Delay Relevant Reliability of CMOS Logic Circuits in Near/Sub Threshold Region

Chen Jiaoyan, Christian Spagnol, Satish Kumar
Supervisor: Dr. Emanuel Popovici
University College Cork

Possible Reliability Model

- **What is Reliability?**

  Permanent errors: failure to switch from 0-1 / 1-0.

  Performance, i.e. speed/delay: The higher the speed is, the lower the reliability might be.

- **How to link the delay to Reliability?**

$$R(d, f) = 1 - P\left(d > \frac{1}{f}\right)$$

d= delay; f= reference frequncy; P=Probability

How to Get the Reliability Model

- VDD @ Near/Sub threshold region (<0.4V), Models: Inverter, AND
- 45nm process, hspice (monte carlo) simulation: 10,000 samples
- Variation to concern: VDD, process variation, temperature

  VDD: @0.3V with 50mV variation (gauss distribution, sigma =1)

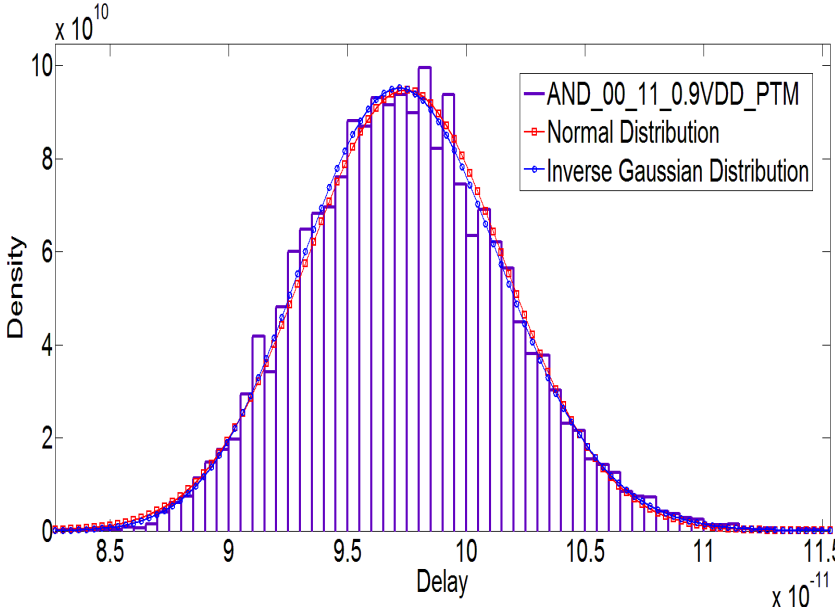
  Process Variation: Vthn=0.32V, Vthp= -0.3V with 50mV variation (gauss distribution, sigma =3), plus 10% variation for thickness of the oxide.

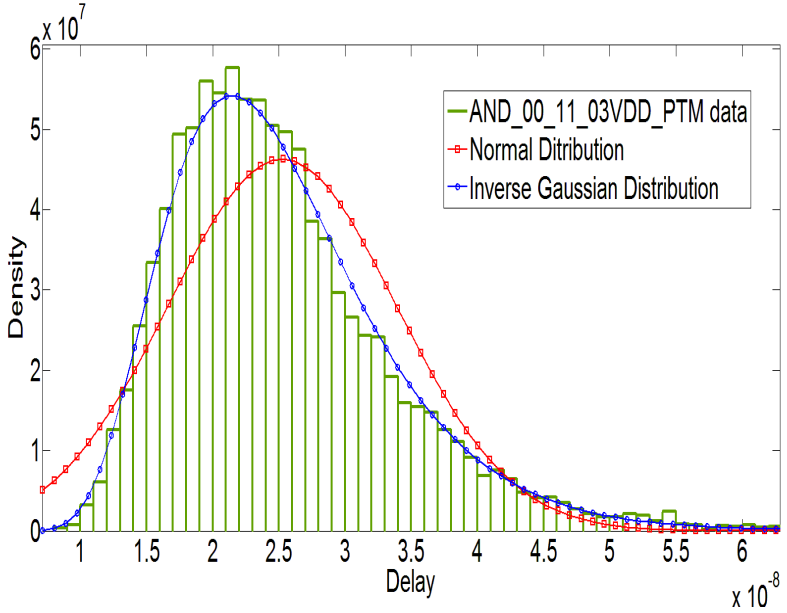
  Assumption: Input follows the change of VDD.

