5725 Programming Guide
(For Internal Use Only)

NOTE

This document is TrueView 5725 programming guide for Tvia internal used only.

Reversion History

TrueView 5725 Programming Guide (internal version) History

Date	Version	By	Comments
09/14/2005	Version 0.1	Yanwei Yuan	First release
09/16/2005	Version 0.2	Yanwei Yuan	Added PIP function description Added reset de-bounce circuit description Added Sync input PAD diagram Added Crystal PAD diagram
10/13/2005	Version 0.3	Yanwei Yuan	Update the vds_proc hb & dis_hb timing

Content

1 Function Description

TrueView 5725 is a low pin count, low-cost of advanced and highly integrated Digital and Analog Video Display Processor providing the key features needed to design HD-READY or Progressive scan Televisions and low-end LCD TV.

1.1 Features

The TrueView 5725 provides the following features:

- ADC
- Clock recovery
- Decimation Filter
- Input Formatter
- Auto mode detection
- Motion adaptive de-interlace
- Memory FIFO Top level module
- OSD engine
- PIP control
- Video display processor
- Digital output
- Host interface & GPIOs
- PLL
- DAC
- Interrupt generator
- Miscellaneous

1.1.1 ADC

- The analog input range: 0.5v –1.0v p-p
- 3:1 analog input mux: RGB0/YUV0, RGB1/YUV1, and RGB2
- The maximal sample rate: 162MSPS
- Resolution: 8 bits
- Power supply: 3.3v +/- 10% for three ADCs and REF.
- Gain control resolution: 8 bits;
- Offset control resolution: 7 bits; offset range: -64LSB to 64LSB;
- Sync detect resolution: 5bits; comparator voltage range 10mv~320mv.
- DNL +/- 0.5 and INL +/- 2
- Programmable analog bandwidth

1.1.2 Clock-recovery Circuit

- H-sync input frequency: 15KHz to 110KHz.
- Output clock frequency: 5MHz to 175MHz.
- SOG separation support Standard SD, HD and Macro vision SOG source

1.1.3 Digital Input Formatter

- 24bit RGB/YUV input
- 8/16bit YUV input
- 8bit 656/601 input
- NTSC/PAL input
- 480P, 576P input
- VGA/SVGA/XVGA input
- 720P, 1080i, 1080P HD input
- Auto UV offset adjustment
- Non-linear horizontal/vertical scaling down

1.1.4 Auto mode detection

- NTSC/PAL, 480P/576P
- 720P/1080i (50/60Hz)
- VGA/SVGA/XVGA/SXGA (60/75/85Hz)
- Auto VGA polarity detection / correction
- Interrupt feature to MCU for fast mode access
- No signals, lost signals and unstable signals status detection

1.1.5 De-interlace

- HD 1080i support
- SD NTSC/PAL
- Direct Edge Correction De-interlace
- Motion adaptive de-interlace
- Background Noise Detection
- 3:2/2:2 pull-down detection

1.1.6 Memory Controller

- 2-8Mbyte memory support
- 16/32bit data access

1.1.7 Video Display Processor

Output formatter

- 480P, 576P, 720P, 1080I, 1080P
- Up to SXGA graphic output formats
- 100/120 interlace double scan
- 75Hz interlace single scan
- 50-75Hz scan rate conversion
- 15~80KHz horizontal frequency
- 16bit YPbPr digital output with syncs
- 24bit YPbPr/RGB digital output with LCD compatible

Video enhancement

- Black/White Level Expansion (BLE/WLE)
- Color Transition Improvement (DCTI)
- Dynamic Range expansion
- Brightness, Saturation, Contrast, HUE
- 2D Peaking

  - Color improvement
  - Non-linear saturation
  - Blue stretch
  - 3D noise reduction
  - Scan Velocity Modulation (SVM)
  - H/V-scaling up

□ **PIP**

  - Support board-level PIP using 2 5725 chips!

□ **OSD**

  - Simple OSD generator to support component video inputs

□ **Host Interface and I/O**

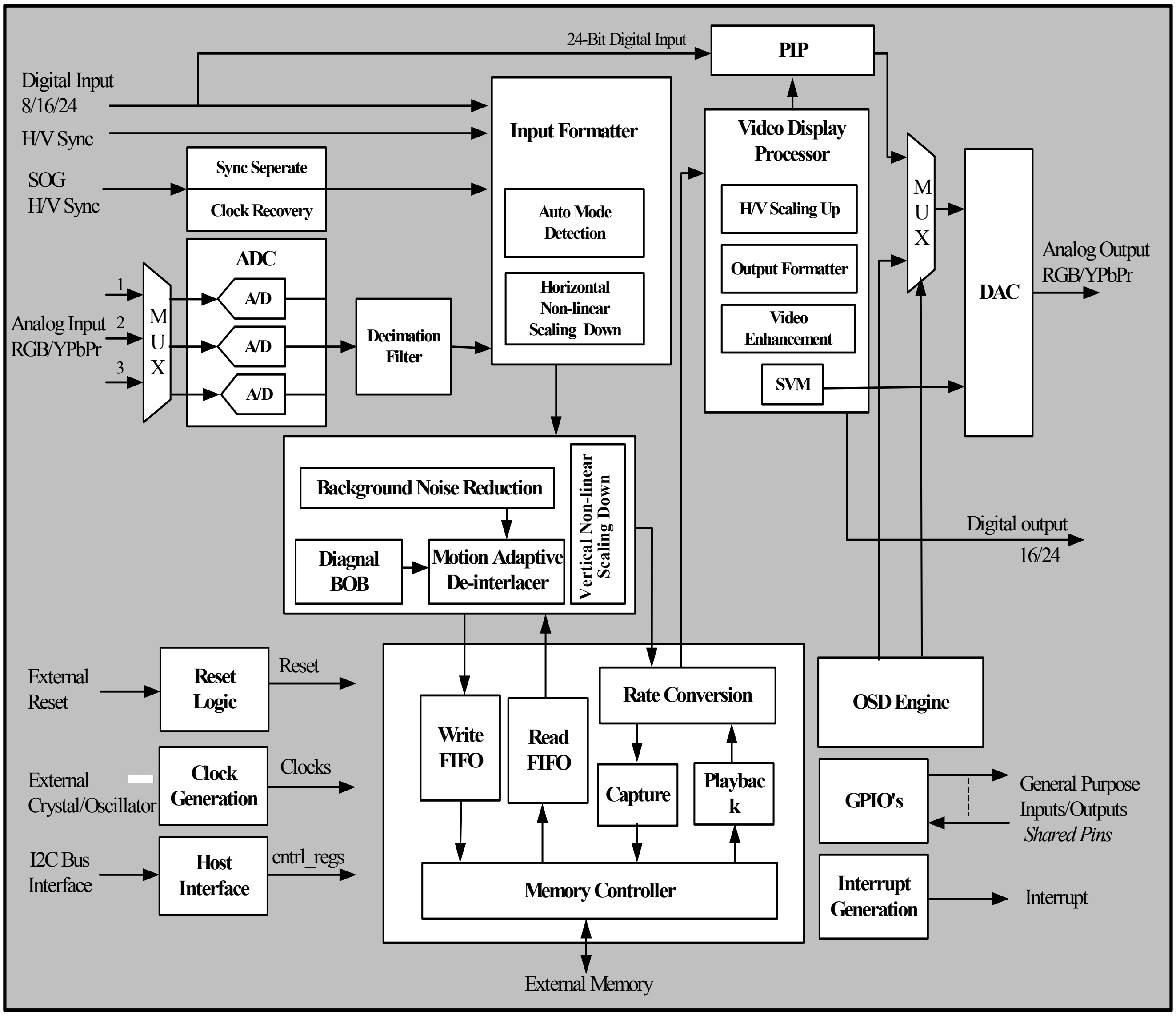
  - Two-wire serial interface
  - 8 GPIOs

□ **Quadruple 10bit DAC**

□ **Package and Technology**

  - 160-pin QFP
  - 0.18 micron, 3.3V / 1.8V technology

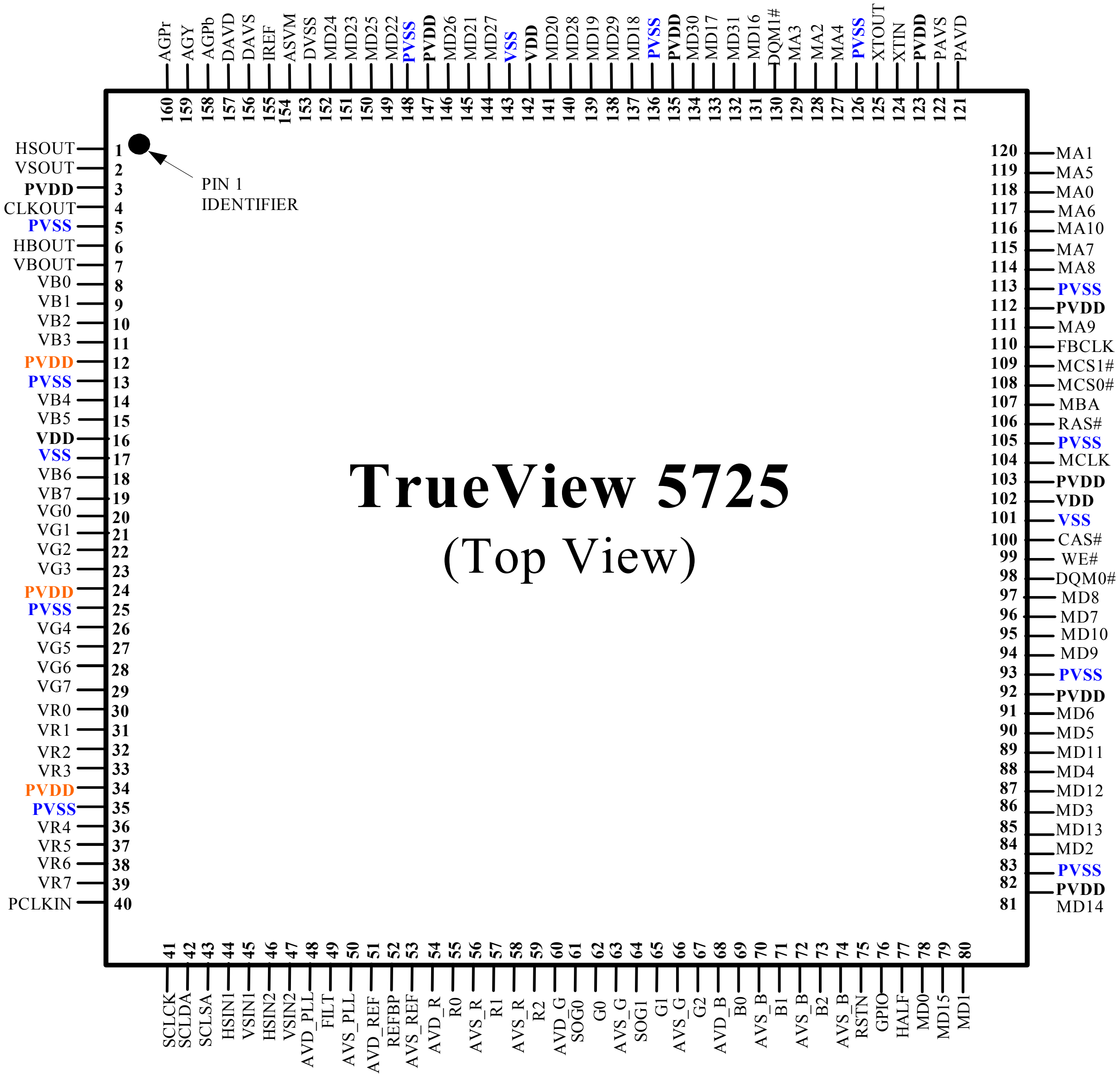
1.2 Block Diagram



2 Pin Information

2.1 Pin Diagram

True View 5725 is manufactured with a 160-pin QFP package. The pin locations are shown in below:



2.2 Pin Description

Pin types include the following:

I	Digital Input
IPD	Digital Input with pull-down
O	Digital Output
OPD	Digital Output with pull-down
IO	Digital Bi-directional (input/output)
IOPU	Digital Bi-directional (input/output) with pull-up
IOPD	Digital Bi-directional (input/output) with pull-down
AI	Analog Input
AO	Analog Output
DP	Digital Power
DG	Digital Ground
AP	Analog Power
AG	Analog Ground

2.2.1 Digital Video Input Pins

Pin Name	Pin #	Type	Pin Description
PCLKIN	40	IPD	Digital input video clock
HSIN1	44	I	Digital input video H-sync Shared with Analog input video H-sync1
VSIN1	45	I	Digital input video V-sync Shared with Analog input video V-sync1
VG [7:0]	29~26, 23~20	IOPD	Video Green Input data. • G data for 24bit RGB mode. • Y data for 4:4:4 mode • YUV data for 8bit 4:2:2 YUV mode. • Y/G data for 24bit PIP sub input Shared with digital video output data
VB [7:0]	19~18, 15~14, 11~8	IOPD	Video Blue Input data. • B data for 24bit RGB mode. • U data for 4:4:4 YUV mode. • UV data for 16bit 4:2:2 YUV mode. • U/B data for 24bit PIP sub input. • Y data for 16bit PIP sub input. Shared with digital video output data
VR [7:0]	39~36, 33~30	IOPD	Video Red Input data. • R data for 24bit RGB mode. • V data for 4:4:4 mode. • Y data for 16bit 4:2:2 mode. • V/R data for 24bit PIP sub input • U/V data for 16bit PIP sub input Shared with digital video output data
CLKOUT	4	IO	Display enable input from DVI/HDMI decoder Shared with video clock output & GPIO bit 7
HSOUT	1	IO	Input H-sync for PIP mode Shared with H-sync output
VSOUT	2	IO	Input V-sync for PIP mode Shared with V-sync output
HALF	77	IOPD	Half tone input Shared with GPIO bit 1

2.2.2 Digital Video Output Pins

Pin Name	Pin #	Type	Pin Description
CLKOUT	4	IO	Digital video display output clock Shared with display enable input & GPIO bit 7
HSOUT	1	IO	Digital video display output H-sync Shared with H-sync input for PIP mode
VSOUT	2	IO	Digital video display output V-sync Shared with V-sync output for PIP mode
VBOUT	7	IO	Display enable output in LCD application Shared with GPIO bit 6 and V-blank output
VG [7:0]	29~26, 23~20	IOPD	• Y/G data for 24bit digital output. Shared with digital video input data
VB [7:0]	19~18, 15~14, 11~8	IOPD	• UV data for 16bit digital output. • R/V data for 24bit digital output. Shared with digital video input data
VR [7:0]	39~36, 33~30	IOPD	• Y data for 16bit digital output. • B/U data for 24bit digital output. Shared with digital video input data

2.2.3 Analog Video input Pins

Pin Name	Pin #	Type	Pin Description
HSIN1	44	I	Analog video input H-sync1 Shared with digital video input H-sync
VSIN1	45	I	Analog video input V-sync1 Shared with digital video input V-sync
HSIN2	46	I	Analog video input H-sync2
VSIN2	47	I	Analog video input V-sync2
SOG0	61	AI	Green input 0 for Sync-On-Green sync tip clamping
SOG1	64	AI	Green input 1 for Sync-On-Green sync tip clamping
R0	55	AI	Analog input for Red input channel 0
G0	62	AI	Analog input for Green input channel 0
B0	69	AI	Analog input for Blue input channel 0
R1	57	AI	Analog input for Red input channel 1
G1	65	AI	Analog input for Green input channel 1
B1	71	AI	Analog input for Blue input channel 1
R2	59	AI	Analog input for Red input channel 2
G2	67	AI	Analog input for Green input channel 2
B2	73	AI	Analog input for Blue input channel 2
REFBP	52	AI	Internal reference bypass
AVD_R	54	AP	Analog power (3.3V)
AVS_R	56	AG	Analog ground
AVS_R	58	AG	Analog ground
AVD_G	60	AP	Analog power (3.3V)
AVS_G	63	AG	Analog ground
AVS_G	66	AG	Analog ground
AVD_B	68	AP	Analog power (3.3V)
AVS_B	70	AG	Analog ground

AVS_B	72	AG	Analog ground
AVS_B	74	AG	Analog ground
AVD_REF	51	AP	Analog power (3.3V)
AVS_REG	53	AG	Analog ground
AVD_PLL	48	AP	Analog power (3.3V) for PLLAD
FILT	49	AI	Connection for External Filters components for PLLAD
AVS_PLL	50	AG	Analog ground for PLLAD

2.2.4 Analog Video Output Interface Pins

Pin Name	Pin #	Type	Pin Description
HSOUT	1	IO	Analog video display output H-sync Shared with H-sync input for PIP mode
VSOUT	2	IO	Analog video display output V-sync Shared with V-sync output for PIP mode
HBOUT	6	IO	Analog video display output H-Blank Shared with GPIO bit 5
VBOUT	7	IO	Analog video display output V-Blank Shared with GPIO bit 6 and DE output for digital video output
AGPb	158	AO	Analog Blue/Pb output
AGY	159	AO	Analog Green/Y output
AGPr	160	AO	Analog Red/Pr output
ASVM	154	AO	Analog SVM output
IREF	155	AI	Full-scale adjust resistor
DVSS	96	AG	Big current GND
DAVS	99	AG	Analog ground for DAC
DAVD	100	AP	Analog power for DAC

2.2.5 Clock Generation Pins

Pin Name	Pin #	Type	Pin Description
XTALO	124	O	External crystal output.
XTALI	125	I	External crystal input.
PAVD	121	AP	Analog power (3.3V) for PLL648
PAVS	122	AG	Analog ground for PLL648

2.2.6 System Interface Pins

Pin Name	Pin #	Type	Pin Description
SCLSA	43	IOPU	Serial bus slave address selection, Shared with GPIO bit 2
SCLCK	41	I	Serial bus clock
SCLDA	42	IO	Serial bus data
RSTN	75	I	External asynchronous reset, low active
GPIO	76	IOPD	GPIO bit 0 Shared with Interrupt Output, low active

2.2.7 SDRAM Interface Pins

Pin Name	Pin #	Type	Pin Description
MD31	132	IOPD	Memory Data Bus [31:0]
MD30	134	IOPD	
MD29	138	IOPD	
MD28	140	IOPD	
MD27	144	IOPD	
MD26	146	IOPD	
MD25	150	IOPD	
MD24	152	IOPD	
MD23	151	IOPD	
MD22	149	IOPD	
MD21	145	IOPD	
MD20	141	IOPD	
MD19	139	IOPD	
MD18	137	IOPD	
MD17	133	IOPD	
MD16	131	IOPD	
MD15	79	IOPD	
MD14	81	IOPD	
MD13	85	IOPD	
MD12	87	IOPD	
MD11	89	IOPD	
MD10	95	IOPD	
MD9	94	IOPD	
MD8	97	IOPD	
MD7	96	IOPD	
MD6	91	IOPD	
MD5	90	IOPD	
MD4	88	IOPD	
MD3	86	IOPD	
MD2	84	IOPD	
MD1	80	IOPD	
MD0	78	IOPD	
DQM0	98	O	Memory data qualify signal 0
DQM1	130	O	Memory data qualify signal 1
MCLK	104	O	SDRAM clock
WE#	99	O	Write enable control for SDRAM
RAS#	106	O	Row address strobe
CAS#	100	O	Column address strobe
FBCLK	110	IOPD	• Feed back clock for SDRAM • Chip Selection 2 for 6MByte external memory
MBA	107	IOPD	SDRAM bank select Shared with GPIO bit 3
MCS1#	109	IOPD	Memory chip Selection 1, And shared with GPIO bit 4.

MCS0#	108	IOPD	Memory chip Selection 0
MA10	116	IOPD	Memory address bus [10:0]
MA9	111	IOPD	
MA8	114	IOPD	
MA7	115	IOPD	
MA6	117	IOPD	
MA5	119	IOPD	
MA4	127	IOPD	
MA3	129	IOPD	
MA2	128	IOPD	
MA1	120	IOPD	
MA0	118	IOPD	

2.2.8 Digital Power and Ground Pins

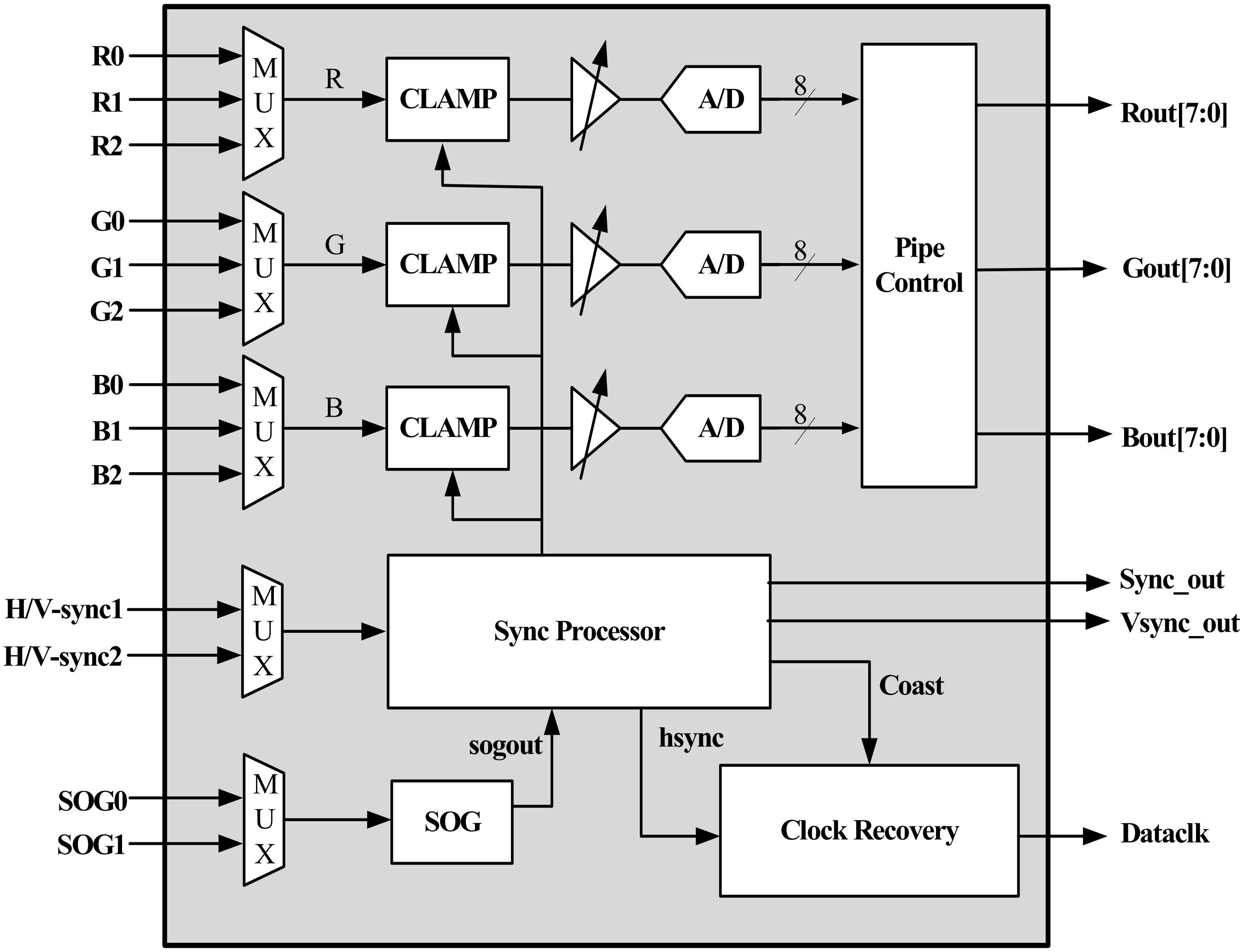
Pin Name	Pin #	Type	Pin Description
VSS	17,101,143	DG	Core Power GND
VDD	16,102,142	DP	1.8V Core Power
PVDD	3, 12, 24, 34, 82, 92, 103, 112, 123, 135, 147,	DP	3.3V I/O Power
PVSS	5, 13, 25, 35, 83, 93, 105, 113, 126, 136, 148	DG	I/O GND

2.3 Pin List by Pin Order

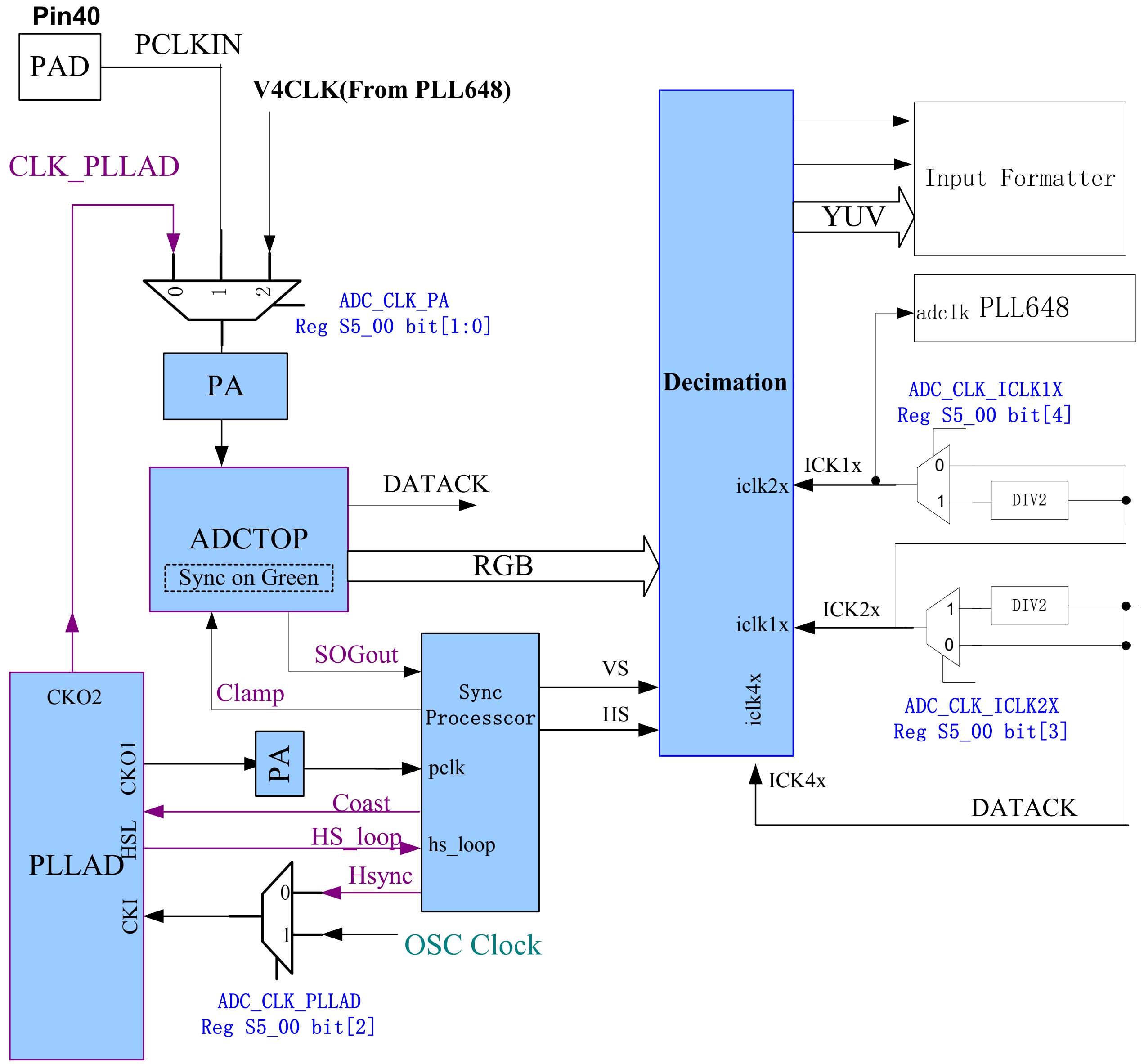
Pin #	Name	Pin #	Name	Pin #	Name	Pin #	Name
1	HSOUT	41	SCLCK	81	MD14	121	PAVD
2	VSOUT	42	SCLDA	82	PVDD	122	PAVS
3	PVDD	43	SCLSA	83	PVSS	123	PVDD
4	CLKOUT	44	HSIN1	84	MD2	124	XTIN
5	PVSS	45	VSIN1	85	MD13	125	XTOUT
6	HBOUT	46	HSIN2	86	MD3	126	PVSS
7	VBOUT	47	VSIN2	87	MD12	127	MA4
8	VB0	48	AVD_PLL	88	MD4	128	MA2
9	VB1	49	FILT	89	MD11	129	MA3
10	VB2	50	AVS_PLL	90	MD5	130	DQM1
11	VB3	51	AVD_REF	91	MD6	131	MD16
12	PVDD	52	REFBP	92	PVDD	132	MD31
13	PVSS	53	AVS_REF	93	PVSS	133	MD17
14	VB4	54	AVD_R	94	MD9	134	MD30
15	VB5	55	R0	95	MD10	135	PVDD
16	VDD	56	AVS_R	96	MD7	136	PVSS
17	VSS	57	R1	97	MD8	137	MD18
18	VB6	58	AVS_R	98	DQM0	138	MD29
19	VB7	59	R2	99	WE#	139	MD19
20	VG0	60	AVD_G	100	CAS#	140	MD28
21	VG1	61	SOG0	101	VSS	141	MD20
22	VG2	62	G0	102	VDD	142	VDD
23	VG3	63	AVS_G	103	PVDD	143	VSS
24	PVDD	64	SOG1	104	MCLK	144	MD27
25	PVSS	65	G1	105	PVSS	145	MD21
26	VG4	66	AVS_G	106	RAS#	146	MD26
27	VG5	67	G2	107	MBA	147	PVDD
28	VG6	68	AVD_B	108	MCS0#	148	PVSS
29	VG7	69	B0	109	MCS1#	149	MD22
30	VR0	70	AVS_B	110	FBCLK	150	MD25
31	VR1	71	B1	111	MA9	151	MD23
32	VR2	72	AVS_B	112	PVDD	152	MD24
33	VR3	73	B2	113	PVSS	153	DVSS
34	PVDD	74	AVS_B	114	MA8	154	ASVM
35	PVSS	75	RSTN	115	MA7	155	IREF
36	VR4	76	GPIO	116	MA10	156	DAVS
37	VR5	77	HALF	117	MA6	157	DAVD
38	VR6	78	MD0	118	MA0	158	AGPb
39	VR7	79	MD15	119	MA5	159	AGY
40	PCLKIN	80	MD1	120	MA1	160	AGPr

3 ADC

3.1 Function Block diagram



3.2 5725 analog part clock diagram



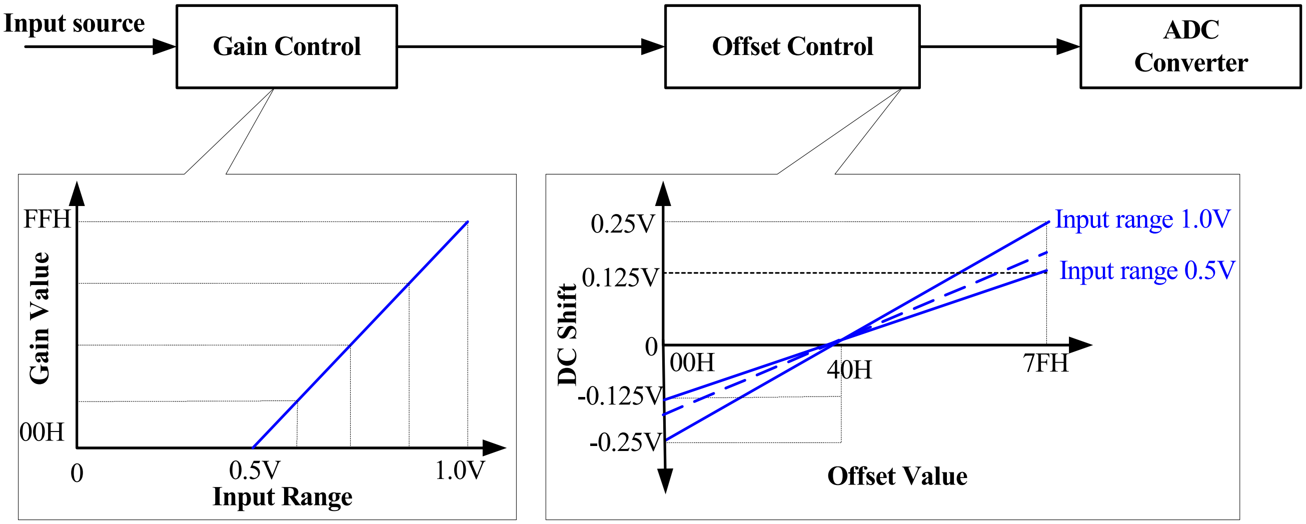
3.3 ADC Top

3.3.1 ADC parameters

- The analog input range: 0.5v –1.0v p-p
- 3:1 analog input mux: RGB0/YUV0, RGB1/YUV1, and RGB2
- The maximal sample rate: 162MSPS
- Resolution: 8 bits
- Power supply: 3.3v +/- 10% for three ADCs and REF.
- Clamp ground or midpoint
- Gain control resolution: 8 bits;
- Offset control resolution: 7 bits; offset range: -64LSB to 64LSB;

- Sync detect resolution: 5bits; comparator voltage range 10mv~320mv.
- DNL +/- 0.5 and INL +/- 2
- Power-down mode
- Programmable analog bandwidth

3.3.2 ADC gain and offset control



The ADC can accommodate input signals with inputs ranging from 0.5V to 1.0V full scale, the full-scale range is set in three 8-bit registers (Red Gain, Green Gain, and Blue Gain), the interaction of input range and gain control is illustrated in the above figure!

The ADC offset control shift the input source a DC level, there are three 7-bit registers (Red Offset, Green Offset, Blue Offset) provide independent control for each channel, The offset controls provide a ± 63 LSB adjustment range, the interaction of input range and offset control is illustrate the above figure!

If the input range is 1.0V, the adjustable range is ±0.25V; every step is 4mV (0.25 / 63 = 0.004V);

If the input range is 0.5V, the adjustable range is ±0.125V; every step is 2mV (0.125 / 63 = 0.002V);

Gain (Dec) = [Vpp / (2x1.2153)]-168 (the Vpp is the peak to peak voltage of input signal)

We recommend that you give the offset contrl 64(dec), also you can vary the offset voltage by this formula following.

Offset voltage=(Vpp/2)+[Vpp / (2x128)] * (offsetcontrl-64)

Reference Register Map:

Register Name	Register Address	Register Description
ROFCNTRL [6:0]	Reg_S5_06 [6:0]	Offset control for R channel ADC
GOFCNTRL [6:0]	Reg_S5_07 [6:0]	Offset control for G channel ADC
BOFCNTRL [6:0]	Reg_S5_08 [6:0]	Offset control for B channel ADC
RGCNTRL [7:0]	Reg_S5_09 [7:0]	Gain control for R channel ADC
GGCNTRL [7:0]	Reg_S5_0a [7:0]	Gain control for G channel ADC
BGCNTRL [7:0]	Reg_S5_0b [7:0]	Gain control for B channel ADC

3.3.3 ADC Channel Select

ADC has 3 input data channel, 2 SOG input channel and 2 separated H/V-sync input channel. We assume data input channel is 0, 1, 2, SOG input is 0, 1, Separated H/V-sync is 0, 1

- 2 SOG input only can be used for data channel 0, 1
- 2 separated H/V-sync an be used for data channel 0, 1, 2
- Data channel select is shared with SOG channel select.
- Data channel select is independent with H/V-sync channel select.

Reference Register Map:

Register Name	Register Address	Register Description
ADC_INPUT_SEL [1:0]	Reg_S5_02 [7:6]	ADC input source selection • 00: RGB0 / YUV0 / SOG0 • 01: RGB1 / YUV1 / SOG1 • 10: RGB2 / YUV2 • 11: Reserved
Sp_ext_sync_sel	Reg_S5_20 [3]	External HS/VS select 0: hsin1 / vsin1 1: hsin2/vsin2

3.4 Sync Processor

Sync processor is mainly for:

- Sync status detect for H/V-sync or composite sync polarity and active.
- SOG separation support Standard (PAL NTSC) and HD (ITU-R BT.709-5) Macro vision SOG source sync separate
- COAST generate for clock recovery
- H-sync & V-sync retiming with Dataclk to phase aligned
- CLAMP generate for ADC convert!

3.4.1 Clamping

The key to clamping is to identify a portion (time) of the signal when the graphic system is known to be producing black. An offset if then introduced which results in the A/D converts producing a black output (code 00h) when the known black input is present. The offset then remains in place when other signal levels are processed, and the entire signal is shifted to eliminate offset errors.

Sync processor internal clamping generator can program to set the number of pixel times that should pass after the trailing edge of H-sync before clamping starts. The **sp_cs_clp_st** and **sp_cs_clp_sp** can adjust the clamping position and duration.

3.4.1.1 RGB Clamping

Register Name	Value	Register Address	Register Description
ADC_SOGEN	0	Reg_S5_02 [0]	SOG enable signal, 0: disable, 1:enable
ADC_RYSEL_R	0	Reg_S5_03 [1]	Clamp ground or mid-scale for R ADC 0: Clamp to GND, 1: Clamp to MID
ADC_RYSEL_G	0	Reg_S5_03 [2]	Clamp ground or mid-scale for G ADC 0: Clamp to GND, 1: Clamp to MID
ADC_RYSEL_B	0	Reg_S5_03 [3]	Clamp ground or mid-scale for B ADC 0: Clamp to GND, 1: Clamp to MID

Sp_cs_clp_st	Option	Reg_S5_42 [3:0] Reg_S5_41 [7:0]	Clamping start position
Sp_cs_clp_sp	Option	Reg_S5_44 [3:0] Reg_S5_43 [7:0]	Clamping stop position
Sp_clamp_manual	Option	Reg_S5_56 [2]	Clamp manual control: 0, auto clamp enable
Sp_ht_diff_reg	Option	Reg_S5_59 [3:0] Reg_S5_58 [7:0]	Auto clamping enable of H-total difference threshold. E.g. 5
Sp_vt_diff_reg	Option	Reg_S5_5b [2:0] Reg_S5_5a [7:0]	Auto clamping enable of V-total difference threshold. E.g. 5
Sp_stable_cnt_reg	Option	Reg_S5_5c [7:0]	Auto clamping enable of frame threshold. E.g. 6

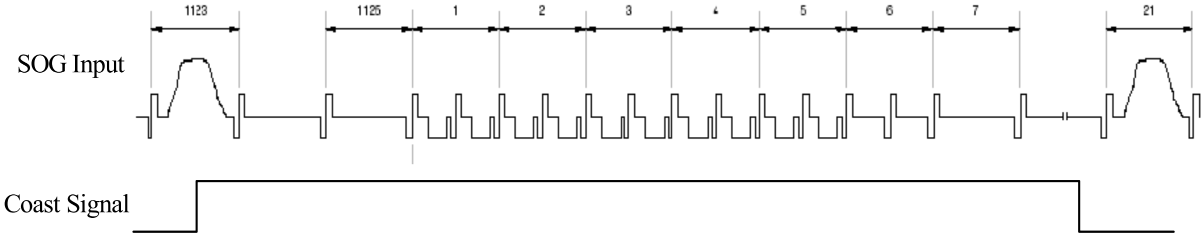
3.4.1.2 SOG Clamping

Register Name	Value	Register Address	Register Description
ADC_SOGEN	1	Reg_S5_02 [0]	SOG enable signal, 0: disable, 1:enable
ADC_RYSEL_R	1	Reg_S5_03 [1]	Clamp ground or mid-scale for R ADC 0: Clamp to GND, 1: Clamp to MID
ADC_RYSEL_G	0	Reg_S5_03 [2]	Clamp ground or mid-scale for G ADC 0: Clamp to GND, 1: Clamp to MID
ADC_RYSEL_B	1	Reg_S5_03 [3]	Clamp ground or mid-scale for B ADC 0: Clamp to GND, 1: Clamp to MID
Sp_cs_clp_st	Option	Reg_S5_42 [3:0] Reg_S5_41 [7:0]	Clamping start position
Sp_cs_clp_sp	Option	Reg_S5_44 [3:0] Reg_S5_43 [7:0]	Clamping stop position
Sp_clamp_manual	0	Reg_S5_56 [2]	Clamp manual control: 0, auto clamp enable
Sp_ht_diff_reg	Option	Reg_S5_59 [3:0] Reg_S5_58 [7:0]	Auto clamping enable of H-total difference threshold. E.g. 5
Sp_vt_diff_reg	Option	Reg_S5_5b [2:0] Reg_S5_5a [7:0]	Auto clamping enable of V-total difference threshold. E.g. 5
Sp_stable_cnt_reg	Option	Reg_S5_5c [7:0]	Auto clamping enable of frame threshold. E.g. 6

3.4.2 Coast Generation

In some systems, H-sync is disturbed during the Vertical Sync period (V-sync). In some cases, H-sync pulses disappear. In other systems, such as those that employ Composite Sync (Csync) signals or embedded Sync-On-Green (SOG), H-sync includes equalization pulses or other distortions during V-sync. To avoid upsetting the clock generator during V-sync, it is important to ignore these distortions.

Sync processor generate coast signal to eliminate this problem. It is an asynchronous input that disables the PLLAD input and allows the clock to free-run at its then-current frequency. The PLLAD can free-run for several lines without significant frequency drift.



Reference Register Map:

Register Name	Register Address	Register Description
Sp_pre_coast	Reg_S5_38 [7:0]	Set the coast will valid before vertical sync line number
Sp_post_coast	Reg_S5_39 [7:0]	When line counter reach this value coast will go down.
Sp_h_cst_st	Reg_S5_4e [3:0] Reg_S5_4d [7:0]	H-coast start position, H.total – this value used for macro-vision
Sp_h_cst_sp	Reg_S5_50 [3:0] Reg_S5_4f [7:0]	H-coast stop position, H.total – this value used for macro-vision
Sp_hcst_auto_en	Reg_S5_55 [7]	If enable, h-coast will start as (V-total – hcst_st)

3.5 Clock Recovery

5725 use a PLLAD to implement clock recovery.

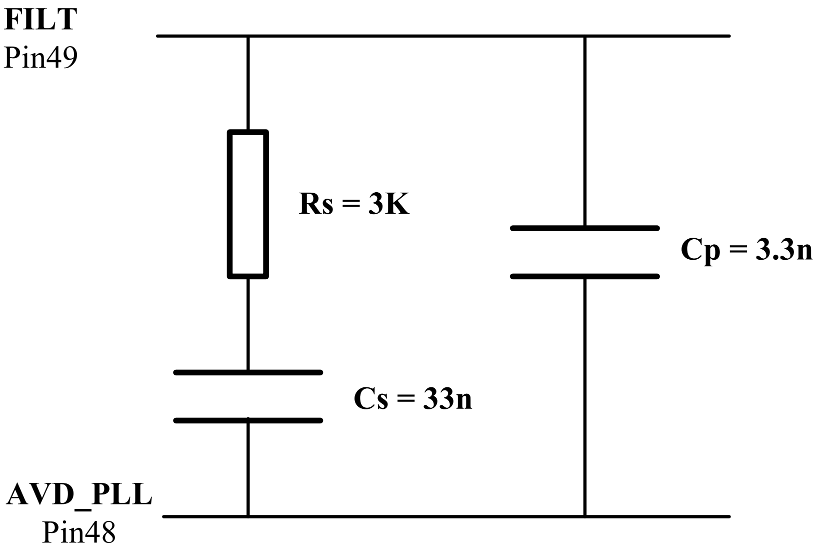
PLLAD generates pixel clock and over-sampling clock for ADC; its source is from H-sync And Coast. And a recovered H-sync is generated for sync processor, the frequency range of recovery pixel clock is from lower than 5MHz to 175Mhz. When the COAST signal is presented, it maintains its output frequency in the absence of H-sync.

3.5.1 PLLAD parameters

H-sync input frequency:	15KHz to 110KHz.
Output clock frequency:	5MHz to 175MHz.
PLL RMS jitter:	typ (400ps) max (700ps).
Analog Power:	3.3v +- 10%.
Lock in time:	~100ms
Clock output duty cycle:	45% to 55%.
Temperature range:	-40’c ~ 85’c.
Process:	0.18UM Process.

3.5.2 Loop Filter

The PLLAD characteristic determined by the loop filter design, by the PLLAD Charge Pump Current, And by the VCO ranges setting. The loop filter design is illustrated in the following figure.



3.5.3 Register Setting Rules

FCKI:	**Input clock frequency (h-sync), the frequency range is 10KHz~110KHz;**
FCKO1:	**Output clock frequency (pixel rate), the frequency range is 5Mhz~175Mhz;**
OSR:	**The over-sampling ratio (Fcko2/Fcko1), it is limited to 1, 2 and 4.**
CS:	**The PLLAD's loop filter capacitance, default value is 33nF.**

3.5.3.1 ND [11:0]

1. ND [11:0]: fixed to 0, it is used for test.

3.5.3.2 MD [11:0]

2. MD [11:0]: Round (Round (1000 * fcko1 / fcki, 1), 0) - 1

Note:

1. # Round (num, n) is function to rounds a number, in which num is the numeric expression to be rounded, n specifies how many places to the right of the decimal are included in the rounding
2. # MD[11:0] must be smaller than 4096.

3.5.3.3 KS [1:0]:

If	**FCKO1<=22MHz**	then KS [1:0]=2'b11;
If	**22MHz<FCKO1<=44MHz**	then KS [1:0]=2'b10;
If	**44MHz<FCKO1<=88MHz**	then KS [1:0]=2'b01;
If	**FCKO1>88MHz**	then KS [1:0]=2'b00;

3.5.3.4 CKOS [1:0]:

If OSR=1 then CKOS [1:0]=KS [1:0];
If OSR=2 and KS [1:0]= 2'b11, then CKOS [1:0]= 2'b10;
KS [1:0]= 2'b10, then CKOS [1:0]= 2'b01;
KS [1:0]= 2'b01, then CKOS [1:0]= 2'b00;
KS [1:0]= 2'b00, then PLLAD cannot generate 2X over-sampling clock.

If OSR=4 and KS [1:0]= 2’b11, then CKOS [1:0]= 2’b01;
KS [1:0]= 2’b10, then CKOS [1:0]= 2’b00;
KS [1:0]= 2’b01, then PLLAD cannot generate 4X over-sampling clock.
KS [1:0]= 2’b00, then PLLAD cannot generate 4X over-sampling clock.

3.5.3.5 FS

If FCKO1*PD>120 then FS=1;
If FCKO1*PD<=120 then FS=0;

Note:

1 PD=1 when KS [1:0]=2’b00;

2 PD=2 when KS [1:0]=2’b01;

3 PD=4 when KS [1:0]=2’b10;

4 PD=8 when KS [1:0]=2’b11;

3.5.3.6 ICP [2:0]

Icpval=(FCKI*2*PI/SR)^2 * CS * M * PD / KVCO

If	**Icpval <= 60u**	then ICP [2:0]=3’b000;
If	**60<Icpval <= 110u**	then ICP [2:0]=3’b001;
If	**110<Icpval <= 175u**	then ICP [2:0]=3’b010;
If	**175<Icpval <= 300u**	then ICP [2:0]=3’b011;
If	**300<Icpval <= 425u**	then ICP [2:0]=3’b100;
If	**425<Icpval <= 650u**	then ICP [2:0]=3’b101;
If	**650<Icpval <= 1200u**	then ICP [2:0]=3’b110;
If	**Icpval > 1200u**	then ICP [2:0]=3’b111;

Note:

1 u is 10^-6

2 PI=3.1415927

3 When FCKO1<=32, SR=10; when FCKO1>32, SR=15. In which the SR means PLL Stability Ratio.

4 M=MD [11:0]+1 is PLL divide ratio

5 When FS=1, KVCO=90000000; when FS=0 KVCO=66000000.

6 Icpval is the PLL's Charge Pump current.

Reference Register Map:

Register Name	Register Address	Register Description
PLLAD_VRORST	Reg_S5_11 [0]	Initial VCO control voltage
PLLAD_LEN	Reg_S5_11 [1]	PLL lock detector circuit enable signal
PLLAD_PDZ	Reg_S5_11 [4]	=0, PLLAD in power down mode =1, PLLAD in work mode
PLLAD_BPS	Reg_S5_11 [65]	=1, bypass input clock to CKO1 and CKO2
PLLAD_LAT	Reg_S5_11 [7]	Trigger bit for ND, MD, KS, CKOS, ICP When you change the ND, MD, KS, CKOS, ICP value, you must enable PLLAD_LAT generate a rising pulse to latch the update value

PLLAD_R [1:0]	Reg_S5_12 [1:0]	PLL lock detector circuit input cock delay control
PLLAD_S [1:0]	Reg_S5_13 [3:2]	PLL lock detector circuit input cock delay control
PLLAD_FS	Reg_S5_11 [5]	VCO gain selection, 0, default, 1, high gain
PLLAD_KS [1:0]	Reg_S5_16 [5:4]	Post divider selection control
PLLAD_CKOS [1:0]	Reg_S5_16 [7:6]	CKO2 divider selection control
PLLAD_MD [11:0]	Reg_S5_13 [3:0] Reg_S5_12 [7:0]	Feedback programmable divider control
PLLAD_ICP [2:0]	Reg_S5_17 [2:0]	Programmable charge pump current control
PLLAD_ND [11:0]	Reg_S5_15 [3:0] Reg_S5_14 [7:0]	Reference clock divider control

3.6 Decimation Filters

5725 ADC can over-sample by a factor 2x or 4x, the A/D outputs first pass through decimation filters That reduce the data rate to 1x the pixel rate. The decimation filter is a half-band filter. Over-sampling and decimation filtering can effectively increase the overall signal-to-noise ratio.

- If ADC do 4x or 2x over-sample, we must do decimation filter, if the source is RGB domain, we must enable decimation color space conversion for RGB to YUV, and set the clock 2x, 4x divider.

3.6.1 Decimation setting for ADC 4X over-sampling

Register Name	Value	Register Address	Register Description
Adc_clk_iclk2x	1	Reg_S5_00 [3]	Enable ADC_ICLK2x = ADC datack / 2
Adc_clk_iclk1x	1	Reg_S5_00 [4]	Enable ADC_ICLK1x = ADC_ICLK2x / 2
Dec1_byps	0	Reg_S5_1f [0]	Enable decimator 1 work
Dec2_byps	0	Reg_S5_1f [1]	Enable decimator 2 work
Dec_matrix_byps	Option	Reg_S5_1f [2]	=1, YUV input, bypass RGB2YUV conversion =0, RGB input, do RGB2YUV conversion

3.6.2 Decimation setting for ADC 2X over-sampling

Register Name	Value	Register Address	Register Description
Adc_clk_iclk2x	0	Reg_S5_00 [3]	Enable ADC_ICLK2x = ADC datack
Adc_clk_iclk1x	1	Reg_S5_00 [4]	Enable ADC_ICLK1x = ADC_ICLK2x / 2
Dec1_byps	1	Reg_S5_1f [0]	Bypass decimator 1
Dec2_byps	0	Reg_S5_1f [1]	Enable decimator 2 work
Dec_matrix_byps	Option	Reg_S5_1f [2]	=1, YUV input, bypass RGB2YUV conversion =0, RGB input, do RGB2YUV conversion

3.6.3 Decimation setting for No ADC over-sampling

Register Name	Value	Register Address	Register Description
Adc_clk_iclk2x	0	Reg_S5_00 [3]	Enable ADC_ICLK2x = ADC datack
Adc_clk_iclk1x	0	Reg_S5_00 [4]	Enable ADC_ICLK1x = ADC_ICLK2x
Dec1_byps	1	Reg_S5_1f [0]	Bypass decimator 1
Dec2_byps	1	Reg_S5_1f [1]	Enable decimator 2 work
Dec_matrix_byps	Option	Reg_S5_1f [2]	=1, YUV input, bypass RGB2YUV conversion =0, RGB input, do RGB2YUV conversion or you can do RGB2YUV conversion with module input formatter.

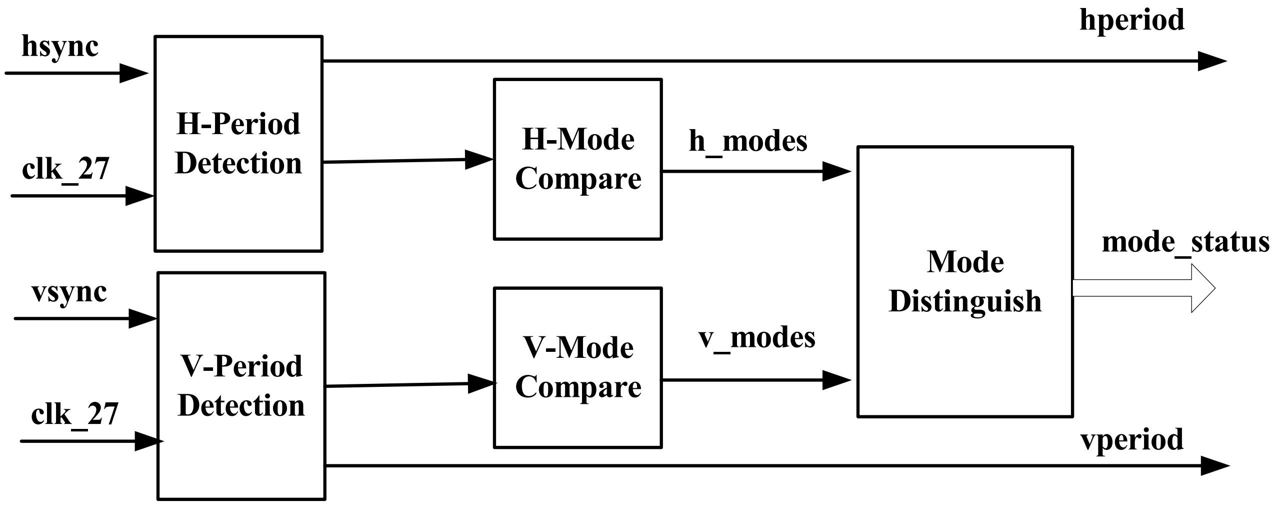
3.7 Phase Adjustment

There is 2 PA in 5725, one is for adjust the clock to ADC, the other is for adjust the clock to sync Processor, PA is 5-bit phase adjustment; its accuracy is 1/32 clock period, its wok range is 10MHz to 200MHz.

Register Name	Register Address	Register Description
PA_sp_bypsz	Reg_S5_19 [0]	=0, PA for sync processor in bypass mode =1, PA for sync processor in working mode
PA_sp_S [4:0]	Reg_S5_19 [5:1]	Sync processor clock phase adjustment, larger value mean more delay, (1 LSB = T / 32)
PA_sp_lockoff	Reg_S5_19 [6]	Lock circuit disable control of sync processor PA
PA_sp_lat	Reg_S5_19 [7]	Trigger bit for PA_sp_S, PA_sp_S will be latched when PA_sp_lat rising edge!
PA_adc_bypsz	Reg_S5_18 [0]	=0, PA for ADC in bypass mode =1, PA for ADC in working mode
PA_adc_S [4:0]	Reg_S5_18 [5:1]	ADC clock phase adjustment, larger value mean more delay, (1 LSB = T / 32)
PA_adc_lockoff	Reg_S5_18 [6]	Lock circuit disable control for ADC PA
PA_adc_lat	Reg_S5_18 [7]	Trigger bit for PA_adc_S, PA_adc_S will be latched when PA_adc_lat rising edge!

4 Auto Mode Detect

True View 5725 auto mode detect can detect all monitor and TV supported modes. It includes NTSC, PAL, VGA, SVGA, XGA, SXGA and HD modes and their supported refresh rate. This module use crystal clock, such as 27MHz, to sample all input horizontal sync and vertical sync, compare the internal detected value and the register value to distinguish each modes and their refresh rate.



4.1 Video mode detection

Video mode detection will calculate the input video source H/V period use fixed clock (crystal clock), then compare the H/V period with the reference value (register programmable); it will distinguish the different video mode. Its hsync and vsync can from 3 sources:

- Discrete H/V-sync from external PAD.
- 656 H/V-blank from internal input formatter.
- Separated H/V-sync from internal sync processor.

4.1.1 H/V period calculation

Commonly, we use 27MHz as reference clock, then generate a write enable of crystal clock divided by 4, that is 6.75MHz, to sample H/V sync and generate H/V period. For 601 source, its H/V sync is directly from external input, for 656 source, its H/V sync is encoded with input formatter module.

Note: For 656 input, you must program Reg_S1_00 [3] (if_sel_656) to 1.

TrueView 5725 use half line timing to decide the input source is interlace or progressive for 576p and 1080i 50Hz mode. The following table is the value of different modes H/V-detection value:

4.1.2 V-Modes distinguish

Firstly, it compares the V-period value with our programmed value to decide some modes:

Mode	Total Resolution (H x V)	Frame Rate (Hz)	V-period (DEC)	V-period/ 16 (DEC)	V-period / 16 (HEX)	Detect Result
NTSC	858x525i	30	525	32.8125	20	ntsc_int
	858x525p	60	1050	65.625	41	ntsc_prg
PAL	864x625i	25	625	39.0625	27	pal_int
	864x625p	50	1250	78.125	4E	v1250_mode
VGA	800x525	60	1050	65.625	41	ntsc_prg
	840x500	75	1000	62.5	3E	vga_mode
	832x509	85	1018	63.625	3F	
SVGA	1056x628	60	1256	78.5	4E	v1250_mode
	1056x625	75	1250	78.125	4E	
	1048x631	85	1262	78.875	4E	
XGA	1344x806	60	1612	100.75	64	xga_vmod
	1328x806	70	1612	100.75	64	
	1312x800	75	1600	100	64	
	1376x808	85	1616	101	65	
SXGA	1688x1066	60	2132	133.25	85	sxga_vmod
	1688x1066	75	2132	133.25	85	
	1728x1072	85	2144	134	86	
720P	1980x750	50	1500	93.75	5D	720p_vmod
	1650x750	60	1500	93.75	5D	
1080I	2200x1125	30	1125	70.3125	46	hd1125i_mode
	2376x1250	25	1250	78.125	4E	v1250_mode
	2640x1125	25	1125	70.3125	46	hd1125i_mode
1080P	2200x1125	60	2250	140.625	8C	hd1125p_mode
	2376x1250	50	2500	156.25	9C	hd_2376_1250p
	2640x1125	50	2250	140.625	8C	hd1125p_mode

Note: The highlight area is the value we should program!

