intel®

Intel® IXP28XX Product Line of Network Processors

RDRAM Address Remap Application Note

June 2005

Order Number: 308331-001US

Contents

Figures

Tables

Revision History

Date	Revision	Description
June 2005	001	Initial Release.

1.0 Introduction

Two levels of address mappings are used in RDRAM to avoid resource conflicts and to increase efficiency.

The first level address remapping (Remap 1) is the address interleaving across RDRAM channels to improve concurrency and bandwidth utilization (see Figure 1, “Remap 1 Address Remapping Flowchart”). It is implemented by hardware to direct contiguous blocks of addresses to different channels.

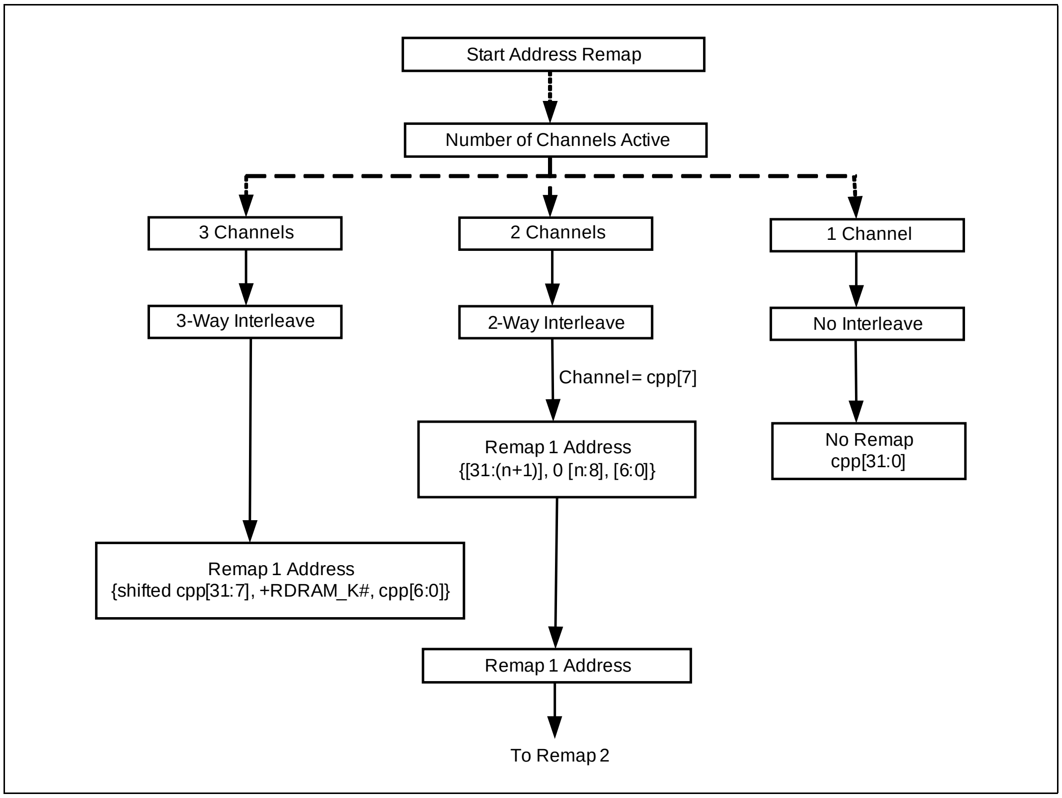
In addition to interleaving across the different RDRAM channels, address swapping may be done within a channel to distribute memory accesses across RDRAM devices or banks - this is called second-level address remapping (Remap 2) (Figure 2, “Remap 2 Address Remapping Flowchart”). This type of second-level address remapping is determined by RDRAM_CONTROL register address_remap bits [7:6].

The available options are RDRAM device optimization, bank optimization or no second-level remap. The second remap is done based on the first level remapped address. If no remap is set in address_remap bits, the second level remapped address is same as the first level remapped address.

Remap 1 address bits [30:4] are written to RDRAM_ERROR_STATUS_1 register bits [30:4]. Error_status_1 is updated for every memory transaction until a parity or ECC error is detected. If a correctable error is detected, error_status_1 is held until either an uncorrectable error occurs or until cleared by software. If an uncorrectable error is detected, error_status_1 is held until cleared by software. The value of error_status_1 can then be processed to determine the actual RDRAM address which caused the error.

