-6

- o 1976 - ASSUMED AVAILABILITY OF CF IN FEBRUARY 1977
- o FEBRUARY 1977 - DELIVERY SLIPPED TO MARCH 1977
- o APRIL 1977 - CF PDR HELD
- o MAY 1977 - CF RESPONSIBILITY MOVED TO BILLERICA
- o MAY 1977 - SIL-6 DECISION MADE
- o JULY 1977 - WORKING VERSION OF SIL-6 AVAILABLE FOR USE

9/19/77 PDR/CM

RT&C PROJECT - SIL-6



RT&C PROJECT - RELEASE CONTENT

- o RELEASE 1.0
 - CP-V BASED
 - TTY TERMINAL SUPPORT
 - IRBT, HASP
 - RBT, 2780/3780
 - (-- CP-V SOFTWARE FACTORY)

- o RELEASE 2.0
 - REAL TIME
 - REMOTE CP
 - VIP TERMINAL SUPPORT
 - CP-6 SOFTWARE FACTORY
 - TRANSACTION PROCESSING

- o RELEASE 3.0
 - 66/85 SUPPORT

RT&C PROJECT - RELEASE 1.0 STAGING

- o A - SYSTEM (805T/S) - JAN 1978
 - INTEGRATED INTO HOST H-SYSTEM
 - SUPPORT FOR DELTA, CP AND USER PROGRAMS
 - TI SUPPORT
 - ECHOPLEX
 - CANNED SALUTATION AND PROMPTS
 - NO ESCAPE SEQUENCES
 - CP-V ACTION FOR CONTROL Y,X AND BREAK AND RUB-OUT

RT&C PROJECT - RELEASE 1.0 STAGING (CONT.)

- o B - SYSTEM - JULY 1977
 - INTEGRATED INTO HOST L-SYSTEM
 - T/S FOR LANGUAGE PROCESSOR CHECKOUT
 - TAB STOPS
 - TYPE AHEAD
 - ASCII APL TERMINAL SUPPORT
 - SYMBIONT I/O MODE
 - OC I/O MODE
 - USER PROGRAM ACCESS TO TERMINAL CONTROL ATTRIBUTES
 - MAJORITY OF ESCAPE SEQUENCES

RT&C PROJECT - RELEASE 1.0 STAGING (CONT.)

- o C - SYSTEM
 - INTEGRATED INTO HOST O-SYSTEM
 - RELEASE 1.0 FUNCTIONALITY
 - 2780/3780 AND HASP SUPPORT
 - AUTOBAUD
 - FULL TAB CAPABILITY
 - PAGINATION
 - TRANSPARENT I/O
 - FULL ESCAPE SEQUENCE CAPABILITY

RT&C PROJECT - ISSUES

- o SNA AND HDNA
- o CP-6 SOFTWARE FACTORY
- o DIAGNOSTICS
- o HOST RESIDENT SYSTEM
- o REAL TIME

RT&C PROJECT - SNA, HDNA ISSUE

- NOT PART OF ORIGINAL PROJECT STRATEGY - CHANGE OF SCOPE

- SNA AND IBM CO-RESIDENCY SEEN AS HIGHER PRIORITY BY MARKETING

- HDNA AS DEFINED IS NOT CONSISTENT WITH CP-6 RESPONSE TIME REQUIREMENTS

- CONTINUES TO EVOLVE

- NET RESULT - NOT CONSIDERED IN CP-6/1.0

RT&C PROJECT - CP-6 SOFTWARE FACTORY

- o MOVE FROM CP-V TO CP-6

- o SIL-6

- o DUAL (DYNAMIC UNIVERSAL ASSEMBLY LANGUAGE)

- o L6 ASSEMBLER

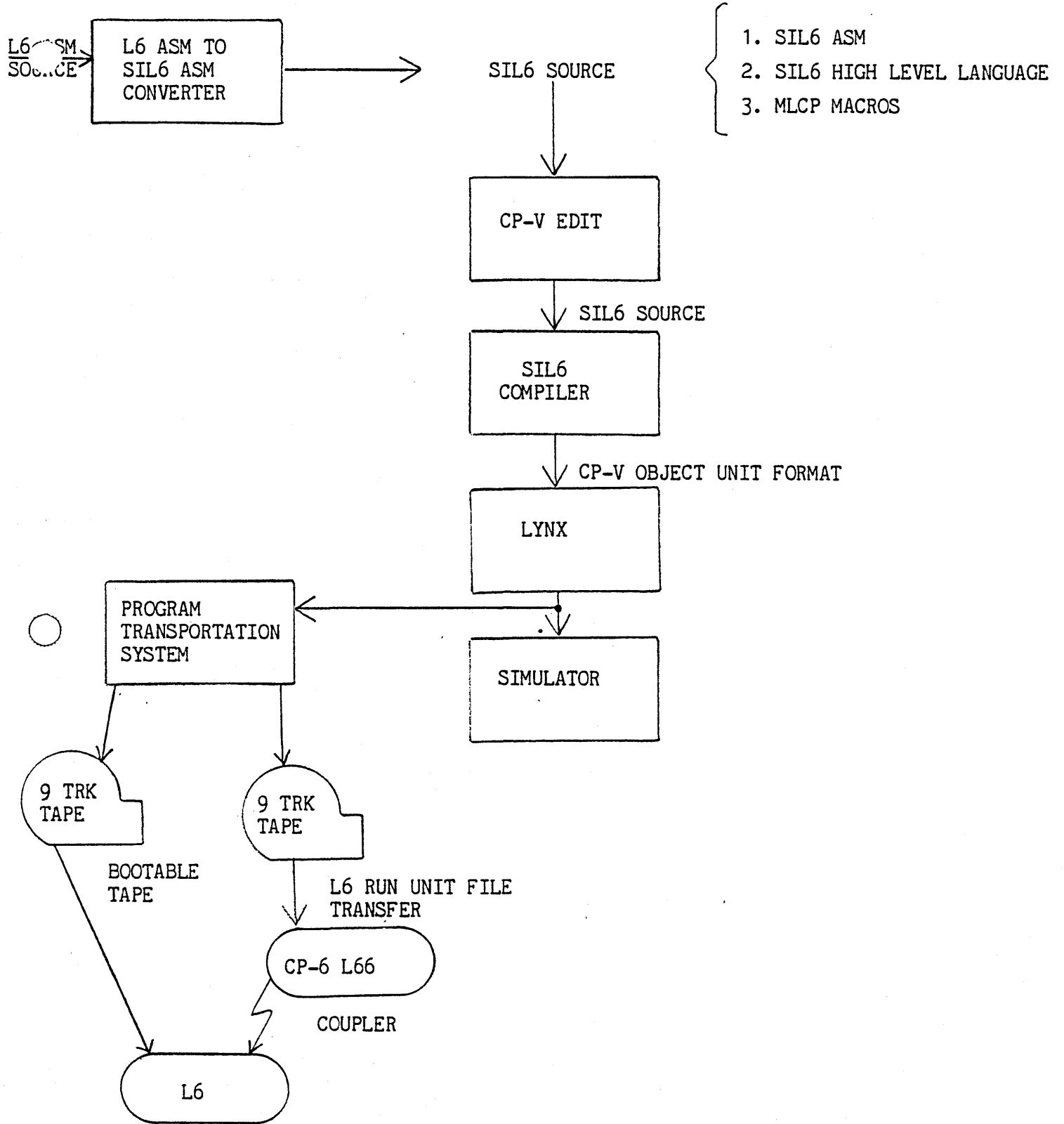
- o L6/PL-6

- o BILLERICA SYSTEMS IMPLEMENTATION LANGUAGE
 - PL-6
 - PLH

L6 SOFTWARE FACTORY ENVIRONMENTS

1. INITIAL (RELEASE 1.0) CP-6 DEVELOPMENT USING CP-V RESIDENT SOFTWARE FACTORY
2. RELEASE 2 AND 3 DEVELOPMENT USING CP-6 RESIDENT SOFTWARE FACTORY
3. CUSTOMER REAL-TIME DEVELOPMENT (RELEASE 2 AND BEYOND) USING CP-6 RESIDENT FACTORY

CP-V RESIDENT L6 SOFTWARE FACTORY



SIL6 LANGUAGE

- HIGH LEVEL LANGUAGE
- DATA INDEPENDENCE
- BLOCK PROGRAM STRUCTURES
- ACCESSIBILITY TO L6 REGISTERS
- SYMBOL REF/DEF STACK FOR SYMBOLIC DEBUGGING
- SUPPLEMENTARY ASSEMBLER MNEMONICS FOR TIME-CRITICAL OR HARDWARE RELATED CODE
- MLCP MNEMONICS
- IMPLEMENTATION IN AP-EASY TO EXPAND

SUMMARY OF SIL6 CAPABILITIES

- PROGRAM STRUCTURE COMMANDS
 - LOOP/EXITLOOP/ENDLOOP
 - IF/ELSEIF/OTHERWISE/ENDLOOP
 - DOCASE/CASE/ENDCASE
 - CALL/SUBROUTINE/RETURN

- ITEM DEFINITION FUNCTIONS
 - TYPE - BIT, BYTE, WORD, DOUBLEWORD, POINTER
 - ATTRIBUTES - SIGNED, WRITE PROTECT, DIMENSION, FIELD SPECIFICATION WITHIN A WORD, OFFSET FROM A BASE, FIXED ADDRESS
 - ITEM MAY BE MEMORY LOCATION OR REGISTER
 - LITERAL ADDRESSES AND VALUES

SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- ITEM MANIPULATION COMMANDS
 - MOVE
 - INCREMENT/DECREMENT
 - SETT/RESET
 - CLEAR

- COMPARISON FUNCTIONS
 - NOT
 - EQUAL/NEQUAL
 - GT/AGT
 - GTE/AGTE
 - LT/ALT
 - LTE/ALTE

SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- COMPUTING FUNCTIONS

- SUM
- DIF
- PRODUCT
- QUOTIENT
- MODULO
- AND
- OR
- EOR/XOR

- NON-COMPUTING FUNCTIONS

- HIGHDIM/LOWDIM
- ASC
- MAXOFFSET/MINOFFSET/NEXTOFFSET
- AMAX/AMIN
- ADDR

SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- ASSIGNED HARDWARE REFERENCES

- R1 - R7
- B1 - B7
- CARRY/NO CARRY
- OVERFLOW/NO OVERFLOW
- IOACK/IONACK
- LEVEL
- INHIBIT_LEVEL

- MISCELLANEOUS COMMANDS

- ENABLE
- DISABLE/INHIBIT
- TRIGGER

SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- DECISION TABLE PROCEDURES
 - TABLE
 - TEST(S)
 - RULE
 - ACTION(S)
 - REPEAT
 - RETURN_TRUE
 - RETURN_FALSE
 - ENDTABLE

SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- ASSEMBLY LIST CONTROL
 - PAGE
 - TITLE
 - WARNING
 - LOLIST
 - HEX

- ASSEMBLY CONTROL STATEMENTS
 - DEBUG
 - STARTSIL6
 - STARTASM
 - LABEL
 - ORGG
 - ODDBOUND/EVENBOUND
 - DC
 - RESV
 - ATEXT/ATEXTC

SIL6 EXAMPLE

```
ALPHA      FNAME      WORD,(OFFSET,3),(FIELD,3,7),WP
BETA       FNAME      WORD,(OFFSET,15),(FIELD,1,9),(DIM,1,10)
GAMMA      FNAME      BYTE,(OFFSET,1)
BASE       FNAME      (BREG,1)
INDEX      FNAME      (RREG,2),(FIELD,0,7)
           .
           .
           .
           IF EQUAL (GAMMA(BASE),ALPHA(BASE))
           MOVE ALPHA(BASE),TO,BETA(BASE,INDEX))
           OTHERWISE
           MOVE SUM(ALPHA(BASE),BETA(BASE,INDEX)),TO,GAMMA(BASE)
           INCREMENT INDEX
           ENDIF
```

9/14/77 TC

SIL6 DECISION TABLE EXAMPLE

```
TESTCASE      TABLE      PASSTEXT, TEXT1, TEST2
              RULE        T, F, T
              ACTIONS     ACTION1, ACTION2, REPEAT
              RULE        F, T, T
              ACTIONS     ACTION2, MOVE(SUM(B3, B4)), TO, B4
              ENDTABLE
              .
              .
TEST1         SUBROUTINE  NOLOCAL(B4)
              IF          R4
              RETURN_TRUE
              ENDIF
              RETURN_FALSE
              ENDSUB
              .
              .
ACTION2      SUBROUTINE  SAVE(R2, R4)
              MOVE       R5, TO, R4
              ENDSUB
              .
              .
              .
```

DEBUGGING

- SIMULATOR AND L6 DEBUGGER - SUBSET OF CP-6
DELTA FUNCTIONALITY AND COMMAND SYNTAX
 - BREAKPOINTS
 - TRACE
 - STEP
 - MODIFY
 - DISPLAY
 - DUMP
 - SYMBOLIC INPUT

9/14/77 TC

DEBUGGING - CONTINUED

- BES2 MDUMP
- UPLINE DUMP VIA COUPLER
- DUMP ANALYZER
- MLCP DEBUGGER
- EVENT RECORDING

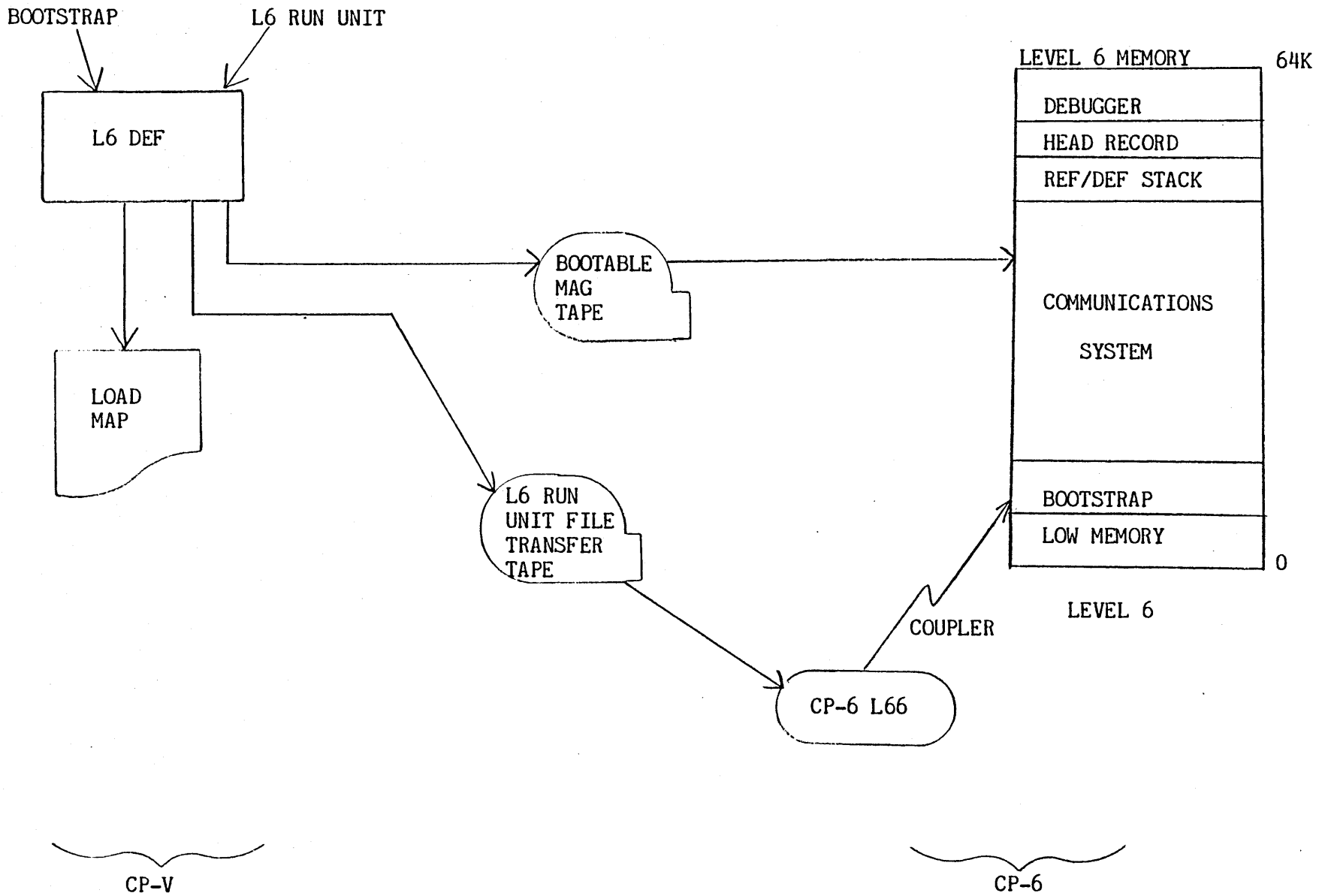
9/14/77 TC

SIMULATOR

- HANDLES INTERRUPTS, TRAPS
- L6/36 OR L6/43 INSTRUCTION SET ACCEPTED
- PROVIDES INSTRUCTION TIMING SUMMATION
- I/O INSTRUCTIONS CAUSE DISPLAY OF PERTINENT INFORMATION -
NO I/O SIMULATION IS ATTEMPTED
- EXTENSIVE DEBUGGER

9/14/77 TC

PROGRAM TRANSPORTATION SYSTEM



SUPPORT TOOLS

- EXTRACT
- DRAW
- L6 ASM TO SIL6 ASM CONVERTER
- GLOBAL XREF

9/14/77 TC

RELEASE SUPPORT

- PATCHER - WILL PATCH L6 RUN UNIT ON CP-V
OR CP-6
- REMOTE DEBUGGER
- UPLINE DUMP VIA COUPLER
- DUMP ANALYZER

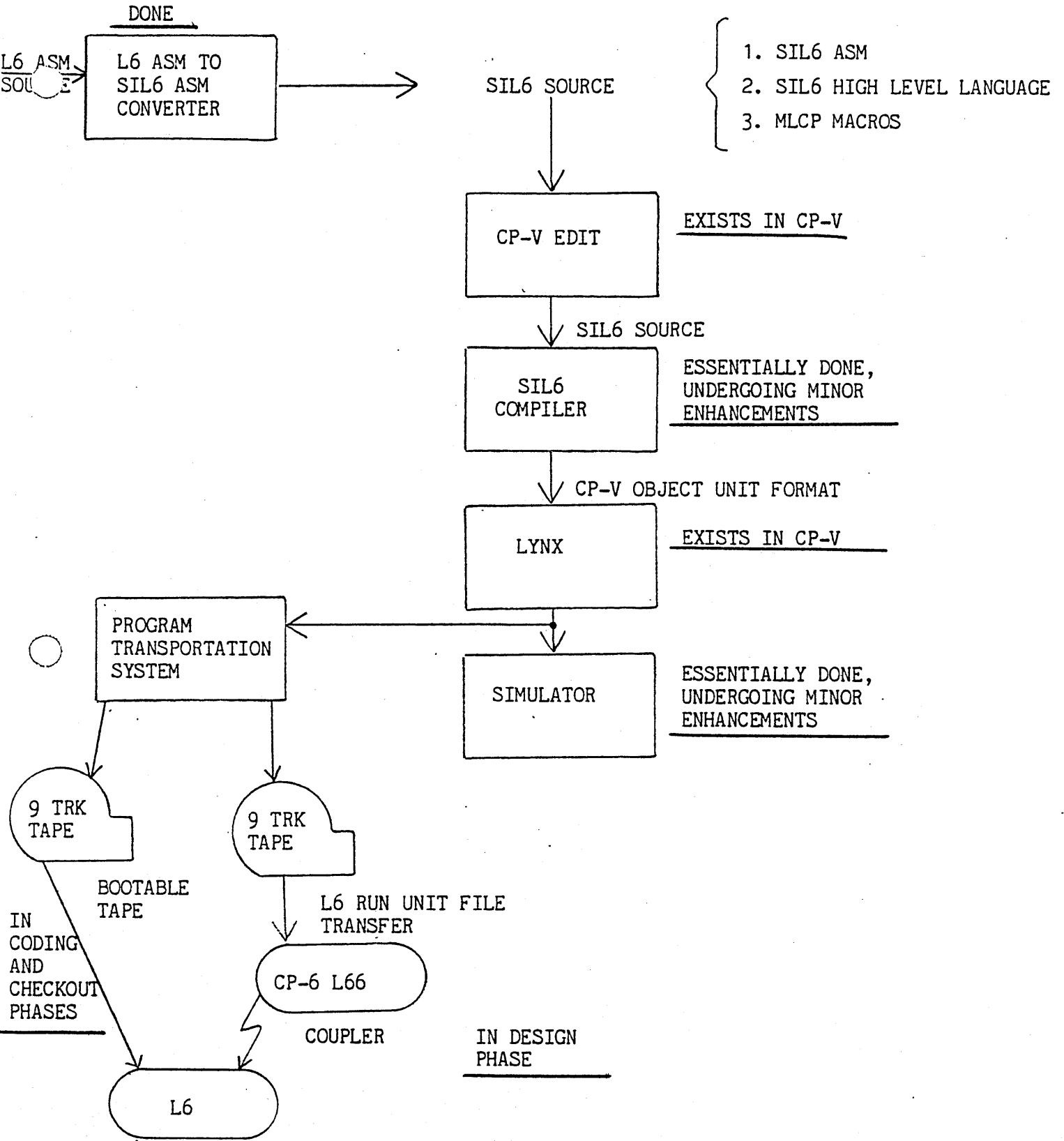
9/14/77 TC

CONVENTIONS AND STANDARDS

- CP-6 COMPATIBLE
- DOCUMENTATION STANDARDS
- NAMING CONVENTIONS AND FUNCTIONAL CODE GROUPS
- CODING STANDARDS
- CONTROLLED DEVELOPMENT ACCOUNTS

