Towards DDSM-resilient systems

Paul Marchal, Marco Facchini, Satyakiran Munaga, Wim Dehaene, Francky Catthoor
The Ambient Intelligence Dream

Secure, trustworthy computing and communication embedded in every-thing and every-body.

A pervasive, context aware ambient, sensitive and responsive to the presence of people
AMI is based on *programmable tiles*

- Programmable cores for flexibility or thus for a lower manufacturing cost
- For the ‘milliwatt range’ (10-100Gop/sW) = energy-delay optimal

E.g., Philips Nexperia, ST Nomadic, TI OMAP
The easiest way for increasing the EDP? Technology scaling!

Ideal scaling model

- One scaling factor for all geometry: x 1/S
- One scaling factor for all voltages: x 1/U

For “constant electric field” scaling: U = S

Consequences

<table>
<thead>
<tr>
<th>Area/Device</th>
<th>WL</th>
<th>1/S^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic Delay</td>
<td>$C_{gate} \frac{V_{DD}}{I_{sat}}$</td>
<td>1/S</td>
</tr>
<tr>
<td>Power</td>
<td>$C_{gate} V_{DD}^2$</td>
<td>1/S^3</td>
</tr>
<tr>
<td>Energy-Delay</td>
<td>Power.Delay^2</td>
<td>1/S^5</td>
</tr>
</tbody>
</table>
Classic scaling breaks down in face of DDSM effects

1. Wide variation in performance
   - Sensitivity to noises due to reduced noise margin

2. Reliability: hard break down of dev./intercon.
   - Increased leakage
   - Unreliable storage due to process variations
   - Vulnerability to particle-caused upsets (soft errors)

3. New devices
   - Low Vt
   - Substr. noise
   - Power density
   - Thin oxides
   - High-k dielectrics
   - X-coupling

- SCE
- IR drop
- Process variations
- Low Vdd
- X-coupling
Classic scaling breaks down in face of DDSM effects

Logic gates

- Sensitivity to noises due to reduced noise margin
- Wide variation in performance
- Reliability: hard break down of dev./intercon.
- Increased leakage
- Unreliable storage due to process variations
- Vulnerability to particle-caused upsets (soft errors)

Memory
Delay variations translate into functional yield loss

- Strict timing between any path between any pair of registers required
- Violations to these constraints result in functional yield loss
Boosting yield requires many design margins.

Accumulation of design margins results in over-design, and thus a large power overhead or a worse EDP.
Statistical timing analysis/optimization for limiting the over-design

before:

\[ t_{\text{circuit}} (VDD, \text{Size}, T_{wc}, Vt_{wc}, L_{wc}, \text{Noise}_{wc}) < t_{\text{deadline}} \]

Ensure that the circuit meets the deadline for all worst-case conditions of the ambient/process/design corners.

High complexity required to avoid worst-case assumptions

Not useable for coping with time-varying, ambient variations as no statistical models exists for coping with this.

now

\[ P[t_{\text{circuit}} (VDD, \text{Size}, T, Vt, L, \text{Noise}) < t_{\text{deadline}}] > 99.7 \]

Given the statistical distributions of ambient/process/design conditions ensure that the probability that the circuits delay meets the deadline is sufficiently high.
Circuit/architectural techniques for limiting the over-design

Razor (U. Michigan)

A method for detecting errors

Allows to remove margins until errors occur and then to recover from that...

18x18-bit Multiplier Block at 90 MHz and 27 C

Supply Voltage (V)

Error rate

Environmental margin @ 1.6V

Zero-margin @ 1.6V

Error rate
A third alternative solution: self-timed circuits

Synchronous design
+ predictable performance
+ tool chain for designing, verifying, testing
- overhead due to timing closure

Self-timed approach:
+ no timing closure issues
- no performance guarantees
- large overhead

4-phase protocol
Open research questions

1. When does the self-timed approach outperform the synchronous one?
2. Can we reduce the EDP of these self-timed systems by exploiting domain/application specific knowledge?
Our approach for answering both questions

1. Avoid tool issues: reuse the desynchronization flow

2. Investigate robust and low cost completion detection circuits

3. Look for potential reductions of the overhead of self-timed at the architectural level

4. Performance closure by feedback control
1. Reuse of ‘an almost synchronous design-flow’

De-synchronization (cfr. TCAD Cortadella 2005):
- Start with synchronous design
- Split flip-flops into latches
- Replace global clock with latch controllers and matched delays
- Latch controllers communicate with neighboring controllers with handshake protocol

Provably equivalent behavior and timing to that of original synchronous circuits

=> same HDL, logic synthesis, layout, verification, extraction, automated test pattern generation as in the synchronous design flow.

