Digital Circuits: why they leak, how to counter

Ingrid Verbauwhede

Ingrid.verbauwhede-at-esat.kuleuven.be

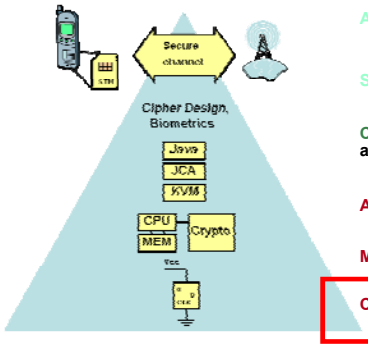
KU Leuven, COSIC

Acknowledgements:
Current and former Ph.D. students

Goal

- Fundamental understanding of CMOS circuits
- So as to build models
- And understand short comings of models
- To understand "Special logic styles and hardware countermeasures," the official title of this lecture.

Design methodology: consider all design abstraction levels



Application: **e-commerce, smart energy**

Security analysis: **TPM, light weight ?**

Crypto Algorithm/Protocol: **crypto, entity authentication**

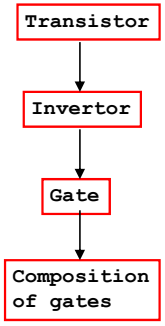
Architecture: **Co-design, HW/SW, SOC**

Micro-Architecture: **co-processor design**

Circuit: **Circuit techniques to combat side channel analysis attacks**

WHY: **1. To get low power/ low energy**
2. To be secure

Outline: bottom-up



- CMOS circuits: operation
- Power consumption – "sources of information leakage"
- Circuit styles and link to "Power models"
- Side effects of gates
- Side channel attack resistance
- Conclusions and reflections

Outline

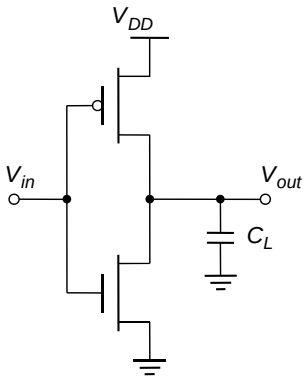
Transistor → Invertor

- CMOS circuits: operation
- Power consumption – “sources of information leakage”
  - Current
  - Dynamic power
  - Static power

