IT6613 HDMI 1.4 3D Transmitter Programming Guide

Ver 1.00

Tseng, Jau-Chih

ITE Tech. Inc.

Last Update Date: 2011/01/04

History

2010/8/11 created by Tseng, Jau-Chih.

Index

Chap 1 Introduce

IT6613 uses I^2C bus to program its internal registers. The I^2C slave address is defined as 0x98 when PCADR pin (pin#50) is pulled low, otherwise it is 0x9A.

There are two banks of registers in IT6613. The bank switching depends on the content of reg0F (in term reg0F means the register accessed by I^2C sub-address is 0x0F), 0 means the given subaddress of I^2C refers to the register set of bank 0; and 1 means the given subaddress of I^2C refers to the register set of bank 1.

In following term, reg0yy or regyy means the register set of bank 0 with subaddress 0xyy, which yy is from 0x00 to 0xFF; and reg1zz means the register set of bank 1 with subaddress 0xzz and reg0F='1'.

Programming of IT6613

The programming of IT6613 is separated into following parts:

- Initial.
- DDC Command Programming
- Interrupt Handling.
- Program video mode.
- Program audio mode.
- Activate HDCP
- Infoframe Programming

And provide software sample interface.

Chap 2 Initial

To initial IT6613, the following steps should be implement.

1. Chip identifying
2. Reset internal circuit.
3. Enable clock ring.
4. Set default DVI mode and turn off all packet.
5. Set default interrupt mask for event handling.

Chip Identifying

Before programming IT6613 chip, system needs to identify if the chip is IT6613.

Reg	Register Name	Bit	Definition	Default Value
01	Vender ID	7:0		0xCA
02	Device ID	7:0		0x13
03	Device ID	3:0		0x6
	Revision ID	7:4		0x0

As the table, if I^2C slave on address 0x98/0x9A responses the sub-address 0x01 with value 0xCA, and sub-address 0x02 and sub-address 0x03 with 0x13 and 0x06 (in low nibble), the device is IT6613.

Reset internal circuit.

Reg04 is the software reset control register of IT6613. To reset all circuit of IT6613, reg04[5] = '1' then reset to zero, all circuit will be zero.

To enable functions for using, the bits of reg04 should be cleared to zero for enabling. In initial state, reg04 should be 0x1D to wait for function enable.

Enable clock ring.

IT6613 clock ring is defined in drive control registers, reg61. To initial the clock ring, reg61 should be set to 0x10, and reg62~reg66 with hardware default value after reset.

Set default DVI mode and turn off all packet.

IT6613 usually set to DVI mode under initial state. RegC0 is set to zero for DVI mode, and RegC1~RegD0 are all set to zero for disabling all HDMI packets; reg158 is set to zero for output RGB444 mode under DVI mode.

Set default interrupt mask for event handling.

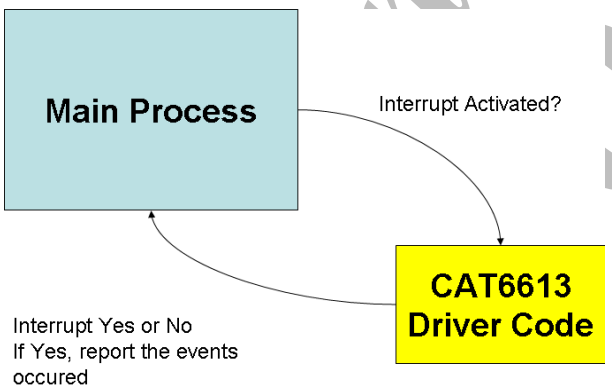
IT6613 interrupt mask are defined in reg09~reg0B. The bits in these registers are for each interrupt status responding switch, '0' for enabling the responding on status registers, and '1' for disabling the responding of status registers bit.

After initial, the system can be prepare to output with programming video and audio mode.

Chap 3 Handling Interrupt

IT6613 activates interrupt pin (PIN24) while events defined in interrupt mask registers occurred. The interrupt activating polarity is programmed in reg05[7]. To check interrupt status is also to read back the register value reg0E[7], where '1' identifying interrupt activated and '0' for otherwise.

Off	Register Name	bit	Description	Default Value
05	REG_INTPol	7	0: INT active low 1: INT active high	0
	REG_INTIOMode	6	1: Open-Drain mode 0: Push-Pull Mode	1
	REGPDTxCLk	0	1: TxCLK power down 0: TxCLK active	1



When system process detected if an interrupt activated by the INT pin or reg0E[7] of IT6613, the driver should report what event occurred. To define the events of activating interrupt, the bits of events should be cleared to zero in reg09, reg0B, reg0A. For example, to monitor the hot-plug event changing event, reg09[0] should be set to zero, then while hot-plug change event occurring it trigger the interrupt signal and set reg06[0] as '1'.

To clear the interrupt and interrupt status recorded in reg06, reg07, and reg08, the corresponding bits in reg0C, reg0D, and reg0E[1] should set to '1' then set reg0E[0] with '1' to clear interrupt and status. For example, after processed the hot-plug change event, the status flag and interrupt have to be cleared, then next event can be detected and processed.

The definition of these registers is shown as following table:

Off	Register Name	bit	Description	Default Value
Interrupt Status				
06	RInt_AudioOvFlwStus	7	R, Reset by REGAudReset	
	Reserved	6		
	RDDC_Stus_NoACK	5	R	
	Rint_DDCFIFOErr	4	R, Reset by RDDC_Req=0x9 REG_MastersSel='1'	
	Reserv			
	RInt_DDCBusHang	2	R, Reset by RDDC_Req=0xF REG_MastersSel='1'	
	RInt_RxSENStus	1	R, Reset by REG_RxSENClr	
	RInt_HPDStus	0	R, Reset by REG_HPDClr	
07	RInt_PktISRC2Stus	7	R, Reset by REG_ISRC2Clr	
	RInt_PktISRC1Stus	6	R, Reset by REG_ISRC1Clr	
	RInt_PktACPStus	5	R, Reset by REG_PktACPClr	
	RInt_PktNullStus	4	R, Reset by REG_PktNullClr	
	RInt_PktGenStus	3	R, Reset by REG_PktGenCr	
	RInt_KSVListChkStus	2	R, Reset by REG_KSVListChkClr	

	RInt_AuthDoneStus	1	R, Reset by REG_AuthenDoneClr	
	RInt_AuthFailStus	0	R, Reset by REG_AuthFailClr	
08	Reserved	7		
	Rint_AudCTSStus	6	R , Reset by Reg_AudCTSClr	
	RInt_VSyncStus	5	R, Reset by REG_VsyncClr	
	RInt_VidStableStus	4	R, Reset by REG_VidStaleClr	
	RInt_PktMpgStus	3	R, Reset by REG_PktMpgClr	
	RInt_PktSPDStus	2	R, Reset by REG_PktSPDClr	
	RInt_PktAudStus	1	R, Reset By REG_PktAudClr	
	RInt_PktAVIStus	0	R, Reset by REG_PktAVIClr	
Interrupt Mask Registers				
09	REG__AudioOvFlwMask	7	1: disable this interrupt.	1
	Reserved	6	0: Enable this interrupt	1
	REG__DDCNoACKMask	5		1
	REG__DDCFIFOErrMask	4		1
	Reserved	3		1
	REG__DDCBusHangMask	2		1
	REG__RxSENMask	1		1
	REG__HPDMask	0		1
0A	REG__PktAVIMask	7		1
	REG__PktISRCMask	6		1
	REG__PktACPMask	5		1
	REG__PktNullMask	4		1
	REG__PktGenMask	3		1
	REG__KSVListChkMask	2		1
	REG__AuthDoneMask	1		1
	REG__AuthFailMask	0		1
0B	REG_AudCTSMask	5		1
	REG__VsyncMask	4		1
	REG__VidStableMask	3		1
	REG__PktMpgMask	2		1
	REG__PktSPDMask	1		1
	REG__PktAudMask	0		1
Interrupt Clear				
0C	REG__PktACPClr	7	1: Clear the interrupt	0
	REG__PktNullClr	6		0
	REG__PktGenClr	5		0
	REG__KSVListChkClr	4		0
	REG__AuthDoneClr	3		0
	REG__AuthFailClr	2		0
	REG__RxSENClr	1		0
	REG__HPDClr	0		0
0D	REG__VsyncClr	7	1: Clear the interrupt	0
	REG__VidStableClr	6		0
	REG__PktMpgClr	5		0
	REG__PktSPDClr	4		0
	REG__PktAudClr	3		0
	REG__PktAVIClr	2		0
	REG__PktISRC2Clr	1		0
	REG__PktISRC1Clr	0		0
System Status				
0E	RInt_FSMON	7	R. 1: Interrupt is active.	
	RHPDetect	6	R Hot Plug Detect: 1: plug on. 0: plug off	
	RxSENDetect	5	R	
	TxVidStable	4	R. Video input status: 1: stable video input. 0: unstable video input.	
	RegCTSIntStep[1:0]	3:2	R/W	11
	Reg_AudCTSClr	1	Clear AduCTS interrupt	0
	Reg_IntActDone_	0	1: Make interrupt clear active. 0: Disable interrupt clear action	0

We usually check the following interrupt events:

1. To detect if the HDMI sink connect and active.

   The hot-plug status responds on reg06[0], it is activated for plug status change, high to low or low to high of the HPD pin of IT6613. When this event occurs, system need to handle the sink detecting or turning the output off depends on the HPD status (reg0E[6]).

   To enable the event handling, system should set reg09[0] = '0'.

   To clear this status, reg0C[0] should be '1' while reg0E[0] written with '1'.

   The reciever sense status responds on reg06[1], it is activated for TMDS differential terminate change, ON to OFF or OFF to ON. When this event occurs, system need to handle the sink detecting or turning the output off depends on the HPD status (reg0E[5]).

   To enable the event handling, system should set reg09[1] = '0'.

   To clear this status, reg0C[1] should be '1' while reg0E[0] written with '1'.

2. When video enable, to detect the input video stable for setting output analog front end.

   The video input stable status change event responds on reg08[4], it is actives for input stable changing from stable to unstable or unstable to stable, under reg04[3] = '1'. If this event occurs means there has been unstable input and need to fire AFE while input stable again (reg0E[4] = '1').

   To enable the event handling, reg0B[3] = '0'.

   To clear this status, reg0D[6] should be '1' while reg0E[0] written with '1'.

3. DDC status events.

   When IT6613 issue a request on DDC bus, the bus hang will activate the reg06[2] = '1'. This interrupt status can only clear by Abort DDC command.

   When the DDC fetch FIFO has error, the reg06[4] will be activated and only can be clear by DDC FIFO clear command.

4. HDCP status events.

   When activate the authentication, there are three related events: authenticate done, authenticate fail, and wait for KSV FIFO.

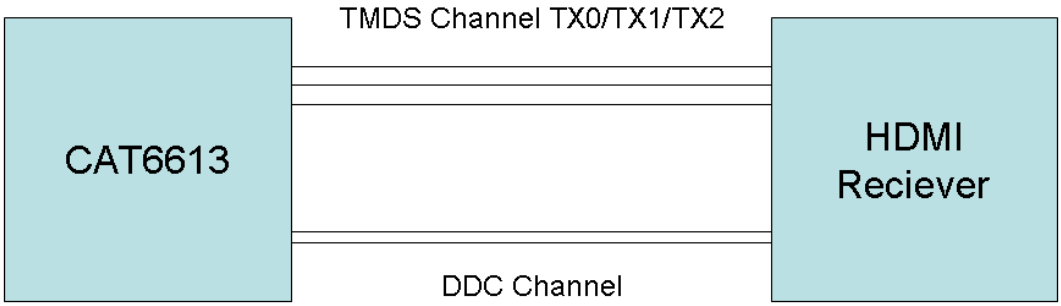
   Authentication done event responded in reg07[1] is activated when HDCP authentication integration verification Ri == Ri', for each i include zero when sink is a receiver or does not include zero when sink is repeater. This event is enabled by reg0A[1] = '0', and cleared by setting reg0C[3] = '1' when reg0E[0] is written by '1'. System usually enable this after the authentication start and disable this after first authentication done.

   Authentication fail event responded in reg07[0] is activated whenever HDCP authentication fail, include the initial authentication checking fail or any integrating check fail. Whenever this event occurs, the HDCP authentication should be restart. This event is enabled by reg0A[0] = '0', and cleared by setting reg0C[2] when writing reg0E[0] with '1'.

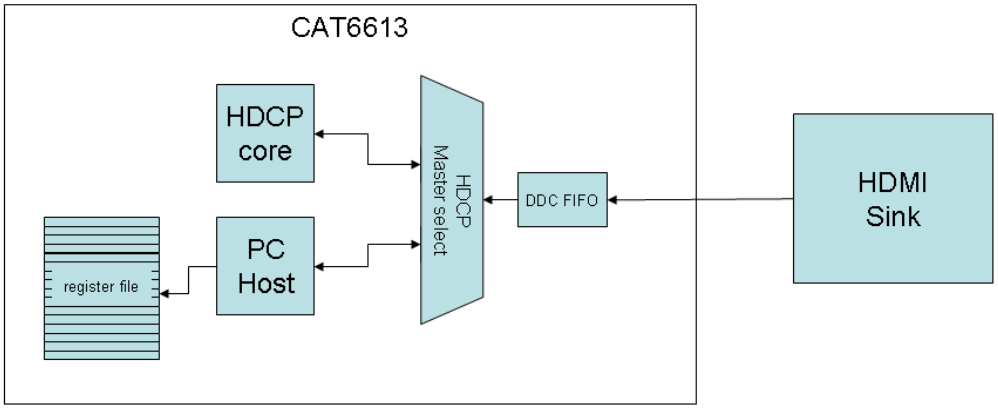
   KSVListCheck event responded in reg07[2] is activated when the sink is an HDCP repeater and the first stage of R0=R0' is checked, then IT6613 should collect the KSVList, confirm them are correct wht V=V' after SHA-1 encoding. When system receive this event, it should begin to check the BCaps of HDCP sink and do the following action. The detail will be describe in the chapter about HDCP authentication. This event is enabled by reg0A[2] = '0', and cleared by setting reg0C[4] with '1' when writing reg0E[0] = '1'.

Chap 4 DDC Command Programming

HDMI transmitter communicates with HDMI sink to exchange data for HDCP authentication and to fetch VESA EDID via DDC bus.



IT6613 has internal DDC master circuits for fetching sink EDID data and exchanging HDCP data. Instead of software implement by firmware, IT6613 provide DDC command to fetch data via DDC. Driver can use these commands to get EDID and HDCP B status registers via DDC.



As described in above figure, there are two DDC masters in IT6613. One is PC-host to fetch data by software controlling, and the other is HDCP core which automatically fetching data via DDC bus.

Reg	Register Name	bit	Definition	Default Value
10	Reg_MasterSel	0	Switch HDCP controller or PC host to command the DDC port 0: HDCP 1: PC	0
11	RDDC_Header[7:0]	7:0	PC DDC request slave address: 0x74 when access Rx HDCP 0xA0 when access Rx EDID	
12	RDDC_ReqOffSet[7:0]	7:0	Register address	
13	RDDC_ReqByte[7:0]	7:0	Register R/W byte number	
14	RDDC_Segment[7:0]	7:0	EDID segment	
15	DDC_SDA	7	R. DDC SDA pin status	
	DDC_SCL	6	R. DDC SCL pin status	
	ROM_SDA	5	R. ROM SDA pin status	
	ROM_SCL	4	R. ROM SCL pin status	

	RDDC_Req[3:0]	3:0	PC DDC request command 0x0 : Sequential Burst Read 0x3: EDID read 0x9: DDC FIFO clear 0xA: GenerateSCL clock pulse 0xF: Abort DDC command.	
16	RDDC_Status[7:0]	 7 6 5 4 3 2 1 0	Read Only. RDDC_Stus_Done '0' : DDC is not complete '1' : DDC transfer is complete RDDC_Active RDDC_Stus_NoACK '1' : DDC has something error RDDC_Stus_WaitBus '1' : DDC has something error RDDC_Stus_ArbiLose '1' : DDC has something error RDDC_FIFOFull RDDC_FIFOEmpty TxFIFO status VRValid	
17	RDDC_ReadFIFO	7:0	R. Read DDC FIFO content. There are 32 DDC FIFO, which can read back from the byte. See Fig. 1	

To select DDC master, software needs to set the reg10[0]. If software wish to fetch data from DDC bus, reg10[0] should be set as '1'. reg10[0] should be set as zero that will be described in HDCP chapter.