Below table is the related register definition:

Register Name	Register Address	Register Description
MD_ntsc_int_cntrl	Reg_S1_62 [5:0]	NTSC interlace mode detect v-period
MD_pal_int_cntrl	Reg_S1_63 [5:0]	PAL interlace mode detect v-period
MD_ntsc_prg_cntrl	Reg_S1_64 [6:0]	NTSC progressive mode detect v-period
MD_VGA_cntrl	Reg_S1_65 [6:0]	VGA mode detect v-period except VGA 60Hz
MD_SEL_VGA60	Reg_S1_65 [7]	Set to 1, VGA 60Hz mode Set to 0, NTSC progressive mode
MD_V1250_vcntrl	Reg_S1_68 [6:0]	Vertical 1250 line mode detect v-period
MD_XGA_cntrl	Reg_S1_6d [6:0]	XGA mode detect v-period
MD_SXGA_cntrl	Reg_S1_72 [7:0]	SXGA mode detect v-period
MD_HD720p_cntrl	Reg_S1_76 [6:0]	HD720p mode detect v-period
MD_HD_1125i_cntrl	Reg_S1_79 [6:0]	1080i mode 1125 lines detect v-period
MD_HD_1125p_cntrl	Reg_S1_7c [7:0]	1080p mode 1125 lines detect v-period
MD_HD_1250p_cntrl	Reg_S1_7f [7:0]	1080p mode 2376x1250 mode v-period

4.1.3 H-modes distinguish

With V-modes compare, there are still some modes can't distinguish, so we need H-modes compare to decide the modes accurately.

Mode	Total Resolution (H x V)	Line Rate (KHz)	H-period （DEC) 6.75M/Line Rate	Hperiod （HEX)	Detect Result	Status
NTSC	858x525i	15.75	429	1AD	ntsc_int	sd_mode
	858x525p	31.5	214	D6	ntsc_prg	
PAL	864x625i	15.625	432	1B0	pal_int	
	864x625p	31.25	216	D8	pal_prg	
VGA	800x525	31.5	214	D6	vga_60	vga_mode
	840x500	37.5	179	B3	vga_75	
	832x509	43.3	156	9C	vga_85	
SVGA	1056x628	37.9	178	B2	svga_60	svga_mode
	1056x625	46.9	144	90	svga_75	
	1048x631	53.7	125	7D	svga_85	
XGA	1344x806	48.4	140	8C	xga_60	xga_mode
	1328x806	56.5	120	78	xga_70	
	1312x800	60	113	71	xga_75	
	1376x808	68.7	99	63	xga_85	
SXGA	1688x1066	64	106	6A	sxga_60	xsxga_mode
	1688x1066	80	84	54	sxga_75	
	1728x1072	91.1	74	4A	sxga_85	
720P	1980x750	37.5	180	B4	720p_50	hd720p_mode
	1650x750	45	150	96	720p_60	
1080I	2200x1125	33.75	200	C8	1125i_60	hd1080i_mode
	2376x1250	31.25	216	D8	hd2376_1250i	
	2640x1125	28.125	241	F1	1125i_50	
1080P	2200x1125	67.5	100	64	hd1125p_60	hd1080p_mode
	2376x1250	62.5	108	6C	hd_2376_1250p	
	2640x1125	56.25	120	78	hd1125p_50	

Note: The highlight area is the value we should program!

Below table is the related register definition:

Register Name	Register Address	Register Description
MD_vga_75Hz_cntrl	Reg_S1_66 [7:0]	VGA 75Hz mode h-period
MD_vga_85Hz_cntrl	Reg_S1_67 [7:0]	VGA 85Hz mode h-period
MD_V1250_hcntrl	Reg_S1_69 [7:0]	V1250 mode h-period
MD_svga_60Hz_cntrl	Reg_S1_6a [7:0]	SVGA 60Hz mode h-period
MD_svga_75Hz_cntrl	Reg_S1_6b [7:0]	SVGA 75Hz mode h-period
MD_svga_85Hz_cntrl	Reg_S1_6c [7:0]	SVGA 85Hz mode h-period
MD_xga_60Hz_cntrl	Reg_S1_6e [7:0]	XGA 60Hz mode h-period
MD_xga_70Hz_cntrl	Reg_S1_6f [6:0]	XGA 70Hz mode h-period
MD_xga_75Hz_cntrl	Reg_S1_70 [6:0]	XGA 75Hz mode h-period
MD_xga_85_cntrl	Reg_S1_71 [6:0]	XGA 85Hz mode h-period
MD_sxga_60Hz_cntrl	Reg_S1_73 [6:0]	SXGA 60Hz mode h-period
MD_sxga_75Hz_cntrl	Reg_S1_74 [6:0]	SXGA 75Hz mode h-period
MD_sxga_85Hz_cntrl	Reg_S1_75 [6:0]	SXGA 85Hz mode h-period
MD_HD720p_60Hz_cntrl	Reg_S1_77 [7:0]	HD720p 60Hz mode h-period
MD_HD720p_50Hz_cntrl	Reg_S1_78 [7:0]	HD720p 50Hz mode h-period
MD_HD2200_1125i_cntrl	Reg_S1_7a [7:0]	1080i mode 2200x1125i h-period
MD_HD2640_1125i_cntrl	Reg_S1_7b [7:0]	1080i mode 2640x1125i h-period
MD_HD2200_1125p_cntrl	Reg_S1_7d [7:0]	1080p mode 2200x1125p h-period
MD_HD2640_1125p_cntrl	Reg_S1_7e [7:0]	1080p mode 2640x1125p h-period

4.1.4 Mode Status bits

After you complete the mode detection register programming, the auto mode detect can distinguish the different modes automatically and send the status bits to read only register, you can read out the status bits via reading register Reg_S0_00 ~ Reg_S0_05:

Register Bit	Modes Status	Register Bit	Modes Status
Reg_S0_00 [0]	Vertical timing stable	Reg_S0_00 [1]	Horizontal timing stable
Reg_S0_00 [2]	Both H & V timing stable	Reg_S0_00 [3]	NTSC interlace mode
Reg_S0_00 [4]	NTSC progressive mode	Reg_S0_00 [5]	PAL interlace mode
Reg_S0_00 [6]	PAL progressive mode	Reg_S0_00 [7]	SD video source mode
Reg_S0_01 [0]	VGA 60Hz mode	Reg_S0_00 [1]	VGA 75Hz mode
Reg_S0_01 [2]	VGA 85Hz mode	Reg_S0_01 [3]	VGA source mode
Reg_S0_01 [4]	SVGA 60Hz mode	Reg_S0_01 [5]	SVGA 75Hz mode
Reg_S0_01 [6]	SVGA 85Hz mode	Reg_S0_01 [7]	SVGA source mode
Reg_S0_02 [0]	XGA 60Hz mode	Reg_S0_02 [1]	XGA 70Hz mode
Reg_S0_02 [2]	XGA 75Hz mode	Reg_S0_02 [3]	XGA 85Hz mode
Reg_S0_02 [4]	XGA source mode	Reg_S0_02 [5]	SXGA 60Hz mode
Reg_S0_02 [6]	SXGA 75Hz mode	Reg_S0_02 [7]	SXGA 85Hz mode
Reg_S0_03 [0]	SXGA source mode	Reg_S0_03 [1]	Graphic source mode
Reg_S0_03 [2]	HD 720P 50Hz mode	Reg_S0_03 [3]	HD 720P 60Hz mode
Reg_S0_03 [4]	HD 720P source mode	Reg_S0_03 [5]	HD 2200x1125i mode (1080i_60)
Reg_S0_03 [6]	HD 2376x1250i mode (1250i)	Reg_S0_03 [7]	HD 2640x1125i mode (1080i_50)
Reg_S0_04 [0]	HD 1080i mode	Reg_S0_04 [1]	HD 2200x1125P mode
Reg_S0_04 [2]	HD 2376x1250p mode	Reg_S0_04 [3]	HD 2640x1125P mode
Reg_S0_04 [4]	HD 1080P source mode	Reg_S0_04 [5]	HD source mode
Reg_S0_04 [6]	Interlace source mode	Reg_S0_04 [7]	Progressive source mode

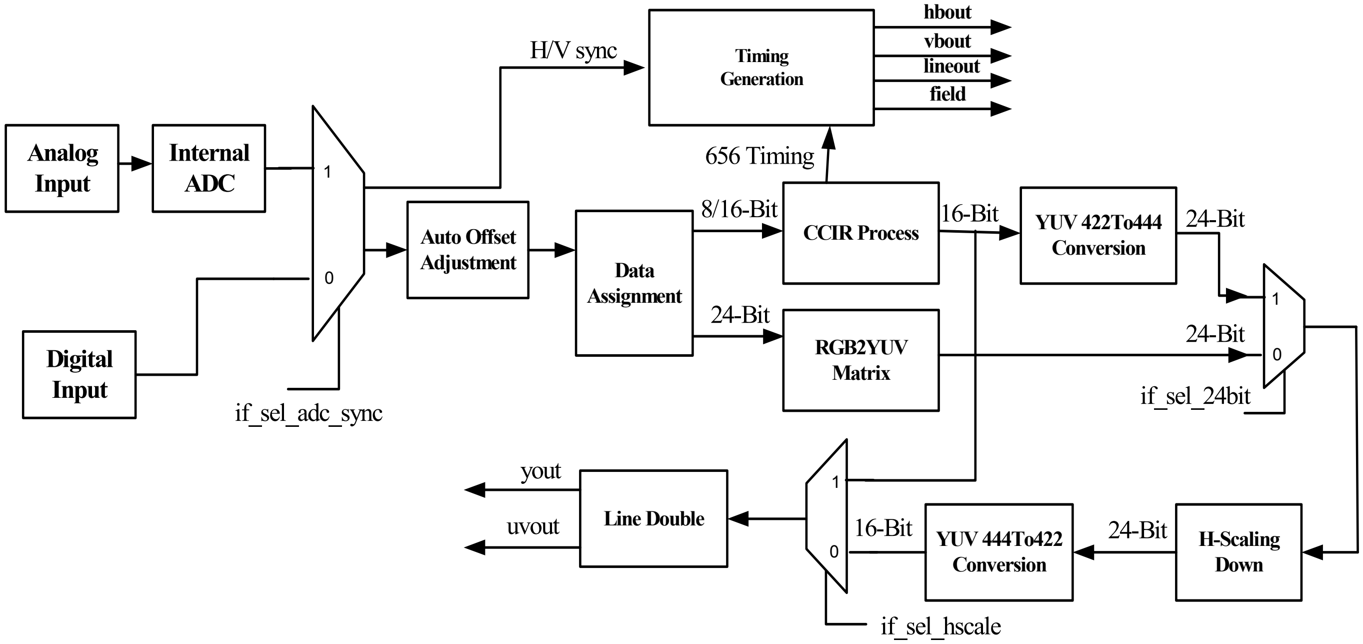
At the same time, you can read the H-period and V-period via reading register reg06 ~ reg08:

	Register Address	Register Meaning
If_hperiod [8:0]	Reg_S0_07 [0], Reg_S0_06 [7:0]	Input source H-total pixel / 4
If_vperiod [10:0]	Reg_S0_08 [3:0],Reg_S0_07 [7:1]	Input source V-total lines

5 Input Formatter

TrueView 5725 Input Formatter provides input data selection and conversion and generates internal timing for other blocks using.

Below are 5725 input formatter function block diagrams:



5.1 Auto Offset Adjustment

Auto offset adjustment is used when 24-bit YUV data input, auto adjust U/V data to color for color Correction.

When H/V blank period, we detect the U/V value, we assume the U/V data should be 8'h80, at the same time, we allow a range for it (if_auto_ofst_U_range, if_auto_V_range), when the difference value in the range, we think the correct value is zero, when the difference out of the range, we think the correct value is Offset = 8'h80 – din; and we will keep the correct value for a line or a field/frame (controlled by if_auto_ofst_prd).

```
If (data out of range)
        Data_out = Data_in + Offset;
Else
        Data_out = Data_in;
```

5.2 Data Assignment

TrueView 5725 input formatter data can select from internal ADC output or/and external digital video port.

- When from internal ADC, the video format is 24-bit RGB/YUV;
- When from external digital video port, the video format can be 24bit RGB, 8/16/24bit YUV input, CCIR656mode only support 8 bit data.

- When 5725 work on PIP mode, it can input video stream from digital port and analog port together.

TrueView 5725 internal data path is 16bit 4:2:2 YUV, so all types of input video will be converted to 16bit 4:2:2 YUV format.

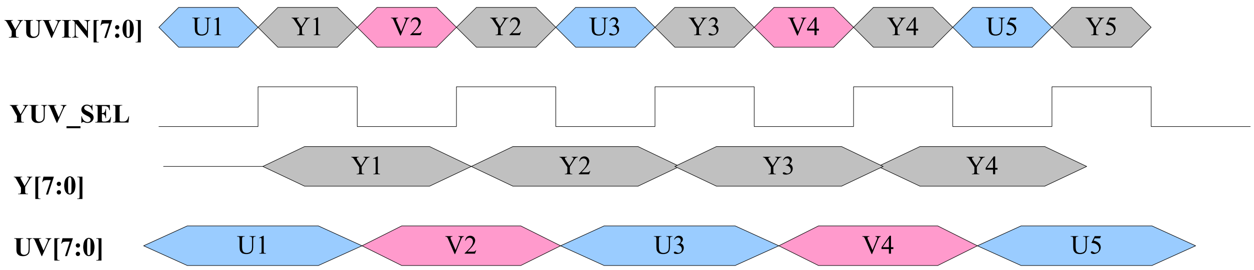
Data Type	Video Port Assignment		
24-Bit YUV	VG [7:0]	Pin29~26,Pin23~20	Y Data for 24-bit YUV 4:4:4 mode
	VB [7:0]	Pin19~18,Pin15~14, Pin11~8	U Data for 24-bit YUV 4:4:4 mode
	VR [7:0]	Pin39~36,Pin33~30	V data for 24-bit YUV 4:4:4 mode
24-Bit RGB	VG [7:0]	Pin29~26,Pin23~20	Green Data for 24-bit RGB mode
	VB [7:0]	Pin19~18,Pin15~14, Pin11~8	Blue Data for 24-bit RGB mode
	VR [7:0]	Pin39~36,Pin33~30	Red data for 24-bit RGB mode
16-Bit YUV	VB [7:0]	Pin19~18,Pin15~14, Pin11~8	U/V Data for 16-bit YUV 4:2:2 mode
	VR [7:0]	Pin39~36,Pin33~30	Y data for 16-bit YUV 4:2:2 mode
8-bit YUV	VG [7:0]	Pin29~26,Pin23~20	YUV Data for 8-bit 601/656 YUV 4:2:2 mode

5.3 CCIR Processing

CCIR Processing is encode the CCIR656 timing and convert the 8 bit data to 16 bit, at the same time, it Detect the input timing and generate the H/V-sync and field signal!

5.3.1 8bit mode:

The input stream is 8bit 601/656 UYVY format. Use the switch to separate the UYVY to UV and YY. Look at the diagram:



YUV_SEL is controlled by H-sync, so we distinguish Y and UV by H-sync basis, H-sync ODD clocks can be Y (or UV), then EVEN clocks can be UV (or Y).

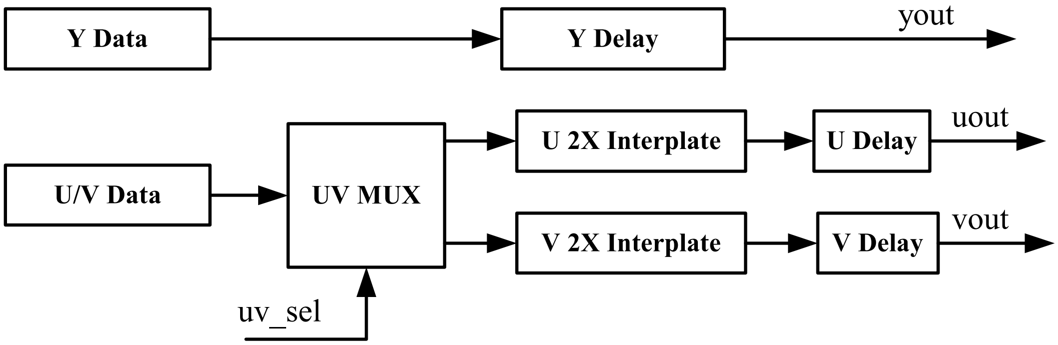
You can Flip Y and UV when you set if_uv_revert to 1.

5.3.2 16bit mode:

The data path is the standard 4:2:2 format. Bypass!

5.4 YUV 4:2:2 to 4:4:4 Conversion

YUV 4:2:2 to 4:4:4 conversion is used to convert 16bit YUV 4:2:2 data (output from CCIR processing) to 24bit YUV 4:4:4 data to do horizontal scaling down because horizontal scaling down is based 24bit YUV data format!



The following register is used to control YUV 4:2:2 to 4:4:4 conversion:

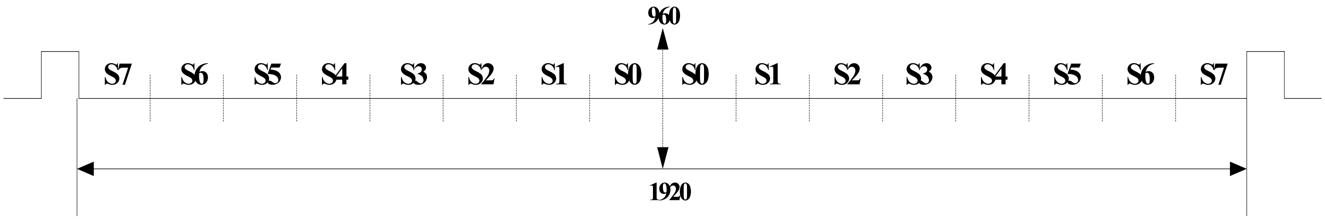
- lf_uv_flip: flip UV data for uv_sel
- lf_u_delay: U data pipe delay
- lf_v_delay: V data pipe delay
- lf_tap6_byps: UV 2x interpolation filter bypass control
- lf_y_delay: Y data pipe delay

5.5 RGB2YUV Color Space Conversion

When the data is 24Bit RGB input, we must do the color space conversion except bypass mode.

5.6 Horizontal Scaling Down

The non-linear scaling down is the core of horizontal scaling down. A line will be divided into 16 segments. Each segment has its own scaling factor. In fact, if we program all the segment scaling factors the same, the scaling down will be linear scaling down. The following figure shows the factor relation of each segment. (Assume the line have 1920 valid pixels):



1) When the scaling from M lines to N lines, then the value is (M-N/N) * 4095;

2) Horizontal phase adjustment bypass control Reg_S1_02 [3], when set to 1, horizontal phase Adjustment for Y, U and V will bypass.

3) The following is the horizontal scaling down limitation of interlaced input source because of line_double limitation.

- 1080i (2200x1125i 1920x1080i)
  Horizontal scaling down limitation is 820 (must larger than 820 and less than 1004).
- 1080m (2640x1125i 1920x1080i)
  Horizontal scaling down limitation is 600 (must larger than 600 and less than 1004).
- 1250i (2376x1250i 1920x1080i)
  Horizontal scaling down limitation is 732 (must larger than 732 and less than 1004).

Reference Register Map:

Register Name	Register Address	Register Description
If_hs_sel_lpf	Reg_S1_02 [1]	Low pass filter or interpolate select, HD:1, SD:0
If_hs_int_lpf_byps	Reg_S1_02 [2]	Bypass interpolate or lpf
If_hs_pshift_byps	Reg_S1_02 [3]	Bypass horizontal scaling phase adjustment
If_hs_rate_seg0	Reg_S1_03 [7:0]	Horizontal non-linear scaling down 1st segment scaling ratio.
If_hs_rate_seg1	Reg_S1_04 [7:0]	Horizontal non-linear scaling down 2nd segment scaling ratio.
If_hs_rate_seg2	Reg_S1_05 [7:0]	Horizontal non-linear scaling down 3rd segment scaling ratio.
If_hs_rate_seg3	Reg_S1_06 [7:0]	Horizontal non-linear scaling down 4th segment scaling ratio.
If_hs_rate_seg4	Reg_S1_07 [7:0]	Horizontal non-linear scaling down 5th segment scaling ratio.
If_hs_rate_seg5	Reg_S1_08 [7:0]	Horizontal non-linear scaling down 6th segment scaling ratio.
If_hs_rate_seg6	Reg_S1_09 [7:0]	Horizontal non-linear scaling down 7th segment scaling ratio.
If_hs_rate_seg7	Reg_S1_0a [7:0]	Horizontal non-linear scaling down 8th segment scaling ratio.
If_hs_rate_low	Reg_S1_0b [3:0]	Horizontal non-linear scaling down rate low 4 bit
If_hs_dec_factor	Reg_S1_0b [5:4]	Horizontal non-linear scaling down DEC factor: 00: 1x 01: 1/2x, 10:1/4x, 11: 1/8x
If_sel_hscale	Reg_S1_0b [6]	=0, not scaled 8/16 bit input data =1, scaled 8/16 bit input data or 24 bit input

5.7 Timing generation

Timing generation is used to generate timing for Horizontal scaling down and other modules:

- Line: used for line double and de-interlace
- hb0/vb0: used for de-interlace.
- Hb1/vb1: used for WFF/RFF
- HB1/vb2: used for capture
- Hbin: used for h-scaling down.

5.8 HD Bypass data path

In bypass mode, the input data will be converted to RGB 24-bit format, and then sent to DAC to display. So, this path includes dynamic range and YUV2RGB two major functions.

5.8.1 Dynamic range expansion

1) **Y3 = y_gain/128 * yin [7:0] + y_offset;**
Y gain adjustment: Reg_S1_31 [7:0] = (0~2x) * 128
Y offset adjustment: Reg_S1_32 [7:0], its range is –128 ~ 127

2) **U3 = u_gain/128 * uin [7:0] + u_offset;**
Firstly, we convert the U from signed to unsigned number.
U gain adjustment: Reg_S1_33 [7:0] = (0~2x) * 128
U offset adjustment: Reg_S1_34 [7:0], its range is –128 ~ 127

3) **V3 = v_gain/128 * vin [7:0] + v_offset;**
Firstly, we convert the V from signed to unsigned number.
V gain adjustment: Reg_S1_35 [7:0] = (0~2x) * 128
V offset adjustment: Reg_S1_36 [7:0], its range is –128 ~ 127

The dynamic range expansion will convert all kinds of YCbCr or YUV color space what the YUV2RGB module wanted. We can get the gain and offset value based on input source type and YUV2RGB module matrix. Dynamic range expansion bypass control is reg21 [6], when it is 1; YUV data bypass the dynamic range expansion process.

5.8.2 YUV2RGB color space conversion matrix

This module is an 8-bit YUV to RGB conversion matrix, it formulas are:

1) G = Y – 0.388U – 0.764V
2) B = Y + 2U
3) R = Y + 1.5V

5.8.3 Timing generation for bypass mode

For bypass mode, we must keep the input video stream’s line rate and frame rate, but we can adjust the horizontal and vertical sync position and blanking width and position:

Reference Registers Map:

Register Name	Register Address	Register Description
Hd_hsync_rst [10:0]	Reg_S1_38 [2:0] Reg_S1_37 [7:0]	Horizontal total value
Hd_ini_st [10:0]	Reg_S1_3a [2:0] Reg_S1_39 [7:0]	Horizontal reset pulse start position
Hd_hb_st [11:0]	Reg_S1_3c [3:0] Reg_S1_3b [7:0]	Horizontal blanking start position
Hd_hb_sp [11:0]	Reg_S1_3e [3:0] Reg_S1_3d [7:0]	Horizontal blanking stop position
Hd_hs_st [11:0]	Reg_S1_40 [3:0] Reg_S1_3f [7:0]	Horizontal sync start position
Hd_hs_sp [11:0]	Reg_S1_42 [3:0] Reg_S1_41 [7:0]	Horizontal sync stop position

Hd_vb_st [11:0]	Reg_S1_44 [3:0] Reg_S1_43 [7:0]	Vertical blanking start position
Hd_vb_sp [11:0]	Reg_S1_46 [3:0] Reg_S1_45 [7:0]	Vertical blanking stop position
Hd_vs_st [11:0]	Reg_S1_48 [3:0] Reg_S1_47 [7:0]	Vertical sync start position
Hd_vs_sp [11:0]	Reg_S1_4a [3:0] Reg_S1_49 [7:0]	Vertical sync stop position
Hd_ext_hb_st [11:0]	Reg_S1_4c [3:0] Reg_S1_4b [7:0]	External horizontal blanking start position
Hd_ext_hb_sp [11:0]	Reg_S1_4e [3:0] Reg_S1_4d [7:0]	External horizontal blanking stop position
Hd_ext_vb_st [11:0]	Reg_S1_50 [3:0] Reg_S1_4f [7:0]	External vertical blanking start position
Hd_ext_vb_sp [11:0]	Reg_S1_52 [3:0] Reg_S1_51 [7:0]	External vertical blanking stop position

5.8.4 Blank insertion

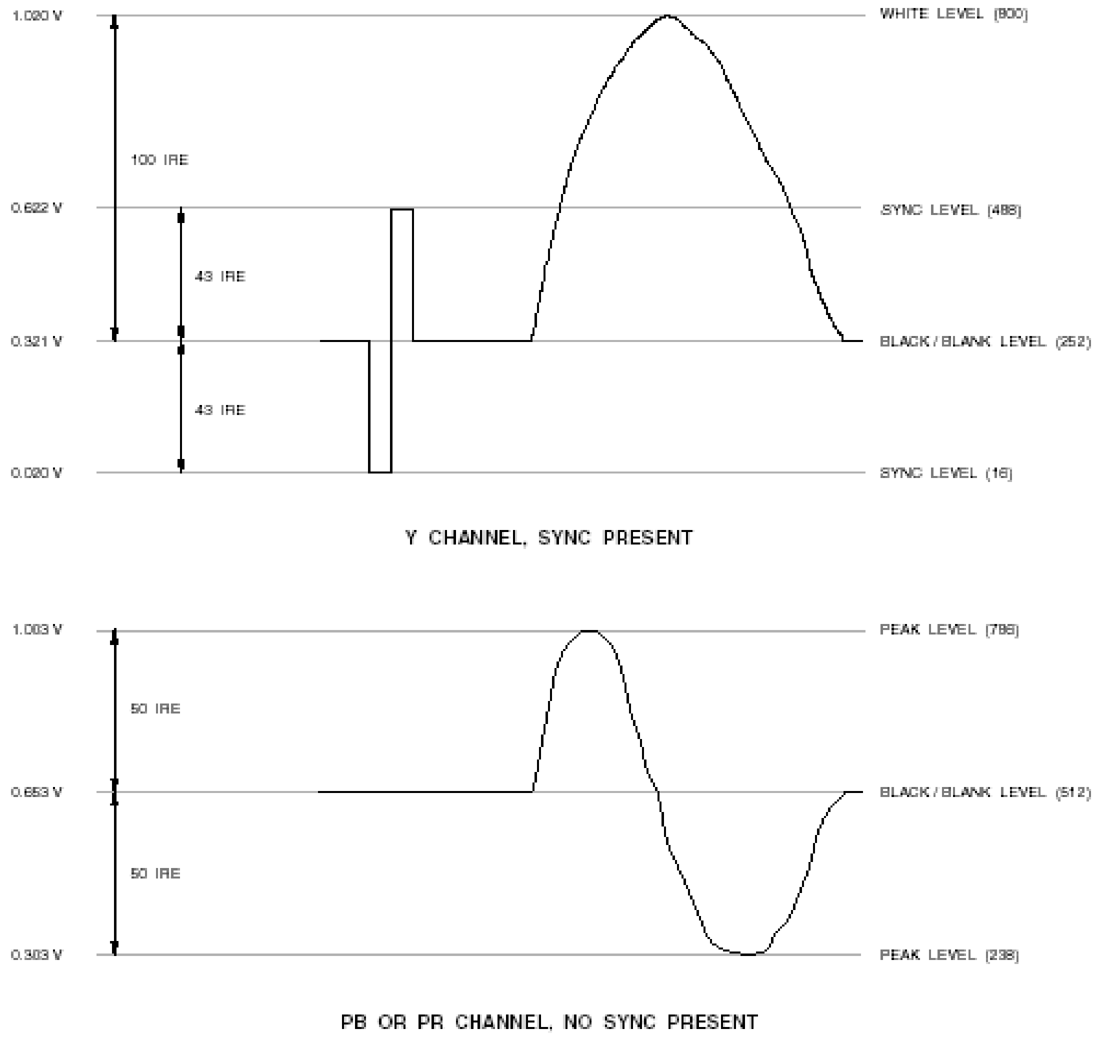
Trueview 5725 blanking insertion is used to fix some application blank level notconsistent. It can be SYNC on Y or SYNC on YPbPr.

or the SYNC levels, 9883 only can clamping the middle point, so the SYNC level will be misunderstand as active data to display, we insert the blank level can avoid this issue, when the data is blank level, we can force the data to a fixed value by program registers.

Reference Register Map:

Register Name	Register Address	Register Description
Hd_blk_gy_data	Reg_S3_53 [7:0]	Blanking insert GY data
Hd_blk_bu_data	Reg_S3_54 [7:0]	Blanking insert BU data
Hd_blk_rv_data	Reg_S3_55 [7:0]	Blanking insert RV data

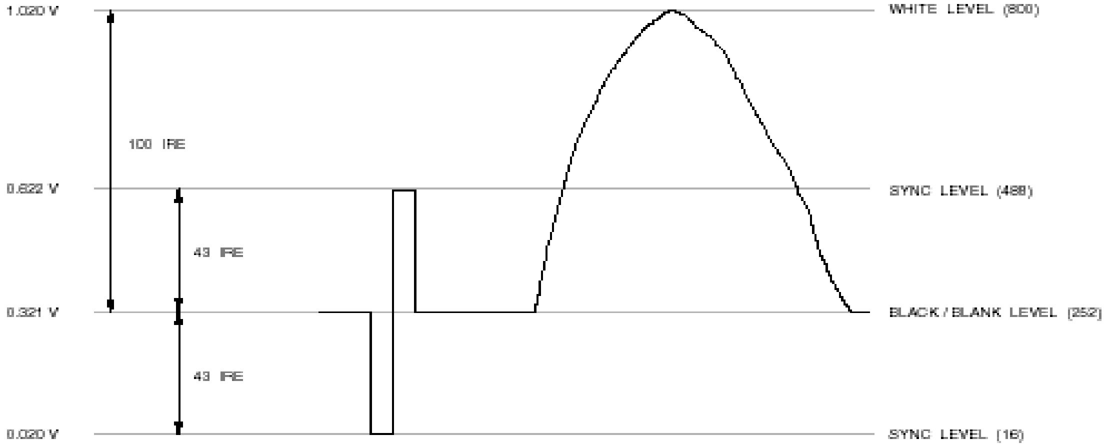
Firgure1: HDTV Analog YPbPr Levels, SYNC on Y.



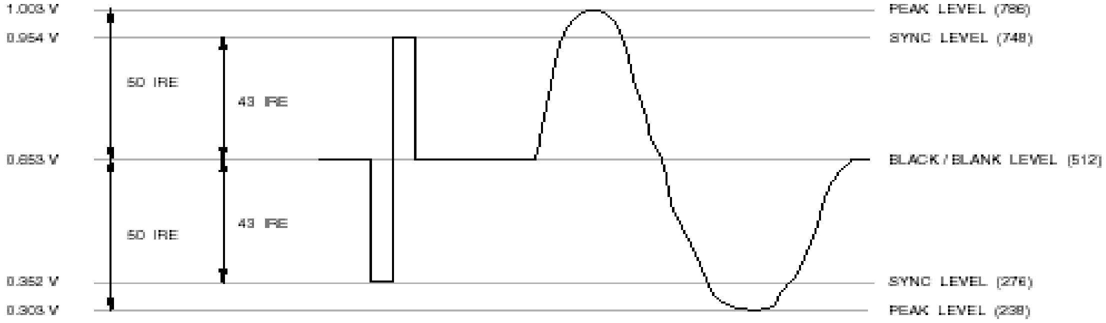
Y CHANNEL, SYNC PRESENT

PB OR PR CHANNEL, NO SYNC PRESENT

Firgure2: HDTV Analog YPbPr Levels, SYNC on YPbPr.



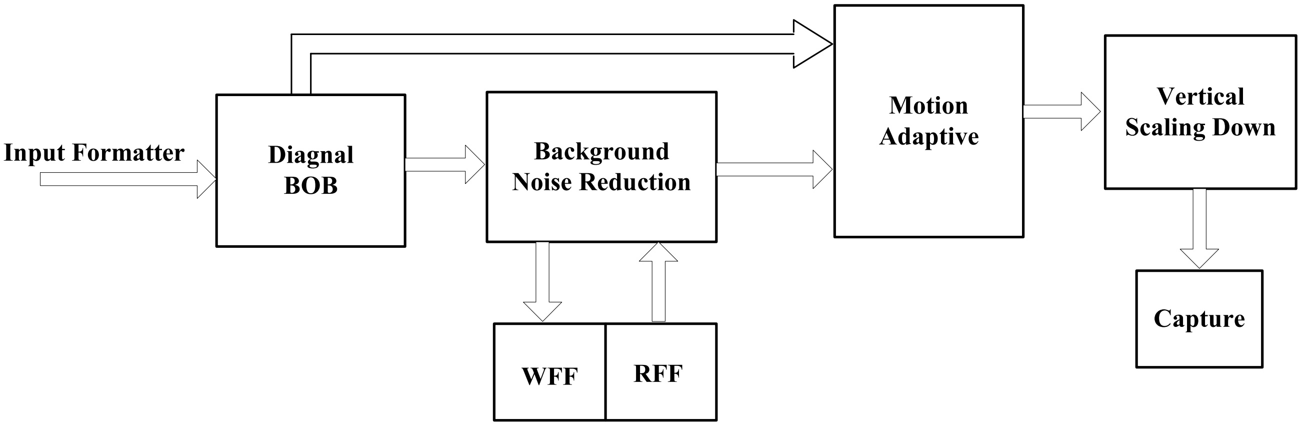
Y CHANNEL, SYNC PRESENT



PB OR PR CHANNEL, SYNC PRESENT

6 De-interlace

True View 5725 de-interlace is used to convert the interlace source to progressive source, when progressive input, we can bypass it, it includes three blocks: diagonal BOB de-interlace (diag_bob) and motion adaptive de-interlace (deint_madpt), and background noise reduction.



6.1 Diagonal BOB De-interlace (diag_bob)

Diag_bob will transfer interlace source to progressive scan data using pixel-base, programming angle detection to avoid jagged-edge artifacts.

6.1.1 Line Double

Line double interpolates one line through copying original line. So one line buffer FF1024 is used here. (Note: For Line Buffer capacity limitation, our horizontal total can't exceed 1024). Line double have some control bits named "**if_lin_db***". In golden setting provided by logic group, they have been adjusted. So don't modified these register unless input source is not standard NTSC/PAL.

Reference Register Map

Register Name	Register Address	Register Description
If_ld_sel_prov	Reg_S1_0b [7]	Line double select progressive line reset
If_ld_ram_byps	Reg_S1_0c [0]	Bypass line double FIFO, set it to 1 in progressive mode.
If_ld_st [3:0]	Reg_S1_0c [4:1]	Line double reset start position

Note: For progressive input, you need set register if_ld_sel_prov and if_ld_ram_byps to 1.

6.1.2 Line Pdelay

Line Pdelay delay the original data one line for angle detect and data interpolated. So one line buffer FF1024 is used here. (Note: For Line Buffer capacity limitation, our horizontal active can't exceed 1024).). Line double have some control bits named "**diag_bob_pldly**". In golden setting provided by logic group, they have been adjusted. So don't modified these register unless input source is not standard NTSC/PAL.

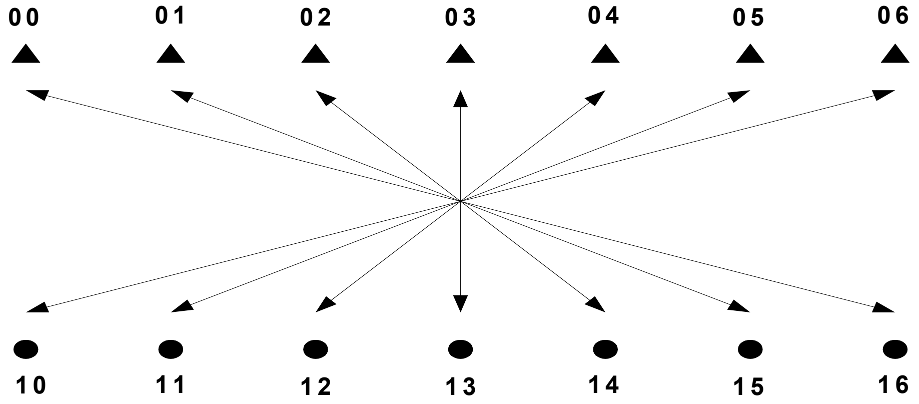
Reference Register Map:

Register Name	Register Address	Register Description
Diag_bob_pldy_ram_byps	Reg_S2_00 [7]	Bypass line delayer FIFO
Diag_bob_pldy_sp [8:0]	Reg_S2_02 [0] Reg_S2_01 [7:0]	The delay between line delayer read reset and write reset.

6.1.3 Angle detection

Angle detection detect two line's angle in pixel base**.** In Angle detection, there are maximum 7 angles used to detect most similar pixels (see blew figure). The couple pixels that have minimum difference are selected. And the interpolated pixel is generated through averaging the most similar pixels. The angle level could be selected by register bit " **diag_bob_det_byps[1:0]**". And all angle detection could be bypassed by control bit "**diag_bob_min_byps**" and then only vertical pixel (03 and 13 in figure) average is used.

Note: In 5725, angle detection will be bypassed when high frequency/less still detection is active. About high frequency and less still, please read 6.2.5.1 and 6.2.5.2.
By default, both Y and UV data are used for angle detected. In 5725, there is one register bit "**diagbob__min_cbyps**" could bypass UV data in angle detection.



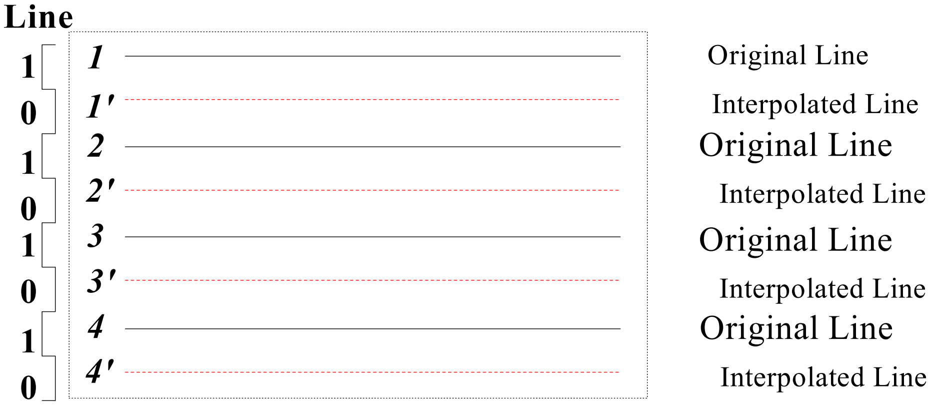
6.1.4 Weave Logic

Weave logic is to merge original line and interpolation line (which is generated by angle detection) together to generate a progressive scan data.

The main control signal of weave logic is LINE from input formatter.

- When LINE=1, original line will output;
- when LINE=0, interpolation line will output.

There is a control bit "**weave_byps**" could disable LINE. When "**weave_byps**" equals one, output data will always be original line.



Reference Registers Map:

Register Name	Register Address	Register Description
Diag_bob_min_byps	Reg_S2_00 [0]	=1, bypass diagonal bob, only average two vertical pixels
Diag_bob_coef_sel	Reg_S2_00 [1]	Select low pass filter coefficients for pixel difference.
Diag_bob_weave_byps	Reg_S2_00 [2]	=1, bypass diagonal bob, only average two vertical pixels
Diag_bob_det_byps	Reg_S2_00 [4:3]	Bit0 on, bypass arc tan (1/4), Bit1 one, bypass arc tan (1/6)
Diag_bob_ytap3_byps	Reg_S2_00 [5]	Bypass the diagonal bob interpolate filter.
Diag_bob_min_cbyps	Reg_S2_00 [6]	Bypass diagonal bob UV min selection.

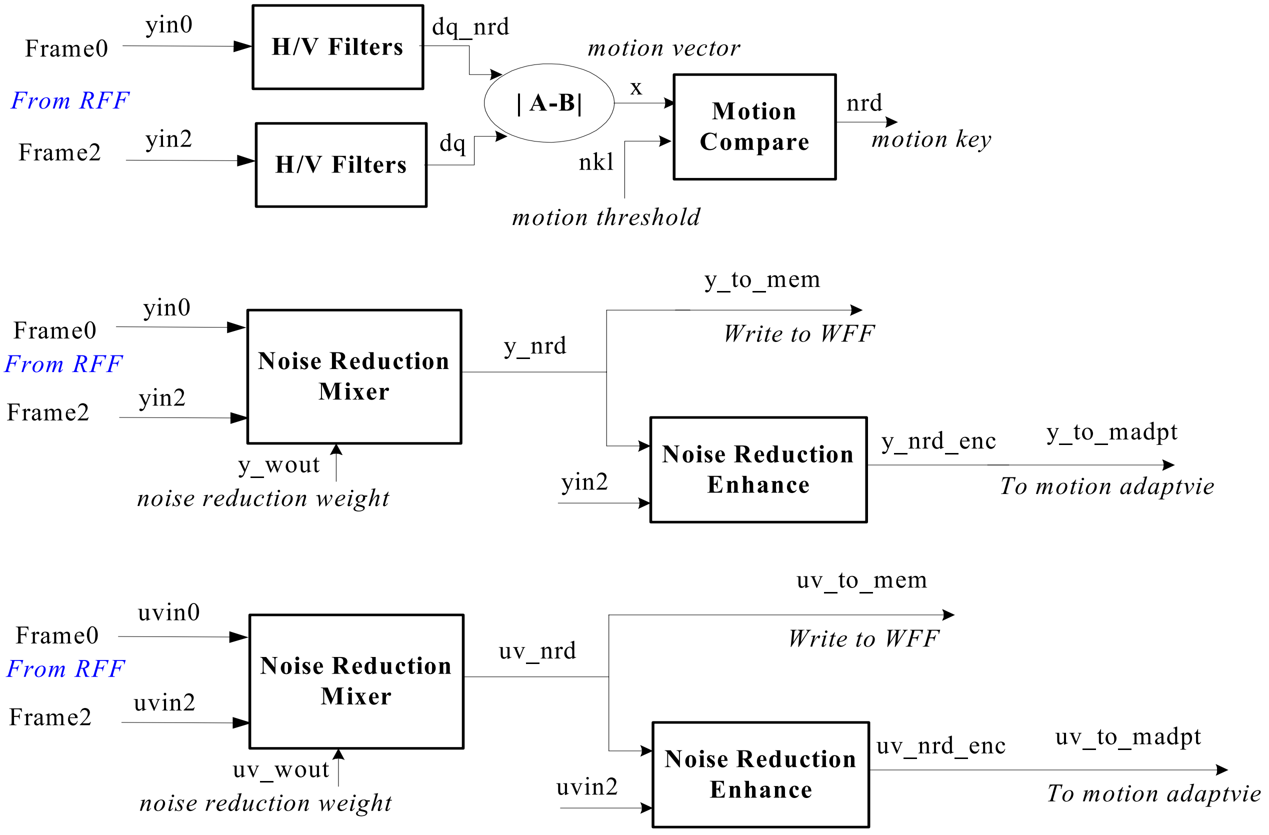
6.2 Background Noise Reduction

Background noise reduction is a process that uses a digital filtering algorithm on the digital image data to reduce the amount of random noise (like film grain, RF noise, comb filter artifacts in composite video sources, film speckles, dirt, scratches etc.).

TrueView 5725 background noise reduction is motion adaptive 3D noise reduction. Firstly, it detect the motion vector on a pixel by pixel basis, then it compare the motion vector with the defined motion threshold (Register defined value), if motion vector larger than the motion threshold, it will bypass noise reduction, if motion vector is smaller than the motion threshold, it will do noise reduction.

Just as a coin has two sides, background noise reduction maybe has the trailing artifact, so we must give a balance for noise and motion!

Note: There are 2 groups threshold and H/V filter controls, one is for normal condition, and the other is for high-noise/still condition. For moving condition, we will not do or do less background noise reduction to avoid artifact; for still or noisy condition, we will do more background noise reduction to improve the picture quality!



6.2.1 Motion Key Generation

Motion key is used to decide the background noise reduction do or not, ~~if it is 0, it indicates the video~~ source is motion stream and will not do background noise reduction, if it is 1, it indicates the video source is still stream and do noise reduction.

In auto mode, Firstly, frame0 and frame2 data via Vertical and Horizontal IIR to filter the high frequency noise, then calculate the difference pixel by pixel, if the difference less then the register defined threshold, nrd = 1, else nrd = 0;

In manual mode, nrd will equal to the manual ID you defined.

Reference Registers Map:

Register Name	Address	Register Description
Madpt_y_htap_cntrl [3:0]	Reg_S2_32 [3:0]	Y horizontal filter control for background noise reduction
Madpt_y_vtap_cntrl [2:0]	Reg_S2_32 [6:4]	Y vertical filter control for background noise reduction
Madpt_nrd_sel	Reg_S2_32 [7]	=1: nrd_mixer = yin2 + yin0; =0: nrd_mixer = yin2 + yin0_vtap_hlpf.
Madpt_m_htap_cntrl [3:0]	Reg_S2_33 [3:0]	Y horizontal filter control for background noise reduction in huge noise condition
Madpt_m_vtap_cntrl [2:0]	Reg_S2_33 [6:4]	Y vertical filter control for background noise reduction in huge noise condition
Mapdt_cmp_en	Reg_S2_35 [6]	Motion compare auto mode enable, set to 0, in manual mode, 1, auto mode
Madpt_cmp_user_id	Reg_S2_35 [7]	Motion compare ID when manual mode
Madpt_cmp_low_threshold [7:0]	Reg_S2_36 [7:0]	Motion compare low-level threshold.
Madpt_cmp_high_threshold [7:0]	Reg_S2_37 [7:0]	Motion compare high-level threshold, used when still or high noise condition.

6.2.2 Noise reduction operation

6.2.2.1 Noise reduction mixer

Noise reduction mixer is used to blend the images between 2 fields, Y and UV can control respectively. If you set wout_byps to 1, it will not do background noise reduction; if you set wout_byps to 0,
You can set the wout value to control the background noise reduction volume! 0 is max, 6 is min, 7 will freeze the screen and display the RFF data.

Wout_byps	Wout	Out
1	*	A
0	0	(1*A + 7*B) / 8
0	1	(2*A + 6*B) / 8
0	2	(3*A + 5*B) / 8
0	3	(4*A + 4*B) / 8
0	4	(5*A + 3*B) / 8
0	5	(6*A + 2*B) / 8
0	6	(7*A + 1*B) / 8
0	7	B

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_y_wout_byps	Reg_S2_34 [0]	Bypass Y noise reduction mixer
Madpt_y_wout [2:0]	Reg_S2_34 [3:1]	Coefficient for Y noise reduction
Madpt_uv_wout_byps	Reg_S2_34 [4]	Bypass UV noise reduction mixer
Madpt_uv_wout [2:0]	Reg_S2_34 [7:5]	Coefficient for UV noise reduction

6.2.2.2 Noise reduction enhancer

Noise reduction enhancer is used when heavy noise environment, it will increase the background noise reduction weight to enhance the background noise reduction performance. It is only a FIR filter:

Out = (A + B) / 2

For Y-channel, You can enable it when heavy noise environment, for normal environment, you should bypass it. For UV-channel, it will always do noise reduction enhancer.

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_nrd_out_sel	Reg_S2_35 [2]	Y-channel noise reduction enhancer enable, set to 1 for heavy noise condition

6.2.3 Noise reduction control

Noise reduction Y-channel and UV-channel can control separately, if you want enable UV background noise reduction, you must enable UV-deinterlacer.

In Background noise reduction mode:

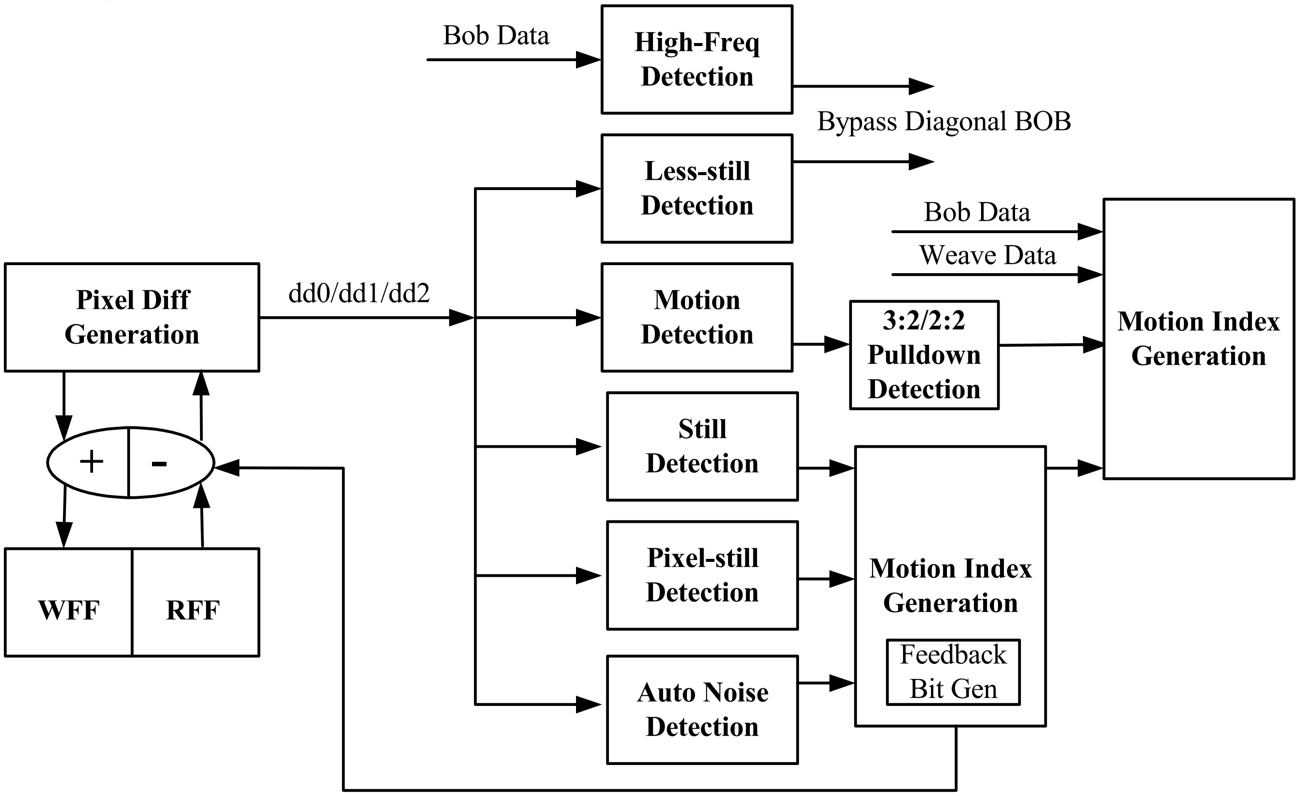
Register Name	Register Address	Value	Register Description
Madpt_y_nrd_en	Reg_S2_35 [0]	1	Sets to 1 enable Y-channel noise reduction.
Madpt_uv_nrd_en	Reg_S2_35 [1]	1	Sets to 1 enable UV-channel noise reduction.
Madpt_dd0_sel	Reg_S2_35 [3]	0	Set to 0 in background noise reduction mode
Madpt_y_htap_cntrl	Reg_S2_32 [3:0]	Option	Set the coefficient for H-filter
Madpt_y_vtap_cntrl	Reg_S2_32 [6:4]	Option	Set the coefficient for V filter
Madpt_m_htap_cntrl	Reg_S2_33 [3:0]	Option	Set the coefficient for H-filter
Madpt_m_vtap_cntrl	Reg_S2_33 [6:4]	Option	Set the coefficient for V-filter
Madpt_vt_filter_cntrl	Reg_S2_16 [6]	1	Do VT filter every line

In No Background noise reduction mode:

Register Name	Register Address	Value	Register Description
Y_NRD_EN	Reg_S2_35 [0]	0	Sets to 0 disable Y-channel noise reduction.
UV_NRD_EN	Reg_S2_35 [1]	0	Sets to 0 disable UV-channel noise reduction.
DD0_SEL	Reg_S2_35 [3]	1	Set to 1 without background noise reduction.
Y_HTAP_CNTRL	Reg_S2_32 [3:0]	4'hf	Bypass the coefficient for H-filter
Y_VTAP_CNTRL	Reg_S2_32 [6:4]	3'h7	Bypass the coefficient for V filter
M_HTAP_CNTRL	Reg_S2_33 [3:0]	4'hf	Bypass the coefficient for H-filter
M_VTAP_CNTRL	Reg_S2_33 [6:4]	3'h7	Bypass the coefficient for V-filter
VT_FILTER_CNTRL	Reg_S2_16 [6]	0	Do VT filter in interpolated line

6.3 Motion Adaptive De-interlace

Motion Adaptive De-interlace (deint_madpt) automatically determines the type of incoming video content-film; static interlaced video and moving interlace video. Different algorithms are applied for each of content type. 5725's motion adaptive De-interlace is for YUV domain adaptive.



Motion adaptive de-interlace will come from background noise reduction and diagonal BOB:

(1) Background noise reduction data includes 3 frames data, last 2 frame, last frame and the current frame data.

(2) Diagonal BOB data only refer to the current frame BOB data.

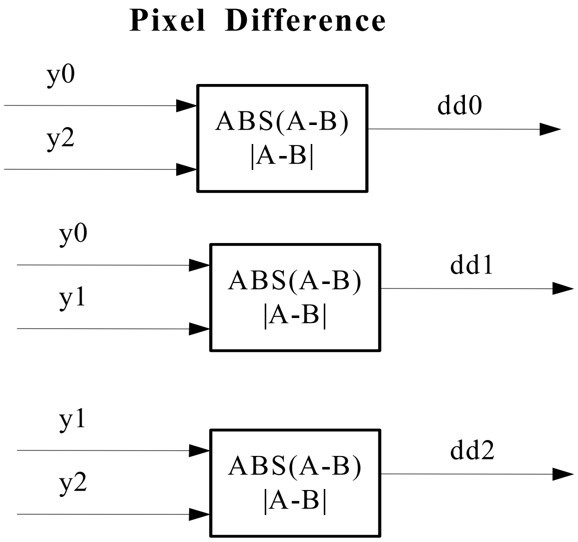
6.3.1 Pixel Difference Generation

Pixel difference Generation is to generate pixel-base difference value (absolute value) between timing domains. In TrueView 5725, total 3 frames Y data (Y0, Y1, Y2) is used. These Y data is the result of background noise reduction and read from RFF. So they have been removed the background noise.

Because Y0, Y1, Y2 are still treated to have odd/even polarity, Y0, Y1, Y2 field polarity should be odd, even, odd or even, odd, even.

DD0: |Y0 – Y2|, It has the same field polarity, so they will be used for still, less still, block still, and 3:2 pulldown detection.
DD1: |Y1 – Y0|, It is used for 2:2 pulldown detection.
DD2: |Y2 – Y1|, It is used for 2:2 pulldown detection.

Below is the pixel difference diagram:

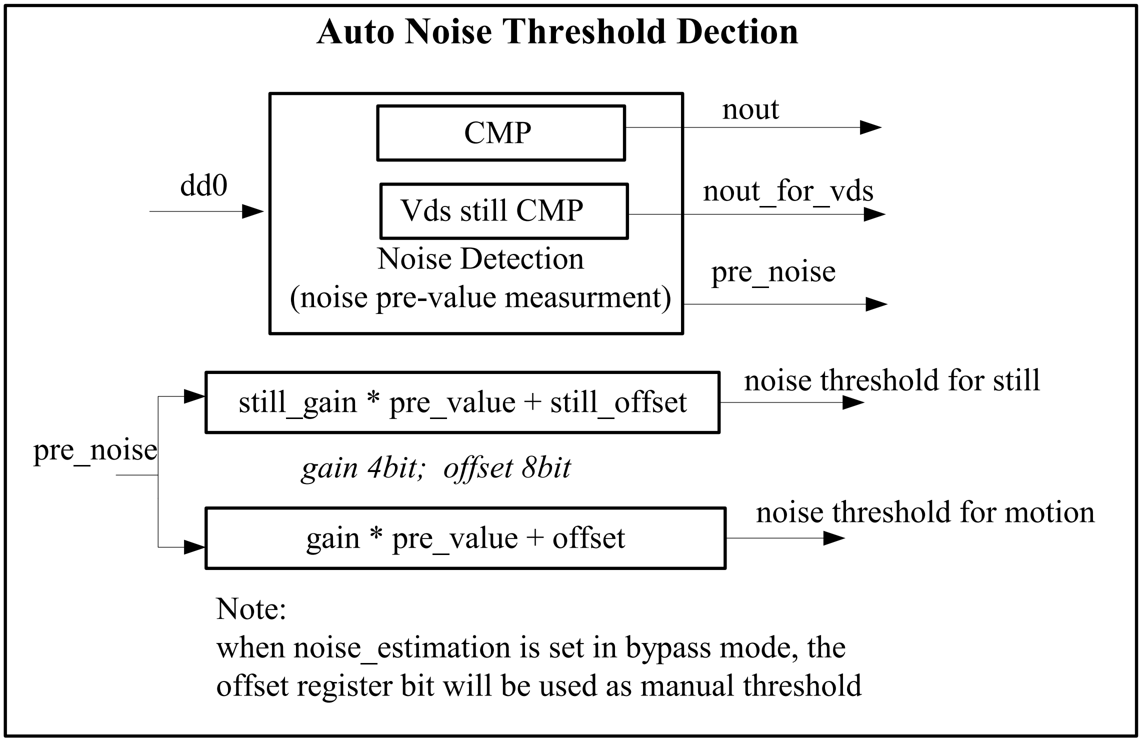


Note: when enable background noise reduction, dd0_sel should set to 1; when disable background noise reduction, dd0_sel should set to 0.

