设备2组杨蕙铭拆分情况

一，文件拆分

（1）

im ltm rbk.c拆分为imltmrbk.c和imltmrbk1.c；

im ltm.h拆分为imltm.h和imltm1.h；

（2）

im sdramc.c拆分为imsdramc.c和imsdramc1.c；

im sdramc.h拆分为imsdramc.h和imsdramc1.h；

（3）

im pro b2b.c拆分为improb2b.c和improb2b1.c；

im pro.h拆分为impro.h和impro1.h。

二，公有方法变更

（1）imltmrbk.c文件

INT32 Im\_LTM\_RBK\_SW\_Reset( UCHAR pipe\_no )

修改为：

INT32 im\_ltm\_rbk\_sw\_reset( UCHAR pipe\_no )

INT32 Im\_LTM\_RBK\_Ctrl\_Axi( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_AXI\* const ctrl\_axi )

修改为：

INT32 im\_ltm\_rbk\_ctrl\_axi( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_AXI\* const ctrl\_axi )

INT32 Im\_LTM\_RBK\_Get\_AxiReadStat( UCHAR pipe\_no, T\_IM\_LTM\_RBK\_AXI\_RCH\_STAT\* const axi\_read\_stat )

修改为：

INT32 im\_ltm\_rbk\_get\_axireadstat( UCHAR pipe\_no, T\_IM\_LTM\_RBK\_AXI\_RCH\_STAT\* const axi\_read\_stat )

INT32 Im\_LTM\_RBK\_Get\_AxiWriteStat( UCHAR pipe\_no, T\_IM\_LTM\_RBK\_AXI\_WCH\_STAT\* const axi\_write\_stat )

修改为：

INT32 im\_ltm\_rbk\_get\_axiwritestat( UCHAR pipe\_no, T\_IM\_LTM\_RBK\_AXI\_WCH\_STAT\* const axi\_write\_stat )

INT32 Im\_LTM\_RBK\_Get\_AxiCtrlParam( UCHAR pipe\_no, T\_IM\_LTM\_RBK\_AXI\_CTRL\_PARAM\* const axi\_ctrl\_prm )

修改为：

INT32 im\_ltm\_rbk\_get\_axictrlparam( UCHAR pipe\_no, T\_IM\_LTM\_RBK\_AXI\_CTRL\_PARAM\* const axi\_ctrl\_prm )

INT32 Im\_LTM\_RBK\_Set\_AxiCtrlParam( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_AXI\_CTRL\_PARAM\* const axi\_ctrl\_prm )

修改为：

INT32 im\_ltm\_rbk\_set\_axictrlparam( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_AXI\_CTRL\_PARAM\* const axi\_ctrl\_prm )

INT32 Im\_LTM\_RBK\_Ctrl\_Common( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_COMMON\* const rbk\_ctrl )

修改为：

INT32 im\_ltm\_rbk\_ctrl\_common( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_COMMON\* const rbk\_ctrl )

INT32 Im\_LTM\_RBK\_Ctrl\_ModeSDRAMInput( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_SDRAM\_INPUT\* const rbk\_ctrl\_sdram\_input )

修改为：

INT32 im\_ltm\_rbk\_ctrl\_modesdraminput( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_SDRAM\_INPUT\* const rbk\_ctrl\_sdram\_input )

INT32 Im\_LTM\_RBK\_Set\_InAddr\_Info( UCHAR pipe\_no, const T\_IM\_LTM\_INADDR\_INFO\* const in\_addr )

修改为：

INT32 im\_ltm\_rbk\_set\_inaddr\_info( UCHAR pipe\_no, const T\_IM\_LTM\_INADDR\_INFO\* const in\_addr )

INT32 Im\_LTM\_RBK\_Set\_OutData\_Info( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_OUTDATA\_INFO\* const out\_data\_info )

修改为：

INT32 im\_ltm\_rbk\_set\_outdata\_info( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_OUTDATA\_INFO\* const out\_data\_info )

INT32 Im\_LTM\_RBK\_Ctrl\_ModeSRODirect( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_SRO\_DIRECT\* const rbk\_ctrl\_sro\_direct )