- Temperature has little impact on delay compared with other two.

Related Work

- In [1], with similar aspects of variation, it claimed that Normal (Gaussian) approximation matched the Probability Density Functions (PDFs) of the propagation delays.
- However, No strong theoretical proof for this approximation. So is it true?
- PTM (predictive model) AND gate input switch from 00-11 @0.9VDD, 0.3VDD with process variation.

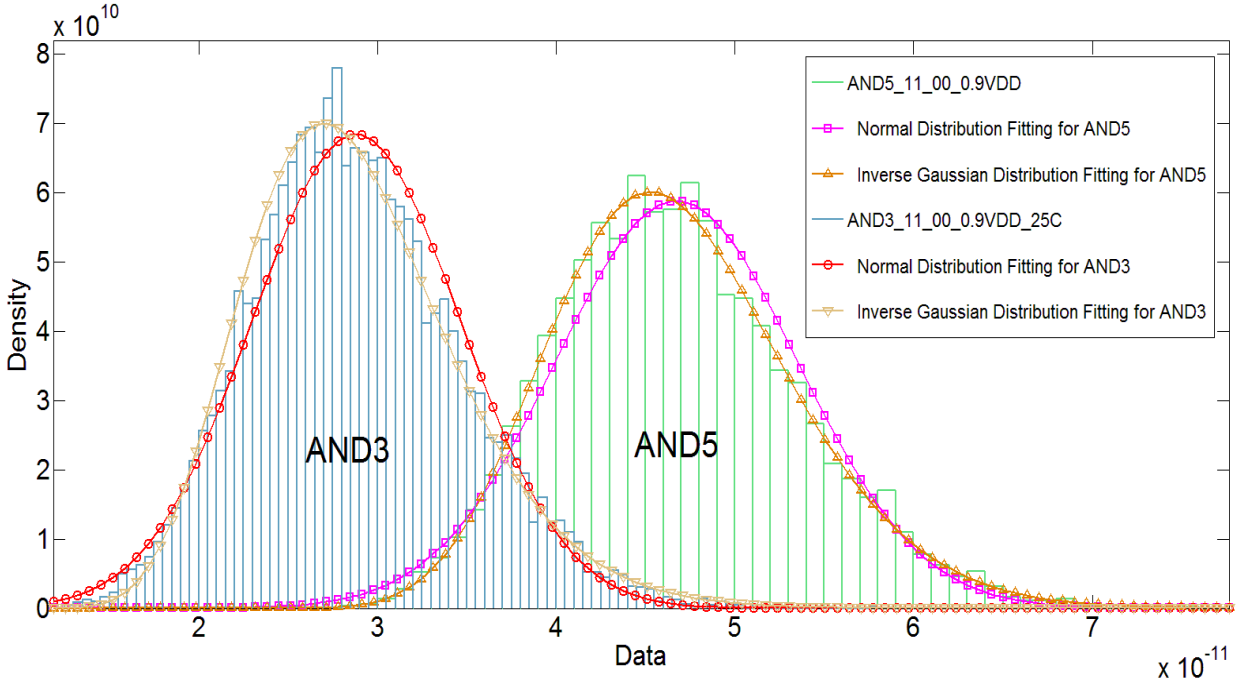




[1]: 'Fast Error Aware Model for Arithmetic and Logic', Zaynoun and Khairy, ICCD12

Further Validation

- Five cascaded AND gates suffer both process variation and power supply variation @ 0.9V.
- Only the third and fifth AND gates are shown for the clairty.
- Again, Inverse Guassian fitting beats the normal distribution.



Normal or Inver Gaussian & Why?

- Normal (Gaussian) distribution: a number falling between any two real numbers. Consider the negative delay, possible? Plus, it is symmetric.

