NVIDIA Announces CPU for Giant AI and High Performance Computing Workloads

NVIDIA Grace is the company’s first data center CPU, an Arm-based processor that will deliver 10x the performance of today’s fastest servers on the most complex AI and high performance computing workloads.

NVIDIA, Monday, April 12, 2021
Announcements

• Quiz today
• Assignment 4 due in two weeks
Outline

• Finish dynamic voltage scaling
• Reducing switching activity
• Leakage management
5.F Dynamic Voltage Scaling
Alternative: Error Detection

Bull, ISSCC’2010
Design for Dynamically Varying VDD

- Static CMOS logic.
- Ring oscillator.
- Dynamic logic (& tri-state busses).
- Sense amp (& memory cell).

Max. allowed $|dV_{DD}/dt| \rightarrow \text{Min. } C_{DD} = 100\text{nF (0.6\mu m)}$

Circuits continue to properly operate as $V_{DD}$ changes
Static CMOS robustly operates with varying $V_{DD}$.

$V_{in} = 0, V_{out} = V_{DD}$

0.6 $\mu$m CMOS: $|dV_{DD}/dt| < 200$V/$\mu$s

- Static CMOS robustly operates with varying $V_{DD}$. 

$\max. \tau = 4$ns
Ring Oscillator

**Simulated with $dV_{DD}/dt = 20V/\mu s$**

- Output $f_{CLK}$ instantaneously adapts to new $V_{DD}$. 
Dynamic Logic

\[ V_{DD} \]
\[ V_{out} \]
\[ V_{in} \]
\[ \text{clk} \]

\[ V_{DD} \]
\[ V_{out} \]
\[ V_{in} \]
\[ \text{clk} \]

\[ \Delta V_{DD} > V_{TP} \]

\[ \Delta V_{DD} > V_{be} \]

Errors

- False logic low: \( \Delta V_{DD} > V_{TP} \)
- Latch-up: \( \Delta V_{DD} > V_{be} \)

0.6\( \mu \)m CMOS: \( |dV_{DD}/dt| < 20V/\mu s \)

- Cannot gate clock in evaluation state.
- Tri-state busses fail similarly \( \rightarrow \) Use hold circuit.
Dynamic operation can increase energy efficiency > 10x.

Burd
ISSCC’00

Measured System Performance & Energy

Dhrystone 2.1 MIPS

Energy (mW/MIPS)

Dynamic $V_{DD}$

Static $V_{DD}$

85 MIPS @ 5.6 mW/MIPS (3.8V)

6 MIPS @ 0.54 mW/MIPS (1.2V)

• Dynamic operation can increase energy efficiency > 10x.
**V\textsubscript{DD}-Hopping**

Application slicing and software feedback guarantee real-time operation.

Two hopping levels are sufficient.

Transition time between $f$ levels = 200\textmu s
Dithering Between Supply Levels

- Done with switched-capacitor DC-DC converters which efficiently work only at discrete levels

Keller et al, ESSCIRC’16
Dithering Between Supply Levels

• Dithering fills in between fixed DC-DC modes

Keller et al, ESSCIRC’16
5.G Reducing Switching Activity Through Logic Design
## Power/Energy Optimization Space

<table>
<thead>
<tr>
<th>Energy</th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td></td>
<td>Logic design</td>
<td>Clock gating</td>
</tr>
<tr>
<td></td>
<td>Scaled $V_{DD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trans. sizing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multi-$V_{DD}$</td>
<td></td>
</tr>
<tr>
<td><strong>Leakage</strong></td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
<td>Multi-$V_{DD}$ Variable $V_{Th}$</td>
</tr>
<tr>
<td></td>
<td>Scaling $V_{DD}$</td>
<td>+ Input control</td>
</tr>
<tr>
<td></td>
<td>+ Multi-$V_{Th}$</td>
<td></td>
</tr>
</tbody>
</table>
Basic Idea

• $E \sim \alpha CV^2$

• Reduce switching activity, $\alpha$, through logic and architectural transformations

• Many options
  • Switching activity lower with deeper logic
  • Pipelining has significant effect
  • Reduce the number of clocked devices in a flip-flop
    • e.g. group generation of clk_b
  • A few logic ideas follow
Circuit-Level Activity Encoding

Conditional Inversion Coding for Interconnect

from [Stan94]
(1994 International Workshop on Low-power Design)
Number Representation

• Input signals are noise most of the time

- Sign-extension activity significantly reduced using sign-magnitude representation
5.H Clock Gating
# Power/Energy Optimization Space

<table>
<thead>
<tr>
<th>Energy</th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td></td>
<td>Logic design</td>
<td>Clock gating</td>
</tr>
<tr>
<td></td>
<td>Scaled V$_{DD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trans. sizing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multi-V$_{DD}$</td>
<td></td>
</tr>
<tr>
<td>Leakage</td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
<td>Multi-V$<em>{DD}$ Variable V$</em>{Th}$</td>
</tr>
<tr>
<td></td>
<td>Scaling V$_{DD}$</td>
<td>+ Input control</td>
</tr>
<tr>
<td></td>
<td>+ Multi-V$_{Th}$</td>
<td></td>
</tr>
</tbody>
</table>
Clock Gating

Requires a bit more complex gate ...
Well handled in today’s EDA tools
Clock Gating

- Enabling clock needs to be synchronized

Sequential cell

Clk
En
Gated Clk
Clock Gating Efficiently Reduces Power

Without clock gating: 30.6mW
With clock gating: 8.5mW

90% of F/F’s were clock-gated.
70% power reduction by clock-gating alone.

