Contents

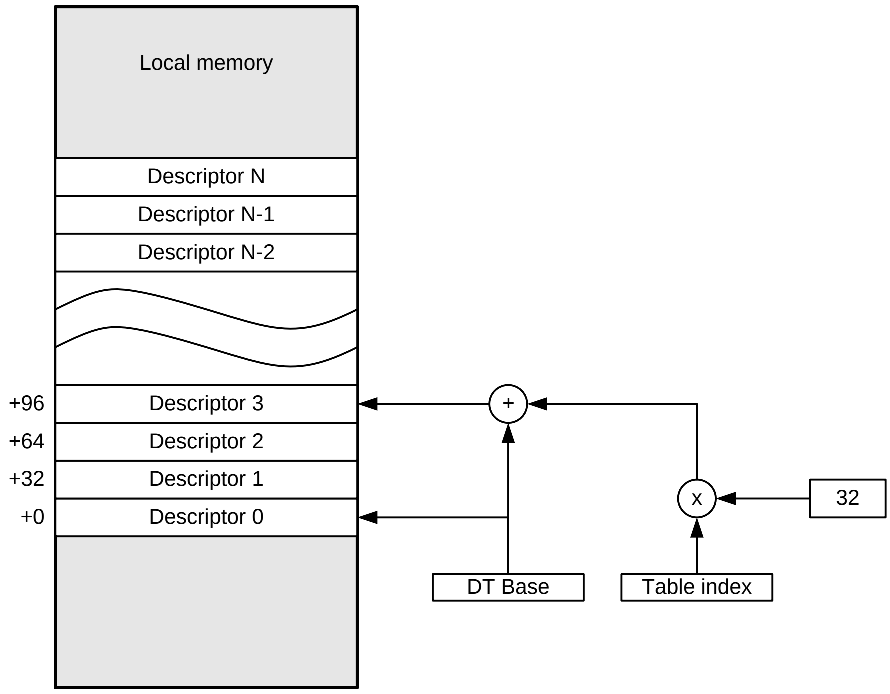
Address translation and memory protection

Objects and object descriptors

Data and program code are stored in RAM in structures called objects. Each object is described by a set of properties that determine the location of the object in memory and the modes of access to it. The object can be accessed via the object selector. The object selector is a 32-bit value used to retrieve an object descriptor from the descriptor table.

31 24	23 0
CPU	Table index

The object descriptor specifies the rules for accessing the object and its physical location. Descriptors of objects are located in a special table - the descriptor table. The lower 24 bits of the selector determine the descriptor number in the table.



The high byte of the selector determines the number of the processor in the local memory of which the object is located. The 00h number of the processor number is reserved and is used to indicate the processor on which the code is executed. If the CPU of the selector matches the CPUNR from the control register, this also indicates the addressing of the object located in the memory of the current processor.

The descriptor table can contain four types of descriptors:

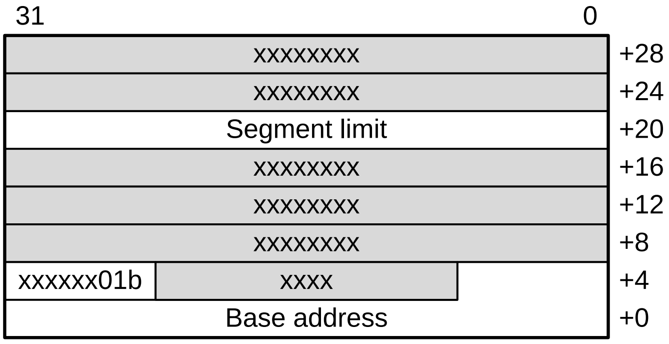
- Empty entry points where actual object descriptors can be placed;
- Descriptors of free memory segments that can be allocated to objects;
- Descriptors of objects or segments of objects;
- Data stream descriptors.

Empty entry point

31		0	
xxxxxxxx			+28
xxxxxxxx			+24
xxxxxxxx			+20
xxxxxxxx			+16
xxxxxxxx			+12
xxxxxxxx			+8
xxxxxx00b	xxxx		+4
xxxxxxxxxx			+0

Free positions are indicated by the zero bit state [1:0] of the control byte. The remaining fields can contain any information and do not matter for the operation of the memory allocation system built into the processor.

