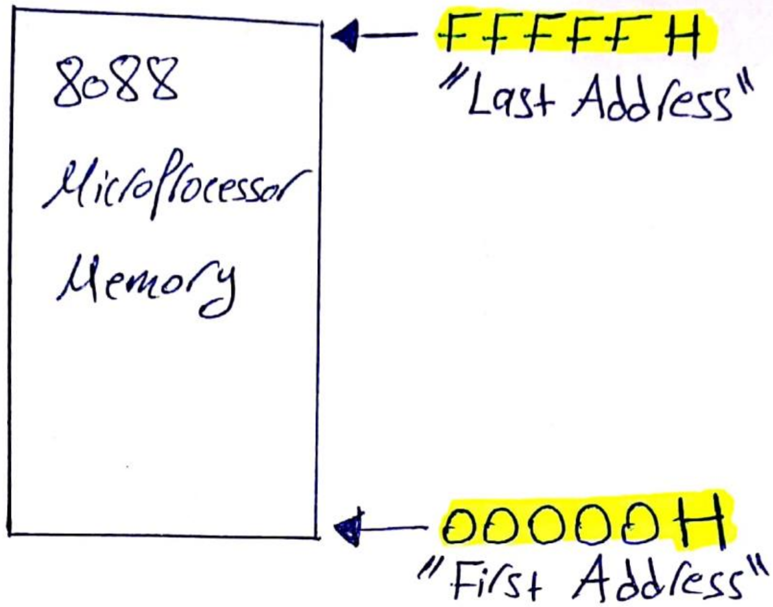
8088 MicroProcessor Memory Size = 1 MB

$\boxed{1\,MB = 2^{20}}$ → That is all the Memory in 8088



A. 256 KB RAM (Read & Write)

Consists of 128 KB and 32 KB

⇒ 256 KB = 1·(128 KB RAM) + 4·(32 KB RAM)

⇒ So we need 5 chips
- 4 chips of 32 KB
- 1 chip of 128 KB

256 KB RAM

Inside The 8088 Memory

From 00000H ↓ 3FFFFH

32 KB IC	38000H – 3FFFFH
32 KB IC	30000H – 37FFFH
32 KB IC	28000H – 2FFFFH
32 KB IC	20000H – 27FFFH
128 KB IC	00000H – 1FFFFH

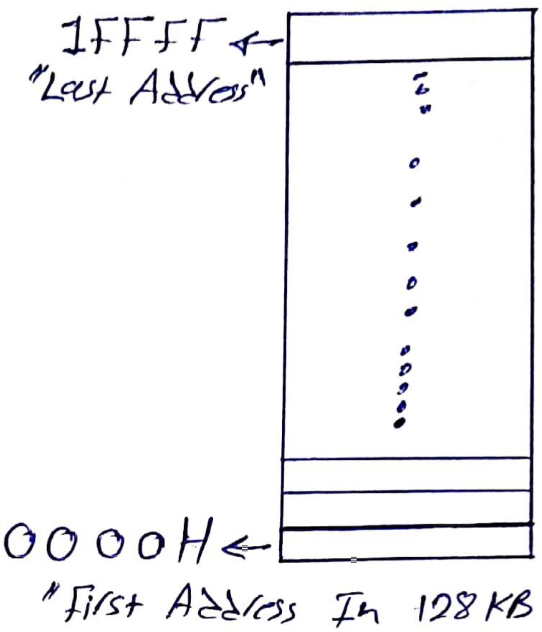
(*) Chip #1 : 128 KB

$128\ KB = 2^{17}$

Which Means There is 131072 different Addresses To Write or Read On it.

Start: 0 0000 0000 0000 0000

End: $\underbrace{1}_{1}\ \underbrace{1111}_{F}\ \underbrace{1111}_{F}\ \underbrace{1111}_{F}\ \underbrace{1111}_{F}$



We will use Address Pins $\boxed{A_0 \rightarrow A_{16}}$ to locate the Address of the Data we need to write or read with The Memory.

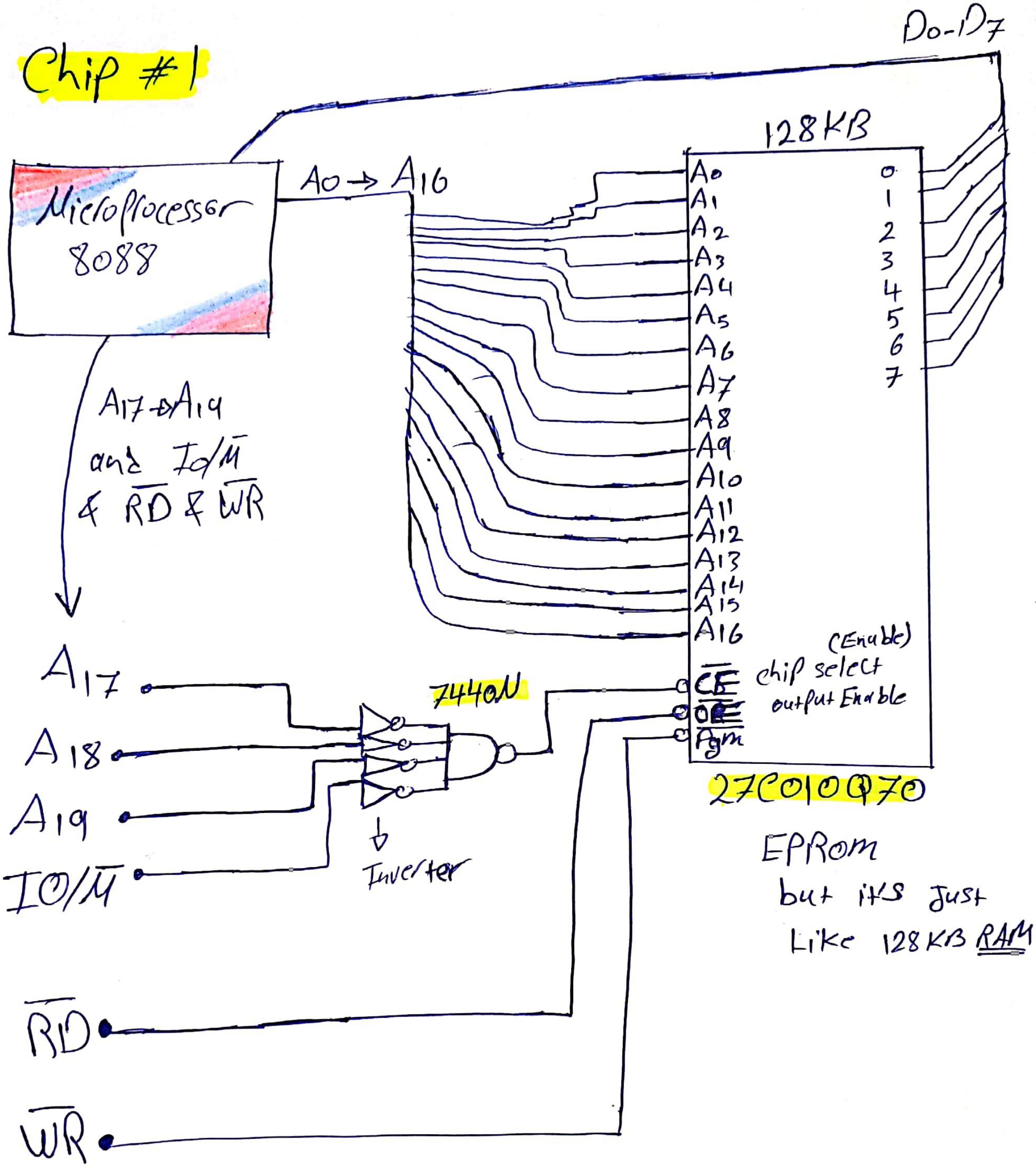
We will use Address Pins $\boxed{A_{17} \rightarrow A_{19}}$ to decode The Address of the chip we need to use.

