MEDIATEK

MIPI General Introduction

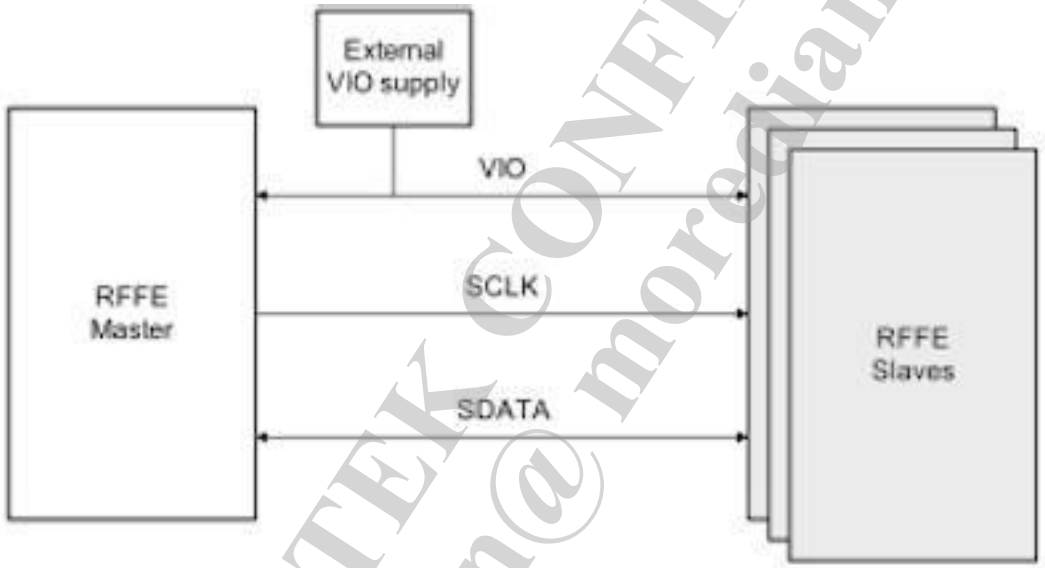
ACS1/RF
20180521

Overview

- RF FE MIPI Basic conception
  - MIPI的物理连接
  - 软件中MIPI控制行为要素
  - MIPI HW PORT BUS
  - MIPI USID
  - MIPI Event/Data
  - MIPI data的波形解析
  - MIPI timing——Unit
  - MIPI timing——One MIPI Data time
  - MIPI timing——RX WIN
  - MIPI timing——TX
  - 93 modem PA on event timing
  - 93 modem ASM/PA on event timing
  - 93 modem TPC event timing check
- MIPI Configuration Guide
  - 2G Part
  - TDSCDMA Part
  - WCDMA Part
  - LTE Part
  - C2K Part
- Reference document

MIPI的物理连接

- 从物理连接上看，MIPI interface只有VIO, SCLK, SDATA三个网络。



- 同一个MIPI port理论上可以最多挂载15个不同USID的器件。

Note: MIPI的data/clock线上挂载的到地电容不要超过20pF!

软件中MIPI控制行为要素

- **MIPI port**

  定义和BB接口的物理连接。

- **USID**

  器件的固有标识，用于识别当前被操控器件。

- **Event**

  定义TRX ON/OFF的时候，需要执行的步骤。

- **Data**

  完成Event定义，所需要执行的MIPI data。一连串的MIPI data组成一个MIPI Event

- **Timing**

  定义每一个Event执行的时间点。

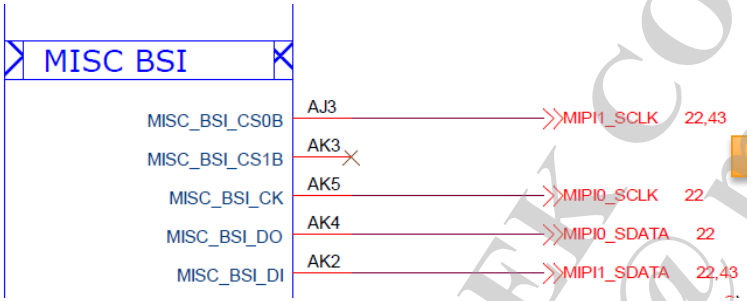
MIPI HW PORT BUS

- 93 Modem提供4个MIPI port ,供给外部MIPI器件使用。

建议Follow reference design MIPI 分配，Default Code 修改比较小

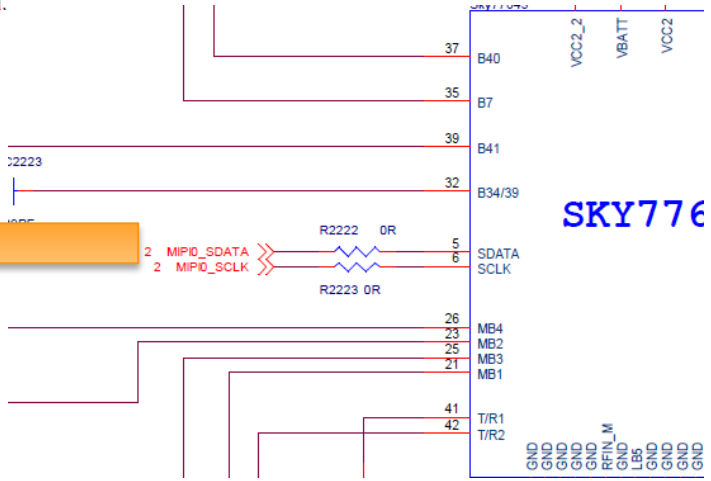
CHINA+EU

port	device	2G	3G	TDS	C2K	LTE	Other
MIPI-0	TXM / eLNA	ASM / PA	ASM	ASM / PA	ASM	ASM	B38/B41 PRx eLNA
MIPI-1	PA1(Sky77651) / PA2(Sky77652) /PA3(QM52042) / PMIC(RF7061)		PA1 / PA2	PA2	PA1 / PA2	PA1 / PA2	PMIC(HPUE)
MIPI-2	DRX ASM / eLNA		ASM	ASM	ASM	ASM	B38/B41 DRx eLNA
MIPI-3	ANT tuner	OLAT (TAS)	OLAT (TAS)	OLAT (TAS)	OLAT (TAS)	OLAT (TAS)	



BB interface

RFFE interface



MIPI USID

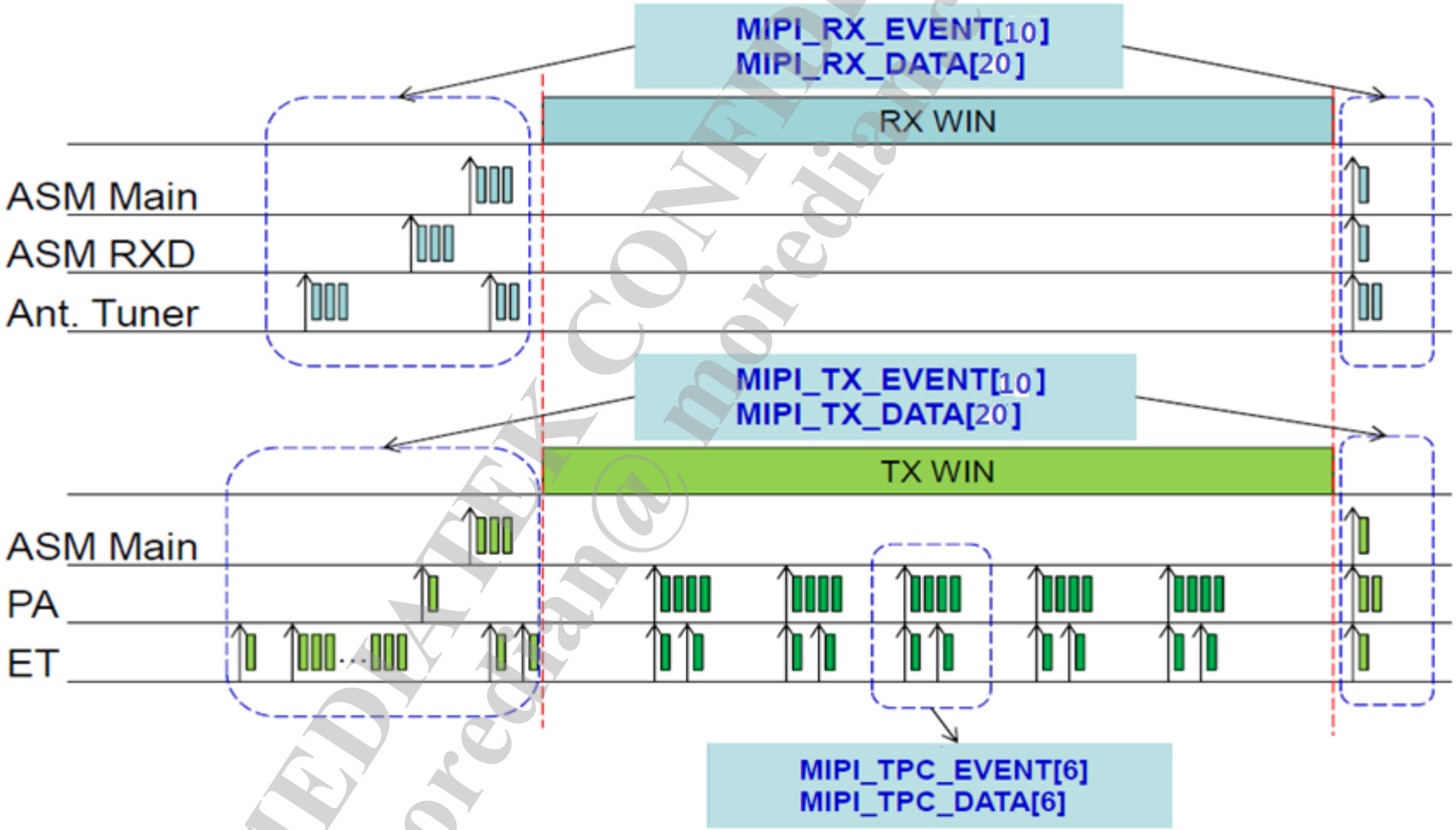
- 通常在器件的datasheet中，即可找到其USID：

Register 30, Address 0x1E (MAN_ID)					
Register 30	**Description**	**Trigger Support**	**R/W**	**Default**	
[7:0]	MANUFACTURER_ID[7:0]	No	R	10100101	Manufacturer ID = 0x1A5
Register 31, Address 0x1F (USID)					
Register 31	**Description**	**Trigger Support**	**R/W**	**Default**	
[7:6]	Reserved	No	R	00	
[5:4]	MANUFACTURER_ID[9:8]		R	01	Manufacturer ID = 0x1A5
[3:0]	USID		R/W	1111	USID = 0xF
Register 32, Address 0x20 (EXT_PRODUCT_ID)					
Register 32	**Description**	**Trigger Support**	**R/W**	**Default**	
[7:0]	EXT_PRODUCT_ID	No	R	00000100	EXT_PRODUCT_ID = 0x04

Note：当同一个MIPI Port上的器件出现USID冲突时，需要在mml1_custom_mipi.c里修改USID定义。

MIPI Event /Data

- 在RF链路上有很多器件，要完成TRX ON/OFF的动作，需要将这些器件一一设置。每个Event由一连串的data组合而成



MIPI Event/Data

- lte_custom_mipi.c Limited number of items in LTE RF MIPI data &event tables
- 93 modem LTE Rx Off event (data num <=8 笔)

```
LTE_MIPI_EVENT_TABLE_T LTE_Band1_MIPI_RX_EVENT_SetDefault[] =
{
   /* No.      elm type      , data idx       , evt_type          , evt_offset
   /*                         { start, stop },                      ( us )
   { /* 0  */ LTE_MIPI_ASM , { 0     , 1    }, LTE_MIPI_TRX_ON    , LTE_FDD_MIPI_ASM_F
   { /* 1  */ LTE_MIPI_ASM , { 2     , 3    }, LTE_MIPI_TRX_OFF   , LTE_FDD_MIPI_ASM_F
   { /* 2  */ LTE_MIPI_NULL, { 0     , 0    }, LTE_MIPI_EVENT_NULL, 0
};
```

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band1_MIPI_RX_DATA_SetDefault[] =
{
   //No.      elm type      , port_sel          , data_seq      , USID                          ,{ {
   { /* 0  */ LTE_MIPI_ASM , LTE_MIPI_PORT0    , LTE_REG_W     , MIPI_USID_ASM0_SetDefault     ,{ {
   { /* 1  */ LTE_MIPI_ASM , LTE_MIPI_PORT2    , LTE_REG_W     , MIPI_USID_ASM1_SetDefault     ,{ {
   { /* 2  */ LTE_MIPI_ASM , LTE_MIPI_PORT0    , LTE_REG_W     , MIPI_USID_ASM0_SetDefault     ,{ {
   { /* 3  */ LTE_MIPI_ASM , LTE_MIPI_PORT2    , LTE_REG_W     , MIPI_USID_ASM1_SetDefault     ,{ {
   { /* 4  */ LTE_MIPI_NULL, 0                 , 0             , 0                             ,{ {
};
```

NOTE: Data number cannot exceed 12 in each event
e.g. start: 2, stop: 15 , 15-2=13 >12 (X)

Case	Max Num.
RX Event	10
RX Data	20
TX Event	10
TX Data	20
ANT Event	5
ANT Data	10
Sub-band Num	10
Sub-band Num per data	5
TPC Event	6
TPC Data	6
TPC section	9
TPC Section Data	5
TAS Event	10
TAS Data	10

