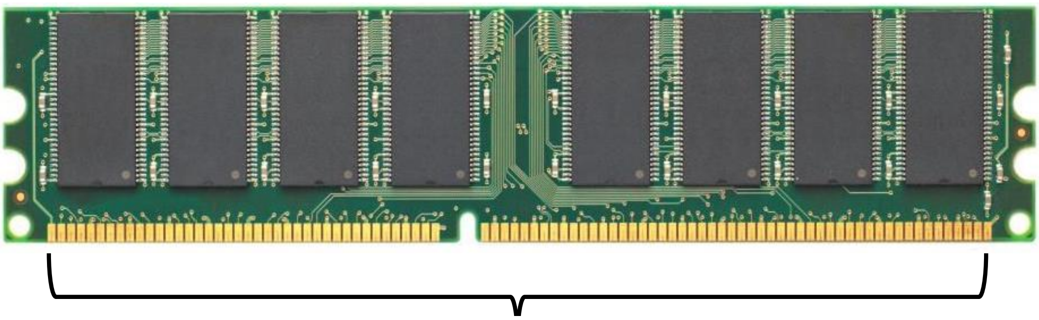
# Lecture 2

## Memory Card with 8 Chips

- Each chip is 16 M x 1 b (16 mega bits)
- Card total is 16 M x 8 b (16 mega byte)
- We can install several memory card modules on the main board of computer.

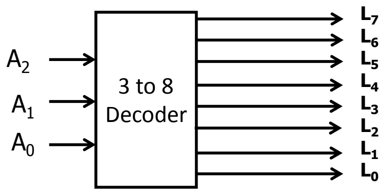


DATA BUS / ADDRESS BUS / CONTROL BUS

# Example : 3-to-8 Decoder

- There are 3 inputs, and $2^3 = 8$ output lines.
- Depending on the inputs, only one of the output lines will be 1, all other output lines will be 0.

**INPUTS**



**OUTPUTS**

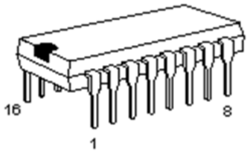
# Truth Table for 3-to-8 Decoder

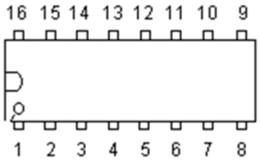
- Only one output line is selected at a time.

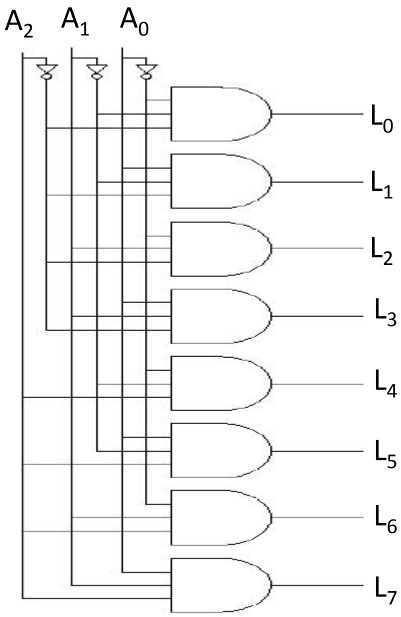
| INPUTS | | | OUTPUTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $A_2$ | $A_1$ | $A_0$ | $L_7$ | $L_6$ | $L_5$ | $L_4$ | $L_3$ | $L_2$ | $L_1$ | $L_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# Inside of 3-to-8 Decoder

- Decoders are manufactured as standard Integrated Circuits.

