

Document Title	SWS_MFXLibrary: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1	SWS_MFXLibrary	3
1.1	Specification Item SWS_Mfx_00024	3

1 SWS_MFXLibrary

1.1 Specification Item SWS_Mfx_00024

Trace References:

[SRS_BSW_00305](#)

Content:

Here is the list of implemented functions.

Function ID[hex]	Function prototype
0x070	uint8 Mfx_Mul_u8u8_u8(uint8 , uint8);
0x071	uint8 Mfx_Mul_s8s8_u8(sint8 , sint8);
0x072(*)	uint8 Mfx_Mul_s8u8_u8(sint8 , uint8);
0x073(*)	sint8 Mfx_Mul_s8u8_s8(sint8 , uint8);
0x074	sint8 Mfx_Mul_s8s8_s8(sint8 , sint8);
0x075	uint16 Mfx_Mul_u16u16_u16(uint16 , uint16);
0x076(*)	uint16 Mfx_Mul_s16u16_u16(sint16 , uint16);
0x077(*)	sint16 Mfx_Mul_s16u16_s16(sint16 , uint16);
0x078(***)	sint32 Mfx_Mul_s16u16_s32(sint16 , uint16);
0x079(**)	sint32 Mfx_Mul_s16s16_s32(sint16 , sint16);
0x07A(*)	sint32 Mfx_Mul_s32u16_s32(sint32 , uint16);
0x07B(**)	uint32 Mfx_Mul_u16u16_u32(uint16 , uint16);
0x07C(**)	uint32 Mfx_Mul_u32u16_u32(uint32 , uint16);
0x07D	uint16 Mfx_Mul_s16s16_u16(sint16 , sint16);
0x07E	uint8 Mfx_Mul_s16s16_u8(sint16 , sint16);
0x07F	sint8 Mfx_Mul_s16s16_s8(sint16 , sint16);
0x080	sint16 Mfx_Mul_s16s16_s16(sint16 , sint16);
0x081	uint32 Mfx_Mul_u32u32_u32(uint32 , uint32);
0x082	sint32 Mfx_Mul_u32u32_s32(uint32 , uint32);
0x083(*)	uint16 Mfx_Mul_s32u32_u16(sint32 , uint32);
0x084(*)	uint32 Mfx_Mul_s32u32_u32(sint32 , uint32);
0x085(*)	sint32 Mfx_Mul_s32u32_s32(sint32 , uint32);
0x086	uint32 Mfx_Mul_s32s32_u32(sint32 , sint32);
0x087	uint8 Mfx_Mul_s32s32_u8(sint32 , sint32);
0x088	sint8 Mfx_Mul_u32u32_s8(uint32 , uint32);
0x089	sint8 Mfx_Mul_s32s32_s8(sint32 , sint32);
0x08A	sint16 Mfx_Mul_u32u32_s16(uint32 , uint32);
0x08B	sint16 Mfx_Mul_s32s32_s16(sint32 , sint32);
0x08C	uint16 Mfx_Mul_s32s32_u16(sint32 , sint32);
0x08D	sint32 Mfx_Mul_s32s32_s32(sint32 , sint32);
0xA21	sint16 Mfx_Mul_u32s32_s16(uint32 , sint32);

Function ID[hex]	Function prototype
0xA22	sint8 Mfx_Mul_u32s32_s8(uint32 , sint32);
0xA23	uint8 Mfx_Mul_u32s32_u8(uint32 , sint32);
0xA24	uint16 Mfx_Mul_u32u32_u16(uint32 , uint32);
0xA25	uint8 Mfx_Mul_u32u32_u8(uint32 , uint32);
0xA26	uint8 Mfx_Mul_u8s8_u8(uint8 , sint8);
0xA27	sint8 Mfx_Mul_u8s8_s8(uint8 , sint8);
0xA28	uint16 Mfx_Mul_u16s16_u16(uint16 , sint16);
0xA29	sint16 Mfx_Mul_u16s16_s16(uint16 , sint16);
0xA2A	sint32 Mfx_Mul_u16s16_s32(uint16 , sint16);
0xA2B	uint16 Mfx_Mul_u32s32_u16(uint32 , sint32);
0xA2C	uint32 Mfx_Mul_u32s32_u32(uint32 , sint32);
0xA2D	sint32 Mfx_Mul_u32s32_s32(uint32 , sint32);
0xA2E	sint16 Mfx_Mul_u16u16_s16(uint16 , uint16);
0xA2F	uint8 Mfx_Mul_u16u16_u8(uint16 , uint16);
0xA70	uint8 Mfx_Mul_u16s16_u8(uint16 , sint16);
0xA71	sint8 Mfx_Mul_u16u16_s8(uint16 , uint16);
0xA72	sint8 Mfx_Mul_u16s16_s8(uint16 , sint16);
0xA74	sint8 Mfx_Mul_u8u8_s8(uint8 , uint8);

Note:

- The function prototypes with the (*) in the above table, do not follow the common requirement that with mixed data type the first parameter is the unsigned one and then comes the signed one. The new added function prototypes are correcting this.

The function prototypes with the (**) in the above table, are already covered by the 32 bit services function prototypes (e.g; Mfx_Mul_s32u16_s32 → Mfx_Mul_s32s32_s32) or can be written as normal C line.

