Freescale Semiconductor
Release Notes

Document Number: CW_SC_3900FP_v10.9.0
Rev. 1.0, 11/2015

CodeWarrior Development Studio for StarCore 3900FP DSP v10.9.0 Release Notes

Release Description

Release 10.9.0 of CodeWarrior Development Studio for StarCore sc3900FP DSP provides MAPLE firmware A30, SDOS 05.16.00 support, and new features in software analysis, debugger and build tools.

Contents

Release Content

1.1 MAPLE Firmware

This is the most up to date reference manual for external link for customers:

http://www.freescale.com/go/230826425

Files available:

MAPLE-B3RM_RevE_B4860Rev2_A30.pdf

MAPLE-B3RM_RevE_B4420Rev2_A30.pdf

1.1.1 MAPLE3W A30 Firmware Release

The init parameter called timer_period, should not be of small values, thus overloading the MAPLE RISCS with the parse Microcode.

This release includes the following PEs:

- eFTPE2 (x2) – F1 verified.
- CRCPE – F1 verified.
- CGPE – F1 verified.
- CRPE ULB2 is F1 verified
- CRPE DL2 is F1 verified.
- TCPE – F1 verified.
- PUFFT2_EDF – F1 Verified (See exceptions below).

Code Size: 46KB

PUFFT2_EDF

- Alternative BW support (ALT_UBW)
- PUFFT external interrupt triggers trigger a newly interrupt controller, known as MPISR. The MPISR programming model is described in the RM, including but not limited to the following aspects:
    - ISR acknowledge address value and MMU task ID;
    - Interrupt vector linking the incoming trigger to one or more PUFFT processes;
    - Per PUFFT process trigger counters;

1.1.2 MAPLE3LW A30 Firmware Release

The init parameter called timer_period, should not be of small values, thus overloading the MAPLE RISCS with the parse Microcode.

This release includes the following PEs:

- eFTPE2 (x3) – F2 verified.
- EQPE2 – F2 verified.
- DEPE – F1 verified

- CRCPE – F1 verified.
- eTVPE2 – F1 verified.
- PDSCH2 EDF – F1 verified (with exceptions)
- PUFFT2 EDF – F1 verified (with exceptions)
- PUSCH2 EDF – F1 verified for a limited feature list

Code Size: 152KB

EQPE2

- Non-Interpolation mode in standalone EQPE enabled
- EQPE2 features are according to PUSCH2-EDF feature list
- MMSE Repeat mode in standalone EQPE enabled

PUSCH2 EDF

- New in this release:
    - User parameter RI_BITS_0/1/2/3/4 – new value "3" marks the carrier has no RI
    - New user parameter CQI_EXT_BITS enables adding CQI bit amount on account of the carriers without RI
    - New user parameter NV_BETA_SRC enables setting NV BETA source per user
    - Bug fix: when CQI_MAX_OUT_EN=1 take pupe data from an offset calculated internally according to CQI amount formula and not from CQI_MAX_OUT offset
    - UL-COMP Post EQPE combining
    - UL-COMP LLR combining

PDSCH2 EDF

- New in this release:
    - Frequency correction on eFTPE input samples
    - Enabling position RS in MBSFN subframes
    - Programmable patterns for padding OFDM symbols as well as DTX
    - Disabling Positioning Reference Signals positioning on selected resource blocks
    - DM-RS support for 3GPP LTE Release 11
    - Multiple CS-RS support for coordinate transmission
    - Multiple CSI-RS support for coordinate transmission
- Not supported features:
    - 7.5KHz
    - In case of multi sector configuration, PDSCH may not meet throughput with MNOS=2 and mixed NCP/ECP
    - In case of multi sector configuration, PDSCH may not meet throughput with external padding with MNOS<4
    - Latency optimizations

PUFFT2 EDF

- Alternative BW support (ALT_UBW)
- PUFFT external interrupt triggers trigger a newly interrupt controller, known as MPISR. The MPISR programming model is described in the RM, including but not limited to the following aspects:
    - ISR acknowledge address value and MMU task ID;
    - Interrupt vector linking the incoming trigger to one or more PUFFT processes;
    - Per PUFFT process trigger counters;
- The input buffer base address (parameter MPUFFT<s>AIBA<x>, see section 9.5.3.1.9.1 in RM) should be 16 bytes aligned.

EFTPE2

- Unified output/input scale data structure in repeat mode (enable zero intervention for scale adjustment in case of multiple eFTPE jobs)

1.2 SmartDSP OS

SmartDSP OS R05.16.00 release includes support for firmware A30, L2 cache error interrupt, CPRI reconfiguration level 3, CPRI auxiliary demo extensions, support for placing StarCore images over 4GB, L1 defense reset from debug state.

1.2.1 Kernel Features

L2 Cache Error Interrupt

SmartDSP OS R05.16.00 introduces support for L2 cache error interrupt. Application can now enable this interrupt for several L2 cache error indications and register a callback to handle them.
This feature also supports setting thresholds for the interrupt indication.

Usage of the feature is demonstrated in advanced_kernel demo.

