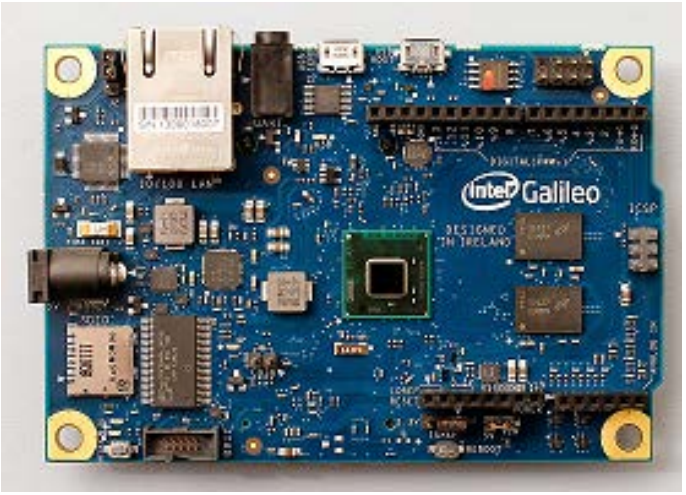
# Embedded Systems Programming

## x86 Memory and Interrupt (Module 8)

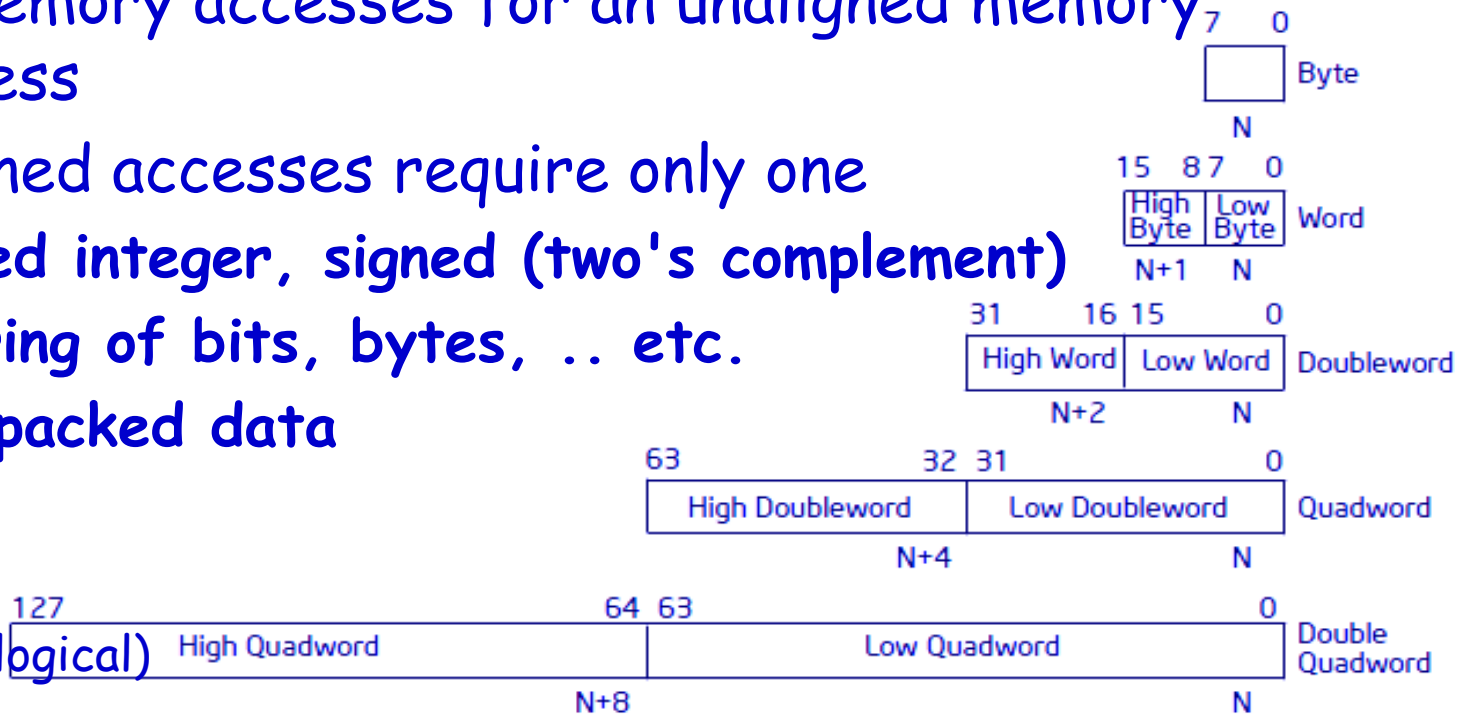
*Yann-Hang Lee*
*Arizona State University*
*yhlee@asu.edu*
*(480) 727-7507*

