EECE416 Microcomputer Fundamentals & Design

Computer Architecture

Dr. Charles Kim

Howard University

https://www.hackster.io/hardwareweekend#what

Hackster Hardware Weekend

Partners | Mentors and speakers | What to expect | Schedule

Back to hackster.io

Register

Hackster Hardware Weekend

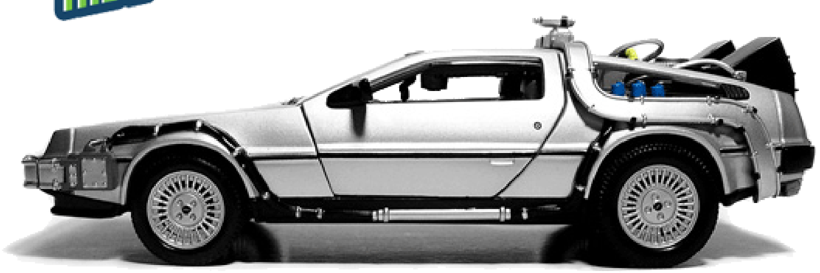
10 Cities | 10 Hackathons | 10 Meetups | 1 DeLorean

Ready to hack a better future?

The Hackster Hardware Weekend is a celebration of makers, open source hardware and the Hackster community across America. From Seattle to NYC and everything in between, we are hitting the road driving an original DeLorean DMC 12 to run the coolest Hackathon & Meetup series of the year, with loads of hardware kits, amazing speakers, software freebies and great people. Our goal is simple: educate, connect and help people invent and create the seeds of a better future.

What will you build with the DeLorean? **Share your ideas on the DeLorean projects page.**

HACK TO THE FUTURE



Schedule

Back to hackster.io

Register

Day 1

10:00 am	Check-in and breakfast
11:00 am	Kickoff
11:30 am	Training workshops
12:30 pm	Lunch served
1:00 pm	Start hacking
8:00 pm	Dinner served
12:00 am	Go home and get some rest

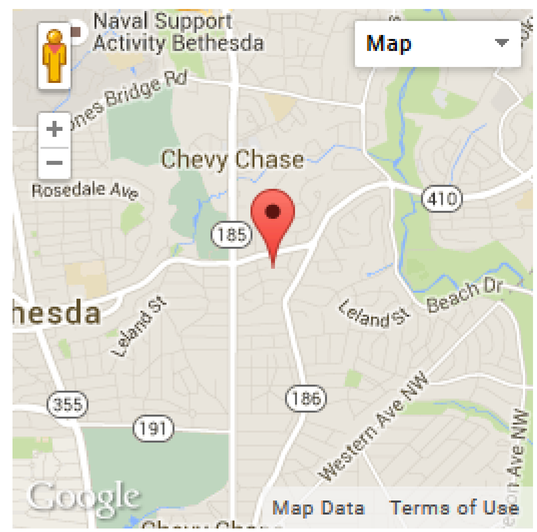
Day 2

9:00 am	Work on your hacks and get breakfast
12:30 pm	Lunch break with industry speaker
4:00 pm	Upload your hacks to Hackster
5:30 pm	Demo your hacks!
6:30 pm	Dinner served while judges deliberate
7:00 pm	Winners announced, prizes awarded
8:00 pm	Day two ends, thanks for participating!

5404 Wisconsin Ave, Suite 700 (Microsoft)
Chevy Chase
Chevy Chase, MD 20815-3522

Saturday, September 19, 2015 at 10:00 AM - Sunday, September 20, 2015 at 8:00 PM (EDT)

When & Where



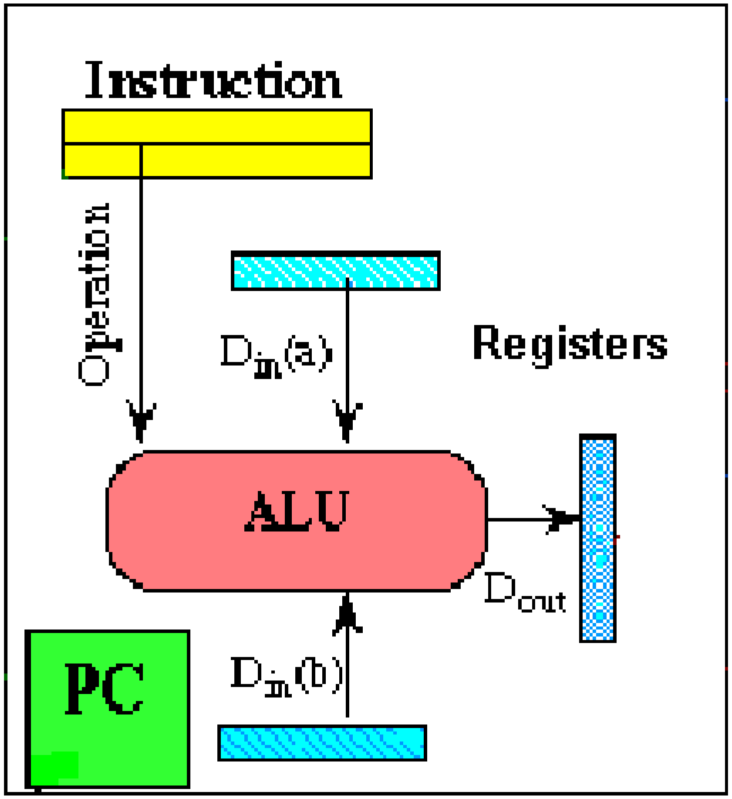
“Computer Architecture”

- 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

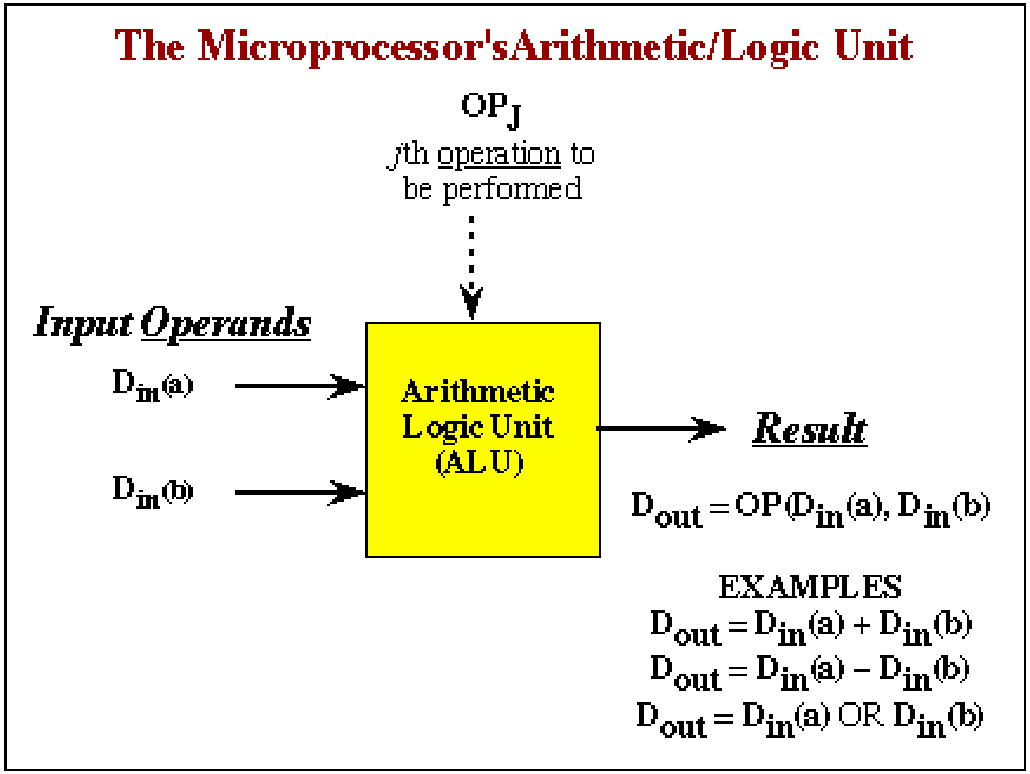
- **Computer System**
  - CPU (with PC, Register, SR) + Memory
- **Micro-Architecture:**
  - “conceptual design and fundamental operational structure of a computer system”
  - “blueprint and functional description of requirements and design implementations of a computer”
  - focusing on the way the CPU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

Microprocessor

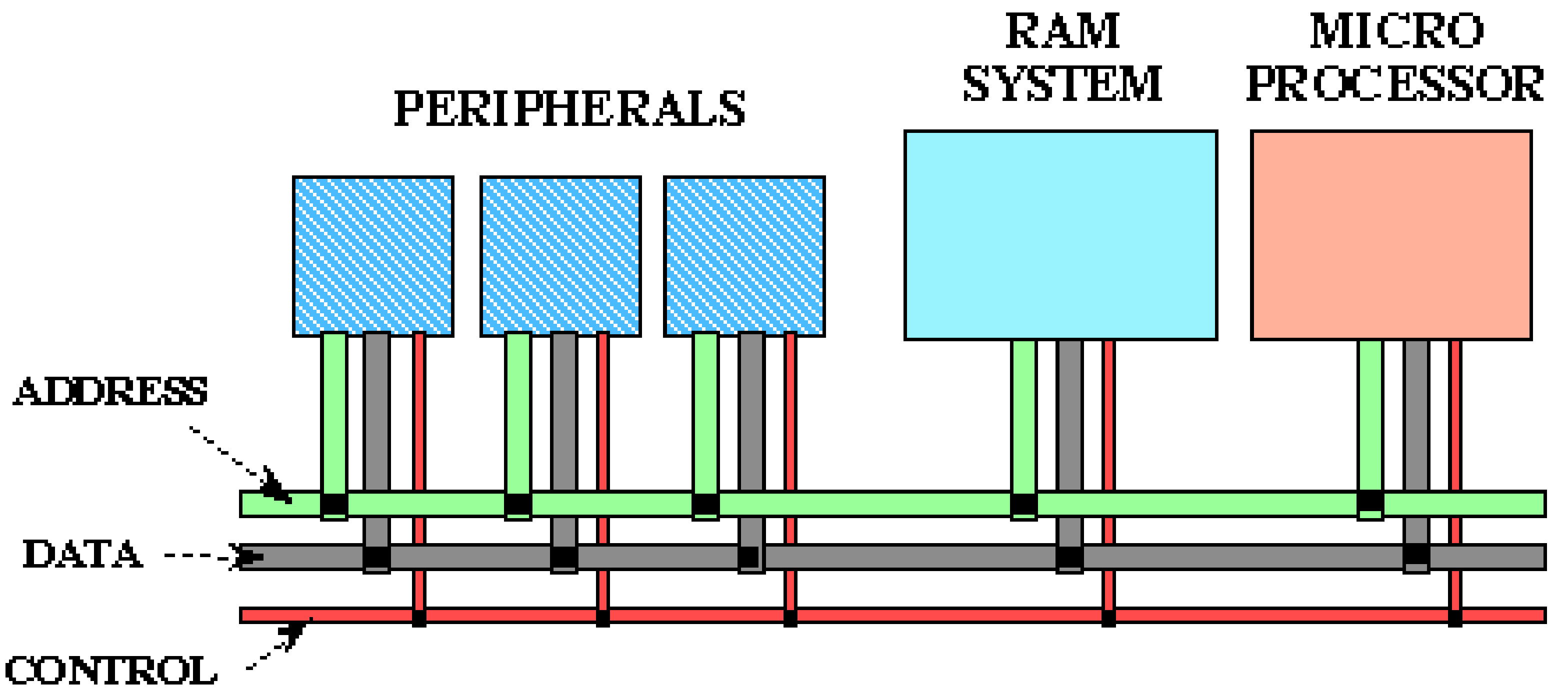


Micro-Architecture

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

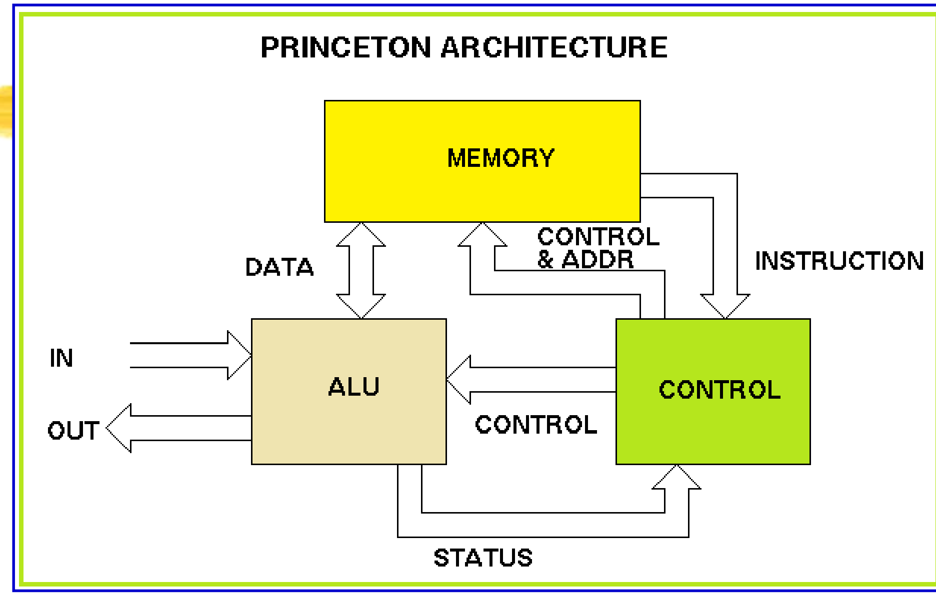


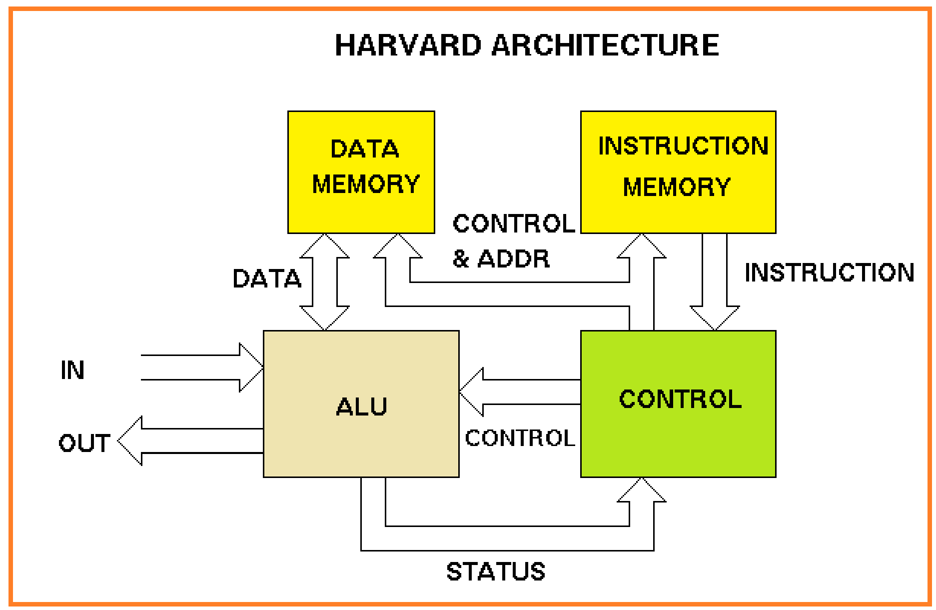
Microprocessor Bus



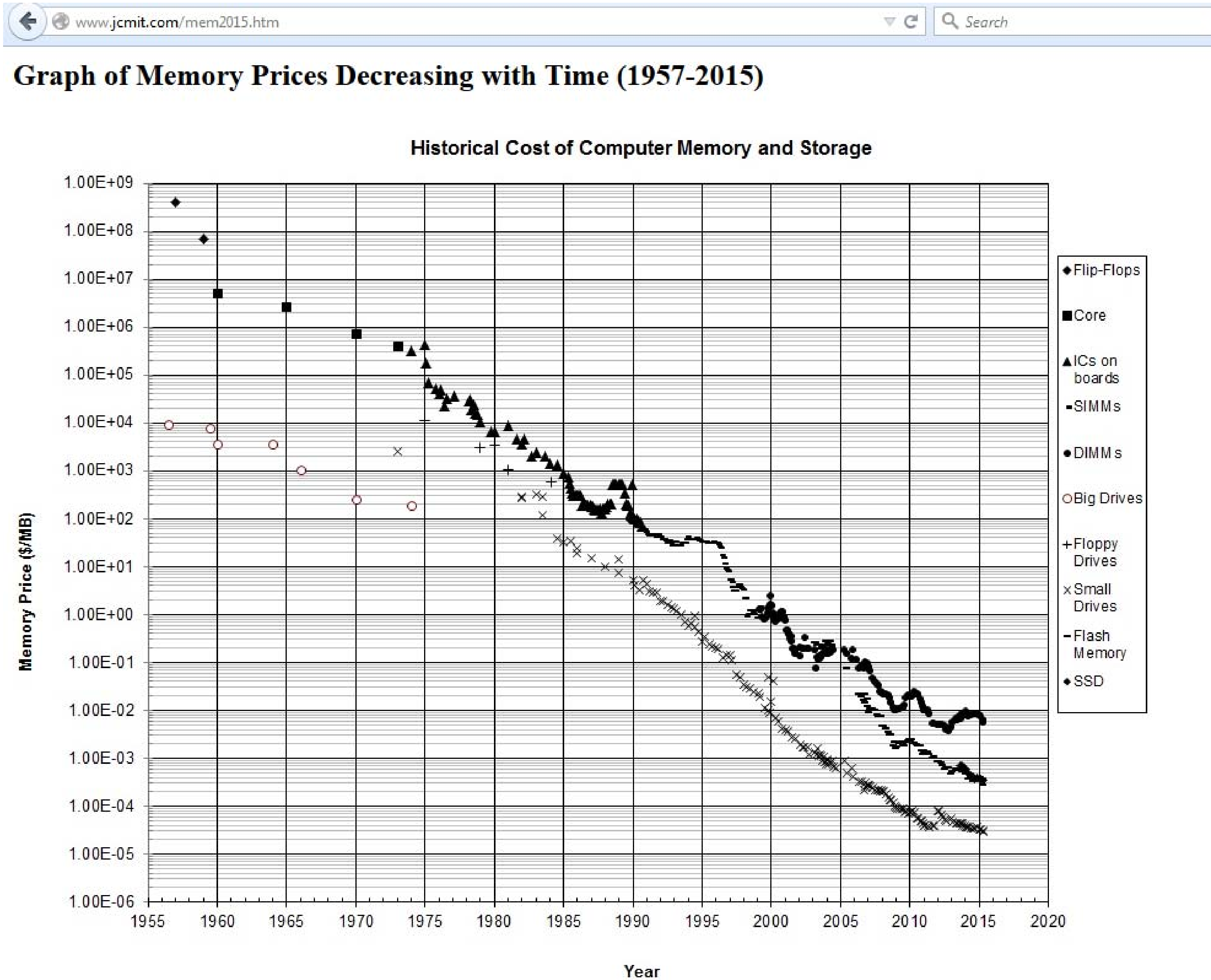
Architecture by CPU+MEM organization

- 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





Memory Price



Memory Price

www.jcmit.com/memoryprice.htm

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			

"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

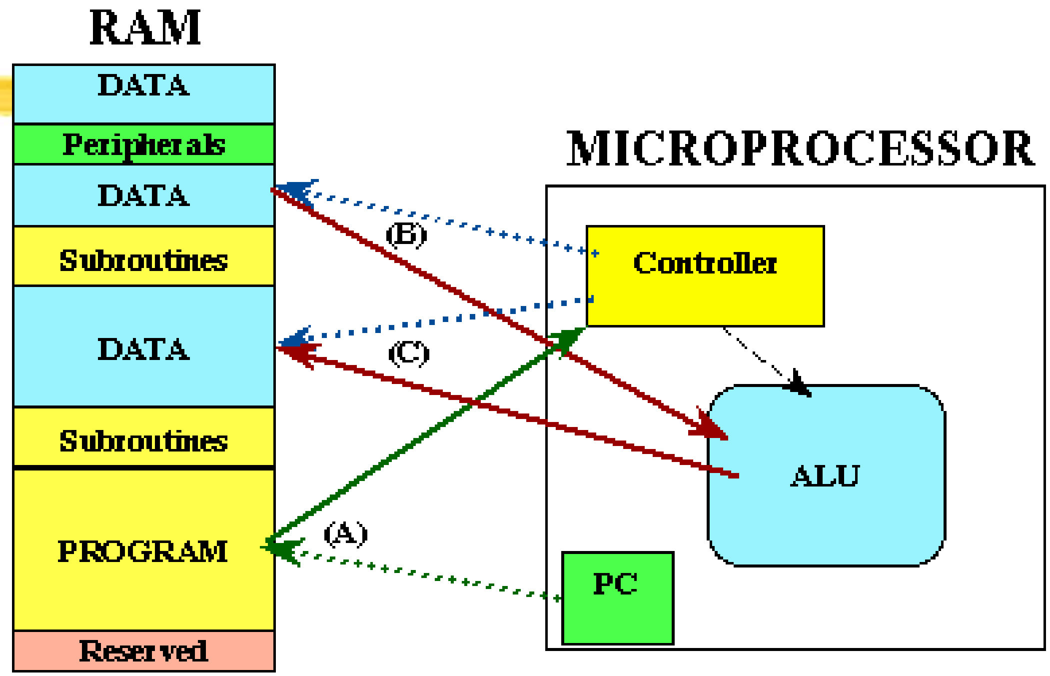
Basic five-stage pipeline in a RISC machine (IF = Instruction Fetch, ID = Instruction Decode, EX = Execute, MEM = Memory access, WB = Register write back). In the fourth clock cycle (the green column), the earliest instruction is in MEM stage, and the latest instruction has not yet entered the pipeline.

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.

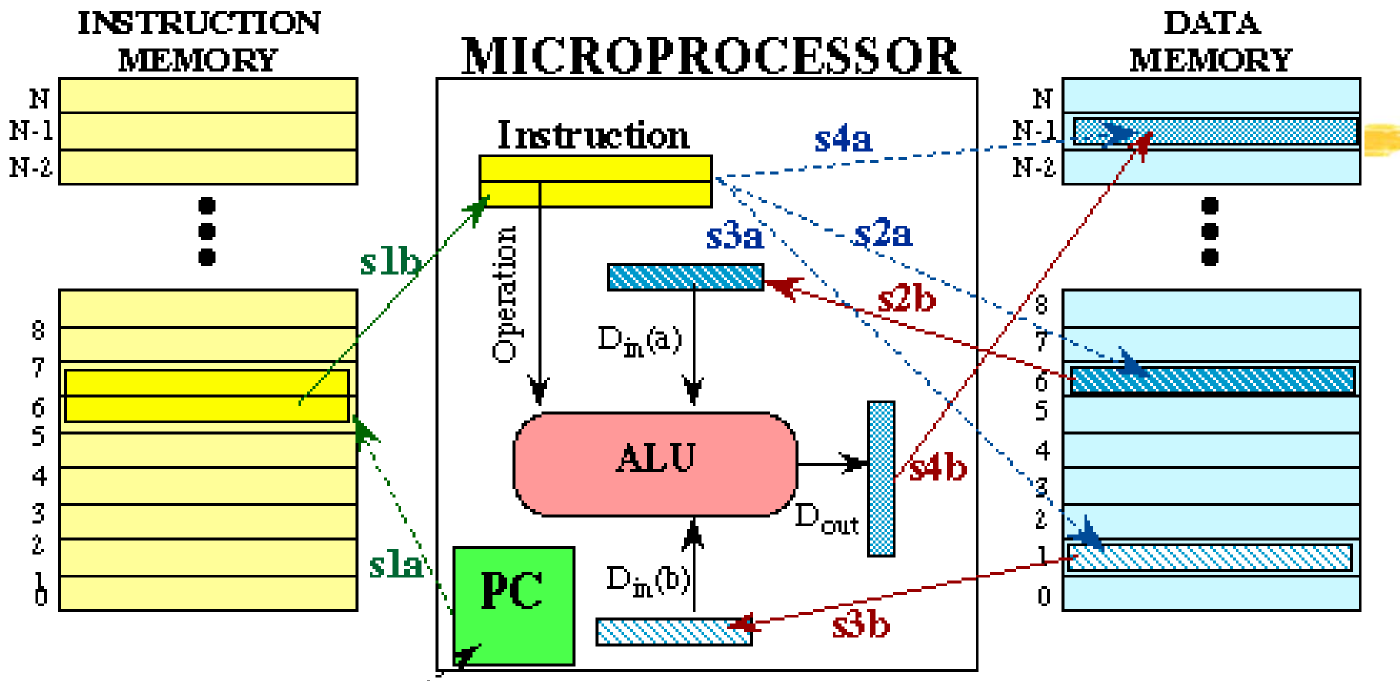


•CPU can be either reading an instruction or reading/writing data from/to the memory.

•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 <u>instruction fetches and data access do not contend</u> for a single memory pathway.