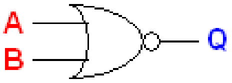




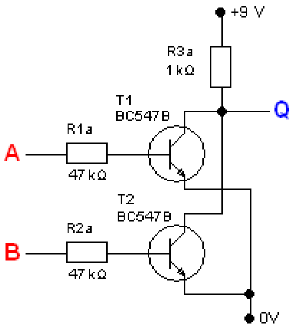


# Logic Gates and their Electronic Circuit Implementation

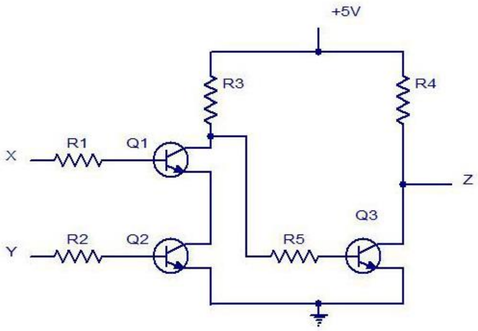
**NOR Gate**



**NOR Gate Circuit**
**RTL (Resistor Transistor Logic)**



**AND Gate Circuit**



# Multiplexer, Demultiplexer, Decoder

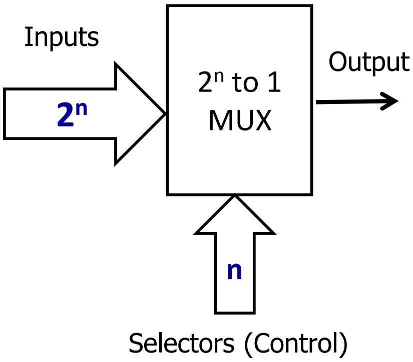
MULTIPLEXER

Only one of the Inputs is selected, and it is forwarded to the output.

DEMULTIPLEXER

Only one of the Outputs is selected, and the input is forwarded to it.

If Input is constantly "1", this is a DECODER.



Inputs

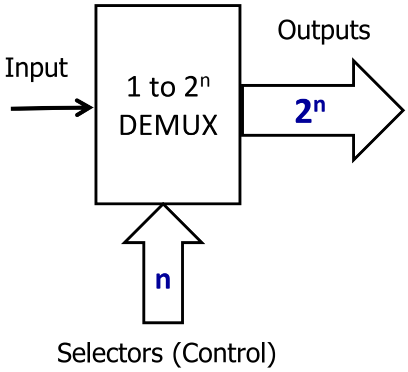
$2^n$

$2^n$ to 1 MUX

Output

n

Selectors (Control)



Input

1 to $2^n$ DEMUX

Outputs

$2^n$

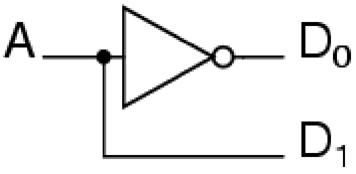
n

Selectors (Control)

# Simplest Example: 1-to-2 line decoder

A is the address and D is the selected lines.
$D_0$ is NOT A. ($D_0$ = A ‘ )
$D_1$ is A ($D_1$ = A )

Circuit (Inverter)

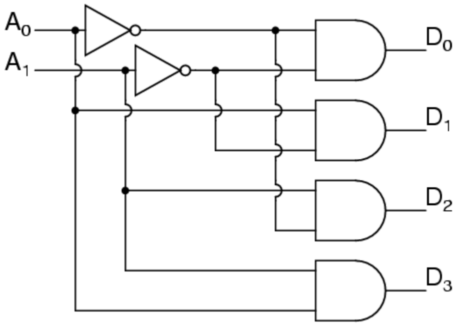


Truth Table

| A | $D_1$ | $D_0$ |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

# Example: 2-to-4 line decoder

Circuit (Inverters + And gates)

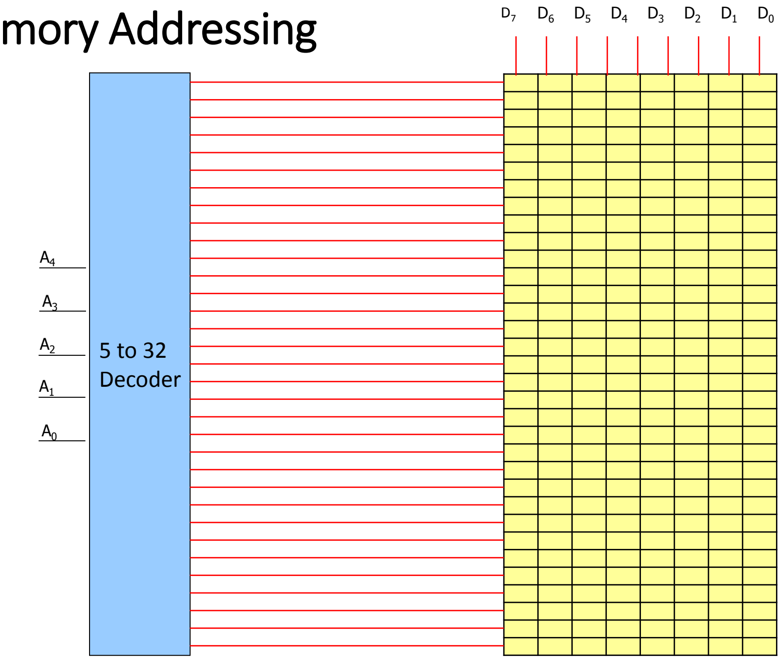


Truth Table

| $A_1$ | $A_0$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

# Example : One Dimensional Memory Addressing

Data Bus (8 bits)



Address Bus
(5 bits)

32 x 8 bit
Memory

# Some Terminology

- **COMBINATIONAL CIRCUIT**
  - Outputs depend only on the current inputs.

