JSC State Research Center of the Russian Federation
Concern CSRI Electropribor

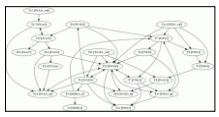
A real-time fault-tolerant and power-efficient multicore system on chip

Alexander M. Gruzlikov,
N.V. Kolesov, D.V. Kostygov and M.V. Tolmacheva

IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip
Nanyang Executive Centre, NTU, Singapore. October 4, 2019

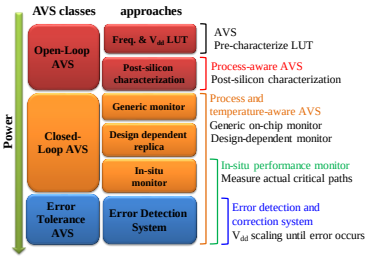
Outline

RT Multi core/Many-core System & Taxonomy of AVS Techniques (Background)



❑ Main features:

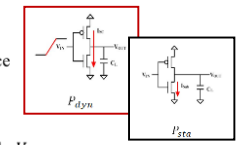
- periodical input data stream;
- parallel and asynchronous real-time (RT) computations;
- fixed list of tasks to be solved;
- high dimensionality;
- multiple causes of faults: (hardware and/or software failures, ...).

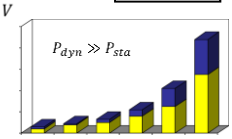


Tuck-Boon Chan and Andrew B. Kahng. 2012. Tunable sensors for process-aware voltage scaling. In Proceedings of the International Conference on Computer-Aided Design (ICCAD '12). ACM, New York, NY, USA, 7-14.

Power Model (Background)

- ✓ Total power consumption:
  $P_{dyn} + P_{sta}$
- ✓ Dynamic Power: current switching capacitance
  $P_{dyn} \sim V^2 * f * \#cores$
- ✓ Static Power: leakage current of device
  $P_{sta} \sim \#cores$
- ✓ Delay: the delay caused by the circuit at supply V
  $D \sim \frac{1}{V}$
- ✓ Criterion:
  $\min_A P_{dyn}(A)$





1. Preeti Ranjan Panda, B. V. N. Silpa, Aviral Shrivastava, and Krishnaiah Gummidipudi. 2010. Power-Efficient System Design (1st ed.). Springer Publishing Company, Incorporated.
2. Michael Keating, David Flynn, Rob Aitken, Alan Gibbons, and Kaijian Shi. 2007. Low Power Methodology Manual: For System-On-Chip Design. Springer Publishing Company, Incorporated.

Power Model. Example. (Background)

Example:

$$S_0 = (\#cores_0, V_0, f_0, P_0)$$

Assume that both the frequency and the supply voltage are reduced by a factor of k:

$$P_1 = \frac{P_0}{k^3}$$

And increase the number of cores by k times:

$$\#cores_2 = k * \#cores_0$$

$$P_2 = \frac{P_0}{k^2}$$

Distributed Diagnostics Model (Background)

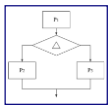
- ✓ The model of PMC represents a system consisting of modules capable each to use alone its functional and testing facilities for checking and verifying the technical state of some other modules. The system state as a whole is determined automatically by comparing the estimates obtained by the system modules themselves.
- ✓ Determination of the system state is called self-diagnosis, and the systems themselves are referred to as self-diagnosable.
- ✓ The greatest number of faulty modules existing simultaneously in the system is called the fault multiplicity denoted by t.
- ✓ System with fault multiplicity equal at most to $k \leq t$ is called the t-diagnosable system.

- ✓ PMC-model rules:
  - $\#cores \geq 2t + 1$;
  - there are no two cores testing each other;
  - each cores is tested by at least by t cores.

1. Preparata, F.P., Metze, G., and Chien, R.J., On Connection Assignment Problem of Diagnosable Systems, *IEEE Trans. El. Comput.*, 1967, vol. EC-16, No. 12, pp. 848–854
2. Y.K. Dimitriev, «Necessary and sufficient conditions for t-diagnosability of multiprocessor computer systems for various models of nonreliable testing established using the system graph-theoretical model» Automation and Remote Control, vol. 77, no. 6. pp. 1060-1070, Jun 2016.

Problem statement

Object: The system includes a set of identical cores interacting via shared memory. It is assumed that for each core, you can set the value of its clock frequency and voltage.



- example of the flow graph of a task.

In real-time system, this task performed with a period of information input. Assume that each operator $P_1 - P_3$ of the task is implemented by a separate software module and runs on a separate core. Thus, architectures A of a system consists of three cores.

Suppose the chip has additional cores that are not used.