9/14/77 TC

CP-V RESIDENT L6 SOFTWARE FACTORY



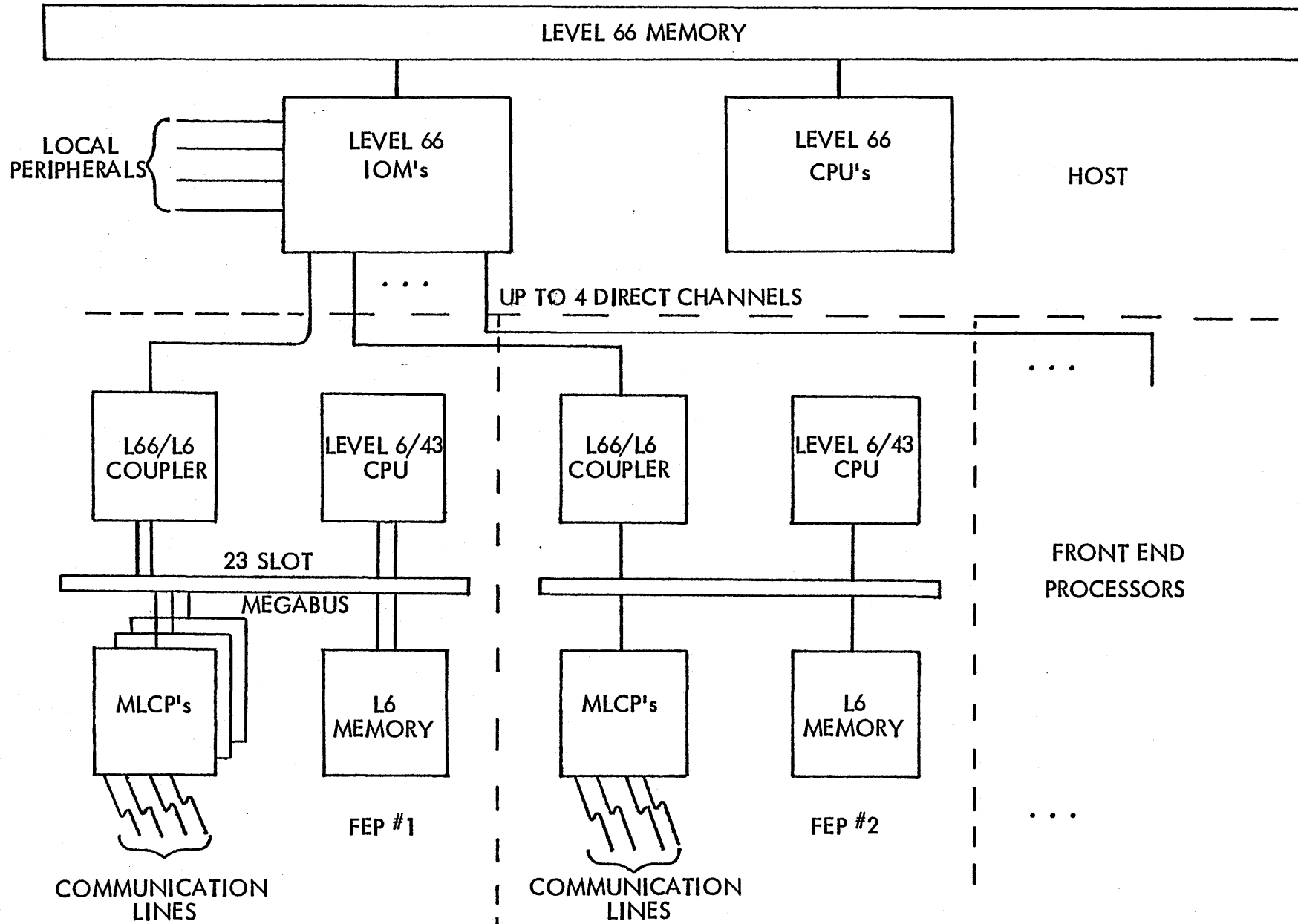
CP-6 COMMUNICATIONS - RELEASE 1.0

- o FUNCTIONALITY
- o HARDWARE
 - o OVERVIEW
 - o L66/L6 COUPLER
 - o LEVEL 6 MLCP
- o LEVEL 6 SOFTWARE DESIGN
 - o OVERVIEW
 - o NUCLEUS
 - o FRONT END INTERFACE
 - o COC HANDLER
 - o RBT/IRBT HANDLER
- o LEVEL 66 SOFTWARE

COMMUNICATIONS SOFTWARE REQUIREMENTS -- RELEASE 1

- OFFLOAD PROCESSING FROM THE L66
- USE LEVEL 6/43
- CP-V STYLE TERMINAL SUPPORT
 - TTY COMPATIBLE TERMINALS
 - 2780/3780 COMPATIBLE RBTS
 - HASP COMPATIBLE IRBTS
 - CUSTOM DEVICES (TRANSPARENT I/O)
- HOST TO HOST COMMUNICATION (CP-6 TO CP-6, CP-6 TO CP-5)
- ESTABLISH A BASE FOR FUTURE RELEASES

CP-6 HARDWARE ENVIRONMENT - COMMUNICATIONS VIEW



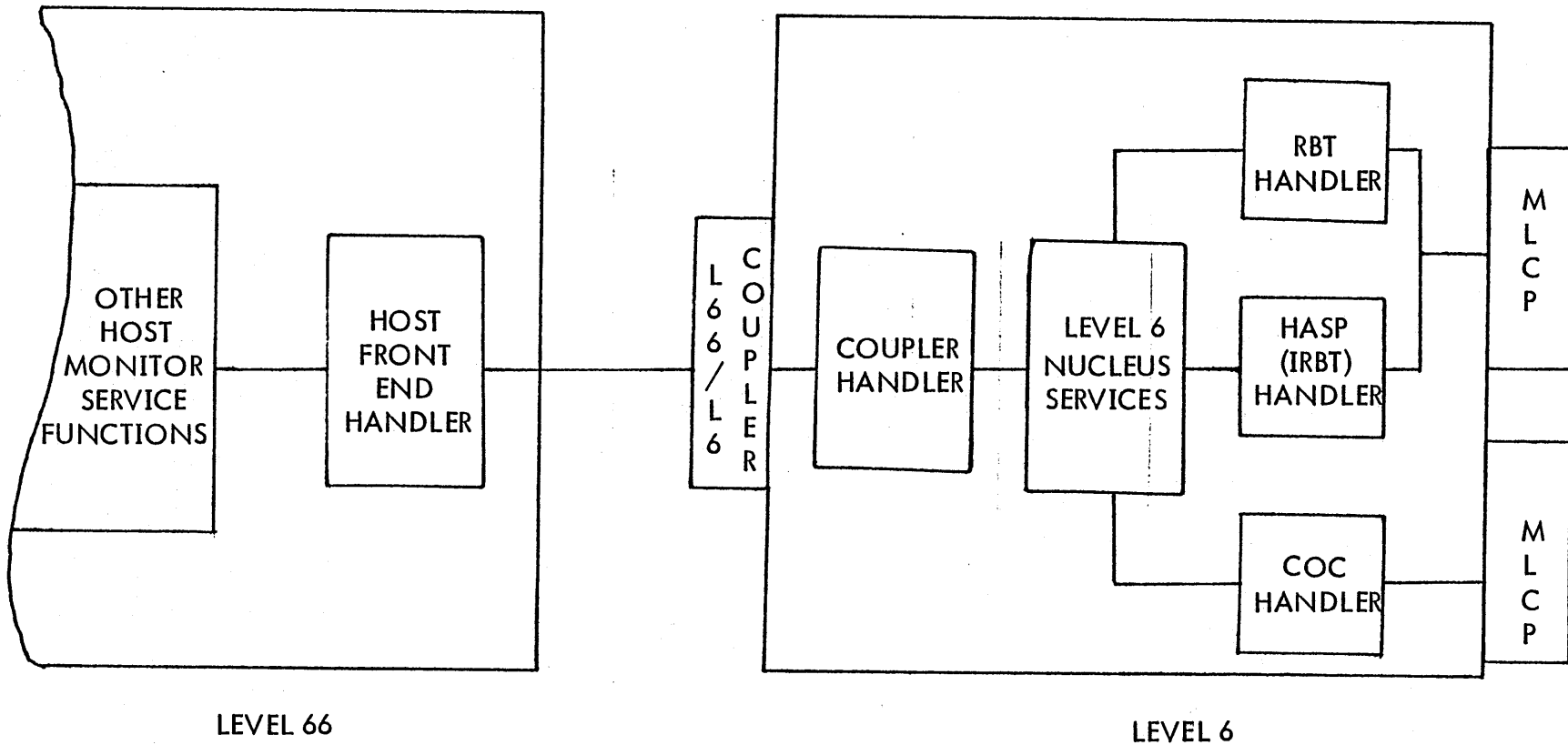
L66/L6 COUPLER HARDWARE

- HOST INITIATED BOOT LOAD
- BYTE ORIENTED ASCII MODE; 1 MEGABYTE TRANSFER
- BINARY MODE; 500 KB TRANSFER
- LEVEL 6 INITIATED I/O WITH COMMAND CHAINING

MLCP HARDWARE CAPABILITIES

- GENERAL CAPABILITIES
 - PROGRAMMABLE
 - EIGHT RECEIVE/TRANSMIT CHANNEL PAIRS
 - CONTROL CHARACTER AND END-OF-MESSAGE DETECTION
 - L6 COMMAND CHAINING CAPABILITY
- ASYNCHRONOUS LINE SUPPORT (COC)
 - PROGRAMMABLE LINE SPEED AND CHARACTER FORMAT (AUTOBAUD)
 - SPEEDS UP TO 19.2K BITS/SEC
- SYNCHRONOUS LINE SUPPORT
 - AUTOMATIC CYCLIC REDUNDANCY CHECK
 - SPEEDS UP TO 72K BITS/SEC

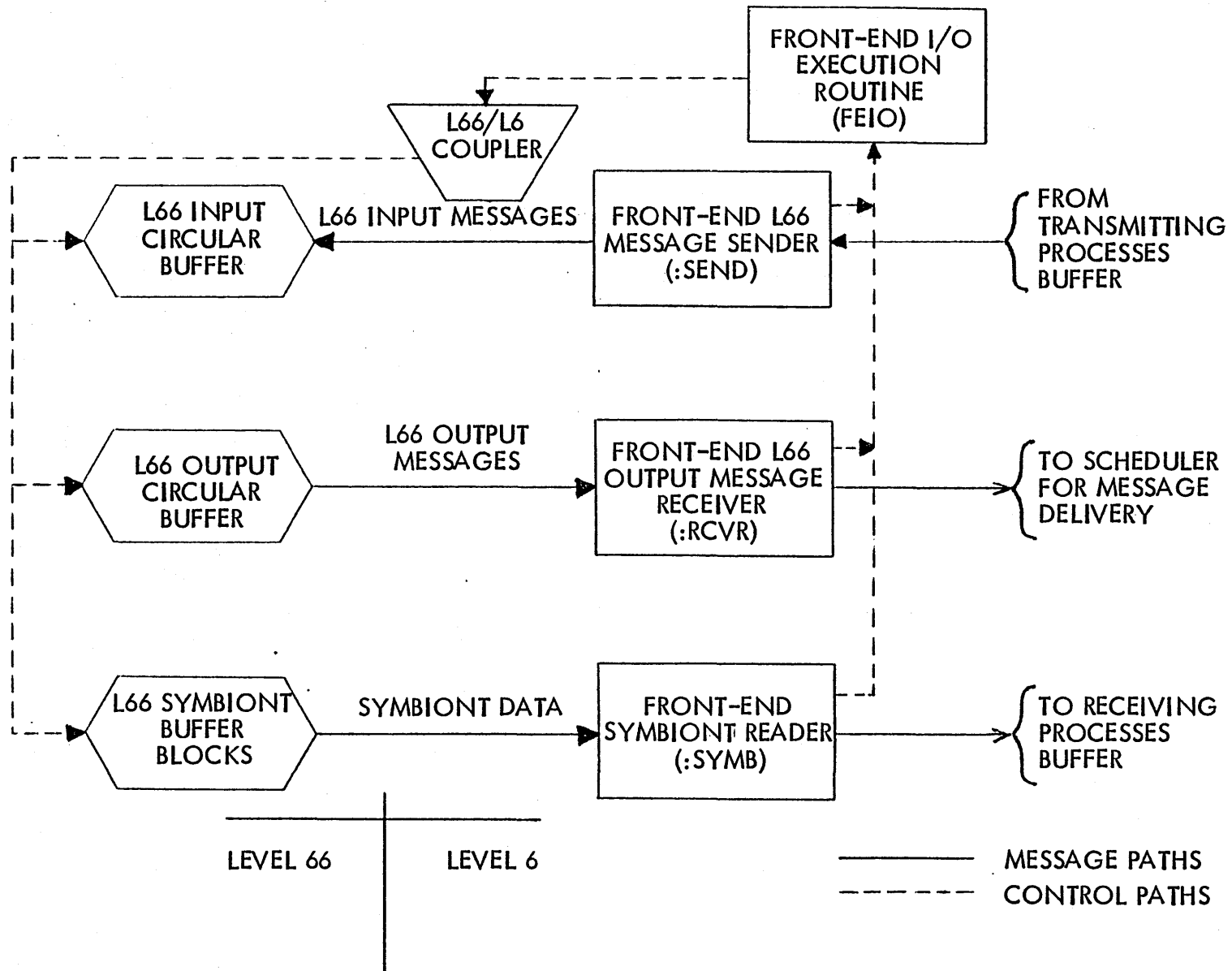
LEVEL 6 SOFTWARE STRUCTURE FOR RELEASE 1



LEVEL 6 NUCLEUS SERVICES

- CLOCK SERVICES
- MEMORY ALLOCATOR
- SCHEDULER
 - WAKE-UP
 - GENERAL SERVICE
 - LOGICAL PATH SERVICE
 - MESSAGE SERVICE
- ADMINISTRATIVE SERVICES
 - LOGICAL PATH CONNECTION, DISCONNECTION
 - ERROR LOGGING
 - CONFIGURATION CONTROL

L6 FRONT END INTERFACE STRUCTURE AND MESSAGE FLOW



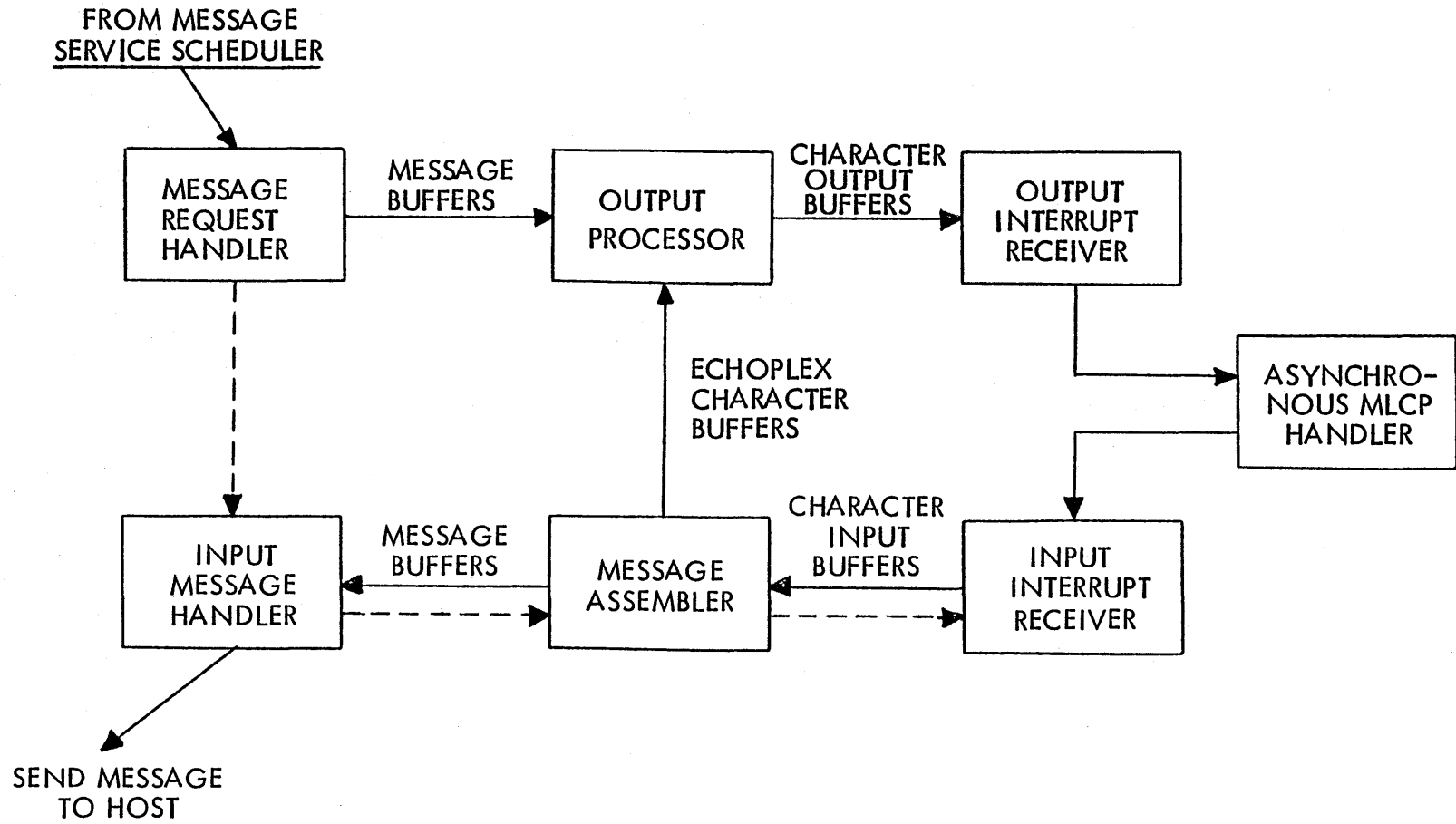
CP-6 LEVEL 6 COC HANDLER CAPABILITIES

- TYPE AHEAD
- TERMINAL TYPES AND TIMING ALGORITHMS
- TRANSPARENT I/O
- PAGINATION
- INPUT EDITING AND CONVERSION
- TABULATION CONTROL
- TERMINAL COUPLING
- TERMINAL TAPE INPUT

COC USER CONTEXT MAINTAINED IN THE LEVEL 6

- INPUT PROMPTS
- TAB STOPS
- ACTIVATION SET
- TERMINAL TYPE, AND TRANSLATION
- TIMING ALGORITHM
- MODES OF OPERATION IN EFFECT
- IMAGE OF PREVIOUS INPUT RECORD FOR RECALL

