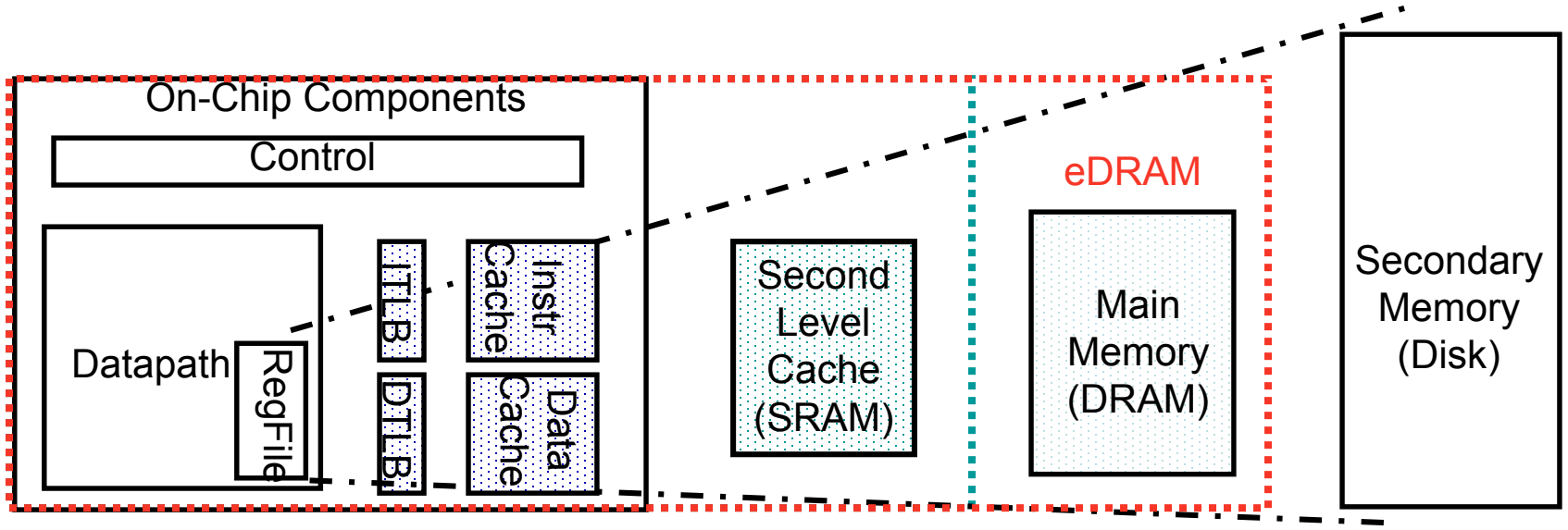
Design with Microprocessors

Year III Computer Sci. English

1-st Semester

Lecture 12: Memory interfacing

Typical Memory Hierarchy [1]



Speed (ns):	.1's	1's	10's	100's	1,000's
Size (bytes):	100's	10K's	100K's … M's	100M's … G's	100G's …T's
Cost/capacity:	highest				lowest

ITLB = Instruction Translation Look-aside Buffer
DTLB = Dual Translation Look-aside Buffer

Translation lookaside buffer (**TLB**) is a cache that memory management hardware uses to improve virtual address translation speed.[1] The majority of CPUs includes one or more TLBs in the memory management hardware, and it is nearly always present in any hardware that utilizes paged or segmented virtual memory. The TLB is sometimes implemented as content-addressable memory (CAM). The CAM search key is the virtual address and the search result is a physical address.

Classification of memories [1]

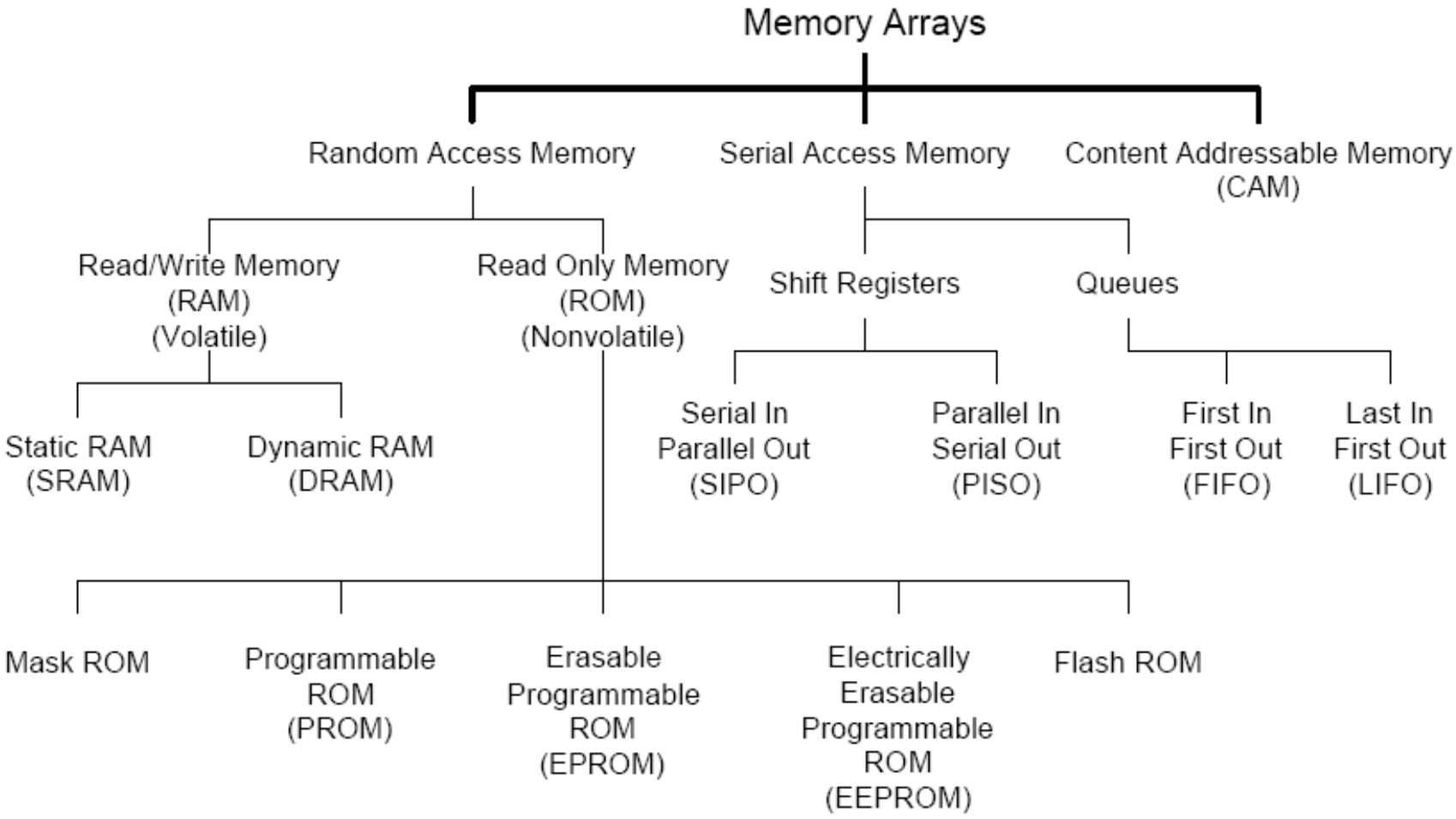
Read Write Memories (RWM)		NVRWM	ROM
Random Access	Non-Random Access	EPROM	Mask-prog. ROM
SRAM (cache, register file)	FIFO, LIFO	EEPROM	
DRAM (main memory)	Shift Register	FLASH	Electrically-prog. PROM

Function – functionality, nature of the storage mechanism

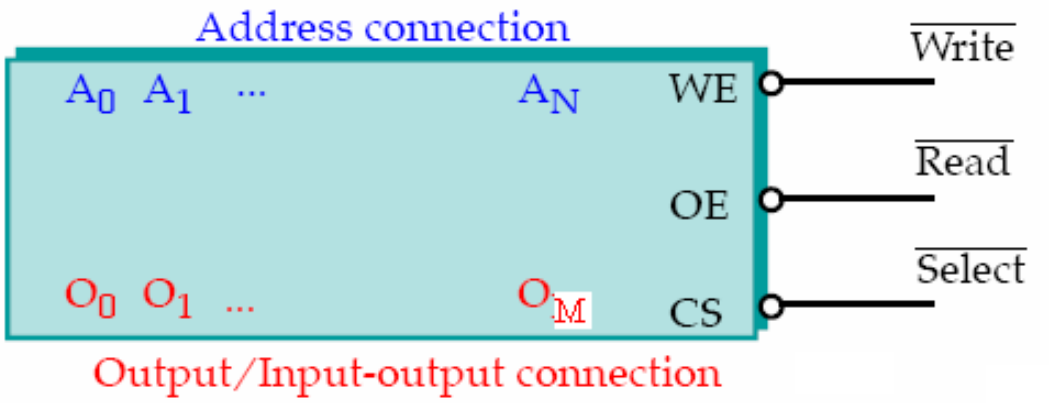
- static and dynamic; volatile and nonvolatile (NV); read only (ROM)

Access pattern – random, serial, content addressable

Classification of memories



Generic pin configuration [2]



Address pins (inputs)

N+1 $\Rightarrow$ no. of memory locations = 2^{N+1}

Data pins

M+1 $\Rightarrow$ size of memory locations

Bidirectional, 3-state output (#OE)

Control pins (inputs)

