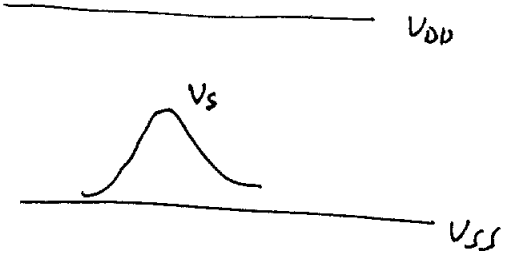
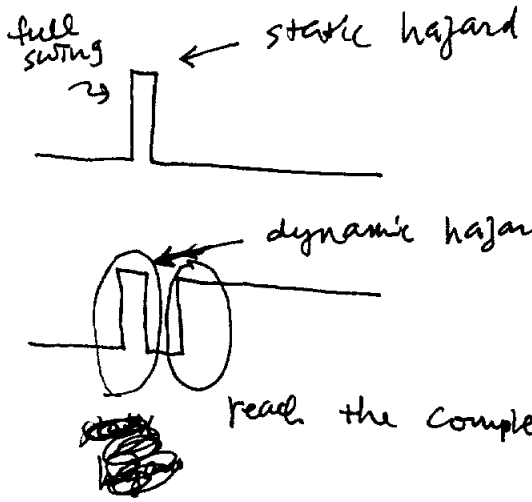
Power Estimation

- Emphasize on transistor level and gate level.
- will cover RT-level later.
- Concentrate on static CMOS technology.
- In logic level power analysis, ASIC design style is assumed with precharacterized library data.
- Def. A <u>glitch</u> is an incomplete signal transition on signal S, i.e., the voltage V_S leaves from its static value and returns to its initial value without performing a full $V_{DD} - V_{SS}$ voltage swing.



- Def. Hazard: A <u>hazard</u> is a useless full-swing signal transition of a signal S within one clock cycle s.t. the signal leaves its static value, reaches the complementary value and returns to its previous static value (static hazard).



read the complementary value of initial value.

- Estimation vs. Analysis.

power estimation: a forecast of the expected power consumption of a device.

(Applied before detailed design data are available)

(Top-down design fashion).

* used to evalnate different design alternatives.

power analysis: Calculates the power consumption of a given physical implementation.

(bottom up process).

* used to verify the design.

- Average power estimation (analysis)

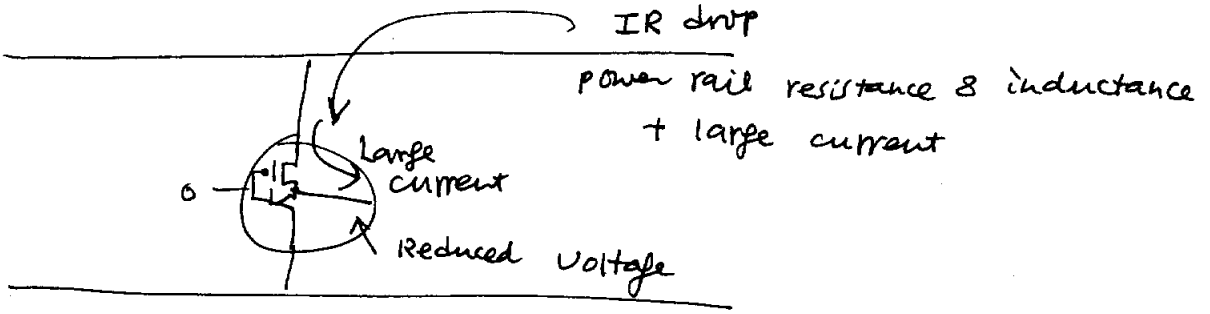
  * Deal with average activity of the circuit

  * for calculation of average heat dissipation of the die.
    average battery life time.

- Worst power estimation (analysis)

  * for reliability analysis.

    ➡ Large supply current ⟶ Temporary malfunction



∴ Cause malfunction

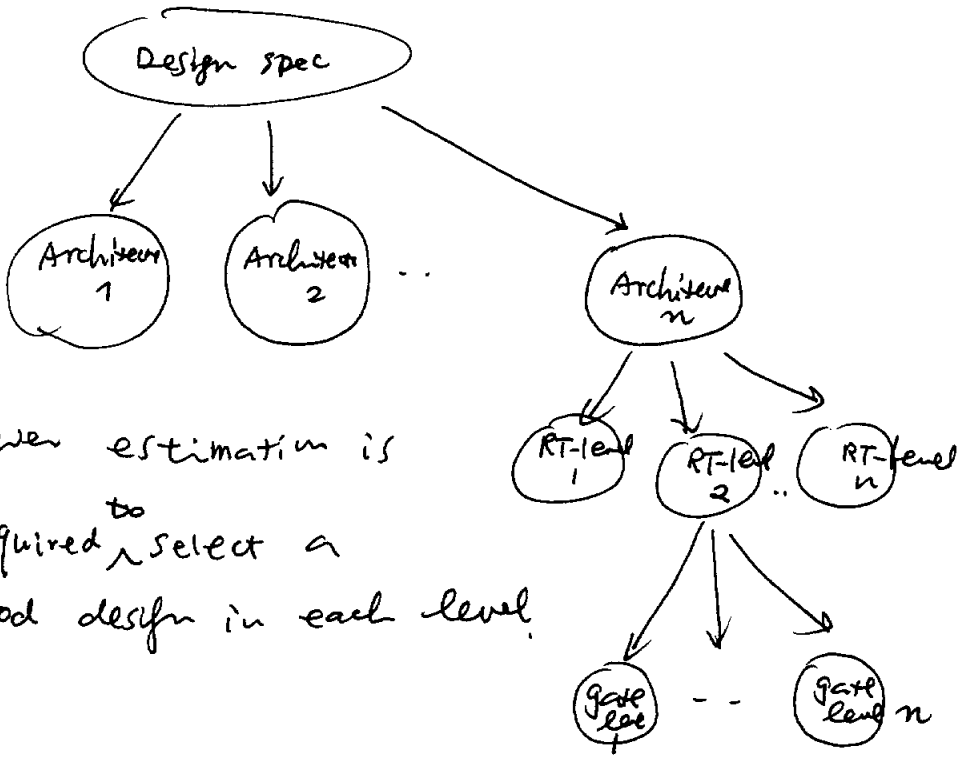
Large supply current ⟶ permanent damage

∵ electromigration effects

e.g.; bridge or open faults caused by metal migration.

Applications of power Estimation & Analysis.

① Optimization for Low-power



* power estimation is required to select a good design in each level.

* Estimation can't be accurate.
  Can only get relative values.

② Quality:

Compare specified power consumption with the achieved power dissipation

③ Reliability Analysis:

Electromigration: power analysis might suggest a good wire dimensi (worst power)

Hot spot: Can be avoided by power optimization. (worst power)

package: Select a good packaging to dissipate heat (avefe power)

Data Dependency

- Power dissipation of a ckt depends on primary input patterns
- Power analysis requires knowledge of typical activity at inputs. (can be achieved by simulation for ASIC chips)

- Expected value $E(s)$ for a signal $s(t)$:

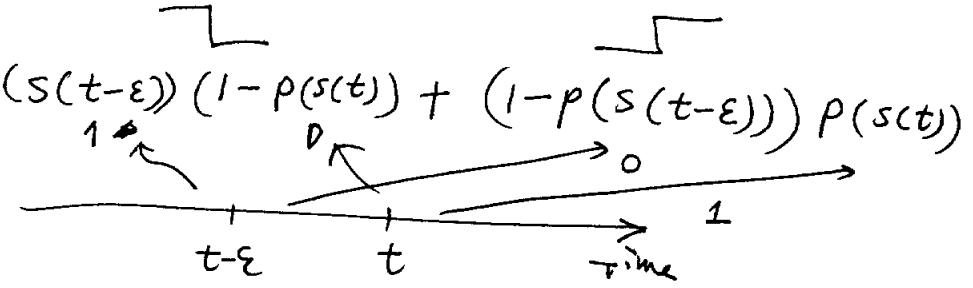
$$E(s) = \lim_{T\to\infty} \frac{1}{T} \int_{t=0}^{t=T} s(t)\,dt \qquad s(t) = 0 \text{ or } 1$$

$$= P(s=1) = p(s).$$

$$E(\bar{s}) = p(\bar{s}) = p(s=0) = 1 - p(s).$$

- Signal transition probability

$$P_{sw}(s(t)) = p(s(t-\varepsilon))\,(1 - p(s(t)) + (1 - p(s(t-\varepsilon)))\,p(s(t))$$



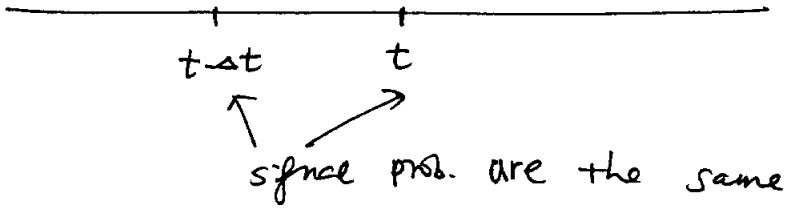
$$= p(s(t-\varepsilon)) + p(s(t)) - 2p(s(t-\varepsilon)\,p(s(t)))$$

- Signal correlations
  - Uncorrelated signals
  - Structural correlation
  - temporally correlated signals — sequential correlation.
  - spatially correlated signals
  - spatiotemporal correlated signals.

