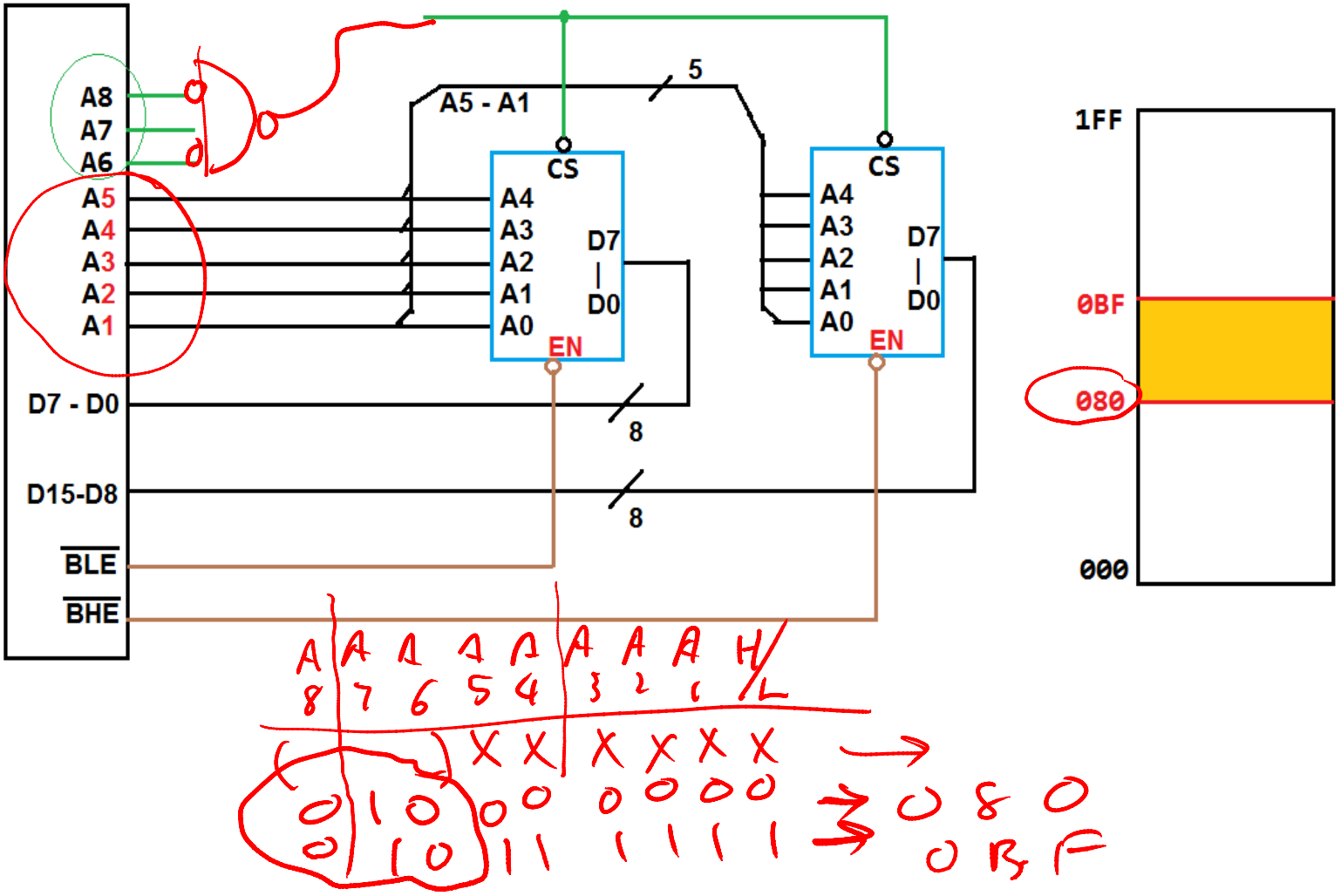
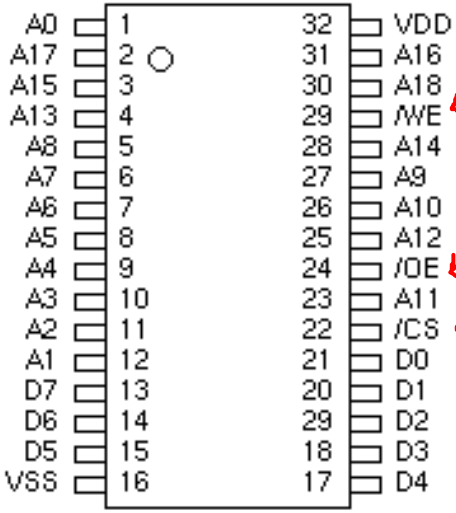
Memory Address 16 -- ?

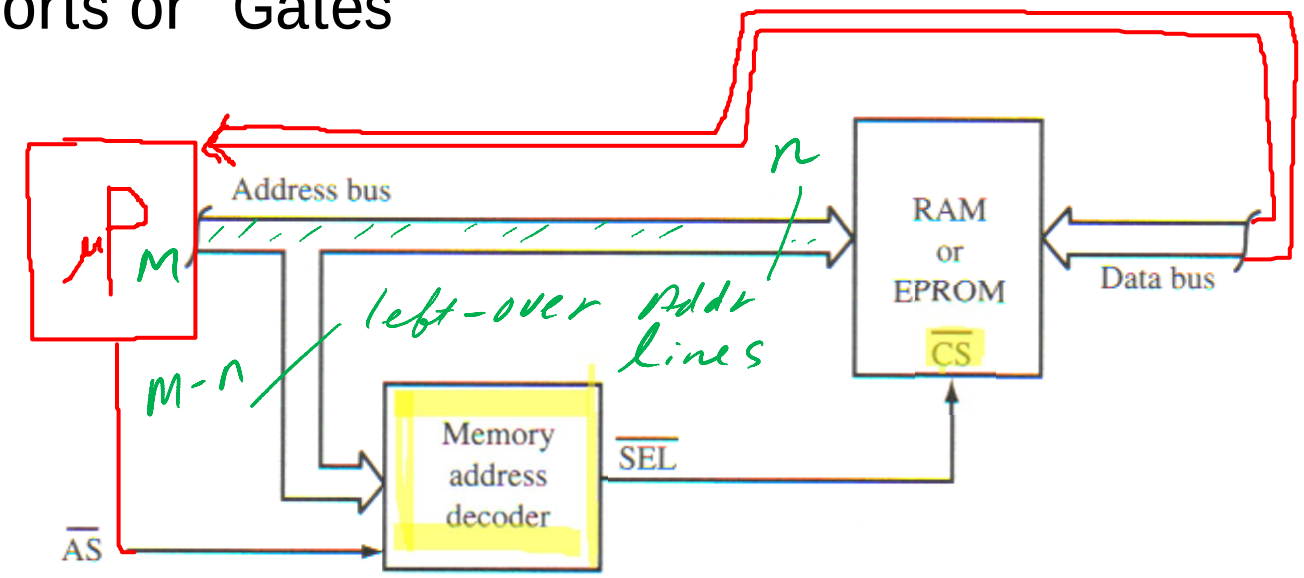
- Design a memory decoder for the depicted address segment.
- NOTE: **BLE/BHE → EN (each)** and **MEM Decoding → CS (common)**



Recap & More Exercise for - Memory Address Decoding

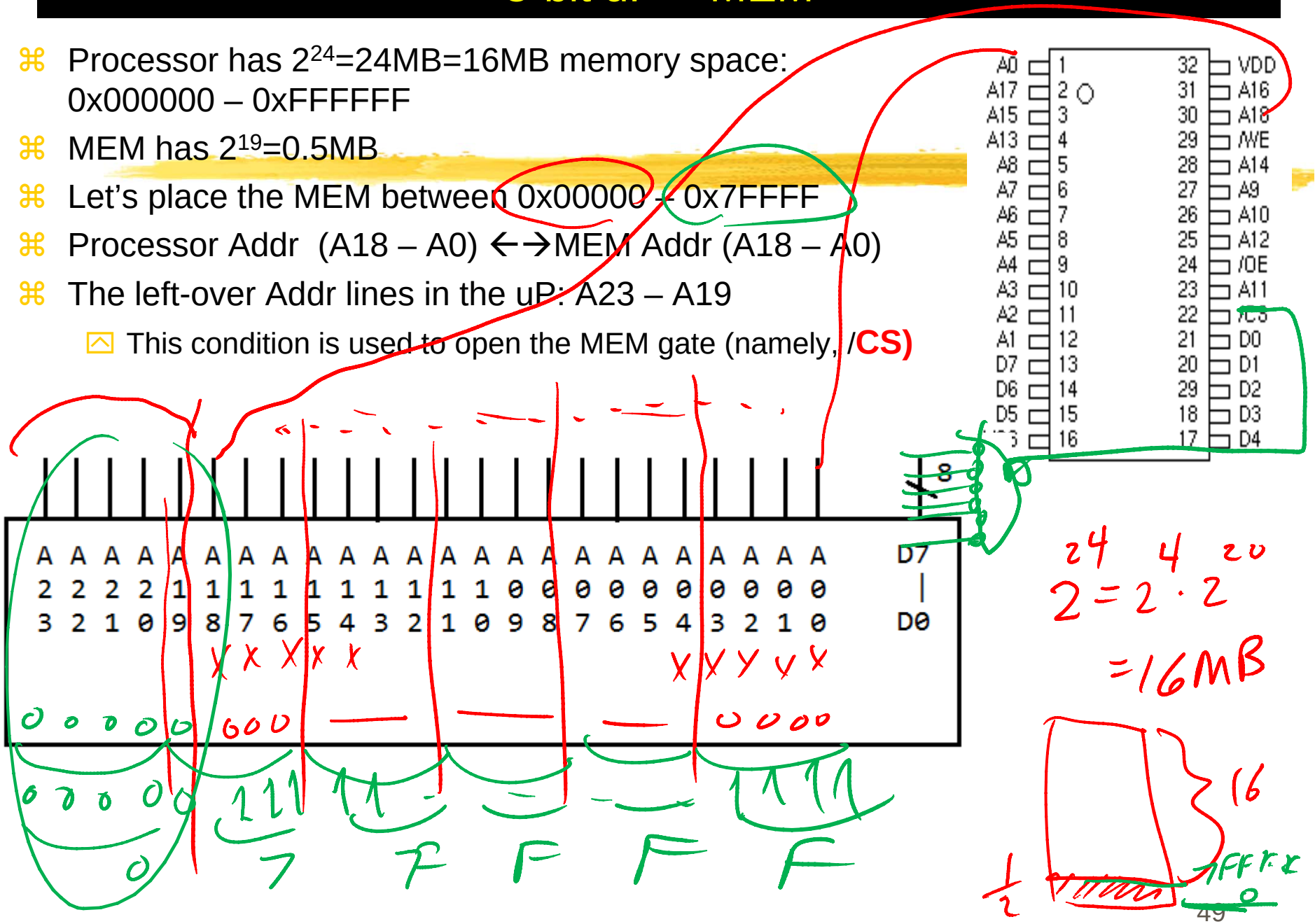
- How Much Memory?
  - How Many Address Lines?
    - 1K → 10 lines
    - MEM 32K → 15 lines
- MEMORY PINOUTS
  - Address Lines
  - Data Lines
  - "Defense" ports or "Gates"
    - /CS
    - /CE
    - /OE
    - /WE



