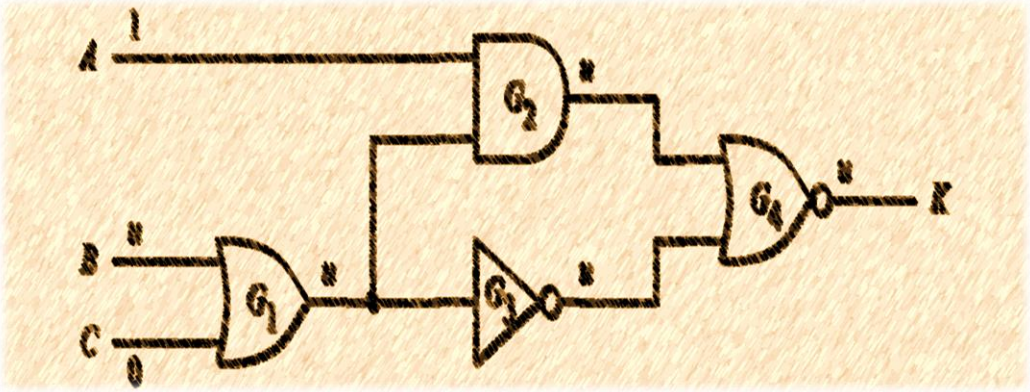
# Logic Simulation

- Introduction
- **Simulation Models**
  - **Logic States**
  - **Logic Gate Evaluation**
  - **Delay Models**
- Logic Simulation Techniques
- Issues of Logic Simulations
- Conclusions



# Logic States

- ***Binary logic*** **(0,1)**
- ***Three-valued logic*** **(0,1, *u*) aka. *Ternary logic* [Eichelberger 65]**
  - ***u*** **(unknown) for logic value cannot be determined**
    - **aka. "x"**
- ***Four-valued logic*** **(0,1, *u,z*)**
  - ***z*** **= floating node without conducting path to VDD or GND**
    - ***aka. high impedance***
  - **Note: do not confuse "*u*" and "*z*"**

| AND | 0 | 1 | u | z |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | u | u |
| u | 0 | u | u | u |
| z | 0 | u | u | u |

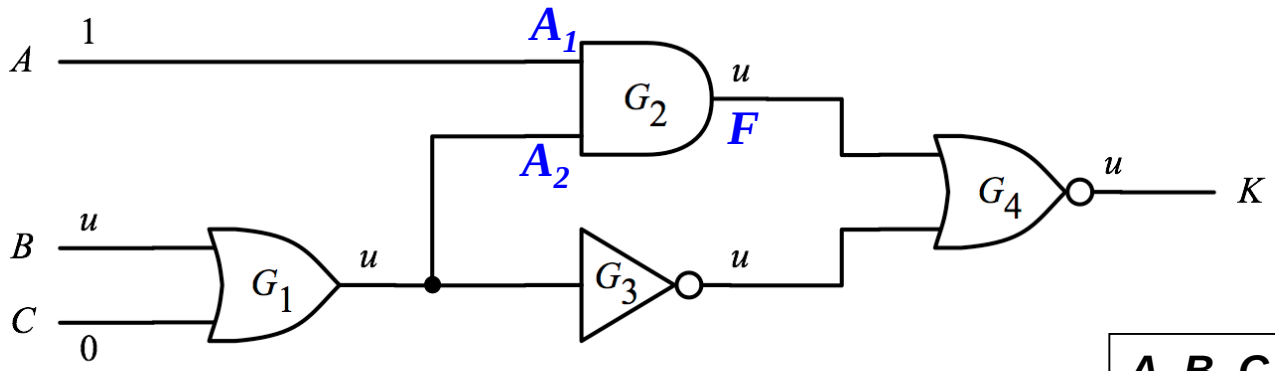
| OR | 0 | 1 | u | z |
|---|---|---|---|---|
| 0 | 0 | 1 | u | u |
| 1 | 1 | 1 | 1 | 1 |
| u | u | 1 | u | u |
| z | u | 1 | u | u |

