20.2 A Single-Chip Programmable Platform Based on a Multithreaded Processor and Configurable Logic Clusters

Young-Don Bae, Seong-Il Park, Yongseok Yi, In-Cheol Park

Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea

To cope with the growing design complexity and time-to-market pressure, today's designs are based on programmable processors and intellectual properties (IPs). According to this trend, many FPGA providers are now announcing new configurable devices that integrate one or more processors on an array of configurable logic blocks (CLBs) as their future products [1]. However, the devices are not sufficient to accommodate the features required in embedded system design [2]. In addition to fast processing, the processor must deal with a number of tasks to execute many threaded operations and control peripheral operations. As the operations are overlapped in execution, the processor should be equipped with fast task-switching capability. In addition, the logic array must be designed considering IP-based designs. As the conventional FPGA architecture is based on a flattened array of logic blocks, IPs are intermingled at the time of placement and routing, leading to many iterations of placement and routing in timing verification. A configurable device, called a single-chip programmable platform (SPP), deals with the problems.

Figure 20.2.1 shows the overall structure of the SPP containing a 32b multithreaded RISC processor (MT-RISC), configurable logic clusters (CLCs), and programmable bi-directional FIFO memories. Instead of conventional single-threaded processors, a multithreaded processor is embedded to eliminate task-switching overhead required to execute multiple threads concurrently [3]. A hardwired task scheduler is also embedded to assist the task switching. The task-scheduling algorithm is not fixed but configured by changing the parameters of the task scheduler. Configurable logic blocks (CLBs) are grouped into a set of CLCs and each CLC is occupied by only one IP to make the placement of the IP independent of those of other IPs. This clustering approach leads to easy timing verification at the cost of a little degradation of CLB usage. As many peripheral IPs such as UART and USB require memory buffers that consume a lot of area if configured in CLBs, FIFO memories are integrated to save area and guarantee performance. The FIFO memories are organized as circular buffers that temporarily store the data to be transferred between the processor and CLCs. The size and direction of a FIFO memory can be configured depending on the given specification of the peripheral IP such as event frequency, maximal latency and so on. The IP mapped to a CLC is regarded as a memory-mapped device in view of MT-RISC. A programmable address decoder is embedded to allocate a memory region for the corresponding IP mapped to a CLC. To provide direct communication paths among MT-RISC and CLCs, the decoder includes a number of control registers that can be used to control IPs.

The architecture of MT-RISC is shown in Figure 20.2.2. The processor handles up to 15 threads each of which is supported by 16 general-purpose registers including a stack pointer, a link register and a program counter. The instruction set includes thread control instructions and synchronization instructions as well as basic RISC instructions. The basic instructions are made similar to ARM to take advantages of good code density and well-established development environment. The processor includes a 32x32 MAC unit in its datapath to enhance multimedia processing capability. Compared to the conventional multiplier-only architecture, the MAC unit improves performance up to 23% at the same clock frequency when executing computation-intensive routines.

In conventional embedded systems, a task switching takes many clock cycles [4]. This overhead is almost eliminated by using MT-RISC. In addition, multithreading allows rapid processing of concurrent events that are essential in real-time systems, leading to fast responses to the events. Compared to a single-threaded processor that has the same basic instructions, the average response time is reduced to 3 % in experiments, as shown in Figure 20.2.3.

To reduce the switching overhead and assist the operating system, a hardware task scheduler is integrated on the chip. The dependencies among tasks incur temporal unavailability of shared variables, resulting in considerable waste of computing power. While the conventional real-time scheduling techniques rely on priorities and statistics, this task-scheduling mechanism exploits the dependencies among tasks [5]. Therefore, waste of computing power is completely eliminated. The organization of the task-scheduling mechanism is shown in Figure 20.2.4. The shared variable manager (SVM) monitors read and write operations on each shared variable and the task scheduler (TS) determines the next thread to be executed based on the current status of shared variables and FIFO buffers. All data transfer instructions of the processor are designed to communicate with SVM and TS.

A CLC consisting of 64 CLBs can be configured separately. As shown in Figure 20.2.5, inter-cluster switch blocks are used to connect CLCs. A CLB comprises four 4-input SRAM-based look-up tables (4-LUTs) and four registers. A local routing scheme enables the outputs of 4-LUTs to be fed to the inputs of 4-LUTs in the same CLB. To reduce interconnection delay, 10 short channels, 4 mid-length channels and 2 long channels are provided for a CLB. Experimental results over MCNC benchmarks show that the CLC operate at 50MHz on the average.

