رس ریزپردازنده
کتر سید امید فاطمی

Microprocessor System Design

Omid Fatemi

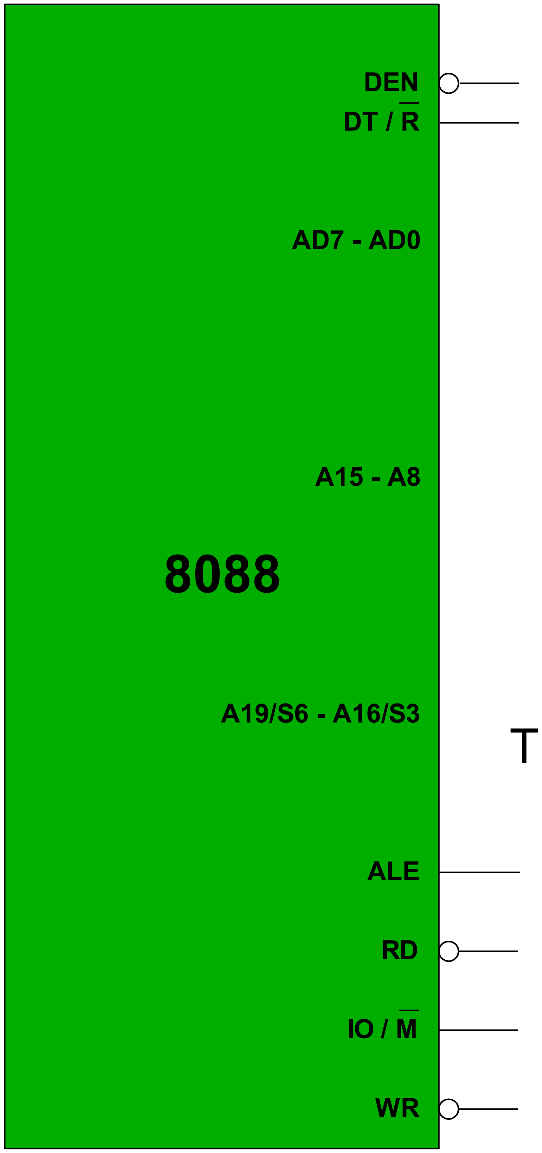
Memory Interfacing

(omid@fatemi.net)

Outline

- Memory module
- Interfacing to memory
- Address decoding
- Chip select
- Memory configurations

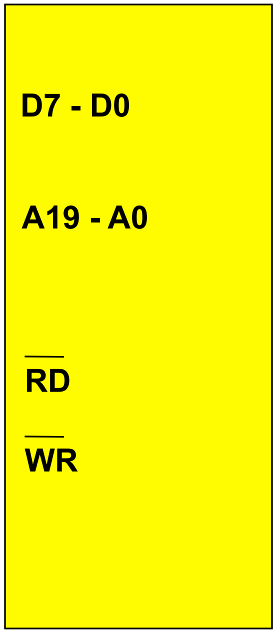
Minimum Mode



The first thing to provide?

A Memory Module

- The Size:
  - Data bus?
  - Address bus?
- Controls?
  - Read
  - Write



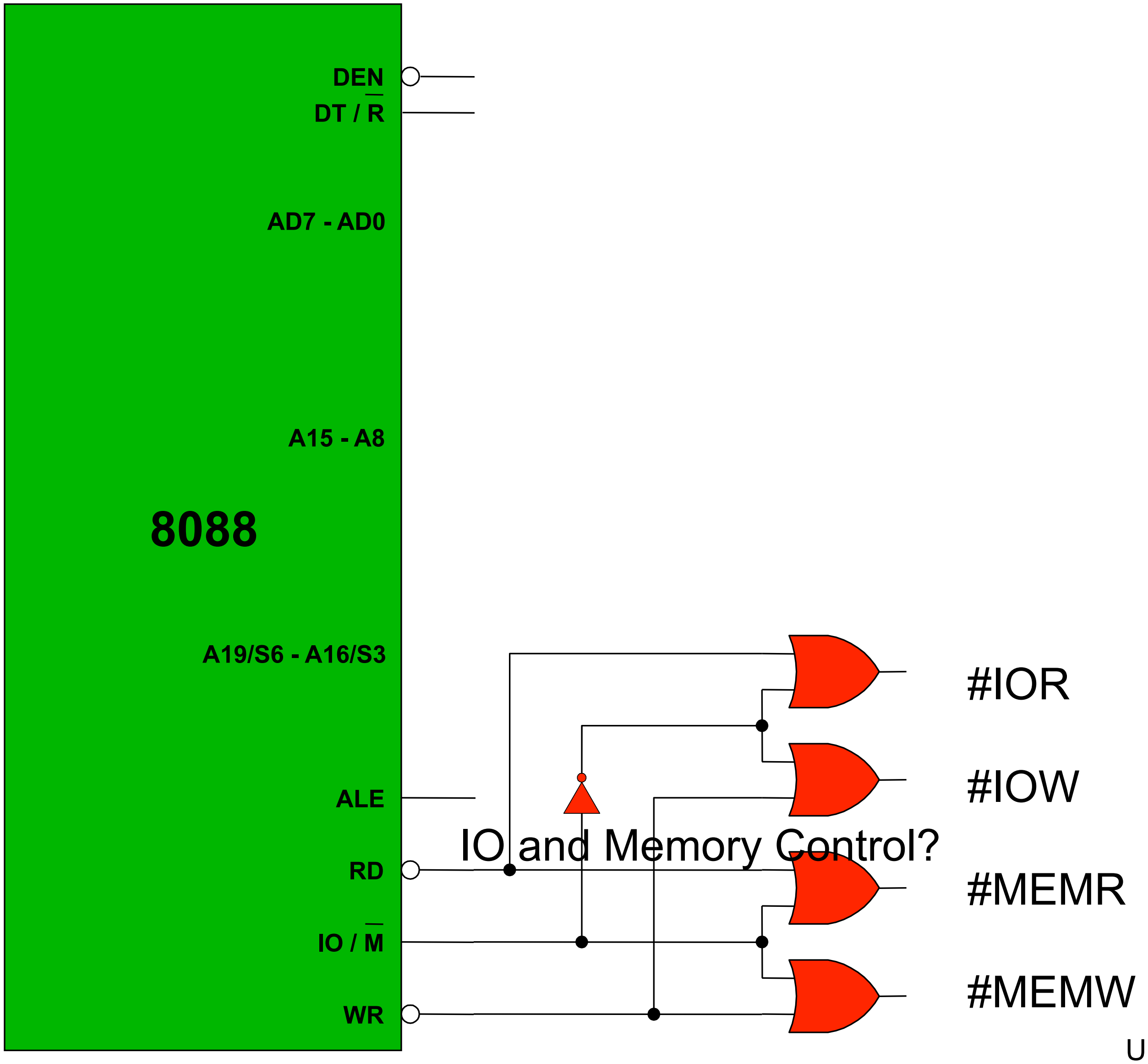
D7 - D0

A19 - A0

$\overline{RD}$

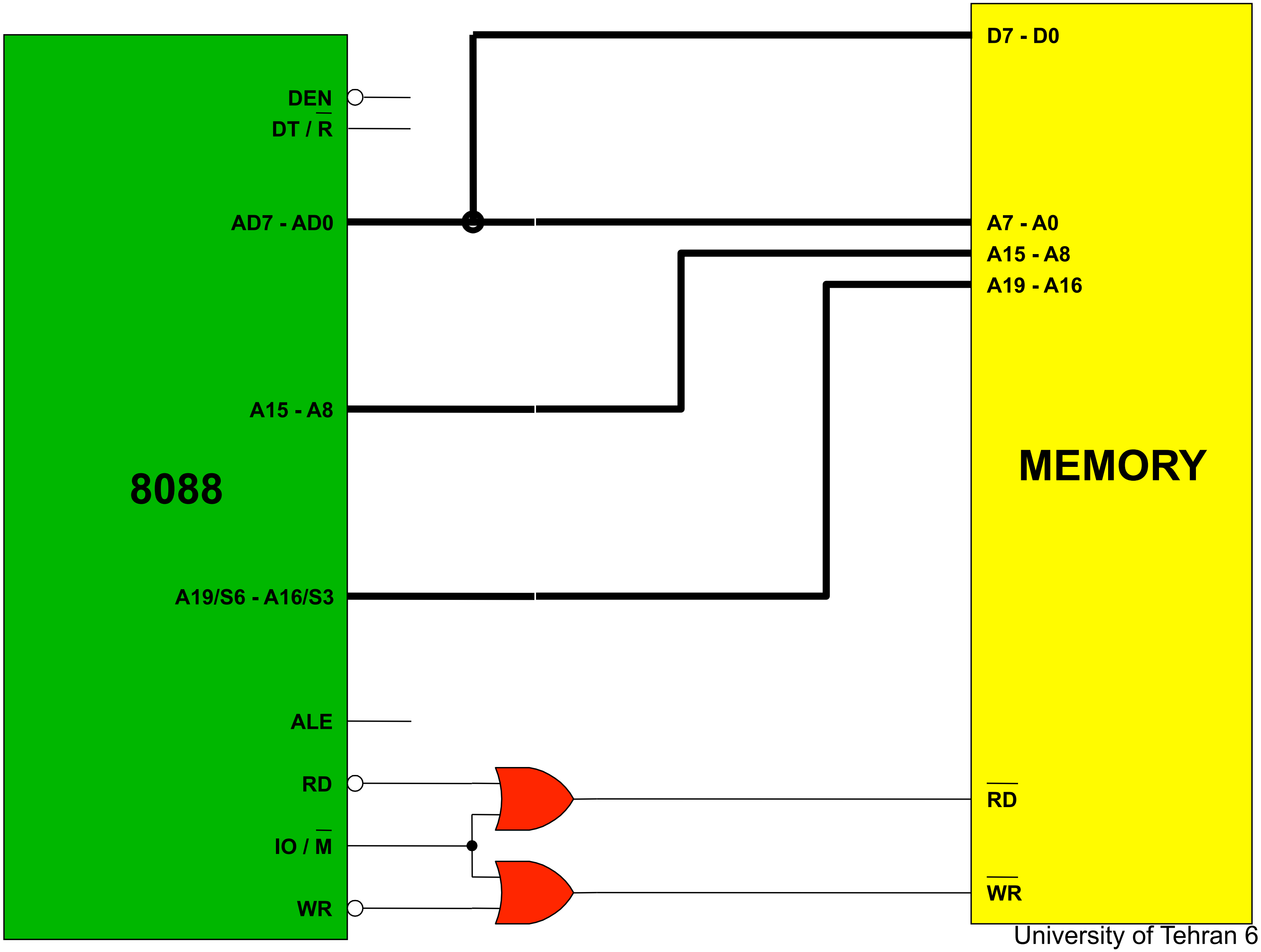
$\overline{WR}$

Minimum Mode

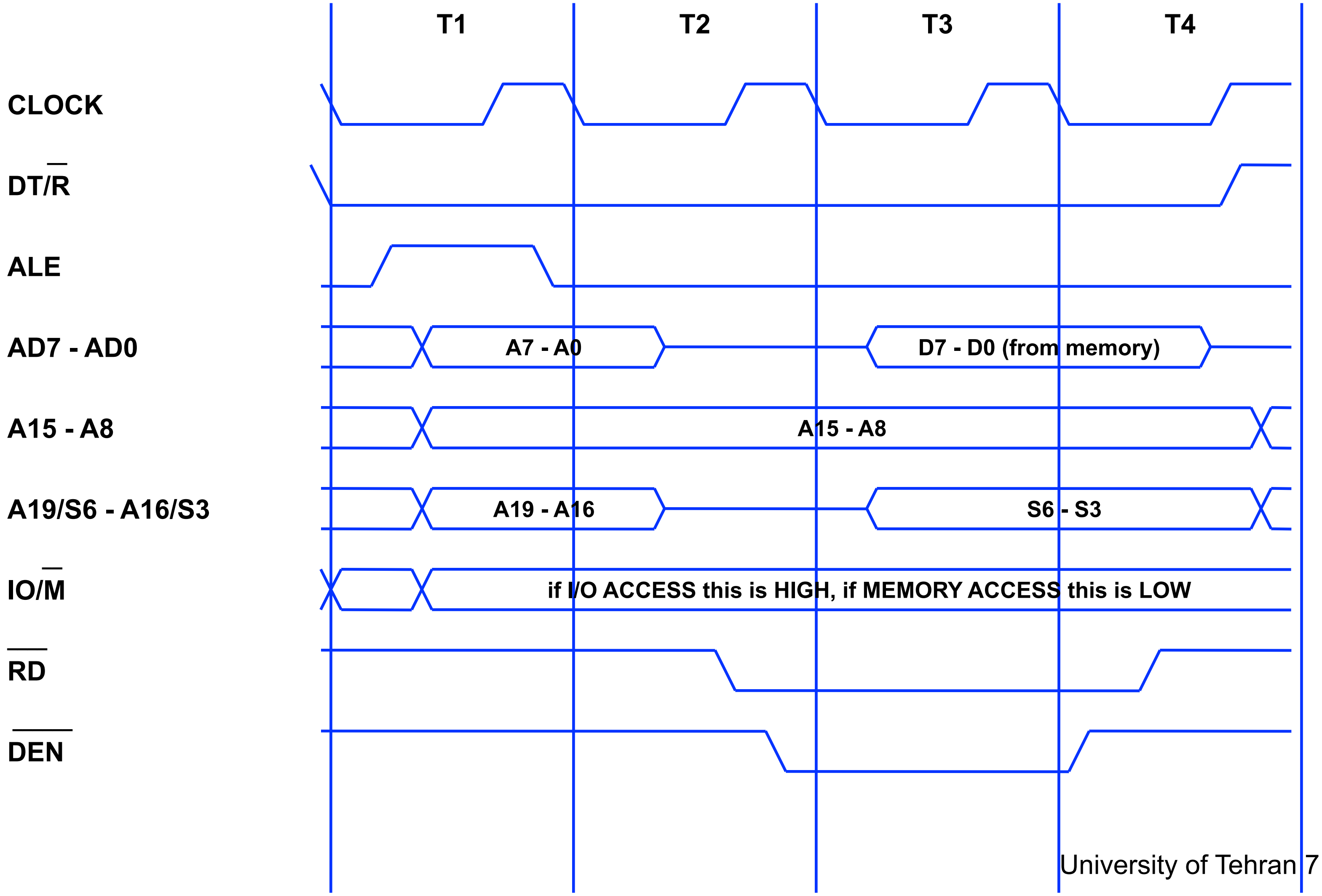


IO and Memory Control?

Minimum Mode



Processor Timing Diagram of 8088 (Minimum Mode) for Memory or I/O Read

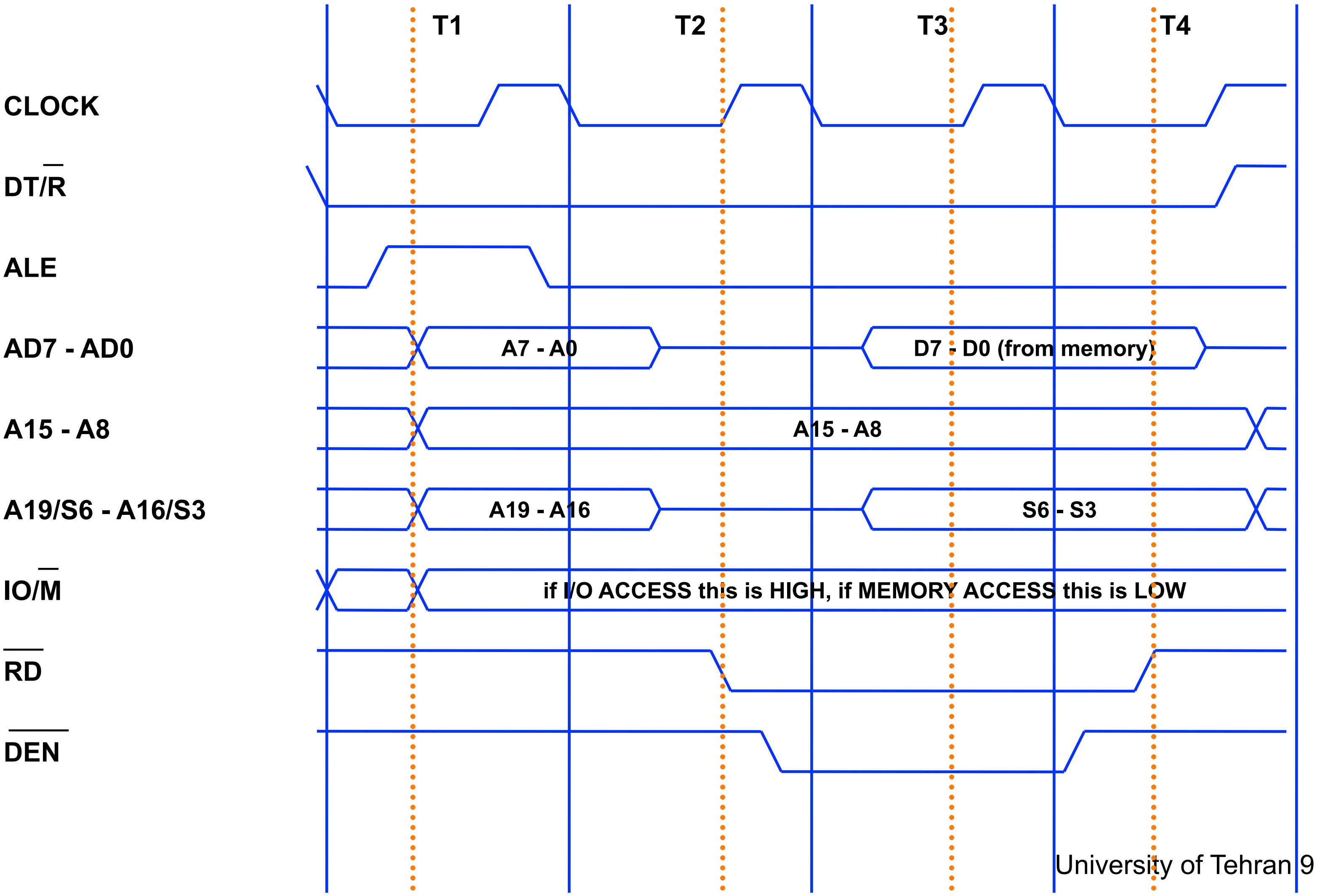


Will the circuit be able to perform memory read?

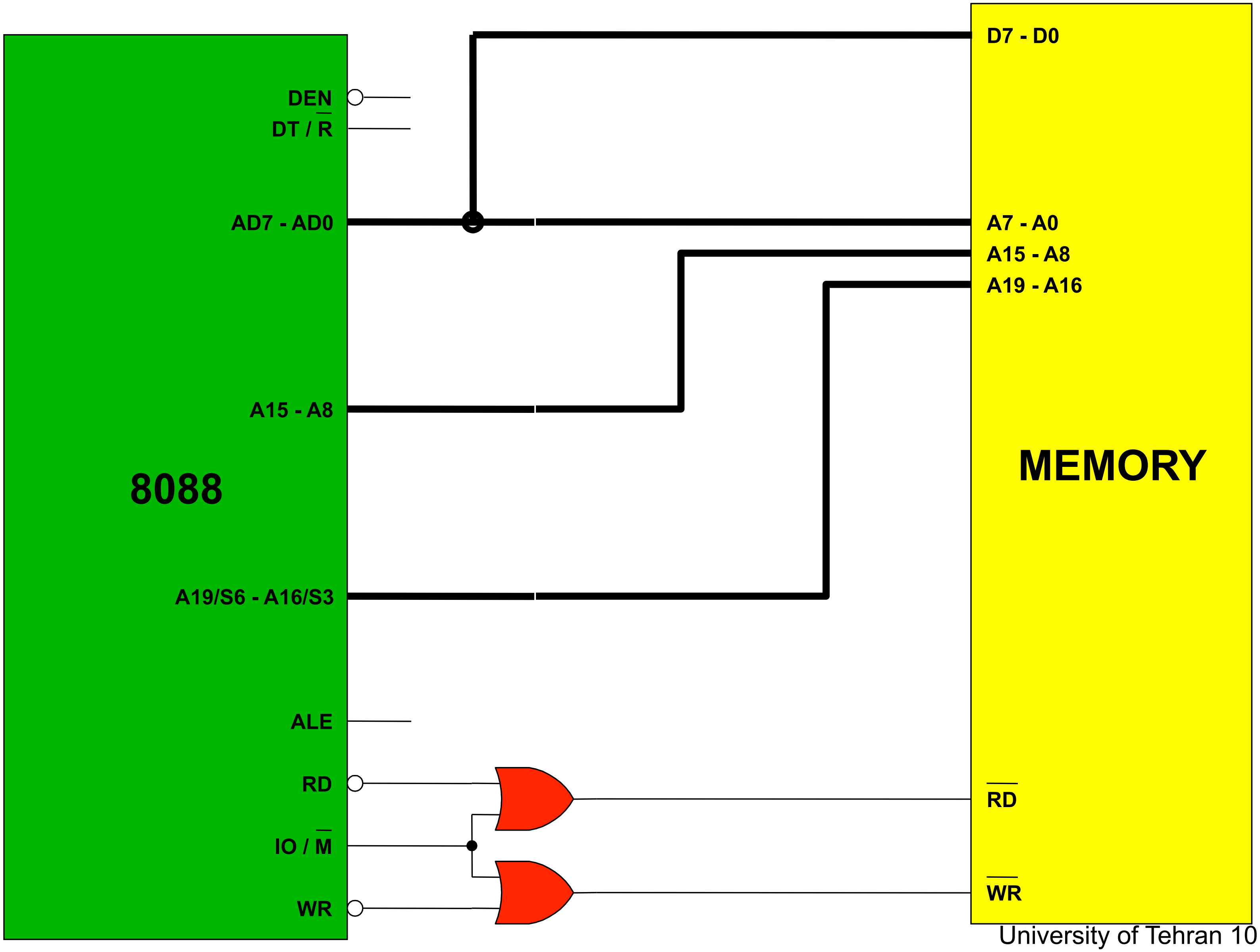
```
;assume that initially the values
;of the registers are:
;BX = 1234, DS = 9000

MOV AL, [BX]
```

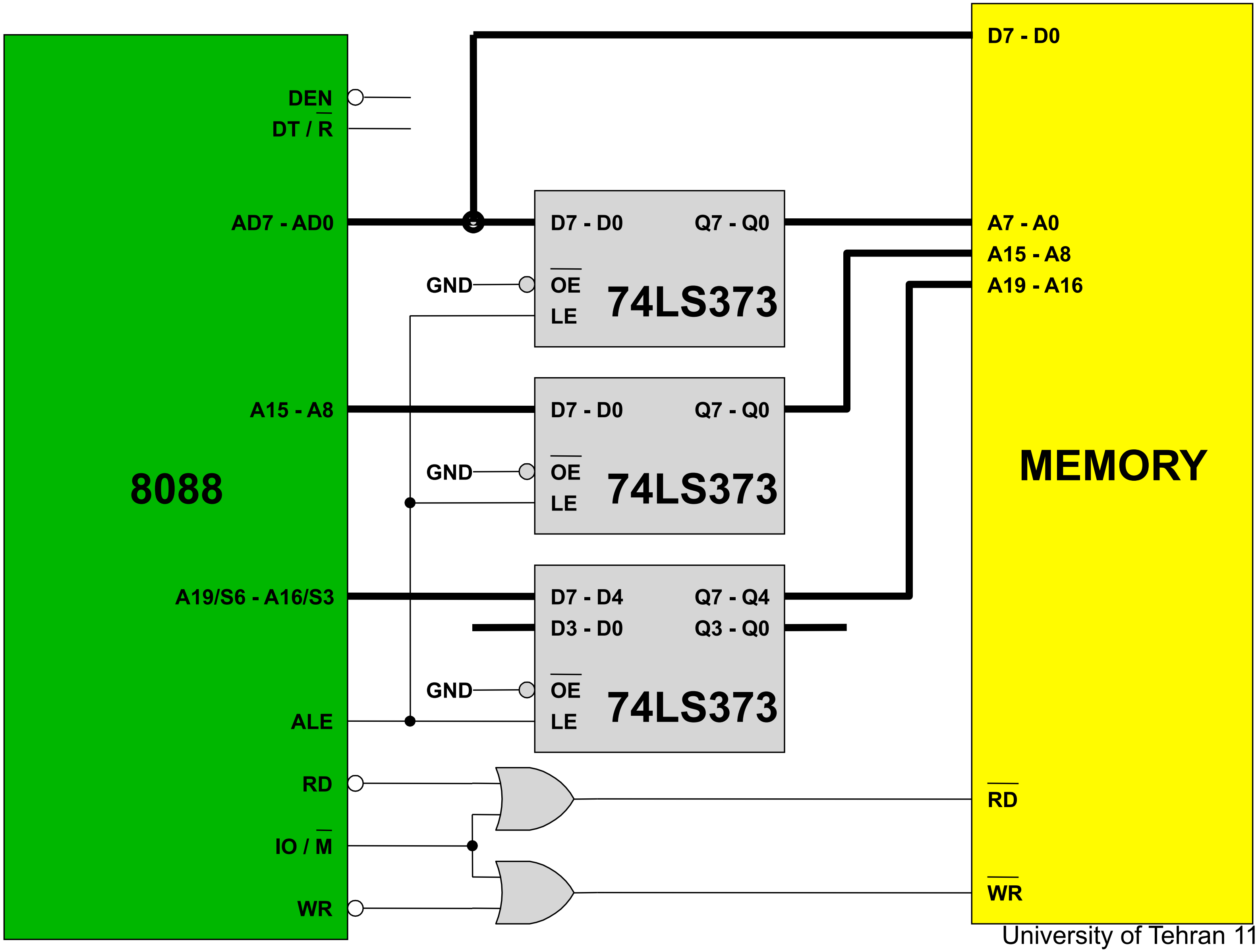
Processor Timing Diagram of 8088 (Minimum Mode) for Memory or I/O Read



