Computer Architecture Fundamentals

Load/Store/ALU

31 30	29 25	24 19	18 14	13	12 5	4 0
op	rd	op3	rs1	opf		rs2
op	rd	op3	rs1	0	asi	rs2
op	rd	op3	rs1	1	simm13	

Branches/sethi

31 30	29	28 25	24 22	21 0
op	a	cond	op2	disp22
op	rd		op2	disp22

Call

31 30	29 0
op	dip30

Computer Architecture: *A definition*

Computer architecture is the abstraction of a physical system (microcode and hardware) as seen by the machine-language programmer or a compiler writer. It is the definition of the conceptual structure and fuctional behavior of a processor as opposed to such attributes as the processor's underlying data flow and controls, logic design, and circuit technology."

From: Myers, G. J., "Advances in Computer Architecture," Second Edition, John Wiley & Sons, New York, N. Y., 1978.

Machine Code is Computer Architecture

- The machine code of a microprocessor describes and defines the architecture of a computer.
- Described in Manual of Operation or Architecture Manual

Assembly Language is not Computer Architecture

- Assembly Language is a representation of machine code that facilitates its programming
- There isn't a one-to-one correspondence between machine code and assembly language
- Assembly language may represent instructions or elements such as registers that are non existent on the machine code.
  - *Stack Pointer in SPARC*
  - *Move instruction in SPARC*

Rule of thumb

An element of a computer is architectural if it has a representation in the machine code of its processor.

Architectural Aspects

- Instructions
- Operand access modes
- Instruction formats
- Data types
- Primary memory
- Registers
- Interrupts, Traps, Exceptions
- Instruction execution flow
- Processor operation modes

Non Architectural Aspects

- Secondary memory (hard disk, flash memory, solid state drive, etc.)
- Buses
- Periferals
- Control unit
- ALU
- Fabrication technology
- Logic circuits

Aspects that may or may not be Architectural

- Caches
- Pipelines
- Stacks
- Input/output (I/O)

Always apply the rule of thumb

Fundamental Architectural Elements

- Instructions
- Registers
- Primary memory
- Data types
- Instruction formats
- Addressing modes
- Input/Output
- Interruptions
- Processor operation modes

Instructions

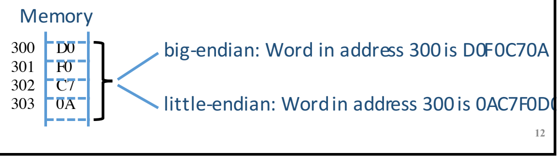
- Data transfer (move, load, stores, swaps)
- Arithmetic & logic (add, subtract, compare, and, or, shift, rotate)
- Control (branch, jump, trap, software interrupt, subroutine)
- I/O (input, output)
- Floating Point (add, subtract, multiply, divide)

Registers

- General purpose integer
- Program counter or instruction pointer
- Stack pointer
- Program status register
- Condition codes
- Floating point

Primary Memory (Architectural Memory)

- Is the random access memory space accessed through machine code instructions
- Typically organized in bytes (Intel 8086 exception)
- 2^{32} locations in most modern architectures
- Numbers represented with more than one location are stored big-endian or little-endian



big-endian: Word in address 300 is D0F0C70A

little-endian: Word in address 300 is 0AC7F0D0

Data Types

- Integer (sign or unsign)
  - *Literal – less than 8 bytes*
  - *Byte – 8 bits*
  - *Halfword – 16 bits*
  - *Word – 16 or 32 bits*
  - *Longword – 32 bits*
  - *Doubleword – 2 Words*
  - *Quadword – 4 Words*
- Floating Point
  - *32 and 64 bits*

Instruction Formats

- Determines the significance of the bits that constitutes an instruction
- Fixed size (RISC architectures)
- Variable size (CISC architectures)

Addressing Modes

Addressing Mode	Effective Address
Immediate	Operand is number included in the instruction
Register	Operand is the content of a register
Direct	A number in the instruction
Register Indirect	The content of a register
Based	The content of a register plus a number included in the instruction (displacement)
Based Indexed	The addition of the content of two register
Relative	The content of PC plus a number included in the instruction (offset)