A prototype chip containing a four-threaded RISC, three CLCs, three FIFO blocks and 8kB on-chip SRAM is fabricated using 0.35μm CMOS technology with 4 metal layers. A micrograph of the chip is shown in Figure 20.2.6. The chip occupies 8x8mm^2 including pads, and a single power supply of 3.3V is used for logic core and IO pads.

References:

[1] P. Schaumont et al. "A Quick Safari Through the Reconfiguration Jungle," Proc. ACM/IEEE DAC 2001, pp. 172-177.

[2] K. Keutzer et al. "System-Level Design: Orthogonalization of Concerns and Platform-based Design," IEEE Transactions on Computer-Aided Design, Vol. 19, No. 12, Dec. 2000.

[3] S. Storino et al. "A Commercial Multithreaded RISC Processor," ISSCC Digest of Technical Papers, pp. 234-235, Feb. 1998.

[4] P. R. Nuth et al. "A Mechanism for Efficient Context Switching," Proceedings of IEEE International Conference on Computer Design, 1991, pp .301-304.

[5] J. Kreuzinger et al. "Real-time Scheduling on Multithreaded Processors," Proc. of IEEE International Conference on Real-Time Computing Systems and Applications, 2000, pp. 155-159.

0-7803-7335-9

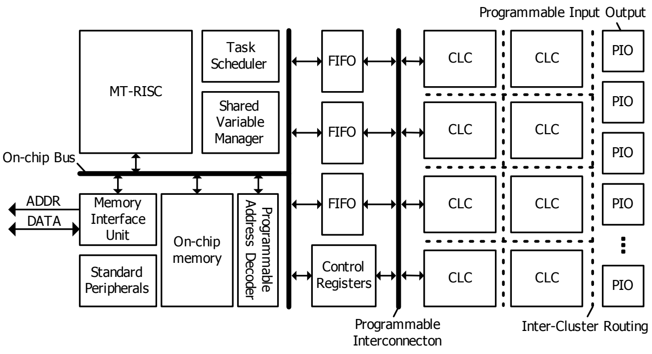

Figure 20.2.1: Single-chip programmable platform.

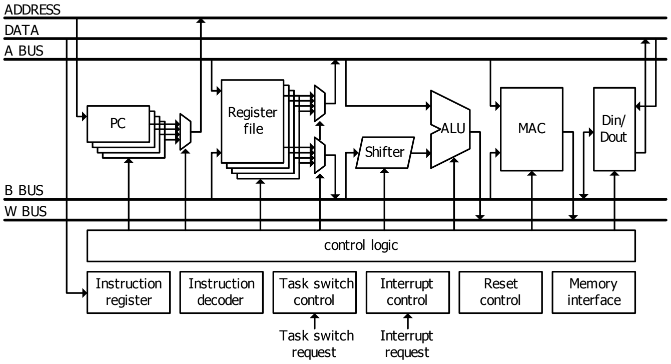

Figure 20.2.2: Architecture of MT-RISC.

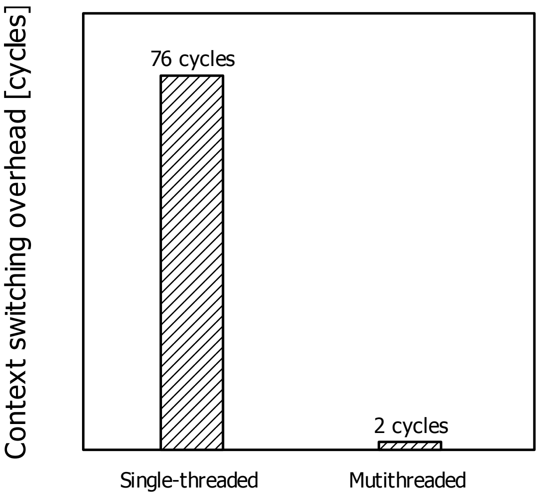

Figure 20.2.3: Context switching overhead.