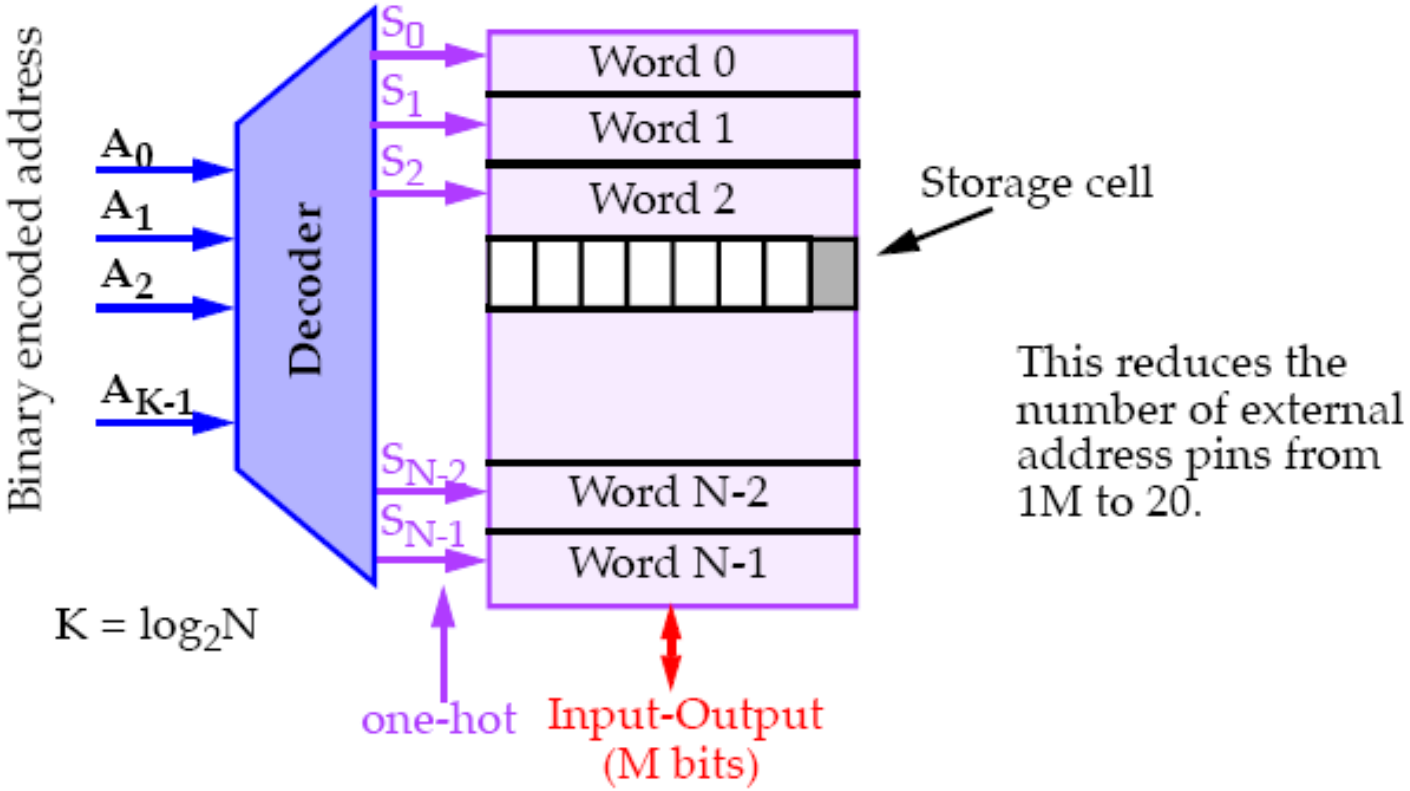
Chip select/enable $\Rightarrow$ enables the device (usually by decoding the base address of the chip)

Read (#OE) /write (#WE) $\Rightarrow$ operation

Memory architecture [2]

Memory Architecture

Add a decoder to solve the package problem:



This does not address the **memory aspect ratio** problem:

The memory is 128,000 time higher than wide ($2^{20}/2^3$) !

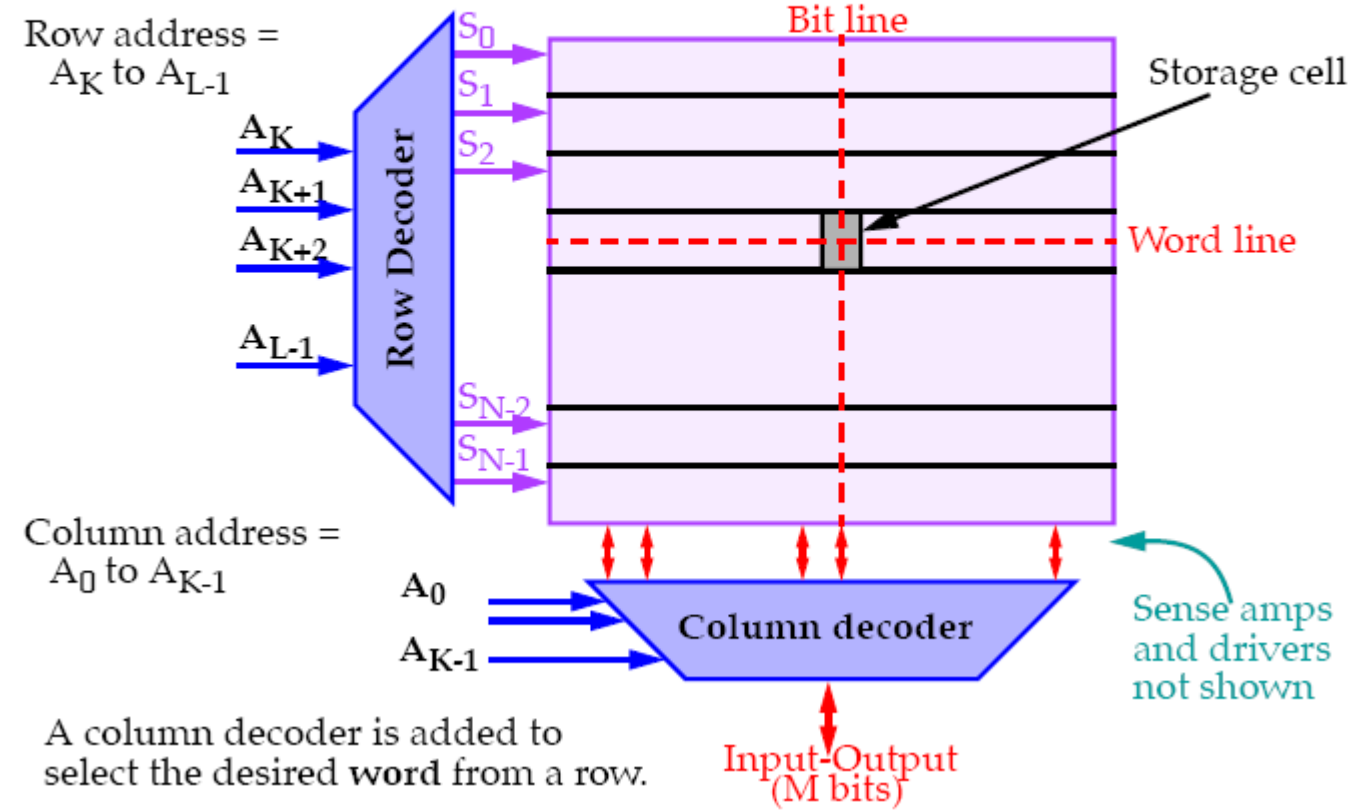
Besides the bizarre shape factor, the design is *extremely slow* since the vertical wires are VERY long (delay is at least linear to length).

Memory architecture [2]

Memory Architecture

The vertical and horizontal dimensions are usually very similar, for an aspect ratio of *unity*.

Multiple words are stored in each row and selected simultaneously:



A column decoder is added to select the desired **word** from a row.

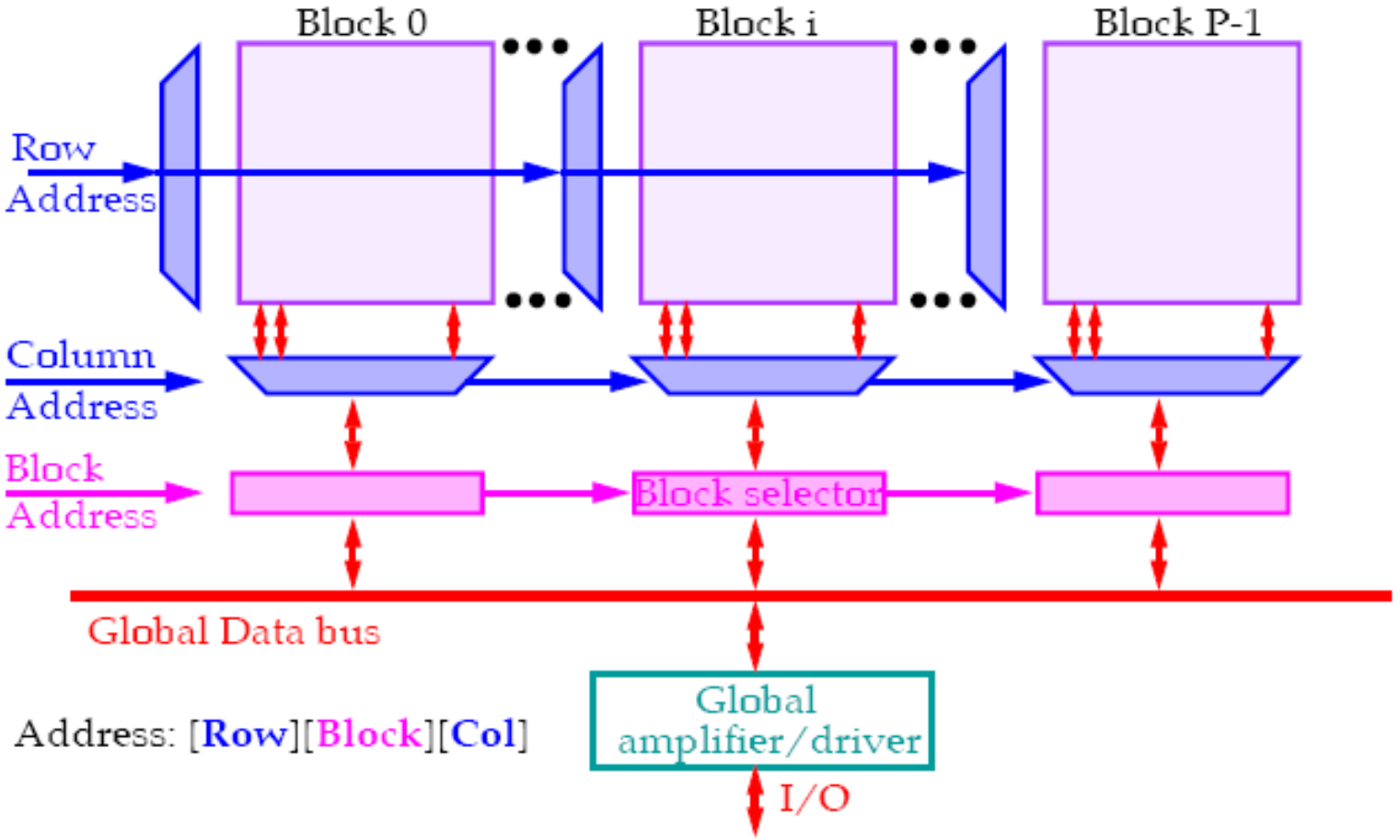
Memory architecture [2]

Memory Architecture

This strategy works well for memories up to 64 Kbits to 256 Kbits.

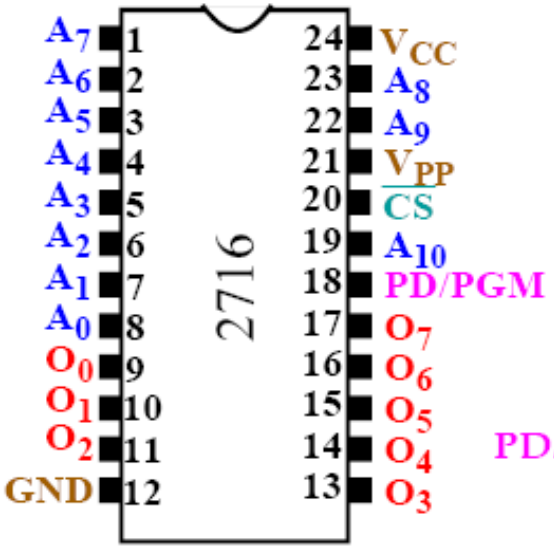
Larger memories start to suffer excess delay along bit and word lines.