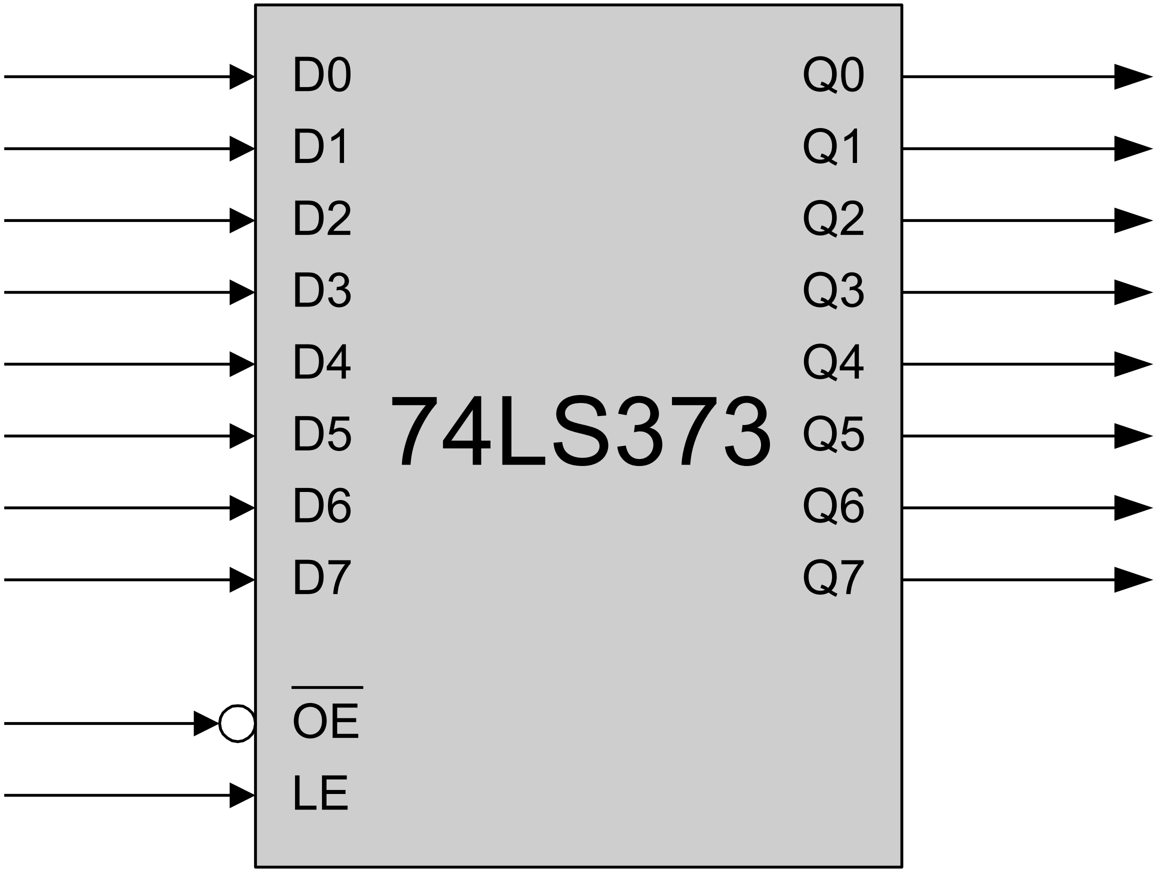
Minimum Mode



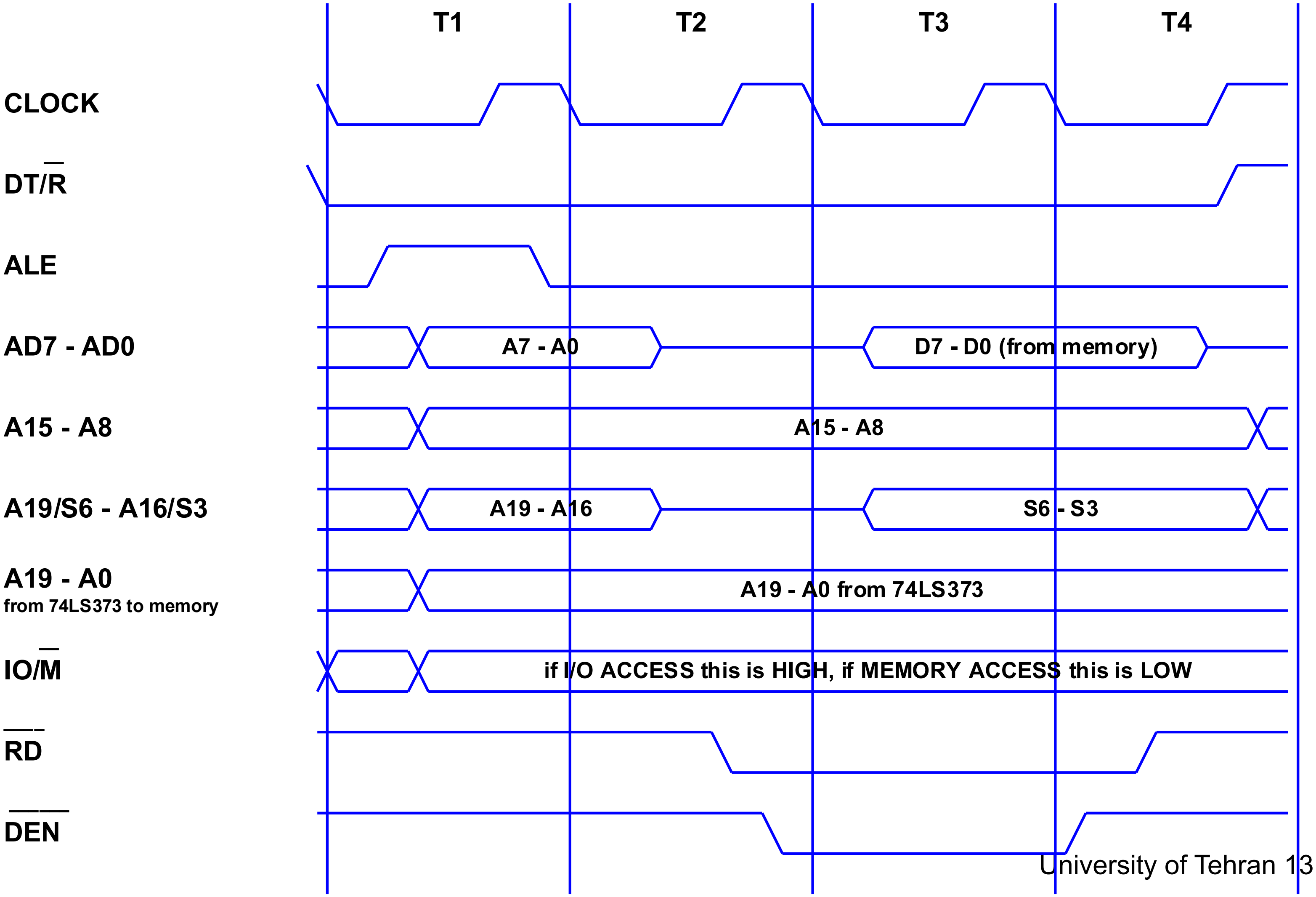
Minimum Mode



Octal Transparent Latch with 3-State Output

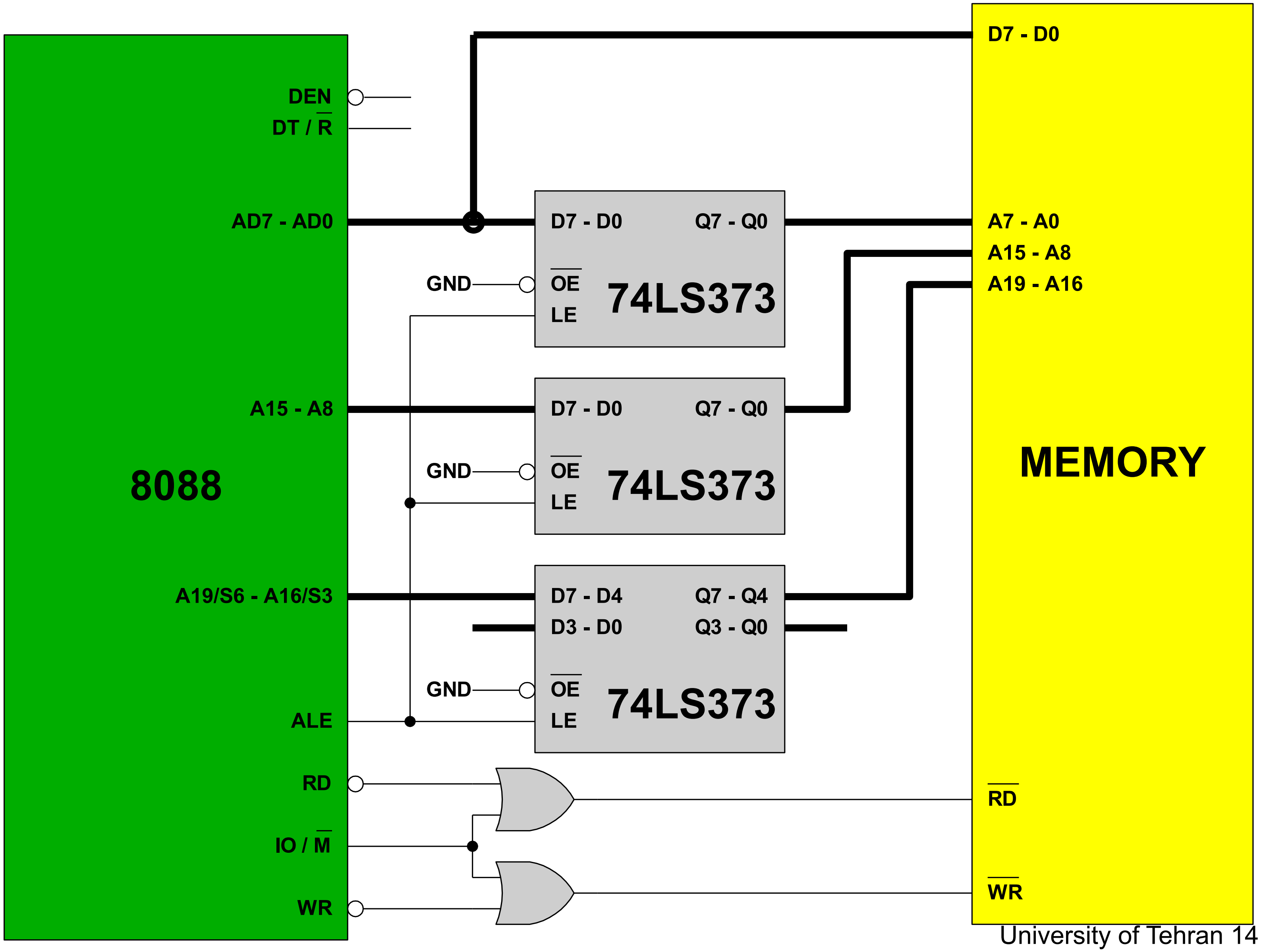


Processor Timing Diagram of 8088 (Minimum Mode) for Memory or I/O Read (with 74373)

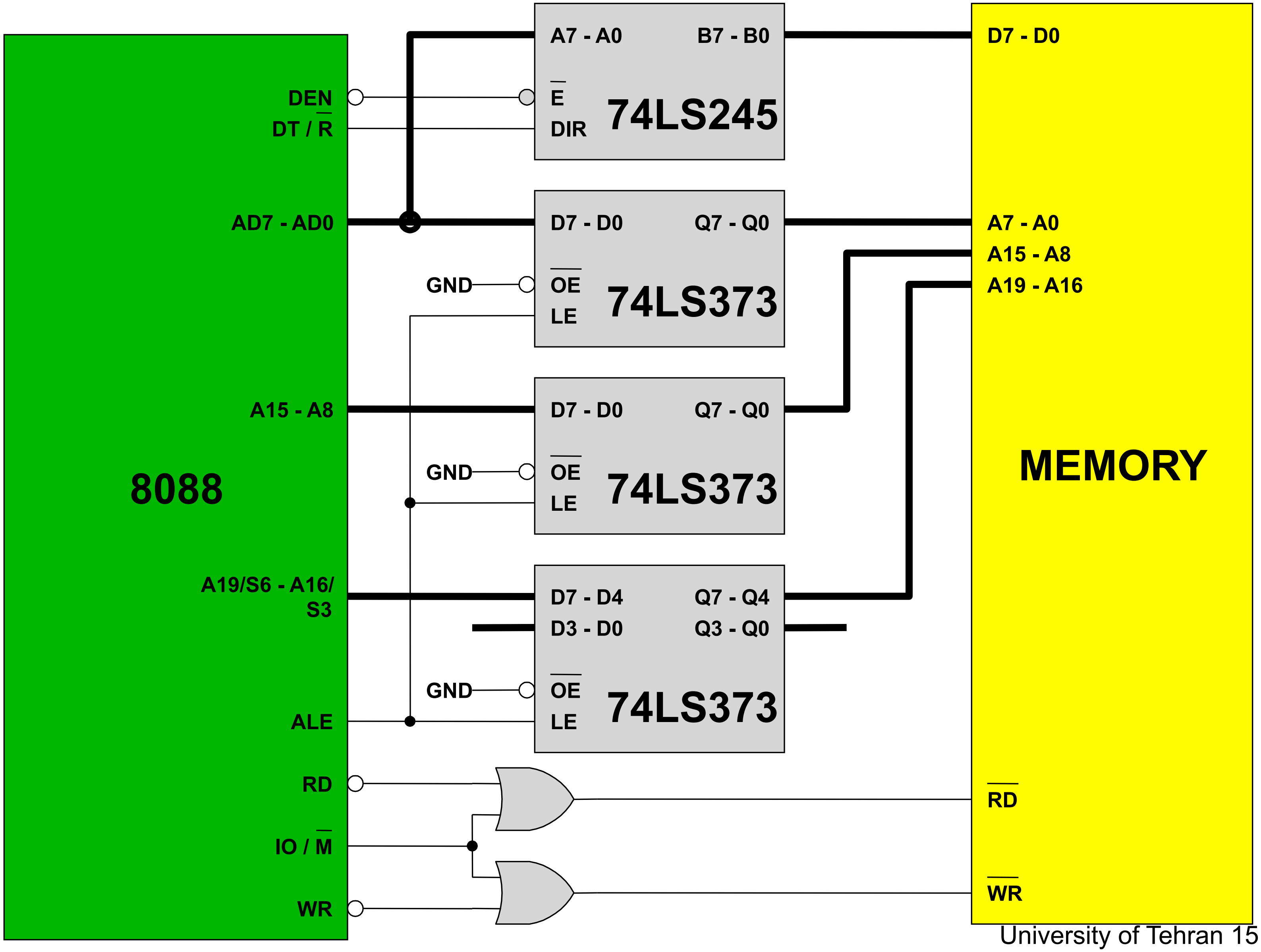


Minimum Mode

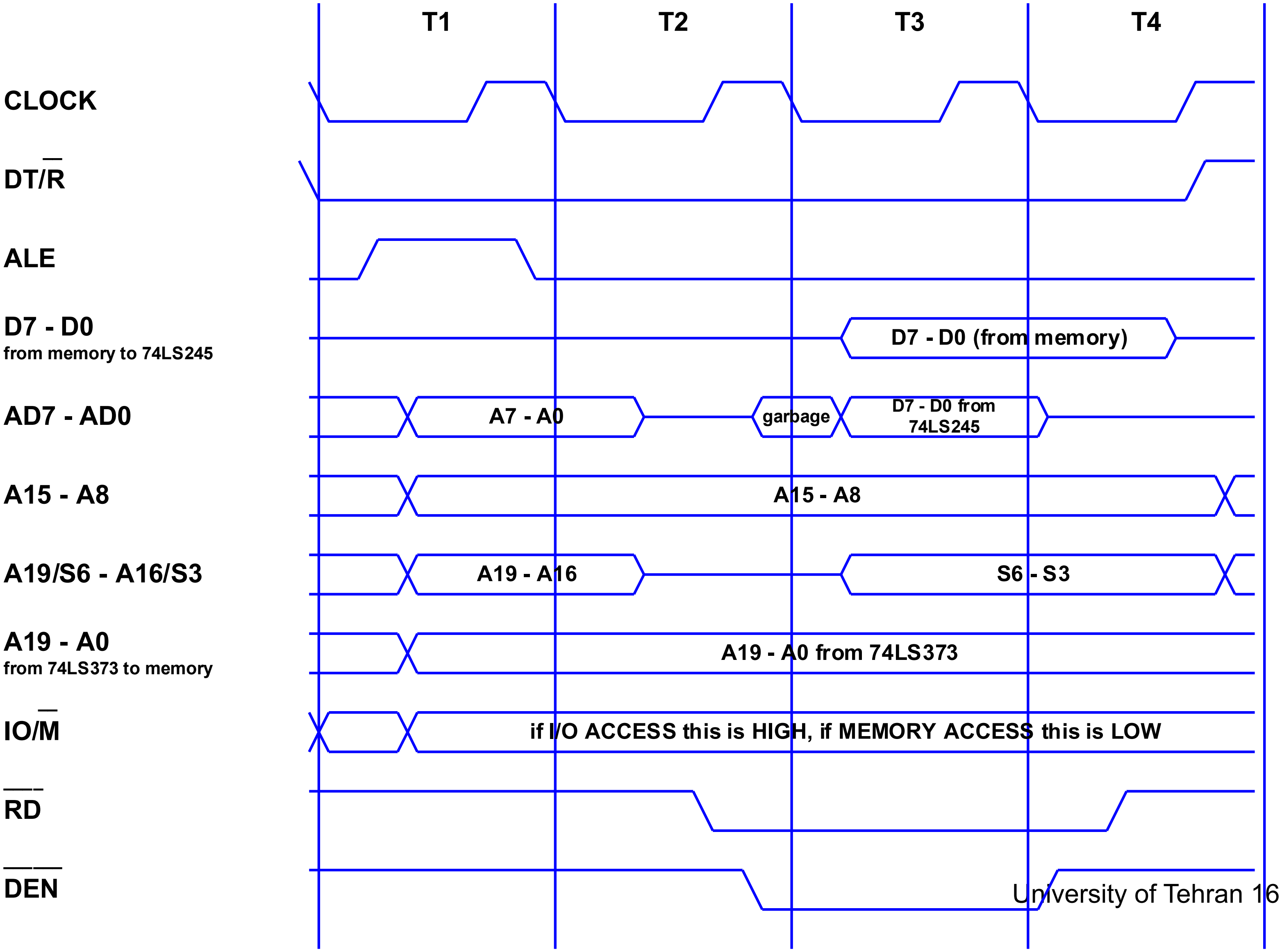
What about Data read and write



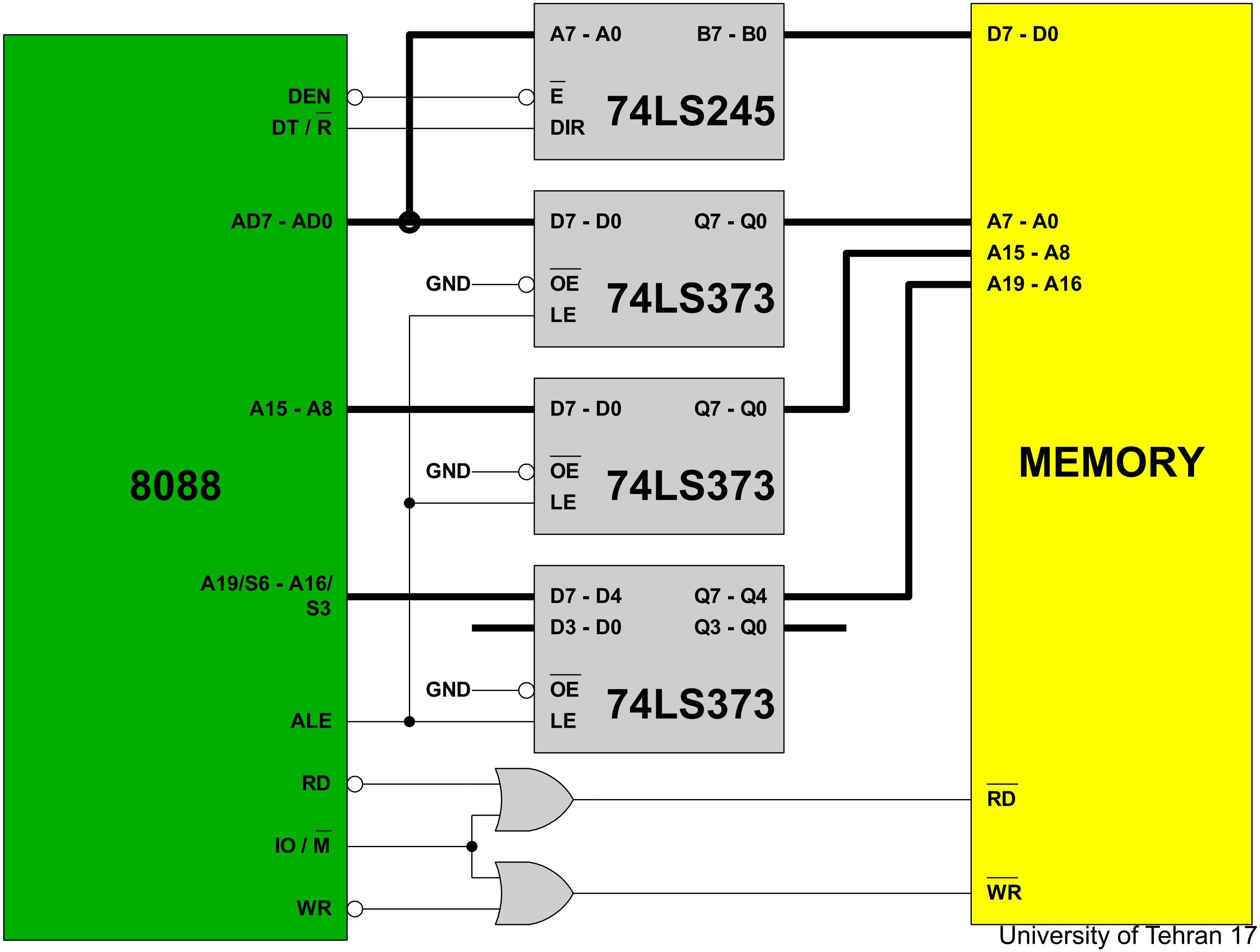
Minimum Mode



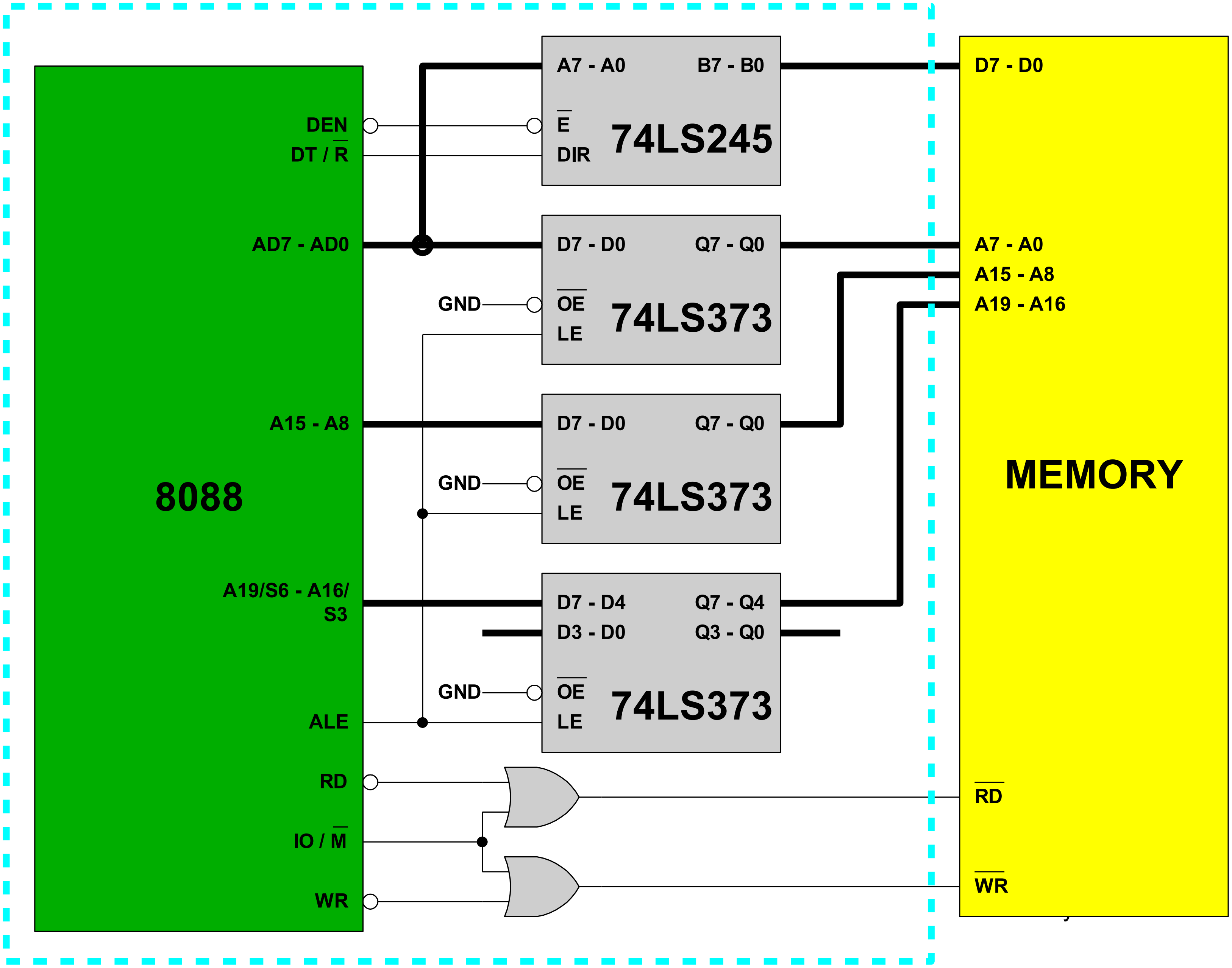
Processor Timing Diagram of 8088 (Minimum Mode) for Memory or I/O Read (with 74245)



