my DIC file

SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

D I S T R I B U T E D I N T E R A C T I V E C O M P U T I N G N O T E 928

issued by

SOFTWARE ENGINEERING INFRASTRUCTURE

Dr R W Witty

Meeting between academia and GEC, York,
3 and 4 November 1983

25 November 1983

DISTRIBUTION:

F R A Hopgood
D A Duce
W P Sharpe
D Talbot
R W Witty
STI/Infrastructure file

(see next page)

1. INTRODUCTION

This meeting was the software engineering counterpart to the IKBS Milton Hill House meeting held in June 1983.

The meeting was chaired by Mr David Talbot, the Alvey SE Director. Those attending included representatives from each SE site named as potential recipients for a GEC machine. Full list of attendees appended.

D Talbot welcomed everyone on behalf of SERC and Alvey. RWW then outlined the objectives of the meeting as:

(1) Inform investigators about

(a) STI Infrastructure policy

(b) new GEC Series 63 machine

(c) UNIX on Series 63

(d) Alvey IKBS/SE collaborative MUM project.

(2) Academic community to discuss its requirements with GEC.

2. STI INFRASTRUCTURE POLICY

The policy document was circulated to attendees prior to the meeting.

3. GEC SERIES 63 MACHINE

Mike Todd and Tony St John gave a fairly detailed description of the machine. They wisely concentrated on the areas of most interest such as the virtual memory system and only outlined input-output.

4. UX63 AND TOOLS

Dave Williams described UX63. This is AT & T Series 3 UNIX with virtual memory and some Berkeley extensions running on top of a GEC specific kernel based on the OS4000/OS6000 kernel.

Details of the implementation, release dates and contents of releases were given. Multi-user UNIX with PDP-11 style memory management is already working in Dunstable.

Release 1	Dec '83	PDP11 memory management
Release 2	Nov '84	Virtual memory
Release 3	June '84	Full utilities and compilers.

The academics received the details from GEC in a friendly and constructive mood. This was helped by the fact that UNIX was actually working on the '63'. The technical progress since the IKBS Milton Hill House meeting was clear and led to a much better presentation and reception.

Graham Hill gave a good talk on AI languages. New figures showed increased LISP performance - this was encouraging. Franzlisp is being implemented as an interface to Cambridge LISP.

5. ALVEY IKBS/SE COLLABORATION

Bill Sharpe gave a good description of the collaboration between Alvey and GEC to produce the IKBS MUM infrastructure. This went down well with the academics who were encouraged to see that the funding agencies and GEC were putting in money and men not just dumping machines on academics. The SE community requested implementation projects for SE tools similar to the IKBS initiative.

6. SE REQUIREMENTS

In summary, the SE community confirmed that their requirements were basically well represented in the STI Infrastructure policy document.

6.1 Oxford

ISO Pascal not variants

No 'C'

Modula 2 instead of C

Standard version of ML, efficiently implemented.

6.2 Imperial

UNIX	ML/LCF
SERCnet	Modula 2
Prolog	Hope

6.3 York

as above inc ML/LCF
data base for IPSE
conferencing
Ada
Mosses' SIS

6.4 Manchester

Standard ML/LCF
Hope
CLU

6.5 Newcastle

LAN (ethernet) and WAN implementations suitable for RPC/IPC implementation e.g. datagram interface or UNIX device driver interface.

6.6 UMIST

Typed languages not untyped
ML, Hope not Lisp

Modula 2)
CLU) for data abstraction
Pascal Plus)

Good Pascal

7. PERFORMANCE

Overall the SE community felt that GEC were implementing the right things. The community expressed considerable concern over the lack of quantitative information (Lisp was an impressive exception).

The community wished to see

(a) clear performance targets set for the implementors

(b) measurements of achieved performance.

The community has been heavily involved in the PERQ collaboration and is keen to see that performance is achieved not just functionality. GEC took the message seriously. RWW has since fed the performance issue into the Alvey/GEC Steering Committee and Mike Todd as Project Officer undertook to carry out the above requests on targets and achievements.

8. SUMMARY

A much more constructive meeting than Milton Hill House, probably because GEC had more progress to report and the uncertainty factor was so much lower.

The community expressed its requirements and the expected non trivial subset common to IKBS and SE emerged, giving support to the Alvey strategy of keeping both programmes in line with each other.