Address		Action
s1a	---------	s1b: get instruction
s2a	---------	s2b: get first data input
s3a	---------	s3b: get second data input
s4a	---------	s4b: store data output

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)

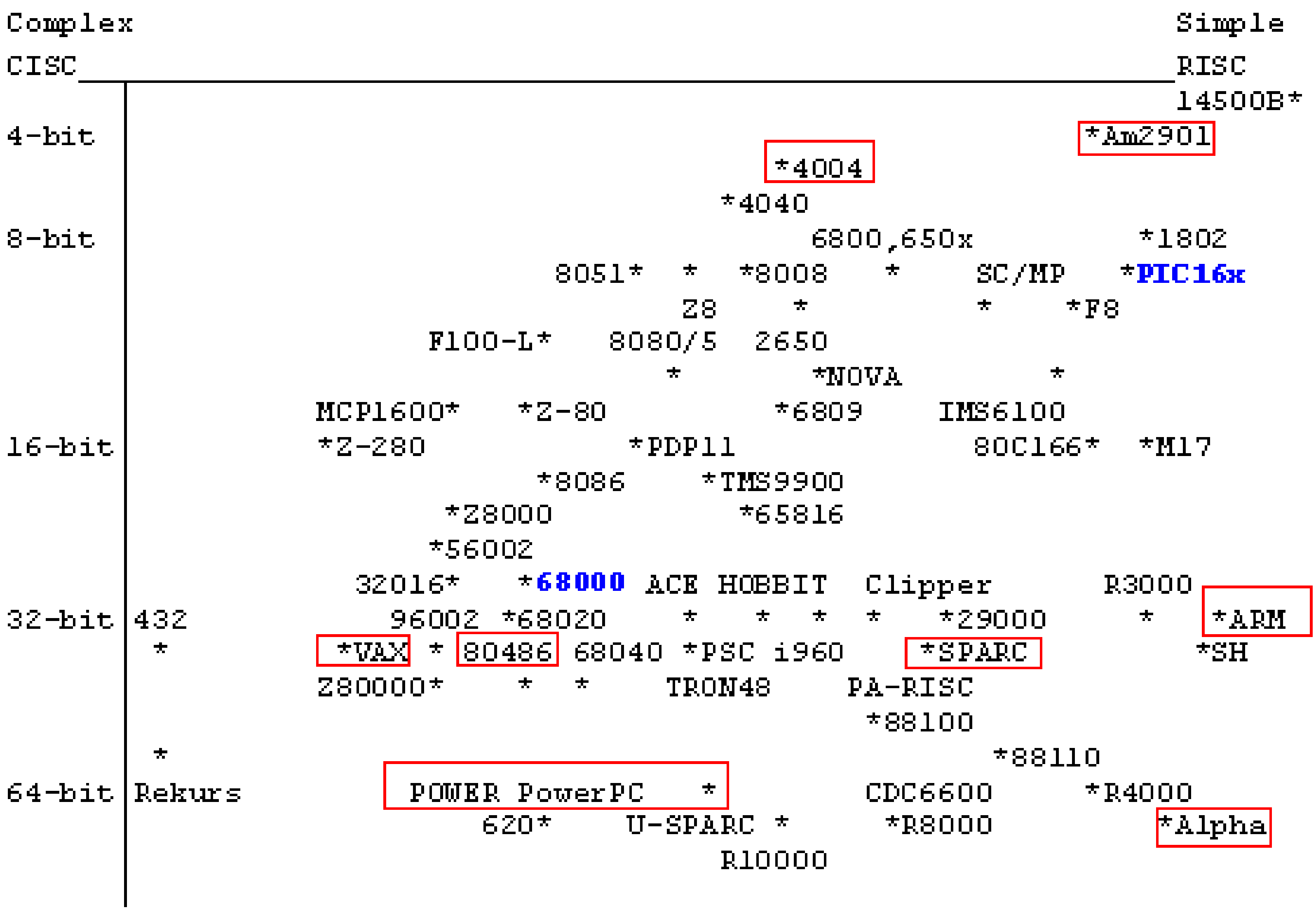
CISC

- 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



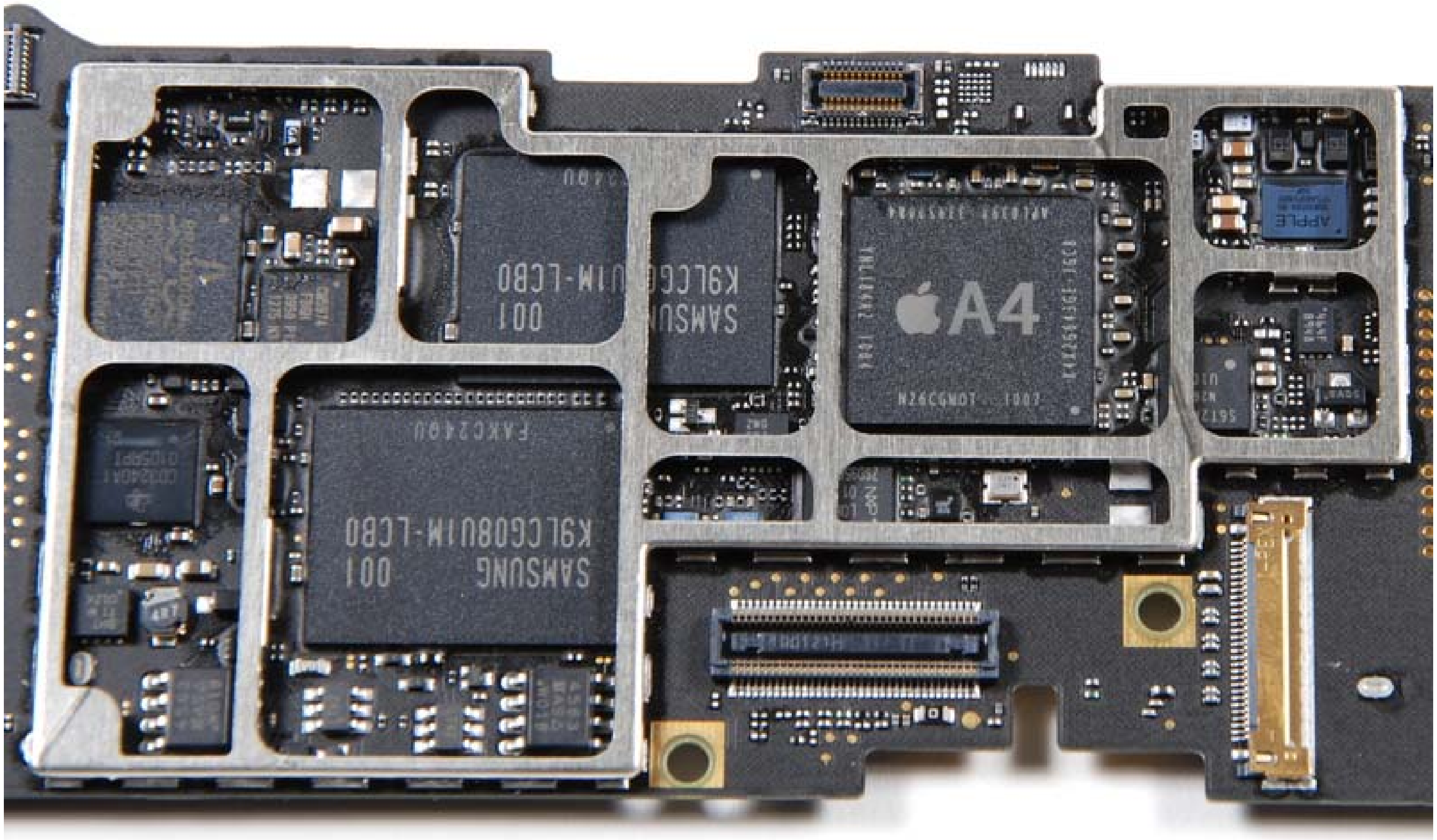
Intel inside?

- 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?

?

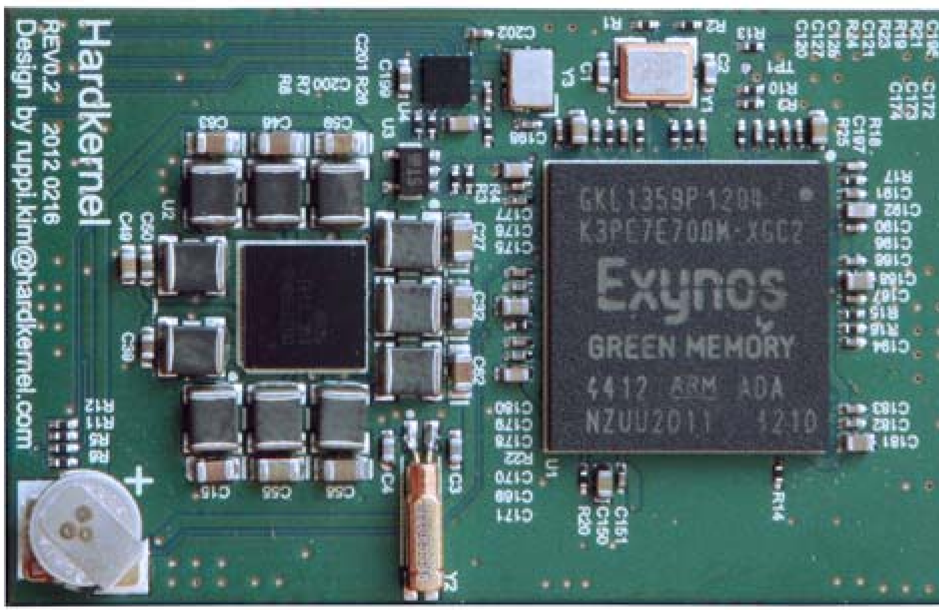
What's inside?

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



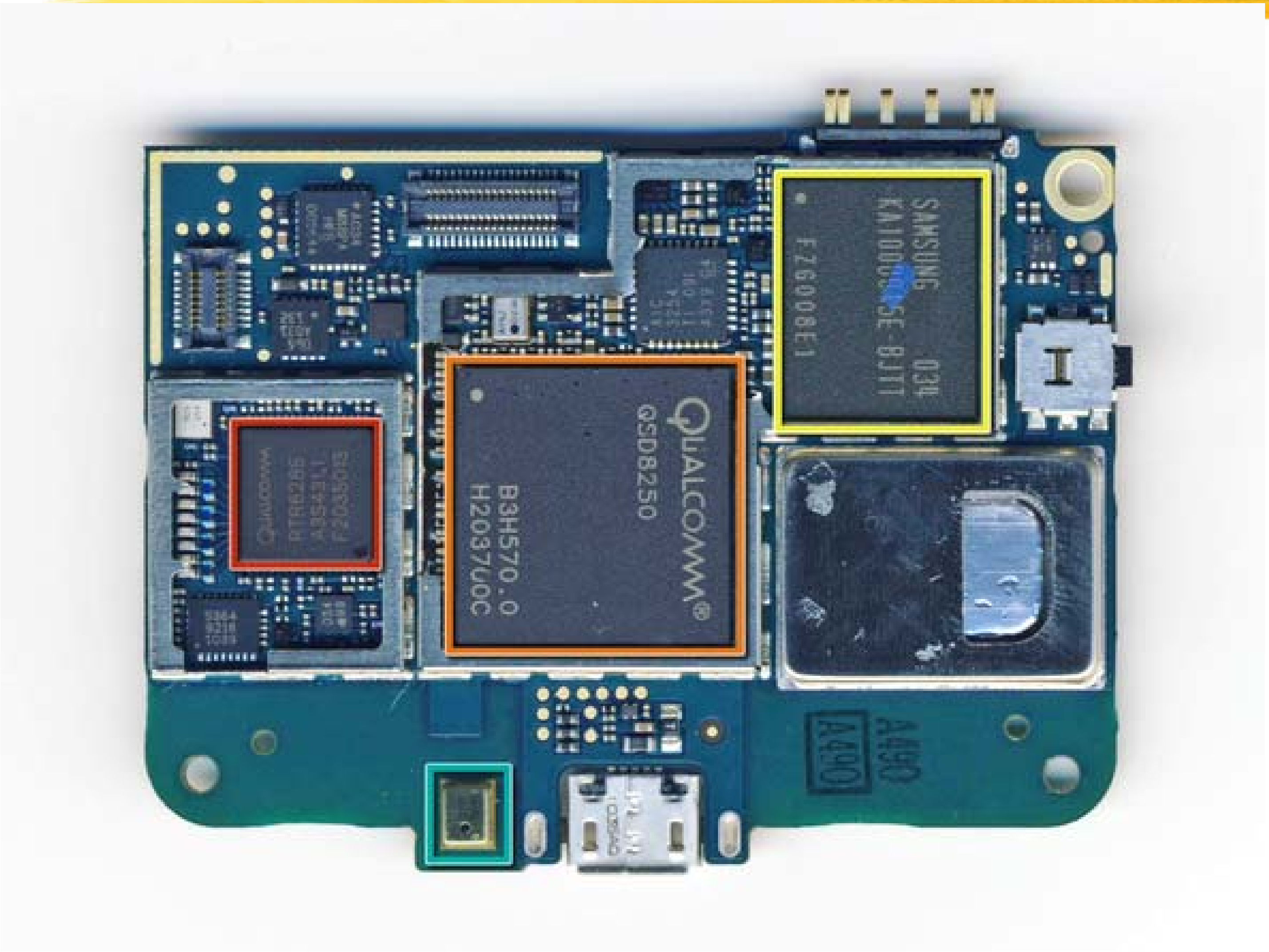
What's inside?

- 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

Intel Atom

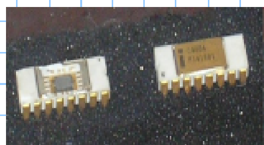
Produced	2008–present
Common manufacturer(s)	Intel
Max. CPU clock	800 MHz to 2 GHz
FSB speeds	400 MHz to 667 MHz
Min. feature size	45nm
Instruction set	x86, x86-64 (not for the N and Z series)
Cores	1, 2
Package(s)	441-ball μFCBGA
Core name(s)	Silverthorne Diamondville

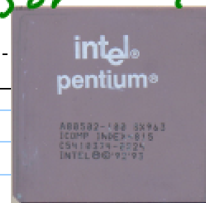
Intel Atom is the brand name for a line of x86 and x86-64 CPUs (or microprocessors) from Intel, designed in 45 nm CMOS and used mainly in Netbooks. The Atom Z series is code-named Silverthorne and the Atom N series is code-named Diamondville. As of June 2009, the most used chips in the Netbook retail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

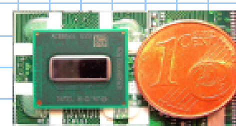
Intel Processor History

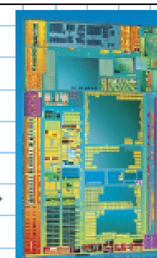
Note Title

8/30/2011









Embedded Atom processor

4-bit

P5

P6

Atom Z540 N270

core i7

"Tunnel Creek"

80632
80618

"Stellarton"

Tunnel Creek + FPGA

4004

8008

8080

8085

8086

8088

80286/80186

80386

486

Pentium

Pentium Pro

Pentium II

Celeron

Pentium III

Pentium 4

Itanium

Core Duo

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 3 4 5 6 7 8 9 0 1 2

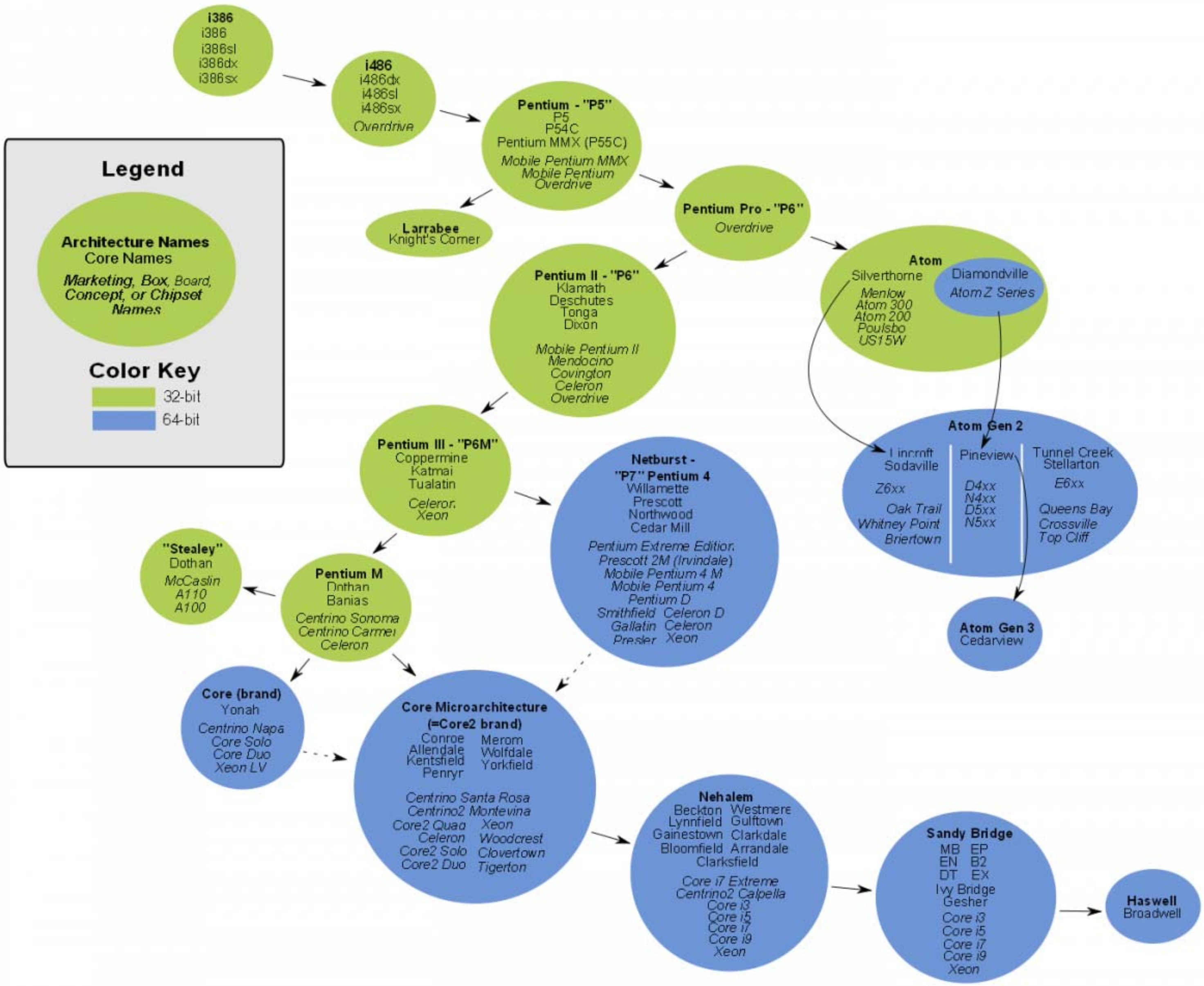
7 8 9 0 1

Processor Performance

Processor Performance Over Time and Other Key Features of the Intel Architecture

Intel Processor	Date of Product Intro-duction	Perfor-mance in MIPs[1]	Max. CPU Frequency at Intro-duction	No. of Transis-tors on the Die	Main CPU Register Size[2]	Extern. Data Bus Size[2]	Max. Extern. Addr. Space
8086	1978	0.8	8 MHz	29 K	16	16	1 MB
Intel 286	1982	2.7	12.5 MHz	134 K	16	16	16 MB
Intel386™ DX	1985	6.0	20 MHz	275 K	32	32	4 GB
Intel486™ DX	1989	20	25 MHz	1.2 M	32	32	4 GB
Pentium®	1993	100	60 MHz	3.1 M	32	64	4 GB
Pentium Pro	1995	440	200 MHz	5.5 M	32	64	64 GB

32/64-bit Intel Architectures and their Core Names and Code Names



Intel 805XX Product Codes

Intel 805xx product codes

Intel discontinued the use of part numbers such as 80486 in the marketing of mainstream x86-architecture microprocessors with the introduction of the Pentium brand in 1993. However, numerical codes, in the 805xx range, continued to be assigned to these processors for internal and part numbering uses. The following is a list of such product codes in numerical order:

Product code	Marketing name(s)	Codename(s)
80500	Pentium	P5 (A-step)
80501	Pentium	P5
80502	Pentium	P54C, P54CS
80503	Pentium with MMX Technology	P55C, Tillamook
80521	Pentium Pro	P6
80522	Pentium II	Klamath
80523	Pentium II, Celeron, Pentium II Xeon	Deschutes, Covington, Drake
80524	Pentium II, Celeron	Dixon, Mendocino
80525	Pentium III, Pentium III Xeon	Katmai, Tanner
80526	Pentium III, Celeron, Pentium III Xeon	Coppermine, Cascades
80528	Pentium 4, Xeon	Willamette (Socket 423), Foster
80529	*cancelled*	Timna
80530	Pentium III, Celeron	Tualatin

Intel 805XX Product Codes

80531	Pentium 4, Celeron	Willamette (Socket 478)
80532	Pentium 4, Celeron, Xeon	Northwood, Prestonia, Gallatin
80533	Pentium III	Coppermine (cD0-step)
80534	Pentium 4 SFF	Northwood (small form factor)
80535	Pentium M, Celeron M 310–340	Banias
80536	Pentium M, Celeron M 350–390	Dothan
80537	Core 2 Duo T5xxx, T7xxx, Celeron M 5xx	Merom
80538	Core Solo, Celeron M 4xx	Yonah
80539	Core Duo, Pentium Dual-Core T-series	Yonah
80541	Itanium	Merced
80542	Itanium 2	McKinley
80543	Itanium 2	Madison
80546	Pentium 4, Celeron D, Xeon	Prescott (Socket 478), Nocona, Irwindale, Cranford, Potomac
80547	Pentium 4, Celeron D	Prescott (LGA 775)
80548	*canceled*	Tejas and Jayhawk
80549	Itanium 2 90xx	Montecito

Intel 805XX Product Codes

80550	Dual-Core Xeon 71xx	Tulsa
80551	Pentium D, Pentium EE, Dual-Core Xeon	Smithfield, Paxville DP
80552	Pentium 4, Celeron D	Cedar Mill
80553	Pentium D, Pentium EE	Presler
80554	Celeron 800/900/1000 ULV	Shelton
80555	Dual-Core Xeon 50xx	Dempsey
80556	Dual-Core Xeon 51xx	Woodcrest
80557	Core 2 Duo E4xxx. E6xxx, Dual-Core Xeon 30xx, Pentium Dual-Core E2xxx	Conroe
80560	Dual-Core Xeon 70xx	Paxville MP
80562	Core 2 Quad, Core 2 Extreme QX6xxx, Quad-Core Xeon 32xx	Kentsfield
80563	Quad-Core Xeon 53xx	Clovertown
80564	Xeon 7200	Tigerton-DC
80565	Xeon 7300	Tigerton
80566	Atom Z5xx	Silverthorne
80567	Itanium 91xx	Montvale
80569	Core 2 Quad Q9xxx, Core 2 Extreme QX9xxx, Xeon 33xx	Yorkfield

Intel 805XX Product Codes

80570	Core 2 Duo E8xxx, Xeon 31xx	Wolfdale
80571	Core 2 Duo E7xxx, Pentium Dual-Core E5xxx, Pentium Dual-Core E2210	Wolfdale-3M
80573	Xeon 5200	Wolfdale-DP
80574	Core 2 Extreme QX9775, Xeon 5400	Harpertown
80576	Core 2 Duo P7xxx, T8xxx, P8xxx, T9xxx, P9xxx, SL9xxx, SP9xxx, Core 2 Extreme X9xxx	Penryn
80577	Core 2 Duo P7xxx, P8xxx, SU9xxx, T6xxx, T8xxx	Penryn-3M
80578	LE80578	Vermilion Range
80579	EP80579	Tolapai
80580	Core 2 Quad Q8xxx, Q9xxx, Xeon 33xx	Yorkfield-6M
80581	Core 2 Quad Q9xxx	Penryn-QC
80582	Xeon 74xx	Dunnington
80583	Xeon 74xx	Dunnington-QC
80584	Xeon X33x3 LV	Yorkfield CL
80585	Core 2 Solo SU3xxx, Celeron 7xx, 9xx	Penryn-L
80586	Atom 2xx, N2xx	Diamondville
80587	Atom 3xx	Diamondville DC
80588	Xeon L3014, E3113	Wolfdale-CL

Intel 806XX Product Codes

Intel 806xx product codes

Product code	Marketing name(s)	Codename(s)
80601	Core i7, Xeon 35xx	Bloomfield
80602	Xeon 55xx	Gainestown
80603	Itanium 93xx	Tukwila
80604	Xeon 65xx, Xeon 75xx	Beckton
80605	Core i5-7xx, Core i7-8xx, Xeon 34xx	Lynnfield
80606	*canceled*	Havendale
80607	Core i7-7xx QM, Core i7-8xx QM, Core i7-9xx XM	Clarksfield
80608	*canceled*	Auburndale
80609	Atom	Lincroft
80610	Atom N400, D400, D500	Pineview

Intel 806XX Product Codes

80611	*canceled*	Larrabee
80612	Xeon C35xx, Xeon C55xx	Jasper Forest
80613	Core i7-9xxX, Xeon 36xx	Gulftown
80614	Xeon 56xx	Westmere-EP
80615	Xeon E7-28xx, Xeon E7-48xx	Westmere-EX
80616	Pentium G6xxx, Core i3-5xx, Core i5-6xx	Clarkdale
80617	Core i5-5xx, Core i7-6xxM/UM/LM	Arrandale
80618	Atom	Tunnel Creek
80620	Xeon	Sandy Bridge-EP-8, Sandy Bridge-EP-4
80621	Xeon	Sandy Bridge-EP-8, Sandy Bridge-EP-4
80622	Xeon	Sandy Bridge-EP-8
80623	Xeon E3-xxxx, Core i3/i5/i7-2xxx, Pentium Gxxx	Sandy Bridge-HE-4, Sandry Bridge-M-2
80627	Core i3/i5/i7-2xxxM,, Pentium Bxxx, Celeron Bxxx	Sandy Bridge-HE-4, Sandy Bridge-H-2, Sandy Bridge-M-2
80632	Atom	Tunnel Creek
80640	Atom	Penwell
80641	Atom	Cedar View

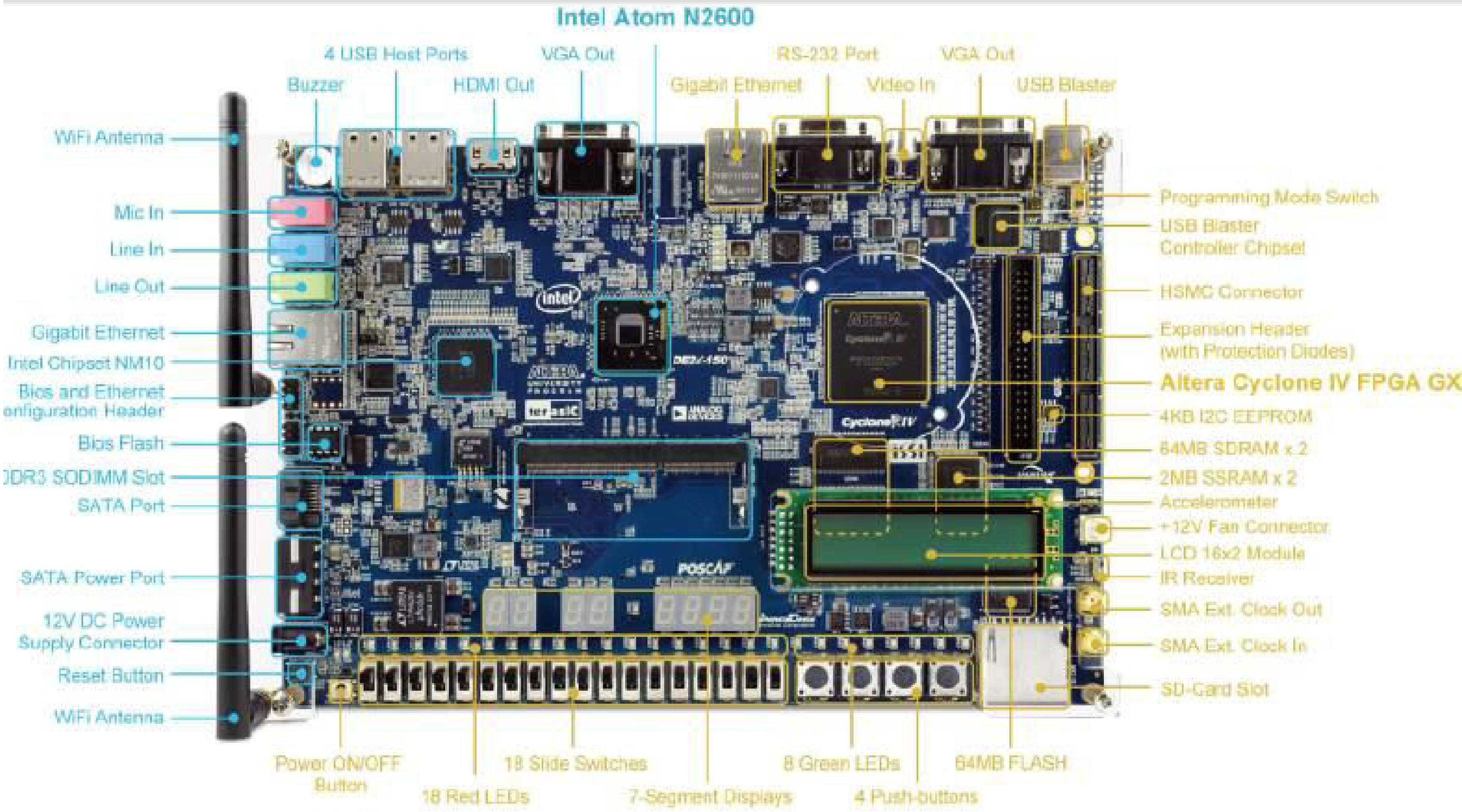
New Intel Atom DE2i-150 Board

DE2i-150 Kit Contents

- Development Board
- System CD
- Quartus II CD
- Quick Start Guide
- USB Cable
- Power Cable
- IR Remote
- Loopback Board

