www.mwftr.com

EECE416 Microcomputer Fundamentals & Design

Computer Architecture

Dr. Charles Kim

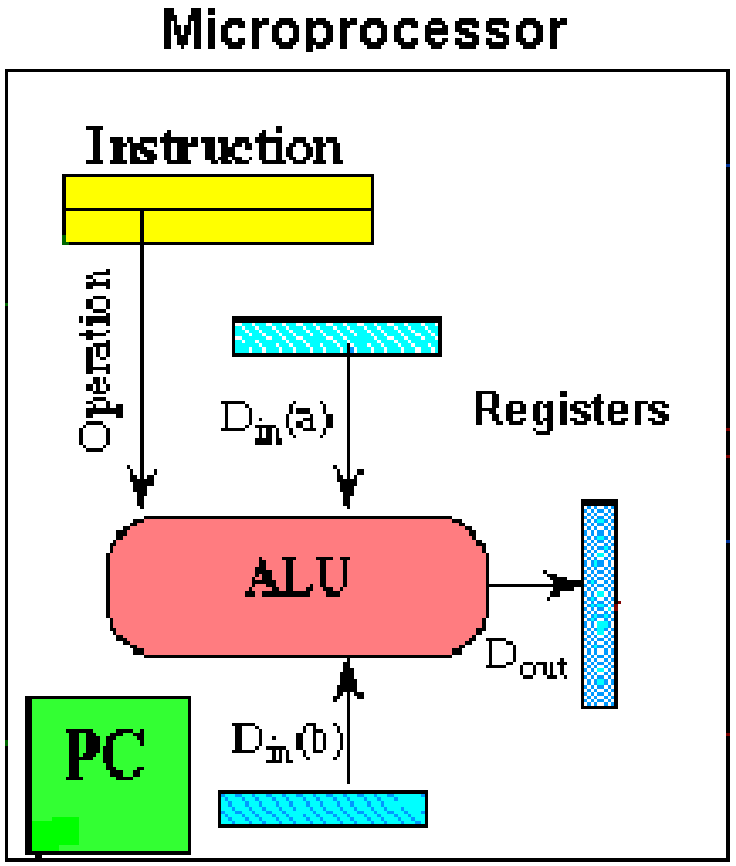
Howard University

"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
    - Car – views of manufacturers (or mechanics) and drivers

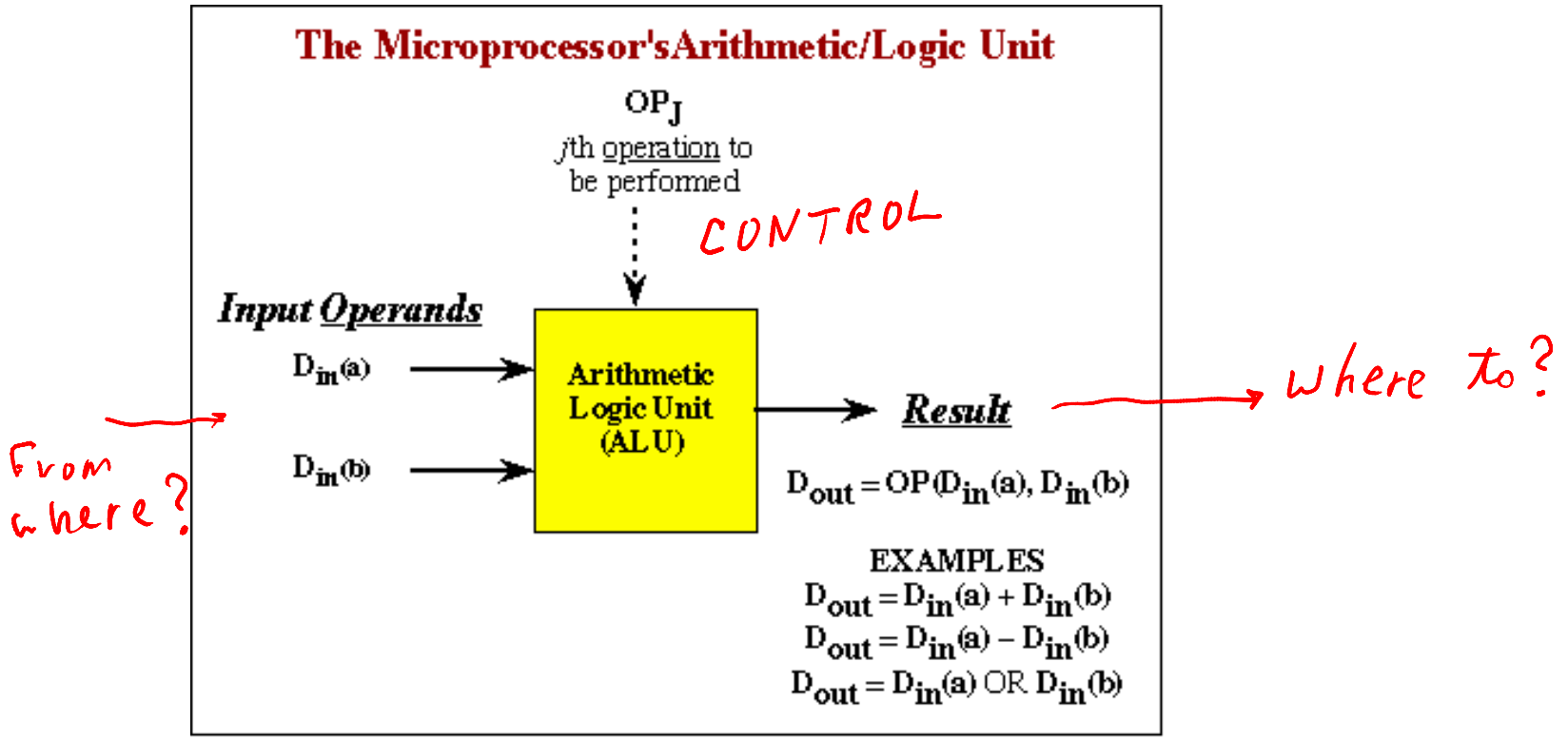
Micro-Architecture

- **Computer System**
  - CPU (with PC, Register, SR) + Memory
- **Micro-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.

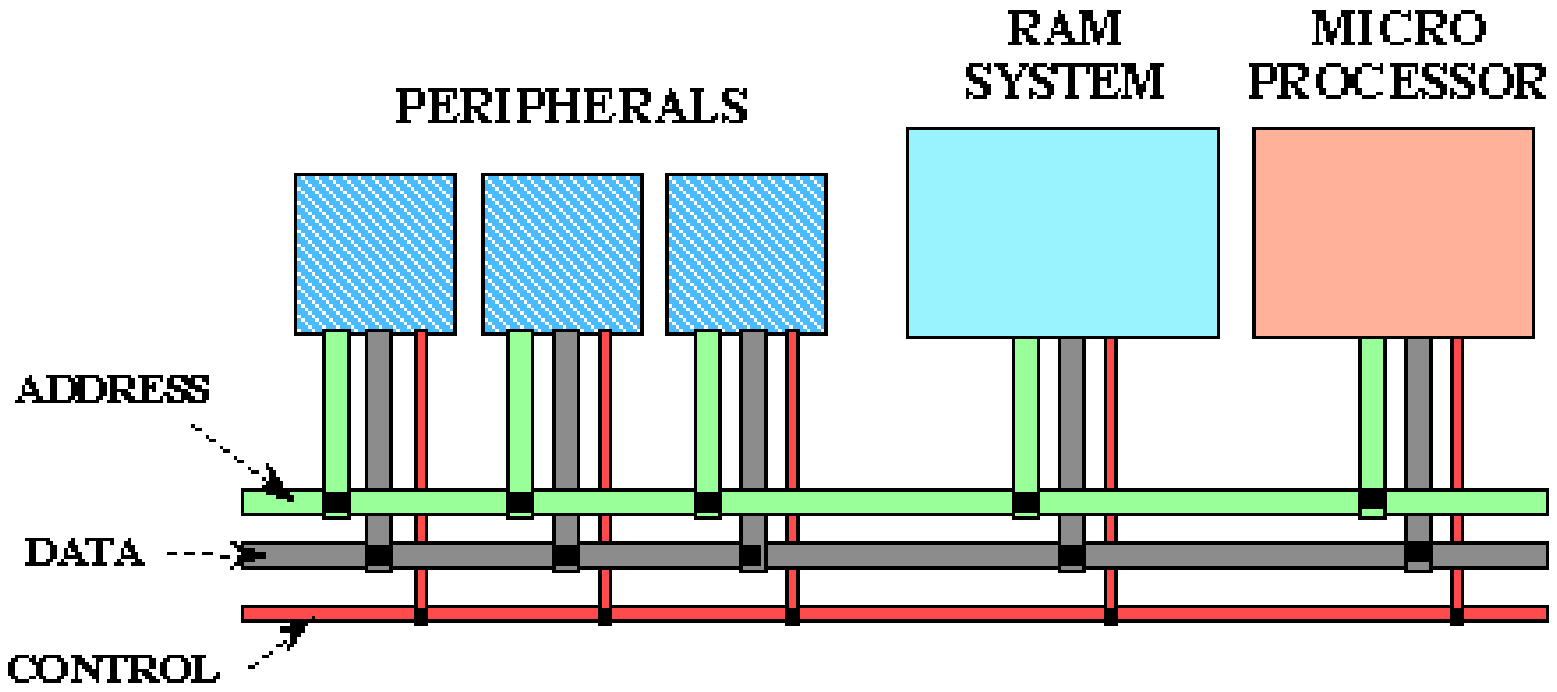


Micro-Architecture

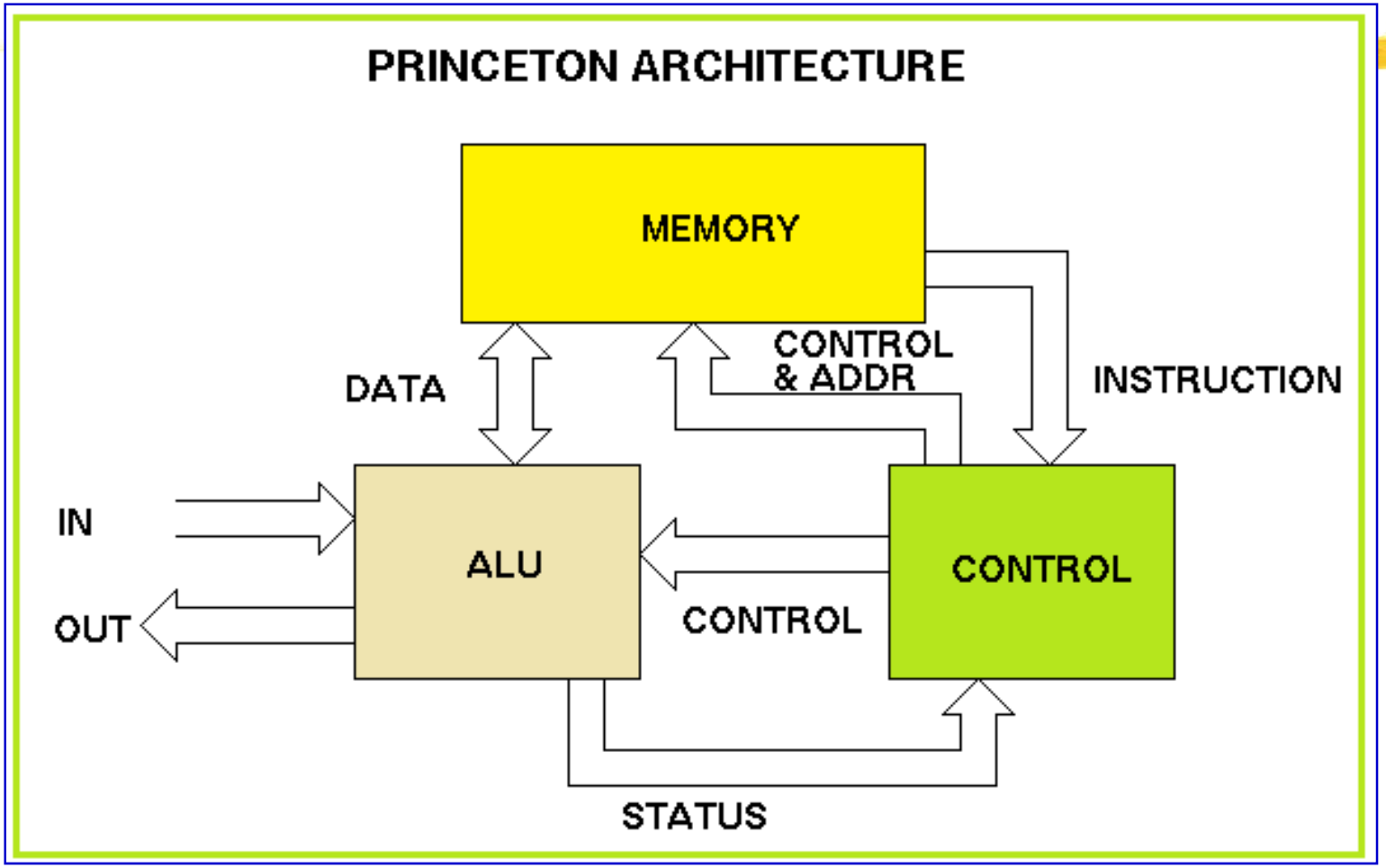
- **ALU (Arithmetic Logic Unit)**
  - **Fundamental building block of CPU**
  - **Binary Full Adder**