*Summer 2014*

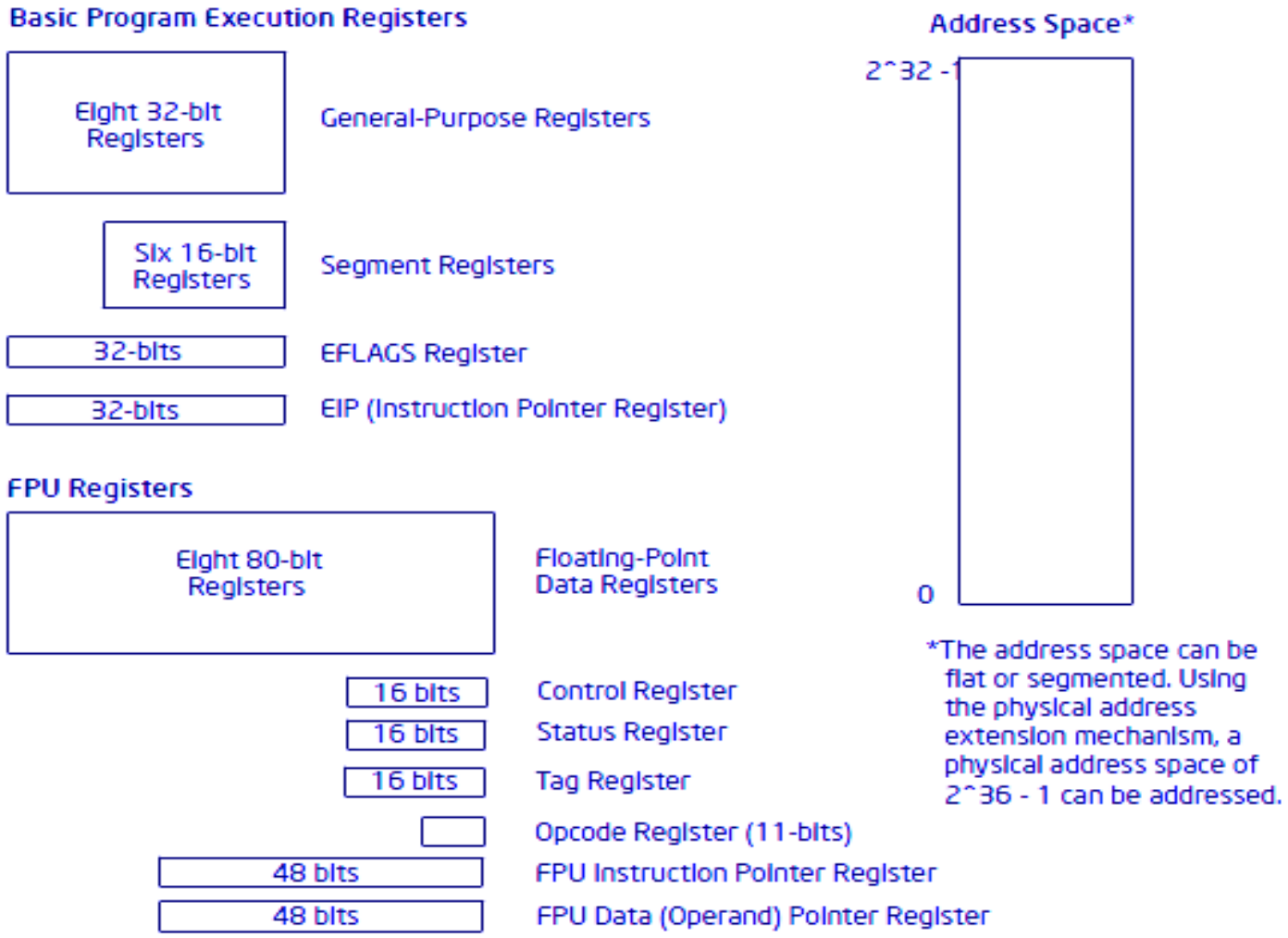


# X86 ISA Data Representations

- Little-endian byte ordering in memory
- Words, doublewords, and quadwords do not need to be aligned in memory on natural boundaries.
  - 2 memory accesses for an unaligned memory access
  - aligned accesses require only one
- Unsigned integer, signed (two's complement)
- FP, string of bits, bytes, .. etc.
- SIMD packed data
- Pointer
  - Near
  - Far (logical)