	A_{19} A_{18} A_{17}	A_{16} ... A_0
Starting Address :	0 0 0	0 0000 0000 0000 0000
Ending Address :	0 0 0	1 1111 1111 1111 1111

(*) NAND-Inputs $= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}}$

(*) Enabling Signals : $IO/\overline{M}$ and $\overline{RD}$ and $\overline{WR}$

Chip #1



Finish chip #1

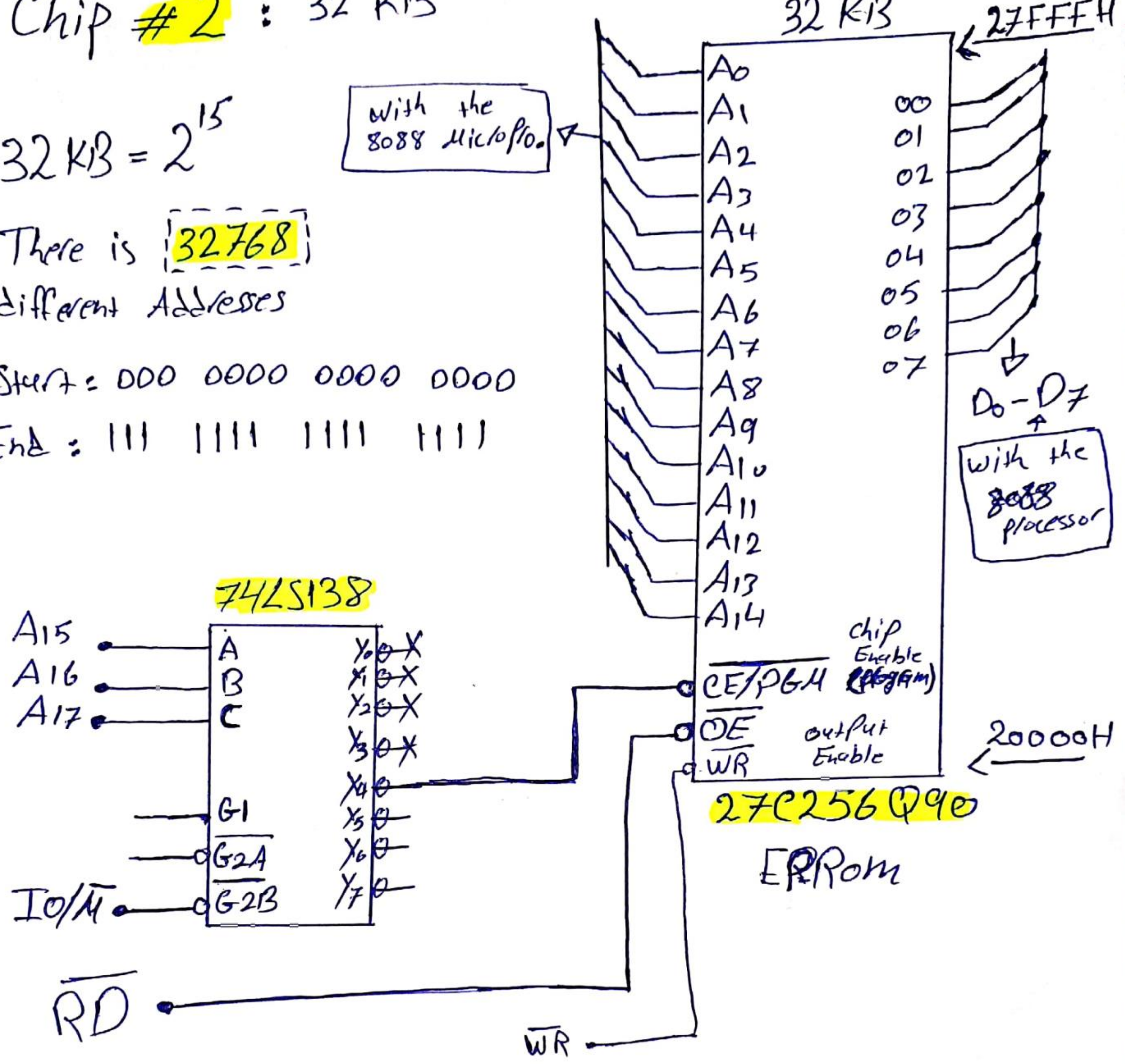
Chip #2 : 32 KB

$32\,KB = 2^{15}$

There is 32768 different Addresses

Start : 000 0000 0000 0000

End : 111 1111 1111 1111



~~Starting Address:~~

	A19 A18 A17 A16 A15	A14 … A0
Starting Address :	0 0 1 0 0	000 0000 0000 0000
Ending Address :	0 0 1 0 0	111 1111 1111 1111
	A19 A18 A17 A16 A15	