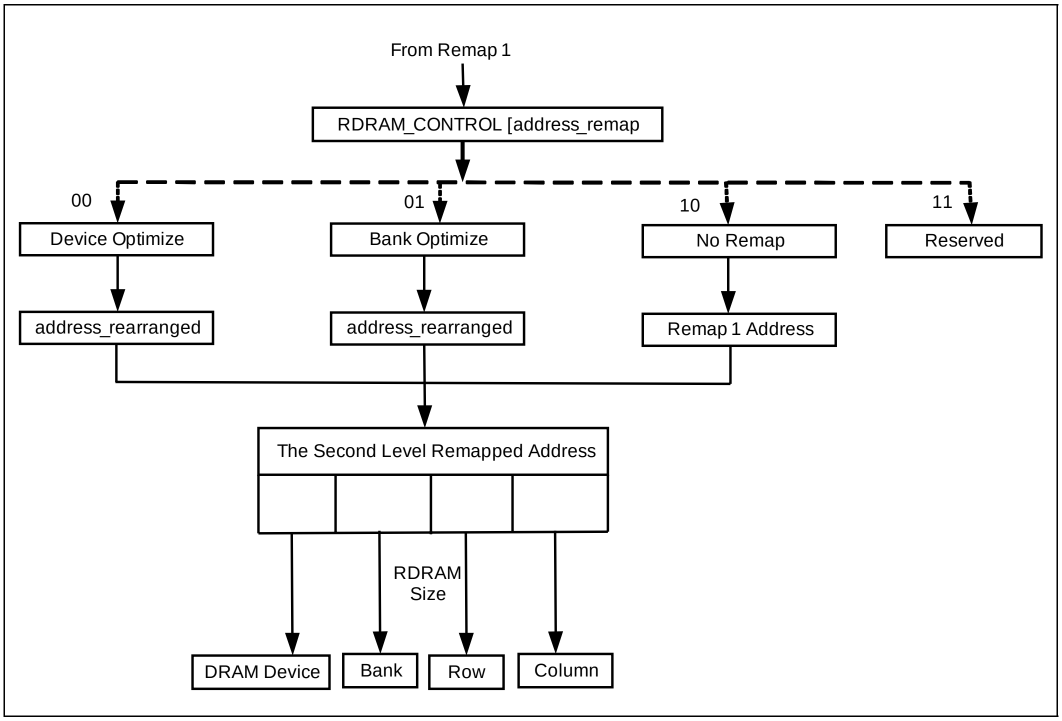
The second level remapped address is partitioned to reflect RDRAM column, row, bank and device ID.

Figure 1. Remap 1 Address Remapping Flowchart [1]



1. Verilog* syntax used for bitfield concatenation.

Figure 2. Remap 2 Address Remapping Flowchart



2.0 First Level RDRAM Address Remap

Remapping takes three forms: three channel, two channel, or single channel active.

2.1 Three Channels Active [1]

The number of active RDRAM channels is set in RDRAM_CONTROL register bits [12:11] (see the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manual*, Section 5 Control and Status Registers). If Bits [12:11] are "10" three channels are active. The channel remapping is calculated using a Modulo-3 reduction. This uses the value in address bits [30:7] and the result of the Modulo-3 operation is used to select a channel.

Channel Number = ((cpp address & ~0x7f)>> 7)%3

The highest bit (bit 31) of the CPP address should always be zero because of the 2 GB RDRAM channel size limit. Any CPP address with "1" in bit 31 is an illegal address, which would cause an error in the address remapping.

1. Verilog* syntax used in this section.

The address within the channel is remapped by shifting CPP address bits [30:7] right by the amount specified in the *Intel® IXP2800 Network Processor Hardware Reference Manual* (Section 5 DRAM, Address Rearrangement for 3-way Interleave), and then adding the contents of one of the CSRs (RDRAM_K0 to RDRAM_K11) or 0 to the shifted address. This sum is then concatenated to the lower seven bits [6:0] of the CPP address.

Remap 1 address= {(shift cpp_address[30:7] right by # bits + RDRAM_K# register), cpp_address[6:0])}

The K register to use is identified in the *Intel® XP2800 Network Processor Hardware Reference Manual* (Section 5 DRAM, Address Rearrangement for 3-way Interleave). The values of CSRs RDRAM_K0 through K11 are based on the DRAM channel size and the values are listed in the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manual*, Section 5 RDRAM. See the tables for:

- RDRAM Constants (Hexadecimal) for 3-Channel Mode Part 1
- RDRAM Constants (Hexadecimal) for 3-Channel Mode Part 2
- RDRAM Constants (Hexadecimal) for 3-Channel Mode, Part 3

Example 1.

cpp_address bits [22:7] are all "1" and bits [24:23] are not "11". This coincides with the RDRAM_K7 value identified in the *Intel® XP2800 Network Processor Hardware Reference Manual*. The remapped address [30:7] is thus equal to cpp_address bits [30:7] shifting right by 18 bits and then adding the content of RDRAM_K7. The lower bits [6:0] stay unchanged.