A **third dimension** is added to the address space to solve this problem:



ROM/EPROM [2]

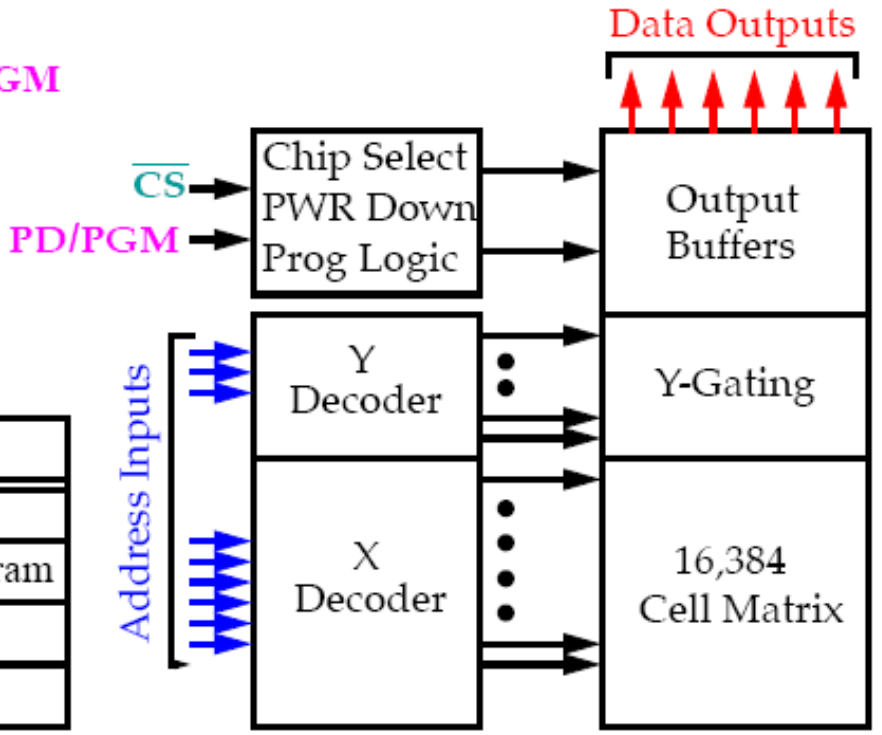
Intel 2716 EPROM (2K X 8):



2K x 8 EPROM

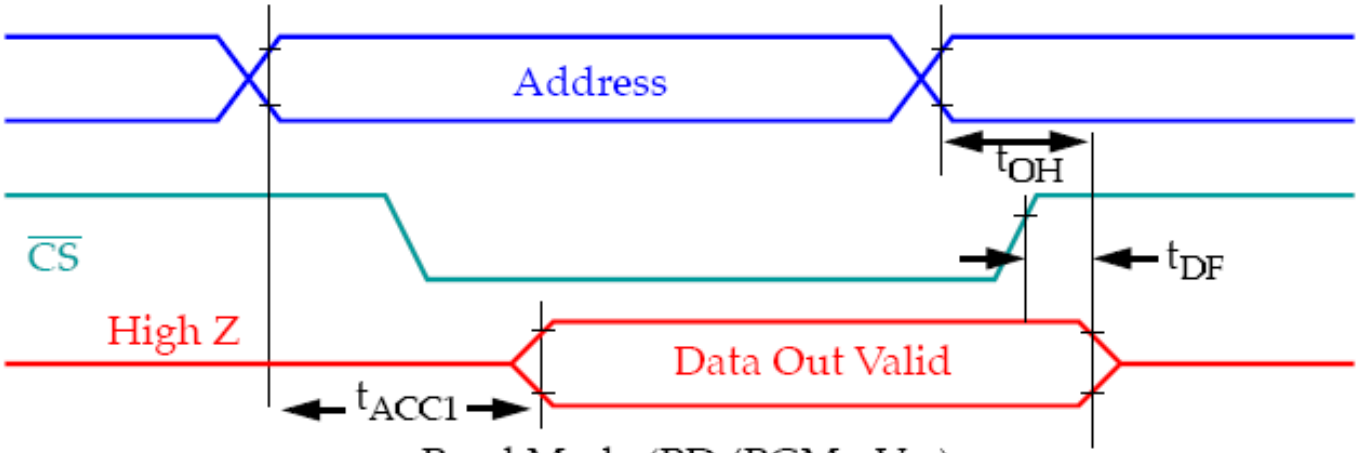
Pin(s)	Function
A_0-A_{10}	Address
PD/PGM	Power down/Program
$\overline{CS}$	Chip Select
O_0-O_7	Outputs

V_{PP} is used to program the device by applying 25V and pulsing PGM while holding $\overline{CS}$ high.



ROM/EPROM [2]

2716 Timing diagram:



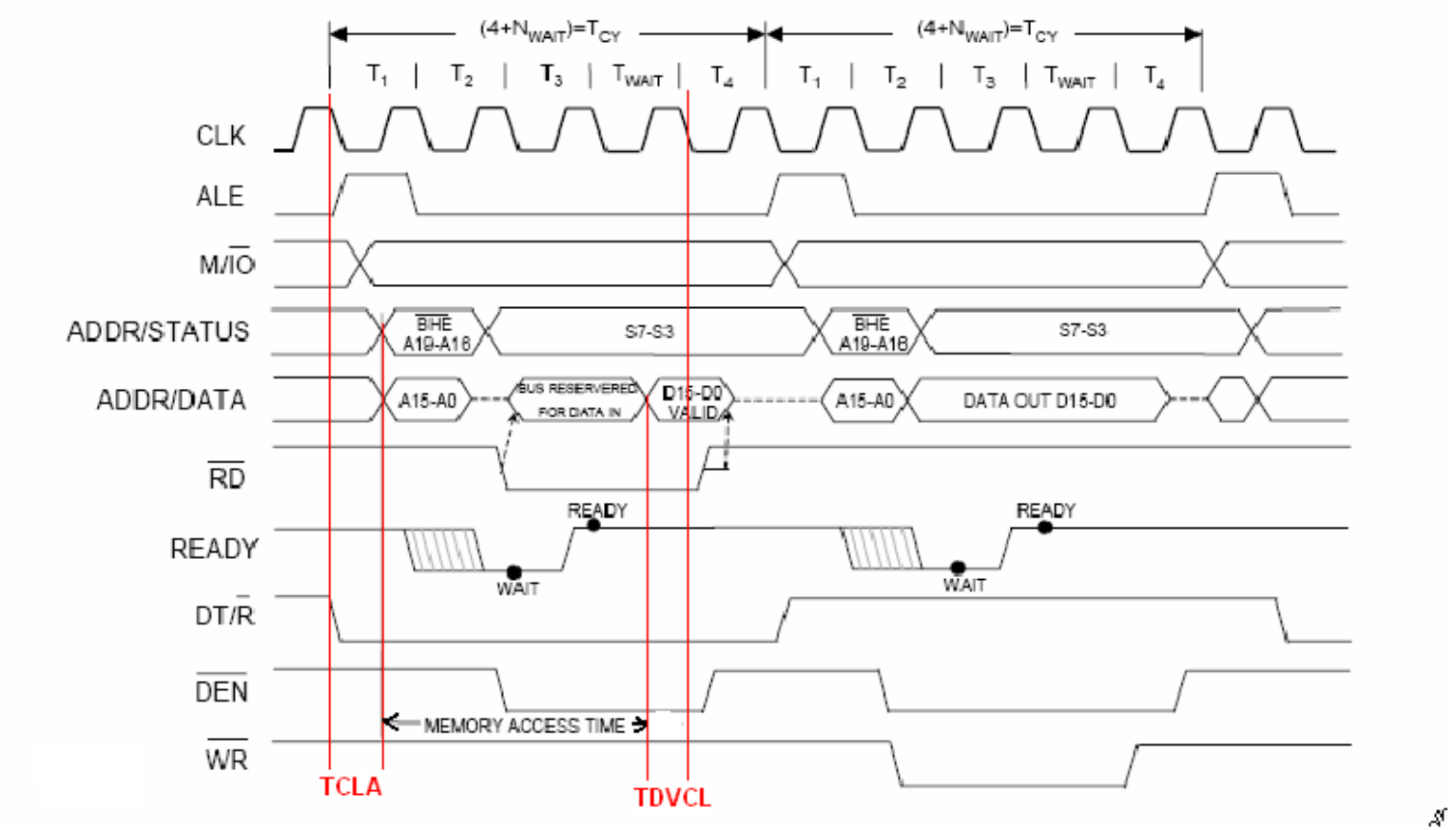
Read Mode (PD/PGM =V_{IL})

Sample of the data sheet for the 2716 A.C. Characteristics.

Symbol	Parameter	Limits			Unit	Test Condition
		Min	Typ.	Max		
t_{ACC1}	Addr. to Output Delay		250	450	ns	PD/PGM= $\overline{CS}$ =V_{IL}
t_{OH}	Addr. to Output Hold	0			ns	PD/PGM= $\overline{CS}$ =V_{IL}
t_{DF}	Chip Deselect to Output Float	0		100	ns	PD/PGM=V_{IL}
...	...	...	...	...	...	...

This EPROM requires a wait state for use with the 8086 (*460ns* constraint).

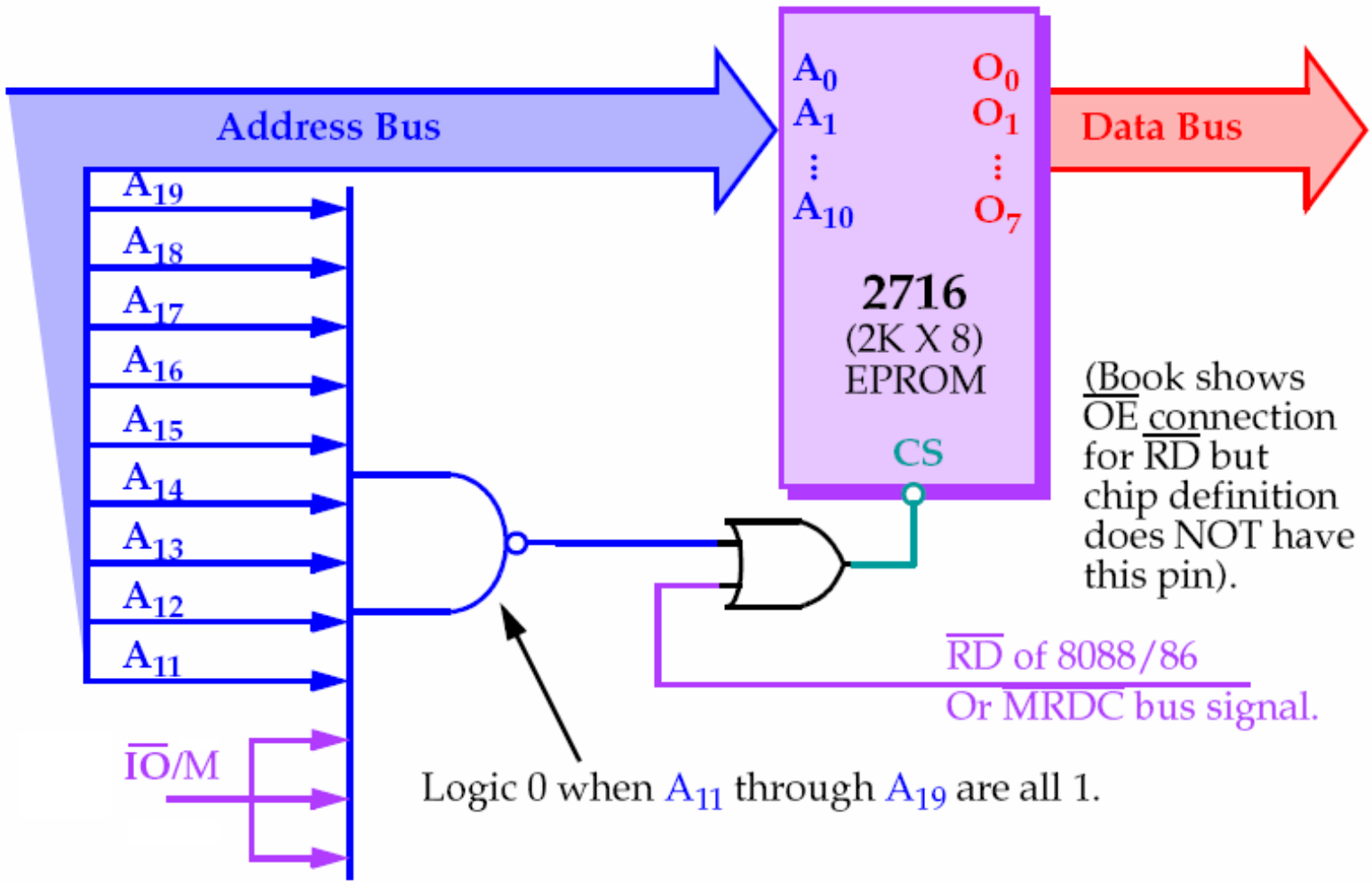
Bus Timing (8086) [3]



The maximum memory access time=3T-TCLAV-TDVCL which, for the 5MHz 8088, is 600-110-30=460ns. Actually the memory access time must be less than this since there will be propagation delays in going through buffers (about another 40ns).

