V-code: ???

First-Order Transients in Linear Electrical Networks (2)

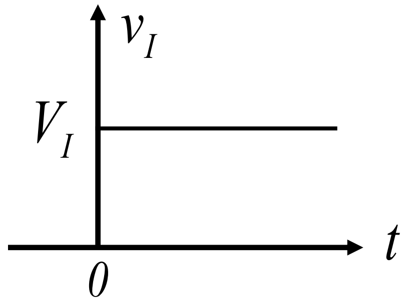
Lecture 13
November 5th, 2018

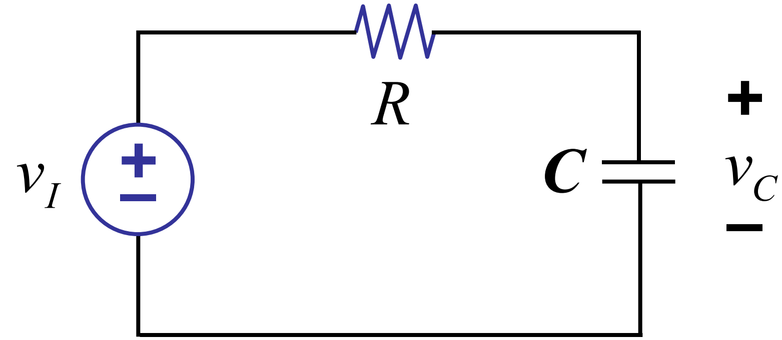
Jae W. Lee (jaewlee@snu.ac.kr)
Computer Science and Engineering
Seoul National University

***Slide credits**: Prof. Anant Agarwal at MIT*

V-code: ???

Review: Analysis of RC Circuits

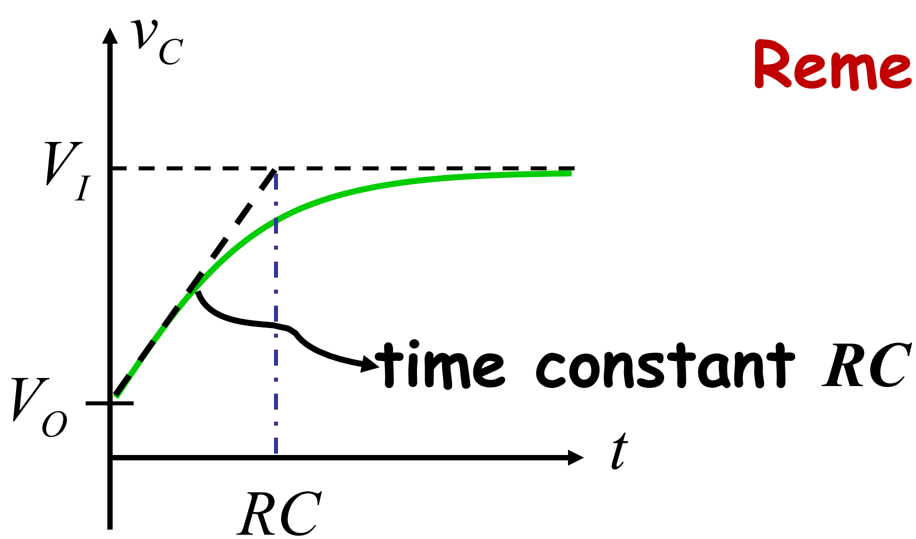




$$v_C(0) = V_O$$

$$v_C = V_I + (V_O - V_I)\, e^{\frac{-t}{RC}}$$ 1

Remember this!



V-code: ???

Outline

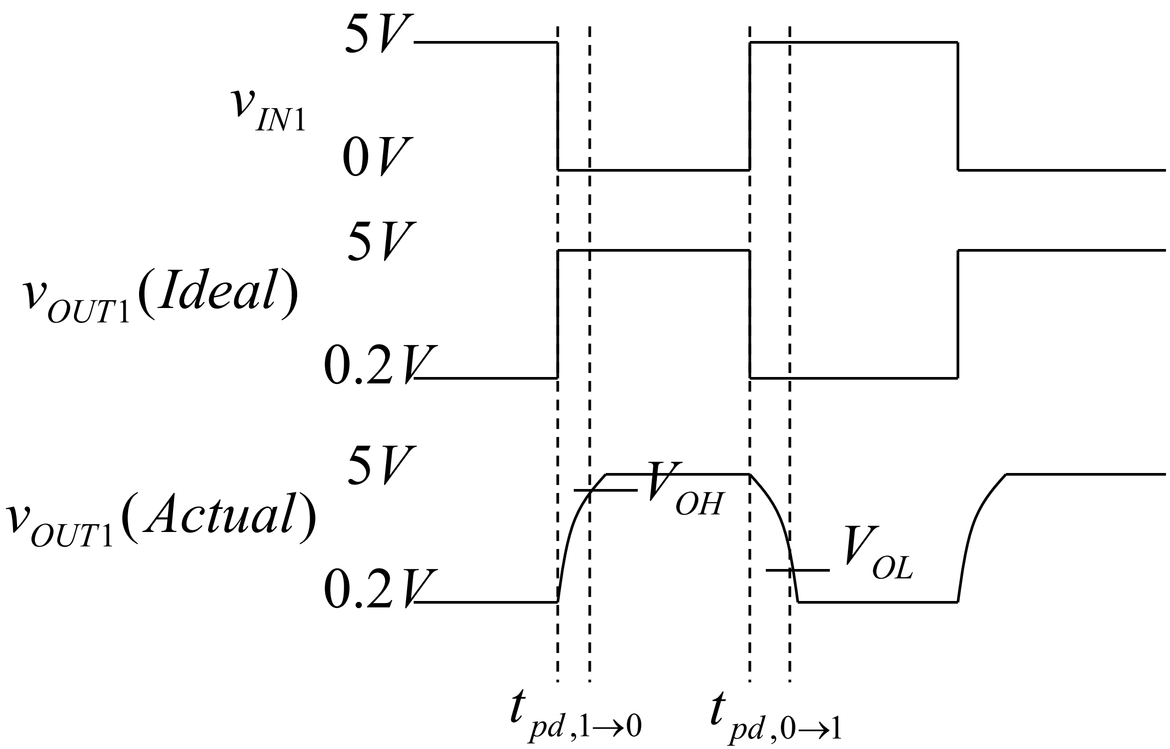
Textbook: 10.4, 10.5, 10.6.2, 10.7

- **Propagation Delay and the Digital Abstraction**
- State and State Variables
- Digital Memory

V-code: ???