修改为：

INT32 im\_ltm\_rbk\_ctrl\_modesrodirect( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_CTRL\_SRO\_DIRECT\* const rbk\_ctrl\_sro\_direct )

INT32 Im\_LTM\_RBK\_Ctrl\_Resolution\_Conversion( UCHAR pipe\_no, const UCHAR conversion\_on\_off )

修改为：

INT32 im\_ltm\_rbk\_ctrl\_resolution\_conversion( UCHAR pipe\_no, const UCHAR conversion\_on\_off )

INT32 Im\_LTM\_RBK\_Set\_FrameStop( UCHAR pipe\_no, UCHAR onoff )

修改为：

INT32 im\_ltm\_rbk\_set\_framestop( UCHAR pipe\_no, UCHAR onoff )

INT32 Im\_LTM\_RBK\_Set\_ParamHold( UCHAR pipe\_no, const UCHAR hold\_enable )

修改为：

INT32 im\_ltm\_rbk\_set\_paramhold( UCHAR pipe\_no, const UCHAR hold\_enable )

INT32 Im\_LTM\_RBK\_Get\_RbkBusy( UCHAR pipe\_no, BOOL\* const busy\_status )

修改为：

INT32 im\_ltm\_rbk\_get\_rbkbusy( UCHAR pipe\_no, BOOL\* const busy\_status )

INT32 Im\_LTM\_RBK\_ContStart( UCHAR pipe\_no )

修改为：

INT32 im\_ltm\_rbk\_contstart( UCHAR pipe\_no )

INT32 Im\_LTM\_RBK\_Stop( UCHAR pipe\_no )

修改为：

INT32 im\_ltm\_rbk\_stop( UCHAR pipe\_no )

INT32 Im\_LTM\_RBK\_WaitEnd( DDIM\_USER\_FLGPTN\* const p\_flgptn, DDIM\_USER\_FLGPTN waiptn, DDIM\_USER\_TMO tmout )

修改为：

INT32 im\_ltm\_rbk\_waitend( DDIM\_USER\_FLGPTN\* const p\_flgptn, DDIM\_USER\_FLGPTN waiptn, DDIM\_USER\_TMO tmout )

（2）imltmrbk1.c文件

INT32 Im\_LTM\_RBK\_Ctrl\_RGBTrimming( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_RGB\_TRIMMING\* const rbk\_trimming )

修改为：

INT32 im\_ltm\_rbk\_ctrl\_rgbtrimming( UCHAR pipe\_no, const T\_IM\_LTM\_RBK\_RGB\_TRIMMING\* const rbk\_trimming )

VOID Im\_LTM\_RBK\_Int\_Handler( UCHAR pipe\_no )

修改为：

VOID im\_ltm\_rbk\_int\_handler( UCHAR pipe\_no )

INT32 Im\_LTM\_RBK\_Start( UCHAR pipe\_no )

修改为：

INT32 im\_ltm\_rbk\_start( UCHAR pipe\_no )

INT32 Im\_LTM\_RBK\_Init( UCHAR pipe\_no )

修改为：

INT32 im\_ltm\_rbk\_init( UCHAR pipe\_no )

（3）improb2b.c文件

INT32 Im\_PRO\_ELF\_Ctrl\_NSL( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_NSL\_CTRL\* nsl\_ctrl )

修改为：

INT32 im\_pro\_elf\_ctrl\_nsl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_NSL\_CTRL\* nsl\_ctrl )

INT32 Im\_PRO\_ELF\_Set\_A0\_PAEN( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

修改为：

INT32 im\_pro\_elf\_set\_a0\_paen( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

INT32 Im\_PRO\_ELF\_Set\_NSL\_PAEN( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

修改为：

INT32 im\_pro\_elf\_set\_nsl\_paen( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

INT32 Im\_PRO\_Get\_RdmaAddr\_ELF\_Cntl( E\_IM\_PRO\_UNIT\_NUM unit\_no, const T\_IM\_PRO\_RDMA\_ELF\_ADDR\*\* addr )