• uncorrelated input signals

- very rare
- Assume all inputs are independent.
- Assume all signals are stationary

$$P(S(t)) = P(S(t-\Delta t)) = P(S),$$



$$\therefore \quad P_{SW}(S_i(t)) = P_{SW}(S_i) = 2P(S_i)(1-P(S_i))$$

$$P(O_{AND}) = \prod_{\text{all inputs}} P(i_{AND}) \quad \leftarrow \text{input } i \text{ equals } 1$$

$$P_{SW}(O_{AND}) = 2P(O_{AND})(1-P(O_{AND}))$$

$$E(SW) = P(S_i)(1-P(S_i)) \cdot 1 + P(1-P(S_i)) \cdot P(S_i) \cdot 1$$

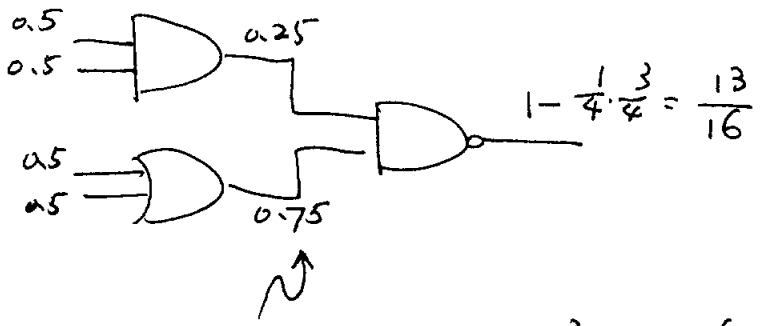
$$= 2P(S_i)(1-P(S_i))$$

$$\therefore P_{SW}(S_i(t)) = E(SW).$$

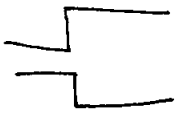
$$P(O_{OR}) = 1 - \prod_{\text{all inputs}} (1-P(i_{OR})) \quad \xleftarrow{\text{input } i \text{ equals } 1.}$$

$$P_{SW}(O_{OR}) = 2P(O_{OR})(1-P(O_{OR}))$$

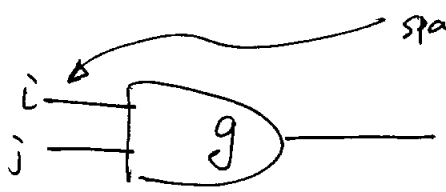
Exaple:



$P_{sw} = 2\left(\frac{1}{4}\right)\left(\frac{3}{4}\right) = \frac{6}{16} = \frac{3}{8}$ from P_{sw} we can estimate power dissipation



- Spatial Correlations.



spatial correlation between inputs.

E.g. if $i=0 \rightarrow j=0$

if $j=1 \rightarrow i=1$

* only patterns 00 and 11 can be applied to i & j.

* Both patterns are equally likely.

i	j	output
$0\rightarrow1$	$0\rightarrow1$	$0\rightarrow1$ *
$1\rightarrow0$	$1\rightarrow0$	$1\rightarrow0$ *
$0\rightarrow0$	$0\rightarrow0$	$0\rightarrow0$
$1\rightarrow1$	$1\rightarrow1$	$1\rightarrow1$

$E_g(sw) = \frac{1}{2}$

$E_g(sw) = P(0\rightarrow1)\cdot 1 + P(1\rightarrow0)\cdot 1$

$= \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 1$

$= \frac{1}{2}$

Without the spatial correlation:

$\therefore E_g(sw) = 2\left(\frac{1}{4}\right)\left(\frac{3}{4}\right) = \frac{3}{8}$

$E_g(sw) = P(0\to1)\cdot 1 + P(1\to0)\cdot 1$

$= P(0)\cdot P(1)\cdot 1 + P(1)P(0)\cdot 1$

$= \frac{3}{4}\cdot\frac{1}{4} + \frac{1}{4}\cdot\frac{3}{4}$

$= \frac{3}{8}$

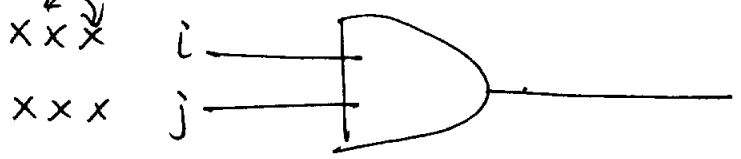
i	j	output
a→b	c→d	
16 → (2)	(2)	6 of the changes in input cause output switching

16 different input changes.

$\therefore \frac{6}{16} = \frac{3}{8}$

Temporal Correlation

timing relation.



Example: Every 0 applied to i is immediately followed by 1.

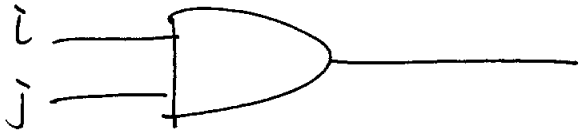
Every 1 —— j —— 0.

$E_g(sw) = \frac{4}{9}$

i	j	output
0→1	0→0	0→0
0→1	0→1	0→1 *
0→1	1→0	0→0
1→0	0→0	0→0
1→0	0→1	0→0
1→0	1→0	1→0 *
1→1	0→0	0→0
1→1	0→1	0→1 *
1→1	1→0	1→0 *

$\frac{4}{9}$

- Spatiotemporal Correlation



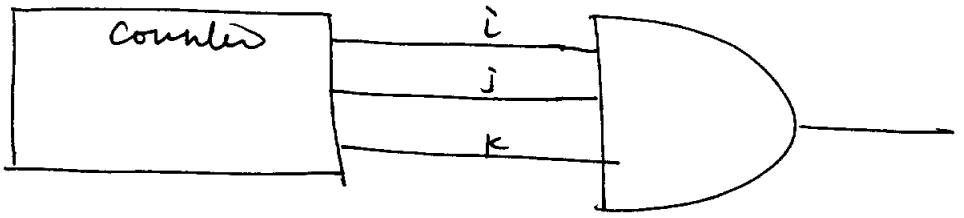
Ex. ~~i changes exactly if j changes.~~

Only patterns 00 and 11 can be applied to i & j, and 0 in i is immediately followed by 1.

i	j	output
0→1	0→1	0→1 *
1→0	1→0	1→0 *
1→1	1→1	1→1

$Eq(SW) = \frac{2}{3}$

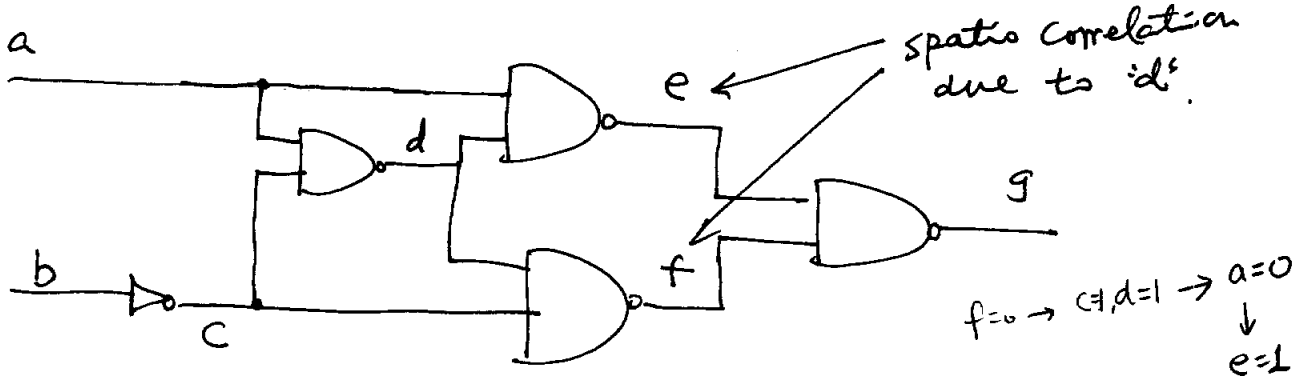
Example :



i	j	k
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

The value of this vector depends on previous history (temporal) of each bit (spatio).
∴ spatiotemporal.

