NFP-SDK-6.0.x to NFP-SDK-6.1 Software Migration Guide

NFP SDK version 6.1 Preview

Micro-C

- Proprietary and Confidential -

Product code
070-00008-000

NFP-SDK-6.0.x to NFP-SDK-6.1 Software Migration Guide: Micro-C

COPYRIGHT

No part of this publication or documentation accompanying this Product may be reproduced in any form or by any means or used to make any derivative work by any means including but not limited to by translation, transformation or adaptation without permission from Netronome Systems, Inc., as stipulated by the United States Copyright Act of 1976. Contents are subject to change without prior notice.

WARRANTY

Netronome warrants that any media on which this documentation is provided will be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of shipment. If a defect in any such media should occur during this 90-day period, the media may be returned to Netronome for a replacement.

NETRONOME DOES NOT WARRANT THAT THE DOCUMENTATION SHALL BE ERROR-FREE. THIS LIMITED WARRANTY SHALL NOT APPLY IF THE DOCUMENTATION OR MEDIA HAS BEEN (I) ALTERED OR MODIFIED; (II) SUBJECTED TO NEGLIGENCE, COMPUTER OR ELECTRICAL MALFUNCTION; OR (III) USED, ADJUSTED, OR INSTALLED OTHER THAN IN ACCORDANCE WITH INSTRUCTIONS FURNISHED BY NETRONOME OR IN AN ENVIRONMENT OTHER THAN THAT INTENDED OR RECOMMENDED BY NETRONOME.

EXCEPT FOR WARRANTIES SPECIFICALLY STATED IN THIS SECTION, NETRONOME HEREBY DISCLAIMS ALL EXPRESS OR IMPLIED WARRANTIES OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to some users of this documentation. This limited warranty gives users of this documentation specific legal rights, and users of this documentation may also have other rights which vary from jurisdiction to jurisdiction.

LIABILITY

Regardless of the form of any claim or action, Netronome’s total liability to any user of this documentation for all occurrences combined, for claims, costs, damages or liability based on any cause whatsoever and arising from or in connection with this documentation shall not exceed the purchase price (without interest) paid by such user.

IN NO EVENT SHALL NETRONOME OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE DOCUMENTATION BE LIABLE FOR ANY LOSS OF DATA, LOSS OF PROFITS OR LOSS OF USE OF THE DOCUMENTATION OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, PUNITIVE, MULTIPLE OR OTHER DAMAGES, ARISING FROM OR IN CONNECTION WITH THE DOCUMENTATION EVEN IF NETRONOME HAS BEEN MADE AWARE OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL NETRONOME OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE DOCUMENTATION BE LIABLE TO ANYONE FOR ANY CLAIMS, COSTS, DAMAGES OR LIABILITIES CAUSED BY IMPROPER USE OF THE DOCUMENTATION OR USE WHERE ANY PARTY HAS SUBSTITUTED PROCEDURES NOT SPECIFIED BY NETRONOME.

Revision History

Date	Revision	Description
14 September 2018	000	Initial release

Table of Contents

1. Introduction 6
1.1. Overview 6
1.2. Issues addressed in standard library and flowenv 6

2. Changes 8
2.1. Added *‘cmd_’* prefixes 8
2.2. File cleanup 8
2.3. Moved components 8
2.4. Removed ptr32/ptr40 shorthand defines 8
2.5. Removed source inclusion 9
2.6. Ensured header and source files can compile on their own 9
2.7. Support for updated NFCC -ng 10
2.8. Added fixed width declspec definitions 10
2.9. Integral data types 11
2.10. Typical pitfalls discovered when converting projects 11

..... 12
A.1. Examples of *'cmd_'* and *'ptrXX'* suffixes 12

3. Technical Support 25

List of Tables

2.1. Defaulting declspecs in NFCC Mainline SDK-6.0.X 10
2.2. Defaulting declspecs in NFCC NG SDK-6.X 10

1. Introduction

1.1 Overview

The micro C standard library needed to undergo some significant changes from version 6.0.x to 6.1.x. Most of these changes were necessitated to use standard library with flowenv in the same application while others were to address long standing requests.

1.2 Issues addressed in standard library and flowenv

The following list describes the issues which were present in standard library and flowenv in previous versions of the SDK.

- Duplicate functions in intrinsic.c (stdlib) and me.c (flowenv) , same names and implementations.
- Unused #define statements from nfp_compatibility.h
- Overlapping functions in standard library intrinsic.c and flowenv me.c
    - __ME(void)
    - int signal_test(volatile SIGNAL *sig_ptr)
    - int bit_test(unsigned int data, unsigned int bit_pos)
    - many of the mem/cls functions in flowenv overlap with standard library
    - mem_read_atomic
    - mem_write_atomic
    - mem_incr32
    - mem_decr32
    - mem_incr64
    - mem_decr64
    - mem_add32_imm
    - mem_add64_imm
    - cls_read_be
    - cls_write_be
    - cls_hash_mask
- Indirect referencing: flowenv only supports NFP6000 while the standard library NFP6000/3200.
- Amount of macros in standard library, far fewer in flowenv.
- Most functions and tests in standard library focus on intrinsic memory functions, imem, emem, ctm, cls.
- There are no data flow processing functions in standard library which necessitates interoperability with flowenv.

- No NG support in the standard library.

It was well known and understood at the time that things would break if we wanted to use standard library together with flowenv. If it is decided to move from 6.0 to 6.1 the changes could be substantial, of course it all depends on the application.

Without knowing the applications use of the standard library it is difficult to provide a full list of all the changes required for the application.

The following sections should ensure developers are aware of changes and when build problems appear they should be able to and resolve them.

2. Changes

The following sections describe the changes addressing the issues identified.

2.1 Added *‘cmd_’* prefixes

Flowenv and the 6.0.x version of standard library shared the same namespace for functions. To overcome this issue “cmd_” prefixes were added to most of the functions defined in the standard library . This allows both libraries to be compiled in conjunction with one another. Projects which are using the the standard library will also have to add cmd_* prefixes to their function calls. If this is not added then the compiler will pop out a warning stating that you defined a function implicitly.

An example of adding cmd_ prefixes: mem_read32_ptr32 -> cmd_mem_read32_ptr32.