修改为：

INT32 im\_pro\_get\_rdmaaddr\_elf\_cntl( E\_IM\_PRO\_UNIT\_NUM unit\_no, const T\_IM\_PRO\_RDMA\_ELF\_ADDR\*\* addr )

INT32 Im\_PRO\_ELF\_Set\_Edge\_Adj( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_HIGH\_FREQ\_AREA\_EDGE\_ADJ\* edge\_adj )

修改为：

INT32 im\_pro\_elf\_set\_edge\_adj( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_HIGH\_FREQ\_AREA\_EDGE\_ADJ\* edge\_adj )

INT32 Im\_PRO\_FSHD\_Start( E\_IM\_PRO\_UNIT\_NUM unit\_no )

修改为：

INT32 im\_pro\_fshd\_start( E\_IM\_PRO\_UNIT\_NUM unit\_no )

INT32 Im\_PRO\_FSHD\_Stop( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR force )

修改为：

INT32 im\_pro\_fshd\_stop( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR force )

INT32 Im\_PRO\_FSHD\_Ctrl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_COMMON\_CTRL\* fshd\_ctrl )

修改为：

INT32 im\_pro\_fshd\_ctrl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_COMMON\_CTRL\* fshd\_ctrl )

INT32 Im\_PRO\_FSHD\_Set\_PAEN( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

修改为：

INT32 im\_pro\_fshd\_set\_paen( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

INT32 Im\_PRO\_FSHD\_Set\_Area( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_AREA\_INFO\* fshd\_area )

修改为：

INT32 im\_pro\_fshd\_set\_area( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_AREA\_INFO\* fshd\_area )

INT32 Im\_PRO\_FSHD\_Set\_Blend( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_BLEND\_CTRL\* blend\_ctrl )

修改为：

INT32 im\_pro\_fshd\_set\_blend( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_BLEND\_CTRL\* blend\_ctrl )

INT32 Im\_PRO\_FSHD\_Set\_Frame( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_FRAME\_CTRL\* fshd\_ctrl )

修改为：

INT32 im\_pro\_fshd\_set\_frame( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_FRAME\_CTRL\* fshd\_ctrl )

INT32 Im\_PRO\_FSHD\_Set\_Concentric( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_CONCENTRIC\_CTRL\* fshd\_ctrl )

修改为：

INT32 im\_pro\_fshd\_set\_concentric( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_FSHD\_CONCENTRIC\_CTRL\* fshd\_ctrl )

E\_IM\_PRO\_FSHD\_LUT\_SEL Im\_PRO\_FSHD\_Get\_Ctbl\_Number( E\_IM\_PRO\_UNIT\_NUM unit\_no )

修改为：

E\_IM\_PRO\_FSHD\_LUT\_SEL im\_pro\_fshd\_get\_ctbl\_number( E\_IM\_PRO\_UNIT\_NUM unit\_no )

INT32 Im\_PRO\_FSHD\_Set\_Table( E\_IM\_PRO\_UNIT\_NUM unit\_no, E\_IM\_PRO\_FSHD\_LUT\_SEL tbl\_sel, T\_IM\_PRO\_FSHD\_TABLE\* shd\_tbl )

修改为：

INT32 im\_pro\_fshd\_set\_table( E\_IM\_PRO\_UNIT\_NUM unit\_no, E\_IM\_PRO\_FSHD\_LUT\_SEL tbl\_sel, T\_IM\_PRO\_FSHD\_TABLE\* shd\_tbl )

INT32 Im\_PRO\_FSHD\_Set\_Coeff\_Read\_Line\_Wait( E\_IM\_PRO\_UNIT\_NUM unit\_no, USHORT req\_interval\_clk )

修改为：

INT32 im\_pro\_fshd\_set\_coeff\_read\_line\_wait( E\_IM\_PRO\_UNIT\_NUM unit\_no, USHORT req\_interval\_clk )

INT32 Im\_PRO\_Get\_RdmaAddr\_FSHD\_Cntl( E\_IM\_PRO\_UNIT\_NUM unit\_no, const T\_IM\_PRO\_RDMA\_FSHD\_ADDR\*\* addr )