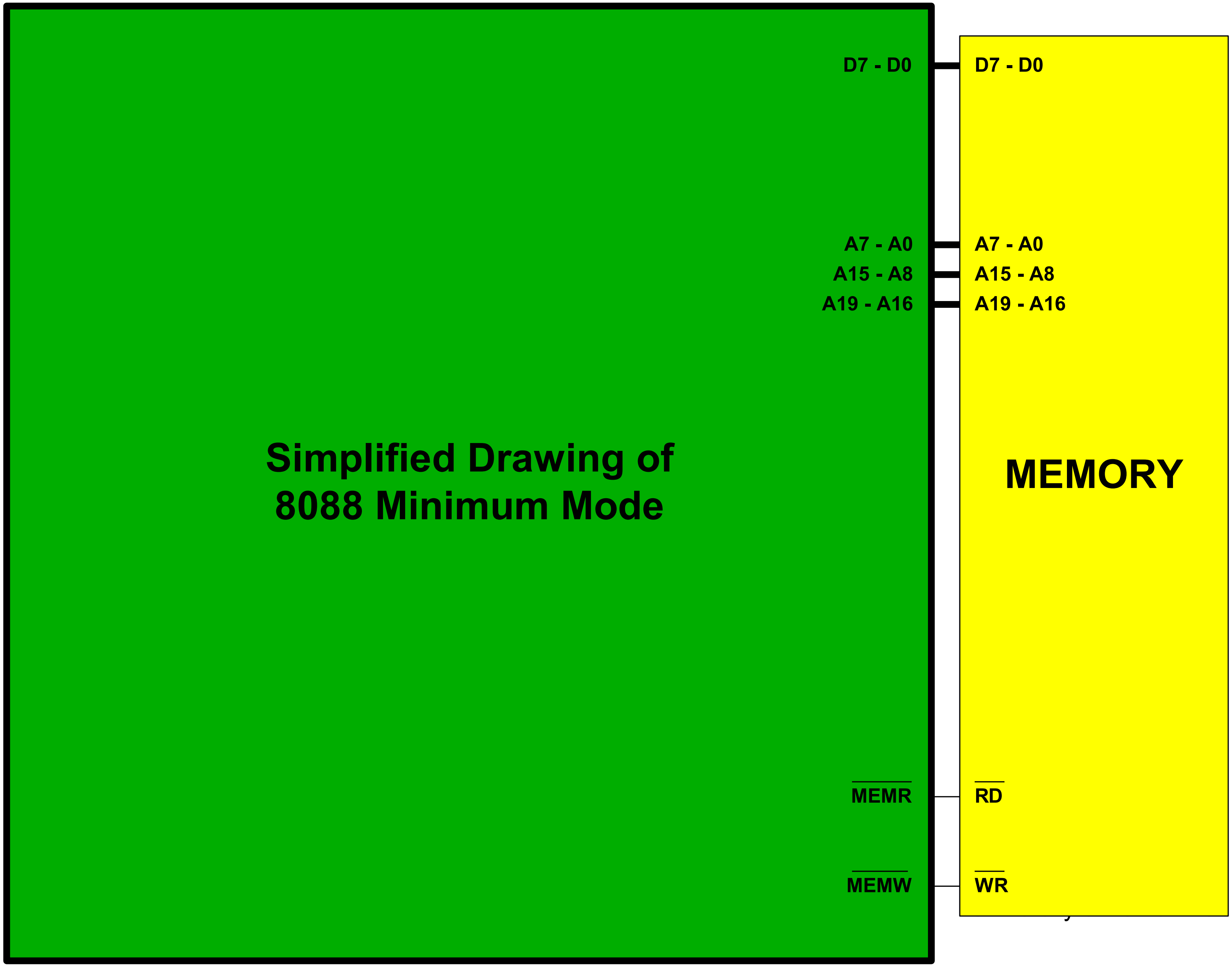
Minimum Mode



Minimum Mode



Minimum Mode



Simplified Drawing of 8088 Minimum Mode

D7 - D0

A7 - A0

A15 - A8

A19 - A16

$\overline{\text{MEMR}}$

$\overline{\text{MEMW}}$

MEMORY

D7 - D0

A7 - A0

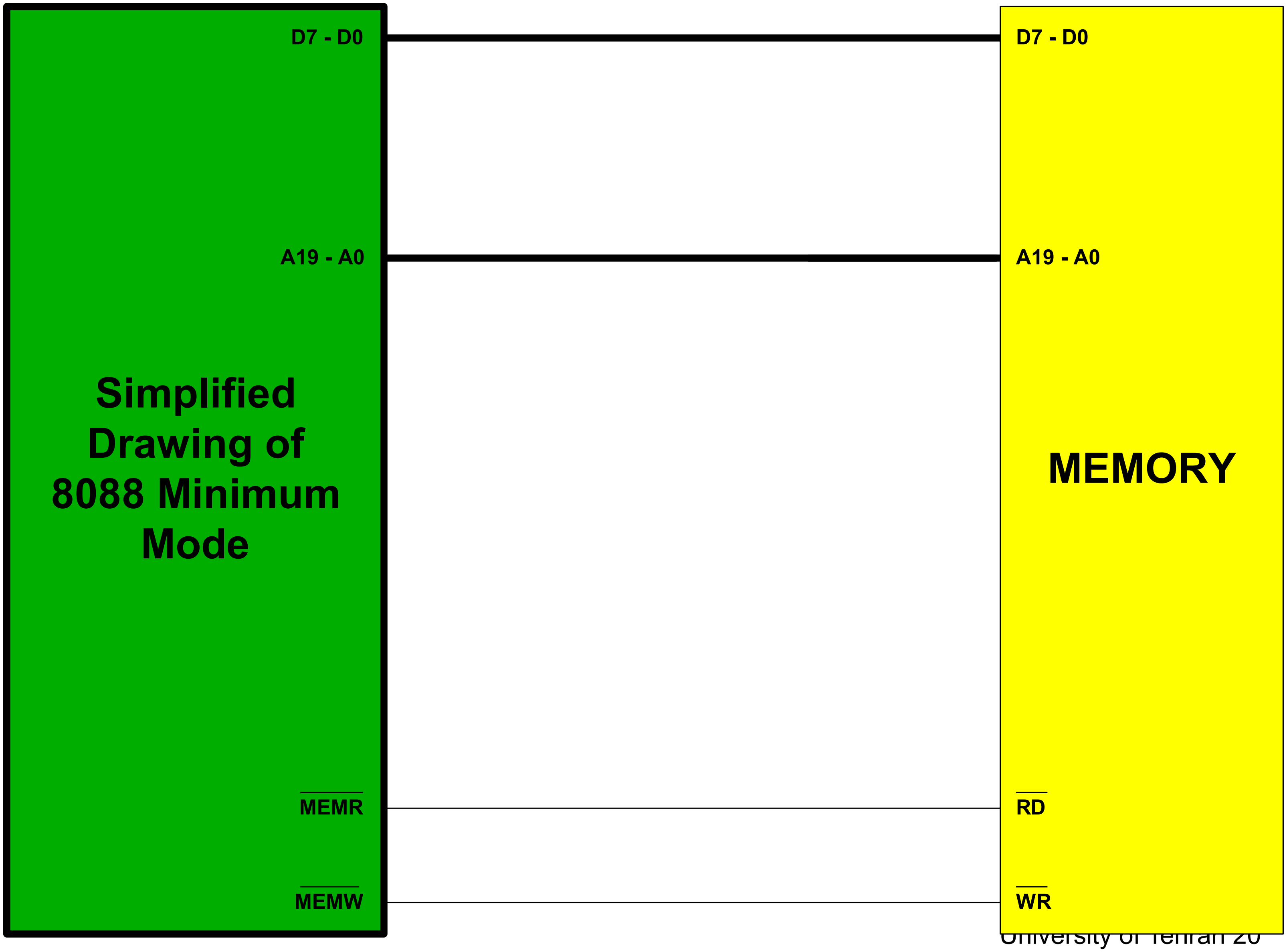
A15 - A8

A19 - A16

$\overline{\text{RD}}$

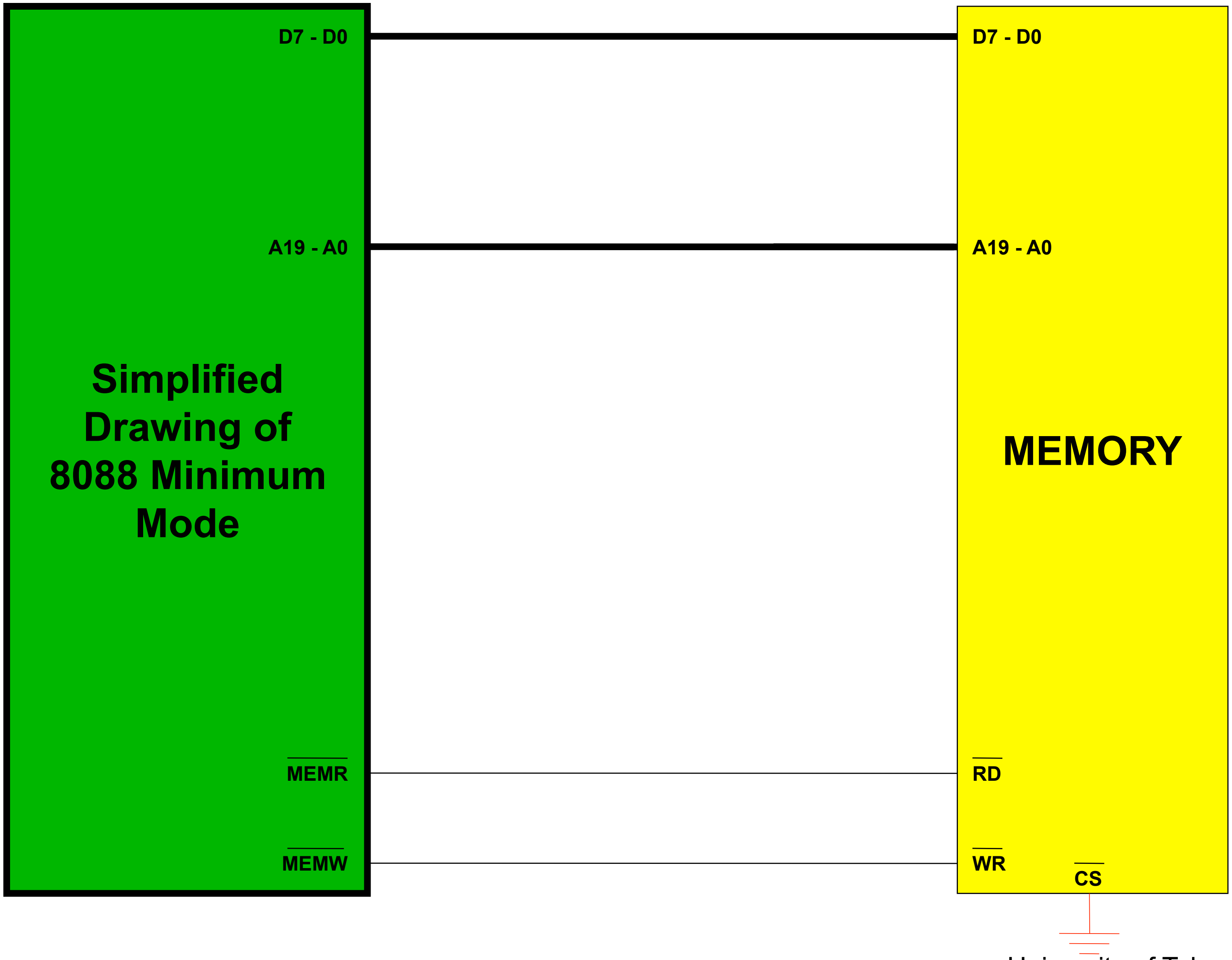
$\overline{\text{WR}}$

Minimum Mode



Minimum Mode

2^{20} bytes or 1MB



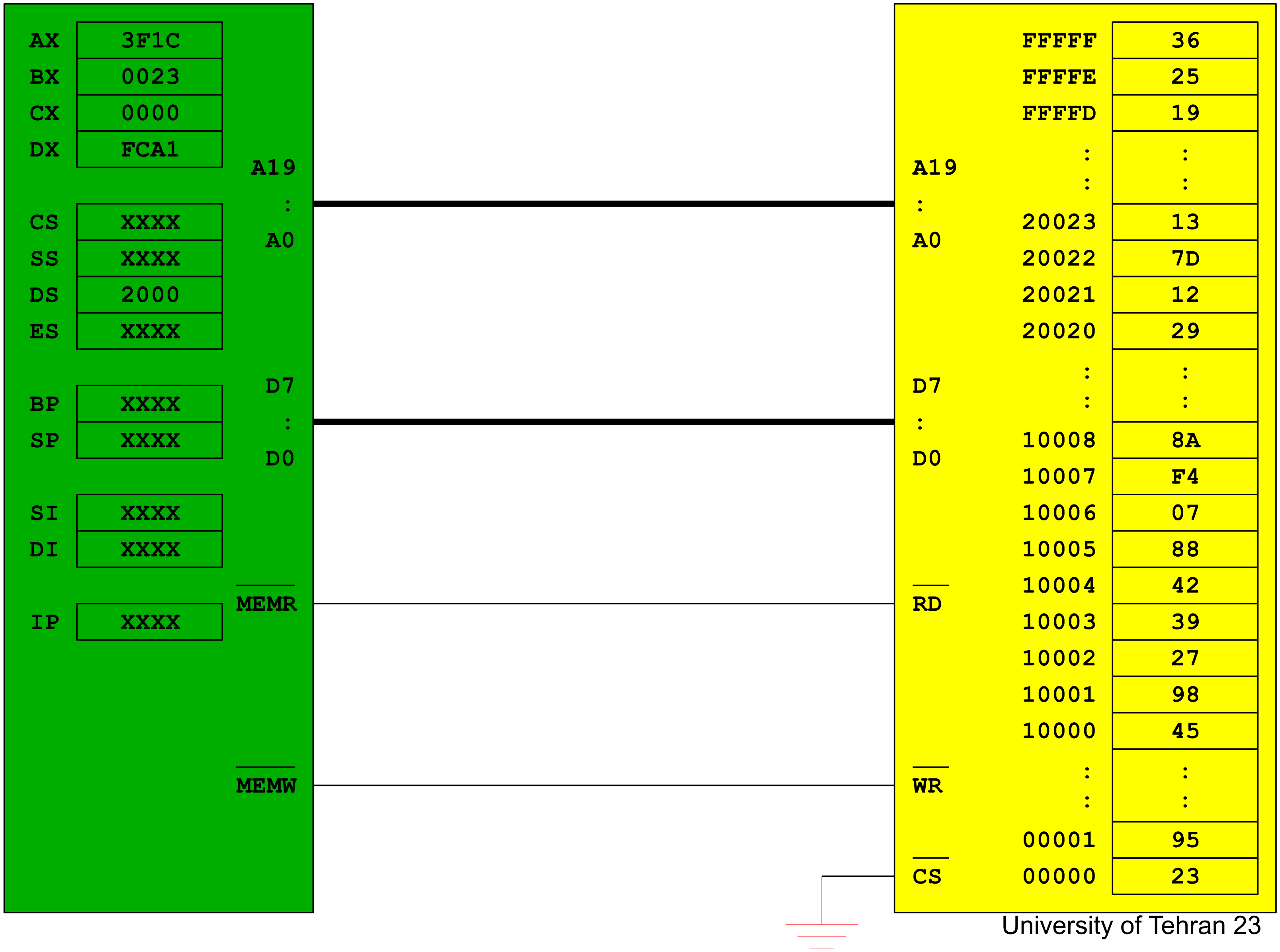
What are the memory locations of a 1MB (2^{20} bytes) Memory?

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
00000	**0000**	**0000**	**0000**	**0000**	**0000**
FFFFF	**1111**	**1111**	**1111**	**1111**	**1111**

Example: 34FD0

0011 0100 11111 1101 0000

Interfacing a 1MB Memory to the 8088 Microprocessor



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

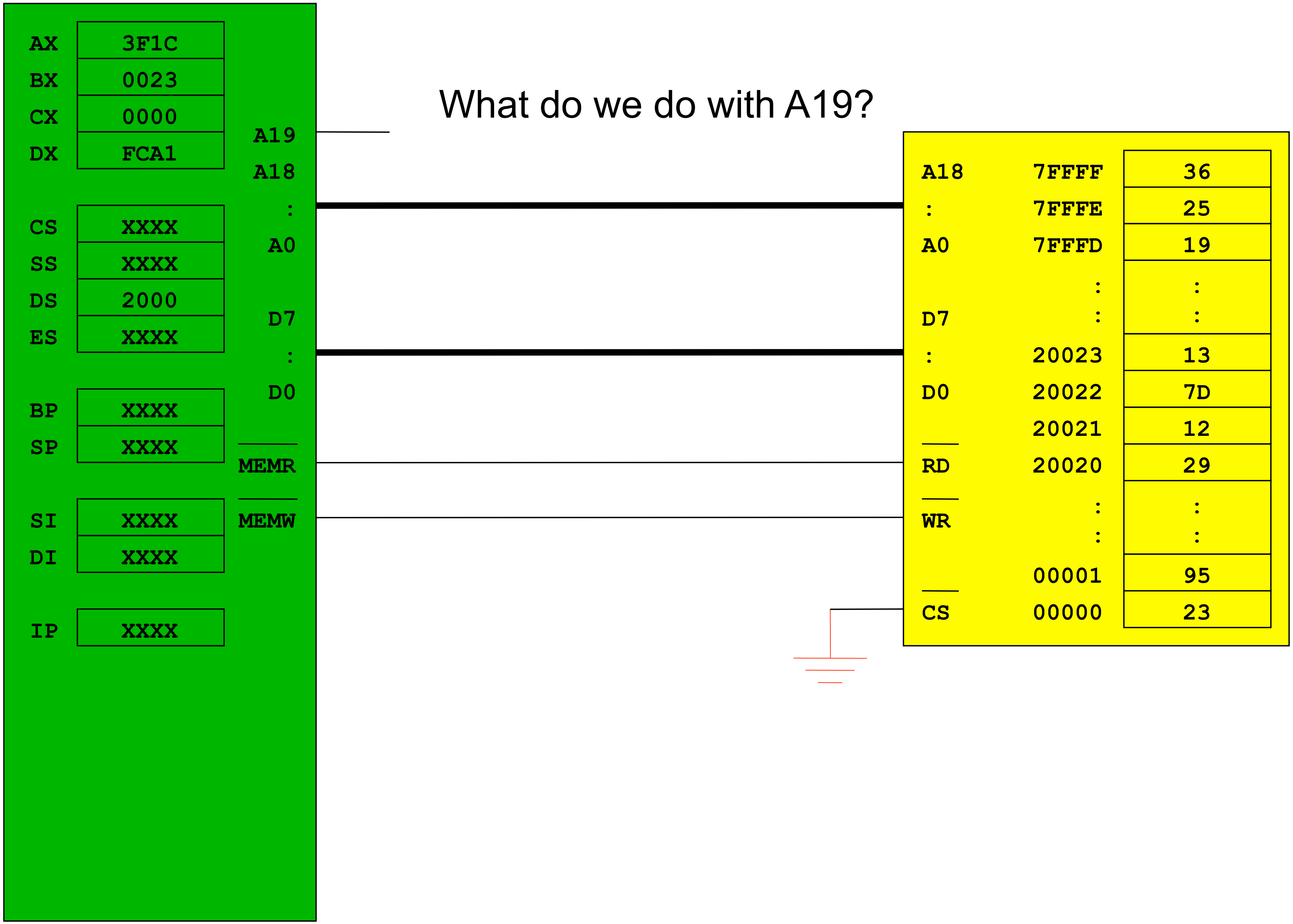
FFFFF	36
FFFFE	25
FFFFD	19
: :	: :
20023	13
20022	7D
20021	12
20020	29
: :	: :
10008	8A
10007	F4
10006	07
10005	88
10004	42
10003	39
10002	27
10001	98
10000	45
: :	: :
00001	95
00000	23

Instead of Interfacing 1MB, what will happen if you interface a 512KB Memory?

What are the memory locations of a 512KB (2^{19} bytes) Memory?

A18 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
00000	000	0000	0000	0000	0000
7FFFF	111	1111	1111	1111	1111

Interfacing a 512KB Memory to the 8088 Microprocessor

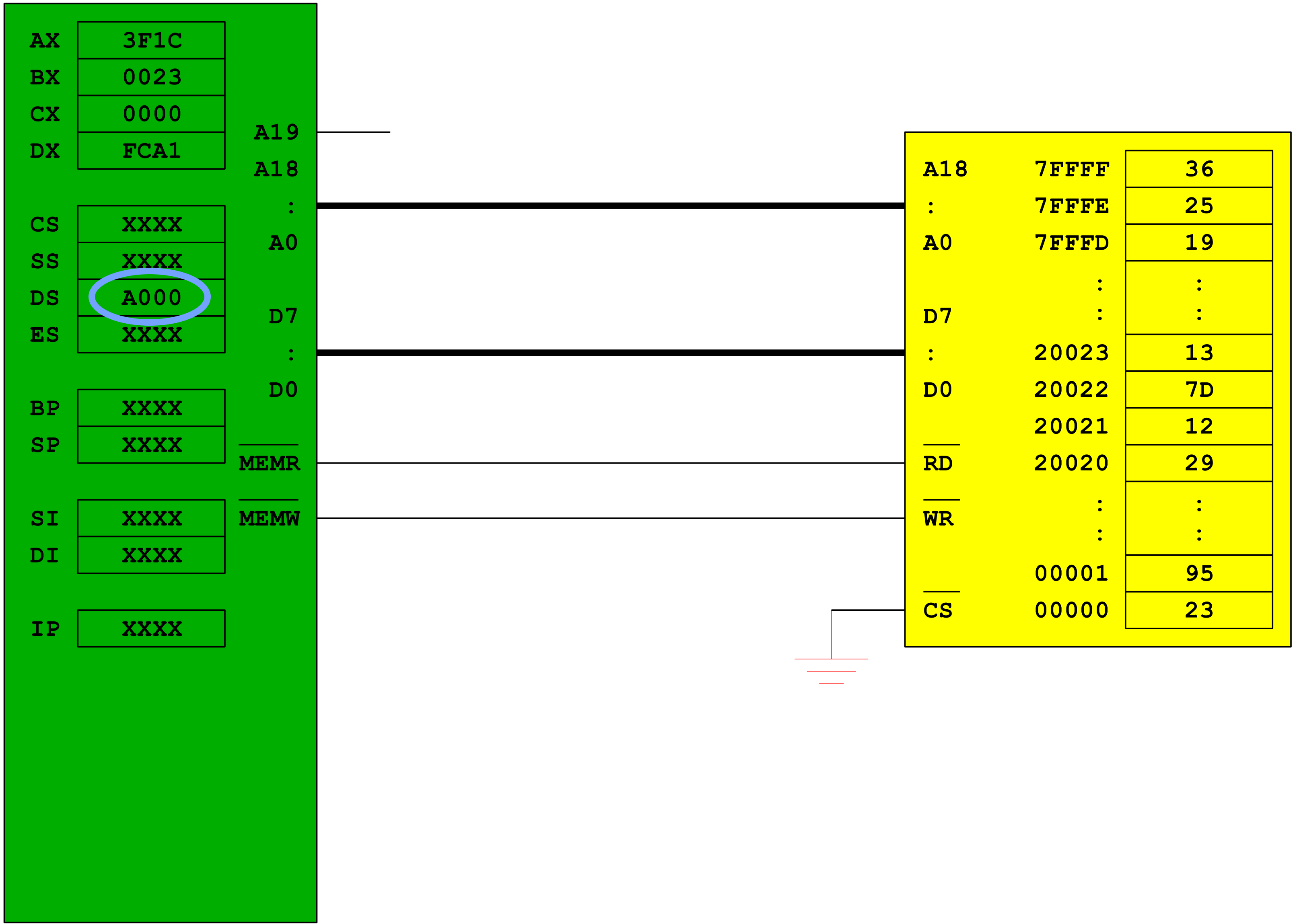


What do we do with A19?

AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
:	:
00001	95
00000	23

What if you want to read physical address A0023?



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	A000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
:	:
00001	95
00000	23

What if you want to read physical address A0023?

