Redefining the term 'Reconfigurable-Chips': *Unsolving* the Rubik's Cube

Nachiappan Chidambaram N., Mahmut T. Kandemir, Chita R. Das
[nachi, kandemir, das] @ cse.psu.edu
The Pennsylvania State University, PA, USA 16802

Nature! The perfect guide.

Engineers, scientists and particularly we -- the architects -- have much to learn from the nature. Nature's creation of complex living beings from single-celled amoeba has been possible due to its three vital capabilities: evolution, learning, and self-repair. Each one of these capabilities inspires us to build *adaptable* and *efficient* systems for the future.

Taking inspiration from the brain, where dynamic reconfiguration of neuron inter-connectivity occurs continually during learning [1], the field of re-configurable computing emerged a few decades back. These systems have the ability to adapt and make substantial changes to the data path and control flow on the re-configurable fabric. But, these systems, in the true sense, do not physically displace the components on the chip to evolve into a more intelligent system. In this abstract, we break free from conventional thinking and propose a thought-provoking idea with a realistic model to design a *true re-configurable chip*. This chip is designed to adapt to a wide dimension of an application mix's needs. Moreover, we strike a communication-centric approach, rather than a traditional computation-centric one.

Proposal.

We propose a computer system design, where the processor is built in a 3D nxnxn Rubik's cube form of organization. Each face of the cube initially consists of a type of system component, which could either be big cores (high performance cores), small cores (wimpy cores), GPU cores, accelerators, caches/eDRAM/emerging memory technologies, DRAM banks, or any other component not yet invented. A sample 3x3x3 processor design[1] is shown in Figure 1. With such system designs, we can achieve quick and easy system reconfiguration by just rotating the layers of the cube. The Rubik's cube has to be 'un'-solved (by a mechanical controller placed at the center) in such a way that it places components (tiles) in optimal positions based on application requirements. For example, consider a computationally intensive single-threaded application that has large working data set with no requirement of GPU cores for its execution. Such an application would not benefit when many cores and GPUs are clustered together, with shared caches and DRAM banks located farther away. Instead, if the Rubik's cube is manipulated to have one big core tile surrounded by many caches and DRAM bank tiles in its neighborhood, then amount of data physically traveling on the fabric can be substantially reduced (design shown in Figure 2). This will, in turn, reduce all communication latencies and save significant amount of energy consumed by the application.

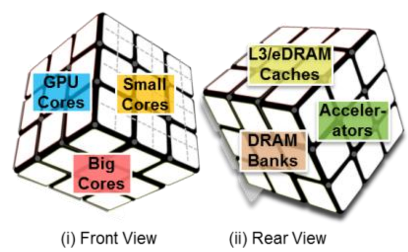

Figure 1: Processor designed as a 3x3x3 Rubik's cube. nxnxn designs can also be formed in similar fashion.

To assist execution of multiple applications, if the layer rotations are done intelligently, one can manage to achieve an optimal trade-off between tile placements and different applications' requirements.

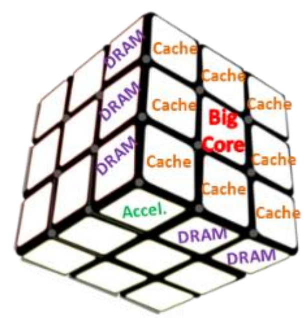

Figure 2: Example showing a reconfigured core based on an application's requirement.

Architecting a Rubik's Cube.

Understanding this system design involves understanding a few interesting properties of a Rubik's cube. First, the center tile of each face is unmovable. However, it can rotate (along with its closest neighbor cubes). Second, other tiles can move anywhere on the system to relocate, except for the center of edge tiles. They can move only to other center of edge positions; they cannot become corner tiles. Third, all the edge and corner tiles are physically attached to one and two tiles respectively, and cannot be separated. The system is designed with these constraints in regard.

[1] Even 4 or 5 sided Rubik's structures can be used at the cost of increased design complexity.

Organization. At the center of the Rubik's cube is the *core controller,* that controls when and how the layers are rotated to achieve a required configuration. The core controller is a mechanical device with a simple microcontroller within it that acts based on inputs from the master big core (the physically attached big core). It can *rotate* a face of the cube as depicted with arrows in the Figure 3, or it can *swivel* a layer horizontally or vertically to mix the tiles (*unsolve*). The bottom face's center tile is removed to make way for the base connector that interfaces the Rubik's cube with power or an external hub to interface other devices. (Implementation details regarding mechanical arm movement around core controller have been omitted for brevity.)

