New 78K0R Series 16-bit Microcomputers

MIZOGUCHI Makoto, ISHIKAWA Kiyoshi, NAKANO Masataka
KUMAGAI Toshiyuki, ISOGAI Hideo, TAKANA Kentaro

Abstract

In order to meet the needs of the embedded systems market by such means as performance improvements, power consumption reductions and a shortening of the development period, NEC Electronics has newly developed 16-bit microcomputers of the 78J0R series. The aim has been to "achieve the performance of a 16-bit microcomputer with the power consumption of an 8-bit microcomputer," "secure upward compatibility with the 8-bit microcomputers (78K0)" and "provide a comfortable development environment." The measures taken for reducing the power consumption include the adoption of an advanced flash-mixed CMOS process and circuitry and system devices including those for the CPU (Central Processor Unit) core. The 78K0R series will offer the user high performance and low power consumption in a variety of fields.

Keywords

microcomputer, 16-bit microcomputer, flash memory, flash microcomputer, high performance
low power consumption, embedded equipment, development environment

1. Introduction

The fields covered by embedded equipment have recently expanding widely, and microcomputers are now embedded in every kind of device. This trend of expansion is expected to continue in the future.

The user of embedded equipment has a permanent need for function improvements and microcomputers for use in system control are always required to improve their functions and performances. At the same time, a rise in environmental considerations has tended to make a reduction in their power consumption, which is a key issue.

The 78K0R series of new 16-bit microcomputers meets the above market needs by achieving 16-bit performance with 8-bit class power consumption. This achievement benefits the user who wants to improve equipment performance with a minimum increase in power consumption and contributes to the enhancement of customer competitiveness.

2. Concept of the New 16-bit Microcomputers

NEC Electronics have previously developed 8-bit microcomputers of the 78K0 series and deployed various products.

Nevertheless, the recent trend toward equipment function improvement has led to an increase in the number of 8-bit microcomputer users claiming. "The 8-bit microcomputer cannot offer much headroom in performance" or "The 8-bit microcomputer will soon become inadequate." These users considered shifting to 16-bit microcomputers, but existing 16-bit microcomputers including our previous products were unable to satisfy their needs due to "increased current consumption," and "unsuitability due to the appropriation of software assets," etc. In addition, when using a new microcomputer, it is necessary to create an environment for software development. The development environment can be divided into a software environment including the compiler and debugger, and the hardware environment for emulating microcomputer operations.

With regard to the software environment, there has been a need for ease of use to be met by improving the performance (code efficiency) of the compiler and to provide tools for the simplified development of driver-level software. In the case of the hardware environment, this should use the actual microcomputer and has not been available until completion of the microcomputer development, thus posing a bottleneck by a shortening of the set development period of users. This situation has resulted in a need for the early provision of a suitable environment and for enabling the use of the same development environment as for the 8-bit microcomputers of the 78K0 series.

We started development of the new 16-bit microcomputer series aiming at meeting the above needs with the concepts of "achieving the performance of 16-bit microcomputer with the power consumption of 8-bit microcomputer," "securing the upward compatibility to 8-bit microcomputers (78K0)" and "providing a comfortable development environment."

3. 16–bit Performance with 8–bit Power Consumption

As seen in Section 2 above, the market needs a microcomputer with both high performance and low power consumption. Accordingly, we have developed our new products by targeting low-power-consumption microcomputers that not only improve the processing performance but are also able to reduce the power-to-performance ratio compared to the previous 8-bit and 16-bit microcomputers. Since increasing the operation frequency for improving the processing performance results in increasing the power consumption, we decided to pipeline the CPU based on an analysis of previous products and in order to improve performance without increasing the operation frequency. The clock supply control was thus set as being of primary importance in the power saving effect and became the main issue to be tackled in order to implement the requisite power consumption reduction.

1) CPU Pipelining

Our 8-bit microcomputers of the 78K0 series require an average of 4 clocks in order to execute each instruction. Increasing the operation frequency can improve the performance but the power consumption is increased proportionally. Therefore, we adopted a pipeline configuration that was capable of executing an instruction with a single clock. The pipeline configuration divides CPU operations into several stages and runs them concurrently to increase the overall processing speed. It therefore allows the processing performance to be improved without changing the operation frequency. We then examined the optimum pipeline configuration for achieving the target current drain and performance for the new microcomputers. We were able to confirm thus that a 3-stage pipeline is able to achieve the target performance with the minimum current drain. In this way, we were able to improve the performance without increasing the operation frequency (**Fig. 1**).