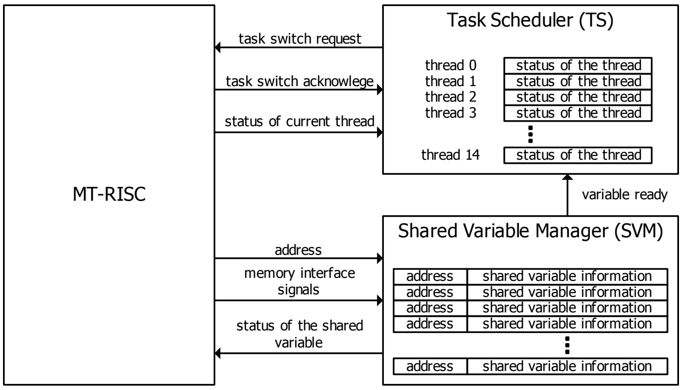

Figure 20.2.4: Task scheduling mechanism.

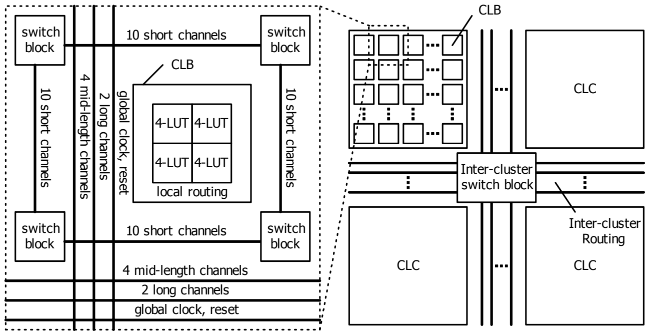

Figure 20.2.5: Configurable logic cluster (CLC).

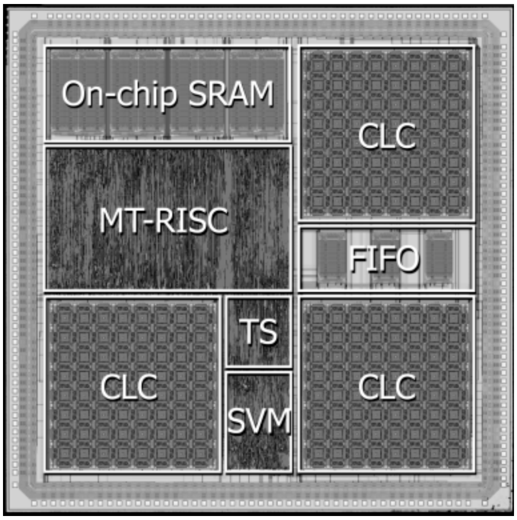

Figure 20.2.6: Micrograph of the protoype chip.