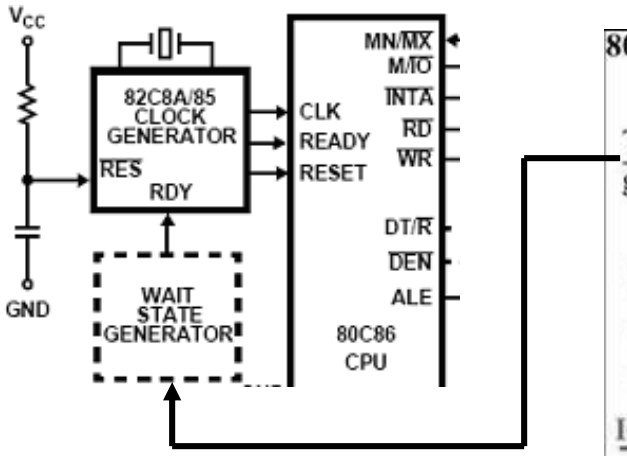
ROM/EPROM [2]

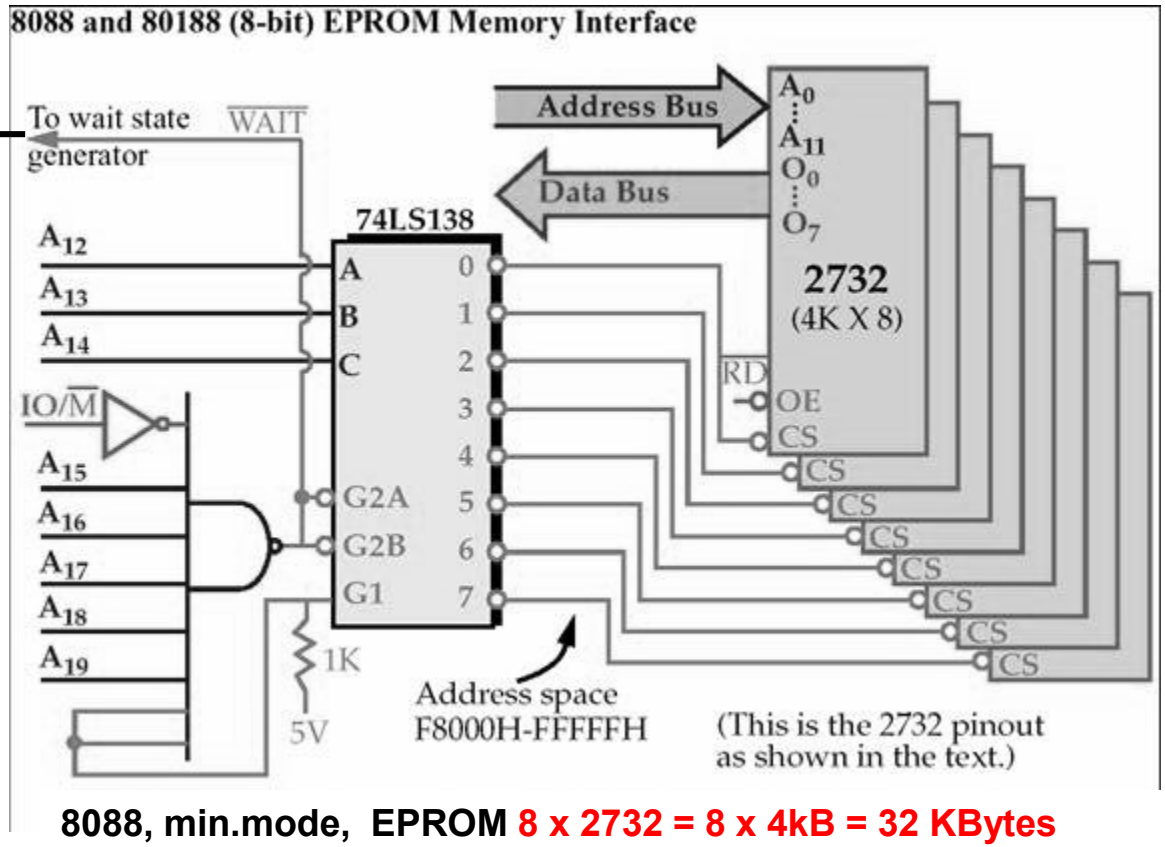
Memory Address Decoding



Address range: FF800h - FFFFFh

ROM/EPROM

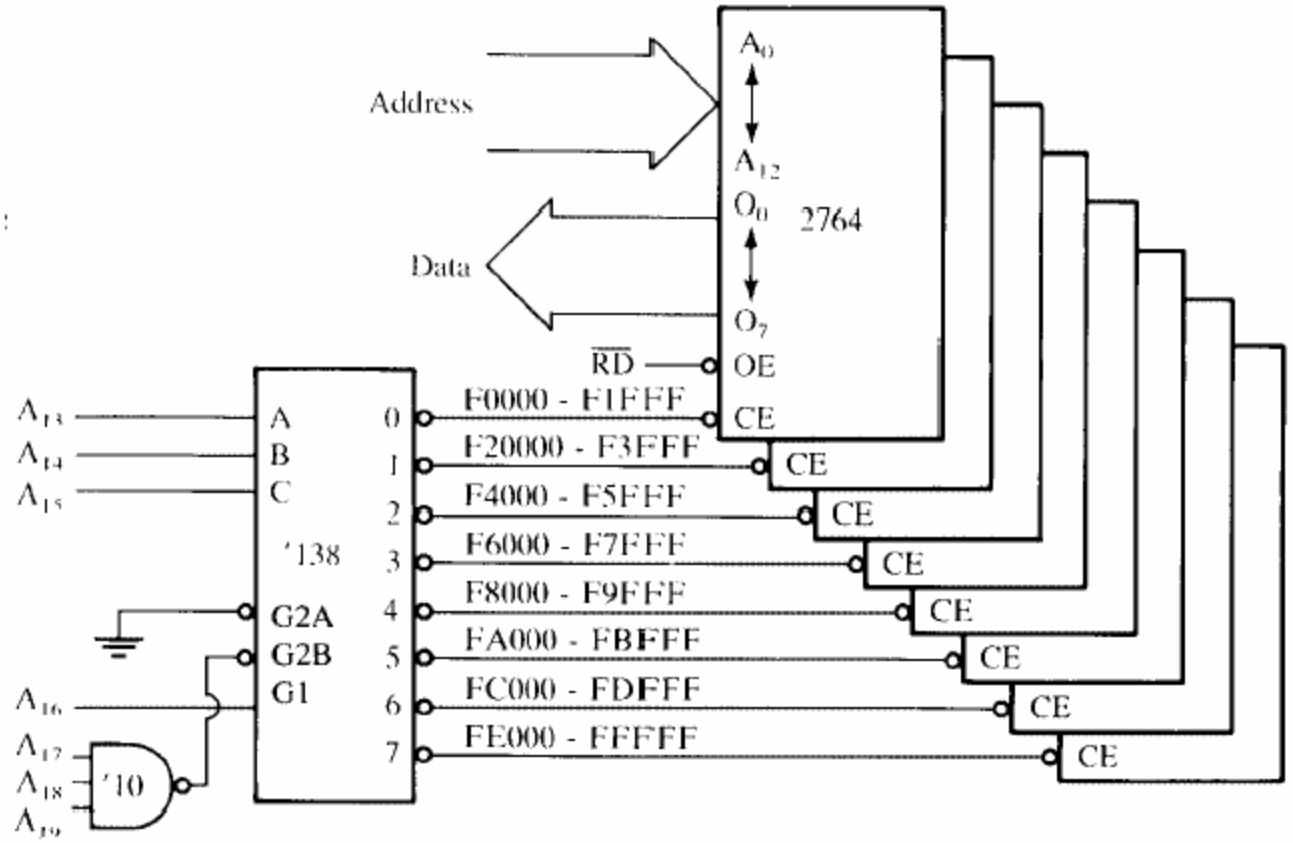




EPROM	Density (bits)	Capacity (bytes)
2716	16K	2K × 8
2732	32K	4K × 8
27C64	64K	8K × 8
27C128	128K	16K × 8
27C256	256K	32K × 8
27C512	512K	64K × 8
27C010	1M	128K × 8
27C020	2M	256K × 8
27C040	4M	512K × 8

8088, min.mode, EPROM 8 x 2732 = 8 x 4kB = 32 KBytes
Address mapping: 0: F8000- F8FFF; 1: F9000- F9FFF; ….. 7: FF00-FFFFF
•EPROM access (450ns) + decoder delay (12ns) > 8088 mem access (460ns)
•1 wait state (1 clock = 200ns): total access time 460+200=660 ns

ROM/EPROM



8088: 8 x 2764 = 8 x 8kB = 64 KB

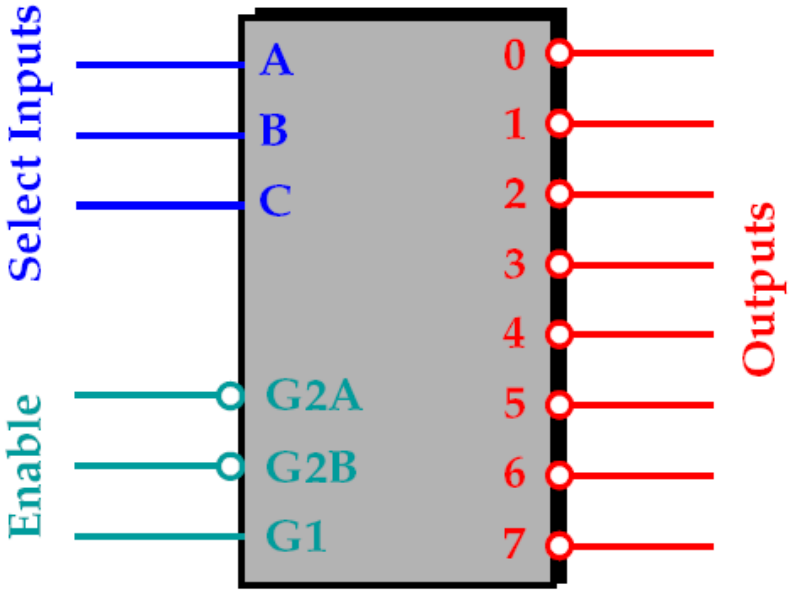
Memory address space: F0000 – FFFFF

Homework: re-design the adress decoder to map the memory blok in the lower part of the memory space !