We will use A17 & A16 & A15

With decoder 74LS138 to Decode the Address

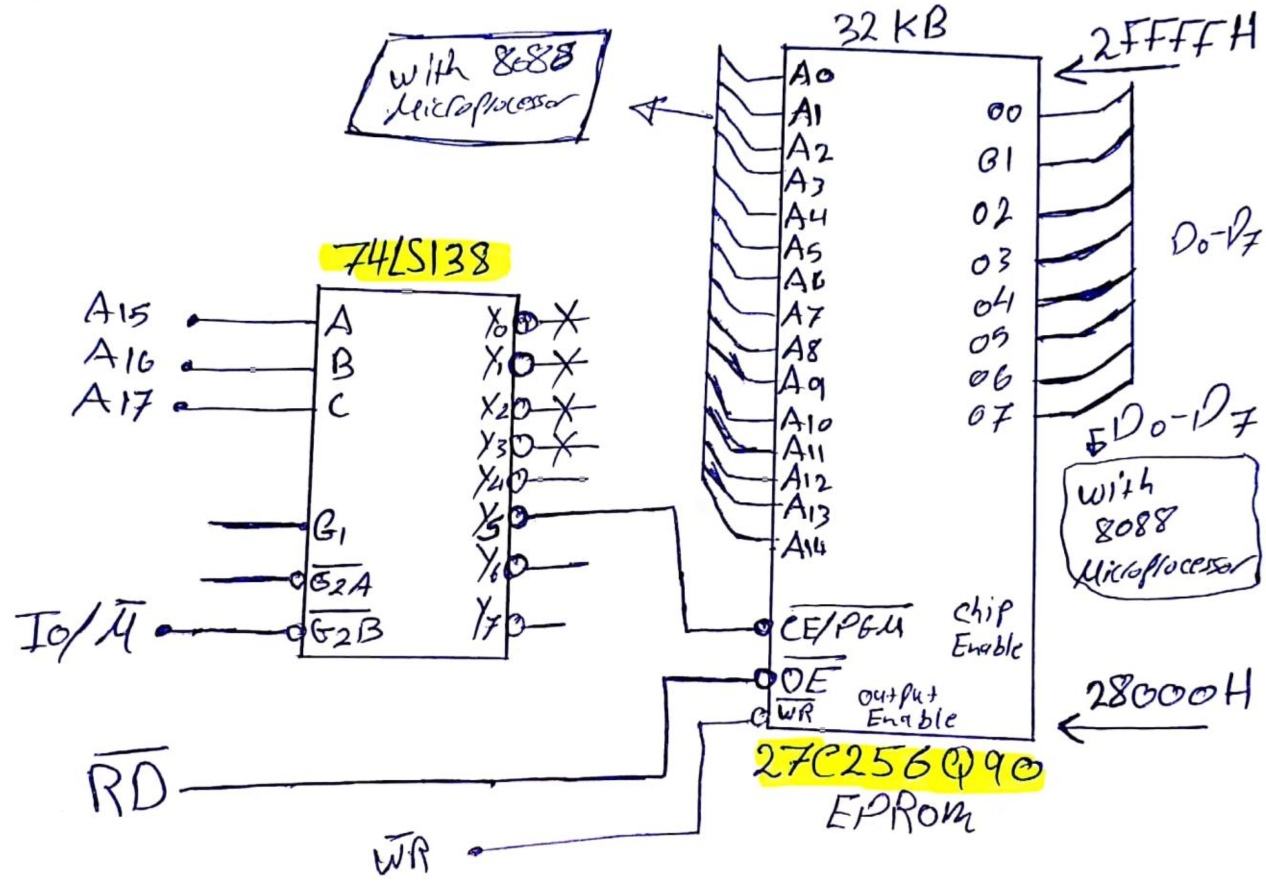
Chip #3 : 32 KB:

32 KB = 2^{15}

inside The memory chip — in 8088 whole memory

Starts : 000 0000 0000 0000 [28000H]

End : 111 1111 1111 1111 [2FFFFH]



	A19 A18	A17 A16 A15	A14 … A0
Starting Address :	00	101	000 0000 0000 0000
Ending Address :	00	101	111 1111 1111 1111