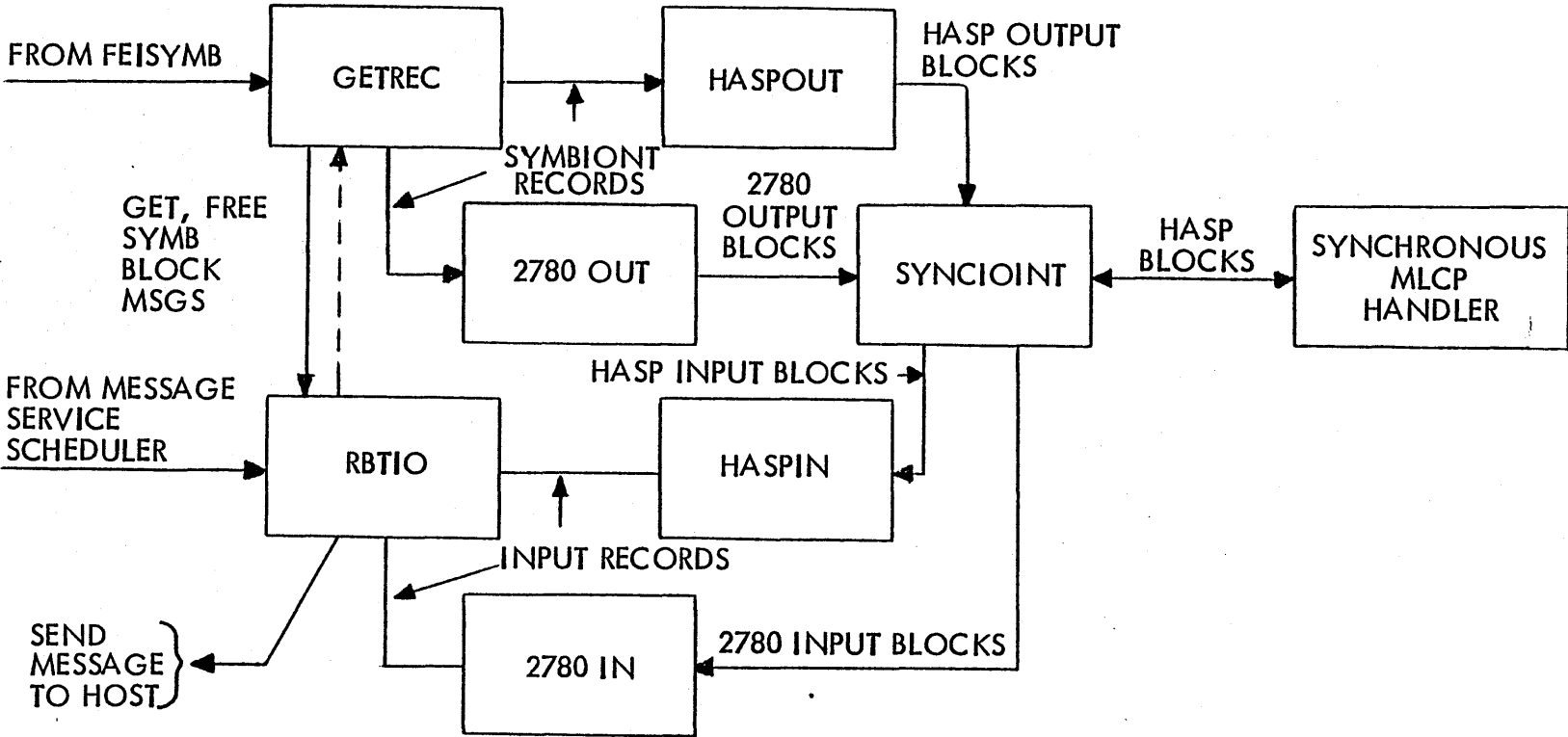
CP-6 LEVEL 6 COC HANDLER STRUCTURE



CP-6 REMOTE BATCH CAPABILITIES

- SUBMIT JOBS, RECEIVE RESULTS, AND TRANSMIT FILES
- 2780/3780 AND HASP-MULTILEAVING PROTOCOL SUPPORT
- HOST TO HOST COMMUNICATION
 - MASTER OR SLAVE OVER EACH CONNECTION
- WORKSTATION CONCEPT
 - CP-6 HOST SUPPLIES CHARACTERICS DURING CONNECTIONS
 - WIDE RANGE OF DEVICES CAN BE SUPPORTED
- LOGICAL PATH TO HOST FOR EACH ACTIVE DEVICE
- REMOTE OPERATOR COMMANDS
 - DEVICE AND FILE CONTROL
 - OBTAIN STATUS
 - COMMUNICATION WITH HOST OPERATOR

CP-6 LEVEL 6 RBT/IRBT HANDLER STRUCTURE



HOST RESIDENT COMMUNICATIONS SUPPORT

- ADMINISTRATIVE SERVICES
- INITIALIZATION AND RECOVERY
- LEVEL 6 CRASH SUPPORT
- WORKSTATION DEFINITION
- HOST LOGON
- COMMUNICATION RELATED I/O FUNCTIONS
 - L66 COUPLER HANDLER
 - INTERFACE TO READ/WRITE COMMANDS
 - INTERFACE TO LOGICAL CONNECTIONS (OPEN)
 - PROGRAM CONTROL OF TERMINAL MODES

CP-6 LANGUAGE PROCESSORS

(DEVELOPED AT LADC)

PL-6

BASIC

ANS FORTRAN

APL

TEXT

CP-6 LANGUAGE PROCESSORS

HIGHLIGHTS

FUNCTIONALITY \geq CP-V LANGUAGES

COMPATIBILITY WITH ANS

COMPATIBILITY WITH H.I.S UNIFICATION

COMMON RUN-TIME LIBRARY FOR APL, BASIC, FORTRAN

SHARED PROCESSORS

PL-6 OVERVIEW

o LANGUAGE REQUIREMENTS

o COMPILER HISTORY

o COMPILER DESIGN

- o LANGUAGE DEFINITION
- o CODE GENERATION
- o EFFICIENT USE OF HARDWARE
- o EFFICIENT ACCESS TO MONITOR SERVICES

LANGUAGE DEFINITIONS

PL-6

PL-1

ADDRESS RESOLUTION

COMPILATION

COMPILATION/RUNTIME

ASYNCHRONOUS EVENT PROCESSING

USER CONTROLLED

ON CONDITION CHAIN IN
AUTOMATIC

AUTO ALLOCATION STRATEGY

USER CONTROLLED

NO CONTROL

EFFICIENCY FEATURES

PL-6

PL-1

DEDICATED POINTER REGISTER

ASSUME CLAUSE

NONE

REGISTERS PRESERVED ACROSS CALLS

PRESERVED CLAUSE

SAVES ALL REGISTERS

REGISTER HISTORY CONTROL ON BASED
ASSIGNMENTS

REMEMBERS REGISTER
HISTORY SPOIL
CLAUSE

FORGETS HISTORY

STRING FACILITYEIS INSTRUCTIONALTRET FUNCTION

CONCAT/INSERT	MLR	TRUNCATION
XLATE	MVT	TRUNCATION
INDEX	NONE	NOT FOUND
INDEX1/INDEX1R	SCM/SCMR	NOT FOUND
INDEX2/INDEX2R	SCD/SCDR	NOT FOUND
SEARCH/SEARCHR	TCT/TCTR	NOT FOUND
BINCHAR/BINXCHAR	BTD	--
CHARBIN/XCHARBIN	DTB	--
EDITSTR	MVE	--
EDITCHAR/	MVNE	--

MONITOR SERVICES

MACRO DEFINITIONS

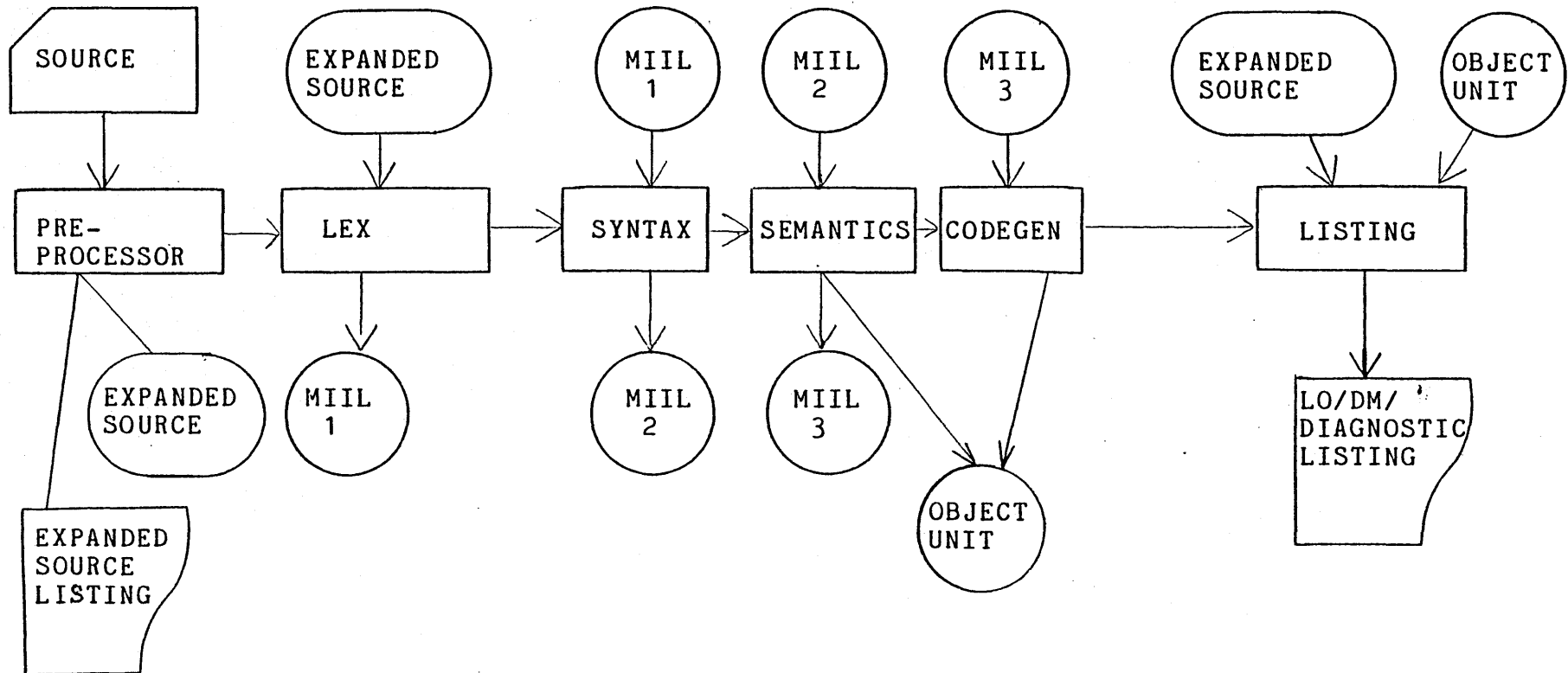
ENTRY DEFINITIONS

VECTOR/SIZEV FUNCTIONS

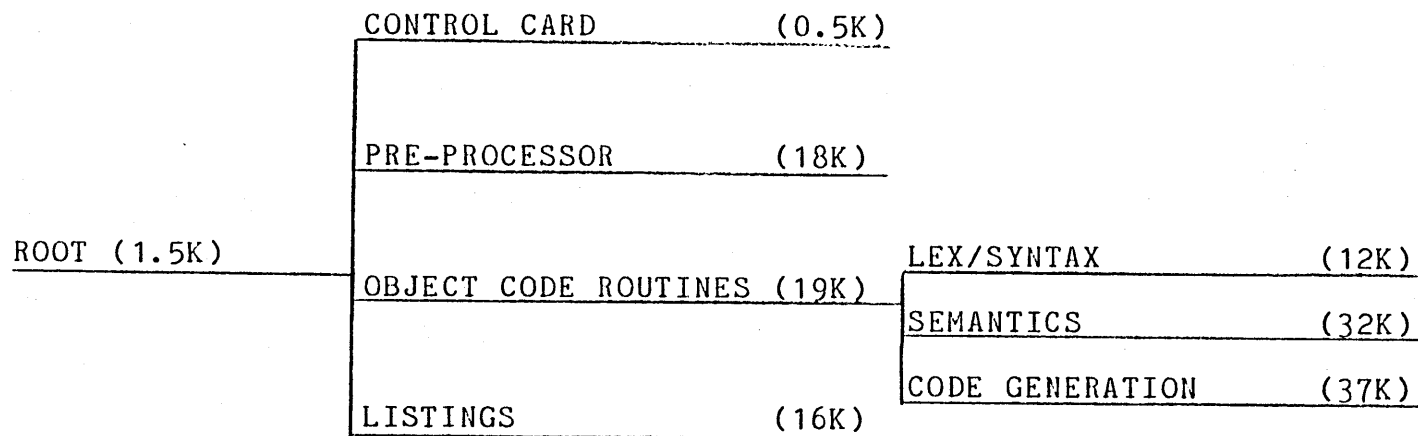
EXAMPLE:

```
%MAC FPT_TRUNC (FPTN=FPT=TRUNC, STCLAS=STATIC,DCB=NIL);  
    DCL 1 FPT STCLASS DALIGNED,  
        2 P,  
        3 V_BIT(72) DALIGNED INIT(VECTOR(FPTN.V)),  
        2 V DALIGNED,  
        3 DCB# UBIN(18) UNAL INIT(DCBNUM(DCB));  
  
%MEND;  
  
DCL M$TRUNC ENTRY(1) CONV(1, 14) ALTRET;  
%FPT_TRUNC (DCB=M$SI);  
CALL M$TRUNC(FPT_TRUNC) ALTRET(ERROR);
```

PL-6 EXECUTION/INPUT-OUTPUT FLOW



PL-6 OVERLAY STRUCTURE



BASIC

- o SHORT REVIEW OF BASIC LANGUAGE FEATURES
- o DESIGN GOALS
- o DESIGN TECHNIQUES
- o DESIGN OVERVIEW
- o WHERE ARE WE NOW?

