NUMERO: 1675A - ANNO: 2015

A P P U N T I

STUDENTE: Fissore

MATERIA: Sistemi Elettronici a Basso Consumo LowPower, Prof.Zamboni

Low-Power Electronics System

Prof. Maurizio Zamboni

maurizio.zamboni@polito.it
phone number +39 11 090 4079
Electronics Department
Politecnico di Torino

SEBC-L1 MZ 1

Appunti di Giorgio Fissore
Disponibili in centro stampa

Low-Power Electronics System

Timetable

SEBC-L1 MZ 2

7 volte in lab:
10 marzo
17 marzo
14 aprile
28 aprile
12 maggio
26 maggio
9 giugno?

Low-Power System Design

FINAL EXAM

Discussion on course topics

Lab sessions (1/3 of final grade)

Optional Final Project instead of Lab Reports (2/3 of final grade + bonus) (suggested only if really motivated) to reach 30LODE/30

SEBC-L1 MZ 3

Esame solo orale, con possibilità di sostenerlo praticamente tutte le settimane concordato con il docente

Le relazioni di laboratorio possono essere consegnate anche dopo aver dato gli esami.

Low-Power System Design

COURSE MATERIAL:

Low-Power Design Essentials - J. Rabaey - Springer, 2009

Low-power CMOS VLSI Circuit Design – K. Roy, S. C. Prasad **J. WILEY & SONS**

Workbook as a collection of slides coming from different sources (available also on "Portale della didattica")

Slides partially adapted from Low-Power Design Essentials, Springer 2009. © J. Rabaey.

Series of papers also available on "Portale della didattica"

SEBC-L1 MZ 7

Il libro che potrebbe essere più interessante.

Ci sono in più appunti presi qualche anno fa

SEBC-L1 MZ 8

Lecture 1
Introduction to Low-Power Design

- Motivation
- Historical Drivers of Low-Power Design
- Microprocessor Scaling
- Power Sources
- Low-Power Design Methods

SEBC-L1 MZ 9

Mod by Giorgio Fissore, pag 3

L1-Introduction.key - 22 Feb 2015

Motivation for Low-Power Design

International Technology Roadmap for Semiconductors

Year of Introduction	1999	2000	2001	2004	2008	2011	2014
Technology node [nm]	180		130	90	60	40	*30*
Supply [V]	1.5-1.8	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	*0.3-0.6*
Wiring levels	6-7	6-7	7	8	9	9-10	*10*
Max frequency [GHz].Local-Global	1.2	1.6-1.4	2.1-1.6	3.5-2	7.1-2.5	11-3	*14.9 -3.6*
Max µP power [W]	90	106	130	160	171	177	*186*
Bat. power [W]	1.4	1.7	2.0	2.4	2.1	2.3	*2.5*

Node years: 2007/65nm, 2010/45nm, 2013/33nm, 2016/23nm

SEBC-L1 MZ 13

Historical Drivers of Low-Power Design

- Pocket calculators
- Hearing aids
- Implantable pacemakers and cardiac defibrilators
- Portable military equipment for individual soldiers
- Wristwatches
- Wireless computing ..... BUT NOT ONLY!!!!

SEBC-L1 MZ 14

Historical Drivers of Low-Power Design

- 20% of electrical energy in Amsterdam due to telecom!!!
- In US 9% of energy consumption due to Internet!!!
- 2 Mbytes transfer through the net consumes the energy of 1 pound of coal 1 libra di carbone

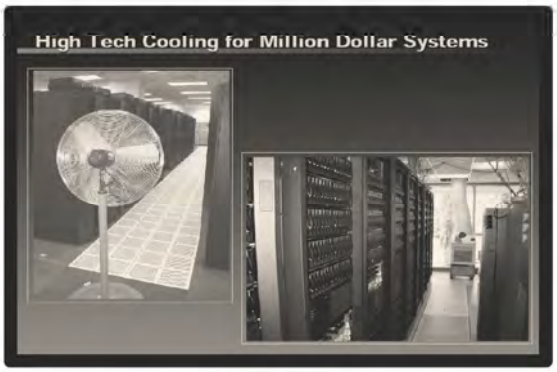
SEBC-L1 MZ 15

Mod by Giorgio Fissore, pag 5

L1-Introduction.key - 22 Feb 2015

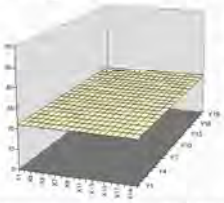
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 7 di 402

Power the Dominant Design Constraint



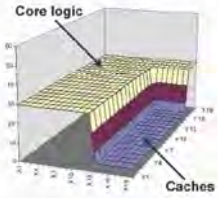
SEBC-L1 [Ref: R. Schmidt, ACEED'03] MZ 19

Chip Architecture and Power Density



Integration of diverse functionality on SoC causes major variations in activity (and hence power density)

Today: steep gradients



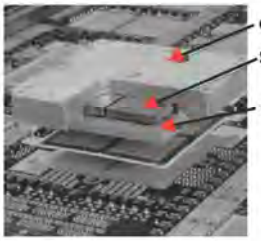
The past: temperature uniformity

Temperature variations cause performance degradation – higher temperature means slower clock speed

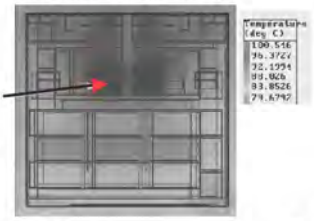
SEBC-L1 [Ref: R. Yung, ESSCIRC'02] MZ 20

La tecnologia costruita "più in 3D" genera gradienti di temperatura all'interno del dispositivo.
(si sceglie la distribuzione dei vari componenti all'interno del chip, anche in base a questi gradienti - es no due parti vicine a 30° e 100° per non ridurre la vita dello stesso)

Temperature Gradients (and Performance)



IBM Power PC 4 temperature map



Hot spot:
138 W/cm²
(3.6 x chip avg flux)

SEBC-L1 [Ref: R. Schmidt, ACEED'03] MZ 21

Mod by Giorgio Fissore, pag 7

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 9 di 402

Battery Storage a Limiting Factor

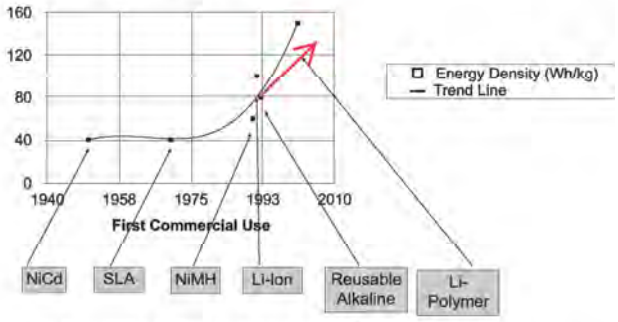
- Basic technology has evolved little
  - store energy using a chemical reaction
- Battery capacity increases between 3% and 7 % per year (doubled during the 90's, relatively flat before that)
- Energy density/size, safe handling are limiting factor

Energy density of material	kWh/kg
Gasoline	14
Lead-Acid	0.04
Li polymer	0.15

For extensive information on energy density of various materials, check http://en.wikipedia.org/wiki/Energy_density

SEBC-L1 MZ 25

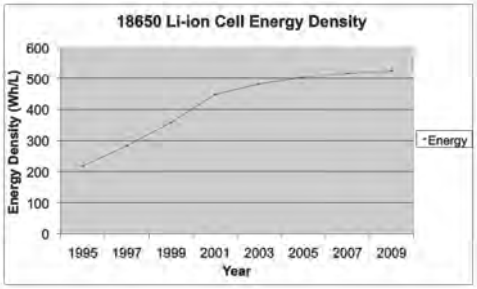
Battery Evolution



Accelerated since the 1990's, but slower than IC power growth.

SEBC-L1 MZ 26

Battery Technology Saturating



Battery capacity naturally plateaus as systems develop

SEBC-L1 [Courtesy: M. Doyle, Dupont] MZ 27

Mod by Giorgio Fissore, pag 9

L1-Introduction.key - 22 Feb 2015

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 11 di 402

How much Energy Storage in 1 cm³?

	J/cm³	μW/cm³/year
Micro Fuel cell	3500	110
Primary battery	2880	90
Secondary battery	1080	34
Ultracapacitor	100	3.2

SEBC-L1 MZ 31

Power the Dominant Design Constraint (3)

Exciting emerging applications require "zero-power"

Example: Computation/Communication Nodes for Wireless Sensor Networks

Meso-scale low-cost wireless transceivers for ubiquitous wireless data acquisition that

- are fully integrated
  - Size smaller than 1 cm³
- are cheap
  - At or below 1$
- minimize power/energy dissipation
  - Limiting power dissipation to 100 μW enables **energy scavenging**
- and form self-configuring, robust, ad-hoc networks containing 100's to 1000's of nodes

SEBC-L1 [Ref: J. Rabaey, ISSCC'01] MZ 32

Applicazioni a "zero power", applicazioni a consumo praticamente nullo che potrebbero non richiedere una batteria.
Ciò vuol dire trovare il modo di alimentare questi sistemi direttamente dall'ambiente.
Vedi slide 35 per esempi

How to Make Electronics Truly Disappear?

From 10's of cm³ and 10's to 100's of mW

To 10's of mm³ and 10's of μW

SEBC-L1 MZ 33

Mod by Giorgio Fissore, pag 11

L1-Introduction.key - 22 Feb 2015

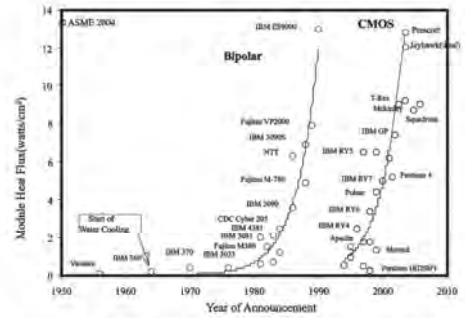
Power versus Energy

- Power in high performance systems
  - Heat removal
  - Peak power - power delivery
- Energy in portable systems
  - Battery life
- Energy/power in "zero-power systems"
  - Energy-scavenging and storage capabilites
- Dynamic (energy) vs. static (power) consumption
  - Determined by operation modes

SEBC-L1 MZ 37

I parametri di potenza ed energia sono importanti per motivi diversi:
La potenza è legata principalmente alla capacità di portare via calore dal mio dispositivo.
La potenza di picco è invece legata al fatto che questa causa un'alta corrente di picco, che, se il filo ha una resistenza anche piccola, genera alte tensioni che possono disturbare; in più, se non abbastanza spesso, il filo rischia di rompersi.
L'energia ci interessa invece in termini di tempo di vita della batteria.

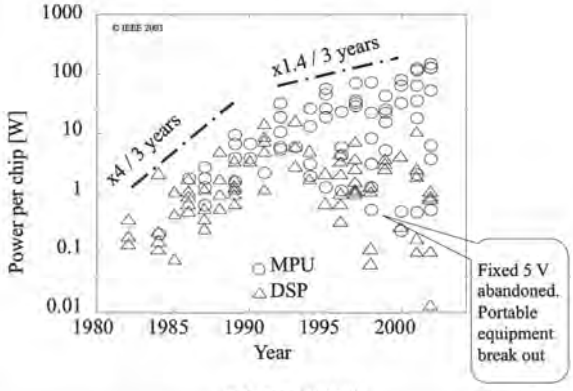
Power Evolution over Technology Generations



Introduction of CMOS over bipolar bought industry 10 years (example: IBM mainframe processors)

SEBC-L1 [Ref: R. Chu, JEP'04] MZ 38

Power Trends for Processors



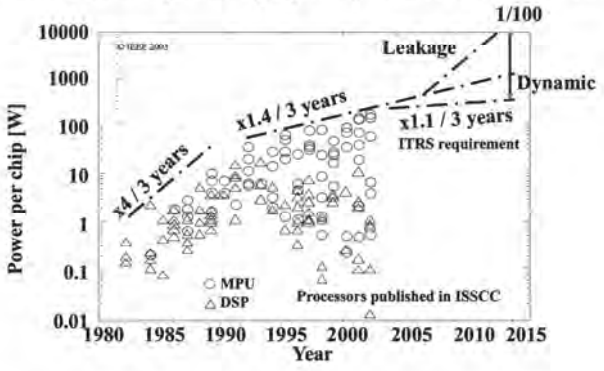
SEBC-L1 [Ref: T. Sakurai, ISSCC'03] MZ 39

Mod by Giorgio Fissore, pag 13

L1-Introduction.key - 22 Feb 2015

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 15 di 402

Static Power (Leakage) may Ruin Moore's Law

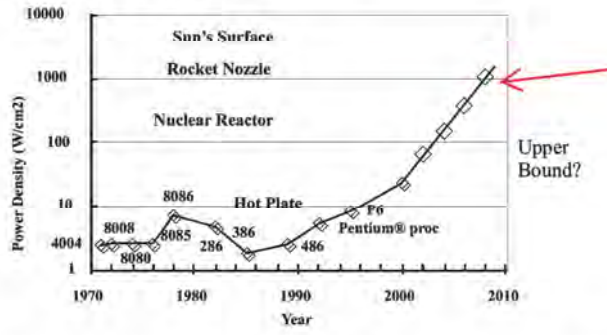


[Ref: T. Sakurai, ISSCC 03]

SEBC-L1 MZ 43

Power Density Increases

Unsustainable in the long term

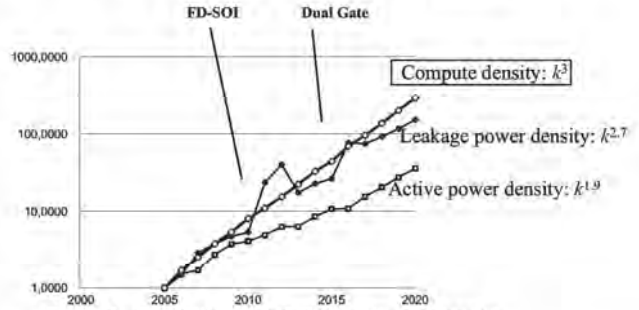


[Courtesy: S. Borkar, Intel]

SEBC-L1 MZ 44

Se non avessimo ridotto i consumi, un processore utilizzerebbe la potenza di un razzo in accensione

Projecting Into the Future



Power density (active and static) accelerating again.

Technology innovations help, but impact limited.

SEBC-L1 MZ 45

Mod by Giorgio Fissore, pag 15

L1-Introduction.key - 22 Feb 2015

A 20 nm Scenario

Assume V_{DD} = 1.2V

- delay < 5 ps
- Assuming no architectural changes, digital circuits could be run at 30 GHz FO4
- Leading to power density of 20 kW/cm² (??)

Reduce V_{DD} to 0.6V

- FO4 delay = 10 ps
- The clock frequency is lowered to 10 GHz
- Power density reduces to 5 kW/cm² (still way too high)

SEBC-L1 [Ref: S. Borkar, Intel] MZ 49

A 20 nm Scenario (cntd)

Assume optimistically that we can design FETs (Dual-Gate, FinFet, or whatever) that operate at 1 kW/cm² for FO4 = 10 ps and V_{DD} = 0.6 V [Frank, Proc. IEEE, 3/01]

- For a 2cm x 2cm high-performance microprocessor die, this means 4kW power dissipation.
- If die power has to be limited to 200W, only 5% of these devices can switching at any time, assuming that nothing else dissipates power.

SEBC-L1 [Ref: S. Borkar, Intel] MZ 50

An Era of Power-Limited Technology Scaling

- **Technology innovations offer some relief**
  - Devices that perform better at low voltage without leaking too much
- **But also are adding major grieve**
  - Impact of increasing process variations and various failure mechanisms more pronounced in low-power design regime.
- **Most plausible scenario**
  - Circuit and system level solutions essential to keep power/energy dissipation in check
  - Slow down growth in computational density, and use obtained slack to control power density increase.
  - Introduce design techniques to operate circuit at nominal, not worst-case, conditions

SEBC-L1 MZ 51

Mod by Giorgio Fissore, pag 17

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 19 di 402

Where is Power Dissipated in CMOS?

- Active (Dynamic) power
  - (Dis)charging capacitors
  - Short-circuit power
    - Both pull-up and pull-down on during transition
- Static (leakage) power
  - Transistors are imperfect switches
- Static currents
  - Biasing currents

Quando il circuito necessita di una CC continua. Contributo che si cerca di evitare, ma non sempre si riesce.

Esempio: sense amplifier usati nelle flash, hanno bisogno di essere alimentati in continua

SEBC-L2

Active (or Dynamic) Power

Key property of active power:

$$P_{dyn} \propto f$$

with f the switching frequency

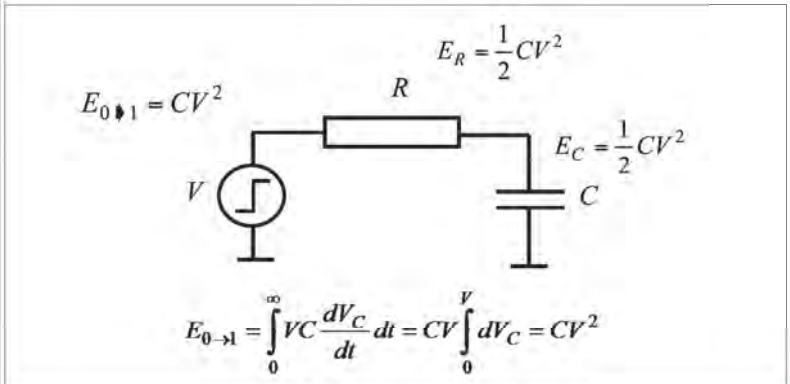
Sources:

- Charging and discharging capacitors
- Temporary glitches (dynamic hazards)
- Short-circuit currents

SEBC-L2 MZ 5

Charging Capacitors

Applying a voltage step



Value of R does not impact energy!

Mod by Giorgio Fissore, pag 19

SEBC-L2 MZ 6

Charging Capacitors

Using constant voltage or current driver?

$$E_{constant\_current} < E_{constant\_voltage}$$
if
$$T > 2RC$$

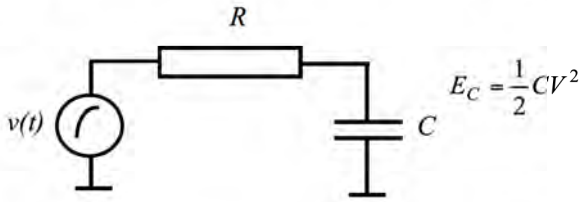
Energy dissipated using constant current charging can be made arbitrarily small at the expense of delay:
Adiabatic charging

Note: $t_p(RC) = 0.69\ RC$
$t_{0\to 90\%}(RC) = 2.3\ RC$

SEBC-L2 MZ 10

Charging Capacitors

Driving using a sine wave (e.g. from resonant circuit)



Energy dissipated in resistor can be made arbitrarily small if frequency $\omega << 1/RC$
(output signal in phase with input sinusoid)

SEBC-L2 MZ 11

Dynamic Power Consumption

Power = Energy/transition • Transition rate

$$= C_L V_{DD}^2 \bullet f_{0\to 1}$$

$$= C_L V_{DD}^2 \bullet f \bullet P_{0\to 1}$$

$$= C_{switched} V_{DD}^2 \bullet f$$

- Power dissipation is data dependent – depends on the switching probability
- Switched capacitance $C_{switched} = P_{0\to 1} C_L = $ Esw C_L
  (Esw is called the switching activity)

SEBC-L2 MZ 12

Questo transition rate può essere espresso come prodotto tra (sistemi sincroni):
la frequenza ed il numero di volte che il nostro disp cambia stato (switching activity)
Switching activity = (cambi di stato del disp)/(colpi di CLK)

Mc

FORMULA DA RICORDARE:
.
$P=Cl*Vdd^2*f*Esw$

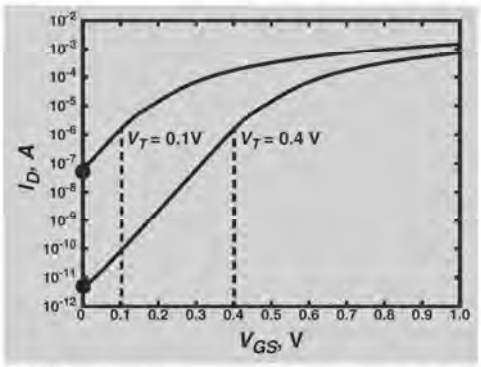
Transistors Leak

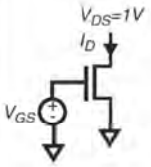
- Drain leakage
  - Diffusion currents
  - Drain-induced barrier lowering (DIBL)
- Junction leakages
  - Gate-induced drain leakage (GIDL)
- Gate leakage
  - Tunneling currents through thin oxide

SEBC-L2 MZ 16

Transistori piccoli >> cattivi interruttori, poichè differenze tra interdizione e zona attiva troppo piccole fanno si che il transistor non si spenga mai del tutto.

Sub-threshold Leakage





Off-current increases exponentially when reducing V_{TH}

$$I_{leak} = I_0 \frac{W}{W_0} 10^{\frac{-V_{TH}}{S}} \Rightarrow P_{leak} = V_{DD} I_{leak}$$

SEBC-L2 MZ 17

Se, per qualsiasi motivo, il dispositivo riduce la sua tensione di soglia, la sua transcaratteristica (I/V) si sposta a sinistra (vedi immagine). E ciò porta ad aumentare la corrente di parecchio (quattro ordini di grandezza); il legame tra la off-current (corrente a transistor spento) e la Vt è infatti esponenziale!!

Off-current aumenta exp con Vt

Sub-Threshold Leakage

Leakage current increases with drain voltage (mostly due to DIBL)

$$I_{leak} = I_0 \frac{W}{W_0} 10^{\frac{-V_{TH}+\lambda_d V_{DS}}{S}} \quad (\text{for } V_{DS} > 3\,kT/q)$$

Hence

$$P_{leak} = \left(I_0 \frac{W}{W_0} 10^{\frac{-V_{TH}}{S}}\right)\left(V_{DD} 10^{\frac{\lambda_d V_{DD}}{S}}\right)$$

Leakage Power strong function of supply voltage

SEBC-L2 MZ 18

Si può inoltre dimostrare che la I_leak aumenta sempre exp con la Vds.

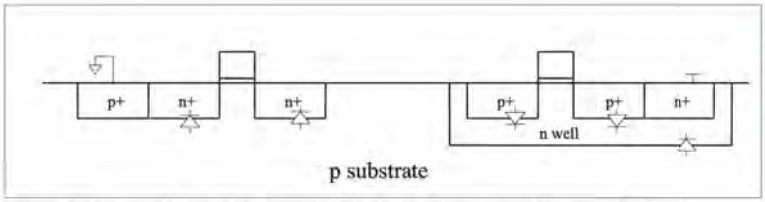
Dobbiamo quindi cercare di avere trans con la minima Vds possibile

Mod by Giorgio Fiss

Mettendo due trans in serie (apparentemente assurdo) dimezzo la Vds su di loro, e quindi abbasso I_leak non di un mezzo, ma di un esponenziale a base 10

Other sources of static power dissipation

- Diode (drain-substrate) reverse bias currents



- Electron-hole pair generation in depletion region of reverse-biased diodes
- Diffusion of minority carriers through junction
- For sub-50nm technologies with highly-doped pn junctions, **tunneling through narrow depletion region** becomes an issue

Strong function of temperature

Much smaller than other leakage components in general

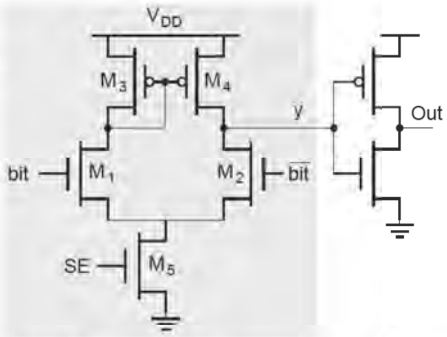
SEBC-L2 MZ 22

Il consumo statico vero e proprio è invece dovuto alla presenza di elementi, come il sense amplifier, regolatori di tensione,.. che necessitano di una corrente continua per funzionare. (è ad esempio un problema nelle memorie)

Other sources of static power dissipation

- Circuit with dc bias currents:

sense amplifiers, voltage converters and regulators, sensors, mixed-signal components, etc



Should be turned off if not used, or standby current should be minimized

SEBC-L2 MZ 23

Summary of Power Dissipation Sources

$$P \sim \alpha \times (C_L + C_{CS}) \times V_{swing} \times V_{DD} \times f + (I_{DC} + I_{Leak}) \times V_{DD}$$

- α – switching activity
- C_L – load capacitance
- C_{CS} – short-circuit capacitance
- V_{swing} – voltage swing
- f – frequency
- I_{DC} – static current
- I_{leak} – leakage current

$$P = \frac{energy}{operation} \times rate + static\ power$$

SEBC-L2 MZ 24

Potenza statica

Potenza dinamica con :
-alfa = switching activity
- Cl + Cs (somma delle capacità della linea e di sc)

Mod by Giorgio Fissore, pag 25

Model not Appropriate Any Longer

Traditional scaling model

If $V_{DD} = 0.7$, and Freq $= (\frac{1}{0.7})$,

Power $= CV_{DD}^2 f = (\frac{1}{0.7} \times 1.14^2) \times (0.7^2) \times (\frac{1}{0.7}) = 1.3$

Maintaining the frequency scaling model

If $V_{DD} = 0.7$, and Freq $= 2$,

Power $= CV_{DD}^2 f = (\frac{1}{0.7} \times 1.14^2) \times (0.7^2) \times (2) = 1.8$

While slowing down voltage scaling

If $V_{DD} = 0.85$, and Freq $= 2$,

Power $= CV_{DD}^2 f = (\frac{1}{0.7} \times 1.14^2) \times (0.85^2) \times (2) = 2.7$

SEBC-L2 MZ 28

The New Design Philosophy

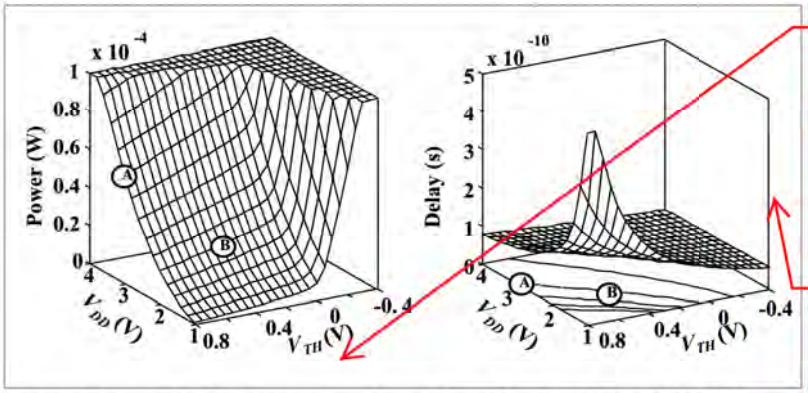
- Maximum performance (in terms of propagation delay) is too power-hungry, and/or not even practically achievable
- Many (if not most) applications either can tolerate larger latency, or can live with lower than maximum clock-speeds
- Excess performance (as offered by technology) to be used for energy/power reduction

Trading off speed for power

SEBC-L2 MZ 29

Un progetto sarà tanto più buono, quanto più riesco a mediare tra questi due parametri.

Relationship Between Power and Delay



For a given activity level, power is reduced while delay is unchanged if both V_{DD} and V_{TH} are lowered such as from A to B.

[Ref: T. Sakurai and T. Kuroda, numerous references]

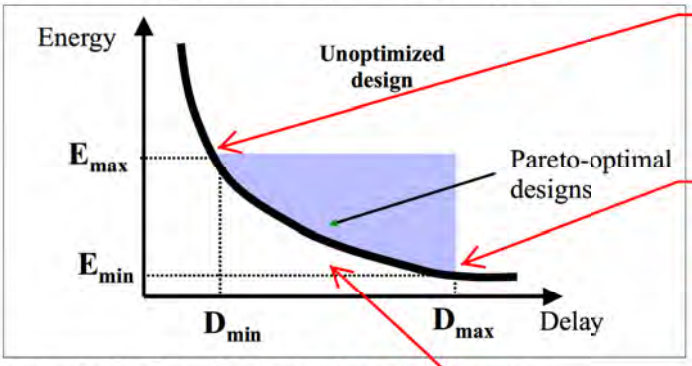
SEBC-L2 MZ 30

-Aumentare la tensione di alimentazione fa crescere quadraticamente il consumo.
-Aumentare la tensione di soglia fa crescere esponenzialmente il consumo

Il ritardo invece dipende da quanto sono distanti la Vt e la Vdd.
I punti A e B qui si trovano in punti di uguale ritardo; se però li guardiamo sul piano del consumo, si vede come tra i due punti ci sia una grande differenza.
Ho quindi bisogno di tecnologie che mi permettano di scegliere la Vt per ottimizzare il legame tra Vt e Vdd

Mod by Giorgio Fissore, pag 27

Exploring the Energy-Delay Space



In energy-constrained world, design is trade-off process

- Minimize energy for a given performance requirement
- Maximize performance for given energy budget

SEBC-L2 [Ref: D. Markovic, JSSC'04] MZ 34

Punto "lavora alla max frequenza con questa energia disponibile"

Punto "lavora consumando la minima energia avendo questo come massimo ritardo"

Questa curva limite (iperbole) è il minimo di energia-ritardo e rappresenta il punto massimo fino a cui posso portare il circuito.
Posso però avere vincoli legati ad energie e ritardi min/Max in cui posso spostarmi.
Es: "io ti do questa energia massima; vai a fare un progetto che la sfrutti, e che riesca ad avere il minimo ritardo possibile"
oppure "la macchina deve andare a 100Mhz, fai le tue scelte architetturali in maniera che la tua macchina consumi meno possibile".

Summary

- Power and energy are now primary design constraints
- Active power still dominating for most applications
  - Supply voltage, activity and capacitance the key parameters
- Leakage becomes major factor in sub-100nm technology nodes
  - Mostly impacted by supply and threshold voltages
- Design has become energy-delay trade-off exercise!

SEBC-L2 MZ 35

Limits to Low-Power Design

- Moore's Law: # Transistors/chip grows 1.5 X every year
- Power-delay product ($P\,t_d$) declined by $1/10^5$ since late 1940's
- Limits to low-power design:
  - Fundamental
  - Material
  - Device
  - Circuit
  - System
  - Practical considerations

SEBC-L2 MZ 36

Mod by Giorgio Fissore, pag 29

Material Limits

- Independent of devices and circuits
- Consider semiconductor cube of undoped material of dimension Δx, embedded in 3D matrix of such cubes
- Limit on switching energy and time (P3) calculated in terms of electrostatic energy stored in cube and transit time of a carrier through the cube:
  - $P = \frac{1}{2} \varepsilon_m E_c^2 \sigma_s^3 t_d^2$
  - σ_s = carrier saturation velocity
  - E_c = self-ionizing electric field strength
  - t_d = cube transit time

SEBC-L2 MZ 40

Material Limits (cont'd)

- Heat removal consideration (P4): Fourier's Law of heat conduction
  - $P = \pi K \sigma_s \Delta T t_d$
  - K = thermal conductivity of semiconductor, A = surface area of heat flow, ΔT = temperature gradient
  - For Si, $P/t_d = 0.21\ W/ns$
  - For GaAs, $P/t_d = 0.69\ W/ns$ (unsuitable: even if faster needs to conduct more than 3 times as much heat for $= t_d$)
  - Silicon-on-Insulator (SOI): $K_{eq} = 0.029\ K_{Si}$, $0.02\ K_{Si}$, $0.013\ K_{Si}$ (most suitable)

SEBC-L2 MZ 41

Material Limits (concluded)

- Interconnect material limit (speed-of-light) (*iL2b(τ)*):
  - Propagation time through interconnect of length L of a material with relative dielectric constant ε_r must be:

$$t_d \geq \frac{L}{c_0/\sqrt{\varepsilon_r}}$$

Mod by Giorgio Fissore, pag 31

SEBC-L2 MZ 42

Circuit Limits (concluded)

- Global interconnect limit (corner to corner) (*iL2c* (τ)):
  - Response time:
  - $\tau = (2.3\ R_{tr} + R_{int})\ C_{int}$
  - R_{int} = Total resistance of interconnection
  - R_{tr} = Output resistance of driver,
  - $R_{int} < 2.3\ R_{tr}$ to neglect delay due to wiring reistance

SEBC-L2 MZ 46

System Limits

- Chip architecture
- Power-delay product of CMOS technology for the chip
- Heat removal capacity of chip package (50 W/cm)
- Clock frequency
- Chip physical size
- Selected architecture (only as an example):
  - Systolic array of 1024 identical square macrocells, each one of dimension L
  - 5-Level clock distribution H-tree
  - Maximum Manhattan distance for clock within macrocell is L, for logic signal it is $2L$

SEBC-L2 MZ 47

System Limits (cont'd)

- Limit on logic gate dimension:

$$[A_{rl}]^{1/2} = \overline{R}_{rl} M \frac{p_w}{e_w n_w}$$

$\overline{R}_{rl}$ = average length of the interconnections

- Gate logic area is logic limited for $n_w = 4$ (# wiring levels), $p_w = 0.2\ mm$ (wiring pitch), $e_w = 0.75$ (wiring efficiency factor) $\overline{R}_{rl} = 6$, $M = 3$ *(fanout of gate)*, and for other values

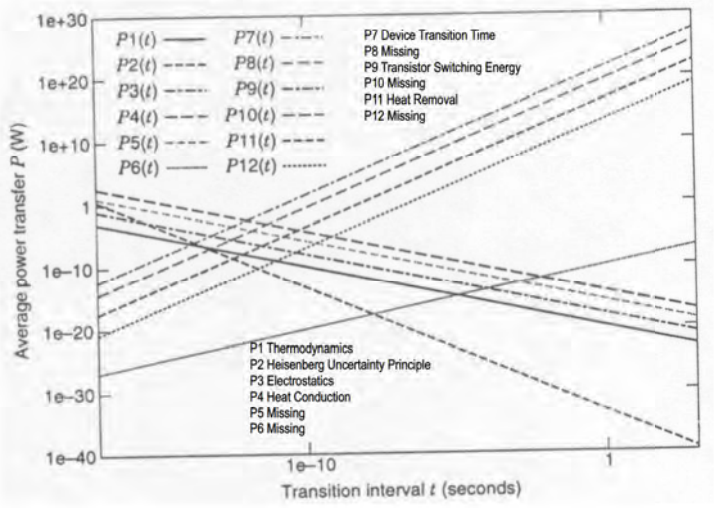
- $t_d = t_{drl} + \frac{T_{cc}}{n_{cp}}$ Tcc = *corner to corner response time*, n_{cp} = *critical path delay*

(effective propagation delay of the composite gate)

SEBC-L2 MZ 48

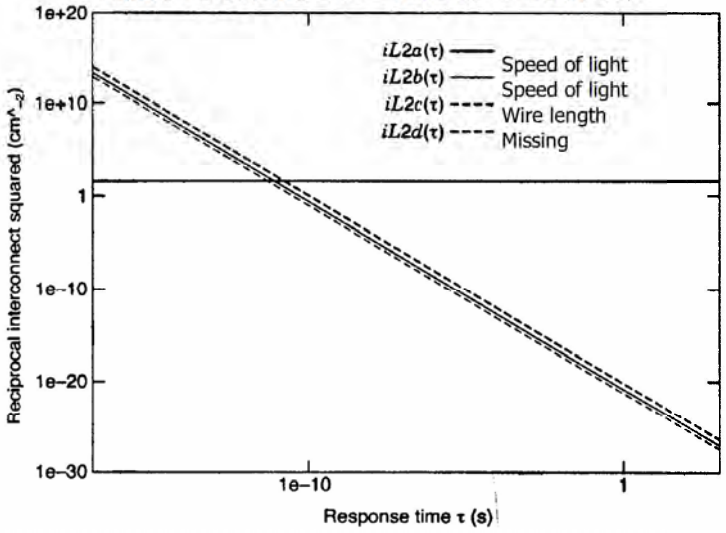
Mod by Giorgio Fissore, pag 33

P Limits Plotted



SEBC-L2 MZ 52

Interconnect Limits Plotted



SEBC-L2 MZ 53

Mod by Giorgio Fissore, pag 35

Why power estimation is necessary

- Determining realistic power requires accurate (i.e. switch-level or below) simulation of the final implementation of a design.
- Infeasible for complex circuits.
- Similar to timing analysis but:
  - Complex relation with other quantities (e.g. circuit capacitance).
  - More difficult to evaluate.
- Power estimators are not just simulators!!!!

SEBC-L3 MZ 4

Power Estimation Issues

- Power estimation implies estimation of:
  - Switching activity.
  - Capacitance.

P = Esw*f*C*Vdd'2

- Different expectations depending on the level of abstraction:
  - Low-levels:
    - Absolute figures
    - Accuracy
    - Design Validation
  - High-levels:
    - Relative figures
    - Estimation time !!!!!!!
    - Design exploration !!!!!

A livello basso, voglio che quando dalla libreria prendo es il full adder, sappia esattamente quanto consuma.
L'analisi a livello basso dev'essere molto precisa, fino al uW, e ciò è fattibile poichè gli oggetti sono piccoli

A livello più alto invece mi va bene una cosa più approssimativa, es quanto consumo se metto una seconda ALU?

SEBC-L3 MZ 5

Gate-level Power Estimation

- Modeling issues
  - Delay models
  - Spatial and temporal correlations
  - Signal feedback (sequential circuits)
- Estimation methods:
  - Simulation-based
  - Probabilistic
  - Statistical

Partiamo da LIV BASSO

.

Devo modellare il circuito, facendo si di avere un modello facilmente utilizzabile (cambiano gli ingressi >> prevedo facilmente le uscite).

.

Dovrò modellizzare questi tre parametri.

.

.

Potrò formularli con questi tre metodi (probabilistico quasi mai usato, e statistico più diffuso)

Mod by Giorgio Fissore, pag 37

SEBC-L3 MZ 6

Real-delay Model

- Real delay mode implies:
  1. Different pin-to-pin delays
  2. Different rise and fall delays
  3. Gates with inertial delays (Input values before and after the input transition must be stable for a time at least equal to the delay of the gate
- 1 and 2 may increase the amount of switching
- 3 may decrease the amount of switching (some transitions are filtered)
- Because of opposite changes, error may cancel
- Unit-delay model estimates are roughly 10 to 15 % smaller, on average

SEBC-L3 MZ 10

Come se il gate avesse una certa banda, tutti i segnali che arrivano a frequenza più alta non passano

E' più preciso, ma anche molto più lungo da calcolare.
Cmq il fatto che 1 e 2 possano aumentare gli switch e 3 li abbassi (non lasciando passare alcuni glitch) si bilanciano abbastanza.
Si usa perciò, la maggior parte delle volte, il modello a unit delay per fare simulazioni.

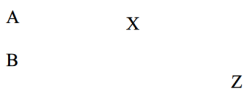
Correlations:

- Spatial correlation:
  Two signals are not independent in the same clock cycle
  - Sources:
    - Structural signal dependencies due to reconvergent fanouts.
    - Input pattern dependencies due to particular sequences of inputs.
- Temporal correlation:
  The values assumed by a signal in two consecutive clock cycles are not independent.
  - Sources:
    - Feedback in sequential circuits
    - Input pattern dependencies due to particular sequences of inputs

SEBC-L3 MZ 11

Calcolare in modo probabilistico il consumo delle porte non è affatto facile, poichè non si possono nemmeno considerare i vari elementi del circuito come stocasticamente indipendenti;
vi sono infatti sia correlazioni spaziali, sia temporali.
Questo porta il modello probabilistico ad essere esageratamente complicato (ingestibile)

Problem: Reconvergent Fanout



Reconvergence

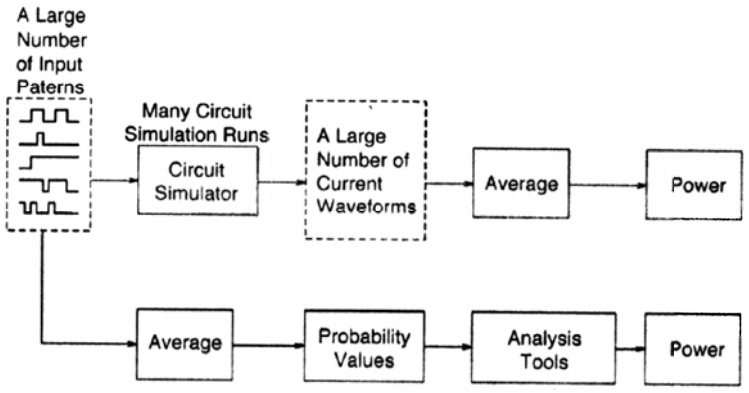
P(Z=1) = P(B=1) x P(X=1 | B=1)

Becomes complex and intractable real fast

SEBC-L3 MZ 12

Mod by Giorgio Fissore, pag 39

Flows for Power Estimation



SEBC-L3 MZ 16

Simulation-Based Methods

- Based on the simulation of input patterns.
- Also called *dynamic*
- Simplified solutions:
  - Assume same supply voltage throughout the whole chip.
  - Reduce the problem to that of estimating the power supply currents drawn by the components.
  - Logic simulation can be used to estimate the currents.
- Strongly pattern-dependent

SEBC-L3 MZ 17

Questo metodo è fortemente dipendente dall'accuratezza dei pattern in ingresso;
la difficoltà sta proprio nel trovare buoni vettori di ingresso.
Se buona funziona bene, altrimenti rischia di non essere accurato.

Simulation-Based Methods

- Advantages:
  - Use accurate models (i.e. timing, power) of state-of-the-art circuit simulators (e.g. SPICE).
  - Very accurate estimates
- Disadvantages:
  - Require complete information about input patterns (exhaustive input pattern simulation).
  - Very costly in time => applicability limited to small circuits
  - Mainly used for design validation.

SEBC-L3 MZ 18

Mod by Giorgio Fissore, pag 41

Probabilistic Methods: Definitions

- The starting hypothesis is that the system is a *synchronous* one, with a clock signal that controls all the internal activities (one transition/clock cycle).
- Signal probability $p^x(n)$ of node n : average fraction of clock cycles (!!!) in which the steady state of n is x
- Transition probability, $p^{ij}(n)$ of node n : average fraction of clock cycles in which the steady state of n changes from i to j.
- Switching activity, $Esw(n)$ or $A(n)$ of node n : expected number of transitions at n per clock cycle.

SEBC-L3 MZ 22

p'x(n) = #medio di CLK in cui il segnale vale 1

p'ij(n) = #medio di CLK in cui lo stato di n cambia da i a j

Computing Signal Probabilities

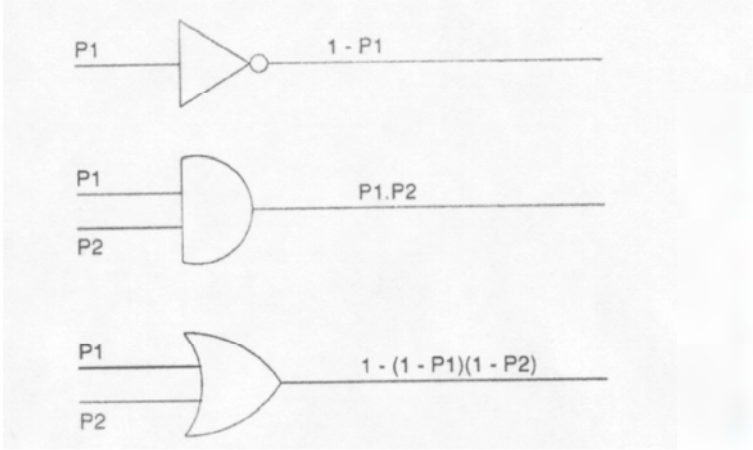
- Output probability of a basic boolean function is calculated with standard probabilistic theory
- for an AND gate, if inputs have P1 and P2 signal prob. the output will have P(out) = P1*P2
- for an OR gate, the output will have

$$P(out) = 1 - (1-P1)(1-P2)$$

(the probability to have a 1 as output is equal to 1 less the product of the probabilities to have both inputs at 0)

SEBC-L3 MZ 23

Example Signal Probabilities



SEBC-L3 MZ 24

Se non considero i problemi della riconvergenza e della correlazione, se mi calcolo le probabilità dei segnali per le varie porte, sono in gradi di calcolarmi la Esw in ciascun punto del circuito

Mod by Giorgio Fissore, pag 43

Statistical methods

- Advantages:
  - **Fast**
  - **Can manage large circuits**
  - **Desired degree of accuracy as a function of # of measurements**
  - **No special arrangements for sequential circuits (although a different convergence criterion may be required).**
- Disadvantages:
  - *Gaussian assumption may be wrong for some circuits*
  - *For some input sequences # of measurements may be large*
  - *Not suitable for computing power of individual gates.*

SEBC-L3 MZ 28

Parker-McCluskey Signal Probability Calculation Algorithm

(algoritmo per il calcolo delle prob che tiene conto anche della riconvergenza - ank se cmq non lo useremo-)

- Signal probabilities are used to accurate estimate signal activity
- It is essential to accurately calculate signal prob. for further use in estimating *activity*
- A general algorithm, proposed by Parker and McCluskey is widely used for a generic system (no requirements on clock regime...... also purely combinatorial !!!!!!)

SEBC-L3 MZ 29

Parker-McCluskey Signal Probability Calculation Algorithm

- Inputs: Signal probabilities of all circuit inputs
- Outputs: Signal probabilities of all circuit nodes
- Step 1: Assign a variable for each input and logic gate
- Step 2: Going from inputs to outputs, compute symbolic probability of each gate output
- Step 3: Suppress all exponents in symbolic expressions

Premise: Reconvergent fanouts make signals correlated, and higher-order powers of probabilities cannot be present in symbolic expressions when primary inputs are independent (intuitively....)