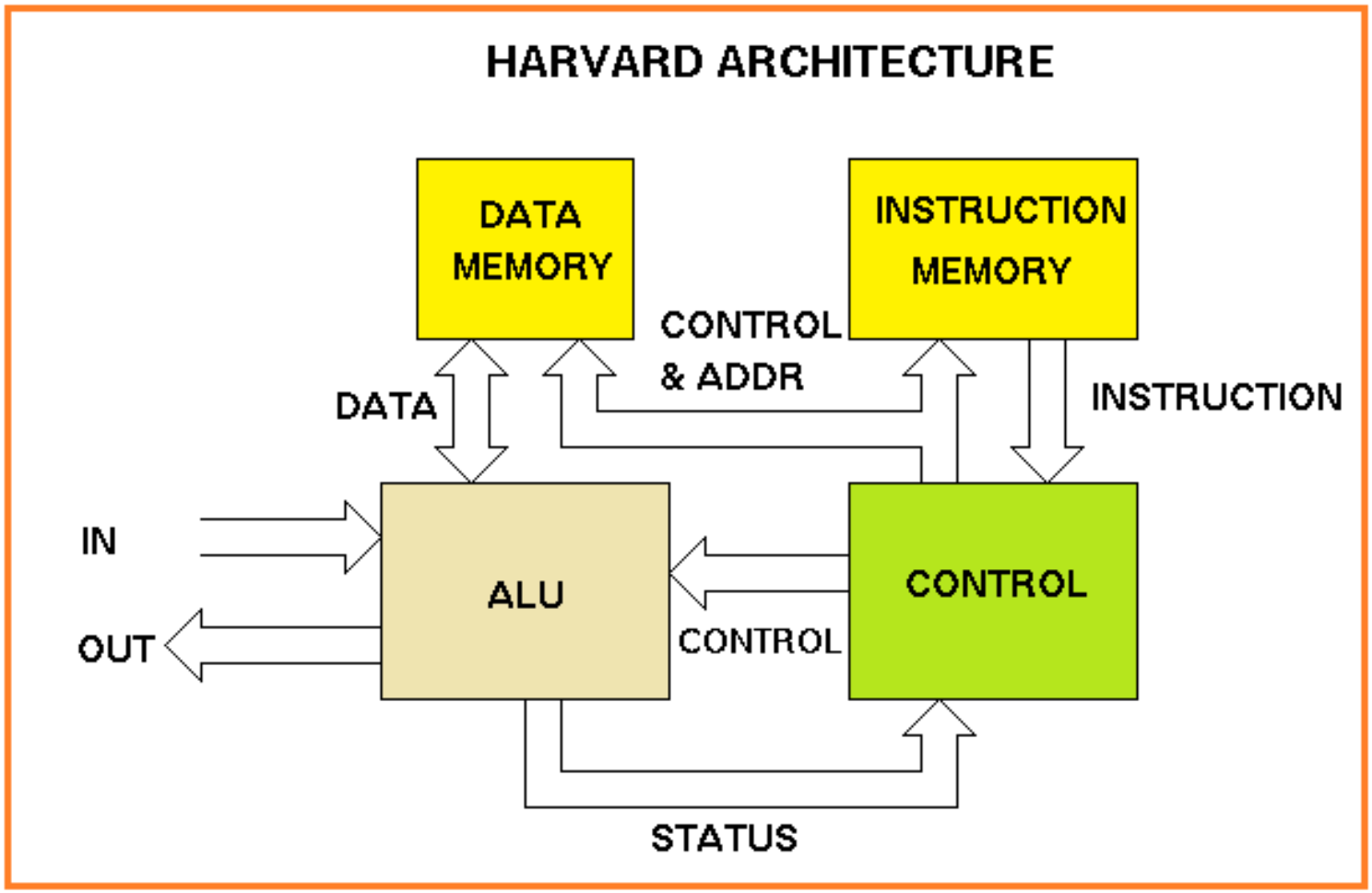
Microprocessor Bus



Architecture by CPU+MEM organization

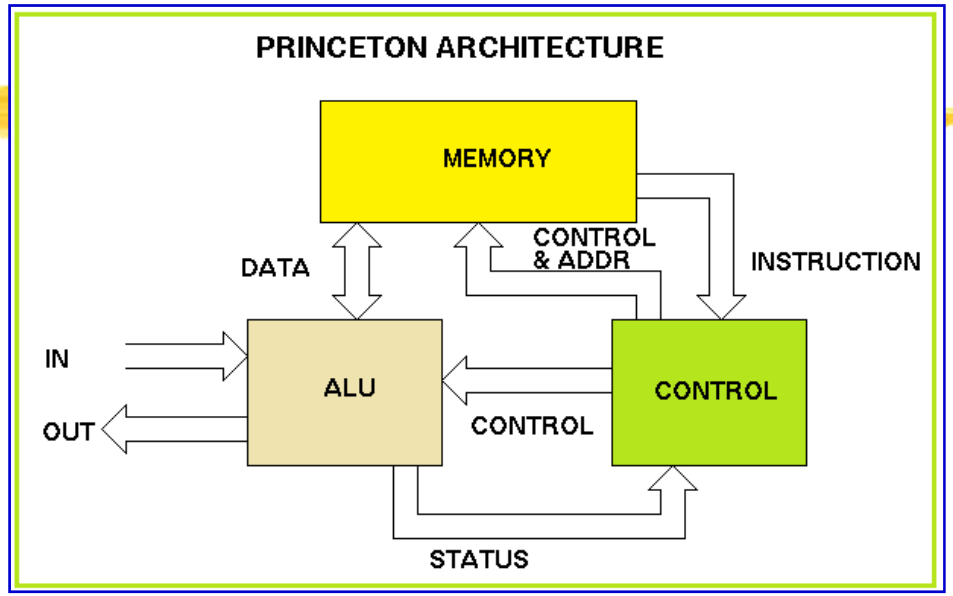


Architecture by CPU+MEM organization

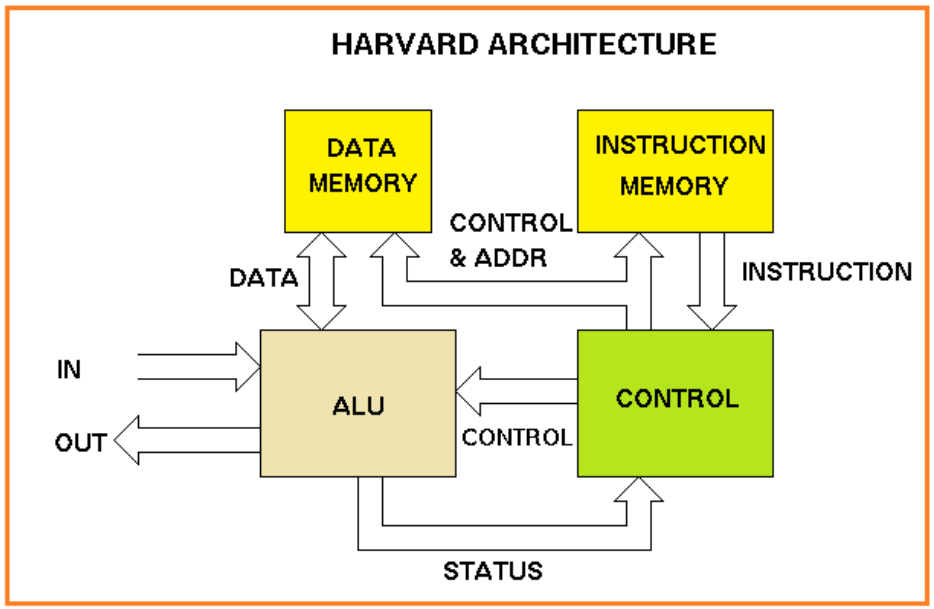


Architecture by CPU+MEM organization

- Princeton (or von Neumann) Architecture
  - MEM contains both Instruction and Data
  - Von Neumann Bottleneck – CPU <→ Memory



- Harvard Architecture
  - Data MEM and Instruction MEM
  - Higher Performance – via **Pipeline**
  - Higher MEM Bandwidth

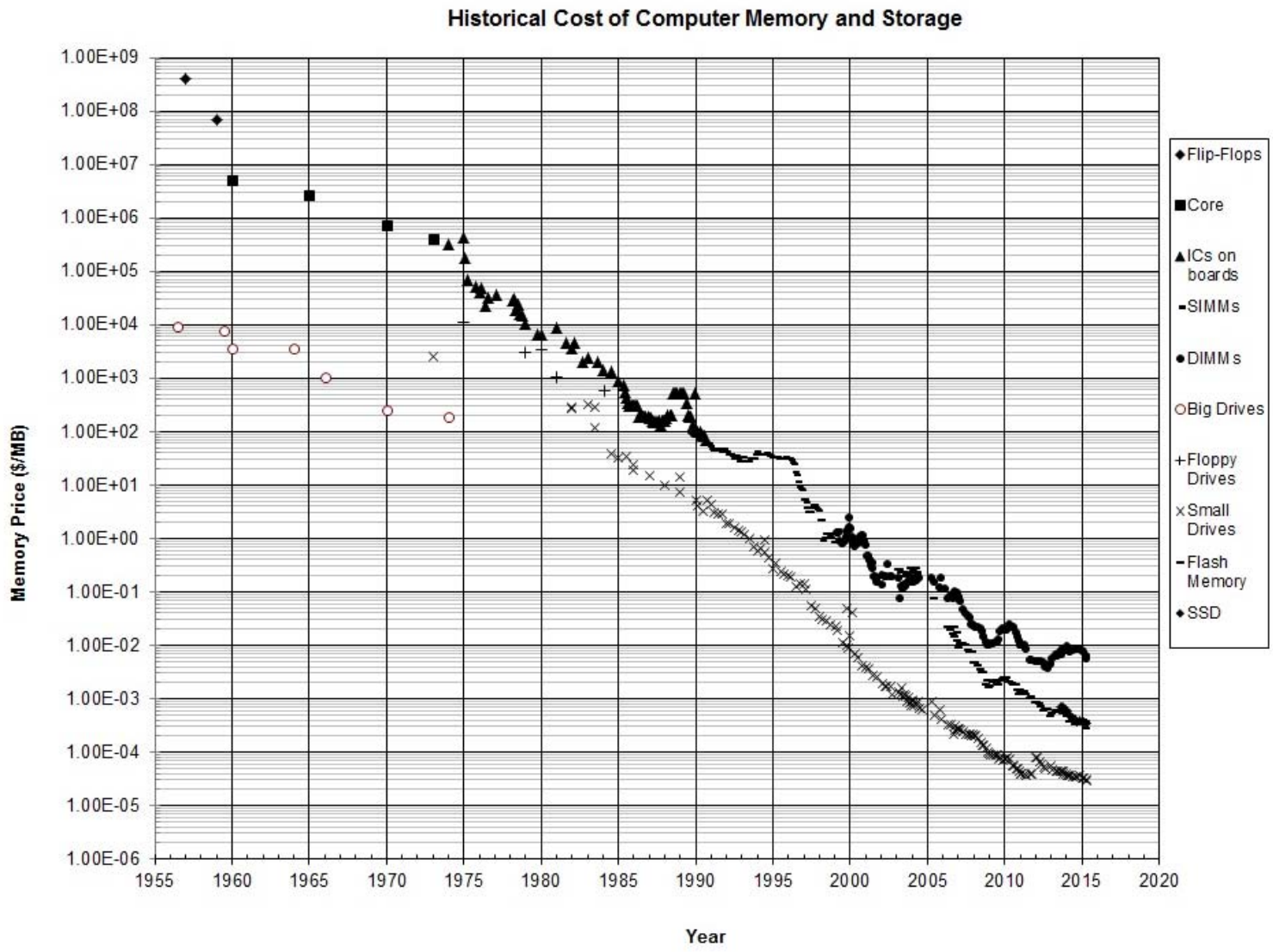


Memory Price

www.jcmit.com/mem2015.htm

Search

Graph of Memory Prices Decreasing with Time (1957-2015)



Memory Price

www.jcmit.com/memoryprice.htm

Search

Chart of Memory Price Through Time
Download Excel Spreadsheet of Memory Data
Home