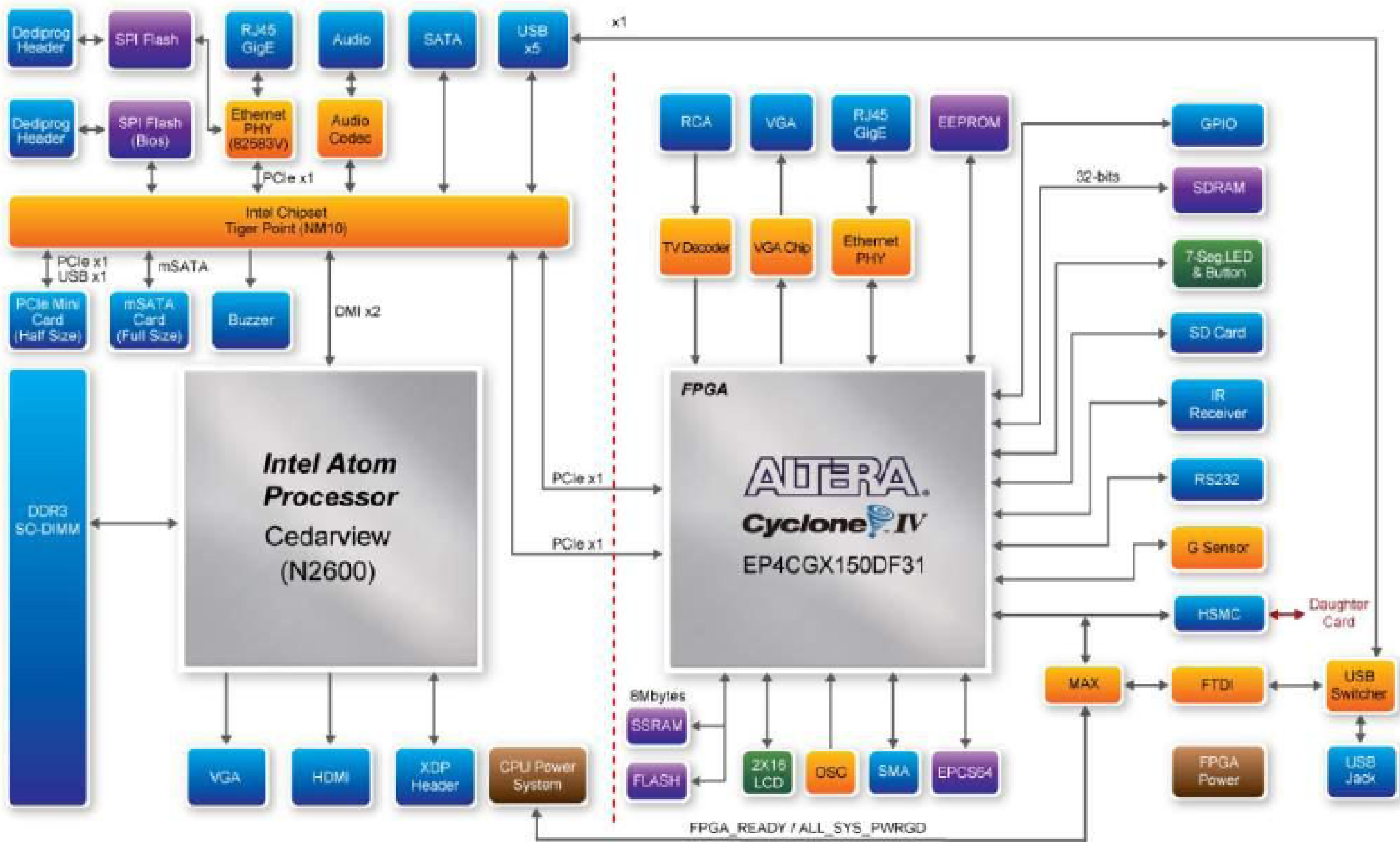
DE2i-150 Kit

DE2i-150 Floorplan

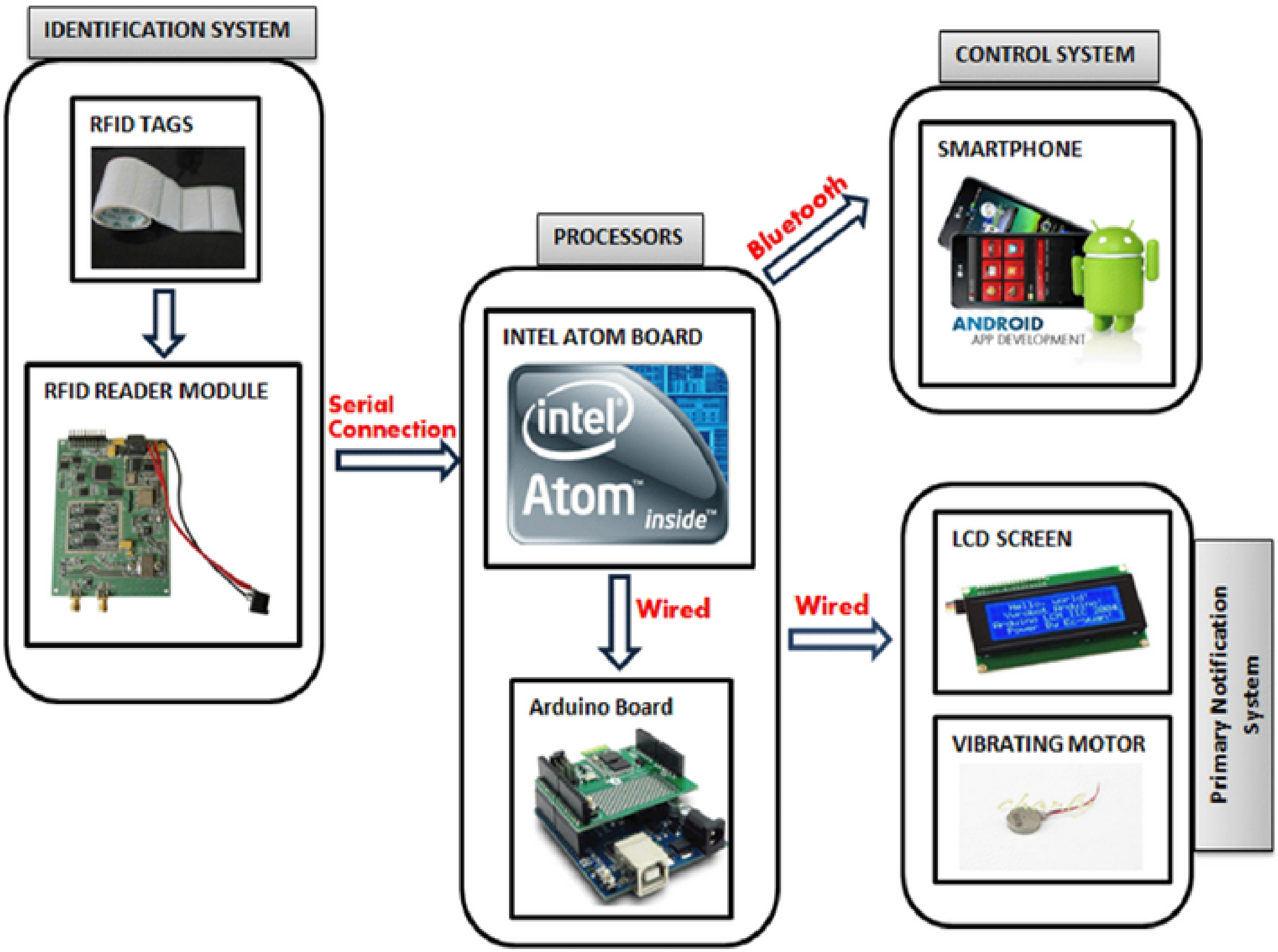


DE2i-150 Kit

DE2i-150 Block Diagram



What/How HU students did? An example



Intel 386 - Datasheet

Intel386™ SX MICROPROCESSOR

DIR 16 Page 12 Phys 24

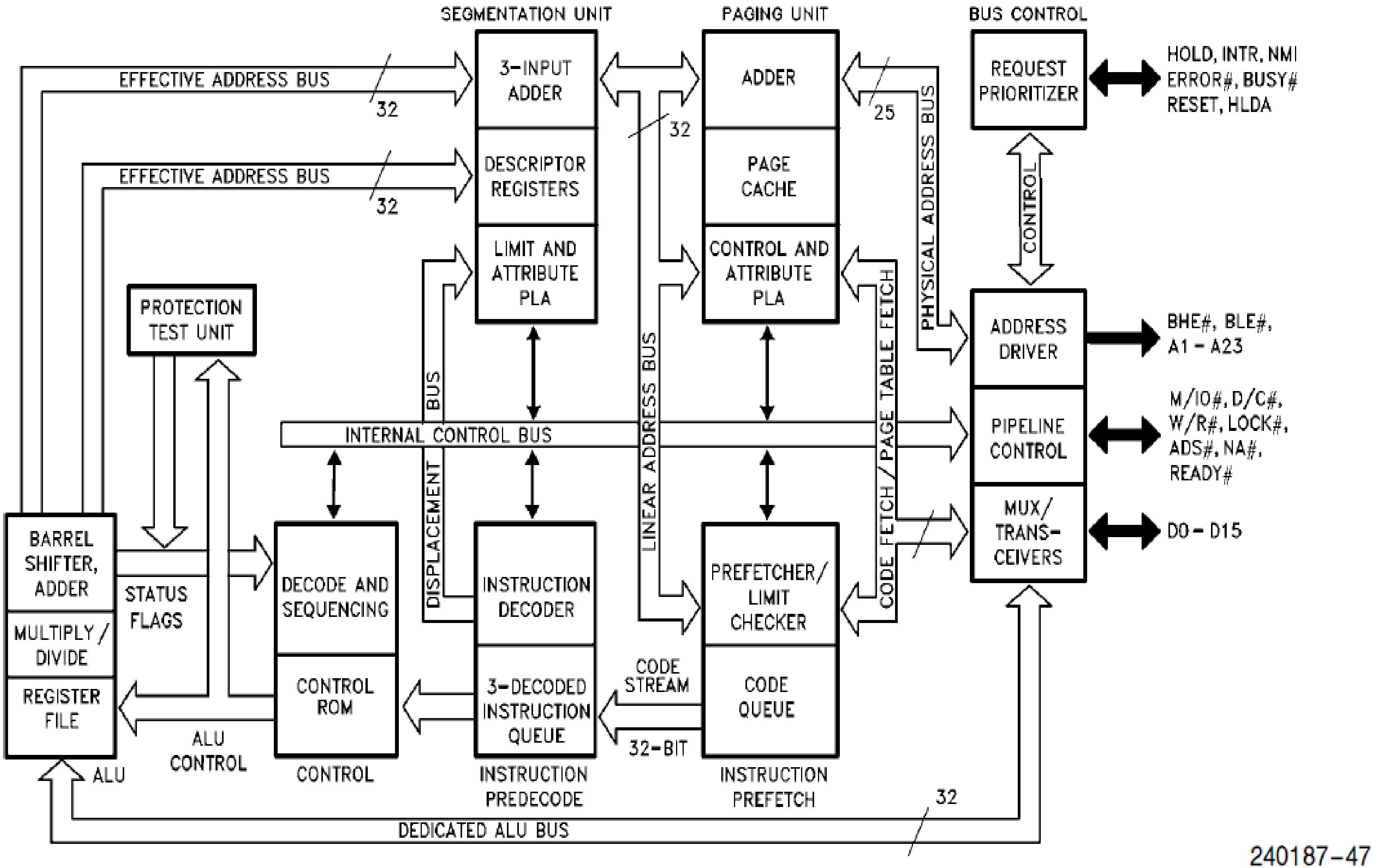
- Full 32-Bit Internal Architecture
  - 8-, 16-, 32-Bit Data Types
  - 8 General Purpose 32-Bit Registers
- Runs Intel386™ Software in a Cost Effective 16-Bit Hardware Environment
  - Runs Same Applications and O.S.'s as the Intel386™ DX Processor
  - Object Code Compatible with 8086, 80186, 80286, and 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

24

- Large Uniform Address Space
  - 16 Megabyte Physical
  - 64 Terabyte Virtual 46
  - 4 Gigabyte Maximum Segment Size 32
- 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)

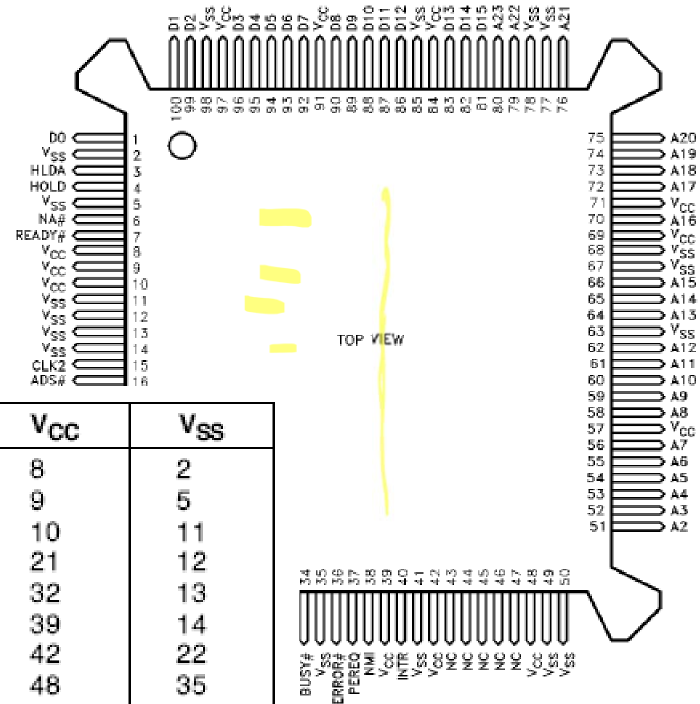
Intel 386 - Datasheet



Intel386™ SX Pipelined 32-Bit Microarchitecture

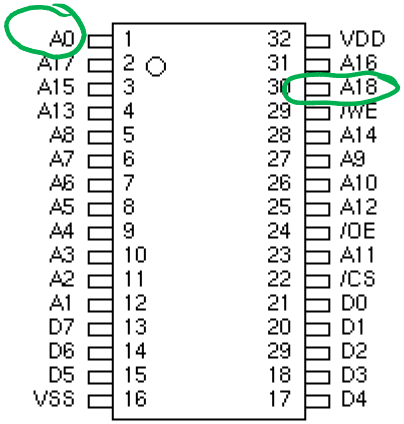
Intel 386 - Brief

- 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



# of Address Lines

2^n

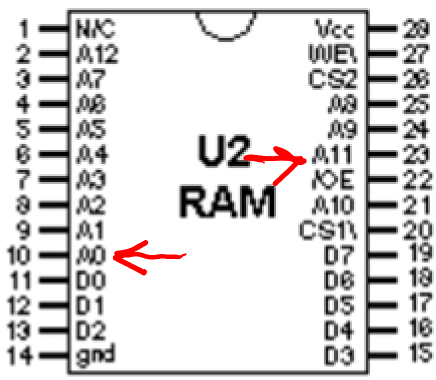
$2^{19} = 2^9 \cdot 2^{10} = 512\,KB$

2^{10} 1KB

2^{20} 1MB

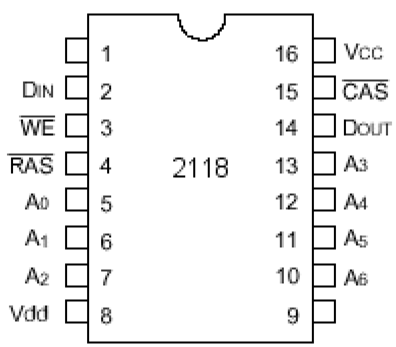
2^{30} 1GB

2^{40} 1TB



$2^{12} = 2^2 \cdot 2^{10} = 4\,KB$

serial memory

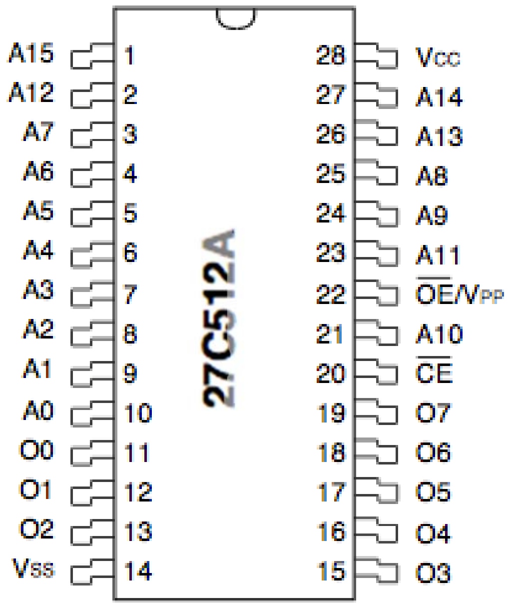


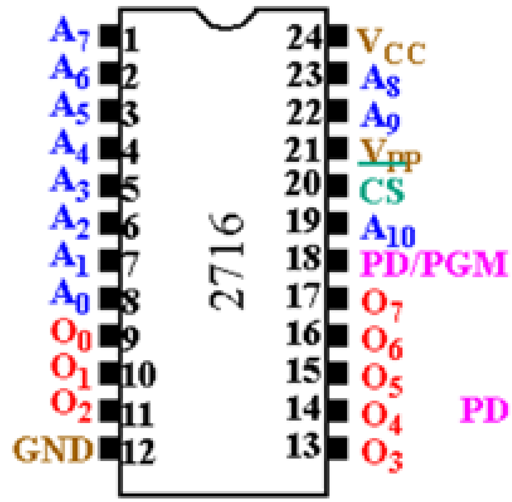
$2^7 = 128\,KB$

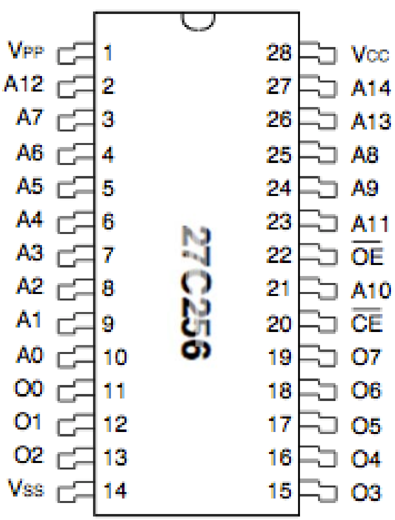
Review on Number Systems

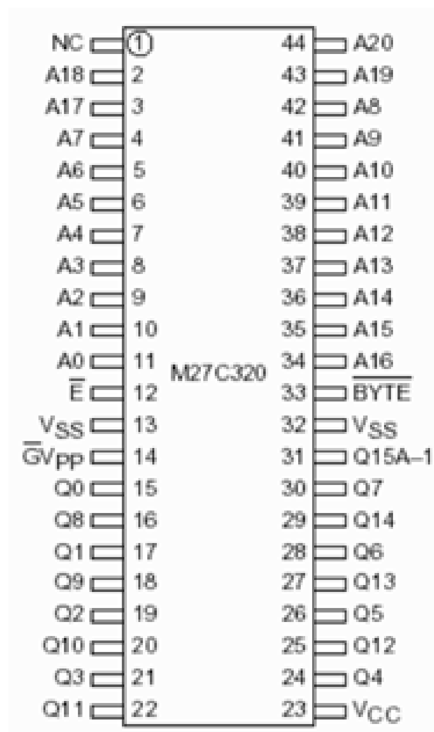
	Binary	Hexadecimal	Decimal
1.	100	________	________
2.	10101101	________	________
3.	1101110101	________	________
4.	11111011110	________	________
5.	10000000001	________	________
6.	________	8EF	________
7.	________	10	________
8.	________	A52E	________
9.	________	70C	________
10.	________	6BD3	________
11.	________	________	100
12.	________	________	527
13.	________	________	4128
14.	________	________	11947
15.	________	________	59020

Memory Size and Address 2



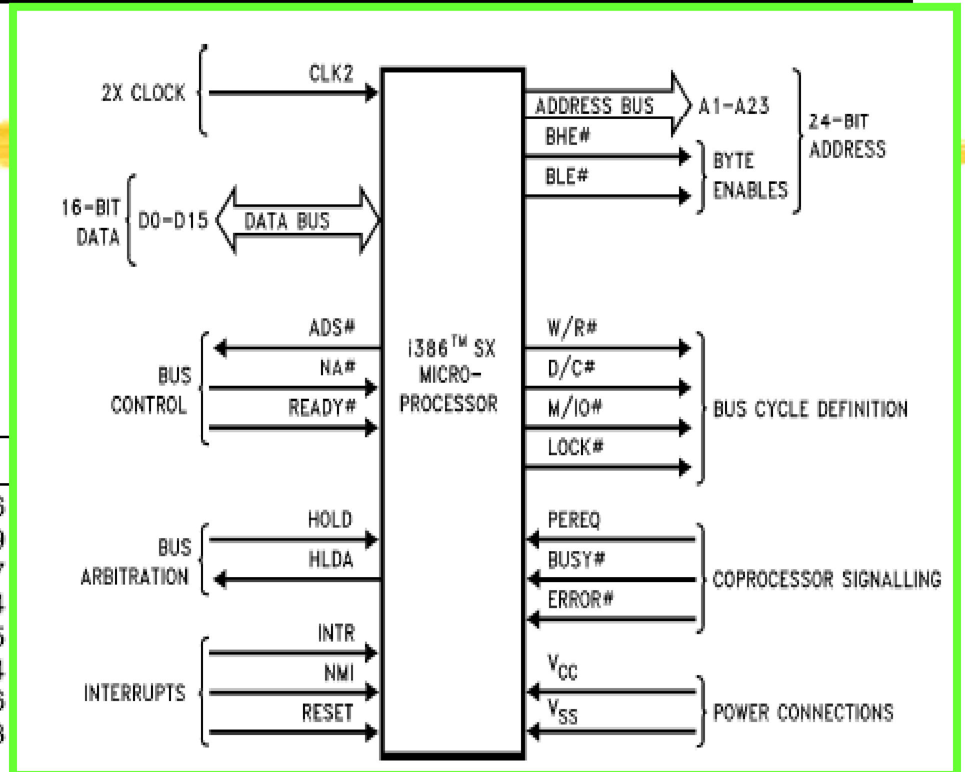






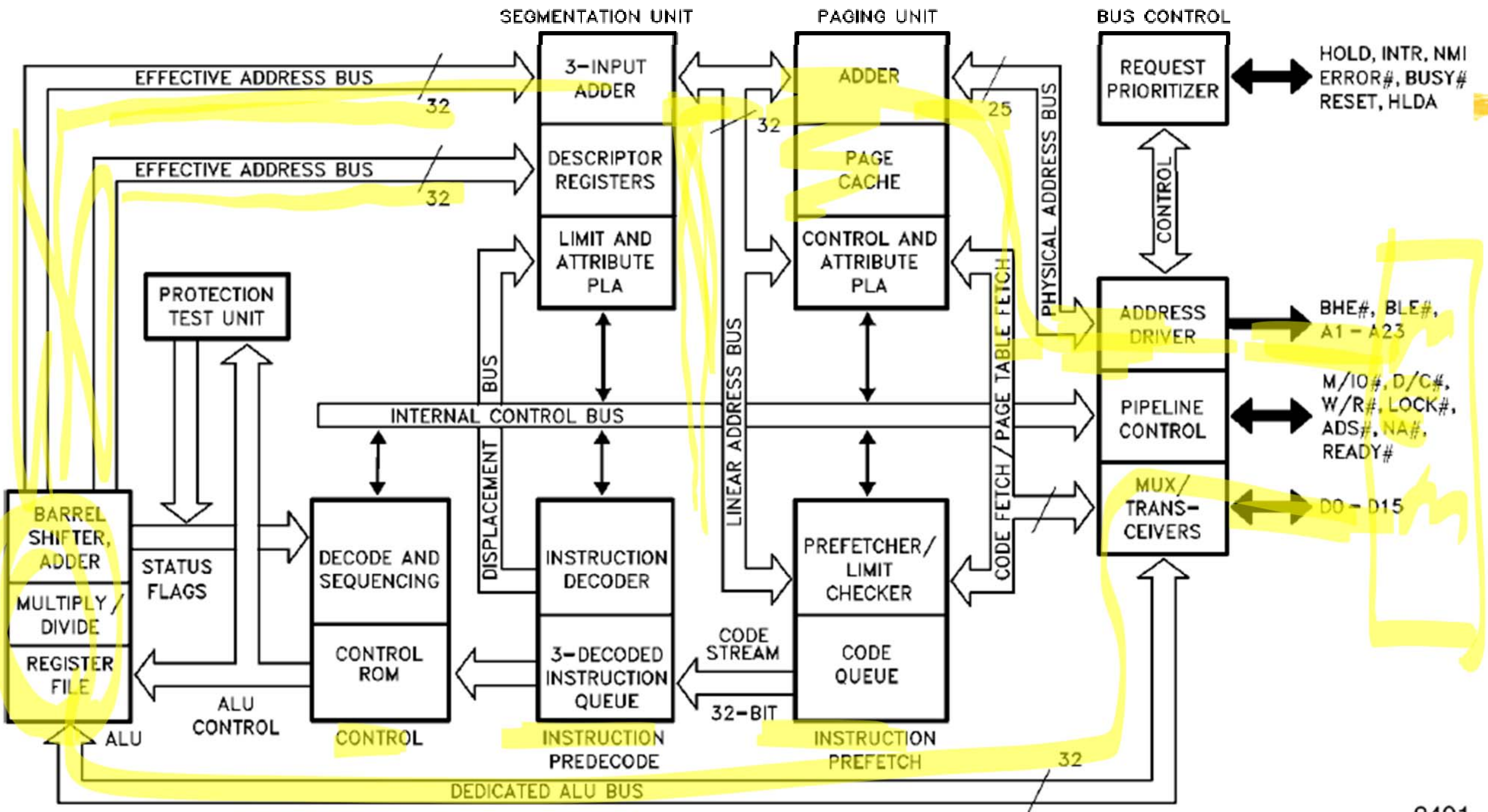
Intel 386 - Brief

- Address: A23- A1 (where is A0?)
  - BLE# and BHE# (“Byte Enable”)
- Data: D15 – D0
- Control



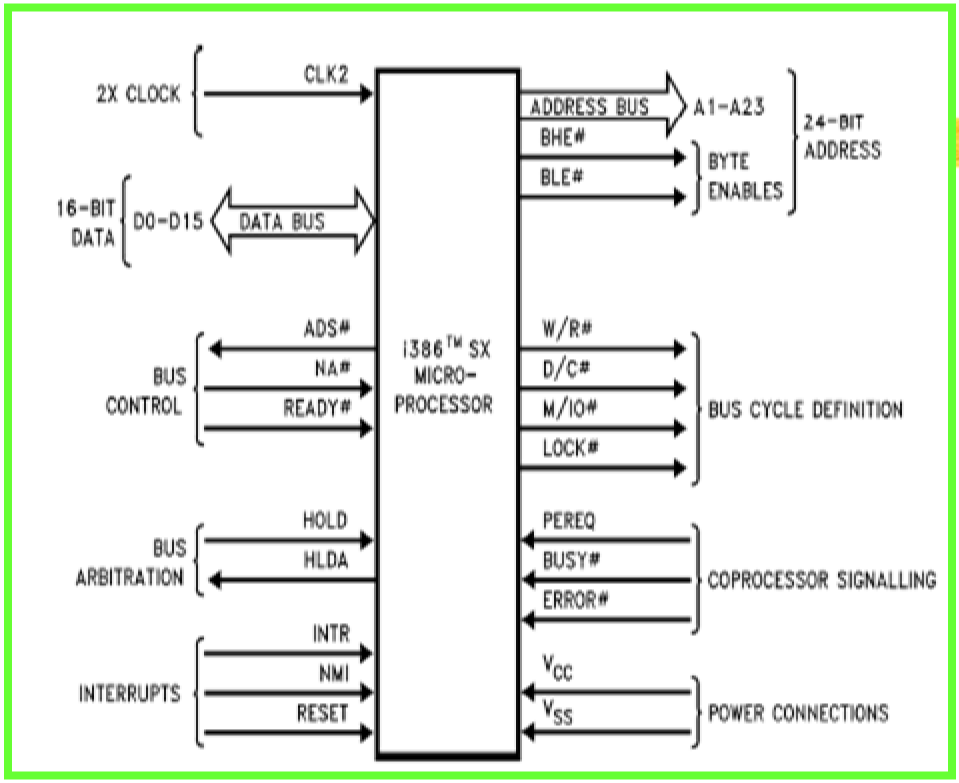
Address		Data		Control				
A_1	18	D_0	1	ADS#	16			
A_2	51	D_1	100	BHE#	19			
A_3	52	D_2	99	BLE#	17			
A_4	53	D_3	96	BUSY#	34			
A_5	54	D_4	95	CLK2	15			
A_6	55	D_5	94	D/C#	24			
A_7	56	D_6	93	ERROR#	36			
A_8	58	D_7	92	FLT#	28			
A_9	59	D_8	90	HLDA	3			
A_{10}	60	D_9	89	HOLD	4	47	69	49
A_{11}	61	D_{10}	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							

