



EE141-Fall 2012 Digital Integrated Circuits

Lecture 5 Overview of Semiconductor Memory

EECS141

Lecture #5

1¹

Announcements

- Homework #2 due today
- Homework #3 due next Thursday

EECS141

Lecture #5

2²

Class Material

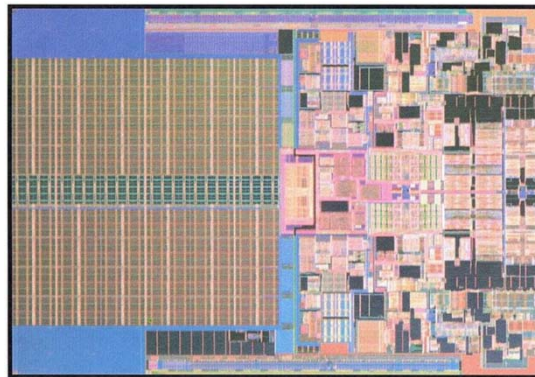
- Last lecture
 - Detailed Switch Model
 - CMOS Gates
 - Design Rules
- Today's lecture
 - Overview of semiconductor memory
- Reading (Chapter 12.1, 12.2.3, 12.3.1)



Semiconductor Memory

Why Memory?

Intel 45nm Core 2



Semiconductor Memory Classification

Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	Non-Random Access	EPROM E ² PROM FLASH	Mask-Programmed Programmable (PROM)
SRAM DRAM	FIFO LIFO Shift Register CAM		

Random Access Memories (RAM)

□ STATIC (SRAM)

Data stored as long as supply is applied
Larger (6 transistors/cell)
Fast
Differential (usually)

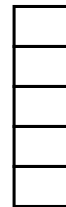
□ DYNAMIC (DRAM)

Periodic refresh required
Smaller (1-3 transistors/cell)
Slower
Single Ended

Random Access Chip Architecture

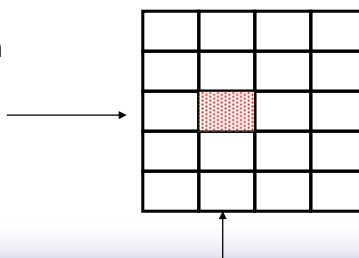
□ Conceptual: linear array

- Each box holds some data
- But this does not lead to a nice layout shape
- Too long and skinny



□ Create a 2-D array

- Decode Row and Column address to get data



Basic Memory Array

CORE:

- keep square within a 2:1 ratio
- rows are **word lines**
- columns are **bit lines**
- data in and out on columns

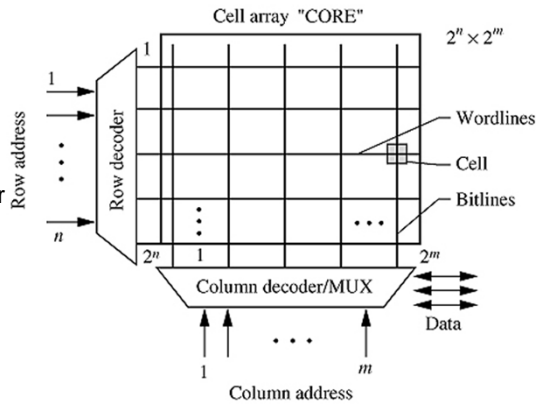
DECODERS:

- needed to reduce total number of pins; N+M address lines for 2^{N+M} bits of storage

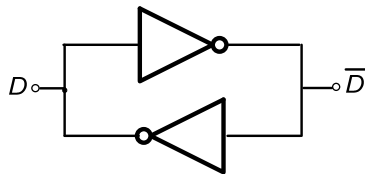
Ex: if N+M=20 $\rightarrow 2^{20} = 1\text{Mb}$

MULTIPLEXING:

- used to select one or more columns for input or output of data

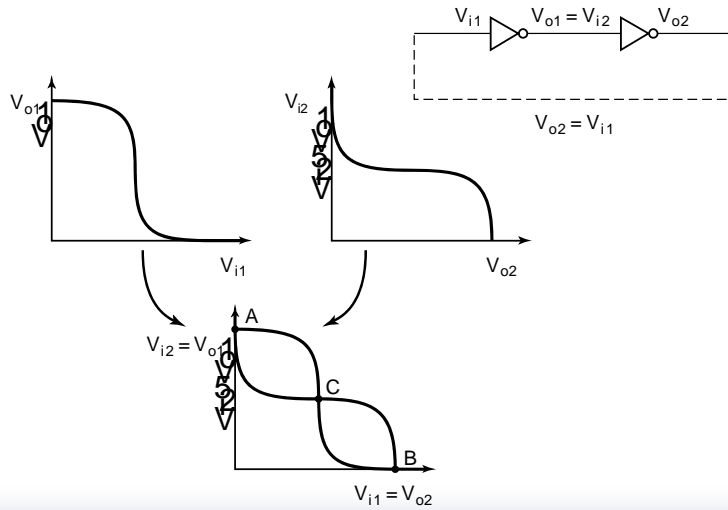


Basic Static Memory Element



- If D is high, \bar{D} will be driven low
 - Which makes D stay high
- Positive feedback

