(12) UK Patent Application (19) GB (11) 2626447 (13) A

(43) Date of Reproduction by UK Office 24.07.2024

(21) Application No: 2401725.3

(22) Date of Filing: 06.07.2022

Date Lodged: 08.02.2024

(30) Priority Data:
(31) 2021153025 (32) 21.09.2021 (33) JP

(86) International Application Data:
PCT/JP2022/026785 Ja 06.07.2022

(87) International Publication Data:
WO2023/047762 Ja 30.03.2023

(71) Applicant(s):
Hitachi Ltd
(Incorporated in Japan)
6-6 Marunouchi 1-chome, Chiyoda-ku, Tokyo, 1008280, Japan

(72) Inventor(s):
Kotaro Shimamura
Teruaki Sakata
Yuki Tanaka

(74) Agent and/or Address for Service:
Mewburn Ellis LLP
Aurora Building, Counterslip, Bristol, BS1 6BX, United Kingdom

(51) INT CL:
G06F 13/36 (2006.01) *G06F 12/04* (2006.01)
G06F 13/38 (2006.01) *G06F 13/40* (2006.01)

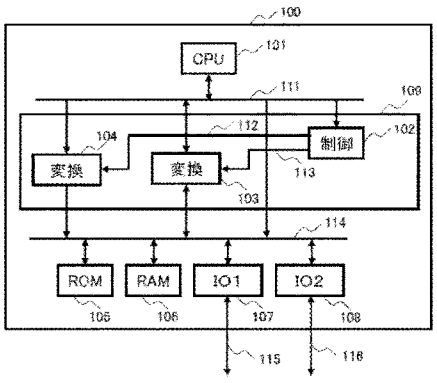
(56) Documents Cited:
WO 2010/039794 A1 JP 200003304 A
JP 8202646

(58) Field of Search:
INT CL G06F
Other: Public JP utililty model applns (examined 1922-1996),(unexamined 1971-2022); JP utility models (regst specs 1996-2022),(public regst applns 1994-2022

(54) Title of the Invention: **Processor and endian conversion method**
Abstract Title: **Processor and endian conversion method**

(57) Provided is a means of preventing, in a processor having a CPU, a plurality of peripheral circuits, and an endian conversion circuit provided between the CPU and the plurality of peripheral circuits, a written value from being interpreted as a value different from that intended by a software creator when the peripheral circuits interpret the written value as a multi-byte value, to facilitate the creation of software. The endian conversion circuit has a table indicating a relationship between addresses and types of connection destinations, and a data conversion circuit and a byte-enable conversion circuit in the endian conversion circuit are controlled separately by using type information retrieved from the table through use of the address of the access destination.

[図1]



102 Control
103, 104 Conversion

GB 2626447 A

FIG. 1

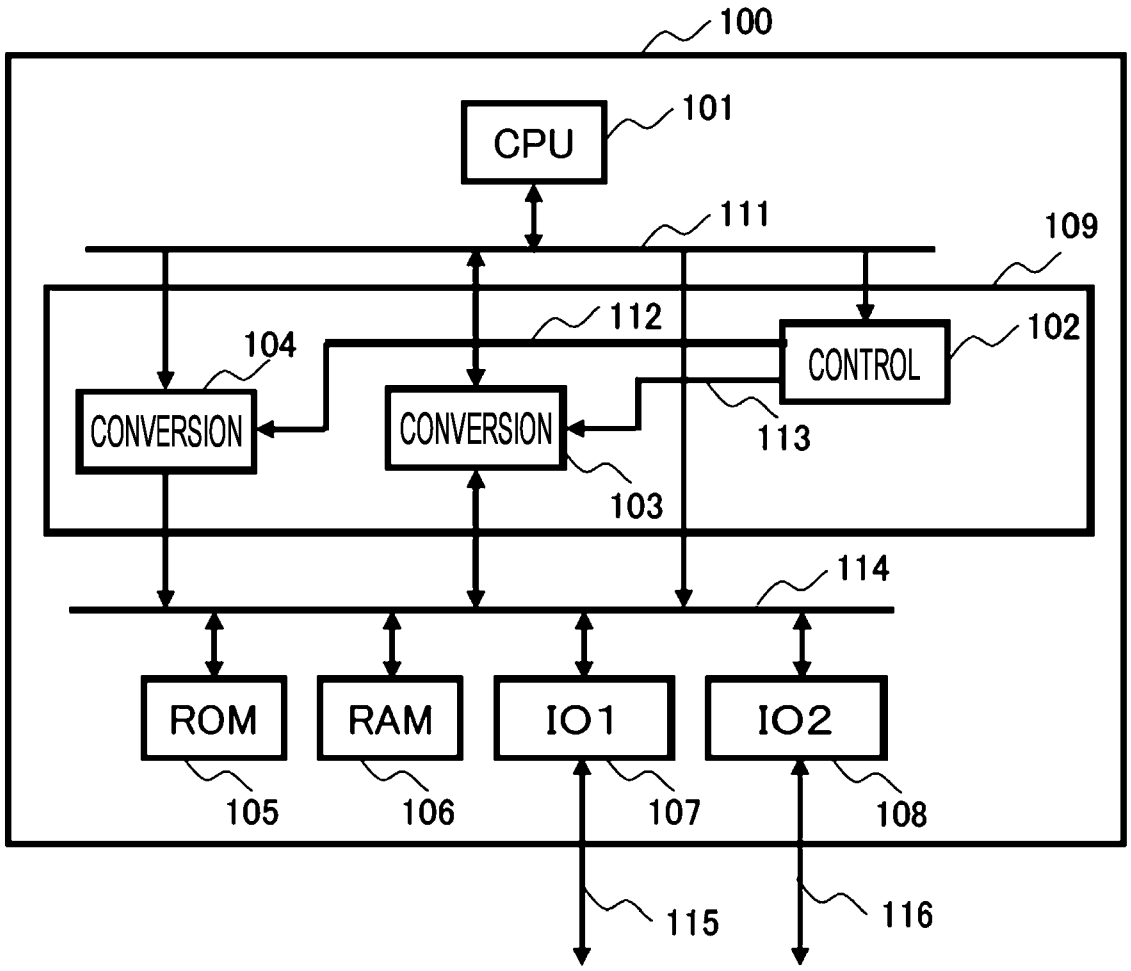

100
CPU
101
111
109
112
104
CONTROL
102
CONVERSION
CONVERSION
113
103
114
ROM
RAM
IO1
IO2
105
106
107
108
115
116

FIG. 2

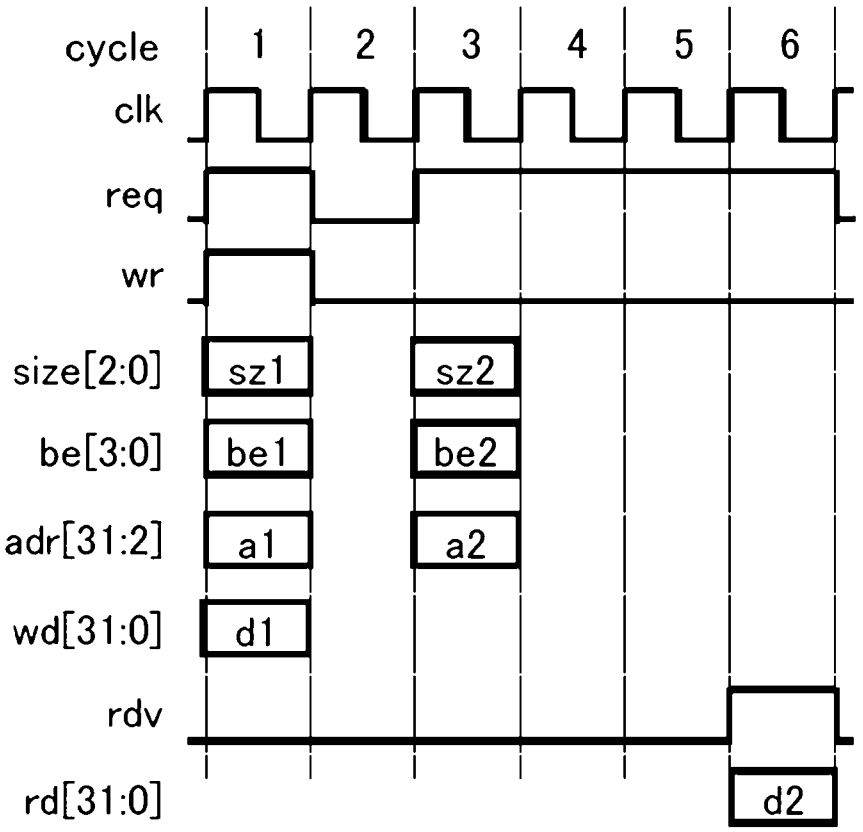


FIG. 3A

Little Endian

ACCESS TYPE		BUS (be[3:0])
SIZE	ADDRESS	
1	4n	0001
1	4n+1	0010
1	4n+2	0100
1	4n+3	1000
2	4n	0011
2	4n+2	1100
4	4n	1111

FIG. 3B

Big Endian

ACCESS TYPE		BUS (be[3:0])
SIZE	ADDRESS	
1	4n	1000
1	4n+1	0100
1	4n+2	0010
1	4n+3	0001
2	4n	1100
2	4n+2	0011
4	4n	1111

FIG. 4A

AIT (ATTRIBUTE INFORMATION TABLE)

BUS 111(adr)	Endian	TYPE	TARGET
00000000-EFFFFFFF	1	0	ROM, RAM
F0000000-F7FFFFFF	1	1	IO1
F8000000-FFFFFFFF	0	1	IO2

FIG. 4B

CSGR (CONTROL SIGNAL GENERATION RULE)

Endian	TYPE	BUS 111 (size)	SIGNAL LINE 112	SIGNAL LINE 113	TARGET
0	-	-	0	00	IO2
1	0	-	1	01	ROM, RAM
1	1	001	1	01	IO1
1	1	010	1	10	IO1
1	1	100	1	00	IO1