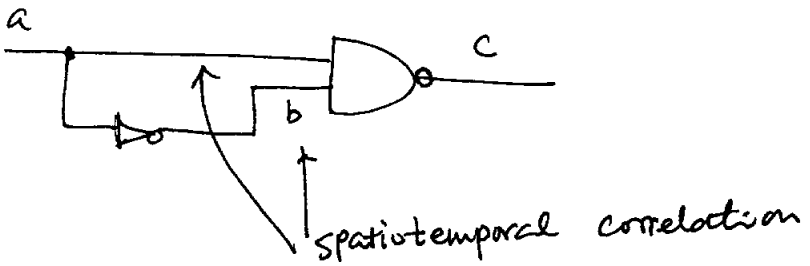
- Structural correlation

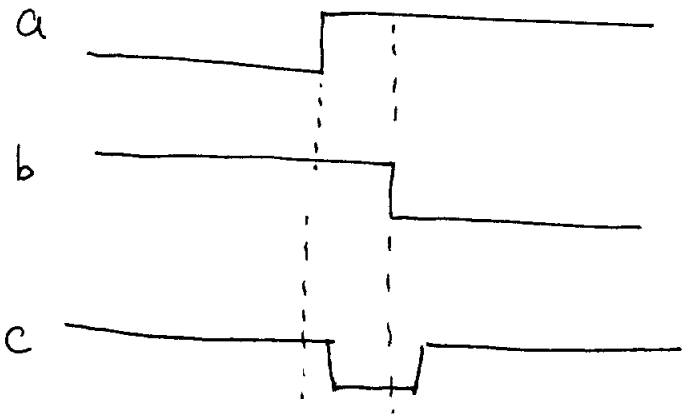


a	b	c	d	e	f
0	0	1	1	1	0
0	1	0	1	1	1
1	0	1	0	1	1
1	1	0	1	0	1

if f=0 → e=1

if e=0 → f=1

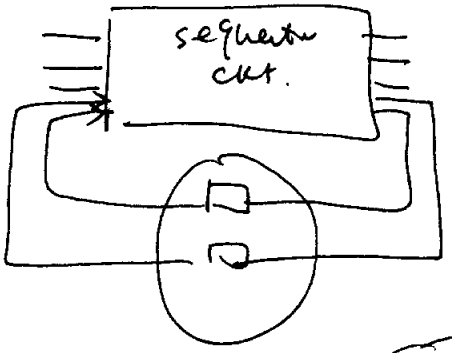


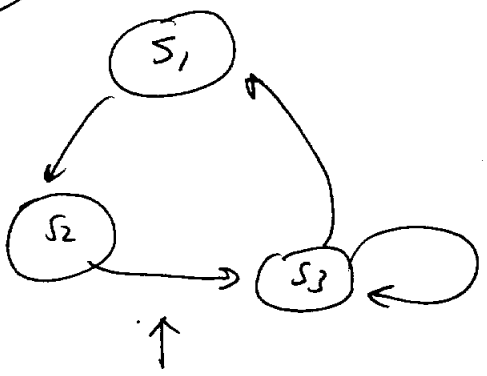


b has relation with a (spatio)

b has delay of a (temporal).

- Sequential Correlations.





temporal correlation.

Relative Importance.

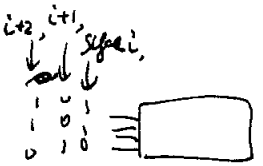
Average relative error

$$e_{average} = \left| \frac{\sum E(sw)_{accurate} - \sum E(sw)_{ignored-Correlati-}}{\sum E(sw)_{accurate}} \right|$$

↑ for all signals

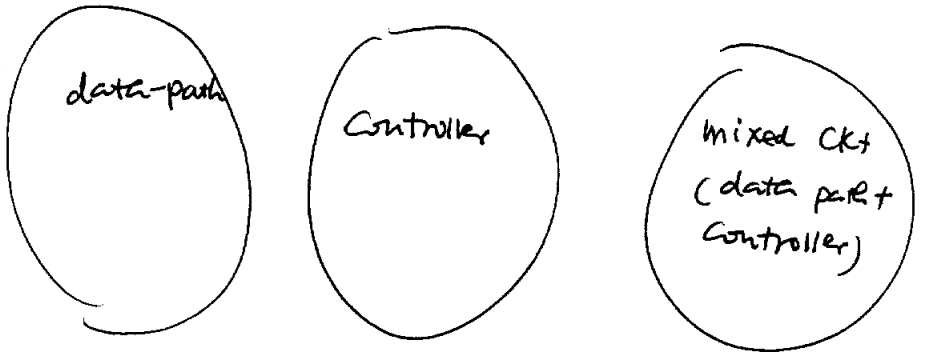
Note: An overestimate of a signal can be compensated by an underestimate of another signal.

Relative signal error

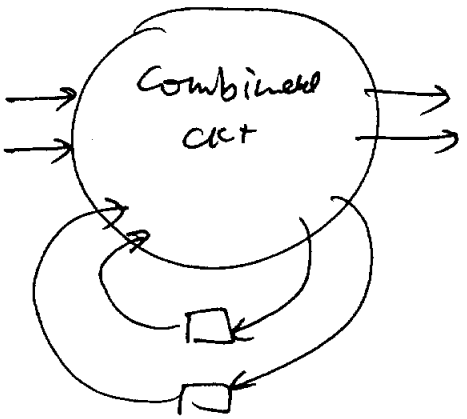


$$e_{signals} = \frac{1}{\#\ signals} \sum_{signal\ i} \left| \frac{E(sw)_{accurate} - E(sw)_{ignore-Correlation}}{E(sw)_{accurate}} \right|$$

Example: Build up a ckt & do simulation



Controller:



* ignore the sequential correlation at ~~signals~~ state

$e_{average} = 25\%$ $e_{signal} = 120\%$

* ignore spatial and temporal correlations at PI's are less critical.

Data-path: No sequential elements.

* ignore spatial and temporal correlations at PIs

$e_{average} = 8\%$ $e_{signal} = 30\%$

Mixed (IIR Filter):

* ignore spatial & temporal correlation at PI's

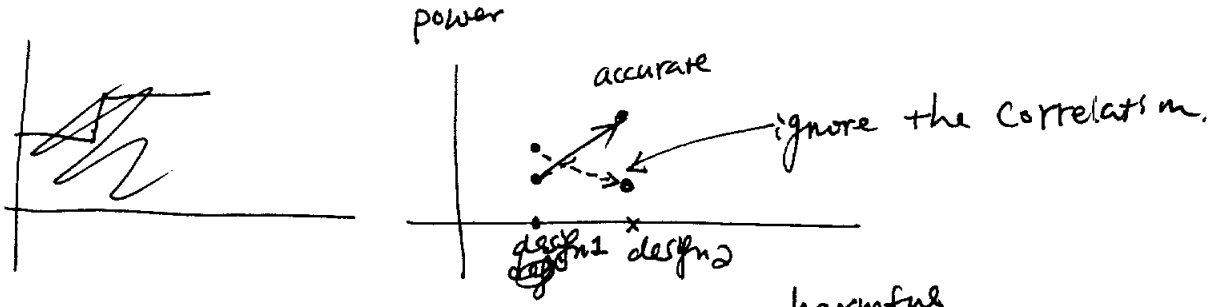
$e_{averge} = 26\%$ $e_{signal} = 5\text{-}6\%$

Conclusions:

① Errors made for signals >> Errors made for average

② Error by ignoring signal correlation may hide the improvement or deterioration of optimizing transformation

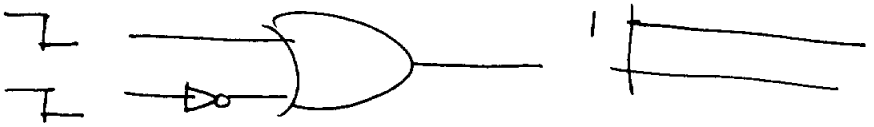
E.g.



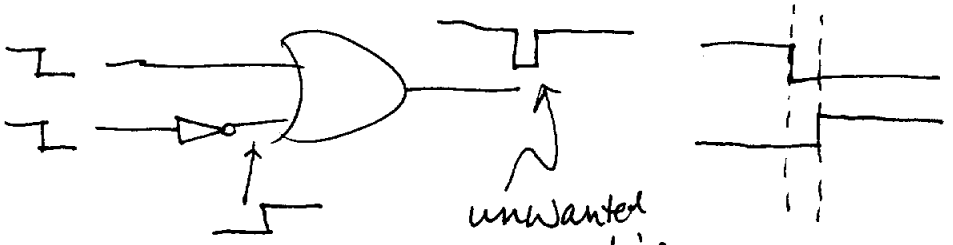
This causes harmful power optimization.

Delay Models

- Also affect the switching activity and make power analysis difficult



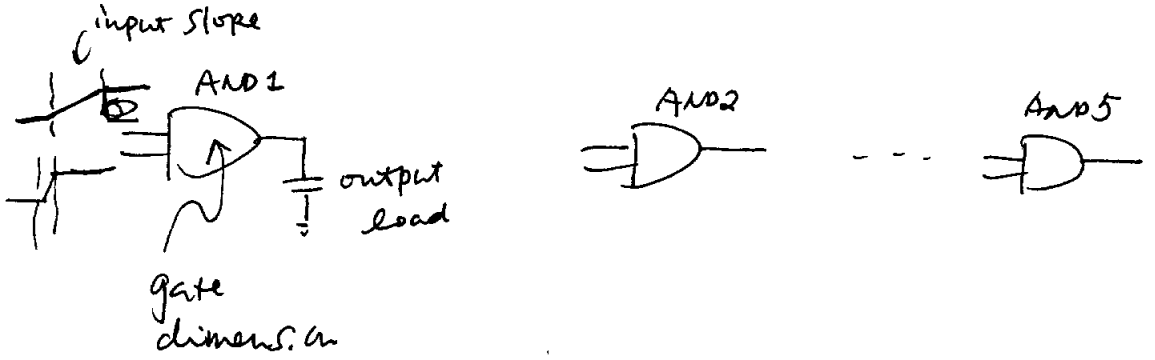
Zero delay model



and consume power.