Propagation Delay

- **Definitions**



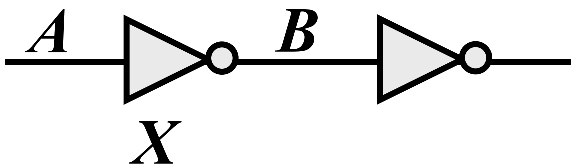
Propagation delay $t_{pd} = \max(t_{pd,1\to 0}, t_{pd,0\to 1})$

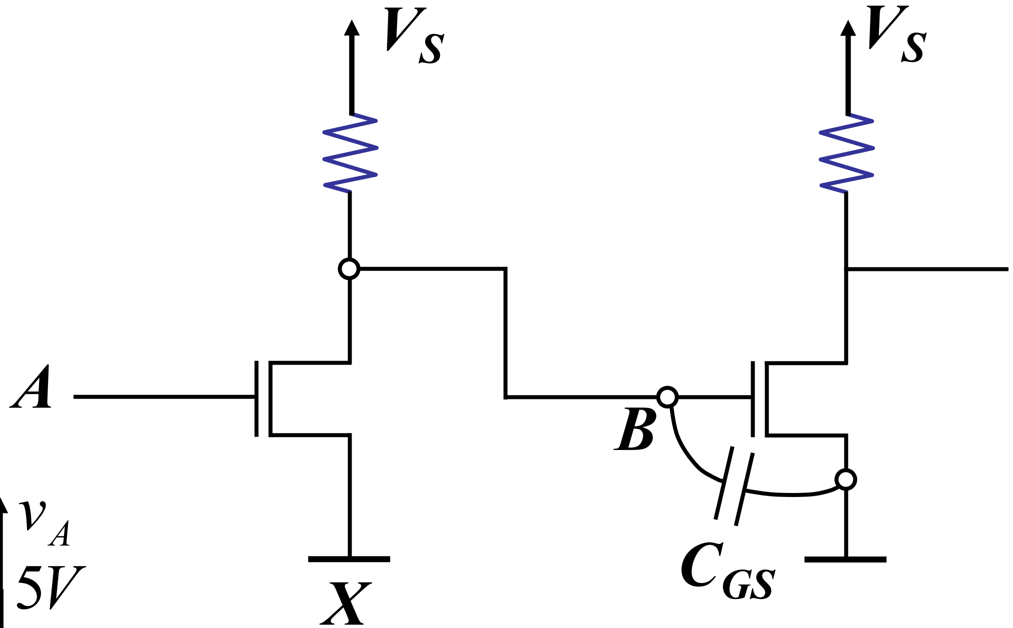
V-code: ???

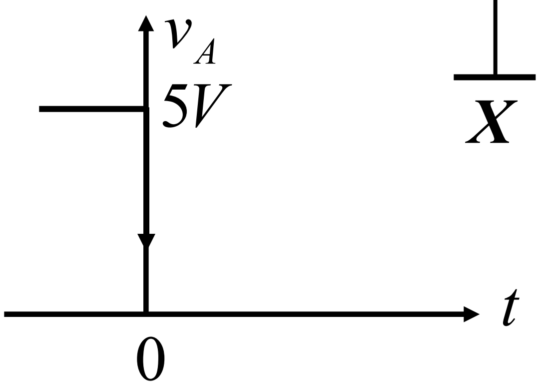
Propagation Delay

Let's apply the result to an inverter.

First, rising delay t_r at B





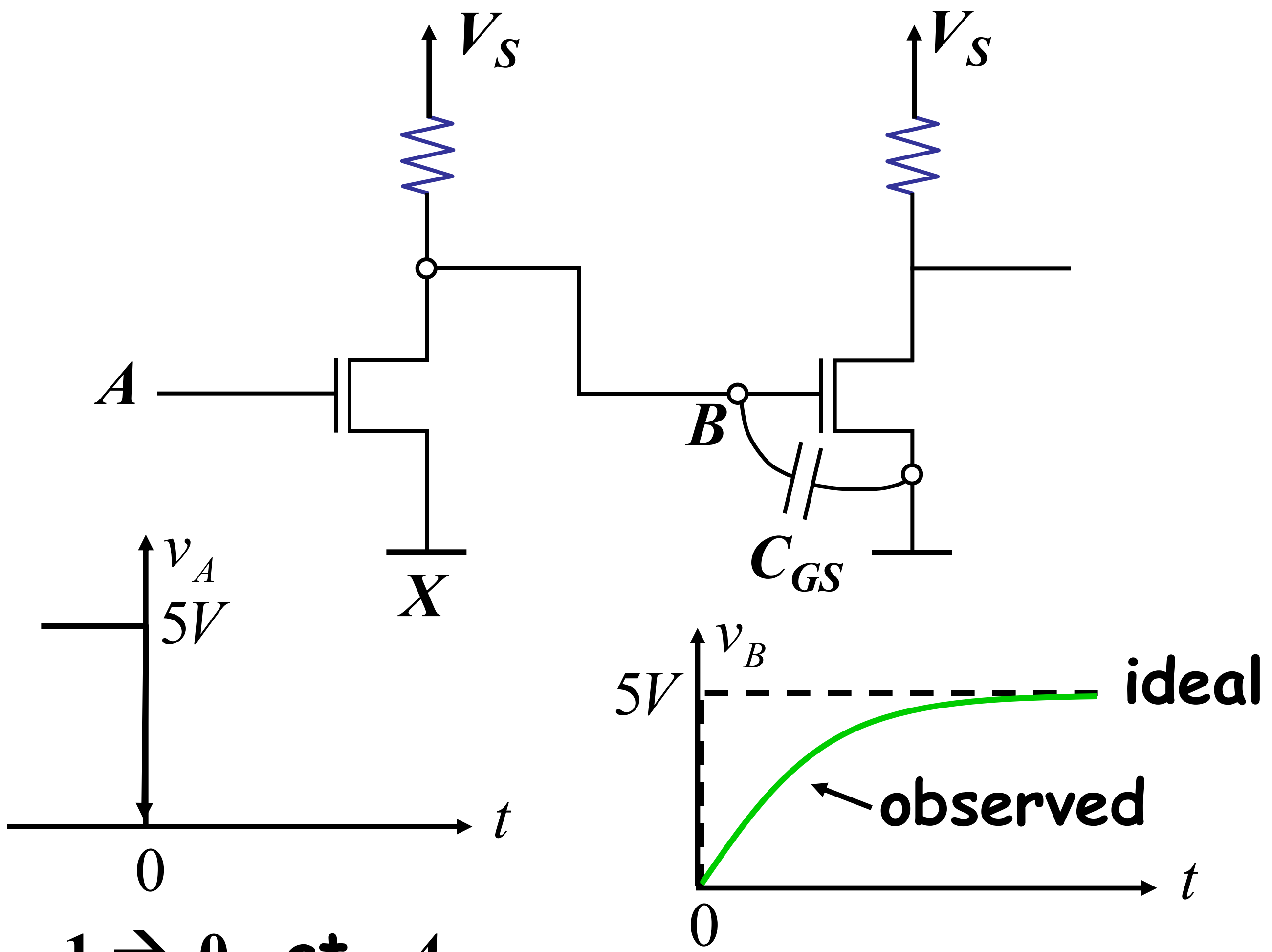


$1 \rightarrow 0$ at A

V-code: ???

