Interrupt control device of small hardware size which deals with much interrupt processing flexibility

Kleine Unterbrechungssteuerungseinrichtung zur flexiblen Unterbrechungsbehandlung

Dispositif de commande d'interruption de petite taille avec flexibilité de traitement des interruptions

Designated Contracting States:
DE FR GB NL

Priority: 11.05.1995 JP 11313095

Date of publication of application: 13.11.1996 Bulletin 1996/46

Divisional application: 09157347.7 / 2 077 503

Proprietor: Panasonic Corporation
Kadoma-shi
Osaka 571-8501 (JP)

Inventors:
• Miyaji, Shinya
  Nara-shi, Nara 631 (JP)

• Higaki, Nobuo
  Higashi-Yodogawa-ku,
  Osaka-shi,
  Osaka (JP)

Representative: Crawford, Andrew Birkby et al
A.A. Thornton & Co.
235 High Holborn
London WC1V 7LE (GB)

References cited:
EP-A- 0 306 042
EP-A- 0 637 802

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).
(1) Field of the Invention

[0001] The present invention relates to an interrupt control device having excellent flexibility of interrupt program design in a microcomputer.

(2) Description of the Related Art

[0002] Recently, as electronic technique and information processing technique developed, performance of microprocessors has been upgraded. They are now widely used in various electronic devices.

[0003] The following two methods have been applied to a conventional interrupt control device used in information processing devices like microprocessors and the like: method of vectored interrupt in which a start address of an interrupt processing program is different for each interrupt factor; and method of fixed address in which a start address of an interrupt processing program is fixed for each interrupt level.

[0004] Method of vectored interrupt is used in an 8086 microprocessor of U.S. Intel, Inc., an SH7032 microprocessor of Hitachi, Ltd. and the like. In this method of vectored interrupt, a vector number is assigned to each interrupt factor. When an interrupt occurs, a program in operation branches to an interrupt processing program. At this time, a vector table which stores the start address of each interrupt processing programs is referred to, each start address corresponding to a vector number. Normally, vector numbers are expressed by eight bits, which enables dealing with 256 interrupt factors. The vector table is 1 Kbyte long when each address is expressed by 32 bits.

[0005] When an interrupt occurs, the interrupt control device used in method of vectored interrupt generates a vector number corresponding to an interrupt factor, and reads the start address of an interrupt processing program from the vector table, the start address corresponding to the vector number. Then, the interrupt control device causes an address of a program in operation to branch to the start address of an interrupt processing program. Specifically, such a branch is controlled by hardware such as a microprogram and the like. After the branch, the interrupt processing program is executed. When the execution of the interrupt processing program is completed, the original program's operation is started again at the point at which it was interrupted by control of hardware.

[0006] For method of vectored interrupt, a programmer creates an interrupt processing program for each interrupt factor, places the first address of each program into the vector table as the start address. Then, a processing program of each interrupt factor is started by control of hardware. This method is suitable for an information processing device for executing various kinds of applications of general versatility.


[0008] Method of fixed address is used in a V810 microprocessor of Nippon Electronic Cooperation, and the like. For 16 interrupt levels, 16 start addresses of interrupt processing programs are fixed at FFFFFEn0h (n=0-Fh). In other words, 16 start addresses are fixed at every 16th bytes.

[0009] When an interrupt occurs, the interrupt control device used in method of fixed address causes a branch from an address of a program in operation to the fixed start address corresponding to an interrupt level. Specifically, such a branch is controlled by hardware. As there is no need to refer to the vector table in this method, high-speed branch can be realized compared to method of vectored interrupt.

[0010] However, the above mentioned interrupt control device used in method of vectored interrupt has the following drawbacks: hardware configuration becomes complicated and a circuit size becomes big due to generation of vector numbers and reference to the vector table, and it takes a long time for a program when an interrupt occurs. Especially, when the interrupt control device is used in an embedded microcomputer, as capacity of built-in ROM is limited, presence of the vector table reduces room for other programs. And as the number of interrupt signals when are actually used is fewer than vector numbers, there are areas which are not used in the vector table. Therefore ROM cannot be not used efficiently.

[0011] The above mentioned interrupt control device used in method of fixed address has a following drawback: after a branch to the fixed start address of an interrupt processing program is caused by hardware, it is necessary to cause one more branch from the start address of the interrupt processing program by a branch instruction, as the fixed start addresses corresponding to interrupt levels are placed at short intervals (for example, at every 16 bytes). Therefore, it takes a long interrupt processing time. If the intervals between the fixed start addresses are wide enough to allow interrupt processing programs, it is possible to speed up the interrupt processing, but ROM cannot be used efficiently.

[0012] An interrupt control device of an embedded microcomputer should satisfy the following two requests.

[0013] One is that an interrupt level, which shows priority of an interrupt signal, should be set flexibly. This is because an interrupt control device used in an embedded microcomputer is a single chip into which many different types of I/O devices are installed, the combination of I/O devices being determined based on the intended use of the microcomputer. Here, each interrupt signal from an I/O device should be treated differently according to the use of the microcomputer.

[0014] The other is that the interrupt control device should be of small chip area so as to allow the installation of many other I/O devices on the same chip.
SUMMARY OF THE INVENTION

[0015] It is an object of the present invention to provide an interrupt control device of small hardware size, which is capable of dealing with whatever combination of I/O devices flexibly and used in an imbedded microcomputer.

[0016] The object can be achieved by the following features:

(1) An interrupt control device of an embedded microcomputer including I/O devices and a processor core comprising: a program storage unit for storing interrupt processing programs, each corresponding to an interrupt level signal, in sequential areas; a start address hold unit for holding start addresses, which can be updated, of the interrupt processing programs; a level hold unit for holding an interrupt level, which can be updated, of each interrupt signal; an interrupt reception unit for, when at least one out of a plurality of interrupt signals inputted from the I/O devices becomes effective, receiving an interrupt signal of a highest interrupt level out of the inputted interrupt signals and outputting an interrupt level signal of the same interrupt level; and a control unit for controlling a branch, when the interrupt level signal is outputted, by fetching one of the start addresses which corresponds to the interrupt level signal from the start address hold unit and setting the start address in a program counter.

According to the above construction, as the interrupt signal from outside of the processor core is converted into an interrupt level signal which has an interrupt level of the level hold unit, an interrupt level of an interrupt signal from an I/O device can be set flexibly. And, as a vector table and hardware for converting a vector into an interrupt request signal becomes unnecessary, the hardware size can be minimized. Therefore, chip area of the interrupt control device for the embedded microcomputer can be minimized. It is possible to set interrupt levels of interrupt signals from I/O devices, which are customized for each intended use of the microcomputer, flexibly. And it is also possible to increase areas for installation.