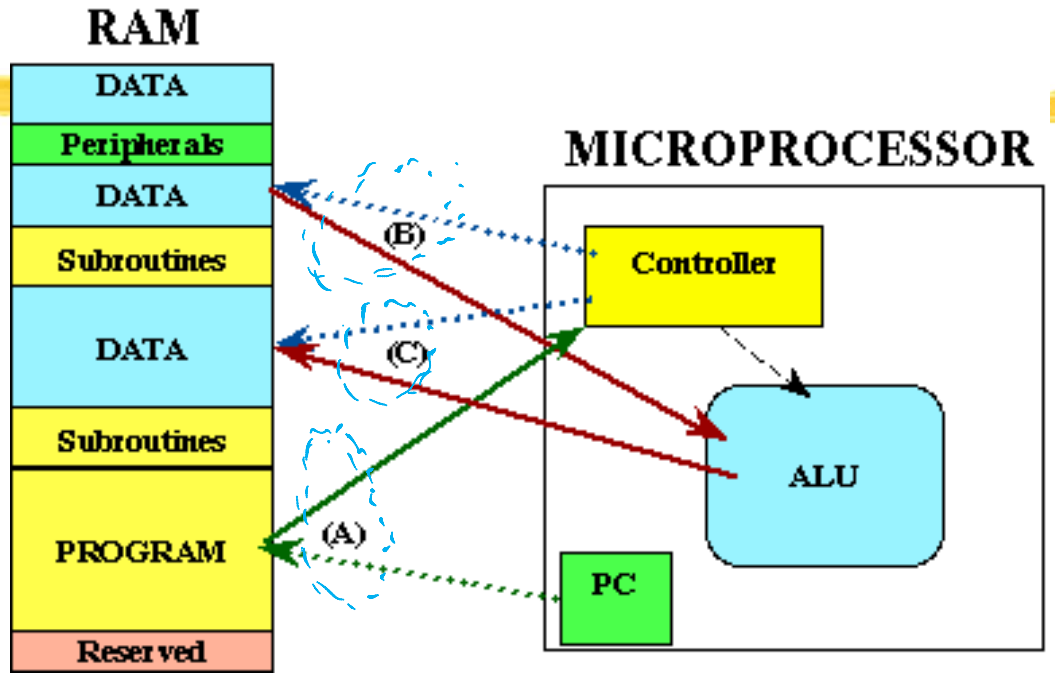
date (X)	$/Mbyte (Y)	Date		Ref:	Page	Company	Size	Cost	Speed	Memory Type	J
							KByte	US $	nsec		S
1957.00	411,041,792	1957		Phister 366		C.C.C.	0.00098	392.00	10000	transistor Flip-Flop	
1959.00	67,947,725	1959		Phister 366		E.E.Co.	0.00098	64.80	10000	vacuum tube Flip-Flop	
1960.00	5,242,880	1960		Phister 367		IBM	0.00098	5.00	11500	IBM 1401 core memory	
1965.00	2,642,412	1965		Phister 367		IBM	0.00098	2.52	2000	IBM 360/30 core memory	
1970.00	734,003	1970		Phister 367		IBM	0.00098	0.70	770	IBM 370/135 core memory	
1973.00	399,360	1973	Jan	PDP8/e User Price List		DEC	12	4680.00		Core memory 8KwordX12 bit	
1974.00	314,573	1974		Phister 367		IBM	0.00098	0.30	800	IC Memory for IBM 370/125	
1975.00	421,888	1975	Jan	Radio-Electronics		MITS	0.25	103.00	1000	Altair 8800 256 Byte Static Board	
1975.08	180,224	1975	Feb			MITS	1	176.00		Altair 1K Static Board	
1975.25	67,584	1975	Apr			MITS	4	264.00		Altair 4K DRAM Board	
1975.75	49,920	1975	Oct			MITS	4	195.00		Altair 4K Static(2102) RAM Board	
1976.00	40,704	1976	Jan			MITS	4	159.00		Altair 4K Static(2102) RAM Board	
1976.17	48,960	1976	Mar			MITS	16	765.00		Altair 16K Static RAM Board	
1976.42	23,040	1976	Jun			SD Sales	4	90.00		SD Sales 4K Static Board	
1976.58	32,000	1976	Aug				8	250.00		8K Static RAM Board	
1977.08	36,800	1977	Feb			TDL	16	575.00		S-100 16K	
1978.17	28,000	1978	Mar				64	1750.00		S-100 64K	
1978.25	29,440	1978	Apr				16	460.00			
1978.33	19,200	1978	May				16	300.00			
1978.50	24,000	1978	Jul			Extensis	64	1500.00			
1978.58	16,000	1978	Aug				8	125.00			
1978.75	15,200	1978	Oct				32	475.00			
1979.00	10,528	1979	Jan	Interface Age	124		32	329.00			
1979.75	6,704	1979	Oct			SD Sales - Jade	64	419.00		S-100, SD Sales/Jade 64K Kit	
1980.00	6,480	1980	Jan	Interface Age	121		64	405.00			

Princeton Architecture

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

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

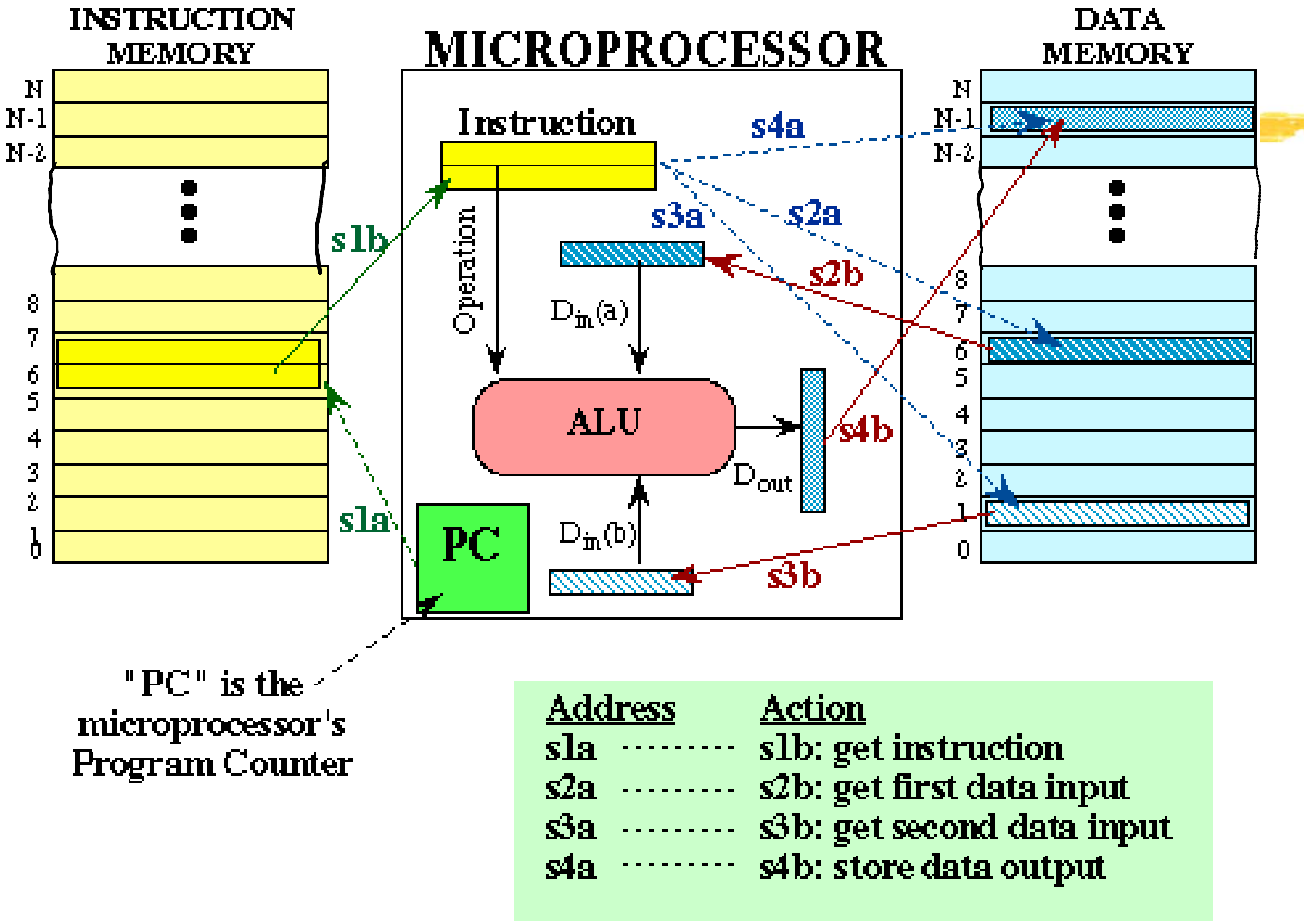
3.**Step (C)**: Following completion of the instruction, the controller provides the address, to the memory unit, at which the data result generated by the operation will be stored.



- 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 the 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
  - 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

- Architecture of late 1980s -- present
- Dramatic memory size increases/cost decreases
- Better pipelining
- Central to High Performance Computing
- Advanced optimizing compilers
- Instructions are of a uniform length

“Pipeline”?

⌘Instruction Pipeline

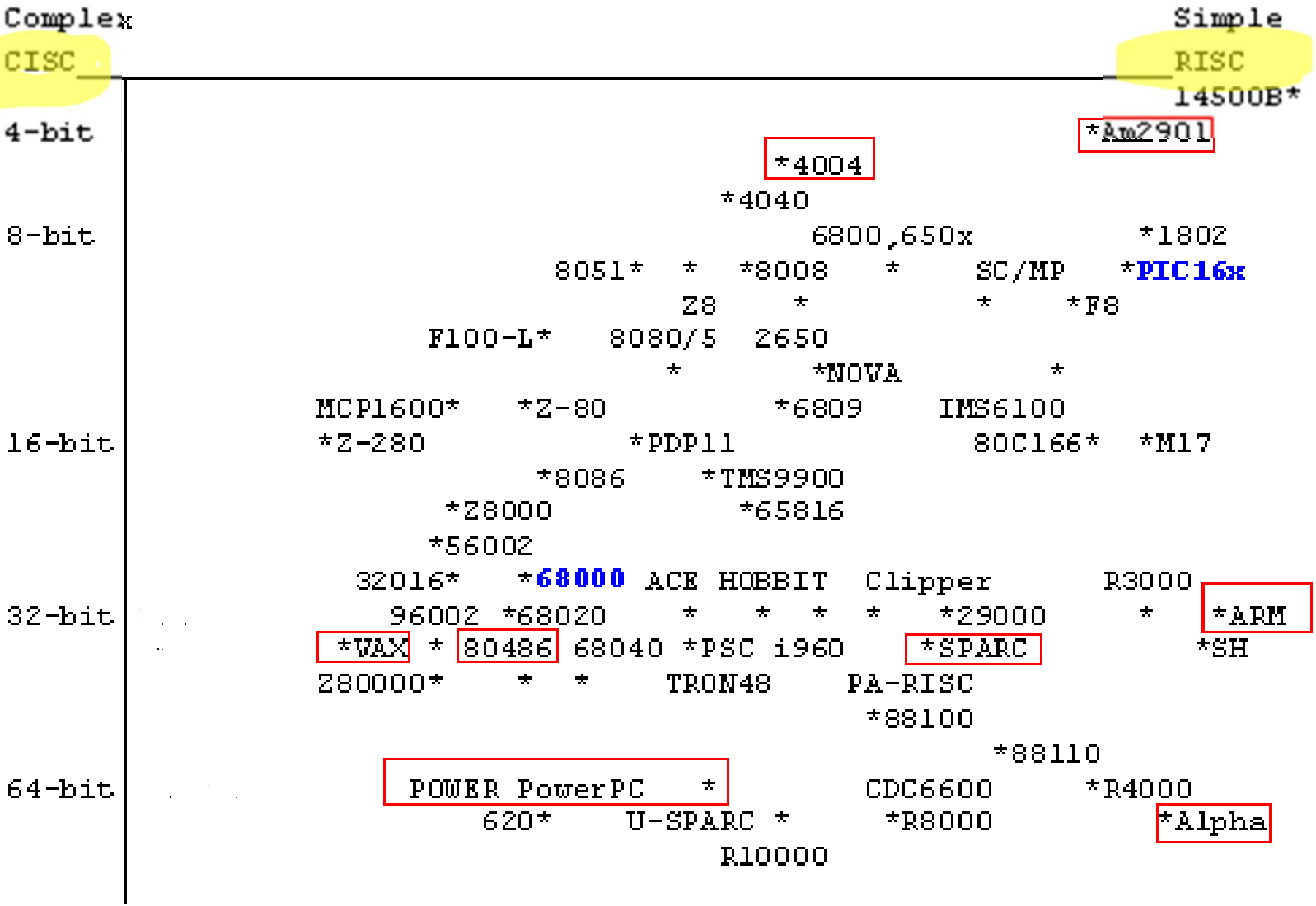
An **instruction pipeline** is a technique used in the design of computers to increase their instruction throughput (the number of instructions that can be executed in a unit of time). Pipelining does not reduce the time to complete an instruction, but increases instruction throughput by performing multiple operations in parallel.

The term pipeline is an analogy to the fact that there is fluid in each link of a pipeline, as each part of the processor is occupied with work.

Instr. No.	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

Basic five-stage pipeline in a RISC machine (IF = Instruction Fetch, ID = Instruction Decode, EX = Execute, MEM = Memory access, WB = Register write back). In the fourth clock cycle (the green column), the earliest instruction is in MEM stage, and the latest instruction has not yet entered the pipeline.

