**修改记录**

本次分配到了一个文件，Im\_mxic。

1. **拆分变更**

将im\_mxic拆分成了immxic、mxicutlis、mxicacccapa、mxicmasterslave、mxicmaster、mxicslot、mxicmonitoraccess、mxiclevelport，共计8个类。

所有类的头文件和源文件均在ImageMacro文件夹。

**（二）公有方法变更：**

1、immxic.h：

INT32 Im\_MXIC\_Init( E\_IM\_MXIC\_UNIT unit ); 改为：

INT32 im\_mxic\_init( ImMxic \*self, E\_IM\_MXIC\_UNIT unit );

INT32 Im\_MXIC\_Reset( E\_IM\_MXIC\_UNIT unit ); 改为：

INT32 im\_mxic\_reset( ImMxic \*self, E\_IM\_MXIC\_UNIT unit );

INT32 Im\_MXIC\_Start\_Config( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_CONFIG\_ARBITER const\* const target\_arbiter ); 改为：

INT32 im\_mxic\_start\_config( ImMxic \*self, E\_IM\_MXIC\_UNIT unit, const T\_IM\_MXIC\_CONFIG\_ARBITER\* const targetArbiter );

INT32 Im\_MXIC\_Start\_Memory\_Access\_Detect( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_MEMORY\_ACCESS\_SLAVE const\* const param );改为：

INT32 im\_mxic\_start\_memory\_access\_detect( ImMxic \*self, E\_IM\_MXIC\_UNIT unit, const T\_IM\_MXIC\_MEMORY\_ACCESS\_SLAVE\* const param );

INT32 Im\_MXIC\_Stop\_Memory\_Access\_Detect( E\_IM\_MXIC\_UNIT unit );改为：

INT32 im\_mxic\_stop\_memory\_access\_detect( ImMxic \*self, E\_IM\_MXIC\_UNIT unit );

INT32 Im\_MXIC\_Set\_Memory\_Access\_Detect( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_MEMORY\_ACCESS\_SLAVE const\* const param );改为：

INT32 im\_mxic\_set\_memory\_access\_detect( ImMxic \*self, E\_IM\_MXIC\_UNIT unit, const T\_IM\_MXIC\_MEMORY\_ACCESS\_SLAVE\* const param );

INT32 Im\_MXIC\_Get\_Memory\_Access\_Detect( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_MEMORY\_ACCESS\_SLAVE\* const param );改为：

INT32 im\_mxic\_get\_memory\_access\_detect( ImMxic \*self, E\_IM\_MXIC\_UNIT unit,

T\_IM\_MXIC\_MEMORY\_ACCESS\_SLAVE\* const param );

INT32 Im\_MXIC\_Set\_Reg\_Read\_Switch( E\_IM\_MXIC\_UNIT unit, UCHAR sw );改为：

INT32 im\_mxic\_set\_reg\_read\_switch( ImMxic \*self, E\_IM\_MXIC\_UNIT unit, UCHAR sw );

INT32 Im\_MXIC\_Get\_Reg\_Read\_Switch( E\_IM\_MXIC\_UNIT unit, UCHAR \*const sw );改为：

INT32 im\_mxic\_get\_reg\_read\_switch( ImMxic \*self, E\_IM\_MXIC\_UNIT unit, UCHAR \*const sw );

INT32 Im\_MXIC\_Set\_Output\_Port( UCHAR target, E\_IM\_MXIC\_PORT\_GR output\_port );改为：

INT32 im\_mxic\_set\_output\_port( ImMxic \*self, UCHAR target, E\_IM\_MXIC\_PORT\_GR outputPort );

INT32 Im\_MXIC\_Get\_Output\_Port( UCHAR target, E\_IM\_MXIC\_PORT\_GR\* const output\_port );改为：

INT32 im\_mxic\_get\_output\_port( ImMxic \*self, UCHAR target, E\_IM\_MXIC\_PORT\_GR\* const outputPort );