Courtesy M. Ohashi, Matsushita, ISSCC 2002
Dynamic high level clock gating activity

When dynamic high level clock gating is enabled the clock of the integer core is cut in the following cases:
- the integer core is empty and there is an instruction miss causing a linefill
- the integer core is empty and there is an instruction TLB miss
- the integer core is full and there is a data miss causing a linefill
- the integer core is full and data stores are stalled because the linefill buffers are busy.

When dynamic clock gating is enabled, the clock of the system control block is cut in the following cases:
- there are no system control coprocessor instructions being executed
- there are no system control coprocessor instructions present in the pipeline
- performance events are not enabled
- debug is not enabled.

When dynamic clock gating is enabled, the clock of the data engine is cut when there is no data engine instruction in the data engine and no data engine instruction in the pipeline.
Local Clock Gating
Local Clock Gating

Data-Transition
Look-Ahead

Pulse
Generator

‘Clock on demand’
Flip-flop

EECS241B L21 LEAKAGE
Complex Designs

Clock System Architecture

Variable Supply

Fixed Supply

Bus Clock

Phase Aligner

Fischer, ISSCC'05
## Power /Energy Optimization Space

<table>
<thead>
<tr>
<th>Energy</th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td></td>
<td>Logic design</td>
<td>Clock gating</td>
</tr>
<tr>
<td></td>
<td>Scaled $V_{DD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trans. sizing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multi-$V_{DD}$</td>
<td></td>
</tr>
<tr>
<td><strong>Leakage</strong></td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
<td>Multi-$V_{DD}$ Variable $V_{Th}$ + Input control</td>
</tr>
<tr>
<td></td>
<td>Scaling $V_{DD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ Multi-$V_{Th}$</td>
<td></td>
</tr>
</tbody>
</table>
5.1 Lowering Leakage During Design: Multiple Thresholds
Power /Energy Optimization Space

<table>
<thead>
<tr>
<th>Energy</th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td></td>
<td>Logic design</td>
<td>Clock gating</td>
</tr>
<tr>
<td></td>
<td>Scaled $V_{DD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trans. sizing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multi-$V_{DD}$</td>
<td></td>
</tr>
<tr>
<td><strong>Leakage</strong></td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td></td>
<td>Trans sizing</td>
<td>multi-$V_{DD}$</td>
</tr>
<tr>
<td></td>
<td>Scaling $V_{DD}$</td>
<td>Variable $V_{Th}$</td>
</tr>
<tr>
<td></td>
<td>+ Multi-$V_{Th}$</td>
<td>+ Input control</td>
</tr>
</tbody>
</table>

---

EECS2418 L21 LEAKAGE
Using Multiple Thresholds

- Cell-by-cell $V_T$ assignment (not block level)
- Allows us to minimize leakage
- Achieves all-low-V performance

Yano, SSTCW'00
Typical Technologies

• 2-3 Thresholds
  • To choose from 4-6 in a node
  • In bulk and finfet, but not in FDSOI (unless doped)

• Threshold voltage diff \(\sim 5-10x\) in leakage
5.1 Lowering Leakage During Design: Longer Channels
# Power /Energy Optimization Space

<table>
<thead>
<tr>
<th>Energy</th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td>Logic design</td>
<td>Clock gating</td>
<td>DFS, DVS</td>
</tr>
<tr>
<td>Scaled $V_{DD}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trans. sizing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-$V_{DD}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Leakage</strong></td>
<td>Stack effects</td>
<td>Sleep T’s</td>
</tr>
<tr>
<td>Trans sizing</td>
<td>Multi-$V_{DD}$</td>
<td>DVS,</td>
</tr>
<tr>
<td>Scaling $V_{DD}$</td>
<td>Variable $V_{Th}$</td>
<td>Variable $V_{Th}$</td>
</tr>
<tr>
<td>+ Multi-$V_{Th}$</td>
<td>+ Input control</td>
<td></td>
</tr>
</tbody>
</table>

EECS2418 L21 LEAKAGE
Longer Channels

- 10% longer gates reduce leakage by 35% (in 130nm)
- Increases switching energy by 21% with $W/L = \text{const.}$

- Attractive when don’t have to increase $W$ (memory)
- Doubling $L$ reduces leakage by 3x (in 0.13um)
- Much stronger effect in 28nm!
- Effect improves with shorter channel devices
Poly Bias

• 28FDSOI example
Next Lecture

• Low-power design
  • Power gating