**Parallel Computing of Graph-based Functions in Re-RAM**

Because of the continual reduction in feature size, CMOS is nearing its physical limits, necessitating the hunt for feasible successor technologies beyond the scaling limit. ReRAM is a nonvolatile memory technology with low power consumption, built-in computation, and high logic synthesis efficiency.

Instead of logic primitives, the Multiply-Accumulate (MAC) operation is used in the Binary Decision Diagram (BDD) approach. The BDD nodes are immediately assigned to parallel MAC processes.

The And-Inverter Graph (AIG) is a computer architecture-based automated compilation technique. Converting any Boolean function is possible.

The edges of graph-based representations illustrate wires linking two-input AND gates that correspond to nodes, which can be modified to display inverters between nodes.

ReRAM MAC Computation is used to run several MAC procedures concurrently.

MIGs are the most advanced graph structure for ReRAM-based synthesis in Graph Based Computation since they need the fewest operations and devices.

A calculation is wordline parallel if it uses one wordline and multiple bitlines; bitline parallel if it uses one bitline and multiple wordlines; and mixed parallel if it utilises both wordlines and bitlines in parallel.

Each node in BDD-based Parallel Computation must be implemented as a 2x1 multiplexer.

All offspring of both nodes must be calculated in AIG-based Parallel Computation. There must be no data dependencies between the nodes, and they must share a wordline operand with host devices on the same wordline, with the content utilised for no other computations.

Each node in M-AIG-based Parallel Computation represents an m-Input And Gate or PI; each input edge can be connected to the constant 1 or to a child node, and if the input edge is connected to a node, it can be complemented to signify inversion.

The proposed technique drastically decreases the number of devices and procedures required. On average, it decreases the number of operations by 66%. In all areas and operations, BDD and AIG thrive. For lower values of m, m-AIG has outperformed AIG.

**Power Aware Computing**

Power, energy, and temperature all have an impact on processor design. CPU clock frequency stagnation and emphasis on parallelism will improve energy efficiency in the future. Software design, in addition to hardware, may have a substantial influence. The ability to assess power and energy usage is a must-have function. In this experiment, the PAPI library was employed, which provides broad and portable access to hardware counters connected to the CPU and other components. In the trials, the Xeon Phi Knights Landing (KNL) CPU architecture was employed. The study made use of Dense Linear Algebra (DLA) kernels (BLAS kernels).

Kernels commonly used in high-performance computing applications were chosen to analyse and evaluate the impact of power consumption and energy demands. Level 1 is concerned with scalar and vector operations, Level 2 with matrix-vector operations, and Level 3 with matrix-matrix operations. Memory Bond Levels 1 and 2 are in the Memory Bond Class, whereas Compute Intensive Level 3 is in the Compute Intensive Class.

The compute costly technique dgemm and the memory bound class dgemv were examined and analysed.

PAPI is a library that uniformly gathers performance counter data from various hardware and software components. PAPI is composed of numerous components that allow for the monitoring of power consumption and use via various interfaces.

The FLAT mode, according to dgemm Kernel behaviour, uses the MCDRAM as physical addressable memory space rather than a cache.

The performance of the dgemv Kernel reduces between the two storages, with the exception of DDR4, which performs four times slower than MCDRAM, and the results are the same as in Hybrid Mode.

According to the findings of this study, employing high bandwidth MCDRAM on KNL is critical for high efficiency and low power consumption, and Hybrid mode is the best option if the application demands a lot of processing.

**Temperature-Aware Computer Systems Opportunities and Challenges**

It has not been sufficient to build with power in mind to halt the flow of problems such as heat density. Localised heating occurs at a significantly quicker pace than chip-wide heating. Even in the worst-case scenario, most high-power applications are still 20% or more less expensive.

The need for Architectural-Level Thermal Management originates from the fact that the architectural domain of each computing system is unique, and workload creation is necessary to regulate instruction level parallelism. Hot spots and temperature gradients are mentioned in the computer system design handbook. The function of system design and operating system is critical in this case.

Thermal modelling is required at the design stage to avoid thermally induced temporal and spatial nonuniformities in computation.

A computing system's compact model of a parametric microarchitecture must track temperatures at the microarchitectural unit level, be modelled in such a way that a new compact model for different microarchitectures can be created, solve the RC circuit's differential equations quickly, and be boundary and initial-condition independent.

Temperature vs average power density for gcc with a power averaging time of 0.033 seconds is plotted using temperature-tracking Dynamic Frequency Scaling.

Because CMOS carrier mobility is temperature dependent, the frequency of Temperature-tracking Dynamic Frequency Scaling is also linearly dependent on the operating temperature. When an application's temperature limit is reached, it may simply modify the frequency to compensate for the greater temperature.

Dynamic Voltage Scaling is a thermal management technique. As the operating voltage approaches the threshold voltage, circuits switch more slowly, hence reducing the processor voltage must be accompanied by a drop in frequency.

According to the findings, Migrating Computation is the best DTM technique at 0.8 K/W because the floorplan alone is enough to reduce the operating temperature of the primary integer register file, MC can use ILP to conceal the extra latency of the spare register file, and total abolition of activity in the primary register file enables it to cool rapidly, lowering the use of the slower secondary register file.