(2) The start address hold unit may include: a low-order address hold unit for holding a low-order address of the start address of the interrupt processing program of each level; a constant generation unit for generating a constant to be used as a high-order address of the start address, and wherein the control unit controls the branch to the start address composed of the low-order address of the low-order address hold unit and the high-order address generated from the constant generation unit when the interrupt request signal occurs, the high-order and low-order addresses corresponding to the interrupt level of the interrupt request.

According to the above construction, it is possible to minimize the hardware size and save cost by using the same high-order address of a start address for each interrupt signal.

(3) The start address hold unit may include: a first hold unit for holding a leading address of the sequential areas of interrupt processing programs in the program storage unit as a base address; and a second hold unit for holding a relative start address expressed by a differential between the start address of each interrupt processing program and the base address, the relative start address corresponding to an interrupt level of each interrupt processing program.

wherein the control unit controls the branch to the start address when the interrupt level signal is outputted, the start address being obtained by adding the base address and the relative start address, each corresponding to the interrupt level of the interrupt request.

According to the above construction, it is possible to calculate a start address more flexibly. Moreover, if the allocated sequential storage areas of the interrupt processing programs is changed, changing the base address can cope with that. If individual interrupt processing program is changed, changing the relative start address can cope with that. Therefore, it is possible to design a program more flexibly.

(4) The interrupt control device of (1) may further comprise a fixed address generation unit for generating a start address of an interrupt processing program for a specific non-maskable interrupt level signal, and wherein the control unit controls a branch to the start address generated by the fixed address generation unit when the specific non-maskable interrupt level signal occurs.

According to the above construction, it is possible to distinguish a specific non-maskable interrupt level signal mainly used for emergency from an interrupt level signal frequently used by an application.

(5) One interrupt level signal having an interrupt level higher than a predetermined level may correspond to one interrupt processing program in the program storage unit, and a plurality of interrupt level signals, each having an interrupt level one of equal to and lower than the predetermined level, may correspond to one interrupt processing program in the program storage unit, the start address hold unit including: a first hold unit for holding the start address corresponding to the interrupt level higher than the predetermined level; and a second hold unit for holding the start address corresponding to a plurality of interrupt levels which are one of equal to and lower than the predetermined level.

According to the above construction, for an interrupt level signal of a high priority level, quickness and promptness can be guaranteed for sure. And, hardware configuration can be simplified by packing interrupt signals of low priority levels into one interrupt processing program, thereby realizing flexible inter-
rupt control depending on characteristics needed for each interrupt processing.
(6) The level hold unit may include the same number of interrupt control registers as the interrupt signals, each interrupt control register having an interrupt level setting field.
(7) The interrupt control device of (6) may further include: a first initialization unit for setting each start address of the interrupt processing programs in the start address hold unit right after the microcomputer is activated; and a second initialization unit for setting each level of the interrupt processing programs in the interrupt control registers right after the microcomputer is activated.

According to the above construction, dynamic switching between interrupt processing programs during one interrupt factor is possible. When an interrupt factor of IPT6 is a timer interrupt used for measuring intervals between buzzing times and intervals between warnings on display, it is possible to switch interrupt processing programs by storing the start address of an interrupt processing program which supports the former case or the start address of an interrupt program which supports the latter case in the start address hold unit according to necessity.

(8) The interrupt control device of (7) may further include a setting change unit for dynamically changing, when the microcomputer is in activation, at least one of the level in the interrupt control register and a start address in the start address hold unit corresponding to one of the interrupt processing programs.
(9) The program storage unit may further include a first program for setting each start address of the interrupt processing programs in the start address hold unit and a second program for setting each level of the interrupt processing programs in the interrupt control registers, wherein the first and the second programs are executed by the processor core right after the microcomputer is activated.
(10) The program storage unit may further include a third program for changing at least one of the level in the interrupt control register and the start address in the start address hold unit, each corresponding to one of the interrupt processing programs, wherein the third program is dynamically executed by processor core when the microcomputer is in activation.

DESCRIPTION OF THE PREFERRED EMBODIMENT

(Embodiment 1)

[0018] Fig. 1 is a block diagram showing configuration of a processor comprising an interrupt control device of Embodiment 1 of the present invention.
[0019] The present processor comprises ROM 101, RAM 102, register unit 103, address register 104, select register 105, select register 106, pre-fetch counter 107, incrementer 108, decode program counter 109, execution program counter 110, driver 111, instruction register 112, status register 113, decoder 114, microinstruction register 115, arithmetic logic operation unit 116, processor status word 117, operation result buffer 118, operand address register 119, select register 120, select register 121, drivers 122-130, selectors 131 and 132, internal address bus 140, internal data bus 141, internal data bus 142, interrupt signals 150, interrupt reception unit 151, constant generator 152, and interrupt control unit 154 in decoder 114.
[0020] ROM 101 and RAM 102 store instructions and data, respectively. Instruction word, address, and data are 32 bits long.
[0021] Register unit 103 is composed of a plurality of registers. In the present embodiment, they are data registers, address registers, and a stack pointer, each of them being 32 bits long.
[0022] Start address registers 104 is composed of seven registers (IRQV0-IRQV6), each corresponding to interrupt level signals 152 (level 0 - level 6). Each of IRQV0-IRQV6 holds low-order 16 bits of the start address of an interrupt processing program which corresponds to an interrupt level from level 0 - level 6. For example, IRQV2 holds low-order 16 bits of the start address of an interrupt processing program of interrupt level 2. Such start address is set by usual transfer instructions (MOV instructions) which exists at the outset of a program (main routine) or other necessary places. A predetermined high-order 16 bits are added to the start address by control of
interrupt control unit 154 when an interrupt request occurs. Then, the start address is supplied to ROM 101 as a fetch address and stored in pre-fetch counter 107 at the same time.

[0023] Selectors 105 and 106 select data from RAM 102 or data on internal address bus 104. Selector 105 outputs the selected data to start address register 104, and selector 106 outputs the selected data to register unit 103.

[0024] Pre-fetch counter 107 and incrementor 108 constitute a counter which sequentially generates addresses of instructions to be fetched according to control of decoder 114.

[0025] Decode program counter 109 holds an address of an instruction in an instruction decoding stage.

[0026] Execution program counter 110 holds an address of an instruction in an instruction execution stage.

[0027] Driver 111 outputs the address of an instruction in execution program counter 110 to internal data bus 141 according to control of decoder 114.

[0028] Instruction register 112 holds an instruction which is fetched from ROM 101.

[0029] Status register 113 holds various status flags which are necessary for decoding instructions.