9. APPENDICES

1. Agenda

2. List of participants

3. GEC slides.

DIC4/19/jg

SOFTWARE ENGINEERING INFRASTRUCTURE

3/4 NOVEMBER 1983

A G E N D A

Chairman : Mr David Talbot
Alvey Software Engineering Director

Thursday 3rd November

18.30-18.45	Introduction : R W Witty
18.45-20.15	Series 63 : GEC
20.30	Dinner

Friday 4th November

09.00-10.15	Hardware details : GEC
10.15-10.30	Coffee
10.30-11.30	UX63 : Unix on the 63 : GEC
11.30-12.00	Specialist Languages : GEC
12.00-13.00	GEC/University Collaborative Project : SERC
13.00-14.00	Lunch
14.00-15.00	SE Community Requirements : academe
15.00-16.00	General Discussion

SOFTWARE ENGINEERING INFRASTRUCTURE

3/4 NOVEMBER 1983

LIST OF PARTICIPANTS

Mr D Talbot	Alvey
Dr R W Witty	)
	) Alvey/SERC
Mr W P Sharpe	)
Dr H Barringer,	Manchester
Dr P Hitchcock,	Newcastle
Dr J P Black,	Newcastle
Mr R Boot,	NCC
Prof I Pyle,	York
Prof J Welsh,	UMIST
Dr R J Cunningham,	Imperial
Dr S Abramsky,	Imperial
Prof C A R Hoare,	Oxford
Mr B Sufrin,	Oxford
Dr D Coleman,	UMIST
Dr S Young,	UMIST
Dr I Wand,	York
Mr C Forsyth,	York
Dr M Todd,	GEC
Mr A O St John,	GEC
Dr A G Hill,	GEC
Mr K St Pier,	GEC
Dr D T Williams,	GEC
Dr D Worsnip,	SERC, Swindon

SOFTWARE ENGINEERING INFRASTRUCTURE MEETING

4TH NOVEMBER 1983

A G E N D A

		SPEAKER
9.00	INTRODUCTION	TONY ST. JOHN
9.15	HARDWARE ASPECTS	
	ORDER CODE	TONY ST. JOHN
	VIRTUAL ADDRESSING	
	CONNECTIVITY	
10.15	COFFEE	
10.30	SOFTWARE ASPECTS	
	SYSTEM SOFTWARE	DAVE WILLIAMS
	STANDARD LANGUAGE PRODUCTS	KEITH ST. PIER
11.30	SPECIAL LANGUAGES	GRAHAM HILL
	PROLOG	
	LISP	
	POPLOG	
12.00	COLLABORATIVE PROJECT	
	ALVEY OBJECTIVES & PROCEDURES	ROB WITTY
	IKBS/SE INFRASTRUCTURE IMPLEMENTATION PROJECT	MIKE TODD
13.00	LUNCH	
14.00	DISCUSSION SESSION	
15.00	FINISH	

SE 4/11/83
AOS1

ORDER CODE

REGISTER SET - NORMAL MODE

- 16 GENERAL PURPOSE, 32 BITS
- 2 FLOATING POINT 128 BITS
- 16 VIRTUAL ADDRESS SPACE DESCRIPTOR 64 BITS
- 3 REPEAT CONTROL REGISTERS
- 1 CONDITION & PRIVILEGE LEVEL REGISTER

ORDER CODE

OPERATIONS

ARITHMETIC	:	INTEGER, FLOATING, PACKED DECIMAL, CHARACTER
LOGICAL	:	BOOLEAN, SHIFTS, BIT OPERATIONS
CONTROL	:	BRANCH, PARAMETER CHECK, ENTRY POINT, LEVEL CHANGE
I/O	:	CONNECT, INTERRUPT

STRING ETC BY REPEAT CONSTRUCTS

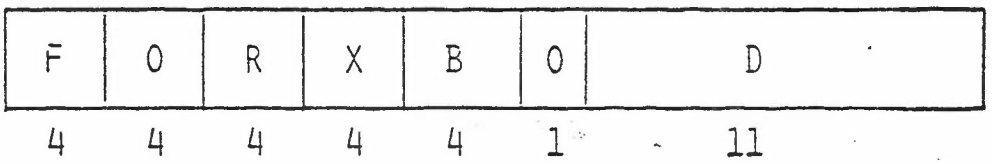
SE 4/11/83
AOS3

ADDRESS CALCULATION

INSTRUCTION FORMAT 1

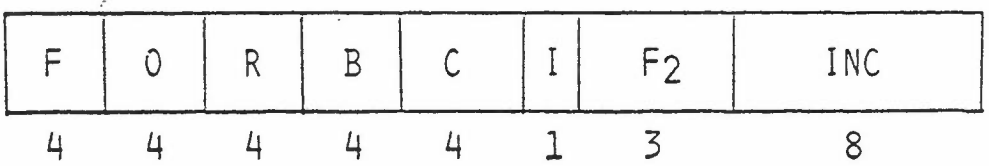
VA = (B) + D

INSTRUCTION FORMAT 2



VA = (B) + [(X) + D] SCALED

INSTRUCTION FORMAT 3



VA = (B)

