EECE416 Microcomputer Fundamentals

Computer Architecture

Dr. Charles Kim

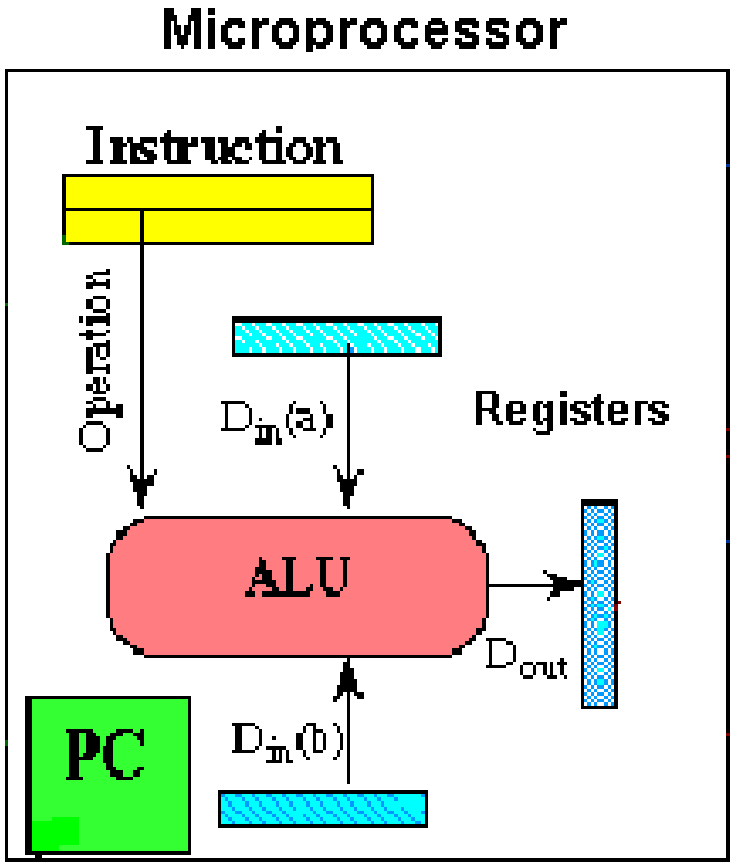
Howard University

"Computer Architecture"

- Computer Architecture
  - Art of selecting and interconnecting hardware components to create functional unit (or computer)
  - 2 points of view
    - **Instruction Set architecture (ISA):**
      - the code that a CPU reads and acts upon. It is the machine language (or assembly language), including the instruction set, word size, memory address modes, processor registers, and address and data formats
      - Interface between H/W and S/W
      - programmers' point of view
    - **Microarchitecture** (or computer organization):
      - describes the data paths, data processing elements and data storage elements, size of cache, and describes how they should implement the ISA
      - Optimization
      - Power Management
      - system designers' point of view.
  - Analogy: House (rooms) – views of builders and residents

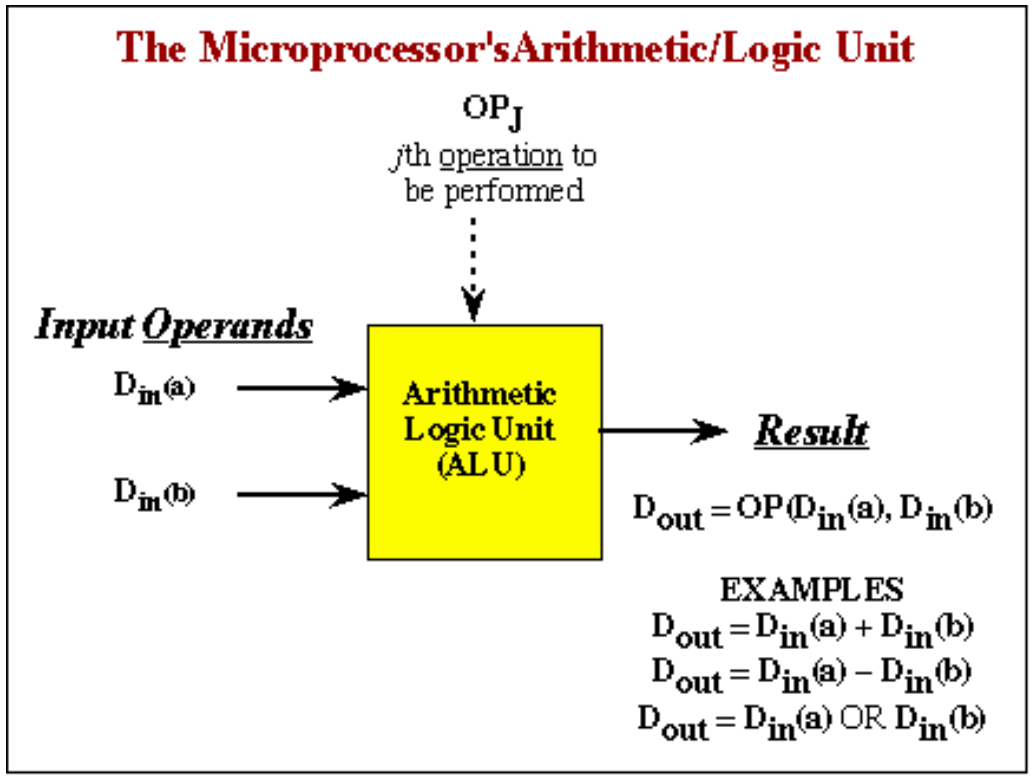
Micro-Architecture

- **Computer System**
  - CPU (with PC, Register, SR) + Memory
- **Computer Architecture:**
  - "conceptual design and fundamental operational structure of a computer system"
  - "blueprint and functional description of requirements and design implementations of a computer"
  - focusing on the way the CPU performs and accesses memory.

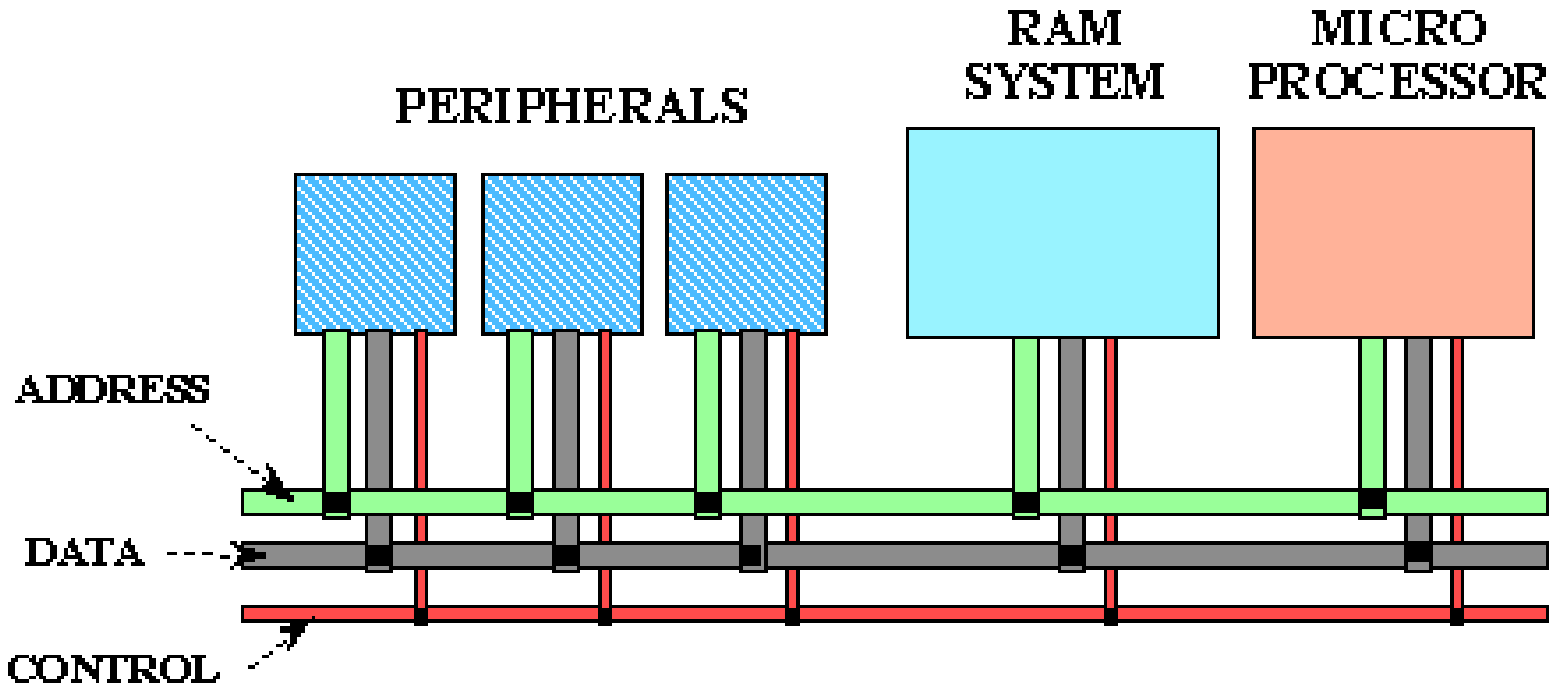


Micro-Architecture

- **ALU (Arithmetic Logic Unit)**
- **Fundamental building block of CPU**
- **Binary Full Adder**

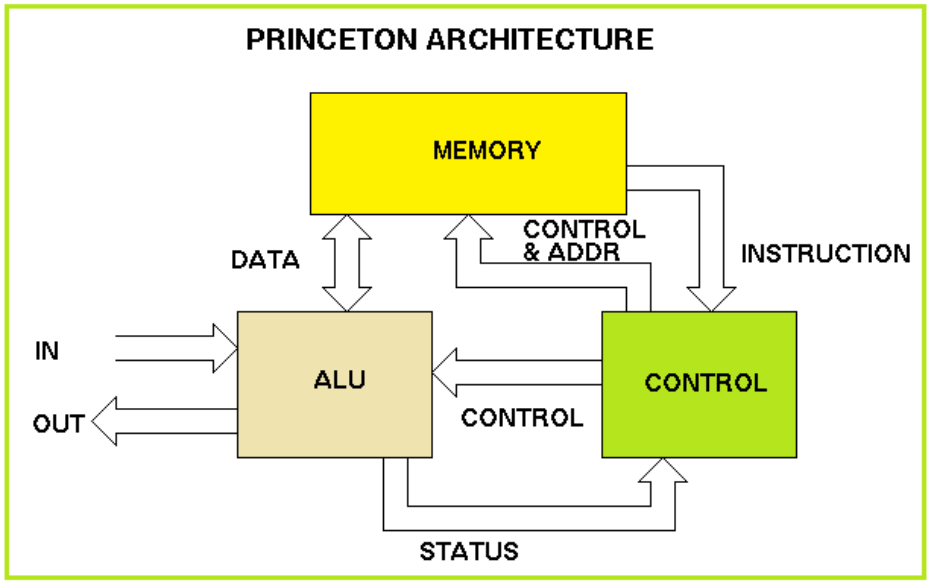


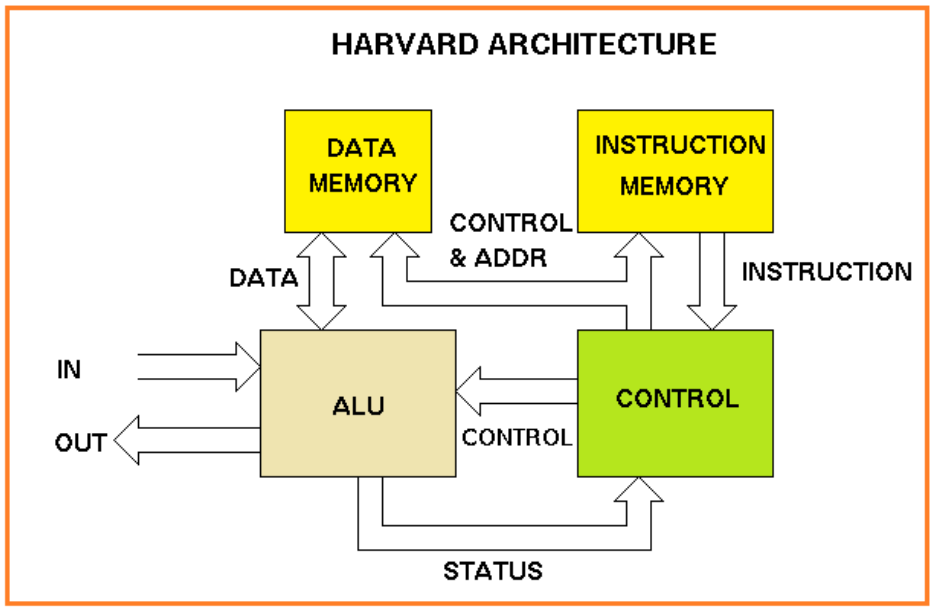
Microprocessor Bus



Architecture by CPU+MEM organization

- Princeton (or von Neumann) Architecture
  - MEM contains both Instruction and Data
  - Von Neumann Bottleneck – CPU <→ Memory
  - Cache
- Harvard Architecture
  - Data MEM and Instruction MEM
  - Higher Performance – via Pipeline
  - Better for DSP
  - Higher MEM Bandwidth



