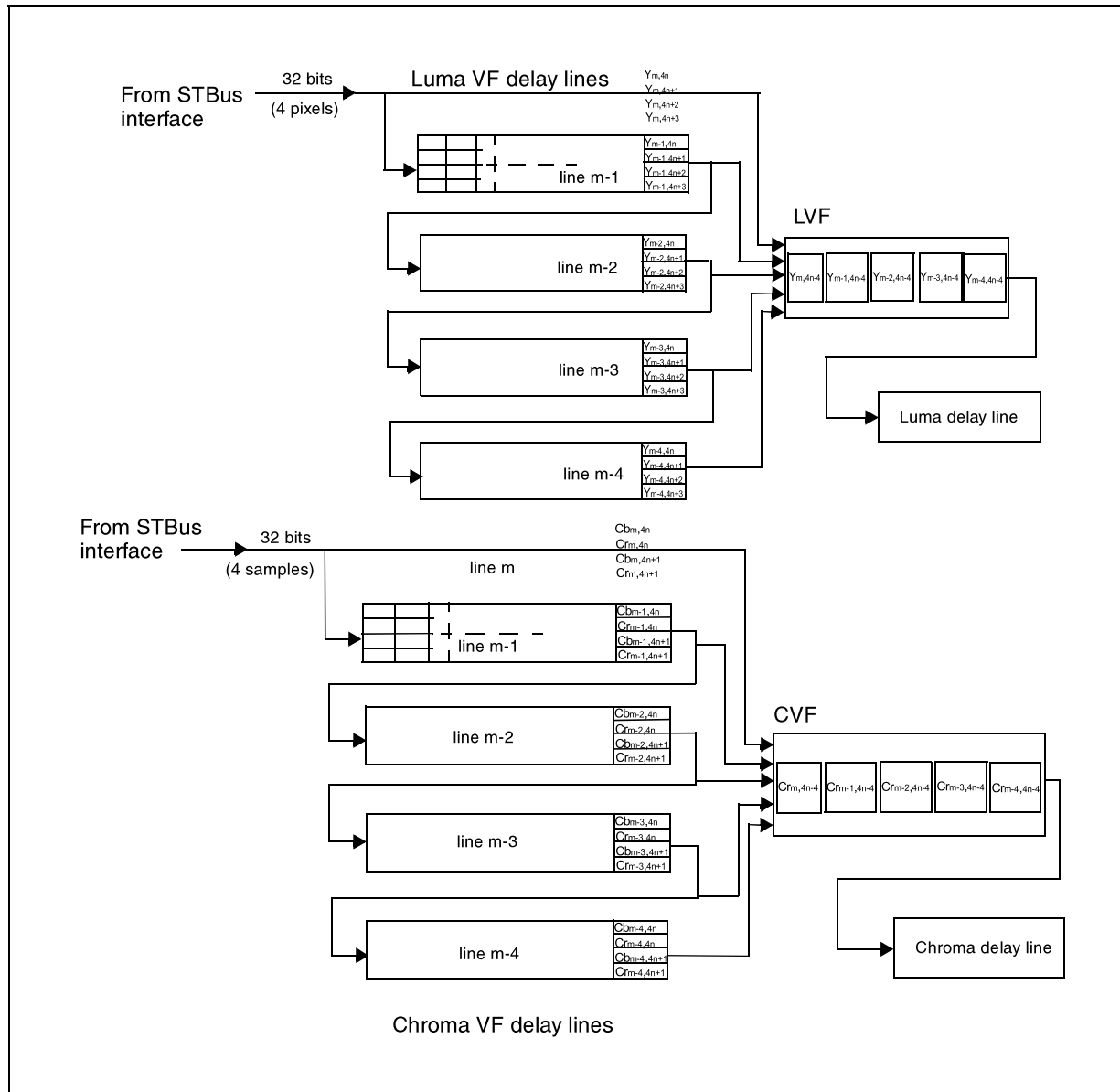


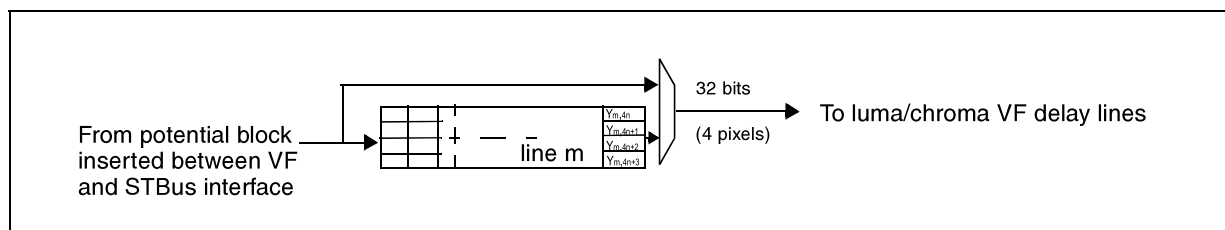
## 49.6.4 Vertical filters (VF)

Figure 160: Vertical filters



Chroma samples are stored as they come: 1 sample of Cb, 1 sample of Cr,... (see Figure 160). Between the delay lines for vertical filtering and VF itself, a small serializer reads the samples four by four, and feeds the VF one sample at a time. This means that the RAMs are read one cycle out of four for vertical filtering.

Figure 161: Luma vertical SRC



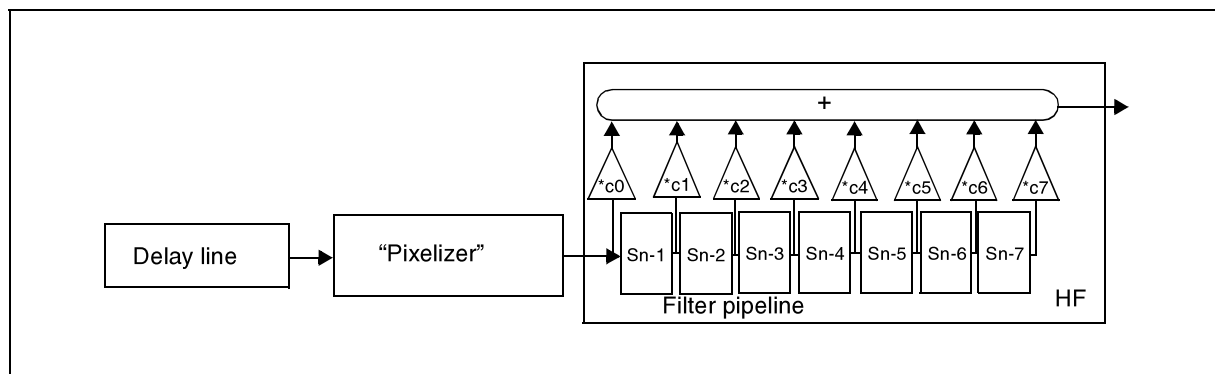
### 49.6.5 Horizontal SRC

Luma and chroma delay lines after vertical filters are built with one single port RAM each. One video line is written in the delay line from the VF, and read by the horizontal filter (HF) at the same time. The arbiter gives the priority to the HF for read.

Each RAM is 480 x 40 bit SPS - 1920 pixels (10 bits per pixel).

Horizontal filtering is represented in [Figure 162](#):

**Figure 162: Horizontal SRC**



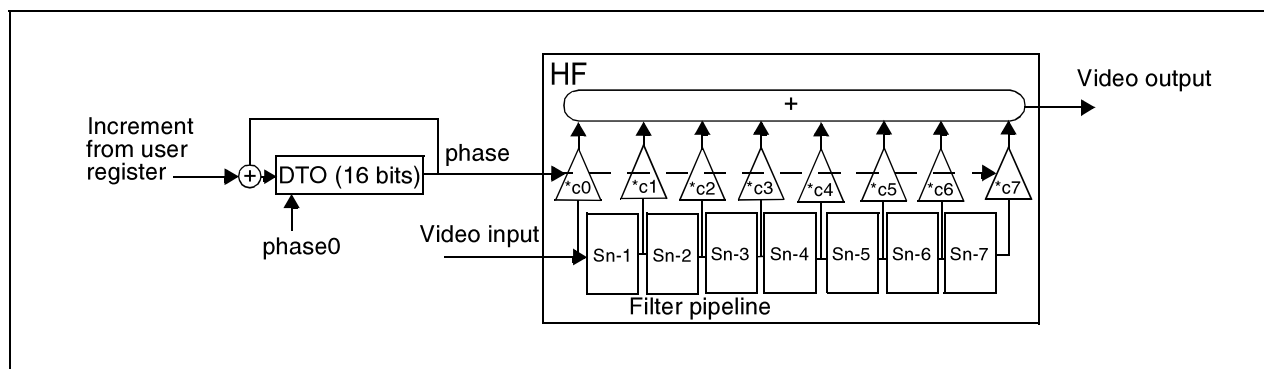
In the worst case (zoom out by four), in luma horizontal filter we should be able to shift the samples in the filter pipeline by up to four and feed up to four new samples from the “pixelizer”. Every clock cycle, the pixelizer must be able to read 0 or 4 samples and output 0, 1, 2, 3 or 4 samples to the filter.

There are two horizontal chroma filters. If the output is 4:4:4, the maximum zoom out is two (programmed zoom out by four, plus upsampling by two). In this case, at maximum, two Cr or Cb samples are fed in the filters each PIX1X clock cycle. If 4:2:2 output format is programmed, zoom out by 4 of chroma is possible. That means 4 new chroma samples have to be loaded in filter pipeline in the worst case. In the same time the output of chroma HF has to be generated one cycle out of two of PIX1X clock.

### 49.6.6 Filter description

All filters are classical polyphase filters. At each clock cycle, an increment value is added to the 13 LSBs of the previous value of DTO, and put as a new value in the DTO register. The value of DTO is used to control the filter pipeline. It is taken from DTO register and  $2^7$  value is added to that value for rounding. The 3 MSBs of that value give the information on how many new video samples are added into the filter pipeline (0, 1, 2, 3 or 4). The next five bits give the set of coefficients to be used in the filter (from 0 to 31; see [Figure 163](#)).

**Figure 163: Polyphase filter block diagram**

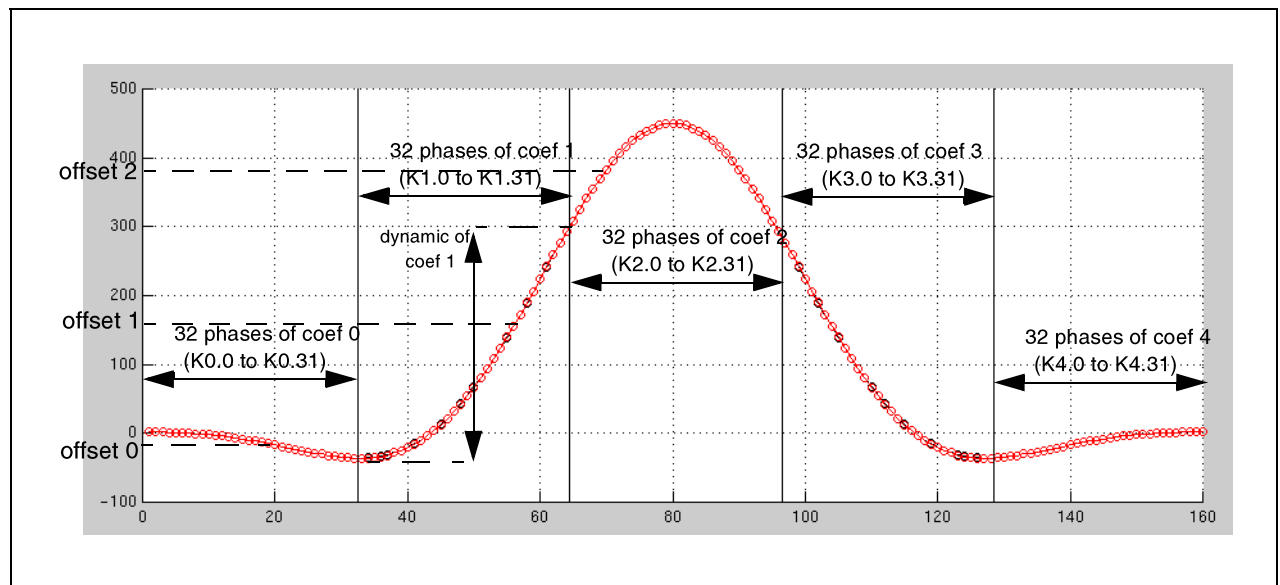


### 49.6.7 Programming of filter coefficients

An example of five tap polyphase filter (VSRC) set of coefficients is shown in Figure 155. There are 5\*32 coefficients on sine curve which is shown. It is easy to see that it is symmetrical curve, so only half of coefficients need to be stored in memory. All 5\*32 coefficients are 10-bits values. Nevertheless, 8 or 9 bits are enough for the dynamic of each group of 32 coefficients (32 phases of a coefficient in 5 tap filter). That is why each coefficient is defined by 3 values:

- 8-bit signed value,
- 10-bit offset value, common to all 32 phases (in fact 64 because of symmetry), and
- 1-bit shift value, common to all 32 phases (64 because of symmetry) which means that an 8-bit value is multiplied by 2 or not to cover the dynamic of 32 phases (coef1 set of 32 phases in the above example).

Figure 164: Example: five tap polyphase filter (VSRC) set of coefficients



### 49.6.8 Programming of source/target size and increment

The increment of vertical luma filter should be  $\text{Height}_{\text{SOURCE}} / \text{Height}_{\text{TARGET}} * 2^{13}$ . If the input format is YCbCr 4:2:2, then the increment (and initial phase) for vertical chroma filter must be the same as luma one. If input format is YCbCr 4:2:0 macro block then the increment of vertical chroma filter is the half of vertical luma filter increment because chroma upsampling is done in the same time as filtering (4:2:0 -> 4:2:2).

Increment of horizontal luma filter should be  $\text{Width}_{\text{SOURCE}} / \text{Width}_{\text{TARGET}} * 2^{13}$ . For the horizontal chroma filter, the increment should be half of the luma increment because the horizontal upsampling is done in the same time (4:2:2 -> 4:4:4).

The minimum value for source height is 5, for width, 34 (17 samples of chroma).

The increment value and initial phase are used to generate pixels at the output. The DISP\_TARGET\_SIZE register determines the number of pixels/lines generated. This means, if the increment does not respect the ratio between target and source size, the generation of pixels will be stopped after target size is reached, or the last pixel is repeated to reach the target size. In the case where it should be stopped earlier then all pixels of source picture are consumed, VF and HF continue to send the requests to previous block until all pixels of the source picture are sent (until the last pixel/line is received). This is useful where the de-interlacer operates before the vertical filters because it needs to clean all the pixels of previously processed line from its FIFOs.

### 49.6.9 Nonlinear zoom

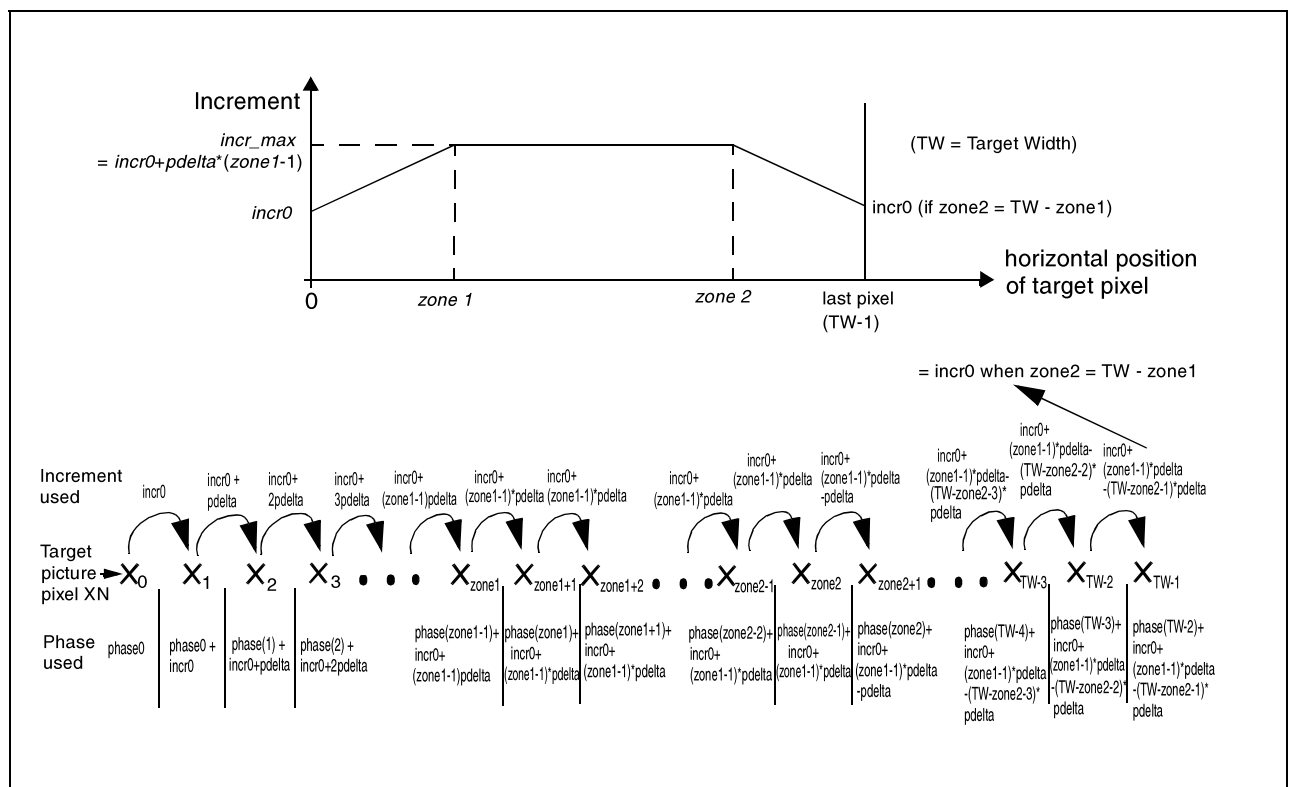
A typical application of nonlinear zoom is to display a 4:3 picture on a 16:9 screen, with minimum deformation of the central part of the picture and greater deformation of the left and right borders. This feature is implemented by linearly changing the increment (zoom factor) **only** in horizontal SRC at the beginning and at the end of the screen as shown in [Figure 165](#).

The first generated pixel uses *phase0* to find the corresponding coefficients; the second uses *phase0 + incr0*; the third uses *phase1 + incr0 + pdelta* and so on. *phase0* and *incr0* are the initial phase and increment as defined by registers DISP\_CHR\_HSRC and DISP\_LUMA\_HSRC (defined for nonlinear zoom). *pdelta* is the 'increment of the increment' defined by register DISP\_PDELTA. Zones where nonlinear zoom is applied are defined by pixel/samples zone 1 and zone 2 as shown. This means that the increment increases from pixel/sample 0 to pixel/sample zone 1, and decreases from pixel/sample zone 2 to the last pixel/sample of the target picture.

**Note:** The maximum value of increment is  $2^{15}$  (0x8000), which corresponds to zoom out by 4. If this value is reached before zone1, then *incr\_max* is  $2^{15}$  and the increment stops growing. When zone2 is reached, the increment goes down, and at the end of picture it reaches a value smaller than *incr0* (when  $\text{zone2} = \text{TW} - \text{zone1}$ ). This should be avoided, by programming values of *incr0*, *pdelta* and *zone1* that do not give  $\text{incr\_max} = 2^{15}$  before zone 1.

Zones 1 and 2 are defined by registers DISP\_NLZZD\_Y and DISP\_NLZZD\_C respectively for luma and chroma HSRC.

**Figure 165: Increment function for nonlinear zoom**



### 49.6.10 Output format

The display can respond to a request from the compositor by sending pixels in either 4:4:4 or 4:2:2 format.

**4:4:4 output format** means there will be luma and chroma sent back to the compositor request every PIX1X clock cycle.

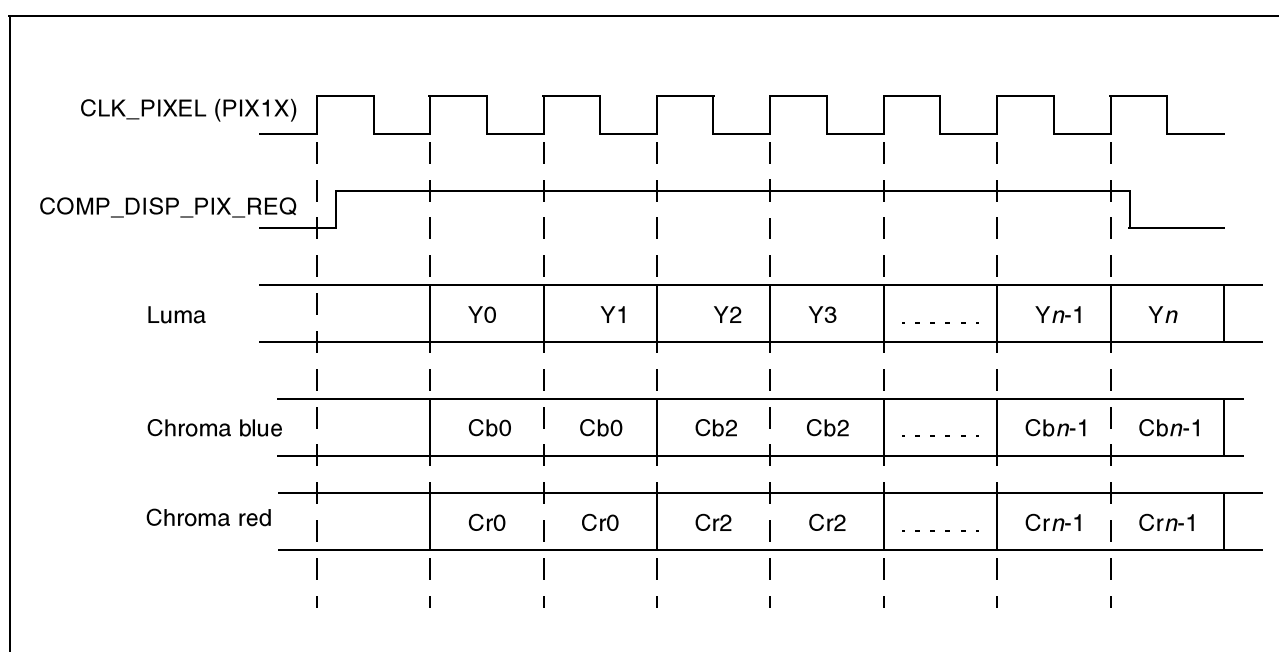
**4:2:2 output format** means there will be luma sent back every PIX1X clock cycle but chroma is sent back one cycle out of two PIX1X clock cycles.

The number of pixels per line provided to or by the VFC & HFC block is always even.

The first pixel going in the VFC and HFC is always an existing pixel (for example Y0, Cb0, Cr0). The block never starts on a pixel without chroma Cb Cr data.

In [Figure 166](#), variable  $n$  is odd (for example if we have to display four pixels,  $n = 3$ ). The chroma  $cb(n-1)$  and  $cr(n-1)$  pixels are repeated for luma  $Y_n$  pixel.

**Figure 166: 4:2:2 output format**



## 49.7 Soft reset

Soft reset clears the pipeline and prepares the block for a new field/frame properly: it reset all FSMs, delay lines, pipeline flags (first/last pixel/line flags...), sample/line counters and stops all STBus requests. No user register is reset by soft reset. There is no specific user register bit for soft reset but it is done on each vertical synchronization active edge ('automatic' soft reset).

## 50 Main and auxiliary display registers

All control registers are double buffered and are accessible through the STBus interface. Registers containing the filter coefficients (DISP\_L(C)H(V)F\_COEF) are loaded directly from memory and are not double buffered.

Register addresses are provided as either

*MainDisplayBaseAddress* + offset,  
*AuxDisplayBaseAddress* + offset or  
*LMUBaseAddress* + offset.

The *MainDisplayBaseAddress* is:

0x1900 2000.

The *AuxDisplayBaseAddress* is:

0x1900 3000.

The *LMUBaseAddress* is:

0x1900 4000.

**Table 161: Main and auxiliary display processor register summary**

Register	Description	Offset	Type
DISP_CTRL	DISP control	0x000	R/W
DISP_LUMA_HSRC	DISP luma horizontal source parameters	0x004	R/W
DISP_LUMA_VSRC	DISP luma vertical source parameters	0x008	R/W
DISP_CHR_HSRC	DISP chroma horizontal source FSM parameters	0x00C	R/W
DISP_CHR_VSRC	DISP chroma vertical source parameters	0x010	R/W
DISP_TARGET_SIZE	DISP target pixmap size	0x014	R/W
DISP_NLZZD_Y	DISP nonlinear zoom - zone definition for luma	0x018	R/W
DISP_NLZZD_C	DISP nonlinear zoom - zone definition for chroma	0x01C	R/W
DISP_PDELTA	DISP nonlinear zoom - increment step definition	0x020	R/W
Reserved	-	0x024 - 0x07C	-
DISP_MA_CTRL	DISP memory access control	0x080	R/W
DISP_LUMA_BA	DISP luma source pixmap memory location	0x084	R/W
DISP_CHR_BA	DISP chroma source pixmap memory location	0x088	R/W
DISP_PMP	DISP pixmap memory pitch	0x08C	R/W
DISP_LUMA_XY	DISP luma first pixel source position	0x090	R/W
DISP_CHR_XY	DISP chroma first pixel source position	0x094	R/W
DISP_LUMA_SIZE	DISP memory source pixmap size (luma)	0x098	R/W
DISP_CHR_SIZE	DISP memory source pixmap size (chroma)	0x09C	R/W
DISP_HFP	DISP horizontal filters pointer	0x0A0	R/W
DISP_VFP	DISP vertical filters pointer	0x0A4	R/W
Reserved	-	0x0A8 - 0x0F8	-
DISP_PKZ	DISP maximum packet size	0x0FC	R/W
DISP_LHF_COEF	DISP luma horizontal filter coefficients	0x100 - 0x188	RO/LLU
DISP_LVF_COEF	DISP luma vertical filter coefficients	0x18C - 0x1E0	RO/LLU
DISP_CHF_COEF	DISP chroma horizontal filter coefficients	0x200 - 0x288	RO/LLU
DISP_CVF_COEF	DISP chroma vertical filter coefficients	0x28C - 0x2E0	RO/LLU

Confidential

Filter coefficients size: 476 bytes, others 84 bytes - **total: 560 bytes.**

**Table 162: LMU register map**

Register	Description	Offset	Type
LMU_CTRL	LMU control	0x00	R/W
LMU_LMP	LMU luma buffer start pointer	0x04	R/W
LMU_CMP	LMU chroma buffer start pointer	0x08	R/W
LMU_BPPL	LMU number of block-pairs per line	0x0C	R/W
LMU_CFG	LMU configuration	0x10	R/W
LMU_VINL	LMU number of input lines	0x14	R/W
LMU_MRS	LMU minimum space between STBus requests	0x18	R/W
LMU_CHK	LMU maximum chunk size	0x1C	R/W
LMU_STA	LMU status	0x20	RO
LMU_ITM	LMU interrupt mask	0x24	R/W
LMU_INT_STA	LMU interrupt status	0x28	RO
LMU_AFD	LMU accumulated field difference	0x2C	R/W

## 50.1 Main and auxiliary display registers

The following registers are duplicated for both main and auxiliary display processors. Their addresses are described as *DisplayBaseAddress* + offset, where *DisplayBaseAddress* is either *MainDisplayBaseAddress* or *AuxDisplayBaseAddress*.

### DISP\_CTRL

### DISP control

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISP_EN		HF_UPDATE_EN		VF_UPDATE_EN		Reserved					BIGTNOTLITTLE		CHF_EN		YHF_EN		4:2:2_OUT		Reserved												

Address: *DisplayBaseAddress* + 0x000

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the operating mode of the DISP pipe, for the current viewport display.

[31] **DISP\_EN**: DISP bloc enable<sup>a</sup>

0: DISP is disabled (no output is generated)      1: DISP is enabled

[30] **HF\_UPDATE\_EN**: Horizontal filter update enable

0: The coefficients for the H filters are not loaded

1: The coefficients for the H filters are updated from memory after next vsync

This bit is autocleared after it is taken into account (on next Vsync)

[29] **VF\_UPDATE\_EN**: Vertical filter update enable

0: The coefficients for the V filters are not loaded

1: The coefficients for the V filters are updated from memory after next vsync

This bit is autocleared after it is taken into account (on next VSYNC)

[28:24] **Reserved**

[23] **BIGTNOTLITTLE**: Bitmap

0: Little endian bitmap

1: Big endian bitmap<sup>b</sup>

[22] **CHF\_EN**

0: Chroma HF is disabled (coefficients are 0001 0000 for all DTO phases)

1: Chroma HF is enabled (coefficients taken from DISP\_CHF\_COEF registers)<sup>c</sup>

[21] **YHF\_EN**

0: Luma HF is disabled (coefficients are 0001 0000 for all DTO phases)

1: Luma HF is enabled (coefficients taken from DISP\_LHF\_COEF registers)

[20] **4:2:2\_OUT**

0: 4:4:4 output format (luma and chroma output each cycle of PIX1X clock)

1: 4:2:2 output format (chroma output only one cycle out of two of PIX1X clock)

[19:0] **Reserved**

- This bit is taken into account after next vsync active edge. All STBus requests are stopped properly and video pipeline is purged. If power down is to be done (clock off) then it should be done after vsync active edge following this one.
- This concerns the video picture pixmap loaded from memory when 4:2:2 R input format is selected.
- This feature is needed only for horizontal SRC and not for vertical, because in the vertical SRC this can be achieved by software (programming coefficients 0 0100 for all 32 phases). In HSRC because of symmetry and **even** number of taps what can be programmed is 0001 1000 and not 0001 0000!



**DISP\_LUMA\_HSRC****DISP luma horizontal source parameters**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FYP_RPT			HSRC_INIT_PHASE															HSRC_INC													

Address: *DisplayBaseAddress* + 0x004

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the configuration for the horizontal sample rate conversion of luma samples.

[31:29] **FYP\_RPT**: First luma pixel repeat

First pixel is repeated FYP\_RPT times when loaded into luma HF. This is a 3-bit nonsigned value with max value of four (1xx values are taken as four - 100). For example:

001: First pixel is repeated once (this means for the first HF output sample generation there are two occurrences of first pixel in HF pipe)

[28:16] **HSRC\_INIT\_PHASE**

The horizontal sample rate converter state-machine initial phase, 0.13 format.

[15:0] **HSRC\_INC**

The horizontal sample rate converter state-machine increment, in 3.13 format. Maximum value is 100.0000 0000 0000 0. Any value 1xx.xxxx xxxx xxxx x is taken as 100.0000 0000 0000 0

**DISP\_LUMA\_VSRC****DISP luma vertical source parameters**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FYLR	VSRC_INIT_PHASE															VSRC_INC														

Address: *DisplayBaseAddress* + 0x008

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the configuration for the vertical sample rate converter.

[31] **Reserved**

[30:29] **FYLR**: First luma line repeat

First line is repeated FYLR times when loaded into luma VF. This is 2-bit nonsigned value with a maximum value of 2 (11 value is taken as two - 10). For example:

11: First line is repeated twice (this means for the first VF output line generation there are three occurrences of the first line in the VF pipe)

[28:16] **VSRC\_INIT\_PHASE**

The vertical sample rate converter state-machine initial phase, 0.13 format.

[15:0] **VSRC\_INC**

The vertical sample rate converter state-machine increment, in 3.13 format. Maximum value is 100.0000 0000 0000 0. Any value 1xx.xxxx xxxx xxxx x is taken as 100.0000 0000 0000 0

**DISP\_CHR\_HSRC****DISP chroma horizontal source FSM parameters**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCPR			HSRC_INIT_PHASE															HSRC_INC													

Address: *DisplayBaseAddress* + 0x00C

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: The DISP horizontal chroma sample rate converter's final state machine parameters register provides the configuration for the horizontal sample rate converting of chroma components.