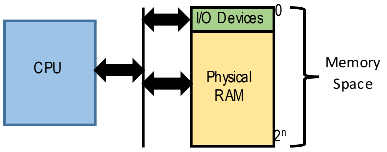
Primary Memory (Architectural Memory)

- Virtual Memory
  - *Mapping of architectural memory with physically implemented memory*
  - *Facilitates multiprocessing*
  - *Provides data protection scheme*

Input/Output

- Separate I/O storage space
- No architectural support

Memory-Mapped I/O



Interruptions

- Change the normal order of execution of instructions
- Save processor state (architectural registers)
- Take instruction execution to another memory space

Types of Interruptions

- Interrupts – caused by asynchronous external events
- Exceptions – caused when a special condition is detected during the execution of an instruction
- Traps – caused by instructions explicitly designed for requesting interrupts (software interrupt)

Processor Operation Modes

- Processors can usually operate on different modes
- Allow the operating system to protect critical information and prevent unwarranted access to the system
- Establish execution privileged shells
  - *Some instructions can only be executed on a specific mode*
  - *Any instruction can be executed in the most privileged mode*

Three Signature Architectures

- IBM System 360
- Digital VAX-11
- Intel 8086

IBM System 360 Architecture

- Introduced by International Bussiness Machines (IBM) in 1964
- Flagship architecture of IBM during the 60s and 70s
- Evolved into IBM System 370
- Close to two thirds of the market share by 1965

IBM 360: Instructions

- Data transfer
- Arithmetic, logic
- **Decimal**
- Branch
- Subroutines
- Floating Point
- Input/Output
- **Translate**

IBM 360: Registers

- 16 32-bit general purpose registers
- 4 64-bit floating point registers
- IA – instruction address
- PSW – Program Status Word

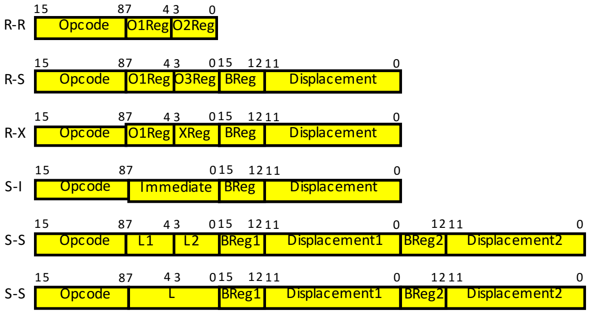
IBM 360: Memory

- 2^{24} bytes (2^{32} on later versions)
- Instructions must begin on even addresses.
- Big-Endian format

IBM 360: Data Types

Type	Size
Character	8 bits
Halfword	16 bits
Word	32 bits
Double Word	64 bits
Quad Word	128 bits
Short Floating Point	32 bits
Long Floating Point	64 bits
Decimal	1-16 bytes

IBM 360: Instruction Formats



R-R: 15 Opcode 8 | 7 O1Reg 4 | 3 O2Reg 0

R-S: 15 Opcode 8 | 7 O1Reg 4 | 3 O3Reg 0 | 15 BReg 12 | 11 Displacement 0

R-X: 15 Opcode 8 | 7 O1Reg 4 | 3 XReg 0 | 15 BReg 12 | 11 Displacement 0

S-I: 15 Opcode 8 | 7 Immediate 0 | 15 BReg 12 | 11 Displacement 0

S-S: 15 Opcode 8 | 7 L1 4 | 3 L2 0 | 15 BReg1 12 | 11 Displacement1 0 | BReg2 12 | 11 Displacement2 0

S-S: 15 Opcode 8 | 7 L 4 3 0 | 15 BReg1 12 | 11 Displacement1 0 | BReg2 12 | 11 Displacement2 0