- **SEQUENTIAL CIRCUIT**
  - Outputs depend both on the inputs and the state of circuit (previous outputs).
  - Memory units must be Sequential circuits.

- **FLIP-FLOP** is a memory unit that is triggered by a clock signal.

- **LATCH** is a memory unit that is not triggered by a clock signal.
  - The value of latch can be changed whenever the latch is enabled.

- **CACHE** is a memory faster than RAM.
  - Stores frequently used instructions and data.
  - Primary cache (Level1): Located within the CPU.
  - Secondary cache (Level2): Located near the CPU.

# Example Metrics of Memory

| Memory Type | Capacity | Access Time |
|---|---|---|
| Registers in CPU | 1 KB | 1 ns |
| Caches in CPU | 1 MB | 10 ns |
| Main Memory (RAM) | 10 GB | 100 ns |
| Hard Disk Drive | 1 TB | 10 ms |

# Metrics used for Memory

1 Byte = 8 bits (Binary DigiT)

| Unit | Name | Synonym | Real Value (base 2) | Approximate Value (base 10) |
|---|---|---|---|---|
| 1 KB | Kilo | Thousand | $2^{10}$ = 1024 | $10^{3}$ = 1000 |
| 1 MB | Mega | Million | $2^{20}$ | $10^{6}$ |
| 1 GB | Giga | Billion | $2^{30}$ | $10^{9}$ |
| 1 TB | Tera | Trillion | $2^{40}$ | $10^{12}$ |

# Metrics used for CPU speed

**Hertz** (clock rate) is the unit of frequency measurement.
It represents number of clock cycles (of periodic pulse wave) **per second**.

| Frequncy Unit | Unit name | Synonym | Hertz Value |
|---|---|---|---|
| 1 KHz | Kilo | Thousand | $10^{3}$ |
| 1 MHz | Mega | Million | $10^{6}$ |
| **1 GHz** | **Giga** | **Billion** | $\mathbf{10^{9}}$ |
| 1 THz | Tera | Trillion | $10^{12}$ |

| Time Unit | Unit name | Synonym | Seconds Value |
|---|---|---|---|
| 1 ms | Mili | One thousandth of a second. | $10^{-3}$ |
| 1 μs | Micro | One millionth | $10^{-6}$ |
| **1 ns** | **Nano** | **One billionth** | $\mathbf{10^{-9}}$ |
| 1 ps | Pico | One trillionth | $10^{-12}$ |

# Example : CPU Clock Period

Assume the CPU clock frequency is 100 MHz.

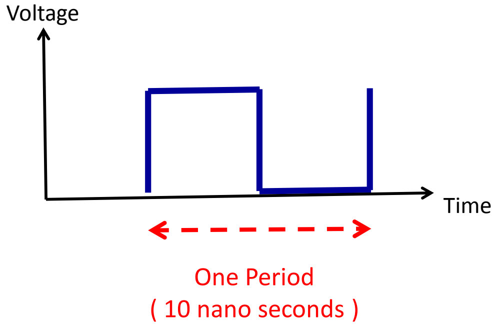
**QUESTION:** Find the pulse wave period in terms of nano seconds.