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
A0023	1010	0000	0000	0010	0011

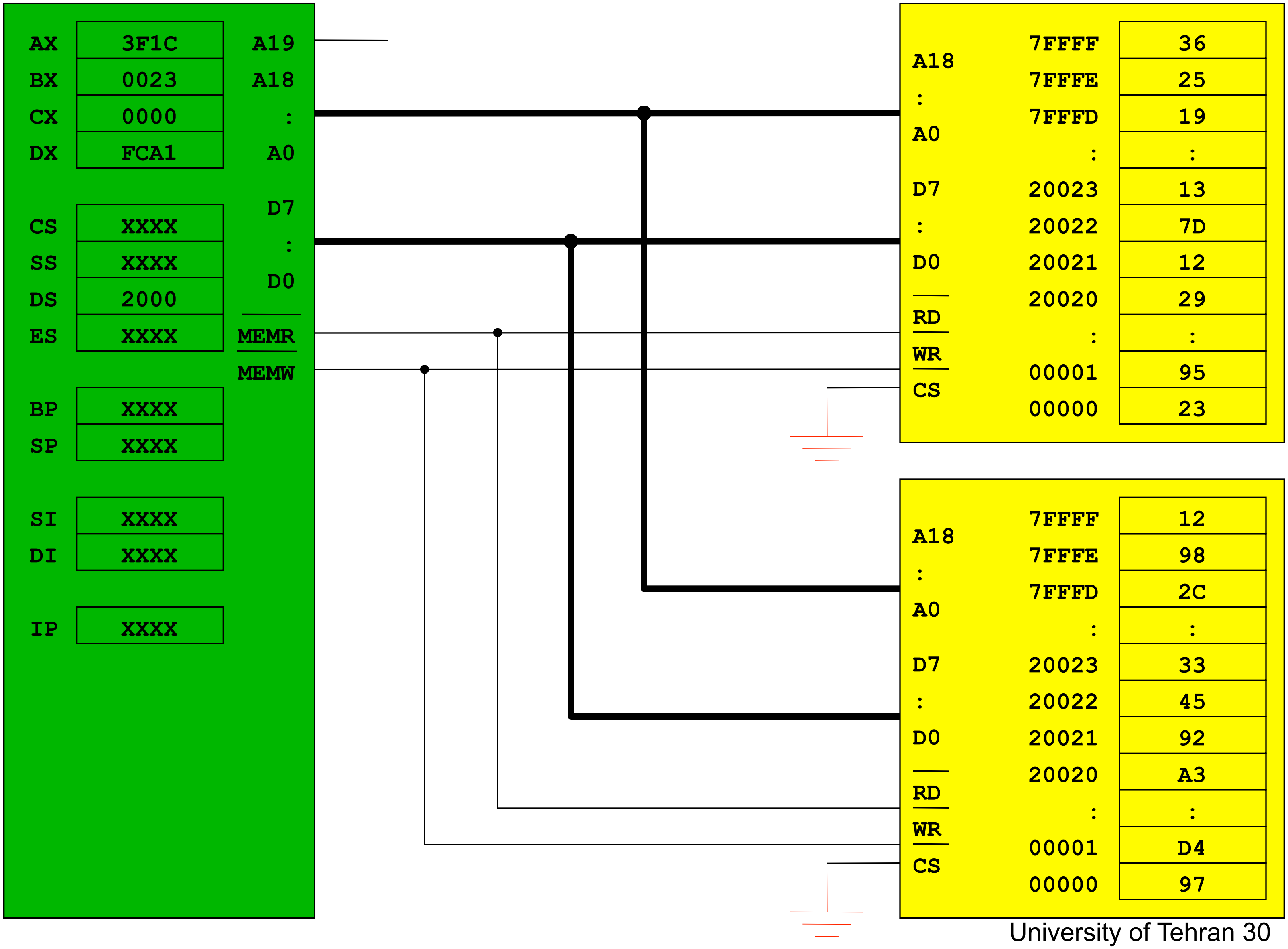
A19 is not connected to the memory so even if the 8088 microprocessor outputs a logic "1", the memory cannot "see" this.

What if you want to read physical address 20023?

A18 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
20023	0010	0000	0000	0010	0011

For memory it is the same as previous one.

Interfacing two 512KB Memory to the 8088 Microprocessor



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

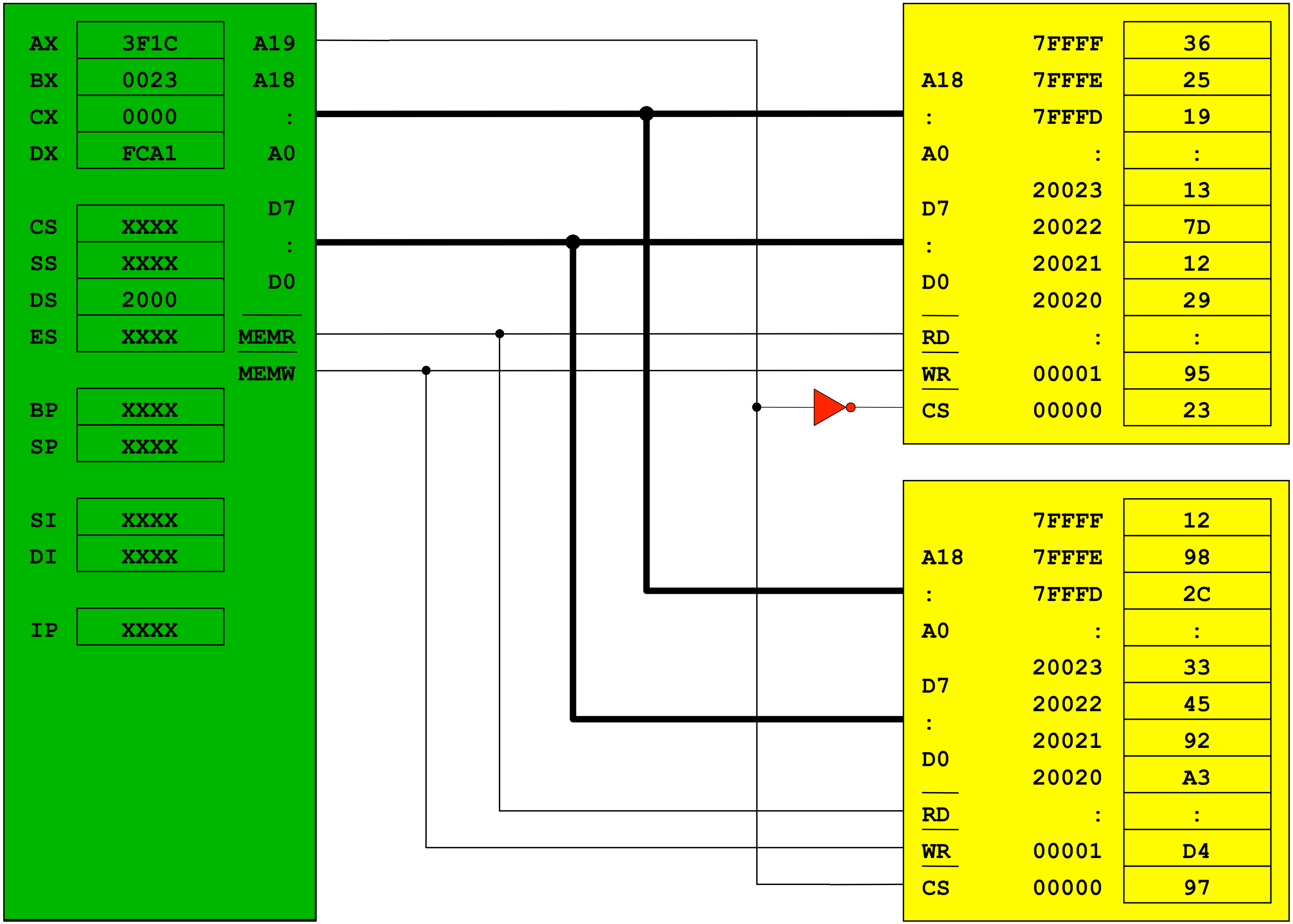
7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

Interfacing two 512KB Memory to the 8088 Microprocessor

- **Problem: Bus Conflict. The two memory chips will provide data at the same time when microprocessor performs a memory read.**
- **Solution: Use address line A19 as an "arbiter". If A19 outputs a logic "1" the upper memory is enabled (and the lower memory is disabled) and vice-versa.**

Interfacing two 512KB Memory to the 8088 Microprocessor



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

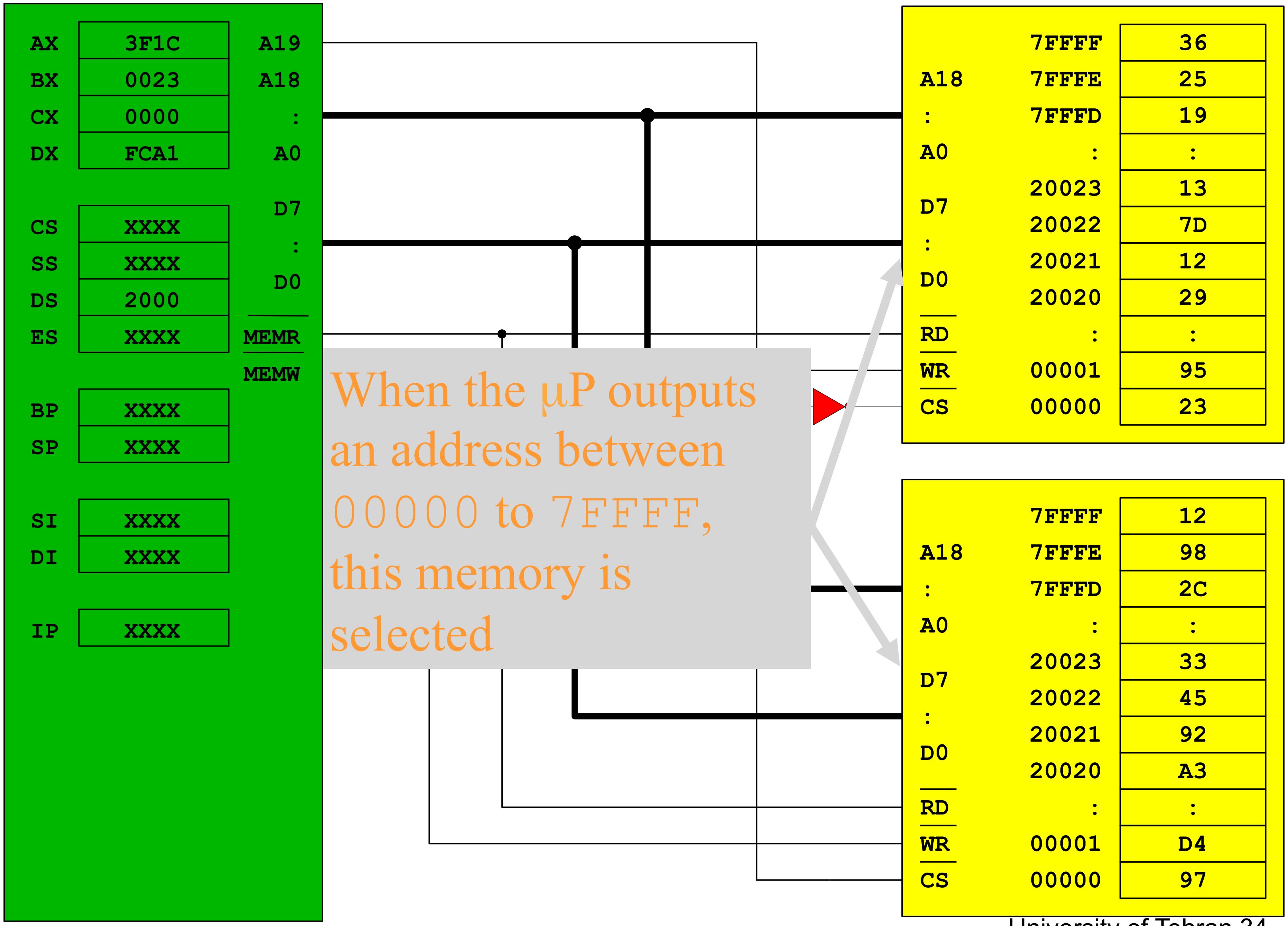
7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

What are the memory locations of two consecutive 512KB (2^{19} bytes) Memory?

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
00000	0000	0000	0000	0000	0000
7FFFF	0111	1111	1111	1111	1111
80000	1000	0000	0000	0000	0000
FFFFF	1111	1111	1111	1111	1111

Interfacing two 512KB Memory to the 8088 Microprocessor



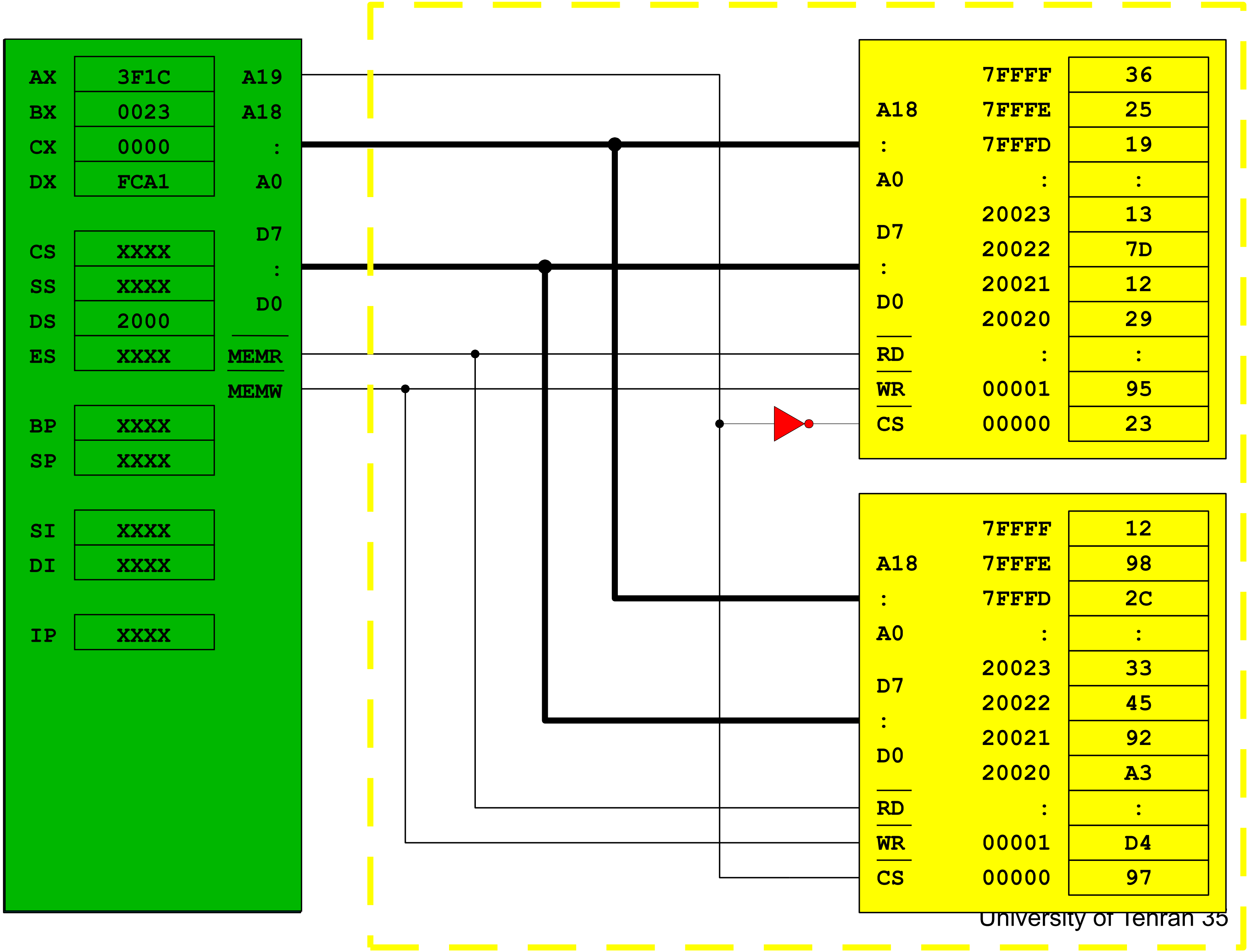
AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

When the µP outputs an address between 00000 to 7FFFF, this memory is selected

Interfacing two 512KB Memory to the 8088 Microprocessor

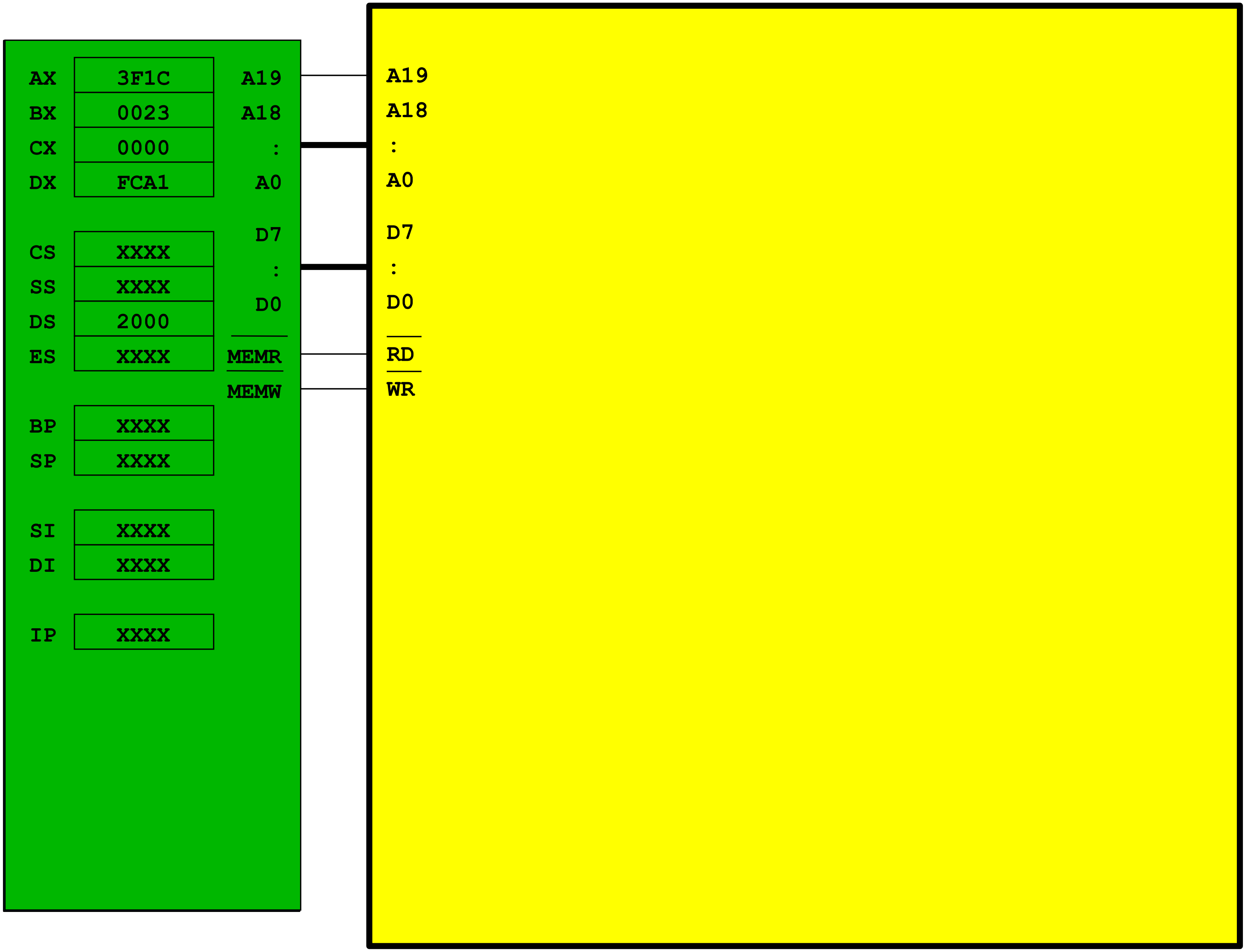


AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

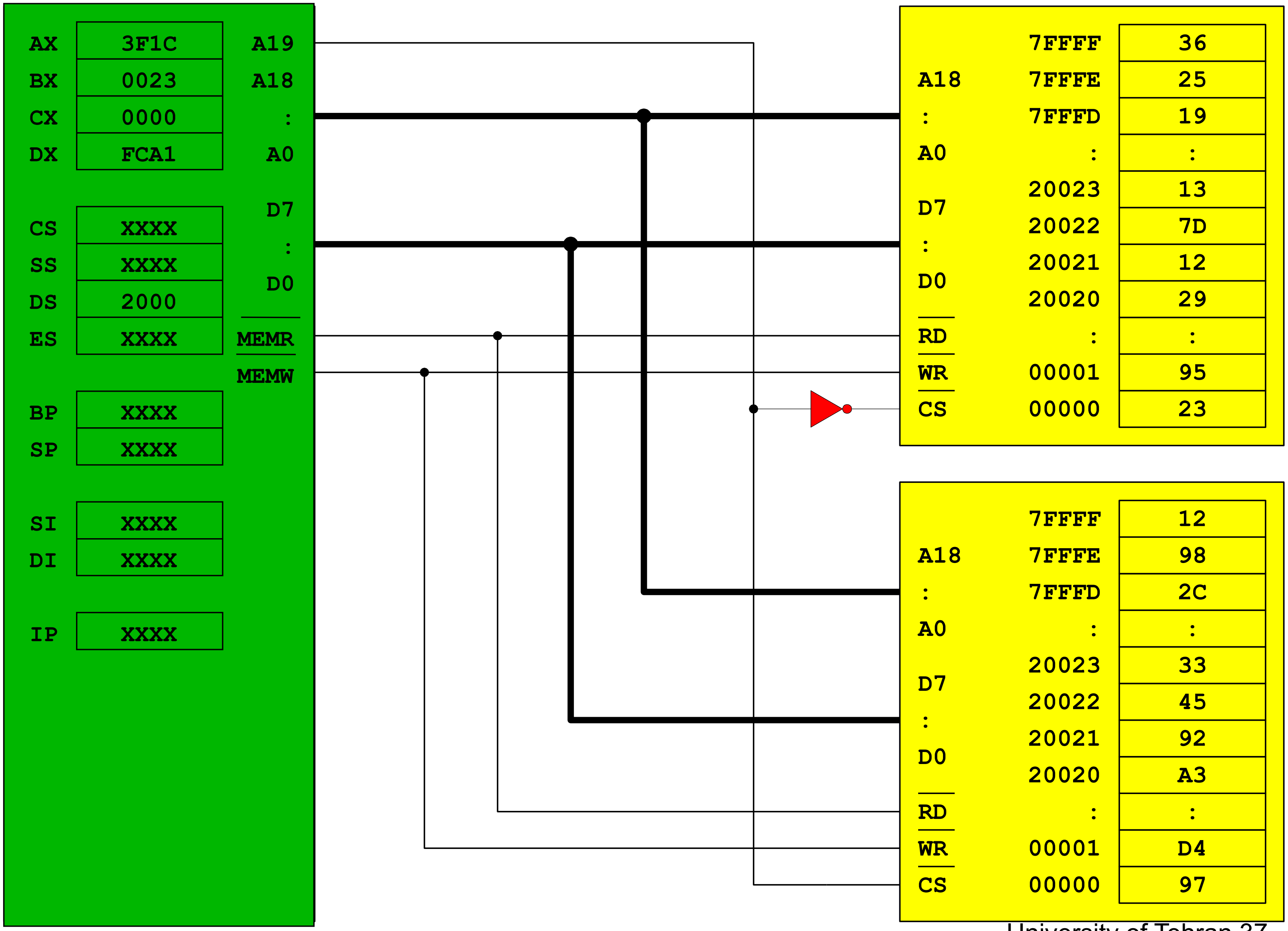
7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

Interfacing two 512KB Memory to the 8088 Microprocessor



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

What if we remove the lower memory?

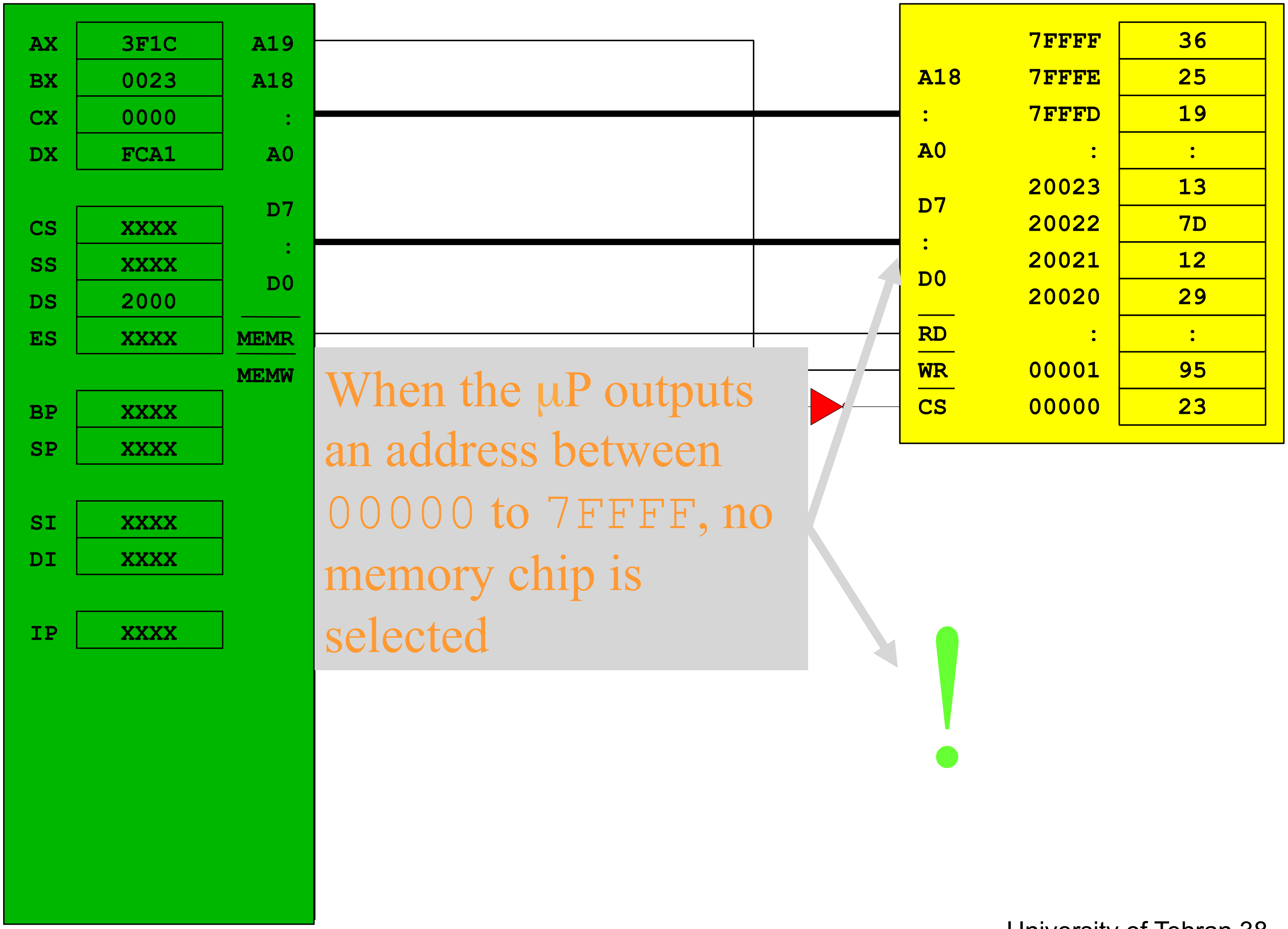


AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

What if we remove the lower memory?