IBM 360: Addressing Modes

Addressing Mode	Effective Address Operand 1	Effective Address Operand2	Effective Address Operand 3
R-R	Operand is [O1Reg]	Operand is [O2Reg]	
R-S	Operand is [O1Reg]	[BReg] + Displacement	Operand is [Reg3]
R-X	Operand is [O1Reg]	[XReg] + [BReg] + Displacement	
S-I	Immediate	[BReg] + Displacement	
S-S	[BReg1] + Displacement1	[BReg2] + Displacement2	
S-S	[BReg1] + Displacement1	[BReg2] + Displacement2	

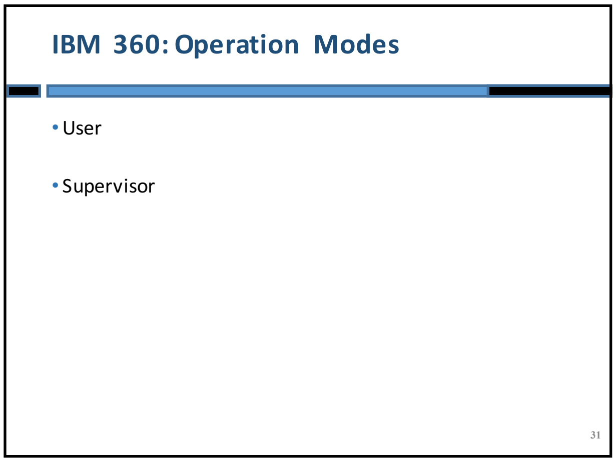
In S-S formats L, L1, And L2 indicate the size of the data in number of bytes

IBM 360: Input/Output

- Channels
- Separate processing unit

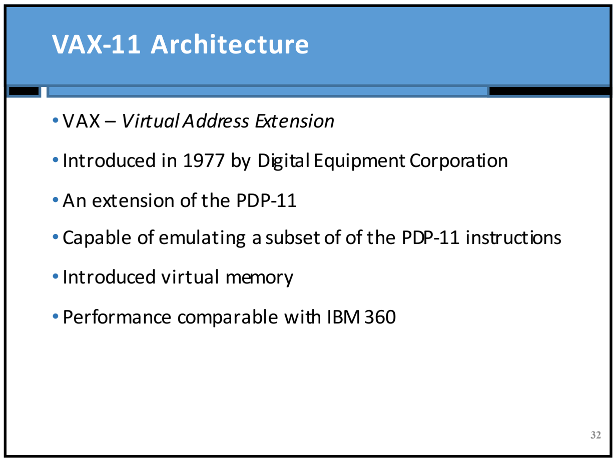
IBM 360: Interrupts

- Input/Output – I/O channels
- Program – program excecution exceptions
- Supervisor – Supervisor Call instruction
- External – external event
- Machine Check – conditions associated with I/O channels or CPU