CMOS invertor

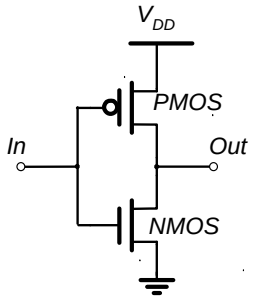
power and energy fundamentals

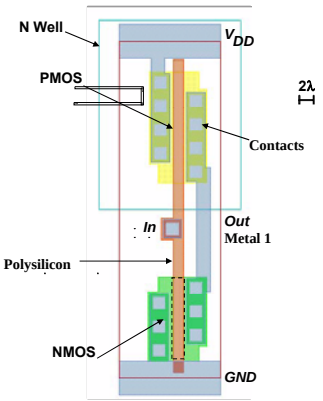
The CMOS Inverter: A First Glance



Slide courtesy: J. Rabaey

CMOS Inverter



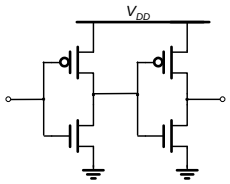


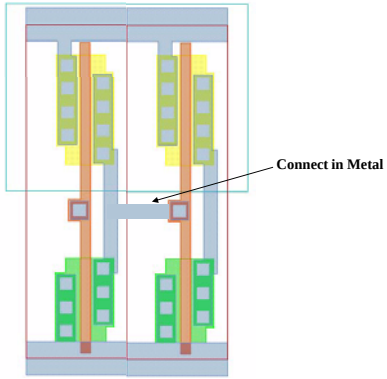
Slide courtesy: J. Rabaey

Two Inverters

Share power and ground

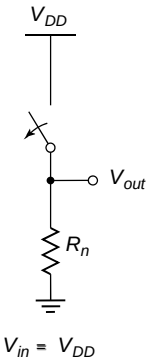
LEGO style: Abut cells

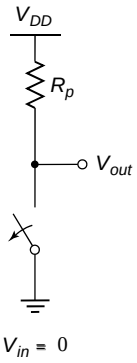




Slide courtesy: J. Rabaey

AC/DC of CMOS Inverter: DC





= STATIC behavior

$V_{OL} = 0$
$V_{OH} = V_{DD}$

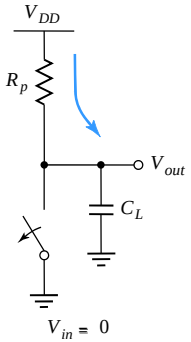
Why we like CMOS!!
- **Full swing**
- **NO DC current!!***

*to first order, see further

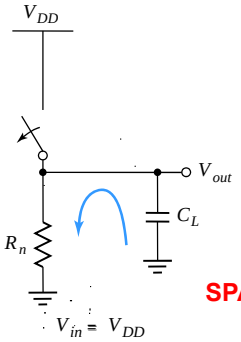
Slide courtesy: J. Rabaey

AC/DC of CMOS Inverter: AC

= DYNAMIC behavior



(a) Low-to-high



(b) High-to-low

$t_{pHL} = f(R_{on}.C_L)$
$= 0.69\ R_{on}C_L$

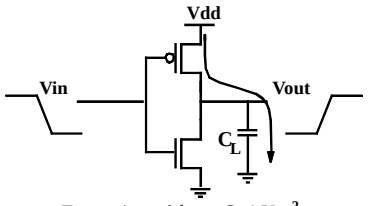
SPA, DPA attack AC!!

Slide courtesy: J. Rabaey

Where Does Power Go in CMOS?

- **Dynamic Power Consumption = AC**
  - **Charging and discharging capacitors**
- [Short Circuit Currents = AC]
  - Short circuit path between supply rails during switching
  - No longer an issue in deep submicron
- **Leakage = DC**
  - **Leaking diodes and transistors**

AC – Dynamic Power consumption



$$\text{Energy/transition} = C_L * V_{dd}^2 * \alpha$$

$$\text{Power} = \text{Energy/transition} * f = C_L * V_{dd}^2 * \alpha * f$$

- **Energy = independent of clock frequency!**
- **Energy = depends on activity α !**
- **Energy, power = independent of transistor sizes**
- **Need to reduce C_L, $V_{dd,}$ α and *f* to reduce power**

AC – SPA, DPA

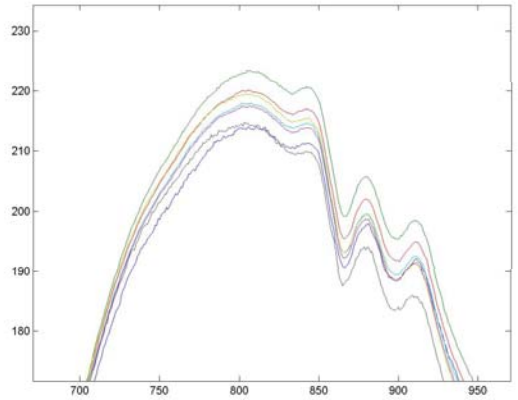
- SPA and DPA monitor power
- Which values depend on data?
  - Monitor α, the activity of circuit
  - Monitor C_L, the capacitance
- Hamming weight:
  - Measures activity between current and (past) known value
  - Typically for pre-charged values
- Hamming Distance:
  - Measures activity between current and previous value
  - Typical for standard cell based design
  - Also for FPGA

Example: power model bus

- 8 bit bus on a smart card, pre-charged
- (relatively) large capacitance
- Hamming weight model
  = numbers of bits set to 1

Side-note: on a pre-charged bus which is pre-set to 1, maximum power consumption is for data all zero.