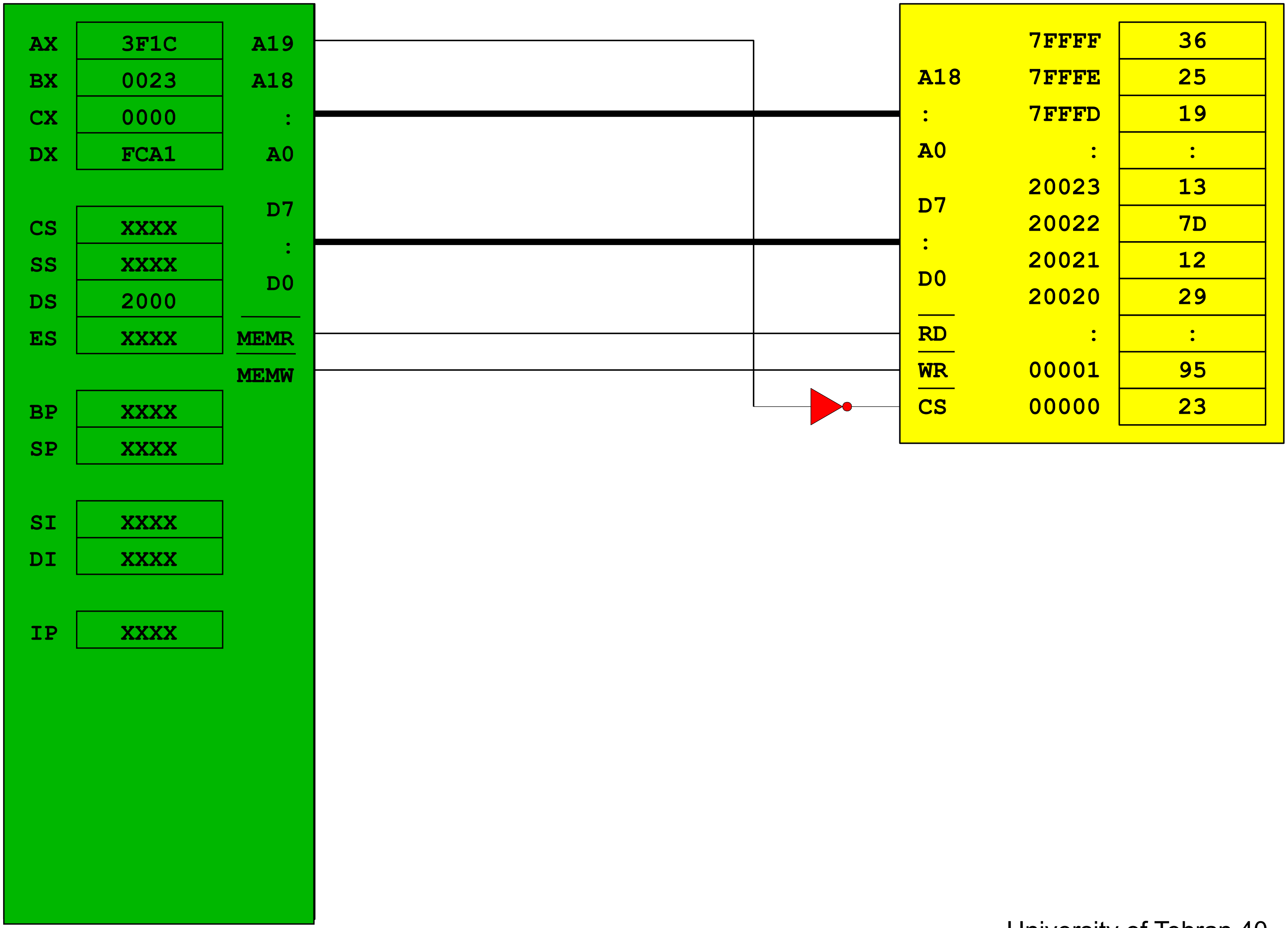


When the µP outputs an address between 00000 to 7FFFF, no memory chip is selected

Full and Partial Decoding

- Full Decoding
  - When all of the “useful” address lines are connected the memory/device to perform selection
- Partial Decoding
  - When some of the “useful” address lines are connected the memory/device to perform selection
  - Using this type of decoding results into roll-over addresses

Full Decoding



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

Full Decoding

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
80000	1000	0000	0000	0000	0000
FFFFF	1111	1111	1111	1111	1111

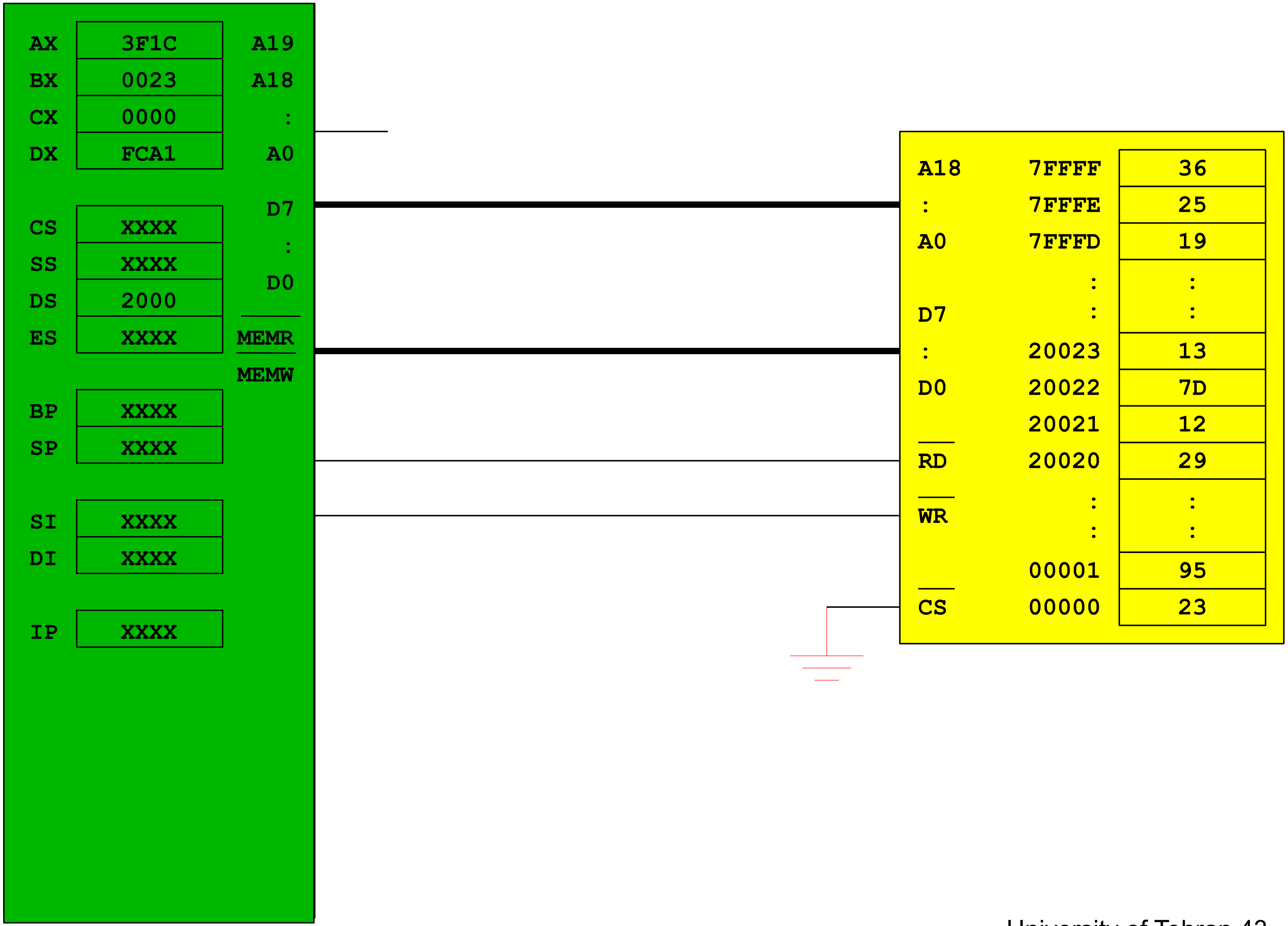
A19 should be a logic “1” for the memory chip to be enabled

Full Decoding

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210
00000	0000	0000	0000	0000	0000
7FFFF	0111	1111	1111	1111	1111

Therefore if the microprocessor outputs an address between 00000 to 7FFFF, whose A19 is a logic "0", the memory chip will not be selected

Partial Decoding



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
:	:
00001	95
00000	23

Partial Decoding

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
00000	0000	0000	0000	0000	0000
7FFFF	0111	1111	1111	1111	1111
80000	1000	0000	0000	0000	0000
FFFFF	1111	1111	1111	1111	1111

The value of A19 is INSIGNIFICANT to the memory chip, therefore A19 has no bearing whether the memory chip will be enabled or not

Partial Decoding

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210
00000	0000	0000	0000	0000	0000
7FFFF	0111	1111	1111	1111	1111
80000	1000	0000	0000	0000	0000
FFFFF	1111	1111	1111	1111	1111

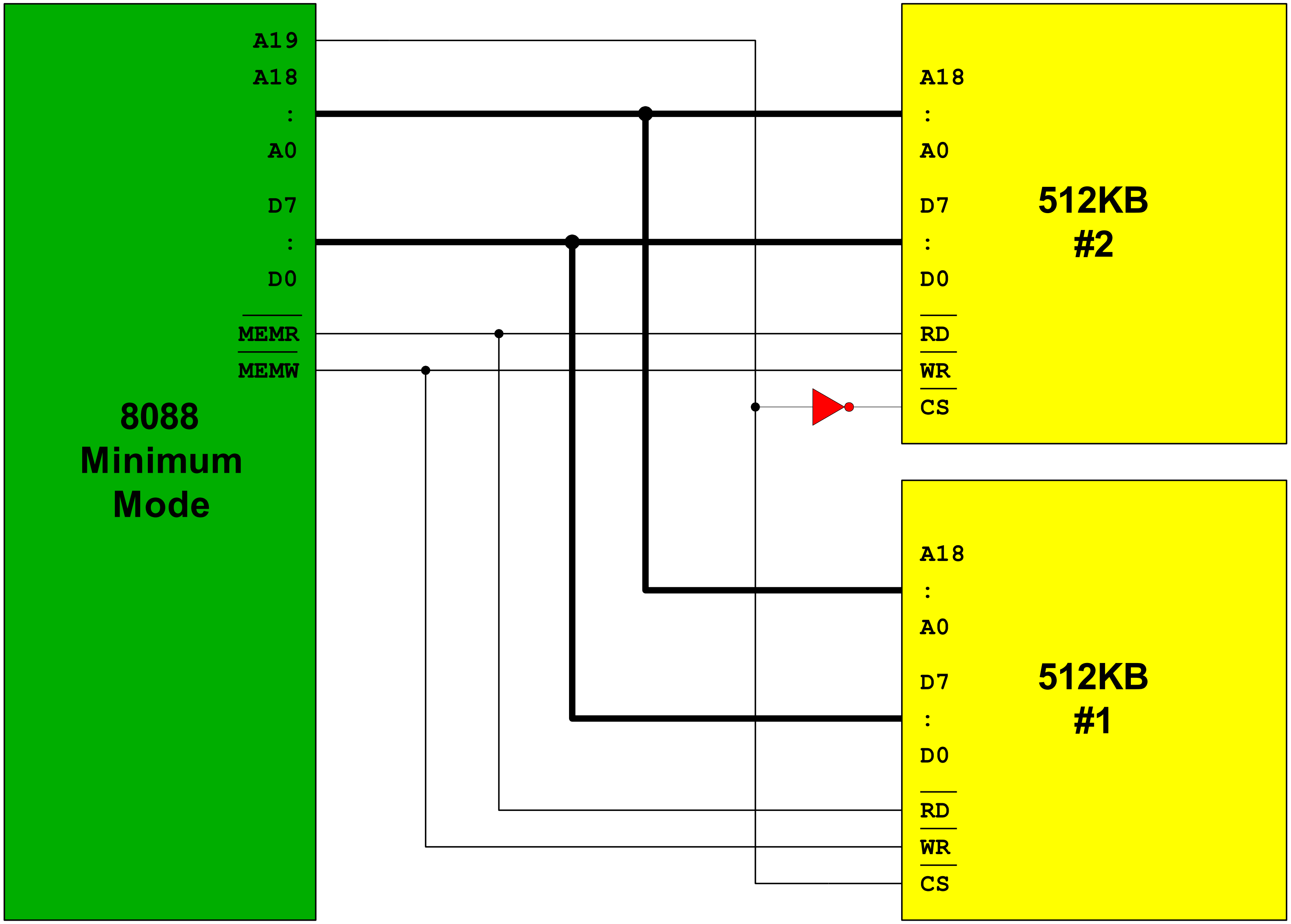
ACTUAL ADDRESS

Partial Decoding

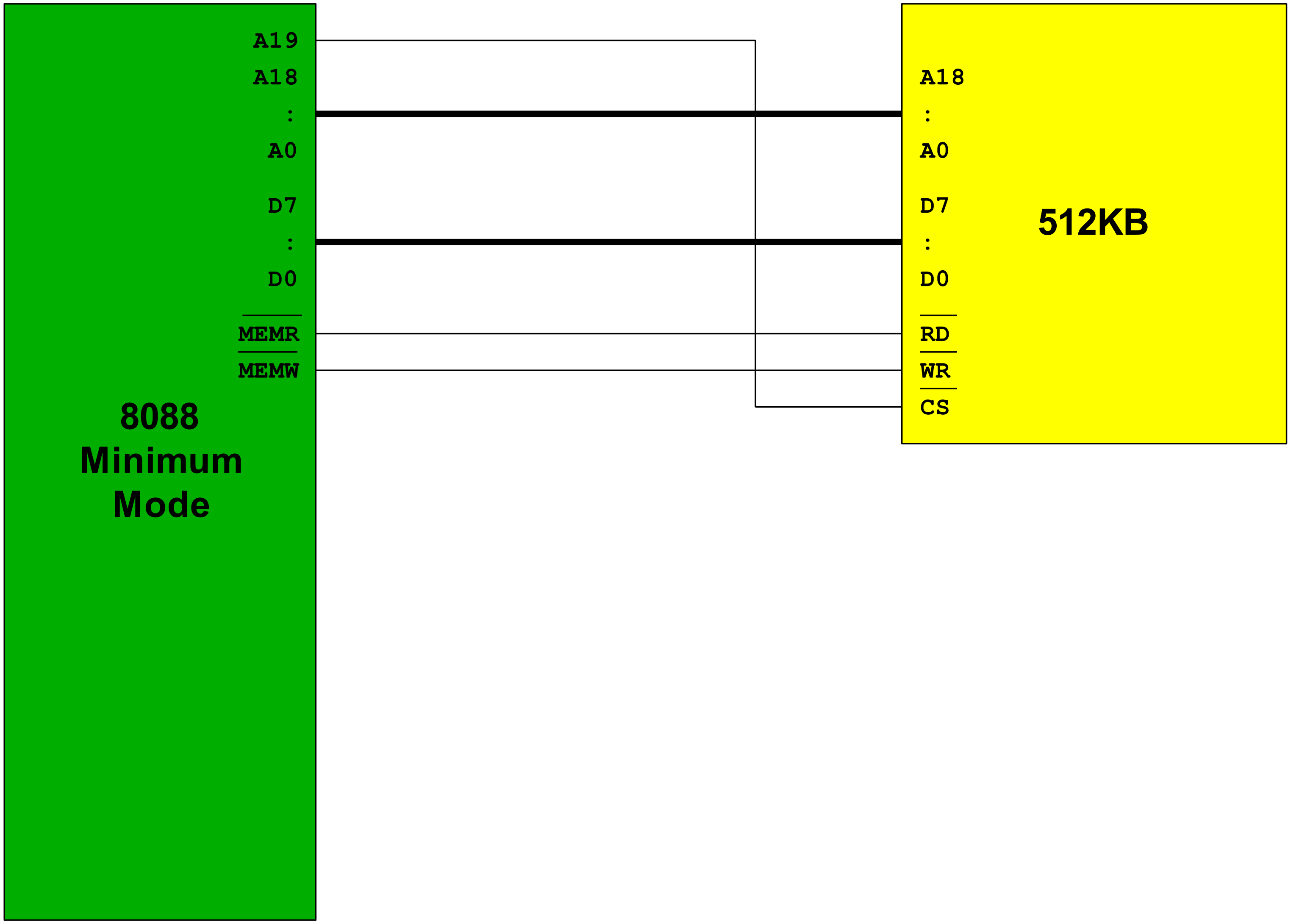
A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
00000	**0000**	**0000**	**0000**	**0000**	**0000**
7FFFF	**0111**	**1111**	**1111**	**1111**	**1111**
80000	**1000**	**0000**	**0000**	**0000**	**0000**
FFFFF	**1111**	**1111**	**1111**	**1111**	**1111**

ACTUAL ADDRESS

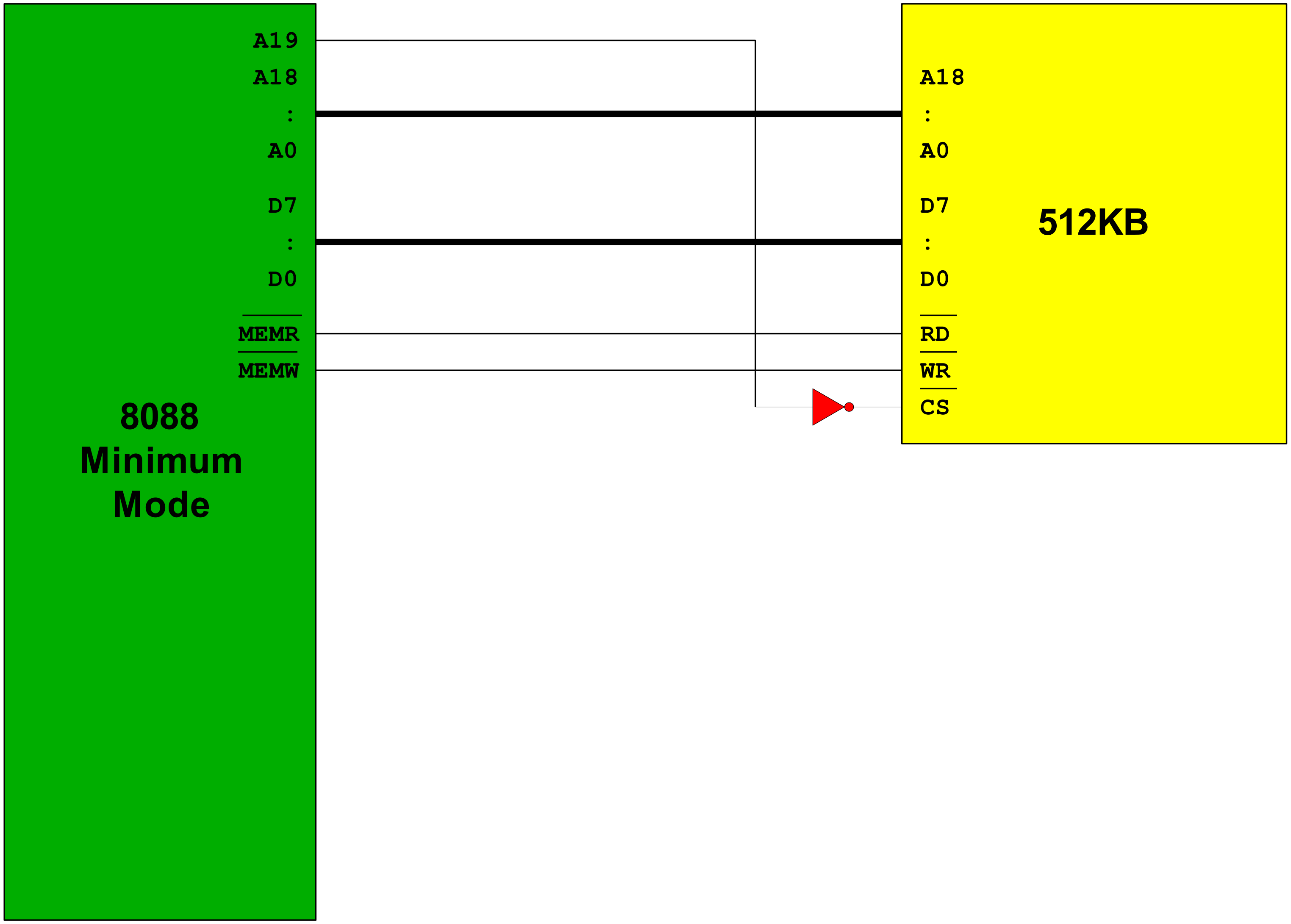
Interfacing two 512K Memory Chips to the 8088 Microprocessor



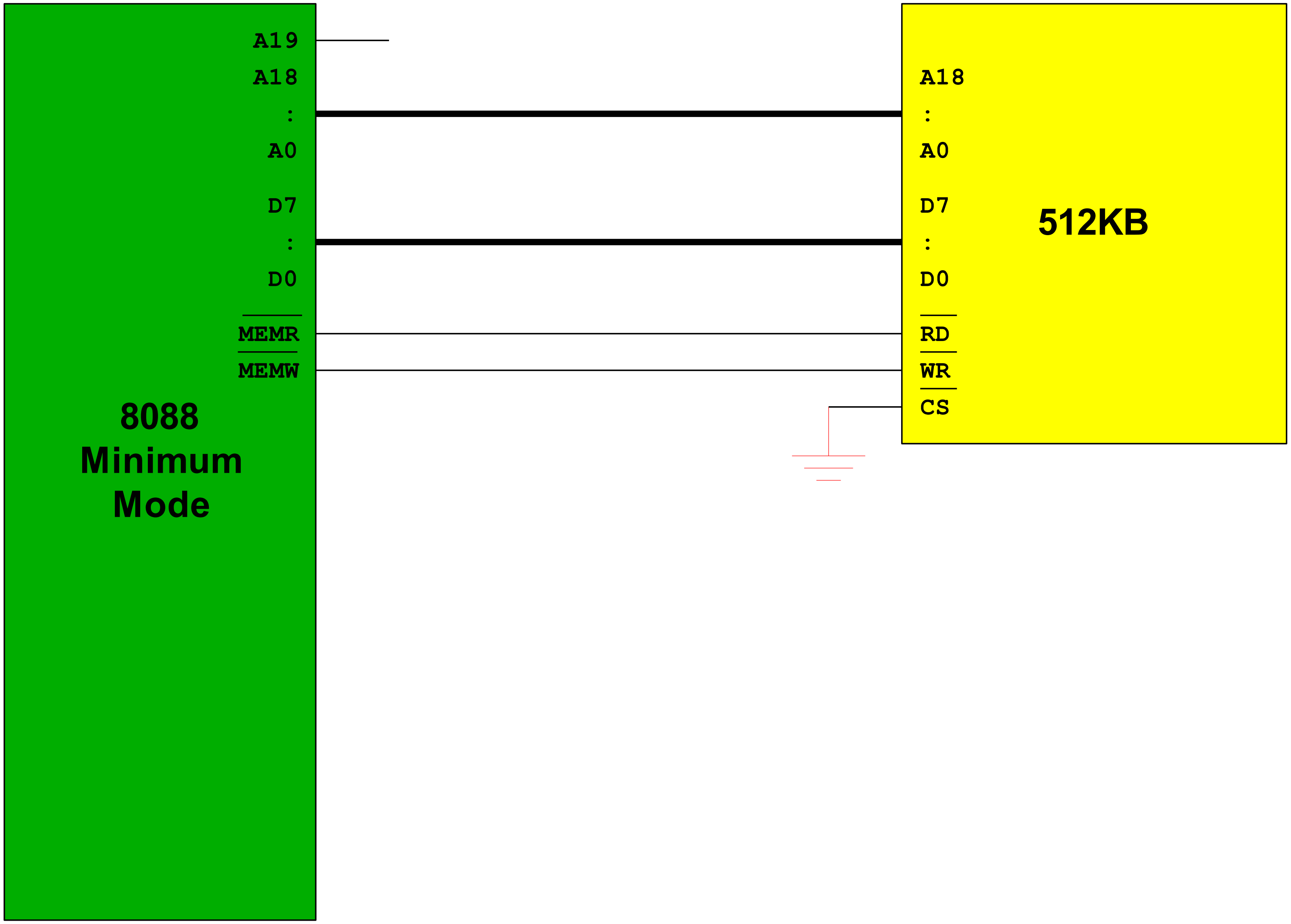
Interfacing one 512K Memory Chips to the 8088 Microprocessor