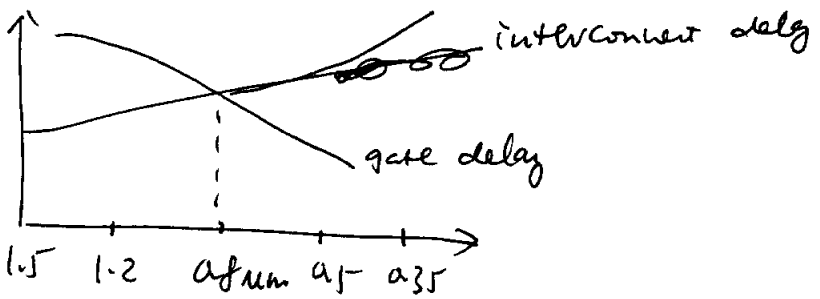
- Delay models highly affect the hazard and glitch analysis.
- The ratio of hazardous components to the total power dissipation is about 9% ~ 38%. (mean value of this value is 15-20%).
- Zero Delay.
  - Cycle-based simulation: the value of each net is computed only once at the end of the clock cycle.
  - hazards will not be detected.
  - For early estimation power of Boolean functions.
  - Ignores glitch or hazard power consumption
    ∴ underestimate the power dissipation.
- ~~Delay~~ Unit delay
  - Assume a common constant delay for all gates.
  - Can generate and propagate hazards.
  - ~~no~~ limited real application

- Real delay
  - Based on cell library characterization.
  - Based on pin-to-pin delay for output load and input slope.
  - Should be able to handle glitch and hazard



Create table for transition time and power consumption for each cell.

- Interconnect delay dominates in deep submicron, and must be considered



- Should use backannotation data after layout has been available.

- power analysis (estimation)

  * pattern dependent simulation techniques:

    Most commertial tools use this method.

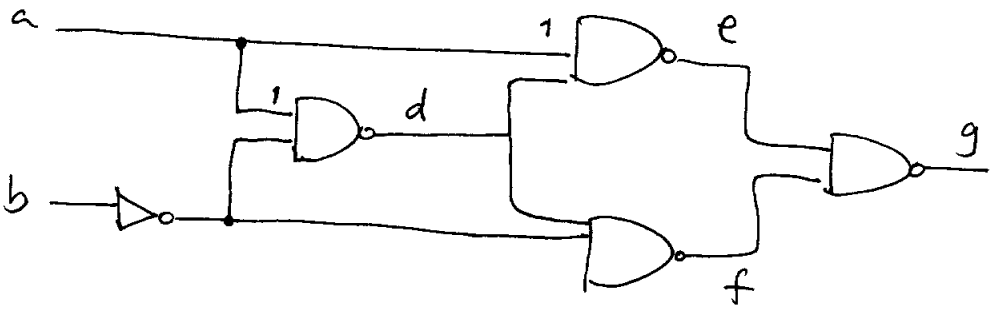
  * pattern Independent Analysis. ← symbolic simulation
    ← probabilistic simulation.

    Impossible solution so far.

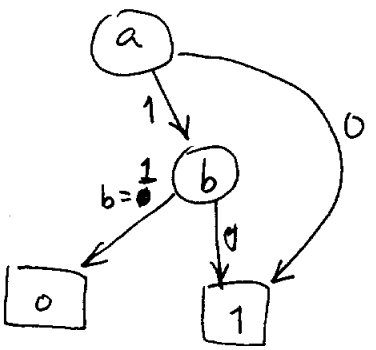
- pattern Independent Analysis

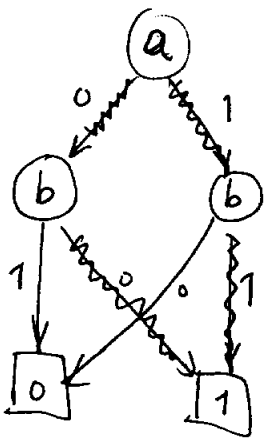
  - will just show the concepts.

  - Symbolic simulation: calculates the switching activity of each node from statistical properties of the PI's in an analytical way.
  - Show by example:



Binary decision diagram





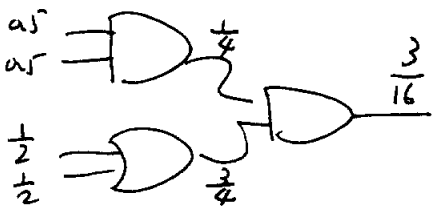
$P(e) = 1 - P(a) \cdot P(b)$

$P(g) = P(\bar{a}) \cdot P(\bar{b}) + P(a) \cdot P(b)$

• Impossible solution if ckt is large.

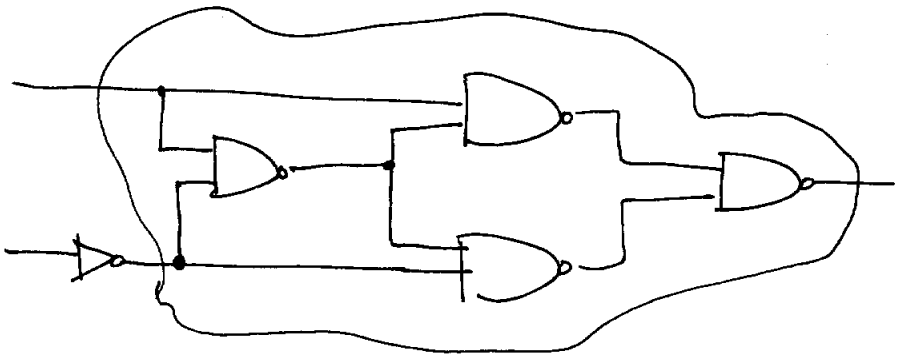
(For medium to large ckts, the symbolic formulae become too large to build).

probabilistic simulation.



* use exact probability simulation for reconvergent fanout ckt area

* Example:



* Use exact prob. analysis.

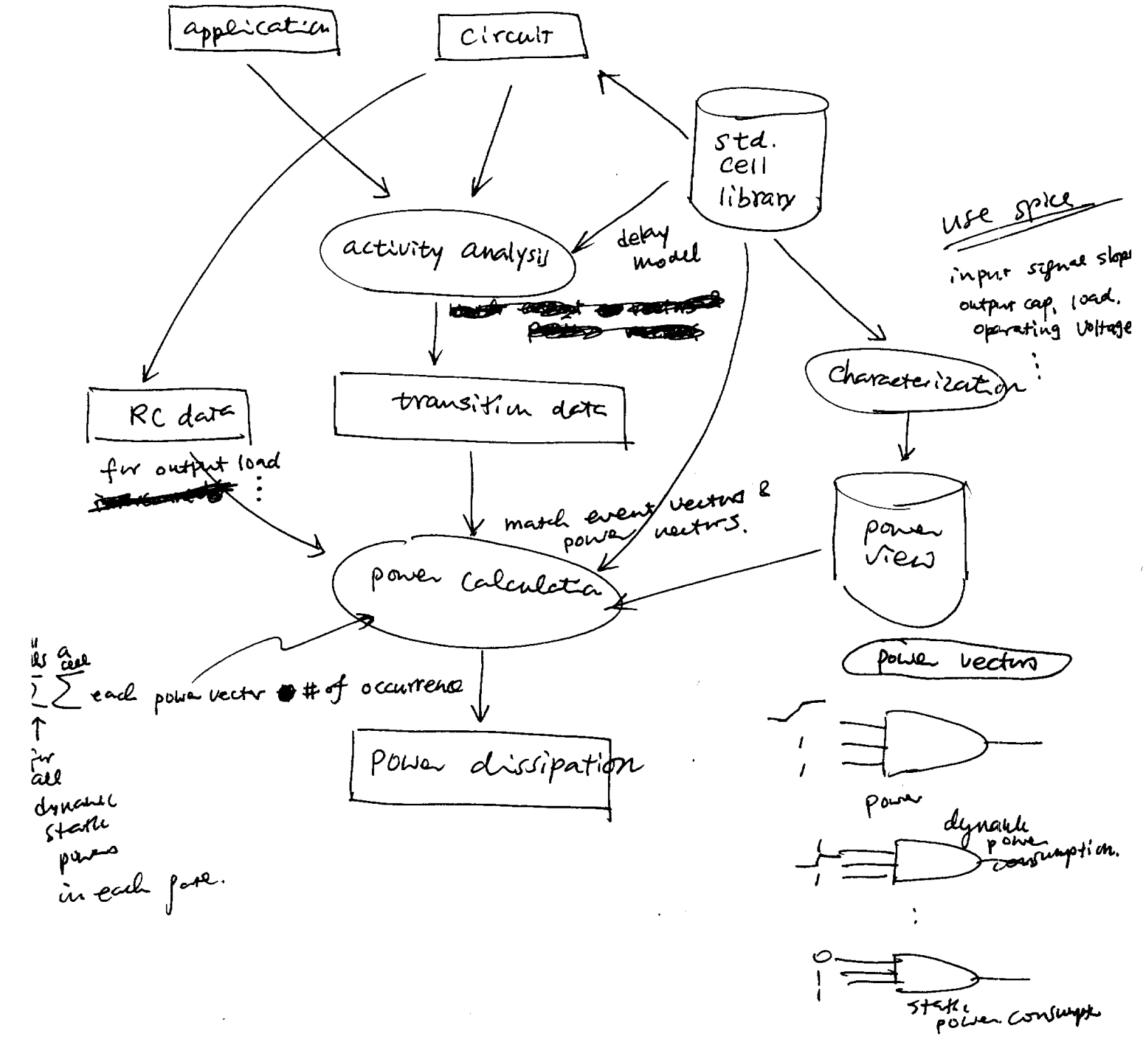
* Impossible if ckt size is large.