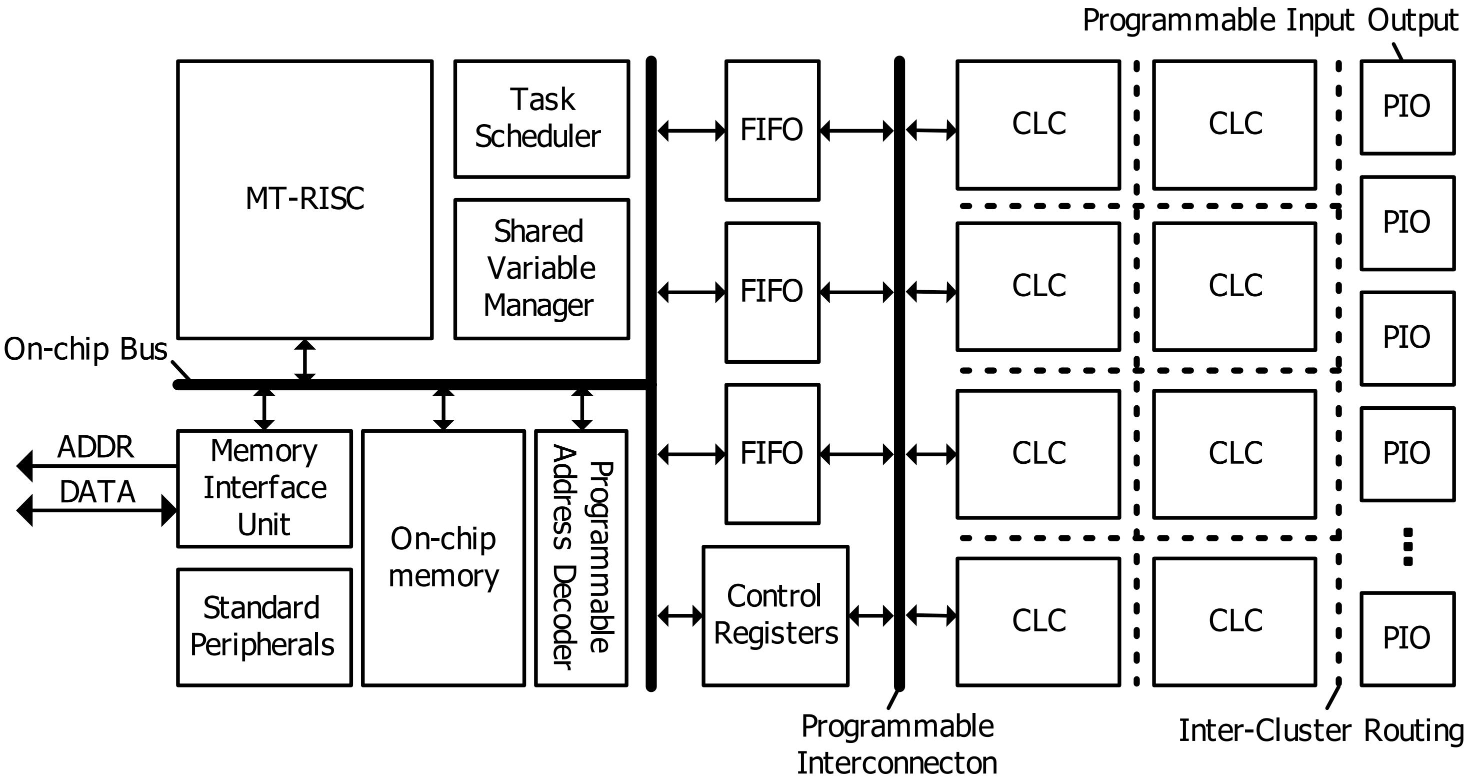

Figure 20.2.1: Single-chip programmable platform.