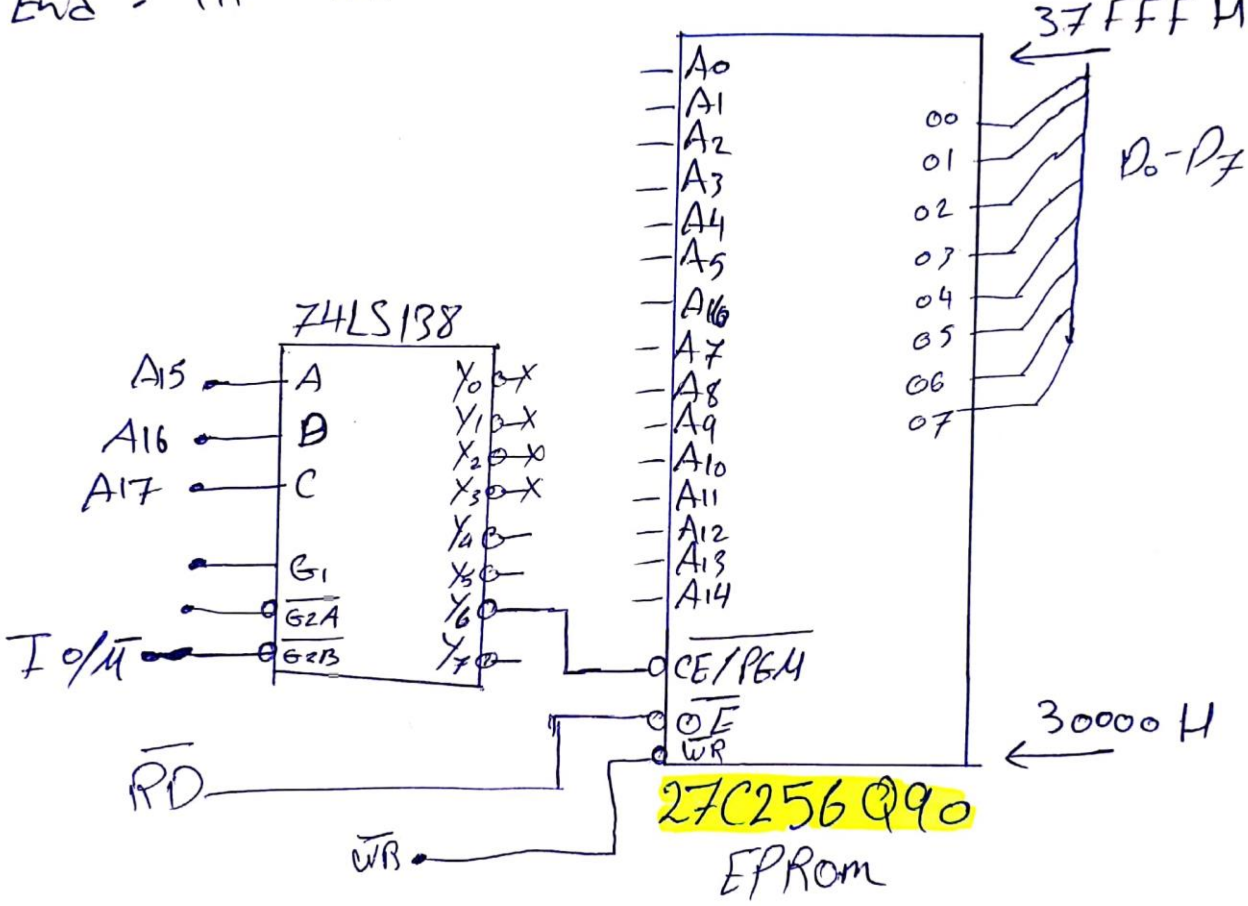
We will use A17, A16, A15 in the same 74LS138 decoder that we used in chip #2

Chip #4 : 32 KB :

32 KB = 2^{15}

Starts: 000 0000 0000 0000 [30000H]

End : 111 1111 1111 1111 [37FFFH]



	A19 A18 A17 A16 A15	A14 … A0
Starting Address :	0 0 1 1 0	000 0000 0000 0000
Ending Address :	0 0 1 1 0	111 1111 1111 1111

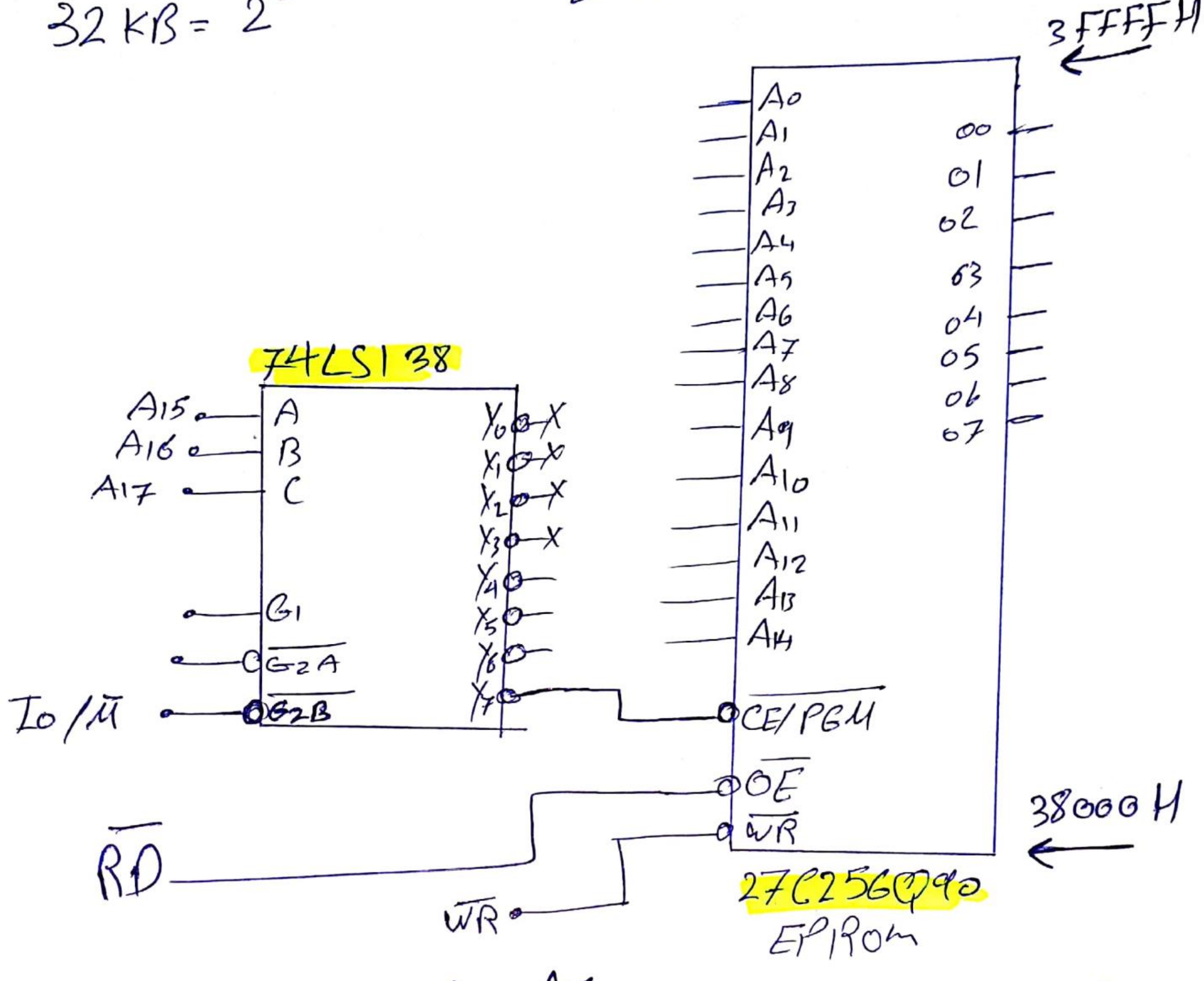
we will use the same 74LS138 decoder used in chip #2 & #3