Propagation Delay

First, rising delay t_r at B

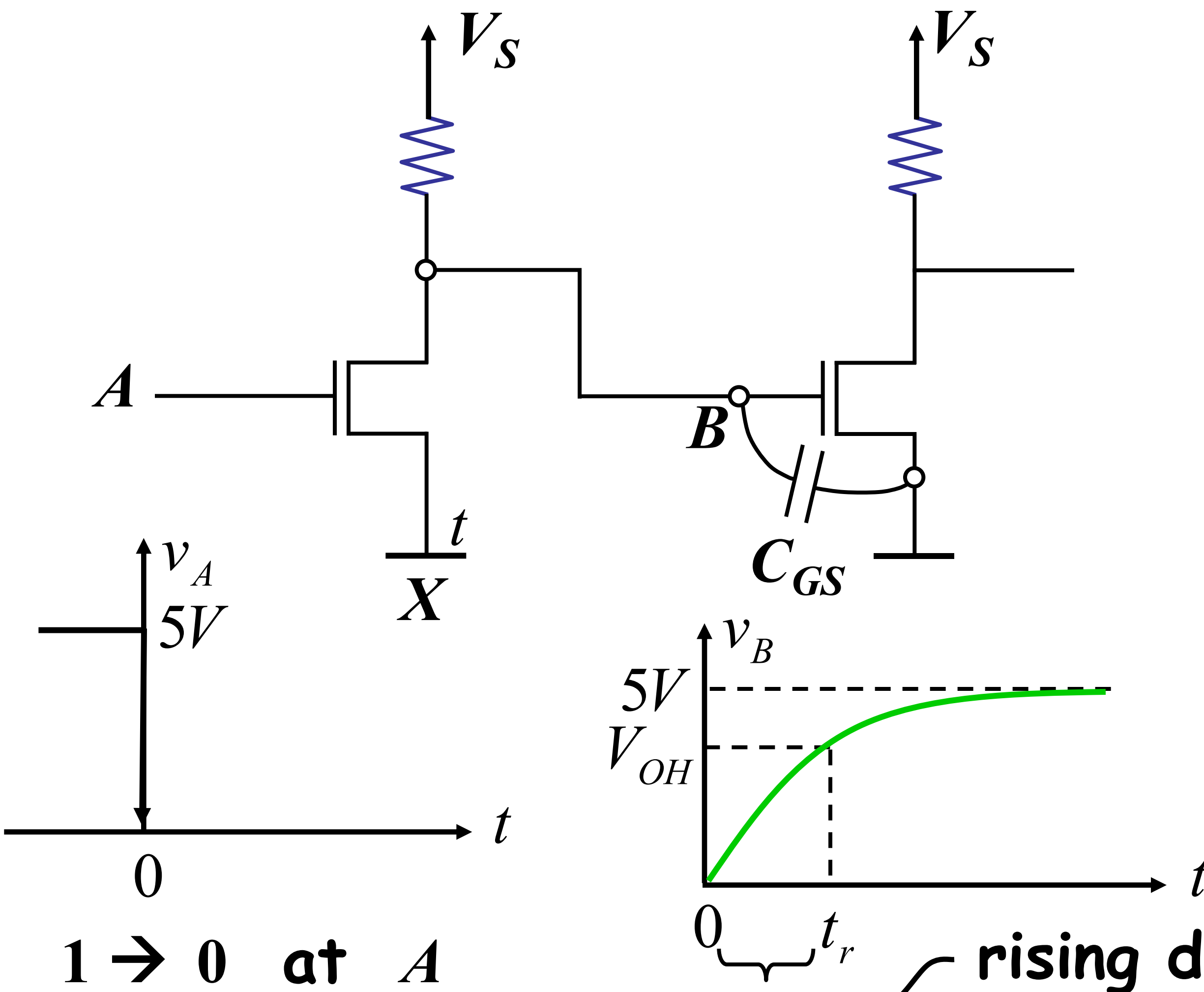


$1 \rightarrow 0$ at A

V-code: ???

Propagation Delay

First, rising delay t_r at B



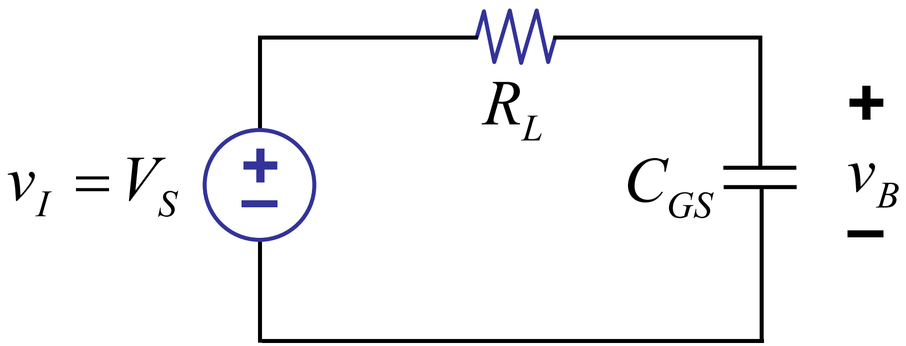
1 → 0 at A

rising delay of X

V-code: ???

Propagation Delay

Equivalent circuit for $0 \rightarrow 1$ at B



$$v_I = V_S, \quad v_B(0) = 0 \quad \text{for } t \geq 0$$

From ①

$$v_B = V_S + (0 - V_S)\, e^{\frac{-t}{R_L C_{GS}}}$$

Now, we need to find t for which $v_B = V_{OH}$.

V-code: ???

Propagation Delay

First, rising delay t_r at B

Or

$$v_{OH} = V_S - V_S\, e^{\frac{-t}{R_L C_{GS}}}$$

Find t_r **:**

$$V_S\, e^{\frac{-t_r}{R_L C_{GS}}} = V_S - V_{OH}$$

$$\frac{-t_r}{R_L C_{GS}} = \ln \frac{V_S - V_{OH}}{V_S}$$

$$t_r = -R_L C_{GS} \ln \frac{V_S - V_{OH}}{V_S}$$

V-code: ???

Propagation Delay

First, rising delay t_r at B

Or

$$v_{OH} = V_S - V_S\, e^{\frac{-t}{R_L C_{GS}}}$$

Find t_r :

$$V_S\, e^{\frac{-t_r}{R_L C_{GS}}} = V_S - V_{OH}$$

$$\frac{-t_r}{R_L C_{GS}} = \ln \frac{V_S - V_{OH}}{V_S}$$

$$t_r = -R_L C_{GS} \ln \frac{V_S - V_{OH}}{V_S}$$

e.g.