6.3.2 Auto noise detection

Auto noise detection is used to estimate the noise value in video data automatically. It can change its noise threshold based on different noise condition. In 5725 there are two groups noise value generated for still detection, and motion detection.

Below is the auto noise detection diagram:


Auto Noise Threshold Dection
CMP
nout
dd0
Vds still CMP
nout_for_vds
Noise Detection
(noise pre-value measurment)
pre_noise
still_gain * pre_value + still_offset
noise threshold for still
pre_noise
gain 4bit; offset 8bit
gain * pre_value + offset
noise threshold for motion
Note:
when noise_estimation is set in bypass mode, the
offset register bit will be used as manual threshold

6.3.2.1 Pre_value measured

1) Noise pre-value is measured; there is a control bit "**madpt_noise_det_sel**" (Reg_S2_03 [0]) to select noise pre_value measurement in active timing or in blanking timing.
2) Noise pre-value measurement is based on pixel difference between Y0 and Y2. (Because Y0 and Y2 must be in same field polarity.) Pre-value will accumulate the pixel's difference through a looooooong counter (about 0.39 second) as a cycle.
3) Every auto noise detection period is composed of 128/256/512/1024 cycles (defined by **madpt_noise_det_rst** (Reg_S2_03 [4:3])), it will select the minimum accumulate value as this auto noise detection period pre-value,
4) For the auto noise detection period pre-value, there is auto noise detection shift control (**madpt_noise_det_shift** (Reg_S2_03 [2:1])) to select its precision.
5) At the same time, the auto detection period pre-value will be compared with **madpt_noise_threshold_nout** (Reg_S2_04 [6:0]) to generate nout, and compared with **madpt_noise_threshold_vds** (Reg_S2_05 [6:0]) to generate nout_vds.

Note:

<1> Nout will be high if pre-value greater than **madpt_noise_threshold_nout**;

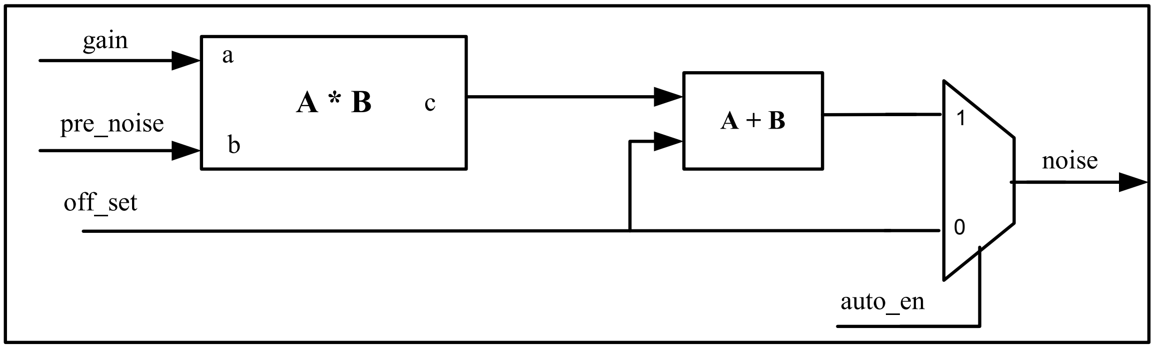
- It will bypass diagonal BOB if **madpt_en_nout_for_less_still** (Reg_S2_21 [5]) set to 1;
- It will select high noise reduction if **madpt_en_nout_for_nrd** (Reg_S2_3a [2]) set to 1;
- It will divide motion index if **enable_nout_still** (Reg3e [5]) set to 1.

<2> Nout_vds will be high if pre-value greater than **madpt_noise_threshold_vds**:

- It will enable vds_proc high noise 3D noise reduction.

6.3.2.2 Noise level estimated

For still detection, and motion detection there are two separated, same-function noise estimation modules. Each module has own gain control; offset control and manual programmable value. (Manual programmable value is shared with gain/offset bits which selected by one enable bit.)



Auto noise estimation register map			
Estimate	**Gain**	**Offset**	**Auto Enable**
Motion	Reg_S2_09 [3:0]	Reg_S2_06 [7:0]	Reg_S2_0a [4]
Still	Reg_S2_09 [7:4]	Reg_S2_07 [7:0]	Reg_S2_0a [5]

Note: Gain is 0~2x. Offset is 0~1020.

6.3.3 3:2 pulldown mode detection

6.3.3.1 Motion detection

Motion detection is to detect single frame is motion or not. The result will be sent to 3:2 pulldown detection to control the pulldown sequence and reset logic. Motion detection will accumulate whole active timing's pixel difference in one frame. Then the result is compare with noise accumulation result.

If Acc_pixel > Acc_noise, motion;
If Acc_pixel <= Acc_noise, no motion

Because 3:2 pulldown detection algorithm, pixel difference **DD0** is used to guarantee they have the same field polarity. If the auto mode enable, the **pre_noise * gain + offset** will be the noise threshold, if the manual mode enable, the **offset** will be the noise threshold.

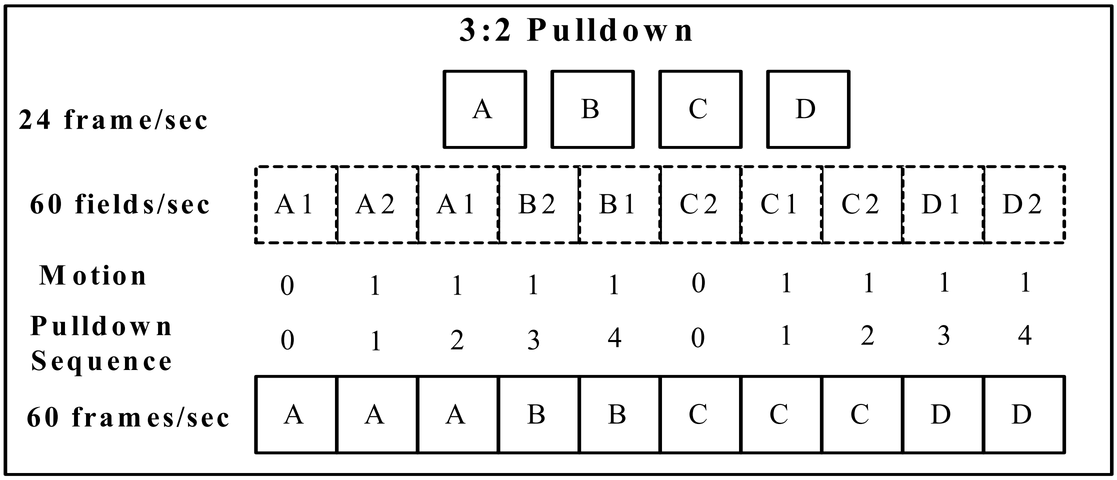
Reference Register Map:

Register Name	Register Address	Register Description
Madpt_gm_noise_value [7:0]	Reg_S2_06 [7:0]	In manual mode, motion noise threshold; In auto mode, motion noise offset
Madpt_noise_est_gain [3:0]	Reg_S2_09 [3:0]	Global noise gain value in auto mode.
Madpt_noise_est_en	Reg_S2_0a [4]	=1, enable motion noise auto detection; =0, disable motion noise auto detection.

6.3.3.2 3:2 pulldown estimation

Because film is 24 frames per second, NTSC source is 60 fields per second. When film is converted to NTSC source, some operations are needed just like figure below. These operations call 3:2 pulldown. 3:2 pulldown estimation is to convert the NTSC fields to original frame as A, B, C, D.

3:2 pulldown detection need find out these special patterns through detecting frame motion (It must be 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 ………). There is a counter in 3:2 pulldown detection to count special pattern's repeat times. When the counter value is equal programmable bits "**madpt_32pulldown_lock_rst**", 3:2 pulldown lock ID will go high (active). Once input pattern is not 3:2 pulldown's special pattern, the count will be reset, 3:2 pulldown lock ID will go low. The lock ID also could be in manual mode, which is controlled by "**madpt_32pulldown_id**" and "**madpt_en_32pulldown**".



60 fields/sec	A1	A2	A1	B2	B1	C2	C1	C2	D1	D2
Motion	0	1	1	1	1	0	1	1	1	1
Pulldown Sequence	0	1	2	3	4	0	1	2	3	4
60 frames/sec	A	A	A	B	B	C	C	C	D	D

3:2 pulldown sequence will provide recent frame's position in 3:2 pattern (in A1? A2? B1…). The sequence is generated by a counter. And there are control bits "**madpt_32pulldown_offset**" to adjust the sequence.

Note: For 3:2 pulldown, you can observe the Faroudja falsh-line pattern to guarantee

Reference Register Map:

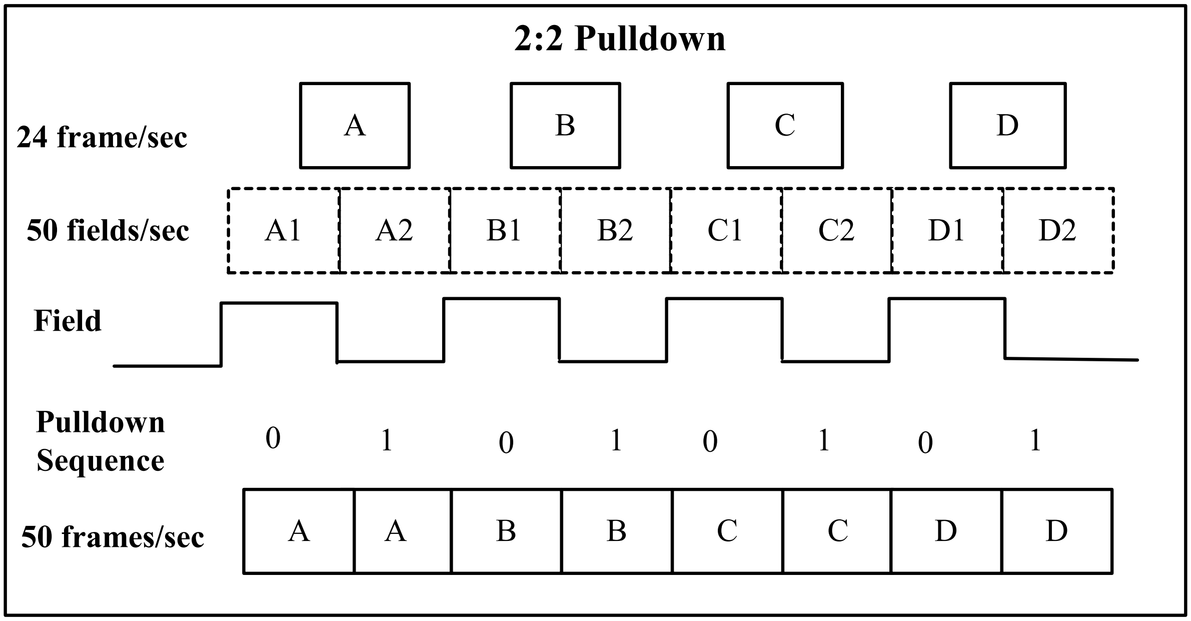
Register Name	Register Address	Register Description

Madpt_en_pulldown32	Reg_S2_11 [3]	=1, 32pulldown auto mode;
		=0, 32pulldown manual mode;

Madpt_pulldown32_id	Reg_S2_11 [4]	User's 32pulldown ID in manual mode
Madpt_pulldown32_offset [2:0]	Reg_S2_11 [7:5]	Sequence offset adjust to make 3:2 pull down sequence right
Madpt_pulldown32_lock_rst [6:0]	Reg_S2_12 [6:0]	Frame lock counter for 3:2 pulldown

6.3.4 2:2 pulldown mode detection

2:2 pulldown is the operation that 24Hz film is converted to 50Hz PAL source. 24Hz film will be separated to 48 filed and display in higher speed.



Just as the above diagram, we accumulate the field difference a long time, we can control the accumulate time through set **madpt_22pulldown_det_cntrl**. if the accumulated value is larger than the threshold **madpt_pulldown22_threshold**, we think it is 2:2 pulldown mode; if the accumulated value less then the threshold, we thinks it is a normal interlace mode.

2:2 pulldown sequences is controlled by field signal, we can flip the 2:2 pulldown sequence through set **madpt_pulldown22_offset,** and we can set 2:2 pulldown manual mode through set **madpt_en_pulldown22** and **madpt_pulldown22_id**.

Reference Register Map:

Register Name	Register Address	Register Description

Madpt_en_pulldown22	Reg_S2_13 [0]	=1, 22pulldown auto mode;
		=0, 22pulldown manual mode;

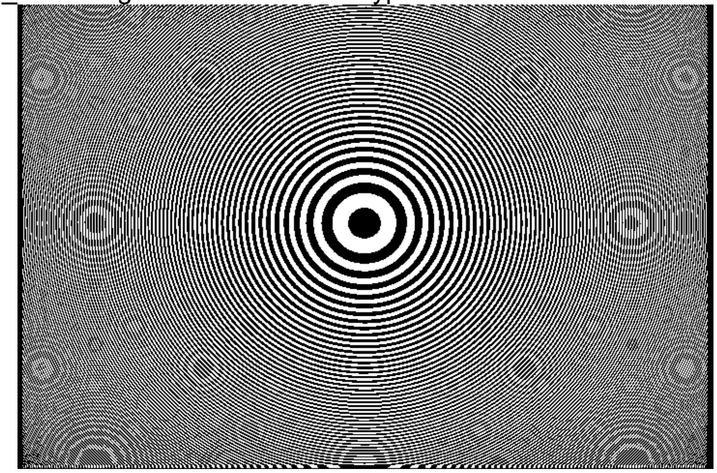
Madpt_pulldown22_id	Reg_S2_13 [1]	User's 22pulldown ID in manual mode
Madpt_pulldown22_offset	Reg_S2_13 [2]	Sequence offset adjust to make 2:2 pull down sequence right
Madpt_pulldown22_det_cntrl [2:0]	Reg_S2_13 [5:3]	2:2 pull down detect period select
Madpt_pulldown22_threshold [17:0]	Reg_S2_16 [1:0], Reg_S2_15 [7:0], Reg_S2_14 [7:0],	2:2 pull down detect threshold.

6.3.5 Mode detection

5725 De-interlace processor includes three mode detection modules: high-frequency detection, less_still detection, still detection and pixel still detection. Their algorithm is similar except some little difference. High-frequency detection is based on pixel difference in the same frame. Less_still and still are based on **DD0** in timing domain. High-freq, less_still, still are frame based, but pixel still is pixel based.

6.3.5.1 High-frequency detection

In 5725, some patterns (just like the figure below) with high-frequency may cause diag_bob's angle detection do wrong adjustment. High-frequency detection is to find out these patterns in frame-base. If high-frequency go active, diag_bob's angle detection will be bypassed.



High-frequency detection first detect in single frame. Accumulate absolute pixel difference in one frame's active timing. Do same accumulation in noise threshold **madpt_hfreq_noise** in same timing.

If Acc (pixel) > Acc (threshold), this single frame is high-frequency pattern.

Else, this single frame is not high-frequency pattern.

Then there is a counter to count how much frames high-frequency frames will repeat. Once the counter value is larger than control bits ("**madpt_hfreq_lock**"), high-frequency detection output will go active.

At the same time, there is a unlock counter to count how many frames non-high-frequency frame will repeat. Once this unlock counter value is larger than control bits ("**madpt_hfreq_unlock**"), high-frequency detection output will go passive.

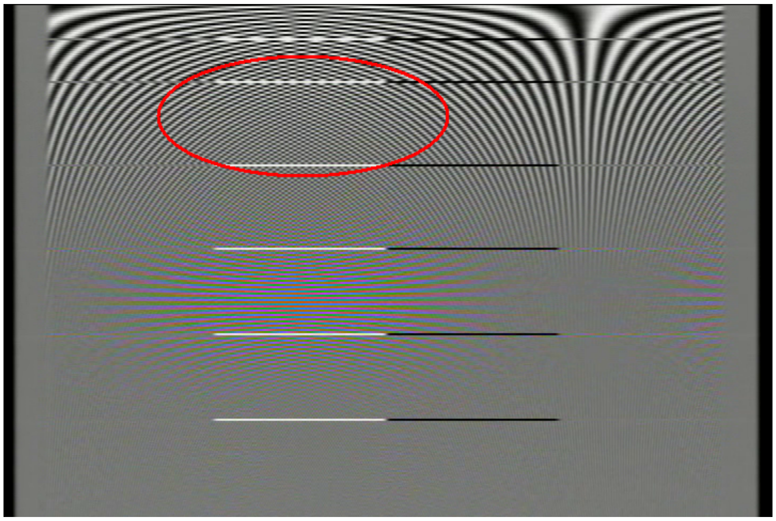
High-frequency also support manual mode that high-frequency detection result could be forced to zero/one through programming register bit ("**madpt_hfreq_id**" and **madpt_hfreq_det_en**").

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_hfreq_det_en	Reg_S2_20 [0]	=1, high frequency detection in auto mode; =0, high frequency detection in manual mode;
Madpt_hfreq_id	Reg_S2_20 [1]	User's high frequency ID in manual mode
Madpt_hfreq_lock [3:0]	Reg_S2_20 [7:4]	Lock counter for high_freq detection
Madpt_hfreq_unlock [2:0]	Reg_S2_21 [2:0]	Unlock counter for high_freq detection
Madpt_hfreq_noise [7:0]	Reg_S2_1f [7:0]	High_freq noise threshold value

6.3.5.2 Less_still detection

Less_still detection use Y0-Y2 pixel difference for Y0 and Y2 have the same filed polarity. Less_still will bypass diag_bob to avoid diag_bob's artifact in still picture. Because diag_bob may increase background noise.



Firstly, accumulate pixel difference **DD0** in one frame active period. Do same accumulation on **madpt_less_noise_value** in same timing.

If Acc(DD0) > Acc(less_still_noise), recent Y0 and Y2 could be treated as less_still frame. Else, recent Y0 and Y2 could be treated as non_less_still frame.

And there is a counter count how much frames less_still frame will repeat. Once counter value is larger than control bits ("**madpt_less_still_lock**"), less_still detection output will go active.

At the same time, there is a unlock counter to count how much frames non_less_still frame will repeat. Once this unlock counter value is larger than control bits ("**madpt_less_still_unlock**"), less_still detection output will go passive.

Less_still also support manual mode that less_still detection result could be forced to zero/on thorough programming register bit ("**madpt_less_still_id**" and "**madpt_less_still_det_en**").

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_less_still_det_en	Reg_S2_10 [0]	=1, less still detection in auto mode; =0, less still detection in manual mode;
Madpt_less_still_id	Reg_S2_10 [1]	User's less still ID in manual mode
Madpt_less_still_lock [3:0]	Reg_S2_10 [7:4]	Lock counter for less still detection
Madpt_less_still_unlock [1:0]	Reg_S2_10 [3:2]	Unlock counter for less still detection
Madpt_less_still_noise_value [7:0]	Reg_S2_08 [7:0]	Less still noise threshold value

6.3.5.3 Still detection

Still detection use **DD0** pixel differences for Y0 and Y2 have the same-filed polarity. If input source is still status, motion index will be divided by two or 4.

Firstly, accumulate pixel difference in one frame active timing. Do same accumulation on **madpt_still_noise_value** in the same timing.
If Acc (DD0) > Acc (still_noise), recent Y0 and Y2 could be treated as still frame.
Else, recent Y0 and Y2 could be treated as non_still frame.

And there is a counter count how much frames still frame will repeat. Once counter value is larger than control bits ("**madpt_still_lock**"), still detection output will go active.
At the same time, there is a unlock counter to count how much frames non_still frame will repeat. Once this unlock counter value is larger than control bits ("**madpt_still_unlock**"), still detection output will go passive.
Still also support manual mode that still detection result could be forced to zero/on thorough programming register bit ("**madpt_still_id**" and "**madpt_still_det_en**").

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_still_det_en	Reg_S2_0f [0]	=1, still detection in auto mode; =0, still detection in manual mode;
Madpt _still_id	Reg_S2_0f [1]	User's still ID in manual mode
Madpt_still_lock [3:0]	Reg_S2_0f [7:4]	Lock counter for still detection
Madpt_still_unlock [1:0]	Reg_S2_0f [3:2]	Unlock counter for still detection
Madpt_still_noise_value [7:0]	Reg_S2_07 [8:0]	In manual mode, still noise threshold; In auto mode, still noise offset.
Madpt_divid_sel	Reg_S2_18 [1]	=1, MI divided by 2 in still, =0, MI divided by 4 in still

6.3.5.4 Pixel still generation

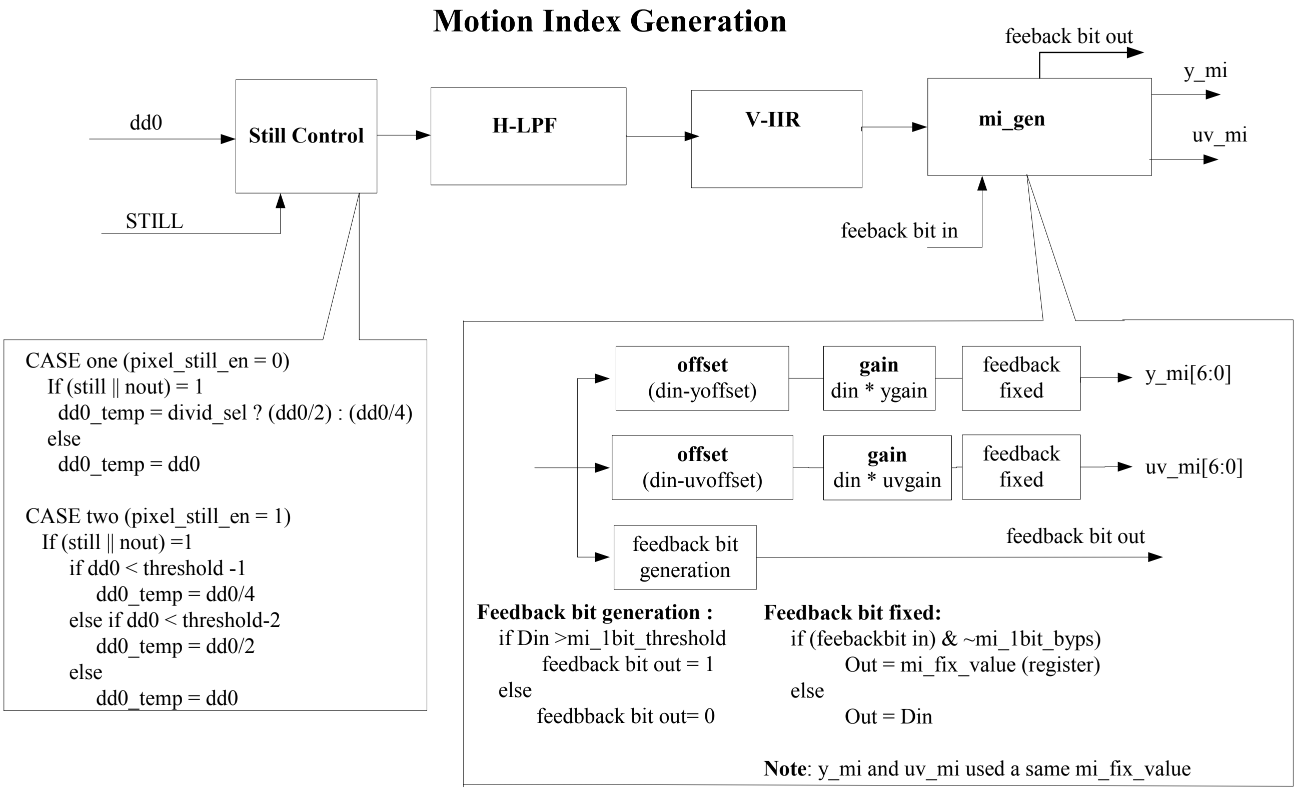
Pixel still is mainly used to decrease the small motion artifact, it is pixel based, when pixel difference **DD0** larger then the threshold value, the motion index will be divided by 2 or 4 for the still frame. It will not affect the non-still frame:
When the frame is still frame and you enable pixel still:
When madpt_pixel_still_threshold_1 < DD0 < madpt_pixel_still_threshold_2, MI divided by 2;
When DD0 < madpt_pixel_still_threshold_1, MI divided by 4;
When DD0 > madpt_pixel_still_threshold_2, MI not divided.

For pixel still working mode:

Register Name	Register Address	Value	Register Description
Madpt_Pixel_still_threshold_1	Reg_S2_1a [7:0]	Option	Define the pixel still threshold one
Mapdt_Pixel_still_threshold_2	Reg_S2_lb [7:0]	Option	Define the pixel still threshold two
Madpt_bit_still_en	Reg_S2_19 [0]	1	Enable pixel still working mode

6.3.6 Motion Index generation

Blew is motion index estimation's path:



```
CASE one (pixel_still_en = 0)
   If (still || nout) = 1
     dd0_temp = divid_sel ? (dd0/2) : (dd0/4)
   else
     dd0_temp = dd0

CASE two (pixel_still_en = 1)
  If (still || nout) =1
       if dd0 < threshold -1
            dd0_temp = dd0/4
       else if dd0 < threshold-2
            dd0_temp = dd0/2
       else
            dd0_temp = dd0
```

Feedback bit generation :

```
  if Din >mi_1bit_threshold
        feedback bit out = 1
  else
        feedbback bit out= 0
```

Feedback bit fixed:

```
  if (feebackbit in) & ~mi_1bit_byps)
        Out = mi_fix_value (register)
  else
        Out = Din
```

Note: y_mi and uv_mi used a same mi_fix_value

Motion index is based on pixel difference **DD0** value. When still and pixel still go active, there is a logic controlled by "**madpt_divid_sel**", "**madpt_divid_byps**" and "**madpt_bit_still_en**".

Madpt_bit_still_en = 0, (Reg_S2_19 [0]) Pixel still disable mode		
Madpt_divid_byps (Reg_S2_18 [0])	**Madpt_divid_sel (Reg_S2_18 [1])**	**Motion Index**
0	0	When divid_nout | still =1, difference ¼
0	1	When divid_nout | still =1, difference ½
1	X	Difference bypass
Madpt_bit_still_en = 1, (Reg_S2_19 [0]) Pixel still enable mode		
Still	**Condition**	**Motion Index**

1	DD0 < threshold_1	Difference ¼
1	DD0 < threshold_2	Difference ½
1	DD0 >= threshold_2	Difference bypass

6.3.6.1 Horizontal Domain Filter

Horizonal coefficient is programmable. Register bit is (“**Madpt_htap_coeff**”)
You can bypass the horizontal filter via Register bit is (“**Madpt_htap_byps**”)
Reference Register Map:

Register Name	Register Address	Register Description
Madpt_htap_byps	Reg_S2_18 [3]	=1, Bypass Motion index horizontal filter
Madpt_htap_coeff [3:0]	Reg_S2_18 [7:4]	Coefficient of Motion index horizontal filter

6.3.6.2 Vertical Domain Filter

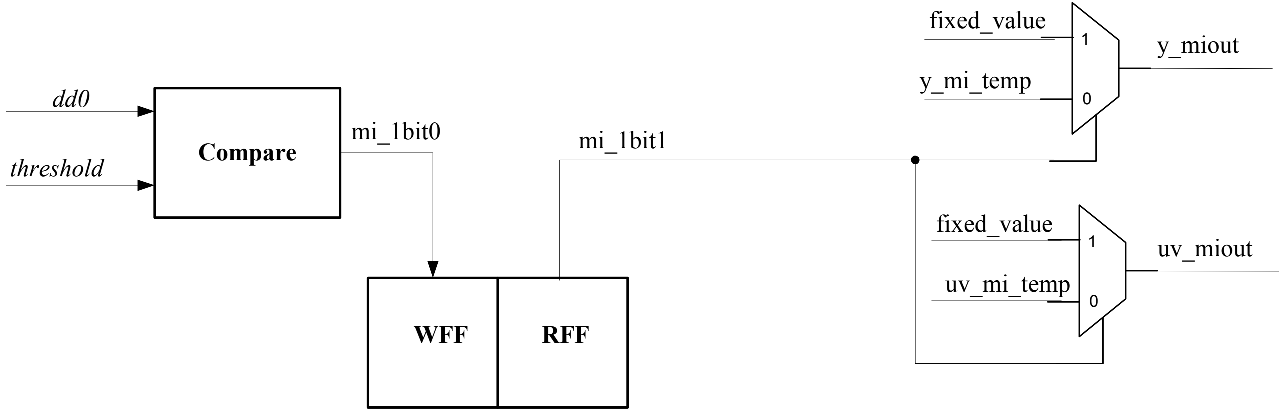
Vertical IIR filter input data round selected between 1 and ½ by control bit (“**Madpt_vtap2_round_sel**”).
Vertical IIR filter Coefficient is programmable through register bit (“**Madpt_vtap2_coeff**”).
And whole Vertical-IIR filter could be bypassed. Control bit is (“**Madpt_vtap2_byps**”).

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_vtap2_byps	Reg_S2_19 [2]	=1, Bypass Motion index vertical filter
Madpt_vtap2_round_sel	Reg_S2_19 [3]	=1, input for MI V filter will be divided by 2
Madpt_vtap2_coeff [3:0]	Reg_S2_19 [7:4]	Coefficient of Motion index vertical filter

6.3.6.3 Feedback Bit Estimation

Feedback bit is mainly decrease the trailing artifact for the big motion, when the pixel difference **DD0** larger than the threshold, then mi_1bit0 feedback bit will be 1 and write to WFF, then we read the feedback bit from RFF named mi_lbit1 and mi_1bit2 for last 1 frame & last 2 frame feedback bit. When the feedback bit is 1, we will select the fixed motion index value (register bits, can set them to ‘h7f)!



Reference Register Map:

Register Name	Register Address	Register Description

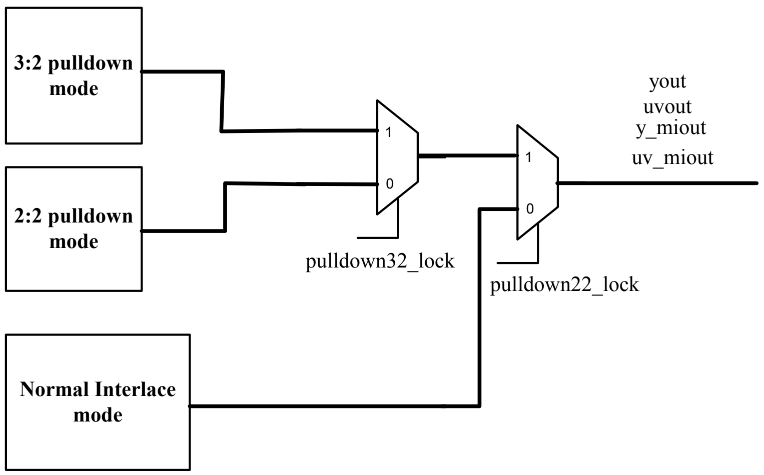
Madpt_mi_1bit_byps	Reg_S2_0c [4]	=1, Bypass feedback bit feature; =0, enable feedback bit feature.
Madpt_mi_1bit_frame2_en	Reg_S2_0c [5]	=1, enable frame2's feedback bit
Madpt_mi_threshold [6:0]	Reg_S2_0d [6:0]	Feedback bit compare threshold
Madpt_1bit_fixed_mi_value [6:0]	Reg_S2_0e [6:0]	Motion index fixed value when feedback it is 1

6.3.7 Motion Adaptive

Motion adaptive is mode auto adaptive for 3:2 pulldown, 2:2 pulldown, and normal interlace mode, and it output the weave output data. When you disable the auto mode adaptive **madpt_mo_adp_y_en** (Reg_S2_16 [4]) and **madpt_mo_adp_uv_en** (Reg_S2_16 [5]), it will look the source as normal interlace source.

Below is the diagram of motion adaptive:

Motion Adaptive



6.3.7.1 3:2 pulldown mode

For 3:2 pulldown mode, its field data is separated from film-frame data, so we only need re-merge the corresponding two fields into a frame, its motion index is zero. So we select the separated field as the 3:2 pulldown sequences from the current field, the last field, and the last two field data.

6.3.7.2 2:2 pulldown mode

For 2:2 pulldown mode, its field data is also separated from film-frame data. So we only need re-merge the corresponding tow fields into a frame. Its motion index is zero. So we select the separated field as the 2:2 pulldown sequences from the current field, the last field, and the last tow filed data.

6.3.7.3 Normal interlace mode

For normal interlace mode, its original source is interlace data, each field have motion, so we must merge the field databased on the motion index. We select the current field data and the last field data as the source and merge them based on motion index.

6.3.7.4 Weave out

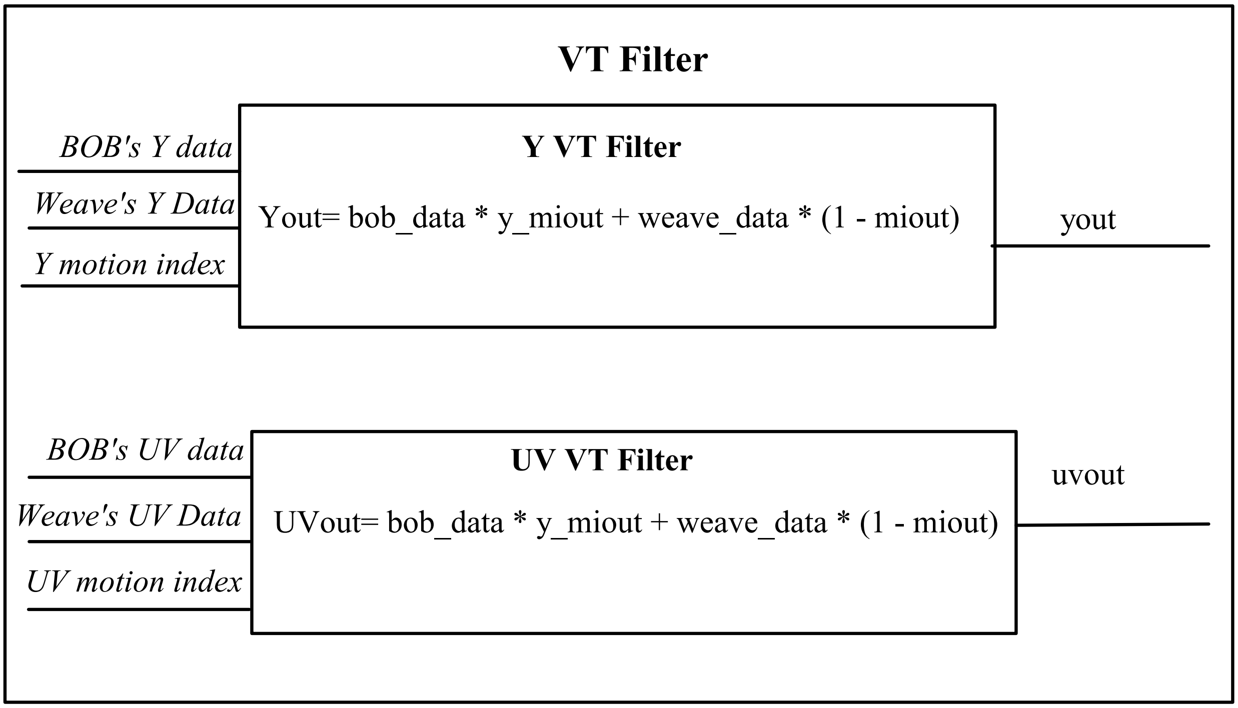
Weave out is used to merge the 2 field data as a frame data based on the line signal, when the line is 1, select the current field data, when line signal is 0, select the last field data.
Note: For 3:2 pull down and 2:2 pull down mode, it's current and last filed is select by pull down sequence.

6.3.8 VT Filter

For TrueView 5725, there are 2 VT filters, one is for Y, and the other is for UV. VT filter have 2 group inputs, Din0 is BOB's data input, which is the diagonal BOB output data. Din1 is Weave's data input, which is the motion adaptive output data.

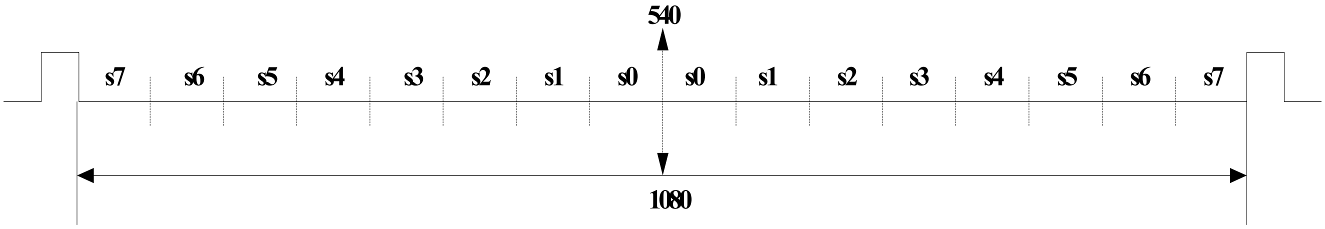
In interlace mode, set **madpt_vt_sel_prgv** (Reg_S2_16 [7]) to 0, do motion adaptive;
In progressive mode, set **madpt_vt_sel_prgv** (Reg_S2_16 [7]) to 1, keep original data.

Below is the VT filter diagram:



6.4 Vertical Scaling Down

5725 vertical scaling down is a non-linear scaling down because of change the aspect ratio between The HDTV and SDTV, The vertical lines will be divided into 16 periods, assume the source has 1080 lines, and then it will be divided 16 segments as the following figure. In fact, if we program all the segment scaling factors the same, the scaling down will be linear scaling down.



6.4.1.1 Vertical IIR Filter

There is vertical IIR filter is used before vertical scaling down, it is low pass filter in vertical domain, mainly for remove the high frequency in vertical, you can bypass them via setting register.

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_viir_byps	Reg_S2_26 [6]	=1, Bypass vertical IIR filter for scaling down When no scaling down, you can bypass it.
Madpt_viir_round_sel	Reg_S2_26 [7]	=1, (x0+x1)/2 output, =0, x0 output
Madpt_viir_coef [6:0]	Reg_S2_27 [6:0]	The vertical IIR coefficient.

6.4.1.2 Phase Adjustment

In 5725 vertical scaling down, phase adjustment is used. Phase's value is generated by DDA algorithm. And DDA algorithm also generate decimation's enable signal.

- Register bit ("**madpt_v_scale_rate***") is used to adjust vertical scaling down's rate. If you vertical scaling down from M lines to N lines, then the scaling rate is (M-N/N) * 4095
- Register bit ("**madpt_sel_phase_ini**") could be used to select scaling down DDA algorithm initial value.
- Register bit ("**madpt_y_vscale_byps**", "**madpt_uv_vsacle_byps**") could bypass Y and UV's phase adjustment when they set to one.

Reference Register Map:

Register Name	Register Address	Register Description
Madpt_y_vscale_byps	Reg_S2_02 [6]	=1, Bypass Y non-linear vertical scaling down
Madpt_uv_vscale_byps	Reg_S2_02 [7]	=1, Bypass UV non-linear vertical scaling down
Madpt_vscale_dec_factor [1:0]	Reg_S2_31 [1:0]	00, 1x 01, 1/2x 10: 1/4x 11:1/8x
Madpt_vscale_rate_low [3:0]	Reg_S2_28 [7:4]	Vertical non-linear scaling down low 4 bits
Madpt_vscale_rate_seg0 [7:0]	Reg_S2_29 [7:0]	Non-linear scaling down 1st segment rate
Madpt_vscale_rate_seg1 [7:0]	Reg_S2_2a [7:0]	Non-linear scaling down 2nd segment rate
Madpt_vscale_rate_seg2 [7:0]	Reg_S2_2b [7:0]	Non-linear scaling down 3rd segment rate
Madpt_vscale_rate_seg3 [7:0]	Reg_S2_2c [7:0]	Non-linear scaling down 4th segment rate
Madpt_vscale_rate_seg4 [7:0]	Reg_S2_2d [7:0]	Non-linear scaling down 5th segment rate
Madpt_vscale_rate_seg5 [7:0]	Reg_S2_2e [7:0]	Non-linear scaling down 6th segment rate
Madpt_vscale_rate_seg6 [7:0]	Reg_S2_2f [7:0]	Non-linear scaling down 7th segment rate
Madpt_vscale_rate_seg7 [7:0]	Reg_S2_30 [7:0]	Non-linear scaling down 8th segment rate

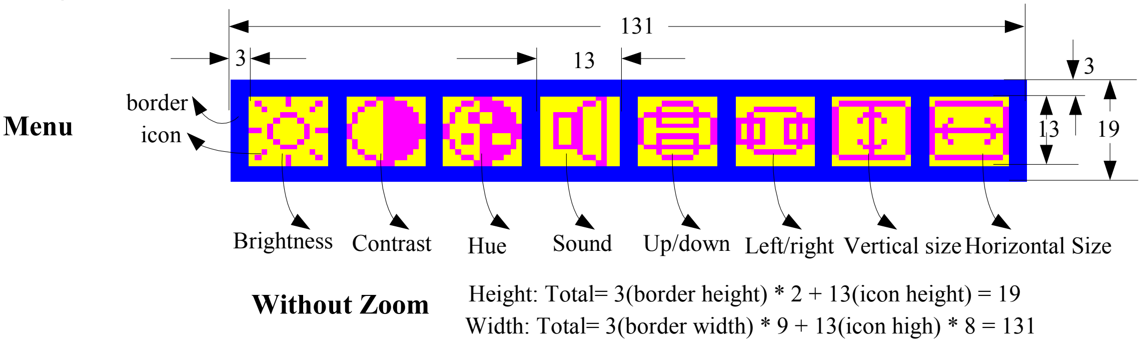
7 OSD Engine

TureView 5725 OSD engine is an 8-icons simple graphic engine. It is finished by hardware. It has the following features:

- Eight 13x13 hard-wired OSD ROM
- Horizontal programmable 1~8 zoom
- Vertical programmable 1~4 zoom
- Eight colors selectable
- Support RGB & YCbCr output
- OSD menu display in row or column style

7.1 OSD menu

OSD menu is used to display and select the icon you want to modify. The following Picture is copied from our simulation result:



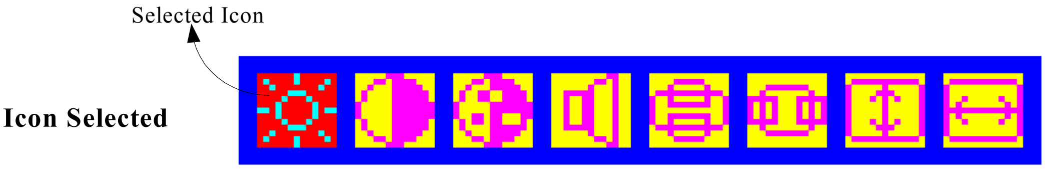
131
3
13
3
Menu
border
icon
13
19
Brightness
Contrast
Hue
Sound
Up/down
Left/right
Vertical size
Horizontal Size
Without Zoom
Height: Total= 3(border height) * 2 + 13(icon height) = 19
Width: Total= 3(border width) * 9 + 13(icon high) * 8 = 131

Just as what you saw, there are 8 icons display on the menu, brightness, contrast, hue, sound, up/down adjustment, left/right adjustment, horizontal size adjustment, and vertical size adjustment. We can't change and select the icons because they are hardwired in design, but we can select the color and zoom in horizontal or vertical.

When without zoom function, the border height and width is 3, the icon height and width is 13 * 13, so horizontal total is 131, vertical total is 19; the following is the register setting:

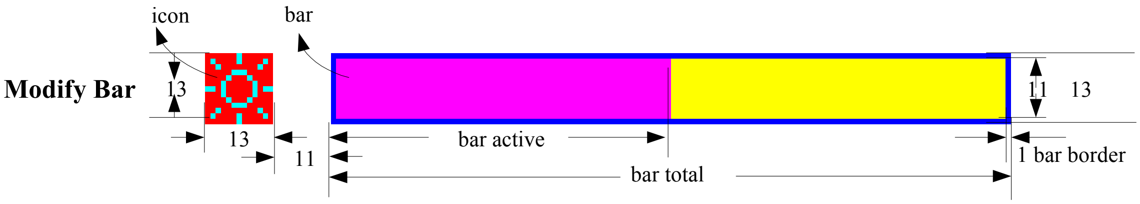
7.2 OSD ICON Selected:

When you want to modify some parameter of the system, firstly, you must select the corresponding icon, the selected icon will display with different color to distinguish it.



7.3 OSD Bar Modify

When we selected some icon, we want enter this icon's bar to modify its value, 5715 OSD engine provide 8 modify bar for you to select, the modify bar includes 2 parts, one is the icon, it tell you what parameter you will change, the other is the value bar, the bar's foreground color is used to the active value, the bar's background color is used for the total value.



Without Zoom

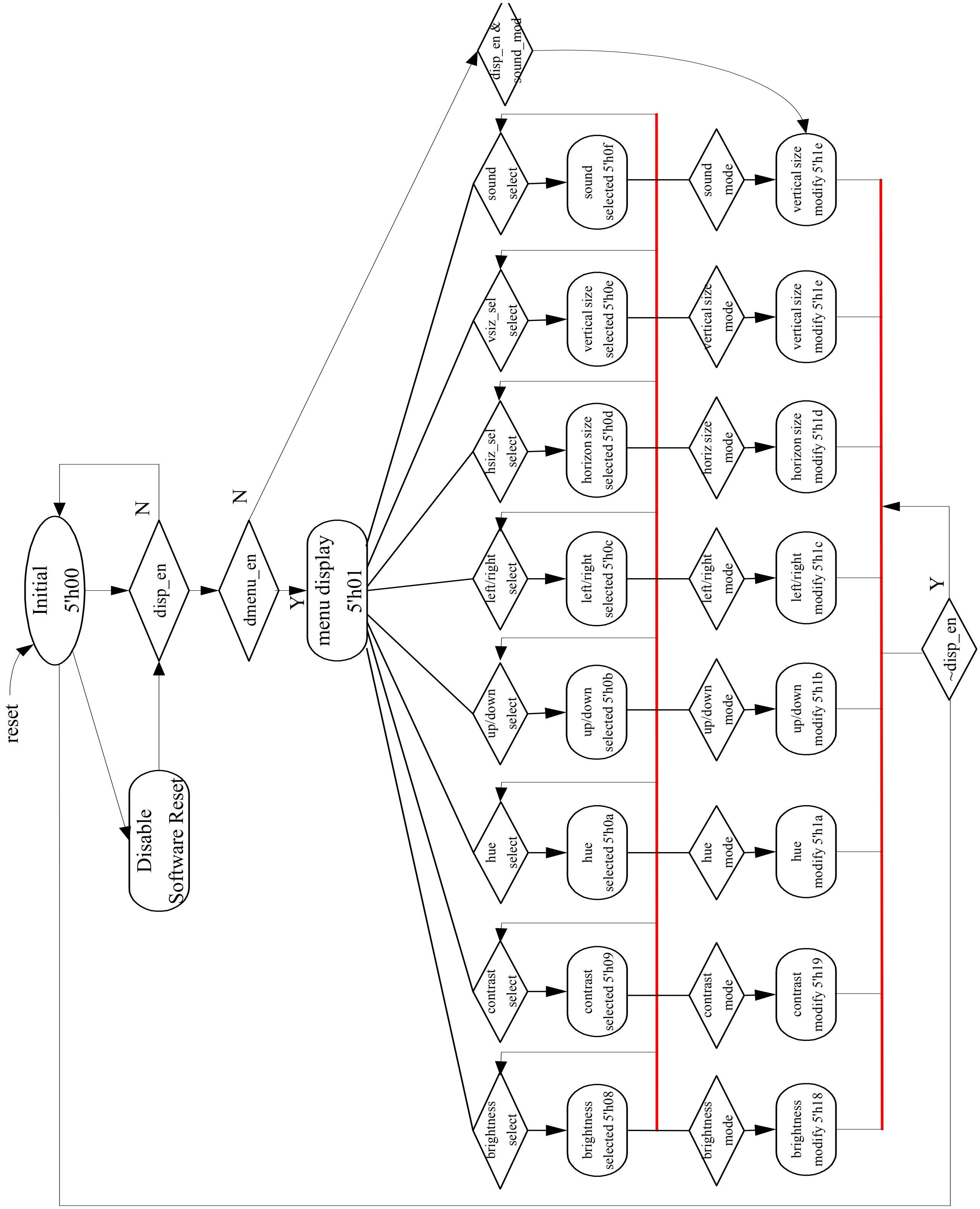
ICON: Weight: 13 Height: 13
Bar Height: Total= 1(bar border height) * 2 + 11(bar height) = 13
Bar Width: Total= 1(bar border width) * 2 + bar total(programmable)
Bar active : register programmable, 0=<bar active =< 255
Bar total : register programmable, 0=<bar total =< 255

7.4 OSD Engine Register Map

Register Name	Register Address	Register Description
OSD Soft Reset	Reg_S0_90 [0]	OSD Engine software reset 1, reset OSD engine
Horizontal Zoom	Reg_S0_90 [3:1]	3'b000: original size 3'b001: 2 times of original size 3'b010: 3 times of original size ………………………………… 3'b111: 8 times of original size
Vertical Zoom	Reg_S0_90 [5:4]	2'b00: original size 2'b01: 2 times of original size 2'b10: 3 times of original size 2'b11: 4 times of original size
OSD display enable	Reg_S0_90 [6]	OSD display enable 1, display OSD
OSD menu display enable	Reg_S0_90 [7]	OSD menu display enable 1, display OSD menu
OSD menu icon selection	Reg_S0_91 [3:0]	4'b0001: brightness icon selected 4'b0010: contrast icon selected 4'b0011: hue icon selected 4'b0100: sound icon selected 4'b1000: up/down icon selected 4'b1001: left/right icon selected 4'b1010: vertical size icon selected 4'b1011: horizontal size icon selected Others: reserved for future usage

Register Name	Register Address	Register Description
Bar modify selection	Reg_S0_91 [7:4]	4’b0001: brightness bar selected 4’b0010: contrast bar selected 4’b0011: hue bar selected 4’b0100: sound bar selected 4’b1000: up/down bar selected 4’b1001: left/right bar selected 4’b1010: vertical size bar selected 4’b1011: horizontal size bar selected Others: reserved for future usage
Menu/Bar foreground color	Reg_S0_92 [2:0]	3’b000: Black 3’b100: Red 3’b001: Blue 3’b101: Magenta 3’b010: Green 3’b110: Yellow 3’b011: Cyan 3’b111: White
Menu/Bar background color	Reg_S0_92 [5:3]	3’b000: Black 3’b100: Red 3’b001: Blue 3’b101: Magenta 3’b010: Green 3’b110: Yellow 3’b011: Cyan 3’b111: White
Menu/Bar border color	Reg_S0_93 [0], Reg_S0_92 [7:6]	3’b000: Black 3’b100: Red 3’b001: Blue 3’b101: Magenta 3’b010: Green 3’b110: Yellow 3’b011: Cyan 3’b111: White
Selected Icon foreground color	Reg_S0_93 [3:1]	3’b000: Black 3’b100: Red 3’b001: Blue 3’b101: Magenta 3’b010: Green 3’b110: Yellow 3’b011: Cyan 3’b111: White
Selected Icon background color	Reg_S0_93 [6:4]	3’b000: Black 3’b100: Red 3’b001: Blue 3’b101: Magenta 3’b010: Green 3’b110: Yellow 3’b011: Cyan 3’b111: White
Command status bit	Reg_S0_93 [7]	Command finish status bit, used by firmware, when we write OSD command, we set this bit to 0,when we complete OSD command, we set this bit to 1.
Mode select	Reg_S0_94 [0]	1, OSD menu display in row style 0, OSD menu display in column style
RGB/YCbCr output select	Reg_S0_94 [2]	1, OSD as YCbCr output 0, OSD as RGB output
H-Starting address	Reg_S0_95 [7:0]	Menu/Bar display horizontal stating address, the real address={value, 3’h0}
V-Starting address	Reg_S0_96 [7:0]	Menu/Bar display vertical stating address, the real address={value, 3’h0}
Bar total length	Reg_S0_97 [7:0]	Bar display total length
Bar active value	Reg_S0_98 [7:0]	Bar display active length

7.5 OSD Engine state machine



OSD Engine works as the following steps:

- After hardware reset or software reset (Reg60 [0]), OSD engine will enter the initial state.
- When initial state, you should disable OSD reset (hardware or software) and enable the OSD display (Reg60 [6]). Now you have 2 choices, one is enter the OSD menu via enabling OSD menu display (Reg60 [7]), the other is directly enter the sound modify via enabling sound modify selection (Reg61 [7:4] == 4'b0100).
- When in the OSD menu state, you can select the one icon from the 8-icons displayed on the OSD menu. There are 4 register bits you can program (Reg61 [3:0], detailed register description refer to OSD register map). When you select the icon, it will change its color to indicate its active state.
- When you select an icon, you have 2 choices, one is change to other icon via programming Reg61 [3:0] different value. The other is enter this icon's/bar value modification via programming Reg61 [7:4] the same value as Reg61 [3:0].
- When you modify the bar value, you can change the bar active and total length via programming Reg67 [7:0] and Reg 68 [7:0].
- When you complete the bar's modification, you should disable the OSD display and enable software reset to come back the initial state.

7.6 Example batch job for OSD Engine

```
REM About the OSD brightness icon selects and bar modification
I2CRW AE F0 00          REM Select the segment 0
I2CRW AE 90 00          REM Software Reset OSD Engine
I2CRW AE 91 01          REM Select Brightness icon
I2CRW AE 92 75          REM Set the icon foreground and background color
I2CRW AE 93 c6          REM Set the active icon/bar and border foreground and background color
I2CRW AE 94 01          REM Set OSD in row mode, RGB output
I2CRW AE 95 10          REM Set the OSD display horizontal position
I2CRW AE 96 10          REM Set the OSD display vertical position
I2CRW AE 90 c0          REM Disable software reset and enable OSD display

I2CRW AE 91 11          REM Select Brightness bar modify
I2CRW AE 92 75          REM Set the icon foreground and background color
I2CRW AE 97 80          REM Set the Bar total length
I2CRW AE 98 40          REM Set the Bar active length.

REM About the OSD contrast icon selects and bar modification
I2CRW AE 90 00          REM Software Reset OSD Engine
I2CRW AE 91 02          REM Select Brightness icon
I2CRW AE 92 75          REM Set the icon foreground and background color
I2CRW AE 93 c6          REM Set the active icon/bar and border foreground and background color
I2CRW AE 94 01          REM Set OSD in row mode, RGB output
I2CRW AE 95 10          REM Set the OSD display horizontal position
I2CRW AE 96 10          REM Set the OSD display vertical position
I2CRW AE 90 c0          REM Disable software reset and enable OSD display

I2CRW AE 91 22          REM Select Brightness bar modify
I2CRW AE 92 75          REM Set the icon foreground and background color
I2CRW AE 97 80          REM Set the Bar total length
I2CRW AE 98 40          REM Set the Bar active length.
```

8 Memory FIFO Top Level

5725 Memory FIFO Top Level provides external memory accessing interface. It includes Write FIFO, Read FIFO, Capture, Playback and Memory controller.

8.1 Write FIFO and Read FIFO

For TrueView 5725, it can support YUV de-interlace or Y de-interlace only.

For Y de-interlace only mode, Write FIFO receive Y data from diagonal BOB de-interlace and send it to memory controller, which will access external memory. Read FIFO will receive two frame Y data from memory controller and send it to motion adaptive de-interlace.

For YUV de-interlace mode, Write FIFO received Y/UV data from background noise reduction and send it to memory controller (when line ID is 1, write Y data, when line ID is 0, write UV data). Read FIFO will receive Y/UV data from memory controller and send it to motion background noise reduction (when line ID is 1, read the last 2 field Y/UV data; when line ID is 0, read the last field Y/UV data).

8.1.1 Address Control

I In 5725, write FIFO and Read FIFO use double-buffer access to make sure they could write one frame and read two frames data without *tearing* effect at the same time, fetch number is RFF will read pixel number of a line, offset is WFF/RFF from current line to next line address offset. Offset and fetch number is based on 2 double word, Assume horizontal active pixel number is 720:

- Rff_wff_offset = (720 / 8) * 2 = 180 = B4 (hex)……………..2 is for double buffer
- Rff_fetch_num = (720 / 8) = 180 = 5A (hex)

Write FIFO and read FIFO shared buffer A start address in register (“**rff_wff_sta_addr_a [20:0]**”).
Write FIFO and read FIFO shared buffer B start address in register (“**rff_wff_sta_addr_b [20:0]**”).
Write FIFO and read FIFO shared offset in register (“**rff_wff_offset [9:0]**”).
Write FIFO and read FIFO shared fetch number in register (“**rff_fetch_num [9:0]**”).