[31:29] **FCP\_RPTR**

First chroma pixel is repeated FCP\_RPT (First chroma pixel repeat) times when loaded into chroma HF. This is 3 bit nonsigned value with max value of four (1xx values are taken as four - 100). For example: 011: First pixel is repeated three times (this means for the first HF output sample generation there are four occurrences of first chroma pixel in HF pipe)

[28:16] **HSRC\_INIT\_PHASE**

The horizontal sample rate converter state-machine initial phase, 0.13 format

[15:0] **HSRC\_INC**

The horizontal sample rate converter state-machine increment, in 3.13 format. Maximum value is 010.0000 0000 0000 0. Any greater value is taken as 010.0000 0000 0000 0

**DISP\_CHR\_VSRC****DISP chroma vertical source parameters**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FCLR		VSRC_INIT_PHASE													VSRC_INC															

Address: *DisplayBaseAddress* + 0x010

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: The DISP chroma vertical sample rate converter's parameters register provides the configuration for vertical sample rate converting of chroma components<sup>1</sup>.

[31] **Reserved**

[30:29] **FCLR**

First chroma line is repeated FCLR (First chroma line repeat) times when loaded into chroma VF. This is 2 bit nonsigned value with max value of two (11 value is taken as two - 10). For example: 00: First line is not repeated (this means for the first VF output line generation there is only one occurrence of first chroma line in VF pipe)

[28:16] **VSRC\_INIT\_PHASE**

The vertical sample rate converter state-machine initial phase, 0.13

[15:0] **VSRC\_INC**

The vertical sample rate converter state-machine increment, in 3.13 format. Maximum value is 100.0 0000 0000 0000. Any value 1xx.x xxxx xxxx xxxx is taken as 100.0 0000 0000 0000

1. If 4:2:2 R input format is set, VSRC Increment, VSRC Initial Phase and F<sub>CPR</sub> must be programmed as luma (DISP\_CHR\_VSRC and DISP\_LUMA\_VSRC must have the same value)

**DISP\_TARGET\_SIZE****DISP target pixmap size**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	HEIGHT	Reserved	WIDTH
----------	--------	----------	-------

Address: *DisplayBaseAddress* + 0x014

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the size of resized picture to be displayed.

[31:27] **Reserved**

[26:16] **HEIGHT**: Pixmap height in lines, defined as the resize resulting number of lines that is to be displayed.

[15:11] **Reserved**

[10:0] **WIDTH**: Pixmap width in pixels

**DISP\_NLZZD\_Y****DISP nonlinear zoom - zone definition for luma**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	LUMA_ZONE1	Reserved	LUMA_ZONE2
----------	------------	----------	------------

Address: *DisplayBaseAddress* + 0x018

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the borders of nonlinear zoom in the target picture for luma in pixel units; see [Section 49.6.9: Nonlinear zoom on page 504](#).

[31:27] **Reserved**

[26:16] **LUMA\_ZONE1**: Limit between first nonlinear zone and linear zone of the target picture in pixels

[15:11] **Reserved**

[10:0] **LUMA\_ZONE2**: Limit between linear zone and second nonlinear zone of the target picture in pixels

Confidential

**DISP\_NLZZD\_C****DISP nonlinear zoom - zone definition for chroma**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CHROMA_ZONE1										Reserved						CHROMA_ZONE2									

Address: *DisplayBaseAddress* + 0x01C

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the borders of nonlinear zoom in the target picture for chroma in output chroma sample units; see [Section 49.6.9: Nonlinear zoom on page 504](#).

[31:27] **Reserved**

[26:16] **CHROMA\_ZONE1**

Limit between first nonlinear zone and linear zone of the target picture in output chroma sample units.

[15:11] **Reserved**

[10:0] **CHROMA\_ZONE2**

Limit between linear zone and second nonlinear zone of the target picture in output chroma sample units.

**DISP\_PDELTA****DISP nonlinear zoom - increment step definition**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDELTA_LUMA																PDELTA_CHROMA															

Address: *DisplayBaseAddress* + 0x020

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the steps used to increment luma and chroma increments in HSRCs for nonlinear zooms; see [Section 49.6.9: Nonlinear zoom on page 504](#). If this feature is not used, PDELTA\_LUMA and PDELTA\_CHROMA should be 0 (linear zoom).

[31:16] **PDELTA\_LUMA**

Step with which the luma increment of luma HSRC is incremented within the range [pixel0, pixel luma\_zone1] and decremented within the range [pixel luma\_zone2, last pixel]. pixel0, and last pixel correspond to first and last chroma sample of target picture.

[15:0] **PDELTA\_CHROMA**

Step with which the chroma increment of chroma HSRC is incremented within the range [sample0, sample chroma\_zone1] and decremented within the range [sample chroma\_zone2, last sample]. Sample0 and last sample correspond to first and last chroma sample of target picture.

## DISP\_MA\_CTRL

## DISP memory access control

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	MB_FIELDC	MIN_SPC_BETWEEN_REQS	PIX_LOAD_LINE	COEF_LOAD_LINE	MB_FIELDY	IN_FORMAT
----------	-----------	----------------------	---------------	----------------	-----------	-----------

Address: *DisplayBaseAddress* + 0x080

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides memory access control of the DISP.

[31:27] **Reserved**

[26] **MB\_FIELDC**

Chroma access mode in macro-block organized frame buffers (YCbCr420 MB and YCbCr422 MB format)

0: Access in frame mode

1: Access in field mode (every other line)

[25:16] **MIN\_SPC\_BETWEEN\_REQS**

Defines the minimum number of STBus cycles between two memory messages (bursts).<sup>a</sup>

The number of cycles between two messages is max('Min Space Between Reqs' + 1, message duration)

[15:11] **PIX\_LOAD\_LINE**

Defines how many lines STBus plug waits after vsync before sending the first request in the field to load the display pipe. When changing the coefficient set in VSRC and HSRC the pixel request should be sent after new coefficient set is loaded, that is, Pix load line > Coef load line (see above). If this is not the case, then the display pipe will be loaded just after new coefficients load.

[10:6] **COEF\_LOAD\_LINE**

Video line number after VSYNC, during which the filter coefficients are loaded via the STBus interface (buffer 1) when V/HFilter update enable is 1. This is a nonsigned value from 0 to 31. Loading of new coefficients is done after Coef Load Line's hsync active edge.

[5] **MB\_FIELDY**

Luma access mode in macro-block organized frame buffers for (YCbCr420 MB and YCbCr422 MB format)

0: Access in frame mode

1: Access in field mode (every other line)

[4:0] **IN\_FORMAT**: Input format

10010: YCbCr4:2:2 R 0x12

10011: YCbCr4:2:2 MB (or 4:2:2 MB) 0x13

10100: YCbCr4:2:0 MB (or 4:2:0 MB) 0x14

- a. For example, in the case of vertical zoom out, four video lines need to be loaded from time to time during the display of one video line (64  $\mu$ s) which means 4 x 45 requests of 128 bit words of luma (12 messages) should be done and there is 64  $\mu$ s/133 MHz = 8512 STBus cycles. So, these requests can be spaced by 8512/12 = 709 STBus clock cycles (by putting 709 value in register, the 'peaks' on the STBus can be avoided). The counter used for this purpose is reset by the STBus request

**DISP\_LUMA\_BA****DISP luma source pixmap memory location**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK_NUM						PIXMAP_MEM_PTR_(LUMA)																									

Address: *DisplayBaseAddress* + 0x084

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Gives the location of first Y of the FRAME in 4:2:0 MB and 4:2:2 MB mode or first Cb or Cr of the field in 4:2:2 R mode (of the frame in progressive video mode). It contains the memory location for the first pixel (top-left corner) of the source: of first luma in the frame when 4:2:0 MB and 4:2:2 MB input format is used or first Cb or Cr in the field when 4:2:2 R input format is used. In 4:2:2 R mode, the memory address for pixel [0,0] must be aligned on a 32-bit word address boundary (2 LSBs at 0). In 4:2:X MB mode, the memory address for pixel [0,0] must be aligned on a 2048-bit word address boundary (8 LSBs at 0).

[31:26] **64MB\_BANK\_NUM**: 64 MByte bank number

[25:0] **PIXMAP\_MEM\_PTR\_(LUMA)**: Pixel map memory pointer (luma)

First Y/YCbCr pixel byte address, in the selected 64 MByte bank (note that the whole bitmap to be displayed must be totally included into the same bank)

**DISP\_CHR\_BA****DISP chroma source pixmap memory location**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK_NUM						PIXMAP_MEM_PTR_(CHROMA)																									

Address: *DisplayBaseAddress* + 0x088

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Gives the location of first chroma sample of the FRAME in 4:2:0 MB and 4:2:2 MB mode. The memory address for pixel [0,0] must be aligned on a 2048-bit word address boundary (8 LSBs at 0).

[31:26] **64MB\_BANK\_NUM**: 64 MByte bank number

[25:0] **PIXMAP\_MEM\_PTR\_(CHROMA)**: Pixel map memory pointer (chroma)

First chroma byte address, in the selected 64 MByte bank (note that the whole bitmap to be displayed must be totally included into the same bank)

Confidential

## DISP\_PMP

## DISP pixmap memory pitch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PITCH_VAL															

Address: *DisplayBaseAddress* + 0x08C

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: The DISP pixmap memory pitch for the displayed pixmap, as stored in the memory, in 4:2:0 MB and 4:2:2 MB format or of YCrCb in 4:2:2 R format, in bytes.

Note: *The pitch is the distance inside the memory, in bytes, between two vertically adjacent samples.*

In 4:2:0 MB and 4:2:2 MB format, the picture is always stored in FRAME format and the pitch gives the number of luma pixels per line rounded to higher integer number of macroblocks. If the line length in luma pixel unit is  $16 \cdot n + m$  (where  $0 < m < 16$ ), the pitch value should be programmed to  $16 \cdot (n+1)$  (the four LSBs are always at 0).

In 4:2:2 R format, the pitch value is always used as it is (in field access it defines the distance in bytes between two vertically adjacent samples of the same field and in frame access it defines the distance between two vertically adjacent samples in the frame). The pitch value must be a multiple of four bytes (the two LSBs are always 0).

## DISP\_LUMA\_XY

## DISP luma first pixel source position

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Y1								Reserved								X1							

Address: *DisplayBaseAddress* + 0x090

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register gives the position of first source pixel when 4:2:2 R input format is used or first luma sample when 4:2:0 MB or 4:2:2 MB input format is used. Register provides XY location in pixel unit of the first source pixel to be accessed in memory comparing to the top left corner of complete source image (0, 0) defined by DISP\_LUMA\_BA (see [Figure 153: Source and target definition on page 496](#)).

[31:27] **Reserved**

[26:16] **Y1**: First source image line number in complete original picture that should be accessed from memory.

[15:11] **Reserved**

[10:0] **X1**: First source image pixel number in complete original line that should be accessed from memory.

**DISP\_CHR\_XY****DISP chroma first pixel source position**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					Y2												Reserved					X2									

Address: *DisplayBaseAddress* + 0x094

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: The DISP position of first chroma sample of the source picture that will be loaded and resized in DISP when 4:2:0 MB or 4:2:2 MB is the input format. Register provides XY location in chroma sample unit of the first source chroma sample to be accessed in memory comparing to the top left corner of complete source image (0, 0) defined by DISP\_CHR\_BA. Example: X2 = 5 means 6th Cr and 6th Cb are the first chroma samples in the line.

[31:27] **Reserved**

[26:16] **Y2**

First source image chroma line number in complete original picture that should be accessed from memory

[15:11] **Reserved**

[10:0] **X2**

First source image chroma pixel number in complete original line that should be accessed from memory

**DISP\_LUMA\_SIZE****DISP memory source pixmap size (luma)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					HEIGHT1												Reserved					WIDTH1									

Address: *DisplayBaseAddress* + 0x98

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: The DISP memory source pixmap size register provides the size of the source pixmap in pixel unit (part of complete source picture to be loaded from memory) when 4:2:2 R input format is used or pixmap size in luma pixel unit when 4:2:0 MB or 4:2:2 MB format is used. Minimum allowed value of WIDTH1 is 34. Minimum allowed value of HEIGHT1 is 5.

[31:27] **Reserved**

[26:16] **HEIGHT1**

Pixmap height, in lines, being defined as the number of video lines that must be read from memory

[15:11] **Reserved**

[10:0] **WIDTH1**: Pixmap width in pixels

Confidential



**DISP\_CHR\_SIZE****DISP memory source pixmap size (chroma)**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	HEIGHT2	Reserved	WIDTH2
----------	---------	----------	--------

Address: *DisplayBaseAddress* + 0x09C

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: The DISP memory source pixmap size register provides the size of the source pixmap in chroma sample unit (part of complete source picture to be loaded from memory) when 4:2:0 MB or 4:2:2 MB input format is used. Minimum allowed value of HEIGHT2 is 5. Minimum allowed value of WIDTH2 is 16. Example: WIDTH2 =  $n$  means there are  $n$  Cb and  $n$  Cr to be read!

[31:27] **Reserved**

[26:16] **HEIGHT2**

Pixmap height in chroma lines, being defined as the number of chroma lines that must be read from memory.

[15:11] **Reserved**

[10:0] **WIDTH2**: Pixmap width in chroma samples

**DISP\_HFP****DISP horizontal filters pointer**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

64MB_BANK_NUM	H_FILTER_PTR	Reserved
---------------	--------------	----------

Address: *DisplayBaseAddress* + 0x0A0

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Contains a memory pointer to the set of filter coefficients that must be used for the horizontal sample rate converters (Luma and Chroma).  
The coefficients are loaded only if DISP\_CTRL[30]= 1 (HFilter update enable) after the next vertical synchronization pulse.

[31:26] **64MB\_BANK\_NUM**: 64 MByte bank number

[25:4] **H\_FILTER\_PTR**

The four LSBs address bits are “don’t care”, because **the filter coefficients structure must be aligned on a 128-bit word boundary**

Memory location where to retrieve the filter coefficients (35\*2 32-bit words that must be fully contained in the specified bank). First 35 32-bit words are luma coefficients, then next thirty-five 32-bit words are chroma coefficients for HSRC.

[3:0] **Reserved**

Confidential

**DISP\_VFP****DISP vertical filters pointer**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK_NUM										V_FILTER_PTR																			Reserved		

Address: *DisplayBaseAddress* + 0x0A4

Type: R/W

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Contains a memory pointer to the set of filter coefficients that must be used for the vertical sample rate converters (Luma and chroma).

The coefficients are loaded only if DISP\_CTRL[29] = 1 (VFilter Update enable) after next vertical synchronization pulse.

[31:26] **64MB\_BANK\_NUM**: 64 MByte bank number

[25:4] **V\_FILTER\_PTR**

4 LSBs address bits are “don’t care”, because **the filter coefficients structure must be aligned on a 128-bit word boundary**

Memory location where to retrieve the filter coefficients (22\*2 32-bit words that must be fully contained in the specified bank). First 22 32-bit words are luma coefficients, then next twenty-two 32-bit words are chroma coefficients for VSRC.

[3:0] **Reserved**

**DISP\_PKZ****DISP maximum packet size**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																									BIGNOTLITTLE	Reserved	MAX_PKT_SIZE				

Address: *DisplayBaseAddress* + 0x0FC

Type: R/W

Buffer: Immediate

Reset: 0

Description: The DISP PKZ register is a 5-bit register for controlling the static parameters of the DISP pipelines.

[31:6] **Reserved**

[5] **BIGNOTLITTLE**: CPU endianness

0: Little endian CPU

1: Big endian CPU<sup>a</sup>

[4:3] **Reserved**

[2:0] **MAX\_PKT\_SIZE**: Maximum packet size during an STBus transaction

000: Message size (16 packets)

001: 16 STBus words

010: 8 STBus words

011: 4 STBus words

100: 2 STBus words

101: 1 STBus words

others: Reserved

a. This concerns filter coefficients load from memory

## DISP\_LHF\_COEF

## DISP luma horizontal filter coefficients

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

0x00	K 7.3 (K 0.29)			K 7.2 (K 0.30)			K 7.1 (K 0.31)			K 7.0		
0x04	K 7.7 (K 0.25)			K 7.6 (K 0.26)			K 7.5 (K 0.27)			K 7.4 (K 0.28)		
0x08	K 7.11 (K 0.21)			K 7.10 (K 0.22)			K 7.9 (K 0.23)			K 7.8 (K 0.24)		
0x0C	K 7.15 (K 0.17)			K 7.14 (K 0.18)			K 7.13 (K 0.19)			K 7.12 (K 0.20)		
0x10	K 7.19 (K 0.13)			K 7.18 (K 0.14)			K 7.17 (K 0.15)			K 7.16 (K 0.16)		
0x14	K 7.23 (K 0.9)			K 7.22 (K 0.10)			K 7.21 (K 0.11)			K 7.20 (K 0.12)		
0x18	K 7.27 (K 0.5)			K 7.26 (K 0.6)			K 7.25 (K 0.7)			K 7.24 (K 0.8)		
0x1C	K 7.31 (K 0.1)			K 7.30 (K 0.2)			K 7.29 (K 0.3)			K 7.28 (K 0.4)		
0x20	K 6.3 K(1.29)			K 6.2 K(1.30)			K 6.1 K(1.31)			K 6.0		
0x24	K 6.7 K(1.25)			K 6.6 K(1.26)			K 6.5 K(1.27)			K 6.4 K(1.28)		
0x28	K 6.11 K(1.21)			K 6.10 K(1.22)			K 6.9 K(1.23)			K 6.8 K(1.24)		
0x2C	K 6.15 K(1.17)			K 6.14 K(1.18)			K 6.13 K(1.19)			K 6.12 K(1.20)		
0x30	K 6.19 K(1.13)			K 6.18 K(1.14)			K 6.17 K(1.15)			K 6.16 K(1.16)		
0x34	K 6.23 K(1.9)			K 6.22 K(1.10)			K 6.21 K(1.11)			K 6.20 K(1.12)		
0x38	K 6.27 K(1.5)			K 6.26 K(1.6)			K 6.25 K(1.7)			K 6.24 K(1.8)		
0x3C	K 6.31 K(1.1)			K 6.30 K(1.2)			K 6.29 K(1.3)			K 6.28 K(1.4)		
0x40	K 5.3 K(2.29)			K 5.2 K(2.30)			K 5.1 K(2.31)			K 5.0		
0x44	K 5.7 K(2.25)			K 5.6 K(2.26)			K 5.5 K(2.27)			K 5.4 K(2.28)		
0x48	K 5.11 K(2.21)			K 5.10 K(2.22)			K 5.9 K(2.23)			K 5.8 K(2.24)		
0x4C	K 5.15 K(2.17)			K 5.14 K(2.18)			K 5.13 K(2.19)			K 5.12 K(2.20)		
0x50	K 5.19 K(2.13)			K 5.18 K(2.14)			K 5.17 K(2.15)			K 5.16 K(2.16)		
0x54	K 5.23 K(2.9)			K 5.22 K(2.10)			K 5.21 K(2.11)			K 5.20 K(2.12)		
0x58	K 5.27 K(2.5)			K 5.26 K(2.6)			K 5.25 K(2.7)			K 5.24 K(2.8)		
0x5C	K 5.31 K(2.1)			K 5.30 K(2.2)			K 5.29 K(2.3)			K 5.28 K(2.4)		
0x60	K 4.3 K(3.29)			K 4.2 K(3.30)			K 4.1 K(3.31)			K 4.0		
0x64	K 4.7 K(3.25)			K 4.6 K(3.26)			K 4.5 K(3.27)			K 4.4 K(3.28)		
0x68	K 4.11 K(3.21)			K 4.10 K(3.22)			K 4.9 K(3.23)			K 4.8 K(3.24)		
0x6C	K 4.15 K(3.17)			K 4.14 K(3.18)			K 4.13 K(3.19)			K 4.12 K(3.20)		
0x70	K 4.19 K(3.13)			K 4.18 K(3.14)			K 4.17 K(3.15)			K 4.16 K(3.16)		
0x74	K 4.23 K(3.9)			K 4.22 K(3.10)			K 4.21 K(3.11)			K 4.20 K(3.12)		
0x78	K 4.27 K(3.5)			K 4.26 K(3.6)			K 4.25 K(3.7)			K 4.24 K(3.8)		
0x7C	K 4.31 K(3.1)			K 4.30 K(3.2)			K 4.29 K(3.3)			K 4.28 K(3.4)		
0x80	K 0.0			K 1.0			K 2.0			K 3.0		
0x84	Reserved		Shift 1 & 6	Offset K1 & K6			Reserved	Div factor		Shift 0 & 7	Offset K0 & K7	
	Reserved		Shift 3 & 4	Offset K3 & K4			Reserved		Shift 2 & 5	Offset K2 & K5		

Address: *DisplayBaseAddress* + 0x100 .. 0x188

Type: RO/Link List update

Buffer: Immediate

**K X.Y:**

X: 0 - 7 is the coefficient's order in the 8 tap polyphase filter

Y: 0 - 31 is the interphase order

Two's complement form is used for K X.Y

**Div Factor:** Filter output division

000: Output of the filter is divided by 256

001: Output of the filter is divided by 512

010: Output of the filter is divided by 1024

011: Output of the filter is divided by 2048

100: Output of the filter is divided by 4096

others: reserved

**Offset KX & KY:**

DC value of the coefficient KX.N and KY.N, where N is 0 to 31 interphase value

**Shift X & Y:**

0: Coefficients KX.N and KY.N are directly read from register before adding DC value

1: Coefficients KX.N and KY.N are multiplied by 2 (shifted) before adding DC value

Internally the coefficients are extended to 10 bits as follows:

If '**Shift X & Y**' bit in registers corresponding to the coefficient is 1, the coefficient from register is multiplied by 2 (shifted), otherwise it is just extended to 10 bit two's complement value.

Then an offset is added to that value (two's complement value from registers 0x84 and 0x88 - '**Offset X & Y**').

At the end, in order to have the sum of coefficients equal to 1 for each subposition (0 dB attenuation at 0 Hz frequency), different values of scaling are programmable by the '**Div factor**'.

To summarize, the value of coefficients internally used by the filter is given by:

$$K = \frac{K_{reg} \cdot 2^{Shift} + Offset}{2^{(8 + Divfactor)}}$$

The coefficients are loaded only if DISP\_CTRL[30] = 1 (HFilter update enable).

## DISP\_LVF\_COEF

## DISP luma vertical filter coefficients

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x8C	K 4.3 (K 0.29)								K 4.2 (K 0.30)								K 4.1 (K 0.31)								K 4.0							
0x90	K 4.7 (K 0.25)								K 4.6 (K 0.26)								K 4.5 (K 0.27)								K 4.4 (K 0.28)							
0x94	K 4.11 (K 0.21)								K 4.10 (K 0.22)								K 4.9 (K 0.23)								K 4.8 (K 0.24)							
0x98	K 4.15 (K 0.17)								K 4.14 (K 0.18)								K 4.13 (K 0.19)								K 4.12 (K 0.20)							
0x9C	K 4.19 (K 0.13)								K 4.18 (K 0.14)								K 4.17 (K 0.15)								K 4.16 (K 0.16)							
0xA0	K 4.23 (K 0.9)								K 4.22 (K 0.10)								K 4.21 (K 0.11)								K 4.20 (K 0.12)							
0xA4	K 4.27 (K 0.5)								K 4.26 (K 0.6)								K 4.25 (K 0.7)								K 4.24 (K 0.8)							
0xA8	K 4.31 (K 0.1)								K 4.30 (K 0.2)								K 4.29 (K 0.3)								K 4.28 (K 0.4)							
0xAC	K 3.3 K(1.29)								K 3.2 K(1.30)								K 3.1 K(1.31)								K 3.0							
0xB0	K 3.7 K(1.25)								K 3.6 K(1.26)								K 3.5 K(1.27)								K 3.4 K(1.28)							
0xB4	K 3.11 K(1.21)								K 3.10 K(1.22)								K 3.9 K(1.23)								K 3.8 K(1.24)							
0xB8	K 3.15 K(1.17)								K 3.14 K(1.18)								K 3.13 K(1.19)								K 3.12 K(1.20)							
0xBC	K 3.19 K(1.13)								K 3.18 K(1.14)								K 3.17 K(1.15)								K 3.16 K(1.16)							
0xC0	K 3.23 K(1.9)								K 3.22 K(1.10)								K 3.21 K(1.11)								K 3.20 K(1.12)							
0xC4	K 3.27 K(1.5)								K 3.26 K(1.6)								K 3.25 K(1.7)								K 3.24 K(1.8)							
0xC8	K 3.31 K(1.1)								K 3.30 K(1.2)								K 3.29 K(1.3)								K 3.28 K(1.4)							
0xCC	K 2.3 K(2.29)								K 2.2 K(2.30)								K 2.1 K(2.31)								K 2.0							
0xD0	K 2.7 K(2.25)								K 2.6 K(2.26)								K 2.5 K(2.27)								K 2.4 K(2.28)							
0xD4	K 2.11 K(2.21)								K 2.10 K(2.22)								K 2.9 K(2.23)								K 2.8 K(2.24)							
0xD8	K 2.15 K(2.17)								K 2.14 K(2.18)								K 2.13 K(2.19)								K 2.12 K(2.20)							
0xDC	Reserved				Div factor		Shift 2		K 0.0								K 1.0								K 2.16							
0xE0	Offset K2										Shift 1 & 3		Offset K1 & K3										Shift 0 & 4		Offset K0 & K4							

Address: *DisplayBaseAddress* + 0x18C .. 0x1E0

Type: RO/Link List update

Buffer: Immediate

Reset: 0

Description: The DISP vertical luma filter registers 0x8C - 0xD8 are 32-bit registers containing four coefficients each (actually four subpositions of a coefficient). Each register coefficient value is given in two's complement format. Two other registers give supplementary information about filter coefficients.

**K X.Y**

X: 0 - 4 is the coefficient's order in the 5 tap polyphase filter

Y: 0 - 31 is the interphase order

Two's complement form is used for K X.Y

**Div Factor:** Filter output division

00: Output of the filter is divided by 64

01: Output of the filter is divided by 128

10: Output of the filter is divided by 256

11: Output of the filter is divided by 512

**Offset KX & KY**

DC value of the coefficient KX.N and KY.N, where N is 0 to 31 interphase value

**Shift X & Y**

0: Coefficients KX.N and KY.N are directly read from register before adding DC value

1: Coefficients KX.N and KY.N are multiplied by 2 (shifted) before adding DC value

Internally the coefficients are extended to 10 bits as follows:

If '**Shift X and Y**' bit in registers corresponding to the coefficient is 1, the coefficient from register is multiplied by 2 (shifted), otherwise it is just extended to 10 bit two's complement value.

Then an offset is added to that value (two's complement value from registers 0xDC and 0xE0 - '**Offset X and Y**').

At the end, in order to have the sum of coefficients equal to 4 for each subposition (12 dB gain at 0 Hz frequency), different values of scaling are programmable by '**Div Factor**'. The gain of 4 is due to 8-bit input to 10-bit output conversion.

To summarize, the value of coefficients internally used by the filter is given by:

$$K = \frac{K_{reg} \cdot 2^{Shift} + Offset}{2^{(6 + Divfactor)}}$$

The coefficients are loaded only if DISP\_CTRL[29] = 1 (VFilter update enable).

## DISP\_CHF\_COEF

## DISP chroma horizontal filter coefficients

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

0x00	K 7.3 (K 0.29)			K 7.2 (K 0.30)			K 7.1 (K 0.31)			K 7.0			
0x04	K 7.7 (K 0.25)			K 7.6 (K 0.26)			K 7.5 (K 0.27)			K 7.4 (K 0.28)			
0x08	K 7.11 (K 0.21)			K 7.10 (K 0.22)			K 7.9 (K 0.23)			K 7.8 (K 0.24)			
0x0C	K 7.15 (K 0.17)			K 7.14 (K 0.18)			K 7.13 (K 0.19)			K 7.12 (K 0.20)			
0x10	K 7.19 (K 0.13)			K 7.18 (K 0.14)			K 7.17 (K 0.15)			K 7.16 (K 0.16)			
0x14	K 7.23 (K 0.9)			K 7.22 (K 0.10)			K 7.21 (K 0.11)			K 7.20 (K 0.12)			
0x18	K 7.27 (K 0.5)			K 7.26 (K 0.6)			K 7.25 (K 0.7)			K 7.24 (K 0.8)			
0x1C	K 7.31 (K 0.1)			K 7.30 (K 0.2)			K 7.29 (K 0.3)			K 7.28 (K 0.4)			
0x20	K 6.3 K(1.29)			K 6.2 K(1.30)			K 6.1 K(1.31)			K 6.0			
0x24	K 6.7 K(1.25)			K 6.6 K(1.26)			K 6.5 K(1.27)			K 6.4 K(1.28)			
0x28	K 6.11 K(1.21)			K 6.10 K(1.22)			K 6.9 K(1.23)			K 6.8 K(1.24)			
0x2C	K 6.15 K(1.17)			K 6.14 K(1.18)			K 6.13 K(1.19)			K 6.12 K(1.20)			
0x30	K 6.19 K(1.13)			K 6.18 K(1.14)			K 6.17 K(1.15)			K 6.16 K(1.16)			
0x34	K 6.23 K(1.9)			K 6.22 K(1.10)			K 6.21 K(1.11)			K 6.20 K(1.12)			
0x38	K 6.27 K(1.5)			K 6.26 K(1.6)			K 6.25 K(1.7)			K 6.24 K(1.8)			
0x3C	K 6.31 K(1.1)			K 6.30 K(1.2)			K 6.29 K(1.3)			K 6.28 K(1.4)			
0x40	K 5.3 K(2.29)			K 5.2 K(2.30)			K 5.1 K(2.31)			K 5.0			
0x44	K 5.7 K(2.25)			K 5.6 K(2.26)			K 5.5 K(2.27)			K 5.4 K(2.28)			
0x48	K 5.11 K(2.21)			K 5.10 K(2.22)			K 5.9 K(2.23)			K 5.8 K(2.24)			
0x4C	K 5.15 K(2.17)			K 5.14 K(2.18)			K 5.13 K(2.19)			K 5.12 K(2.20)			
0x50	K 5.19 K(2.13)			K 5.18 K(2.14)			K 5.17 K(2.15)			K 5.16 K(2.16)			
0x54	K 5.23 K(2.9)			K 5.22 K(2.10)			K 5.21 K(2.11)			K 5.20 K(2.12)			
0x58	K 5.27 K(2.5)			K 5.26 K(2.6)			K 5.25 K(2.7)			K 5.24 K(2.8)			
0x5C	K 5.31 K(2.1)			K 5.30 K(2.2)			K 5.29 K(2.3)			K 5.28 K(2.4)			
0x60	K 4.3 K(3.29)			K 4.2 K(3.30)			K 4.1 K(3.31)			K 4.0			
0x64	K 4.7 K(3.25)			K 4.6 K(3.26)			K 4.5 K(3.27)			K 4.4 K(3.28)			
0x68	K 4.11 K(3.21)			K 4.10 K(3.22)			K 4.9 K(3.23)			K 4.8 K(3.24)			
0x6C	K 4.15 K(3.17)			K 4.14 K(3.18)			K 4.13 K(3.19)			K 4.12 K(3.20)			
0x70	K 4.19 K(3.13)			K 4.18 K(3.14)			K 4.17 K(3.15)			K 4.16 K(3.16)			
0x74	K 4.23 K(3.9)			K 4.22 K(3.10)			K 4.21 K(3.11)			K 4.20 K(3.12)			
0x78	K 4.27 K(3.5)			K 4.26 K(3.6)			K 4.25 K(3.7)			K 4.24 K(3.8)			
0x7C	K 4.31 K(3.1)			K 4.30 K(3.2)			K 4.29 K(3.3)			K 4.28 K(3.4)			
0x80	K 0.0			K 1.0			K 2.0			K 3.0			
0x84	Reserved		Shift 6	Offset K1 & K6				Reserved	Div factor		Shift 0 & 7	Offset K0 & K7	
0x88	Reserved		Shift 3 & 4	Offset K3 and K4				Reserved		Shift 2 & 5	Offset K2 & K5		

Address: *DisplayBaseAddress* + 0x200 .. 0x288

**K X.Y**

X: 0 - 7 is the coefficient's order in the 8 tap polyphase filter

Y: 0 - 31 is the interphase order

Two's complement form is used for K X.Y

**Div Factor:** Filter output division

000: Output of the filter is divided by 256

001: Output of the filter is divided by 512

010: Output of the filter is divided by 1024

011: Output of the filter is divided by 2048

100: Output of the filter is divided by 4096

others: reserved

**Offset KX and KY**

DC value of the coefficient KX.N and KY.N, where N is 0 to 31 interphase value

**Shift X and Y**

0: Coefficients KX.N and KY.N are directly read from register before adding DC value

1: Coefficients KX.N and KY.N are multiplied by 2 (shifted) before adding DC value

Internally the coefficients are extended to 10 bits in the next way:

If '**Shift X and Y**' bit in registers 0x74 and 0x78 corresponding to the coefficient is 1, the coefficient from register is multiplied by 2 (shifted), otherwise it is just extended to 10 bit two's complement value.