FIG. 5

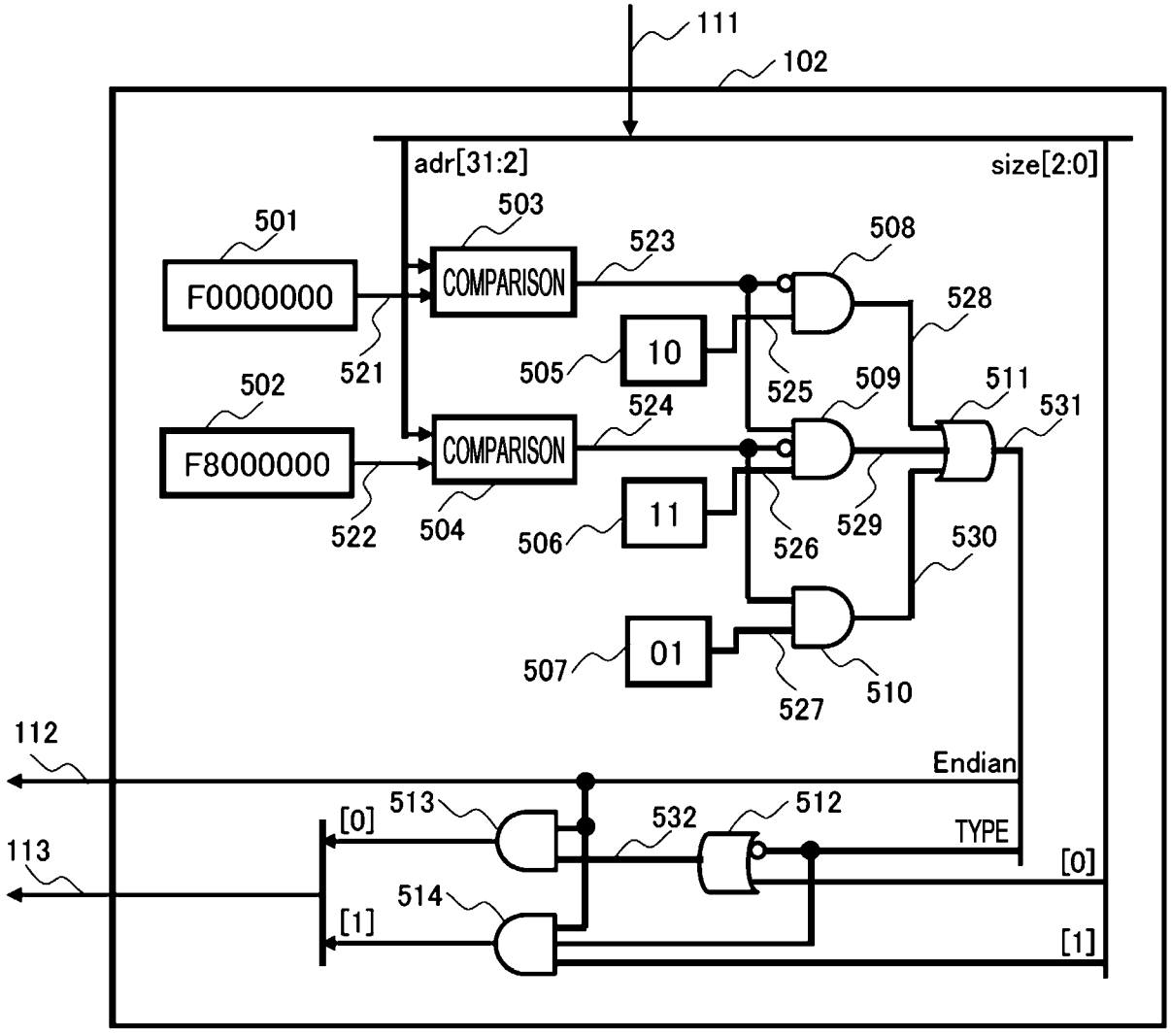

111
102
adr[31:2]
size[2:0]
501
503
523
508
F0000000
COMPARISON
528
10
505
521
525
502
509
511
531
524
F8000000
COMPARISON
11
522
504
506
526
529
530
01
507
527
510
112
Endian
513
[0]
532
512
TYPE
113
[0]
514
[1]
[1]

FIG. 6

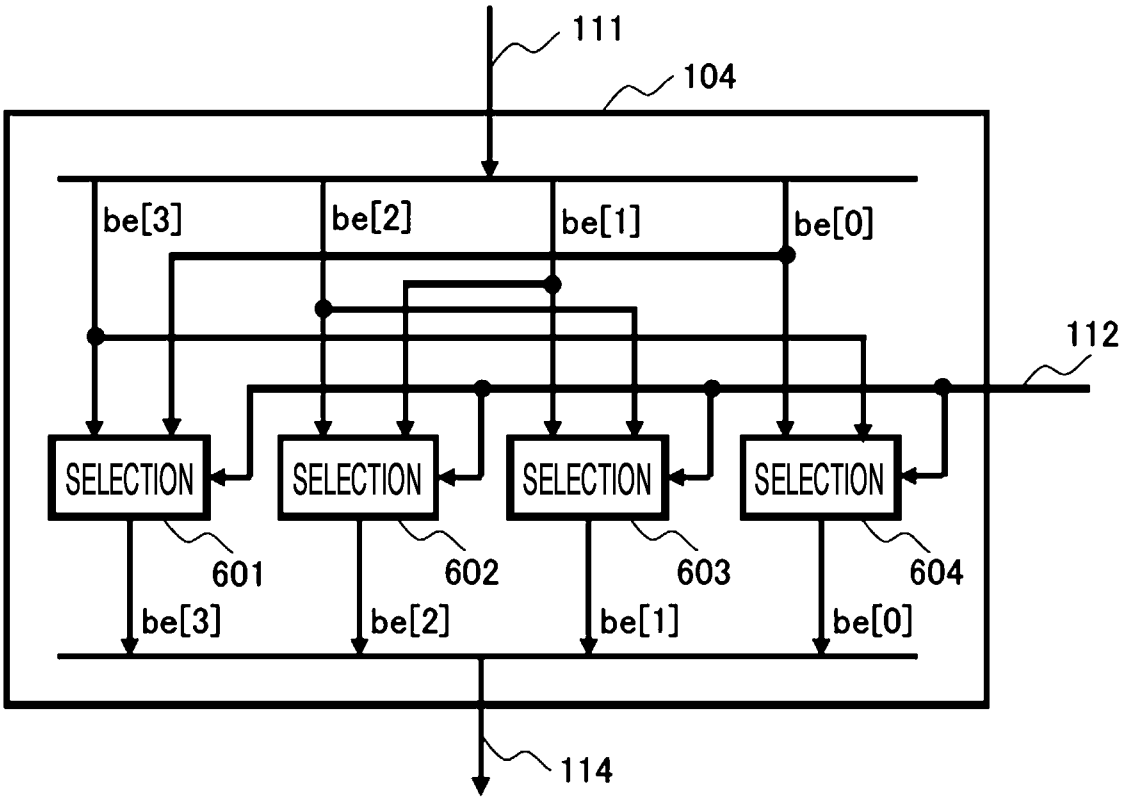


FIG. 7

SIGNAL 112	OUTPUT be[3]	OUTPUT be[2]	OUTPUT be[1]	OUTPUT be[0]
0	INPUT be[3]	INPUT be[2]	INPUT be[1]	INPUT be[0]
1	INPUT be[0]	INPUT be[1]	INPUT be[2]	INPUT be[3]

FIG. 8

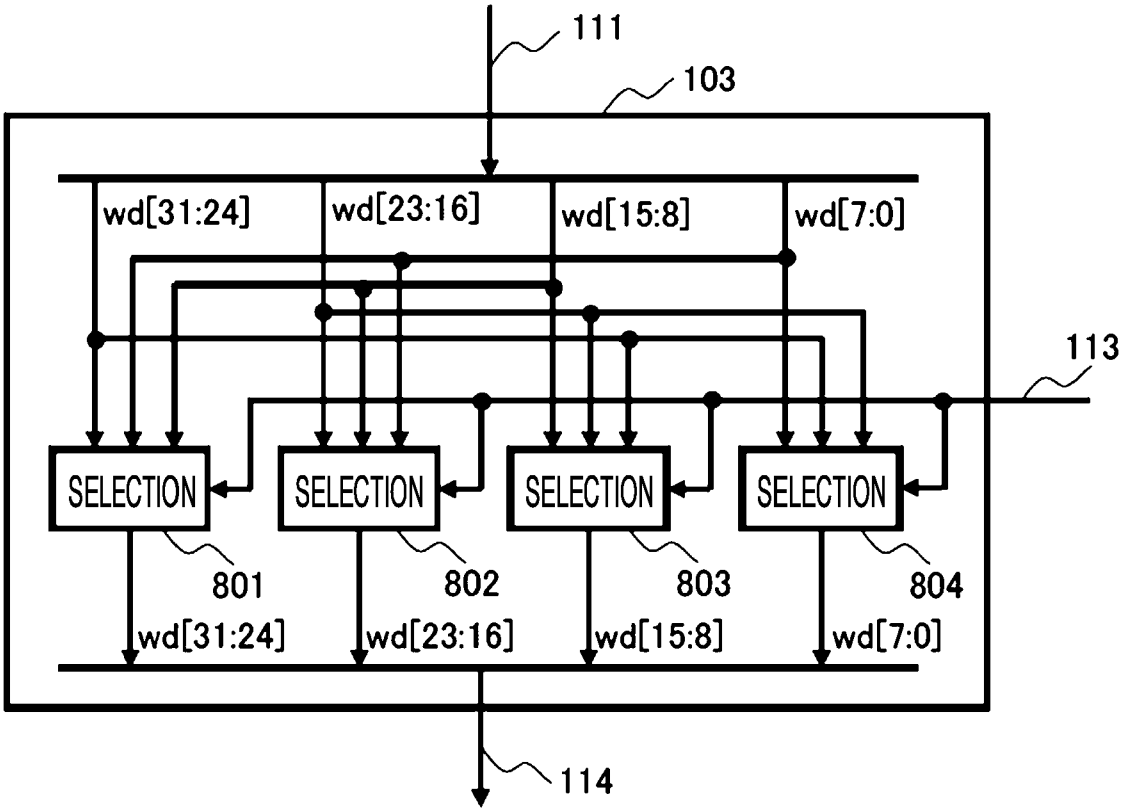


FIG. 9

SIGNAL 113	OUTPUT wd[31:24]	OUTPUT wd[23:16]	OUTPUT wd[15:8]	OUTPUT wd[7:0]
00	INPUT wd[31:24]	INPUT wd[23:16]	INPUT wd[15:8]	INPUT wd[7:0]
01	INPUT wd[7:0]	INPUT wd[15:8]	INPUT wd[23:16]	INPUT wd[31:24]
10	INPUT wd[15:8]	INPUT wd[7:0]	INPUT wd[31:24]	INPUT wd[23:16]

FIG. 10

ACCESS TYPE		BUS111		BUS114					
SIZE	ADDRESS			ROM, RAM		IO1		IO2	
		be	wd	be	wd	be	wd	be	wd
1	4n	0001	---s	1000	s---	1000	s---	0001	---s
1	4n+1	0010	--r-	0100	-r--	0100	-r--	0010	--r-
1	4n+2	0100	-q--	0010	--q-	0010	--q-	0100	-q--
1	4n+3	1000	p---	0001	---p	0001	---p	1000	p---
2	4n	0011	--rs	1100	sr--	1100	rs--	0011	--rs
2	4n+2	1100	pq--	0011	--qp	0011	--pq	1100	pq--
4	4n	1111	pqrs	1111	srqp	1111	pqrs	1111	pqrs

FIG. 11A

AIT (ATTRIBUTE INFORMATION TABLE)

BUS111(adr)	Endian	TYPE	REGISTER SIZE	TARGET
00000000-EFFFFFFF	1	0	-	ROM, RAM
F0000000-F1FFFFFF	1	1	1	IO1
F2000000-F3FFFFFF	1	1	2	IO1
F4000000-F7FFFFFF	1	1	4	IO1
F8000000-FFFFFFFF	0	1	-	IO2

FIG. 11B

CSGR (CONTROL SIGNAL GENERATION RULE)

Endian	TYPE	REGISTER SIZE	BUS 111 (size)	SIGNAL LINE 112	SIGNAL LINE 113	TARGET
0	–	–	–	0	00	IO2
1	0	–	–	1	01	ROM, RAM
1	1	1	–	1	01	IO1
1	1	2	001	1	01	IO1
1	1	2	010	1	10	IO1
1	1	2	100	1	10	IO1
1	1	4	001	1	01	IO1
1	1	4	010	1	10	IO1
1	1	4	100	1	00	IO1

FIG. 12

ACCESS TYPE		BUS 111		BUS 114					
SIZE	ADDRESS			SIZE =1		SIZE =2		SIZE =4	
		be	wd	be	wd	be	wd	be	wd
1	4n	0001	---s	1000	s---	1000	s---	1000	s---
1	4n+1	0010	--r-	0100	-r--	0100	-r—	0100	-r—
1	4n+2	0100	-q--	0010	--q-	0010	--q-	0010	--q-
1	4n+3	1000	p---	0001	---p	0001	---p	0001	---p
2	4n	0011	--rs	1100	sr--	1100	rs—	1100	rs—
2	4n+2	1100	pq--	0011	--qp	0011	--pq	0011	--pq
4	4n	1111	pqrs	1111	srqp	1111	rspq	1111	pqrs