| NOT | 0 | 1 | u | z |
|---|---|---|---|---|
| | 1 | 0 | u | u |

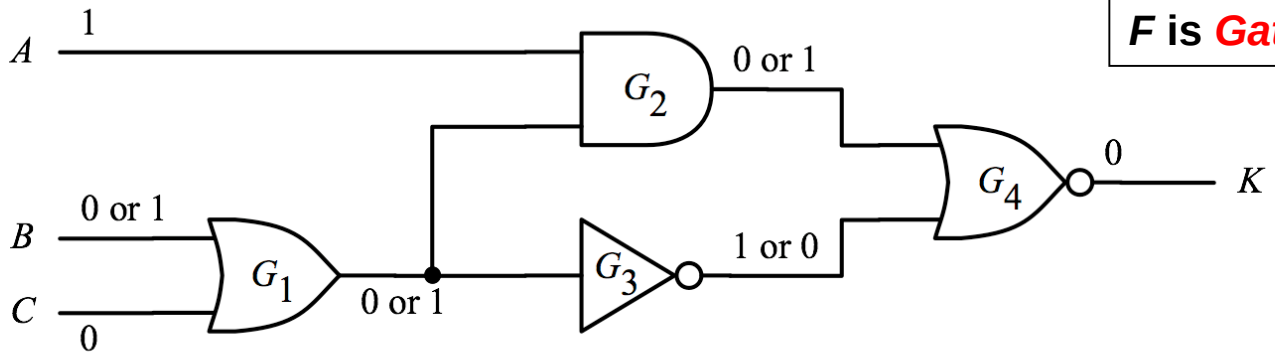
**More States, More Accuracy, More CPU Time**

# Ternary Logic Simulation

- **Problem: output *K* should be zero**




(a) Ternary logic simulation: $K = u$

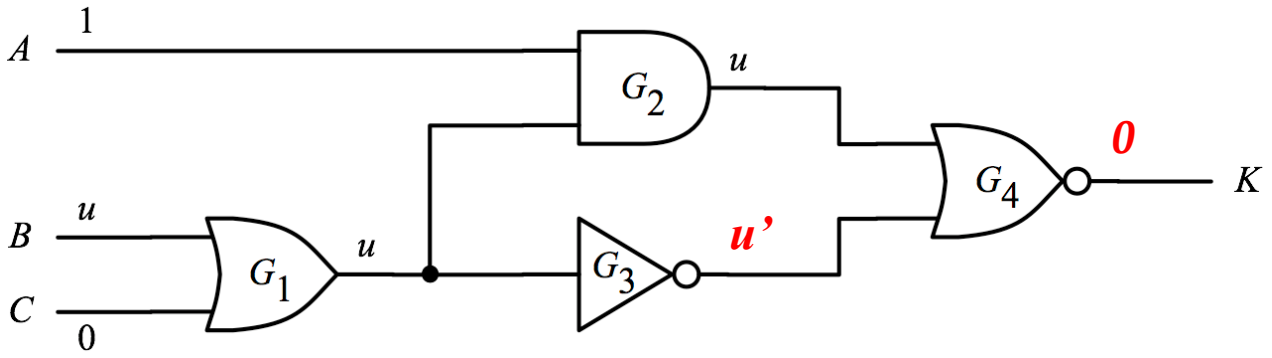



(b) Enumerate all possible cases ($B = 0$ and 1): $K = 0$

***A, B, C* are *Primary Inputs (PI)***
***$A_1$ $A_2$* are *Gate Inputs* for *$G_2$***
***K* is *Primary Output (PO)***
***F* is *Gate Output* for *$G_2$***

# Possible Solution

- **Introduce *u*' to represent inverse of *u***




- **What is problem with this solution?**
  - **For every different source of *u*, need one unique *u*'**
    - **Too many *u*'s !**
    - **Need *symbolic simulation* to distinguish all *u*'s**

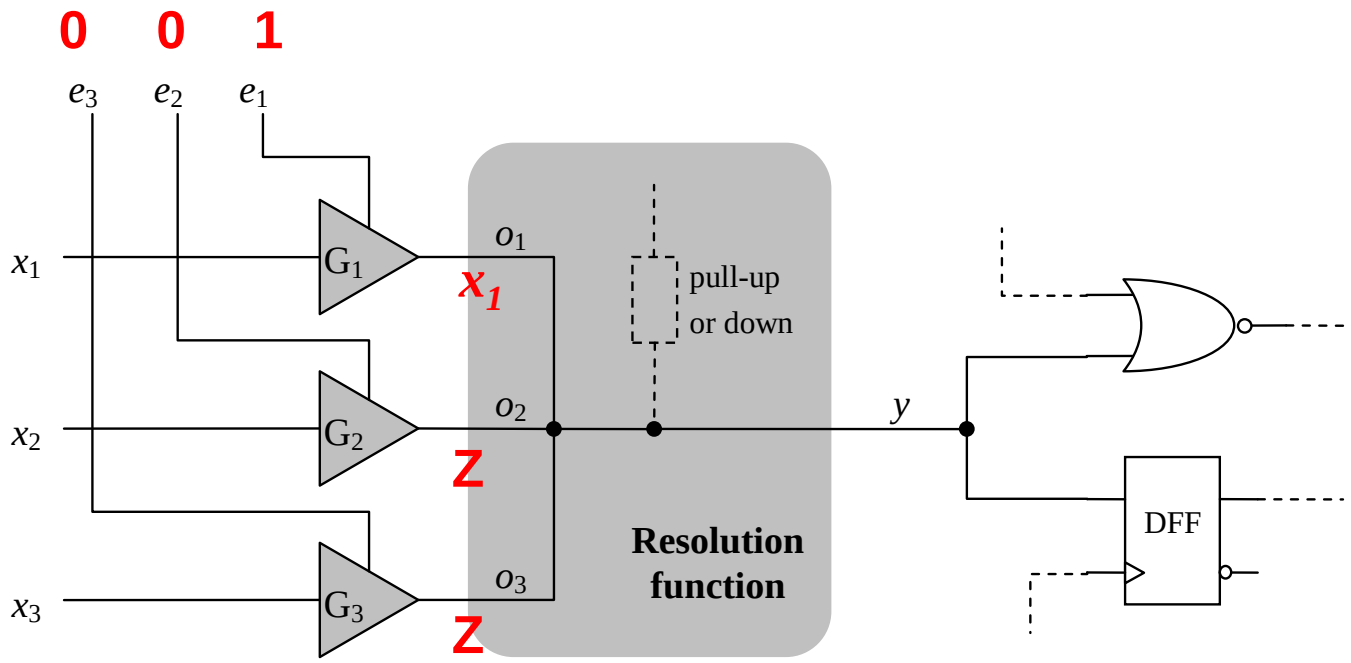
**Moral: Simulation Results not 100% accurate. Must Check!**

