Virtual Memory

* ISA has large VA space.
  - Allows user to uniquely identify lots
* Physical Memory is smaller
  - Cost issue

* Virtual Memory Management
  - Access Control
  - Translation

* The VAX Model

![Diagram of VAX Model]

Process 1
Process 2
Process n

Balance Set

Virtual Memory

Physical Memory

Working Set
Page Tables

* One for each region

* For example, the P0 Page Table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>P0 Base Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PTE for Page 0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PTE for Page 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PTE for Page 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>L-1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Sequentially stored in System Virtual Space

* P0LR used for ACV checks

* PTE used for ACV, TNV checks
The Abstraction

Which Page Table
Example: 00 P0PT

1. Is Page No. < P0LR?
   NO: ACV Fault!

2. Get Correct PTE

Index by Page No.

PTE for Page 1

3. V Prot PFN The PTE
   Check Protection
   NO: ACV Fault!

4. Is Page Resident (i.e. V = 1)
   NO: TNV Fault!

5. All Cool: PA
The PTE

31  V  Prot  M  PFN  0

Can I Believe The PFN

Has the Page Been Written

In which frame is the Page Stored

Do I have the Right to do this access

\{NO,R,R/W\}^{K,E,S,U}

81 Possibilities in 4 bits

Note: No Ref. Bit!
VA of x:

PTE of Page Containing X:

PA of x: And now we can access x
VAX Hardware Memory Map

(Assuming 9 bit VA Space)

16 pages, 32 bytes each
Example:
VA of a Byte on Page 1, Pθ Space

\[ \begin{align*}
\text{VA of } X & = \text{Page 1 of Pθ Space. We need to find the } \\
& \text{VA of the PTE of Page 1, Pθ Space.}
\end{align*} \]

\[ \begin{align*}
100100000 & \quad \text{PθBR} \\
+ & \quad \text{VA of the PTE of Page 1, Pθ Space}
\end{align*} \]

\[ \begin{align*}
100100100 & \quad \text{VA of } X \\
\times 4 & \quad \text{Page 1 of θ System Space. We need to find} \\
& \text{the PTE of the Page 1 of System Space}
\end{align*} \]

\[ \begin{align*}
10000100 & \quad \text{PA of Byte the PTE of Page 1, System Space}
\end{align*} \]

\[ \begin{align*}
1 \text{ ACCESS} & \quad \text{PFN of Page 1, System Space}
\end{align*} \]

\[ \begin{align*}
00100100 & \quad \text{PFN of Page 1, Pθ Space}
\end{align*} \]

\[ \begin{align*}
00001010 & \quad \text{PA of } X
\end{align*} \]
**Flat Model**

```
31  LOGICAL ADDR  0  \rightarrow
     \downarrow
     \rightarrow
32
```

**Real Address**

```
0  \rightarrow
19  SEGMENT SELECT  0000
     \downarrow
     \rightarrow
20
```

**Segmented Model**

```
  \rightarrow
logical addr
     \downarrow
     \rightarrow
32
```

```
Seg. Selector
13
```

```
Discriptor Table
```

```
Size determined by limit Reg.
```
Segment Registers

16 bits

CS
SS
DS
ES
SS

TI (which tab)

GDT
GDTR

LDT
LDTR

$2^{13} \times 8$ bytes each

Segment Descriptor

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| BASE [15:8] | L | LIMIT [15:8] |

G: Granularity 1 byte/4K bytes
P: Segment Present
PL: Privilege level
TYPE: Segment type
Segmentation AND Paging

Program

MOV A, PW

Linear Address Space

A is Logical Address

BASE

Linear Address of A

Physical Memory

PA

INVT

FRAME OF PHYSICAL MEM.

CR3: PDBR

CR0: 11

Virtual translation enabled

PACK DIRECTLY

BASE PTE

PACK TABLES

BYTE ON PAGE = BYTE ON FRAME

Linear Address

PTE

PFN

PAGE SWAP

PRIV: S - U R W

DIRECT ACCESS PRESENT