There are numerous commands defined in IT6613 PC-host master, to read EDID data and HDCP sink registers. These commands are described following:

Command – DDC Burst Read (0x0)

Software can use this command to read data via DDC bus. To issue the burst read command, the following registers should be programmed:

- Reg11 – I^2C address. Write 0x74 for reading B registers of HDCP, and write 0xA0 to read EDID data. However, to read data of EDID usually use EDID read command (0x03).
- Reg12 – the subaddress of I^2C access on DDC bus. For example, the following table is for reading the registers of HDCP sink while reg11=0x74

sub address	Meaning
0x00	BKSV
0x08	Ri'
0x0A	Pj'
0x20~0x30	V'.H0 ~ V'.H4
0x40	BCaps
0x41/0x42	BStatsus

  By the way, if the data fetched via DDC with HDCP address, IT6613 will automatically put them into the corresponding registers for HDCP between reg3B ~ reg45 instead of the DDC FIFO. For other data, the read back value are kept in DDC FIFO read back from reg17.

- Reg13 – the count to read from DDC bus.
- Reg15 – Only bit [3:0] can be write of this registers, and should be 0x00 for burst read.

After the reg15 fired, the data are ready when reg16[7] = '1' or DDC fail by reg16[5:3] contains any bit as '1'. After the DDC done, software can read the data from corresponding registers (for HDCP) or DDC FIFO.

Get BKSV

Reg10 = 0x01

Reg11 = 0x74

Reg12 = 0x00

Reg13 = 0x05

Reg15 = 0x00

The five bytes of BKSV are ready in reg3B ~ reg3F when reg16[7] = '1'; fail for otherwise.

reg	name	bit	description	default, value
3B	BKSV [7:0]	7:0	Read only.	
3C	BKSV [15:8]	7:0	Read only.	
3D	BKSV [23:16]	7:0	Read only.	
3E	BKSV [31:24]	7:0	Read only.	
3F	BKSV [39:32]	7:0	Read only.	

Get BCaps

Reg10 = 0x01

Reg11 = 0x74

Reg12 = 0x40

Reg13 = 0x01

Reg15 = 0x00

BCaps is ready in reg43 when reg16[7] = '1'; fail for otherwise.

reg	name	bit	description	default, value
43	Bcaps[7:0]	7	HDMI_Reserved	1
		6	HDCP Repeater capability.	
		5	KSV FIFO ready.	
		4	FAST. 1: the device supports 400KHz transfers.	
		3	reserved. must be zero.	
		2	reserved. must be zero.	
		1	1: HDCP 1.1 Features. support HDCP Enhanced encryption status signaling (EESS), Advance Cipher, and Enhanced Link Verification options.	
		0	1: Fast reauthenticagtion. When set to 1, the receiver is capable of receiving (unencrypted) video signal during the session re-authentication. All HDMI-capable receivers shall be capable of performing the fast re-authentication even if this bit is not set. This bit does not change while the HDCP receiver is active.	

Get BStatus

Reg10 = 0x01

Reg11 = 0x74

Reg12 = 0x41

Reg13 = 0x02

Reg15 = 0x00

The Bstatus is ready in reg44/reg45 when reg16[7] = ‘1’; fail for otherwise.

reg	name	bit	description	default, value
45	Bstatus[15:8]	7	reserved 0.	
		6	reserved 0.	
		5	Reserved for future possible HDMI used.	
		4	HDMI_Mode 1: HDMI mode. 0: DVI mode.	
		3	MAX_CASCADE_EXCEEDED Topology error indicator. 1: more than seven levels of video repeater have been cascaded together.	
		2:0	Three-bit repeater cascade depth.	
44	Bstatus[7:0]	7	1: more than 127 downstream devices or KSV fifo.	
		6:0	Total number of attached downstream devices.	

Get KSV FIFO List

Reg10 = 0x01

Reg11 = 0x74

Reg12 = 0x43

Reg13 = count of attached down stream device × 5. Maximum is 30, minimum is 5, zero value is not acceptable.

Reg15 = 0x00

The KSV List is ready in reg17 which DDC FIFO when reg16[7] = ‘1’; fail for otherwise.

Get V’

Reg10 = 0x01

Reg11 = 0x74

Reg12 = 0x20

Reg13 = 0x14

Reg15 = 0x00

The 20 bytes of V’ are ready in reg51~reg54 with reg50[2:0] = 0~4. when reg16[7] = ‘1’; fail for otherwise.

Reg	Name	Bit	Description	Default Value
50	SHASel[2:0]	2:0	See SHA_Rd_ByteX registers below	
51	SHA_Rd_Byte1[7:0]	7:0	V0h[7:0] when SHASel=”000” V1h[7:0] when SHASel=”001” V2h[7:0] when SHASel=”010” V3h[7:0] when SHASel=”011” V4h[7:0] when SHASel=”100” Mi[7:0] when SHASel=”101”	
52	SHA_Rd_Byte2[7:0]	7:0	V0h[15:8] when SHASel=”000” V1h[15:8] when SHASel=”001” V2h[15:8] when SHASel=”010” V3h[15:8] when SHASel=”011” V4h[15:8]when SHASel=”100” Mi[15:8] when SHASel=”101”	

53	SHA_Rd_Byte3[7:0]		V0h[23:16] when SHASel="000" V1h[23:16] when SHASel="001" V2h[23:16] when SHASel="010" V3h[23:16] when SHASel="011" V4h[23:16] when SHASel="100" Mi[23:16] when SHASel="101"	
54	SHA_Rd_Byte4[7:0]	7:0	V0h[31:124] when SHASel="000" V1h[31:24] when SHASel="001" V2h[31:24] when SHASel="010" V3h[31:24] when SHASel="011" V4h[31:24] when SHASel="100" Mi[31:24] when SHASel="101"	
55	Aksv_Rd_Byte5[7:0]	7:0	Mi[39:32] when SHASel="000" Mi[47:40] when SHASel="001" Mi[55:48] when SHASel="010" Mi[63:56] when SHASel="011"	

In HDCP spec, V' is in V0[31:0], V1[31:0], V2[31:0], V3[31:0], and V4[31:0]. When getting V' via DDC is done, the V' can be get by the following steps:

Set Reg50 = 0x00 ; then
V0[7:0] = reg51 ; V0[15:8] = reg52 ; V0[23:16] = reg53 ; V0[31:24] = reg54 , and
Set Reg50 = 0x01 ; then
V1[7:0] = reg51 ; V1[15:8] = reg52 ; V1[23:16] = reg53 ; V1[31:24] = reg54 , and
Set Reg50 = 0x02 ; then
V2[7:0] = reg51 ; V2[15:8] = reg52 ; V2[23:16] = reg53 ; V2[31:24] = reg54 , and
Set Reg50 = 0x03 ; then
V3[7:0] = reg51 ; V3[15:8] = reg52 ; V3[23:16] = reg53 ; V3[31:24] = reg54 , and
Set Reg50 = 0x04 ; then
V4[7:0] = reg51 ; V4[15:8] = reg52 ; V4[23:16] = reg53 ; V4[31:24] = reg54 .

Command – EDID Read (0x3)

To get the EDID data, DDC master should write segment with I^2C address 0x60 then ask the bytes with I^2C address 0xA0. (That is the major difference to burst read.) The programming of EDID read should set the following registers:

- Reg11 – Should set 0xA0 for EDID fetching.
- Reg12 – Set the starting offset of EDID block on current segment.
- Reg13 – Set the number of byte to read back. The data will be put in DDC FIFO, therefore, cannot exceed the size (32) of FIFO.
- Reg14 – The segment of EDID block to read.
- Reg15 – DDC command should be 0x03.

After reg15 written 0x03, the command is fired and successfully when reg16[7] = '1' or fail by reg16[5:3] contains any bit '1'. When EDID read done, EDID can be read from DDC FIFO.

Note: By hardware implementation, the I^2C access sequence on PCSCL/PCSDA should be
<start>-<0x98/0x9A>-<0x17>-<Restart>-<0x99/0x9B>-<read data>-<stop>
If the sequence is the following sequence, the FIFO read will be fail.
<start>-<0x98/0x9A>-<0x17>-<stop>-<start>-<0x99/0x9B>-<read data>-<stop>

Command – DDC FIFO Clear (0x9)

To avoid the remaining data confusing the data fetching from DDC, before data read (burst or EDID), we suggest to clear DDC FIFO at first. To clear DDC FIFO should write reg0F with 0x09, and while reg16[7] = '1', the action is done. Otherwise, the

Command – Abort DDC Bus (0xF)

Sometime the DDC bus will hang by slave incorrect action, IT6613 provide the abort DDC command to resume DDC bus. Writing reg10 with 0xF abort the DDC bus (with sending numerous '1'-'0' pair to bus). If DDC bus aborting is done, reg16[7] will be '1', otherwise reg16[5:3] will contain any bit as '1'.

For processing HDCP fail, DDC bus aborting and DDC FIFO clearing are also necessary, before process them, HDCP should be turned off (by reg04[0] = '1') and DDC master should switch from HDCP core to PC-host with writing '1' to reg10.

Chap 5 Program Video Mode

To enable the video of IT6613, the input signal type and output TMDS should be programmed.

First, the video circuit should be enabled with reg04[5][3] = '0' '0'.

Programming Input Signal

For setting the input signal type, reg70[4:2] should be indicated. If input signal do not include DE (data enable) or use sync-embedded mode, the Pattern Sync/DE Generation Registers (reg90~regA3) need to be programmed by timing standard.

Following registers are for input signal and color mode setting:

Video register and input signal controlling registers

Reg	Register Name	bit	Definition	Default Value
04	RegSoftRefRst	5	Software RCLK reset.	0
	RegSoftARst	4	Software Audio clock base signal reset.	1
	REGSoftVRst	3	Software Video clock base signal reset.	1
	REGAudReset	2	Audio FIFO reset.	1
	REGHDCP_rst	0	HDCP reset.	0
Input Data Format Registers				
70	Reg_InColMod[1:0]	7:6	00: RGB mode 01: YUV422 mode 10: YUV444 mode	00
	Reg_PCLKDiv2	5	0: IO clock = TxCLK 1: IO clk=1/2 *TxCLK	0
	Reg_2x656Clk	4	1: CCIR656 mode(YUV422, 8/12 bit mode) 0: non- CCIR656 mode	0
	Reg_SyncEmb	3	1: Sync Embedded mode 0: Sync Sep mode	0
	Reg_InDDR	2	1: Input DDR 0: Input SDR	0
72	Reg_EnDither	7	Enable dither function	0
	Reg_EnUdFilt	6	Enable Cr/CB up/down sampling function	0
	Reg_DNFreeGo	5	Dither Noise Pattern	0
	Reg_CSCSel[1:0]	1:0	00 : No color space converstion. 10: RGB to YUV 11: YUV to RGB	00

Setting Sync Embedded and DE Generating

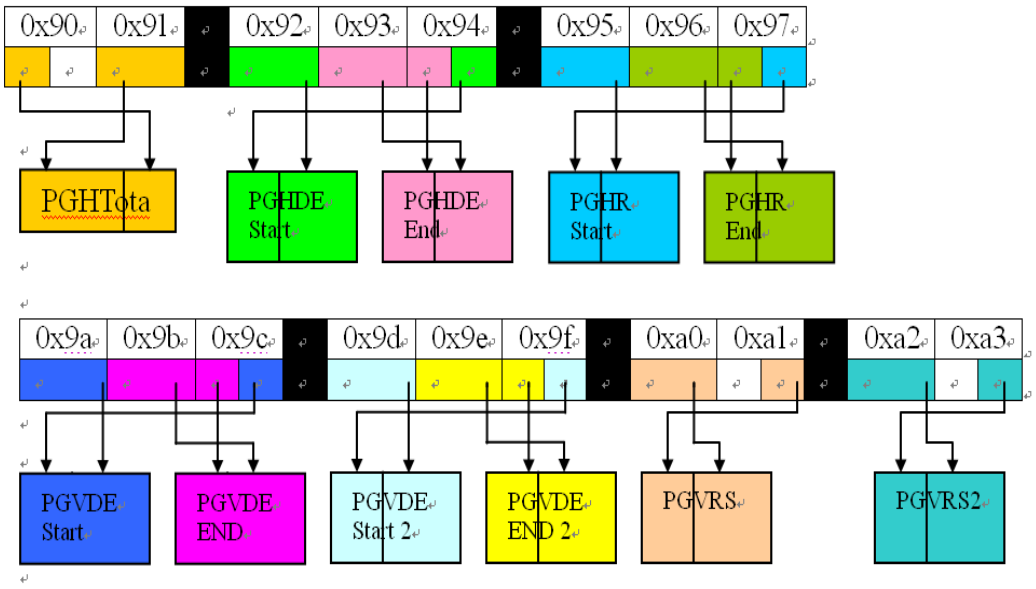
If the input signal do not include DE even though the sync is embedded, the output signal have to be programmed. If the timing generating registers are not programmed, the input video stable will not detected (reg0E[4]) and the output timing will be wrong.

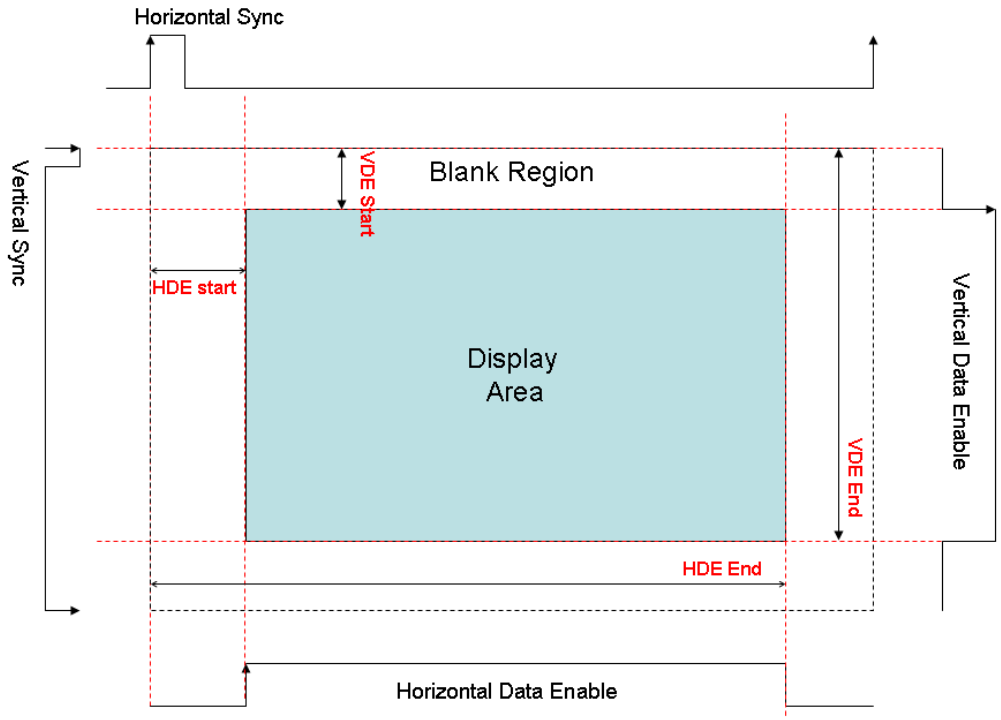
Pattern Sync/DE Generation Registers

Reg	Register Name	bit	Definition	Default Value
90	Reg_PGHTotal[3:0]	7:4	PG Horizontal Total; See also Reg 0x91. When Reg 0xA8[3]=1, this is used for debug, read back only	00
	RegGenSync	3	Generate Sync	0
	RegVSPol	2	Generated Vertical Sync Polarity	0
	RegHSPol	1	Generated Horizontal Sync Polarity	0
	Reg_GenDE	0	DE generation Enable	0
91	Reg_PGHTotal[11:4]	7:0	PG Horizontal Total.	
92	Reg_PGHDES[7:0]	7:0	PG Horizontal Display Start; Low Byte.	
93	Reg_PGHDEE[7:0]	7:0	PG Horizontal Display End; Low Byte.	
94	Reg_PGHDEE[11:8]	7:4	PG Horizontal Display End ; High Byte.	
	Reg_PGHDES[11:8]	3:0	PG Horizontal Display Start ; High Byte.	
95	Reg_PGHRS[7:0]	7:0	PG Horizontal Sync Start; Low Byte.	
96	Reg_PGHRE[7:0]	7:0	PG Horizontal Sync End; Low Byte.	
97	Reg_PGHRE[11:8]	7:4	PG Horizontal Sync End ; High Byte.	