2.2 File cleanup

The following set of files were removed from the standardlibrary. Where it was required their contents were allocated to a more sensible file:

- nfp_internal.h
- nfp_compatibility.h
- nfp_legacy.h
- nfp_math.h & nfp_math.c

2.3 Moved components

Flowenv and ng_nfd was moved to the following locations in the toolchain:

- nfp-sdk-6.1.0-preview/p4/components/flowenv -> nfp-sdk-6.1.0-preview/components/flowenv.
- nfp-sdk-6.1.0-preview/p4/components/ng_nfd -> nfp-sdk-6.1.0-preview/components/ng_nfd.

2.4 Removed ptr32/ptr40 shorthand defines

nfp_compatibility.h contained shorthand defines for cam/cls/mem/pcie commands. These shorthand annotations called their respective function ptr32/ptr40 variants depending upon if nfcc the “__ADDRESSING_DEFAULT_40BIT” define. It was found that this can lead to ambiguous design and was removed.

These shorthand defines must be replace with their actual function calls in order to compile projects. Appendix A contains a list of defines which were removed and also shows how the function will look with both the prefixes and suffixes added. If a user is ever unsure which suffix to use (‘_ptr32’ or ‘_ptr40’), then using ‘_ptr40’ is the safer choice.

2.5 Removed source inclusion

Intrinsic.c used to include the source the files listed below, however the inclusion was removed to make the standard library modular and to improve compile times:

- #include "nfp_override.c"
- #include "nfp_arm.c"
- #include "nfp_cls.c"
- #include "nfp_cluster_target.c"
- #include "nfp_crypto.c"
- #include "nfp_mem.c"
- #include "nfp_pcie.c"
- #include "nfp_ila.c"
- #include "nfp_nbi.c"
- #include "nfp_math.c"

The functions inside these source files can still be used, as long as the source code is added during the build process.

As an example, if a user wanted to make use of the cls functions then they would have to include the source file nfp_cls.c during the build process and include nfp_cls.h where they want to access the aforementioned functions. Below is an example of including the source file during the build process:

```
nfcc -chip nfp-6xxxc-b0 c:\NFP_SDK_6.x-devel\components\standardlibrary\microc\src\libc.c
-FIc:\NFP_SDK_6.x-devel\components\standardlibrary\microc\src\intrinsic.c
-FIc:\NFP_SDK_6.x-devel\components\standardlibrary\microc\src\rtl.c
-FIc:\NFP_SDK_6.x-devel\components\standardlibrary\microc\src\nfp_cls.c …
```

2.6 Ensured header and source files can compile on their own

This is also part of general clean up duty and links up closely with the previous point. Interdependencies were discovered between header and source files. Each source and header in the standard library was updated so that they can compile(or at least be included in the latter case) on their own.

It might be required that projects revisited their inclusions when they start receiving warnings like below, this warning will also be displayed when the source file wasn’t included during the build process:
"*warning: #266 function declared implicitly* "

2.7 Support for updated NFCC -ng

Default values for implicit declspecs has been changed between NFCC NG and NFCC mainline. In order to protect against incorrect memory usage, declspecs must be made explicit. This change will not have any effect on an application if NFCC mainline is still used. If decls specs are not made explicit when NFCC -ng is used then unintended memory regions might be accessed.

Below are examples of how declspecs which are reliant on implicit declarations can lead to issues when using -ng and how to fix them:

Table 2.1. Defaulting declspecs in NFCC Mainline SDK-6.0.X

Implicit declspec	Compiled as
__addr40	__addr40 __mem
__addr32	__addr32 __mem
__mem	__addr40 __imem
__ctm	__addr40 __ctm

Table 2.2. Defaulting declspecs in NFCC NG SDK-6.X

Implicit declspec	Compiled as	Explicit declaration
__addr40	compile error	__addr40 __mem
__addr32	compile error	__addr32 __mem
__mem	__addr40 __imem	__addr40 __mem
__ctm	__addr32 __ctm	__addr40 __ctm

2.8 Added fixed width declspec definitions

Convenient __declspec(...) definitions have been added to nfp.h to improve readability in projects. This can be used for -ng explicit declarations. The following is a list of what has been added:

```
#define __mem32          __declspec(addr32 mem)
#define __mem40          __declspec(addr40 mem)
#define __cls32          __declspec(addr32 cls)
#define __cls40          __declspec(addr40 cls)
#define __ctm32          __declspec(addr32 ctm)
#define __ctm40          __declspec(addr32 ctm)
```

2.9 Integral data types

Integral data types were replaced with standard gcc type naming. The following type defs were added in stdint.h to facilitate this:

```
typedef signed char int8_t; typedef short int int16_t; typedef int int32_t; typedef long
long int int64_t; typedef unsigned char uint8_t; typedef unsigned short int uint16_t;
typedef unsigned int uint32_t; typedef unsigned long long int uint64_t;
```

For more information see standardlibrary/microc/include/stdint.h

2.10 Typical pitfalls discovered when converting projects

The following is a list of typical pitfalls which has been discovered when converting projects:

- Source files omitted during build process.
- Incorrect set of header files included in project files.
- Did not update log2(x) -> __log2(x).
- Some intrinsic names have changed such as:
  - __sleep() -> _sleep(x)
  - __ME() ->_ME()
  - __island()->_island()

Appendix A.

A.1 Examples of *'cmd_'* and *'ptrXX'* suffixes

This appendix decribes how to replace shorthand annotation with *'cmd_'* and *'_ptrXX'* pre/suffixes. Note XX should be replaced with '32' or '40' depending upon how their application was built. If a user is ever unsure which suffix to use ('_ptr32' or '_ptr40'), then using '_ptr40' is the safer choice.

