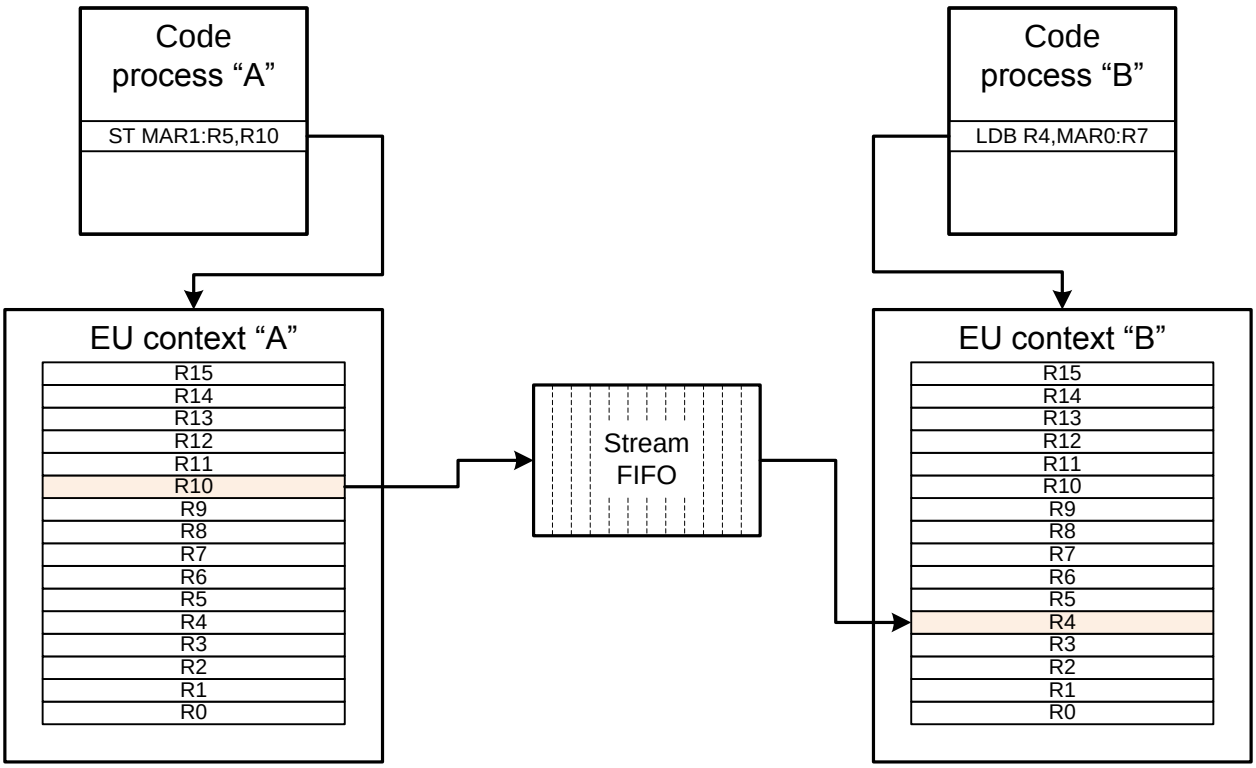
Contents

Hardware data streams

Function

Streams are designed to easily and quickly transfer data from one process to another with minimal application of software negotiation of the data transfer process. In the simplest case, coordination may not apply at all. A stream is actually a FIFO queue, in which the sequence of data output is the same as the sequence of loading them.



Data-sharing processes can work on the same processor as well as on different processors of a multiprocessor network. If both processes are running on the same core, then the local stream controller is used. If the processes run on different cores, the work of the stream is always supported by the processor's stream controller, which executes the code that reads the data from the stream.

Temporarily unreadable data streams store their data in buffers allocated in RAM.

The stream is accessed exactly by the same processor instructions as accessing any data object located in the local memory. The EU address translation unit or FPU block determines the type of the object descriptor to access the stream and directs the read or write transaction to the stream controller. When writing data to the stream, it is sufficient to specify only the selector, the offset is ignored, since the stream controller itself will determine where to put the data. When reading data, the address bit A0 has a value and is used to indicate the type of information to be read.