[0030] Decoder 114 decodes the instruction in instruction register 112 and the contents of status register 113, and outputs microinstructions to realize the instruction. Decoder 114 comprises interrupt control unit 154 inside, which controls a branch to an interrupt processing program and return to the previous program.

[0031] Microinstruction register 115 outputs control signals so that the microinstruction from decoder 114 is realized.

[0032] Arithmetic logic operation unit 116 carries out data operation and address calculation according to control of decoder 114.

[0033] Processor status word 117 holds various flags showing internal status of a processor, including IMSK field (interrupt mask field) which designates a possible range of level interruptions.

[0034] Operation result buffer 118 holds an operation result generated by arithmetic logical operation unit 116.

[0035] Operand address register 119 holds an address of RAM 102.

[0036] Selector 120 selects and outputs one of a fetch address from incrementor 108, an operation result in operation result buffer 118, and an address in operand address register 119 to ROM 101 according to control of decoder 114.

[0037] Selector 121 selects and outputs an operation result in operation result buffer 118 or an address in operand address register 119 according to control of decoder 114.

[0038] Drivers 122-130 are three-state drivers which drive data/address according to the control signals from microinstruction register 115.

[0039] Selector 131 selects and outputs data from internal data bus 141 or internal data bus 142 to one of two input ports of arithmetic logic operation unit 116.

[0040] Selector 132 selects and outputs data from one of internal data bus 142, internal address bus 140, operation result buffer 118, and constant generator 153 to the other input port of arithmetic logic operation unit 116.

[0041] Interrupt signals 150 are seven signals (IPT0-IPT6), which post interrupt requests from various I/O devices outside of the processor to interrupt reception unit 151.

[0042] Interrupt reception unit 151 receives level interrupts, a non-maskable interrupt, and a reset interrupt, and posts them to decoder 114. One level interrupt occurs due to an interrupt signal. Interrupt signals (IPT0-IPT6) are converted to interrupt level signals (IRQ0-IRQ6). The non-maskable interrupt has a higher priority than a level interrupt, and can be received at any time. The reset interrupt has the highest priority, and this is an interrupt for doing processing necessary for reset. Input reception unit 151 comprises interrupt control registers inside, each corresponding to interrupt signals (IPT0-IPT6), and outputs an interrupt level signal according to an interrupt control register when an interrupt signal is inputted. In each interrupt control register, an interrupt level, which should occur when an interrupt signal is inputted, is set. In the present embodiment, IRQ0-IRQ6, which corresponds to IPT0-IPT6, respectively, are outputted. Interrupt reception unit 151 outputs a non-maskable interrupt signal (NMI) and a reset interrupt signal to decoder 114, without converting them.

[0043] Interrupt level signals 152 are IRQ0-IRQ6, the precedence decreasing one by one from IRQ0. Interrupt level signals 152 request interrupt processing from decoder 114.

[0044] Constant generator 153 generates 16-bit and 32-bit constants. Here, the 16-bit constant represents high-order 16 bits of the start address of processing of the level interrupt. Two 32-bit constants are generated: one is the start address of interrupt processing of NMI; and the other is the start address of interrupt processing of the reset interrupt. In the present embodiment, the 16-bit constant is 4000h, the start address of processing of NMI is set at 40000000h, and the start address of processing of the reset interrupt is set at 40000000h.

[0045] Interrupt control unit 154 is actually a part of decoder 114. When one of interrupt level signals 152 is inputted, interrupt control unit 154 issues microinstructions to control a branch to interrupt processing program.

[0046] Specifically, interrupt control unit 154 issues the following microinstructions when IRQn (n=0-6) is inputted: (1.1) a microinstruction to update the stack pointer (SP→SP); (1.2) a microinstruction to save the contents of execution program counter 110 on the stack (EXPC→(SP-4)); (1.3) a microinstruction to save the contents of processor status word 117 on the stack (PSW→SP); (1.4) a microinstruction to update IMSK field in processor status word 117 to an interrupt level of IRQn; (1.5) a microinstruction to make arithmetic logic operation unit 116 add the start address in IRQVn corresponding to the level
Input request signal IPT3 is converted to interrupt level signal IRQ3 of level 3 by interrupt reception unit 151, and outputted to decoder 114. Interrupt control unit 154 in decoder 114 issues the following microinstructions when interrupt level signal IRQ3 is inputted: (1.1) the microinstruction to update the stack pointer (SP→SP); (1.2) the microinstruction to save the contents of execution program counter 110 on the stack (EXPC→SP-4); (1.3) the microinstruction to save the contents of processor status word 117 on the stack (PSW→SP); (1.4) the microinstruction to update IMSK field in processor status word 117 to an interrupt level of IRQ3; (1.5) the microinstruction to make arithmetic logic operation unit 116 add the start address in IRQV3 corresponding to the IRQ3 and a 16-bit constant in constant generator 153; and (1.6) the microinstruction to output a result of the addition to ROM101 and store it in pre-fetch counter 107 at the same time.

Microinstruction register 115 outputs various control signals in order to realize the microinstructions mentioned above. Depending on microinstructions from (1.1)-(1.4), microinstruction register 115 outputs control signals so that stack pointer is updated or previous address and PSW are saved.

According to the microinstruction of (1.5), microinstruction register 115 outputs control signals so that arithmetic logic operation unit 116 adds low-order 16 bits (3810h) of the start address of an interrupt processing program of IRQV3, which corresponds to IRQ3, and the high-order 16 bits (IRQV3)

to ROM 101 and store it in pre-fetch counter 107 at the same time.

As mentioned above, according to the interrupt control device of the present embodiment, for each interrupt level signal 152 (IRQ0-IRQ6), the start address can be set arbitrarily by constant generator 153 and start address register 104. Therefore, interrupt processing programs for each level can be located sequentially in ROM 101, so ROM 101 can be used efficiently. Moreover, as the interrupt processing programs do not bifurcate, a branch instruction for further branching to the main part of the IRQn and the 16-bit constant in constant generator 153; and (1.6) a microinstruction to output a result of the addition to ROM 101 and store it in pre-fetch counter 107 at the same time.

Interrupt control unit 154 issues the following microinstructions when the reset interrupt is inputted: (2.1) a microinstruction to update the stack pointer (SP→SP); (2.2) a microinstruction to save the start address in IRQV3 corresponding to IRQ3 and the 2-bit constant in constant generator 153; and (2.3) a microinstruction to output a result of the addition (40003810h) is outputted to ROM101 and operation result buffer 118).
of the interrupt processing programs after a branch to
the start of the processing programs is not necessary.
So high-speed interrupt processing can be realized.