Shorthand define	Added prefix/suffix
cls_add	cmd_cls_add_ptrXX
cls_add64	cmd_cls_add64_ptrXX
cls_add64_imm	cmd_cls_add64_imm_ptrXX
cls_add64_ind	cmd_cls_add64_ind_ptrXX
cls_add_imm	cmd_cls_add_imm_ptrXX
cls_add_imm_sat	cmd_cls_add_imm_sat_ptrXX
cls_add_ind	cmd_cls_add_ind_ptrXX
cls_add_sat	cmd_cls_add_sat_ptrXX
cls_add_sat_ind	cmd_cls_add_sat_ind_ptrXX
cls_cam_lookup	cmd_cls_cam_lookup_ptrXX
cls_cam_lookup16	cmd_cls_cam_lookup16_ptrXX
cls_cam_lookup16_add	cmd_cls_cam_lookup16_add_ptrXX
cls_cam_lookup16_add_ind	cmd_cls_cam_lookup16_add_ind_ptrXX
cls_cam_lookup16_ind	cmd_cls_cam_lookup16_ind_ptrXX
cls_cam_lookup24	cmd_cls_cam_lookup24_ptrXX
cls_cam_lookup24_add	cmd_cls_cam_lookup24_add_ptrXX
cls_cam_lookup24_add_extend	cmd_cls_cam_lookup24_add_extend_ptrXX
cls_cam_lookup24_add_inc	cmd_cls_cam_lookup24_add_inc_ptrXX
cls_cam_lookup24_add_inc_ind	cmd_cls_cam_lookup24_add_inc_ind_ptrXX
cls_cam_lookup24_add_ind	cmd_cls_cam_lookup24_add_ind_ptrXX
cls_cam_lookup24_add_lock	cmd_cls_cam_lookup24_add_lock_ptrXX
cls_cam_lookup24_ind	cmd_cls_cam_lookup24_ind_ptrXX
cls_cam_lookup8	cmd_cls_cam_lookup8_ptrXX
cls_cam_lookup8_add	cmd_cls_cam_lookup8_add_ptrXX
cls_cam_lookup8_add_ind	cmd_cls_cam_lookup8_add_ind_ptrXX
cls_cam_lookup8_ind	cmd_cls_cam_lookup8_ind_ptrXX
cls_cam_lookup_add	cmd_cls_cam_lookup_add_ptrXX

Shorthand define	Added prefix/suffix
cls_cam_lookup_add_ind	cmd_cls_cam_lookup_add_ind_ptrXX
cls_cam_lookup_ind	cmd_cls_cam_lookup_ind_ptrXX
cls_clear_bits	cmd_cls_clear_bits_ptrXX
cls_clear_bits_imm	cmd_cls_clear_bits_imm_ptrXX
cls_clear_bits_ind	cmd_cls_clear_bits_ind_ptrXX
cls_cmp_read_write	cmd_cls_cmp_read_write_ptrXX
cls_cmp_read_write_ind	cmd_cls_cmp_read_write_ind_ptrXX
cls_decr	cmd_cls_decr_ptrXX
cls_decr64	cmd_cls_decr64_ptrXX
cls_hash_mask	cmd_cls_hash_mask_ptrXX
cls_hash_mask_clear	cmd_cls_hash_mask_clear_ptrXX
cls_hash_mask_ind	cmd_cls_hash_mask_ind_ptrXX
cls_incr	cmd_cls_incr_ptrXX
cls_incr64	cmd_cls_incr64_ptrXX
cls_meter	cmd_cls_meter_ptrXX
cls_queue_lock	cmd_cls_queue_lock_ptrXX
cls_queue_lock_ind	cmd_cls_queue_lock_ind_ptrXX
cls_queue_unlock	cmd_cls_queue_unlock_ptrXX
cls_read	cmd_cls_read_ptrXX
cls_read_be	cmd_cls_read_be_ptrXX
cls_read_be_ind	cmd_cls_read_be_ind_ptrXX
cls_read_ind	cmd_cls_read_ind_ptrXX
cls_read_le	cmd_cls_read_le_ptrXX
cls_read_le_ind	cmd_cls_read_le_ind_ptrXX
cls_ring_get	cmd_cls_ring_get_ptrXX
cls_ring_get_ind	cmd_cls_ring_get_ind_ptrXX
cls_ring_get_safe	cmd_cls_ring_get_safe_ptrXX
cls_ring_get_safe_ind	cmd_cls_ring_get_safe_ind_ptrXX
cls_ring_init	cmd_cls_ring_init_ptrXX
cls_ring_journal	cmd_cls_ring_journal_ptrXX
cls_ring_journal_ind	cmd_cls_ring_journal_ind_ptrXX
cls_ring_pop	cmd_cls_ring_pop_ptrXX
cls_ring_pop_ind	cmd_cls_ring_pop_ind_ptrXX
cls_ring_pop_safe	cmd_cls_ring_pop_safe_ptrXX
cls_ring_pop_safe_ind	cmd_cls_ring_pop_safe_ind_ptrXX

Shorthand define	Added prefix/suffix
cls_ring_put	cmd_cls_ring_put_ptrXX
cls_ring_put_ind	cmd_cls_ring_put_ind_ptrXX
cls_set_bits	cmd_cls_set_bits_ptrXX
cls_set_bits_imm	cmd_cls_set_bits_imm_ptrXX
cls_set_bits_ind	cmd_cls_set_bits_ind_ptrXX
cls_statistic	cmd_cls_statistic_ptrXX
cls_statistic_imm	cmd_cls_statistic_imm_ptrXX
cls_sub	cmd_cls_sub_ptrXX
cls_sub64	cmd_cls_sub64_ptrXX
cls_sub64_imm	cmd_cls_sub64_imm_ptrXX
cls_sub64_ind	cmd_cls_sub64_ind_ptrXX
cls_sub_imm	cmd_cls_sub_imm_ptrXX
cls_sub_imm_sat	cmd_cls_sub_imm_sat_ptrXX
cls_sub_ind	cmd_cls_sub_ind_ptrXX
cls_sub_sat	cmd_cls_sub_sat_ptrXX
cls_sub_sat_ind	cmd_cls_sub_sat_ind_ptrXX
cls_swap	cmd_cls_swap_ptrXX
cls_tcam_lookup	cmd_cls_tcam_lookup_ptrXX
cls_tcam_lookup16	cmd_cls_tcam_lookup16_ptrXX
cls_tcam_lookup16_ind	cmd_cls_tcam_lookup16_ind_ptrXX
cls_tcam_lookup24	cmd_cls_tcam_lookup24_ptrXX
cls_tcam_lookup24_ind	cmd_cls_tcam_lookup24_ind_ptrXX
cls_tcam_lookup8	cmd_cls_tcam_lookup8_ptrXX
cls_tcam_lookup8_ind	cmd_cls_tcam_lookup8_ind_ptrXX
cls_tcam_lookup_ind	cmd_cls_tcam_lookup_ind_ptrXX
cls_test_and_add	cmd_cls_test_and_add_ptrXX
cls_test_and_add64	cmd_cls_test_and_add64_ptrXX
cls_test_and_add64_imm	cmd_cls_test_and_add64_imm_ptrXX
cls_test_and_add_imm	cmd_cls_test_and_add_imm_ptrXX
cls_test_and_add_imm_sat	cmd_cls_test_and_add_imm_sat_ptrXX
cls_test_and_add_sat	cmd_cls_test_and_add_sat_ptrXX
cls_test_and_add_sat_ind	cmd_cls_test_and_add_sat_ind_ptrXX
cls_test_and_clear_bits	cmd_cls_test_and_clear_bits_ptrXX
cls_test_and_clear_bits_imm	cmd_cls_test_and_clear_bits_imm_ptrXX
cls_test_and_clear_bits_ind	cmd_cls_test_and_clear_bits_ind_ptrXX