```
,{ { subband-0 freq    ,addr ,data }, { subband-1 freq    ,addr ,data }, { subband-2 freq    ,addr ,data }, { subband-3 freq    ,addr ,data }, { subband-4 fre
 9250 /*100 kHz*/ ,0x02 ,0x03 }, {  9337 /*100 kHz*/ ,0x02 ,0x03 }, {  9425 /*100 kHz*/ ,0x02 ,0x03 }, {  9512 /*100 kHz*/ ,0x02 ,0x03 }, {  9600 /*100 kHz*/ ,
 9250 /*100 kHz*/ ,0x01 ,0x03 }, {  9337 /*100 kHz*/ ,0x01 ,0x03 }, {  9425 /*100 kHz*/ ,0x01 ,0x03 }, {  9512 /*100 kHz*/ ,0x01 ,0x03 }, {  9600 /*100 kHz*/ ,
 9250 /*100 kHz*/ ,0x02 ,0x00 }, {  9337 /*100 kHz*/ ,0x02 ,0x00 }, {  9425 /*100 kHz*/ ,0x02 ,0x00 }, {  9512 /*100 kHz*/ ,0x02 ,0x00 }, {  9600 /*100 kHz*/ ,
 9250 /*100 kHz*/ ,0x01 ,0x00 }, {  9337 /*100 kHz*/ ,0x01 ,0x00 }, {  9425 /*100 kHz*/ ,0x01 ,0x00 }, {  9512 /*100 kHz*/ ,0x01 ,0x00 }, {  9600 /*100 kHz*/ ,
 ,{ {     0 /*100 kHz*/ ,0    ,0    }, {     0 /*100 kHz*/ ,0    ,0    }, {     0 /*100 kHz*/ ,0    ,0    }, {     0 /*100 kHz*/ ,0    ,0    }, {     0 /*100 k
```

MIPI Event /Data

- 所需要的MIPI Data从datasheet中可以获得。要仔细辨清该往哪个寄存器里写什么样的data。

Register 0	Description	Trigger Support	R/W	Default	Notes
[7]	Trigger Select	Trigger0	R/W	0	0 = Trigger 0,1,2 or'd together 1 = Trigger 0,1,2 fire independently
[6:3]	PA Band Select Control Mode		R/W	0000	Control Mode 0000 = PA's Disabled 0001 = MB1_TX 0010 = MB2_TX 0011 = MB3_TX 0100 = MB4_TX 0101 = MB5_TX 0110 = HB2_TX 0111 = HB3_TX 1000 = HB1_TX 1001 = HB4_TX 1010 = Don't Care 1011 = Don't Care 1100 = Don't Care 1101 = Don't Care 1110 = Don't Care 1111 = PA's Disabled (High switch isolation)
[2]	PA Enable		R/W	0	PA Enable 0 = PA Off 1 = PA On
[1]	PA Mode		R/W	0	PA Mode 0 = HPM 1 = LPM
[0]	Reserved		R/W	0	Zero required

Sky77652-11 HM MMMB PA

MIPI Event /Data

- 所需要的MIPI Data从datasheet中可以获得。要仔细辨清该往哪个寄存器里写什么样的data。

Sky77928-11

Bit Position	Description	Trigger Support	R/W	Default	Notes		
Register 0, Address 0x00 (PA Control)							
[7]	Reserved	Trigger0	R/W	0	Reserved; (set to 1)		
[6]	Reserved			0	Reserved; (set to zero)		
[5]	Gain Control (linear)			0	0 = nominal gain	1 = reduced gain	
[4:0]	TxFEM Mode Control			00000	0x00 = PA off	0x0B = LB EDGE/Linear Tx	0x0F = HB EDGE/Linear Tx
					0x0A = LB GMSK/Vramp Tx	0x0E = HB GMSK/Vramp Tx	Other = Reserved (Do Not Use)
Register 1, Address 0x01 (BIAS_CTRL)							
[7:4]	PA Stage 3 Bias (DAC3)	Trigger0	R/W	0000	0000 = 250 µA	0110 = 1750 µA	1100 = 3250 µA
					0001 = 500 µA	0111 = 2000 µA	1101 = 3500 µA
					0010 = 750 µA	1000 = 2250 µA	1110 = 3750 µA
					0011 = 1000 µA	1001 = 2500 µA	1111 = 4000 µA
					0100 = 1250 µA	1010 = 2750 µA	
					0101 = 1500 µA	1011 = 3000 µA	
[3:0]	PA Stage 1-2 Bias (DAC12)			0000	0000 = 250 µA	0110 = 1750 µA	1100 = 3250 µA
					0001 = 500 µA	0111 = 2000 µA	1101 = 3500 µA
					0010 = 750 µA	1000 = 2250 µA	1110 = 3750 µA
					0011 = 1000 µA	1001 = 2500 µA	1111 = 4000 µA
					0100 = 1250 µA	1010 = 2750 µA	
					0101 = 1500 µA	1011 = 3000 µA	
Register 2, Address 0x02 (ASM_CTRL)							
[7:4]	(RESERVED)	Trigger0	R/W	0000	Reserved (set to all zeroes)		
[3:0]	LB Switch Control		R/W	0000	0000 = Standby	0011 = L_TRx3	0110 = L_TRx6
					0001 = L_TRx1	0100 = L_TRx4	0111 = L_TRx7
					0010 = L_TRx2	0101 = L_TRx5	1111 = LB PA Out
							Other = isolation
Register 3, Address 0x03 (ASM_CTRL2)							
[7:5]	HB Switch Control	Trigger0	R/W	000	000 = Standby	010 = H_TRx2	100 = H_TRx4
					001 = H_TRx1	011 = H_TRx3	Other = isolation
[4:0]	(RESERVED)			00000	(Reserved; set to zero)		
Register 4, Address 0x04 (CPL_CTRL)							
[7:5]	(RESERVED)	Trigger0		000	(Reserved; set to zero)		
[4:3]	Coupler Mode		R/W	00	00 = Standby	10 = MB Coupler ON	
					01 = LB Coupler ON	11 = HB Coupler ON	
[2]	Coupler Directionality		R/W	0	0 = Forward	1 = Reverse	
[1:0]	(RESERVED)		R/W	00	Reserved; set to 00		
Register 5, Address 0x05 (ASM_CTRL3)							
[7:4]	(RESERVED)		R/W	0x00	Reserved (set to zero)		
[3:0]	MB Switch Control			0000	0000 = Standby	0011 = M_TRx3	1111 = MB PA Out
					0001 = M_TRx1	0100 = M_TRx4	
					0010 = M_TRx2	0101 = M_TRx5	Other = isolation

MIPI data的波形解析

➢ 当问题难以厘清时，往往需要手动量取MIPI信号波形做确认。
如下所示为MIPI data的物理形状解析和表达格式：

- Example of Register write command
  - Slave id = 15 (0xF)
  - Register = 1
  - Data = 0xC

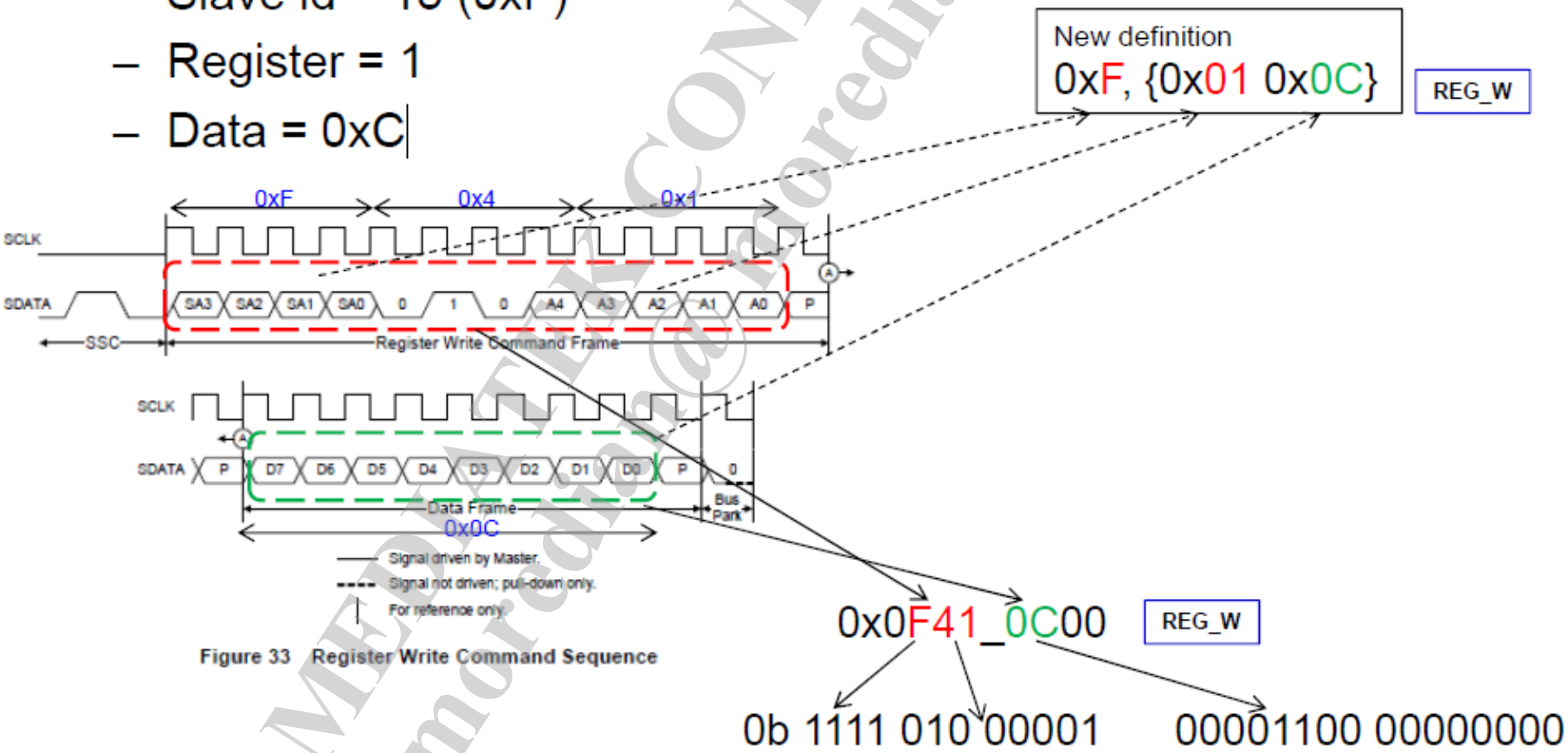


Figure 33 Register Write Command Sequence

MIPI timing——Unit

- MTK platform RAT timing unit
  - 2G timing Unit QB ， 1QB=0.923uS
  - 3G WCDMA uS
  - 3G TDSCDMA echip, echip=(1/8)chip=0.78125/8=0.09765625uS=0.1us
  - 4G LTE uS
  - C2K uS
- 93 Modem NVRAM Edit Timing Value related to L1 code Timing Value
  - 2G timing Nvram Value= L1 Code value
  - 3G WCDMA Nvram Value= [(L1 Code value)***3.84**]
    ul1d_mipi_public.h #define US2CHIPCNT(us) ((us)*3.84)
  - 3G TDSCDMA 不建议修改
  - 4G LTE Nvram Value=[(L1 Code value)*61.44]
    mml1_mipi_public.h #define US2OFFCNT(us) ((us)*6144/100) ;
  - C2K Nvram Value=L1 code Value

MIPI timing——One MIPI Data time

- MTK不同平台RF FE Modem One MIPI data time period[**Tcd**]
  - MT6177M/MT6177 Modem MIPI CLK 25M

    One MIPI data time period = 24/25MHz=**0.96uS**
  - MT6176 Modem MIPI CLK 22.75M

    One MIPI data time period = 24/22.75MHz=**1.05uS**
  - MT6169 Modem MIPI CLK 20M

    One MIPI data time period = 24/20MHz=**1.2uS**

- 目前通常采用的MIPI command sequence，一笔MIPI data 含有24bits, 具体如下图

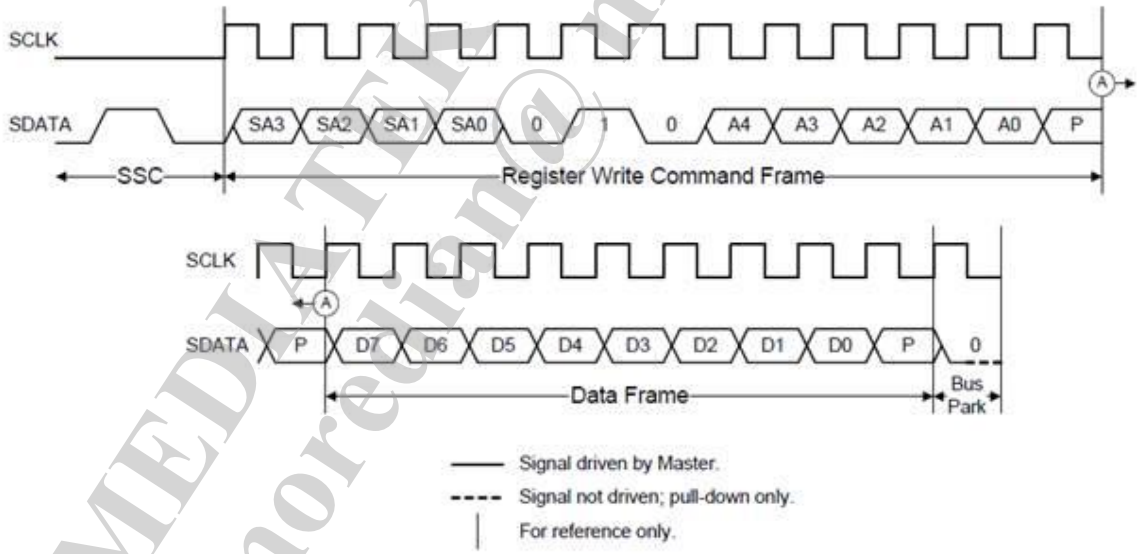
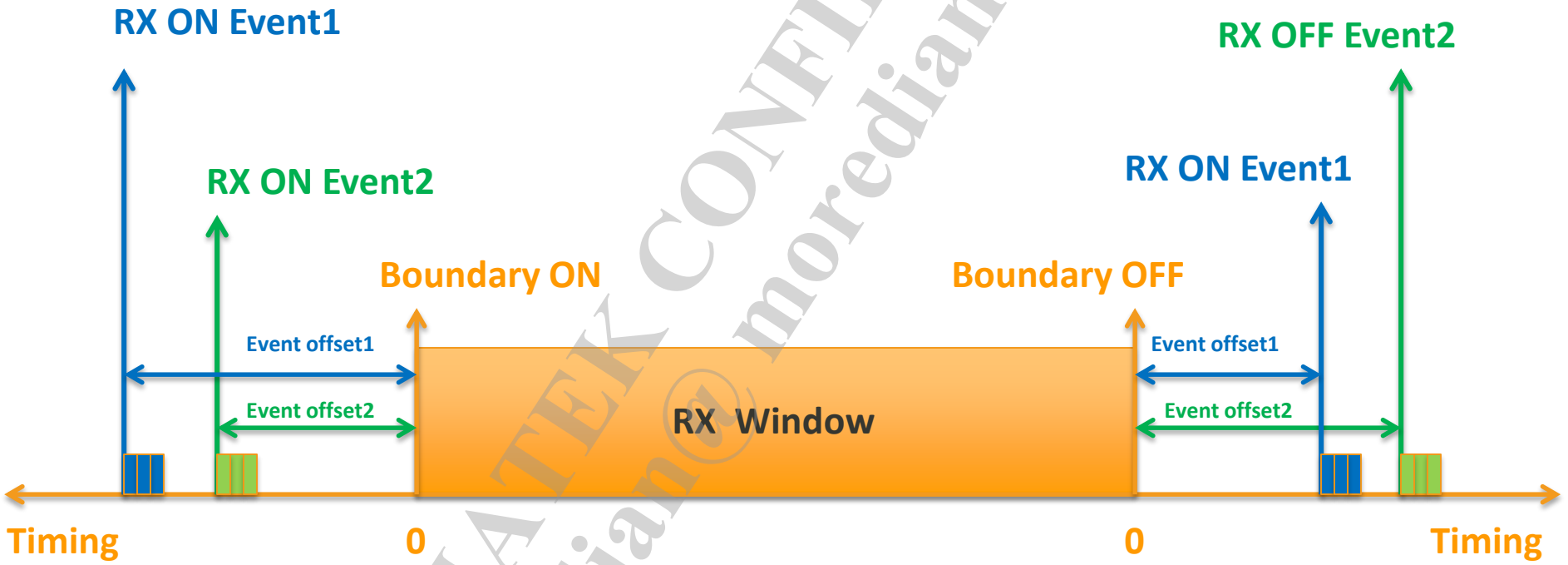


Figure 33 Register Write Command Sequence

MIPI timing——RX WIN

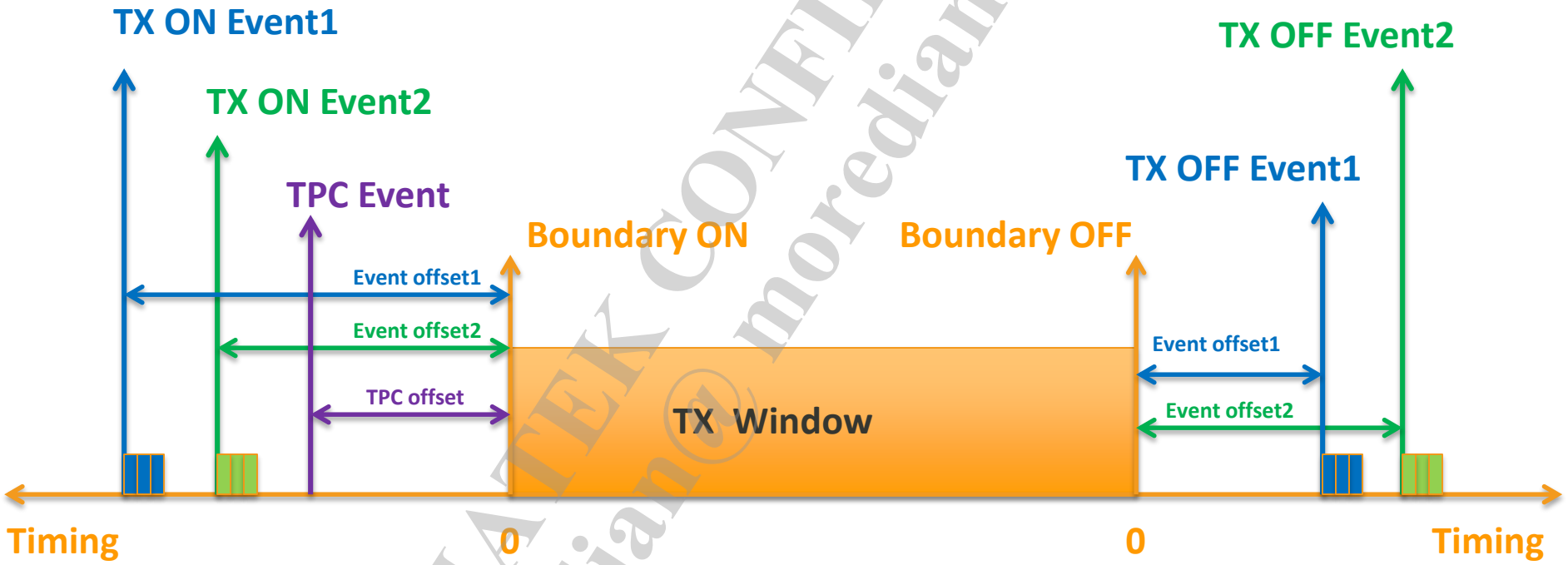
- 通常以TX/RX Boundary 作为时序的参考基准点，对某一笔MIPI Event定义其ON或OFF的时间
- LTE Rx Window Timing



Note:请参考默认设定的时序值！时序改的太过分可能会导致编译错误，或其他奇怪的问题。

MIPI timing——TX

- 通常以TX/RX Boundary 作为时序的参考基准点，对某一笔MIPI Event定义其ON或OFF的时间。
- LTE TX Window timing

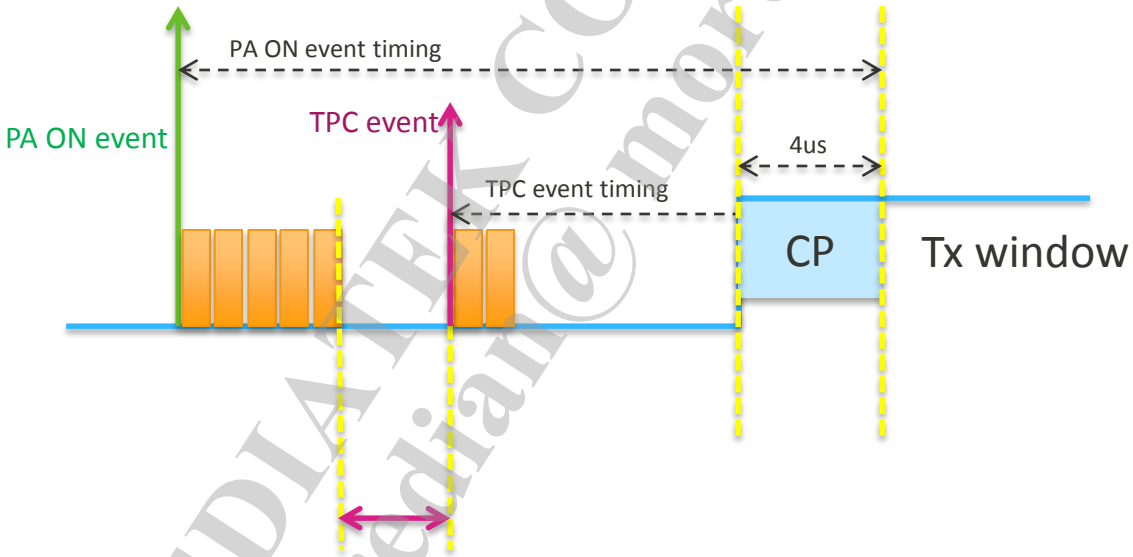


Note：请参考默认设定的时序值！
改的太过分可能会导致编译错误，或其他奇怪的问题。
尤其注意TPC的timing要放在PA ON的后面！

93 modem PA on event timing (1/2)

- LTE Timing

Criteria:
The timing difference between PA ON event timing - (PA on data #)*Tcd and TPC event timing should be larger than 5.5us



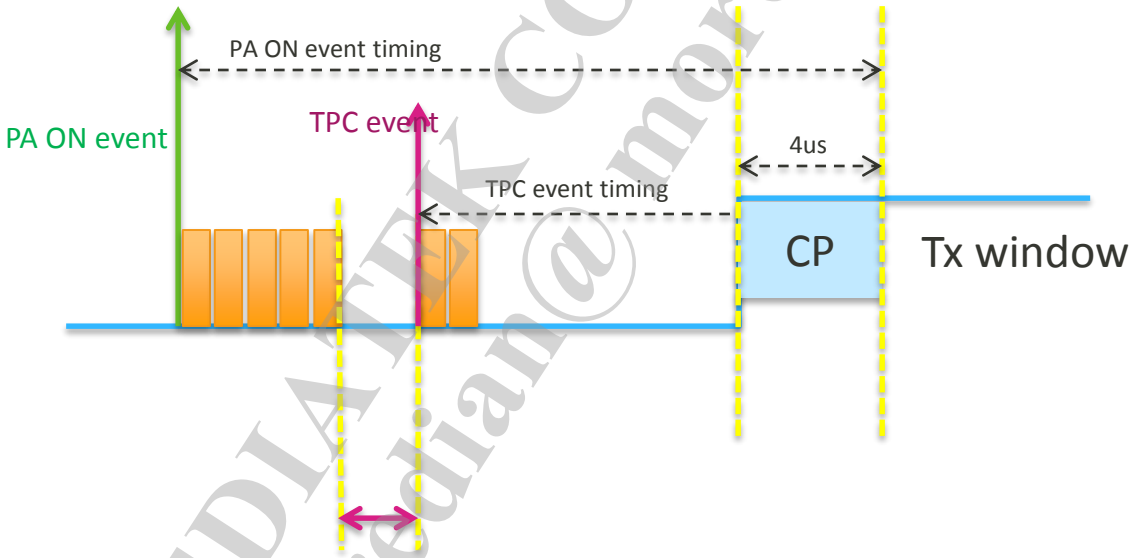
At least >= 5.5us

PA ON event timing - (PA on data #)*Tcd - TPC event >= 5.5us

93 modem PA on event timing (2/2)

- LTE Timing

Criteria:
The timing difference between PA ON event timing - (PA on data #)*Tcd and TPC event timing should be larger than 5.5us



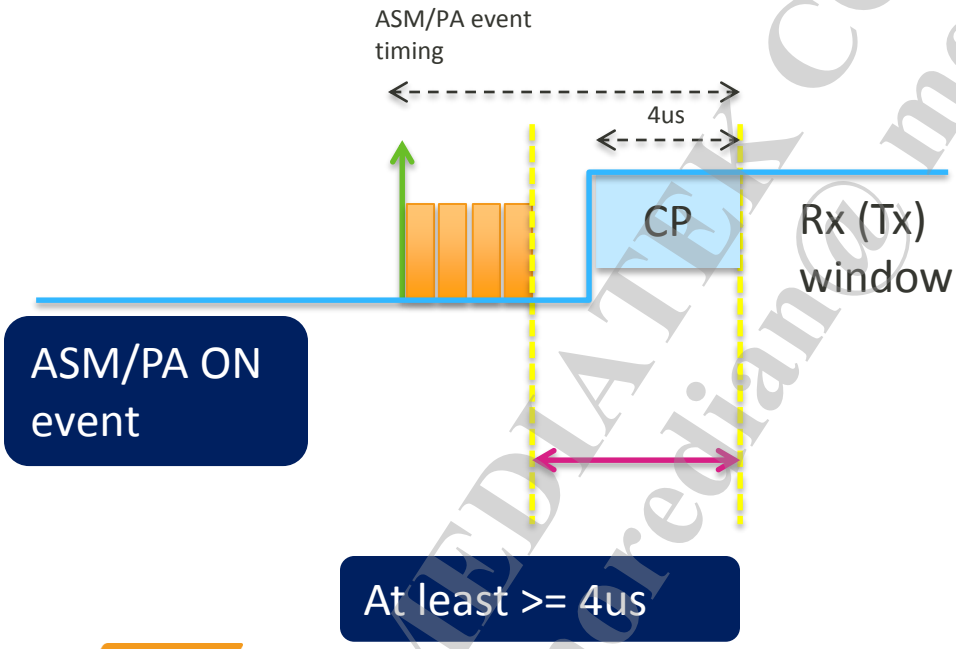
if < 5.5us
=> Show Warning message!!!

PA ON event timing - (PA on data #)*Tcd - TPC event >= 5.5us

93 modem ASM/PA on event timing (1/2)

LTE Timing

Criteria:
The timing of ASM/PA ON event timing - (ASM/PA on data #)*Tcd should be larger than 4us (due to CP 4us consideration)

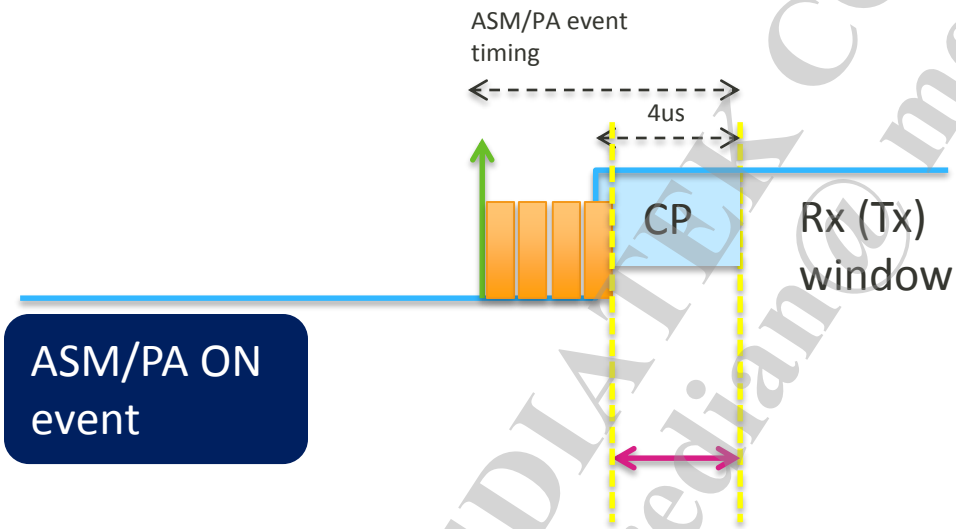


ASM/PA ON event timing - (ASM/PA on data #)*Tcd >= 4us

93 modem ASM/PA on event timing (2/2)

LTE Timing

Criteria:
The timing of ASM/PA ON event timing - (ASM/PA on data #)*Tcd should be larger than 4us (due to CP 4us consideration)



ASM/PA ON event

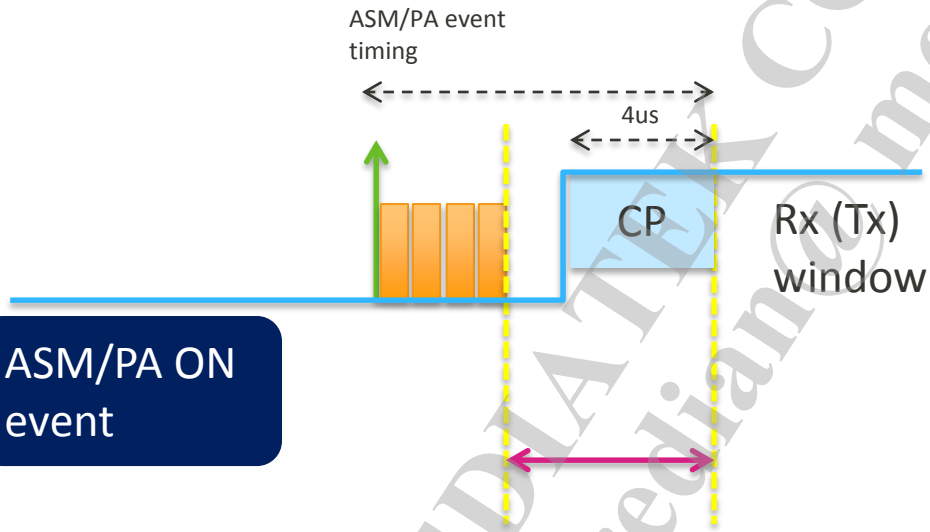
if < 4us
=> Show Warning message!!!

ASM/PA ON event timing - (ASM/PA on data #)*Tcd >= 4us

93 modem TPC event timing check

- LTE Timing

Criteria:
The timing of TPC event timing – (TPC Data #)*Tcd should be larger than 4us (due to CP 4us consideration)



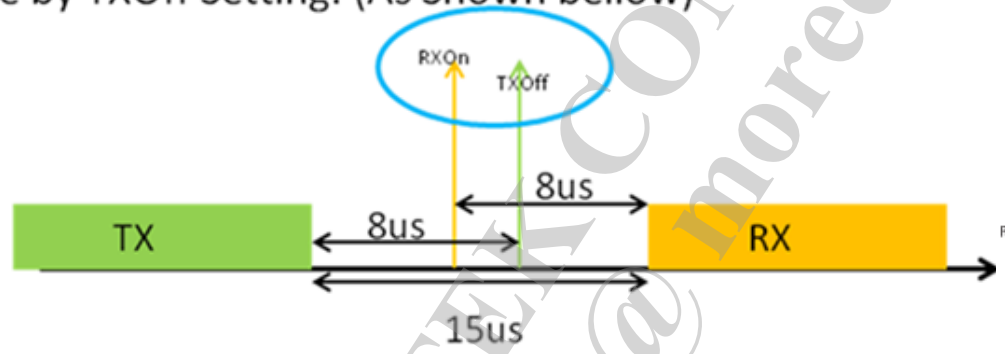
TPC event timing - (TPC on data #)*Tcd >= 4us

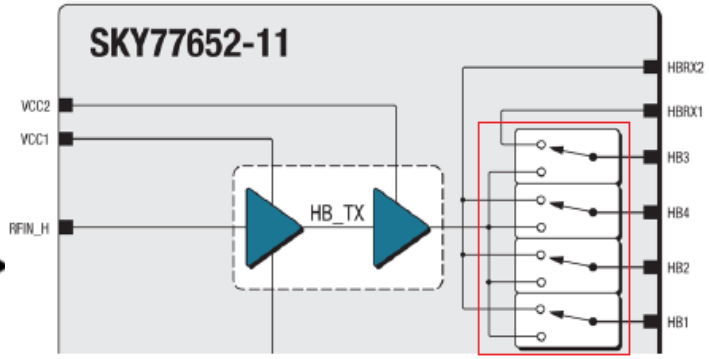
93 modem LTE TDD event timing

LTE Timing

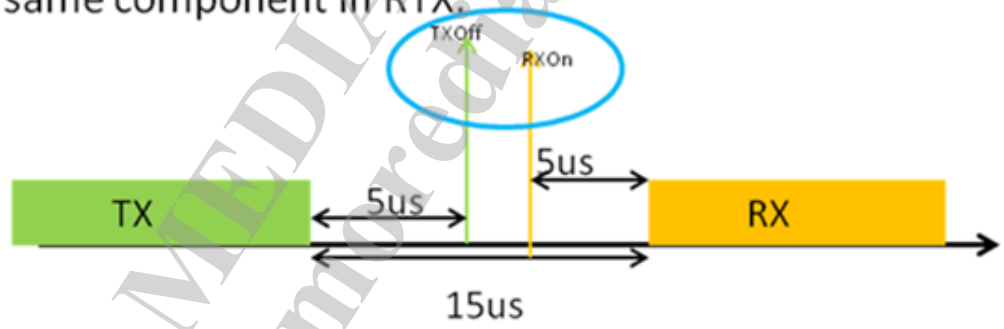
- LTE TDD TX OFF&RX ON timing Limit:
  Tx_OFF +Rx_ON(Left most Event） <=15us
- 3P4T Switch are shared to Tx Path and Rx Path for LTE TDD MMMB PA

- In some scenario, Timing difference between TX and RX burst will less than 15us. If you use the same component in RTX, RXOn Setting will be replace by TXOff Setting. (As Shown bellow)





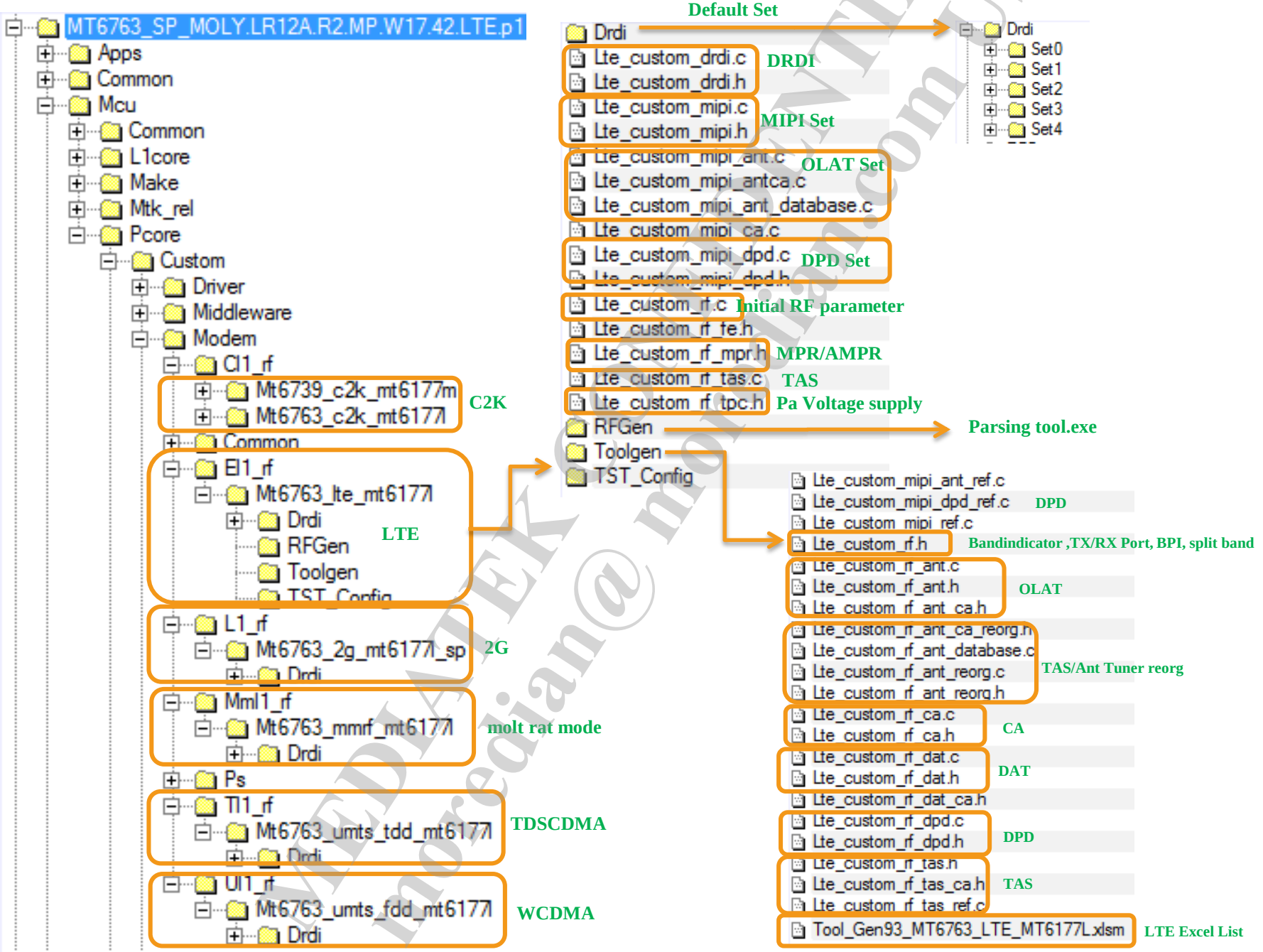
- So we suggest that RXOn And TXOff timing should be less than 5us,if you use the same component in RTX.