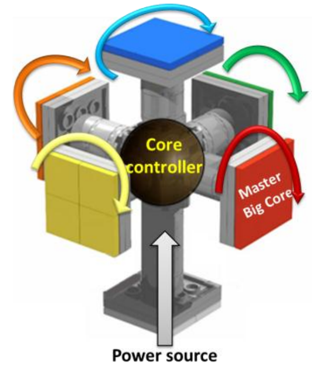

Figure 3: Revealing the core controller, power source and the physically attached tiles.

Power. The system would receive power through the core controller. From there, power has to be distributed to the tiles in the cube. Unfortunately, in such a physically re-configuring system, the power cannot be supplied through electrical wires for obvious reasons. Instead, wireless inductive electrical power transfer [2] is used to power each tile. Such technology is already used in many electronic systems now, e.g. Nokia Lumia 920 [3]. Wireless power consortium has established interface standards and companies like Qi, WiPower and WiTricity already offer such products that can supply power in a wireless fashion to run electronic components or charge batteries. The consortium that aims at making wireless electricity ubiquitous for the future has more than 100 top companies (including Samsung, LG, Haier, Sony and Verizon) joining hands.

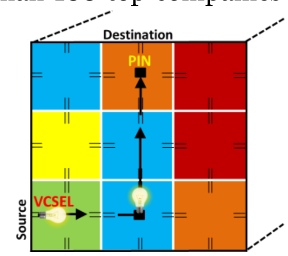

Figure 4: Optical inter-connection approach used to communicate across cores.

Interconnection. While reconfiguring the system helps to minimize the number of hops in the network, if our interconnection network design is adept at transferring data at high speed and high bandwidth, the data transfer latency would be negligible. With this aim, we design a tile-to-tile interconnect using an optical approach. The fabric will have a low-cost CMOS-based optical interconnect that can be integrated into a microprocessor substrate [4]. Each cube is equipped with vertical-cavity surface-emitting lasers (VCSELs), positive-intrinsic-negative (PIN) photo detectors and polymer wave-guides. The VCSELs are placed at the sides of the tiles, while the PINs are at the center as shown in the Figure 4. Similar network designs have been proposed and implemented by Intel, IBM and many others [5].

Double your gains. When using a 3x3x3 Rubik's cube (with around 43 quintillion[2] possible configurations!), the system would re-align itself to one configurations based on the application-mix's demands. In a datacenter, when multiple applications from many users are running concurrently, choosing an *ideal* configuration for all applications would not only benefit the vendors, by conserving energy and increasing utilization, but also the users by decreasing their application runtime through reduction in communication latency.

Why Rubik's cube? Many current designs are only 2D, or pseudo-3D in nature. Only a true 3D system organization would aid physical reconfiguration of components. While many doubt that a Rubik's cube design would limit the degrees of freedom, it would not because the number of possible combinations is huge, and one of them would likely achieve a near optimal configuration. Many properties[3] of a Rubik's design would enable natural, easy and quick plug-and-play style of re-configurable design. Is it not an apt fit for a physically re-configurable architecture design?

Impact for architects. *Finally,* architects would *consider* moving away from running simulations to solving Rubik's cube. Wouldn't that be fun for us?

References

[1] Danielle S. Bassett, et. al," Dynamic reconfiguration of human brain networks during learning", PNAS 2011.
[2] Komerath, N.; Chowdhary, G.; , "Retail Beamed Power for a Micro Renewable Energy Architecture: Survey," International Symposium on Electronic System Design (ISED), 2010.
[3] http://theweek.com, Sep4, "The Nokia Lumia 920's wireless charger: A revolution for smartphones?"
[4]Sahai, A, Graham, D, "Optical wireless power transmission at long wavelengths," International Conference on Space Optical Systems and Applications, 2011.
[5] Intel, Optical Interconnect Built Inside Silicon Chips.

[2] Quintillion = 10^{18}. For a 4x4x4 Rubik's cube, the number of configurations tends to $7 * 10^{45}$!

[3] Any required configuration can be achieved in a maximum on 20 rotations for a 3x3x3 Rubik's cube.
Figure on the right shows how the layers in Rubik's cube can be rotated or swiveled to achieve a target configuration.