# *High Impedance* State (Z)

- **Tri-state bus driver**

$$o_i = \begin{cases} x_i & if\ e_i = 1 \\ Z & if\ e_i = 0 \end{cases}$$

- **Only one driver enabled at a time**
  - ***$e_1$,$e_2$, $e_3$ are one-hot signals***

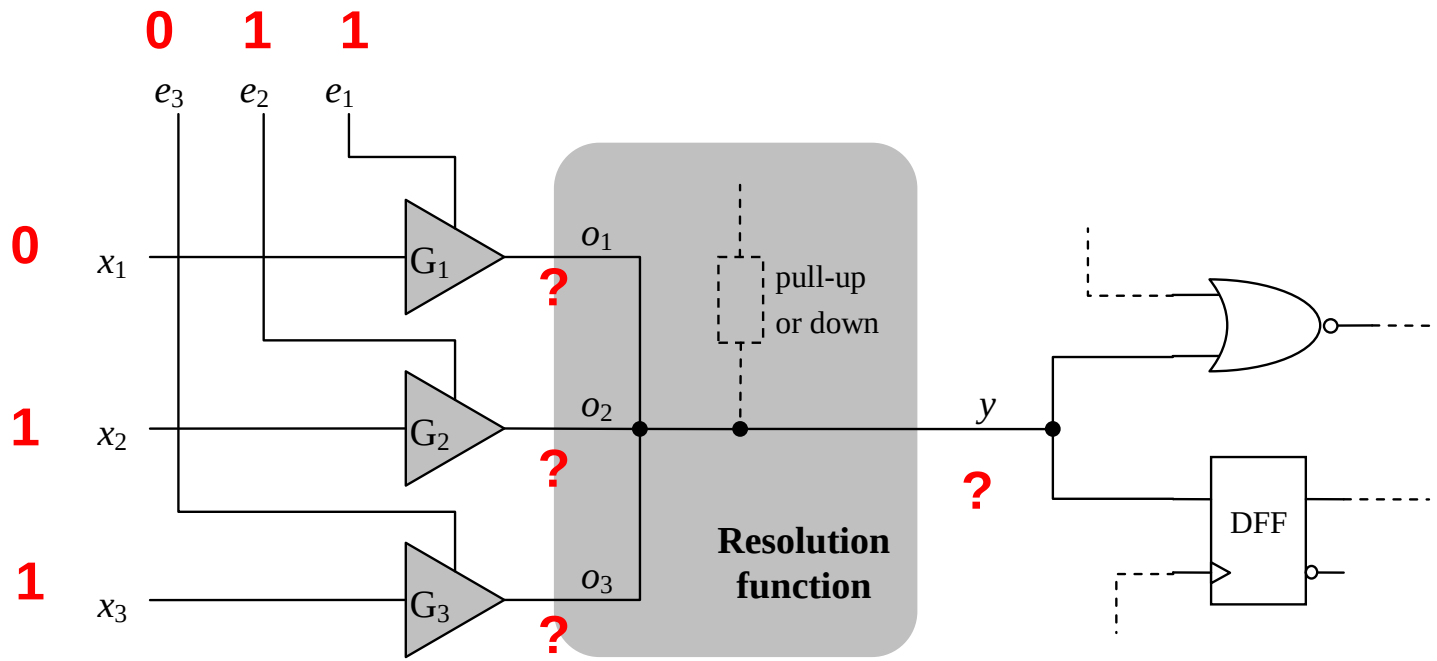



(WWW Fig 3.6)

