# AnyCore-1: A Comprehensively Adaptive 4-way Superscalar Core

Rangeen Basu Roy Chowdhury, Anil Kumar Kannepalli, and Eric Rotenberg
North Carolina State University, Raleigh, NC, 27695, USA

With the end of Dennard scaling, processor architectures are becoming increasingly heterogeneous to eke out the most performance and energy efficiency from silicon. Adaptive superscalar cores have the ability to dynamically adjust their execution resources to match the instruction-level parallelism (ILP) of different program phases. The goal of adaptivity is to maximize performance in as energy-efficient a manner as possible. This is achieved by disabling execution resources that contribute only marginally to performance for the code at hand. Researchers have proposed various adaptive features, including size-adjustable structures (reorder buffer, physical register file, issue queue, load and store queues, caches, etc.), superscalar width (number of pipeline ways), and pipeline depth. The benefits of adaptivity are eroded by its circuit-level overheads. Unfortunately, circuit-level overheads cannot be effectively estimated or appreciated without a hardware design. To this end, we developed a register-transfer-level (RTL) design of a comprehensively adaptive superscalar core, called **AnyCore**. AnyCore RTL is the centerpiece of the overall **AnyCore Toolset** [1] which enables computer architects to explore, and even fabricate, adaptive superscalar cores. The toolset also includes flows for synthesis, physical design, and post-synthesis/post-layout power estimation. The RTL and low-level design flows support clock-gating and power-gating of deconfigured lanes and structure partitions, to further reduce dynamic and static power.

We used the AnyCore toolset to fabricate the **AnyCore-1 chip** in a IBM 130 nm process. To our knowledge, AnyCore-1 is the first prototype of a width and size adaptive superscalar core (Table-1). The chip implements a 4-way adaptive superscalar core with a maximum fetch width of 4 and a maximum issue width of 5. The core includes L1 instruction and data caches. All memories, including the L1 caches and branch predictor structures, are synthesized (standard-cell based RAMs). Besides the CPU core, the chip also includes serializers/deserializers for off-chip communication, performance counters, reconfiguration logic, and debug logic. The chip implements clock-gating of deconfigured lanes and structure partitions. The L1 caches are not clock-gated. We did not implement power-gating as static power is negligible in the older foundry process. Moreover, the older standard cell library does not have the cells required for power-gating. The chip is 25 mm$^2$ and has 100 pads, 79 of which are signal pads. The 79 signal pads primarily consist of dedicated instruction and data buses (for cache miss handling), a debug bus for directly interacting with the chip, such as configuring the adaptive core, directly reading/writing the caches, *etc.*, and clock, reset, and assorted control signals. The layout and other physical design information are shown in Figure-1.

The chip is packaged in a CQFP-100 package (Figure-2) that has 100 pins, 25 on each side. The package is housed in a compatible 100-pin socket on a custom-designed 4-layer PCB. The package and socket can be seen in the top-side image of the PCB (Figure-1). The bottom-side, not shown, has a standard FMC LPC connector and other passive components (decoupling capacitors, power measurement resistor). The LPC connector mates the PCB to an FPGA board (we currently use the Xilinx ML605). The FPGA is programmed with a testbench that services cache miss requests, acts as a bridge for the debug bus, and manages dynamic reconfigurations of the CPU core. A software program on a host PC manages the FPGA testbench and the AnyCore-1 chip. It uses a USB-UART bridge to communicate to the FPGA and the chip. This software is used by the user to load benchmarks to the testbench, configure the chip, and read performance counters from the chip. Dynamic reconfiguration of the CPU core takes about 50 cycles and can be triggered in three ways: (1) A special instruction embedded in the benchmark, (2) a scheduler in the host software that responds to bottlenecks in the core (as gauged by performance counters), and (3) manually by the user.

Our experiments with the chip so far involved testing the functionality of the chip and measuring its power consumption for various configurations of the CPU core. While AnyCore-1 was designed for a max frequency of 66 MHz, we run it at 10 MHz for our testing. Running it slower only impacts the absolute values of power and does not change the relative trends. We used handcrafted microbenchmarks (Table-2) for our tests. The results show that power consumption and IPC scale with the configured core size (Figure-3). The power delta between the smallest and biggest configurations is 11 mW. Power consumption in the smallest configuration is 25 mW. The inflated baseline power is primarily due to the fully-synthesized, always-clocked caches. This is evident from the idle power measurements shown in Table-3. The four measurements cover the smallest and biggest configurations with clock disabled and clock enabled. The CPU pipeline is explicitly stalled (fetch gated) in all cases. With the clock disabled, power is near-zero, which shows that the leakage power is negligible. With the clock enabled, power is 21 mW and 25 mW for the smallest and biggest configurations, respectively. This is solely clock tree power as the CPU pipeline is stalled. Although we cannot precisely quantify the contribution of the always-clocked caches (because we cannot clock-gate the minimal pipeline configuration), a practical estimate is about 20 mW.