修改为：

INT32 im\_pro\_get\_rdmaaddr\_fshd\_cntl( E\_IM\_PRO\_UNIT\_NUM unit\_no, const T\_IM\_PRO\_RDMA\_FSHD\_ADDR\*\* addr )

INT32 Im\_PRO\_Get\_RdmaAddr\_FSHD\_Ctbl( E\_IM\_PRO\_UNIT\_NUM unit\_no, E\_IM\_PRO\_FSHD\_LUT\_SEL tbl\_sel, const T\_IM\_PRO\_RDMA\_FSHD\_CTBL\_ADDR\*\* addr )

修改为：

INT32 im\_pro\_get\_rdmaaddr\_fshd\_ctbl( E\_IM\_PRO\_UNIT\_NUM unit\_no, E\_IM\_PRO\_FSHD\_LUT\_SEL tbl\_sel, const T\_IM\_PRO\_RDMA\_FSHD\_CTBL\_ADDR\*\* addr )

VOID Im\_PRO\_ELF\_Stop\_NSL( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR force )

修改为：

VOID im\_pro\_elf\_stop\_nsl( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR force )

INT32 Im\_PRO\_ELF\_Start\_NSL( E\_IM\_PRO\_UNIT\_NUM unit\_no )

修改为：

INT32 im\_pro\_elf\_start\_nsl( E\_IM\_PRO\_UNIT\_NUM unit\_no )

VOID Im\_PRO\_ELF\_Stop( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR force )

修改为：

VOID im\_pro\_elf\_stop( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR force )

INT32 Im\_PRO\_ELF\_Start( E\_IM\_PRO\_UNIT\_NUM unit\_no )

修改为：

INT32 im\_pro\_elf\_start( E\_IM\_PRO\_UNIT\_NUM unit\_no )

（4）improb2b1.c文件

VOID ( E\_IM\_PRO\_UNIT\_NUM unit\_no )

修改为：

VOID im\_pro\_b2btop\_init( E\_IM\_PRO\_UNIT\_NUM unit\_no )

INT32 Im\_PRO\_B2BTOP\_SW\_Reset( E\_IM\_PRO\_UNIT\_NUM unit\_no )

修改为：

INT32 im\_pro\_b2btop\_sw\_reset( E\_IM\_PRO\_UNIT\_NUM unit\_no )

INT32 Im\_PRO\_B2BTOP\_Control\_Clock( E\_IM\_PRO\_UNIT\_NUM unit\_no, E\_IM\_PRO\_B2BTOP\_CLK\_TYPE clk\_type, UCHAR on\_off, UCHAR wait\_skip )

修改为：

INT32 im\_pro\_b2btop\_control\_clock( E\_IM\_PRO\_UNIT\_NUM unit\_no, E\_IM\_PRO\_B2BTOP\_CLK\_TYPE clk\_type, UCHAR on\_off, UCHAR wait\_skip )

INT32 Im\_PRO\_B2BTOP\_Ctrl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_B2BTOP\_CTRL\* ctrl )

修改为：

INT32 im\_pro\_b2btop\_ctrl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_B2BTOP\_CTRL\* ctrl )

INT32 Im\_PRO\_ELF\_Set\_PAEN( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

修改为：

INT32 im\_pro\_elf\_set\_paen( E\_IM\_PRO\_UNIT\_NUM unit\_no, UCHAR paen\_trg )

INT32 Im\_PRO\_ELF\_Ctrl\_Common( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_COMMON\_CTRL\* elf\_common\_ctrl )

修改为：

INT32 im\_pro\_elf\_ctrl\_common( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_COMMON\_CTRL\* elf\_common\_ctrl )

INT32 Im\_PRO\_ELF\_Ctrl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_CTRL\* elf\_ctrl )

修改为：

INT32 im\_pro\_elf\_ctrl( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_CTRL\* elf\_ctrl )

INT32 Im\_PRO\_ELF\_Set\_Noisefunc( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_NOISE\_FUNC\* nfunc )

