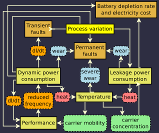
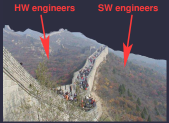
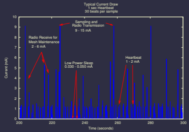
Digital Integrated Circuits – EECS 312

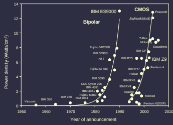
http://robertdick.org/eecs312/

Teacher:	Robert Dick	GSI:	Shengshou Lu
Office:	2417-E EECS	Office:	2725 BBB
Email:	dickrp@umich.edu	Email:	luss@umich.edu
Phone:	734–763–3329		
Cellphone:	847–530–1824		









Review

- What is charge sharing?
- Why are there two different expressions for the voltage to which V_{out} settles?
- Is leakage a significant factor in charge sharing?
- How can it be prevented?
- What is volatile memory?
- What is non-volatile memory?
- What is static memory?
- What is dynamic memory?

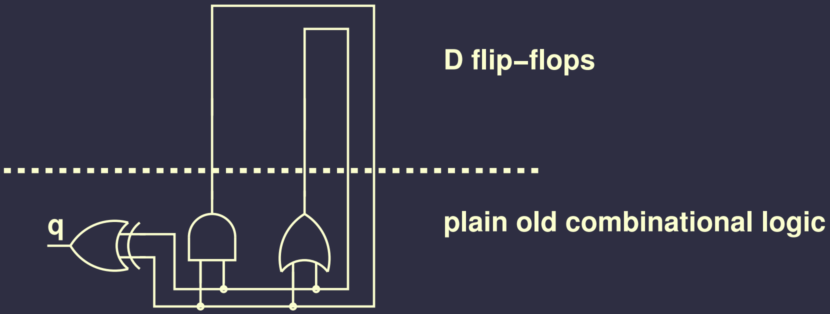
Derive and explain.

Lecture plan

1. Latches and flip-flops
2. Memory array structures
3. Memory array structures
4. Dynamic random access memory
5. Homework

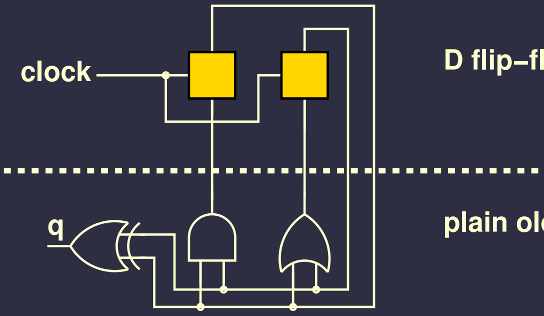
Combinational vs. sequential logic

- No feedback between inputs and outputs – combinational
  - Outputs a function of the current inputs, only



Combinational vs. sequential logic

- No feedback between inputs and outputs – combinational
  - Outputs a function of the current inputs, only
- Feedback – sequential

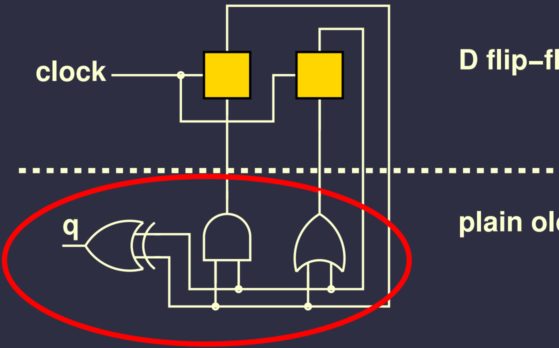


D flip–flops

plain old combinational logic

Combinational vs. sequential logic

- No feedback between inputs and outputs – combinational
  - Outputs a function of the current inputs, only



D flip–flops

plain old combinational logic

Sequential logic

- Outputs depend on current state and (maybe) current inputs
- Next state depends on current state and input
- For implementable machines, there are a finite number of states
- Synchronous
  - State changes upon clock event (transition) occurs
- Asynchronous
  - State changes upon inputs change, subject to circuit delays

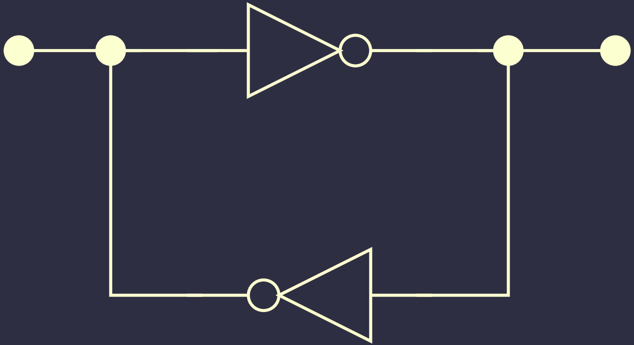
Flip-flop introduction

- Stores, and outputs, a value.
- Puts a special clock signal in charge of timing.
- Allows output to change in response to clock transition.
- More on this later.
  - Timing and sequential circuits

Introduction to sequential elements

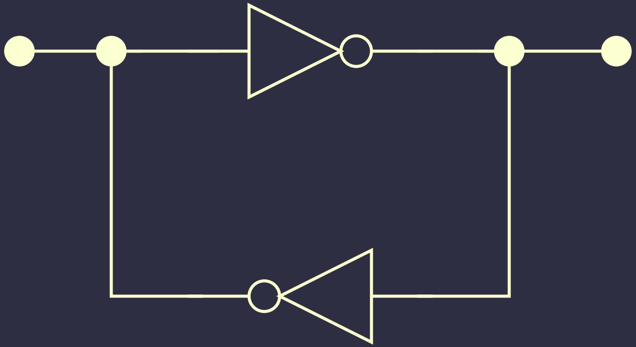
- Feedback and memory.
- Memory.
- Latches.