2) Clock Control of Peripheral Devices

It was found with the previous products that their clock supply circuitry continued working even when the peripheral functions were not used, resulting in the wasteful consumption of about 20% of the operation current even in the standby mode. As shown in the previous configuration in **Fig. 2** , the prescaler circuit generates several kinds of clocks and supplies them individually to the peripheral functions. The current is consumed wastefully due to the clock generator and to the load on the wiring paths to the peripheral circuits, whether these were in use or not. To eliminate this waste, we incorporated a mechanism for stopping the clock to each peripheral function as desired. As shown in the new configuration in Fig. 2, this function supplies only one clock signal to each peripheral function and places a prescaler in each of the peripheral functions. When a peripheral function is stopped it also stops the clock signal supply to it in the clock generator circuit. This procedure has made it possible to eliminate power wastage completely by the peripheral functions when they are stopped.

Reduction of power consumption in the standby mode is also a key point. We investigated the circuits that are used

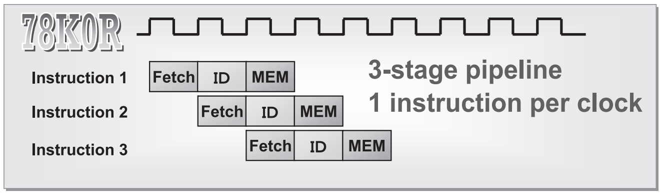

Fig. 1 Pipeline.

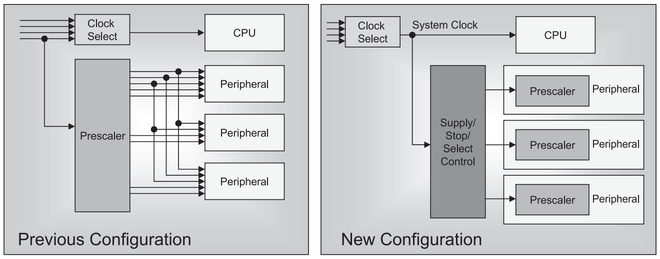

Fig. 2 Clock supply paths.

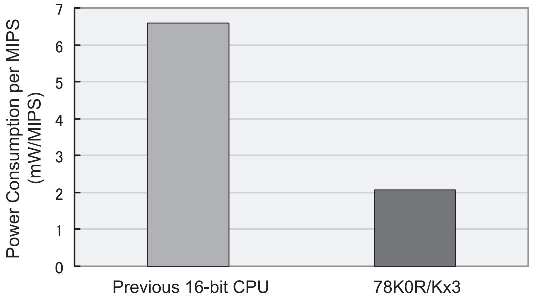

Fig. 3 Power/performance ratio comparison.

only for normal operation and which are unnecessary when the CPU is stopped and added a control that stops the clock supply to these circuits inside the clock generator circuit. Together with the clock supply control to the peripheral functions described in the previous paragraph, this strategy has reduced the standby current by about 40% when all of the peripheral circuits are stopped compared to the estimated standby current of the circuit based on the previous clock control.

The efforts as described above have allowed us to achieve the highest power-to-performance ratio (1.8mW per MIPS) in the16-bit microcomputer industry and below 1/3rd of our previous 16-bit microcomputers as shown in **Fig. 3** .

4. Upward Compatibility with 8-bit Microcomputers

While making use of the software assets of the previous 78K0 series 8-bit microcomputers, we also optimized the instruction set and assembler and expanded the memory space to achieve an improved performance.

(1) Simultaneous Achievement of Instruction Compatibility and Performance Improvement

We adopted automatic instruction conversion of the assembler for the 78K0R series so that instruction compatibility may be obtained with the 78K0 series at the assembler language level. For the specification in the expanded address space, we mounted a CS register with the capability of independent specification of the higher 4 address bits to enable specification of the 20-bit address space using a 16-bit operant. This has made the 78K0R possible to use the register pair branching and 16-bit absolute address branching instructions of the 78K0 without modification.

For the improvement of performance, we added shift instructions (SHR, SHL, etc.) for high-speed processing of multiplications and divisions such as 2, 4, 1/2 and 1/4 in place of the multiplication instruction (MULU) that is often used in address calculations during handling of 16-bit arrays of data, etc.

