About the U8 MCU

Kenichiro Nagatomo Ryuji Kousaka
Tomomi Miyano Yoshiro Takada

While the performance of 8-bit microcomputers has improved in recent years as the functionality of digital home appliances has become more sophisticated, there is also a growing demand for ultra-low power consumption when the microcomputer is not running.

Oki Electric has developed the U8 core, an original 8-bit microcomputer architecture, in order to meet the needs for both high performance and ultra-low power consumption, and has realized a product with this built-in 8-bit microcomputer.

This paper describes the features and usefulness of the U8 core, microcomputers that use it to achieve ultra-low power consumption, and its software development environment.

Design based on silicon platform architecture

The U8 core is designed based on the concept of Silicon Platform Architecture (SPA). This SPA is an architecture that attempts to build an LSI system by combining software on a strictly defined and configured LSI hardware platform [1].

Due to the miniaturization of today's LSIs and the increasing degree of integration that accompanies advances in manufacturing technology, LSIs will become obsolete within a year or two. On the other hand, software that implements applications based on the LSI cannot be easily updated. In other words from the point of view of software, it is necessary to be able to easily add parts that implement newly required functions while using existing software as is. In addition, it is required to integrate more functions. In LSIs built by integrating logic and memory on this platform, software can be executed in common regardless of their configuration. By adopting the SPA concept, software and hardware are conceptually separated in this way, and hardware can be converted to LSI using a more advanced process, greatly improving the applicability of software [2].

The U8 core is described in RTL (Register Transfer Level) in order to thoroughly implement this SPA concept. The U8 core is a synthesizable core and is designed to be process-independent.

U8 core architecture overview

The U in the U8 core comes from the Unified Core (Unicore), and the 8 comes from the bit length of an 8-bit microcomputer. Microcomputers are generally 8-bit, 16-bit, and 32-bit, each with a different architecture. However, the unicore architecture has the same instruction system even if the bit length is different from 8, 16, 32, and the binary code is designed to be compatible between cores with different bit lengths. In this way, multiple cores with different bit lengths are called unified cores in the sense that they are configured under a unified architecture, and the 8-bit version is named U8.

The U8 core is designed as a RISC (Reduced Instruction Set Computer) architecture core with a three-stage pipeline. It was originally designed so that arithmetic operations, logic operations, bit processing, etc. could all be processed once through registers. However, some application programs were implemented on this core, and based on the results of evaluating this architecture, improvements were made to the architecture. Programs that control embedded devices have a greater proportion of program flow control than arithmetic processing, so the command system was changed to suit these processes. Specifically, instructions for directly manipulating bits in memory and instructions for incrementing and decrementing memory contents have been added. As a result, code efficiency has been improved, and about 70% of instructions can be executed in one clock, making it possible to achieve high performance.

The gate size of the U8 core is about 10K gates, and the performance is about 10MIPS at 32MHz operation in the evaluation by Drystone (one of the benchmark tests to measure CPU performance). Compared to conventional 8-bit microcontroller cores and 16-bit microcontroller cores that use the CISC (Complex Instruction Set Computer) architecture, the U8 core has the same code

efficiency and two to three times the performance.

Furthermore, the U16 core, which extends the arithmetic circuit and data bus width to 16 bits, achieves twice the processing performance compared to the U8 core. In applications that involve bit processing, byte processing, and controlling embedded devices, the performance improves by approximately 1.4 times.

U8 Core Features

The U8 core is an original 8bit RISC core designed with low power in mind, and its features are as follows.

(1) Pipeline RISC architecture

As shown in F.1, it uses a 3-stage pipeline RISC architecture for instruction fetch (IF), instruction decode (ID), and instruction execution (EX), and most byte instruction processing operates in one clock. This makes it possible to reduce power consumption by speeding up processing and shortening instruction execution time.

(2) Rich instruction set

100 kinds of instruction sets such as transfer, arithmetic operation, logical operation, comparison, memory operation, branch, conditional branch, call/return, stack operation, shift instruction, etc. are prepared, and it is an instruction set that can flexibly support various applications.

(3) Adoption of instruction and register config considering C language

It has 8-bit x 16 general-purpose registers and supports 16-bit and 32-bit arithmetic operations, loading, storing, and stack operations. Despite being 8-bits, it is designed with an instruction set and register configuration highly compatible with the C programming language.