$$R_L = 1K \qquad V_S = 5V$$

$$C_{GS} = 0.1\, pF \qquad V_{OH} = 4V$$

$$t_r = -1 \times 10^3 \times 0.1 \times 10^{-12}\, ln \frac{5-4}{5}$$

$$= 0.16\, ns$$

$$RC = 0.1\, ns\,!$$

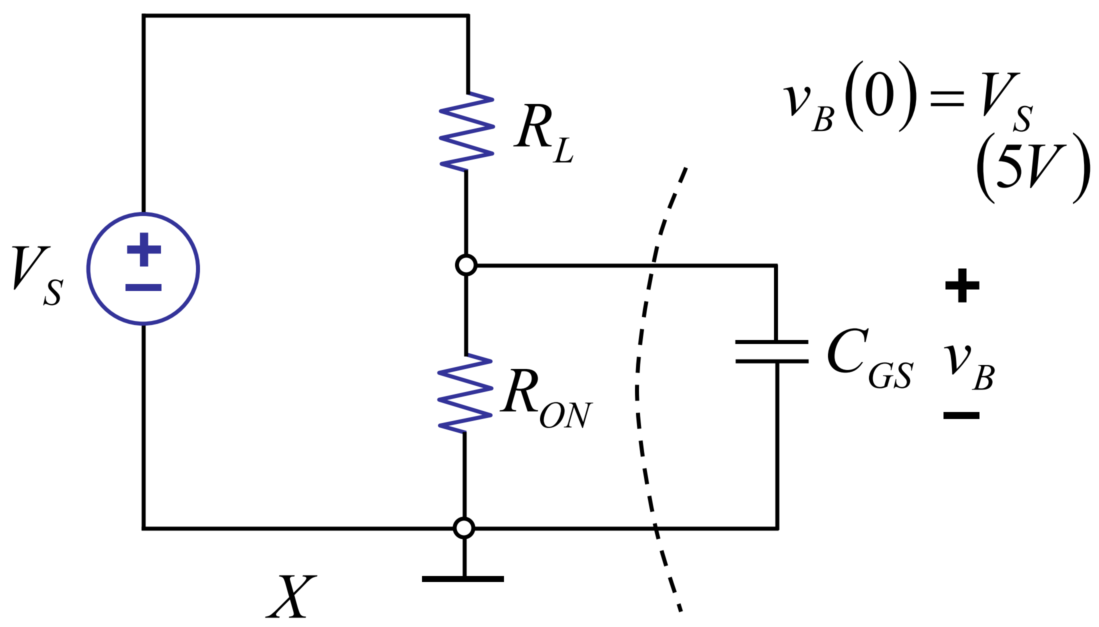
V-code: ???

Propagation Delay

Falling Delay t_f

Falling delay t_f is
the t for which v_B falls to V_{OL}

Equivalent circuit for 1 → 0 at *B*

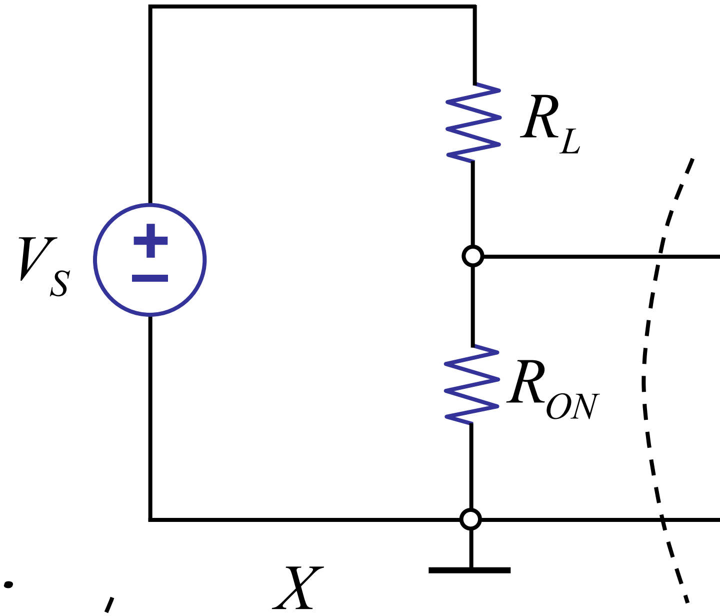


V-code: ???

Propagation Delay

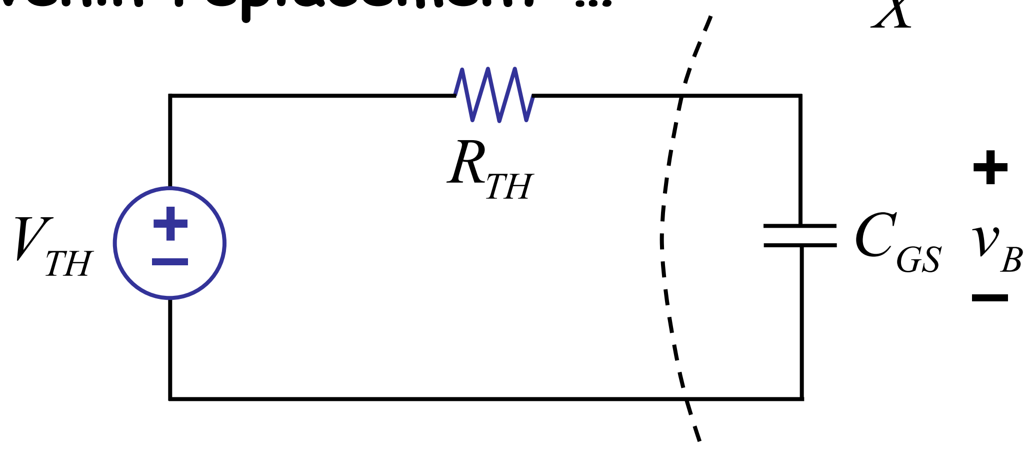
Falling Delay t_f

Equivalent circuit for $1 \rightarrow 0$ at B



$$v_B(0) = V_S \; (5V)$$

Thévenin replacement …



$$R_{TH} = R_L \parallel R_{ON}$$

$$V_{TH} = V_S \frac{R_{ON}}{R_{ON} + R_L}$$

V-code: ???

Propagation Delay

Falling Delay t_f

From 1

$$v_B = V_{TH} + (V_S - V_{TH})\, e^{\frac{-t}{R_{TH}C_{GS}}}$$

Falling decay t_f is
the t for which v_B falls to V_{OL}

$$V_{OL} = V_{TH} + (V_S - V_{TH})\, e^{\frac{-t_f}{R_{TH}C_{GS}}}$$

or

$$t_f = -R_{TH}C_{GS} \ln \frac{V_{OL} - V_{TH}}{V_S - V_{TH}}$$

V-code: ???

Propagation Delay

Falling Delay t_f

$$t_f = -R_{TH} C_{GS} \ln \frac{V_{OL} - V_{TH}}{V_S - V_{TH}}$$

e.g. $R_L = 1K$ $\quad V_S = 5V$ $\quad R_{ON} = 10\Omega$

$C_{GS} = 0.1\,pF$ $\quad V_{OL} = 1V$

$R_{TH} \approx 10\Omega, \quad V_{TH} \approx 0V$

$$t_f = -10 \cdot 0.1 \cdot 10^{-12} \ln \frac{1}{5}$$

$$= 1.6\,ps$$

$$RC = 1\,ps\,!$$

V-code: ???