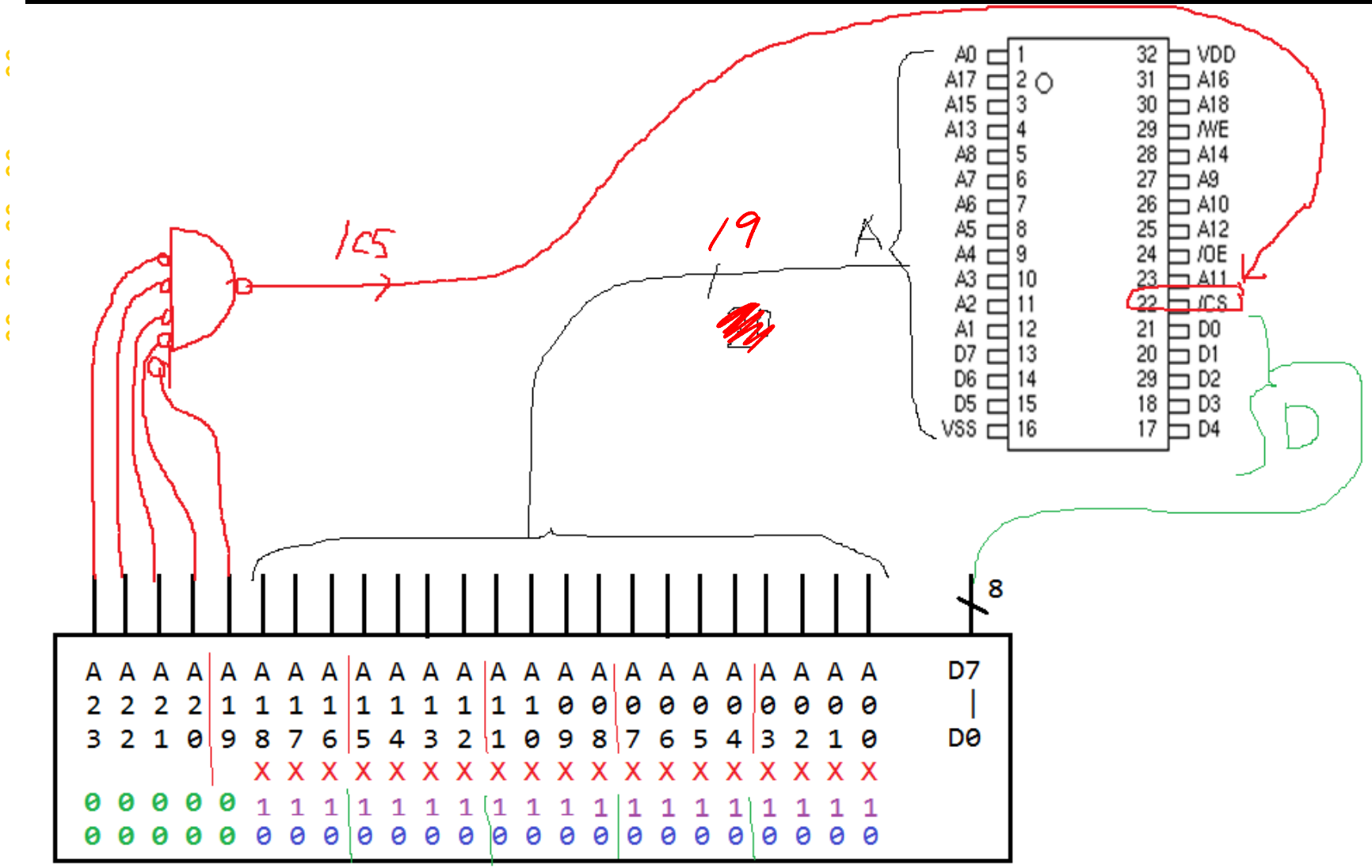


8-bit uP + MEM

- Processor has 2^{24}=24MB=16MB memory space: 0x000000 – 0xFFFFFF
- MEM has 2^{19}=0.5MB
- Let's place the MEM between 0x00000 - 0x7FFFF
- Processor Addr (A18 – A0) ←→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)**

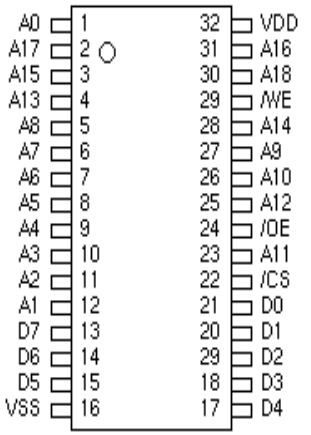


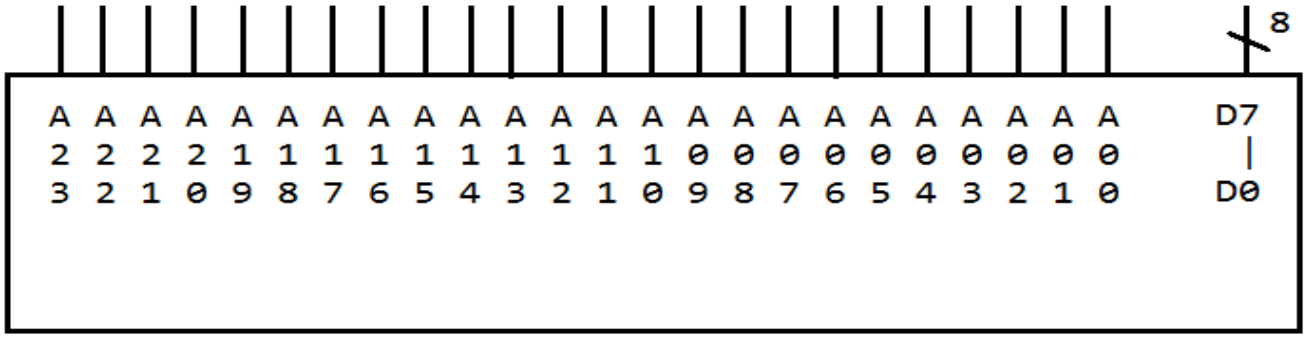
8-bit uP + MEM


A0 1
A17 2
A15 3
A13 4
A8 5
A7 6
A6 7
A5 8
A4 9
A3 10
A2 11
A1 12
D7 13
D6 14
D5 15
VSS 16
32 VDD
31 A16
30 A18
29 /WE
28 A14
27 A9
26 A10
25 A12
24 /OE
23 A11
22 /CS
21 D0
20 D1
29 D2
18 D3
17 D4
CS
19
A
D
8
A A A A A A A A A A A A A A A A A A A A A A A A
2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
X X X X X X X X X X X X X X X X X X X X
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
D7
D0

8-bit uP + MEM Exercise

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

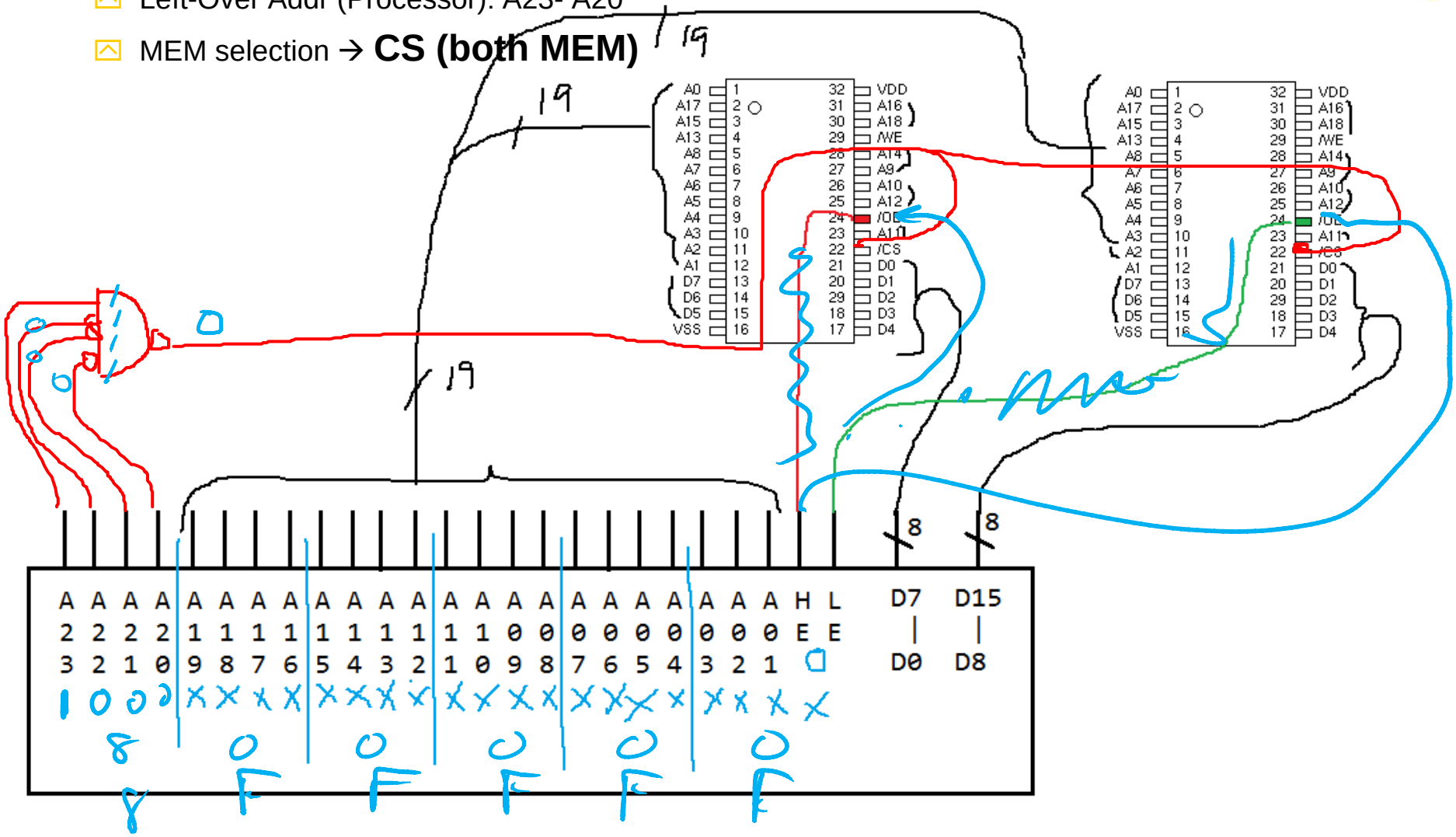




Names:_____

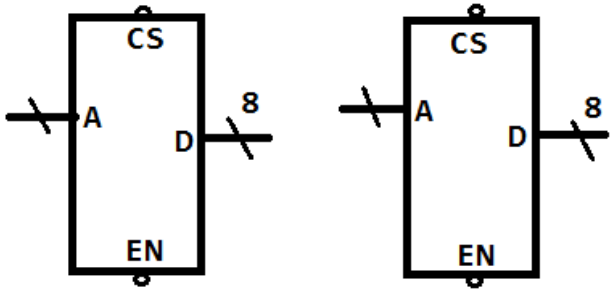
16-bit uP + MEM

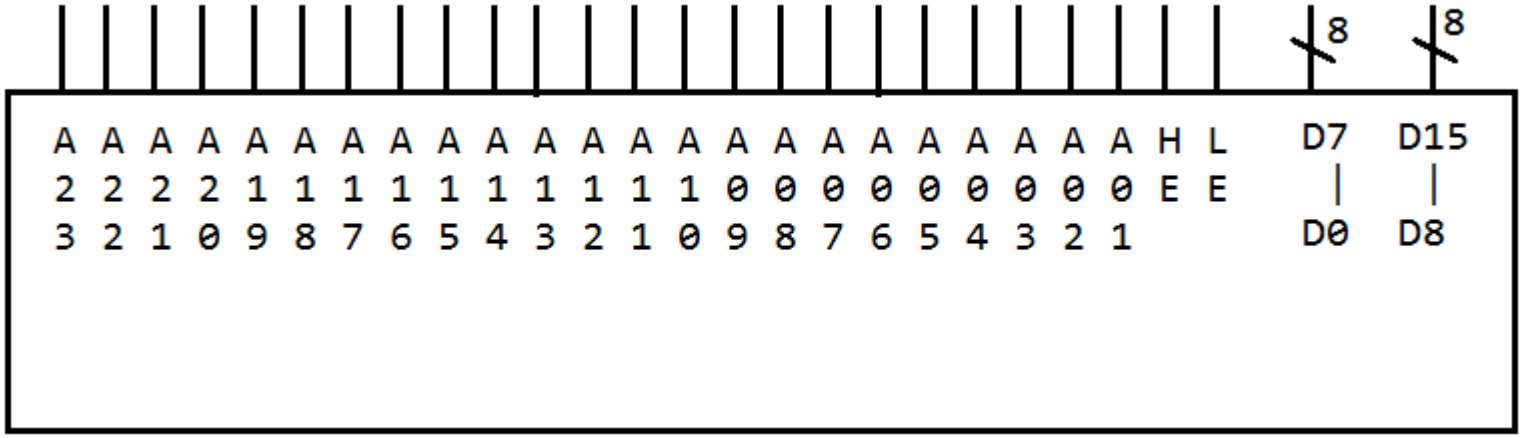
- Processor does not have A0
  - BHE (UDS) and BLE (LDS), instead.
  - BHE, BLE -→ **EN (Each MEM)**
- **Processor Addr ←→ MEM Addr**
  - A19 – A1 (Processor): A18 – A0 (MEM)
  - Left-Over Addr (Processor): A23- A20
  - MEM selection → **CS (both MEM)**



