Power, Energy, and Dependability in Architecture

Jaynaryan T Tudu

Indian Institute of Technology Tirupati, India

9th Feb, 2021
CS5202-Computer System Architecture

Computer Architecture: The Challenges

The objective: to improve performance

The challenges:

- Power or Energy
- Dependability
- Security
- Area
- Cost

40 Years of Processor Trend

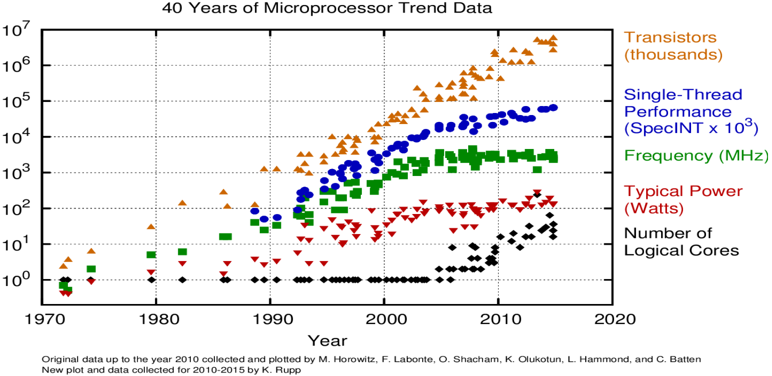


Figure : Why the MHz (clock frequency) is flattening?

Observation from Dark-silicon

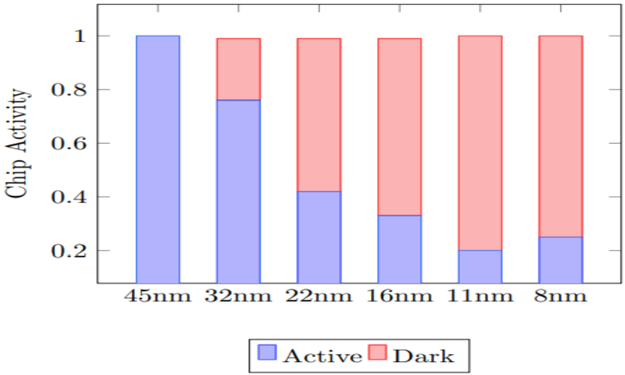


Figure : We are not able to use all the available on-chip resources at at time!

References: [1]

[1] A Perspective on Dark Silicon; Anil Kanduri, Amir M. Rahmani, Pasi Liljeberg, Ahmed Hemani, Axel Jantsch, and Hannu Tenhunen

The Basics of Power Consumption/Dissipation

Power:
$$P = VI$$

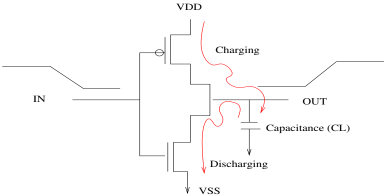


Figure : Dynamic power

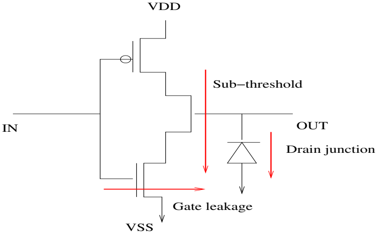


Figure : Static (Leakage power)

$$P = \alpha C V^2 f$$

Computing the Power Dissipation

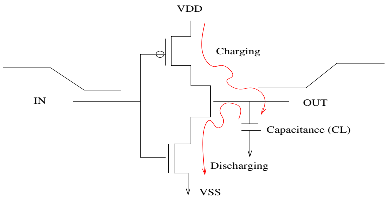


Figure : An inverter with pmos and nmos

Two things to observe:

- The energy at V_{dd} supply node (E_{vdd})
- The energy at V_{out} output node after activity (E_c)

Computing the Power Dissipation

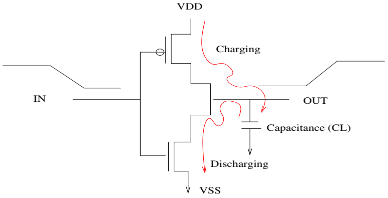


Figure : An inverter with pmos and nmos

There are two kinds of activities here: charging and discharging.

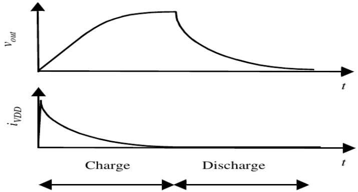


Figure : *iv* behavior of charging and discharging

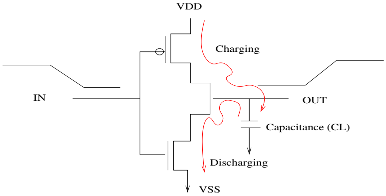


Figure : An inverter with pmos and nmos

$$P = IV$$
$$E_{vdd} = \int_0^\infty i_{vdd}(t) V_{dd}\, dt = V_{dd} \int_0^\infty i_{vdd}(t)\, dt$$
$$= V_{dd} \int_0^\infty C_l \frac{d\, v_{out}}{dt}\, dt = C_l\, V_{dd} \int_0^\infty d\, v_{out} = C_l V_{dd} V_{dd}$$