Outline

Textbook: 10.4, 10.5, 10.6.2, 10.7

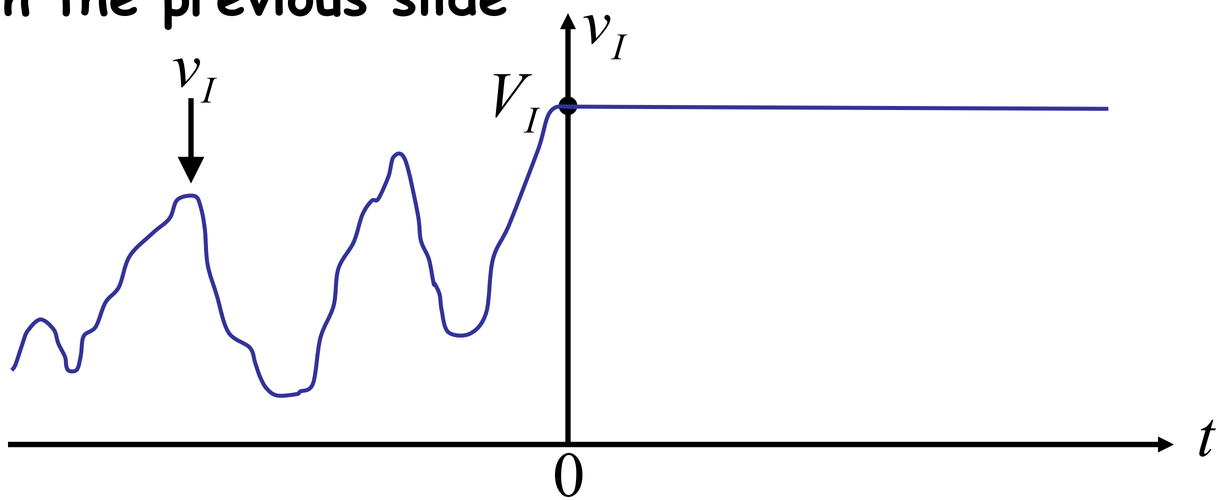
- Propagation Delay and the Digital Abstraction
- **State and State Variables**
- Digital Memory

V-code: ???

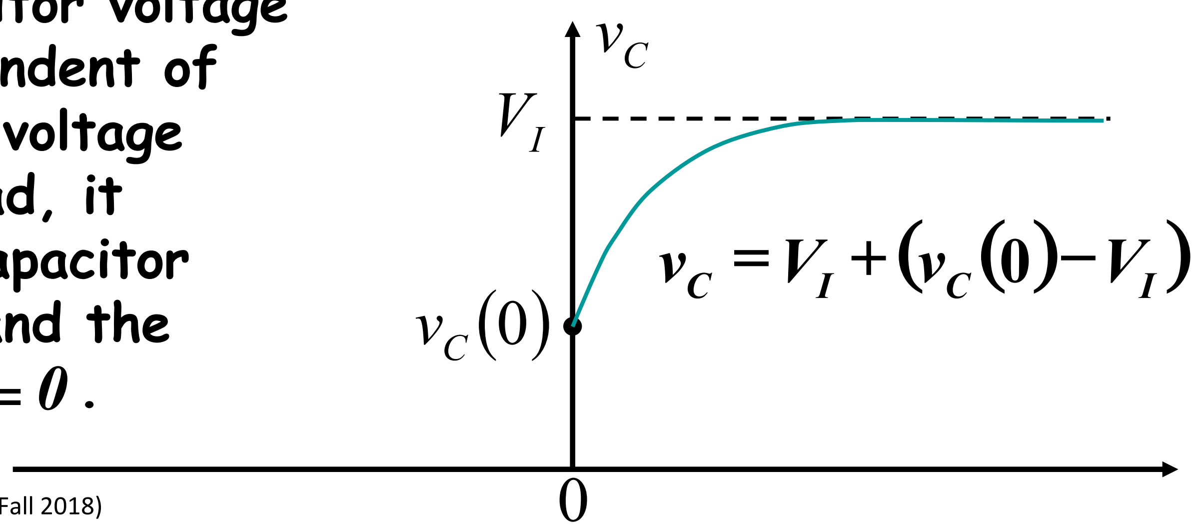
State and State Variables

This lecture will dwell on the memory property of capacitors.

For the RC circuit in the previous slide



Notice that the capacitor voltage for $t \geq 0$ is independent of the form of the input voltage before $t = 0$. Instead, it depends only on the capacitor voltage at $t \geq 0$, and the input voltage for $t = 0$.



$$v_C = V_I + \left(v_C(0) - V_I\right) e^{\frac{-t}{RC}}$$

V-code: ???

State and State Variables

State : summary of past inputs relevant to predicting the future

$$q = C\,V$$

for linear capacitors, capacitor voltage V is also state variable

state variable, actually

V-code: ???

State and State Variables

Back to our simple RC circuit ①

$$v_C = f\left(v_C(0), v_I(t)\right)$$

$$v_C = V_I + \left(v_C(0) - V_I\right) e^{\frac{-t}{RC}}$$

Summarizes the past input relevant to predicting future behavior

V-code: ???

State and State Variables

An alternative method to solve the problem:

Solve the transient problem by superposition!

$$\frac{dv_C}{dt} = -\frac{v_C}{RC} + \frac{v_I(t)}{RC}$$

$\frac{d}{dt}$ **(state variable)=** K_1**(State variable present value) +** K_2 **(input variable)**

Total Solution = zero-input response + zero-state response

V-code: ???

State and State Variables

An alternative method to solve the problem:

Solve the transient problem by superposition!

$$\frac{dv_C}{dt} = -\frac{v_C}{RC} + \frac{v_I(t)}{RC}$$

Zero-input response: $$\frac{dv_C}{dt} = -\frac{v_C}{RC}$$

Zero-state response: $$\frac{dv_C}{dt} = -\frac{v_C}{RC} + \frac{v_I(t)}{RC}$$

V-code: ???

State and State Variables

We are often interested in circuit response for

- zero state $v_C(0) = 0$
- zero input $v_I(t) = 0$

Correspondingly,

- zero state response or *ZSR*

$$v_C = V_I - V_I\, e^{\frac{-t}{RC}} \quad \text{———} \quad (2)$$

Total solution = (2) + (3)

- zero input response or *ZIR*

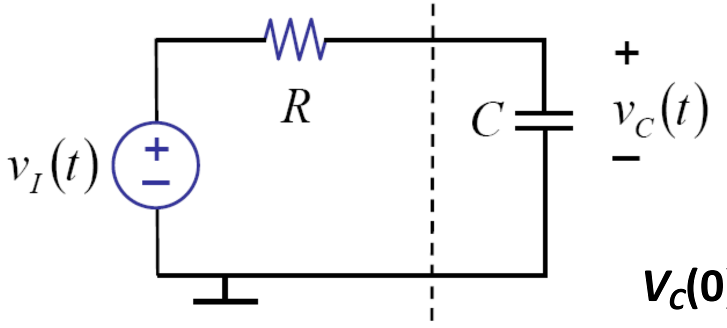
$$v_C = v_C(0) e^{\frac{-t}{RC}} \quad \text{———} \quad (3)$$

V-code: ???