In addition, to deal with delays due to branch instructions that would occur even when the processing speed of computation instructions are increased by the pipeline configuration, we packaged skip instructions (SKC, SKZ, etc.) that can disable the subsequent unexecuted instructions depending on the result of condition judgment. This has made is possible to process every branch instruction in one clock while it would normally take 2 clocks per branch for processing. We also added 16-bit computation instructions, etc., and succeeded in a high performance of 13 MIPS with Dhrystone 1.1.

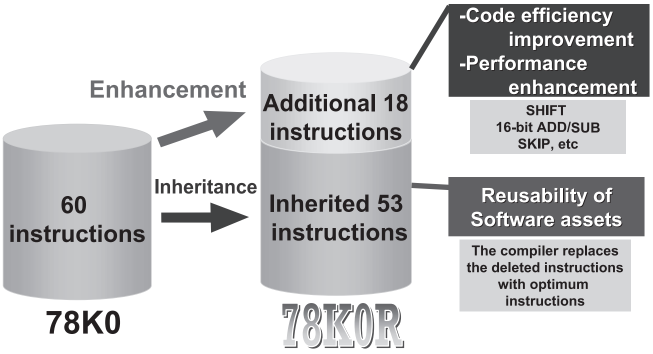
-Code efficiency improvement
-Performance enhancement
Enhancement
Additional 18 instructions
SHIFT
16-bit ADD/SUB
SKIP, etc
Inheritance
60 instructions
Inherited 53 instructions
Reusability of Software assets
The compiler replaces the deleted instructions with optimum instructions
78K0
78K0R

Fig. 4 Instruction addition with 78K0R.

As a result, enhancement of performance was achieved while maintaining compatibility with the 78K0 as shown in **Fig. 4** .

(2) Expansion of Memory Space

The memory space is set to a maximum of 1MB in linear addresses. This was determined in consideration of the ROM memory size and the built-in high-speed RAM and SFR (Special Function Register) spaces required by the 16-bit market. We expanded the space using the linear address method (memory space with a one-dimensional array) that is easier to use than the bank switching method (multidimensional array of memory spaces with one-dimensional arrays aiming at increasing the memory space without increasing the number of bits in the address bus). We also applied other measures at the same time as expanding the memory space.

One of these is the provision of an address specification method that references data placed in the program area. Three bytes are required to specify an address (operand) in a 1MB memory space. The instructions with 1-byte opcodes (Operation codes) should be placed on the instruction map by prioritizing those that are important for performance and code efficiency. For this reason it has been necessary to create a mechanism that makes it possible to specify addresses in the 1MB memory space using a 2-byte operand that can usually be used to specify only the 64k-byte addresses. To resolve this issue, we prepared an ES register for specifying the higher 4 bits of the address independently and the PREFIX instruction. We also added

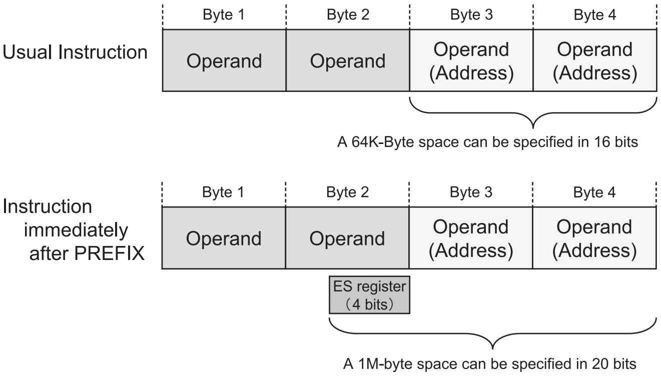
Byte 1
Byte 2
Byte 3
Byte 4
Usual Instruction
Operand
Operand
Operand (Address)
Operand (Address)
A 64K-Byte space can be specified in 16 bits
Instruction immediately after PREFIX
ES register (4 bits)
A 1M-byte space can be specified in 20 bits

Fig. 5 ES register and PREFIX instruction.

a mechanism for adding the value of the ES register as the higher 4 bits of the address indicated by the operand of the instruction immediately after the PREFIX instruction as shown in **Fig. 5** . This procedure enabled the specification of 20-bit addresses and placed the address specifications in the 1MB memory space using 2-byte operands.