Remap 1 address= (((cpp_address[30:7]>>18)+RDRAM_K7)<<7) + (cpp_address & 0x7f)

2.2 Two Channels Active

If only two channels are active, channels 0 and 1 should be populated and channel 2 should be left unpopulated. CPP address bit 7 is used as channel select. If bit 7 is equal to 0, the address is mapped to channel 0 while if bit 7 is equal to 1, the address is mapped to channel 1.

Remap 1 address: {[31:(n+1)],0,[n:8],[6:0]}.

The value of "n" is determined by the RDRAM channel size specified in Table 1. This syntax identifies the concatenation of different fields of the CPP address result to form the Remap 1 output. The MSBs of Remap 1 are CPP address bit 31 to bit n+1. The next bit is 0 followed by bit n to bit 8. The seven LSBs are CPP address bits [6:0].

Example 2.

The channel size is 8 MB. Remap 1 address={[31:24], 0, [23:8], [6:0]}

The channel size is 64 MB. Remap 1 address={[31:27], 0, [26:8], [6:0]}.

Table 1. RDRAM Channel Size

RDRAM Channel Size	8 MB	16 MB	32 MB	64 MB	128 MB	256 MB	512 MB	1024 MB
N value	23	24	25	26	27	28	29	30

2.3 One Channel Active

If only one channel is active, there is no address interleaving. The value of address_remap1 is copied directly from the CPP address.

3.0 Second Level RDRAM Address Remap

In addition to interleaving across the different RDRAM channels, address swapping may be done within a channel to distribute memory accesses across RDRAM devices or banks. RDRAM_CONTROL register address_remap bits [7:6] determine the second level address remap (address_remap is defined in the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manual*).

Address_remap bits [7:6] are equal to:

00 RDRAM device optimization

01 RDRAM bank optimization

10 No remap. Remap 2 address = remap 1 address

11 Reserved

The output of the first-level remapping is used as the input to the second-level remapping.

3.1 RDRAM Device Optimization

If the number of RDRAM devices on a channel is a power of two, device optimization attempts to make consecutive blocks of CPP addresses map to different devices on the channel. The address remapping is implemented when RDRAM_CONTROL[address_remap] is set to "00".

Note: For more information see the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manual*, (Address Bank Remapping (Optimize RDRAMs) or the *Intel® XP2800 Network Processor Hardware Reference Manual* (Address Bank Interleaving).

The Address is rearranged based on the RDRAM channel size and RDRAM_CONTROL[bank_remap] (bits [18:17]). The RDRAM channel size is programmed in RDRAM_CONTROL[size] (bits [10:8]). For example, if channel size is 256 MB and bank_remap is 01, the second level remapped address is {0000, 9:14, 7:8, 27:15, 6:0}. This syntax identifies the concatenation of different fields of the Remap 1 result to form the Remap 2 output. The four MSBs of Remap 2 are zero. The next six bits of Remap 2 are Remap 1[9:14]. The next two bits of Remap 2 are Remap 1[7:8]. The next thirteen bits of Remap 2 are Remap 1[27:15] and the seven LSBs of Remap 2 are remap1[6:0]. Also, note that when the range is written with a smaller number first, e.g. [9:14], it signifies the bit-reversal of that field.

Note: This example corresponding to column 3and row 8 of the Address Bank Interleaving table in the *Intel® XP2800 Network Processor Hardware Reference Manua*l (see Section 5). The unused address bits in the table have value of 0. In this example bits [31:28] are unused.

3.2 RDRAM Bank Optimization

If the number of RDRAM devices on a channel is not a power of two, bank optimization can be used. Bank optimization attempts to make consecutive blocks of CPP addresses map to different banks within a single device on the channel. The address is remapped when RDRAM_CONTROL[address_remap] is set to “01”.

Note: For more information see the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manua*l, (in Section 5 - see Address Bank remapping (Optimize Banks)).

The Address is rearranged based on register RDRAM_CONTROL[bank_remap] (bits [18:17]). The bits [31:30] after Remap2 are equal to “00” for all bank_remap options, and the lower address bits [6:0] are not remapped. For example, if RDRAM_CONTROL[bank_remap] is programmed to 01, the second level remapped address is {00, 29:24, 9:14, 7:8, 23:15, 6:0}. Please refer to Section 3.1 for an explanation of the {,,,} concatenation syntax.

3.3 No Remap 2

The second level address remapping is optional. The remapping can be disabled by programming RDRAM_CONTROL[address_remap] to “10”. In this case, the second level remapped address is equal to the first level remapped address.

RDRAM_CONTROL[address_remap] can not be programmed to “11”, which is reserved.

4.0 RDRAM Physical Address Interpretation

RDRAM physical address information such as RDRAM column, row, bank and device ID, can be interpreted from the Remap 2 address. The interpretation is based on the RDRAM size (see the *Intel® XP2800 Network Processor Hardware Reference Manua*l for RDRAM device size options). Table 2 shows how to partition the second level remapped address to determine RDRAM device, bank, row and column.