# Quiz

**Q: If $e_1$,$e_2$, $e_3$ are NOT one-hot signals what is y?**

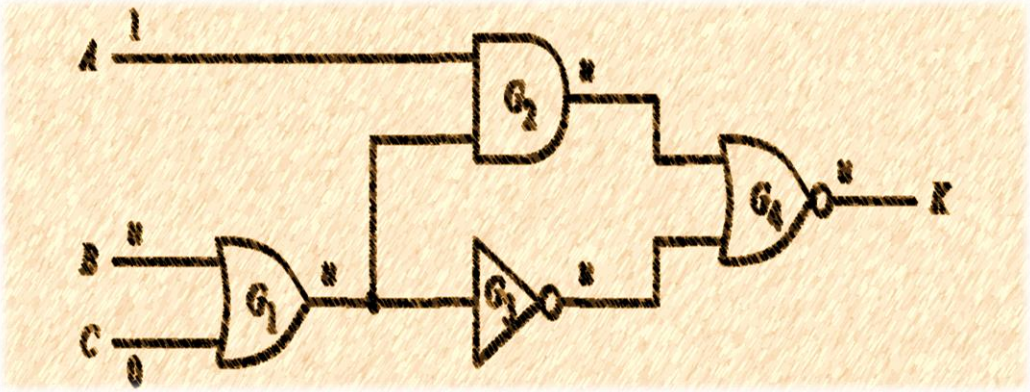
**A:**




(WWW Fig 3.6)

# Logic Simulation

- Introduction
- Simulation Models
  - Logic States
  - Logic Gate Evaluation
    - Truth table
    - Input scanning
  - Delay Models
- Logic Simulation Techniques
- Issues of Logic Simulations
- Conclusions



# Gate Evaluation Using Truth Table

- **Determine output value based on a pre-stored truth table**
- **Example: 4-valued logic truth table**

| AND | 0 | 1 | u | z |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | u | u |
| u | 0 | u | u | u |
| z | 0 | u | u | u |

| OR | 0 | 1 | u | z |
|---|---|---|---|---|
| 0 | 0 | 1 | u | u |
| 1 | 1 | 1 | 1 | 1 |
| u | u | 1 | u | u |
| z | u | 1 | u | u |

| NOT | 0 | 1 | u | z |
|---|---|---|---|---|
| | 1 | 0 | u | u |

**Quiz:**
**What is size of truth table for n-input gate using 4 valued-logic?**
**A:**

- **Practical implementation**
  - **Break n-input gate into multiple 2-input gates**
  - **Use for-loop to evaluate gate one by one**