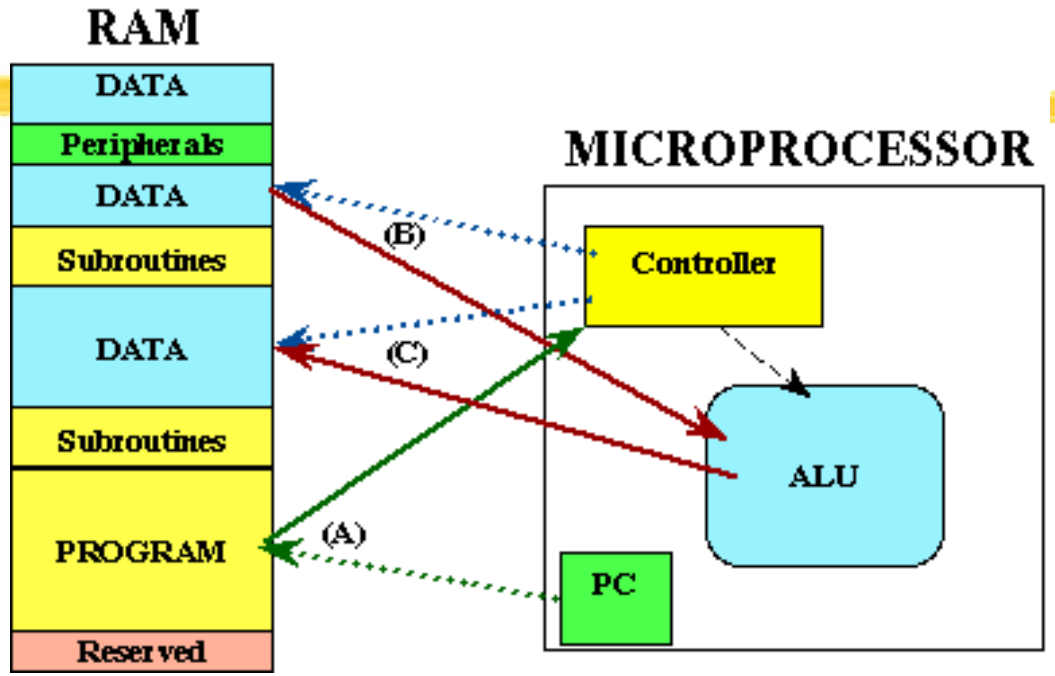


Princeton Architecture

1.**Step (A):** The address for the instruction to be next executed is read into

(**Step (B):** The controller "decodes" the instruction

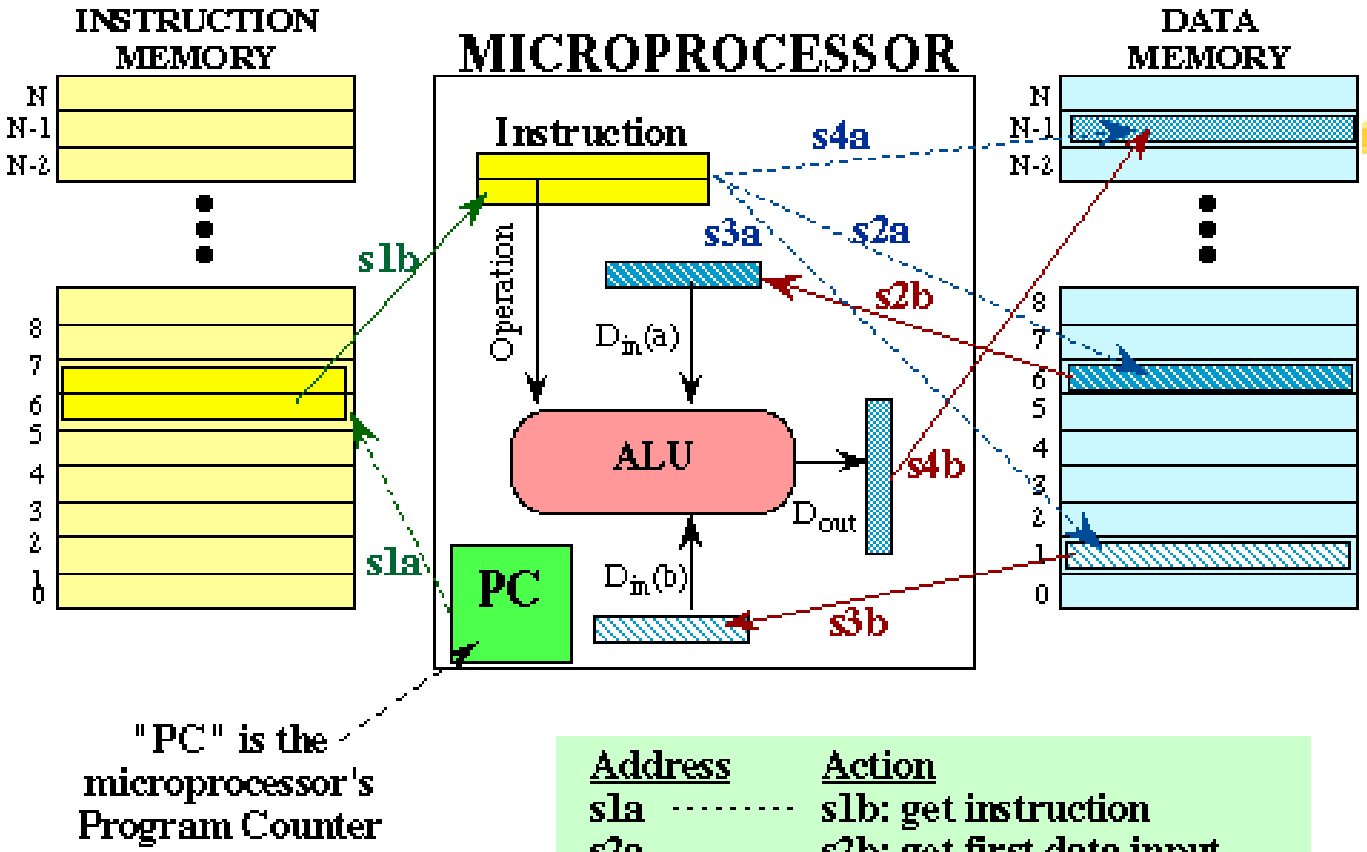
3.**Step (C)**: Following completion of the instruction, the controller provides the address, to the memory unit, at which the data result generated by the operation will be stored.



- CPU can be either reading an instruction or reading/writing data from/to the memory.
- Both cannot occur at the same time since the instructions and data use the same bus system

Harvard Architecture

- 1. CPU can both read an instruction and perform a data memory access at the same time.
- 2. Faster for a given circuit complexity because instruction fetches and data access do not contend for a single memory pathway.



Architecture by Instructions and Executions

- CISC (Complex Instruction Set Computer)
  - Variety of instructions for complex tasks directly to hardware
  - Easy to translate high-level language to assembly
  - Complex Hardware
  - Instructions of varying length
- RISC (Reduced Instruction Set Computer)
  - Fewer and simpler instructions
  - Each instruction takes the same amount of time
  - Less complex hardware
  - High performance microprocessors
  - Pipelined instruction execution (several instructions are executed in parallel)

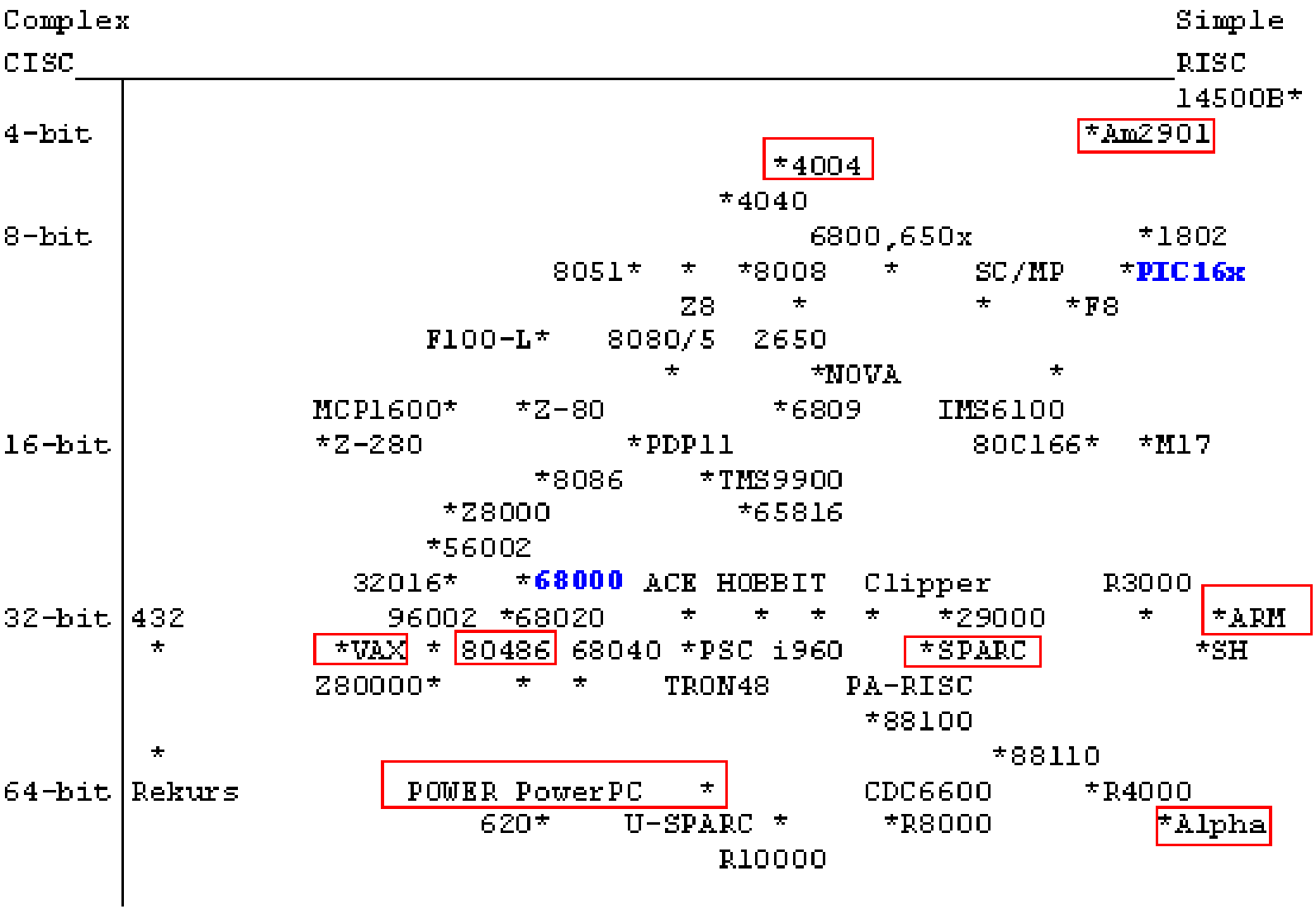
CISC

- Architecture of prior to mid-1980's
  - IBM390, Motorola 680x0, Intel80x86
- Basic Fetch-Execute sequence to support a large number of complex instructions
- Complex decoding procedures
- Complex control unit
- One instruction achieves a complex task