Completion detection circuit based on matched delays breaks down in face of extreme process variations.
2. Robust completion detection methods

- Existing completion detection techniques
  - Delay lines:
    a dummy circuit models the worst-case possible delay of the data-path. Difficult to guarantee under large variations unless by taking design margins.
  - Current sensing:
    detect based on the current whether a circuit is ready or not. Not robust
  - Encoding schemes are probably more robust:
    most general encoding scheme for logic: dual rail encoding
The ‘safe’ dual rail approach...

Completion detection is encoded inside the signal.
...has large overhead.

static CMOS

more activity
more wires

bit-level
completion
detection
logic

differential cascode
switch logic

© imec 2005
Can we reduce the cost of robust completion detection circuits?

- Reducing the cost of dual rail
  - Reducing the wire-overhead:
    - custom layout of a data-path
    - layout with a module-generator
    - more stacked circuits (gives potentially a better energy-delay product)
  - Compound logic

- Hybrid solutions
  - selective insertion of completion detection circuits in combination with clocked/delay line based parts based on a thorough analysis of which parts of the data path are most affected by performance variations.

- Collaboration with Alex Yakovlev, U. Newcastle and Wim Dehaene, ESAT.
3. Reducing the de-synchronization overhead at the architectural level

- Many papers exist which investigate "energy-delay optimal architectural organization" for synchronous architectures.
- What about the optimal architectural organization for self-timed systems?
  - E.g., impact of sync. overhead on parallelism/logic depth?

Collaboration with FEENICS team at IMEC
How do we ensure that the available MOPS of the platform are at any time larger than the MOPS required by the application (with the lowest possible energy consumption)?
...with a feedback control system

Application

- These feedback control systems are common in embedded systems today:
  - Vdd/VT hopping, DVS, TCM, ...
  (see Ph.d. P. Yang for overview)

- Crux lies in low-cost performance/energy measurement

- Specific/local knobs to increase the performance of the asynchronous system.

Knobs to adjust speed

(part of) Ph.D. of Satyakiran Munaga
An example: run-time configurable memories for performance closure (1)
An example: run-time configurable memories for performance closure (1)

Normalized Access Delay

Normalized Energy per Access

worst-case
design point

Normalized Access Delay

Normalized Energy per Access

worst-case
design point
An example: run-time configurable memories for performance closure (2)

![Graph showing average energy vs. execution time](image)

- 1/10000
- 1/1000
- 1/100
- 1/10000: slack
- 1/1000: slack
- 1/100: slack

More information: Papanikolaou05, CODES
Classic scaling breaks down in face of DDSM effects

Logic gates

- Sensitivity to noises due to reduced noise margin
- Wide variation in performance
- Reliability: hard break down of dev./intercon.
- Increased leakage
- Unreliable storage due to process variations
- Vulnerability to particle-caused upsets (soft errors)

Memory
The classical reliability approach...

Definition of Reliability

- 1st failure of a device anywhere in the chip will cause chip failure
- impact of failure at circuit level is not considered in the reliability assessment
- designers do not (have to/want to) care about reliability, reliability is qualified at technology level, is considered job of technology/process engineer
...breaks down in light of DDSM effects

- Ever more stringent constraints on devices/interconnect as the chip size increases
- Increase in intrinsic fields and currents, increased power density (e.g., increased Vdd to cope with S/D leakage increases electrical field across gate)
- Introduction of many new materials (*high k gate dielectrics, metal gates, low k dielectrics, Cu metallization, nano-devices*)

Reliability can no longer be guaranteed at the device level
Moreover, devices/interconnects wear out rather than abruptly breakdown.

Thicker oxides: breakdown occurs by abrupt change in current or voltage, breakdown is always hard.

Thinner oxides: breakdown only leads to small change in current (soft breakdown), followed by a long period of gradual progressive wear-out.

Hosoi et al., IEDM 2002
Building reliable systems from unreliable components

- How do we cope with **wear out** of the devices?
  - measurement and test issues which can only be resolved after better understanding how reliability mechanisms impact the circuits/systems
  - delay-variation resilient architecture?
  - preventing/healing wear out of the system (e.g., UCSD Simunic et al.)?

- How do we cope with the **inevitable hard breakdowns**?
  - How do build a fault-tolerant/bullet proof architecture in an energy-efficient way? (e.g., U. Mich. Constantinides et al)

- What about **transient functional errors** (e.g., SER)?

Giacomo Paci
(with the help of the reliability group of IMEC)
Can new devices help in resolving some DDSM issues?