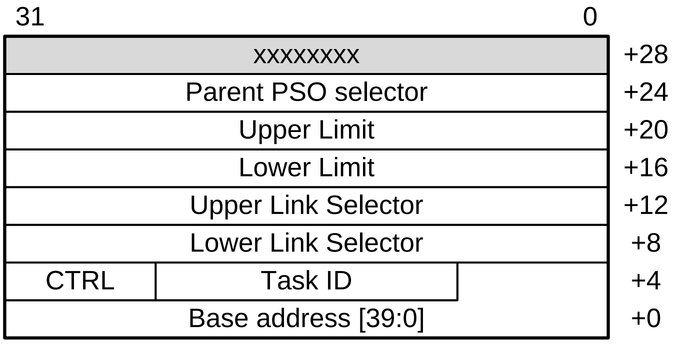
Free memory block descriptor



31		0	
xxxxxxxx			+28
xxxxxxxx			+24
Segment limit			+20
xxxxxxxx			+16
xxxxxxxx			+12
xxxxxxxx			+8
xxxxxx01b	xxxx		+4
Base address			+0

In the descriptor, only 2 fields are used - the base address and block length (upper limit), expressed in 32-byte paragraphs.

Object descriptors



31		0	
xxxxxxxx			+28
Parent PSO selector			+24
Upper Limit			+20
Lower Limit			+16
Upper Link Selector			+12
Lower Link Selector			+8
CTRL	Task ID		+4
Base address [39:0]			+0

The descriptor contains:

- 40-bit base address of the 32-byte paragraph. This address must be shifted 5 bits to the left to obtain the byte address.
- Task ID is a 16-bit value that is used to check whether the process can access the object. If the TaskID of the process is not zero and the TaskID of the object is not zero, then both values are compared and if they are not equal, access to the object is blocked and a system error is generated.
- CTRL – control byte.
- Lower Link Selector - the selector of transition to the bottom segment of the object. Used if the offset specified in the transaction is less than the Lower Limit value.
- Upper Link Selector – the selector of transition to the upper segment of the object. Used when the offset is greater than or equal to Upper Limit.
- Lower Limit - the lower limit of a segment or object. The value is expressed in 32-byte blocks and compared with the bits [36: 5] of the offset when testing the limits of a segment or object. If the offset value is less than the lower limit, then access to such object or segment of the object is impossible.
- Upper Limit - the upper limit of a segment or object at which access to a segment or object is no longer possible. The limit is compared with the bits [36: 5] of the offset.

- Parent PSO selector - the PSO selector of the process to which the given object or segment belongs. The processor equipment independently sets the desired value to this position when executing the MEMALLOC instruction. This field can be used to clear memory from objects created by a process whose operation is terminated.

The format of the object control byte.

7	6	5	4	3	2	1	0
X	NE	WE	RE	DPL		1	0

Bits [1: 0] define the type of descriptor - 2 - object. DPL - Defines the privilege level required to access the object. If CPL is numerically less than or equal to DPL, then access to the object is allowed. RE - allows reading an object, if 1. WE - allows write to the object, if 1. NE - allows access to an object from another processor, if 1.

Stream descriptors

<table>
<tr><td colspan="2">31</td><td colspan="2">0</td><td></td></tr>
<tr><td colspan="4">xxxxxxxx</td><td>+28</td></tr>
<tr><td colspan="4">Parent PSO selector</td><td>+24</td></tr>
<tr><td colspan="4" rowspan="2">xxxxxxxx
xxxxxxxx</td><td>+20</td></tr>
<tr><td>+16</td></tr>
<tr><td>S</td><td colspan="2">xxxxxxx</td><td>WDT[7:0]</td><td>+12</td></tr>
<tr><td colspan="2">xxx</td><td colspan="2">Pointer mask [19:0]</td><td>+8</td></tr>
<tr><td>CTRL</td><td colspan="2">Task ID</td><td></td><td>+4</td></tr>
<tr><td colspan="4">Base address [39:0]</td><td>+0</td></tr>
</table>

The descriptor contains:

- 40-bit base address of the 32-byte block. This is the base address of the circular buffer in which the FIFO queue is organized when data can not be retrieved from the stream or the stream is not temporarily used.
- Task ID is a 16-bit value that is used to check whether the process can access the object.
- Pointer mask - used to mask the read and write pointer. The mask is superimposed by a logical "AND". For example, if the mask is 0FFFh, it means that the buffer size is 4096 data items, but not bytes. The size of the buffer in bytes will be L = S * 2 ^ (Pointer mask + 1) - the length of the buffer multiplied by the dimension of the data element.
- WDT - the field specifies the timeout for transactions to read data from the stream. Each stream read transaction for which there is no data in the stream expects the data to arrive during [WDT] ticks of the system timer, after which the transaction is terminated forcibly. The value of NaN is passed to the kernel instead of the data.
- S - 2-bit field, which determines the bit width of the data element. The data element can be 8, 16, 32 or 64 bit.