16-bit uP Memory Decoding EXERCISE

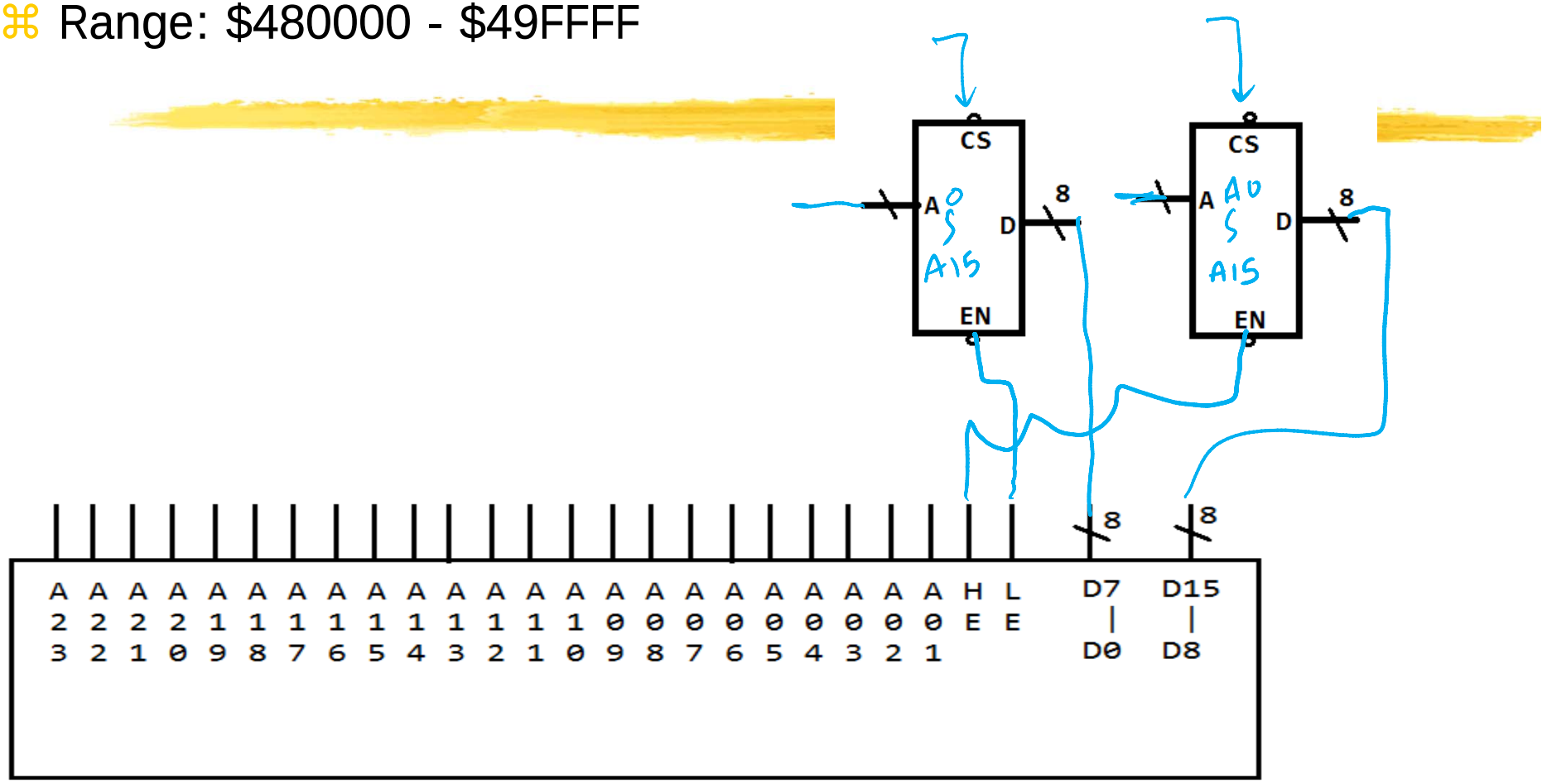
- Q: 64K Word (or 128 KB) of RAM, with it's starting address at $480000
- A: 64KB → 16 lines each MEM
  - Range: **$480000 - $49FFFF**
  - BHE and BLE for A_0 line → **EN(Enable)**
  - Upper address lines → **CS** for both MEM





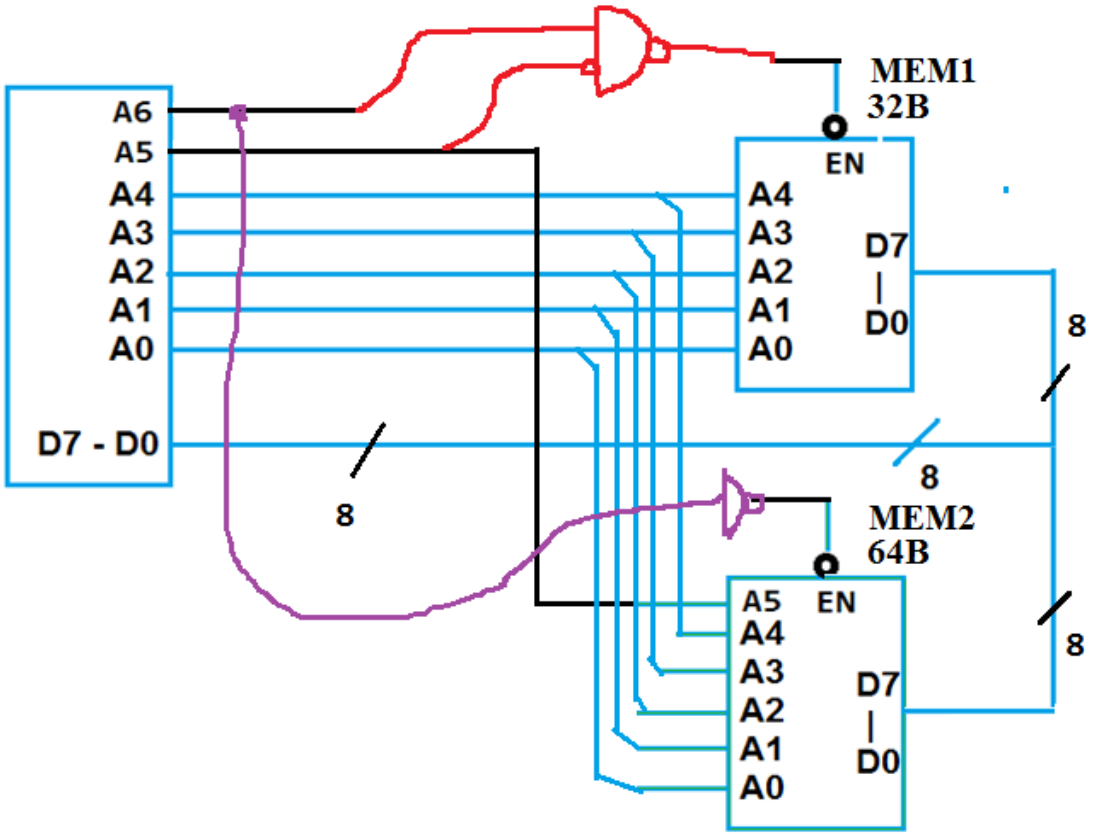
16-bit uP Memory Decoding EXERCISE

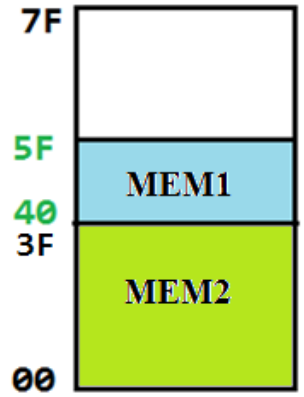
- Range: $480000 - $49FFFF



Names:_____

Multiple Memories





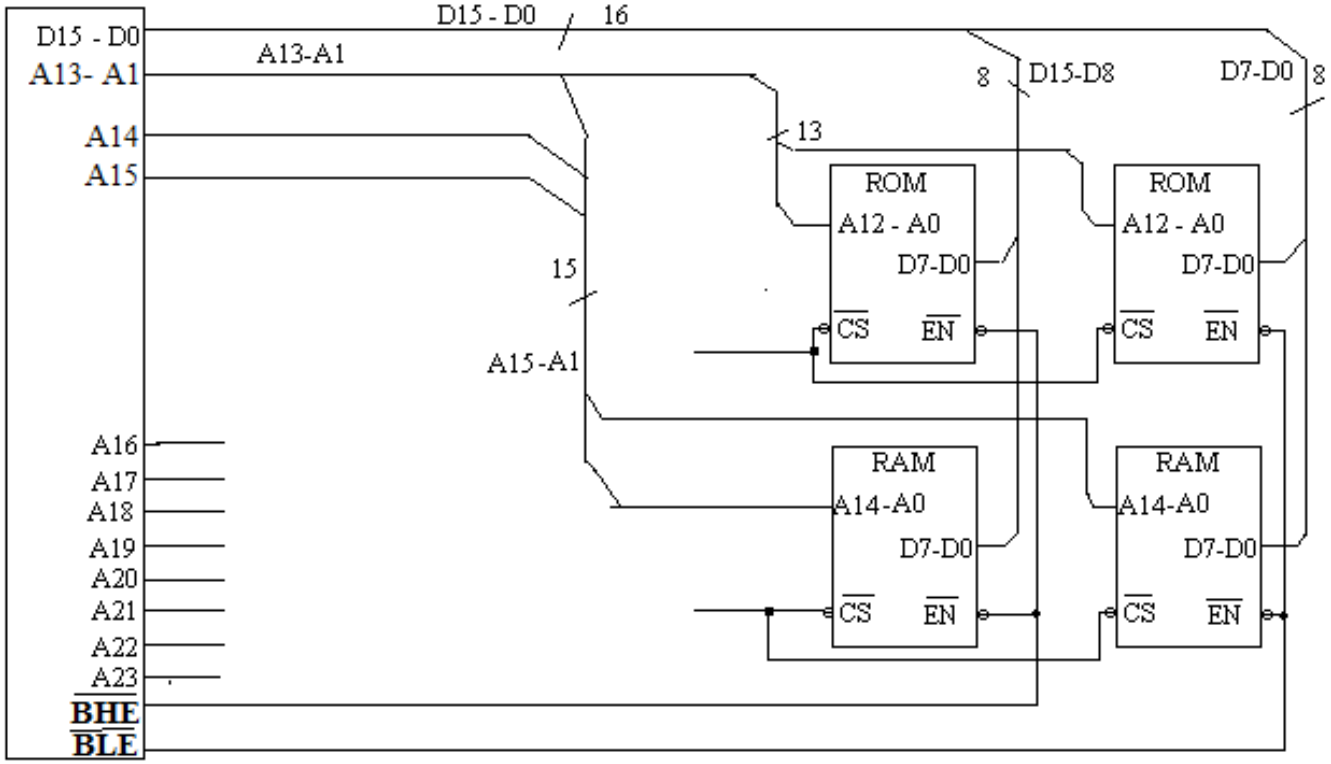
A6	A5	A4	A3	A2	A1	A0	
		X	X	X	X	X	MEM1
1	0	1	1	1	1	1	
1	0	0	0	0	0	0	
		X	X	X	X	X	MEM2
0	1	1	1	1	1	1	

