A Process and Environmental Variation Tolerance Scheme for ULP Shared-memory Processor Clusters

EU FP7 Project
PHIDIAS
(GA n. 318013)

Daniele Bortolotti, Andrea Bartolini, Luca Benini
MicrelLab (DEI) - University of Bologna - Italy

EU FP7 ERC Project
MULTITHERMAN
(GA n. 291125)

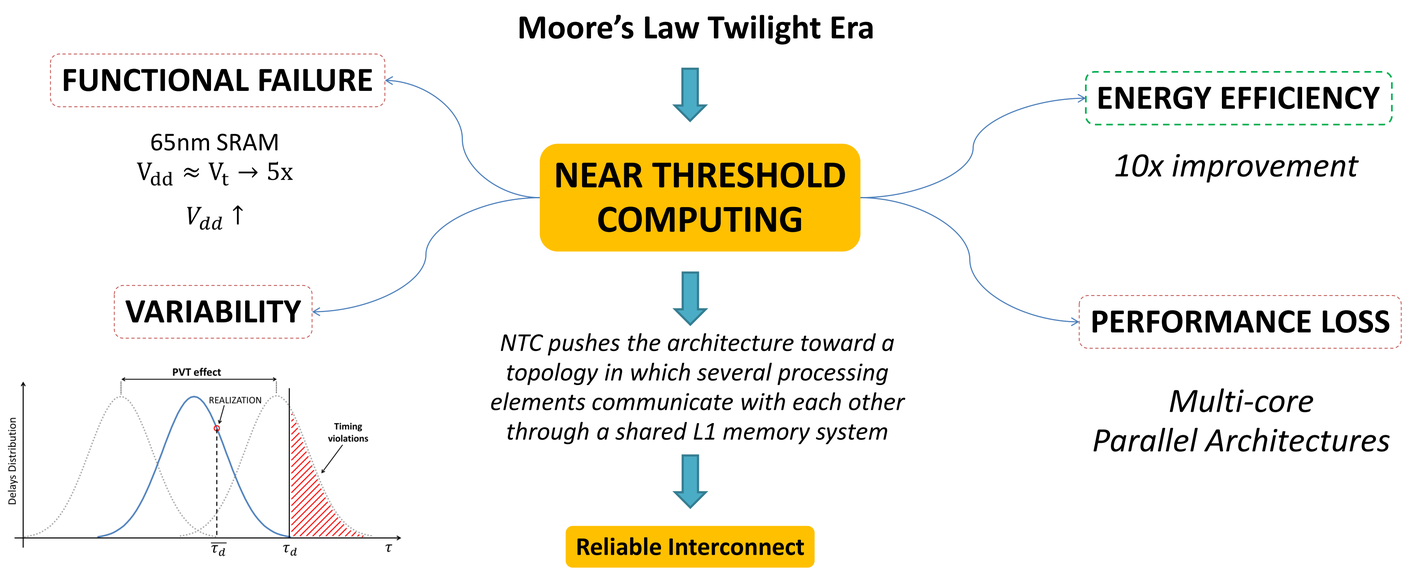
KEYWORDS

Near Threshold Computing (NTC) – Ultra-Low Power (ULP) – energy efficiency – many-core accelerators – tightly-coupled clusters
resilient interconnection network – dynamic variability tolerance – lightweight runtime solution – execution correctness