(4) Equipped with memory and bit manipulation instructions

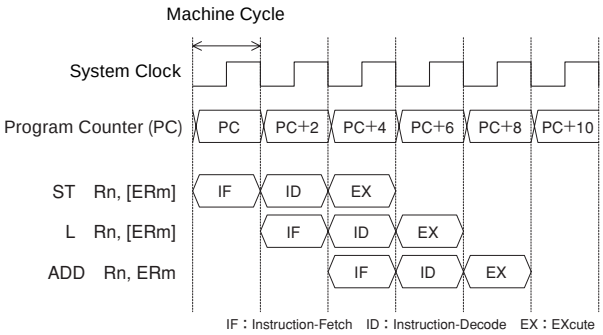
It provides instructions for memory increment/decrement/bit manipulation, which are not available in the conventional load/store type RISC architecture. This enables faster memory operations than conventional RISC architectures.

(5) Coprocessor support

Dedicated coprocessor connection bus, register transfer instructions between coprocessor and U8 core, and direct transfer between coprocessor and memory, assuming attachments such as DSP and encryption circuits, considering application to communication LSI and voice LSI I have an order ready. This enables high-speed data transfer between the coprocessor and memory, which is not possible with conventional RISC architecture.

(6) Fast interrupt transition handling

It supports three types of interrupts: maskable interrupts, non-maskable interrupts, and software interrupts. Each interrupt has a register inside the U8 core for holding the status when an interrupt occurs. When an interrupt occurs, the interrupt transition cycle is completed in 3 cycles by saving the state at the time of interrupt generation to the above registers during the interrupt transition cycle, enabling faster interrupt transition processing than before.



F.1 3-stage pipeline RISC architecture

Functions of ML610501/ML610503

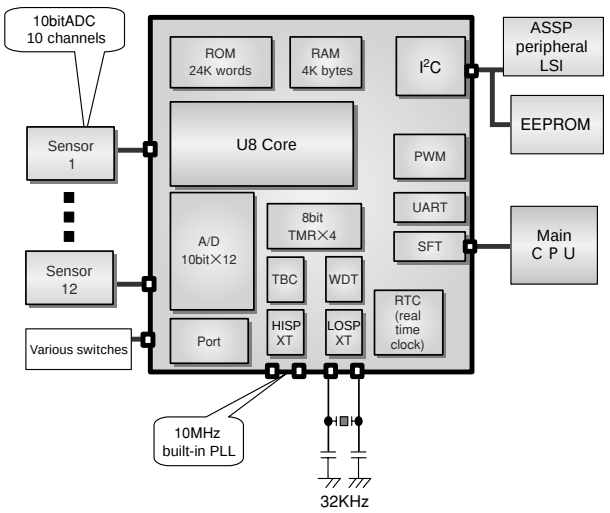
ML610501 has been commercialized as a general-purpose microcomputer product with a U8 core, and ML610503 is under development. The specifications of each are shown in T.1, and the block diagram of the ML610503 is shown in F.2.

The ML610501 can operate with a low power consumption of 1.5 μA during HALT mode and 6 μA even when operating at 32 kHz. It is most suitable as a microcomputer. It also uses an ultra-compact 64-pin BGA package of 6mm x 6mm size, which is expected to contribute to space saving. The ML610503, which is currently under development, is characterized by a faster operating frequency, enhanced RAM and I/O, and a built-in real-time clock compared to the ML610501.

In portable devices, in order to have time information, it is necessary to either attach an external real-time clock or incorporate a clock program. When using an external real-time clock LSI, implementation space and communication ports are required. Additionally, if a clock program is embedded in the controlling microcontroller, programming and evaluation of the clock program are necessary, resulting in a significant development burden on the customer. Moreover, it becomes problematic as the power consumption becomes higher than that of a standalone

T.1 Specifications of ML610501/ML610503

	ML610501	ML610503 (under development)
Power Supply/Clock	2.7～3.6V/5MHz	2.7～3.6V/10MHz
	1.8～3.6V/4MHz	1.8～3.6V/4MHz
Supply Current	1.5μA@32KHz(HALT)	2μA@32KHz(HALT)
		0.4μA@32KHz(RTC Only)
Minimum instruction execution time	30μs(32KHz system clock)	30μs(32KHz system clock)
	200ns(5MHz system clock)	100ns(10MHz system clock)
Internal ROM	48KB	48KB
Internal RAM	2048B	4096B
I/O port	IN:8	IN:8
	I/O:33	I/O:48
Timer	8-bit×4	8-bit×4
	Watch Dog Timer	Watch Dog Timer
	Time base Counter×2	Time base Counter×2
SIO Port	UART×1	UART×1
	Shift clock×1	Shift clock×1
	I2C×2(master)	I2C×1(master)
ADC	10-bit×10	10-bit×12
PWM	16-bit×1	16-bit×1
Interrupt Controller	external:8	external:8
	internal:15	internal:15
Operating temperature	−20℃～+70℃	−20℃～+70℃
RTC (Real Time Clock)		SEC,MIN,HOUR, WEEK,YEAR
Package	64PIN BGA(6mm×6mm)	84PIN BGA(6mm×6mm)
	64PIN BGA(7mm×7mm)	

