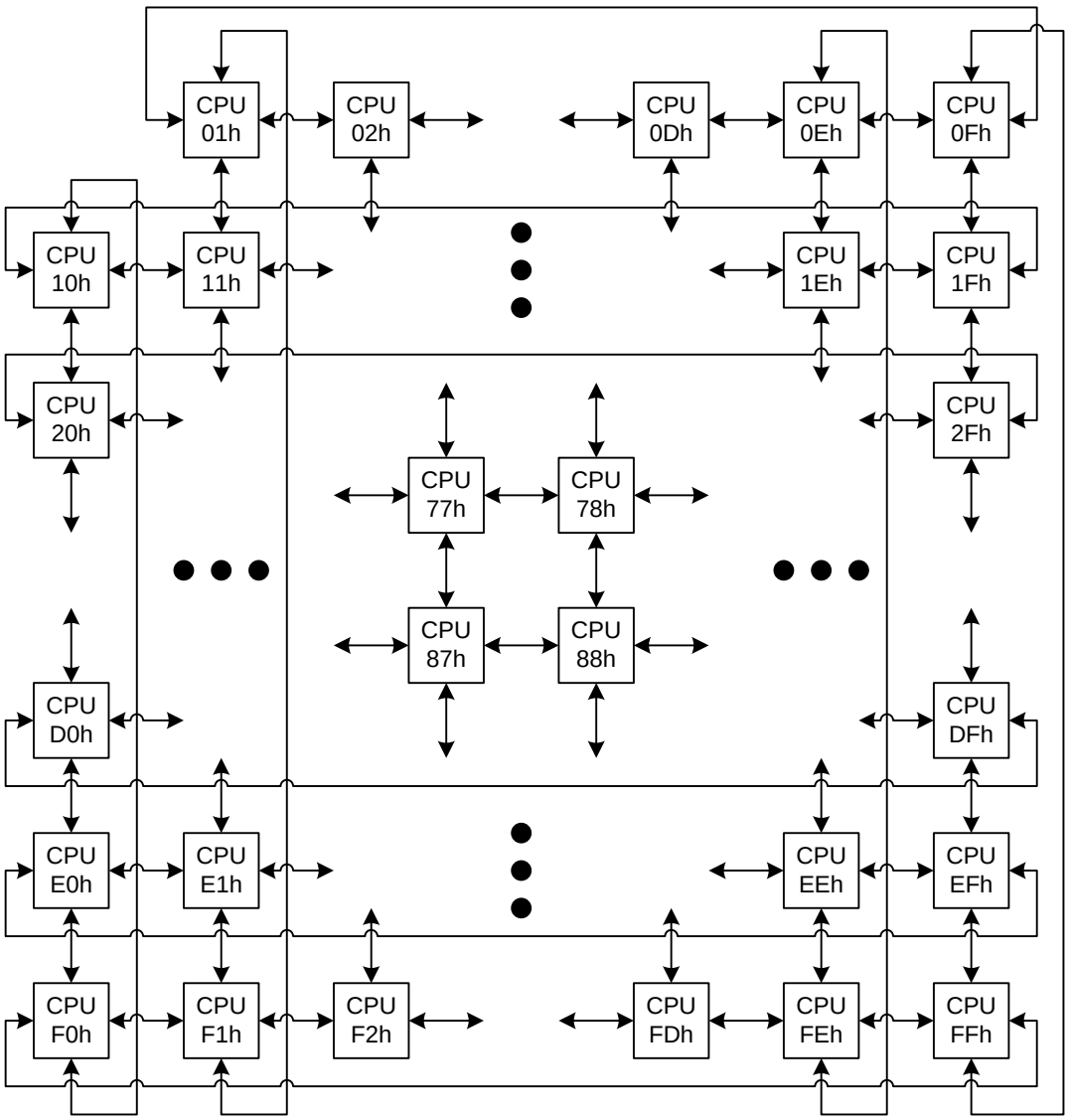
Contents

Multiprocessor

Network topology

A multiprocessor network is a two-dimensional array in which each processor has four possible directions for transmitting information. Conventionally they are called northern, eastern, southern and western. The choice of the direction of data transfer is made depending on the ratio of the processor number of the message recipient and the processor's own number.



The processor number is 8-bit and is indicated in the high-order byte of the object selector. If the byte is zero, then the selector refers to the local resource. For example, the 23000004h selector in the processor with the number 23h will point to the same object as the selector 04h.