NOTES:

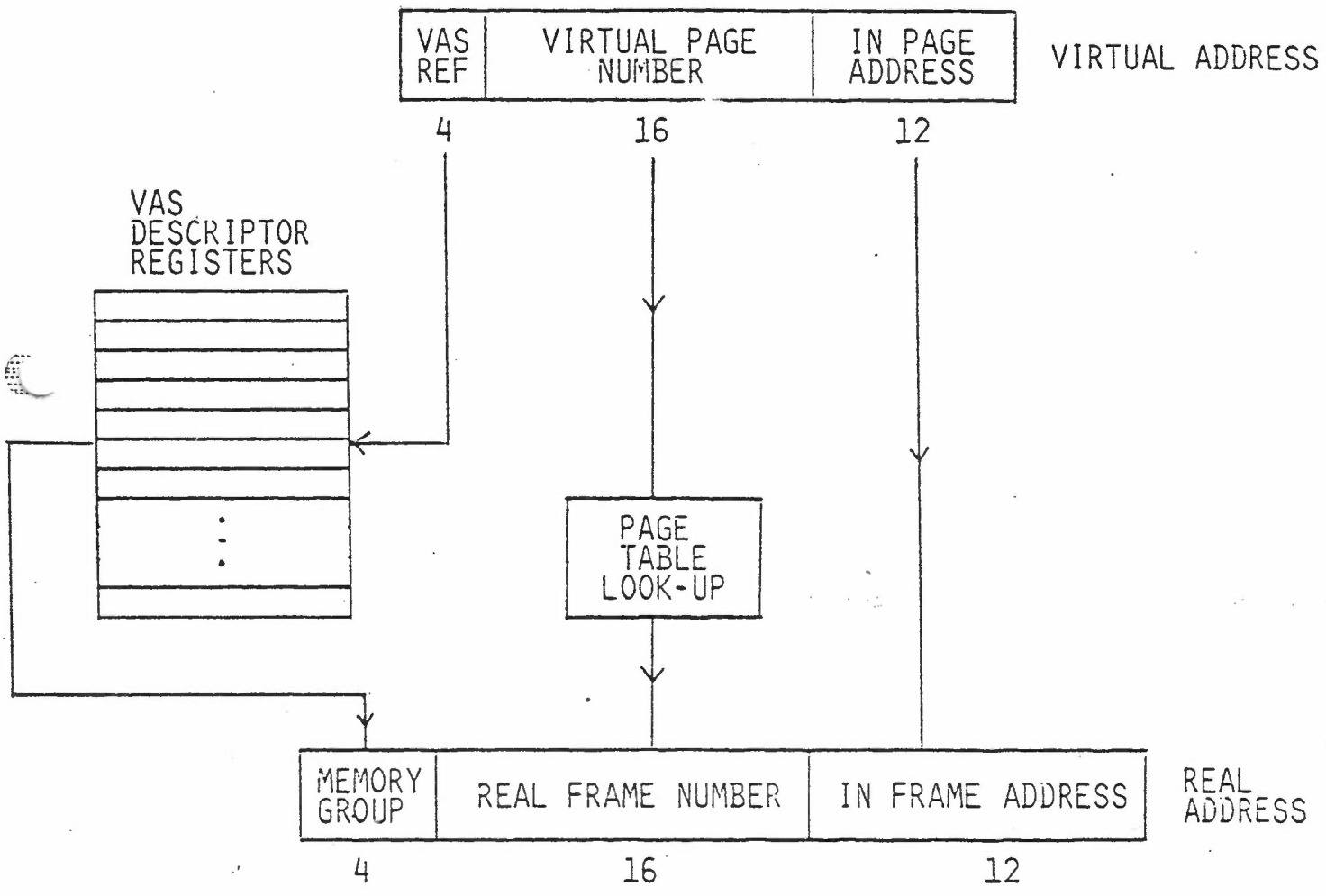
F, 0 — Define Format and Operation

R, B — Any of 16 32 bit Registers for Integer Ops

R — Either of 2 FP Registers for FP Ops

X — Any of 16 32 bit Registers

ADDRESS MAPPING



NOTES:

- PAGE TABLES USUALLY LOCKED INTO STORE
- QUICK LOOK-UP MEMORY (QLM) ENHANCES PAGE TABLE ACCESS
- VAS IS ONE OF 16 VIRTUAL ADDRESS SPACES

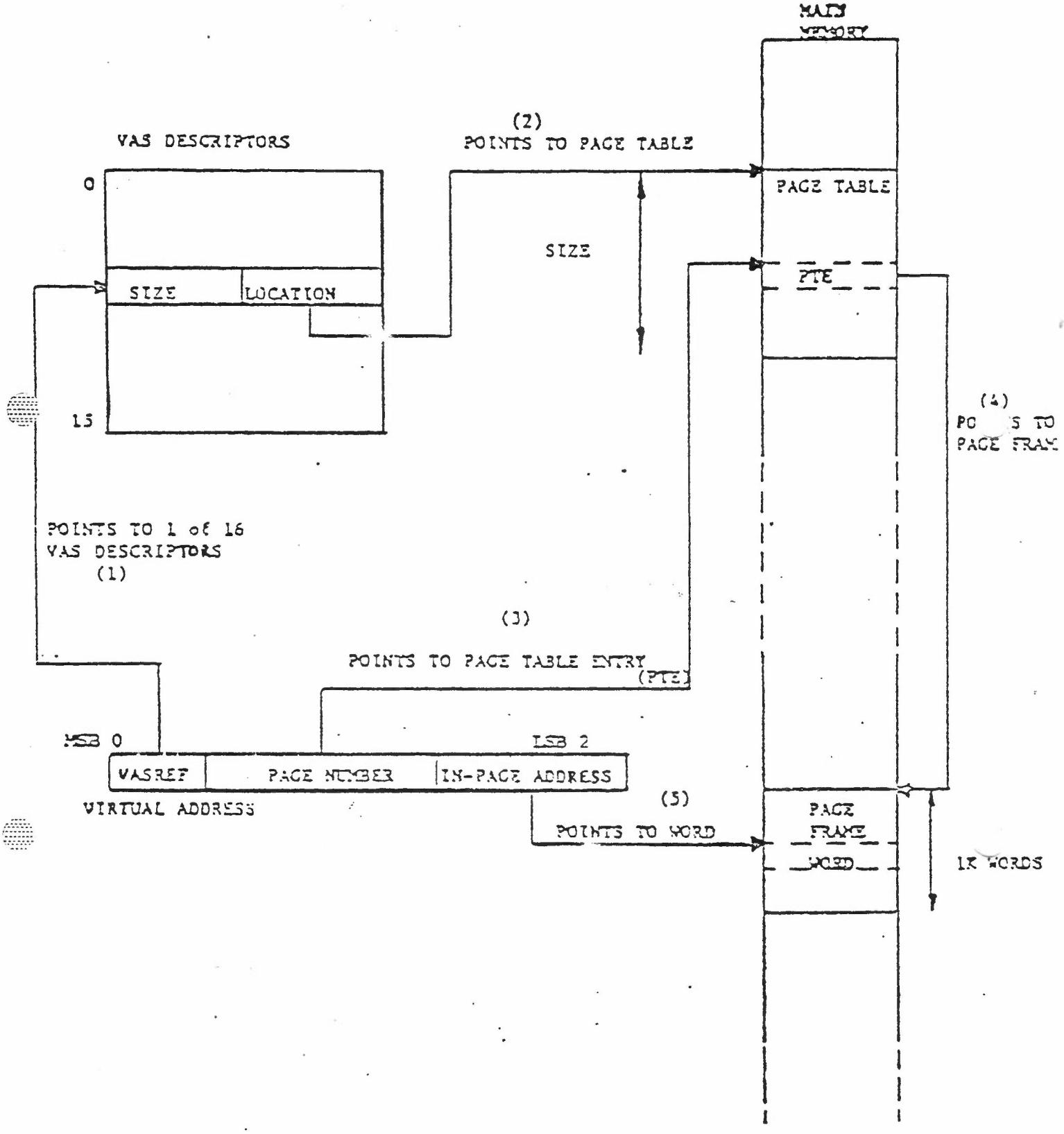
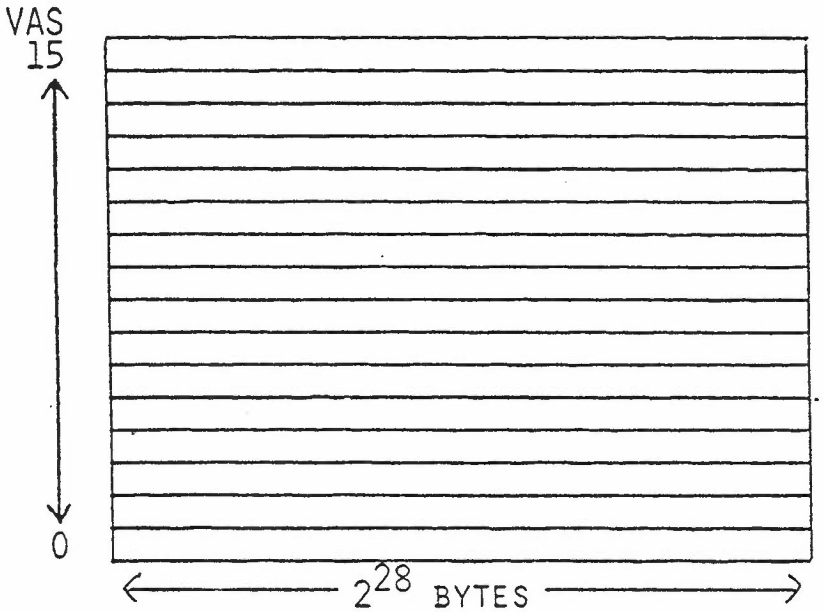