BASIC DESIGN GOALS

o RELIABILITY

o MAINTAINABILITY

o EXTENDIBILITY

BASIC DESIGN TECHNIQUES

- o TOP-DOWN DESIGN
- o FUNCTIONAL FLOWS-FLOWCHARTS
- o MODULARITY
- o STRUCTURE

BASIC PROCESSOR DESIGN OVERVIEW

- o THREE MAJOR MODULES

- EDITOR

- COMPILER

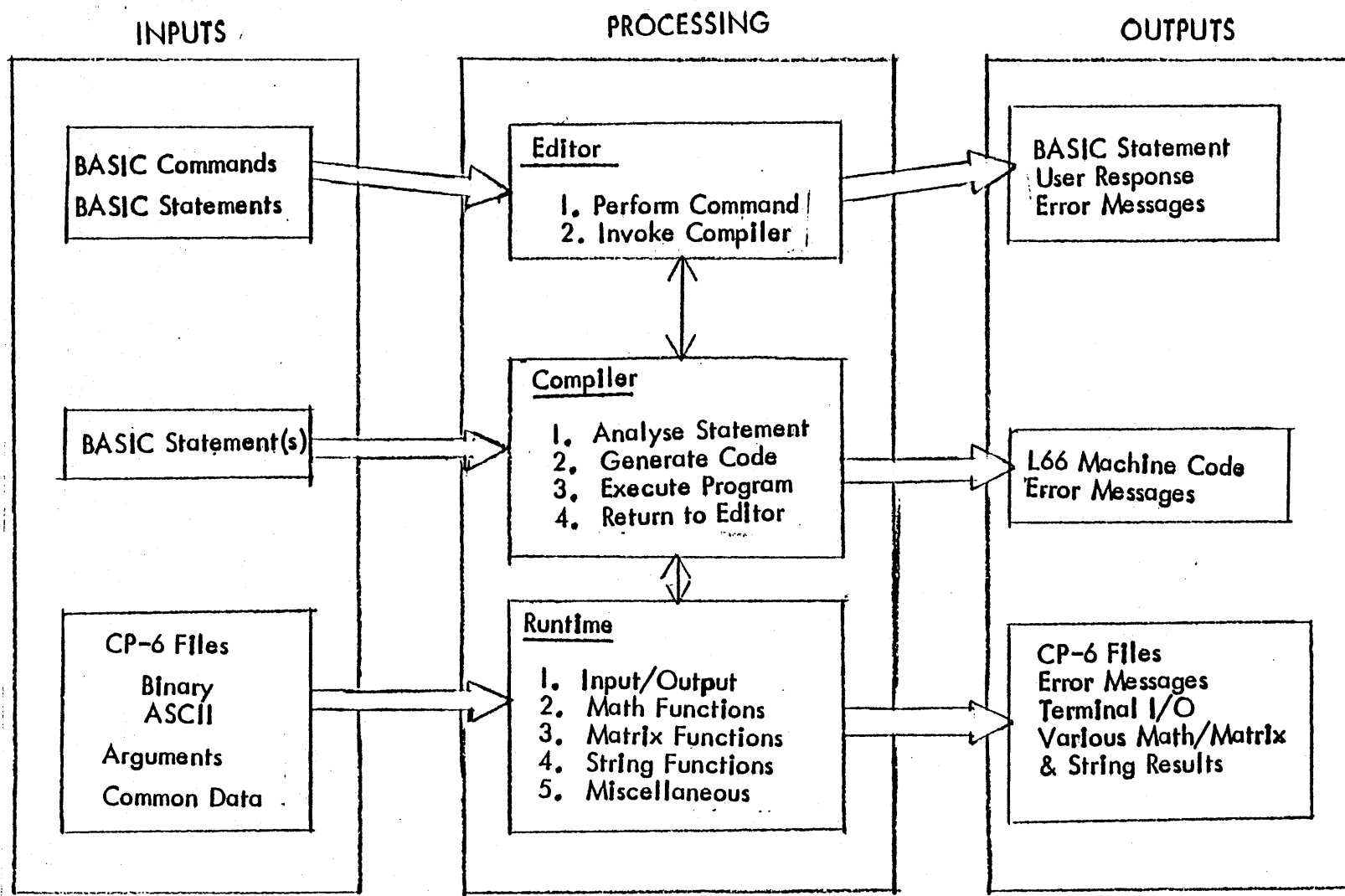
- RUNTIME

- o THREE MINOR MODULES

- GENERAL UTILITIES

- MEMORY MANAGEMENT

- CP-6 INTERFACE



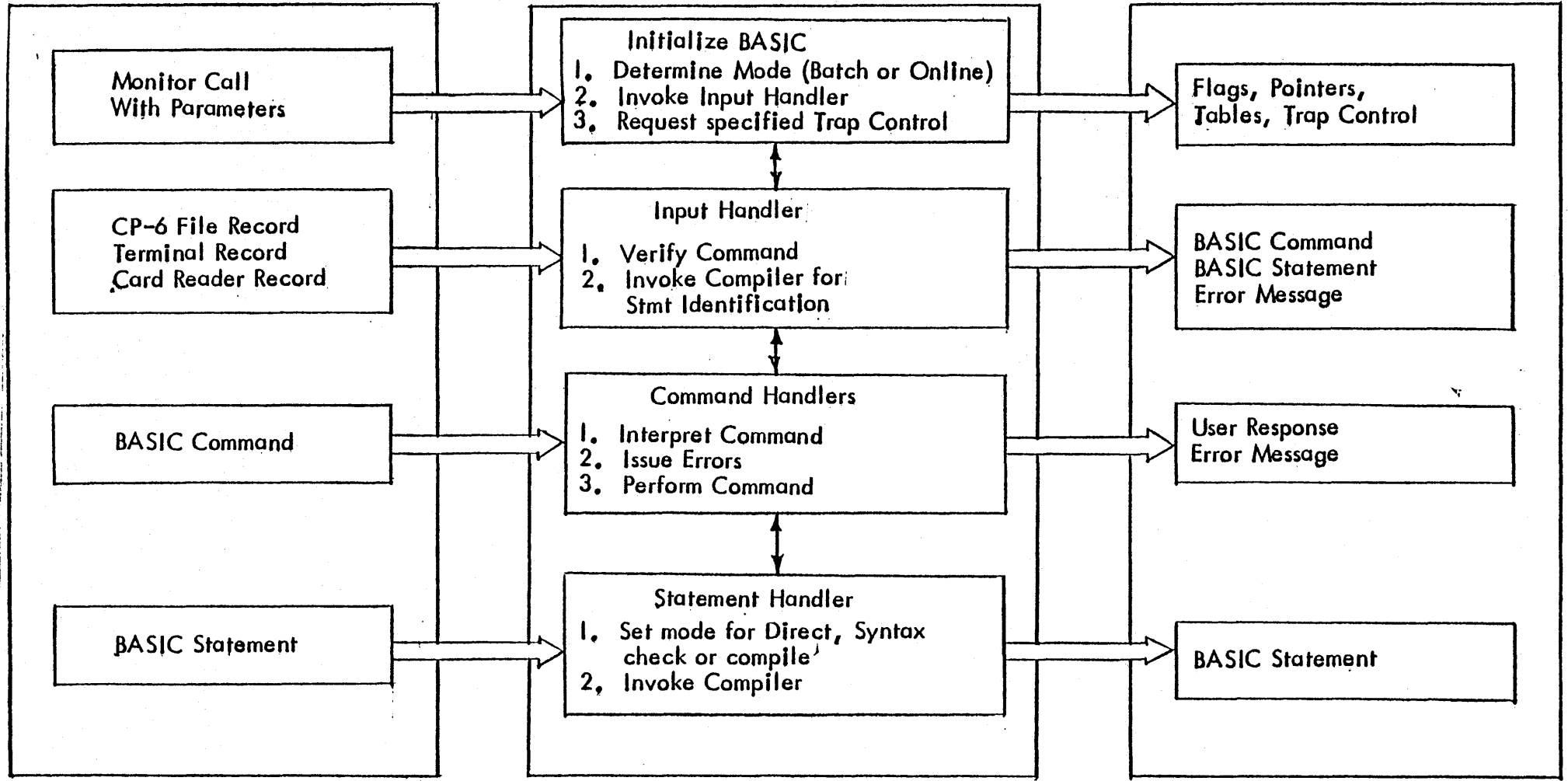
 Data Flow
  Control Flow

BASIC MAJOR MODULES - HIPO

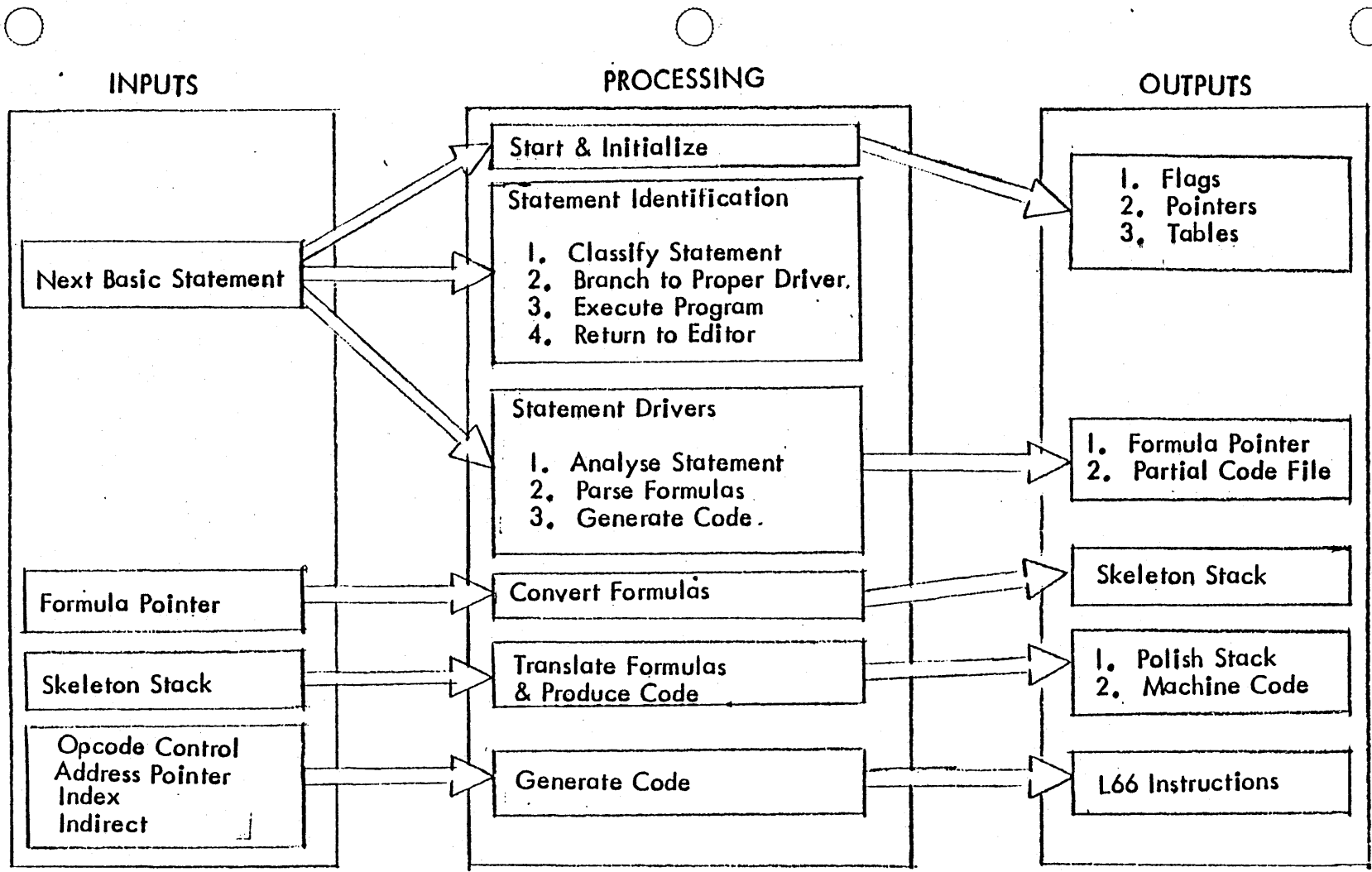
INPUTS

PROCESSING

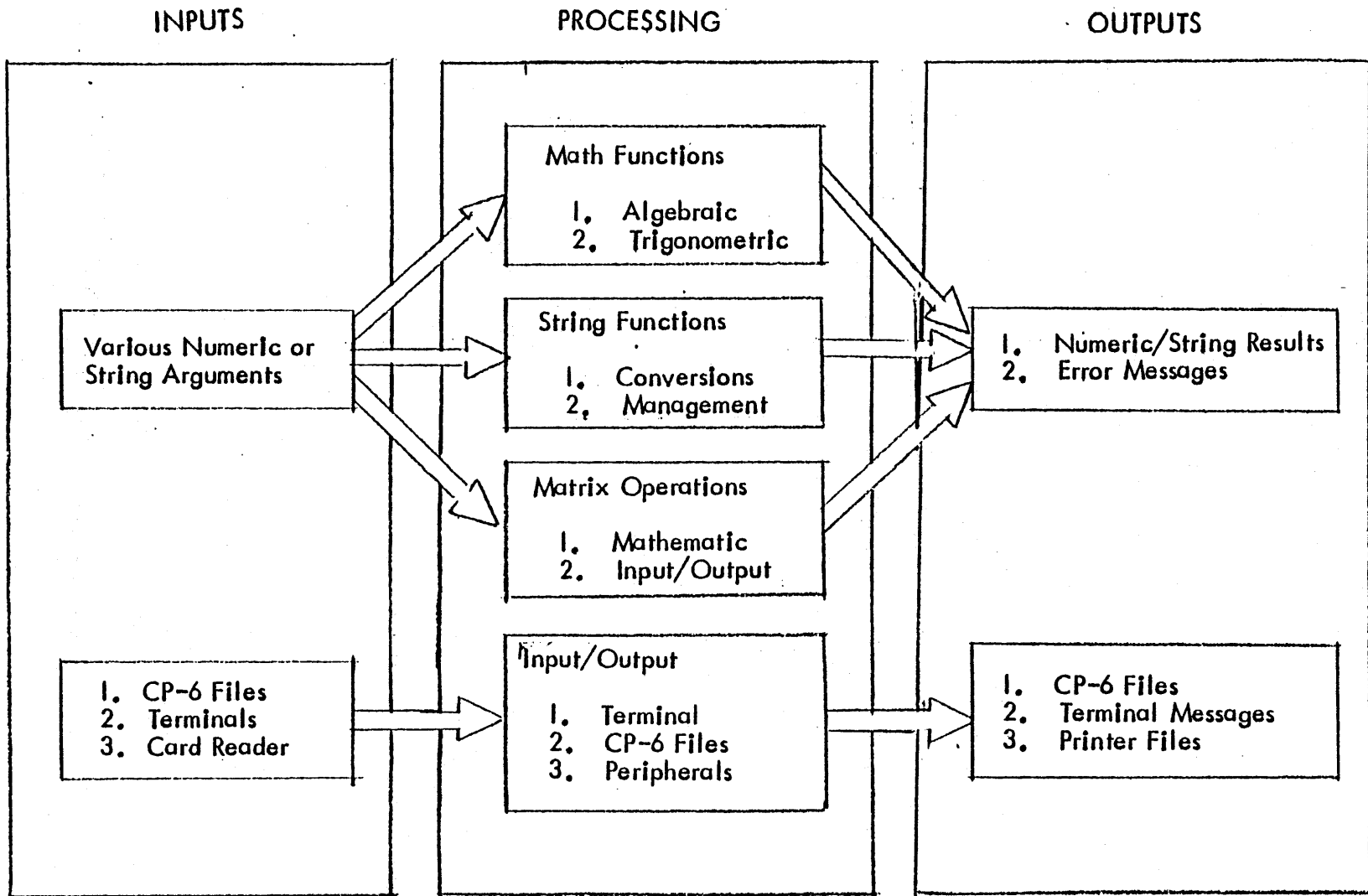
OUTPUTS



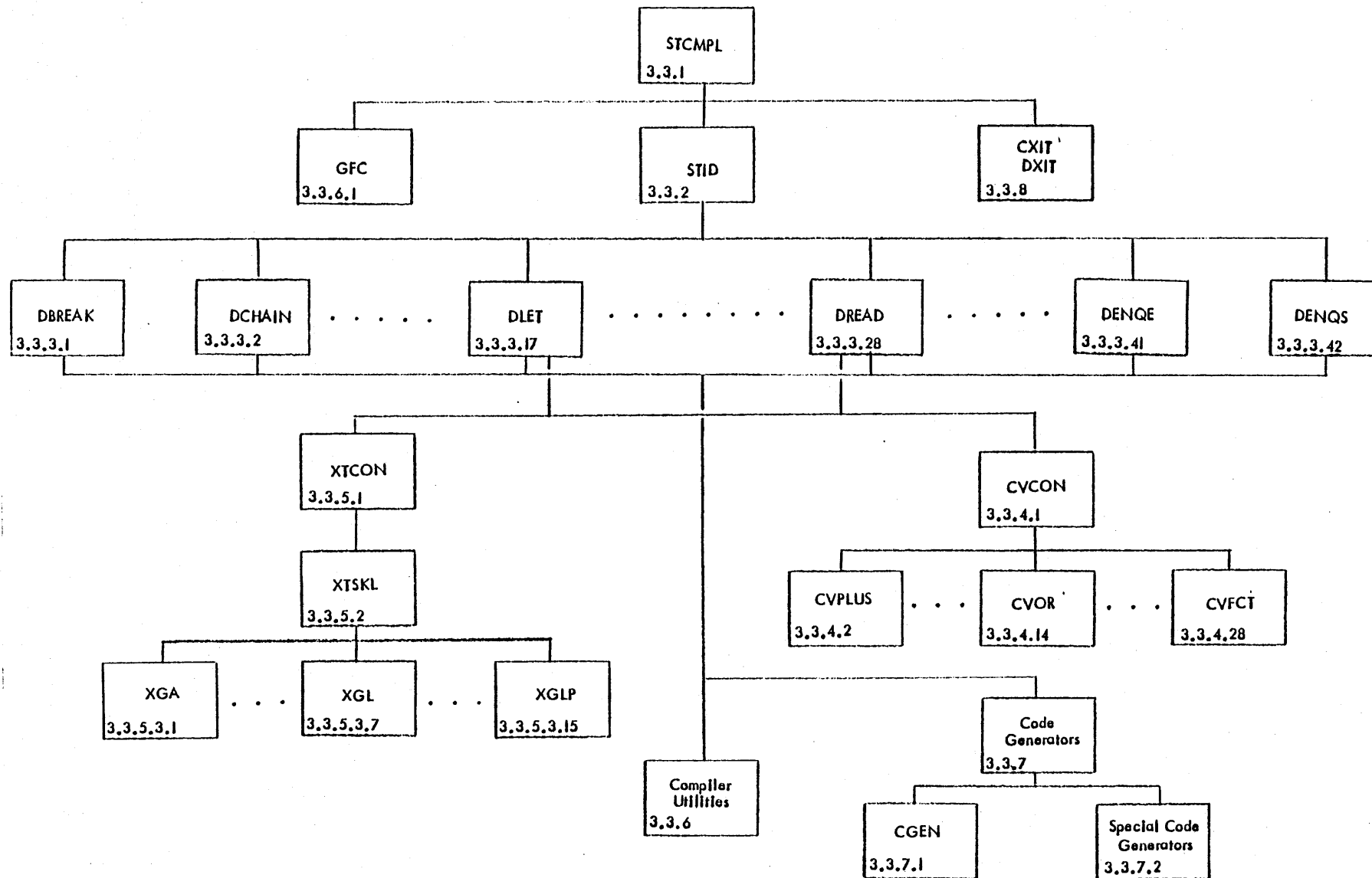
BASIC EDITOR - HIPO



BASIC COMPILER - HIPO



BASIC RUNTIME - HIPO



BASIC COMPILER - FUNCTIONAL HIERARCHY

WHERE ARE WE NOW?

- o EPS-1 RELEASED
- o PRELIMINARY DESIGN COMPLETE
- o DETAIL DESIGN ON SCHEDULE NEAR 20% COMPLETE
- o ONE MINOR MODULE CODED/CHECKED OUT

ANS FORTRAN FOR CP-6

LANGUAGE FEATURES

- o 77 ANS FORTRAN LANGUAGE
- o IBM FORTRAN LANGUAGE EXTENSIONS
- o XEROX CP-V ANS FORTRAN EXTENSIONS

IBM FORTRAN LANGUAGE EXTENSIONS

o DIRECT ACCESS I/O

- DEFINE FILE STATEMENT

- FIND STATEMENT

- INDEXED READS AND WRITES

o NAMELIST DRIVEN I/O

- NAMELIST STATEMENT (WITH N/L NAMES)

- READS AND WRITES WITH NAMELIST NAMES

o ALTERNATE RETURNS FROM SUBPROGRAMS

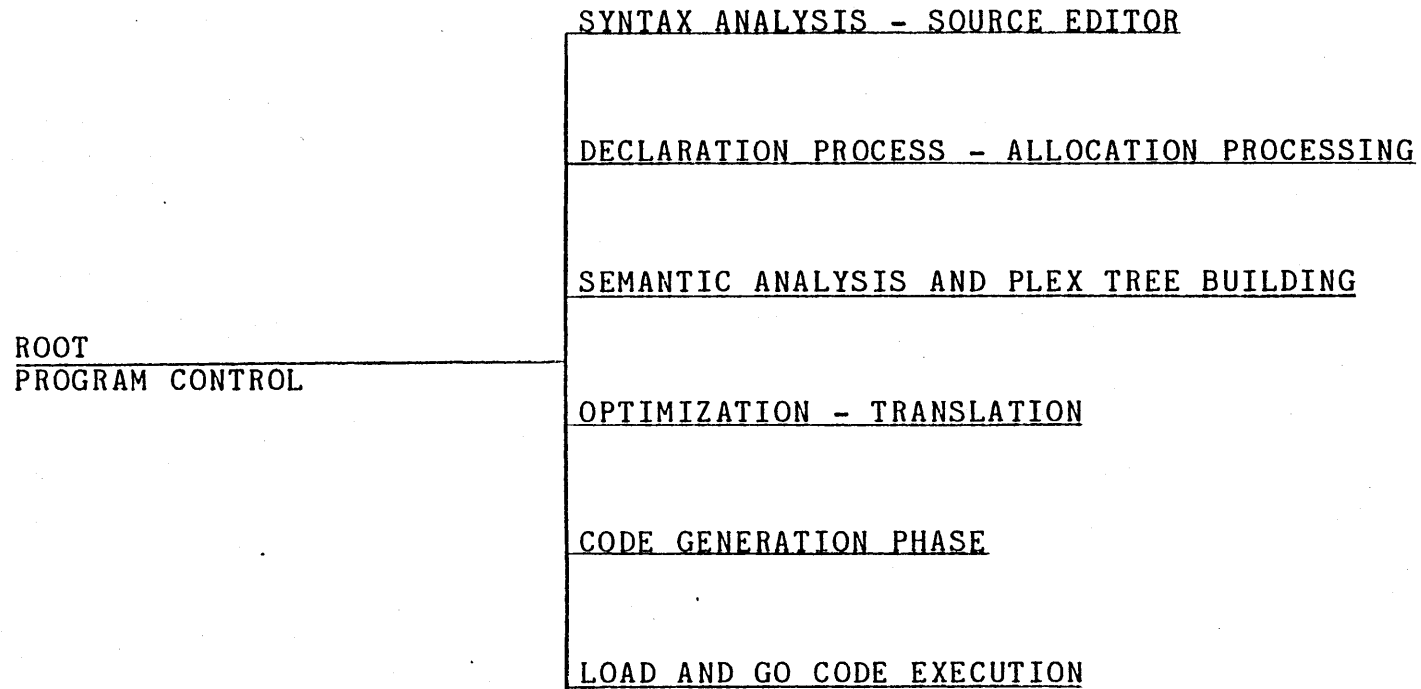
XEROX CP-V ANS FORTRAN LANGUAGE EXTENSIONS

- o ABNORMAL STATEMENT
- o CONSTANT STATEMENT
- o DOUBLE COMPLEX DATA TYPE
- o GLOBAL STATEMENT
- o INCLUDE STATEMENT
- o INPUT STATEMENT
 - FREE FORM
 - WITH LIST
- o NAMELIST STATEMENT
 - WITHOUT ID LIST
 - WITH ID LIST, WITHOUT N/L NAMES
- o NORMAL STATEMENT
- o OUTPUT STATEMENT
- o PARAMETER STATEMENT (INIVAC FORM)
- o READ/WRITE DISK STATEMENTS
- o VIRTUAL STATEMENT
- o ENCODE STATEMENT
- o DECODE STATEMENT

ANS FORTRAN COMPILER DESCRIPTION

- o DESIGN BASED OF CP-V ANS FORTRAN
 - SIMILAR LANGUAGE SET
 - BLOCK OPTIMIZATION
 - BASIC DESIGN IS TRANSPORTABLE
 - FAMILIARITY OF DESIGN
- o MULTIPASS COMPILER
- o OVERLAYED PROCESSOR

OVERLAY STRUCTURE OF THE
CP-6 ANS FORTRAN COMPILER



COMPILER OPTIMIZATION

- o BLOCK OPTIMIZATION
 - o CONGRUENT SUBSCRIPTS EXPRESSIONS
 - o CONGRUENT ARITHMETIC EXPRESSIONS
 - o ALTERNATE RECOGNITION
 - VARIABLES KNOWN AS EXPRESSIONS
 - VARIABLES KNOWN AS CONSTANTS
 - EXPRESSIONS KNOWN AS VARIABLES
 - o COMPILE TIME CONSTANT ARITHMETIC
 - ADD, SUB, MULTIPLE, DIVIDE, EXPONENTIATION
 - o CONGRUENT ADDRESS REGISTER USAGE

COMPILER OPTIMIZATION (CONT.)

o ALTERNATE RECOGNITION

- VARIABLES KNOWN AS EXPRESSIONS

ASSOCIATION OF A VARIABLE WITH AN EXPRESSION:

A = B*C
.
.
.
D = A [.,,D=RESULT OF B*C]

- VARIABLES KNOWN AS CONSTANTS

ASSOCIATION OF A VARIABLE WITH A CONSTANT AT COMPILE TIME:

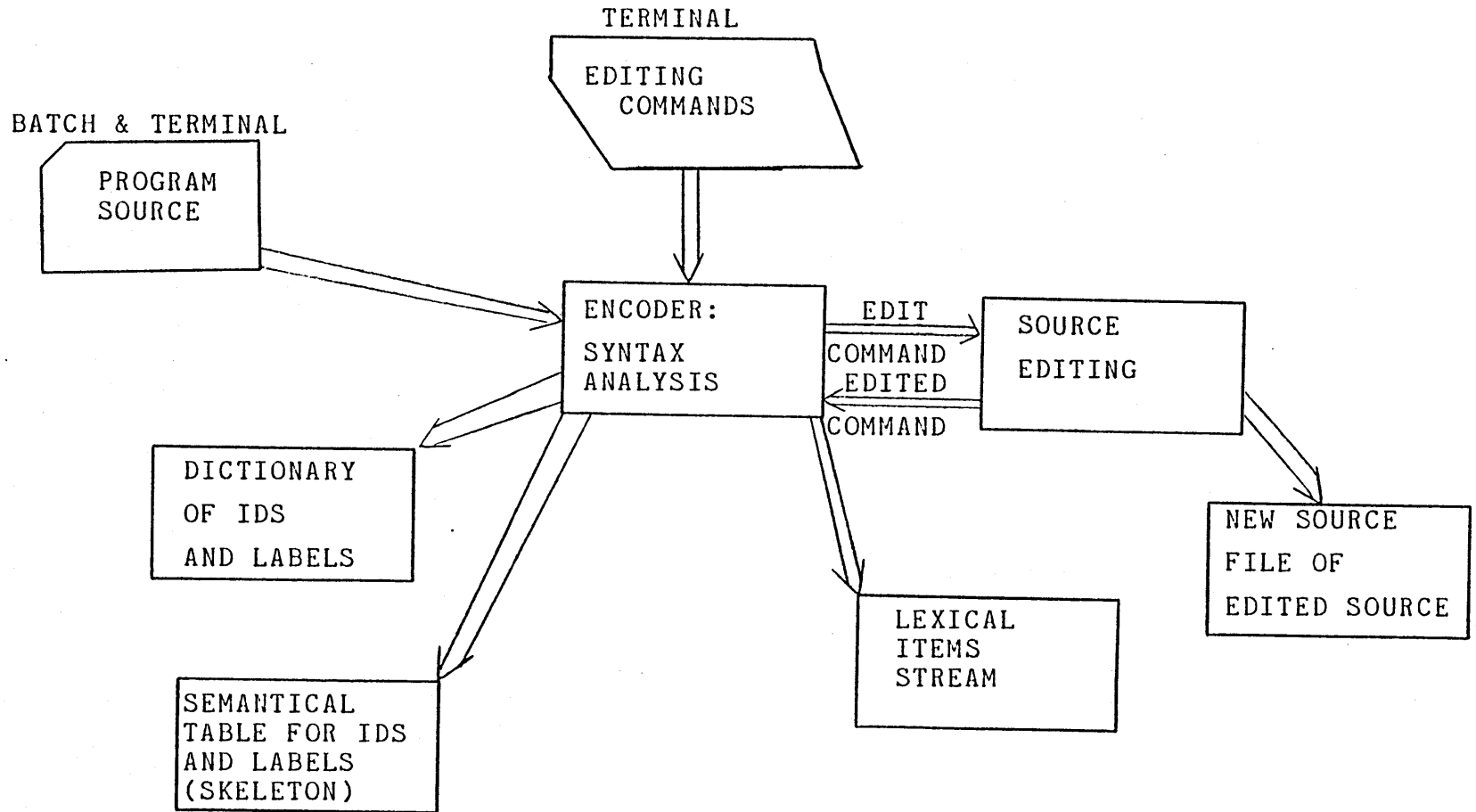
A = 10.0
.
.
.
D = A*4 [.,,D=40.0]