# Input Scanning

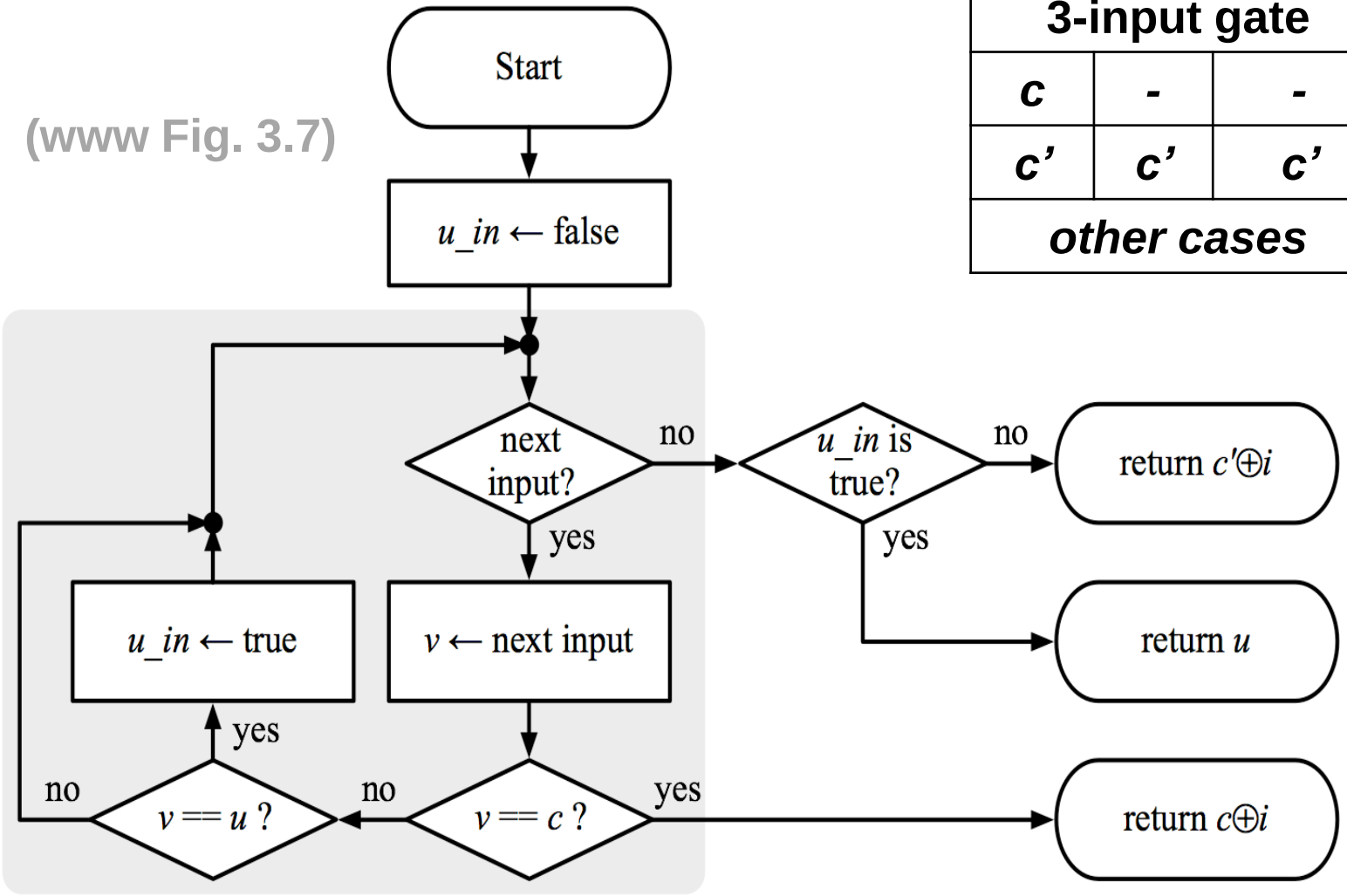
- Idea: check gate inputs one by one
  - Determine gate output by number of *c, i, u*
- Gates can be characterized by *
  - *Controlling value, c*
    - *c* = input value that decides gate output
    - regardless of other gate inputs
  - Inversion, *i*
    - *i* =1, gate output is inverted w.r.t. gate input
    - *i* =0, otherwise

| | *c* | *i* |
|---|---|---|
| AND | 0 | 0 |
| OR | 1 | 0 |
| NAND | 0 | 1 |
| NOR | 1 | 1 |

*Assumes only elementary gates : AND, OR, NAND, and NOR

# Input Scanning Algorithm

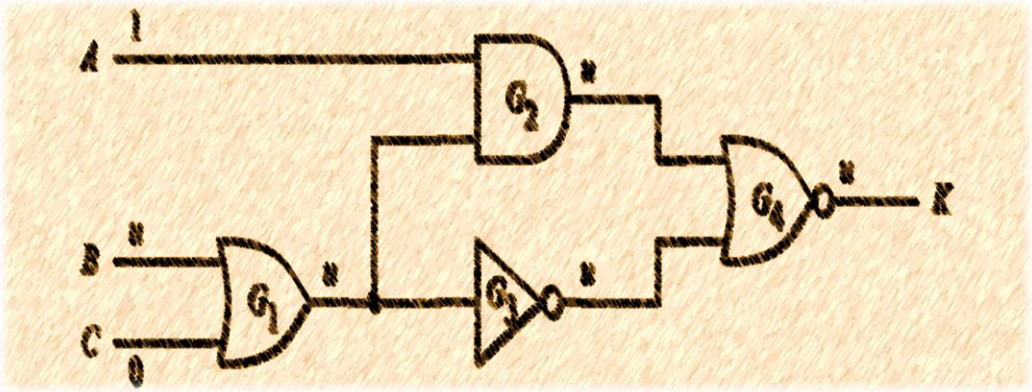
(www Fig. 3.7)




| 3-input gate | | | O/P |
|---|---|---|---|
| ***c*** | **-** | **-** | ***c⊕i*** |
| ***c'*** | ***c'*** | ***c'*** | ***c'⊕i*** |
| ***other cases*** | | | ***u*** |

# Logic Simulation

- Introduction
- Simulation Models
  - Logic States
  - Logic Gate Evaluation
  - Delay Models
    - Gate delay model
    - Wire delay (not in lecture)
- Logic Simulation Techniques
- Issues of Logic Simulations
- Conclusions