Pattern-Dependent simulation Techniques

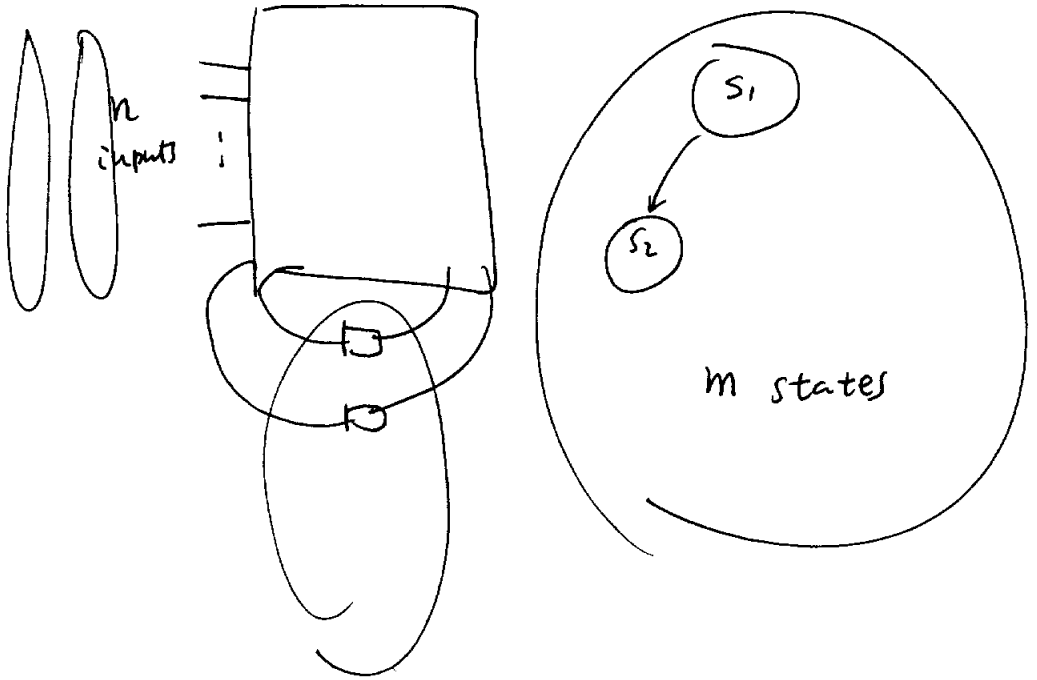
• Most commertial tools use this approach.

• Can be done by different levels of abstraction.

- Electrical level: no library characterization required
- Abstract level: Library characterization, activity analysis, power calculation.
- Basic idea



- Exhaustive simulation
  - Brute force method



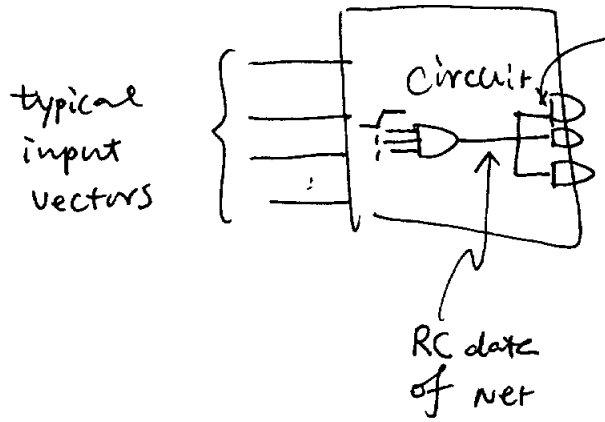
  - The total # of simulation vectors is $m \cdot 2^n \cdot 2^n$

first input → 2^n; 2nd input → 2^n

Try all different input transitions for each state !!!

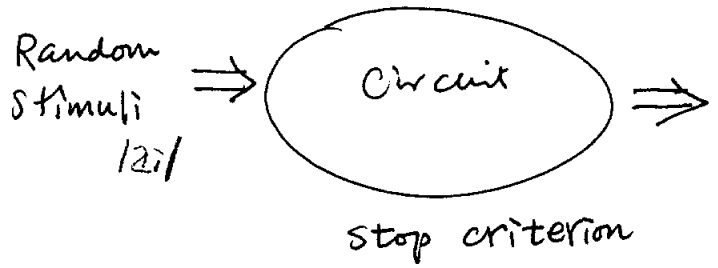
  - impossible solution

- Application pattern

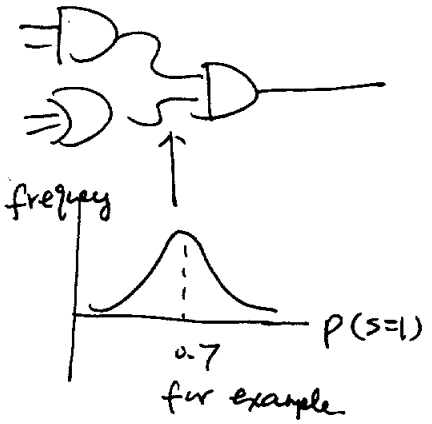


# of inputs ⟷ accuracy (trade off)

  - Try to find a minimum set of stimuli (representative) for a required accuracy.
  - described next.



· Basic idea: for typical ckts and long observation periods, the signals behave like normally distributed stochastic variables.



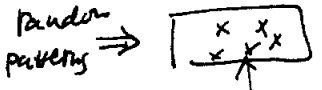
· ~~Confidence Intervals~~
stopping criteria

The total average power dissipated in a ckt during time interval T is

$$P_T = \frac{V_{dd}^2}{2} \sum_{i=1}^{m} C_i \frac{N_{x_i}(T)}{T}$$

m nodes; # of transition at node i

Basic idea:



* observe P_T.
* stop if criteria satisfied

T → may be many clock cycles

- Assume P_T is normal distribution for any T.
- we perform N different simulations, each with time T.
- The sample average is η_T
- Sample standard deviation S_T

we have $(1-\alpha)$ x 100% confidence that

$$\left| \eta_T - E[P_T] \right| < \frac{t_{\alpha/2} S_T}{\sqrt{N}}$$

where $t_{\alpha/2}$ is obtained from t distribution.

with $(N-1)$ degree of freedom.

$$t = \frac{(\bar{x} - \mu)\cdot\sqrt{n}}{\hat{\sigma}}$$

↑ used for σ^2 unknown

$$\therefore \frac{\left| \eta_T - E[P_T] \right|}{\eta_T} < \frac{t_{\alpha/2} \cdot S_T}{\eta_T \cdot \sqrt{N}} < \varepsilon$$

↑ would like to have error tolerance ε.

Try to control

$$\frac{t_{\alpha/2} \cdot S_T}{\eta_T \cdot \sqrt{N}} < \varepsilon.$$

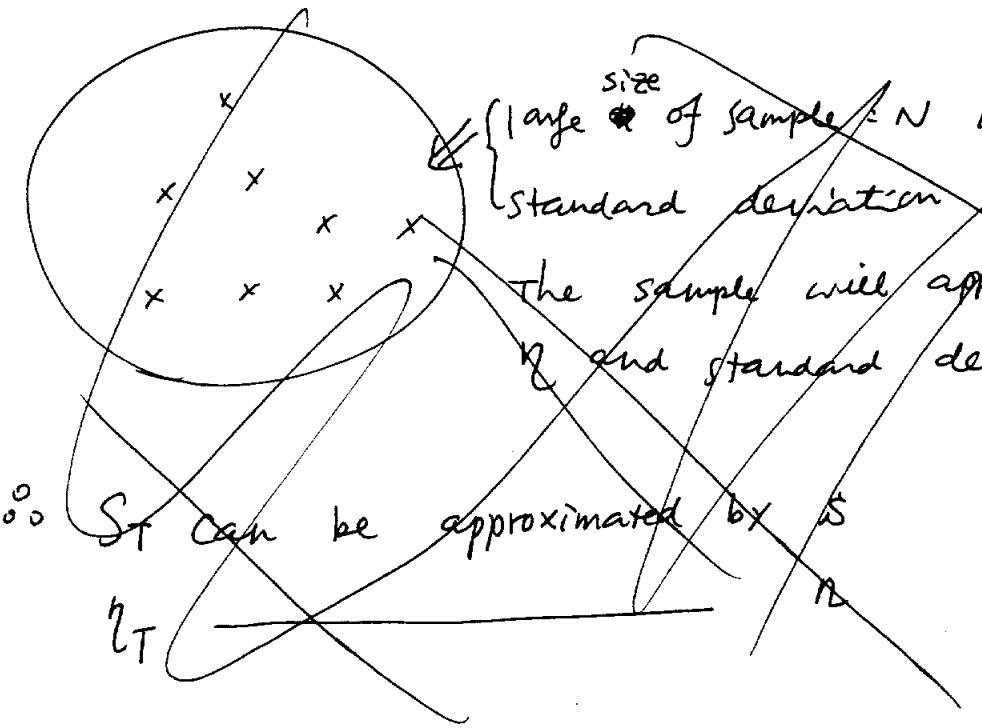
I have 99% ($\alpha = 0.01$) of confidence that sample mean and true mean are different by ε.

$\therefore$ the minimum value of N is

$$N \geq \left(\frac{t_{\alpha/2} \cdot S_T}{\eta_T \cdot \varepsilon} \right)^2$$

- Determine a minimum sample size N for a specified error tolerance ε and a specified confidence interval $(1-\alpha)$ 100%.

· By Law of Large Numbers



{large size of sample = N with mean η and standard deviation S.

the sample will approach true mean η and standard deviation σ.

∴ S_T can be approximated by S

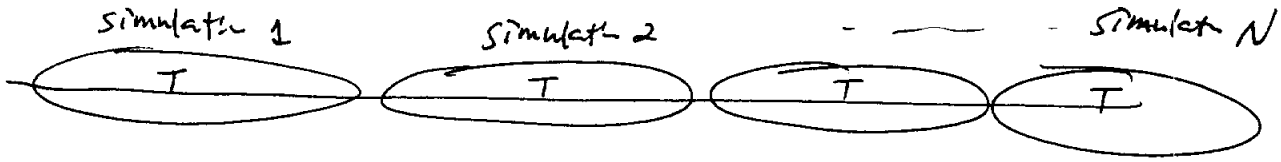
η_T ≈ η

∴ N can be determined.

