Timing-Aware high level power estimation of industrial interconnect module

Amal Ben Ameur¹, Antonio Genov¹,², François Verdier¹, Loic Leconte²

(1)University of Cote d'Azur, LEAT, Sophia Antipolis, France
(2)NXP SEMICONDUCTORS, Mougins, France
<table>
<thead>
<tr>
<th></th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Context &amp; Motivations</td>
</tr>
<tr>
<td>2</td>
<td>PwClkARCH library</td>
</tr>
<tr>
<td>3</td>
<td>Design Under Test</td>
</tr>
<tr>
<td>4</td>
<td>Simulations, Results and Correlation</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Context and Motivations
PwClkARCH library

- C++/SystemC-TLM library inspired from the UPF standard
- Based on the co-simulation between a power model description and a SystemC-TLM based virtual prototype
- Ensures the estimation of the power metrics dynamically, and the application of hardware power management strategies (clock gating, power gating, DVFS, etc.)
PwClkARCH library

1. Decomposition of the architecture into different **power domains** and **clock domains**.

2. **PMU** implementation in SystemC-TLM that interfaces the functional and the power models.

3. **Simulation** of the complete architecture and the generation of power results.

4. **Verification** of the **consistency** between the power management properties and the functional behavior.
PwClkARCH library

SC_HAS_PROCESS(Interconnect);
Interconnect(sc core::sc_module_name name, protocol::bus type UsedProtocol,
    arbitration::ArbiterSettings arbiter settings, sc core::sc_time clk_period)
{
    #ifdef PWARCH
    sc core::sc_object& obj = dynamic_cast <sc core::sc_object&>(*this);
    m_PowObserver = new Observer<sc core::sc_object&>(obj,"_ob");
    this->attach(*m_PowObserver);
    de = Design_elem::get_DE(obj);
    SC THREAD(checkActivity);
    #endif
}

void checkActivity()
{
    while(true)
    {
        wait(m_ActivateEvent);
        de->set_functional_state(false);//ACTIVE
        de->Update_dpow(1.6);//Set the activity factor
        wait(m_DeactivateEvent);
        de->set_functional_state(true);//IDLE
    }
}

Design Element

If consumption changes
Then update power

Total Power :
\[ P_t = P_d + P_s \]  (1)

Dynamic Power :
\[ P_d = P_{cc} + P_{co} \]  (2)

Static Power :
\[ P_s = V_{dd} \cdot \text{leakage} \]  (5)

Short-circuit Power :
\[ P_{cc} = V_{dd} \cdot I_{cc} \]  (3)

Power dissipation :
\[ P_{co} = \alpha \cdot f \cdot C_l \cdot V_{dd}^2 \]  (4)

\[ P_t = V_{dd} I_{cc} + \alpha f C_l V_{dd}^2 + \frac{V^2}{R_{leakage}} \]
• NXP i.MX8QM SoC

• Many Subsystems
  • Audio, VPU, GPU, CPU, SCU ...

• **Switch Matrix** – Connection of all subsystems
Design under test - description

- Interconnection module
- Implements multiple QoS and routing algorithms
- Contains multiple sub-blocks
- ~5 power domain
- More than 25 clock domains

auto clock gating

+ power gating
Design under test - testbench

- The stimuli come from generic or application-specific Traffic Generators (SystemC initiator modules)
- Two DRAM memory models
MemCopy 256KB

- 256KB DRC read accesses
- Each data access is 16 bytes
- Sequential memory accesses (prev_addr+32bytes).
  - 4K interleaving applied in order to optimize the memories usage.
  - The spikes come from the moments of interleaving between the two memories.
- Good power consumption silicon correlation
Display Refresh HD1080-1920x1080@60-32b with DC prefetch enabled

- One frame of transactions.
- Only one Group/QOS is used.
- Data paths are 256 bits.
- Pixel size 32 bits.
- Pixel frequency 138.5 MHz.
- Refresh rate 60Hz (Frame rate 60fps).
- Each data access is 32 bytes and we consider sequential memory accesses.
- 4K interleaving applied in order to optimize the memories usage.

- Good power consumption silicon correlation
# Silicon correlation

<table>
<thead>
<tr>
<th>Power distribution</th>
<th>Silicon/Simulation correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch Matrix dynamic power</td>
<td>85 - 99.5%</td>
</tr>
<tr>
<td>Switch Matrix static power</td>
<td>85 - 90.9%</td>
</tr>
<tr>
<td>2 DRC dynamic power</td>
<td>90 - 99.9%</td>
</tr>
<tr>
<td>2 DRC static power</td>
<td>90 - 96.1%</td>
</tr>
<tr>
<td>Total power under 1.1V</td>
<td>88% – 98%</td>
</tr>
</tbody>
</table>
Conclusion

Comparing to previous PwClkARCH utilization:
1. Complexity of the functional model is tripled.
2. The number of design elements and clock domains is significantly increased.
4. The extracted power related metrics present a promising correlation with silicon measurements.

PwClkARCH performances:
1. Easily implemented on complex NXP IP.
2. Promising early stage power consumption silicon correlation.
3. Quick bugs detection/evaluation/correction.

Importance of industrial collaborators for PwClkARCH:
1. Industrial testing and proof-of-concept.
2. Maturation.
3. Bugs detection and correction.
4. Further add-on ideas.

We are searching for more collaborators interested by this new technology (PwClkARCH)
Contact: francois.verdier@univ-cotedazur.fr
Thank You!

Q&A