MEDIATEK

MIPI Configuration Guide

MT6177 Modem code tree



90/91 MIPI的相关设定文件

	File Name	Defines
l1_rf	l1d_custom_mipi.h	2G_MIPI_ENABLE; T/Rx events timing(should not modify)
	l1d_custom_mipi.c	MIPI Register and Data
	l1d_custom_rf.h	BPI Timing; BPI configration; T/Rx Ports; power rollback
	m12193.c	.ini default value
tl1_rf	tl1d_custom_rf.h	TDS_MIPI_ENABLE; BPI configration; T/Rx Ports
	tl1d_custom_mipi.h	MIPI Register and Data
	t12193.h	BPI & MPI Timing;
	t12193.c	.ini default value
	tl1d_custom_drdi.h	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬
ul1_rf	ul1d_custom_mipi.h	3G_MIPI_ENABLE;定义了在ul1d_custom_mipi.c中每个band的event/data的个数(should not modify)
	ul1d_custom_mipi.c	MIPI Register and Data; TPC
	ul1d_custom_rf.h	BPI configuration; PRx/DRx/Tx ports and Power rollback; Band indicator
	wcustomdata.c	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬
	wcustomdata.h	
	u12193.c	.ini default value
el1_rf	lte_custom_drdi.h	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬
	lte_custom_drdi.c	
	lte_custom_mipi.h	MIPI_FRONT_END_ENABLE;定义了在lte_custom_mipi.c中每个band的event/data的个数(should not modify)
	lte_custom_mipi.c	MIPI Register and Data; TPC
	lte_custom_rf.h	BPI configration;PRx/DRx/Tx ports and Power rollback; Band indicator
	lte_custom_rf.c	.ini default value
mml1_rf	mml1_custom_drdi.h	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬
	mml1_custom_drdi.c	
	mml1_custom_mipi.h	USID
	mml1_custom_mipi.c	在同一MIPI port, 使用不同vender提供的MIPI元件, 但USID一样时, 可在此文件中将某一元件的USID更换为另一unique的USID
	mml1_custom_rf.h	CLK enable

93 MIPI的相关设定文件(1)

	File Name	Defines
l1_rf	l1d_custom_mipi.h	2G_MIPI_ENABLE; T/Rx events timing(should not modify)
	l1d_custom_mipi.c	MIPI Register and Data
	l1d_custom_rf.h	BPI Timing; BPI configration; T/Rx Ports; power rollback
	m12193.c	.ini default value
	l1d_custom_rf_dat.h	Dat [Dynamic_Antenna_Tuner] L1D_DAT_ENABLE， Ant tuner state setting
	l1d_custom_rf_dat.c	Dat [Dynamic_Antenna_Tuner] L1D_DAT_CAT_A/B_DATABASE_Set
	l1d_custom_rf_tas.h	TAS [Transmit Antenna Switch];TAS Enable,Ant State Setting
	l1d_custom_rf_tas.c	L1D_TAS_CAT_A/B_DATABASE_Set
tl1_rf	tl1d_custom_rf.h	TDS_MIPI_ENABLE; BPI configration; T/Rx Ports
	tl1d_custom_mipi.h	MIPI Register and Data ;TAS Enable,
	t12193.h	BPI & MPI Timing;
	t12193.c	.ini default value
	tl1d_custom_drdi.h	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬
	tl1d_custom_rf_dat.h	Dat [Dynamic_Antenna_Tuner] TL1D_DAT_ENABLE， Ant tuner state setting
	tl1d_custom_rf_dat.c	Dat [Dynamic_Antenna_Tuner] TL1D_DAT_CAT_A/B_DATABASE_Set
	tl1d_custom_rf_tas.h	TAS [Transmit Antenna Switch];TAS Enable,Ant State Setting
	tl1d_custom_rf_tas.c	TL1D_TAS_CAT_A/B_DATABASE_Set
ul1_rf	ul1d_custom_mipi.h	3G_MIPI_ENABLE; 定义了在ul1d_custom_mipi.c中每个band的event/data的个数(should not modify)
	ul1d_custom_mipi.c	MIPI Register and Data; TPC
	ul1d_custom_rf.h	BPI configration;PRx/DRx/Tx ports and Power rollback; Band indicator
	wcustomdata.c	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬
	wcustomdata.h	
	u12193.c	.ini default value
	ul1d_custom_rf_ca.h	CA CC0&CC1,TRX Port/TRX BPI/MIPI Route Setting; No Care
	ul1d_custom_mipi_ca.c	CA MIPI Rx Route Setting； No Care
	ul1d_custom_rf.c	NCCA Setting/ PA PMIC supply Setting; No Care
	ul1d_custom_rf_dat.h	Dat [Dynamic_Antenna_Tuner] UL1D_DAT_ENABLE， Ant tuner state setting
	ul1d_custom_rf_dat.c	Dat [Dynamic_Antenna_Tuner] UL1D_DAT_CAT_A/B_DATABASE_Set
	ul1d_custom_rf_tas.h	TAS [Transmit Antenna Switch];TAS Enable,Ant State Setting
	ul1d_custom_rf_tas.c	UL1D_TAS_CAT_A/B_DATABASE_Set
	ul1d_custom_rf_dpd.h	Digital Pre-Distortion(DPD)/
	ul1d_custom_rf_dpd.c	DPD Enable, dpd_apt_temperature_th_by_rfic/PCFE power threshold setting

93 MIPI的相关设定文件(2)

	File Name	Defines	
el1_rf	lte_custom_drdi.h	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬	Manual setting
	lte_custom_drdi.c		
	lte_custom_mipi.h	MIPI_FRONT_END_ENABLE;定义了在lte_custom_mipi.c中每个band的event/data的个数 (should not modify)	
	lte_custom_mipi.c	MIPI Register and Data; TPC	
	lte_custom_mipi_ca.c	CA new route MIPI Route Setting	
	lte_custom_rf.c	.ini default value	
	lte_custom_mipi_ant.c	OLAT[open loop ant tuner] NCA,ant tuner MIPI setting	
	lte_custom_mipi_antca.c	OLAT[open loop ant tuner] CA ant tuner MIPI setting	
	lte_custom_mipi_ant_database.c	TAS DPDT BPI setting，TAS Ant tuner MIPI Seting	
	lte_custom_rf_mpr.h	LTE MPR/AMPR Setting	
	lte_custom_rf_tpc.h	PA PMIC supply setting/sptial RB Tx Power Backoff/temp THRES setting	
	lte_custom_rf_tas.c	EL1D_TAS_CAT_A/B_DATABASE_Set	
	lte_custom_rf_fe.h	BPI Timing setting	
	lte_custom_mipi_dpd.h	Dat [Dynamic_Antenna_Tuner] UL1D_DAT_ENABLE， Ant tuner state setting	
	lte_custom_mipi_dpd.c	DPD Enable, dpd_apt_temperature_th_by_rfic/PCFE power threshold setting	
	lte_custom_rf.h	Bandindicator,BPI configuration,PRx/DRx/Tx ports ;ELNA HPUE;Split&Filter Band	Toolgen /Parsing Tool. Gen file from Tool_Gen93_MT67**_LTE_MT6177*.xlsm
	lte_custom_rf_ca.h	Rx Alternate FE route/Tx Alternate FE route define	
	lte_custom_rf_ca.c	CA capability setting	
	lte_custom_rf_ant.h	OLAT Tuner Setting	
	lte_custom_rf_ant.c	CA OLAT Tuner Setting	
	lte_custom_rf_ant_ca.h	OLAT ANT_Tuner, ANT_ALTERNATE_CONFIGURE Setting[CA No single Band route]	
	lte_custom_mipi_ant_ref.c	OLAT MIPI TX&RX Setting	
	lte_custom_rf_ant_database.c	Reorge Ant Tuner Database Setting, LTE_ANT_CAT_A/B_DATABASE_Set[DAT/TAS]	
	lte_custom_rf_ant_reorg.h	Reorge By band Ant tuner setting	
	lte_custom_rf_ant_ca_reorg.h	Reorge CA Ant tuner state setting	
	lte_custom_rf_ant_reorg.c	Reorge ant tuner mapping	
	lte_custom_rf_dat.h	Dat [Dynamic_Antenna_Tuner] EL1D_DAT_ENABLE， Ant tuner state setting	
	lte_custom_rf_dat.c	Dat EL1D_DAT_CAT_A/B_DATABASE_Seting	
	lte_custom_rf_dat_ca.h	Dat CA EL1D_DAT_ENABLE， Ant tuner state setting	
	lte_custom_rf_dpd.c	DPD Enable, dpd_apt_temperature_th_by_rfic/PCFE power threshold setting	
	lte_custom_rf_dpd.h	Digital Pre-Distortion(DPD)/	
	lte_custom_mipi_dpd_ref.c	Digital Pre-Distortion(DPD)/	
	lte_custom_rf_tas.h	TAS [Transmit Antenna Switch];TAS Band Enable,Ant State Setting	
	lte_custom_rf_tas_ca.h	TAS CA(PCC/SCC_ TAS Enable,Ant State Setting	
	lte_custom_rf_tas_ref.c	TAS database table setting, LTE_TAS_CAT_A&B_DATABASE_Set Setting	

93 MIPI的相关设定文件(3)

	File Name	Defines	Note
Cl1_rf	c2k_custom_drdi.c	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬	
	c2k_custom_drdi.h		
	c2k_custom_elna.c	C2K eLAN Set	
	c2k_custom_elna.h		
	c2k_custom_mipi.c	MIPI Set	
	c2k_custom_mipi.h	MIPI timng	
	c2k_custom_rf.c	Initial rf parameter	
	c2k_custom_rf.h	Bandindicator/TRX Port/BPI	
	c2k_custom_rf_dat.c	Dat CL1D_DAT_CAT_A/B_DATABASE_Seting	
	c2k_custom_rf_dat.h	Dat [Dynamic_Antenna_Tuner] CL1D_DAT_ENABLE， Ant tuner state setting	
	c2k_custom_rf_dpd.c	DPD Enable	
	c2k_custom_rf_dpd.h	DPD	
	c2k_custom_rf_tas.c	CL1D_TAS_CAT_A/B_DATABASE_Set	
	c2k_custom_rf_tas.h	TAS [Transmit Antenna Switch];TAS Band Enable,Ant State Setting	
	c2k_custom_rf_tuner.c	OLAT[open loop ant tuner] ant tuner MIPI setting	
	c2k_custom_rf_tuner.h	Ant state set	
	c2k_custom_rf_tas_tst.h	TST bandindicator enable/TST Ant state by band	
mml1_rf	mml1_custom_drdi.h	Dynamic Radio-setting Dedicated Image(DRDI),单软多硬	
	mml1_custom_drdi.c		
	mml1_custom_mipi.h	USID define	
	mml1_custom_mipi.c	Initial CW Table Setting ; LPM(sleep mode) CW Setting;同一MIPI port, 使用不同vender提供的MIPI元件, 但USID一样时, 可在此文件中将某一元件的USID更换为另一unique的USID;	
	mml1_custom_rf.h	CLK enable is invalid(*.DWS Setting by DCT Tool)	
	mml1_custom_elna.h	ELNA Enable/ Control Type/timing/BPI Control Logical setting	
	mml1_custom_elna.c	ELNA MIPI setting	
	mml1_custom_ext_buck_ldo.c	HPUE External DC-DC Setting	
	mml1_custom_rf.c	TADC for 8 temperature mapping	
	mml1_custom_rf.h	CLK enable and CLK buffer Drivering are invalid(*.DWS Setting by DCT Tool); IQ SWAP;	

CONFIDENTIAL B

MML1 Part

MML1--USID

- 在mml1_custom_mipi.h中定义使用到器件的USID