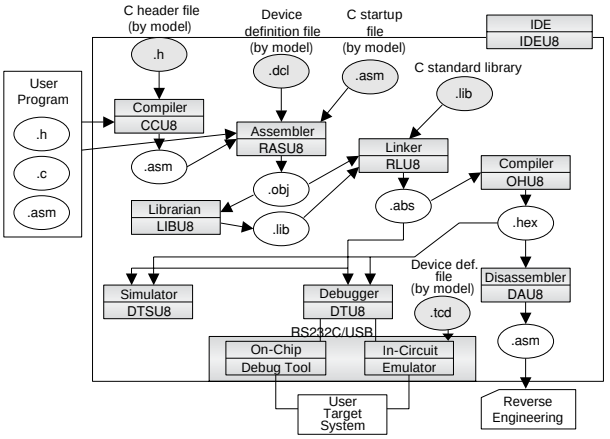


F.2 ML610503 block diagram

real-time clock. However, with ML610503, it has an integrated real-time clock that operates independently of the CPU. Even when the CPU is in HALT or STOP state, it can operate with power consumption similar to that of a standalone real-time clock (0.4μA).

IDE for U8 MCU

This article outlines the U8 integrated development environment, which is a development environment for supporting software debugging and system evaluation for embedded system development using the U8 microcomputer. F.3 shows the configuration.



F.3 Configuring the U8 IDE

(1) U8 integrated development environment

The U8 integrated development environment integrates development support software such as C compilers, assembler packages and debuggers into one, and also has functions for improving software development such as project management functions and programming editors. The U8 integrated development environment can be used as a tool that integrates the program development process, from program creation using an editor to compilation, assembly, and linking, as well as the program debugging process using a debugger.

(2) C compiler (CCU8)

CCU8 is a C compiler that conforms to the ANSI standard, and has various command-line options to optimize for generating code suitable for the target embedded system. In addition, dedicated pragmas (pseudo-instructions for passing specific information to the compiler) are also provided to generate efficient code by taking advantage of the U8 core's specific instructions and memory structure.

(3) Assembler package (MACU8)

The MACU8 assembler package is a general term for assembler, linker, librarian, and object converter, and is software that converts U8 programs written in assembly language into object code.

(4) Simulator (DTSU8)

The DTSU8 simulator is a software tool for evaluating and debugging programs created with the CCU8 compiler and MACU8 assembler package. It simulates the operation of the U8 microcomputer at the instruction level on a PC and provides an advanced and easy-to-use interactive debugging environment.

(5) Debugger (DTU8)

The DTU8 debugger is software that has the same GUI (Graphical User Interface) as the DTSU8 simulator, and supports debugging and evaluation of application programs by connecting to an in-circuit emulator.

(6) In-circuit emulator

An in-circuit emulator is hardware that emulates in real time the same functions as the debug target microcomputer. It also has the debugging functions shown in T.2 for efficient debugging and evaluation of application programs.

T.2 In-circuit emulator debug functions

Function	Specification
Interface	USB : USB 1.1 compliant
Emulation Functions	Real-Time Emulation Step Emulation (Step In/Step Over)
Break Function	Hardware BP (with pass count), software BP, RAM data match, forced break, external break
Trace Function	Trace information: execution address, PSW, RAM address, RAM data, probe
Real-Time Display Function	Viewing RAM during real-time emulation
Execution Time Measure. Func.	Exec. time measure. by internal timer Execution time measurement by execution cycle measurement
Probe Cable Function	Break signal input, sync signal output, trace data input

Current Status and Future of U8 MCUs

So far, I have briefly described the general-purpose U8 microcontrollers ML610501 and ML610503, but they are starting to be incorporated into microcontrollers for battery-powered meters, custom microcontrollers for speech synthesis control, and LSIs for specific applications.

In this way, the main features of the U8 core, low power consumption and high performance, are expected to further expand the application of the U8 core to system LSIs in the future. ◆◆

References

1) Mukai: Issues and Responses in System LSI Development -Significance and Strategy of SPA-, Oki Electric Research & Development Issue 184, Vol.67 No.3, pp.3-6, October 2000
2) Kizumi et al.: μPLAT® Hardware Development, Oki Electric Research & Development Issue 184, Vol.67 No.3, pp.45-48, October 2000

About the Authors

Kenichiro Nagatomo. Leader of Team 2, System Development Department 2, Oki Techno Collage System Headquarters
Ryuji Kousaka. Leader of Team 1, System Development Department 2, Oki Techno Collage System Headquarters
Tomomi Miyano. Team 2, System Development Department 2, Oki Techno Collage System Headquarters
Toshio Takata. Managing Director of Oki Techno Collage Co., Ltd.