Feedback and memory



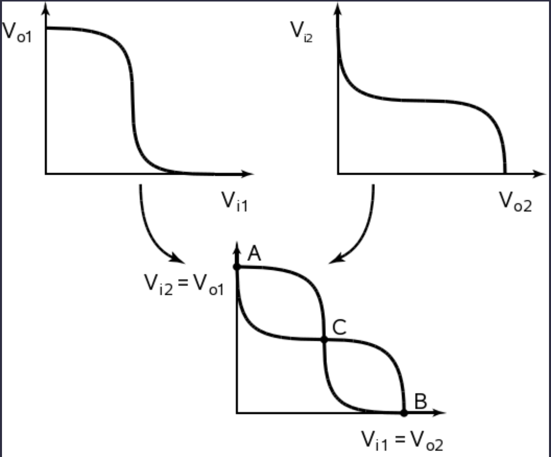
- Feedback or physical state are the root of memory.
- Can compose a simple loop from inverters.

Feedback and memory

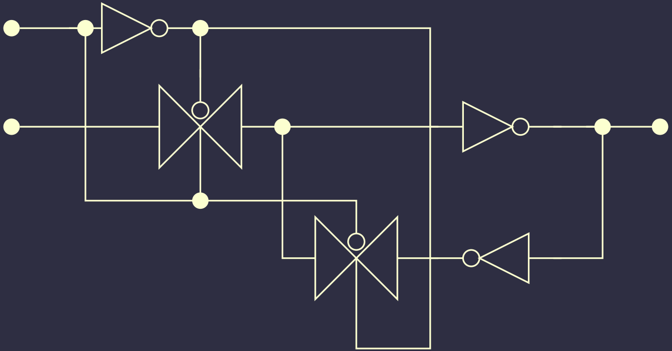


- Feedback or physical state are the root of memory.
- Can compose a simple loop from inverters.
- However, there is no way to switch the value.

Bistability

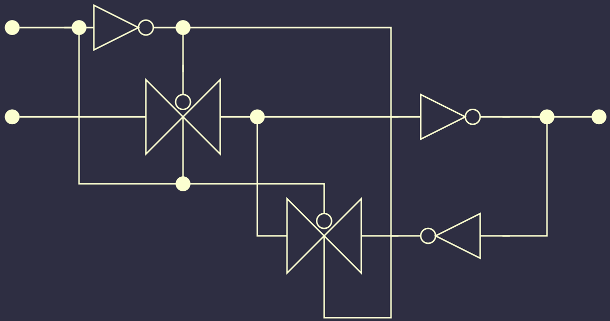


TG and NOT-based memory



- Can break feedback path to load new value
- However, potential for timing problems

TG and NOT-based memory

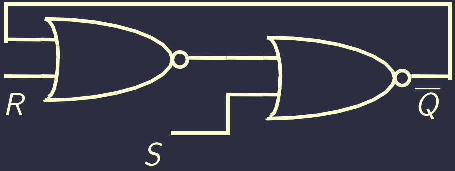


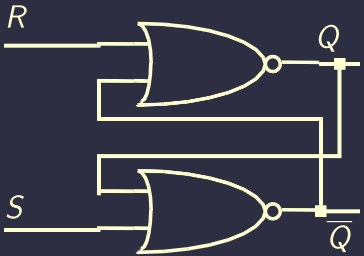
- Can break feedback path to load new value.
- How can this be made more area-efficient?
- Resize transistors, remove transistors, use state?

Section outline

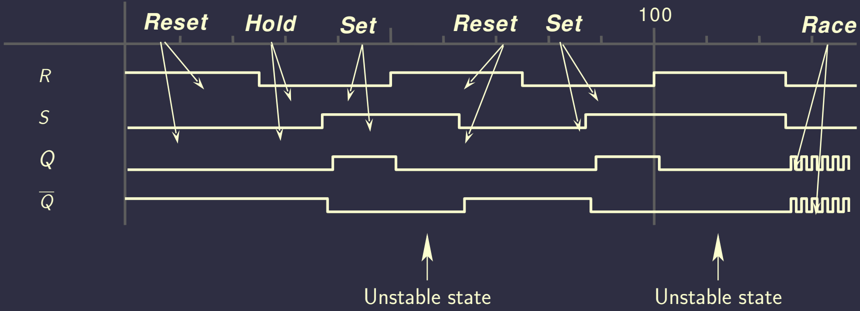
1. Latches and flip-flops
    - Reset/set latches
    - Clocking conventions
    - D flip-flop
    - Other memory elements