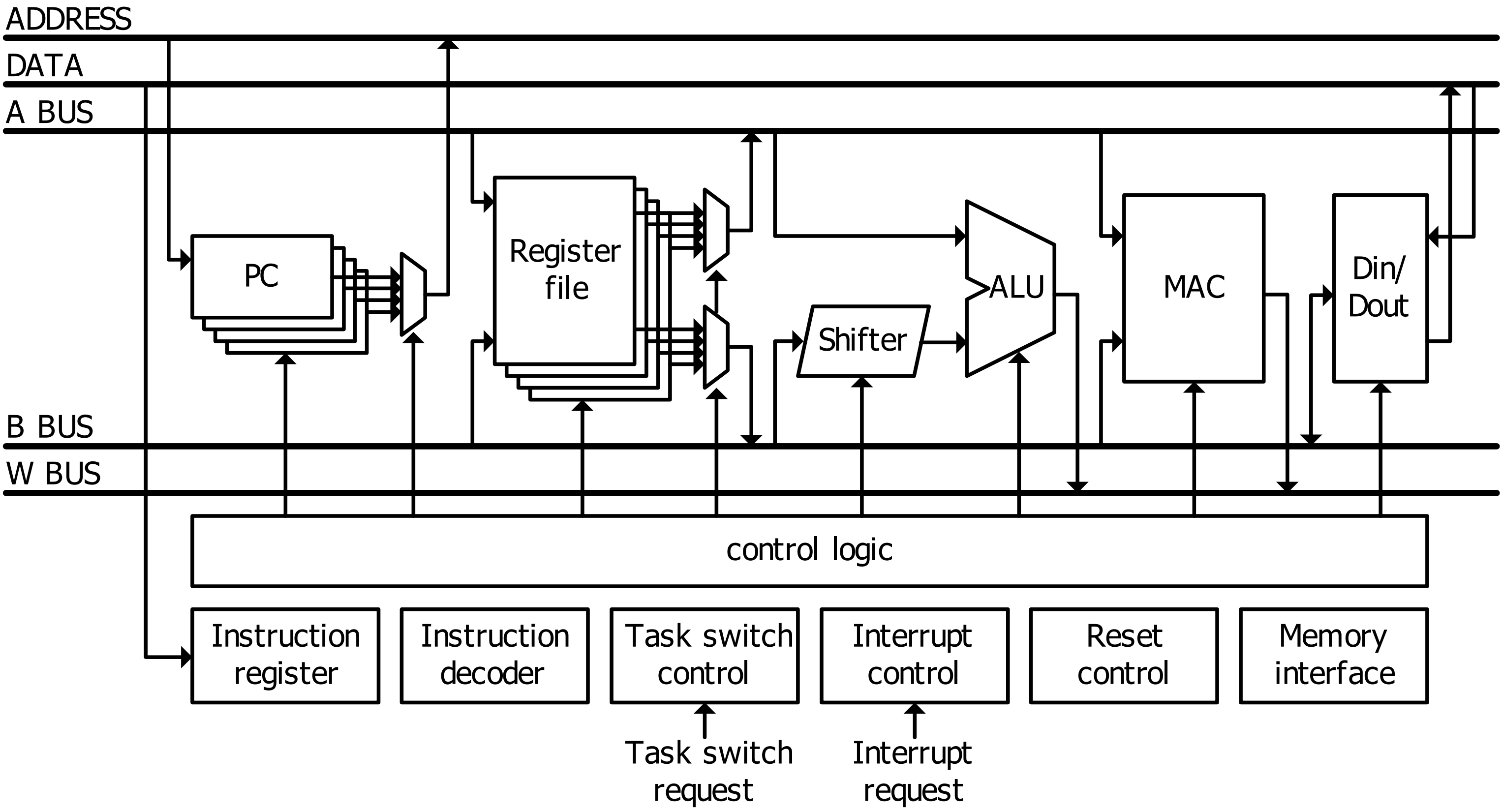

Figure 20.2.2: Architecture of MT-RISC.