Then an offset is added to that value (two's complement value from registers 0x84 and 0x88 - '**Offset X and Y**').

At the end, in order to have the sum of coefficients equal to 1 for each subposition (0 dB attenuation at 0 Hz frequency), different values of scaling are programmable by '**Div Factor**'.

To summarize, the value of coefficients internally used by the filter is given by:

$$K = \frac{K_{reg} \cdot 2^{Shift} + Offset}{2^{(8 + Divfactor)}}$$

The coefficients are loaded only if DISP\_CTRL[30] = 1 (HFilter update enable).



## DISP\_CVF\_COEF

## DISP chroma vertical filter coefficients

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x8C	K 4.3 (K 0.29)								K 4.2 (K 0.30)								K 4.1 (K 0.31)								K 4.0							
0x90	K 4.7 (K 0.25)								K 4.6 (K 0.26)								K 4.5 (K 0.27)								K 4.4 (K 0.28)							
0x94	K 4.11 (K 0.21)								K 4.10 (K 0.22)								K 4.9 (K 0.23)								K 4.8 (K 0.24)							
0x98	K 4.15 (K 0.17)								K 4.14 (K 0.18)								K 4.13 (K 0.19)								K 4.12 (K 0.20)							
0x9C	K 4.19 (K 0.13)								K 4.18 (K 0.14)								K 4.17 (K 0.15)								K 4.16 (K 0.16)							
0xA0	K 4.23 (K 0.9)								K 4.22 (K 0.10)								K 4.21 (K 0.11)								K 4.20 (K 0.12)							
0xA4	K 4.27 (K 0.5)								K 4.26 (K 0.6)								K 4.25 (K 0.7)								K 4.24 (K 0.8)							
0xA8	K 4.31 (K 0.1)								K 4.30 (K 0.2)								K 4.29 (K 0.3)								K 4.28 (K 0.4)							
0xAC	K 3.3 K(1.29)								K 3.2 K(1.30)								K 3.1 K(1.31)								K 3.0							
0xB0	K 3.7 K(1.25)								K 3.6 K(1.26)								K 3.5 K(1.27)								K 3.4 K(1.28)							
0xB4	K 3.11 K(1.21)								K 3.10 K(1.22)								K 3.9 K(1.23)								K 3.8 K(1.24)							
0xB8	K 3.15 K(1.17)								K 3.14 K(1.18)								K 3.13 K(1.19)								K 3.12 K(1.20)							
0xBC	K 3.19 K(1.13)								K 3.18 K(1.14)								K 3.17 K(1.15)								K 3.16 K(1.16)							
0xC0	K 3.23 K(1.9)								K 3.22 K(1.10)								K 3.21 K(1.11)								K 3.20 K(1.12)							
0xC4	K 3.27 K(1.5)								K 3.26 K(1.6)								K 3.25 K(1.7)								K 3.24 K(1.8)							
0xC8	K 3.31 K(1.1)								K 3.30 K(1.2)								K 3.29 K(1.3)								K 3.28 K(1.4)							
0xCC	K 2.3 K(2.29)								K 2.2 K(2.30)								K 2.1 K(2.31)								K 2.0							
0xD0	K 2.7 K(2.25)								K 2.6 K(2.26)								K 2.5 K(2.27)								K 2.4 K(2.28)							
0xD4	K 2.11 K(2.21)								K 2.10 K(2.22)								K 2.9 K(2.23)								K 2.8 K(2.24)							
0xD8	K 2.15 K(2.17)								K 2.14 K(2.18)								K 2.13 K(2.19)								K 2.12 K(2.20)							
0xDC	Reserved				Div factor		Shift 2		K 0.0								K 1.0								K 2.16							
0xE0	Offset K2										Shift 1 & 3		Offset K1 & K3										Shift 0 & 4		Offset K0 & K4							

Address: *DisplayBaseAddress* + 0x28C .. 0x2E0

Type: RO/Link List update

Buffer: Immediate

Reset: 0

Description: The DISP vertical chroma filter registers 0x8C - 0xD8 are 32-bit registers containing four coefficients each (actually four subpositions of a coefficient). Each register coefficient value is given in two's complement format. Two other registers give supplementary information about filter coefficients.

#### K X.Y

X: 0 - 4 is the coefficient's order in the 5 tap polyphase filter

Y: 0 - 31 is the interphase order

two's complement form is used for K X.Y

**Div factor:** Filter output division

00: Output of the filter is divided by 64

01: Output of the filter is divided by 128

10: Output of the filter is divided by 256

11: Output of the filter is divided by 512

#### Offset KX & KY

DC value of the coefficient KX.N and KY.N, where N is 0 to 31 interphase value

#### Shift X & Y

0: Coefficients KX.N and KY.N are directly read from register before adding DC value

1: Coefficients KX.N and KY.N are multiplied by 2 (shifted) before adding DC value

Internally the coefficients are extended to 10 bits as follows:

If '**Shift X and Y**' bit in registers corresponding to the coefficient is 1, the coefficient from register is multiplied by 2 (shifted), otherwise it is just extended to 10 bit two's complement value.

Then an offset is added to that value (two's complement value from registers 0xDC and 0xE0 - '**Offset X and Y**').

At the end, in order to have the sum of coefficients equal to 4 for each subposition (12 dB gain at 0 Hz frequency), different values of scaling are programmable by '**Div Factor**'. The gain of 4 is due to 8-bit input to 10 bit output conversion.

To summarize, the value of coefficients internally used by the filter is given by:

$$K = \frac{K_{reg} \cdot 2^{Shift} + Offset}{2^{(6 + Divfactor)}}$$

The coefficients are loaded only if DISP\_CTRL[29] = 1 (VFilter update enable).

## 50.2 LMU registers

## LMU\_CTRL

## LMU control

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MC_Y	DTI_Y	Y_EN	CLK_Y	FMC_Y	FML_Y						Reserved	MC_C	DTI_C	C_EN	CK_C	FMC_C	FML_C													

Address: *LMUBaseAddress* + 0x00

Type: R/W

Buffer: Double buffered: Vsync

Reset: 0

Description: This register controls the LMU activity.

[31] **Reserved**

[30] **MC\_Y:**

0: the LMU checks for motion by comparing individual luma pixels from the current top/bottom field with the co-located luma pixel in the previous top/bottom field.

1: the motion is calculated using a pair of horizontally adjacent luma pixels.

The comparison is always between same type fields: top field compared to previous top field, and bottom field compared to previous bottom field. When using a pixel pair, individual pixels are compared, but the highest motion result is applied for both pixels.

[29] **DTI\_Y:**

0: temporal interpolation operates as described in the LMU specification.

1: this bit disables temporal interpolation for the luma pixels.

[28] **Y\_EN:**

When set to 1, the luma LMU is enabled; when set to 0, the luma LMU is disabled

[27:26] **CLK\_Y:**

Depending on the size of the picture being processed by the LMU, it may not be necessary for the LMU to operate at full speed. These bits allow the LMU operating speed to be controlled so that the STBus bandwidth consumed by the luma LMU can be evenly distributed over picture time.

00: divide by 1, turbo mode (no clock throttling)

01: divide by 2

10: divide by 3

11: divide by 4

[25:24] **FMC\_Y:** The film-mode control bits select the method for upsampling to create the interpolated luma lines.

00: motion-adaptive de-interlacing, used for video sources

01: missing lines obtained from following field, used for film source

10: missing lines obtained from preceding field, used for film source

11: blank missing lines, simulates interlaced display

[23:16] **FML\_Y:** The LMU's film mode detection algorithm compares the current field to the same field in the previous frame. The comparison starts with line 2 of the luma field and continues until the line number programmed in FML\_Y. A separate comparison is made for luma and chroma portions of the picture.

[15] **Reserved**

[14] **MC\_C:**

0: the LMU checks for motion by comparing individual chroma pixels from the current top/bottom field with the co-located chroma pixel in the previous top/bottom field.

1: the motion is calculated using a pair of horizontally adjacent chroma pixels.

The comparison is always between same type fields: top field compared to previous top field, and bottom field compared to previous bottom field. When using a pixel pair, individual pixels are compared, but the highest motion result is applied for both pixels. Applying the same motion value to both pixels in a pixel-pair avoids the situation where motion is detected in the Cb pixel but not in the Cr pixel.

- # Confidential

**LMU luma buffer start pointer**

Address:	<i>LMUBaseAddress</i> + 0x04
Type:	R/W
Buffer:	Double buffered: Vsync
Reset:	0
Description:	Pointer to the start of the memory LMU luma buffer, in 256-byte units.

**LMU chroma buffer start pointer**

Address:	<i>LMUBaseAddress</i> + 0x08
Type:	R/W
Buffer:	Double buffered: Vsync
Reset:	0
Description:	Pointer to the start of the memory LMU chroma buffer, in 256-byte units.

**LMU\_BPPL****LMU number of block-pairs per line**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																BLK_PAIRS_PER_LINE										
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--------------------	--	--	--	--	--	--	--	--	--	--

Address: *LMUBaseAddress* + 0x0C

Type: R/W

Buffer: Double buffered: Vsync

Reset: 0

Description: This unsigned register should be programmed with a value that indicates the number of 16-pixel blocks per picture width in the display window.

**LMU\_CFG****LMU configuration**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																							TOG	TNB	422
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----	-----	-----

Address: *LMUBaseAddress* + 0x10

Type: R/W

Buffer: Double buffered: Vsync

Reset: 0

Description: This register configures the chroma input format and the type of the input field.

[31:3] **Reserved**

[2] **TOG**: toggle type of input field. When set the input field selection on the display automatically toggles on each Vsync. This only allows the field to be set once at a sequence start up. When reset, the selected field can be frozen or set every field by the application.

[1] **TNB**: top not bottom field. Indicates which field of the picture is read from memory by the display on the next Vsync. When set, the top field is selected, when reset, the bottom field is selected. If TOG is set, the selected field automatically toggles on each Vsync.

[0] **422**:

0: 420 input field

1: 422 input field

**LMU\_VINL****LMU number of input lines**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																VINL										
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	--

Address: *LMUBaseAddress* + 0x14

Type: R/W

Buffer: Double buffered: Vsync

Reset: 0

Description: This register specifies the number of input lines to the display formatters in terms of progressive luma count.

For 4:2:0 inputs, it must be programmed with a value that is a multiple of four for interlace inputs.

For 4:2:2 inputs, it must be programmed with a value that is a multiple of two for interlace inputs.

This register is unsigned.

**LMU\_MRS****LMU minimum space between STBus requests**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																MRS															

Address: *LMUBaseAddress* + 0x18

Type: R/W

Buffer: Double buffered: Vsync

Reset: 0

Description: This register specifies the number of CLK\_SYS clock cycles between two consecutive STBus requests (luma/chroma write to memory and luma/chroma read from memory).

**LMU\_CHK****LMU maximum chunk size**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								WRC	WRY	RDC	RDY				

Address: *LMUBaseAddress* + 0x1C

Type: R/W

Buffer: Double buffered: Vsync

Reset: 0

Description: This register configures the number of STBus transactions linked together using the LCK bit. It indicates the number of packets in a message.

[31:8] **Reserved**

[7:6] **RDY**: Luma STBus read interface.

00: A message is 8 packets

01: A message is 1 packet

10: A message is 2 packets

11: A message is 4 packets

[5:4] **RDC**: Chroma STBus read interface.

00: A message is 8 packets

01: A message is 1 packet

10: A message is 2 packets

11: A message is 4 packets

[3:2] **WRY**: Luma STBus write interface.

00: A message is 8 packets

01: A message is 1 packet

10: A message is 2 packets

11: A message is 4 packets

[1:0] **WRC**: Chroma STBus write interface.

00: A message is 8 packets

01: A message is 1 packet

10: A message is 2 packets

11: A message is 4 packets

**LMU\_STA****LMU status**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CDN	YDN	CFF	YFF	YWE	CWE	YRE	CRE

Address: *LMUBaseAddress* + 0x20

Type: RO

Buffer: Immediate

Reset: Undefined

Description: This register shows the activity of the LMU and potential problems.

[31:8] **Reserved**

[7] **CDN**: Chroma LMU done for the current field

[6] **YDN**: Luma LMU done for the current field

[5] **CFF**: Chroma WR fifo to STBus is full

[4] **YFF**: Luma WR fifo to STBus is full

[3] **YWE**: STBus packet transfer error for Luma WR interface

[2] **CWE**: STBus packet transfer error for Chroma WR interface

[1] **YRE**: STBus packet transfer error for Luma RD interface

[0] **CRE**: STBus packet transfer error for Chroma RD interface

**LMU\_ITM****LMU interrupt mask**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CDN	YDN	CFF	YFF	YWE	CWE	YRE	CRE

Address: *LMUBaseAddress* + 0x24

Type: R/W

Buffer: Immediate

Reset: 0 (all interrupts disabled)

Description: Any bit set in this register enables the corresponding interrupt in the LMU\_IRQ line. An interrupt is generated whenever a bit in register LMU\_STA changes from 0 to 1 and the corresponding mask bit is set.

**LMU\_INT\_STA****LMU interrupt status**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CDN	YDN	CFF	YFF	YWE	CWE	YRE	CRE

Address: *LMUBaseAddress* + 0x28

Type: RO

Buffer: Immediate

Reset: Undefined

Description: When a bit in register LMU\_STA changes from 0 to 1, the corresponding bit in register LMU\_INTS is set, independent of the state of LMU\_ITM. If any set LMU\_INTS bit is unmasked, the line LMU\_IRQ is asserted. Reading LMU\_INTS clears all bits in this register. When LMU\_INTS is 0, the LMU\_IRQ line returns deasserted.

**LMU\_AFD****LMU accumulated field difference**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DIFF_Y								DIFF_C							

Address: *LMUBaseAddress* + 0x2C

Type: RO

Buffer: Double buffered: Vsync

Reset: 0

Description: This unsigned status register returns two 8-bit values that indicate the accumulated difference between the current field and the same field of the previous frame; bits 0 to 7 indicate the difference between chroma fields, bits 8 to 15 indicate the difference between luma fields. The accumulated difference may be used by software to detect if a received picture is from a film source.

The difference is not actually accumulated over the entire field. Instead, 16 horizontally adjacent pixels in the current field are compared to the equivalent 16 pixels of the previous frame. The magnitude of each pixel difference is accumulated in a 12-bit register. At the end of the 16-pixel sample, the upper 8 bits of the accumulated difference are stored in an 8-bit maximum difference register.

This process is repeated over the picture. At the end of each 16-pixel calculation, the current accumulated difference is compared to the stored difference, and the greater of the two is stored. This is carried out for both luma and chroma pixels, and the results are accumulated separately. At the end of the field (that is, during vertical reset), the maximum difference register for luma is assigned to field DIFF\_Y of LMU\_AFD, while the maximum difference register for chroma is assigned to field DIFF\_C.

If LMU\_CTRL.DTI\_Y is set to 1, field LMU\_AFD.DIFF\_Y is invalid. Similarly, if bit LMU\_CTRL.DTI\_C of is set to 1, then field LMU\_AFD.DIFF\_C is invalid.

Confidential



## 51 Compositor

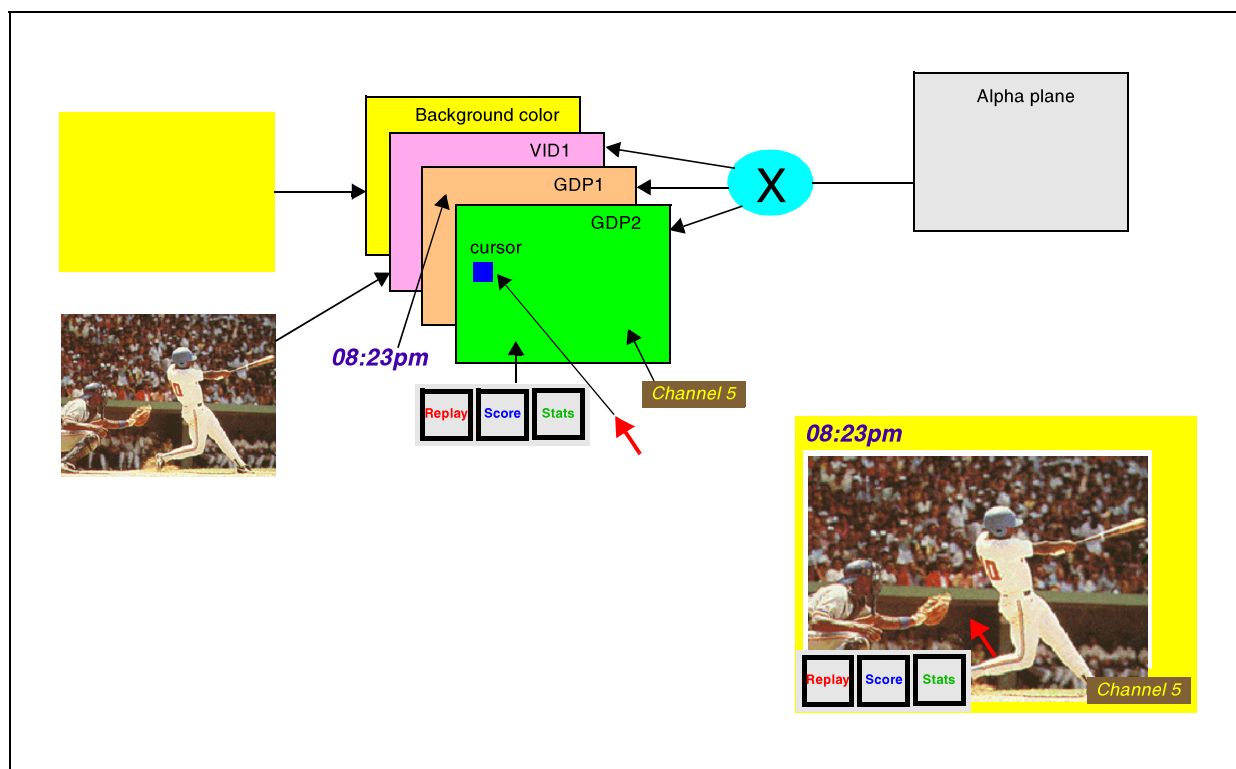
### 51.1 Overview

The compositor comprises two real-time, multiplane digital mixers. The main mixer (MIX1) composes up to five layers: a background color, a video plane, two graphics planes, and a cursor plane. The auxiliary mixer (MIX2) composes the video 2 plane with the graphics 2 plane.

The video planes are supplied from the main and auxiliary display processors. Pixel data for the 2D-graphics planes and cursor plane are read directly from memory.

After real time processing by the display plane pipelines, pixel data are mixed in mixer 1 or mixer 2. The output of mixer 1 supports up to full HD resolutions and is intended as the main TV display (Figure 167). The output of mixer 2 (Figure 168) supports up to full SD resolutions and is intended as an auxiliary display for applications including connection to a VCR. The mixer outputs are fed to the STx7100's output stage.

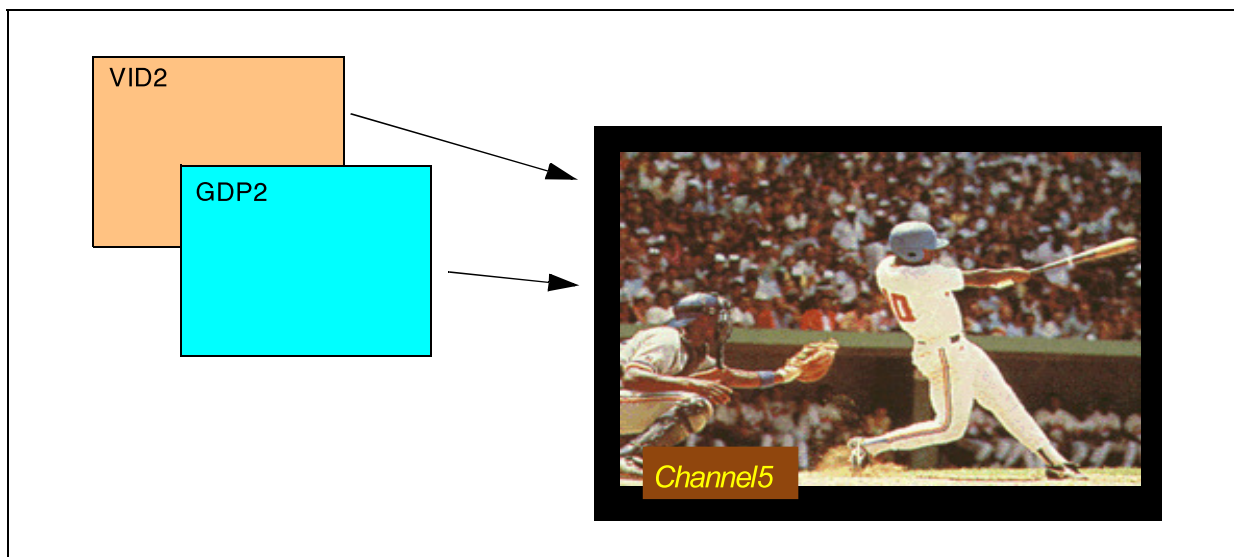
Figure 167: Mixer 1 plane



The compositor also comprises additional components that can be used to enhance the display presentation of video and graphics. These include an alpha plane attachment and a cross-bar

router. Their functions are described below. A capture pipeline is also provided for capturing main or auxiliary video streams or mixer 1 or 2 output streams and storing them in memory.

**Figure 168: Mixer 2 planes**



## 51.2 Compositor layout

Figure 169 shows a block diagram of the compositor. It presents the dataflow and memory access of all the compositor modules.

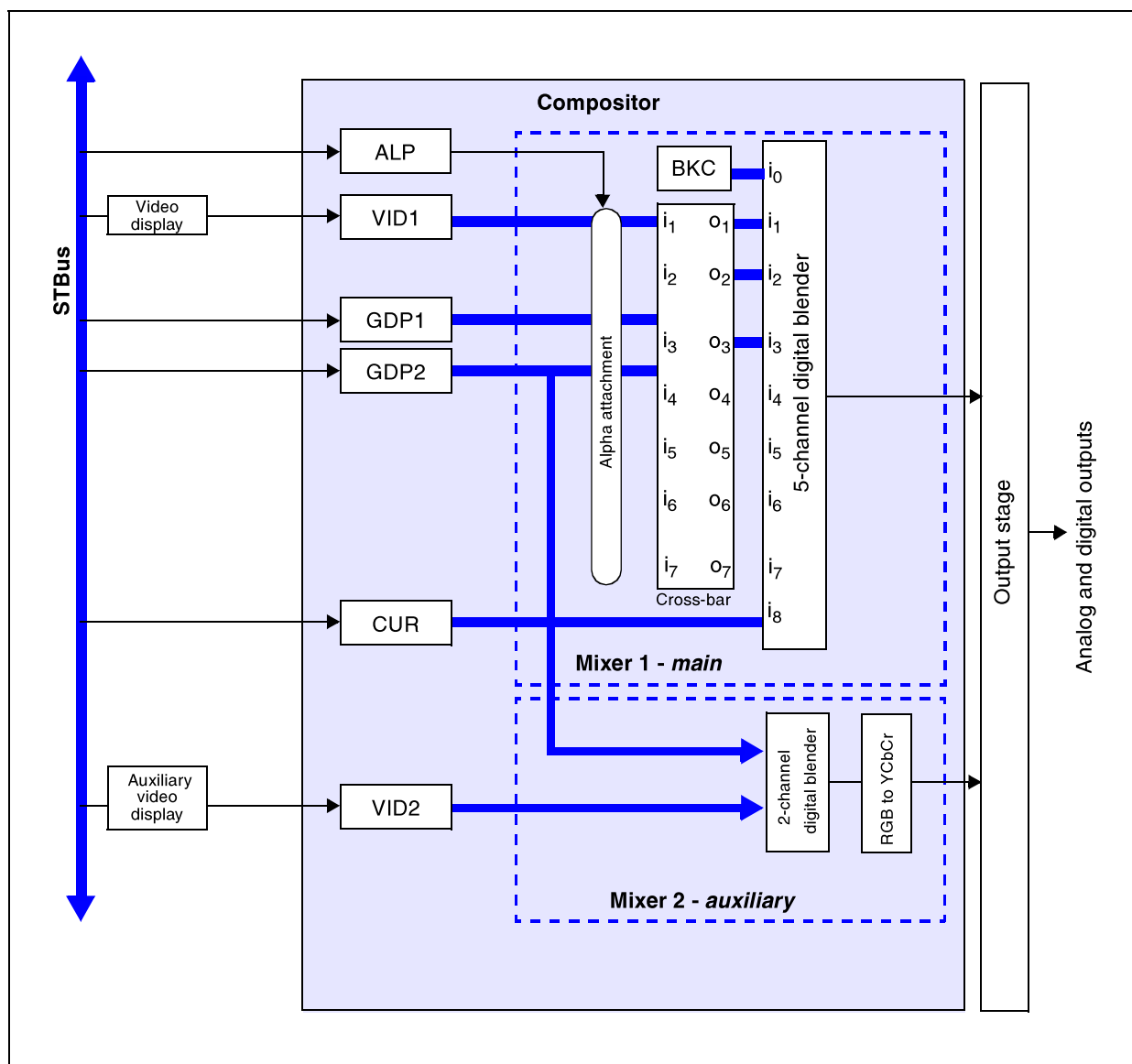
The graphics and cursor pipelines read pixel data and related control information directly from memory. The video input pipelines accept data from the main and auxiliary video display pipelines. Video and graphics data captured for the compositor data flow by the capture pipeline is written back to memory with a resolution up to 32 bits/pixel. The real-time processing performed by each pipeline is controlled by register programming.

Digital mixer 1 successively blends video layers VID1, both graphics layers (GDP1 and GDP2), the cursor layer and a background color. A cross-bar router enables the hierarchy of the GDP1 and VID1 layers to be programmed. The resulting order is background color, GDP2, (VID1 and GDP1 in programmed order) and cursor from background to foreground. Each layer can be independently enabled or disabled. The blending operates in the RGB color domain, so each layer supplies an RGB signal (3x12 bits), with transparency information that provides the weighting coefficients for the mixing operation at a given depth.

Digital mixer 2 successively blends one video layer (VID2) with one graphics layer (GDP2). For mixer 2, the priority is fixed with GDP2 in front of video.

All sub-blocks are controlled by hardware registers. All these registers can be read but not necessarily written. The graphics planes are link-list based and have their register set written through the memory (register download is controlled directly by the hardware after initialization). All other registers can be written. The registers are listed in [Chapter 52 on page 554](#). Each plane block supports a specific set of bitmap formats. All bitmap formats are described in [Section 47.8: Local memory storage of supported graphics formats on page 458](#). Each plane starts reading data from memory when it is enabled in mixer 1 or mixer 2.

Figure 169: Graphics and video block diagram



Confidential

## 51.3 Digital mixer 1 (MIX1) - main display output

The main display output mixer mixes all the display planes and generates the main output (it cannot be used for the auxiliary output). The output can be high- or standard-definition, and is usually used for TV.

Mixer 1 is controlled by registers prefixed with `GAM_MIX1_`.

Each graphics and video layer can be enabled or disabled for display in register `GAM_MIX1_CTRL`.

Mixer 1 is composed of three independent sub-blocks: the blender, the cross-bar router, and the alpha plane attachment unit.

### 51.3.1 Four-channel blender

The digital mixer successively blends the four layers, from background to foreground. Each layer can be independently enabled or disabled. Blending operates in the RGB color domain, so each layer, except for the cursor, supplies an RGB signal (3 x 12 bits) with transparency information that provides the weighting coefficients for the mixing operation at a given depth.

The background color register `GAM_MIX1_BKC` is a 24-bit RGB register included in the mixer, and is functionally equivalent to a plane filled with solid color. This plane is always opaque with no associated alpha value. The limits of this plane are specified according to a programmable rectangular window. Outside this window, the background color is forced to the blanking color (black  $R = G = B = 0$ ).

The computation unit outputs a 4:4:4 digital RGB signal. A global rectangular window for defining the active video area is provided (`GAM_MIX1_AVO` and `GAM_MIX1_AVS`). Outside this window, the mixer outputs a default blanking color (black  $R = G = B = 0$ ). A window indicator signal is provided, synchronously with the RGB data bus, for external use (such as a video blanking signal).

Mixer 1 takes into account bits `IGNOREONMIX1` and `FORCEONMIX1` (`GAM_GDPn_PPT`), provided by the GDP pipelines, on any enabled GDP layer. If bit `IGNOREONMIX1` is set, the current viewport is not displayed. If bit `FORCEONMIX1` is set, then the GDP viewport color information can be displayed outside the active video area window, instead of the blanking color.

*Note: The window indicator flag is not affected.*

From an application point of view, this last feature is useful when a VBI waveform has been synthesized as a graphics object, and uses a GDP pipeline to be inserted in the analog output on a VBI line.

Two configurable signals can indicate whether the current output contains a certain amount of graphics information, or if it is composed of pure video content. In some systems, external processing operations can be applied selectively, according to the pixel video or graphic content.

### 51.3.2 Cross-bar router

The cross bar router is used to re-order the plane hierarchy, so that the user can program the order that the video and graphics planes appear in the display, from background to foreground. The planes that can be re-ordered include the video plane and the two graphics planes (`VID1`, `GDP1` and `GDP2`). The background color and cursor layers are not affected. The depth of each layer in the hierarchy is programmed in register `GAM_MIX1_CRB`.

### 51.3.3 Alpha plane attachment unit

The alpha plane processor can combine the alpha channel provided by the alpha plane pipeline with the alpha component of either GDP1, GDP2 or VID1.

Any of these pipelines supplies RGB color information, together with two alpha components. One gives the blending weight for this current RGB pixel ( $\alpha_{\text{RGB}}(\text{IN})$ ), whereas the other gives the weight for the intermediate pixel resulting from the blend operations of all the background layers ( $\alpha_{\text{BKG}}(\text{IN})$ ). Let  $\alpha_{\text{alpha\_plane}}$  be the value provided by the alpha plane pipeline.

The alpha processor reinjects the alpha plane value on both components, using:

$$\alpha_{\text{RGB}}(\text{OUT}) = \alpha_{\text{RGB}}(\text{IN}) \times \alpha_{\text{alpha\_plane}}$$

$$\alpha_{\text{BKG}}(\text{OUT}) = 1.0 - ((1.0 - \alpha_{\text{BKG}}(\text{IN})) \times \alpha_{\text{alpha\_plane}})$$

This is correct for both the VID and GDP pipelines handling either premultiplied or non-premultiplied colors.

## 51.4 Digital mixer 2 (MIX2) - auxiliary display output

Mixer 2 is a much simpler engine, for the auxiliary display only, with just two inputs, generic display pipeline GDP2 and video VID2. It is controlled by the registers prefixed with `GAM_MIX2_`.

Each graphics and video layer can be enabled or disabled for display in `GAM_MIX2_CTRL`.

Blending operates in the RGB color domain, from the RGB signals (3 x 12 bits) supplied by the layers, together with the associated transparency information.

There is no background color register, but a default black background color ( $R = G = B = 0$ ).

The computation unit outputs a 4:4:4 digital RGB signal. A global rectangular window for defining the active video area is provided (`GAM_MIX2_AVO` and `GAM_MIX2_AVS`). Outside this window, the mixer outputs a default blanking color (black  $R = G = B = 0$ ). A window indicator signal is provided, synchronously with the RGB/YCbCr data bus, for external use (such as a video blanking signal, for instance).

If GDP2 is enabled, mixer 2 takes into account the property register `GAM_GDPn_PPT`, bits `IGNOREONMIX2` and `FORCEONMIX2`, provided by the GDP2 pipeline. If `IGNOREONMIX2` is set, the current viewport is not displayed. If `FORCEONMIX2` is set, then the GDP viewport color information can be displayed outside the active video area window, instead of the blanking color.

*Note: The window indicator flag is not affected.*

From an application point of view, this last feature is useful when a VBI waveform has been synthesized as a graphics object, and uses the GDP2 pipeline to be inserted in the analog output, on a VBI line.

Two configurable signals indicate whether the current output contains graphics information, or if it is composed of pure video content. In some systems, external processing operations can be applied selectively, according to the pixel video or graphics content.