Reset/set latch





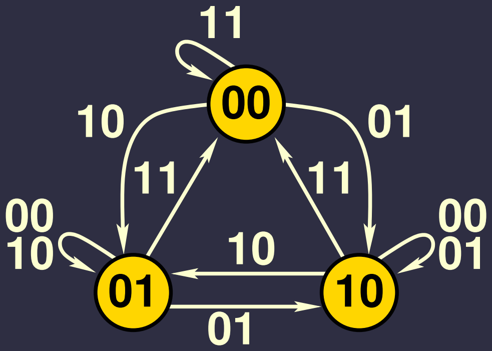
Reset/set timing



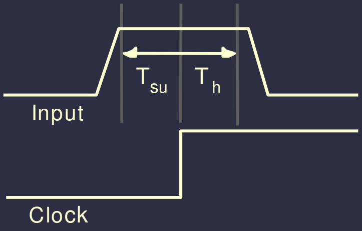
RS latch state diagram

output=$Q\ \overline{Q}$

input=$R\ S$

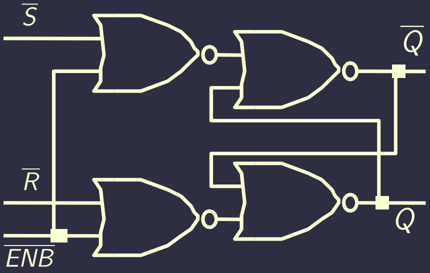


Clocking terms

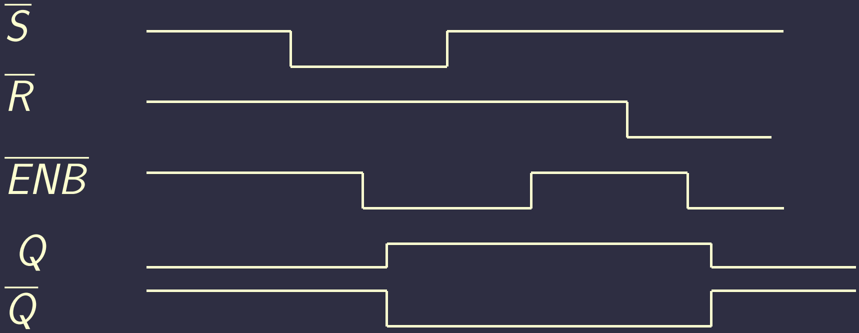


- Clock – Rising edge, falling edge, high level, low level, period
- Setup time: Minimum time before clocking event by which input must be stable (T_{SU})
- Hold time: Minimum time after clocking event for which input must remain stable (T_H)
- Window: From setup time to hold time

Gated RS latch



Gated RS latch



Memory element properties

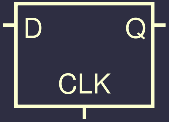
Type	Inputs sampled	Outputs valid
Unclocked latch	Always	LFT
Level-sensitive latch	Clock high (T_{SU} to T_H) around falling clock edge	LFT
Edge-triggered flip-flop	Clock low-to-high transition (T_{SU} to T_H) around rising clock edge	Delay from rising edge

Section outline

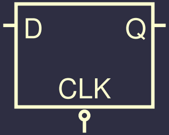
1. Latches and flip-flops
   - Reset/set latches
   - Clocking conventions
   - D flip-flop
   - Other memory elements

Clocking conventions

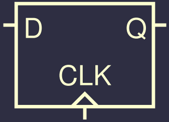
Active–high transparent



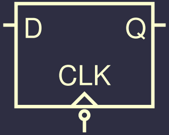
Active–low transparent



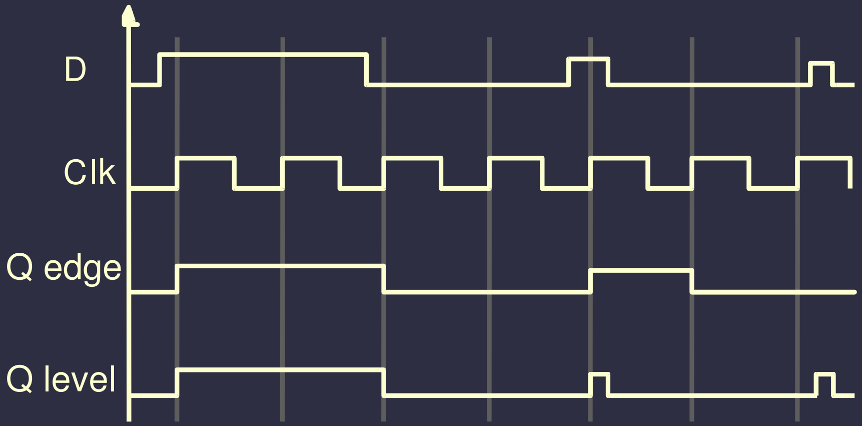
Positive (rising) edge



Negative (falling) edge



Timing for edge and level-sensitive latches

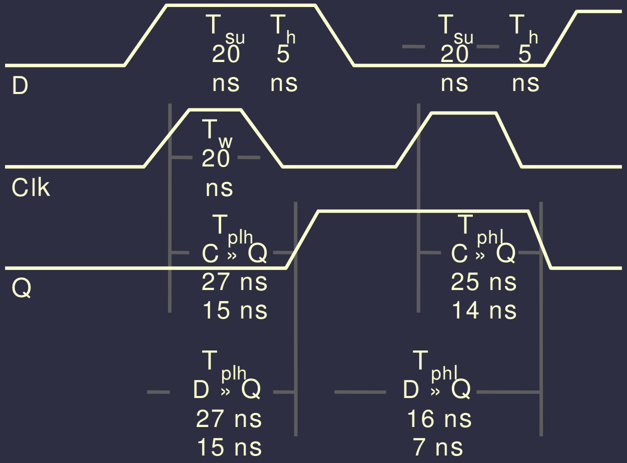


Latch timing specifications

- Minimum clock width, T_W
  - Usually period / 2
- Low to high propegation delay, P_{LH}
- High to low propegation delay, P_{HL}
- Worst-case and typical