RISC

- Favorable changes for RISC
  - **Caches** to speed instruction fetches
  - Dramatic memory size increases/cost decreases
  - Better ***pipelining***
  - Advanced optimizing compilers
- Characteristics of RISC
  - Instructions are of a uniform length
  - Increased number of registers to hold frequently used variables (16 - 64 Registers)
  - Central to High Performance Computing

Processor Classification



INTEL VS. ARM ("Advanced RISC Machine")

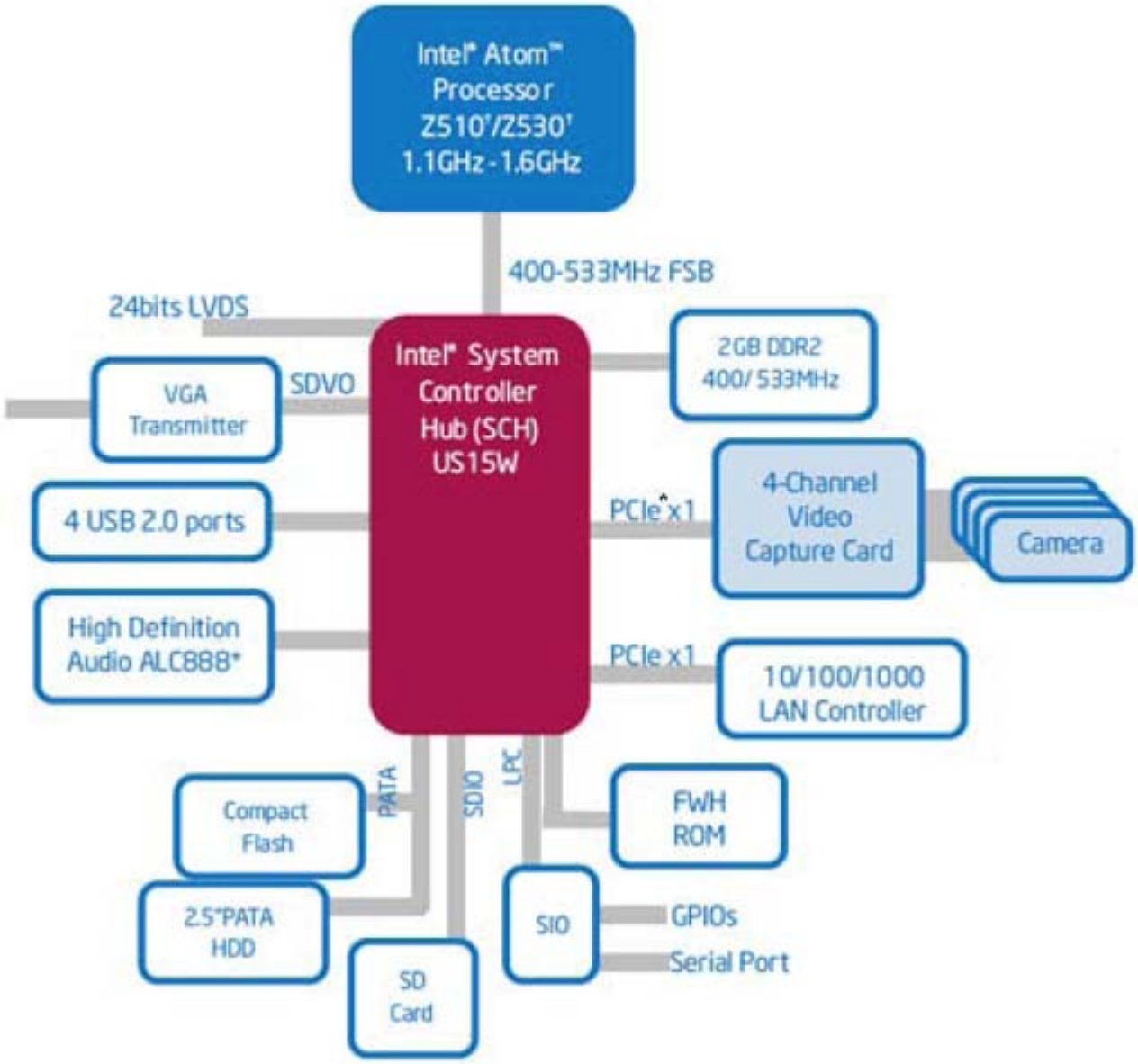
- Next PCs
  - Smart phones (on ARM)
  - Mobile Devices – MP3, Digicam (on ARM)
  - Run on Intel's x86? --- Intel's wish
- ARM revisited
  - No chip hardware – license only (powerful and variety of licensees) → cell phones etc
  - SoC device (CPU + I/O + Peripherals+ Memory + etc)
- INTEL
  - Does not want to License x86 (Lesson from AMD)
  - New approach for SoC: Atom based X86 SoC

Intel Atom

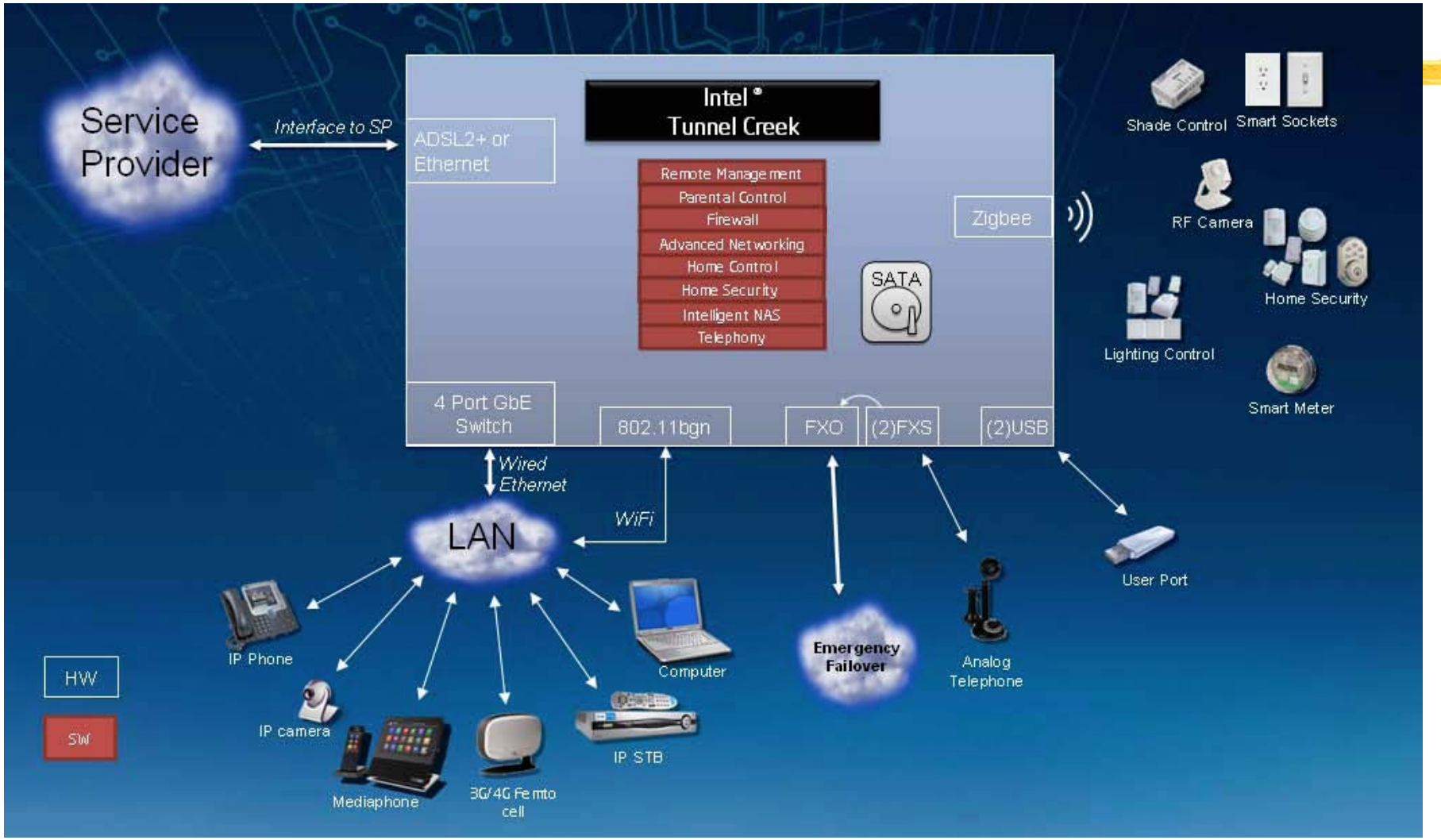
Produced	2008–present
Common manufacturer(s)	Intel
Max. CPU clock	800 MHz to 2 GHz
FSB speeds	400 MHz to 667 MHz
Min. feature size	45nm
Instruction set	x86, x86-64 (not for the N and Z series)
Cores	1, 2
Package(s)	441-ball µFCBGA
Core name(s)	Silverthorne Diamondville

Intel Atom is the brand name for a line of x86 and x86-64 CPUs (or microprocessors) from Intel, designed in 45 nm CMOS and used mainly in Netbooks. The Atom Z series is code-named Silverthorne and the Atom N series is code-named Diamondville. As of June 2009, the most used chips in the Netbook retail market are Z520, Z530, and N270.