Chip # 5: 32 KB

Starts [38000H]

Ends [3FFFFH]

$32\text{ KB} = 2^{15}$



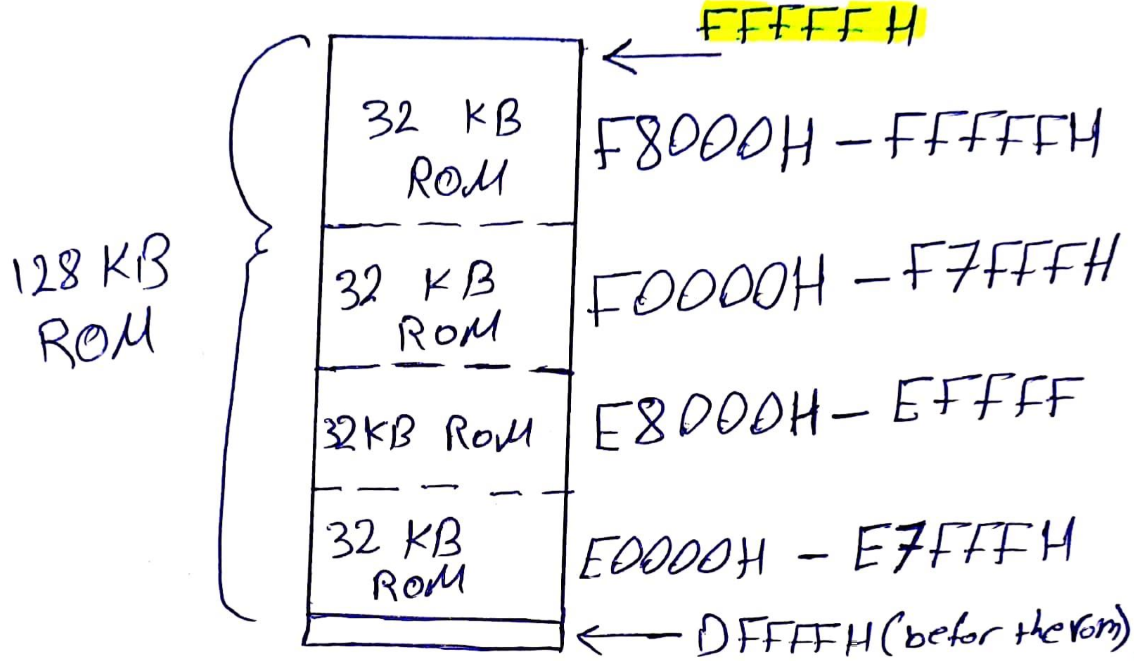
	A19 A15	A14 A0
Starting Address :	00111	000 0000 0000 0000
Ending Address :	00111	111 1111 1111 1111

We will use the same 74LS138 Decoder

(*) B. 128KB ROM (Read only)

Ends at FFFFFH

128KB ROM = 4.(32 KB RAM chip)



$(FFFFF)H = (1048575)_{10}$ / $128KB = 2^{17} = (131072)_{10}$

$(1048575)_{10} - (131072)_{10} = (917503)_{10} = (DFFFF)H$

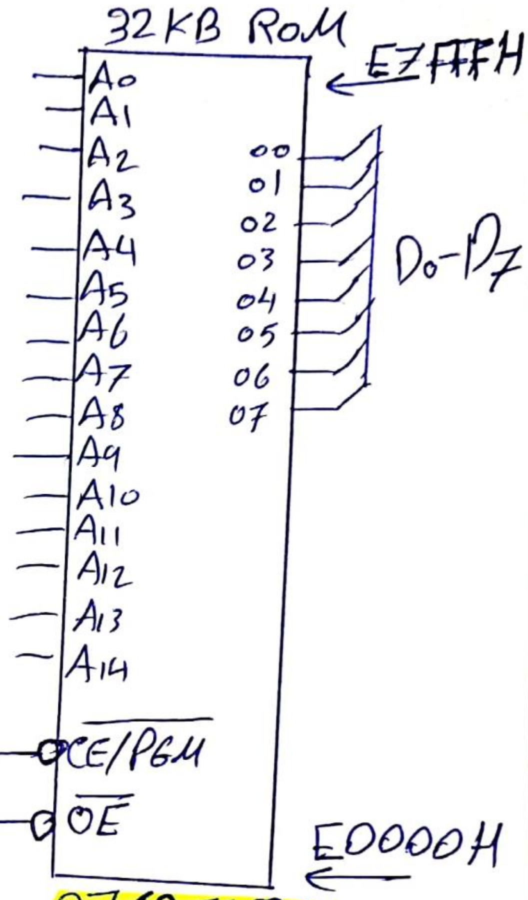
So : Starting Address of the 128 KB ROM (which means the first 32 KB Rom chip) is E0000H