Group #:__ Names:_____

Multiple Memories

ROM: 0xFB0000 - 0xFB3FFF

RAM: 9xF90000 - 0xF9FFFF



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 Po
- Printer F
- Address

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)

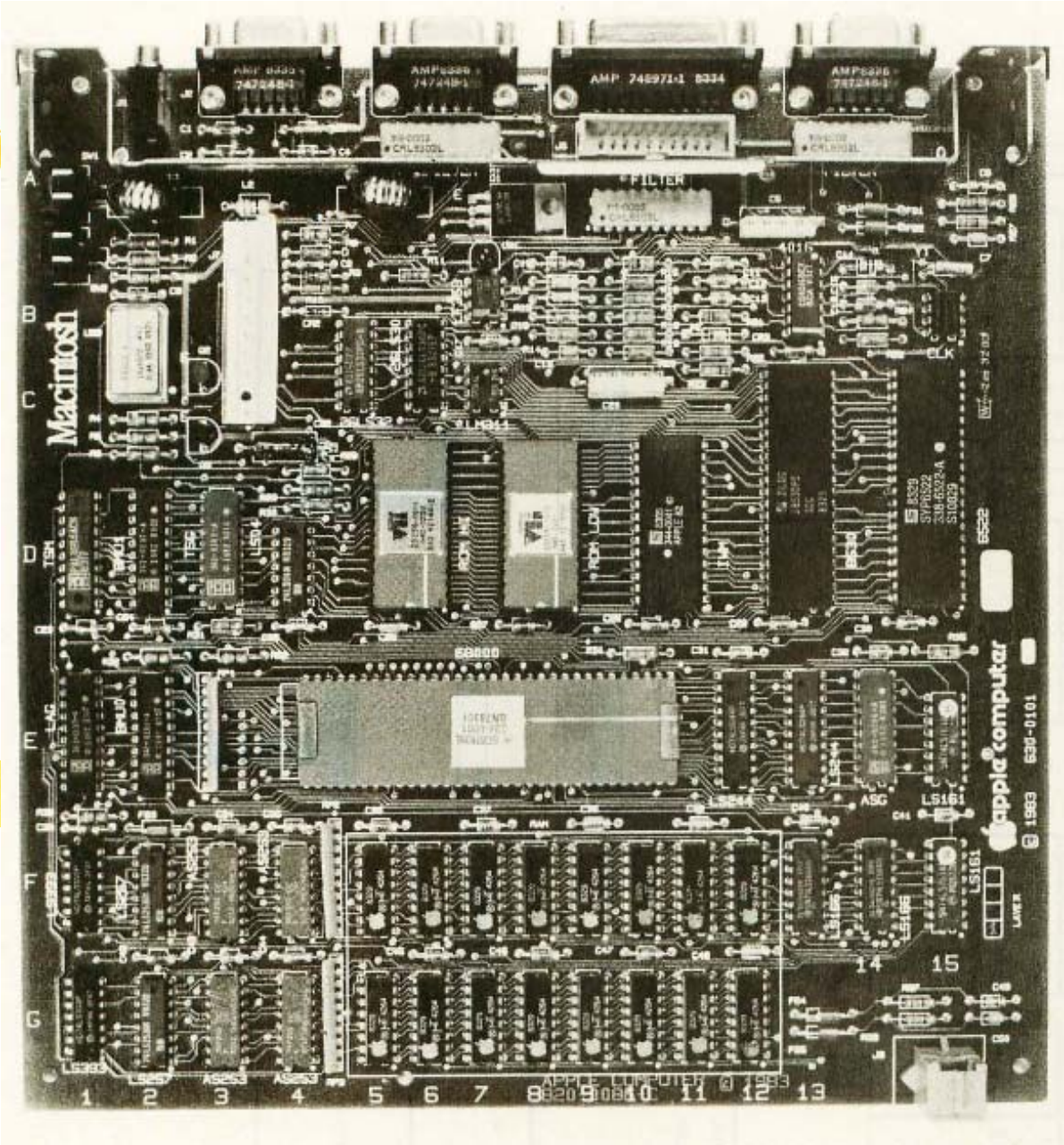
Prices
Standard system, $1995–$2495; Mac Paint and Mac Write (together), bundled at no charge for the first 100 days, $195 (for the two) thereafter; Macintosh Pascal, BASIC, Logo, Terminal, and Assembler/Debugger, $99 each; Mac Draw and Mac Project, $125 each; keypad, $99; second disk drive, $395; Imagewriter 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

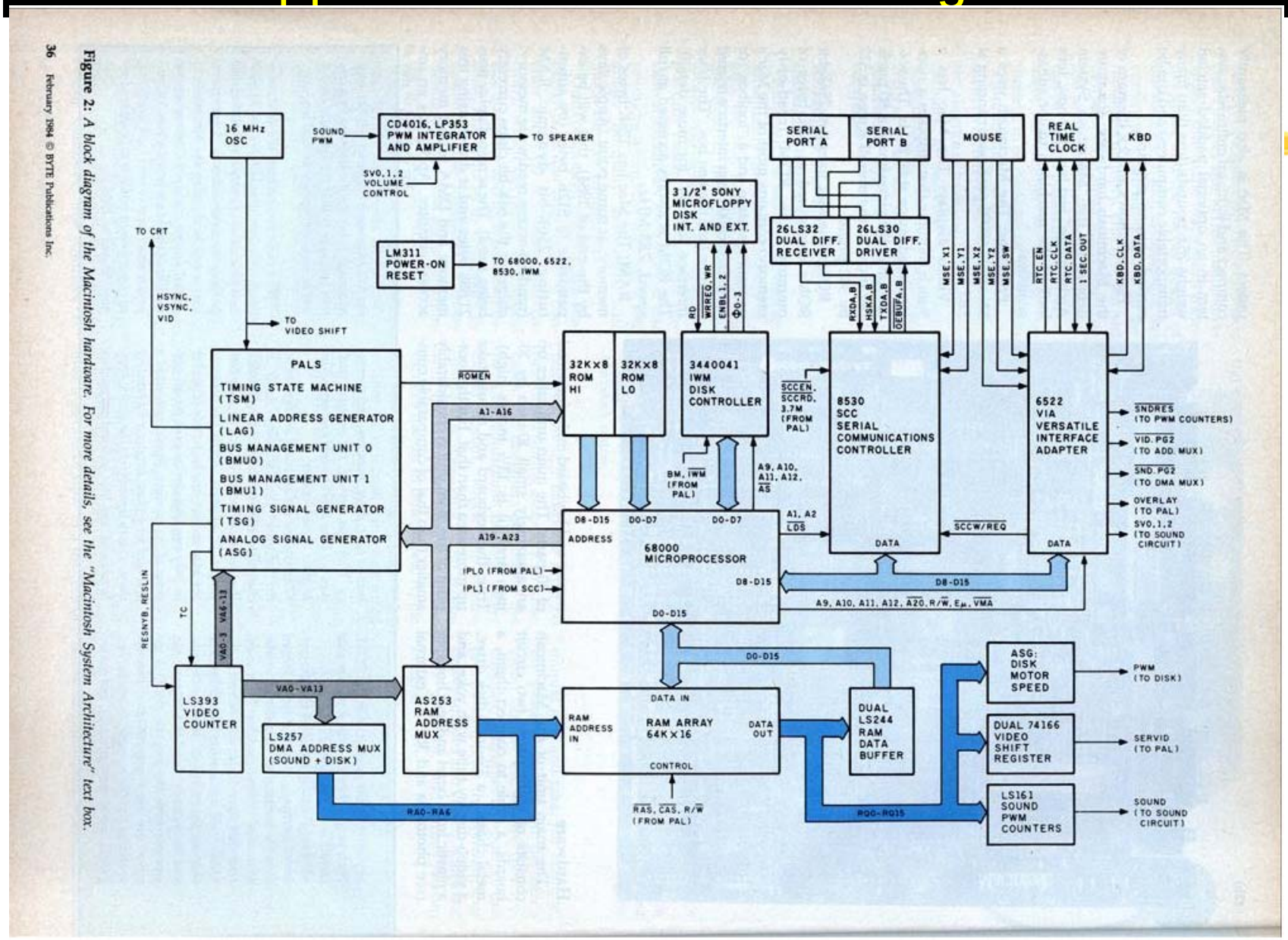
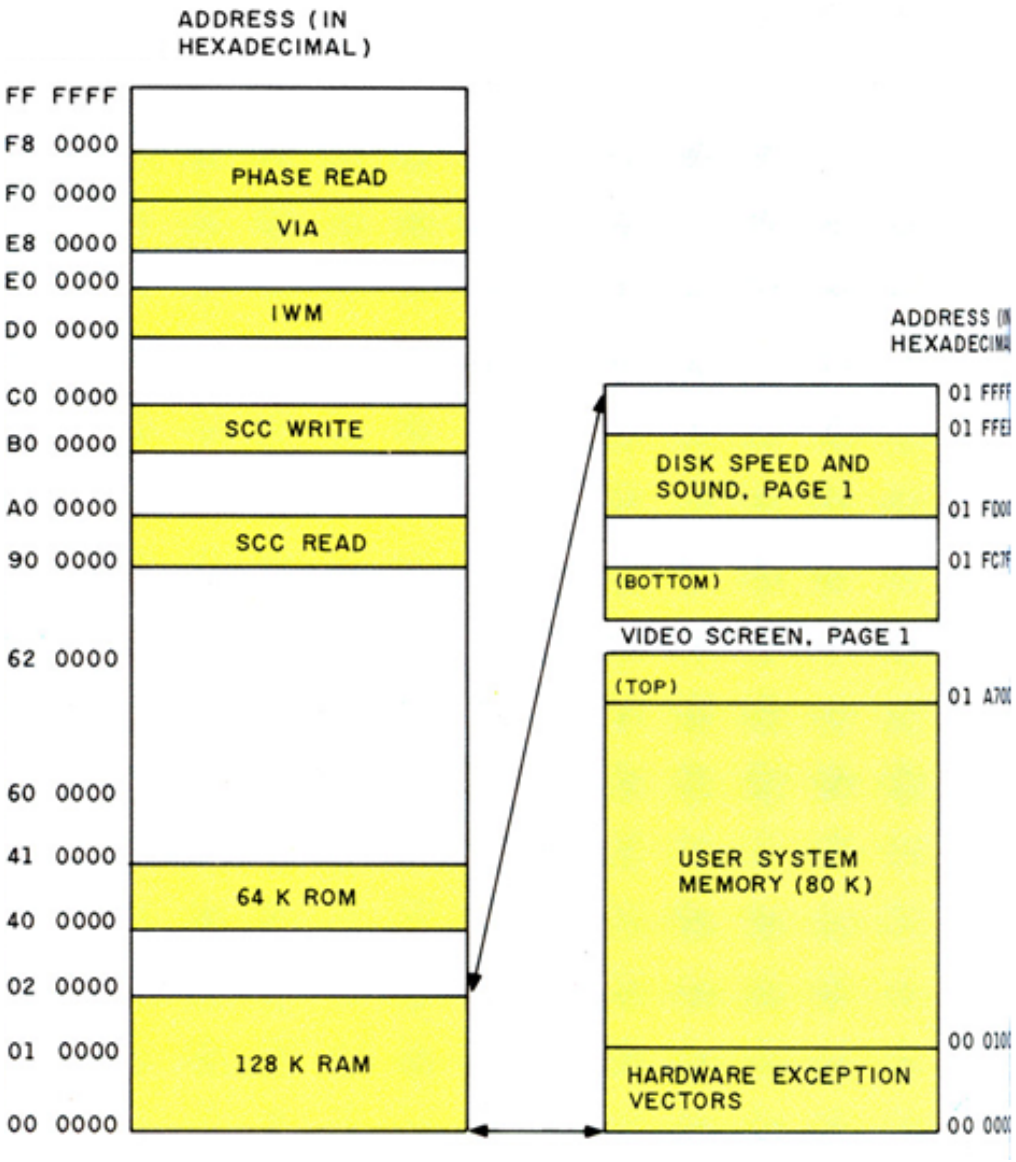