修改为：

INT32 im\_pro\_elf\_set\_noisefunc( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_NOISE\_FUNC\* nfunc )

INT32 Im\_PRO\_ELF\_Set\_LGT\_EDG\_Filter( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_LGT\_EDG\_FUNC\* lgt\_filter )

修改为：

INT32 im\_pro\_elf\_set\_lgt\_edg\_filter( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_LGT\_EDG\_FUNC\* lgt\_filter )

INT32 Im\_PRO\_ELF\_Set\_Area( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_AREA\* elf\_area )

修改为：

INT32 im\_pro\_elf\_set\_area( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_AREA\* elf\_area )

INT32 Im\_PRO\_ELF\_Set\_E\_L\_Filter( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_L\_E\_FUNC\* e\_l\_filter )

修改为：

INT32 im\_pro\_elf\_set\_e\_l\_filter( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_L\_E\_FUNC\* e\_l\_filter )

INT32 Im\_PRO\_ELF\_Set\_F9\_Filter( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_f9\_SAMPLING\* f9\_filter )

修改为：

INT32 im\_pro\_elf\_set\_f9\_filter( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_f9\_SAMPLING\* f9\_filter )

INT32 Im\_PRO\_ELF\_Set\_F9\_RGBGain( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_f9\_RGB\_GAIN\* rgb\_gain )

修改为：

INT32 im\_pro\_elf\_set\_f9\_rgbgain( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_f9\_RGB\_GAIN\* rgb\_gain )

INT32 Im\_PRO\_ELF\_Set\_1s5\_Sampling( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_1s5\_SAMPLING\* s1s5\_sampling )

修改为：

INT32 im\_pro\_elf\_set\_1s5\_sampling( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_1s5\_SAMPLING\* s1s5\_sampling )

INT32 Im\_PRO\_ELF\_Set\_5s9\_Sampling( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_5s9\_SAMPLING\* s5s9\_sampling )

修改为：

INT32 im\_pro\_elf\_set\_5s9\_sampling( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_5s9\_SAMPLING\* s5s9\_sampling )

INT32 Im\_PRO\_ELF\_Set\_Noise\_Adj( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_Noise\_Adj\* noise\_adj\_filter )

修改为：

INT32 im\_pro\_elf\_set\_noise\_adj( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_Noise\_Adj\* noise\_adj\_filter )

INT32 Im\_PRO\_ELF\_Set\_A0\_Sampling( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_a0\_FILTER\* a0\_filter )

修改为：

INT32 im\_pro\_elf\_set\_a0\_sampling( E\_IM\_PRO\_UNIT\_NUM unit\_no, T\_IM\_PRO\_ELF\_a0\_FILTER\* a0\_filter )

（5）imsdramc.c

INT32 Im\_SDRAMC\_Set\_Clock\_Enable\_DataBuf( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_DATABUF dbno, E\_IM\_SDRAMC\_EN enable )

修改为：

INT32 im\_sdramc\_set\_clock\_enable\_databuf( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_DATABUF dbno, E\_IM\_SDRAMC\_EN enable )

INT32 Im\_SDRAMC\_Strat\_SelfRefresh( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_strat\_selfrefresh( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_Stop\_SelfRefresh( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_stop\_selfrefresh( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_Start\_ManualMode( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_start\_manualmode( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_Stop\_ManualMode( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_stop\_manualmode( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_ReLock\_DLL( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_relock\_dll( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_Start\_Training( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_TRAINING\_MODE mode )

修改为：

INT32 im\_sdramc\_start\_training( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_TRAINING\_MODE mode )

INT32 Im\_SDRAMC\_Start\_Access\_Detection( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_AREA area, const T\_IM\_SDRAMC\_ACCESS\_DETECTION\* const acs\_det )

修改为：

INT32 im\_sdramc\_start\_access\_detection( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_AREA area, const T\_IM\_SDRAMC\_ACCESS\_DETECTION\* const acs\_det )

INT32 Im\_SDRAMC\_Stop\_Access\_Detection( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_AREA area )

修改为：

INT32 im\_sdramc\_stop\_access\_detection( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_AREA area )