FIG. 13

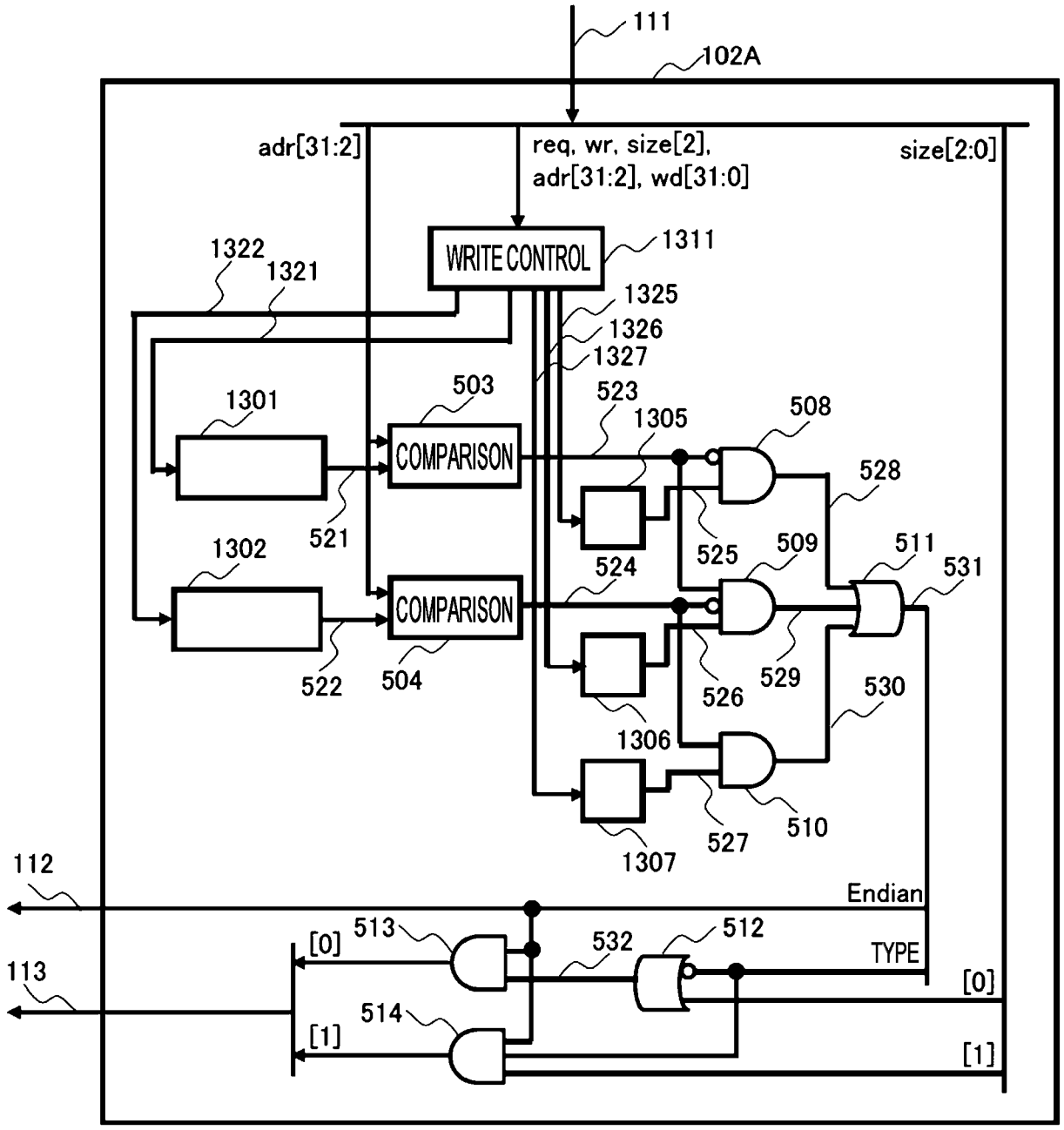

111
102A
adr[31:2]
req, wr, size[2],
adr[31:2], wd[31:0]
size[2:0]
WRITE CONTROL
1311
1322
1321
1325
1326
1327
1301
503
523
1305
508
COMPARISON
528
521
525
509
511
531
1302
524
COMPARISON
529
530
522
504
526
1306
510
527
1307
112
Endian
513
[0]
532
512
TYPE
113
[0]
514
[1]
[1]

FIG. 14

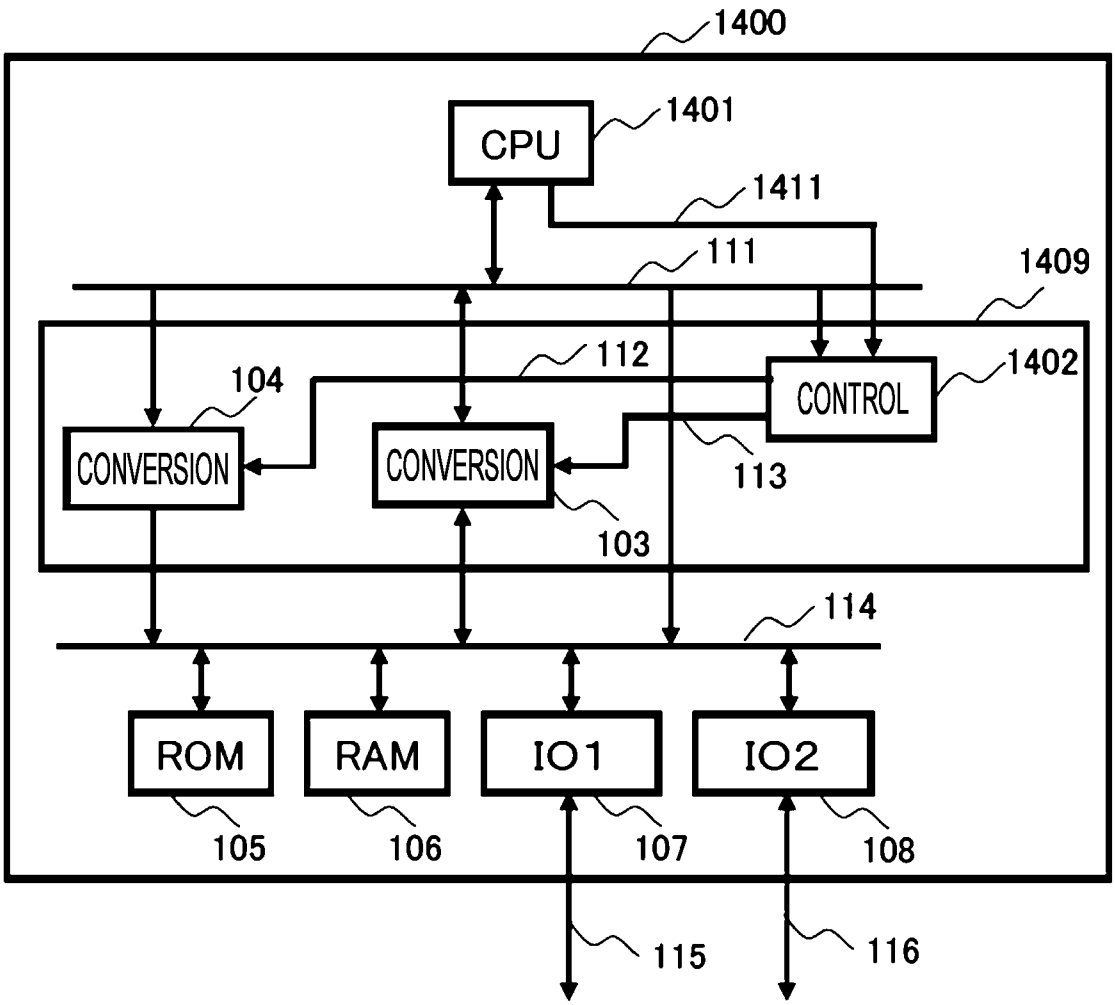


FIG. 15A

AIT (ATTRIBUTE INFORMATION TABLE)

BUS 111(adr)	Endian	TYPE	TARGET
00000000-EFFFFFFF	1	0	ROM, RAM
F0000000-F7FFFFFF	1	1	IO1
F8000000-FFFFFFFF	0	1	IO2

FIG. 15B

CSGR (CONTROL SIGNAL GENERATION RULE)

SIGNAL LINE 1411	Endian	TYPE	BUS 111 (size)	SIGNAL LINE 112	SIGNAL LINE 113	TARGET
0	0	–	–	0	00	IO2
0	1	0	–	1	01	ROM, RAM
0	1	1	001	1	01	IO1
0	1	1	010	1	10	IO1
0	1	1	100	1	00	IO1
1	0	1	001	1	01	IO2
1	0	1	010	1	10	IO2
1	0	1	100	1	00	IO2
1	1	0	–	0	00	ROM, RAM
1	1	1	–	0	00	IO1

FIG. 16

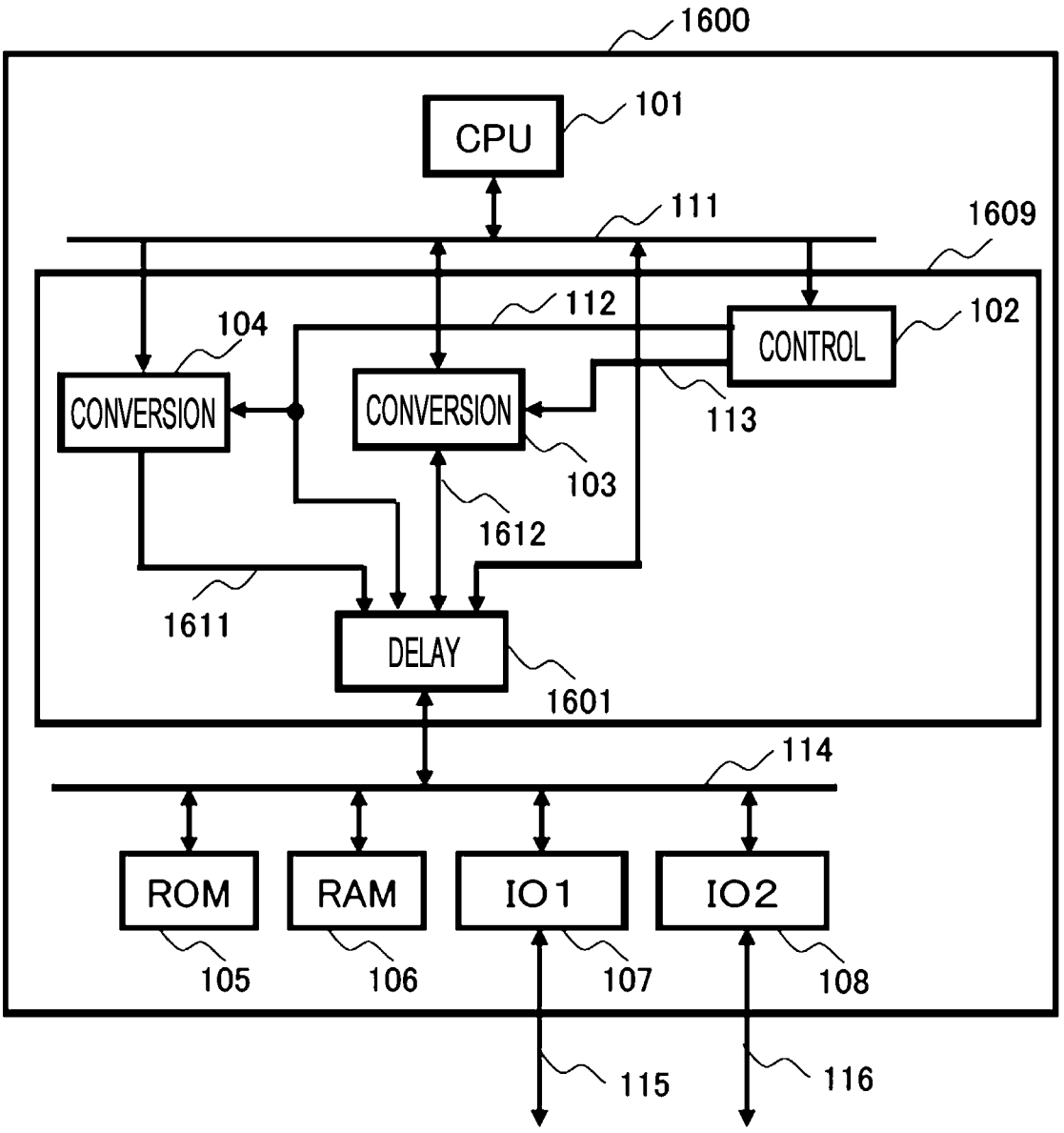

1600
CPU
101
111
1609
112
104
CONTROL
102
CONVERSION
CONVERSION
113
103
1612
1611
DELAY
1601
114
ROM
RAM
IO1
IO2
105
106
107
108
115
116