CPA – Hamming Weight model



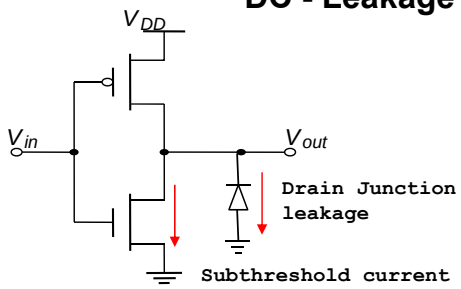
AC - Correlation Power Analysis

- R := reference state
  - Which bit pattern was previously present? E.g.
    - A pre-charged value
    - An opcode on the bus
    - A previously stored value in a register
- Power model: $a \times HW(SBox(x_i \otimes k') \otimes R) + b$
  - a,b are constant, linear model
  - HW is defined as Hamming Weight = counts number of 1's.

DC

leakage currents as
Side-channel information *leakage*

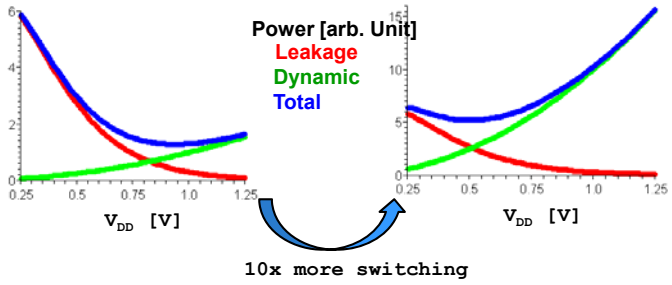
DC - Leakage current



Problem in deep submicron (below 45 nm)
Depends strongly on "threshold voltage" Vt
- Vt is set by processing
- "High Vt" "low Vt" Standard cell library
- Low power FPGAs vs High performance FPGAs

Vt and Vdd effect on leakage

- Vt, Vdd combination for low power, given a target clock frequency

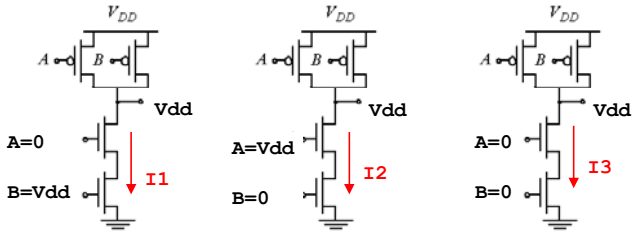


Memory
→ Leakage dominance

High performance microprocessor
→ dynamic power dominance

[slide credit: Wim Dehaene]

DC – leakage of NAND gate



Nand gate: out = '1', but **I1 ≠ I2 ≠ I3**

- New source of information,
- Available even when device is at 'rest'
- Time window to attack larger
- Less a problem for memory because differential structure

If you are looking for a nice research topic.

Outline

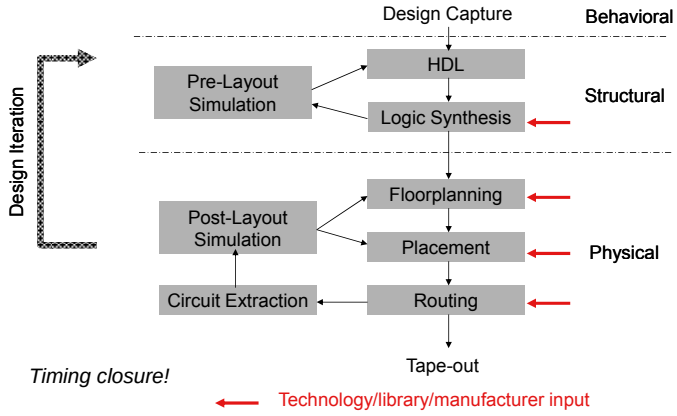
Transistor → Invertor → Gate

- CMOS circuits: operation
- Power consumption – "sources of leakage"
- Circuit styles and link to "Power models"
  - Static CMOS
  - Dynamic, pre-charged CMOS
  - Differential CMOS
  - Dynamic – differential CMOS
  - Link to Hamming Weight – Hamming Distance
- Side effects of gates
- Side channel attack resistance
- Conclusions and reflections