## 51.5 Generic display pipelines (GDP1 and GDP2)

The generic display pipelines have the following features:

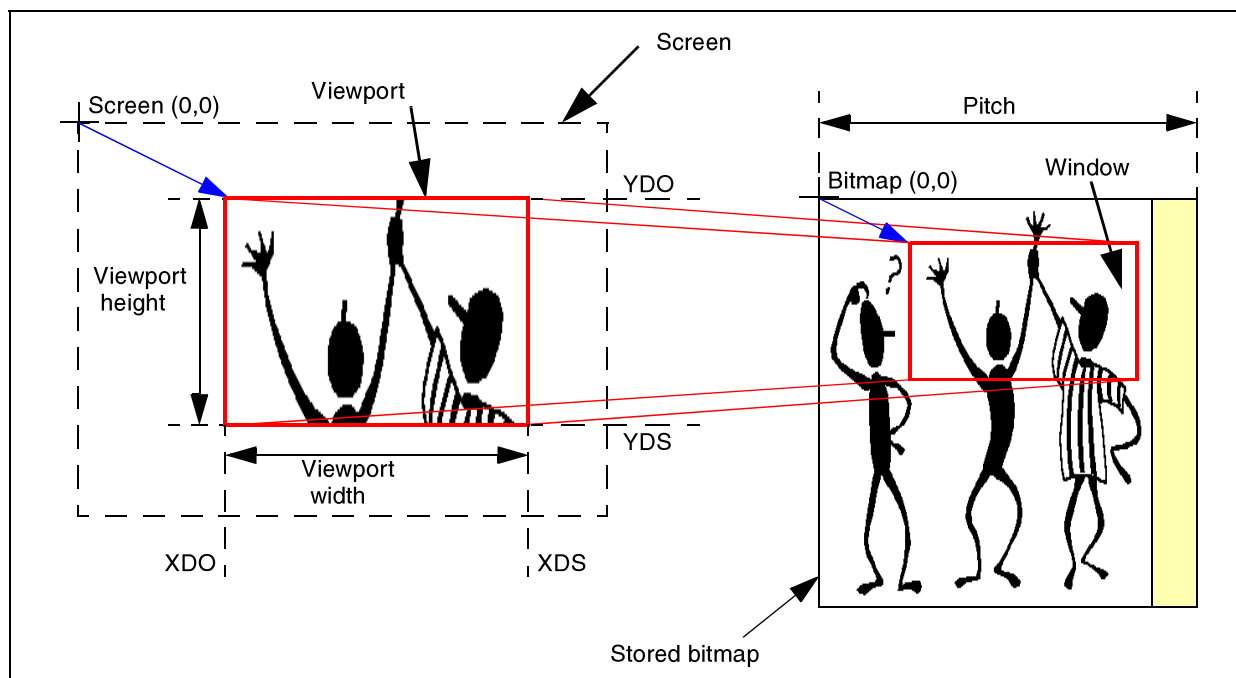
- Link-list-based display engine, for multiple viewport capabilities.
- ARGBargb formats including ARGB1555, ARGB4444, RGB565, RGB888, ARGB8565, ARGB8888.
- YCbCr4:2:2R, YCbCr888 and AYCbCr8888 format support.
- Premultiplied or non-premultiplied RGB component support.
- Little endian and big endian bitmap support.
- Color space conversion matrix, (YCbCr 601/709, chroma signed/unsigned to RGB).
- Gain and offset adjustment.
- Per-pixel alpha channel combined with per-viewport global alpha.
- 5-tap horizontal sample rate converter, for horizontal upsampling. This can be used to adapt the pixel aspect ratio. The resolution is 1/8th pixel (polyphase filter with eight subpositions).
- Color keying capability.

The output format of GDP is RGB-12-12-12 that goes to the digital mixer along with the 8-bit alpha and 8-bit (1-alpha) values. See full description of the supported graphics in [Section 47.8: Local memory storage of supported graphics formats on page 458](#).

### 51.5.1 General description of GDPs

A GDP can handle a multiple-viewport display, using a display instruction list (link list) stored in the external memory. (A viewport is a physical rectangular area on the screen. It is defined by XDO / XDS / YDO / YDS with reference to the top-left corner of the screen.)

**Figure 170: Example: a resized window attached to a viewport**



One or several viewports can be attached to the GDP layer, depending on its capabilities. When several viewports are defined within a layer, only one viewport can be handled by each video scan line. In a given layer, each screen location where no viewport is defined is automatically transparent (the alpha channel to the mixer is forced to 0).

A bitmap stored in external memory must be attached to the viewport. All or part of this bitmap can be visible in the viewport; this is the bitmap window. The window is defined by the bitmap

color format, bitmap pitch, linear start address, width and height. The linear start address is the address of the bitmap pixel that is to be displayed in the top-left corner of the viewport, that is the address of the window (0,0) pixel.

Most of the time, the window and the viewport have the same size. Their sizes are different each time a resizing factor is applied to the window, so that the rescaled bitmap matches with the viewport size.

When programmed sizes are not consistent, the display is locally undefined.

The screen is described by a display link-list that must be built in the external memory. For each viewport, a node is defined that contains the viewport configuration, bitmap window settings, display options (such as global transparency, filtering mode, gain/offset adjustment). The node also contains a memory pointer to the node describing the next viewport to display.

The display link-list is generally circular. For an interlaced display, the link-list is field-based. A node must be provided for the top and the bottom fields (they can be different, particularly the start address).

**Figure 171: Typical display link-list configuration for a progressive display**

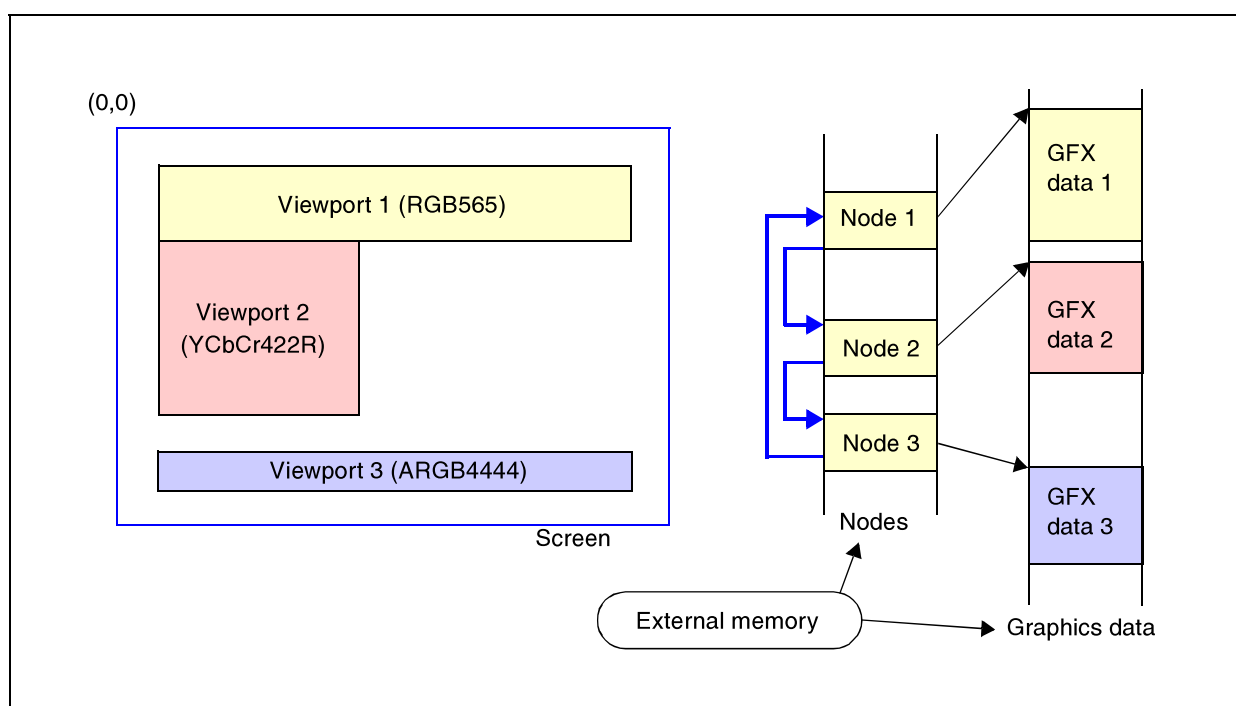
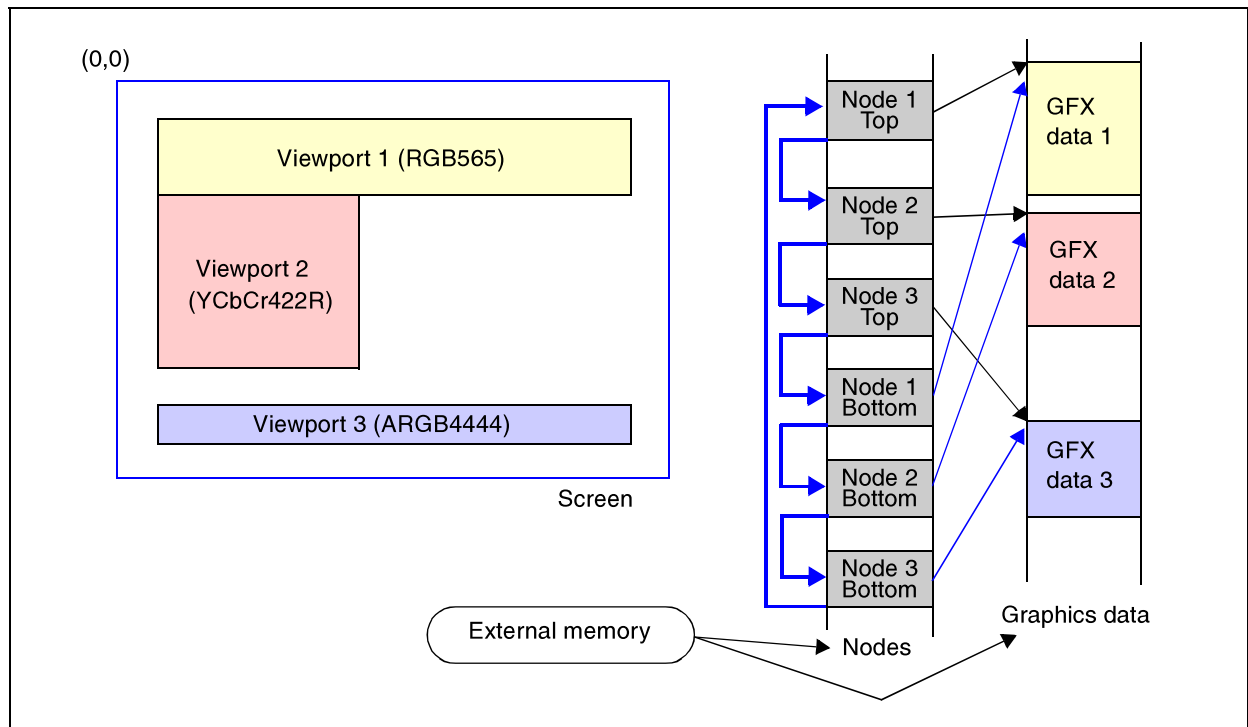




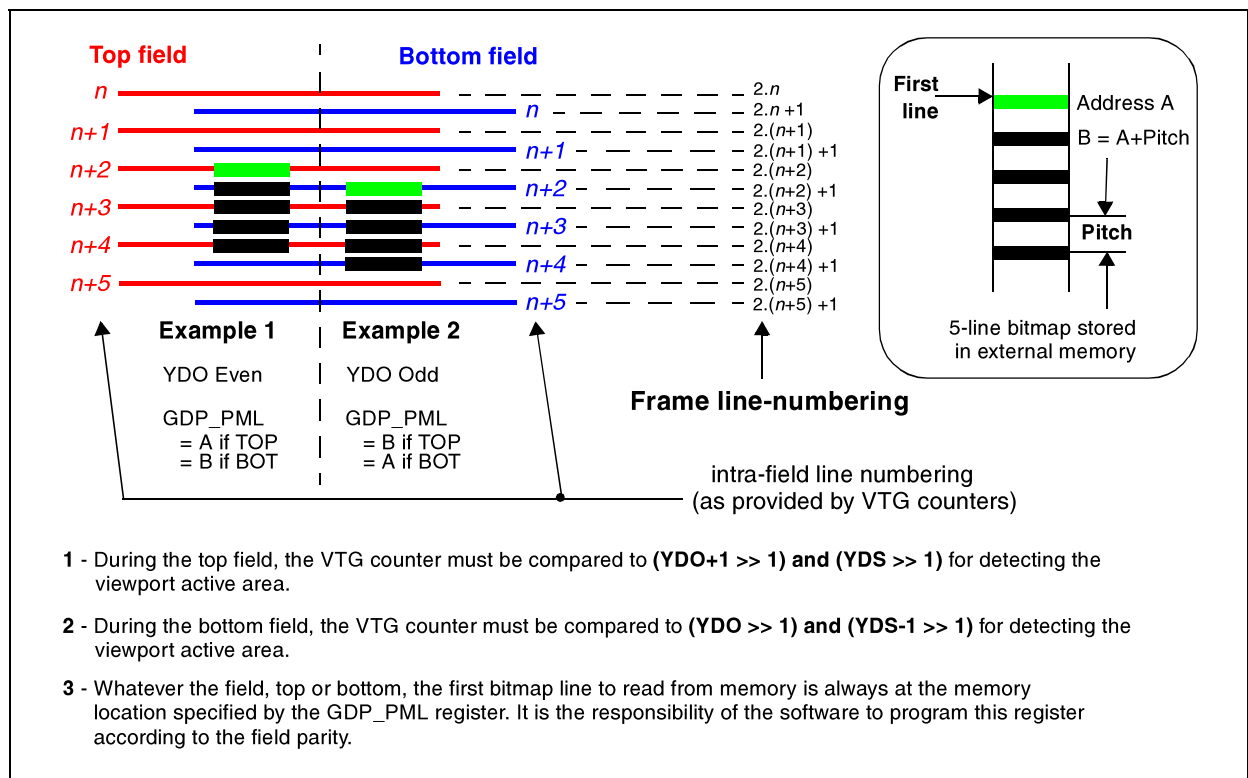
Figure 172: Typical display link-list configuration for an interlaced display



YDO and YDS are specified with respect to a frame line-numbering, even in an interlaced display.

The next figure specifies how the hardware must consider the register values, depending on the top or bottom field, so that each viewport can be vertically positioned with a one-line accuracy.

Figure 173: Line numbering convention in an interlaced display



For the progressive display configuration, this consideration is of course trivial.



### How to start the display

To start the display on a GDP, a link-list must be programmed in memory. Register `GAM_GDPn_NVN` must be set by the CPU with the address of the first node to be displayed within this link-list. Then the corresponding GDP enable bit in register `GAM_MIXn_CTRL` must be set. This write operation is synchronized on the next Vsync event. On that event, the GDP pipeline fetches the first node from memory, and starts to retrieve pixel data according to the display parameters specified in the node.

To obtain clean node switching, particularly in the case of two vertically adjacent viewports (the horizontal blanking interval can be quite short), the hardware uses an internal dual register bank in toggle mode for the nodes. The next node loading process is always anticipated (for node  $N$ ), and occurs at the beginning of the first line displayed for node  $(N - 1)$ . This mechanism is transparent to the programmer.

To stop the display, the `GAM_MIXn_CTRL` enable bit must be set back to 0. This is taken into account synchronously with the field (frame) rate.

### How to modify the display parameters

Hardware and software may conflict when accessing the nodes if the software wants to modify a display parameter. It is recommended to toggle between two link-lists stored in memory each time one or more display parameters have to be modified.

For the final node of either of these link-lists, which corresponds to the lowest viewport, bit `WAIT_NEXT_VSYNC` in register `GAM_GDPn_CTRL` must be set to 1. This prevents the node anticipation process from occurring: loading is delayed until the next Vsync event.

When the hardware is working from the current displayed link-list, the software is free to make any modification in the other link-list, such as viewport parameters, viewport insertion/deletion. Once the link-list has been updated, the software simply updates the `GAM_GDPn_NVN` register field in the last node of the current link-list. This is a single memory write access, and thus cannot conflict with a hardware access (no partially updated parameters). As this node has been programmed not to anticipate the next node loading, the hardware waits until the next Vsync, and then correctly switches to the new link-list.

If memory update for register `GAM_GDPn_NVN` is synchronized in the Vsync software handler, the link-list switch occurs with a one field delay (or one frame if progressive).

For an interlaced display, this scheme can be simplified by using a single link-list and updating the top parameters while the bottom field is displayed and vice-versa.

### Bandwidth considerations

In terms of the bandwidth requirement (BR), the general formula to estimate the GDP weight on a given system is the following:

$$\text{BR (in Mbyte/s)} = (\text{pixel frequency in MHz}) \times (\text{number of bytes per pixel}) \times (\text{horizontal resampling factor})$$

Examples:

- 601 rate / RGB565 / x1:  $\text{BR} = 13.5 \times 2 \times 1 = 27 \text{ Mbyte/s}$
- PAL SQ rate / YCbCr422R / x1:  $\text{BR} = 14.75 \times 2 \times 1 = 29.5 \text{ Mbyte/s}$
- 601 rate / ARGB8888 / zoom out x2:  $\text{BR} = 13.5 \times 4 \times 2 = 108 \text{ Mbyte/s}$

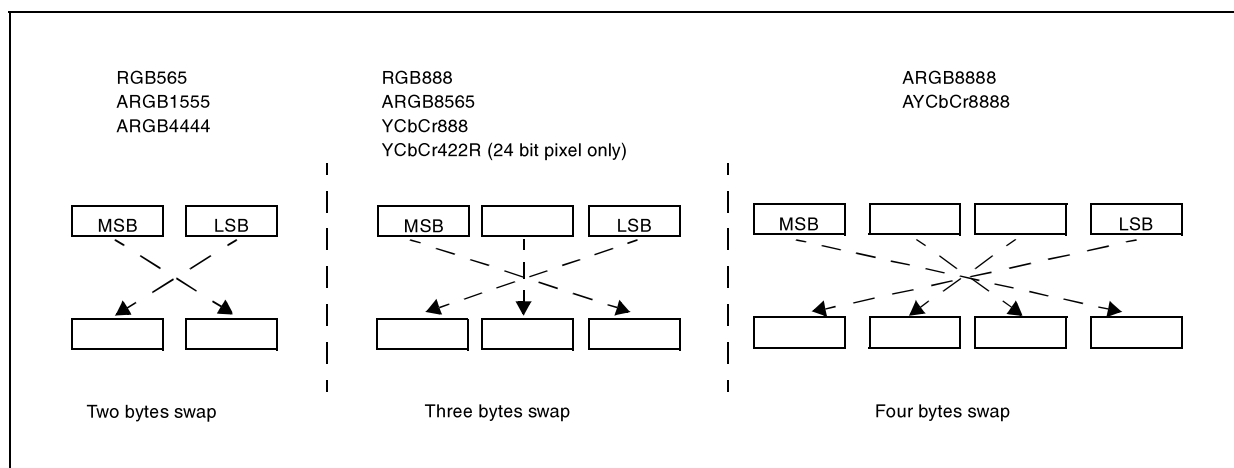
## 51.5.2 Description of unitary functions

### Input formatter

This sub-block converts the 128-bit STBus data bus, into an internal pixel bus. The clock rate for the pixel bus is PIXCLK.

If the bitmap is big endian, the endianness conversion occurs on the original pixels as read from memory, before their mapping on the internal 32-bit pixel bus. Figure 174 shows the byte swap performed on big endian pixels, according to the different color formats. For each viewport, the endianness of the bitmap may be defined using `GAM_GDPn_CTRL.BIGNOTLITTLE`.

**Figure 174: Endianness conversion for big endian pixels**



Another bit that is not included in the viewport node (`GAM_GDPn_PKZ.BIGNOTLITTLE`) indicates if the nodes, filters coefficients, are stored in big or little endian. This bit should reflect the endianness of the host CPU. It performs a four-byte swap on the 32-bit word fetched when loading the nodes and the filter coefficients.

**Table 163: 32-bit pixel bus format, at the formatter output**

Input color format	Alpha		Color data					
	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
<b>RGB565</b>	128		R5/3MSB or 000		G6/2MSB or 00		B5/3MSB or 000	
<b>RGB888</b>	128		R8		G8		B8	
<b>ARGB1555</b>	GLOBAL_ALPHA_0 if A1=0 GLOBAL_ALPHA_1 if A1=1		R5/3MSB or 000		G5/3MSB or 000		B5/3MSB or 000	
<b>ARGB8565</b>	A8 <sup>a</sup>		R5/3MSB or 000		G6/2MSB or 00		B5/3MSB or 000	
<b>ARGB8888</b>	A8 <sup>a</sup>		R8		G8		B8	
<b>ARGB4444</b>	0/A4/100 (but keep 0 and 128)		R4/4MSB or 0000		G4/4MSB or 0000		B4/4MSB or 0000	
<b>YCbCr888</b> <b>YCbCr422R</b>	128		Cr8		Y8		Cb8	
<b>AYCbCr888</b> <b>8</b>	A8 <sup>a</sup>		Cr8		Y8		Cb8	

a. The GDP supports either 0 to 128 or 0 to 255 8-bit alpha. When a 255 range is used, the input alpha value is converted to a 0 to 128 component using the following formula:

$$A_{0..128} = (A_{0..255} + 1) \times 2^{-1}.$$

When there is no alpha channel in the input color format, a 128 alpha value is forced, except for the ARGB1555 mode. In this case, the global alpha registers (0 and 1) are used in the input formatter block. When bit 15 of the incoming pixel (A1) is 0, the alpha output is forced to the value of GLOBAL\_ALPHA\_0. When bit 15 is 1, the alpha output is forced to the value of GLOBAL\_ALPHA\_1.

Two possibilities are provided to extend the component depth (4/5/6 bits to 8 bits): either the LSBs are filled with 0, or the MSBs are repeated, so that the full 8-bit color range can be used (see LSB\_STUFFING\_MODE in register [GAM\\_GDPn\\_CTRL](#)). For 4:2:2 to 4:4:4 conversion, the interpolation scheme is similar to the one used in the blitter (except for the chroma extended mode that is not supported). This is described in [Chapter 47: 2D graphics processor \(blitter\) on page 428](#).

### Color space converter

If the input format is YCbCr888, AYCbCr8888 or YCbCr4:2:2R, a YCbCr to RGB color space converter is required. It uses either the 601 matrix or the 709 matrix. It is bypassed for an RGB input. In this version of the compositor, the video RGB color space is used to fit with NTSC/PAL test patterns. The output is 10:10:10 signed RGB.

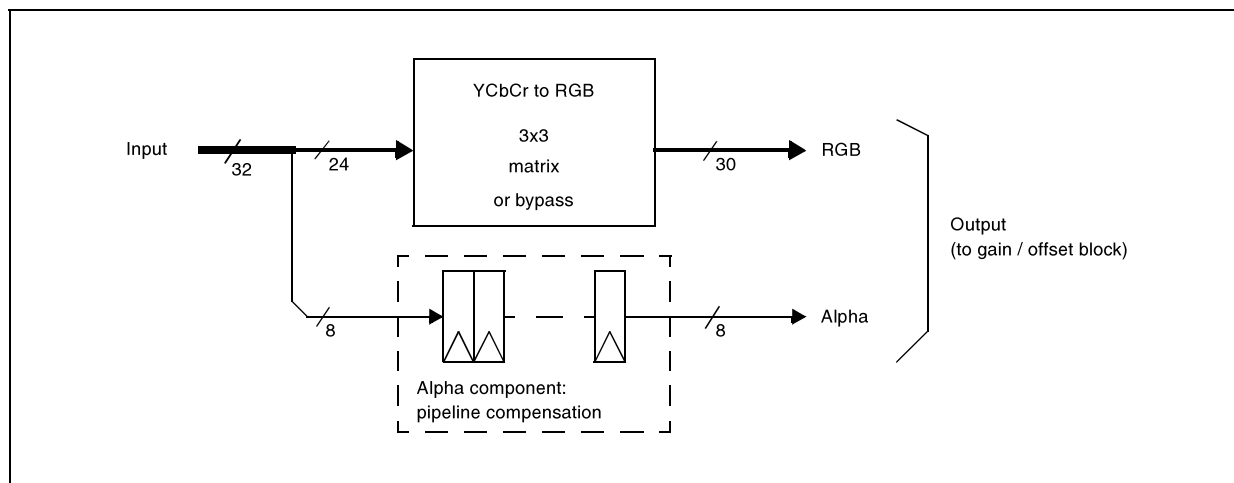
**Table 164A: 601 colorimetry / floating-point matrix / digital range**

YCbCr to video RGB integer matrix									
R	=	1	x Y	+	0	x (Cb - 128)	+	351/256	x (Cr - 128)
G	=	1	x Y	-	86/256	x (Cb - 128)	-	179/256	x (Cr - 128)
B	=	1	x Y	+	444/256	x (Cb - 128)	+	0	x (Cr - 128)

**Table 164B: 709 colorimetry / floating-point matrix / digital range**

YCbCr to video RGB integer matrix									
R	=	1	x Y	+	0	x (Cb - 128)	+	394/256	x (Cr - 128)
G	=	1	x Y	-	47/256	x (Cb - 128)	-	117/256	x (Cr - 128)
B	=	1	x Y	+	464/256	x (Cb - 128)	+	0	x (Cr - 128)

**Figure 175: Color space converter block diagram**



## 2D-resize, horizontal filter

The GDP contains a horizontal filter that can be used for upsampling of graphics displays. Downsizing and vertical resizing are not supported. The filter is based on an 5-tap sample rate converter with eight phases and is programmed using `GAM_GDPn_HSRC`. The SRC increment is programmed using a 2.8 format and the initial phase can be programmed to a resolution of eight subpixel positions.

The horizontal filter coefficients are stored in memory. The 40 filter coefficients are loaded from memory pointed to by `GAM_GDPn_HFP` and can be updated for each viewport if required. Resizing can be performed with the filter enabled for interpolation or disabled for pixel repetition. The horizontal filter can be used for upsampling and is provided to allow storage of HD resolution graphics with a lower resolution (for example, x1/4 compared with full resolution storage) and also for aspect ratio conversion of square pixel generated graphics for nonsquare pixel aspect ratio display or vice versa.

## Color key

If the current pixel color is part of the key range, then the associated alpha component of this pixel is forced to 0, making this pixel totally transparent when mixed with the background layers in the digital mixer. The color components are forced to black.

**Table 165: Color key block output if match**

Input format	Output if color key match
ARGBargb	A = R = G = B = 0
YCbCr422R / YCbCr888 / AYCbCr8888 Chroma offset binary (unsigned)	A = 0, Y = 16, Cb = Cr = 128
YCbCr422R / YCbCr888 / AYCbCr8888 Chroma two's complement (signed)	A = 0, Y = 16, Cb = Cr = 0

The color key feature is enabled for a viewport by setting `GAM_GDPn_CTRL.COLOR_KEY_EN` to 1. When set, the color key operates as follows:

`COLOR_KEY_MATCH = TRUE`

**IF**

`[(Rin < Rmin) or (Rin > Rmax)] AND GAM_GDPn[21:20] = 11`

`OR (Rmin <= Rin <= Rmax) AND GAM_GDPn[21:20] = 01`

`OR GAM_GDPn[21:20] = x0`

**AND**

`[(Gin < Gmin) or (Gin > Gmax)] AND GAM_GDPn[19:18] = 11`

`OR (Gmin <= Gin <= Gmax) AND GAM_GDPn[19:18] = 01`

`OR GAM_GDPn[19:18] = x0`

**AND**

`[(Bin < Bmin) OR (Bin > Bmax)] AND GAM_GDPn[17:16] = 11`

`OR (Bmin <= Bin <= Bmax) AND GAM_GDPn[17:16] = 01`

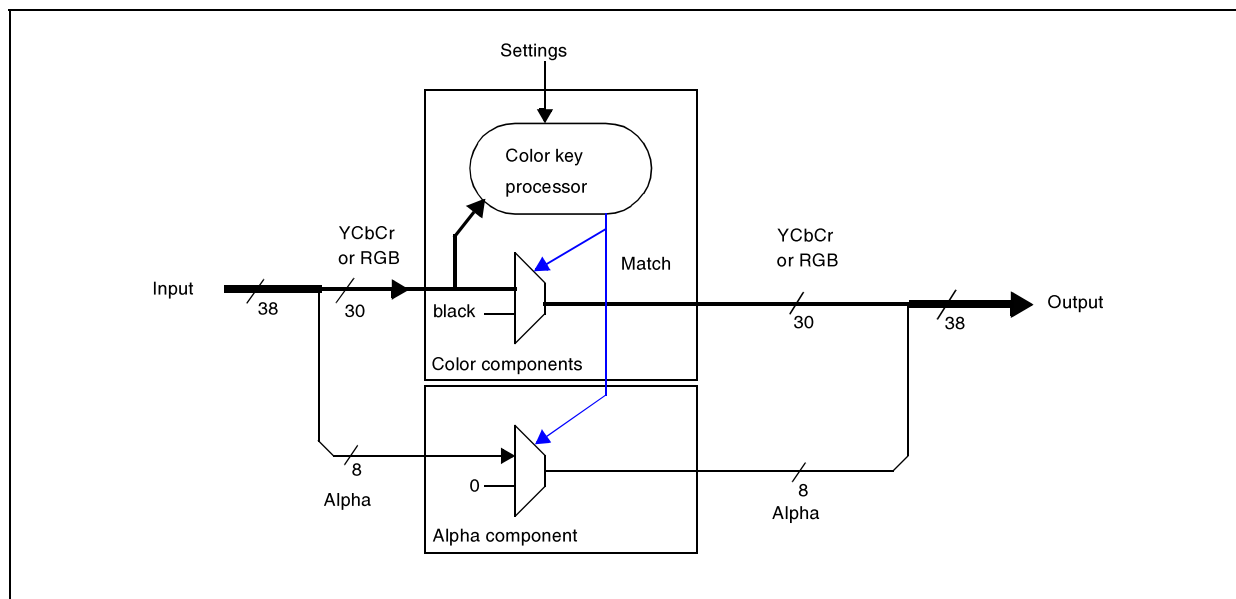
`OR GAM_GDPn[17:16] = x0`

When used on a YCbCr input the correspondence is the following: G/Y, Cb/B, Cr/R.

The values  $R_{MIN}$ ,  $G_{MIN}$ ,  $B_{MIN}$  are programmed in `GAM_GDPn_KEY1`.

The values  $R_{MAX}$ ,  $G_{MAX}$  and  $B_{MAX}$  are programmed in `GAM_GDPn_KEY2`.

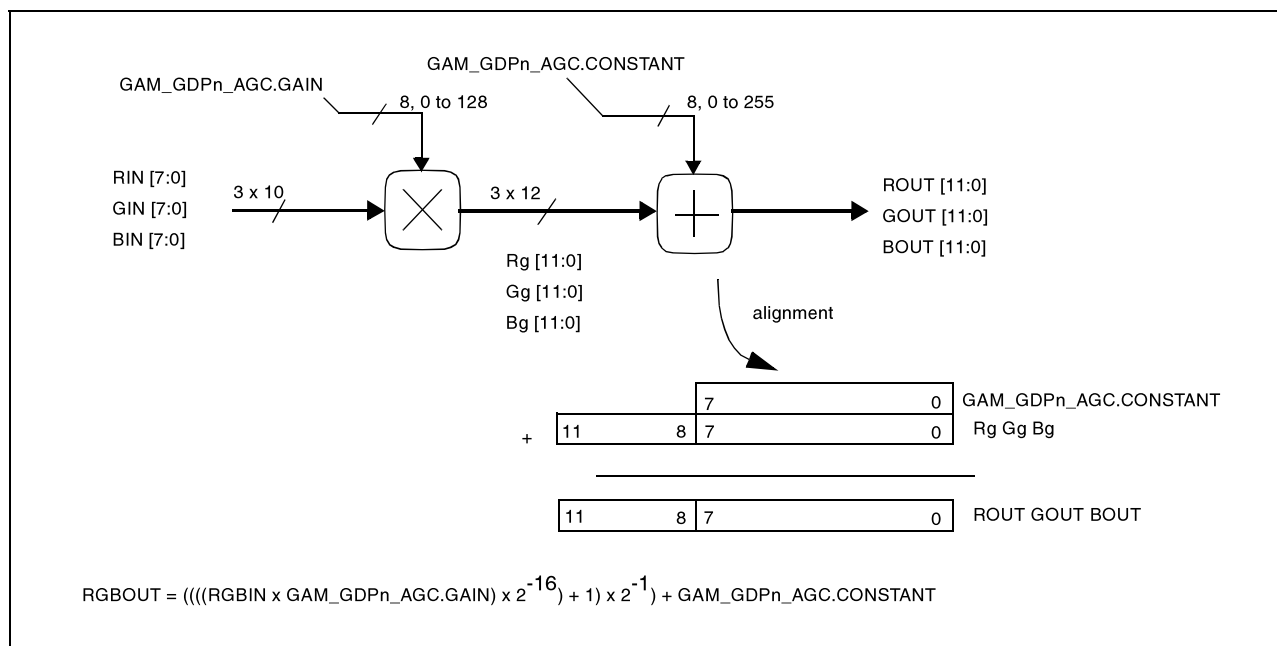
Figure 176: Color key block diagram



### Gain/offset adjustment

A dynamic range adjustment is provided. The three RGB components can be multiplied by a fixed coefficient, before they are sent to the digital mixer. This is useful to avoid highly saturated colors to be mixed with video. The black level can be adjusted as well. When used simultaneously, these features act like a contrast adjustment.