Table 2. Partition Remap to RDRAM Physical Address

RDRAM Size (MB)	RDRAM Device ID	RDRAM Bank	RDRAM Row	RDRAM Column
64/72	[27:23]	[22:18][1]	[17:19][1]	[8:4][1]
128/144	[28:24]	[23:19]	[18:10]	[9:4]
256/288	[29:25]	[24:20]	[20:12]	[10:4]
512/576	[30:26]	[25:21]	[20:11]	[10:4]

1. These values have not yet been verified.

4.1 Parity or ECC error

Previous sections of this application note described how to remap CPP addresses to physical address. But more often the starting point is the error location address in RDRAM_ERROR_STATUS_1 [30:4]. If there is a parity or ECC error, Remap 1 address[30:4] is written to register RDRAM_ERROR_STATUS_1[30:4]. Using the address in RDRAM_ERROR_STATUS_1[30:4] the remap2 address can be calculated. The physical location with error (column, row, bank and device) is indicated in the Remap 2 address (see previous section).

Example 3.

RDRAM configuration:
RDRAM_CONTROL register value: 0x1600
DRAM device size: 288 MB
CPP address: 0x25646781

RDRAM_CONTROL register (definition is in the Programer's Reference Manual):
DRAM address remap (bits [7:6] = "00"): device optimization
DRAM channel size (bits [10:8] = "110"): 512 MB
Number of channels (bits [12:11] = "10"): 3 channels
DRAM bank remap (bits [18:17]): "00"

4.2 Remap 1

Remap 1 address = {(shift cpp_address [30:7] right by m bits + RDRAM_K# register), cpp_address[6:0])}

This address has bits [10:7], and are all "1" so it is corresponding to row 13 in the Address Rearrangement for 3-way Interleaving table in the *Intel® XP2800 Network Processor Hardware Reference Manual*. Therefore, the CPP address will shift to right by 6 bits (m=6) and RDRAM_K1 will be use in the calculation (see the Address Rearrangement for 3-way Interleaving table). The RDRAM_K1 value is 3C0000 as specified in the RDRAM Constants (Hexadecimal) for 3-Channel Mode Part 1 table in the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manua*l, (in Section 5).

Remap 1 address = (((cpp_address[30:7]>>6)+3C0000)<<7)+(cpp_address & 0x7f)

CPP_address [30:7] = 0,0100,1010,1100,1000,1100,1111 >>6

= 0000,0001,0010,1011,0010,0011 + 0011,1100,0000,0000,0000,0000 (=3C0000)

= 0011,1101,0010,1011,0010,0011 <<7

= 0001,1110,1001,0101,1001,0001,1000,0000 + 0000,0000,0000,0000,0000,0000,0000,0001 (cpp_addr&0x7f)

= 0001,1110,1001,0101,1001,0001,1000,0001 (= 1e959181)

If there is parity or ECC error, RDRAM_ERROR_STATUS_1[30:4] is 1e959181.

4.3 Remap 2

The Address remap is "00" so device optimization is used. See the *Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manual*, or the *Intel® XP2800 Network Processor Hardware Reference Manual* for details on the address remapping.

For Remap 2, the address corresponds to row 9 and column 2 (see Table 3). Therefore the Remap 2 address is {000,[7:14],[28:15],[6:0]}.

After the addresses rearrange, the Remap 2 address is 0x189e9581 (see Section 3.0).

Based on a 288 MB DRAM size, the physical address corresponds to row 4 in Table 2 (see Section 4.0).

Table 4 shows the Remap 2 Physical Address.

Table 3. Remap 2 Address

Remap 2 Address Bit	Remap 1 Address Bit	Bit Value	Notes
0	0	1	LSBs are not remapped.
1	1	0	
2	2	0	
3	3	0	1
4	4	0	
5	5	0	
6	6	0	
7	15	1	8 - the beginning of the 28:15 range.
8	16	1	
9	17	0	
10	18	1	
11	19	0	5
12	20	1	
13	21	0	
14	22	0	
15	23	1	9
16	24	0	
17	25	1	
18	26	1	
19	27	1	e
20	28	1	
21	14	0	The beginning of the 7:14 range.
22	13	0	
23	12	1	9
24	11	0	

Table 3. Remap 2 Address

Remap 2 Address Bit	Remap 1 Address Bit	Bit Value	Notes
25	10	0	
26	9	0	
27	8	1	8
28	7	1	
29	unused	0	Forced to 0.
30	unused	0	
31	unused	0	1

Table 4. Remap 2 Physical Address

	Device	Bank	Row	Column
Remap Address 2	[29:25]	[24:20]	[20:12]	[10:4]
	0C	09	1E9	58