	Reg_PGHRE[11:8]	3:0	PG Horizontal Sync Start ; High Byte.	
98	Reg_PGVTotal[7:0]	7:0	PG Vertical Total; Low Byte.	
99	Reg_PGVTotal[10:8]	2:0	PG Vertical Total; High Byte.	
9A	Reg_PGVDES[7:0]	7:0	PG Vertical Display Start; Low Byte.	
9B	Reg_PGVDEE[7:0]	7:0	PG Vertical Display End; Low Byte.	
9C	Reg_PGVDEE[10:8]	6:4	PG Vertical Display End; High Byte.	
	Reg_PGVDES[10:8]	2:0	PG Vertical Display Start; High Byte.	
9D	Reg_PGVDES2nd[7:0]	7:0	PG 2nd Field Vertical Display Start; Low Byte.	
9E	Reg_PGVDEE2nd[7:0]	7:0	PG 2nd Field Vertical Display End; Low Byte.	
9F	Reg_PGVDEE2nd[10:8]	6:4	PG 2nd Field Vertical Display Start; High Byte.	
	Reg_PGVDES2nd[10:8]	2:0	PG 2nd Field Vertical Display End; High Byte.	
A0	Reg_PGVRS[7:0]	7:0	PG Vertical Sync Start; Low Byte.	
A1	Reg_PGVRE[3:0]	7:4	PG Vertical Sync End	
	Reg_PGVRS[10:8]	2:0	PG Vertical Sync Start; High Byte.	
A2	Reg_PGVRS2nd[7:0]	7:0	PG 2nd Field Vertical Sync Start; Low Byte.	
A3	Reg_PGVRE2nd[3:0]	7:4	PG 2nd Field Vertical Sync End	
	Reg_PGVRS2nd[10:8]	2:0	PG 2nd Field Vertical Sync Start; High Byte.	

register mapping





Generate DE with Input Horizontal Sync and Vertical Sync Presented

Reg90[0] = '1' for DE Generating setting.

Reg90[3] = '0' for no sync gen.

PGHTotal[11:0] = don't care;

PGHDES[11:0] = HDE Start – 2 ;

PGHDEE[11:0] = HDE End – 2 = HActive + PGHDES ;

PGVTotal[11:0] = don't care.

PGVDES[10:0] = VDE Start – 1 ;

PGVDEE[10:0] = PGVDES + VActive = VDE End – 1;

PGVDES2[10:0] = VDE Start of field 2 (from sync 1) – 1 ; (check the timing table)

If DE and Sync are both given, input is 24bit or 16bit, reg70=0xX0, reg90[3][0] = '0' '0', Timing Generation Registers are don't care.

If Sync are provided and DE is muxed in Data (CCIR656/CCIR601), reg70[3] = '1', reg90[3]

Generating Output H/V Sync and DE with Input Sync

Reg90[0] = '1' for DE Generating setting.

Reg90[3] = '1' for H/V sync generating.

<<Reference CEA-861-D>>

PGHTotal = (HTotal/2)-2 *(Adjust active video Horiaontal Position)*

*(if **progressive mode** PGHTotal must be write 0xFFF)*

PGHDEStart = PGHRE +V Back Proch

PGHDEEnd = PGHDEStart+Horiaontal Clocks of Active video

PGHRS = By Customer define *(Adjust active video Horiaontal Position)*

PGHRE = PGHRS+HSYNC width

PGVTotal is not use

PGVDES= PGVRS+ VSYNC width+ V Front Proch1

PGVDEE= PGVDES+ Active Vertical Line per field

PGVRS= By Customer define*(Adjust active video vertical Position)*

PGVRE=(PGVRS+ VSYNC width) % 16

Only use in interlace mode

PGVDES2= PGVDEE+field2 Vertical Blanking Lines

PGVDEE2= PGVDES2+ Active Vertical Line per field

PGVRS2= PGVDES2+V Front Proch2

PGVRE2=(PGVRS2+ VSYNC width) % 16

Sync Embedded

Reg70[3] = '1'

Reg90[3][0] = '0' '0'

PGHTotal = HTotal/2 + FrontPorch – 2 ;

PGHRS = for 16bit, = DEEnd to HSyncStart – 2 = FrontPorch – 2
PGHRE = For 16bit, = DEEnd to HSyncEnd– 2 = FrontPorch – HSyncWidth - 2
PGVRS = scan line from VDE end to VSync Start = VFrontPorch
PGVRE = scan line from VDE end to VSync End = VFrontPorch + VSync Width

For PGVRE is only 4 bit available, we just define the bit [10:4] use as the same value of PGVRS

PGVRS2 = scan line from VDE end to VSync2 Start = VFrontPorch + VTotal

PGVRE2 = scan line from VDE end to VSync2 End = VFrontPorch + VSyncEnd + VTotal

For PGVRE2is only 4 bit available, we just define the bit [10:4] use as the same value of PGVRS2

For VRS2 will check the field change for odd field, if the field is later than VSync2, VSync2 will be ignore. Then the VSync should be pushed later to avoid this issue but will produce timing not meet standard.

VIC	mode	Horizontal Setting for 16bit					Vertical Setting				PCLK	VFREQ
		0x90	0x91	0x95	0x96	0x97	0xA0	0xA1	0xA2	0xA3		
1	640x480@60	0xF0	0x31	0x0E	0x6E	0x00	0x0A	0xC0	0xFF	0xFF	25175000	60
2	480p60	0xF0	0x31	0x0E	0x4c	0x00	0x09	0xF0	0xFF	0xFF	27000000	60
3		0xF0	0x31	0x0E	0x4c	0x00	0x09	0xF0	0xFF	0xFF	27000000	60

4	720p60	0x76	0x33	0x6c	0x94	0x00	0x05	0xA0	0xFF	0xFF	74175000	60
5	1080i60	0x26	0x4A	0x56	0x82	0x00	0x02	0x70	0x34	0x92	74175000	60
6	480i60	0xE0	0x1B	0x11	0x4F	0x00	0x04	0x70	0x0A	0xD1	27000000	60
7		0xE0	0x1B	0x11	0x4F	0x00	0x04	0x70	0x0A	0xD1	27000000	60
8	240p60	0x00	0xff	0x11	0x4F	0x00	0x04	0x70	0xFF	0xFF	27000000	60
9		0x00	0xff	0x11	0x4F	0x00	0x04	0x70	0xFF	0xFF	27000000	60
10	2880x480i60	0xe0	0x1b	0x11	0x4F	0x00	0x04	0x70	0x0A	0xD1	54000000	60
11		0xe0	0x1b	0x11	0x4F	0x00	0x04	0x70	0x0A	0xD1	54000000	60
12	2880x240p60	0x00	0xff	0x11	0x4F	0x00	0x04	0x70	0xFF	0xFF	54000000	60
13		0x00	0xff	0x11	0x4F	0x00	0x04	0x70	0xFF	0xFF	54000000	60
14	1440x480p60	0x00	0xff	0x1e	0x9A	0x00	0x09	0xF0	0xFF	0xFF	54000000	60
15		0x00	0xff	0x1e	0x9A	0x00	0x09	0xF0	0xFF	0xFF	54000000	60
16	1080p60	0x06	0xff	0x56	0x82	0x00	0x04	0x90	0xFF	0xFF	148350000	60
17	576p50	0x00	0xff	0x0a	0x4A	0x00	0x05	0xA0	0xFF	0xFF	27000000	50
18		0x00	0xff	0x0a	0x4A	0x00	0x05	0xA0	0xFF	0xFF	27000000	50
19	720p50	0x46	0x59	0xB6	0xDE	0x11	0x05	0xA0	0xFF	0xFF	74250000	50
20	1080i50	0x66	0x73	0x0e	0x3A	0x22	0x02	0x70	0x34	0x92	74250000	50
21	576i50	0xA0	0x1B	0x0A	0x49	0x00	0x02	0x50	0x3A	0xD1	27000000	50
22		0xA0	0x1B	0x0a	0x49	0x00	0x02	0x50	0x3A	0xD1	27000000	50
23	288p50	0x00	0xff	0x0a	0x49	0x00	0x02	0x50	0xFF	0xFF	27000000	50
24		0x00	0xff	0x0a	0x49	0x00	0x02	0x50	0xFF	0xFF	27000000	50
25	2880x576i50	0xA0	0x1B	0x0a	0x49	0x00	0x02	0x50	0x3A	0xD1	54000000	50
26		0xA0	0x1B	0x0a	0x49	0x00	0x02	0x50	0x3A	0xD1	54000000	50
27	2880x288p50	0x00	0xff	0x0a	0x49	0x00	0x02	0x50	0xFF	0xFF	54000000	50
28		0x00	0xff	0x0a	0x49	0x00	0x02	0x50	0xFF	0xFF	54000000	50
29	1440x576p50	0x00	0xff	0x16	0x96	0x00	0x05	0xA0	0xFF	0xFF	54000000	50
30		0x00	0xff	0x16	0x96	0x00	0x05	0xA0	0xFF	0xFF	54000000	50
31	1080p50	0x06	0xff	0x0e	0x3a	0x22	0x04	0x90	0xFF	0xFF	148500000	50

Setting Color

Color Setting Registers are as following described:

Reg	Name	bit	Description	Default
70	Reg_InColMod[1:0]	7:6	00: RGB mode 01: YUV422 mode 10: YUV444 mode	00
	Reg_PCLKDiv2	5	0: IO clock = TxCLK 1: IO clk=1/2 *TxCLK	0
	Reg_2x656Clk	4	1: CCIR656 mode(YUV422, 8/12 bit mode) 0: non- CCIR656 mode	0
	Reg_SyncEmb	3	1: Sync Embedded mode 0: Sync Sep mode	0
	Reg_InDDR		1: Input DDR 0: Input SDR	0
72	Reg_EnDither	7	Enable dither function	0
	Reg_EnUdFilt	6	Enable Cr/Cb up/down sampling function	0
	Reg_DNFreeGo	5	Dither Noise Pattern	0
	Reg_CSCSel[1:0]	1:0	00 : No color space converstion. 10: RGB to YUV 11: YUV to RGB	00
Color Space Conversion				
73	Reg_YoffSet	7:0	Y blank level	0x10
74	Reg_CoffSet[7:0]	7:0	C blank level	0x80
75	Reg_RGBOffSet[7:0]	7:0	R/G/B blank level	0x00
76	Reg_Matrix11V[7:0]	7:0	Color space conversion Matrix	
77	Reg_Matrix11V[13:8]	4:0		
78	Reg_Matrix12V[7:0]	7:0	Color space conversion Matrix	
79	Reg_Matrix12V[13:8]	5:0		
7A	Reg_Matrix13V[7:0]	7:0	Color space conversion Matrix	
7B	Reg_Matrix13V[13:8]	5:0		
7C	Reg_Matrix21V[7:0]	7:0	Color space conversion Matrix	
7D	Reg_Matrix21V[13:8]	5:0		
7E	Reg_Matrix22V[7:0]	7:0	Color space conversion Matrix	
7F	Reg_Matrix22V[13:8]	5:0		
80	Reg_Matrix23V[7:0]	7:0	Color space conversion Matrix	
81	Reg_Matrix23V[13:8]	5:0		

82	Reg_Matrix31V[7:0]	7:0	Color space conversion Matrix	
83	Reg_Matrix31V[13:8]	5:0		
84	Reg_Matrix32V[7:0]	7:0	Color space conversion Matrix	
85	Reg_Matrix32V[13:8]	5:0		
86	Reg_Matrix33V[7:0]	7:0	Color space conversion Matrix	
87	Reg_Matrix33V[13:8]	5:0		
Output Color Mode in AVI Infoframe				
158	REGPktAVIInfoY[1:0]	6:5	Output Color Mode '00' – RGB444 mode '01' – YCbCr422 mode '10' – YCbCr444 mode	
	REGPktAVIInfoA	4		
	REGPktAVIInfoB[1:0]	3:2		
	REGPktAVIInfoS[1:0]	1:0		

To program color setting of IT6613, software should set

1. Program the input color mode (depends on the input signal type).
2. Program the output color mode setting , in AVI Infoframe Y field (reg158[6:5]), even though the output is under DVI mode (have to set as RGB444 mode).
3. Program the color converting matrix.

Color space converting table

		RGB to YUV				YUV to RGB			
		RGB to YUV 601		RGB to YUV 709		YUV to RGB 601		YUV to RGB 709	
	reg	16~ 235	0 ~ 255	16~ 235	0 ~ 255	16~ 235	0 ~ 255	16~ 235	0 ~ 255
Reg_CSCSel[1:0]	72[1:0]	10	10	10	10	11	11	11	11
Reg_YoffSet[7:0]	73	0x00	0x10	0x00	0x10	0x00	0x10	0x00	0x10
Reg_CoffSet[7:0]	74	0x80	0x80	0x80	0x80	0x80	0x80	0x80	0x80
Reg_RGBOffSet[7:0]	75	0x00	0x10	0x00	0x10	0x00	0x10	0x00	0x10
Reg_Matrix11V[13:0]	76	0xB2	0x09	0xB8	0xE5	0x00	0x4F	0x00	0x4F
	77	0x04	0x04	0x05	0x04	0x08	0x09	0x08	0x09
Reg_Matrix12V[13:0]	78	0x64	0x0E	0xB4	0x78	0x6A	0x81	0x53	0xBA
	79	0x02	0x02	0x01	0x01	0x3A	0x39	0x3C	0x3B
Reg_Matrix13V[13:0]	7A	0xE9	0xC8	0x93	0x81	0x4F	0xDF	0x89	0x4B
	7B	0x00	0x00	0x00	0x00	0x3D	0x3C	0x3E	0x3E
Reg_Matrix21V[13:0]	7C	0x93	0x0E	0x49	0xCE	0x00	0x4F	0x00	0x4F
	7D	0x3C	0x3D	0x3C	0x3C	0x08	0x09	0x08	0x09
Reg_Matrix22V[13:0]	7E	0x18	0x84	0x18	0x84	0xF7	0xC2	0x51	0x56
	7F	0x04	0x03	0x04	0x03	0x0A	0x0C	0x0C	0x0E
Reg_Matrix23V[13:0]	80	0x56	0x6E	0x9F	0xAE	0x00	0x00	0x00	0x00
	81	0x3F	0x3F	0x3F	0x3F	0x00	0x00	0x00	0x00
Reg_Matrix31V[13:0]	82	0x49	0xAC	0xD9	0x49	0x00	0x4F	0x00	0x4F
	83	0x3D	0x3D	0x3C	0x3D	0x08	0x09	0x08	0x09
Reg_Matrix32V[13:0]	84	0x9F	0xD0	0x10	0x33	0x00	0x00	0x00	0x00
	85	0x3E	0x3E	0x3F	0x3F	0x00	0x00	0x00	0x00
Reg_Matrix33V[13:0]	86	0x18	0x84	0x18	0x84	0xDB	0x1E	0x87	0xE7
	87	0x04	0x03	0x04	0x03	0x0D	0x10	0x0E	0x10

Following then above table, depends on the input and output color relationship, program the value into the reg72~reg87 for converting the color table, the color mode is OK.

Deep Color Setting

If the sink support color setting in HDMI EDID extension, IT6613 can support deep color up to 36bit (12 bit for each color channel).

Reg	Register Name	bit	Definition	Default Value
C1	REGAVMute	0	Set AVMute	1

			'0': not AVMute '1': AVMute	
	REGBlueScrMute	1	Set Blue screen output when AVMute='1'	0
	REGNoDefPhase	2		0
	REGPhaseReSync	3		0
	REGColorDepth	6:4	Color Depth 000 – color depth is not indicate (as 8bit) 100 – 8/8/8 bit color 101 – 10/10/10 bit color 110 – 12/12/12 bit color	000

Deep color setting is under regC1[6:4], just need to set the bits. However, the analog front end setting need to be considered depends on PCB layout.

If the sink did not support deep color, suggest the regC1[6:4] as '000' (undefined.)

Analog Front End

IT6613 output signal setting is on reg61~reg67. The setting depends on the output TMDS clock frequency. The setting are as following table.