FIG. 17

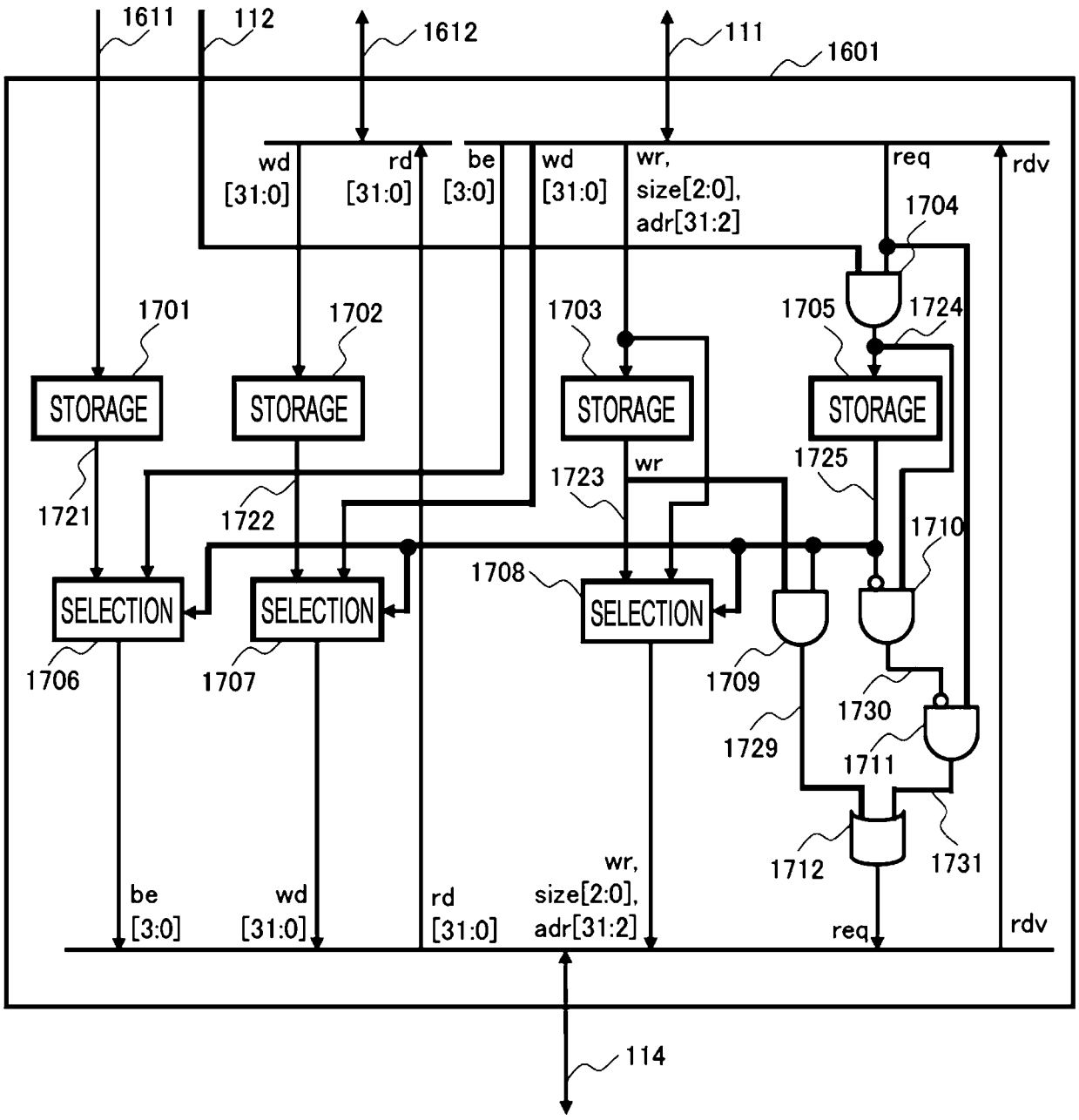
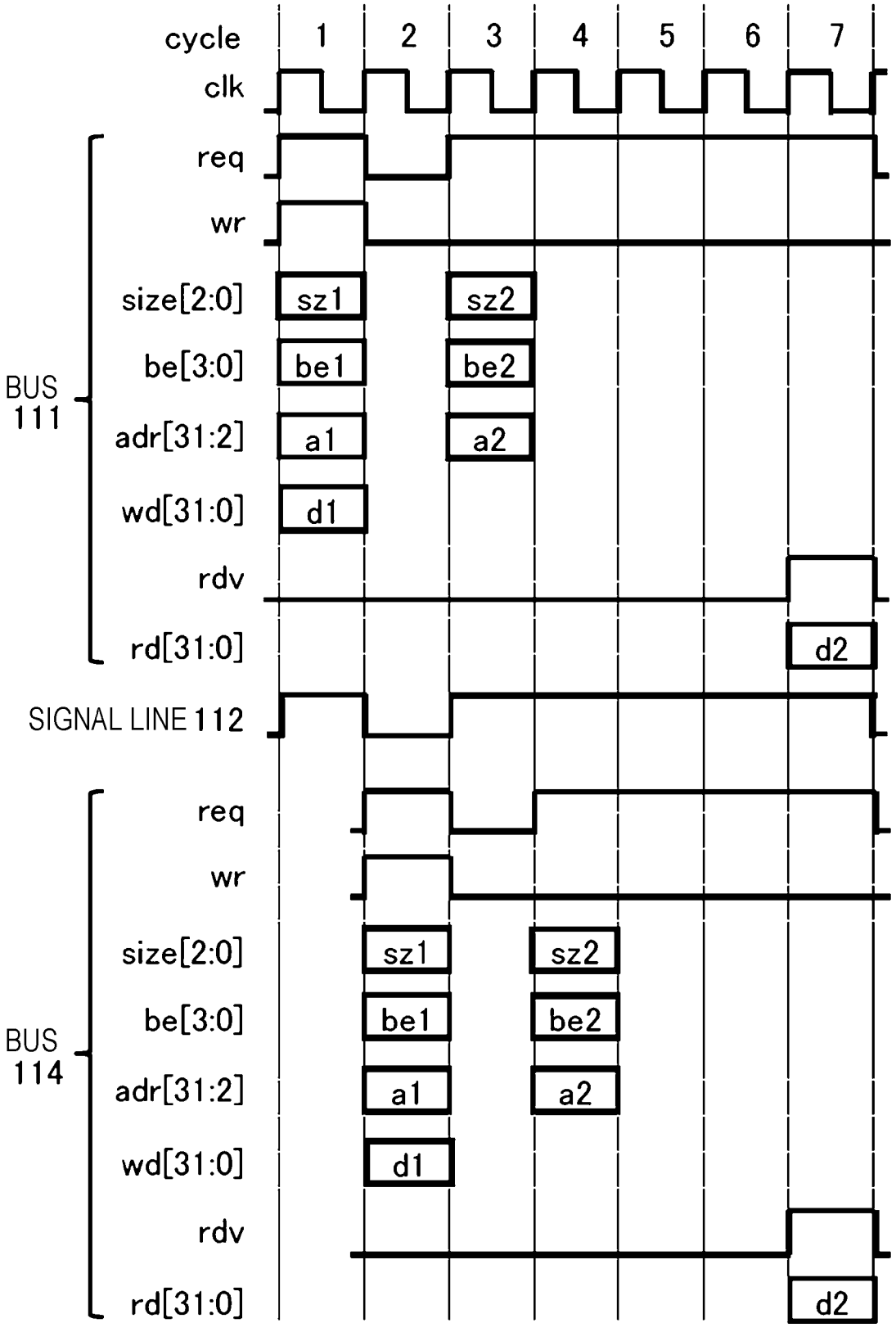

1611
112
1612
111
1601
wd [31:0]
rd [31:0]
be [3:0]
wd [31:0]
wr, size[2:0], adr[31:2]
req
rdv
1704
1701
1702
1703
1705
1724
STORAGE
STORAGE
STORAGE
STORAGE
1725
wr
1723
1721
1722
1710
1708
SELECTION
SELECTION
SELECTION
1706
1707
1709
1730
1729
1711
1712
1731
wr, size[2:0], adr[31:2]
be [3:0]
wd [31:0]
rd [31:0]
req
rdv
114

FIG. 18


cycle
1
2
3
4
5
6
7
clk
req
wr
size[2:0]
sz1
sz2
be[3:0]
be1
be2
adr[31:2]
a1
a2
wd[31:0]
d1
rdv
rd[31:0]
d2
BUS 111
SIGNAL LINE 112
req
wr
size[2:0]
sz1
sz2
be[3:0]
be1
be2
adr[31:2]
a1
a2
wd[31:0]
d1
rdv
rd[31:0]
d2
BUS 114

DESCRIPTION

Title of Invention: PROCESSOR AND ENDIAN CONVERSION METHOD

Technical Field

[0001]

The present disclosure relates to a processor including an endian conversion circuit provided between a CPU and a peripheral circuit, and an endian conversion method.

Background Art

[0002]

JP H08-202646 A (PTL 1) is a background art of the present technical field. This publication describes an endian conversion circuit (converter) for facilitating connection between a CPU (main processor) and a peripheral circuit (I/O device) in a case where endians are different between the CPU and the peripheral circuit.

Citation List

Patent Literature

[0003]

PTL 1: JP H08-202646 A

Summary of Invention

Technical Problem

[0004]

The endian conversion circuit of PTL 1 rearranges pieces of data in a case where the endians are different between the CPU and the peripheral circuit. For example, in a case where a hexadecimal numerical value 00000001 of 4 bytes is written from the CPU to a peripheral function, this value is converted into 01000000. This corresponds to the fact that a method for allocating an address to a numerical value of 4 bytes is different between little endian and big endian. Specifically, in the little endian, addresses are allocated in the order from a low-order byte to a high-order byte, whereas in the big endian, addresses are allocated in the order from the high-order byte to the low-order byte. Accordingly, in order to store the same data at the same address when the data is interpreted in units of 1 byte, it is necessary to switch an arrangement order of the pieces of data. However, in a case where the data is interpreted as a numerical value of 4 bytes, different values are written. When the peripheral circuit interprets a write value as a value of 4 bytes, the write value is interpreted as a value different from a value intended by a creator of software. This is not limited to the case of 4 bytes, and in general, in a case where the write value is interpreted in units larger than 1 byte, the write value is interpreted as the value different from the value intended by the creator of the software.

[0005]

An object of the present disclosure is to provide means for preventing a write value from being interpreted as a value different from a value intended by a creator of software and facilitating creation of software in a processor including a CPU, a plurality of peripheral circuits, and an endian conversion circuit provided between the CPU and the plurality of peripheral circuits in a case where the peripheral circuit interprets the write value as a value of a plurality of bytes.

[0006]

Other objects, and novel features will be apparent from the description of the present specification and the accompanying drawings.

Solution to Problem

[0007]

A brief description of an outline of a representative embodiment of the present disclosure is as follows.

[0008]

In order to solve the above problem, in a processor according to an embodiment, an endian conversion circuit includes a table indicating a relationship between an address and a type of a connection destination, and a data conversion circuit and a byte enable conversion circuit of the endian

conversion circuit are separately controlled by using type information extracted from the table by using an address of an access destination.

Advantageous Effects of Invention

[0009]

According to the processor according to the above embodiment, in a case where the endians of the CPU and the peripheral circuit are different, the byte enable conversion circuit converts byte enable in accordance with a difference in method of allocating an address to data of a plurality of bytes, and the data conversion circuit converts data in consideration of how many bytes the peripheral circuit interprets a numerical value. As a result, the peripheral circuit can correctly interpret the value intended by the creator of the software.

[0010]

Other objects, configurations, and effects will be made apparent in the following descriptions.

Brief Description of Drawings

[0011]

[FIG. 1] FIG. 1 is a first example of a configuration diagram of a processor to which the present invention is applied.

[FIG. 2] FIG. 2 is an example of a timing chart of a bus 111 and a bus 114 of FIG. 1.