$$Period\ (seconds) = \frac{1}{Frequency}$$

$$= \frac{1}{100*10^6} = 10^{-8}\ seconds$$

$$= 10^{-8}\ seconds\ * 10^9\ nanoseconds/second$$

$$= 10\ nanoseconds$$



# SRAM versus DRAM

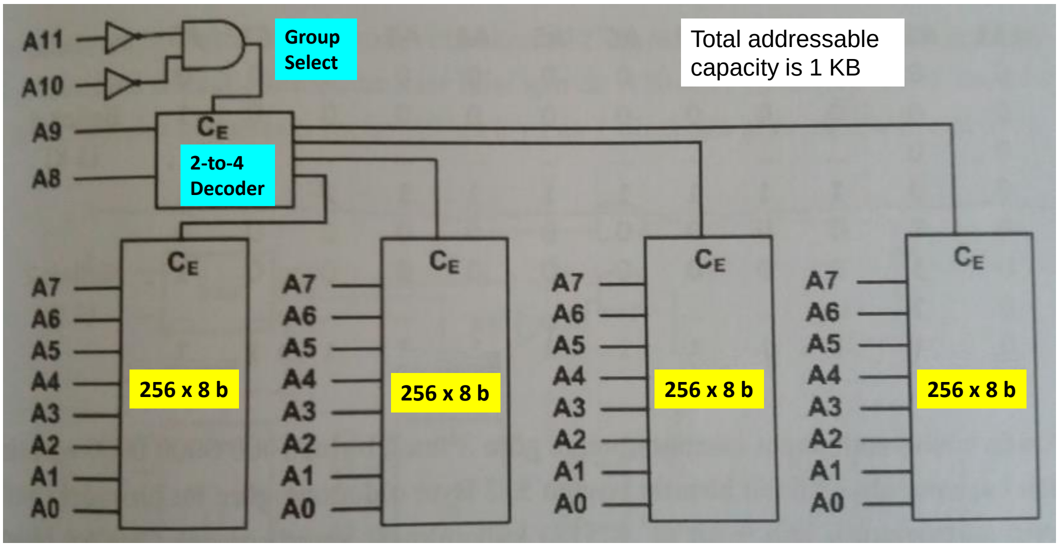
| Memory Type | Number of bits in a chip | Speed | Where used |
|---|---|---|---|
| SRAM  (Static) | 8 | Fast | Cache |
| DRAM  (Dynamic) | 1 | Slow | Main memory |

# Lecture 3

## Example 1 : Memory Design

- Data Bus = 8 bits,  Address Bus = 12 bits
- Each memory chip is 256 x 8 bits.
- $A_0$ through $A_7$ used for location selection inside a chip ($2^8$ = 256 rows)
- Decoder used for chip selections (**$C_E$** : Chip enable)
- $A_{10}$ and $A_{11}$ used to select the decoder itself.

For simplicity, Data Bus and R/W ' is ignored.



Total addressable capacity is 1 KB

# Memory Address Map

| MEMORY CHIP | DECODER SELECT | | CHIP SELECT | | LOCATION SELECTION WITHIN A CHIP | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
| MEMORY 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 0 | ... | ... | ... | ... | ... | ... | ... | ... |
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MEMORY 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 1 | ... | ... | ... | ... | ... | ... | ... | ... |
| | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MEMORY 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 1 | 0 | ... | ... | ... | ... | ... | ... | ... | ... |
| | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MEMORY 4 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 1 | 1 | ... | ... | ... | ... | ... | ... | ... | ... |
| | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

# Memory Address Map (Binary and Hexadecimal)

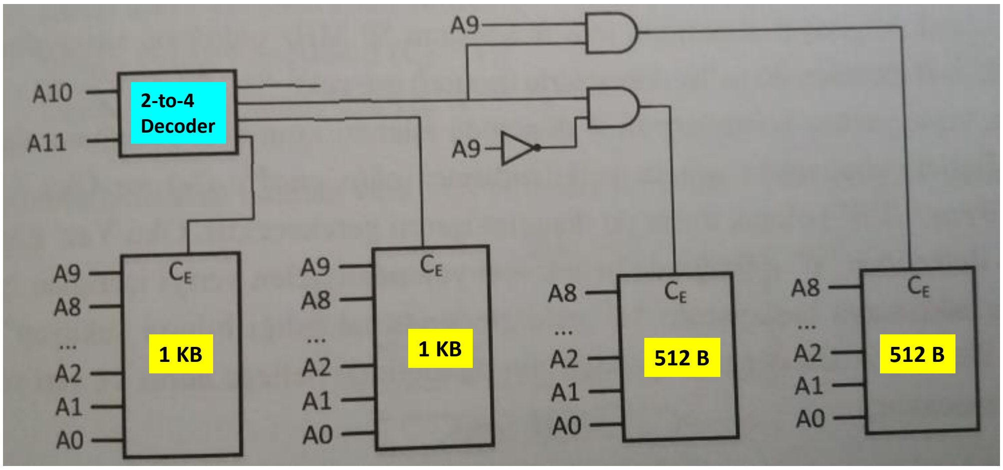
| CHIP | SMALLEST ADDRESS | | | | | | | | | | | | BIGGEST ADDRESS | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
| M1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M4 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| GAP | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