Processor Classification

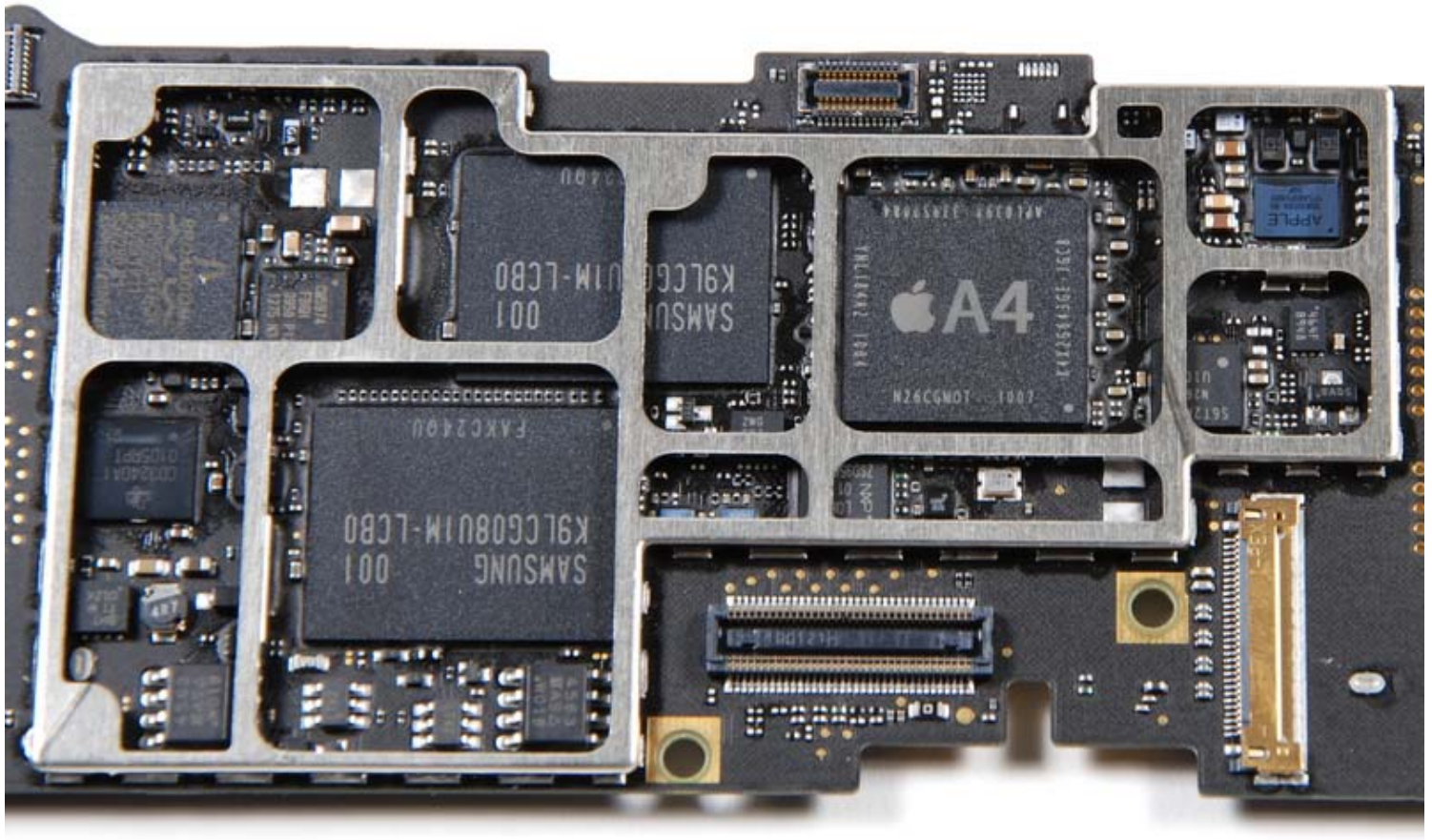


Intel inside?

- Next PCs or Mobile Computing Devices
  - Smart phones
    - Apple Processors
    - ARMs
    - Qualcomm
  - Mobile Devices – Smartphones, MP3, Digicam (on ARM)
  - Run on Intel's x86? --- Intel's wish; what happened to Lumina?

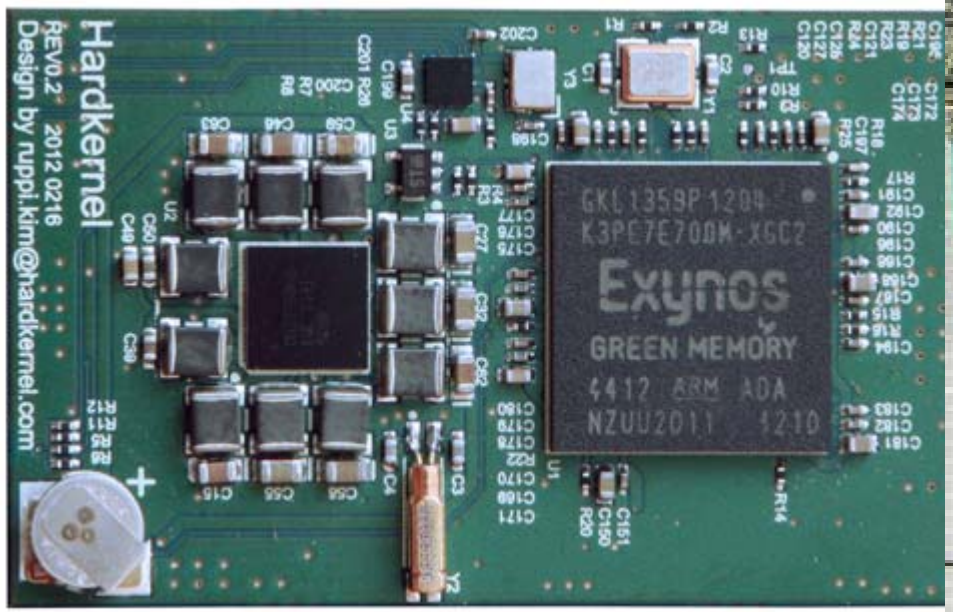
What's inside?

- iPhone: 1GHz-A4 microprocessor, 256MB Samsung RAM,



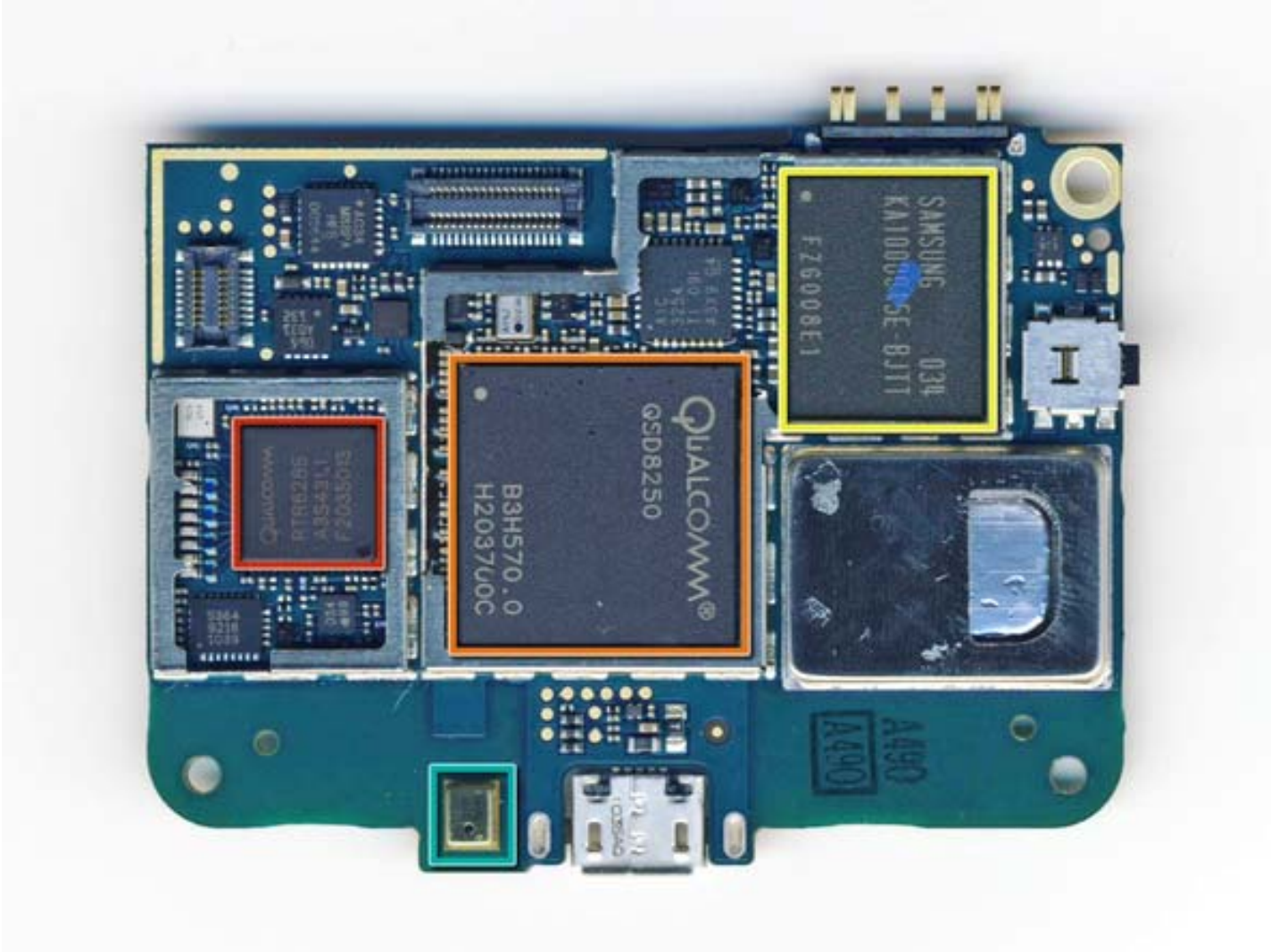
What's inside?

- Samsung Galaxy
  - Samsung Exynos quad-core A9 processor
  - 1GB Memory
  - Intel Wireless processor
  - Broadcom Global Navigation Satellite System receiver



What's Inside?

- HTC



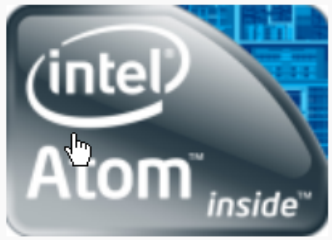
What's inside?

- Nokia Lumina
  - 1.4GHz Qualcomm CPU, 512MB RAM, 16GB Storage,

INTEL VS. ARM ("Advanced RISC Machine")

- ARMs
  - 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
- Recent Stride with "Intel Atom Inside"
  - Main processor for Laptops and Netbooks and Tablets
  - Motorola Phones: Razr
  - Apple computers with Intel chips

Intel Atom

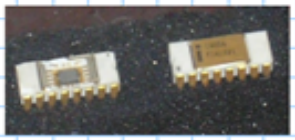


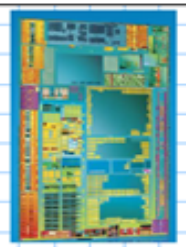
Produced	2008–present
Common manufacturer(s)	Intel
Max. CPU clock	800 MHz to 2 GHz
FSB speeds	400 MHz to 667 MHz
Min. feature size	45nm
Instruction set	x86, x86-64 (not for the N and Z series)
Cores	1, 2
Package(s)	441-ball µFCBGA
Core name(s)	Silverthorne Diamondville