- EXPRESSIONS KNOWN AS VARIABLES

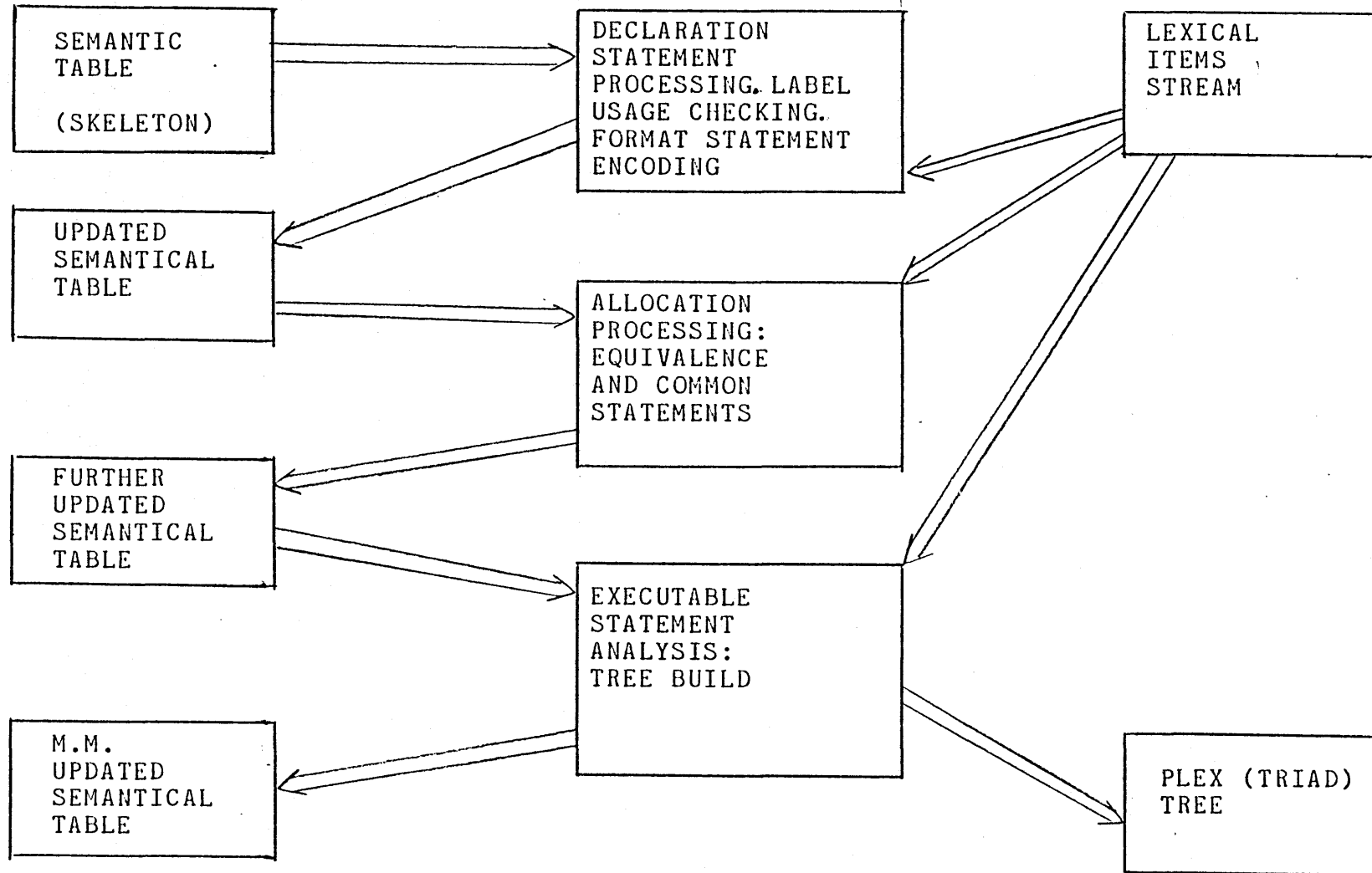
ASSOCIATION OF AN EXPRESSION WITH A VARIABLE:

A = B*C
.
.
.
D = B*C [.,,D=A]

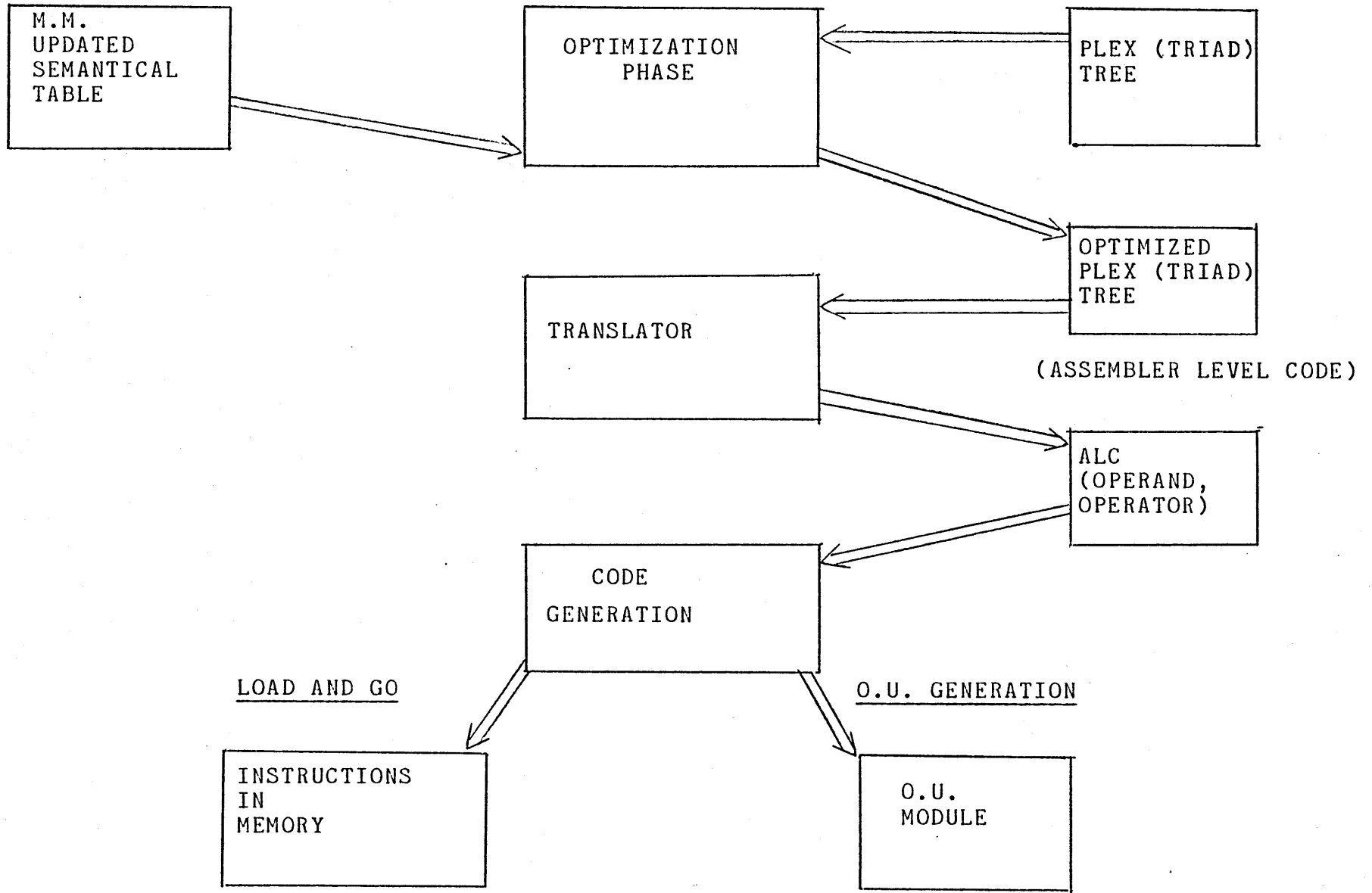
FRONT END PROCESSOR



SEMANTICAL PHASE PROCESSING



CODE GENERATION PROCESSING



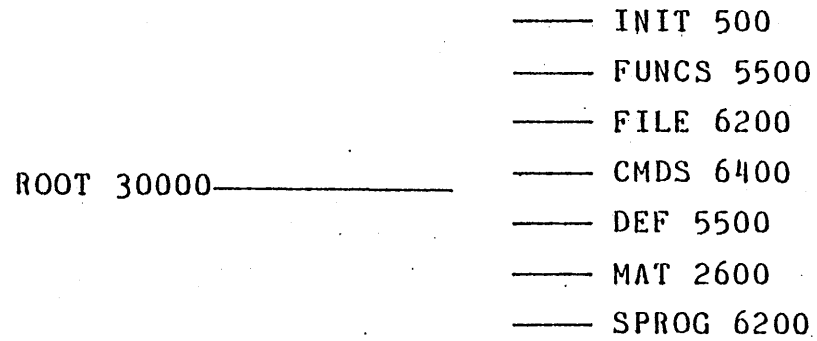
COMPETITIVE ADVANTAGES OF CP-6 APL

1. A rich set of functions exceeding those of competitive products.
2. An environment which admits expansion easily.
3. Relatively efficient operations on large data aggregates.
4. Workspace capacity significantly greater than that of competitive products.

DIFFERENCES BETWEEN CP-V APL & CP-6 APL

1. Hardware differences
2. Implementation vehicle differences
3. Operating system differences
4. New features
5. Delayed features

PROCESSOR STRUCTURE



<u>Name</u>	<u>Includes</u>
ROOT	Principal monitor interface, input, output, workspace management, codestringer, codestring executor, operator execution drivers
INIT	Processor initialization
FUNCS	Certain function evaluators, formatted output
FILE	File I/O and shared variables
CMDS	All commands
DEF	Function definition & error reporting
MAT	The domino operator
SPROG	System programmer functions: workspace management, text editing, and canonical representation

Sizes above are in 36 bit words and have a tolerance of $\pm 20\%$, for procedure only.

WORKSPACE DATA

- STATIC DATA - 3 pages in the ROOT and 1 page in most overlays
- AUTOMATIC DATA - 1 page
- DYNAMIC SEGMENT 1 - The symbol table and the data blocks
- DYNAMIC SEGMENT 2 - The execution stack

INTERNALLY PERCEIVED RISKS

1. THE IMPLEMENTATION VEHICLE, PL6
2. THE CHECKOUT ENVIRONMENT AND LIMITED CHECKOUT TIME
3. PERSONNEL

Competitive Advantages of CP-6 APL

- I. A rich set of functions exceeding those of many competitive products.
 - A. File I/O, blind I/O
 - 1) The full CP-6 file system facilities are available to the APL user. This includes shared update as well as enqueue/dequeue to control access in the shared mode.
 - 2) Both translated and untranslated(blind) I/O is provided.
 - B. Several unique commands are provided
 - 1))SEAL allows the installation to provide proprietary APL functions for execution only. Also, individual functions in a workspace may be locked to likewise prevent inspection of the algorithm.
 - 2))OBSERVE provides the ability to inspect the evaluations undertaken for execution of any line in the minutest detail.
 - 3))CATCH allows the user to intercept all assignments to a particular variable to facilitate debugging.
 - C. System commands may be executed via the unary epsilon operator.
 - D. Sidetracking of errors and breaks is provided.
 - E. The system programmer functions
 - 1) Canonical representation to change function definition to and from text.
 - 2) Workspace management to investigate attributes of workspace constituents.
 - 3) Text editing to provide text index, search and replacement and comparison.
- II. An environment which admits expansion easily.
 - A. Both batch and on-line operations are accomodated.
 - B. A very flexible terminal interface mechanism admits considerable variety.
 - 1) Files are included.
 - 2) Terminals without APL characters are accomodated.
 - C. The PL6 software factory has several beneficial aspects.
 - 1) Listings can contain much more documentary content.
 - 2) The language promotes the use of "structured" constructs.
 - D. The modularization of the CP-6 APL product will separate data, procedure and monitor interface.
 - E. The commentary in the source for the processor will contain all the information to produce the technical and data base manuals via an automated process.

Competitive Advantages of CP-6 APL - Contd.

- III. Relatively efficient operations on large data aggregates.
 - A. Code is "compiled" for inner loops.
 - B. Use is made of the common library at all radix translation and elementary function evaluation times.
 - C. Codestring is used to facilitate execution.

- IV. Workspace capacity is significantly greater than that of the competition.
 - A. A large virtual memory is provided with little operating system preemption.
 - B. Multiple segment dynamic memory will be utilized.
 - C. Typically CP-6 APL will provide 256K bytes vs. 32K bytes.
 - D. CP-6 APL will provide overflow symbol table processing beyond the "hashed" symbol table capacity.

Differences between CP-V APL & CP-6 APL

It should be emphasized that the CP-V APL design is, to a large extent, being copied over into the APL processor for CP-6. CP-V APL is a field-proven, highly competitive product. However, there are differences which should be exposed during review.

- I. Differences caused by change of hardware
 - A. Word and byte size, but fortunately 36>32 and 9>8.
 - B. Floating point range is reduced which may require scaling of some processes.
 - C. The instruction set and the CPU organization are quite different.
 - 1) The compiled code for interpretation of operators is new.
 - 2) The use of PL6 minimizes much of this effect.
 - D. Traps are different.
- II. Differences caused by change of primary implementation vehicle.
 - A. The "structured" and documentary effects have been reviewed earlier.
 - B. PL6 generated code will be far from that which would be generated to provide minimum procedure residency as was the intent in the CP-V product.
 - C. The ability to produce technical documentation in an automated manner was mentioned earlier.
 - D. As an emphasis note, all of the above noted differences tend to promote maintainability.
- III. Differences caused by change of operating system.
 - A. Some additional effects of hardware change are reflected in the operating system interface.
 - 1) Exceptional conditions vary.
 - 2) NSA hardware is different from the Sigma map.
 - 3) File granule size is different.
 - B. Many of the facilities of CP-6 expand upon those which had been in CP-V.
 - 1) File open
 - 2) File attributes
 - 3) "Privilege"
 - C. Dynamic memory management is particularly different.

Differences between CP-V APL & CP-6 APL - Contd.

IV. New features have been added to CP-6 APL.

A. Distinguished names

- 1) Had been available as I-BEAM and T-BAR functions.
- 2) Must be made capable of localization.

B. Shared variables are a new CP-6 APL facility which will be provided by utilizing field-proven CP-V facilities.

- 1) Shared update files
- 2) Enqueue and dequeue

V. Delayed features - Two features of the CP-V APL processor have been eliminated from the initial release of the CP-6 product.

A. Graphics functions

B. The interface to the data base manager, IDS-II.

Processor Structure

I. ROOT

- A. Principal monitor interfaces
 - 1) Memory management
 - 2) SI, LO, DO management
 - 3) Exceptional condition handlers
- B. Input
 - 1) Mnemonic translation
 - 2) Visual fidelity
- C. Output
 - 1) Mnemonic translation
 - 2) Line formation
- D. Workspace management
 - 1) Symbol table
 - 2) Data blocks
 - 3) Execution stack
- E. Command recognition
 - 1) Initial processing of commands
 - 2) Root interface to error processes
- F. Codestring
 - 1) Translate to/from codestring
 - 2) Codestring execution
- G. Operator execution drivers
 - 1) Simple operators
 - 2) Mixed and composite operators
 - 3) Index processes
 - 4) Intrinsic functions

II. INIT - Once per user processor invocation processes.

III. FUNCS

- A. Certain function evaluators
- B. Formatted output

Processor Structure - Contd.

IV. FILE

- A. File and transparent I/O
- B. Shared variables

V. CMDS - All commands starting with)

VI. FUNDEF

- A. Function definition
- B. Error management

VII. MAT - Matrix inversion

VIII. SPROG - System programmer functions; namely, canonical representation, workspace management, and text editing.

Workspace Data

- I. Static data - 3 pages for the ROOT and 1 page in several overlays. This is the main internal working space for the workspace containing the fixed size data which is, in general, not displayable.
- II. Automatic data - 1 page is required for certain local data which is temporary in nature.
- III. Dynamic segment 1
 - A. Most of the displayable and copyable fixed length workspace data.
 - B. The symbol table.
 - C. The data blocks.
- IV. Dynamic segment 2
 - A. The fixed length state control data.
 - B. The execution stack.

Internally Perceived Risks

- I. The implementation vehicle, PL6
 - A. Effects on processor and workspace size.
 - B. Effects on speed of various processes.

- II. Personnel - It is an exceptional circumstance which could cause this to be surfaced as a risk, but the health of the only team member from the original CP-V team is of concern.

- III. Checkout
 - A. The novelty of CP6 is bound to cause problems, not the least of which is concern about timely availability.
 - B. The current schedule does not admit sufficient time between earliest assumed availability of checkout hardware/software and delivery time to accommodate even the most optimistic checkout schedule.

CP-6 TEXT

SHARED PROCESSOR UNDER CP-6

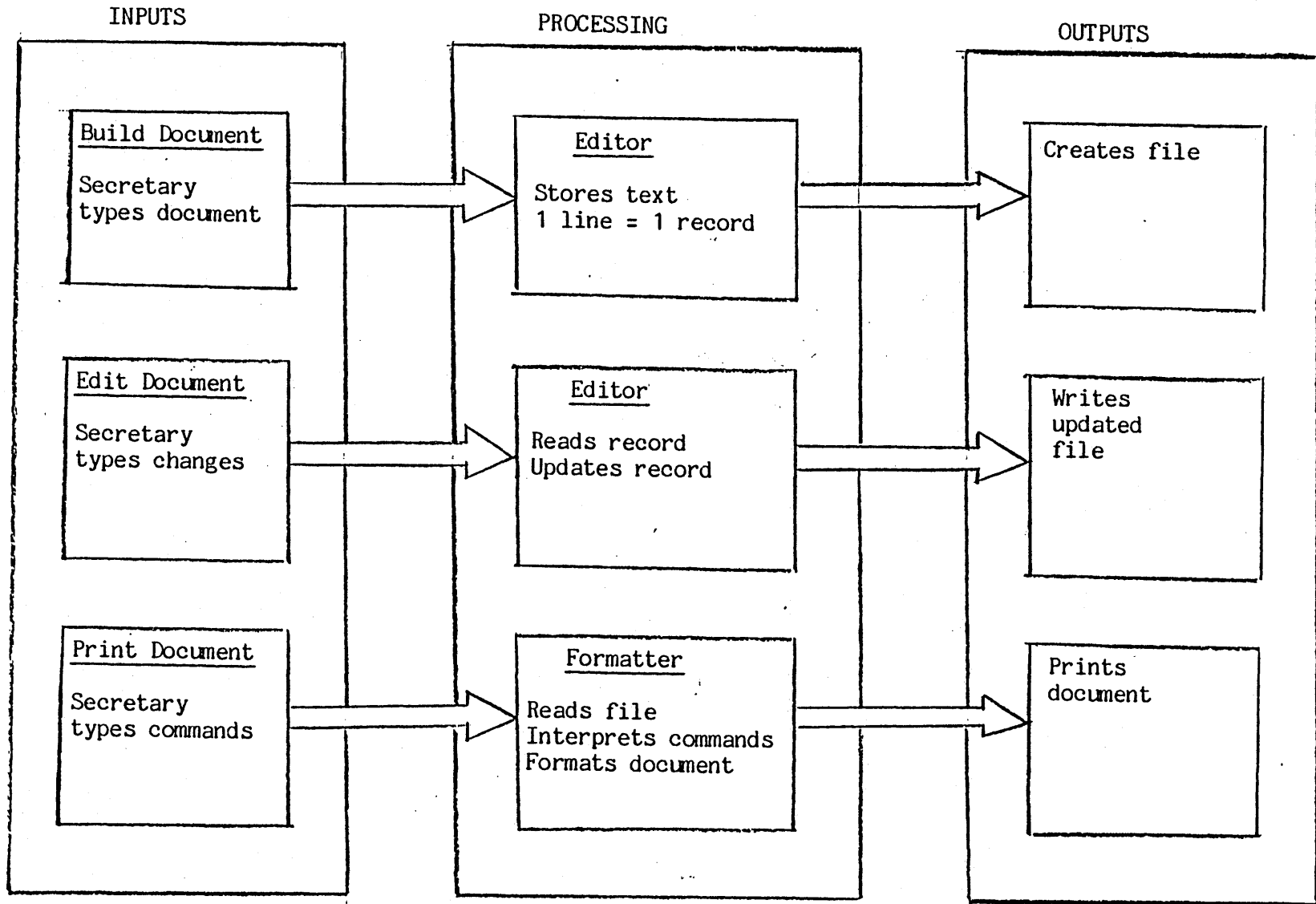
FAST DOCUMENT CREATION

AUTOMATIC FORMATTING CAPABILITIES

NAME-AND-ADDRESS FILES

ON-LINE AND BATCH OPERATION

CP-6 TEXT



CP-6 LANGUAGE PROCESSORS

SUMMARY

DESIGN PHASE - EXCELLENT VISIBILITY

- o BUILDING UPON PROVEN CP-V DESIGN

CODING PHASE - HIGH CONFIDENCE

CHECKOUT PHASE - LOTS OF EXPOSURE

CP-6 LANGUAGE PROCESSORS

RISKS

CONCURRENT DEVELOPMENT OF OPERATING SYSTEM

DEBUGGING - WHERE IS ERROR?