Reference Register Map:

Control Name	Address	Description
Rff_wff_sta_addr_a	Reg_S4_53 [4:0] Reg_S4_52 [7:0] Reg_S4_51 [7:0]	WFF/RFF buffer A starting address
Rff_wff_sta_addr_b	Reg_S4_56 [4:0] Reg_S4_55 [7:0] Reg_S4_54 [7:0]	WFF/RFF buffer B starting address
Rff_wff_offset	Reg_S4_58 [1:0] Reg_S4_57 [7:0]	WFF/RFF offset
Rff_fetch_num	Reg_S4_5a [1:0] Reg_S4_59 [7:0]	RFF fetch number
Wff_yuv_deinterlace	Reg_S4_4a [0]	=1, enable WFF YUV de-interlace =0, enable WFF Y only de-interlace
Rff_yuv_deinterlace	Reg_S4_50 [6]	=1, enable RFF YUV de-interlace =0, enable RFF Y only de-interlace

8.1.1.1 Y de-interlace mode

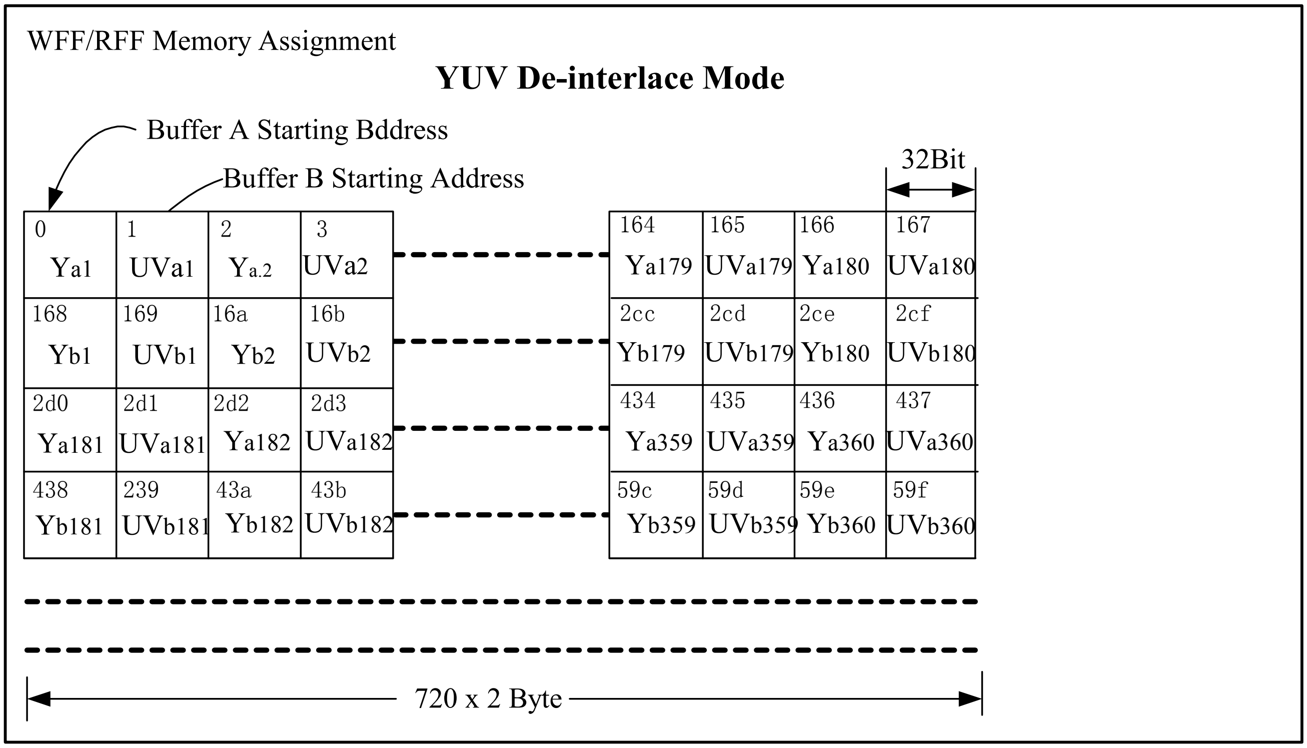
For Y de-interlace mode, Buffer A starting address is 0, Buffer B starting address is 1, WFF address will add 2 each step. For EVEN frame, Buffer A will receive Y data and store them at EVEN address (0, 2, 4, 6…); For ODD frame, Buffer B will receive Y data and store them at ODD address (1, 3, 5, 7………); RFF will capture progressive Y from Buffer A and Buffer B, that is to say, RFF will read 2 frames progressive Y data a time.

8.1.1.2 YUV de-interlace mode

The following example assumed Horizontal active is 720 pixels!

For YUV de-interlace mode, Buffer A starting address is 0, Buffer B starting address is 168(hex), WFF address will add 2 each step. Write FIFO receive interlace Y data and send them to Buffer A EVEN address when line signal is high, Write FIFO receive interlace UV data and send them to Buffer A ODD address when line address is low; when next frame, Write FIFO will receive interlace Y data and send them to buffer B EVEN address when line signal is high, Write FIFO will receive interlace UV data and send them to Buffer B ODD address when line signal is low. Read FIFO capture interlace Y and UV from Buffer A (last frame2) when line signal is high, and capture interlace Y and UV from Buffer B (last frame1) when line signal is low, that is to say, Read FIFO will read 2 frames interlace Y and UV data line by line.

For reduce bandwidth, write FIFO and read FIFO use a special storage in memory chip. Please read the diagram in below


WFF/RFF Memory Assignment
YUV De-interlace Mode
Buffer A Starting Bddress
Buffer B Starting Address
32Bit
720 x 2 Byte

0 Ya1	1 UVa1	2 Ya.2	3 UVa2	164 Ya179	165 UVa179	166 Ya180	167 UVa180
168 Yb1	169 UVb1	16a Yb2	16b UVb2	2cc Yb179	2cd UVb179	2ce Yb180	2cf UVb180
2d0 Ya181	2d1 UVa181	2d2 Ya182	2d3 UVa182	434 Ya359	435 UVa359	436 Ya360	437 UVa360
438 Yb181	239 UVb181	43a Yb182	43b UVb182	59c Yb359	59d UVb359	59e Yb360	59f UVb360

For example: if input source is NTSC, H-active is 720, V-active is 480

WFF_yuv_deinterlace = 1,
RFF_yuv_deinterlace = 1,
Buffer A start address =0,
Buffer B start address =168(hex),
Offset = (720/8)*2 = B4 (hex) (64-bit width based)
Fetch number = 720/8 = 5A (hex) (64-bit width based)

8.1.2 Safeguard Control

Write FIFO has safeguard function. When write FIFO output address is larger than safeguard, write FIFO request will stop. The safeguard enable control register bit is (“**wff_safe_guard**”). Because write FIFO is in double buffer mode, there are two safeguard for buffer A and buffer B (“**wff_safe_guard_a [20:0]**” and “**wff_safe_gurad_b [20:0]**”).

Reference Register Map:

Control Name	Address	Description
wff_safe_gurad_a	Reg_S4_46 [4:0] Reg_S4_45 [7:0] Reg_S4_44 [7:0]	WFF buffer A safe-guard address
Wff_safe_guard_b	Reg_S4_49 [4:0] Reg_S4_48 [7:0] Reg_S4_47 [7:0]	WFF buffer B safe-guard address
Wff_safe_guard	Reg_S4_42 [3]	=1, enable WFF safe-guard function; =0, disable WFF safe-guard function.

8.2 Capture and Playback

Capture is to write progressive scan generated by De-interlace processor to external memory. Playback will read interlace/progressive data for displaying or cross-color noise reduction.

8.2.1 Access modes

In 5725, capture and playback will these access modes: bypass mode, single buffer, double buffer, interlace mode, cross-color noise reduction mode.

8.2.1.1 Bypass mode

For bypass mode, it will bypass capture/playback, de-interlace will send the progressive YUV data to vds_proc directly without write into memory! . So this mode only is for testing, the display clock, line rate and frame rate must sync lock with the input clock, line rate and frame rate.

8.2.1.2 Single buffer mode

For single buffer mode, need buffer A only. It will cause tearing in display, unless sync lock mode is enabled. So this mode only is for testing, the display clock, line rate and frame rate must sync lock with the input clock, line rate and frame rate.

8.2.1.3 Double buffer mode

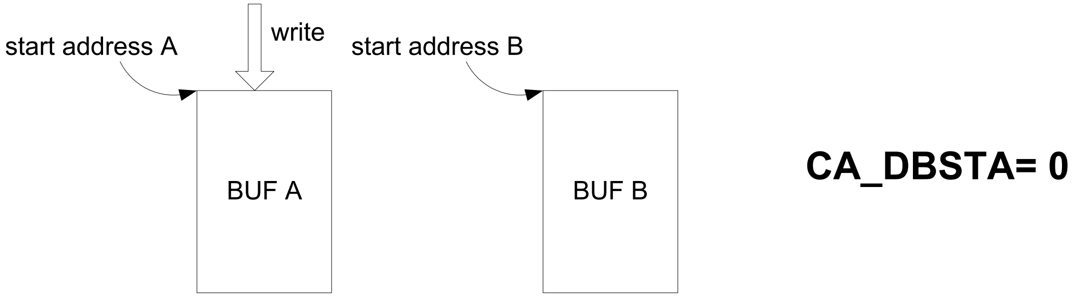
In this mode, there will be two starting address for capture and playback in frame buffer. So when each vertical loading address, start address will do shifting operation.

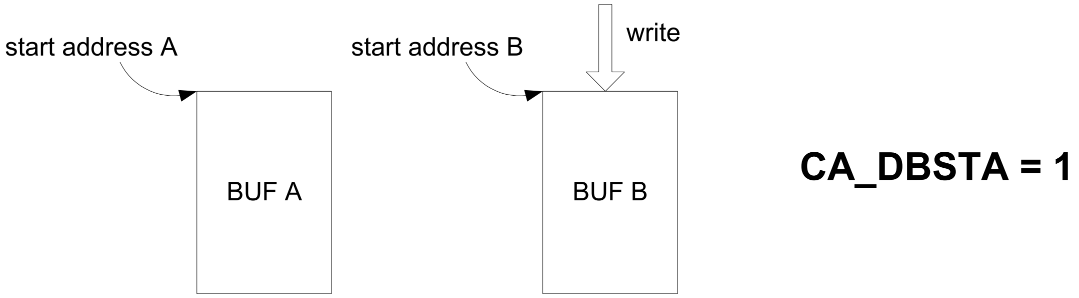
In 5725 normal working mode, capture frequency will be equal to or slower than playback frequency. So when each vertical loading address, capture will do shifting (A→ B → A → B …); while playback will do judgment for buffer status. That is to say, when buffer status indicator show capture write in buffer A, then playback reading will be jump from A to B or still stay in buffer B. In this way, we can insure capture and playback engine working in different frame buffer, avoiding tearing phenomenon.

When input source frame rate is bigger than the output frame rate, that is to say, capture frequency will be faster than playback frequency. So when each vertical loading address, playback will do shifting (A→ B → A →B ……); while capture will adjust the buffer status following the playback. That is to say, when buffer status indicator show playback read in buffer A, then capture will write in buffer B, or jump from A to B.

A. Input frame rate <= output frame rate (set pb_up_dow_rbuf_sel (Reg_S4_2e [1]) = 0)

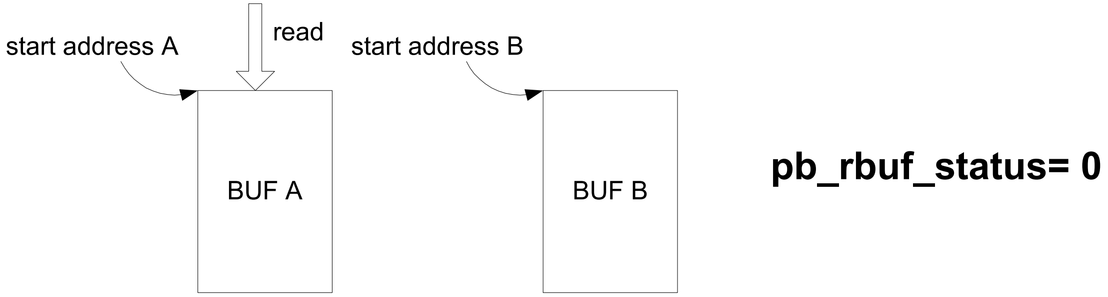
CA_DBSTA is the indicator for buffer status and tell that playback which buffer capture is used.

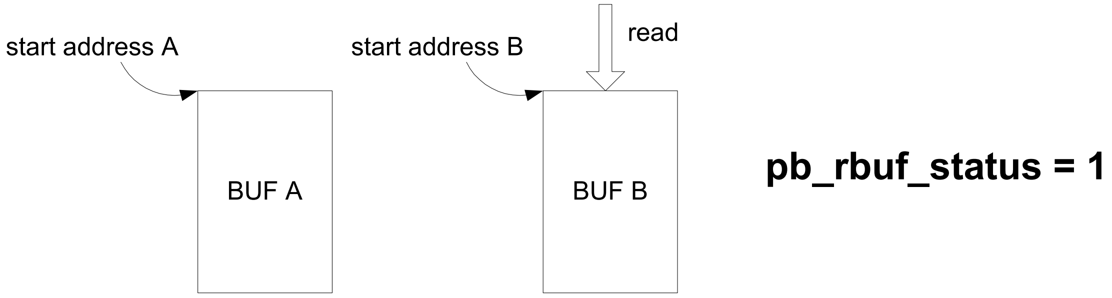




B. Input frame rate > output frame rate (set **pb_up_dow_rbuf_sel (Reg_S4_2e [1]) = 1)**

PB_RBUF_STATUS is the indicator for playback buffer status and tell the capture which buffer can write.





8.2.1.4 Interlace mode

For this mode is used for interlace output, 100i, 120i, 60i, 75i, you'd better use double buffer. It will fetch the interlace data from memory and send to vds_proc. We should enable double buffer to avoid tearing.

Set **pb_db_field_en** (Reg_S4_2b [4]) to 1, double field display mode enabled.
Set **pb_db_buffer_en** (Reg_S4_2b [5]) to 1, playback double buffer enable.
Set **cap_double_buffer** (Reg_S4_21 [3]) to 1, capture double buffer enable.

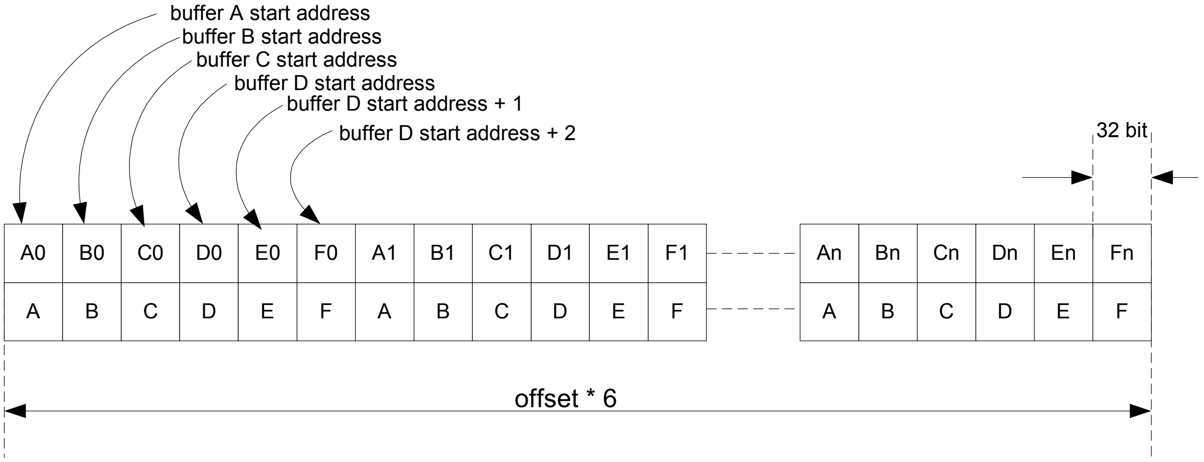
8.2.1.5 Cross-color noise reduction mode

For cross-color noise reduction mode includes NTSC mode and PAL mode, playback will fetch two frames data from memory at same time; and send the frame data to vds_proc do noise reduction.

1. NTSC: (set RegBF [3:0] noise reduction command to 4'hD)
   60Hz to 60Hz (memory storage 4-frame data)
   60Hz to 120iHz (memory storage 4-frame data)
2. PAL: (set RegBF [3:0] noise reduction command to 4'h9)
   50Hz to 50hz (memory storage 6-frame data)
   50Hz to 60Hz (memory storage 6-frame data)
   50Hz to 100iHz(memory storage 6-frame data)

Capture and playback share noise reduction command register (“**pb_cap_noise_cmd [3:0]”**).
Capture and playback share buffer A start address in register (“**pb_cap_buf_sta_addr_a [20:0]**”).
Capture and playback share buffer B start address in register (“**pb_cap_buf_sta_addr_b [20:0]**”).
Capture and playback share buffer C start address in register (“**pb_cap_buf_sta_addr_c [20:0]**”).
Capture and playback share buffer D start address in register (“**pb_cap_buf_sta_addr_d [20:0]**”).
Capture and playback shared offset in register (“**pb_cap_offset [9:0]**”).
Capture and playback shared fetch number register **(“pb_fetch_num [9:0]”**)

In PAL mode, Buffer E, buffer F's start address will be setting automatically. Please read the picture in below.



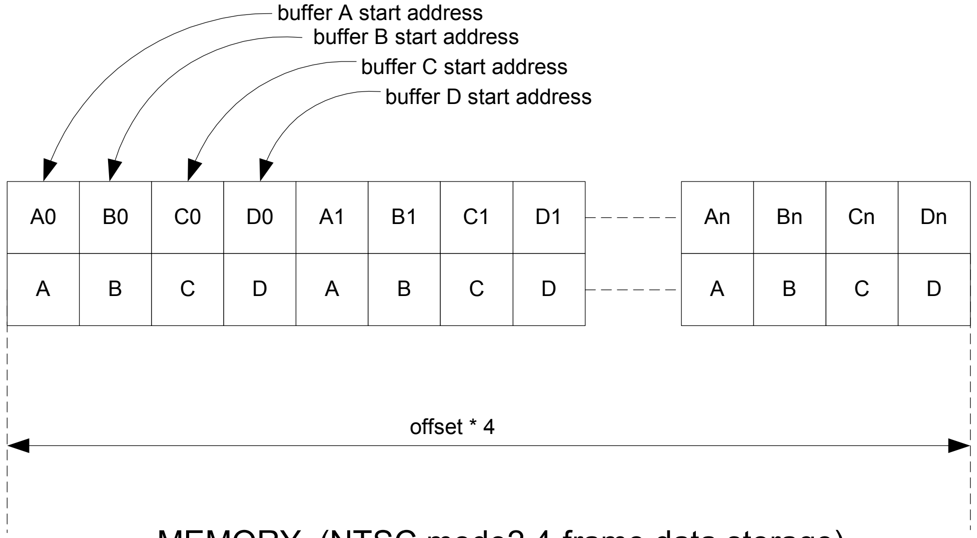
A0	B0	C0	D0	E0	F0	A1	B1	C1	D1	E1	F1	An	Bn	Cn	Dn	En	Fn
A	B	C	D	E	F	A	B	C	D	E	F	A	B	C	D	E	F

MEMORY (PAL mode2 6-frame data storage)

Programming example: input PAL source, H-active 720, V-active 576
Buffer A start address: 0
Buffer B start address: 1
Buffer C start address: 2
Buffer D start address: 3

Offset: B4(hex) (720/2/2 = 180)
Fetch number: B4(hex) (720/2/2 = 180)

In NTSC mode, Please read the picture in below:



MEMORY (NTSC mode2 4-frame data storage)

Programming Example: Input source NTSC, H-active 720, V-active 480
Buffer A start address: 0
Buffer B start address: 1
Buffer C start address: 2
Buffer D start address: 3
Offset: B4(hex) (720/2/2=180)
Fetch number: B4(hex) (720/2/2=180)

Capture and playback access memory's mode control is by register bits **cap_double_buffer, pb_db_field_en, pb_db_buffer_en, pb_cap_noise_cmd.** Please see blew for detail setting.

	Single Buffer	**Double Buffer**	**Interlace Mode**	**NTSC Noise Reduction**	**PAL Noise Reduction**
Reg_S4_21 [3]	0	1	1	0	0
Reg_S4_2b [5]	0	1	1	0	0
Reg_S4_30 [3:0]	0000	0000	0	1101	1001
Reg_S4_2b [4]	0	0	1	0	0

Reference Register Map:

Control Name	Address	Description
Capture_enable	Reg_S4_21 [0]	=1, enable capture =0, disable capture
Cap_double_buffer	Reg_S4_21 [3]	=1, enable capture double buffer function =0, disable capture double buffer function
Pb_enable	Reg_S4_2b [7]	=1, enable playback =0, disable playback
Pb_byps	Reg_S4_2b [3]	=1, bypass cap/pb, send de-interlace data to vds_proc =0, normal working mode
Pb_db_field_en	Reg_S4_2b [4]	=1, interlace mode output enable =0, progressive mode output enable
Pb_db_buffer_en	Reg_S4_2b [5]	=1, enable playback double buffer function =0, disable playback double buffer function
Pb_up_dow_rbuf_sel	Reg_S4_2e [1]	=1, select frame rate up-down mode. =0, select frame rate down-up mode.
Pb_cap_noise_cmd	Reg_S4_30 [3:0]	Select 3D noise reduction mode
Pb_cap_offset	Reg_S4_38 [1:0] Reg_S4_37 [7:0]	Capture/playback offset
Pb_fetch_num	Reg_S4_3a [1:0] Reg_S4_39 [7:0]	Capture/playback fetch number
Pb_cap_buf_sta_addr_a	Reg_S4_33 [4:0] Reg_S4_32 [7:0] Reg_S4_31 [7:0]	Capture/playback buffer A staring address
Pb_cap_buf_sta_addr_b	Reg_S4_36 [4:0] Reg_S4_35 [7:0] Reg_S4_34 [7:0]	Capture/playback buffer B staring address
Pb_cap_buf_sta_addr_c	Reg_S4_3d [4:0] Reg_S4_3c [7:0] Reg_S4_3b [7:0]	Capture/playback buffer C staring address
Pb_cap_buf_sta_addr_d	Reg_S4_40 [4:0] Reg_S4_3f [7:0] Reg_S4_3e [7:0]	Capture/playback buffer D staring address

8.2.2 Safegurad Control

Capture has safeguard function. When capture output address is larger than safeguard, capture's write request will stop. Enable safeguard function bit is register ("**cap_safe_guard_en**"). In single buffer mode, buffer A safeguard is ("**cap_safe_guard_a**"). In double buffer mode, buffer B safeguard is ("**cap_safe_guard_b)**"). In 3D noise reduction mode, Buffer A and buffer B safeguard ("**cap_safe_guard_a**") could be used. So it is treated as the last buffer's safeguard.

Reference Register Map:

Control Name	Address	Description
Cap_safe_gurad_a	Reg_S4_26 [4:0] Reg_S4_25 [7:0] Reg_S4_24 [7:0]	Capture buffer A safe-guard address
Cap_safe_guard_b	Reg_S4_29 [4:0] Reg_S4_28 [7:0] Reg_S4_27 [7:0]	Capture buffer B safe-guard address
Cap_safe_guard_en	Reg_S4_21 [5]	=1, enable capture safe-guard function; =0, disable capture safe-guard function.

8.2.3 Rate Conversion

Playback read memory data according timing generated by video processor. So the output video data's resolution could be changed through modifying video processor's output timing. In 5725, output's vertical refresh can equal or larger or less than input source vertical refresh. In order to avoid *tearing*, double buffer is used whatever output vertical's refresh is.

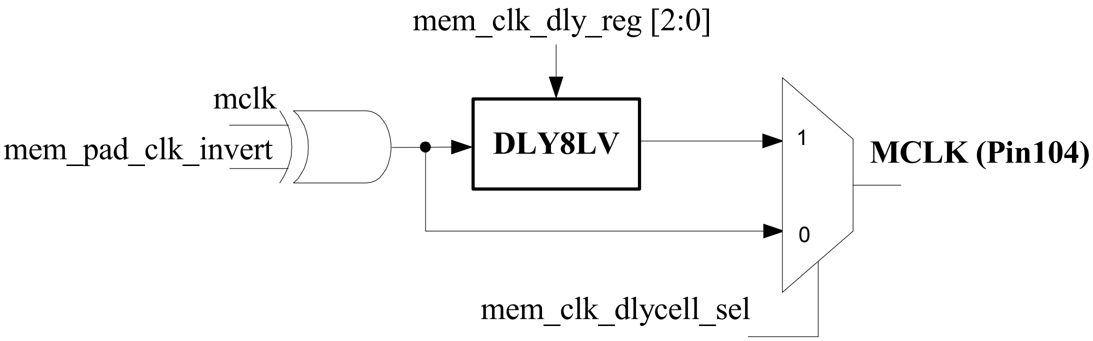
8.3 Memory Controller

Memory Controller accepts write/read FIFO, capture/playback's read/write request and access external SDRAM in position. In 5725, 2M/4M/8M size external SDRAM is support. And the external SDRAM could be 16bit/32bit width.

8.3.1 External Memory Interface Timing Adjustment

8.3.1.1 Memory Clock output

5725 memory controller provides external SDRAM clock. Below is the data path to adjust output memory clock delay:

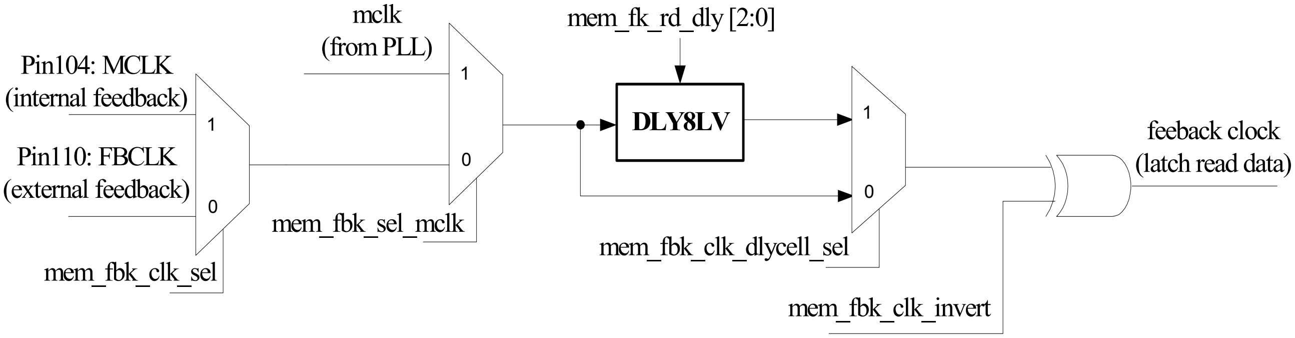


Reference Register Map:

Control Name	Address	Description
Mem_pad_clk_invert	Reg_S4_13 [0]	=1, invert MCLK send to PAD =0, not invert MCLK
Mem_clk_dlycell_sel	Reg_S4_12 [1]	=0, bypass delay cell =1, select DLY8V cell
Mem_clk_dly_reg	Reg_S4_1b [6:4]	MCLK delay control with DLY8LV • 000: delay 0.00 ns 001: delay 0.25 ns • 010: delay 0.50 ns 011: delay 0.75 ns • 100: delay 1.00 ns 101: delay 1.50 ns • 110: delay 2.00 ns 111: delay 3.00 ns

8.3.1.2 Feedback Clock input

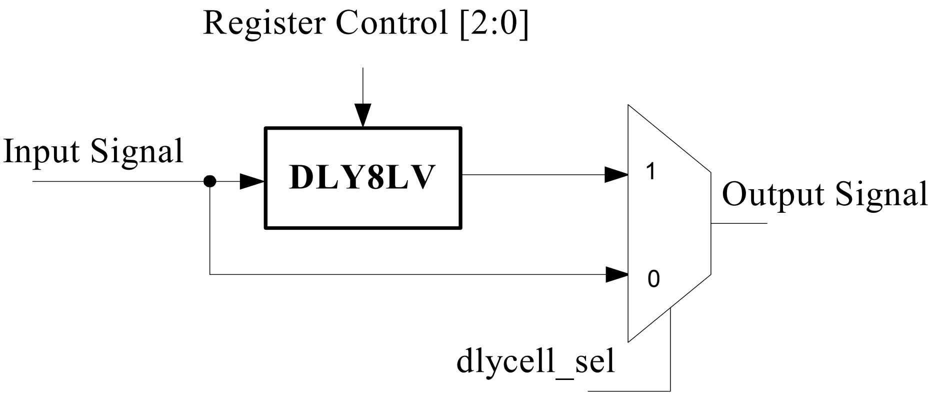
5725 memory controller provides feedback clock. Below is the data path to adjust feedback memory clock delay:



Reference Register Map:

Control Name	Address	Description
Mem_fbk_clk_sel	Reg_S4_11 [0]	=0, feedback clock from external PAD. =1, feedback clock from internal PAD.
Mem_fbk_sel_mclk	Reg_S4_11 [1]	=0, feedback clock from PAD. =1, feedback clock from PLL.
Mem_fbk_clk_dlycell_sel	Reg_S4_12 [2]	=0, bypass delay cell =1, select DLY8LV cell
Mem_fk_rd_dly	Reg_S4_04 [2:0]	Feedback clock delay control with DLY8LV • 000: delay 0.00 ns 001: delay 0.25 ns • 010: delay 0.50 ns 011: delay 0.75 ns • 100: delay 1.00 ns 101: delay 1.50 ns • 110: delay 2.00 ns 111: delay 3.00 ns
Mem_fbk_clk_invert	Reg_S4_13 [2]	=0, feedback clock not invert =1, feedback clock invert

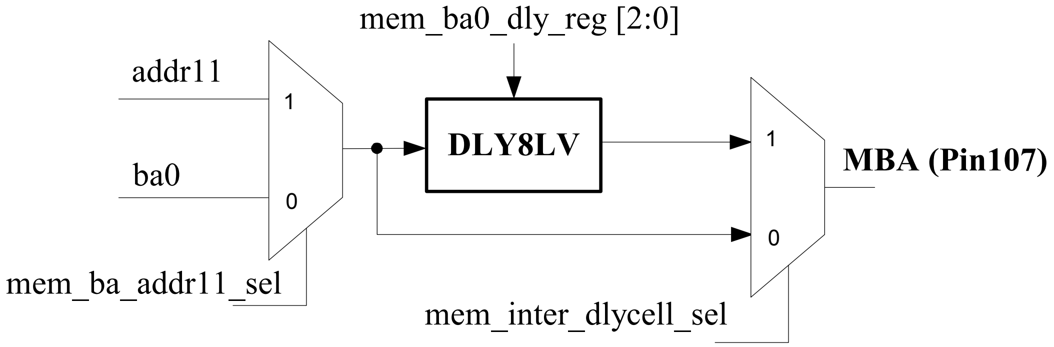
8.3.1.3 DQM, RAS, CAS, WE, address, data output control



Reference Table:

Pin Name	Pin Number	Register Control	dlycell_sel
DQM [1:0]	130, 98	Mem_dqm_dly_reg	Mem_inter_dlycell_sel Reg_S4_12 [0]
		Reg_S4_1a [6:4]	
RAS#	106	Mem_ras_dly_reg	
		Reg_S4_19 [2:0]	
CAS#	100	Mem_cas_dly_reg	
		Reg_S4_19 [6:4]	
WE#	99	Mem_we_dly_reg	
		Reg_S4_1a [2:0]	
MA [10:0]	116, 111, 114, 115, 117, 119, 127, 129, 128, 120, 118	Mem_adr_dly_reg	
		Reg_S4_1b [2:0]	
MA [31:0]	132, 134, 138, 140, 144, 146, 150, 152, 151, 149, 145, 141, 139, 137, 133, 131, 79, 81, 85, 87, 89, 95, 94, 97, 96, 91, 90, 88, 86, 84, 80, 78	Mem_data_dly_reg	
		Reg_S4_18 [2:0]	

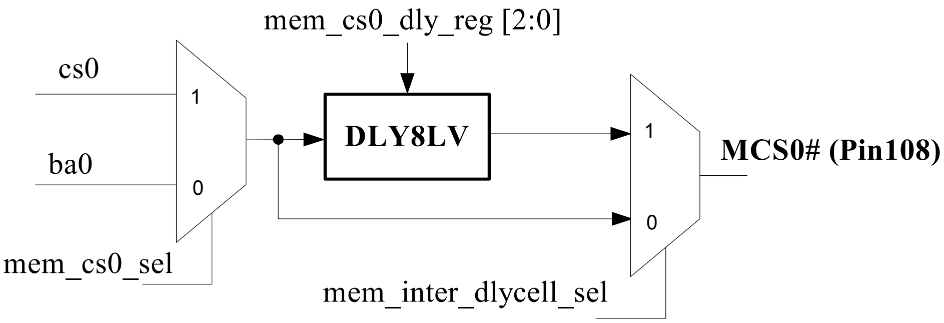
8.3.1.4 MBA output control



Reference Register Map:

Control Name	Address	Description
Mem_ba_addr11_sel	Reg_S4_0d [4]	=0, output banking select to PAD MBA =1, output address 11 to PAD MBA
Mem_inter_dlycell_sel	Reg_S4_12 [0]	=0, Bypass delay cell. =1, Select DLY8LV as delay cell.
Mem_ba0_dly_reg	Reg_S4_1d [2:0]	BA0 delay control with DLY8LV • 000: delay 0.00 ns 001: delay 0.25 ns • 010: delay 0.50 ns 011: delay 0.75 ns • 100: delay 1.00 ns 101: delay 1.50 ns • 110: delay 2.00 ns 111: delay 3.00 ns

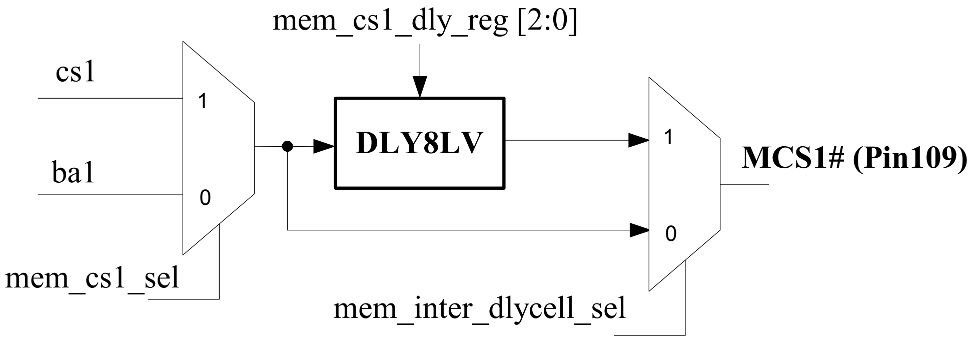
8.3.1.5 MCS0# output control



Reference Register Map:

Control Name	Address	Description
Mem_cs0_ba0_sel	Reg_S4_0d [5]	=0, output banking select0 to PAD MCS0# =1, output chip select0 to PAD MCS0#
Mem_inter_dlycell_sel	Reg_S4_12 [0]	=0, Bypass delay cell. =1, Select DLY8LV as delay cell.
Mem_cs0_dly_reg	Reg_S4_1c [2:0]	MCS0# delay control with DLY8LV • 000: delay 0.00 ns 001: delay 0.25 ns • 010: delay 0.50 ns 011: delay 0.75 ns • 100: delay 1.00 ns 101: delay 1.50 ns • 110: delay 2.00 ns 111: delay 3.00 ns

8.3.1.6 MCS1# output control



Reference Register Map:

Control Name	Address	Description
Mem_cs1_ba1_sel	Reg_S4_0d [5]	=0, output banking select1 to PAD MCS1# =1, output chip select1 to PAD MCS1#
Mem_inter_dlycell_sel	Reg_S4_12 [0]	=0, Bypass delay cell. =1, Select DLY8LV as delay cell.
Mem_cs1_dly_reg	Reg_S4_1c [6:4]	MCS1# delay control with DLY8LV • 000: delay 0.00 ns 001: delay 0.25 ns • 010: delay 0.50 ns 011: delay 0.75 ns • 100: delay 1.00 ns 101: delay 1.50 ns • 110: delay 2.00 ns 111: delay 3.00 ns

8.3.2 External Memory Configuration

8.3.2.1 External Memory Pin Connection

5725 could support 16/32 bit width 2/4/8M external SDRAM. For different size SDRAM, the connection in board is different. And there are some setting is different in memory controller. Please read below for pin connection and memory register setting:

8.3.2.2 External Memory Initialization

According SDRAM specification, the mode register is used to define the specific of operation of the SDRAM. (Please read Micron's SDRAM specification for detail description of SDRAM initialization.) Here is 5725's external memory initialization:

Burst Length:	1
Burst Type:	Sequential
CAS Latency:	Programmable
Operating Mode:	Standard operation
Write Burst Mode:	Programmed Burst Length

8.3.3 Memory golden settings

8.3.3.1 8M memory 129.6MHz memory clock case

ADDR.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Value	87	30	00	00	30	11	42	30	01	99	11	7f	00	74	00	06
ADDR.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Value	00	92	03	01	16	05	00	00	00	00	00	50	00	00	00	00

8.3.3.2 8M memory 162 MHz memory clock case

ADDR.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Value	87	30	00	00	30	11	42	30	01	99	11	7f	00	74	00	06
ADDR.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Value	00	92	01	01	16	05	00	00	00	00	00	00	00	00	00	00

8.3.3.3 8M memory 108 MHz memory clock case

ADDR.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Value	87	30	00	00	36	11	42	30	01	99	11	7f	00	74	00	06
ADDR.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Value	00	92	07	05	16	05	00	00	00	00	00	50	00	00	00	00

8.3.3.4 2M memory 129.6 MHz memory clock case

ADDR.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Value	87	30	00	00	30	11	42	30	01	94	11	7f	00	04	00	06
ADDR.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Value	00	92	03	01	16	05	00	00	00	00	00	50	00	00	00	00

8.3.3.5 2M memory 162 MHz memory clock case

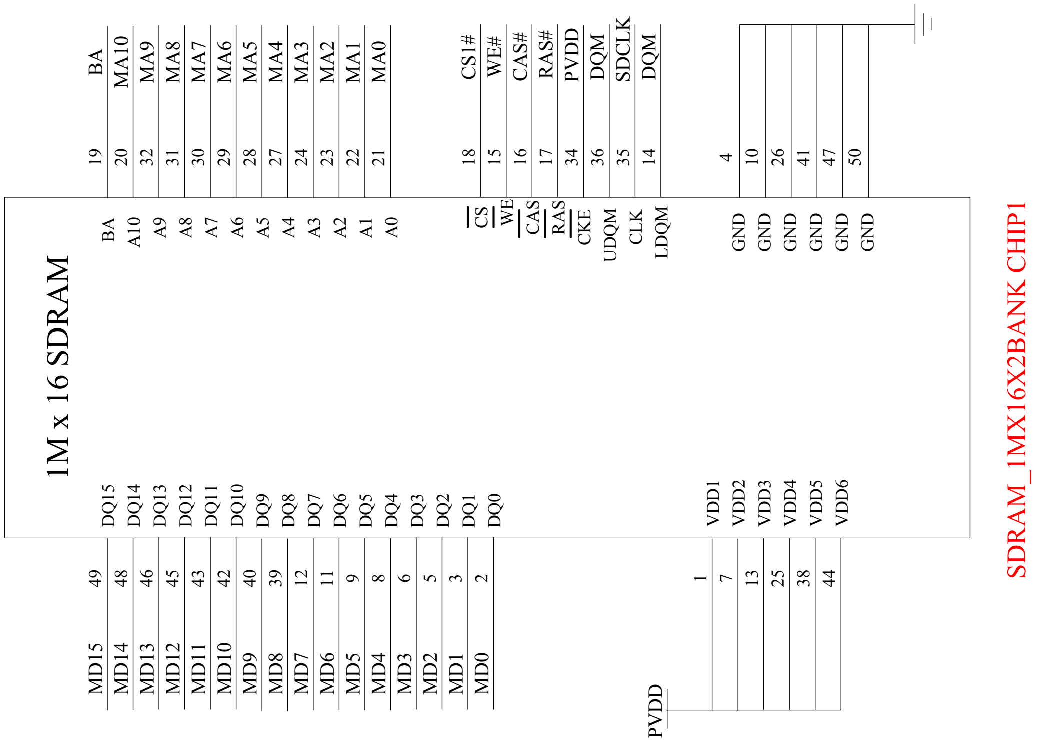
ADDR.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Value	87	30	00	00	30	11	42	30	01	94	11	7f	00	04	00	06
ADDR.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Value	00	92	01	01	16	05	00	00	00	00	00	00	00	00	00	00

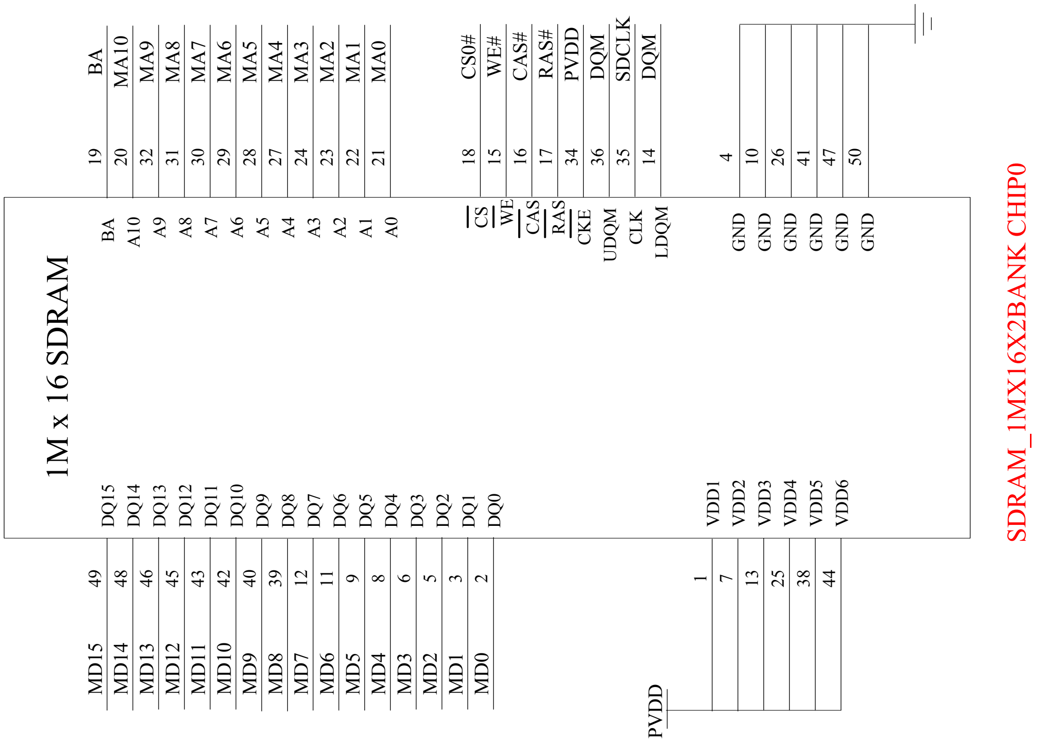
8.3.3.6 2M memory 108 MHz memory clock case

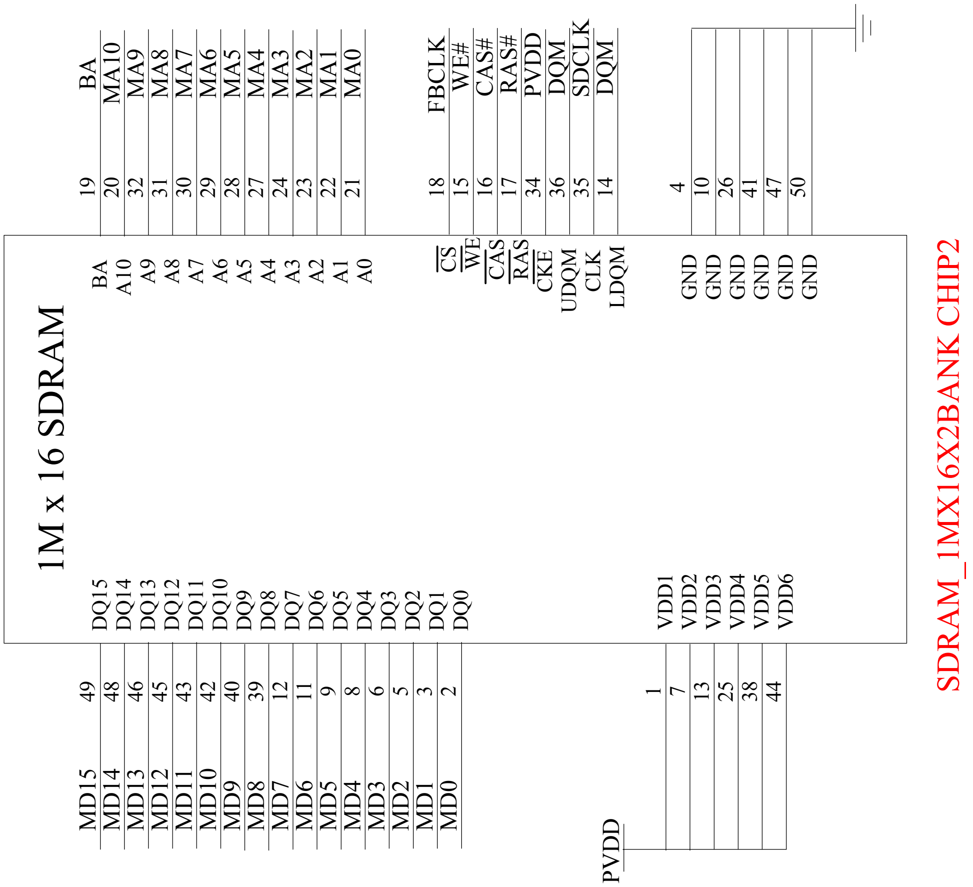
ADDR.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Value	87	30	00	00	36	11	42	30	01	94	11	7f	00	04	00	06
ADDR.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**1A**	**1B**	**1C**	**1D**	**1E**	**1F**
Value	00	92	07	05	16	05	00	00	00	00	00	50	00	00	00	00

8.4 Board Memory Connection:

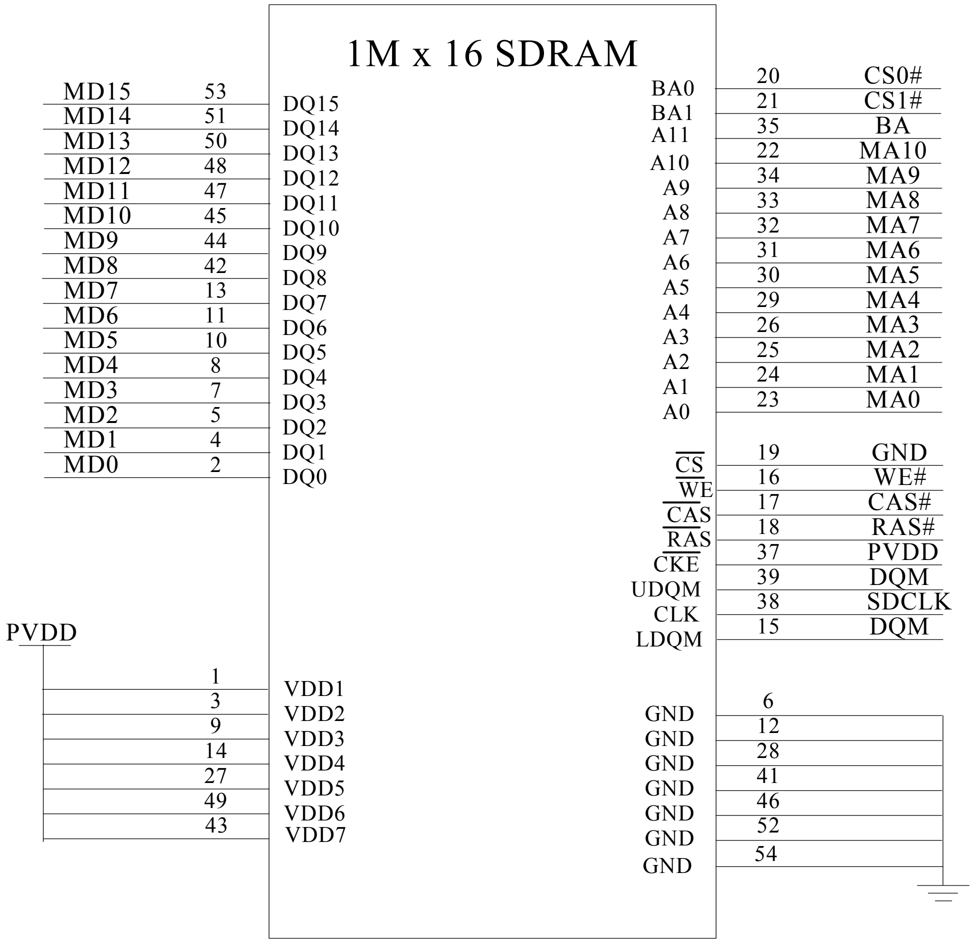
8.4.1 USE THREE 1MX16X2BANK MEMORY





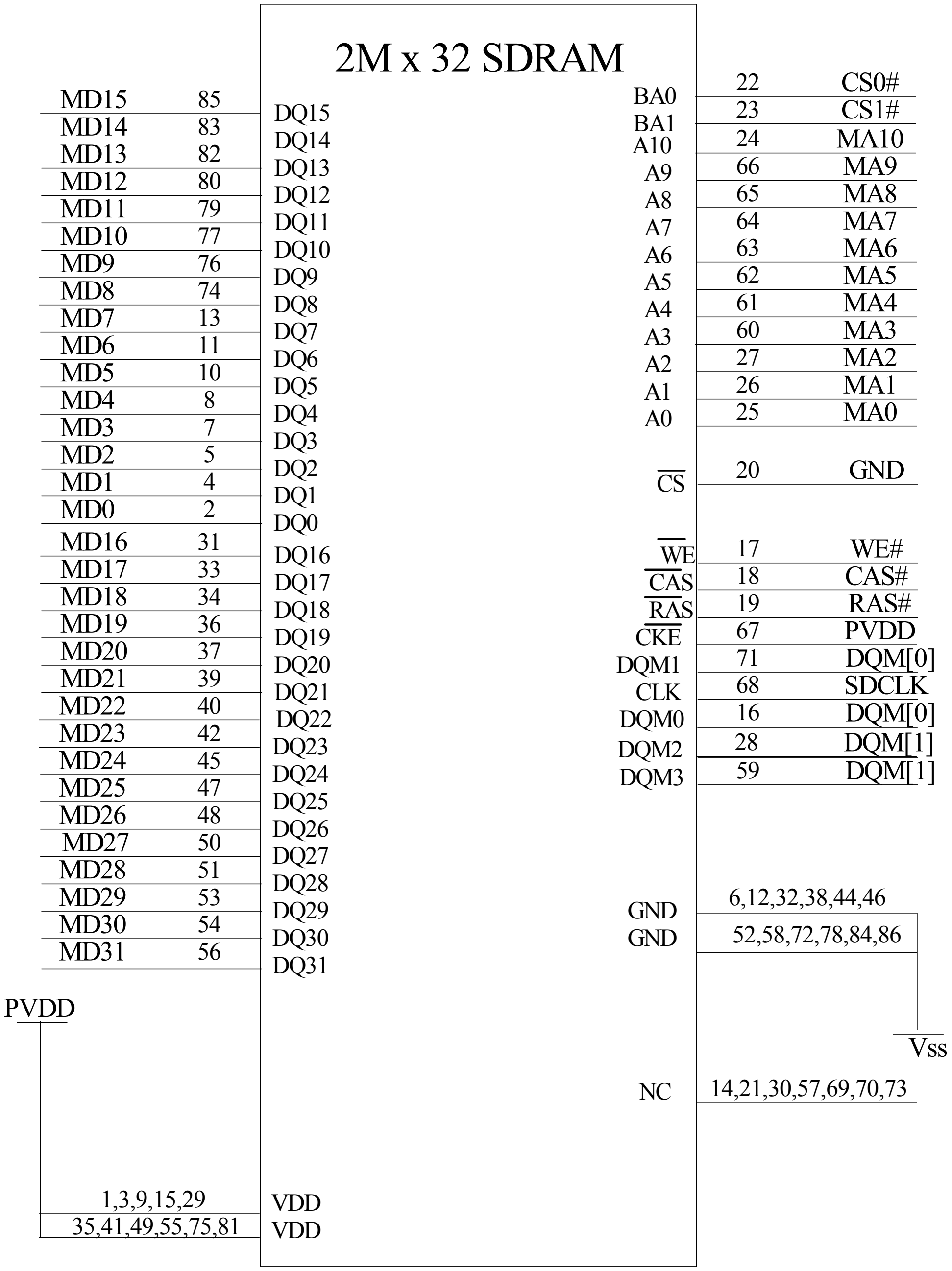

1M x 16 SDRAM
SDRAM_1MX16X2BANK CHIP2
BA 19 BA
MA10 20 A10
MA9 32 A9
MA8 31 A8
MA7 30 A7
MA6 29 A6
MA5 28 A5
MA4 27 A4
MA3 24 A3
MA2 23 A2
MA1 22 A1
MA0 21 A0
FBCLK 18 CS
WE# 15 WE
CAS# 16 CAS
RAS# 17 RAS
PVDD 34 CKE
DQM 36 UDQM
SDCLK 35 CLK
DQM 14 LDQM
4 GND
10 GND
26 GND
41 GND
47 GND
50 GND
MD15 49 DQ15
MD14 48 DQ14
MD13 46 DQ13
MD12 45 DQ12
MD11 43 DQ11
MD10 42 DQ10
MD9 40 DQ9
MD8 39 DQ8
MD7 12 DQ7
MD6 11 DQ6
MD5 9 DQ5
MD4 8 DQ4
MD3 6 DQ3
MD2 5 DQ2
MD1 3 DQ1
MD0 2 DQ0
PVDD
1 VDD1
7 VDD2
13 VDD3
25 VDD4
38 VDD5
44 VDD6

8.4.2 USE ONE 1MX16X4BANK MEMORY

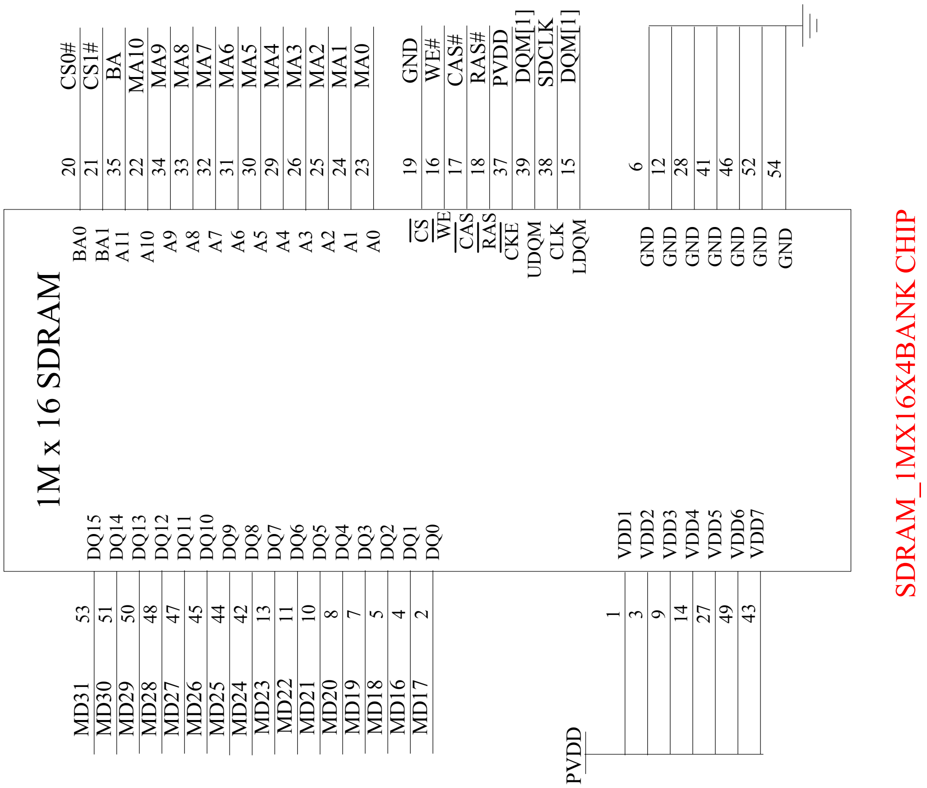


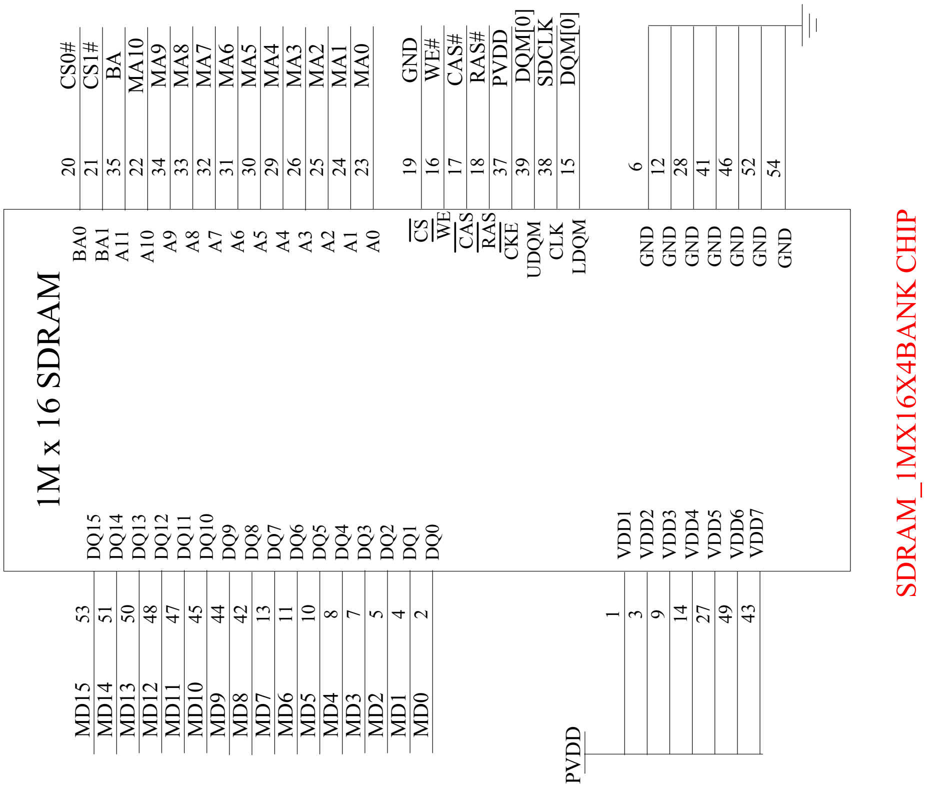
SDRAM_1MX16X4BANK CHIP

8.4.3 USE ONE 512KX32X4BANK MEMORY



8.4.4 USE TWO 1MX16X4BANK MEMORY





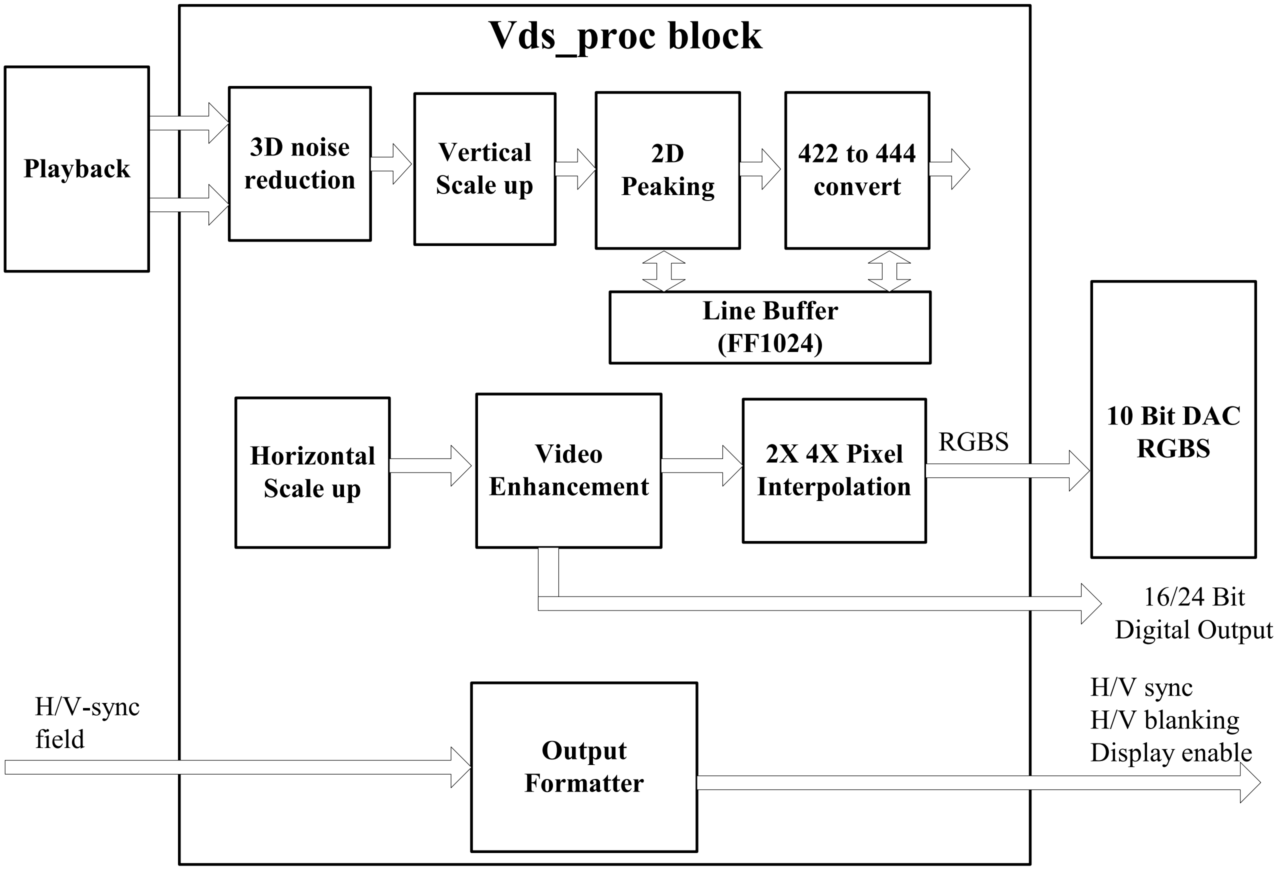
9 Video Processing

TrueView 5725 Video processing (video display processing) will provide:

---Output formatter
---Transient improvement
---Black/white level expansion
---2D Peaking
---Hue Control
---Skin tone correction
---Color enhancement
---Non-linear saturation
---Dynamic range expansion
---Blanking and sync insertion
---SVM generation
---3D noise reduction
---16/24 bit digital output

Note: Pin 77 HALF could open half function in vds_proc. When half = 1, RGB/YUV value will be divided by two, when half =0, RGB/YUV value will keep original.