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 Processor History

Note Title





4-bit

4004

8008

8080

8085

8086

8088

80286/80186

80386

486

Pentium

Pentium Pro

Pentium II

Celeron

Pentium III

Pentium 4

Itanium

Core Duo

Atom Z540 N270

Core i7

P5

P6

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

7 8 9 0

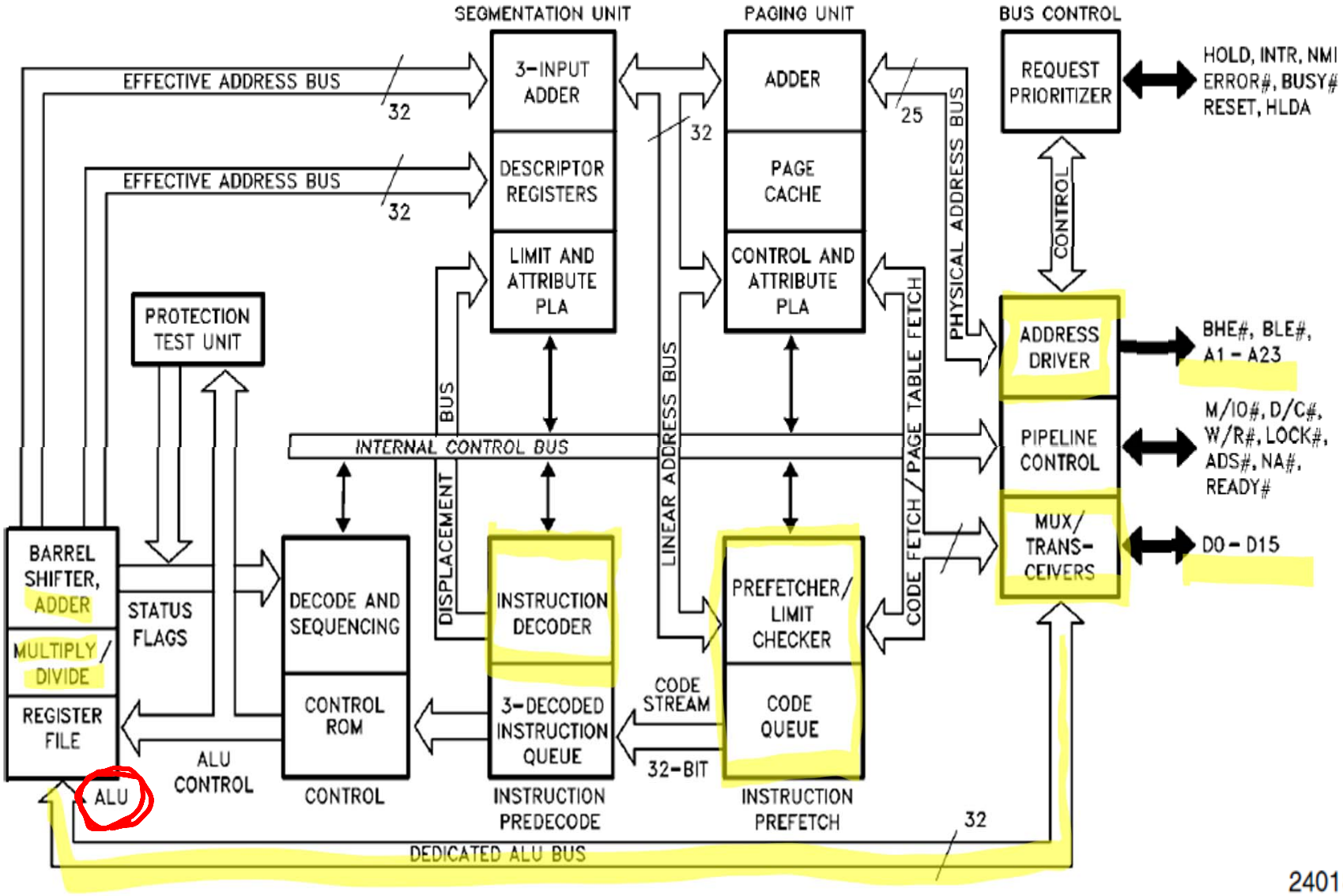
Intel 386 - Datasheet

intel

Intel386™ SX MICROPROCESSOR

- Full 32-Bit Internal Architecture
  - — 8-, 16-, 32-Bit Data Types
  - — 8 General Purpose 32-Bit Registers
- Runs Intel386™ Software in a Cost Effective 16-Bit Hardware Environment
  - — Runs Same Applications and O.S.'s as the Intel386™ DX Processor
  - — Object Code Compatible with 8086, 80186, 80286, and Intel386™ Processors
- High Performance 16-Bit Data Bus
  - — 16, 20, 25 and 33 MHz Clock
  - — Two-Clock Bus Cycles
  - — Address Pipelining Allows Use of Slower/Cheaper Memories
- Integrated Memory Management Unit
  - — Virtual Memory Support
  - — Optional On-Chip Paging
  - — 4 Levels of Hardware Enforced Protection
  - — MMU Fully Compatible with Those of the 80286 and Intel386 DX CPUs
- Virtual 8086 Mode Allows Execution of 8086 Software in a Protected and Paged System
- Large Uniform Address Space
  - — 16 Megabyte Physical
  - — 64 Terabyte Virtual
  - — 4 Gigabyte Maximum Segment Size
- Numerics Support with the Intel387™ SX Math CoProcessor
- On-Chip Debugging Support Including Breakpoint Registers
- Complete System Development Support
  - — Software: C, PL/M, Assembler
  - — Debuggers: PMON-386 DX, ICE™-386 SX
- High Speed CHMOS IV Technology
- Operating Frequency:
  - — Standard (Intel386 SX -33, -25, -20, -16) Min/Max Frequency (4/33, 4/25, 4/20, 4/16) MHz
  - — Low Power (Intel386 SX -33, -25, -20, -16, -12) Min/Max Frequency (2/33, 2/25, 2/20, 2/16, 2/12) MHz
- 100-Pin Plastic Quad Flatpack Package
  (See Packaging Outlines and Dimensions #231369)

Intel 386 - Datasheet

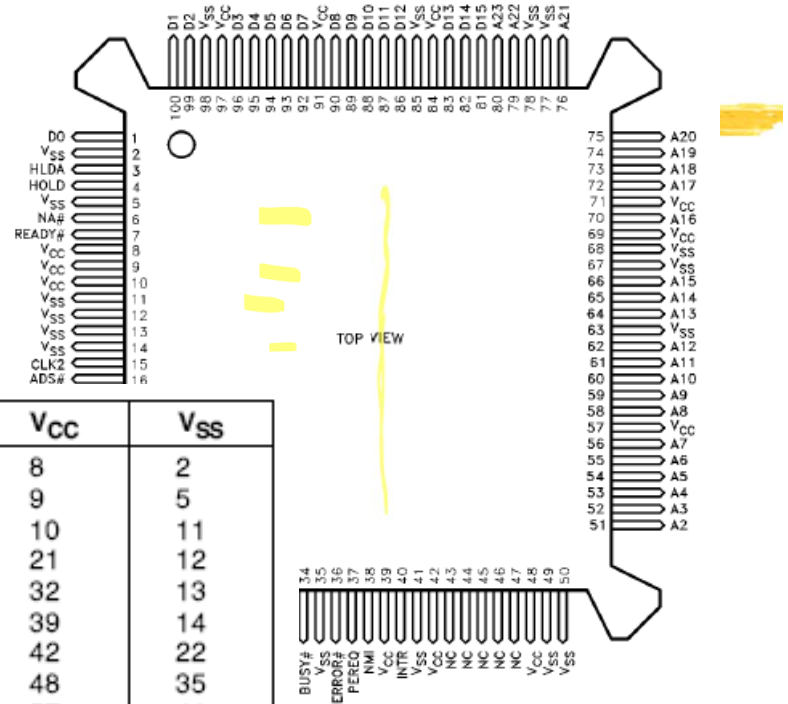


240187-47

Intel386™ SX Pipelined 32-Bit Microarchitecture

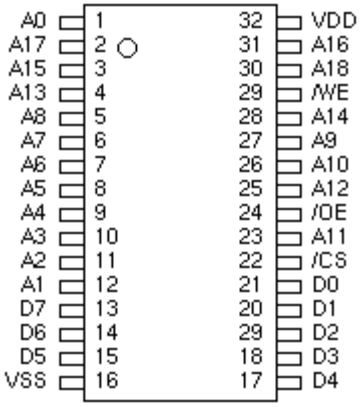
Intel 386 - Brief

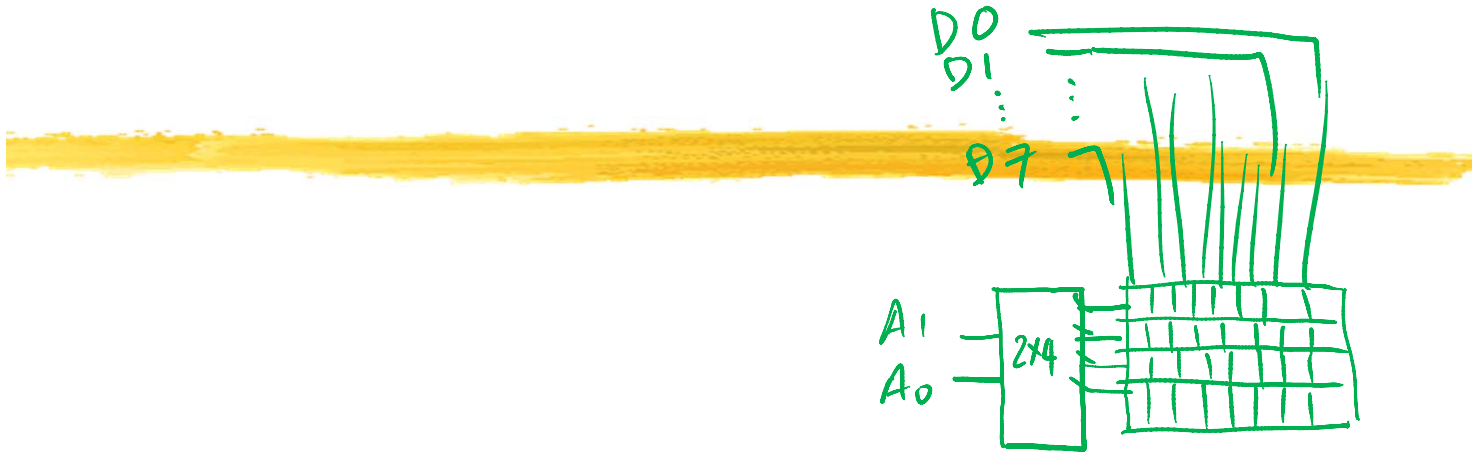
- Address: A23- A1
  - BLE# and BHE# (“Byte Enable”)
- Data: D15 – D0
- Control

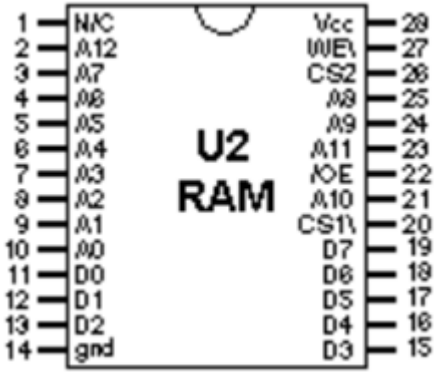


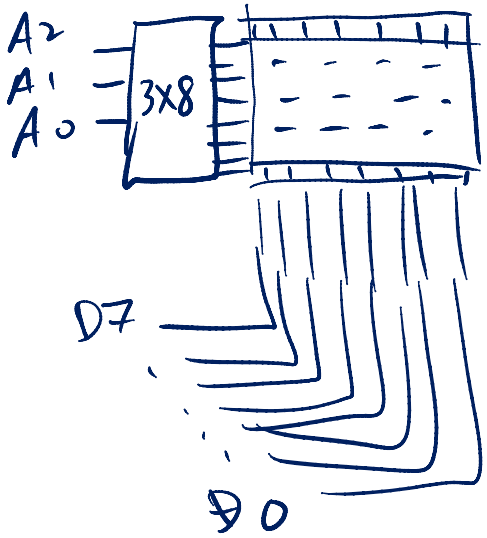
Address		Data		Control		N/C	V_{CC}	V_{SS}
A_1	18	D_0	1	ADS#	16	20	8	2
A_2	51	D_1	100	BHE#	19	27	9	5
A_3	52	D_2	99	BLE#	17	29	10	11
A_4	53	D_3	96	BUSY#	34	30	21	12
A_5	54	D_4	95	CLK2	15	31	32	13
A_6	55	D_5	94	D/C#	24	43	39	14
A_7	56	D_6	93	ERROR#	36	44	42	22
A_8	58	D_7	92	FLT#	28	45	48	35
A_9	59	D_8	90	HLDA	3	46	57	41
A_{10}	60	D_9	89	HOLD	4	47	69	49
A_{11}	61	D_{10}	88	INTR	40		71	50
A_{12}	62	D_{11}	87	LOCK#	26		84	63
A_{13}	64	D_{12}	86	M/IO#	23		91	67
A_{14}	65	D_{13}	83	NA#	6		97	68
A_{15}	66	D_{14}	82	NMI	38			77
A_{16}	70	D_{15}	81	PEREQ	37			78
A_{17}	72			READY#	7			85
A_{18}	73			RESET	33			98
A_{19}	74			W/R#	25			
A_{20}	75							
A_{21}	76							
A_{22}	79							
A_{23}	80							