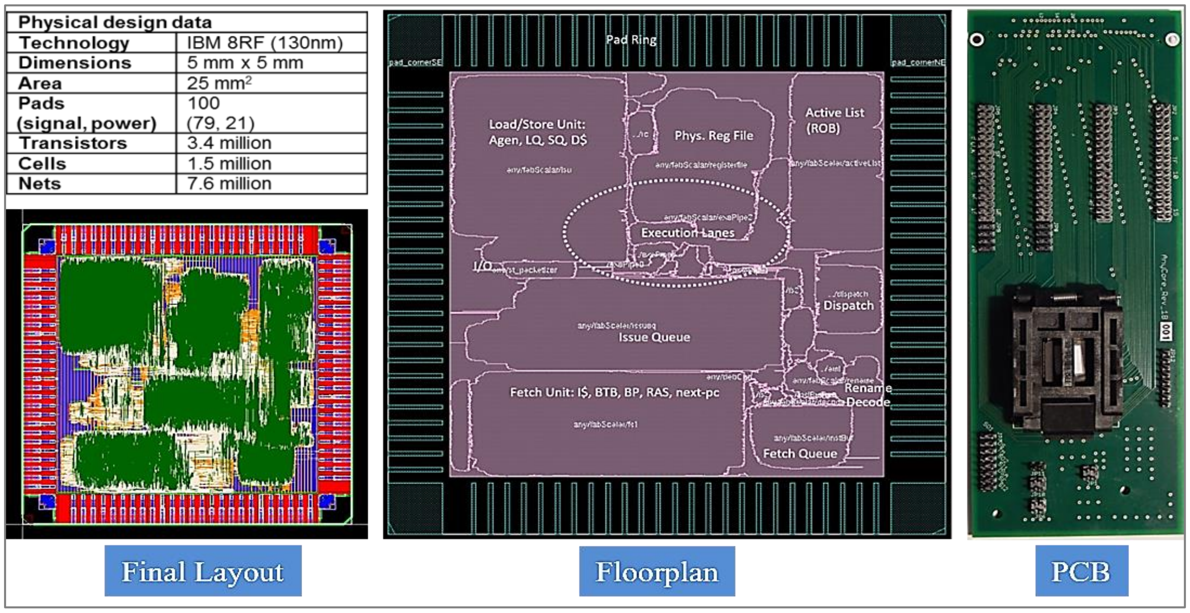
Figure-4 shows the results for a microbenchmark with two distinct loops having different amounts of ILP. Tests were run with and without dynamic reconfiguration enabled. In tests with dynamic reconfiguration, configuration for a code region was chosen oracularly based on *a priori* trials on 10 different configurations. As seen from the figure, dynamic reconfiguration runs lie on the pareto frontier and provide a tradeoff between energy and delay, which the scheduling mechanism can exploit as per its goal. Future work with the chip will involve exploring scheduling mechanisms to extract maximum benefit out of the available adaptivity and testing these mechanisms on SPECINT SimPoints.

| Parameter | Max Size | Legal Configs |
|---|---|---|
| Fetch Width | 4 | 1, 2, 3, 4 |
| Issue Width | 5 | 3, 4, 5 |
| Issue Queue | 64 | 16, 32, 48, 64 |
| Load/Store Queues | 32 | 16, 32 |
| Phys. Register File | 128 | 64, 96, 128 |
| Reorder Buffer | 128 | 64, 96, 128 |

**Table-1:** Max sizes and allowed configurations for key parameters of the core.

| Benchmark | Dynamic Insts. (million) | Avg. IPC | Avg. Power (mW) |
|---|---|---|---|
| Reduce an array to a sum | 235.46 | 3.05 | 36.25 |
| Bubble Sort an array | 36.92 | 0.32 | 27.50 |
| Prime Number Generator | 88.28 | 1.52 | 30.63 |
| Linear Feedback Shift Reg | 67.11 | 1.57 | 30.63 |
| Sum first N real numbers | 67.11 | 4.00 | 36.25 |

**Table-2:** Descriptions of microbenchmarks, their IPC on the biggest configuration, and their average power on the biggest configuration.



| Physical design data | |
|---|---|
| Technology | IBM 8RF (130nm) |
| Dimensions | 5 mm x 5 mm |
| Area | 25 mm$^2$ |
| Pads (signal, power) | 100 (79, 21) |
| Transistors | 3.4 million |
| Cells | 1.5 million |
| Nets | 7.6 million |

**Figure-1:** The left-most picture and table show the layout and design data for AnyCore-1. The middle picture shows the floorplan of the core: pipeline stages and components are marked. The right-most picture shows the top view of the custom 4-layer PCB that we use for testing the chip. The AnyCore-1 test chip is inside the socket.