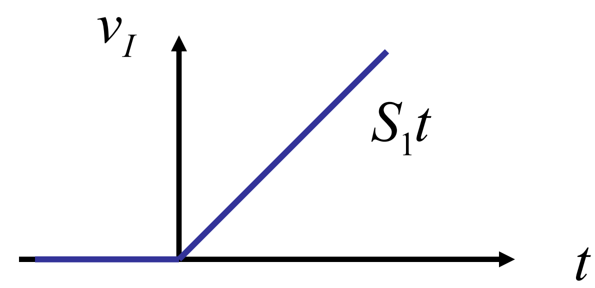
State and State Variables

- **Ramp input as an example: Total solution = ZSR (+ ZIR)**

Thévenin Equivalent:



$V_C(0) = 0$



Ramp input

V-code: ???

Outline

Textbook: 10.4, 10.5, 10.6.2, 10.7

- Propagation Delay and the Digital Abstraction
- State and State Variables
- **Digital Memory**

V-code: ???

Digital Memory

One application of STATE

DIGITAL MEMORY

Why memory?

Or, why is combinational logic insufficient?

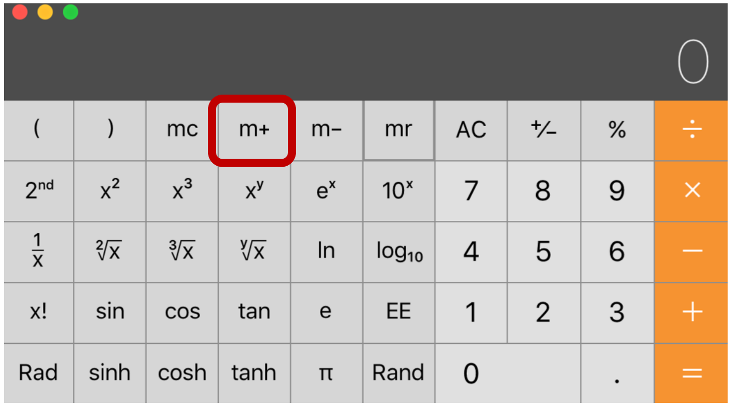
Examples

- Consider adding 6 numbers on your calculator

2 + 9 + 6 + 5 + 3 + 8

M+

("Add the displayed value to the memory")



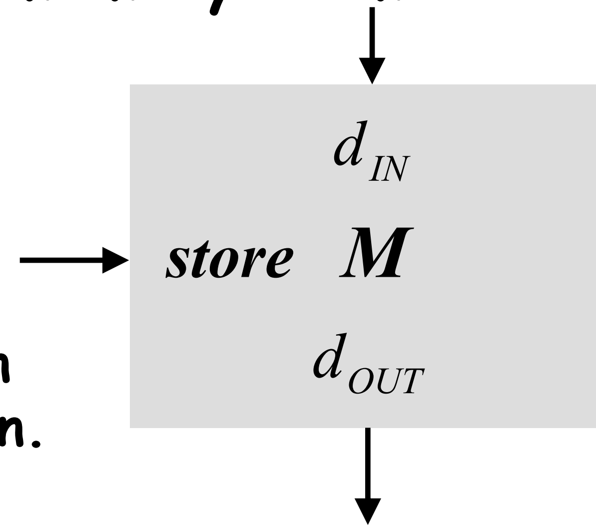
- "Remembering" transient inputs

V-code: ???

Digital Memory

Memory Abstraction

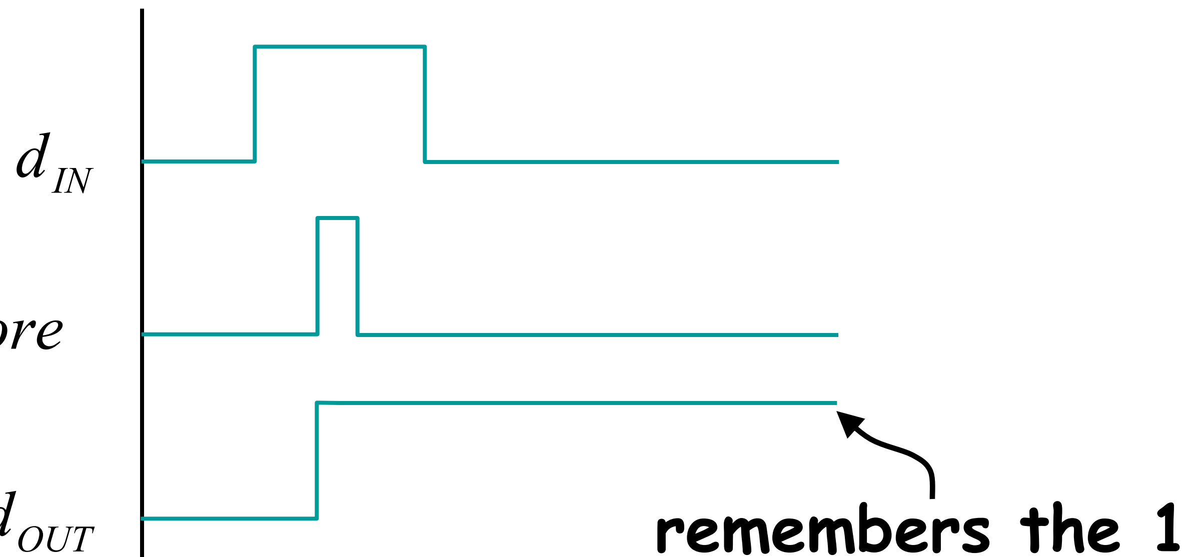
A 1-bit memory element



Remembers input when *store* goes high.

Like a camera that records input (d_{IN}) when the user presses the shutter release button.

The recorded value is visible at d_{OUT}.

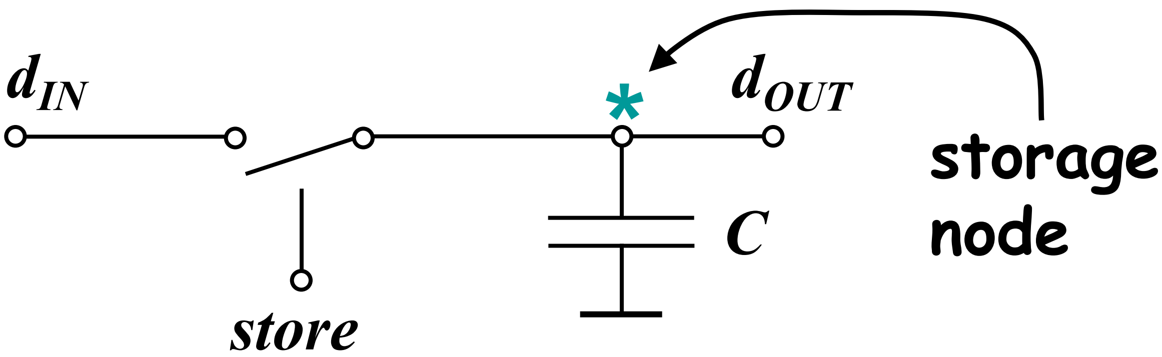


V-code: ???