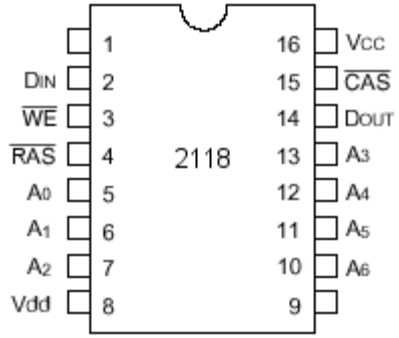
Memory Size and Address



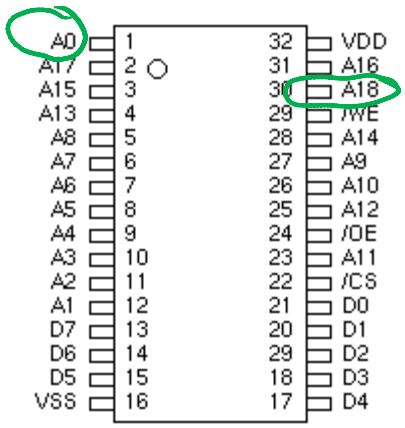








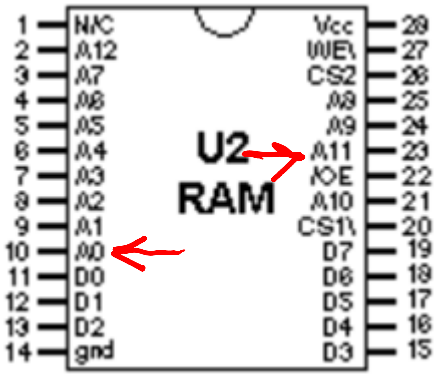
Memory Size and Address



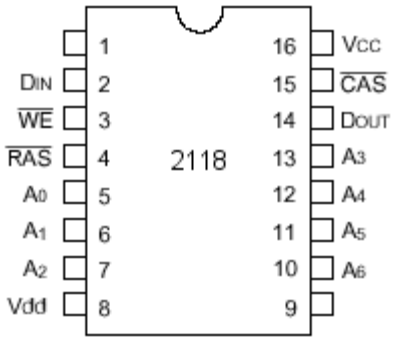
# of Address Lines

2^n

$2^{19} = 2^9 \cdot 2^{10} = 512\,KB$



$2^{13} = 2^3 \cdot 2^{10} = 8\,KB$



serial memory

$2^7 = 128\,B$

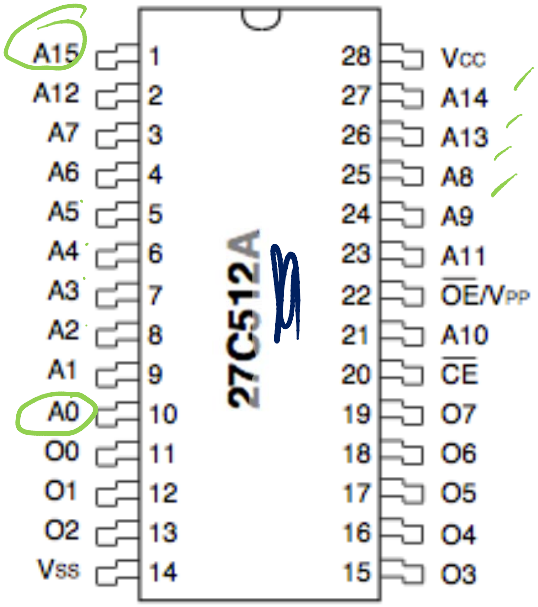
2^{10} 1KB

2^{20} 1MB

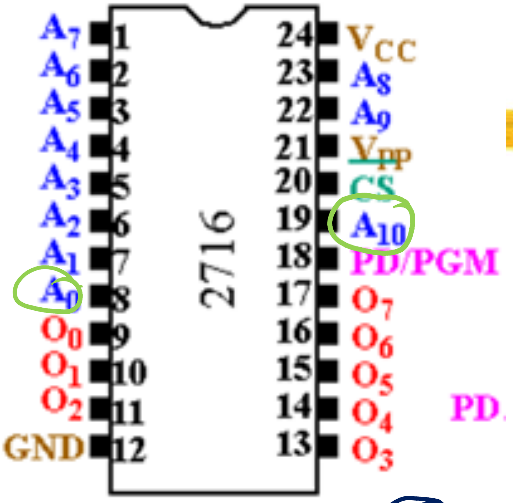
2^{30} 1GB

2^{40} 1TB

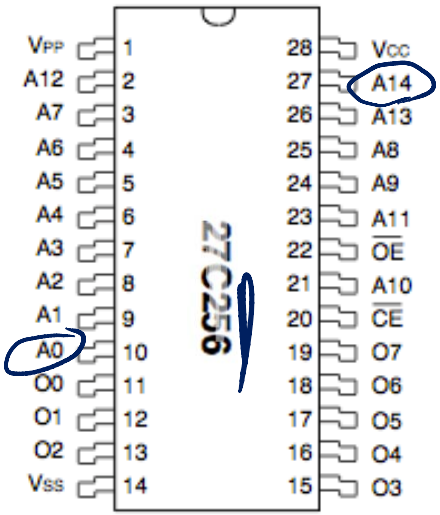
Memory Size and Address 2



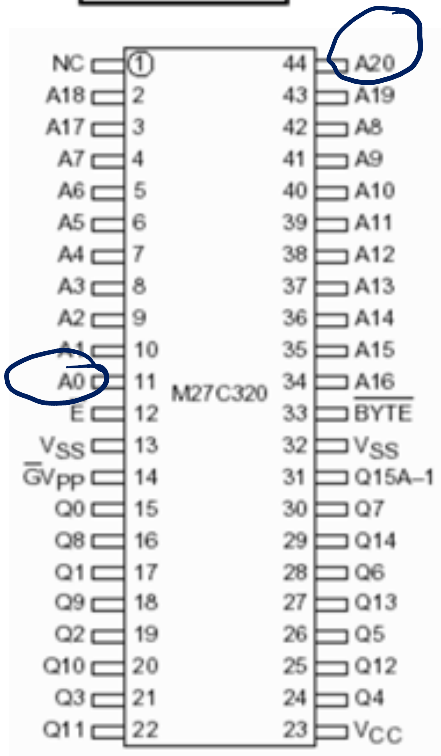
$2^{16} = 2^6 \cdot 2^{10}$
$= 64\,KB$



$2^{11} = 2 \cdot 2^{10}$
$= 2\,KB$



$2^{15} = 2^5 \cdot 2^{10}$
$= 32\,KB$



$2^{21} = 2 \cdot 2^{20}$
$= 2\,MB$

Review on Number Systems

	Binary	Hexadecimal	Decimal
1.	100	4	4
2.	10101101	AD	173
3.	1101110101	375	885
4.	11111011110	7DE	2014
5.	10000000001	401	4097
6.		8EF	
7.	10000	10	16
8.		A52E	
9.	11100001100	70C	
10.	0110101111010011	6BD3	
11.	0110 0100	64	100
12.			527
13.			4128
14.			11947
15.			59020

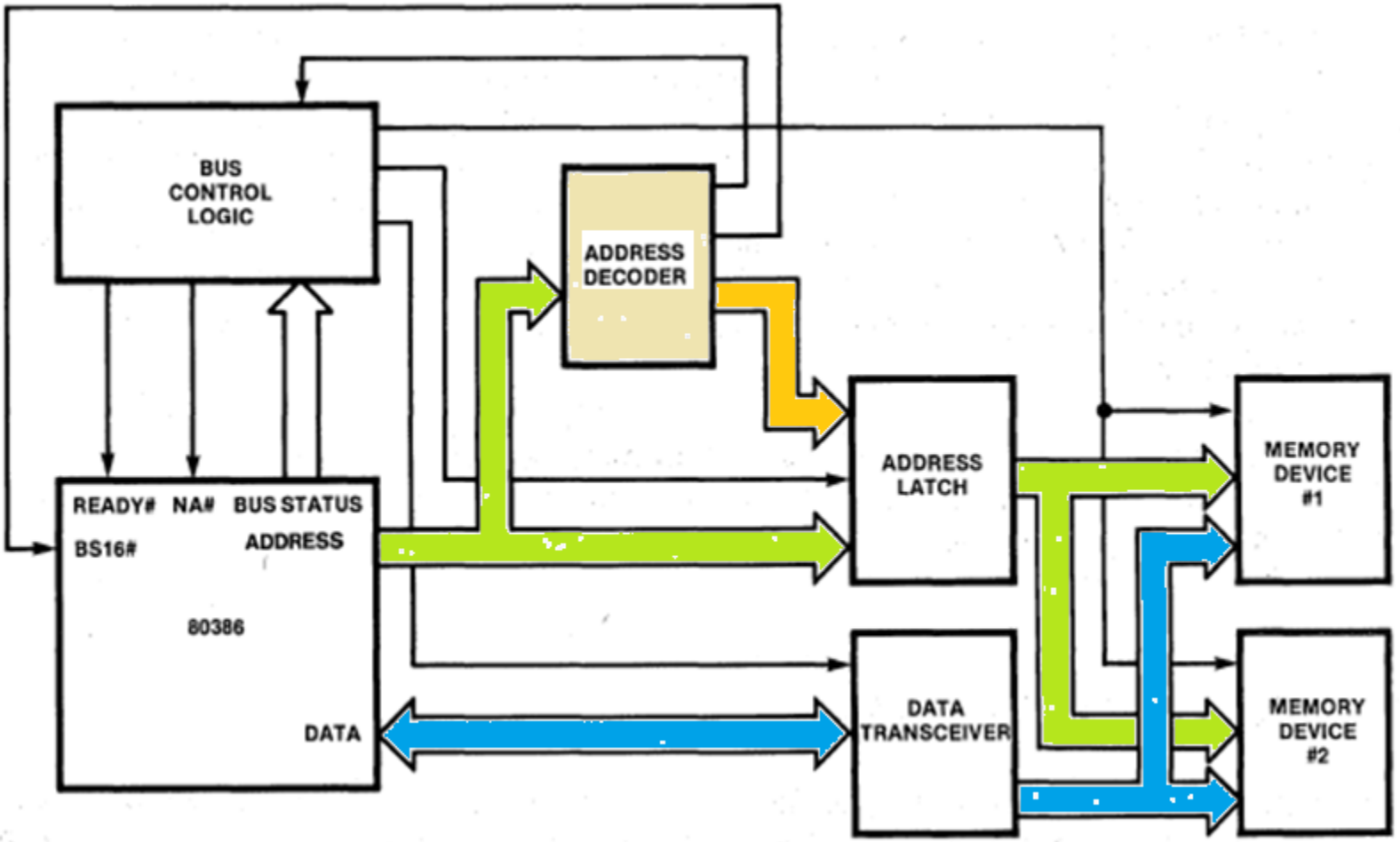
3·256 + 7·16 + 5

1001 9
1010 A
1011 B
1100 C
1101 D
1110 E
1111 F

16

160 + 13 = 173

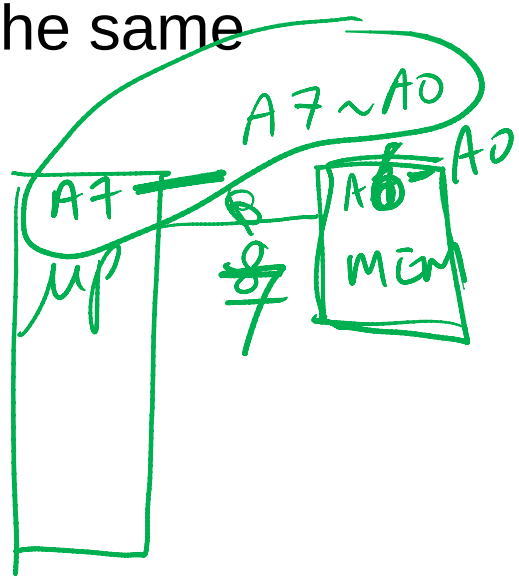
Memory Interface



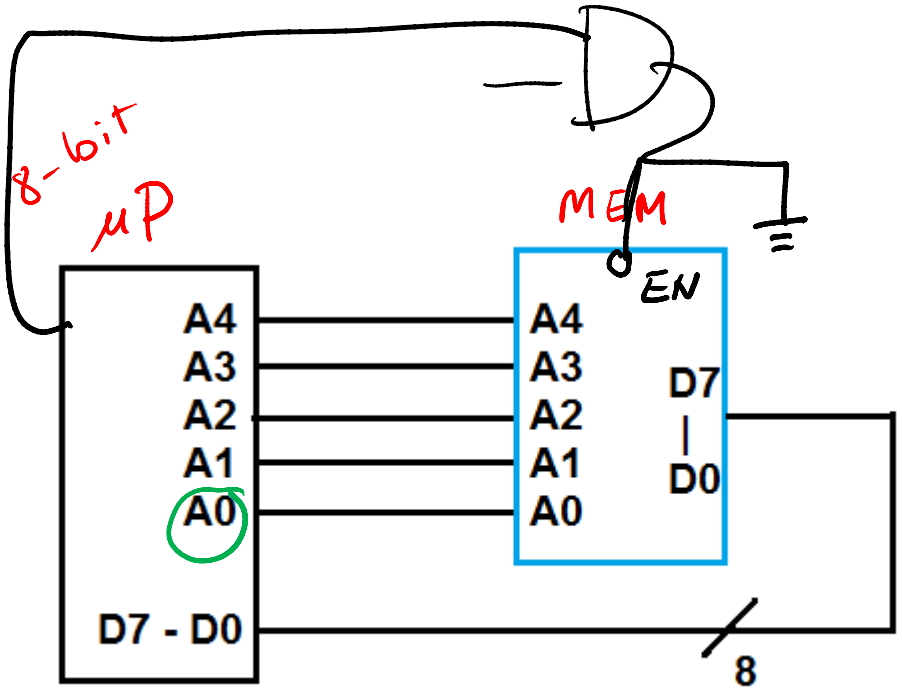
Fan-out

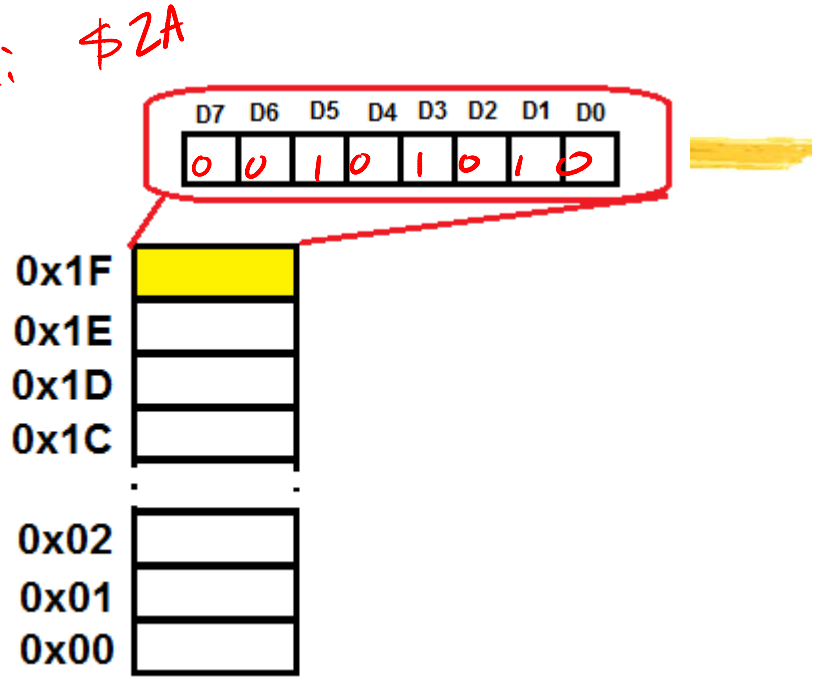
Memory Interface

- 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? → "MEMORY DECODING"
- How to access two MEM locations at the same time (for 16-bit Data bus)?
  - 8 MEM --- Byte Access (8 bits)
  - 16 UDS and LDS --- Motorola
  - 16 BLE and BHE --- Intel