Figure 2: *A block diagram of the Macintosh hardware. For more details, see the "Macintosh System Architecture" text box.*

Memory Map (for Apple Macintosh)

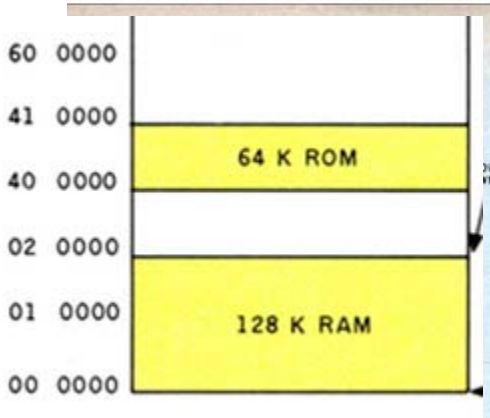


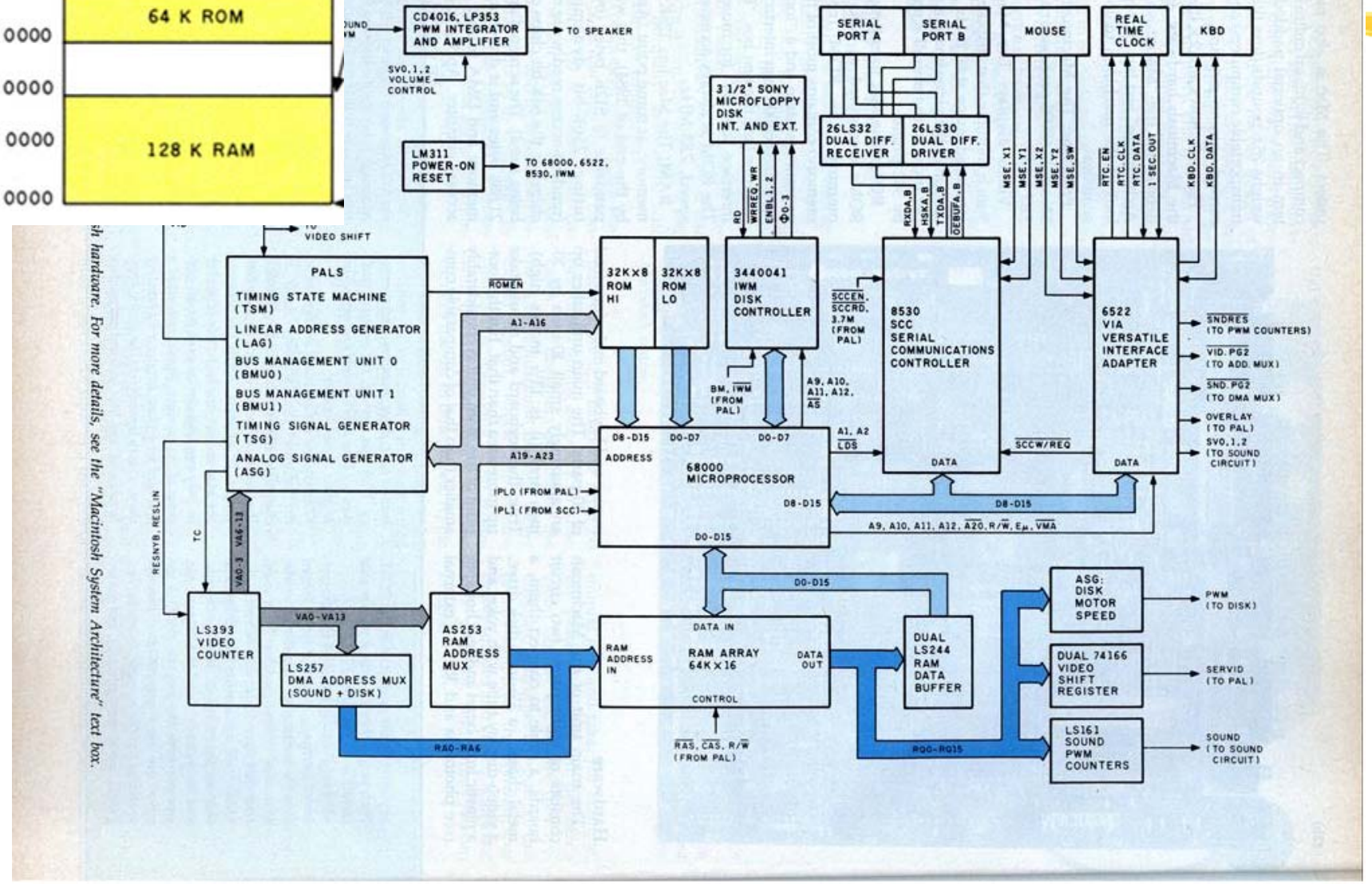
VIA(Versatile Interface Adapter)
---general I/O

IWM (Integrated Woz Machine)
-- floppy disk

SCC (Serial Communications Controller)
-- Serial & Mouse

Apple Macintosh Circuit Diagram

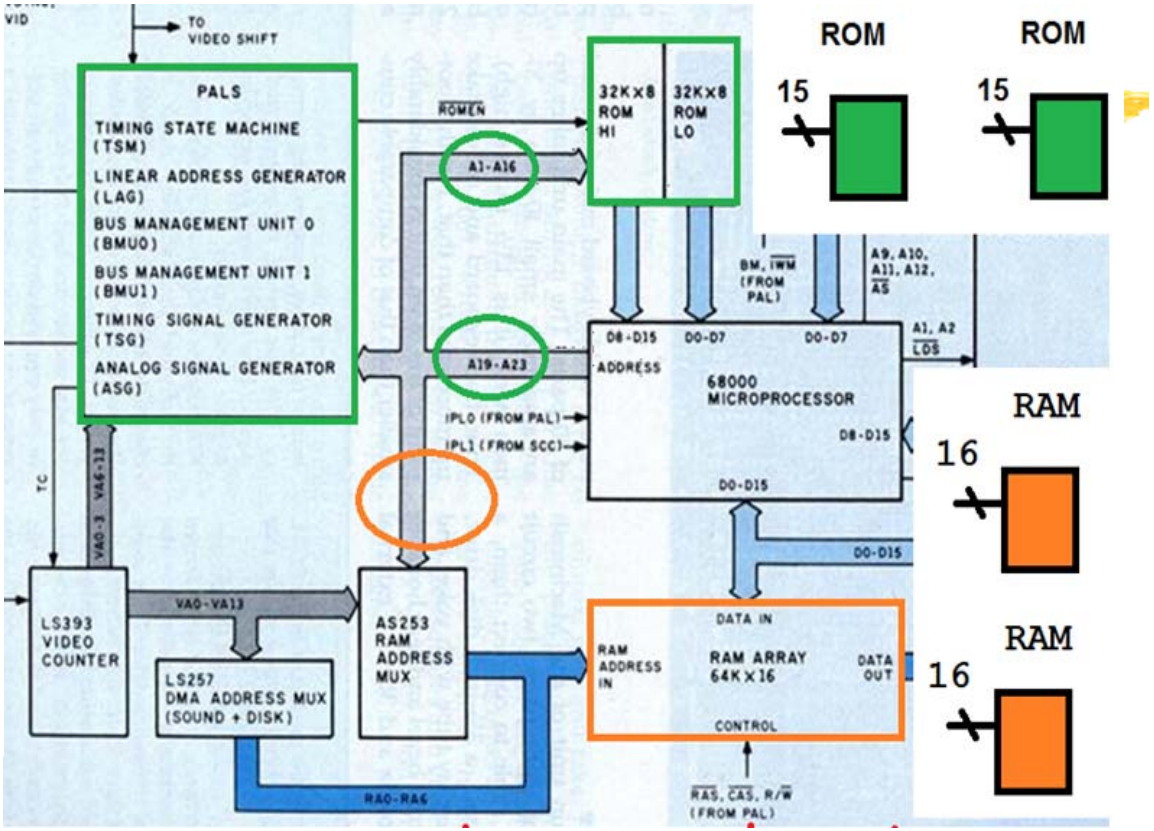




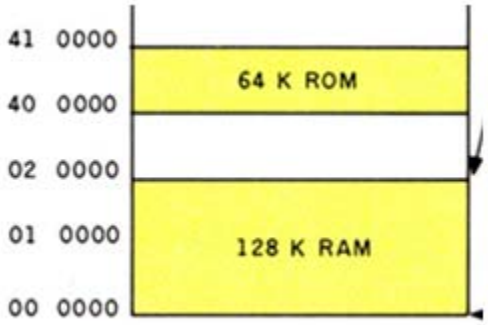
sh hardware. For more details, see the "Macintosh System Architecture" text box.

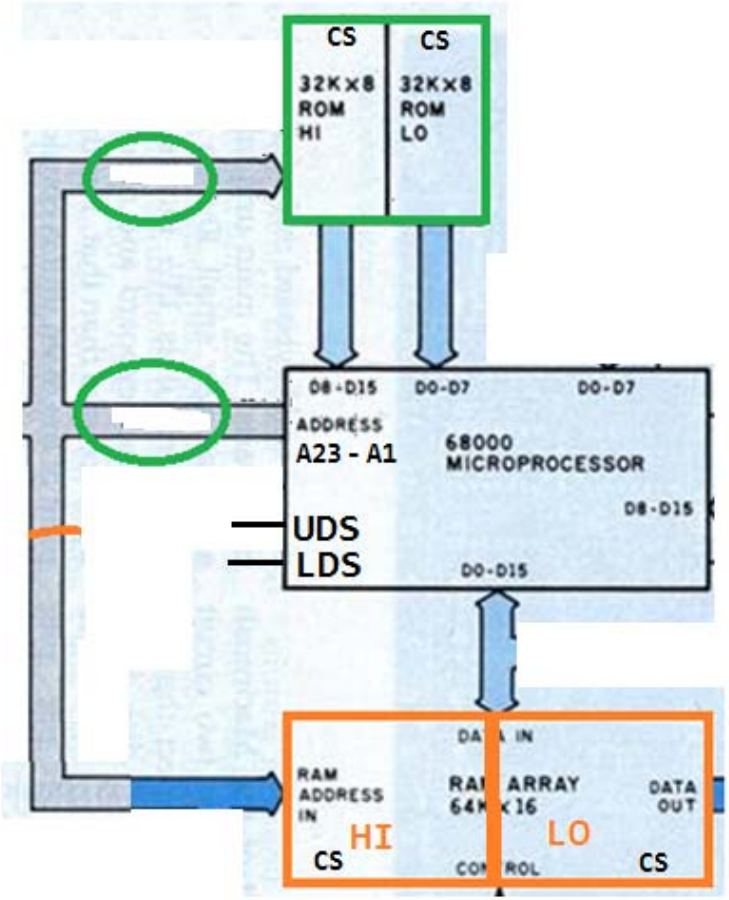
Apple Macintosh MEM Decoding

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.