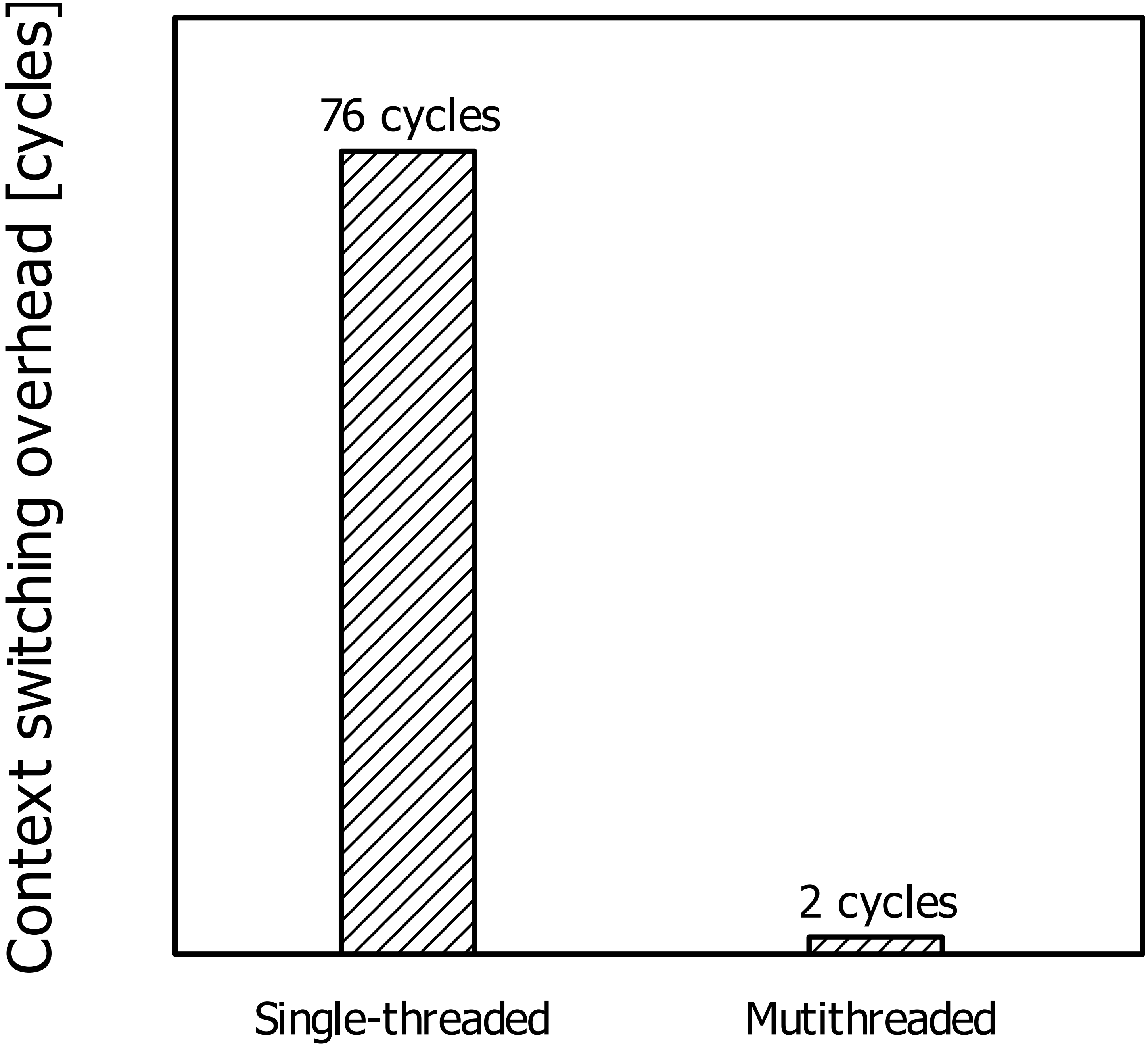

Figure 20.2.3: Context switching overhead.