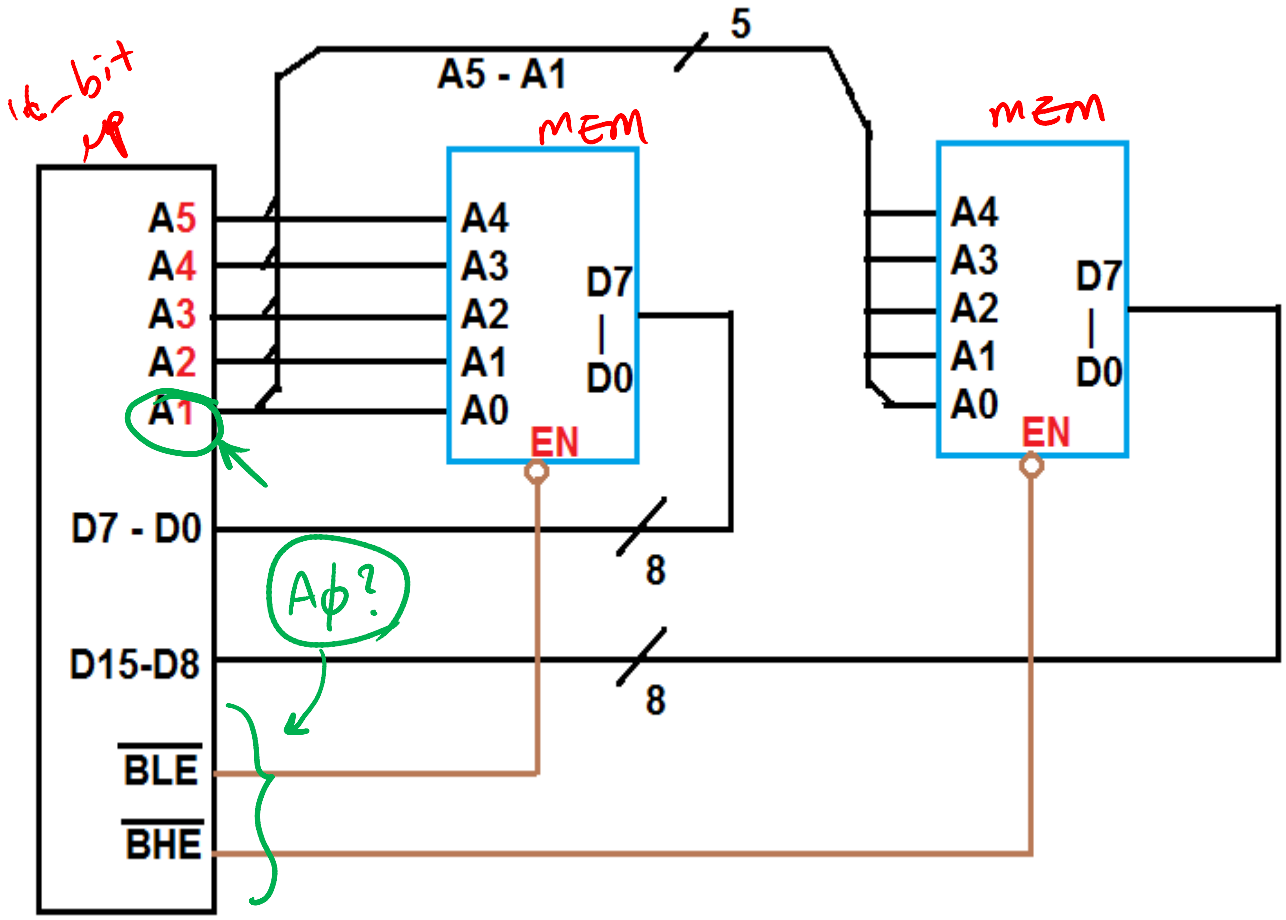
8-bit access

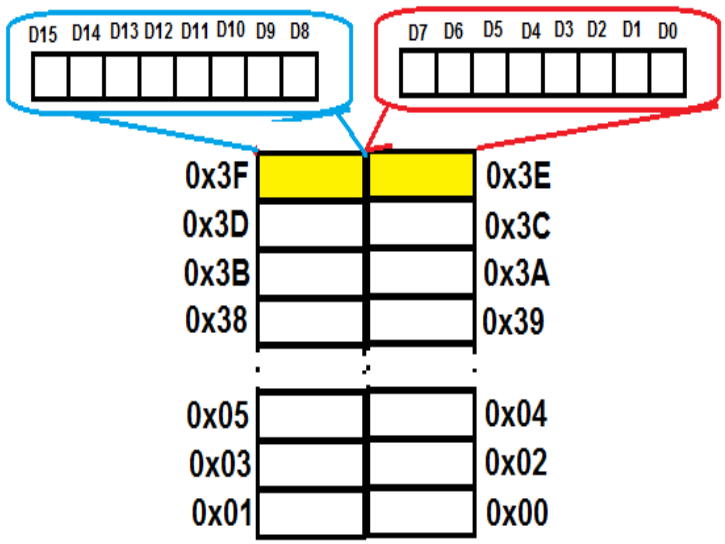

8-bit
μP
MEM
EN
A4
A3
A2
A1
A0
D7 - D0
A4
A3
A2
A1
A0
D7
|
D0
8

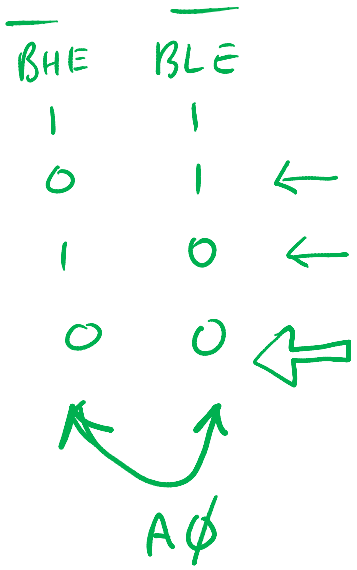

Ex: $2A
D7 D6 D5 D4 D3 D2 D1 D0
0 0 1 0 1 0 1 0
0x1F
0x1E
0x1D
0x1C
0x02
0x01
0x00

16-bit access

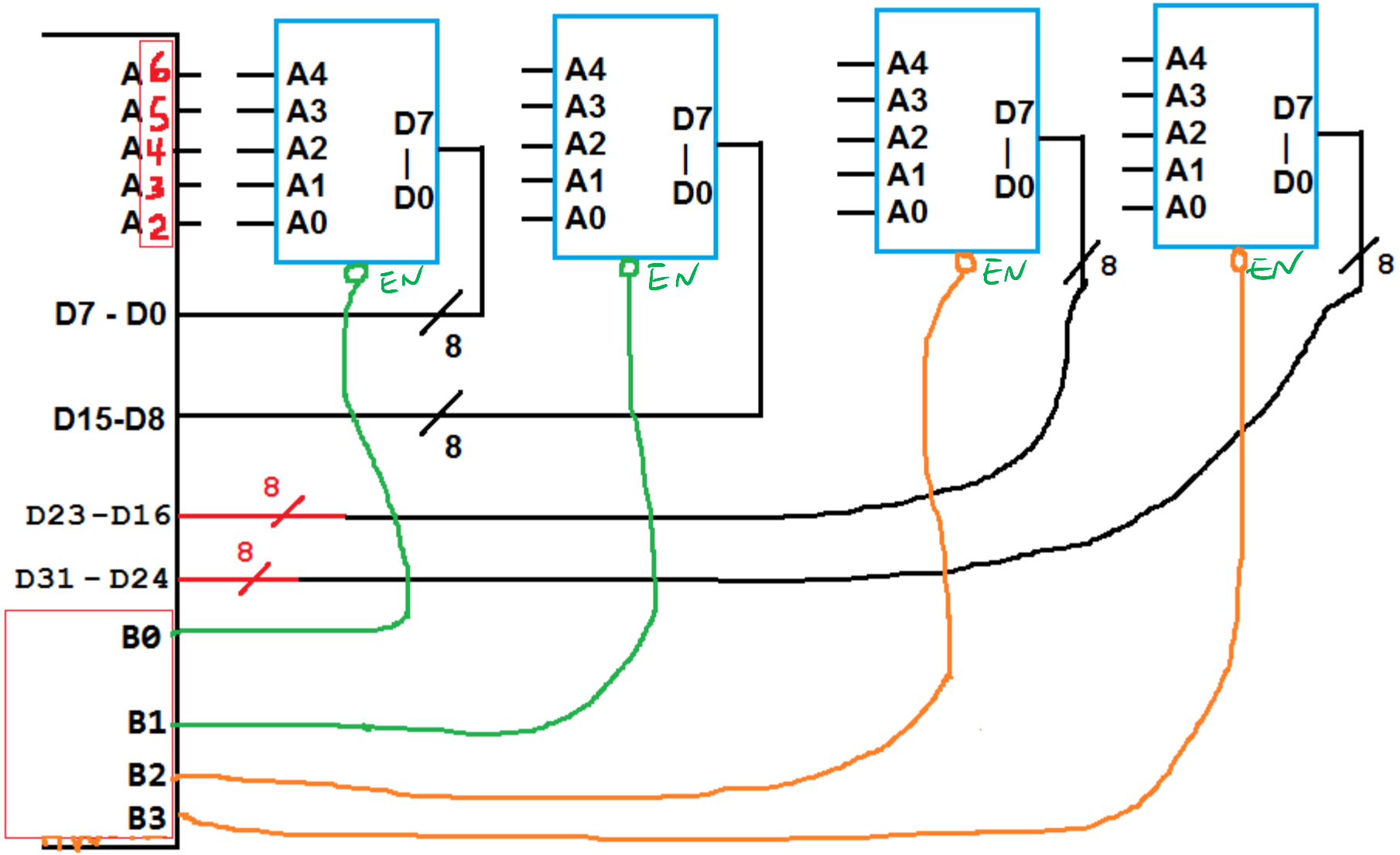
- With backward compatibility with 8-bit access