INT32 Im\_MXIC\_Set\_Output\_Port\_All( T\_IM\_MXIC\_OUTPUT\_PORT const\* const out\_port );改为：

INT32 im\_mxic\_set\_output\_port\_all( ImMxic \*self, const T\_IM\_MXIC\_OUTPUT\_PORT\* const outPort );

INT32 Im\_MXIC\_Get\_Output\_Port\_All( T\_IM\_MXIC\_OUTPUT\_PORT \*const out\_port );改为：

INT32 im\_mxic\_get\_output\_port\_all( ImMxic \*self, T\_IM\_MXIC\_OUTPUT\_PORT \*const outPort );

VOID Im\_MXIC\_Int\_Handler( E\_IM\_MXIC\_UNIT unit );改为：

VOID im\_mxic\_int\_handler( E\_IM\_MXIC\_UNIT unit );

2、mxiclevelport.h：

因参数太多，将原有的参数中的3个整合进了一个结构体LevelPortArgs。

INT32 Im\_MXIC\_Set\_Clock\_Enable( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_CLOCK const\* const clock );改为：

INT32 mxic\_level\_port\_set\_clock\_enable( MxicLevelPort \*self, E\_IM\_MXIC\_UNIT unit, const T\_IM\_MXIC\_CLOCK\* const clock );

INT32 Im\_MXIC\_Get\_Clock\_Enable( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_CLOCK\* const clock );改为：

INT32 mxic\_level\_port\_get\_clock\_enable( MxicLevelPort \*self, E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_CLOCK\* const clock );

INT32 Im\_MXIC\_Set\_LevelPort\_Port( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, UCHAR master, E\_IM\_MXIC\_PORT port );改为：

INT32 mxic\_level\_port\_set\_levelport\_port( MxicLevelPort \*self, LevelPortArgs portArgs,

UCHAR master, E\_IM\_MXIC\_PORT port );

INT32 Im\_MXIC\_Get\_LevelPort\_Port( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, UCHAR master, E\_IM\_MXIC\_PORT\* const port );改为：

INT32 mxic\_level\_port\_get\_levelport\_port( MxicLevelPort \*self, LevelPortArgs portArgs,

UCHAR master, E\_IM\_MXIC\_PORT\* const port );

INT32 Im\_MXIC\_Get\_LevelPort\_Level( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, UCHAR master, E\_IM\_MXIC\_LEVEL\* const level );改为：

INT32 mxic\_level\_port\_get\_levelport\_level( MxicLevelPort \*self, LevelPortArgs levelArgs,

UCHAR master, E\_IM\_MXIC\_LEVEL\* const level );

INT32 Im\_MXIC\_Set\_LevelPort\_Level( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, UCHAR master, E\_IM\_MXIC\_LEVEL level );改为：

INT32 mxic\_level\_port\_set\_levelport\_level( MxicLevelPort \*self, LevelPortArgs levelArgs,

UCHAR master, E\_IM\_MXIC\_LEVEL level );

INT32 Im\_MXIC\_Set\_LevelPort\_All( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_LEVELPORT\* all\_levelport );改为：

INT32 mxic\_level\_port\_set\_levelport\_all( MxicLevelPort \*self, E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_LEVELPORT\* allLevelport );

INT32 Im\_MXIC\_Get\_LevelPort\_All( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_LEVELPORT\* all\_levelport );改为：

INT32 mxic\_level\_port\_get\_levelport\_all( MxicLevelPort \*self, E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_LEVELPORT\* allLevelport );

3、mxicacccapa.h：

Im\_MXIC\_Set\_Acceptance\_Capability( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, UCHAR capability );改为：

INT32 mxic\_acc\_capa\_set\_acceptance\_capability( MxicAccCapa \*self, E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wrArbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, UCHAR capability );

Im\_MXIC\_Set\_Acceptance\_Capability\_All\_Port( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_ACCEPTANCE\_CAPABILITY const\* const all\_capability );改为：