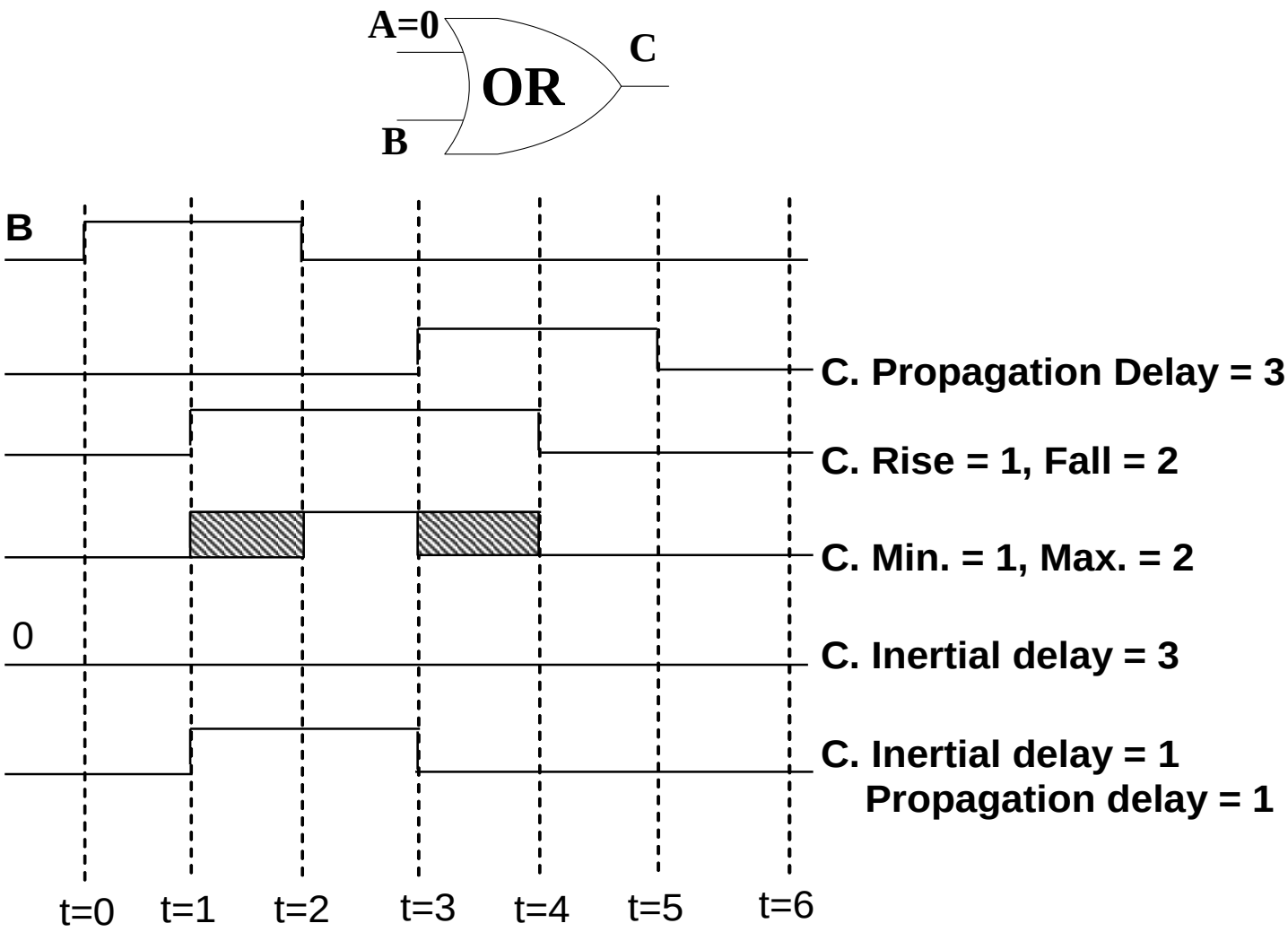
# Gate Delay Models

- ***Propagation delay*** **(aka. *Transport delay*)**
  - **DEF: time from gate input change to output change**
    - ***Zero delay*****: prop. delay of all gates is zero**
    - ***Unit delay:*** **prop. delay of all gates is 1 time unit**
    - ***Multiple delay:*** **prop. delay is multiple of some time unit**
    - ***Rise delay, fall delay***

- ***Ambiguous delay model***
  - ***Max delay, Min delay, Typical (nominal) delay***
- ***Inertial delay model***
  - **Minimum amount of time during which a signal must persist at gate input in order to change gate output**