Below is vds_proc's data path:



9.1 Output Formatter

Vds_proc will generate display timing. Vds_proc has 5 kinds timing generation, only free run timing and different frame size have no relation with the input timing, other kinds timing all based on the input timing, to avoid destroy the TV. It is suggestion that only free run timing is used; don't lock the output timing with the input timing.

9.1.1 Free run mode

In free run timing mode, output timing have no relation with input timing, it will work based on display clock, the following is the relation between output frame rate, vertical total and horizontal total:

Display clock frequency = horizontal total * vertical total * frame rate

Line rate = vertical total * frame rate:

E.g. you select display clock is 27MHz, line rate is 31.47KHz, frame rate is 60Hz

So: H.total = 27M / 31.47K = 858 V.total = 31.47K / 60 = 525

There are free run horizontal & vertical counter based on display clock, the following registers you programming based on the horizontal & vertical counters:

Register Name	Address	Description
Vds_sync_en	Reg_S3_00 [0]	Sync lock with input format sync, You should set to 0 for free run mode
Vds_fieldab_en	Reg_S3_00 [1]	=0, select AABB double field mode =1, select ABAB double field mode
Vds_dfield_en	Reg_S3_00 [2]	=0, output progressive mode =1, output interlace mode
Vds_flock_en	Reg_S3_1a [4]	Frame lock mode enable, You should set to 0 for free run mode
Vds_dif_fr_sel_en	Reg_S3_1f [4]	Different frame size selection enable, You should set to 0 for free run mode
Vds_hsync_rst [11:0]	Reg_S3_02 [3:0] Reg_S3_01 [7:0]	Horizontal total value H.total = display clock / output line rate
Vds_vsync_rst [10:0]	Reg_S3_03 [6:0] Reg_S3_02 [7:4]	Vertical total value V.total = output line rate / output frame rate
Vds_hb_st [11:0]	Reg_S3_05 [3:0] Reg_S3_04 [7:0]	Horizontal blanking start position (for fetch data)
Vds_hb_sp [11:0]	Reg_S3_06 [7:0] Reg_S3_05 [7:4]	Horizontal blanking stop position (for fetch data)
Vds_vb_st [10:0]	Reg_S3_08 [2:0] Reg_S3_07 [7:0]	Vertical blanking start position (for fetch data)
Vds_vb_sp [10:0]	Reg_S3_09 [6:0] Reg_S3_08 [7:4]	Vertical blanking stop position (for fetch data)
Vds_hs_st [11:0]	Reg_S3_0b [3:0] Reg_S3_0a [7:0]	Horizontal sync start position
Vds_hs_sp [11:0]	Reg_S3_0c [7:0] Reg_S3_0b [7:4]	Horizontal sync stop position
Vds_vs_st [10:0]	Reg_S3_0e [2:0] Reg_S3_0d [7:0]	Vertical sync start position
Vds_vs_sp [10:0]	Reg_S3_0f [6:0] Reg_S3_0e [7:4]	Vertical sync stop position

Vds_dis_hb_st [11:0]	Reg_S3_11 [3:0] Reg_S3_10 [7:0]	Horizontal display blanking start position (for display)
Vds_dis_hb_sp [11:0]	Reg_S3_12 [7:0] Reg_S3_11 [7:4]	Horizontal display blanking stop position (for display)
Vds_dis_vb_st [10:0]	Reg_S3_14 [2:0] Reg_S3_13 [7:0]	Vertical display blanking start position (for display)
Vds_dis_vb_sp [10:0]	Reg_S3_15 [6:0] Reg_S3_14 [7:4]	Vertical display blanking stop position (for display)
Vds_ext_hb_st [11:0]	Reg_S3_6e [3:0] Reg_S3_6d [7:0]	Horizontal output blanking start position (for output hb)
Vds_ext_hb_sp [11:0]	Reg_S3_6f [7:0] Reg_S3_6e [7:3]	Horizontal output blanking stop position (for output hb)
Vds_ext_vb_st [10:0]	Reg_S3_71 [2:0] Reg_S3_70 [7:0]	Vertical output blanking start position (for output vb)
Vds_ext_vb_sp [10:0]	Reg_S3_72 [6:0] Reg_S3_71 [7:4]	Vertical output blanking stop position (for output vb)

Note:

- The value of vds_hb_st, vds_hb_sp, vds_hs_st, vds_hs_sp, vds_dis_hb_st, vds_dis_hb_sp, vds_ext_hb_st, vds_ext_hb_sp should less than the value of vds_hsync_rst.
- The value of vds_vb_st, vds_vb_sp, vds_vs_st, vds_vs_sp, vds_dis_vb_st, vds_dis_vb_sp, Vds_ext_vb_sp, vds_ext_vb_sp should less than the value of vds_vsync_rst.

9.1.2 Sync lock mode

In Sync lock timing mode, output timing will lock with input timing in each H-sync and V-sync.

Register Name	Address	Description
Vds_sync_en	Reg_S3_00 [0]	Sync lock with input format sync, You should set to 1 for sync-lock mode
Vds_flock_en	Reg_S3_1a [4]	Frame lock mode enable, You should set to 0 for sync-lock mode

Vds_hsync_rst should larger than the period total of input H total, always program to max value (0xfff).
Vds_vsync_rst should larger than the period total of input V total, always program to max value (0x7ff).
Horizontal blanking and horizontal sync's position setting is based on the input horizontal timing.
Vertical blanking and vertical sync's position setting is based on the input vertical timing.
Note: In sync-lock mode, you must guarantee the display clock is in phase with input clock

9.1.3 Frame lock mode

In frame lock timing mode, input timing V-sync will reset output timing in lock interval.

Register Name	Address	Description
Vds_flock_en	Reg_S3_1a [4]	Frame lock mode enable, You should set to 1 for frame-lock mode
Vds_frame_rst [9:0]	Reg_S3_1a [2:0] Reg_S3_19 [7:0]	Number of frames which will be locked

Vds_frame_rst defines the numbers of input frame at every lock interval (reset the output timing). For example, we want output timing align with the input timing at every 5 input frames, so we program this register to 4.

9.1.4 Different frame size mode

Different frame size is mainly used in Philips 100iAABB and 75i.
VDS_PROC can generate 3 kinds of frame size; its sequence period can be 16 frames, for example:

```
      A  ………………………………… A
      \__________ _____________/
                 V
         Can up to 16 frames
```

A can select 3 kinds different frame size.

9.1.4.1 Philips 100iAABB mode:

Register Name	Address	Description
Vds_dfield_en	Reg_S3_00 [2]	Interlace output timing enable You should set to 1 for Philips 100iAABB mode
Vds_fieldab_en	Reg_S3_00 [1]	Enable ABAB double field mode You should set to 0 for Philips 100iAABB mode
Vds_field_flip	Reg_S3_00 [3]	Flip field signal for fetch data
Vds_dif_fr_sel_en	Reg_S3_1f [4]	Different frame size mode enable You should set to 1 for Philips 100iAABB mode
Vs_fr_select [31:0]	Reg_S3_1e [7:0] Reg_S3_1d [7:0] Reg_S3_1c [7:0] Reg_S3_1b [7:0]	The frame select, it is a 32-bit register for 16 frame size select, bit [1:0] select frame 0 size, bit [3:2] select frame 1 size, bit [2n+1:2n] select frame n size (0<= n <=16). The size select is like this: Frame select bit [2n+1:2n] • 0, vertical total select vds_vsync_rst. • 1, vertical total select vds_vsync_size1. • 2, vertical total select vds_vsync_size2.
Vds_frame_no	Reg_S3_1f [3:0]	Number of frames to be repeated for a cycle. You can select up to 16 frames to be repeated
Vds_en_fr_num_rst	Reg_S3_1f [5]	Frame number reset enable You should set to 1 for Philips 100iAABB mode
Vd_freerun_fid	Reg_S3_1a [5]	Enable internal free run filed index generation You should set to 1 for Philips 100iAABB mode
Vds_fid_aa_dly	Reg_S3_1a [6]	Enable internal free run AABB field delay 1 frame.
Vds_fid_rst	Reg_S3_1a [7]	Enable internal free run field index reset. You should set to 1 for Philips 100iAABB mode
Vds_vsync_rst [10:0]	Reg_S3_03 [6:0] Reg_S3_02 [7:4]	Vertical total value0
Vds_vsyn_size1 [10:0]	Reg_S3_21 [2:0] Reg_S3_20 [7:0]	Vertical total value1
Vds_vsyn_size2 [10:0]	Reg_S3_23 [2:0] Reg_S3_22 [7:0]	Vertical total value2
Pb_db_field_en	Reg_S4_2b [4]	Interlace mode output enable You should set to 1 for Philips 100iAABB mode

9.1.4.2 Philips 75i mode

Register Name	Address	Description
Vds_dfield_en	Reg_S3_00 [2]	Interlace output timing enable You should set to 1 for Philips 75i mode
Vds_fieldab_en	Reg_S3_00 [1]	Enable ABAB double field mode You should set to 1 for Philips 75i mode
Vds_field_flip	Reg_S3_00 [3]	Flip field signal for fetch data
Vds_dif_fr_sel_en	Reg_S3_1f [4]	Different frame size mode enable You should set to 1 for Philips 75i mode
Vs_fr_select [31:0]	Reg_S3_1e [7:0] Reg_S3_1d [7:0] Reg_S3_1c [7:0] Reg_S3_1b [7:0]	The frame select, it is a 32-bit register for 16 frame size select, bit [1:0] select frame 0 size, bit [3:2] select frame 1 size, bit [2n+1:2n] select frame n size (0<= n <=16). The size select is like this: Frame select bit [2n+1:2n] • 0, vertical total select vds_vsync_rst. • 1, vertical total select vds_vsync_size1. • 2, vertical total select vds_vsync_size2.
Vds_frame_no	Reg_S3_1f [3:0]	Number of frames to be repeated for a cycle. You can select up to 16 frames to be repeated
Vds_en_fr_num_rst	Reg_S3_1f [5]	Frame number reset enable You should set to 1 for Philips 75i mode
Vd_freerun_fid	Reg_S3_1a [5]	Enable internal free run filed index generation You should set to 1 for Philips 75i mode
Vds_fid_rst	Reg_S3_1a [7]	Enable internal free run field index reset. You should set to 1 for Philips 75i mode
Vds_vsync_rst [10:0]	Reg_S3_03 [6:0] Reg_S3_02 [7:4]	Vertical total value0
Vds_vsyn_size1 [10:0]	Reg_S3_21 [2:0] Reg_S3_20 [7:0]	Vertical total value1
Vds_vsyn_size2 [10:0]	Reg_S3_23 [2:0] Reg_S3_22 [7:0]	Vertical total value2
Pb_db_field_en	Reg_S4_2b [4]	Interlace mode output enable You should set to 1 for Philips 75i mode

9.1.5 Interlace timing program

In interlace timing mode, playback must fetch the field data from memory, so playback must enable interlace mode, at the same time, we program video processor to output interlace timing

For example: **VCLK=A, Line rate=B, Field rate=C**
So Vds_hsync_rst = (A/B) / 2
Vds_vsync_rst = (B/C) * 2

Register Name	Address	Description
Vds_dfield_en	Reg_S3_00 [2]	Interlace output timing enable You should set to 1 for interlace output mode
Vds_fieldab_en	Reg_S3_00 [1]	=0, for double field AABB mode =1, for double field ABAB mode
Vds_field_flip	Reg_S3_00 [3]	Flip field signal for fetch data
Vd_freerun_fid	Reg_S3_1a [5]	Enable internal free run filed index generation You should set to 1 for Philips 75i mode
Vds_hsync_rst [11:0]	Reg_S3_02 [3:0] Reg_S3_01 [7:0]	Horizontal total value H.total = (display clock / output line rate) / 2
Vds_vsync_rst [10:0]	Reg_S3_03 [6:0] Reg_S3_02 [7:4]	Vertical total value V.total = (output line rate / output frame rate) * 2 The vertical total must be ODD value, so it will be programmed EVEN value
Pb_db_field_en	Reg_S4_2b [4]	Interlace mode output enable You should set to 1 for Philips 75i mode

9.1.6 HB and VB program

9.1.6.1 HB and HS program

In 5725, there are 3 groups horizontal blanking timing.

- Hb_st/hb_sp: is used to fetch playback data.
- Dis_hb_st/dis_hb_sp: is used for display. So it can avoid the red line on the left/right edge.
- Ext_hb_st/ext_hb_sp: is used to output horizontal blanking!

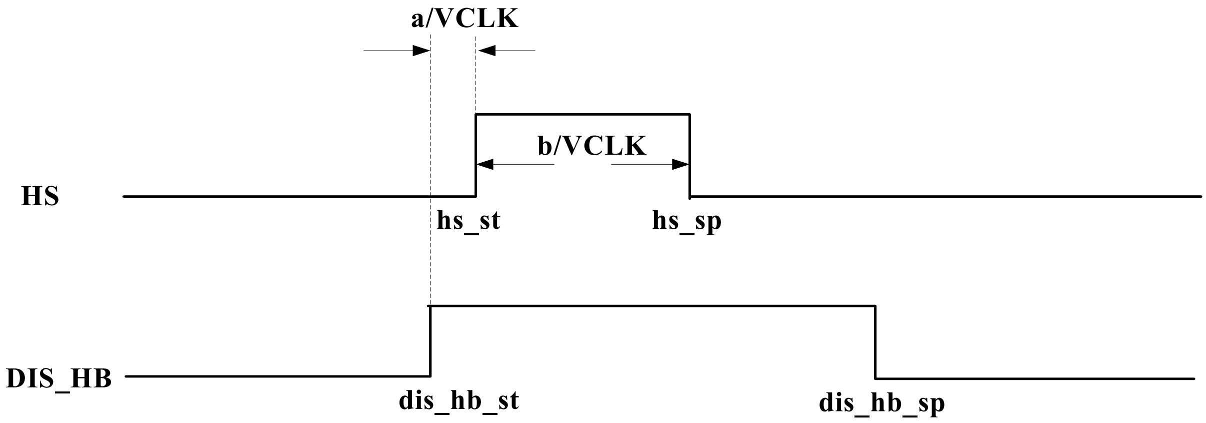
The following table is hb_st/hb_sp & dis_hb_st/dis_hb_sp relations:
Commonly: We set **vds_dis_hb_st = H.total – 1, vds_dis_hb_sp = H.blank – 1;**
Vds_uv_flip = 0, vds_u_delay = 1, vds_v_delay = 0, vds_y_delay = 2;

Scale Ratio	V2CLK=VCLK		V2CLK=2*VCLK	
	HB_ST	HB_SP	HB_ST	HB_SP
No scale	Dis_hb_st-60	Dis_hb_sp-79	Dis_hb_st-49	Dis_hb_sp-68
(987,1023]	Dis_hb_st-62	Dis_hb_sp-82	Dis_hb_st-51	Dis_hb_sp-71
(953,987]	Dis_hb_st-63	Dis_hb_sp-83	Dis_hb_st-52	Dis_hb_sp-72
(921,953]	Dis_hb_st-64	Dis_hb_sp-84	Dis_hb_st-53	Dis_hb_sp-73
(891,921]	Dis_hb_st-65	Dis_hb_sp-85	Dis_hb_st-54	Dis_hb_sp-74
(863,891]	Dis_hb_st-66	Dis_hb_sp-86	Dis_hb_st-55	Dis_hb_sp-75
(837,863]	Dis_hb_st-67	Dis_hb_sp-87	Dis_hb_st-56	Dis_hb_sp-76
(813,837]	Dis_hb_st-68	Dis_hb_sp-88	Dis_hb_st-57	Dis_hb_sp-77
(789,813]	Dis_hb_st-69	Dis_hb_sp-89	Dis_hb_st-58	Dis_hb_sp-78
(767,789]	Dis_hb_st-70	Dis_hb_sp-90	Dis_hb_st-59	Dis_hb_sp-79
(747,767]	Dis_hb_st-71	Dis_hb_sp-91	Dis_hb_st-60	Dis_hb_sp-80
(727,747]	Dis_hb_st-72	Dis_hb_sp-92	Dis_hb_st-61	Dis_hb_sp-81
(708,727]	Dis_hb_st-73	Dis_hb_sp-93	Dis_hb_st-62	Dis_hb_sp-82
(691,708]	Dis_hb_st-74	Dis_hb_sp-94	Dis_hb_st-63	Dis_hb_sp-83
(674,691]	Dis_hb_st-75	Dis_hb_sp-95	Dis_hb_st-64	Dis_hb_sp-84
(658,674]	Dis_hb_st-76	Dis_hb_sp-96	Dis_hb_st-65	Dis_hb_sp-85
(642,658]	Dis_hb_st-77	Dis_hb_sp-97	Dis_hb_st-66	Dis_hb_sp-86

(628,642]	Dis_hb_st-78	Dis_hb_sp-98	Dis_hb_st-67	Dis_hb_sp-87
(614,628]	Dis_hb_st-79	Dis_hb_sp-99	Dis_hb_st-68	Dis_hb_sp-88
(601,614]	Dis_hb_st-80	Dis_hb_sp-100	Dis_hb_st-69	Dis_hb_sp-89
(588,601]	Dis_hb_st-81	Dis_hb_sp-101	Dis_hb_st-70	Dis_hb_sp-90
(575,588]	Dis_hb_st-82	Dis_hb_sp-102	Dis_hb_st-71	Dis_hb_sp-91
(564,575]	Dis_hb_st-83	Dis_hb_sp-103	Dis_hb_st-72	Dis_hb_sp-92
(552,564]	Dis_hb_st-84	Dis_hb_sp-104	Dis_hb_st-73	Dis_hb_sp-93
(542,552]	Dis_hb_st-85	Dis_hb_sp-105	Dis_hb_st-74	Dis_hb_sp-94
(531,542]	Dis_hb_st-86	Dis_hb_sp-106	Dis_hb_st-75	Dis_hb_sp-95
(521,531]	Dis_hb_st-87	Dis_hb_sp-107	Dis_hb_st-76	Dis_hb_sp-96
(511,521]	Dis_hb_st-88	Dis_hb_sp-108	Dis_hb_st-77	Dis_hb_sp-97

Reference Register Map:

Register Name	Address	Description
Vds_hb_st [11:0]	Reg_S3_05 [3:0] Reg_S3_04 [7:0]	Horizontal blanking start position (for fetch data)
Vds_hb_sp [11:0]	Reg_S3_06 [7:0] Reg_S3_05 [7:4]	Horizontal blanking stop position (for fetch data)
Vds_dis_hb_st [11:0]	Reg_S3_11 [3:0] Reg_S3_10 [7:0]	Horizontal display blanking start position (for display)
Vds_dis_hb_sp [11:0]	Reg_S3_12 [7:0] Reg_S3_11 [7:4]	Horizontal display blanking stop position (for display)
Vds_ext_hb_st [11:0]	Reg_S3_6e [3:0] Reg_S3_6d [7:0]	Horizontal output blanking start position (for output hb)
Vds_ext_hb_sp [11:0]	Reg_S3_6f [7:0] Reg_S3_6e [7:3]	Horizontal output blanking stop position (for output hb)



1 seperate sync output

1) v2clk=v4clk
hs_st=dis_hb_st+a/vclk+6(v2clk)
2)v2=2*v4clk
hs_st=dis_hb_st+a/vclk+5.5(v2clk)

2 sync on y

hs_st=dis_hb_st+a/vclk

9.1.6.2 VB and VS program:

In 5725, there are 3 groups vertical blanking timing.

- vb_st/vb_sp: is used to fetch playback data.
- Dis_vb_st/dis_vb_sp: is used for display. So it can avoid the red line on the left/right edge.
- Ext_vb_st/ext_vb_sp: is used to output vertical blanking!

Reference Register Map:

Register Name	Address	Description
Vds_vb_st [10:0]	Reg_S3_08 [2:0] Reg_S3_07 [7:0]	Vertical blanking start position (for fetch data)
Vds_vb_sp [10:0]	Reg_S3_09 [6:0] Reg_S3_08 [7:4]	Vertical blanking stop position (for fetch data)
Vds_dis_vb_st [10:0]	Reg_S3_14 [2:0] Reg_S3_13 [7:0]	Vertical display blanking start position (for display)
Vds_dis_vb_sp [10:0]	Reg_S3_15 [6:0] Reg_S3_14 [7:4]	Vertical display blanking stop position (for display)
Vds_ext_vb_st [10:0]	Reg_S3_71 [2:0] Reg_S3_70 [7:0]	Vertical output blanking start position (for output vb)
Vds_ext_vb_sp [10:0]	Reg_S3_72 [6:0] Reg_S3_71 [7:4]	Vertical output blanking stop position (for output vb)

Assume (Vb_sp - Vb_st) is the width of Vertical blank,
When Vb_sp > Vb_st, it equals (Vb_sp – Vb_st).
When Vb_sp < Vb_st, it equals (V_total – Vb_sp + Vb_st)

If VSCALE_BYPS = 1
(Vb_sp – Vb_st) = V total – V display enable

If VSCALE_BYPS = 0
(Vb_sp – Vb_st) = (V total – V display enable) – 2

9.2 3D noise reduction

In 5725, 3D noise reduction is used for cross color in vds_proc. The current pixels and the ones from the previous frame (For NTSC mode, the two frames is 1 and 3, for PAL mode, the two frames is 1 and 5) at the same location are compared and mixed to decrease or eliminate the Y/C cross-color.

There are two operating modes available: the user controlled mode and the signal adaptive mode. In the user-controlled mode a fixed (user defined) motion index is set for averaging the new and old pixel data or not. This mode can easily lead to smearing effects in moving pictures and scene changes and therefore should not be used for normal operation.

In the adaptive mode the noise reduction motion index is affected by the lower frequencies of the difference (y0 – y1) of the luminance signal. The difference signal 'diff [7:0]' is low-pass filtered. This signal 'mi_out [7:0]' will process offset and gain calculation, sigmoid look up table, will get a motion index to determine whether noise reduction will do:

If the motion index is less than the threshold, the noise reduction will do.

If the motion index is higher than the threshold, the noise reduction will not do to avoid the smearing affects.

The noise reduction coefficient (motion index) calculated from of the luminance signal can optionally be coupled to the color processing circuit in order to control the chrominance noise reduction. The advantage of coupling is that cross color is reduced. The disadvantage is possible smearing of moving colored objects that have little Y-contrast with the background. Therefore, it is suggested to use coupling in applications without active comb-filter and no coupling whenever a comb filter is activated.

9.2.1 Motion Index Generation

If manual mode enable, it will select the register value as motion index, if auto mode enable, it will calculate the motion index based on the formula:

```
If (vds_nr_mig_user_en)
        Motion index = mi_offset [3:0]
else
        Motion index = (diff – offset) * gain
```

Note: vds_nr_mig_user_en: Reg_S3_53 [7]

From the formula, offset ↑, motion index will ↓, so noise reduction will ↑. So the offset is larger, the Motion index will be less; the noise reduction will be greater.

From the formula, gain ↑, motion index will ↑, so noise reduction will ↓. So the gain is larger, the Motion index will be larger; the noise reduction will be less.

9.2.1.1 Motion Index Gain selection

Motion index gain value has 2 groups, one is for normal condition, the other is for still condition; the still condition case gain value is less than the normal condition (e.g. still gain is half of the normal gain). So we will do more Y/C noise reduction for still picture!

Global still generation:

Vds_glb_noise [10:0]	Vds_nr_en_glb_still	Vds_nr_glb_still_menu	Vds_glb_still
Reg_S3_52 [2:0] Reg_S3_51 [7:0]	Reg_S3_55 [6]	Reg_S3_55 [7]	
X	0	0	0
X	0	1	1
ACC > glb_noise	1	0	0
ACC =< glb_noise	1	0	1

Motion Index Gain selection:

Vds_glb_still	Vds_nr_mi_gain [3:0]	Vds_nr_still_gain [3:0]	MI_GAIN [3:0]
	Reg_S3_54 [3:0]	Reg_S3_54 [7:4]	
0	√		Vds_nr_mi_gain
1		√	Vds_nr_still_gain

9.2.1.2 Motion Index Offset Selection

Motion index offset has 2 groups, one is for normal condition, the other is for high noise condition, the High noise condition offset value is larger than the normal condition, so we will do more noise reduction for high noise condition!

High noise condition generation:

Madpt_nosie_threshold_vds [6:0]	nout_for_vds
Reg_S2_05 [6:0]	
ACC > glb_noise	0
ACC =< glb_noise	1

Motion Index Offset selection:

Vds_nr_en_h_noisy	Nout_for_vds	Vds_nr_mi_offset	Vds_nr_noisy_offset	MI_OFFSET
Reg_S3_55 [4]		Reg_S3_53 [6:0]	Reg_S3_56 [6:0]	
0	X	√		Vds_nr_mi_offset
1	0	√		Vds_nr_mi_offset
1	1		√	Vds_nr_noisy_offset

9.2.2 VT Filter Control

There are 2 VT filters used for noise reduction, one is for Y (luminance), the other is for UV (color).
VT filter function description:

```
If (vds_mi_th_en)
        If (byps =1 or motion index = 15 or motion index > (threshold value))
                Doesn't Do Y/C noise reduction;
        Else
                Do complete Y/C noise reduction;
Else
        Do Y/C noise reduction as the motion index value!
```

VT filter control register description:

Name	Address	Register Description
Vds_nr_y_bypass	Reg_S3_52 [4]	Y bypass the noise reduction process control
Vds_nr_c_bypass	Reg_S3_52 [5]	C bypass the noise reduction process control
Vds_nr_mi_th_en	Reg_S3_52 [7]	Motion index threshold control enable
Vds_nr_mi_thresh	Reg_S3_55 [3:0]	Noise reduction motion index threshold

9.3 9.3 H/V Scaling Up

9.3.1 Vertical scaling up

In 5725, vertical scaling up process, use two lines data do 1st order phase adjustment. There will use line buffer FF1024 storage a line data, so for vertical scaling up, our horizontal active input data from playback can't exceed 1024 pixels.

Note: We refer to the input data from playback can't exceed 1024 not the output pixels can't exceed 1024 pixel because we do V-scaling first then do H-scaling!

This line buffer FF1024 is shared with vertical peaking function, so is you enable vertical peaking function you must disable vertical scaling up, if you enable vertical scaling up function, you must disable vertical peaking function.

Register Name	Register Map	Register description
Vds_vscale [9:0]	Reg_S3_18 [6:0] Reg_S3_17 [7:4]	Vertical scaling up coefficient VSCALE = 1024 * (Input V.active / Output V.active)
Vds_vscale_byps	Reg_S3_00 [5]	Bypass vertical scaling up 1st phase adjustment.

9.3.2 Horizontal scaling up

In 5725, we only use 1st order phase adjustment to do scale up.

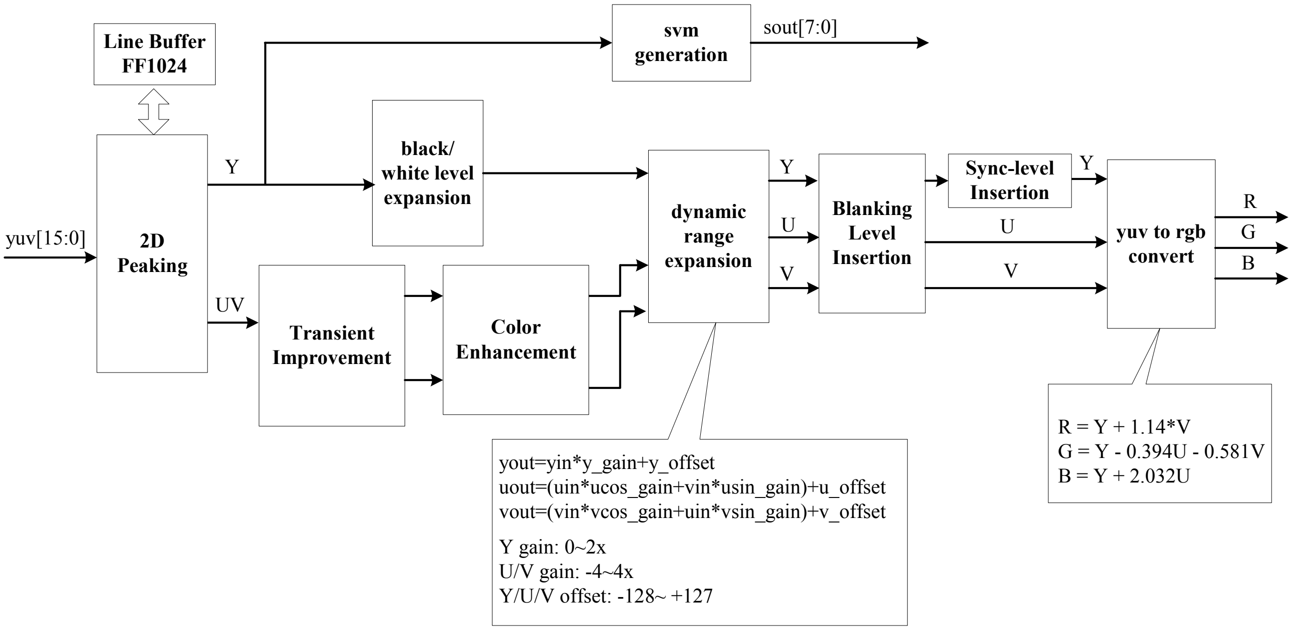
Register Name	Register Map	Register description

Vds_hscale [9:0]	Reg_S3_17 [1:0] Reg_S3_16 [7:0]	Horizontal scaling up coefficient HSCALE = 1024 * (Input H.active / Output H.active)
Vds_hscale_byps	Reg_S3_00 [4]	Bypass horizontal scaling up ^{t} phase adjustment.

9.4 Video Enhancement

Blew is Video Enhancement data path:

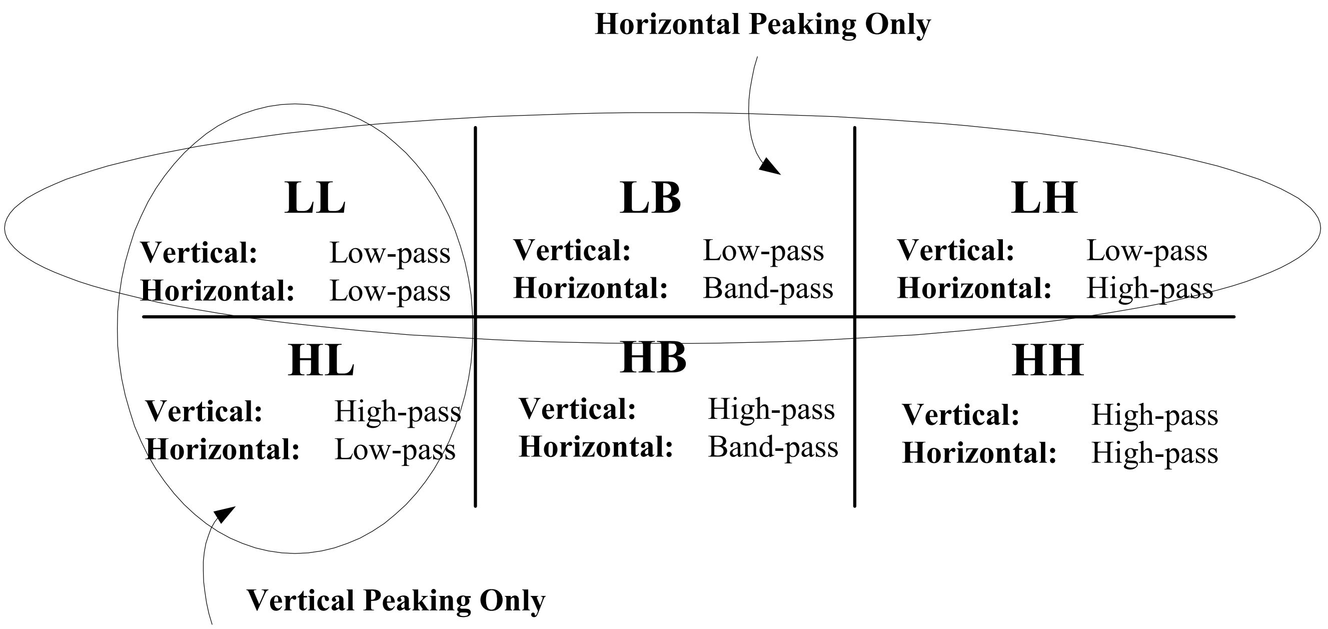
Video Enhancement



9.4.1 2D Peaking

Especially with decoded composite signals and notch filter luminance/color separation, as input signals, It is necessary to improve the luminance/color frequency characteristics, 5725 2D peaking process is edge enhancement used to improve the Y and C sharpness for vertical and horizontal direction, it will be divided six frequency areas: LL, LB, LH, HL, HB, HH. For each area, for Y, there is 2D peaking, for UV, there is only vertical peaking, and there are 3 main controls:

Coring level:	Used to remove low amplitudes in the signals, which are considered as noise
Compareing level:	Used to control peaking range, when input larger than it, don't peaking.
Gain value:	Used to gain value, it bigger, do more peaking.



Note: Vertical peaking use a line buffer FF1024, it shared with vertical scaling up, so vertical peaking and vertical scaling up can’t active at the same time.

Reference Registers Map:

Register Name	Register Map	Register description
Vds_pk_y_h_byps	Reg_S3_4e [0]	=0, enable Y horizontal peaking =1, bypass Y horizontal peaking
Vds_pk_y_v_byps	Reg_S3_4e [1]	=0, enable Y vertical peaking =1, bypass Y vertical peaking
Vds_c_vpk_bypass	Reg_S3_4e [3]	=0, enable UV vertical peaking =1, bypass UV vertical peaking
Vds_pk_lb_core	Reg_S3_44 [2:0]	Vertical low-pass, horizontal band-pass signal coring level
Vsd_pk_lb_cmp	Reg_S3_44 [7:3]	Vertical low-pass, horizontal band-pass signal comparing level
Vds_pk_lb_gain	Reg_S3_45 [5:0]	Vertical low-pass, horizontal band-pass signal gain control Range: (0~4) * 16
Vds_pk_lh_core	Reg_S3_46 [2:0]	Vertical low-pass, horizontal high-pass signal coring level
Vsd_pk_lh_cmp	Reg_S3_46 [7:3]	Vertical low-pass, horizontal high-pass signal comparing level
Vds_pk_lh_gain	Reg_S3_47 [5:0]	Vertical low-pass, horizontal high-pass signal gain control Range: (0~4) * 16
Vds_pk_hl_core	Reg_S3_48 [2:0]	Vertical high-pass, horizontal low-pass signal coring level
Vsd_pk_hl_cmp	Reg_S3_48 [7:3]	Vertical high-pass, horizontal low-pass signal comparing level

Vds_pk_hl_gain	Reg_S3_49 [5:0]	Vertical high-pass, horizontal low-pass signal gain control Range: (0~4) * 16
Vds_pk_hb_core	Reg_S3_4a [2:0]	Vertical high-pass, horizontal band-pass signal coring level
Vsd_pk_hb_cmp	Reg_S3_4a [7:3]	Vertical high-pass, horizontal band-pass signal comparing level
Vds_pk_hb_gain	Reg_S3_4b [5:0]	Vertical high-pass, horizontal band-pass signal gain control Range: (0~4) * 16
Vds_pk_hh_core	Reg_S3_4c [2:0]	Vertical high-pass, horizontal high-pass signal coring level
Vsd_pk_hh_cmp	Reg_S3_4c [7:3]	Vertical high-pass, horizontal high-pass signal comparing level
Vds_pk_hh_gain	Reg_S3_4d [5:0]	Vertical high-pass, horizontal high-pass signal gain control Range: (0~4) * 16
Vds_pk_vl_hl_sel	Reg_S3_43 [0]	Vertical low-pass, horizontal lpf select, 0, tap5, 1, tap3
Vds_pk_vl_hh_sel	Reg_S3_43 [1]	Vertical low-pass, horizontal hpf select, 0, tap5, 1, tap3
Vds_pk_vh_hl_sel	Reg_S3_43 [2]	Vertical high-pass, horizontal lpf select, 0, tap5, 1, tap3
Vds_pk_vh_hh_sel	Reg_S3_43 [3]	Vertical high-pass, horizontal hpf select, 0, tap5, 1, tap3

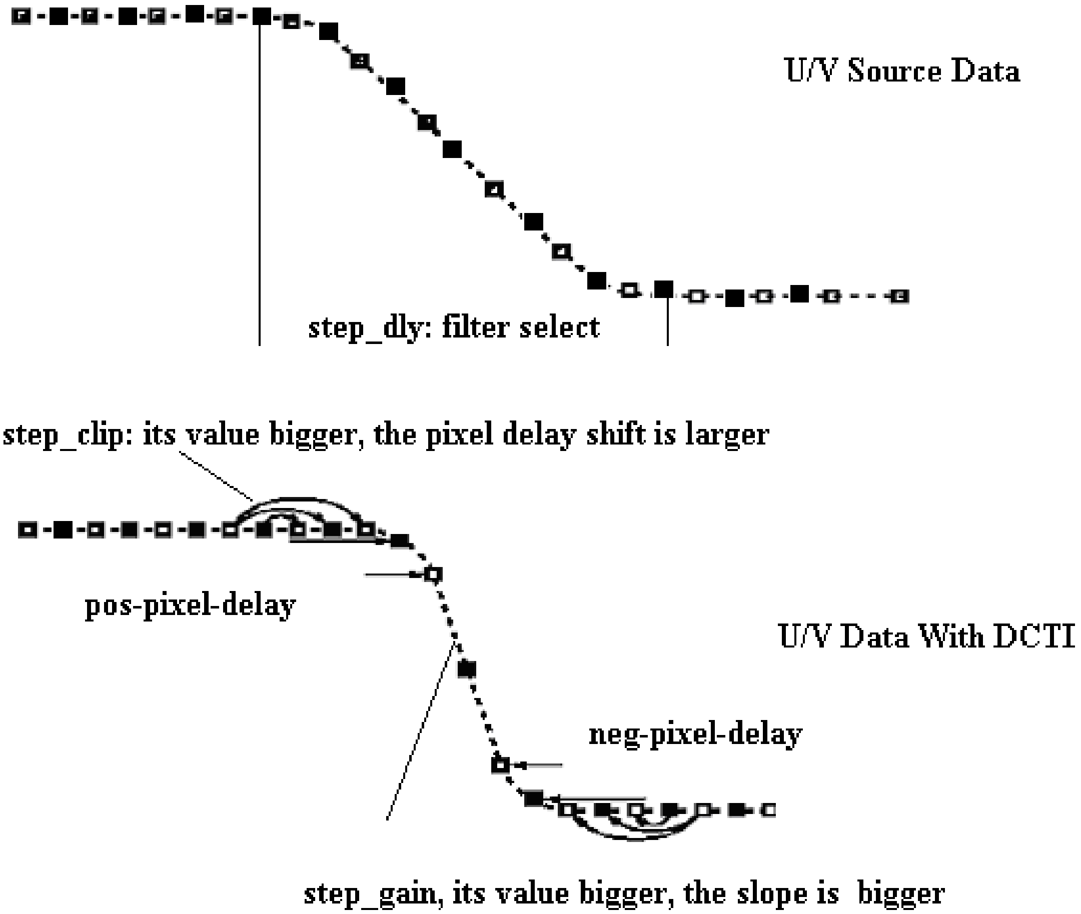
9.4.2 Chroma Transient improvement

Digital Color Transient Improvement (DCTI) are intended to enhance video by replacing the edges of the chrominance with edges that have steeper rise and fall times. The basic principle is to detect horizontal transients and improve their steepness without generating overshoots.

The idea is to vary the data path delay on the basis of a function of the second derivative of the U and V signal. Positive and negative transients are treated alike; the output of the first differentiator therefore is taken as absolute value. The signal is differentiated again and the output used to control the momentary data path delay. The effect at an edge is that during the first half the data path delay is higher than nominal and in the second half it is lower than nominal. This will make the edge much steeper. The control signal that varies the delay is amplified by a user defined gain setting. Increasing this parameter results in a steeper transient.

DCTI are different from peaking in that they do not increase the peak-to-peak video at its output; rather it turns sloped or sinusoidal waveforms into rectangular or square waveforms with the same duty cycles and peak-to-peak amplitude.

The following figure is the principle of DCTI.


U/V Source Data
step_dly: filter select
step_clip: its value bigger, the pixel delay shift is larger
pos-pixel-delay
U/V Data With DCTI
neg-pixel-delay
step_gain, its value bigger, the slope is bigger

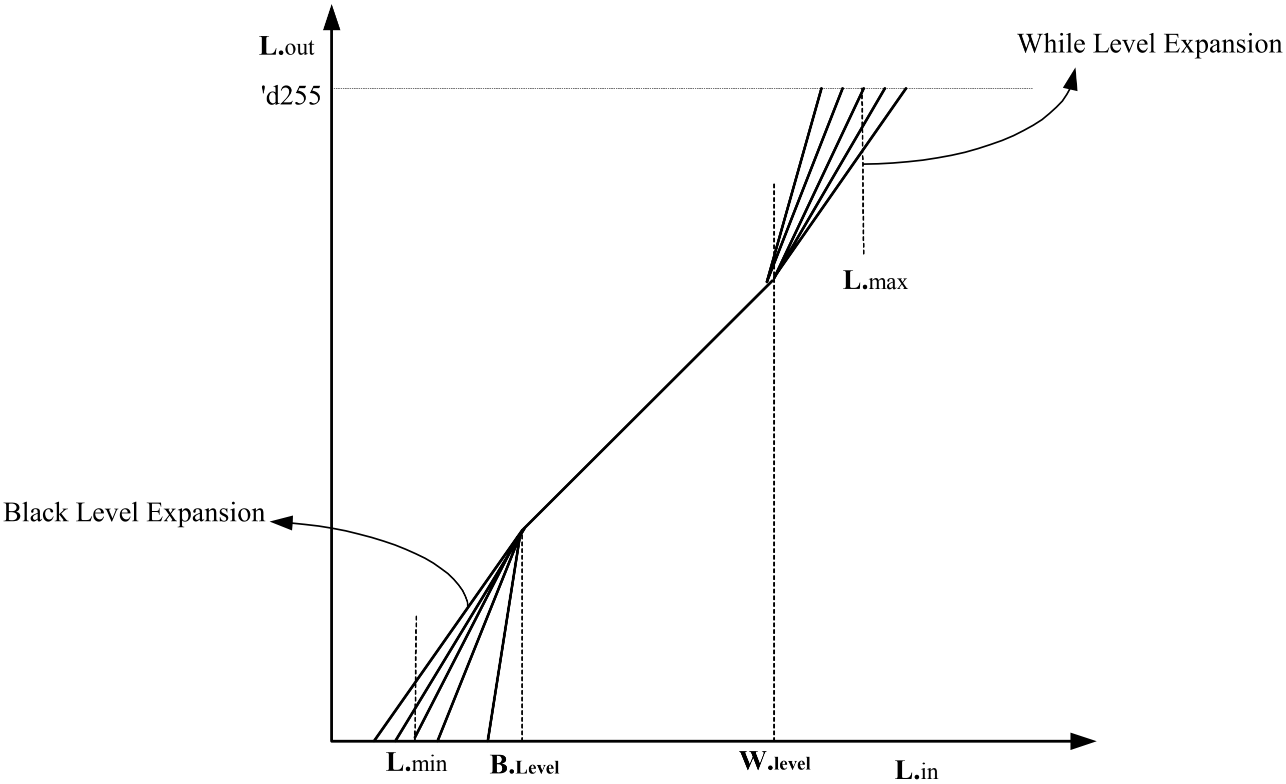
Reference register map

Register Name	Register Map	Register description
Vds_uv_step_byps	Reg_S3_2b [7]	UV step response bypass control
Vds_step_dly_cntrl	Reg_S3_2a [5:4]	UV step response data select control 2'b00: U/V5 – U/V6 2'b01: U/V4 – U/V7 2'b10: U/V3 – U/V8 2'b11: U/V2 – U/V9
Vds_tep_gain	Reg_S3_2b [3:0]	Step response gain control, it can adjust the UV edge improvement, the larger value of this register, the sharper edge will appear, the range of the gain is (0 ~ 4) * 4.

Vds_us_step_clip	Reg_S3_2b [6:4]	UV step response clip control,

9.4.3 Black/White level expansion

The black-level expander enhances the contrast of the picture. Therefore the luminance signal is modified with an adjustable, non-linear function. Dark areas of the picture are changed to black, while bright areas of picture are changed to white. The advantage of this black-level and white level expander is it is performed only if it will be most noticeable to the viewer.



Characteristics of the black-level/white-level expander

Note: There are 2 modes to decide the L.max and L.max value, one is manual mode, and the L.min and L.max are defined by the registers value. The other is adaptive mode, the L.min and L.max is a frames min and max luminance value.

The black-level expander works adaptively. Depending on the measured amplitudes ‘Lmin’ and ‘Lmax’ of a frame luminance and an adjustable coefficient, a tilt point ‘B.level’ is programmed with register bits. Above this value there is no expansion, while all luminance values below this point are expanded according to:

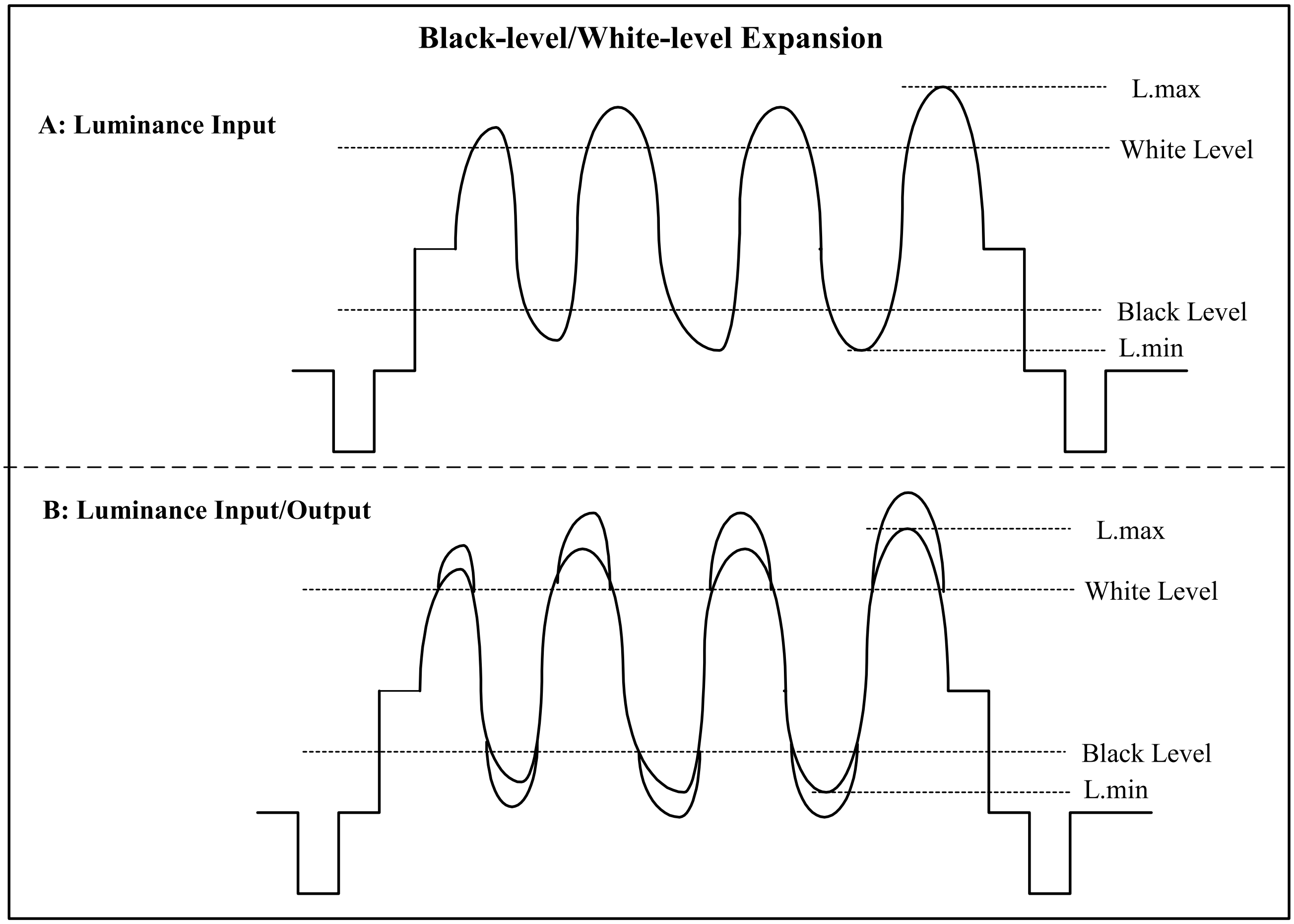
$$\mathbf{Y_{out} = Y_{in} - (B.level - Y_{in}) * (256 - (L_{max} - L_{min}) * B.gain / (256 * 16)}$$

Note: The gain value larger, the black level expansion slope is larger.
The ($L_{max} - L_{min}$) value larger, the black level expansion slope is larger.

The white-level expander works adaptively. Depending on the measured amplitudes ‘Lmin’ and ‘Lmax’ of a frame luminance and an adjustable coefficient, a tilt point ‘W.level’ is programmed with register bits. Below this value there is no expansion, while all luminance values below this point are expanded according to:

$Y_{out} = Y_{in} + (Y_{in-}\ W.level) * (256 – (L_{max} – L_{min}) * W.gain / (256 * 16)$

Note: The gain value larger, the white level expansion slope is larger.
The (L_{max} – L_{min}) value larger, the white level expansion slope is larger.



Reference Register Map

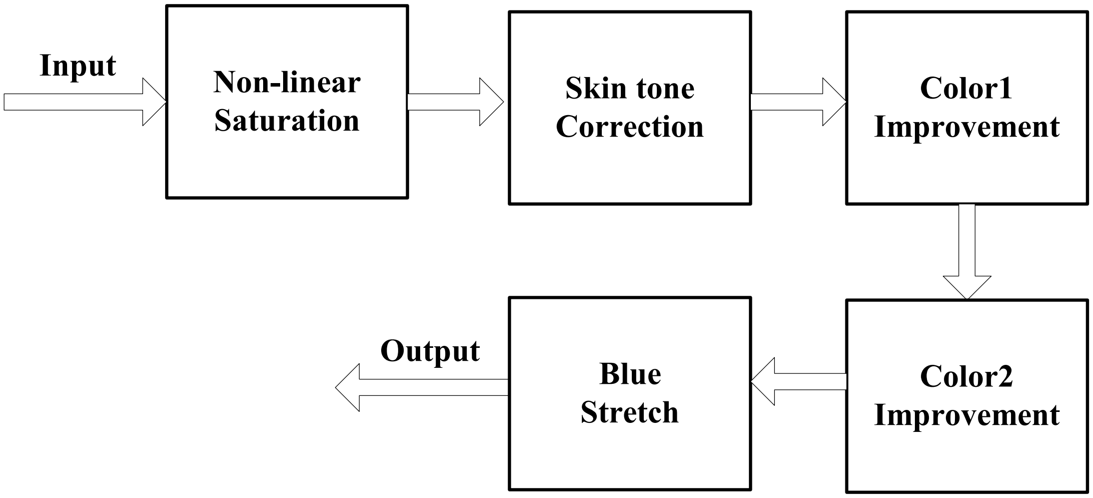
Register Name	Register Map	Register description
Vds_blev_byps	Reg_S3_2a [0]	Black level expansion bypass control
Vds_blev_auto_en	Reg_S3_26 [7]	Set to 1, L.min and L.max of each frame will detect Set to 0, L.min and L.max are register defined
Vds_user_min	Reg_S3_27 [3:0]	User defined min value for black level expansion Real value = {3'b000, user_min, 1'b1}
Vds_user_max	Reg_S3_27 [7:4]	User defined max value for black level expansion Real value = {user_max, 4'b1111}
Vds_blev_level	Reg_S3_28 [7:0]	Black level expansion threshold value, data larger than this value no black level expansion

Vds_blev_gain	Reg_S3_29 [7:0]	The gain value for black level expansion, its range is (0 ~ 16) * 16
Vds_w_lev_byps	Reg_S3_56 [7]	White level expansion bypass control
Vds_w_lev	Reg_S3_57 [7:0]	White level expansion threshold value, data less than this value no white level expansion
Vds_wlev_gain	Reg_S3_58 [7:0]	The gain value for white level expansion, its range is (0~16x) * 16

9.4.4 Color enhancement

5725 color enhance is used to improve the color visual quality, it mainly includes 4 part, non-linear Saturation, skin tone correction, 2 colors Improvement and blue stretch.