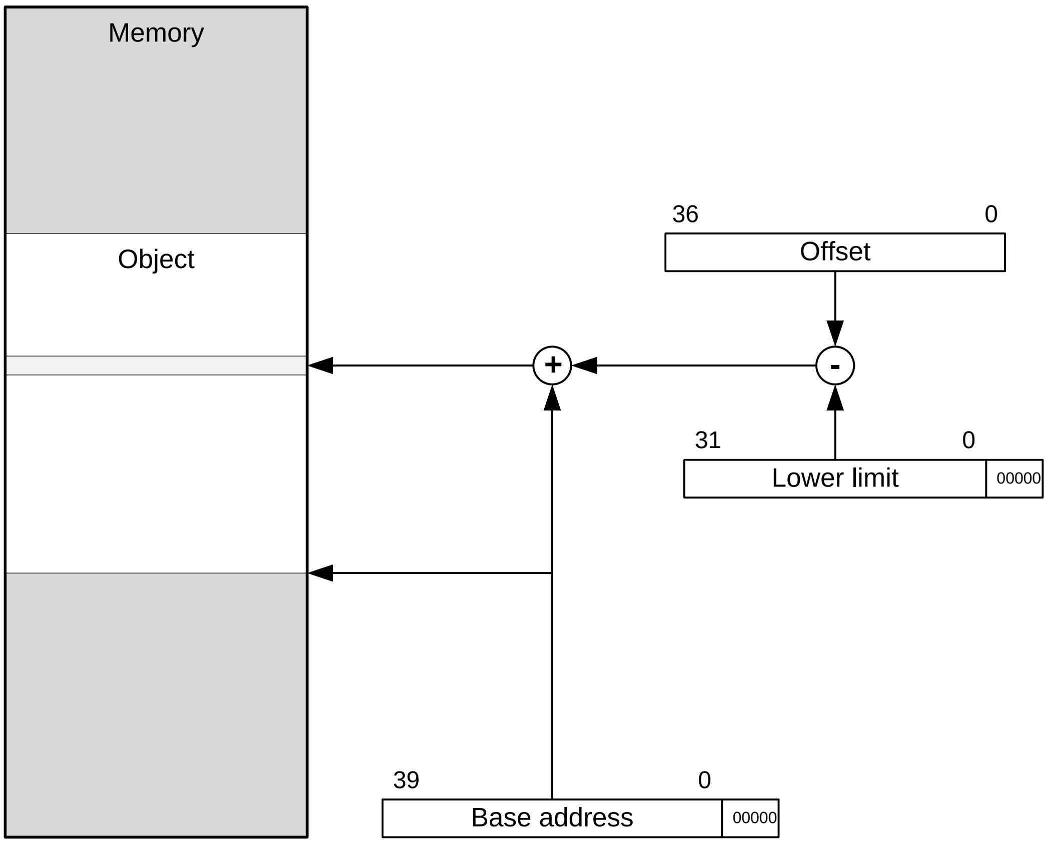
Control byte format.

7	6	5	4	3	2	1	0
X	NE	WE	RE	DPL		1	1

Bits [1:0] Type = 3 define the type of the descriptor – the stream descriptor. NE - allows access to a stream from another processor, if 1.