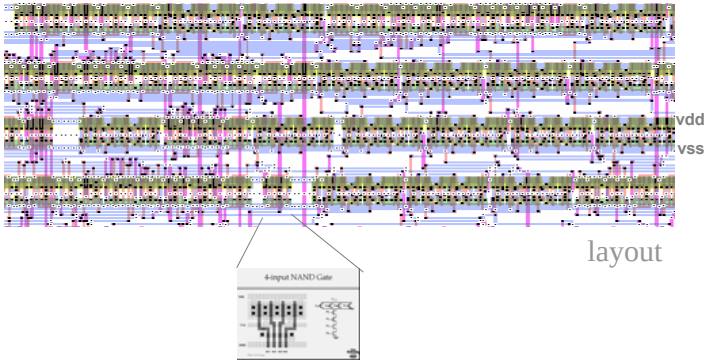
Static CMOS

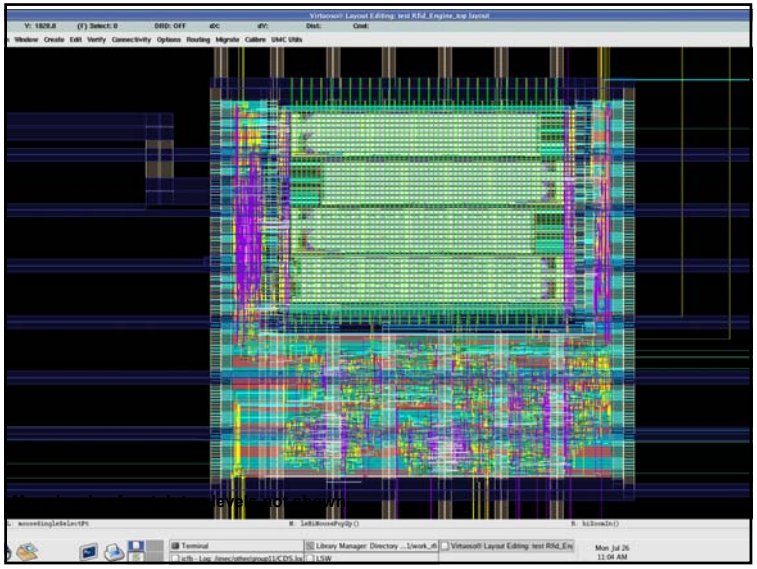
Basics and construction rules

Standard cell automated design flow

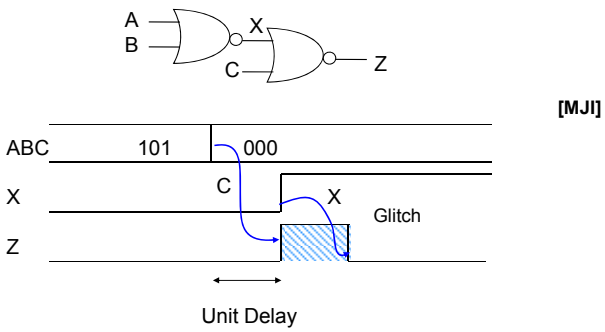


Standard Cell Zoom In





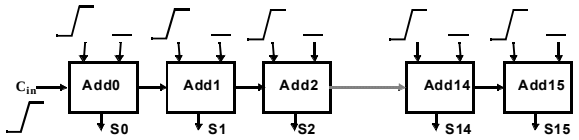
Glitches in static CMOS networks

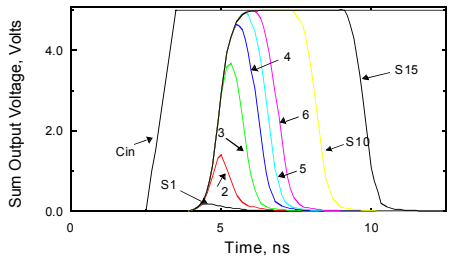


[MJI]