Problem statement:

1. Determination of architecture is reduced to searching for such a redistribution of the computational load between all cores so that power P consumed by the system will be minimal: $A^* = arg\left(\min_A P(A)\right)$
2. Fault Tolerant Control: decentralized control.

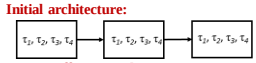
Algorithm 1. Determination of the energy-efficient architecture

Algorithm 1 Determination of the energy-efficient architecture

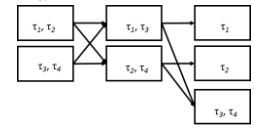
1: *Make initial assignments*:
$A = (1\ 1\ \dots\ 1)$
$\overline{P} = (\overline{P}_1\ \overline{P}_2\ \dots\ \overline{P}_n)$.
2: **for** $M = n_d$ to 0 **do**
3: In $\overline{P} = (\overline{P}_1\ \overline{P}_2\ \dots\ \overline{P}_n)$, choose the component with the maximum value of $\overline{P}_{max}$. Let its number be l.
4: Enter an additional core into the l-th stage. Perform approximately balanced redistribution of the load between the cores of the l-th stage.
5: Recalculate the algorithm parameters: $\overline{P}$, $a_l = a_l + 1$
6: **end for**

$A = (a_1\ a_2\ \dots a_n)$ – system architecture vector;
a_i - the number of cores in the split set of the i-th core;
$\overline{P} = (\overline{P}_1\ \overline{P}_2 \dots \overline{P}_n)$ – the vector of unit power consumption;
$\overline{P}_i$ - unit power consumption of the split set of the i-th core.

Initial architecture:



Energy-efficient architecture:



Analysis of the algorithm for determining energy-efficient architecture

Lemma 1. If we arrange the system stages in descending order of power consumption, then the corresponding optimal sequence, composed of per-stage powers of the sets of the cores used, will be nonincreasing.

This statement confirms the fact that the proposed algorithm is aimed at unloading the most loaded cores.

Theorem 1. Algorithm 1 is «greedy», that is, each of its steps is optimal by criterion:

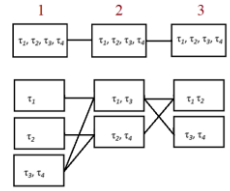
$$A^* = arg\left(\min_A P(A)\right)$$

Theorem 2. Algorithm 1 provides the architecture which is optimal by criterion A^* for a given limitation on the number of additional cores $n_d \le n_{d0}$.

Assume that a system contains three cores on which four task are executed …

Example.

Tasks	$\tau_{1,1}$	$\tau_{1,2}$	$\tau_{1,3}$
τ_1	3	2	1
τ_2	3	1	1
τ_3	3	1	1
τ_4	2	1	1
τ_5	11	5	4
$\bar{P}$	22	10	8



$\#cores_d = 4$

1. Do initial assignments: $M = \#cores_d$, $A = (1\ 1\ 1)$, $\bar{P} = (22, 10, 8)$.
2. Split the core of the 1-stage by adding two cores. Distribute the computational load approximately uniformly between the three cores: {5, 3, 3}. Calculate the average power consumed by the cores of the split set of 1-stage: $\bar{P}'_1 = 2.1 < 10$. $\#cores_d = 2$. Next step of splitting.
3. Split the core of the 2-stage by adding an additional core. Distribute the computing load approximately uniformly between the two cores: {3, 2}. Calculate the average load of the cores of the split set of stage 2: $\bar{P}'_2 = 2 < 8$. $\#cores_d = 1$. Next step of splitting.
4. Split the core of the 3-stage by adding an additional core. Distribute the computing load uniformly between the two cores: {2, 2}. Calculate the average load of the cores of the split set of stage 2: $\bar{P}'_3 = 1$. $\#cores_d = 0$. End.

$$k_1 = \frac{\psi^1}{\psi^1_{max}} = \frac{\tau_{5,1}}{\tau_{5,1} - \tau_{4,1}} = 2{,}2 \quad \bar{P}'_1 = \frac{\bar{P}_1}{k_1^2} = 2{,}1$$

$$k_2 = \frac{\psi^2}{\psi^2_{max}} = \frac{\tau_{5,2}}{\tau_{1,2} + \tau_{2,2}} = 1{,}7 \quad \bar{P}'_2 = \frac{\bar{P}_2}{k_2^2} = 2 < 8$$

$$k_3 = \frac{\psi^3}{\psi^3_{max}} = \frac{\tau_{5,3}}{\tau_{1,3} + \tau_{2,3}} = 2 \quad \bar{P}'_3 = \frac{\bar{P}_3}{k_3^2} = 1$$

Example. Results.