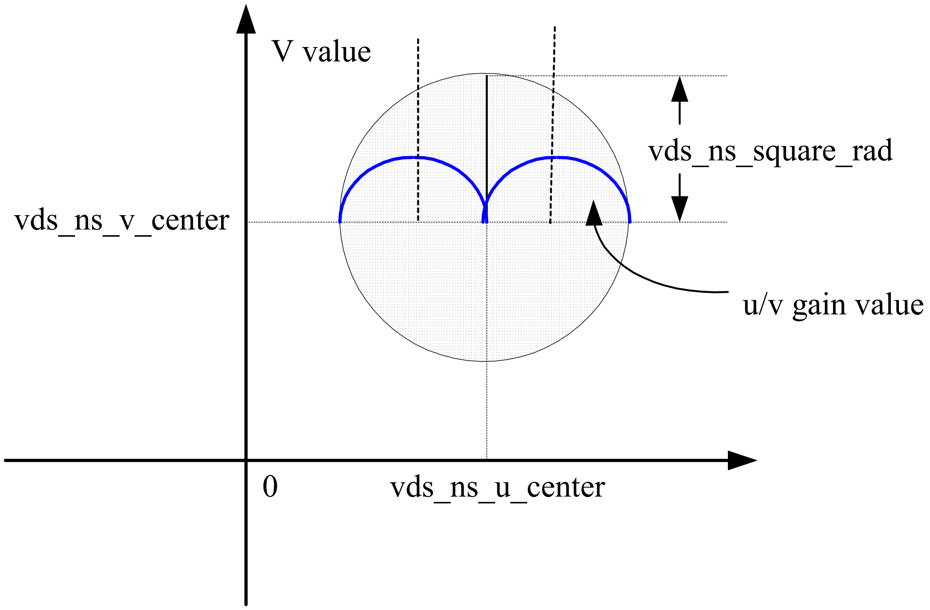
Below is the block diagram:



9.4.4.1 Non-linear saturation

Non-linear saturation is used to enhance the saturation for some area and not affect other area, it Define a circle area based on UV value. In the circle, the saturation will be enhanced; out of the circle, the saturation will keep original.

At the same time, we can set a Y range to control the non-linear saturation area, when the input within the Y range, do non-linear saturation; when the input out of the range, keep original:
Note: the u/v gain value is change smoothly to avoid the saturation impulse artifact.



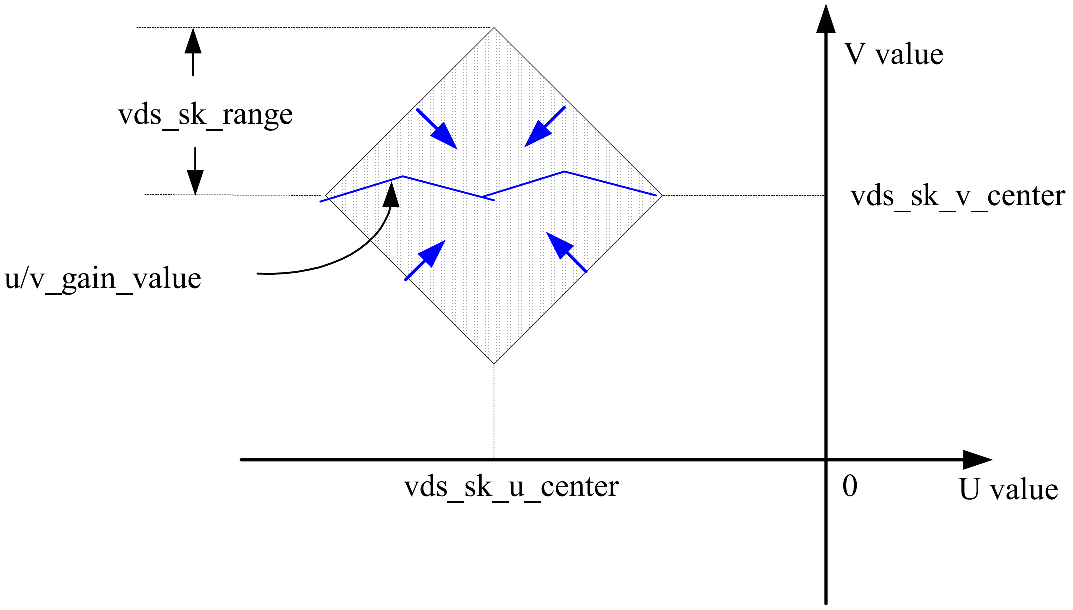
Reference Register Map

Register Name	Register Map	Register description
Vds_ns_u_center [7:0]	Reg_S3_59 [7:0]	Non-linear saturation center point u value
Vds_ns_v_center [7:0]	Reg_S3_5a [7:0]	Non-linear saturation center point v value
Vds_ns_u_gain [6:0]	Reg_S3_5b [6:0]	Non-linear saturation u gain control
Vds_ns_square_rad [14:0]	Reg_S3_5d [5:0] Reg_S3_5c [7:0] Reg_S3_5b [7]	Non-linear saturation uv range $(\mid \text{Uin} - \text{u\_center} \mid)^2 + (\mid \text{Vin} - \text{v\_center} \mid)^2 < \text{range}$
Vds_ns_y_high_th [7:0]	Reg_S3_5e [5:0] Reg_S3_5d [7:6]	Non-linear saturation y high threshold value
Vds_ns_v_gain [6:0]	Reg_S3_5f [4:0] Reg_S3_5e [7:6]	Non-linear saturation v gain control
Vds_ns_y_low_th [4:0]	Reg_S3_60 [1:0] Reg_S3_5f [7:5]	Non-linear saturation y low threshold value
Vds_ns_byps	Reg_S3_60 [2]	Non-linear saturation bypass control
Vds_ns_y_active_en	Reg_S3_60 [3]	Non-linear y active enable

9.4.4.2 Skin tone correction

Skin tone correction is used to correct the skin color when the pixel fall into the skin range we defined, it Define a square area based on UV value. In the square, the color will be corrected; out of the square, the color will keep original.

At the same time, we can set a Y range to control the skin tone correction area, when the input within the Y range, do skin tone correction; when the input out of the range, keep original:
Note: the u/v gain value is change smoothly to avoid the saturation impulse artifact



Reference Register Map

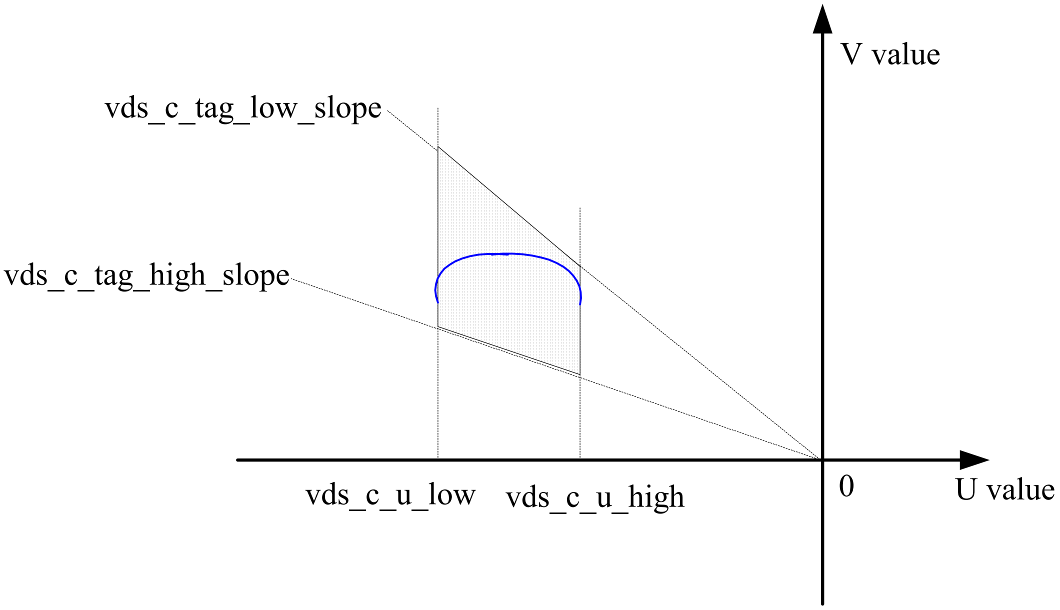
Register Name	Register Map	Register description
Vds_sk_u_center [7:0]	Reg_S3_2c [7:0]	Skin tone correction center point u value
Vds_sk_v_center [7:0]	Reg_S3_2d [7:0]	Skin tone correction center point v value
Vds_sk_y_low_th [7:0]	Reg_S3_2e [7:0]	Skin tone correction y low threshold value
Vds_sk_y_high_th [7:0]	Reg_S3_2f [7:0]	Skin tone correction y high threshold value
Vds_sk_range [7:0]	Reg_S3_30 [7:0]	Skin tone correction range (| Uin – u_center | + | Vin – v_center | < range
Vds_sk_gain [3:0]	Reg_S3_31 [3:0]	Skine tone correction gain control (0 ~ 1) * 15
Vds_sk_y_en	Reg_S3_31 [4]	Skin tone correction y control enable
Vds_sk_byps	Reg_S3_31 [5]	Skin tone correction bypass control

9.4.4.3 Color Improvement

Color improvement is used to enhance some color and not affect other area, it define an echelon area Based on UV value. In the echelon, the color will be enhanced; out of the echelon, the saturation will keep original.

At the same time, we can set a Y range to control the non-linear saturation area, when the input within the Y range, do non-linear saturation; when the input out of the range, keep original.

Note: the u/v gain value is change smoothly to avoid the saturation impulse artifact.



Reference Register Map

Register Name	Register Map	Register description
Vds_c1_tag_low_slope [9:0]	Reg_S3_61 [5:0] Reg_S3_60 [7:4]	Color enhance1 low tan slope
Vds_c1_tag_high_slope [9:0]	Reg_S3_62 [7:0] Reg_S3_61 [7:6]	Color enhance1 high tan slope
Vds_c1_gain [7:0]	Reg_S3_63 [3:0]	Color enhance1 gain control
Vds_c1_u_low [7:0]	Reg_S3_64 [3:0] Reg_S3_63 [7:4]	Color enhance1 u low value
Vds_c1_u_high [7:0]	Reg_S3_65 [3:0] Reg_S3_64 [7:4]	Color enhance1 u high value
Vds_c1_byps	Reg_S3_65 [4]	Color enhance1 bypass control
Vds_c1_y_threshold [7:0]	Reg_66 [4:0] Reg_65 [7:5]	Color enhance1 y threshold value
Vds_c2_tag_low_slope [9:0]	Reg_S3_67 [6:0] Reg_S3_66 [7:5]	Color enhance2 low tan slope
Vds_c2_tag_high_slope [9:0]	Reg_S3_69 [0] Reg_S3_68 [7:0] Reg_S3_61 [7]	Color enhance2 high tan slope
Vds_c2_gain [7:0]	Reg_S3_69 [4:1]	Color enhance2 gain control
Vds_c2_u_low [7:0]	Reg_S3_6a [4:0] Reg_S3_69 [7:5]	Color enhance2 u low value

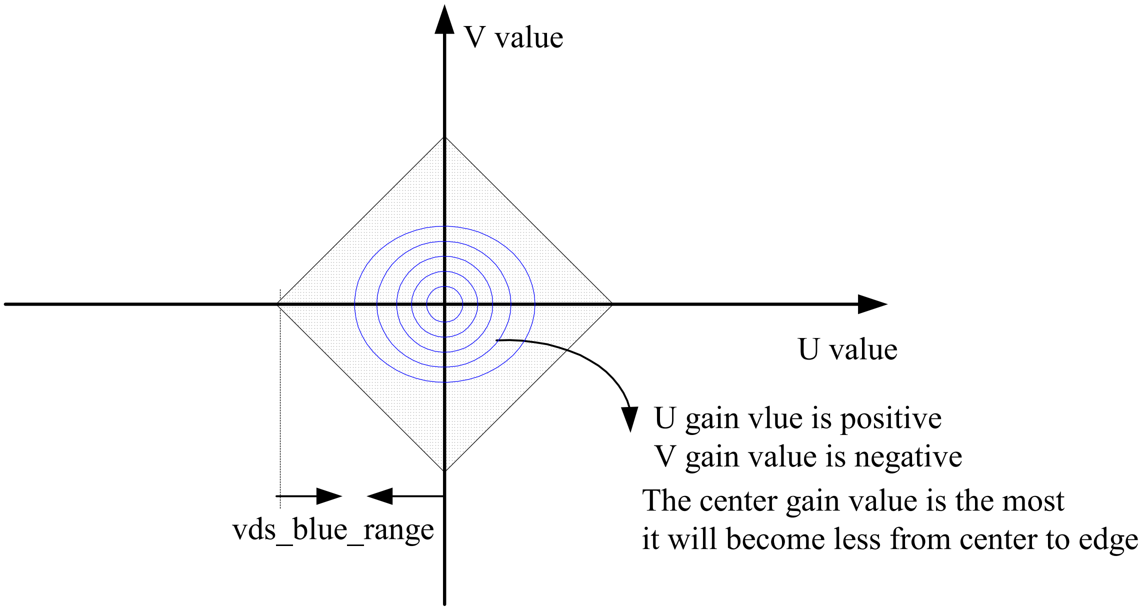
Vds_c2_u_high [7:0]	Reg_S3_6b [4:0] Reg_S3_6a [7:5]	Color enhance2 u high value
Vds_c2_byps	Reg_S3_6b [5]	Color enhance2 bypass control
Vds_c2_y_threshold [7:0]	Reg_6c [5:0] Reg_6b [7:6]	Color enhance2 y threshold value

9.4.4.4 Blue stretch

Blue stretch is used to enhance the blue color, it define a diamond area based on UV value. In the Diamond, the blue color will be enhanced; out of the diamond, the blue color will keep original.

At the same time, we can set a Y range to control blue stretch area, when the input within the Y range, do blue stretch; when the input out of the range, keep original.

Note: the u/v gain value is change smoothly to avoid the saturation impulse artifact.



Reference Register Map

Register Name	Register Map	Register description
Vds_blue_range [2:0]	Reg_S3_73 [2:0]	Blue stretch range select (2 ^ range)
Vds_blue_byps	Reg_S3_73 [3]	Blue stretch range bypass control
Vds_blue_ugain [3:0]	Reg_S3_73 [7:4]	Blue stretch u gain control
Vds_blue_vgain [3:0]	Reg_S3_74 [3:0]	Blue stretch v gain control
Vds_blue_y_lev [3:0]	Reg_S3_74 [7:4]	Blue stretch y level control (reg*16 + 15)

9.4.5 Dynamic range expansion

This process is input data multiply a gain and add an offset. As the following formula:

Y_{out} = Yin * Y_gain + Y_offset;
U_{out} = (Uin * U_cos_gain + Vin * U_sin_gain) + U_offset;
V_{out} = (Vin * V_cos_gain + Uin * V_sin_gain) + V_offset;

The parameters range:

Y_gain:	0 ~ 2x	(255 is the max gain 2x, bin)
Y_offset:	-128 ~ 127	(highest bit is sign bit, 2’s)
U_cos_gain:	-4x ~ 4x	(highest bit is sign bit, 2’s)
V_sin_gain:	-4x ~ 4x	(highest bit is sign bit, 2’s)
U_offset:	-128 ~127	(highest bit is sign bit, 2’s)
V_cos_gain:	-4x ~ 4x	(highest bit is sign bit, 2’s)
U_sin_gain:	-4x ~ 4x	(highest bit is sign bit, 2’s)
V_offset:	-128 ~127	(highest bit is sign bit, 2’s)

It includes several functions, when program its registers, we should consider the following aspects:

9.4.5.1 For color space conversion.

Assume the actual YUV to RGB convert formula is (the reverse formula of RGB to YUV conversion):
R = a*(Y +b) + c*(V +d)
B = a*(Y +b) + e*(U +f)
(U, V range is –128 ~ +127)

Then the dynamic range program should like this:

Y_gain_cc [7:0]	= a * 128
Y_offset_cc [7:0]	= a*b
U_gain_cc [7:0]	= (32*e)/2.032
U_offset_cc [7:0]	= (e*f)/2.032
V_gain_cc [7:0]	= (32*c)/1.14
V_offset_cc [7:0]	= (c*d)/1.14

The upper programming value is used for matrix conversion, if we need tune the brightness, contrast and saturation, we should tune base on this value.

For example:
Input RGB to YUV formula (YCbCr) is:
Y = 0.257*R + 0.504*G + 0.098*B + 16
Cb = - 0.14*R – 0.291*G +0.439*B + 128
Cr = 0.439*R – 0.368*G – 0.071*B + 128
(This Cb and Cr are unsigned data)
Then its reverse YUV to RGB formula should be:
R = 1.164*(Y-16) + 1.596*(Cr-128)
G = 1.164*(Y-16) – 0.813*(Cr-128) – 0.392*(Cb-128)
B = 1.164*(Y-16) + 2.017*(Cb-128)

So, a=1.164, b=-16, c=1.596, d=0, e=2.017, f=0
And we should program the dynamic range like this:

Y_gain_cc	=1.164*128
Y_offset_cc	=1.164*(-16)
U_gain_cc	=32*2.017/2.032
U_offset_cc	=0
V_gain_cc	=32*1.596/1.14
V_offset_cc	=0

9.4.5.2 For Y contrast control

When tune the picture's contrast, the y_gain should be changed, and the final Y gain should be:

Y_gain = y_gain_cc * y_contrast.

For example:

Y contrast is 1.1, and then the final Y gain is:

Y_gain = y_gain_cc * y_contrast = 1.1* y_gain_cc (for color space conversion)

9.4.5.3 For Y brightness control

When tune the picture's brightness, the y_offset should be changed, and the final Y offset should be:

Y_offset = y_brightness + y_offset_cc (for color space conversion)

9.4.5.4 For saturation control

When tune the UV's saturation, the UV gain should be changed, and the UV gain value after saturation tune.

U_gain1 = u_gain_cc * UV_saturation

V_gain1 = v_gain_cc * UV_saturation

9.4.5.5 For Hue control

The hue adjustment formula is:

U' = U*cos(x) + V*sin(x)

V' = V*cos(x) – U*sin(x)

When tune the Hue, the UV gain should be changed, and the final UV gain value after hue tune are (assume the hue adjustment angle is x):

U_cos_gain = u_gain1 * cos(x)

U_sin_gain = u_gain1 * sin(x)

V_cos_gain = v_gain1 * cos(x)

V_sin_gain = v_gain1 * (-sin(x))

Conculsion:

Y_gain = y_gain_cc * y_contrast

Y_offset = y_offset_cc + y_brightness

U_cos_gain = u_gain_cc * UV_saturation * cos (x)

U_sin_gain = u_gain_cc * UV_saturation * sin (x)

U_offset = u_offset_cc

V_cos_gain = v_gain_cc * UV_saturation * cos(x)

V_sin_gain = v_gain_cc * UV_saturation * (-sin(x))

V_offset = v_offset_cc

Reference Registers Map:

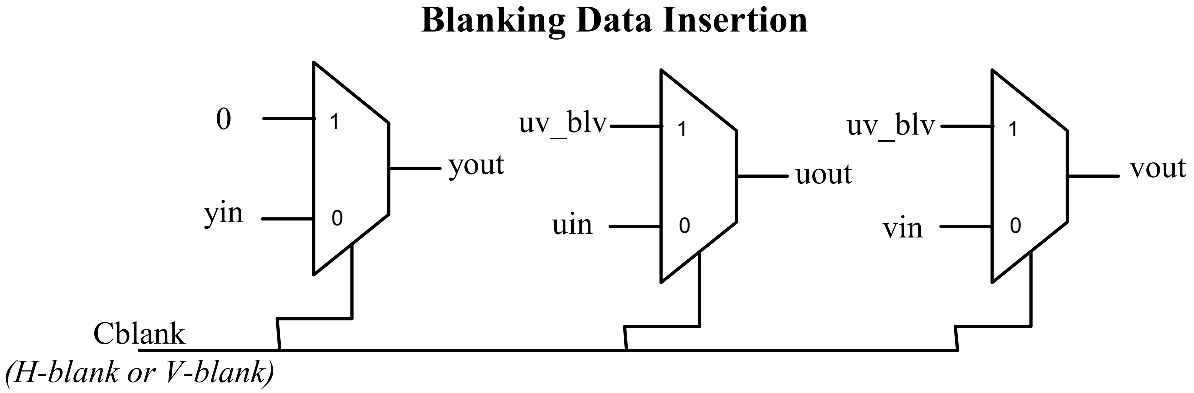
Register Name	Register Map	Register description	Range
Vds_y_gain [7:0]	Reg_S3_35 [7:0]	Y dynamic range gain value.	(0 ~ 2) * 128
Vds_ucos_gain [7:0]	Reg_S3_36 [7:0]	U dynamic range cos gain value.	(-4~4) * 32
Vds_vcos_gain [7:0]	Reg_S3_37 [7:0]	V dynamic range cos gain value.	-(-4~4) * 32
Vds_usin_gain [7:0]	Reg_S3_38 [7:0]	U dynamic range sin gain value.	(-4~4) * 32
Vds_vsin_gain [7:0]	Reg_S3_39 [7:0]	V dynamic range sin gain value.	-(-4~4) * 32
Vds_y_ofst [7:0]	Reg_S3_3a [7:0]	Y dynamic range offset	-128 ~ 127

Vds_u_ofst [7:0]	Reg_S3_3b [7:0]	U dynamic range offset	-128 ~ 127
Vds_v_ofst [7:0]	Reg_S3_3c [7:0]	V dynamic range offset	-128 ~ 127
Vd_dyn_byps	Reg_S3_3e [4]	=1, Bypass dynamic range	

9.4.6 Blanking and sync insertion

For TrueView 5725, there are 2 parts function, blanking data insertion and sync-level insertion:

9.4.6.1 Blanking data insertion

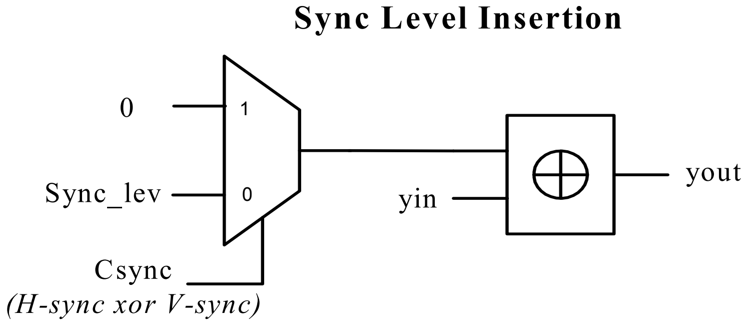


When H/V blanking period, we will insert blanking data, for Y data, we will insert to zero, for U and V data, we will insert the programmable register value to make it flexible,

For example: we want UV blanking level set to 0.6V (total is 1V)

UV_BLEV = 512 * (0.6 – 0.5) = 51 = 0x33

9.4.6.2 Sync level insertion



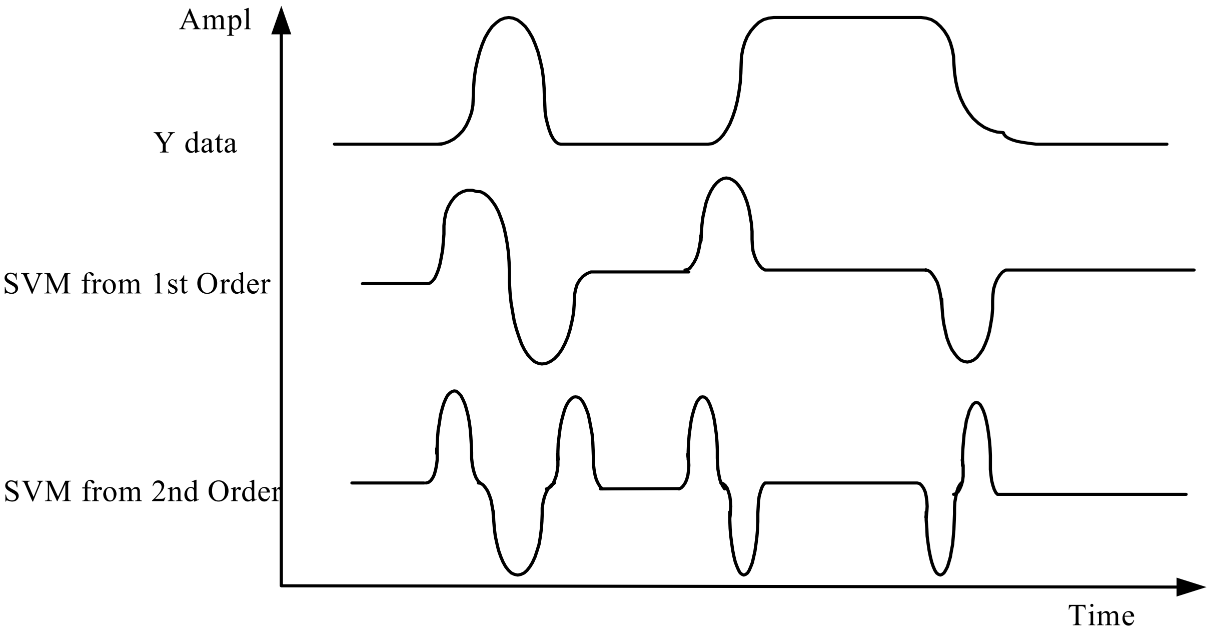
If the output is YPbPr, and sync on Y, we should add sync amplitude on Y.
For example: if the sync amplitude is 0.3V (total 1V), then, sync level = 1023* (0.3/1) =306=0x132(hex).

Reference Registers Map:

Register Name	Register Map	Register description
Vds_uv_blk_val [7:0]	Reg_S3_3f [7:0]	UV blanking level value
Vds_sync_lev [7:0]	Reg_S3_3e [0] Reg_S3_3d [7:0]	Sync signal level value

9.4.7 SVM (Scan Velocity Modulation) generation

Picture tubes equipped with an appropriate yoke can use the Scan Velocity Modulation signal to vary the speed of the electron gun during the entire video scan line dependent upon its content. Transitions from dark to bright will first speed up and then slow down the scan; vice versa for the opposite transition.
The signal delay is adjustable by -6 ~ 1.5 video clocks in respect to the analog RGB output signals.
This is useful to match the different group delay of analog RGB amplifiers to the one for the SVM yoke current.



SVM Signal Waveform

Reference Registers Map:

Register Name	Register Map	Register description
Vds_svm_bpf_cntrl [1:0]	Reg_S3_32 [1:0]	SVM data generation selection control 2'b00: SVM data = a0 – a4 2'b01: SVM data = a1 – a4 2'b10: SVM data = a2 – a4 2'b11: SVM data = a3 – a4
Vds_svm_vclk_delay [1:0]	Reg_S3_32 [5:4]	SVM data delay by VCLK control 2'b00: Delay 1 VCLK 2'b01: Delay 2 VCLK 2'b10: Delay 3 VCLK 2'b11: Delay 4 VCLK
Vds_svm_v4clk_delay [1:0]	Reg_S3_40 [5:4]	SVM data delay from 1 to 4 V4CLKs 2'b00: Delay 1 V2CLK 2'b01: Delay 2 V2CLK 2'b10: Delay 3 V2CLK 2'b11: Delay 4 V2CLK
Vds_svm_gain [7:0]	Reg_S3_33 [7:0]	SVM data gain value, its range is (0 ~ 16) *16

Vds_svm_offset [7:0]	Reg_S3_34 [7:0]	SVM data offset value, its range is 0 ~ 255
Vds_svm_2nd_byps	Reg_S3_32 [3]	Bypass 2nd order signal generation
Vds_svm_pol_flip	Reg_S3_32 [2]	SVM polarity flip control bit, set 1 to flip SVM Signal
Vds_svm_sigmoid-byps	Reg_S3_32 [7]	Bypass the sigmoid function; make the SVM signal shaper.

9.4.8 YUV to RGB color space conversion

The convert matrix is:

$R = Y + 1.14*V$
$G = Y – 0.394*U – 0.581*V$
$B = Y + 2.032*U$

9.4.9 Interpolation

5725 vds_proc has 1st stage, 2nd stage interpolation for VCLK, V2CLK and V4CLK clock domain conversion:

- When V2CLK = 2 * VCLK, enable vds_proc first interpolation, else, bypass it.
- When V4CLK = 2 * V2CLK, enable vds_proc second interpolation, else, bypass it.

Reference Registers Map:

Register Name	Register Map	Register description

Vds_1st_int_byps	Reg_S3_40 [0]	=1, 1st stage 2x interpolation bypass;
		=0, 1st stage 2x interpolation enable;

Vds_2nd_int_byps	Reg_S3_40 [1]	=1, 2nd stage 2x interpolation bypass;
		=0, 2nd stage 2x interpolation enable;

10 Host Interface

True View 5725 uses two-wire serial bus interface.
Note: Only salve mode is supported. And maximum speed is 400KBits/sec (fast mode).

10.1 I2C Slave Address Selection

5725 host interface support two slave address AE and 2E, which is selected by pin43 SCLSA.

- When pin43 add external pull-down (Use 10K pull down resistor), 5725 slave address is 2EH.
- When there is no external pull-down, 5725 slave address is AEH. (Pin43 has internal pull-up).

And pin43 also shares with GPIO bit2. Detail description is in **13.1.3 GPIO bit2**

10.2 I2C writing

When writing to 5725, the slave address is **AE/2E**H. A control sequence as following:

- ***Write to One Control Register***
  Start Signal
  Slave Address Byte (R/*W* Bit = Low)
  Base Address Byte
  Data Byte to Base Address
  Stop Signal

- ***Write to Consecutive Control Registers***
  Start Signal
  Slave Address Byte (R/*W* Bit = Low)
  Base Address Byte
  Data Byte to Base Address
  Data Byte to (Base Address + 1)
  Data Byte to (Base Address + 2)
  Data Byte to (Base Address + 3)
  ………………
  Stop Signal

10.3 I2C Reading

When **reading** from 5725, the slave address is **AF/2F**H. A control sequence as following:

- ***Read from One Control Register***
  Start Signal
  Slave Address Byte (R/*W* Bit = Low)
  Base Address Byte
  Start Signal
  Slave Address Byte (R/*W* Bit = High)
  Data Byte from Base Address
  Stop Signal

- ***Read from Four Consecutive Control Registers***
  Start Signal
  Slave Address Byte (R/*W* Bit = Low)
  Base Address Byte
  Start Signal
  Slave Address Byte (R/*W* Bit = High)
  Data Byte from Base Address
  Data Byte from (Base Address + 1)
  Data Byte from (Base Address + 2)
  Data Byte from (Base Address + 3)
  ……………………….
  Stop Signal

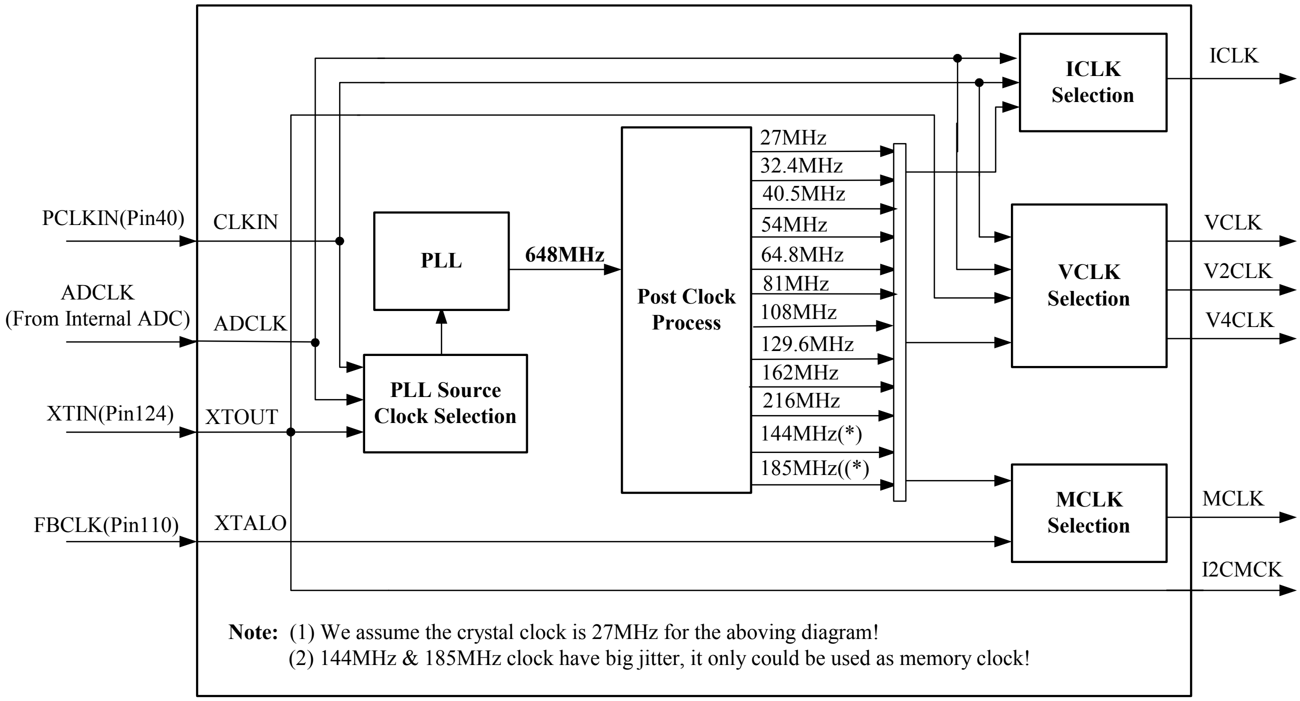
10.4 Chip ID Check

5725 chips ID is read only register from Reg_S0_0B ~ Reg_S0_0D, you can read back them and confirm it:

- Reg_S0_0B: A3H
- Reg_S0_0C: 1AH
- Reg_S0_0D: 01H

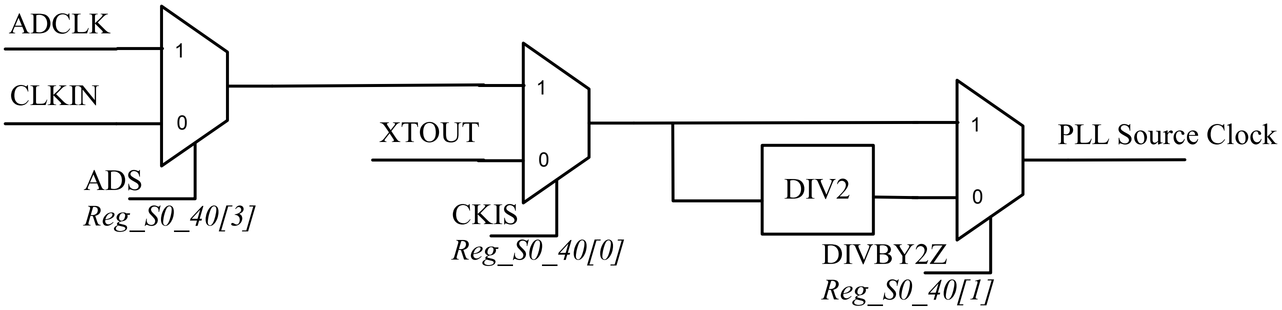
11 PLL648

5725 built-in PLL is used to generate all internal digital clocks as well as display clock. The following is a block diagram of the PLL648:



Note: (1) We assume the crystal clock is 27MHz for the aboving diagram!
(2) 144MHz & 185MHz clock have big jitter, it only could be used as memory clock!

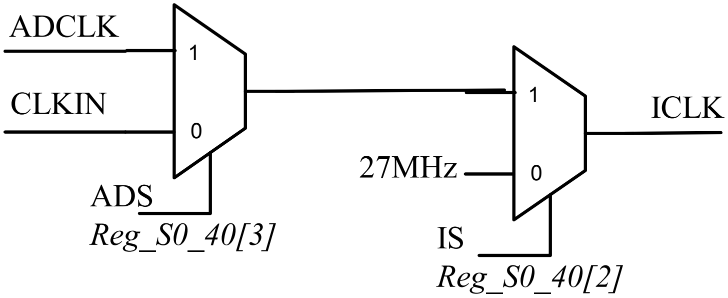
11.1 PLL Source Clock Selection



PLL source clock is used to generate the 648MHz clock, so we must guarantee the clock is stable, little jitter. At the most case, we will select the crystal/oscillator clock as PLL clock source.
We can select the ADCLK or CLKIN as PLL clock source for testing and special case (e.g. sync lock mode).

11.2 ICLK Selection

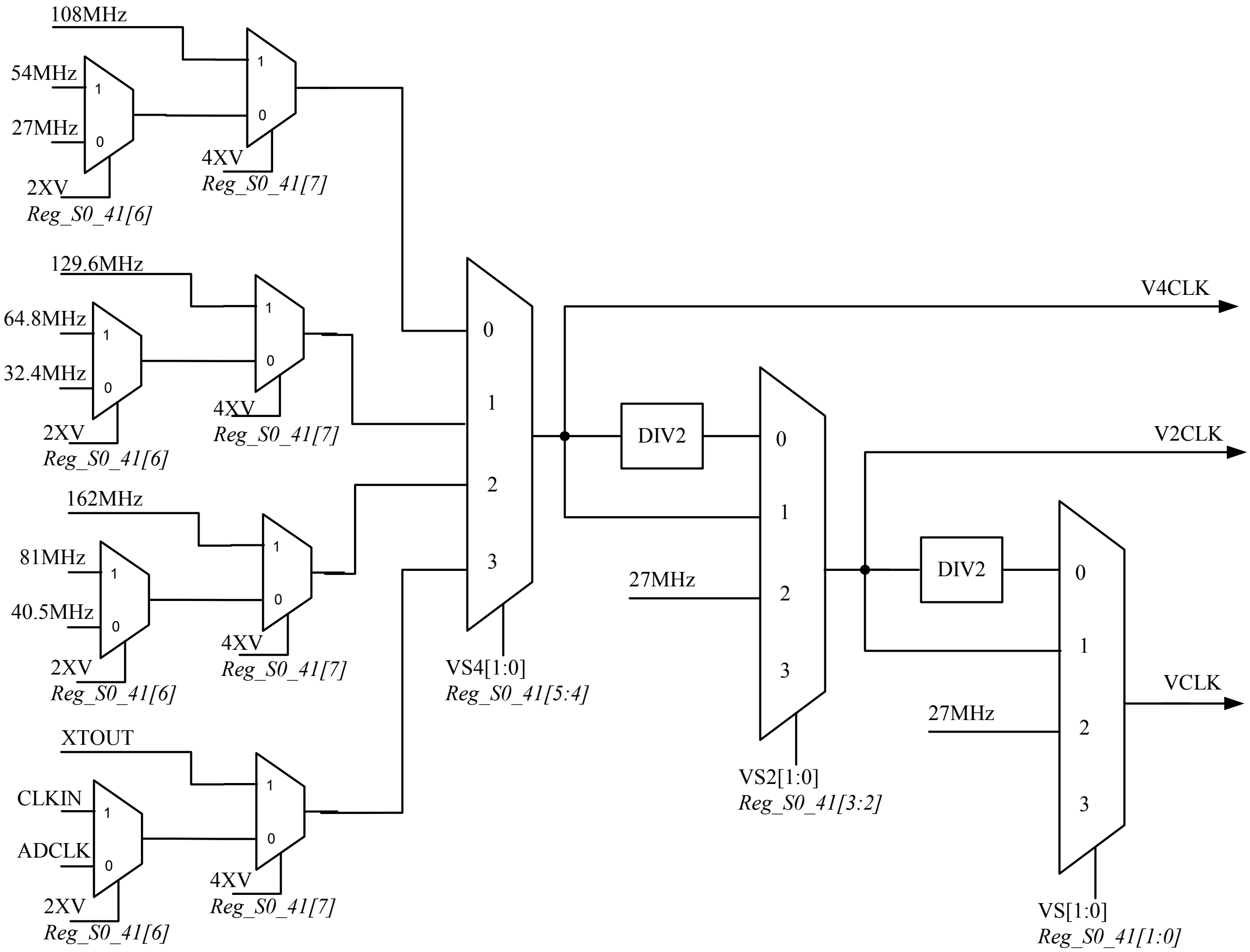
ICLK is used for input formatter, de-interlace, and so on.



ICLK will capture the input data and do de-interlace, so we must keep the ICLK in phase will the input data, if the data from ADC, we should select the ADCLK as ICLK, if the data from digital decoder, we will select the CLKIN from PAD as ICLK, we can select 27MHz clock from PLL648.

11.3 VCLK Selection

VCLK/V2CLK/V4CLK is used for vds_proc as display clock;



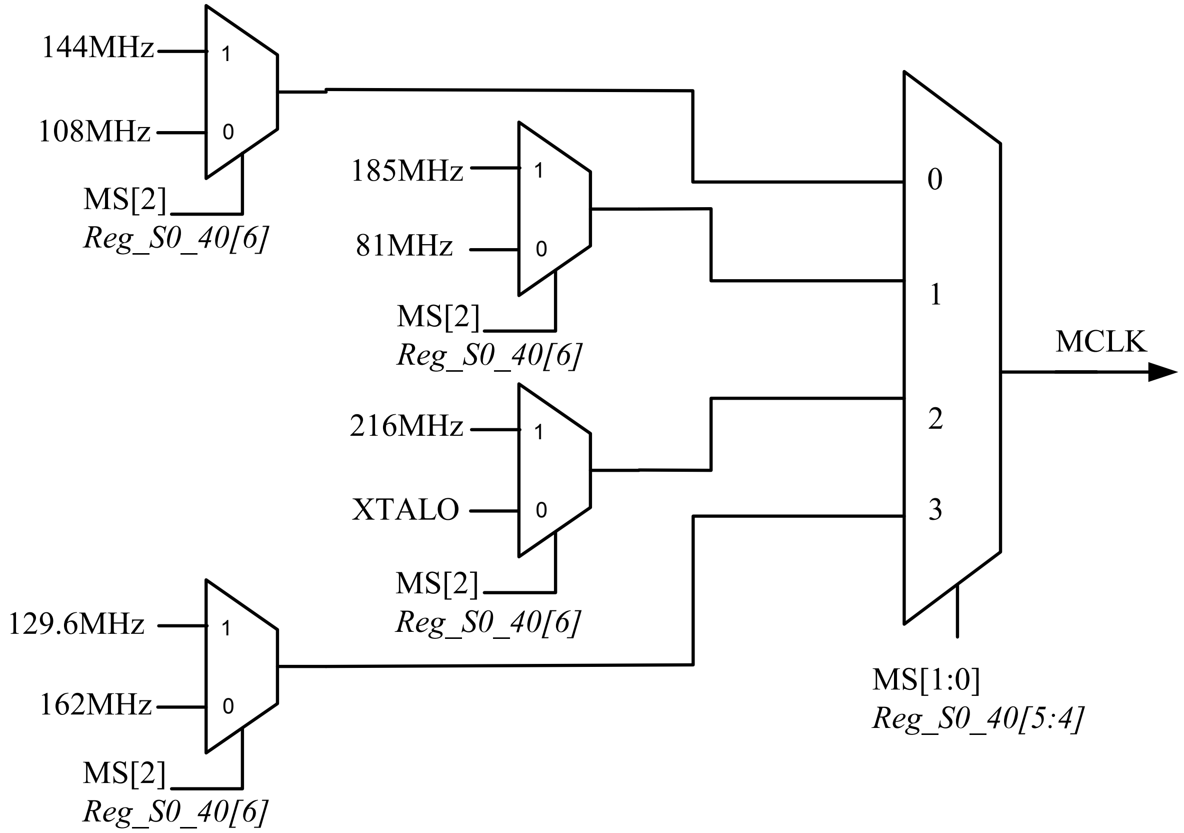
TrueView 5725 VCLK Selection Table

Register Settings					**Clock Frequency**			
PLL_4XV	PLL_2XV	PLL_VS4	PLL_VS2	PLL_VS	VCLK	V2CLK	V4CLK	Note
1'b1	X	2'b00	2'b00	2'b00	27MHz	54MHz	108MHz	*
1'b0	1'b1	2'b00	2'b01	2'b00	27MHz	54MHz	54MHz	*
1'b0	1'b0	2'b00	2'b01	2'b01	27MHz	27MHz	27MHz	*
1'b1	X	2'b01	2'b00	2'b00	32.4MHz	64.8MHz	129.6MHz	
1'b0	1'b1	2'b01	2'b01	2'b00	32.4MHz	64.8MHz	64.8MHz	
1'b0	1'b0	2'b01	2'b01	2'b01	32.4MHz	32.4MHz	32.4MHz	
1'b1	X	2'b10	2'b00	2'b00	40.5MHz	81MHz	162MHz	
1'b0	1'b1	2'b10	2'b01	2'b00	40.5MHz	81MHz	81MHz	
1'b0	1'b0	2'b10	2'b01	2'b01	40.5MHz	40.5MHz	40.5MHz	
1'b1	X	2'b00	2'b01	2'b00	54MHz	108MHz	108MHz	
1'b0	1'b1	2'b00	2'b01	2'b01	54MHz	54MHz	54MHz	
1'b1	X	2'b01	2'b01	2'b00	64.8MHz	129.6MHz	129.6MHz	
1'b0	1'b1	2'b01	2'b01	2'b01	64.8MHz	64.8MHz	64.8MHz	
1'b1	X	2'b10	2'b01	2'b00	81MHz	162MHz	162MHz	
1'b0	1'b1	2'b10	2'b01	2'b01	81MHz	81MHz	81MHz	
1'b1	X	2'b00	2'b01	2'b01	108MHz	108MHz	108MHz	
1'b1	X	2'b01	2'b01	2'b01	129.6MHz	129.6MHz	129.6MHz	
1'b1	X	2'b10	2'b01	2'b01	162MHz	162MHz	162MHz	
1'b1	X	2'b11	2'b01	2'b01	XTOUT	XTOUT	XTOUT	
1'b0	1'b1	2'b11	2'b01	2'b01	CLKIN	CLKIN	CLKIN	
1'b0	1'b0	2'b11	2'b01	2'b01	ADCLK	ADCLK	ADCLK	

Note: 1) When V2CLK need 27MHz to be in phase with CLKIN, set PLL_VS2 [1:0]=2'b10;
2) When VCLK need 27MHz to be in phase with CLKIN, set PLL_VS [1:0]=2'b10;

11.4 MCLK Selection

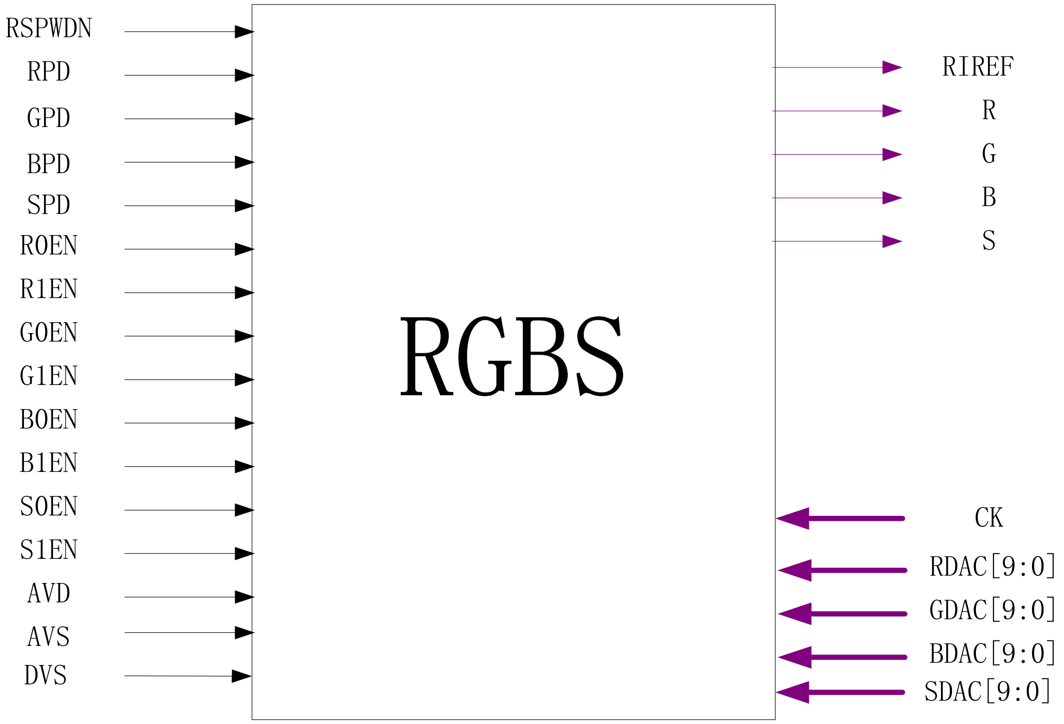
MCLK is used for memory controller, WFF/RFF, Capture and playback! We select the MCLK for memory bandwidth and external memory chip!



Register Setting	MCLK Frequency
MS [2:0]: Reg_S0_40 [6:4]	MCLK
3’b000	108MHz
3’b001	81MHz
3’b010	XTALO
3’b011	162MHz
3’b100	144MHz
3’b101	185MHz
3’b110	216MHz
3’b111	129.6MHz

12 Digital to Analog Conversion

Here is 5725 RGBS DAC interface:



12.1 DAC DC Characteristics

Parameter	Min	Typical	Max	Unit	Notes
DAC Resolution/Channel			10	Bits	
The number of channels	4				
INL	-1.0		+1.0	LSB	
DNL	-1.0		+1.0	LSB	
Full Scale (gain) Error	-5.0		+10.0	%	Of Full Scale
DAC Full Scale Voltage	665	700	770	mV	1, 2
DAC current		85		mA	1
DAC to DAC Matching			6	%	
LSB Current		18.2		uA	1
Monotonicity	Guaranteed				

NOTES:

1 For VESA Video Levels, the R_{iref} in the board is 160ohms. The output load is double terminated 75ohms ((37.5ohm) and 10pf per channel. Full-scale for all four channels is 0.7V.

2. For good linearity, the full-scale voltage should be less than 1V when the output load is 75ohms and 10pf per channel.

12.2 DAC AC Characteristics

Parameter	Min	Typical	Max	Unit	Notes
Pixel Clock Rate			162	MHz	
RGB Video Output Rise Time (10-90% of full-scale)		2.6	4.7	ns	1
RGB Video Output Fall Time (10-90% of full-scale)		2.5	4.0	ns	1

NOTES:

1. As measured with 37.5ohm and 10pf load.

12.3 IREF resistor formula:

The max output voltage of the RGBS is:

Vout=1.25*2046*Rout/(864*Riref)

The Rout is the loading of the R, G, B and S output. The Riref is the resister connects to the pin IREF in the 5725.

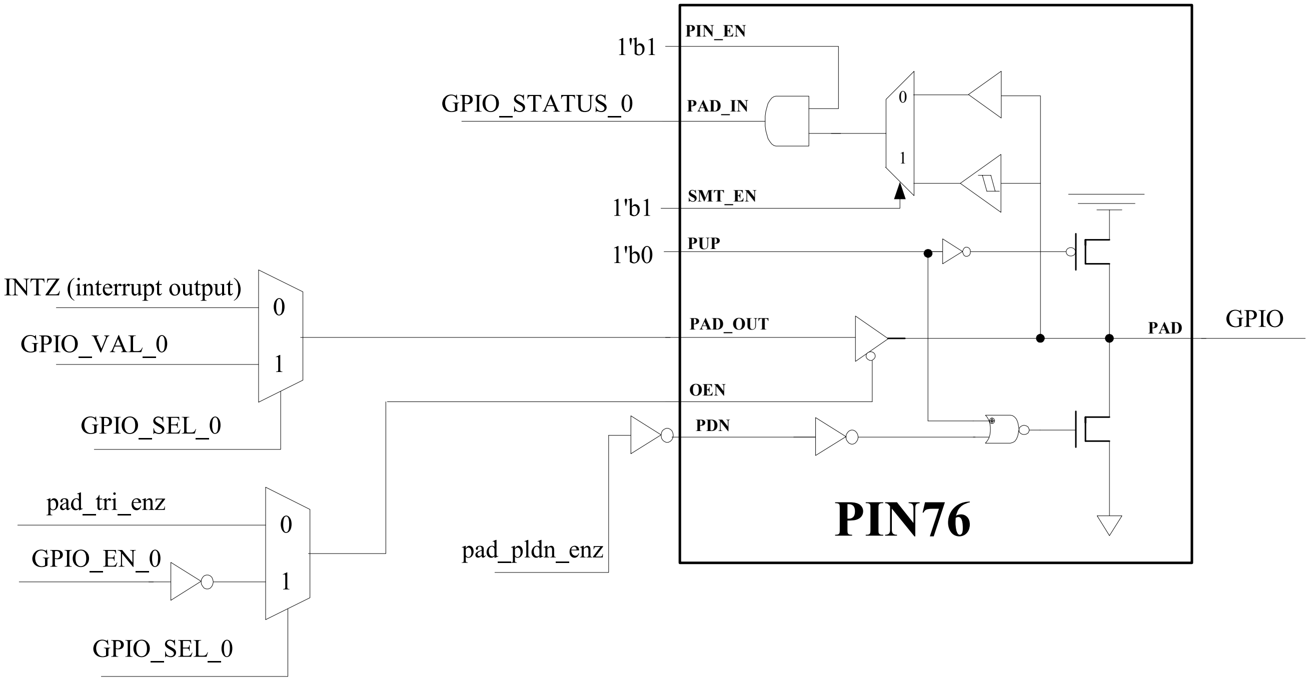
13 Miscellaneous

13.1 GPIO

TureView 5725 has total 8 GPIO pins. They are shared with other Pins. All of them could be set to output control or input status.

13.1.1 GPIO Bit0

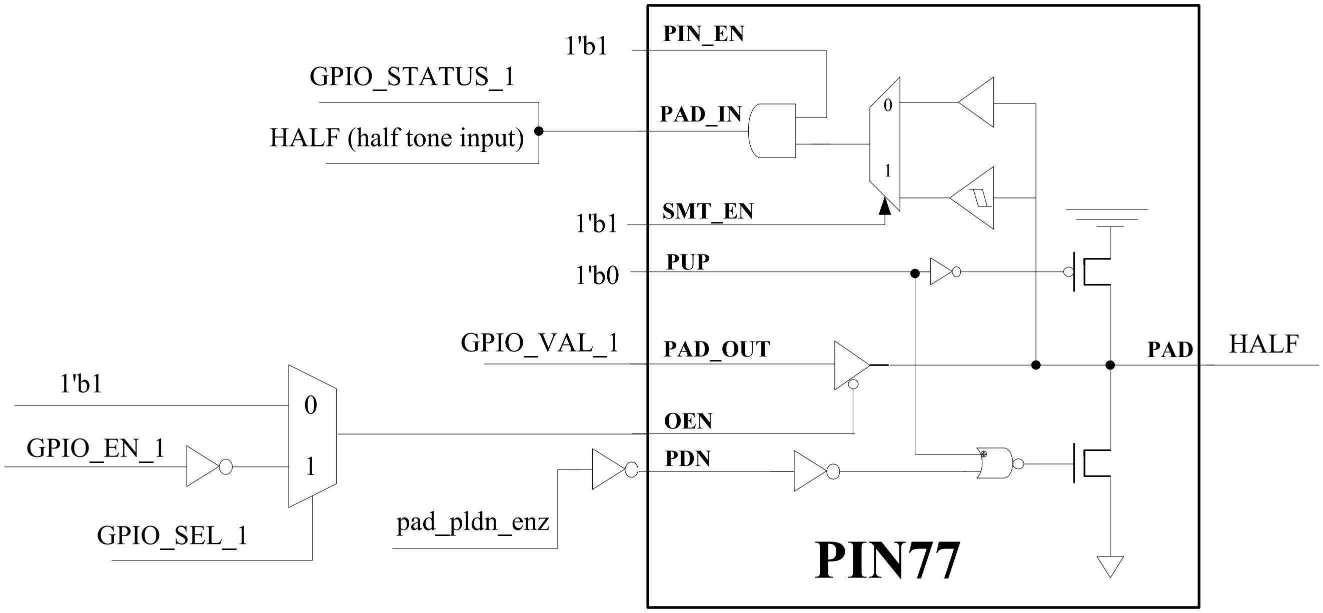
GPIO bit 0 is in Pin76; it shared with INTZ (Interrupt output).
Below is the GPIO_0 control diagram:



Control Name	Address	Description
GPIO_STATUS_0	Reg_S0_0E [0]	Read only registers for GPIO Bit0 Status
GPIO_SEL_0	Reg_S0_52 [0]	=1, enable Pin76 as GPIO bit0; =0, enable Pin76 as Interrupt output.
GPIO_EN_0	Reg_S0_53 [0]	=1, enable GPIO bit0 output; =0, disable GPIO bit0 output.
GPIO_VAL_0	Reg_S0_54 [0]	GPIO bit0 output value.
Pad_tri_enz	Reg_S0_49 [4]	=0, enable output tri-state gate; =1, enable output.
Pad_pldn_enz	Reg_S0_49 [5]	=0, enable pull down transistor; =1, disable pull down transistor.

13.1.2 GPIO Bit1

GPIO bit1 is Pin77; it shared with half tone input.
Below is the GPIO_1 control diagram:

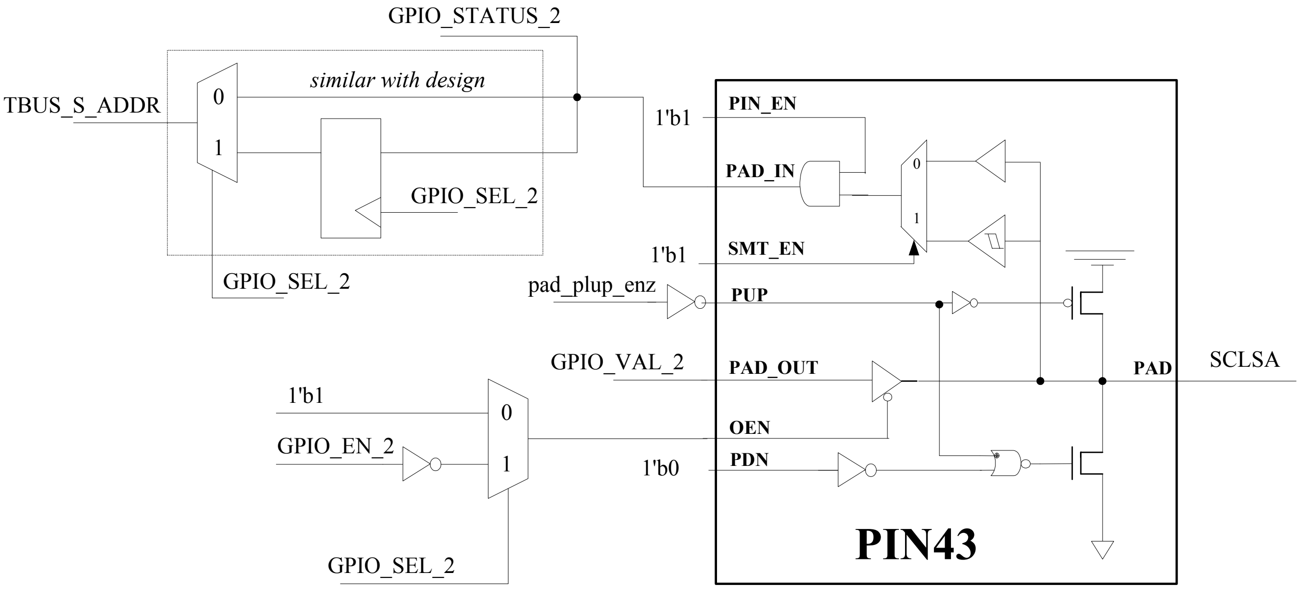


When pin90 is used as GPIO port, half tone enable bit (“**vds_half_en**”) must be set to zero.
When pin90 is used as HALF Tone input, half tone enable bit (“**vds_half_en**”) must be set to one.