Positive Feedback: Bi-Stability

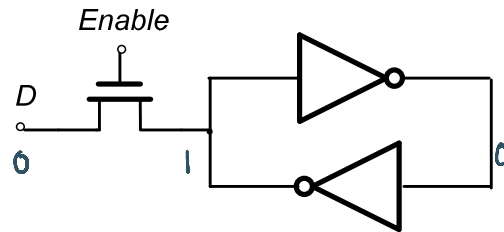


EECS141

Lecture #5

11

Writing into a Cross-Coupled Pair



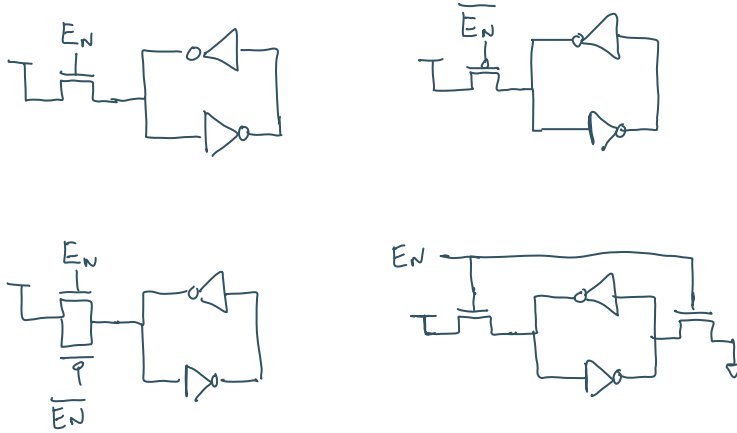
Access transistor must be able to overpower the feedback

EECS141

Lecture #5

12

Writing a "1"

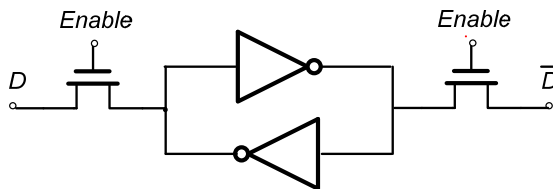


EECS141

Lecture #5

13

Memory Cell



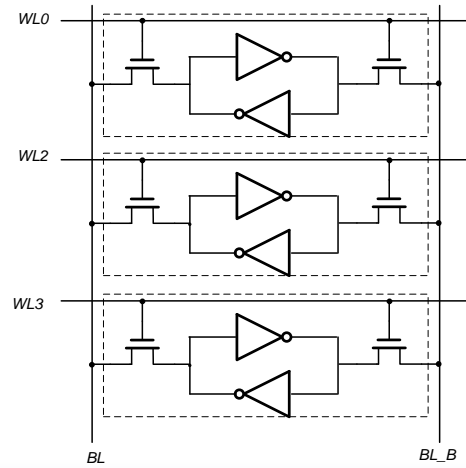
Complementary data values are written (read) from two sides

EECS141

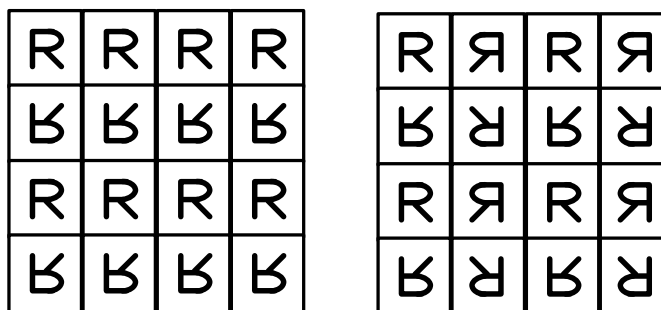
Lecture #5

14

SRAM Column



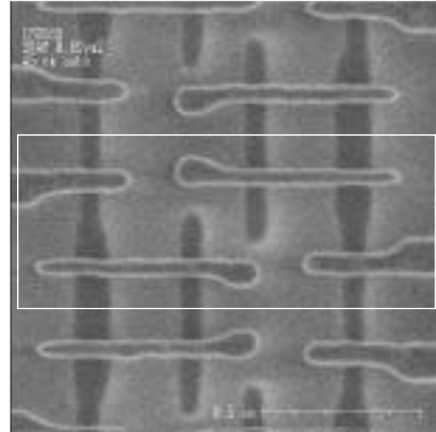
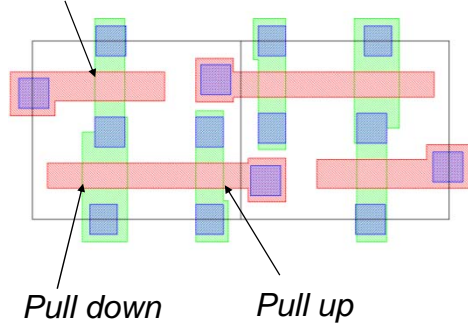
SRAM Array Layout