Figure 3.2.1-1 Principle of Address Mapping

PROCESS ADDRESS SPACE



TOTAL SPACE AVAILABLE TO A SINGLE PROCESS

$= 16 \times 2^{28}$ BYTES

$= 2^{32}$ BYTES

PAGE SIZE IS 4K BYTES

VAS DESCRIPTOR REGISTERS

- 16 IN ALL, ADDRESSED BY MS 4 BITS OF VIRTUAL ADDRESS
- CONTENTS:

  PAGE TABLE LOCATION
  PAGE TABLE SIZE
  VAS IDENTIFIER (VASID)
  KEY FOR HASHING INTO QUICK LOOK UP MEMORY
  ADDRESSING MODE (ABSOLUTE/VIRTUAL)
  MEMORY GROUP
  ACCESS FLAGS
  PRIVILEGE LEVEL
  EXECUTION MODE (NATIVE/COMPATIBILITY).

TOTAL SPACE AVAILABLE TO SYSTEM

= 64K x 2^{28} BYTES

= 2^{44} BYTES

PROTECTION

- EACH VAS HAS A LEVEL 0-15 ASSIGNED

  0 = MOST PRIVILEGED

  15 = LEAST PRIVILEGED

- PROCESSOR HAS AN EXECUTION LEVEL HELD IN A REGISTER

- CODE EXECUTING AT LEVEL n CAN ONLY ACCESS DATA IN A VAS OF LEVEL $\geqslant n$ (OR ITS OWN VAS)

- CURRENT EXECUTION LEVEL $\geqslant$ CODE VAS LEVEL

- ACCESS PROTECTION ON VAS AND PAGE:

  READ
  WRITE
  EXECUTE

- VAS DESCRIPTOR REGISTERS CAN ONLY BE MANIPULATED BY PRIVILEGED INSTRUCTIONS

SUMMARY

- 16 VAS's IMMEDIATELY ADDRESSABLE IN A PROCESS
- EACH VAS UP TO 2^{28} BYTES
- PAGE SIZE IS 4K BYTES
- QLM GIVES SPEEDY ACCESS TO RECENTLY USED PAGE TABLE ENTRIES
- HARDWARE ENFORCES PROTECTION BASED ON LEVELS AND ACCESS PERMISSION

CONFIGURATION LIMITS

LOGIC BOARDS

MANDATORY

2	PROCESSOR	
1	MEMORY CONTROLLER	
2	MEMORY BOARDS	(1 MBYTE → 4 MBYTE)
1	LOCAL SYSTEM MULTIPLEXOR	(PRODUCES BYTE BUS)
1	DISC SP OR NISP + DISC_7	

OPTIONAL

FLOATING POINT ACCELERATOR

UP TO 10 MORE MEMORY BOARDS

TRADED AGAINST

UP TO 10 MORE SECONDARY PROCESSOR BOARDS

UP TO 7 DEVICE CONTROLLERS

E.G. ACDC FOR 16 V24 CHANNELS

SE 4/11/83
AOS4

CONFIGURATION LIMITS

ADDRESSING

16	MEMORY BOARDS	(1MB → 4MB)
15	SECONDARY PROCESSORS	
1024	LOGICAL CHANNELS PER BYTE BUS	
2^{32}	BYTES OF VM PER PROCESS	
2^{44}	BYTES OF VM SYSTEM WIDE	