Figure 177: Gain/offset diagram



### Alpha out sub-block

This block provides the relative weights that are used by the digital mixer when mixing the GDP output on top of the background layers (BKG). The following C-like code shows what is output by the pipeline, depending on the input format (premultiplied or not, special case for ARGB1555), and on the ALPHA\_HBORDER\_EN / ALPHA\_VBORDER\_EN configuration bits in register [GAM\\_GDPn\\_CTRL](#).

Naming conventions:

- Alpha<sub>PIXEL</sub> is the alpha component attached to the current pixel at the output of the color space converter block.
- Alpha<sub>GDP</sub> is the weight provided to the mixer for the GDP color components
- Alpha<sub>BKG</sub> is the weight provided to the mixer for the background color components

```
IF (InputFormat == ARGB1555)
GlobalAlpha = 128; // GLOBALALPHA0/1 registers used in input formatter block
ELSE
    GlobalAlpha = GlobalAlpha0_register;
IF [(FirstPixel or LastPixel) AND ALPHA_HBORDER_EN]
OR ((FirstLine OR LastLine) AND ALPHA_VBORDER_EN)
GlobalAlpha = GlobalAlpha >> 1; // Alpha divided by 2 on viewport edges
IF (PREMULTIPLIED_FORMAT)
Alpha_GDP = GlobalAlpha // Alpha_PIXEL already applied on color components
ELSE
    Alpha_GDP = (((GlobalAlpha x Alpha_PIXEL) >> 6) + 1) >> 1;
    Alpha_BKG = 128 - { [(((GlobalAlpha x Alpha_PIXEL) >> 6) + 1) >> 1 ],
```

GLOBALALPHA0/1 registers refer to [GAM\\_GDPn\\_AGC.GLOBAL\\_ALPHA\\_0](#) and [GLOBAL\\_ALPHA\\_1](#).  
ALPHA\_VBORDER\_EN, ALPHA\_HBORDER\_EN, and PREMULTIPLIED\_FORMAT are in [GAM\\_GDPn\\_CTRL](#).

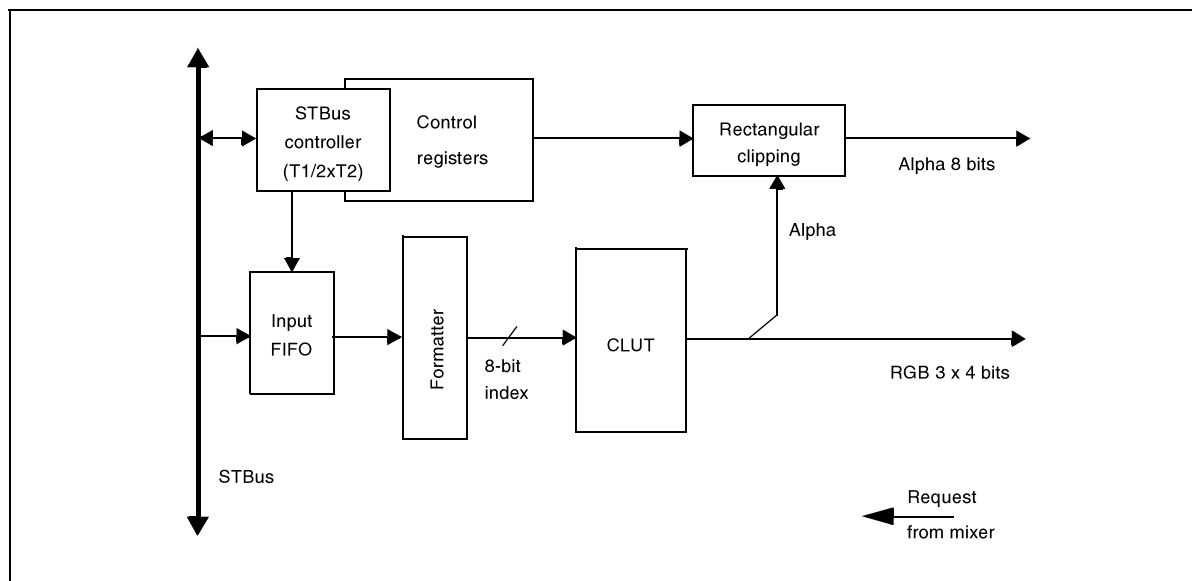
## 51.6 Cursor plane (CUR)

The cursor is defined as a 128x128 pixel area held in off-chip memory, in ACLUT8 format. Each cursor entry is a 16-bit ARGB4444 color + alpha value. The alpha factor of four-bits is for handling an anti-aliased cursor pattern on top of the composed output picture.

The cursor plane has the following features:

- ACLUT8 format, with ARGB4444 CLUT entries, so 256 colors can be simultaneously displayed for the cursor pattern, among 4096 colors associated with a 16-level translucency channel.
- Size is programmable up to 128x128.
- Hardware rectangular clipping window, out of which the cursor is never displayed (per-pixel clipping, so only part of the cursor can be out of this window, and consequently transparent).
- Current bitmap is specified using a pointer register to an external memory location, making cursor animation very easy.
- Programmable pitch, so that all cursor patterns can be stored in a single global bitmap.

Figure 178: CUR functional block diagram



### 51.6.1 General description of cursor plane

#### Cursor overview

The cursor pipeline can display a cursor pattern, stored in external memory in ACLUT8 format. An internal CLUT makes the color expansion from index to true-color values, using 16-bit ARGB4444 entries.

- The vertical size of the cursor pattern can range from 1 to 128 (register [GAM\\_CUR\\_SIZE](#)).
- The maximum horizontal size depends on the memory alignment of the cursor pixel data. Assume that [CUR\\_MEM\\_ADDR](#) ([GAM\\_CUR\\_PML](#)) is the memory address for pixel (0,0), with respect to a top-left origin. The maximum width is:

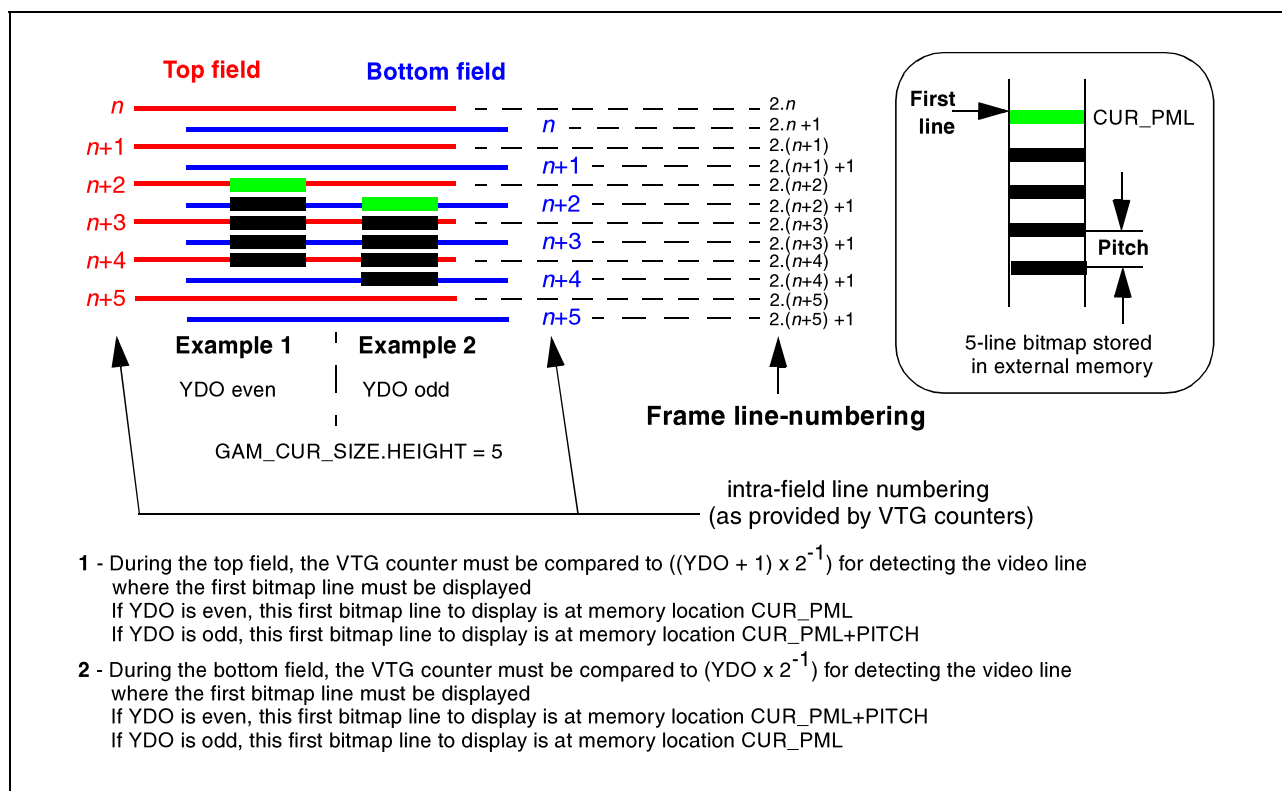
$$\text{MAX\_CURSOR\_WIDTH} = 128 - (\text{CUR\_MEM\_ADDR} \bmod 16)$$

When the cursor data are aligned on a 128-bit word memory, the maximum width is 128. The worst case is when  $(\text{CUR\_MEM\_ADDR} \bmod 16) = 15$  (example: 0xAAAF); the maximum width becomes 113.

- The cursor pipeline is controlled through a register file directly accessible by the CPU. As these registers are double-buffered with internal update on Vsync, any change is taken into account during the next field to be displayed.
- The cursor pattern can be changed simply by modifying register [GAM\\_CUR\\_PML](#) (GUI cursor change, cursor animation).
- The CLUT is loaded from memory during each vertical blanking interval, if this loading process is enabled (see [GAM\\_CUR\\_CTRL.REFRESH](#)). A dedicated register provides the memory address to retrieve the CLUT data ([GAM\\_CUR\\_CML](#)).
- The YDO value (in register [GAM\\_CUR\\_VPO](#)) is specified with respect to frame line-numbering, even in an interlaced display.

The next figure specifies how the hardware uses these register values, depending on the parity of the current field (top or bottom), so that each viewport can be vertically positioned with a one-line accuracy. Thus these registers need not be programmed according to the current displayed field. The pipeline automatically fetches the right data in memory.

Figure 179: Cursor line numbering assumption in an interlaced display



For progressive display configuration, this consideration is not important.

### How to start the cursor display

To start displaying the cursor, the corresponding enable bit in register [GAM\\_MIX1\\_CTRL](#) must be set. This write operation is synchronized on the next Vsync event. On that event, the cursor pipeline fetches the CLUT from memory, if the CLUT refresh process is enabled.

Then the cursor pipeline retrieves pixel data from memory, according to the display parameters specified in the node. A single access is made.

To stop the display, the [GAM\\_MIX1\\_CTRL](#) enable bit must be set back to 0. This is taken into account synchronously with the field (frame) rate.

### Bandwidth considerations

In terms of bandwidth requirements, the cursor pipeline weight on a given system is as follows:

- During the active video scanning area, one or two requests that occur at the beginning of the video line (total: 128 bytes or less, according to the cursor width). Normally, one burst is OK, except if there is a memory page crossing (the page size is assumed to be 512 bytes).
- During the vertical blanking interval, four or five consecutive requests to refresh the CLUT, that occur just after the Vsync event (five in the case of a memory page crossing).



## 51.6.2 Unitary functions

### Input formatter

There is no endianness issue concerning the bitmap, as the format is 8 bpp.

A register bit [GAM\\_CUR\\_PKZ.BIGNOTLITTLE](#) indicates if the palette is stored in big or little endian. It performs a two bytes swap on the 16-bit ARGB4444 entries.

**Table 166: Endianness conversion for big endian CLUT entries**

D[15:8]	D[7:0]	Big endian		Little endian	CPU address
AR	GB	AR0	Entry 0	GB0	0
ARGB4444		GB0		AR0	1
		AR1	Entry 1	GB1	2
		GB1		AR1	3
					4
					5
					6
					7

Two bytes swap

### Clipping window

A rectangular clipping window can be defined. Outside this window, the cursor pattern, even if defined, is forced to be transparent (alpha = 0).

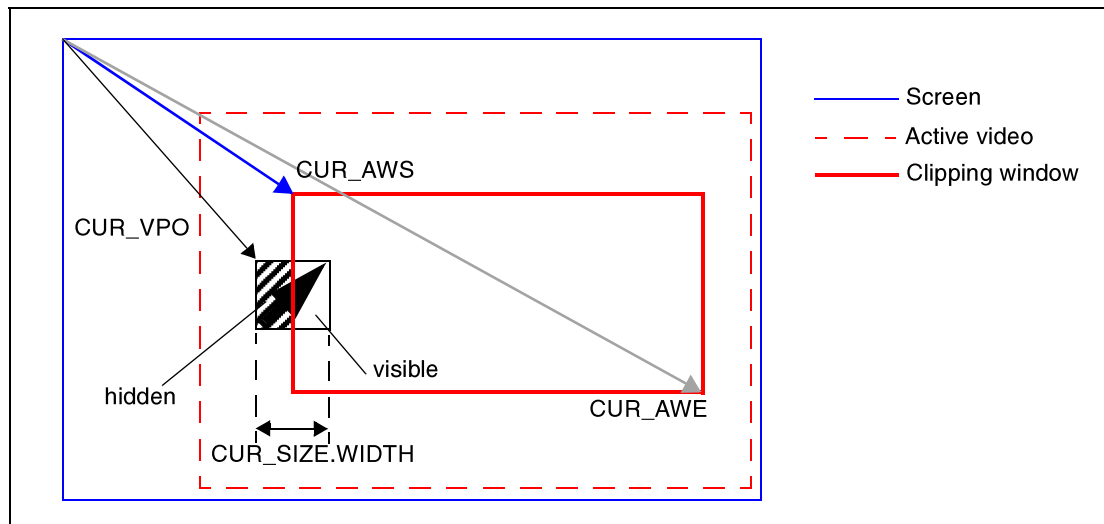
The clipping window is defined using two registers: [GAM\\_CUR\\_AWS](#) (active window start) and [GAM\\_CUR\\_AWE](#) (active window end). This is specified with respect to the frame numbering convention (such as a GDP or VID viewport).

[Figure 180](#) illustrates the use of the hardware cursor clipping window. Bits XDO and YDO are signed (register [GAM\\_CUR\\_VPO](#)), so that the top-left corner of the cursor pattern can be positioned anywhere in the screen, even in horizontal and vertical blanking.

There is no enable bit for the hardware window: the feature is disabled by programming the window to match the screen size.

**Note:** The same screen look can be obtained without using the hardware window. As the address generator supports a pitch parameter ([GAM\\_CUR\\_PMP](#)) that is different from the cursor width, it is possible to use the base address [GAM\\_CUR\\_CML.CLUT\\_MEM\\_ADDR](#) and the width ([GAM\\_CUR\\_SIZE.WIDTH](#)) parameters in conjunction.

**Figure 180: Cursor clipping window**



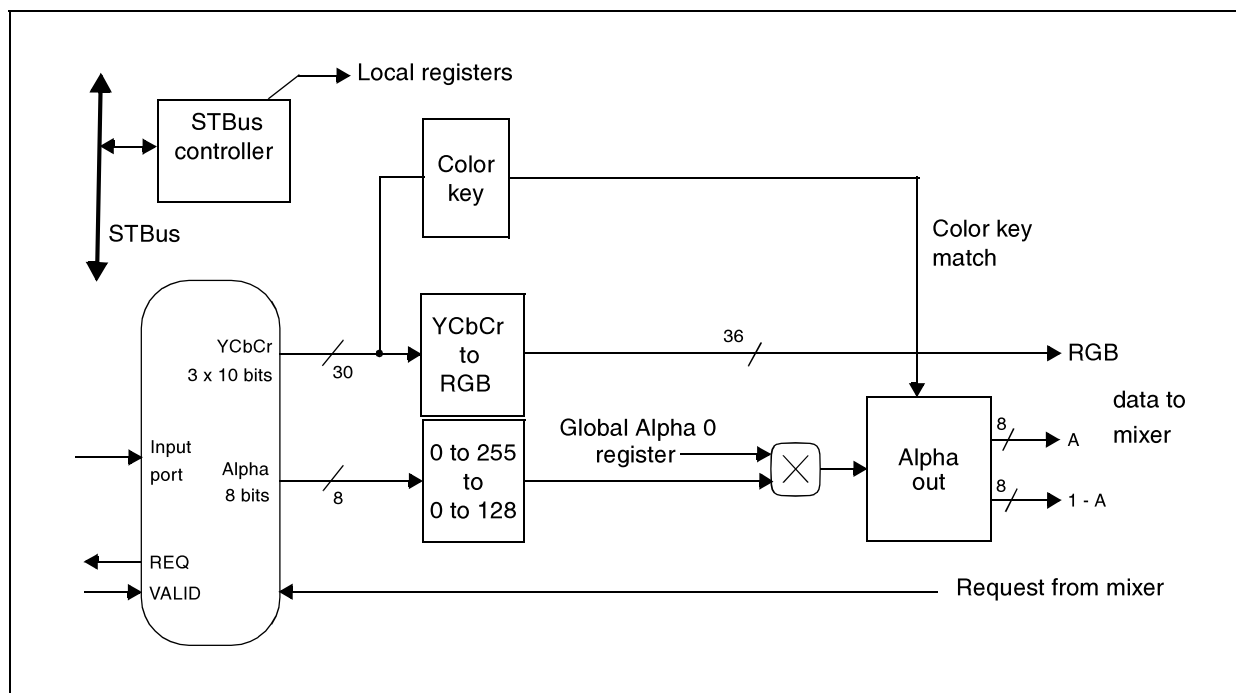
## 51.7 Video plug (VID)

The STx7100 compositor has two video input plugs, one main (VID1) and one auxiliary (VID2). These inputs accept 3 x 10 bit YCbCr video from the main and auxiliary display processors respectively. Figure 181 shows a functional block diagram of the video input plug. The Alpha channel from the input port is not used, so that the alpha weights provided to the mixers are purely based on the global alpha values in the GAM\_VIDn\_ALP registers.

The plug provides the following features:

- Color key capability,
- Global alpha blending (combined with the per pixel alpha channel, if any),
- Vertical and horizontal edge smoothing.

Figure 181: VID functional block diagram



### 51.7.1 Color data path

- Programmable matrix: all input components are 10-bit signed. The nominal range is:  
 $-448 \leq Y \leq +373$   
 $-420 \leq Cb, Cr \leq +420$

Prior to the matrix, a  $(512 - \text{GAM\_VIDn\_MPR1.Y\_OS})$  offset is added, in order to recover an unsigned luminance component.

Fixed video color space matrix:

Table 167: 601 colorimetry / integer matrix / digital range

RGB to YCbCr integer matrix as implemented									
Y	=	77/256	xR	+	150/256	xG	+	29/256	xB
		306/1024			601/1024			117/1024	
Cb	=	- 44/256	xR	-	87/256	xG	+	131/256	xB + 128
		177/1024			347/1024			523/1023	
Cr	=	131/256	xR	-	110/256	xG	-	21/256	xB + 128
		523/1024			438/1024			85/1024	

Table 168: 709 colorimetry / integer matrix / digital range

RGB to YCbCr integer matrix as implemented										
Y	=	54/256	xR	+	183/256	xG	+	19/256	xB	
		218/1024			732/1024			74/1024		
Cb	=	- 30/256	xR	-	101/256	xG	+	131/256	xB	+
		120/1024			404/1024			524/1024		128
Cr	=	131/256	xR	-	119/256	xG	-	12/256	xB	+
		524/1024			476/1024			48/1024		128

The matrices are providing 12 bit signed gamma-corrected RGB components, with a dynamic range (-2048 to 2047).

The color key processor operates in the YCbCr color space, according to the following equation:

```

COLOR_KEY_MATCH =
((CRin < CRmin) OR (CRin > CRmax)) AND (CRoutside = True) AND (EnableCR = True)
OR (CRmin <= CRin <= CRmax) AND (CRoutside = False) AND (EnableCR = True)
OR (EnableCR = False)
AND
((Yin < Ymin) OR (Yin > Ymax)) AND (Youtside = True) AND (EnableY = True)
OR ( Ymin <= Yin <= Ymax) AND (Youtside = False) AND (EnableY = True)
OR (EnableY = False)
AND
((CBin < CBmin) OR (CBin > CBmax)) AND (CBoutside = True) AND (EnableCB = True)
OR ( CBmin <= CBin <= CBmax) AND (CBoutside = False) AND (EnableCB = True)
OR (EnableCB = False)

```

*Note: The min and max registers are 8-bit registers (GAM\_VIDn\_KEY1/2). Thus, the two LSBs of the incoming components are ignored by the color key processor.*

### 51.7.2 Alpha data path

The alpha data path provides a programmable global alpha value for the video layer.

The alpha out block then performs the following processing:

- Alpha borders: the alpha component can be divided by two on the first and last viewport columns ([GAM\\_VIDn\\_CTRL.AB\\_H](#)) and/or on the first and last viewport lines ([AB\\_V](#)).
- If the color key is enabled and there is a match, the alpha output is forced to 0, so that the pixel is transparent.
- Both alpha and 128-alpha are provided to the next processing blocks (digital mixer).

## 51.8 Alpha plane (ALP)

This is an alpha-only version of the GDP pipeline, providing an 8-bit alpha bus. Only the differences with the GDP specification are provided here.

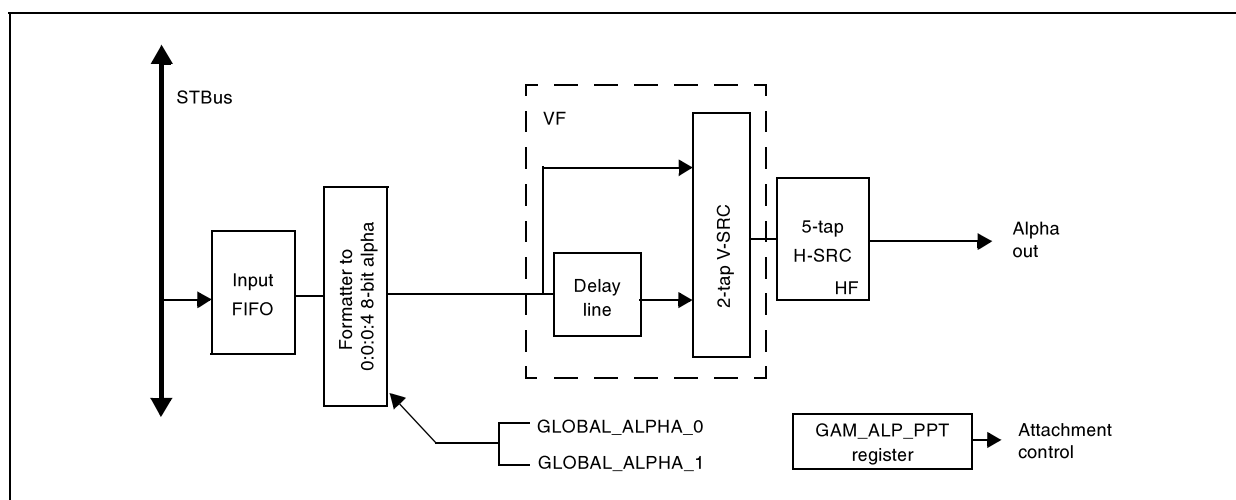
This alpha component can then be attached to a color layer (video or graphics), and is combined with the alpha component of the layer. This occurs in the digital mixer block. The features are:

- Link-list-based display engine, for multiple viewport capabilities.
- A1 and A8 formats supported (0 to 128 or 0 to 255 range for A8).
- 5-tap horizontal sample rate converter, for horizontal upsampling. This can be used to adapt the pixel aspect ratio.

The resolution is 1/8th pixel (polyphase filter with 8 subpositions)

The ALP output format is an 8-bit alpha value, ranging from 0 to 128.

**Figure 182: ALP functional block diagram**



### 51.8.1 General description

#### ALP overview

ALP can handle multiple-viewport display, from a display instruction list (link list) stored in the external memory. Typically, an alpha plane viewport should be set up to match exactly the video or graphics viewport it is attached to.

The alpha plane can be used only on MIX1.

For the viewport and window definition, refer to the relevant GDP subsection. Outside a viewport, the alpha plane pipeline outputs 128.

#### How to start the display and modify the display parameters

This is similar to GDP; refer to the relevant GDP subsection.

#### Bandwidth considerations

In terms of bandwidth requirement (BR), the general formula to estimate the ALP weight on a given system is the following:

$$\text{BR (in Mbyte/s)} = (\text{pixel frequency in MHz}) \times (\text{number of bits per pixel}) \times (\text{horizontal resampling factor}) / 8$$

Examples:

- 601 rate / A1 / x1: BR = 13.5 x 1 x 1 / 8 = 1.69 Mbyte/s
- PAL SQ rate / A8 / x1: BR = 14.75 x 8 x 1 / 8 = 14.75 Mbyte/s
- 601 rate / A8 / zoom out x2: BR = 13.5 x 8 x 2 / 8 = 27 Mbyte/s

## 51.8.2 Unitary functions

### Input formatter

This sub-block converts the 128-bit STBus data bus, into an internal 8-bit alpha bus. The clock rate for the pixel bus is PIXCLK.

There is no endianness issue concerning the bitmap, as the format is either 1 or 8 bpp.

Bit [GAM\\_ALP\\_PKZ.BIGNOTLITTLE](#) (that is not included in the viewport node) indicates if the nodes and filter coefficients are stored in big or little endian. This bit should reflect the endianness of the host CPU. It performs a four-byte swap on the 32-bit word fetched when loading the nodes and the filter coefficients.

**Table 169: 8-bit alpha bus format, at the formatter output**

Input color format	Alpha
A1	GLOBAL_ALPHA_0 if A1 = 0 GLOBAL_ALPHA_1 if A1 = 1
A8	A8 <sup>a</sup>

- a. The ALP supports either 0 to 128 or 0 to 255 8-bit alpha. In case 255 range is used, the input alpha value is converted in to a 0 to 128 component, using the following formula:

$$A_{0..128} = (A_{0..255} + 1) \times 2^{-1}$$

In A1 mode, when the bit is 0, then the alpha output is forced to the GLOBAL\_ALPHA\_0 value, when the bit is 1, the alpha output is forced to the GLOBAL\_ALPHA\_1 value.

*The output of the formatter can be optionally complemented: (128 - Alpha) operator is applied if [GAM\\_ALP\\_CTRL.REV\\_ALPHA\\_EN](#) is set to 1.*

### 2D-resize, vertical filter and horizontal filter

This is similar to GDP; refer to [Section 51.5.1: General description of GDPs on page 538](#).

## 52 Compositor registers

### 52.1 Introduction

Register addresses are provided as *CompositorBaseAddress* + *SubsystemOffset* + register offset.

The *CompositorBaseAddress* is:

0x1920 A000.

Subsystems and *SubsystemOffsets* are shown in [Table 170](#).

**Table 170: Compositor subsystems and subsystem register offsets**

Subsystem	Offset	
	Name	Value
CURSOR	<i>CUROffset</i>	0x0000
GDP1	<i>GDP1Offset</i>	0x0100
GDP2	<i>GDP2Offset</i>	0x0200
ALP	<i>ALPOffset</i>	0x0600
VID1	<i>VID1Offset</i>	0x0700
VID2	<i>VID2Offset</i>	0x0800
MIX1	<i>MIX1Offset</i>	0x0C00
MIX2	<i>MIX2Offset</i>	0x0D00

Each subsystem register block occupies a 64 x 32-bit word address range.

Confidential

Table 171 shows the full list of compositor registers, in a matrix format. Each register name can be built up from the table as follows: GAM\_subsystem\_function, for example GAM\_CUR\_CTRL.

Table 171: Compositor register summary

Offset (0x)	Subsystem					
	CUR	GDPn	VIDn	ALP	MIX1	MIX2
0000	CTRL	CTRL	CTRL	CTRL	CTRL	CTRL
0004		AGC	ALP	AGC	BKC	
0008		HSRC		HSRC		
000C	VPO	VPO	VPO	VPO	BCO	
0010		VPS	VPS	VPS	BCS	
0014	PML	PML		PML		
0018	PMP	PMP		PMP		
001C	SIZE	SIZE		SIZE		
0020	CML	VSRC		VSRC		
0024		NVN		NVN		
0028	AWS	KEY1	KEY1		AVO	AVO
002C	AWE	KEY2	KEY2		AVS	AVS
0030		HFP		HFP		
0034		PPT		PPT	CRB	
0038					ACT	ACT
003C						
0040		HFC0		HFC0		
0044		HFC1		HFC1		
0048		HFC2		HFC2		
004C		HFC3		HFC3		
0050		HFC4		HFC4		
0054		HFC5		HFC5		
0058		HFC6		HFC6		
005C		HFC7		HFC7		
0060		HFC8		HFC8		
0064		HFC9		HFC9		
0070			BC			
0074			TINT			
0078			CSAT			
00FC	PKZ	PKZ		PKZ		

Complete descriptions of these registers are given in the following sections.