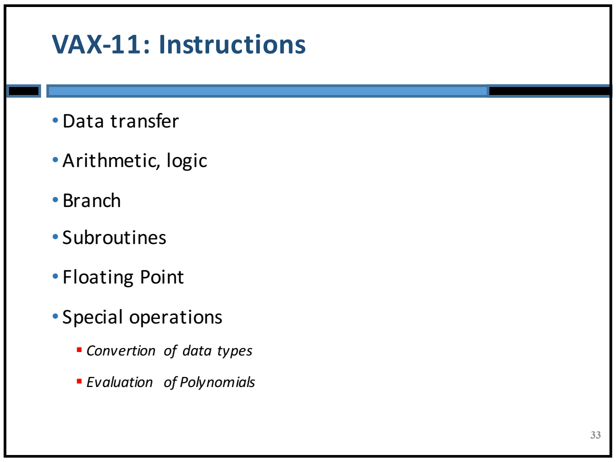
IBM 360: Operation Modes

- User
- Supervisor



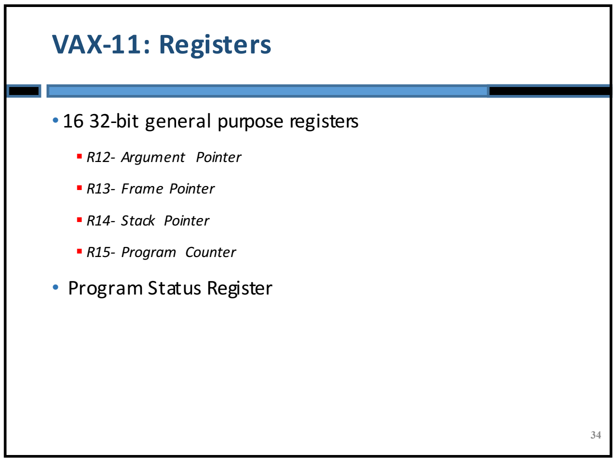
VAX-11 Architecture

- VAX – *Virtual Address Extension*
- Introduced in 1977 by Digital Equipment Corporation
- An extension of the PDP-11
- Capable of emulating a subset of of the PDP-11 instructions
- Introduced virtual memory
- Performance comparable with IBM 360



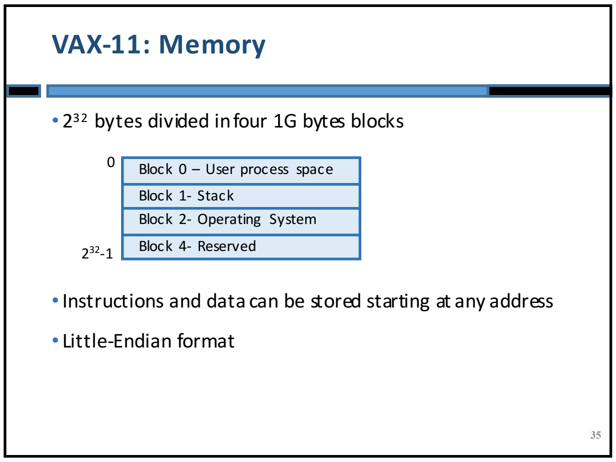
VAX-11: Instructions

- Data transfer
- Arithmetic, logic
- Branch
- Subroutines
- Floating Point
- Special operations
  - *Convertion of data types*
  - *Evaluation of Polynomials*



VAX-11: Registers

- 16 32-bit general purpose registers
  - *R12- Argument Pointer*
  - *R13- Frame Pointer*
  - *R14- Stack Pointer*
  - *R15- Program Counter*
- Program Status Register

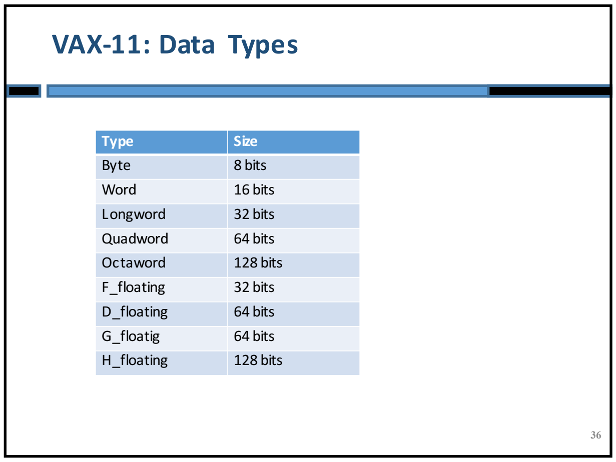


VAX-11: Memory

- 2^{32} bytes divided in four 1G bytes blocks

0	Block 0 – User process space
	Block 1- Stack
	Block 2- Operating System
2^{32}-1	Block 4- Reserved

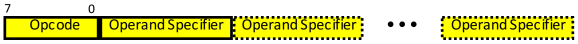
- Instructions and data can be stored starting at any address
- Little-Endian format



VAX-11: Data Types

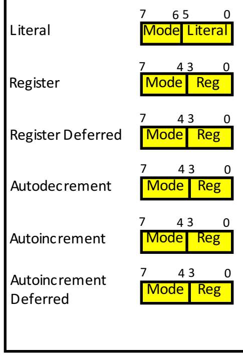
Type	Size
Byte	8 bits
Word	16 bits
Longword	32 bits
Quadword	64 bits
Octaword	128 bits
F_floating	32 bits
D_floating	64 bits
G_floatig	64 bits
H_floating	128 bits