Address decoders

Memory Address Decoding

The 3-to-8 Line Decoder (74LS138)



Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	**0**	1	1	1	1	1	1	1
0	0	1	0	0	1	1	**0**	1	1	1	1	1	1
0	0	1	0	1	0	1	1	**0**	1	1	1	1	1
0	0	1	0	1	1	1	1	1	**0**	1	1	1	1
0	0	1	1	0	0	1	1	1	1	**0**	1	1	1
0	0	1	1	0	1	1	1	1	1	1	**0**	1	1
0	0	1	1	1	0	1	1	1	1	1	1	**0**	1
0	0	1	1	1	1	1	1	1	1	1	1	1	**0**

Note that all *three* Enables (G2A, G2B, and G1) must be active, e.g. low, low and high, respectively.

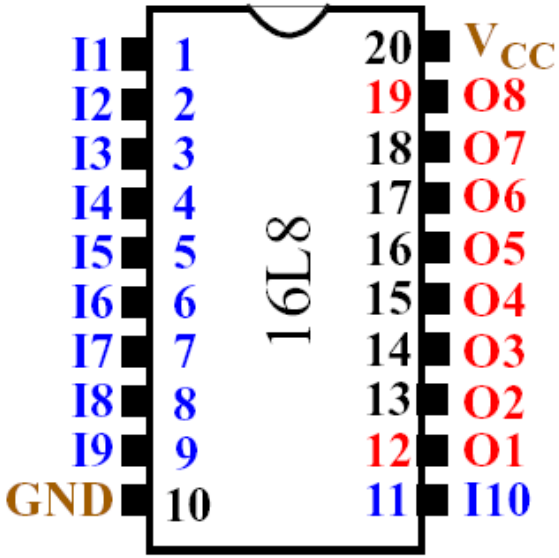
Address decoders

Memory Address Decoding

AMD 16L8 PAL decoder.

It has 10 fixed inputs (Pins 1-9, 11), two fixed outputs (Pins 12 and 19) and 6 pins that can be either (Pins 13-18).

Programmed to decode address lines A_{19} - A_{13} onto 8 outputs.

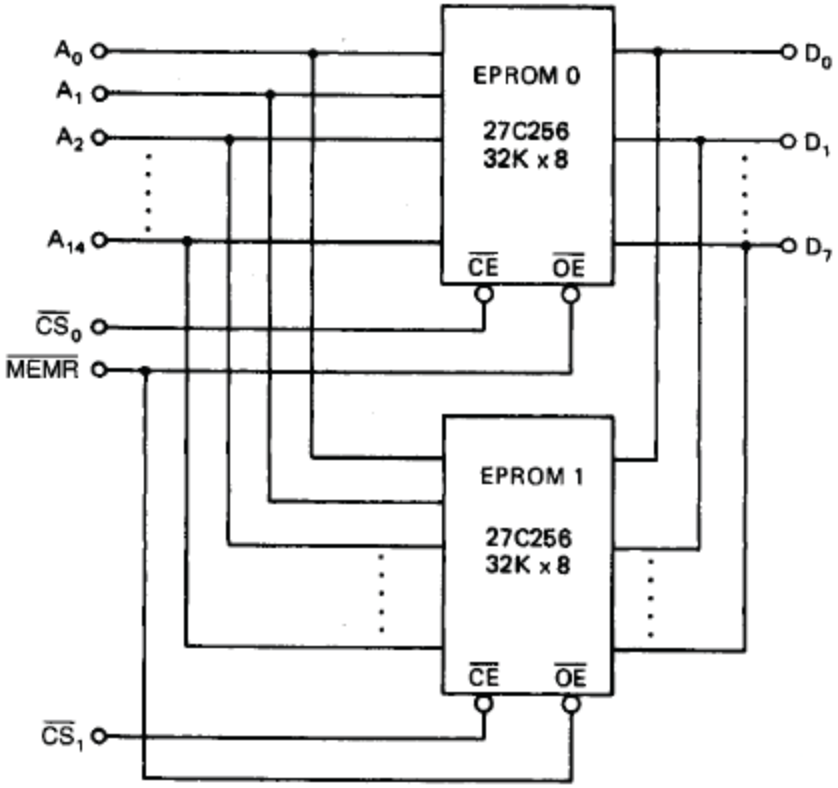



```
;pins 1    2    3    4    5    6    7    8    9    10
      A19  A18  A17  A16  A15  A14  A13  NC   NC   GND
;pins 11   12   13   14   15   16   17   18   19   20
      NC   O8   O7   O6   O5   O4   O3   O2   O1   VCC
```

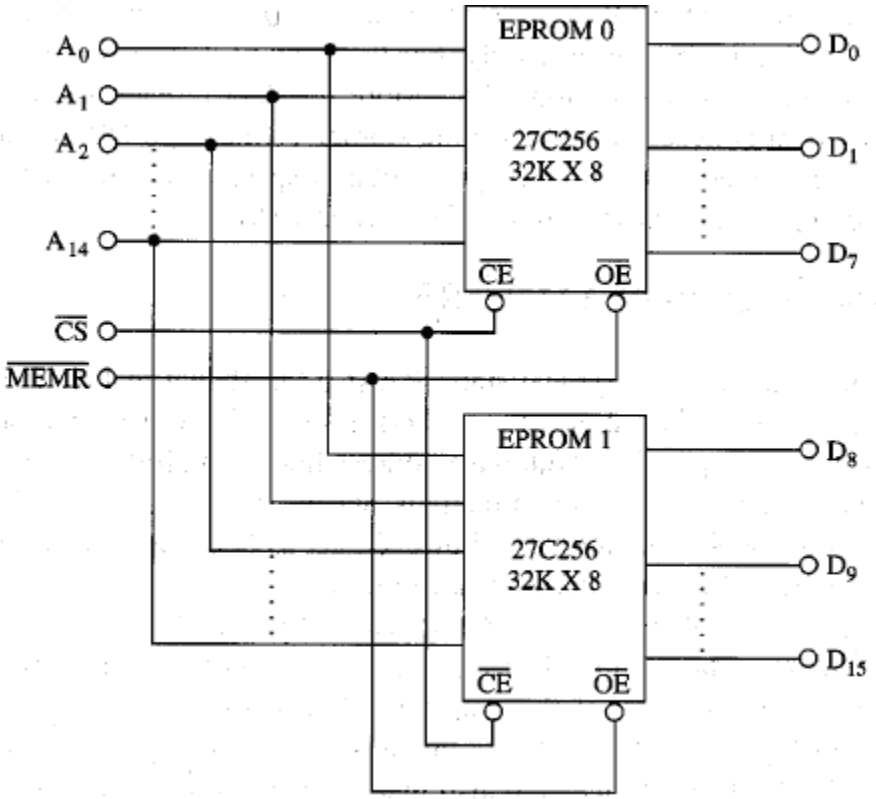
Equations:

```
/O1 = A19 * A18 * A17 * A16 * /A15 * /A14 * /A13
/O2 = A19 * A18 * A17 * A16 * /A15 * /A14 *  A13
/O3 = A19 * A18 * A17 * A16 * /A15 *  A14 * /A13
/O4 = A19 * A18 * A17 * A16 * /A15 *  A14 *  A13
/O5 = A19 * A18 * A17 * A16 *  A15 * /A14 * /A13
/O6 = A19 * A18 * A17 * A16 *  A15 * /A14 *  A13
/O7 = A19 * A18 * A17 * A16 *  A15 *  A14 * /A13
/O8 = A19 * A18 * A17 * A16 *  A15 *  A14 *  A13
```

Memory connections

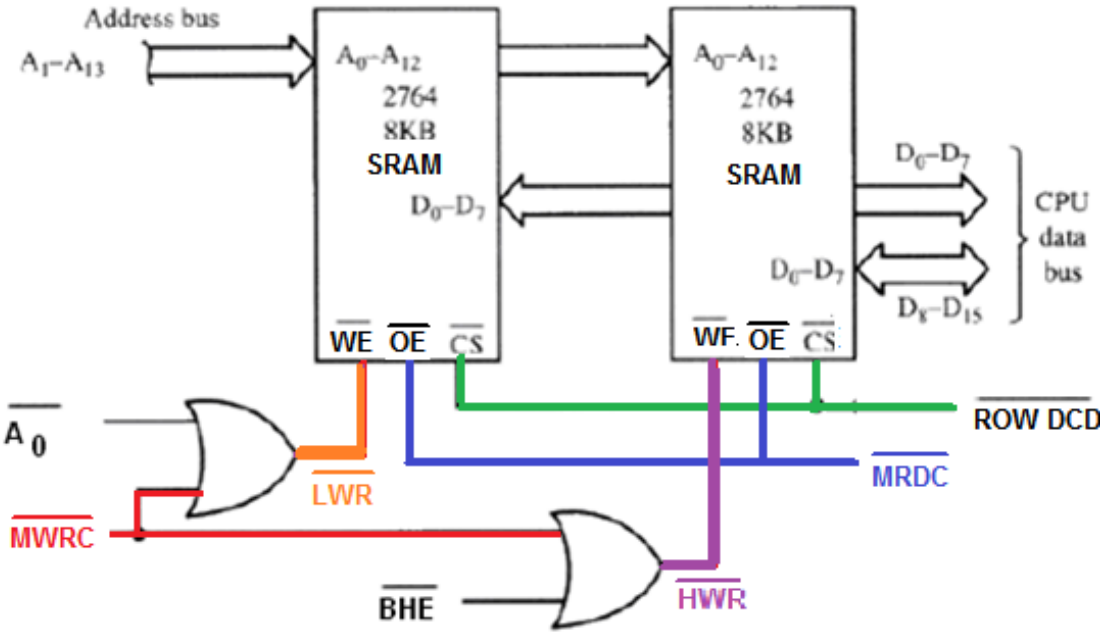


Parallel connection: extends the memory depth



Series connection: extends the memory (word) width

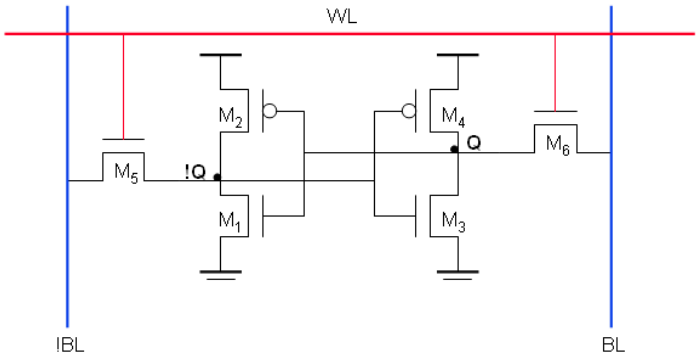
Connection to a 16 bit Data Bus (8086, 80286, 80186)