Forse "ogni volta che trovi esponenti cancella" o qlcs del genere

Mo... pag 45

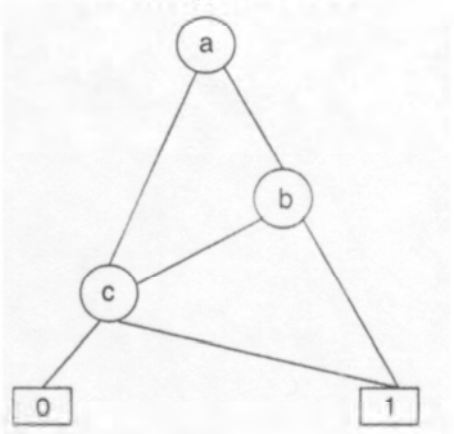
SEBC-L3 MZ 30

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 47 di 402

Binary Decision Diagram to Calculate Signal Probability

- Traverse *Binary Decision Diagram* (BDD) from root in depth-first traversal, with post-order evaluation of P () at every node, to determine:
  - $P(f) = P(x_1)\, P(f_{x1}) + P(\overline{x_1})\, P(f_{\overline{x1}})$

Example BDD



$f = a\,b + c$

the tree rooted to the left of *a* represents $f\bar{a}$; the right..... *fa*

Estimating Signal Activity

- Probabilistic techniques:
  - Boolean Difference (Sellers):
    - Symbolic Boolean method to calculate signal activity (see notes)
    - Requires an *Extended State Transition Graph* (ESTG) to calculate activities for sequential circuits
    - Use Chapmann-Kolmolgorov Equations
      - Markov Probabilistic Process Model
    - Need to use an approximate solution method – exact method is too slow
      - Approximate method is exact for tree-structured pipelined circuits
  - Only gives lower-bound on activity, because the method ignores glitching power
    - Due to use of zero-delay logic simulation model
  - Inactive circuit parts still contribute inordinately to power estimate
    - Due to assumption that even turned-off inputs have 0.5 prob.

Mod by Giorgio Fissore, pag 47

Simultaneous Switching Problems

- The method works theoretically with ALL the clocking regimes (also combinational only regimes !).
- The method shown fails in case of simultaneous switching of inputs.
- A new algorithm, based on Generalized Boolean Difference, can be used to consider this condition (see book… a lot of mathematics….and logic…).

SEBC-L3 MZ 40

PROBLEMA!!
questo metodo si perde tutte le commutazioni multiple
ed in più, come visto prima, non prende tutti (gli switch? i glitch?).
.
E' un metodo quindi assolutamente deficitario.

Simultaneous Switching Problems

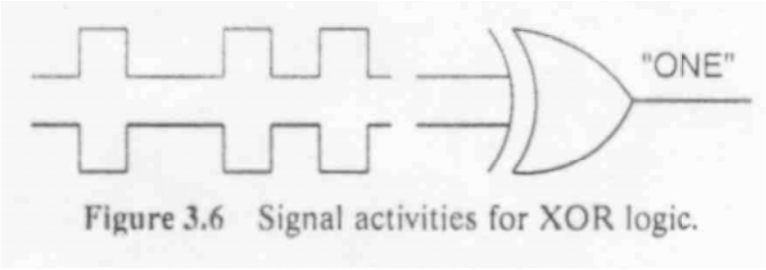


Figure 3.6 Signal activities for XOR logic.

SEBC-L3 MZ 41

Simultaneous Switching Problems

- A different approach can be used for the majority of digital systems;
- Suppose to have a synchronous clock (fck) system where combinational paths are inserted between latches.

SEBC-L3 MZ 42

Mod by Giorgio Fissore, pag 49

Circuit function

- The probability for the output to make a transition is given by the probability than the current state is 0 times the probability that the next state is 1 *plus* the probability than the current state is 1 times the probability that the next state is 0.
- As a consequence:

$Esw = (p(O=0)p(O=1) + p(O=1)p(O=0) = 3/8$

SEBC-L3 MZ 46

Circuit function

- Example: 2-input XOR **static** CMOS gate:

  O = XOR (A,B)
- Assume only one input transition per clock cycle (synchronous clock system!!!)
- Inputs are equiprobable: p(A=1)=p(B=1)=1/2
- The probability for the output to 1 is

p(O=1) = (1-p(A=1))p(B=1) + (p(A=1))(1-p(B=1))= 1/2

p(O=0)= 1- P(O=1) = 1/2

SEBC-L3 MZ 47

Circuit function

- The probability for the output to make a transition is given by the probability than the current state is 0 times the probability that the next state is 1 *plus* the probability than the current state is 1 times the probability that the next state is 0.
- As a consequence:

$Esw = (p(O=0)p(O=1) + p(O=1)p(O=0) = ½$

- The *Esw* probability that the output switches heavily depends on the circuit function!!!!!

SEBC-L3 MZ 48

Mod by Giorgio Fissore, pag 51

Circuit Technology

- Example: 2-input NOR **Dynamic** CMOS gate:
  O = NOR (A,B)
- Assume only one input transition per clock cycle (synchronous clock system!!!)
- Inputs are equiprobable: p(A=1)=p(B=1)=1/2
- The probability for the output to be discharged is:
  p(O=0) = 3/4
- Therefore the probability for C_L to be re-charged at the next clock cycle equals p(O=0)

SEBC-L3 MZ 52

Transition Probabilities for Dynamic Gates

	$P_{0\to1}$
AND	$(1-P_AP_B)$
OR	$(1-P_A)(1-P_B)$
EXOR	$(1 - (P_A + P_B - 2P_AP_B))$

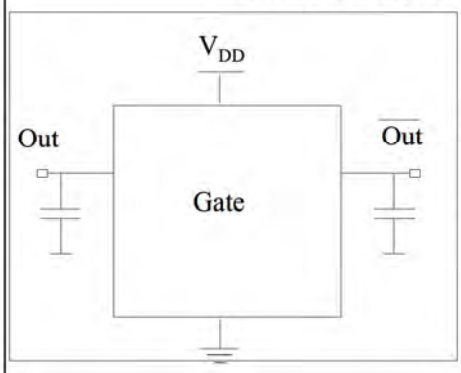
Switching Activity for Precharged Dynamic Gates

$$P_{0->1} = P_0$$

La tecnologia con cui realizzo le porte impatta pesantemente la Esw del mio circuito!!

SEBC-L3 MZ 53

Differential Logic?



Static:
Activity is doubled
Dynamic:
Transition probability is 1!

Hence: power always increases.

Usando una logica differenziale l'attività risulta raddoppiata, se non ho grossi vantaggi, questa strada va quindi scartata

Mod by Giorgio Fissore, pag 53

SEBC-L3 MZ 54

Circuit Topology

- *Tree Structure (Static CMOS gates):*

  O1 = A B ; O2 = C D ; O = O1 O2

- All inputs are equiprobable:

P(O1=1) = 1/4 P(O1=0) =3/4

P(O2=1) = 1/4 P(O2=0) =3/4

P(O=1) = 1/16 P(O=0) = 15/16

Esw(O1) = 3/8 *Esw(O2) = 3/8*

Esw(O) = 15/128

SEBC-L3 MZ 58

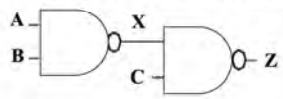
Circuit Topology

- Timing skew between signals may cause spurious transitions resulting in extra power dissipation!!!
- The dynamic component of the switching activities is due to the **glitches** (next lectures…).
- These glitches can depend on the topology and relative delays: chain topology is less robust to skew as balanced tree, where the delay of each stage can be controlled to be similar (next slides).
- Dynamic CMOS gates are not affected by glitches, since each gate output can make at most one power consuming transition per clock cycle!!!

SEBC-L3 MZ 59

Glitching in Static CMOS

also called: dynamic hazards



ABC 101 000

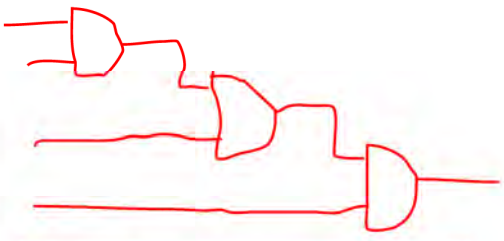
X

Z

Unit Delay

Observe: No glitching in dynamic circuits

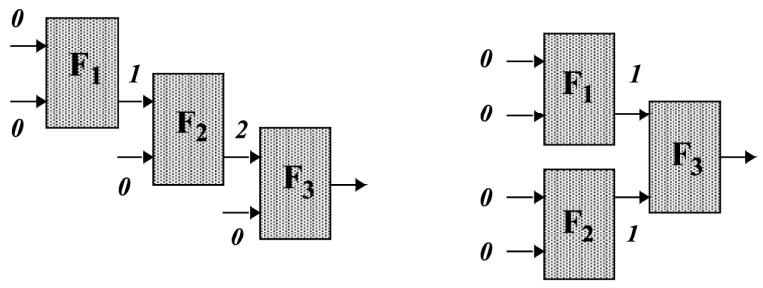
SEBC-L3 MZ 60



Un glitch nella catena, si propaga per tutta la catena

Mod by Giorgio Fissore, pag 55

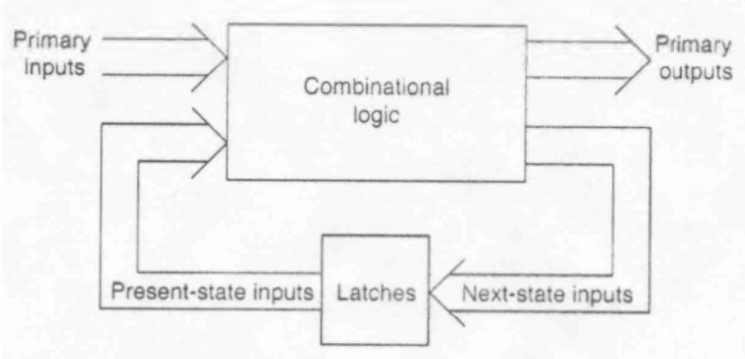
How to Cope with Glitching?



Equalize Lengths of Timing Paths Through Design

SEBC-L3 MZ 64

Representing Sequential Circuits



SEBC-L3 MZ 65

Nell'evoluzione della macchina ho una forte correlazione sia spaziale che temporale. Diventa quindi quasi impossibile fare dei calcoli probabilistici.

Representing Sequential Circuits

- Accurate power estimation is more difficult since signal prob. and activities at the state inputs are not known ahead.
- The prob. of being in different states must be determined in the steady state.
- Taking only spatial correlations of signals into consideration:

Pi= [p1..pn] signal prob. of *n* independent primary inputs

Pout = F(Pi,Ps) i=inputs, s = state

Pn = F'(Pi,Ps) Pn= signal prob. for next-state inputs

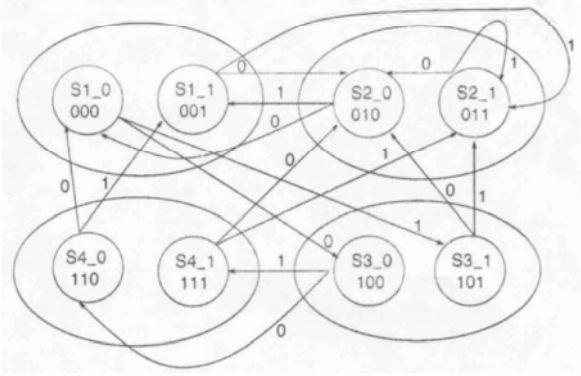
- In steady-state, noting that Pn = Ps, it is possible to solve the set of equations for state prob.

Mod by Giorgio Fissore, pag 57

SEBC-L3 MZ 66

Extended State Transition Graph

- Represent each state with 2 present state bits and one present input bit to facilitate correct prob. calculation (the variable on each edge is the next input (at time T)

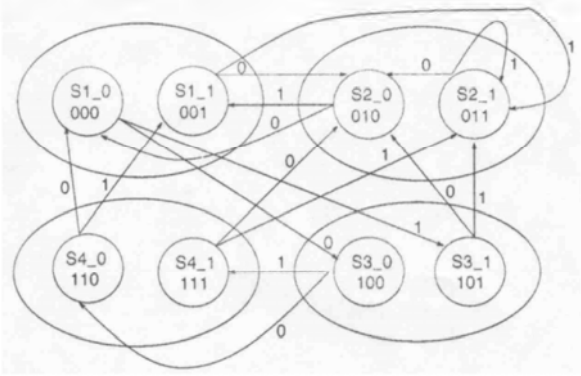


SEBC-L3 MZ 70

"non vogliamo fare simulazioni".
Allora espando ogni singolo stato, mettendo nei nuovi sottostati anche i bit in ingresso (es stato s1 con bit_ingr 1 e bit_ingr 0).
Così non servirebbero simulazioni, ma il numero di stati esplode troppo in fretta.

Extended State Transition Graph

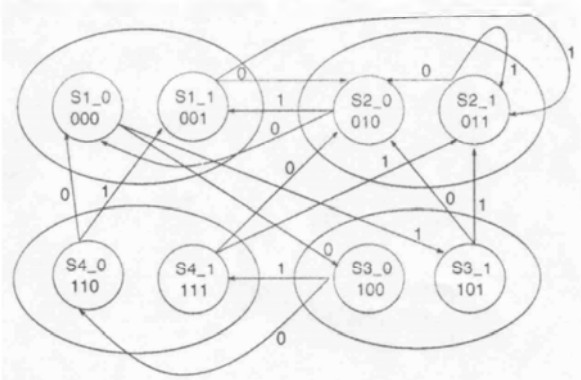
- Each state is split in 2**N states, where N is the number of primary inputs



SEBC-L3 MZ 71

Extended State Transition Graph

- State transitions do not depend on the history: signal probabilities and activities can be calculated....but..... for large FSM it becomes very CPU expensive!!!!



SEBC-L3 MZ 72

Mod by Giorgio Fissore, pag 59

Why Is Correlation a Problem?

- I^T and $ns2^0$ appear to be independent
  - No common ancestor node in graph
- But, $ns2^0$ is topologically dependent on node I^0 (input)
  - I^0 and I^T are temporally correlated
  - Gives erroneous power estimate unless we correct for this
    - Define I as a temporally reconvergent node, rather than a topologically reconvergent node

SEBC-L3 MZ 76

Summary of Power Estimation

- Probabilistic techniques: Not useful
  - Only give lower-bound on activity, ignore glitching power
  - Inactive circuit parts contribute inordinately to power estimate
- Major Problem – unable to estimate glitching power
- Probability theory needs to be augmented with Monte-Carlo analysis for power estimation

SEBC-L3 MZ 77

Mod by Giorgio Fissore, pag 61

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 63 di 402

Necessary Relationships

- $|\bar{p} - P_{avg}| < \frac{t_{\alpha/2} s}{\sqrt{N}}$ $\qquad \frac{|\bar{p} - P_{avg}|}{\bar{p}} < \frac{t_{\alpha/2} s}{\bar{p}\sqrt{N}}$
- Required condition: $\frac{t_{\alpha/2} s}{\bar{p}\sqrt{N}} < \epsilon$
- Required # simulations for desired confidence level:

$$N \geq \left(\frac{t_{\alpha/2} s}{\bar{p}\epsilon} \right)^2$$

  - Sometimes need only 10 simulations to get accuracy
  - CPU time comparable to probabilistic techniques
  - May not accurately estimate power of individual gates that switch very infrequently

SEBC-L4 MZ 4

Modification for Low-Activity Gates

- n_i, $1 \leq i \leq N$, # transitions at node during simulation i
- $\bar{n} = n_i / N$ = average # transitions at node
  - Close to normal distribution for large N (Central Limit Theorem of Statistics)
- β = true expected value of ave. # node transitions
- s = measured std. dev. of N values of n_i
- $(1-\alpha) \times 100\%$ = confidence level
- $z_{\alpha/2}$ obtained from normal distribution

SEBC-L4 MZ 5

Necessary Relationships (Low-Activity)

$$\frac{\beta - \bar{n}}{\bar{n}} \leq \frac{z_{\alpha/2} s}{\bar{n}\sqrt{N}} \qquad (3.33)$$

$$N \geq \left(\frac{z_{\alpha/2} s}{\epsilon \bar{n}} \right)^2$$

$$N \geq \left(\frac{z_{\alpha/2} s}{\epsilon \beta_{min}} \right)^2$$

$$|\beta - \bar{n}| \leq \frac{z_{\alpha/2} s}{\sqrt{N}} \leq \beta_{min}\, \epsilon$$

- Low-activity nodes take longest time to converge, have least effect on ave. power dissipation & circuit reliability

Mod by Giorgio Fissore, pag 63

SEBC-L4 MZ 6

Problems with Example

- G1 = {s1 s0, $\overline{s1}$ $\overline{s0}$} and G2 = {$\overline{s1}$ s0, s1 $\overline{s0}$}
- P(transition between G1 and G2) is very low
  - Near-closed state sets
- suppose y = output = ($\overline{s1}$ $\overline{s0}$ + s1 s0) x1
  - Considering only G1, y = x1
  - Considering only G2, y = 0
  - Very different probability behavior for G1 and G2 sets
  - Data sample set is *biased* – causes errors

SEBC-L4 MZ 10

Removal of Biasing

- Must know $P(G_1)$ and $P(G_2)$
- Compute normalized activity:
  - $a(y) = P(G_1) \times a(y, G_1) + P(G_2) \times a(y, G_2)$
  - May assume that PIs are temporally uncorrelated (could be wrong) or that they are Markov (future value depends only on present value and not on the past)
  - With Markov assumption, can implicitly compute $P(G_1)$ and $P(G_2)$
  - Transform STG into ESTG that is Markov
  - P^k_{warmup} is probability of reaching G_i after k clock cycles with any initial state
  - $|P^k_{warmup}(G_i) - P(G_i)| \leq N_s |\lambda_2|^k$
  - λ_2 is second-largest eigenvalue of transition matrix
  - N_s = # ESTG states

SEBC-L4 MZ 11

Removal of Biasing (concluded)

- k = # clock cycles needed for warmup

$$k \geq \frac{\ln[N_s/\epsilon_G P(G_i)]}{\ln 1/\lambda_2} \qquad (3.39)$$

- Repeat Procedure N times, and take mean of samples

$$\frac{\left(\sum_{j=1}^{N\cdot P(G_1)} a_j(y|G_1) + \sum_{j=1}^{N\cdot P(G_2)} a_j(y|G_2)\right)}{N} = \frac{N\cdot P(G_1)a(y|G_1) + N\cdot P(G_2)a(y|G_2)}{N} = P(G_1)a(y|G_1) + P(G_2)a(y|G_2)$$

SEBC-L4 MZ 12

Mod by Giorgio Fissore, pag 65

Three-Valued Logic Simulation

Used by logic simulators to model static hazards

In the table AND values calculated according to the possible hazard (X)

AND	0	1	X
0	0	0	0
1	0	1	X
X	0	X	X

SEBC-L4 MZ 16

Modo più semplice per simulare in maniera logica i glitch (ma non usati), logica a tre stati, con il terzo X che rappresenta il glitch.

Six-Valued Logic Simulation

- t' is time instant in between t and t + 1 to detect static hazards

Logic Representation	Bit Sequences at $t, t', t+1$
0—Static 0	000
1—Static 1	111
R—Rising	0U1
F—Falling	1U0
SH0—Static 0 hazard	0U0
SH1—Static 1 hazard	1U1

SEBC-L4 MZ

Metodo utilizzato effettivamente dai simulatori; tiene conto di glitch statici.

U = undefined

Six-Valued Logic for AND Gate (Pessimistic)

- Truth table of the logic simulations for the 6-valued logic;
- Pessimistic since some hazards might not be present under certain delay conditions

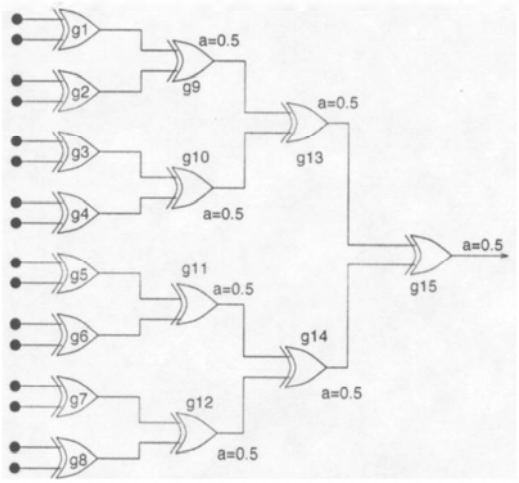
AND	0	1	R	F	SH0	SH1
0	0	0	0	0	0	0
1	0	1	R	F	SH0	SH1
R	0	R	R	SH0	SH0	R
F	0	F	SH0	F	SH0	F
SH0	0	SH0	SH0	SH0	SH0	SH0
SH1	0	SH1	R	F	SH0	SH1

SEBC-L4 MZ 18

Possiamo così verificare, ad esempio l'effetto di un glitch statico 0>1 all'interno della nostra catena di AND, metodo pessimistico (buona cosa perchè vogliamo sapere quanto consuma al max) poichè si ipotizza che se c'è la possibilità che ci sia un glitch, questo ci sia effettivamente.

Mod by Giorgio Fissore, pag 67

Nominal Balanced Path Delay Models



SEBC-L4 MZ 22

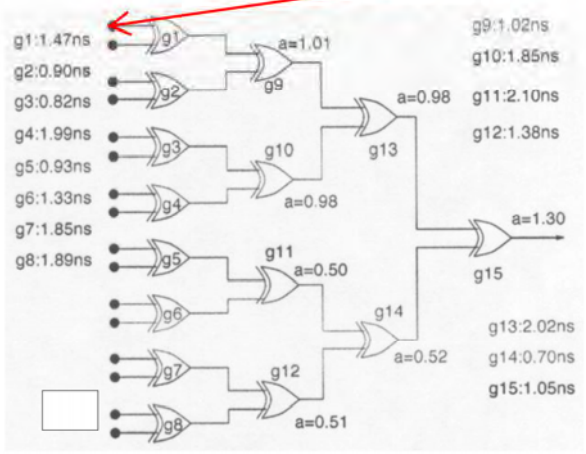
Vediamo come i ritardi influenzano i glitch, in due immagini a 1) ritardo fisso 2) ritardo non costante (pag dopo)

Tree-Structured Random Delay Variations in Circuit

- Unfortunately gate delays may have variations
- As a consequence glitches occur and the activities in individual nodes change (possibly increasing!!!)
- To capture this random behavior, the sources of uncertainty are represented by probability distributions.

SEBC-L4 MZ 23

Tree-Structured Random Delay Variations in Circuit



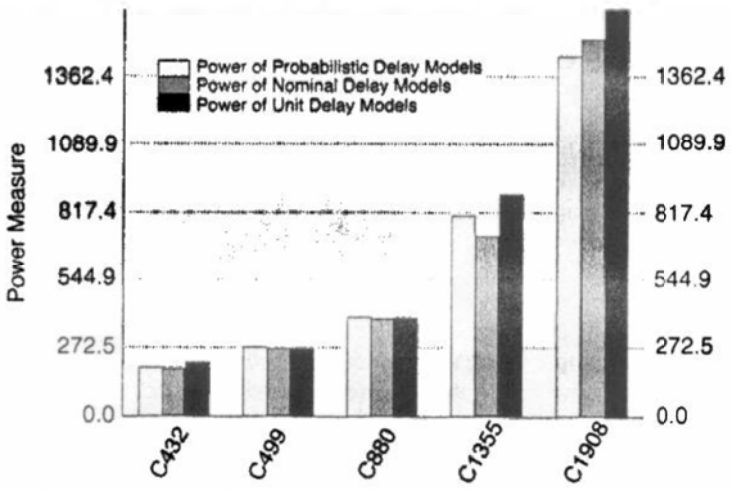
SEBC-L4 MZ 24

Addirittura più di una commutazione per colpo di CLK

Tanto più la tecnologia si rimpicciolisce, tanto più aumenta la distribuzione dei parametri (fondamentale sotto i 30 nm dove la dispersione è talmente vasta che non ha più senso parlare di valori tipici)

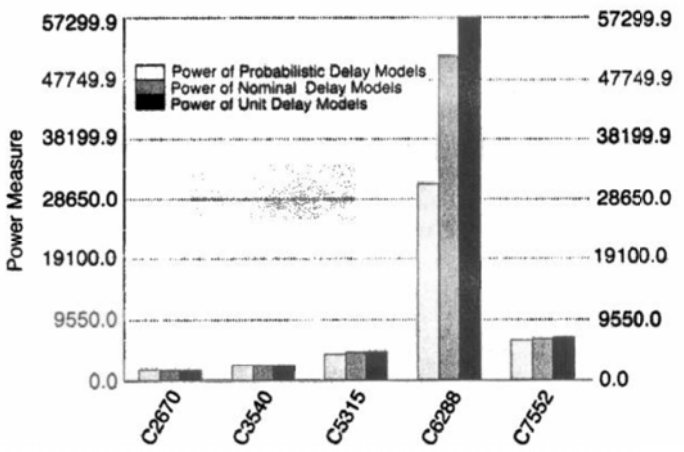
Mod by Giorgio Fissore, pag 69

Relative Dynamic Power



SEBC-L4 MZ 28

Power with Different Delay Models



SEBC-L4 MZ 29

Sensitivity Analysis

- Monte-Carlo methods are accurate if exact probabilities and activities are available for PIs
- Should represent average power dissipation as a range [$Power_{min}$, $Power_{max}$]
- Takes 2.3 CPU s to do one symbolic simulation for c432 – need to do 2^{36} simulations to cover all input cases (impractical... more than 5000 years!)
  - Use power sensitivities instead

PI: Primary Input

Mod by Giorgio Fissore, pag 71

SEBC-L4 MZ 30

Power Estimation via Vector Compaction

- Work of Tsui, Marculescu, Marculescu, and Pedram
- Method: Generate compact vector set that represents the original vector set, but takes much less time to simulate
- Given vector sequence S_1 of length L_1 with property P_1, generate a much shorter sequence S_2 of length L_2 with similar property P_2 and use it to estimate power
- Properties P_1 and P_2 are the pair-wise transition probabilities among all possible input combinations
- Compacted the sequence 20 times, while keeping power estimation error within 5 %

SEBC-L4 MZ 34

Summary of Power Estimation

- Probabilistic techniques: Not useful
  - Only give lower-bound on activity, ignore glitching power
  - Inactive circuit parts contribute inordinately to power estimate
- Statistical techniques: Useful
  - Repeatedly simulate circuit with logic simulator, noting node activities
  - Randomly-generated inputs
  - Statistical mean estimation techniques with Monte Carlo simulation
- Glitching Power estimated by Monte Carlo methods + probabilistic delay
- Power Sensitivity used to estimate min. and max. average power
- Power estimated with input vector compaction or information theory

SEBC-L4 MZ 35

Unico metodo compatibile con l'utilizzo durante il progetto dell'architettura. (faccio il progetto, simulo, modifico, controllo se meglio o peggio).

Gate-Level Power Estimation: Summary

- Essential factors to capture for accurate logic-level power estimation:
  - Delay models
  - Signal correlations
- Available techniques guarantee very low estimation errors:
- Three main classes of methods:
  - Simulation-based
  - Probabilistic
  - Statistical
- At this level, other metrics can be estimated (e.g. peak power)

SEBC-L4 MZ 36

Così si chiude la stima di potenza a livello gate; va però ricordato che per arrivare a questo livello, bisogna prima passare per la sintesi, e questa può richiedere qualche ora.
Passo allora alla valutazione al livello dei registri RTL, che mi permette di fare una valutazione presintesi.

Mod by Giorgio Fissore, pag 73

RTL-Level Power Estimation

- Analytical models relate power dissipation to physical quantities to express activity and capacitance:
  - Best suited for:
    - Black-box estimates (no data are available for individual blocks).
    - Fixed structure components, e.g. memories.
- Empirical models are based on a measure of power, from which a model is built:
  - Best suited for library-based approaches.
  - Can be built for custom functional blocks, as long as some real power figures are available (e.g. taken from previous implementations).

SEBC-L4 MZ 40

Metodi analitici, non usati, cercano di tirare fuori una qualche equazione (funzionano meglio per strutture estremamente ripetitive es memorie, o si usa in black box, dove non ho nessun altra possibilità)

Utilizziamo questo

Analytical Complexity-Based Methods

- Complexity is used as capacitance estimate.
- *Complexity = Equivalent gate count*
- Power of a block is roughly estimated as:

$$P = (\#\ of\ equivalent\ gates)\ P_{gate}$$

- Limitations:
  - Based on power consumption of a reference gate => No account for circuit implementation or clocking strategy.
  - Fixed, static *activity factor* assumed.

SEBC-L4 MZ 41

Analytical Activity-Based Methods

- Activity is regarded as dominant factor of power consumption.
- Models are based on information-theory concepts: exploit correlation between activity end *entropy*.
- Abstract power model:

$$P=1/2\, f\, V^2_{dd} C(H) S(H)$$

- Use entropy H to express:
  - S (average switching activity)
  - C (average capacitance).
- Need of efficient methods for entropy calculation.
- Drawbacks: No delay modelling and no accounting of glitching effects.

SEBC-L4 MZ 42

Mod by Giorgio Fissore, pag 75

Not Just Macromodelling

- Macromodelling is suited for datapath blocks.
- Ad-hoc models are required for other components:
  - Memory
  - Controllers
  - Interconnect

SEBC-L4 MZ 46

Memory models

- Parameters of the model belong to different categories:
  - Structural: Rows, bit-width, bit per column, number of addressing lines.
  - Timing: Time of the write-enable signal during write.
  - Access-mode: Read, write, extra-read-before-write.
  - Technology: Supply voltage.
  - Dynamic: Address/data bus activity and number of toggles per access.
- *Non-componentized* (simpler) vs. *componentized* (more complex) models depending on target accuracy.

SEBC-L4 MZ 47

Controller models

- Controller is a behavioral description (state machine).
- Accuracy loss is unavoidable.
- Approach: Account for complexity in the model:

$$P = F(Activity, Complexity)$$

- Switching activity:

$$Esw(I) = input\ switching\ activity$$
$$Esw(O) = output\ switching\ activity$$

- Complexity:

$$Ni = \#\ of\ inputs$$
$$Nc = \#\ of\ "cubes"$$
$$No = \#\ of\ outputs$$

SEBC-L4 MZ 48

Mod by Giorgio Fissore, pag 77

Lecture 5
Circuit Level Power Estimation

- **Domino CMOS Power Estimation**
- **Circuit-level methods**
- **ATPG methods**
- **Summary**

SEBC-L5 MZ 1

Appunti di Giorgio Fissore
Disponibili in centro stampa

Domino CMOS Power Estimation

- Unlike static CMOS, domino logic uses a precharging circuit and a 2-phase clocking regime
- During the precharging phase output is pulled-up while in the evaluation phase output possibly is discharged.
- Ignoring short-circuit and leakage currents, power must be calculated differently from static CMOS

SEBC-L5 MZ 2

Domino CMOS Power Estimation

- Ignore direct-path short-circuit currents
- Average power over all circuit nodes:

$$\text{Power}_{avg} = \frac{1}{2} V_{dd}^2 \sum_i C_i A(i)$$

- Ci= capacitive load of node i, A(i)= activity of node i
- Normalized power dissipation measure:

$$\Phi = \sum_i \text{fanout}_i \times a(i) \qquad (3.64)$$

fanout_i is # fanouts of node **i (proportional to Ci),**
a(i) =normalized activity = A(i)/Tck

SEBC-L5 MZ 3

Mod by Giorgio Fissore, pag 79

Circuit Reliability

- Signal activity is a good measure of electro-migration and hot-electron degradation (happens only in saturation) in circuits
  - Negative electro-migration leads to a *hillock*, or metal build-up, that can short adjacent metal lines
- Hot electron injection into gate oxide layer degrades g_m and V_t, cumulative over time, limits useful life of transistor

SEBC-L5 MZ 7

Il mio prodotto, una volta venduto, deve continuare a funzionare per un po'.
Esistono dei fenomeni che invece ne accorciano la vita:
-Elettromigrazione: se la densità di corrente aumenta troppo, questa inizia ad erodere la pista nei punti più critici (come un fiume che, se troppo impetuoso, va a modificare il letto in cui scorre). Questo fenomeno inoltre, si autoalimenta con una reazione positiva: più una pista viene assottigliata, più, a parità di corrente, aumenterà la densità di corrente in quel punto, fino ad erodere completamente quella pista (trasformandola in un circuito aperto). FENOMENO PERICOLOSO perchè al momento della produzione il circuito funziona, ma risulterà danneggiato magari dopo qualche mese.
-Hot-electron: elettroni ad alta energia, tendono ad alterare la struttura elettrica del dispositivo (vengono modificate transconduttanze, tensioni di soglia,..) fino a rendere il dispositivo non più funzionante.

Circuit Reliability

- *Mean time to failure* (MTF) due to electro-migration:

$$MTF = \frac{K}{J^2} \qquad (3.69)$$

- Minimize electro-migration by minimizing:

$$E = \sum_i C_i A_i \qquad (3.70)$$

- Pavg = IddVdd= ½ (Vdd**2) Ai Ci

→ to minimize Idd it is necessary to minimize Ai Ci !!!

- J = Average Current Density
- K has statistical distribution, independent of J

SEBC-L5 MZ 8

La densità di corrente, è legata al numero di commutazioni (qui chiamato Ai, ma è sempre la switching activity).
>> la corrente media è legata alla somma di tutte le Esw moltiplicate per le capacità
>> noi ci occuperemo solo di ottimizzare la potenza, ma in questa maniera aumenteremo anche sempre la vita del dispositivo! (es se guido un'auto senza tirarla, ridurrò i consumi e ne allungherò la vita).

Circuit Reliability (cont'd)

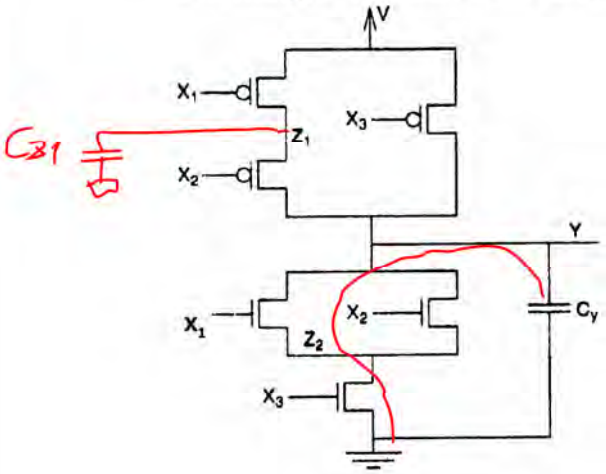
- As MOS scaled down, hot carriers are injected into gate oxide, due to lateral electrical field produced by source-drain voltage
- Trapping of these carriers degrades transistor transconductance and voltage threshold.
- More important, because oxide charging is cummulative over time → reduction of the life of the device
- $H_{gate} = A_{gate}\, f_{gate}$ = hot electron degradation

A= gate activity; f= fanout

SEBC-L5 MZ 9

Mod by Giorgio Fissore, pag 81

Example: CMOS Gate



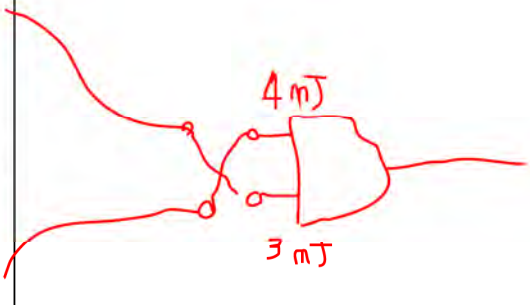
SEBC-L5 MZ

Esempio 1
$x_1 = 0$
$x_2 = 0$
$x_3 = 1$ >> Vu = 1
poi, cambia x_1 creando un percorso per scaricare Cy.
In questo caso, visto che x_2 sopra conduce ancora, si scaricherà anche C_{z1}.
Esempio 2
Se invece al posto di x_1 commutasse x_2 (dal punto di vista logico ha lo stesso effetto), avrei una separazione tra il nodo z_1 ed il nodo z_2 >>C_{z1} questa volta non si scarica!!

QUINDI: gli ingressi, indistinguibili dal punto di vista logico, presentano ritardi e consumi diversi!! (diversa è la capacità da scaricare) >>se ho segnali da far commutare velocemente, posso ottimizzarli collegandoli all'ingresso con il ritardo minore (x_2). Questo causerà anche un minore consumo di potenza.

Signal Connections to Minimize Power

- If x2 =0, x1 0→1 and x3 =1 the capacitance to be discharged is Cy+Cz1
- If x1=0, x2 0→1, x3 =1 only Cy is discharged during falling transition.
- It is important to connect external inputs to internal pins according to the activity!!!
- For instance if signal A has high activity and signal B has low activity, then connect A to x_2 and B to x_1 to minimize power



Regola generale: "più un transistor è vicino all'uscita, più va veloce e meno consuma" (meno nodi intermedi da caricare e scaricare)

SEBC-L5 MZ 14

"pin swapping": collegare ingressi con Esw maggiore, a quei pin che consumano meno!

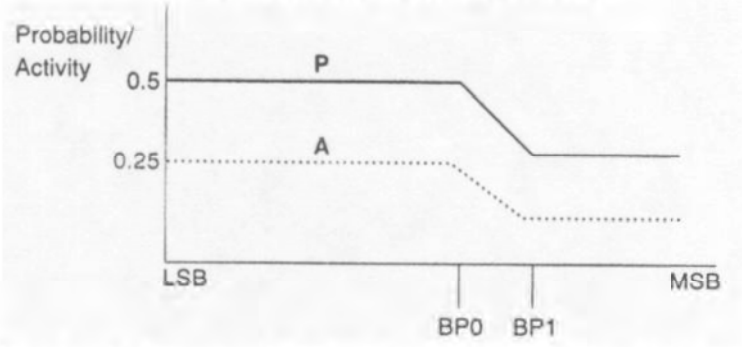
Signal Connections to Minimize Power

- Due to the internal nodes capacitance they need to be treated as circuit nodes distinct from outputs
- Activities of internal nodes must therefore be considered (np-hard!!!): unfortunately they depend on the previous clock cycles (as in sequential circuits)
- An accurate analysis of conducting paths from the nodes and Vdd and Gnd is required.

SEBC-L5 MZ 15

Mod by Giorgio Fissore, pag 83

Word-Level Data Model



- Signal prob. of lower order bits of a word are uncorrelated in space and time (independent of data distribution)
- BP0 and BP1 are breakpoints tied to statistical parameters and define when correlation starts

SEBC-L5 MZ 19

Most Significant Bit Data Models

- F_1 = bivariate normal distribution function
- F_{01} = univariate normal distribution function
- ρ_1 = lag 1 correlation coefficient
- Breakpoint equations (where MSB correlation starts):

$$BP_0 = \log_2\left(\frac{3\sigma}{32}\right)$$

$$BP_0 = \log_2\left(\frac{3\sigma}{32}\right) - \log_2(1-\rho_1^2)^{0.5}$$

$$P_{MSBs} = F_1\left(\frac{\mu}{\sigma}\right)$$

$$A_{MSBs} = F_{01}\left(\frac{\mu}{\sigma}, \rho_1\right)$$

$$BP_1 = \log_2(|\mu| + 3\sigma)$$

SEBC-L5 MZ 20

Information Theoretic Approaches

- Estimate power at Register Transfer Level
  - Estimate entropy and use it to find signal activity
- Entropy H (x) defined in terms of signal probability p:

$$H(x) = p\log_2\frac{1}{p} + (1-p)\log_2\frac{1}{1-p}$$

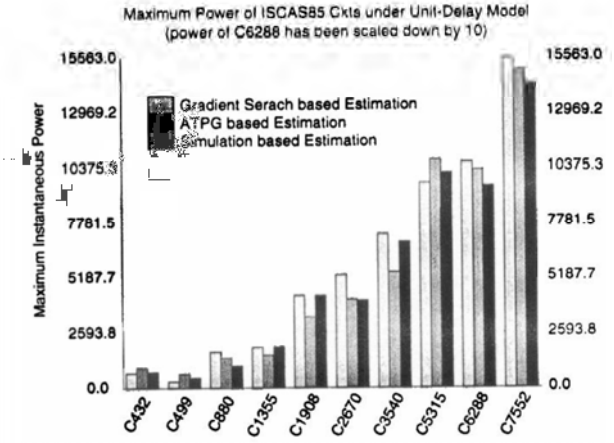
- For discrete-valued signal x, with n values:

$$H(x) = \sum_{i=1}^{n} p_i \log_2\frac{1}{p_i}$$

SEBC-L5 MZ 21

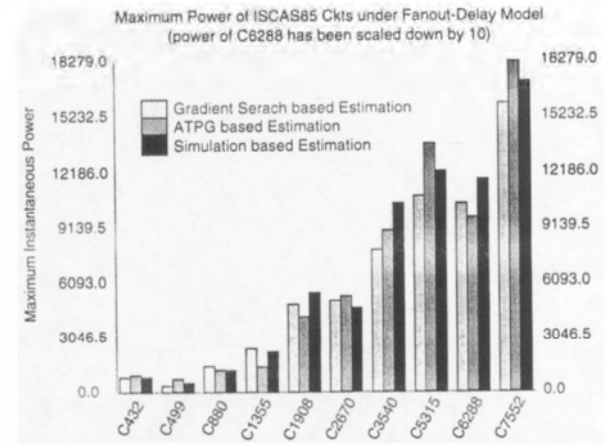
Mod by Giorgio Fissore, pag 85

ATPG vs. Gradient vs. Simulation Methods – Unit Delay Model



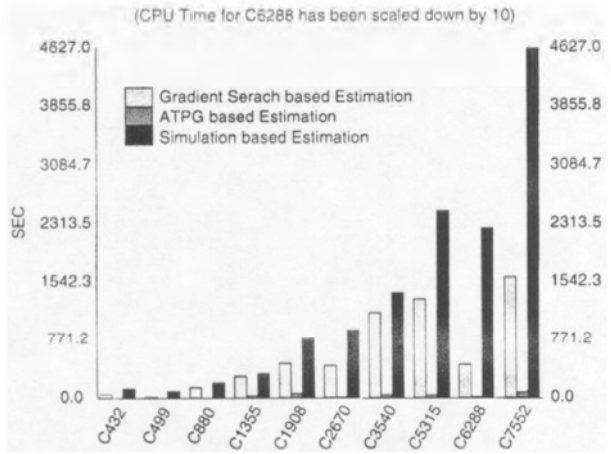
SEBC-L5 MZ 25

ATPG vs. Gradient vs. Simulation Methods – Fanout Delay Model



SEBC-L5 MZ 26

CPU Time Comparisons for Unit Delay Model Circuits



SEBC-L5 MZ 27

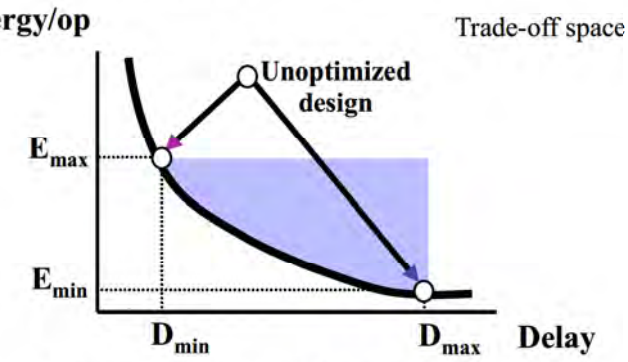
Mod by Giorgio Fissore, pag 87

Summary of Power Estimation

- Circuit-level and High-level power estimators exist
- Maximum power estimated with ATPG tools, using steepest-descent gradient descent, or genetic algorithms

Mod by Giorgio Fissore, pag 89

Energy-Delay Optimization and Trade-off



Maximize throughput for given energy or
Minimize energy for given throughput

Other important metrics: Area, Reliability, Reusability

SEBC-L6 MZ 4

Esempi di quanto posso ottimizzare sui vari livelli

The Design Abstraction Stack

A very rich set of design parameters to consider!
It helps to consider options in relation to their abstraction layer

.	System/Application	Choice of algorithm
.	Software	Amount of concurrency
70%	(Micro-)Architecture	Parallel versus pipelined, general purpose versus application specific
30%	Logic/RT	logic family, standard cell versus custom
.	Circuit	sizing, supply, thresholds
.	Device	Bulk versus SOI (silicon on insulator)
.		
3%		

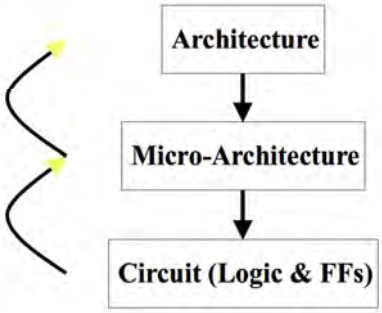
SEBC-L6 MZ 5

Posso cercare di ottimizzare il mio progetto su livelli differenti. Tendenzialmente, più vado verso l'alto, più ho risparmi significativi.