CONTENTION FOR COMPUTER TIME

COMPUTER RESOURCES

ADEQUATE COMPUTER TIME

SUFFICIENT TERMINALS

PL-6 PERFORMANCE

SIZE

SPEED

STAFFING

CP-6 PRELIMINARY DESIGN REVIEW
PHOENIX SUPPLIED SOFTWARE

- PRODUCTS
 - COBOL-74
 - SORT/MERGE
- IMPLEMENTATION TOOLS
 - PL/I
 - GMAP6
- TOPICS
 - PRODUCT STRUCTURE
 - COMPATIBILITY
 - PERFORMANCE
 - TESTING
 - SCHEDULE
 - MAINTENANCE STRATEGY

PRODUCT STRUCTURE

- BASE DESIGN
 - EXISTING GCOS PRODUCTS
- DEVELOPMENT STRATEGY
 - MAXIMIZE COMMON SOURCE BASE
 - ISOLATE HOST SYSTEM INTERFACES
 - ADAPT TO CP-6 I/O
- EXCEPTION - SORT/MERGE
 - MAJOR REDEVELOPMENT
 - PERFORMANCE SENSITIVITY
 - INTIMATE USE OF MONITOR SERVICES

COMPATIBILITY

- WITH CP-V
 - MAXIMIZE CONSISTENT WITH OVERALL CP-6 GOALS
- WITH GCOS
 - HIGHLY DESIRABLE TO FACILITATE:
 - COMMON PROCESSORS - COSTS
 - ULTIMATE MIGRATION
- WITH IBM
 - NOT ADDRESSED IN WORK LEADING TO RELEASE 1

PERFORMANCE

- EXPECTATIONS
 - EXISTING DESIGN
 - EXISTING BASE PROCESSOR

- EXCEPTIONS
 - MONITOR I/O
 - COMMON I/O ADAPTATION
 - PURE PROCEDURE

TESTING

- BY DEVELOPMENT SHOP LEADING TO INITIAL RELEASE AS OPPOSED TO CURRENT GCOS PRACTICE OF USING SEPARATE AND FORMALIZED TEST ORGANIZATION

MAINTENANCE STRATEGY

- CP-6 FIELD SUPPORT GROUP
 - SCREEN STARS
 - RESPONSE DATABASE
 - INTERFACES WITH THE FIELD AND THE DEVELOPMENT GROUPS
- PHOENIX DEVELOPMENT GROUP
 - INTERFACES WITH CP-6 FIELD SUPPORT GROUP
 - COMMON PROCESSOR FIXES TESTED ON GCOS
 - HOST INTERFACE FIXES TESTED ON CP-6

CP-6 PRELIMINARY DESIGN REVIEW

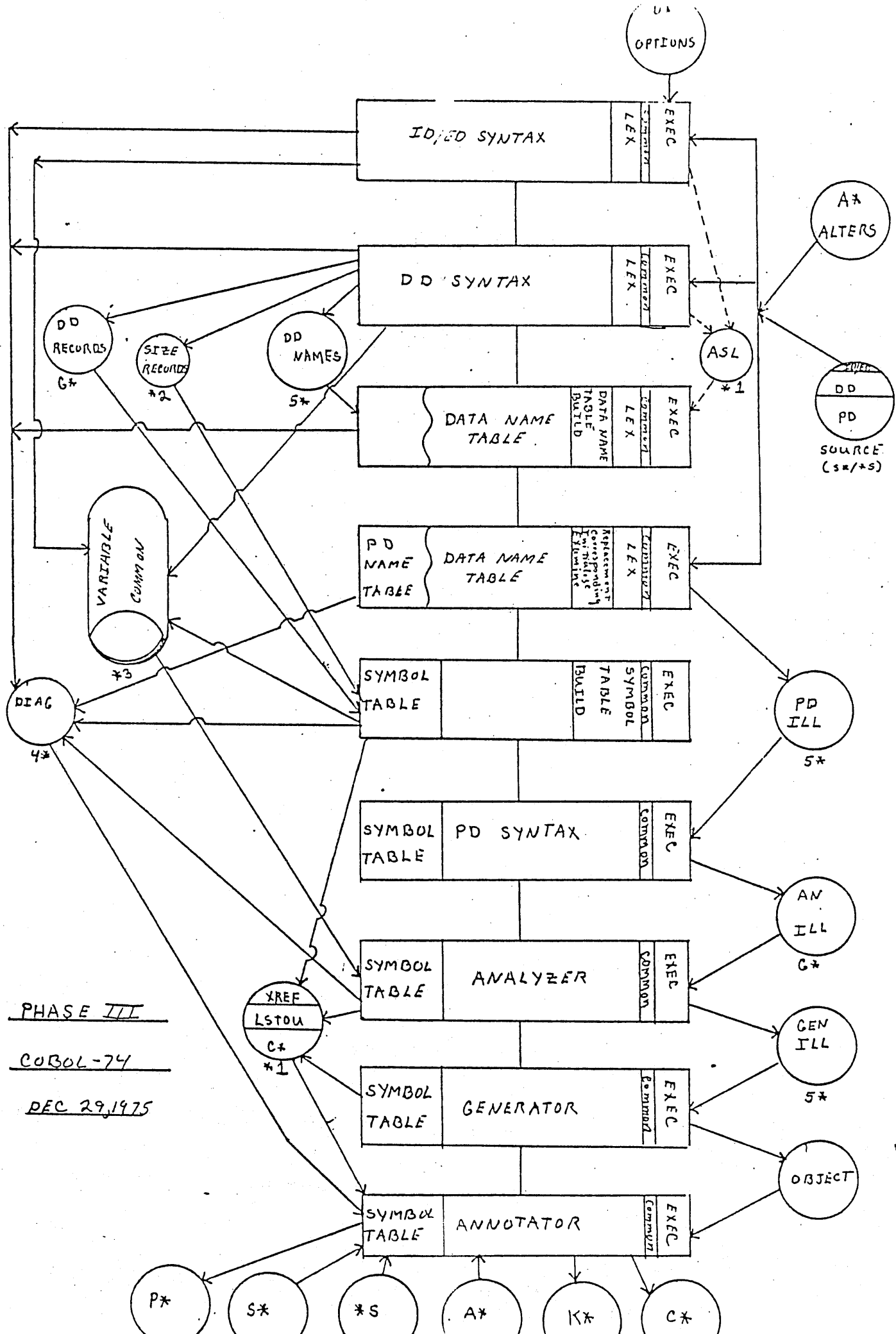
COBOL-74

- STRUCTURE
- COMPATIBILITY
 - CP-5
 - GCOS-III
- PERFORMANCE
- TESTING
- SCHEDULE & MILESTONES
- MAINTENANCE STRATEGY

PL/I

- STRUCTURE
- SCHEDULE

STRUCTURE

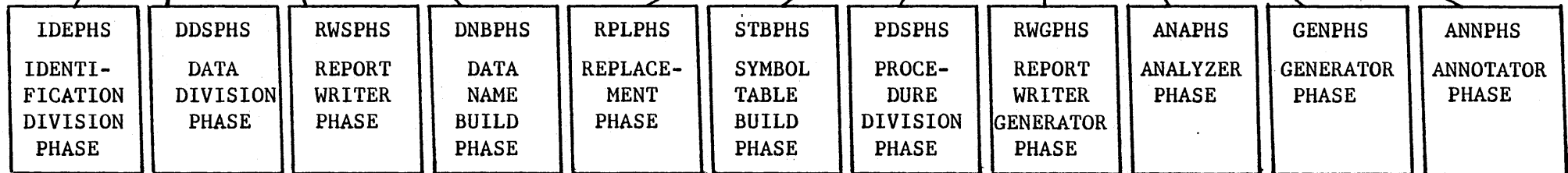


PHASE III
 COBOL-74
 DEC 29, 1975

COBOL 74
EXEC AND
SUPPORT
ROUTINES

LEXEXC

MINEXC



SOURCE
PROCESSOR

- MAP
- LSTIN
- ALTNO
- RLSCOMP

LSTOU
PROCESSOR

- LSTOU
- DECK
- XREF

COBOL-74 COMPILER EXECUTIVE

- CONTROLS FLOW OF COMPILER FROM PHASE TO PHASE
- PROVIDES ABORT RECOVERY AND WRAP-UP
- PROVIDES I/O INTERFACE WITH CP-6
- PROVIDES CP-6 MONITOR INTERFACES
- PROVIDES OTHER MISCELLANEOUS SUPPORT FUNCTIONS

COBOL-74 GENERATOR PHASE

TRANSLATION OF INTERNAL LANGUAGE LIST (ILL) AND
EXPRESSION TRIAD TABLE (ETT) ITEMS INTO OBJECT
CODE.

- INPUT
 - INTERMEDIATE FILE: ILL FROM ANALYZER.
 - CORE RESIDENT TABLES: ETT, ST, LTS, SYSSM, ETC.
- OUTPUT
 - CP-6 OBJECT FILE
 - *1 FILE (LSTOU INFORMATION)

COBOL-74 ANNOTATOR

- PRODUCES SOURCE LISTING OF COBOL COMPILER
- PRODUCES SYMBOLIC LISTING OF GENERATED OBJECT CODE
- PRODUCES SYMBOL CROSS REFERENCE REPORT
- PRODUCES STORAGE MAP
- PRODUCES OBJECT DEBUG SCHEMA
- PRODUCES COMPILER STATISTICS

COMPATIBILITY

- CP-5
 - FLOATING POINT DATA TYPES
 - MULTIPLE ENTRY POINTS
 - LABEL PROCESSING USE PROCEDURES
 - RUNTIME OPTIONS
 - 30 CHARACTER ID-NAMES

- GCOS-III
 - INCORPORATION OF CP-5 ENHANCEMENTS
 - SINGLE COMPILER SOURCE

PERFORMANCE

- COMPILER
 - SIZE 42K BASE
 - SPEED 3000 LPM
 - NOT SHARED

- OBJECT
 - EFFICIENCY
 - SIZE
 - SHAREABLE

LANGUAGE FEATURE TESTING

- USE EXISTING TEST PROCEDURES
- TEST IN LEVEL 66 ENVIRONMENT

CP-6 OBJECT TESTING

- USE EXISTING TEST PROGRAMS
- COMPILE ON LEVEL 66; EXECUTE ON CP-6
- REQUIRES EXTENSIVE MODIFICATION OF TEST PROCEDURES

COMPILER TESTING

- USE EXISTING TEST PROGRAMS, TRANSPORTED TO CP-6
IN SOURCE FORM
- USE EXISTING (SELF-CHECKING) TEST PROCEDURES
MODIFIED TO EXECUTE ON CP-6

SCHEDULE & MILESTONES

	<u>START</u>	<u>DATE</u> <u>COMPLETE</u>
● FUNCTIONAL ENHANCEMENTS	732	834
FLOATING POINT DATA TYPE		
MULTIPLE ENTRY POINTS		
LABEL PROCESSING USE PROCEDURES		
● CP-6 OBJECT I/O	733	827
● CP-6 RUNTIME LIBRARY	733	826
● CP-6 OBJECT UNIT	730	830
● DEBUG SCHEMA	728	832
● COMPILE TIME MODIFICATIONS	730	830
● COMPILER INTEGRATION	826	839
● SYSTEM TEST	840	913

MAINTENANCE STRATEGY

INTERFACE - LADC

- RESPONSES/VERIFICATION
- TESTING ON GCOS-III (LANGUAGE)
- TESTING ON CP-6
- CORRECTIONS ISSUED

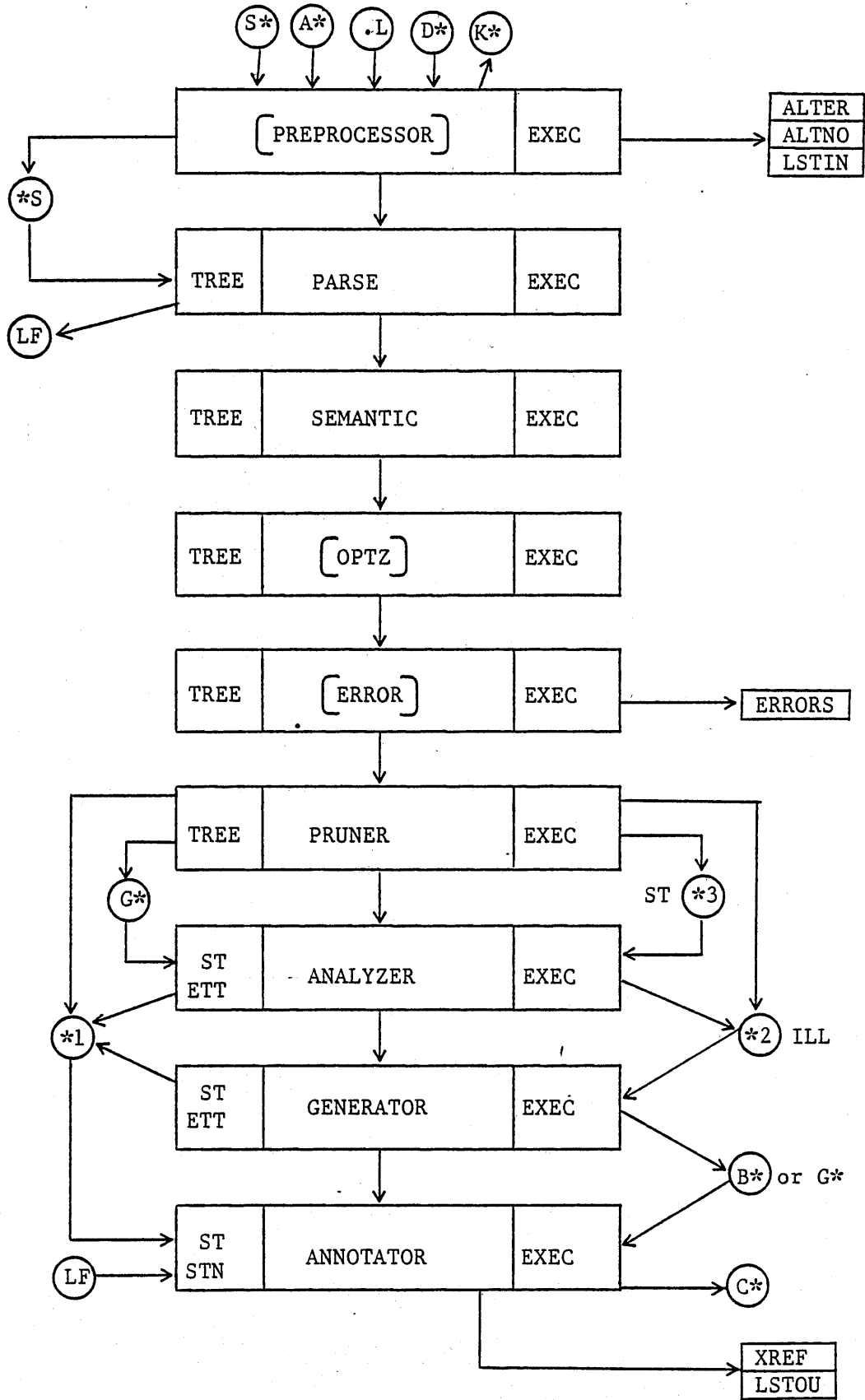
OBJECTIVE - SINGLE COMPILER SOURCE

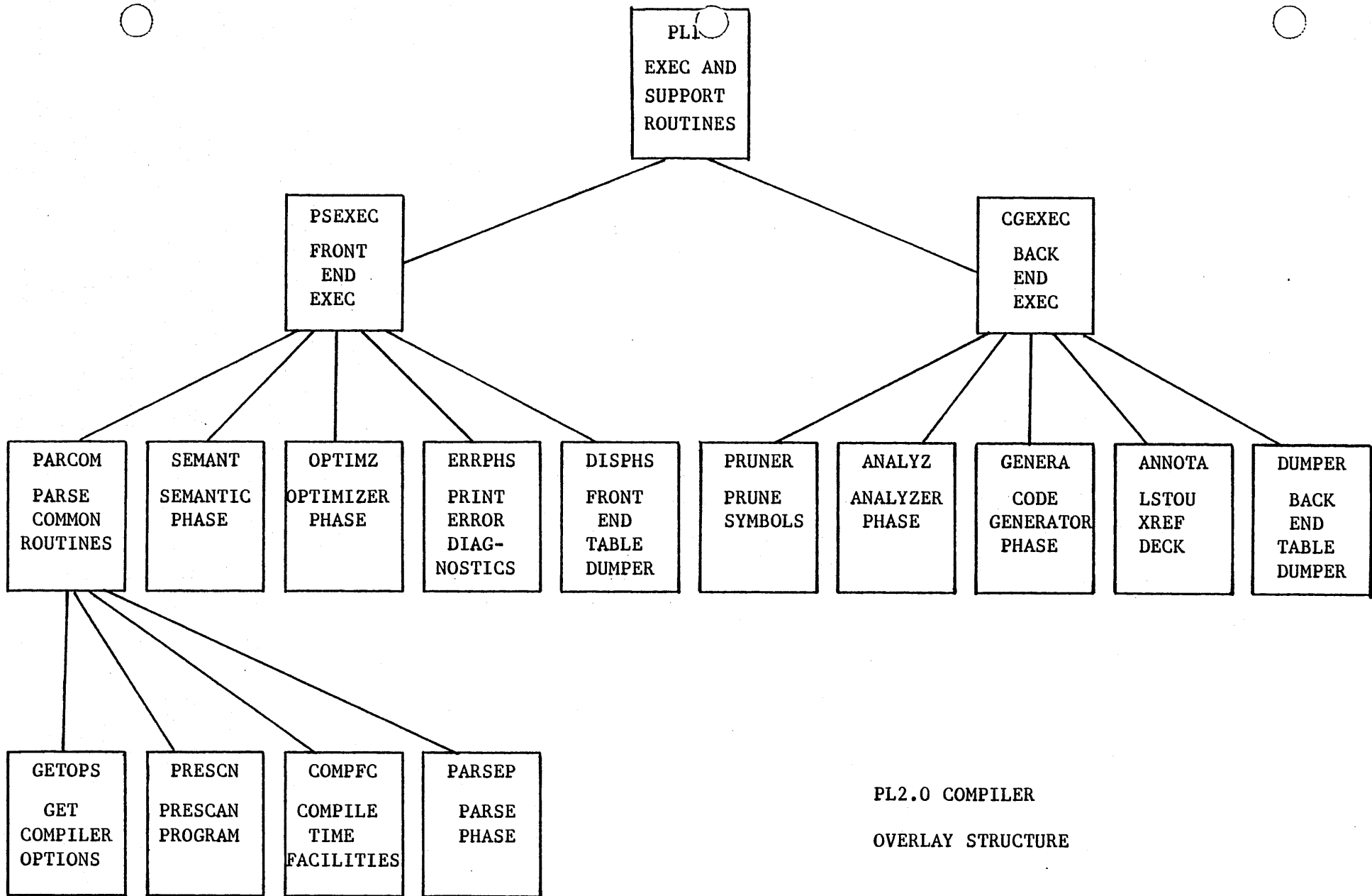
- CP-6 I/O INTERFACE
- CP-6 MONITOR INTERFACE
- CP-6 SORT INTERFACE
- CP-6 IDS-II INTERFACE
- CP-6 OBJECT UNITS
- CP-6 RUNTIME LIBRARY
- CP-6 DE BUG SCHEMA

PL/I

- IMPLEMENTATION LANGUAGE (SUBSET)
 - COBOL-74
 - IDS-II
- CROSS COMPILER ON GCOS-III
- EXPORT FACILITY TO CP-6 (OBJECT)
- PL/I RUNTIME ON CP-6