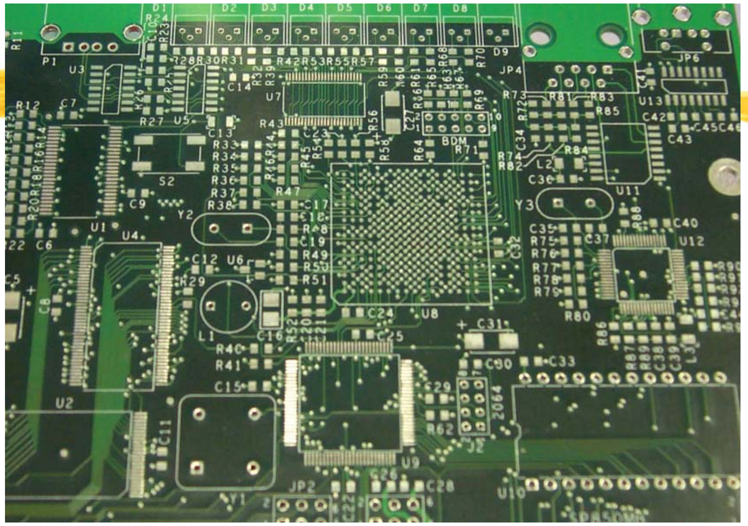
386 Micro-Architecture

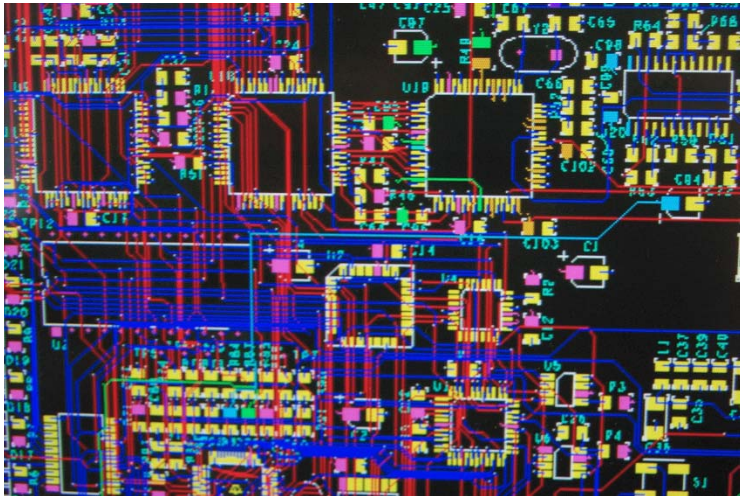


Intel386™ SX Pipelined 32-Bit Microarchitecture

Connecting with Memory, I/O, and Peripherals

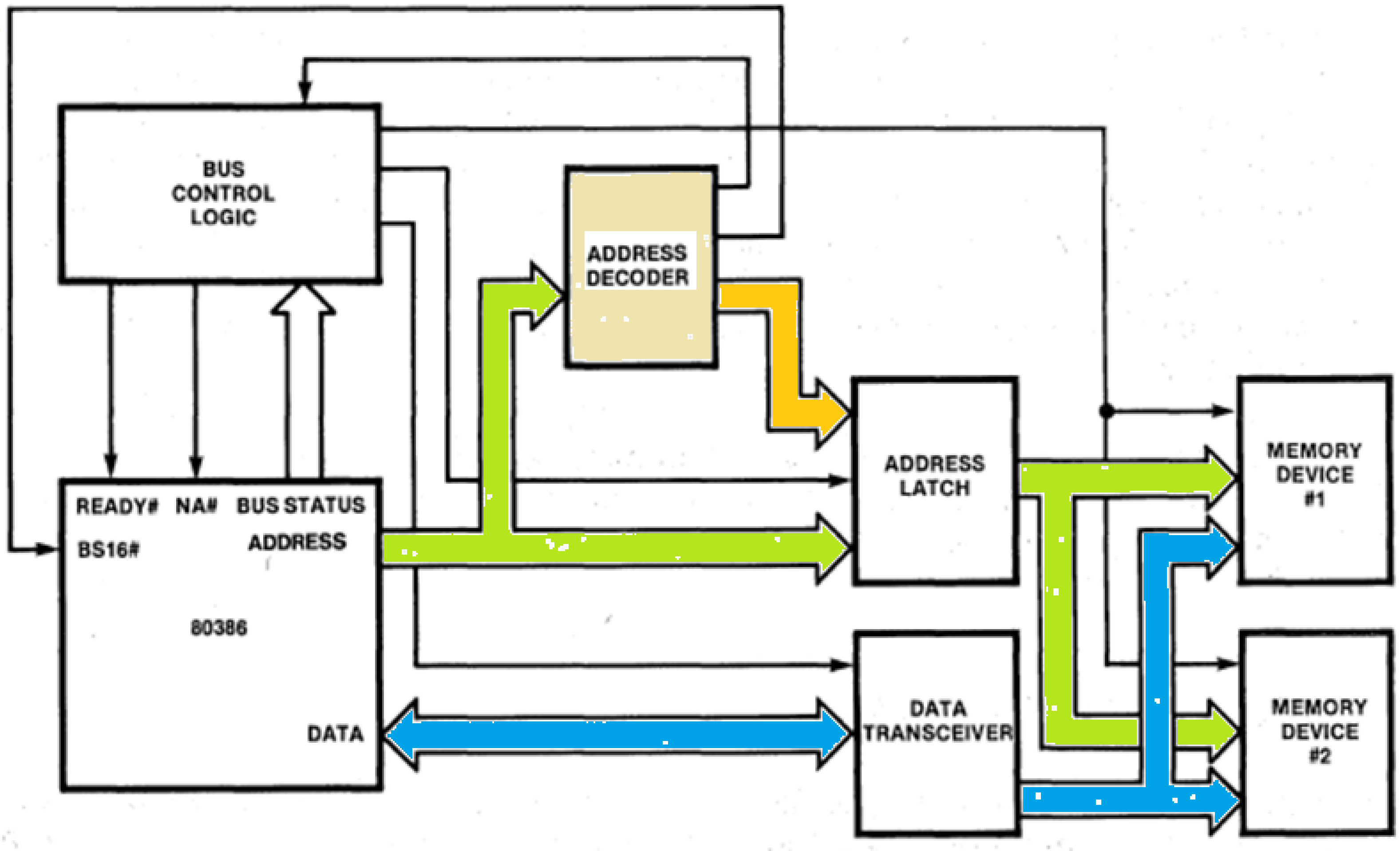






- ⌘ Single Board Computers
- ⌘ Processor Boards
- ⌘ Kits

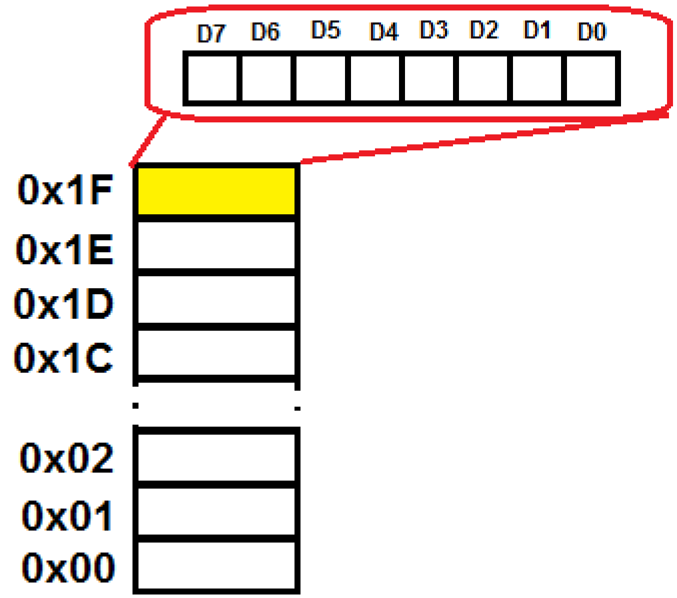
Memory Interface



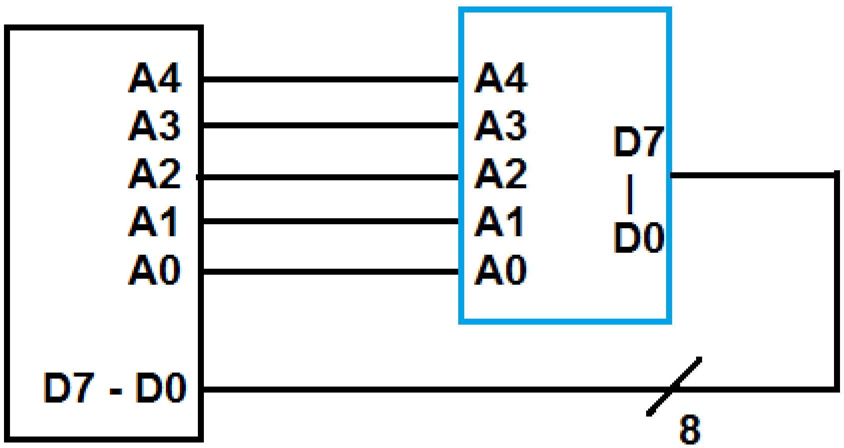
Memory Interface

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

8-bit access



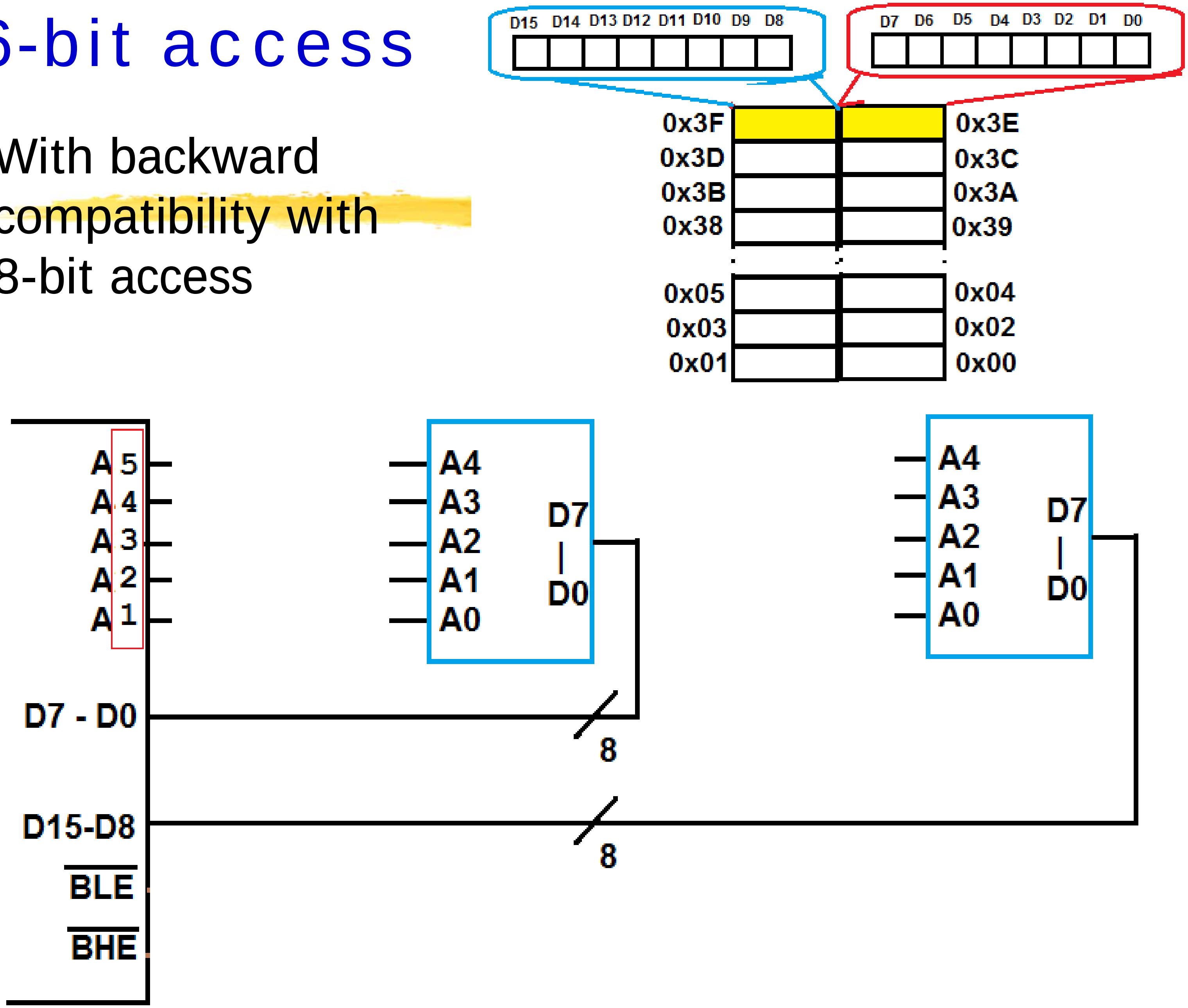
D7
D6
D5
D4
D3
D2
D1
D0
0x1F
0x1E
0x1D
0x1C
0x02
0x01
0x00



A4
A3
A2
A1
A0
D7 - D0
A4
A3
A2
A1
A0
D7
|
D0
8

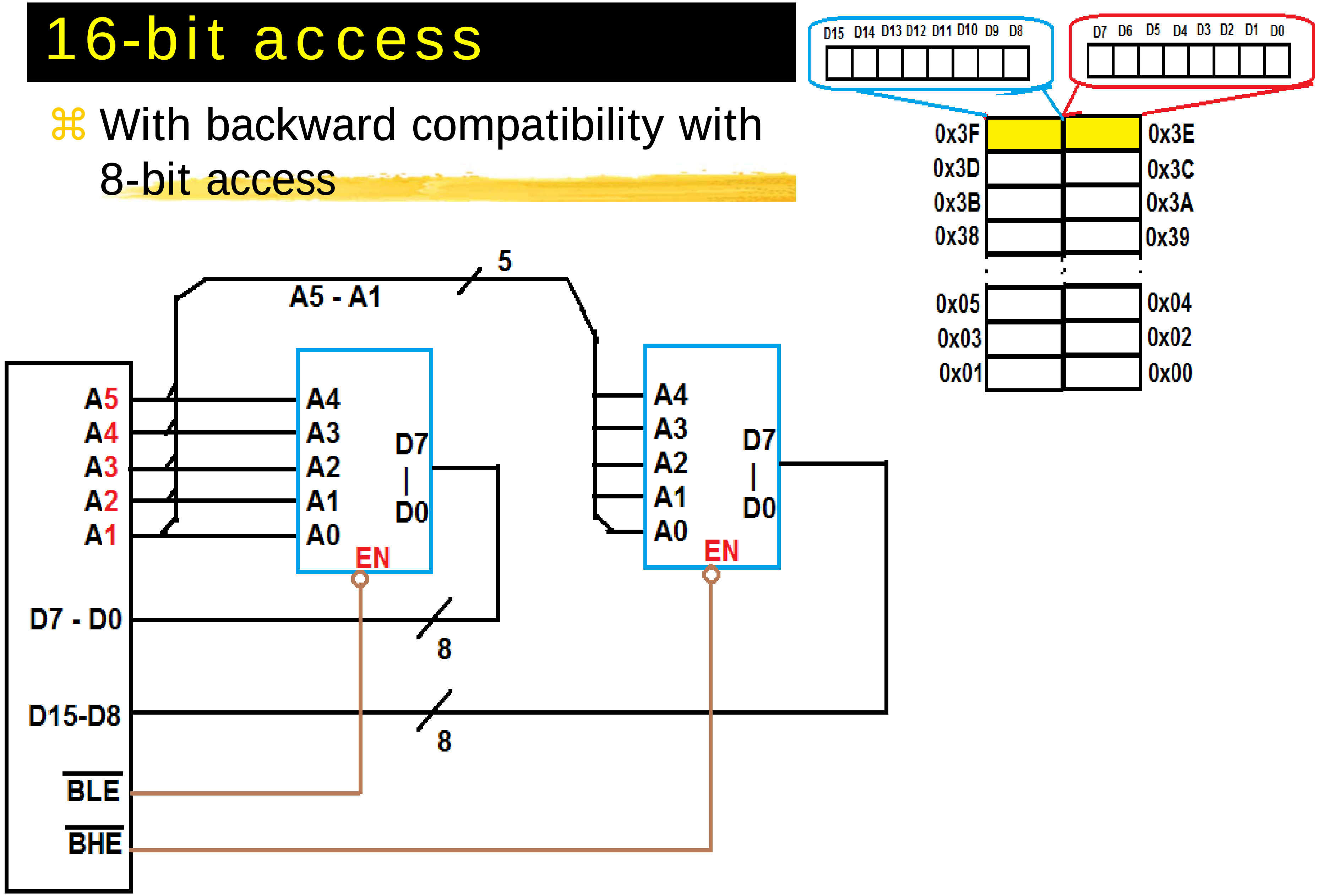
16-bit access

- With backward compatibility with 8-bit access

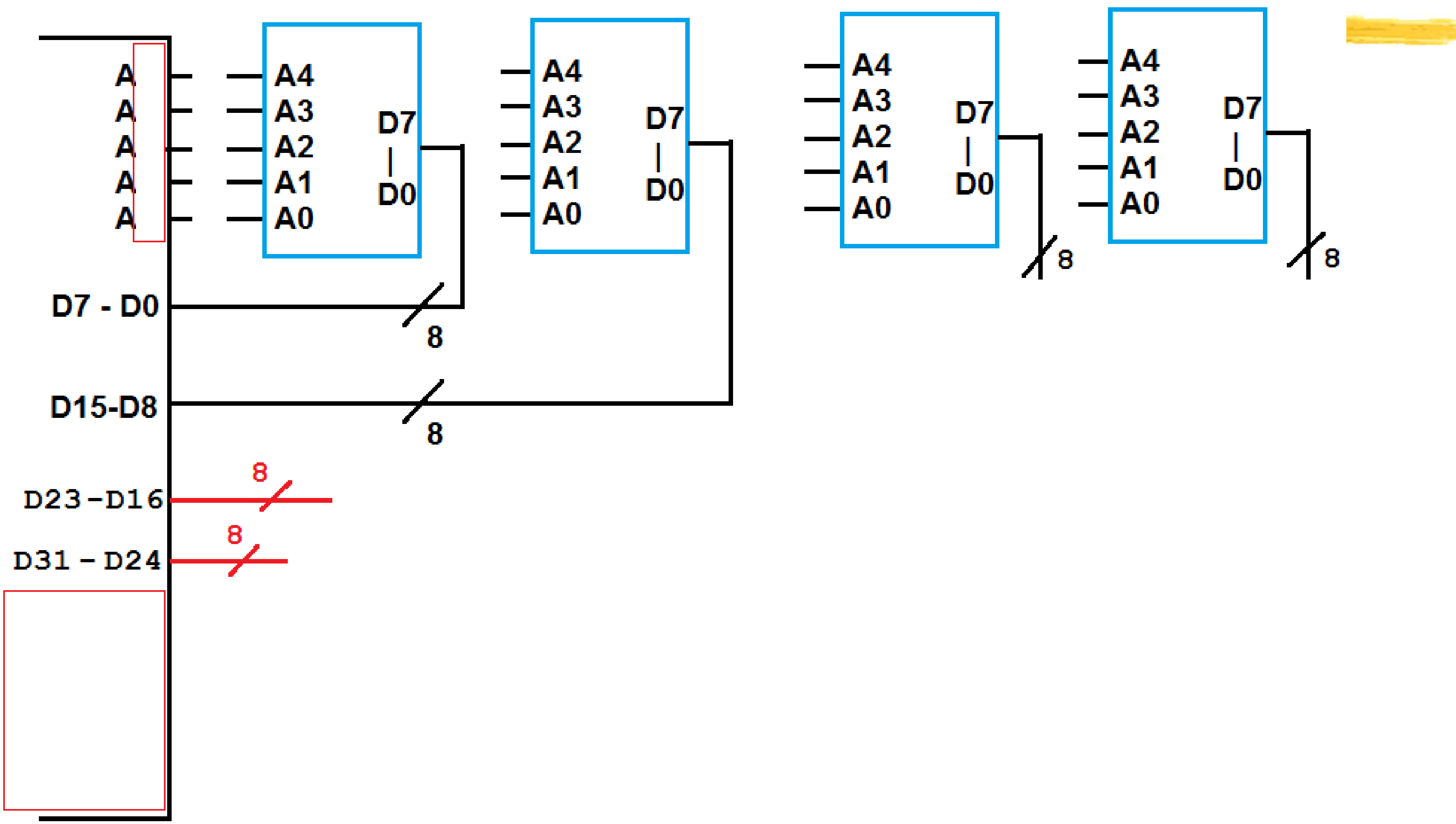


16-bit access

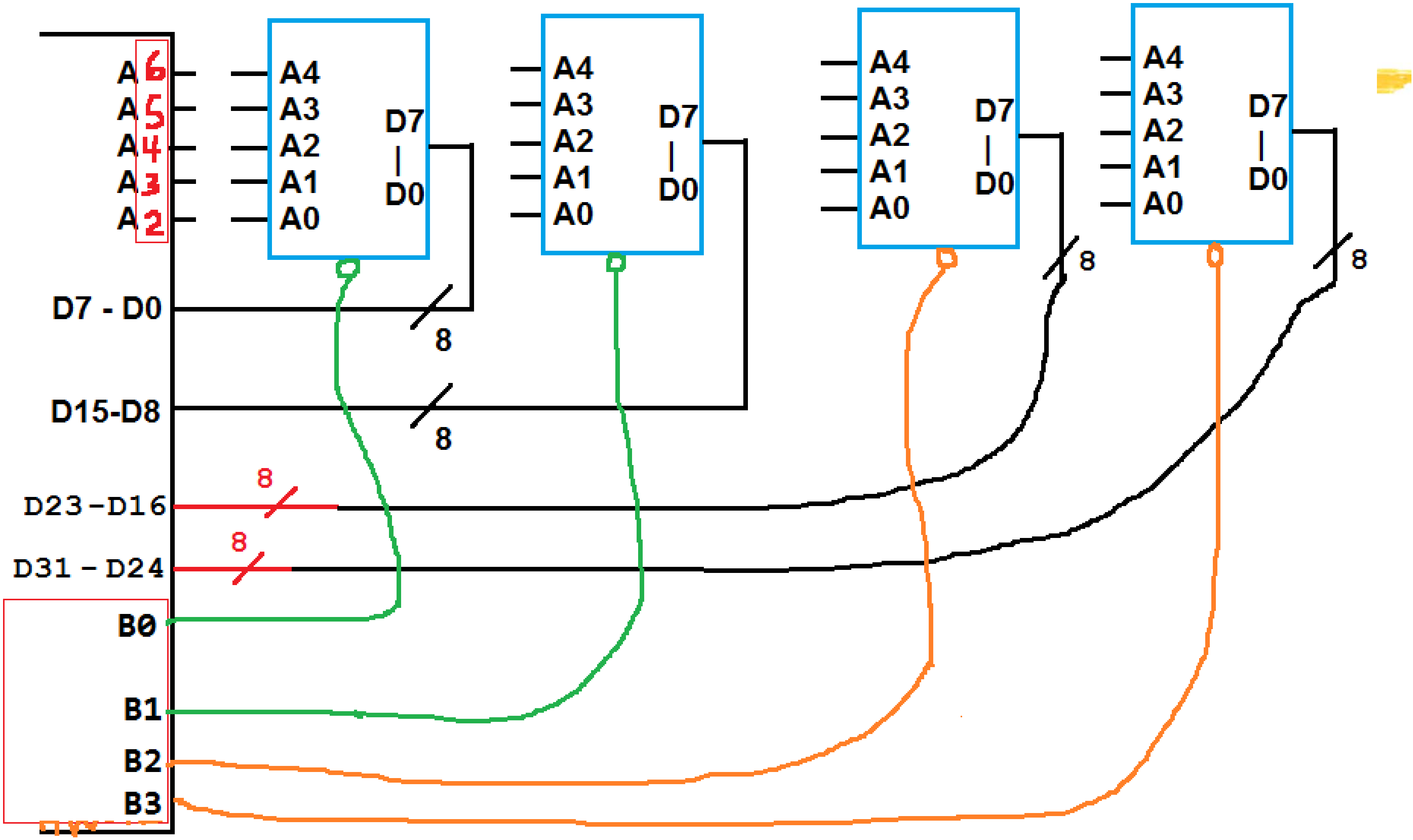
- With backward compatibility with 8-bit access



32-bit access with backward compatibility?

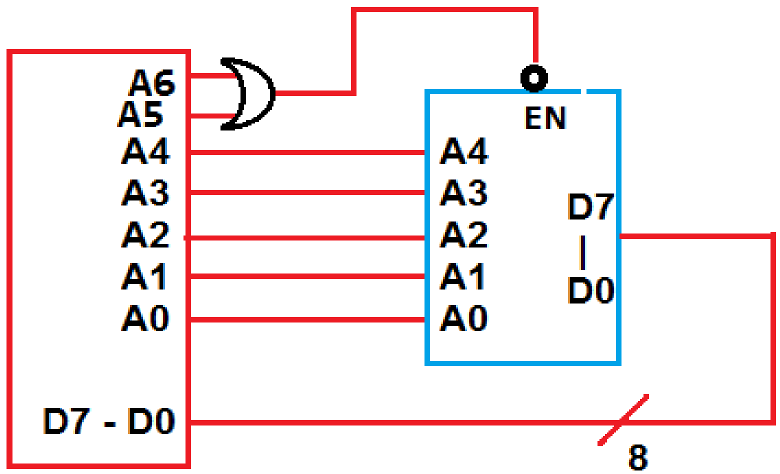


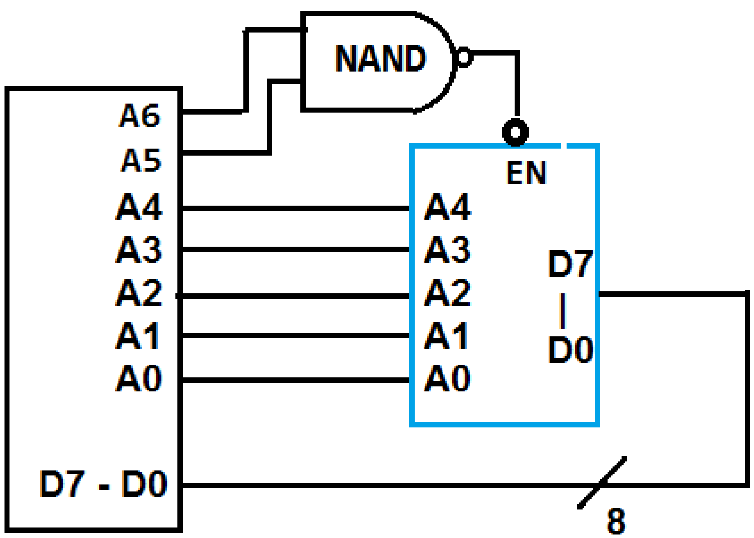
32-bit access with backward compatibility

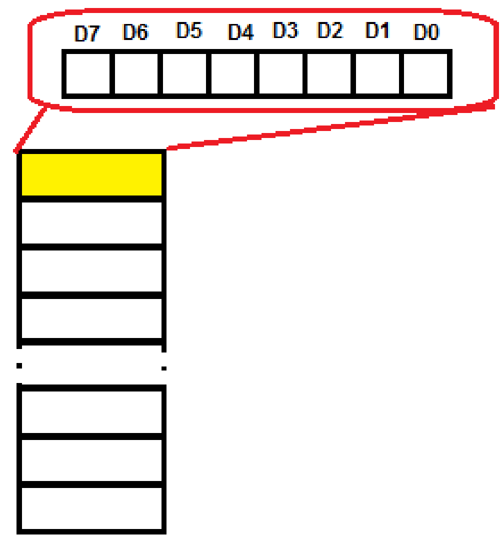


Memory Address 8

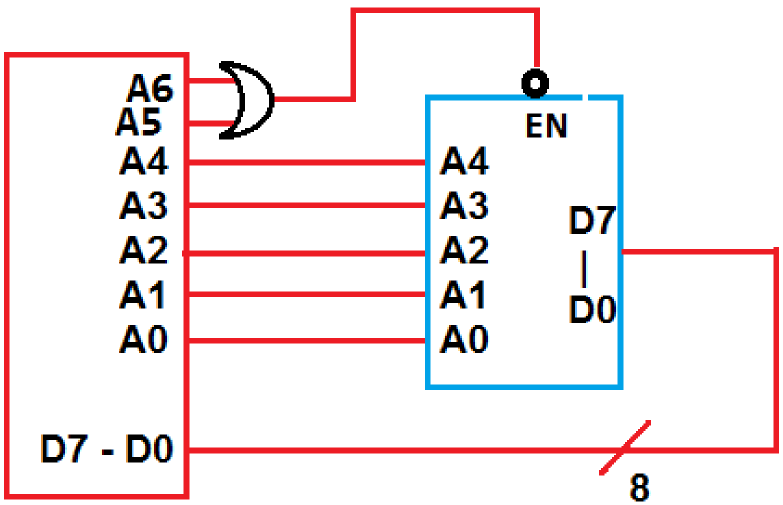
⌘Address location (first and last addresses)?