Anche l'algoritmo può essere modificato per ottimizzare i consumi (così come per le prestazioni).
Es. "probabilmente l'algoritmo che muove meno dati è quello che consuma di meno"

Noi non ci occuperemo solo di questo livello (si vedrà in altri corsi)

Optimization Can/Must Span Multiple Levels



Design optimization combines top-down and bottom-up:
"meet-in-the-middle"

SEBC-L6 MZ 6

Mod by Giorgio Fissore, pag 91

Reducing Active Energy @ Design Time

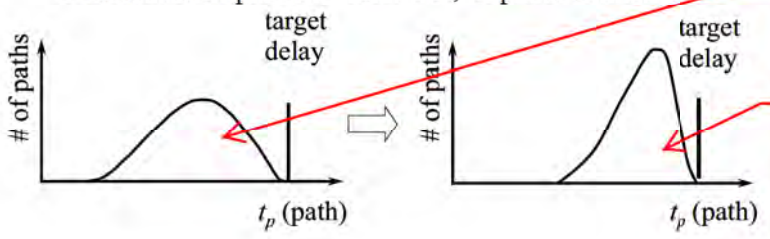
$$E_{active} \sim \alpha \cdot C_L \cdot V_{swing} \cdot V_{DD}$$
$$P_{active} \sim \alpha \cdot C_L \cdot V_{swing} \cdot V_{DD} \cdot f$$

- Reducing voltages
  - Lowering the supply voltage (V_{DD}) at the expense of clock speed
  - Lowering the logic swing (V_{swing})
- Reducing transistor sizes (C_L)
  - Slows down logic
- Reducing activity (α)
  - Reducing switching activity through transformations
  - Reducing glitching by balancing logic

SEBC-L6 MZ 10

Observation

- Downsizing and/or lowering the supply on the critical path lowers the operating frequency
- Downsizing non-critical paths reduces energy for free, but
  - Narrows down the path delay distribution
  - Increases impact of variations, impacts robustness



SEBC-L6 MZ 11

Prima del corso di low power avevo percorsi più lenti e percorsi più veloci distribuiti abbastanza uniformemente; la cosa importante era non superare un target delay

Ora invece cerchiamo di accumulare tutti i ritardi verso il limite; in questo modo, pur mantenendo lo stesso ritardo, ottimizzo la potenza (pensare alla curva potenza-ritardo)

Attenzione che spostare i ritardi sempre più vicino al target delay impatta anche la robustezza del circuito, poichè se un ritardo aumenta per qualche motivo, è ora più probabile che questo superi la soglia causando un malfunzionamento.

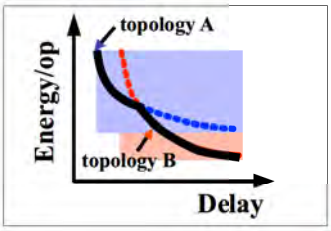
Circuit Optimization Framework

minimize	Energy (V_{DD}, V_{TH}, W)
subject to	Delay (V_{DD}, V_{TH}, W) $\leq D_{con}$

Constraints

$V_{DD}^{min} < V_{DD} < V_{DD}^{max}$
$V_{TH}^{min} < V_{TH} < V_{TH}^{max}$
$W^{min} < W$

- Reference case
  - D_{min} sizing @ V_{DD}^{max}, V_{TH}^{ref}



[Ref: V. Stojanovic, ESSCIRC'02]

SEBC-L6 MZ 12

Mod by Giorgio Fissore, pag 93

Library Gate Delay Model

- For every input terminal I_i and output terminal O_j of every gate:
  - $T^{i}_{i,j}(G)$ – fanout load independent delay (intrinsic)
  - $R_{i,j}(G)$ – additional delay per unit fanout load
- Total gate propagation delay from input to output:

$$T^{I}_{i,j}(G) + R_{i,j}(G)C_j(G)$$

  - Normalize all activities d_y by dividing them by clock activity (2f) - f is the clock frequency -
- Probability of rising or falling transition at y:

$$p_y^{\updownarrow} = d_y \qquad (4.27)$$

$$p_y^{\uparrow} = p_y^{\downarrow} = \tfrac{1}{2} p_y^{\updownarrow} \qquad (4.28)$$

SEBC-L6 MZ 16

Technology Mapping for Low Power

- Problem statement:
  - Given Boolean network optimized in a technology-independent way and a target library, bind network nodes to library gates to optimize a given cost
- Method:
  - Decompose circuit into trees
  - Use dynamic programming to cover trees
  - Cost function:

$$\text{Area}(g) + \sum_{n_i \in \text{inputs}(g)} \text{MinArea}(n_i) \qquad (4.23)$$

  - Traverse tree once from leaves to root

SEBC-L6 MZ 17

Extension for Low-Power Design

- Power dissipation estimate:

$$\text{Power} = \sum_{i=1}^{i=n} \frac{1}{2} V_{dd}^2 a_i C_i f \qquad (4.24)$$

- Estimate partial power consumption of intermediate solutions
- Cost function:

$$\text{power}(g, n) = \text{power}(g) + \sum_{n_i \in \text{inputs}(g)} \text{MinPower}(n_i) \qquad (4.25)$$

- power(g,n) =cost of choosing gate g to match at node n
- MinPower (n_i) is minimum power cost for input pin n_i of g
- power (g) = 0.5 f V_{DD}^2 a_i C_i
- Formulation:

$$\text{Minimize} \quad wP + (1 - w)R \quad \text{such that } T \le T_{max}$$

  - R = Total Area, w gives their relative importance; T = delay

SEBC-L6 MZ 18

Mod by Giorgio Fissore, pag 95

Calculating Transition Probability

- Hard to find p_{zi}^{l} (NP-hard)
  - Hard to determine prior state of internal circuit nodes
  - Assume that when state cannot be determined, a transition occurred (upper power limit)
  - More accurate bound: Observe that # conducting paths from node to V_{dd} must change from 0 to > 0 followed by similar change in # conducting paths to V_{ss}
    - Use # conducting paths that is smaller
    - $$p_{z_i}^{\uparrow} = \begin{cases} p_{z_i,V_{dd}}^{\uparrow} & \text{if } p_{z_i,V_{dd}}^{\uparrow} \le p_{z_i,V_{ss}}^{\uparrow} \\ p_{z_i,V_{ss}}^{\uparrow} & \text{otherwise} \end{cases}$$
  - Use serial-parallel graph edge reduction techniques

SEBC-L6 MZ 22

Transistor Reordering

- Already know delay of longest paths through each gate input from static timing analyzer
- Should (for NAND or NOR) connect latest arriving signal to input with smallest delay
  - Break gate inputs into permutable sets and swap inputs
  - Hard to compute which input order is best – can afford to enumerate all possible orderings and try them
    - Compute prob. (signal is switching while all other signals in permutable set are on) – gives maximum internal node C charging / discharging

SEBC-L6 MZ 23

Transistor Reordering: Pin Swapping

- Library gates generally have pins that are functionally equivalent.
- Permutations of these pins are possible.
- Internal capacitance ignored: assign high switching nodes to low input capacitance pins.
- Internal capacitance considered: must consider all possible permutations of assignments.
- Explicit enumeration is feasible (inputs in a cell typically less than 6-8)

SEBC-L6 MZ 24

Rispetto a prima stanno aumentando i vantaggi di porte a più ingressi.
Ora non sempre la cascata di porte è migliore rispetto alla singola porta a tanti ingressi.
Questo complica il pin swapping, poichè vengono usate sempre più spesso queste porte, ma lo rende sempre più importante, dato che gli ingressi sono equivalenti dal punto di vista logico, ma cambiano se si guardano i ritardi.
.
Queste modifiche fanno parte del livello di SCELTA DELLA TECNOLOGIA!

Mod by Giorgio Pissore, pag 97

Transistor Resizing Methods

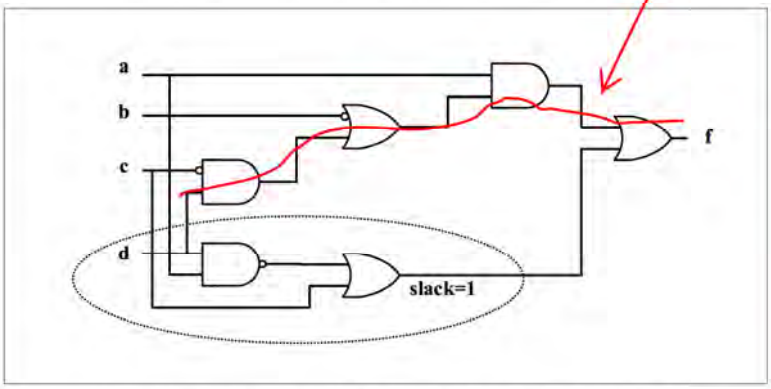
- Typical cell libraries have several instances of cells with different sizes.
- Replace some gates of the circuit with library gates having smaller area, but smaller capacitive load.
  Smaller gates are also slower....
- Idea:
  - Replace only gates not belonging to critical paths.
- Can be used to further improve the result of power-targeted technology mapping.

SEBC-L6 MZ 28

Questo metodo può portare a miglioramenti di prestazioni di qualche percento.

Critical path

Technology mapping



Larger gates reduce capacitance, but are slower

SEBC-L6 MZ 29

Come vado a lavorare?
individuo il percorso critico e vado a lavorare sugli altri percorsi, tendendo a portare anche loro sul limite di ritardo. (nel caso a fianco posso ridurre le dimensioni delle porte cerchiate).
Questo processo fa parte del TECNOLOGY MAPPING, la scelta della tecnologia.

La logica CMOS è per natura invertente. Come realizzare una NAND a quattro ingressi?

Technology Mapping: 4 input AND

Example: 4-input AND

- (a) Implemented using 4 input NAND + INV
- (b) Implemented using 2 input NAND + 2-input NOR

Gate type	Area (cell unit)	Input cap. (fF)	Library 1: High-Speed Average delay (ps)	Library 2: Low-Power Average delay (ps)
INV	3	1.8	$10.7 + 5.4\ C_L$	$12.0 + 6.0\ C_L$
NAND2	4	2.0	$10.3 + 5.3\ C_L$	$16.3 + 8.8\ C_L$
NAND4	5	2.0	$13.6 + 5.8\ C_L$	$22.7 + 10.2\ C_L$
NOR2	3	2.2	$7.0 + 3.8\ C_L$	$16.7 + 8.9\ C_L$

(delay formula: C_L in fF)
(numbers calibrated for 90 nm)

SEBC-L6 MZ 30

Mod by Giorgio Fissore, pag 99
NAND e NOR a due ingressi hanno dimensioni diverse!!

Buffer Insertion

- Selective insertion of buffers to reduce capacitance and transition times.
- Requires accurate timing information (i.e. fully mapped netlist).
- Example: for a NAND gate driving two FFs a buffer can be inserted to speed up the signal transition times.
  - Internal power without buffer:

NAND = 3 ; FF 5 x 2 ➔ Transition time = 4

  - Internal power with buffer:

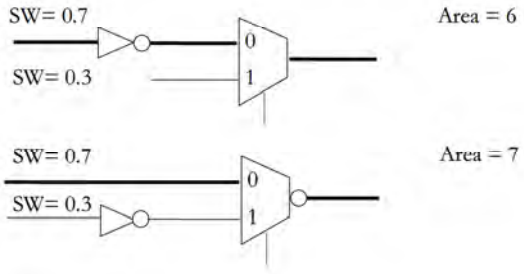
NAND = 2,5 ; Buffer = 1; FF=4,5 x 2 ➔ Transition time = 2

SEBC-L6 MZ 34

Esempio: immaginiamo di avere una NAND che deve pilotare due FFs.
.
La NAND (ottimizzata per un carico) ha un suo consumo (es 2.5 per un gate, 3 per due gate), ogni FF consuma 5 >> tot = 5*2+3=13, e transizioni lente
.
Risintesi: aggiungo un buffer (che consuma 1).
ora la NAND pilota solo più un carico, e il buffer è in grado di dare fronti più ripidi; grazie a questi il circuito va più veloce, e i FFs consumano meno, poichè viene ridotta la loro corrente di corto.

Phase Assignment

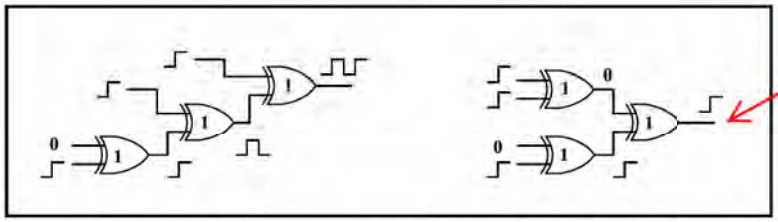
- Selectively reduce transitions on high-activity nets by assigning the most proper polarity.
- Trade off possible area increase for power



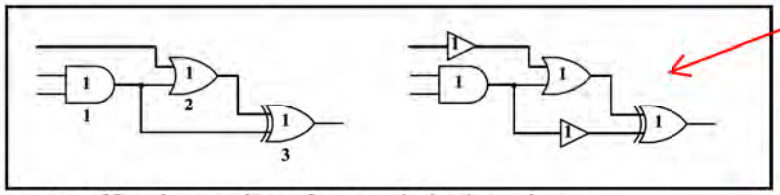
SEBC-L6 MZ 35

Esempio 2:
Guardando il caso a lato, se nella mia libreria ho anche un mux invertente (con consumi simili a quello non invertente), allora se sposto l'inverter sul secondo ingresso, ne più che dimezzo i consumi! (grazie alla minore Esw)

Logic Restructuring



Logic restructuring to minimize spurious transitions



Buffer insertion for path balancing

SEBC-L6 MZ 36

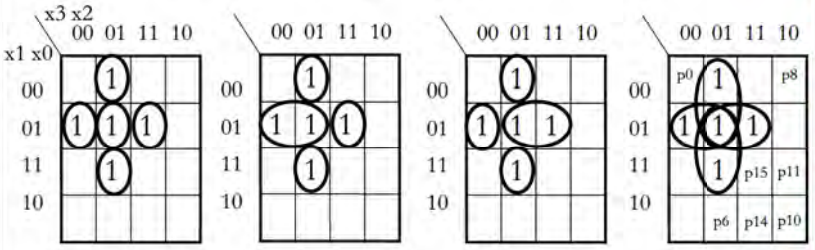
Altri esempi:

La soluzione ad albero è sempre meno sensibile ai glitch

Inserisco un buffer per rendere più bilanciati i percorsi (dal pt di vista dei ritardi)

Mod by Giorgio Fissore, pag 101

Impact of Signal probabilities



Esw(a) = p1(1-p1)+ p4(1-p4)+ p5(1-p5)+ p13(1-p13)+ p7(1-p7)

Esw(b) = Esw(a) – 2 p1 p5

Esw(c) = Esw(a) – 2 p5 p13

Esw(d) = Esw(a) – 2 p1 p5 – 2 p4 p5 – 2 p5 p7 – 2 p5 p13 +3 p5 (1- p5)

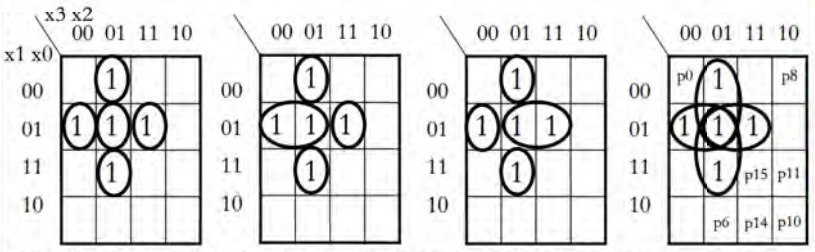
Cost(a)=20
Cost(b)=15
Cost(c)=15
Cost(d)=12

SEBC-L6 MZ 40

a) posso coprire i 5 uni con 5 implicanti con una somma di 5 termini>>avrei 5 AND a 4 ingressi che vanno dentro ad una OR. (se coprissi gli zeri, or in ingresso, and in uscita)
La formula Esw(a) riguarda proprio le 5 AND (si dovrebbe poi ancora aggiungere la Esw dell'uscita).
[il modo più semplice per etichettare gli implicanti è chiamarli con il numero ottenuto concatenando le coordinate, es 01-00 >> 4]

Da corsi precedenti si è imparato che mettere implicanti più grandi riduce i costi, diminuendo il numero degli ingressi.
Posso infatti in certe condizioni unire due gate:
b) i due uni hanno in comune la coordinata 01 >>unisco due AND a 4 ingressi e ne metto una AND a 3 ingressi (risparmio 5). in questo modo risparmio le comm per i casi in cui passo da sx a dx (p1p5) più quelli in cui passo dxsx (p5p1). la Esw della or di uscita non viene considerata in quanto costante.

Impact of Signal probabilities



- Esw(b) and Esw(c) are obtained by merging cubes into larger ones.
- Esw(b), Esw(c) < or = Esw(a) regardless of signal probabilities.
- Does finding larger cubes result in less activity? **In general, it does not!!!**
- The value of Esw(d) w.r.t. Esw(a) depends on signal prob.
- Example: If the input signals have all 0.5 probabilities, Esw(d) is higher than any of Esw(a), Esw(b) and Esw(c).

SEBC-L6 MZ 41

-Implicanti più grandi, usati per ridurre i costi, riducono anche la Esw!! (sia se copro i minterm -0- che i maxterm -1-)
-Mettere implicanti grandi che si sovrappongono (caso 4), abbassa i costi, ma porta generalmente ad un incremento della Esw (es per uscire dall'1 centrale, ci sono 4 commutazioni, per spostarsi dall'1 centrale ad un 1 a fianco, 3 commutazioni!!).
-fare sempre attenzione alle coperture multiple per la potenza
-tuttavia, sempre la combinazione 4, fornisce un output più immune da glitch; infatti passare da un implicante all'altro aumenta il rischio di glitch.
-la scelta di una config 3 o 4 ad esempio, deve avvenire in base all'impiego atteso (es se l'uscita dovesse spostarsi sempre sull'1, la 4 consuma meno magari) e alle specifiche (es no glitch).

Impact of Signal probabilities

- The cost function must be a function of signal probabilities.
- Merging cubes into larger cubes reduces switching.
  - Example: Covering vertex i and j to form a cube reduces Esw of *2 pi pj*.
- Covering a vertex more than once increases switching.
  - Example: Covering vertex with n cubes increases Esw of *(n-1) pi (1-pi)*
- Rules of thumb:
  - The larger the number of vertices that a merged cube can cover, the larger the power reduction.
  - Usually, covering a vertex more than once causes unnecessary power dissipation.

SEBC-L6 MZ 42

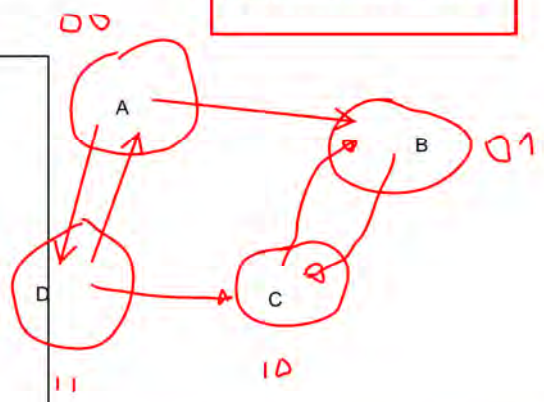
Mod by Giorgio Fissore, pag 103

Don't care minimization

- For each node, compute the (observability and satisfiability) don't cares.
- Compute the (global) node probability.
  Zero-delay models can be used for this computation.
  - Probability is at a desired value:
    Minimize the function using the don't cares with a traditional cost function.
  - Probability is not at a desired value:
    Minimize the function using the don't cares with modified cost function that selects the implicants with proper probability.

SEBC-L6 MZ 46

ES macchina a stati



Logic-Level Optimizations

Posso cercare di associare ad ogni arco la Prob di passare da quell'arco

SEBC-L6 MZ 47

FSM and Combinational Logic Synthesis

- Consider likelihood of state transitions during state assignment
  - Minimize # signal transitions on present state inputs V
- Consider signal activity when selecting best common sub-expression to pull out during multi-level logic synthesis
  - Factor highest-activity common sub-expression out of all affected expressions

Mod by Giorgio Fissore, pag 105

SEBC-L6 MZ 48

Relationship Between State Assignment and Power

- Hamming distance between states S_i and S_j:
  - H (S_i, S_j) = # bits in which the assignments differ
- Average Power: $\text{Power}_{avg} = \frac{1}{2} V_{DD}^2 \sum C_i D(i)$
  - D (i) = signal activity at node i
  - Approximate C_i with fanout factor at node i
- Average power proportional to:

$$\Phi = \sum_i \text{fanout}_i D(i)$$

SEBC-L6 MZ 52

Handling Present State Inputs

- Find state transitions (Si, Sj) of highest probability
- Minimize H (Si, Sj) by changing state assignment of Si, Sj
- Requires system simulation of circuit over many clock periods, noting signal values and transitions
- If one-hot design is used, note that H = 2 for all states
  - Impossible to obtain optimum power reduction
  - Uses too many flip-flops
- Optimization cost function:

$$\gamma = \sum_{\text{over all edges}} p_{ij} H(S_i, S_j)$$

SEBC-L6 MZ 53

La potenza sarà quindi proporzionale ad una variabile costo (gamma) costituita da una sommatoria su tutti gli archi, della prob di passare per quell'arco per la distanza di Hamming tra i due stati.

Codifica One Hot non ottimizzabile. Vediamo invece gli algoritmi di ottimizzazione per macchine con stati codificati.

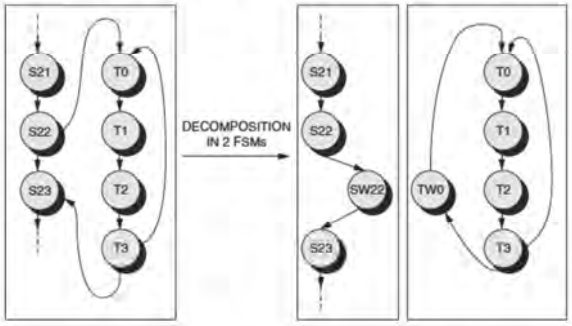
Simulated Annealing Optimization Algorithm

- Allowed moves:
  - Interchange codes of two states
  - Assign an unassigned code to a state that is randomly picked for an exchange
- Accept move if it decreases γ
- If move increases γ, accept with probability:
  $e^{-|\delta(\gamma)| / \text{Temp}}$
- where $|\delta(\gamma)|$ is the value of the change in the objective function and Temp is the temperature

SEBC-L6 MZ 54

Se c'è un solo minimo di consumo, gli algoritmi tirano a caso le variabili e vedono in che direzione mi sposto; se vado verso il minimo, tengo: "Algoritmi a Gradiente". Nelle funzioni in cui esistono minimi locali, però questi metodi non funzionano.
Qui si fa:
-se diminuisce potenza, accetta
-se aumenta, (posso accettare, ma riduco la probabilità di accettare per cercare un minimo più profondo?)>> metodo simulated annealing (molte iterazioni però)

FSM Decomposition



- Example: Complex FSM that executes a small subroutine (T0..T3):
- Simply SW22 and TW0 act as interface wait states between the two FSMs mutually exclusive.

SEBC-L6 MZ 58

Una delle due rimane congelata. Ovviamente conviene separare in punti per cui si parlano poco.

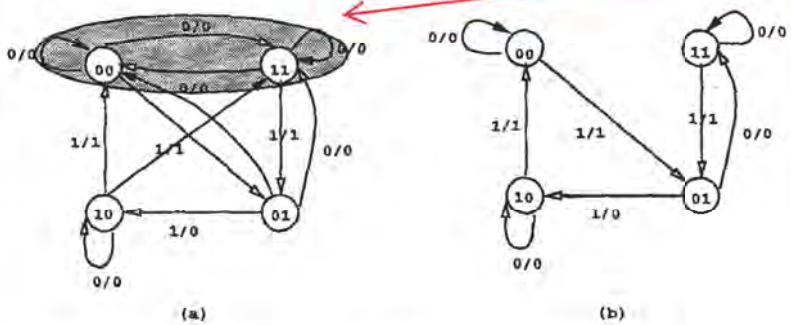
STG Restructuring

- *Idea*: Restructure the STG so that the switching activity on the state lines gets minimized.
- *Key assumption*: State equivalence.
- *Implementation*: Add and remove edges in the STG without changing the behaviour of the FSM.
- The method must be coupled with re-synthesis, since the restructured STG must be translated into logic. It yields 10% power savings; it is applicable only to small FSMs.

SEBC-L6 MZ 59

STG: State Transition Graph (pallogramma)

STG Restructuring



- (a) shows the augmented STG and (b) the reduced STG.
- The states in the shaded area are equivalent states, and the edges with *solid arrows* are the **ghost** edges.
- The reduced STG now has fewer bit changes per state transition than the original STG.

SEBC-L6 MZ 60

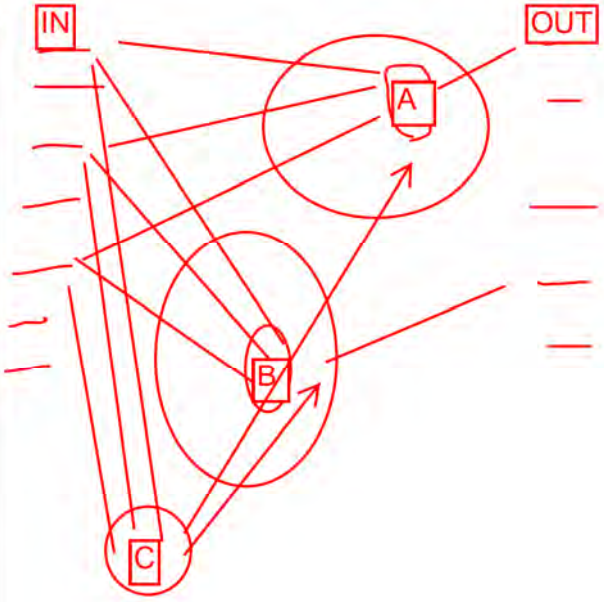
Il pallogramma originale ha solo gli archi bianchi.
Si può notare che gli stati 00 e 11 sono equivalenti visti dall'esterno >>riordinando evito commutazioni tra i due stati equivalenti
-Serve se sbagli il pallogramma

Mod by Giorgio Fissore, pag 109

Power-Driven Multi-Level Logic Optimization

- Use Berkeley MIS tool
  - Takes set of Boolean functions as input
  - Procedure kernel finds all cube-free multiple or single-cube divisors of each Boolean function
  - Retains all common divisors
  - Factors out best few common divisors
  - Substitution procedure simplifies original functions to use factored-out divisor
- Original criteria for selecting common divisor:
  - Chip area saving
- New criterion: power saving

SEBC-L6 MZ 64



Boolean Expression Factoring

- $g = g(u_1, u_2, \ldots, u_K)$, $K \geq 1$ is common sub-expression
- When g factored out of L functions, signal probabilities and activities at all circuit nodes are unchanged
- Capacitances at output of driver gates $u_1, u_2, \ldots, u_K$ change
- Each drives L-1 fewer gates than before
- Reduced power:

$$(L-1)V_{dd}^2 C_0 \sum_{k=1}^{K} n_{u_k} D(u_k)$$

- D (x) = activity at node x; C_0 = load C due to a fanout = 1 gate
- n_{ui} = # gates belonging to node g and driven by u_K (there are also gates not belonging to g also driven by u_K)
- Factored g drives the same # of nets of previous L instances of the same expression

SEBC-L6 MZ 65

E' probabile che succeda questa cosa: l'uscita numero 1, ha al suo interno una porta (A) funzione di tre ingressi.
esiste poi un'altra uscita che utilizza quella stessa porta con gli stessi ingressi (B).
Dal punto di vista della velocità, può convenire avere due repliche della stessa porta, ma non dal pt di vista della potenza.
Si fa quindi una cosiddetta "fattorizzazione", eliminando A e B e sostituendole con una porta C, funzione degli stessi ingressi che porta ad entrambe le uscite.
Il risparmio di potenza nasce da due ragioni:
-Sugli ingressi: al posto di pilotare L porte, ne devono pilotare una sola: ciascun ingresso ha un fan-out legato al numero di capacità che deve caricare. Con L repliche della porta, ho un risparmio di (vedi formula a lato) L-1 per la sommatoria degli ingressi per loro Esw.
-Sulle porte, poichè ciascuna sottofunzione consuma, ho di nuovo un risparmio di L-1 volte per il risparmio della sommatoria su tutti i nodi interni per la loro Esw

Factoring (continued)

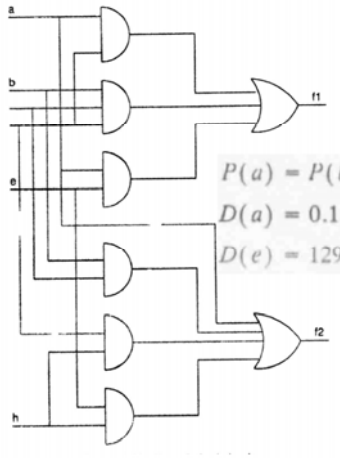
- Only one copy now of g instead of L copies
  - L-1 fewer copies of internal nodes $v_1, v_2, \ldots, v_m$ in factored-out hardware for switching and dissipating power
- Power saving: $(L-1)V_{dd}^2 C_0 \sum_{m=1}^{M} n_{v_m} D(v_m)$
- Total power saving:

$$\Delta W(g) = (L-1)V_{dd}^2 C_0 \left(\sum_{k=1}^{K} n_{u_k} D(u_k) + \sum_{m=1}^{M} n_{v_m} D(v_m) \right) \quad (4.21)$$

SEBC-L6 MZ 66

Mod by Giorgio Fissore, pag 111

Example Unoptimized Circuit



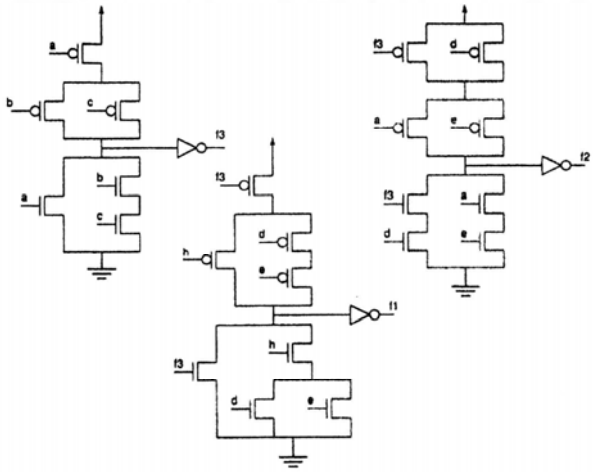
$P(a) = P(b) = P(c) = P(d) = P(e) = P(h) = 0.5$

$D(a) = 0.1, D(b) = 0.6, D(c) = 3.6, D(d) = 21.6,$

$D(e) = 129.6, D(h) = 3.6$

SEBC-L6 MZ 70

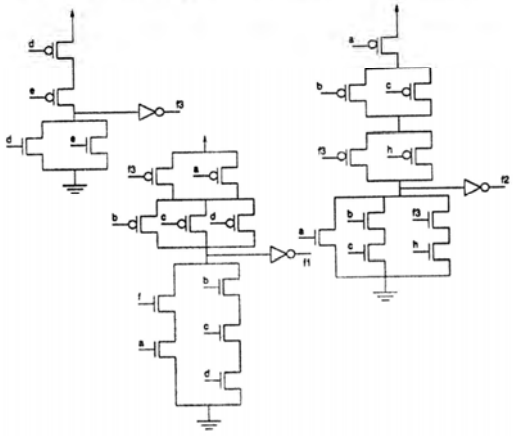
Optimization for Area Alone



SEBC-L6 MZ 71

Optimization for Low-Power Alone

- Large area but reduces power from 476.12 to 423.12



SEBC-L6 MZ 72

Mod by Giorgio Fissore, pag 113

Glitch Reduction and Pipelining

- In such cases, signals with a high switching rate should be mapped to reduce the levels of logic.
- Other paths with low switching rates can be optimized for area.
- The user needs to partition the design to separate high-switching and low-switching paths, while weigthing the benefits against the disadvantages.

SEBC-L6 MZ 76

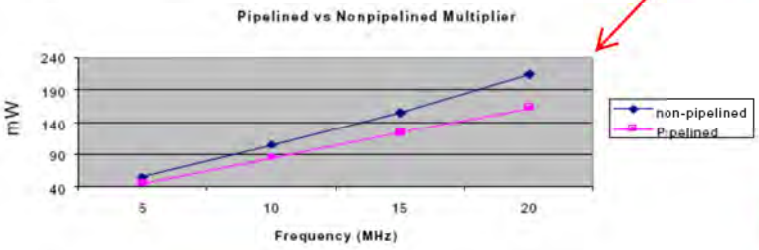
Glitch Reduction and Pipelining

- Pipelining is another technique that involves introducing the registers in the middle of long combinatorial paths.
- This adds latency but increases the speed and reduces the levels of logic.
- The introduction of extra registers consumes power but minimizes the glitches drastically.
- This glitch reduction has some power advantage.

SEBC-L6 MZ 77

Glitch Reduction and Pipelining

- For example, a pipelined 16x16-bit unsigned multiplier generated from ACTGEN (Actel's macro-generation utility) consumes less power than its unpipelined counterpart



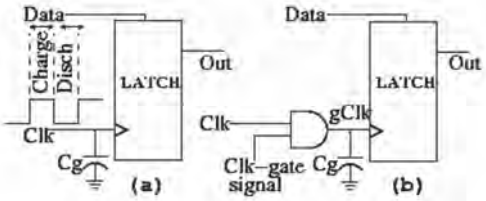
Esempio vecchio, ma calzante, che mostra la riduzione dei consumi di un moltiplicatore tramite la pipeline.
Si paga, naturalmente, il retiming.

Mod by Giorgio Fissore, pag 115

SEBC-L6 MZ 78

Clock gated latch (b)

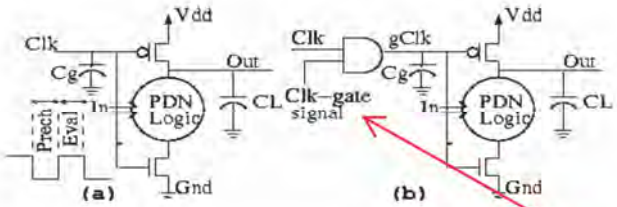
- The clock is gated by ANDing it with a control *Clk-gate signal.*
- When the latch is not required to switch state, *Clk-gate signal* is turned off and the clock is not allowed to charge/discharge Cg, saving clock power. Because the AND gate's capacitance itself is much smaller than Cg, there is a net power saving.



SEBC-L6 MZ 82

Dynamic logic cell (a)

- Cg is the effective gate capacitance.
- CL also consumes power: at the *precharge* phase of the clock, CL charges through the PMOS pre-charge transistor and during the *evaluate* phase, it discharges or retains value depending on the input to the pull-down logic.
- Whether CL consumes power or not, depends on *both* the current input and previous output.

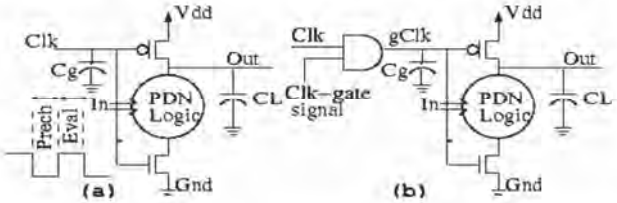


SEBC-L6 MZ 83

Vediamo ora cosa succede in logica dinamica:
"rete di pd collegata a massa dal clk, clk = 0, viene caricata Cl a Vdd, clk ad 1, viene attivata la rete di pd, l'uscita si carica o scarica a seconda della funzione booleana."
Qui abbiamo un motivo in più per adottare il clk_gating, poichè se prendo un latch statico con clk, consumo solo sull'ingresso (l'uscita non commuta). Nella logica dinamica invece, se l'uscita del mio latch deve essere a zero, con clk attivo, consumo, non solo sul clk interno, ma anche sull'uscita (carica - valuta - carica - valuta...)

Dynamic logic cell (b)

1. If CL holds a "1" at the end of a cycle, and the next cycle output evaluates to a "1", CL does not consume any power.
   - Precharging an already-charged CL does not consume power.
2. If CL holds a "0" at the cycle end, CL consumes precharge power, *irrespective* of what the inputs are in the next cycle.
   - Even if the input does not change, this precharge power is consumed. If the next output is a "1", no discharging occurs; otherwise, more power is consumed in discharging CL.



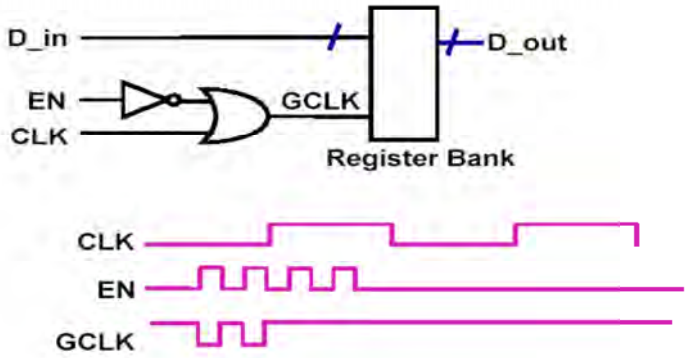
SEBC-L6 MZ 84

Al posto della AND posso bloccare il clk anche con una OR, bloccando così a piacere il clk a 0 o ad 1.
Mentre questo non faceva differenza in logica statica, qui ha invece un effetto.
Cl è infatti una capacità molto piccola, che quindi ha delle perdite; se blocco il clk in fase di valutazione, creo un percorso verso massa che la scarica pian piano perdendone l'informazione.
>>Devo sempre bloccare CLK in fase di precarica (ad 1)

Mod by Giorgio Fissore, pag 117

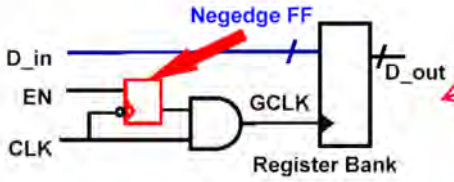
OR-Based Clock Gating

- Glitches in enable signal appear at clock

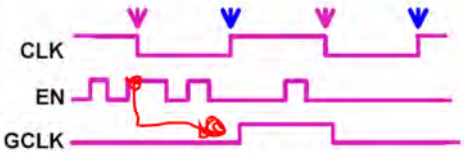


SEBC-L6 MZ 88

FF-based clock gating



- Enable captured at negative edge
- Gated-clock produced at following positive edge



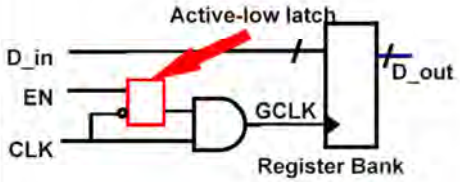
Half clock cycle wasted on enable path!

SEBC-L6 MZ 89

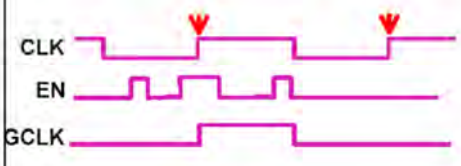
Campiono l'EN su tutti i fronti di discesa quando clk scende campiono e decido se far passare il prossimo fronte.
PRINCIPIO:
sulla fase bassa campiono clk
sulla fase alta campiono en

Il segnale di EN però in questo modo ha solo mezzo colpo di clk per essere generato, perchè l'EN deve essere stabile prima del fronte di discesa, sennò non viene campionato; e quindi rischierebbe di limitare la velocità.
Risolvo il problema sostituendo il FF con un LATCH.

Latch-based clock gating



- Structure behaves like a master-slave, which captures enable signal at posedge of clock

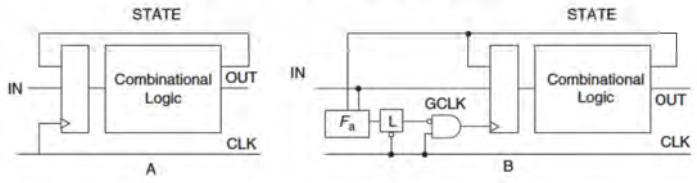


SEBC-L6 MZ 90

Con un Latch attivo a zero, quando il clk è basso, il latch è trasparente; possono passare anche i glitch, ma ciò non è un problema poichè essendo clk = 0, la porta AND lo blocca subito.
Quando clk sale, l'ultimo valore dell'enable viene salvato!
Qui ho un periodo intero e l'EN gioca lo stesso ruolo di tutte le altre uscite della macchina a stati.
Per questo si usa questa tecnica in tutti i meccanismi di Clk_gating

Mod by Giorgio Fissore, pag 119

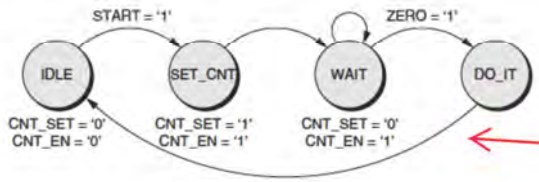
FSM clock gating



- During Idle conditions of FSMs instead of looping in current state, clock is gated and woken up only when selective INput will be active (10 to 30% power saving)
- The basic idea of gated-clock FSM is that it is not useful to have switching activity in the next-state logic or to distribute the clock if the state register will sample the same vector

SEBC-L6 MZ 94

FSM clock gating



- Example of a FSM that interacts with a timer counter to implement a very long delay of thousands of clock cycles before executing a complex but very short operation (in the DO_IT state).
- We can use the clock-gating techniques to freeze the clock and the input signals as long as the ZERO flag from the time-out counter is not raised

SEBC-L6 MZ 95

Dopo che è arrivato lo start, quando entro nel wait, spengo il clk e lo tengo spento, fino a quando non arriva il segnale di fine conta.

System Clock Gating

- Clock gating can be applied to systems, subsystems and blocks.
- Sometimes it is advisory to apply it at a lower level.
- For instance, in pipelined superscalar uP, it may be used dynamically…. Why?
- Pipeline activity should be monitored to activate or shut down the different functional units…

SEBC-L6 MZ 96

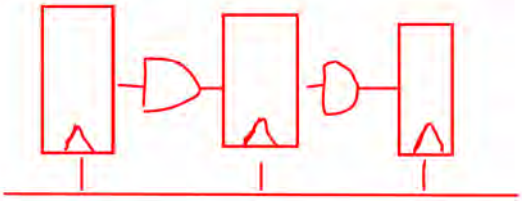
Se ho una macchina pipelinata, tipicamente la sua evoluzione è data da "registri-logica_comb-registri-logica_comb" ed in particolare ci sono più registri.

Mod by Giorgio Fissore, pag 121

DCG Superscalar uP Clock Gating

- *Deterministic clock gating (DCG)* : for many of the pipeline stages, a circuit block's usage in a specific cycle in the near future is *deterministically* known a few cycles ahead of time.
- DCG exploits this advance knowledge to clock-gate the unused Blocks: Exec. units, pipeline latches of back-end stages after issue, L1 D-cache wordline decoders, etc.
- In an out-of-order pipeline, whether these blocks will be used is known at the *end of issue* based on the instructions issued.
- There is *at least one cycle* of register read stage between issue and the stages using exec. units, D-cache wordline decoder and the back-end pipeline latches.
- DCG exploits this one-cycle advance knowledge to clock-gate the unused blocks without impacting the clock speed.

SEBC-L6 MZ 100



Superscalar uP Clock Gating

- DCG's deterministic methodology has three key advantages over PLB's predictive methodology:

1. PLB's ILP prediction is not 100% accurate:
   - If the predicted ILP is lower than the actual ILP, PLB ends up clock-gating useful blocks and incurs performance loss.
   - If the predicted ILP is higher than the actual ILP, PLB leaves unused blocks not clock-gated and incurs lost opportunity.
   - In contrast, DCG *guarantees* no performance loss and no lost opportunity for the blocks whose usage can be known in advance.

SEBC-L6 MZ 101

Attenzione, chiudere una pipe va fatto con attenzione, poichè magari ci sono ancora dei dati che viaggiano dentro; questo problema non si presenta nel DCG.

Superscalar uP Clock Gating

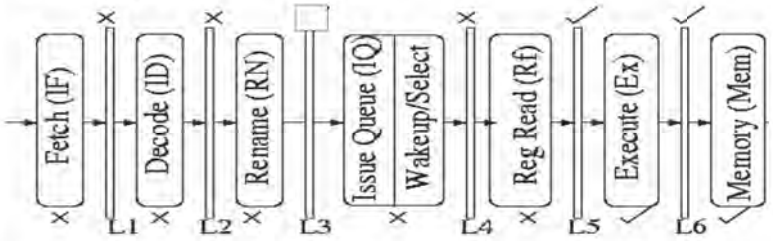
2. PLB's clock-gating granularities (circuit and time) are coarse; circuit granularity is a cluster (i.e., *many* back-end stages from register read through writeback are considered *together*). Time granularity is a 256-cycle window (i.e., clusters stay clock-gated for 256-cycle windows).
   - In contrast, DCG clock-gates at finer granularities of a few (1-2) cycles and smaller circuit blocks (execution units, D-cache address decoders, and pipeline latches).
   - Because DCG's blocks are still substantially larger than the few gates added for clock gating, DCG amortizes the overhead. PLB's coarser granularity makes it less effective and less flexible than DCG, which is a general technique applicable to non-clustered microarchitectures.
3. DCG uses no extra heuristics and is significantly simpler.

SEBC-L6 MZ 102

Mod by Giorgio Fissore, pag 123

DCG for pipeline latches

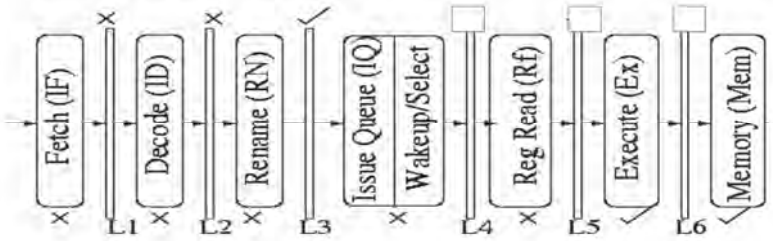
- Only the first three instructions enter the rename stage.
- Hence, we can determine the number of instructions that will enter the rename stage at the end of decode and clock-gate the unnecessary parts of the rename latch.
- We have the entire rename stage to setup the clock-gate control of the rename latch.