SmartDSP OS Heaps and Memory Section Names

SmartDSP OS R05.16.00 introduces a more user friendly naming convention for SmartDSP OS heaps and memory sections.
As SmartDSP OS moved to hardware coherency architecture the default heaps has changed since it is no longer preferable the default shared heap to be configured with non-cacheable attribute.
The difference between hardware and software coherency architectures made our heap naming confusing.
Hence, SmartDSP OS heaps names were modified to contain the information whether they are cacheable or not.
Examples:
Instead of:

```
OS_MEM_LOCAL = (OS_VALID_NUM | OS_SMARTDSP_HEAP | OS_MEM_CACHEABLE_TYPE),
OS_MEM_LOCAL_NONCACHEABLE = (OS_VALID_NUM | OS_SMARTDSP_HEAP),
```

The names are now:

```
OS_MEM_LOCAL_CACHEABLE = (OS_VALID_NUM | OS_SMARTDSP_HEAP | OS_MEM_CACHEABLE_TYPE),
OS_MEM_LOCAL_NONCACHEABLE = (OS_VALID_NUM | OS_SMARTDSP_HEAP),
```

In linker files,
Instead of:

```
descriptor_shared_data_m3 {
.shared_data_m3
.shared_data_m3_bss
} > shared_data_m3_cacheable_descriptor;
```

The names will be:

```
descriptor_shared_data_m3_cacheable {
.shared_data_m3_cacheable
.shared_data_m3_cacheable_bss
} > shared_data_m3_cacheable_descriptor;
```

In os_config.h,
Instead of:

```
#define OS_HEAP_SIZE 0x4000 /* Heap size */
```

The names will be:

```
#define OS_HEAP_CACHEABLE_SIZE 0x4000 /* Heap size */
```

All SmartDSP OS kernel modules, drivers and demos were modified to use the new naming conventions.
Note: this change implies that all applications must use the new naming convention in order to build properly.

New Boot Flow for Supporting High Addresses

SmartDSP OS R05.16.00 introduces a new boot flow to enable completely locating the StareCore images (code and data) over 4GB memory address (addresses higher than 32 bit).

The StarCore is a 32-bit core and can access 36-bit physical addresses using the MMU. However, when the DSP core gets out of reset with MMU disabled by default (1:1 mapping), so all startup functions and data which are accessed before MMU programming had to be placed within the 32bit address space.

In order to support using only addresses higher than 32-bit address space, the following changes were made:

- MMU descriptors are now programmed by the loader (PPC or CodeWarrior) to map the data and the code, before DSP begins executing code. Hence when DSP begins to run, its MMU is already enabled and translating addresses.

  Because the memory accesses can use cacheable attributes the loader will also have to initialize and configure the L2 and L1 caches.

- To support that, CodeWarrior 10.9.0 elf2xx application now adds this configuration as part of the bin image which it generates. This means that when the PPC loads the DSP binary image, this load now also configures the DSP MMU, to enable address translation (also CodeWarrior debugger was modified to enable the same when running DSP from CodeWarrior). To enable this, please see below the linker files changes which are needed.

- Configuration of DSP MMU was removed from SmartDSP OS startup code hooks. As DSP now begins to run with MMU completely configured (including caches and memory protection). Normal caches usage should be that the caches are enabled by the loader before DSP begins to run (configuration in linker file). However, SmartDSP OS still enables to configure caches for debug purposes.

- In L1 Defense reset, the DSP MMU is zeroed. This means that the PPC must reconfigure the DSP MMU before DSP begins to run again. This cannot be done while the DSP is in reset (MMU configurations are not enabled). In order to overcome that, the PPC now puts DSP in boot hold-off while DSP is in reset. When reset is de-asserted, DSP does not begin to run, but stays at boot hold off. At this point, the DSP image is being loaded by the PPC, and this includes the DSP MMU configuration. After loading is done, DSP boot hold-off is released and DSP begins to run.

- SmartDSP OS QDS support default linker file (os_b4860_linker_ads_defines.l3k) now enables high address for DSP DDR. For using lower address for DSP, another linker file was added (os_b4860_linker_ads_defines_4g.l3k).

- CodeWarrior remote system now supports 2 options for using high addresses or lower addresses. For more information, please refer to CodeWarrior documentation.

All SmartDSP OS demos were modified to include the new conventions, and to use only higher than 32-bit addresses. An exception to that is debug_hooks demo, which is using the new conventions at 32-bit addresses (old memory map).
See memory map chapter for description of updated memory map.

Note: all applications using CodeWarrior 10.9.0 must include the changes in linker files in order to build properly. This is true even if not using addresses higher than 32-bit by the DSP.

Application Linker File Required Changes
All application linker files were modified to support the new boot flow and memory map.

- DDR partitioning linker file b4xx0_mem_partition_link.l3k - Since there is no 1:1 mapping, the virtual addresses are specifically stated.
- Shared/local memory linker file os_b4xx0_link.l3k All descriptors are now defined in non-1:1 mapping address_translation. The address translation format has changed, e.g. os_shared_data_cacheable_descriptor (SHARED_DATA_MMU_DEF_LOW, SHARED_DATA_MMU_DEF_HIGH): SHARED_DDR0;
- SmartDSP OS system and background stack descriptors Id is fixed (2 for system, 3 for background), and hence must be reserved in order to guarantee that they are used for this purpose: os_system_stack_descriptor (STACK_DATA_MMU_DEF_LOW, STACK_DATA_MMU_DEF_HIGH) : LOCAL_DDR0, AFTER(local_data_ddr0_nocacheable_descriptor) descriptor(2); os_background_stack_descriptor (STACK_DATA_MMU_DEF_LOW, STACK_DATA_MMU_DEF_HIGH) : LOCAL_DDR0, AFTER(os_system_stack_descriptor) descriptor(3);

- For formal description of the linker changes convention, please refer to CodeWarrior documentation. See also SmartDSP OS demos for examples of new convention usage.

L1 Defense Reset from Debug State

SmartDSP OS R05.16.00 introduces support for reset when DSP is in debug state. This feature enables recovering from DSP MMU error. This is being done by the PPC modifying the DSP PC to a pre-defined DSP function, from which the DSP can safely continue to run, until it is reset by the PPC. The safe location recommended function is osL1dResetRequest(). The DSP virtual address of this function should be known to PPC.