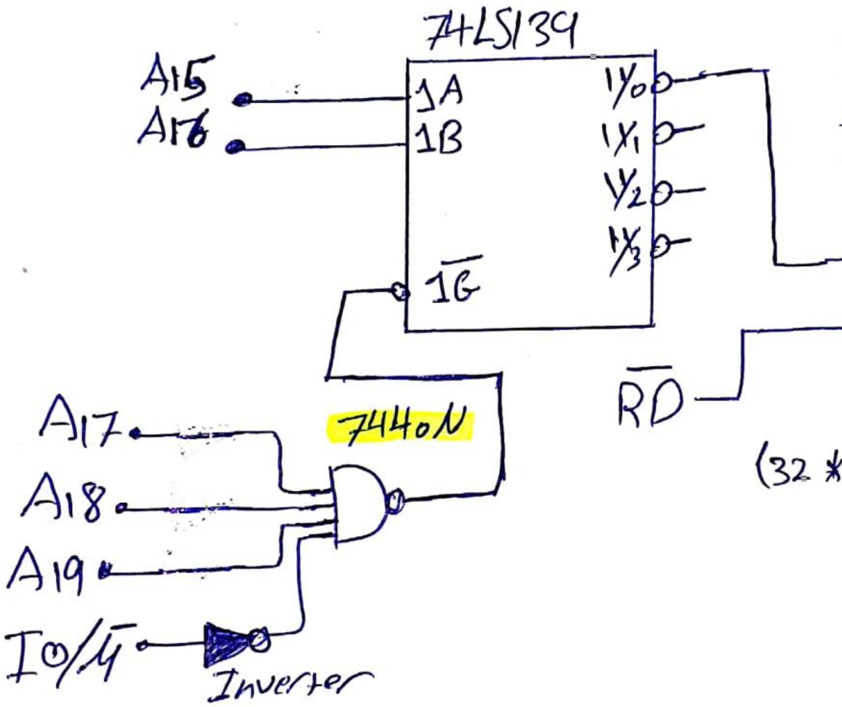
Chip #1: 32KB ROM

$32KB = 2^{15}$ ($A_0 - A_{14}$) Address

($A_{15} - A_{19}$ for decoding)

Chip #1 : E0000H – E7FFFH





27C256Q90

(32 * 8) 32 KB EPRom IC

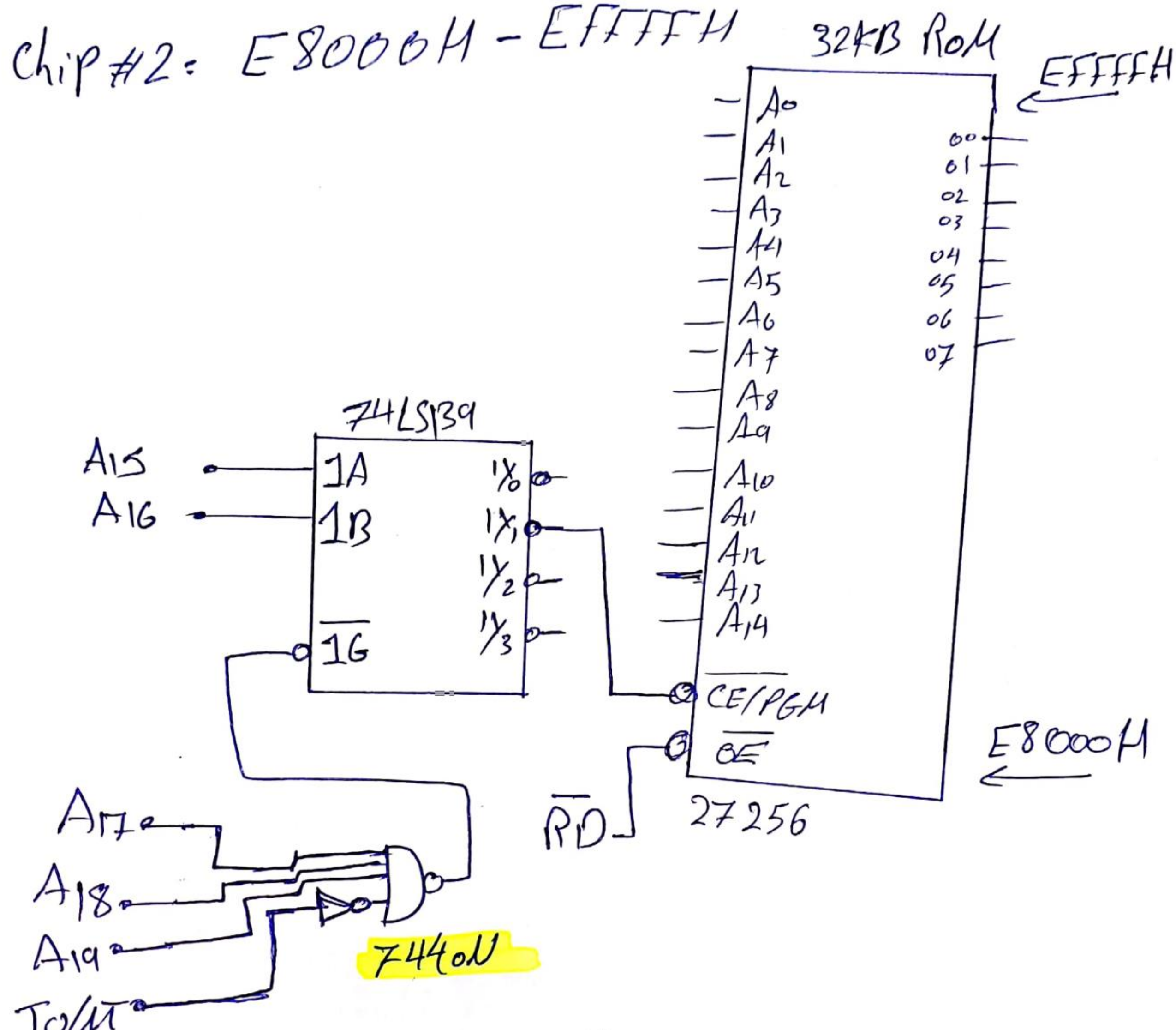
but we will not write on it

	A_{19} … A_{15}	A_{14} … A_0
Starting Address :	1 1 1 0 0	000 0000 0000 0000
Ending Address :	1 1 1 0 0	111 1111 1111 1111