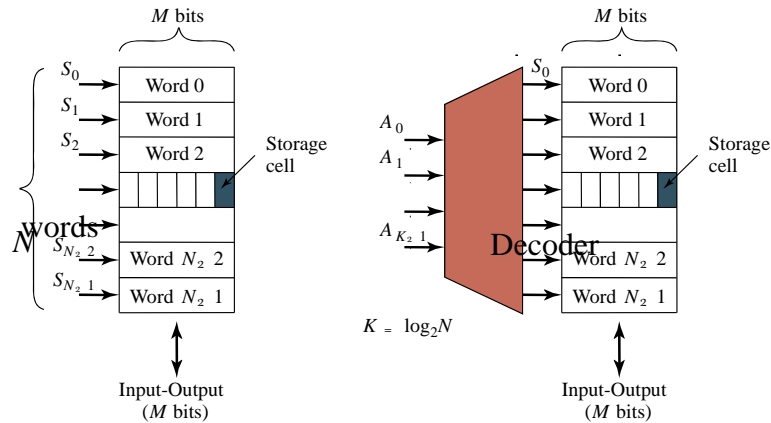
65nm SRAM

□ ST/Philips/Motorola

Access Transistor



Decoders



Intuitive architecture for $N \times M$ memory
 Too many select signals:
 N words == N select signals

Decoder reduces the number of select signals
 $K = \log_2 N$

Row Decoders

Collection of 2^M complex logic gates
Organized in regular and dense fashion

(N)AND Decoder

$$WL_0 = \overline{A_0} \overline{A_1} \overline{A_2} \overline{A_3} \overline{A_4} \overline{A_5} \overline{A_6} \overline{A_7} \overline{A_8} \overline{A_9}$$

$$WL_{511} = A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9$$

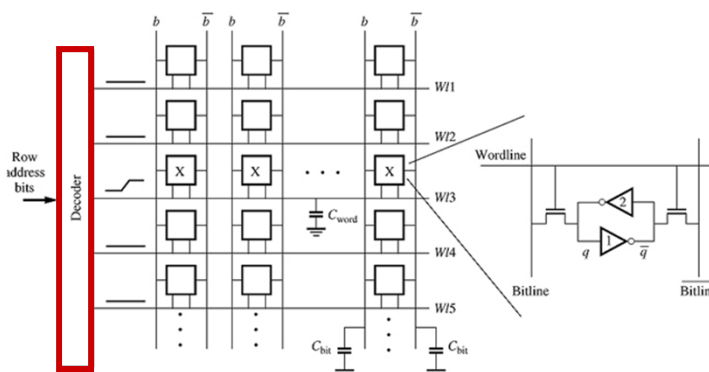
$$WL_{16} =$$

NOR Decoder

$$WL_0 = \overline{A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$

$$WL_{511} = \overline{\overline{A_0} + \overline{A_1} + \overline{A_2} + \overline{A_3} + \overline{A_4} + \overline{A_5} + \overline{A_6} + \overline{A_7} + \overline{A_8} + \overline{A_9}}$$

Decoder Design Example



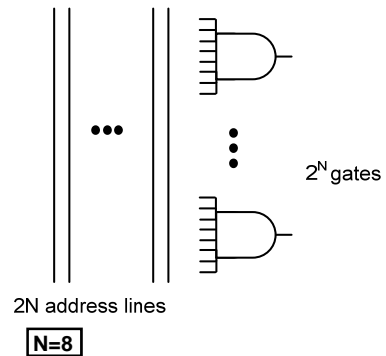
- Look at decoder for 256x256 memory block (8KBytes)

Problem Setup

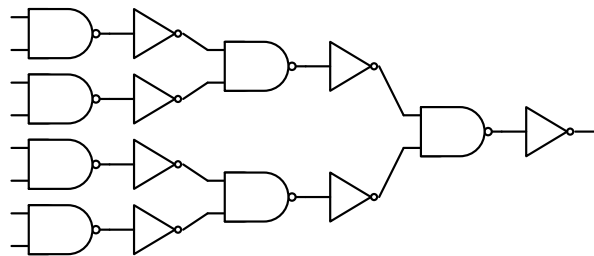
- Goal: Build fastest, lowest possible power decoder with static CMOS logic

- What we know

- Basically need 256 AND gates, each one of them drives one word line



Possible AND8



- Build 8-input NAND gate using 2-input gates and inverters
- Is this the best we can do?
- Is this better than using fewer NAND4 gates?

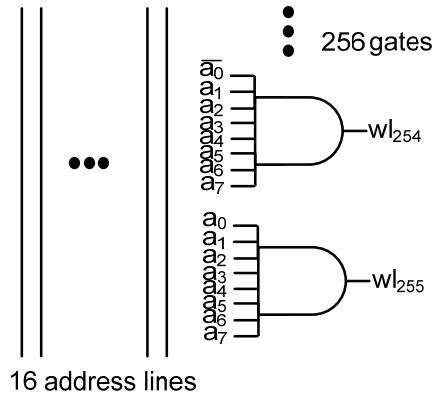
Possible Decoder

- 256 8-input AND gates

- Each built out of tree of NAND gates and inverters

- Need to drive a lot of capacitance (SRAM cells)

- What's the best way to do this?



Next Lecture

- Buffer delay optimization