```
#define MIPI_USID_INIT0_SetDefault          0x0000
#define MIPI_USID_ASM0_SetDefault           0x000C //Main 3P18T(TXM)
#define MIPI_USID_ASM1_SetDefault           0x000A //DRX ASM DP12T(LMB)
#define MIPI_USID_ASM2_SetDefault           0x000B //DRX ASM SP6T(HB)
#define MIPI_USID_ASM3_SetDefault           0x0009
#define MIPI_USID_PA0_SetDefault            0x000F //HMB PA
#define MIPI_USID_PA1_SetDefault            0x000D //LB PA
#define MIPI_USID_PA2_SetDefault            0x000E //B42 PA
#define MIPI_USID_ANT0_SetDefault           0x0006
#define MIPI_USID_TAS_SetDefault            0x0007
#define MIPI_USID_EXT_VPA_SetDefault        0x0005
```

- 在mml1_custom_mipi.c,修改USID,同一个MIPI port上挂载的器件有USID冲突时，在mml1_custom_mipi.c中修改掉其中一个器件的USID。按照如下示意:90 modem即使不同组MIPI BUS，也不允许相同的USID设置

```
/*****************************************************************************
 * MML1 MIPI Change USDI Table Data
 *****************************************************************************/
const MML1_MIPI_USID_CHANGE_T MML1_MIPI_USID_CHANGE_TABLE_SetDefault[MML1_MIPI_MAX_USID_CHANGE_NUM] =
{
  // USID change type   , port_sel       , current USID , PRODUCT_ID , MANUFACTORY_ID   new USID
  {USID_REG_W          , MML1_MIPI_PORT0 , 0xE          , 0x9B       , 0x1A5          , 0xC    }, // Main 3P18T
  {USID_REG_W          , MML1_MIPI_PORT1 , 0xF          , 0x00       , 0x1A5          , 0xD    }, // LB PA
  {USID_REG_W          , MML1_MIPI_PORT1 , 0xF          , 0x10       , 0x286          , 0xE    }, // B42 PA
  {USID_REG_W          , MML1_MIPI_PORT2 , 0x9          , 0x18       , 0x134          , 0xB    }, // DRX ASM SP6T(HB)
  {USID_NULL,0,0,0,0,0},
};
```

Note:如果是完全相同的两个器件，则无法区分修改!

MML1--Initial&LPM CW Table

- 在mml1_custom_mipi.c, 做RF FE HW Part的初始化

```
const MML1_MIPI_INITIAL_CW_T MML1_MIPI_INITIAL_CW_SetDefault[MML1_MIPI_MAX_INITIAL_CW_NUM] =
{
  // elm type    , port_sel      , data_seq ,    USID          , addr , data  , wait_time(us)
  {MML1_MIPI_ASM , MML1_MIPI_PORT0, MML1_REG_W, MIPI_USID_ASM0_SetDefault  , {0x1C, 0x38} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT1, MML1_REG_W, MIPI_USID_PA0_SetDefault   , {0x1C, 0x38} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT1, MML1_REG_W, MIPI_USID_PA1_SetDefault   , {0x1C, 0x38} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT1, MML1_REG_W, MIPI_USID_PA2_SetDefault   , {0x1C, 0x38} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT2, MML1_REG_W, MIPI_USID_ASM1_SetDefault  , {0x1C, 0x38} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT2, MML1_REG_W, MIPI_USID_ASM2_SetDefault  , {0x1C, 0x38} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
};
```

Note：Initial Table 初始化的MIPI Port及USID 必须与HW 设计一致，且各个RAT MIPI.c， CW 中MIPI 及USID也必须与一致，否则会出现Modem Assert

- 在mml1_custom_mipi.c,LPM(Sleep mode) CW Table 设置

```
const MML1_MIPI_LPM_CW_T MML1_MIPI_LPM_CW_SetDefault[MML1_MIPI_MAX_LPM_CW_NUM] =
{
  // elm type    , port_sel      , data_seq ,    USID          , addr , data  , wait_time(us)
  {MML1_MIPI_ASM , MML1_MIPI_PORT0, MML1_REG_W, MIPI_USID_ASM0_SetDefault  , {0x1C, 0xB8} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT1, MML1_REG_W, MIPI_USID_PA0_SetDefault   , {0x1C, 0xB8} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT1, MML1_REG_W, MIPI_USID_PA1_SetDefault   , {0x1C, 0xB8} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT1, MML1_REG_W, MIPI_USID_PA2_SetDefault   , {0x1C, 0xB8} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT2, MML1_REG_W, MIPI_USID_ASM1_SetDefault  , {0x1C, 0xB8} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_ASM , MML1_MIPI_PORT2, MML1_REG_W, MIPI_USID_ASM2_SetDefault  , {0x1C, 0xB8} , 0 }, // Specific ID, Standard MIPI, PM_TRIG = normal mode
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
  {MML1_MIPI_END_PATTERN,0,0,0,{0,0},0},
};
```

MML1--ELNA CW

- mml1_custom_elna.h， ELNA category setting/control interface/Settling time/PIN selected

```
//ELNA Catgory Setting:
#define MML1_FE_ELNA_NONE_CATEGORY_SetDefault     MML1_ELNA_OFF
#define MML1_FE_ELNA1_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA2_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA3_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA4_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA5_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA6_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA7_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA8_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA9_CATEGORY_SetDefault         MML1_ELNA_OFF
#define MML1_FE_ELNA10_CATEGORY_SetDefault        MML1_ELNA_OFF
#define MML1_FE_ELNA11_CATEGORY_SetDefault        MML1_ELNA_OFF
#define MML1_FE_ELNA12_CATEGORY_SetDefault        MML1_ELNA_OFF
```

```
//Select eLNA control interface
#define MML1_FE_ELNA_NONE_CONTROL_INTERFACE_SetDefault  MML1_CONTROL_NONE
#define MML1_FE_ELNA1_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA2_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA3_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA4_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA5_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA6_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA7_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA8_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA9_CONTROL_INTERFACE_SetDefault      MML1_CONTROL_NONE
#define MML1_FE_ELNA10_CONTROL_INTERFACE_SetDefault     MML1_CONTROL_NONE
#define MML1_FE_ELNA11_CONTROL_INTERFACE_SetDefault     MML1_CONTROL_NONE
#define MML1_FE_ELNA12_CONTROL_INTERFACE_SetDefault     MML1_CONTROL_NONE
```

```
//PIN selected
//  The settings for ELNA1
#define PDATA_MML1_FE_ELNAX_PIN0_SetDefault           MML1_ELNA_PIN_NONE  // the 1st BPI pin number for the ELNA control, set MML1_ELNA_PIN_NONE if the BPI contorl is not used.
#define PDATA_MML1_FE_ELNAX_PIN1_SetDefault           MML1_ELNA_PIN_NONE  // the 2nd BPI pin number for the ELNA control, set MML1_ELNA_PIN_NONE if the BPI contorl is not used.
#define PDATA_MML1_FE_ELNAX_PIN0_ON_DATA_SetDefault   1                   // the ON mode BPI setting (1 or 0) for the 1st BPI pin
#define PDATA_MML1_FE_ELNAX_PIN1_ON_DATA_SetDefault   1                   // the ON mode BPI setting (1 or 0) for the 2nd BPI pin
#define PDATA_MML1_FE_ELNAX_PIN0_BYPASS_DATA_SetDefault 0                 // the BYPASS mode BPI setting (1 or 0) for the 1st BPI pin
#define PDATA_MML1_FE_ELNAX_PIN1_BYPASS_DATA_SetDefault 0                 // the BYPASS mode BPI setting (1 or 0) for the 1st BPI pin
```

Note： X=0,1….12,详细参照ELNA 配置相关文档,如果没有做ELNA,请将Pin Control 设置成**MML1_ELNA_PIN_NONE**,否则会造成该BPI不受控

MML1--clock buffer

- 在mml1_custom_rf.h文件中的clock enable已经无效，都放在*.DWS中处理（DCT Tool Setting）

```
/*  Enable or disable the clock1, 2, 3, and 4 output  */
/*  1 : Enable                                        */
/*  0 : Disable                                       */
/*-------------------------------------------------------*/
#define MML1_CLK1_EN_SetDefault                  1 /* CLK1 is enabled for BB */
#define MML1_CLK2_EN_SetDefault                  0
#define MML1_CLK3_EN_SetDefault                  1
#define MML1_CLK4_EN_SetDefault                  1
```

CONFIDENTIAL B

GSM Part

GSM Part --RX

- 列出RX通路ON/OFF时所需的操作步骤：

Event No.	Event Type	Action	Data	Data No.
1	RX ON	设置到Idle状态	将ASM M Port 设置成Idle	0
			将ASM H Port 设置成Idle	1
2	RX ON	设置L TRX Port Active状态	打开ASM L Port对应的RX Path	2
3	RX OFF	关闭RX 通路	将ASM 设置成idle状态	3

Note： Ex:Sky77928-11,
Off Event Sky77928-11 ,2G/TD/LTE TD需要SW Reset 0x23,0x80;

GSM Part --RX

- RX部分的格式设置介绍：

Timing在l1d_custom_mipi.h中定义

```
/*  Event RX Timing Define                                  */
/*----------------------------------------------------------*/
/*MT6177L*/ #define QB_MIPI_RX_ON0_Set0      152
/*MT6177L*/ #define QB_MIPI_RX_ON1_Set0      65
/*MT6177L*/ //#define QB_MIPI_RX_ON2       173
/*MT6177L*/ //#define QB_MIPI_RX_ON3       173
/*MT6177L*/ //#define QB_MIPI_RX_ON4       173
/*MT6177L*/ //#define QB_MIPI_RX_ON5       173
/*MT6177L*/ //#define QB_MIPI_RX_ON6       173
/*MT6177L*/ //#define QB_MIPI_RX_ON7       173
/*MT6177L*/ #define QB_MIPI_RX_OFF0_Set0     5
/*MT6177L*/ //#define QB_MIPI_RX_OFF1      173
/*MT6177L*/ //#define QB_MIPI_RX_OFF2      173
/*MT6177L*/ //#define QB_MIPI_RX_OFF3      173
/*MT6177L*/ //#define QB_MIPI_RX_OFF4      173
```

Timing

```
const sGGE_MIPI_CTRL_TABLE_BAND GGE_MIPI_CTRL_TABLE_GSM900_Set0=
{
  {
    /* GGE_MIPI_CTRL_TABLE_GSM900.mipi_rx_ctrl_table.mipi_rxctrl_event[] */
    { /*          element      data idx      event type      , event timing  */
      /*   No.     type   , { start, stop },                  ( QB )     */
      { /* 0 */  GGE_MIPI_ASM , {  0 , 1 }, GGE_MIPI_TRX_ON   , QB_MIPI_RX_ON0_Set0  },
      { /* 1 */  GGE_MIPI_ASM , {  2 , 2 }, GGE_MIPI_TRX_ON   , QB_MIPI_RX_ON1_Set0  },
      { /* 2 */  GGE_MIPI_ASM , {  3 , 3 }, GGE_MIPI_TRX_OFF  , QB_MIPI_RX_OFF0_Set0 },
```

Event

Data

```
/* GGE_MIPI_CTRL_TABLE_GSM900.mipi_rx_ctrl_table.mipi_rxctrl_data[] */
{     /*   No.    elm type  , port select    , data format          ,usid              ,{ { subband arfcn, addr, data }, { subband arfcn, ad
  { /* 0 */  GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W       ,MIPI_USID_ASM0_Set0 ,{ {   124       , 0x05, 0x00 }, {
  { /* 1 */  GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W       ,MIPI_USID_ASM0_Set0 ,{ {   124       , 0x03, 0x00 }, {
  { /* 2 */  GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W       ,MIPI_USID_ASM0_Set0 ,{ {   124       , 0x02, 0x02 }, {
  { /* 3 */  GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W_EXT_1ST ,MIPI_USID_ASM0_Set0 ,{ {   124       , 0x23, 0x80 }
```

GSM Part --TX

- 详细列出TX通路ON/OFF时所需的操作步骤:

Event No.	Event Type	Action	Data	Data No.
1	TX ON	HM TRX Port 设置Idle状态	将ASM MTRX Port设成Idle mode	0
			将ASM HTRX Port设成Idle mode	1
2	TX ON	Enable PA	设定0x01寄存器的Bias	2
			设置0X00，设定GMSK/8PSK LB PA out	3
			设定0x02，切换 L TRX Port	4
3	TX OFF	关闭TX 通路	将ASM设成low Power 或SW Reset	5

GSM Part --TX

- TX部分的格式设置介绍：

Timing在l1d_custom_mipi.h中定义

```
/*MT6177L*/ #define QB_MIPI_TX_ON0_Set0      23
/*MT6177L*/ #define QB_MIPI_TX_ON1_Set0      246
/*MT6177L*/ #define QB_MIPI_TX_ON2_Set0      10
/*MT6177L*/ #define QB_MIPI_TX_ON3_Set0      2
/*MT6177L*/ //#define QB_MIPI_TX_ON4      173
/*MT6177L*/ //#define QB_MIPI_TX_ON5      173
/*MT6177L*/ //#define QB_MIPI_TX_ON6      173
/*MT6177L*/ //#define QB_MIPI_TX_ON7      173
/*MT6177L*/ #define QB_MIPI_TX_OFF0_Set0     20
/*MT6177L*/ #define QB_MIPI_TX_OFF1_Set0     35
/*MT6177L*/ //#define QB_MIPI_TX_OFF2     173
/*MT6177L*/ //#define QB_MIPI_TX_OFF3     173
/*MT6177L*/ //#define QB_MIPI_TX_OFF4     173
```

Timing

```
GGE_MIPI_CTRL_TABLE_GSM900.mipi_tx_ctrl_table.mipi_txctrl_event[] */
/*          element      data idx      event type      , event timing  */
/*   No.     type   , { start, stop },                  ( QB )     */
{ /* 0 */  GGE_MIPI_ASM , {  0 , 1  }, GGE_MIPI_TRX_ON     , QB_MIPI_TX_ON1_Set0  },
{ /* 1 */  GGE_MIPI_PA  , {  2 , 4  }, GGE_MIPI_TRX_ON     , QB_MIPI_TX_ON2_Set0  },
{ /* 2 */  GGE_MIPI_ASM , {  5 , 5  }, GGE_MIPI_TRX_OFF    , QB_MIPI_TX_OFF1_Set0 },
```

Event

Data

```
GGE_MIPI_CTRL_TABLE_GSM900.mipi_tx_ctrl_table.mipi_txctrl_data[] */
  /* No.     elm type  , port select     , data format          ,usid                , { { subband arfcn, addr, data }, { subband arfcn, addr, data }, { sub
{ /* 0 */ GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W      ,MIPI_USID_ASM0_Set0 , { {  124 , 0x05, 0x00 }, {  GGE_NU
{ /* 1 */ GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W      ,MIPI_USID_ASM0_Set0 , { {  124 , 0x03, 0x00 }, {  GGE_NU
{ /* 2 */ GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W      ,MIPI_USID_ASM0_Set0 , { {  124 , 0x01, 0xF7 }, {  GGE_NU
{ /* 3 */ GGE_MIPI_PA  , GGE_MIPI_PORT0  , GGE_MIPI_REG_W      ,MIPI_USID_ASM0_Set0 , { {  124 , 0x00, GGE_MIPI_PA_G8 },
{ /* 4 */ GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W      ,MIPI_USID_ASM0_Set0 , { {  124 , 0x02, 0x0F }, {  GGE_NU
{ /* 5 */ GGE_MIPI_ASM , GGE_MIPI_PORT0  , GGE_MIPI_REG_W_EXT_1ST ,MIPI_USID_ASM0_Set0 , { {  124 , 0x23, 0x80 }, {  GG
```

这里才是真正设定GMSK/8PSK模式的地方：

```
/* GGE_MIPI_CTRL_TABLE_GSM900.mipi_tx_ctrl_table.mipi_txctrl_pa_data */
{ /* GMSK Data */
  /* subband0 data , subband1 data , subband2 data , subband3 data , subband4 data  */
  {
    {  0x8A    , 0x8A    , 0x8A    , 0x8A    , 0x8A    },
  },
  /* 8PSK Data */
  /* subband0 data , subband1 data , subband2 data , subband3 data , subband4 data  */
  {
    {  0x8B    , 0x8B    , 0x8B    , 0x8B    , 0x8B    },
  }
```

GSM Part --TX

- TX部分的格式设置介绍：

```
/* GGE_MIPI_CTRL_TABLE_GSM900.mipi_txmid_ctrl_table.mipi_txmidctrl_data[] */
{    /* No.      elm type  ,  port select      ,  data format          ,usid              , {  { subband arfcn, addr, data }, { subband arfcn, addr, data }, { subband
  { /*  0 */  GGE_MIPI_PA  ,  GGE_MIPI_PORT0    ,  GGE_MIPI_REG_W       ,MIPI_USID_ASM0_Set0  , {  {   124        , 0x00, GGE_MIPI_PA_G8 }, {
  { /*  1 */  GGE_MIPI_NULL, GGE_MIPI_DATA_NULL , GGE_MIPI_REG_W_NULL  ,MIPI_USID_INIT0_Set0 , {  {   GGE_NULL_ARFCN, 0x00, 0x00 }      , {
}
```

- 设置Multi-Slot TX CW Table，register address/Value, MIPI Port 设置正确

CONFIDENTIAL B

TDSCDMA Part

TDSCDMA

- tl1d_custom_mipi.h文件中需要特别注意的地方：

此处并非USID！
TDS代码中用这个slave id区分器件，从而分配时序。