we will use A16 , A1~~7~~5 (A_{15}, A_{16}) for Address decoding with 74LS139

Chip #2: 32KB Rom

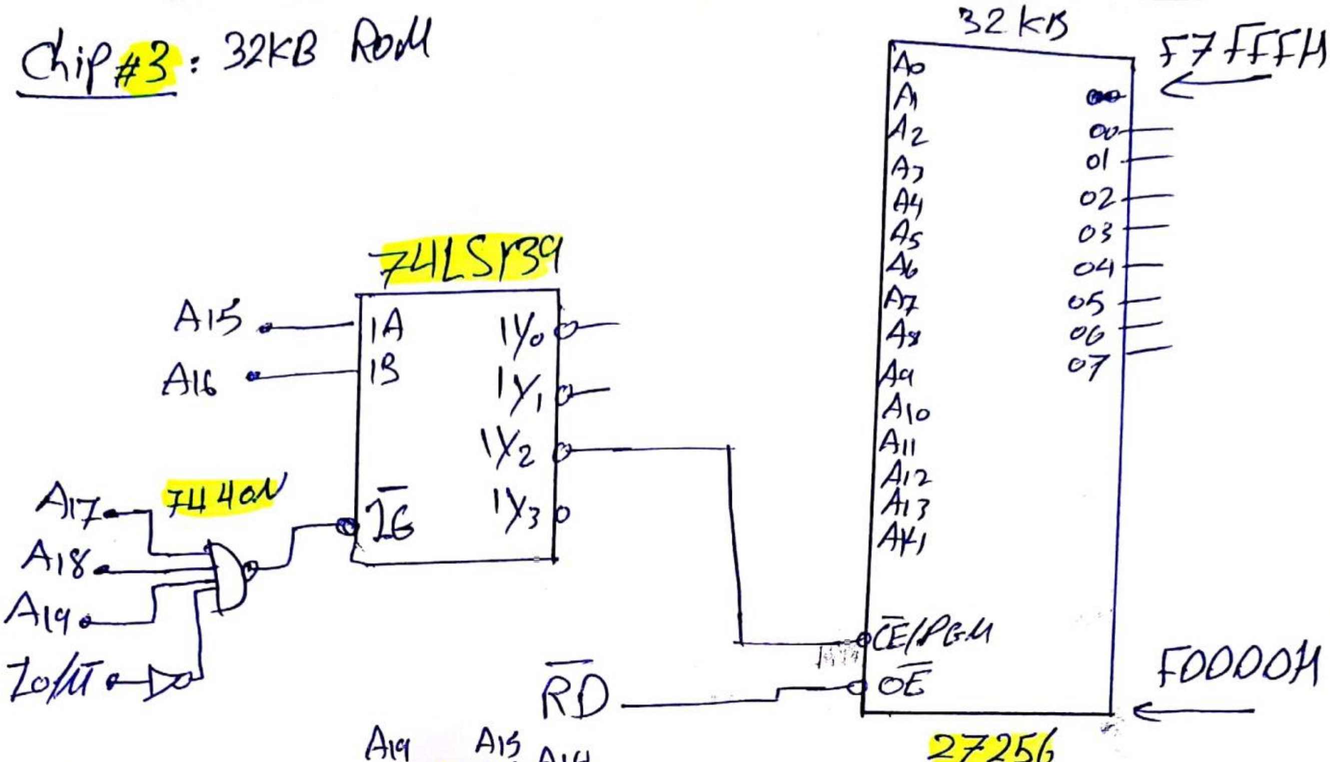
Chip #2: E8000H - EFFFFH



	A19 ... A15	A14 ... A0
Starting Address :	1110 1	000 0000 0000 0000
Ending Address :	1110 1	111 1111 1111 1111

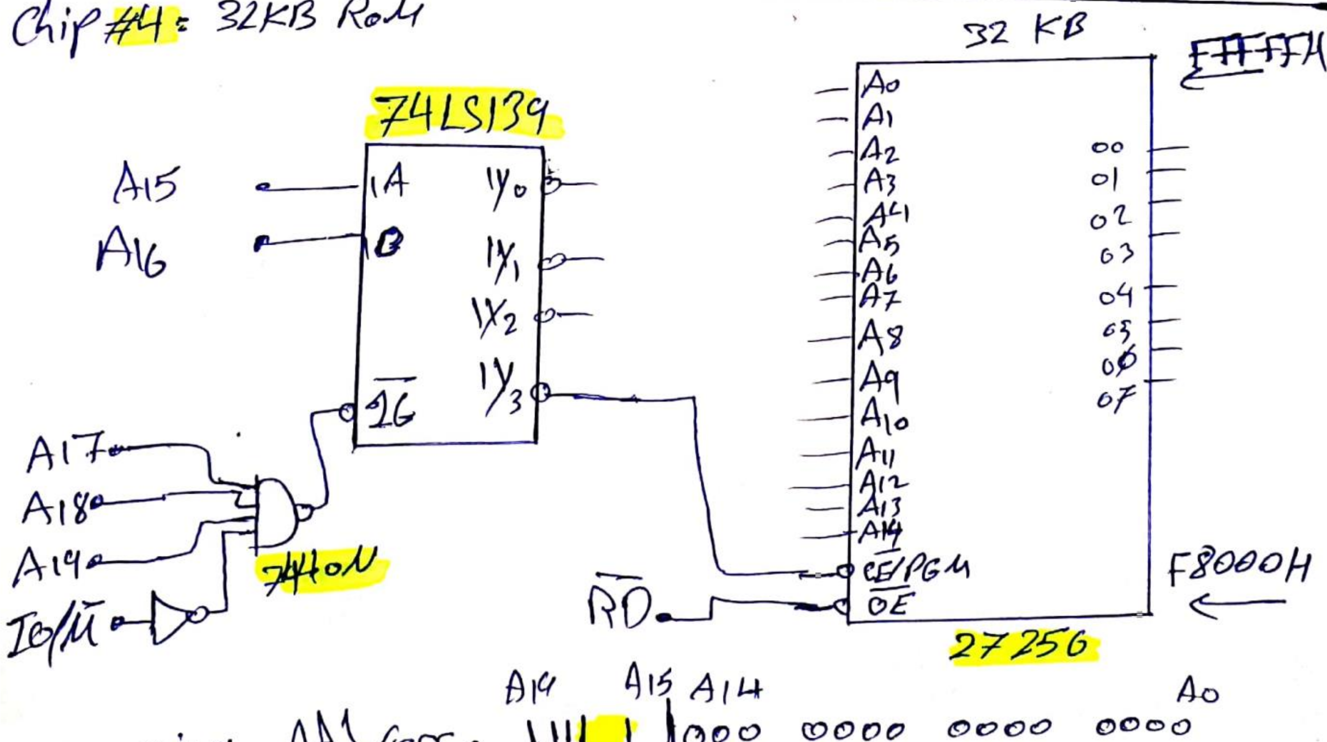
we'll use the same decoder we used in chip #1