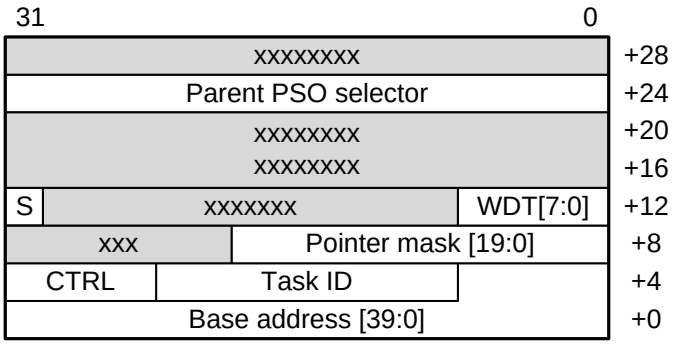
Only one process can extract data from the stream. Several processes can fill the flow, but in this case, the recipient must have a mechanism for recognizing which process each individual data item is received from.

The data elements transmitted through the stream have a fixed bit width, the value of which is set in the stream descriptor. You can pass a data item of a different bit length to the stream than it is set in the stream descriptor, but either the high-order bits will contain undefined data (if an element of a lower bit length is transmitted) or the high-order bits will be lost (if an element of greater bit width is transmitted).

It does not matter which LD instruction (B / W / D / Q / O) the stream is read from. The field Size of the data in the AFR associated with a general-purpose register will be set according to the bit width specified in the stream descriptor.

A process that receives data from a stream can learn the state of the stream. To do this, you need to read the stream with an offset containing 1 in bit A0. At A0 = 1, the counter of the data elements accumulated in the FIFO stream is subtracted from the stream. When A0 = 0, an attempt is made to read the data.

The stream descriptor contains 4 values that control the flow.



The base address determines the position in the local buffer memory for temporary storage of data items not yet read from the stream.

The pointer mask is used to reset the write and read counters by the logical AND operation of the pointer and mask. The mask indirectly determines the length of the buffer, expressed in data elements, but not in bytes. The length of the buffer in

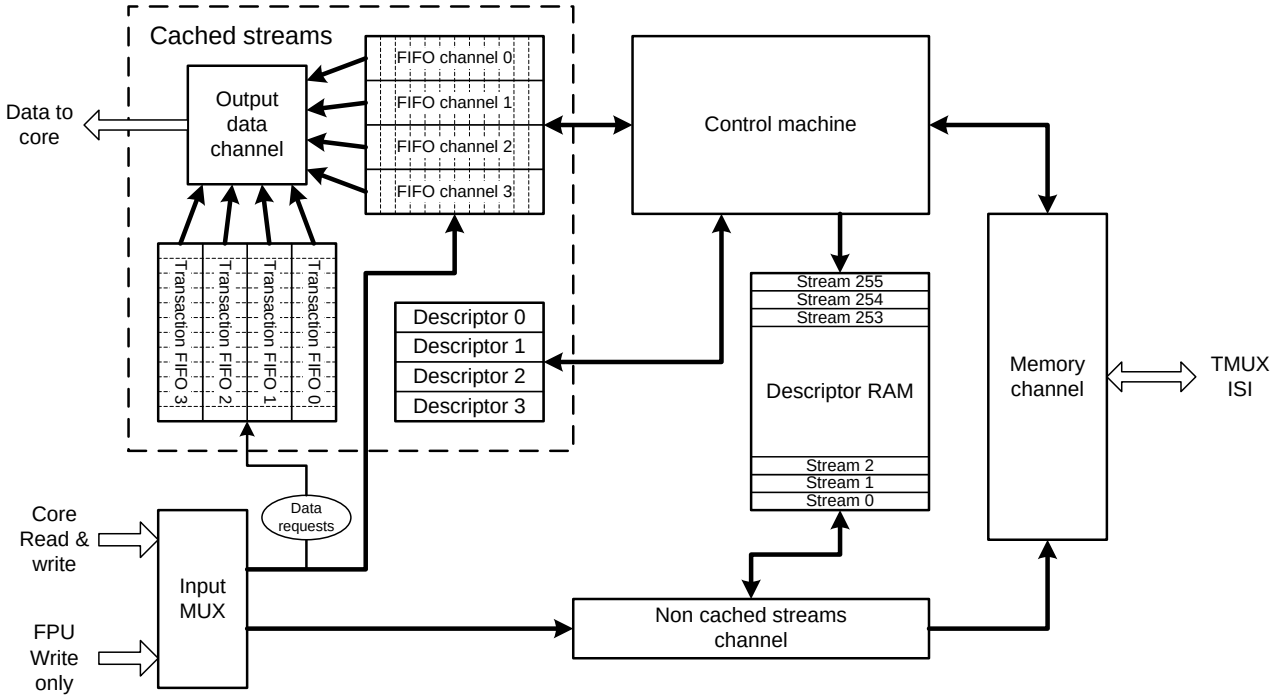
bytes is calculated by the formula (2 ^ N) * (2 ^ Size), where N is the number of bits of the mask set to 1, Size is the number from the Size field of the stream descriptor.

