

一、MI_VPE

mi_vpe.h 33x 新增 API

```
MI_S32 MI_VPE_LDCBegViewConfig(MI_VPE_CHANNEL VpeCh);
MI_S32 MI_VPE_LDCEndViewConfig(MI_VPE_CHANNEL VpeCh);
MI_S32 MI_VPE_LDCSetViewConfig(MI_VPE_CHANNEL VpeCh, void *pConfigAddr, MI_U32
u32Size);
```

设置 VPE Ldc pass bin cfg

mi_vpe_datatype.h

(1) 33X 移除 如下定义

```
#define MI_VPE_MAX_CHANNEL_NUM (2)
#define MI_VPE_REALTIMEMODE_MAXCHNL_NUM (1)
#define MI_VPE_MAX_PORT_NUM (4)
#define MI_VPE_MAX_WORKINGLIST_NODE (2)
#define MI_VPE_CHANNEL_MAX_WIDTH (2560)
#define MI_VPE_CHANNEL_MAX_HEIGHT (1440)
```

上面定义会随着 chip 变化，改到定义在 MI 内部。

目前应该没有客户 demo 用到上面的定义。

(2) 新增 struct, 参数

typedef enum

```
{
    E_MI_VPE_ZOOM_LDC_NULL,
    E_MI_VPE_ZOOM_LDC_PORT0 = 0X01,
    E_MI_VPE_ZOOM_LDC_PORT1 = 0X02,
    E_MI_VPE_ZOOM_LDC_PORT2 = 0X04,
    E_MI_VPE_ZOOM_LDC_MAX = E_MI_VPE_ZOOM_LDC_PORT0 |
    E_MI_VPE_ZOOM_LDC_PORT1 | E_MI_VPE_ZOOM_LDC_PORT2,
}MI_VPE_ChnPortMode_e;
```

typedef struct MI_VPE_ChannelAttr_s

```
{
    MI_U16 u16MaxW;
    MI_U16 u16MaxH;
    MI_SYS_PixelFormat_e ePixFmt;
    MI_VPE_HDRTyp_e eHDRTyp;
```

```

MI_VPE_SensorChannel_e eSensorBindId;

MI_BOOL bNrEn;
MI_BOOL bEdgeEn;
MI_BOOL bEsEn;
MI_BOOL bContrastEn;
MI_BOOL bUvInvert;
MI_BOOL bRotation;
MI_VPE_RunningMode_e eRunningMode;
MI_VPE_IsplnitPara_t tIsplnitPara;
MI_BOOL bEnLdc; // true port3 for ldc or for lsp(skip pass1)
MI_U32 u32ChnPortMode; // MI_VPE_ChnPortMode_e 设置 port realtime/frame 属性
}MI_VPE_ChannelAttr_t;

typedef struct MI_VPE_ChannelPara_s
{
    MI_VPE_PqParam_t stPqParam; // only dvr use
    MI_VPE_HDRTType_e eHDRTType;
    MI_VPE_3DNR_Level_e e3DNRLevel;
    MI_BOOL bMirror; // isp mirror
    MI_BOOL bFlip; // isp flip
    MI_BOOL bWdrEn; //Wdr on/off not use
} MI_VPE_ChannelPara_t;

```

二、MI_SYS

```

typedef struct MI_SYS_FrameBufExtraConfig_s
{
    //Buf alighment requirement in horizontal
    MI_U16 u16BufHAlignment;
    //Buf alighment requirement in vertical
    MI_U16 u16BufVAlignment;
    //Buf alighment requirement in UV
    MI_U16 u16BufUVAlignment;
    //Clear Padding flag
    MI_BOOL bClearPadding;
}MI_SYS_FrameBufExtraConfig_t;

```

620D vdec 需要 UV 分量 32 对齐，之前是 hardcode 写在 sys 中，导致其它模块对齐后画面花屏，新开变量让 vdec 设置。

三、MI_VIF/MI_SENSOR/ MI_VENC/ MI_RGN/MI_DIVP 和 32x 保持一致。

四、MI_LDC 新增 Module

五、MI_AI/MI_AO

与 32x 的 API 相比，33x 主要是新增 I2S 的相关设置
添加更多的 mclk 设置

```
typedef enum{
    E_MI_AUDIO_I2S_MCLK_0,                //not enable
    E_MI_AUDIO_I2S_MCLK_12_288M,         //12.288M
    E_MI_AUDIO_I2S_MCLK_16_384M,         //16.384M
    E_MI_AUDIO_I2S_MCLK_18_432M,         //18.432M
    E_MI_AUDIO_I2S_MCLK_24_576M,         //24.576M
    E_MI_AUDIO_I2S_MCLK_24M,             //24MHz
    E_MI_AUDIO_I2S_MCLK_48M,             //48MHz
}MI_AUDIO_I2sMclk_e;
```

添加 I2S Tdm slave mode

```
typedef enum
{
    E_MI_AUDIO_MODE_I2S_MASTER,          /* I2S master mode */
    E_MI_AUDIO_MODE_I2S_SLAVE,           /* I2S slave mode */
    E_MI_AUDIO_MODE_TDM_MASTER,          /* TDM master mode */
    E_MI_AUDIO_MODE_TDM_SLAVE,           /* TDM slave mode */
    E_MI_AUDIO_MODE_MAX,
}MI_AUDIO_Mode_e;
```

五、MI_CIPHER

mi_cipher.h

接口变动：

```
MI_S32 MI_CIPHER_Encrypt(MI_HANDLE handle, void* srcAddr, void* dstAddr , MI_U32  
u32ByteLen, MI_U32 *pu32dstByteLen);
```

```
MI_S32 MI_CIPHER_Decrypt(MI_HANDLE handle, void* srcAddr, void* dstAddr, MI_U32  
u32ByteLen, MI_U32 *pu32dstByteLen);
```

mi_cipher_datatype.h

新增类型：

```
typedef enum {  
    E_MI_CIPHER_KEY_SIZE_128 = 0,  
    E_MI_CIPHER_KEY_SIZE_192,  
    E_MI_CIPHER_KEY_SIZE_256,  
} MI_CIPHER_KeySize_e;
```

结构体变动：

```
typedef struct MI_CIPHER_Config_s  
{  
    MI_CIPHER_KeySize_e eKeySize;  
    MI_U8 key[MI_CIPHER_KEY_SIZE_MAX];  
    MI_U8 iv[MI_CIPHER_AES_BLOCK_SIZE];  
    MI_CIPHER_ALG_e eAlg;  
} MI_CIPHER_Config_t;
```