Control Name	Address	Description
GPIO_STATUS_1	Reg_S0_0E [1]	Read only registers for GPIO Bit1 Status
GPIO_SEL_1	Reg_S0_52 [1]	=1, enable Pin77 as GPIO bit1; =0, enable Pin77 as half tone input.
GPIO_EN_1	Reg_S0_53 [1]	=1, enable GPIO bit1 output; =0, disable GPIO bit1 output.
GPIO_VAL_1	Reg_S0_54 [1]	GPIO bit1 output value.
Pad_pldn_enz	Reg_S0_49 [5]	=0, enable pull down transistor; =1, disable pull down transistor.
Vds_half_en	Reg_S3_00 [6]	=1, enable half tone input; =0, disable half tone input;

13.1.3 GPIO Bit2

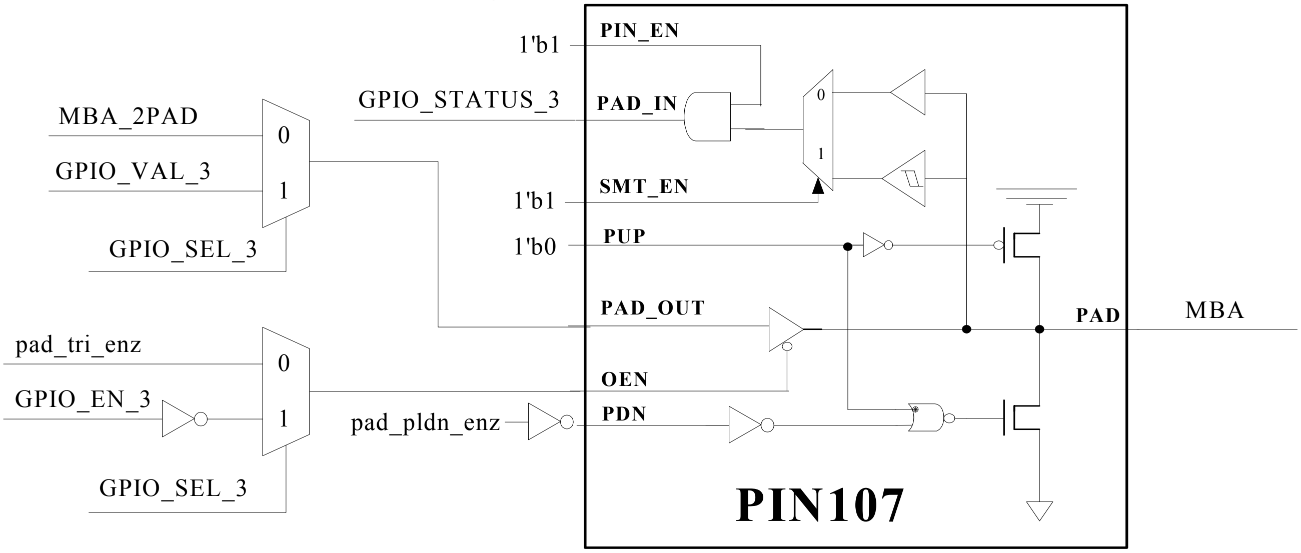
GPIO bit2 is in pin43, it shared with I2C slave address selection. I2C slave address selection is descripted in 10 Host Interface.
Below is the GPIO_2 control diagram:



Control Name	Address	Description
GPIO_STATUS_2	Reg_S0_0E [2]	Read only registers for GPIO Bit2 Status
GPIO_SEL_2	Reg_S0_52 [2]	=1, enable Pin43 as GPIO bit2; =0, enable Pin43 as I2C slave address select.
GPIO_EN_2	Reg_S0_53 [2]	=1, enable GPIO bit2 output; =0, disable GPIO bit2 output.
GPIO_VAL_2	Reg_S0_54 [2]	GPIO bit2 output value.
Pad_pldn_enz	Reg_S0_49 [5]	=0, enable pull down transistor; =1, disable pull down transistor.

13.1.4 GPIO Bit3

GPIO bit3 is in Pin107; it shared with MBA (memory bank selection).

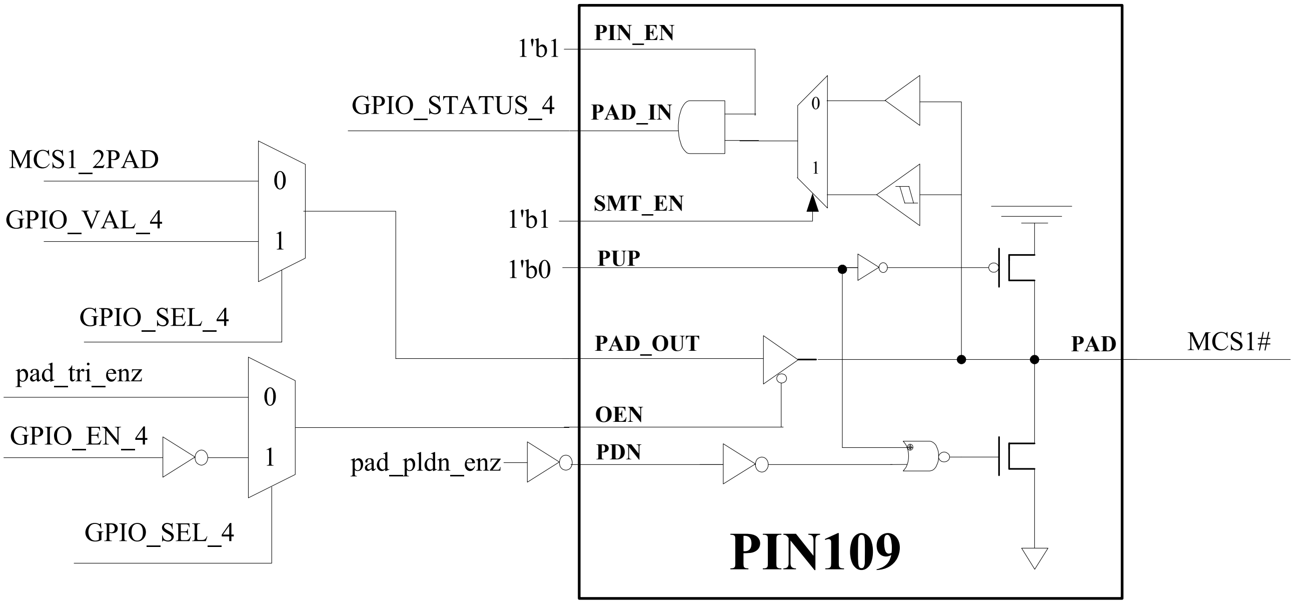


It can be used as GPIO bit3 when external memory is 8M Byte 32-bit width connected;
It must be used as Memory bank selection for other external memory connected.

Control Name	Address	Description
GPIO_STATUS_3	Reg_S0_0E [3]	Read only registers for GPIO Bit3 Status
GPIO_SEL_3	Reg_S0_52 [3]	=1, enable Pin107 as GPIO bit3; =0, enable Pin107 as memory bank select.
GPIO_EN_3	Reg_S0_53 [3]	=1, enable GPIO bit3 output; =0, disable GPIO bit3 output.
GPIO_VAL_3	Reg_S0_54 [3]	GPIO bit3 output value.
Pad_tri_enz	Reg_S0_49 [4]	=0, enable output tri-state gate; =1, enable output.
Pad_pldn_enz	Reg_S0_49 [5]	=0, enable pull down transistor; =1, disable pull down transistor.

13.1.5 GPIO Bit4

GPIO bit4 is Pin109; it shared with mcs1# (memory chip 1 select).



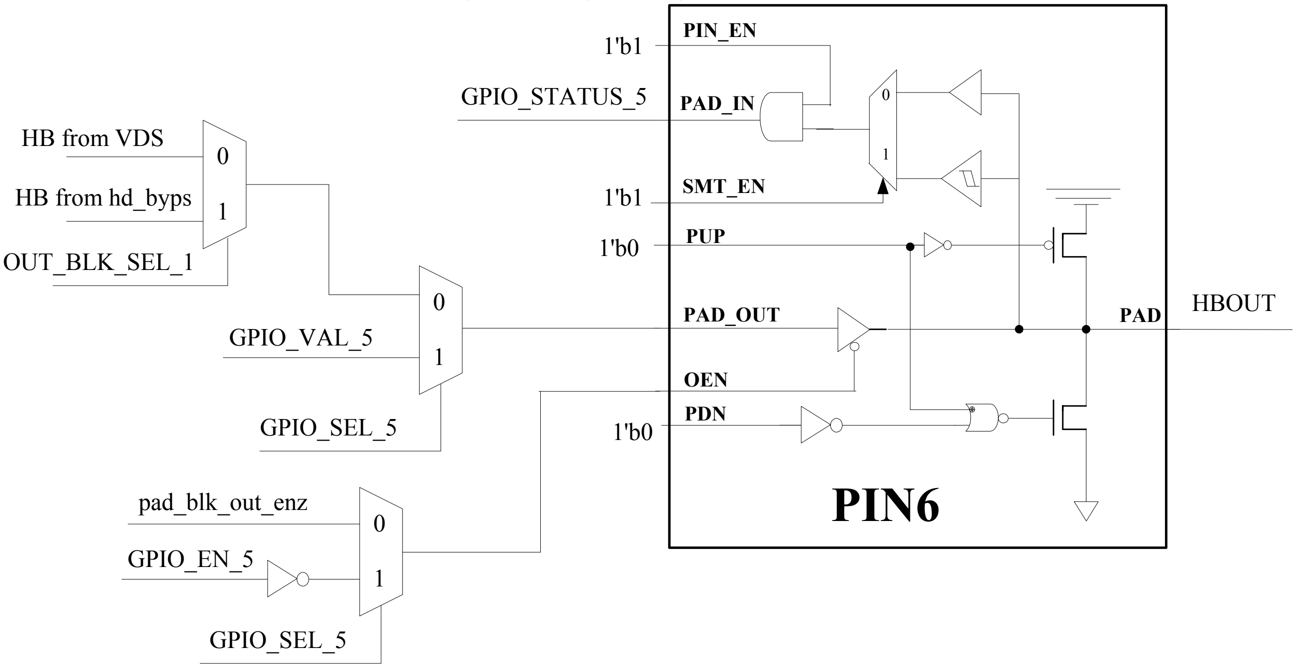
Pin109 can be used as GPIO bit4 when external 2Mbyte 16-bit width memory connected;
Pin109 must be used as memory chip1 select when other external memory connected.

Control Name	Address	Description
GPIO_STATUS_4	Reg_S0_0E [4]	Read only registers for GPIO Bit4 Status
GPIO_SEL_4	Reg_S0_52 [4]	=1, enable Pin109 as GPIO bit4; =0, enable Pin109 as memory chip1 select.
GPIO_EN_4	Reg_S0_53 [4]	=1, enable GPIO bit4 output; =0, disable GPIO bit4 output.
GPIO_VAL_4	Reg_S0_54 [4]	GPIO bit4 output value.

Pad_tri_enz	Reg_S0_49 [4]	=0, enable output tri-state gate; =1, enable output.
Pad_pldn_enz	Reg_S0_49 [5]	=0, enable pull down transistor; =1, disable pull down transistor.

13.1.6 GPIO Bit5

GPIO bit5 is Pin6; it shared with HBOUT (H-blanking output);

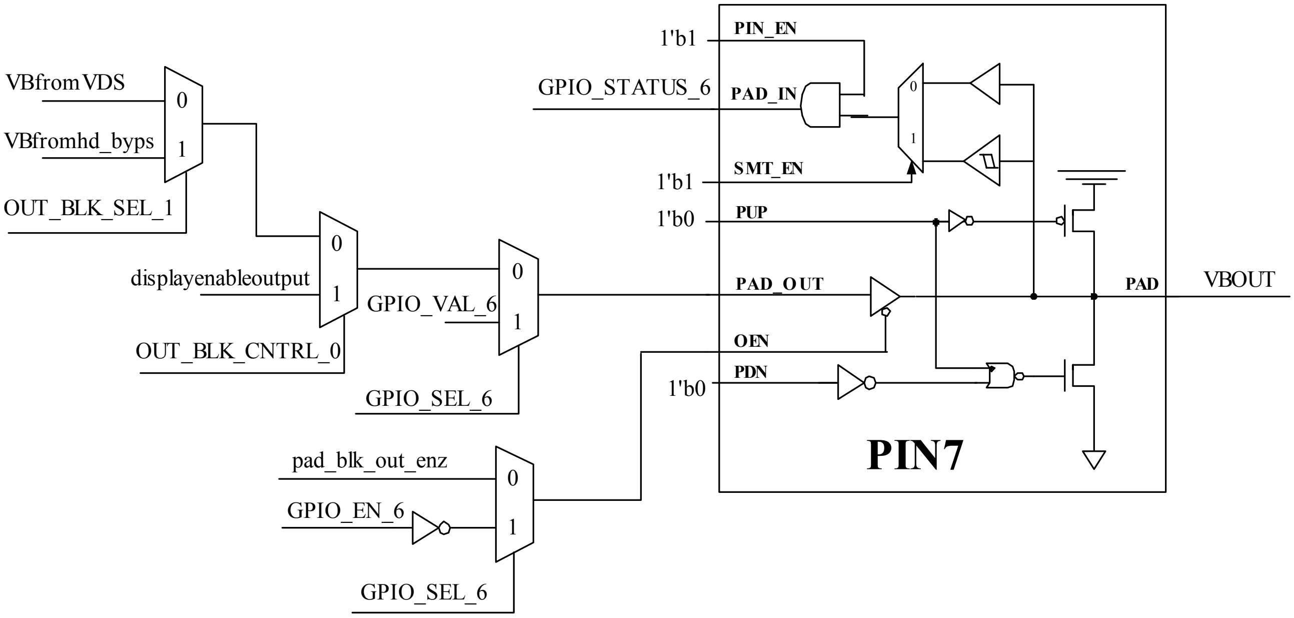


Pin6 is used as GPIO bit5 when H-blanking output is disabling.
Pin6 is used as H-blanking output when H-blanking output enable.

Control Name	Address	Description
GPIO_STATUS_5	Reg_S0_0E [5]	Read only registers for GPIO Bit5 Status
GPIO_SEL_5	Reg_S0_52 [5]	=1, enable Pin6 as GPIO bit5; =0, enable Pin6 as H-blanking output.
GPIO_EN_5	Reg_S0_53 [5]	=1, enable GPIO bit5 output; =0, disable GPIO bit5 output.
GPIO_VAL_5	Reg_S0_54 [5]	GPIO bit5 output value.
OUT_BLANK_SEL_1	Reg_S0_50 [1]	=0, hbout/vbout from vds_proc; =1, hbout/vbout from if_hd_byps.
PAD_BLK_OUT_ENZ	Reg_S0_49 [3]	=0, enable hbout/vbout output; =1, disable hbout/vbout output.

13.1.7 GPIO Bit6

GPIO bit6 is Pin7; it shared with vbout (v-blanking output or display enable output).

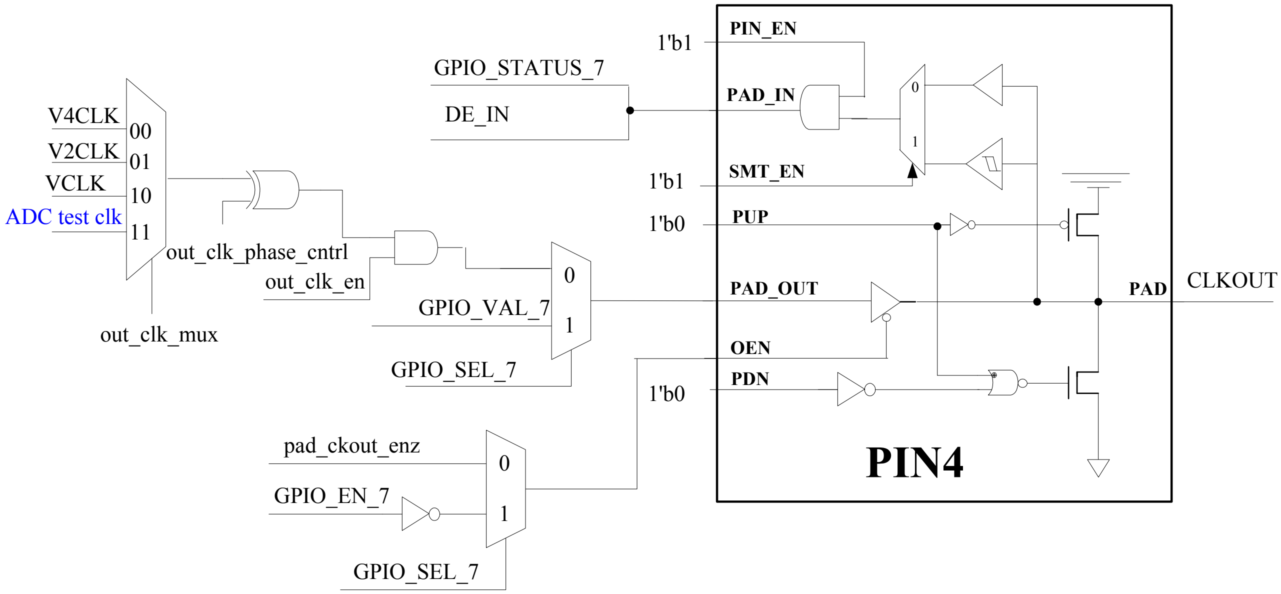


Pin7 is used as GPIO bit6 when V-blanking output and display enable output are disabling.
Pin7 is used as V-blanking output when GPIO bit6 and display output enable are disabling.
Pin7 is used as display enable output when GPIO bit6 and V-blanking output disabling.

Control Name	Address	Description
GPIO_STATUS_6	Reg_S0_0E [6]	Read only registers for GPIO Bit6 Status
GPIO_SEL_6	Reg_S0_52 [6]	=1, enable Pin7 as GPIO bit5; =0, enable Pin7 as V-blanking or display enable output .
GPIO_EN_6	Reg_S0_53 [6]	=1, enable GPIO bit6 output; =0, disable GPIO bit6 output.
GPIO_VAL_6	Reg_S0_54 [6]	GPIO bit6 output value.
OUT_BLANK_SEL_0	Reg_S0_50 [0]	=0, pin7 output v-blanking; =1, pin7 output composite DE.
OUT_BLANK_SEL_1	Reg_S0_50 [1]	=0, hbout/vbout from vds_proc; =1, hbout/vbout from if_hd_byps.
PAD_BLK_OUT_ENZ	Reg_S0_49 [3]	=0, enable hbout/vbout output; =1, disable hbout/vbout output.

13.1.8 GPIO Bit7

GPIO bit7 is Pin4; it shared with CLKOUT (clock output or display enable input);



Pin4 is used as GPIO bit7 when clock output and display enable input are disabling.
Pin4 is used as clock output when GPIO bit7 and display enable input are disabling.
Pin4 is used as display enable input when GPIO bit7 and clock output are disabling.

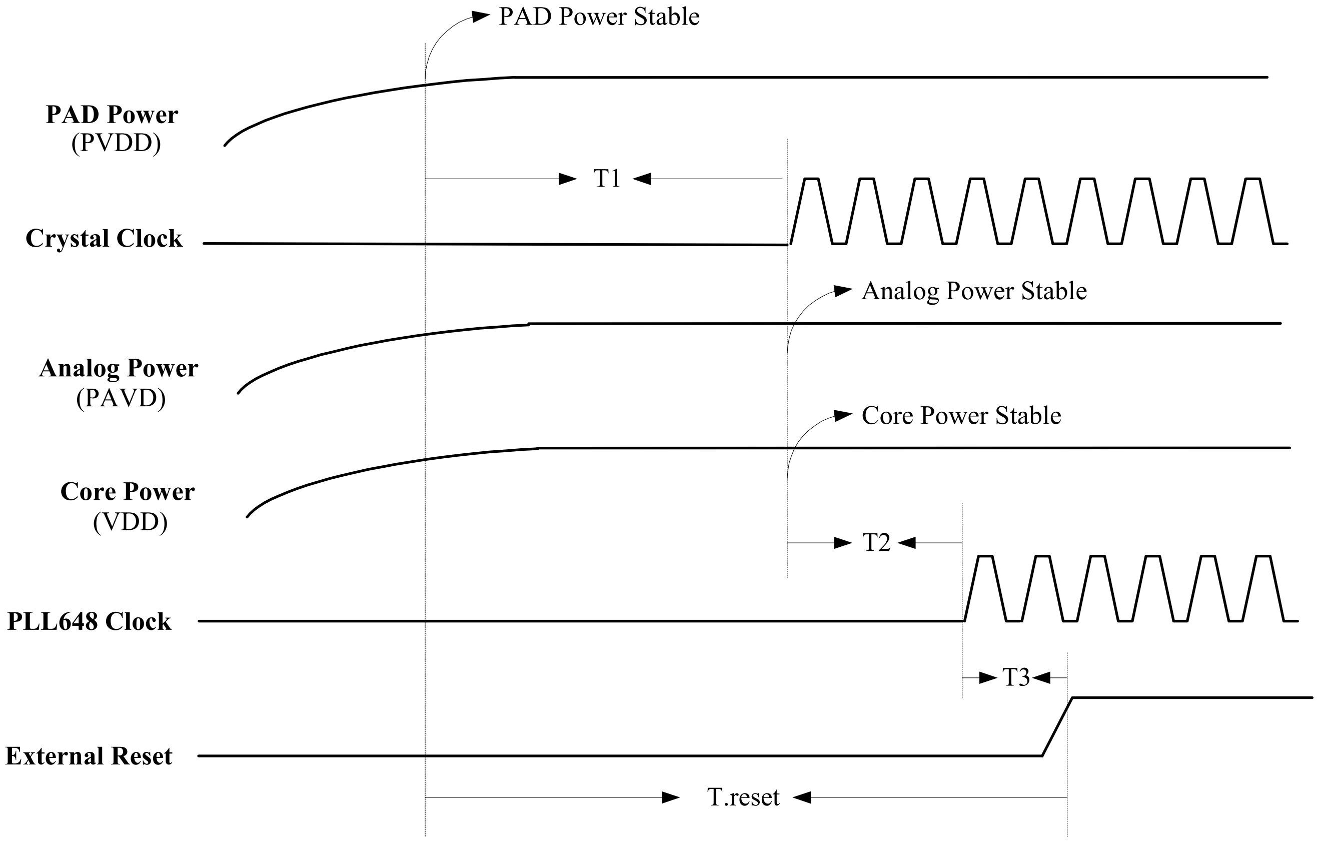
Control Name	Address	Description
GPIO_STATUS_7	Reg_S0_0E [7]	Read only registers for GPIO Bit7 Status
GPIO_SEL_7	Reg_S0_52 [7]	=1, enable Pin4 as GPIO bit7; =0, enable Pin4 as clock output or display enable input.
GPIO_EN_7	Reg_S0_53 [7]	=1, enable GPIO bit7 output; =0, disable GPIO bit7 output.
GPIO_VAL_7	Reg_S0_54 [7]	GPIO bit7 output value.
PAD_CKOUT_ENZ	Reg_S0_49 [1]	=0, enable clkout output; =1, disable clkout output.
OUT_CLK_PHASE_CNTRL	Reg_S0_4F [1]	=0, display clock output directly; =1, display clock will invert to output.
OUT_CLK_MUX	Reg_S0_4F [3:2]	=00, V4CLK output, =01, V2CLK output, =10, VCLK output, =11, ADCLK output.
OUT_CLK_EN	Reg_S0_4F [4]	=0, disable output clock to PAD; =1, enable output clock to PAD.

13.2 Reset

13.2.1 External reset

5725 external reset is Pin75. It will reset 5725 whole digital and analog parts. It should be low active

13.2.1.1 External reset timing spec:



Time Label	Meaning Description
T1	Time from PAD power stable to Crystal PAD generate stable crystal clock. It is about 20ms
T2	Time from Analog and Core power stable and crystal clock stable to PLL648 generate stable clocks, it is about 5ms
T3	Time from PLL648 generate stable clocks to reset digital logic. It is about 1ms
T.reset	Time from PAD/Analog/Core power stable to external reset de-acitve. It must be not less than 30ms.

13.2.1.2 External reset de-bounce circuit

5725 external reset implement de-bounce circuit, it can remove the glitch (less than 40 crystal clocks) in the reset Pin.

E.g. for 27MHz crystal clock, it can remove the glitch less than 1480ns.

13.2.2 Software reset

In 5725, there are also some software reset register bits, these software reset only could reset state machine and internal timing, but could not reset register bits' value. And host interface's state machine is only reset by external reset.

You can reset some module when you needn't it working to save power!

13.2.2.1 Analog Part Software Reset

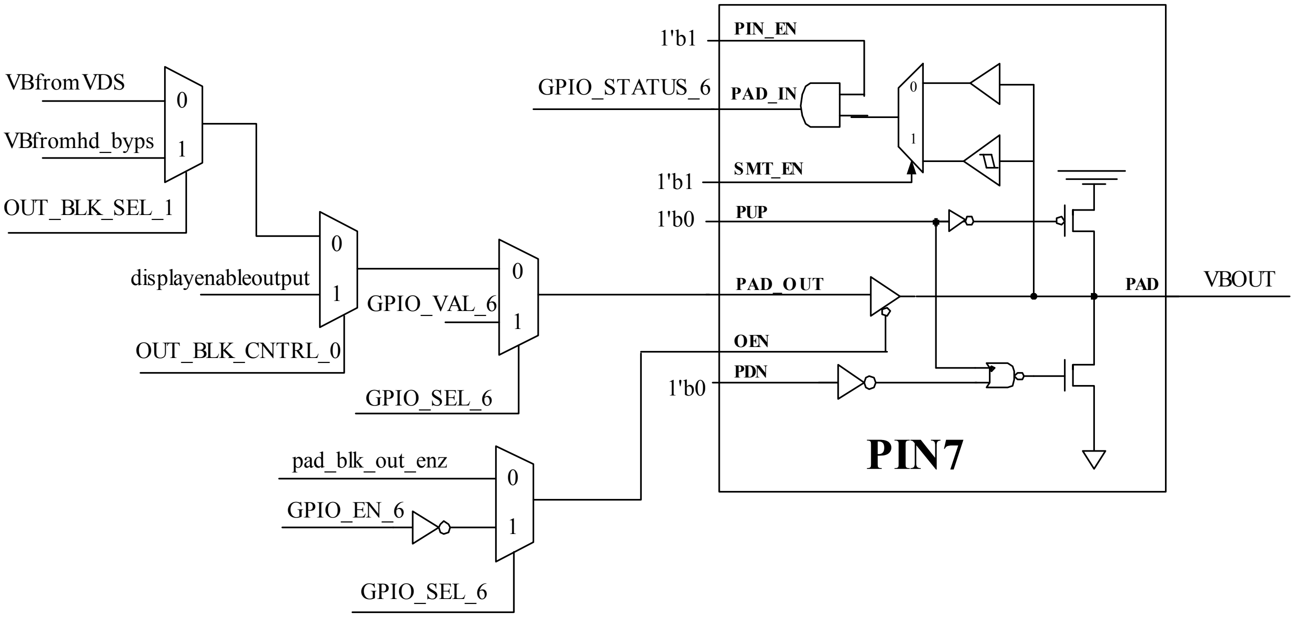
Control Name	Address	Description
ADC_POWDZ	Reg_S5_03 [0]	=0, Reset ADCTOP in power down mode =1, Enable ADCTOP in work mode
PLLAD_PDZ	Reg_S5_11 [4]	=0, Reset PLLAD in power down mode =1, Enable PLLAD in work mode
DAC_RGBS_PWDNZ	Reg_S0_44 [0]	=0, Reset RGBS in power down mode =1, Enable RGBS in work mode
SFTRST_SYNC_RSTZ	Reg_S0_47 [2]	=0, Software reset sync_proc; =0, Enable sync_proc in work mode.
SFTRST_DEC_RSTZ	Reg_S0_47 [0]	=0, Software reset decimation filter; =0, Enable decimation filter in work mode.

13.2.2.2 Digital Part Software Reset

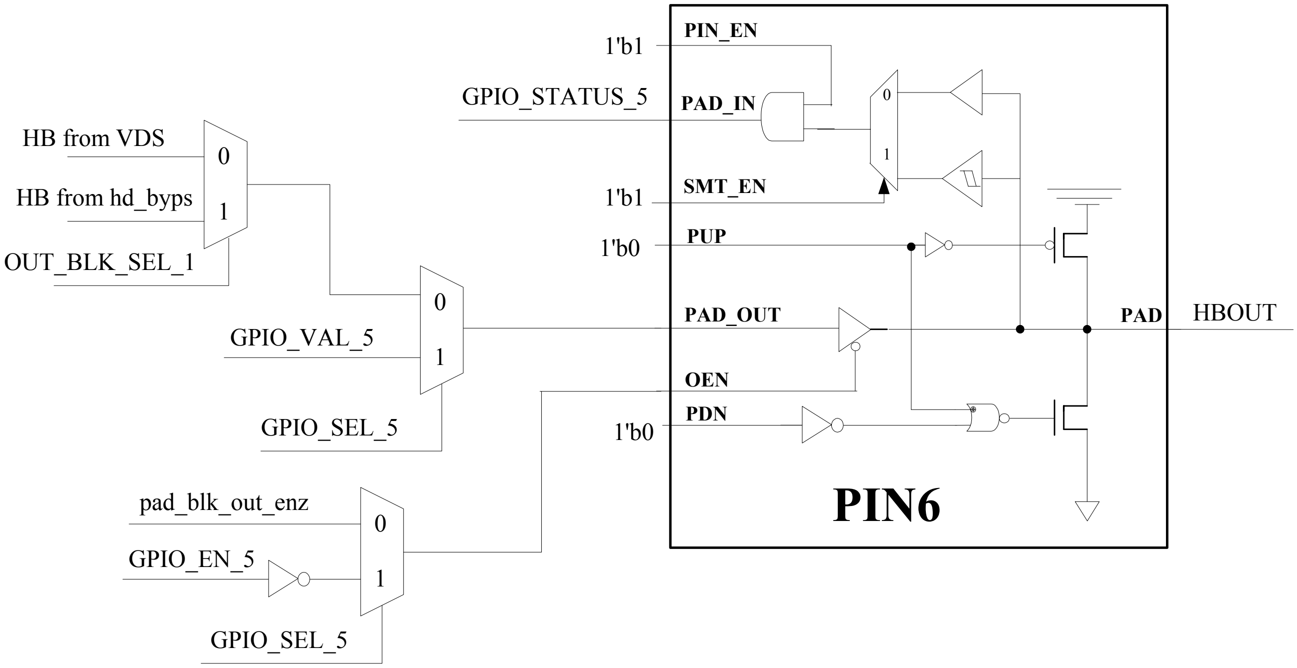
Control Name	Address	Description
SFTRST_IF_RSTZ	Reg_S0_46 [0]	=0, Software reset module input formatter =1, Enable module input formatter normal working
SFTRST_DEINT_RSTZ	Reg_S0_46 [1]	=0, Software reset module de-interlacer =1, Enable module de-interlacer normal working
SFTRST_MEM_FF_RSTZ	Reg_S0_46 [2]	=0, Software reset module mem_ff (wff/rff/cap/pb) =1, Enable module mem_ff (wff/rff/cap/pb) normal working
SFTRST_MEM_RSTZ	Reg_S0_46 [3]	=0, Software reset module memory controller =1, Enable module memory controller normal working
SFTRST_FIFO_RSTZ	Reg_S0_46 [4]	=0, Software reset all FIFO (ff64/ff512) =1, Enable all FIFO (ff64/ff512) normal working
SFTRST_OSD_RSTZ	Reg_S0_46 [5]	=0, Software reset module osd engine =1, Enable module osd engine normal working
SFTRST_VDS_FF_RSTZ	Reg_S0_46 [6]	=0, Software reset module vds_proc =1, Enable module vds_proc normal working
SFTRST_MODE_RSTZ	Reg_S0_47 [1]	=0, Software reset module mode detect =1, Enable module mode detect normal working
SFTRST_HBBYPS_RSTZ	Reg_S0_47 [3]	=0, Software reset module IF_HD-bypass =1, Enable module IF_HD_pybass normal working
SFTRST_INT_RSTZ	Reg_S0_46 [4]	=0, Software reset module interrupt generation =1, Enable module interrupt generation normal working

13.3 CLK out & SYNC/Blank out:

13.3.1 Blanking out control

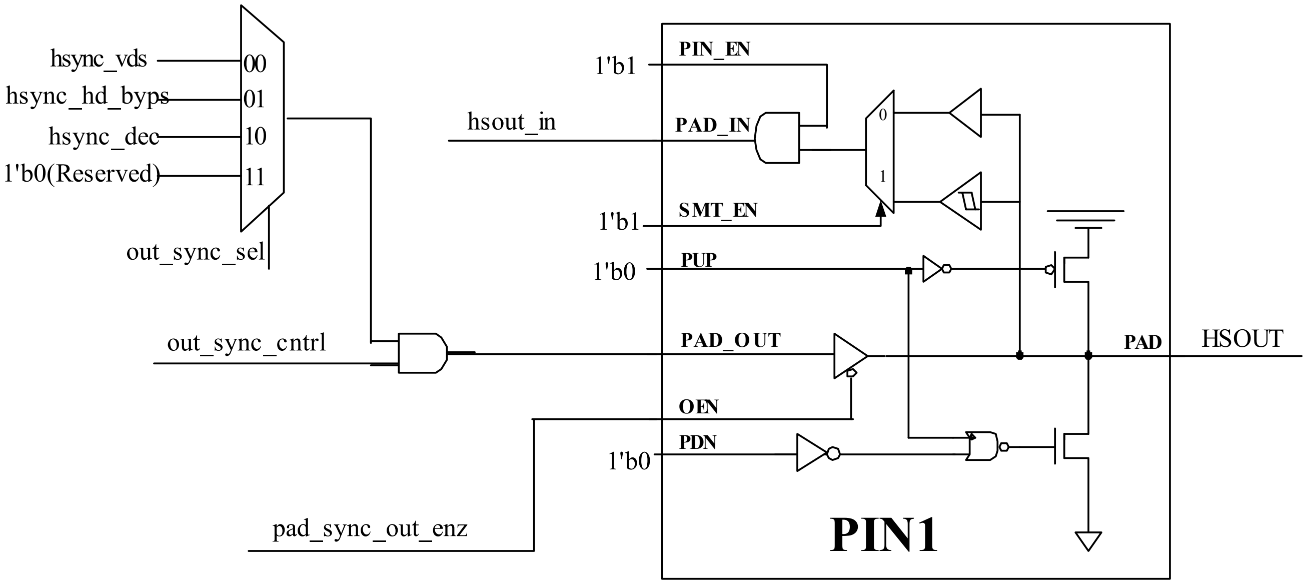


GPIO_STATUS_6	REG S0_0E bit[6]	Read only status
GPIO_SEL_6	REG S0_52 bit[6]	=1, enable pin7 as GPIO bit6
GPIO_EN_6	REG S0_53 bit[6]	=1, enable GPIO bit6 output
GPIO_VAL_6	REG S0_54 bit[6]	GPIO bit6 output value
pad_blk_out_enz	REG S0_49 bit[3]	=0 enable H/V blank output; =1 disable
OUT_BLK_SEL_0	REG S0_50 bit[0]	=0 select VB out; =1 select DE out
OUT_BLK_SEL_1	REG S0_50 bit[1]	=0 from vds_proc; =1 from HD_bypas

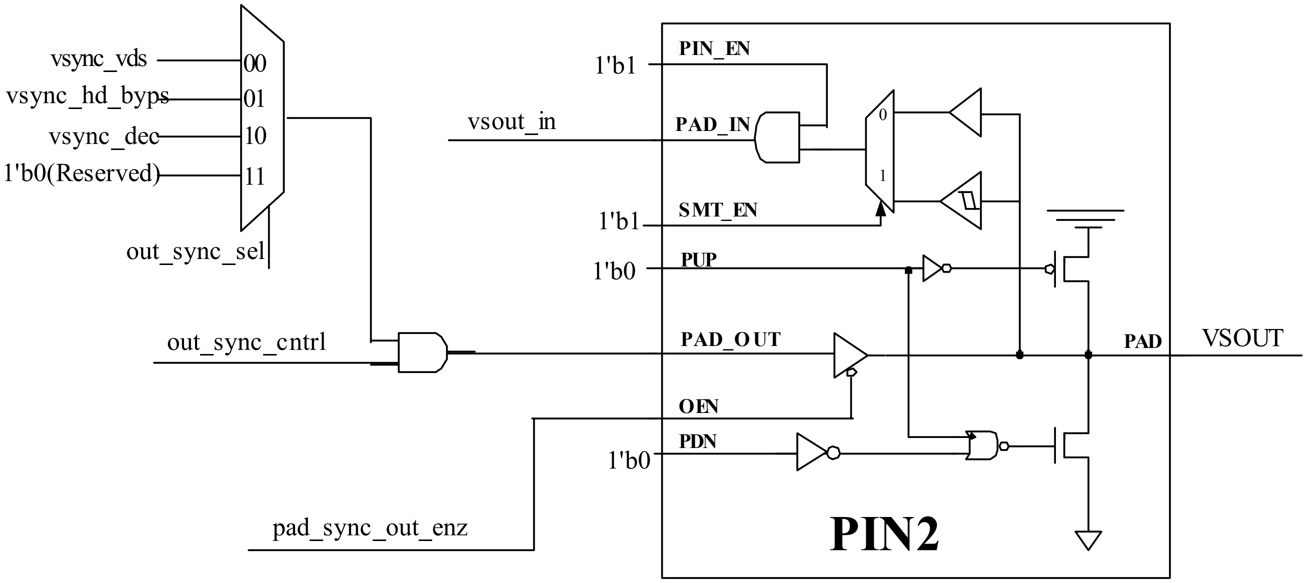


GPIO_STATUS_5	REG S0_0E bit[5]	Read only status
GPIO_SEL_5	REG S0_52 bit[5]	=1, enable pin6 as GPIO bit5
GPIO_EN_5	REG S0_53 bit[5]	=1, enable GPIO bit5 output
GPIO_VAL_5	REG S0_54 bit[5]	GPIO bit5 output value
pad_blk_out_enz	REG S0_49 bit[3]	=0 enable H/V blank output; =1 disable
OUT_BLK_SEL_1	REG S0_50 bit[1]	=0 from vds_proc; =1 from HD_bypas

13.3.2 Sync out control

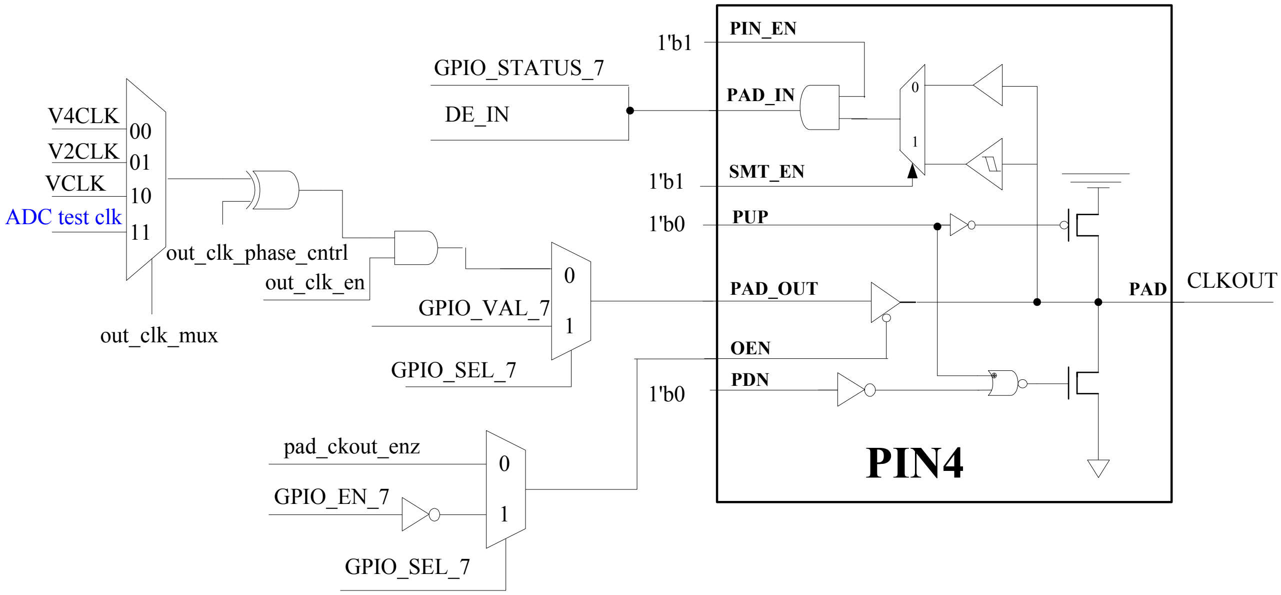


out_sync_sel	REGS0_4Fbit[7:6]	00:fromvds_proc,01:fromif_hd_byps 10:fromdecimationfilter11:Reserved
out_sync_cntrl	REGS0_4Fbit[5]	=1,enableh/vsync outputtoPAD
pad_sync_out_enz	REGS0_49bit[2]	=0,enablehsoutoutput;=1disable



out_sync_sel	REGS0_4Fbit[7:6]	00:fromvds_proc,01:fromif_hd_byps 10:fromdecimationfilter11:Reserved
out_sync_cntrl	REGS0_4Fbit[5]	=1,enableh/vsync outputtoPAD
pad_sync_out_enz	REGS0_49bit[2]	=0,enablehsoutoutput;=1disable

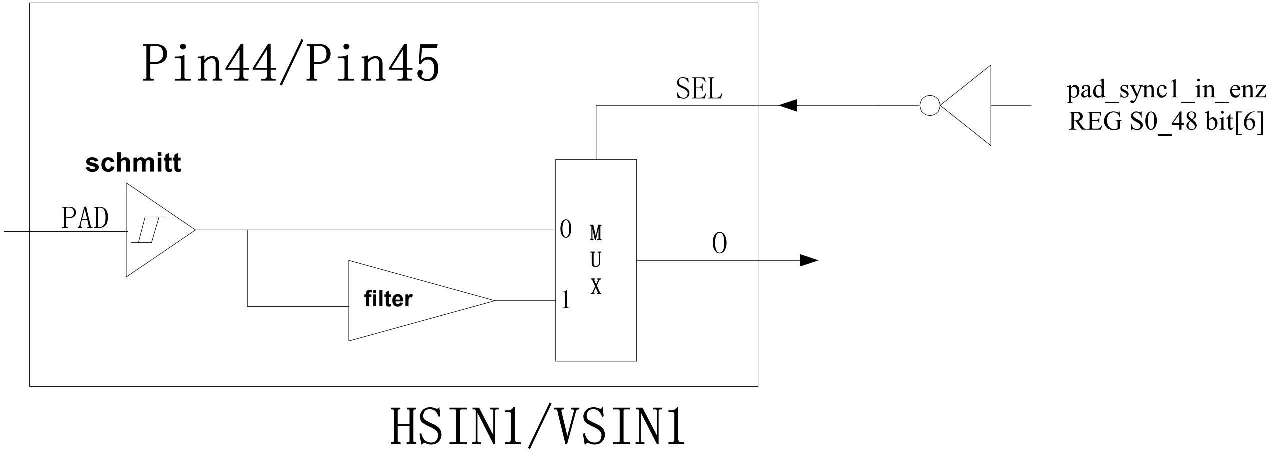
13.3.3 Clock out control

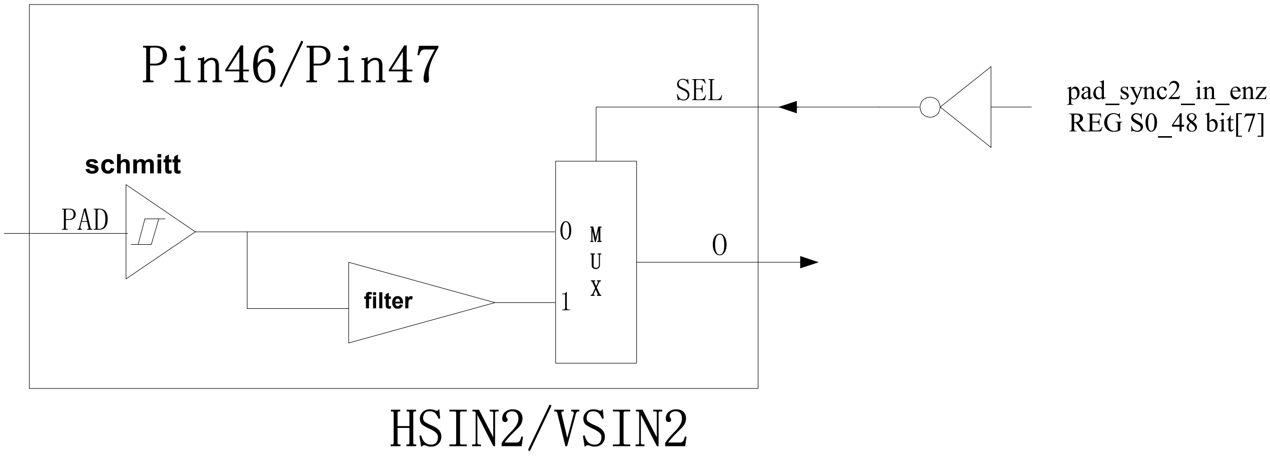


GPIO_STATUS_7	REG S0_0E bit[7]	Read only status
GPIO_SEL_7	REG S0_52 bit[7]	=1, enable pin4 as GPIO bit7
GPIO_EN_7	REG S0_53 bit[7]	=1, enable GPIO bit7 output
GPIO_VAL_7	REG S0_54 bit[7]	GPIO bit7 output value
pad_ckout_enz	REG S0_49 bit[1]	=0 enable clock output; =1 disable
out_clk_phase_cntrl	REG S0_4F bit[1]	=1, output clock inverted
out_clk_mux	REG S0_4F bit[3:2]	ouput clock select control
out_clk_en	REG S0_4F bit[4]	=1, enable clock output to PAD

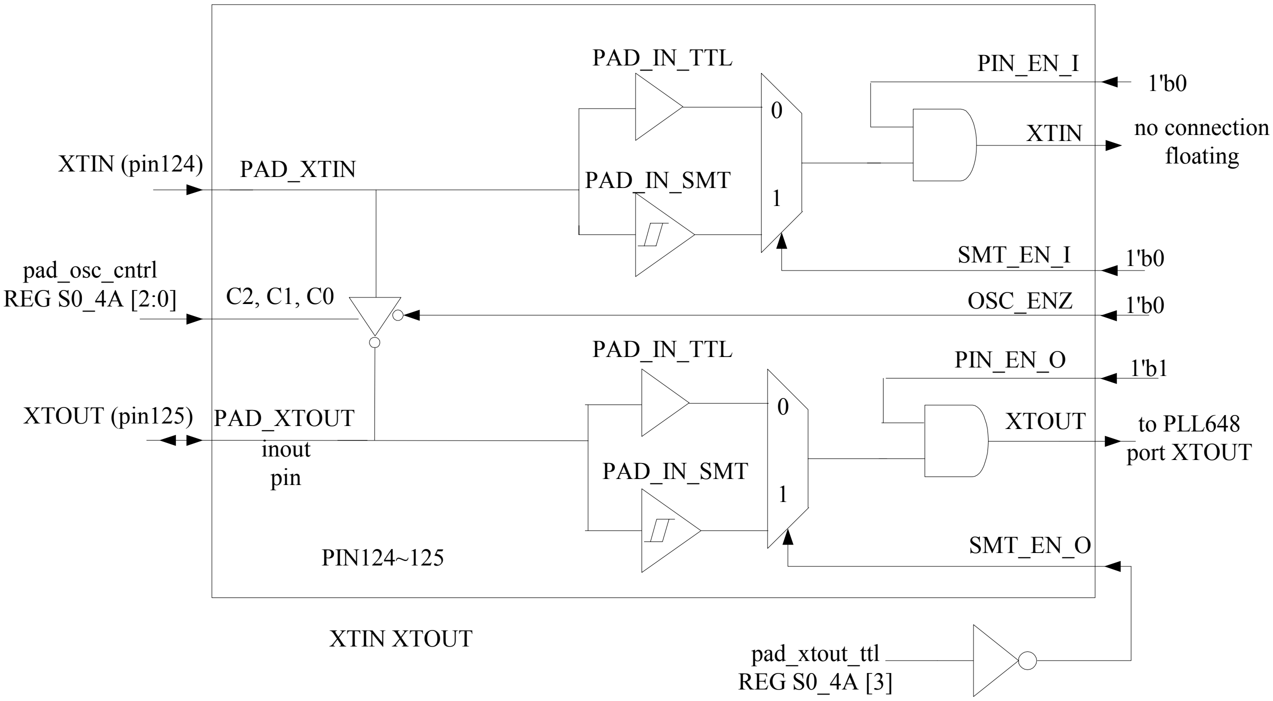
13.4 Sync Input PAD

There is filter for sync input PAD, when the input sync has bigger jitter, you can open the filter.





13.5 Crystal PAD



14 Interrupt

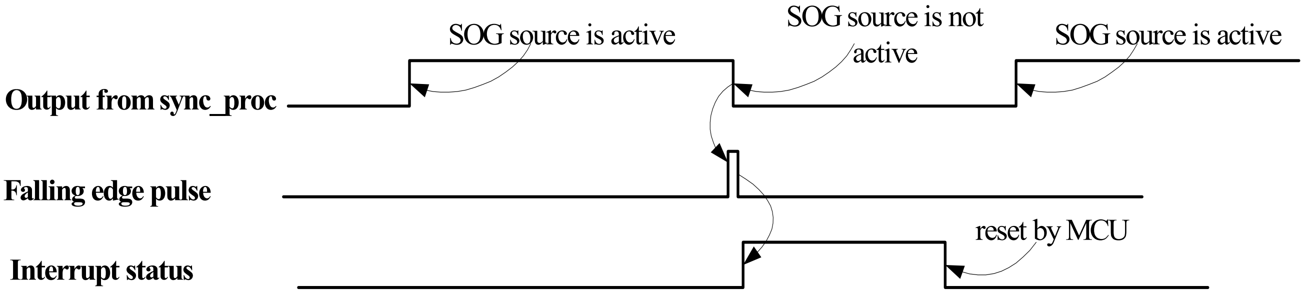
True View 5725 has one interrupt output pin (Pin76, shared with GPIO bit0).