VAX-11: Instruction Formats

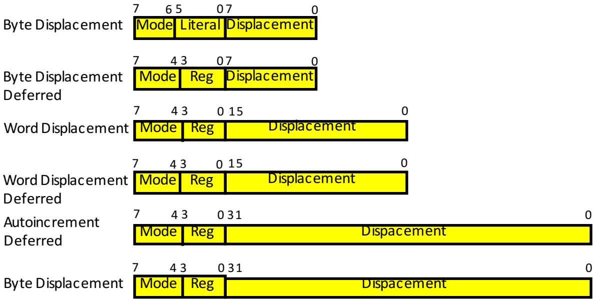


- *There can be from 1 to 6 operand specifiers*
- *An operand specifier can have from 1 to six bytes*

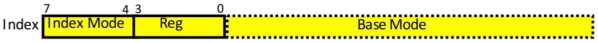
VAX-11: Operand Specifiers



VAX-11: Operand Specifiers



VAX-11: Operand Specifiers



Base Mode: Any addressing mode except Literal

VAX-11: Addressing Modes

Addressing Mode	Effective Address
Literal	Operand is literal value
Indexed	Base Mode EA + [[Reg]**length*]
Register	Operand is content of Reg
Register deferred	[Reg]
Autodecrement	Decrement Reg by *length*, then [Reg]
Autoincrement	[Reg], then increment Reg by *length*
Autoincrement deferred	[[Reg]], then increment Reg by 4
Displacement	Displacement + [Reg]
Displacement deferred	[Displacement + [Reg]]

length – number of bytes of data (byte=1, word=3, long=4)

VAX-11: Input/Output

- Memory-mapped I/O
- I/O registers mapped to main memory locations

VAX-11: Interrupts and Exceptions

- Interrupts - caused by external events usually not directly related to the executed code
- Exceptions – interruptions directly related with the executed code

VAX-11: Operation Modes

- Kernel –operating system management
- Executive – file management
- Supervisor- system processess
- User – user programs

Intel 8086 Architecture

- Introduced in 1978 by Intel Corporation
- Successor of Intel 8080/8085
- Predecessor of Intel® 64 and IA-32 Architectures
- Intel 8088- spin-off with additional 8-bit data bus
- 16 bit architecture
- Microprocessor implementation

Intel 8086: Instructions

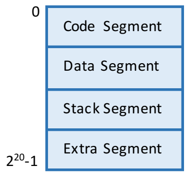
- Data transfer
- Arithmetic, logic
- Branch
- Subroutines
- **Stack Management**
- Flag Manipulation
- **String Manipulation**
- Input/Output
- Floating Point (with Floating Point Co-processor 8087)

Intel 8086: Registers

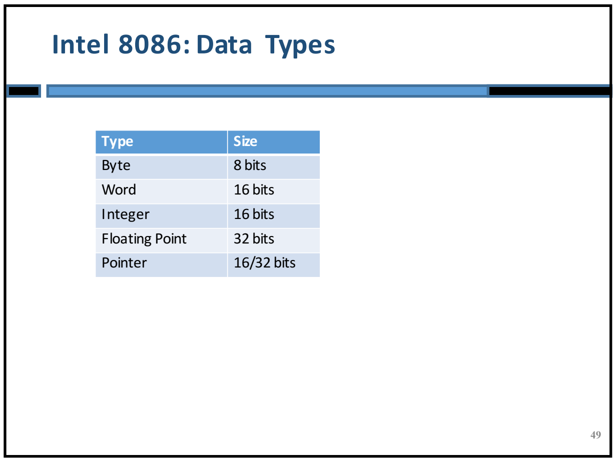
- 4 16-bit main "general purpose" registers
  - *AX (AH|AL), BX (BH|BL), CX (CH|CL), DX (DH|DL)*
- 4 Index registers
  - *SI, DI, BP, SP*
- 4 Segment registers
  - *CS, DS, SS, ES*
- IP – Instruction Pointer
- SR - Status Register