SEBC-L6 MZ 106

DCG for pipeline latches

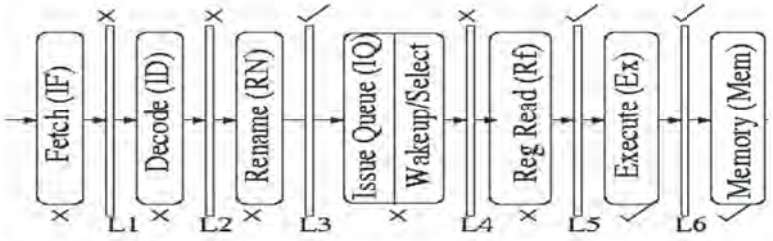
- Because we can identify which and how many instructions are selected to issue only at the very end of issue, we do not have enough time to clock-gate the issue latch.
- We can clock-gate the latches for the rest of the pipeline stages (i.e., register read (Rf), execute (Ex), memory access (Mem) and writeback (WB)).



SEBC-L6 MZ 107

DCG for pipeline latches

- At the beginning of each of the stages we know how many instructions are entering the stage, and we can exploit the time during the stage to set up the clock-gate control for these latches.

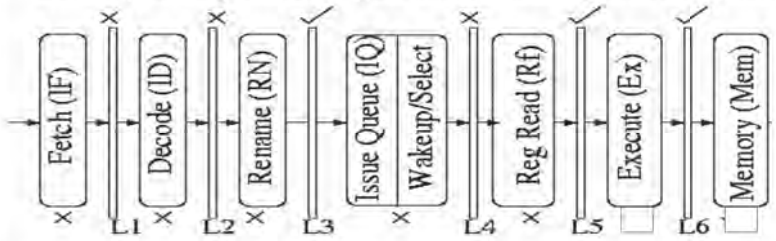


SEBC-L6 MZ 108

Mod by Giorgio Fissore, pag 125

DCG for pipeline stages

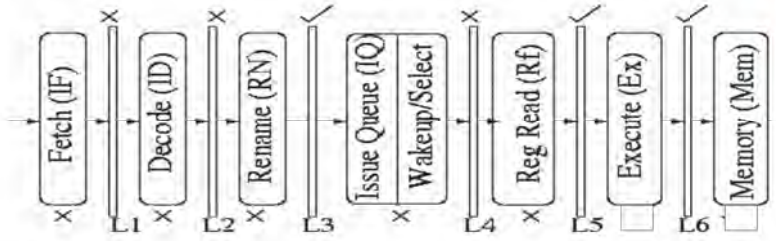
- Execution units can be clock gated, since are often implemented with dynamic logic for high performance.
- Based on the instructions issued, we deterministically know at the end of issue which unit is going to be used in the cycle after the register read stage.
- Hence, we can clock gate the rest of the unused execution units, by setting the clock gate control during the read cycle.



SEBC-L6 MZ 112

DCG for pipeline stages

- Caches use dynamic logic for wordline decoding and the writeback stage uses result bus driver to route data to regfile.
- Instructions that enter the execute stage go through the memory and writeback stages.
- We can use the same execute clock gate control (delayed by 1 or 2 ck cycles) to clock-gate the relevant logic in these stages.



SEBC-L6 MZ 113

Se ho capito bene questo accade perchè se disabilito una parte della Ex, allora so che nei clk seguenti quelle parti non dovranno scrivere in memoria o nel Rf e con gli stessi segnali di clk_gate disabilito anche parti di Rf e Memoria (non sicuro per Rf dato che c'è una x nel disegno a lato) -il writeback scrive nel Rf??

DCG for execution stage

- At the end of instruction issue, we know which execution units will be used in the execute stage, a few cycles into the future.
- The selection logic in a conventional issue queue not only selects which instructions are to be issued based on execution unit availability, but also matches instructions to execution unit.
- Hence, we leverage the selection logic to provide information about which execution units will remain unused and clock-gate those units.

SEBC-L6 MZ 114

Mod by Giorgio Fissore, pag 127

DCG for execution stage

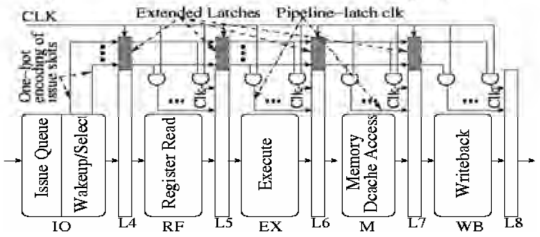
- If execution units keep toggling between gated and non-gated modes, the control circuitry keeps switching, resulting in an increased overhead due to the power consumed by control.
- Current charging and discharging may also cause large di/dt noise in the supply line.
- To alleviate these problems, among the execution units of the same type, we statically assign priorities to the units, so that the higher-priority units are always chosen to be used before the lower priority units (*sequential priority policy*).
- Thus, most of the time the (lower-) higher-priority units stay in (gated) non-gated mode, minimizing the power overhead.
- This *sequential priority policy* is easy to implement and does not affect overall performance.

SEBC-L6 MZ 118

Attenzione però che questa tecnica richiede l'uso di altri FF, e altri segnali di EN, per gestire questo controllo del gating, che possono limitare il risparmio.
Devo quindi cercare di ridurre anche la switching activity di questi segnali.
Allora, in funzione del carico, cerco di assegnarlo a partire dall'unità "più sfortunata" a quella "più fortunata".
Quindi, se ad esempio ho 4 unità che possono gestire il gating, le ordino e se devo bloccare qualcosa, lo faccio fare all'unità più a basso livello; se lei è occupata lo passo a quello sopra, e così via (quindi magari la prima lavora il 100% del tempo, la seconda il 70%, la terza il 50% e la quarta quasi mai)>>meno Esw.

DCG – pipeline stages

- We clock-gate pipeline latches at the end of rename, register read, execute, memory and writeback stages.
- For rename, the number of clock-gated latches in any cycle is known from the decode stage in the previous cycle.
- For latches in the other stages, the number of clock-gated latches in any cycle is known from the issue stage.
- We augment the issue stage to generate a one-hot encoding of how many instructions are issued every cycle.

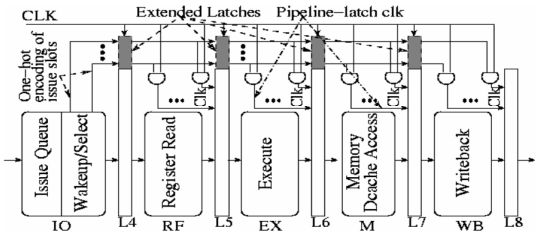


SEBC-L6

Ognuna di queste può essere spenta se il token che viaggia è arrivato al punto giusto.
Posso quindi avere un controllo totale e deterministico (quindi sicuro), sulla mia macchina, che, se è possibile, da sicuramente un risparmio e assicura di non aver perdite in prestazioni.

DCG – pipeline stages

- The encoding has a "0" for an empty issue slot, and a "1" for a full issue slot for an issued instruction, for all the issue slots.
- Much like the execution units, the one-hot encoding is passed down the pipe via extended pipeline latches.
- The outputs of the extended latches carrying the one-hot encoding are AND'ed with the clock line to generate a set of gated clock inputs for pipeline latches corresponding to individual issue slots.



SEBC-L6 MZ 120

Mod by Giorgio Fissore, pag 129

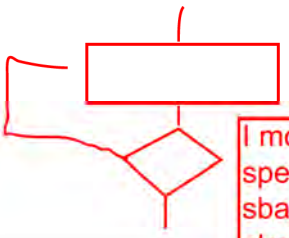
Fetch Gating

- • Wrong-path fetching wastes power
  - – Mispredicted branch causes fetching of wrong-path instructions
  - – Power is consumed fetching/executing these instructions
  - – Then they are discarded
- • Problem worsens with deeper pipeline, larger window, wider fetch
  - – Takes more cycles to resolve mispredicted branch
  - – Fetch/execute more wrong-path instructions

SEBC-L6 MZ 124

Fetch Gating

- • Use branch confidence mechanism to gate fetch
  - – If branch prediction not confident, stall fetch unit until branch resolves
  - – Large power savings with low to moderate performance degradation



SEBC-L6

Abbiamo già visto come nella pipe si inizi ad eseguire istruzioni prima che quelle precedenti siano terminate, e questo può dare problemi nei salti. Abbiamo già analizzato tutti i trucchi per minimizzare le perdite in prestazioni, vediamo ora come minimizzare le perdite in potenza, dovute al fatto di seguire una strada che poi non prendo (nel frattempo ho già eseguito istruzioni che poi consumano).
Questo diventa sempre più influente quanto più la pipe diventa profonda.

I moderni uP sono in grado di riconoscere i loop; e i loop tra l'altro spesso hanno un numero fisso di cicli, e quindi si può arrivare a non sbagliare poichè si sa dopo quanti giri si esce dal loop scegliendo l'altra strada.
Attenzione, quando ci si chiede se si salterà o no, posso anche ricevere come risposta "non lo so", e quindi decidere se preferire le prestazioni (riempio cmq la pipe) o la potenza (non la riempio e aspetto).

Fetch Gating

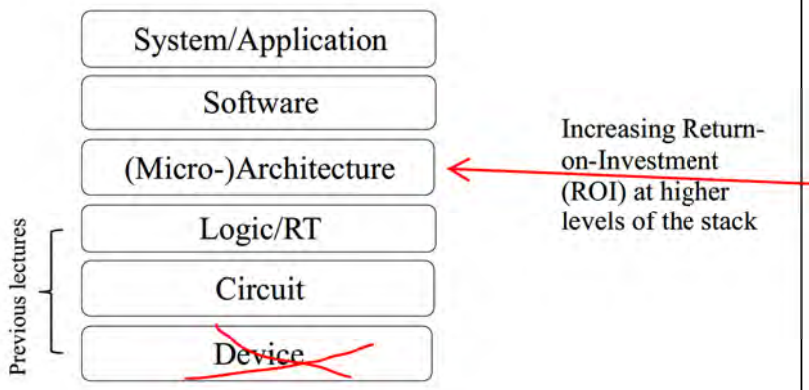
- • Scenarios
  - – *Confident* / *Correct*: Best case
  - – *Unconfident* / *Incorrect*: Save power without degrading performance
  - – *Confident* / *Incorrect*: Lost opportunity to save power without degrading performance
  - – *Unconfident* / *Correct*: Degrade performance with no power savings (stall unnecessarily)

SEBC-L6 MZ 126

L'unità di branch prediction viene potenziata sempre di più e diventa sempre più importante con pipe lunghe (es con una pipe a 25 stadi, avendo al limite un salto ogni 25 istruzioni, rischio di buttarne sempre via 24, con una brutta predizione).

Mod by Giorgio Fissore, pag 131

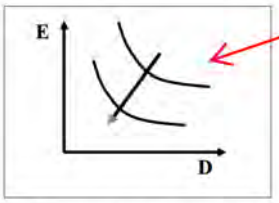
The Design Abstraction Stack



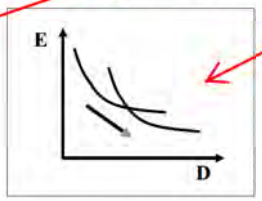
Ora guardiamo questo livello

SEBC-L7 MZ 4

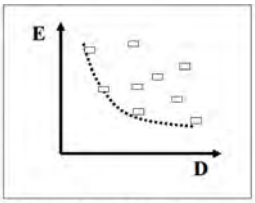
Expanding the Playing Field



Removing inefficiencies (1)



Alternative topologies (2)



Discrete options (3)

Architetture "strict better" (a meno che ad esempio non riduco consumi e aumento prestazioni, ma magari uso più area

Architetture diverse da scegliere in base al punto di lavoro desiderato

Architecture and system transformations and optimizations reshape the E-D curves

SEBC-L7 MZ 5

Alternative topologies: Shift register

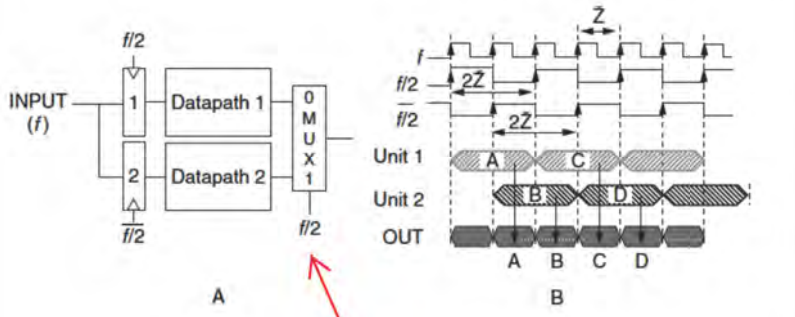
- Instead of a N bits shift register at fck ..... it is possible to derive a two rows N/2 bits shift registers with a terminal MPX..... clocked at fck/2 (one chain is p.e.t. and the other one is n.e.t.)
- Same throughput.... but .... each stage doubled the available time..... between two consecutive clock sampling.
- As a consequence it is possible to decrease power supply without penalties.
- Small increase in total Capacitance (only mpx... load...) but a consistent power reduction!

NOTA 1, 26/03

Mod by Giorgio Fissore, pag 133

SEBC-L7 MZ 6

Redesigned a Parallel Implementation



L'uscita viene prelevata a fasi alterne del clk

SEBC-L7 MZ 10

Redesigned a Parallel Implementation

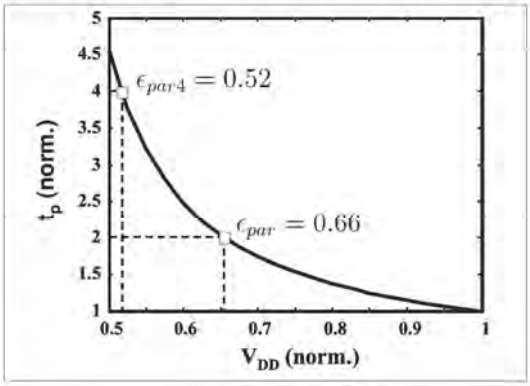
TABLE 10.1 8-bit Adder Power Simulation With the CoolChip Library

2 µm Technology	F	V_{dd} (V)	Power	%
8-bit adder	f = 7 MHz	4.5	540 µW	100
2-// 8-bit adder	f/2 = 3.5 MHz	4.5	760 µW	140
2-// 8-bit adder	f/2 = 3.5 MHz	3.0	339 µW	63
2-// 8-bit adder	f/2 = 3.5 MHz	2.5	235 µW	44

Più del 50% di risparmio!!

SEBC-L7 MZ 11

Example: 90nm Technology



Assuming $ov_{par} = 7.5\%$

$$P_{par} = 0.66^2 \cdot \frac{2.15}{2} \cdot P_{ref} = 0.47 P_{ref}$$

$$P_{par4} = 0.52^2 \cdot \frac{4.3}{4} \cdot P_{ref} = 0.29 P_{ref}$$

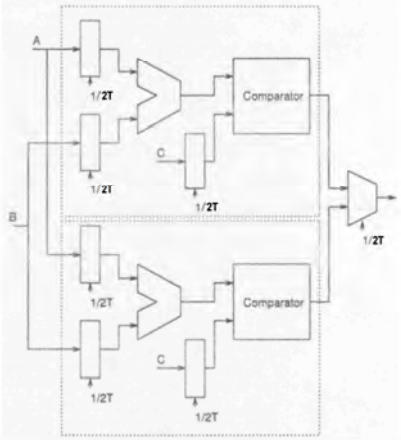
Cmq ricordarsi che l'area è proporzionale alla parallelizzazione, e se aumento il numero di oggetti, vado anche ad aumentare il consumo di potenza di leackage, che ora non stiamo considerando. Andrà cercato un compromesso.

Mod by Giorgio Fissore, pag 135

SEBC-L7 MZ 12

Redesigned Parallel Implementation > 2 X Area Increase

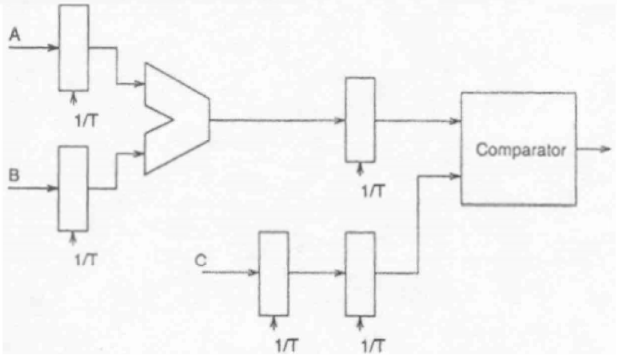
$$P_{par} = (2.15C)(0.58V)^2(0.5f) \approx 0.36P \qquad (4.19)$$



SEBC-L7 MZ 16

Redesigned Pipelined Implementation

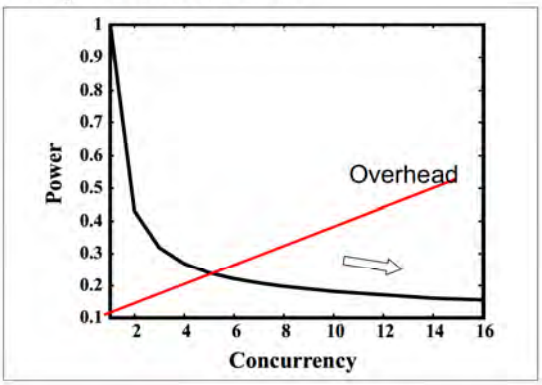
$$P_{pipe} = (1.15C)(0.58V)^2(f) \approx 0.39P \qquad (4.20)$$



SEBC-L7 MZ 17

Increasing use of Concurrency Saturates

- Can combine parallelism and pipelining to drive V_{DD} down
- But, close to process threshold, overhead of excessive concurrency starts to dominate



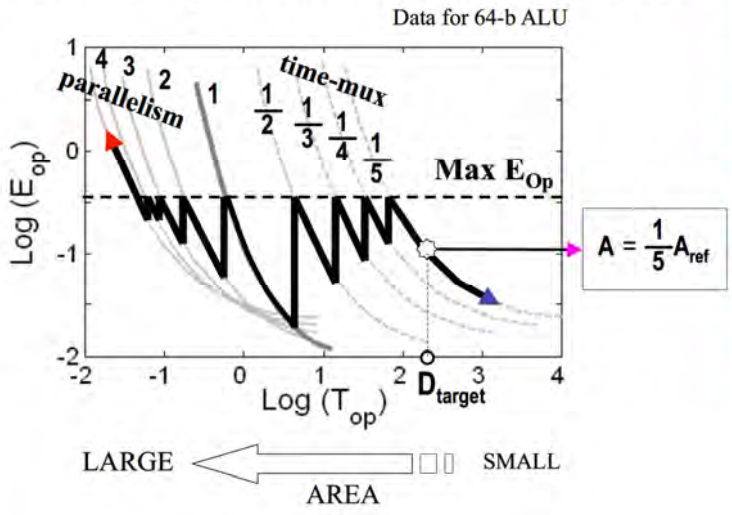
Assuming constant % overhead

SEBC-L7 MZ 18

Cosa stiamo facendo? stiamo aumentando la concorrenza interna.
All'aumentare di questa, la potenza scende con un andamento tipo quello in figura.
Vediamo come passando da 1 a 2 la riduzione è molto grande, da 2 a 4 ancora significativa, e dopo trascurabile.
Va inoltre considerato l'overhead che cresce linearmente con la concorrenza

Mod by Giorgio Fissore, pag 137

Concurrency and Multiplexing Combined



SEBC-L7 MZ 22

Posso spostarmi sia nel mondo della concorrenza che nel mondo del demultiplexing, in base a throughput richiesto e alla potenza da consumare.
Posso scegliere l'architettura migliore in base a ritardo o consumo.
Diventa quindi fondamentale valutare la potenza in fase di progetto, poichè questa può portare a diverse soluzioni architetturali.
Perciò in base al fatto che il datapath sia il punto critico o meno, mi gioco concorrenza o demux.

Some Energy-Inspired Design Guidelines

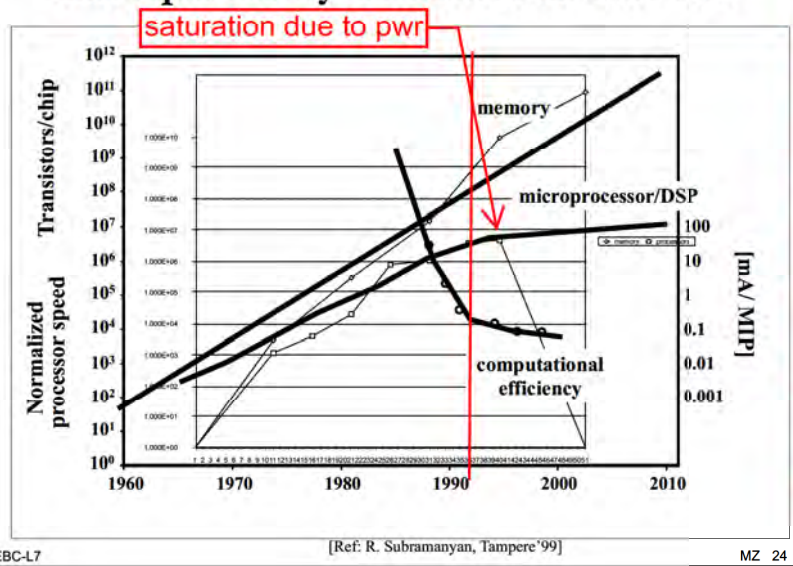
- **For maximum performance**
  - Maximize use of concurrency at the cost of area
- **For given performance**
  - Optimal amount of concurrency for minimum energy
- **For given energy**
  - Least amount of concurrency that meets performance goals
- **For minimum energy**
  - Solution with minimum overhead (that is – direct mapping between function and architecture)

Non è il vincolo generalmente richiesto (tranne "formula 1")

Non preoccupandoci di niente altro, abbassiamo tutto al minimo e non vogliamo overhead!

SEBC-L7 MZ 23

Concepts Slowly Embraced in Late 90's



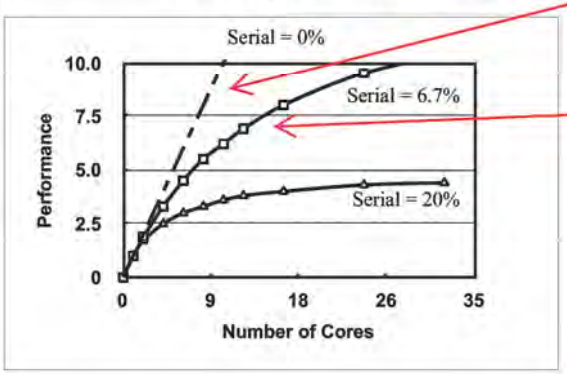
[Ref: R. Subramanyan, Tampere'99]

SEBC-L7 MZ 24

-Una volta, lo sviluppo era puramente tecnologico: nuova tecnologia, va al doppio della frequenza! meglio! (ank se consuma il doppio).
-Da dopo la metà degli anni ottanta, ci si è preoccupati anche di aumentare l'efficienza (computational efficiency)
-Negli anni 90 (anche con l'avvento del mondo mobile) la potenza ha iniziato a non essere più gestibile.
-si è iniziato ad usare anche la pipeline, ma è saturata anche quella.

Mod by Giorgio Fissore, pag 139

The Quest for Concurrency



Amdahl's Law: $Speedup = \frac{1}{Serial + \frac{1-Serial}{N}}$

MZ 28

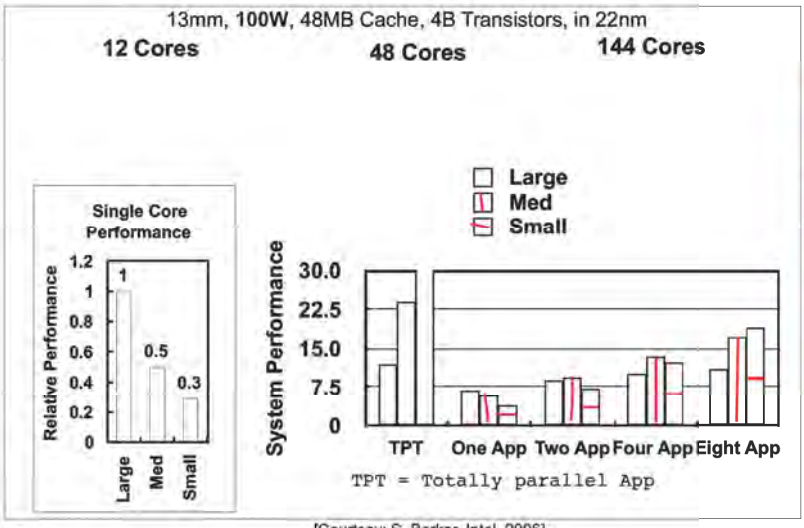
In teoria aumentando il numero di processori dovrebbe dare questo incremento (lineare) di performance

In realtà, però se faccio girare un'applicazione su due processori, il vero incremento è limitato dalle data dependencies

Se invece di un processore, ne uso N, e la percentuale di operazioni seriali (e quindi non parallelizzabili) è quella, ho un incremento di prestazoni dato da 1 fratto il numero di seriali più il numero di paralleli diviso N.
Potrei dover cambiare l'algoritmo in maniera che si adatti meglio a lavorare in sequenziale.

Serial: parametro introdotto dagli informatici, che rappresenta la percentuale di operazioni seriali (non parallelizzabili)

The Quest for Concurrency



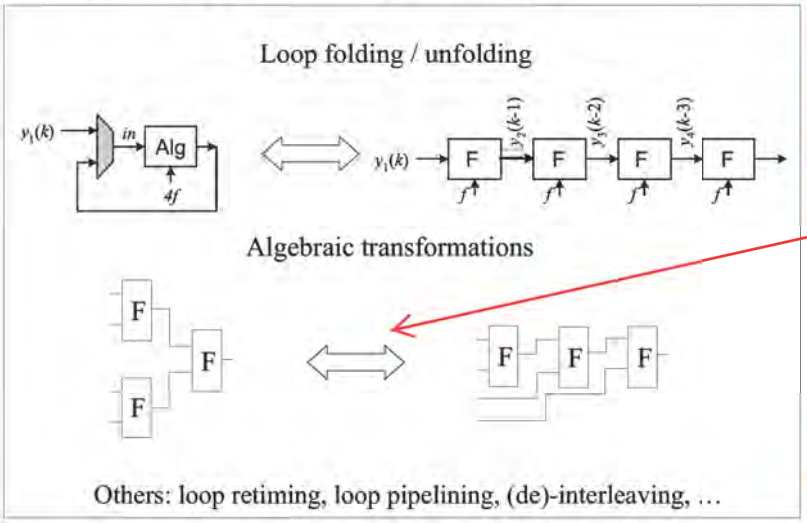
[Courtesy: S. Borkar, Intel, 2006]

SEBC-L7

MZ 29

Posso decidere di mettere core più grandi o più piccoli, ad area fissa.
Core grandi, maggiore prestazione/core, ma ridotto numero.
Core piccoli, minore prestazione/core, ma numero di core più elevato.
In base al parallelismo dell'applicazione, una soluzione large, med, small può essere preferibile rispetto alle altre.

Manipulating Concurrency Through Transformations



[Ref: A, Chandrakasan, TCAD'95; D. Markovic, JSSC'07]

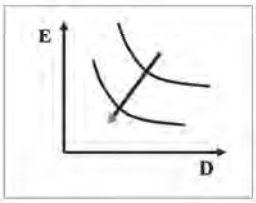
SEBC-L7

MZ 30

La concorrenza dell'applicazione, in genere è limitata dai loop.
Si fa quindi un loop folding o unfolding: srotoliamo la nostra applicazione, cercando di eliminare o ridurre i loop.
Pago che aumento l'HW, al posto che avere un solo componente che ripete tre volte l'operazione, ne metto tre in serie, potendo così pipelinare facendoli lavorare in parallelo.

Mod by Giorgio Fissore, pag 141

Improving Computational Efficiency



Implementations for a given function maybe inefficient and can often be replaced with more efficient versions **without penalty in energy or delay**

Inefficiencies arise from:

- Over-dimensioning or over-design
- Generality of function
- Design methodologies
- Limited design time
- Need for flexibility, re-use and programmability

SEBC-L7 MZ 34

-Magari ho bisogno di un incrementatore (fa A + 1) e ci metto un sommatore che ha sempre 1 in un ingresso.
Ho quindi messo un componente sovradimensionato, poichè fa operazioni più generali di quelle che deve fare.
-O magari ho già fatto la butterfly, poi gli devo abbassare il parallelismo, ma decido di non cambiare i componenti, sovradimensionato >> consuma di più.

Improving Computational Efficiency

Some simple guidelines:

- Match computation and architecture
  - ★Dedicated solutions superior by far
- Preserve locality present in algorithm
  - ★Getting data from far away is expensive
- Exploit signal statistics
  - ★Correlated data contains less transitions than random data
- Energy on demand
  - ★Only spend energy when truly needed

SEBC-L7 MZ 35

LINEE GUIDA PER IL LOW POWER

Le soluzioni che vanno bene dappertutto, sono sempre poco ottimizzate

Spostare i dati dalla memoria consuma sempre, cercare di limitare i grandi spostamenti.

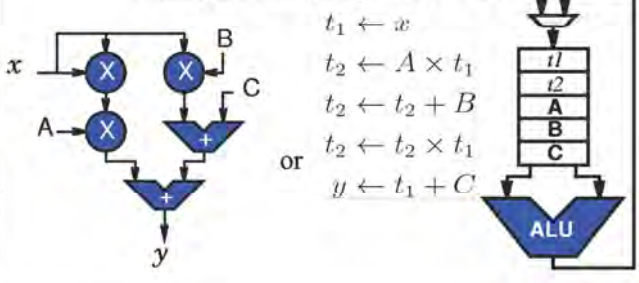
Un buon progetto low power, non può prescindere dalla statistica degli ingressi.

Uso energia solo se serve: clk_gating o sleep mode, se non usato

Matching Computation and Architecture

- Choice of computational architecture can have major impact on energy efficiency (see further)

Example: Compute $y = A \cdot x^2 + B \cdot x + C$



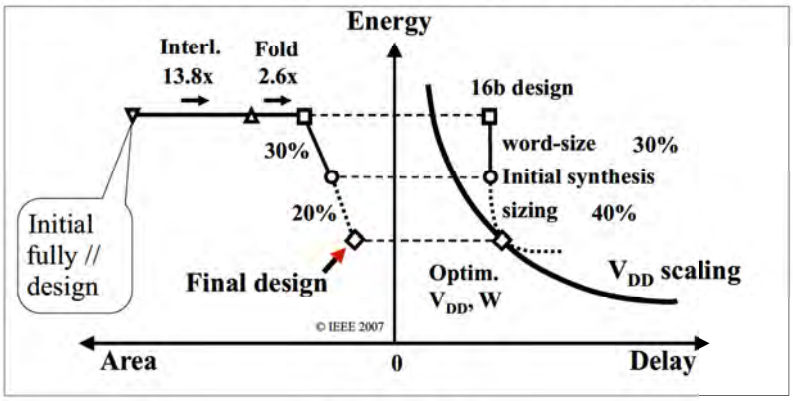
SEBC-L7 MZ 36

Mod by Giorgio Fissore, pag 143

Energy-Area-Delay Tradeoff in SVD

Impact of combined optimizations

- Folding, interleaving, sizing, word length, voltage scaling
- 64x area & 16x energy reduction compared to direct mapping

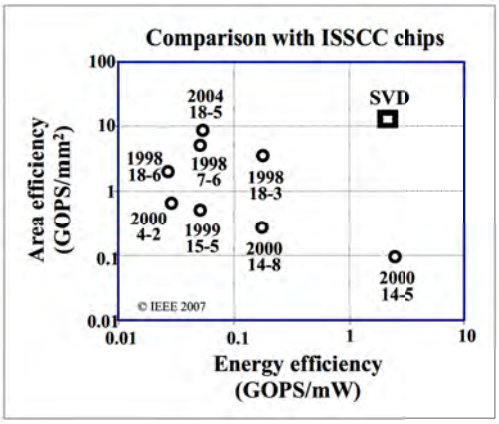


[Ref: D. Markovic, JSSC'07]

SEBC-L7 MZ 40

Power/Area Optimal 4x4 SVD Chip

- 2.1 GOPS/mW
  - ★ 100 MHz clock
  - ★ 70 GOPS
  - ★ Power = 34mW
- 20 GOPS/mm²
  - ★ 3.5mm²
  - ★ 70 GOPS



[Ref: D. Markovic, JSSC'07]

SEBC-L7 MZ 41

Precomputation-Based Optimization

- Basic idea: Precompute (with low-overhead hardware) circuit output logic values 1 cycle before they are needed
- Use precomputed information in next clock cycle to disable unneeded hardware, reduces switching activity
- Must be careful: Precomputation hardware can add to area and lengthen clock period

SEBC-L7 MZ 42

Analogo architetturale del clock gating.
Se io facendo un'operazione intuisco già quale sarà il risultato, è inutile che faccia lavorare le unità successive.

Costi:
-la tecnica di pre-valutazione potrebbe aumentare un po' il ritardo (magari la si evita sui percorsi critici)
-necessità di un HW aggiuntivo per fare queste valutazioni, che consuma a sua volta.

Mod by Giorgio Fissore, pag 145

Can Precompute Outputs Needed 2 or More Clocks Later

- Can reduce switching activity by 12.5% in the adder-comparator circuit of next slide

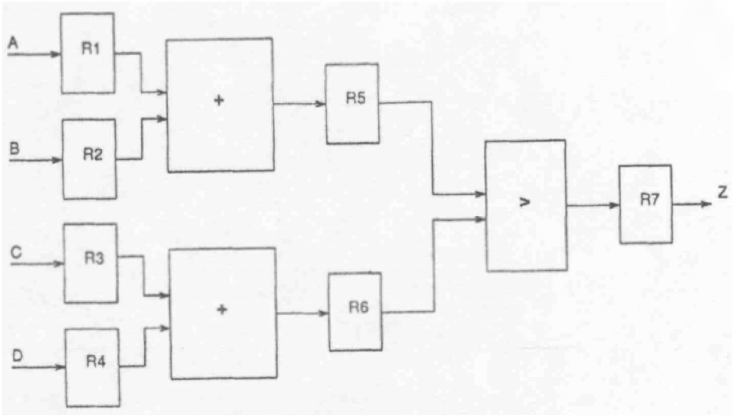
$$f_1 = A(n-1) \cdot B(n-1) \cdot \overline{C(n-1)} \cdot \overline{D(n-1)}$$

$$f_2 = \overline{A(n-1)} \cdot \overline{B(n-1)} \cdot C(n-1) \cdot D(n-1)$$

- Data are used two clock cycles later

SEBC-L7 MZ 46

Example Adder-Comparator

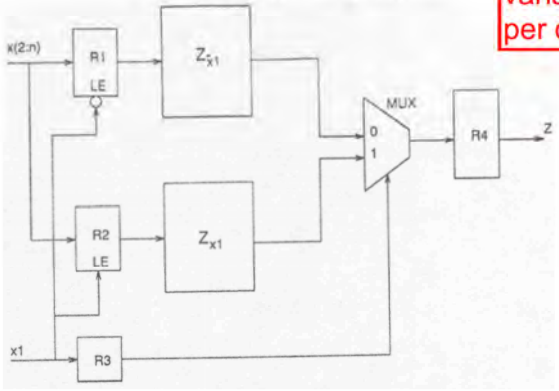


SEBC-L7 MZ 47

Man mano che aumenta la complessità, magari il risparmio peggiora un po' (es con due porte risparmio il 12%), ma su un hw più complicato

Precomputation with Shannon's Expansion Theorem

$$Z = x_j Z_{x_j} + \overline{x}_j Z_{\overline{x}_j}$$



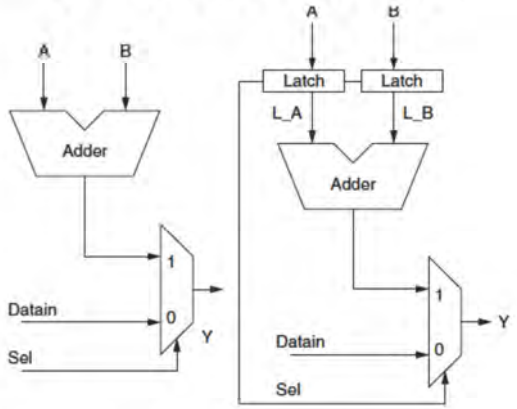
SEBC-L7 MZ 48

"una qualunque funzione di n variabili booleane può essere scomposta in termini moltiplicati per una variabile (normale e complementata), moltiplicata per dei cofattori che dipendono da....."

"prendi l'ingresso che da più problemi come Esw, allora fai l'espansione in due blocchi che non dipendono più da quell'ingresso e che sono un po' più compatti; poi con la prevalutazione ne spengo uno o l'altro" (all'incirca.....)

Mod by Giorgio Pissore, pag 147

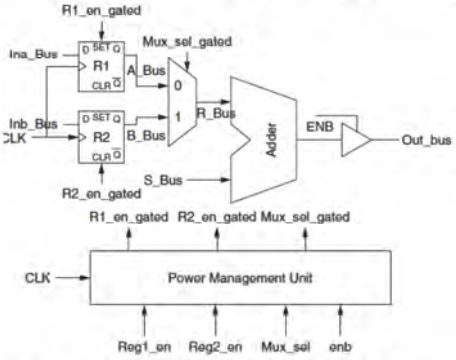
Guarded Evaluation



- ALU updatd only when used in the MPX, otherwise previous value is preserved.

SEBC-L7 MZ 52

Control-signal gating

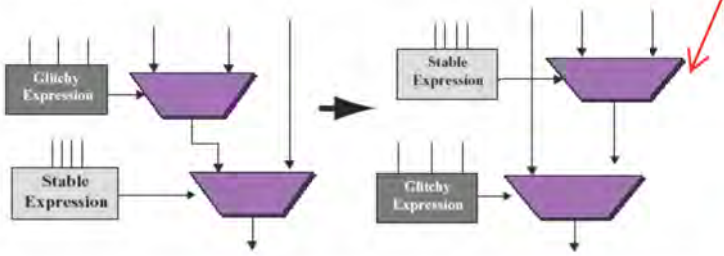


- Observability Don't care Concept (ODC) used to detect when a bus is not used and to stop data propagation through the module driving the bus.
- When ENB inactive, mux and reg ctrls are gated

SEBC-L7 MZ 53

Logic Depth Reduction for Frequently Switching Signals

- By reordering "if - then...else" constructs, the user can move glitchy or fast-changing signals down the logic cone.
- This reduces switching activity propagation and power consumption.



SEBC-L7 MZ 54

Se metto più a valle l'elemento più "ballerino", magari con più glitch, le sue commutazioni non si propagheranno sugli elementi sotto di lui. (anche questo quindi dipende dalle statistiche di Esw)
-Analogo al pin-swapping

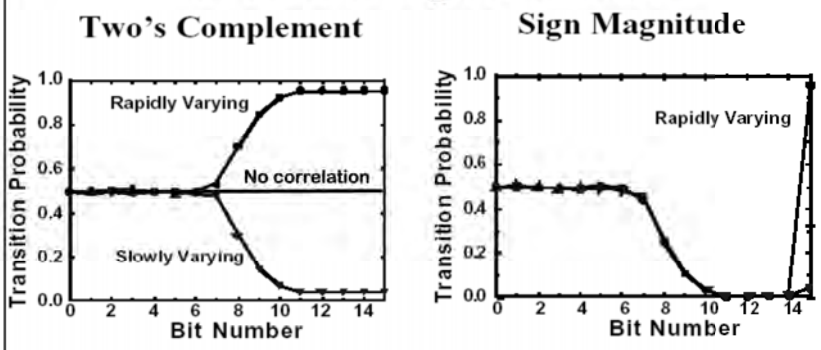
Mod by Giorgio Fissore, pag 149

Number representation

- In 2's complement changing from +1 to -1 (on a 8 bit word) implies 7 switches (00000001 to 11111111)
- In sign and magnitude the same change implies only one switch (00000001 to 10000001)
- If we consider Gaussian data, expressed on a 16 bit data bus with three distributions:
  - No correlation between consecutive words
  - Data rapidly varying (positive correlation)
  - Data slowly varying (negative correlation)

SEBC-L7 MZ 58

Number representation



- **Sign-extension activity significantly reduced using sign-magnitude representation**

SEBC-L7 MZ 59

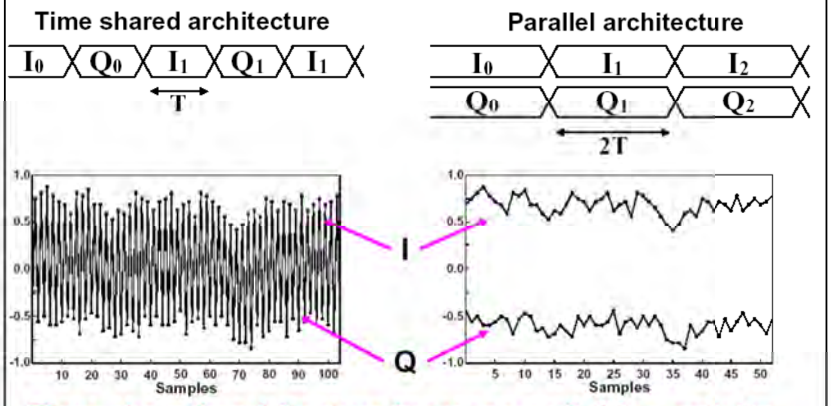
Number representation - Results

- Bit 0 to 6: Same transition probabilities
- Bit 7 to 10: Sign and magnitude is slightly better
- Bit 11 to 14: Sign and magnitude is dramaticaly better
- Bit 15: Same transition probabilities
- Remarks:
  - S & M reduces bus transitions...but
  - 2's Complement has a reduced arithmetic burden (simpler algorithms...less power)
  - Best solution: 2's complement inside arithmetic circuits ... adding encoding-decoding logic at the interface ALUs/Busses

SEBC-L7 MZ 60

Mod by Giorgio Fissore, pag 151

Reduce the Switching activity



Time shared architecture increase switching activity

SEBC-L7 MZ 64

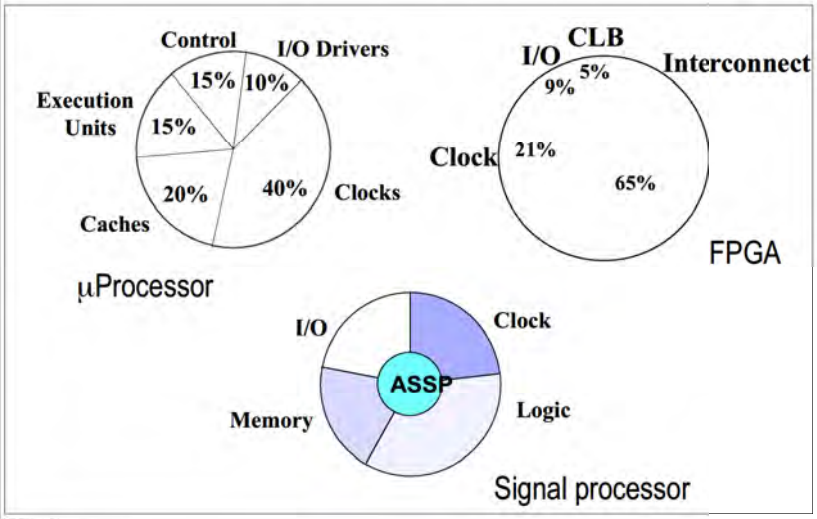
Summary

- Architectural level synthesis
  - Resynthesize state variable equations to save power
  - Scale down supply voltage, and introduce parallelism and pipelining to make up for slow-down of hardware
  - Choose number system in order to reduce switching.
  - Select low - power bus coding

SEBC-L7 MZ 65

Mod by Giorgio Fissore, pag 153

Communication Dominant Part of Power Budget



SEBC-L7a MZ 4

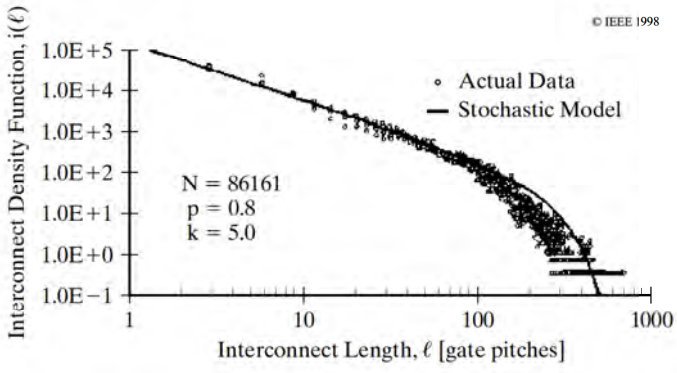
Idealized Wire Scaling Model

Parameter	Relation	Local Wire	Constant Length	Global Wire
W, H, t		$1/S$	$1/S$	$1/S$
L		$1/S$	1	$1/S_C$
C	LW/t	$1/S$	1	$1/S_C$
R	L/WH	S	S^2	S^2/S_C
$t_p \sim CR$	L^2/Ht	1	S^2	S^2/S_C^2
E	CV^2	$1/SU^2$	$1/U^2$	$1/(S_CU^2)$

S=Tech. Scale factor (e.g. 1.44)
Sc= Chip scale factor (e.g. 0.88)
U= Vdd scale factor

SEBC-L7a MZ 5

Distribution of Wire Lengths on Chip



[Ref: J. Davis, C&S'98]

SEBC-L7a MZ 6

Mod by Giorgio Fissore, pag 155

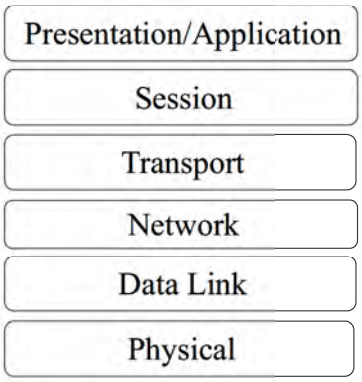
Reducing Interconnect Power/Energy

- Same philosophy as with logic: reduce capacitance, voltage (or voltage swing) and/or activity
- A major difference: sending a bit(s) from one point to another is fundamentally a **communications / networking problem**, and it helps to consider it as such.
- Abstraction layers are different:
  - For computation: device, gate, logic, micro-architecture
  - For communication: wire, link, network, transport
- Helps to organize along abstraction layers, well understood in the networking world: the OSI protocol stack

SEBC-L7a MZ 10

OSI Protocol Stack

- Reference model for wired and wireless protocol design — Also useful guide for conception and optimization of on-chip communication
- Layered approach allows for orthogonalization of concerns and decomposition of constraints

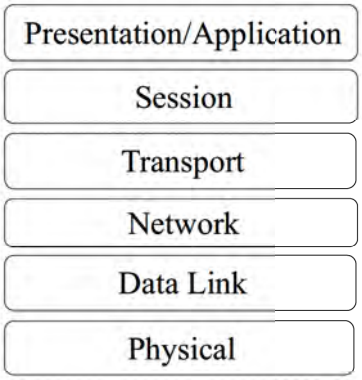


SEBC-L7a [Ref: M. Sgroi, DAC'01] MZ 11

The Physical Layer

Transmit bits over physical interconnect medium (wire)

- Physical medium
  - Material choice, repeater insertion
- Signal waveform
  - Discrete levels, pulses, modulated sinusoids
- Voltages
  - Reduced swing
- Timing, synchronization



So far, on-chip communication almost uniquely "level-based"

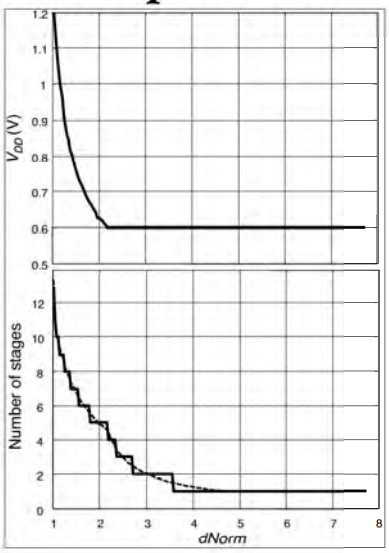
SEBC-L7a MZ 12

Mod by Giorgio Fissore, pag 157

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 159 di 402

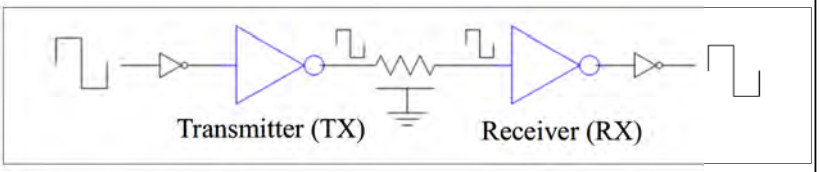
Multi-dimensional Optimization

- Design parameters: Voltage, number of stages, buffer sizes
- Voltage scaling has largest impact, followed by selection of number of repeaters
- Transistor sizing secondary.