Digital Memory

Building a memory element …

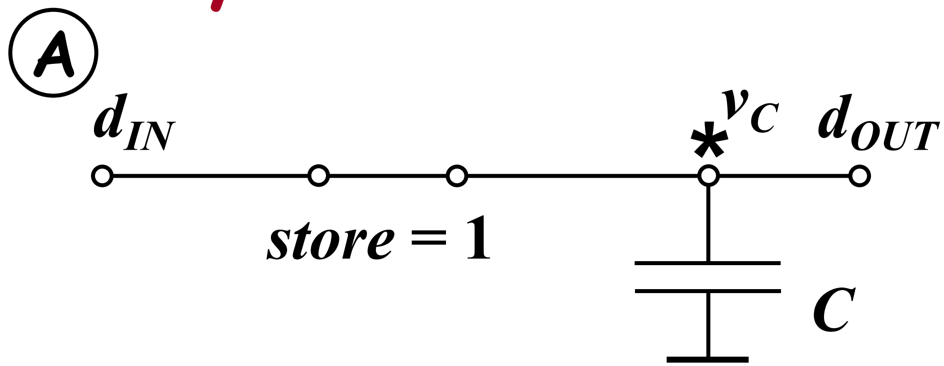
Ⓐ First attempt

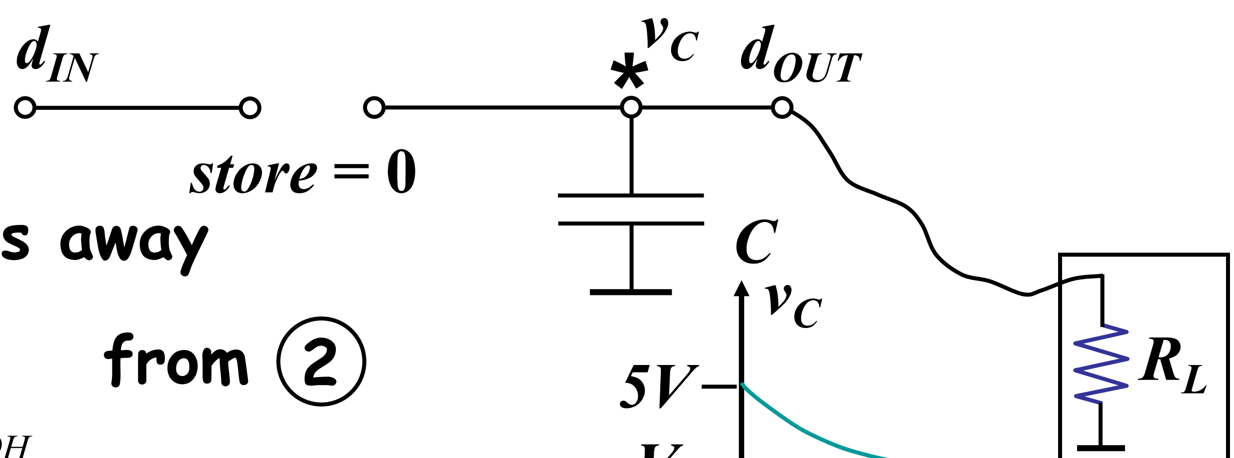


V-code: ???

Digital Memory

Building a memory element …



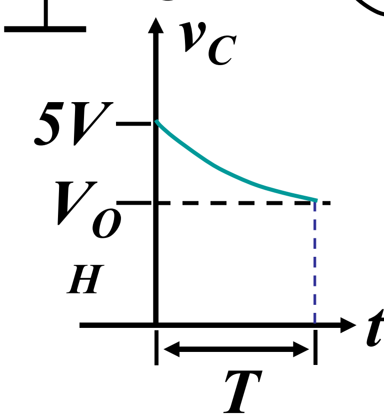


Stored value leaks away

$$v_C = 5 \cdot e^{\frac{-t}{R_L C}}$$

from 2

$$T = -R_L C \ln \frac{V_{OH}}{5}$$



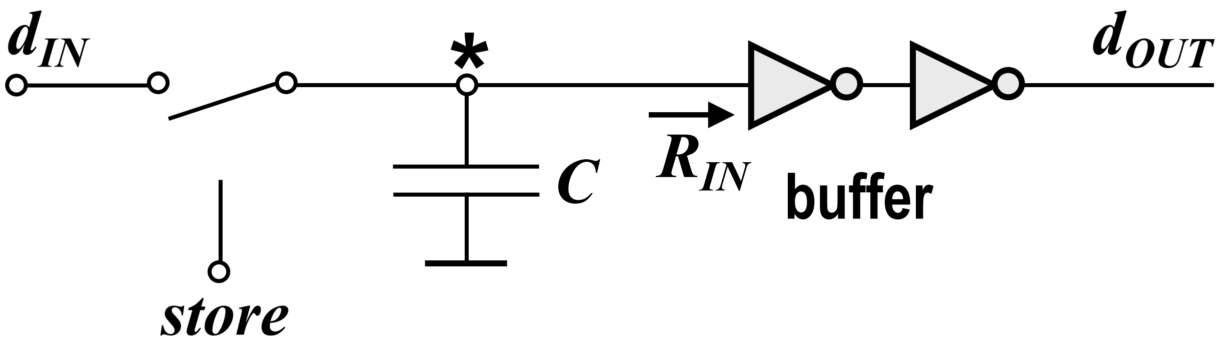
$store$ **pulse width >> $R_{ON}\ C$**

V-code: ???

Digital Memory

Building a memory element …

Ⓑ Second attempt → buffer



Input resistance R_{IN}

$$T = -R_{IN} C \ln \frac{V_{OH}}{5}$$

$$R_{IN} >> R_L$$

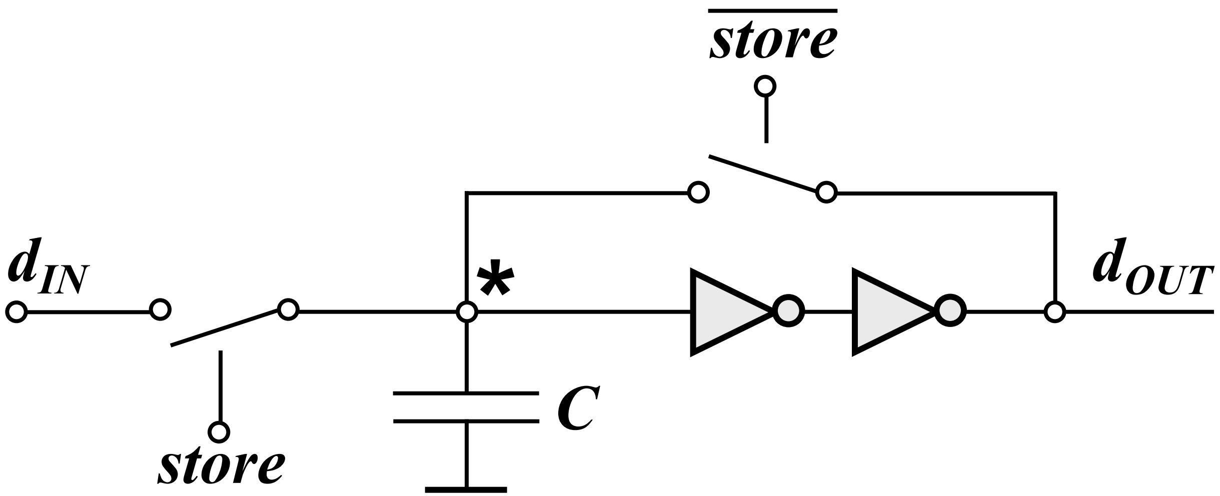
Better, but still not perfect.