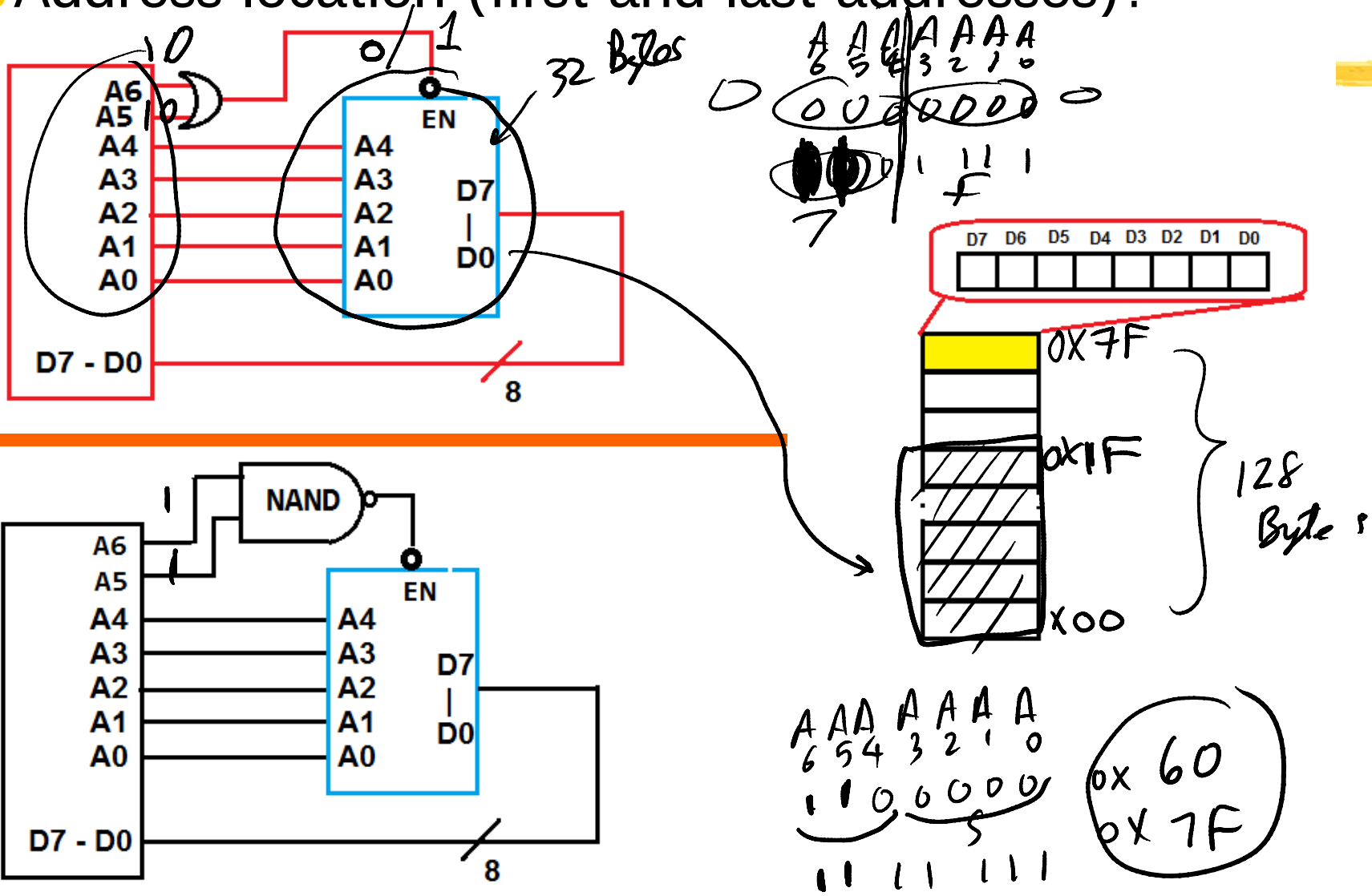


32-bit access with backward compatibility

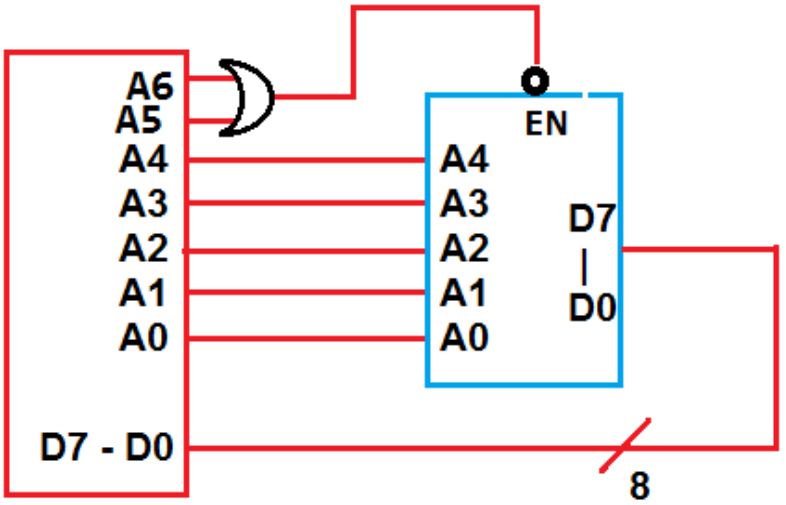


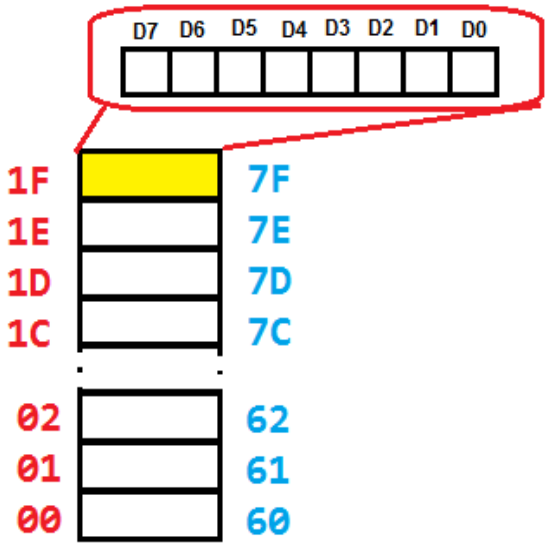
Memory Address 8

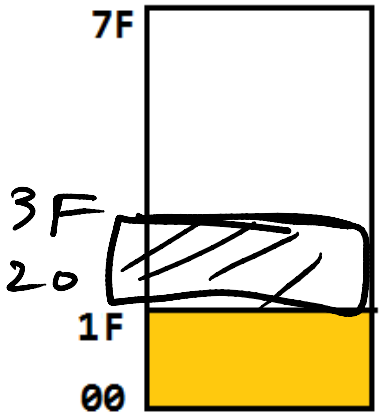
- Address location (first and last addresses)?

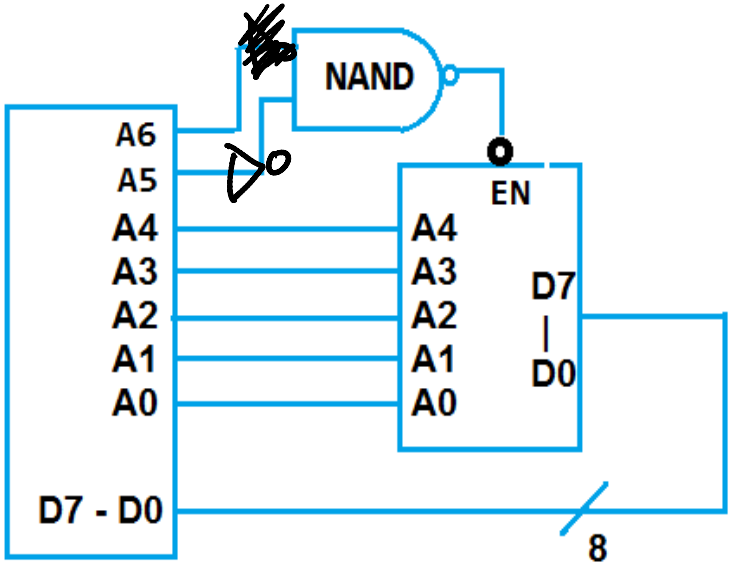


Memory Address 8



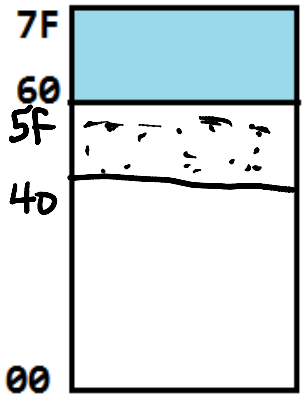






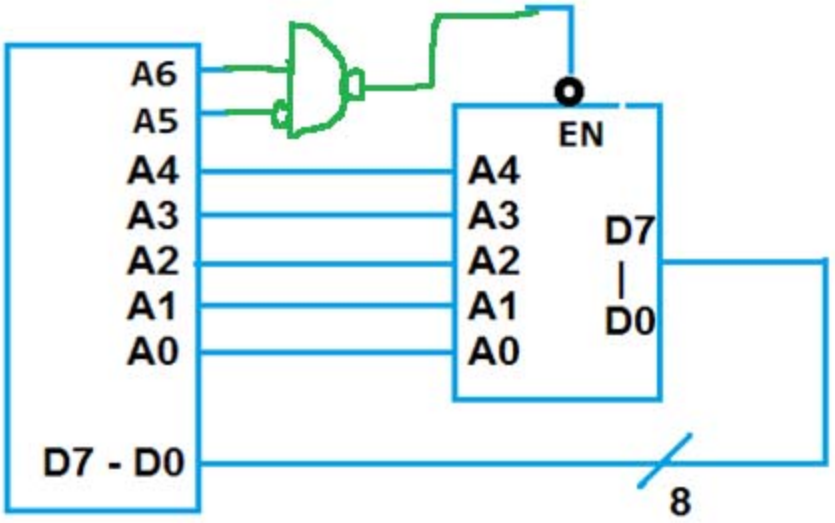
01

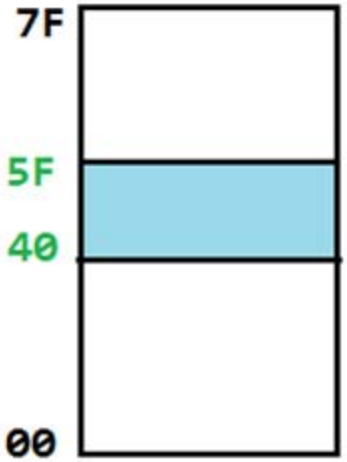
A6	A5	A4	A3	A2	A1	A0		
0	0	X	X	X	X	X	000 0000	00
							001 1111	1F
1	1	X	X	X	X	X	110 0000	60
							111 1111	7F



Memory Address 8 ---?

- Design a Memory Decoder so that the 32Byte memory locates as depicted.

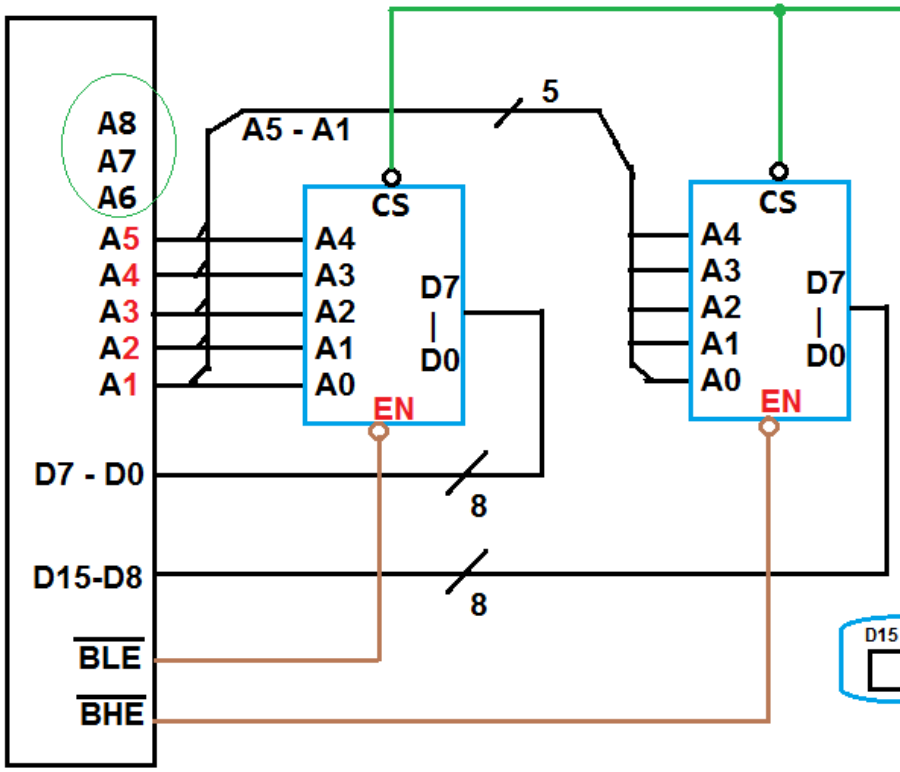


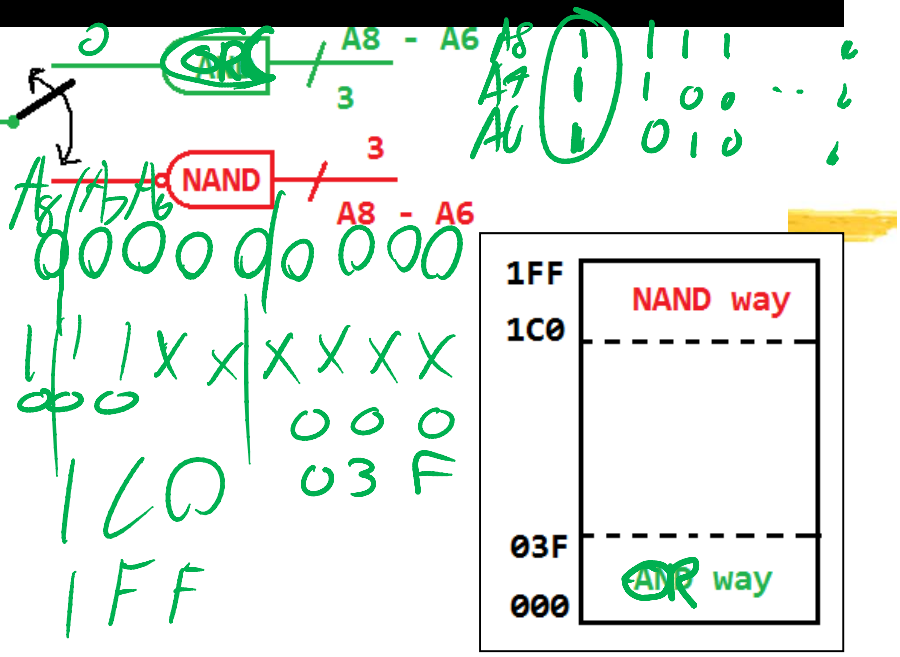


A6	A5	A4	A3	A2	A1	A0	
		X	X	X	X	X	
1	0	1	1	1	1	1	Higest Address
1	0	0	0	0	0	0	Lowest Address

Direct Connection between Process and Memory

Memory Address 16





A A A A A A A A H/L		
8 7 6 5 4 3 2 1		
0 0 0 X X X X X	--> 0 0000 0000	000
	0 0011 1111	03F
1 1 1 X X X X X	--> 1 1100 0000	1C0
	1 1111 1111	1FF