- Total possible capacity is $2^{12}$ = 4 KB
- Addressable capacity is 1 KB.
- The last 3 KB is unused gap space.
- Addresses 400 thru FFF are undefined (unaccessable) !

| CHIP | SMALLEST ADDRESS ($A_{11} – A_0$) | BIGGEST ADDRESS ($A_{11} – A_0$) | CAPACITY |
|---|---|---|---|
| M1 | 000 | 0FF | 256 B |
| M2 | 100 | 1FF | 256 B |
| M3 | 200 | 2FF | 256 B |
| M4 | 300 | 3FF | 256 B |
| GAP | 400 | FFF | 3 KB |

# Example 2 : Memory Design with different chips

- Total addressable capacity is = (1 KB * 2) + (512 B * 2) = 3 KB



## Memory Address Map (Binary and Hexadecimal)

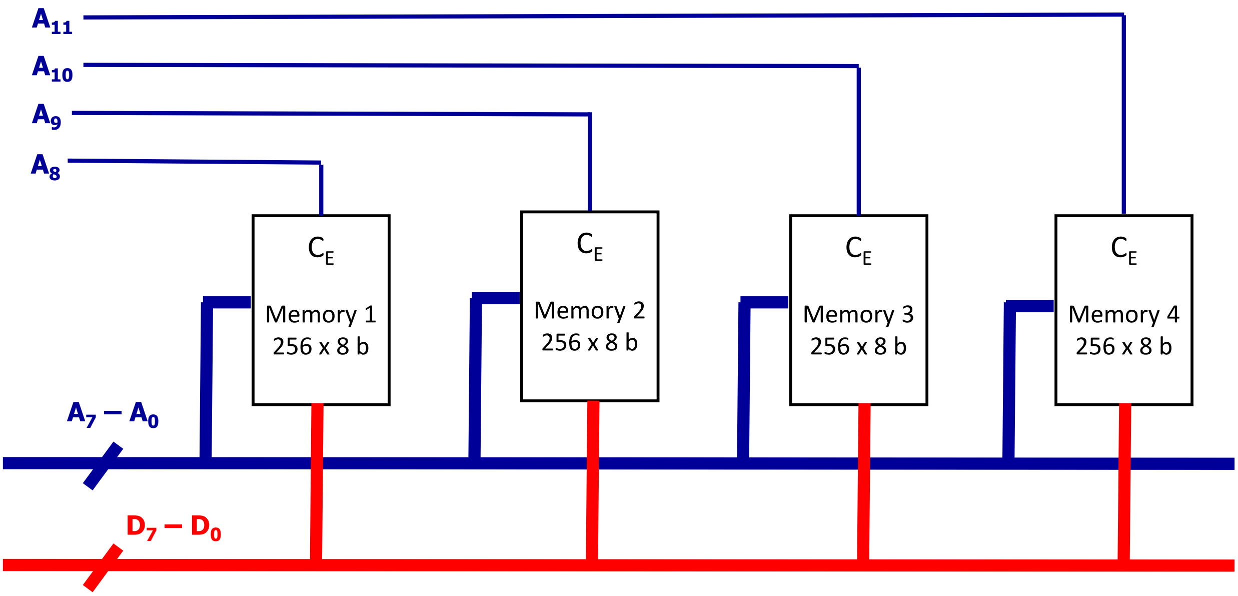
| CHIP | SMALLEST ADDRESS | | | | | | | | | | | | BIGGEST ADDRESS | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
| M1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M4 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| EMPTY | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

- Total possible capacity is $2^{12}$ = 4 KB
- Addressable capacity is 3 KB.
- The last 1 KB is unused gap space.

| CHIP | SMALLEST ADDRESS ($A_{11} – A_0$) | BIGGEST ADDRESS ($A_{11} – A_0$) | CAPACITY |
|---|---|---|---|
| M1 | 0 0 0 | 3 F F | 1 KB |
| M2 | 4 0 0 | 7 F F | 1 KB |
| M3 | 8 0 0 | 9 F F | 512 B |
| M4 | A 0 0 | B F F | 512 B |
| GAP | C 0 0 | F F F | 1 KB |

# Example 3 : Memory Design without Address Decoder

- Total addressable capacity is $= 2^8 \times 4 = 256$ B x 4 = 1 KB



# Memory Address Map (Binary and Hexadecimal)

| CHIP | SMALLEST ADDRESS | | | | | | | | | | | | BIGGEST ADDRESS | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
| M1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| EMPTY | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

- Total possible capacity is $2^{12} = 4$ KB

- Addressable capacity is 1 KB.

