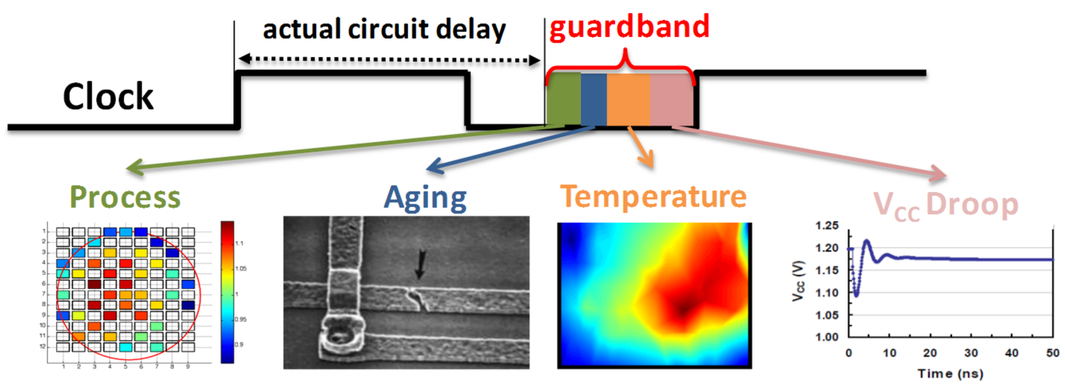
Aging-Aware Compiler-Directed VLIW Assignment for GPGPU Architectures

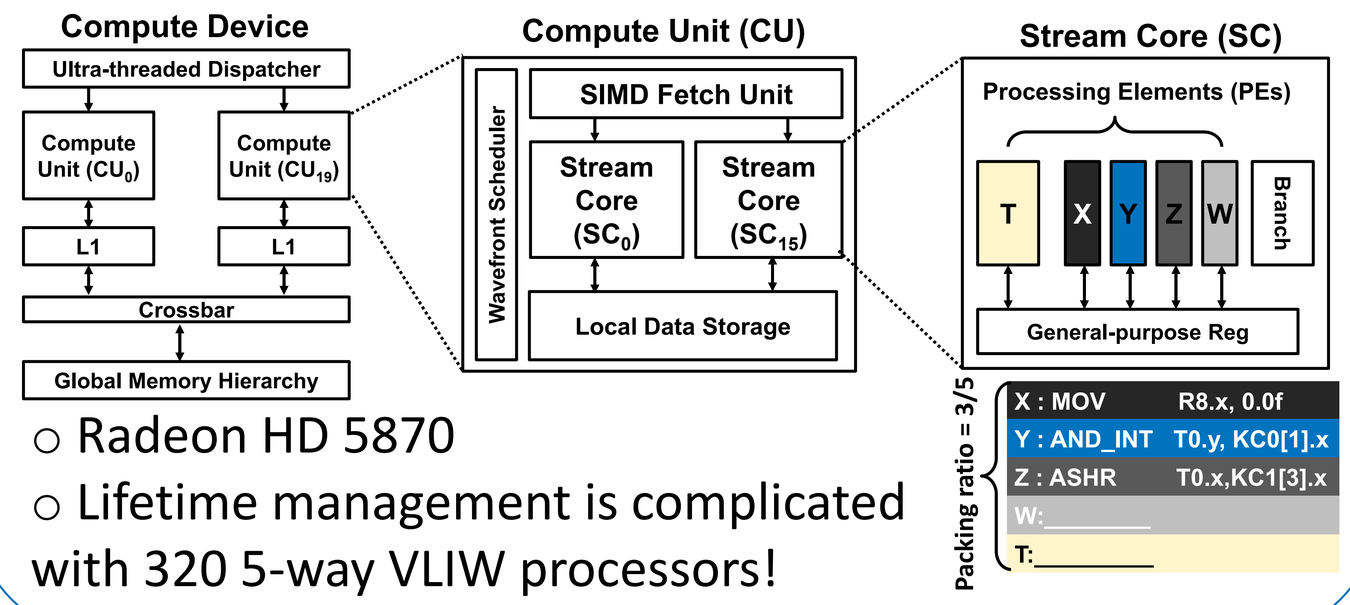
Abbas Rahimi[†], Luca Benini[‡] and Rajesh K. Gupta[†]
[†]UC San Diego and [‡]Università di Bologna