[FIG. 3A] FIG. 3A is an example of an operation of byte enable of little endian of the bus 111 and the bus 114 of FIG. 1.

[FIG. 3B] FIG. 3B is an example of an operation of byte enable of big endian of the bus 111 and the bus 114 of FIG. 1.

[FIG. 4A] FIG. 4A is a diagram illustrating a first example of an attribute information table of a control circuit 102 of FIG. 1.

[FIG. 4B] FIG. 4B is a diagram illustrating a first example of a control signal generation rule of the control circuit 102 of FIG. 1.

[FIG. 5] FIG. 5 is a first example of the control circuit 102 of FIG. 1.

[FIG. 6] FIG. 6 is an example of a byte enable conversion circuit 104 of FIG. 1.

[FIG. 7] FIG. 7 is an example of an operation of the byte enable conversion circuit 104 of FIG. 1.

[FIG. 8] FIG. 8 is an example of a data conversion circuit 103 of FIG. 1.

[FIG. 9] FIG. 9 is an example of an operation of the data conversion circuit 103 of FIG. 1.

[FIG. 10] FIG. 10 is a first example of operations of

the bus 111 and the bus 114 of FIG. 1.

[FIG. 11A] FIG. 11A is a diagram illustrating a second example of the attribute information table of the control circuit 102 of FIG. 1.

[FIG. 11B] FIG. 11B is a diagram illustrating a second example of the control signal generation rule of the control circuit 102 of FIG. 1.

[FIG. 12] FIG. 12 is a second example of the operations of the bus 111 and the bus 114 of FIG. 1.

[FIG. 13] FIG. 13 is a second example of the control circuit 102 of FIG. 1.

[FIG. 14] FIG. 14 is a second example of the configuration diagram of the processor to which the present invention is applied.

[FIG. 15A] FIG. 15A is a diagram illustrating an attribute information table of a control circuit 1402 of FIG. 14.

[FIG. 15B] FIG. 15B is a diagram illustrating a control signal generation rule of the control circuit 1402 of FIG. 14.

[FIG. 16] FIG. 16 is a third example of the configuration diagram of the processor to which the present invention is applied.

[FIG. 17] FIG. 17 is an example of a delay circuit 1601 of FIG. 16.

[FIG. 18] FIG. 18 is an example of a timing chart of a bus 111 and a bus 114 of FIG. 16.

Description of Embodiments

[0012]

Hereinafter, embodiments and examples will be described with reference to the drawings. However, in the following description, the same components are denoted by the same reference signs, and repeated description may be omitted. Note that, in order to make the description clearer, the drawings may be schematically illustrated as compared with actual aspects, but these drawings are merely examples and do not limit the interpretation of the present invention.

[0013]

A processor according to an embodiment has the following configuration.

[0014]

That is, a processor (100) includes a CPU (101), a plurality of peripheral circuits (105 to 108), and an endian conversion circuit (109) provided between the CPU (101) and the plurality of peripheral circuits (105 to 108). The endian conversion circuit (109) includes a table (AIT) that stores type information (endian attribute or type attribute) indicating a relationship between an address and a type of a connection destination, a byte enable conversion circuit

(104), and a data conversion circuit (103). The endian conversion circuit (109) separately controls the byte enable conversion circuit (104) and the data conversion circuit (103) by using the type information (endian attribute or type attribute) extracted from the table (AIT) by using an output address of an access destination output from the CPU (101).

[0015]

In addition, an endian conversion method according to the embodiment has the following configuration.

[0016]

An endian conversion method of a processor (100) that includes a CPU (101), a plurality of peripheral circuits (105 to 108), and an endian conversion circuit (109) provided between the CPU (101) and the plurality of peripheral circuits (105 to 108). The endian conversion circuit (109) includes a table (AIT) that stores type information (endian attribute or type attribute) indicating a relationship between an address and a type of a connection destination, a byte enable conversion circuit (104), and a data conversion circuit (103). The endian conversion method includes a) a step of extracting the type information (endian attribute or type attribute) from the table (AIT) by using an output address of an access destination output from the CPU (101), and b) a step of separately controlling the byte enable

conversion circuit (104) and the data conversion circuit (103) based on the extracted type information (endian attribute or type attribute).

[0017]

According to the processor and the endian conversion method according to the above embodiment, when endians of the CPU and the peripheral circuit are different, the byte enable conversion circuit converts byte enable in accordance with a difference in method for allocating an address to data of a plurality of bytes, and the data conversion circuit converts data in consideration of how many bytes the peripheral circuit interprets a numerical value. As a result, the peripheral circuit can correctly interpret the value intended by the creator of the software.

[0018]

Hereinafter, examples will be described with reference to the drawings.

Example 1

[0019]

FIG. 1 is a first example of a configuration diagram of a processor to which the present invention is applied.

[0020]

A processor 100 of the present example includes a CPU 101, an endian conversion circuit 109, a first peripheral circuit (ROM) 105, a second peripheral circuit (RAM) 106, a

third peripheral circuit (first IO circuit) 107, a fourth peripheral circuit (second IO circuit) 108, a first bus (hereinafter, referred to as a bus) 111, and a second bus (hereinafter, referred to as a bus) 114. The CPU 101 is connected to the bus 111, and the first peripheral circuit (ROM) 105, the second peripheral circuit (RAM) 106, the third peripheral circuit (first IO circuit) 107, and the fourth peripheral circuit (second IO circuit) 108 are connected to the bus 114. The endian conversion circuit 109 is connected between the bus 111 and the bus 114. The ROM is, for example, a non-volatile memory such as a read only memory, and the RAM is, for example, a volatile memory such as a random access memory.

[0021]

The endian conversion circuit 109 includes a control circuit 102, a data conversion circuit 103, and a byte enable conversion circuit 104.

[0022]

The CPU 101 executes the following processing according to a program stored in the first peripheral circuit (ROM) 105. The CPU 101 first fetches input information from the third peripheral circuit (first IO circuit: IO1) 107 and the fourth peripheral circuit (second IO circuit: IO2) 108 via the buses 114 and 111, performs predetermined processing, and then writes output information to the third peripheral

circuit (first IO circuit) 107 and the fourth peripheral circuit (second IO circuit) 108 via the buses 111 and 114. The second peripheral circuit (RAM) 106 is used for saving the progress of processing.

[0023]

When a fetch request for the input information is received from the CPU 101 via the buses 111 and 114, the third peripheral circuit (first IO circuit) 107 outputs the input information fetched from a signal line 115 to the CPU 101 via the buses 114 and 111. When a write request for the output information is received from the CPU 101 via the buses 111 and 114, the third peripheral circuit (first IO circuit) 107 also retains a write value therein and outputs a write value to the signal line 115.

[0024]

An operation of the fourth peripheral circuit (second IO circuit) 108 is also similar to an operation of the third peripheral circuit (first IO circuit) 107.

[0025]

The control circuit 102 receives information on an access address and an access size from the CPU 101 via the bus 111, generates control signals of the byte enable conversion circuit 104 and the data conversion circuit 103, and outputs the control signals to signal lines 112 and 113.

[0026]

The byte enable conversion circuit 104 performs predetermined conversion on byte enable of the bus 111 according to the control signal (first signal) received from the signal line 112, and outputs the byte enable to the bus 114.

[0027]

The data conversion circuit 103 performs predetermined conversion on write data of the bus 111 according to the control signal (second signal) received from the signal line 113, and outputs the converted data to the bus 114. The data conversion circuit 103 also performs predetermined conversion on read data of the bus 114 according to the control signal received from the signal line 113, and outputs the converted read data to the bus 111.

[0028]

FIG. 2 is an example of a timing chart of the bus 111 and the bus 114 of FIG. 1. Since operations of the bus 111 and the bus 114 are similar to each other, the operations of the buses are described without being distinguished from each other.

[0029]

clk is a clock signal (1 bit), req is a request signal (1 bit), wr is a write signal (1 bit), size[2:0] is a size signal (size information) (3 bits), be[3:0] is a byte enable signal (4 bits), adr[31:2] is an address signal (30 bits),

wd[31:0] is write data (32 bits), rdv is a read data valid signal (1 bit), and rd[31:0] is read data (32 bits). size[2:0] includes 3-bit signals of size[2], size[1], and size[0], size[2] is a most significant bit, and size[0] is a least significant bit. Similarly, be[3:0] includes 4-bit signals of be[3] to be[0], adr[31:2] includes 30-bit signals of adr[31] to adr[2], wd[31:0] includes 32-bit signals of wd[31] to wd[0], rd[31:0] includes 32-bit signals of rd[31] to rd[0], and a larger numerical value in [] is a high-order bit. req, wr, size[2:0], be[3:0], adr[31:2], and wd[31:0] are signals in a direction from the CPU 101 to the peripheral circuits 105 to 108, and rdv and rd[31:0] are signals in a direction from the peripheral circuits 105 to 108 to the CPU 101. An address space is 32 bits, but since the write data and the read data are 4 bytes, low-order 2 bits of an address are not transmitted, and instead, which byte of pieces of data of 4 bytes is valid is indicated by byte enable. An operation of the byte enable will be described later. The size signal is one of three binary values 001, 010, and 100, and indicates transfer (data transfer size) of 1 byte, 2 bytes, and 4 bytes.

[0030]

Each of the bus 111 and the bus 114 can be considered to include, for example, a data bus including 32 data signal lines for transferring pieces of data such as wd[31:0] and

rd[31:0], an address bus including 30 address signal lines for transferring address signals such as adr[31:2], and a control bus including 9 control signal lines for transferring a request signal (req:1 bit), a write signal (wr:1 bit), a size signal (size[2:0]:3 bits), and a byte enable signal (be[3:0]:4 bits).

[0031]

A first cycle (cycle = 1) is data transfer for writing. At the time of writing, req and wr become 1, and valid values (sz1, be1, a1, and d1) are output for size[2:0], be[3:0], adr[31:2], and wd[31:0], respectively.

[0032]

Third to sixth cycles are data transfer for reading. At the time of reading, req becomes 1 and wr becomes 0. In addition, valid values (sz2, be2, and a2) are output for size[2:0], be[3:0], and adr[31:2] in the initial cycle (third cycle). The number of read cycles is not fixed, and the transfer is ended in a cycle in which rdv becomes 1. A valid value (d2) is output for rd[31:0] in a cycle in which rdv becomes 1. Note that, although not illustrated in FIGS. 1 and 2, the peripheral circuits 105 to 108 output values of rdv and rd[31:0]. For rdv, OR of the outputs of the four peripheral circuits is taken on the bus 114. FIG. 2 illustrates a state after OR is taken. In addition, the output value of the peripheral circuit that has output 1 to

rdv on the bus 114 is selected for rd[31:0]. FIG. 2 illustrates a state after the selection. In a case where there is no peripheral circuit that has output 1 to rdv, the value of rd[31:0] is ignored.

[0033]

FIGS. 3A and 3B are examples of operations of byte enable of the bus 111 and the bus 114 of FIG. 1. FIG. 3A is an example of an operation of a byte enable of little endian of the bus 111 and the bus 114 of FIG. 1. FIG. 3B is an example of an operation of a byte enable of big endian of the bus 111 and the bus 114 of FIG. 1. Since the operations are different between little endian little endian and big endian, these operations are separately described. It can be said that the little endian is a first endian, and the big endian is a second endian different from the first endian.

[0034]

As described above, there are three cases where sizes of data to be transferred are 1 byte, 2 bytes, and 4 bytes.

[0035]

In the case of 1-byte (8-bit) transfer (in a case where the size signal: size[2:0] is 001), there are four cases (FIG. 3A: be[3:0]: 0001, 0010, 0100, and 1000, FIG. 3B: be[3:0]: 1000, 0100, 0010, and 0001) depending on which byte of pieces of data of 4 bytes data is output to (that is, in a case where each of the bus 111 and the bus 114 includes 32

data lines DL31 to DL0 as the data bus, there are four cases where data of 1 byte is output to which data line group among data line groups DL31 to DL24, DL23 to DL16, DL15 to DL8, and DL7 to DL0). In the little endian, an address of a low-order byte is small, and an address of a high-order byte is large. On the other hand, in the big endian, an address of a low-order byte is large, and an address of a high-order byte is small.