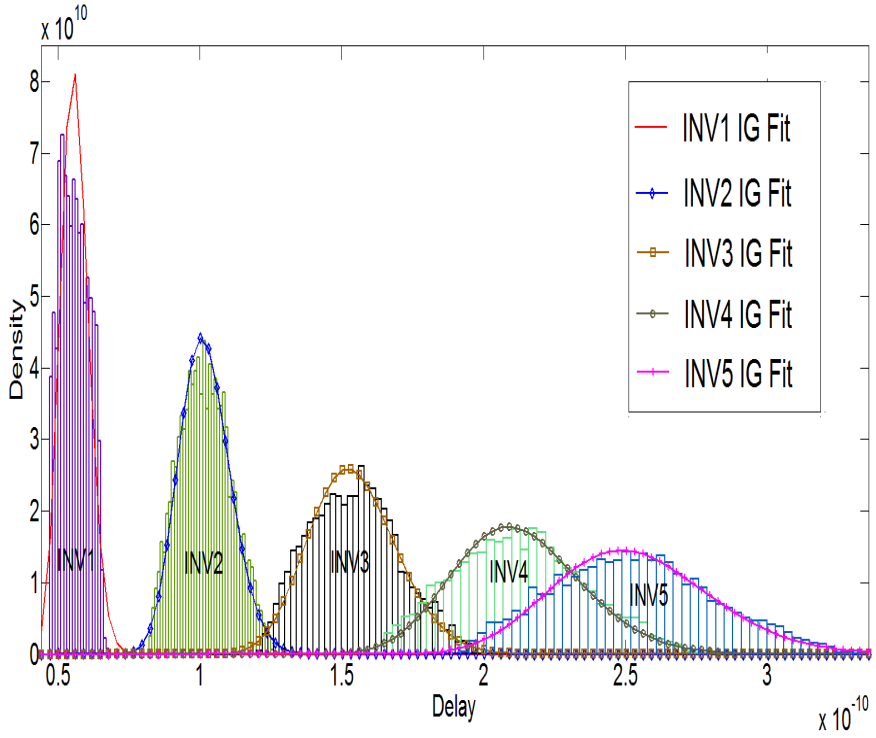
Versus

- Inverse Gaussian distribution: a Brownian Motion with positive drift takes to reach a fixed positive level (the movement of electron).
- Brownian Motion: an idealised approximation to actual random physical processes with finite time scale.

Any doubt? Yes, is it because guassian distribution is used in those variations? Let's try something else.

Inverse Gaussian valid for Uniform distribution?

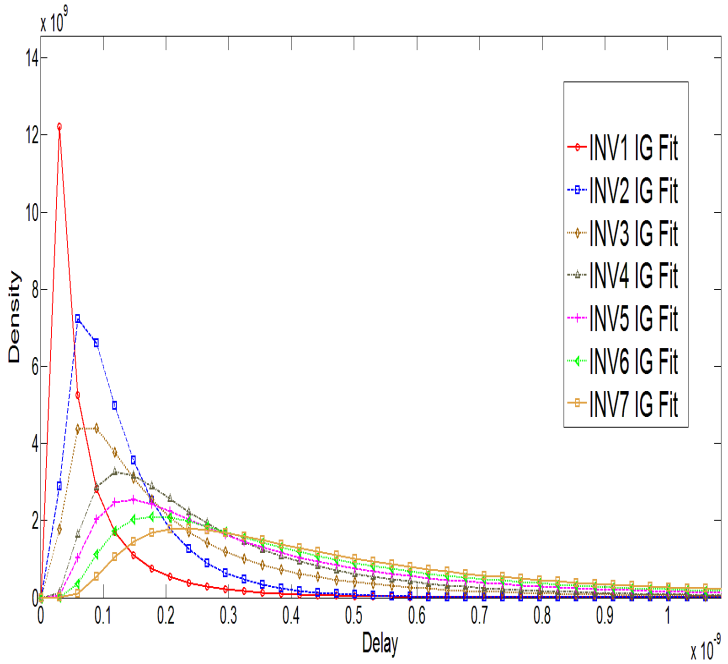
- We built a chain of five inverters where uniform distribution was used for the Vth variation, VDD=0.3V.



After 1st stage, the delay fitted again to the inverse Gaussian distribution.

How to take advantage of Inverse Gaussian Dis.?

- A chain of seven Inverters was simulated with both VDD and process variations. So how will the PDFs propagate?



There are two parameters for Inverse Gaussian Distribution: μ is the mean parameter; λ is the shape parameter.