Glitch = Useless transition = Waste of energy
[Low Power community has addressed this]

Most famous example: RippleCarry Adder

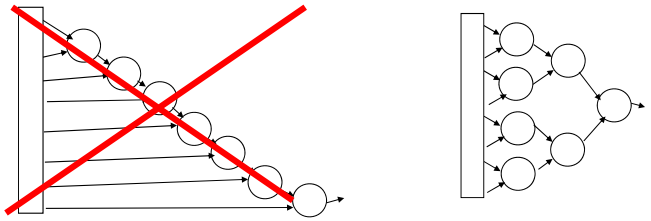




From Rabaey, 1995
Design for low power

Glitch Reduction: Path balancing

- Avoids glitching: general design practice for low power technique
- Principle: transform algorithm into tree like structure
- Then balance delay paths in all paths to output
- Examples:
  - Log adder replaces Ripple Adder
  - Wallace tree replaces Carry-Save multiplier
- Synthesis tools will transform for you “automatically”.



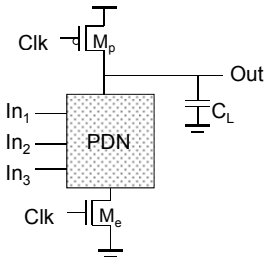
Dynamic CMOS

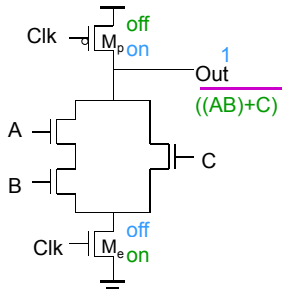
Basics and construction rules

Dynamic CMOS

- In static circuits at every point in time (except when switching) the output is connected to either GND or V_{DD} via a low resistance path.
  - fan-in of *n* requires 2*n* (*n* N-type + *n* P-type) devices
- Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.
  - requires on *n* + 2 (*n*+1 N-type + 1 P-type) transistors

Dynamic Gate





Two phase operation

Precharge (Clk = 0)

Evaluate (Clk = 1)

Conditions on Output

- Once the output of a dynamic gate is discharged, it cannot be charged again until the next precharge operation.
- Inputs to the gate can make at most one transition during evaluation.
- Output can be in the high impedance state during and after evaluation (PDN off), state is stored on C_L

Thus by construction, dynamic gates cannot glitch!

Circuits against side channel attacks

How they leak
How to solve it

Remember AC – SPA, DPA

$$\text{Energy/transition} = C_L * V_{dd}^2 * \alpha$$

$$\text{Power} = \text{Energy/transition} * f = C_L * V_{dd}^2 * \alpha * f$$

- SPA and DPA monitor power
- Address α, C_L
  - Monitor α, the activity of circuit
  - Monitor C_L, the capacitance

Transition independent power consumption …

- …doesn't create any side channel information
- No Hamming distance, No Hamming weight

- When logic values are measured by charging and discharging capacitances, we need to use a fixed amount of energy for every transition

switch a constant load capacitance

C_L = constant

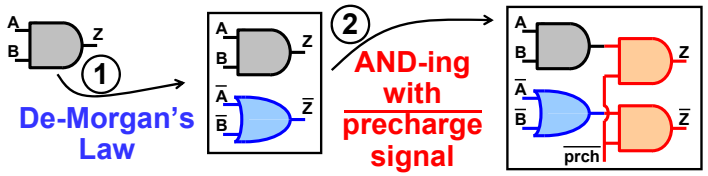
switch once every cycle

α = 1

Dynamic and differential logic

- Dynamic & differential
  - α = 1
  - No glitches
- Differential (with design effort):
  - CL is constant
  - Also includes differential routing
  - Static Leakage current is data independent