This feature is demonstrated in: \demos\starcore\b4860\L1_Defense_integration_demo

Please note that Linux BSP (running on PPC) should be used with a version supporting this feature.

1.2.2 Driver Features

CPRI Reconfiguration Level 3

SmartDSP OS R05.16.00 introduces CPRI reconfiguration level 3 support. Reconfiguration level 3 allows modification of auxiliary mode parameter.
This enables load balancing tuning between CPRI units connected in a daisy chain (see B4860 Reference Manual).

Usage of the feature is demonstrated in cpri_auxiliary demo.

L1 Defense Support for CPRI Reconfiguration Scenarios

SmartDSP OS R05.16.00 introduces support for L1 Defense warm reset during CPRI reconfiguration in several scenarios. This feature allows SmartDSP OS to identify and recover from a warm reset during CPRI reconfiguration flow.

This feature was tested for the following cases:

- L1 Defense mode1 during reconfiguration level1
- L1 Defense mode2 during reconfiguration level1
- L1 Defense mode3 during reconfiguration level1
- L1 Defense mode3 during reconfiguration level0

Enabling additional cases is planned for future releases.

MAPLE-B3 PUSCH_EDF COP Channel Control Command

Added an API to allow modifying the user's Neighbor Ready parameters, after a PUSCH job was dispatched, and before MAPLE_PUSCH_USER_NEIGHBOR_READY opcode command was issued.

MAPLE3 Firmware Update

SmartDSP OS R05.16.00 is released with B4860/B4420 MAPLE3LW/MAPLE3W Firmware Release A30.
Note: there are API changes in the PUSCH2_EDF and PDSCH2_EDF.

The following new features were added to support the new μCode version:

- EQPE standalone with repeat mode.
- PUSCH2 EDF:
    - API Change for Steering bits enums in PUSCH driver:
        - MAPLE_PUSCH_STEERING_F0_PTR,
        - MAPLE_PUSCH_STEERING_F1_PTR,
        - MAPLE_PUSCH_STEERING_F2_PTR and MAPLE_PUSCH_STEERING_F3_PTR were replaced by
        - MAPLE_PUSCH_STEERING_F_PTR,

        - MAPLE_PUSCH_STEERING_GEN0_PTR,
        - MAPLE_PUSCH_STEERING_GEN1_PTR and
        - MAPLE_PUSCH_STEERING_GEN2_PTR respectively.
    - Changes in the PUSCH interrupts scheme – see simple demo for usage example:
        - Added ALL Ctrl Done interrupt, to allow the user to receive an interrupt when the last User control is done. This is due to fact that in CoMP the users are handled in an unexpected order.
        - Added All PU Termination interrupt.
        - In order for the ALL PU and ALL EQ Termination interrupts to work, the user must enable it on the BD, and in addition, should also provide the number of terminated users using 2 new fields for the PUSCH job num_pu_term and num_eq_term respectively.
    - The PUSCH_CH_CMD_GET_TERM_STATUS Ctrl Command now return 3 status flags: PUSCH_TERM_STATUS_ALL_CTRL_DONE, PUSCH_TERM_STATUS_ALL_EQ_TERM_DONE and PUSCH_TERM_STATUS_ALL_PU_TERM_DONE
    - New User Types for additional CoMP processing.
    - User Parameters Headers (UPH) RI_BITS_x (x == 0…4) now support a new value, to signal that the carrier has no RI.
    - Added a flag to the UPH to enable per user NV BETA source.
    - Added new field CQI_EXT_BITS to enable adding CQI bits amount on account of the carriers without RI.
- PUFFT:
    - PUFFT now supports new timing scheme – MAPLE Interrupt Service Routine (MPISR) which supports programmable link between SoC timers interrupts and PUFFT sector processing. Starting at this release, a device level interrupt may trigger one or more pufft jobs. This relieves the system designer from assigning more than a single device level interrupt in a case where all are synchronized
- eFTPE2:
    - Packed structure arrangement of the CMP_RSN and ADP_OVA_SCL_ST status fields of each of the jobs in the BD repeat option or of a single job in a non-repeat mode – see FTPE_BD_PACKED_EXP_STAT.
- PDSCH2 EDF:
    - PDSCH Minimum channel size is 3, to allow runtime optimizations.
    - API Change: Changes in the CSI Reference Signals and CS Reference Signals handling:
        - CS None Zero RS Ports and CS Zero Powered RS ports are deprecated.
        - Changes in maple_pdsch_ctrl_gain_rnti_t
        - Changes were made in the PDSCH Job hdr_config enumerations
        - Changes in the UE RS header structure.
    - Added 3 new steering bits fields to the driver to support steering for Import Data, CSI Header and CSRS Header Pointers.
    - Added flags to configure CSI RS headers parameters.
    - Added CSI RS Header Address to the PDSCH Job Extension structure
    - Added FCG bit in extended BD, user can now configure PDSCH to perform frequency correction on the data prior to IFFT processing.
    - Positioning RB Disable – user is now able to configure in which resource blocks, positioning-RS should not be placed in slot 0/1 using NPRSRBEN0/1 in extended BD.
    - Supporting multiple padding patterns for TDD support (using MPDSCHPDCP, MPDSCHPDC1P, and NT fields). Added MPDSCHPDC1P and NT fields to support this new feature.
    - Enabling padding pattern for Hybrid MBSFN (added MPDSCHPDC2P field).
- Added maple_version_t structure to help the user retrieve MAPLE firmware version details when calling the COP Device Ctrl command MAPLE_CMD_GET_UCODE_VERSION.

1.2.3 Demos

CPRI Auxiliary

CPRI auxiliary demo was enhanced to demonstrate execution of level 3 CPRI reconfiguration. The demo switches between channels 0&1 and channels 2&3 and changes the balance between the 1st and the 3rd CPRI units.
The demo is located in: SmartDSP\demos\starcore\b4860\cpri_auxiliary_demo.