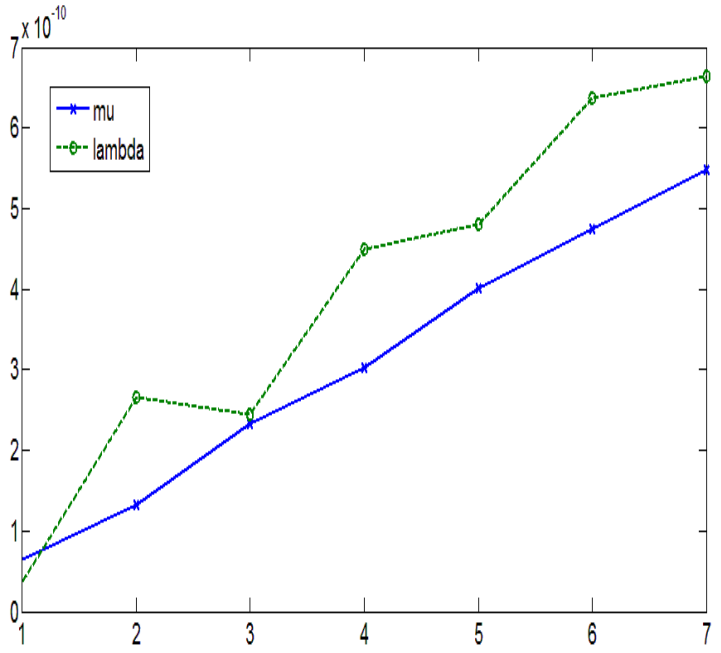
$$f(x;\mu,\lambda) = \left[\frac{\lambda}{2\pi x^3}\right]^{1/2} \exp\frac{-\lambda(x-\mu)^2}{2\mu^2 x}$$

Scaling:

$$X \sim IG(\mu,\lambda) \quad \Rightarrow \quad tX \sim IG(t\mu, t\lambda).$$

How good is scaling?

	Input Switch	μ (e^{-10})	Δμ(e^{-10})	λ (e^{-10})	Δλ(e^{-10})
INV1	1-0(charging)	0.65	/	0.37	/
INV2	0-1(discharging)	1.3	0.65	2.6	2.23
INV3	1-0(charging)	2.3	1	2.4	-0.2
INV4	0-1(discharging)	3.0	0.7	4.5	2.1
INV5	1-0(charging)	4.0	1	4.8	0.3
INV6	0-1(discharging)	4.7	0.7	6.4	1.6
INV7	1-0(charging)	5.5	0.8	6.6	0.2
Average (charging)		/	0.9	/	0.1
Average (discharging)		/	0.7	/	2.0