Reg	Register Name	bit	Definition	Default Value
61	REG_DRV_PWD	5	Power down signal for HDMI_TX_DRV. '1' all flip-flops in the transmitter are powered down while all other analog parts are powered off.	0
	REG_DRV_RST	4	Reset signal for HDMI_TX_DRV. '1': all flip-flops in the transmitter, including those in the BIST pattern generator, are reset.	1
	REG_DRV_PDRXDET	2		0
	REG_DRV_TERMON	1		1
	REG_DRV_ENCAL	0		1
62	REG_XP_GAINBIT	7	Video frequency band selection For video clock frequency < 80Mhz, set to '0', otherwise set to '1'.	1
	REG_XP_PWDPLL	6	Power down signal for TMDSTXPLL018C When '0', normal operation When '1', TMDSTXPLL018C is powerdowned	0
	REG_XP_ENI	5	When '1', the charge pump current of TMDSTIPLL018 is increased.	0
	REG_XP_ER0	4	Adjust filter parameters of TMDSTXPLL018C When '0', base filter resistance value When '1', increased filter resistance value ** If XP_GAINBIT_LV = '0', XP_ER0_LVshould be set to '1'	0
	REG_XP_RESETB	3	Low-active reset signal for TMDSTXPLL018C When '0', TMDSTXPLL018C is reset. When '1', normal operation	1
	REG_XP_PWDI	2	Decides whether the output bias currents provided by TMDSTXPLL018C to other analog blocks are power-downed (turned off) or not. When '0', output bias currents are not turned off even if XP_PWDPLL_LV is asserted. When '1', output bias currents are unconditionally turned off, which at the same time renders the VCO inoperable (CAUTION!!)	0
	REG_XP_DEI	1	When '1', charge pump current of TMDSTXPLL018C is decreased.	0
	REG_XP_DER	0		0
63	REG_RTERM_SEL	7		0
	REG_IP_BYPASS	6		0
	REG_DRV_ISW	5:3		000
	REG_DRV_ISWK	2:0		000

64	REG_IP_GAINBIT	7	Video frequency band selection For video clock frequency < 80Mhz, set to '0', otherwise set to '1'.	1
	REG_IP_PWDPLL	6	Powerdown signal for TMDSIPLL018 When '0', normal operation When '1', TMDSIPLL018 is powerdowned	0
	REG_IP_CKSEL	4:5		00
	REG_IP_ER0	3	Adjust filter parameters of TMDSIPLL018 When '0', base filter resistance value When '1', increased filter resistance value ** If IP_GAINBIT_LV = '0', IP_ER0_LV should be set to '1'	0
	REG_IP_RESETB	2	Low-active reset signal for TMDSIPLL018 When '0', TMDSIPLL018 is reset. When '1', normal operation	1
	REG_IP_ENC	1		0
	REG_IP_EC1	0		0
65	REG_CAL_UPDATE	7		
	REG_CAL_MANUAL	6		
	REG_CAL_CLK_MODE	4:5		00
	REG_DRV_VSW[1:0]	3:2	TMDS output signal level enhancement When '00', no enhancement. When '01', 12% enhancement. When '10', 24% enhancement. When '11', 35% enhancement.	00
	REG_RING_SLOW	1	Set RING_SLOW_LV=1 to speed up the frequency of RING_CK_LV	0
	REG_RING_FAST	0	Set RING_FAST_LV = '1' to slow down the frequency of RING_CK_LV	0
66	REG_AFE_ENTEST	6		0
	REG_AFE_ENBIST	5		0
	REG_CAL_RTERM_MANUAL	4:0		0x10
67	REGAFELFSRVal	7		0
	REGDisAFELFSR	6		0
	REG_RTERM_VAOUE	4:0	Read back	

If Reg61[5] = '1' , the AFE will be powered down.

If Reg61[4] = '1', the AFE will be reset. Either one of these two bits is one, TMDS have no output.

	TMDS Clock < 80MHz	80MHz < TMDS Clock < 170MHz	TMDS Clock > 170MHz
reg61	0x00	0x00	0x00
reg62	0x18	0x88	0x88
reg63	0x10	0x10	0x10
reg64	0x0C	0x84	0x84
reg65	Default	Default	Default
reg66	Default	Default	Default
reg67	Default	Default	Default

Above table is the setting value for analog front end setting. While reg62~reg67 are set, reg04[3] should be clear to zero for enabling the video circuit. While input video stable (when reg0E[4] = '1'), reg61 should be written with the value 0x03 (as fire action) whenever video stable from unstable to stable.

Reset Video after SetupAFE and FireAFE.

After set the AFE parameters, it is suggested to reset video (reg04[3] = '1').

1. Reg61[5:4] = '11' to power down AFE.
2. Set reg62~reg65 value.
3. Reg04[3] = '0'
4. Wait for video stable ready (reg0E[4] = '1')
5. Reg04[3] = '1'

6. Reg04[3] = ‘0’

7. Wait for video stable ready (reg0E[4] = ‘1’)

8. ret61 = 0x00

DVI HDMI Mode

IT6613 HDMI mode setting is in regC0, all HDMI packet and audio can only run under HDMI mode (regC0 = ‘1’).

If set regC0 = ‘0’, it is DVI mode.

Reg	Register Name	bit	Definition	Default Value
C0	REGHDMIMode	0	Set TX Mode ‘0’: DVI mode ‘1’: HDMI mode	0

Mute

When regC1[0] = ‘1’, CAT6611 send blank screen, with (black screen under regC1[1] = ‘0’ or blue screen under regC1[1] = ‘1’).

Under HDMI mode (regC0[0] = ‘1’), AVMute should set in general control package (GCP), the general control package have to be programmed. If regC6[1:0] = ‘11’, the general control package will be enabled and sent repeating. The AVMute and color depth function contained in GCP will send with the setting of regC6.

Reg	Register Name	bit	Definition	Default Value
C1	REGAVMute	0	Set AVMute ‘0’: not AVMute ‘1’: AVMute	1
	REGBlueScrMute	1	Set Blue screen output when AVMute=’1’	0
C6	REGPktGenCtrlRpt	1	Repeat General Control packet ‘0’: send once ‘1’: one for each field	0
	REGPktGenCtrlEn	0	Enable General Control packet ‘0’ : disable ‘1’: enable	0

Video Pixel Repetition

AVI Packet				
reg	Name	bit	Definition	Default Value
15C	REGPktAVIInfoPR[3:0]	3:0	0000 – 1X (no repetition) 0001 – 2X (one repetition) 0011 – 4X otherwise – no define	0x0

In HDMI and CEA861/D, there is pixel repetition defined in AVI Infoframe databyte[5][3:0]. IT6613 implement hardware repetition in AVI Infoframe databyte[5]. If the source video signal is the original timing (for example, 480i with 720x480i@60Hz with 13.5MHz), to write Infoframe is OK.

However, if the input source is not follow the CEA timing, or duplicate clock at first, then the pixel repetition function of AVI Infoframe should be disabled and set repetition manually with reg59[4] = '1'. When reg59[4] = '1', the pixel repetition will refer to reg59[7:6] values as the following table.

reg	Name	bit	Definition	Default Value
59	REGManualPLLPR	7:6	VCLK frequency depends on REGManualPLLPR when REGDisLockPR='1' 00: 1x 01: 2x 11: 4x	0
	REGEnTxCnt	5	Enable TxCLK to count REFCLK '1' : enable '0' : disable	0
	REGDisLockPR	4	'1' – Pixel repetition refer to the setting in reg59[7:6]. '0' – Pixel repetition refer to th setting oin reg15C[1:0]	0
	REGVidLatEdge	3	Video Data Latch Edge	
	REGAudDiv	1:0	Audio down-sampling selection "00" : no down-sampling "01" : divided by 2 "10" : no defined "11" : divided by 4	0x0

Invert Video Data Latch Edge

The quality of video need the data latch edge setting depends on the PCB layout. The video data latch edge setting is in reg59[3].

reg	Name	bit	Definition	Default Value
59	REGManualPLLPR	7:6	VCLK frequency depends on REGManualPLLPR when REGDisLockPR='1' 00: 1x 01: 2x 11: 4x	0
	REGEnTxCnt	5	Enable TxCLK to count REFCLK '1' : enable '0' : disable	0
	REGDisLockPR	4	'1' – Pixel repetition refer to the setting in reg59[7:6]. '0' – Pixel repetition refer to th setting oin reg15C[1:0]	0
	REGVidLatEdge	3	Video Data Latch Edge	0
	REGAudDiv	1:0	Audio down-sampling selection "00" : no down-sampling "01" : divided by 2 "10" : no defined "11" : divided by 4	0x0

Chap 6 Program Audio Mode

When HDMI video mode is ready and the TMDS clock is higher than 27MHz, the audio of IT6613 can be enabled.

To enable audio, regE0[3:0] should be set to zero at first, then enable audio reference clock (reg04[4] = '0') and audio FIFO (reg04[2] = '0'). Then program the audio clock recovery, audio format register (except audio channel enable), audio channel status, then enable audio channel enable while input clock stable (if SPDIF).

IEC60958 Programming

This section describes the general setting of IT6613 audio. The sequence to program the audio should following the steps:

1. Disable audio channel with regE0[3:0] = '0000'.
2. Enable audio reference clock and audio FIFO with reg04[4][2] = '0' '0'
3. Depends on the audio sample clock and TMDS clock, program the N/CTS.
4. Program audio format in regE0[7:4], and regE1~regE5.
5. Program audio channel status if not refer the S/PDIF channel status (when regE3[4] = '0').
6. If the audio sample frequency provided by system is not reliable, count the correct audio frequency with auto generated CTS and the formula between N and CTS, and set the N and channel status by detected value while reg5F[5] = '1'.
7. If input is from I2S, regE0[3:0] = 'xxx1' depends on audio source number.
8. If input is from SPDIF, regE0[3:0] = '0001' when reg5F[5] = '1'.

Audio Clock Recovery (N/CTS)

IT6613 provides the auto generating clock time stamp (CTS) mechanism with regC5[0] = '0'.

Reg	Register Name	bit	Definition	Default Value
C5	REGPktAudNCTSSel	1	Audio CTS selection '0': hardware auto count '1': user defined	0
	REGSinglePkt	0	Single Packet mode '0': burst packet mode '1': signal packet mode	0

CTS/N registers

N/CTS Packet				
reg	Name	bit	Definition	Default Value
130	REGPktAudCTS[7:0]	7:0		
131	REGPktAudCTS[15:8]	7:0		
132	REGPktAudCTS[19:16]	3:0		
133	REGPktAudN[7:0]	7:0		0x80
134	REGPktAudN[15:8]	7:0		0x18
135	REGPktAudN[19:16]	3:0		0x0
135	REGPktAudCTSCnt[3:0]	7:4	Read Only	
136	REGPktAudCTSCnt[11:4]	7:0	Read Only	
137	REGPktAudCTSCnt[19:12]	7:0	Read Only	

Usually, the CTS do not need to program, when software provide 20bit N value, IT6613 automatically generates the CTS depends the formula:

$(CTS_{average}) = (TMDS\_Clock \times N)/(128 \times Fs)$

Where Fs is the input audio sample frequency, and TMDS_Clock is the clock of TMDS including the pixel repetition and deep color ratio. For example, in 1440×480i@24bit mode, the TMDS clock is 13.5MHz×2 = 27MHz ; and for 1080p60@36bit mode, the TMDS clock is 148.5MHz × 36/24 = 222.75MHz.

If the audio sample frequency is undetermined, software can provide a default N value (such as 0x1880), while the audio clock is locked, the follow the readback CTS and TMDS clock to count the actually Fs, then set the N with suggested value as following rule:

128×Fs / 1500Hz <= N <= 128×Fs / 300Hz ,

therefore, a recommended optimal value of N comes by

N = 128×Fs / 1000Hz

Recommended N and CTS are as following table:

	32KHz	
TMDS Clock(MHz)	N	CTS
25.2/1.001	4576	28125
25.2	4096	25200
27	4096	27000
27×1.001	4096	27027
54	4096	54000
54×1.001	4096	54054
74.25/1.001	11648	210937-210938
74.25	4096	74250
148.5/1.001	11648	421850
148.5	4096	148500
Other	4096	Measure

Recommended N and CTS for 44.1KHz and Multiples

	44.1KHz		88.2KHz		176.4KHz	
TMDS Clock(MHz)	N	CTS	N	CTS	N	CTS
25.2/1.001	7007	31250	14014	31250	28028	31250
25.2	6272	28000	12544	28000	25088	28000
27	6272	30000	12544	30000	25088	30000
27×1.001	6272	30030	12544	30030	25088	30030
54	6272	60000	12544	60000	25088	60000
54×1.001	6272	60060	12544	60060	25088	60060
74.25/1.001	17836	234375	35672	234375	71344	234375
74.25	6272	82500	12544	82500	25088	82500
148.5/1.001	8918	234375	17836	234375	35672	234375
148.5	6272	165000	12544	165000	25088	165000
Other	6272	Measured	12544	Measured	25088	Measured

Recommended N and CTS for 48KHz and Multiples

	48KHz		96KHz		192KHz	
TMDS Clock(MHz)	N	CTS	N	CTS	N	CTS
25.2/1.001	6864	28125	13728	28125	27456	28125
25.2	6144	25200	12288	25200	24576	25200
27	6144	27000	12288	27000	24576	27000
27×1.001	6144	27027	12288	27027	24576	27027
54	6144	54000	12288	54000	24576	54000
54×1.001	6144	54054	12288	54054	24576	54054
74.25/1.001	11648	140625	23296	140625	23296	140625
74.25	6144	74250	12288	74250	24576	74250
148.5/1.001	5824	140625	11648	140625	23296	140625
148.5	6144	148500	12288	148500	24576	148500
Other	6144	Measured	12288	Measured	24576	Measured

CTS will be automatically generated by regC5[1] = ‘0’. Therefore, the N value on the table above should be assigned only. However, sometime the CTS should be assigned manually. Write the CTS register, and write regC5 with the sequence:

regF8 = 0xC3

regF8 = 0xA5 // the password register is enabled and write protection is disabled.

regC5[1] = ‘1’

regF8 = 0xFF // enabling the write protection of regC5.

Audio Format

IT6613 programs linear pulse code modulation (LPCM) audio in registers regE0~regE5.

Audio format register definition

Reg	Register Name	bit	Definition	Default Value
E0	REGAudSWL[1:0]	7:6	00: 16 bits 01: 18 bits 10: 20 bits 11: 24 bits	11
	REGSPDIFTC	5		0
	REGAudSel	4	0: I2S 1: SPDIF	0
	REGAudioEn[3:0]	3:0	Enable Audio Source [0] for audio source 0 [1] for audio source 1 [2] for audio source 2 [3] for audio source 3 '0': disable '1': enable	0
E1	REGAudFullPkt	6	Enable audio full packet mode '0': not full packet mode '1': full packet mode	1
	REGAudLatEdge	5	0: use rising edge to sample WS and I2S 1: use falling edge to sample WS and I2S	0
	REGAudFmt[4:0]	[4:0]	REGAudFmt[0] 0: Standard I2S 1: 32-bit I2S REGAudFmt[1] 0: Left-justified 1: Right-justified REGAudFmt[2] 0: Data delay 1T correspond to WS 1: No data delay correspond to WS REGAudFmt[3] 0: WS='0' is left channel 1: WS='0' is right channel REGAudFmt[4] 0: MSB shift first 1: LSB shift first	0x01
E2	REGFifo3Sel[1:0]	7:6	Audio FIFO 3 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	11
	REGFifo2Sel[1:0]	5:4	Audio FIFO 2 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	10
	REGFifo1Sel[1:0]	3:2	Audio FIFO 1 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	01
	REGFifo0Sel[1:0]	1:0	Audio FIFO 0 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	00
E3	REGAudMulCh	7	Read only. Depends on REGPktAudEn	
	REGPktZeroCTS	6	Enable zero CTS value '0': disable '1': enable	0
	REGChStSel	4	Channel status selction '0': from user defined '1': from SPDIF interface	0
	REGS3RLChg	3	Audio source 3 R/L swap '0': not swap R/L channel '1': swap R/L channel	0

	REGS2RLChg	2	Audio source 2 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
	REGS1RLChg	1	Audio source 1 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
	REGS0RLChg	0	Audio source 0 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
E4	REGAudSPxFlat[3:0]	7:4	User defined audio flat bit [0] for source 0 [1] for source 1 [2] for source 2 [3] for source 3	0x0
	REGAudErr2Flat	3	Auto audio error to flat setting '0': disable '1': enable	1
	REGAudMute2Flat	2		0
E5	SpdifCompFit	4	Read Back.	0
	REGAudHBR	3	0: Low Bit Rate 1: High Bit Rate	0
	REG1Baud	1	0: I2S/SPDIF 1: DSD/DST 1 bit audio	0
E6	REGSACDBitInv	7	0: normal, 1: bit inverse	0
	REGSACDMode	6	0: normal 1: Phase mode	0

IT6613 supports audio with up to four sources (eight channels), and sample word length with maximum 24bits (16/18/20/24 bits). There are two input types of audio source interface, I2S or S/PDIF. If audio come from the I2S, regE0[4] should set as zero and regE0[3:0] should set with corresponding sources. The word length of audio sample should set in regE0[7:4]. If the source interface is S/PDIF, regE0[4] should be '1' and regE0[3:0] = '0001'. Under S/PDIF interface, audio channel enable should be enabled after audio clock locked by reg5F[5] = '1'. If using I2S interface, don't care the bit but only enable the audio channel.