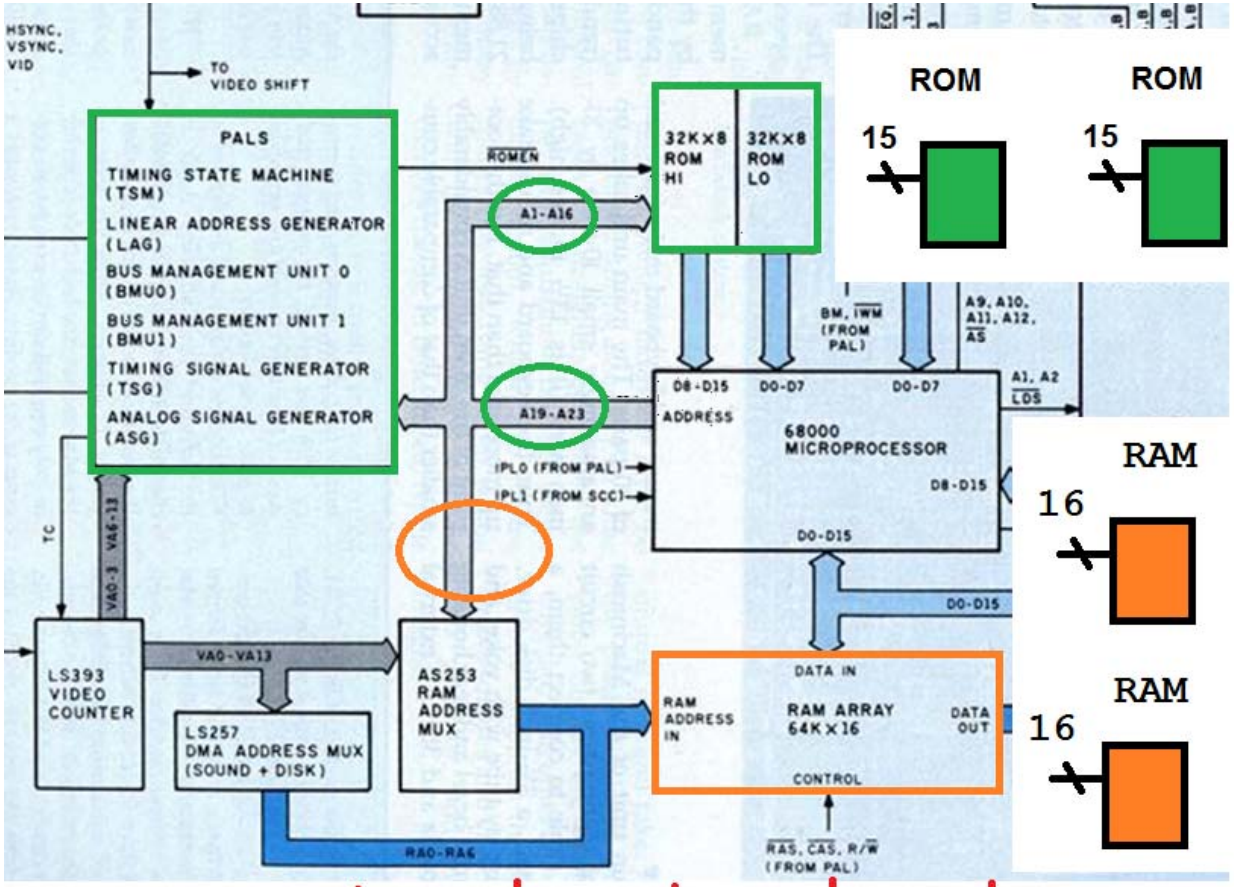


Design for Apple Macintosh MEM Decoding



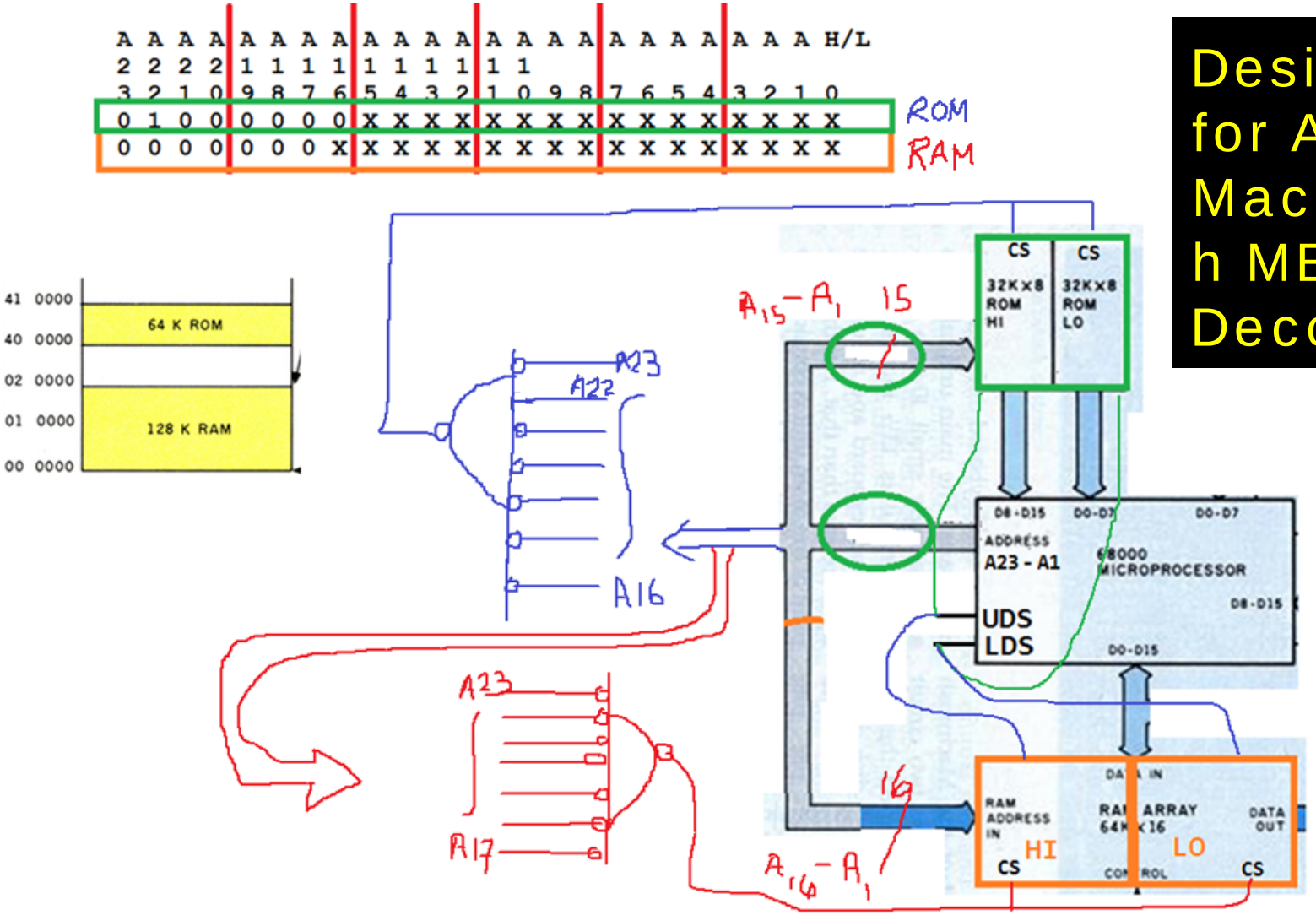


Apple Macintosh MEM Decoding



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	1	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Design for Apple Macintosh MEM Decoding



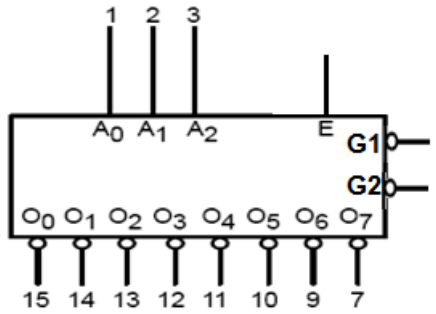
Decoder/Multiplexer

- 74LS138

MOTOROLA

1-OF-8 DECODER/ DEMULTIPLEXER

LOGIC SYMBOL



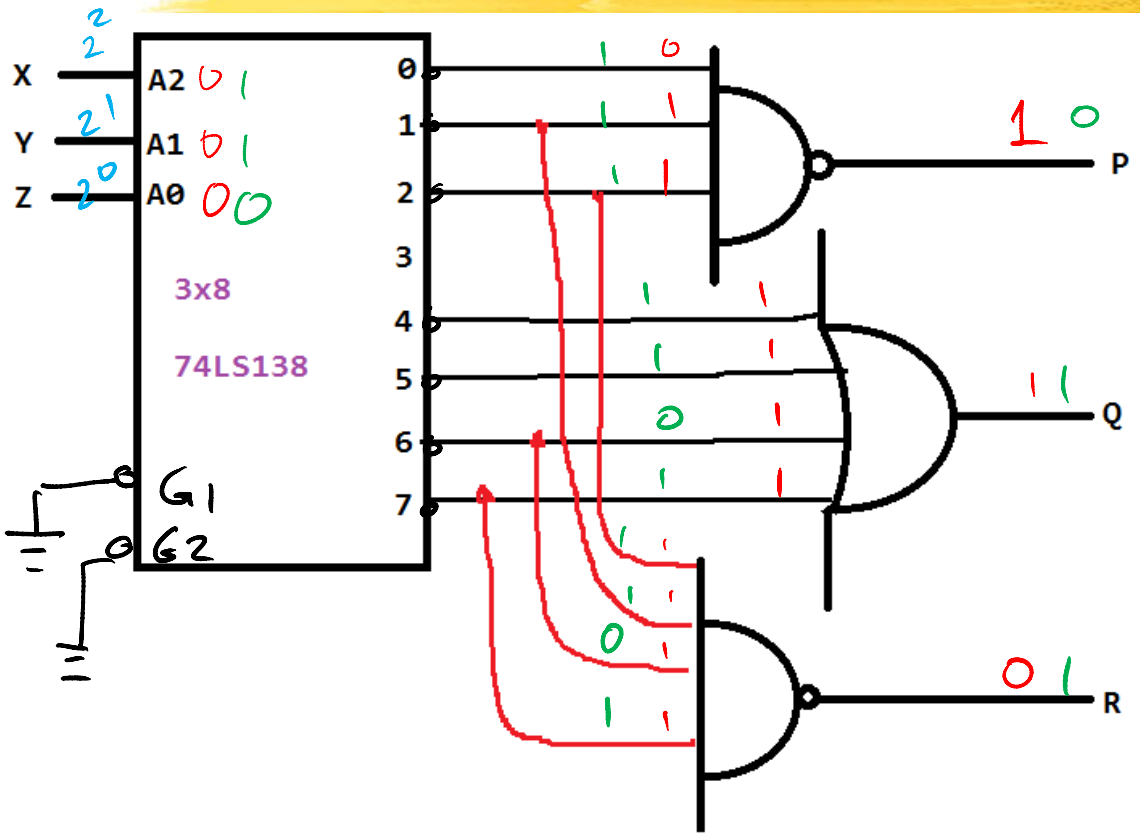
For the chip to work:

E: High
G1: Low
G2: Low

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

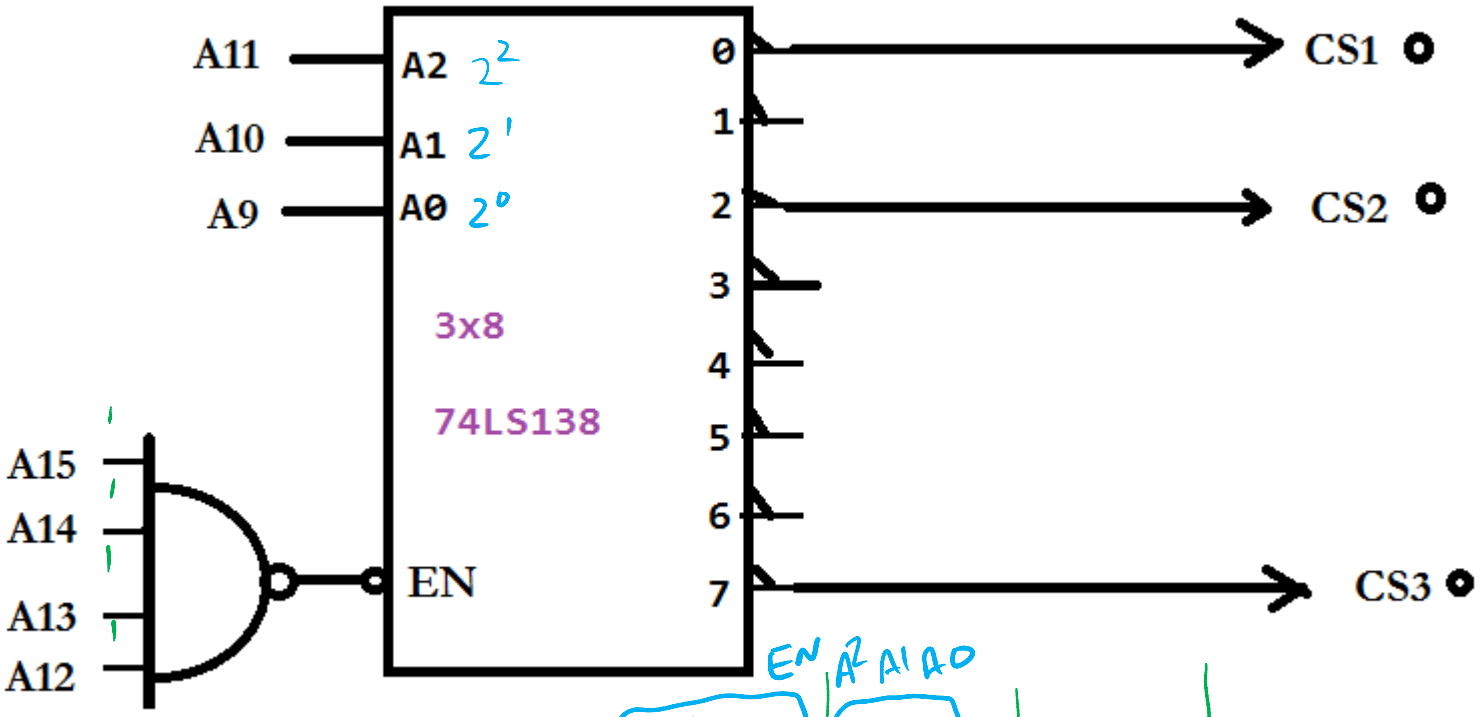
Decoder/Multiplexer



X Y Z	P	Q	R
0 0 0	1	0	0
0 0 1			
0 1 0			
0 1 1			
1 0 0			
1 0 1			
1 1 0	0	1	1

Decoder/Multiplexer

⌘ 74LS138



EN A2 A1 A0

A																MEM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	(F000
1	1	1	1	0	0	0	x	x	x	x	x	x	x	x	x	– FIFF)