Moreover, according to the present interrupt
control device, a vector table is not necessary. So hard-
ware configuration becomes simple.

Moreover, as arithmetic logic operation unit 116
comprises more complicated address calculation func-
tion, it can calculate the start address by some opera-
tions, not by simple synthesizing as mentioned in the
present embodiment. For example, the start address of
an interrupt processing program can be calculated by 32-
bit base address + 16-bit relative address. In such a case,
a base address register is used instead of constant gen-
erator 153. The base address register holds the leading
address of an area of a plurality of interrupt processing
programs in ROM 101 as base address. And start ad-
dress register 104 holds the relative start address corre-
sponding to an interrupt level: the relative start address
is a differential between the start address of each inter-
rupt processing program and the base address. In this
way, more flexible program design can be realized.

(Embodiment 2)

Fig. 3 is a block diagram showing configuration of
a processor comprising an interrupt control device of
Embodiment 2 of the present invention. Here, the same
components that are shown in Fig. 1 of Embodiment 1
are used with the same reference numbers, except for
start address register 304, decoder 314, microinstruction
register 315, driver 324, selector 332, interrupt reception
unit 351, constant generator 353, and interrupt control
unit 354. The following explanation touches on only the
differences with Embodiment 1.

Start address registers 304 are composed of
seven registers (IRQV0-IRQV6), each corresponding to
interrupt level signals 152 (level 0 - level 7), like start
address registers 104 of Embodiment 1. However, one
point is different: each of IRQV0-IRQV6 holds 32 bits,
not the low-order 16 bits, of the start address of an inter-
rupt processing program which corresponds to an inter-
rupt level.

Decoder 314 is the same as decoder 114 of
Embodiment 1 except that it comprises interrupt control
unit 354 inside.

Microinstruction register 315 outputs control
signals so that a microinstruction from decoder 314 is
realized.

Driver 324 receives the 32-bit start address
from start address register 304 and outputs it to selector
120.

Selector 332 is the same as selector 132 of Em-
bodyment 1 except that constant generator 353 takes the
place of constant generator 153.

Input reception unit 351 is the same as input
reception unit 151 of Embodiment 1. But in the present
embodiment, the contents of control registers are ex-
plained in detail. Fig. 4 shows an example of bit con-
figuration of the control registers. In Fig. 4, ICR0-ICR6 re-
present seven control registers, each corresponding to in-
terrupt signals (IPT0-IPT6), respectively. Bits L2-L0 rep-
resent a level field in which priority level of each inter-
rupt signal is set, and a bit IE represents an interrupt
enable bit which shows whether an interrupt is allowed.
The rest of the bits 0-11 are not used in the present em-
bodyment. When an interrupt signal (IPT0) is inputted to
interrupt reception unit 351 and the interrupt enable bit
of a corresponding interrupt control register (ICR0)
shows that an interrupt is allowed, interrupt reception unit
351 outputs interrupt level signal 152 (one of IRQ0-
IRQ6), whose level is set in the level field.

Constant generator 353 is the same as constant
generator 153 of Embodiment 1 except for one thing:
constant generator 353 does not generate the 16-bit con-
stant.

Interrupt control unit 354 is actually a part of
decoder 114. When one of interrupt level signals 152 is
inputted, interrupt control unit 354 issues microinstruc-
tions in order to control a branch to an interrupt process-
ing program.

Specifically, interrupt control unit 354 issues the
following microinstructions when IRQn is inputted from
interrupt reception unit 351: (4.1) a microinstruction to
update the stack pointer (SP−→SP); (4.2) a microin-
struction to save the contents of execution program coun-
ter 110 on the stack (EFP−→(SP-4)); (4.3) a microin-
struction to save the contents of processor status word
117 on the stack (PSW−→SP); (4.4) a microinstruction to
update IMSK field in processor status word 117 to an
interrupt level of IRQn; and (4.5) a microinstruction to
update IRQVn corresponding to the level of the IRQn to
ROM 101 through selector 120 and store it in pre-fetch
counter 107 at the same time. When the reset interrupt
is inputted, the same microinstructions as (2.1) and (2.2)
of Embodiment 1 are issued. When the NMI is inputted,
the same microinstructions as (3.3)-(3.5) of Embodiment
1 are issued. The number of microinstructions from (4.1)-(4.5) is not necessarily fixed as mentioned above.
So long as the contents are reserved, any other number
is possible and they can even be consolidated into one
microinstruction.

The following explanation touches on opera-
tions of the input control device of Embodiment 2 of the
present invention.

Interrupt control registers ICR0-ICR6 and start
address registers IRQV0-IRQV6 are set as follows.

Fig. 5A is for interrupt control registers ICR0-
ICR6. Fig. 5B is for start address registers IRQV0-IRQV6.
Fig. 5C is for a program needed for their setting.

Fig. 5A shows ICR0 is set at level 0, ICR1 at
level 1, ICR2-ICR5 at level 2, and ICR6 at level 3. In other
words, level 0 is assigned to IPT0, level 1 to IPT1, and
level 2 to IPT2-IPT5, and level 3 to IPT6. In this case,
interrupt processing program of level 0 corresponds to
IPT0, and interrupt processing program of level 1 corre-
responds to IPT1. And interrupt processing program of level 2 corresponds to IPT2-IPT5. This example case is created on the assumption that a high-speed communication I/O device is connected to the processor. Here, reception processing by which each data received by the communication I/O device is transferred to the processor, and transmission processing by which data to be transmitted from the communication I/O device is transferred to the communication I/O device are realized by interrupt processing. It is essential that such reception processing and transmission processing be executed quickly and promptly so as to keep up with the communication speed (which is to say, the interrupt processing must be completed in the intervals between the reception and transmission of such data). For such a set up, the reception interrupt signal and transmission interrupt signal are set at IPT0 and IPT1 respectively, which sets their interrupt levels at 0 and 1. When IPT2-IPT6 are for interrupt processing which does not require quickness and promptness, they can be set at the same level or at any levels arbitrarily as shown in Fig. 5A.

[0077] Fig. 5B shows interrupt processing programs of level 0-level 3 have the size of 8 Kbytes (2000h bytes), 4 Kbytes (1000h), 16 Kbytes (4000h), and 1 Kbyte (400h), respectively. For example, when the start address of an interrupt processing program of level 0 is set at 40000010h, IRQV0 is set at 40000010h. Then the interrupt processing program is located between 40000010h and 4000200Fh. In order to locate an interrupt processing program of level 1 right after that, IRQV1 should be set at 40002010h. An interrupt processing program of level 2 should correspond to each of interrupt signals IPT2-IPT5, and IRQV2 should be set at 40003010h. Likewise, IRQV3 should be set at 40007010h, so that the interrupt processing programs are located without intervals.