1.2.4 Performance

SmartDSP OS R05.16.00 kernel performance figures are shown in Table 1.

The mode of calculation was such that each measurement was taken 5 times; once with cold instruction caches and the other without.

The WCS (worst case scenario) is the cold cache result.

The BCS (best case scenario) is the best of the results.

Table 1 SmartDSP B4860 and B4420 Performance Figures

		B4860			B4420		
		WCS	**BCS**	**AVG**	**WCS**	**BCS**	**AVG**
HWI prolog/epilog from HWI	HWI Epilog	260	49	49	259	49	49
	HWI Prolog	77	72	72	83	72	72
HWI prolog/epilog from SWI	HWI Epilog	230	89	89	233	88	90
	HWI Prolog	76	72	72	75	72	72
HWI prolog/epilog from Task	HWI Epilog	136	126	130	135	126	129
	HWI Prolog	76	72	72	75	72	72
High priority SWI from low priority SWI	SWI Epilog	196	81	82	190	81	82
	SWI Prolog	263	104	104	262	104	105
High priority	Task	106	106	106	106	106	106

Task from low priority Task	Epilog						
	Task Prolog	218	113	113	218	113	113
Low priority SWI from high priority SWI	SWI Epilog	143	136	139	141	138	139
	SWI Prolog	55	55	55	55	55	55
SWI from Task	SWI Epilog	149	139	139	141	139	139
	SWI Prolog	94	94	94	94	94	94

1.2.5 Important Notes

Note on B4860/B4420 rev2 cacheable spinlocks

SmartDSP OS R05.16.00 contains B4860/B4420 rev2.2 support for cacheable spinlocks with light barriers for cacheable memory. Therefore, "`os_shared_data_descriptor`" inside B4860/B4420 rev2 linker files must be with cacheable coherent attributes.

Note on ISR Stack size:

ISR stack size is defined in `os_config.h` and not in the linker file (as opposed to the CodeWarrior stationary configuration)

```
#define OS_STACK_SIZE <size>
```

Note on System Stack mapping:

The user MUST take care to map the system stack to the virtual address 0xBB400000 and to provide the actual physical mapping. Towards this, the default lcf has the symbol `_ARCH_SYS_STACK_VIRT_`BASE declared.

The relevant portion of the core's private lcf is as follows:

```
unit private (*) {
       MEMORY {
              os_system_stack_descriptor ("rw"): org = _ARCH_SYS_STACK_VIRT_BASE;
              os_background_stack_descriptor ("rw"): org = _ARCH_BCK_STACK_VIRT_BASE;
       }
       SECTIONS {
              descriptor_os_system_stack {
```

```
                .os_system_stack
            } > os_system_stack_descriptor;
            descriptor_os_background_stack {
                .os_background_stack
            } > os_background_stack_descriptor;
        }
}

address_translation (*) {
        os_system_stack_descriptor
        (0, ((STACK_DATA_MMU_DEF_HIGH << 32) | STACK_DATA_MMU_DEF_LOW)) :
        LOCAL_DDR0, AFTER(local_data_ddr0_descriptor);
        os_background_stack_descriptor
        (0, ((STACK_DATA_MMU_DEF_HIGH << 32) | STACK_DATA_MMU_DEF_LOW)) :
        LOCAL_DDR0, AFTER(os_system_stack_descriptor);
}

// Stack setting
_StackStartPhys = physical_address("StackStart");
```

Note on projects

All projects must add the relevant SoC (B4860/B4420) as well as the relevant core (SC3900) as compiler preprocessors.

1.2.6 Known Issues and Limitations

DPAA Integration Demo in High Addresses

DPAA integration demo using high addresses was not tested in this release due to lack of Linux BSP support. It will be tested once support is available. DPAA standalone test was successfully tested.

Debug Print

Debug print integration demo was not tested in this release due to environment problems. It will be tested once they are resolved. Debug print standalone tested was successfully tested.

Maple PDSCH2

The following Maple PDSCH new features were not tested for this release:

- Enabling padding pattern for Hybrid MBSFN (added MPDSCHPDC2P field).
- Supporting Positioning-RS and MBSFN-RS in the same subframe (no API change).

B4420 SOC timers

A suspected issue in B4420 SOC timers not enabling all B4420 SOC timers to invoke interrupts towards MAPLE. Due to this, the MAPLE PUFFT B4420 demo currently uses a single SOC timer.

CPRI Ethernet errata workaround

Scenario in which VSS and Ethernet data types of the same CPRI unit are owned by different cores is not tested.

Debug Print L1 Defense

When warm reset is invoked, if a core has begun sending a debug print message to NEXUS, messages from the other cores are discarded in the C-NPC until this core has completed sending the message (which happens after returning from reset).

B4420 Level 0 CPRI Reconfiguration

B4420 level 0 reconfiguration is supported for CPRI IQ and VSS usage only (Ethernet and HDLC are not yet supported).

DPAA L1 defense support

L1 defense SmartDSP OS module currently supports using DPAA integration scenario only. This means that Linux is initializing QMAN and BMAN and is allocating portals for DSP usage.

CPRI HDLC CRC error

A CRC error and packed discarded errors are sometimes received from CPRI hardware upon receive of first HDLC. Following frames are received correctly and without error.

While the issue needs to be checked, SmartDSP OS CPRI HDLC demo ignores this first RX callback in case it contains these errors.

MBECC and BD UNDDERRUN errors are disabled in HDLC demo, since those errors are being invoked frequently, without impact on transmitted or received data.