$t_{\alpha/2}$: typically between 2.0 and 5.0

ϵ : constant

$$\therefore N \propto \frac{S_T^2}{\eta_T^2}$$



find S_T & η_T for all simulations till

$$\frac{t_{\alpha/2} \cdot S_T}{\eta_T \sqrt{N}} < \epsilon.$$

Stop !!!!

- For combinational ckts, sample sizes of 20-50 might be sufficient to converge.

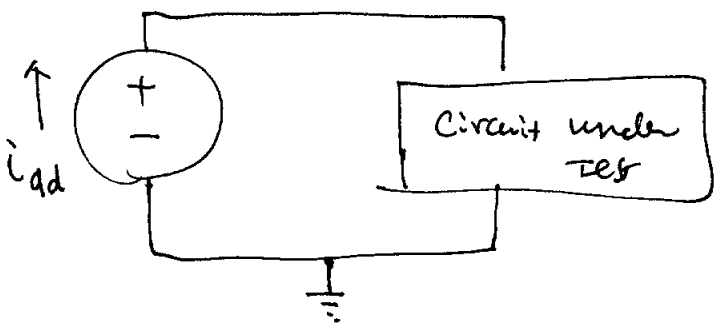
- Low-activity nodes may not converge fast. However, low-activity nodes do not contribute too much power consumption, any how.

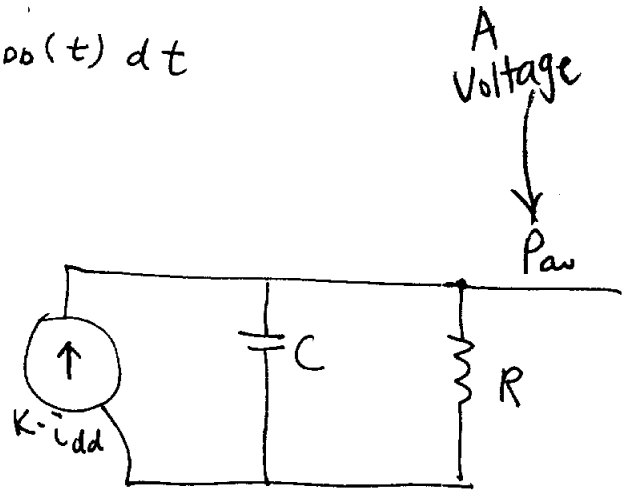
- Transistor level power analysis
  - Spice
  - IRSIM
  - PowerMill

- Circuit Level — spice (A differential equation solver).
  - very accurate
  - Run time is prohibitive.
  - Mainly used to build up cell library.
  - How to estimate power by spice?
  - Basic idea

- can also take care of short-ckt power

$$P_{av} = \frac{1}{T}\int_0^T p(t)\,dt = \frac{V_{DD}}{T}\int_0^T i_{DD}(t)\,dt$$





$$C \frac{dP_{ave}}{dt} = K \cdot i_{DD}$$

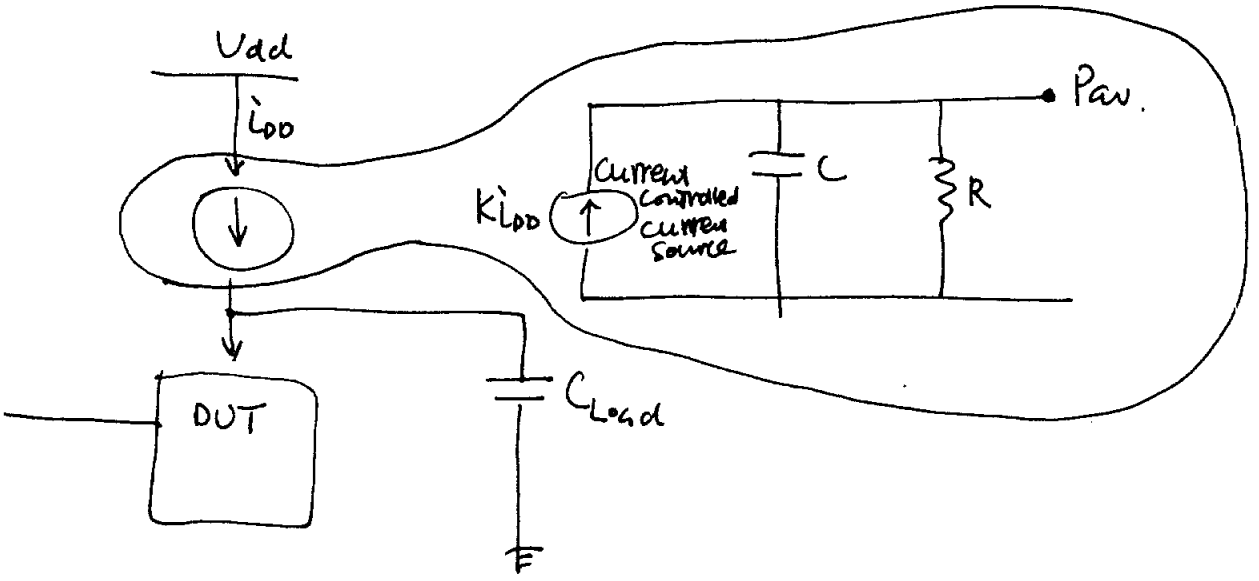
$$\therefore \quad P_{ave} = \frac{K}{C} \int_0^T i_{DD}(t)\, dt$$

∴ We just need to observe the output voltage P_{ave} above if parameters can be cleverly chosen.

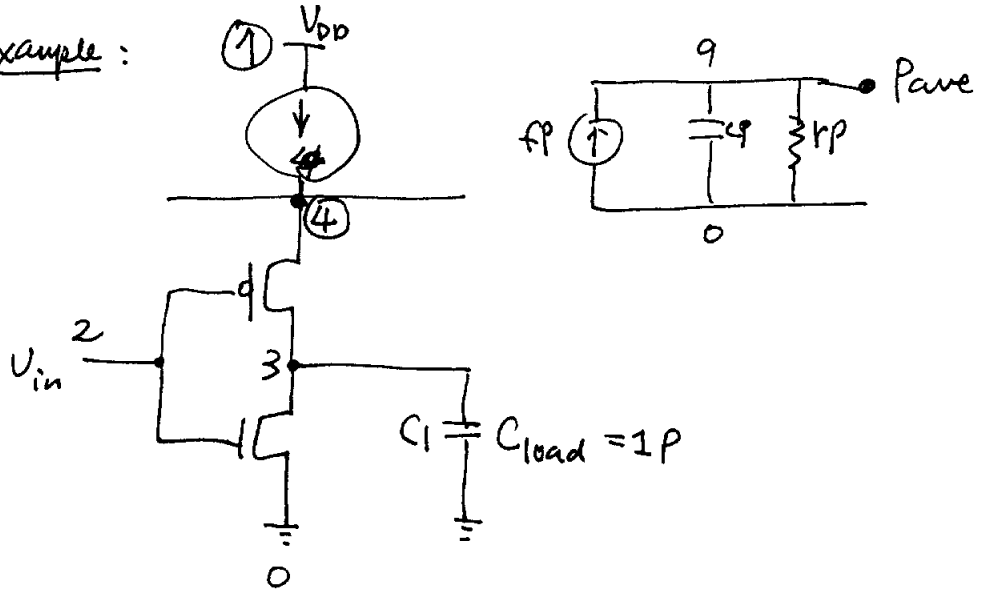
$$\therefore \quad \frac{U_{DD}}{T} = \frac{K}{C}$$

Note: R is only provided for DC-convergence reasons, should be chosen as high as possible to minimize leakage

∴ Setup:



Example :



	drain	gate	source	sub	type		
mn	3	2	0	0	nmod	w = 10u	l = 1u
mp	3	2	4	1	pmod	w = 20u	l = 1u
Vdd	1	0	5				
	node	node	volt				

Vtstp 1 4 0 ← dummy voltage source

0V

No voltage drop from 1 → 4

To use a current-controlled current source, a dummy independent voltage source is often placed into the path of the controlling current.