Since the number of the processor 00h is used as an alternative pointer to accessing the local resource, the processor with the number 00h can not exist in the multiprocessor two-dimensional matrix.

The processor number is divided into two components - vertical (bits [7:4]) and horizontal (bits [3:0])

Each processor, being a network node, can not only generate its own transactions and process transactions addressed to it, but also route transit transactions generated by other processors and addressed to other processors. This function is performed by the RTE block. It has 4 32-bit MpMII interfaces - north, east, south and west. There are situations when more than one transaction needs to be

redirected to the same direction. In this case, the priority is given to the transaction whose route is longer.

Transactions

Each processor can form five types of transactions: data writing, data reading, message transmission, message confirmation and data acknowledgment. It is possible not only to write data to objects located in the memory of another processor, but also to read them. This is done by conventional machine read / write instructions (LD and ST instructions). The program code can also be located in the RAM of another processor. Such a mode of sampling program code will slow down the processor, but can be used at the stage of initializing the software.

Read / write and send messages are encapsulated in packets. There are seven types of packages.

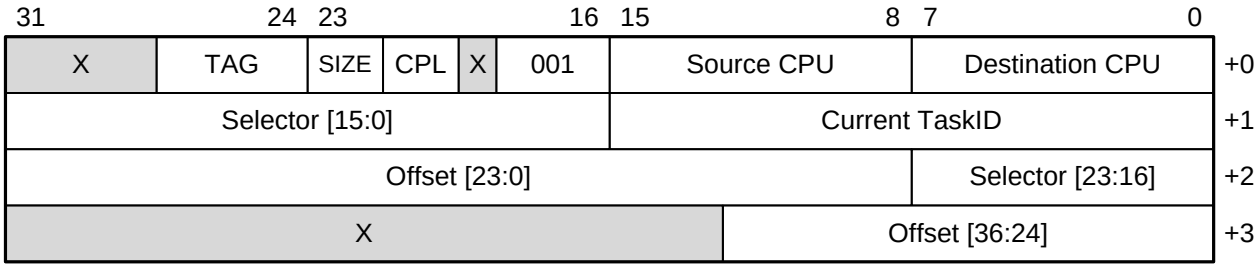
1. Reading data with a full description of the transaction parameters.
2. Reading data with a brief description of the transaction parameters.
3. Writing data with full description of the transaction.
4. Writing data with a brief description of the transaction parameters.
5. Respond to a data read request or a protection violation message.
6. Message transfer.
7. Respond to message transfer.

Data write transactions are not accompanied by acknowledgments if they were successful, but if an access error to the object was detected, the status packet with the type of violation will be returned. The following is a description of the packets and their formatting for transmission over the MpMII interfaces.

Attempts to write data to local objects of the missing processors in the network do not lead to any consequences, but read attempts can lead to a blocking of the processor requesting data. To prevent this from happening, FPU uses the emergency stop timers for all open read transactions. The timeout value is set to 15 ticks of the system timer. When the cancellation timer is triggered, the FPU sends invalid data to the core, followed by the NaN flag. If the current privilege level is other than 0, an error message is also generated. With zero CPL, no such message is generated, only the non-number in the AFR of the general-purpose register is set.

Read data with a full description of the transaction parameters

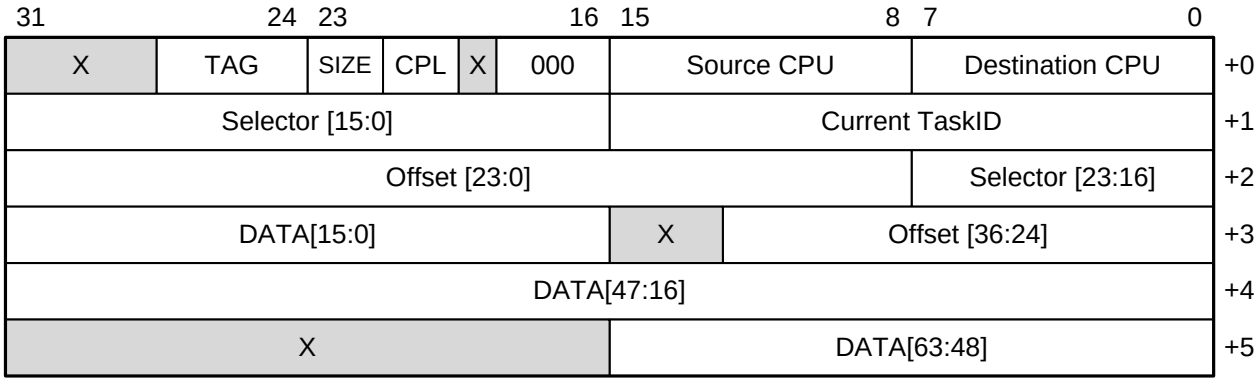
The length of the packet is 4 double words. Logical address - selector, offset and processor number are transmitted completely.