[0078] Fig. 5C shows an example of a program for the above setting. ICR0-ICR6 and IRQV0-IRQV3 are set by transfer instructions (MOV instructions). IRQV4-IRQV6 are not set, as they are not used.

[0079] Next explanation touches on a branch to an interrupt processing program of a level interrupt.

[0080] Here, ICR0-ICR6 and IRQV0-IRQV3 are set as shown in Fig. 5, and IPT0 is inputted.

[0081] Interrupt signal IPT0 is converted to, by interrupt reception unit 351, interrupt level signal IRQ0 of level 0 designated by a level field of ICR0. Then, IRQ0 is outputted to decoder 314. Interrupt control unit 354 in decoder 314 issues the following microinstructions when IRQ0 is inputted: (4.1) a microinstruction to update the stack pointer (SP-8→sup); (4.2) a microinstruction to save the contents of decode program counter 109 on the stack (EXPC→(SP-4)); (4.3) a microinstruction to save the contents of processor status word 117 on the stack (PSW→SP); (4.4) a microinstruction to update IMSK field in processor status word 117 to the level of the IRQ0; and (4.5) a microinstruction to output IRQV0, which corresponds to the IRQ0, to ROM 101 and store it in prefetch counter 107 at the same time.

[0082] Microinstruction register 115 outputs various control signals so that the microinstructions mentioned above are realized. Depending on a microinstruction from (4.1) - (4.5), microinstruction register 115 outputs control signals so that stack pointer is updated or previous address and PSW are saved.

[0083] According to the microinstruction of (4.5), microinstruction register 115 outputs control signals so that the 32-bit start address (40000010h) of an interrupt processing program is outputted to ROM 101 as a fetch address from IRQV0, which corresponds to the IRQ0 (start address register 304-selector 120→ROM 101) and it is stored in pre-fetch counter 107 at the same time (selector 120→pre-fetch counter 107). Then, a program branches to 40000010h, and an interrupt processing program of level 0 stored between 40000010h and 4000200Fh in ROM 101 is executed.

[0084] Unlike the operations mentioned above, interrupt levels can be changed dynamically. Here is an example of Fig. 5A. In order to assign higher priority to IPT6 than IPT2-IPT5 when interrupt processing due to IPT0 and IPT1 is completed and IPT0 and IPT1 are not used anymore, level field of ICR6 should be set at 0 again, and IRQV0 should be set at the start address (40007010h) of an interrupt processing program of level 6 again. Such setting can also be realized by transfer instructions as shown in Fig. 5C.

[0085] If interrupt processing due to IPT0 and IPT1 should occur after that, original setting should be restored.

[0086] Moreover, dynamic switching between interrupt processing programs during one interrupt factor is possible. When an interrupt factor of IPT6 is a timer interrupt used for measuring intervals between buzzing times of a buzzer and intervals between warnings on display, it is possible to switch interrupt processing programs by storing the start address of an interrupt processing program for the former case or the start address of an interrupt program for the latter case in IRQV6 according to necessity.

[0087] As mentioned above, when interrupt signals 150 are inputted, the interrupt control device of the present embodiment causes a branch to the start address given in start address register 304 according to a level designated by the interrupt control register. As the start address can be set arbitrarily, the interrupt processing programs for each level can be located in sequential areas in ROM 101, enabling more effective use of ROM 101. Moreover, as the interrupt processing programs do not bifurcate, a branch instruction for further branching to the main part of the interrupt processing programs after a branch to the start of the processing programs is not necessary. So high-speed interrupt processing can be realized. Moreover, priority levels of interrupt signals 150 can be set arbitrarily. A high level is assigned to an interrupt factor, which requires quickness and promptness of interrupt processing, and low levels are assigned to interrupt factors which do not require neither of them.
Therefore, flexibility of interrupt control increases.

According to the present interrupt control device, the vector table is not needed. Therefore, its hardware configuration becomes simple.

Moreover, instead of start address registers 104 and 304 in the above embodiments, an I/O register can be used. In such a case, the I/O register is mapped into a memory area like ROM 101 and RAM 102.

Moreover, in Embodiments 1 and 2, the start address is fixed by constant generators 153 and 353 when the reset interrupt and the NMI occur. But IRQV for the reset interrupt and IRQV for the NMI can be set and treated in the same manner as the level interrupt.

Claims

1. An interrupt control device of an embedded microcomputer including I/O devices and a processor core comprising:

   program storage means (101) for storing interrupt processing programs, each corresponding to an interrupt level signal (152), in sequential areas;
   start address hold means (104) for holding start addresses, which can be updated, of the interrupt processing programs;
   level hold means for holding an interrupt level, which can be updated, of each interrupt signal;
   interrupt reception means for, when at least one out of a plurality of interrupt signals (150) inputted from the I/O devices becomes effective, receiving an interrupt signal of a highest interrupt level out of the inputted interrupt signals (150) and outputting an interrupt level signal (152) of the same interrupt level;
   control means (154) for controlling a branch, when the interrupt level signal is outputted, by fetching one of the start addresses which corresponds to the interrupt level signal from the start address hold means (104) and setting the start address in a program counter,

   wherein the start address hold means includes:

   low-order address hold means for holding a low-order address of the start address of the interrupt processing program of each level;
   constant generation means (153) for generating a constant to be used as a high-order address of the start address, and

   wherein the control means (154) controls the branch to the start address composed of the low-order address of the low-order address hold means and the high-order address generated from the constant generation means when the interrupt request signal occurs, the high-order and low-order addresses corresponding to the interrupt level of the interrupt request.

2. The interrupt control device of claim 1, wherein the start address hold means includes:

   first hold means for holding a leading address of the sequential areas of interrupt processing programs in the program storage means as a base address; and
   second hold means for holding a relative start address expressed by a differential between the start address of each interrupt processing program and the base address, the relative start address corresponding to an interrupt level of each interrupt processing program,

   wherein the control means controls the branch to the start address when the interrupt level signal is outputted, the start address being obtained by adding the base address and the relative start address, each corresponding to the interrupt level of the interrupt request.

3. The interrupt control device of claim 1, further comprising:

   fixed address generation means for generating a start address of an interrupt processing program for a specific non-maskable interrupt level signal, and

   wherein the control means controls a branch to the start address generated by the fixed address generation means when the specific non-maskable interrupt level signal occurs.