- There are total 8 bits interrupt signals in 5725, they will be OR to output
- Each interrupt signal could be enable/disable by MCU
- MCU could read 5725's interrupt status register to get the interrupt's ID
- Interrupt signal and interrupt status register bit could only be clear by MCU
- When input source is switch between 656, SOG or external mode. MCU must program the MUX before mode detection.
    - When in SOG mode, use sync_proc's output sync
    - When in external sync mode (digital or analog input), use input sync from PAD
    - When in 656 mode, use 656 timing from 656 decoder

14.1 Interrupt 0: SOG source not stable

When SOGOUT from ADC is not stable for sync separate, this signal will go low; else it will be high. Interrupt0 will generate when the SOG source is not active, that is to say, when SOG from ADC is not stable, when we detect this interrupt we should do:

- Turn Off on-screen display
- Turn Off clamp



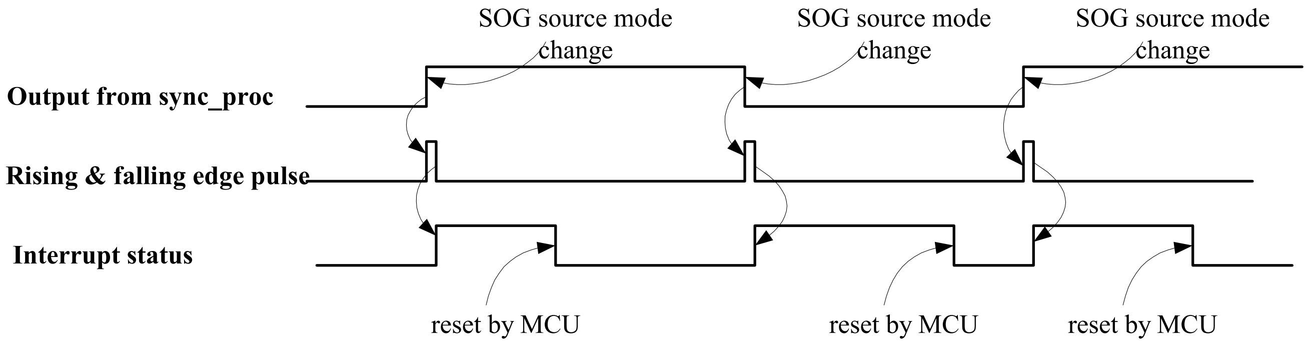
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable0	Reg_S0_59 [0]	The enable control for interrupt bit0
Int_rst0	Reg_S0_58 [0]	The reset control for interrupt bit0
Int_stautus0	Reg_S0_0f [0]	The interrupt status for interrupt bit0

14.2 Interrupt 1: Mode switch for SOG

When SOG source change mode, this signal will be inverted, interrupt1 will generated when SOG Source mode switch, when this interrupt generate we should do:

- Turn Off on-screen display
- Turn Off clamp



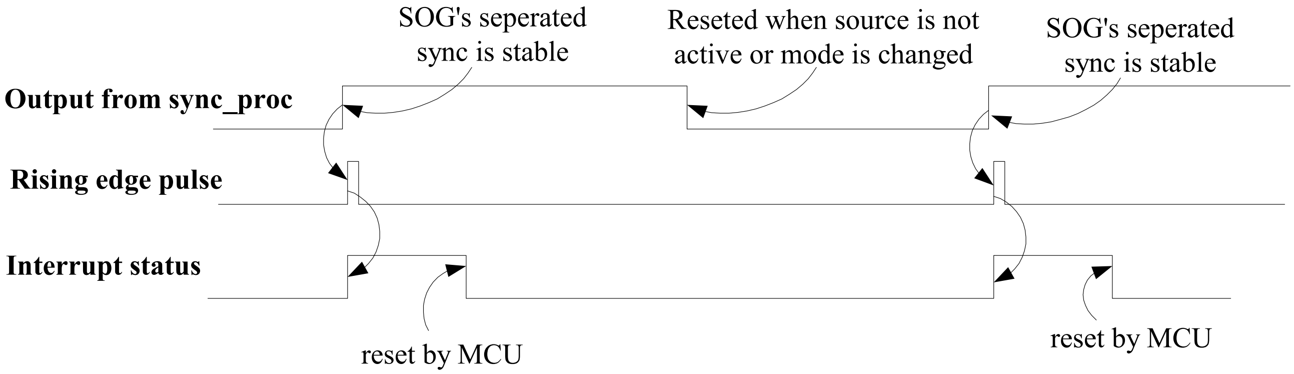
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable1	Reg_S0_59 [1]	The enable control for interrupt bit1
Int_rst1	Reg_S0_58 [1]	The reset control for interrupt bit1
Int_stautus1	Reg_S0_0f [1]	The interrupt status for interrupt bit1

14.3 Interrupt 2: SOG source stable

When the result of SOG's sync separate is stable, this signal will be high; else it will be low. And when SOG source is not active or SOG source change mode, it will be low also. Interrupt2 will generate when the SOG source is sable. When we detect this interrupt we should do:

- Turn On clamp
- Switch mode detection source to SOG
- Program register based mode detection's result
- Turn On on-screen display


SOG's seperated sync is stable
Reseted when source is not active or mode is changed
SOG's seperated sync is stable
Output from sync_proc
Rising edge pulse
Interrupt status
reset by MCU
reset by MCU

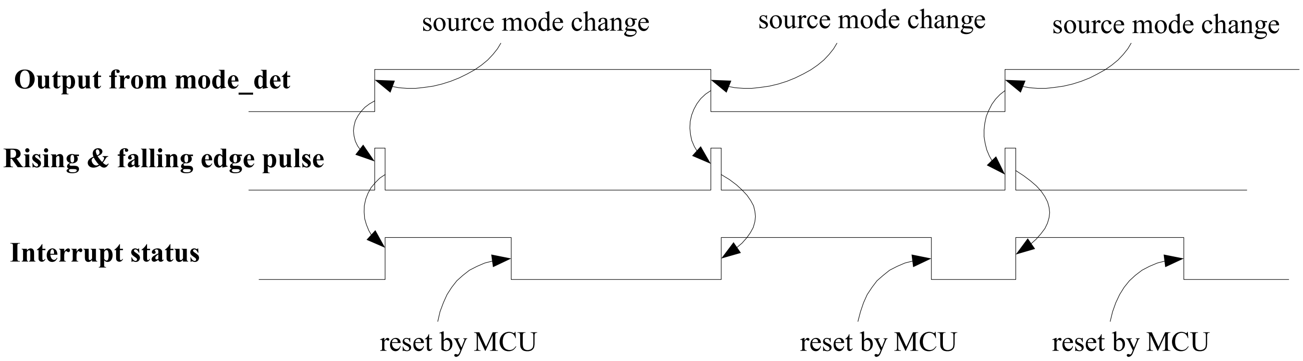
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable2	Reg_S0_59 [2]	The enable control for interrupt bit2
Int_rst2	Reg_S0_58 [2]	The reset control for interrupt bit2
Int_stautus2	Reg_S0_0f [2]	The interrupt status for interrupt bit2

14.4 Interrupt 3: Mode switch for 656/external sync

This signal is only useful when input is in 656/external sync mode. When input mode is changed, this Signal will be inverted. Interrupt3 will generated when mode switch for 656/external sync. When we detect this interrupt we should do:

- Close the display screen.
- Program register following mode detection's result
- Turn On on-screen display



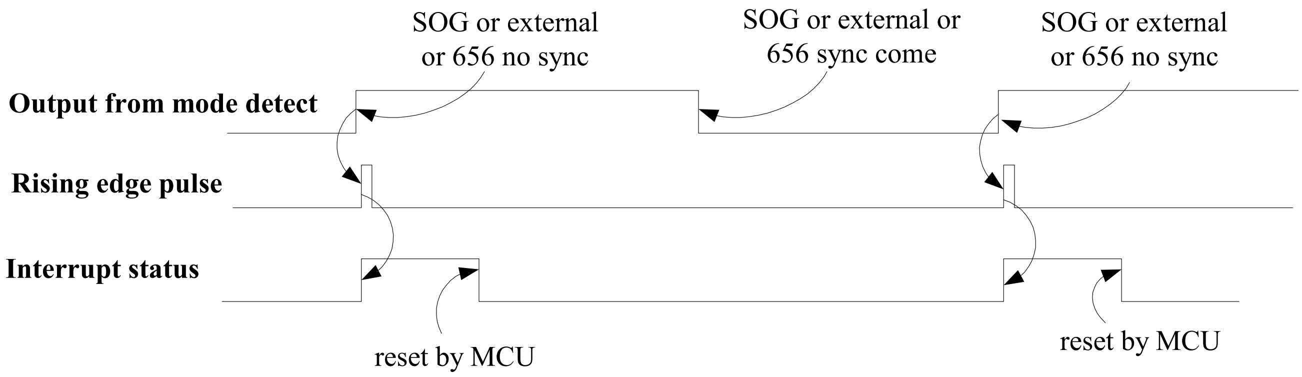
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable3	Reg_S0_59 [3]	The enable control for interrupt bit3
Int_rst3	Reg_S0_58 [3]	The reset control for interrupt bit3
Int_stautus3	Reg_S0_0f [3]	The interrupt status for interrupt bit3

14.5 Interrupt 4: No sync

This signal is used to detect the SOG/656/external sync source timing information, if there is no sync in A defined period; it will go high, else it will go low. Interrupt4 will generate when there is no sync, when we detect this interrupt we should do:

- Close the display screen.
- Switch channel or wait to sync stable
- Program register following mode detection's result



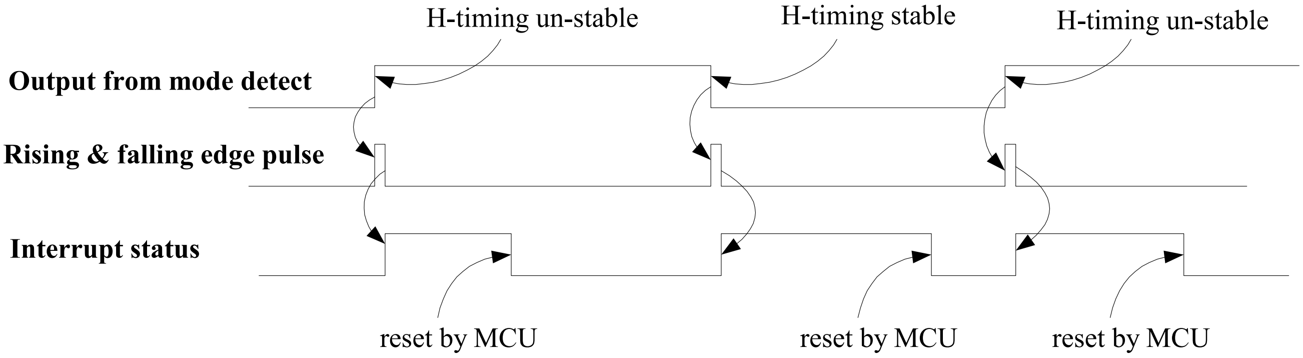
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable4	Reg_S0_59 [4]	The enable control for interrupt bit4
Int_rst4	Reg_S0_58 [4]	The reset control for interrupt bit4
Int_stautus4	Reg_S0_0f [4]	The interrupt status for interrupt bit4

14.6 Interrupt 5: H-timing switch

The signals is high when horizontal timing is detected un-stable, when horizontal timing is stable, it will Go low, interrupt 5 will generate when horizontal timing switch between stable and un-stable. When we detect this interrupt we should do:

- Program as mode detection result



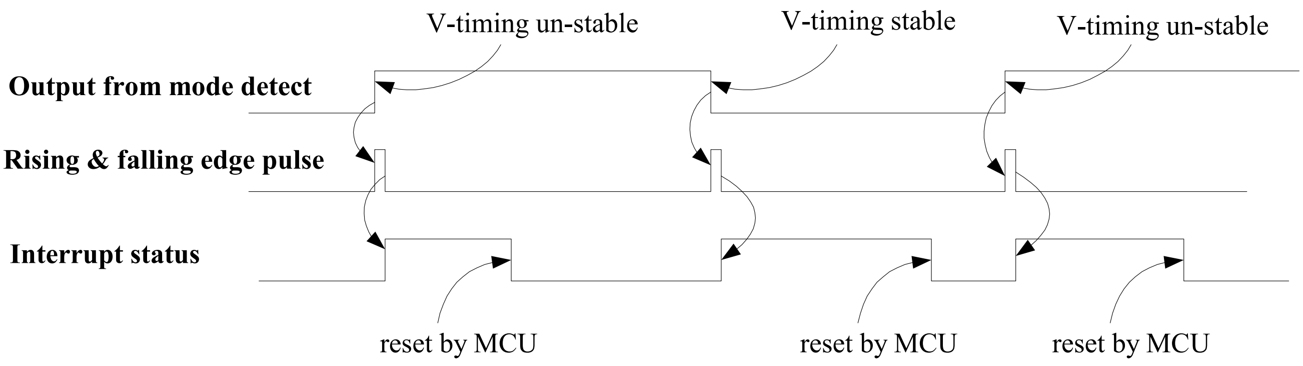
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable5	Reg_S0_59 [5]	The enable control for interrupt bit5
Int_rst5	Reg_S0_58 [5]	The reset control for interrupt bit5
Int_stautus5	Reg_S0_0f [5]	The interrupt status for interrupt bit5

14.7 Interrupt 6: V-timing switch

The signals is high when vertical timing is detected un-stable, when vertical timing is stable, it will Go low, interrupt 6 will generate when vertical timing switch between stable and un-stable. When we detect this interrupt we should do:

- Program as mode detection result



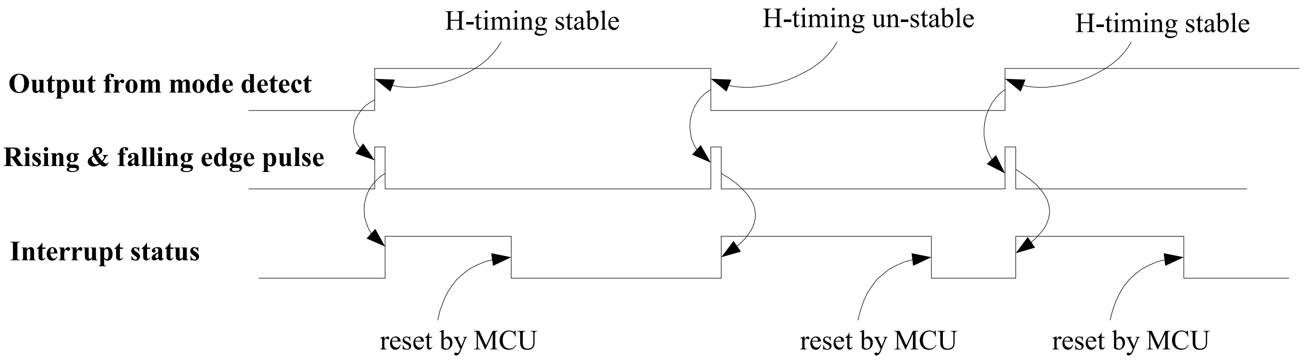
Reference Register Map:

Register Name	Register Address	Register Description
Int_enable6	Reg_S0_59 [6]	The enable control for interrupt bit6
Int_rst6	Reg_S0_58 [6]	The reset control for interrupt bit6
Int_stautus6	Reg_S0_0f [6]	The interrupt status for interrupt bit6

14.8 Interrupt 7: H-timing stable

The signals is high when horizontal timing is detected stable, when horizontal timing is un-stable, it will Go low, interrupt 7 will generate when horizontal timing switch between stable and un-stable. When we detect this interrupt we should do:

- Program as mode detection result



Reference Register Map:

Register Name	Register Address	Register Description
Int_enable7	Reg_S0_59 [7]	The enable control for interrupt bit7
Int_rst7	Reg_S0_58 [7]	The reset control for interrupt bit7
Int_stautus7	Reg_S0_0f [7]	The interrupt status for interrupt bit7

15 Video Port Usage

5725 have both analog and digital video input port.

For analog input port:

- It has three ADC input channel 0, 1, 2
- It has 2 separate sync h/v or composite sync for ADC channel 0, 1, 2
- It has 2 sog input for ADC input channel 0, 1

For analog output port:

- It has a set of RGBS, H/V-sync, H/V-blank output
- It can output sync-on-Y YPbPr or sync-on-green RGB signal

For digital input port:

- It has one digital 24-bit input port
- It can support 8-bit 601/656 YUV, 16-bit YUV, 24-bit YUV/RGB input
- It can accept de-in (display enable input) signal

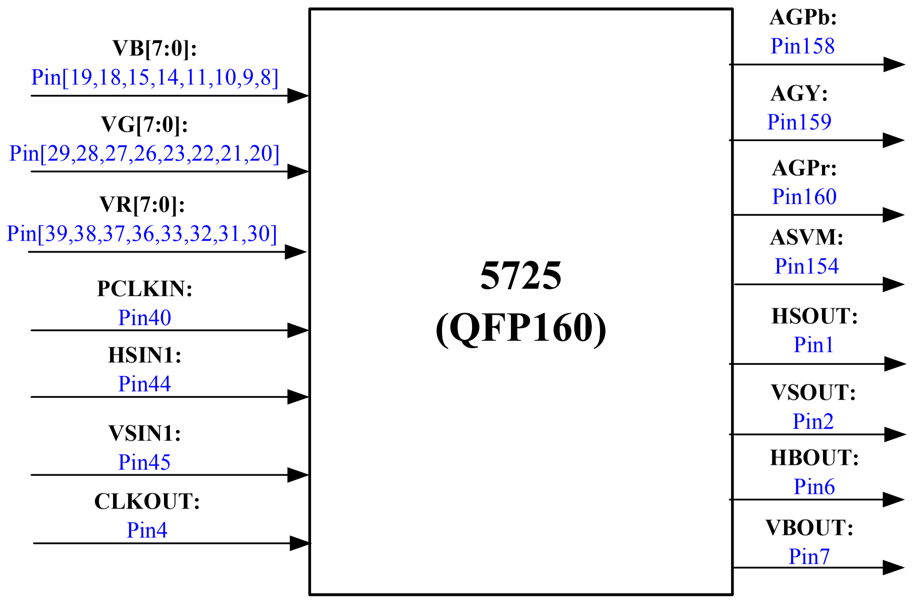
For digital output port:

- It shared with 24-bit digital input port
- It can output 16-bit YUV, 24-bit YUV/RGB.
- It can output de-out (display enable output) signal
- It can output H/V-sync and pixel clock

The following page is the port usage and corresponding registers program:

15.1 Digital 24-bit YUV/RGB Input with Analog Output Mode

System Solution 1: Digital 24-bit YUV/RGB Input with Analog Output Mode



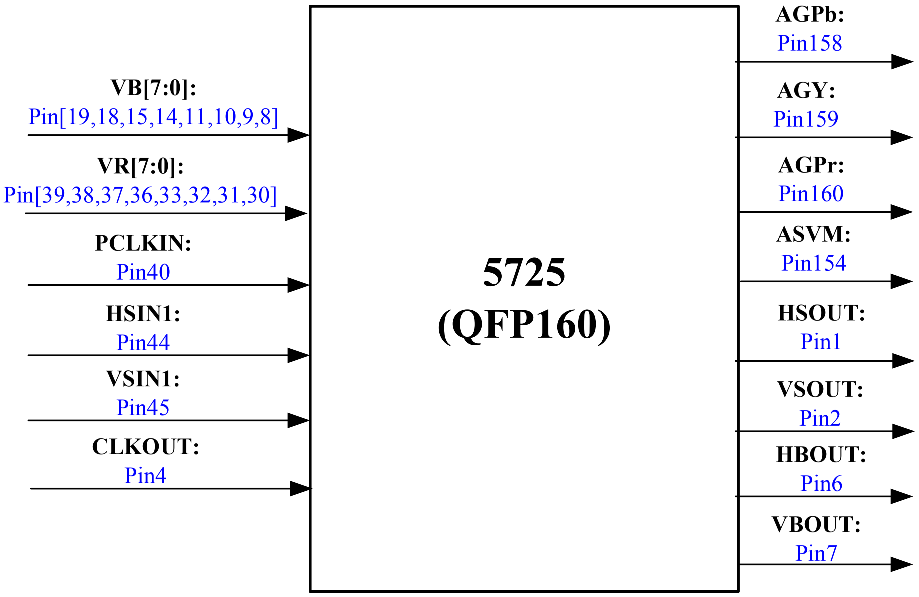
5725 Key Register Setting		
Register Name	**Address**	**Value**
pad_bout_en	Reg_S0_48[0]	1'b0
pad_bin_enz	Reg_S0_48[1]	1'b0
pad_rout_en	Reg_S0_48[2]	1'b0
pad_rin_enz	Reg_S0_48[3]	1'b0
pad_gout_en	Reg_S0_48[4]	1'b0
pad_gin_enz	Reg_S0_48[5]	1'b0
pad_ckout_enz	Reg_S0_49[1]	1'b1
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b0
out_blank_sel_0	Reg_S0_50[0]	1'b0
if_sel_adc_sync	Reg_S1_28[2]	1'b0
if_sel_656	Reg_S1_00[3]	1'b0
if_sel16bit	Reg_S1_00[4]	X
if_sel24bit	Reg_S1_01[7]	1'b1

Note: "X" means either "0" or "1" is OK.

Input Pin	Description	Output Pin	Description
VB[7:0]	Digital Blue/U data input	AGPb	Analog Blue/Pb output
VG[7:0]	Digital Green/Y data input	AGY	Analog Green/Y output
VR[7:0]	Digital Red/V data input	AGPr	Analog Red/Pr output
PCLKIN	Pixel clock input	ASVM	Analog SVM output
HSIN1	Video H-sync input	HSOUT	Video H-sync output
VSIN1	Video V-sync input	VSOUT	Video V-sync output
CLKOUT	DE input from DVI/HDMI	HBOUT	Video H-blank output
		VBOUT	Video V-blank output

15.2 Digital 16-bit YUV 4:2:2 Input with Analog Output Mode

System Solution 2: Digital 16-bit YUV 4:2:2 Input with Analog Output Mode



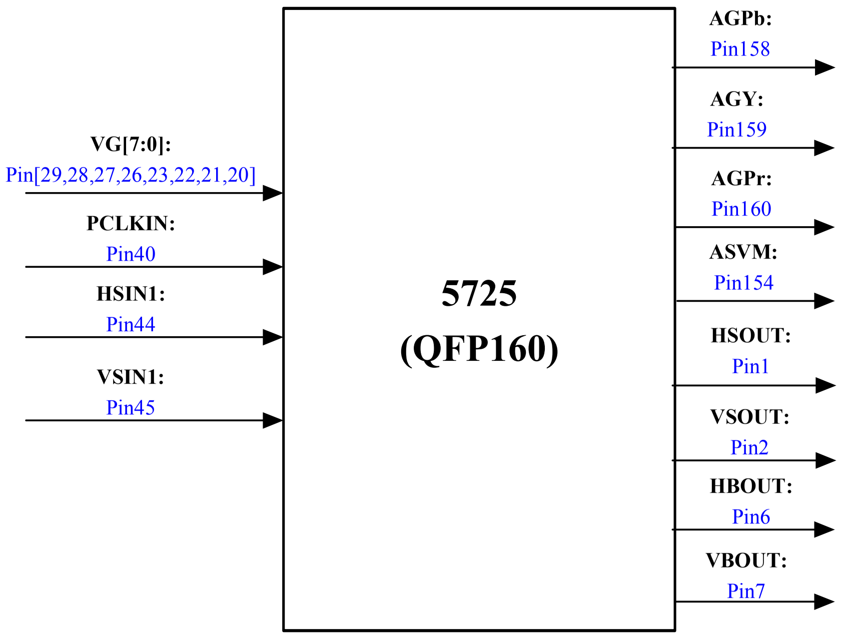
Input Pin	Description	Output Pin	Description
VB[7:0]	Digital UV data input	AGPb	Analog Blue/Pb output
VR[7:0]	Digital Y data input	AGY	Analog Green/Y output
PCLKIN	Pixel clock input	AGPr	Analog Red/Pr output
HSIN1	Video H-sync input	ASVM	Analog SVM output
VSIN1	Video V-sync input	HSOUT	Video H-sync output
CLKOUT	DE input from DVI/HDMI	VSOUT	Video V-sync output
		HBOUT	Video H-blank output
		VBOUT	Video V-blank output

5725 Key Register Setting		
Register Name	Address	Value
pad_bout_en	Reg_S0_48[0]	1'b0
pad_bin_enz	Reg_S0_48[1]	1'b0
pad_rout_en	Reg_S0_48[2]	1'b0
pad_rin_enz	Reg_S0_48[3]	1'b0
pad_gout_en	Reg_S0_48[4]	X
pad_gin_enz	Reg_S0_48[5]	X
pad_ckout_enz	Reg_S0_49[1]	1'b1
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b0
out_blank_sel_0	Reg_S0_50[0]	1'b0
if_sel_adc_sync	Reg_S1_28[2]	1'b0
if_sel_656	Reg_S1_00[3]	1'b0
if_sel16bit	Reg_S1_00[4]	1'b1
if_sel24bit	Reg_S1_01[7]	1'b0

Note: "X" means either "0" or "1" is OK.

15.3 Digital 8-bit 601/656 4:2:2 YUV Input with Analog Output Mode

System Solution 3: Digital 8-bit 601/656 4:2:2 YUV Input with Analog Output Mode



Input Pin	Description	Output Pin	Description
VG[7:0]	Digital YUV data input	AGPb	Analog Blue/Pb output
PCLKIN	Pixel clock input	AGY	Analog Green/Y output
HSIN1	Video H-sync input	AGPr	Analog Red/Pr output
VSIN1	Video V-sync input	ASVM	Analog SVM output
		HSOUT	Video H-sync output
		VSOUT	Video V-sync output
		HBOUT	Video H-blank output
		VBOUT	Video V-blank output

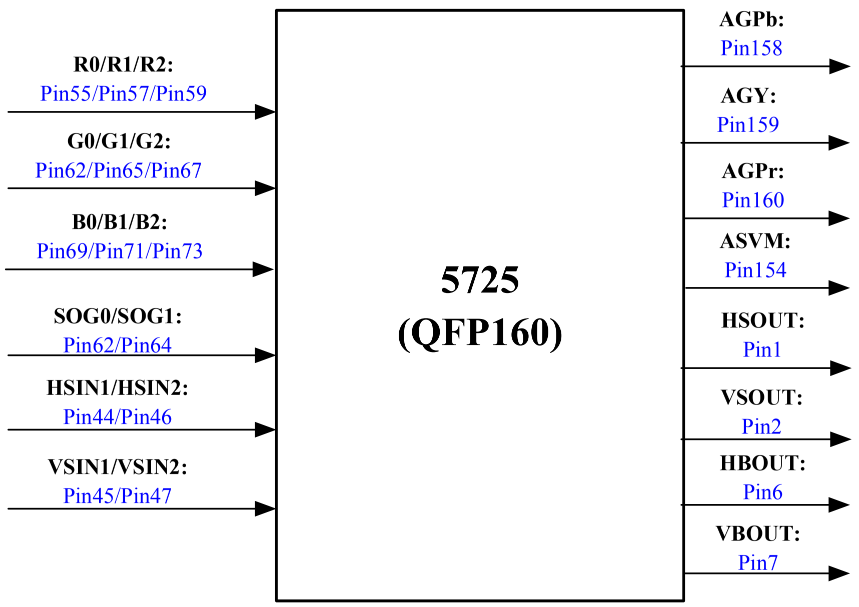
5725 Key Register Setting		
Register Name	Address	Value
pad_bout_en	Reg_S0_48[0]	X
pad_bin_enz	Reg_S0_48[1]	X
pad_rout_en	Reg_S0_48[2]	X
pad_rin_enz	Reg_S0_48[3]	X
pad_gout_en	Reg_S0_48[4]	1'b0
pad_gin_enz	Reg_S0_48[5]	1'b0
pad_ckout_enz	Reg_S0_49[1]	X
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b0
out_blank_sel_0	Reg_S0_50[0]	1'b0
if_sel_adc_sync	Reg_S1_28[2]	1'b0
if_sel_656	Reg_S1_00[3]	Note*
if_sel16bit	Reg_S1_00[4]	1'b0
if_sel24bit	Reg_S1_01[7]	1'b0

Note: "X" means either "0" or "1" is OK.

Note* : if 656 input , set it to 1;
if 601 input , set it to 0.

15.4 Analog YUV/RGB Input with Analog Output Mode

System Solution 4: Analog YUV/RGB Input with Analog Output Mode



5725 Key Register Setting		
Register Name	**Address**	**Value**
pad_bout_en	Reg_S0_48[0]	X
pad_bin_enz	Reg_S0_48[1]	X
pad_rout_en	Reg_S0_48[2]	X
pad_rin_enz	Reg_S0_48[3]	X
pad_gout_en	Reg_S0_48[4]	X
pad_gin_enz	Reg_S0_48[5]	X
pad_ckout_enz	Reg_S0_49[1]	X
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b0
out_blank_sel_0	Reg_S0_50[0]	1'b0
if_sel_adc_sync	Reg_S1_28[2]	1'b1
if_sel_656	Reg_S1_00[3]	1'b0
if_sel16bit	Reg_S1_00[4]	X
if_sel24bit	Reg_S1_01[7]	1'b1

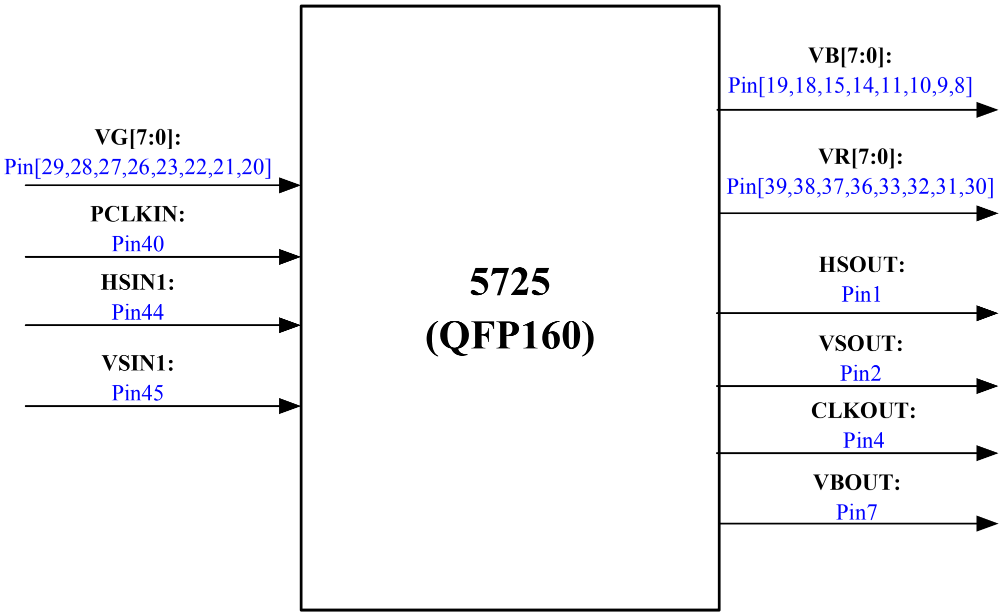
Note: "X" means either "0" or "1" is OK.

Input Pin	Description	Output Pin	Description
R0/R1/R2	Analog R/V input of CHN0/1/2	AGPb	Analog Blue/Pb output
G0/G1/G2	Analog G/Y input of CHN0/1/2	AGY	Analog Green/Y output
B0/B1/B2	Analog B/U input of CHN0/1/2	AGPr	Analog Red/Pr output
SOG0/SOG1	Analog SOG/Y input of CHN0/1	ASVM	Analog SVM output
HSIN1/HSIN2	Video H-sync input 1/2*	HSOUT	Video H-sync output
VSIN1/VSIN2	Video V-sync input 1/2*	VSOUT	Video V-sync output
		HBOUT	Video H-blank output
		VBOUT	Video V-blank output

Note*: HSIN1/VSIN1 and HSIN2/VSIN2 could be programmed for analog channel 0/1/2

15.5 Digital 8-bit 601/656 4:2:2 YUV Input with 16bit Digital Output Mode

System Solution 5: Digital 8-bit 601/656 4:2:2 YUV Input with 16bit Digital Output Mode



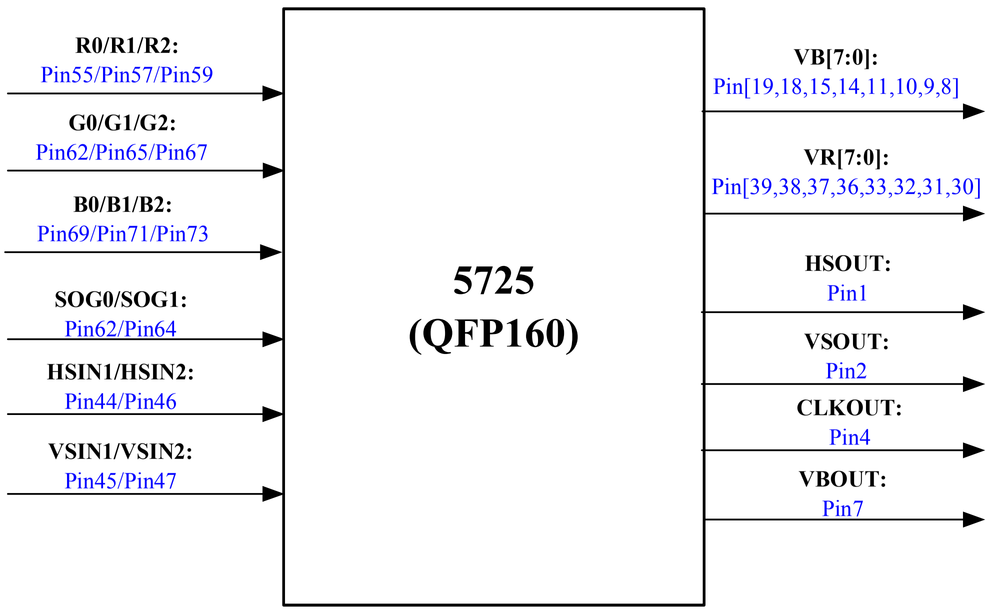
5725 Key Register Setting		
Register Name	**Address**	**Value**
pad_bout_en	Reg_S0_48[0]	1'b1
pad_bin_enz	Reg_S0_48[1]	1'b1
pad_rout_en	Reg_S0_48[2]	1'b1
pad_rin_enz	Reg_S0_48[3]	1'b1
pad_gout_en	Reg_S0_48[4]	1'b0
pad_gin_enz	Reg_S0_48[5]	1'b0
pad_ckout_enz	Reg_S0_49[1]	1'b0
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b1
out_blank_sel_0	Reg_S0_50[0]	1'b1
if_sel_adc_sync	Reg_S1_28[2]	1'b0
if_sel_656	Reg_S1_00[3]	1'b1
if_sel16bit	Reg_S1_00[4]	1'b0
if_sel24bit	Reg_S1_01[7]	1'b0

Note: "X" means either "0" or "1" is OK.

Input Pin	Description	Output Pin	Description
VG[7:0]	Digital YUV data input	VB[7:0]	Digtial UV data output
PCLKIN	Pixel clock input	VR[7:0]	Digital Y data output
HSIN1	Video H-sync input	HSOUT	Video H-sync output
VSIN1	Video V-sync input	VSOUT	Video V-sync output
		CLKOUT	Video display clock output
		VBOUT	Display enable output for LCD

15.6 Analog RGB/YUV Input with 16bit Digital Output Mode

System Solution 6: Analog RGB/YUV Input with 16bit Digital Output Mode



5725 Key Register Setting		
Register Name	Address	Value
pad_bout_en	Reg_S0_48[0]	1'b1
pad_bin_enz	Reg_S0_48[1]	1'b1
pad_rout_en	Reg_S0_48[2]	1'b1
pad_rin_enz	Reg_S0_48[3]	1'b1
pad_gout_en	Reg_S0_48[4]	X
pad_gin_enz	Reg_S0_48[5]	X
pad_ckout_enz	Reg_S0_49[1]	1'b0
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b1
out_blank_sel_0	Reg_S0_50[0]	1'b1
if_sel_adc_sync	Reg_S1_28[2]	1'b1
if_sel_656	Reg_S1_00[3]	X
if_sel16bit	Reg_S1_00[4]	X
if_sel24bit	Reg_S1_01[7]	1'b1

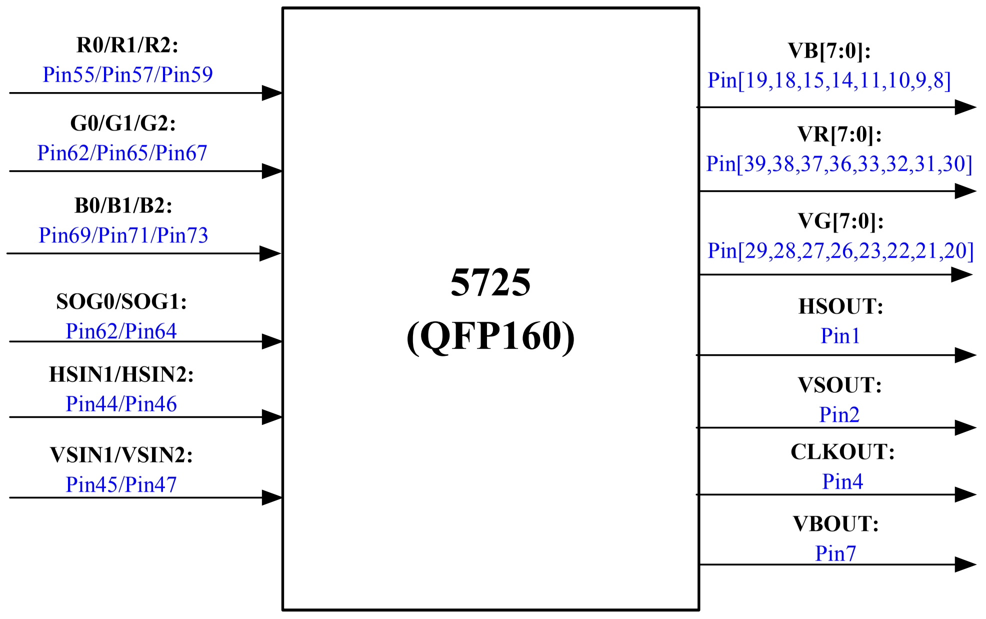
Note: "X" means either "0" or "1" is OK.

Input Pin	Description	Output Pin	Description
R0/R1/R2	Analog R/V input of CHN0/1/2	VB[7:0]	Digtial UV data output
G0/G1/G2	Analog G/Y input of CHN0/1/2	VR[7:0]	Digital Y data output
B0/B1/B2	Analog B/U input of CHN0/1/2	HSOUT	Video H-sync output
SOG0/SOG1	Analog SOG/Y input of CHN1/2	VSOUT	Video V-sync output
HSIN1/HSIN2	Video H-sync input 1/2*	CLKOUT	Video display clock output
VSIN1/VSIN2	Video V-sync input 1/2*	VBOUT	Display enable output for LCD

Note*: HSIN1/VSIN1 and HSIN2/VSIN2 could be programmed for analog channel 0/1/2

15.7 Analog RGB/YUV Input with 24bit Digital Output Mode

System Solution 7: Analog RGB/YUV Input with 24bit Digital Output Mode



5725 Key Register Setting		
Register Name	**Address**	**Value**
pad_bout_en	Reg_S0_48[0]	1'b1
pad_bin_enz	Reg_S0_48[1]	1'b1
pad_rout_en	Reg_S0_48[2]	1'b1
pad_rin_enz	Reg_S0_48[3]	1'b1
pad_gout_en	Reg_S0_48[4]	1'b1
pad_gin_enz	Reg_S0_48[5]	1'b1
pad_ckout_enz	Reg_S0_49[1]	1'b0
pad_sync_out_enz	Reg_S0_49[2]	1'b0
pad_blk_out_enz	Reg_S0_49[3]	1'b0
vds_do_16b_en	Reg_S3_50[7]	1'b0
out_blank_sel_0	Reg_S0_50[0]	1'b1
if_sel_adc_sync	Reg_S1_28[2]	1'b1
if_sel_656	Reg_S1_00[3]	1'b0
if_sel16bit	Reg_S1_00[4]	X
if_sel24bit	Reg_S1_01[7]	1'b1

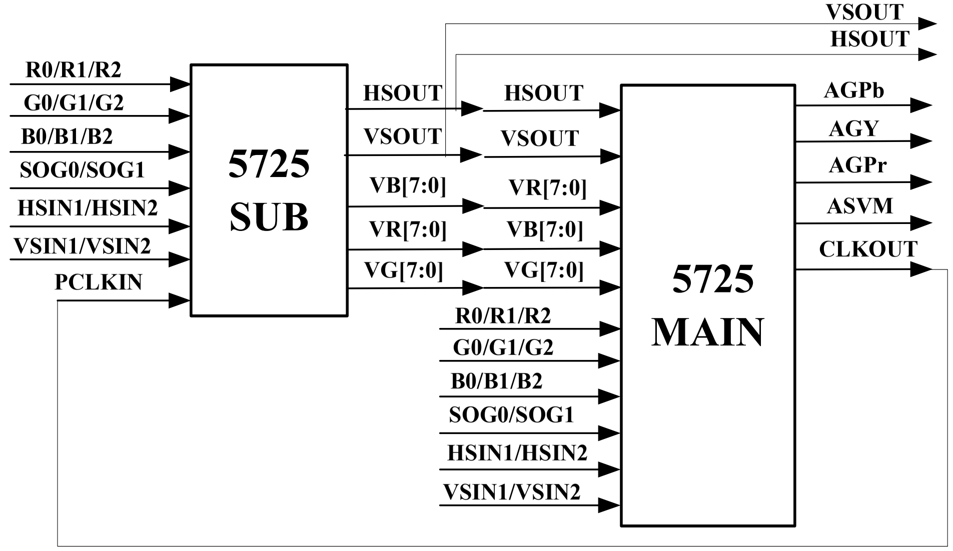
Note: "X" means either "0" or "1" is OK.

Input Pin	Description	Output Pin	Description
R0/R1/R2	Analog R/V input of CHN0/1/2	VB[7:0]	Digtial R/V data output
G0/G1/G2	Analog G/Y input of CHN0/1/2	VG[7:0]	Digtial G/Y data output
B0/B1/B2	Analog B/U input of CHN0/1/2	VR[7:0]	Digital B/U data output
SOG0/SOG1	Analog SOG/Y input of CHN1/2	HSOUT	Video H-sync output
HSIN1/HSIN2	Video H-sync input 1/2*	VSOUT	Video V-sync output
VSIN1/VSIN2	Video V-sync input 1/2*	CLKOUT	Video display clock output
		VBOUT	Display enable output for LCD

Note*: HSIN1/VSIN1 and HSIN2/VSIN2 could be programmed for analog channel 0/1/2

15.8 5725+5725 PIP 24-bit Mode

System Solution 8: 5725+5725 PIP 24-bit Mode



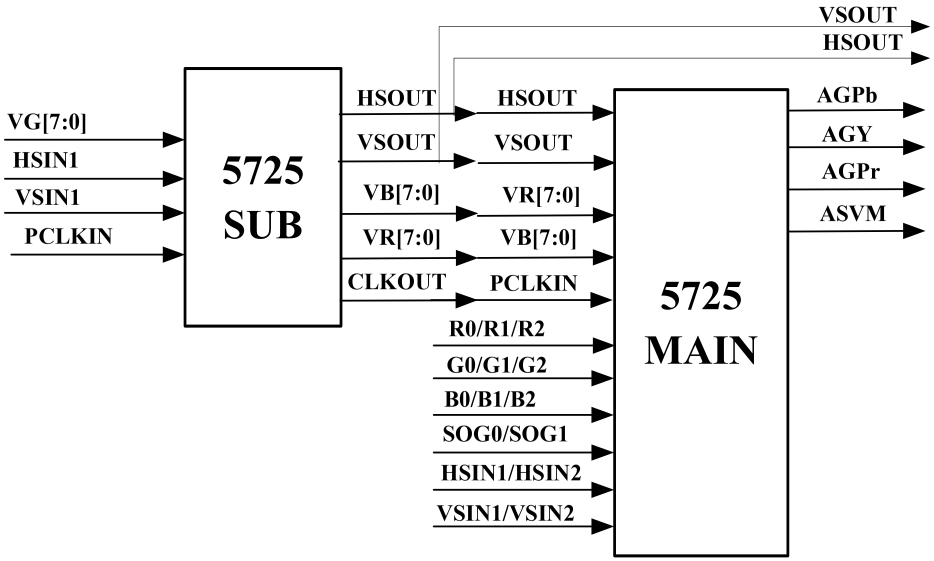
Chip	Input Pin	Description	Output Pin	Description
Sub	R0/R1/R2	Analog R/V input	VB[7:0]	Digtial R/V data output
	G0/G1/G2	Analog G/Y input	VG[7:0]	Digtial G/Y data output
	B0/B1/B2	Analog B/U input	VR[7:0]	Digital B/U data output
	SOG0/SOG1	Analog SOG/Y input	HSOUT	Video H-sync output
	HSIN1/HSIN2	Video H-sync input	VSOUT	Video V-sync output
	VSIN1/VSIN2	Video V-sync input		
	PCLKIN	Video pixel clock input		
Main	HSOUT	H-sync input for PIP	AGPb	Analog Blue/Pb output
	VSOUT	V-sync input for PIP	AGY	Analog Green/Y output
	VB[7:0]	Input B/U data for PIP	AGPr	Analog Red/Pr output
	VG[7:0]	Input G/Y data for PIP	ASVM	Analog SVM output
	VR[7:0]	Input R/V data for PIP	CLKOUT	Vidoe clock output
	R0/R1/R2	Analog R/V input		
	G0/G1/G2	Analog G/Y input		
	B0/B1/B2	Analog B/U input		
	SOG0/SOG1	Analog SOG/Y input		
	HSIN1/HSIN2	Video H-sync input		
	VSIN1/VSIN2	Video V-sync input		

5725 Key Register Setting			
Register Name	**Address**	**Sub**	**Main**
pad_bout_en	Reg_S0_48[0]	1'b1	1'b0
pad_bin_enz	Reg_S0_48[1]	1'b1	1'b0
pad_rout_en	Reg_S0_48[2]	1'b1	1'b0
pad_rin_enz	Reg_S0_48[3]	1'b1	1'b0
pad_gout_en	Reg_S0_48[4]	1'b1	1'b0
pad_gin_enz	Reg_S0_48[5]	1'b1	1'b0
pad_ckout_enz	Reg_S0_49[1]	X	1'b0
pad_sync_out_enz	Reg_S0_49[2]	1'b0	1'b1
pad_blk_out_enz	Reg_S0_49[3]	X	X
vds_do_16b_en	Reg_S3_50[7]	1'b0	1'b0
out_blank_sel_0	Reg_S0_50[0]	X	X
if_sel_adc_sync	Reg_S1_28[2]	1'b1	1'b1
if_sel_656	Reg_S1_00[3]	1'b0	1'b0
if_sel16bit	Reg_S1_00[4]	X	X
if_sel24bit	Reg_S1_01[7]	1'b1	1'b1

Note: "X" means either "0" or "1" is OK.

15.9 5725+5725 PIP 16-bit Mode

System Solution 9: 5725+5725 PIP 16-bit Mode



Chip	Input Pin	Description	Output Pin	Description
Sub	VG[7:0]	Digital YUV data input	VB[7:0]	Digtial UV data output
	PCLKIN	Pixel clock input	VR[7:0]	Digital Y data output
	HSIN1	Video H-sync input	HSOUT	Video H-sync output
	VSIN1	Video V-sync input	VSOUT	Video V-sync output
			CLKOUT	Vidoe clock output
Main	HSOUT	H-sync input for PIP	AGPb	Analog Blue/Pb output
	VSOUT	V-sync input for PIP	AGY	Analog Green/Y output
	VB[7:0]	Input Y data for PIP	AGPr	Analog Red/Pr output
	VR[7:0]	Input U/V data for PIP	ASVM	Analog SVM output
	PCLKIN	Pixel clock input		
	R0/R1/R2	Analog R/V input		
	G0/G1/G2	Analog G/Y input		
	B0/B1/B2	Analog B/U input		
	SOG0/SOG1	Analog SOG/Y input		
	HSIN1/HSIN2	Video H-sync input		
	VSIN1/VSIN2	Video V-sync input		

5725 Key Register Setting			
Register Name	**Address**	**Sub**	**Main**
pad_bout_en	Reg_S0_48[0]	1'b1	1'b0
pad_bin_enz	Reg_S0_48[1]	1'b1	1'b0
pad_rout_en	Reg_S0_48[2]	1'b1	1'b0
pad_rin_enz	Reg_S0_48[3]	1'b1	1'b0
pad_gout_en	Reg_S0_48[4]	X	X
pad_gin_enz	Reg_S0_48[5]	X	X
pad_ckout_enz	Reg_S0_49[1]	1'b0	X
pad_sync_out_enz	Reg_S0_49[2]	1'b0	1'b1
pad_blk_out_enz	Reg_S0_49[3]	X	X
vds_do_16b_en	Reg_S3_50[7]	1'b1	X
out_blank_sel_0	Reg_S0_50[0]	X	X
if_sel_adc_sync	Reg_S1_28[2]	1'b0	1'b1
if_sel_656	Reg_S1_00[3]	1'b1	1'b0
if_sel16bit	Reg_S1_00[4]	1'b0	X
if_sel24bit	Reg_S1_01[7]	1'b0	1'b1

Note: "X" means either "0" or "1" is OK.

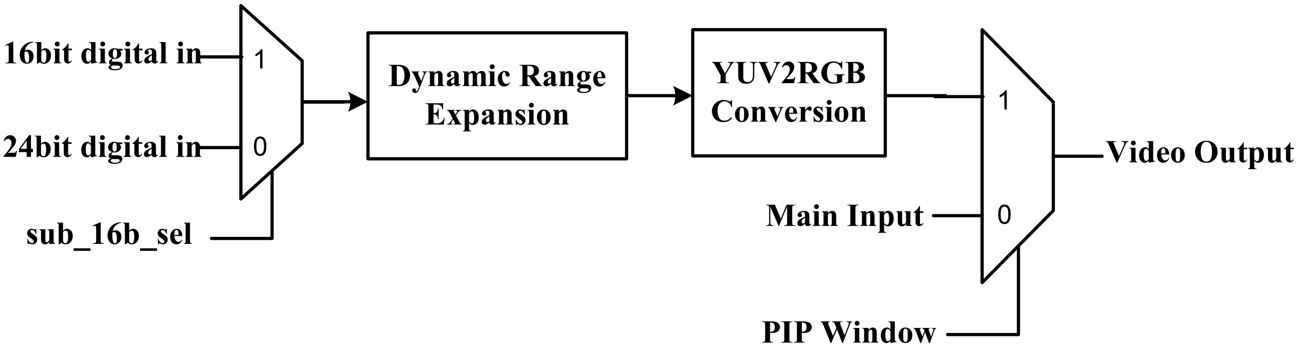
16 PIP

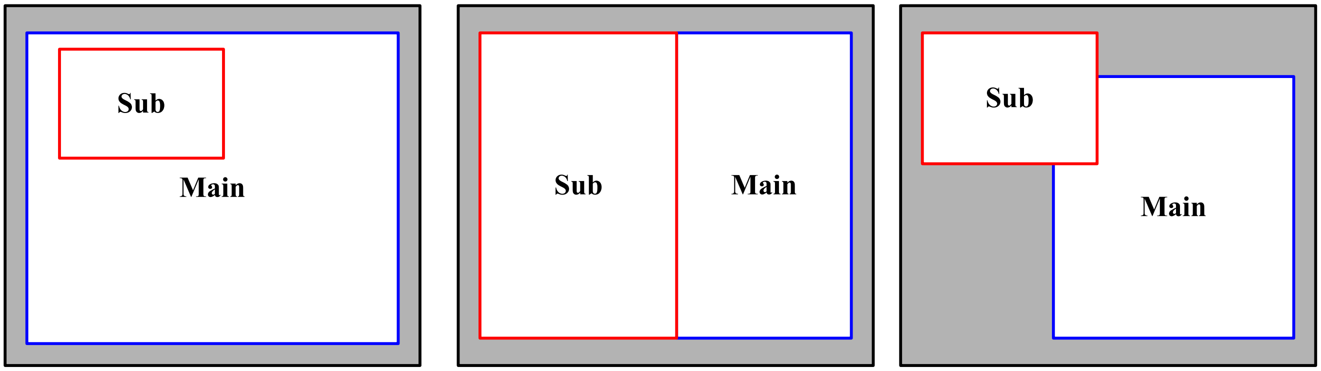
5725 PIP function need implemented with two 5725 chips, one is for main channel, and the other is for sub Channel, the sub channel is 16/24-bit digital input:

- When sub is 16-bit digital input, main can be 8-bit digital input or analog input;
- When sub is 24-bit digital input, main can be analog input only;

Because sub channel used digital video port, so PIP function only work for analog video output.

The following is the block diagram of PIP function:





16.1 Dynamic range expansion

For PIP sub channel input data, we can adjust the gain and offset as following formula:

Yout = Yin * Y_gain + Y_offset;
Uout = Uin * U_gain + U_offset;
Vout = Vin * V_gain + V_offset;

Reference Registers Map:

Register Name	Register Map	Register description	Range
Pip_y_gain [7:0]	Reg_S3_82 [7:0]	PIP sub Y dynamic range gain value.	(0 ~ 2) * 128
Pip_u_gain [7:0]	Reg_S3_83 [7:0]	PIP sub U dynamic range gain value.	(0~4) * 64

Pip_v_gain [7:0]	Reg_S3_84 [7:0]	PIP sub V dynamic range gain value.	(0~4) * 64
Pip_y_ofst [7:0]	Reg_S3_85 [7:0]	PIP sub Y dynamic rang offset value.	-128 ~ 127
Pip_u_ofst [7:0]	Reg_S3_86 [7:0]	PIP sub U dynamic rang offset value.	-128 ~ 127
Pip_v_ofst [7:0]	Reg_S3_87 [7:0]	PIP sub V dynamic rang offset value.	-128 ~ 127

16.2 RGB to YUV Color Space Conversion

The RGB to YUV color space conversion matrix is:
You can set pip_dyn_byps (Reg_S3_81 [0]) to bypass the matrix.
G = Y - 0.394U - 0.581V
R = Y + 1.14*V
B = Y + 2.032U

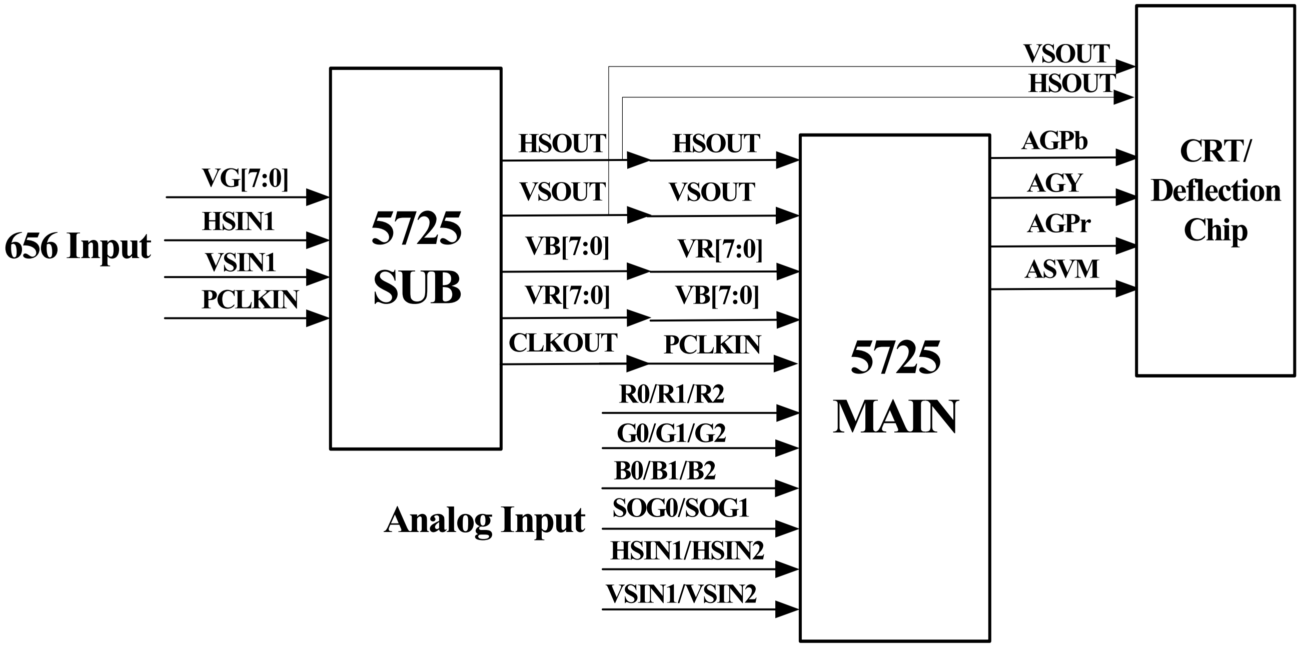
16.3 PIP Window Setting

For PIP window, it based on the horizontal and vertical counter in vds_proc, we can set the PIP window Size and position via setting the following registers:

Register Name	Register Map	Register description
Pip_h_st [11:0]	Reg_S3_89 [3:0] Reg_S3_88 [7:0]	PIP window horizontal start position
Pip_h_sp [11:0]	Reg_S3_8b [3:0] Reg_S3_8a [7:0]	PIP window horizontal stop position
Pip_v_st [10:0]	Reg_S3_8d [2:0] Reg_S3_8c [7:0]	PIP window vertical start position
Pip_v_sp [10:0]	Reg_S3_8f [2:0] Reg_S3_8e [7:0]	PIP window vertical stop position

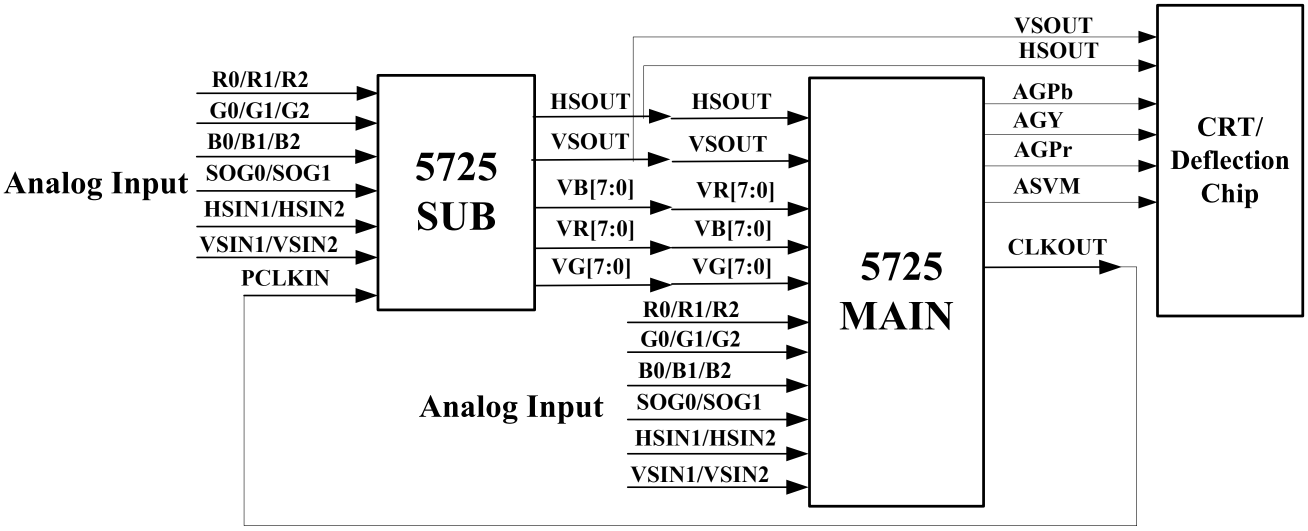
16.4 PIP sub 16-bit digital input

- In MAIN chip, analog video input as main source.
- In MAIN chip, VDS_PROC is working in sync-lock with sub timing.
- In MAIN chip, set VCLK=V2CLK=V4CLK=PCLKIN.
- In SUB chip, 656 digital input as sub source.
- In SUB chip, no interpolation, VCLK = V2CLK = V4CLK
- In this mode, VCLK may have some limitation by PLL.



16.5 PIP sub 24-bit digital input

- In MAIN chip, analog video input as main source.
- In MAIN chip, VDS_PROC is working in sync-lock with sub timing.
- In SUB chip, analog input as sub source.
- In SUB chip, set VCLK=V2CLK=V4CLK=PCLKIN.
- In MAIN chip, no interpolation, VCLK = V2CLK = V4CLK



17 Appendix

17.1 Chip Speed Limitation:

Clock Name	Design Max Frequency
I2CMCK	27MHz
ICLK	80MHz
VCLK	108MHz
MCLK	162MHz

Note: This information is based on the SDF STA result of LGC design!

17.2 How to program dynamic range expansion based on various source

17.2.1 RGB input and RGB output

17.2.1.1 RGB2YUV conversion with input formatter formula

The Input RGB to YUV convert formula is:

Y = 0.587*G +0.114*b+0.299*R;
U = 0.564*(B-Y) +/- 128;
V = 0.713*(R-Y) +/- 128;

So the output YUV2RGB formula should be:

R = Y + 1.4025*(V – 128)
B = Y + 1.773* (U – 128)
G = Y – 0.3443(U –128) – 0.7144(V-128)

A = 1, b = 0, c = 1.4025, d = 0, e = 1.773, f = 0

So:

Y_gain[7:0]	= a * 128 = 128 = 0x80
Y_offset[7:0]	= a*b = 0 = 0x00
U_cos_gain[7:0]	= (32*e)/2.032 = 28 = 0x1c
U_sin_gain [7:0]	= 0 = 0x00
U_offset[7:0]	= (e*f)/2.032 = 0 = 0x00
V_cos_gain[7:0]	= (32*c)/1.14 = 39= 0x27
V_sin_gain [7:0]	= 0 = 0x00
V_offset[7:0]	= (c*d)/1.14 = 0 = 0x00

17.2.1.2 RGB2YUV conversion with decimation filter formula

The Input RGB to YUV convert formula is:

Y = 0.587*G +0.114*b+0.299*R;
U = 0.5625*(B-Y) +/- 128;
V = 0.6875*(R-Y) +/- 128;

So the output YUV2RGB formula should be:

R = Y + 1.454*(V – 128)
B = Y + 1.778* (U – 128)

G = Y – 0.3453(U-128) – 0.7406(V-128)

A = 1, b = 0, c = 1.454, d = 0, e = 1.778, f = 0
So:

Y_gain[7:0]	= a * 128 = 128 = 0x80
Y_offset[7:0]	= a*b = 0 = 0x00
U_cos_gain[7:0]	= (32*e)/2.032 = 28 = 0x1c
U_sin_gain [7:0]	= 0 = 0x00
U_offset[7:0]	= (e*f)/2.032 = 0 = 0x00
V_cos_gain[7:0]	= (32*c)/1.14 = 41= 0x29
V_sin_gain [7:0]	= 0 = 0x00
V_offset[7:0]	= (c*d)/1.14 = 0 = 0x00

17.2.2 YUV input and RGB output

We assume Input RGB to YUV formula (YCbCr) is:
Y = 0.257*R + 0.504*G + 0.098*B + 16
Cb = - 0.14*R – 0.291*G +0.439*B + 128
Cr = 0.439*R – 0.368*G – 0.071*B + 128
(This Cb and Cr are unsigned data)
Then its reverse YUV to RGB formula should be:
R = 1.164*(Y-16) + 1.596*(Cr-128)
G = 1.164*(Y-16) – 0.813*(Cr-128) – 0.392*(Cb-128)
B = 1.164*(Y-16) + 2.017*(Cb-128)

So, a=1.164, b=-16, c=1.596, d=0, e=2.017, f=0
And we should program the dynamic range like this:

Y_gain_cc	=1.164*128 = 0x95
Y_offset_cc	=1.164*(-16) = 0xed
U_gain_cc	=32*2.017/2.032 = 0x20
U_offset_cc	=0
V_gain_cc	=32*1.596/1.14 = 0x2d
V_offset_cc	=0