Intel Atom Z530



Tunnel Creek



Intel 386 - Brief

- Address: A31- A2
  - BE3 – BE0 (“Byte Enable”)
- Data: D31 – D0
- Registers
- Control Registers

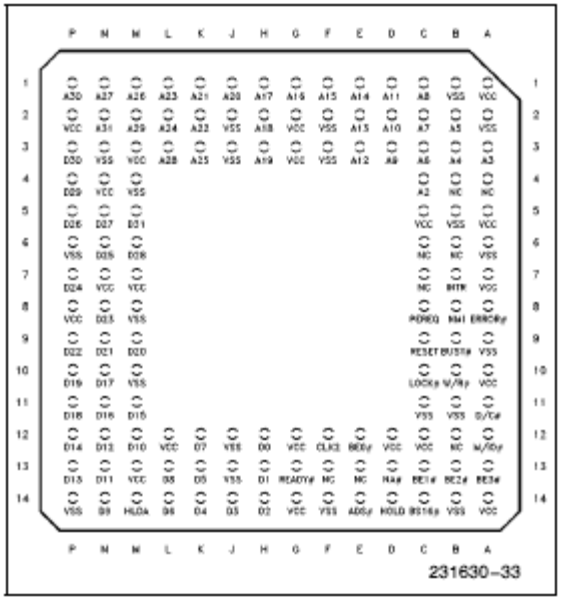


Figure 1-1. Intel386™ DX PGA Pinout—View from Top Side

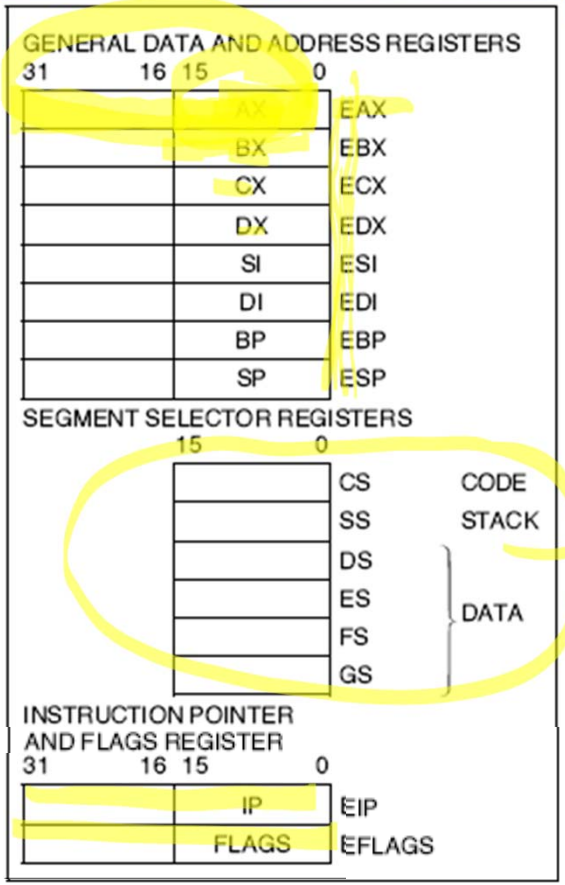
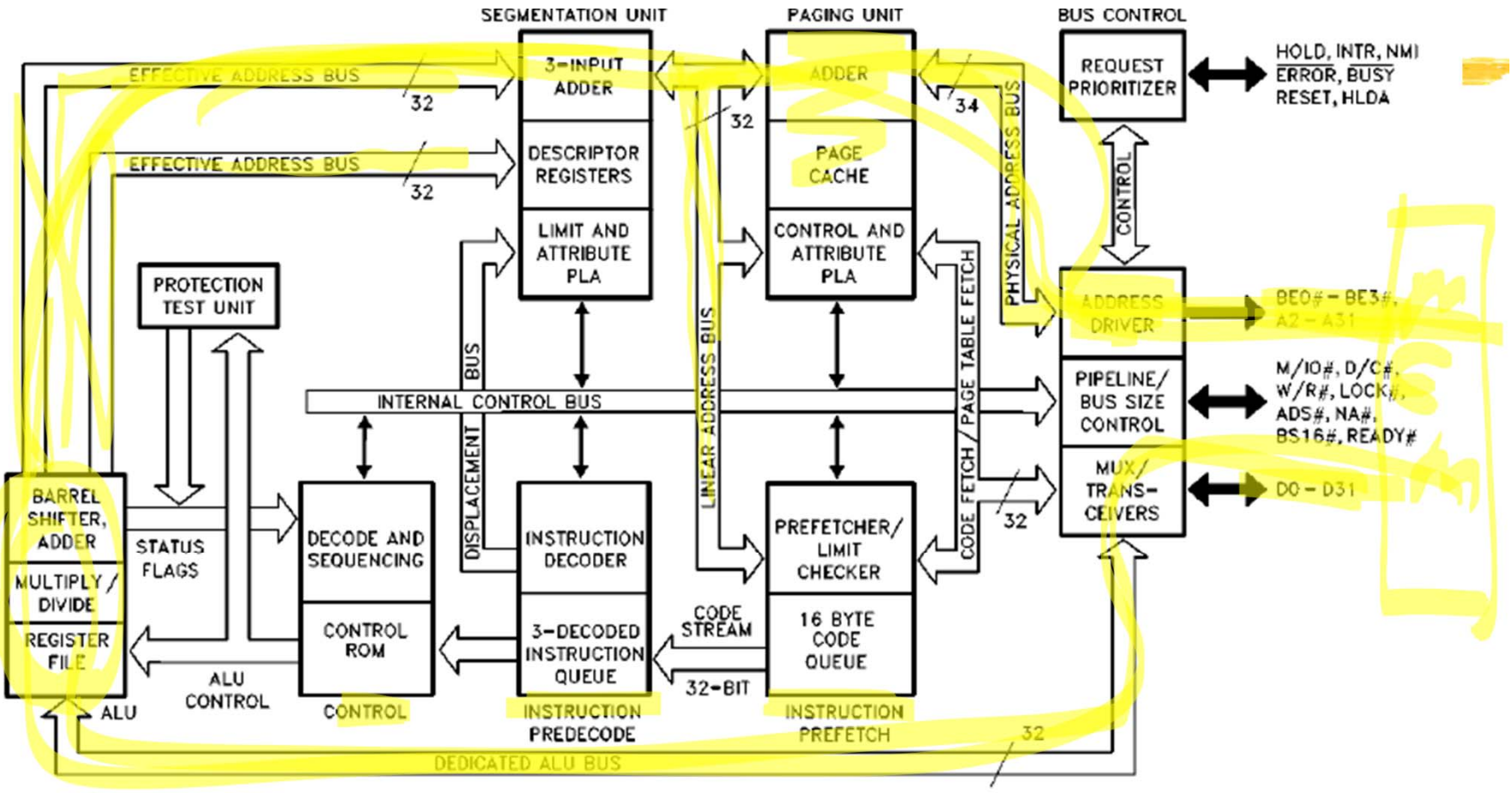


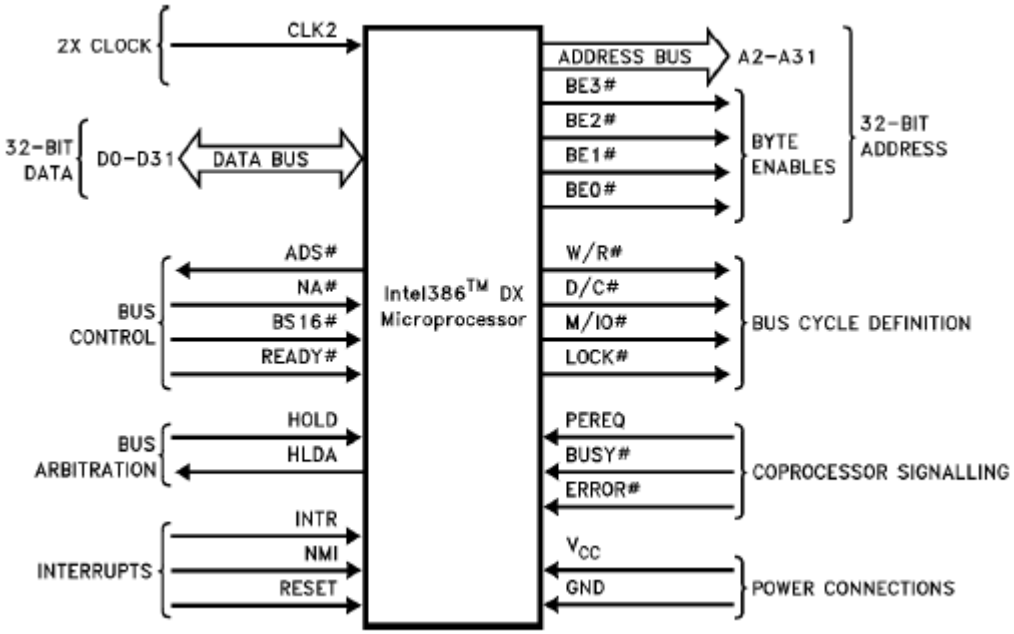
Figure 2-1. Intel386™ DX Base Architecture Registers