Reg	Register Name	bit	Definition	Default Value
5F	IP_LOCK	7	R.	
	XP_LOCK	6	R.	
	OSFreqLock	5	R.	
	TxCLKCnt[11:8]	3:0	R. Ring_OSC counter readback.	

For LPCM audio, regE1 = 0x01 while audio source is only one and regE1 = 0x41 when audio source number is larger or equal to two.

RegE2 defined the mapping between input audio sources and output HDMI audio sample sources. As described in the table, usually set as 0xE4.

RegE3[3:0] defines the channel swapping, for each audio source, the corresponding bit defined if the right and left channel swapping.

RegE3[4] is only using under S/PDIF interface. If the input audio source is not in LPCM format, our audio channel status cannot set the related information but can refer the original holding channel status in the digital interface. If regE3[4] = '1', the channel status in reg191~reg198 will be ignored and refer to the original channel status of S/PDIF input.

RegE4[7:4] defined the flat bit of audio. While corresponding bit is set to '1', the audio source will become flat (as mute).

RegE5 and regE6 should be zero.

Audio Channel Status

We support IEC60958 channel status in reg191~reg198. As following table, the corresponding bit is defined in IEC60958-3 spec. Filled the registers with correct audio source number, sample frequency, audio sample word length, and the other information.

Audio Channel Status				
Reg	Name	bit	Definition	Default Value
191	REGAudChStD[2:0]	6:4	ICE60958-3 p9 bit[5:3] Additional format information depends on linear PCM audio mode: 5|4|3 ----- 000: 2 audio channels without pre-emphasis. 001: 2 audio channels with 50 μs/15 μs pre-emphasis. 010: reserved 011: reserved All other combination are reserved and shall not be used until further defined.	
	REGAudChStC	3	refer to ICE60958-3 p9 bit[2] 0: Software for which copyright is asserted. 1: Software for which no copyright is asserted.	
	REGAudNLPCM	2	refer to ICE60958-3 p9 bit[1] 1: None-PCM setting. 0: for I2S setting.	0
	REGAudMono	0	Monochrome bit 1: if there is only one audio source	0
192	REGAudChStCat	7:0	Audio category code groups. Refer to *IEC60958-3 5.3.2.* channel status byte 2 (bit 15-8)	
193	REGAudChStSrc	3:0	refer to *IEC60958-3 p11 bit 16-19* Source number, 0~15, 0 means don't take number into account.	
194	REGAudChStCH	3:0	refer to *IEC60958-3 p11 bit 23-20* single and dual channel operating modes are defined in *IEC60958-1* Channel number of source 0 L-channel	
198	REGAudChStCA	7:4	refer to *IEC60958 p12 bit 29~28* Clock accuracy 00 Level II 01 Level I 10 Level III 11 Interface frame rate not matched to sampling frequency.	
	REGAudChStFs	3:0	Sample frequency indicated in *IEC60958-3 p11 bit 24~27.* Sample frequency of software indicated 27..24 -------- 0000 44.1 KHz 1000 88.2 KHz 1100 176.4 KHz **1001 768KHz (For HBR)** 0110 24 Khz 0010 48Khz 1010 96Khz 1110 192KHz 0011 32KHz 0000 sampling frequency not indicated.	0x0

199	REGAudChStOFs	7:4	Sample frequency indicated in *IEC60958-3 p11 bit 24~27.* Original Sampling Frequency 27..24 --------- 1111 44.1 KHz 0111 88.2 KHz 0011 176.4 KHz **0110 768KHz** 1001 24 Khz 1101 48Khz 0101 96Khz 0001 192KHz 1100 32KHz 0000 sampling frequency not indicated.	
	REGAudChStWL	3:0	Audio sample word length 1101 21 bits 1011 24 bit 1001 23 bit 0101 22 bit 0011 20 bit 0001 Word length not inidicated 1100 17 bit 1010 20 bit 1000 19 bit 0100 18 bit 0010 16 bit 0000 Word length not indicated	

HBR (High Bit Rate)

For HBR setting : Black blank fold means the REGISTERS have to be programmed.

(1) Audio channel register

Reg	Register Name	bit	Definition	Default Value
E0	REGAudSWL[1:0]	7:6	00: 16 bits 01: 18 bits 10: 20 bits 11: 24 bits	11
	REGSPDIFTC	5		0
	REGAudSel	4	0: I2S/ **HBR from I2S** 1: SPDIF / **HBR SPDIF**	0
	REGAudioEn[3:0]	3:0	Enable Audio Source [0] for audio source 0 [1] for audio source 1 [2] for audio source 2 [3] for audio source 3 '0': disable '1': enable **** "0001" for HBR SPDIF** **** "1111" for HBR from I2S**	0
E5	SpdifCompFit	4	Read Back.	0
	REGAudHBR	3	0: Low Bit Rate **1: High Bit Rate**	0
	REG1Baud	1	**0: I2S/SPDIF (include HBR)** 1: DSD/DST 1 bit audio	0
E6	REGSACDBitInv	7	0: normal, 1: bit inverse	0
	REGSACDMode	6	0: normal 1: Phase mode	0

(2) Clock Control register

59	REGEnTxCnt	5	Enable TxCLK to count REFCLK '1' : enable '0' : disable	0

	REG10bLatEdge	4	Not used; FPGA only	
	REGVidLatEdge	3	Video Data Latch Edge	0
	REGAudDiv	1:0	Audio down-sampling selection **"00" : no down-sampling** "01" : divided by 2 "10" : no defined "11" : divided by 4	0x0

High bit rate audio cannot be down sampling, Reg59[1:0] must be '00'

(3) Audio infoFrame packet

Audio InfoFrame Packet				
168	*REGPktAudInfoCC[2:0]*	*2:0*	*Channel Number* *000 Refer to Stream Header* *001 2 channel* *010 3 channel* *011 4 channel* *100 5 channel* *101 6 channel* *110 7 channel* *111 8 channel*	
169	*REGPktAudInfoSF[2:0]*	*4:2*		
16B	*REGPktAudInfoCA[7:0]*	*7:0*	*Must "1F" for HBR from I2S* *Must "00" for HBR spdif*	*0x0*
16C	*REGPktAudInfoDM*	*7*		
16C	*REGPktAudInfoLSV[3:0]*	*6:3*		
16D	*REGPktAudInfoSUM[7:0]*	*7:0*		

(4)Audio channel status

Audio Channel Status				
Reg	Name	bit	Definition	Default Value
191	**REGAudChStD[2:0]**	6:4	ICE60958-3 p9 bit[5:3] Additional format information depends on linear PCM audio mode: 5|4|3 --- 000: 2 audio channels without pre-emphasis. 001: 2 audio channels with 50 μs/15 μs pre-emphasis. 010: reserved 011: reserved All other combination are reserved and shall not be used until further defined.	
	REGAudChStC	3	refer to ICE60958-3 p9 bit[2] 0: Software for which copyright is asserted. 1: Software for which no copyright is asserted.	
	REGAudNLPCM	2	**1: 61937/DSD/HBR** 0: for I2S setting.	0
	REGAudMono	0	Monochrome bit 1: if there is only one audio source **** must be '0' for HBR**	0
192	**REGAudChStCat**	7:0	Audio category code groups. Refer to *IEC60958-3 5.3.2.* channel status byte 2 (bit 15-8)	
193	REGAudChStSrc	3:0	refer to *IEC60958-3 p11 bit 16-19* Source number, 0~15, 0 means don't take number into account.	
194	REGAudChStCH	3:0	refer to *IEC60958-3 p11 bit 23-20* single and dual channel operating modes are defined in *IEC60958-1* Channel number of source 0 L-channel	
198	REGAudChStCA	7:4	refer to *IEC60958 p12 bit 29~28* Clock accuracy 00 Level II 01 Level I 10 Level III 11 Interface frame rate not matched to sampling frequency.	

	REGAudChStFs	3:0	Sample frequency indicated in *IEC60958-3 p11 bit 24~27.* Sample frequency of software indicated 27..24 -------- 0000 44.1 KHz 1000 88.2 KHz 1100 176.4 KHz **1001 768KHz (for HBR)** 0110 24 Khz 0010 48Khz 1010 96Khz 1110 192KHz 0011 32KHz 0000 sampling frequency not indicated.	0x0
199	REGAudChStOFs	7:4	Sample frequency indicated in *IEC60958-3 p11 bit 24~27.* Original Sampling Frequency 27..24 --------- 1111 44.1 KHz 0111 88.2 KHz 0011 176.4 KHz **0110 768 Khz (for HBR)** 1001 24 Khz 1101 48Khz 0101 96Khz 0001 192KHz 1100 32KHz 0000 sampling frequency not indicated.	
	REGAudChStWL	3:0	Audio sample word length 1110 21 bits 1101 24 bit 1100 23 bit 1010 22 bit 1001 20 bit 1000 Word length not inidicated 0110 17 bit 0101 20 bit 0100 19 bit 0010 18 bit 0001 16 bit 0000 Word length not indicated	

Summary :

(1) HBR from I2S

RegE0 = 0xCF
RegE3 = 0x80
RegE5 = 0x08
Reg59 = 0x08
Reg16B = 0x1F
Reg191 = 0x04
Reg198 = 0x09
Reg199 = 0x6B

(2) HBR from spdif

RegE0 = 0xD1
RegE3 = 0x10
RegE5 = 0x08
Reg59 = 0x08
Reg16B = 0x00
Reg191 = 0x04
Reg198 = 0x09
Reg199 = 0x6B

DSD (One Bit Audio)

(1)Audio channel register

Reg	Register Name	bit	Definition	Default Value
E0	REGAudSWL[1:0]	7:6	00: 16 bits 01: 18 bits 10: 20 bits 11: 24 bits	11
	REGSPDIFTC	5		0
	REGAudSel	4	0: I2S/ HBR from I2S 1: SPDIF / HBR SPDIF	0
	REGAudioEn[3:0]	**3:0**	**Enable Audio Source** **[0] for audio source 0** **[1] for audio source 1** **[2] for audio source 2** **[3] for audio source 3** **'0': disable** **'1': enable**	0
	REGFifo2Sel[1:0]	5:4	Audio FIFO 2 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	10
	REGFifo1Sel[1:0]	3:2	Audio FIFO 1 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	01
	REGFifo0Sel[1:0]	1:0	Audio FIFO 0 source selection "00": from audio source 0 "01": from audio source 1 "10": from audio source 2 "11": from audio source 3	00
E3	REGAudMulCh	7	Read only. Depends on REGPktAudEn	
	REGPktZeroCTS	6	Enable zero CTS value '0': disable '1': enable	0
	REGChStSel	**4**	**Channel status selction** **'0': from user defined (HBR from I2S)** **'1': from SPDIF interface**	**0**
	REGS3RLChg	3	Audio source 3 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
	REGS2RLChg	2	Audio source 2 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
	REGS1RLChg	1	Audio source 1 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
	REGS0RLChg	0	Audio source 0 R/L swap '0': not swap R/L channel '1': swap R/L channel	0
E5	SpdifCompFit	4	Read Back.	0
	REGAudHBR	3	0: Low Bit Rate 1: High Bit Rate	0
	REG1Baud	1	0: I2S/SPDIF (include HBR) **1: DSD/DST 1 bit audio**	0
E6	REGSACDBitInv	7	0: normal, 1: bit inverse	0
	REGSACDMode	6	0: normal 1: Phase mode	0

(2)Clock Control register

59	REGEnTxCnt	5	Enable TxCLK to count REFCLK '1' : enable '0' : disable	0

	REG10bLatEdge	4	Not used; FPGA only	
	REGVidLatEdge	3	Video Data Latch Edge	0
	REGAudDiv	1:0	Audio down-sampling selection **"00" : no down-sampling** "01" : divided by 2 "10" : no defined "11" : divided by 4	0x0

For DSD, HBR must set as '0', Divider = no down sampling.

(3) Audio infoFrame packet

Audio InfoFrame Packet				
168	**REGPktAudInfoCC[2:0]**	2:0	Channel Number 000 Refer to Stream Header 001 2 channel 010 3 channel 011 4 channel 100 5 channel 101 6 channel 110 7 channel 111 8 channel	
169	REGPktAudInfoSF[2:0]	4:2		
16B	REGPktAudInfoCA[7:0]	7:0		0x0
16C	**REGPktAudInfoDM**	7		
16C	**REGPktAudInfoLSV[3:0]**	6:3		
16D	**REGPktAudInfoSUM[7:0]**	7:0		

(4)Audio channel status

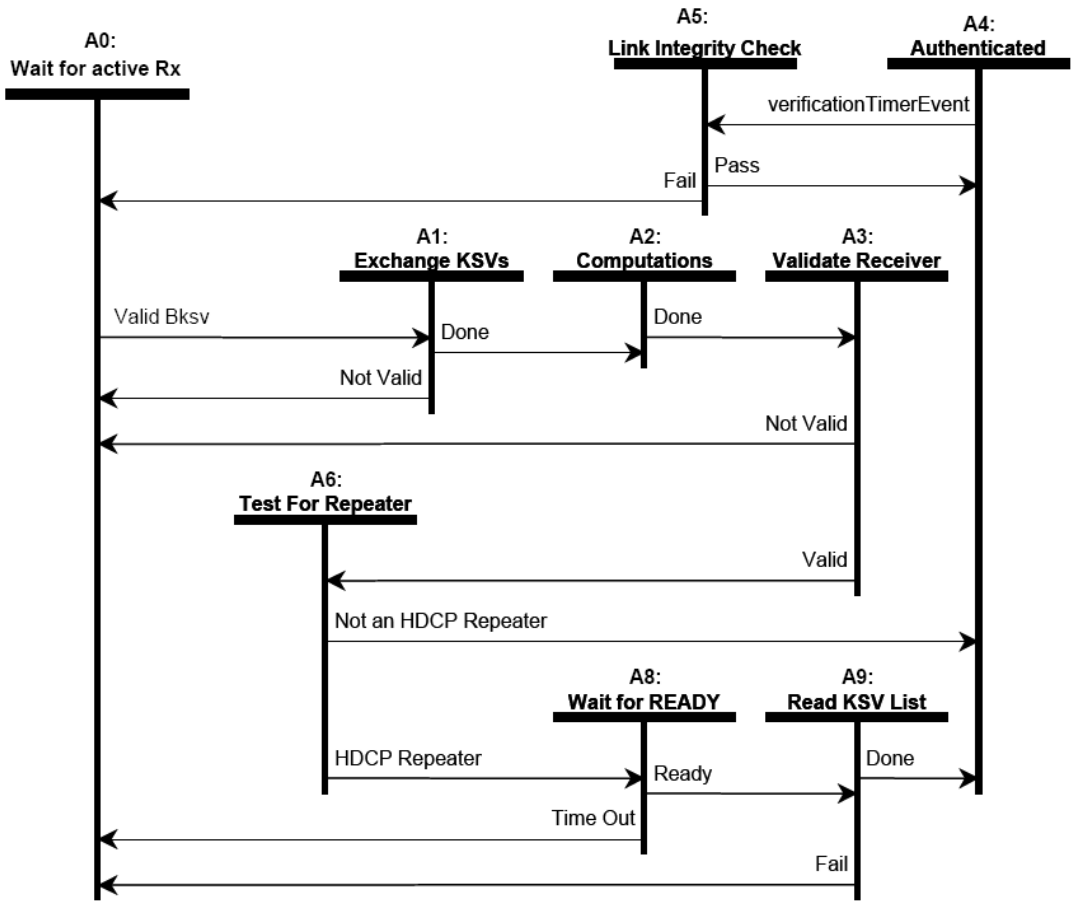
Audio Channel Status				
Reg	Name	bit	Definition	Default Value
191	**REGAudChStD[2:0]**	6:4	ICE60958-3 p9 bit[5:3] Additional format information depends on linear PCM audio mode: 5|4|3 --- 000: 2 audio channels without pre-emphasis. 001: 2 audio channels with 50 μs/15 μs pre-emphasis. 010: reserved 011: reserved All other combination are reserved and shall not be used until further defined.	
	REGAudChStC	3	refer to ICE60958-3 p9 bit[2] 0: Software for which copyright is asserted. 1: Software for which no copyright is asserted.	
	REGAudNLPCM	2	**1: 61937/DSD** 0: for I2S setting.	0
	REGAudMono	0	Monochrome bit 1: if there is only one audio source	0
192	**REGAudChStCat**	7:0	Audio category code groups. Refer to *IEC60958-3 5.3.2.* channel status byte 2 (bit 15-8)	
193	REGAudChStSrc	3:0	refer to *IEC60958-3 p11 bit 16-19* Source number, 0~15, 0 means don't take number into account.	
194	REGAudChStCH	3:0	refer to *IEC60958-3 p11 bit 23-20* single and dual channel operating modes are defined in *IEC60958-1* Channel number of source 0 L-channel	
198	REGAudChStCA	7:4	refer to *IEC60958 p12 bit 29~28* Clock accuracy 00 Level II 01 Level I 10 Level III 11 Interface frame rate not matched to sampling frequency.	