```
/*  MIPI Module FIXED                                                  */
/*---------------------------------------------------------------------*/
/*MT6177L*/   #define TDD_PA_Set0              0x1     /*DO NOT CHANGE*/
/*MT6177L*/   #define TDD_ASM_Set0             0x2     /*DO NOT CHANGE*/
/*MT6177L*/   #define TDD_ETM_Set0             0x3     /*DO NOT CHANGE*/
/*MT6177L*/   #define DATA_NULL_Set0           0xFFFF  /*DO NOT CHANGE*/
/*MT6177L*/   #define DEVICE_NULL_Set0         0xF0F0  /*DO NOT CHANGE*/
/*MT6177L*/   #define PA_FLAG_Set0             0xF000  /*DO NOT CHANGE*/
```

设置TDD_TXM_FLAG_BAND34/39_Set0,如果band34/39 Tx reuse TXM

```
/*  MIPI Module CONFIGURATION                                          */
/*---------------------------------------------------------------------*/
/*MT6177L*/   #define TDD_TXM_FLAG_BAND34_Set0    (1 <<0) /*1: TXM DEVICE*/
                                                 /*0: NORMAL DEVICE*/
/*MT6177L*/   #define TDD_TXM_FLAG_BAND39_Set0    (1 <<1) /*1: TXM DEVICE*/
                                                 /*0: NORMAL DEVICE*/
```

TDSCDMA

- tl1d_custom_mipi.h 重新定义USID by band and MIPI

```
/*MT6177L*/  #define TDD_ASM_USID_BAND34_Set0   (MIPI_USID_ASM0_Set0 <<3)
/*MT6177L*/  #define TDD_ASM_USID_BAND39_Set0   (MIPI_USID_ASM0_Set0 <<3)
/*MT6177L*/  #define TDD_PA_USID_BAND34_Set0    (MIPI_USID_ASM0_Set0 <<3)
/*MT6177L*/  #define TDD_PA_USID_BAND39_Set0    (MIPI_USID_ASM0_Set0 <<3)
/*MT6177L*/  #define TDD_ETM_USID_Set0          (0 << 3)//no use

/*-----------------------------------------------------------------------------
/*  MIPI Port                                                                */
/*-----------------------------------------------------------------------------
/*MT6177L*/  #define MIPI_PORT0_Set0      (0 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT1_Set0      (1 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT2_Set0      (2 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT3_Set0      (3 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT4_Set0      (4 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT5_Set0      (5 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT6_Set0      (6 <<12) /*PORT_NUM<<12*/
/*MT6177L*/  #define MIPI_PORT7_Set0      (7 <<12) /*PORT_NUM<<12*/

/*MT6177L*/  #define MIPI_USID_INIT0_PORT0_Set0   0x0000
/*MT6177L*/  #define MIPI_USID_ASM0_PORT0_Set0    0x000F
/*MT6177L*/  #define MIPI_USID_ASM0_PORT3_Set0    0x003F
/*MT6177L*/  #define MIPI_USID_ASM1_PORT0_Set0    0x0008
/*MT6177L*/  #define MIPI_USID_ASM2_PORT0_Set0    0x000A
/*MT6177L*/  #define MIPI_USID_ASM3_PORT0_Set0    0x0009
/*MT6177L*/  #define MIPI_USID_PA0_PORT0_Set0    0x000F
/*MT6177L*/  #define MIPI_USID_PA0_PORT3_Set0    0x003F
/*MT6177L*/  #define MIPI_USID_PA1_PORT0_Set0    0x000E
/*MT6177L*/  #define MIPI_USID_ANT0_PORT0_Set0    0x0006
```

TDSCDMA--RX ON/OFF

```
typedef enum
{
  TDSCDMA_REG_R  = 0,
  TDSCDMA_REG_W  = 1,
}T_OP_TYPE;
```

0x05，0x01 TXM 切换到B39 Rx Port

```
/*MIPI Data Table For Band39*/
/*Data Num Do Not Exceed 16 For Each Scenario*/
{
  /*Band39 Rx On Data*/
  {
    /*          MODULE    ,       PORT      ,  OPERATION ,       USID              , ADDR, DATA*/
    {/* 0*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0 , 0x05, 0x01 },
    {/* 1*/ DATA_NULL_Set0, 0               , 0             , 0                       , 0   , 0    },
```

Module一定要设对，否则会调用错时序！
有些器件如sky779XX，又是ASM又是PA。在RX的Event里面就把它定义成MIPI_ASM，在TX Event里面就把他定义成MIPI_PA。

```
#define TDD_ASM_USID_BAND34_Set0   (MIPI_USID_ASM0_Set0 <<3)
#define TDD_ASM_USID_BAND39_Set0   (MIPI_USID_ASM0_Set0 <<3)
#define TDD_PA_USID_BAND34_Set0    (MIPI_USID_ASM0_Set0 <<3)
#define TDD_PA_USID_BAND39_Set0    (MIPI_USID_ASM0_Set0 <<3)
#define TDD_ETM_USID_Set0          (0 << 3)//no use
```

```
/*Band39 Rx Off Data*/
{
  /*          MODULE    ,       PORT      ,  OPERATION ,       USID              , ADDR, DATA*/
  {/* 0*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0 , 0x05, 0x00 },
  {/* 1*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0 , 0x23, 0x80 },
```

Note： Ex:Sky77928-11,Off Event Sky77928-11,2G/TD/LTE TD需要SW Reset 0x23,0x80;

TDSCDMA--TX on

- TX On Table element type 必须设置为TDD_PA_Set*,I Bias 取自Vendor提供的参考List

```
/*Band39 Tx On Data*/
{
  /*          MODULE    ,       PORT       ,    OPERATION ,       USID              , ADDR, DATA*/
  {/* 0*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x00, PA_FLAG_Set0 },
  {/* 1*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0  , 0x04, 0x10 },
  {/* 2*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0  , 0x05, 0x0F },
```

ASM On

```
/*PA Mode Data for BAND39*/
/*Data Num Do Not Exceed 8 for Each Scenario*/
{
  {
    /*          MODULE    ,       PORT       ,    OPERATION ,       USID              , ADDR, DATA*/
    {/* 0*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x00, 0x8F },//PA HIGH MODE
    {/* 1*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x01, 0x8E },//PA HIGH MODE
    {/* 2*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x04, 0x10 },//PA HIGH MODE
    {/* 3*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x02, 0x00 },//PA HIGH MODE
    {/* 4*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    },//PA HIGH MODE
    {/* 5*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    },//PA HIGH MODE
    {/* 6*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    }//PA HIGH MODE
  },
  {
    /*          MODULE    ,       PORT       ,    OPERATION ,       USID              , ADDR, DATA*/
    {/* 0*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x00, 0xAF },//PA MIDDLE MODE
    {/* 1*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x01, 0x28 },//PA MIDDLE MODE
    {/* 2*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x04, 0x10 },//PA MIDDLE MODE
    {/* 3*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x02, 0x00 },//PA MIDDLE MODE
    {/* 4*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    },//PA MIDDLE MODE
    {/* 5*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    },//PA MIDDLE MODE
    {/* 6*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    }//PA MIDDLE MODE
  },
  {
    /*          MODULE    ,       PORT       ,    OPERATION ,       USID              , ADDR, DATA*/
    {/* 0*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x00, 0xAF },//PA MIDDLE MODE
    {/* 1*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x01, 0x28 },//PA MIDDLE MODE
    {/* 2*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x04, 0x10 },//PA MIDDLE MODE
    {/* 3*/ TDD_PA_Set0   , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_PA_USID_BAND39_Set0   , 0x02, 0x00 },//PA MIDDLE MODE
    {/* 4*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    },//PA MIDDLE MODE
    {/* 5*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    },//PA MIDDLE MODE
    {/* 6*/ DATA_NULL_Set0, 0                , 0             , 0                       , 0   , 0    }//PA MIDDLE MODE
  }
},
```

Max data num =8

H Mode

M Mode

L Mode

TDSCDMA--TX off

```
/*Band39 Tx Off Data*/
{
   /*          MODULE    ,      PORT       ,    OPERATION ,      USID                  , ADDR, DATA*/
   {/* 0*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0  , 0x00, 0x00 },
   {/* 1*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0  , 0x05, 0x00 },
   {/* 2*/ TDD_ASM_Set0  , MIPI_PORT0_Set0 , TDSCDMA_REG_W , TDD_ASM_USID_BAND39_Set0  , 0x23, 0x80 },
```

Note： Ex:Sky77928-11,Off Event Sky77928-11,2G/TD/LTE TD需要SW Reset 0x23,0x80;

CONFIDENTIAL B

WCDMA Part

WCDMA--RX

Event No.	Event Type	Action	Data	Data No.
1	RX ON	设置到active状态	打开PRX ASM TRX Path	0
			打开DRX ASM TRX Path	1
2	RX OFF	设置到Idle状态	PRX ASM Idle mode	2-3
			DRX ASM Idle mode	4

```
const UL1_MIPI_EVENT_TABLE_T UMTS_MIPI_RX_EVENT_UMTSBand1_Set0[UL1_MIPI_RX_EVENT_NUM] =
{
  /* No.    elm type , data idx     , evt_type      , evt_offset    */
  /*              { start, stop },            ( us )      */
  { /* 0  */ UL1_MIPI_ASM      , { 0   , 0   }, UL1_MIPI_TRX_ON, US2CHIPCNT(200)   },
  { /* 1  */ UL1_MIPI_ASM_RXD, { 1   , 1   }, UL1_MIPI_TRX_ON, US2CHIPCNT(200)   },
  { /* 2  */ UL1_MIPI_ASM      , { 2   , 3   }, UL1_MIPI_TRX_OFF, US2CHIPCNT(10)   },
  { /* 3  */ UL1_MIPI_ASM_RXD, { 4   , 4   }, UL1_MIPI_TRX_OFF, US2CHIPCNT(10)   },
```

Timing fixed!

```
const UL1_MIPI_DATA_SUBBAND_TABLE_T UMTS_MIPI_RX_DATA_UMTSBand1_Set0[UL1_MIPI_RX_DATA_NUM] =
{
  //No.    elm type , port_sel     , data_seq  , user id,      { { sub 0 ,addr, data }},{ sub 1 ,addr, data }},{ sub 2 ,addr, data}}, { sub 3, addr, d
  { /* 0  */ UL1_MIPI_ASM,      UL1_MIPI_PORT0, UL1_REG_W    , MIPI_USID_ASM0_Set0, { { 21124 ,{0x05, 0x05}}, { 21262 ,{0x05, 0x05}}
  { /* 1  */ UL1_MIPI_ASM_RXD, UL1_MIPI_PORT2, UL1_REG_W    , MIPI_USID_ASM1_Set0, { { 21124 ,{0x00, 0x04}}, { 21262 ,{0x00, 0x04}}
  { /* 2  */ UL1_MIPI_ASM,      UL1_MIPI_PORT0, UL1_REG_W    , MIPI_USID_ASM0_Set0, { { 21124 ,{0x05, 0x00}}, { 21262 ,{0x05, 0x00}}
  { /* 3  */ UL1_MIPI_ASM,      UL1_MIPI_PORT0, UL1_REG_W    , MIPI_USID_ASM0_Set0, { { 21124 ,{0x04, 0x00}}, { 21262 ,{0x04, 0x00}}
  { /* 4  */ UL1_MIPI_ASM_RXD, UL1_MIPI_PORT2, UL1_REG_W    , MIPI_USID_ASM1_Set0, { { 21124 ,{0x00, 0x00}}, { 21262 ,{0x00, 0x00}}
```

每个subband都要修改！

WCDMA--TX

Event No.	Event Type	Action	Data	Data No.
1	Tx OFF	PA idle 状态	设置寄存器0x00 idle mode	0
			设置寄存器0x01 idle mode	1
			设置寄存器0x02 idle mode	2
			设置寄存器0x03 idle mode	3
2	TX On	2G TX Port Off	设置寄存器0x00 TX Off mode	4
		H TRX Port off	H TRX Port Off mode	5
		L TRX Port off	L TRX Port Off mode	6
		M TRX Port active	设置寄存器0x05 on mode	7
		Tx Coupler Active	设置寄存器0x04 Tx Coupler	8

WCDMA——TX

```
const UL1_MIPI_EVENT_TABLE_T UMTS_MIPI_TX_EVENT_UMTSBand1_Set0[UL1_MIPI_TX_EVENT_NUM] =
{
      /* No.     elm type , data idx      , evt_type      , evt_offset    */
   /*                 { start, stop },            ( us )       */
   { /* 0  */ UL1_MIPI_PA,   { 0    , 3    }, UL1_MIPI_TRX_OFF,   US2CHIPCNT(10)  },
   { /* 1  */ UL1_MIPI_ASM,  { 4    , 8    }, UL1_MIPI_TRX_ON,    US2CHIPCNT(200) },
```

Timing Fixed!

```
const UL1_MIPI_DATA_SUBBAND_TABLE_T UMTS_MIPI_TX_DATA_UMTSBand1_Set0[UL1_MIPI_TX_DATA_NUM] =
{
  //No.     elm type , port_sel      , data_seq  , user id,       { { sub 0 ,addr, data }},{ sub 1 ,addr, data }},{ sub 2 ,addr, data}}, { sub 3, add
  { /* 0 */ UL1_MIPI_PA  , UL1_MIPI_PORT1, UL1_REG_W  ,   MIPI_USID_PA0_Set0    ,{ { 19224 ,{0x00, 0x00}}, { 19362 ,{0x00, 0x00}},
  { /* 1 */ UL1_MIPI_PA  , UL1_MIPI_PORT1, UL1_REG_W  ,   MIPI_USID_PA0_Set0    ,{ { 19224 ,{0x01, 0x00}}, { 19362 ,{0x01, 0x00}},
  { /* 2 */ UL1_MIPI_PA  , UL1_MIPI_PORT1, UL1_REG_W  ,   MIPI_USID_PA0_Set0    ,{ { 19224 ,{0x02, 0x00}}, { 19362 ,{0x02, 0x00}},
  { /* 3 */ UL1_MIPI_PA  , UL1_MIPI_PORT1, UL1_REG_W  ,   MIPI_USID_PA0_Set0    ,{ { 19224 ,{0x03, 0x00}}, { 19362 ,{0x03, 0x00}},
  { /* 4 */ UL1_MIPI_ASM , UL1_MIPI_PORT0, UL1_REG_W  ,   MIPI_USID_ASM0_Set0   ,{ { 19224 ,{0x00, 0x80}}, { 19362 ,{0x00, 0x80}},
  { /* 5 */ UL1_MIPI_ASM , UL1_MIPI_PORT0, UL1_REG_W  ,   MIPI_USID_ASM0_Set0   ,{ { 19224 ,{0x02, 0x00}}, { 19362 ,{0x02, 0x00}},
  { /* 6 */ UL1_MIPI_ASM , UL1_MIPI_PORT0, UL1_REG_W  ,   MIPI_USID_ASM0_Set0   ,{ { 19224 ,{0x03, 0x00}}, { 19362 ,{0x03, 0x00}},
  { /* 7 */ UL1_MIPI_ASM , UL1_MIPI_PORT0, UL1_REG_W  ,   MIPI_USID_ASM0_Set0   ,{ { 19224 ,{0x05, 0x05}}, { 19362 ,{0x05, 0x05}},
  { /* 8 */ UL1_MIPI_ASM , UL1_MIPI_PORT0, UL1_REG_W  ,   MIPI_USID_ASM0_Set0   ,{ { 19224 ,{0x04, 0x10}}, { 19362 ,{0x04, 0x10}},
```

Data

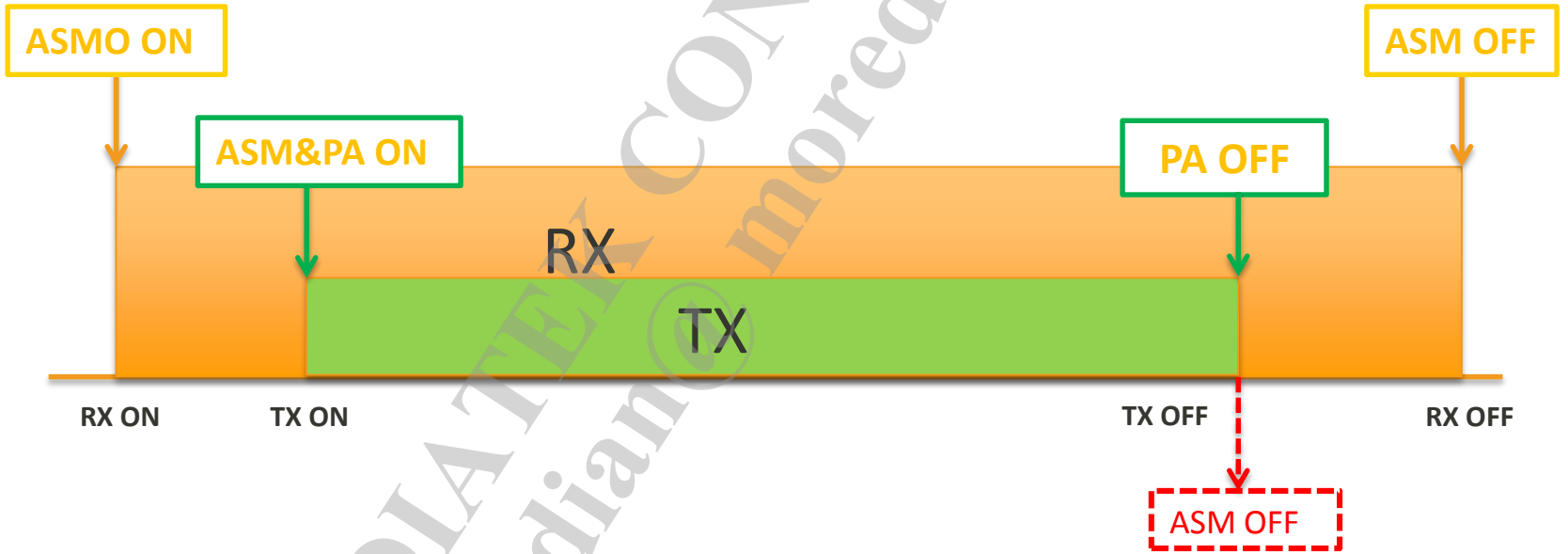
Note:

- WCDMA TX部分不用配置PA ON，PA ON的data都在TPC段配置。这样可以减少几笔data，改善某些test case的margin。
- 由TX Event的定义可以看出，data0~3是设成PA OFF的。如果这个地方错填了PA ON的值，会导致WCDMA待机电流居高不下。

WCDMA--TX

- 为什么WCDMA TX没有设ASM OFF?