PL/I STRUCTURE





PL2.0 COMPILER
OVERLAY STRUCTURE

PL/I SCHEDULE

GCOS-III PL/I

- CURRENT STATUS
RECOMPILED COBOL-74
RECOMPILING ITSELF
- FIELD TEST FW744
- PRODUCT PL/I FW752

CP-6 PL/I CROSS COMPILER

- IMPLEMENTATION FW747-FW831
- SYSTEM TEST FW832-FW843
- AVAILABILITY (COBOL-74) FW835
- TWO ADDITIONAL PEOPLE FW826

CP-6 PL/I COMPILER

- AVAILABLE FW852

CP-6 PRELIMINARY DESIGN REVIEW

SORT/MERGE

- PRODUCT STRUCTURE
- CP-V COMPATIBILITY
- PERFORMANCE
- TESTING
- SCHEUDLE
- MAINTENANCE STRATEGY

SORT/MERGE PRODUCT STRUCTURE

- USES BOTH CP-V AND GCOS III ALGORITHMS
 - REPLACEMENT/SELECTION SORT TOURNAMENT
 - BOTH POLYPHASE AND STANDBY TAPE SORT
 - MINIMUM TREE DISK SORT
- USES CP-6 MONITOR FOR INPUT/OUTPUT
- IMPLEMENTED IN COMBINATION OF PL/6 AND ASSEMBLY LANGUAGE
- IMPLEMENTATION AND DOCUMENTATION VIA CP-6 SOFTWARE FACTORY ON CP-V

SORT/MERGE PRODUCT STRUCTURE

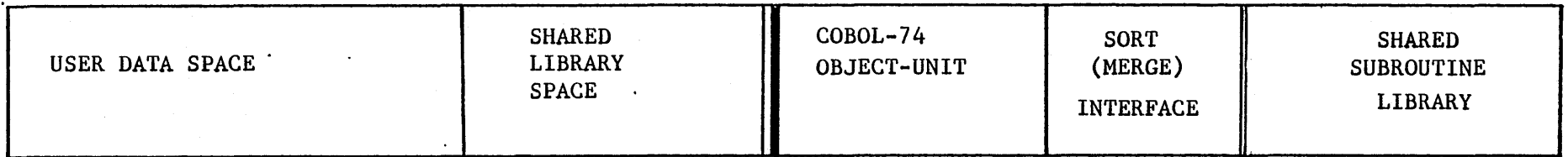
- COMBINATION OF
 - LINKABLE SUBROUTINES
 - SHARED SUBROUTINE LIBRARY MODULES

- LINKABLE SUBROUTINES
 - COBOL-74/SORT INTERFACE
 - COBOL-74/MERGE INTERFACE
 - SORT FREESTANDING DRIVER
 - MERGE FREESTANDING DRIVER

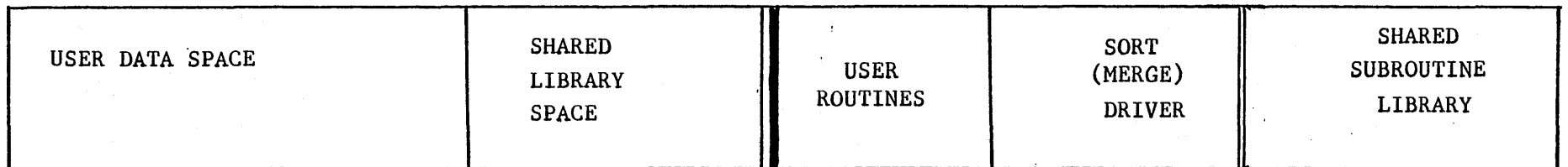
- SHARED SUBROUTINE LIBRARY MODULES
 - PARAMETER VALIDATION
 - KEY COMPARISON COMPILER
 - SORT TOURNAMENT DRIVER
 - SORT COLLATION DRIVER
 - SORT COLLATION INPUT ROUTINE
 - SORT COLLATION OUTPUT ROUTINE
 - SORT STRING DISTRIBUTION MANAGER
 - SORT STRING COMBINATION MANAGER
 - MERGE COLLATION DRIVER

S O R T / M E R G E P R O D U C T S T R U C T U R E

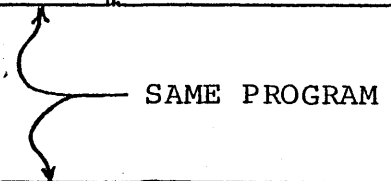
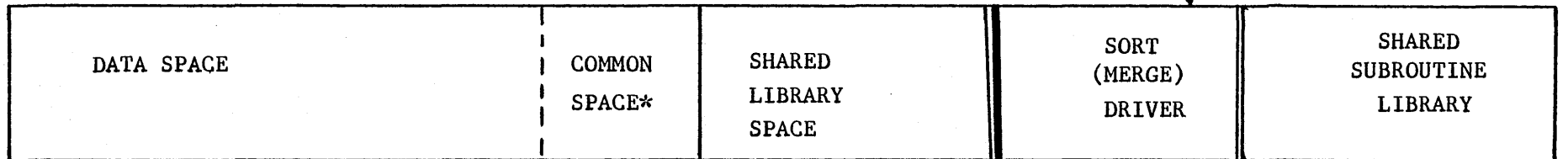
COBOL-74 SORT/MERGE



"CO-RESIDENT" SORT/MERGE



FREESTANDING SORT/MERGE



*WHEN INVOKED VIA M\$LINK

SORT/MERGE COMPATIBILITY

- HIGH CP-V COMPATIBILITY IN
 - KEY DESCRIPTION AND TRANSLATION
 - RECORD DESCRIPTION
 - FILE POSITIONING
 - USER OWN-CODE INTERFACES
- FULL COMPATIBILITY WITH COBOL-74
- WILL EXECUTE IN EITHER BATCH OR TIMESHARING MODE
- FULL COMPATIBILITY WITH ANS FORMAT TAPES AND ALL CP-6 FILE TYPES
- DIFFERS FROM CP-V IN
 - BLOCK DIRECTIVE DELETED IN FAVOR OF LABEL INFORMATION
 - LIMIT DIRECTIVE MODIFIED
 - NEW DATA TYPES ADDED FOR COBOL-74

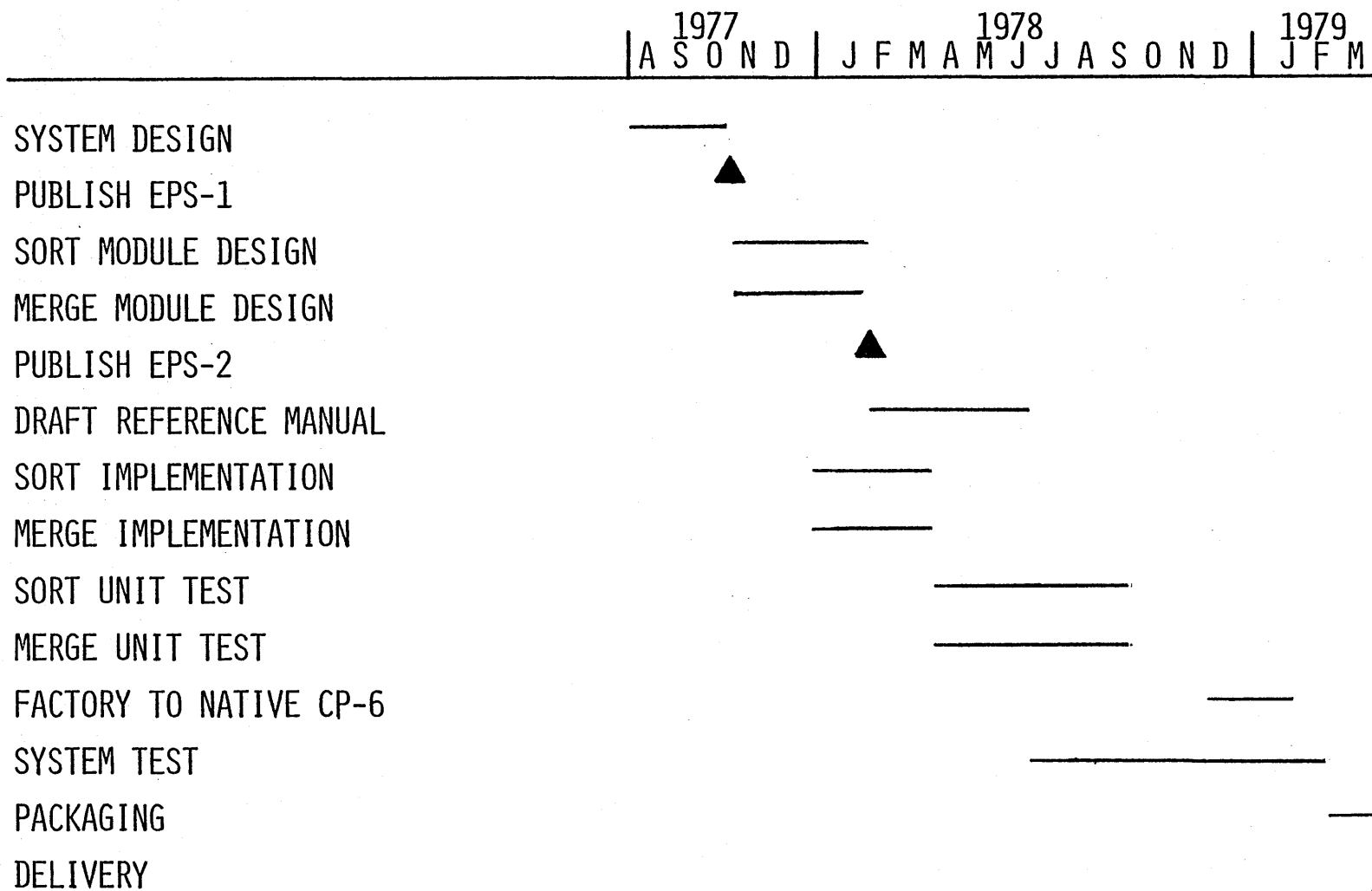
SORT/MERGE PERFORMANCE

- GOALS
 - PERFORMANCE EQUIVALENT TO GCOS III
 - DISK SORT SPACE NOT TO EXCEED 1.2 X
INPUT FILE SIZE
- FINAL PERFORMANCE HIGHLY DEPENDENT ON MONITOR
I/O PERFORMANCE

SORT/MERGE TESTING

- PLAN TO USE SORT/MERGE SYSTEM AS OPERATING SYSTEM TEST VEHICLE
- PLAN EXTENDED UNIT AND SYSTEM TESTING TO REFINE PERFORMANCE
- PLAN EXTENDED COBOL-74 SORT/MERGE TESTING VIA FEDERAL AUDIT ROUTINES

SORT/MERGE SCHEDULE



SORT/MERGE MAINTENANCE STRATEGY

- MAINTAINED ON CP-6 SYSTEM.

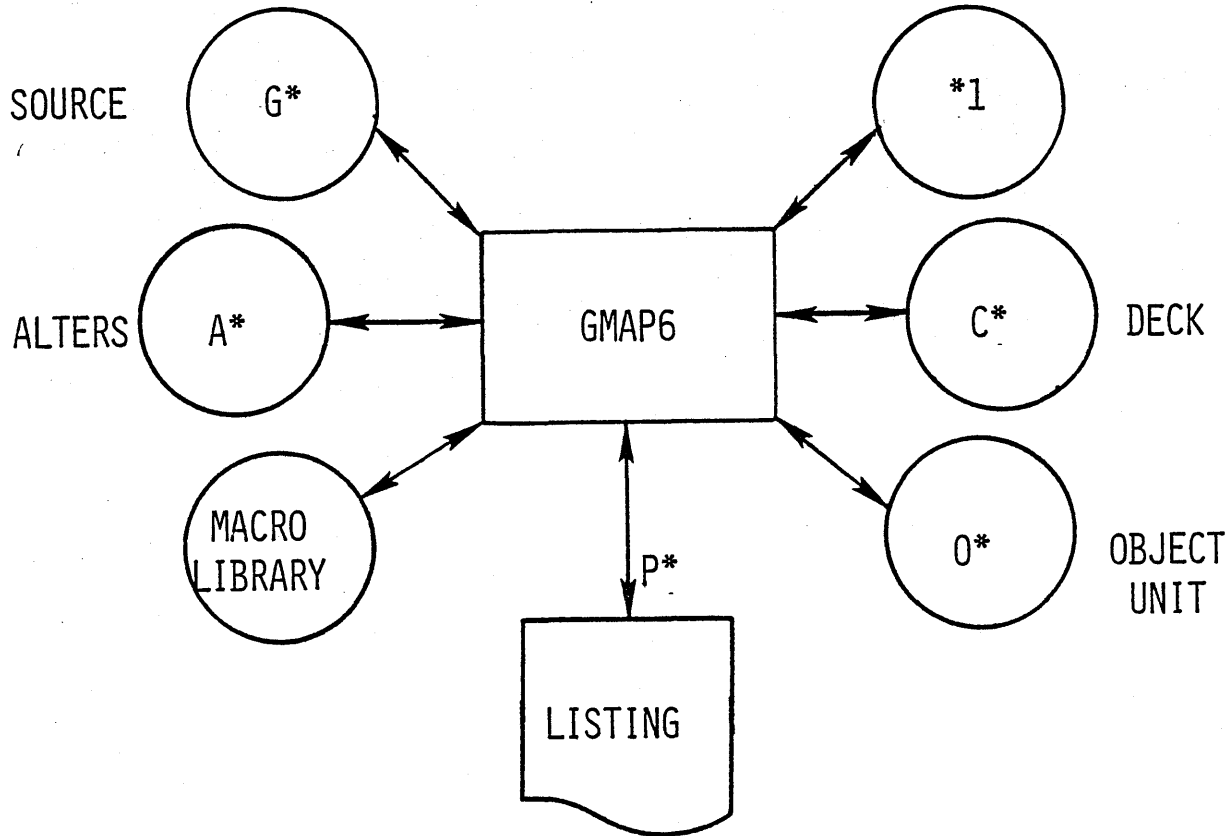
JAW
770914

CP-6 PRELIMINARY DESIGN REVIEW

GMAP6

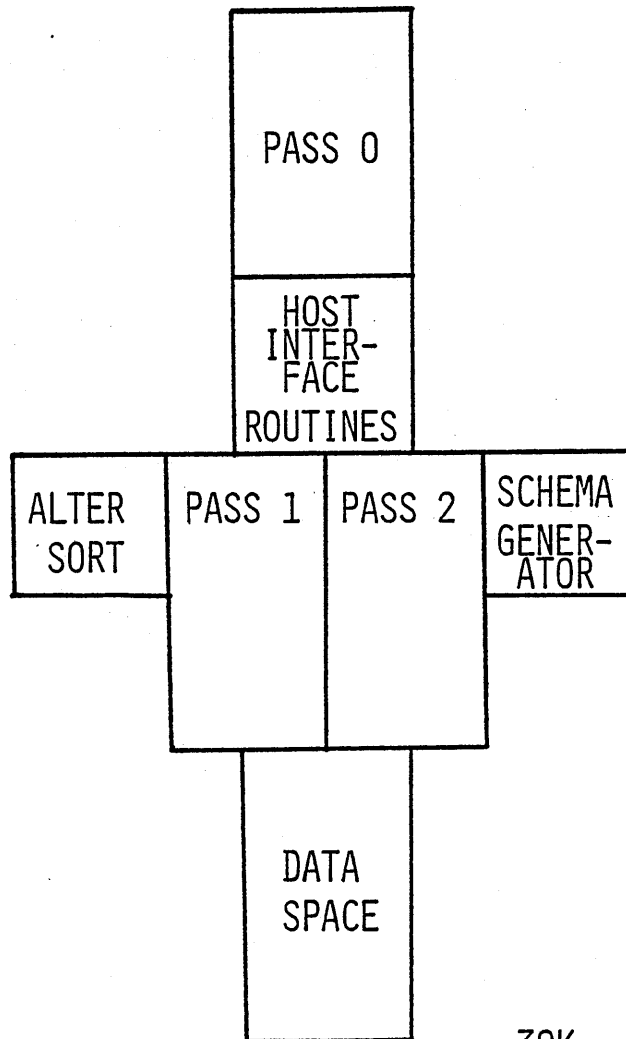
- PROCESSOR STRUCTURE
- COMPATIBILITY
- PERFORMANCE
- TESTING
- SCHEDULE
- MAINTENANCE STRATEGY

GAM
770914



GMAP6 STRUCTURE
FILE UTILIZATION

GMAP6 STRUCTURE
STORAGE UTILIZATION



GMAP6 COMPATIBILITY

- GCOS66
 - COMMON PROCESSOR/COMMON LANGUAGE
 - SYSTEM PERSONALITY MACROS DIFFER
 - ALTER MECHANISM MAY DIFFER
 - LOGICALLY SIMILAR OBJECT UNITS

- CP-V
 - NONE

- GCOS-III
 - LARGELY COMMON LANGUAGE
 - GMAP6 ADDS SECTION/SEGMENT VISIBILITY
 - SOME DIFFERENCES IN PSEUDO-INSTRUCTIONS
 - DIFFERENT OBJECT OUTPUT FORMATS

GMAP6 PERFORMANCE

- TARGET IS PARITY WITH GCOS EXECUTION OF GMAP AND APPROXIMATELY SAME CORE REQUIREMENTS.
- GMAP ON GCOS VARIES (DEPENDING ON MACRO USE) FROM 5000 - 20000 SOURCE STATEMENTS PER MINUTE OF PROCESSOR TIME ON A 66/80.
- ACTUAL PERFORMANCE MAY VARY FROM THIS TARGET AS A FUNCTION OF CP-6 I/O. HOWEVER, NO PERFORMANCE PROBLEMS ENVISIONED AT THIS TIME.
- GMAP6 IS NOT SHARED IN RELEASE 1.

GAM
770914

GMAP6 TESTING

- IMPLEMENTATION VERSIONS
 - LIMITED SPECIFIC TESTING FOR NEW FEATURES
 - LIMITED GENERAL TESTING
 - ACTUAL USE IN DEVELOPMENT OF CP-6 AND LANGUAGE SYSTEM SUPPORT ROUTINES

- FINAL VERSION
 - SELF ASSEMBLED
 - LIMITED GENERAL TESTING
 - ACTUAL USE IN ASSEMBLY OF ALL SYSTEM TEST VERSIONS OF CP-6 AND ITS PRODUCT SOFTWARE

GAM
770914

GMAP6
DEVELOPMENT SCHEDULE
MILESTONES

- PHASE 1 FW 748
 - PROVIDE SECTION DECLARATION
 - SYMDEF/SYMREF ENHANCEMENTS
 - PROVIDE FOR SYSTEM MACRO SUBSTITUTION
 - PRODUCE EXPORT OBJECT UNIT
 - RUN UNDER GCOS III
 - INTERNAL RELEASE ONLY
- PHASE 2 FW 813
 - PRODUCE PROGRAM SCHEMA
 - ENHANCED RELOCATION FLEXIBILITY
 - STILL RUNS UNDER GCOS III
 - STILL PRODUCES EXPORT OBJECT UNIT
 - INTERNAL RELEASE ONLY
- BEGIN NATIVE TEST FW 826
 - RUNS UNDER CP6
 - NO OBJECT UNIT OUTPUT
 - TEST VEHICLE ONLY
- BEGIN SYSTEM TEST FW 839
 - MERGE GMAP66 (ADF2V) FEATURES
 - COMMON PROCESSOR WITH GCOS66
 - CAPABLE OF ASSEMBLING SELF
 - PRODUCES CP6 OBJECT UNIT
- SHIP FW 913

GAM

770914

GMAP6
MAINTENANCE STRATEGY

- COMMON PROCESSOR WITH GMAP66
 - SINGLE SOURCE BASE
 - SINGLE PROJECT TEAM
- SPECIALIZED HOST SYSTEM INTERFACE
 - ISOLATES CP-6 INTERFACES
- PROCESSOR PACKAGING
 - INITIALLY DELIVERED AS EXPORT OBJECT UNITS
TO BE CONVERTED AND LINKED ON CP-6 FOR
SYSTEM TEST
 - FINAL PRODUCT DELIVERED AS CP-6 OBJECT UNITS

GAM
770914

CP-6 PRELIMINARY DESIGN REVIEW
PHOENIX SUPPLIED SOFTWARE

- SUMMARY
 - BASED ON EXISTING PRODUCTS
 - MAXIMIZE COMMON SOURCE
 - HIGH DEGREE OF COMPATIBILITY WITH CORRESPONDING GCOS PRODUCTS
 - PERFORMANCE TARGET - PARITY WITH GCOS

I-D-S/II FOR CP-6

- A CODASYL STRUCTURED DATA BASE SYSTEM
- TRANSLATORS CONVERT DDL AND DMCL → SCHEMA
- TRANSLATORS CONVERT DDL → SUBSCHEMA
- GCOS III I-D-S/II PLUS
 - INCREASED SECURITY
 - FIXED 1024 WORD PAGE SIZE
 - SHARED DBCS PROCEDURE (ASL)
 - MULTI-LANGUAGE INTERFACE
 - IMPORTANT EDMS FEATURES
 - ON-LINE TRANSLATION OF DDL, DMCL

I-D-S/II

MOTIVATING FACTORS

- NEW HARDWARE
 - NSA, WORKSPACES, ASL
 - SHARED PROCEDURE IN DBCS

- CP-6 AND GCOS-III SYSTEM DIFFERENCES
 - CONCURRENT ACCESS CONTROL
 - BUFFER MANAGEMENT
 - JOURNALING
 - FILE I/O
 - COMMAND LANGUAGE