Variability is about Scale and Cost



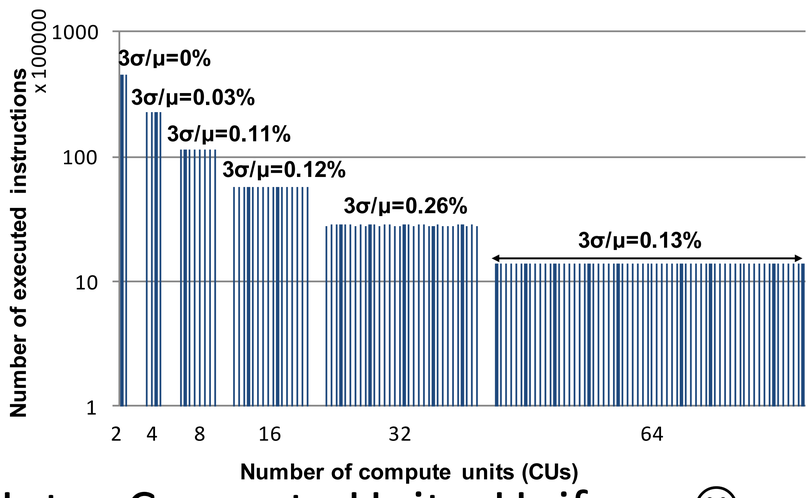
- NBTI-induced performance degradation
- ΔV_{TH} = F (Process, Temp, Voltage, Stress)
- Knobs to tweak: {Power-gating, Duty cycling}

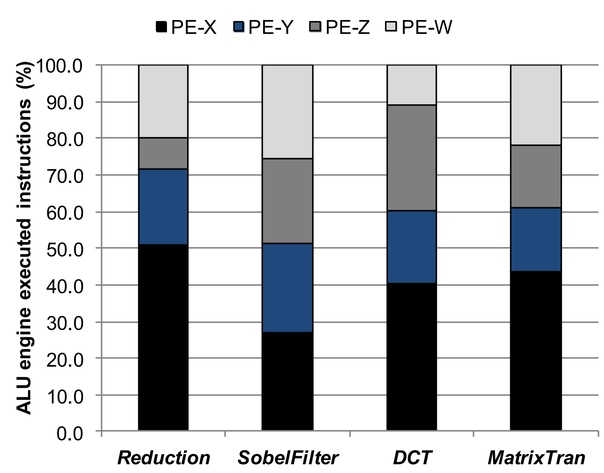
AMD Evergreen GPGPU Architecture

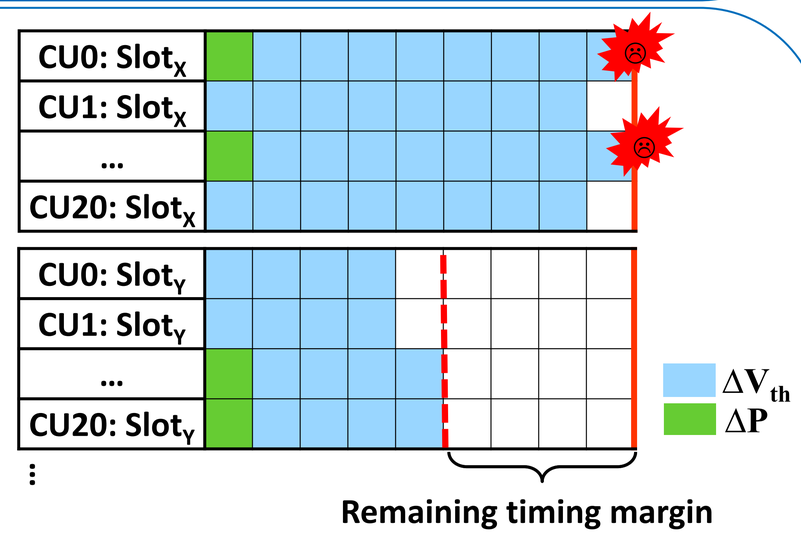


- Radeon HD 5870
- Lifetime management is complicated with 320 5-way VLIW processors!