INT32 Im\_SDRAMC\_Get\_Access\_Detection( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_AREA \*area, E\_IM\_SDRAMC\_PORT\* port, ULONG\* const addr )

修改为：

INT32 im\_sdramc\_get\_access\_detection( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_AREA \*area, E\_IM\_SDRAMC\_PORT\* port, ULONG\* const addr )

INT32 Im\_SDRAMC\_Start\_Monitor( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_MONMODE mode, BOOL write\_access, BOOL read\_access, BOOL auto\_clear )

修改为：

INT32 im\_sdramc\_start\_monitor( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_MONMODE mode, BOOL write\_access, BOOL read\_access, BOOL auto\_clear )

INT32 Im\_SDRAMC\_Stop\_Monitor( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_stop\_monitor( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_Get\_Monitor( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_MONITOR\* const mon )

修改为：

INT32 im\_sdramc\_get\_monitor( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_MONITOR\* const mon )

INT32 Im\_SDRAMC\_Start\_Monitor\_Mxic( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_MONMODE mode, BOOL write\_access, BOOL read\_access, BOOL auto\_clear )

修改为：

INT32 im\_sdramc\_start\_monitor\_mxic( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_MONMODE mode, BOOL write\_access, BOOL read\_access, BOOL auto\_clear )

INT32 Im\_SDRAMC\_Stop\_Monitor\_Mxic( E\_IM\_SDRAMC\_CH ch )

修改为：

INT32 im\_sdramc\_stop\_monitor\_mxic( E\_IM\_SDRAMC\_CH ch )

INT32 Im\_SDRAMC\_Get\_Monitor\_Mxic( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_MONITOR\_MXIC\* const mon )

修改为：

INT32 im\_sdramc\_get\_monitor\_mxic( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_MONITOR\_MXIC\* const mon )

VOID Im\_SDRAMC\_Int\_Handler( E\_IM\_SDRAMC\_CH ch )

修改为：

VOID im\_sdramc\_int\_handler( E\_IM\_SDRAMC\_CH ch )

（5）imsdramc1.c

INT32 Im\_SDRAMC\_Set\_AcceptanceCapability( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, const UCHAR rnum, const UCHAR wnum )

修改为：

INT32 im\_sdramc\_set\_acceptancecapability( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, const UCHAR rnum, const UCHAR wnum )

INT32 Im\_SDRAMC\_Get\_AcceptanceCapability( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, UCHAR\* const rnum, UCHAR\* const wnum )

修改为：

INT32 im\_sdramc\_get\_acceptancecapability( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, UCHAR\* const rnum, UCHAR\* const wnum )

INT32 Im\_SDRAMC\_Set\_BurstLimit( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, const E\_IM\_SDRAMC\_BURST\_LIMIT rsize, const E\_IM\_SDRAMC\_BURST\_LIMIT wsize )

修改为：

INT32 im\_sdramc\_set\_burstlimit( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, const E\_IM\_SDRAMC\_BURST\_LIMIT rsize, const E\_IM\_SDRAMC\_BURST\_LIMIT wsize )

INT32 Im\_SDRAMC\_Get\_BurstLimit( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, E\_IM\_SDRAMC\_BURST\_LIMIT\* const rsize, E\_IM\_SDRAMC\_BURST\_LIMIT\* const wsize )

修改为：

INT32 im\_sdramc\_get\_burstlimit( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, E\_IM\_SDRAMC\_BURST\_LIMIT\* const rsize, E\_IM\_SDRAMC\_BURST\_LIMIT\* const wsize )

INT32 Im\_SDRAMC\_Set\_RemapAddr( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, const LONG addr )

修改为：

INT32 im\_sdramc\_set\_remapaddr( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, const LONG addr )

INT32 Im\_SDRAMC\_Get\_RemapAddr( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, LONG\* const addr )

修改为：

INT32 im\_sdramc\_get\_remapaddr( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, LONG\* const addr )

INT32 Im\_SDRAMC\_Set\_Ch\_Select( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port)