Shorthand define	Added prefix/suffix
cls_test_and_set_bits	cmd_cls_test_and_set_bits_ptrXX
cls_test_and_set_bits_imm	cmd_cls_test_and_set_bits_imm_ptrXX
cls_test_and_set_bits_ind	cmd_cls_test_and_set_bits_ind_ptrXX
cls_test_and_sub	cmd_cls_test_and_sub_ptrXX
cls_test_and_sub64	cmd_cls_test_and_sub64_ptrXX
cls_test_and_sub64_imm	cmd_cls_test_and_sub64_imm_ptrXX
cls_test_and_sub_imm	cmd_cls_test_and_sub_imm_ptrXX
cls_test_and_sub_imm_sat	cmd_cls_test_and_sub_imm_sat_ptrXX
cls_test_and_sub_sat	cmd_cls_test_and_sub_sat_ptrXX
cls_test_and_sub_sat_ind	cmd_cls_test_and_sub_sat_ind_ptrXX
cls_write	cmd_cls_write_ptrXX
cls_write8	cmd_cls_write8_ptrXX
cls_write8_be	cmd_cls_write8_be_ptrXX
cls_write8_be_ind	cmd_cls_write8_be_ind_ptrXX
cls_write8_ind	cmd_cls_write8_ind_ptrXX
cls_write8_le	cmd_cls_write8_le_ptrXX
cls_write8_le_ind	cmd_cls_write8_le_ind_ptrXX
cls_write_be	cmd_cls_write_be_ptrXX
cls_write_be_ind	cmd_cls_write_be_ind_ptrXX
cls_write_ind	cmd_cls_write_ind_ptrXX
cls_write_le	cmd_cls_write_le_ptrXX
cls_write_le_ind	cmd_cls_write_le_ind_ptrXX
cls_xor_bits	cmd_cls_xor_bits_ptrXX
cls_xor_bits_ind	cmd_cls_xor_bits_ind_ptrXX
mem_add32	cmd_mem_add32_ptrXX
mem_add32_imm	cmd_mem_add32_imm_ptrXX
mem_add32_imm_sat	cmd_mem_add32_imm_sat_ptrXX
mem_add32_ind	cmd_mem_add32_ind_ptrXX
mem_add32_sat	cmd_mem_add32_sat_ptrXX
mem_add32_sat_ind	cmd_mem_add32_sat_ind_ptrXX
mem_add64	cmd_mem_add64_ptrXX
mem_add64_imm	cmd_mem_add64_imm_ptrXX
mem_add64_imm_sat	cmd_mem_add64_imm_sat_ptrXX
mem_add64_ind	cmd_mem_add64_ind_ptrXX
mem_add64_sat	cmd_mem_add64_sat_ptrXX

Shorthand define	Added prefix/suffix
mem_add64_sat_ind	cmd_mem_add64_sat_ind_ptrXX
mem_cam128_init	cmd_mem_cam128_init_ptrXX
mem_cam128_lookup16	cmd_mem_cam128_lookup16_ptrXX
mem_cam128_lookup16_add	cmd_mem_cam128_lookup16_add_ptrXX
mem_cam128_lookup24	cmd_mem_cam128_lookup24_ptrXX
mem_cam128_lookup24_add	cmd_mem_cam128_lookup24_add_ptrXX
mem_cam128_lookup24_add_inc	cmd_mem_cam128_lookup24_add_inc_ptrXX
mem_cam128_lookup32	cmd_mem_cam128_lookup32_ptrXX
mem_cam128_lookup32_add	cmd_mem_cam128_lookup32_add_ptrXX
mem_cam128_lookup8	cmd_mem_cam128_lookup8_ptrXX
mem_cam128_lookup8_add	cmd_mem_cam128_lookup8_add_ptrXX
mem_cam128_set16	cmd_mem_cam128_set16_ptrXX
mem_cam128_set24	cmd_mem_cam128_set24_ptrXX
mem_cam128_set3	cmd_mem_cam128_set32_ptrXX
mem_cam128_set8	cmd_mem_cam128_set8_ptrXX
mem_cam256_init	cmd_mem_cam256_init_ptrXX
mem_cam256_lookup16	cmd_mem_cam256_lookup16_ptrXX
mem_cam256_lookup16_add	cmd_mem_cam256_lookup16_add_ptrXX
mem_cam256_lookup24	cmd_mem_cam256_lookup24_ptrXX
mem_cam256_lookup24_add	cmd_mem_cam256_lookup24_add_ptrXX
mem_cam256_lookup24_add_inc	cmd_mem_cam256_lookup24_add_inc_ptrXX
mem_cam256_lookup32	cmd_mem_cam256_lookup32_ptrXX
mem_cam256_lookup32_add	cmd_mem_cam256_lookup32_add_ptrXX
mem_cam256_lookup8	cmd_mem_cam256_lookup8_ptrXX
mem_cam256_lookup8_add	cmd_mem_cam256_lookup8_add_ptrXX
mem_cam256_set16	cmd_mem_cam256_set16_ptrXX
mem_cam256_set24	cmd_mem_cam256_set24_ptrXX
mem_cam256_set32	cmd_mem_cam256_set32_ptrXX
mem_cam256_set8	cmd_mem_cam256_set8_ptrXX
mem_cam384_init	cmd_mem_cam384_init_ptrXX
mem_cam384_lookup16	cmd_mem_cam384_lookup16_ptrXX
mem_cam384_lookup16_add	cmd_mem_cam384_lookup16_add_ptrXX
mem_cam384_lookup24	cmd_mem_cam384_lookup24_ptrXX
mem_cam384_lookup24_add	cmd_mem_cam384_lookup24_add_ptrXX
mem_cam384_lookup24_add_inc	cmd_mem_cam384_lookup24_add_inc_ptrXX