5. Provision of a Comfortable Development Environment

We have prepared tools that can improve the ease of software development that can be used commonly with the 78K0 series 8-bit microcomputers as the development environment of the 78K0R series. These tools include the “Applifet2” device driver configurator (tool for supporting the creation of control programs for microcomputer’s peripherals), the “SM +” software simulator, the MINICUBE2 on-chip debugging emulator that can also be used with the 78K0 series 8-bit microcomputers, and the IECUBE full-performance in-circuit emulator (full ICE or in-circuit emulator) that can be provided at an early stage. We also developed the CC78K0R compiler in consideration of the increase of developments using the C-language with the 16-bit microcomputers. We especially applied new measures to the CC78K0R compiler and the IECUBE full ICE in order to round-off the concept of “providing a comfortable development environment.”

For the compiler, the CC78K0R has been designed for the 78K0R series of 16-bit CPU core. Since most of the users of 16-bit microcomputers use the C-language in programming, it has been necessary to improve the code efficiency (i.e. to decrease the code size). So we adopted the latest optimization technology in addition to the technologies used with the 78K0 8-bit CPUs. However, when we conducted benchmarking at this stage, we regretfully found that our compiler was still not able to outclass our competitors’ compilers. Consequently, we proceeded to examine techniques that could improve the code efficiency further and discovered that the code efficiency can be improved greatly by organizing any similar code detected into a single subroutine automatically (this technique is called the automatic subroutine organization technique). Nevertheless, because of anxieties that the use of this technique would make debugging difficult due to the difficulty of obtaining debugging information, we developed an original auto search algorithm for optimization without compromising the debugging efficiency. As a result, we have succeeded in developing a compiler that can outclass those of our competitors by improving the code efficiency by about 20%, as shown in **Fig. 6** .

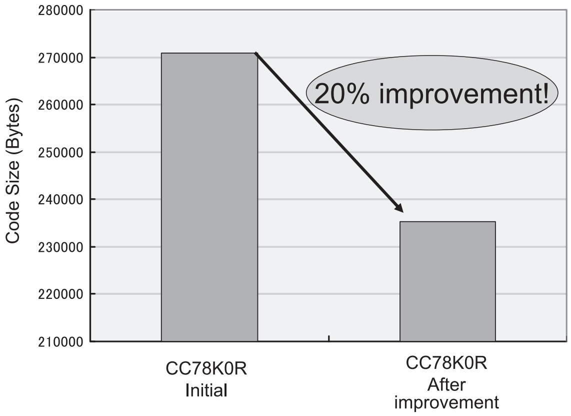

Fig. 6 Improvement in the code efficiency (Code size reduction).

For the full ICE, the emulation chip of the previous full ICE was implemented in the microcomputer in order to ensure functional equivalence, so it was not available until prototyping of the microcomputer completed. With the full ICE for the 78K0R series, we implemented logic functions using the FPGA and the analog section that cannot use FPGA, using equivalent functions based on a combination with other devices. This made it possible to provide the full ICE as soon as the microcomputer circuit design is finished as shown in **Fig. 7** . An early provision of the development environment allows the user to proceed to software development earlier, thus providing a significant advantage in the early development of sets.

Implementation of a full ICE based on FPGA has previously been used only for limited users, but this technique is accompanied by a major problem, which is the necessity of redesigning the async circuitry into sync circuitry because the async circuitry in the microcomputer cannot be incorporated

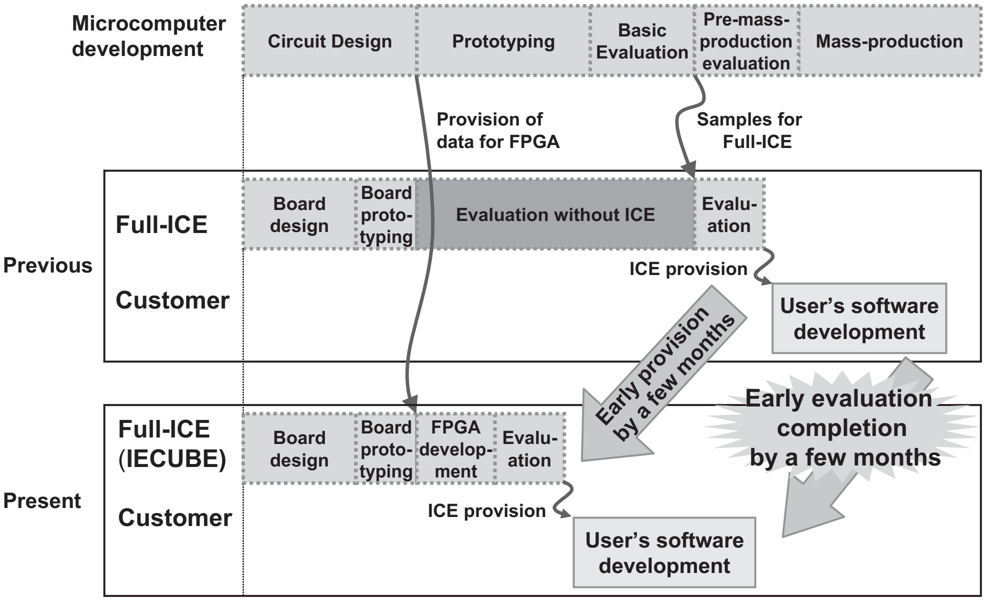

Fig. 7 Provision of full ICE, difference in user's development completion period.