- **Total unused gap space = 12 x 256 B = 3 KB (IRREGULARLY SPACED !)**

| CHIP | SMALLEST ADDRESS ($A_{11} - A_0$) | BIGGEST ADDRESS ($A_{11} - A_0$) | CAPACITY |
|---|---|---|---|
| GAP | 0 0 0 | 0 F F | 1 x 256 B |
| M1 | 1 0 0 | 1 F F | 256 B |
| M2 | 2 0 0 | 2 F F | 256 B |
| GAP | 3 0 0 | 3 F F | 1 x 256 B |
| M3 | 4 0 0 | 4 F F | 256 B |
| GAP | 5 0 0 | 7 F F | 3 x 256 B |
| M4 | 8 0 0 | 8 F F | 256 B |
| GAP | 9 0 0 | F F F | 7 x 256 B |

# Calculation of unused (gaps) address bytes

| Smallest Address | Biggest Address | Count of Bytes |
|---|---|---|
| 0 0 0 | 0 F F | 256 |

| Smallest Address | Biggest Address | Count of Bytes |
|---|---|---|
| 3 0 0 | 3 F F | 256 |

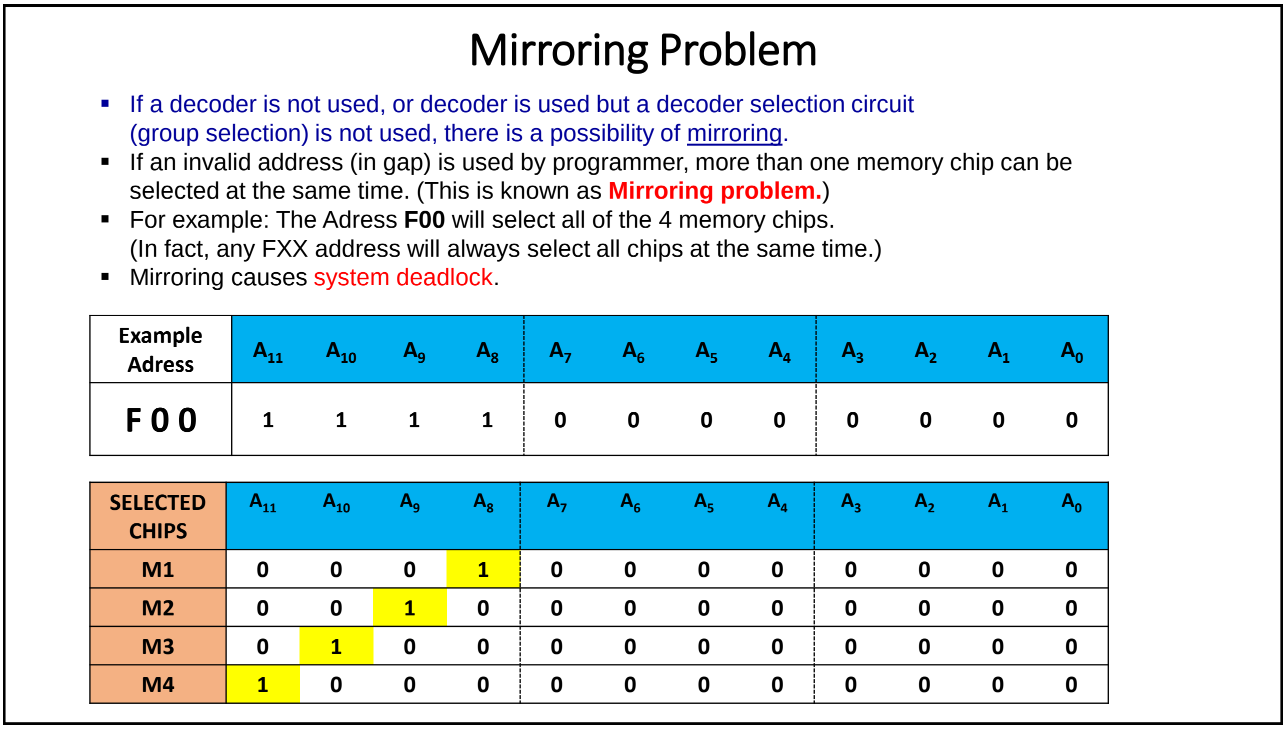
| Smallest Address | Biggest Address | Count of Bytes |
|---|---|---|
| 5 0 0 | 5 F F | 256 |
| 6 0 0 | 6 F F | 256 |
| 7 0 0 | 7 F F | 256 |