Shorthand define	Added prefix/suffix
mem_cam384_lookup32	cmd_mem_cam384_lookup32_ptrXX
mem_cam384_lookup32_add	cmd_mem_cam384_lookup32_add_ptrXX
mem_cam384_lookup8	cmd_mem_cam384_lookup8_ptrXX
mem_cam384_lookup8_add	cmd_mem_cam384_lookup8_add_ptrXX
mem_cam384_set16	cmd_mem_cam384_set16_ptrXX
mem_cam384_set24	cmd_mem_cam384_set24_ptrXX
mem_cam384_set32	cmd_mem_cam384_set32_ptrXX
mem_cam384_set8	cmd_mem_cam384_set8_ptrXX
mem_cam512_init	cmd_mem_cam512_init_ptrXX
mem_cam512_lookup16	cmd_mem_cam512_lookup16_ptrXX
mem_cam512_lookup16_add	cmd_mem_cam512_lookup16_add_ptrXX
mem_cam512_lookup24	cmd_mem_cam512_lookup24_ptrXX
mem_cam512_lookup24_add	cmd_mem_cam512_lookup24_add_ptrXX
mem_cam512_lookup24_add_inc	cmd_mem_cam512_lookup24_add_inc_ptrXX
mem_cam512_lookup32	cmd_mem_cam512_lookup32_ptrXX
mem_cam512_lookup32_add	cmd_mem_cam512_lookup32_add_ptrXX
mem_cam512_lookup8	cmd_mem_cam512_lookup8_ptrXX
mem_cam512_lookup8_add	cmd_mem_cam512_lookup8_add_ptrXX
mem_cam512_set16	cmd_mem_cam512_set16_ptrXX
mem_cam512_set24	cmd_mem_cam512_set24_ptrXX
mem_cam512_set32	cmd_mem_cam512_set32_ptrXX
mem_cam512_set8	cmd_mem_cam512_set8_ptrXX
mem_clr	cmd_mem_clr_ptrXX
mem_clr_imm	cmd_mem_clr_imm_ptrXX
mem_clr_ind	cmd_mem_clr_ind_ptrXX
mem_compare_write	cmd_mem_compare_write_ptrXX
mem_compare_write_ind	cmd_mem_compare_write_ind_ptrXX
mem_decr32	cmd_mem_decr32_ptrXX
mem_decr64	cmd_mem_decr64_ptrXX
mem_incr32	cmd_mem_incr32_ptrXX
mem_incr64	cmd_mem_incr64_ptrXX
mem_lock128	cmd_mem_lock128_ptrXX
mem_lock128_init	cmd_mem_lock128_init_ptrXX
mem_lock256	cmd_mem_lock256_ptrXX
mem_lock256_init	cmd_mem_lock256_init_ptrXX

Shorthand define	Added prefix/suffix
mem_lock384	cmd_mem_lock384_ptrXX
mem_lock384_init	cmd_mem_lock384_init_ptrXX
mem_lock512	cmd_mem_lock512_ptrXX
mem_lock512_init	cmd_mem_lock512_init_ptrXX
mem_lockq128_init	cmd_mem_lockq128_init_ptrXX
mem_lockq128_lock	cmd_mem_lockq128_lock_ptrXX
mem_lockq128_unlock	cmd_mem_lockq128_unlock_ptrXX
mem_lockq256_init	cmd_mem_lockq256_init_ptrXX
mem_lockq256_lock	cmd_mem_lockq256_lock_ptrXX
mem_lockq256_unlock	cmd_mem_lockq256_unlock_ptrXX
mem_mask_compare_write	cmd_mem_mask_compare_write_ptrXX
mem_microq128_get	cmd_mem_microq128_get_ptrXX
mem_microq128_init	cmd_mem_microq128_init_ptrXX
mem_microq128_pop	cmd_mem_microq128_pop_ptrXX
mem_microq128_put	cmd_mem_microq128_put_ptrXX
mem_microq256_get	cmd_mem_microq256_get_ptrXX
mem_microq256_init	cmd_mem_microq256_init_ptrXX
mem_microq256_pop	cmd_mem_microq256_pop_ptrXX
mem_microq256_put	cmd_mem_microq256_put_ptrXX
mem_read32	cmd_mem_read32_ptrXX
mem_read32_be	cmd_mem_read32_be_ptrXX
mem_read32_be_ind	cmd_mem_read32_be_ind_ptrXX
mem_read32_ind	cmd_mem_read32_ind_ptrXX
mem_read32_le	cmd_mem_read32_le_ptrXX
mem_read32_swap	cmd_mem_read32_swap_ptrXX
mem_read32_swap_be	cmd_mem_read32_swap_be_ptrXX
mem_read32_swap_be_ind	cmd_mem_read32_swap_be_ind_ptrXX
mem_read32_swap_ind	cmd_mem_read32_swap_ind_ptrXX
mem_read32_swap_le	cmd_mem_read32_swap_le_ptrXX
mem_read64	cmd_mem_read64_ptrXX
mem_read64_be	cmd_mem_read64_be_ptrXX
mem_read64_be_ind	cmd_mem_read64_be_ind_ptrXX
mem_read64_ind	cmd_mem_read64_ind_ptrXX
mem_read64_le	cmd_mem_read64_le_ptrXX
mem_read64_le_ind	cmd_mem_read64_le_ind_ptrXX