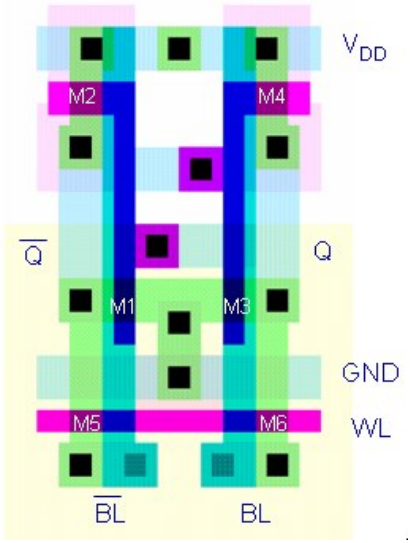


$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

RAM

Static (SRAM) [1]





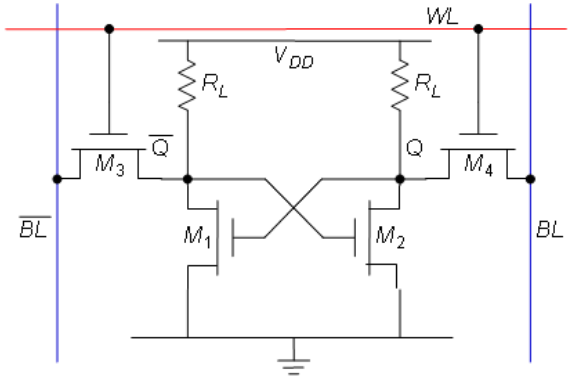


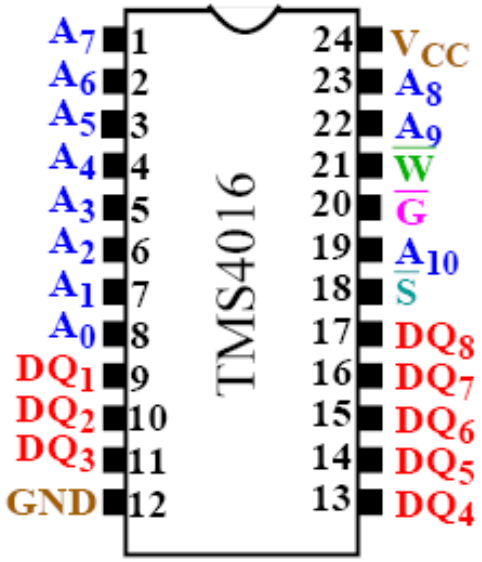
Table 12-2 Comparison of CMOS SRAM cells used in 1-Mbit memory (from [Takada91])

	Complementary CMOS	Resistive Load	TFT Cell
Number of transistors	6	4	4 (+2 TFT)
Cell size	58.2 μm^2 (0.7-μm rule)	40.8 μm^2 (0.7-μm rule)	41.1 μm^2 (0.8-μm rule)
Standby current (per cell)	10^{-15} A	10^{-12} A	10^{-13} A

SRAM [2]

SRAMs

TI TMS 4016 SRAM (2K X 8):



2K x 8 SRAM

Pin(s)	Function
A_0-A_{10}	Address
DQ_0-DQ_7	Data In/Data Out
$\overline{S}$ ($\overline{CS}$)	Chip Select
$\overline{G}$ ($\overline{OE}$)	Read Enable
$\overline{W}$ ($\overline{WE}$)	Write Enable

Virtually identical to the EPROM with respect to the pinout.

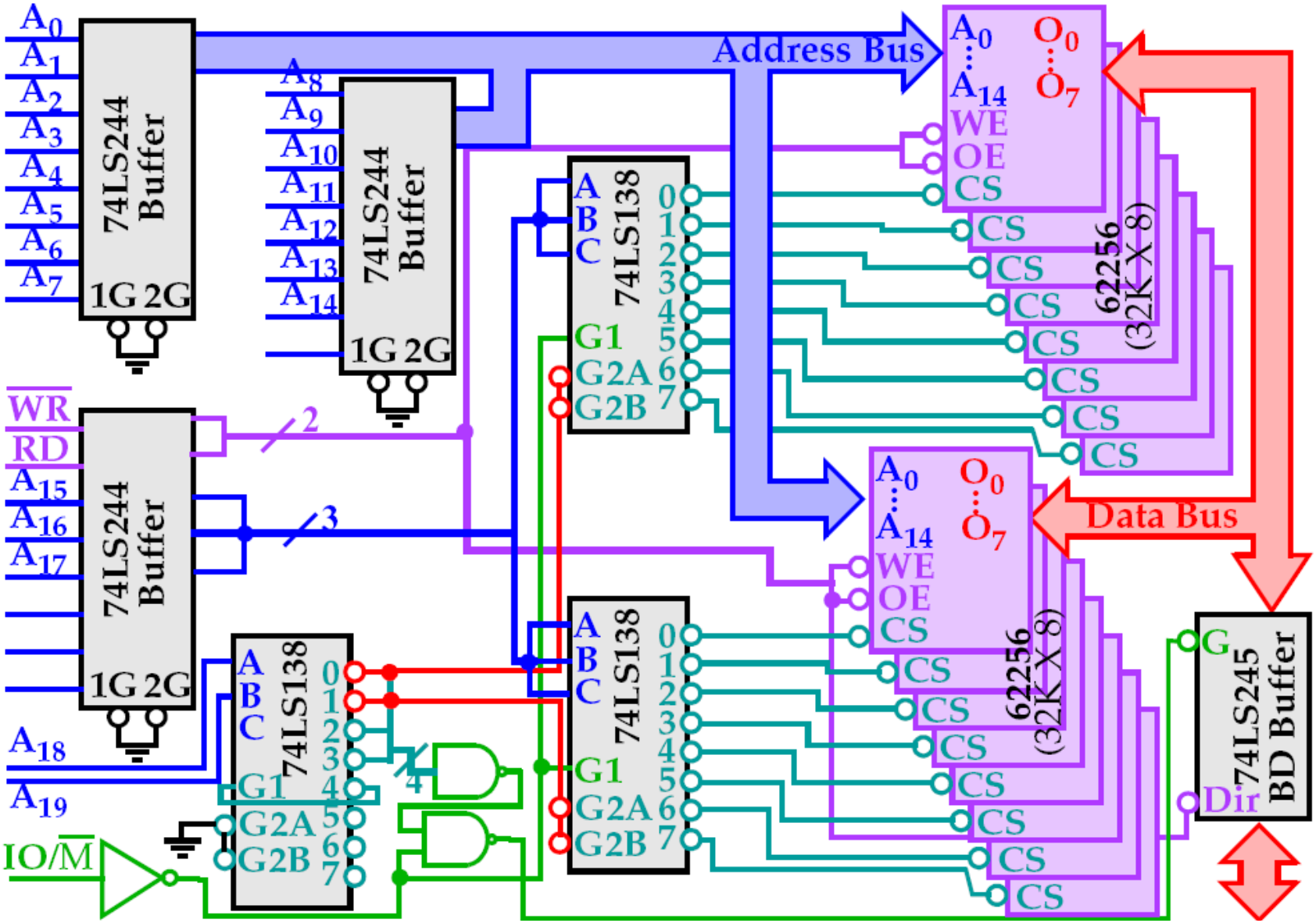
However, access time is faster (250ns).

See the timing diagrams and data sheets in text.

SRAMs used for *caches* have access times as low as *10ns*.

SRAM

8088 and 80188 (8-bit) RAM Memory Interface



16, 32 …. bit data bus interface

No concern on read operation:

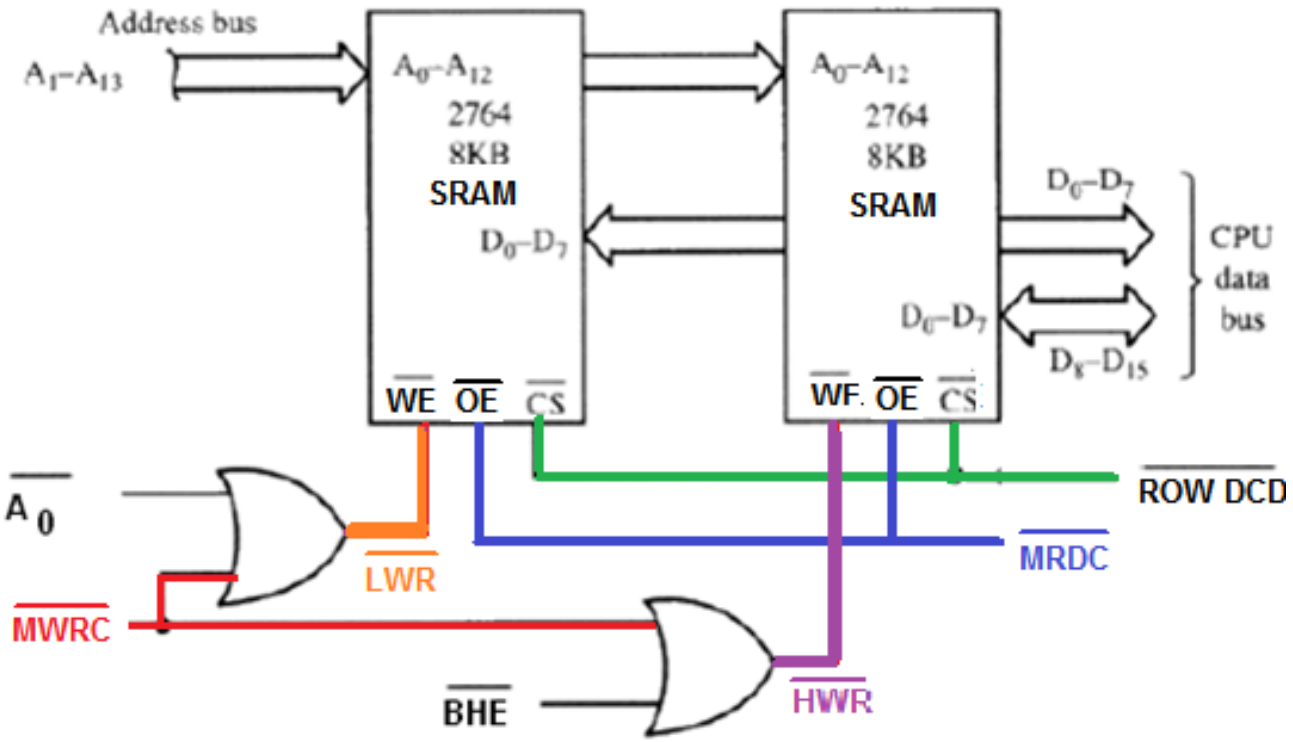
- memory outputs a whole word (16 bit, 32 bit …)
- CPU reads the desired byte (BL, BH) or word

(Ex: MOV AL, mem ; MOV AH, mem; MOV AX, mem)

Only writing must be handled:

