Aging-Aware Compiler-Directed VLIW Assignment for GPGPU Architectures

Abbas Rahimi†, Luca Benini‡ and Rajesh K. Gupta†
†UC San Diego and ‡Università di Bologna

Variability is about Scale and Cost
- NBTI-induced performance degradation
- $\Delta V_{TH} = F(\text{Process, Temp, Voltage, Stress})$
- Knobs to tweak: {Power-gating, Duty cycling}

AMD Evergreen GPGPU Architecture

GPGPU Workload Variation
- Inter-Compute Units: Uniform
- Inter-Processing Elements: Uneven!
- Stress consumes timing margin
- Inter-Stream Cores: SIMD fashion
- $\text{PE}_x$ executes $\sim$40% of instructions
- The most aged PE limits lifetime

Aging-Aware Compilation

1.Observability: Reading NBTI sensors measurements
2.Prediction: Static code analysis technique estimates the percentage of instructions that will carry out on every PE (a linear calibration module later fits the predicted $\Delta V_{TH}$ shift to the observed $\Delta V_{TH}$ shift).
3.Controllability: Uniform slot assignment assigns fewer/more instructions to higher/lower stressed slots.
4.Periodic healthy kernel generation

Results
- Adapting kernels periodically leads to a uniform $\Delta V_{TH}$ among PEs (without power-gating)
- *Reduces $\Delta V_{TH}$: up to 49% (11%) and on average 34% (6%) in presence/absence of power-gating supports compared to naive kernels.
- *Entire flow, on average 13ms
- *Kernel disassembly using online CAL (95% total time)
- *Static code analysis: 220K–900K cycles
- *Slot assignment algorithm ≤ 2K cycles

Work in progress: Memory subsystems, reducing $\Delta V_{TH}$ by up to 43% for register files.