[0036]

In 2-byte (16-bit) transfer (in a case where the size signal: size[2:0] is 010), only a case where the address is a multiple of 2 is allowed. Accordingly, there are two cases (FIG. 3A:: be[3:0]: 0011, and 1100, FIG. 3B: 1100 and 0011) depending on which byte of high-order 2 bytes or low-order 2 bytes of pieces of data of 4 bytes there is data (that is, in a case where each of the bus 111 and the bus 114 includes 32 data lines DL31 to DL0 as data buses, there are two cases where pieces of data of 2 bytes is output to which data line group among data line groups DL31 to DL16 and DL15 to DL0). 2-byte transfer at an address 4n corresponds to an overlapping of 1-byte transfers at addresses 4n and 4n+1, and 2-byte transfer at an address 4n+2 corresponds to an overlapping of 1-byte transfers at addresses 4n+2 and 4n+3.

[0037]

In 4-byte (32-bit) transfer (in a case where the size

signal: size[2:0] is 100), since all pieces of data of 4 bytes is valid, all bits of byte enable become 1 (be[3:0]: 1111), and values of byte enable are the same between little endian and big endian.

[0038]

FIGS. 4A and 4B are first examples of an operation of the control circuit 102 of FIG. 1. FIG. 4A is a diagram illustrating a first example of an attribute information table AIT of the control circuit 102 of FIG. 1. FIG. 4B is a diagram illustrating a first example of a control signal generation rule CSGR of the control circuit 102 of FIG. 1.

[0039]

The operation of the control circuit 102 is described in the attribute information table AIT and the control signal generation rule CSGR. The attribute information table AIT stores address ranges and pieces of attribute information of the peripheral circuits.

[0040]

The attribute information table AIT receives, as an input, an address of the bus 111 (obtained by concatenating 0 of 2 bits to low-order bits of adr[31:2]), and outputs attribute information of the corresponding peripheral circuit. The attribute information can also be referred to as type information of the peripheral circuit. The attribute information includes two kinds of endian (Endian) and type.

As the endian, 0 is the little endian, and 1 is the big endian. When the type is 0 (first type) indicates that data is interpreted as a set of pieces of data of 1 byte regardless of the value of the size signal (size[2:0]) of the bus 111. When the type is 1 (second type) indicates that pieces of data of 2 bytes or more are interpreted as data of a size indicated by the value of the size signal (size[2:0]) of the bus 111.

[0041]

The control signal generation rule CSGR indicates a rule for generating values to be output to the signal lines 112 and 113 by using, as inputs, the endian, the type, and the size signal (size[2:0]) of the bus 111 output from the attribute information table AIT. Although not specified in this table, it is assumed that the endian of the CPU is the little endian.

[0042]

(The value of the control signal of) the signal line 112 becomes 0 (first value) when an endian attribute is 0, and becomes 1 (second value) when an endian attribute is 1. This indicates that the conversion of the byte enable is not performed when the endians of the CPU and the peripheral circuit are the same, and the conversion of the byte enable is performed when the endians are different.

[0043]

(The value of the control signal of) the signal line 113 takes three values, 00 (first value) indicates that data is not converted, 01 (second value) indicates that data is converted in units of 1 byte, and 10 (third value) indicates that data is converted in units of 2 bytes. Details of the conversion contents will be described later. The signal line 113 becomes 00 when an endian attribute is 0. This indicates that the conversion of the data is not performed when the endians of the CPU and the peripheral circuit are the same. The signal line 113 also constantly becomes 01 (conversion in units of 1 byte) when an endian attribute is 1 and a type attribute is 0. On the other hand, when an endian attribute is 1 and a type attribute is 1, the output values are different depending on the value of the size signal (size[2:0]) of the bus 111. When size[2:0] is 001 (1 byte), the output value becomes 01 (conversion in units of 1 byte), when size[2:0] is 010 (2 bytes), the output value becomes 10 (conversion in units of 2 bytes), and when size[2:0] is 100 (4 bytes), the output value becomes 00 (no conversion).

[0044]

FIG. 5 is a first example of the control circuit 102 of FIG. 1.

[0045]

The control circuit 102 of the present example includes

fixed value output circuits 501, 502, 505, 506, and 507, comparison circuits 503 and 504, AND circuits 508, 509, 510, 513, and 514, and OR circuits 511 and 512.

[0046]

The fixed value output circuit 501 outputs a hexadecimal fixed value F0000000 to a signal line 521.

[0047]

The fixed value output circuit 502 outputs a hexadecimal fixed value F8000000 to a signal line 522.

[0048]

The comparison circuit 503 compares the address of the bus 111 (obtained by concatenating 0 of 2 bits to low-order bits of adr[31:2]) with a value of the signal line 521, and outputs the result to a signal line 523. The output value is 0 in a case where the address is smaller than the hexadecimal fixed value F0000000, and is 1 when the address is greater than or equal to the fixed value F0000000.

[0049]

The comparison circuit 504 compares the address of the bus 111 (obtained by concatenating 0 of 2 bits to low-order bits of adr[31:2]) with a value of the signal line 522, and outputs the result to a signal line 524. The output value is 0 in a case where the address is smaller than the hexadecimal fixed value F8000000, and is 1 when the address is greater than or equal to the fixed value F8000000.

[0050]

The fixed value output circuit 505 outputs a binary fixed value 10 to a signal line 525.

The fixed value output circuit 506 outputs a binary fixed value 11 to a signal line 526.

The fixed value output circuit 507 outputs a binary fixed value 01 to a signal line 527.

The AND circuit 508 outputs a value of the signal line 525 to a signal line 528 when a value of the signal line 523 is 0. This corresponds to a first row of the attribute information table AIT of FIG. 4A. The AND circuit 508 also outputs 00 to the signal line 528 when the value of the signal line 523 is 1. In this case, a value of the signal line 528 does not influence the output of the OR circuit 511.

[0051]

The AND circuit 509 outputs a value of the signal line 526 to a signal line 529 when the value of the signal line 523 is 1 and a value of the signal line 524 is 0. This corresponds to a second row of the attribute information table AIT of FIG. 4A. The AND circuit 509 also outputs 00 to the signal line 529 when the value of the signal line 523 is 0 or the value of the signal line 524 is 1. In this case, a value of the signal line 529 does not influence the output of the OR circuit 511.

[0052]

The AND circuit 510 outputs a value of the signal line 527 to a signal line 530 when the value of the signal line 524 is 1. This corresponds to a third row of the attribute information table AIT of FIG. 4A. The AND circuit 510 also outputs 00 to the signal line 530 when the value of the signal line 524 is 0. In this case, the value of the signal line 530 does not influence the output of the OR circuit 511.

[0053]

The OR circuit 511 takes OR of three values of the signal lines 528, 529, and 530, and outputs the result to a signal line 531. Since two of the three values of the signal lines 528, 529, 530 are 00, an operation of the OR circuit 511 substantially selects one of the three values of the signal lines 528, 529, and 530. The value output to the signal line 531 corresponds to the output of the attribute information table AIT of FIG. 4A.

[0054]

A value of an endian attribute of the signal line 531 is output to the signal line 112 as it is.

[0055]

The OR circuit 512 outputs 1 to a signal line 531 when a type attribute of the signal line 532 is 0 or bit 0 of the size signal (size[2:0]) of the bus 111 is 1 (corresponding to 1 byte), and outputs 0 otherwise.

[0056]

The AND circuit 513 outputs 1 to bit 0 of the signal line 113 when both the endian attribute of the signal line 531 and a value of the signal line 532 are 1. This operation corresponds to a case where 01 is output to the signal line 113 in second and third rows of the control signal generation rule CSGR of FIG. 4B. The AND circuit 513 also outputs 0 when the endian attribute of the signal line 531 or the value of the signal line 532 is 0.

[0057]

The AND circuit 514 outputs 1 to bit 1 of the signal line 113 when both the endian attribute and the type attribute of the signal line 531 are 1 and bit 1 of the size signal (size[2:0]) of the bus 111 is 1 (corresponding to 2 bytes). This operation corresponds to a case where 10 is output to the signal line 113 in a fourth row of the control signal generation rule CSGR of FIG. 4B. The AND circuit 514 also outputs 0 to bit 1 of the signal line 113 when bit 1 of the endian attribute and the type attribute of the signal line 531, or the size signal (size[2:0]) of the bus 111 is 0.

[0058]

FIG. 6 is an example of the byte enable conversion circuit 104 of FIG. 1.

[0059]

The byte enable conversion circuit 104 includes

selection circuits 601, 602, 603, and 604.

[0060]

When a value of the signal line 112 is 0, the selection circuit 601 selects bit 3 (be[3]) of the byte enable signal of the bus 111, and outputs the selected bit 3 (be[3]) to bit 3 (be[3]) of the byte enable signal of the bus 114. The selection circuit 601 also selects bit 0 (be[0]) of the byte enable signal of the bus 111 when the value of the signal line 112 is 1, and outputs, as an endian-converted byte enable signal, the selected bit 0 to bit 3 (be[3]) of the byte enable signal of the bus 114.

[0061]

The same applies to operations of the selection circuits 602, 603, and 604. A left input is selected when the value of the signal line 112 is 0 and a right input is selected when the value of the signal line 112 is 1, and is output as the endian-converted byte enable signal.

[0062]

FIG. 7 is an example of an operation of the byte enable conversion circuit 104 of FIG. 1.

[0063]

When the signal line 112 is 0, the byte enable conversion circuit 104 outputs the same value as the input.

[0064]

When the signal line 112 is 1, the byte enable

conversion circuit 104 switches high-order bites and low-order bits of the input in a reverse order, and outputs the input.

[0065]

FIG. 8 is an example of the data conversion circuit 103 of FIG. 1. The data conversion is performed on the write data (wd[31:0]) and the read data (rd[31:0]), but only the conversion circuit of the write data is illustrated in FIG. 8. The conversion circuit of the read data has the same configuration.

[0066]

The data conversion circuit 103 includes selection circuits 801, 802, 803, and 804.

[0067]

When a value of the signal line 113 is 00, the selection circuit 801 selects bits 31 to 24 (wd[31:24]) of the write data of the bus 111, and outputs the bits 31 to 24 (wd[31:24]) of the write data of the bus 114 without endian conversion. When the value of the signal line 113 is 01, the selection circuit 801 also selects bits 7 to 0 (wd[7:0]) of the write data of the bus 111, and outputs the selected bits as the endian-converted data to bits 31 to 24 (wd[31:24]) of the data of the bus 114. When the value of the signal line 113 is 10, the selection circuit 801 also selects bits 15 to 8 (wd[15:8]) of the write data of the bus 111, and

outputs the selected bits as the endian-converted data to bits 31 to 24 (wd[31:24]) of the write data of the bus 114.

[0068]

The same applies to the operations of the selection circuits 802, 803, and 804. A left input is selected and output when the value of the signal line 113 is 00, a center input is selected and output when the value of the signal line 113 is 01, and a right input is selected and output when the value of the signal line 113 is 10.

[0069]

FIG. 9 is an example of an operation of the data conversion circuit 103 of FIG. 1.

[0070]

When the signal line 113 is 00, the data conversion circuit 103 outputs the same value as the switched input.

[0071]

When the signal line 113 is 01, the data conversion circuit 103 switches high-order bits and low-order bits of the input in units of bytes (8 bits) in a reverse order, and outputs the switched input.

[0072]

When the signal line 113 is 10, the data conversion circuit 103 switches high-order 16 bits (bits 31 to 16) and low-order 16 bits (bits 15 to 0) and outputs the switched input.

[0073]

FIG. 10 is a first example of operations of the bus 111 and the bus 114 of FIG. 1. An operation of the read data is not described in FIG. 10, but is similar to the write data.

[0074]

The first peripheral circuit (ROM) 105 and the second peripheral circuit (RAM) 106 interpret write data in units of 1 byte regardless of the access size. Thus, the endian conversion is performed such that pieces of data of the same address are the same in byte units. For example, when the access size is 4 bytes, a least significant byte (s) of the bus 111 is output to a most significant byte of the bus 114. This is because the least significant byte corresponds to an address 4n on the bus 111, whereas the most significant byte corresponds to an address 4n on the bus 114.