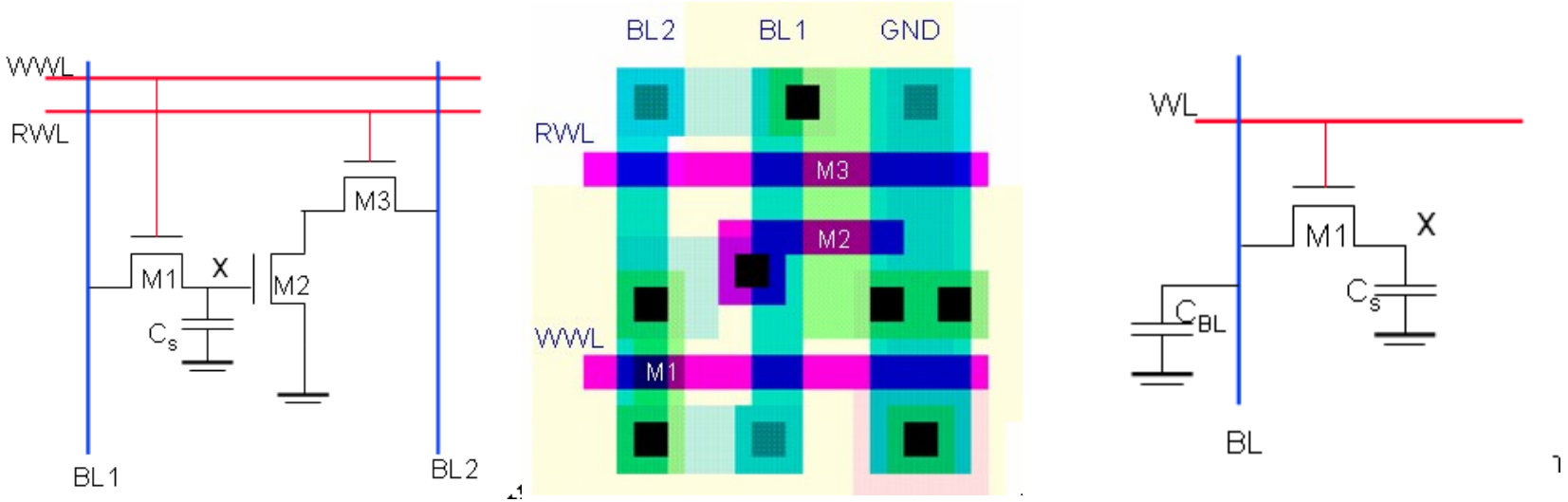
- LWR, HWR

Example:



DRAM

Dynamic (DRAM) [1]



Size: ½ SRAM cell ⇒ higher capacity ⇒ address pins are multiplexed

Refresh: 1 .. 4 msec ⇒ special circuit inside DRAM

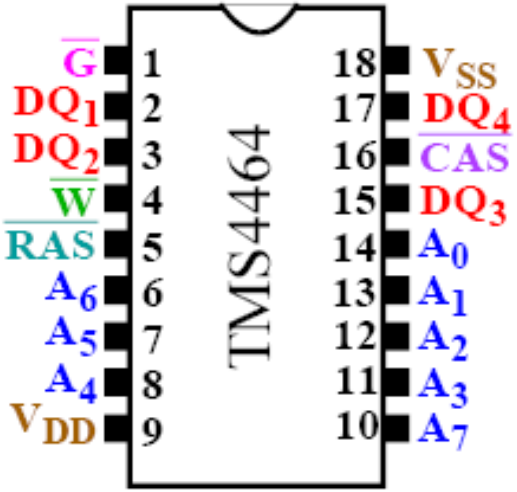
read, write, refresh cycles

SDR, DDR, Rambus

DRAM [2]

DRAMs

TI TMS4464 DRAM (64K X 4):



64K x 4 DRAM

Pin(s)	Function
A_0-A_7	Address
DQ_0-DQ_4	Data In/Data Out
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
$\overline{G}$	Output Enable
$\overline{W}$	Write Enable

The TMS4464 can store a total of 256K bits of data.

It has **64K** addressable locations which means it needs **16** address inputs, but it has only 8.

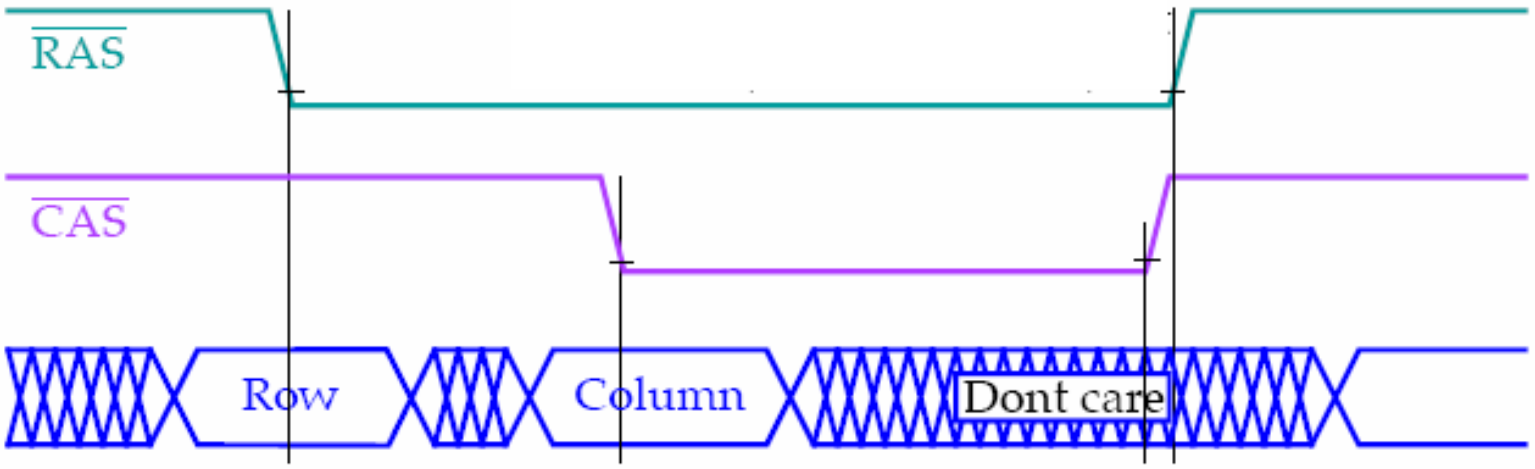
The row address (A_0 through A_7) are placed on the address pins and strobed into a set of internal latches.

The column addres (A_8 through A_{15}) is then strobed in using CAS.

DRAM [2]

DRAMs

TI TMS4464 DRAM (64K X 4) Timing Diagram:



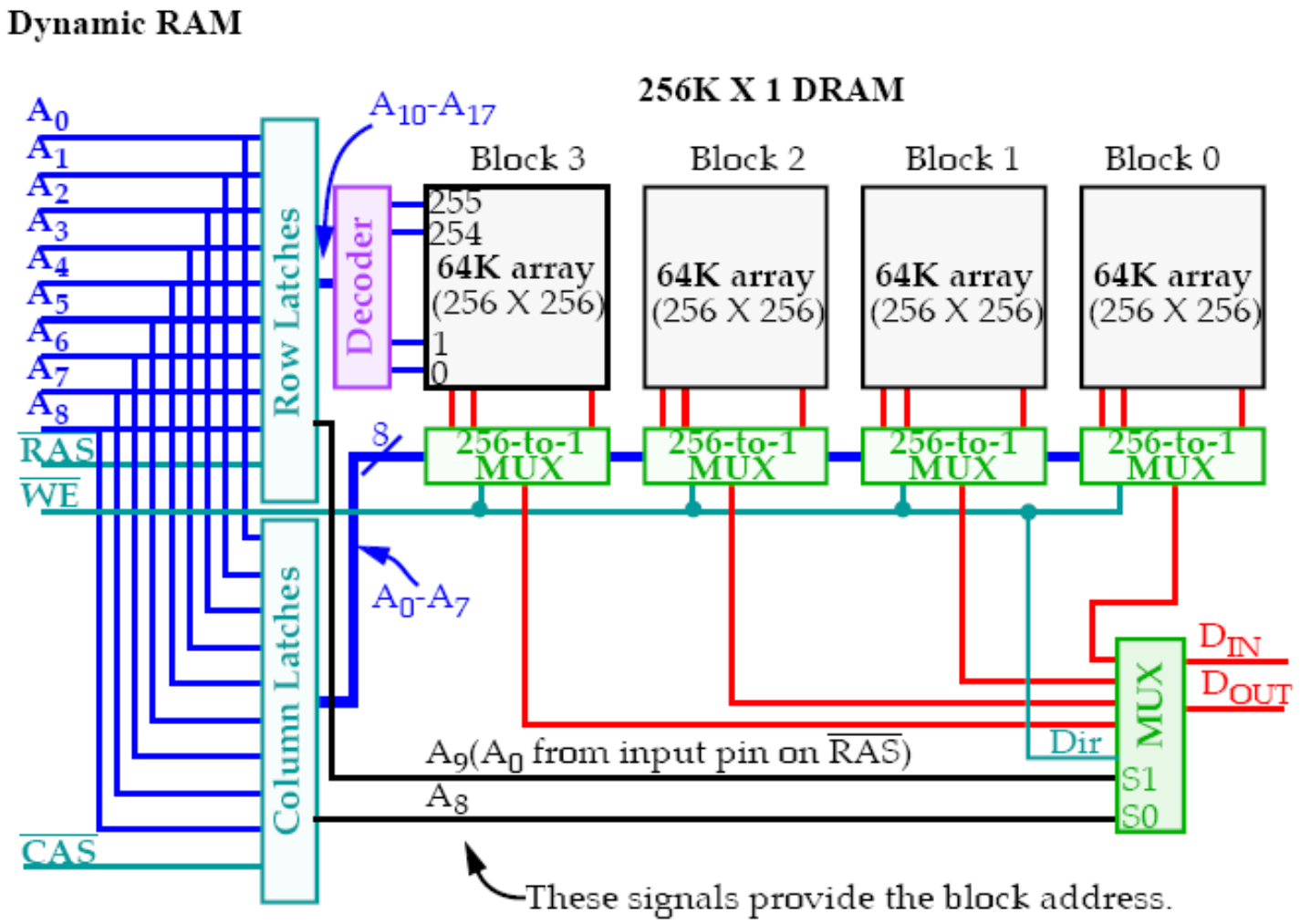
$\overline{\text{CAS}}$ also performs the function of the chip select input.

#RAS & #CAS should be provided by a DRAM controller

DRAM controller should multiplex (in time) the address lines (ex A0-15) to:

- row address (ex. A8-15)
- column address (ex. A0-7)

256K X 1 DRAM – internal structure [2]



DRAM refresh

Special refresh cycle

- Occurs **transparently** while other memory components operate and is called *transparent refresh* or *cycle stealing*.
- A RAS-only cycle strobes a row address into the DRAM,
- The capacitors are recharged for the selected row by reading the bits out internally and then writing them back.

Example:

256K X 1 DRAM (256 rows x 256 columns x 4 blocks)

$\Rightarrow$ refresh must occur every *15.6μs* (4ms/256).

For the 8086, a read or write occurs every *800ns (4x200=4xTclk)*.