	REGAudChStFs	3:0	Sample frequency indicated in *IEC60958-3 p11 bit 24~27.* Sample frequency of software indicated 27..24 -------- 0000 44.1 KHz 1000 88.2 KHz 1100 176.4 KHz 0110 24 Khz 0010 48Khz 1010 96Khz 1110 192KHz 0011 32KHz 0000 sampling frequency not indicated.	0x0

For DSD setting :
RegE0 = 0xCF
RegE5 = 0x02
RegE6 = 0x00
Reg59 = 0x08
Reg191 = 0x04

Chap 7 Activate HDCP

IT6613 supports high-bandwidth digital content protection (HDCP) with Tx format. The state transition diagram is as the following figure:



HDCP Transmitter Authentication Protocol State Diagram

The transition from A0 to A4 and A9 to A4 are done by hardware functions, and repeater KSV List checking is implement by software. They are described in following.

HDCP Progress

Following registers are for HDCP programming.

Reg	Register Name	Bit	Definition	Default Value
1F	Reg_AnGen	0	Write this bit '1' to enable Cipher Hardware generating a random number. Write '0' to stop the Cipher Hardware. The generated Random number can be read back from register 30~37	0
20	REGAEnable1p1Feature	1	Enable HDMI Tx HDCP1.1 Feature	0
	REGCPDesired	0	'1' to enable HDCP	0
21	Rauthen_Fire	7:0	Write '1' to stare HDCP authentication process	X
22	REGList_Fail	1	Write this bit when process KSVList Check interrupt Routine. 1: HDCP Authentication FSM will return to wait and try state.	0

	REGList_Done	0	Write this bit when process KSVList Check interrupt routine. Write this '1' after PC check KSV FIFO list.	0
28~2F	An	63:0	Random number used at HDCP Authentication.	
30~37	VgenAn	63:0	Read only These 8 bytes are generated random number. To generate random number, see Reg1F[0]	
3B	BKSV [7:0]	7:0	Read only.	
3C	BKSV [15:8]	7:0	Read only.	
3D	BKSV [23:16]	7:0	Read only.	
3E	BKSV [31:24]	7:0	Read only.	
3F	BKSV [39:32]	7:0	Read only.	
40	BRi[7:0]	7:0	Read only.	
41	BRi[15:8]	7:0	Read only.	
42	BPj[7:0]	7:0	Read only.	
43	Bcaps[7:0]	7	HDMI_Reserved	
		6	HDCP Repeater capability.	
		5	KSV FIFO ready.	
		4	FAST. 1: the device supports 400KHz transfers.	
		3	reserved. must be zero.	
		2	reserved. must be zero.	
		1	1: HDCP 1.1 Features. support HDCP Enhanced encryption status signaling (EESS), Advance Cipher, and Enhanced Link Verification options.	
		0	1: Fast reauthenticagtion. When set to 1, the receiver is capable of receiving (unencrypted) video signal during the session re-authentication. All HDMI-capable receivers shall be capable of performing the fast re-authentication even if this bit is not set. This bit does not change while the HDCP receiver is active.	
45	Bstatus[15:8]	7	reserved 0.	
		6	reserved 0.	
		5	Reserved for future possible HDMI used.	
		4	HDMI_Mode 1: HDMI mode. 0: DVI mode.	
		3	MAX_CASCADE_EXCEEDED Topology error indicator. 1: more than seven levels of video repeater have been cascaded together.	
		2:0	Three-bit repeater cascade depth.	
44	Bstatus[7:0]	7	1: more than 127 downstream devices or KSV fifo.	
		6:0	Total number of attached downstream string devices.	
46	Rauthenticated	7	R	
	RautheFailStatus	6:0	*Fail Status for debug*	

To start HDCP authentication, the action for the following steps are as descripted.

A0 – Wait for active Rx

For starting the HDCP authentication, software must judge

1. Sink is a HDCP sink device
2. Sink have valid BKSV
3. (For HDMI requirement) Sink is HDMI device

Switch DDC master to PC-Host, then get BCaps and BStatus. If the BCaps exists and valid, then get BKSV, check if BKSV contains 20 bits as '1' and the other 20 bits as '0', and check it is not revoked by source SRM (*check the HDCP specification, chap 3.*). If BStatus[12] = '1', the HDCP sink is in HDMI mode.

If the BKSV checking is valid and is a HDMI sink, then start HDCP authentication by the following

steps:

1. Enable HDCP circuit by reg04[0] = ‘0’.
2. Generate a pseudo random number An by reg1F = 0x01 for a period time, then stop the cipher to fix the number.
   Read the eight bytes random number from reg30~reg37, then write them to reg28~reg2F.
3. Set reg20[1] = ‘1’ if sink support HDCP 1.1 feature while BCaps[1] = ‘1’; and set reg20[0] = ‘1’ to enable the HDCP.
4. Set reg22 = 0x00.
5. Set interrupt mask reg0A[2:0] = ‘000’ to listen authentication done, authentication fail, and authentication wait for repeater KSV list checking. Set reg0C[4:2] = ‘111’ and reg0E[0] = ‘1’ to clear all HDCP interrupt status before HDCP authentication start.
6. To start hardware authentication, the DDC master should be changed to HDCP core. Set reg10 = 0x00.
7. Write any value (we usually write 0xFF) to reg21 to fire HDCP authentication, IT6613 will do A1~A3 automatically until it issues a interrupt with status responded in any bit of reg07[2:0] with value ‘1’.

Whenever int07[1] = ‘1’ after HDCP authentication fired, the authentication is fail. If sink is a pure receiver, reset HDCP with reg04[0] = ‘1’ and reg20 = ‘00’, then restart the authentication; if sink is a repeater, set reg22 = 0x3, then set reg04[0] = ‘1’, reg20 = ‘00’ and reg22 = 0x00.

If sink is a receiver, after hardware exchange the KSV and computed the information successfully, reg47[7] will be ‘1’ and int07[1] = ‘1’ with interrupt.

If sink is a repeater, int07[2] will be ‘1’ after the information exchanging and computation done successfully. Then the status will be.

Handling the state transition for A0 to A6 should be interrupt handler.

A1 – Exchange KSVs

The KSV exchanging is done by hardware automatically.

If any interrupt occurs with int07[1] = ‘1’, switch state to A0 and reset HDCP circuit.

A2 – Computations

The computations are done by hardware automatically.

If any interrupt occurs with int07[1] = ‘1’, switch state to A0 and reset HDCP circuit.

A3 – Validate Receiver

The validate receiver is done before authenticate start, thus it can be ignored.

If any interrupt occurs with int07[1] = ‘1’, switch state to A0 and reset HDCP circuit.

A6 – Test for Repeater

Switch state to A6 if interrupt received with int07[0] = ‘1’ or int07[2] = ‘1’. If sink is receiver only, the interrupt report authentication done successfully and direct switch state to A4; if sink is repeater, the interrupt with int07[2] = ‘1’ will be received and need to wait for KSV list ready by polling BCaps. Switch the state to A8 and switch DDC master to PC-Host with reg10=0x01.

A8 – Wait for Ready

On the state, software should keep polling BCaps with KSV FIFO Ready bit (BCaps[5] = ‘1’). If the

time is out of 5 seconds after A6 without KSV FIFO ready, the authentication should be fail and turned to A0. Before fail it, reg22 should be 0x03 to turn off the state machine inner chip.

If BCaps[5] = '1', it means the KSV FIFO is ready. For some sink implementation, we suggest to wait 500 milliseconds after KSV FIFO ready, then switch state to A9.

A9 – Read KSV List

While the KSV FIFO ready responded in BCaps[5] = '1', software should get KSV FIFO list via DDC command. The downstream count (not include the repeater itself) should be between 1 to 6.

To get KSV FIFO list with count = downstream × 5.

After get KSV FIFO list, do SHA-1 encoding with KSV List, BStatus, and M0 in 8 bit value which computed by HDCP hardware cipher. It will get a 20-byte V from the SHA-1 encoding.

Reg	Name	Bit	Description	Default Value
50	SHASel[2:0]	2:0	See SHA_Rd_ByteX registers below	
51	SHA_Rd_Byte1[7:0]	7:0	V0h[7:0] when SHASel="000" V1h[7:0] when SHASel="001" V2h[7:0] when SHASel="010" V3h[7:0] when SHASel="011" V4h[7:0] when SHASel="100" Mi[7:0] when SHASel="101"	
52	SHA_Rd_Byte2[7:0]	7:0	V0h[15:8] when SHASel="000" V1h[15:8] when SHASel="001" V2h[15:8] when SHASel="010" V3h[15:8] when SHASel="011" V4h[15:8]when SHASel="100" Mi[15:8] when SHASel="101"	
53	SHA_Rd_Byte3[7:0]		V0h[23:16] when SHASel="000" V1h[23:16] when SHASel="001" V2h[23:16] when SHASel="010" V3h[23:16] when SHASel="011" V4h[23:16] when SHASel="100" Mi[23:16] when SHASel="101"	
54	SHA_Rd_Byte4[7:0]	7:0	V0h[31:24] when SHASel="000" V1h[31:24] when SHASel="001" V2h[31:24] when SHASel="010" V3h[31:24] when SHASel="011" V4h[31:24] when SHASel="100" Mi[31:24] when SHASel="101"	
55	Aksv_Rd_Byte5[7:0]	7:0	Mi[39:32] when SHASel="000" Mi[47:40] when SHASel="001" Mi[55:48] when SHASel="010" Mi[63:56] when SHASel="011"	

The 64bit M0 value is get from

```
reg50 = 0x05 ; M0[7:0]   = reg51 ;
reg50 = 0x05 ; M0[15:8]  = reg52 ;
reg50 = 0x05 ; M0[23:16]  = reg53 ;
reg50 = 0x05 ; M0[31:24]  = reg54 ;
reg50 = 0x00; M0[39:32]  = reg55;
reg50 = 0x01; M0[47:40]  = reg55;
reg50 = 0x02; M0[55:48]  = reg55;
reg50 = 0x03; M0[63:56]  = reg55;
```

Then get the V' from sink, if the V' is equal to V in all bytes, the authentication is done.

If anything error occurs, set reg22 = 0x3 to cancel the state machine for repeater checking, then reset HDCP circuit with reg04[0] = '1', and switch the state to A0 for authentication fail.

If all items are successful, set reg10 = 0x00 for switching DDC master to HDCP core, and set reg22 with 0x01 for continuing the authentication, and switch state to A4 for successfully authenticated.

A4 – Authenticated

In this state, HDCP is authenticated and the output data should be encrypted. We suggested to set reg0A[1] = ‘1’ to turn off the interrupt for authenticated done, by ignoring the handling for each authenticated successfully by Ri==Ri’ check of A5.

A5 – Link Integrity Check

For each 128 frame HDCP source should collect next Ri to do link integrity check. IT6613 does it automatically after HDCP authentication fired, and will issue a authentication done interrupt if reg0A[1] is ‘0’ when link integrity check successfully, and received a fail interrupt in reg07[0] if reg0A[0] = ‘0’. Whenever received this interrupt, HDCP source state should switch back to A0.

Reset HDCP

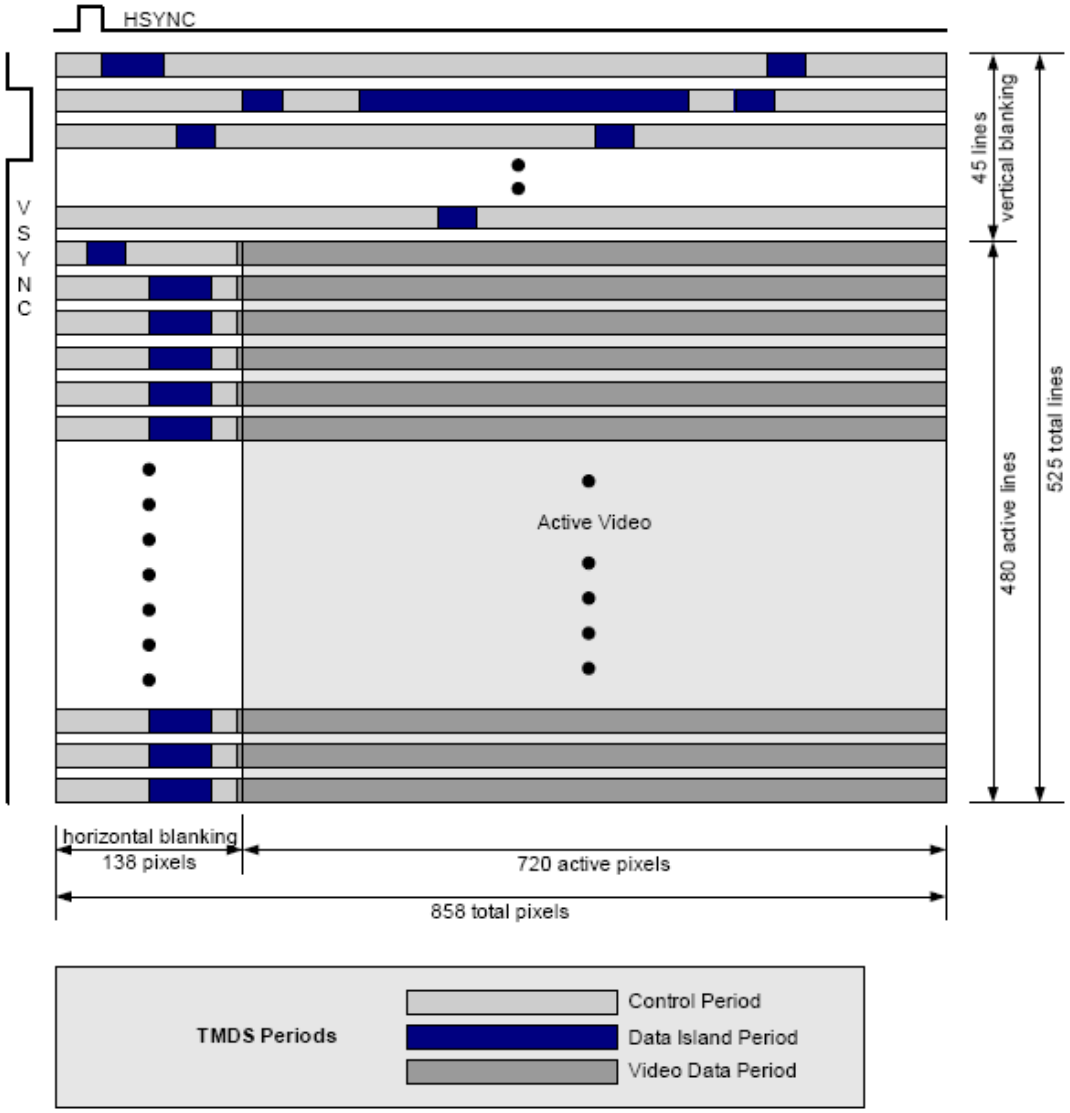
To reset HDCP, reg04[0] = ‘1’, reg20 = ‘0’. If state is in A6~A9, reg22 should write 3 at first, then write to zero.

Encryption

IT6613 enabled output encryption while HDCP done, and blanked while HDCP fail when HDCP is enabled and reg20[0] = ‘1’. If reg20[0] = ‘0’, HDCP cannot start.

Chap 8 HDMI Infoframe/Package

Unlike DVI mode, the TMDS link contains audio sample packets (programmed with IT6613 audio registers) and auxiliary data in data island period. The auxiliary data describe the video type, color information, audio auxiliary information, mute and other informations. As following figure (*refer to HDMI specification 1.3 section 5.1.2*)



Informative Example: TMDS periods in 720x480p video frame

Before enabling the HDMI package on IT6613, the mode should be switched to HDMI mode with regC0[0] = '1'.