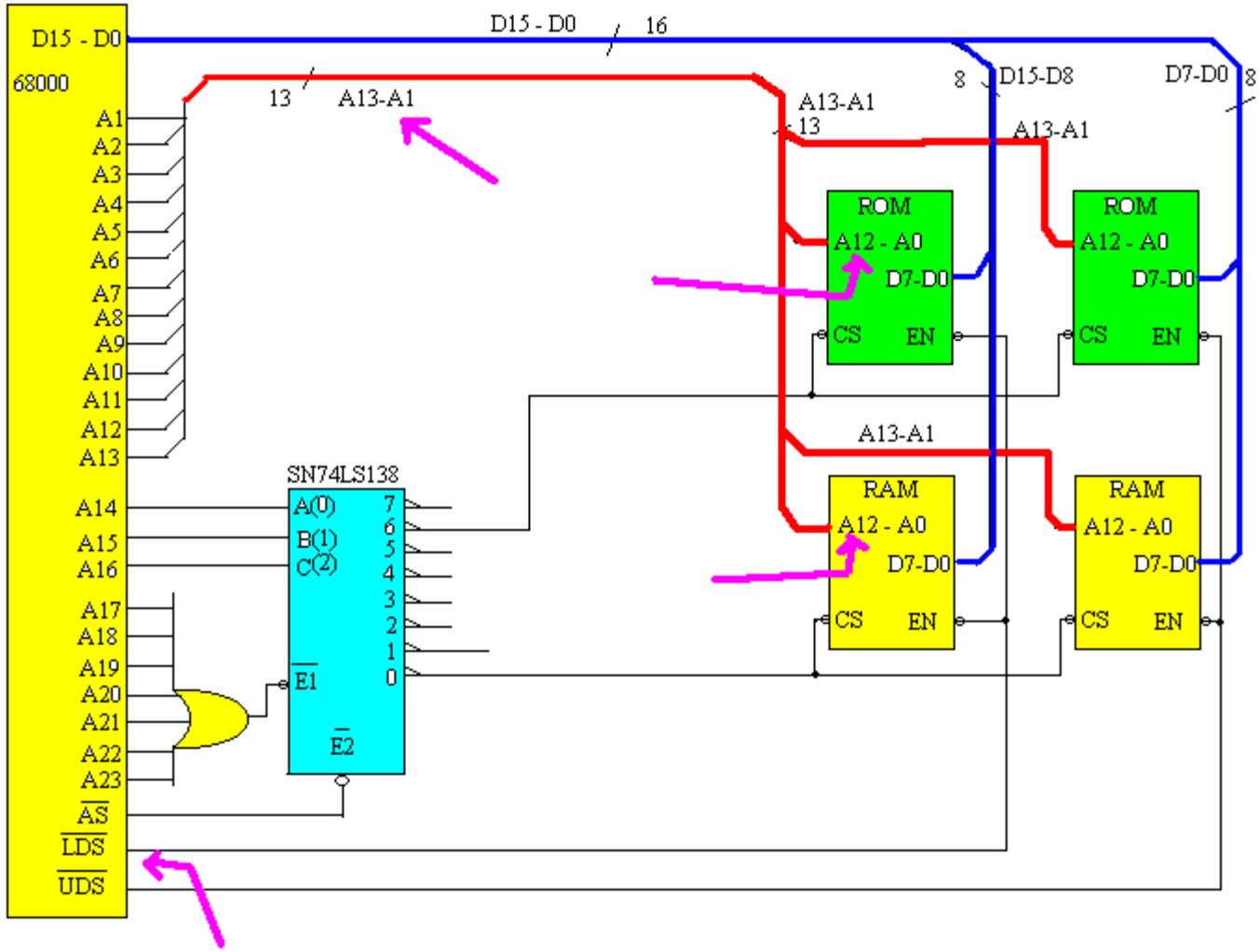
CS1 :

CS2 :

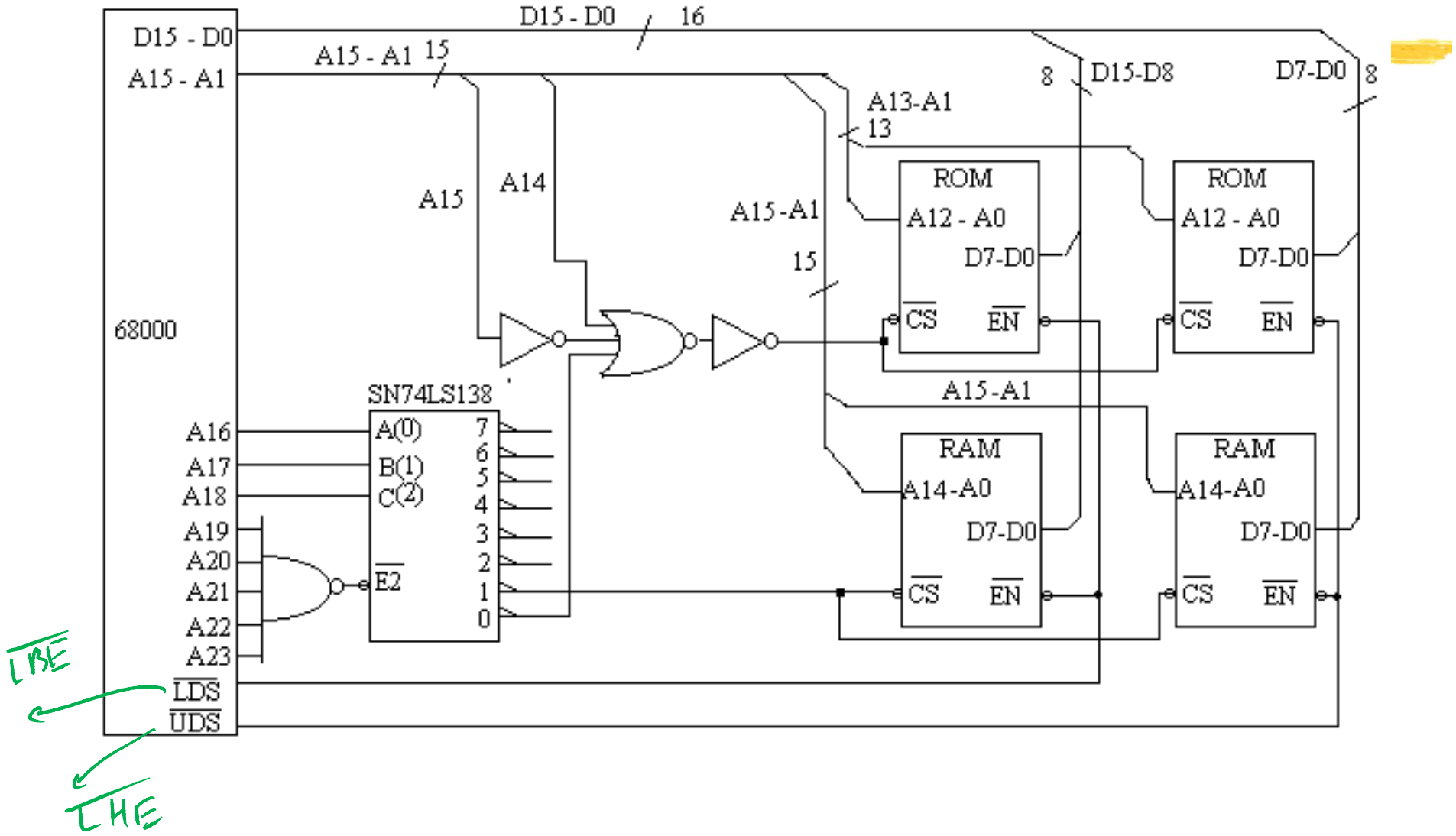
CS3 :

Memory Decoding with Byte/Word Access -

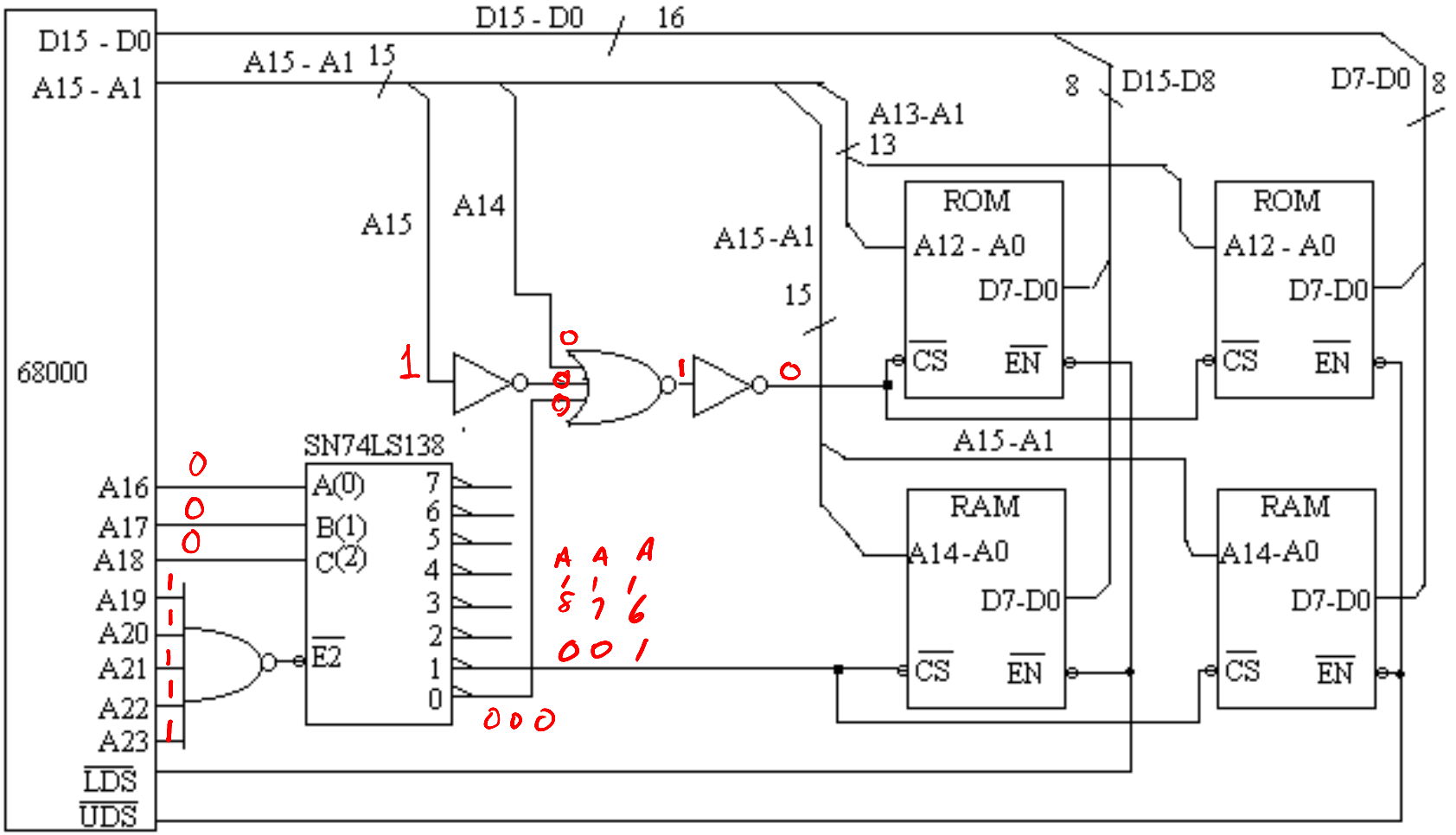
- Questions :
  - 1. Size of ROM
  - 2. Size of RAM
  - 3. Memory Map



Can You Draw a Memory Map of this?