## 52.2 Register maps

Table 172: Mixers 1 and 2, cursor, video-plug and capture register map

Description	Register	Offset	Type
Control	<a href="#">GAM_MIX1_CTRL</a>	0x0000	R/W
	<a href="#">GAM_MIX1_CRB</a>	0x0034	
	<a href="#">GAM_MIX1_ACT</a>	0x0038	
Background color	<a href="#">GAM_MIX1_BKC</a>	0x0004	
	<a href="#">GAM_MIX1_BCO</a>	0x000C	
	<a href="#">GAM_MIX1_BCS</a>	0x0010	
Active video area definition	<a href="#">GAM_MIX1_AVO</a>	0x0028	
	<a href="#">GAM_MIX1_AVS</a>	0x002C	
Control	<a href="#">GAM_MIX2_CTRL</a>	0x0000	
	<a href="#">GAM_MIX2_ACT</a>	0x0038	
Active video area definition	<a href="#">GAM_MIX2_AVO</a>	0x0028	
	<a href="#">GAM_MIX2_AVS</a>	0x002C	
Control register	<a href="#">GAM_CUR_CTRL</a>	0x0000	
Viewport offset	<a href="#">GAM_CUR_VPO</a>	0x000C	
Pixmap	<a href="#">GAM_CUR_PML</a>	0x0014	
	<a href="#">GAM_CUR_PMP</a>	0x0018	
	<a href="#">GAM_CUR_SIZE</a>	0x001C	
	<a href="#">GAM_CUR_CML</a>	0x0020	
Clipping window	<a href="#">GAM_CUR_AWS</a>	0x0028	
	<a href="#">GAM_CUR_AWE</a>	0x002C	
Miscellaneous	<a href="#">GAM_CUR_PKZ</a>	0x00FC	
Control register	<a href="#">GAM_VIDn_CTRL</a>	0x0000	
Alpha blending	<a href="#">GAM_VIDn_ALP</a>	0x0004	
Video viewport definition	<a href="#">GAM_VIDn_VPO</a>	0x000C	
	<a href="#">GAM_VIDn_VPS</a>	0x0010	
Color key control	<a href="#">GAM_VIDn_KEY1</a>	0x0028	
	<a href="#">GAM_VIDn_KEY2</a>	0x002C	
Picture control	<a href="#">GAM_VIDn_BC</a>	0x0070	
	<a href="#">GAM_VIDn_TINT</a>	0x0074	
	<a href="#">GAM_VIDn_CSAT</a>	0x0078	
Reserved	-	0x0030 - 0x003C	

Confidential



Table 173: Generic display pipeline register map

Description	Register	Offset	Type	128-bit word alignment				
Control register	<a href="#">GAM_GDPn_CTRL</a>	0x0000	RO/LLU	Node	Memory node			
Blending and dynamic range control	<a href="#">GAM_GDPn_AGC</a>	0x0004		128-bit word 1				
Horizontal sample rate converter control	<a href="#">GAM_GDPn_HSRC</a>	0x0008						
Viewport definition	<a href="#">GAM_GDPn_VPO</a>	0x000C						
	<a href="#">GAM_GDPn_VPS</a>	0x0010	RO/LLU	Node				
Pixmap-related settings	<a href="#">GAM_GDPn_PML</a>	0x0014		128-bit word 2				
	<a href="#">GAM_GDPn_PMP</a>	0x0018						
	<a href="#">GAM_GDPn_SIZE</a>	0x001C						
Reserved	-	0x0020	-	Node				
Pointer to the next viewport node	<a href="#">GAM_GDPn_NVN</a>	0x0024	RO/LLU	128-bit word 3				
Color key	<a href="#">GAM_GDPn_KEY1</a>	0x0028						
	<a href="#">GAM_GDPn_KEY2</a>	0x002C						
Pointer the horizontal filter coefficients	<a href="#">GAM_GDPn_HFP</a>	0x0030	RO/LLU	Node	Memory node			
Viewport properties	<a href="#">GAM_GDPn_PPT</a>	0x0034		128-bit word 4				
Reserved		0x0038	-					
		0x003C						
Horizontal filter coefficients	<a href="#">GAM_GDPn_HFCn</a>				Filter coefficient structure			
	<a href="#">GAM_GDPn_HFC0</a>	0x0040	RO/LLU	HF coefficient				
	<a href="#">GAM_GDPn_HFC1</a>	0x0044		128-bit word 1				
	<a href="#">GAM_GDPn_HFC2</a>	0x0048						
	<a href="#">GAM_GDPn_HFC3</a>	0x004C						
	<a href="#">GAM_GDPn_HFC4</a>	0x0050	RO/LLU	HF coefficient				
	<a href="#">GAM_GDPn_HFC5</a>	0x0054		128-bit word 2				
	<a href="#">GAM_GDPn_HFC6</a>	0x0058						
	<a href="#">GAM_GDPn_HFC7</a>	0x005C						
	<a href="#">GAM_GDPn_HFC8</a>	0x0060	RO/LLU	HF coefficient 128-bit word				
	<a href="#">GAM_GDPn_HFC9</a>	0x0064						
Reserved	-	0x0068 - 0x006C	-	3				
STBus protocol/maximum packet size	<a href="#">GAM_GDPn_PKZ</a>	0x00FC	R/W	-				

Confidential

Table 174: Alpha plane register map

Description	Register	Offset	Type	128-bit word alignment		
Control register	GAM_ALP_CTRL	0x0000	RO/LLU	Node	Memory node	
	GAM_ALP_ALP	0x0004		128-bit word 1		
Horizontal sample rate converter control	GAM_ALP_HSRC	0x0008				
Viewport definition	GAM_ALP_VPO	0x000C	RO/LLU	Node		128-bit word 2
	GAM_ALP_VPS	0x0010				
Pixmap-related settings	GAM_ALP_PML	0x0014				
	GAM_ALP_PMP	0x0018				
	GAM_ALP_SIZE	0x001C				
Reserved	-	0x0020	-	Node		128-bit word 3
Pointer to the next viewport node	GAM_ALP_NVN	0x0024	R/W/LLU			
Reserved	-	0x0028	-			
		0x002C				
Pointer the horizontal filter coefficients	GAM_ALP_HFP	0x0030	RO/LLU	Node	128-bit word 4	
Viewport properties	GAM_ALP_PPT	0x0034				
Reserved	-	0x0038	-			
		0x003C				
Horizontal filter coefficients	GAM_ALP_HFCn				Filter coefficient structure	
	GAM_ALP_HFC0	0x0040	RO/LLU	HF coefficient.		
	GAM_ALP_HFC1	0x0044		128-bit word 1		
	GAM_ALP_HFC2	0x0048				
	GAM_ALP_HFC3	0x004C				
	GAM_ALP_HFC4	0x0050	RO/LLU	HF coefficient		
	GAM_ALP_HFC5	0x0054		128-bit word 2		
	GAM_ALP_HFC6	0x0058				
	GAM_ALP_HFC7	0x005C				
	GAM_ALP_HFC8	0x0060	RO/LLU	HF coefficient		
	GAM_ALP_HFC9	0x0064		128-bit word 3		
Reserved		0x0068 - 0x006C	-			
STBus protocol / packet maximum size	GAM_ALP_PKZ	0x00FC	R/W	Not relevant		

Confidential

## 52.3 Register descriptions

### GAM\_ALP\_CTRL

### ALP control

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
WAIT_NEXT_VSYNC		HFILTER_UPDATE_EN		REV_ALPHA_EN		Reserved		A1_INBYTE_ORDER		A1_BYTE_START											Reserved											H_RESIZE_EN		Reserved			ALPHA_RANGE		ALPHA_FORMAT					

Address: *CompositorBaseAddress + ALPOffset + 0x00*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the operating mode of the ALP pipe, for the current viewport.

[31] **WAIT\_NEXT\_VSYNC**

- 0: The next node (as specified by [GAM\\_ALP\\_NVN](#)) is immediately loaded
- 1: ALP pipeline waits for the next VSync event before it loads the next node

[30] **HFILTER\_UPDATE\_EN**: This bit is taken into account for any viewport node within a frame (field).

- 0: The coefficients for the H filter are not loaded
- 1: The coefficients for the H filter are updated from memory (see [GAM\\_ALP\\_HFP](#))

[29] **REV\_ALPHA\_EN**: In reverse mode, (128-alpha) operator is applied at the pipeline input.

- 0: Alpha mode
- 1: Reverse alpha mode

[28] **Reserved**

[27] **A1\_INBYTE\_ORDER**: A1 format only: specifies sample ordering inside a byte

- 0: Screen most right sample in the MSB
- 1: Screen most right sample in the LSB

[26:24] **A1\_BYTE\_START**: A1 format only: specifies the bit location of the first alpha sample, in the first byte

- 000: First 1-bit alpha sample is bit 0
- 001: First 1-bit alpha sample is bit 1
- 
- 111: First 1-bit alpha sample is bit 7

[23:11] **Reserved**: Set to 0

[10] **H\_RESIZE\_EN**: Horizontal resize enable

- 0: Disabled
- 1: Enabled

[9:6] **Reserved**

[5] **ALPHA\_RANGE**: for A8 format, this bit specifies the alpha range.

- 0: 0 to 128 range (128 = opaque)
- 1: 0 to 255 range (255 = opaque)

[4:0] **ALPHA\_FORMAT**: alpha format for the bitmap associated to the current viewport

- 11000: A1 (0x18)
- 11001: A8 (0x19)

**GAM\_ALP\_ALP****ALP alpha**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																GLOBAL_ALPHA_1								GLOBAL_ALPHA_0							

Address: *CompositorBaseAddress* + *ALPOffset* + 0x04

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the 2 8-bit alpha values to be used when A1 format is selected at the pipeline input. This register has no effect when A8 is selected.

[31:16] **Reserved**

[15:8] **GLOBAL\_ALPHA\_1**: For A1 format, this 8-bit value is used if the input bit is one.  
The register range is 0 to 128. (0: Fully transparent, 128: Fully opaque)

[7:0] **GLOBAL\_ALPHA\_0**: For A1 format, this 8-bit value is used if the input bit is zero.  
The register range is 0 to 128. (0: Fully transparent, 128: Fully opaque)

**GAM\_ALP\_HSRC****ALP horizontal sample rate converter**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							HF_FILTER_MODE	Reserved							HSRC_INIT_PHASE	Reserved							HSRC_INC								

Address: *CompositorBaseAddress* + *ALPOffset* + 0x08

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the configuration for the horizontal sample rate converter. The alpha plane HSRC should only be programmed for upscaling.

[31:25] **Reserved**

[24] **HF\_FILTER\_MODE**  
0: Only horizontal resizing  
1: Filter is enabled

[23:19] **Reserved**

[18:16] **HSRC\_INIT\_PHASE**: The horizontal sample rate converter state-machine initial phase, 0 to 7

[15:10] **Reserved**

[9:0] **HSRC\_INC**: The horizontal sample rate converter state-machine increment, in 2.8 format

**GAM\_ALP\_VPO****ALP viewport offset**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDO	Reserved	XDO
----------	-----	----------	-----

Address: *CompositorBaseAddress + ALPOffset + 0x0C*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the x, y location of the viewport top-left pixel, with respect to the current video timebase.

[31:27] **Reserved**

[26:16] **YDO**: Y location for the first line of the viewport (top), with respect to frame numbering

[15:12] **Reserved**

[11:0] **XDO**: X location for the first pixel of the viewport (left)

**GAM\_ALP\_VPS****ALP viewport stop**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDS	Reserved	XDS
----------	-----	----------	-----

Address: *CompositorBaseAddress + ALPOffset + 0x10*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the x, y location of the viewport bottom-right pixel, with respect to the current video timebase.

[31:27] **Reserved**

[26:16] **YDS**: Y location for the last line of the viewport (bottom), with respect to frame numbering

[15:12] **Reserved**

[11:0] **XDS**: X location for the last pixel of the viewport (right)

**GAM\_ALP\_PML****ALP pixmap memory location**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

64MB_BANK	PIXMAP_ADDR
-----------	-------------

Address: *CompositorBaseAddress + ALPOffset + 0x14*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains the memory location for the first alpha component to be read (top-left corner).

[31:26] **64MB\_BANK**: 64 Mbyte bank number

[25:0] **PIXMAP\_ADDR**: First pixel byte address, in the selected 64 Mbyte bank.

*Note: The whole bitmap to be displayed must be totally included in the same bank.*

Confidential

**GAM\_ALP\_PMP****ALP pixmap memory pitch**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PITCH_VAL															

Address: *CompositorBaseAddress* + *ALPOffset* + 0x18

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains the memory pitch for the alpha bitmap, as stored in the memory.

[31:16] **Reserved**

[15:0] **PITCH\_VAL**: Memory pitch for the displayed pixmap

*Note: the pitch is the distance inside the memory, in bytes, between two vertically adjacent pixels.*

**GAM\_ALP\_SIZE****ALP pixmap size**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					PIXMAP_HEIGHT											Reserved					PIXMAP_WIDTH										

Address: *CompositorBaseAddress* + *ALPOffset* + 0x1C

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the size of the alpha bitmap attached to the viewport.

[31:27] **Reserved**

[26:16] **PIXMAP\_HEIGHT**

Pixmap height in lines, being defined as the number of lines that must be read from memory for the current field in an interlaced display.

[15:11] **Reserved**

[10:0] **PIXMAP\_WIDTH**: Pixmap width in pixels

Confidential

**GAM\_ALP\_NVN****ALP next viewport node**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK						NEXT_NODE_ADDR																						Reserved			

Address: *CompositorBaseAddress + ALPOffset + 0x24*

Type: R/W/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains a memory pointer to the next viewport node to be executed within the link list.

*Note: The CPU can directly write into this register; this is required at least to enable the DMA link-list process to start.*

[31:26] **64MB\_BANK**: 64 Mbyte bank number

[25:4] **NEXT\_NODE\_ADDR**: Memory location for the next node to be loaded (the node must be fully contained in the specified bank). 4 LSBs address bits are “don’t care”, because the node structure must be aligned on a 128-bit word boundary.

[3:0] **Reserved**

**GAM\_ALP\_HFP****ALP horizontal filter pointer**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK						H_FILTER_PTR																						Reserved			

Address: *CompositorBaseAddress + ALPOffset + 0x30*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains a memory pointer to the set of filter coefficients that must be used for the horizontal sample rate converter.

A new set of coefficients may be used for each individual viewport. The coefficients are loaded only if **GAM\_ALP\_CTRL**[30] = 1 (HFILTER\_UPDATE\_EN).

[31:26] **64MB\_BANK**: 64 Mbyte bank number

[25:4] **H\_FILTER\_PTR**

Memory location when to retrieve the filter coefficients (10 32-bit words that must be fully contained in the specified bank). 4 LSBs address bits are “don’t care”, because the filter coefficients structure must be aligned on a 128-bit word boundary.

[3:0] **Reserved**

Confidential

**GAM\_ALP\_PPT****ALP properties**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								ALPHA_ ATTACHMENT		Reserved					

Address: *CompositorBaseAddress + ALPOffset + 0x34*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: Provides the mixer(s) with special information on the viewport currently displayed.

[31:8] **Reserved**

[7:4] **ALPHA\_ATTACHMENT**: This field indicates the layer the alpha plane must be combined with.

0000: no attachment

0001: attach to VID1

0010: Reserved

0011: attach to GDP1

0100: attach to GDP2

0101: Reserved

0110: Reserved

0111: Reserved

1XXX: Reserved

[3:0] **Reserved**

**GAM\_ALP\_HFCn****ALP HF coefficients**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Address: *CompositorBaseAddress + ALPOffset + register offset*

HFC0: 0x40, HFC1: 0x44, HFC2: 0x48, HFC3: 0x4C, HFC4: 0x50, HFC5: 0x54,

HFC6: 0x58, HFC7: 0x5C, HFC8: 0x60, HFC9: 0x64

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Confidential



**GAM\_ALP\_PKZ****ALP maximum packet size**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										BIGNOTLITTLE	Reserved	PKT_SIZE			

Address: *CompositorBaseAddress + ALPOffset + 0xFC*

Type: R/W

Buffer: Immediate

Reset: 0x10

Description: This register is a 3-bit register for controlling the maximum size of a data packet during an STBus transaction. These bits should be set to 0 in the STx7100 device.

[31:6] **Reserved**

[5] **BIGNOTLITTLE**: CPU endianness

0: little endian CPU

1: big endian CPU

[4:3] **Reserved**: Set to 0

[2:0] **PKT\_SIZE**: Maximum packet size during an STBus transaction

000: message size

001: 16 STBus words

010: 8 STBus words

011: 4 STBus words

100: 2 STBus words

101: 1 STBus words

This is a static register (not part of the link-list).

**GAM\_CUR\_CTRL****CUR control**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																															REFRESH	Reserved

Address: *CompositorBaseAddress + CUROffset + 0x00*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for controlling the CUR core.

[31:2] **Reserved**

[1] **REFRESH**: 0: Disable cursor CLUT refresh

1: Enable cursor CLUT refresh

[0] **Reserved**

Confidential

**GAM\_CUR\_VPO****Viewport offset**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				YDO								Reserved				XDO															

Address: *CompositorBaseAddress + CUROffset + 0x0C*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register provides the screen x, y location of the cursor pattern (upper-left pixel).

[31:28] **Reserved**

[27:16] **YDO**: Vertical start location for CUR viewport, wrt line frame-numbering

[15:13] **Reserved**

[12:0] **XDO**: Horizontal start location for CUR viewport, wrt VTG horizontal counter

**GAM\_CUR\_PML****First pixel memory location**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUR_BANK_NUM								CUR_MEM_ADDR																							

Address: *CompositorBaseAddress + CUROffset + 0x14*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the address for the first pixel memory location.

[31:26] **CUR\_BANK\_NUM**: 64 Mb bank number

[25:0] **CUR\_MEM\_ADDR**: First pixel memory byte address

**GAM\_CUR\_PMP****Pixmap memory pitch**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PIX_MEM_PITCH															

Address: *CompositorBaseAddress + CUROffset + 0x18*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the memory pitch for the cursor bitmap.

[31:16] **Reserved**

[15:0] **PIX\_MEM\_PITCH**: CUR pixmap memory pitch in bytes

Confidential

**GAM\_CUR\_SIZE****Pixmap memory size**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PIXMAP_HEIGHT								Reserved								PIXMAP_WIDTH							

Address: *CompositorBaseAddress* + *CUROffset* + 0x1C

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the horizontal and vertical size of the cursor pattern.

[31:24] **Reserved**

[23:16] **PIXMAP\_HEIGHT**: Cursor pattern height (in pixel unit)

[15:8] **Reserved**

[7:0] **PIXMAP\_WIDTH**: Cursor pattern width (in pixel units)

**GAM\_CUR\_CML****CLUT memory location**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLUT_BANK_NUM								CLUT_MEM_ADDR																							

Address: *CompositorBaseAddress* + *CUROffset* + 0x20

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the memory pointer for the color look-up table (CLUT).

[31:26] **CLUT\_BANK\_NUM**: 64 Mb bank number

[25:0] **CLUT\_MEM\_ADDR**: CLUT memory byte address

**GAM\_CUR\_AWS****Active window start**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								AWS_Y								Reserved								AWS_X							

Address: *CompositorBaseAddress* + *CUROffset* + 0x28

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the coordinates for the top-left location of the cursor active window.

[31:27] **Reserved**

[26:16] **AWS\_Y**: Vertical coordinate

[15:12] **Reserved**

[11:0] **AWS\_X**: Horizontal coordinate

Confidential

**GAM\_CUR\_AWE**      **Active window end**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						AWE_Y										Reserved						AWE_X									

Address: *CompositorBaseAddress + CUROffset + 0x2C*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: Contains the coordinates for the bottom-right location of the cursor active window.

[31:27] **Reserved**

[26:16] **AWE\_Y**: Vertical coordinate

[15:12] **Reserved**

[11:0] **AWE\_X**: Horizontal coordinate

**GAM\_CUR\_PKZ**      **Packet size control**

31	30	29	28	28	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																									BIGNOTLITTLE	Reserved		PKT_SIZE			

Address: *CompositorBaseAddress + CUROffset + 0xFC*

Type: R/W

Buffer: Immediate

Reset: 0

Description: Controls the packet size on an STBus transaction. In this device, the packet size must be set to 0.

[31:3] **Reserved**

[5] **BIGNOTLITTLE**: CLUT endianness  
0: Little endian CLUT

1: Big endian CLUT

[4:3] **Reserved**

[2:0] **PKT\_SIZE**: Packet size

000: Packet size = message size

010: Packet size = 8 x 128-bit words

100: Packet size = 2 x 128-bit words

001: Packet size = 16 x 128-bit words

011: Packet size = 4 x 128-bit words

101: Packet size = 1 x 128-bit words

Confidential

## GAM\_GDPn\_CTRL

## GDP control

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAIT_NEXT_VSYNC	HFILTER_UPDATE_EN	LSB_STUFFING_MODE	Reserved	CHROMA_FORMAT	601/709_SEL	PREMULT_FORMAT	BIGNOTLITTLE	Reserved	R/CR_COLOR_KEY_CFG	G/Y_COLOR_KEY_CFG	B/CB_COLOR_KEY_CFG	Reserved	COLOR_KEY_EN	ALPHA_VBORDER	ALPHA_HBORDER_EN	Reserved	H_RESIZE_EN	Reserved	ALPHA_RANGE	COLOR_FORMAT											

Address: *CompositorBaseAddress + GDPnOffset + 0x00*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the operating mode of the GDP pipe for the current viewport display.

[31] **WAIT\_NEXT\_VSYNC**

- 0: The next node (as specified by [GAM\\_GDPn\\_NVN](#)) is immediately loaded  
 1: The GDP must wait for the next Vsync event before it loads the next node

[30] **HFILTER\_UPDATE\_EN**

This bit is taken into account for any viewport node within a frame (field)

- 0: Coefficients for the H filter are not loaded  
 1: Coefficients for the H filter are updated from memory (see [GAM\\_GDPn\\_HFP](#))

[29] **LSB\_STUFFING\_MODE**

This configuration bit is used by the input formatter, in order to build the 32-bit internal pixel bus, whatever the input color format. When set, this bit preserves the full dynamic range 0.0 to 1.0, whatever the input format.

*Note: The color key is affected by this setting.*

- 0: If the number of bits per component at the input is fewer than 8, missing LSBs are filled with 0.  
 1: If the number of bits per component at the input is fewer than 8, missing LSBs are filled by the appropriate number of MSBs.

[28:27] **Reserved**

[26] **CHROMA\_FORMAT**

- 0: The color space converter assumes Cb/Cr use offset binary representation  
 1: The color space converter assumes Cb/Cr use two's complement signed representation

[25] **601/709\_SEL**

- 0: The color space conversion uses the 601 colorimetry  
 1: The color space conversion uses the 709 colorimetry

[24] **PREMULT\_FORMAT**: Premultiplied format

- 0: RGB components are not premultiplied by the alpha component  
 1: RGB components are premultiplied by the per-pixel alpha component  
*Note: This is only meaningful for ARGB4444, ARGB8565 and ARGB8888 formats.*

[23] **BIGNOTLITTLE**: 0: Little endian bitmap                      1: Big endian bitmap

[23:22] **Reserved**

[21:20] **R/CR\_COLOR\_KEY\_CFG**: R/CR color key configuration

- x0: R/Cr component ignored (disabled = always match)  
 01: R/Cr enabled: match if ( $R/Cr_{min} \leq R/Cr \leq R/Cr_{max}$ )  
 11: R/Cr enabled: match if ( $(R/Cr < R/Cr_{min})$  or  $(R/Cr > R/Cr_{max})$ )

- [19:18] **G/Y\_COLOR\_KEY\_CFG**: G/Y color key configuration  
x0: G/Y component ignored (disabled = always match)  
01: G/Y enabled: match if ( $G/Y_{\min} \leq G/Y \leq G/Y_{\max}$ )  
11: G/Y enabled: match if ( $(G/Y < G/Y_{\min})$  or  $(G/Y > G/Y_{\max})$ )
- [17:16] **B/CB\_COLOR\_KEY\_CFG**: B/CB color key configuration  
x0: B/Cb component ignored (disabled = always match)  
01: B/Cb enabled: match if ( $B/Cb_{\min} \leq B/Cb \leq B/Cb_{\max}$ )  
11: B/Cb enabled: match if ( $(B/Cb < B/Cb_{\min})$  or  $(B/Cb > B/Cb_{\max})$ )
- [15] **Reserved**
- [14] **COLOR\_KEY\_EN**  
0: Color key feature is disabled  
1: Color key feature is enabled
- [13] **ALPHA\_VBORDER\_EN**: Provides soft horizontal edges for the viewport, with flicker reduction effect.  
0: Feature disabled  
1: Alpha component is divided by 2 on the first line and the last line of the viewport
- [12] **ALPHA\_HBORDER\_EN**: Provides soft vertical edges for the viewport.  
0: Feature disabled  
1: Alpha component is divided by 2 on the first and last columns of the viewport
- [11] **Reserved**: Set to 0
- [10] **H\_RESIZE\_EN**  
0: Horizontal resize block is disabled  
1: Horizontal resize block is enabled
- [9:6] **Reserved**
- [5] **ALPHA\_RANGE**: for color format with an 8-bit alpha component (ARGB8888 / ARGB8565), this bit specifies the alpha range.  
0: 0 to 128 range (128 = opaque)  
1: 0 to 255 range (255 = opaque)
- [4:0] **COLOR\_FORMAT**: color format for the bitmap associated to the current viewport  
0 0000: RGB565 (0x0)  
0 0100: ARGB8565 (0x4)  
0 0110: ARGB1555 (0x6)  
1 0000: YCbCr888 (0x10)  
1 0101: AYCbCr8888 (0x15)  
0 0001: RGB888 (0x1)  
0 0101: ARGB8888 (0x5)  
0 0111: ARGB4444 (0x7)  
1 0010: YCbCr4:2:2R (0x12)

**GAM\_GDPn\_AGC****GDP alpha gain constant**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

CONSTANT								GAIN								GLOBAL_ALPHA_1								GLOBAL_ALPHA_0							
----------	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	----------------	--	--	--	--	--	--	--	----------------	--	--	--	--	--	--	--

Address: *CompositorBaseAddress + GDPnOffset + 0x04*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides display parameters, such as global translucency factors, black level adjustment, and dynamic range adjustment.

[31:24] **CONSTANT**

This 8-bit register is used to adjust the black level of the RGB signal at the output of the GDP block, between 0 and 25% of the total dynamic range (that is 10 bit, 0 to 1023).

A value of 0 sets the black level to 0%. A value of 255 sets the black level to 25%.

Warning: The gain register must be consistent with the black level programmed value, otherwise some saturation effects may occur.

[23:16] **GAIN**

This 8-bit register is used to adjust the dynamic range of the RGB components at the output of the GDP block, between 0 and 100%. A value of 128 corresponds to a 100% range

[15:8] **GLOBAL\_ALPHA\_1**

For any color format except ARGB1555, the Alpha 1 register is unused.

For ARGB1555 color format, this is the pixel transparency that is applied to the current pixel if the alpha bit within the pixel (bit 15) is 1.

The register range is 0 to 128. (0: fully transparent, 128: fully opaque)

[7:0] **GLOBAL\_ALPHA\_0**

For any color format except ARGB1555, Alpha 0 is the global transparency associated with the viewport, that is combined with the per-pixel alpha component, if any.

For ARGB1555 color format, this is the pixel transparency that is applied to the current pixel if the alpha bit within the pixel (bit 15) is 0.

The register range is 0 to 128. (0: fully transparent, 128: fully opaque)

**GAM\_GDPn\_HSRC****GDP horizontal sample rate converter**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							HF_FILTER_MODE	Reserved							HSRC_INIT_PHASE	Reserved							HSRC_INC								

Address: *CompositorBaseAddress + GDPnOffset + 0x08*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the configuration for the horizontal sample rate converter. The GDPn plane HSRC should only be programmed for upscaling.

[31:25] **Reserved**

[24] **HF\_FILTER\_MODE**

1: The filter is enabled, otherwise only horizontal resizing.

[23:19] **Reserved**

[18:16] **HSRC\_INIT\_PHASE**: Horizontal sample rate converter state-machine initial phase, 0 to 7.

[15:10] **Reserved**

[9:0] **HSRC\_INC**: Horizontal sample rate converter state-machine increment, in 2.8 format.

**GAM\_GDPn\_VPO****GDP viewport offset**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					YDO							Reserved					XDO														

Address: *CompositorBaseAddress + GDPnOffset + 0x0C*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the x, y location of the viewport top-left pixel, with respect to the current video timebase.

[31:27] **Reserved**

[26:16] **YDO**: Y location for the first line of the viewport (top), with respect to frame numbering.

[15:12] **Reserved**

[11:0] **XDO**: X location for the first pixel of the viewport (left).

Confidential



**GAM\_GDPn\_VPS****GDP viewport stop**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								YDS								Reserved								XDS							

Address: *CompositorBaseAddress + GDPnOffset + 0x10*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the x, y location of the viewport bottom-right pixel, with respect to the current video timebase.

[31:27] **Reserved**

[26:16] **YDS**: Y location for the last line of the viewport (bottom), with respect to frame numbering

[15:12] **Reserved**

[11:0] **XDS**: X location for the last pixel of the viewport (right)

**GAM\_GDPn\_PML****GDP pixmap memory location**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_bank								PIXMAP_ADDR																							

Address: *CompositorBaseAddress + GDPnOffset + 0x14*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains the memory location for the first pixel to be displayed (top-left corner).

[31:26] **64MB\_BANK**: 64 Mbyte bank number

[25:0] **PIXMAP\_ADDR**: First pixel byte address, in the selected 64 Mbyte bank

*Note: the whole bitmap to be displayed must be totally included into the same bank.*

**GAM\_GDPn\_PMP****GDP pixmap memory pitch**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PITCH_VAL															

Address: *CompositorBaseAddress + GDPnOffset + 0x18*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains the memory pitch for the displayed pixmap, as stored in the memory.

[31:16] **Reserved**

[15:0] **PITCH\_VAL**: Memory pitch for the displayed pixmap.

*Note: the pitch is the distance inside the memory, in bytes, between two vertically adjacent pixels.*

**GAM\_GDPn\_SIZE****GDP pixmap size**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						HEIGHT										Reserved						WIDTH									

Address: *CompositorBaseAddress + GDPnOffset + 0x1C*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides the size of the displayed pixmap attached to the viewport.

[31:27] **Reserved**

[26:16] **HEIGHT**: Pixmap height, in lines, being defined as the number of lines that must be read from memory for the current field in an interlaced display

[15:11] **Reserved**

[10:0] **WIDTH**: Pixmap width, in pixels

**GAM\_GDPn\_NVN****GDP next viewport node**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK						NEXT_NODE_ADDR																						Reserved			

Address: *CompositorBaseAddress + GDPnOffset + 0x24*

Type: R/W/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains a memory pointer to the next viewport node to be displayed within the link list.

*Note: The CPU can directly write into this register; this is required at least to enable the display link-list process to start.*

[31:26] **64MB\_BANK**: 64 Mbyte bank number

[25:4] **NEXT\_NODE\_ADDR**: 4 LSBs address bits are “don’t care”, because the node structure must be aligned on a 128-bit word boundary  
Memory location for the next node to be loaded (the node must be fully contained in the specified bank.

[3:0] **Reserved**

Confidential

**GAM\_GDPn\_KEY1****Color keying - lower limit**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								R/CR_MIN								G/Y_MIN								B/CB_MIN							

Address: *CompositorBaseAddress + GDPnOffset + 0x28*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register is a 24-bit register containing the values for the lower limit of the color range to be detected when using the color key feature.

[31:24] **Reserved**

[23:16] **B/CB\_MIN**: Minimum value for the blue component in ARGBargb color formats; minimum value for the Cb component in YCbCr color formats

[15:8] **G/Y\_MIN**: Minimum value for the green component in ARGBargb color formats; minimum value for the luminance component in YCbCr color formats

[7:0] **R/CR\_MIN**: Minimum value for the red component in ARGBargb color formats; minimum value for the Cr component in YCbCr color formats

**GAM\_GDPn\_KEY2****Color keying - upper limit**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								R/CR_MAX								G/Y_MAX								B/CB_MAX							

Address: *CompositorBaseAddress + GDPnOffset + 0x2C*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register is a 24-bit register containing the values for the upper limit of the color range to be detected when using the color key feature.

[31:24] **Reserved**

[23:16] **B/CB\_MAX**: Maximum value for the blue component in ARGBargb color formats; maximum value for the Cb component in YCbCr color formats

[15:8] **G/Y\_MAX**: Maximum value for the green component in ARGBargb color formats; maximum value for the luminance component in YCbCr color formats

[7:0] **R/CR\_MAX**: Maximum value for the red component in ARGBargb color formats; maximum value for the Cr component in YCbCr color formats

Confidential

**GAM\_GDPn\_HFP****GDP horizontal filter pointer**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64MB_BANK						H_FILTER_PTR																								Reserved	

Address: *CompositorBaseAddress + GDPnOffset + 0x30*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register contains a memory pointer to the set of filter coefficients that must be used for the horizontal sample rate converter.

A new set of coefficients may be used for each individual viewport. The coefficients are loaded only if **GAM\_GDPn\_CTRL**[30] = 1 (HFILTER\_UPDATE\_EN).

[31:26] **64MB\_BANK**: 64 Mbyte bank number

[25:4] **H\_FILTER\_PTR**: 4 LSBs address bits are “don’t care”, because the filter coefficients structure must be aligned on a 128-bit word boundary.

Memory location when to retrieve the filter coefficients (ten 32-bit words that must be fully contained in the specified bank)

[3:0] **Reserved**

**GAM\_GDPn\_PPT****GDP properties**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												FORCEON_MIX2	FORCEON_MIX1	GNOREON_MIX2	GNOREON_MIX1

Address: *CompositorBaseAddress + GDPnOffset + 0x34*

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

Description: This register provides special information to the mixer(s), concerning the viewport currently displayed.