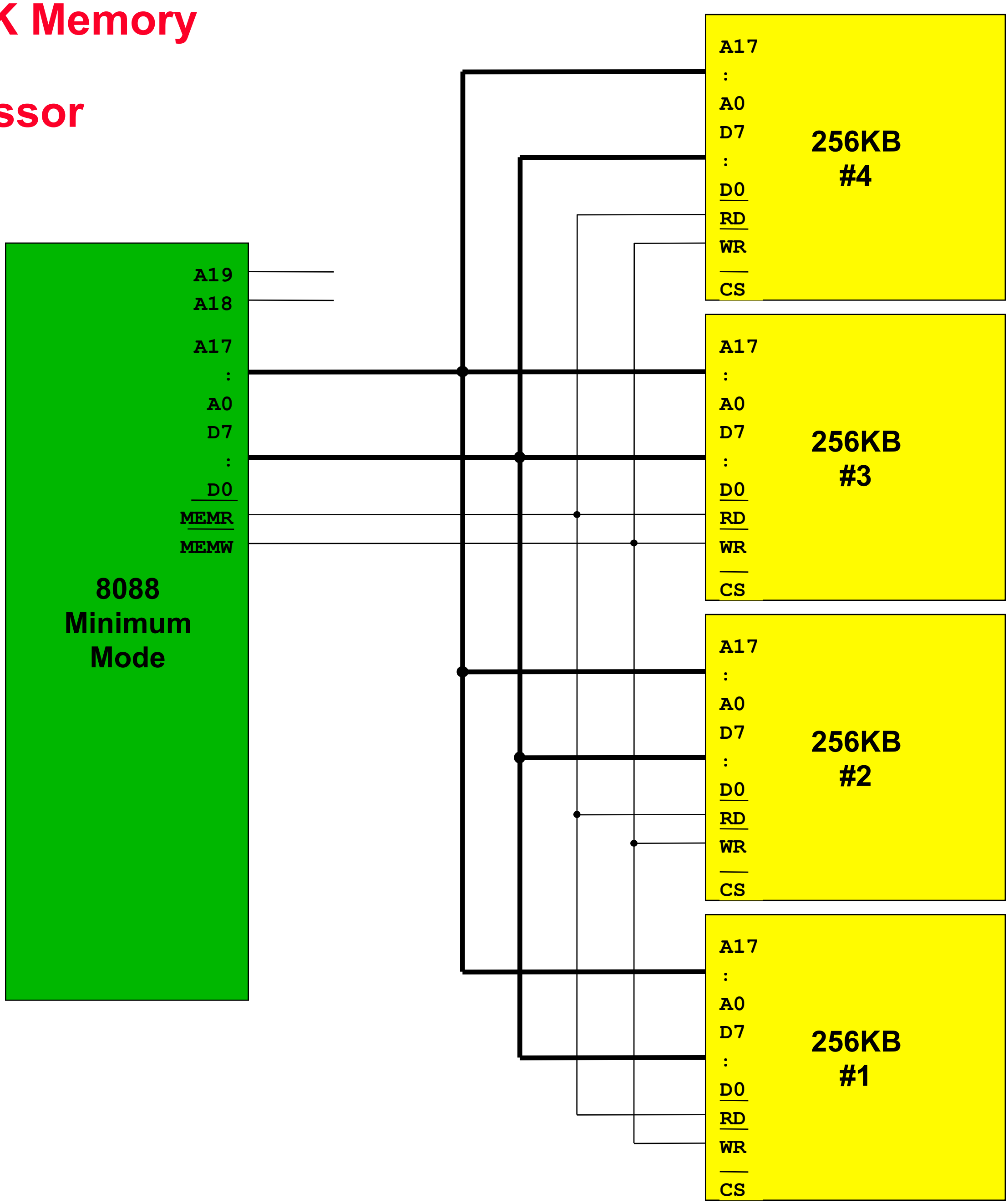
Interfacing one 512K Memory Chips to the 8088 Microprocessor (version 2)



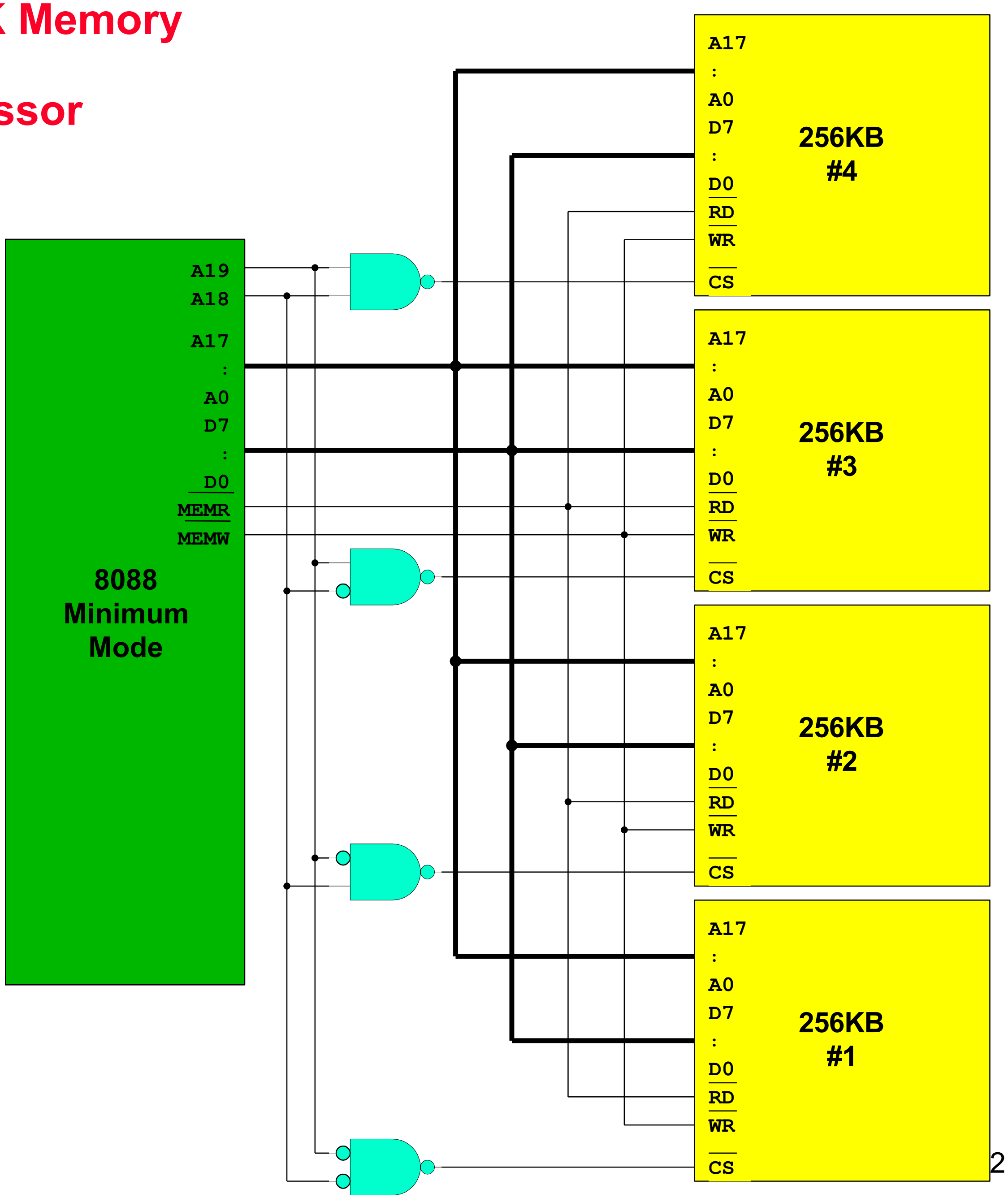
Interfacing one 512K Memory Chips to the 8088 Microprocessor (version 3)



Interfacing Four 256K Memory Chips to the 8088 Microprocessor



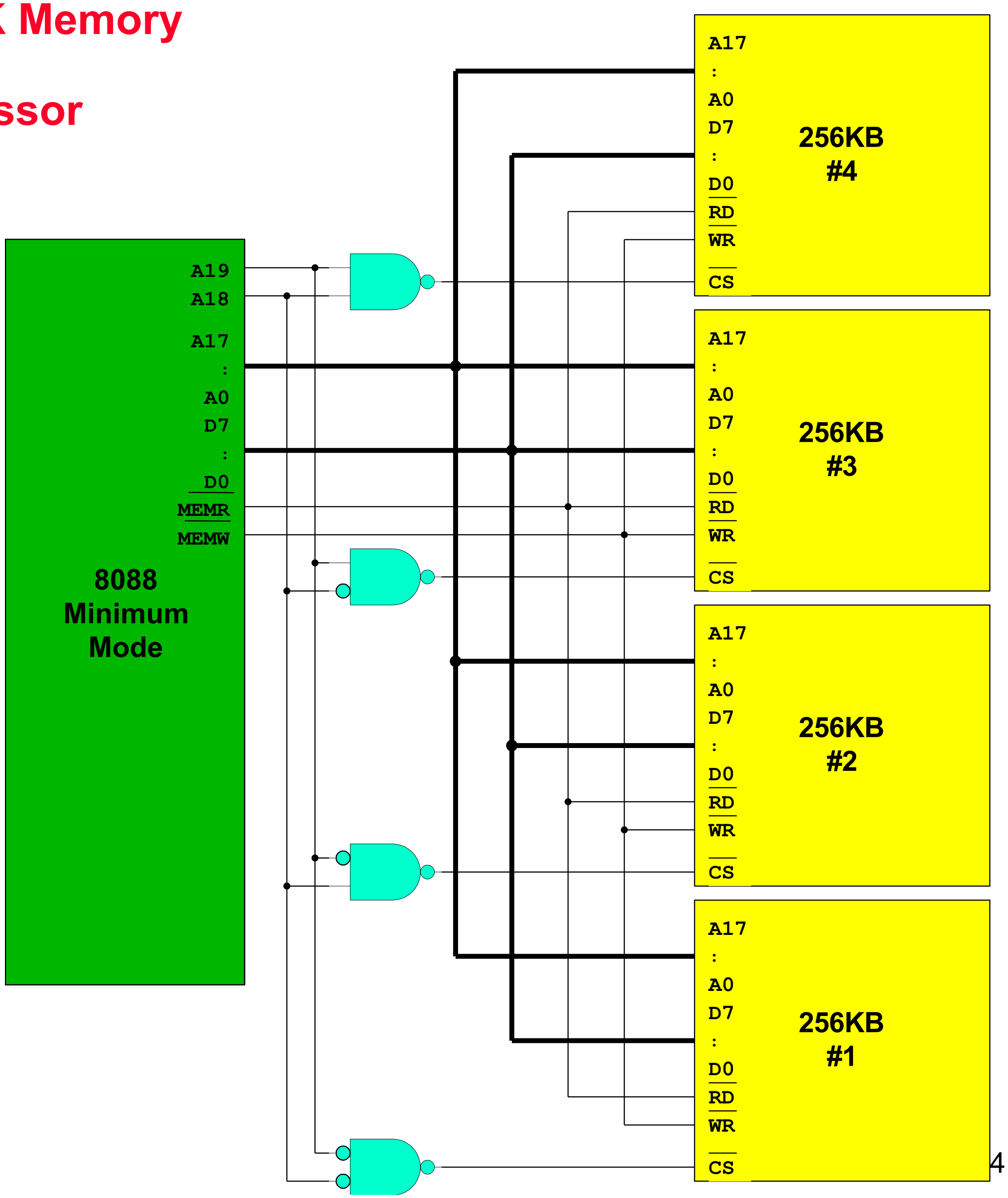
Interfacing four 256K Memory Chips to the 8088 Microprocessor



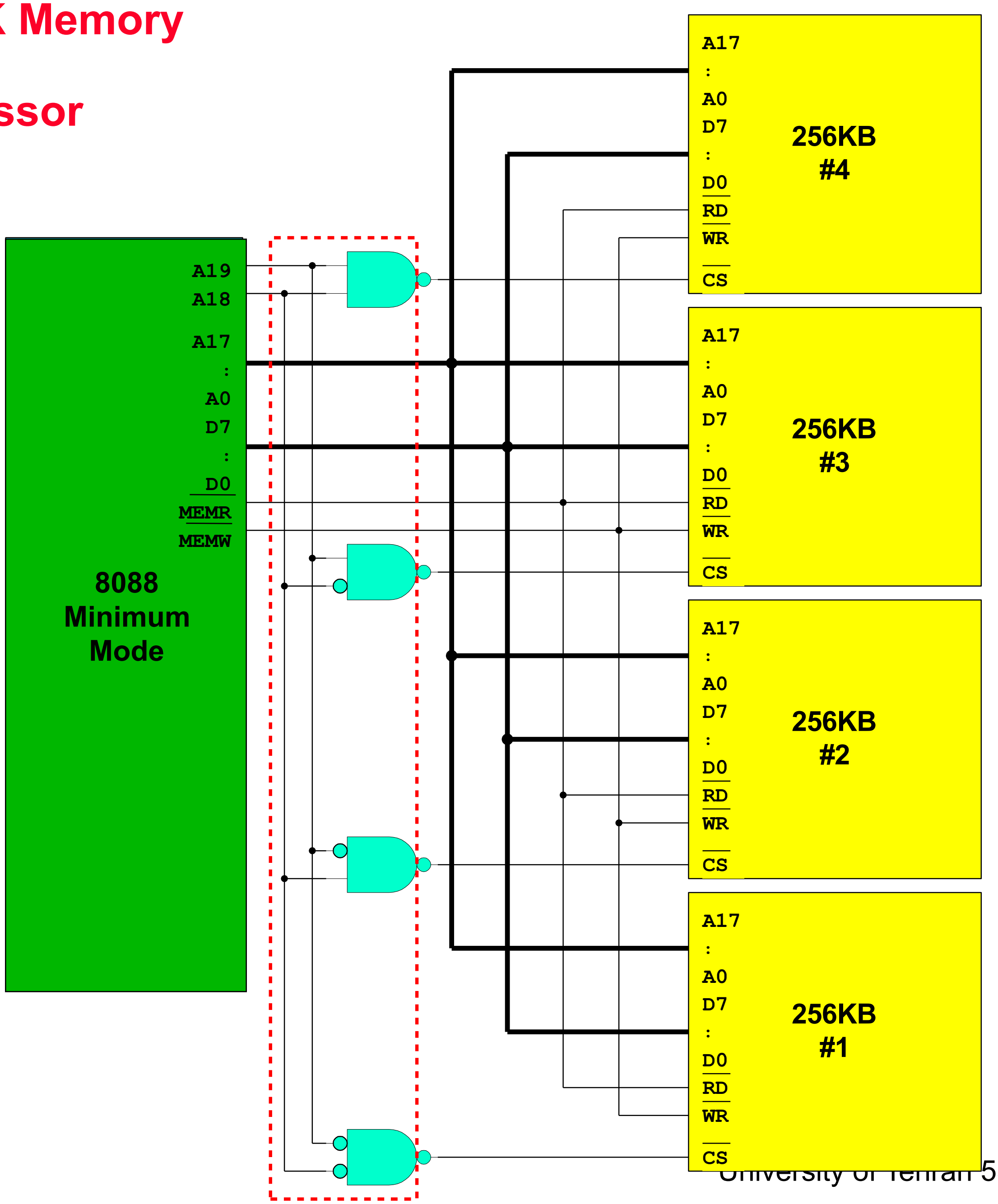
Memory chip#__ is mapped to:

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
-----	----	----	----	----	----
-----	----	----	----	----	----

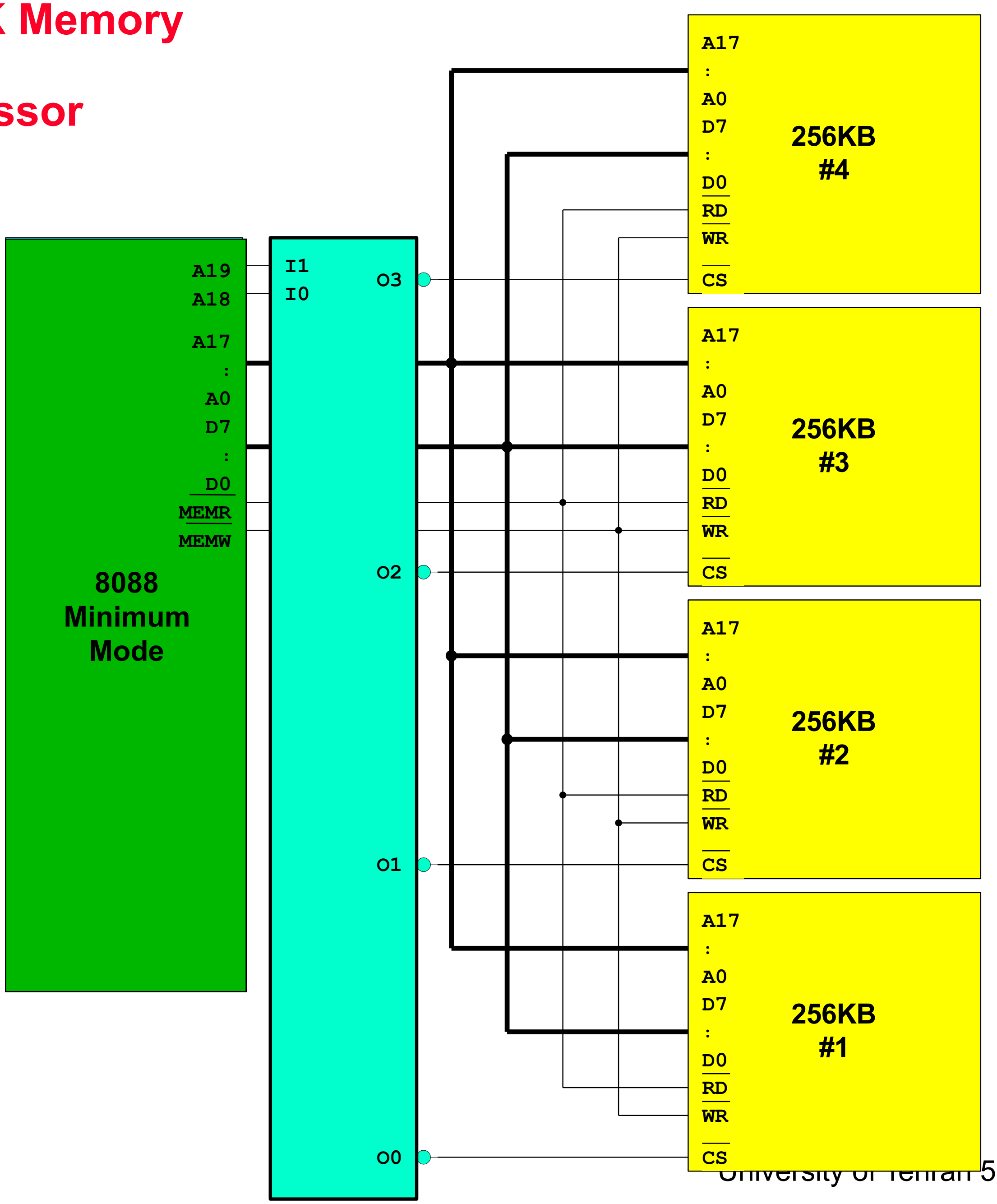
Interfacing four 256K Memory Chips to the 8088 Microprocessor



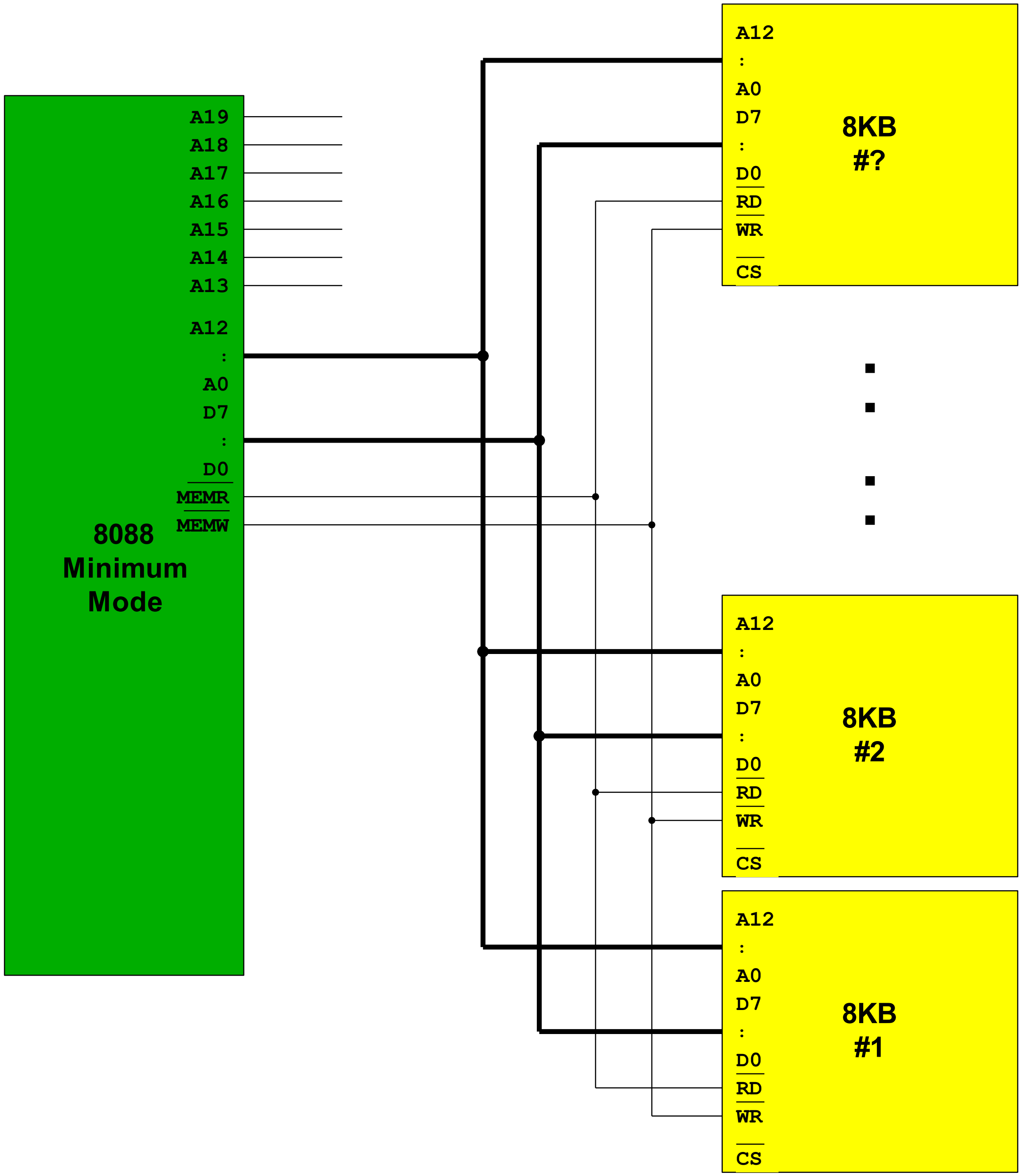
Interfacing four 256K Memory Chips to the 8088 Microprocessor



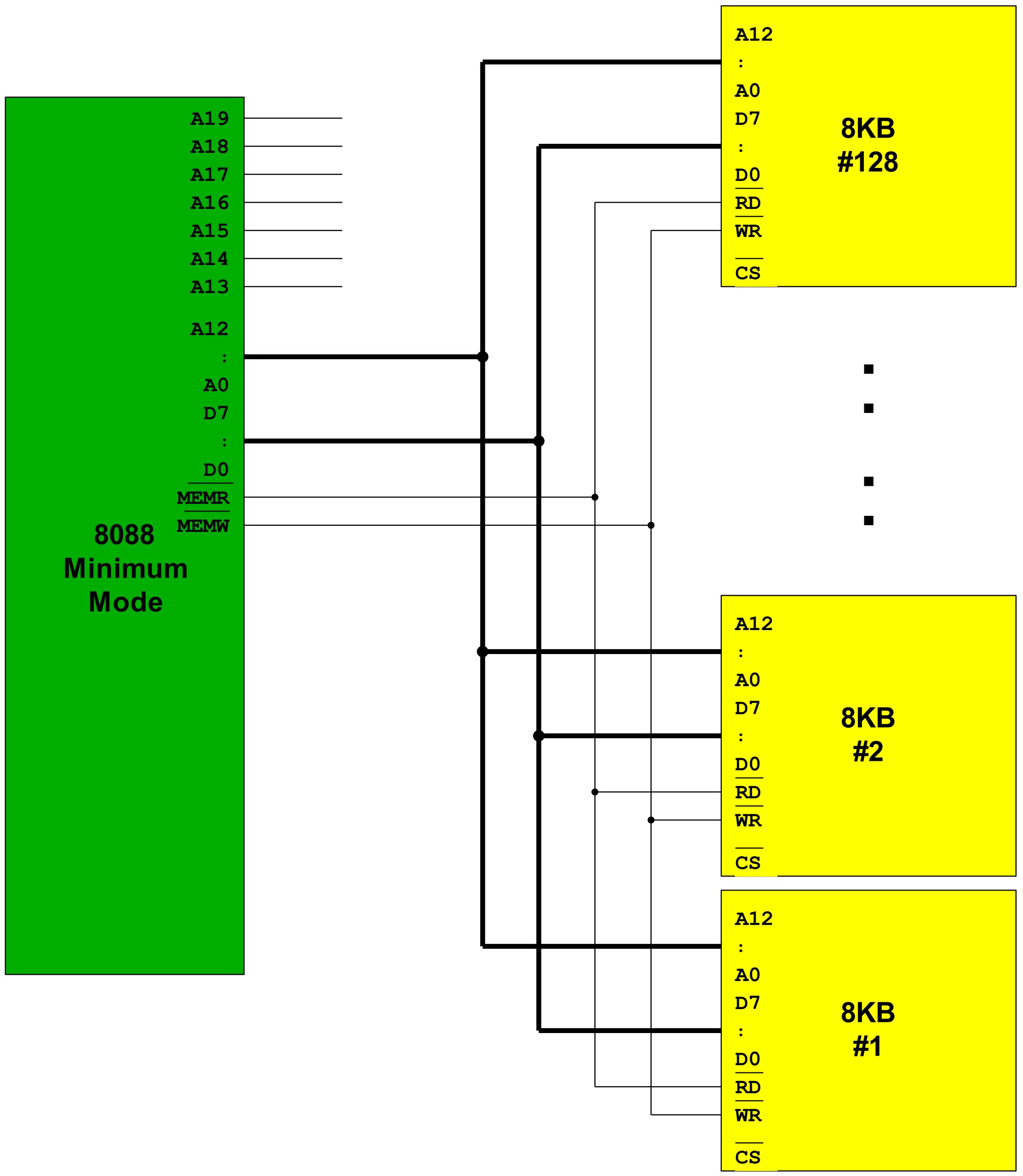
Interfacing four 256K Memory Chips to the 8088 Microprocessor