Intel 8086: Memory

- 2^{20} 16-bit cells divided in four 1M word segments

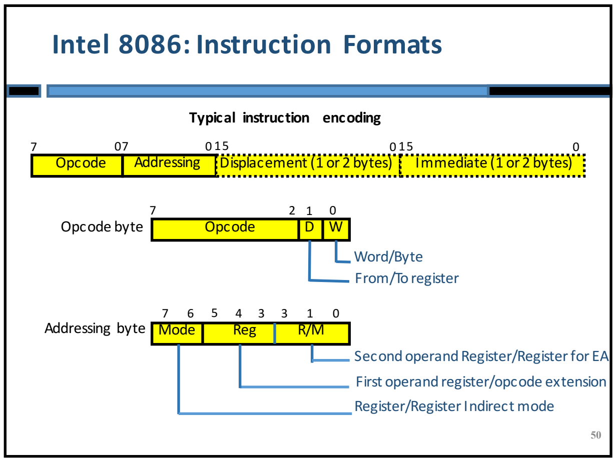


- Instructions and data can be stored starting at any address
- Little Endian format



Intel 8086: Data Types

Type	Size
Byte	8 bits
Word	16 bits
Integer	16 bits
Floating Point	32 bits
Pointer	16/32 bits



Intel 8086: Instruction Formats

Typical instruction encoding

7 07 015 015 0

Opcode | Addressing | Displacement (1 or 2 bytes) | Immediate (1 or 2 bytes)

Opcode byte: 7 Opcode 2 | 1 D | 0 W

W: Word/Byte

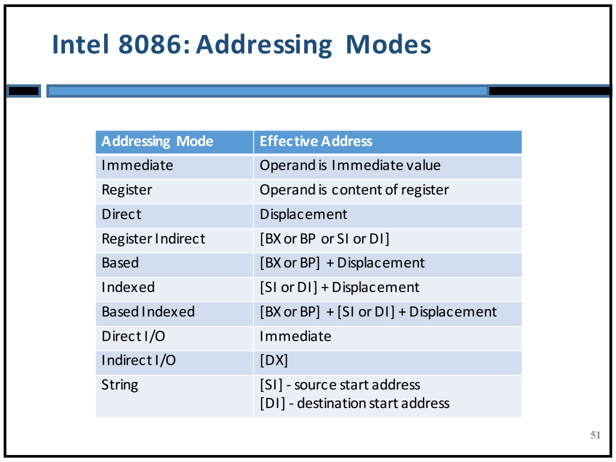
D: From/To register

Addressing byte: 7 6 Mode | 5 4 3 Reg | 3 1 0 R/M

R/M: Second operand Register/Register for EA

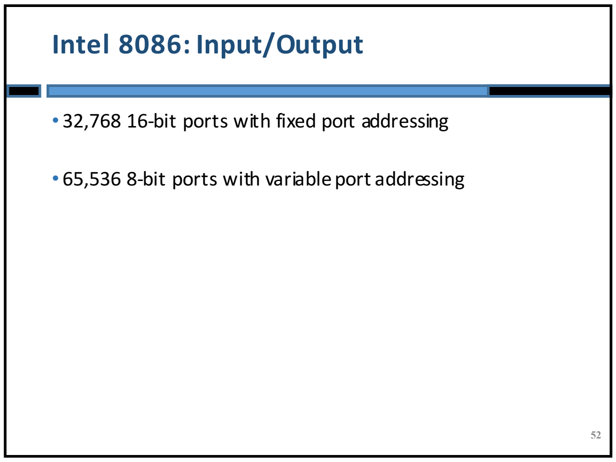
Reg: First operand register/opcode extension