SEBC-L7a MZ 16

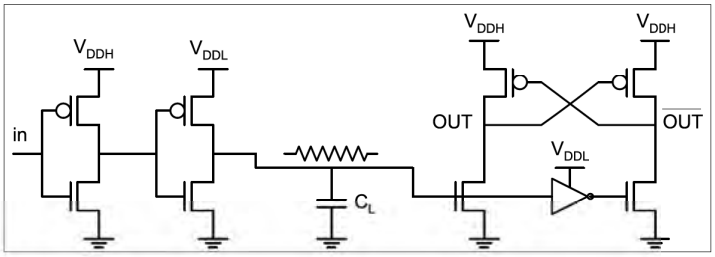
Reduced Swing



- $E_{bit} = CV_{DD}V_{swing}$
- Concerns:
  - Overhead (area, delay)
  - Robustness (supply noise, crosstalk, process variations)
  - Repeaters?

SEBC-L7a MZ 17

Traditional Level Converter



- Requires two discrete voltage levels
- Asynchronous level conversion adds extra delay

SEBC-L7a [Ref: H. Zhang, TVLSI'00] MZ 18

Mod by Giorgio Fissore, pag 159

Signaling Protocols

Network

d_{in} req_{in} ack_{in} d_{out} req_{out} ack_{out}

Processor
Module
(μProc, ALU, MPY, SRAM…)

Globally Asynchronous

self-timed
handshaking protocol

Allows individual modules to dynamically trade-off performance for energy-efficiency

SEBC-L7a MZ 22

Signaling Protocols

Network

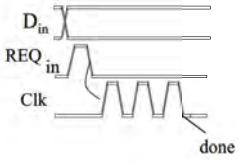
d_{in} req_{in} ack_{in} d_{out} req_{out} ack_{out}

Physical Layer
Interface Module

d_{in} d_{out} clk done

Processor
Module
(mProc, ALU, MPY, SRAM…)

Globally Asynchronous



Locally Synchronous

SEBC-L7a MZ 23

The Data Link /Media Access Layer

Reliable transmission over physical link and sharing interconnect medium between multiple sources and destinations (MAC)

- Bundling, serialization, packetizing
- Error detection and correction
- Coding
- Multiple-access schemes

Presentation/Application
Session
Transport
Network
Data Link
Physical

SEBC-L7a MZ 24

Mod by Giorgio Fissore, pag 161

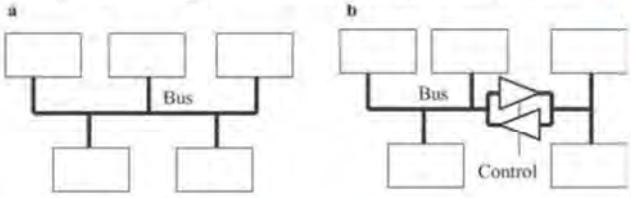
Low Power Bus Design

- Low power bus design techniques
  - lower switching activity, reduce capacitance to be switched
  - choice of drivers (3-state vs. open-collector; current-mode low swing drivers)
  - minimize bus length by module placement and bus routing
  - build hierarchical bus
  - Optical interconnect?

SEBC-L7a MZ 28

Low Power Bus Design

- Bus segmentation –
  - transform a long heavily loaded global bus into a partitioned multistage network by inserting pass transistors on the bus lines to separate various local buses (segments)
  - partitioning can reduce bus power by 60%



Going from a common bus (a) to a split bus (b).

SEBC-L7a MZ 29

Tenendo conto del fatto che le interconnessioni sui bus sono le più pesanti, è un miglioramento significativo

il fatto che si posizionino bus breaker (e dove metterli) porta a fare un'ottimizzazione del placement >> le parti che si parlano di più vanno messe vicine >> richiede simulazioni

Maggiore è la località delle informazioni, minore è il consumo

Il posizionamento di bus breaker riduce le capacità da caricare, e al contempo aumenta la concorrenza!! (un bus spezzato in due si comporta come due bus)
>> questo aumento di concorrenza, può permettere di abbassare la frequenza di clk (e quindi la Vdd), laddove l'utilizzo continuo del bus costituisca il limite della frequenza.

Low Power Bus Design

- Power consumption can be simply reduced by rescheduling data transfer on existing busses

SEBC-L7a MZ 30

Mod by Giorgio Fissore, pag 163

Example : Total Switching

- After scheduling and bus binding

Total switching

$$SW = SW(a,e) + SW(b,f) + SW(c,x) + SW(d,y) + SW(e,d) + SW(f,z) + SW(d,a) + SW(z,b) + SW(x,c) + SW(y,d)$$

Inter-loop data transitions

Clock Step	Scheduled Operations	bus 1	bus 2	bus 3	bus 4
1	+1 ,+2	a	b	c	d
2	+3 ,+4	e	f	x	y
3	+5	d	z		

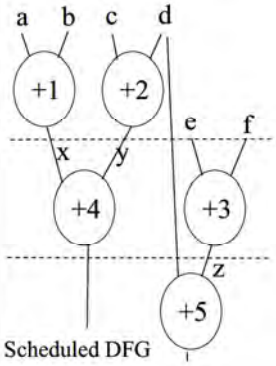
SEBC-L7a MZ 34

Bus binding: decido quale dato deve viaggiare su quale bus.
(si suppone che queste addizioni vengano effettuate in maniera continua)
in prima approx lo posso decidere a caso

La Esw è data da tutti i passaggi da un dato all'altro sui vari bus; nella prox slide vediamo che questa soluzione comporta una grande Esw

Example : Total Switching (2)

- Result by brute-force scheduling and binding



Scheduled DFG

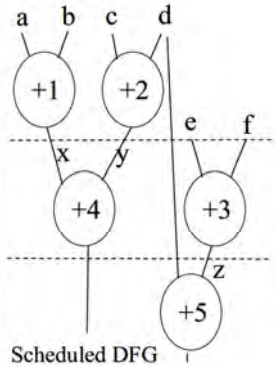
Clock Step	Scheduled Operations	bus 1	bus 2	bus 3	bus 4
1	+1 ,+2	a	b	c	d
2	+3 ,+4	e	f	x	y
3	+5	d	z		

Values with high average switching

SEBC-L7a MZ 35

Example : Bus Rebinding

- Result by bus rebinding



Scheduled DFG

Clock Step	Scheduled Operations	bus 1	bus 2	bus 3	bus 4
1	+1 ,+2	a	b	c	d
2	+3 ,+4	e	y	f	x
3	+5		d		z

Values with high average switching

Values with low average switching

SEBC-L7a MZ 36

Devo decidere dove deve viaggiare ciascun dato, cosa passa prima e cosa passa dopo. Questo metodo è efficace solo se abbiamo la possibilità di fare simulazioni e siamo in presenza di loop ripetuti molte volte.

Mod by Giorgio Fissore, pag 165

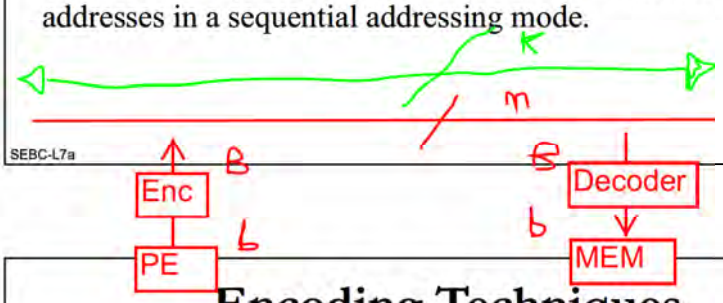
Low Power Bus Design

• We first introduce the terminology and notation that will be used throughout this lecture:

b(t): Address value to be sent on the bus at time *t* (source word at time *t*).

B(t): Encoded value on the bus lines at time *t* (code word at time *t*).

S: Stride value, which is the difference between consecutive addresses in a sequential addressing mode.



SEBC-L7a MZ 40

Prima diamo un po' di termini da usare qui di seguito:
-b(t): il valore non codificato che va mandato sul bus al tempo t
-B(t): il valore codificato che viaggia sul bus nell'istante t
[b e B possono essere sia dati che indirizzi]
-S: stride, quando mando solo indirizzi, differenza tra un indirizzo inviato e l'indirizzo successivo mentre vado in sequenza (ad esempio, se mando 4 dati alla volta lo stride sarà di 4).
>>incremento minimo dell'indirizzo quando vado in sequenza

Encoding Techniques

• A number of encoding techniques rely on introducing redundancy to save power.
• These techniques add one or more extra bits to the original bus.
• However, the extra bus lines cannot be tolerated in many systems because the extra bits require hardware changes and often cause incompatibility with standard bus interfaces.
• Consequently, a great deal of effort has been spent in finding irredundant encoding techniques that reduce the switched capacitance on the bus while preserving compatibility with existing bus interfaces and the rest of the system.

SEBC-L7a

Queste tecniche di codificazione vanno subito distinte in due macro categorie: rindondanti e non rindondanti.
-trasmetto su un numero di linee pari al numero di bit di informazione che voglio inviare
-trasmetto su un numero di linee maggiore rispetto al numero di bit da inviare (k bit rindondanti nel mio disegno di sopra).
Chiaramente una trasmissione che usi rindondanza mi può permettere di ridurre meglio la Esw, lasciandomi più libertà.

Meglio usare rindondanza o meno quindi?
dipende >> se mi faccio il mio progetto tutto per i fatti miei posso mettere tutta la rindondanza che voglio, ma se invece devo magari usare un protocollo di trasmissione standard,.. allora ciò non è vero.

Bus-Invert method

- Consider an *N*-bit (non-multiplexed) bus.
- The idea is that if the Hamming distance between two consecutive patterns is larger than $N/2$, the second pattern can be inverted so as to reduce the inter-pattern Hamming distance to below $N/2$.
- One redundant bit is needed to distinguish between the original and inverted patterns on the bus.
- The Bus-Invert method tends to perform well when sending random patterns, which is often the case on data busses.
- However, this method is largely ineffective on address buses, which tend to exhibit a high degree of sequentiality.

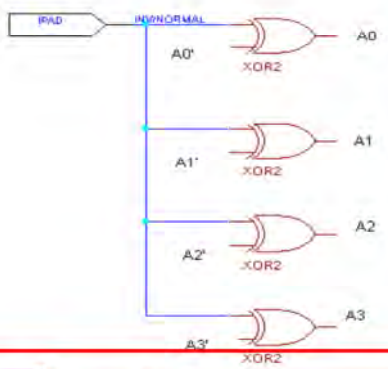
SEBC-L7a MZ

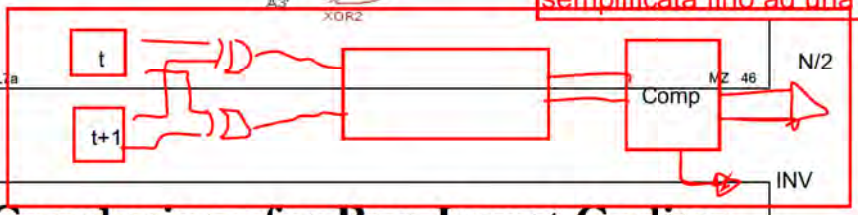
Con un bit di rindondanza:
Il caso peggiore per le commutazioni sarebbe di dover passare da tutti zeri a tutti uni; mando quindi un bit in più (chiamato bus invert) che ci dice se viaggia il dato vero, o il complemento ad uno del dato
>>se b(t) e b(t+1) differiscono di più di N/2 bit, allora con questo bit di inversione abbatto il numero di commutazioni. (le riduco tutte le volte che ci sono più di N/2 comm).
-Pago che ho bisogno di un bit rindondante, e che non sono più in uno std.
-Cosa/quando guadagno?
Se sto inviando dati casuali, il guadagno è elevato dato che c'è una buona probabilità di avere molti bit che commutano.
Se invece sto inviando dati con forte correlazione(es indirizzi eccetto nei salti), allora il metodo non è efficiente >> dipende tutto dalla conoscenza della statistica degli ingressi!

Decoding circuit at memory end

In questo caso la decodifica è molto semplice: basta far passare i dati dentro ad un EXOR. Più complicata è la codifica.

Ciò che capita spesso è che ci sia un solo oggetto che parla (e quindi codifica) e tanti oggetti che ascoltano (e decodificano).
Quando codifico, non è detto che enc e dec abbiano la stessa complessità; quando si sceglie un codice piuttosto che un altro, va considerato il fatto che spesso è importante più abbassare il consumo/costo del decoder piuttosto che quello dell'encoder.
Con l'inv, vediamo che ciò è rispettato, con una dec semplificata fino ad una sola porta.





Codificatore

SEBC-L7a MZ 46

Conclusions for Bus-Invert Coding

- Maximum number of transitions is reduced from N to N/2
- Thus range for number of transitions on bus is 0-N/2
- Power Saving can range from about

(N-1)/Nx100% to 1/(N+1) x 100%

- Device at memory end would be needed to invert the bits according to INV/NORMAL'
- The average number of transitions is also lowered by less than 25% (because of the binomial distribution of the distance between consecutive patterns)

SEBC-L7a MZ 47

Bus invert coding:
-va bene per dati casuali
-no per indirizzi
-decodifica semplicissima

Extensions of Bus-Coding

- Partial Bus Coding- In this format only the most active lines of a given bus are coded instead of the whole bus
- Gray Code- Schemes other than plain inversion where some code such as gray code or some other code can be used to reduce the number of transitions.

SEBC-L7a MZ 48

Vediamo ora cosa si può fare per gli indirizzi

La codifica Gray andrebbe bene, e non richiede altri bus di rindondanza (ok per comunicazione std), ma, anche se abbassa i consumi (Esw), non li abbatte mai del tutto; li riduce solo ad 1 quando vado in sequenza)
>>con bit di rind posso fare di meglio.

Transition-based code

- Encode the input pattern using limited-weigth codes, which guarantee a good control of the values of *p(0)* and *p(1)* of each bit line. Then apply bit encoding using the transition-based scheme
- The encoded schemes are ok for data bus transmission
- Different techniques can be used to reduce activity on the address bus lines
- Among them:
  - gray code (irredundant)
  - T0 code (redundant)
  - Working Zone code (redundant)
  - Beach code (irredundant)

SEBC-L7a MZ 52

Se gli zeri e gli uni non sono equiprobabili, ma l'info che viaggia è sbilanciata verso gli zeri o gli uni che invio, una codifica transition based può essere anche molto vantaggiosa.
Se ad esempio ho un p(1) molto bassa, allora Esw (trans) = p(1)
Esw (level) ~ 2*p(1)

.

>>> devo allora cercare di muovermi verso una codifica molto sbilanciata tra zeri ed uni

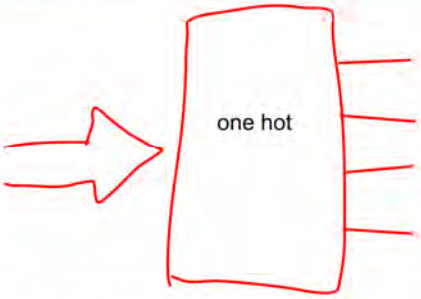
.

>>> aggiungo quindi dei blocchetti di enc e dec che trasformino ancora il dato in maniera sbilanciata. (es codifica one-hot, possibile solo però con pochissimi bit in questo caso)

Grey code

- Two consecutive words have unit Hamming distance
- Due to the instruction locality, most of the accesses are sequential
- A significant number of switches can be eliminated
- The main problem is that address generation circuits are more complex than binary ones.
- Architectural solutions:
  - Gray address generators using conversion circuitry
  - Binary address generators and external conversion circuitry
- Grey code is almost optimal for irredundant codes
- To increase power saving.... redundant codes are needed

SEBC-L7a MZ 53



Con la one-hot e un codice transition-based, comunque cambi il dato in ingresso, ci sarà comunque una sola transizione, dovuta all'unico 1 inviato

Il problema è che enc e dec sono molto più costosi; generare indirizzi in modo gray al posto che nel modo binario è costoso.
"lavoro internamente in modo binario, e converto in gray prima di andare sul canale, poi dalla memoria devo di nuovo decodificare e andare a prendere l'indirizzo"
>>è possibile semplificare se tutto lo spazio di indirizzamento del PE è occupato dalla memoria (accade spesso nei uC) >> allora, dato che la conversione è 1 ad 1, il decoder è inutile, e si salvano le cose nell'indirizzo codificato gray (il PE pensa di chiedere l'indirizzo 5, poi in codifica gray è 47, salvo in 47 >> non ce ne importa poichè tanto il processore codificherà sempre gli indirizzi e non si accorgerà di dove realmente sono salvati).

T0 Code

• The *T0* code exploits data sequentiality to reduce the switching activity on the address bus (asymptotically tends to zero.. in case of complete sequentiality).
• The observation is that addresses are sequential except when control flow instructions are encountered or exceptions occur.
• T0 adds a redundant bus line, called *INC*. If the addresses are sequential, the sender freezes the value on the bus and sets the *INC* line. Otherwise, *INC* is deasserted and the original address is sent.
• On average 60% reduction in address bus switching activity is achieved by T0 coding.

SEBC-L7a MZ 54

Mod by Giorgio Fissore, pag 171

T0-C code

- It improves T0 code in a number of important ways:
- First of all, it eliminates the redundant bit.
- Second, it results in higher power saving on the bus.
- Similar to T0 code, the basic saving happens as a result of freezing the bus when addresses are sequential.
- Suppose that we suppress the redundant bit in T0 code; when b(t) and b(t-1) are sequential addresses, we simply freeze the bus, and in all other cases, we send the original source word on the bus.
- This simple scheme would fail, for example, when we encounter backward branches where the branch target address is the same as the current (frozen) bus value.

SEBC-L7a MZ 58

Questo codice richiede un po' più di intelligenza dalla parte del codificatore e del decodificatore: "quando incrementi in realtà non mandarmi niente" >> no HW aggiuntivo, va bene per trasm std vediamo un esempio:

b(t)	T0-C	"cosa capisco"
39	39	39
40	39	40
41	39	41
42	39	42
57	57	57
58	57	58
59	57	59
60	57	60
61	61	57

T0-C code

Consider the following simple example:

$b^{(t)}$	$B^{(t)}$
39	39
40	39
41	39
39	39 ?

- As it can be seen, when we reach the last row of the table, no valid code word can be generated for source word 39.
- If we use 39 as the code word, the receiver (decoder) cannot determine whether the source word was 39 (backward jump) or 42 (next sequential address).
- So the problem occurs when the data on the bus is equal to the branch target itself: that is why spatial redundancy was originally introduced into the T0 code.
- However, there is a better way to resolve this problem.

SEBC-L7a MZ 59

Questa tecnica entra in crisi durante i loop; infatti se incremento sempre di uno e poi devo saltare al PRIMO indirizzo del loop, lui non riconoscerà l'indirizzo di salto, ma penserà che vuoi incrementare (in questo caso non mi riconosce 57).
Qual'è l'unico valore che non mi sarei mai aspettato di ricevere allora? l'indirizzo che raggiungerei incrementando di uno!
Allora se da 60 voglio tornare a 57, mando 61, che è l'unico indirizzo che non potrei voler raggiungere con un salto, e lui lo interpreterà allora con "torna all'inizio del loop".

Questa tecnica è usatissima perciò in moltissime schede!!
l'unico caso in cui non funziona è un'istruzione singola di assembler: test and set >> rimane in un loop su una sola istruzione fino a che non viene settato un bit. (se ho capito bene)
Serve ad esempio con prenotazioni da più terminali: quando inizio la prenotazione, setto un semaforo rosso che impedisce agli altri di entrare....poi fin che non decido se prenotare o no, quel punto non è accessibile ad altri.
>>> In caso di stride diverso da 1, allora il numero da mandare per tornare all'inizio del loop sarà (b(t-1) + S)

T0-C code

- To correctly handle backward branches with target addresses equal to the current bus value, a special pattern has to be sent to the rx. This cannot be a fixed pattern because we assume that jumps to any and all addresses are allowed (picking any fixed pattern to designate this case may create a potentially large activity on the bus, and at the same time, requires that the fixed pattern not be used as a regular jump address).
- In T0-C when such a case occurs, we set the code word to b(t-1) + S.
- This is the only pattern that the rx should not expect from the sender.
- When the rx sees a value of b(t-1) + S, it knows that the sequential addressing has been stopped because the bus value has changed.
- It computes the new jump address, it recognizes that this jump address is the same as the next sequential address.
- Therefore, if a special case were not encountered, there would be no need for the sender to unfreeze the bus value.

SEBC-L7a MZ 60

Mod by Giorgio Fissore, pag 173

Working Zones coding

- Offset definition:
  - with respect to the base address of the zone
  - with respect to the previous reference to that zone
- Hardware requirement
  - One reg for each working zone at Tx end
  - One reg for each working zone at Rx end
  - Additional bus lines to signal to Rx what zone is active
- Implementation
  - Consider a limited number of working zones
    - additional reg not needed
    - reduced number of additional lines
  - Use existing bus lines to transmit offset encoded (one-hot scheme)

SEBC-L7a MZ 64

Naturalmente più il mio sw è concentrato in un numero limitato di zone, tanto più questa tecnica è efficace.
>> lavorare in sistemi embedded, in cui spesso si sta in un loop dentro alla stessa zona di lavoro, porta a risparmi anche molto alti! (magari 40%)

Working Zones coding

- Further reduce the activity during offset tx by adopting a transition signaling code (XOR based)
- When a reference does not correspond to any of the chosen working zones, the entire reference is transmitted (and signaled to rx)
- Limitations:
  - Applications may have a large number of working zones. To keep the number of reg under control use a caching mechanism for active zones .... Smaller power savings
  - Adding bus lines may be unacceptable (standards,etc...). Use fewer existing lines for offset tx and the remaining for zone identifier .... Smaller power savings...

SEBC-L7a MZ 65

Working Zones coding - Results

- Motion estimation algorithm:

  47% savings in # of transitions with respect to binary
- Quicksort algorithm:

  67% savings in # of transitions with respect to binary

SEBC-L7a MZ 66

Mod by Giorgio Fissore, pag 175

Beach solution coding - Results

- Experiments on sw functions typical of embedded systems
  - Image processing
  - Automotive control
  - DSP applications
  - Robotics
- Average savings over binary address encoding: 41,9 %

SEBC-L7a MZ 70

Information-Theoretic Code

- The encoding algorithm exploits the *correlator* concept
- The target is to minimize word transition probabilities (minimize the number of one transmitted)
- The correlator maps ***ones*** to **transitions** with the algorithm $B(t) = B(t\text{-}1)\ xor\ b(t)$
- In general the encoder must minimize the average number of 1's while guaranteeing unique decodability of $B(t)$
- Symmetric operations occur in the decoding phase

SEBC-L7a MZ 71

Codificatore transition based

Operazione per convertire la codifica da level based a transition based

Attenzione! non tutte le codifiche vanno bene, ma devo capire se per ogni cod esiste una correlazione 1 ad 1 per tornare indietro

Information-Theoretic Code

- The encoding algorithm:
  - Sorts the pairs of input data word according to their probabilities
  - Starting from the most probable pair:
    - Assign minimum-one codes
    - Update decodability constraints
  - Define the encoder and decoder function
- Input data probability is required; as a consequence this method can be used only in embedded systems.
- Optimal solution is impractical... approximate solutions used instead (clustering encoding, discretized encoding)

SEBC-L7a MZ 72

Servono dati molto precisi sulla statistica dei dati in ingresso, poi si può fare:
-andiamo a vedere i dati in ingresso ed ordiniamoli in base alla loro probabilità (dai dati più prob a quelli meno).
-assegniamo ai dati più probabili quelli con il numero minore di uni, e andiamo a scendere.
>>sarebbe la soluzione ottimale, tuttavia risulta spesso impraticabile (es se dobbiamo trasmettere dati su 64 bit); si cercano quindi delle configurazioni semplificate per fare processi di questo tipo.
-cmq non vedremo troppo nello specifico questi codici a seguire

Offset-Xor-S Code

• This encoding will become much more effective if the coding algorithm is modified as follows (resulting in a code that we will call *Offset-Xor with Stride* or *Offset-Xor-S* for short):

B(t) = (b(t) - b(t-1) - S) xor B(t-1)

• The reason for Offset-Xor-S improvement over Offset-Xor is that it avoids switching activity when sequential addresses are encoded.

(S = Stride value, is the difference between consecutive addresses in a sequential addressing mode)

SEBC-L7a MZ 76

Consecutive source word Xor problem

• Sometimes, even if the difference between b(t) and b(t-1) + S is small, their Hamming distance may be quite large.

• This usually occurs for source words b(t) and b(t-1) that are located at opposite sides of 2^N, e.g., 61 and 69 are located at the two sides of 64.

• In these cases, although the offset is small, b(t) xor (b(t-1) + S) contains many ones and thus causes many transitions on the bus when it is Exclusive-Or'ed with the value on the bus.

• This is the "consecutive source word Xor problem".

SEBC-L7a MZ 77

Table 2- Encoder hardware synthesis and power estimation

	T0-Xor	T0-C	Offset-Xor-SM	Offset-Xor-SMC
Number of literals	440	767	661	2693
Area of Encoder (in thousands)	334	410	399	1043
Number of gates	306	386	379	1136
Power dissipated by encoder & decoder (uW)	266	642	740	1822

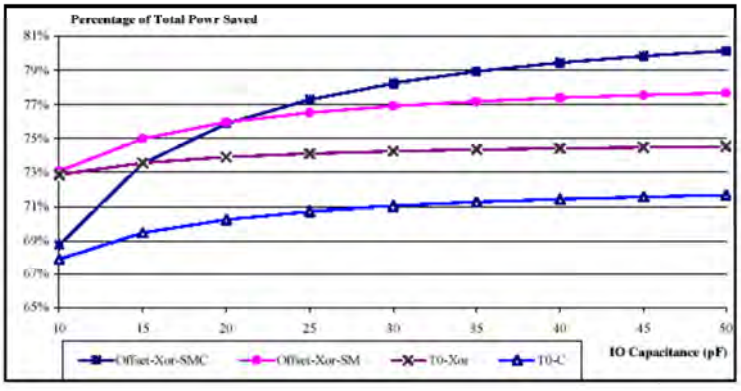


Figure 1- Comparison of total power savings of different encoding techniques

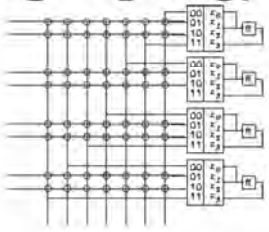
SE

Come si vede qui, non esistono codici ottimali indipendenti dalla capacità della linea da caricare.

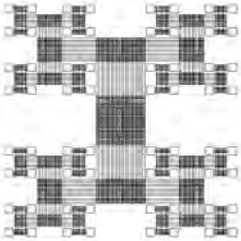
Mod by Giorgio Fissore, pag 179

Networking Topology

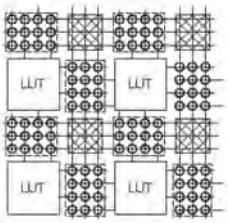
- Homogeneous
  - Crossbar, Butterfly, Torus, Mesh,Tree, …
- Heterogeneous
  - Hierarchy



Crossbar



Tree



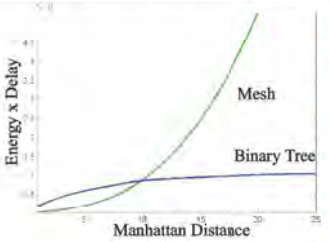
Mesh (FPGA)

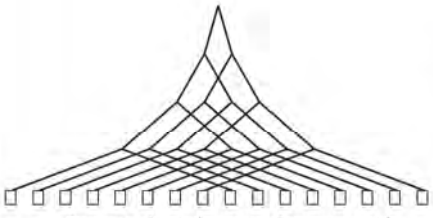
SEBC-L7a MZ 82

Network Topology Exploration

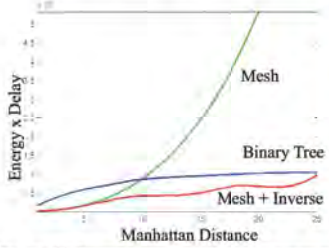
Soluzione ad albero binario (come si vede, molto migliore rispetto a soluzione mesh)

Short connections in tree are redundant





Inverse clustering complements mesh



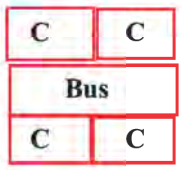
SEBC-L7a [Ref: V. George, Springer'01] MZ 83

La soluzione ad albero binario ha però un inconveniente, poichè consuma troppo sulle corte distanze, per cui la mesh (passare l'info da un PE a quello a fianco). Questa soluzione infatti privilegia le trasmissioni tra elementi lontani rispetto a vicini

Allora la seconda soluzione è una sorta di mix tra le due soluzioni: "l'albero inverso". Collego tra loro ai due rami dell'albero, due oggetti distanti tra loro (es metà della distanza); per le corte distanze si passerà invece da un oggetto a quello a fianco con la mesh. (credo per tutte le distanze più corte di quella tra due estremi di un ramo) E' la tecnologia più usata all'interno dell'FPGA.

Circuit-Switched versus Packet Based

- On-Chip Reality: Wires (bandwidth) are relatively cheap, buffering and routing expensive
- Packet-switched approach versatile
  - Preferred approach in large networks
  - But … routers come with large overhead
  - Case study Intel: 18% of power in link, 82% in router
- Circuit-switched approach attractive for high-data rate quasi-static links
- Hierarchical combination often preferred choice



Bus to connect over short distances

La gran parte della potenza viene spesa nel routing!! nel caricarsi da un punto all'altro,..

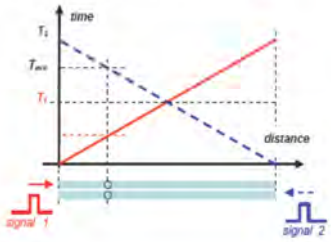
Mod by Giorgio Fissore, pag 181

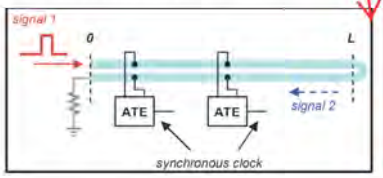
SEBC-L7a MZ 84

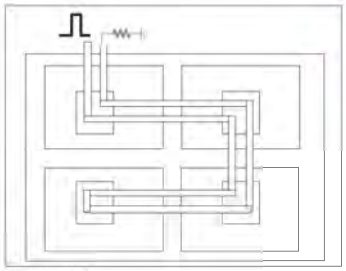
Alternative Clock Distribution Schemes

Example: Transmission-Line Based Clock Distribution

Canceling skew in perfect transmission line scenario







ATEs extract the average between the early and late arrivals (red and blu signals) without any skew

SEBC-L7a [Ref: V. Prodanov, CICC'06] MZ 88

Approssimativamente:
"Far andare il clk da 0 a L e poi farlo tornare indietro.
Se vado a vedere il segnale che viaggia, esso raggiungerà prima il primo PE, poi il secondo PE,.. ma questa cosa accadrà al contrario al ritorno
Se io faccio una media tra i due tempi di arrivo, sarei in grado di ricostruire il clk in modo preciso"

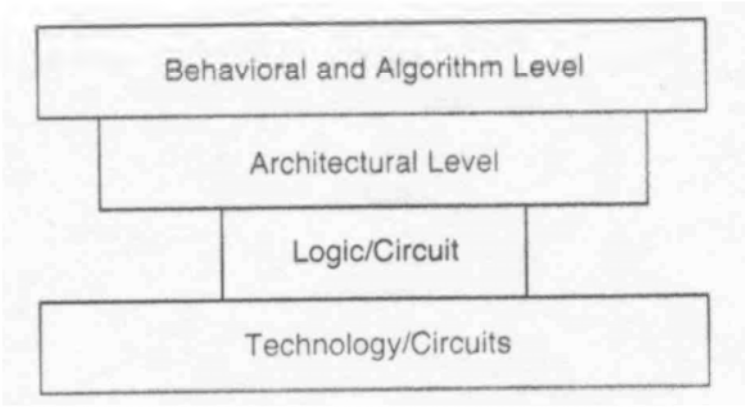
Summary

- Interconnect important component of overall power dissipation
- Structured approach with exploration at different abstraction layers most effective
- Lot to be learned from communications and networking community – yet, techniques must be applied judiciously
  - Cost relationship between active and passive components different
- Some exciting possibilities for the future: 3D-integration, novel interconnect materials, optical or wireless I/O

SEBC-L7a MZ 89

Mod by Giorgio Fissore, pag 183

Algorithm-Level Power Reductions vs. Other Levels



SEBC-L8 MZ 4

Behavioral power optimization

- At the behavioral–level, large power savings can be obtained through the application of transformations.
  - Strategy: Modify the computational structure of the algorithm while preserving its I/O behaviour.
  - Objective: Optimize the power dissipation of the final circuit while meeting the functional throughput of the system.
- After algorithm optimization, an RT-level architecture is created through behavioural synthesis that includes power in its cost function.

SEBC-L8 MZ 5

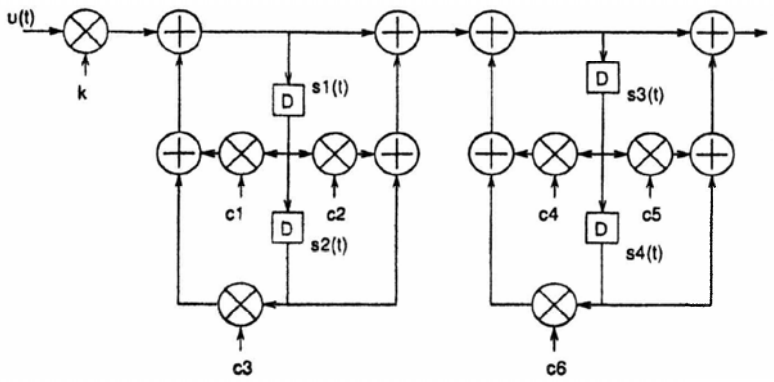
Behavioral power optimization

- Two key approaches to behavioral-level power optimization that involve algorithm transformation:
  - Enabling of supply voltage reduction through application of speed-up transformations (i.e. transformations commonly used for performance optimization).
  - Minimization of the effective capacitance through application of more generic transformations.

SEBC-L8 MZ 6

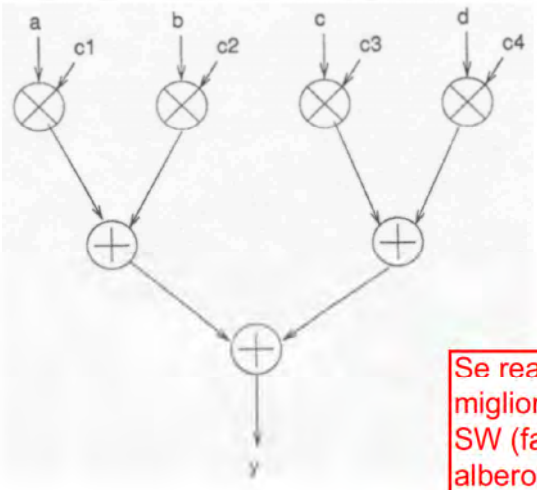
Possiamo ottimizzare principalmente su due piani:
1) il mio algoritmo funziona su N step che richiedono un certo tempo T (e quindi una certa Vdd,..): modificare il data flow diagram in maniera da diminuire il numero di step (magari metto un moltiplicatore in più >> meno clk >> più tempo a disposizione >> posso ridurre la frequenza e quindi la Vdd)
>> Uso le tecniche di speed up, ma questa volta non per aumentare le prestazioni (come già visto, ma ora agisco a livello comportamentale)
2) cerco di modificare il grafo, per diminuire la capacità complessiva che si fa commutare (es mi accorgo che il moltiplicatore ha una capacità equivalente più alta del sommatore >> cerco di implementare lo stesso alg con somme al posto che moltiplicazioni)

Data Flow Graph of IIR Filter



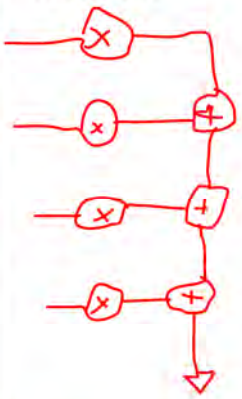
SEBC-L8 MZ 10

Perfectly Balanced Addition Tree



SEBC-L8

Array lineare



Se realizzo questo filtro in HW, la soluzione ad albero è la migliore. Tuttavia il filtro potrebbe essere realizzato solo via SW (faccio un'operazione per volta)>> la soluzione ad albero ottimizzata è quella in cui si fanno passare prima, in ordine, i coefficienti più piccoli e poi man mano quelli più grandi. (se ho capito bene)
Nell'array lineare invece, più sono lontano dall'uscita, più devo avere coefficienti piccoli per ottimizzare la Esw

Filter Implementations

- Can be bit-serial or word-parallel arithmetic
- W bits fed in parallel to adders and multipliers
- At time t + 1, z of W bits change from time t values
- Activity β (t) = z / W
  - β (t) is a random variable stochastic process – strict sense stationary
- Average power dissipation proportional to:

$$\text{Cost} = \sum_{i=1}^{i=\gamma} \theta_i(0, N) \cdot C_i \qquad (4.18)$$

- θ_i =average activity on node i
- In bit-serial implementations, intra-word bit differences, and not inter-word bit differences, cause node activity

Mod by Giorgio Fissore, pag 187

SEBC-L8 MZ 12

Power Optimization Algorithm

1. Simulate circuit at functional level
   - Using random, mutually-independent input values
2. Note signal activities at all adder inputs
3. Restructure adder trees using above 2 hypotheses
   - Move additions with high activity closer to root of computation tree
4. Recompute average activities
5. Iterate until no additional power is saved

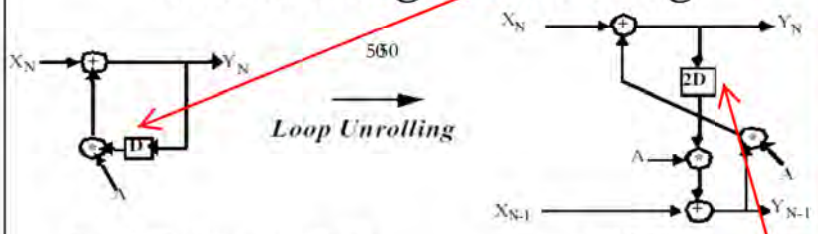
- Method shown to save up to 23% of power

SEBC-L8 MZ 16

Senza lavorare a livello transistor,.., ma solo modificando l'ordine delle operazioni posso salvare 1/4 della potenza!!

Posso modellare un ritardo con un registro chiamato D

Filter retiming and unrolling



C_{eff} = Effective normalized mult-add capacitance = 1
Voltage = 5 c.p.=2
Throughput = 1
Power = 25

Half Speed for same Throughput!

crit.path=4
C_{eff} = 2 /2
Voltage = 5
Throughput = 2 /2
Power = 50 = 50 /2

$Yn = X(n) + AY(n-1) = X(n) + AX(n-1) + A^2Y(n-2)$

Loop-unrolling does not reduce power consumption but allows distributivity and constant propagation

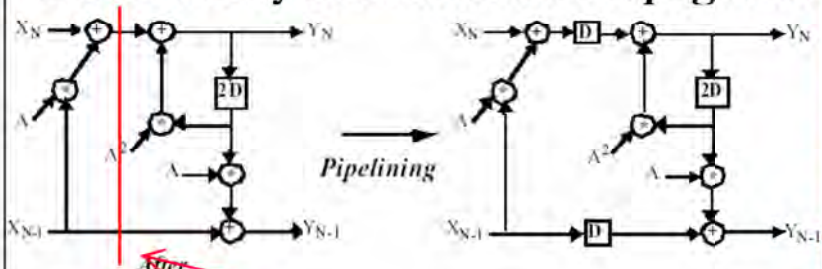
SEBC-L8 MZ 17

Loop unrolling: al posto che fare 1000 cicli (ad esempio in cui si passa ricorsivamente in un sommatore e un moltiplicatore), raddoppio i componenti che realizzano il loop e faccio la metà dei giri (ora ad ogni ciclo si fanno due somme+moltiplicazioni)

Qui ho raddoppiato la lunghezza del critical path.
Se lavoro alla stessa frequenza, con stessa Vdd, raddoppio il througput, e raddoppia anche la capacità_efficace, e quindi pure la potenza consumata.
>> raddoppiando i componenti, vado al doppio della velocità, ma ciò non ci fa guadagnare in potenza (non posso infatti abbassare Val, perchè anche se siamo a f_dimezzata, il carico di lavoro da fare in un clk è doppio).

Cosa guadagno dal pt di vista della potenza? guadagno nel fatto che viene cambiato l'algoritmo >> ho così delle catene più lunghe e su di esse posso fare più ottimizzazioni (vedi slide sotto)

Distributivity and Constant Propagation



After Algebraic Transformations & Constant Propagation
C_{eff} = 3 /2 c.p.=3
Voltage = 3.7
Throughput = 2 /2
Power =40 (20% reduction)

Critical path reduced!!!!

C_{eff} = 3 c.p.=2
Voltage = 2.9
Throughput = 2
Power = 25 (x2 reduction)

Pipelining!!!!!!
Further Critical path reduction;
System works at half speed

SEBC-L8 MZ 18

Yu = X(n) + AX(n-1) + A^2Y(n-2)
>> al posto che fare due moltiplicazioni per A, facendo l'unrolling posso moltiplicare direttamente per A^2
>> con la ristrutturazione del loop, ora ho diminuito il percorso critico da 4 a 3
>> quindi questa volta quando abbasso la frequenza, posso ridurre anche la tensione di alimentazione.

Potrei mettere qui un livello di pipe (barriera di registri), riducendo il percorso critico (c.p. nelle slide) a 2 (ok, aumenta un po' la latenza) e abbassare la frequenza.
>> posso avere il doppio del throughput con lo stesso consumo, o lo stesso consumo con la metà della potenza.
>>se gli oggetti un po' perdono (p_leackage) in verità il risparmio sarà un po' minore

Speed-Up transformations for Vdd Reduction: Comments:

- Applying "groups" af transformations, rather than isolated ones, almost always results in better power optimization.
  - The probability of being stuck in local minima is minimized

SEBC-L8 MZ 22

Transformations for Ceff Optimization

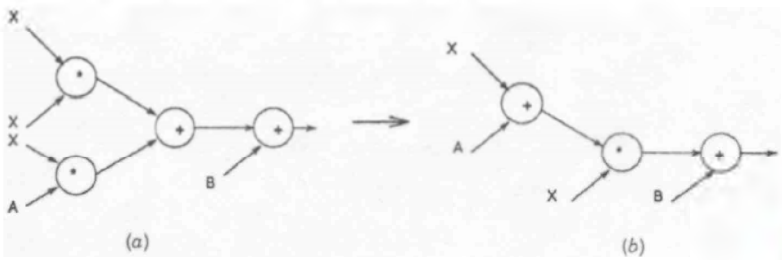
- Operation reduction:
  - The easiest way to reduce the total switched capacitance consists of reducing the number of operations in the CDFG.
  - Transformations targeting the reduction in the number of operations may affect performance.
  - Two examples:
    - Evaluation of a second order polynomial
    - Evaluation of a third order polynomial.