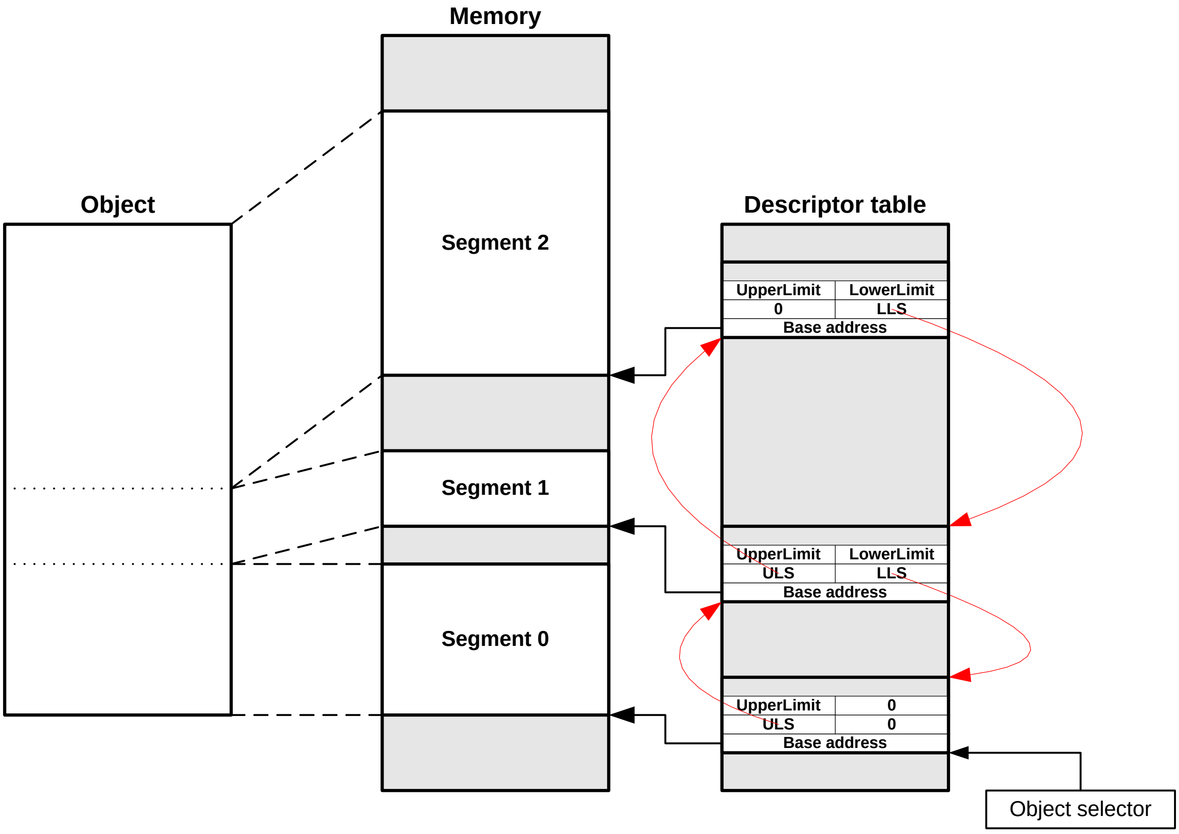
Address translation

The translation of a logical two-component address consisting of an object selector and an offset is performed using object descriptors. Access to an object / segment is allowed if the offset is greater than or equal to the lower limit and is strictly less than the upper limit. To form the physical address, the value of the lower limit extended to the right by five zero bits is subtracted from the offset and then the resulting value is added with a 40-bit base address also complemented by the 5th zero bits on the right. In fact, 5 bits are copied from the original offset to the resulting physical address without changing.



Object segmentation mechanism

With the help of two limit values, the mechanism of object segmentation is supported, the separation of one large object, for which there is no continuous block of memory, into several segments. Segments are linked together by means of link selectors. The lower link selector is used to select the descriptor if the offset is less than the lower limit. The upper link selector is used when the offset is greater than or equal to the upper limit.



If the offset is not within the acceptable range for the selected descriptor, then the next descriptor is sampled and so on until a segment is found that is suitable for the range of offsets. End segments in the chain are marked by zero selectors of communication. If, for example, the upper link selector is 0 and the equality or excess of the object's upper limit is detected by an offset, a violation of the access mode to the object is detected.

Object protection

Protection by privilege level. Process can't access to the object, if his CPL numerically greater than DPL field from the descriptor control byte.

Protection by TaskID. One or several process may have unique identifier – TaskID. Current TaskID from the CSR register is compared with TaskID from the descriptor if none of them is equal to 0. Access to object will be denied if process TaskID not equal to descriptor TaskID.

Protection by access type. Object can be closed for write operations (WE=0) or read operations (RE=0), or not accessible from multiprocessor network (NE=0).