```
const UL1_MIPI_EVENT_TABLE_T UMTS_MIPI_TX_EVENT_UMTSBand1_Set0[UL1_MIPI_TX_EVENT_NUM] =
{
    /* No.    elm type , data idx      , evt_type      , evt_offset    */
  /*                 { start, stop },              ( us )          */
  { /* 0  */ UL1_MIPI_PA,   { 0   , 3   }, UL1_MIPI_TRX_OFF,   US2CHIPCNT(10)  },
  { /* 1  */ UL1_MIPI_ASM,  { 4   , 8   }, UL1_MIPI_TRX_ON,    US2CHIPCNT(200) },
```



- 假设在TX OFF的时候就关闭ASM，那么RX通路也会被同时关闭，在某些场景下会失去网络链接!

WCDMA--TPC

```
const UL1_UMTS_MIPI_TPC_T UMTS_MIPI_TPC_UMTSBand1_Set0 =
{
  /* Event */
  {
  /* No.        elm type , data idx      , evt_type      , evt_offset    */
  /*                    { start, stop },            ( us )       */
   { /* 0, Prf_7 */ UL1_MIPI_PA, { 0   , 3    }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) }, /*highest power*/
   { /* 1, Prf_6 */ UL1_MIPI_PA, { 4   , 7    }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) },
   { /* 2, Prf_5 */ UL1_MIPI_PA, { 8   , 11   }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) },
   { /* 3, Prf_4 */ UL1_MIPI_PA, { 12  , 15   }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) },
   { /* 4, Prf_3 */ UL1_MIPI_PA, { 16  , 19   }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) },
   { /* 5, Prf_2 */ UL1_MIPI_PA, { 20  , 23   }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) },
   { /* 6, Prf_1 */ UL1_MIPI_PA, { 24  , 27   }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) },
   { /* 7, Prf_0 */ UL1_MIPI_PA, { 28  , 31   }, UL1_MIPI_TPC_SET  , US2CHIPCNT(20) }, /*lowest power*/
  },
```

PA的状态都由TPC设置段决定

按从上而下的顺序，WCDMA TX功率由高到低！每一档位的设定请结合校准使用的ini文件，参考Vendor给的值。

```
{
//elm type  , port_sel      , data_seq, user id,      {{sub 0, addr, data}}, {{sub 1, addr, data}}, {{sub 2, addr, data} , {{sub 3, addr, data}}, {{sub 4, addr, data}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0xCA}}, {19362, {0x01, 0xCB}}, {19500, {0x01, 0xCB}}, {19638, {0x01, 0xCC}}, {19776, {0x01, 0xCC}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x8B}}, {19362, {0x03, 0x8B}}, {19500, {0x03, 0x8B}}, {19638, {0x03, 0x8B}}, {19776, {0x03, 0x8B}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x47}}, {19362, {0x01, 0x47}}, {19500, {0x01, 0x47}}, {19638, {0x01, 0x47}}, {19776, {0x01, 0x47}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x46}}, {19362, {0x01, 0x46}}, {19500, {0x01, 0x46}}, {19638, {0x01, 0x46}}, {19776, {0x01, 0x46}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x36}}, {19362, {0x01, 0x36}}, {19500, {0x01, 0x36}}, {19638, {0x01, 0x36}}, {19776, {0x01, 0x36}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x35}}, {19362, {0x01, 0x35}}, {19500, {0x01, 0x35}}, {19638, {0x01, 0x35}}, {19776, {0x01, 0x35}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x25}}, {19362, {0x01, 0x25}}, {19500, {0x01, 0x25}}, {19638, {0x01, 0x25}}, {19776, {0x01, 0x25}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x16}}, {19362, {0x01, 0x16}}, {19500, {0x01, 0x16}}, {19638, {0x01, 0x16}}, {19776, {0x01, 0x16}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x00, 0x1C}}, {19362, {0x00, 0x1C}}, {19500, {0x00, 0x1C}}, {19638, {0x00, 0x1C}}, {19776, {0x00, 0x1C}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x01, 0x15}}, {19362, {0x01, 0x15}}, {19500, {0x01, 0x15}}, {19638, {0x01, 0x15}}, {19776, {0x01, 0x15}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x02, 0x20}}, {19362, {0x02, 0x20}}, {19500, {0x02, 0x20}}, {19638, {0x02, 0x20}}, {19776, {0x02, 0x20}}}
 {UL1_MIPI_PA  , UL1_MIPI_PORT1 , UL1_REG_W ,  MIPI_USID_PA0_Set0,  {{19224, {0x03, 0x86}}, {19362, {0x03, 0x86}}, {19500, {0x03, 0x86}}, {19638, {0x03, 0x86}}, {19776, {0x03, 0x86}}}
```

CONFIDENTIAL B

LTE Part

LTE--RX

➢ LTE FDD Band RX通路ON/OFF时所需的操作步骤

Event No.	Event Type	Action	Data	Data No.
0	RX ON	设置到Idle状态	将PRX ASM设成active mode	0
			打开DRX ASM对应的TRX Path	1
1	RX OFF	设置到Idle状态	将PRX ASM设成idle mode	2
			将DRX ASM设成low power mode	3

Timing在lte_custom_mipi.h中定义

```
/*MIPI ASM    */
/*FDD RX ON   */
#define LTE_FDD_MIPI_ASM_RX_ON0_Set0   US2OFFCNT(15)
#define LTE_FDD_MIPI_ASM_RX_ON1_Set0   US2OFFCNT(0)

/*MIPI ASM    */
/*FDD RX OFF  */
#define LTE_FDD_MIPI_ASM_RX_OFF0_Set0  US2OFFCNT(5)
#define LTE_FDD_MIPI_ASM_RX_OFF1_Set0  US2OFFCNT(0)
```

```
LTE_MIPI_EVENT_TABLE_T LTE_Band1_MIPI_RX_EVENT_Set0[] =
{
  /* No.     elm type     , data idx      , evt_type      , evt_offset          */
  /*                  { start, stop },          ( us )      */
  { /* 0 */ LTE_MIPI_ASM , { 0   , 1   }, LTE_MIPI_TRX_ON  , LTE_FDD_MIPI_ASM_RX_ON0_Set0 },
  { /* 1 */ LTE_MIPI_ASM , { 2   , 3   }, LTE_MIPI_TRX_OFF , LTE_FDD_MIPI_ASM_RX_OFF0_Set0},
  { /* 2 */ LTE_MIPI_NULL, { 0   , 0   }, LTE_MIPI_EVENT_NULL, 0                        },
};
```

Event

Timing

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band1_MIPI_RX_DATA_Set0[] =
{
  //No.     elm type     , port_sel      , data_seq     , USID                ,{ { subband-0 freq   ,addr ,data }, { subband-1 freq
  { /* 0 */ LTE_MIPI_ASM , LTE_MIPI_PORT0 , LTE_REG_W   , MIPI_USID_ASM0_Set0 ,{ { 21100 /*100 kHz*/ ,0x05 ,0x05 },
  { /* 1 */ LTE_MIPI_ASM , LTE_MIPI_PORT2 , LTE_REG_W   , MIPI_USID_ASM1_Set0 ,{ { 21100 /*100 kHz*/ ,0x00 ,0x04 },
  { /* 2 */ LTE_MIPI_ASM , LTE_MIPI_PORT0 , LTE_REG_W   , MIPI_USID_ASM0_Set0 ,{ { 21100 /*100 kHz*/ ,0x05 ,0x00 },
  { /* 3 */ LTE_MIPI_ASM , LTE_MIPI_PORT2 , LTE_REG_W   , MIPI_USID_ASM1_Set0 ,{ { 21100 /*100 kHz*/ ,0x00 ,0x00 },
  { /* 4 */ LTE_MIPI_NULL, 0              , 0           ,                       ,{ { 0     /*100 kHz*/ ,0    ,0    }, { 0     /*100 kHz*/ ,0
};
```

Data

LTE--RX

LTE TDD Band [B38/40/41]

Event No.	Event Type	Action	Data	Data No.
0	RX ON	设置到ASM Active状态	打开PRX ASM对应的TRX Path	0
			打开DRX ASM对应的TRX Path	1
1	RX ON	设置到3P4T Active状态	将PA 3P4T 切换成TX output	2
2	RX OFF	设置到Idle状态	将 PRX ASM设成idle mode	3
			将DRX ASM设成idle mode	4
			将MMM PA 3P4T开关关闭	5

```
/*MIPI ASM    */
/*TDD RX ON   */
#define LTE_TDD_MIPI_ASM_RX_ON0_Set0  US2OFFCNT(15)
#define LTE_TDD_MIPI_ASM_RX_ON1_Set0  US2OFFCNT(0)

/*MIPI ASM    */
/*TDD RX OFF  */
#define LTE_TDD_MIPI_ASM_RX_OFF0_Set0 US2OFFCNT(5)
#define LTE_TDD_MIPI_ASM_RX_OFF1_Set0 US2OFFCNT(0)
```

Timing

```
LTE_MIPI_EVENT_TABLE_T LTE_Band40_MIPI_RX_EVENT_Set0[] =
{
  /* No.    elm type     , data idx      , evt_type          , evt_offset
  /*                  { start, stop },              ( us )Event      */
  { /* 0 */ LTE_MIPI_ASM , { 0   , 2 }, LTE_MIPI_TRX_ON    , LTE_TDD_MIPI_ASM_RX_ON0_Set0  },
  { /* 1 */ LTE_MIPI_ASM , { 3   , 5 }, LTE_MIPI_TRX_OFF   , LTE_TDD_MIPI_ASM_RX_OFF0_Set0 },
  { /* 2 */ LTE_MIPI_NULL, { 0   , 0 }, LTE_MIPI_EVENT_NULL, 0                             },
};
```

Data

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band40_MIPI_RX_DATA_Set0[] =
{
  //No.    elm type     , port_sel       , data_seq    , USID                  ,{ { subband-0 freq   ,addr ,data }, { subband-1 freq
  { /* 0 */ LTE_MIPI_ASM , LTE_MIPI_PORT0 , LTE_REG_W , MIPI_USID_ASM0_Set0   ,{ { 23000 /*100 kHz*/ ,0x03 ,0x40 }, {
  { /* 1 */ LTE_MIPI_ASM , LTE_MIPI_PORT2 , LTE_REG_W , MIPI_USID_ASM2_Set0   ,{ { 23000 /*100 kHz*/ ,0x00 ,0x08 }, {
  { /* 2 */ LTE_MIPI_ASM , LTE_MIPI_PORT1 , LTE_REG_W , MIPI_USID_PA0_Set0    ,{ { 23000 /*100 kHz*/ ,0x02 ,0x02 }, {
  { /* 3 */ LTE_MIPI_ASM , LTE_MIPI_PORT0 , LTE_REG_W , MIPI_USID_ASM0_Set0   ,{ { 23000 /*100 kHz*/ ,0x03 ,0x00 }, {
  { /* 4 */ LTE_MIPI_ASM , LTE_MIPI_PORT2 , LTE_REG_W , MIPI_USID_ASM2_Set0   ,{ { 23000 /*100 kHz*/ ,0x00 ,0x00 }, {
  { /* 5 */ LTE_MIPI_ASM , LTE_MIPI_PORT1 , LTE_REG_W , MIPI_USID_PA0_Set0    ,{ { 23000 /*100 kHz*/ ,0x02 ,0x00 },
  { /* 6 */ LTE_MIPI_NULL, 0              , 0         , 0                     ,{ {    0 /*100 kHz*/ ,0   ,0    }, {    0 /*100 kHz*/ ,0
};
```

LTE--TX

- 列出TX通路ON/OFF时所需的操作步骤：

Event No.	Event Type	Action	Data	Data No.
0	TX ON	Enable PA	设定PA寄存器0	0
			设定PA寄存器1	1
			设定PA寄存器4	2
			设定TXM coupler 寄存器0x04	3
1	TX OFF	关闭PA	设定PA寄存器0	4
		关闭 TXM TX Coupler	设定TXM Tx Coupler off	5
2	TX ON	打开ASM	将ASM切到TRX Port	6

Note: 和WCDMA一样，FDD-LTE在TX OFF的时候不能关闭ASM，但是TDD-LTE在TX OFF的时候可以关闭ASM。目前设定都是TX不去关ASM，都由RX去关ASM。

LTE--TX

- LTE FDD Band

Timing在lte_custom_mipi.h中定义

```
/*MIPI ASM    */
/*FDD TX ON   */
#define LTE_FDD_MIPI_ASM_TX_ON0_Set0   US2OFFCNT(15)
#define LTE_FDD_MIPI_ASM_TX_ON1_Set0   US2OFFCNT(0)

/*MIPI PA     */
/*FDD TX ON   */
#define LTE_FDD_MIPI_PA_TX_ON0_Set0    US2OFFCNT(25)
#define LTE_FDD_MIPI_PA_TX_ON1_Set0    US2OFFCNT(0)

/*MIPI PA     */
/*FDD TX OFF  */
#define LTE_FDD_MIPI_PA_TX_OFF0_Set0   US2OFFCNT(20)
#define LTE_FDD_MIPI_PA_TX_OFF1_Set0   US2OFFCNT(0)
```

Timing

Event

```
LTE_MIPI_EVENT_TABLE_T LTE_Band1_MIPI_TX_EVENT_Set0[] =
{
  /* No.    elm type     , data idx      , evt_type          , evt_offset      */
  /*                  { start, stop },              ( us )                    */
  { /* 0 */ LTE_MIPI_PA   , { 0   , 3   }, LTE_MIPI_TRX_ON  , LTE_FDD_MIPI_PA_TX_ON0_Set0  }, //PA On
  { /* 1 */ LTE_MIPI_PA   , { 4   , 5   }, LTE_MIPI_TRX_OFF , LTE_FDD_MIPI_PA_TX_OFF0_Set0 }, //PA Off
  { /* 2 */ LTE_MIPI_ASM  , { 6   , 6   }, LTE_MIPI_TRX_ON  , LTE_FDD_MIPI_ASM_TX_ON0_Set0 }, //WIN on
  { /* 3 */ LTE_MIPI_NULL , { 0   , 0   }, LTE_MIPI_EVENT_NULL, 0                          },
};
```

Data

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band1_MIPI_TX_DATA_Set0[] =
{
  //No.    elm type     , port_sel       , data_seq     , USID                ,{ { subband-0 freq   ,addr ,data }, { subband-1 freq
  { /* 0 */ LTE_MIPI_PA   , LTE_MIPI_PORT1 , LTE_REG_W , MIPI_USID_PA0_Set0  ,{ { 19200 /*100 kHz*/  ,0x00 , 0x0C}, {
  { /* 1 */ LTE_MIPI_PA   , LTE_MIPI_PORT1 , LTE_REG_W , MIPI_USID_PA0_Set0  ,{ { 19200 /*100 kHz*/  ,0x01 , 0x00}, {
  { /* 2 */ LTE_MIPI_PA   , LTE_MIPI_PORT1 , LTE_REG_W , MIPI_USID_PA0_Set0  ,{ { 19200 /*100 kHz*/  ,0x03 , 0x00}, {
  { /* 3 */ LTE_MIPI_PA   , LTE_MIPI_PORT0 , LTE_REG_W , MIPI_USID_ASM0_Set0 ,{ { 19200 /*100 kHz*/  ,0x04 , 0x10}, {
  { /* 4 */ LTE_MIPI_PA   , LTE_MIPI_PORT1 , LTE_REG_W , MIPI_USID_PA0_Set0  ,{ { 19200 /*100 kHz*/  ,0x00 , 0x00}, {
  { /* 5 */ LTE_MIPI_PA   , LTE_MIPI_PORT0 , LTE_REG_W , MIPI_USID_ASM0_Set0 ,{ { 19200 /*100 kHz*/  ,0x04 , 0x00}, {
  { /* 6 */ LTE_MIPI_ASM  , LTE_MIPI_PORT0 , LTE_REG_W , MIPI_USID_ASM0_Set0 ,{ { 19200 /*100 kHz*/  ,0x05 , 0x05}, {
  { /* 7 */ LTE_MIPI_NULL , 0               , 0         ,                     ,{ { 0     /*100 kHz*/  ,0    , 0    }, { 0     /*100 kHz*/
};
```

LTE--TX

LTE TDD Band

TX通路ON/OFF时所需的操作步骤：

Event No.	Event Type	Action	Data	Data No.
0	TX ON	Enable PA	设定PA寄存器0	0
			设定PA寄存器1	1
			设定PA寄存器2	2
			设定PA寄存器3	3
		打开TXM Coupler	设定TXM寄存器4	4
		DRX ASM Idle	设定DRX ASM Idle	5
1	TX OFF	关闭PA	设定PA寄存器0	6
		关闭PA 3P4T	设定PA寄存器4	7
2	TX ON	打开ASM	将ASM TXM 切到TRX Port	8

Note: 和WCDMA一样，FDD-LTE在TX OFF的时候不能关闭ASM，但是TDD-LTE在TX OFF的时候可以关闭ASM。目前设定都是TX不去关ASM，都由RX去关ASM。

LTE--TX

- LTE TDD

Timing在lte_custom_mipi.h中定义

```
/*MIPI ASM    */
/*TDD TX ON    */
#define LTE_TDD_MIPI_ASM_TX_ON0_Set0   US2OFFCNT(15)
#define LTE_TDD_MIPI_ASM_TX_ON1_Set0   US2OFFCNT(0)

/*MIPI PA    */
/*TDD TX ON    */
#define LTE_TDD_MIPI_PA_TX_ON0_Set0    US2OFFCNT(25)
#define LTE_TDD_MIPI_PA_TX_ON1_Set0    US2OFFCNT(0)

/*MIPI PA    */
/*TDD TX OFF    */
#define LTE_TDD_MIPI_PA_TX_OFF0_Set0   US2OFFCNT(20)
#define LTE_TDD_MIPI_PA_TX_OFF1_Set0   US2OFFCNT(0)
```

Timing

```
LTE_MIPI_EVENT_TABLE_T LTE_Band40_MIPI_TX_EVENT_Set0[] =
{
  /* No.     elm type     , data idx      , evt_type          , evt_offset          */
  /*                  { start, stop },                ( us ) Event      */
  { /* 0  */ LTE_MIPI_PA  ,{ 0   , 5 }, LTE_MIPI_TRX_ON  , LTE_TDD_MIPI_PA_TX_ON0_Set0  }, //PA On
  { /* 1  */ LTE_MIPI_PA  ,{ 6   , 7 }, LTE_MIPI_TRX_OFF , LTE_TDD_MIPI_PA_TX_OFF0_Set0 }, //PA Off
  { /* 2  */ LTE_MIPI_ASM ,{ 8   , 8 }, LTE_MIPI_TRX_ON  , LTE_TDD_MIPI_ASM_TX_ON0_Set0 }, //TDD ASM at Tx on
  { /* 3  */ LTE_MIPI_NULL,{ 0   , 0 }, LTE_MIPI_EVENT_NULL, 0                          },
};
```

Data

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band40_MIPI_TX_DATA_Set0[] =
{
  //No.     elm type     , port_sel       , data_seq    , USID                ,{ { subband-0 freq   ,addr ,data }, { subband-1 freq
  { /* 0  */ LTE_MIPI_PA  , LTE_MIPI_PORT1 , LTE_REG_W   , MIPI_USID_PA0_Set0   ,{ { 23000 /*100 kHz*/  ,0x02 , 0x00}, {
  { /* 1  */ LTE_MIPI_PA  , LTE_MIPI_PORT1 , LTE_REG_W   , MIPI_USID_PA0_Set0   ,{ { 23000 /*100 kHz*/  ,0x00 , 0x44}, {
  { /* 2  */ LTE_MIPI_PA  , LTE_MIPI_PORT1 , LTE_REG_W   , MIPI_USID_PA0_Set0   ,{ { 23000 /*100 kHz*/  ,0x01 , 0x00}, {
  { /* 3  */ LTE_MIPI_PA  , LTE_MIPI_PORT1 , LTE_REG_W   , MIPI_USID_PA0_Set0   ,{ { 23000 /*100 kHz*/  ,0x03 , 0x00}, {
  { /* 4  */ LTE_MIPI_PA  , LTE_MIPI_PORT0 , LTE_REG_W   , MIPI_USID_ASM0_Set0  ,{ { 23000 /*100 kHz*/  ,0x04 , 0x18},
  { /* 5  */ LTE_MIPI_PA  , LTE_MIPI_PORT2 , LTE_REG_W   , MIPI_USID_ASM2_Set0  ,{ { 23000 /*100 kHz*/  ,0x00 , 0x00},
  { /* 6  */ LTE_MIPI_PA  , LTE_MIPI_PORT1 , LTE_REG_W   , MIPI_USID_PA0_Set0   ,{ { 23000 /*100 kHz*/  ,0x00 , 0x00}, {
  { /* 7  */ LTE_MIPI_PA  , LTE_MIPI_PORT0 , LTE_REG_W   , MIPI_USID_ASM0_Set0  ,{ { 23000 /*100 kHz*/  ,0x04 , 0x00},
  { /* 8  */ LTE_MIPI_ASM , LTE_MIPI_PORT0 , LTE_REG_W   , MIPI_USID_ASM0_Set0  ,{ { 23000 /*100 kHz*/  ,0x03 , 0x40}
  //{ /* 7  */ LTE_MIPI_ASM , LTE_MIPI_PORT0 , LTE_REG_W   , MIPI_USID_ASM0_Set0  ,{ { 23000 /*100 kHz*/  ,0x03 , 0x0
  { /* 8  */ LTE_MIPI_NULL, 0              , 0           ,{ { 0     /*100 kHz*/  ,0    , 0  }, { 0     /*100 kHz*/ ,0
};
```

LTE--TPC