SEBC-L8 MZ 23

Esistono trasformazioni per ridurre la potenza, sia che riescono a mantenere invariato il throughput, sia che lo devono diminuire

Operation Reduction Methods

- Reduce # operators in data flow graph
- Computes $X^2 + AX + B$
- Same critical paths but reduced number of #op
- Reduction maintaining throughput:



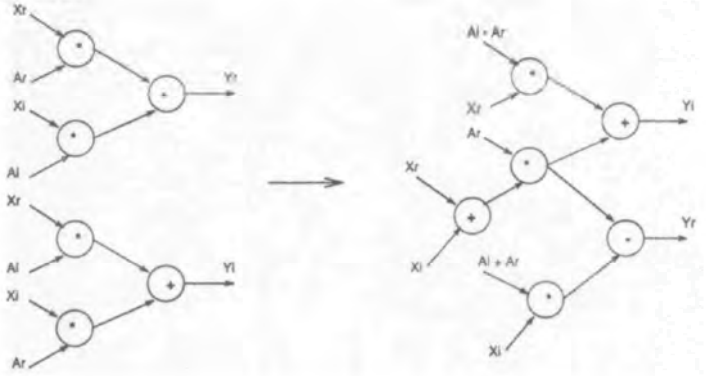
SEBC-L8 MZ 24

Ricordo che stiamo facendo modifiche comportamentali, quindi tutto ciò potrebbe ad esempio riguardare tutte modifiche SW

Mod by Giorgio Fissore, pag 191

Operation Substitution Methods

- Apply algebraic transformations:
  2 adders, 1 subtractor, 3 multipliers, critical path of length 3.
- 1 adder instead of 1 multiplier, but smaller Vdd scaling allowed.



SEBC-L8 MZ 28

Transformations for Ceff Optimization

- A typical transformation in the category of operation substitution is the conversion of mplications with constants into shift-add operations.
- Large applicability for DSP circuits, where multiplications with constants are quite common. For example in a FIR Filter:

	BEFORE (Mply)		AFTER (Add&Shift)	
	Switched Cap.	% of total Cap	Switched Cap.	% of total Cap
EX.Unit	739.65	64.80	93.07	21.63
REG/CLK	179.57	15.73	161.40	37.50
CTRL	65.45	5.73	83.79	19.47
INTERCONN.	156.69	13.64	92.10	21.40
TOTAL	1141.36	100.00	430.36	100.00

SEBC-L8 MZ 29

Al posto delle moltiplicazioni diventa: "fai la somma, poi shifta di 2, somma di nuovo e shifta di nuovo,.."
E' possibile che si riduca in questo modo anche il costo delle interconnessioni, poichè il mtply potrebbe essere più lontano dalla ALU.
Questo metodo risulta vincente in quanto come si vede qui riduce di più del 50% il costo dell'operazione!
-Oltretutto, non è nemmeno detto che il moltiplicatore vada più veloce del somma e shift.
-Quest'ultimo infatti, nel caso di somma di una costante può essere molto più veloce di un moltiplicatore parallelo.
(non sono sicuro di aver scritto tutto giusto nell'ultimo punto)

Transformations for Ceff Optimization

- Optimization of resource usage:
  - It is often possible to reduce the amount of required hardware, while preserving the number of control steps.
  - After application of some transformations, operations are distributed more uniformly over the available time, thus allowing a denser schedule.
- Transformations that can help for this purpose are:
  - Re-timing,
  - Associativity,
  - Distributivity,
  - Commutativity.
- The number of resources decreases, but the control logic to handle the sharing increases.

SEBC-L8 MZ 30

Mod by Giorgio Fissore, pag 193

Transformations for Ceff Optimization

- Word-Length reduction:
  - The word-length strongly affects all the key parameters of a design.
  - Reasons for minimizing the word-length when power is a target:
    - Fewer bits imply fewer switching => Lower switching capacitance.
    - Fewer bits imply that operations can be done faster => More Vdd scaling allowed
    - Fewer bits reduce the total number of transfer lines and the average interconnect length and capacitance.
  - Useful transformations are:
    - Associativity
    - Commutativity

SEBC-L8 MZ 34

La riduzione del numero di bit, è un parametro su cui possiamo lavorare parecchio (e possiamo farlo in parallelo a tutte le altre ottimizzazioni che stiamo facendo)

La nostra libertà sulla scelta del numero di bit è limitata dalla precisione necessaria, ma questa è legata all'algoritmo scelto!
>> algoritmi che richiedono meno precisione danno meno consumi!
>> tra l'altro, lavorando su parole più piccole, è possibile anche che il critical path possa essere percorso più velocemente. (magari aumenta però il numero di clk necessari)

Di tutte queste ottimizzazioni, il limite invalicabile è la precisione necessaria sul risultato, che va cmq mantenuta.

Transformations for Ceff Optimization

- In some cases both the number of power expensive operations and the required word-length can be reduced. In other cases, reduction of the word-length requires an increase in the number of operations.
- Example: Eighth-Order Avenhaus Bass-Pass Filter.
  - Direct implementation:
    - Critical path: 20 clock cycles.
    - Numerical stability requires 23-bit words.
    - Effective critical path (accounts for word-length): 980 ns.
  - Parallel implementation:
    - Critical path: 28 clock cycles.
    - Numerical stability requires 11-bit words.
    - Effective critical path (accounts for word-length): 610 ns.
    - Power = 0.25 P

SEBC-L8 MZ 35

Differential Coefficients for Finite Impulse Response (FIR) Filters

- Discrete-time Linear Time-Invariant FIR system:

$$Y_j = \sum_{n=0}^{N-1} C_n \cdot X_{j-n} \qquad (4.1)$$

  - C_i are the filter coefficients
  - N is # taps or filter length
- *Differential Coefficients Method* (DCM) reduces computations to save power
  - Uses differences between coefficients rather than direct-form computation
  - Uses various orders of differences
  - Requires more storage devices and storage accesses

SEBC-L8 MZ 36

Mod by Giorgio Fissore, pag 195

Second-Order Differences

- $\delta^2_{k\ 2/k} = \delta^1_{k\ 1/k} - \delta^1_{k\ 2/k-1}$ for $k = 2, \ldots, N-1$ (4.10)
- Coefficient expressions:

$$C_0 = C_0$$
$$C_1 = C_0 + \delta^1_{0/1}$$
$$C_2 = C_1 + \delta^1_{0/1} + \delta^2_{0/2}$$
$$C_3 = C_2 + \delta^1_{0/1} + \delta^2_{0/2} + \delta^2_{1/3}$$
$$\vdots$$
$$C_{N-1} = C_{N-2} + \delta^1_{0/1} + \delta^2_{0/2} + \delta^2_{1/3} + \cdots + \delta^2_{N-3/N-1}$$

$$C_k = C_{k-1} + \delta^1_{k-2/k-1} + \delta^2_{k-2/k} \quad \text{for } k = 2, \ldots, N-1 \quad (4.11)$$

- Needs just 2 extra storage variables and 2 extra additions per product to compute FIR output with 2nd-order differences compared with direct form computation

SEBC-L8 MZ 40

Vedere questa parte sulle dispense

Generalized m*th*-Order and Negative Differences

- *m*th-order differences require storage of m intermediate results for each product term, of size *N*, so need *mN* storage variables and m additions per product term compared with direct form
- Differences can be positive or negative
  - Possible to get absolute value of partial product with negative differences

SEBC-L8 MZ 41

Sorted Recursive Differences (SRD)

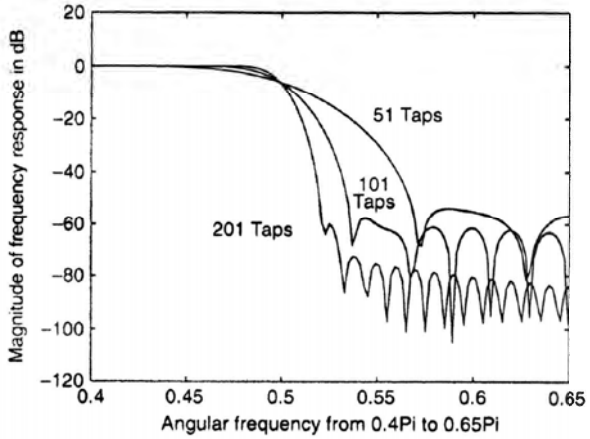
- DCM only applicable to systems where envelope generated by coefficient sequences (or differences) is a smoothly-varying continuous function
  - Mainly for low-pass FIR filters
- Recursively sort coefficients and use various orders of differences to reduce computation
- Use transposed direct form of FIR output computation
  - No restriction on applicable coefficient sequence
  - Word length reduction not the same for each coefficient

SEBC-L8 MZ 42

"differenze ricorsive ordinate"

Mod by Giorgio Fissore, pag 197

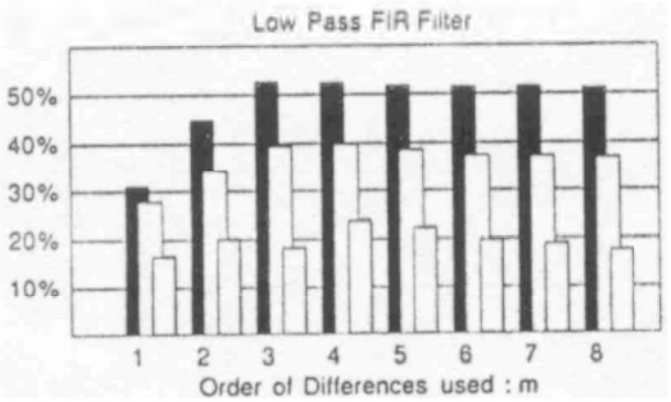
Frequency Response of SRD Low-Pass Filter and Hamming Window



SEBC-L8 MZ 46

Savings in Adds for Low-Pass Filter

- Black: N = 201, Grey: N = 101, White: N = 51



SEBC-L8 MZ 47

Savings in Shifts Using SRD

FIR System	N	Simulated S_{SHIFT} at opt m 8 bits	12 bits	16 bits	24 bits
Low pass	51	87.10	82.27	82.81	19.35
	101	95.19	94.27	93.33	83.62
	201	98.64	99.29	94.35	93.51
Band stop	51	89.13	84.83	81.22	74.14
	101	94.19	93.40	92.71	74.16
	201	98.51	89.62	92.86	93.13
Band pass	51	88.17	85.42	81.63	46.71
	101	94.80	93.03	82.51	74.28
	201	98.52	97.99	97.48	92.78
High pass	51	86.05	82.09	84.32	20.96
	101	95.56	94.85	84.60	85.15
	201	98.61	99.46	94.82	96.44

Maximum percentage savings in the net number of SHIFTs

SEBC-L8 MZ 48

Mod by Giorgio Fissore, pag 199

Multiple Constant Multiplication(MCM)

The algorithm for MCM uses an iterative matching process that consists of the following steps:

- Express each constant in the set using a binary format (such as signed, unsigned, 2's complement representation).
- Determine the number of bit-wise matches (non-zero bits) between all of the constants in the set.
- Choose the best match.
- Eliminate the redundancy from the best match. Return the remainders and the redundancy to the set of coefficients.
- Repeat Steps 2-4 until no improvement is achieved.

SEBC-L8 MZ 52

Questo algoritmo lavora con i seguenti passi:
-esprimi ciascuna costante in forma binaria
-vai a vedere tra le varie costanti quanti bit sono ad uno nelle stesse posizioni
-scegli il numero che presenta più uni comuni
-elimina la ridondanza sul best match e tieni la ridondanza e il resto dei coefficienti.
-go to loop

Example:

Constant	Value	Unsigned
a	237	11101101
b	182	10110110
c	93	01011101

Binary representation of constants

"reminder", il resto

Constant	Unsigned
Rem. of a	10100000
b	10110110
Rem. of c	00010000
Red. of a,c	01001101

Updated set of constants 1st iteration

Constant	Unsigned
Rem. of a	00000000
Rem. of b	00010110
Rem. of c	00010000
Red. of a,c	01001101
Red. of Rem a,b	10100000

Updated set of constants 2nd iteration

SEBC-L8 MZ 53

1) binario
2) la coppia a,c ha più parti comuni
3) b non lo tocchiamo, devo ridurre a e c
4) ridondanza: 01001101, la tolgo da a,c e salvo il reminder
5) il match migliore con b ora è tra rem_a,b; faccio una nuova ridondanza tra rem_a,b >> 10100000 e salvo i nuovi valori di rem_a, rem_b, red of rem_a,b e lascio invariato c

Linear Transformations

- A general form of linear transformation is given as:

$$y = T*x$$

where, T is an m by n matrix, y is length-m vector and x is a length-n vector. It can also be written as:

$$y_i = \sum_{j=1}^{n} t_{ij} x_j, i = 1,\ldots,m$$

- The following steps are followed:
  - Minimize the number of shifts and adds required to compute the products $t_{ij}x_j$ by using the iterative matching algorithm.
  - Formation of unique products using the sub-expression found in the 1st step.
  - Final step involves the sharing of additions, which is common among the y_i's. This step is very similar to the MCM problem.

SEBC-L8 MZ 54

"prodotto matrice-vettore": ogni termine del vettore moltiplicherà tutti i termini di una colonna della matrice.
Qui l'ottimizzazione può essere particolarmente efficace
>> vado a costruirmi i miei risultati come una somma di prodotti.

Mod by Giorgio Fissore, pag 201

Polynomial Evaluation

Evaluating the polynomial:

$$x^{13} + x^7 + x^4 + x^2 + x$$

- Without considering the redundancies this polynomial evaluation requires 22 multiplications.
- Examining the exponents and considering their binary representations:

  1 = 0001, 2 = 0010, 4 = 0100, 7 = 0111, 13 = 1101.
- x^7 can be considered as $x^4 \times x^2 \times x^1$. Applying sub-expression sharing to the exponents the polynomial can be evaluated as follows:

$$x^8 \times (x^4 \times x) + x^2 \times (x^4 \times x) + x^4 + x^2 + x$$

- The terms x^2, x^4 and x^8 each require one multiplication as shown below:

$$x^2 = x \times x, \quad x^4 = x^2 \times x^2, \quad x^8 = x^4 \times x^4$$

- Thus, we require 6 instead of 22 multiplications.

SEBC-L8

Oltre che nelle trasformazioni lineari questo metodo funziona bene anche nelle polinomiali.

Esprimo x^{13} come somma di potenze di due.

Queste, sono tecniche che sfruttano la ridondanza delle operazioni, riuscendo ad eliminare la ripetizione delle operazioni già eseguite, e (quando queste operazioni si trovano negli inner loop?) possono dare grandi risparmi con poca difficoltà

Sub-expression Sharing in Digital Filters

- Example of common sub-expression elimination within a single multiplication :

$$y = 0.101000101 * x.$$

This may be implemented as:

$$y = (x >> 1) - (x >> 3) + (x >> 7) - (x >> 9).$$

Alternatively, this can be implemented as,

$$x2 = x - (x >> 2)$$

$$Y = (x2 >> 1) + (x2 >> 7)$$

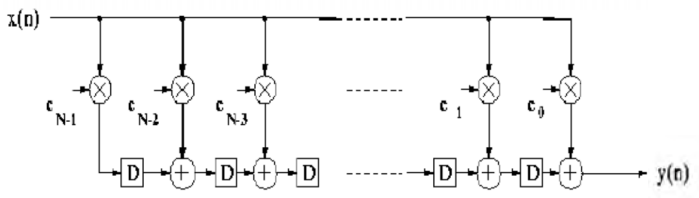
which requires one less addition.

SEBC-L8 MZ 59

- In order to realize the sub-expression elimination transformation, the N-tap FIR filter:

$$y(n) = c_0x(n) + c_1x(n-1) + \ldots + c_8x(n-N+1)$$

must be implemented using transposed direct-form structure also called data-broadcast filter structure as shown below:



SEBC-L8 MZ 60

Mod by Giorgio Fissore, pag 203

- Remove each occurrence of each sub-expression and replace it by a value of 2 or -2 in place of the first (row major) of the 2 terms making up the sub-expression.

-1		2		1						2	
					-2		-1		-1		-2
	-2										
						1		-1			

- Record the definition of the sub-expression. This may require a negative value of shift which will be taken care of later.

$$x3 = x1 - x1[-1] >> (-1)$$

SEBC-L8 MZ 64

- Continue by finding more sub-expressions until done.

-1		3								2	
					-3						-2
	-2										
						1		-1			

5. Write out the complete definition of the filter.

$$x2 = x1 - x1[-1] >> (-1)$$
$$x3 = x2 + x1 >> 2$$
$$y = -x1 + x3 >> 2 + x2 >> 10 - x3[-1] >> 5 - x2[-1] >> 11$$
$$-x2[-2] >> 1 + x1[-3] >> 6 - x1[-3] >> 8.$$

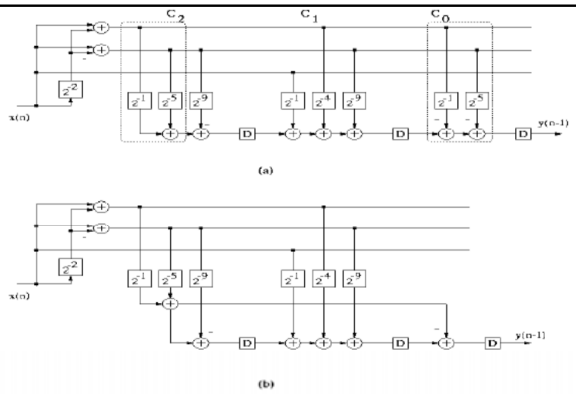
SEBC-L8 MZ 65

- If any sub-expression definition involves negative shift, then modify the definition and subsequent uses of that variable to remove the negative shift as shown below:

$$x2 = x1 >> 1 - x1[-1]$$
$$x3 = x2 + x1 >> 3$$
$$y = -x1 + x3 >> 1 + x2 >> 9 - x3[-1] >> 4 - x2[-1] >> 10$$
$$- x2[-2] + x1[-3] >> 6 - x1[-3] >> 8.$$

Mod by Giorgio Fissore, pag 205

SEBC-L8 MZ 66



3-tap FIR filter with coefficients $c_2 = 0.101010\bar{1}010\bar{1}$, $c_1 = 0.10010100101\bar{1}$ and $c_0 = 0.\bar{1}0\bar{1}0\bar{1}010000$. 2 additions in the dotted square in (a) are shared in (b). Filter requires only 7 additions and 7 shifts as opposed to 12 adds and 12 shifts in standard multiplierless implementation.

SEBC-L8 MZ 70

Behavioral synthesis for Low power

- Objective: Automatic translation of the behavioral specification of a digital system into a RTL description.
  - Subject to a given set of design constraints.
  - The RTL description consists of a control unit and some data-paths.
  - The (possibly partitioned) control unit is generated using common logic synthesis techniques.
  - The data-paths are mapped onto a library of macro-components.
- Three main steps:
  - Operation scheduling.
  - Resource allocation.
  - Binding (or resource sharing)

SEBC-L8 MZ 71

Behavioral synthesis for Low power

- Traditional objectives of behavioral synthesis:
  - Resource minimization under throughput constraints.
  - Throughput maximization under resource constraints.
- Behavioral synthesis for low power includes power in the cost function used for optimization.

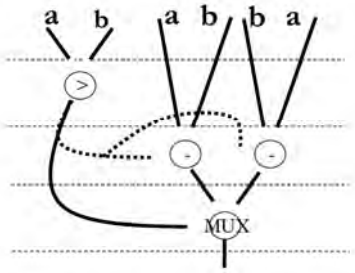
L'esigenza di progettare lowpower, ha cambiato l'approcio con cui ci si avvicina ad un progetto:
-Prima era sempre un progetto che (da un throughput fissato) portava alla minimizzazione delle risorse; o come seconda strada c'era "le risorse sono fissate, vai a cercare la soluzione che massimizzi il throughput"
-ora invece, è presente questa nuova variabile e bisogna spostarsi sul piano consumo-ritardo, per trovare la soluzione più ottimizzata

Mod by Giorgio Fissore, pag 207

SEBC-L8 MZ 72

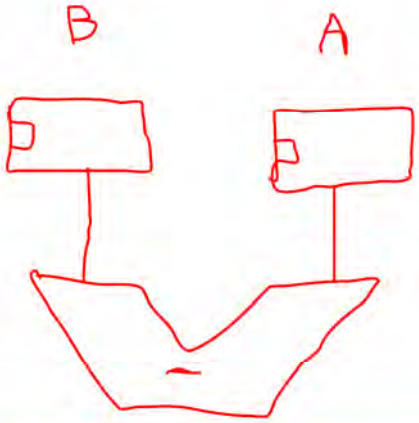
Operation scheduling for Low power

- Example: computation of |a-b|.
  - Second solution: Scheduling using three control steps.
    First option: Unshared subtractor.



- The two subtractors are never active simultaneously

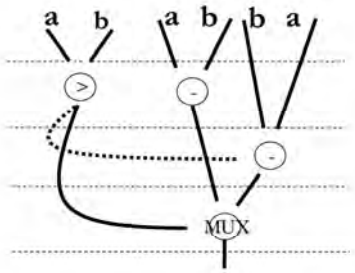
SEBC-L8 MZ 76



Se invece al primo clk faccio la comparazione, delle due sottrazioni, ne farò sempre solo una!
-il ciclo aumenta a 3 clk, ma si farà sempre un'operazione in meno

Operation scheduling for Low power

- Example: computation of |a-b|.
  - Second solution: Scheduling using three control steps.
    Second option: Shared subtractor.



- Only the computation of *b-a* can be sometimes be avoided

SEBC-L8 MZ 77

Altra alternativa, volta all'utilizzo di un solo sommatore, dato che questo ha cmq una p_leakcage anche se non viene utilizzato:
-comparo, e nel frattempo faccio una sottrazione (che mando ad un ingresso del mux).
-poi, se invece b<a, faccio l'altra operazione che mando all'altro ingresso del mux, altrimenti niente;
-con il comparatore seleziono l'ingresso del mux da far uscire.
Quale convenga rispetto a prima, dipende da tecnologia, stat_ingressi,..

Operation scheduling for Low power

- Automatic scheduling:
  - *Basic intuition*: Schedule in the same time interval the operations that are mutually exclusive.
  - *Advantage*: Only one operation performs some useful computation in that time interval.
  - *Requirement*: The schedule must satisfy the throughput and resource constraints.

SEBC-L8 MZ 78

Mod by Giorgio Fissore, pag 209

Multiple Supply Voltage Scheduling

- Simultaneous powering-up a chip with different supply voltage offers a good opportunity for power optimization.
- Advantage:
  - Low algorithm and/or architectural costs.
- Technological constraint:
  - Availability of multiple Vdd voltages on the chip

SEBC-L8 MZ 82

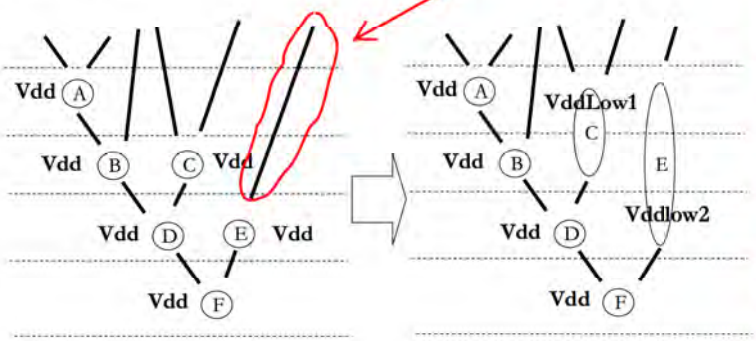
Multiple Supply Voltage Scheduling

- Basic idea:
  - Power the modules on the critical paths at the highest possible voltage, so as to preserve speed.
  - Lower the supply voltage of the modules which are not of the critical paths, so as to reduce power.
- The use of level shifters at the boundaries of the different modules may be needed.
- The area and power costs due to such shifters must be taken into account.

SEBC-L8 MZ 83

Multiple Supply Voltage Scheduling

- Example:



SEBC-L8 MZ 84

E potrebbe essere spalmato su 3 clk

Questo metodo
+ permette di risparmiare molto, laddove posso disporre di più alimentazioni
(+ le ottimizzazioni qui possono essere analizzate con gli stessi algoritmi usati per l'analisi del timing)
- aumento la complessità del mio integrato:
-abbiamo bisogno di più alimentazioni (tecnologia più avanzata in grado di gestirle)
-ora, C manda la sua uscita a D, che lavora con una tensione diversa >> i due blocchi non sono più compatibili, ma ho bisogno di level shifter (che portano via area e consumano) e che rendono compatibili i due livelli alti

Lecture 9 System-Level Power Optimization

- Motivation - the compelling need for low power systems
- Power reduction at
  - conceptualization and modeling levels
  - design level - design of power efficient
    - hardware units
    - memories and
    - communication buses
- Conclusions

SEBC-L9 MZ 1

Appunti di Giorgio Fissore
Disponibili in centro stampa

La regola del "prima di progettare, prendi un foglio di carta e fai uno schema" funziona anche e molto per il low power.
La scelta del modello di sistema, può infatti dare dei risparmi mostruosi!
Ma come fare, quando non si sa ancora nemmeno come si faranno le cose?

Ogni algoritmo lavora basandosi su questi tre parametri:
-processing (elaborazione)
-interconnection (scambio dati)
-storaging (memorizzazione)

Power Reduction Techniques

- Static techniques for low power
  - Applied at conceptualization and design time
  - Synthesis for low power
  - Compilation for low power
- Dynamic techniques for low power
  - Dynamic power management (DPM) - use run time behavior to reduce power consumption when system is serving light load or when idle
    - Dynamic voltage scaling (DVS) - change voltage at run time to manage power
    - Shutdown unused I/O devices, NIC, display or HDs

SEBC-L9 MZ 2

Esistono due macro-tipi di ottimizzazioni che si possono fare: Statiche e Dimamiche
-Statiche: sono quelle che abbiamo usato fin ora;
-Dinamiche (posso fare miglioramenti molto più significativi):
-Dynamic Voltage Scaling (DVS), adatto il consumo di potenza al carico computazionale volta per volta (runtime)>> sistema usatissimo su tutti i up moderni.
-spengo i moduli inutilizzati, dopo un certo periodo di inattività, e li risveglio all'arrivo di nuovi dati.
>> in ogni caso, non so in partenza quanto risparmierò, ma si vedrà durante l'utilizzo)

Hardware Technologies for Low Power

$$P_{dyn} = f \times V_{dd}^2 \times C_L \times \alpha$$

- Very low supply voltage technologies.
- Multiple supply voltages on a single chip.
- Techniques for handling dynamically variable supply voltage and/or clock speed.

SEBC-L9 MZ 3

Mod by Giorgio Fissore, pag 213

Specification and Implementation Models

- Functional models
  - addresses functionality and requirements
  - executable (VHDL, C++, Java; for simulation) or non executable (task graph)
- Implementation models
  - describe the target realization for systems.
  - system complexity: modular, component oriented, hierarchical.
  - Implementation models for energy efficient systems modelling:
    - Spreadsheet model expresses a combination of components and evaluates overall energy budget
    - Power state machine model captures the power consumption of systems and their constituents as they evolve through a sequence of operational states.

SEBC-L9

I modelli implementativi permettono di descrivere un sistema mettendo in evidenza le caratteristiche di consumo legate alla complessità del sistema (la modularità, la gerarchia,..).
Come descriverli? due tecniche:
-Spreadsheet: stimo il consumo partendo da dei valori implementati (es. faccio una ALU a 4,8,16 bit e vedo quanto consumano -con una funzione interpolante posso stimare anche i valori intermedi-
poi scelgo una memoria a 1 GB, 2GB,.. e vedo il consumo >> poi scelgo quali sono le combinazioni più ottimizzate)
-Power State Machine: faccio un pallogramma in cui ogni stato non è legato tanto a cosa fa, quanto più a quanto consuma (tra l'altro il passaggio tra stati consuma)

Energy Efficient Design from Executable Functional Models

- **Algorithm selection for low power**
  - For a given common function, make a library of multiple different algorithms
  - Characterize each algorithm with performance & power
  - Perform system optimization by:
    - heuristic to select an implementation algorithm and supply voltage that trades off performance for power.
- **Algorithm computational energy**
  - Computational energy of the algorithm can be estimated using CDFG
  - Characterize each elementary operation with a computational energy metric
  - Compose rules to compute energy cost of a complex CDFG.
  - Energy of elementary operations is obtained by assuming implementation styles and extracting cost per operation through experiments.

SEBC-L9 MZ 8

Energy Efficient Design from Executable Functional Models

- **Algorithm communication and storage energy**
  - communication and storage cost is hidden in specifications
  - storage and communication energy: relate to locality of computation data
  - data variables with long life time -> increased storage need -> more power
  - problem: locality analysis from CDFG is hard, information not explicitly available
- **Computational kernels**
  - are the inner loops of an algorithm where most of the time is spent during execution
  - extract them by profiling data on executable system level model
  - implement on dedicated power-optimal hardware
  - during execution of kernel, rest of system can be shutdown, hence save power.

SEBC-L9 MZ 9

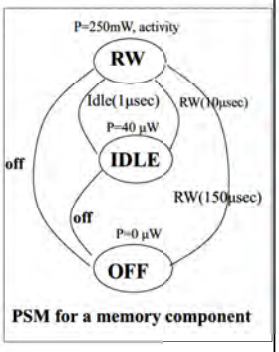
Andiamo a stabilite un costo della comunicazione con la memoria. Può essere fatto anche in maniera molto semplice, es
-più sono lontani da memoria, più costa
-life time delle variabili: più sono lunghi, più ho bisogno di mem grande, più costa

Computational kernels:
es se sto eseguendo un inner loop, nel frattempo posso spegnere il resto

Mod by Giorgio Fissore, pag 215

Energy efficient design from implementation models

- **Power State Machine (PSM)**
  - State based model for system components
    - states represent modes of operations
    - arcs represent legal transitions between op. modes
    - states are labelled with power dissipation values
    - transitions are labelled with triggering events, energy costs and transition times.
  - Advantages:
    - study how system reacts to different workloads
    - model interactions between components
    - analyze the effects of power management.
  - Drawbacks: complex component model



PSM for a memory component

SEBC-L9 MZ 13

>> andare a capire quale sia il set di transizioni migliori per realizzare le varie funzioni, muovendo il pallogramma della power state machine.
Questa caratterizzazione può essere fatta anche in base al carico di lavoro da eseguire.

Low Power Application Specific Units

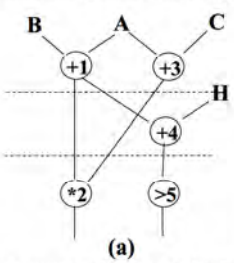
- Usually give better power efficiency, but have low flexibility
- Power reduction techniques: low power RTL, logic level and physical level techniques
  - Power Driven Voltage Scaling (*PDVS*) and scaling down *Vdd* reduce power but performance may diminish.
  - multiple supply voltages on a single chip (globally asynchronous and locally synchronous systems (GALS))
  - reduce clock frequency, load capacitance and switching activity.
  - set clock frequency of a component that is not performing useful work to zero and nullify dynamic power consumption of that component
- A. Hemani: transformed single clock industrial designs into GALS - 70% power reduction

SEBC-L9 MZ 14

Scelta numero 1: utilizzo di unità application specific, con l'obiettivo di ridurre i consumi.
(es faccio un processore specifico per eseguire un determinato algoritmo).
-E' la scelta che ci può dare più vantaggi
-ma pago una riduzione della flessibilità
>> vedremo tra poco il legame tra consumo e flessibilità.

Low Power through Switching Activity Reduction

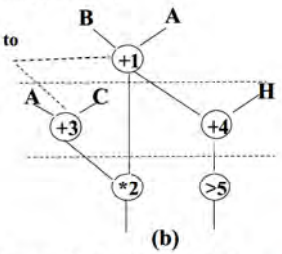
- Reduce the number of basic operations, ⇒ transform DFG to minimize the number of operations
- Reduce switching of the inputs to functional unit (FU) ⇒ increase correlation between successive patterns at the input of FU.
- Scheduling and binding for reduced switching activity



(a)

Less power efficient schedule and binding

+1 and +3 bound to the same adder



(b)

Power efficient schedule & binding

SEBC-L9 MZ 15

Mod by Giorgio Fissore, pag 217

Design of Power-Efficient Memory Subsystems

- Memory accesses are slow and consume more power with increasing memory size
  - reduce memory storage requirements of the applications
  - during system conceptualization use principle of temporal locality to reduce memory storage requirements
  - improve locality and reduce need for temporary storage of results of computation by consuming them ASAP
  - Reduce memory need by data compression
- Advanced hierarchical memory architectures for low power

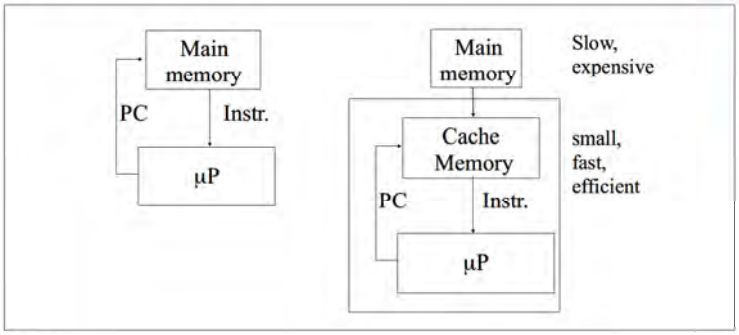
SEBC-L9 MZ 19

Locality of Reference

Fetching data and instructions from local rather than global resources reduces access cost (interconnect, access energy)

Prime example: memory hierarchy

register files, caches, instruction loop buffers, memory partitioning, distributed memory

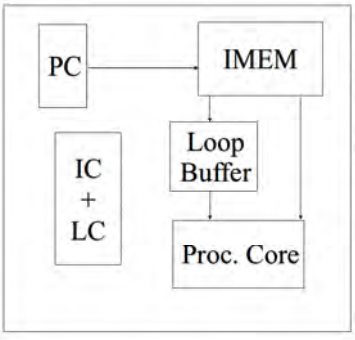


SEBC-L9 MZ 20

Usare una gerarchia di memoria è utile nella stessa maniera del pipelining: permette alternativamente sia di andare più veloce che risparmiare di più.

Locality of Reference

- (Hardware) instruction loop buffer



- On first iteration, code cached in loop buffer
- Fetched from loop buffer on subsequent iterations
- Popular feature in DSPs

SEBC-L9 MZ 21

la gerarchizzazione della memoria può ancora essere portata avanti.

Se vado a vedere come sono fatti i comandi da eseguire, si nota che la maggior parte sono fatti da loop fatti da pochissime istruzioni: se allora metto i dati del loop, dentro ad un loop buffer vicinissimo al uP, consumo meno e vado ancora più veloce che nella cache.

Mod by Giorgio Fissore, pag 219

Memory design

- Memories and caches of a digital system usually account for a large fraction of the total system power.
  - Example: L1 and L2 caches of a DEC Alpha chip dissipate 25% of the total power.
- Tagets:
  - Minimization of power due to memory accesses.
  - Minimization of power due to data transfers.

SEBC-L9 MZ 25

prima considerazione: tutti i PE che devono fare elaborazioni sui dati, richiedono tipicamente una grande quantità di memoria. Per ridurre i consumi questa va allora inserita molto vicina alla cpu. >> grandi quantità di cache.
In molti sistemi, la cache consuma tantissimo (es 25% del totale)

Memory design

- Minimization of memory access power:
  - Fixed memory access patterns:
    - Optimize memory hierarchy
  - Fixed memory architecture:
    - Optimize memory access patterns.
  - Concurrent optimization of memory architecture and access patterns is still an open issue.
- Minimization of information transfer power:
  - Code density optimization.
  - Data density optimization.

SEBC-L9 MZ 26

Conoscendo l'ordine con cui faccio i vari accessi, quante volte accedo ai vari dati, posso gerarchizzare la memoria.

Se invece conosco l'architettura della memoria, allora devo ottimizzare i pattern di accesso ad essa, in maniera da utilizzare il più possibile i blocchi vicini ai PE

Minimization of Memory Access Power

- Basic concept: ”Close” vs. ”far” memory accesses:
  - Close: Faster, less energy consuming, smaller block sizes.
  - Far: Slower, more energy consuming, larger block sizes.

SEBC-L9 MZ 27

Mod by Giorgio Fissore, pag 221

Cache/Memory Partitioning

- Multi-bank caches:
  - Use independently-addressable banks.
  - Two dimensional partitioning: M modules with B banks each.
  - Power savings achieved through exploitation of reduced capacitance of smaller memories.
  - Ad-hoc: low-power bank selection circuitry is used.
- Partitioned memories:
  - Exploit sleep-mode features to shut down individual banks
  - Design memory partition so as to maximize the sleep-time.
  - Typical memory traces are used to drive the partitioning process.

SEBC-L9 MZ 31

Più la cache è grande, più l'accesso ad essa consuma.
>> l'idea è di partizionarla in blocchi
>> devo però cercare di accedere ad un solo blocco alla volta, in maniera da far dormire gli altri
>> queste tecniche sono quindi tanto più utili, quanto più conosco il tracing degli elementi, e posso così dividerli tra i vari blocchi in maniera da accedere sempre e solo ad uno (funzionano molto bene nei sistemi embedded)
>> ottimizzazione fondamentale.

Cache/Memory Partitioning

- In the case of embedded systems, the dynamic memory access profile may be available.
- Map most frequent addresses onto small portions close to the processor.
- Since energy per access increases with memory size... partition memory into banks!
  - The energy savings obtained must compensate the overhead of adding banks (longer wires, bank selection logic, etc..)

SEBC-L9 MZ 32

Memory Access Pattern Optimization

- Address sequentialization:
  - Exploitation of multiple (smaller) memories.
  - Low-transition bus encoding can also be viewed as a tool for making addresses sequential (i. e. ..... Gray coding).
- Localization of execution:
  - Ad-hoc memory (or cache) for storing frequently executed code.

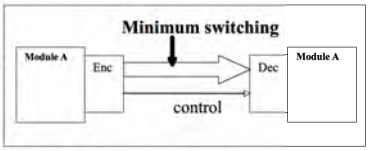
SEBC-L9 MZ 33

Se io so che in una memoria metto del codice, tanto meglio farò un accesso sequenziale, tanto meglio sarà ottimizzato il sistema.

Mod by Giorgio Fissore, pag 223

Low Power Communication Resources

- At physical level communication power is reduced :
  - scaling down the voltage swing on the high capacitance wires of the bus
  - scaling down the average number of signal transitions
- Low power data encoding
- Arbitration protocols
  - bus access control
  - reduce bus power by scheduling & binding highly correlated data streams consecutively on the bus



SEBC-L9 MZ 37

Si tende al giorno d'oggi ad utilizzare, al posto di una batteria di capacità C, due batterie di capacità C/2, per varie ragioni.

Battery-Driven Power management

- Multi-battery systems are becoming popular giving a new flexibility in power supply configuration.
- Usually batteries are discharged one after the other.
- Problem: under a fixed discharge rate, the battery lifetime does not scale linearly with its capacity (one battery with C capacity has a lifetime greater than two batteries – C/2 capacity).
- As a consequence multi-batteries seemed penalized ..but...
- A battery recovers some deliverable charge when it has some rest..... schedule the active battery source to give rest time..

SEBC-L9 MZ 38

Ciò porta ad un problema: se lavoro con la batteria C e la scarico a rateo di scarico costante (lineare) questa durerà un certo tempo T.
Se invece faccio lo stesso con le due C/2 scaricandole allo stesso rateo una alla volta, si vedrà come questa durerà meno di T (meno di T/2 per scaricare la singola batteria).
Ciò è dovuto al fatto che quando la batteria sembra scarica, in realtà ha ancora dell'energia disponibile, che se aspetto potrò utilizzare.
>> il trucco è: dopo aver usato una batteria per un po' la lascio riposare, poi devo utilizzare un po' l'altra, e torno poi sulla prima
>> devo fare uno scheduling anche delle batterie!! che decida quando collegarmi ad un'altra batteria, e qual'è il tempo migliore di uso di una batteria stessa.

Battery-Driven Power management

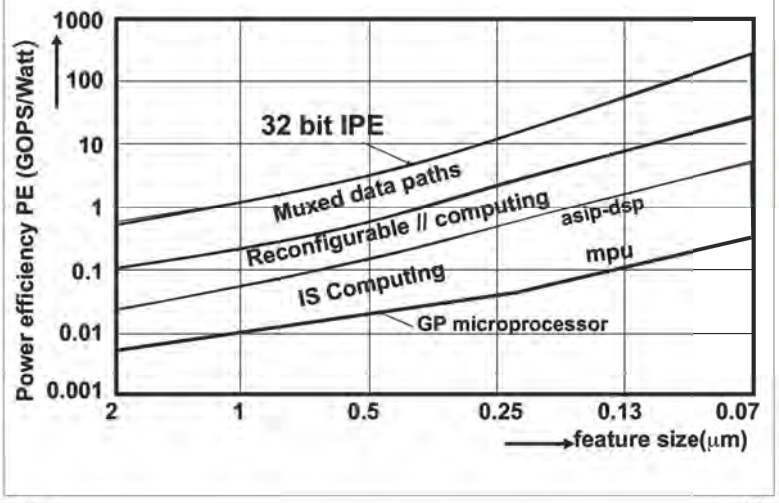
- Issues in battery scheduling:
  - when replacing the battery
  - which battery must be connected
- This can be done by a Static or a Dynamic scheduling
- Static Scheduling:
  - The order is fixed; as the "switching frequency" increases, lifetime tends asimptotically to that of a monolithic battery with the same capacity.
  - For each battery select a rest time comparable to the time constant of the battery itself (order of seconds).
- Dynamic Scheduling:
  - Rest times are adapted to the actual battery conditions.
  - Better for etherogeneus battery systems and irregular curr

SEBC-L9

Questi scheduling sono di due tipi: statici e dinamici.
-in quello statico, stabilisco un ordine con cui fare questo battery switching (trovare l'algoritmo che statisticamente riesce a far vedere questo insieme di batterie il più vicino possibile ad una batteria singola)
+vantaggio: lo faccio una volta per tutte
-nel dynamic scheduling, con algoritmi molto più complicati, le decisioni sono prese runtime in base allo stato delle batterie.

In funzione del carico può essere meglio lavorare con una batteria piuttosto che con un'altra (una rende meglio per piccole correnti, un'altra si stanca in fretta, ma può fornire efficientemente correnti più alte,..)

The Cost of Flexibility – Evolution

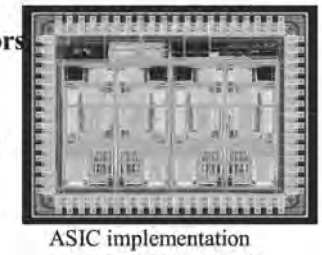


SEBC-L9 MZ 43

Quanto visto prima si mantiene anche con l'evoluzione della tecnologia: si vede come l'efficienza in generale aumenta, ma i tre ordini di grandezza tra sw e hw dedicato, si mantiene.

The Cost of Flexibility – Example

Least-Mean-square Pilot Correlators for CDMA
(1.67 MSymbols Data Rate)
Complexity: 300 Mmult/sec and 360 Mmac/sec



ASIC implementation
1.2-2.4 GOP @ 12 mW

Architecture comparison – single correlator

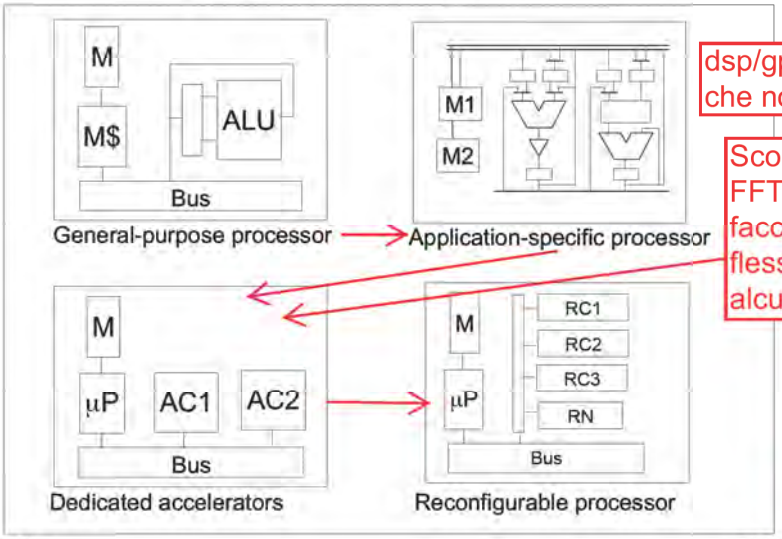
Type	Power	Area
Commercial DSP	460 mW	1100 mm²
Configurable Proc.	18 mW	5.5 mm²
Dedicated	3 mW	1.5 mm²

[Ref: N. Zhang, PhD'01]

SEBC-L9 MZ 44

Oltretutto, non solo il consumo viene abbattuto, ma anche l'area!! (ovvio poichè in hw dedicato, non c'è tutta la parte di gestione delle istruzioni, ci sono parti inutili,...)
>> la flessibilità ha un costo improponibile in sistemi low power.