Latch timing specifications

Example, negative (falling) edge-triggered flip-flop timing diagram

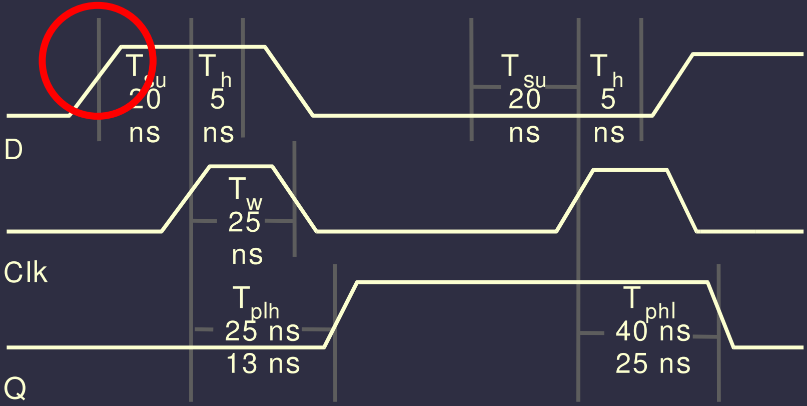

D
Clk
Q
T_{su} 20 ns
T_h 5 ns
T_{su} 20 ns
T_h 5 ns
T_w 20 ns
T_{plh} C » Q 27 ns 15 ns
T_{phl} C » Q 25 ns 14 ns
T_{plh} D » Q 27 ns 15 ns
T_{phl} D » Q 16 ns 7 ns

FF timing specifications

- Minimum clock width, T_W
  - Usually period / 2
- Low to high propagation delay, P_{LH}
- High to low propagation delay, P_{HL}

FF timing specifications

Example, positive (rising) edge-triggered flip-flop timing diagram



RS latch states

S	R	Q^+	$\overline{Q}^+$	Notes
0	0	Q	$\overline{Q}$	
0	1	0	1	
1	0	1	0	
1	1	1	1	unstable

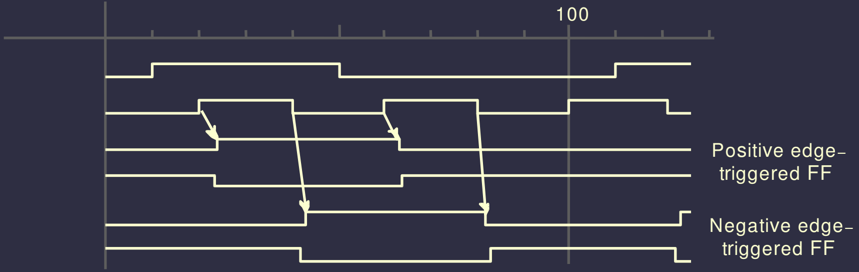
Section outline

1. Latches and flip-flops
   - Reset/set latches
   - Clocking conventions
   - D flip-flop
   - Other memory elements

Falling edge-triggered D flip-flop

- Use two stages of latches
- When clock is high
  - First stage samples input w.o. changing second stage
  - Second stage holds value
- When clock goes low
  - First stage holds value and sets or resets second stage
  - Second stage transmits first stage
- $Q^+ = D$
- One of the most commonly used flip-flops

Edge triggered timing



RS clocked latch

- Storage element in narrow width clocked systems.
- Dangerous.
- Fundamental building block of many flip-flop types.

D flip-flop

- Minimizes input wiring.
- Simple to use.
- Common choice for basic memory elements in sequential circuits.

Toggle (T) flip-flops

- State changes each clock tick
- Useful for building counters
- Can be implemented with other flip-flops
    - D with XOR feedback

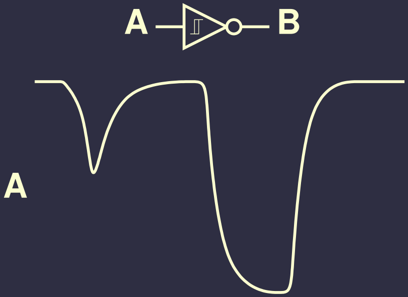
Asynchronous inputs

- How can a circuit with numerous distributed edge-triggered flip-flops be put into a known state?
- Could devise some sequence of input events to bring the machine into a known state.
  - Complicated.
  - Slow.
  - Not necessarily possible, given trap states.
- Can also use sequential elements with additional asynchronous reset and/or set inputs.