Reg	Register Name	bit	Definition	Default Value
C0	REGHDMIMode	0	Set TX Mode '0': DVI mode '1': HDMI mode	0

The programming of HDMI auxiliary information is described in following sections.

HDMI Packet and CEA861/D Infoframe

An HDMI packet includes a 24-bit packet header, and a 28 bytes length packet body. A packet header includes an 8-bit Packet Type and 16 bits of packet-specific data.

The packet header format is as following table:

Byte / Bits#	7	6	5	4	3	2	1	0
HB0	Packet Type							
HB1	Packet-specific data							
HB2	Packet-specific data							

A sink shall be able to receive, with no adverse effects, any packet defined in the HDMI 1.0 spec including any InfoFrame packet with an InfoFrame Type defined in CEA861/D.

The packet type of HDMI spec defined as the following table:

Packet Type Value		Packet Type
0x00		NULL Packet
0x01		Audio Clock Regeneration (N/CTS)
0x02		Audio Sample (Include LPCM and IEC 61937 compressed format)
0x03		General Control
0x04		ACP Packet
0x05		ISRC1 Packet
0x06		ISRC2 Packet
0x07		One Bit (DSD) Audio Sample Packet
0x08		DST Audio Packet (***IT6613 does not support***)
0x09		High Bit Rate (HBR) Audio Stream Packet (IEC 61937)
0x0A		Gamut Metadata Packet
0x80+ Infoframe Type	0x81	Vendor Specific Infoframe
	0x82	AVI Infoframe
	0x83	Source Product Descriptor (SPD) InfoFrame
	0x84	Audio InfoFrame
	0x85	MPEG Source InfoFrame

The relate information of each packet specification are defined in CEA861/D specification for infoframe and in HDMI 1.4 specification with the other packet. The enabling registers of HDMI packets are in RegC1~RegD0, and see the following section description.

General Control Package

General control packet defined in HDMI 1.3 spec section 5.3.6 controls the output color depth, Pixel Packing Phase, and AVMute set/clear. IT6613 automatically handle the pixel packing phase, and send the color depth and AVMute status set in regC1 as following table described.

Reg	Register Name	bit	Definition	Default Value
C1	REGColorDepth	6:4	Color Depth 000 – not indicated, treated as 8 bit (or 12bit on YCbCr422) 100 – 8 bits per channel, 24 bits per pixel 101 – 10 bits per channel, 30 bits per pixel 110 – 12 bits per channel, 36 bits per pixel	000
	REGPhaseReSync	3		0
	REGNoDefPhase	2		0
	REGBlueScrMute	1	Set Blue screen output when AVMute='1'	0
	REGAVMute	0	Set AVMute '0': not AVMute '1': AVMute	1
C6	REGPktGenCtrlRpt	1	Repeat General Control packet '0': send once '1': one for each field	0
	REGPktGenCtrlEn	0	Enable General Control packet '0' : disable '1': enable	0

To send regC6[0] = '1' will enable GCP once if regC6[1] = '0', and sends frequently while regC6[1] = '1'.

Infoframe

The infoframe defined in CEA861/D which implemented in HDMI are as following

Byte / Bit #	7	6	5	4	3	2	1	0
HB0	1	Info Frame ID in CEA861/D						
HB1	Infoframe Version							
HB2	0	0	0	Infoframe Length				
PB0	Checksum							
PB1	Data byte 1 defined in CEA861/D nfoframe							
PB2	Data byte 2 defined in CEA861/D Infoframe							
PB3 .. PB26	-							
PB27	Data byte 27 defined in CEA861/D Infoframe							

The checksum of infoframe is the two's complement of byte summary HB0~HB2 and PB1~PB27.

$$PB0 = 0 - (\sum_{i=0}^{2} HBi + \sum_{i=1}^{27} PBi)$$

IT6613 provides determined registers for AVI Infoframe, audio infoframe, and Mpeg source infoframe identically, and general purpose registers in reg138~reg156 controlling with regC9.

AVI Infoframe

IT6613 provides AVI infoframe with version 2 in reg158~reg165.

Reg	Register Name	bit	Definition	Default Value
CD	REGPktAVIInfoRpt	1	Repeat AVI InfoFrame packet '0': send once '1': one for each field	0
	REGPktAVIInfoEn	0	Enable AVI InfoFrame packet '0': disable '1': enable	0

AVI Infoframe controlling registers is regCD with enabling bit in regCD[0], and repeating bit for each field while regCD[1] = '1'.

The infoframe data byte defined in following byte, which checksum in PB0 is defined in reg15D, and only 13 bytes valid in AVI infoframe.

AVI Packet					
Reg	**Register Name**	**bit**	**Definition**	**Default Value**	**PB**
158	REGPktAVIInfoY[1:0]	6:5	Output HDMI Color Space 00 – RGB444 mode 01 – YCbCr422 mode 10 – YCbCr444 mode		1
	REGPktAVIInfoA	4			
	REGPktAVIInfoB[1:0]	3:2			
	REGPktAVIInfoS[1:0]	1:0			
159	REGPktAVIInfoC[1:0]	7:6			2
	REGPktAVIInfoM[1:0]	5:4			
	REGPktAVIInfoR[3:0]	3:0			
15A	REGPktAVIInfoITC	7			3
	REGPktAVIInfoEC[2:0]	6:4			
	REGPktAVIInfoQ[1:0]	3:2			
	REGPktAVIInfoSC[1:0]	1:0			
15B	REGPktAVIInfoVIC[6:0]	6:0			4
15C	Reserved	7:4		0000	5
	REGPktAVIInfoPR[3:2]	3:2		0	
	REGPktAVIInfoPR[1:0]	1:0	Set the pixel repetition if reg59[4] = '0' 00 – no repetition 01 – 2X repetition 11 – 4X repetition	0x0	

			otherwise – undefined		
15D	**REGPktAVIInfoSUM[7:0]**	**7:0**	**The checksum of infoframe**		**0**
15E	REGPktAVIInfo06PB[7:0]	7:0			6
15F	REGPktAVIInfo07PB[7:0]	7:0			7
160	REGPktAVIInfo08PB[7:0]	7:0			8
161	REGPktAVIInfo09PB[7:0]	7:0			9
162	REGPktAVIInfo10PB[7:0]	7:0			10
163	REGPktAVIInfo11PB[7:0]	7:0			11
164	REGPktAVIInfo12PB[7:0]	7:0			12
165	REGPktAVIInfo13PB[7:0]	7:0			13

The checksum in reg15D = 0 - (sum of (reg158~reg15C) + sum of (reg15E~reg165) + 0x82 + 2 + 0x0D).

Audio Infoframe

Audio infoframe defined in CEA861/D can be enabled by regCE[0] and send repeating with regCE[1] = '1'.

Reg	Register Name	bit	Definition	Default Value
CE	REGPktAudInfoRpt	1	Repeat Audio InfoFrame packet '0': send once '1': one for each field	0
	REGPktAudInfoEn	0	Enable Audio InfoFrame packet '0': disable '1': enable	0

IT6613 provides audio infoframe with version 1 only in reg168, reg169, reg16B, and reg16C, and the other six byte defined in CEA861/D are assumed as zero. Checksum byte is defined in reg16D.

Audio InfoFrame Packet					
Reg	**Register Name**	**bit**	**Definition**	**Default Value**	**PB**
168	REGPktAudInfoCC[2:0]	2:0	Channel Number 000 Refer to Stream Header 001 2 channel 010 3 channel 011 4 channel 100 5 channel 101 6 channel 110 7 channel 111 8 channel		1
169	REGPktAudInfoSF[2:0]	4:2	Sampling Frequency 000 Refer to stream header 001 32KHz 010 44.1KHz(CD) 011 48KHz 100 88.2KHz 101 96KHz 110 176.4KHz 111 192KHz		2
16B	REGPktAudInfoCA[7:0]	7:0	refer to the CEA861/D definition.	0x0	4
16C	REGPktAudInfoDM	7	DM_INH Down-mix Inhibit Flag '0' Permitted or no information about any assertion of this '1' Prohibited		5
	REGPktAudInfoLSV[3:0]	6:3	Level shift values of db		
16D	REGPktAudInfoSUM[7:0]	7:0			0

reg16D = 0 – (reg168+reg169+reg16B+reg16C+0x84+0x01+0x0A).

MPEG InfoFrame

Mpeg source infoframe is enabled in regD0[0] and send repeating on each field when regD0[1] = '1'.

Reg	Register Name	bit	Definition	Default Value
D0	REGPktMpgInfoRpt	1	Repeat MPEG InfoFrame packet '0': send once '1': one for each field	0
	REGPktMpgInfoEn	0	Enable MPEG InfoFrame packet '0': disable '1': enable	0

The data byte definition is in the following table, note the reg18A is not pure mpeg definition:

Reg	Name	bit	Definition	PB sequence
18A	REGPktMpgInfoFR	0	DB5[4]	5
	REGPktMpgInfoMF[1:0]	2:1	DB5[1:0]	
18B	REGPktMpgInfo01PB[7:0]	7:0	MB#0 (Mpeg Bit Rate Hz)	1
18C	REGPktMpgInfo02PB[7:0]	7:0	MB#1	2
18D	REGPktMpgInfo03PB[7:0]	7:0	MB#2	3
18E	REGPktMpgInfo04PB[7:0]	7:0	MB#3	4
18F	REGPktMpgInfoSUM[7:0]	7:0	checksum	0

The checksum in reg18F = 0 – (reg18B+reg18C+reg18D+reg18E) - (reg18A[0] * 16) - (reg18A[2:1])

SPD/VendorSpec/ISRC1/ISRC2/General Purpose

The registers named "NULL Packet" in IT6613 are for general purpose, which is enabled by regC9[0] and sent each field once when regC9[1] = '1'.

Reg	Register Name	bit	Definition	Default Value
C9	REGPktNullRpt	1	Repeat Null packet '0': send once '1': one for each field	0
	REGPktNullEn	0	Enable Null packet '0': disable '1': enable (mutual exclusive with ACP/ISRC1/ISRC2 packet)	0

The layout in reg138~reg156 is fully compatible to the HDMI packet format. There are three bytes of header and 28 bytes of packet body. For those types sharing these registers should enable exclusive. For software implement, to disable the packet with regC9[0], then write the content, then enable it again to implement multiple package types using these registers.

Reg	Name	Bit	Definition	
138	REGPktNull0Hdr[7:0]	7:0	HDMI Packet HB0	
139	REGPktNull1Hdr[7:0]	7:0	HDMI Packet HB1	
13A	REGPktNull2Hdr[7:0]	7:0	HDMI Packet HB2	
13B	REGPktNull00PB[7:0]	7:0	HDMI Packet PB0	
13C ~ 155	REGPktNull01PB[7:0] ~ REGPktNull26PB[7:0]	7:0	HDMI Packet PB1 ~ HDMI Packet PB26	
156	REGPktNull27PB[7:0]	7:0	HDMI Packet PB27	

ACP register

ACP packet can be programmed with regCA[0] = '1' and sending frequently with regCA[1] = '1'.

Reg	Register Name	bit	Definition	Default Value
CA	REGPktACPRpt	1	Repeat ACP packet '0': send once '1': one for each field	0

	REGPktACPEn	0	Enable ACP packet '0': disable '1': enable (mutual exclusive with Null/ISRC1/ISRC2 packet)	0

The data of ACP packet can put in reg16E~reg17E, as following registers:

Reg	Name	bit	Content	Default Value
16E	REGPktACPType[1:0]	1:0	ACP Type defined in HB1 0x00 = Generic Audio 0x01 = IEC 60958-Identified Audio 0x02 = DVD-Audio 0x03 = Super Audio CD	
16F	REGPktACP00PB[7:0]	7:0	Dependent upon ACP type value	
170	REGPktACP01PB[7:0]	7:0		
171	REGPktACP02PB[7:0]	7:0		
172	REGPktACP03PB[7:0]	7:0		
173	REGPktACP04PB[7:0]	7:0		
174	REGPktACP05PB[7:0]	7:0		
175	REGPktACP06PB[7:0]	7:0		
176	REGPktACP07PB[7:0]	7:0		
177	REGPktACP08PB[7:0]	7:0		
178	REGPktACP09PB[7:0]	7:0		
179	REGPktACP10PB[7:0]	7:0		
17A	REGPktACP11PB[7:0]	7:0		
17B	REGPktACP12PB[7:0]	7:0		
17C	REGPktACP13PB[7:0]	7:0		
17D	REGPktACP14PB[7:0]	7:0		
17E	REGPktACP15PB[7:0]	7:0		

The reg16E should be programmed with the value, and reg16F ~ reg17E should be filled by the value of ACP content.

Gamut

The gamut metadata packet is enabled by regC2[0] = '1', and the gamut data update while the regC2[1] = '1' after the gamut data written in reg1A0~reg1B4.

Reg	Register Name	bit	Definition	Default Value
C2	REGPktGBDEn	0	Enable Gamut Data Packet	0
	REGPktGBDUpd	1	Gamut Data update	0

IT6613 only provides the gamut boundary descriptions (GBD) profile 0 which send each gamut packet per field. The reg1A0 to reg1B4 indicate the PB00~PB20 for

Color Gamut					
Reg	Name	Bit	Definition		Default Value
1A0	REGPktGBDFF	7	Format Flag – Identifies whether subsequent data describes gamut range boundary or gamut vertices boundary. 0 – indicates vertices only description, 1 – indicates range description		
	REGPktGBDCP[1:0]	4:3	GDB Color Precision '00' – 8bit '01' – 10bit '10' – 12bit		
	REGPktGBDCS[1:0]	1:0	GBD Color space '00' – ITU BT709 '01' – xvYCC601 '10' – xvYCC709 '11' – XYZ		
			Format Flag = 0	Format Flag = 1	
1A1	REGPktBGD01Data[7:0]	7:0	Number Vertices H	Packed_Range_Data, packed range data according to following sequence: Min_Red_Data Max_Red_Data Min_Green_Data Max_Green_Data Min_Blue_Data Max_Blue_Data	
1A2	REGPktBGD02Data[7:0]	7:0	Number Vertices L		
1A3	REGPktBGD03Data[7:0]	7:0	Packed GBD Vertices data [0..VSIZE – 1]		
1A4	REGPktBGD04Data[7:0]	7:0			
1A5	REGPktBGD05Data[7:0]	7:0			
1A6	REGPktBGD06Data[7:0]	7:0			
1A7	REGPktBGD07Data[7:0]	7:0			
1A8	REGPktBGD08Data[7:0]	7:0			
1A9	REGPktBGD09Data[7:0]	7:0			
1AA	REGPktBGD10Data[7:0]	7:0			
1AB	REGPktBGD11Data[7:0]	7:0			

1AC	REGPktBGD12Data[7:0]	7:0			
1AD	REGPktBGD13Data[7:0]	7:0			
1AE	REGPktBGD14Data[7:0]	7:0			
1AF	REGPktBGD15Data[7:0]	7:0			
1B0	REGPktBGD16Data[7:0]	7:0			
1B1	REGPktBGD17Data[7:0]	7:0			
1B2	REGPktBGD18Data[7:0]	7:0			
1B3	REGPktBGD19Data[7:0]	7:0			
1B4	REGPktBGD20Data[7:0]	7:0			

For format flag = 0, VSIZE is the number of bytes in the Packed_GBD_Vertices_Data according to

$$VSIZE = INT(3 \times Number\_Vertices \times BGD\_Color\_Precision / 8 + 0.99999)$$

where INT() is a function returning the integer part of the number.

Data Packing –

GBD data is efficiently packed with each 8-, 10- and 12-bit value taking exactly 8-, 10- or 12-bits in the packet. The GBD_Color_Precision field specifies the packing and precision of the GBD data. The following two tables define the packing for 10- and 12-bit values using a representative sequence of values, A, B, C … with A_low representing the low-order bits and A_high, the high-order of value A.

10-bit packing

	7	6	5	4	3	2	1	0
0	A_high							
1	A_low				B_high			
2	B_low					C_high		
3	C_low						D_high	
4	D_low							
5	E_High							
6	E_low				F_High			
7	F_low					G_High ...		

12-bit packing

	7	6	5	4	3	2	1	0
0	A_high							
1	A_low					B_high		
2	B_low							
3	C_high							
4	C_low					D_high		
5	D_low							

Chap 9 3D Support

HDMI1.4 defined the 3D format and 3D infoframe in Vendor-Specific InfoFrame Packet.