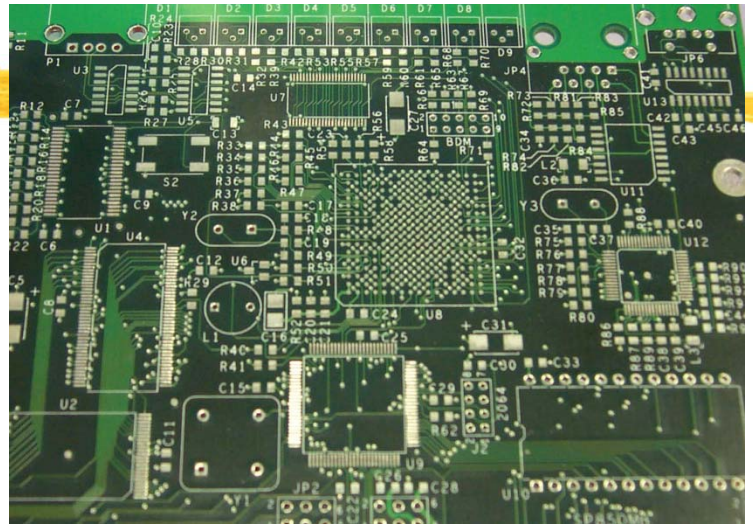
BLOCK Diagram

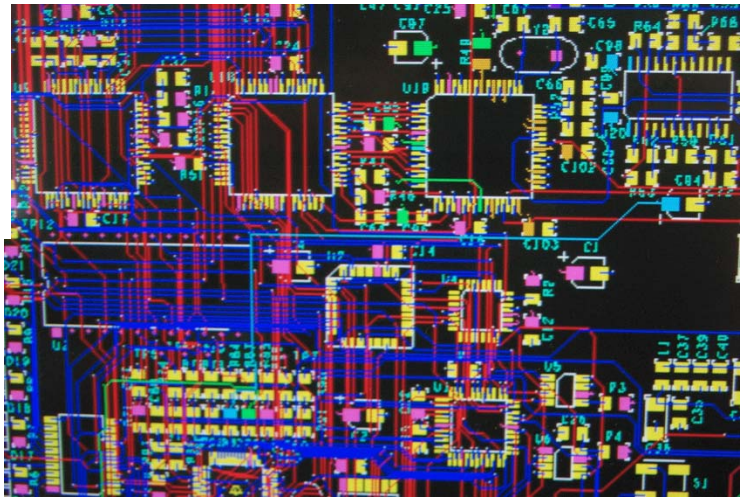


Intel386™ DX Pipelined 32-Bit Microarchitecture

Connecting with Memory, I/O, and Peripherals

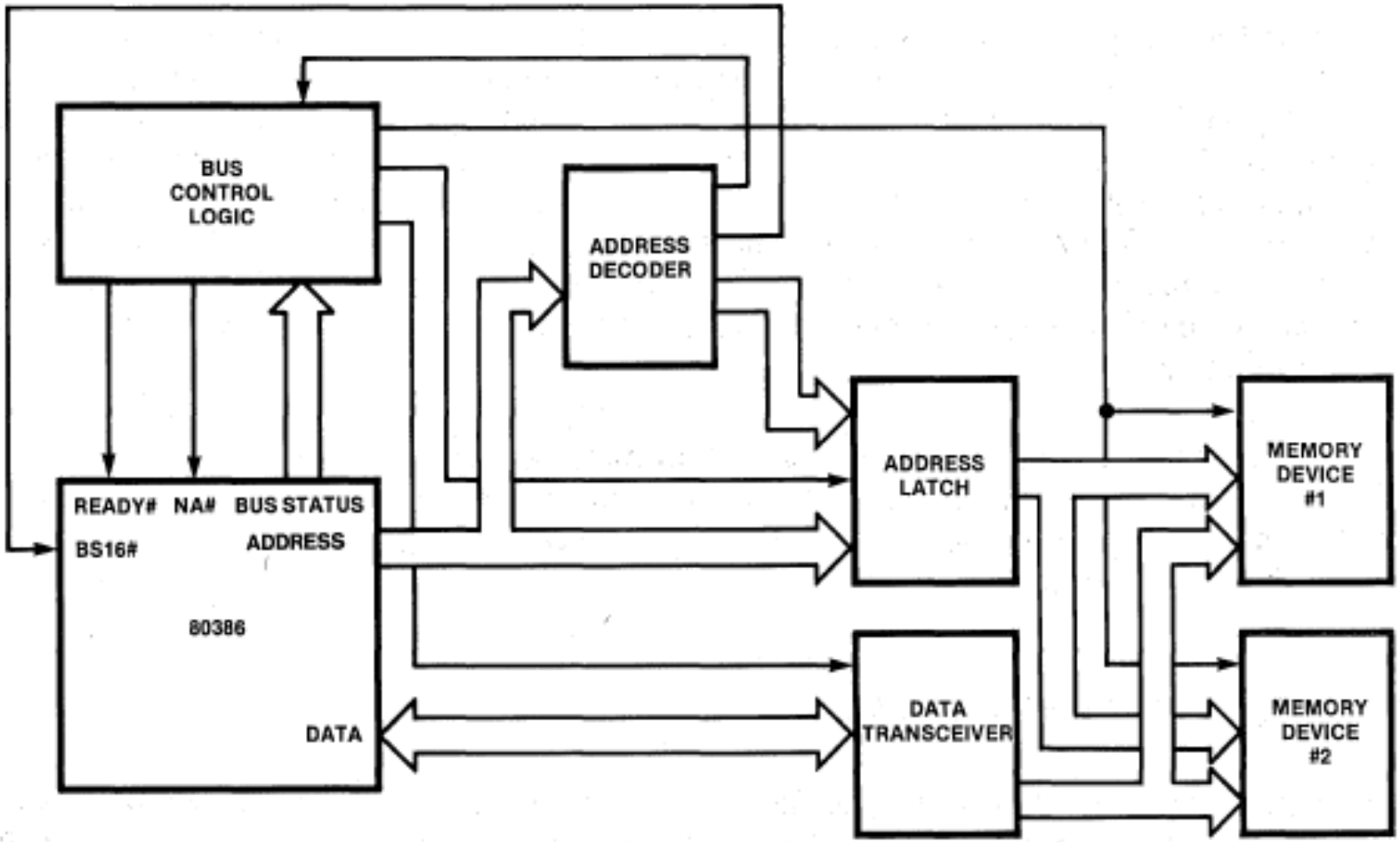






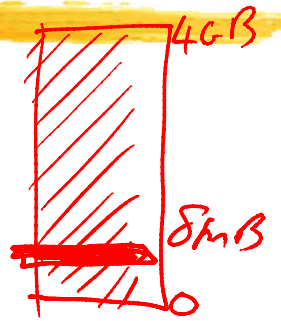
- Single Board Computers
- Processor Boards
- Kits

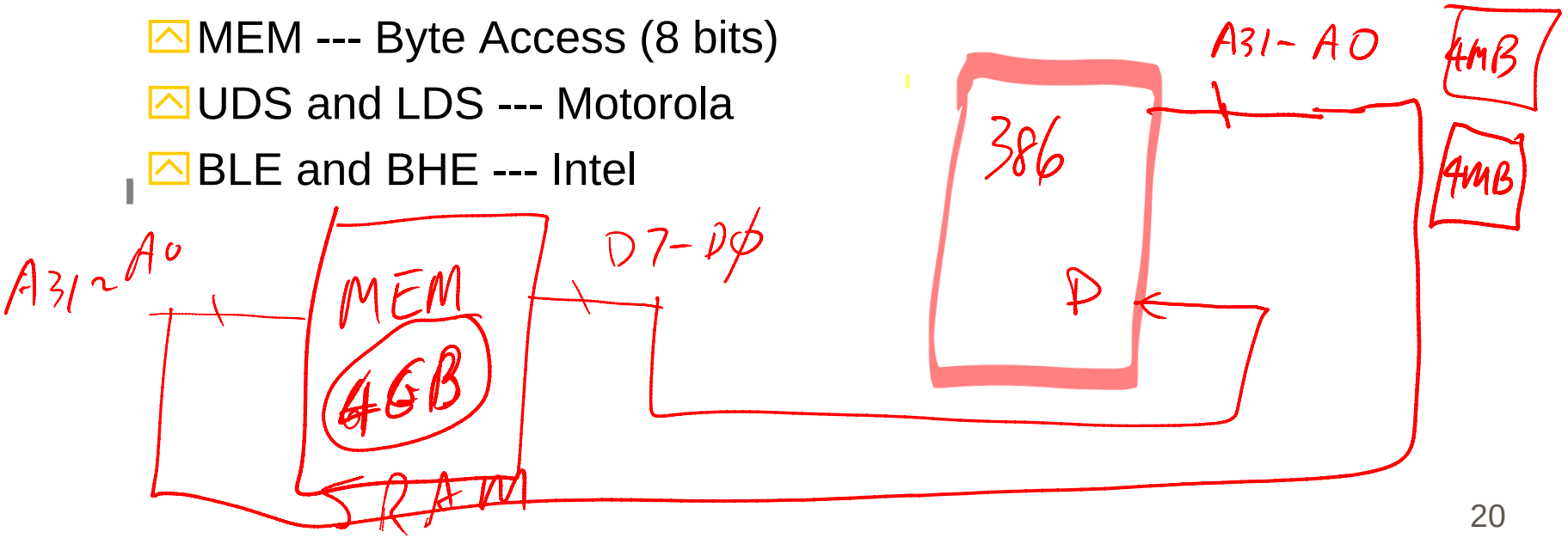
Memory Interface



Memory Interface

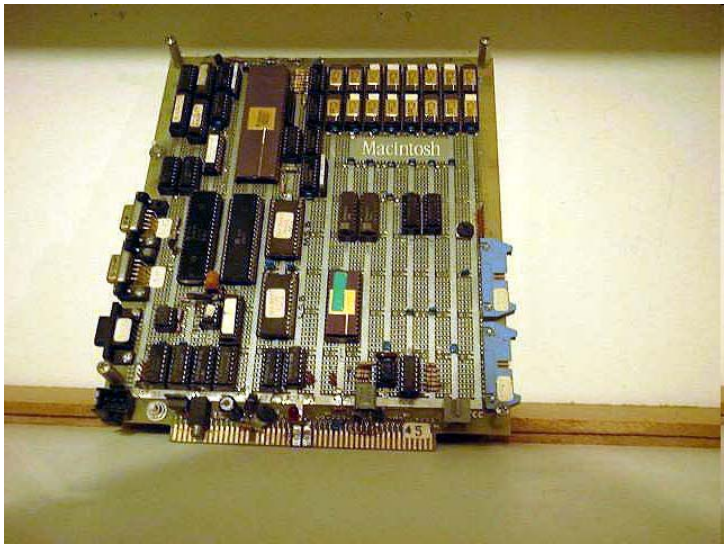
- Interface between a processor and a (pair) of memory (of smaller than the maximum memory space)
- Where do we place the memory in the memory space?
- How to access two MEMs at the same time (for 16-bit Data bus)?
  - MEM --- Byte Access (8 bits)
  - UDS and LDS --- Motorola
  - BLE and BHE --- Intel





Apple Macintosh

- CPU: 8MHz Motorola 68000
- Introduced in 1984
- Memory: 128KB (512KB in later version) RAM, 64KB ROM
- 3.5” 400KB Floppy Disk
- Application: MacWrite and MacPaint
- Mouse
- 9” B&W Monitor
- Keyboard
- Serial Port (DB-9)
- Printer Port
- Addressing: 24-bit