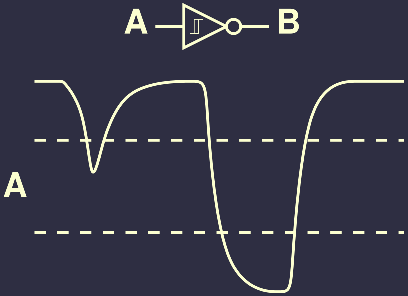
Section outline

1. Latches and flip-flops
   - Reset/set latches
   - Clocking conventions
   - D flip-flop
   - Other memory elements

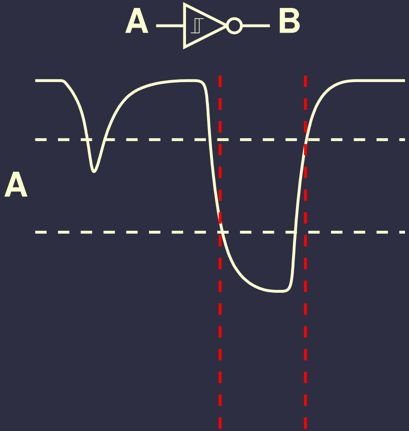
Schmitt triggers



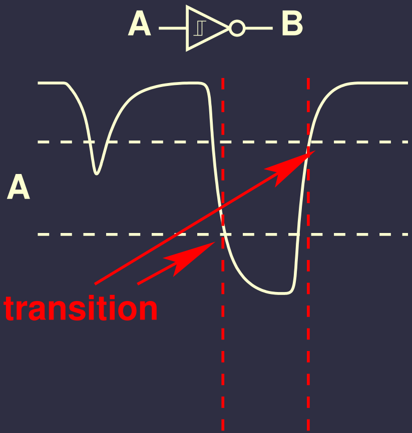
Schmitt triggers



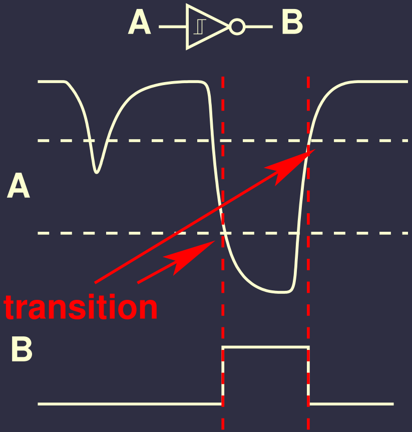
Schmitt triggers



Schmitt triggers



Schmitt triggers



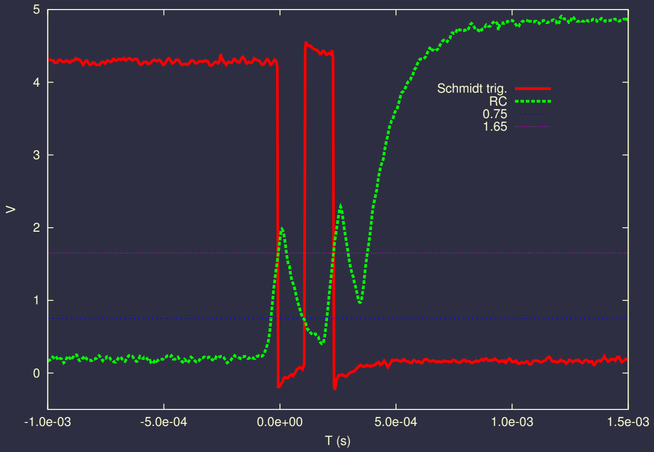
Reason for gradual transition

- A logic stage is an RC network
- Whenever a transition occurs, capacitance is driven through resistance
- Consider the implementation of a CMOS inverter

Debouncing

- Mechanical switches bounce!
- What happens if multiple pulses?
  - Multiple state transitions
- Need to clean up signal

Debouncing


5
4
3
2
1
0
V
-1.0e-03
-5.0e-04
0.0e+00
5.0e-04
1.0e-03
1.5e-03
T (s)
Schmidt trig.
RC
0.75
1.65

Latch and flip-flop equations

RS

$$Q^+ = S + \overline{R}\ Q$$

D

$$Q^+ = D$$

T

$$Q^+ = T \oplus Q$$

Review

- What are t_{su} and t_h?
- Define
  - Level-sensitive.
  - Edge-triggered.
  - Latch.
  - Flip-flop.
- What is the symbol for a falling edge triggered D flip-flop?
- Show a circuit design for a Schmitt-trigger inverter.

Derive and explain.

Distributed loads and Elmore delay

Derive the propagation delay of an aluminum wire that is 2 cm long and 500 nm wide. Does using a lumped model introduce significant error? You may assume a sheet resistance of $0.075\,\Omega/\square$. Derive the propagation delay of a copper wire with the same shape. State, and verify, any assumptions.

Derive and explain.

More on transistor sizing

$$f(a, b, c) = \overline{ab + c}$$

Derive and explain.

Lecture plan

1. Latches and flip-flops
2. Memory array structures
3. Memory array structures
4. Dynamic random access memory
5. Homework

Volatile memory

- SRAM cell and architecture overview.
- DRAM cell and architecture overview.

Non-volatile memory

- ROM.
- EPROM.
- EEPROM.
- Flash.

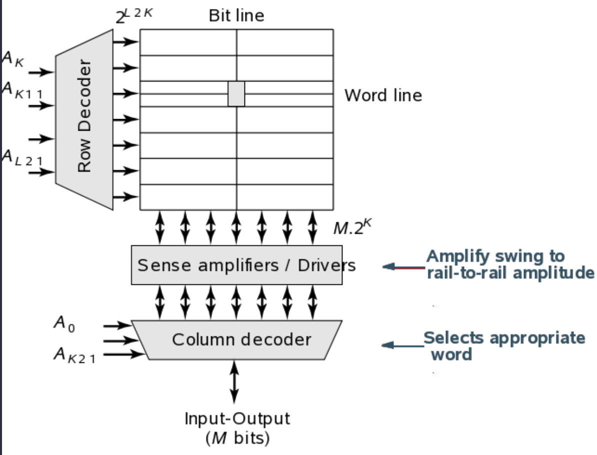
Floating gate technology

- UV erase.
- Electrical erase.
- Block erase.