V-code: ???

Digital Memory

Building a memory element …

Ⓒ Third attempt → buffer + refresh



Does this work?

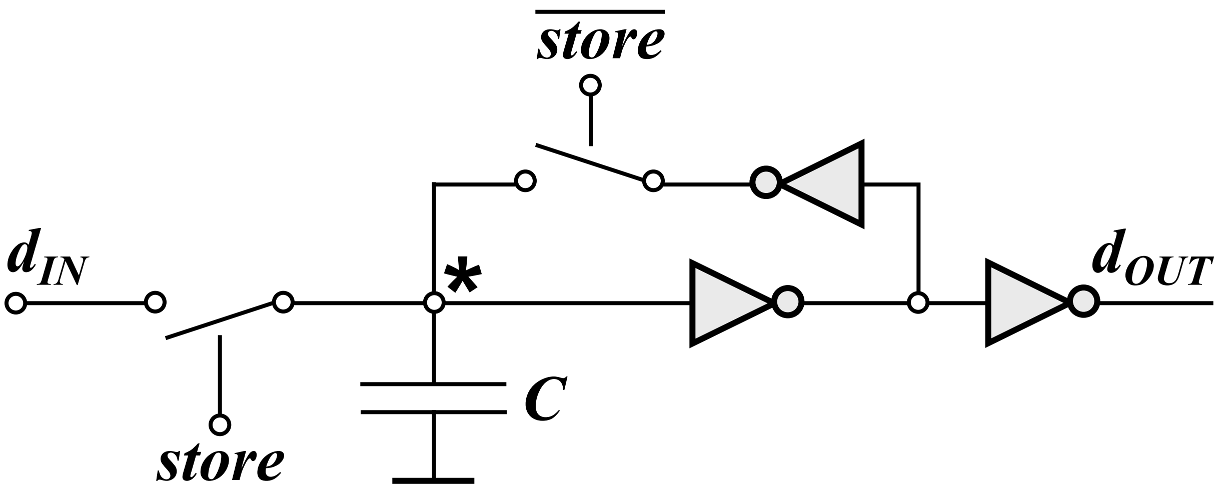
No. External value can influence storage node.

V-code: ???

Digital Memory

Building a memory element …

(D) Fourth attempt → buffer + decoupled refresh



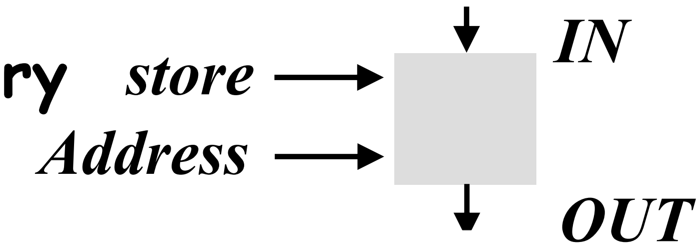
Works!

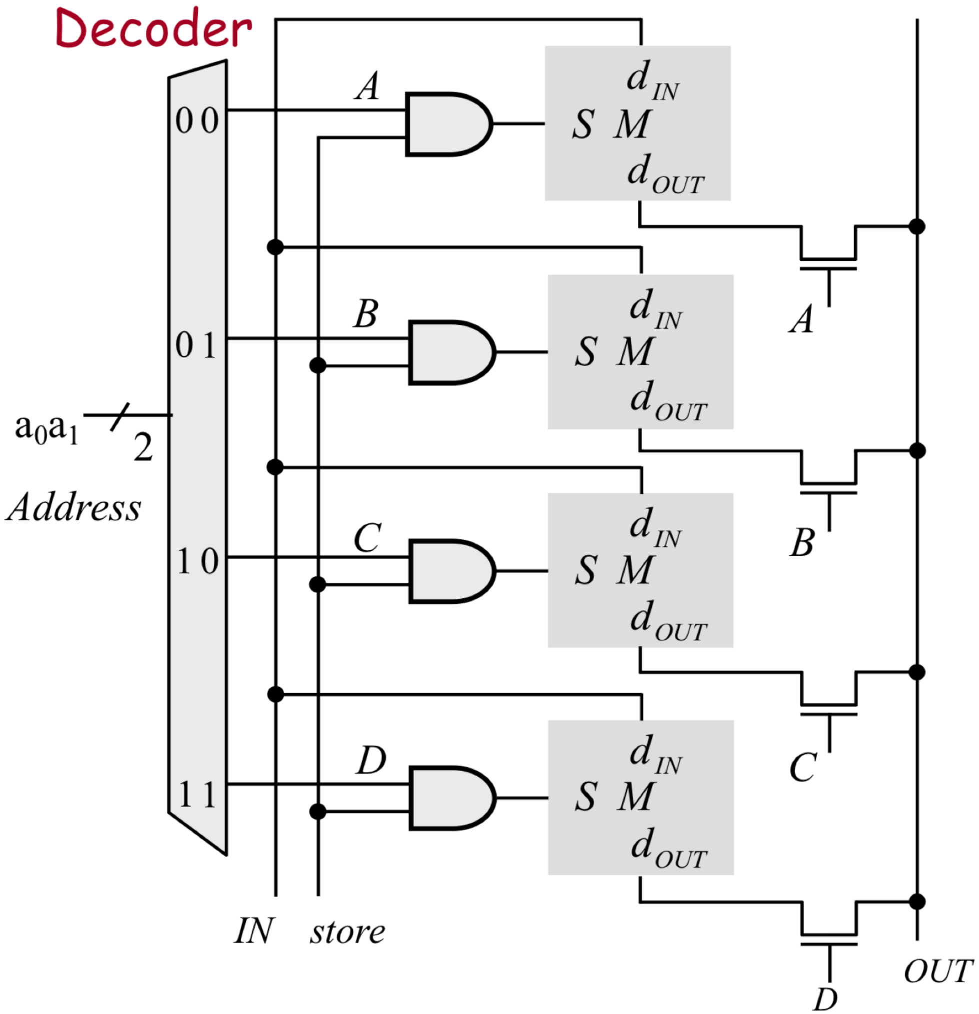
V-code: ???

Digital Memory

A Memory Array

4-bit memory





V-code: ???

Digital Memory

Truth table for decoder

a_0	a_1	A	B	C	D
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1