INT32 mxic\_acc\_capa\_set\_acceptance\_capability\_all\_port( MxicAccCapa \*self, E\_IM\_MXIC\_UNIT unit, const T\_IM\_MXIC\_ALL\_ACCEPTANCE\_CAPABILITY\* const allCapability );

INT32 Im\_MXIC\_Get\_Acceptance\_Capability\_All\_Port( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_ACCEPTANCE\_CAPABILITY\* const all\_capability );改为：

INT32 mxic\_acc\_capa\_get\_acceptance\_capability\_all\_port( MxicAccCapa \*self, E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_ACCEPTANCE\_CAPABILITY\* const allCapability );

Im\_MXIC\_Set\_Acceptance\_Capability\_Group( E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER\_GR arbiter, E\_IM\_MXIC\_PORT\_GR port, UCHAR capability );改为：

INT32 mxic\_acc\_capa\_set\_acceptance\_capability\_group( MxicAccCapa \*self, E\_IM\_MXIC\_WR\_ARBITER wrArbiter, E\_IM\_MXIC\_SPEC\_ARBITER\_GR arbiter, E\_IM\_MXIC\_PORT\_GR port, UCHAR capability );

Im\_MXIC\_Get\_Acceptance\_Capability\_Group( E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER\_GR arbiter, E\_IM\_MXIC\_PORT\_GR port, UCHAR\* capability );改为：

INT32 mxic\_acc\_capa\_get\_acceptance\_capability\_group( MxicAccCapa \*self, E\_IM\_MXIC\_WR\_ARBITER wrArbiter,

E\_IM\_MXIC\_SPEC\_ARBITER\_GR arbiter, E\_IM\_MXIC\_PORT\_GR port, UCHAR\* capability );

4、mxicutlis.h

INT32 Im\_MXIC\_Get\_Acceptance\_Capability( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wrArbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, UCHAR\* capability);改为：

INT32 mxic\_utlis\_get\_acceptance\_capability( MxicUtlis \*self, E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wrArbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, UCHAR\* capability );

5、mxicslot.h

因参数太多，将原有的参数中的3个整合进了一个结构体SlotArgs。

INT32 Im\_MXIC\_Set\_Slot\_Priority( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT\_LEVEL\_CTRL level\_number, T\_IM\_MXIC\_SLOT\_PRIORITY\_LEVEL const\* const priority ); 改为：

INT32 mxic\_slot\_set\_slot\_priority( MxicSlot \*self, SlotArgs priArgs, E\_IM\_MXIC\_PORT port,

E\_IM\_MXIC\_SLOT\_LEVEL\_CTRL levelNumber, const T\_IM\_MXIC\_SLOT\_PRIORITY\_LEVEL\* const priority )

INT32 Im\_MXIC\_Get\_Slot\_Priority( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT\_LEVEL\_CTRL level\_number, T\_IM\_MXIC\_SLOT\_PRIORITY\_LEVEL\* const priority );改为：

INT32 mxic\_slot\_get\_slot\_priority( MxicSlot \*self, SlotArgs priArgs, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT\_LEVEL\_CTRL levelNumber, T\_IM\_MXIC\_SLOT\_PRIORITY\_LEVEL\* const priority );

INT32 Im\_MXIC\_Set\_Slot\_Priority\_All\_Arbiter( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_SLOT\_PRIORITY\_LEVEL const\* const all\_priority );改为：

INT32 mxic\_slot\_set\_slot\_priority\_all\_arbiter( MxicSlot \*self, E\_IM\_MXIC\_UNIT unit,

const T\_IM\_MXIC\_ALL\_SLOT\_PRIORITY\_LEVEL\* const allPriority );

INT32 Im\_MXIC\_Get\_Slot\_Priority\_All\_Arbiter( E\_IM\_MXIC\_UNIT unit, T\_IM\_MXIC\_ALL\_SLOT\_PRIORITY\_LEVEL\* const all\_priority );改为：