Chip #3: 32KB Rom

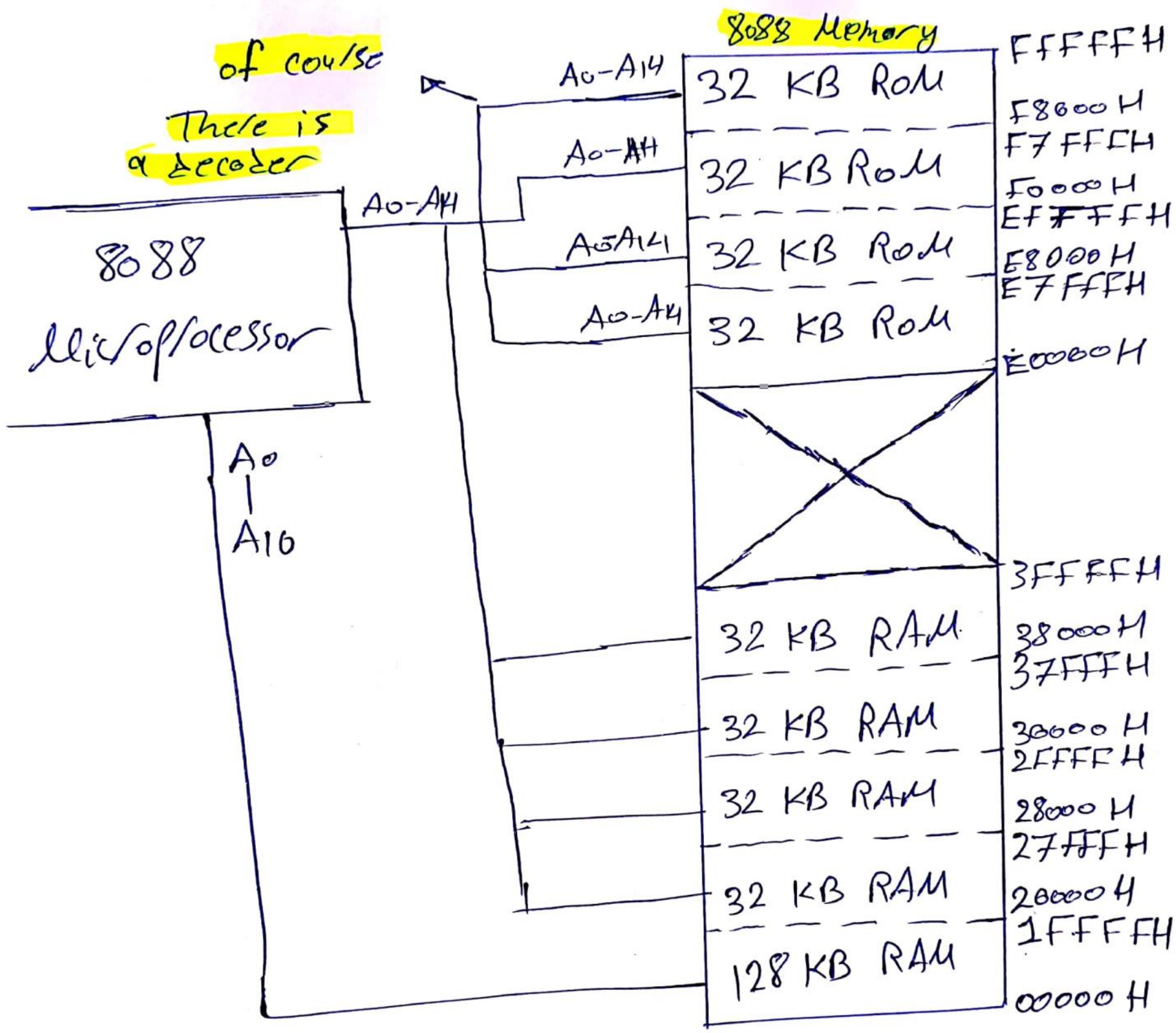


	A19 A15	A14 A0
Starting Address:	1111 0	000 0000 0000 0000
Ending Address:	1111 0	111 1111 1111 1111

Chip #4: 32KB Rom



	A19 A15	A14 A0
Starting Address:	1111 1	000 0000 0000 0000
Ending Address:	1111 1	111 1111 1111 1111


of course
There is
a decoder
8088 Memory
8088
Microprocessor
A0-A14
A0-A14
A0-A14
A0-A14
A0-A14
A0
A16
32 KB ROM
32 KB ROM
32 KB ROM
32 KB ROM
32 KB RAM
32 KB RAM
32 KB RAM
32 KB RAM
128 KB RAM
FFFFFH
F8000H
F7FFFH
F0000H
EFFFFH
E8000H
E7FFFH
E0000H
3FFFFH
38000H
37FFFH
30000H
2FFFFH
28000H
27FFFH
20000H
1FFFFH
00000H