GPGPU Workload Variation



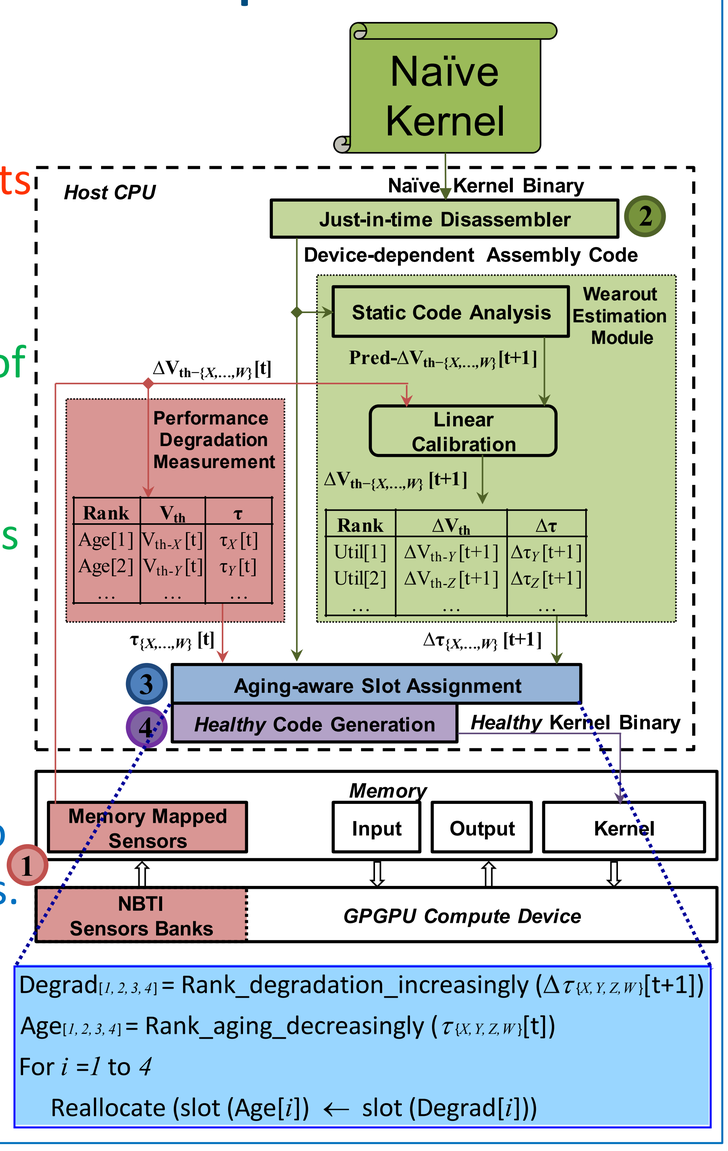




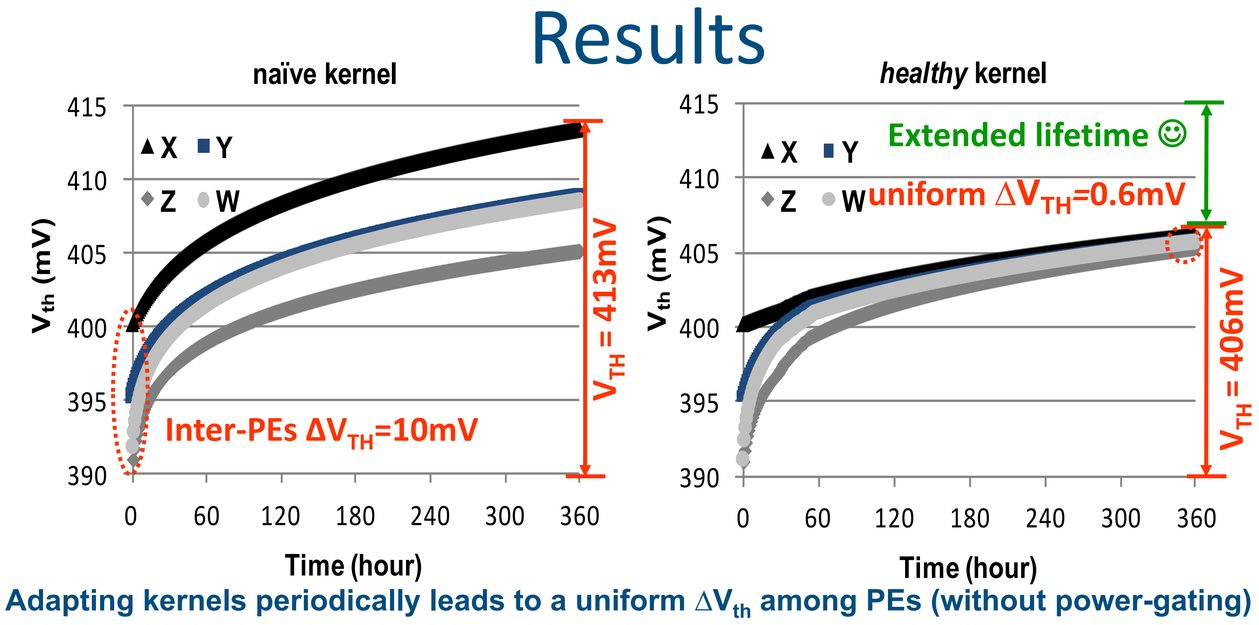
- Inter-Compute Units: Uniform ☺
- Inter-Stream Cores: SIMD fashion ☺
- Inter-Processing Elements: Uneven! ☹
- PE_X executes ~40% of instructions ☹
- Stress consumes timing margin
- The most aged PE limits lifetime

Aging-Aware Compilation

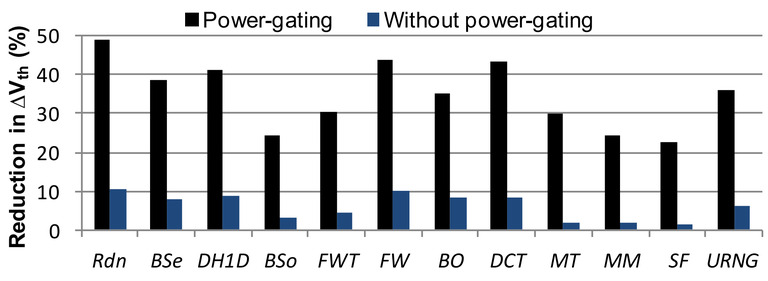
1. **Observability**: Reading NBTI sensors measurements
2. **Prediction:** Static code analysis technique estimates the percentage of instructions that will carry out on every PE (a linear calibration module later fits the predicted ΔV_{TH} shift to the observed ΔV_{TH} shift).