- XEROX PARC
 - NEW (OLD) FEATURES

I-D-S/II

MAJOR EDMS & I-D-S/II DIFFERENCES

- NO IDS INDEXED RECORD SET RELATIONSHIPS
- NO IDS LOGICAL RESTRUCTURING PACKAGE
- DIFFERENT DATA TYPES

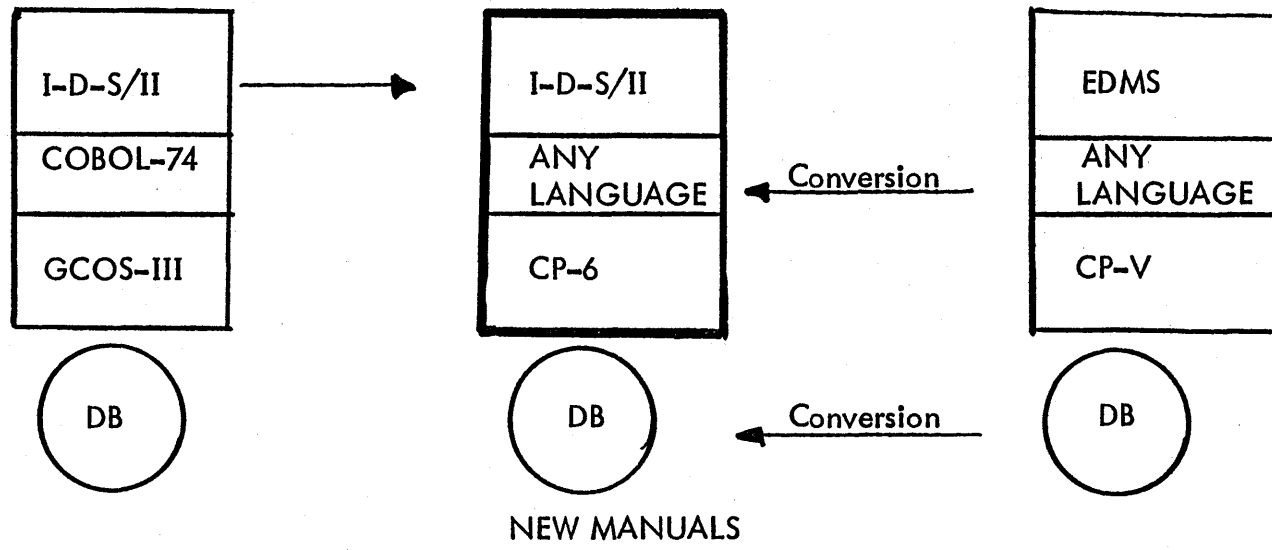
I-D-S/II

NEW FEATURES FOR XEROX PARC

- RUN TIME TRACE
- RUN TIME STATISTICS
- MULTI LANGUAGE INTERFACE
- CHECKSUM OF INTEGRATED FILE PAGES
- ENCRYPTION OF INTEGRATED FILE PAGES
- ENCRYPTION OF COMMON JOURNAL

I-D-S/II

RESOLVING DIFFERENCES



I-D-S/II

- PRODUCT DELIVERED TO US AS A GCOS III RELEASE

- STRUCTURE
 - DBCS 16,000 Lines GMAP
 - DBACS 32,000 Lines PL/1
 - UTILITIES 12,000 Lines GMAP

- DROPPING
 - INTERACTIVE I-D-S/II
 - PHYSICAL RESTRUCTURING UTILITY
 - DELETE UTILITY

I-D-S/II PERFORMANCE

- 64 μ SEC FOR EACH DBCS CLIMB
- SOME COMPUTE FOR COBOL-74 OIL SLICK
- EXPECT PERFORMANCE TO BE THE SAME AS GCOS III AND EDMS.
- CP-6 WILL HAVE SHARED DBCS PROCEDURE AND SHARED TRANSLATORS AND UTILITIES

I-D-S/II MAINTENANCE

- EACH NEW GCOS III I-D-S/II RELEASE WILL BE A PROBLEM TO BE CONSIDERED INDIVIDUALLY.
- ERRORS AND CUSTOMER PROBLEMS WILL BE SCANNED BEFORE ASKING FOR PHOENIX SUPPORT.
- ALL FIXES FROM PHOENIX WILL REQUIRE REVIEW BEFORE INSTALLATION.

I-D-S/II

TESTING AND VERIFICATION

- PHOENIX 4J TESTS
- CP-6 FUNCTION TESTS

I-D-S/II

DEPENDENCIES

I-D-S/II DB3.0	OCT 77
ASSEMBLER	DEC 77
CP-6	APR 78
PL/1 FOR CP-6	SEP 78
COBOL FOR CP-6	DEC 78
DELIVERY	APR 79

TWO PERSONS AT PRESENT, ONE OPENING

IDP FOR CP-6

- INTERACTIVE DATA BASE QUERY LANGUAGE
- TOPOLOGY ANALYSIS
- RETRIEVAL OF SEQUENTIAL DATA FILES
- AUTOMATIC OR SPECIFIED REPORT FORMAT
- REASONABLY SMALL (EST 15,000 LINES OF PL-6)

IDP STRUCTURE

- COMMAND MODULE
- VALIDATION MODULE
- RETRIEVAL MODULE
- DICTIONARY MODULE
- TOPOLOGY MODULE
- REPORT MODULE

RPG-II RUN TIME

IDP FLOW

- USER ENTERS IDP STATEMENTS OR RUNS A FILE OF IDP STATEMENTS FILLING IN BLANKS
- IDP ANALYZES TOPOLOGY
- IDP GENERATES PROGRAM IN MEMORY AND EXECUTES WITH RUN TIME LIBRARY

IDP

- PERFORMANCE - SAME AS IDP IN CP-V
- MAINTENANCE - ENTIRELY AT LADC
- TESTING - CP-V REGRESSION TESTS PLUS NEW TESTS
- DEPENDENCIES - CP-6, I-D-S/II, SORT
- SCHEDULE - 2Q79

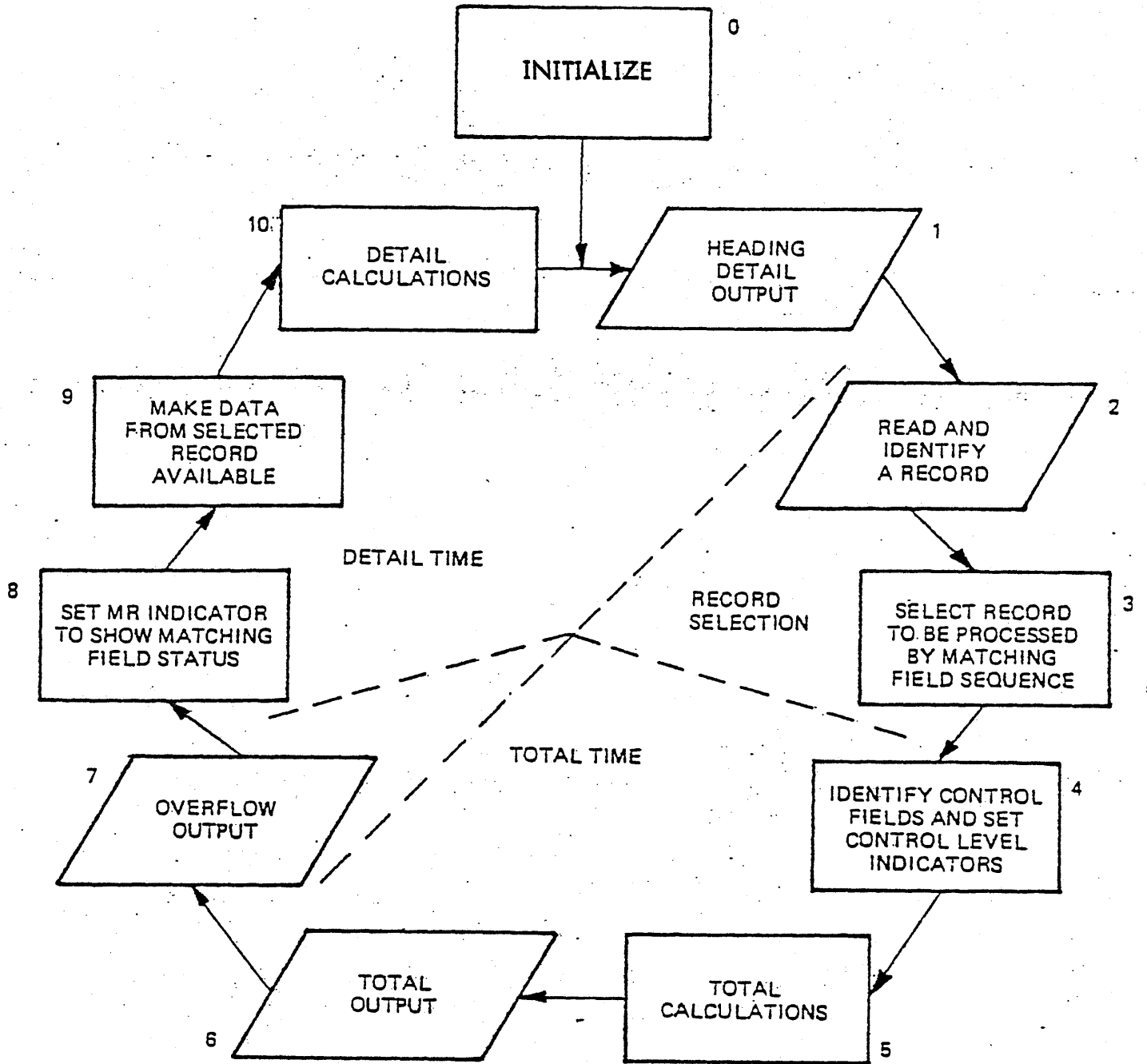
- PRESENTLY ONE PERSON, ONE OPENING

RPGII FOR CP-6

- COMMERCIAL DATA PROCESSING
- PROGRAM PHASES
 - COMPILATION
 - EXECUTION
- COMPATIBILITY

LANGUAGE ELEMENTS

- FILE PROCESSING
- FORMS CONTROL
- OUTPUT EDITING
- OPERATIONS CODES
- FIXED SPECIFICATION FORMS
- FIXED PROGRAM LOGIC



CP-V RPG VS. CP-6 RPGII

- COMPLETE SUBSET
- RELATIVE I/O
- LOOK AHEAD
- SPREAD CARD/PLACE
- ARRAYS/TABLES
- OPERATION CODE EXTENSIONS

OVERALL STRUCTURE

- 530 DESIGN
 - MOM (MACRO OPERATION MACHINE)
 - INTERPRET
 - RUN-TIME

- COMPILATION PHASES
 - FILE AND EXTENSION PROCESSOR
 - INPUT PROCESSOR
 - CALCULATIONS PROCESSOR
 - OUTPUT PROCESSOR
 - FINALIZE PROCESSOR

- PERFORMANCE

- SIZE

- SPEED

TESTING AND VERIFICATION

- EXISTING TESTS
- OUTSIDE VENDOR

DEPENDENCIES AND SCHEDULE

- CP-6
- PL-6 STRING FUNCTIONS
- ASSEMBLER
- RELEASE
- RECRUIT

SOFTWARE TEST AND RELEASE CONTROL

HONEYWELL

- TESTING, ANALYSIS AND RELEASE DECISION BY ENGINEERING MANAGEMENT
- COMPREHENSIVE TESTING OF FUNCTION, PERFORMANCE, IMPROPER USE
- STABILITY DETERMINED BY EXPOSURE TO LADC USERS
- SPECIFIC INDIVIDUAL ASSIGNED AS RELEASE MANAGER
- RELEASE DECISION BY MANAGER, LADC

9/19/77 IPR/SK

○
TEST TOOLS

○
HONEYWELL

- EXTERNAL LOAD GENERATOR
(QUESTA SYSTEM)
- HARDWARE MONITOR
- TIME-SHARING USER SIMULATOR

9/19/77 IPR/SK

TEST AND CONTROL GROUP

HONEYWELL

- PROVIDES SUPPORT TO THE DEVELOPMENT STAFF AND
RELEASE MANAGER
 - DEVELOPS TEST TOOLS AND TECHNIQUES
 - PROVIDES MEASUREMENT AND ANALYSIS SUPPORT
 - LIBRARIAN TO PROVIDE ORGANIZED, DOCUMENTED
TEST DATA AND HISTORY

- TEST GROUP ESTABLISHED
 - STAFFING IN PROCESS NOW
 - TOOL BUILDING IN PROCESS NOW
 - ACTIVE MEASUREMENT BEGINS IN EARLY 1978,
1 YEAR BEFORE RELEASE

9/19/77 IPR/SK

CP-V TO CP-6 CONVERSION

- PRINCIPAL ORIENTATION TO CONVERSION WHILE CP-V SYSTEM IN PLACE, BUT CAN BE DONE ON CP-6
- WILL NOT CONVERT MACHINE LANGUAGE PROGRAMS
- WILL CONVERT BULK OF:
 - HIGH LEVEL SOURCE PROGRAMS
 - DATA FILES WITH KNOWN DATA TYPES
- TAPE IS PRIMARY MEDIUM OF CONVERSION
- SPECIFIC CONVERTERS DEVELOPED BY EACH DEVELOPMENT GROUP

HIGH LEVEL SOURCE PROGRAMS

- CP-6 ANS FORTRAN COMPATIBLE WITH CP-V ANS FORTRAN
- COBOL USES CAPS AND PROGRAMS REQUIRE SOME MANUAL ASSISTANCE
- APL CONVERTER GENERATES APL SOURCE FROM WORKSPACES
- BASIC CONVERTER CONVERTS ALL STATEMENTS WHICH HAVE SAME SEMANTICS IN ANS BASIC
- EDMS SCHEMA CONVERSION TO IDS DATA DEFINITION LANGUAGE

FILE CONVERSION METHODS:

- ALL CHARACTER FILES TRANSPORTABLE VIA ANS TAPE (UNLESS KEYED FILES)
 - CP-V CAN WRITE ASCII, CP-6 CAN READ EBCDIC
- OTHER FILES USE ONE OF TWO METHODS:
 - DATA CONVERSION PACKAGE, CP-6 TAPE WRITE PACKAGE ON CP-V
 - CP-V TAPE READ PACKAGE, DATA CONVERSION PACKAGE ON CP-6

8/31/77 DH

CP-6 MAJOR MILEPOSTS

- Transition to Honeywell July 1976
- Begin CP-6 Architecture Phase July 1976
- Level 66A Installation November 1976
- Begin CP-6 Detail Design and Implementation Phase January 1977
- A-System First L66 Code February 1977
- B-System Software Factory May 1977
- C-System PL-6 July 1977
- Level 66B Installation September 1977
- Complete CP-6 Architecture Phase September 1977
- F-System - File Management November 1977
- H-System - Time Sharing February 1978

8/31/77 RJL

CP-6 MAJOR MILEPOSTS (continued)

• Four Million Words on L66B	February	1978
• Second CP L66B	March	1978
• K-System - Batch	June	1978
• Language Processors	July	1978
• Demonstration (K-System)	August	1978
• First Release Implementation Complete	October	1978
• In-House ALPHA Test	November	1978
• In-House Beta Test	January	1979
• Customer Benchmark	March	1979
• Final System Build (S-System)	May	1979
• Controlled Field Release	June	1979

CP-6 PROGRESS TO DATE

DESIGN REVIEW -

- 40 Documents
- Over 2100 pages

SOURCE

- In Checkout
16,000 (PL-6) 6,000 (Assembler) D-System

SOURCE

- Awaiting Checkout (Estimate)
26,500 (PL-6) 12,000 (Assembler)

8/31/77 RJL

CP-6 PROGRESS TO DATE

A-SYSTEM

Checkout Bootstrap Environment

Mini I/O

Mini DELTA

Mini Boot

Used GCOS GMAP - ABS LOAD

Used NSA (WSQO)

B-SYSTEM

Software Factory Test

BMAP

LINK

DEF

Established Monitor Environment (WSQ1)

C-SYSTEM

PL-6 Test

Unit Test

Memory Management

CPU Scheduler

Service Decoder

Fault Handler

D-SYSTEM

Stabilize C-System

Basis for Parallel Checkout and Integration

8/31/77 RJL

A AARDVARK

MINI-DELTA
MINI-I/O

A BISON

FACTORY LINK
FACTORY DEF
INITIAL

A COYOTE

MM
SCHLD
SERVICE DECODER
FAULT HANDLER
PL-6 MODULES

A DINGO

TEST DRIVERS
STABLE SYSTEM

8/31/77

RJL

Δ FOX

FILE MANAGE(1)
GET GRANULE
OPEN/CLOSE
KEYED FILE
BOOT SUBK
GHOST I
DEVICE LP

Δ GIRAFFE

SYSTEM * FILES
TAPE LABELS
CONSEC FILES

Δ HORTEBEST

CURT GRANULE
TAPE FILES
RANDOM FILES(1)
FILE BLOCK ACCESS
CP-V TO CPG TRANSPORT

Δ ICE BEAR

TAPE MOUNTS
SHARED FILES
EFT

Δ JACKAL

TAP RANDOM FILES
*TFILES
RELATIVE/INDEXED FILE

Δ ELAND

STEP(1)
FIRST USER

Δ FALCON

SHARED PRO(1)
JIT MAKER

Δ GANDER

SP. SHARED PRO
SYSMAX(1)
JIBEX(1)

Δ HORSE

DISK BOOT
SYSMAX(2)
SHARED PRO(2)
FETCH
ENQ/DEQ

Δ IBIS

M/EVENT
EXIT CONTROL
JOBSTART
ASL

Δ JACKRABBIT

DINGO

Δ EAGLE

Δ FLICKER

Δ GOPHER

Δ HARE

Δ IGUANA

Δ ELEPHANT

IOQ(1)
KEYIN(10S)

Δ FROG

XDELTA(1)
MUPARSE

Δ HAMSTER

USER DELTA(1)

Δ JERBOA

JIBEX(2)
EDIT
USER DELTA(2)
XDELTA(2)

Δ GOBILIA

RD/WRT UC
HOST FE

Δ HIPPO

STREAM OPEN
ONLINE USER
HOST LOGON

Δ IMPALA

OUTPUT COOP
OUTSYM(1)
KEYIN(1)

Δ JELLYFISH

INPUT COOP
OUTSYM(2)
KEYIN(2)
EXECUTE FILES
BATCH JOBS
MBS(1)
PRESCAN(1)

8/31/77

RJL

AKINGFISHER
MULTI-VOLUME TAPES

ALEMMING
MULTIPLE PACK SETS
CATALOGUE
HQP/PCON
GAC(1)
JOURNAL FILES

AM...
ENCLOSURE
COMPRESSION
STOW/FETCH
GAC(2)

AKIWI
HOLDING
RECOVERY(1)
MULTI PROCESSING
PMDAT

ALEOPARD
PM
PMDAT

AMONKEY
HOLDING
RECOVERY(2)

AJAGUAR

AKANGAROO

ALION

AMINK

ANIGHTENGALF

AKOALA
PCL
TIGR
ANLX(1)

ALARK
SUPER
CONTROL
LOGON
LINK
LEAD
RATES
IBEX(3)

AMARMOT
TOQ(2)
ANLX(2)
ACCOUNTING

AKOALA
MILDEV
INSYM
MRS(2)
RESOURCE MGT
PRESCH(2)

ALLAMA
FORMS
WSN
DOWNLINE LOAD
FRONT END FORMAT
LG ADMINISTRATION

Δ BET TEST

Δ BENCHMARK

Δ RELEASE

Δ OCELOT

Δ PUMA

Δ QUAIL

Δ RABBIT

Δ SQUIRREL

8/31/77

RJL

COMMENT ON PL-6

- PL-6 to Date
 - No (serious) Code Generator Bugs
- CP-6 Code in PL-6
 - Straight Forward Checkout
 - L66 Hands-on
 - L66 Simulator
- Good Trend
- Too Soon To Tell

NSA PROBLEM RESOLUTION

- CP-6 - Dependent upon NSA
- CP-6 / NSA Problem Definition
 - L. Krasny (LADC)/R. Mynatt (Phx) - Change Request
- Two Sets of NSA Boards
 - In Hardware Engineering for LADC
- Approved Change Request (C.O.)
 - Install on System P1 (redwire).
 - Install on LADC Boards (Betty McCulley-Phx).
- FED (R. Fawcett) Tests and Sends Boards to LADC (P. Germain).
- NSA T&D (P. Drown)/Phx Sends Tape to LADC (H. Geshwind).
- LADC FED (E. Steinhauer) Coordinates with PHX FED (R. Fawcett).

NSA PROBLEM RESOLUTION (Continued)

What's Missing ?

- Single Control Point
 - Hardware Changes
 - T & D
- Installation and Checkout Expertise For New Hardware at LADC

SOLUTION

- New Product Test Engineering
 - Coordination
 - Installation

8/31/77

RJL

CP-6 PROBLEMS AND ISSUES

- Test and Diagnostics
 - TOLTS Interface to CP-6
 - HEALS FED Errorlog Spec
- CP-V Drain on Management and Programmer Resources
- Software Factory - Dual 560
 - Production for CP-6
 - Test for CP-V

8/31/77 RJL