B4420 CPRI QDS Cross matrix configuration

B4420 QDS Cross matrix configuration script (external to SmartDSP OS) seems to be not configuring the CPRI lanes connections properly. As a result, it is not possible to test B4420 CPRI tests using optical connection (SFP) on the board. For this reason, CPRI auxiliary demo and auto negotiation demo were not tested on B4420 (so does not contain B4420 targets).

In following releases, after connectivity is corrected, B4420 targets will be tested and added to the demos.

CPRI units connection with cross cable on single SoC

When connecting 2 CPRI units on the same SoC using a cross cable (optical), only when the 2 CPRI units are configured as CPRI masters, the data is exchanged correctly.

If configuring one unit as slave (theoretically the correct configuration), data is corrupted or NULL.

This issue impacts CPRI full negotiation demo, which is using cross cable.

This will be fixed in following SmartDSP OS releases.

1.2.7 Compatibility

Silicon version and configuration

SmartDSP OS R05.16.00 supports B4860/B4420 rev2.2 – all tests were tested on B4860 rev2.2 silicon on QDS boards. B4860/B4420 tests were run using core frequency 1.2GHz, DDR frequency 1.86GHz, MAPLE frequency 600MHz, platform frequency 667MHz.

The B4860 reset configuration word (RCW) used for testing is:

140E0018 0F001218 00000000 00000000

1A8D0000 A000A400 FC025000 A9000000

01000000 00000000 00000000 0001B1F8

00000000 14000020 00000000 00000011

The B4420 reset configuration word (RCW) used for testing is:

140C0018 0F001218 00000000 00000000

1A9E0000 80002400 FC025000 A9000000

00000000 00000000 00000000 0003b000

00000000 14000020 00000000 00000011

Memory map

The DSP memory map used by R05.16.00 was modified to use higher than 32bit addresses, as is as following (B4860/B4420):

Memory region	Virtual address	Physical address	Size
DSP DDR (DDR controller 1)	0x80000000 -> 0xFFFFFFFF	0x100000000 -> 0x17FFFFFFF	0x80000000 (2GB)
CCSRBAR	0x7F000000 -> 0x7FFFFFF	0xFFE000000 -> 0xFFEFFFFFF	0x1000000
MAPLE MBUS	0x79000000 -> 0x79BFFFF	0xFE0000000 -> 0xFE0 BFFFF	0x400000 * 3 = 0xc00000)
M3 memory (if used)	0x7D000000 -> 0x7D07FFFF	0xC40000000-> 0xC4007FFFF	Up to 0x80000 (512KB)
DSP cluster registers (Bank0)	0xFEC00000 -> 0xFEC3FFFF	-	0x40000
Shared control structure (used by PA and DSP)	0xFFF00000 -> 0xFFFFFFFF	0x17FFF00000 -> 0x17FFFFFFF	0x100000 (max size)
PA DDR (DDR2 controller)	(PA access only)	0x00000000 -> 0x7FFFFFF	0x80000000 (2GB)
PA shared DDR	0x60000000 - 0x6EFFFFF	(PA configurable)	0xF000000 (240MB max size)
SC shared DDR	0x6F000000 – 0x6FFFFFF	(PA configurable)	0x1000000 (16MB max size)

CodeWarrior

SmartDSP OS R05.16.00 libraries and applications were compiled and tested with CodeWarrior for StarCore 10.9.0, Build 446.

Linux BSP

SmartDSP OS R05.16.00 is compatible with the B4860 QDS Linux BSP version “QORIQ SDK 1.5”.

1.3 Build Tools

New version of build tools v24.07.01.013 adds the following features:

- License upgrade to 10.9

- Add support for –save_temps option. This option allows saving the intermediate files (including the preprocessed file) while completing the compilation process.
- Improve pragma_novect to disable vectorization in the backend for a specific loop region.
- Add __restart_core library function call to be used for L1 defense validation.
- Add pragma and attribute novector to disable vectorization at function level.
- Add boundary check support in the linker.
- Add support to split a merged executable in its initial elf components.
- Improve compile time in DDA computation.
- Add support to place the startup above 4GB.
    - New directive added to to reserve a certain descriptor:
      reserve_descriptor_id(TYPE, ID)
      where: TYPE represents the type of the descriptor, with possible values: “data”, “code”
      and ID represents the id of the descriptor to be reserved.
    - API change in address_translation directive:

```
address_translation (task_list) [map11]
{
  virt_mem ( attribute [, application_attribute] ) : PHYS_MEM [org, len];
}
```

    is replaced by:

```
address_translation (task_list) [map11]
{
 virt_mem(regA_val,  regC_val):  PHYS_MEM[org  VAL|AFTER(virtual_memory2_name)]
 [descriptor(ID)];
}
```

  Where regA_val and regC_val represent the values used to program registers M_PSDAx / M_DSDAx and M_PSDCx / M_DSDCx respectively. The two values have to be set in both cases (bare-board and SDOS based applications).
  The new optional parameter "descriptor(ID)" will enable the user to specify which descriptor ID to be used in order to map the virtual memory in the MMU.