Apple Macintosh Circuit Diagram

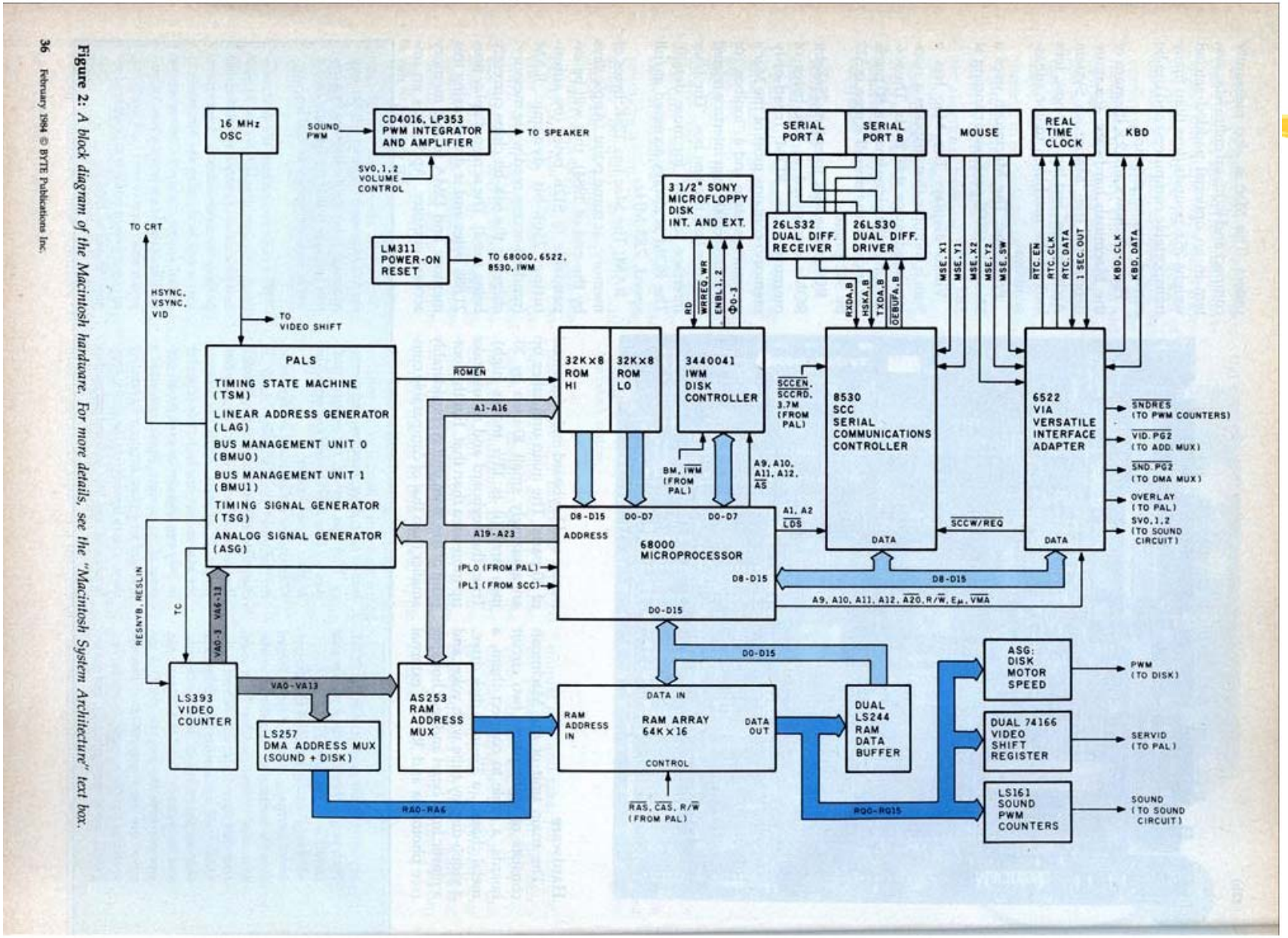


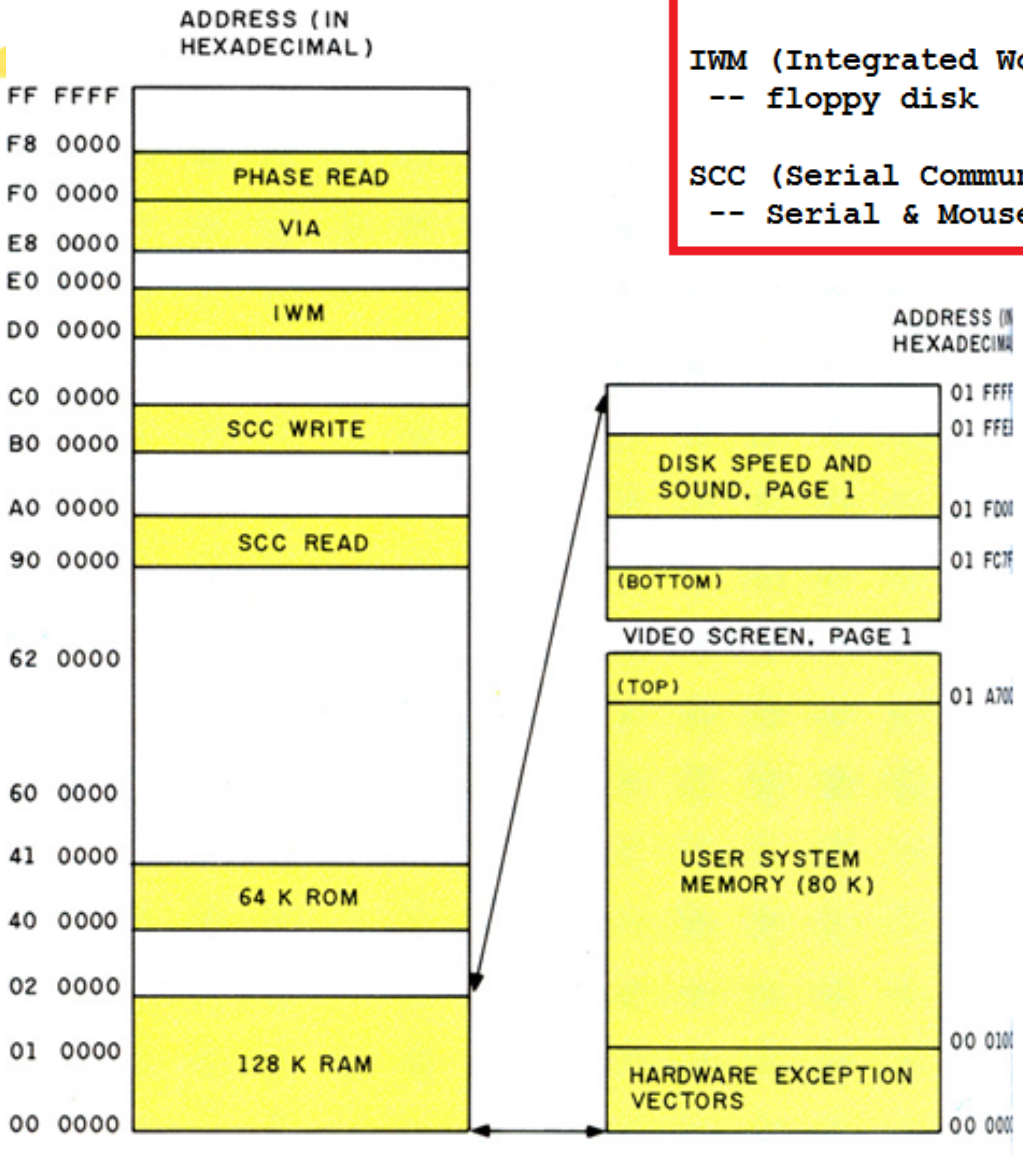
Figure 2: *A block diagram of the Macintosh hardware. For more details, see the "Macintosh System Architecture" text box.*

Memory Map (for Apple Macintosh)

VIA(Versatile Interface Adapter)
 ---general I/O

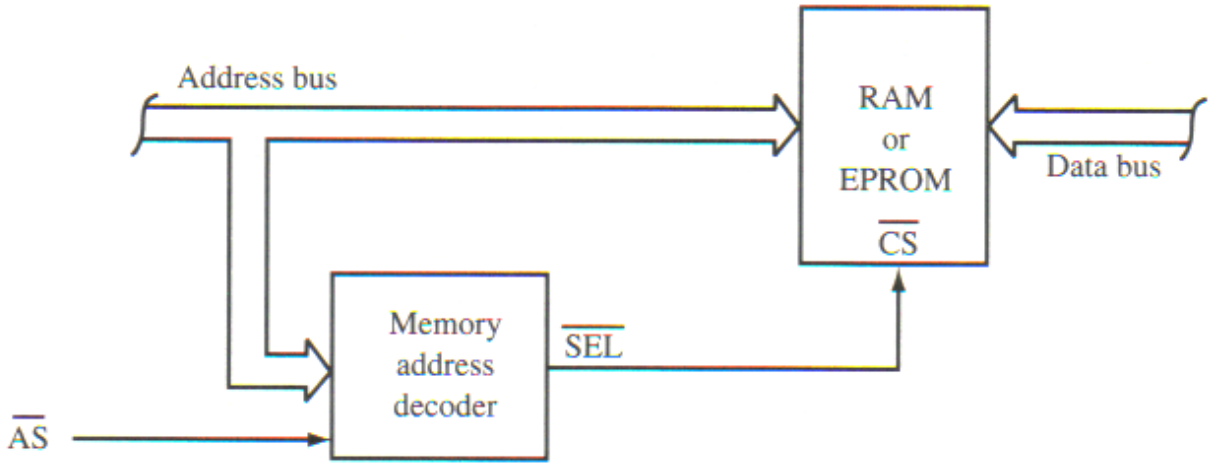
IWM (Integrated Woz Machine)
 -- floppy disk

SCC (Serial Communications Controller)
 -- Serial & Mouse



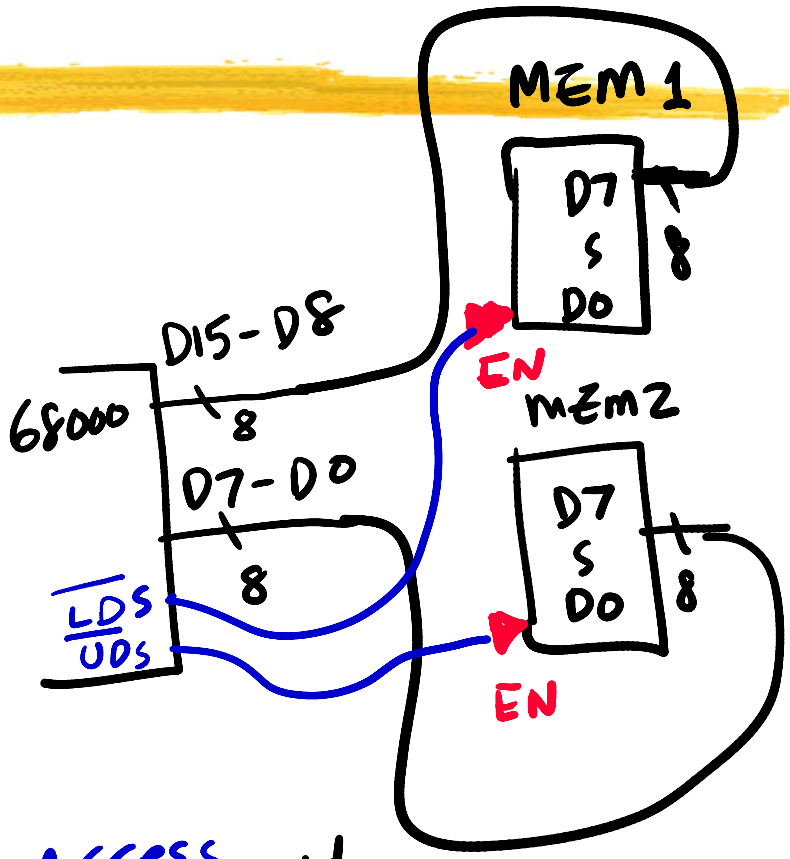
Memory Address Decoding

- How Much Memory?
- How Many Address Lines?
- 1K → 10 lines
- 32K → 15 lines
- 23 ADDR lines → 8MB?

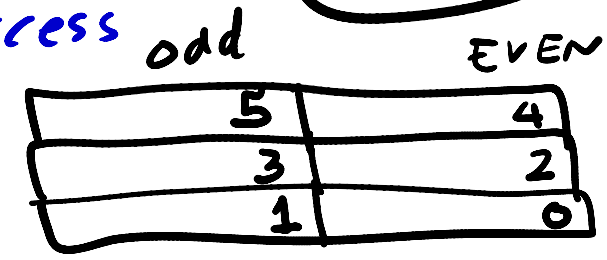


UDS and LDS (for 68000)

- Upper Data Strobe
  - For accessing upper byte of memory
  - D15 – D8 part of CPU
  - D7 – D0 part of memory
- Lower Data Strobe
  - For accessing lower byte of memory
  - D7-D0 part of CPU
  - D7 – D0 part of Memory
- Both together works as A0 line