[31:4] **Reserved**

[3] **FORCEON\_MIX2**: When set, this bit indicates to MIX2 that the current viewport must always be displayed, even out of the MIX2 active video area (defined with MIX2\_AVO / MIX2\_AVS registers).  
*Note: The GDP pipeline must be enabled in the MIX2\_CTRL register.*

[2] **FORCEON\_MIX1**: When set, this bit indicates to MIX1 that the current viewport must always be displayed, even out of the MIX1 active video area (defined with MIX1\_AVO / MIX1\_AVS registers).  
*Note: The GDP pipeline must be enabled in the MIX1\_CTRL register.*

[1] **IGNOREON\_MIX2**: When set, this bit indicates to MIX2 that the current viewport must not be displayed, although the GDP pipeline is enabled in the MIX2\_CTRL register

[0] **IGNOREON\_MIX1**: When set, this bit indicates to MIX1 that the current viewport must not be displayed, although the GDP pipeline is enabled in the MIX1\_CTRL register

**GAM\_GDPn\_HFCn      GDP HF coefficients registers**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

--	--	--	--

Address: *CompositorBaseAddress* + *GDPnOffset* + register offset  
 HFC0: 0x40, HFC1: 0x44, HFC2: 0x48, HFC3: 0x4C, HFC4: 0x50, HFC5: 0x54,  
 HFC6: 0x58, HFC7: 0x5C, HFC8: 0x60, HFC9: 0x64

Type: Read/link list update

Buffer: Double-bank, automatic hardware toggle

Reset: 0

**GAM\_GDPn\_PKZ      GDP maximum packet size**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved					BIGNOTLITTLE	Reserved	PKT_SIZE
----------	--	--	--	--	--------------	----------	----------

Address: *CompositorBaseAddress* + *GDPnOffset* + 0xFC

Type: R/W

Buffer: Immediate

Reset: 0x10

Description: This register is a 3-bit register for controlling the maximum size of a data packet during an STBus transaction. These bits must be set to 0 in the STx7100 device.

[31:6] **Reserved**

[5] **BIGNOTLITTLE**: CPU endianness.

0: little endian CPU

1: big endian CPU

[4:3] **Reserved**

Must be set to 0.

[2:0] **PKT\_SIZE**: Maximum packet size during an STBus transaction

000: message size

001: 16 STBus words

010: 8 STBus words

011: 4 STBus words

100: 2 STBus words

101: 1 STBus word

This is a static register (not part of the link-list).

**GAM\_MIX1\_CTRL****MIX control**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																AP_DISP_EN	Reserved				CUR_DISP_EN	Reserved				GDP2_DISP_EN	GDP1_DISP_EN	Reserved	VID1_DISP_EN	BKC_DISP_EN	

Address: *CompositorBaseAddress + MIX1Offset + 0x00*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for controlling the MIX1 core.

[31:16] **Reserved**

[15] **AP\_DISP\_EN**: Alpha plane is enabled for attachment with a given layer that is enabled on the mixer output

[14:10] **Reserved**

[9] **CUR\_DISP\_EN**: CUR display is enabled on the mixer output

[8:5] **Reserved**

[4] **GDP2\_DISP\_EN**: GDP 2 display is enabled on the mixer output

[3] **GDP1\_DISP\_EN**: GDP 1 display is enabled on the mixer output

[2] **Reserved**

[1] **VID1\_DISP\_EN**: Video 1 display is enabled on the mixer output

[0] **BKC\_DISP\_EN**: Background color display is enabled on the mixer output

**GAM\_MIX1\_BKC****Background color**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RED_COMP								GREEN_COMP								BLUE_COMP							

Address: *CompositorBaseAddress + MIX1Offset + 0x04*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the background solid color components used by the MIX1 core.

[31:24] **Reserved**

[23:16] **RED\_COMP**: Red component

[15:8] **GREEN\_COMP**: Green component

[7:0] **BLUE\_COMP**: Blue component

**GAM\_MIX1\_BCO****Background color offset**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDO	Reserved	XDO
----------	-----	----------	-----

Address: *CompositorBaseAddress* + *MIX1Offset* + 0x0C

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the top-left corner of the background color rectangular area. Outside this window, the blanking color is displayed (black).

[31:27] **Reserved**

[26:16] **YDO**: Vertical start location of the MIX background color window, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDO**: Horizontal start location of the MIX background color window, wrt VTG horizontal counter

**GAM\_MIX1\_BCS****Background color stop**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDS	Reserved	XDS
----------	-----	----------	-----

Address: *CompositorBaseAddress* + *MIX1Offset* + 0x10

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the bottom-right corner of the background color rectangular area. Outside this window, the blanking color is displayed (black).

[31:27] **Reserved**

[26:16] **YDS**: Vertical stop location of the MIX background color window, wrt line frame-numbering

[11:0] **XDS**: Horizontal stop location of the MIX background color window, wrt VTG horizontal counter

Confidential

**GAM\_MIX1\_AVO****Active video offset**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					YDO												Reserved					XDO									

Address: *CompositorBaseAddress + MIX1Offset + 0x28*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the top-left corner of the active video rectangular area on the MIX1 output. Typically, it should be programmed according to the video standard currently in use.

[31:27] **Reserved**

[26:16] **YDO**: Vertical start location of the MIX active video window, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDO**: Horizontal start location of the MIX active video window, wrt VTG horizontal counter

**GAM\_MIX1\_AVS****Active video stop**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					YDS												Reserved					XDS									

Address: *CompositorBaseAddress + MIX1Offset + 0x2C*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the bottom-right corner of the active video rectangular area on the MIX output. Typically, it should be programmed according to the video standard currently in use.

[31:27] **Reserved**

[26:16] **YDS**: Vertical stop location of the MIX active video window, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDS**: Horizontal stop location of the MIX active video window, wrt VTG horizontal counter



**GAM\_MIX1\_CRB****MIX 1 cross-bar control**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																DEPTH3		DEPTH2		DEPTH1	
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--------	--	--------	--	--------	--

Address: *CompositorBaseAddress + MIX1Offset + 0x34*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for controlling the mixer 7<>7 cross-bar.

[31:21] **Reserved**

[20:0] **DEPTHn[2:0]**: input identifier for depth *n* (depth 0: background color, depth 8: cursor, depth *n1* in front of depth *n2* if *n1* > *n2*)

000: Nothing displayed at depth *n*

001: VID1 displayed at depth *n*

010: Reserved

011: GDP1 displayed at depth *n*

100 to 111: Reserved

**GAM\_MIX1\_ACT****MIX active content flags control**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																CUR_ONVIDACTIVE	Reserved														GDP2_ONVIDACTIVE	GDP1_ONVIDACTIVE	Reserved		VID1_ONVIDACTIVE	BKC_ONVIDACTIVE	Reserved														CUR_ONGFXACTIVE	Reserved																GDP2_ONGFXACTIVE	GDP1_ONGFXACTIVE	Reserved		VID1_ONGFXACTIVE	BKC_ONGFXACTIVE
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----------------	----------	--	--	--	--	--	--	--	--	--	--	--	--	--	------------------	------------------	----------	--	------------------	-----------------	----------	--	--	--	--	--	--	--	--	--	--	--	--	--	-----------------	----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	------------------	------------------	----------	--	------------------	-----------------

Address: *CompositorBaseAddress + MIX1Offset + 0x38*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: Defines the layers that contribute to the construction of GFXActive1 and VideoActive1 flags. A layer is considered as active for a given pixel if its associated alpha component is not zero.

**XXX\_ONVIDACTIVE**

1: The XXX layer is taken into account for the VideoActive flag.

**XXX\_ONGFXACTIVE**

1: The XXX layer is taken into account for the GFXActive flag.

Confidential

**GAM\_MIX2\_CTRL****MIX control**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												GDP2_DISP_EN	Reserved	VID2_DISP_EN	Reserved

Address: *CompositorBaseAddress + MIX2Offset + 0x00*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for controlling the MIX 2 core.

[31:5] **Reserved**

[4] **GDP2\_DISP\_EN**: GDP2 display enable

1: Enable GDP 2 display on the mixer output

[3] **Reserved**

[2] **VID2\_DISP\_EN**: VID2 display enable

1: Enable video 2 display on the mixer output

[1:0] **Reserved**

**GAM\_MIX2\_AVO****Active video offset**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					YDO										Reserved					XDO											

Address: *CompositorBaseAddress + MIX2Offset + 0x28*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the top-left corner of the active video rectangular area on the MIX2 output. Typically, it should be programmed according to the video standard currently in use.

[31:27] **Reserved**

[26:16] **YDO**: Vertical start location of the MIX active video window, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDO**: Horizontal start location of the MIX active video window, wrt VTG horizontal counter

**GAM\_MIX2\_AVS****Active video stop**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDS								Reserved	XDS							
----------	-----	--	--	--	--	--	--	--	----------	-----	--	--	--	--	--	--	--

Address: *CompositorBaseAddress + MIX2Offset + 0x2C*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the bottom-right corner of the active video rectangular area on the MIX output. Typically, it should be programmed according to the video standard currently in use.

[31:27] **Reserved**

[26:16] **YDS**: Vertical stop location of the MIX active video window, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDS**: Horizontal stop location of the MIX active video window, wrt VTG horizontal counter

**GAM\_MIX2\_ACT****MIX active content flags control**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	GDP2_ONVID_ACTIVE
	Reserved
	VID2_ONVID_ACTIVE
Reserved	GDP2_ONGFX_ACTIVE
	Reserved
	VID2_ONGFX_ACTIVE
Reserved	

Address: *CompositorBaseAddress + MIX2Offset + 0x38*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: Defines the layers that contribute to the construction of GFXActive2 and VideoActive2 flags. A layer is considered as active for a given pixel if its associated alpha component is not zero.

**XXX\_ONVID\_ACTIVE**: If set to 1, the XXX layer is taken into account for the VideoActive flag

**XXX\_ONGFX\_ACTIVE**: If set to 1, the XXX layer is taken into account for the GFXActive flag

Confidential

## GAM\_VIDn\_CTRL

## VID control

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GNORE_MX2	GNORE_MX1	Reserved			CFORM	709NOT601	Reserved			CKEY_CFG					Reserved	CKEY	AB_V	AB_H	Reserved							PSI_SAT_EN	PSI_TINT_EN	PSI_BC_EN			

Address: *CompositorBaseAddress + VIDnOffset + 0x00*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for controlling VID core. Color key and alpha border features are not available for a video layer displayed via MIX2.

[31] **IGNORE\_MX2**: When set, this bit indicates to MIX2 that the video data must be ignored when blending, even if enabled in MIX2\_CTRL. (But the mixer request is generated)

[30] **IGNORE\_MX1**: When set, this bit indicates to MIX1 that the video data must be ignored when blending, even if enabled in MIX1\_CTRL. (But the mixer request is generated)

[29:27] **Reserved**

[26] **CFORM**: Chroma format (0=offset 128, 1=signed)

[25] **709NOT601**: Colorimetry selection (0=601, 1=709)

[24:22] **Reserved**

[21:16] **CKEY\_CFG**: Configuration of color key

[21:20] **VID\_CTRL**[5:4]: Cr component

x0: Cr component ignored (disabled = always match)

01: Cr enabled: match if ( $Cr_{min} \leq Cr \leq Cr_{max}$ )

11: Cr enabled: match if ( $(Cr < Cr_{min})$  or  $(Cr > Cr_{max})$ )

[19:18] **VID\_CTRL**[3:2]: Y component

x0: Y component ignored (disabled = always match)

01: Y enabled: match if ( $Y_{min} \leq Y \leq Y_{max}$ )

11: Y enabled: match if ( $(Y < Y_{min})$  or  $(Y > Y_{max})$ )

[17:16] **VID\_CTRL**[1:0]: Cb component

x0: Cb component ignored (disabled = always match)

01: Cb enabled: match if ( $Cb_{min} \leq Cb \leq Cb_{max}$ )

11: Cb enabled: match if ( $(Cb < Cb_{min})$  or  $(Cb > Cb_{max})$ )

[15] **Reserved**

[14] **CKEY**: When this bit is set, the color key feature is enabled.

[13] **AB\_V**: Enable AlphaVBorder: When this bit is set, the alpha component on the first and last lines of the viewport is divided by 2 (edge smoothing)

[12] **AB\_H**: Enable AlphaHBorder: When this bit is set, the alpha component on the first and last columns of the viewport is divided by 2 (edge smoothing)

[11:3] **Reserved**

[2] **PSI\_BC\_EN**: Enable brightness and contrast correction with PSI block

[1] **PSI\_TINT\_EN**: Enable tint correction with PSI block

[0] **PSI\_SAT\_EN**: Enable chroma saturation with PSI block

**GAM\_VIDn\_ALP****Global alpha**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	GLOBAL_ALPHA_VAL
----------	------------------

Address: *CompositorBaseAddress* + *VIDnOffset* + 0x04

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the value for the VID global alpha.

[31:8] **Reserved**

[7:0] **GLOBAL\_ALPHA\_VAL**: Global blending coefficient for the video layer  
The register range is 0 to 128. (0: fully transparent, 1:fully opaque)

**GAM\_VIDn\_VPO****Viewport offset**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDO	Reserved	XDO
----------	-----	----------	-----

Address: *CompositorBaseAddress* + *VIDnOffset* + 0x0C

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the top-left corner of the video viewport.

[31:27] **Reserved**

[26:16] **YDO**: Vertical start location of the VID viewport, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDO**: Horizontal start location of the VID viewport, wrt VTG horizontal counter

**GAM\_VIDn\_VPS****Viewport stop**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	YDS	Reserved	XDS
----------	-----	----------	-----

Address: *CompositorBaseAddress* + *VIDnOffset* + 0x10

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the X and Y coordinate values for defining the bottom-right corner of the video viewport.

[31:27] **Reserved**

[26:16] **YDS**: Vertical stop location of the VID viewport, wrt line frame-numbering

[15:12] **Reserved**

[11:0] **XDS**: Horizontal stop location of the VID viewport, wrt VTG horizontal counter

Confidential

**GAM\_VIDn\_KEY1****Lower limit of the color keying range**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CR_MIN								Y_MIN								CB_MIN							

Address: *CompositorBaseAddress + VIDnOffset + 0x28*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for the lower limit of the VID color keying range.

[31:24] **Reserved**

[23:16] **CR\_MIN**: Minimum value of Cr component

[15:8] **Y\_MIN**: Minimum value of Y component

[7:0] **CB\_MIN**: Minimum value of Cb component

**GAM\_VIDn\_KEY2****Upper limit of the color keying range**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CR_MAX[7:0]								Y_MAX[7:0]								CB_MAX[7:0]							

Address: *CompositorBaseAddress + VIDnOffset + 0x2C*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0

Description: This register contains the values for the upper limit of the VID color range.

[31:24] **Reserved**

[23:16] **CR\_MAX**: Maximum value of Cr component

[15:8] **Y\_MAX**: Maximum value of Y component

[7:0] **CB\_MAX**: Maximum value of Cb component

**GAM\_VIDn\_BC****Brightness and contrast**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CONTRAST								BRIGHTNESS							

Address: *CompositorBaseAddress + VIDnOffset + 0x70*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0x0000 8000

Description: Contains the values for the luminance gain and offset. Both values are coded on 8 bits and need to be adapted to the input luminance depth.

[31] **Reserved**

[15:8] **CONTRAST**: Adjust luminance dynamic range (unsigned value, 128 centered)

[7:0] **BRIGHTNESS**: Adjust luminance intensity (2's signed value)

Confidential

**GAM\_VIDn\_TINT****Tint (hue/chroma phase)**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																TINT						Reserved
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	----------

Address: *CompositorBaseAddress + VIDnOffset + 0x74*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0x0000 0000

Description: This register codes the chrominance phase shift. This value is coded on 6 signed bits, the effective rotation angle  $t$  in radians will be  $TINT \times 2^{-6}$ , This gives a resolution of  $1/64$  radian or 0.90 degrees (therefore, considering that 1 LSB~1degree is a reasonable approximation) and a maximum codable rotation of about  $\pm\pi/6$ .

[31:8] **Reserved**

[7:2] **TINT**: Chrominance phase shift angle value (2's signed value)

[1:0] **Reserved**

**GAM\_VIDn\_CSAT****Chroma saturation**

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved																SAT						Reserved
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	----------

Address: *CompositorBaseAddress + VIDnOffset + 0x78*

Type: R/W

Buffer: Double-buffered, update on VTG Vsync

Reset: 0x0000 0080

Description: This register contains the values for the chrominance gain (from 0 to 2). This value is coded on 6 bits, giving a gain step of 3% (from 0 to 1.97)

[31:8] **Reserved**

[7:2] **SAT**: Chrominance saturation value (unsigned, 32 centered)

[1:0] **Reserved**

Confidential

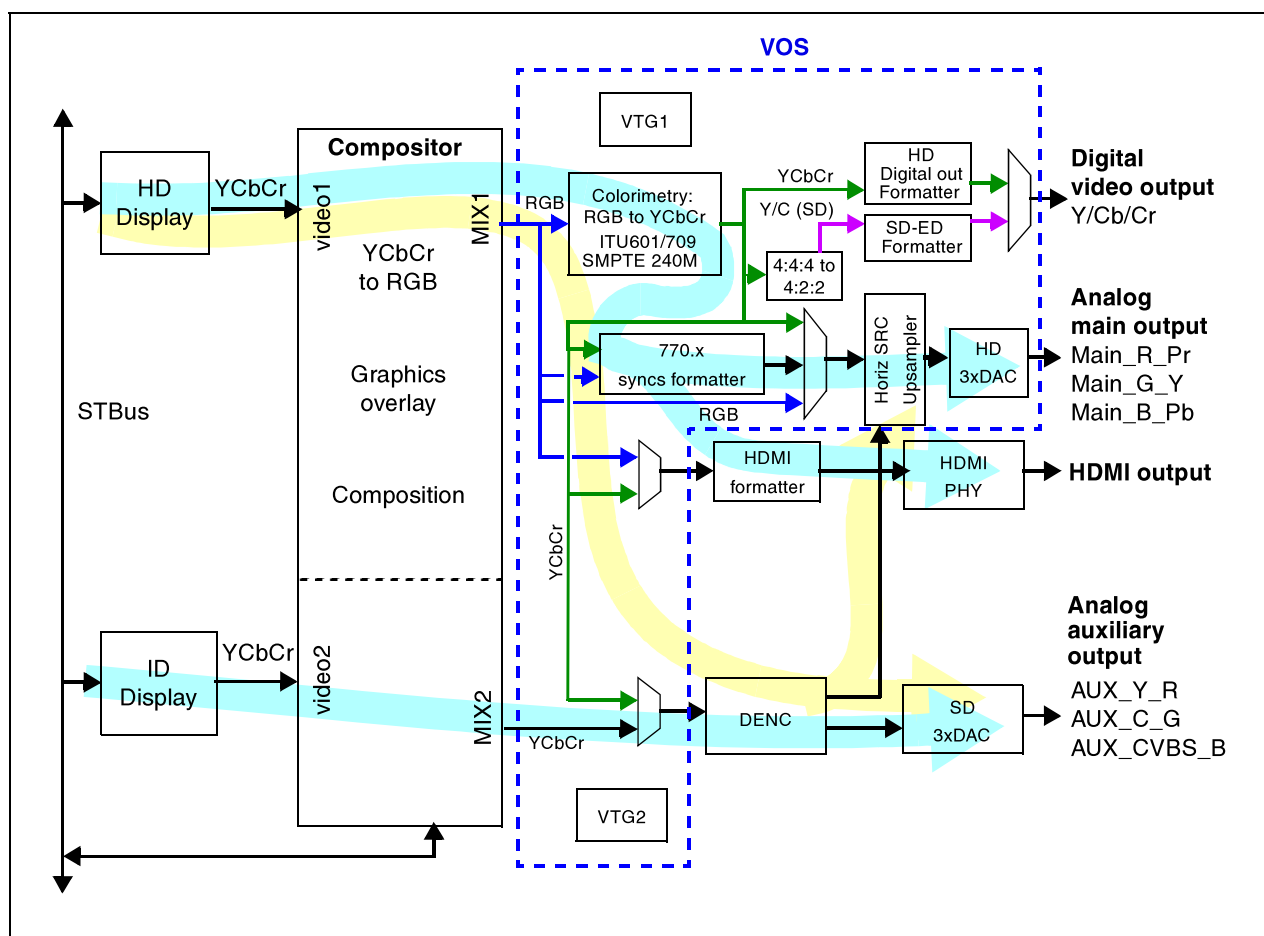
## 53 Video output stage (VOS)

The VOS receives video data from the main (mixer 1) and auxiliary (mixer 2) compositor outputs. This data is formatted and delivered to the DENC, HD and SD video DACs, HDMI formatter and digital video output (DVO).

### 53.1 Display and video output subsystem overview

Figure 183 shows the different units of the VOS.

Figure 183: Video output stage block diagram



#### 53.1.1 Standard application: HD on main output and SD on auxiliary output

In this configuration, the two main data paths of the VOS (main and auxiliary) are used independently to display an HD video on the main output and an SD video on the auxiliary output. These two data paths are shown in light blue in the above Figure 183. In this chapter, HD means any format other than 480i (SD interlaced).

##### Main data path (HD)

The main data path is paced by VTG1 and uses the compositor mixer 1. The HD display pipe reads video frame buffers from memory which are delivered to the compositor mixer 1. The compositor mixer overlays and blends the video with graphics, manages the windowing, and generates RGB components. The RGB signals are either used, or passed through an RGB-to-YCbCr matrix supporting ITU-R BT601, ITU-R BT709, or SMPTE 240M colorimetry. Data then goes to HDMI and/or analog output. The HDMI comprises a digital formatting and HDCP



encryption unit followed by a physical interface (HDMI PHY). In front of the analog output, a programmable formatter is provided allowing different output modes. The DACs are preceded by a sample-rate-converter (SRC) which upsamples the video signal to adapt the pixel rate to the video DAC sampling rate. This SRC has programmable coefficients. The video signal is interpolated using programmable set of coefficients designed to keep the maximum signal bandwidth. With a dedicated set of coefficients, a “constrained output” can be generated whereby the video bandwidth is cut down. For example, the analog video resolution can be degraded by 1/4, such that the full-resolution video is only available on the encrypted HDMI output.

### Main Analog output

This output delivers HD format up to 3H. The analog video can be full-scale RGB, full-scale YPbPr, EIA-770.2 compatible YPbPr with embedded bi-level syncs or EIA 770.3 compatible YPbPr with embedded tri-level syncs. The video signal is reduced by 70% to conform to the EIA 770.x standards.

The following display standards are available:

SMPTE 274M (1920 x 1080i, 1125 total lines per frame), SMPTE 293M (720 x 483p, 525 total lines), SMPTE 295M (1920 x 1080i, 1250 total lines), SMPTE 296M (1280 x 720p, 750 total lines), ARIB-BS4 (525p, 750p, 1125i total lines).

Macrovision encoding is supported on this output when configured for 525p output.

Full scale YPbPr or RGB component without embedded sync information can also be generated.

### Auxiliary analog output

This output delivers an SD interlaced decoded video, possibly with graphics overlay and ancillary data. For details, refer to [Chapter 51: Compositor on page 533](#) and [Chapter 55: Digital encoder \(DENC\) on page 644](#).

### HDMI output

This output supports 480p, 720p and 1080i formats as per HDMI specification.

### Digital Video (DVO) output

This digital video output is provided mainly for debug and diagnosis. The following modes are available:

- HD (1080i or 720p): YCbCr 16-bit 4:2:2 with embedded sync (Digital SMPTE 274M, SMPTE 295M),
- SD progressive (480p): 4:2:2 with embedded sync (SMPTE 293M).

### Auxiliary datapath (SD)

The auxiliary data path is paced by VTG2 and uses the compositor mixer 2. The ID display pipe reads video frame buffers from the memory which are delivered to the compositor mixer 2. The mixer overlays and blends the video with graphics, manages windowing, and generates YCbCr output components with CCIR 601 colorimetry. Those video components are then connected to the DENC, which generates Y, C and CVBS signals which are converted in analog with a triple SD DACs (auxiliary analog output).

The auxiliary analog video output signal is available either in composite (CVBS) or S-VHS (Y/C) format. The DENC supports the insertion of various VBI data on programmed lines, including close-caption and teletext.

### 53.1.2 Alternate application: SD on main and auxiliary outputs

An SD interlaced video can be delivered simultaneously to the HD video and SD video DACs. The data path used in this configuration is shown in yellow in [Figure 183](#).

The data path is paced by VTG1 and uses the compositor mixer 1. The HD display pipe reads video frame buffers from the memory and feeds the compositor mixer 1 which overlays and blends the video with graphics, manages windowing, and generates RGB components, all in SD format. In this mode, VTG2, ID Display pipe and compositor mixer 2 are not used. The RGB components are then passed through an RGB to YCbCr conversion matrix (601 colorimetry) and are connected to the DENC inputs. The DENC provides 2 sets of outputs to drive both the main and auxiliary analog video DACs. Between the DENC and the HD video DACs is the upsampler (SRC), which must be configured with the correct set of coefficients to perform a x4 interpolation.

#### Main analog output

This output delivers an analog video in SD format, YPbPr with embedded syncs.

#### Auxiliary analog output

This output delivers an analog SD video in composite CVBS or S-VHS components.

#### HDMI output

This output delivers an interlaced SD video (480i) with pixel repetition to comply with HDMI specifications.

#### Digital video output

This output delivers a digital SD video (480i), YCbCr 8-bit 4:2:2 with or without embedded syncs (ITU-R BT 656 or 601).

## 53.2 HD DACs clocking

The HD DACs clock frequency is programmable and is set to a value high-enough relative to the video spectrum to ease the design of the external analog reconstruction filter (anti-aliasing). The frequency is set to a multiple of the display clock frequency (x2 or x4), and a programmable interpolation filter is provided to upsample the DAC input.

The main master pixel clock (generated by the clock generator B) is set to one of the following values:

- 148.5 MHz: used in applications where the main output is 1080i or 720p at 30 or 60 Hz.
- 148.5/1.001: used in applications where the main output is 1080i or 720p at 29.97 Hz or 59.94 Hz.
- 108 MHz: used in application where the main output is 480p at 59.94 Hz. The display clock in that case is 27 MHz and an x2 upsampling must be used.
- 108 MHz: used in applications where both outputs are 480i at 29.97 Hz. The display clock in that case is 13.5 MHz and an x4 upsampling must be used.
- 108 x 1.001 (108.108 MHz): used in applications where the main output is 480p at 60 Hz.

The auxiliary master pixel clock (generated by the clock generator B) is always 27 MHz.

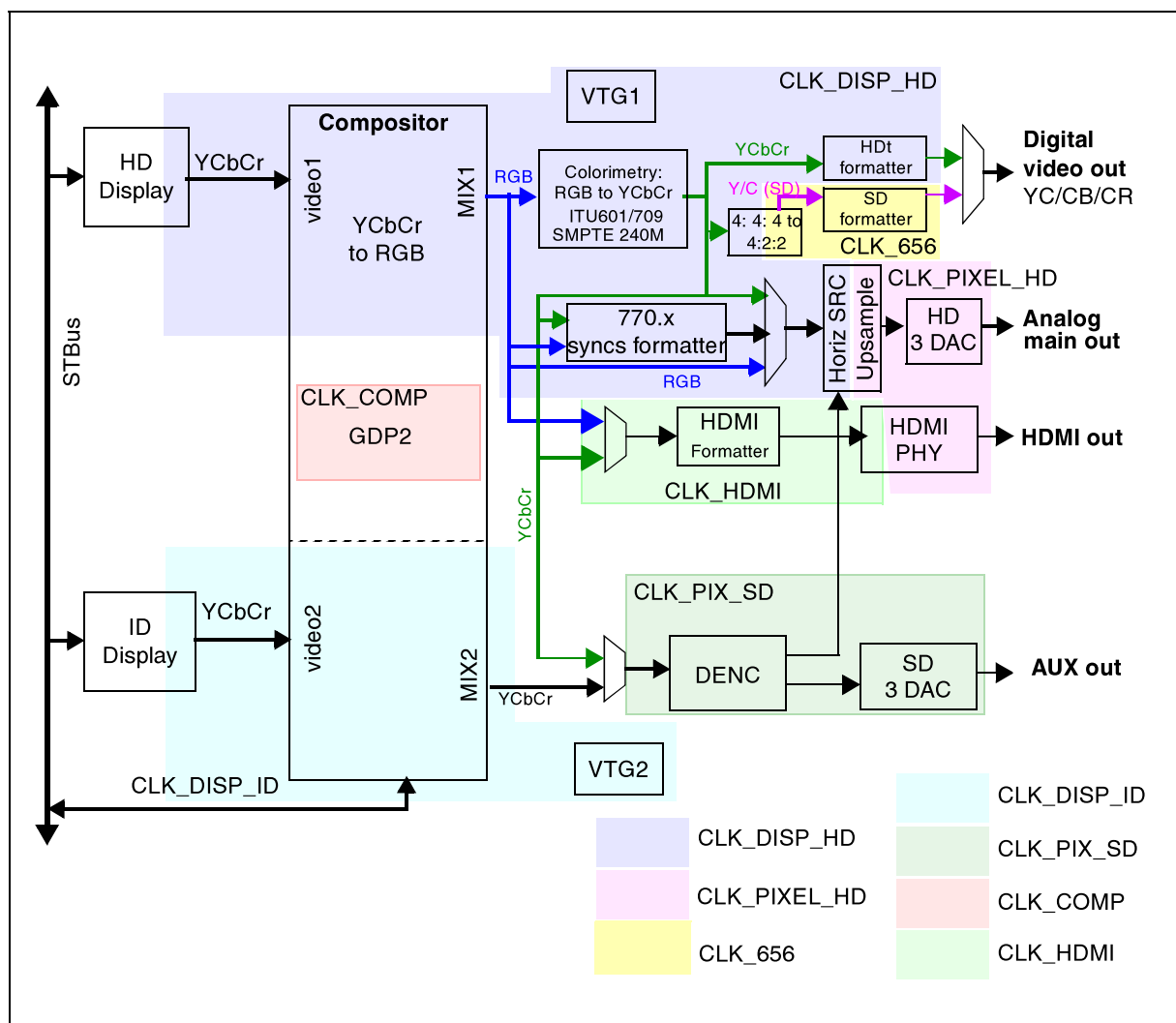
**Note:** *1080i and 720p are 30/60 Hz display standards; 480i and 480p are 29.97/59.94 Hz display standards. For some applications where it is preferable to keep the input rate for the display, the clock is altered by a factor of 1.001 to adapt the output rate by keeping the scanning. This is possible in 1080i, 720p and 480p formats but not in 480i format where the CVBS or C outputs, (chroma modulation) are very sensitive to clock rate and can not accept such a divergence.*

The VOS implements a number of clock domains to make possible different configurations:

- CLK\_PIX\_HD: main master pixel clock, clocking HD DACs and TMDS block (HDMI PHY),
- CLK\_DISP\_HD: pixel clock of main datapath (VTG1, mixer 1, HD display, formatters),
- CLK\_DISP\_ID: pixel clock of auxiliary path (VTG2, mixer 2, ID display),
- CLK\_COMP: GDP2 pixel clock, depends on GDP2 allocation,
- CLK\_PIX\_SD: 2X SD pixel clock (27 MHz) for DENC and SD DACs,
- CLK\_656: 2X pixel clock for CCIR 656/SMPTE 293M formatter,
- CLK\_HDMI: clock for the HDMI digital formatter.

For more details see [Chapter 15: Clocks](#) on page 120.

**Figure 184: VOS clock domains**



### 53.2.1 Applications

Table 175 summarizes some clock frequency values with respect to the display configurations. Related clocks (obtained by division of the same master clock) are shown with the same background color: magenta for main and blue for auxiliary.

Table 175: Clock domains by applications

Application		Clock frequencies, MHz					
		CLK_PIXEL_HD	CLK_DISP_HD	CLK_PIXEL_SD	CLK_DISP_SD	CLK_656	CLK_COMP
Main: 1080i@30 Hz or 720p@60 Hz	GDP2 on main	148.5	74.25	27	13.5	-	74.25
	GDP2 on aux	148.5	74.25	27	13.5	-	13.5
Main: 480p@59.94 Hz	GDP2 on main	108	27	27	13.5	54	27
	GDP2 on aux	108	27	27	13.5	54	13.5
Main: 1080i@29.97 Hz or 720p@59.94 Hz	GDP2 on main	~148.35	~74.176	27	13.5	-	~74.176
	GDP2 on aux	~148.35	~74.176	27	13.5	-	13.5
Main: 480p@60 Hz	GDP2 on main	108.108	27.027	27	13.5	54.054	27.027
	GDP2 on aux	108.108	27.027	27	13.5	54.054	13.5
Main: 480i@29.97 Hz	GDP2 on main	108	13.5	27	-	27	13.5

These clocks are generated by the clock generator B. For details, see [Chapter 15: Clocks on page 120](#).