- Generate prototype.dll for Windows64

Performance Report

Table 2 Build Tools Performance Improvements vs. CW 10.8.3

Benchmark	Average Improvement	Max. Improvement	Max. Degradation
DSP	0.01%	5%	0.9%
Control Code	0%	0%	0%

Compile time report

Table 3 Compile Time Report

Benchmark	Total Compile Time (s)	Average Compile Time (s)	Files compiled in more than 20s
LTE 6.5patch 2	1433	4.00	MatrixInverse_B4860_QDS/ SBL1_ULSPK_MatrixInverse_8x8_BackSubst.c MatrixInverseCholesky_B4860_QDS/

			SBL1_ULSPK_MatInvChol_8x8_2x.c PDSCH_AL_B4860_rev2_ifft_dis_lib/ SBL1_PDSCH_AL_Parser.c PDSCH_AL_B4860_rev2_lib/SBL1_PDSCH_AL_Parser.c pusch_config_0_rev2_lib/ SBL1_PUSCH_AL.c pusch_config_1_rev2_lib/ SBL1_PUSCH_AL.c
SDOS 5.16	783	1.56	maple_driver_config_eqpe.c

Compile time benchmarks were run on **Intel Core i5-3230 CPU@2.60 Ghz, RAM – 4GB, OS – win7 32bit, HDD – 7200 RPM**

1.4 StarCore Debugger

The following features were implemented in this release:

- Move core initialization sequence (MMU translations and cache state) from start-up to debugger
- Update initialization file to support DDR initialization over 4GB
- Update B4 initialization files to support A-007212 errata
- Debugger license upgraded to 10.9

1.5 IDE

The following feature was added in this release:

- ecd option –disableParallelBuild added to disable parallel build from command line on a given build command. This option shall be used along with the build command.

1.6 Software Analysis

The following improvements were added in this release:

- Generating code coverage reports for high amount of trace and high size of executable has been improved. Small code coverage rate reports can be generated for size of executable up to 500 MB while high code coverage rate reports can be generated for size of executable up to 200 MB, no matter the amount of trace for both small and high code coverage rate reports.
- Multicore continuous trace has been enabled. Multicore continuous trace is usefull to avoid multicore trace messages being lost either because of multicore trace messages' high frequency/rate or because trace buffer size is not enough to accommodate all multicore trace messages.
- Exclude symbols feature has been improved to exclude C++ symbols also along with excluding C symbols.

1.7 Documentation

Getting Started Guide for StarCore DSPs:

Quick Start for StarCore DSPs:

Service Pack Updater Quick Start:

- Section updated – "System Requirements"

Targeting manual:

- Update the section 14.4 ELF2xx Utility

StarCore C/C++ Compiler User Guide

- Added a new section , “2.2.21 How to disable automatic vectorization”
- Updated the section, “**4**.2.1.8 #pragma novector and __attribute__((novector))”
- Updated the section 4.1.12 Compiler frontend warning messages
- Updated the Table 4-8. File and message output options for “-save-temps” command

StarCore SC3000 Linker User Guide

- New section - “2.2.16 How to reserve an MMU descriptor ID”
- Sections updated:
    - 2.1.3 How to Define and use a Custom set of Tasks
    - 2.1.5 How to Setup Cache
    - 2.1.8 How to Define Physical Memory Layout for a Multi-core Application
    - 2.1.9 How to Modify the LCF When Each Core Runs Different Code
    - 2.1.12 How to Define Virtual Memory for Read-Write-Execute (RWX) Access
    - 2.2.2 How to Define Private Data Sections for Multiple Cores
    - 2.2.5 How to Map Virtual Memory Areas to Physical Memory Address Space
    - 2.2.7 How to Share Code and Data Partially Among Different Cores
    - 2.2.8 How to Limit Code and Data Visibility at Core Level
    - 2.2.10 How to Run Multiple Tasks on the Same Core
    - 2.2.12 How to place a symbol in an another section in LCF
    - 3.1.2.11 Specifying Address Translation Construct
    - 3.1.4.1 Changes Made to Support Flexible Startup Configuration
    - 3.2.1 Understanding startup environment
    - 8.1 Predefined Symbols for MMU Descriptors
    - 8.2 Predefined Physical Memory Regions

Tracing and Analysis Users Guide:

- New section - “3.12.6 Multicore continuous trace”
- Sections updated:
    - 3.2.1 Simulator Profiling
    - 3.3.1 Simulator Profiling
    - 3.3.2 Hardware Profiling
    - 3.4.1 Using Simulator
    - 3.4.2 Using Hardware
    - 3.5 Viewing Data

- 3.5.1.2 Viewing Timeline Data
- 3.5.1.3 Viewing Critical Code Data
- 3.5.1.5 Viewing Call Tree Data
- 3.5.2.1 Viewing Trace Data
- 3.6.1 Software Analysis View
- 3.6.2 Critical Code View
- 3.7.1.1 Create an Attach Launch Configuration
- 3.11 Importing Offline SC3900 Trace with Hardware Trace Configuration
- Chapter 5 Setting Counterpoints
- 6.1.2.3 Sampling Pane
- 6.1.2.6 Session pane
- 7.1 Run Sample Python Script

SDOS OS Concept User Guide

- New section – "4.7.7 Reconfiguration"
- Updated the section, 4.7.8 CPRI Ethernet Errata A-007968 Workaround

SDOS API Reference Manual – updated for SDOS R05.16.00

License, copyright

The license key was updated in CodeWarrior for StarCore 3900FP DSP v10.9.0 release to key 10.9.

System requirements

Recommended configurations:

- 3GHz Intel® Pentium® P4 processor or better. Dual-core processor preferable.
- Microsoft® Windows 7, Windows Server 2012 R2
- 2GB RAM (Experience on machines with 1GB RAM is significantly reduced)
- 2.3 GB free disk space

Note: 500MB of free space is required on the OS drive, regardless of the free space available on the destination drive.

Getting Started

The Getting Started Guide for StarCore DSPs.pdf provides instructions on how to install the product, configure B4 QDS boards, and how to work with projects.

To install the CodeWarrior software, perform the following steps:
1. Run the installer - the install wizard appears.
2. Follow the wizard's on-screen instructions to install the CodeWarrior software.
3. When installation completes, the **InstallShield Wizard Completed** page appears.