The format are as following:

Table 8-10 HDMI Vendor Specific InfoFrame Packet Header

Byte ¥ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x81							
HB1	Version = 0x01							
HB2	0	0	0	Length = Nv				

Table 8-11 HDMI Vendor Specific InfoFrame Packet Contents

Packet Byte #	7	6	5	4	3	2	1	0
PB0	Checksum							
PB1	24bit IEEE Registration Identifier (0x000C03) (least significant byte first)							
PB2								
PB3								
PB4	HDMI_Video_Format			Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)
(PB5)	HDMI_VIC							
	3D_Structure				Reserved(0)			
(PB6)	3D_Ext_Data				Reserved(0)			
.. PB(Nv)	Reserved (0)							

- Length [5bits] This 5 bits field defines the length of HDMI vendor specific InfoFrame payload.
- HDMI_Video_Format [3bits] This value defines the structure of extended video formats exclusively defined within this HDMI specification.

Table 8-12 HDMI_Video_Format

Value [2…0]	description
000	No additional HDMI video format is presented in this packet.
001	Extended resolution format (e.g. used for 4K x 2K video) present. 1 byte of HDMI_VIC parameter value follows.
010	3D format indication present. 3D_Structure, and potentially 3D_Ext_Data, follows.
011 ~ 111	Reserved for future use

- HDMI_VIC [1byte] HDMI proprietary Video Format Identification Code. When transmitting any video format defined in this section 8.2.3.1, an HDMI Source shall set the HDMI_VIC field to the Video Code for that format. The HDMI_VIC value is defined in section 8.2.3.1.
- 3D_Structure [4bits] This 4 bit field defines the transmission format of 3D video data. The value of "0000" means the Frame packing structure described in this section. The value of "1000" means the Side-by-Side (Half) structure described in this section. The value of "0110" means the Top-and-Bottom[1] structure described in this section. For other values, see Appendix H (Table H-2).

Table 8-13 3D_Structure

Value	Meaning
0000	Frame packing
0001 ~ 0101	Reserved for future use
0110	Top-and-Bottom
0111	Reserved for future use.
1000	Side-by-Side (Half)
1001 ~ 1111	Reserved for future use

For Side-by-Side (Half), the original left and right pictures are sub-sampled to half resolution on the horizontal axis. Sub-sampled pictures are arranged in Side-by-Side layout. See Figure 8-5.

For Top-and-Bottom, the original full left and right pictures are sub-sampled to half resolution on the vertical axis. Sub-sampled pictures are arranged in Top-and-Bottom layout. See Figure 8-6.

- 3D_Ext_Data [4bits] The meaning of this field depends on the 3D_Structure value.
  If 3D_Structure is 1000 (Side-by-Side (Half)), the 3D_Ext_Data field is added in the HDMI Vendor Specific InfoFrame and shall be 00XX (i.e. from 0000 to 0011, horizontal sub-sampling). For other values, see Appendix H.
  If 3D_Structure is 1001~1111, the 3D_Ext_Data field is also added and indicates additional information about the 3D format.
  If 3D_Structure is 0000~0111, the 3D_Ext_Data field shall not be present.

In IT6613 use general packet registers to transmit the common type of HDMI packet.

At first, set regC9 = 0x03 to transmit the HDMI packet defined in reg:

138	REGPktNull0Hdr[7:0]		
139	REGPktNull1Hdr[7:0]		

13A	REGPktNull2Hdr[7:0]		
13B	REGPktNull00PB[7:0]		
13C	REGPktNull01PB[7:0]		
13D	REGPktNull02PB[7:0]		
13E	REGPktNull03PB[7:0]		
13F	REGPktNull04PB[7:0]		
140	REGPktNull05PB[7:0]		
141	REGPktNull06PB[7:0]		
142	REGPktNull07PB[7:0]		
143	REGPktNull08PB[7:0]		
144	REGPktNull09PB[7:0]		
145	REGPktNull10PB[7:0]		
146	REGPktNull11PB[7:0]		
147	REGPktNull12PB[7:0]		
148	REGPktNull13PB[7:0]		
149	REGPktNull14PB[7:0]		
14A	REGPktNull15PB[7:0]		
14B	REGPktNull16PB[7:0]		
14C	REGPktNull17PB[7:0]		
14D	REGPktNull18PB[7:0]		
14E	REGPktNull19PB[7:0]		
14F	REGPktNull20PB[7:0]		
150	REGPktNull21PB[7:0]		
151	REGPktNull22PB[7:0]		
152	REGPktNull23PB[7:0]		
153	REGPktNull24PB[7:0]		
154	REGPktNull25PB[7:0]		
155	REGPktNull26PB[7:0]		
156	REGPktNull27PB[7:0]		

Reg 138 = 0x81.

RegC9 = 0x03.

Chap 10 Software Interface

Required Customer Providing Interface

unsigned HDMITX_ReadI2C_Byte(unsigned char RegAddr)

Parameter:

RegAddr – The I²C sub address of I²C address 0x98/0x9A to get a register contain from IT6613.

Note:

The I²C sequence should be
Start – <I²C Address|Write> - <RegAddr> - Restart - <I²C Address> - <returned data> - End.
The following sequence
Start – <I²C Address> - <RegAddr> - **Stop-Start** - <I²C Address> - <returned data> - End will fail the EDID readback.

HDMITX_WriteI2C_Byte(unsigned char RegAddr, unsigned char Data)

Parameter:

RegAddr – The I²C sub address of I²C address 0x98/0x9A to get a register contain from IT6613.
Data – The data byte to write into IT6613 register of RegAddr.

HDMITX_ReadI2C_ByteN(unsigned char RegAddr, unsigned char pData[], int N)

Parameter:

RegAddr – The I²C sub address of I²C address 0x98/0x9A to get a register contain from IT6613.
pData[] – The data byte array to get data from IT6613 registers starting on RegAddr.
N – the count to read.

Note:

The I²C sequence should be
Start – <I²C Address|Write> - <RegAddr> - Restart - <I²C Address> - <returned data> - End.
The following sequence
Start – <I²C Address> - <RegAddr> - **Stop-Start** - <I²C Address> - <returned data> - End will fail the EDID readback.

HDMITX_WriteI2C_Byte(unsigned char RegAddr, unsigned char pData[], int N)

Parameter:

RegAddr – The I²C sub address of I²C address 0x98/0x9A to get a register contain from IT6613.
pData[] – The data byte array to put data into IT6613 registers starting on RegAddr.
N – the count to write.

I2C_Read_ByteN(unsigned char I²C_Addr, unsigned char Offset, unsigned char pData[], int N) ;

Parameter:

I²C_Addr – The I²C address of I²C salve to get data from.
RegAddr – The I²C sub address of I²C address I²C_Addr to get a data.
pData[] – The data byte array to get data from the salve on the address starting on RegAddr.
N – the count to read.

Note –

IT6613 need access I²C address 0xE0 to do internal judgement.

CAT Provided Interface

Data structure

```
//////////////////////////////////////////////////////////////////
// data structure
//////////////////////////////////////////////////////////////////
typedef struct _INSTANCE_STRUCT {

        BYTE I2C_DEV ; // don't care here.
        BYTE I2C_ADDR ; // don't care here.

        ////////////////////////////////////////////
        // Interrupt Type
        ////////////////////////////////////////////
        BYTE bIntType ;
        ////////////////////////////////////////////
        // Video Property
        ////////////////////////////////////////////
        BYTE bInputVideoSignalType ; // for Sync Embedded,CCIR656,InputDDR
        ////////////////////////////////////////////
        // Audio Property
        ////////////////////////////////////////////
        BYTE bOutputAudioMode ; // = 0 ;
        BYTE bAudioChannelSwap ; // = 0 ;

        BYTE bAuthenticated:1 ;
        BYTE bHDMIMode: 1;
        BYTE bIntPOL:1 ; // 0 = Low Active
        BYTE bHPD:1 ;

} INSTANCE ;
```

void HDMITX_InitInstance(INSTANCE *pInstance)

Parameter: pInstance – a pointer points to a INSTANCE data structure to set initial value of 6613.
Return: N/A.
Remark: This function provide a initial value of internal IT6613 setting. Need to be called before InitIT6613().

void InitIT6613();

Parameter: N/A
Return: N/A
Remark: Before use IT6613 HDMI transmitter, this function should be call at first.

void DisableIT6613()

Parameter: N/A
Return: N/A
Remark: Power down IT6613. When reenable IT6613, system need to call InitIT6613 again.

BOOL CheckHDMITX (unsigned char *pHPD, unsigned char *pHPDChange) ;

Parameter: pHPD – pointer to unsigned char HPD parameter for getting hot plug status from IT6613, which '1' for plugged and '0' for unplugged.
pHPDChange – pointer to unsigned char HPDChange parameter for getting hot plug status change status from IT6613, which '1' means the HPD status was changed, and '0' for otherwise.

Return: N/A
Remark: This function will check and update the internal status for IT6613 software. Especially for HPDChange, if hot-plug status changed and turned off, this function will turn off the analog front end of IT6613 until the next enable from EnableVideoOutput().
Main system program should program the HDMI Tx output when HPDChange and HPD are both '1'.

BOOL ProgramSyncEmbeddedVideoMode (unsigned char VIC, unsigned char bInputSignalType) ;

Parameter: VIC – a unsigned char video code for video mode defined in CEA 861/D spec.
bInputSignalType – For indicating input video timing format, which T_MODE_CCIR656 (1) for CCIR656 format and otherwise for 16bit sync embedded sync.
Return: TRUE – set successfully, FALSE – cannot find a valid VIC for setting, abort.
Remark: If use 16 bit sync embedded signal or CCIR656 format for IT6613 input signal format, system should call this function before EnableVideoOutput call to create signal from IT6613.

BOOL EnableVideoOutput(VIDEOPCLKLEVEL level, unsigned char inputColorMode, unsigned char inputVideoType, unsigned char outputColorMode, BOOL bHDMI) ;

Parameter: level – PCLK_HIGH – for TMDS clock over 80MHz
PCLK_MEDIUM – for TMDS clock less than 80MHz but over 20MHz
PCLK_LOW – for TMDS clock less than 20MHz (480i and 576i)

inputColorMode – indicating the input video mode, should be one of following:

Name	Value	Meaning
F_MODE_RGB24	0	RGB24bit input forma
F_MODE_YUV422	1	24bit, 16bit, or 8bit YCbCr422 input color mode.
F_MODE_YUV444	2	24bit YCbCr444 input color mode.
F_VIDMODE_ITU709_	0x10	if input color support ITU-709 color space, add 0x10 in inputColorMode parameter, otherwise for ITU601.
F_VIDMODE_16_235	0x20	if input color use 16-235 quantization for color converting, add this value to inputColorMode.

inputVideoType – indicating the input video signal format, should be the combination of following value:

Name	Value	Mean
T_MODE_CCIR656	1	Signal is CCIR656 mode, should be set with Sync embedded
T_MODE_SYNCEMB	2	Signal is a sync embedded mode.
T_MODE_INDDR	4	Signal is a double data rate input signal.

outputColorMode – output color use the same color parameter as inputColorMode to indicating the HDMI output color mode.

Name	Value	Mean
F_MODE_RGB24	0	RGB24bit input forma
F_MODE_YUV422	1	24bit, YCbCr422 input color mode.
F_MODE_YUV444	2	24bit YCbCr444 input color mode.

bHDMI – BOOL , TRUE for HDMI output and FALSE for DVI output.
Return: TRUE for setting successfully, FALSE for otherwise.
Remark: When HDMI sink is ready, call this function to enable video output from IT6613.

BOOL EnableAudioOutput(unsigned long VideoPixelClock,
unsigned char bAudioSampleFreq,
int ChannelNumber,
BOOL bSPDIF);

Parameter: VideoPixelClock – intput video pixel clock for generating N and CTS of audio sample package. In Hz unit.
bAudioSampleFreq– input audio sample clock for audio sample, in following value.

AUDFS_22p05KHz	=4
AUDFS_44p1KHz	=0
AUDFS_88p2KHz	=8
AUDFS_176p4KHz	=12
AUDFS_24KHz	=6
AUDFS_48KHz	=2
AUDFS_96KHz	=10
AUDFS_192KHz	=14
AUDFS_32KHz	=3
AUDFS_OTHER	=1

ChannelNumber – input audio channel, number should be indeed of 1 to 8.
bSPDIF – 'TRUE' for S/PDIF format input, and 'FALSE' for I2S format input.
Return: 'TRUE' for setting successfully. 'FALSE' for otherwise.
Remark: If HDMI sink support audio and output with HDMI mode, call this function to set audio format.

void DisableVideoOutput()

Parameter: N/A
Return: N/A
Remark: Disable the video output of IT6613.

void DisableAudioOutput()

Parameter: N/A
Return: N/A
Remark: Disable the audio of IT6613. Need to call EnableAudioOutput to restart the audio.

void SetOutputColorDepthPhase(unsigned char ColorDepth,unsigned char bPhase)

Parameter: ColorDepth – the output color depth on 24/30/36
bPhase – reserved here.
Return: N/A
Remark: When enable HDMI 1.3 deep color function, call this function at first before EnableVideoOutput.

BOOL GetEDIDData(int EDIDBlockID, unsigned char *pEDIDData);

Parameter: EDIDBlockID – the ID of requesting block of EDID.
pEDIDData – a pointer to a unsigned char array with 128 elements.
Return: TRUE for getting EDID data successfully, and FALSE for otherwise.
Remark: This is a interface to get EDID data from HDMI/DVI sink via IT6613 DDC function interface.
System core should handle the EDID parsing for gotten EDID data.

BOOL EnableHDCP(BOOL bEnable) ;

Parameter: bEnable – TRUE for setting HDCP, FALSE for disable HDCP.

Return: TRUE for enabling/disabling successfully, FALSE for otherwise.

BOOL EnableAVIInfoFrame(BOOL bEnable, unsigned char *pAVIInfoFrame);

Parameter: bEnable – TRUE for enabling and setting AVI InfoFrame, FALSE for disable AVI info frame sending of IT6613.
pAVIInfoFrame – a pointer to unsigned char array with AVI InfoFrame package defined in CEA861/B.
Return: TRUE for setting/disabling successfully, FALSE for otherwise.

BOOL EnableAudioInfoFrame(BOOL bEnable, unsigned char *pAudioInfoFrame);

Parameter: bEnable – TRUE for enabling and setting AUDIO InfoFrame, FALSE for disable AUDIO info frame sending of IT6613.
pAUDIOInfoFrame – a pointer to unsigned char array with AUDIO InfoFrame package defined in CEA861/B.
Return: TRUE for setting/disabling successfully, FALSE for otherwise.

void SetAVMute(BOOL bEnable) ;

Parameter: bEnable – TRUE for set AVMute of HDMI signal, sink should set audio and video mute when receiver the AVMUTE package, and FALSE for clear AVMUTE.
Return: N/A
Remark: system should use this function to disable screen output and audio output for switch color format or video output.

CAT Provided SubFunction.

void HDMITX_ChangeDisplayOption(
HDMI_Video_Type OutputVideoTiming,
HDMI_OutputColorMode OutputColorMode);

Parameter: OutputVideoTiming – indicate the output video timing which next using.

```
typedef enum tagHDMI_Video_Type {
    HDMI_Unkown = 0 ,
    HDMI_640x480p60 = 1 ,
    HDMI_480p60,
    HDMI_480p60_16x9,
    HDMI_720p60,
    HDMI_1080i60,
    HDMI_480i60,
    HDMI_480i60_16x9,
    HDMI_1080p60 = 16,
    HDMI_576p50,
    HDMI_576p50_16x9,
    HDMI_720p50,
    HDMI_1080i50,
    HDMI_576i50,
    HDMI_576i50_16x9,
    HDMI_1080p50 = 31,
    HDMI_1080p24,
    HDMI_1080p25,
    HDMI_1080p30,
} HDMI_Video_Type ;
```

OutputColorMode – indicate the next output color mode with following format:

```
typedef enum tagHDMI_OutputColorMode {
    HDMI_RGB444,
    HDMI_YUV444,
    HDMI_YUV422
} HDMI_OutputColorMode ;
```

Return: N/A
Remark: the function in IT6613_sys.c, provide system to call to change mode.
This function will not scale display mode but only notify which mode IT6613 should use.

void HDMITX_DevLoopProc();

Parameter: N/A
Return: N/A
Remark: the function which put into main loop in system main loop.

void ConfigAVIInfoFrame(BYTE VIC, BYTE pixelrep);

Parameter: N/A
Return: N/A
Remark: Configure the AVI infoframe data from ChangeDisplayOption().

void ConfigAudioInfoFrm().;

Parameter: N/A
Return: N/A
Remark: Configure the audio infoframe data from ChangeDisplayOption().

BOOL ParseEDID();

Parameter: N/A
Return: N/A
Remark: Provide a basic EDID parsing function, which