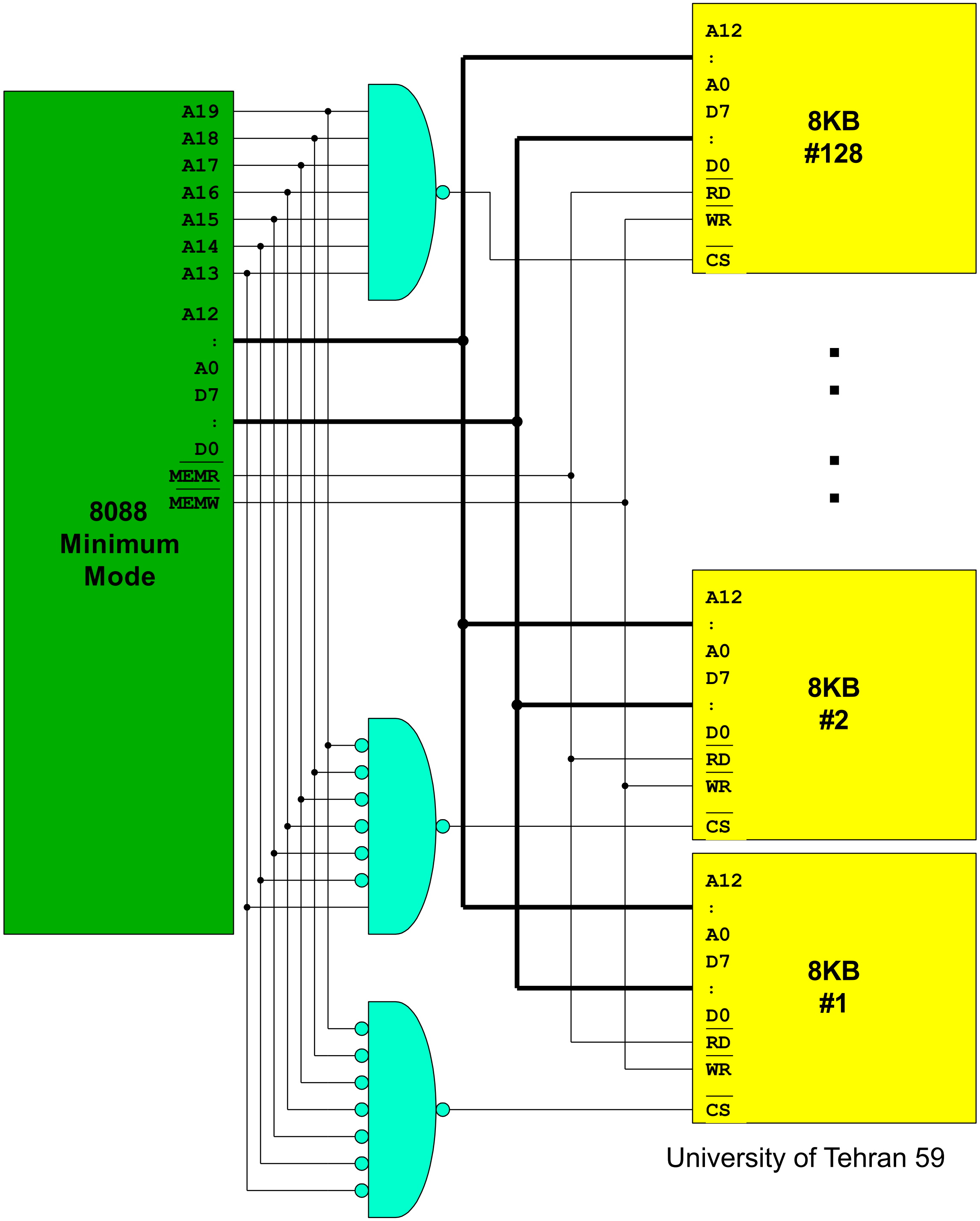
Interfacing several 8K Memory Chips to the 8088 µP



Interfacing 128 8K Memory Chips to the 8088 µP



Interfacing 128 8K Memory Chips to the 8088 μP



Memory chip#__ is mapped to:

A19 to A0 (HEX)	AAAA 1111 9876	AAAA 1111 5432	AAAA 1198 1000	AAAA 7654	AAAA 3210
-----	----	----	----	----	----
-----	----	----	----	----	----

Memory Terms

- Capacity
  - Kbit, Mbit, Gbit
- Organization
  - Address lines
  - Data lines
- Speed / Timing
  - Access time
- Write ability
  - ROM
  - RAM

ROM Variations

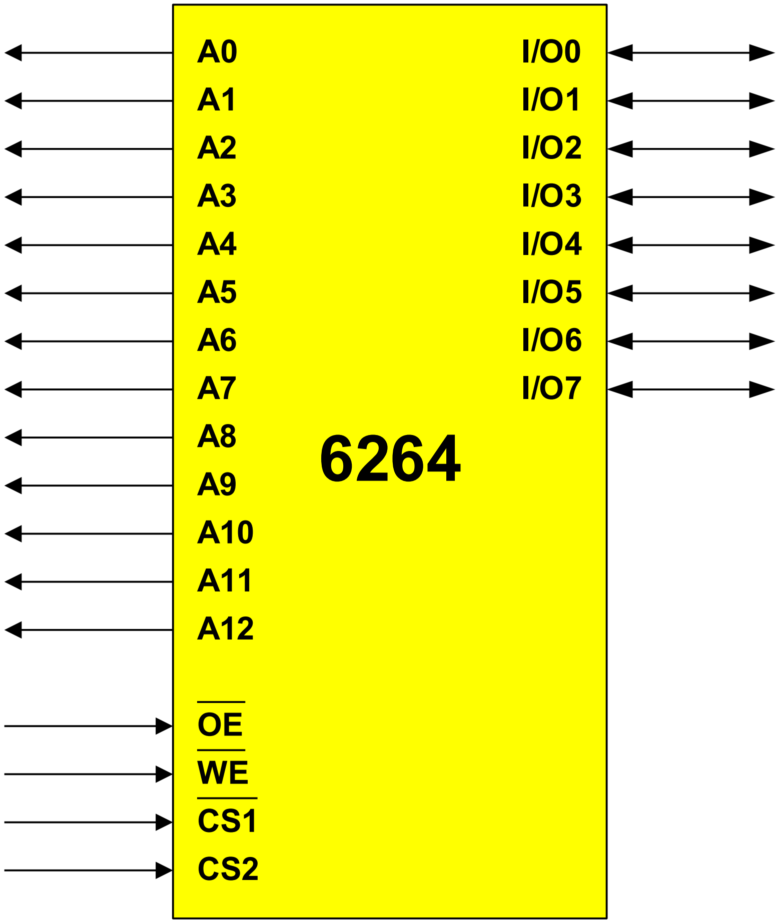
- Mask Rom
- PROM – OTP
- EPROM – UV_EPROM
- EEPROM
- Flash memory

RAM Variations

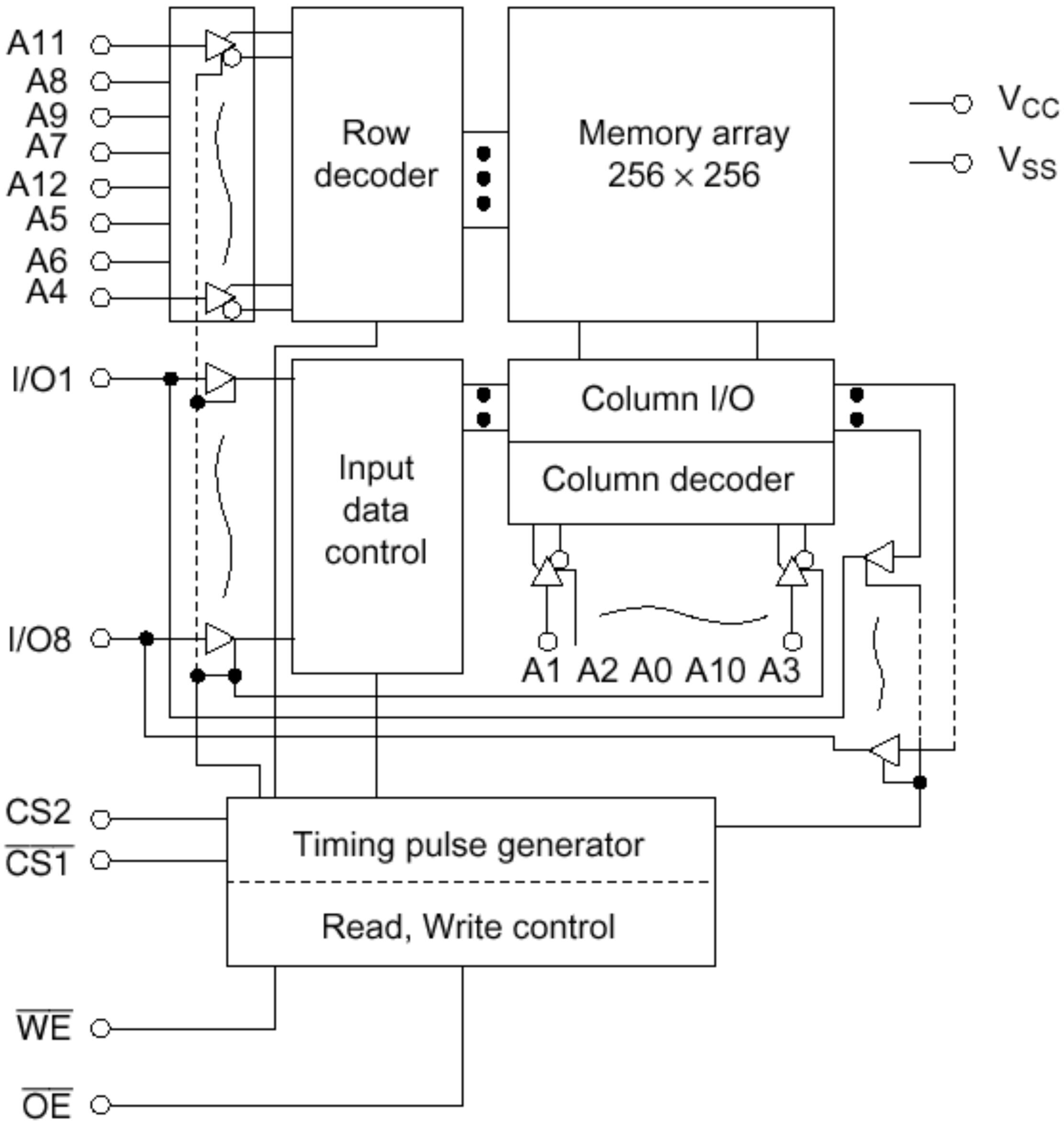
- SRAM
- DRAM
- NV-RAM
  - SRAM – CMOS
  - Internal lithium battery
  - Control circuitry to monitor Vcc

Memory Chip

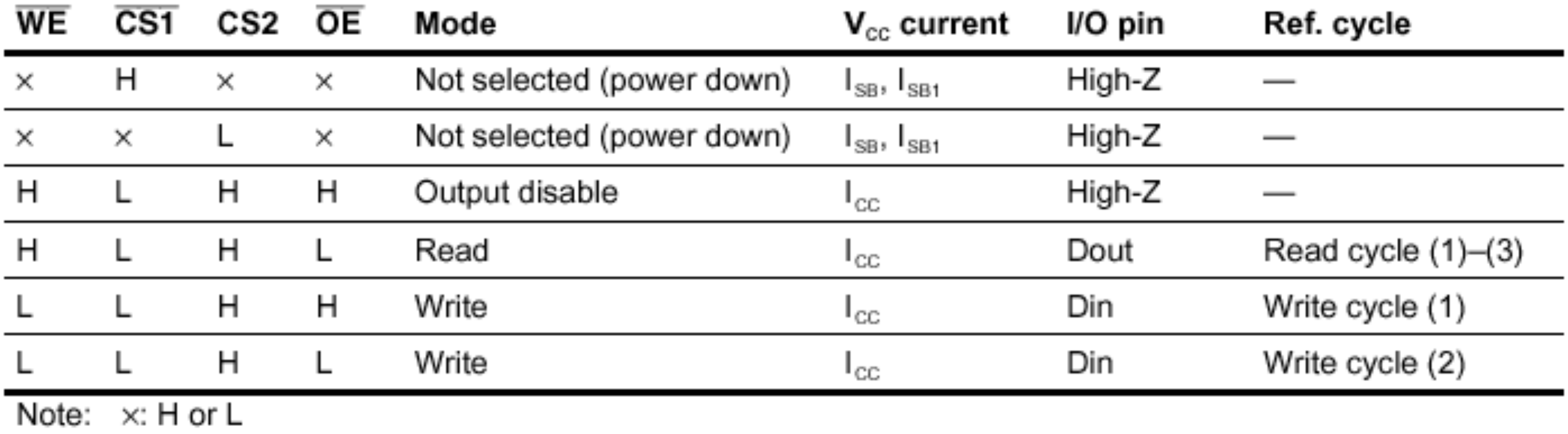
- 8K SRAM
- to be specific:
  – 8Kx8 bits SRAM



6264 Block Diagram



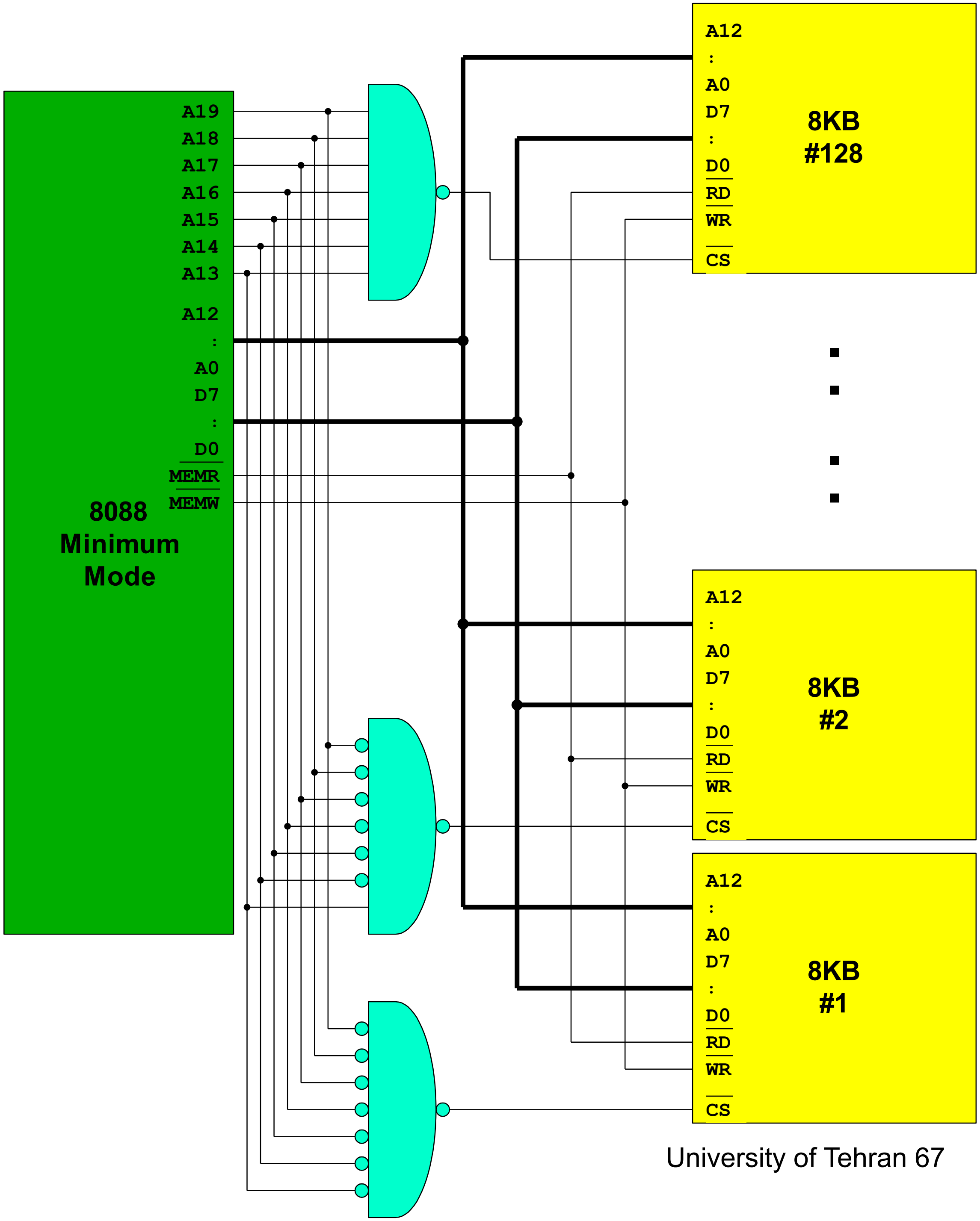
6264 Function Table



$\overline{WE}$	$\overline{CS1}$	CS2	$\overline{OE}$	Mode	V_{CC} current	I/O pin	Ref. cycle
×	H	×	×	Not selected (power down)	I_{SB}, I_{SB1}	High-Z	—
×	×	L	×	Not selected (power down)	I_{SB}, I_{SB1}	High-Z	—
H	L	H	H	Output disable	I_{CC}	High-Z	—
H	L	H	L	Read	I_{CC}	Dout	Read cycle (1)–(3)
L	L	H	H	Write	I_{CC}	Din	Write cycle (1)
L	L	H	L	Write	I_{CC}	Din	Write cycle (2)

Note: ×: H or L

Interfacing 128 8K Memory Chips to the 8088 µP



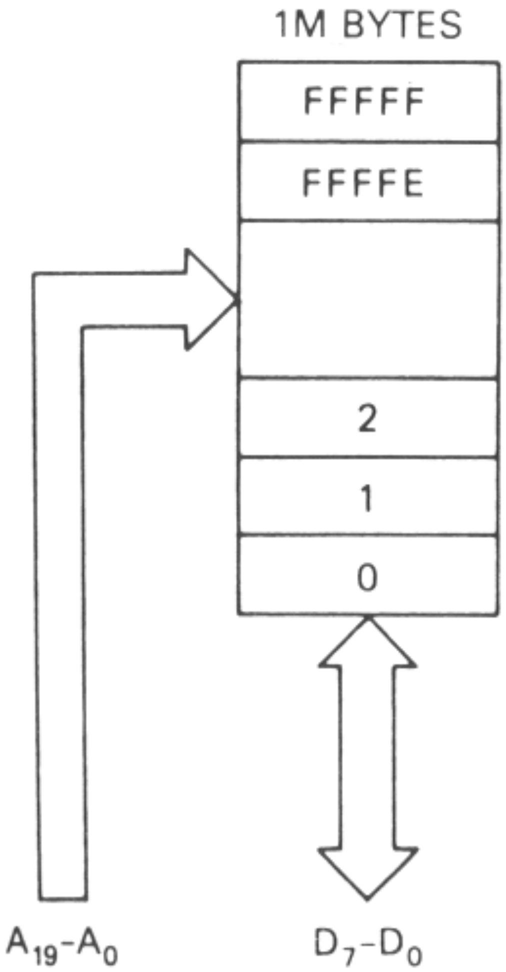
8086 Machine Cycles

- **The key difference between the 8088 and the 8086 is the size of the data bus:**
  - **8-bit for 8088**
  - **16-bit for 8086**

- **Thus the 8086 can transfer 2 bytes of instruction code or data in each machine cycle.**

8088 Memory Organization

Physical Implementation



Memory in the 8088 is stored in one (logical) bank. This bank might consist of several ICs.

All 20 address lines select the appropriate byte, either directly to the chip, or indirectly through chip select circuitry. Unused lines lead to mirror images.

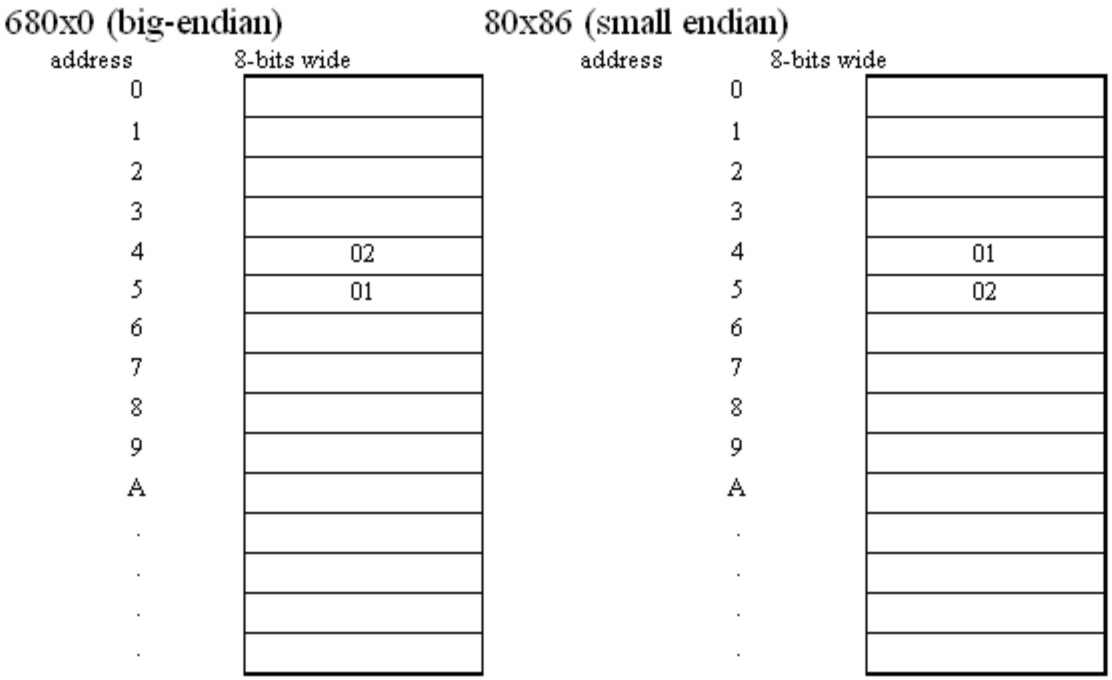
Little Endian / Big Endian

for the 68000:

```
MOVE.W      #513, D0        ; move value 513 into the lower 16 bits of D0
MOVE.W      D0,4            ; store the lower word of D0 into memory 4
```

for the 80x86:

```
MOV   AX,513                ; load AX (16 bits), with the value 513
MOV   [4],AX               ; store AX into memory 4
```



680x0 (big-endian)

address	8-bits wide
0	
1	
2	
3	
4	02
5	01
6	
7	
8	
9	
A	
.	
.	
.	
.	

80x86 (small endian)

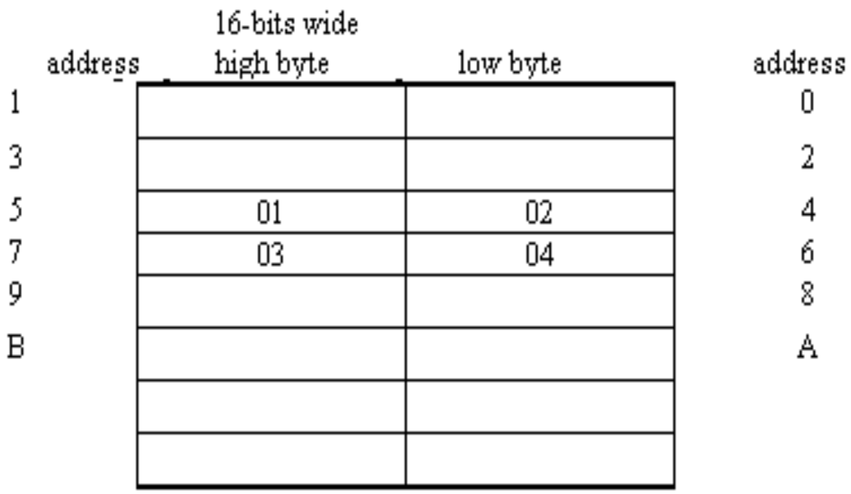
address	8-bits wide
0	
1	
2	
3	
4	01
5	02
6	
7	
8	
9	
A	
.	
.	
.	
.	