# Programmer's model

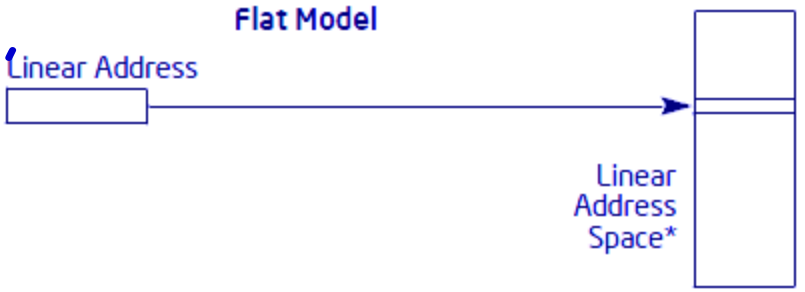


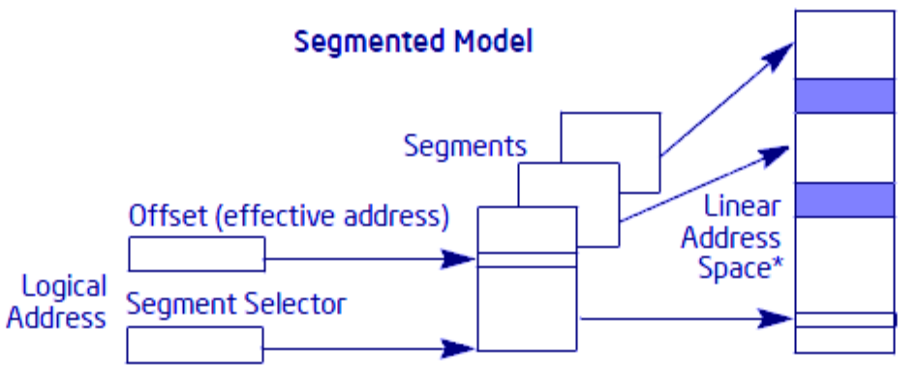
# Modes of Operation

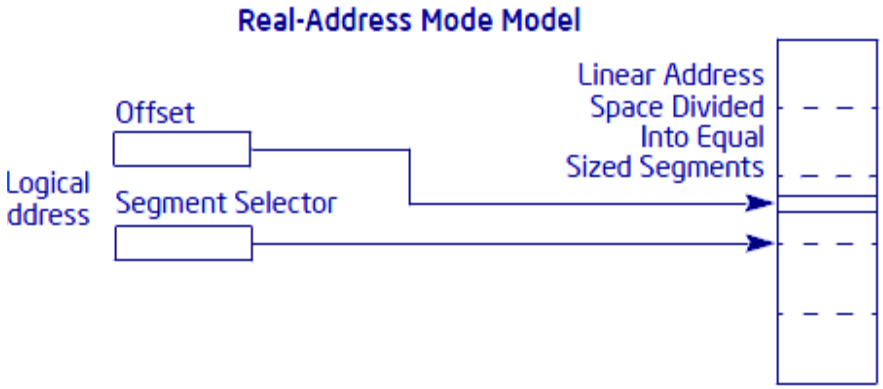
- **Protected mode (32 bits address)**
  - native mode (Windows, Linux), full features, separate memory
  - virtual-8086 mode
- **Real-address mode (20 bits address)**
  - the programming environment of the Intel 8086 processor with extensions
  - native MS-DOS
- **System management mode**
  - power management, system security, diagnostics
- **IA-32e (Intel 64 architecture)**
  - Compatibility mode – similar to 32-bit protected mode
  - 64-bit mode –
    - 16 64-bit general purpose registers
    - default address size is 64 bits and its default operand size is 32 bits.