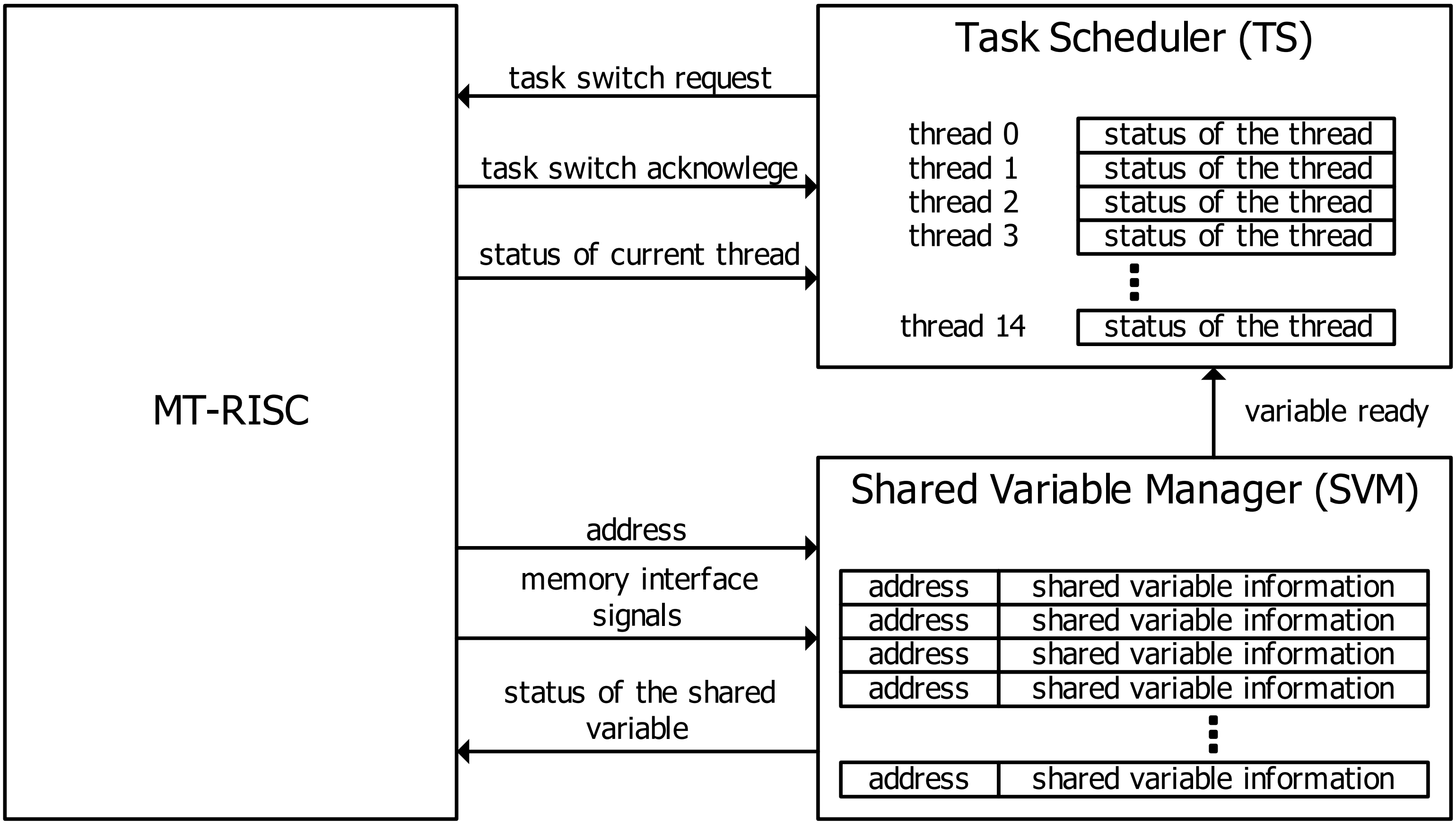

Figure 20.2.4: Task scheduling mechanism.

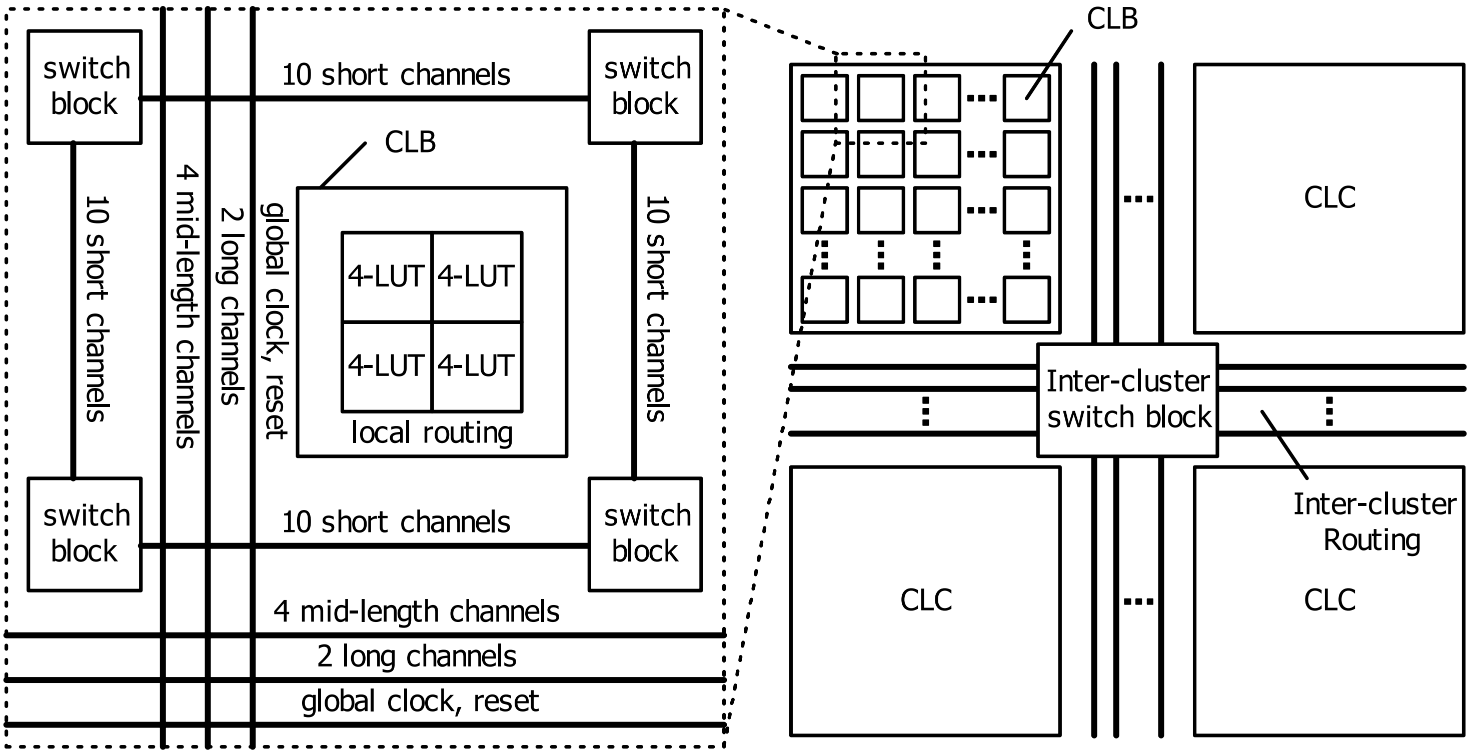

Figure 20.2.5: Configurable logic cluster (CLC).