Shorthand define	Added prefix/suffix
mem_read64_swap	cmd_mem_read64_swap_ptrXX
mem_read64_swap_be	cmd_mem_read64_swap_be_ptrXX
mem_read64_swap_be_ind	cmd_mem_read64_swap_be_ind_ptrXX
mem_read64_swap_ind	cmd_mem_read64_swap_ind_ptrXX
mem_read64_swap_le	cmd_mem_read64_swap_le_ptrXX
mem_read64_swap_le_ind	cmd_mem_read64_swap_le_ind_ptrXX
mem_read8	cmd_mem_read8_ptrXX
mem_read8_be	cmd_mem_read8_be_ptrXX
mem_read8_be_ind	cmd_mem_read8_be_ind_ptrXX
mem_read8_ind	cmd_mem_read8_ind_ptrXX
mem_read_atomic	cmd_mem_read_atomic_ptrXX
mem_read_atomic_ind	cmd_mem_read_atomic_ind_ptrXX
mem_set	cmd_mem_set_ptrXX
mem_set_imm	cmd_mem_set_imm_ptrXX
mem_set_ind	cmd_mem_set_ind_ptrXX
mem_sub32	cmd_mem_sub32_ptrXX
mem_sub32_imm	cmd_mem_sub32_imm_ptrXX
mem_sub32_imm_sat	cmd_mem_sub32_imm_sat_ptrXX
mem_sub32_ind	cmd_mem_sub32_ind_ptrXX
mem_sub32_sat	cmd_mem_sub32_sat_ptrXX
mem_sub32_sat_ind	cmd_mem_sub32_sat_ind_ptrXX
mem_sub64	cmd_mem_sub64_ptrXX
mem_sub64_imm	cmd_mem_sub64_imm_ptrXX
mem_sub64_imm_sat	cmd_mem_sub64_imm_sat_ptrXX
mem_sub64_ind	cmd_mem_sub64_ind_ptrXX
mem_sub64_sat	cmd_mem_sub64_sat_ptrXX
mem_sub64_sat_ind	cmd_mem_sub64_sat_ind_ptrXX
mem_swap	cmd_mem_swap_ptrXX
mem_swap_imm	cmd_mem_swap_imm_ptrXX
mem_swap_ind	cmd_mem_swap_ind_ptrXX
mem_tcam128_init	cmd_mem_tcam128_init_ptrXX
mem_tcam128_lookup16	cmd_mem_tcam128_lookup16_ptrXX
mem_tcam128_lookup24	cmd_mem_tcam128_lookup24_ptrXX
mem_tcam128_lookup32	cmd_mem_tcam128_lookup32_ptrXX
mem_tcam128_lookup8	cmd_mem_tcam128_lookup8_ptrXX

Shorthand define	Added prefix/suffix
mem_tcam128_set16_entry	cmd_mem_tcam128_set16_entry_ptrXX
mem_tcam128_set16_entry_mask	cmd_mem_tcam128_set16_entry_mask_ptrXX
mem_tcam128_set16_entry_value	cmd_mem_tcam128_set16_entry_value_ptrXX
mem_tcam128_set16_word	cmd_mem_tcam128_set16_word_ptrXX
mem_tcam128_set24_entry	cmd_mem_tcam128_set24_entry_ptrXX
mem_tcam128_set24_entry_mask	cmd_mem_tcam128_set24_entry_mask_ptrXX
mem_tcam128_set24_entry_value	cmd_mem_tcam128_set24_entry_value_ptrXX
mem_tcam128_set24_word	cmd_mem_tcam128_set24_word_ptrXX
mem_tcam128_set32_entry	cmd_mem_tcam128_set32_entry_ptrXX
mem_tcam128_set32_entry_mask	cmd_mem_tcam128_set32_entry_mask_ptrXX
mem_tcam128_set32_entry_value	cmd_mem_tcam128_set32_entry_value_ptrXX
mem_tcam128_set32_word	cmd_mem_tcam128_set32_word_ptrXX
mem_tcam128_set8_entry	cmd_mem_tcam128_set8_entry_ptrXX
mem_tcam128_set8_entry_mask	cmd_mem_tcam128_set8_entry_mask_ptrXX
mem_tcam128_set8_entry_value	cmd_mem_tcam128_set8_entry_value_ptrXX
mem_tcam128_set8_word	cmd_mem_tcam128_set8_word_ptrXX
mem_tcam256_init	cmd_mem_tcam256_init_ptrXX
mem_tcam256_lookup16	cmd_mem_tcam256_lookup16_ptrXX
mem_tcam256_lookup24	cmd_mem_tcam256_lookup24_ptrXX
mem_tcam256_lookup32	cmd_mem_tcam256_lookup32_ptrXX
mem_tcam256_lookup8	cmd_mem_tcam256_lookup8_ptrXX
mem_tcam256_set16_entry	cmd_mem_tcam256_set16_entry_ptrXX
mem_tcam256_set16_entry_mask	cmd_mem_tcam256_set16_entry_mask_ptrXX
mem_tcam256_set16_entry_value	cmd_mem_tcam256_set16_entry_value_ptrXX
mem_tcam256_set16_word	cmd_mem_tcam256_set16_word_ptrXX
mem_tcam256_set24_entry	cmd_mem_tcam256_set24_entry_ptrXX
mem_tcam256_set24_entry_mask	cmd_mem_tcam256_set24_entry_mask_ptrXX
mem_tcam256_set24_entry_value	cmd_mem_tcam256_set24_entry_value_ptrXX
mem_tcam256_set24_word	cmd_mem_tcam256_set24_word_ptrXX
mem_tcam256_set32_entry	cmd_mem_tcam256_set32_entry_ptrXX
mem_tcam256_set32_entry_mask	cmd_mem_tcam256_set32_entry_mask_ptrXX
mem_tcam256_set32_entry_value	cmd_mem_tcam256_set32_entry_value_ptrXX
mem_tcam256_set32_word	cmd_mem_tcam256_set32_word_ptrXX
mem_tcam256_set8_entry	cmd_mem_tcam256_set8_entry_ptrXX
mem_tcam256_set8_entry_mask	cmd_mem_tcam256_set8_entry_mask_ptrXX