修改为：

INT32 im\_sdramc\_set\_ch\_select( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port)

INT32 Im\_SDRAMC\_Get\_Ch\_Select( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, E\_IM\_SDRAMC\_CH\* chsel )

修改为：

INT32 im\_sdramc\_get\_ch\_select( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, E\_IM\_SDRAMC\_CH\* chsel )

INT32 Im\_SDRAMC\_Set\_Cmd\_Arbitration\_Rate( E\_IM\_SDRAMC\_CH ch, const T\_IM\_SDRAMC\_CMD\_ARB\_RATE\* const cmd\_arb )

修改为：

INT32 im\_sdramc\_set\_cmd\_arbitration\_rate( E\_IM\_SDRAMC\_CH ch, const T\_IM\_SDRAMC\_CMD\_ARB\_RATE\* const cmd\_arb )

INT32 Im\_SDRAMC\_Get\_Cmd\_Arbitration\_Rate( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_CMD\_ARB\_RATE\* const cmd\_arb )

修改为：

INT32 im\_sdramc\_get\_cmd\_arbitration\_rate( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_CMD\_ARB\_RATE\* const cmd\_arb )

INT32 Im\_SDRAMC\_Set\_Cmd\_Arbitration\_Priority( E\_IM\_SDRAMC\_CH ch, const T\_IM\_SDRAMC\_CMD\_ARB\_PRI\* const cmd\_arb )

修改为：

INT32 im\_sdramc\_set\_cmd\_arbitration\_priority( E\_IM\_SDRAMC\_CH ch, const T\_IM\_SDRAMC\_CMD\_ARB\_PRI\* const cmd\_arb )

INT32 Im\_SDRAMC\_Get\_Cmd\_Arbitration\_Priority( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_CMD\_ARB\_PRI\* const cmd\_arb )

修改为：

INT32 im\_sdramc\_get\_cmd\_arbitration\_priority( E\_IM\_SDRAMC\_CH ch, T\_IM\_SDRAMC\_CMD\_ARB\_PRI\* const cmd\_arb )

INT32 Im\_SDRAMC\_Set\_Mode\_Register( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_RANK rank, const UCHAR ma, const UCHAR data )

修改为：

INT32 im\_sdramc\_set\_mode\_register( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_RANK rank, const UCHAR ma, const UCHAR data )

INT32 Im\_SDRAMC\_Get\_Mode\_Register( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_RANK rank, const UCHAR ma, UCHAR\* const data )

修改为：

INT32 im\_sdeamc\_get\_mode\_register( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_RANK rank, const UCHAR ma, UCHAR\* const data )

INT32 Im\_SDRAMC\_Set\_Clock\_Enable\_BusIf( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, E\_IM\_SDRAMC\_ACCESS access, E\_IM\_SDRAMC\_EN enable )

修改为：

INT32 im\_sdramc\_set\_clock\_enable\_busIf( E\_IM\_SDRAMC\_CH ch, E\_IM\_SDRAMC\_PORT port, E\_IM\_SDRAMC\_ACCESS access, E\_IM\_SDRAMC\_EN enable )

二，私有方法变更

（1）imltmrbk.c文件

static UINT32 im\_ltm\_rbk\_calc\_out\_width( UCHAR pipe\_no )

修改为：

static UINT32 imLtmRbkCalcOutWidth( UCHAR pipe\_no )

static UINT32 im\_ltm\_rbk\_calc\_out\_lines( UCHAR pipe\_no )

修改为：

static UINT32 imLtmRbkCalcOutLines( UCHAR pipe\_no )

static VOID im\_ltm\_rbk\_get\_loop\_val( UCHAR pipe\_no, UCHAR\* loop\_sta, UCHAR\* loop\_end )

修改为：

static VOID imLtmRbkGetLoopVal( UCHAR pipe\_no, UCHAR\* loop\_sta, UCHAR\* loop\_end )

（2）imltmrbk1.c文件

static VOID im\_ltm\_rbk\_manager\_init( UCHAR pipe\_no )

修改为：

static VOID imLtmRbkManagerInit( UCHAR pipe\_no )