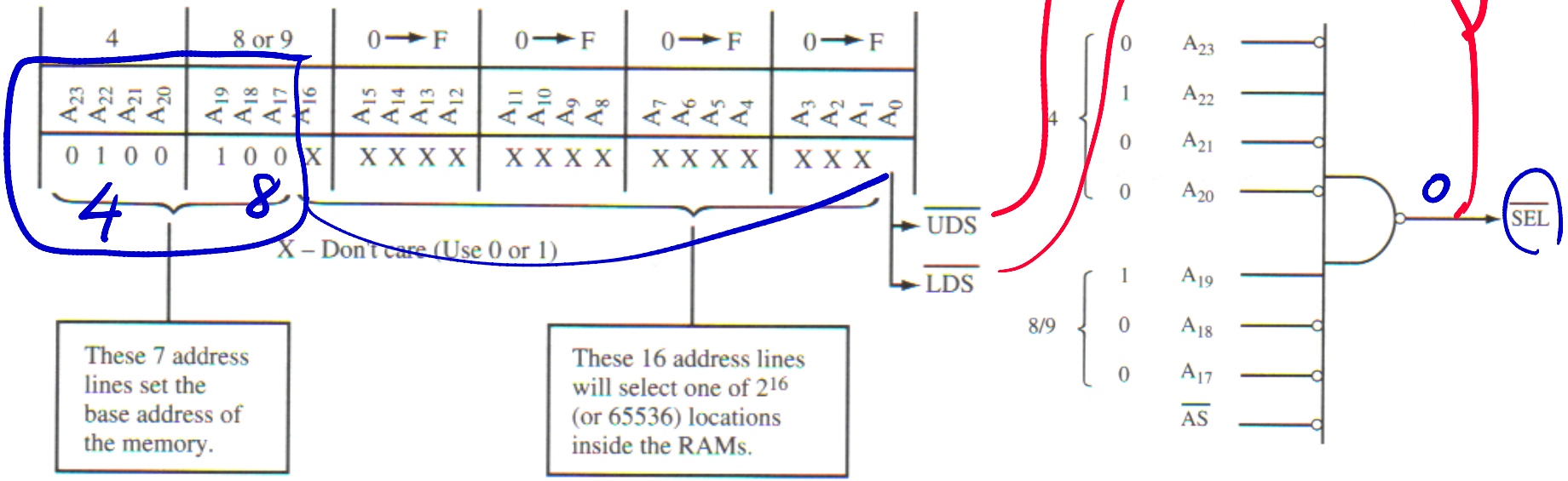


$\overline{LDS}$	$\overline{UDS}$	
0	0	D15 ~ D0
0	1	D15 ~ D8
1	0	D7 ~ D0



Memory Decoding

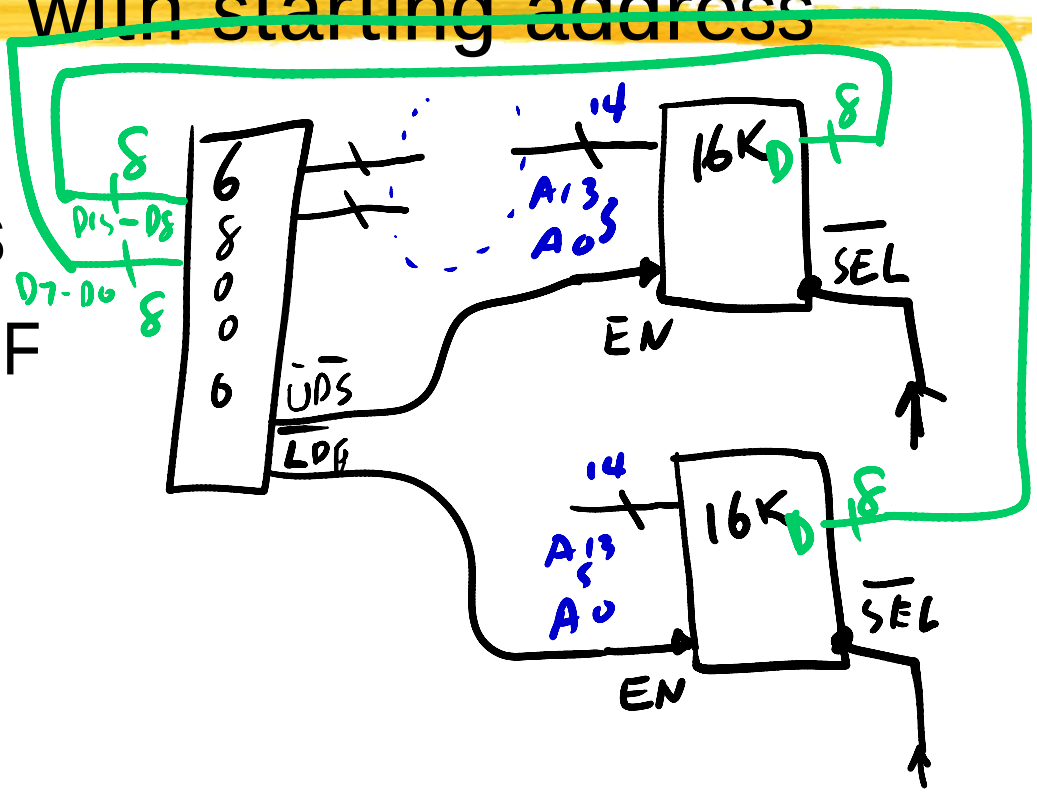
- Q: 64K Word (or 128 KB) of RAM, with it's starting address at $480000
- A: 128KB → 17 lines
  - Range: $480000 - $49FFFF
  - UDS and LDS for A_0 line.



Memory Decoding Example - 1

2 . 16KByte ROMs

- Q: 16K Word ROM with starting address at $300000.
- A: 32KB → 15lines
  - $300000 - $307FFF

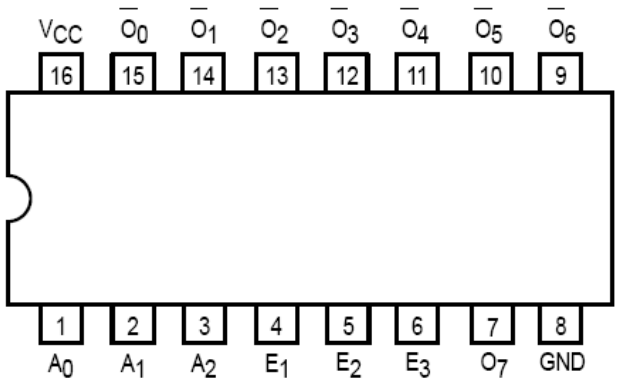


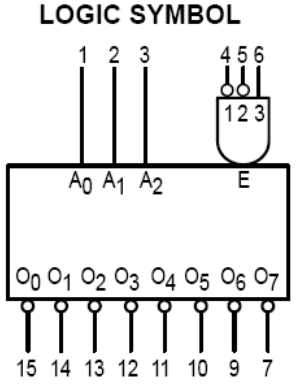
Decoder/Multiplexer

⌘74138

MOTOROLA

1-OF-8 DECODER/ DEMULTIPLEXER

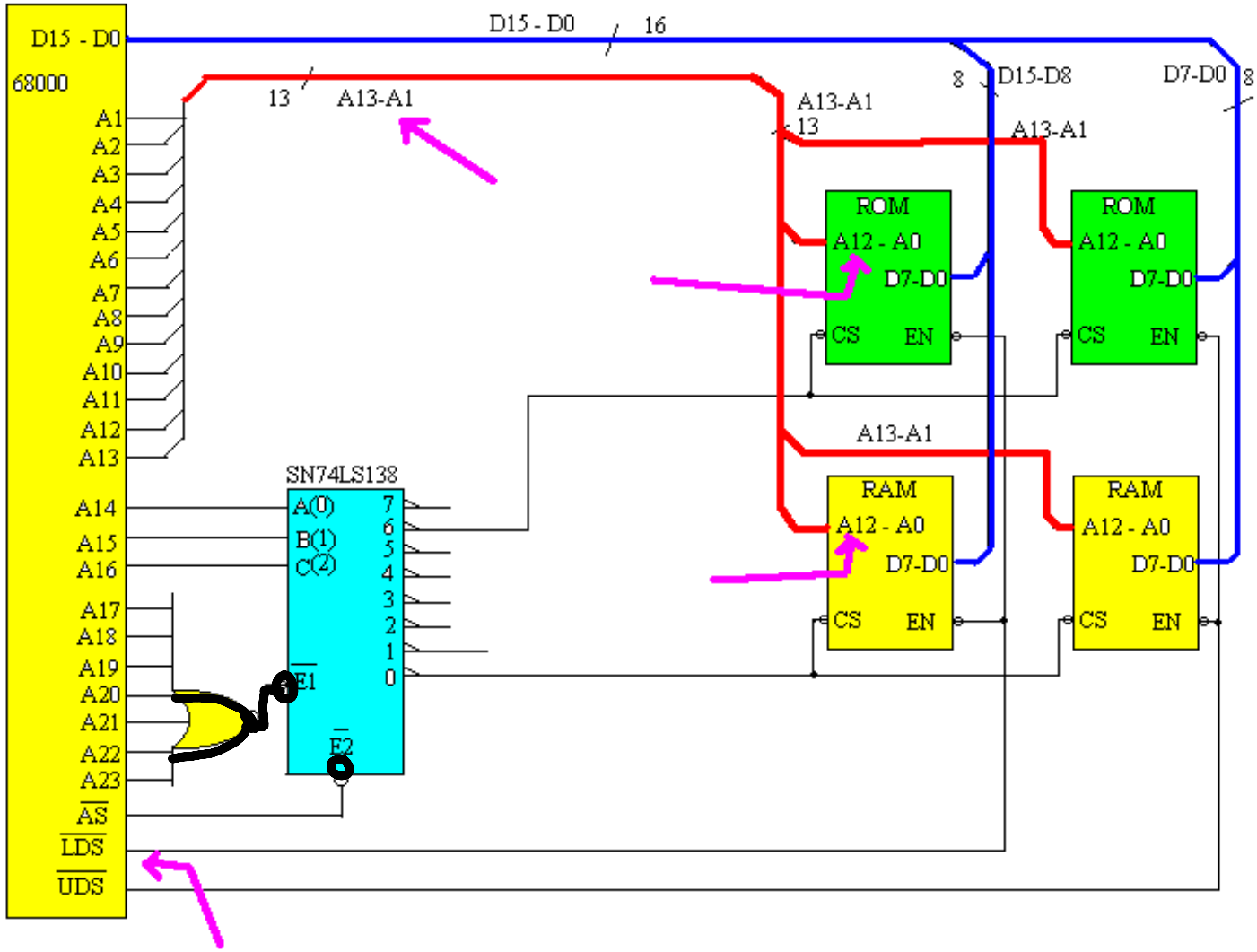




TRUTH TABLE

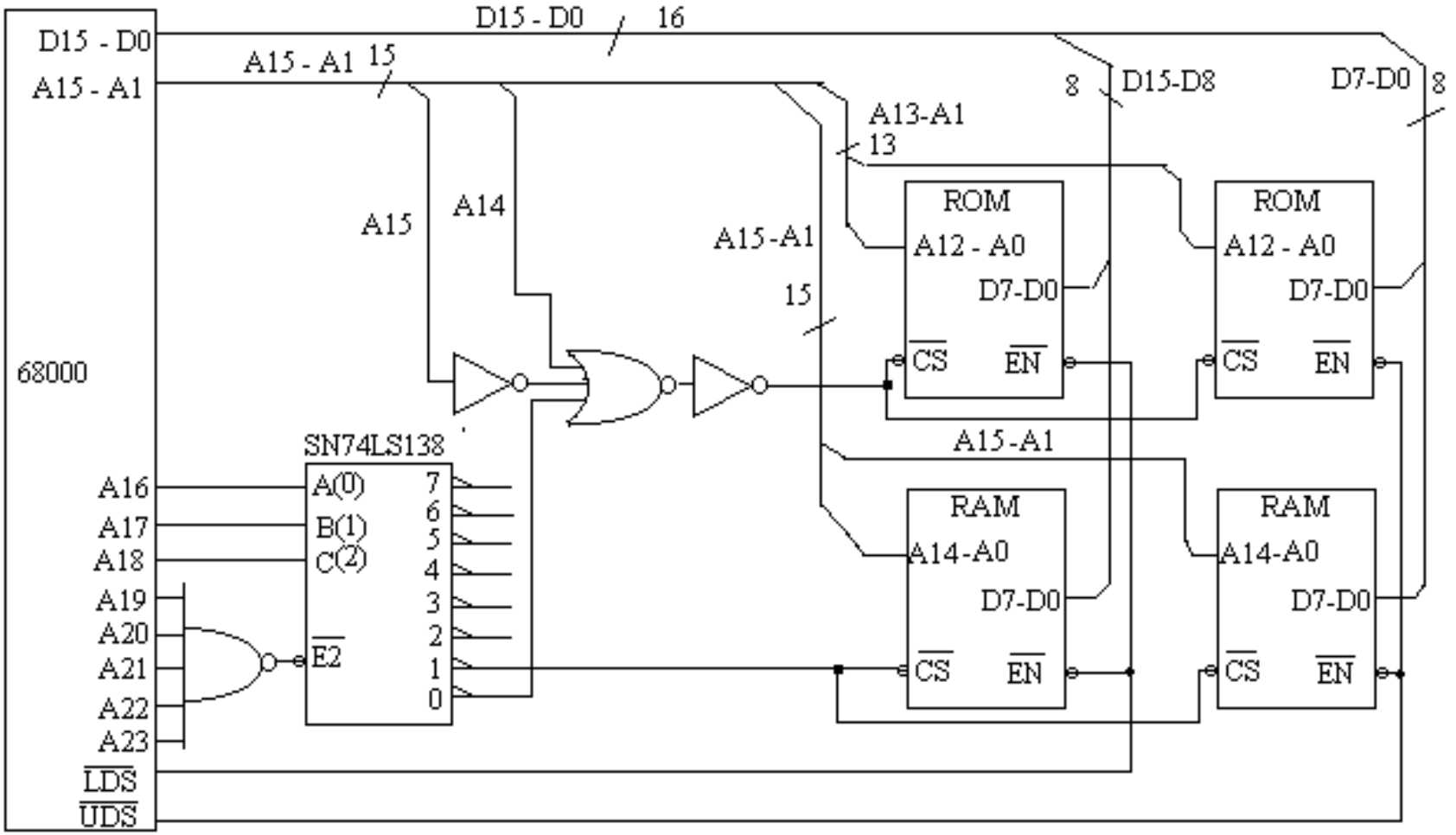
INPUTS						OUTPUTS							
$\overline{E_1}$	$\overline{E_2}$	E_3	A_0	A_1	A_2	$\overline{O_0}$	$\overline{O_1}$	$\overline{O_2}$	$\overline{O_3}$	$\overline{O_4}$	$\overline{O_5}$	$\overline{O_6}$	$\overline{O_7}$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