Shorthand define	Added prefix/suffix
mem_tcam256_set8_entry_value	cmd_mem_tcam256_set8_entry_value_ptrXX
mem_tcam256_set8_word	cmd_mem_tcam256_set8_word_ptrXX
mem_tcam384_init	cmd_mem_tcam384_init_ptrXX
mem_tcam384_lookup16	cmd_mem_tcam384_lookup16_ptrXX
mem_tcam384_lookup24	cmd_mem_tcam384_lookup24_ptrXX
mem_tcam384_lookup32	cmd_mem_tcam384_lookup32_ptrXX
mem_tcam384_lookup8	cmd_mem_tcam384_lookup8_ptrXX
mem_tcam384_set16_entry	cmd_mem_tcam384_set16_entry_ptrXX
mem_tcam384_set16_entry_mask	cmd_mem_tcam384_set16_entry_mask_ptrXX
mem_tcam384_set16_entry_value	cmd_mem_tcam384_set16_entry_value_ptrXX
mem_tcam384_set16_word	cmd_mem_tcam384_set16_word_ptrXX
mem_tcam384_set24_entry	cmd_mem_tcam384_set24_entry_ptrXX
mem_tcam384_set24_entry_mask	cmd_mem_tcam384_set24_entry_mask_ptrXX
mem_tcam384_set24_entry_value	cmd_mem_tcam384_set24_entry_value_ptrXX
mem_tcam384_set24_word	cmd_mem_tcam384_set24_word_ptrXX
mem_tcam384_set32_entry	cmd_mem_tcam384_set32_entry_ptrXX
mem_tcam384_set32_entry_mask	cmd_mem_tcam384_set32_entry_mask_ptrXX
mem_tcam384_set32_entry_value	cmd_mem_tcam384_set32_entry_value_ptrXX
mem_tcam384_set32_word	cmd_mem_tcam384_set32_word_ptrXX
mem_tcam384_set8_entry	cmd_mem_tcam384_set8_entry_ptrXX
mem_tcam384_set8_entry_mask	cmd_mem_tcam384_set8_entry_mask_ptrXX
mem_tcam384_set8_entry_value	cmd_mem_tcam384_set8_entry_value_ptrXX
mem_tcam384_set8_word	cmd_mem_tcam384_set8_word_ptrXX
mem_tcam512_init	cmd_mem_tcam512_init_ptrXX
mem_tcam512_lookup16	cmd_mem_tcam512_lookup16_ptrXX
mem_tcam512_lookup24	cmd_mem_tcam512_lookup24_ptrXX
mem_tcam512_lookup32	cmd_mem_tcam512_lookup32_ptrXX
mem_tcam512_lookup8	cmd_mem_tcam512_lookup8_ptrXX
mem_tcam512_set16_entry	cmd_mem_tcam512_set16_entry_ptrXX
mem_tcam512_set16_entry_mask	cmd_mem_tcam512_set16_entry_mask_ptrXX
mem_tcam512_set16_entry_value	cmd_mem_tcam512_set16_entry_value_ptrXX
mem_tcam512_set16_word	cmd_mem_tcam512_set16_word_ptrXX
mem_tcam512_set24_entry	cmd_mem_tcam512_set24_entry_ptrXX
mem_tcam512_set24_entry_mask	cmd_mem_tcam512_set24_entry_mask_ptrXX
mem_tcam512_set24_entry_value	cmd_mem_tcam512_set24_entry_value_ptrXX

Shorthand define	Added prefix/suffix
mem_tcam512_set24_word	cmd_mem_tcam512_set24_word_ptrXX
mem_tcam512_set32_entry	cmd_mem_tcam512_set32_entry_ptrXX
mem_tcam512_set32_entry_mask	cmd_mem_tcam512_set32_entry_mask_ptrXX
mem_tcam512_set32_entry_value	cmd_mem_tcam512_set32_entry_value_ptrXX
mem_tcam512_set32_word	cmd_mem_tcam512_set32_word_ptrXX
mem_tcam512_set8_entry	cmd_mem_tcam512_set8_entry_ptrXX
mem_tcam512_set8_entry_mask	cmd_mem_tcam512_set8_entry_mask_ptrXX
mem_tcam512_set8_entry_value	cmd_mem_tcam512_set8_entry_value_ptrXX
mem_tcam512_set8_word	cmd_mem_tcam512_set8_word_ptrXX
mem_test_and_add32	cmd_mem_test_and_add32_ptrXX
mem_test_and_add32_imm	cmd_mem_test_and_add32_imm_ptrXX
mem_test_and_add32_imm_sat	cmd_mem_test_and_add32_imm_sat_ptrXX
mem_test_and_add32_ind	cmd_mem_test_and_add32_ind_ptrXX
mem_test_and_add32_sat	cmd_mem_test_and_add32_sat_ptrXX
mem_test_and_add32_sat_ind	cmd_mem_test_and_add32_sat_ind_ptrXX
mem_test_and_add64	cmd_mem_test_and_add64_ptrXX
mem_test_and_add64_imm	cmd_mem_test_and_add64_imm_ptrXX
mem_test_and_add64_imm_sat	cmd_mem_test_and_add64_imm_sat_ptrXX
mem_test_and_add64_ind	cmd_mem_test_and_add64_ind_ptrXX
mem_test_and_add64_sat	cmd_mem_test_and_add64_sat_ptrXX
mem_test_and_add64_sat_ind	cmd_mem_test_and_add64_sat_ind_ptrXX
mem_test_and_clr	cmd_mem_test_and_clr_ptrXX
mem_test_and_clr_imm	cmd_mem_test_and_clr_imm_ptrXX
mem_test_and_clr_ind	cmd_mem_test_and_clr_ind_ptrXX
mem_test_and_compare_write	cmd_mem_test_and_compare_write_ptrXX
mem_test_and_compare_write_ind	cmd_mem_test_and_compare_write_ind_ptrXX
mem_test_and_set	cmd_mem_test_and_set_ptrXX
mem_test_and_set_imm	cmd_mem_test_and_set_imm_ptrXX
mem_test_and_set_ind	cmd_mem_test_and_set_ind_ptrXX
mem_test_and_sub32	cmd_mem_test_and_sub32_ptrXX
mem_test_and_sub32_imm	cmd_mem_test_and_sub32_imm_ptrXX
mem_test_and_sub32_imm_sat	cmd_mem_test_and_sub32_imm_sat_ptrXX
mem_test_and_sub32_ind	cmd_mem_test_and_sub32_ind_ptrXX
mem_test_and_sub32_sat	cmd_mem_test_and_sub32_sat_ptrXX
mem_test_and_sub32_sat_ind	cmd_mem_test_and_sub32_sat_ind_ptrXX