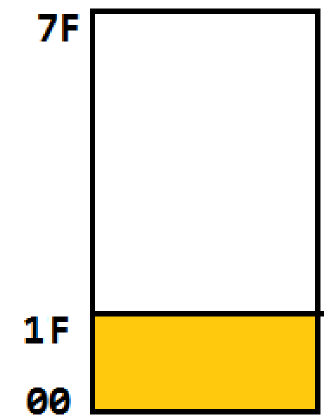


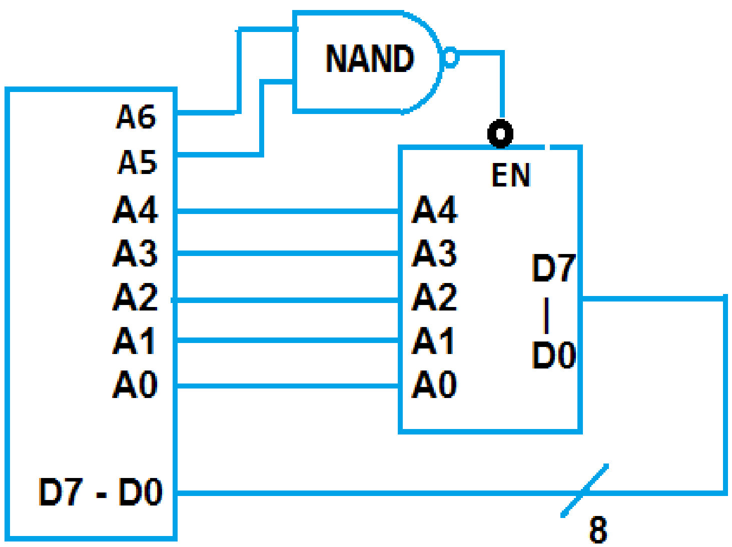


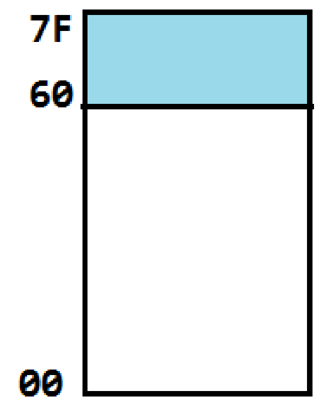


Memory Address 8



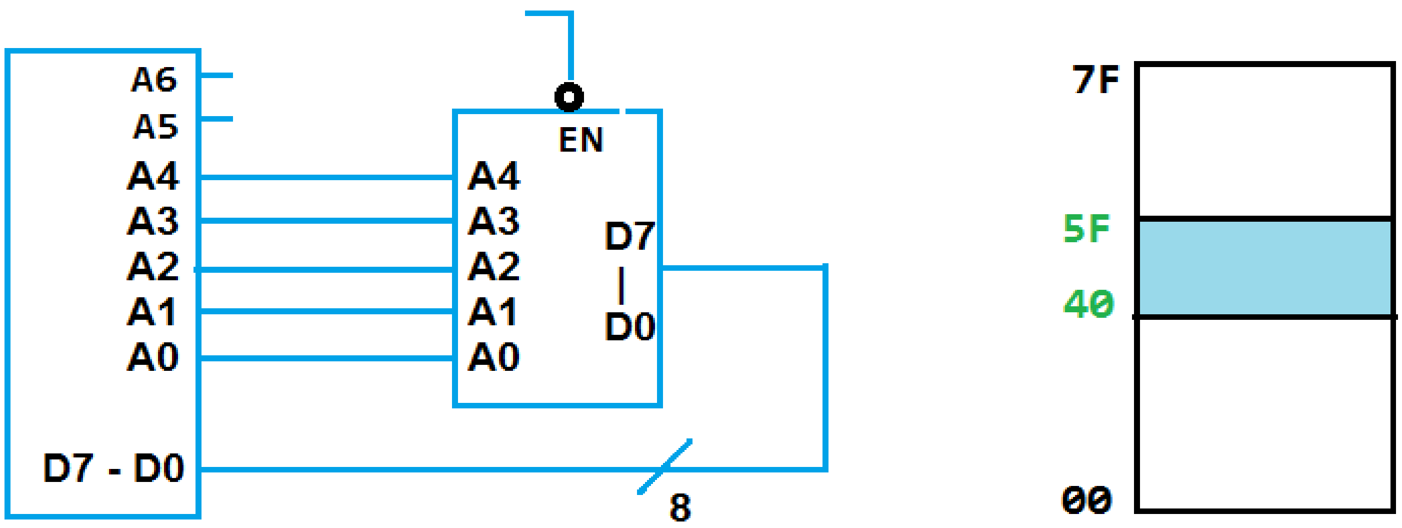






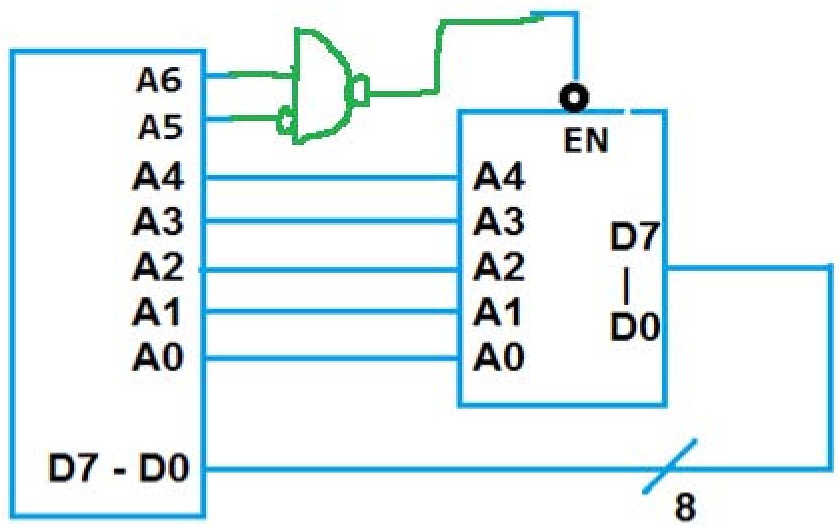
Memory Address 8 ---?

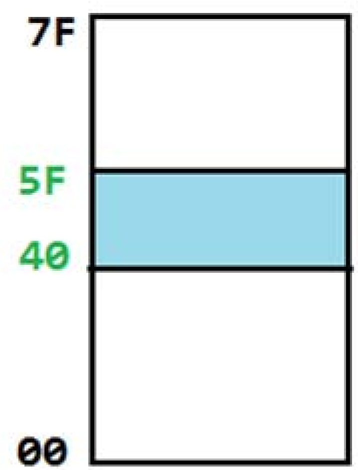
- Design a Memory Decoder so that the 32Byte memory locates as depicted.



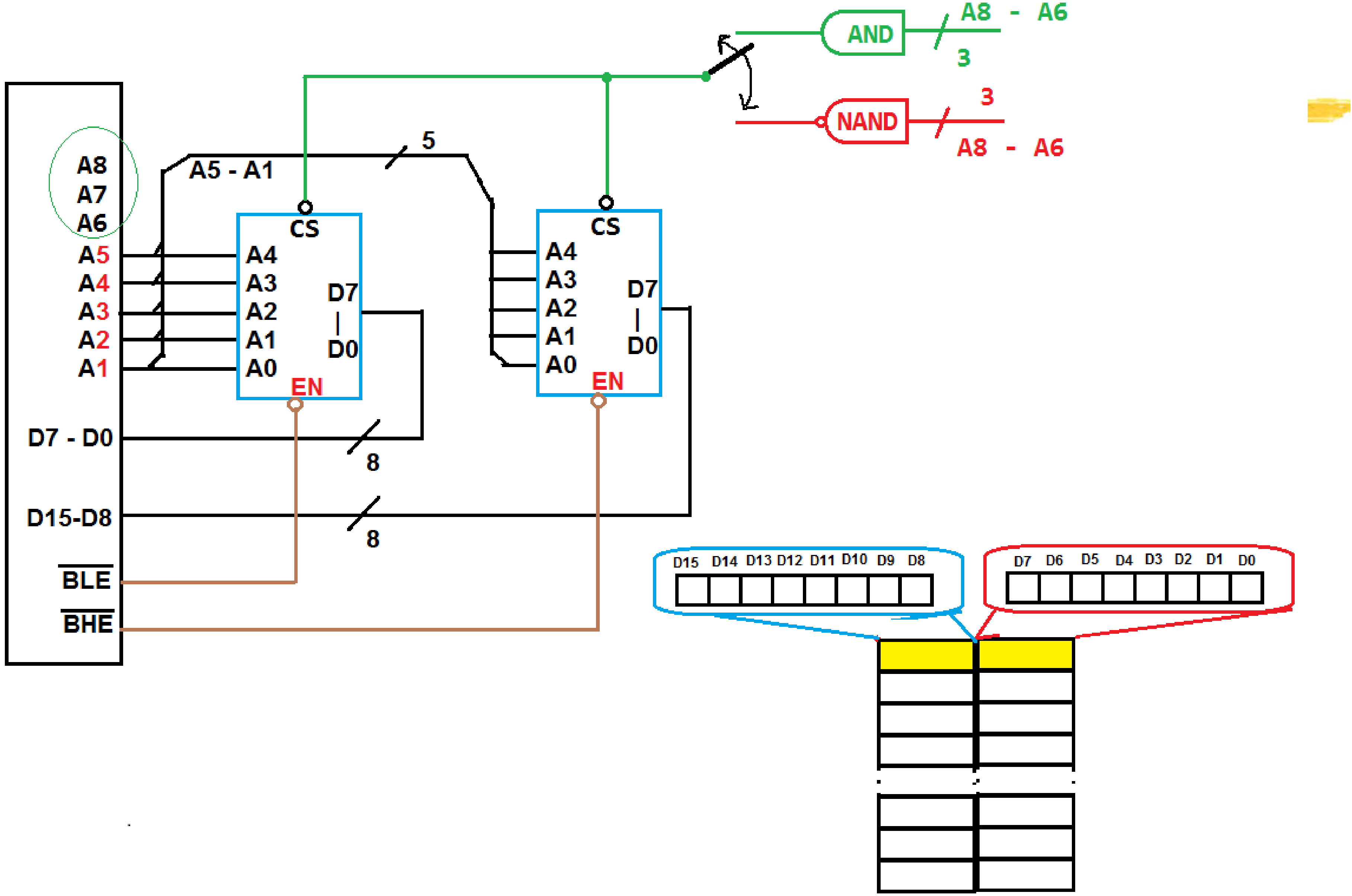
Memory Address 8 ---?

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

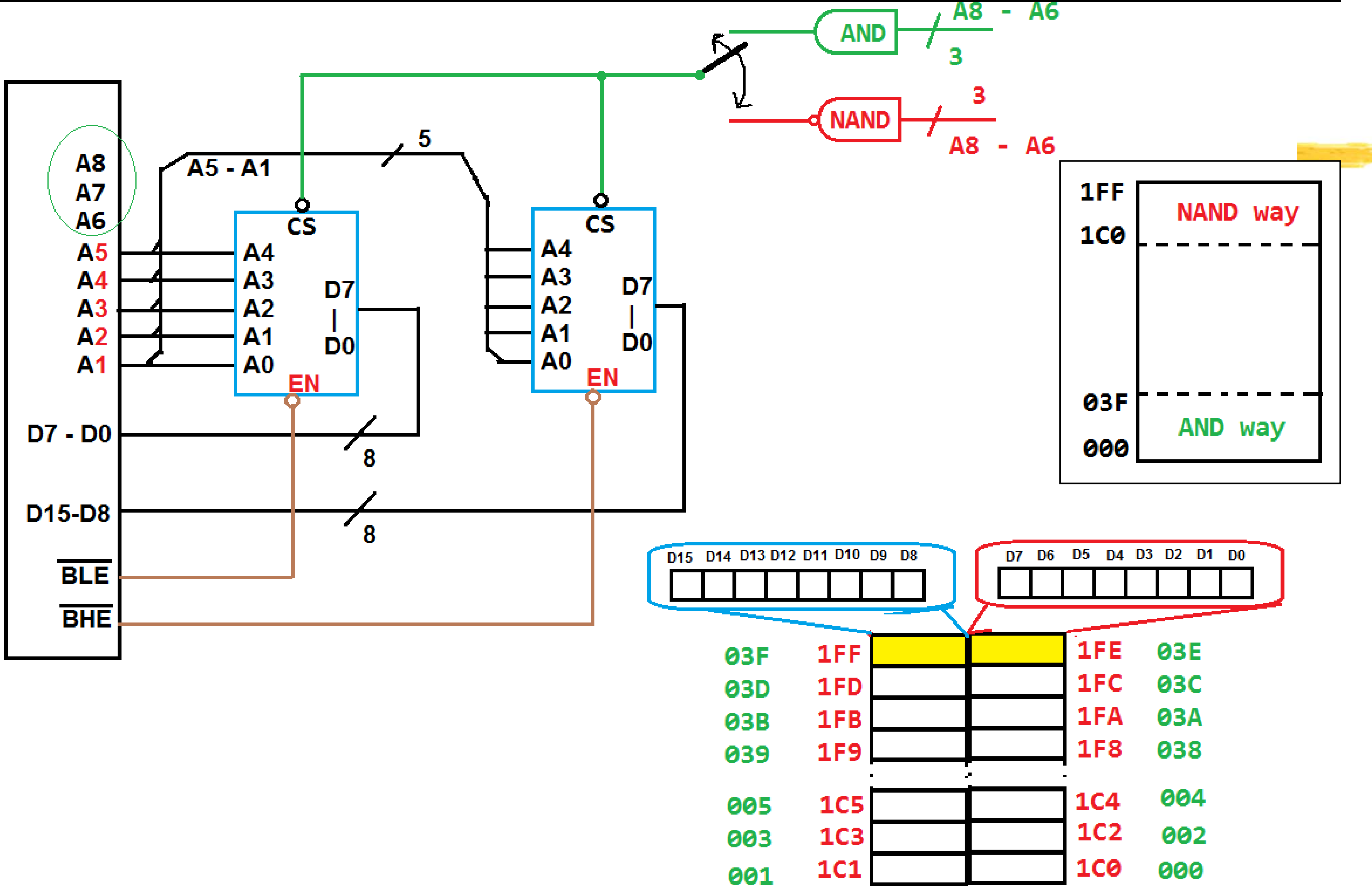




Memory Address 16 – Address Space

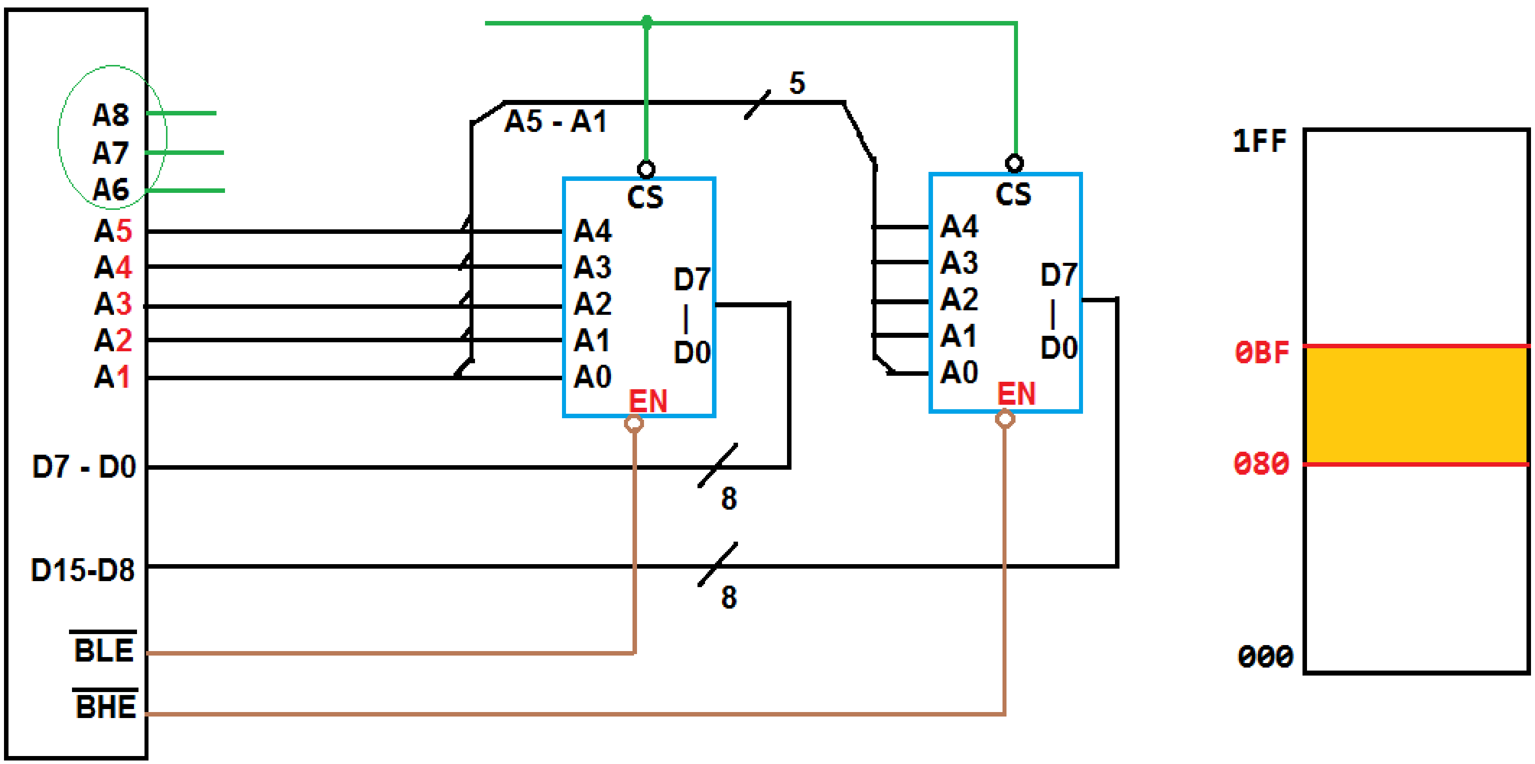


Memory Address 16



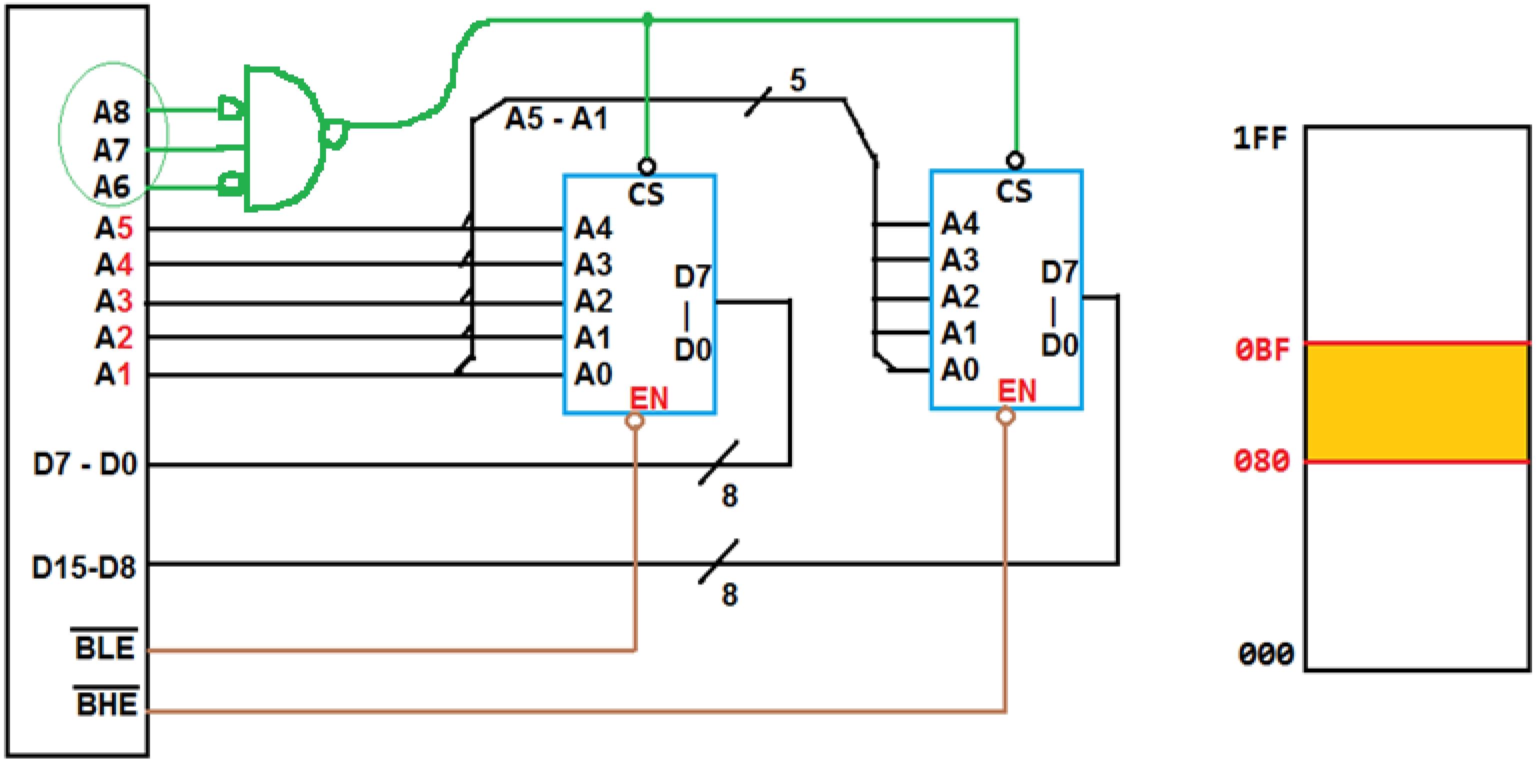
Memory Address 16 -- ?

⌘ Design a memory decoder for the depicted address segment.



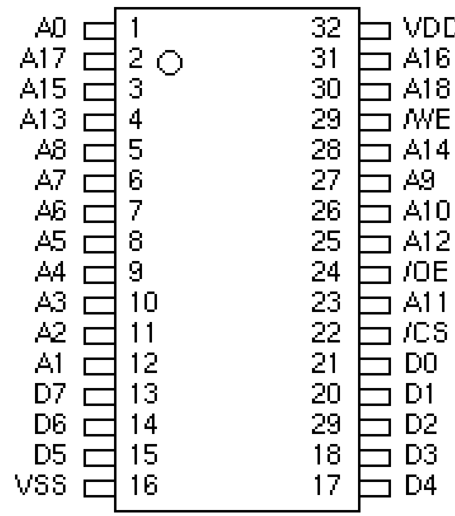
Memory Address 16 -- ?

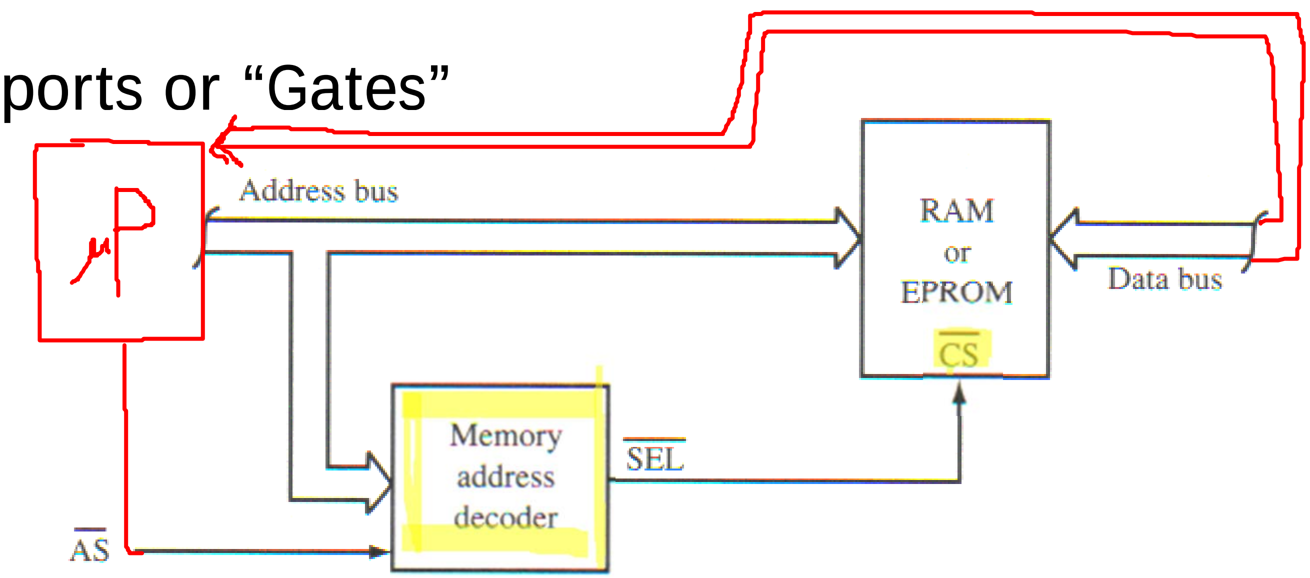
- Design a memory decoder for the depicted address segment.



Memory Address Decoding

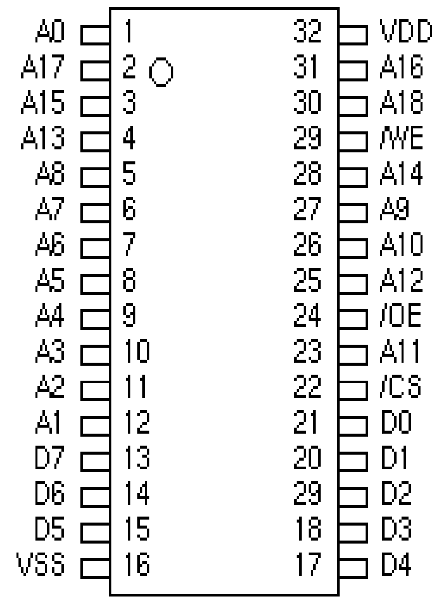
- How Much Memory?
  - How Many Address Lines?
    - 1K → 10 lines
    - 32K → 15 lines
    - 23 ADDR lines → 8MB?
- MEMORY PINOUTS
  - Address Lines
  - Data Lines
  - “Defense” ports or “Gates”
    - /CS
    - /CE
    - /OE
    - /WE

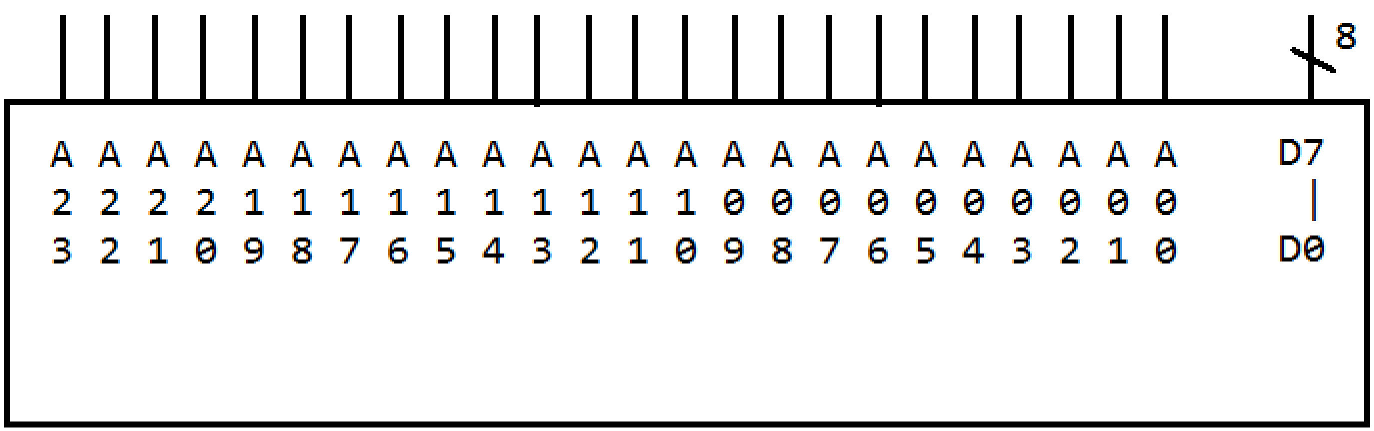




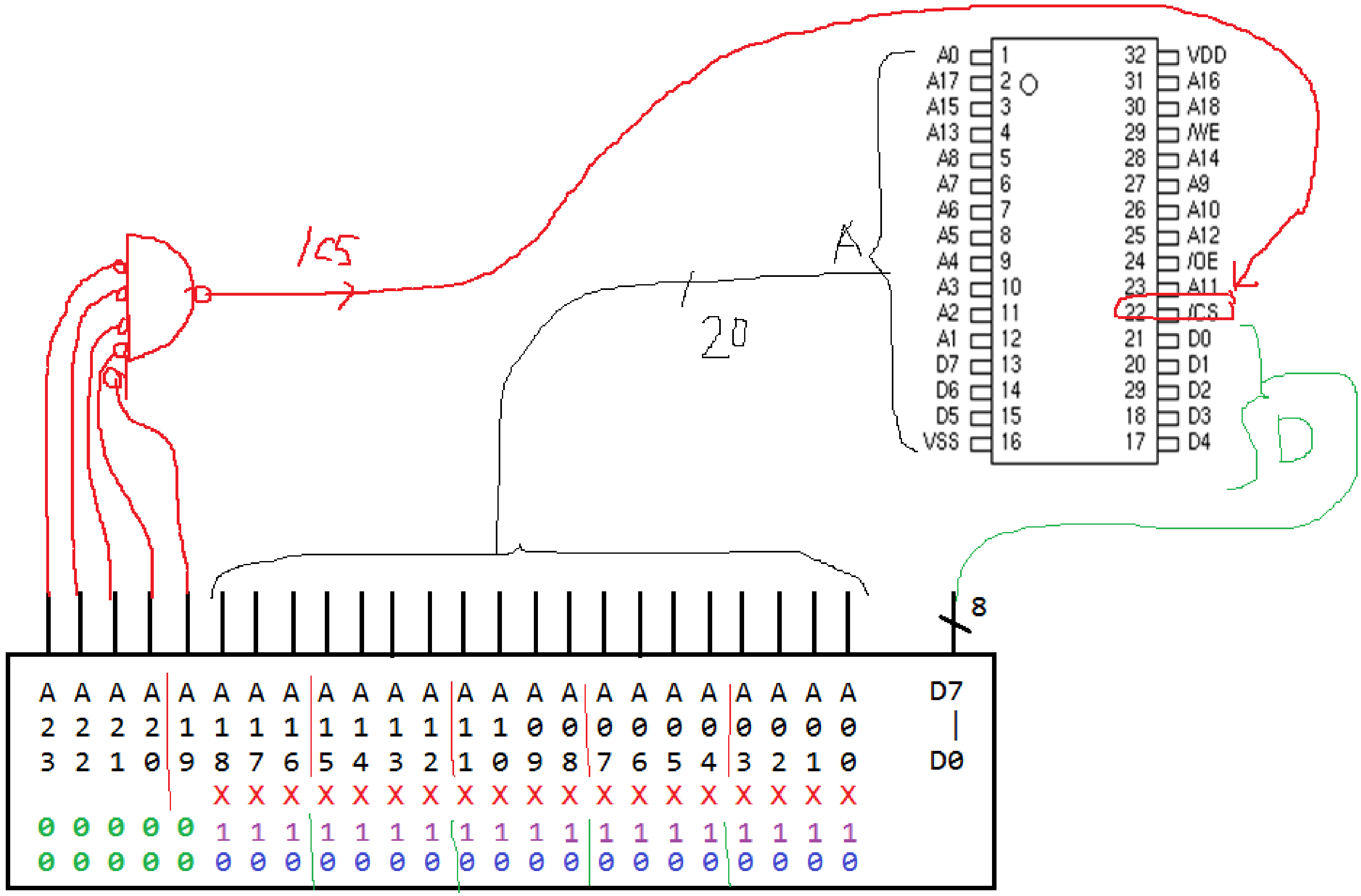
8-bit uP + MEM Exercise

- uP has 2^{24}=24MB=16MB memory space: 000000 - FFFFFF
- MEM has 2^{19}=0.5MB
- Let's place the MEM between $00000 - $7FFFF
- Up Addr ←→MEM Addr: A18 – A0
- The left-over Addr lines in the uP: A23 – A19
  - This condition is used to open the MEM gate (namely, /CS)