Memory Decoding with Byte/Word Access - 2

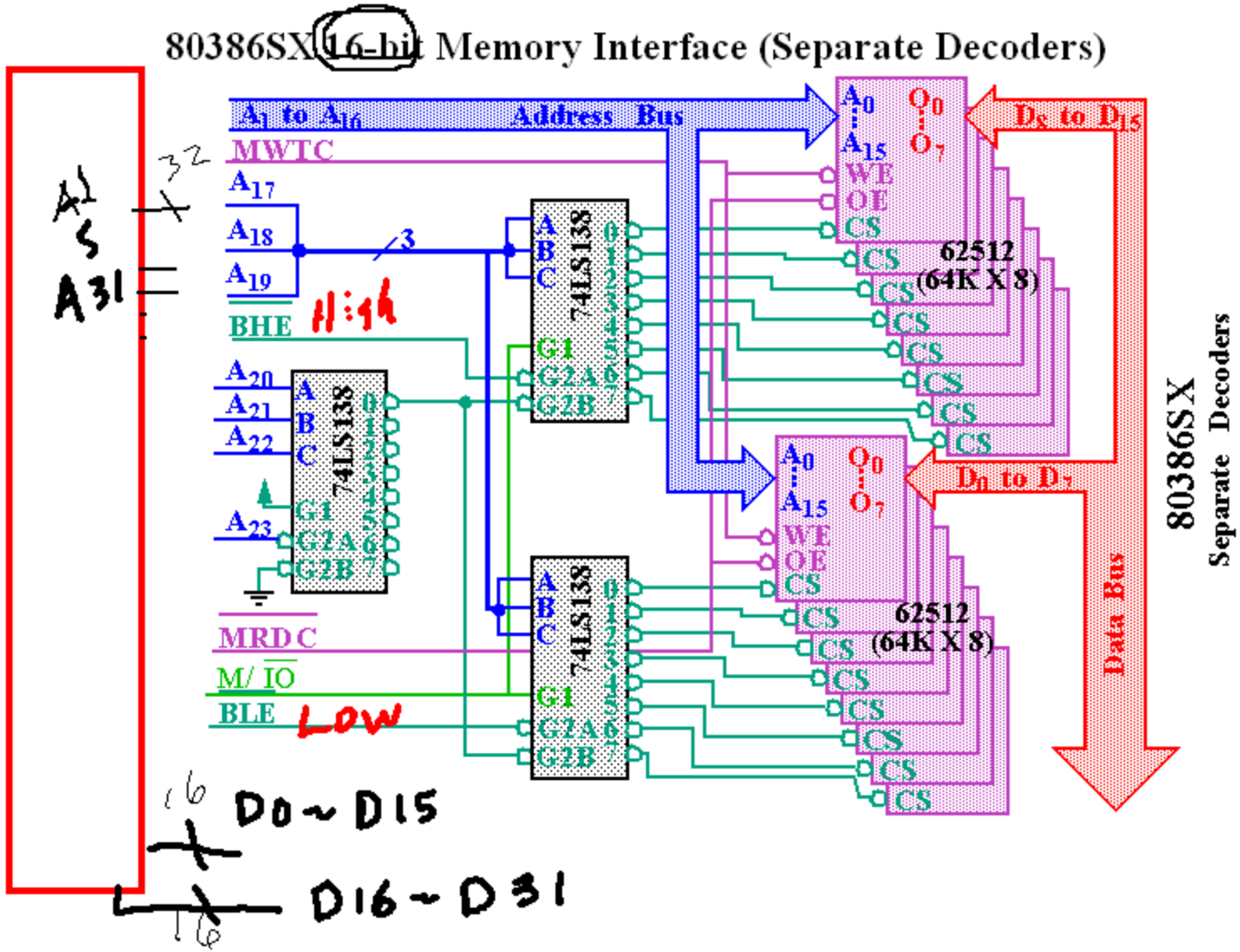


- Questions:
  - 1. Size of ROM
  - 2. Size of RAM
  - 3. Memory Map

Can You Draw a Memory Map of this? - 3



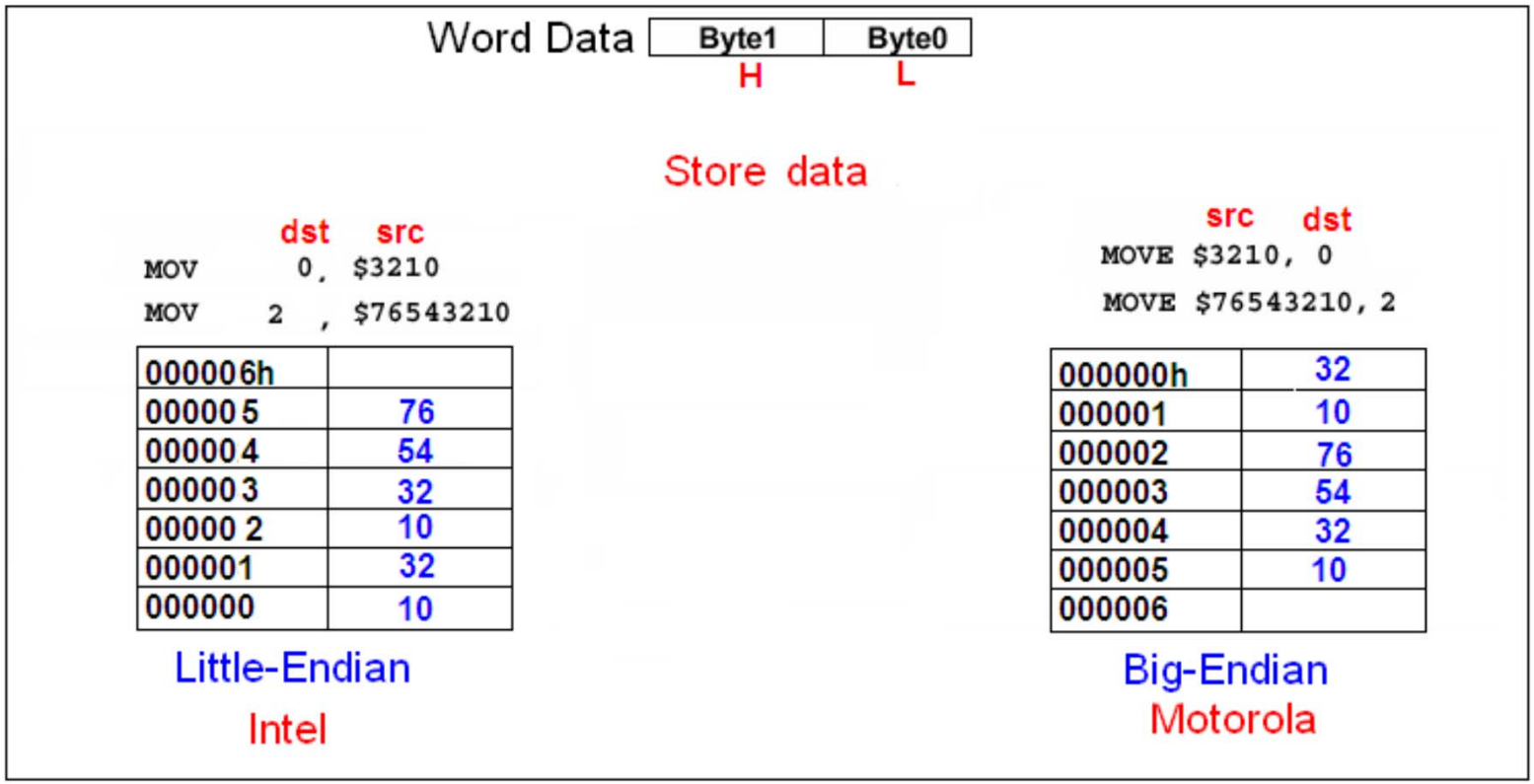
Intel 80386 Memory – Decoding - Assignment



- Find the address range of the **second** chips of the memory pair.
- **Due:**

Big-Endian vs. Little-Endian

- Big-Endian: Words are stored with the lower 8- bits in the higher of the two storage locations: Motorola
  - "Big guy ends at lower address"
- Little- Endian: Lower-order byte stored in the lowest address) processors: Intel 80x86 family
  - Little guys ends at lower address"



“Endianness”

- Endian or Endian-Architecture
  - how multi-byte data is represented by a computer system and is dictated by the CPU architecture of the system
  - Not all computer systems are designed with the same endian architecture
  - Issues with software and interface

Computer System Endianness

Platform	Endian Architecture
ARM*	Bi-Endian
DEC Alpha*	Little-Endian
HP PA-RISC 8000*	Bi-Endian
IBM PowerPC*	Bi-Endian
Intel® 80x86	Little-Endian
Intel® IXP network processors	Bi-Endian
Intel® Itanium® processor family	Bi-Endian
Java Virtual Machine*	Big-Endian
MIPS*	Bi-Endian
Motorola 68k*	Big-Endian
Sun SPARC*	Big-Endian

Common file formats

Little-Endian Format		Big-Endian Format		Variable or Bi-Endian Format	
BMP	(Windows* & OS/2)	PSD	(Adobe Photoshop*)	DXF	(AutoCAD*)
GIF		IMG	(GEM Raster*)	PS	(Postscript*, 8 bit interpreted text, no Endian issue)
FLI	(Autodesk Animator*)	JPEG, JPG		POV	(Persistence of Visionraytracer*)
PCX	(PC Paintbrush*)	MacPaint		RIFF	(WAV & AVI*)
QTM	(MAC Quicktime*)	SGI	(Silicon Graphics*)	TIFF	
RTF	(Rich Text Format)	Sun Raster		XWD	(X Window Dump*)
		WPG	(WordPerfect*)		
Bus Protocols		**Network Protocols**		**Bus Protocols**	
Infiniband		TCP/IP		GMII	(8 bit wide bus, no Endian issue)
PCI Express		UDP			
PCI-32/PCI-64					
USB					

Endian-Neutral Approaches

- Conversion
  - Byte Swap
  - Network I/O Macro
- "Endian Neutral": allowing the code to be ported easily between processors of different Endian-architectures, and without rewriting any code. Endian-neutral software is developed by identifying system memory and external data interfaces, and using Endian-neutral coding practices to implement the interfaces.
- HOMEWORK #1
  - Technical Report on "Endian-Neutral Approaches"
  - What? Why? How?
  - 2-3 pages; 1" margin all sides; 12 pt; Times New Roman; No cover page (Title and your name at the top, and start in the first page); single space; single column; again **the importance of the first paragraph**.
  - Submission: Hardcopy only by 5:10pm Tuesday Sept 25.

386 Instruction Set

- 9 Operation Categories
  - Data Transfer
  - Arithmetic
  - Shift/Rotate
  - String Manipulation
  - Bit Manipulation
  - Control Transfer
  - High Level Language Support
  - Operating System Support
  - Processor Control
- Number of operands: 0, 1, 2, or 3

ADDITION	
ADD	Add operands
ADC	Add with carry
INC	Increment operand by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract operands
SBB	Subtract with borrow
DEC	Decrement operand by 1
NEG	Negate operand
CMP	Compare operands
DAS	Decimal adjust for subtraction
AAS	ASCII Adjust for subtraction
MULTIPLICATION	
MUL	Multiply Double/Single Precision
IMUL	Integer multiply
AAM	ASCII adjust after multiply
DIVISION	
DIV	Divide unsigned
IDIV	Integer Divide
AAD	ASCII adjust before division