The resulting system contains seven cores and is characterized by the following vector of average computing loads for the stages $\bar{P} = (2.1\ 2\ 1)$. Let us estimate approximately the reduction in power consumption obtained. As an estimate of the power consumed by the original system (in conventional units):

$$P_0 = \sum_j \bar{P}_{\Sigma,j} = 40$$

In this case, for the transformed system, the equation takes the form:

$$P = \sum_j \frac{\bar{P}_{\Sigma,j}}{(k_j)^2} = 12.3$$

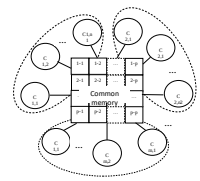
where k_j is the frequency reduction coefficient (supply voltage).

Then, the power reduction is given as:

$$\frac{P_0}{P} = 3.25$$

Design of a decentralized energy-efficient fault-tolerant multicore system on a chip

1. Introduction of redundancy: splitting a core.
2. Finding a failed core : distributed diagnostics (PMC-model).
3. Failure detection decision-making: decentralized procedure.
4. Failure recovery: core fusion.



Sufficient conditions for t-diagnosability (localization of t failed cores)

PMC-model:

1. system of n cores are valid: $n \geq 2t + 1$;
2. there are no two cores testing each other;
3. each core is tested by at least by t cores.

There are two most common approaches:

1. **Case of $t = 1$.** It assumes the allocation of the Hamiltonian cycle in the graph of links of the system (the cycle passing through all the nodes (cores) of the system; in so doing, each vertex is passed no more than once). Further, each core tests only one core of the system, namely, the one following it in the Hamiltonian cycle. Clearly, this diagnostic experiment satisfies the above sufficient conditions.
2. **Case of $t > 1$.** The Hamiltonian cycle is not allocated, and each core tests all of its immediate neighbors. It is obvious that the number of tests depends on the number of neighbors, i.e., communication geometry implemented in the communication system.

Example

Consider the experiment that uses a Hamiltonian cycle in a three-core system:

Failure	Tests		
	$1 \rightarrow 2$	$2 \rightarrow 3$	$3 \rightarrow 1$
1	x	0	1
2	1	x	0
3	0	1	x

Table shows the syndromes of tests. The table rows are matched with core failures, and the columns, with checks. «0» means a positive result of the test, «1» - a negative result, «x» - an undefined result. It can be seen that in any variant of its definition, the rows of the table will not coincide, and, therefore, **the failures will be distinguishable**.

Let us analyze the requirements for a communication system. The shared memory of the communication system is divided into areas that correspond to informational links of specific core pairs. It is clear that, on the one hand, the number of such links should be reduced in order to reduce the area of the chip occupied by the common memory. However, on the other hand, the number and geometry of these links affect the effectiveness of the diagnostic experiment.

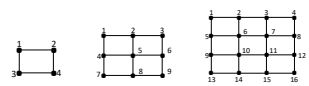
Characteristics of communication system

The applied communication system must not only admit of the organization of the accepted diagnostic experiment in the system, but in addition, it should not make it too long, remaining within the prescribed limitations on the chip area. To design a communication system, it is necessary to know the values of its relevant characteristics.

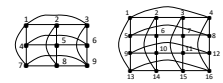
Simulation in the YACSIM environment was carried with the aim to obtain such characteristics.

Three types of communication systems were studied:

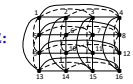
GRIDS:



TORUS:

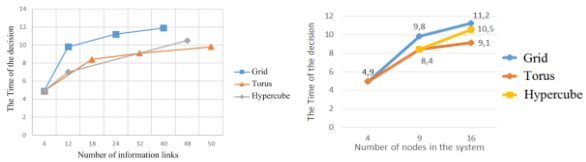


HYPERCUBE:



The aim of the simulation was to determine the duration of the diagnostic experiment, including mutual intercore tests and accumulation of all the results in each core. The simulation was carried out under the assumption that the experiment is carried out with testing of neighboring cores, wherein the time needed for one core to test another one was taken to be 2.1 conventional units, and the time of diagnostic information transmission from the core input to its output, 1.4 conventional units.

Design a communication system



The diagnostic experiment has shown that the torus-type configuration is the most effective communication system from the standpoint of decision-making.

Conclusions

An approach to designing a decentralized fault-tolerant and energy-efficient system on a multicore chip has been proposed. The approach consists in determination of an energy-efficient architecture, which is made possible due to the introduction of additional cores into the system, resulting in a decrease in the clock frequency and supply voltage, and the development of procedures for the system diagnosis and reconfiguration.

JSC State Research Center of the Russian Federation
Concern CSRI Electropribor

Thank you for your attention!

Contacts:
E-mail: agruzlikov@yandex.ru