4. The interrupt control device of claim 1, wherein one interrupt level signal having an interrupt level higher than a predetermined level corresponds to one interrupt processing program in the program storage means (101), and a plurality of interrupt level signals, each having an interrupt level one of equal to and lower than the predetermined level, correspond to one interrupt processing program in the program storage means, and

   wherein the start address hold means (104) includes:

   first hold means for holding the start address corresponding to the interrupt level higher than the predetermined level; and
   second hold means for holding the start address corresponding to a plurality of interrupt levels which are one of equal to and lower than the predetermined level.

5. The interrupt control device of claim 1, wherein the
level hold means includes the same number of interrupt control registers as the interrupt signals, each interrupt control register having an interrupt level setting field.

6. The interrupt control device of claim 6 further including:

first initialisation means for setting each start address of the interrupt processing programs in the start address hold means right after the microcomputer is activated; and

second initialisation means for setting each level of the interrupt processing programs in the interrupt control registers right after the microcomputer is activated.

7. The interrupt control device of claim 7 further including setting change means for dynamically changing, when the microcomputer in activation, at least one of the level in the interrupt control register and a start address in the start address hold means corresponding to one of the interrupt processing programs.

8. The interrupt control device of claim 6, wherein the program storage means (101) further includes a first program for setting each start address of the interrupt processing programs in the start address hold means and a second program for setting each level of the interrupt processing programs in the interrupt control registers, wherein the first and the second programs are executed by the processor core right after the microcomputer is activated.

9. The interrupt control device of claim 9, wherein the program storage means (101) further includes a third program for changing at least one of the level in the interrupt control register and the start address in the start address hold means, each corresponding to one of the interrupt processing programs, wherein the third program is dynamically executed by processor core when the microcomputer in activation.

Patentansprüche

1. Interrupt-Steuervorrichtung eines eingebetteten Mikrocomputers, der Ein/Ausgabevorrichtungen und einen Prozessorkern enthält, wobei die Vorrichtung umfasst:

   eine Programmspeichereinrichtung (101) zum Speichern von Interrupt-Verarbeitungsprogrammen, die jeweils einem Interrupt-Pegelsignal (152) entsprechen, in aufeinanderfolgenden Bereichen;

   eine Startadressen-Halteeinrichtung (104) zum Halten von Startadressen der Interrupt-Verarbeitungsprogramme, die aktualisiert werden können;

   eine Pegel-Halteeinrichtung zum Halten eines Interrupt-Pegels jedes Interrupt-Signals, der aktualisiert werden kann;

   eine Interrupt-Empfangseinrichtung, die, wenn wenigstens eines einer Vielzahl von Interrupt-Signalen (150), die über die Ein-/Ausgabevorrichtung eingegangen werden, wirksam wird, ein Interrupt-Signal eines höchsten Interrupt-Pegels von den eingegebenen Interrupt-Signalen (150) empfängt und ein Interrupt-Pegelsignal (152) des gleichen Interrupt-Pegels ausgibt;

   eine Steuereinrichtung (154), mit der eine Verzweigung gesteuert wird, wenn das Interrupt-Pegelsignal ausgegeben wird, indem eine der Startadressen, die dem Interrupt-Pegelsignal entspricht, aus der Startadressen-Halteeinrichtung (104) abgerufen und die Startadresse in einem Programmzähler eingestellt wird, wobei die Startadressen-Halteeinrichtung enthält:

   eine Halteeinrichtung für Adressen niedriger Ordnung, die eine Adresse niedriger Ordnung der Startadresse des Interrupt-Verarbeitungsprogramms jeder Stufe hält;

   eine Konstanten-Erzeugungseinrichtung (153) zum Erzeugen einer Konstante, die als eine Adresse hoher Ordnung der Startadresse zu verwenden ist, und

   wobei die Steuereinrichtung (154) die Verzweigung zu der Startadresse, die aus der Adresse niedriger Ordnung der Halteeinrichtung für Adressen niedriger Ordnung und der Adresse hoher Ordnung, die von der Konstanten-Erzeugungseinrichtung erzeugt wird, zusammengesetzt ist, wenn das Interrupt-Anforderungssignal auftritt, wobei die Adresse hoher Ordnung und die Adresse niedriger Ordnung dem Interrupt-Pegel der Interrupt-Anforderung entsprechen.

2. Interrupt-Steuervorrichtung nach Anspruch 1, wobei die Startadressen-Halteeinrichtung enthält:

   eine erste Halteeinrichtung, mit der eine führende Adresse der aufeinanderfolgenden Bereiche von Interrupt-Verarbeitungsprogrammen in der Programm-Speichereinrichtung als eine Basisadresse gehalten wird; und

   eine zweite Halteeinrichtung, mit der eine relative Startadresse gehalten wird, die durch eine Differenz zwischen der Startadresse jedes Interrupt-Verarbeitungsprogramms und der Basisadresse ausgedrückt wird, wobei die relative
Interrupt-Verarbeitungsprogramms entspricht,

wobei die Steuereinrichtung die Verzweigung zu der Startadresse steuert, wenn das Interrupt-Pegelsignal ausgegeben wird, und die Startadresse ermittelt wird, indem die Basisadresse und die relative Startadresse addiert werden, die jeweils dem Interrupt-Pegel der Interrupt-Anforderung entsprechen.

3. Interrupt-Steuervorrichtung nach Anspruch 1, die des Weiteren umfasst:

   eine Einrichtung zum Erzeugen fester Adressen, mit der eine Startadresse eines Interrupt-Verarbeitungsprogramms für ein spezifisches nicht maskierbares Interrupt-Pegelsignal erzeugt wird, und

wobei die Steuereinrichtung eine Verzweigung zu der Startadresse steuert, die durch die Einrichtung zum Erzeugen fester Adressen erzeugt wird, wenn das spezifische nicht maskierbare Interrupt-Pegelsignal auftritt.

4. Interrupt-Steuervorrichtung nach Anspruch 1, wobei ein Interrupt-Pegelsignal mit einem Interrupt-Pegel, der höher ist als ein vorgegebener Pegel, einem Interrupt-Verarbeitungsprogramm in der Programmspeichereinrichtung (101) entspricht und eine Vielzahl von Interrupt-Pegelsignalen, die jeweils einen Interrupt-Pegel haben, der entweder dem vorgegebenen Pegel gleich ist oder darunter liegt, einem Interrupt-Verarbeitungsprogramm in der Programmspeichereinrichtung entsprechen, und

wobei die Startadressen-Halteeinrichtung (104) enthält:

   eine erste Halteeinrichtung, die die Startadresse hält, die dem Interrupt-Pegel entspricht, der über dem vorgegebenen Pegel liegt; und
   eine zweite Halteeinrichtung, die die Startadresse hält, die einer Vielzahl von Interrupt-Pegeln entspricht, die entweder dem vorgegebenen Pegel gleich sind oder darunter liegen.