4. Check the Display Documentation checkbox.

5. Click **Finish.**

The SC folder contains several quick start guides for the product:

- Eclipse Quick Reference Card.pdf
- Ethernet TAP Quick Start.pdf
- Getting Started Guide for StarCore DSPs.pdf
- Quick Start for StarCore DSPs.pdf
- Service Pack Updater Quick Start.pdf

Comprehensive User Guides

The complete product documentation can be found in the folder SC\Help\PDF and contains the following documents:

- Build Tools Message Reference Manual.pdf
- CodeWarrior Common Features Guide.pdf
- CodeWarrior TAP Users Guide.pdf
- Ethernet TAP Users Guide.pdf
- EWL C Reference.pdf
- EWL C++ Reference.pdf
- Gigabit TAP Users Guide.pdf
- SmartDSP OS API Reference Manual.pdf
- SmartDSP OS User Guide.pdf
- StarCore ABI Reference Manual.pdf
- StarCore Assembler User Guide.pdf
- StarCore C-C++ Compiler User Guide.pdf
- StarCore FAQ Guide.pdf
- StarCore SC3000 Linker User Guide.pdf
- StarCore Simulator User Guide.pdf
- Targeting StarCore DSPs.pdf
- Tracing and Analysis Users Guide.pdf
- USB TAP Users Guide.pdf

Known Issues and Limitations

Known Issue	Abstract	Workaround
Build Tools		
CMPSC-408	Rtlib function pow and powf are too slow	Use float __powfi(float,int), double __powi(double,int), long double __powli(long double, int) instead of pow and powf if the exponent is integer. These functions are defined in math.h
Simulator		
DTSIM-71	Updated "maple_pusch" demo not running successfully on simulator on some configurations	use the "b4860 pusch shared channel" configuration instead.
DTSIM-72	Maple PDSCH demo Performance target runs forever	Use other Maple PDSCH configurations
DTSIM-369	[SC10.8.2][SDOS5.14] maple_reset demo fails	use the B4420 ISS target, do not define SIMULATOR, change TEST_NUM_OF_ITERATIONS to 2 instead of 200 in file ap_config.h
SDOS		
SDOS-1034	Need support for warm reset even if the dsp cores are in non-interrupted state.	NA
SDOS-1055	Need to modify ocnDmaChainNext() API on Starcore DMA driver to cop with sync CC and EOLSD	NA
Software Analysis		
ENGR00286565	Trace collection does not work on B4420 when using Aurora link	N/A

Please see the SDOS chapter on the list of additional limitations.

Contact Information

User Forum and FAQ

After looking through these release notes, and the documentation that comes with the installation of CodeWarrior, the next best place to look for answers to your questions is the online user forums located at

http://forums.freescale.com

Please check:

- **CodeWarrior for StarCore DSPs** forum for issues related to CodeWarrior development tools. The Frequently Asked Questions about CodeWarrior for StarCore DSP are posted here.
- **StarCore DSPs** forum for issues related to the silicon and hardware platforms.

The forums provide a great way to learn by seeing the questions and answers posted by other users. Of course, you can post your own questions and responses as well.

Contacting Freescale

Finally, if you still have questions not addressed in the release notes, or wants to provide feedback, please use the Freescale online support web page. To use this page, follow these steps:

1. In a web browser, go to http://www.freescale.com/TechSupport.

   Freescale's **Technical Support** web page appears.
2. On this page, click the Create service request online link.

   The **New Service Request — Category/Topic** page appears.
3. From the Category dropdown menu, select Technical Request.
4. From the Topic dropdown menu, select CodeWarrior (or other appropriate topic).
5. Click **Next**.

   The **New Service Request — SR Details** page appears.
6. In this page, enter the requested information.

   At a minimum, enter information in each field marked by an *.
7. Click **Submit**.

   If you are already logged in, the **Service Request Confirmation** page appears. Go to the last step.

   If you are not already logged in, the **Log-in** page appears.
8. If you are a registered member, login with your user name and password.

   The **Service Request Confirmation** page appears. Go to the last step.
9. If you have not yet registered,
   a. If you want to become registered member, click **Register Now** and complete the registration process.

      The **Service Request Confirmation** page appears.
   b. If you do not want to register, supply your contact information in the **I do not want to register - Provide contact information** form and click **Submit**.

      The **Service Request Confirmation** page appears.
10. Click **Done**.

Your service request is submitted.

Release Quality

This list includes the issues reported by external customers that are now fixed:

SmartDSP OS	
ENGR00350311	osSpinLockInitialize is misused in some of the drivers
ENGR00350463	CPRI HDLC issue detected
ENGR00350699	CPRI driver Eth WA - the internal VSS TX buffer allocation may be not aligned
ENGR00353418	CPRI driver internal functions getEthernetTxConfNumber and getHdlcTxConfNumber return wrong values which result in unnecessary clock cycles usage
ENGR00353707	osSioBufferGet(...) and osSioBufferPut(...) are not correctly wrapping to beginning of buffer
ENGR00354646	CPRI Ethernet and HDLC demos invoke printf(...) in some cases just before the end of runtime
ENGR00355741	Removed unrelavant documentation of os_message API.
ENGR00357594	typo error in file sc39xx_dtu.h