- model nmod nmos
  :
- model pmod pmos
  :

```
Vin   2  0   pulse (0 5 8n 2n 2n 8n 20n)
C1    3  0   1p
Fp    0  9   Vtstp   0.025
rp    9  0   100K
Cp    9  0   100p
```

current source — Vtstp; gain — 0.025

Current-dependent current source — Fp; name

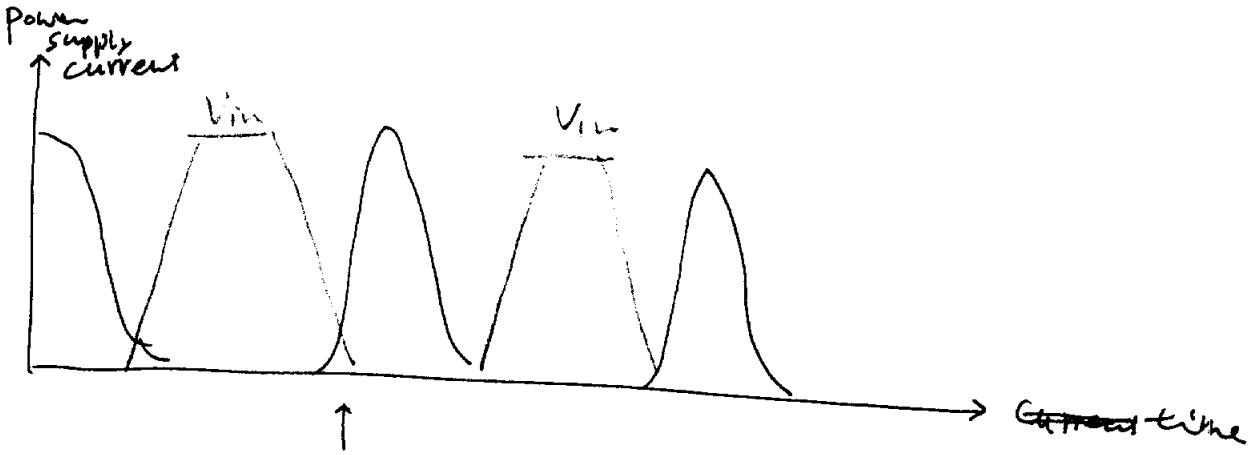
0.025 corresponds to K value of the equation

100K ← 100KΩ

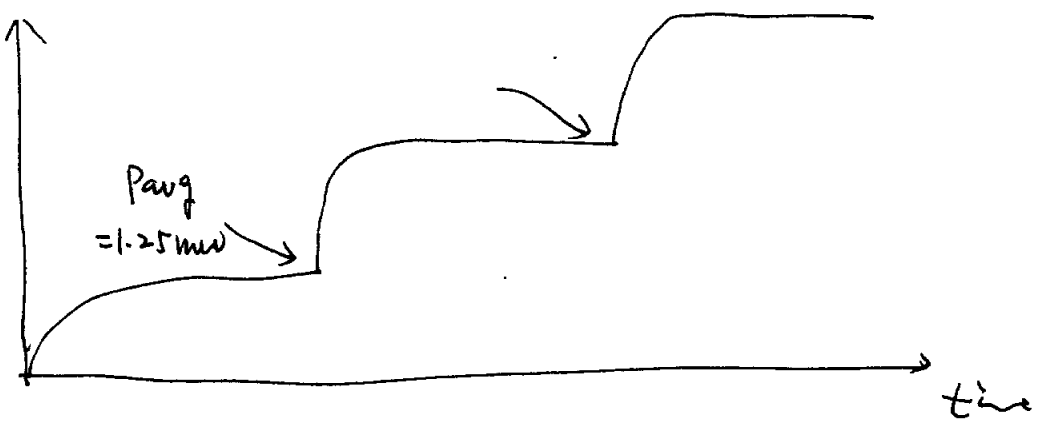
100p ← one hundred pF

$\frac{V_{DD}}{T} = \frac{K}{C}$

```
.tran   1n   60n   uic
.print  tran  V(3)  V(2)
.print  tran  i(Vtsp)
.print  tran  V(9)
.end
```



power supply current drawn from V_{DD} when charge-up the output capacitor



Current Dependent Current Sources — F Elements

F element syntax statements are described in the following paragraphs. The parameter definitions follow.

Current Controlled Current Source (CCCS)

Syntax

Linear

```
Fxxx n+ n- <CCCS> vn1 gain <MAX=val> <MIN=val> <SCALE=val> <TC1=val>
+       <TC2=val>      <M=val> <ABS=1> <IC=val>
```

Polynomial

```
Fxxx n+ n- <CCCS> POLY(NDIM) vn1 <... vnndim> <MAX=val> <MIN=val>
+     <TC1=val> <TC2=val> <SCALE=vals> <M=val> <ABS=1> P0  <P1...>
+     <IC=vals>
```

Piecewise Linear

```
Fxxx n+ n- <CCCS> PWL(1) vn1 <DELTA=val> <SCALE=val><TC1=val> <TC2=val>
+     <M=val>  x1,y1 ... x100,y100 <IC=val>
```

Multi-Input Gates

```
Fxxx  n+ n- <CCCS> gatetype(k) vn1, ... vnk <DELTA=val> <SCALE=val> <TC1=val>
+     <TC2=val> <M=val> <ABS=1> x1,y1  ...  x100,y100  <IC=val>
```

Delay Element

```
Fxxx n+ n- <CCCS> DELAY vn1 TD=val <SCALE=val> <TC1=val><TC2=val>
+       NPDELAY=val
```

Parameter Definitions

ABS	Output is absolute value if ABS=1.
CCCS	the keyword for current controlled current source. Note that CCCS is a reserved word and should not be used as a node name.

DELAY	keyword for the delay element. The delay element is the same as a current controlled current source except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the macromodel process. **Note:** DELAY is a reserved word and should not be used as a node name.
DELTA	used to control the curvature of the piecewise linear corners. The parameter defaults to 1/4 of the smallest breakpoint distances. The maximum is limited to 1/2 of the smallest breakpoint distances.
Fxxx	current controlled current source element name. The parameter must begin with an "F", followed by up to 1023 alphanumeric characters.
gain	current gain
gatetype(k)	can be one of AND, NAND, OR, or NOR. (k) represents the number of inputs of the gate. The x's and y's represent the piecewise linear variation of output as a function of input. In the multi-input gates, only one input determines the state of the output. The above keyword names should not be used as a node name.
IC	initial condition: the initial estimate of the value(s) of the controlling current(s) in amps. If IC is not specified, the default=0.0.
M	number of element in parallel
MAX	maximum output current value. The default is undefined and sets no maximum value.
MIN	minimum output current value. The default is undefined and sets no minimum value.
n+/-	positive or negative controlled source connecting nodes

NDIM polynomial dimensions. If POLY(NDIM) is not specified, a one-dimensional polynomial is assumed. NDIM must be a positive number.

NPDELAY sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep.

That is,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specifed in the .TRAN statement.

P0, P1 ... when one polynomial coefficient is specified, Star-Hspice assumes it to be P1 (P0=0.0) and the source is linear. When more than one polynomial coefficient is specified, the source is nonlinear, and P0, P1, P2 ... represent them.

POLY polynomial keyword function

PWL piecewise linear keyword function

SCALE element value multiplier

TC1,TC2 first and second order temperature coefficients. The SCALE is updated by temperature:

$$SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

TD time delay keyword

vn1 ... names of voltage sources through which the controlling current flows. One name must be specified for each dimension.

x1,...	controlling current through vn1 source. The x values must be in increasing order.
y1,...	corresponding output current values of x

Examples

```
F1 13 5 VSENS MAX=+3 MIN=-3 5
```

This example describes a current controlled current source connected between nodes 13 and 5. The current that controls the value of the controlled source flows through the voltage source named VSENS (to use a current controlled current source, a dummy independent voltage source is often placed into the path of the controlling current). The defining equation is:

$$I(F1) = 5 \cdot I(VSENS)$$

The current gain is 5, the maximum current flow through F1 is 3 A, and the minimum current flow is -3 A. If I(VSENS) = 2 A, I(F1) would be set to 3 amps and not 10 amps as would be suggested by the equation. A user-defined parameter can be specified for the polynomial coefficient(s), as shown below.

```
.PARAM VU = 5
F1 13 5 VSENS MAX=+3 MIN=-3 VU
```

The next example describes a current controlled current source with the value:

$$I(F2)=1e\text{-}3 + 1.3e\text{-}3 \cdot I(VCC)$$

```
F2 12 10 POLY VCC 1MA 1.3M
```

Current flow is from the positive node through the source to the negative node. The direction of positive controlling current flow is from the positive node through the source to the negative node of vnam (linear), or to the negative node of each voltage source (nonlinear).

```
Fd 1 0 DELAY vin TD=7ns SCALE=5
```