S - field Size, which determines the bit capacity - 8/16/32/64 bit.

WDT is the timeout value used to force a transaction to read data from the stream if there is no data. When the transaction ends with a timeout, the not-a-number code is returned to the general-purpose register of the core. The program can analyze the NF flag after reading the stream to determine the validity of the data.

Stream controller equipment

The controller supports up to 256 parallel running streams. The lower 8 bits of the object selector are used to address the descriptor buffer containing the stream descriptors and the current state of the streams. The validity of the descriptor is checked by matching the bit [23: 8] of the selector specified in the transaction and the stored selector bits from the descriptor buffer cell.



At the same time, the core can read data from 4 streams. The descriptors of the 4 streams that the core uses for reading are cached separately and for them there are 4 FIFO hardware buffers with a length of 16 cells each. Four FIFOs store read requests for data on cached streams. When the data arrives to the stream or when the timeout ends, the transaction leaves the FIFO transaction and the data or not-a-number is returned to the general-purpose register of the core.

As data is extracted from the FIFO, the data of their external buffer is downloaded, if any, in the external buffer.

If you write data to a stream that is cached at a given time and the external buffer is empty, the data is sent to the internal queue. If the external buffer is not empty, then the data is written to the external buffer.

The data transferred to the streams, which are temporarily not used by the core and not cached, are located in external FIFO-buffers located in the RAM. These data pass through the channel of non-cached streams.

Stream status

No data. There is a process that extracts data from the stream, but the data does not flow to the stream. In this case, all transactions for reading data from the stream are terminated after the timeout set for this thread expires. The stream is cached and the first data element that enters the stream entry will be written to the internal FIFO. If there is a read request in FIFO transactions, the incoming data element will be immediately sent to the core.

Waiting for data. The data arrives at a speed lower than the speed of their reading from the stream, but the mechanism for completing transactions by timeout does not work. In this mode, the data arriving at the input of the stream passes through one of the four hardware FIFOs and is transferred to the general-purpose register of the core that is waiting for the data.

Pause the stream reader. The hardware queue in the cache is filled with data and when it is completely full, the stream controller will begin to store the incoming data in the circular buffer of the RAM.

Resume the stream reading. If the stream is cached in one of the four hardware channels, the data is first retrieved from the corresponding hardware queue. As the queue is empty, the thread controller starts the process of loading data into FIFO from the RAM.

Opening a new thread on reading. When the core starts to read data from an un-cached stream, the controller checks for the presence of a valid flow descriptor in the descriptor buffer, and if it exists, then it is copied to one of the 4 positions of the stream cache and the subsequent loading of the FIFO data from the corresponding RAM buffer. The stream descriptor is loaded from the descriptor table into the selected cache channel in the event that the stream was not previously used at all. If by the time the new read stream was opened, all 4 channels were used in the stream cache, then the controller selects the cache channel by which the access was the later than others. Such a flow is temporarily removed from the cache. In this case, the data from the hardware FIFO is stored in the RAM only if they entered the hardware FIFO, bypassing the recording to the external buffer of the stream.

Stream overflow. It is observed when the reading speed of the stream is less than the write speed. In this case, the hardware queue will be filled in the stream cache first, and then the circular buffer will be filled. In this case, all incoming data to the stream will be ignored until at least one free cell is found in the buffer.

Opening the stream by recording. The first write transaction to the stream causes the descriptor to be sampled and placed in one of 256 descriptor RAM cells and the data begins to be written to the external buffer.