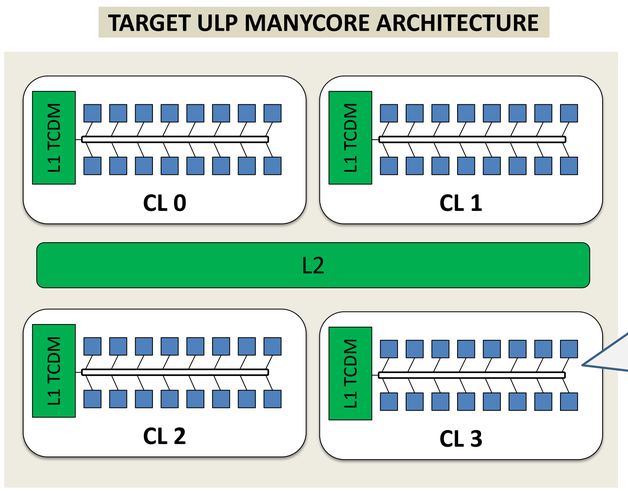
SCENARIO

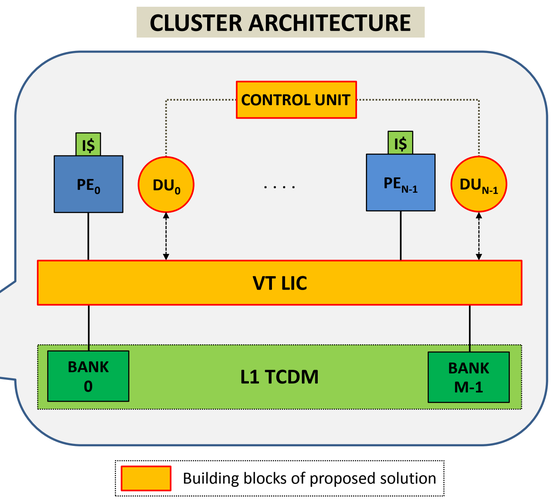
Near Threshold Operation is today a key research area in ultra-low power (ULP) computing, as it promises 10x improvement in energy efficiency compared to super-threshold operation, and it mitigates thermal bottlenecks. Unfortunately near-threshold operation is plagued by greatly increased sensitivity to threshold voltage variations, such as those caused by ambient temperature fluctuation. We focus on tightly-coupled ULP processor cluster architecture where a low latency, high bandwidth processor to memory interconnection network plays a key role.

We propose a lightweight runtime solution to tolerate ambient temperature induced variations by dynamically adapting the processor-to-L1-memory latency without compromising execution correctness



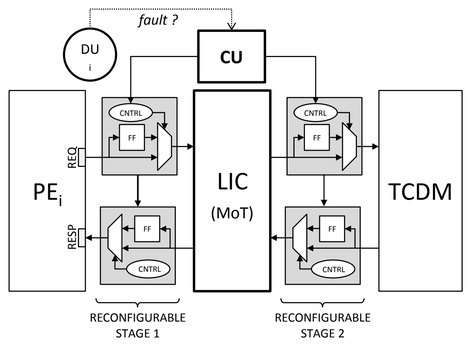
PROPOSED ARCHITECTURE

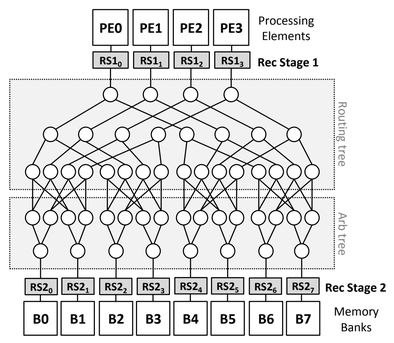




COMPONENTS

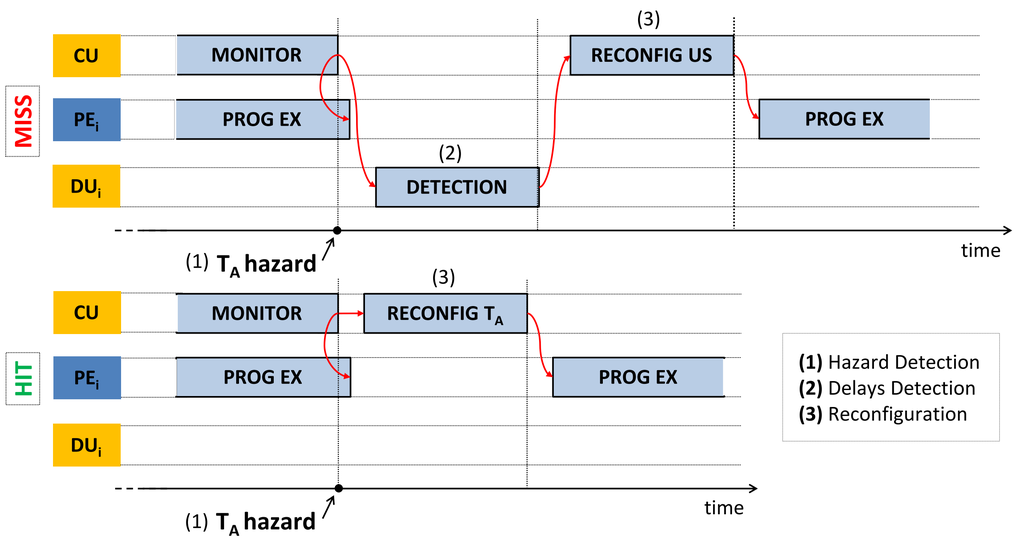
1. **Control Unit**
   - Reconfiguration Cache
   - Ultrasafe Configuration
   - Hazard Temperatures Array
   - Delay Table
2. **Detection Unit**
   - FSM for runtime DETECTION
3. **Variation Tolerant Interconnect**
   - Delay Insertions (up to 2-cycles)