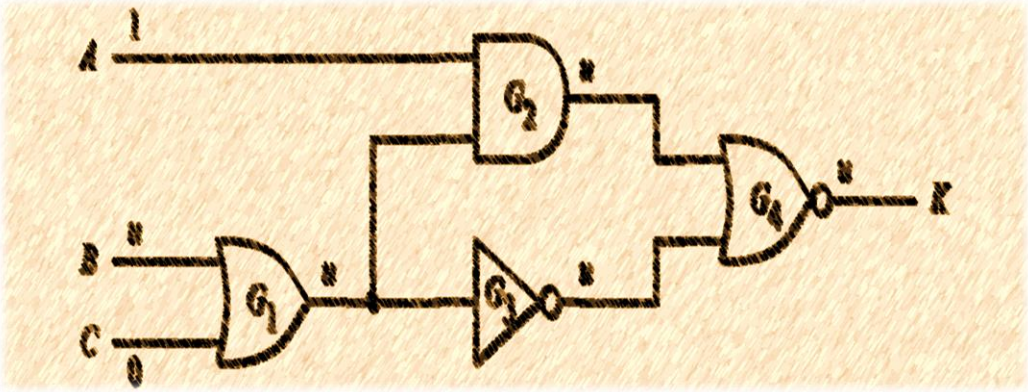
**Different Delay Model, Different Usage**

# Example



# Summary

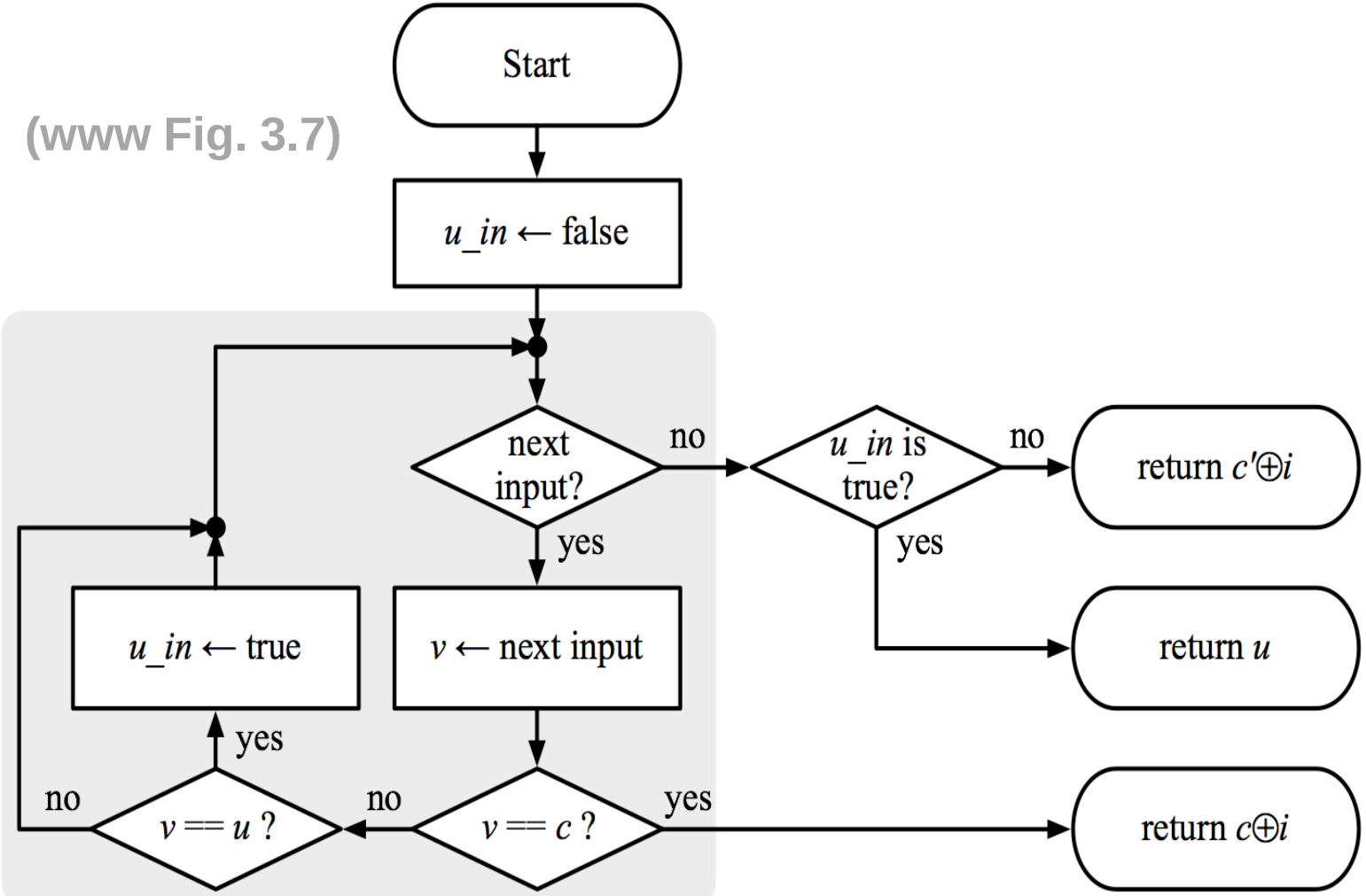
- **Simulation Models**
  - **Logic States**
    - **0,1,u,z**
  - **Logic Gate Evaluation**
    - **Truth table, Input scanning**
  - **Delay Models**
    - **propagation, ambiguous, inertial delay**