Solution based on *Standard* cells

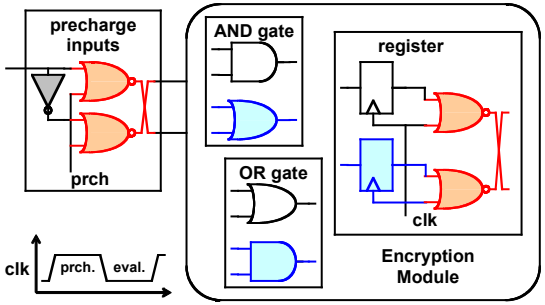


- false output
- with false inputs

- precharge 1: outputs are 0
- precharge 0 - evaluation: 1 output is 1

Wave Dynamic Differential Logic

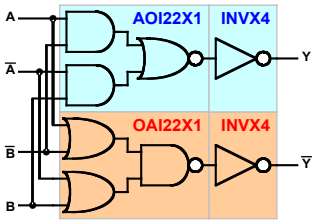
- 0-wave travels from input to output during pre-charge
  - input 0 ⇒ output 0
  - no pre-charge operator
- Differential data travels during evaluation

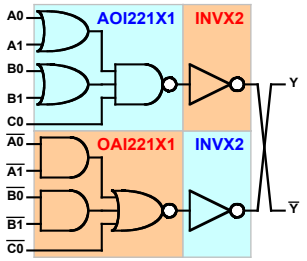


[Tiri,DATE2004]

WDDL library

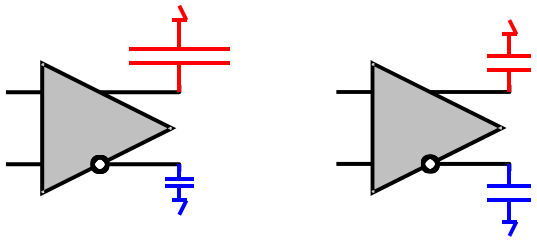
- All functions of and2, or2 operator
- In addition: inverted input, output signals
- XOR2X4: OAI221X2:
- Our WDDL library: 128 cells



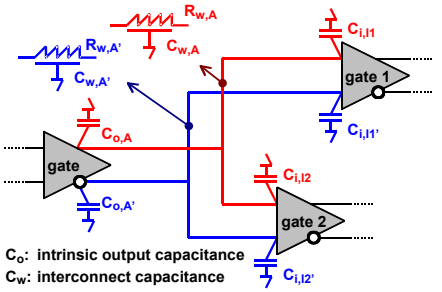


Unbalanced capacitive loads

- For constant power consumption: constant load capacitance.
- Match loads at differential outputs.



Load capacitance breakdown

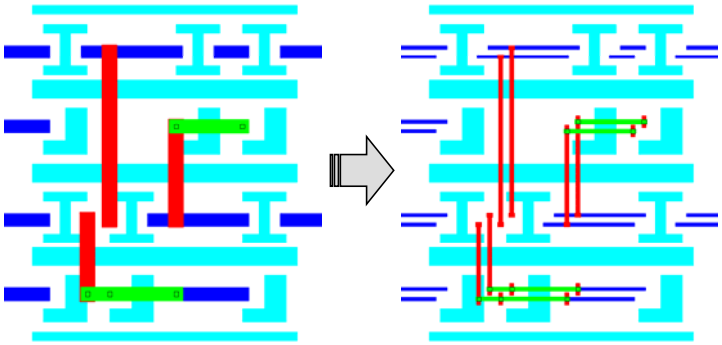


C_o: intrinsic output capacitance
C_w: interconnect capacitance
C_i: input capacitance