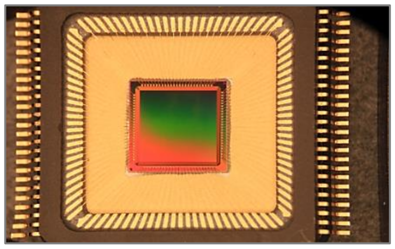


**Figure-2:** AnyCore-1 die (colorful square in the center) bonded to a CQFP-100 package.

| Clock | Configuration | Power (mW) |
|---|---|---|
| **Disabled** | Smallest | 0.01 |
| **Disabled** | Biggest | 0.01 |
| **Enabled** | Smallest | 21 |
| **Enabled** | Biggest | 25 |

**Table-3:** Idle power of AnyCore-1 chip.

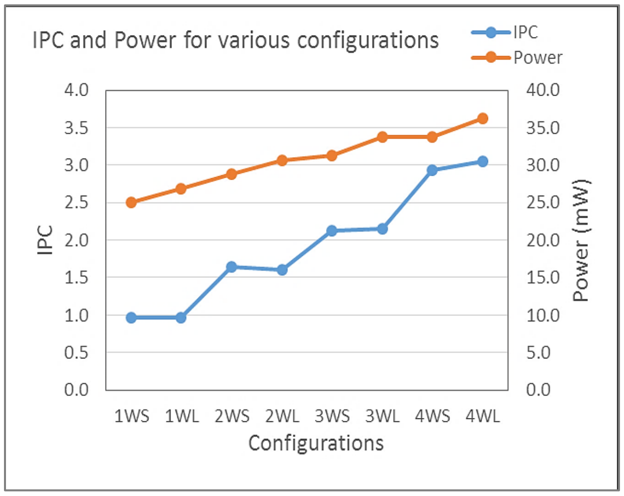


**Figure-3:** IPC and power of the **array reduction** microbenchmark for 8 out of the hundreds of possible configurations. These configurations represent 4 different pipeline widths and 2 sets of structure sizes for each width. For example, 1WS stands for 1-wide pipeline with small structures, whereas, 1WL stands for 1-wide pipeline with large structures. Similarly, 4WS and 4WL stand for 4-wide-small and 4-wide-large, respectively. 1WS is the smallest possible configuration and 4WL is biggest possible configuration.

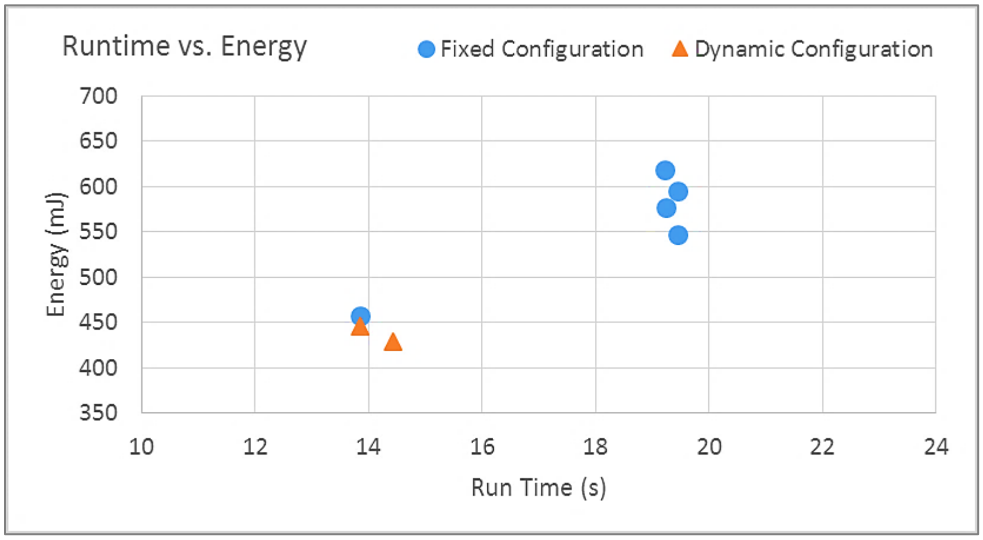


**Figure-4:** Run Time vs. Energy plot for a microbenchmark with two loops having different amounts of ILP. The blue circles represent fixed configurations wherein one configuration was used for the entire benchmark run. The orange triangles represent two different runs with dynamic reconfiguration enabled. The first run does not sacrifice performance to save energy. The second run sacrifices up to 10% performance to reduce the total energy consumption.

## References

[1] R. Basu Roy Chowdhury, A. K. Kannepalli, S. Ku, and E. Rotenberg. AnyCore: A Synthesizable RTL Model for Exploring and Fabricating Adaptive Superscalar Cores. Proceedings of the *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS'16)*, pp. 214-224, April 2016.