The Architectural Choices



SEBC-L9 MZ 45

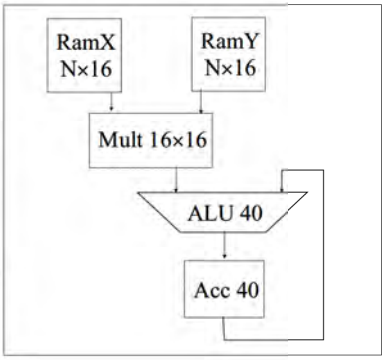
dsp/gpu > magari qui abbiamo ancora parti di hw che non servono (magari non usiamo il mac)

Scopro che il mio programma deve fare anche la FFT >> mi faccio fare il blocchetto dedicato che faccia quell'operazione >> mantengo la flessibilità del uP, ma ottimizzo l'esecuzione di alcune istruzioni ricorrenti (tramite acelleratori)

Mod by Giorgio Fissore, pag 227

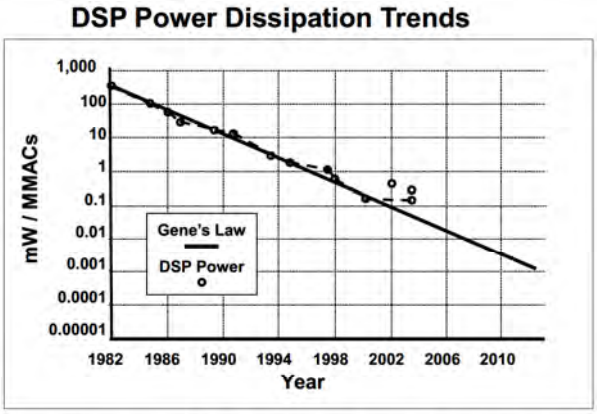
Example 1: DSPs

- The first type of application-specific processor to become popular
- Initially mostly for performance, but energy benefit now also recognized
- Key properties: dedicated memory architecture (multiple data memories), data path specialized for specific functions such as vector multiplies and FFTs
- Over time: introduction of more and more concurrency (VLIW)



SEBC-L9 MZ 49

DSPs Deliver Improving Energy-Efficiency



Energy efficiency of DSPs doubles every 18 months ("Gene's Law"), but…

SEBC-L9 [Ref: G. Frantz, TI] MZ 50

Performances of DSP Processors

DSP Proc.	1982	1992	2002	2012 (?)
Techno (nm)	3000	800	180	20
# Gates	50K	500K	5G	50G
V_{DD} (V)	5.0	5.0	1.0	0.2
GHz	0.020	0.08	0.5	10
MIPS	5	40	5K	50K
MIPS / W	4	80	10K	1G
mW / MIPS	250	12.5	0.1	0.001

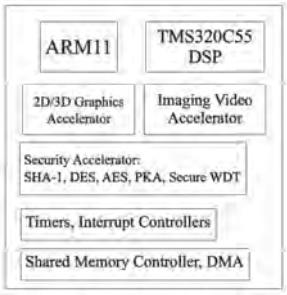
Mod by Giorgio Fissore, pag 229

SEBC-L9 [Ref: G. Frantz, TI] MZ 51

Hardware Accelerators

Often executed functions implemented as dedicated modules and executed as co-processors

- Opportunities: Network processing, MPEG Encode/Decode, Speech, Wireless Interfaces
- Advantage: Energy-efficiency of custom implementation
- Disadvantage: Area-overhead



Example: Computational core of Texas Instruments OMAP 2420 Platform™

[Ref: OMAP Platform, TI]

SEBC-L9 MZ 55

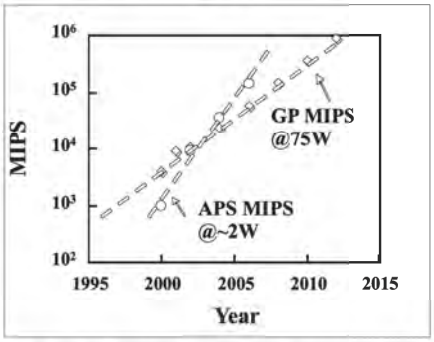
>>
-ho un processore generale che serve per interfacciarmi con l'esterno,..;
-un dps per alcune parti
-molteplici acceleratori HW

Hardware Accelerators

TCP Offload Engine

2.23 mm × 3.54 mm, 260K transistors

Example: networking coprocessor

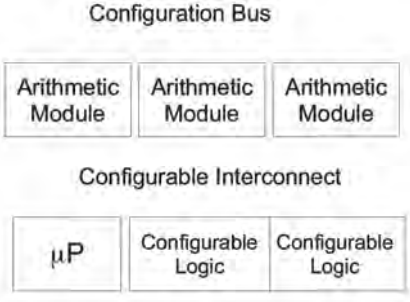


GP = General Purpose processor
AP Accelerator processor

[Courtesy: S. Borkar, Intel'05]

SEBC-L9 MZ 56

(Re)configurable Processors



"Programming in space"
Create dedicated co-processors by reconfiguring interconnect between dedicated computational models.

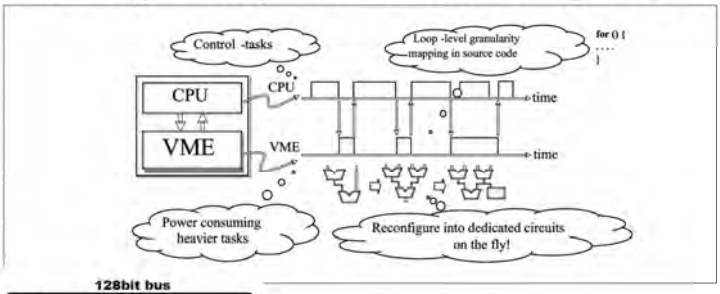
Efficiency of hardwired accelerators, but increased flexibility and reuse (smaller area)

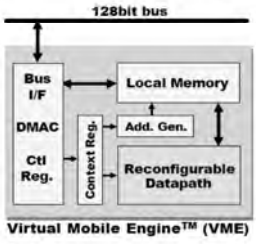
[Ref: H. Zhang, JSSCC'00]

SEBC-L9 MZ 57

Mod by Giorgio Fissore, pag 231

Example: Sony Virtual Mobile Engine (VME)





Virtual Mobile Engine™ (VME)

- Dynamic Reconfigurable vector engine
- Reconfigured on the fly
- One cycle context switch
- Coarse grain heterogeneous type
- Native 24bit data-width
- Max Clock Freq. 166MHz
- Deployed in portable music and game players

Other examples: ADRES, Cluster, CoolDSP, SiliconHive

[Ref: K. Seno, HotChips'04]

SEBC-L9 MZ 61

Remember: Amdahl's Law Still Holds

- Effectiveness of alternative architectures (ASIP, Accelerator, Reconfigurable) determined by the amount of code spawned from GP (general processor)
- Mostly effective for repetitive kernels
- 80-20% rule typically seems to apply
- Transformations can help to improve effectiveness
- Most important: code development and algorithm selection that encourages concurrency

questa legge dice quant'è lo speed up che posso avere in funzione della parallelizzabilità del mio algoritmo.
>> più è parallelizzabile, più posso ottimizzare rispetto all'esecuzione seriale.

SEBC-L9 MZ 62

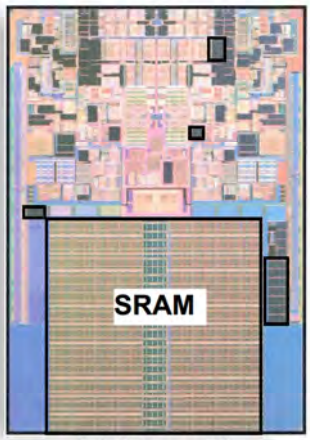
Summary and Perspectives

- Architectural and algorithmic optimization can lead to drastic improvements in energy-efficiency
- Concurrency is an effective means to improve throughput at fixed currency or reduce energy for fixed throughput
- Energy-efficient architectures specialize the implementation of often recurring instructions or functions

Mod by Giorgio Fissore, pag 233

SEBC-L9 MZ 63

Processor Area Becoming Memory Dominated



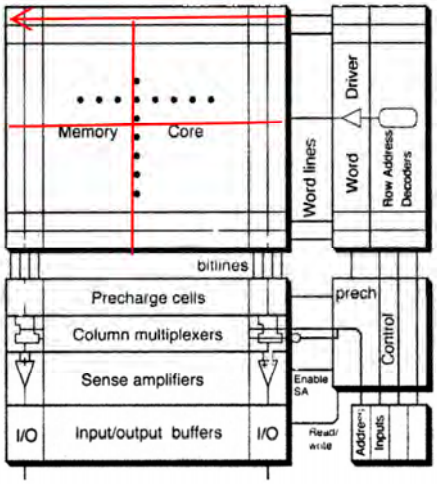
Intel Penryn™
(Picture courtesy of Intel)

- **On chip SRAM contains 50-90% of total transistor count**
  - Xeon: 48M/110M
  - Itanium 2: 144M/220M
- **SRAM is a major source of chip static power dissipation**
  - Dominant in ultra-low power applications
  - Substantial fraction in others

Se si analizzano un po' di processori commerciali, si vede come la SRAM occupa la maggior parte dei loro transistor (50-90%)!! se ciascuno di questi perde anche solo una goccia, il consumo di potenza è enorme!

SEBC-L10 MZ 4

SRAM Organization



la bit line è molto caricata (capacità enorme), quindi il sense amplifier è fondamentale.
Più la memoria è grande, più consuma, poichè la capacità da caricare/scaricare aumenta.

Se posso accedere ad un pezzo di memoria alla volta, vado più in fretta e consumo di meno! (in maniera quadratica)
>> la memoria viene divisa in molti banchi

SEBC-L10 MZ 5

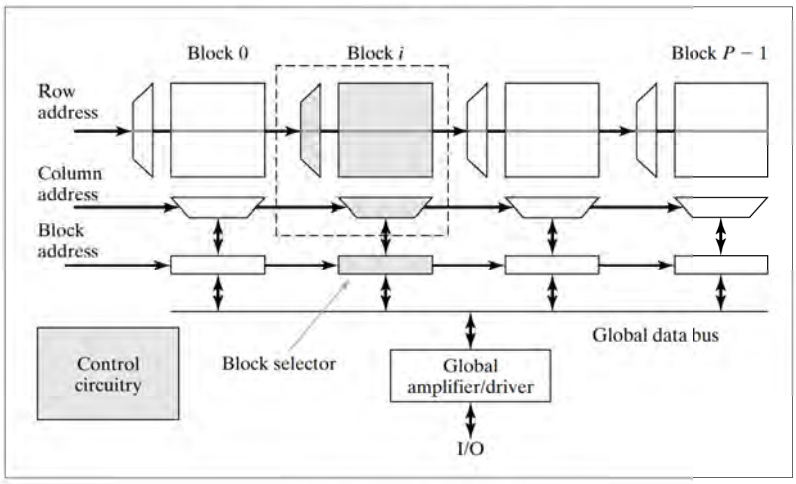
Banked SRAM Organization

- Reduces switched capacitance, reduces power, increases speed
- *R* x *C* SRAM means that any access to row:
  - Enables *R* rows
  - Enables all bit lines
  - C_{cell} is individual cell capacitance
  - Causes *R* x *C* x C_{cell} capacitance to be switched
- Solution: Split memory into *B* banks
  - Only 1 bank enabled for an access, not all banks
  - Switched capacitance now *R* x *C* x C_{cell} / B

Mod by Giorgio Fissore, pag 235

SEBC-L10 MZ 6

Basic Memory Structures



[Ref: J. Rabaey, Prentice'03]

SEBC-L10 MZ 10

SRAM Power Reduction

- Reduce power (in general) by:
  - Lowering switched Capacitance
  - Lowering voltage swing
  - Lowering activity factor
  - Lowering operation frequency
- Easiest to lower voltage swing in memory on bit/word lines
- Limited by:
  - Inability to resolve small voltage differentials at adequate speed
  - Increasing soft bit error rates and degraded signal integrity

SEBC-L10 MZ 11

Appurato che la memoria va divisa in banchi, vediamo ora come ridurre il consumo del singolo banco.
Solite cose:
-ridurre capacità che commutano
-ridurre la dinamica dei segnali che commutano in uscita
-riduzione frequenza operativa
-ecc
Con dei limiti, però:
-non ridurre troppo Vdd per non perdere i dati (minimum data retention voltage)
-necessità di bit aggiuntivi per ricostruire il valore corretto in caso di errori

SRAM Metrics

- Functionality
  - Data retention
  - Readability
  - Writability
  - Soft Errors
- Area
- Power

Why is functionality a "metric"?

- Process variations increase with scaling
- Large number of cells requires analysis of tails (out to 6σ or 7σ)
- Within-die V_{TH} variation due to Random Dopant Fluctuations (RDFs)

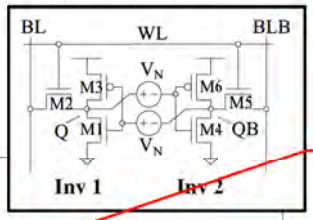
SEBC-L10 MZ

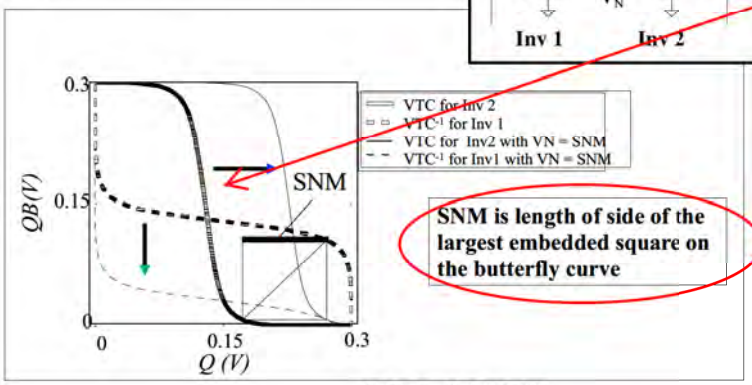
Bisogna definire degli indicatori della memoria che mi permettano di definire se la memoria è robusta o meno (e se devo ancora ottimizzare, o se mi sono spinto troppo in la), una metrica che tenga conto dei parametri tipici di una memoria.
Quelle con cui abbiamo a che fare sono metriche legate alle funzionalità della memoria stessa.

Nel momento in cui passo a tecnologie sempre più scalate, i vari parametri influenzano sempre di più il funzionamento della cella.
Si sta perdendo il concetto informatico di "scrivo zero, leggo zero": soft error, per cui "scrivo zero, leggo tante volte zero, ma qualche volta 1"

Static Noise Margin (SNM)

SNM gives a measure of the cell's stability by quantifying the DC noise required to flip the cell





SNM is length of side of the largest embedded square on the butterfly curve

[Ref: E. Seevinck, JSSC'87]

SEBC-L10 MZ 16

-rappresenta l'ampiezza del generatore di rumore che porta la cella a non cambiare stato-

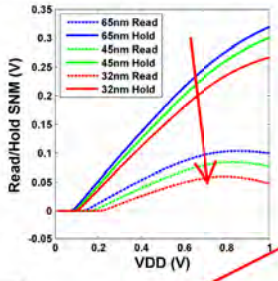
Aggiungere del rumore significa spostare la caratteristica (quella blu viene traslata in orizzontale, quella verde verso il basso).

man mano che questa si sposta, si va a perdere il punto stabile.

Static Noise Margin with Scaling

- Typical cell SNM deteriorates with scaling
- Variations lead to failure from insufficient SNM

Tech and V_{DD} scaling lower SNM

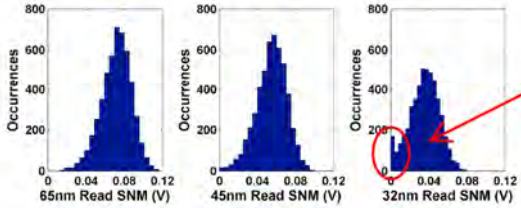


Hold

Read

più la tecnologia diventa piccola, più si riducono i margini.

Variations worsen tail of SNM distribution



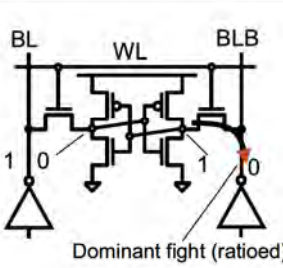
(Results obtained from simulations with Predictive Technology Models – [Ref: PTM; Y. Cao '00])

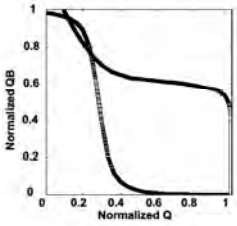
Se non si prendono precauzioni, si rischia di perdere l'informazione in lettura.

Man mano che questa caratteristica peggiora, vanno inserite nella memoria delle strutture di correzione dell'errore.

SEBC-L10 MZ 17

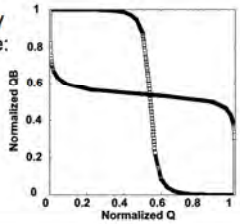
Variability: Write Margin

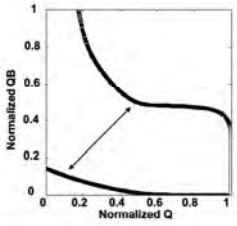




Write failure: Positive SNM

Cell stability prior to write:



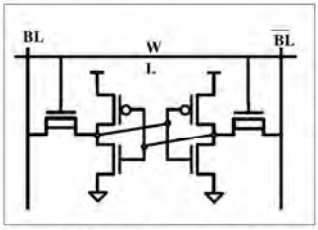


Successful write: Negative "SNM"

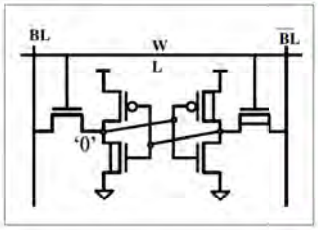
Mod by Giorgio Fissore, pag 239

SEBC-L10 MZ 18

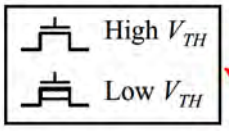
Multiple Threshold Voltages



Dual V_{TH} cells with low V_{TH} access transistors provide good tradeoffs in power and delay



Use high V_{TH} devices to lower leakage for stored '0', which is much more common than a stored '1'



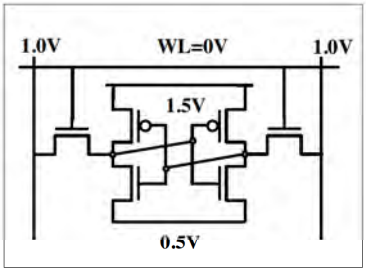
SEBC-L10 [Ref: N. Azizi, TVLSI'03]

Se posso avere nella mia cella transistor con due soglie diverse, in parte ad alta soglia, ed in parte a bassa soglia, posso giocare meglio con i parametri di velocità e consumo:
-la velocità vorrebbe bassa soglia
-il consumo vorrebbe alta soglia.
Esempio (effettivamente utilizzato) voglio privilegiare le letture:
-transistor interni ad alta soglia: rallentano un po' la scrittura, ma consumano poco (due transistor su tre perdono poco).
-pass transistor invece a bassa soglia per velocizzare l'operazione di lettura.

E' stata studiala la statistica dei valori immagazzinati nelle memorie:
Se sono dati, il 90% dei bit immagazzinati sono zeri.
Se istruzioni, l'85% sono zeri.
>> Cerchiamo di ottimizzare principalmente gli zeri, in maniera che i transistor sottoposti a delta_V con uno zero immagazzinato, abbiano Vth più alta, gli altri a Vth più bassa

Multiple Voltages

- Selective usage of multiple voltages in cell array
  - e.g. 16 fA/cell at 25°C in 0.13 μm technology



- High V_{TH} to lower sub-V_{TH} leakage
- Raised source, raised V_{DD}, and lower BL reduce gate stress while maintaining SNM

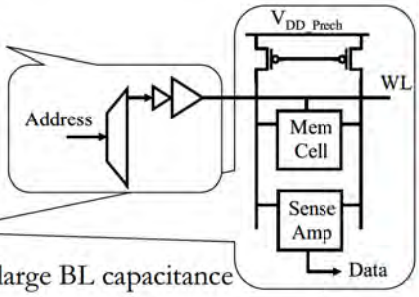
SEBC-L10 [Ref: K. Osada, JSSC'03] MZ 23

>>> ottimizzazione statistica: lavorando sulla statistica, facciamo delle ottimizzazioni che riducono i consumi.

Tanti trucchi u_elettronici che non vediamo, tipo:
La I_leakage dipende (quadraticamente?) dalla delta_V >> se la riduco di un mezzo, riduco ad un quarto la I_leak

Power Breakdown During Read

- Accessing correct cell
  - Decoders, WL drivers
  - For Lower Power:
    - hierarchical WLs
    - pulsed decoders
- Performing read
  - Charge and discharge large BL capacitance
  - For Lower Power :
    - SAs and low BL swing
    - Hierarchical BLs
    - Lower BL precharge
    - Lower V_{DD}
    - May require read assist



SEBC-L10 MZ 24

Consumo
-nella parte di decodifica
-nel pilotare la word line (che ha una capacità enorme rispetto a quella del singolo gate) -scrittura-
-nel pilotaggio della bit line -lettura-

Mod by Giorgio Fissore, pag 241

Reduced Bit Line Voltage Swing

- Can end sense amplifier read operation as soon as differential voltage detection is complete
- Saves fraction of power needed to accomplish read
- ΔV = bit line voltage swing
- V_{core} = core supply voltage
- r = operation fraction that is read
- f = frequency of core operations
- Read power: $\frac{1}{2} C_{eff} V_{core} \Delta V r f$
- Reducing ΔV often fails: increases noise sensitivity, sense amp complexity, reduces RAM performance

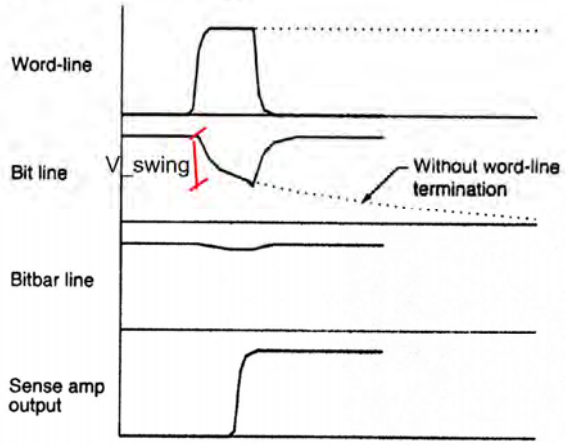
SEBC-L10 MZ 28

Riduzione dei consumi delle bit line:
-ricordiamoci di avere due BL, e che le pilotiamo in fase di lettura.

Se io riduco la dinamica che io accetto sulla bit line (BL Voltage Swing), allora riduco i consumi (ricordare la solita formula a lato).
-Cmq non posso ridurre più di tanto, perchè sennò divento più sensibile al rumore, e perchè il povero sense amplifier fa sempre più fatica. >> esistono altri metodi da applicare

Early Word Line Termination

- Reduces bit line swings

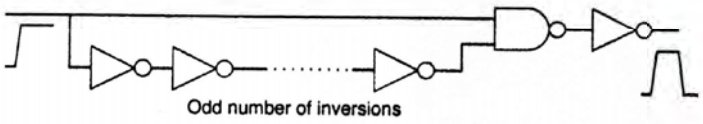


SEBC-L10 MZ 29

Se conosco il tempo necessario a sbilanciare i due bracci (BL), allora posso immaginare di spegnere la WL, in maniera che la BL non si scarichi più da quel momento in poi;
sarò quindi poi in grado di riportarla in su più facilmente >> lo swing non è più di tutta la Vdd
>> devo quindi generare un impulso sulla wl.

Pulsed Word Lines

- Enable word lines only for precise time:
  - Needed to develop bit cell voltage discharge
- Use pulse generator:
  - Gates word line and sense amplifier
  - Need margin for worst-case pulse width:
  - Must estimate actual RAM access time



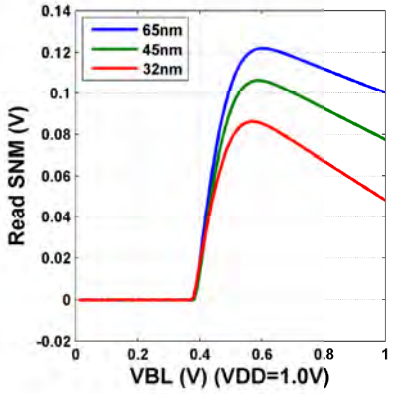
SEBC-L10 MZ 30

Il limite a questa ottimizzazione è dato dal fatto che bisogna progettare gli impulsi in maniera da far lavorare il sistema nella condizione peggiore.
Più la tecnologia diventa piccola, più diventa grande lo spread, e quindi la variabilità dei parametri, più il worst case diventa peggiore, e quindi la tecnica meno efficace.

Mod by Giorgio Fissore, pag 243

Lower Precharge Voltage

- Lower BL precharge voltage decreases power and improves Read SNM
  - Internal bit-cell node rises less
  - Sharp limit due to accidental cell writing if access FET pulls internal '1' low



SEBC-L10 MZ 34

V_{DD} Scaling

- ✓ Lower V_{DD} (and other voltages) via classic voltage scaling
  - ✓ Saves power
  - ✓ Increases delay
  - ✓ Limited by lost margin (read and write)
- ✓ Recover Read SNM with read assist
  - ✓ Lower BL precharge
  - ✓ Boosted cell V_{DD} [Ref: Bhavnagarwala'04, Zhang'06]
  - ✓ Pulsed WL and/or Write-After-Read [Ref: Khellah'06]
  - ✓ Lower WL [Ref: Ohbayashi'06]

SEBC-L10 MZ 35

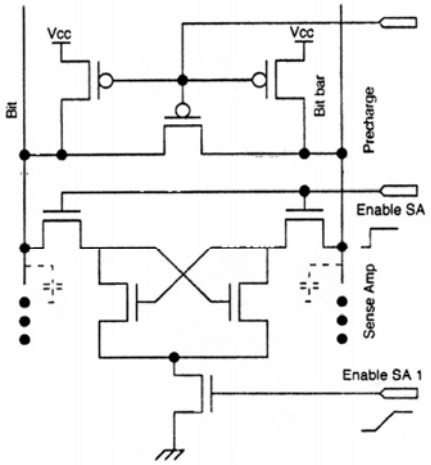
Self-Timed RAM Core

- Different rows have different access speeds
  - Row closest to sense amps is fastest
  - Columns closest to word line drivers enabled first
  - Tailor pulse width to RAM access time
- Use dummy column to time signal flow
  - Forced to known state by shorting one internal node
- Set SR flip-flop to trigger word line
- By time dummy column sense amp generates high:
  - Rest of columns have been sensed
  - Dummy column sense amp resets SR flip-flop, turns off word line
- Dummy column adds insignificant chip area/power overhead
- Called *word line kill* circuit

SEBC-L10 MZ 36

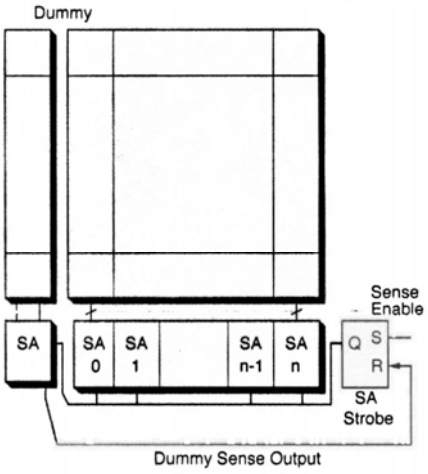
Mod by Giorgio Fissore, pag 245

Differential Charge Sense Amp



SEBC-L10 MZ 40

Self-Timed Sense Amp



SEBC-L10 MZ 41

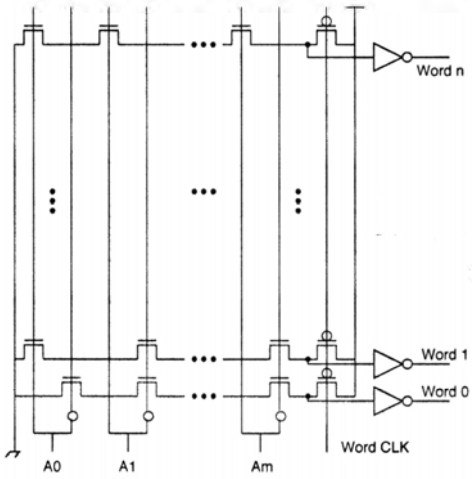
Self-Latching Sense Amp

- Self-latching sense amp automatically limits currents after sense
- Cross-coupled amplifying inverter loop
- Extra transistors transfer bit line voltages to inverter loop

Mod by Giorgio Fissore, pag 247

SEBC-L10 MZ 42

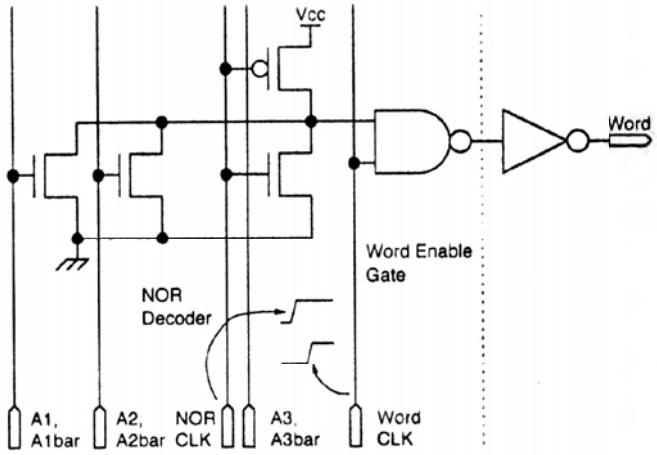
Domino NAND Decoder



SEBC-L10 MZ 46

NAND DECODER: (leggere prima nor decoder sotto)
-Duale alla nor, dove ogni catena è pilotata da tutte le combinazioni dei bit di indirizzo.
Di tutte queste catene, ci sarà un riga con tutti i transistor accesi (quella indirizzata dalla combinazione corretta) che scaricherà la linea; tutte le altre avranno invece almeno un trans spento e quindi non si attiveranno.
Il vantaggio è che questa volta solo un transistor per volta commuta: con una memoria da mille righe, consumerò 1 con una nand, 999 con una nor.
Lo svantaggio è che la scarica dovrà passare per tutta la catena di transistor.

NOR Decoder



SEBC-L10 MZ 47

NOR DECODER
Ho tutte le possibili combinazioni di segnali di ingresso in una struttura a nor, con un transistor che lavora come pull-up per la precarica.
>> in precarica, linea forzata ad uno, in fase di valutazione ogni linea va a zero quando c'è almeno un transistor che conduce >> tutti andranno a zero, tranne uno: quello è la selezione vera e propria.
-Questa struttura è estremamente veloce, in quanto per far scaricare una linea basta un solo transistor a zero: il percorso conduttivo passa per un singolo transistor (soluzione per memorie high speed).
-E' pessima dal punto di vista del low power, poichè tutte le uscite del decoder hanno due commutazioni per colpo di clk

Improve NAND Decoder Speed

- Do not decode A address lines into 1 of 2^A word lines
- Split decoding process
- Decode $A1 < A$ address lines:
  - Use 2^{A1} lines to activate one of second stage decoders
- Second stage decodes $A - A1$ lines into $2^{(A-A1)}$ word lines
  - Get total of: $2^{A1} \times 2^{(A-A1)} = 2^A$ lines
- Recursively repeat to get a tree of intermediate decoders – extreme is to decode 1 address line/stage

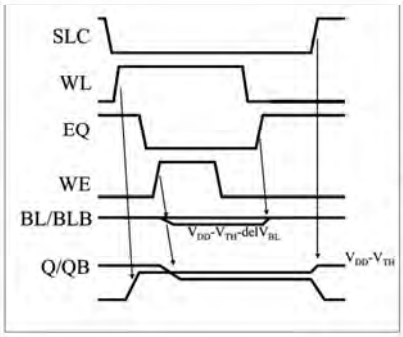
SEBC-L10 MZ 48

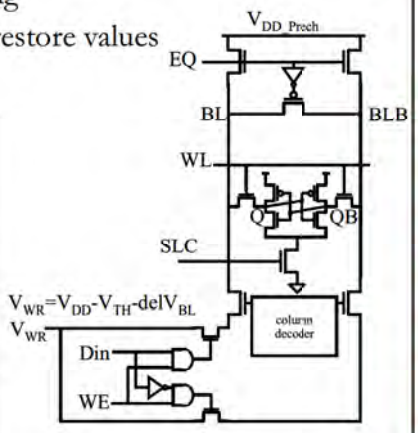
Dato che il ritardo non è lineare con in numero di transistor, splittare il processo di decoding può portare ad un aumento della velocità della nand.
Se ad esempio ho dieci trans, metto due stadi con 5 transistor ciascuno.
Il limite sarebbe avere tanti stadi quanti sono i bit di indirizzo.

Mod by Giorgio Fissore, pag 249

Low-Swing Write

- Drive the BLs with low swing
- Use amplification in cell to restore values





SEBC-L10 [Ref: K. Kanda, JSSC'04] MZ 52

L'altra tecnica per risparmiare è far lavorare le bit line su uno swing ridotto.
Per poter scrivere cmq con questo swing, devo, però poi amplificare la tensione sulle bit_line, in maniera da rendere la scrittura efficace. In questo caso il vantaggio sta nel fatto che riduco lo swing sulla bl, cui sono collegate molte capacità, e lo riporto al livello normale solo localmente, quindi su capacità molto più piccole.

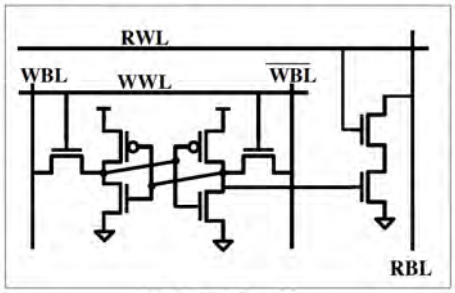
Write Margin

- Fundamental limit to most power-reducing techniques
- Recover write margin with write assist, e.g.
  - Boosted WL
  - Collapsed cell V_{DD} [Itoh'96, Bhavnagarwala'04]
  - Raised cell V_{SS} [Yamaoka'04, Kanda'04]
  - Cell with amplification [Kanda '04]

SEBC-L10 MZ 53

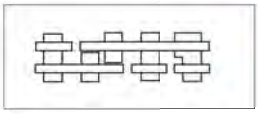
Non-traditional cells

- Key tradeoff is with functional robustness
- Use alternative cell to improve robustness, then trade off for power savings
- e.g. Remove read SNM



8T SRAM cell

- Register file cell
- 1R, 1W port
- Read SNM eliminated
- Allows lower V_{DD}
- 30% area overhead
- Robust layout



SEBC-L10 [Ref: L. Chang, VLSI'05] MZ 54

Quindi la prima cosa da fare per avere memorie a basso consumo, è separare lettura e scrittura, tramite due linee aggiuntive (RWL - read word line- e WWL -write word line)
-Non ho rumore nell'operazione di lettura, quindi si hanno static noise margin molto migliori >> minore tensione di lavoro (o maggiore velocità)
-Pago che ho 8 celle al posto che 6 ed una linea in più (30% di area aggiuntiva).

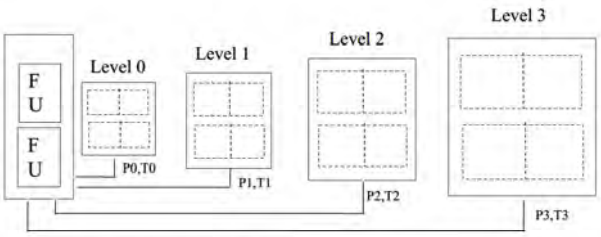
Mod by Giorgio Fissore, pag 251

Design of Power-Efficient Memory Subsystems

- Memory accesses are slow and consume more power with increasing memory size
  - reduce memory storage requirements of the applications
  - during system conceptualization use principle of temporal locality to reduce memory storage requirement
  - improve locality and reduce need for temporary storage of results of computation by consuming them ASAP
  - Reduce memory need by data compression
- Advanced hierarchical memory architectures for low power

SEBC-L10 MZ 58

Hierarchical Memory Models



- Power and access time increases as we move up memory hierarchy
- Exploit non uniformities in access frequencies of data
  - Place frequently accessed locations in low hierarchies to minimize average cost per access

SEBC-L10 MZ 59

Caches: Architectural support

- • Circuit-level technique must be controlled at the architecture level
  - – Data stored in sleeping cell is unreliable or lost
  - – Maximize number of sleep-mode lines while preserving performance
- • Caches tradeoff efficiency for robustness
- • Deactivate (put into sleep mode) unused cache lines

SEBC-L10 MZ 60

Cache: strutture in cui memorizzo blocchi di informazione che immagino di riutilizzare in tempi brevi.
-Un dato viene sostituito quando un set è pieno, e devo inserirne uno nuovo.
-L'approccio low power (dove posso accettare una riduzione di prestazioni), è basato sul fatto che l'informazione prima o poi vada scartata: metto i dati più vecchi in sleep mode.
-Risparmio poichè quei dati non consumano più
-Pago che se poi mi servono, non li ho più.

Mod by Giorgio Fissore, pag 253

Lecture 11
Design & Leakages of Low-Voltage CMOS Devices

- Circuit design style
- Leakage current in deep submicron transistors
- Leakage current Optimization
- Summary

SEBC-L11 MZ 1

Appunti di Giorgio Fissore
Disponibili in centro stampa

Queste considerazioni sono più a livello microelettronico.

Introduction

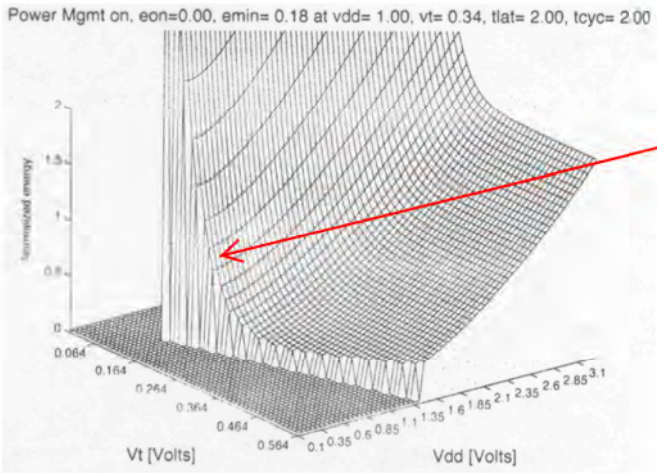
- Lower MOS supply voltages save energy
  - Require low transistor threshold voltages
  - Causes sub-threshold leakage currents to be significant
- Need to control leakage currents with device design
- Need to run slower parts of circuit at lower voltage
- Circuit design style (type of CMOS logic) is critical
- High leakages require modified I_{DDQ} testing
  - Modify to account for operating frequency

SEBC-L11 MZ

Come sempre, riducendo Vdd, si riduce quadraticamente il consumo dei mos, ma si va ad aumentare il ritardo delle porte

Si riduce allora la Vth dei dispositivi per abbassare il ritardo, ma questo porta all'aumento della corrente di leakage, che oltre certi livelli diventa pericolosa.
Come al solito bisogna trovare il compromesso migliore per dimensionare la I_{LEAK}, andando anche eventualmente a cambiare il modo di progettare l'architettura.

Power Versus Supply and V_t

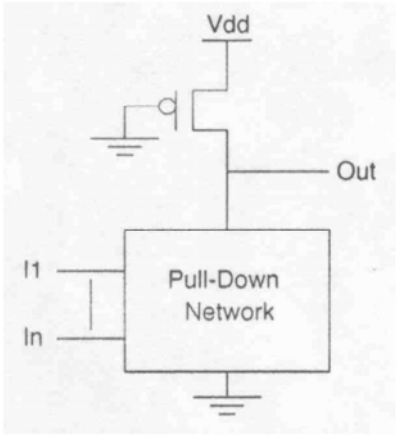


In questa zona, ridurre di poco la Vth, porta i consumi ad aumentare tantissimo. (più a dx invece le variazioni sono molto lievi)

SEBC-L11 MZ 3

Mod by Giorgio Fissore, pag 255

Pseudo-nMOS Logic



SEBC-L11 MZ 7

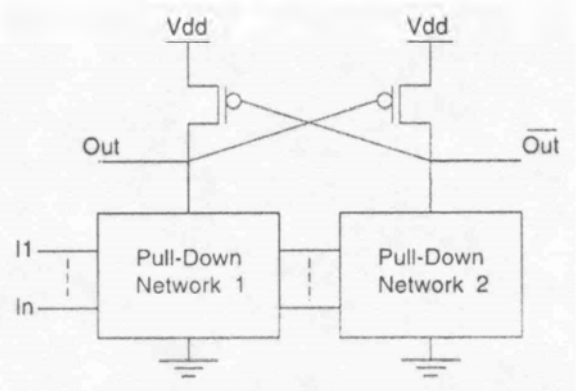
Pseudo-nMOS Logic

- Less complex than static CMOS, so lower C_L, faster
- Unlike nMOS, load device unaffected by body effect
- Ratioed design style good for large fanin NAND/ NOR
- Uses more power than static CMOS
  - Power always flows when pull-down network on

Il vantaggio di questa logica è che devo pilotare la metà dei transistor!! (capacità da pilotare dimezzate rispetto a static-cmos)
-Il problema è che ho un consumo statico quando forzo uno zero in uscita.

SEBC-L11 MZ 8

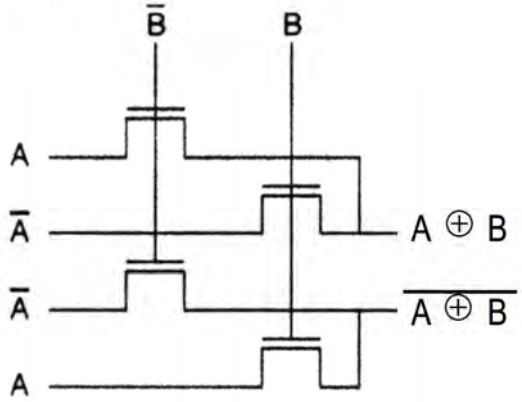
Differential Cascode Voltage Switched Logic (DCVS)



Soluzioni alternative:
-cerco di sfruttare il fatto di avere solo reti di pd, ma eliminando il consumo statico.
Qui ho due reti di pd complementate, ognuna con il suo transistor di pu.
Fanno si che una volta che, finita la commutazione, l'uscita della rete negata interdice il transistor di pu, eliminando il consumo statico.
-riduco l'area poichè sostituisco una rete di pu con una di pd (più piccola)
>> candidata come logica per il basso consumo.

SEBC-L11 MZ 9

Complementary Pass Transistor Logic (CPL) – XOR/XNOR



SEBC-L11 MZ 13

Complementary Pass-Transistor Logic (CPL)

- Advantages:
  - Differential output signals
  - Modular
  - Reduced internal C_L – low power
  - Can support reduced voltage swing
- 32-bit Adder: CPL has 10% lower power-delay product than static CMOS

>> scegliere uno stile di progetto che unisca il ritardo necessario con il minor consumo possibile di potenza.

SEBC-L11 MZ 14

Clocked Logic

Altro stile di logica, è quella legata al clock.