The CPL and TaskID are also transferred, which are necessary to verify the access to the object in the slave processor. The SIZE field specifies the length of the requested data element - 8, 16, 32 or 64 bits.

Write data with full description of the transaction

The length of the packet varies 4, 5 and 6 double words, depending on the width of the transmitted data element.

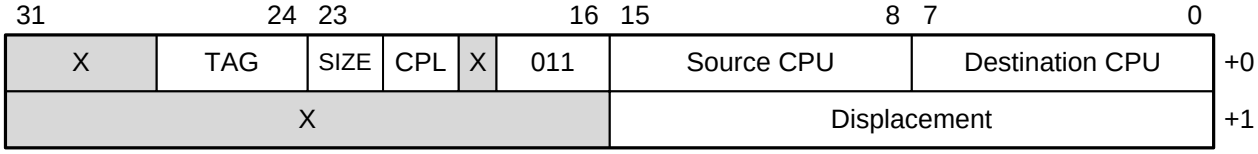


TAG - the transaction tag is assigned to the FPU of the processor that created the transaction. In the special memory buffer for 16 cells in the FPU, addressed by the transaction tag, the values of the object selector and the TaskID are stored. Transactions with the same values of the object selector and TASK ID will be executed with the same value of the tag. The transaction tag is used to generate short packets for repeated calls to the object.

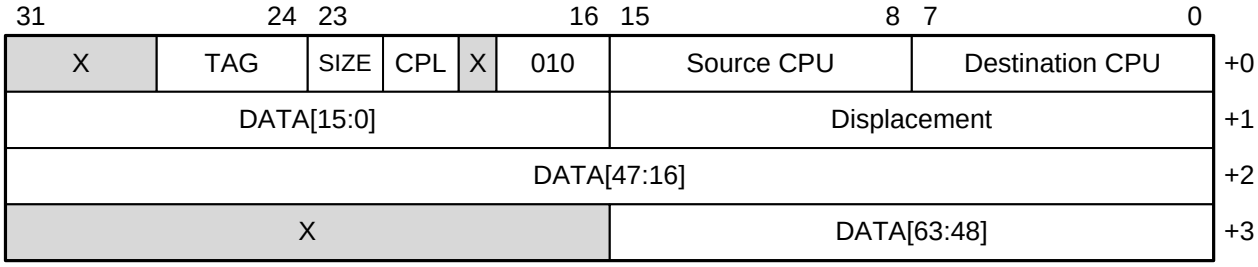
Read and write data with a brief description of the transaction

By accepting the first transaction with a full description of the parameters, the slave processor stores the selector and offset in the buffer addressed by the transaction tag and the processor number of the transaction initiator. After the first transaction with a full description of all the necessary parameters, a short form of the transaction presentation becomes possible. In the short form, a transaction tag and a 16-bit increment of the offset are specified, which is complemented by up to 37 bits with a sign bit, which allows addressing the memory cells in the object both in the direction of increasing addresses and decreasing relative to the previously specified offset or previously calculated. The slice in the cache of the slave processor is updated with each new short transaction.