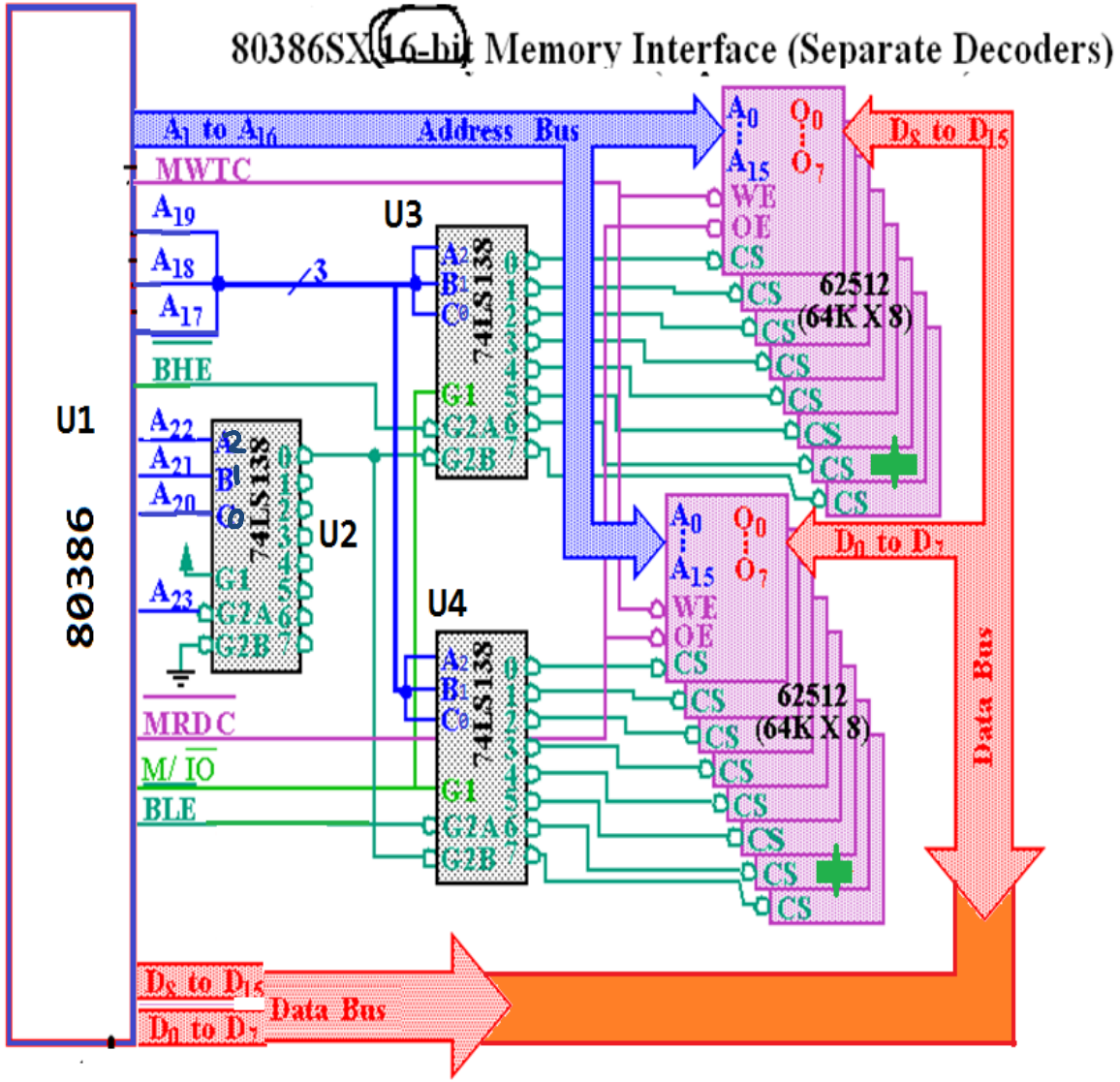


Can You Draw a Memory Map of this? - SOLUTION



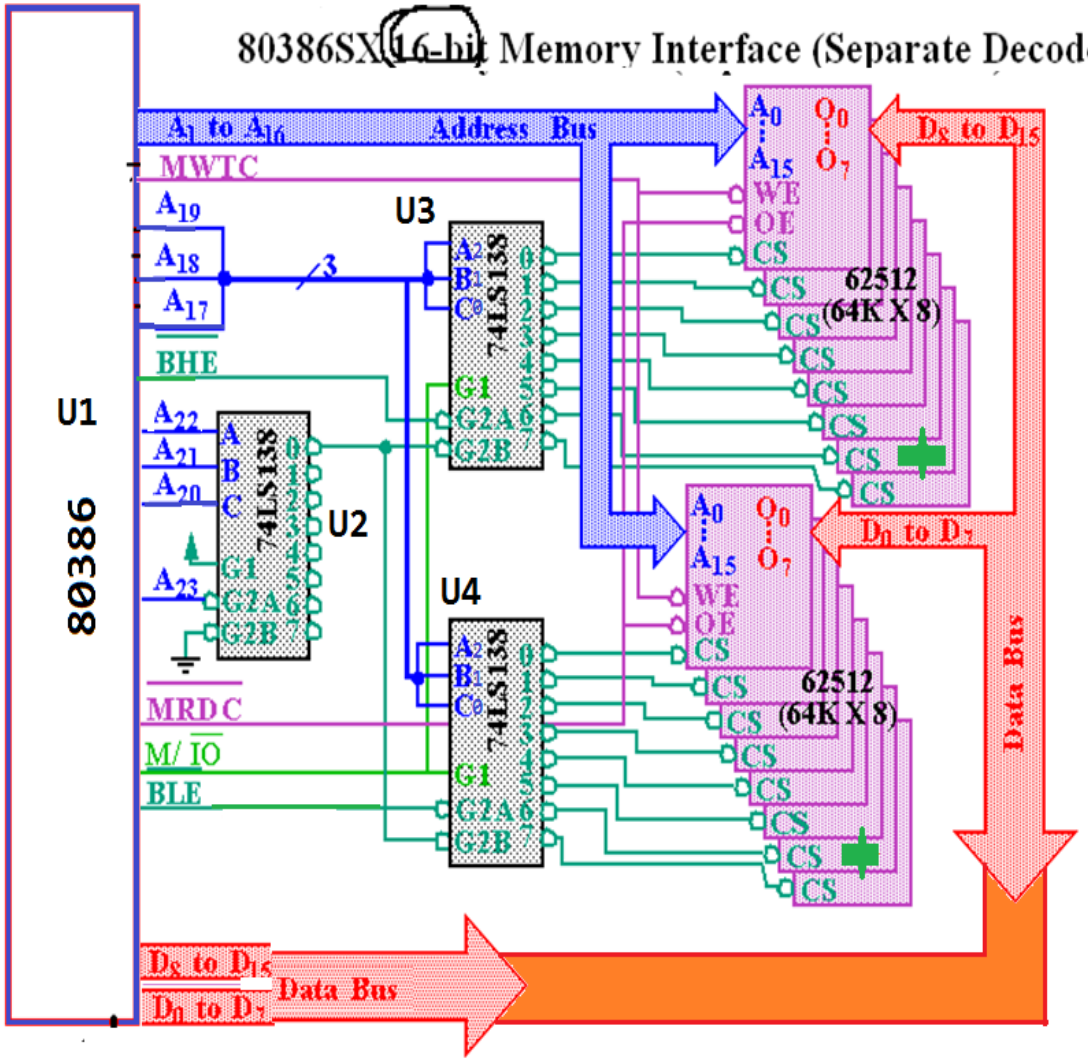
Intel 80386 Memory – Decoding

- 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)

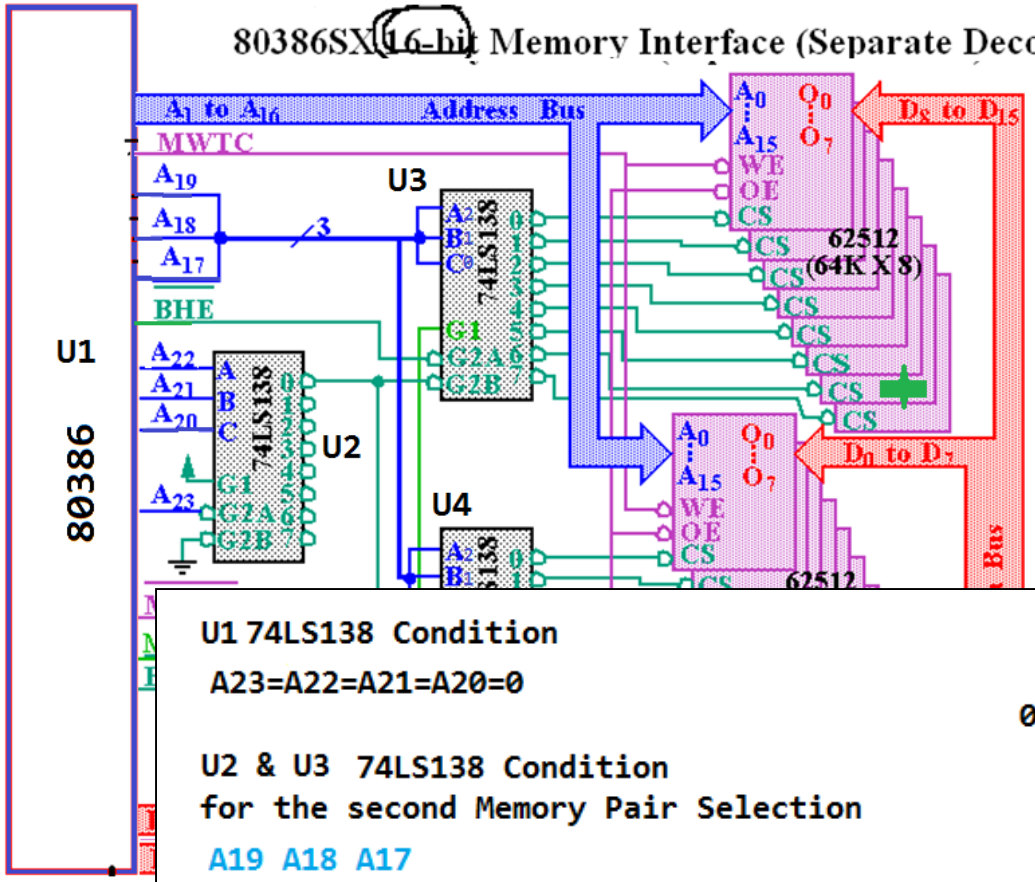


Intel 80386 Memory – Decoding-

- Find the address range of the **second** chips of the memory pair. (Green Mark)



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

0000	110X	XXXX	XXXX	XXXX	XXXX
0	C	0	0	0	0
0	D	F	F	F	F