# Memory Model

- Flat memory model – a single, continuous linear address space of $2^{32}$ bytes
- Segmented model – a logical address consisting of a segment selector and an offset
- Real-address mode – for 8086,
  - 16 segments of 64K
- Linear address space → (paging) physical space

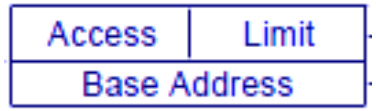


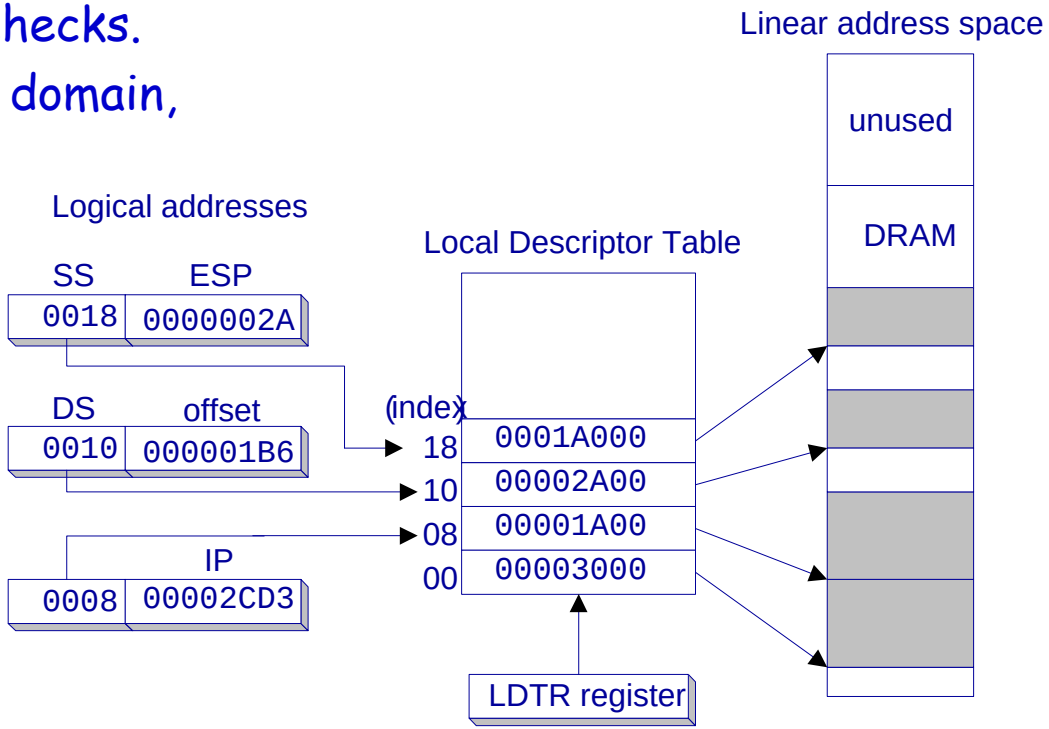



# Protected Mode Memory Management

- Use segment descriptor to protect memory accesses
- Each program has a descriptor table to map segments
  - allow shared segments
- Memory access checks
  - Limit, type, privilege level checks.
  - Restrictions of addressable domain, procedure entry-points, and instruction set.