5. Interrupt-Steuervorrichtung nach Anspruch 1, wobei die Pegel-Halteeinrichtung die gleiche Anzahl von Interrupt-Steuerregistern enthält wie die Interrupt-Signale, und jedes Interrupt-Steuerregister ein Feld zum Einstellen des Interrupt-Pegels aufweist.

6. Interrupt-Steuervorrichtung nach Anspruch 6, die des Weiteren enthält:

   eine erste Initialisierungseinrichtung, mit der jede Startadresse der Interrupt-Verarbeitungsprogramme in der Startadressen-Halteeinrichtung eingestellt wird, nachdem der Mikrocomputer aktiviert ist; und
   eine zweite Initialisierungseinrichtung, mit der jeder Pegel der Interrupt-Verarbeitungsprogramme in den Interrupt-Steuerregistern eingestellt wird, nachdem der Mikrocomputer aktiviert ist.

7. Interrupt-Steuervorrichtung nach Anspruch 7, die des Weiteren eine Einstellungs-Änderungseinrichtung enthält, mit der, wenn der Mikrocomputer aktiviert wird, dynamisch wenigstens des Pegel in dem Interrupt-Steuerregister oder eine Startadresse in der Startadressen-Halteeinrichtung geändert wird, die einem der Interrupt-Verarbeitungsprogramme entsprechen.


9. Interrupt-Steuervorrichtung nach Anspruch 9, wobei die Programmspeichereinrichtung (101) des Weite- ren ein drittes Programm zum Ändern wenigstens des Pegels in dem Interrupt-Steuerregister oder der Startadresse in der Startadressen-Halteeinrichtung enthält, die jeweils einem der Interrupt-Verarbeitungsprogramme entsprechen, wobei das dritte Programm durch den Prozessorkern dynamisch ausgeführt wird, wenn der Mikrocomputer aktiviert wird.

**Revendications**

1. Dispositif de commande d'interruption d'un microordinateur intégré incluant des dispositifs d'entrée/sortie (E/S) et un coeur de processeur comprenant :

   des moyens de stockage de programme (101) pour stocker des programmes de traitement d'interruption, chacun correspondant à un signal de niveau d'interruption (152), dans des zones séquentielles ;
   des moyens de maintien d'adresse de départ (104) pour maintenir des adresses de départ, qui peuvent être mises à jour, des programmes de traitement d'interruption ;
   des moyens de maintien de niveau pour maintenir un niveau d'interruption, qui peut être mis
à jour, de chaque signal d'interruption ;
des moyens de réception d'interruption pour,
lorqu'au moins un parmi une pluralité de signaux d'interruption entrés depuis les dispositifs E/S devient effectif, recevoir un signal d'interruption d'un niveau d'interruption le plus élevé des signaux d'interruption entrés et sortir un signal de niveau d'interruption du même niveau d'interruption ;
des moyens de commande pour comman-
der un branchement, lorsque le signal de niveau d'interruption est sorti, en extrayant une des adresses de départ qui correspondent au signal de niveau d'interruption des moyens de maintien d'adresse de départ et en établissant l'adresse de départ dans un compteur de pro-
gramme,
dans lequel les moyens de maintien d'adresse de départ incluent :
des moyens de maintien d'adresse de poids faible pour maintenir une adresse de poids faible de l'adresse de départ du programme de traitement d'interruption de chaque niveau ;
des moyens de génération de constante pour générer une constante devant être utilisée comme une adresse de poids fort de l'adresse de départ, et
dans lequel les moyens de commande commandent le branchement à l'adresse de départ composée de l'adresse de poids faible des moyens de maintien d'adresse de poids faible et de l'adresse de poids fort générée à partir des moyens de génération de constante lorsque le signal de demande d'interruption se produit, les adresses de poids fort et de poids faible correspondant au niveau d'interruption de la demande d'interruption.

2. Dispositif de commande d'interruption selon la revendication 1, dans lequel les moyens de maintien d'adresse de départ incluent :
des premiers moyens de maintien pour mainte-
nir une adresse en tête des zones séquentielles de programmes de traitement d'interruption dans les moyens de stockage de programme comme une adresse de base ; et
des seconds moyens de maintien pour mainte-
nir une adresse de départ relative exprimée par une différentielle entre l'adresse de départ de chaque programme de traitement d'interruption et l'adresse de base, l'adresse de départ relative correspondant à un niveau d'interruption de chaque programme de traitement d'interruption,
dans lequel les moyens de commande commandent le branchement à l'adresse de départ lorsque le si-
gnal de niveau d'interruption est sorti, l'adresse de départ étant obtenue en ajoutant l'adresse de base et l'adresse de départ relative, chacune correspondant au niveau d'interruption de la demande d'interruption.

3. Dispositif de commande d'interruption selon la revendication 1, comprenant en outre :
des moyens de génération d'adresse fixe pour générer une adresse de départ d'un programme de traitement d'interruption pour un signal de niveau d'interruption non masquable spécifique, et
dans lequel les moyens de commande commandent un branchement à l'adresse de départ générée par les moyens de génération d'adresse fixe lorsque le signal de niveau d'interruption non masquable spécifique se produit.

4. Dispositif de commande d'interruption selon la revendication 1, dans lequel un signal de niveau d'interruption ayant un niveau d'interruption supérieur à un niveau prédéterminé correspond à un programme de traitement d'interruption dans les moyens de stockage de programme, et une pluralité de signaux de niveau d'interruption, chacun ayant un niveau d'interruption entre égal et inférieur au niveau prédéterminé, correspondent à un programme de traitement d'interruption dans les moyens de stockage de programme, et
dans lequel les moyens de maintien d'adresse de départ incluent :
des premiers moyens de maintien pour mainte-
nir l'adresse de départ correspondant au niveau d'interruption supérieur au niveau prédéterminé ; et
des seconds moyens de maintien pour mainte-
nir l'adresse de départ correspondant à une plu-
ralité de niveaux d'interruption qui sont un entre égal et inférieur au niveau prédéterminé.

5. Dispositif de commande d'interruption selon la revendication 1, dans lequel les moyens de maintien de niveau incluent le même nombre de registres de commande d'interruption que les signaux d'interruption, chaque registre de commande d'interruption ayant un champ d'établissement de niveau d'interruption.

6. Dispositif de commande d'interruption selon la revendication 6 incluant en outre :
des premiers moyens d'initialisation pour établir chaque adresse de départ des programmes de
traitement d'interruption dans les moyens de maintien d'adresse de départ juste après l'activation du micro-ordinateur ; et des seconds moyens d'initialisation pour établir chaque niveau des programmes de traitement d'interruption dans les registres de commande d'interruption juste après l'activation du micro-ordinateur.