Therefore, all these marked function prototypes will be removed in a major future release.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #70063: Mfx: Redundant due to covered by 32 bit service

Problem description:

In SWS_Mfx_00024, the following function is covered with 32 bit service.

0xA2A sint32 Mfx_Mul_u16s16_s32(uint16 , sint16);

is covered by the following 32 bit service

0xA2D sint32 Mfx_Mul_u32s32_s32(uint32 , sint32);

Mark it with (**)

–Last change on issue 70063 comment 1–

Agreed solution:

SWS_Mfx_00024]

Here is the list of implemented functions.

Function ID[hex] Function prototype

0x070

uint8 Mfx_Mul_u8u8_u8(uint8 , uint8);

0x071

uint8 Mfx_Mul_s8s8_u8(sint8 , sint8);

0x072(*)

uint8 Mfx_Mul_s8u8_u8(sint8 , uint8);

0x073(*)

sint8 Mfx_Mul_s8u8_s8(sint8 , uint8);

0x074

sint8 Mfx_Mul_s8s8_s8(sint8 , sint8);

0x075

uint16 Mfx_Mul_u16u16_u16(uint16 , uint16);

0x076(*)

uint16 Mfx_Mul_s16u16_u16(sint16 , uint16);

0x077(*)

sint16 Mfx_Mul_s16u16_s16(sint16 , uint16);

0x078

sint32 Mfx_Mul_s16u16_s32(sint16 , uint16);

0x079

sint32 Mfx_Mul_s16s16_s32(sint16 , sint16);

0x07A

sint32 Mfx_Mul_s32u16_s32(sint32 , uint16);

0x07B

uint32 Mfx_Mul_u16u16_u32(uint16 , uint16);

0x07C

uint32 Mfx_Mul_u32u16_u32(uint32 , uint16);

0x07D

uint16 Mfx_Mul_s16s16_u16(sint16 , sint16);

0x07F

sint8 Mfx_Mul_s16s16_s8(sint16 , sint16);

0x080

sint16 Mfx_Mul_s16s16_s16(sint16 , sint16);

0x081

uint32 Mfx_Mul_u32u32_u32(uint32 , uint32);

0x082

```
sint32 Mfx_Mul_u32u32_s32( uint32 , uint32);  
0x083  
uint16 Mfx_Mul_s32u32_u16( sint32 , uint32);  
0x084  
uint32 Mfx_Mul_s32u32_u32( sint32 , uint32);  
0x085  
sint32 Mfx_Mul_s32u32_s32( sint32 , uint32);  
0x086  
uint32 Mfx_Mul_s32s32_u32( sint32 , sint32);  
0x087  
uint8 Mfx_Mul_s32s32_u8( sint32 , sint32);  
0x088  
sint8 Mfx_Mul_u32u32_s8( uint32 , uint32);  
0x089  
sint8 Mfx_Mul_s32s32_s8( sint32 , sint32);  
0x08A  
sint16 Mfx_Mul_u32u32_s16( uint32 , uint32);  
0x08B  
sint16 Mfx_Mul_s32s32_s16( sint32 , sint32);  
0x08C  
uint16 Mfx_Mul_s32s32_u16( sint32 , sint32);  
0x08D  
sint32 Mfx_Mul_s32s32_s32( sint32 , sint32);  
0xA21  
sint16 Mfx_Mul_u32s32_s16( uint32 , sint32);  
0xA22  
sint8 Mfx_Mul_u32s32_s8( uint32 , sint32);  
0xA23  
uint8 Mfx_Mul_u32s32_u8( uint32 , sint32);  
0xA24  
uint16 Mfx_Mul_u32u32_u16( uint32 , uint32);  
0xA25  
uint8 Mfx_Mul_u32u32_u8( uint32 , uint32);  
0xA26  
uint8 Mfx_Mul_u8s8_u8( uint8 , sint8);  
0xA27  
sint8 Mfx_Mul_u8s8_s8( uint8 , sint8);  
0xA28  
uint16 Mfx_Mul_u16s16_u16( uint16 , sint16);  
0xA29  
sint16 Mfx_Mul_u16s16_s16( uint16 , sint16);  
0xA2A  
sint32 Mfx_Mul_u16s16_s32( uint16 , sint16);
```

```

0xA2B
uint16 Mfx_Mul_u32s32_u16( uint32 , sint32);
0xA2C
uint32 Mfx_Mul_u32s32_u32( uint32 , sint32);
0xA2D
sint32 Mfx_Mul_u32s32_s32( uint32 , sint32);
0xA2E
sint16 Mfx_Mul_u16u16_s16( uint16 , uint16);
0xA2F
uint8 Mfx_Mul_u16u16_u8( uint16 , uint16);
0xA70
uint8 Mfx_Mul_u16s16_u8( uint16 , sint16);
0xA71
sint8 Mfx_Mul_u16u16_s8( uint16 , uint16);
0xA72
sint8 Mfx_Mul_u16s16_s8( uint16 , sint16);
0xA74
sint8 Mfx_Mul_u8u8_s8( uint8 , uint8);
( )
    
```

Note:

The function prototypes with the (*) in the above table, do not follow the common requirement that with mixed data type the first parameter is the unsigned one and then comes the signed one. The new added function prototypes are correcting this.

–Last change on issue 70063 comment 47–

BW-C-Level:

Application	Specification	Bus
1	1	1