INT32 mxic\_slot\_get\_slot\_priority\_all\_arbiter( MxicSlot \*self, E\_IM\_MXIC\_UNIT unit,

T\_IM\_MXIC\_ALL\_SLOT\_PRIORITY\_LEVEL\* const allPriority );

INT32 Im\_MXIC\_Set\_Slot\_Mask( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT slot, UCHAR mask\_enable );改为：

INT32 mxic\_slot\_set\_slot\_mask( MxicSlot \*self, SlotArgs maskArgs, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT slot, UCHAR maskEnable );

INT32 Im\_MXIC\_Get\_Slot\_Mask( E\_IM\_MXIC\_UNIT unit, E\_IM\_MXIC\_WR\_ARBITER wr\_arbiter, E\_IM\_MXIC\_SPEC\_ARBITER arbiter, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT slot, UCHAR\* const mask\_enable );改为：

INT32 mxic\_slot\_get\_slot\_mask( MxicSlot \*self, SlotArgs maskArgs, E\_IM\_MXIC\_PORT port, E\_IM\_MXIC\_SLOT slot, UCHAR\* const maskEnable );

6、mxicmaster.h

INT32 Im\_MXIC\_Set\_Master\_Priority( E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR priority );改为：

INT32 mxic\_master\_set\_master\_priority( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR priority );

INT32 Im\_MXIC\_Get\_Master\_Priority( E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR\* const priority );改为：

INT32 mxic\_master\_get\_master\_priority( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR\* const priority );

INT32 Im\_MXIC\_Set\_Master\_Mask( E\_IM\_MXIC\_UNIT unit, UCHAR master, T\_IM\_MXIC\_MASTER\_MASK const\* const mask );改为：

INT32 mxic\_master\_set\_master\_mask( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, T\_IM\_MXIC\_MASTER\_MASK const\* const mask );

INT32 Im\_MXIC\_Get\_Master\_Mask( E\_IM\_MXIC\_UNIT unit, UCHAR master, T\_IM\_MXIC\_MASTER\_MASK\* const mask );改为：

INT32 mxic\_master\_get\_master\_mask( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, T\_IM\_MXIC\_MASTER\_MASK\* const mask );

INT32 Im\_MXIC\_Set\_Master\_Mask\_Factor( E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR enable );改为：

INT32 mxic\_master\_set\_master\_mask\_factor( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR enable );

INT32 Im\_MXIC\_Get\_Master\_Mask\_Factor( E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR\* const enable );改为：

INT32 mxic\_master\_get\_master\_mask\_factor( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR\* const enable );

INT32 Im\_MXIC\_Set\_Master\_Mask\_Target( E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR enable );改为：

INT32 mxic\_master\_set\_master\_mask\_target( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR enable );

INT32 Im\_MXIC\_Get\_Master\_Mask\_Target( E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR\* const enable );改为：

INT32 mxic\_master\_get\_master\_mask\_target( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, E\_IM\_MXIC\_MASK\_GROUP group, UCHAR\* const enable );

INT32 Im\_MXIC\_Set\_Master\_Mask\_Select( E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR select );改为：

INT32 mxic\_master\_set\_master\_mask\_select( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR select );

INT32 Im\_MXIC\_Get\_Master\_Mask\_Select( E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR\* const select );改为：

INT32 mxic\_master\_get\_master\_mask\_select( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR\* const select );

INT32 Im\_MXIC\_Set\_Master\_Transaction( E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR transaction );改为：

INT32 mxic\_master\_set\_master\_transaction( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR transaction );

INT32 Im\_MXIC\_Get\_Master\_Transaction( E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR\* const transaction );改为：

INT32 mxic\_master\_get\_master\_transaction( MxicMaster \*self, E\_IM\_MXIC\_UNIT unit, UCHAR master, UCHAR\* const transaction );