Hot floating gate implementation

- Was once difficult to design uniform-thickness thin oxide layers.
- Tunneling-based programming was difficult.
- Avalanche injection (hot electron) based programming used.
- UV erasure.
- Pure tunneling later became practical (EEPROM).
- Flash uses hot electrons for programming and tunneling for erasing.

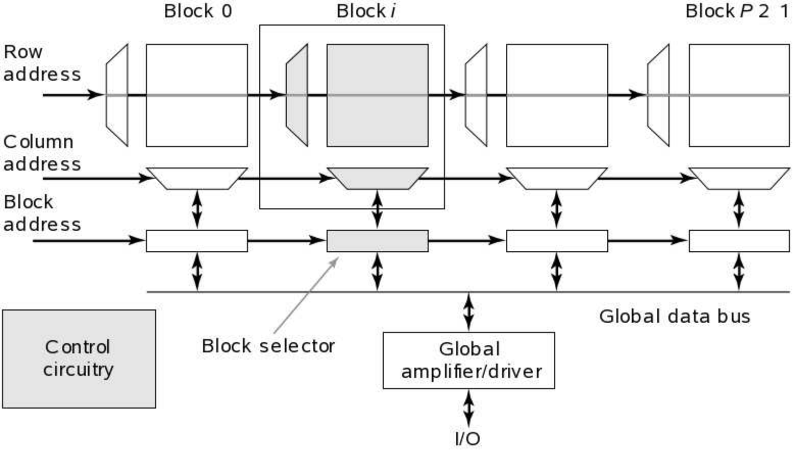
Array memory architecture



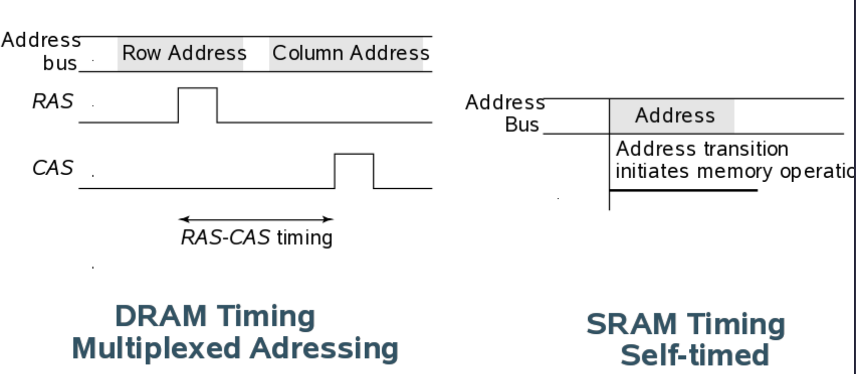
Amplify swing to rail-to-rail amplitude

Selects appropriate word

Block-based memory architecture



Memory timing



Review

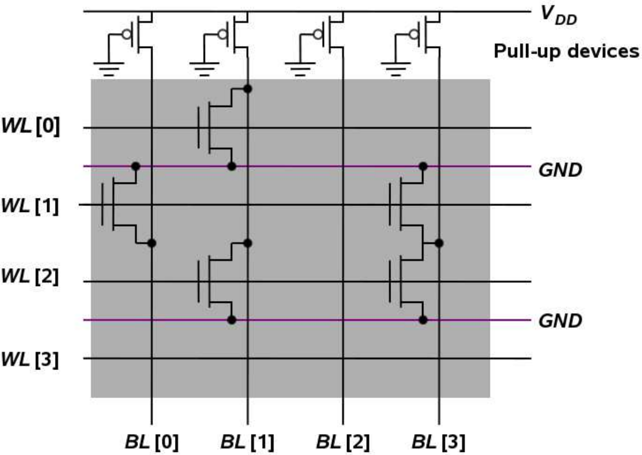
- What are the different ways a floating-gate memory cell can be erased?
- What are the different ways a floating-gate memory cell can be programmed?
- What are the two main DRAM bit cell organizations, and their advantages?
- Why is it difficult to economically put DRAM on the same die as a processor?
- Why are decoders and MUXs used in memory arrays?

Derive and explain.

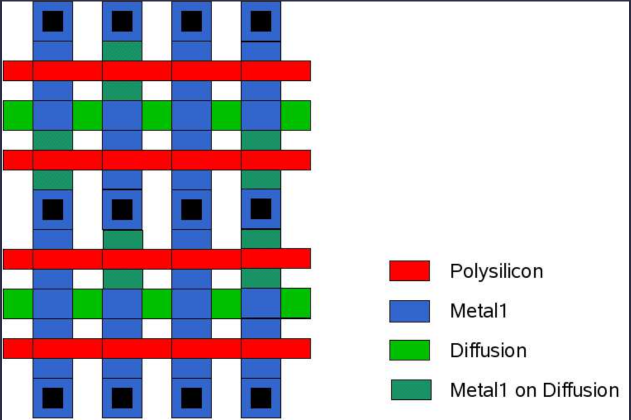
Lecture plan

1. Latches and flip-flops
2. Memory array structures
3. Memory array structures
4. Dynamic random access memory
5. Homework