- Intrinsic caps.: matched
- Interconnect: dominant (Moore's law)
- Balancing interconnect: crucial

$$C_{A} = C_{A'}$$
$$C_{o,A} + C_{w,A} + C_{i,I1} + \ldots C_{i,Ik} = C_{o,A'} + C_{w,A'} + C_{i,I1'} + \ldots C_{i,Ik'}$$
$$C_{w,A} = C_{w,A'}$$

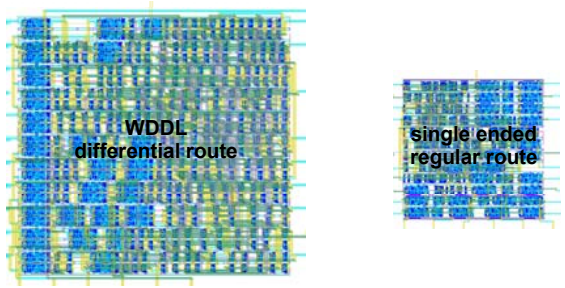
Design example



- Two normal wires replace each fat wire.

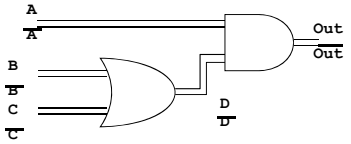
WDDL Example

- Same circuit; two implementations.
  - Insecure reference design
  - Secure design



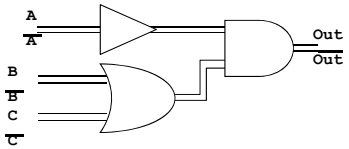
Early propagation effect

- Static CMOS, dynamic CMOS, differential CMOS,
- Timing of transition is data dependent



	A	B	C	D	O	T1	T2
Pre	0	0	0	0	0		
Eval 1	0	X	X	X @T1	0 @T1	2	0
Eval 2	1	0	0	0 @T1	0 @T2	1	1
Eval 3	1	0	1	1 @T1	1 @T2	1	1

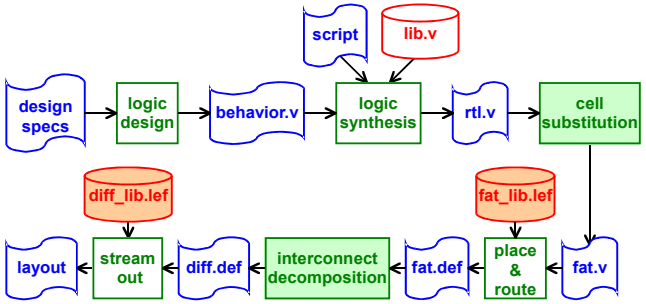
Early propagation effect: balance



	A	B	C	D	O	T1	T2
Pre	0	0	0	0	0		
Eval 1	0	X	X	X @T1	0 @T2	1	1
Eval 2	1	0	0	0 @T1	0 @T2	1	1
Eval 3	1	0	1	1 @T1	1 @T2	1	1

Can prove that it is always possible to balance in WDDL logic.

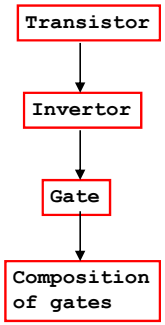
Integration in standard cell design flow: Secure digital design



Few key modifications with minimal influence in backend of regular synchronous static CMOS standard cell design flow

[Tiri,TCAD2006]

Conclusions and reflections



- Fundamental understanding CMOS circuits
  - AC DC behavior
  - Static CMOS: low power, but shows Hamming distance
  - Dynamic CMOS: high speed, no glitches, but shows Hamming weight
  - Dynamic, differential: hides data dependencies
  - Full custom style: SABL
  - Standard cell compatible: WDDL (with construction rules)
- Side effect of CMOS gates:
  - Glitch: only problem of static CMOS
  - Memory effect: static CMOS
  - Early propagation: can be addressed in WDDL
- Future: address DC leakage current
  - Leakage even when there is no operation