Shorthand define	Added prefix/suffix
mem_test_and_sub64	cmd_mem_test_and_sub64_ptrXX
mem_test_and_sub64_imm	cmd_mem_test_and_sub64_imm_ptrXX
mem_test_and_sub64_imm_sat	cmd_mem_test_and_sub64_imm_sat_ptrXX
mem_test_and_sub64_ind	cmd_mem_test_and_sub64_ind_ptrXX
mem_test_and_sub64_sat	cmd_mem_test_and_sub64_sat_ptrXX
mem_test_and_sub64_sat_ind	cmd_mem_test_and_sub64_sat_ind_ptrXX
mem_test_and_xor	cmd_mem_test_and_xor_ptrXX
mem_test_and_xor_imm	cmd_mem_test_and_xor_imm_ptrXX
mem_test_and_xor_ind	cmd_mem_test_and_xor_ind_ptrXX
mem_test_mask_and_compare_write	cmd_mem_test_mask_and_compare_write_ptrXX
mem_ticket_line_init	cmd_mem_ticket_line_init_ptrXX
mem_ticket_line_push_init	cmd_mem_ticket_line_push_init_ptrXX
mem_ticket_release	cmd_mem_ticket_release_ptrXX
mem_ticket_release_push	cmd_mem_ticket_release_push_ptrXX
mem_unlock128	cmd_mem_unlock128_ptrXX
mem_unlock256	cmd_mem_unlock256_ptrXX
mem_unlock384	cmd_mem_unlock384_ptrXX
mem_unlock512	cmd_mem_unlock512_ptrXX
mem_write32	cmd_mem_write32_ptrXX
mem_write32_be	cmd_mem_write32_be_ptrXX
mem_write32_be_ind	cmd_mem_write32_be_ind_ptrXX
mem_write32_ind	cmd_mem_write32_ind_ptrXX
mem_write32_le	cmd_mem_write32_le_ptrXX
mem_write32_swap	cmd_mem_write32_swap_ptrXX
mem_write32_swap_be	cmd_mem_write32_swap_be_ptrXX
mem_write32_swap_be_ind	cmd_mem_write32_swap_be_ind_ptrXX
mem_write32_swap_ind	cmd_mem_write32_swap_ind_ptrXX
mem_write32_swap_le	cmd_mem_write32_swap_le_ptrXX
mem_write64	cmd_mem_write64_ptrXX
mem_write64_be	cmd_mem_write64_be_ptrXX
mem_write64_be_ind	cmd_mem_write64_be_ind_ptrXX
mem_write64_ind	cmd_mem_write64_ind_ptrXX
mem_write64_le	cmd_mem_write64_le_ptrXX
mem_write64_le_ind	cmd_mem_write64_le_ind_ptrXX
mem_write64_swap	cmd_mem_write64_swap_ptrXX

Shorthand define	Added prefix/suffix
mem_write64_swap_be	cmd_mem_write64_swap_be_ptrXX
mem_write64_swap_be_ind	cmd_mem_write64_swap_be_ind_ptrXX
mem_write64_swap_ind	cmd_mem_write64_swap_ind_ptrXX
mem_write64_swap_le	cmd_mem_write64_swap_le_ptrXX
mem_write64_swap_le_ind	cmd_mem_write64_swap_le_ind_ptrXX
mem_write8	cmd_mem_write8_ptrXX
mem_write8_be	cmd_mem_write8_be_ptrXX
mem_write8_be_ind	cmd_mem_write8_be_ind_ptrXX
mem_write8_ind	cmd_mem_write8_ind_ptrXX
mem_write8_le	cmd_mem_write8_le_ptrXX
mem_write8_le_ind	cmd_mem_write8_le_ind_ptrXX
mem_write8_swap	cmd_mem_write8_swap_ptrXX
mem_write8_swap_be	cmd_mem_write8_swap_be_ptrXX
mem_write8_swap_be_ind	cmd_mem_write8_swap_be_ind_ptrXX
mem_write8_swap_ind	cmd_mem_write8_swap_ind_ptrXX
mem_write8_swap_le	cmd_mem_write8_swap_le_ptrXX
mem_write8_swap_le_ind	cmd_mem_write8_swap_le_ind_ptrXX
mem_write_atomic	cmd_mem_write_atomic_ptrXX
mem_write_atomic_imm	cmd_mem_write_atomic_imm_ptrXX
mem_write_atomic_ind	cmd_mem_write_atomic_ind_ptrXX
mem_xor	cmd_mem_xor_ptrXX
mem_xor_imm	cmd_mem_xor_imm_ptrXX
mem_xor_ind	cmd_mem_xor_ind_ptrXX
pcie_read	cmd_pcie_read_ptrXX
pcie_read_ind	cmd_pcie_read_ind_ptrXX
pcie_read_pci	cmd_pcie_read_pci_ptrXX
pcie_read_pci_ind	cmd_pcie_read_pci_ind_ptrXX
pcie_read_rid	cmd_pcie_read_rid_ptrXX
pcie_read_rid_ind	cmd_pcie_read_rid_ind_ptrXX
pcie_write	cmd_pcie_write_ptrXX
pcie_write_ind	cmd_pcie_write_ind_ptrXX
pcie_write_pci	cmd_pcie_write_pci_ptrXX
pcie_write_pci_ind	cmd_pcie_write_pci_ind_ptrXX
pcie_write_rid	cmd_pcie_write_rid_ptrXX
pcie_write_rid_ind	cmd_pcie_write_rid_ind_ptrXX

3. Technical Support

To obtain additional information, or to provide feedback, please email `<support@netronome.com>` or contact the nearest **Netronome** technical support representative.