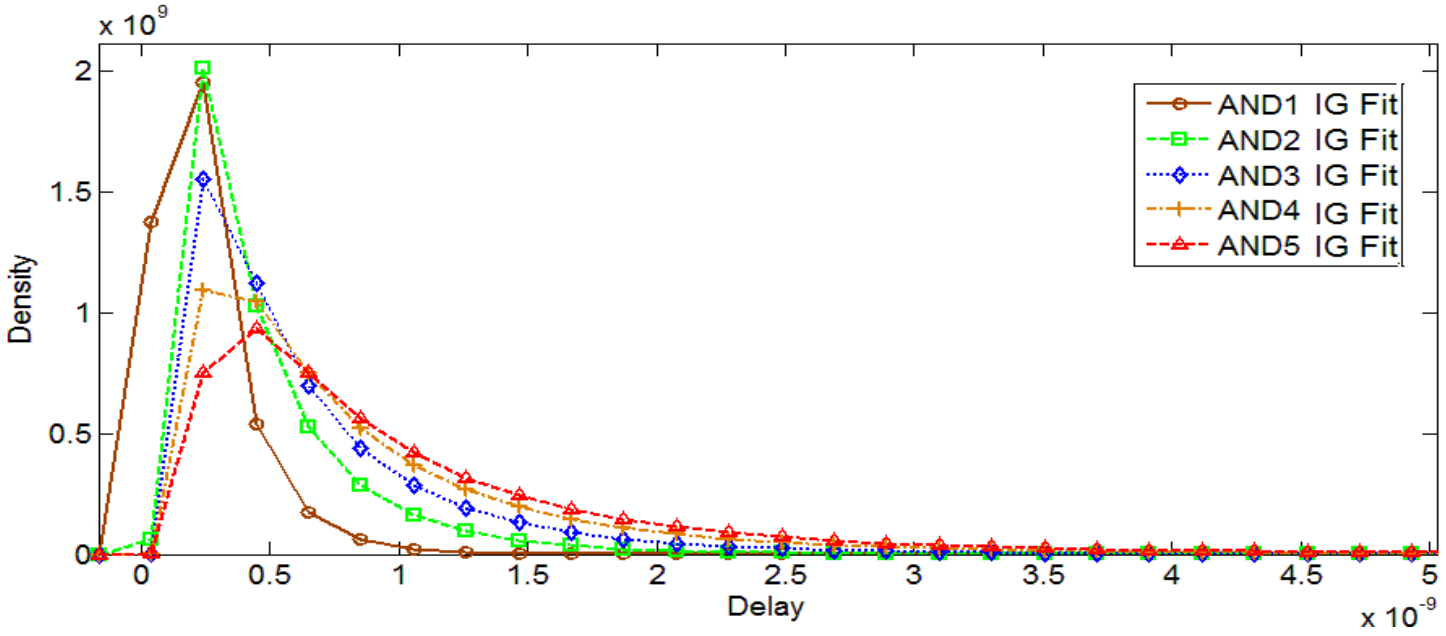


- The increment of μ is quite steady (blue line), that of λ increase like stair-wise (green line) which is due to the characteristic of NMOS and PMOS (discharging and charging).

Is it the same for AND gate?

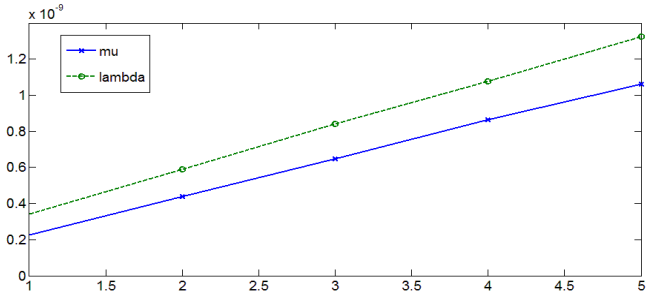
- A chain of five AND gates was simulated with both VDD and process variations.
- Four different events (input switching) were considered:
  10-11, 11-01, 00-11, 11-00 (where we regard 10 and 01 are the same)

10-11

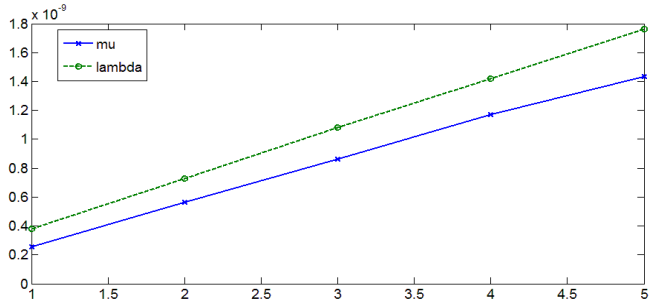


Is it same for AND gate?

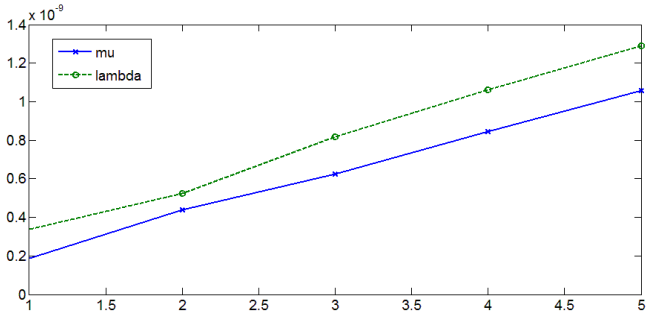
10-11 (output 0-1)



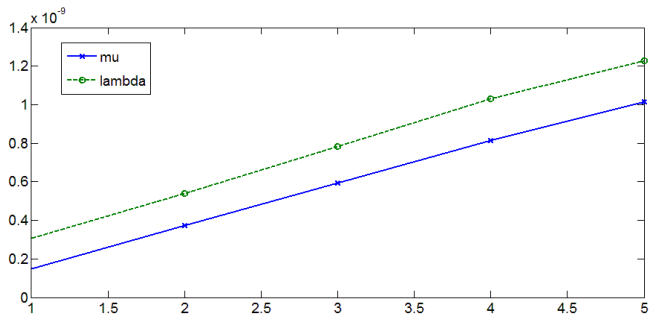
00-11 (output 0-1)



11-01 (output 1-0)



11-00 (output 1-0)



- μ (blue line) and λ (green line) increase almost linearly while the increment is different for different cases which is due to charging and discharging (PMOS and NMOS).