Short read transaction

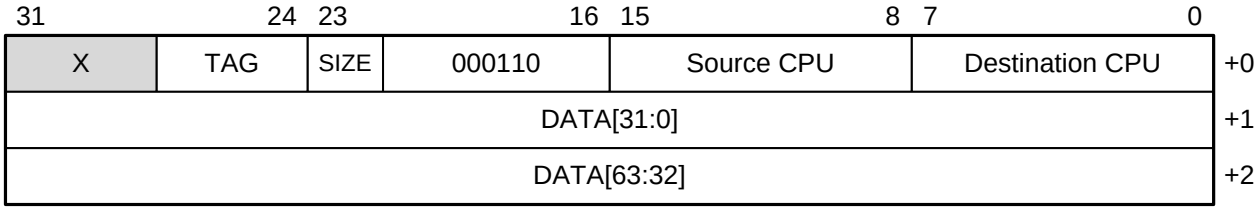


Short write transaction



Both in the master processor and in the slave, the values of the offset used in the previous access to the object are stored. The next request, if it is located no further than 32767 bytes, can be executed with a short description of the transaction. If the difference between the previous offset and the new one exceeds this value, the transaction is described in a long format, with a full indication of the new offset. Since the tag is 4-bit, each processor can open up to 16 transactions simultaneously. As necessary, least used cache cells are replaced with new transactions. Each processor must support up to 256 * 16 = 4096 cache slots for storing incoming transaction parameters, opened by other network processors.

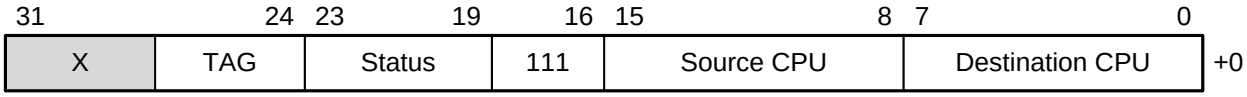
Answers to data read requests



The transaction tag is intended to identify the original read transaction in the processor that previously generated a network read request for data.

Violation report packets

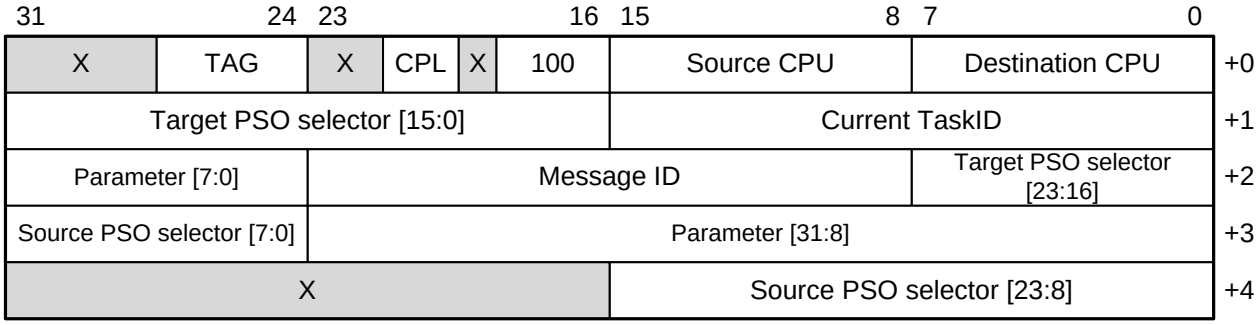
If the slave processor, in the process of accessing the local object, detects any violation of the rules of the object protection system, it returns to the processor that initiated the data transfer an error message.



The Status field encodes information about the type of error. The lower 3 bits are required to contain a value other than 000b, so that the packet is considered an error message packet.

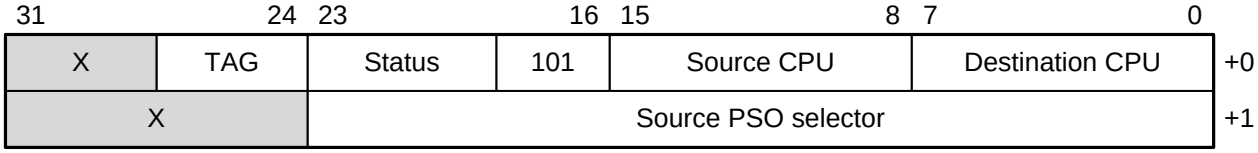
Message packets

Between the processors, it is possible to send regular and system messages. Messages are transmitted using separate frames containing the TaskID task group identifier, the process PSO selector to which the message is sent, the PSO selector of the process that sent the message, and the 32-bit message parameter.



Message acknowledgement packets

These packets are always generated in response to the message packet. The slave processor notifies the source of the message whether the message is received in the receiving message queue of the message or when processing errors occurred.



If STATUS = 0, then the message was successfully added to the process queue to which the message is addressed. The returned PSO selector is placed in the system error queue if Status is not 0 and is used to identify the process that caused the error.