```
LTE_MIPI_EVENT_TABLE_T LTE_Band1_MIPI_TPC_EVENT_Set0[] =
{
  /* No.    elm type     , data idx      , evt_type          , evt_offset   */
  /*                    { start, stop },                 ( us )     */
  { /* 0 */ LTE_MIPI_PA  , { 0    , 2   }, LTE_MIPI_TPC_SET  , US2OFFCNT(10) },
  { /* 1 */ LTE_MIPI_NULL, { 0    , 0   }, LTE_MIPI_EVENT_NULL, 0           },
};
```

TPC的timing必须晚于PA ON!

Timing

```
LTE_MIPI_DATA_TABLE_T LTE_Band1_MIPI_TPC_DATA_Set0[] =
{
  //No.    elm type      , port_sel      , data_seq   , USID                      , address                    , data
  {/* 0 */ LTE_MIPI_PA_SEC, LTE_MIPI_PORT1  , LTE_REG_W  , LTE_MIPI_PA_SECTION_USID  , LTE_MIPI_PA_SECTION_ADDRESS , LTE_MIPI_PA_SECTION_DATA0},
  {/* 1 */ LTE_MIPI_PA_SEC, LTE_MIPI_PORT1  , LTE_REG_W  , LTE_MIPI_PA_SECTION_USID  , LTE_MIPI_PA_SECTION_ADDRESS , LTE_MIPI_PA_SECTION_DATA1},
  {/* 2 */ LTE_MIPI_PA_SEC, LTE_MIPI_PORT1  , LTE_REG_W  , LTE_MIPI_PA_SECTION_USID  , LTE_MIPI_PA_SECTION_ADDRESS , LTE_MIPI_PA_SECTION_DATA2},
  {/* 3 */ LTE_MIPI_NULL , 0          , 0       , 0                       , 0                          },
};
```

Data

```
LTE_MIPI_TPC_SECTION_TABLE_T LTE_Band1_MIPI_PA_SECTION_DATA_Set0[] =
{
  {
    19200, /*100kHz*/
    MIPI_USID_PA0_Set0, /*USID*/
    {
      // PAEn=1
      // PA_SEC_DATA0, PA_SEC_DATA1, PA_SEC_DATA2, PA_SEC_DATA3, PA_SEC_DATA4
      // {addr, data}, {addr, data}, {addr, data}, {addr, data}, {addr, data}
      {{ { 0x1 , 0x15}, { 0x3 , 0x86} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x25}, { 0x3 , 0x86} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x37}, { 0x3 , 0x86} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x66}, { 0x3 , 0x86} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x87}, { 0x3 , 0x86} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x99}, { 0x3 , 0x86} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x9B}, { 0x3 , 0x8B} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x9B}, { 0x3 , 0x8B} ,{ 0x0 , 0x1C} , { 0x0 , 0x0} , { 0x0 , 0x0}}},
      {{ { 0x1 , 0x00}, { 0x3 , 0x00} ,{ 0x0 , 0x00} , { 0x0 , 0x0} , { 0x0 , 0x0}}},//SRS
    }
  },
```

共9个档位，功率从低到高，最后一档设为PA OFF。

LTE--B28

lte_custom_rf.h LTE B28 TRX Port BPI SET

```
#define  BAND_INDICATOR12_Set0                  LTE_Band28
#define  BAND_INDICATOR12_SUPPORT_Set0          SW_CAPABILITY_SUPPORT
#define  LTE_Band28_RX_IO_SEL_Set0              RX_IO_PRX
#define  LTE_Band28_RXD_IO_SEL_Set0             RX_IO_DRX5
#define  PDATA_LTE_Band28_PR1_Set0              0x00000000
#define  PDATA_LTE_Band28_PR2_Set0              0x00000000
#define  PDATA_LTE_Band28_PR3_Set0              LTE_PDATA_OFF
#define  LTE_Band28_RX_eLNAIDX_Set0             MML1_FE_ELNA_NONE
#define  LTE_Band28_RXD_eLNAIDX_Set0            MML1_FE_ELNA_NONE
#define  LTE_Band28_TX_IO_SEL_Set0              TX_IO_LB1
#define  PDATA_LTE_Band28_PT1_Set0              0x00000000
#define  PDATA_LTE_Band28_PT2_Set0              0x00000000
#define  PDATA_LTE_Band28_PT3_Set0              LTE_PDATA_OFF
```

LTE--B28

➢ lte_custom_rf.h LTE 28B TRX PORT BPI SET

```
/* Definition for the band splitting */
/*-----------------------------------*/
// How to set the band-splitting frequency?
// For example, the DL frequencies of the 1st and 2nd sub-bands are 758~780.4MHz and 780.5~802.9MHz, so we define
// BAND_SPLIT_INDICATOR1_DL_END1 as 7805 [ = (780.4+0.1)*10 ]
// BAND_SPLIT_INDICATOR1_DL_END2 as 8030 [ = (802.9+0.1)*10 ]

// [Note]
// 1. The unit for the frequency definition is 100kHz
// 2. BAND_SPLIT_INDICATOR1_DL_ENDn defines the end DL frequency of each part of splitting band PLUS 0.1 MHz
// 3. Since there are only TWO sub-bands, BAND_SPLIT_INDICATOR1_DL_END3 should be 0
// 4. The way to define the UL frequency is the same as the way DL does

// [Example]
// BAND_SPLIT_INDICATOR0          LTE_Band28
// BAND_SPLIT_INDICATOR0_NUM       2  // the num of part of splitting band
// BAND_SPLIT_INDICATOR0_Part1_DL_END  7805  // the end DL frequency of the 1st part of splitting band PLUS 0.1MHz, unit: 100KHz, set to 0 if no part is split
// BAND_SPLIT_INDICATOR0_Part2_DL_END  8030  // the end DL frequency of the 2nd part of splitting band PLUS 0.1MHz, unit: 100KHz, set to 0 if no part is split
// BAND_SPLIT_INDICATOR0_Part3_DL_END  0  // the end DL frequency of the 3rd part of splitting band PLUS 0.1MHz, unit: 100KHz, set to 0 if no part is split or only two parts are split
// BAND_SPLIT_INDICATOR0_Part1_UL_END  7255  // the end UL frequency of the 1st part of splitting band PLUS 0.1MHz, unit: 100KHz, set to 0 if no part is split
// BAND_SPLIT_INDICATOR0_Part2_UL_END  7480  // the end UL frequency of the 2nd part of splitting band PLUS 0.1MHz, unit: 100KHz, set to 0 if no part is split
// BAND_SPLIT_INDICATOR0_Part3_UL_END  0  // the end UL frequency of the 3rd part of splitting band PLUS 0.1MHz, unit: 100KHz, set to 0 if no part is split or only two parts are split


/*-------------------------*/
/* 0th Split Band Settings */
/*-------------------------*/
#define BAND_SPLIT_INDICATOR0_Set0              LTE_Band28
#define BAND_SPLIT_INDICATOR0_NUM_Set0          2
#define BAND_SPLIT_INDICATOR0_Part1_DL_END_Set0     7805
#define BAND_SPLIT_INDICATOR0_Part2_DL_END_Set0     8030
#define BAND_SPLIT_INDICATOR0_Part3_DL_END_Set0     0
#define BAND_SPLIT_INDICATOR0_Part1_UL_END_Set0     7255
#define BAND_SPLIT_INDICATOR0_Part2_UL_END_Set0     7480
#define BAND_SPLIT_INDICATOR0_Part3_UL_END_Set0     0
/*                         */
```

```
/* LTE_Band28-Part1 Split Band Settings */
/*-------------------------*/
#define LTE_Band28_Part1_RX_TBL_IDX_Set0            TBL_IDX12
#define LTE_Band28_Part1_TX_TBL_IDX_Set0            TBL_IDX12
/*-------------------------*/
/* LTE_Band28-Part2 Split Band Settings */
/*-------------------------*/
#define LTE_Band28_Part2_RX_TBL_IDX_Set0            TBL_IDX25
#define LTE_Band28_Part2_TX_TBL_IDX_Set0            TBL_IDX28
#define LTE_Band28_Part2_RX_IO_SEL_Set0             RX_IO_PRX5
#define LTE_Band28_Part2_RXD_IO_SEL_Set0            RX_IO_DRX5
#define LTE_Band28_Part2_TX_IO_SEL_Set0             TX_IO_LB1
#define PDATA_LTE_Band28_Part2_PR1_Set0             0x00000002
#define PDATA_LTE_Band28_Part2_PR2_Set0             0x00000002
#define PDATA_LTE_Band28_Part2_PR3_Set0             LTE_PDATA_OFF
#define PDATA_LTE_Band28_Part2_PT1_Set0             0x00000000
#define PDATA_LTE_Band28_Part2_PT2_Set0             0x00000000
#define PDATA_LTE_Band28_Part2_PT3_Set0             LTE_PDATA_OFF
#define PDATA_LTE_Band28_Part2_VM0_Set0             {0,0,0,0,0,0,0,0}
#define PDATA_LTE_Band28_Part2_VM1_Set0             {0,0,0,0,0,0,0,0}
/*-------------------------*/
/* LTE_Band28-Part3 Split Band Settings */
/*-------------------------*/
#define LTE_Band28_Part3_RX_TBL_IDX_Set0            TBL_IDX_INVALID
#define LTE_Band28_Part3_TX_TBL_IDX_Set0            TBL_IDX_INVALID
#define LTE_Band28_Part3_RX_IO_SEL_Set0             RX_IO_NON_USED
#define LTE_Band28_Part3_RXD_IO_SEL_Set0            RXD_IO_NON_USED
#define LTE_Band28_Part3_TX_IO_SEL_Set0             TX_IO_NON_USED
#define PDATA_LTE_Band28_Part3_PR1_Set0             0x00000000
#define PDATA_LTE_Band28_Part3_PR2_Set0             0x00000000
#define PDATA_LTE_Band28_Part3_PR3_Set0             LTE_PDATA_OFF
#define PDATA_LTE_Band28_Part3_PT1_Set0             0x00000000
#define PDATA_LTE_Band28_Part3_PT2_Set0             0x00000000
#define PDATA_LTE_Band28_Part3_PT3_Set0             LTE_PDATA_OFF
#define PDATA_LTE_Band28_Part3_VM0_Set0             {0,0,0,0,0,0,0,0}
#define PDATA_LTE_Band28_Part3_VM1_Set0             {0,0,0,0,0,0,0,0}
/*
```

LTE--B28

- lte_custom_mipi.c

```
LTE_MIPI_EVENT_TABLE_T LTE_Band28_MIPI_RX_EVENT_Set0[] =
{
  /* No.    elm type    , data idx      , evt_type         , evt_offset          */
  /*                  { start, stop },              ( us )            */
  { /* 0  */ LTE_MIPI_ASM , { 0   , 1   }, LTE_MIPI_TRX_ON   , LTE_FDD_MIPI_ASM_RX_ON0_Set0 },
  { /* 1  */ LTE_MIPI_ASM , { 2   , 3   }, LTE_MIPI_TRX_OFF  , LTE_FDD_MIPI_ASM_RX_OFF0_Set0},
  { /* 2  */ LTE_MIPI_NULL, { 0   , 0   }, LTE_MIPI_EVENT_NULL, 0                        },
};
```

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band28_MIPI_RX_DATA_Set0[] =
{
  //No.    elm type    , port_sel       , data_seq     , USID                ,{ { subband-0
  { /* 0  */ LTE_MIPI_ASM , LTE_MIPI_PORT0   , LTE_REG_W    , MIPI_USID_ASM0_Set0  ,
  { /* 1  */ LTE_MIPI_ASM , LTE_MIPI_PORT2   , LTE_REG_W    , MIPI_USID_ASM1_Set0  ,
  { /* 2  */ LTE_MIPI_ASM , LTE_MIPI_PORT0   , LTE_REG_W    , MIPI_USID_ASM0_Set0  ,
  { /* 3  */ LTE_MIPI_ASM , LTE_MIPI_PORT2   , LTE_REG_W    , MIPI_USID_ASM1_Set0  ,
  { /* 4  */ LTE_MIPI_NULL, 0           , 0         , 0                   ,{ {    0 /*100 kH
};
```

28A

28B

```
freq   ,addr ,data }, { subband-1 freq   ,addr ,data }, { subband-2 freq   ,addr ,data }, { subband-3 freq   ,addr ,data }, { subband-4 freq   ,addr ,data },
{ {  7580 /*100 kHz*/ ,0x02 ,0x05 }, {  7692 /*100 kHz*/ ,0x02 ,0x05 }, {  7805 /*100 kHz*/ ,0x02 ,0x04 }, {  7917 /*100 kHz*/ ,0x02 ,0x04 }, {  8030 /*100 kHz*/ ,0x02 ,0x04 },} },
{ {  7580 /*100 kHz*/ ,0x01 ,0x06 }, {  7692 /*100 kHz*/ ,0x01 ,0x06 }, {  7805 /*100 kHz*/ ,0x01 ,0x05 }, {  7917 /*100 kHz*/ ,0x01 ,0x05 }, {  8030 /*100 kHz*/ ,0x01 ,0x05 },} },
{ {  7580 /*100 kHz*/ ,0x02 ,0x00 }, {  7692 /*100 kHz*/ ,0x02 ,0x00 }, {  7805 /*100 kHz*/ ,0x02 ,0x00 }, {  7917 /*100 kHz*/ ,0x02 ,0x00 }, {  8030 /*100 kHz*/ ,0x02 ,0x00 },} },
{ {  7580 /*100 kHz*/ ,0x01 ,0x00 }, {  7692 /*100 kHz*/ ,0x01 ,0x00 }, {  7805 /*100 kHz*/ ,0x01 ,0x00 }, {  7917 /*100 kHz*/ ,0x01 ,0x00 }, {  8030 /*100 kHz*/ ,0x01 ,0x00 },} },
z*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   },} },
```

LTE--B28

- lte_custom_mipi.c

```
LTE_MIPI_EVENT_TABLE_T LTE_Band28_MIPI_TX_EVENT_Set0[] =
{
  /* No.    elm type     , data idx      , evt_type          , evt_offset          */
  /*                  { start, stop },                ( us )                */
  { /* 0  */ LTE_MIPI_PA  , { 0   , 3   }, LTE_MIPI_TRX_ON   , LTE_FDD_MIPI_PA_TX_ON0_Set0  }, //PA On
  { /* 1  */ LTE_MIPI_PA  , { 4   , 5   }, LTE_MIPI_TRX_OFF  , LTE_FDD_MIPI_PA_TX_OFF0_Set0 }, //PA Off
  { /* 2  */ LTE_MIPI_ASM , { 6   , 6   }, LTE_MIPI_TRX_ON   , LTE_FDD_MIPI_ASM_TX_ON0_Set0 }, //WIN on
  { /* 4  */ LTE_MIPI_NULL, { 0   , 0   }, LTE_MIPI_EVENT_NULL, 0                          },
};
```

```
LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_Band28_MIPI_TX_DATA_Set0[] =
{
  //No.    elm type     , port_sel        , data_seq     , USID                    ,{ { subband-
  { /* 0  */ LTE_MIPI_PA  , LTE_MIPI_PORT1   , LTE_REG_W   , MIPI_USID_PA1_Set0    ,
  { /* 1  */ LTE_MIPI_PA  , LTE_MIPI_PORT1   , LTE_REG_W   , MIPI_USID_PA1_Set0    ,
  { /* 2  */ LTE_MIPI_PA  , LTE_MIPI_PORT1   , LTE_REG_W   , MIPI_USID_PA1_Set0    ,
  { /* 3  */ LTE_MIPI_PA  , LTE_MIPI_PORT0   , LTE_REG_W   , MIPI_USID_ASM0_Set0   ,
  { /* 4  */ LTE_MIPI_PA  , LTE_MIPI_PORT1   , LTE_REG_W   , MIPI_USID_PA1_Set0    ,
  { /* 5  */ LTE_MIPI_PA  , LTE_MIPI_PORT0   , LTE_REG_W   , MIPI_USID_ASM0_Set0   ,
  { /* 6  */ LTE_MIPI_ASM , LTE_MIPI_PORT0   , LTE_REG_W   , MIPI_USID_ASM0_Set0
  { /* 7  */ LTE_MIPI_NULL, 0            , 0          ,0                        ,{ {    0 /*100 |
};
```

28A

28B

```
|- 0 freq    ,addr ,data }, { subband- 1 freq    ,addr ,data }, { subband- 2 freq    ,addr ,data }, { subband- 3 freq    ,addr ,data }, { subband- 4 freq    ,addr ,data },
,{ { 7030 /*100 kHz*/ ,0x00 , 0x05 }, { 7142 /*100 kHz*/ ,0x00 , 0x05 }, { 7255 /*100 kHz*/ ,0x00 , 0x05 }, { 7367 /*100 kHz*/ ,0x00 , 0x05 }, { 7480 /*100 kHz*/ ,0x00 , 0x05 },} },
,{ { 7030 /*100 kHz*/ ,0x01 , 0x00 }, { 7142 /*100 kHz*/ ,0x01 , 0x00 }, { 7255 /*100 kHz*/ ,0x01 , 0x00 }, { 7367 /*100 kHz*/ ,0x01 , 0x00 }, { 7480 /*100 kHz*/ ,0x01 , 0x00 },} },
,{ { 7030 /*100 kHz*/ ,0x03 , 0x00 }, { 7142 /*100 kHz*/ ,0x03 , 0x00 }, { 7255 /*100 kHz*/ ,0x03 , 0x00 }, { 7367 /*100 kHz*/ ,0x03 , 0x00 }, { 7480 /*100 kHz*/ ,0x03 , 0x00 },} },
,{ { 7030 /*100 kHz*/ ,0x04 , 0x08 }, { 7142 /*100 kHz*/ ,0x04 , 0x08 }, { 7255 /*100 kHz*/ ,0x04 , 0x08 }, { 7367 /*100 kHz*/ ,0x04 , 0x08 }, { 7480 /*100 kHz*/ ,0x04 , 0x08 },} },
,{ { 7030 /*100 kHz*/ ,0x00 , 0x00 }, { 7142 /*100 kHz*/ ,0x00 , 0x00 }, { 7255 /*100 kHz*/ ,0x00 , 0x00 }, { 7367 /*100 kHz*/ ,0x00 , 0x00 }, { 7480 /*100 kHz*/ ,0x00 , 0x00 },} },
,{ { 7030 /*100 kHz*/ ,0x04 , 0x00 }, { 7142 /*100 kHz*/ ,0x04 , 0x00 }, { 7255 /*100 kHz*/ ,0x04 , 0x00 }, { 7367 /*100 kHz*/ ,0x04 , 0x00 }, { 7480 /*100 kHz*/ ,0x04 , 0x00 },} },
,{ { 7030 /*100 kHz*/ ,0x02 , 0x05 }, { 7142 /*100 kHz*/ ,0x02 , 0x05 }, { 7255 /*100 kHz*/ ,0x02 , 0x04 }, { 7367 /*100 kHz*/ ,0x02 , 0x04 }, { 7480 /*100 kHz*/ ,0x02 , 0x04 },} },
kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   }, {    0 /*100 kHz*/ ,0   ,0   },} },
```

Note : LTE B28 其他配置（如CA，Ant Tuner ， TAS等参照相关文档;
TRX PORT,BPI 等均在Excel List 配置完毕， 由Parsing tool Gen 出Code， 做编译