| Smallest Address | Biggest Address | Count of Bytes |
|---|---|---|
| 9 0 0 | 9 F F | 256 |
| A 0 0 | A F F | 256 |
| B 0 0 | B F F | 256 |
| C 0 0 | C F F | 256 |
| D 0 0 | D F F | 256 |
| E 0 0 | E F F | 256 |
| F 0 0 | F F F | 256 |

Total unused gap space
12 x 256 B = 3 KB

# Disadvantage of not using a decoder

- If an address decoder is not used, utilization of address bus will be ineffective.
- The possible addresses of memory with a 12-bit Address Bus is $2^{12}$ = 4 KB.
- Without using a decoder, 4 bits are used for chip selection, 8 bits are used for location selection within chips.
- Therefore addressable memory is 4 x $2^8$ = 1 KB.
- The unused gap 3 KB is a waste of space capacity.

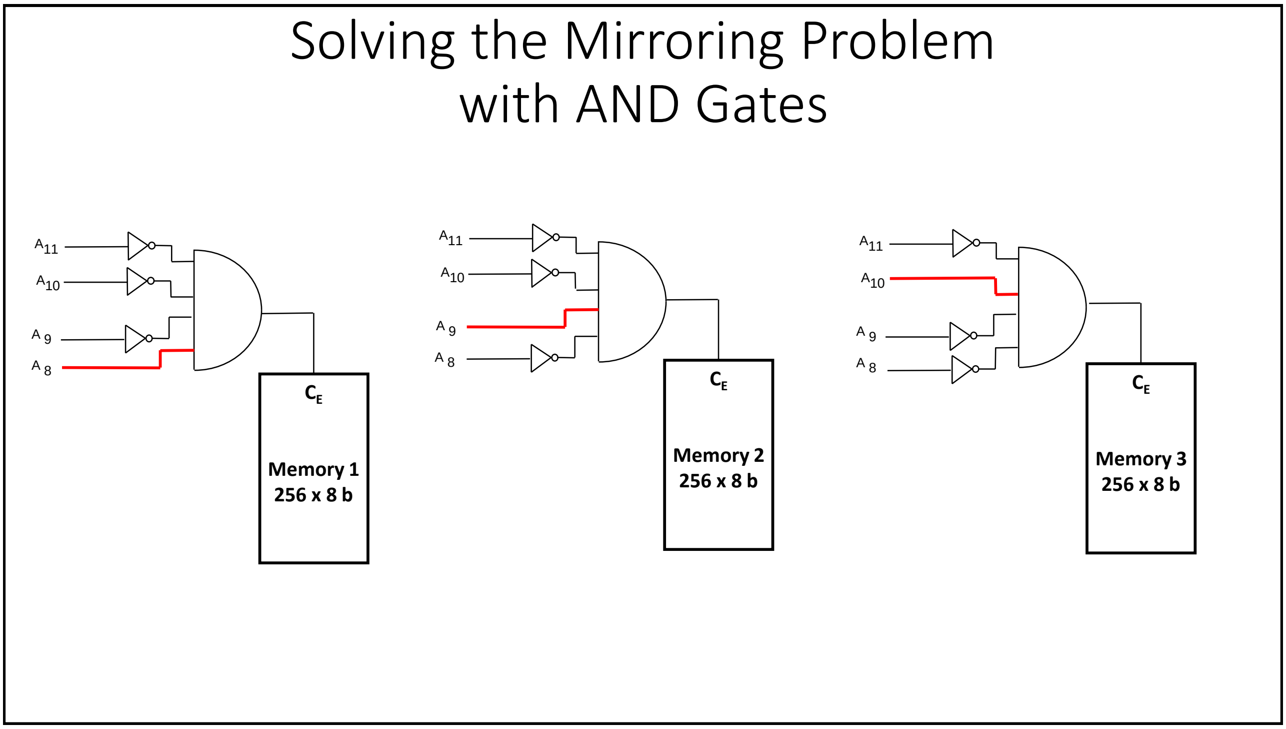


# Mirroring Problem

- If a decoder is not used, or decoder is used but a decoder selection circuit (group selection) is not used, there is a possibility of mirroring.
- If an invalid address (in gap) is used by programmer, more than one memory chip can be selected at the same time. (This is known as **Mirroring problem.**)
- For example: The Adress **F00** will select all of the 4 memory chips. (In fact, any FXX address will always select all chips at the same time.)
- Mirroring causes system deadlock.