10-11	μ (e^{-10})	Δμ(e^{-10})	λ (e^{-10})	Δλ(e^{-10})
AND1	2.3	/	3.4	/
AND2	4.4	2.1	5.9	2.5
AND3	6.5	2.1	8.4	2.5
AND4	8.7	2.2	10.8	2.4
AND5	10.6	1.9	13.3	2.5
Average	/	2.1	/	2.5

00-11	μ (e^{-10})	Δμ(e^{-10})	λ (e^{-10})	Δλ(e^{-10})
AND1	2.6	/	3.8	/
AND2	5.6	3.0	7.3	3.5
AND3	8.6	3.0	10.8	3.5
AND4	11.7	3.1	14.2	3.4
AND5	14.3	2.6	17.6	3.4
Average	/	2.9	/	3.5

11-01	μ (e^{-10})	Δμ(e^{-10})	λ (e^{-10})	Δλ(e^{-10})
AND1	1.9	/	3.4	/
AND2	4.4	2.5	5.3	1.9
AND3	6.3	1.9	8.2	2.9
AND4	8.4	2.2	10.6	2.4
AND5	10.6	2.2	12.9	2.3
Average	/	2.2	/	2.4

11-00	μ (e^{-10})	Δμ(e^{-10})	λ (e^{-10})	Δλ(e^{-10})
AND1	1.5	/	3.1	/
AND2	3.7	2.2	5.4	2.3
AND3	5.9	2.2	7.8	2.4
AND4	8.2	2.3	10.3	2.5
AND5	10.2	2.0	12.3	2.0
Average	/	2.2	/	2.3

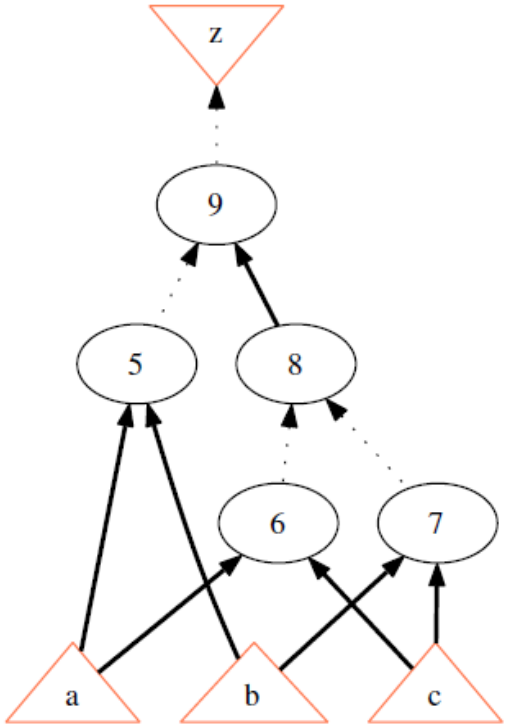
What can we learn?

- We may propagate the PDFs by adding μ and λ respectively.
- So far, we considered the circuits with one type of gate only.
  So how about using INV and AND to build some circuits?
  i.e. Majority Voter, XOR gate

- Next step, running monte carlo simualtion for two above mentioned gate and then to compare with the statistical estimation results (based on inverse Gaussian approximation) in the form of Cumulative distribution function (CDF).