NOR ROM schematic

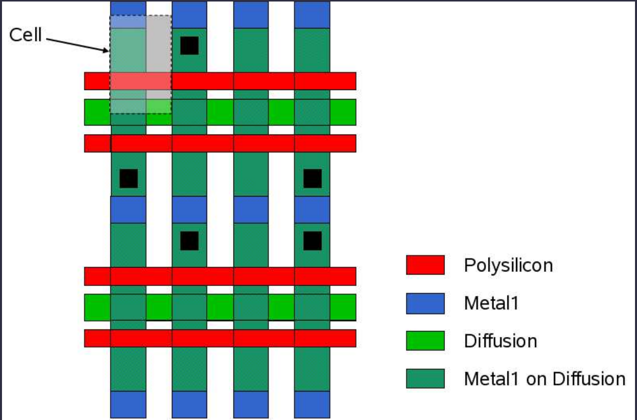


NOR ROM layout



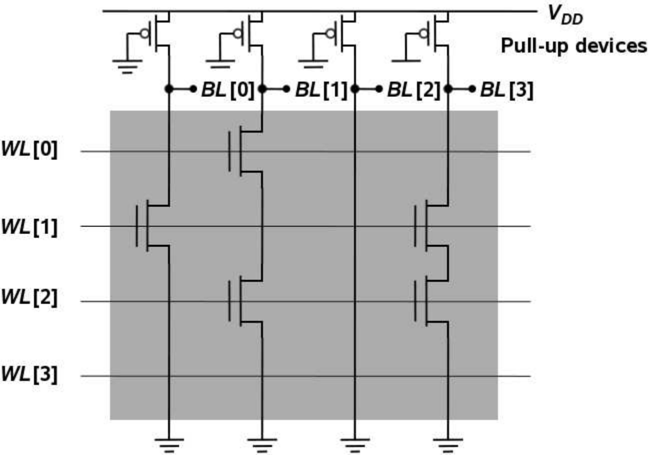
Program using active layer.

NOR ROM layout

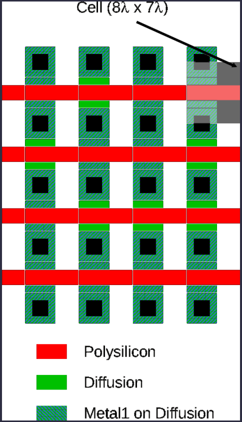


Program using contacts.

NAND ROM schematic

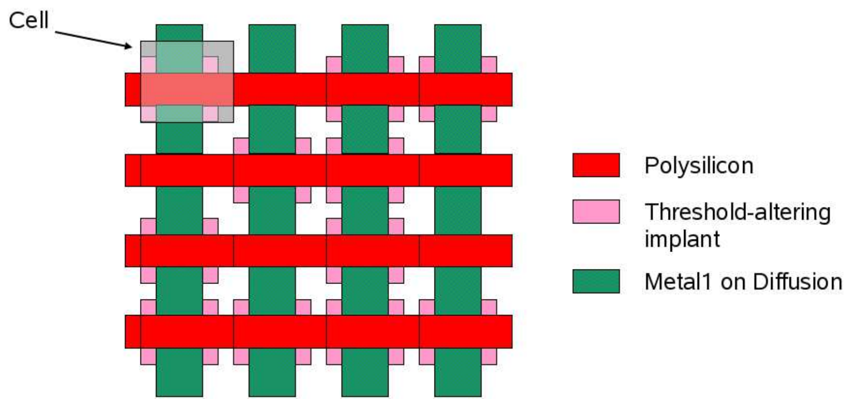


NAND ROM layout



Program using metal layer.

NAND ROM layout

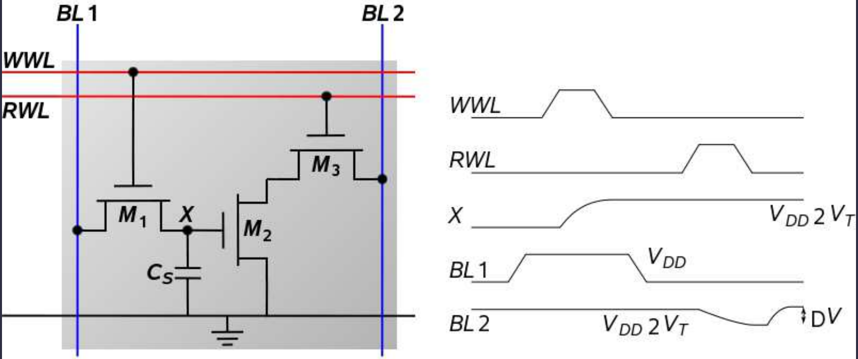


Program using implants.