# FFT 1

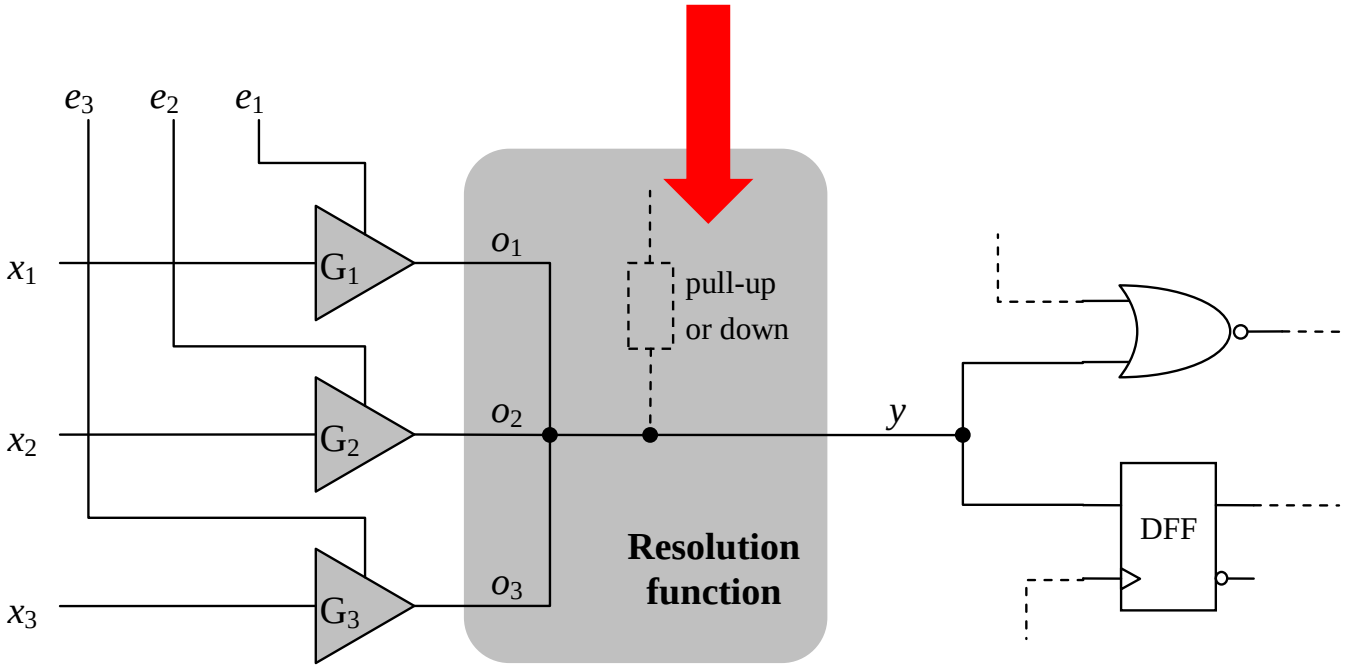
- **Q: From truth table to input scanning, what do we gain/loose?**
  - **HINT: CPU time vs memory**

(www Fig. 3.7)



# FFT 2

- **Q: What is the use of pull-up or pull-down?**
  - **HINT: what happens when $e_1=e_2=e_3=0$**

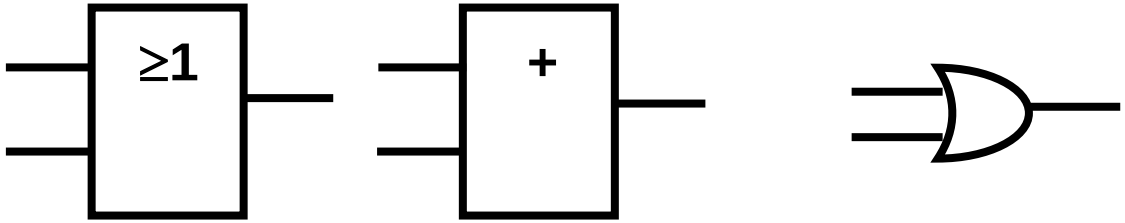



(WWW Fig 3.6)

# Appendix: Logic Symbols

- IEEE logic symbols: rectangular shape v.s. distinctive shape
- AND


- Or




- inverter