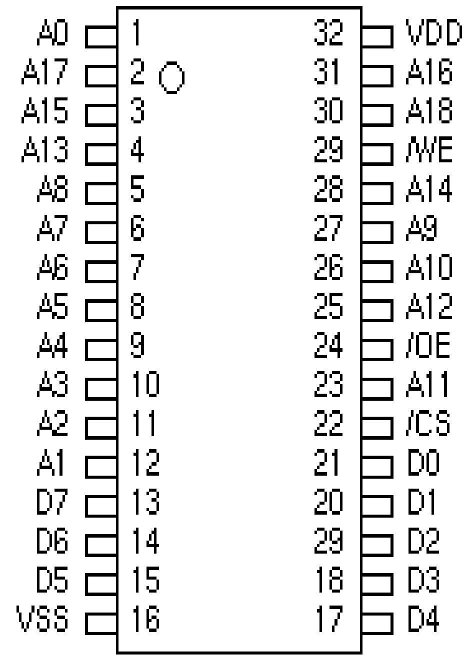


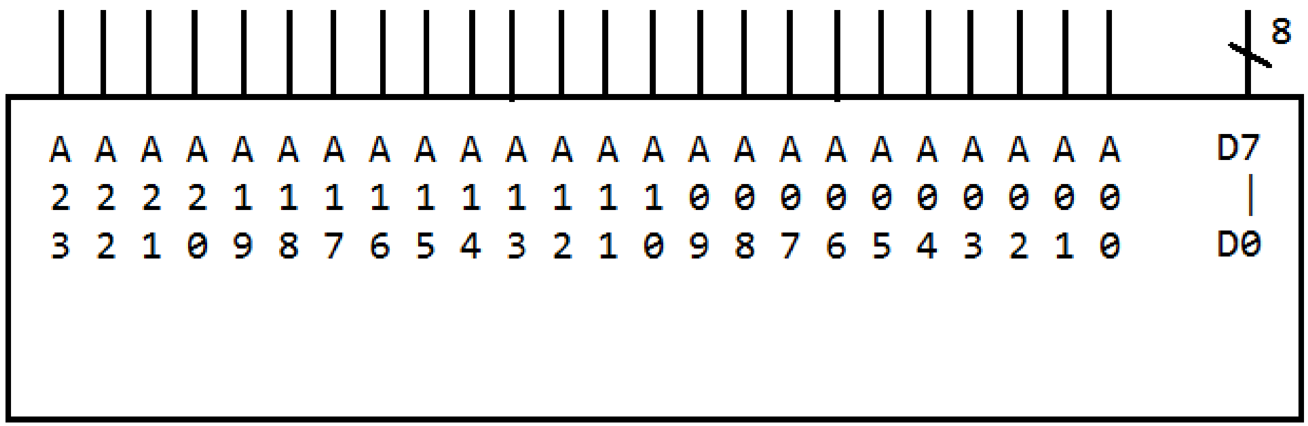
8-bit uP + MEM Exercise



8-bit uP + MEM Exercise

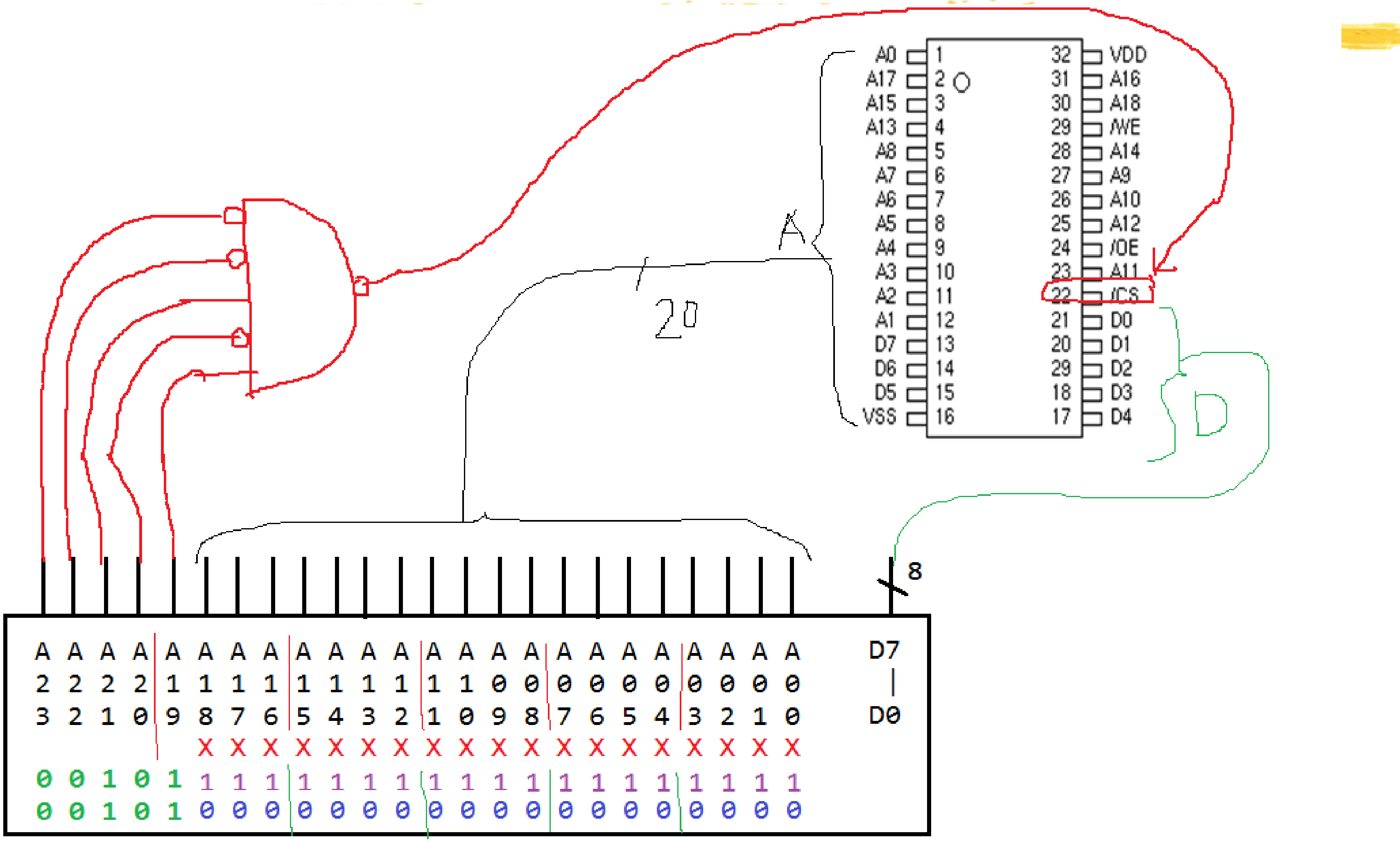
- Now, place 0.5MB size MEM between $280000 - $2FFFFF





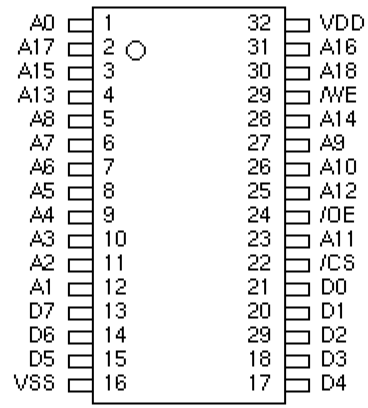
8-bit uP + MEM Exercise

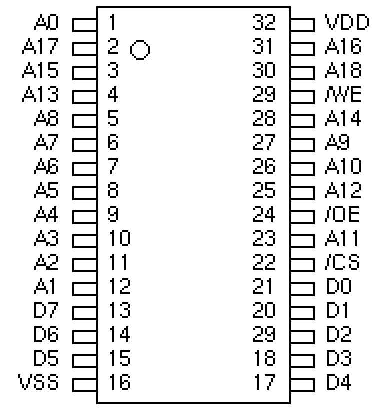
- Now, place 0.5MB size MEM between $280000 - $2FFFFF

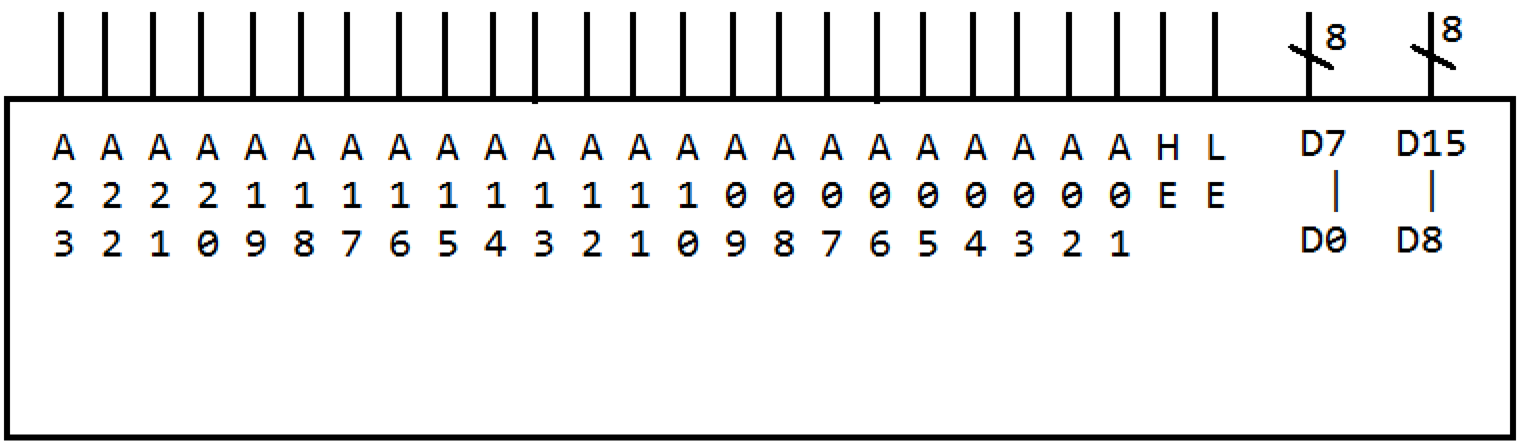


16-bit uP + MEM

- uP does not have A0
  - BHE (UDS) and BLE (LDS), instead.
- uP Addr ←→ MEM Addr
  - A19 – A1 (uP): A18 – A0 (MEM)
  - Left-Over Addr (uP): A23- A20
  - BHE and BLE controls which MEM (or ADDRESS LOCATION) to access
    - BHE LOW: Upper MEM (upper or odd address location)
    - BLE LOW: Lower MEM (lower or even address location)
    - Both BHE amd BLE low: both address locations

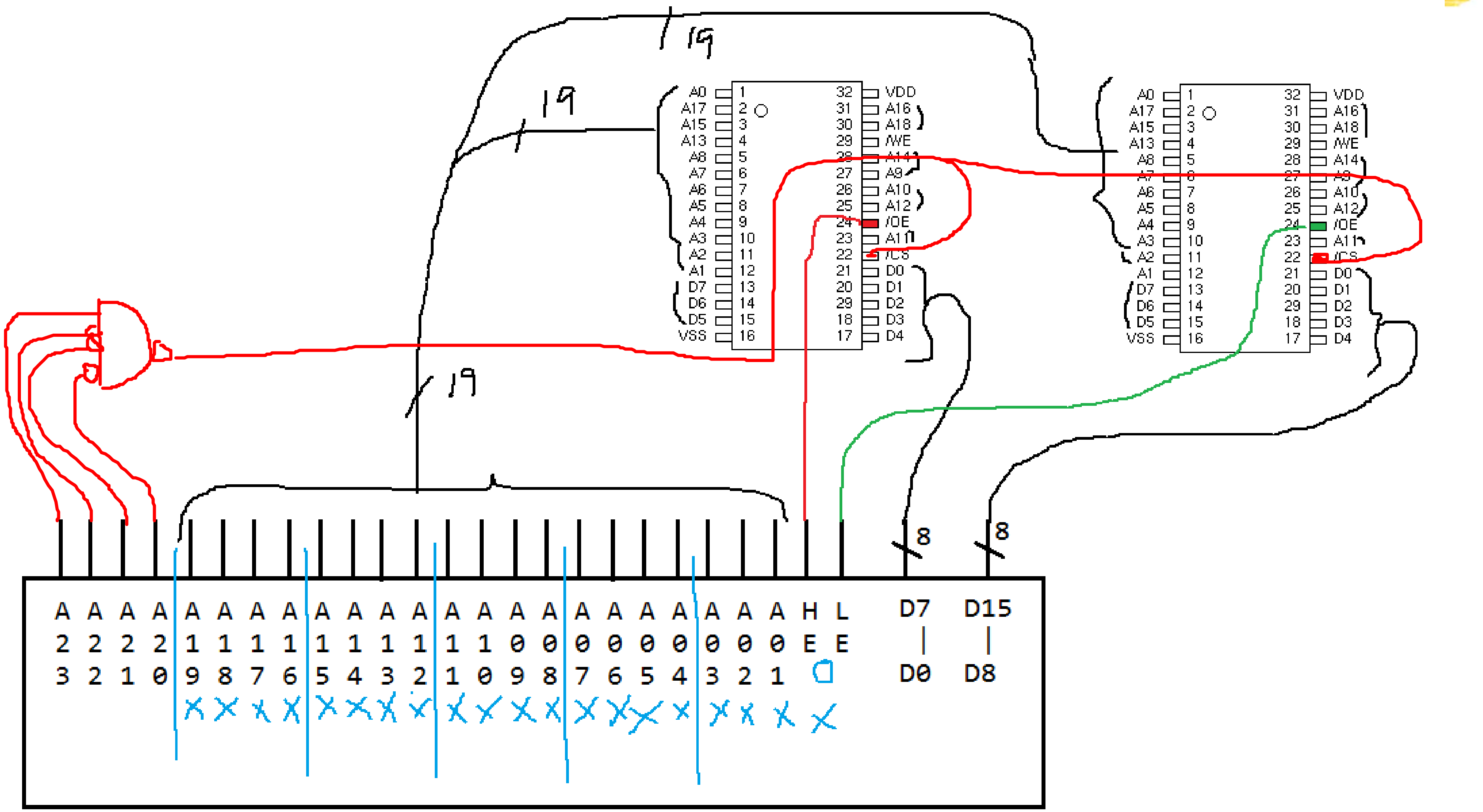






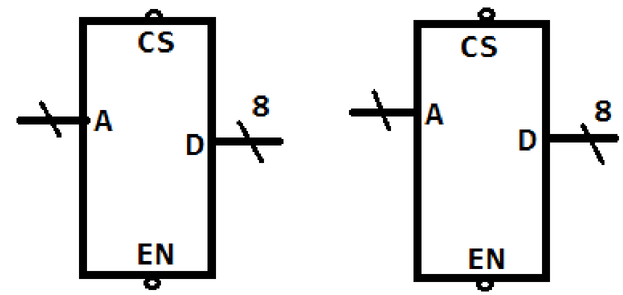
16-bit uP + MEM

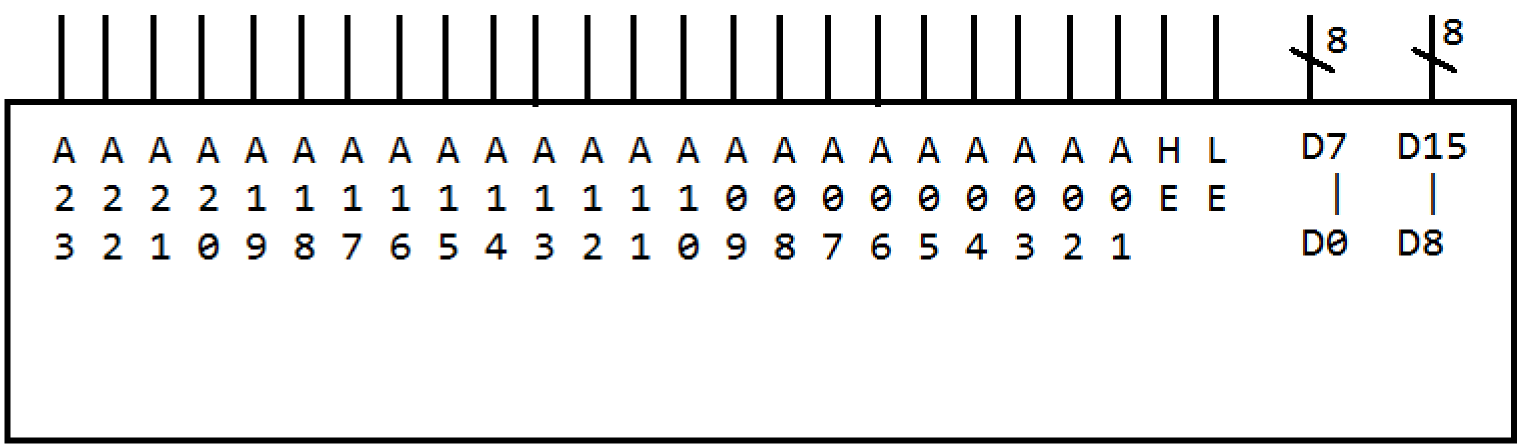
- uP does not have A0
  - BHE (UDS) and BLE (LDS), instead.



Memory Decoding

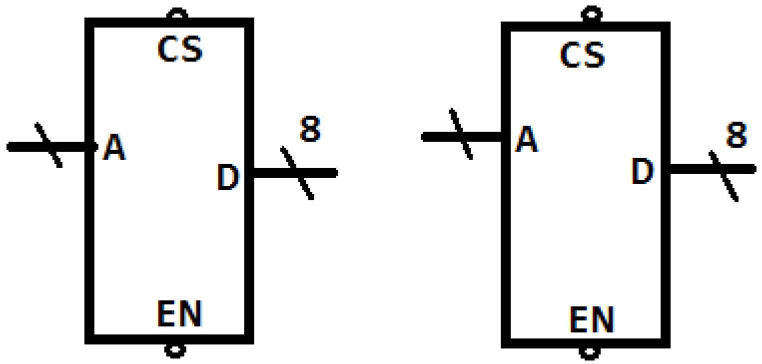
- Q: 64K Word (or 128 KB) of RAM, with it's starting address at $480000
- A: 64KB → 16 lines each MEM
  - Range: $480000 - $49FFFF
  - BHE and BLE for A_0 line → Enable
  - Upper address lines → CS for both MEM

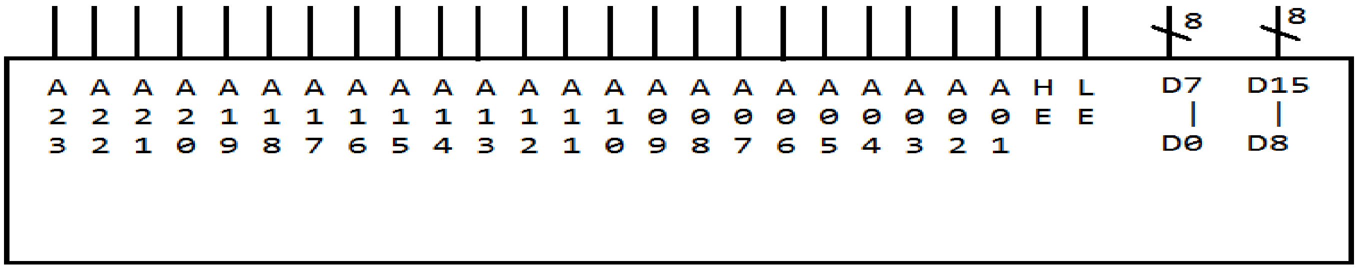




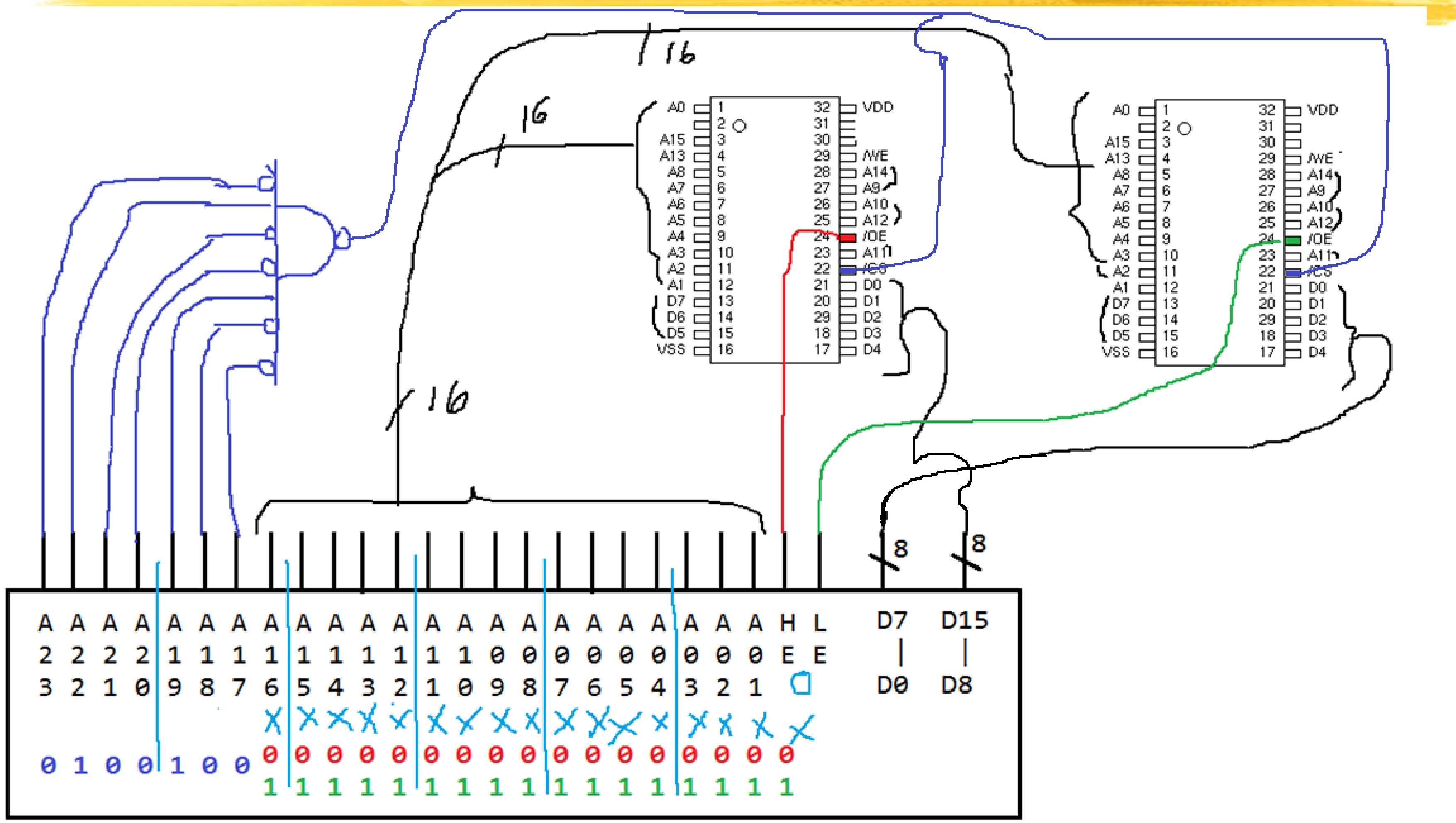
Memory Decoding -------- Name____________

- Q: 64K Word (or 128 KB) of RAM
- A: 64KB → 16 lines each MEM
  - Range: $480000 - $49FFFF



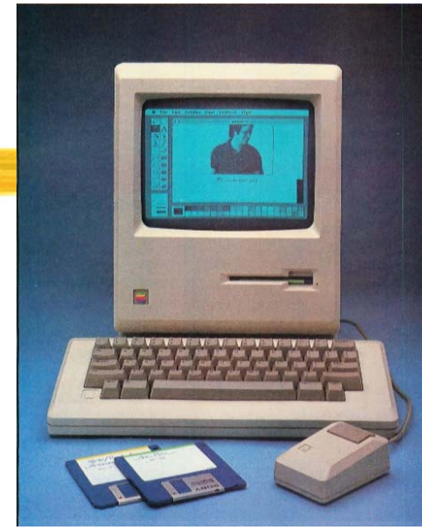


- Q: 64K Word (or 128 KB) of RAM, with it's starting address at $480000
-



Apple Macintosh Classic

- 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



At a Glance

Name
Macintosh

Manufacturer
Apple Computer
20525 Mariani Ave.
Cupertino, CA 95014
(408) 996-1010

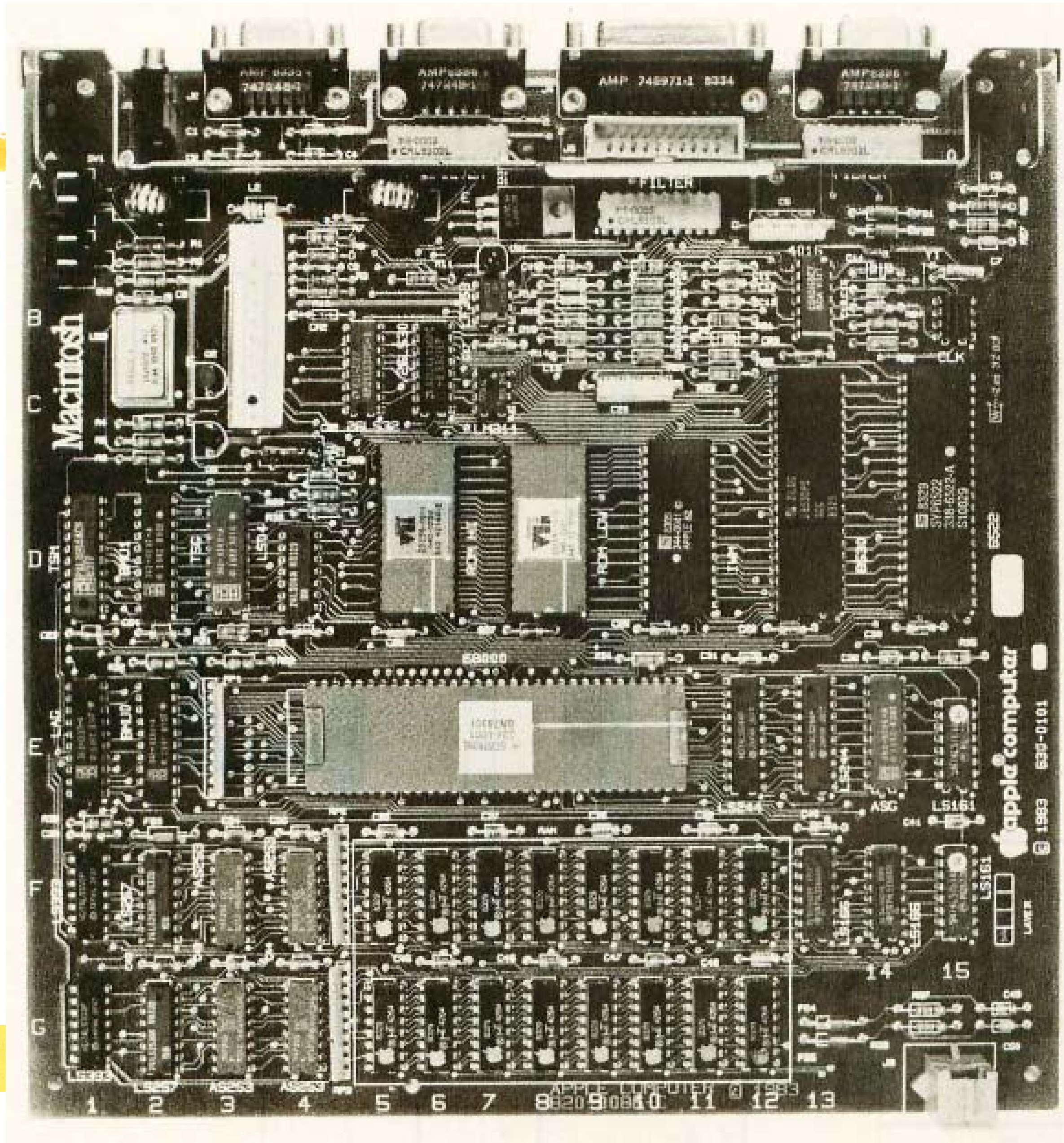
Dimensions
9.75 by 9.75 by 13.5 inches
Weight
Main unit, keyboard and mouse—22.7 lbs.
Power Requirements
105–130 V AC, 60 Hz (U.S. model); 85–135 V AC, 50/60 Hz (international model)

Memory
128K bytes of RAM, 64K bytes of ROM
Standard Configuration
Main unit with 128K bytes of RAM, 64K bytes of ROM, integral Sony 3½-inch disk drive, 9-inch video monitor, two serial ports; external mechanical mouse; external keyboard
Mass Storage
One Sony 3½-inch disk drive; 3½-inch disk holds 400K bytes and is encased in a rigid plastic housing
Video Display
9-inch monitor, noninterlaced 60.15-Hz image, 512- by 342-pixel resolution
Pointing Device
Mechanical mouse
Keyboard
Detached keyboard; 58 keys (59 in international version); autorepeat; two-key rollover

Hardware Options
Second disk drive, keypad, Imagewriter printer, security kit (for chaining computer to table)
Software Options
Mac Paint (drawing program), Mac Write (a simple word processor), Mac BASIC, Mac Pascal, others (see text)
Prices
Standard system, $1995–$2495; Mac Paint and Mac Write (together), bundled at no charge for the first 100 days, $195 (for the two) thereafter; Macintosh Pascal, BASIC, Logo, Terminal, and Assembler/Debugger, $99 each; Mac Draw and Mac Project, $125 each; keypad, $99; second disk drive, $395; Imagewriter printer, $495

Apple Macintosh Classic

- CPU: 8MHz Motorola 68000
- Memory: 128KB (512KB in later version) RAM, 64KB ROM



The product-design goals of small size, light weight, and moderate end-user cost encouraged us to create a low-power, low component-count design.