[0075]

The third peripheral circuit (first IO circuit) 107 interprets data in units of the same size as the access size. Thus, the endian conversion is performed such that an arrangement order of the pieces of data is saved in units of access sizes. For example, in a case where the access size is 2 bytes and the address is 4n, the data is shifted from the low-order bits to the high-order bits while an arrangement order (rs) of the pieces of data is maintained.

This guarantees that the interpretation of the data by the CPU 101 and the interpretation of the data by the third peripheral circuit (first IO circuit) 107 are the same.

[0076]

Since the endian of the fourth peripheral circuit (second IO circuit) 108 is the same as the CPU 101, the endian conversion is not performed.

Example 2

[0077]

FIGS. 11A and 11B illustrate a second example of the operation of the control circuit 102 of FIG. 1. FIG. 11A is a diagram illustrating a second example of the attribute information table of the control circuit 102 of FIG. 1. FIG. 11B is a diagram illustrating a second example of the control signal generation rule of the control circuit 102 of FIG. 1.

[0078]

The operation of the control circuit 102 of FIGS. 4A and 4B is different from the operation of FIGS. 11A and 11B in that an output of a register size is added to the attribute information table AIT of FIG. 11A and an output of the control signal generation rule CSGR of FIG. 11B is different depending on a value of a register size attribute. The register size attribute indicates a size of a unit in which a peripheral circuit of an access destination interprets a write value. As described in the description of FIG. 10, in

the case of the operation of the control circuit 102 of FIGS. 4A and 4B, in a case where the third peripheral circuit (first IO circuit) 107 is accessed, data is interpreted in units of the same size as the access size. In the case of such an operation, for example, when the third peripheral circuit (first IO circuit) 107 accesses an address for which data is interpreted in units of 1 byte by 2 bytes or 4 bytes, there is a problem that a difference in interpretation from the CPU 101 occurs. In order to cope with such a problem, in the operation of the control circuit 102 of FIGS. 11A and 11B, the output of the signal line 113 is determined in accordance with the register size attribute (how many bytes the data is interpreted) of the third peripheral circuit (first IO circuit) 107.

[0079]

In the attribute information table AIT of FIG. 11A, a region of the third peripheral circuit (first IO circuit) 107 is divided into three, and 1 byte is output to the register size attribute in addresses F0000000 to F1FFFFFF, 2 bytes is output to the register size attribute in addresses F2000000 to F3FFFFFF, and 4 bytes is output to the register size attribute in addresses F4000000 to F7FFFFFF. In the first peripheral circuit (ROM) 105, the second peripheral circuit (RAM) 106, and the fourth peripheral circuit (second IO circuit) 107, since a value of the register size attribute

is ignored in the control signal generation rule of FIG. 11B, the output value of the register size attribute may be any one of 1 byte, 2 bytes, and 4 bytes.

[0080]

In the control signal generation rule CSGR of FIG. 11B, the operation at the time of accessing the third peripheral circuit (first IO circuit) 107 is different from the operation in FIG. 4B. When the register size attribute is 1 byte, the output to the signal line 113 is 01 regardless of the value of the size signal (size[2:0]) of the bus 111. This is the same as FIG. 4B when size[2:0] is 001 (1 byte), but the operation when size[2:0] is 010 (2 bytes) and 100 (4 bytes) is different from the operation in FIG. 4B. When the register size attribute is 2 bytes, the operation when size[2:0] is 001 (1 byte) and 010 (2 bytes) is the same as the operation in FIG. 4B, but the operation when size[2:0] is 100 (4 bytes) is different from the operation in FIG. 4B, and the output to the signal line 113 is 10. The operation when the register size attribute is 4 bytes is the same as the operation in FIG. 4B.

[0081]

FIG. 12 is a second example of the operations of the bus 111 and the bus 114 of FIG. 1. FIG. 12 corresponds to the second example of the operation of the control circuit 102 of FIGS. 11A and 11B. Since the operation at the time

of accessing the first peripheral circuit (ROM) 105, the second peripheral circuit (RAM) 106, and the fourth peripheral circuit (second IO circuit) 108 is the same as the operation in FIG. 10, the description is omitted, and only the operation at the time of accessing the third peripheral circuit (first IO circuit) 107 is described. In FIG. 12, size = 1 indicates the operation when the register size attribute is 1 byte, size = 2 indicates the operation when the register size attribute is 2 bytes, and size = 4 indicates the operation when the register size attribute is 4 bytes.

[0082]

When the register size attribute is 1 byte, the operation is different from the operation in FIG. 10 when the access size is 2 bytes and 4 bytes. In both cases, the same data is output to bytes having the same address in byte units. For example, when the access size is 4 bytes, the least significant byte (s) of the bus 111 is output to the most significant byte of the bus 114. This is because the least significant byte of the bus 111 corresponds to an address 4n, whereas the most significant byte of the bus 114 corresponds to the address 4n. When the register size attribute is 1 byte, since the third peripheral circuit (first IO circuit) 107 interprets data in units of 1 byte, this operation guarantees that the CPU 101 and the third

peripheral circuit (first IO circuit) 107 interpret data identically.

[0083]

When the register size attribute is 2 bytes, the operation is different from the operation in FIG. 10 when the access size is 4 bytes. Low-order 2 bytes (rs) of the bus 111 are output to high-order 2 bytes of the bus 114 while the arrangement order of pieces of data is saved. Since the low-order 2 bytes of the bus 111 correspond to addresses 4n and 4n+1, data is shifted to a region of the same address of the bus 114, but since the arrangement order of the pieces of data is saved, this operation guarantees that the CPU 101 and the third peripheral circuit (first IO circuit) 107 interpret data identically. Similarly, high-order 2 bytes (pq) of the bus 111 are also output to low-order 2 bytes of the bus 114 while the arrangement order of the pieces of data is saved.

Example 3

[0084]

FIG. 13 is a second example of the control circuit 102 of FIG. 1.

[0085]

A difference between the control circuit 102 of FIG. 5 and a control circuit 102A of FIG. 13 is that the fixed value output circuits 501, 502, 505, 506, and 507 of FIG. 5

are replaced with storage circuits 1301, 1302, 1305, 1306, and 1307, and values can be rewritten from the bus 111.

[0086]

When the request signal (req) of the bus 111 is 1, the write signal (wr) is 1, the size signal (size[2:0]) is 100 (4 bytes), and the address signal (adr[31:2]) is a predetermined value, a write control circuit 1311 outputs write control signals of the storage circuits 1301, 1302, 1305, 1306, and 1307 to one of signal lines 1321, 1322, 1325, 1326, and 1327. The write control signal includes write enable and write data, and a part of the write data (wd[31:0]) of the bus 111 is selected and output as the write data.

[0087]

With the configuration of the control circuit 102A of FIG. 13, contents (stored values) of the attribute information table AIT of FIG. 4A can be set from the CPU 101. That is, the control circuit has a function of rewriting the stored values of the table AIT according to a request from the CPU 101 or a step of rewriting the stored values of the table AIT according to a request from the CPU 101. As a result, even in a case where the configuration of the peripheral circuit is changed, the same control circuit 102A can be used.

Example 4

[0088]

FIG. 14 is a second example of a configuration diagram of a processor to which the present invention is applied.

[0089]

A difference between the processor 100 of FIG. 1 and a processor 1400 of FIG. 14 is that a CPU 1401 supports both little endian and big endian operations and outputs information indicating which endian the CPU is currently operating to a control circuit 1402 via a signal line 1411. That is, the CPU 1401 has a function of changing the endian to the little endian or the big endian. An endian conversion circuit 1409 changes a rule of the endian conversion in accordance with information regarding the endian of the CPU 1401 received from the signal line 1411. The processor 1400 includes a step of changing the endian of the CPU 1401, a step of transmitting the information regarding the endian of the CPU 1401 to the signal line 1411 between the CPU 1401 and the endian conversion circuit 1409, and a step of changing the rule of the endian conversion of the endian conversion circuit 1409 in accordance with the information regarding the endian of the CPU 1401 received from the signal line 1411.

[0090]

The control circuit 1402 generates signals to be output to the signal lines 112 and 113 in consideration of the information regarding the endian of the CPU 1401 fetched

from the signal line 1411 in addition to a table AIT built therein. Details will be described later.

[0091]

FIGS. 15A and 15B are examples of an operation of the control circuit 1402 of FIG. 14. FIG. 15A is a diagram illustrating an attribute information table AIT of the control circuit 1402 of FIG. 14. FIG. 15B is a diagram illustrating a control signal generation rule CSGR of the control circuit 1402 of FIG. 14.

[0092]

The attribute information table AIT of FIG. 15A is the same as the attribute information table AIT of FIG. 4A.

[0093]

On the other hand, the control signal generation rule CSRG of FIG. 15B is different from the control signal generation rule CSRG of FIG. 4B in that a value of the signal line 1411 is taken into consideration. The operation when the value of the signal line 1411 is 0 is the same as the operation in FIG. 4B. In a case where the value of the signal line 1411 is 1 and the endian attribute is 0, since the endians of the CPU 1401 and the peripheral circuit are different, it is necessary to perform the endian conversion of the byte enable, and the signal line 112 becomes 1. On the other hand, the signal line 113 is 01 when the type attribute is 0, and is 01, 10, 00, or 00 in accordance with

the value of the size signal (size[2:0]) of the bus 111 when the type attribute is 1. Note that, since there is no case where the endian attribute is 0 and the type attribute is 0 in the attribute information table AIT of FIG. 15A, the case is not described in FIG. 15B. In a case where the value of the signal line 1411 is 1 and the endian attribute is also 1, since the endians of the CPU 1401 and the peripheral circuit are the same, it is not necessary to perform the endian conversion, and 0 is output to the signal lines 112 and 113.

Example 5

[0094]

FIG. 16 is a third example of a configuration diagram of a processor to which the present invention is applied.

[0095]

A difference between the processor 100 of FIG. 1 and a processor 1600 of FIG. 16 is that an endian conversion circuit 1609 includes a delay circuit 1601. In the processor 100 of FIG. 1, since it takes time in the processing of the control circuit 102, the data conversion circuit 103, and the byte enable conversion circuit 104, a transfer time from the CPU 101 to the peripheral circuits (105, 106, 107, and 108) becomes long, and thus, there is a possibility that an operation frequency decreases. In the present example, in order to cope with this problem, the decrease in the

operation frequency is prevented by dividing the transfer from the CPU 101 to the peripheral circuits (105, 106, 107, and 108) into two clock cycles by the delay circuit 1601 and shortening a transfer time per one clock cycle. Note that, since the transfer of the read data from the peripheral circuits (105, 106, 107, and 108) to the CPU 101 is delayed from the start of the transfer by 1 clock cycle or more, the operation of the control circuit 102 is completed at this point in time. Since the influence of the decrease in the operation frequency by the endian conversion circuit is slight, the transfer of the read data is not divided into 2 clock cycles in the present example. However, since the transfer time becomes long due to a processing time of the data conversion circuit 103, the transfer of the read data may also be divided into clock cycles in order to eliminate the influence.

[0096]

When the value fetched from the signal line 112 is 0, the delay circuit 1601 outputs byte enable fetched from a signal line 1611 and write data fetched from a signal line 1612 to the bus 114 as they are. When the value fetched from the signal line 112 is 1, the delay circuit 1601 also outputs the byte enable fetched from the signal line 1611 and the write data fetched from the signal line 1612 to the bus 114 with a delay of 1 clock cycle. The delay circuit

1601 also outputs the read data fetched from the bus 114 to the signal line 1612 as it is. The delay circuit 1601 also outputs the read data valid signal fetched from the bus 114 to the bus 111 as it is. That is, in a case where it is necessary to perform endian conversion, the endian conversion circuit 1609 has a function or a step of delaying the byte enable and data by 1 clock cycle by the delay circuit 1601.

[0097]

FIG. 17 is an example of the delay circuit 1601 of FIG. 16.