Mode: Register/Register Indirect mode



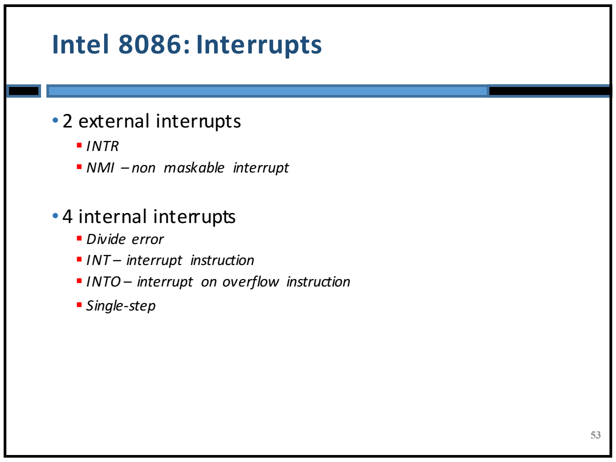
Intel 8086: Addressing Modes

Addressing Mode	Effective Address
Immediate	Operand is Immediate value
Register	Operand is content of register
Direct	Displacement
Register Indirect	[BX or BP or SI or DI]
Based	[BX or BP] + Displacement
Indexed	[SI or DI] + Displacement
Based Indexed	[BX or BP] + [SI or DI] + Displacement
Direct I/O	Immediate
Indirect I/O	[DX]
String	[SI] - source start address [DI] - destination start address



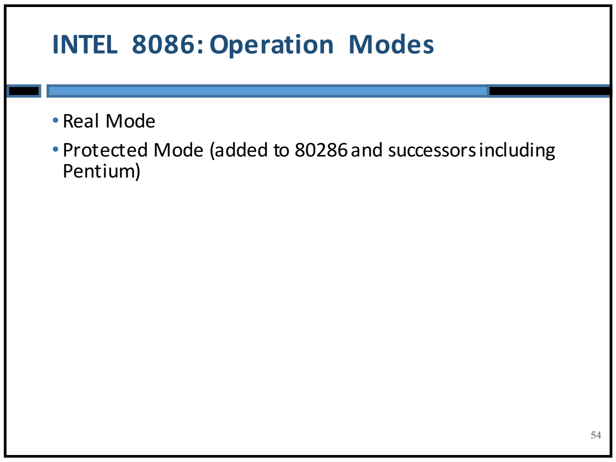
Intel 8086: Input/Output

- 32,768 16-bit ports with fixed port addressing
- 65,536 8-bit ports with variable port addressing



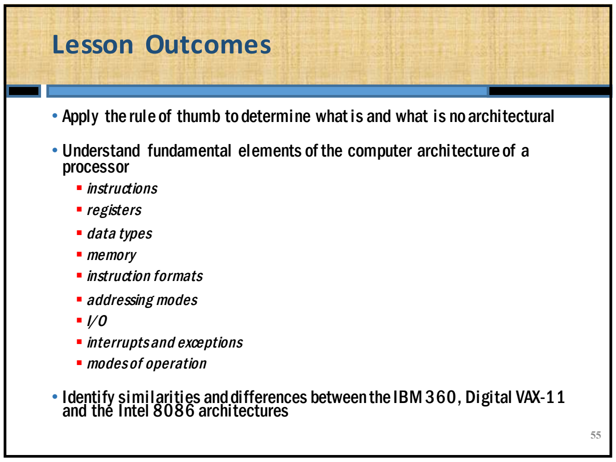
Intel 8086: Interrupts

- 2 external interrupts
  - *INTR*
  - *NMI – non maskable interrupt*
- 4 internal interrupts
  - *Divide error*
  - *INT – interrupt instruction*
  - *INTO – interrupt on overflow instruction*
  - *Single-step*



INTEL 8086: Operation Modes

- Real Mode
- Protected Mode (added to 80286 and successors including Pentium)



Lesson Outcomes

- Apply the rule of thumb to determine what is and what is no architectural
- Understand fundamental elements of the computer architecture of a processor
  - *instructions*
  - *registers*
  - *data types*
  - *memory*
  - *instruction formats*
  - *addressing modes*
  - *I/O*
  - *interrupts and exceptions*
  - *modes of operation*
- Identify similarities and differences between the IBM 360, Digital VAX-11 and the Intel 8086 architectures