80x86 family

- 16-bit Processors
  - 8088 (8-bit data / 20-bit address)
  - 8086/186 (16-bit data / 20-bit address)
  - 80286 (16-bit data / 24-bit address)
- 32-bit Processors
  - 80386 (16/24 or 32/32 common)
  - 80486 (32/32), Pentium, PII (64/32)
  - Pentium Pro, II, III, IV (64/36)
  - PPC 60x (32 or 64/32)
- All 80x86 processors use a 16-bit address for i/o

Memory Alignment in 16-bit Micro

- We have 16-bit data bus
- Why not use it for memory access.
- 1M byte of memory is organized as:
- 512K * 16 bit
- The memory is word-aligned
- Access to even addresses is aligned and simple
- Example: 0102H and 0304H stored in [4H]



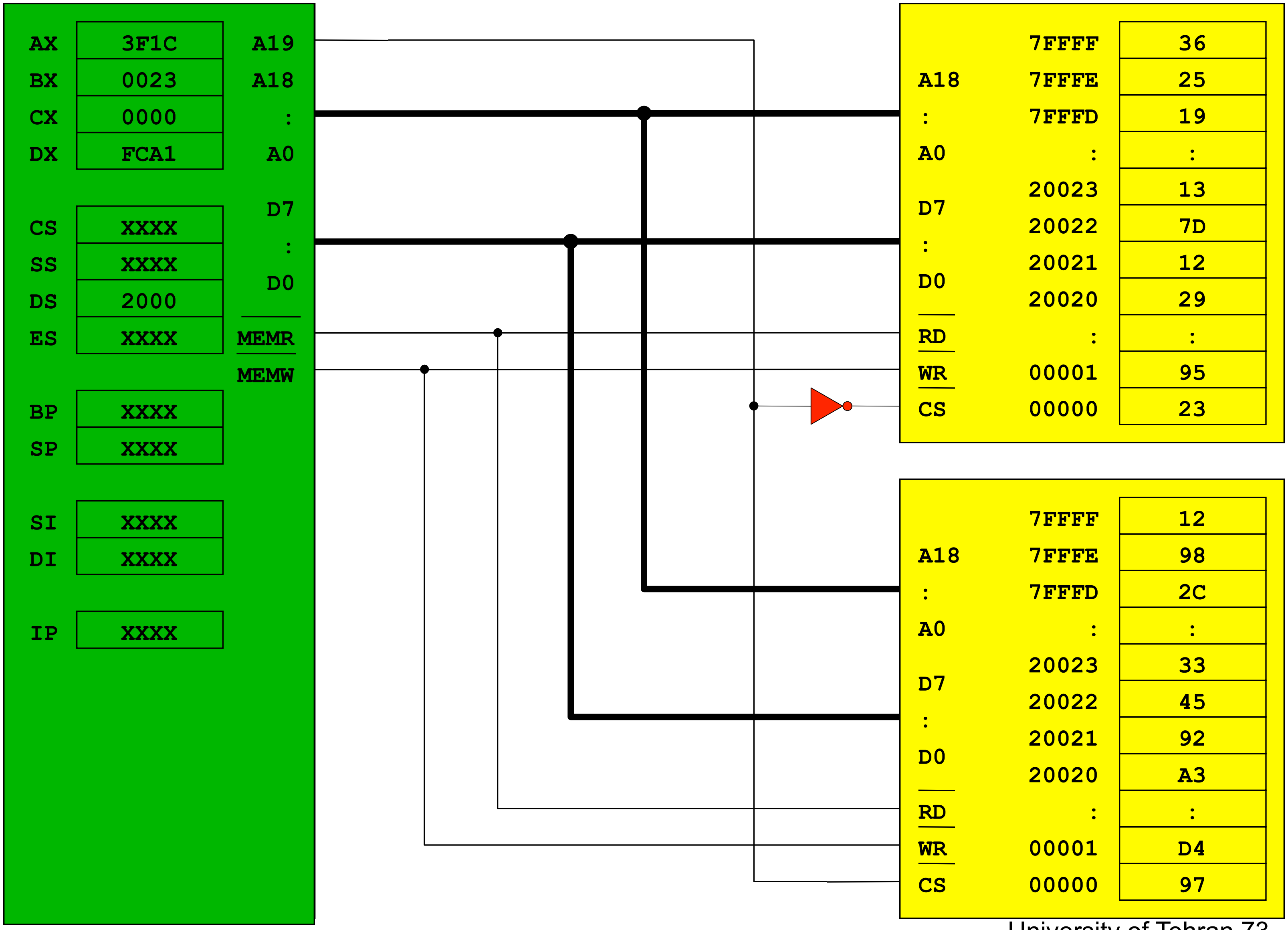
	16-bits wide		
address	high byte	low byte	address
1			0
3			2
5	01	02	4
7	03	04	6
9			8
B			A

What happens on mov AX,[4]?

What happens on mov AX,[5]?

Motorola family of the MC680x0 forbids non-aligned access

Interfacing two 512KB Memory to the 8088 Microprocessor (review)



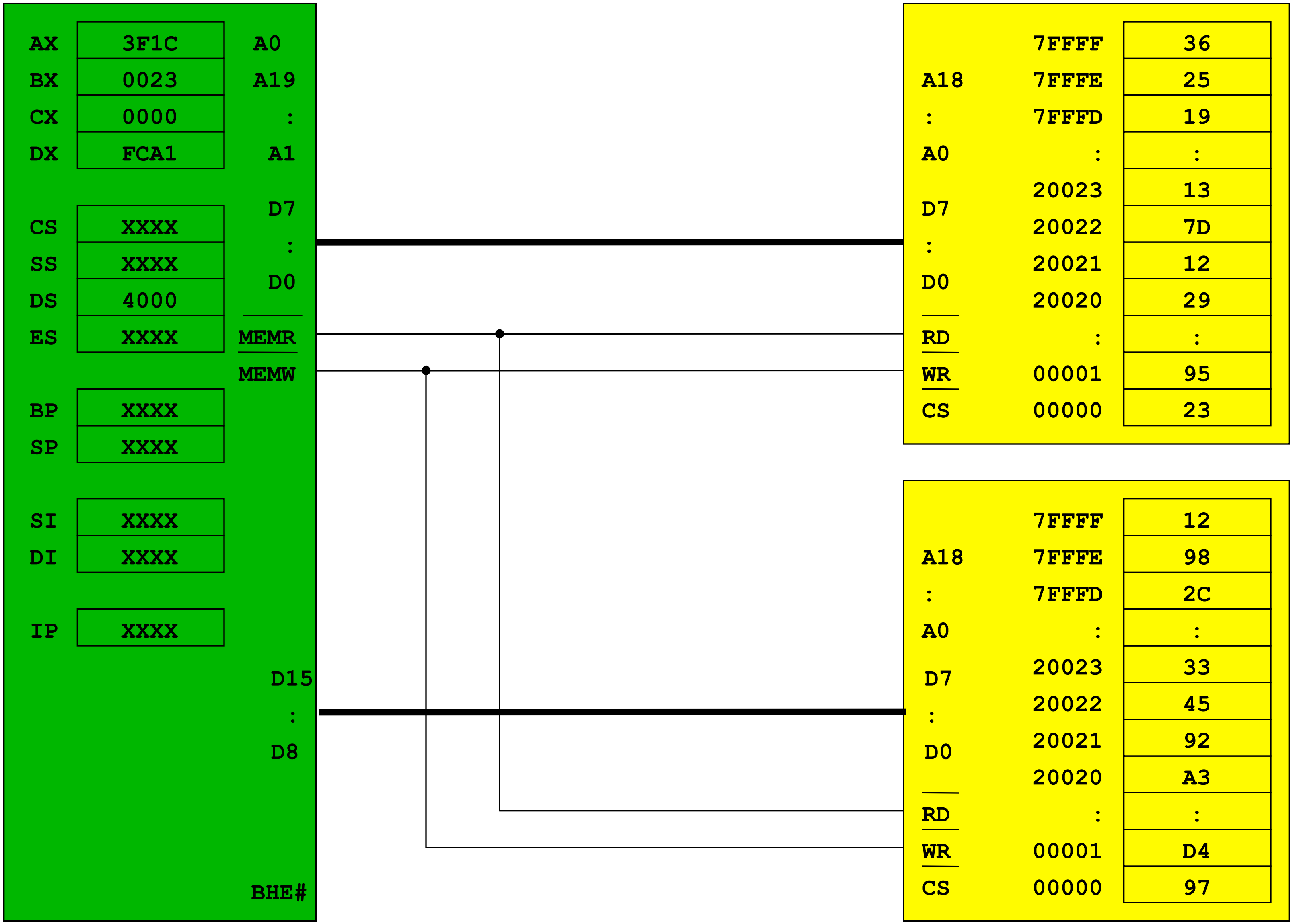
AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	2000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

Interfacing two 512KB Memory to the 8086 Microprocessor

How to connect data lines?
How to connect address lines?



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	4000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	29
:	:
00001	95
00000	23

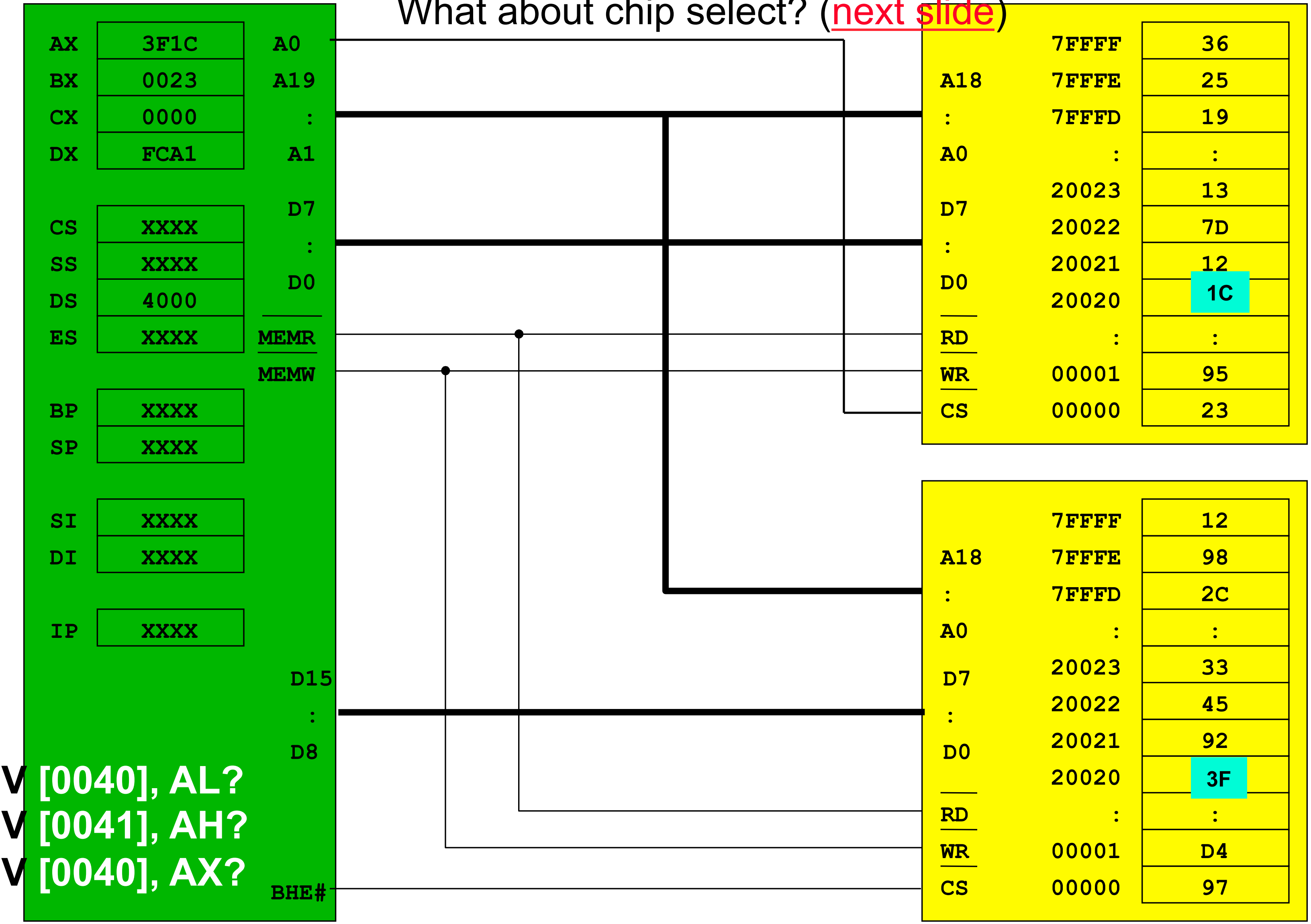
7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	A3
:	:
00001	D4
00000	97

Address lines in 8086

- Byte addressability should be considered.
- Bytes in address 0 and 1 form a word.
- This word could be accessed in one memory cycle.
- Every even address (byte) with the corresponding odd address (byte) form a 16-bit word.
- Address 0 and 1 from processor map to word 0 (row 0). Address 2 , 3 map to word 1. …
- Address 2n and 2n+1 map to word n.

Interfacing two 512KB Memory to the 8086 Microprocessor

How to connect address lines?
What about chip select? (next slide)



AX	3F1C
BX	0023
CX	0000
DX	FCA1
CS	XXXX
SS	XXXX
DS	4000
ES	XXXX
BP	XXXX
SP	XXXX
SI	XXXX
DI	XXXX
IP	XXXX

7FFFF	36
7FFFE	25
7FFFD	19
:	:
20023	13
20022	7D
20021	12
20020	1C
:	:
00001	95
00000	23

7FFFF	12
7FFFE	98
7FFFD	2C
:	:
20023	33
20022	45
20021	92
20020	3F
:	:
00001	D4
00000	97

MOV [0040], AL?
MOV [0041], AH?
MOV [0040], AX?

Memory Bank Select

- **8086/186/286/386SX has 16 Data Lines D15-D0**
- **6264 Only has 8 I/O7 – I/O0**
- **Must Use a “Memory Bank”**
  - **1 SRAM for Storing Bytes with “Even Addresses” (… 0 2)**
  - **1 SRAM for Storing Bytes with “Odd” Addresses” (… 1 3)**
- **8086 has BHE Control Signal – (Bank High Enable)**
- **Can Use Combination of A0 and BHE to Determine Type of Access**

BHE	A0	Access Type
0	0	1 word (16-bits)
0	1	Odd Byte (D15-D8)
1	0	Even Byte (D7-D0)
1	1	No Access

BHE - Bus High Enable

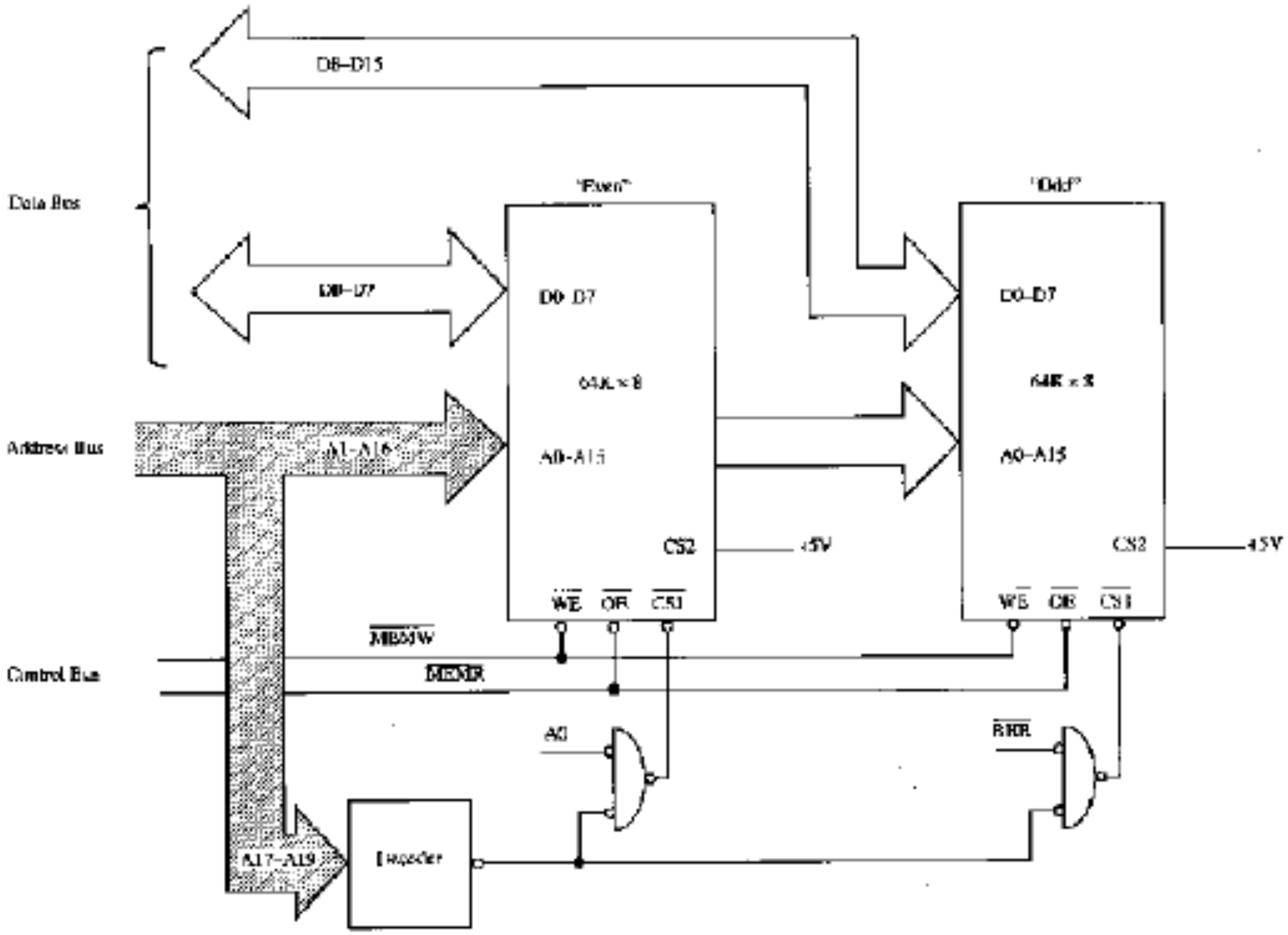
- The BHE pin (#34) is the only physical difference between the 8088 and the 8086.
- It is active low, which corresponds to most of the enable lines of memory devices (they are usually active low).

Physical Selection

The A_0 line and the BHE pin determine which bank is selected.

$\overline{\text{BHE}}$	A_0	Memory Access	
0	0	Word	D_0-D_{15}
0	1	Odd byte	D_8-D_{15}
1	0	Even byte	D_0-D_7
1	1	NONE	

Decoding Circuit with Bank Select



8086 Memory Organization

Physical Implementation (Chapter 3.2)

	16-bits wide		
address	high byte	low byte	address
1			0
3			2
5	01	02	4
7	03	04	6
9			8
B			A

In 8086-based systems, memory is organized into 2 banks. Lines D_0-D_7 are connected to one, and lines D_8-D_{15} to the other.

The A_0 and BHE lines select which bank to enable.

Both banks have identical address decoding circuits.

Dual Memory Banks

- **One consequence of the dual memory bank is that a** word **of data/instructions can only be loaded in one machine cycle if it starts on an even address.**
  - **This affects how you store data in the computer**
  - **Choosing a wrong address could lead to code that is almost 50% slower!**
- **8-bit wide memory must be added in equal sized pairs.**

Memory Alignment in 16-bit Micro (Summary)

- We have 16-bit data bus
- Why not use it for memory access.
- 1M byte of memory is organized as:
- 512K * 16 bit
- The memory is word-aligned
- Access to even addresses is aligned and simple
- Example: 0102H and 0304H stored in [4H]

	16-bits wide		
address	high byte	low byte	address
1			0
3			2
5	01	02	4
7	03	04	6
9			8
B			A

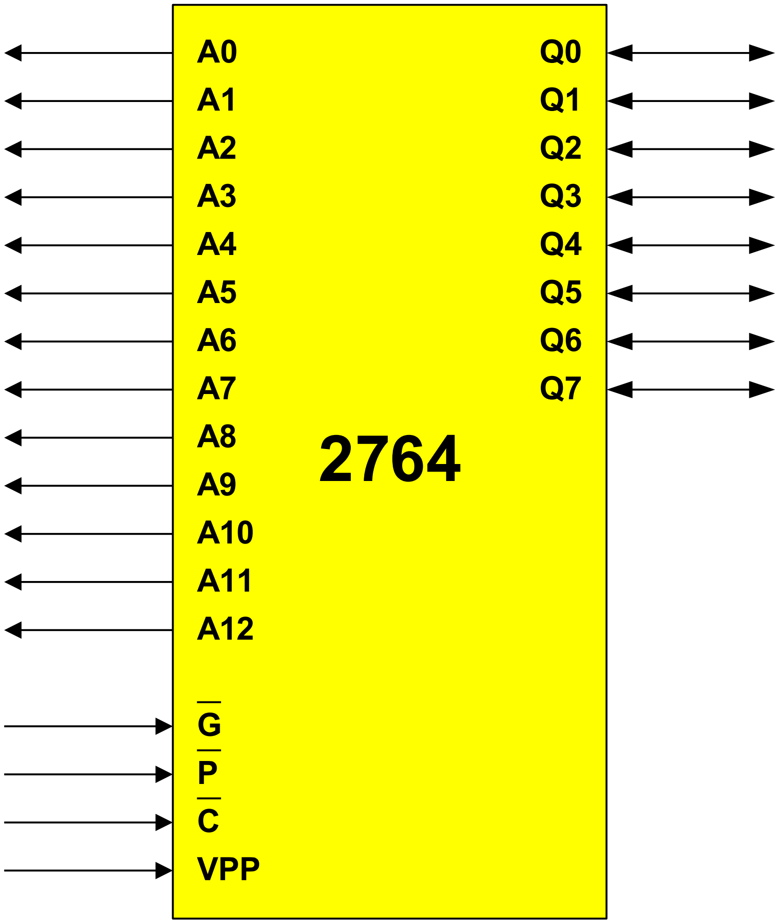
What happens on mov AX,[4]?

What happens on mov AX,[5]?

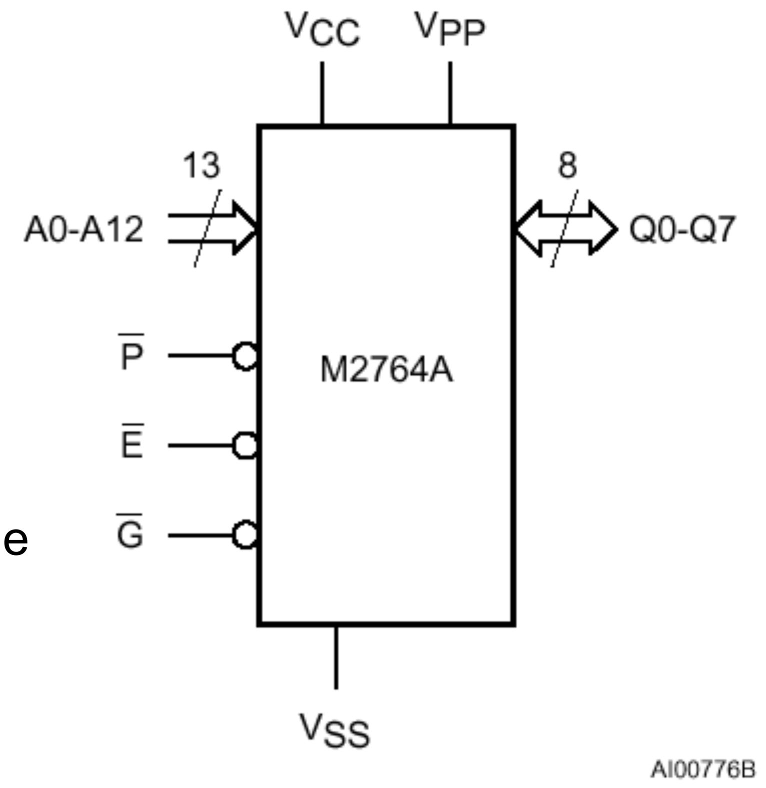
Motorola family of the MC680x0 forbids non-aligned access

Memory Chip

- 8K EPROM
- to be specific:
  - 8Kx8 bits EPROM



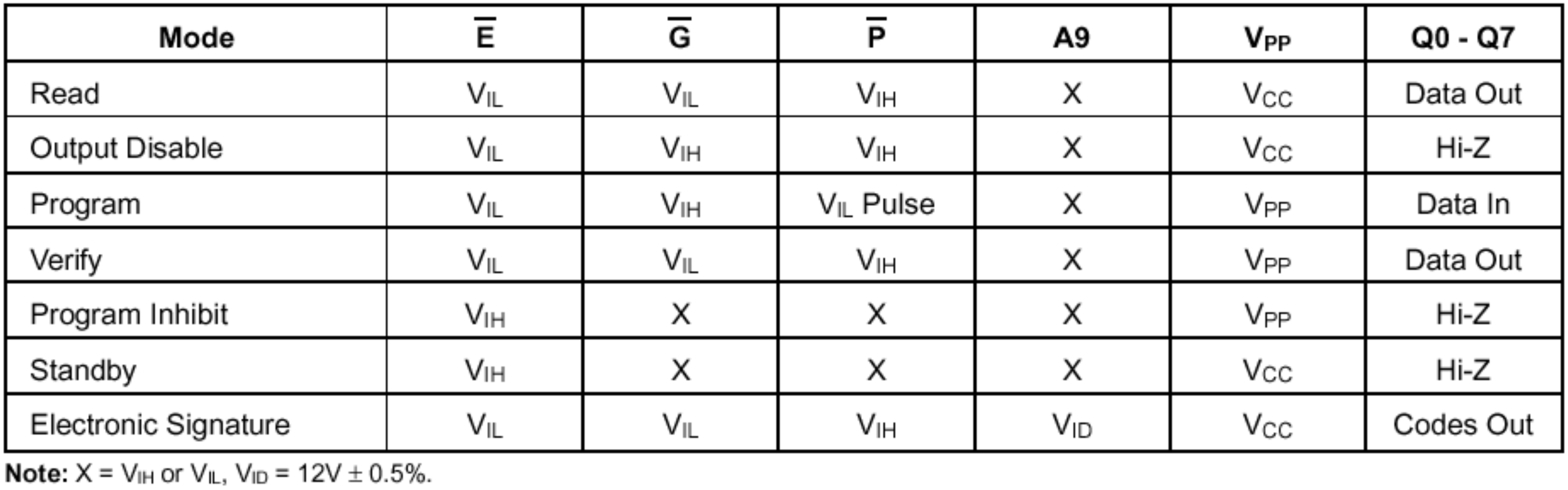
2764 Block Diagram



Chip enable

Output enable

Operating Modes



Mode	$\overline{E}$	$\overline{G}$	$\overline{P}$	A9	V_{PP}	Q0 - Q7
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	Data Out
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	Hi-Z
Program	V_{IL}	V_{IH}	V_{IL} Pulse	X	V_{PP}	Data In
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	Data Out
Program Inhibit	V_{IH}	X	X	X	V_{PP}	Hi-Z
Standby	V_{IH}	X	X	X	V_{CC}	Hi-Z
Electronic Signature	V_{IL}	V_{IL}	V_{IH}	V_{ID}	V_{CC}	Codes Out

Note: X = V_{IH} or V_{IL}, V_{ID} = 12V ± 0.5%.

Programming 2764

- after each erasure for UV-EPROM):
  - all bits of the M2764A are in the “1" state.
- The only way to change a “0" to a ”1" is by ultraviolet light erasure.
- Programming mode when:
  - VPP input is at 12.5V
  - E and P are at TTL low.
- The data to the data output pins.
- The levels required for the address and data inputs are TTL.