Macintosh System Architecture
by Burrell C. Smith

Apple Macintosh Circuit Diagram

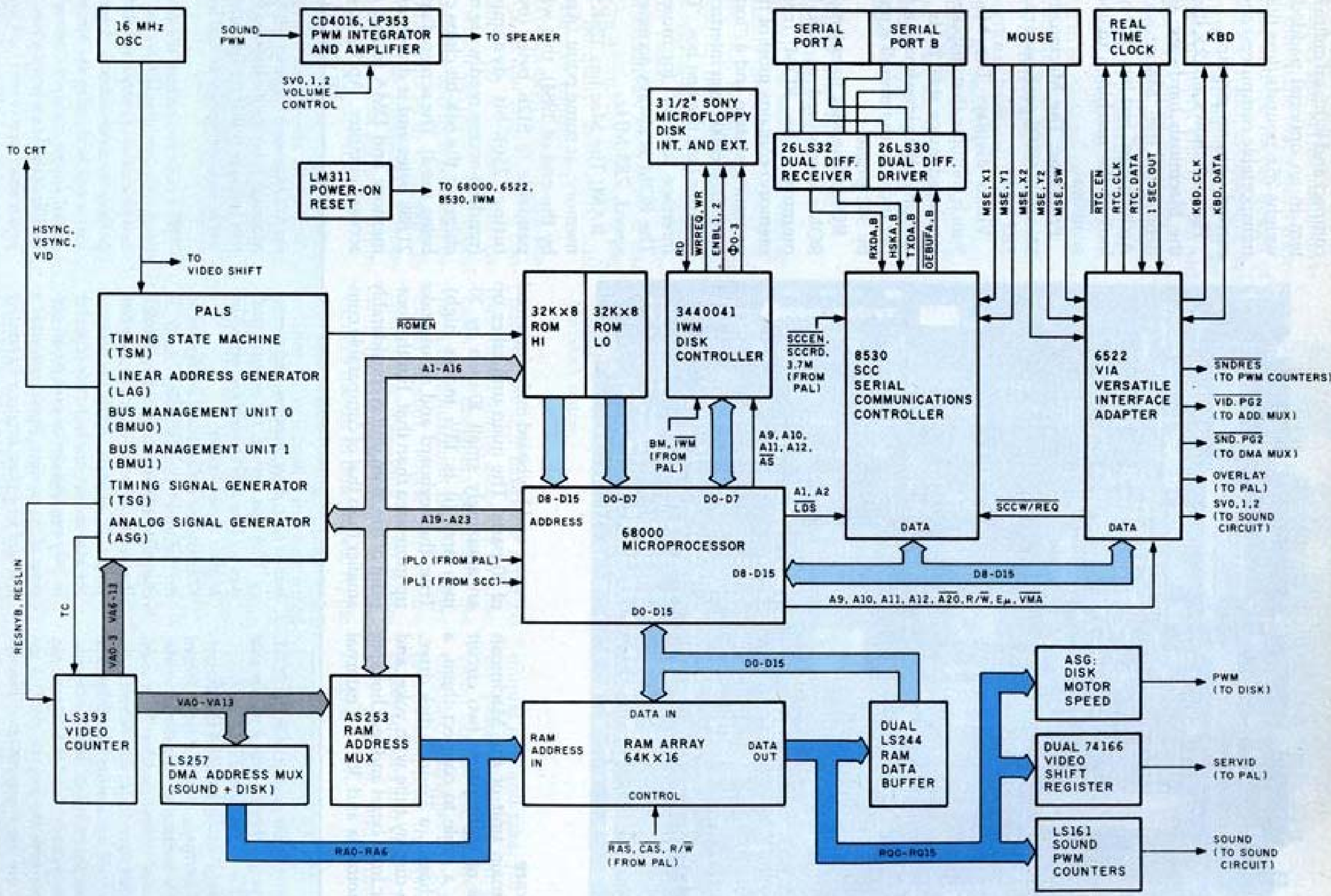


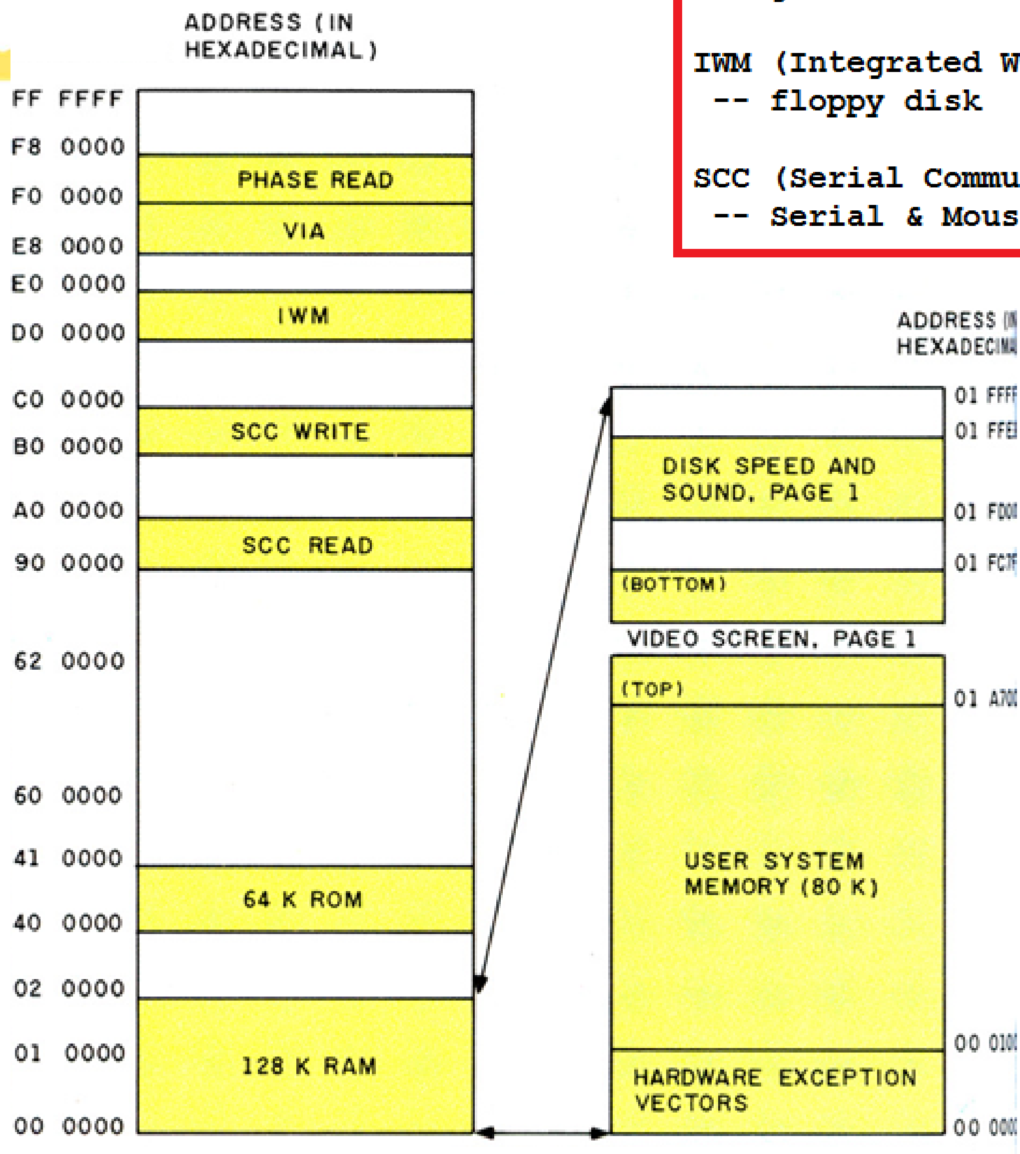
Figure 2: *A block diagram of the Macintosh hardware. For more details, see the "Macintosh System Architecture" text box.*

Memory Map (for Apple Macintosh)

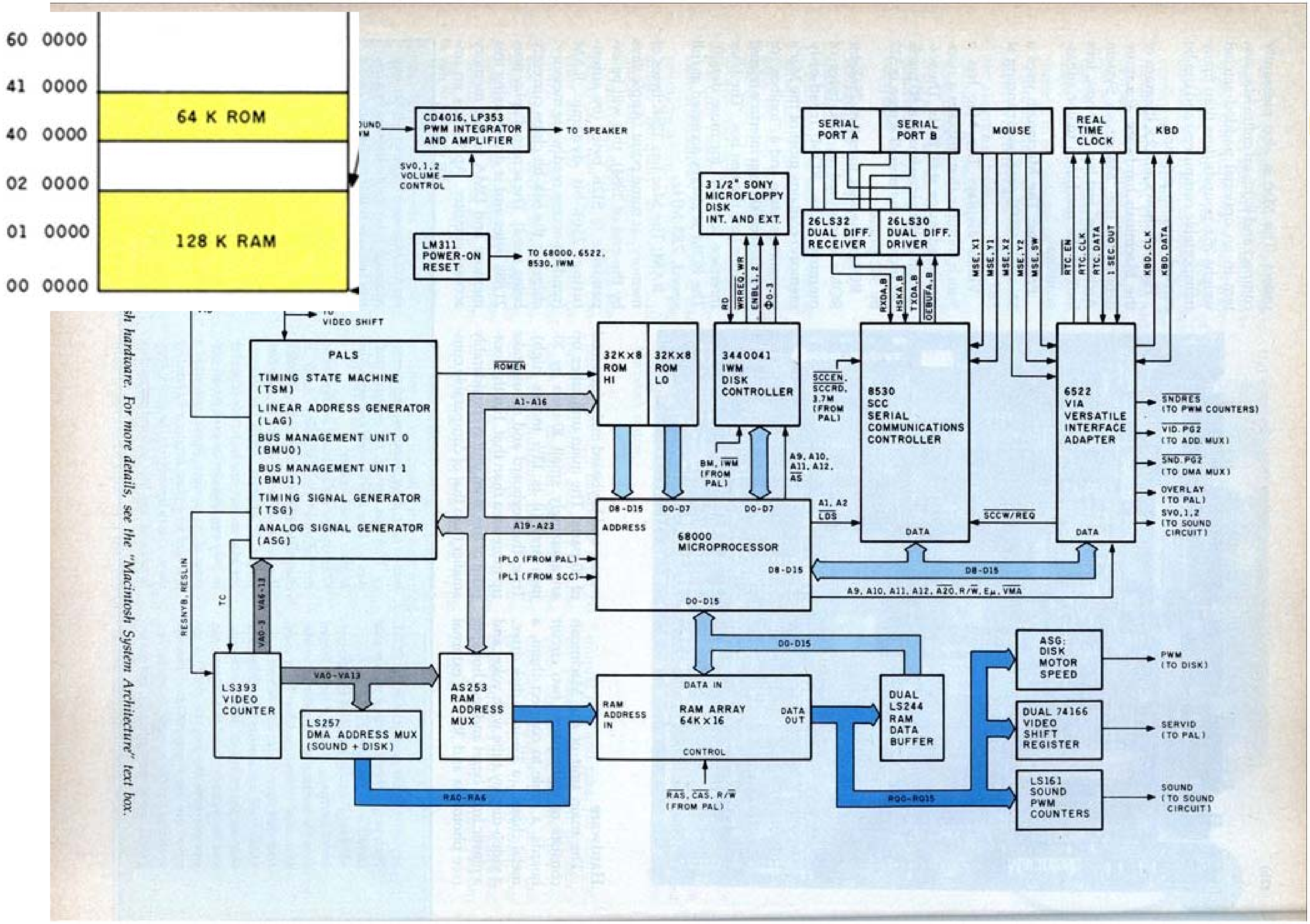
VIA (Versatile Interface Adapter)
---general I/O

IWM (Integrated Woz Machine)
-- floppy disk

SCC (Serial Communications Controller)
-- Serial & Mouse

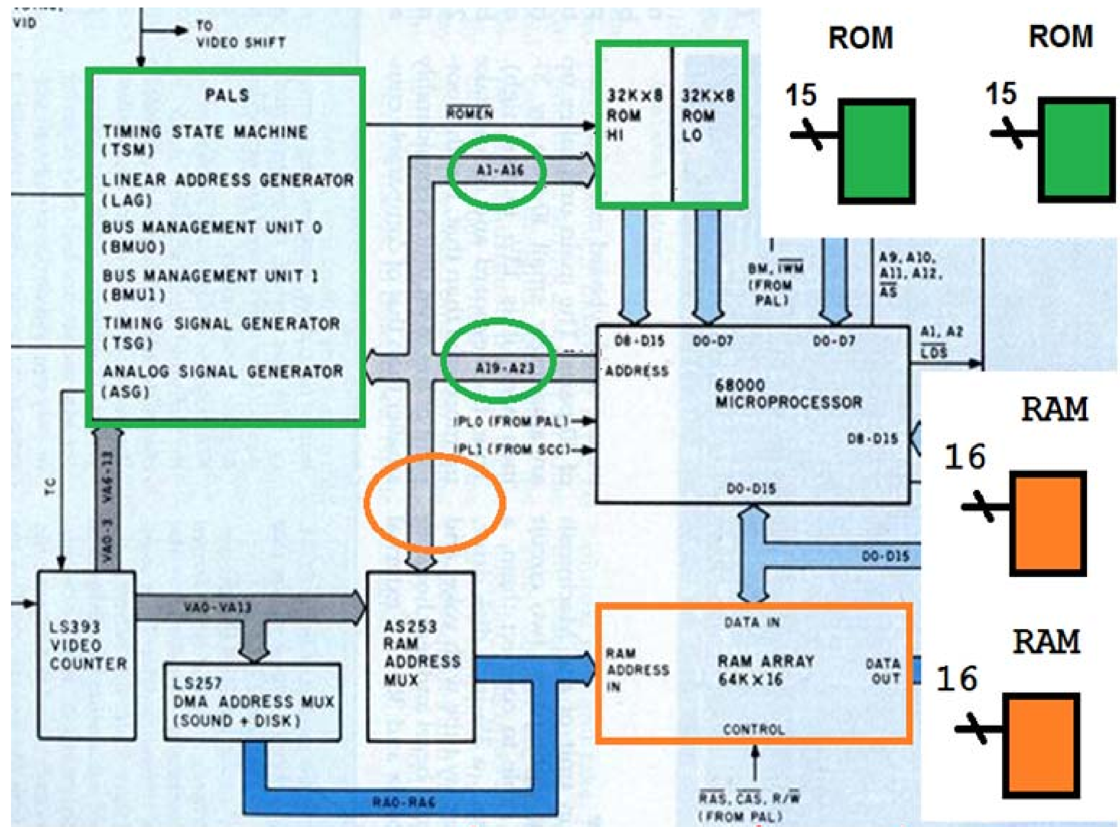


Apple Macintosh Circuit Diagram



Apple Macintosh MEM Decoding

Burrell was working in Apple's service department when he helped Bill Atkinson add more memory to an Apple II computer in an innovative fashion. Bill recommended him to Jef Raskin, who was looking for a hardware engineer to help him with his newly formed Macintosh project.[1] As a member of the design team,[2] Burrell designed five different motherboards during the course of Macintosh development, all of which used techniques based on Programmable Array Logic (PAL) chips to achieve maximum functionality with a minimal chip count.



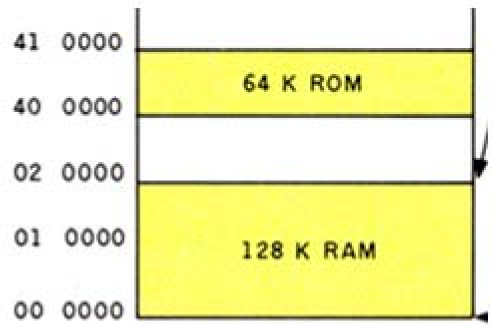
Design for Apple Macintosh MEM Decoding

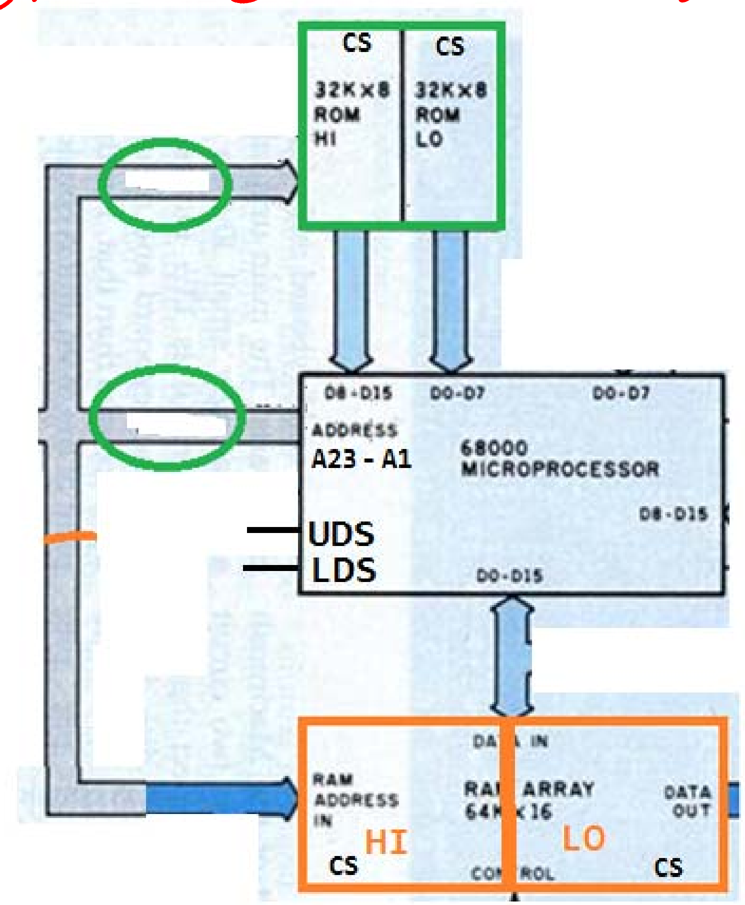
Oct 1 – midterm Exam

5:30 – 6:30 (1 hour)

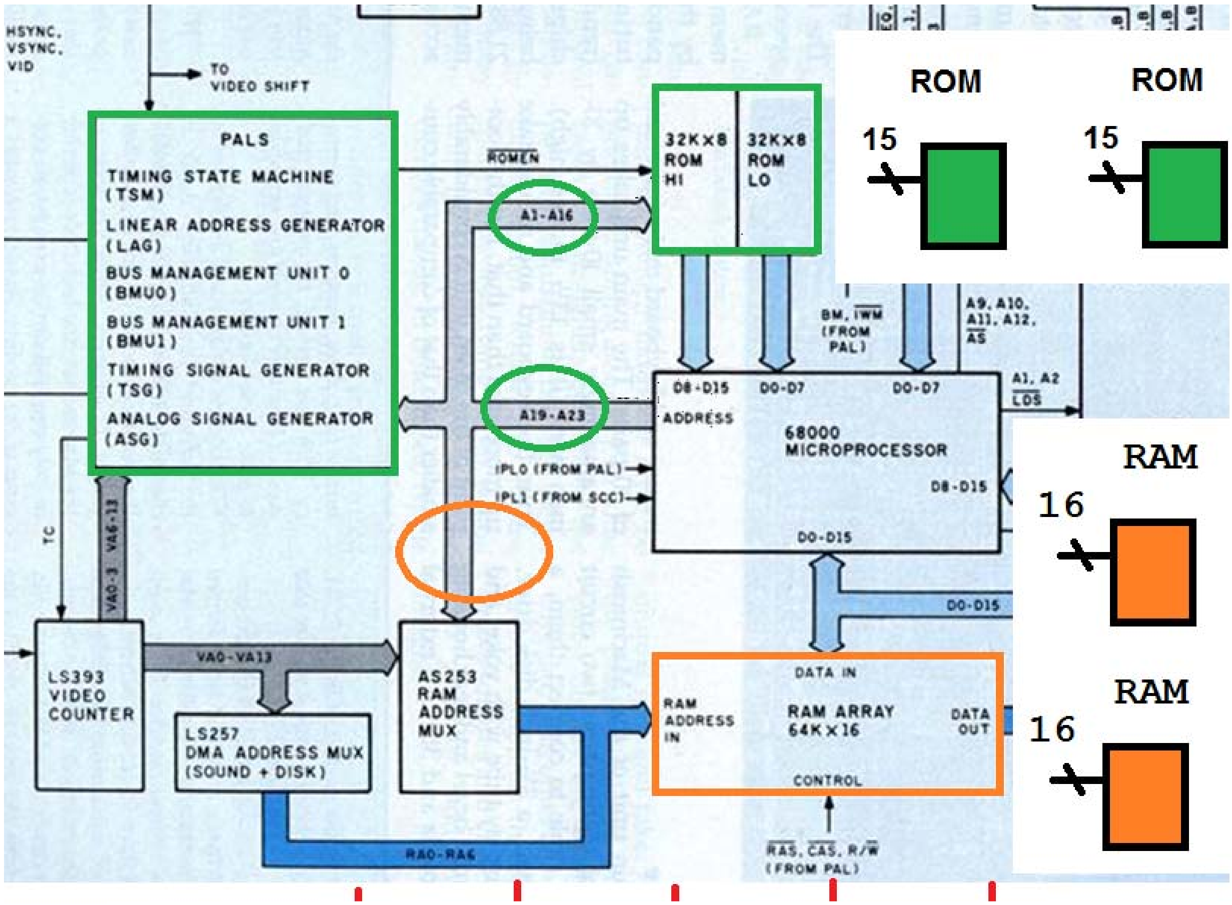
(① Number system 10%

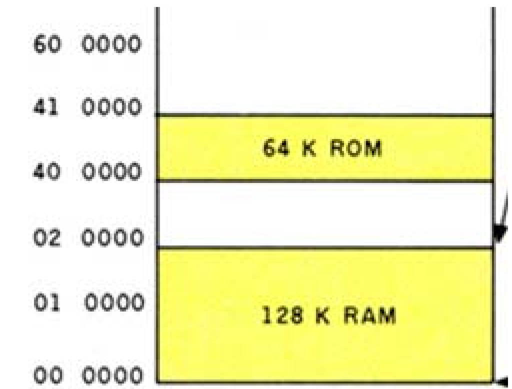
② memory decoding 90%)





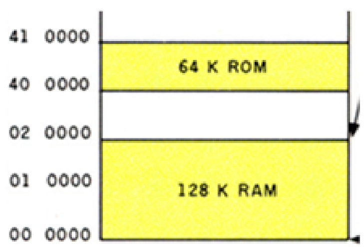
Apple Macintosh MEM Decoding

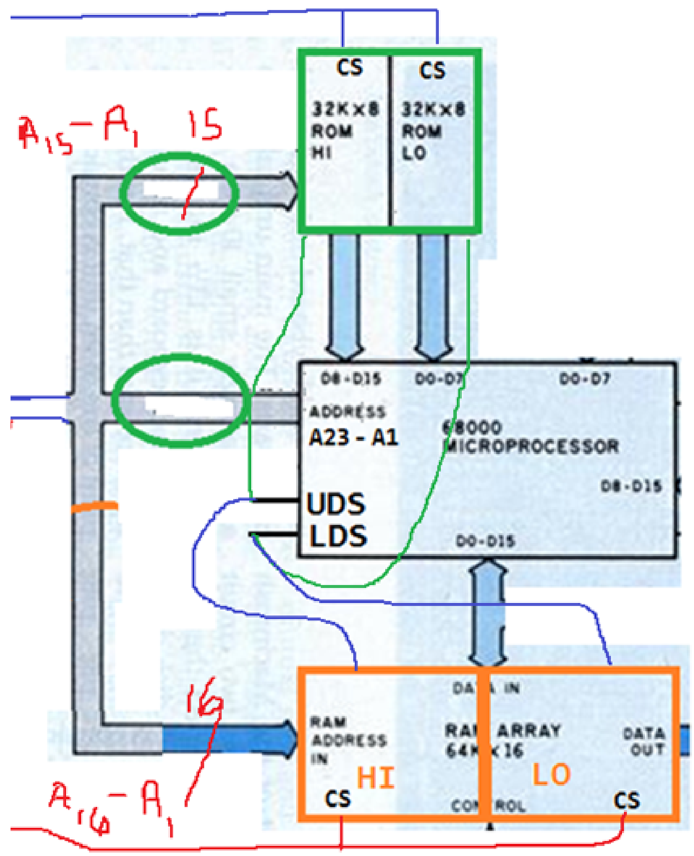




Design for Apple Macintosh MEM Decoding

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	H/L 0	
0	1	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	ROM
0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	RAM