Computing the Energy Consumption/Dissipation

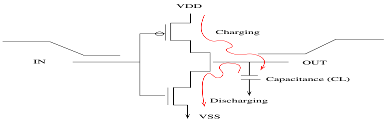


Figure : An inverter with pmos and nmos

$$E_{out} = \int_0^{\infty} i_{vdd}(t) v_{out}(t)\, dt = \int_0^{\infty} C_l \frac{d\, v_{out}}{dt} v_{out}(t) dt$$

$$= C_l \int_0^{\infty} v_{out}(t) d\, v_{out} = \frac{C_l V_{dd}^2}{2}$$

Observation: (Energy dissipation and consumption

$E_{vdd} = C_l\, V_{dd}^2$

$E_{out} = \frac{C_l\, Vdd^2}{2}$

Note: Each switching cycle takes a fixed amount of energy ie $Cl\, V_{dd}^2$

(Principle of energy conservation and work done!)

Computing the Power Dissipation

Switching cycle: a complete event of charging $(0 \rightarrow 1)$and discharging $(1 \rightarrow 0)$

The layman understanding: power = the rate at which energy is being transformed!

$P = Energy/t$

if $t = 1$ second and the total switching is f then the power can be expressed as:
$P_{dyn} = C_l V_{dd}^2 f$ (P here can be referred to as dynamic power or switching power P_{dyn})
In reality, there are many gates who does not complete the switching cycle, therefore an activity factor α is considered.
$P_{dyn} = \alpha C_l V_{dd}^2 f$

Computing the Power Dissipation

Therefore, total power could be expressed as:

$$P_{total} = P_{dyn} + P_{static}$$

We have calculated P_{dyn}!
$P_{dyn} = C_l \, V_{dd}^2 \, f$

The P_{static} can be computed similarly as:
$P_{static} = i_{static} \, V_{dd}$

How to reduce Energy or Power

$P_{dyn} = C_l V_{dd}^2 f$

1. What about reducing C_l ?
2. What about reducing V_{dd} ?
3. What about reducing f, the frequency of switching
4. What about reducing the activity itself? (Architects and Software people)

Most used techniques

- Gating (just turn-off which are not necessary!
- DVFS: dynamic voltage and frequency scaling

Concept of Dependability

A system or component can be called as dependable when it delivers the specified functionality when needed.

Why would a system (the one build with CMOS IC) fail?

- failure due to permanent faults
- failure due to transient faults

The trend is: as the device size reduce the failure increase at exponential rate!

Concept of Dependability

How to measure the dependability of a system?

- State 1: Service accomplished
- State 2: Service interruption

Failure: [service accomplished] $\rightarrow$ [service interruption]
Restoration: [Service interruption] $\rightarrow$ [service accomplished]

Concept of Dependability

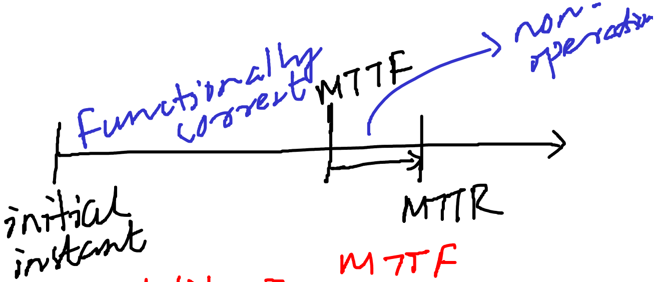
Measuring or Quantifying the dependability.

- Module reliability: continuous service accomplishment or time to failure (during operational)
  - to quantify: mean time to failure (MTTF), given that the time to failure is random variable.
  - The reciprocal quantity: failure in given time. Example: failure per t hrs of operation.
  - What happens after failure! Need to restore or repair. So, this can be quantified with mean time to repair (MTTR).
  - Therefor: mean time between failure = MTTF + MTTR
- Module availability: How do we quantify?

Concept of Dependability

Module availability: How do we quantify?

Definition: measure of service accomplishment with respect to alteration between two states



$$\text{availability} = \frac{MTTF}{MTTF + MTTR}$$

Summary

Tried to put them in the order of importance:

- Power
- Dependability
- Security
- Area
- Cost

Reading Meterials

- John L Henessy and D Patternson, Computer Architecture: A Quantitative Approach, 5th Edition, pp. 36-58 (Chapter 1).
- Chapter 4 (Section: Power Dissipation), CMOS VLSI Design-A Ckt and Sys Perspective; N Weste, D Harris, A Banerjee.
- Addressing Failure in Exascale Computing; Marc Snir, Robert W. Wisniewski, Jacob A. Abraham, Sarita V. Adve, Saurabh Bachi, et al; Intl Journal of High Performance Computing Applications 28(2), 2014.