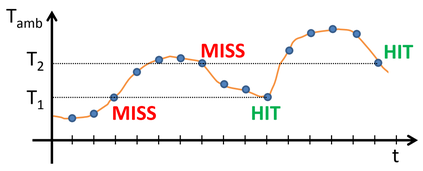
COMPONENTS INTERACTION

TIME DIAGRAM



(1) Hazard Detection
(2) Delays Detection
(3) Reconfiguration

TEMP PROFILE



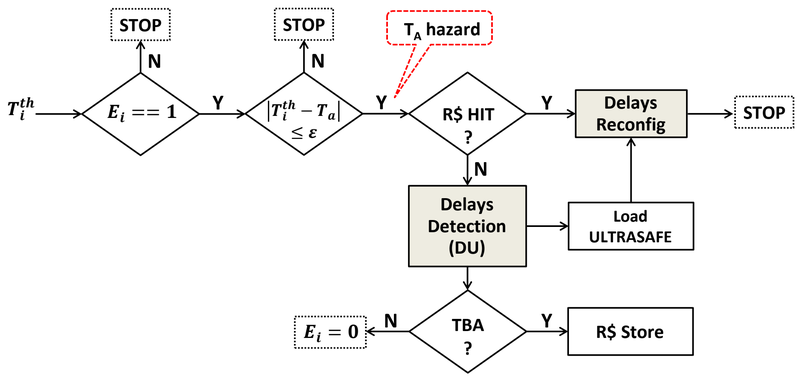
RCACHE enables a *learning procedure*

RCACHE

T°	$D_{0,0}$	$D_{0,1}$	...	$D_{N-1,M-1}$
T_1	00	10	...	10
T_2	01	11	...	00
...	...	...	...	...
$T_?$	ULTRASAFE			

ULTRASAFE configuration avoids timing errors

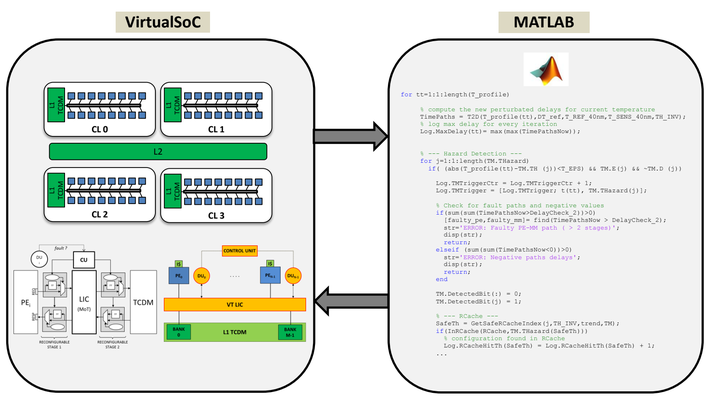
T MONITOR ALGORITHM



- hazard temperatures can be dynamically *disabled*
- *istheresys thresholding* avoids spurious detections

RESULTS

SIMULATION FRAMEWORK




```
for tt=1:1:length(T_profile)

    % compute the new perturbated delays for current temperature
    TimePaths = T2D(T_profile(tt),DT_ref,T_REF_40nm,T_SENS_40nm,TH_INV);
    % log max delay for every iteration
    Log.MaxDelay(tt)= max(max(TimePathsNow));

    % --- Hazard Detection ---
    for j=1:1:length(TM.THazard)
      if( (abs(T_profile(tt)-TM.TH (j))<T_EPS) && TM.E(j) && ~TM.D (j))

        Log.TMTriggerCtr = Log.TMTriggerCtr + 1;
        Log.TMTrigger = [Log.TMTrigger; t(tt), TM.THazard(j)];

        % Check for fault paths and negative values
        if(sum(sum(TimePathsNow>DelayCheck_2))>0)
          [faulty_pe,faulty_mm]= find(TimePathsNow > DelayCheck_2);
          str='ERROR: Faulty PE-MM path ( > 2 stages)';
          disp(str);
          return;
        elseif (sum(sum(TimePathsNow<0))>0)
          str='ERROR: Negative paths delays';
          disp(str)
          return;
        end

        TM.DetectedBit(:) = 0;
        TM.DetectedBit(j) = 1;

        % --- RCache ---
        SafeTh = GetSafeRCacheIndex(j,TH_INV,trend,TM);
        if(!InRCache(RCache,TM.THazard(SafeTh)))
          % configuration found in RCache
          Log.RCacheHitTh(SafeTh) = Log.RCacheHitTh(SafeTh) + 1;
        ...
```