3. **Controllability:**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

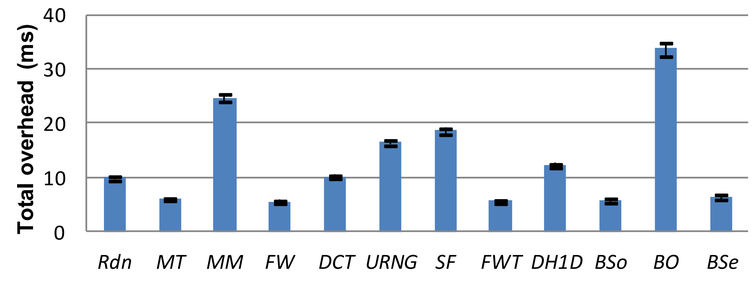


Results



Adapting kernels periodically leads to a uniform ΔV_{th} among PEs (without power-gating)





- * Imposes 0% throughput penalty (maintaining naïve ILP)
- * Reduces ΔVth: up to 49%(11%) and on average 34%(6%) in presence(absence) of power-gating supports compared to naïve 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 ΔV_{th} by up to 43% for register files.

NSF Expedition in Computing, Variability-Aware Software for Efficient Computing with Nanoscale Devices http://variability.org

UC San Diego · UCLA · Michigan · Stanford University · UC Irvine · Illinois