7. Dispositif de commande d'interruption selon la revendication 7 incluant en outre des moyens de changement d'établissement pour changer dynamiquement, lorsque le micro-ordinateur est activé, au moins un entre le niveau dans le registre de commande d'interruption et une adresse de départ dans les moyens de maintien d'adresse de départ correspondant à un des programmes de traitement d'interruption.

8. Dispositif de commande d'interruption selon la revendication 6, dans lequel les moyens de stockage de programme (101) incluent en outre un premier programme pour établir chaque adresse de départ des programmes de traitement d'interruption dans les moyens de maintien d'adresse de départ et un deuxième programme pour établir chaque niveau des programmes de traitement d'interruption dans les registres de commande d'interruption, dans lequel les premier et deuxième programmes sont exécutés par le cœur de processeur juste après l'activation du micro-ordinateur.

9. Dispositif de commande d'interruption selon la revendication 9, dans lequel les moyens de stockage de programme (101) incluent en outre un troisième programme pour changer au moins un entre le niveau dans le registre de commande d'interruption et l'adresse de départ dans les moyens de maintien d'adresse de départ, chacun correspondant à un des programmes de traitement d'interruption, dans lequel le troisième programme est exécuté dynamiquement par le cœur de processeur lorsque le micro-ordinateur est activé.
### Fig. 2A

<table>
<thead>
<tr>
<th>IRQn</th>
<th>program capacity of interrupt processing</th>
<th>start address</th>
<th>IRQVn</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8kbyte(2000h)</td>
<td>40000010h</td>
<td>0010h</td>
<td>40000010h~4000200Fh</td>
</tr>
<tr>
<td>1</td>
<td>4kbyte(1000h)</td>
<td>40002010h</td>
<td>2010h</td>
<td>40002010h~4000300Fh</td>
</tr>
<tr>
<td>2</td>
<td>2kbyte(0800h)</td>
<td>40003010h</td>
<td>3010h</td>
<td>40003010h~4000380Fh</td>
</tr>
<tr>
<td>3</td>
<td>1kbyte(0400h)</td>
<td>40003810h</td>
<td>3810h</td>
<td>40003810h~40003C0Fh</td>
</tr>
<tr>
<td>4</td>
<td>0.5kbyte(0200h)</td>
<td>40003010h</td>
<td>3B10h</td>
<td>40003C10h~40003E0Fh</td>
</tr>
<tr>
<td>5</td>
<td>0.5kbyte(0200h)</td>
<td>40003E10h</td>
<td>3D10h</td>
<td>40003E10h~4000410Fh</td>
</tr>
<tr>
<td>6</td>
<td>0.5kbyte(0200h)</td>
<td>40004110h</td>
<td>3F10h</td>
<td>40004110h~4000430Fh</td>
</tr>
</tbody>
</table>

### Fig. 2B

MOV # x'0010',D1 ; immediate data 0010h → D1
MOV D1,IRQV0 ; D1 → IRQV0
MOV # x'2010',D1 ; immediate data 2010h → D1
MOV D1,IRQV1 ; D1 → IRQV1
MOV # x'3010',D1 ; immediate data 3010h → D1
MOV D1,IRQV2 ; D1 → IRQV2
MOV # x'3810',D1 ; immediate data 3810h → D1
MOV D1,IRQV3 ; D1 → IRQV3
MOV # x'3C10',D1 ; immediate data 3C10h → D1
MOV D1,IRQV4 ; D1 → IRQV4
MOV # x'3E10',D1 ; immediate data 3E10h → D1
MOV D1,IRQV5 ; D1 → IRQV5
MOV # x'4110',D1 ; immediate data 4110h → D1
MOV D1,IRQV6 ; D1 → IRQV6
Fig. 5A

<table>
<thead>
<tr>
<th>ICRn</th>
<th>level field</th>
<th>IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICR0</td>
<td>000b</td>
<td>1</td>
</tr>
<tr>
<td>ICR1</td>
<td>001b</td>
<td>1</td>
</tr>
<tr>
<td>ICR2</td>
<td>010b</td>
<td>1</td>
</tr>
<tr>
<td>ICR3</td>
<td>010b</td>
<td>1</td>
</tr>
<tr>
<td>ICR4</td>
<td>010b</td>
<td>1</td>
</tr>
<tr>
<td>ICR5</td>
<td>010b</td>
<td>1</td>
</tr>
<tr>
<td>ICR6</td>
<td>011b</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 5B

<table>
<thead>
<tr>
<th>IRQn</th>
<th>program capacity of interrupt processing</th>
<th>IRQVn</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8kbyte(2000h)</td>
<td>40000010h</td>
<td>40000010h〜4000200Fh</td>
</tr>
<tr>
<td>1</td>
<td>4kbyte(1000h)</td>
<td>40002010h</td>
<td>40002010h〜4000300Fh</td>
</tr>
<tr>
<td>2</td>
<td>16kbyte(4000h)</td>
<td>40003010h</td>
<td>40003010h〜4000700Fh</td>
</tr>
<tr>
<td>3</td>
<td>1kbyte(0400h)</td>
<td>40007010h</td>
<td>40007010h〜4000740Fh</td>
</tr>
</tbody>
</table>

Fig. 5C

MOV #x' 00001000',D1 ; immediate data 00001000h → D1
MOV D1,ICR0   ; D1 → ICR0
MOV #x' 00003000',D1 ; immediate data 00003000h → D1
MOV D1,ICR1   ; D1 → ICR1
MOV #x' 00005000',D1 ; immediate data 00005000h → D1
MOV D1,ICR2   ; D1 → ICR2
MOV D1,ICR3   ; D1 → ICR3
MOV D1,ICR4   ; D1 → ICR4
MOV D1,ICR5   ; D1 → ICR5
MOV #x' 00007000',D1 ; immediate data 00007000h → D1
MOV D1,ICR6   ; D1 → ICR6
MOV #x' 40000010',D1 ; immediate data 40000010h → D1
MOV D1,IRQV0  ; D1 → IRQV0
MOV #x' 40002010',D1 ; immediate data 40002010h → D1
MOV D1,IRQV1  ; D1 → IRQV1
MOV #x' 40003010',D1 ; immediate data 40003010h → D1
MOV D1,IRQV2  ; D1 → IRQV2
MOV #x' 40003810',D1 ; immediate data 40003810h → D1
MOV D1,IRQV3  ; D1 → IRQV3

18
REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

• EP 0637802 A [0007]