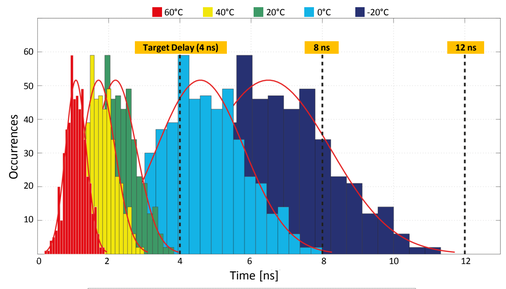
- SystemC-based simulator
- Architecture cycle-accurate models
- HW Detection and Reconfiguraton
- Short scale timing analysis

- Runtime algorithms
- Microbenchmarks
- D&R timing back-annotations
- Large scale timing analysis

http://www-micrel.deis.unibo.it/virtualsoc/

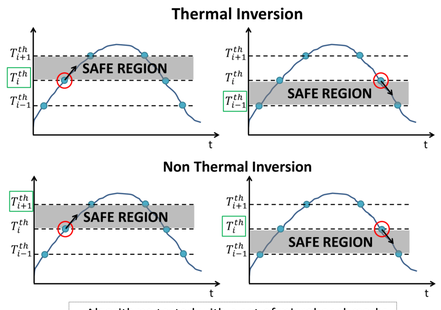
EXPERIMENTAL RESULTS

MODEL



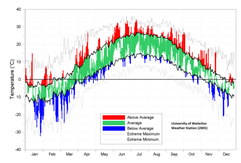
Temperature Effect on Delay Distribution

ALGORITHM DESIGN



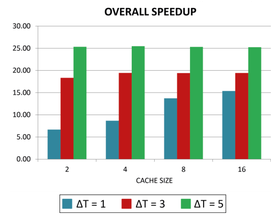
Algorithms tested with a set of microbenchmarks

FULL BENCHMARKS

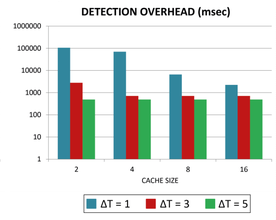


	FREQUENCY SCALING		DYN. TOLERANT	
MAX(τ_d) (ns)	FREQ (MHz)	OVERHEAD (%)	DELAYS	OVERHEAD (%)
4.00	250.00	0	0	0
4.66	214.51	16.55	19	1.83
5.49	181.99	37.37	79	4.52
6.33	158.04	58.19	179	8.17
7.49	133.45	87.34	298	12.39
8.83	113.30	120.65	404	14.73
9.66	103.53	141.47	464	17.01
10.49	95.32	162.29	519	20.10
11.99	83.40	**199.77**	618	**21.52**

OVERALL SPEEDUP

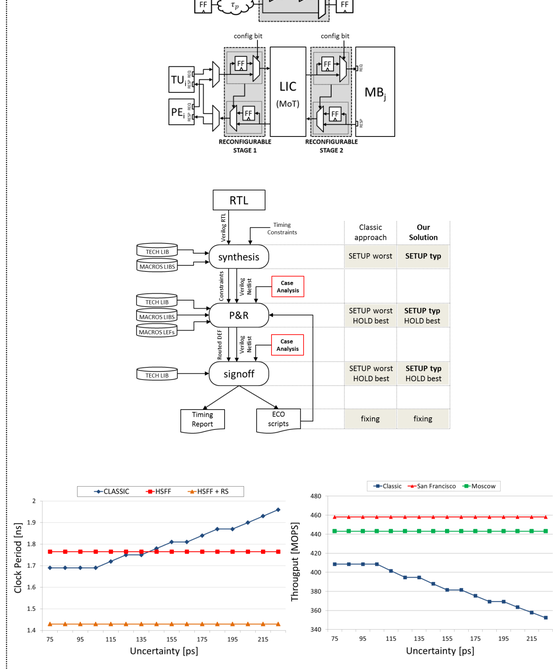


DETECTION OVERHEAD (msec)



MODULE	AREA (μm^2)	UNITS	OVERHEAD(%)
TU	2676	16	1.07 %
RECONF STAGE	1230	48	1.48 %
TM	49598	1	1.24 %
TOTAL	151435	-	3.79 %

EXTENSION



REFERENCES

[1] R.G. Dreslinski et al., "*Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits*", In: Proceedings of the IEEE, vol.98, no.2, pp.253-266, Feb. 2010

[2] Benini L. et al., "*P2012: Building an ecosystem for a scalable, modular and high-efficiency embedded computing accelerator*", In: Proc. of the ACM/IEEE DATE, 2012

[3] Kakoee M.R. et al., "*A resilient architecture for low latency communication in shared-L1 processor clusters*", In: Proc. of the ACM/IEEE DATE, 2012

[4] Dogan A.Y. et al., "*Power/Performance Exploration of Single-core and Multi-core Processor Approaches for Biomedical Signal Processing*", In: Proc. of PATMOS, 2011