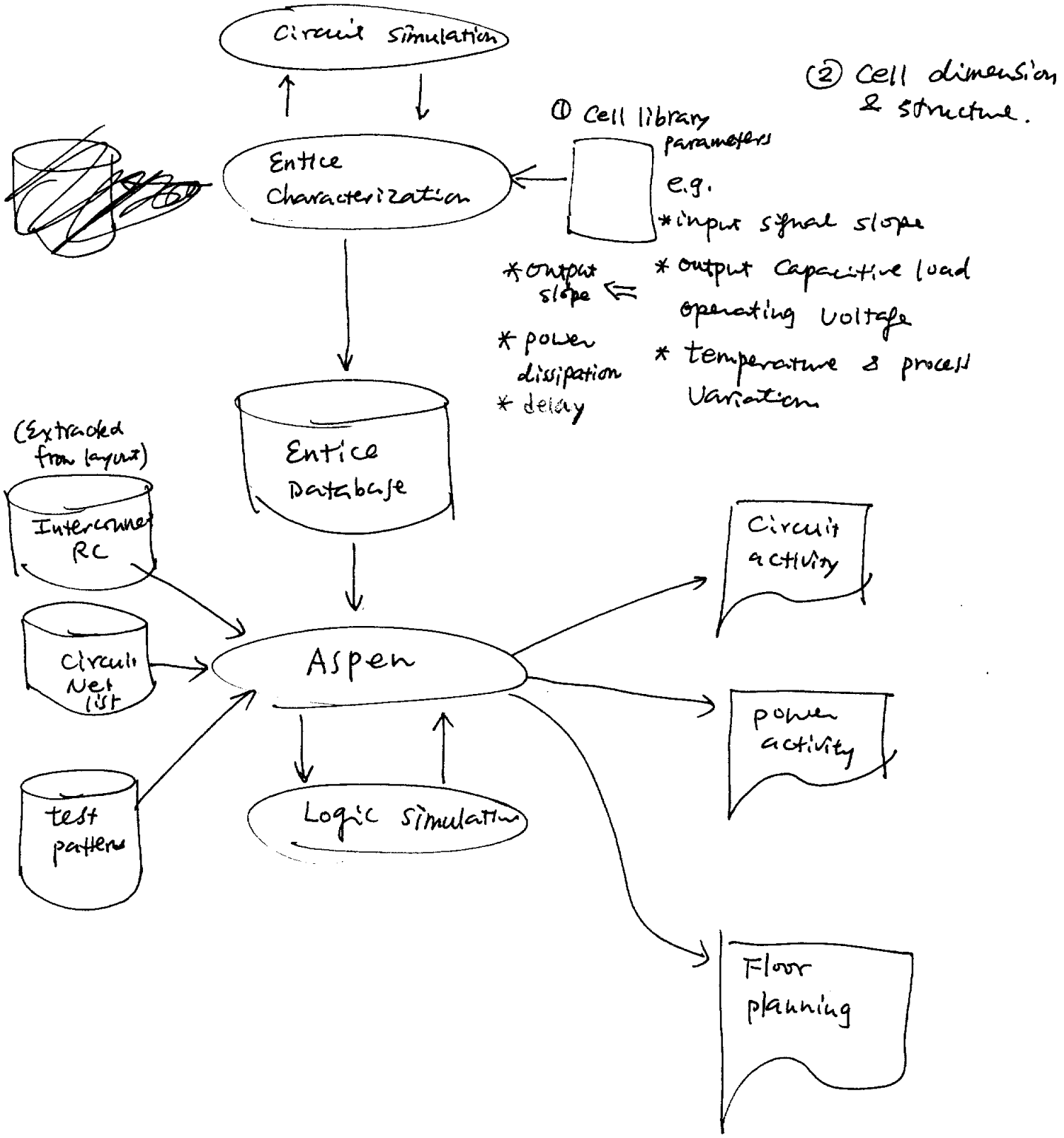
This example is a delayed current controlled current source.

```
Filim 0 out PWL(1) vsrc -1a,-1a 1a,1a
```

The final example is a piecewise linear current controlled current source.

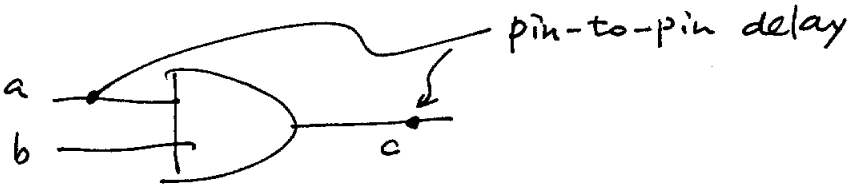
Logic Level power Estimation

* Entice-Aspen developed by Motorola.



power characterization in Entice

- Entice: Cell characterization system that models power and timing delays.



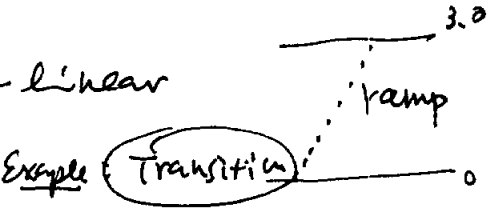
- Delay through a cell depends on:

  Supply voltage, input signal slope, output loading, operating temperature, fabrication process variation.

- power dissipation also depends on the same factors.

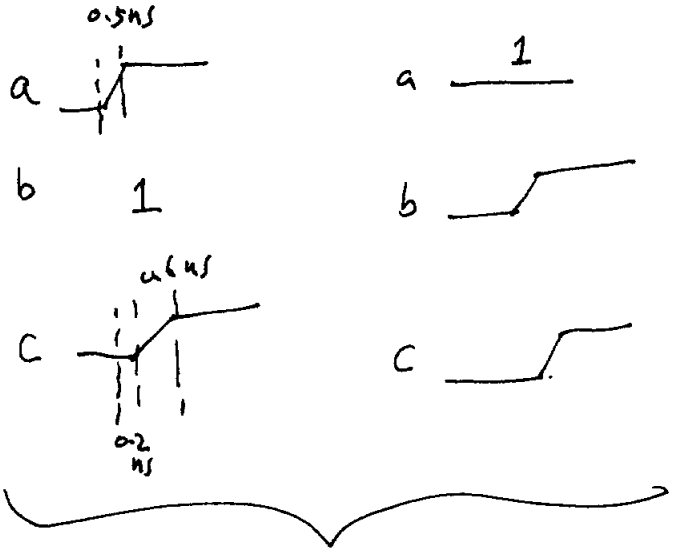
- Support different modeling styles:

  Polynomials, tabular data, piecewise-linear



Example: Transition

Example:



	input waveform			output load
	starting volt	ending volt	rising (falling) time	
range	0V ~ 3.2V	0V ~ 3.2V	0.1 ns ~ 1 ns	100 fF ~ 1600 fF
→	0.6 V	0.6V	0.3 ns	500 fF

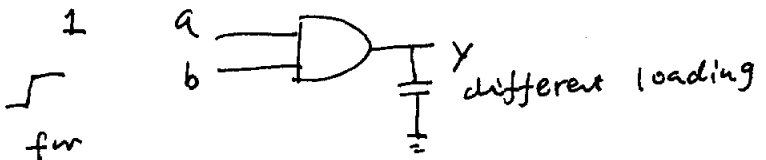
voltage

internal

simulate power & delay.

can be represented by polynomial or table or . . .

- power vectors: discretize all possible power events of a cell.



- Delay vectors — used to calculate output slope which will be used as input slope for next gate.

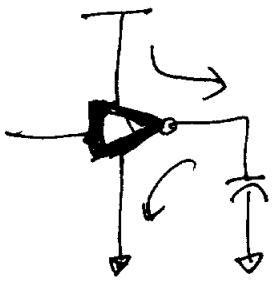
Power Analysis by Aspen

- use verilog XL : * a logic simulator from Cadence
  * can simulate delay for each gate, based on backannotation

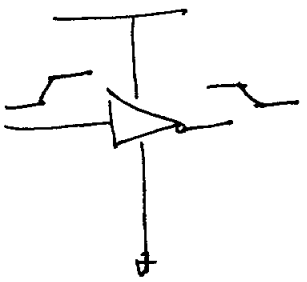
* 3 types of power dissipations are considered:

Dynamic { Capacitive charging/discharging
Short_ckt

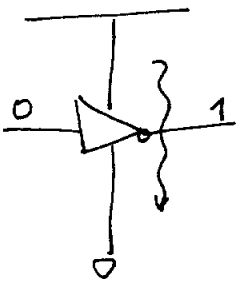
Static { Static leakage power



Capacitive charging



Short-ckt



leakage

Dynamic power vector

static power vector

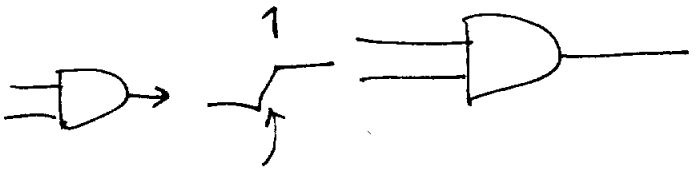
- Capacitive charge & discharge: use dynamic power vector

CV^2f

- output load: routed input capacitance of the load.
- the frequency of charge & discharge for T. (Derived by logic simulation) # of transitions

- Transition power dissipation: (short ckt)

Use dynamic power vectors.

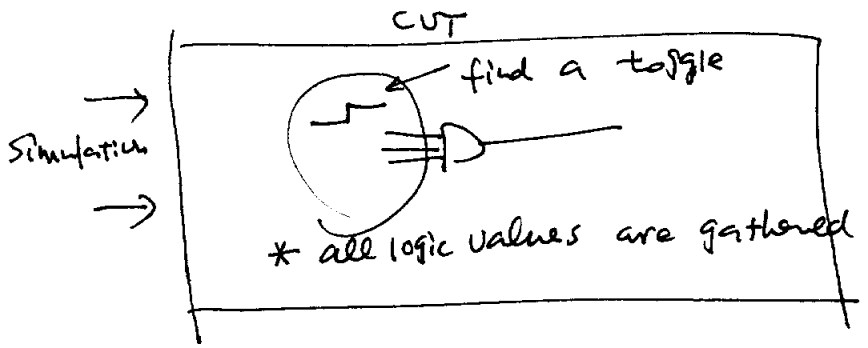


- derived by Verilog XL from previous stage
- check power vector and get the power dissipation
- check delay vector and get the output slope

- Leakage power: depends on ckt state.

check the power vector.
and get the power dissipation.

- When a net toggles, all cell instances incident to the net are examined



* check the power vector

* get the power ~~analysis~~ dissipation from cell library

- Do not consider glitch & hazard
- Accuracy is within 10%, when compared with Spice.

- During simulation, the number of matches between event vectors & power ~~for~~ vectors is calculated
- Dynamic energy consumption:

$$\sum_{\text{for all } i} \left(\#\text{ of power vector } i \text{ matched}\right) \times \text{power consumption for power vector } i$$

↑ include dynamic + short ckt powers

- Static energy consumption: (leakage)

$$\sum_{\text{for all } i} \left(\#\text{ of static power vector matched}\right) \times \text{static power consumption for power vector } i \times \text{activation time}$$