- Sensitivity to noises due to reduced noise margin
- Wide variation in performance
- Reliability: hard break down of dev./intercon.
- Increased leakage
- Unreliable storage due to process variations
- Vulnerability to particle-caused upsets (soft errors)

Logic gates

Memory
Novel MOSFET Structures (ITRS)

Table 59a  Single-gate Non-classical CMOS Technologies

<table>
<thead>
<tr>
<th>Transport-enhanced FETs</th>
<th>Ultra-thin Body SOI FETs</th>
<th>Source/Drain Engineered FETs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strained Si, Ge, SiGe</td>
<td>BOX</td>
<td>S D BOX (&lt;20nm)</td>
</tr>
<tr>
<td>buried oxide</td>
<td>Ground</td>
<td>BOX</td>
</tr>
<tr>
<td>Silicon Substrate</td>
<td>Plane</td>
<td>Bulk wafer</td>
</tr>
<tr>
<td></td>
<td>FD Si film</td>
<td>Schottky barrier isolation</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>S D</td>
</tr>
</tbody>
</table>

Table 59b  Multiple-gate Non-classical CMOS Technologies

<table>
<thead>
<tr>
<th>Multiple Gate FETs</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-Gate (N&gt;2) FETs</td>
</tr>
</tbody>
</table>

[Sotnicki, C&Dmag, 1/05]
...are potentially much better!

Prediction is that the variability on VT will be heavily reduced for fully depleted SOI devices.

[Yamaoka, VLSI04]
... but there are still many trade-offs to be made

S/D, and thus ON-Resistance is large $\Rightarrow$ thick fin?
This heavily affects intrinsic delay!
Smaller subthreshold slope $\Rightarrow$ thin fin?
This is better for leakage

[Imec, Dixit, TEIDev6/05]

[Peit, TEIDev8/02]
...and new circuits to be explored

Different properties lead to different topologies
E.g. multiple gate opens new options.

[Guo, ISLPED05]

70% better SNM!
...moreover, the exploration space is very large

- **Device configurations:**
  - Bulk: standard si, but multi-gate
  - FinFETs

- **Device geometries:**
  - Particularly for FinFETs
  - Which ones are fixed, which ones can we control

- **Technology parameters:**
  - Implant dosage
  - Lithography constraints
  - ....

- **Circuit topologies:**
  - SRAMs: Cells, decoder, sense-amplifiers, feedback (multi-gate devices!), redundancy
  - Logic: ....

- **Wire engineering:**
  - Aspect ratios, pitches, …
co-exploration of devices/circuits

- Technology screening:
  A plurality of technology options exist for EDP optimal devices: scaling the devices, different devices, device geometries, interconnect options, etc. There is a need for a methodology for identifying which technology options are most suited and affordable for the circuit’s energy/delay/yield.

- Energy/delay/yield aware design:
  However, it’s largely unknown how the desired circuits/systems on the energy/delay trade-off can be realized. E.g., what are the circuit/device differences in the design of an ultra-low power (100Mhz) point compared to a more regular (500Mhz-1Ghz) operating point?
  How do different devices or device options impact the circuit/system design?
First tape-out of a FINFET SRAM array

- Objectives:
  - verification of our finfet compact model required for designing larger test structures.
  - do finfets really help to improve stability of the cells?
  - do finfets really reduce leakage?
  - what’s the impact of fin width on the device currents/variation s?
  - comparison of different processing options (such as FUSI vs. metal gate)

16-bit array with devices of different fin widths, fin lengths
Conclusions

- **Delay variations due to DDSM effects translate directly into functional yield loss**
  - they render the timing closure problem very complex at the circuit-level, which cannot be resolved but by accumulating many design margins.
  - we therefore propose to look into the self-timed self-adaptive architectures
    - de-synchronization
    - reducing the cost of completion detection
    - from timing closure to performance closure

- **Reliability will an issue in DDSM technologies**
  - reliability will no longer only manifest itself as abrupt changes in electrical parameters, resulting in permanent errors, but rather behave as wear out of all the devices
  - can we build architectures for coping with these errors?

- **Many promising novel devices are engineered which may alleviate some of the DDSM effects (such as better control over (gate)-leakage), however, their impact on energy-delay-yield of circuits has yet to be explored.**
Tasks

- Front end modeling:
  - What:
    - Parameterized device modeling and calibration
    - Modeling of the local interconnect stack
    - Sensitivity analysis of devices/wires to process variations
  - How:
    - 3D device simulators
    - Analytical models
    - Test structures for measurement of the unknowns

- Design with novel devices
  - Ultra-low power SRAM (low performance)
  - Low-power SRAM (high performance)

- Development of a screening methodology
- Development of a technology migration methodology

- So far, we only focus on memories.