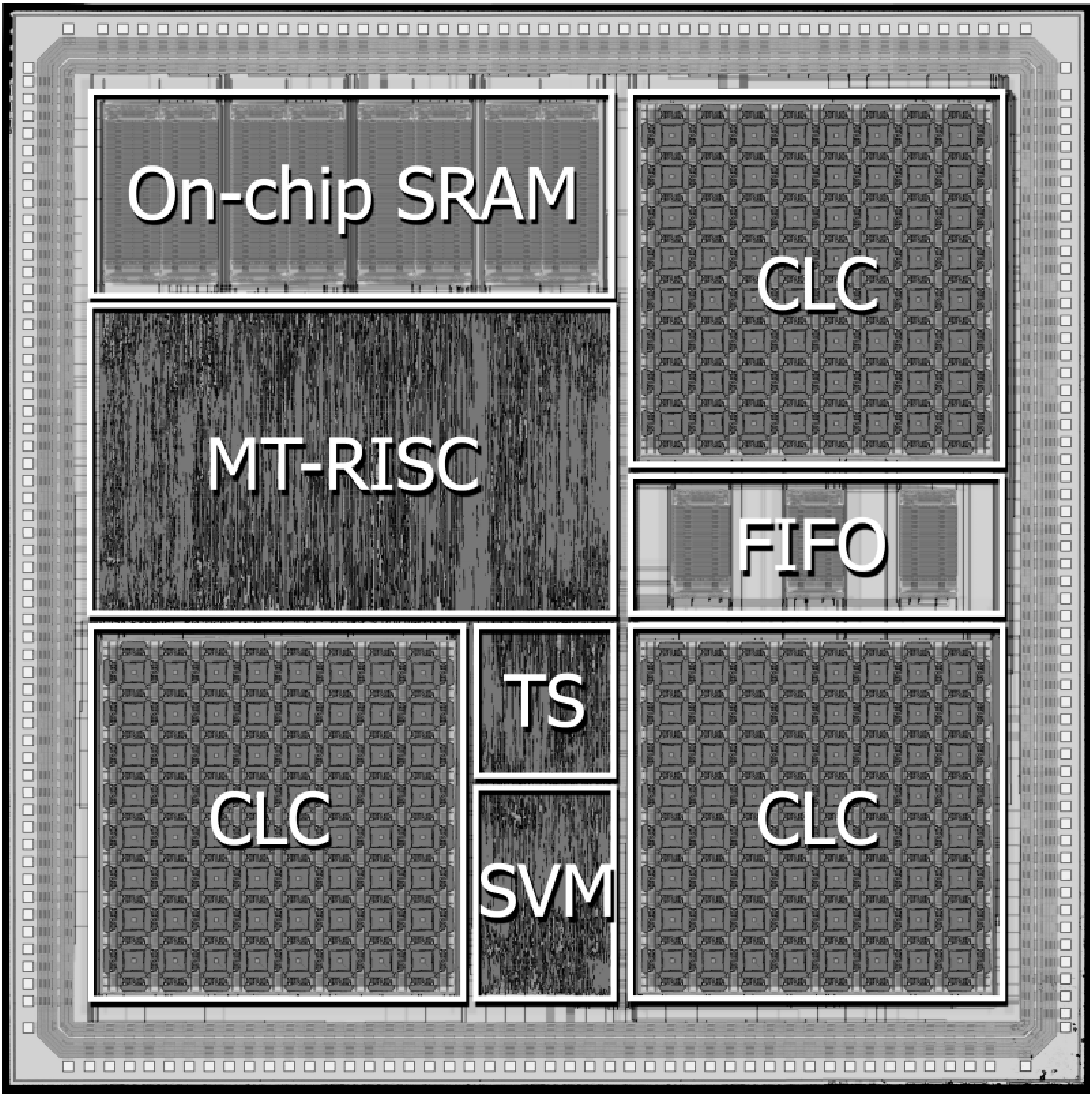

Figure 20.2.6: Micrograph of the protoype chip.