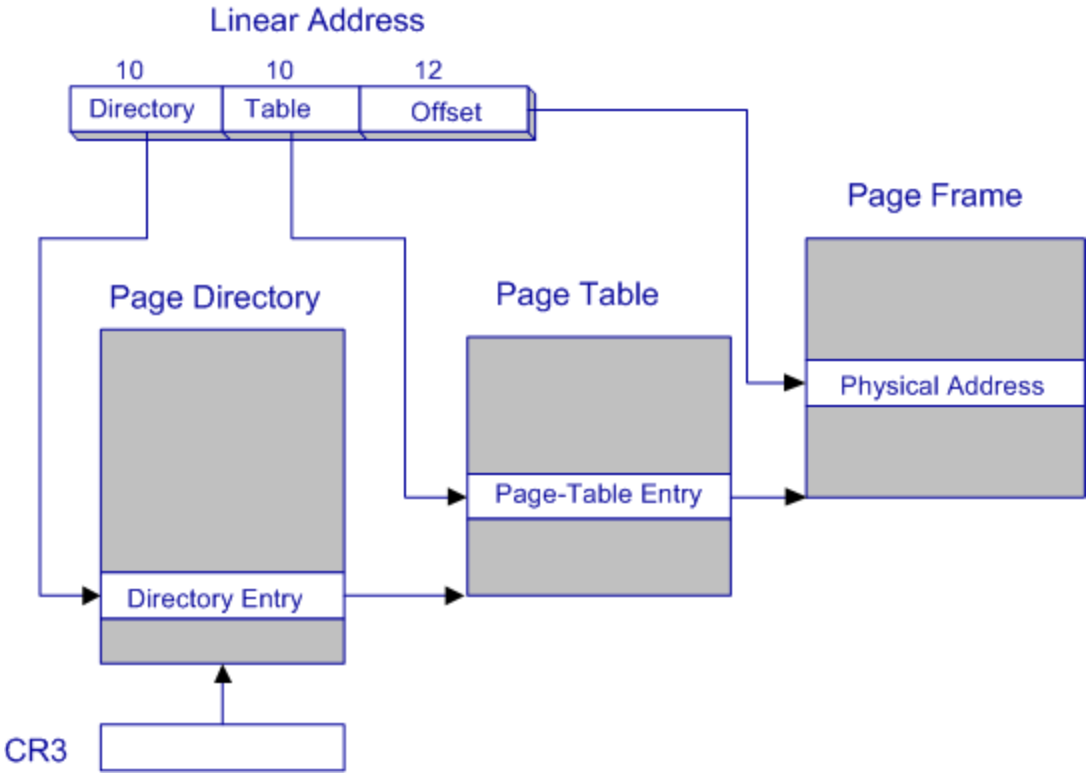



# Virtual Memory and Paging

- **Virtual memory**
  - uses disk as part of the memory, thus allowing sum of all programs can be larger than physical memory
  - Only part of a program must be kept in memory, while the remaining parts are kept on disk.
- **The memory used by the program is divided into small units called pages (4096-byte).**
  - OS maintains page directory and page tables
  - Page translation: CPU converts the linear address into a physical address
  - Page fault: occurs when a needed page is not in memory, and the CPU interrupts the program
- **Virtual memory manager (VMM) – OS utility that manages the loading and unloading of pages**

# Page Translation

- A linear address is divided into a page directory field, page table field, and page frame offset.
- The CPU uses all three to calculate the physical address.



# Interrupt and Exception

- **Interrupt**
  - an asynchronous event that is typically triggered by an I/O device.
- **Exception**
  - a synchronous event that is generated when the processor detects one or more predefined conditions while executing an instruction.
  - three classes of exceptions: faults, traps, and aborts.
- **18 predefined interrupts and exceptions and 224 user defined interrupts**
- **Access handler procedures through entries in the interrupt descriptor table (IDT)**
  - A call to a handler procedure is similar to a procedure call to another protection level

# Interrupt and Exception

- **Interrupt vector references**
  - an interrupt gate (interrupt enable (IF) flag in the EFLAGS register is cleared)
  - a trap gate
- **Gate contains**
  - access rights information
  - segment selector for the code segment of the handler procedure
  - an offset into the code segment to entry point of the handler procedure

| Vector No. | Mnemonic | Description | Source |
|---|---|---|---|
| 0 | #DE | Divide Error | DIV and IDIV instructions. |
| 1 | #DB | Debug | Any code or data reference. |
| 2 | | NMI Interrupt | Non-maskable external interrupt. |
| 3 | #BP | Breakpoint | INT 3 instruction. |
| 4 | #OF | Overflow | INTO instruction. |
| 5 | #BR | BOUND Range Exceeded | BOUND instruction. |
| 6 | #UD | Invalid Opcode (UnDefined Opcode) | UD2 instruction or reserved opcode.[1] |
| 7 | #NM | Device Not Available (No Math Coprocessor) | Floating-point or WAIT/FWAIT instruction. |
| 8 | #DF | Double Fault | Any instruction that can generate an exception, an NMI, or an INTR. |
| 9 | #MF | CoProcessor Segment Overrun (reserved) | Floating-point instruction.[2] |
| 10 | #TS | Invalid TSS | Task switch or TSS access. |
| 11 | #NP | Segment Not Present | Loading segment registers or accessing system segments. |
| 12 | #SS | Stack Segment Fault | Stack operations and SS register loads. |
| 13 | #GP | General Protection | Any memory reference and other protection checks. |
| 14 | #PF | Page Fault | Any memory reference. |
| 15 | | Reserved | |
| 16 | #MF | Floating-Point Error (Math Fault) | Floating-point or WAIT/FWAIT instruction. |
| 17 | #AC | Alignment Check | Any data reference in memory.[3] |
| 18 | #MC | Machine Check | Error codes (if any) and source are model dependent.[4] |
| 19 | #XM | SIMD Floating-Point Exception | SIMD Floating-Point Instruction[5] |
| 20-31 | | Reserved | |
| 32-255 | | Maskable Interrupts | External interrupt from INTR pin or INT *n* instruction. |