## 53.3 Video timing generators (VTG)

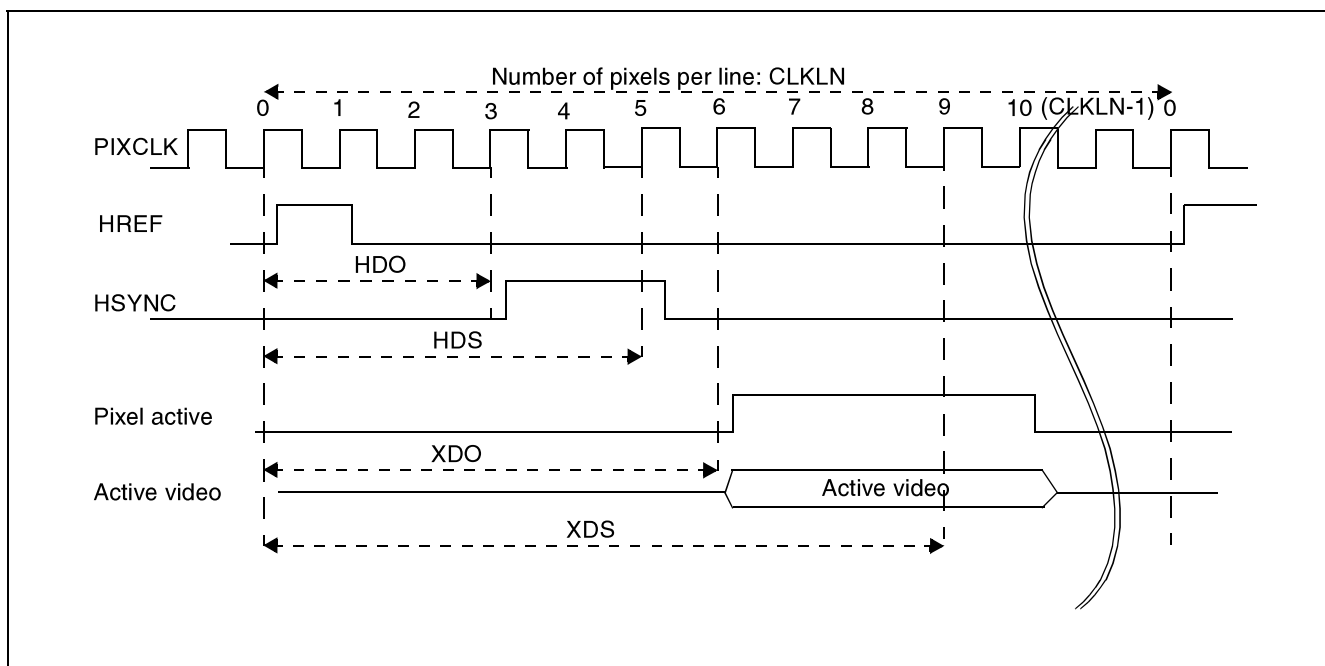
The STx7100 provides two video timing generators: VTG1 and VTG2.

The two VTGs provide horizontal and vertical video synchronization reference signals to the main and auxiliary data paths: in standard applications, VTG1 paces the main path and VTG2 paces the auxiliary path.

Each VTG provides a vertical temporal reference ( $V_{REF}$ ) and an horizontal temporal reference ( $H_{REF}$ ), that are used to generate VSYNC and HSYNC signals throughout the chip. In addition, the VTGs generate the VIDDIGOUTHSYNC and VIDDIGOUTVSYNC output signals. The shape and polarity of these signals are programmable through HDO/HDS and VDO/VDS registers

### 53.3.1 Horizontal synchronization generation

Figure 185: Horizontal synchronization generation



All the pixel numbers in a line are referenced to the rising edge of  $H_{REF}$ .

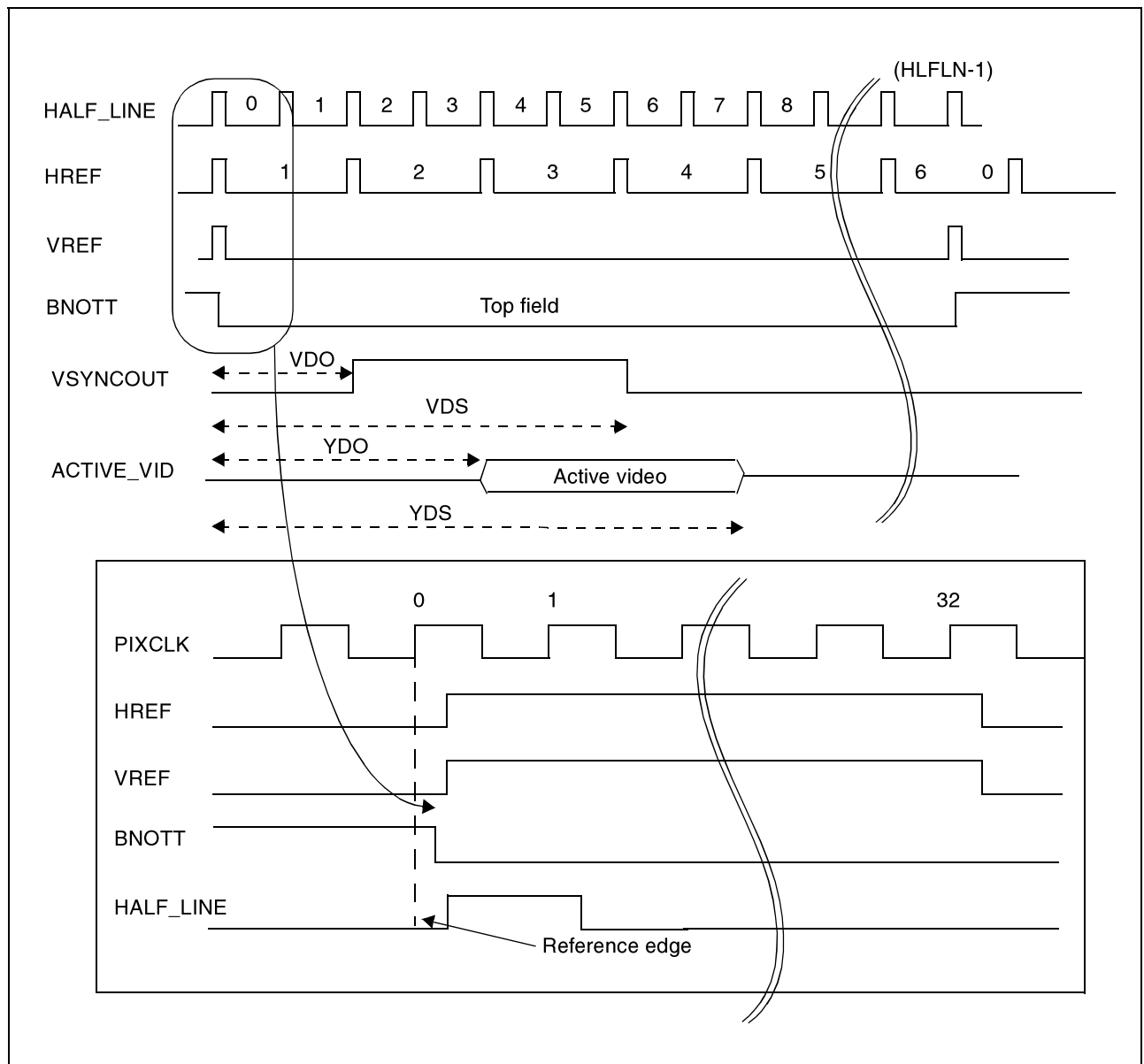
- CLKLN (VTGn\_CLKLN register) specifies the line length in PIXCLK cycles.
- HDO (register VTGn\_HDO) defines the number of PIXCLK cycles between the rising edge of the internal  $H_{REF}$  and the rising edge of  $H_{SYNC}$ .
- HDS (register VTGn\_HDS) defines the number of PIXCLK cycles between the rising edge of the internal  $H_{REF}$  and the falling edge of  $H_{SYNC}$ .
- Registers XDO and XDS, defining the active video line (horizontal dimension of the active video window) are located in the mixers of the compositor.

**Note:** If the HDS value is less than the HDO value, the result is the generation of an active low HSYNC pulse.

Confidential

## 53.3.2 Vertical synchronization generation (interlaced output)

Figure 186A: Vertical synchronization generation: (interlaced output)



The number of half lines per field HLFLN (or number of lines per picture) is specified in the VTGn\_HLFLN register. The half-line counter is incremented with a half-line resolution. It is reset when the counter matches the value programmed into the VTGn\_HLFLN register, generating VREF.

If the value programmed into the VTGn\_HLFLN register is even, a progressive scan display output is generated. If the value in HLFLN is odd, an interlaced output is generated. In both modes, a BNOTT (bottom not top) signal is also generated (in progressive output BNOTT is always 0). For an interlaced picture with an even number of lines (SMPTE 295M standard) a bit is programmed in a register to force the VTG to be in interlaced mode. In this case the picture has HLFLN - 1 half lines in one field and HLFLN + 1 in the other field.

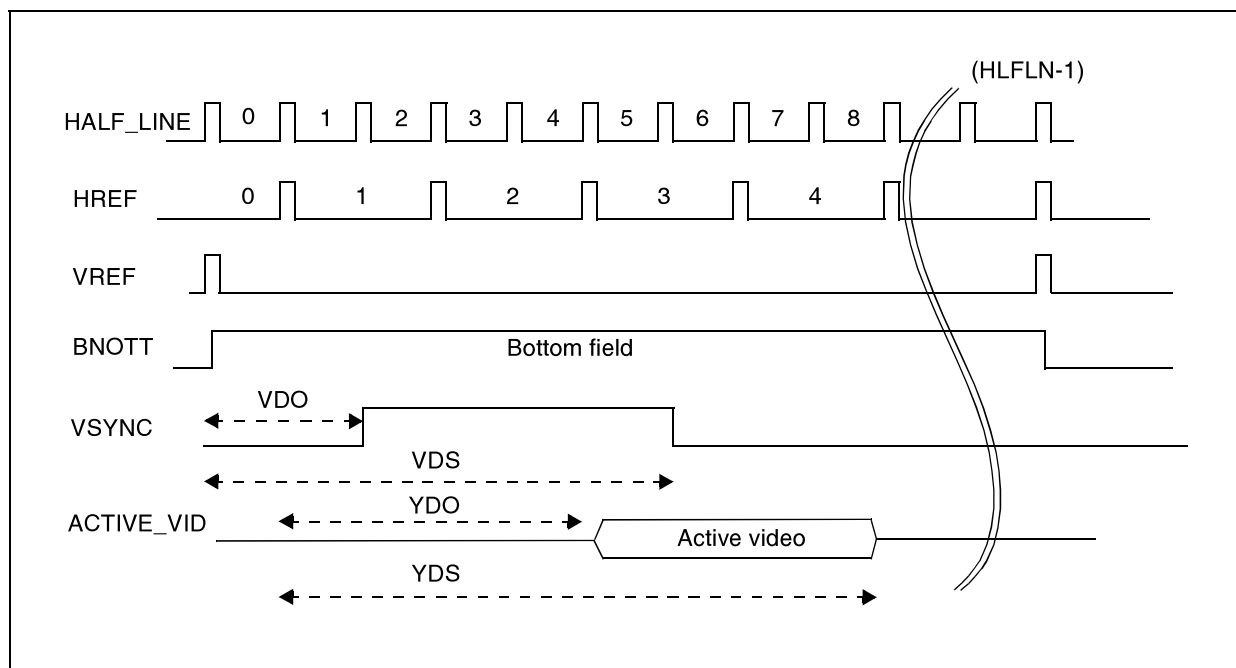
The line counter starts with 1 for VREF top and 0 for VREF bottom. It is then incremented by 1 every  $H_{REF}$ .

The registers YDO and YDS defining the vertical active window are located in the mixers of the compositor. YDO defines the first active line and YDS defines the last active line (in line units). The number of lines per field is  $(YDS - YDO + 1)$ .

The VDO register (VTGn\_VDO) determines the starting position of the vertical drive output pulse relative to VREF (half-line increment).

The VDS register (VTGn\_VDS) determines the ending position of the vertical drive output pulse relative to VREF (half-line increment).

**Figure 187: VTG Output (interlaced picture: bottom field)**



### Vsync positioning relative to HREF (available from cut 2.0 only)

The VHD field of the register VDS holds a 12-bit value that define the VSync position (in pixel clock cycles) relative to the HREF signal. With a 0 value, VSYNC rises one clock cycle after VREF. To get a VSYNC coincident to HSYNC, VHD must be set to HDO (or HDS) if HDO is in the first half of the line, or VHD must be set to HDO (or HDS) - CLKN/2 if HDO is in the second half of the line.

In the example of [Figure 187](#) above, YDO = 3, YDS = 4, VDO = 2 and VDS = 6.

Number of active video lines =  $(YDS - YDO + 1) = 2$

Length of the VSYNCOUT pulse =  $(VDS - VDO) = 4$  half lines.

**Note:** If  $HDO = 0$ , then HSYNCOUT and VSYNCOUT active edge are both generated on the same clock cycle on the line corresponding to VDO.

### 53.3.3 Interrupts

VST (Vsync top) and VSB (Vsync bottom) can be generated by the VTG. These interrupts are available through the ITS (interrupt status) register if the corresponding ITM (interrupt mask) bit is set. In progressive scan, the VTG still generates VST and VSB even though they are not necessary.

A timer is provided to generate a delayed interrupt PDVS (programmable delay on vsync)  $n$  cycles after the rising edge of vsync. The number of cycles is a 24-bit value defined in the register VTGn\_VTMR (vsync delayed interrupt timer).

### 53.3.4 VTG modes

The following synchronization modes are supported by the STx7100 VTGs:

- **VTG1** (Main - HD or SD output): VREF1 and HREF1 are always free running and generated from internal counters.
- **VTG2** (Auxiliary - SD output): the following configurations are possible:
  - VREF2 and HREF2 are free running based on internal counters
  - VREF2 and HREF2 slaved to VREF1 and HREF1 from VTG1. This is to slave the auxiliary video to the main video
  - VREF2 slaved to VREF1 from VTG1, HREF2 based on internal counter. This mode enables to slave VTG2 on a frame basis to a source with a different scanning format.

Figure 188 and Figure 189 show HREF and VREF generation for VTGs 1 and 2 respectively.

Figure 188: VTG1 HREF and VREF generation

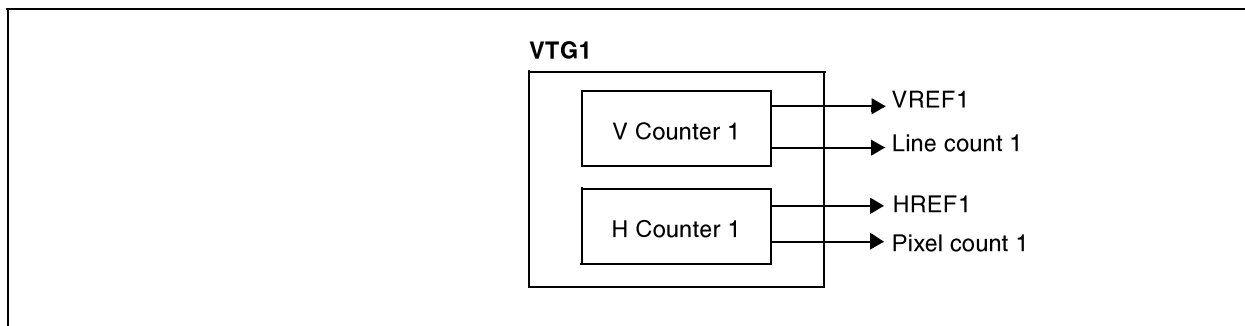
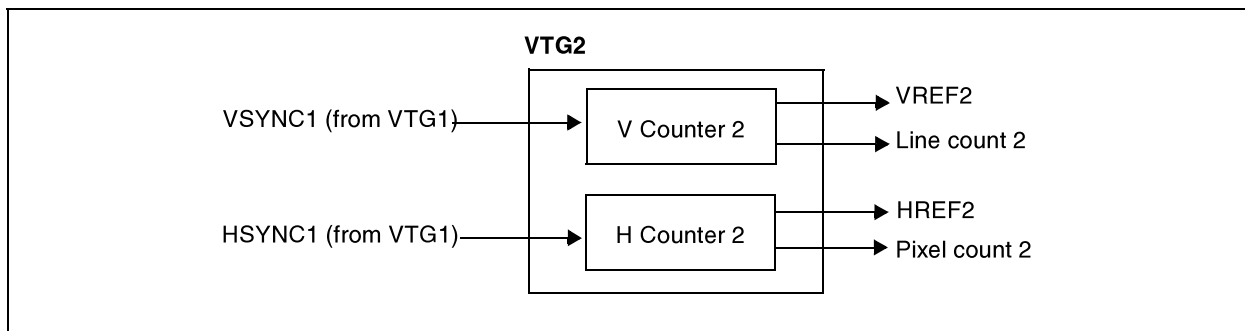


Figure 189: VTG2 HREF and VREF generation



### 53.3.5 Syncs generation for DVO, analog, HDMI and waveform generator

The STx7100 VTGs are able to generate several pairs of Hsync and Vsync signals with different parameters to be used by different units.

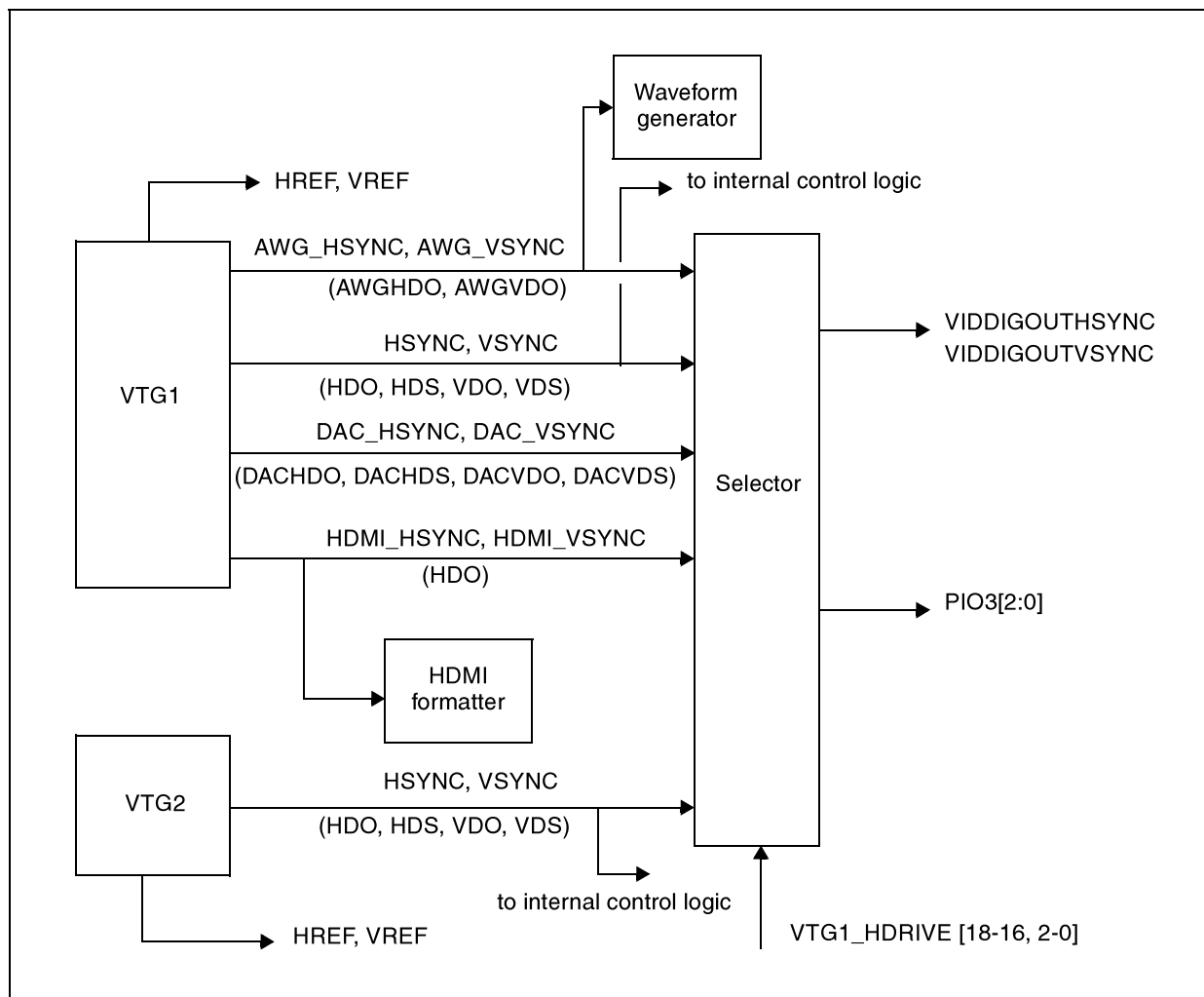
These Hsync and Vsync signals are:

- DVO Hsync and Vsync defined by VTGn\_VDO, VTGn\_VDS, VTGn\_HDO and VTGn\_HDS registers.
- Analog Hsync and Vsync that can be used with RGB analog outputs and defined by VTG1\_DACVDO, VTG1\_DACVDS, VTG1\_DACHDO and VTG1\_DACHDS registers.
- Waveform generator Hsync and Vsync defined by VTG1\_AWGVDO and VTG1\_AWGHDO registers.
- HDMI Hsync and Vsync which can be delayed relative to Hsync and Vsync. The delay is specified in the register VTG1\_HDO.

Any of these signals can be routed to the output pins VIDDIGOUTHSYNC, VIDDIGOUTVSYNC, or PIO3(2:0). The selection is done by configuring the register VTG1\_HDRIVE.



Figure 190: Hsync and Vsync generation



Confidential

### 53.3.6 Synchronization with external video

When receiving an external digital video via the DVP (D1/ITU-R656 4:2:2), the incoming data is sampled with the incoming video clock supplied externally and not controlled by the IC. To synchronize the DVP input to the VTG, the frame buffers fullness must be monitored and frame skip or repeat operation must be performed.

## 53.4 RGB to YCrCb color space conversion

RGB values received by the video output stage are gamma-corrected values (sometimes called R'G'B'). It is assumed that all the video signals connected to the chip are already gamma-corrected. Therefore, the following equations do not perform any gamma correction. If a gamma correction is needed (for non-precorrected graphics downloaded from the web for example), the gamma correction must be done with the 2D graphic processor (see [Chapter 47: 2D graphics processor \(blitter\)](#) on page 428).

Three equations are implemented:

- RGB to ITU-R BT 601 YCbCr,
- RGB to ITU-R BT 709 YCbCr,
- RGB to SMPTE 240M YCbCr.

The selection of the conversion matrix is possible with through configuration bits.

### 53.5 Digital video output formatter

The digital output formatter inserts proper synchronization words (EAV/SAV) into the video data flow according to selected standards and can produce:

- YCbCr, 4:2:2 on 16 bits with embedded sync: digital SMPTE 274M, SMPTE 295M, SMPTE 293M,
- YCbCr, 4:2:2 on 8 bits (8 MSBs of the DVO output) with or without embedded EAV/SAV sync: ITU-R 656.

### 53.6 Analog video output

The analog video output can be either:

- full-scale RGB output
- full-scale YPbPr output
- EIA-770.2 compatible YPbPr with embedded bi-level sync
- EIA 770.3 compatible YPbPr with embedded tri-level sync.

The pure video signal is reduced by 70% to conform to the EIA 770.x standards.

The YPbPr analog output handles the following raster-scanning systems:

**Table 176: YPbPr analog output raster-scanning systems**

Active picture	Frame	maximum PIXCLK frequency, MHz
1920 x 1080i	2200 x 1125i	74.25
1280 x 720p	1650 x 750p	
1920 x 1080i	2200 x 1250i	
720 x 483p	858 x 525p	27

The VOS configuration registers can be programmed to generate video complying with the following display standards: SMPTE 274M (1125i), SMPTE 293M (525p), SMPTE 295M (1250i), SMPTE 296M (750p) and ARIB-BS4 (525p, 750p, 1125i).

#### 53.6.1 Analog syncs generation (bi-level and tri-level)

To generate analog waveforms compliant with standards, a 770.3 and 770.2 formatter rescales the video data and inserts synchronization pulses before digital to analog conversion.

The analog syncs waveform (bi-level and tri-level) can be defined either using a flexible syncs generator or using pre-defined waveforms. This option is defined by the bit 3 of the register DSPCFG\_ANA. The synchronization pulses waveforms are defined with the configuration registers AWG\_CTRL\_0 and AWG\_RAM[0 .. 45].

In 480p, the formatter is able to apply Macrovision protection to the video flow if required.

The programmable configuration parameters allow flexibility for slight deviations from these standards.

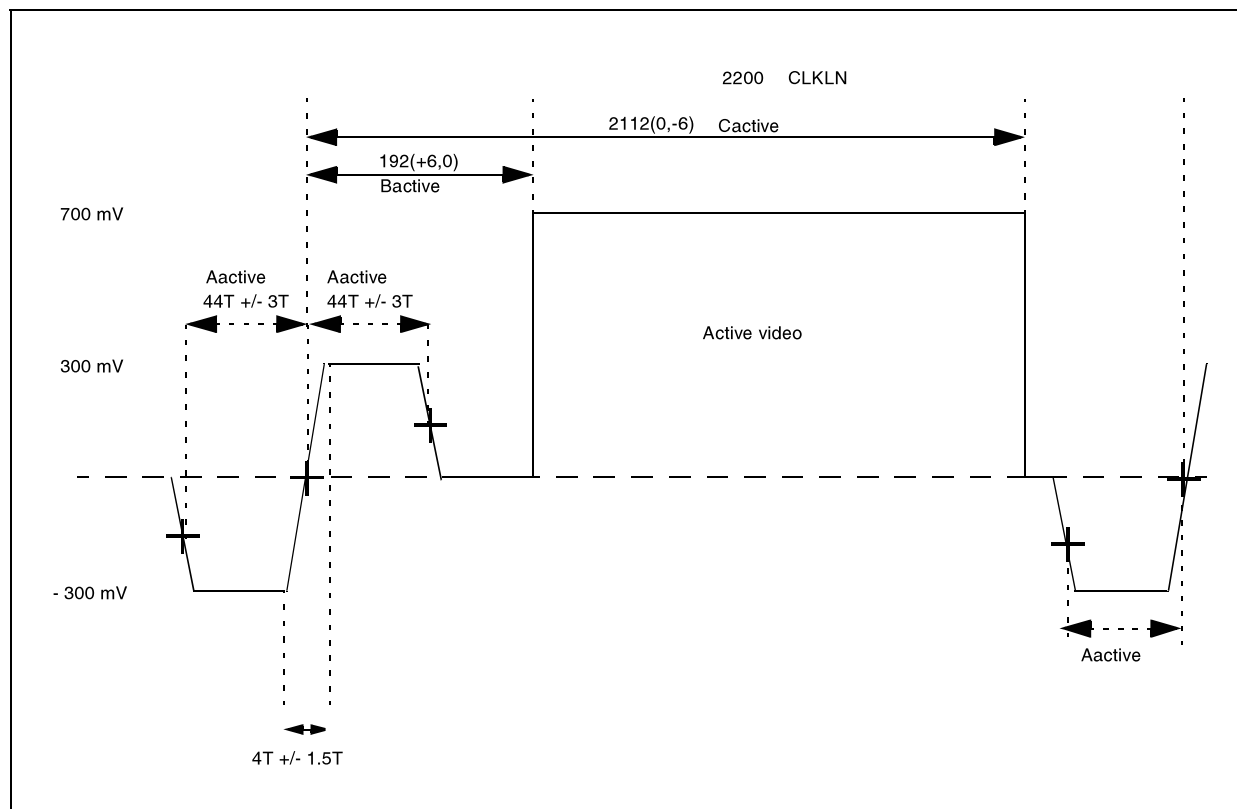
## 53.6.2 YPbPr waveforms - HD formats

### 53.6.2.1 Horizontal timing

#### Analog waveform (1080i)

Figure 191 describes the Y' analog waveform.

Figure 191: Y' analog waveform



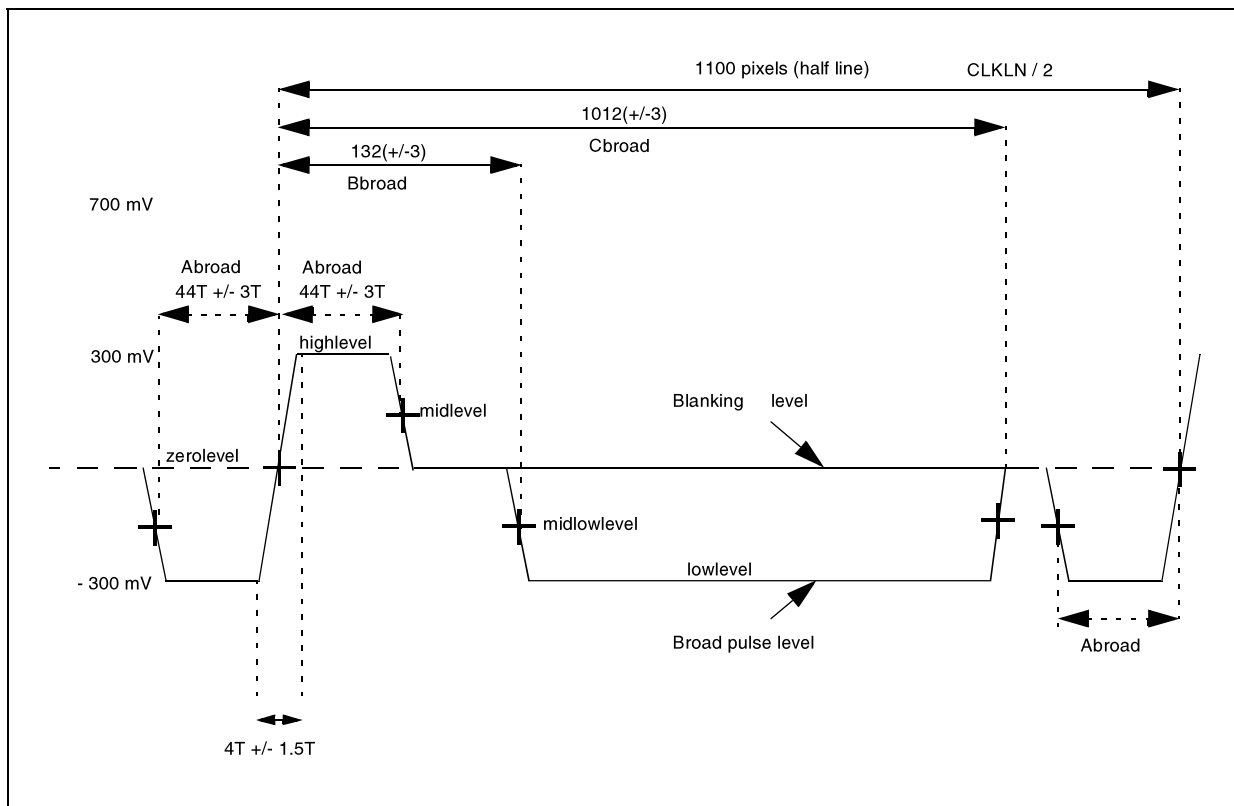
The Pb' and Pr' waveforms are similar, except for the voltage excursion ( $\pm 350$  mV).

Several raster scanning systems can be accommodated by programming *Aactive*, *Bactive*, *Cactive* and *CLKN*.

### Extra-sync pulse

With interlaced pictures, the vertical sync line may include a mid-line tri-level sync pulse. Certain vertical sync lines may therefore contain a broad pulse during the first half line and a broad pulse during the second half line. This is shown in Figure 192.

Figure 192: Vertical timing relating to analog waveform



Values *Abroad*, *Bbroad* and *Cbroad* are programmable to handle several raster scanning systems.

*Aactive* and *Abroad* have the same value which is programmed in register [DHDO\\_ABROAD](#). The blanking waveform (waveform without the broad pulse level) only requires the values *Abroad* and *CLKLN* to be programmed.

The synchronization level is also programmable (five registers for luma and five for chroma: *ZEROLEVEL*, *MIDLEVEL*, *HIGHLEVEL*, *MIDLOWLEVEL* and *LOWLEVEL*).