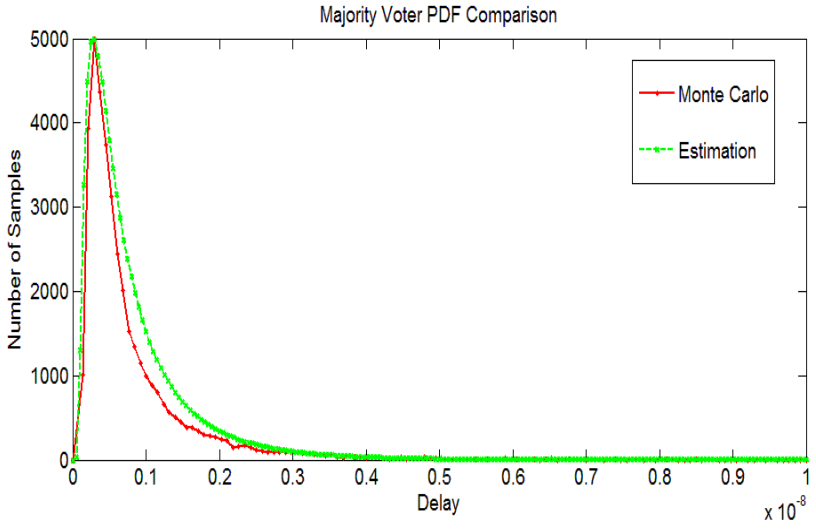
Majority Voter

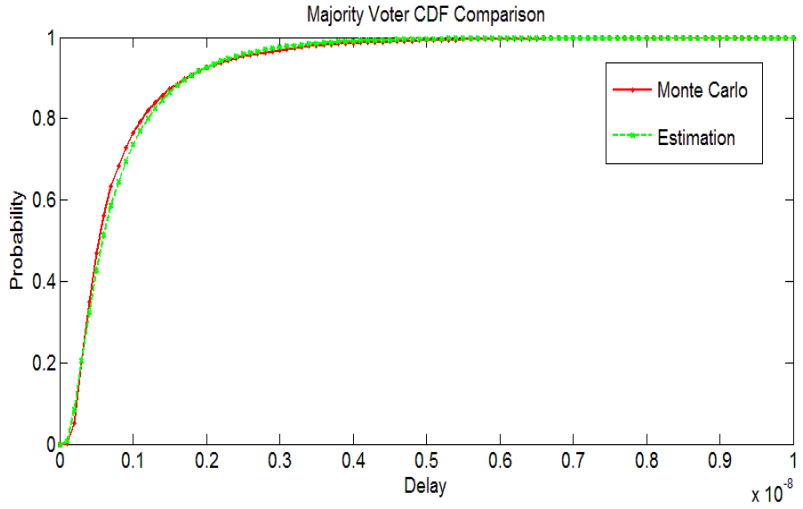
- First, we can get specific μ and λ of every individual cases for both AND and INV.
- And-Inverter Graph (AIG) is used to generate the 3-bit Majority Voter, which gives the output as 1 when more than one input is 1.



- a, b, c are inputs; z is output.
- Dotted line represents a inverter.
- Solid line is just a wire.

Majority Voter – Comparison (CDF)



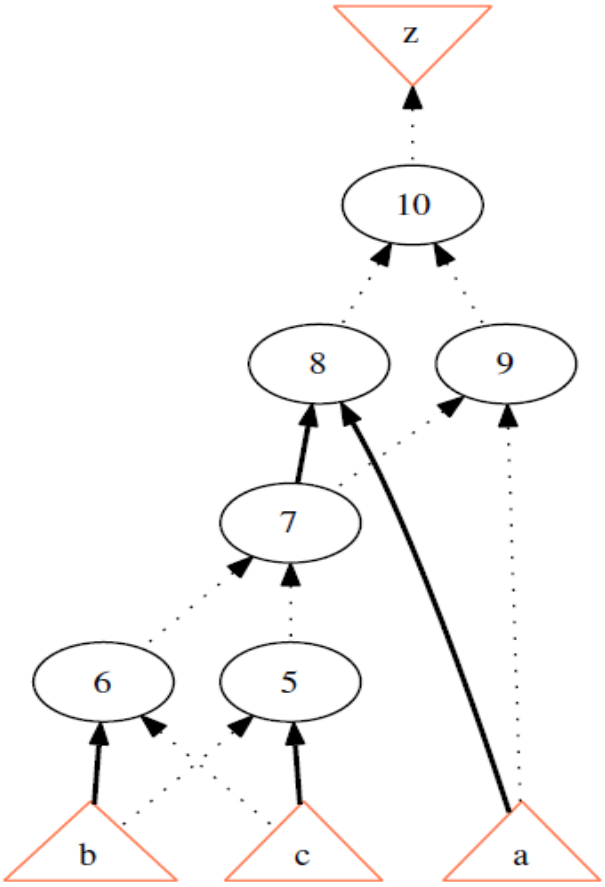


- We considered the longest path in the Majority Voter (3 ANDs, 2 INVs, 22 transistors).
- Red curve stands for Monte Carlo simulation results.
- Green curve stands for statistical approximation.
- Two sets of results are very close within 0.8% average error.