into the FPGA. This problem has hindered achievement of full functional equivalence to the microcomputer in spite of its importance to the full ICE. In addition, early provision of the full ICE has also been delayed due to the need for a specified period for full ICE development after completion of the microcomputer circuit design. This is because of the design labor required for "synchronization of the async circuitry" and a need for verification patterns unique to the ICE. In order to solve these problems, we designed the circuitry of the 78K0R series by taking into consideration the use of FPGA and by eliminating async circuitry wherever possible, etc., from the beginning of the microcomputer development. As a result, it has become possible to provide the full ICE with functional equivalence to the microcomputer immediately after completion of its circuit design. The implementation of full ICE based on FPGA is not only advantageous for advancing the start of development but can also reduce the burden on the upgrading the development environment version. With regard to previous version upgrading, the ICE had to be brought to the servicing shop for the replacement of required parts. However, the version of the 78K0R series can be upgraded by simply rewriting the FPGA data. This means that the full ICE can be kept in the custody of users and upgraded versions can be provided in a timely manner without hindering the development activities of users.

These new measures taken for both the software and hardware environments have enabled a comfortable development environment that improves on that of our competitors and contributes to the improvement of set development efficiency. This is achieved in spite of the increases in function complications of the applications and sets.

6. Conclusion

As described in the above, the new 78K0R series of 16-bit microcomputers can fill the gap between market needs and actual products by achieving the aims of "achieving the performance of the 16-bit microcomputer with the power consumption of the 8-bit microcomputer," "securing the upward compatibility of 8-bit microcomputers (78K0)" and "providing a comfortable development environment," and by contributing added value to user systems.

Actual products of the new 78K0R series of 16-bit microcomputers are already being marketed, such as the 78K0R/Kx3. These are basic microcomputer products equipped with universal functions. However, some applications released in the market require peripheral functions that cannot be operated by microcomputers with basic functions. In the future, we will expand the line of products by introducing functions to match the required applications, such as a version that incorporates the "LCD C/D version" and a "3-phase version incorporating a motor control” as shown in **Fig. 8** .

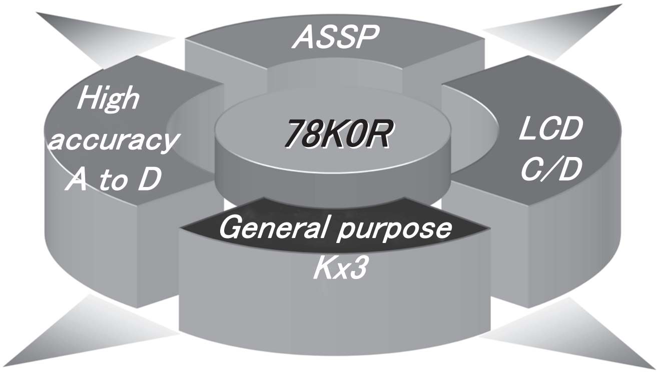

Fig. 8 Future deployment of the 78K0R.

Authors' Profiles

MIZOGUCHI Makoto
Team Manager,
Multipurpose Microcomputer Systems Division,
Microcomputer Operations Unit,
NEC Electronics Corporation

ISHIKAWA Kiyoshi
Assistant Manager,
Multipurpose Microcomputer Systems Division,
Microcomputer Operations Unit,
NEC Electronics Corporation

NAKANO Masataka
Assistant Manager,
Multipurpose Microcomputer Systems Division,
Microcomputer Operations Unit,
NEC Electronics Corporation

KUMAGAI Toshiyuki
Team Manager,
2nd Microcomputer Division,
NEC Electronics Corporation

ISOGAI Hideo
Assistant Manager,
2nd Microcomputer Division,
NEC Electronics Corporation

TAKANA Kentaro
Assistant Manager,
2nd Microcomputer Division,
NEC Electronics Corporation