- Domino Logic
- *Differential Current Switch Logic* (DCSL)
  - Uses less power than Domino logic
- Higher performance at expense of higher power dissipation

Mod by Giorgio Fissore, pag 259

SEBC-L11 MZ 15

Differential Current Switch Logic

- DCVS logic gate modified to reduce internal node voltage swings
- T2, T3, T6, T7 – cross-coupled inverter pair
- T1 & T4 – Precharge outputs high
- Note: T12 sometimes needed to prevent internal node charge buildup
- Advantage: Once evaluation completed, high output is disconnected from n-tree, so further input changes ignored
- No static paths from V_{CC} to ground at end of evaluation
- Can get *completion signal* by NANDing two outputs
- Internal node swings for DCSL (DCSL2) are 1 V (0.3 V)

SEBC-L11 MZ 19

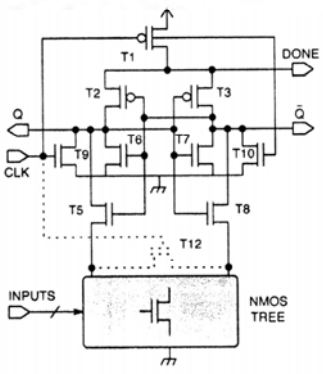
DCSL Gate Operation

- When CLK low, Q and $\overline{Q}$ precharged high
- When CLK high, nMOS tree inputs must be stable
  - T9, T10, T11 switched on by high CLK
  - Precharged outputs switch on T5, T6, T7, T8
  - Q and $\overline{Q}$ discharge asymmetrically towards ground
  - One of the paths to ground is stronger (say Q)
  - So, Q falls faster than $\overline{Q}$
  - Cross-coupled inverter functions as sense amplifier
  - Boosts output voltage differential in the right direction, so $\overline{Q}$ swings high
- Logic limits charge-up of internal nodes of nMOS tree to voltages much smaller than $V_{CC} - V_{tn}$ (limit for DCVS Logic)

SEBC-L11 MZ 20

DCSL2 Gate (Precharged Low)

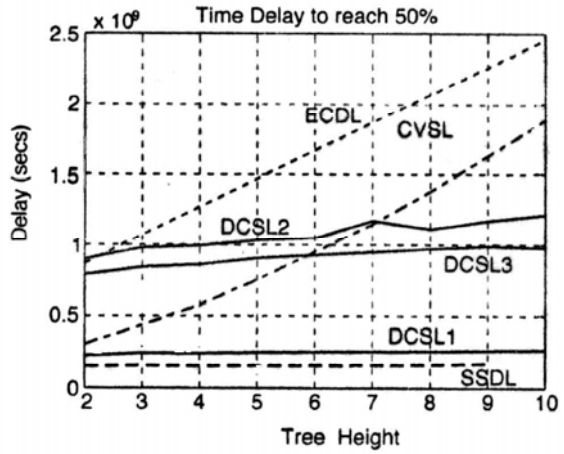
- CLK = high is precharge, CLK = low is evaluation
- Degraded gate propagation delay



SEBC-L11 MZ 21

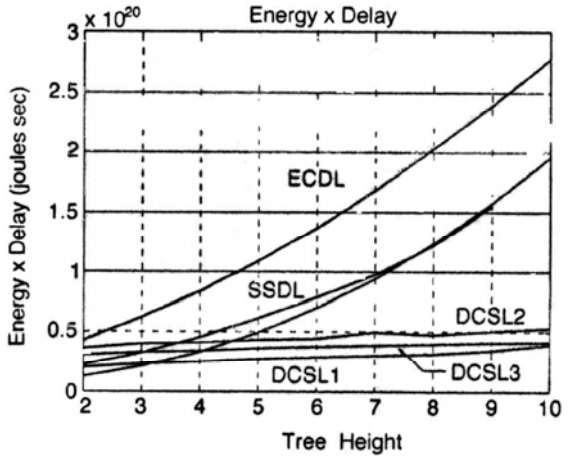
Mod by Giorgio Fissore, pag 261

CLK-to-Q Delay 50%



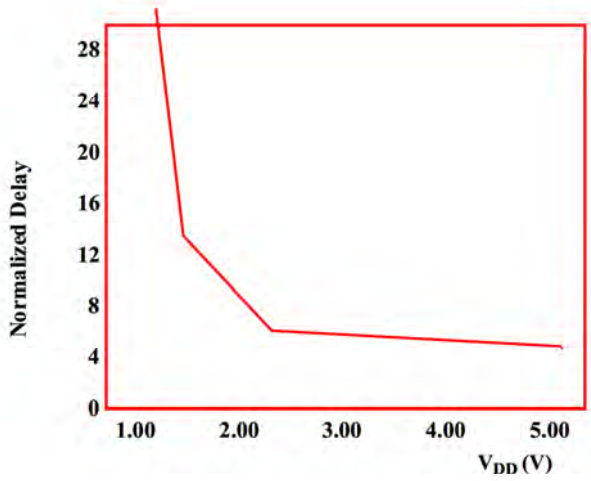
SEBC-L11 MZ 25

Energy X Delay vs. n-Tree Height



SEBC-L11 MZ 26

Delay as a function of V_{DD}



Mod by Giorgio Fissore, pag 263

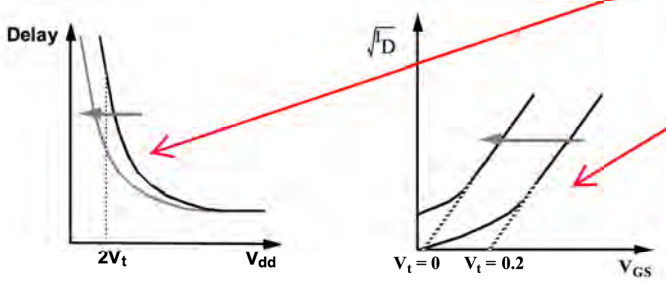
SEBC-L11 MZ 27

Threshold Voltage Scaling

- Reducing the threshold voltage allows the supply voltage to be scaled down to lower Psw without loss in speed.
- Example:
  - Circuit A: Vdd = 1.5 V , Vth = 1 V
  - Circuit B: Vdd = 0.9 V , Vth = 0.5 V
- Circuit A and B have approximately the same delay

SEBC-L11 MZ 31

Lowering the Threshold



Reduces the Speed Loss, But Increases Leakage

Interesting Design Approach:
DESIGN FOR $P_{Leakage} == P_{Dynamic}$

SEBC-L11 MZ 32

Alla riduzione della Vth, vi è uno spostamento delle curve verso sx (diminuzione del ritardo).

Il problema è che, però, anche la curva relativa alle correnti di drain in funzione di Vgs si sposta verso sx >> questo non è affatto positivo, poichè aumenta la conduzione sotto-soglia (in maniera molto più che lineare).

>> si riduce Pdinamica
>> si aumenta Pleak
>>
A noi interessa solo la somma delle due, quindi dobbiamo trovare il minimo.

Limits to Threshold Voltage Scaling

- If threshold voltage scaling is required, low-threshold MOS devices must be used for the design.
- The limit on threshold voltage scaling is imposed by:
  - ✓ The noise margin
  - ✓ The increase of sub-threshold current
- Trade-off between Psw (that decreases as Vth decreases) and $P_{leakage}$ (that increases as Vth decreases)

SEBC-L11 MZ 33

Abbiamo anche altri limiti, dati dal fatto che quando si lavora con tensioni più basse, si abbassano i margini di rumore.

Mod by Giorgio Fissore, pag 265

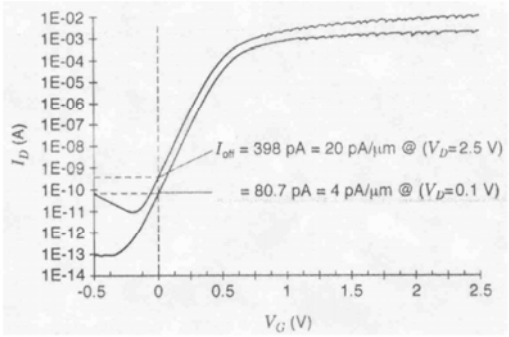
pn Reverse-Bias Current

- Minority-carrier drift near edge of depletion region
- Electron-hole pair generation in depletion region (reverse-biased junction)
- If heavily doped, Zener tunneling may also happen
- MOSFET added leakage:
  - Between drain and well junction (due to overlap of gate to drain to well pn junctions)
  - Carrier generation in drain-to-well depletion regions
- *pn* reverse-bias leakage = f (junction area, doping)

SEBC-L11 MZ 37

Weak Inversion in 3.5 μm Technology – Dominates Leakage

- Happens in nFET when gate is below V_t
- Carriers diffuse through channel (no horizontal E)



SEBC-L11 MZ 38

CMOS Technologies

Technology (μm)	V_{DD} (V)	T_{ox} (Å)	V_T (V)	L_{eff} (μm)	I_{off} (pA/μm)
1.0	5	200	n/a	0.80	4.1×10^{-4}
0.8	5	150	0.60	0.55	5.8×10^{-2}
0.6	3.3	80	0.58	0.35	0.15
0.35	2.5	60	0.47	0.25	8.9
0.25	1.8	45	0.43	0.15	24
0.18	1.6	30	0.40	0.10	86

Mod by Giorgio Fissore, pag 267

SEBC-L11 MZ 39

Gate-Induced Drain Leakage

- Generates carriers into substrate and drain from surface traps or band-to-band tunneling
  - Due to high electric field under gate/drain overlap region that causes deep depletion
  - Happens at low V_G and high V_D bias
  - Localized along channel width between gate and drain
- Appears as hook in last figure:
  - Increasing current for negative V_G values
- Major problem in I_{off} current:
  - Caused by thinner t_{ox}, higher V_{DD}, and lightly doped drains

SEBC-L11 MZ 43

Punchthrough

- Happens when drain and source depletion regions approach each other and touch
- Lets channel current exist deep in sub-gate region
  - Gate loses control of sub-gate region
- Varies quadratically with V_D and with S_t
- Viewed as subsurface version of DIBL

SEBC-L11 MZ 44

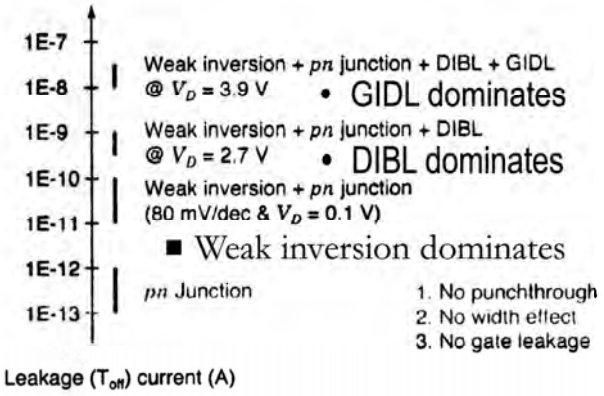
Narrow-Width Effect

- Trench isolation:
  - Dig trench in substrate and fill with SiO_2 to isolate *n* and *p* MOSFETs
- Non-trench isolated technologies:
  - V_t increases for gate widths of 0.5 μm
- Trench isolated technologies:
  - V_t decreases for effective channel widths $W \leq 0.5$ μm

SEBC-L11 MZ 45

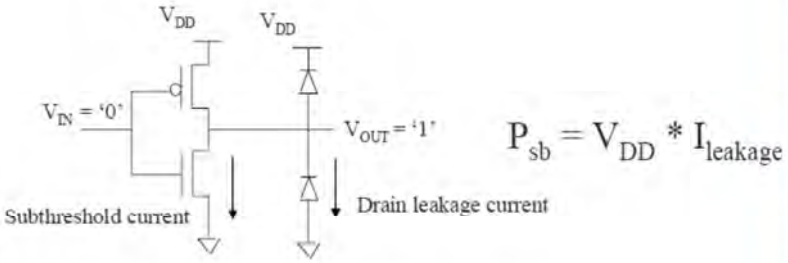
Mod by Giorgio Fissore, pag 269

Leakage Summary



SEBC-L11 MZ 49

Leakege power increase



Subthreshold current is dominant
- Increase exponentially with temperature;
- Increase exponentially with technology scaling

SEBC-L11 MZ 50

Essendo i_leak principalmente dovuto alla corrente inversa, il suo consumo aumenta esponenzialmente con la temperatura.
Quindi non è più solo un problema di progetto per far funzionare dal pt di vista logica, ma anche di struttura.
Se metto un dissipatore, che riesca a ridurre la temperatura, allora consumo meno.
Se scalo la tecnologia (aumento della densità di potenza), allora inevitabilmente la corrente di leak aumenta esponenzialmente.

Leakage Current Estimation

- Ignore diode junction leakage
- Subthreshold leakage increases exponentially with reduction of V_t
- Necessary transistor model elements:
  - Sub-zero V_{GS} for nFET
  - Super-zero V_{GS} for pFET
  - Body effect
  - DIBL

SEBC-L11 MZ 51

Siccome ci sono una ventina di fattori che determinano I_leak, serve un modello che tenga conto di tutti quanti (funzione di Vds, di caratteristiche geometriche del disp, di Vgs,..)

Mod by Giorgio Fissore, pag 271

Considering Leakage @ Design Time

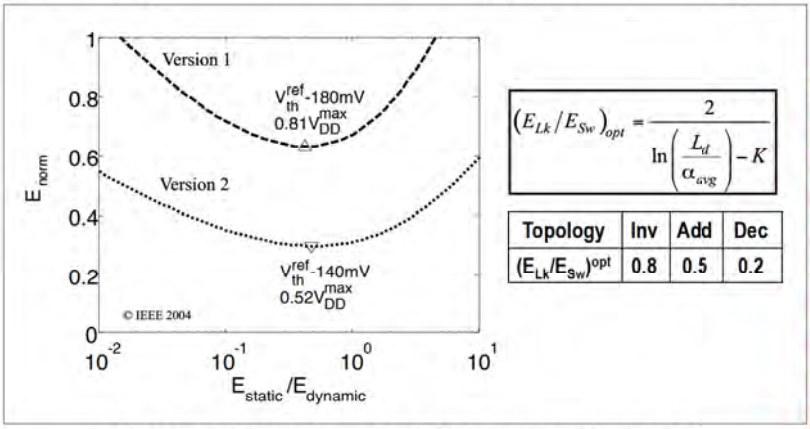
- ✓ Considering leakage as well as dynamic power is essential in sub-100 nm technologies
- ✓ Leakage is not essentially a bad thing
  - ✓ Increased leakage leads to improved performance, allowing for lower supply voltages
  - ✓ Again a trade-off issue …

Vediamo ora come convivere al meglio con il leakage, considerando questo problema durante il progetto. (va considerato tutte le volte che si lavora sotto i 100 nm).

Il leakage, è una cosa che noi abbiamo messo per poter abbassare Vdd >> quindi non dovremo tipicamente portarla a zero, dato che ci permette di abbassare la P_dinamica.

SEBC-L11 MZ 55

Leakage – Not Necessarily a Bad Thing



$$\left(E_{Lk}/E_{Sw}\right)_{opt} = \frac{2}{\ln\left(\frac{L_d}{\alpha_{avg}}\right) - K}$$

Topology	Inv	Add	Dec
$(E_{Lk}/E_{Sw})^{opt}$	0.8	0.5	0.2

Optimal designs have high leakage ($E_{Lk}/E_{Sw} \approx 0.5$)

Must adapt to process and activity variations

[Ref: D. Markovic, JSSC'04]

Si trovano dei minimi, a seconda dei circuiti, quando si hanno consumi statici e dinamici non sbilanciati, ma quando c'è tra questi un rapporto di circa 0.5.
(poi dipende molto dalla funzione logica realizzata, e da quanti alberi nel circuito sono spenti).
>> Il circuito ottimo non porta il leak a zero, ma a circa la metà del consumo della potenza dinamica

SEBC-L11 MZ 56

Reducing Leakage @ Design Time

- ✓ Using longer transistors
  - ✓ Limited benefit
  - ✓ Increase in active current
- ✓ Using higher thresholds
  - ✓ Channel doping
  - ✓ Stacked devices
  - ✓ Body biasing
- ✓ Reducing the voltage!!

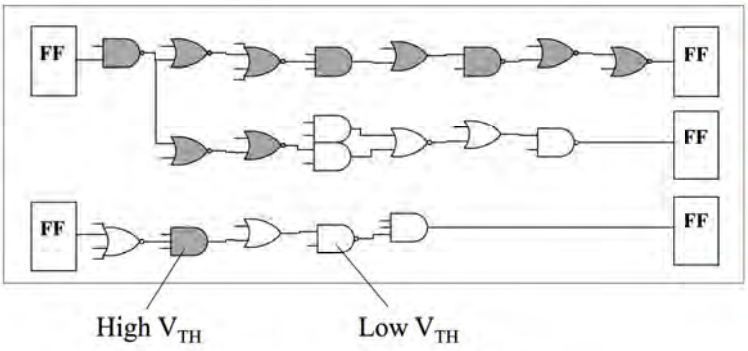
Come faccio a ridurre il leak? (tre tecniche)
-usare transistor più lunghi
-usare soglie più alte
-stacking

Mod by Giorgio Fissore, pag 273

SEBC-L11 MZ 57

Using Multiple Thresholds

- Cell-by-cell V_{TH} assignment (not at block level)
- Achieves all-low-V_{TH} performance with substantial leakage reduction in leakage



[Ref. S. Date, SLPE'94]

SEBC-L11 MZ 61

Tra l'altro, lavorare con due soglie diverse, non comporta nessun problema a livello di compatibilità logica.

Gate grigi: alta soglia

Nell'esempio a lato, il percorso sopra non è critico >> alta soglia
>> Facendo questo, ottengo circa le stesse prestazioni, ma con un leakage ridotto.

Multiple Thresholds Based on Path Criticality

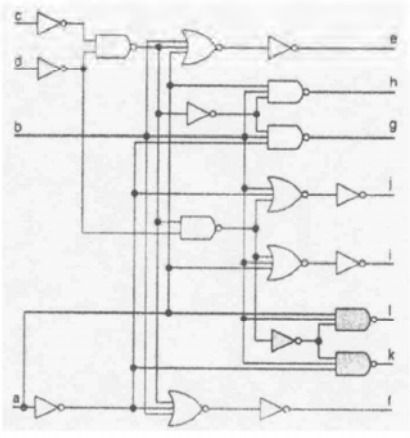
- Use high V_{th} devices on non-critical paths
- Use low V_{th} ones on critical paths
- Selection of threshold level varies with absolute value of V_{th}
- Threshold assignment done automatically by algorithm

SEBC-L11 MZ 62

Queste ottimizzazioni sono fatte dall'algoritmo CAD del sintetizzatore.

Original Circuit

- Shaded nodes are on critical paths



SEBC-L11 MZ 63

Vado ad identificare con l'algoritmo i nodi critici o meno.
E' meglio partire a modificare da tutti gate ad alta soglia, e trasformare poi quelli che non ci stanno nel timing, o tutti a bassa soglia e alzare quelli che hanno più tempo?
Vedremo più avanti

Mod by Giorgio Fissore, pag 275

Multiple Thresholds and Design Methodology

- ✓ Easily introduced in standard cell design methodology by extending cell libraries with cells with different thresholds
  - ✓ Selection of cells during technology mapping
  - ✓ No impact on dynamic power
  - ✓ No interface issues (as was the case with multiple V_{DD}'s)
- ✓ Impact: Can reduce leakage power substantially

SEBC-L11 MZ 67

Dual-V_{TH} Design for High-Performance Design

	High-V_{TH} Only	Low-V_{TH} Only	Dual V_{TH}
Total Slack	-53 psec	0 psec	0 psec
Dynamic Power	3.2 mW	3.3 mW	3.2 mW
Static Power	914 nW	3873 nW	1519 nW

All designs synthesized automatically using Synopsys Flows

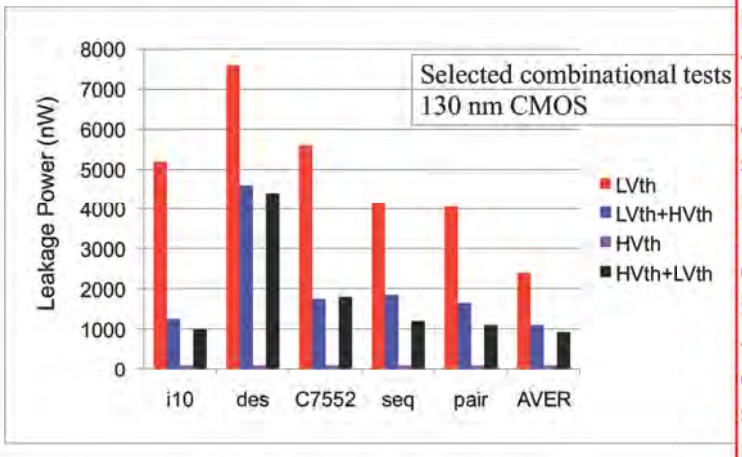
SEBC-L11 [Courtesy: Synopsys, Toshiba, 2004]

Se uso solo H-Vth, ho uno slack di -53 ps >> troppo lento, non sto nel timing.

La potenza dinamica è circa uguale.

La potenza statica, si più che dimezza!

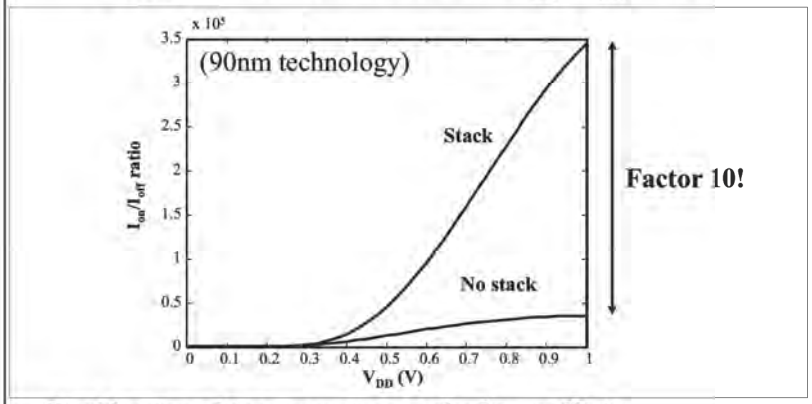
Example: High- vs. Low-Threshold Libraries



SEBC-L11 [Courtesy: Synopsys 2004]

4 Soluzioni:
-Rosso: solo bassa soglia (sempre quella che consuma di più)
-Viola: solo alta soglia (sempre quella che consuma di meno), ma non utilizzabile xk troppo lento.
>> ho due intermedi in cui ho sia alta soglia, che bassa, ma secondo uno dei due criteri di prima.
-Blu: il cad parte da tutte celle definite a bassa soglia, dopo di chè sostituisce nei percorsi non critici le cella a bassa soglia con quelle ad alta soglia. (riduzione di anche più del 50%, es i10, riduzione di un fattore 5)
Il risparmio dipende dalla distribuzione dei ritardi combinatori: meno percorsi critici ci sono, più ho benefici.
-Nera: parti da tutto ad alta soglia, poi sostituisco dove non riesco a starci. Si scopre che le soluzioni in nero portano quasi sempre a risultati migliori (gli algoritmi infatti non trovano sempre il minimo assoluto, e in questo caso lavorano meglio)

Complex Gates Increase I_{on}/I_{off} Ratio



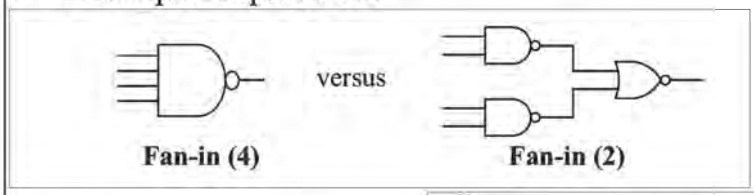
Stacking transistors suppresses submicron effects

- Reduced velocity saturation
- Reduced DIBL effect
- Allows for operation at lower thresholds

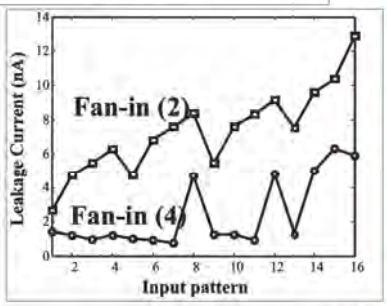
SEBC-L11 MZ 73

Complex Gates Increase I_{on}/I_{off} Ratio

- **Example: 4-input NAND**



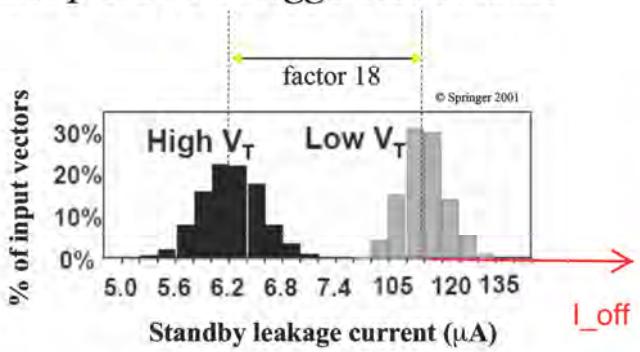
With transistors sized for similar performance:
Leakage of Fan-in(2) = Leakage of Fan-in(4) x 3
(Averaged over all possible input patterns)
(Complex gates come with fewer leakage paths)



SEBC-L11 MZ 74

Fino ad un po' di tempo fa, i circuiti complessi andavano sempre realizzati spezzettandoli in unità il più semplici possibile, essendo le porte elementari molto più veloci.
Però con porte più complesse, ci sono più transistor in serie, e quindi viene ridotta molto I_leak.
-Notare che la I_leak non è un valore costante, ma dipende dalle combinazioni degli ingressi, poichè questi possono attivare percorsi diversi (in caso di stand-by ad esempio, si può risparmiando impostando gli ingressi che fanno consumare di meno).

Example: 32 bit Kogge-Stone Adder



Reducing the threshold by 150 mV increases leakage of single NMOS transistor by factor 60

A complex structures is much less sensitive to Vth increase than a single Transistor (18 vs. 60)

[Ref: S.Narendra, ISLPED'01]

SEBC-L11 MZ 75

Ridurre la soglia di 150 mV porta a consumi di leakage 60 volte maggiori.
Se si usano invece configurazioni più complesse, l'incremento dei consumi di leak è solo più di 18.
>> più un circuito è complesso, meno è sensibile alla riduzione della tensione di soglia.
>> questo ha anche un altro vantaggio: il circuito sarà infatti anche meno sensibile rispetto alle variazioni del parametro "Vth" che avvengono in fabbricazione.

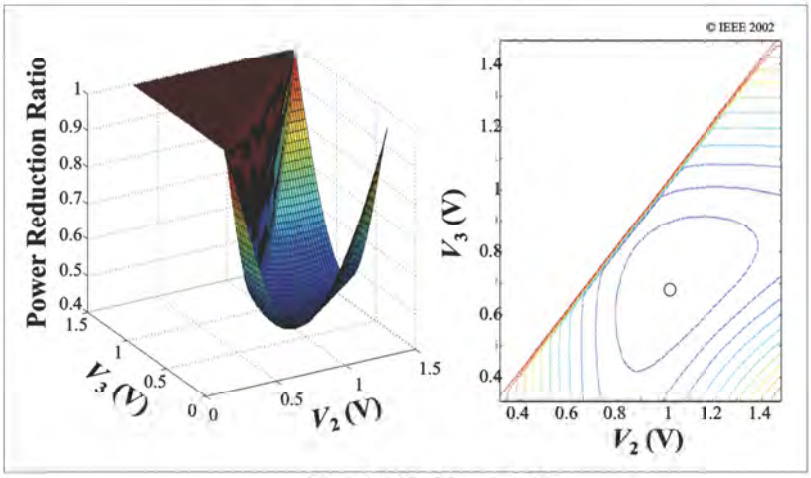
Multiple Supply Voltages

- ✓ **Block-level supply assignment**
  - ✓ Higher throughput/lower latency functions are implemented in higher V_{DD}
  - ✓ Slower functions are implemented with lower V_{DD}
  - ✓ This leads to so-called "voltage islands" with separate supply grids
  - ✓ Level conversion performed at block boundaries
- ✓ **Multiple supplies inside a block**
  - ✓ Non-critical paths moved to lower supply voltage
  - ✓ Level conversion within the block
  - ✓ Physical design challenging

SEBC-L11 MZ 79

Mentre lavorare con Vth differenti è quasi indolore, avere più Vdd ha un costo per il circuito.

Using Three V_{DD}'s



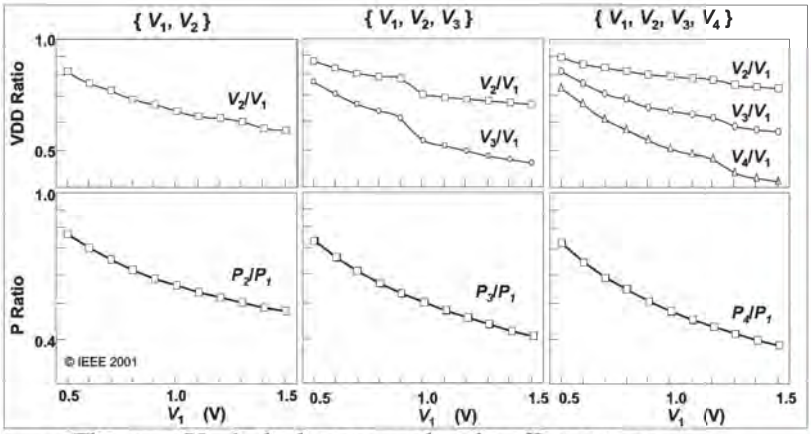
$V_1 = 1.5V,\ V_{TH} = 0.3V$

[Ref: T. Kuroda, ICCAD'02]

SEBC-L11 MZ 80

Ci chiediamo se qui avere più di due Vdd sia influente.
>> dipende dai casi, ma spesso basta

Optimum Number of V_{DD}'s



- The more V_{DD}'s the less power, but the effect saturates
- Power reduction effect decreases with scaling of V_{DD}
- Optimum V_2/V_1 is around 0.7

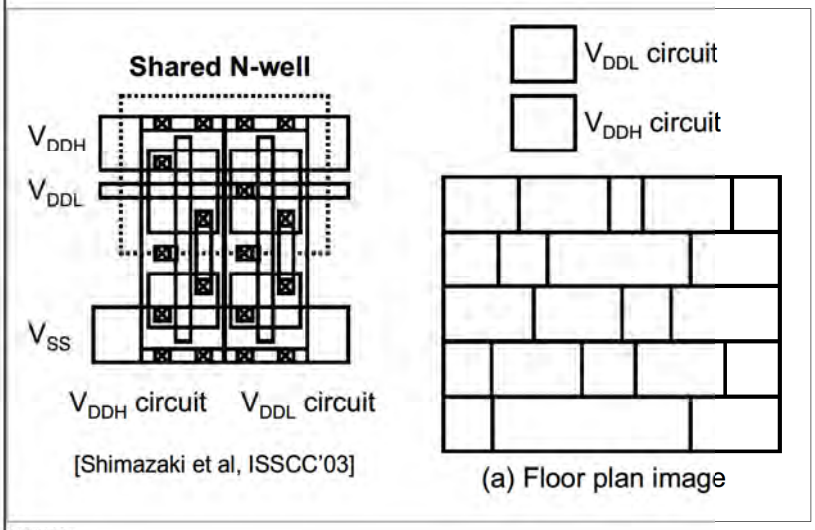
[Ref: M. Hamada, CICC'01]

SEBC-L11 MZ 81

-Proviamo a metterne due, e cerchiamo il rapporto ottimale tra le due Vdd, che provoca il minor consumo.
-Se vado su tre, ottengo un risparmio che potrebbe essere maggiore, ma si vede come c'è comunque un fenomeno di saturazione.
>> Più la Vdd scende, meno è utile avere più alimentazioni.
>> Il rapporto ottimale tra le due Vdd è generalmente 0.7

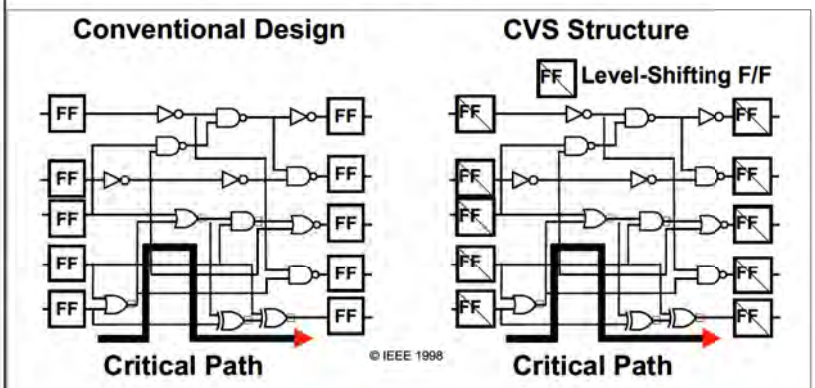
Mod by Giorgio Fissore, pag 281

Shared N-Well



SEBC-L11 MZ 85

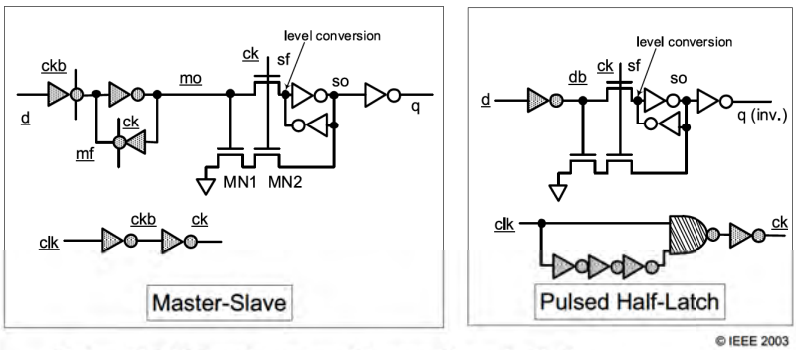
Example: Multiple Supplies in a Block



Lower V_{DD} portion is shared
"Clustered voltage scaling"

SEBC-L11 [Ref: M. Takahashi, ISSCC'98] MZ 86

Level Converting Flip-Flops (LCFFs)



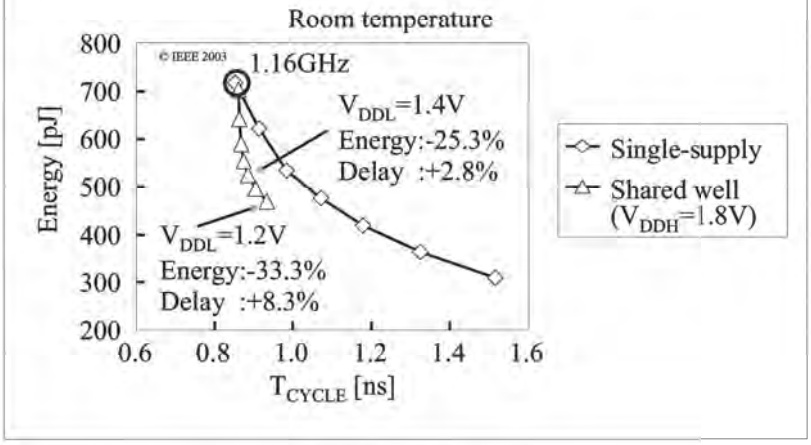
Pulsed Half-Latch versus Master-Slave LCFFs

- Smaller # of MOSFETs / clock loading
- Faster level conversion using half-latch structure
- Shorter D-Q path from pulsed circuit

SEBC-L11 [Ref: F. Ishihara, ISLPED'03] MZ 87

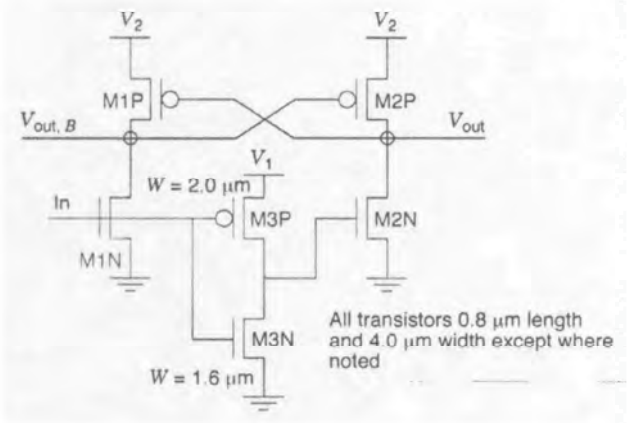
Mod by Giorgio Fissore, pag 283

Measured Results: Energy and Delay

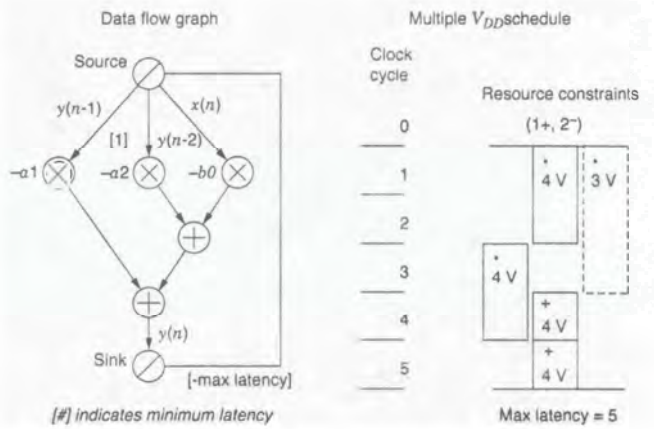


[Ref: Y. Shimazaki, ISSCC'03]

DCVS Voltage Level Converter



Scheduled Data Flow Graph with Multiple Supplies



Mod by Giorgio Fissore, pag 285

Summary (2)

- A few techniques proposed for leakage power optimization during design:
  - Synthesize and map the design onto high Vth cells. Minimum leakage implementation.
  - Replace high – Vth cells on the critical path with low- Vth cells to meet timing constraints.
- Leakage power increase required to meet timing constraints may vary from 20% to 200%.

Mod by Giorgio Fissore, pag 287

FPGA – Static Power

- Historically, low power designs and CPLD devices have been mutually exclusive.
- Early PLDs, implemented on the bipolar processes, consumed hundreds of mA during quiescent operation.
- Migration to the CMOS process reduced power consumption, but most chip designers fabricate product term word lines using a "wired NOR" approach which requires a sense amplifier to improve the propagation speed.
- This approach requires a large amount of standby current due to the linear operation of the sense amplifier.
- To overcome large quiescent current consumption, a technique employing a chain structure of CMOS gates was created.

SEBC-L12 MZ 4

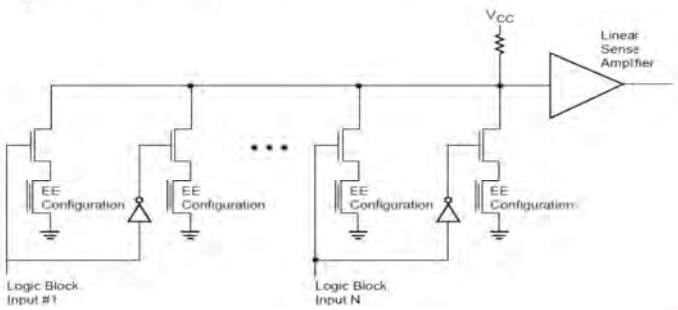
FPGA – Sense Amplifier

- In existing CPLD architectures, the circuits that propagate logic-level transitions in the product term array are derived from the original bipolar PLD design technology.
- Product term word lines have connections for each input into the logic block (and its complement) and thus have a large capacitive load.
- Switching on this line would be slow, therefore sense amplifiers are used at the end of each product term word line in the product-term array to achieve fast propagation delays.
- Because CPLD product terms cannot be decoded (as with memory locations in an EPROM) there must be a sense amplifier for each individual product-term.
- These sense amplifiers operate continuously, drawing supply current even when not switching.

SEBC-L12 MZ 5

FPGA – Sense Amplifier

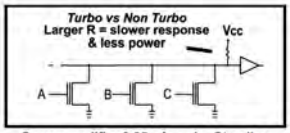
- Logic block inputs are connected to the product term lines.
- The sense amplifiers operate in the linear region, and ensure fast propagation times by amplifying small changes on the product term line, such that it represents a full voltage swing.
- The sense amplifier can recognize a product term voltage change as little as 100 mV.



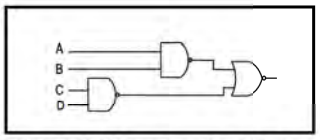
SEBC-L12 MZ 6

La struttura open collector mal si presta a lavorare ad alta frequenza, poichè per far passare da zero ad uno l'uscita bisogna caricare tutte le capacità parassite sulla bit line programmabile, con tempi di propagazione troppo lunghi.
>>> viene inserito quindi a valle un sense amplifier per velocizzare questo processo.
>>> il problema è che ognuna di queste linee deve avere il suo sense amp. >> 400 linee (quindi 400 minterm, nemmeno tanti) richiedono 400 amp
>>> consumano tantissimo!!!
>>> nelle memorie ci sono anche i sense amp., ma sono meno problematici poichè li vengono accesi solo quando servono; nell'fpga invece realizzano funzioni combinatorie e quindi vanno tenuti sempre accesi

RealDigital Design Advantage (Xilinx)



Sense amplifier 0.25mA each - Standby
Higher I_{CC} at Fmax



RealDigital : CMOS Everywhere - Zero Static Power

- Traditional CPLDs - bipolar sense amp product terms
  - Always consumes power
    - Even at standby
  - Performance is traded for power consumption as devices get larger
- CoolRunner-II RealDigital design uses 100% CMOS for product terms
  - Virtually no standby current
  - Combines high performance & ultra low power
  - No power limits on device size

SEBC-L12 MZ 10

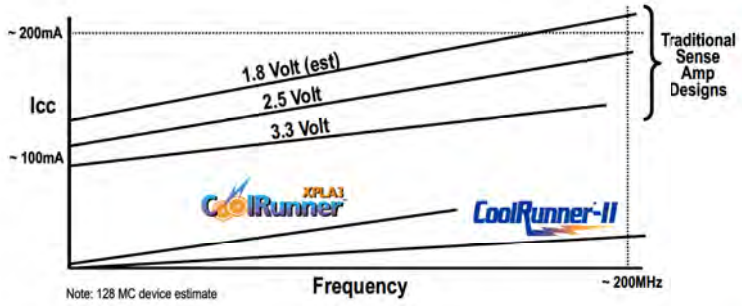
Reducing Static Power

- The switching current behaves in a manner similar to that of combinational logic in a gate array. The static current for each gate is small—about 1 fA.
- The total instantaneous dynamic current is also low, with only the gates in one path of the tree switching at a time.
- The gates in each path of the tree also switch in succession, rather than all at once, thus reducing peak dynamic current.
- Total standby current for Xilinx CoolRunner CPLDs is under 100 microamps (µA)—1000 times less than that exhibited by CPLDs that use the sense amplifier approach.

SEBC-L12 MZ 11

Le correnti di standby di questi dispositivi sono dell'ordine delle centinaia di uA, quindi circa 4 ordini di grandezza in meno rispetto a prima.

Reducing Static Power



SEBC-L12 MZ 12

Prima anche a frequenza zero, c'era un consumo (~100mA-1A)
Ora invece il consumo è legato linearmente alla frequenza, senza offset a parte il leakage.

Mod by Giorgio Fissore, pag 291

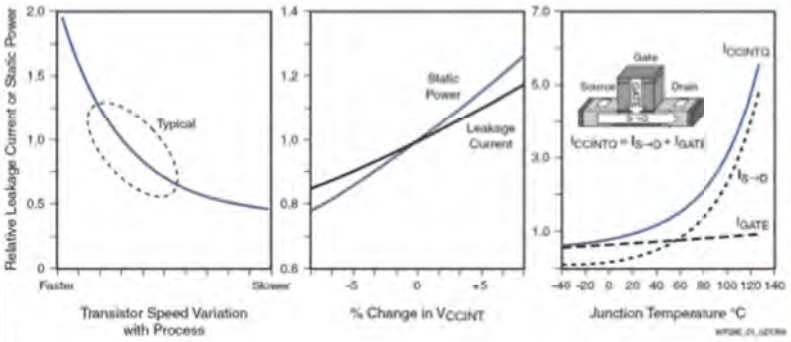
Static Power and Variation with Process, Voltage, and Temperature

- Static power and leakage are also strongly influenced by core voltage (VCCINT), with variations that are approximately the square and cube of VCCINT.
- Static power shows an approximate 15% increase with only a 5% increase in VCCINT. Leakage is strongly influenced by junction (or die) temperature (TJ).

SEBC-L12 MZ 16

La variazione di P_stat ha una dipendenza dalle variazioni dell'alimentazion almeno quadratica, ma anche cubica (Vdd+5% >> Pd +15%)
>> Spesso si lavora con alimentatori con incertezza del 5%, ed è pessimo che ciò porti ad un'incertezza nella P_stat del 15 %.
>> E' fondamentale nel caso delle FPGA preoccuparsi di mantenere Vdd al valore tipico.

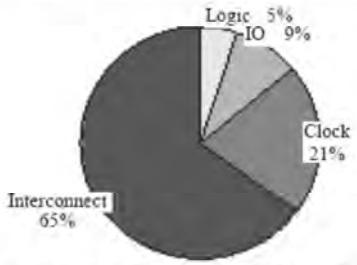
Static Power and Variation with Process, Voltage, and Temperature



SEBC-L12 MZ 17

Si hanno importanti aumenti nella I_leak legate a variazioni su:
-Vdd
-T
-Parametri del circuito
>> soprattutto nelle FPGA è molto importante tenere sotto controllo questi vari parametri. (es si mette un dissipatore un po' più grande del dovuto >> questo porta ad un abbassamento della T_junction con abbassamenti del consumo).

FPGA Dynamic Power



- The fine grain programmability of the FPGA puts stress on the interconnect structure.
- The interconnect is responsible for most of the energy consumption, while logic consumes only 5% of total energy.
- This breakdown is valid for the latest FPGAs since the architecture has remained more or less the same.

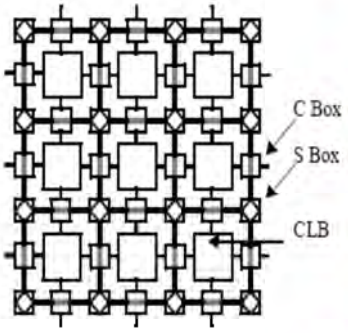
SEBC-L12 MZ 18

Si vede da qui come solo il 5% della potenza dinamica serve per fare qualcosa!!!
>> il 65% è speso dalle interconnessioni, ed è il prezzo da pagare per la programmabilità
>> il 20% è legato al clock.
Si può quindi pensare di fare low power solo lavorando su clk e interconnessioni!!

Mod by Giorgio Fissore, pag 293

FPGA – Interconnection Level 1

- The next level of connections is through a symmetric mesh Architecture
- This provides connections to blocks which cannot be reached through the NNC connections.

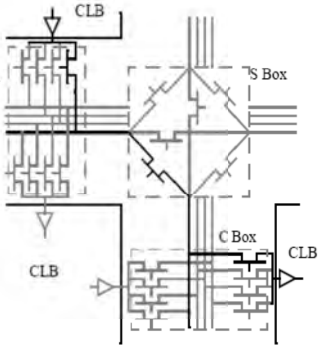


SEBC-L12 MZ 22

Rispetto ai nearest neighbours qui posso decidere di volta in volta a chi parlare scegliendo tra varie strade >> gli incroci sono programmabili.
-Il costo è legato al fatto che ogni volta che passo su un trans., questo, con la sua Req, ha un consumo >> più mi allontano più crescono energia e ritardo (crescita exp di E*D con la distanza) >> solo connessioni non troppo distanti.

FPGA – Interconnection Level 1

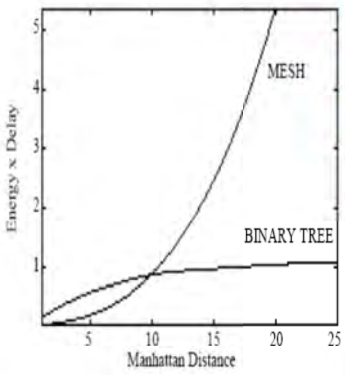
- The Connection C-Box provides full connectivity to the channel, and the Switch S-Box is Xilinx style.
- No major architectural modifications is done at this level, as the basic structure provides good routability.



SEBC-L12 MZ 23

FPGA – Interconnection Level 2

- For longer connection lengths (l) between the logic blocks, the delay increases as l^2 and Energy-Delay as l^3 in a Mesh architecture.
- This can be circumvented by having another level of interconnect which is dedicated for longer connections.
- The figure compares the ED metric of connections using the Mesh architecture and a binary tree connectivity in a 16x16 array.



SEBC-L12 MZ 24

Quando devo parlare con qualcuno più lontano la soluzione mesh non è più efficace (troppi incroci) >> connetto quindi con albero binario (ogni diramazione è uno switch box con un suo consumo)

Mod by Giorgio Fissore, pag 295