Deviation	1ns	3ns	5ns	7ns	9ns	Average (0-10ns)
Majority Voter	3%	1%	0.3%	0.1%	0%	0.8%

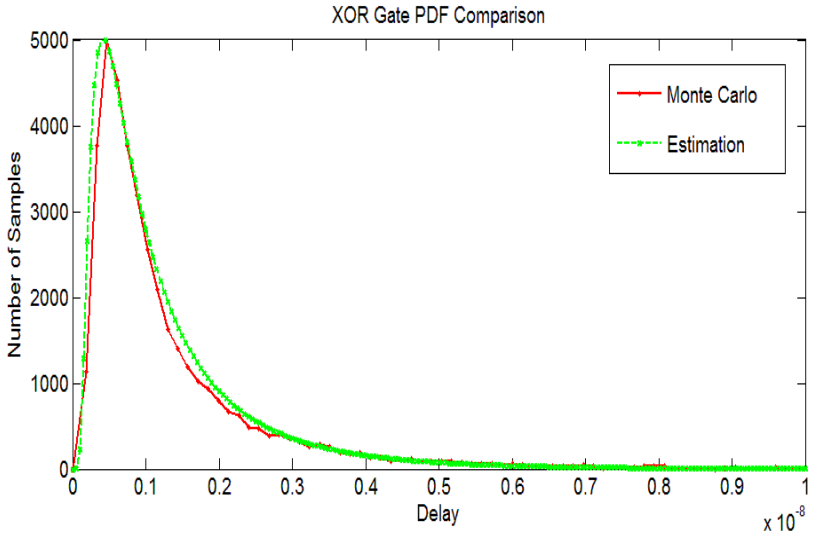
XOR Gate

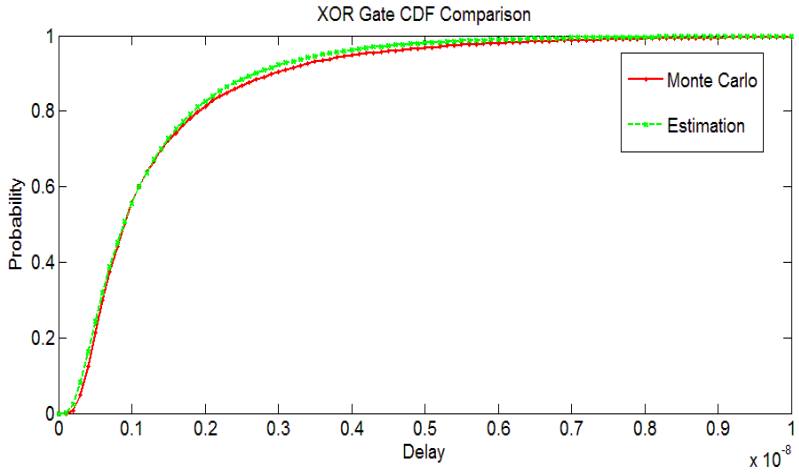
- And-Inverter Graph (AIG) is used to generate the 3-bit XOR gate, which gives the output as 1 when there are odd number of 1's at the inptus.



- a, b, c are inputs; z is output.
- Dotted line represents a inverter.
- Solid line is just a wire.

XOR Gate– Comparison (CDF)





- We considered the longest path in the XOR gate (4 ANDs and 5 INVs, 34 transistors).
- Red curve stands for Monte Carlo simulation results.
- Green curve stands for statistical approximation.
- Two sets of results are very close within 1.2% average error.

Deviation	1ns	3ns	5ns	7ns	9ns	Average (0-10ns)
XOR	0.2%	2%	1.4%	0.7%	0.4%	1.2%

Summary

- Inverse Gaussian (IG) was fitted to the propagation delay, which make more sense than Normal (Gaussian) distribution.
- AND, Inverter were explored and tested. Scaling of IG was demonstrated.
- Majority Voter and XOR were built using AIG.
- The longest path of each circuit was chosen. Monte Carlo simulation vs statistical estimation was carried on.
- The comparison of PDF,CDF graphs were exhibted to show the high accuracy of IG approximation.

Acknowledgement

- This work has been sponsored by the European Commission FP7 FET-Open iRISC (Innovative Reliable Chip Designs from Unreliable Components) project.

- Also, I would like to thank Science Foundation Ireland (SFI).