SE 4/11/83
AOS5

LANGUAGES/UX63 - SHORT TERM DEVELOPMENTS

- Move language products from OS6000
  - Shorter Timescales than new developments
- Linker
  - Preserves existing compiler output interface
  - Amended to output a.out file
- C Compiler
  - CC compiler from System III (CPP & CO)
  - Code generator from OS6000 (written in PASCAL and BABBAGE)
- PASCAL Compiler
  - itself a PASCAL/BABBAGE program
- FORTRAN 77 Compiler
- ADB
  - UNIX code
  - Any modifications straight to System V COFF format
- SDB

4 November 1983
KSP

LANGUAGES/UX63 - MEDIUM TERM DEVELOPMENTS

- To coincide with the introduction of System V
- Existing linker becomes a last compiler pass
  - outputs relocatable COFF files
  - no change to compilers
- LD
  - from System V
  - standard UNIX libraries etc
  - full support for adb, sdb,.....

4 November 1983
KSP

LANGUAGES - FURTHER DEVELOPMENTS

- APPLICABLE TO NEW DEVELOPMENTS IN THE COMMON LANGUAGES
- INTRODUCE A COMMON COMPILER BACK END POLICY
  - INTERMEDIATE DATA STRUCTURE
  - INTERMEDIATE CODE FILE
  - DATA ALLOCATION IN BACK END
- STANDARD BUILDING BLOCKS FOR FRONT ENDS
- OPTIMISERS
  - GLOBAL
  - LOCAL
- DEBUGGING FACILITIES
- SUPPORT FOR INTERLANGUAGE WORKING
  - PROCEDURE CALLING
  - INPUT-OUTPUT
  - STORE MANAGEMENT
- BASED ON UNIX AND C

4 NOVEMBER 1983
KSP

LANGUAGES - PROCEDURE CALLING STANDARD (SERIES 63)

- STANDARD REGISTER ALLOCATION
  - TOP OF STACK
  - FRAME POINTER
  - CODE BASE
  - STATIC DATA BASE
  - LINK
  - ARGUMENT POINTER
  - FUNCTION RESULT (INTEGER/REAL)

- STACK LAYOUT

- ARGUMENT BLOCK
  - FIXED LENGTH ENTRIES
  - QUAD/DOUBLE/REAL/INTEGER/POINTER

- OPTIONS
  (A) USE ALL 'STANDARD' CALLS
  (B) USE PRIVATE INTERNAL CALLS AND EXPLICIT 'STANDARD' CALLS

4 NOVEMBER 1983
KSP

SE/IKBS TOOLS

- PROLOG
- LISP
- ML/LCF
- POP-2

4 NOVEMBER 1983
AGH

PROLOG

- CPROLOG CURRENTLY RUNNING ON

  OS4000
  OS6000

- EDINBURGH AI EVALUATING REPLACEMENT FOR DEC10 PROLOG

4 NOVEMBER 1983
AGH

LISP

- Cambridge University LISP currently running on OS6000
  - Value Cell
  - Function is Value
  - Shallow Bound
  - Tagged Pointers
  - All Function Types (EXPR, FEXPR, MACRO) have both spread and unspread forms
  - Compacting garbage collector
- Other LISP dialects will be catered for by interface packages
- Franz Lisp interface under development

4 November 1983
AGH

LISP PERFORMANCE

TEST	TIME (MSEC)		
	SERIES 63	PSL/750	FZL/750
TAK 18 12 6 (GENERIC)	4540	7463	19.9 SEC
TAK 18 12 6 (INUM)	1160	1377	3.6 SEC
EMPTYTEST 10000	60	85	394
SLOW EMPTYTEST 10000	498	1258	3644
LENGTHTEST 100	1180	2380	4712
CDRLTEST 100	664	2142	
CDDRTEST 100	581	1377	
LISTONLY CDRTEST 1	4040	9656	
ARITHMETICTEST 10000	1540	1632	7736

'TURNER CONSTANT'

INUM	39800
GENERIC	9365

4 NOVEMBER 1983
AGH

ML/LCF

- Currently being implemented using Franz Lisp interface package

4 November 1983
AGH

Pop - 2

- POPLOG

  BEING OBTAINED

  FROM SDL

4 November 1983
AGH