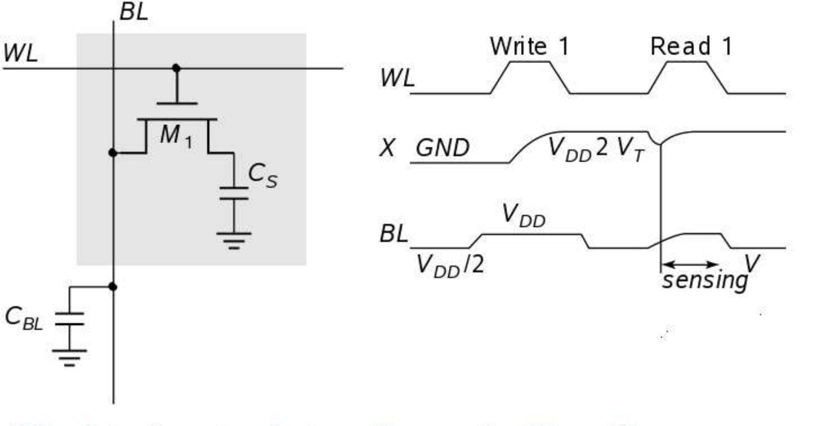
Lecture plan

1. Latches and flip-flops
2. Memory array structures
3. Memory array structures
4. Dynamic random access memory
5. Homework

DRAM

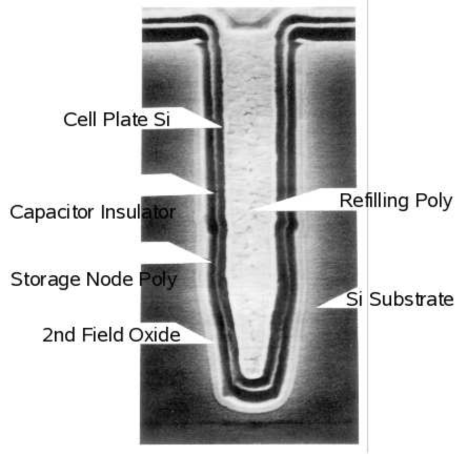

BL1
BL2
WWL
RWL
M_1
M_2
M_3
X
C_S
WWL
RWL
X
V_{DD} 2 V_T
BL1
V_{DD}
BL2
V_{DD} 2 V_T
DV

DRAM

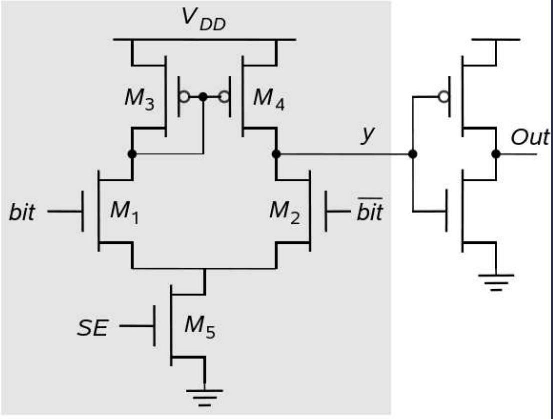


Write: C_S is charged or discharged by asserting WL and BL.

DRAM side view

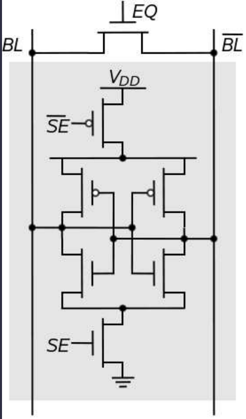


Differential sense amplifier



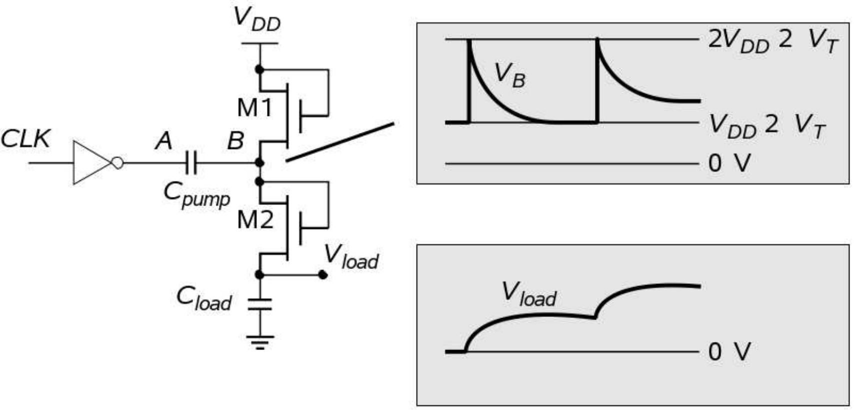
Useful for SRAM, can use two stages.

Latch sense amplifier



Useful for DRAM.

Charge pump


V_{DD}
M1
CLK
A
B
C_{pump}
M2
V_{load}
C_{load}
$2V_{DD}$ 2 V_T
V_B
V_{DD} 2 V_T
0 V
V_{load}
0 V

Upcoming topics

- Theoretical foundations for sizing.

Lecture plan

1. Latches and flip-flops
2. Memory array structures
3. Memory array structures
4. Dynamic random access memory
5. Homework

Homework assignment I

- 31 October: Read Sections 6.3 and 7.1 in J. Rabaey,

  A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*.
  Prentice-Hall, second edition, 2003.
- 7 November: Read Sections 7.2.2, 7.2.3, 7.3.1, 7.3.2, and 7.6.1

  in J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*.
  Prentice-Hall, second edition, 2003.
- 7 November: Project 4.

Homework assignment II

- 12 November: Read Sections 12.1.1, 12.1.2, and 12.2.1 in

  J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*.
  Prentice-Hall, second edition, 2003.
- 14 November: Read Sections 12.3.1, 12.3.2, 12.2.2, and 12.2.3 in

  J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*.
  Prentice-Hall, second edition, 2003.
- 16 November: Homework 4.