Measuring and Predicting Temperature Distributions on FPGAs at Run-time

2011 International Conference on Reconfigurable Computing and FPGAs, November 30 – December 2, Cancun, Mexico

Markus Happe, Andreas Agne and Christian Plessl
Motivation

Thermal management on hybrid multi-core: thermal thread mapping

- step 1: being able to measure the current temperature distribution
- step 2: predict thermal effects of thread re-mappings (thermal model)
Agenda

- Measuring temperature distributions
  - sensor layout
  - local heaters
  - proposed self-calibration technique

- Predicting temperature distributions
  - proposed thermal model
  - proposed on-line parameter learning technique

- Experimental results
  - measuring thermal gradients
  - parameter learning, measurements vs. predictions

- Outlook and summary
Measuring Temperature Distributions: Ring Oscillators

- Frequencies of ring oscillators react to changing temperatures (almost linear)
  - temperature influences switching speed of transistors
  - rising temperatures lead to decreasing oscillation frequencies
  - calibration via external devices: temperature controlled oven, infrared camera
  - related work: [Lopez-Budeo:02], [Velusamy:05]

- Drawbacks
  - frequencies also react to changing workload / voltage drops
  - usually external devices needed for calibration
Measuring Temperature Distributions: Sensor Grid

Measuring temperature distribution

- regular grid of sensors
- when some parts are not reconfigurable: gauge the temperature from neighboring tiles
- sensors connected to a thermal monitor

Measurement procedure

- enable sensors, enable time $T_1$ clock cycles
- count oscillations for $T_2$ clock cycles
- disable sensors

exemplary sensor layout for a Xilinx Virtex-6 FPGA LX240T
Measuring: Creating Thermal Gradients using Local Heaters

**Goals**
- evaluate thermal sensors
- calibrate sensors, learn thermal model

**Local heaters (heat-generating cores)**
- 10,000 toggling flip flops
- constrained to regions
- used for sensor calibration and parameter learning for thermal model (initially)
- replaced by hardware threads (run-time)

**Thermal gradients**
- temporal: on/off all local heaters
- spatial: on/off subset of local heaters

exemplary heater layout for a Xilinx Virtex-6 FPGA LX240T
Measuring: Proposed Self-calibration Technique

Goal: calibrate thermal sensors without external devices

Prerequisite: FPGA contains built-in pre-calibrated thermal diode (Xilinx Virtex 5/6)

Sensor self-calibration
- disable all local heaters and wait until temperature of thermal diode stabilizes
- (1) store current temperature (diode) and counter values of all sensors
- enable all local heaters and wait until temperature of thermal diode stabilizes
- (2) store current temperature (diode) and counter values of all sensors
- compute linear transformation functions $T(f)$ for each sensor from (1) and (2) which translate frequencies (counter values) into temperatures

Assumption
- chip will eventually reach thermal equilibrium in (1) and (2)
Predicting Temperature Distributions: Thermal Model

Background: simulating heat flows
- duality between electrical and thermal phenomena
- heat flow on chip modelled as RC network
- related work: HotSpot

Our thermal model
- chip: regular grid of nodes
- two layers (trade-off: accuracy and computational effort)
- R, C parameters learned at run-time
Heat flow to node $i$ from its neighbors $N(i)$ and the heat sink $S$:

$$I_{n}(i) = \sum_{j \in N(i)} \frac{T(j)-T(i)}{R(i,j)} \quad I_{sink}(i) = \frac{T_{s}-T(i)}{R_{s}(i)}$$

Temperature change:

$$\Delta T_{i} = \frac{I_{n}(i)+I_{source}(i)+I_{sink}(i)}{C(i)} \Delta t$$

- $\Delta T_{i}$: heat difference between current and next time step
- $T(j)$: temperature of tile $j$
- $I_{source}(i)$: heat flow from heat source on tile $i$ (local heater, task) [first layer]
- $I_{sink}(i)$: heat flow to heat sink [second layer]
- $C(i)$: heat capacity of the $i$-th tile
- $\Delta t$: time interval
Model parameters can be defined using material properties by the manufacturer; see HotSpot: [Huang:06]

We try to learn these parameters at run-time:
- The system becomes aware how the heat distributes on its chip.
- We use a reduced number of parameters according to simplified assumptions.
- We use local heaters to create different spatial and temporal thermal gradients.
- The measured temperatures are stored and used to learn the model parameters.

On-line learning algorithm:
- Use a starting point that matches FPGAs (experimental results).
- Randomized hill climbing to learn model parameters.
- Minimizes mean root square error between measurements and simulations.
Predicting Temperature Distributions

Predictions according to learned thermal model

- thermal model is initialized with current temperature readings from sensor grid
- according to the scenario the heat sources for the tiles have to be updated
- heat sources are task-dependent and learned at run-time
- first results: tasks represented by local heaters
- continuously update the temperatures for all tiles according to thermal model
- stop when desired time step or thermal equilibrium is reached
Experimental Results: FPGA Setup – Xilinx Virtex-6 LX240T

- **Sensor grid layout:** 10x15 tiles
  - **Tile:** 16x16 slices

- **Measurement temperature range:** 48°C -> 56°C

- **Local heater layout:** 12 heaters
  - Each side: 5 heaters
Timeline for System Setup

- **Calibrate**
  - sensor self-calibration technique using local heaters and thermal diode

- **Generate gradients**
  - data acquisition using different heating scenarios for learning step

- **Learn model parameters**
  - use randomized hill climbing algorithm to learn model parameters

- **Run:** system can use thermal sensors and predict heat flows
Experimental Results: Temperature Measurement

- Spatial gradients up to 6.5°C
- Temporal gradients up to 8°C

temperature measurement on Virtex-6 LX240T when top five local heaters are enabled
Experimental Results: Learning / Measurements vs. Predictions

- Learning parameters took between 50-60 minutes on MicroBlaze (100 MHz).
- Predicting 12-minute scenario takes 99.5 s on MicroBlaze (100 MHz) for $\Delta t = 0.02$.
- Scenario: heat entire chip (1-3), top (4-5), bottom (6-7), right (8-9), left (10-11) side.
Measurements vs. Predictions

- Video: measurements (left) vs. predictions (right)
- Temperature range: 45°C - 57 °C
Summary and Outlook

Summary

- measuring temperature distribution using sensor grid
- sensor self-calibration using built-in thermal diode and local heaters
- measured spatial gradients up to 6.5 °C on Xilinx Virtex-6 FPGA

- 2-layer thermal model of FPGA (RC network)
- on-line learning technique for model parameters
- predictions according to thermal model: accuracy 0.72°C

Outlook

- quantify calibration error using measurements from infrared camera
- extend proposed methods to hybrid multi-cores
- study thermal effects on thread re-mappings
- develop and analyze different thread remapping strategies