$\Rightarrow$ **19** memory reads/writes per refresh (*15.6μs/0.8μs = 19.5)*

$\Rightarrow$ read/write cycle takes **5%** of the refresh time

DRAM Controllers [2]

⇒ address multiplexing and generation of the DRAM control signals.

These devices tend to get very complex.

We will focus on a simpler device:

Intel 82C08, which can control **two** banks of *256K X 16* DRAM memories for a total of 1 MB.

Microprocessor adress A1 through A18 (18 bits) drive the 9 **Column Address** inputs (AL) and 9 **RowAddress** inputs (AH) of the *82C08*.

Either RAS0/CAS0 or RAS1/CAS1 are strobed depending on the address.

WE (from the 82C08), BHE and A0 are used to determine if a write is to be performed and which byte(s) (low or high or both) is to be written.

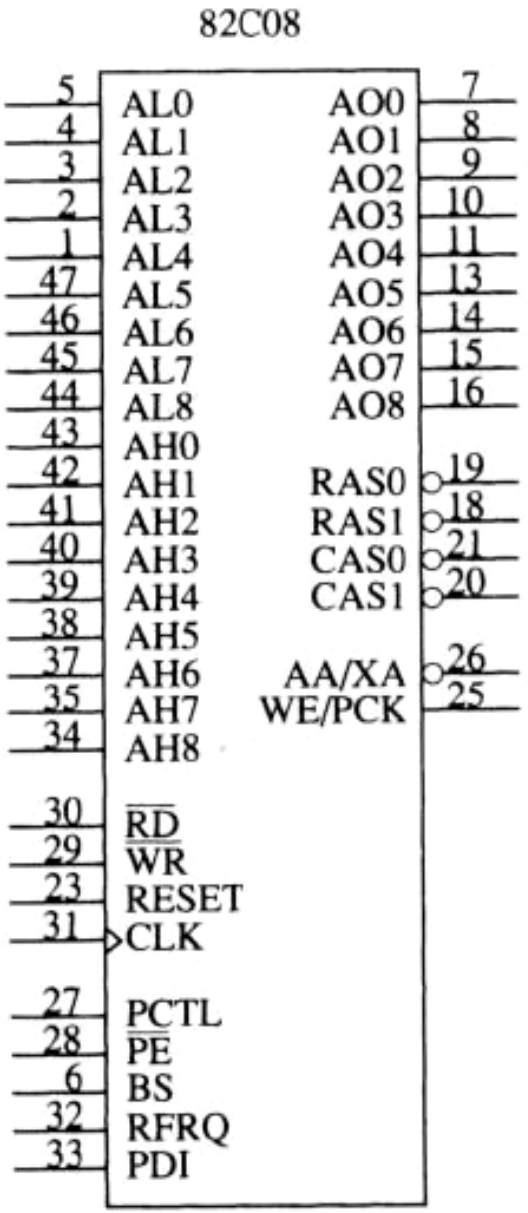
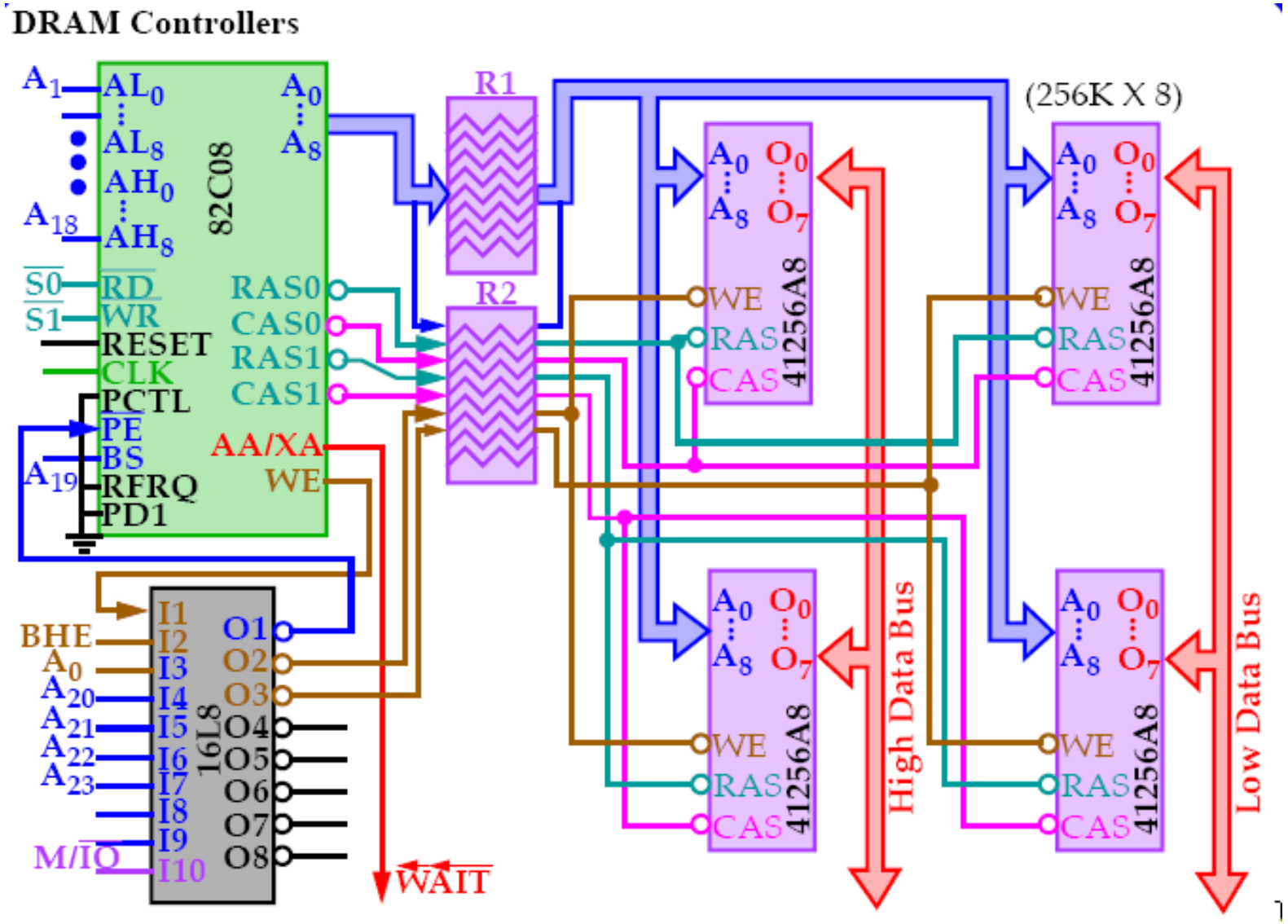
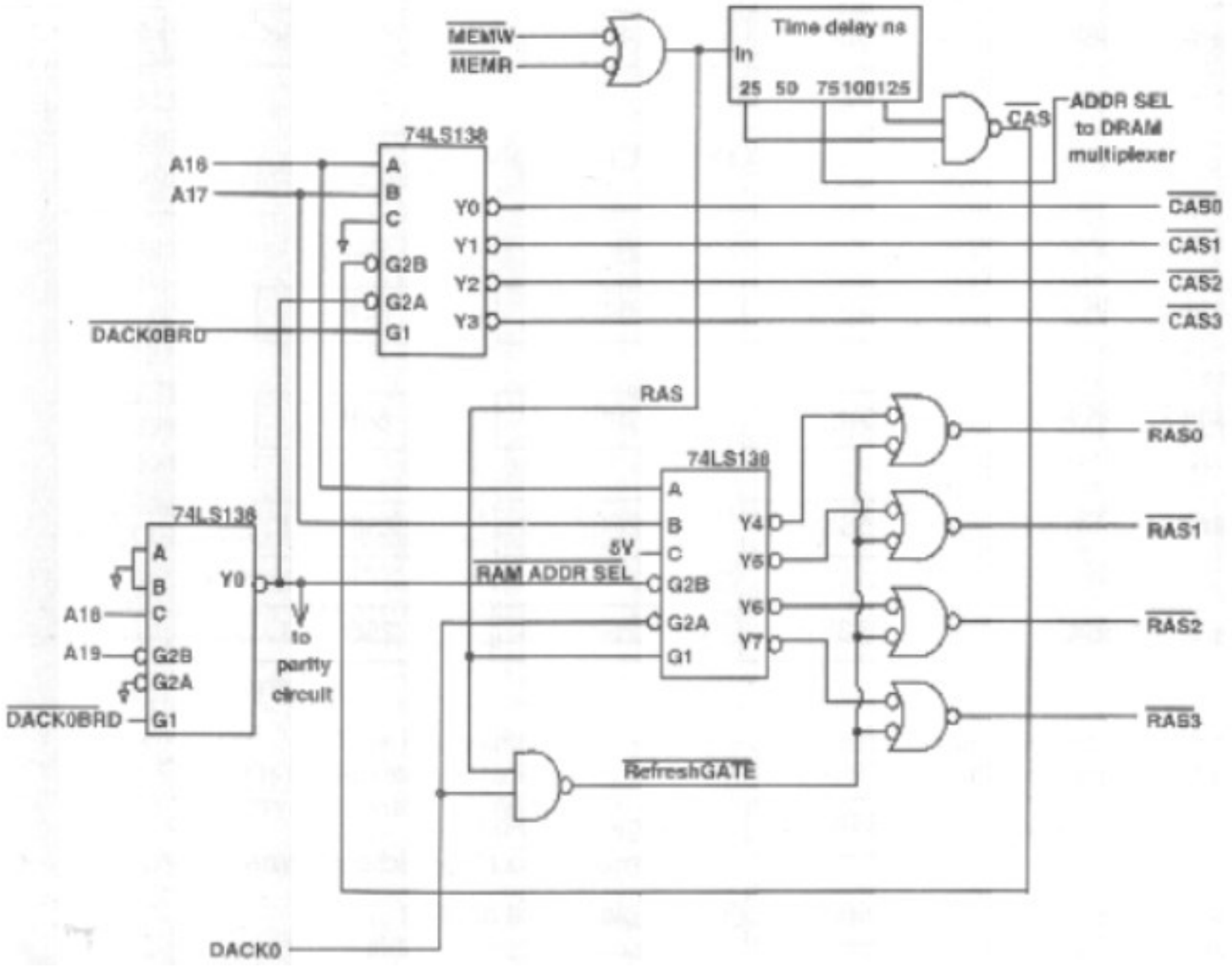


FIGURE 10–41 The 82C08 DRAM controller that controls two banks of memory.

DRAM Controllers [2]



DRAM Controllers



References

[1] CSE477, VLSI Digital Circuits, Fall 2005, Lecture 23: Memory Design ROM, SRAM, DRAM, CAM, http://www.cse.psu.edu/~cg477/

[2] UMBC, System Design and Programming, Fall 2002, Lecture: Memory 1-3.

[3] ELE 3230 - Microprocessors and Computer Systems, Chapter 8

Memory Address Decoding, http://course.ie.cuhk.edu.hk/~ele3230b/lecture.htm

[4] Texas Instruments, TMS320C6000 EMIF: Overview of Support of High Performance Memory Technology, Application Report SPRA631 - April 2000

[5] Barry B. Brey, The Intel Microprocessors: 8086/8088, 80186,80286, 80386 and 80486. Architecture, Programming, and Interfacing, 4-th edition, Prentice Hall, 1996, pp. 312-361.