[0098]

The delay circuit 1601 includes storage circuits 1701, 1702, 1703, and 1705, AND circuits 1704, 1709, 1710, and 1711, selection circuits 1706, 1707, and 1708, and an OR circuit 1712.

[0099]

The storage circuit 1701 outputs a byte enable signal fetched from the signal line 1611 to a signal line 1721 with a delay of 1 clock cycle.

[0100]

The storage circuit 1702 outputs write data (wd[31:0]) fetched from the signal line 1612 to a signal line 1722 with a delay of 1 clock cycle.

[0101]

The storage circuit 1703 outputs a write signal (wr), a size signal (size[2:0]), and an address signal (adr[31:2]) fetched from the bus 111 to a signal line 1723 with a delay of one clock cycle.

[0102]

The AND circuit 1704 takes AND of a byte enable conversion valid signal fetched from the signal line 112 and a request signal (req) fetched from the bus 111, and outputs the result to a signal line 1724.

[0103]

The storage circuit 1705 outputs a state of the signal line 1724 to a signal line 1725 with a delay of one clock cycle.

[0104]

When a value of the signal line 1725 is 1, the selection circuit 1706 outputs a value of the signal line 1721 as the byte enable signal (be[3:0]) of the bus 114. When the value of the signal line 1725 is 0, the selection circuit 1706 also outputs the byte enable signal (be[3:0]) of the bus 111 as the byte enable signal (be[3:0]) of the bus 114.

[0105]

When the value of the signal line 1725 is 1, the selection circuit 1707 outputs a value of the signal line 1722 as the write data (wd[31:0]) of the bus 114. When the

value of the signal line 1725 is 0, the selection circuit 1707 also outputs the write data (wd[31:0]) of the bus 111 as the write data (wd[31:0]) of the bus 114.

[0106]

When the value of the signal line 1725 is 1, the selection circuit 1708 outputs a value of the signal line 1723 as the write signal (wr), the size signal (size[2:0]), and the address signal (adr[31:2]) of the bus 114. When the value of the signal line 1725 is 0, the selection circuit 1708 also outputs the write signal (wr), the size signal (size[2:0]), and the address signal (adr[31:2]) of the bus 111 as the write signal (wr), the size signal (size[2:0]), and the address signal (adr[31:2]) of the bus 114.

[0107]

The AND circuit 1709 takes AND of the write signal (wr) of the signal line 1723 and the value of the signal line 1725, and outputs the result to a signal line 1729.

[0108]

The AND circuit 1710 takes AND of an inverted value of the signal line 1725 and a value of the signal line 1724, and outputs the result to a signal line 1730. As a result, the signal line 1730 becomes 1 only in one cycle at the rising of the signal line 1724.

[0109]

The AND circuit 1711 takes AND of an inverted value of

the signal line 1730 and the request signal (req) of the bus 111, and outputs the result to a signal line 1731. As a result, the signal line 1731 becomes a signal in which the rising of the request signal (req) of the bus 111 is delayed by one cycle only when the signal line 112 is 1.

[0110]

The OR circuit 1712 takes OR of a value of the signal line 1729 and a value of the signal line 1731, and outputs the result as the request signal (req) of the bus 114.

[0111]

FIG. 18 is an example of a timing chart of the bus 111 and the bus 114 of FIG. 16.

[0112]

In a first cycle (cycle = 1), data transfer for writing is performed on the bus 111. Since the signal line 112 is 1, in cycle 2, the same data transfer of writing is performed on the bus 114.

[0113]

In third to seventh cycles, data transfer for reading is performed on the bus 111. Since the signal line 112 is 1, the start of the same data transfer for reading is delayed to the fourth cycle on the bus 114, but the end of the data transfer is the same seventh cycle as the bus 111. This is because, in the delay circuit 1601 of FIG. 17, there is a circuit that delays the request signal (req), the write

signal (wr), the size signal (size[2:0]), the byte enable signal (be[3:0]), and the address signal (adr[31:2]) by one cycle, but there is no circuit that delays the read data valid signal (rdv) and the read data (rd[31:0]) by one cycle.

[0114]

Although the invention made by the present inventors has been specifically described above based on the examples, the present invention is not limited to the above embodiments and examples, and it goes without saying that various modifications can be made.

Reference Signs List

[0115]

100 processor

101 CPU

103 data conversion circuit

104 byte enable conversion circuit

105 first peripheral circuit (ROM)

106 second peripheral circuit (RAM)

107 third peripheral circuit (first IO circuit: IO1)

108 fourth peripheral circuit (second IO circuit: IO2)

109 endian conversion circuit

501, 502, 505, 506, 507 fixed value output circuit (corresponding to table indicating relationship between address and type of connection destination)

1301, 1302, 1305, 1306, 1307 storage circuit (corresponding

to table indicating relationship between address and type of connection destination)

1311 write control circuit

1400 processor

1401 CPU

1409 endian conversion circuit

1411 signal line for transmitting information on endian of CPU

1600 processor

1601 delay circuit

1609 endian conversion circuit

CLAIMS

[Claim 1]

A processor, comprising:

a CPU;

a plurality of peripheral circuits; and

an endian conversion circuit that is provided between the CPU and the plurality of peripheral circuits,

wherein the endian conversion circuit includes a table that stores type information indicating a relationship between an address and a type of a connection destination, a byte enable conversion circuit, and a data conversion circuit, and

the endian conversion circuit separately controls the byte enable conversion circuit and the data conversion circuit by using the type information extracted from the table by using an output address of an access destination output from the CPU.

[Claim 2]

The processor according to claim 1, wherein

the table further has size information regarding a size for interpreting data by the connection destination, and

the endian conversion circuit separately controls the byte enable conversion circuit and the data conversion circuit by using the type information and the size information extracted from the table.

[Claim 3]

The processor according to claim 1, wherein the endian conversion circuit has a function of rewriting a stored value of the table according to a request from the CPU.

[Claim 4]

The processor according to claim 1, wherein

the CPU has a function of changing an endian,

a signal line for transmitting information regarding an endian of the CPU is provided between the CPU and the endian conversion circuit, and

the endian conversion circuit changes a rule of endian conversion in accordance with the information regarding the endian of the CPU received from the signal line.

[Claim 5]

The processor according to claim 1, wherein the endian conversion circuit has a delay circuit that delays byte enable and data by one clock cycle in a case where it is necessary to perform endian conversion.

[Claim 6]

The processor according to claim 1, further comprising:

a first bus; and

a second bus,

wherein the CPU is connected to the first bus, and outputs, to the first bus, data of 1 byte or a plurality of bytes of a first endian, an address signal of an access

destination, a size signal indicating a size of the data, and a byte enable signal indicating which byte of the data is valid,

the plurality of peripheral circuits are connected to the second bus,

the endian conversion circuit is provided between the first bus and the second bus,

the plurality of peripheral circuits include

a memory that interprets data of a second endian different from the first endian for each byte,

a first IO circuit that interprets the data of the second endian for a plurality of bytes, and

a second IO circuit that interprets the data of the first endian for a plurality of bytes,

the table

includes, for the memory, an address range of the memory, and an endian attribute indicating the second endian and a type attribute indicating a first type for interpreting data for each byte regardless of a value of the size signal, as the type information of the memory,

includes, for the first IO circuit, an address range of the first IO circuit, and an endian attribute indicating the second endian and a type attribute indicating a second type for interpreting data of a size indicated by the value of the size signal for each data, as the type

information of the first IO circuit, and

includes, for the second IO circuit, an address range of the second IO circuit, and an endian attribute indicating the first endian and a type attribute indicating the second type, as the type information of the second IO circuit, and

the endian conversion circuit generates a first signal for controlling the byte enable conversion circuit and a second signal for controlling the data conversion circuit based on the endian attribute and the type attribute output from the table by using an address signal of the access destination output from the CPU and the size signal.

[Claim 7]

The processor according to claim 6, wherein,

in a case where the endian attribute indicates the first endian, the first signal is set to a first value, and the byte enable signal is output to the second bus without performing endian conversion using the byte enable conversion circuit, and

in a case where the endian attribute indicates the second endian, the first signal is set to a second value, and a converted byte enable signal obtained by performing endian conversion on the byte enable signal by using the byte enable conversion circuit is output to the second bus.

[Claim 8]

The processor according to claim 6, wherein

a) in a case where the endian attribute indicates the first endian, the second signal is set to a first value, and the data is output to the second bus without performing endian conversion using the data conversion circuit,

b) in a case where the endian attribute indicates the second endian and the type attribute is the first type, the second signal is set to a second value, and the data is output to the second bus without performing endian conversion using the data conversion circuit,

c) in a case where the endian attribute indicates the second endian and the type attribute is the second type,

c1) when the size signal indicates 1 byte, the second signal is set to the second value, and converted data obtained by performing the endian conversion on the data by using the data conversion circuit is output to the second bus,

c2) when the size signal indicates 2 bytes, the second signal is set to a third value, and converted data obtained by performing the endian conversion on the data by using the data conversion circuit is output to the second bus, and

c3) when the size signal indicates 4 bytes, the second signal is set to the first value, and the data is output to the second bus without performing endian conversion using the data conversion circuit.

[Claim 9]

An endian conversion method of a processor that includes a CPU, a plurality of peripheral circuits, and an endian conversion circuit provided between the CPU and the plurality of peripheral circuits, the endian conversion circuit including a table that stores type information indicating a relationship between an address and a type of a connection destination, a byte enable conversion circuit, and a data conversion circuit, the method comprising:

a) a step of extracting the type information from the table by using an output address of an access destination output from the CPU; and

b) a step of separately controlling the byte enable conversion circuit and the data conversion circuit based on the extracted type information.

[Claim 10]

The endian conversion method according to claim 9, wherein

the table further has size information regarding a size for interpreting data by a connection destination,

the a) step includes a step of extracting the size information from the table, and

in the b) step, the byte enable conversion circuit and the data conversion circuit are separately controlled based on the type information and the size information.

[Claim 11]

The endian conversion method according to claim 9, further comprising a step of rewriting a stored value of the table of the endian conversion circuit according to a request from the CPU.

[Claim 12]

The endian conversion method according to claim 9, further comprising:

a step of changing an endian of the CPU;

a step of transmitting information regarding the endian of the CPU to a signal line between the CPU and the endian conversion circuit; and

a step of changing a rule of endian conversion of the endian conversion circuit in accordance with the information regarding the endian of the CPU received from the signal line.

[Claim 13]

The endian conversion method according to claim 9, further comprising a step of delaying byte enable and data by one clock cycle by a delay circuit in a case where it is necessary for the endian conversion circuit to perform endian conversion.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2022/026785

A. CLASSIFICATION OF SUBJECT MATTER

G06F 13/36(2006.01)i; ***G06F 12/04***(2006.01)i; ***G06F 13/38***(2006.01)i; ***G06F 13/40***(2006.01)i
FI: G06F13/36 320A; G06F12/04 510G; G06F13/38 320Z; G06F13/38 330Z; G06F13/40 310

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F13/36; G06F12/04; G06F13/38; G06F13/40

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2022
Registered utility model specifications of Japan 1996-2022
Published registered utility model applications of Japan 1994-2022

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2000-3304 A (NEC CORP) 07 January 2000 (2000-01-07) paragraphs [0016]-[0022]	1-4, 9-12
A		5-8, 13
Y	JP 8-202646 A (NEC CORP) 09 August 1996 (1996-08-09) paragraphs [0017]-[0033]	1-4, 9-12
A		5-8, 13
Y	WO 2010/029794 A1 (RENESAS TECHNOLOGY CORP) 18 March 2010 (2010-03-18) paragraphs [0036]-[0039]	1-4, 9-12
A		5-8, 13

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 September 2022	**27 September 2022**
Name and mailing address of the ISA/JP **Japan Patent Office (ISA/JP)** **3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915** **Japan**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2022/026785

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
JP 2000-3304 A	07 January 2000	(Family: none)	
JP 8-202646 A	09 August 1996	(Family: none)	
WO 2010/029794 A1	18 March 2010	US 2011/0191569 A1 paragraphs [0080]-[0084] CN 102150139 A TW 201020795 A	