ENGR00357806	Bug inside osHwiDispatcherCreate()
ENGR00358482	osCmeExtQuery shouldn't check for second parameter when the third is CME_PQSYNC_MEM
ENGR00358708	CPRI driver: Wrong value written to reg CPRIn_EX_DELAY_CONFIG
ENGR00358860	wrong implementation of osMmuDataErrorDetect and osMmuProgErrorDetect
ENGR00359642	CPRI driver, there is a wrong assertion in debug mode for Auxiliary mode
ENGR00359894	CPRI driver - A few registers which are read-modified-written shall be reset before reconfiguration
ENGR00360129	CPRI 2 groups AN issues
ENGR00361131	CPRI memory map - correcting register name from "iccr" to "ccr"
SDOS-1143	CPRI driver fix for A-006392: CPRI generates shorter reset request than required by standard
SDOS-1118	osTickTime API returns incorrect values
SDOS-1139	osHwTimerValueGet is not protected and may return wrong value if interrupted
SDOS-1131	Clearing Pending Interrupts not done correctly when DSP is in Guest mode
SDOS-1117	osTaskDelete function doesn't return the pid/did before deleting the task
SDOS-1108	need to protect Reading and writing of CME registers in functions osMmuProgVirtProbe, osMmuProgVirtToPhys, osMmuProgSegmentProbe
SDOS-1105	osMmuProgVirtProbe() need to make sure the channel is idle by reading CME_D
SDOS-1109	OS_MEM_DDR0_SHARED have cachable attributes
SDOS-1091	Need to add IBSS barrier in osL1dResetRequest
SDOS-1092	write to the semaphore register did not followed by a barrier
SDOS-1107	osCmeExtQuery poll cme_cr instead of cme_qcr to verify channel is free
SDOS-1129	CPRI possible defect in SDOS 5.13.0
SDOS-1119	osHwTimerValueGet might return incorrect value
SDOS-1112	CPRI reconfiguration level 0 two groups parallel execution prevention is buggy
SDOS-1111	CPRI reconfiguration doesn't reset IQ channels steering bits information
SDOS-1101	CPRI sync - if external then timer state machine must be polled
SDOS-1099	CPRI internal structure field external_sync_active is of wrong type/width
SDOS-1090	OCN DMA SR register is not cleared correctly after l1 defense mode 3 warm reset
SDOS-1089	CPRI eth driver not setting physical address correctly
SDOS-1086	CPRI Ethernet WA, very large Ethernet packets (such as 16KBytes) sets the system in a too long DI state
SDOS-1084	[SDOS 5.16EB2] maple_pusch demo run fail on "Shared Channel" and "Multicore Simple Release" configurations
SDOS-1083	CPRI driver, wrong assertion for shared sync mode
SDOS-1079	[SDOS R5.16EB1] Maple_pdsch run fail on Debug and Release configuration
SDOS-1043	CPRI driver Eth WA - the internal VSS TX buffer allocation may be not aligned
SDOS-1077	maplePdschSanityCheck() in maple_pdsch performs user header checks using incorrect PDSCH_USER_HDR_MASK1
SDOS-1120	Need to update SDOS release notes regarding Job_num constrains
SDOS-1136	Definitions in SmartDSP\drivers\maple\rev3\include\maple_pdsch.h
SDOS-1138	Unprotected critical code in maple soc timer configuration
SDOS-1098	CPRI Eth doesn't support re-opening channel after L1D
SDOS-1123	possible CPRI driver bug in sdos 5.13.0
SDOS-1115	A few C&M (Eth and HDLC) CPRI demos pending on last packet is wrongly calculated
Build Tools	
ENGR00351919	Internal Compiler Error occurred
ENGR00348911	SC compiler does not optimize sign extend instructions
CMPSC-201	compiler does not generate macd.im.2x instruction for __l_macd_im_2x_m() intrinsic causing application to fail
CMPSC-456	Investigate post linking crash on linux64 machine

CMPSC-400	Invalid code generation for dividend/quotient
CMPSC-374	ICE in iscape in file CTemplateTools.c
CMPSC-314	Compiler Crash, issuing internal error
CMPSC-421	Issues with Uninit_And_Zeroed_Globals_To_Bss=TRUE
CMPSC-420	while() loop conter is calculated wrongly
CMPSC-364	Compiler generates bad code with O3
CMPSC-174	Function name is null for some assembly inline functions
CMPSC-307	Wrong results when compiling with O3
Debugger	
DTBPDBG-23	CW debugger shell command ‘cls’ does not work in tcl
Software Analysis	
ENGR00350621	[10.8.3][B4] CW crashes when opening any profiling viewer at decoding stage
ENGR00346219	[10.8.2][B4860][Critical Code] Out of memory at generating code coverage for over 50Mb size files
DTSA-306	IDE crash when clicking on suspend while profiler is enabled on PACC
IDE	
ENGR00353807	[License plugin] No source files should be made visible to the user
Simulator	
STSIM-541	Wrong results when using B4860 ISS model – PUSCH-EDF
Documentation	
DNDOCTRK-858	Remove the support for Windows XP and VISTA from the documentation
DNDOCTRK-871	Document the default value of the compiler frontend warning options
DNDOCTRK-853	Update the Import Trace Configuration Page
DNDOCTRK-854	Update the description of the Trace scenarios: None, Program Trace
DNDOCTRK-855, DNDOCTRK-856	In Tracing and Analysis User Guide.pdf update chapter 7.1
DNDOCTRK-865	In Tracing and Analysis User Guide.pdf, the Edit Group feature is not supported in Timeline viewer
DNDOCTRK-864	In Tracing and Analysis User Guide.pdf, add description for function percentage in Timeline viewer
DNDOCTRK-863	In Tracing and Analysis User Guide.pdf, explain “Export to dot” option in Call Tree viewer
DNDOCTRK-862	In Tracing and Analysis User Guide.pdf, document the supported options in Critical Code viewer
DNDOCTRK-860	In Tracing and Analysis User Guide.pdf, update table 3-17 related to Trace Control description
DNDOCTRK-859	In Tracing and Analysis User Guide.pdf, document counterpoint support only in Profiling trace

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
www.freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: CW_SC_3900FP_v10.8.3
Rev. 1.2
03/2015