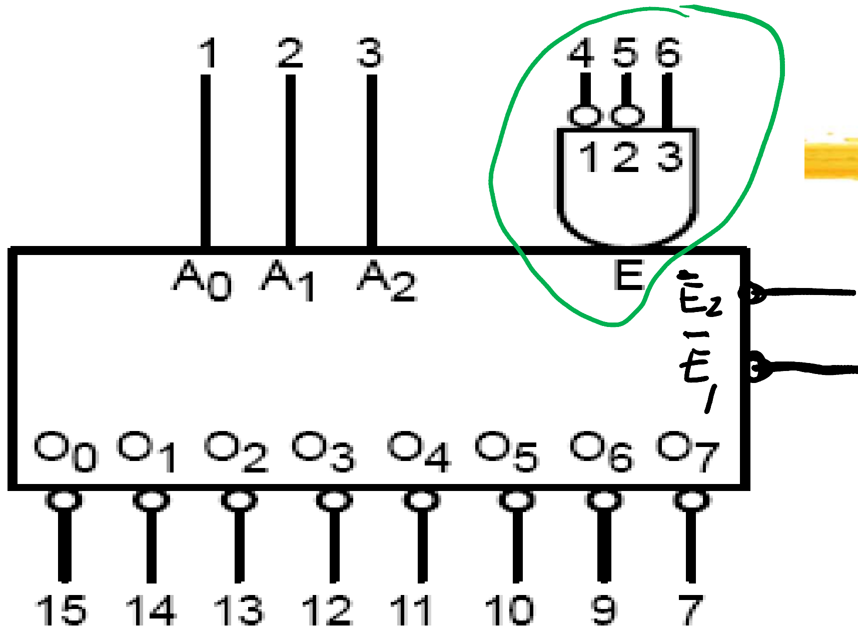
Decoder/Multiplexer

- 74138

MOTOROLA

1-OF-8 DECODER/ DEMULTIPLEXER

LOGIC SYMBOL



$\overline{E_1}$	$\overline{E_2}$
L	L

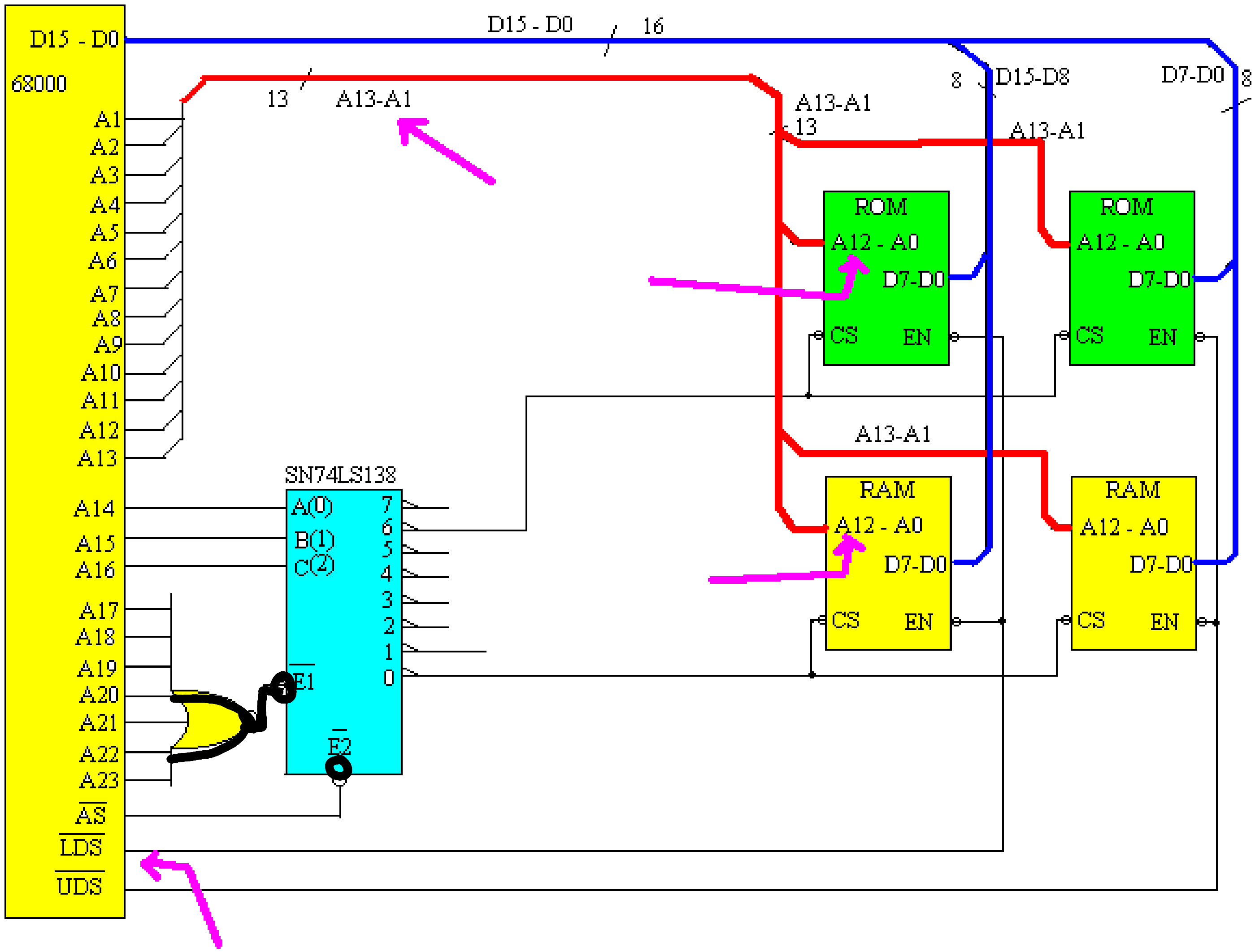
2^0 2^1 2^2

TRUTH TABLE

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

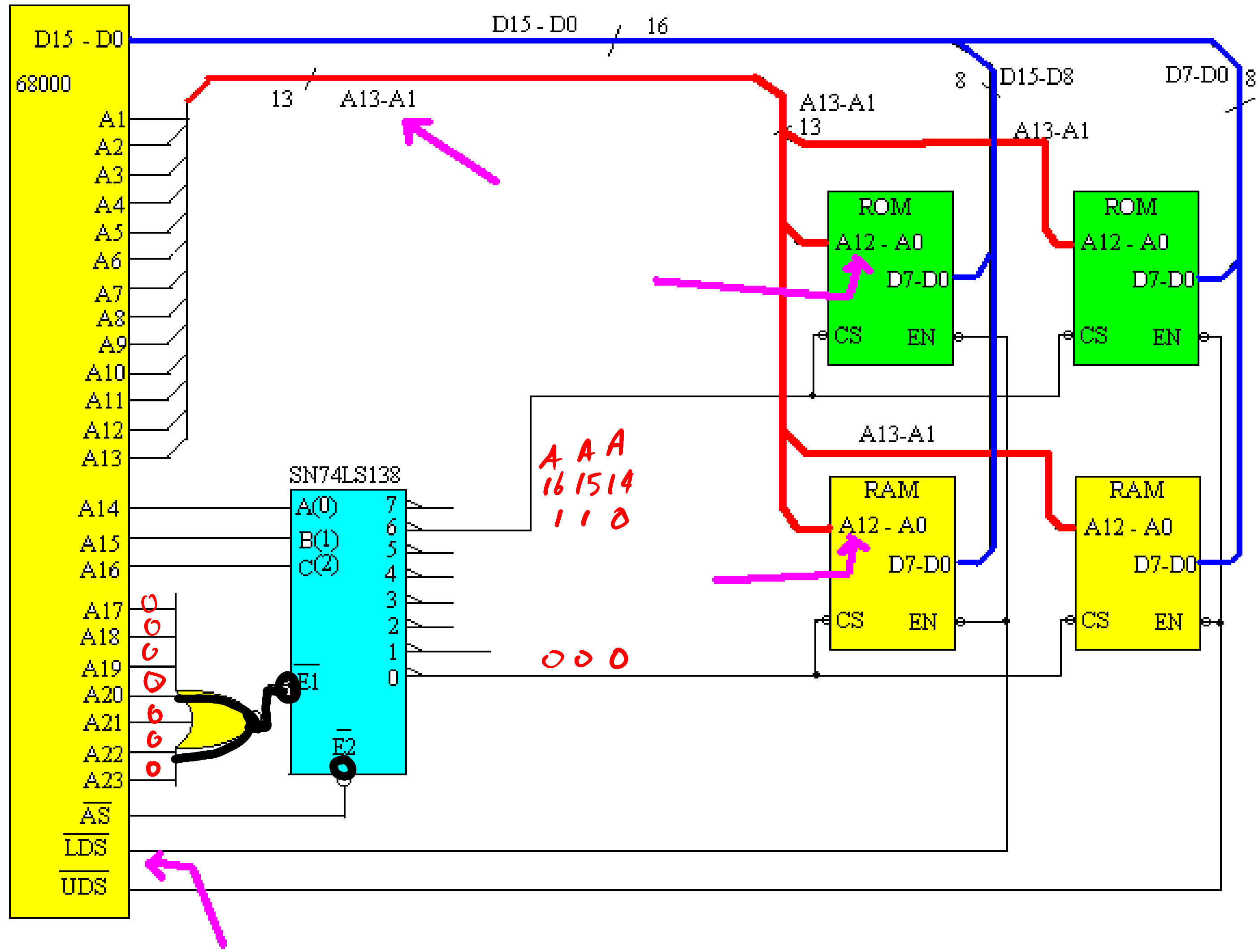
0 1 2 3 6 7

Memory Decoding with Byte/Word Access



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

Memory Decoding with Byte/Word Access - SOLUTION



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

Solution 2

ROM (Left) : $2^{13} = 2^3 \cdot K = 8$ KB (Upper Byte)

ROM (Right) : 8KB (Lower Byte)

RAM (Left) : 8KB (Upper Byte)

RAM (Right) : 8KB (Lower Byte)

$\overline{E} \leftarrow$ A23 ~ A17 = 0 [MEM SEL]

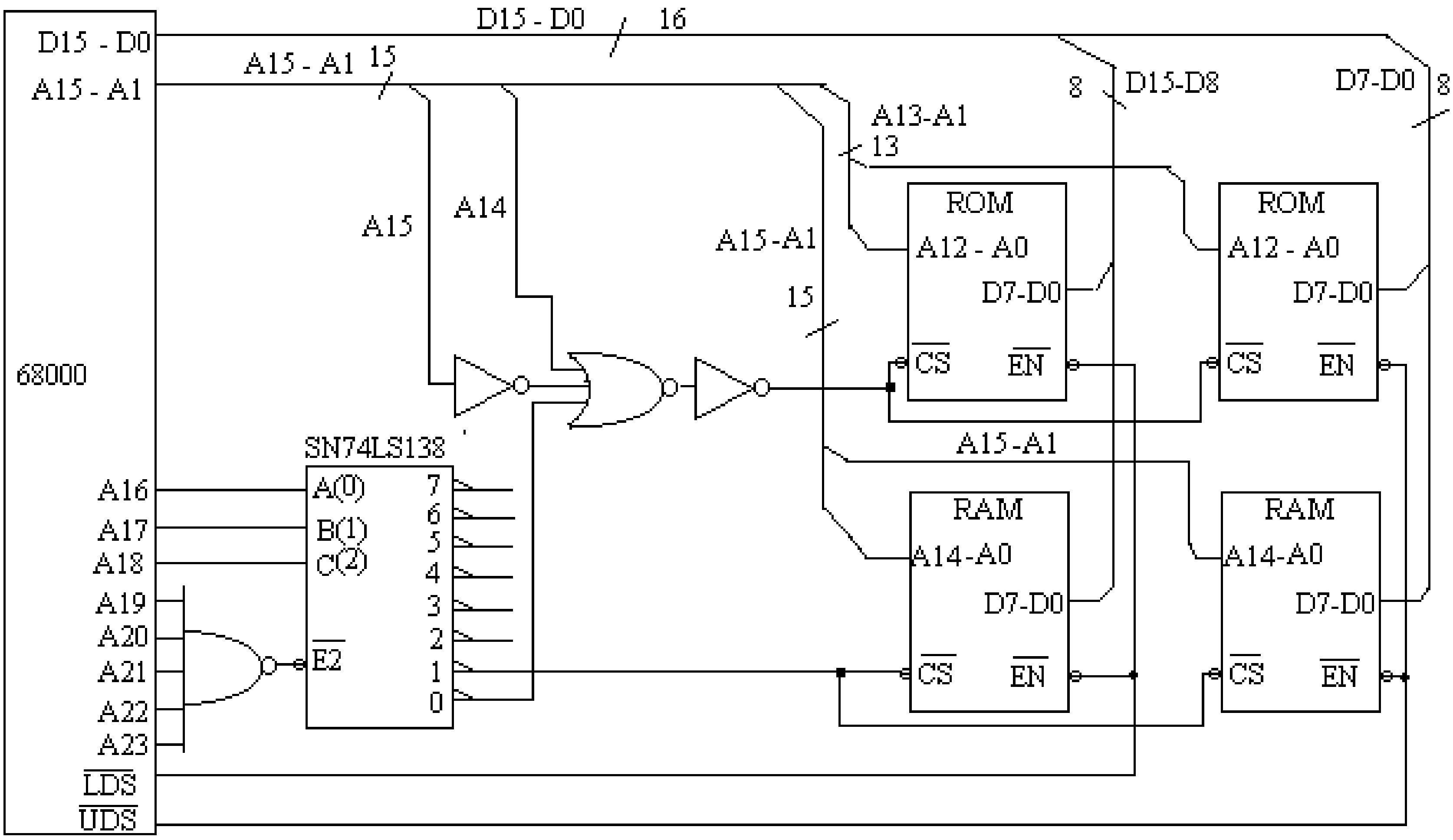
$0 \leftarrow$ A16 ~ A14 = 0 [RAM]

A16 = 1 A15 = 1 A14 = 0 [ROM]

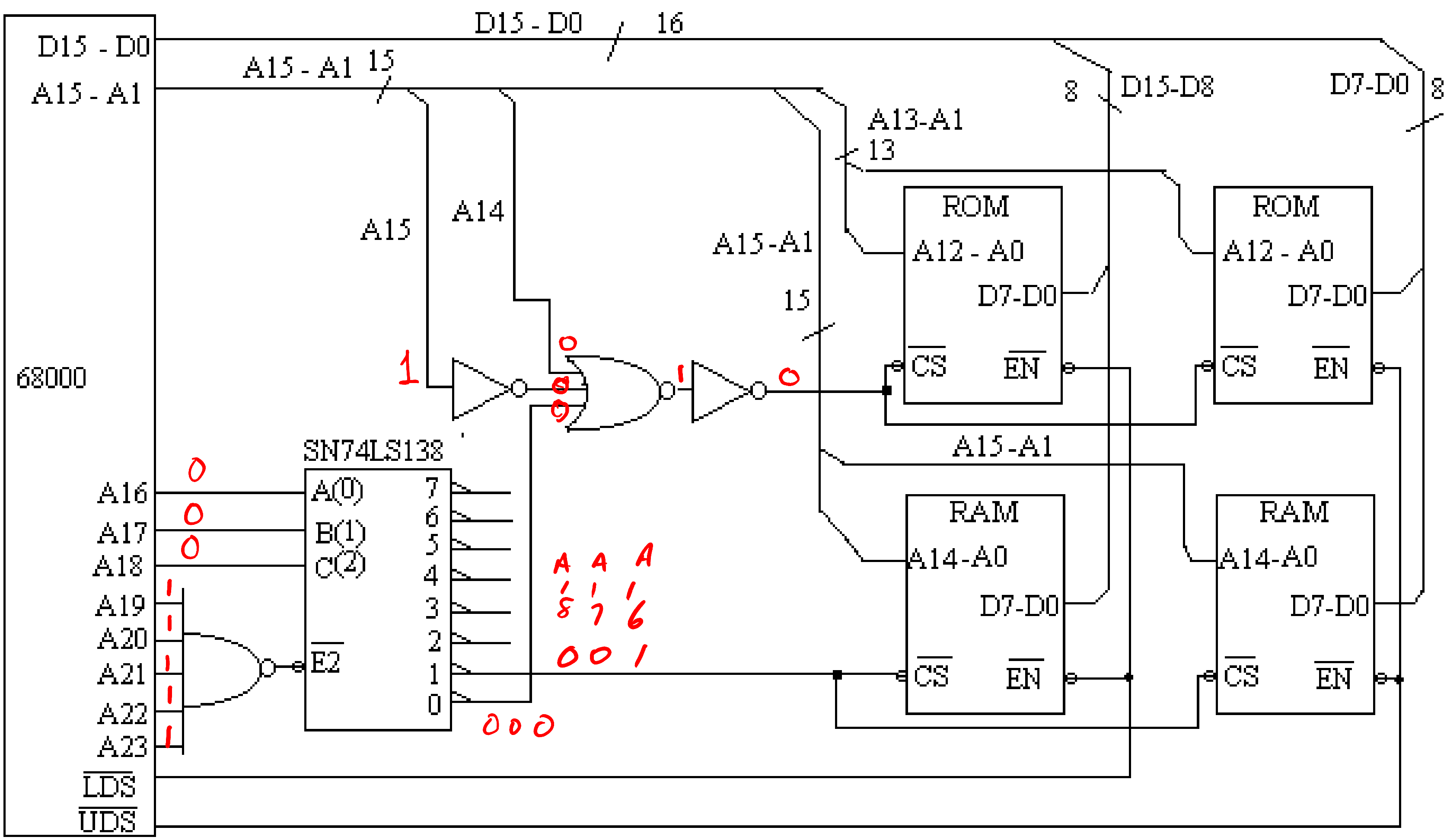
$\left\{ \begin{array}{l} 000\ 000 \\ 003\ FFF \end{array} \right.$ RAM

$\left\{ \begin{array}{l} 018000 \\ 01B000 \end{array} \right\}$ ROM

Can You Draw a Memory Map of this?



Can You Draw a Memory Map of this? - SOLUTION



Solution 3

RAM $\{2^{15} = 32KB\} \times 2 \rightarrow 64kB$

Rom $\{2^{13} = 8kB\} \times 2 \rightarrow 16KB$

<MEM SELECTION>

① $\overline{E_2}$: $A23 \sim A19 = 1$ $(\overline{EN})$

② $\overline{CS}$ for RAM (1): $A18=0$, $A17=0$, $A16=1$

③ $\overline{CS}$ for RoM [complex]: $A18=0$, $A17=0$, $A16=0$
$A14=0$
$A15=1$

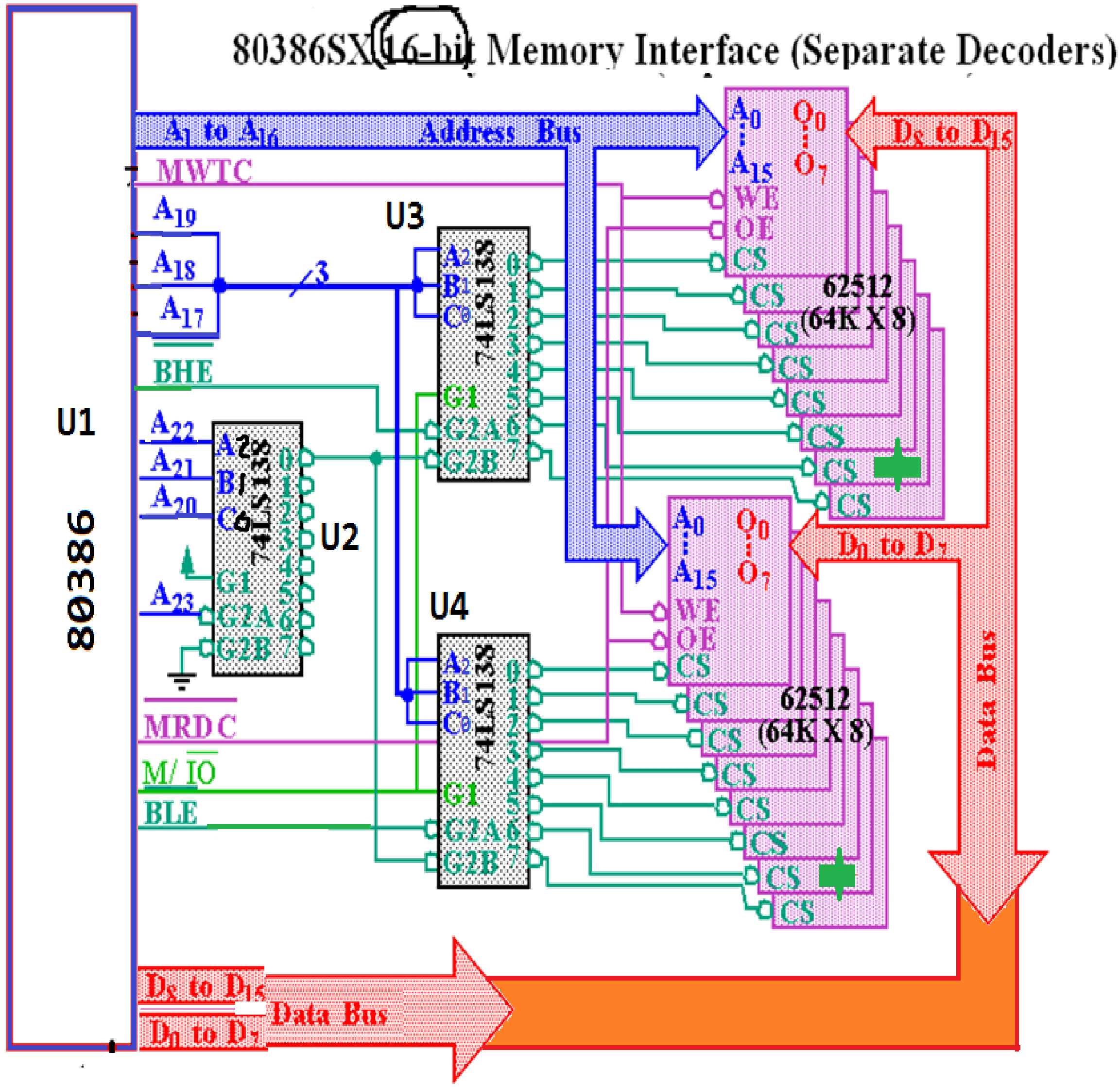
RAM

RoM

RAM: F90000 ~ F9FFFF
RoM: F88000 ~ F8BFFF

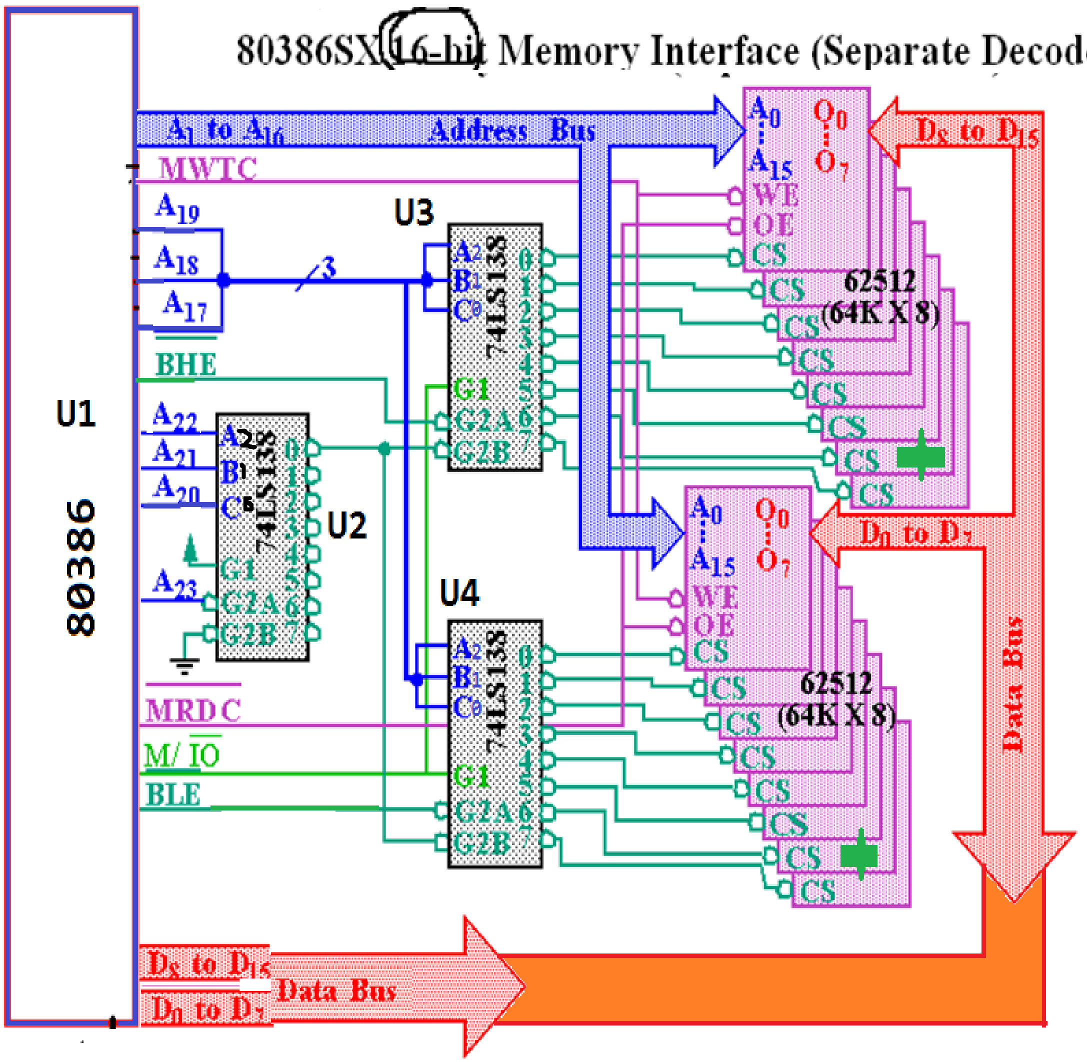
Intel 80386 Memory – Decoding – Quiz

- A pair of memory pack (High and Low bytes) - Eight (8) 64KB Memory Chips
- At each pack, each memory chip is SELECTED (/CS) by an output from U3 (High Byte) and U4 (Low Byte), which are enabled by the single output from U2.
- **Question**: Find the address range of the **second** chip of the pair of memory pack. (Green Mark)
- Ignore MWTC and MRDC, M/IO lines (they are for Write, Read, Memory or I/O operations, respectively)

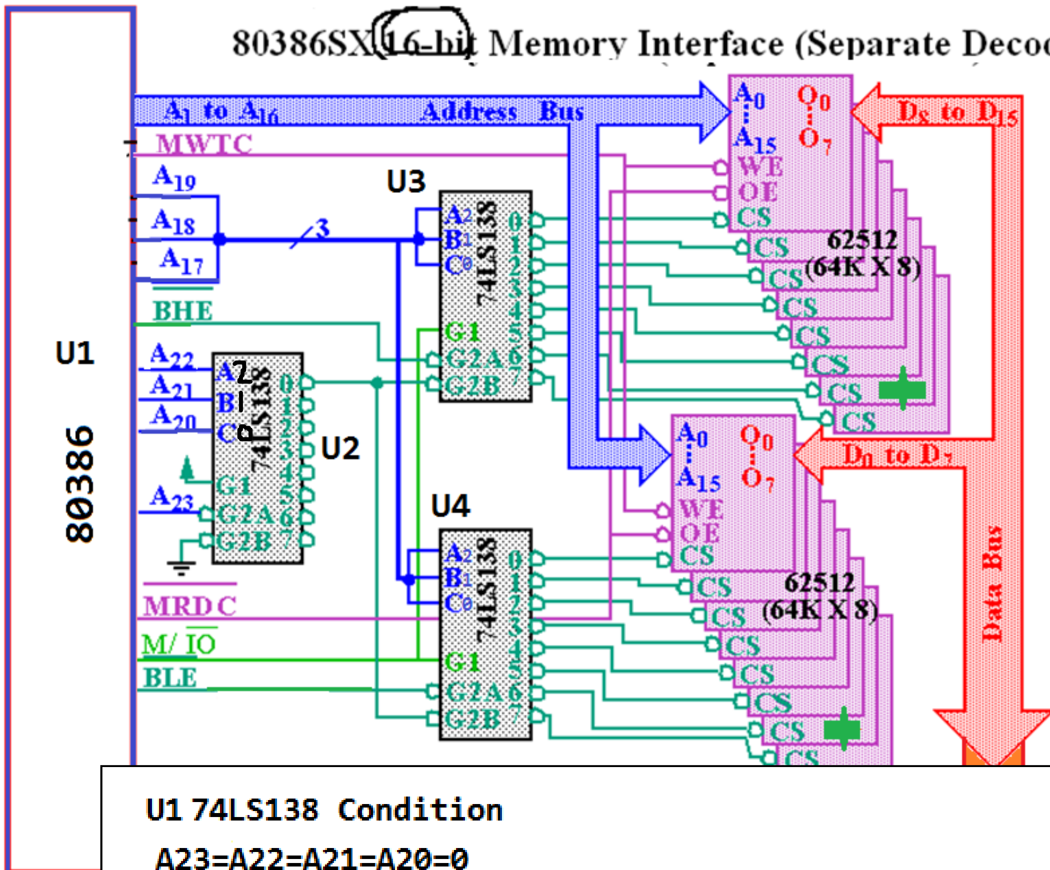


Intel 80386 Memory – Decoding – Quiz

- Find the address range of the **second** chips of the memory pair. (Green Mark)
- **Quiz**: **Individual Work** – (**Name:______________________________**)



Intel 80386 Memory – Decoding – Class Activity



U1 74LS138 Condition

A23=A22=A21=A20=0

U2 & U3 74LS138 Condition
for the second Memory Pair Selection

A19 A18 A17
1 1 0

Entire Memory Space ? 000000 - 0FFFFF