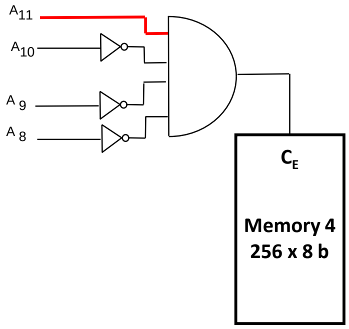
| Example Adress | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F 0 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| SELECTED CHIPS | $A_{11}$ | $A_{10}$ | $A_9$ | $A_8$ | $A_7$ | $A_6$ | $A_5$ | $A_4$ | $A_3$ | $A_2$ | $A_1$ | $A_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



# Solving the Mirroring Problem with AND Gates

# Solving the Mirroring Problem with AND Gates

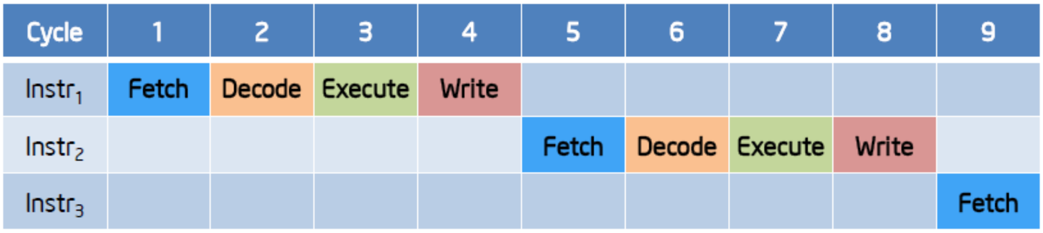


- Normally, an address decoder should be preferred for chip selections.
- This is just an example of alternative solution.

# Lecture 4

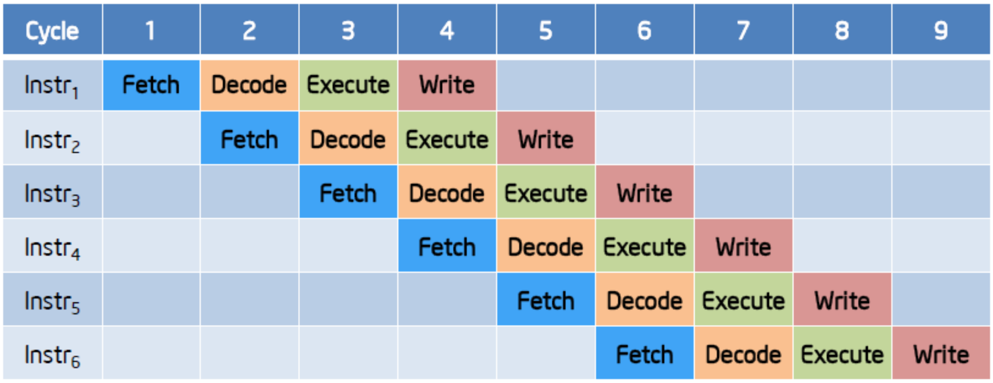
# Sequential Processing

Sequential processing works on one instruction at a time



| Cycle | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Instr$_1$ | Fetch | Decode | Execute | Write | | | | | |
| Instr$_2$ | | | | | Fetch | Decode | Execute | Write | |
| Instr$_3$ | | | | | | | | | Fetch |

# Pipelined Processing



| Cycle | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Instr$_1$ | Fetch | Decode | Execute | Write | | | | | |
| Instr$_2$ | | Fetch | Decode | Execute | Write | | | | |
| Instr$_3$ | | | Fetch | Decode | Execute | Write | | | |
| Instr$_4$ | | | | Fetch | Decode | Execute | Write | | |
| Instr$_5$ | | | | | Fetch | Decode | Execute | Write | |
| Instr$_6$ | | | | | | Fetch | Decode | Execute | Write |

- Latency — elapsed time from start to completion of a particular task
- Throughput — how many tasks can be completed per unit of time
- Pipelining only improves throughput
    - Each job still takes 4 cycles to complete
- Analogy: Factory assembly line