LTE--Tx Bypass No Support

- Tool_Gen93_MT6763_LTE_MT6177L.xlsm 不能配置LTE_Band_filter Bandindicator 及相关设置

Band Filter setting	最新的Excel 会变为不可选择				
Band_Filter_Indicator_ID	(LTE_BAND)	CCA_Support(TX)	Filter_TX_IO_SEL	Filter_PT1	Filter_PT2
BAND_FILTER_INDICATOR0	LTE_BandNone	NO_CCA_SUPPORT	TX_IO_NON_USED	0x00000000	0x00000000
BAND_FILTER_INDICATOR1	LTE_BandNone	NO_CCA_SUPPORT	TX_IO_NON_USED	0x00000000	0x00000000
BAND_FILTER_INDICATOR2	LTE_BandNone	NO_CCA_SUPPORT	TX_IO_NON_USED	0x00000000	0x00000000
BAND_FILTER_INDICATOR3	LTE_BandNone	NO_CCA_SUPPORT			
BAND_FILTER_INDICATOR4	LTE_BandNone	NO_CCA_SUPPORT			

- lte_custom_mipi.c CW Table 配置与非Filter Table 即可，Filter Table 不用配置

LTE_MIPI_EVENT_TABLE_T LTE_**Band38/40/41**_MIPI_**FILTER**_TX_EVENT_Default/Set*[] =

LTE_MIPI_DATA_SUBBAND_TABLE_T LTE_**Band38/40/41**_MIPI_FILTER_**TX_DATA**_Default/Set*[] =

LTE_MIPI_EVENT_TABLE_T LTE_Band38/40/41_MIPI_FILTER_**TPC_EVENT**_Default/Set0[] =

LTE_MIPI_TPC_SECTION_TABLE_T LTE_**Band38/40/41**_MIPI_**FILTER**_PA_SECTION_DATA_Default/Set*[] =

CONFIDENTIAL B

C2K Part

- 下面列举了93 modem在9个工作场景下，需要对哪些RF前端器件进行设置.
  1）Y：表示需要设置　N：表示不需要设置
  2）TX Gate On/OFF 被Tx ON/OFF 替代
  3）TX ON/TX Gate On Event 配置PA Enable,禁止配置PA Bias
  4）TX On Timing 由20us调整10us

SCENARIOS	ASM	ASM TUNER	DASM	DASM TUNER	PA Enable	Bias For PA
RX ON	Y	Y	N	N	N	N
RX OFF	Y	Y	N	N	N	N
RXD ON	N	N	Y	Y	N	N
RX OFF	N	N	Y	Y	N	N
TX ON	Y	Y	N	N	Y	N
TX OFF	N	N	N	N	Y	N
TX GATE ON	N	N	N	N	Y	N
TX GATE OFF	N	N	N	N	Y	N
TX TPC	N	N	N	N	N	Y

C2K--RX

- 列出C2K RX通路ON/OFF时所需的操作步骤

Event No.	Event Type	Action	Data	Data No.
0	RX ON	设置ASM Tx Off	寄存器0x00, Tx off	0
		设置ASM M TRX Port Off	寄存器0x05, off	1
		设置ASM H TRX Port Off	寄存器0x03, off	2
		设置ASM L TRX Port on	寄存器0x02, on	3
1	RX OFF	寄存器0x02, on	寄存器0x02, off	4
3	RX ON	设置DP12T M Port Isolation	寄存器0x00, Isolation	5
		设置DP12T L Port active	寄存器0x01, Active	6
4	RX OFF	设置DP12T M idle	寄存器0x00, Idle	7
		设置DP12T L idle	寄存器0x01, Idle	8

- Note: C2K BC0
  TXM :Sky77928-11
  MMMB PA:Sky77651-11
  DRX Switch:sky13552-669

C2K--RX

- RX部分的设置格式:

Timing在c2k_custom_mipi.h中定义

```
/*MIPI ASM    */
/* RX ON    */
#define C2K_MIPI_ASM_RX_ON0_Set0          M_US(10)
#define C2K_MIPI_ASM_RX_ON1_Set0          M_US(0)

/*MIPI ASM    */
/* RX OFF    */
#define C2K_MIPI_ASM_RX_OFF0_Set0         M_US(5)
#define C2K_MIPI_ASM_RX_OFF1_Set0         M_US(0)

/*MIPI ASM    */
/* DRX ON    */
#define C2K_MIPI_ASM_DRX_ON0_Set0         M_US(11)
#define C2K_MIPI_ASM_DRX_ON1_Set0         M_US(0)

/*MIPI ASM    */
/* DRX OFF    */
#define C2K_MIPI_ASM_DRX_OFF0_Set0        M_US(6)
#define C2K_MIPI_ASM_DRX_OFF1_Set0        M_US(0)
```

Timing

Event

```
/* Rx Events */
C2K_MIPI_EVENT_TABLE_T C2K_BAND_A_MIPI_RX_EVENT_Set0[] =
{
   /* No.     elm type   , data idx      , evt_type       , evt_offset       */
   /*                { start, stop },          ( us )                      */
   { /* 0 */ C2K_MIPI_ASM , { 0   , 3   }, C2K_MIPI_TRX_ON     , C2K_MIPI_ASM_RX_ON0_Set0 },
   { /* 1 */ C2K_MIPI_ASM , { 4   , 4   }, C2K_MIPI_TRX_OFF    , C2K_MIPI_ASM_RX_OFF0_Set0},
   { /* 2 */ C2K_MIPI_ASM , { 5   , 6   }, C2K_MIPI_RX_DIV_ON  , C2K_MIPI_ASM_DRX_ON0_Set0},
   { /* 3 */ C2K_MIPI_ASM , { 7   , 8   }, C2K_MIPI_RX_DIV_OFF , C2K_MIPI_ASM_DRX_OFF0_Set0},
   { /* 4 */ C2K_MIPI_NULL, { 0   , 0   }, C2K_MIPI_EVENT_NULL, 0},
};
```

Data

```
/* Rx Data */
C2K_MIPI_DATA_SUBBAND_TABLE_T C2K_BAND_A_MIPI_RX_DATA_Set0[] =
{
  //No.     elm type     , port_sel       , data_seq     , USID          ,{ { subband-0 freq   ,addr ,data }, { subband-1 freq   ,addr ,dat
  { /* 0 */ C2K_MIPI_ASM , C2K_MIPI_PORT0 , C2K_REG_W    , MIPI_USID_ASM0_Set0  ,{ { 8600 /*100 kHz*/ , {0x00 ,0x80} },
  { /* 1 */ C2K_MIPI_ASM , C2K_MIPI_PORT0 , C2K_REG_W    , MIPI_USID_ASM0_Set0  ,{ { 8600 /*100 kHz*/ , {0x05 ,0x00} },
  { /* 2 */ C2K_MIPI_ASM , C2K_MIPI_PORT0 , C2K_REG_W    , MIPI_USID_ASM0_Set0  ,{ { 8600 /*100 kHz*/ , {0x03 ,0x00} },
  { /* 3 */ C2K_MIPI_ASM , C2K_MIPI_PORT0 , C2K_REG_W    , MIPI_USID_ASM0_Set0  ,{ { 8600 /*100 kHz*/ , {0x02 ,0x06} },
  { /* 4 */ C2K_MIPI_ASM , C2K_MIPI_PORT0 , C2K_REG_W    , MIPI_USID_ASM0_Set0  ,{ { 8600 /*100 kHz*/ , {0x02 ,0x00} },
  { /* 5 */ C2K_MIPI_ASM , C2K_MIPI_PORT2 , C2K_REG_W    , MIPI_USID_ASM1_Set0  ,{ { 8600 /*100 kHz*/ , {0x00 ,0x1F} },
  { /* 6 */ C2K_MIPI_ASM , C2K_MIPI_PORT2 , C2K_REG_W    , MIPI_USID_ASM1_Set0  ,{ { 8600 /*100 kHz*/ , {0x01 ,0x03} },
  { /* 7 */ C2K_MIPI_ASM , C2K_MIPI_PORT2 , C2K_REG_W    , MIPI_USID_ASM1_Set0  ,{ { 8600 /*100 kHz*/ , {0x00 ,0x00} },
  { /* 8 */ C2K_MIPI_ASM , C2K_MIPI_PORT2 , C2K_REG_W    , MIPI_USID_ASM1_Set0  ,{ { 8600 /*100 kHz*/ , {0x01 ,0x00} },
  { /* 12 */ C2K_MIPI_NULL, 0             , 0            ,{ { 0    /*100 kHz*/ , {0   ,0  } }, { 0    /*100 kHz*/ , {0   ,0   }
};
```

C2K--TX

- 列出TX通路ON/OFF时所需的操作步骤:

Event No.	Event Type	Action	Data	Data No.
0	TX ON	Enable PA	将PA设成active mode	0
			设定PA寄存器2	1
1	TX OFF	Dis-enable PA	设定PA寄存器0	2
3	TX Gate On	Enable PA	将PA设成active mode	3
3	TX Gate OFF	Dis-enable PA	将PA设成standby mode	4
4	TX ASM ON	ASM Active	切换到正确的ASM TRX Port	5-7

Note:

93 modem Tx On Event 不能关闭ASM

93 modem Tx On 不能设置PA Bias 寄存器（清零也不行）

93 Modem Tx On/Gate On,同时配置Pa Enable

93 modem Tx OFF/Gatet OFF 关闭PA

C2K--TX

- TX部分的设置格式:

Timing在C2K_custom_mipi.h中定义

```
/* TX Gate OFF   */
#define C2K_MIPI_PA_TX_GATE_OFF0_Set0      M_US(0)
/*MIPI ASM      */
/* TX ON        */
#define C2K_MIPI_ASM_TX_ON0_Set0           M_US(10)
/* TX Gate ON   */
#define C2K_MIPI_PA_TX_GATE_ON0_Set0       M_US(5)
/* TX OFF       */
#define C2K_MIPI_PA_TX_OFF0_Set0           M_US(10)
/*MIPI PA       */
/* TX ON        */
#define C2K_MIPI_PA_TX_ON0_Set0            M_US(10)
```

Timing

Event

```
/* TX Event */
C2K_MIPI_EVENT_TABLE_T C2K_BAND_A_MIPI_TX_EVENT_Set0[] =
{
  /* No.    elm type     , data idx     , evt_type       , evt_offset          */
  /*                { start, stop },          ( us )                          */
  { /* 0 */ C2K_MIPI_PA  , { 0   , 1   }, C2K_MIPI_TRX_ON      , C2K_MIPI_PA_TX_ON0_Set0  },
  { /* 1 */ C2K_MIPI_PA  , { 2   , 2   }, C2K_MIPI_TRX_OFF     , C2K_MIPI_PA_TX_OFF0_Set0 },
  { /* 2 */ C2K_MIPI_PA  , { 3   , 3   }, C2K_MIPI_TX_GATE_ON  , C2K_MIPI_PA_TX_GATE_ON0_Set0  },
  { /* 3 */ C2K_MIPI_PA  , { 4   , 4   }, C2K_MIPI_TX_GATE_OFF , C2K_MIPI_PA_TX_GATE_OFF0_Set0 },
  { /* 4 */ C2K_MIPI_ASM , { 5   , 7   }, C2K_MIPI_TRX_ON      , C2K_MIPI_ASM_TX_ON0_Set0 },
  { /* 6 */ C2K_MIPI_NULL, { 0   , 0   }, C2K_MIPI_EVENT_NULL, 0                      },
};
```

Data

```
/* TX Data */
C2K_MIPI_DATA_SUBBAND_TABLE_T C2K_BAND_A_MIPI_TX_DATA_Set0[] =
{
  //No.     elm type     , port_sel      , data_seq    , USID              ,{ { subband-0 freq   ,addr ,data }, { subband-1 freq   ,addr ,da
  { /* 0 */ C2K_MIPI_PA   , C2K_MIPI_PORT1 , C2K_REG_W   , MIPI_USID_PA1_Set0  ,{ { 8150 /*100 kHz*/ , {0x00 ,0x94} },
  { /* 1 */ C2K_MIPI_PA   , C2K_MIPI_PORT1 , C2K_REG_W   , MIPI_USID_PA1_Set0  ,{ { 8150 /*100 kHz*/ , {0x02 ,0x00} },
  { /* 2 */ C2K_MIPI_PA   , C2K_MIPI_PORT1 , C2K_REG_W   , MIPI_USID_PA1_Set0  ,{ { 8150 /*100 kHz*/ , {0x00 ,0x00} },
  { /* 3 */ C2K_MIPI_PA   , C2K_MIPI_PORT1 , C2K_REG_W   , MIPI_USID_PA1_Set0  ,{ { 8150 /*100 kHz*/ , {0x00 ,0x94} },
  { /* 4 */ C2K_MIPI_PA   , C2K_MIPI_PORT1 , C2K_REG_W   , MIPI_USID_PA1_Set0  ,{ { 8150 /*100 kHz*/ , {0x00 ,0x90} },
  { /* 5 */ C2K_MIPI_ASM  , C2K_MIPI_PORT0 , C2K_REG_W   , MIPI_USID_ASM0_Set0 ,{ { 8150 /*100 kHz*/ , {0x00 ,0x80} },
  { /* 6 */ C2K_MIPI_ASM  , C2K_MIPI_PORT0 , C2K_REG_W   , MIPI_USID_ASM0_Set0 ,{ { 8150 /*100 kHz*/ , {0x02 ,0x06} },
  { /* 7 */ C2K_MIPI_ASM  , C2K_MIPI_PORT0 , C2K_REG_W   , MIPI_USID_ASM0_Set0 ,{ { 8150 /*100 kHz*/ , {0x04 ,0x0B} },
  { /* 8 */ C2K_MIPI_NULL , 0             , 0           ,                     ,{ {    0 /*100 kHz*/ , {0    ,0   } }, {    0 /*100 kHz*/ , {0    ,0   } },
};
```

C2K--TPC

```
/* TPC Event */
C2K_MIPI_EVENT_TABLE_T C2K_BAND_A_MIPI_TPC_EVENT_Set0[] =
{
  /* No.    elm type     , data idx      , evt_type      , evt_offset   */
  /*                   { start, stop },            ( us )          */
  { /* 0  */ C2K_MIPI_PA ,    { 0   , 2   }, C2K_MIPI_TPC_SET   , 5       },
  { /* 1  */ C2K_MIPI_NULL,   { 0   , 0   }, C2K_MIPI_EVENT_NULL, 0       },
};
```

TCP Timing

```
C2K_MIPI_TPC_SECTION_TABLE_T C2K_BAND_A_MIPI_PA_SECTION_DATA_1XRTT_Set0[C2K_MIPI_SUBBAND_NUM] =
{
  {
    8150, /*100kHz*/
    MIPI_USID_PA1_Set0, /*USID*/
    {
      // PAEn=1
      // PA_SEC_DATA0, PA_SEC_DATA1, PA_SEC_DATA2, PA_SEC_DATA3, PA_SEC_DATA4
      // {addr, data}, {addr, data}, {addr, data}, {addr, data}, {addr, data}
       /* 0 */ {{{ 0x1, 0x24}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 1 */ {{{ 0x1, 0x35}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 2 */ {{{ 0x1, 0x35}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 3 */ {{{ 0x1, 0x65}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 4 */ {{{ 0x1, 0x65}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 5 */ {{{ 0x1, 0x85}, { 0x2, 0x00}, { 0x3, 0x88}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 6 */ {{{ 0x1, 0x86}, { 0x2, 0x00}, { 0x3, 0x88}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
       /* 7 */ {{{ 0x1, 0x98}, { 0x2, 0x00}, { 0x3, 0x88}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
    }
```

1XRTT Data

共8个档位，功率从低到高，

- Note:
  1)TPC Bias timing比TX_ON 开的早一些，但是这里TPC Timing 5us 是失效的，实际取决与HW响应时间，大约10uS，稍早于SW TX_ON 10uS
  2)TPC 不能设置PA Enable

C2K--TPC

```
/* TPC Event */
C2K_MIPI_EVENT_TABLE_T C2K_BAND_A_MIPI_TPC_EVENT_Set0[] =
{
  /* No.    elm type     , data idx      , evt_type      , evt_offset   */
  /*                  { start, stop },              ( us )          */
  { /* 0  */ C2K_MIPI_PA ,    { 0    , 2    }, C2K_MIPI_TPC_SET  , 5       },
  { /* 1  */ C2K_MIPI_NULL,   { 0    , 0    }, C2K_MIPI_EVENT_NULL, 0      },
};
```

TCP Timing

```
/*** EVDO TPC Sections ***/
C2K_MIPI_TPC_SECTION_TABLE_T C2K_BAND_A_MIPI_PA_SECTION_DATA_EVDO_Set0[C2K_MIPI_SUBBAND_NUM] =
{
  {
    8150, /*100kHz*/
    MIPI_USID_PA1_Set0, /*USID*/
    {
      // PAEn=1
      // PA_SEC_DATA0, PA_SEC_DATA1, PA_SEC_DATA2, PA_SEC_DATA3, PA_SEC_DATA4
      // {addr, data}, {addr, data}, {addr, data}, {addr, data}, {addr, data}
      /* 0 */ {{{ 0x1, 0x24}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 1 */ {{{ 0x1, 0x35}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 2 */ {{{ 0x1, 0x35}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 3 */ {{{ 0x1, 0x65}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 4 */ {{{ 0x1, 0x65}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 5 */ {{{ 0x1, 0x85}, { 0x2, 0x00}, { 0x3, 0x80}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 6 */ {{{ 0x1, 0x87}, { 0x2, 0x00}, { 0x3, 0x86}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
      /* 7 */ {{{ 0x1, 0xB7}, { 0x2, 0x00}, { 0x3, 0x86}, { 0x0, 0x0 }, { 0x0, 0x0 }}},
    }
```

EVDO Data

共8个档位，功率从低到高，

- Note:
  1)TPC Bias timing比TX_ON 开的早一些，但是这里TPC Timing 5us 是失效的，实际取决与HW响应时间，大约10uS，稍早于SW TX_ON 10uS
  2)TPC 不能设置PA Enable

Reference document

MMRF Driver setting

CS0021-GAK1J-AND-V1.0EN_Platform_System_RF_MMRF_RF_Error_Check_Application_Note
CS0021-GAK1K-AND-V1.0EN_Platform_System_RF_MMRF_RF_Custom_Setting_Application_Note

LTE Driver Setting

CS0021-GAK1A-AND-V1.2EN_Platform_System_RF_LTE_RF_Custom_Setting_Application_Note.docx
CS0021-GAK1G-AND-V1.1EN_Platform_System_RF_LTE_RF_Error_Check_Application_Note
CS0021-GAK1F-AND-V1.0EN_Platform_System_RF_LTE_RF_Default_Value_Settings
LTE Custom Excel And File Generation Tool(MT6177)
LTE custom Excel and file generation tool(MT6177m)
MT6177-Update_NVRAM_By_GP_Tool

WCDMA

CS0021-GAK1B-AND-V1.0EN_Platform_System_RF_WCDMA_RF_Custom_Settings_Application_Note
CS0021-GAK1H-AND-V1.0EN_Platform_System_RF_WCDMA_RF_Error_Check_Application_Note
How_to_use_NVRAM_editor_to_modify_3G_FDD_RF_settings

TDSCDMA

[MT6177]3G_TDD_How_to_Configure_RF_Custom_File
[MT6177]3G_TDD_MT6177_ASSERT_Description
TDSCDMA_RF_MT6763_RF_NVRAM_LID_Introduction_Application_Note

C2K

CS0021-GAK1P-AND-V1.4EN_Platform_System_RF_CDMA_RF_Custom_Settings_Application_Note

Reference document

2/3/4G MIPI

CS0021-GAK1D-AND-V1.1EN_Platform_System_RF_MIPI_Customization_Application_Note

DRDI（单软多硬）

CS0021-GAK1O-AND-v1.0EN_Platform_System_RF_DRDI_Customization_Application_Note

ELNA

CS0021-GAK1L-AND-V1.0EN_Platform_System_RF_eLNA_Customization_Application_Note
CS0021-GAK1AB-AND-V1.0EN_Platform_System_RF_ELNA_Module_Design Consideration

TAS

CS0021-GAK1I-AND-V1.0EN_Platform_System_RF_TAS_ Customization_Setting_Application_Note

RF Calibration

MTK RF Calibration Application Note
Platform_System_RF_Calibration_and_Test_Flow_Application_Note

Nvram LID edit

CS0021-GAK1C-AND-v1.3EN_Platform_System_RF_RF_NVRAM_LID_Introduction_Application_Note

GP tool

CS0021-GAK1N-AND-V1.0EN_Platform_System_RF_MMRF_GP_Tool_SOP_Application_Note

everyday genius