PT12

3D Motion Adaptive Video Noise Reducer

User Manual

Revision 0.4
2nd March 2019

Revisions

Date	Revisions	Version
08-10-2011	First draft.	0.1
02-08-2015	960H and 1280H operation added.	0.2
16-05-2016	Manual mode added. HD operation added. Improved technical description. Register functions updated.	0.3
02-03-2019	Document reformatted.	0.4

Contents

Tables

Figures

Introduction

PT12 is a motion adaptive recursive noise reducer IP core for video and imaging applications.

PT12 accepts video from SD (525/625 NTSC/PAL) or HD video sources (the IP core has been tested up to 1080p/50-60Hz operation).

The core accepts separate Y (luma) and Cb/Cr (chroma) which it recursively filters using an external frame delay memory as the delay element.

To prevent blurring of moving objects, motion is detected by differencing the Y channel across the frame delay. The degree of noise reduction is thereby changed according to motion detected in the image thereby avoiding blur.

PT12 is especially helpful where it precedes an MPEG encoder where residual noise in the image is detected as motion by the encoder and therefore uses up valuable bandwidth. Other applications include pre-processing for large screen displays, image sensor noise reduction or scientific and industrial imaging. For the latter two applications the motion adaption may be turned off for greater levels of noise reduction.

Control and status registers are written to and read from using a conventional 8-bit wide microprocessor interface.

The intellectual property block is provided as RTL compliant Verilog source code for all FPGAs and ASICs. Typical resource usage for an Altera FPGA is shown in Table 1.

Logic Elements	Memory Bits	M9K blocks	9x9 Multipliers	18x18 multipliers
682	2722	4	0	0

Table 1 PT12 Altera FPGA resource requirements

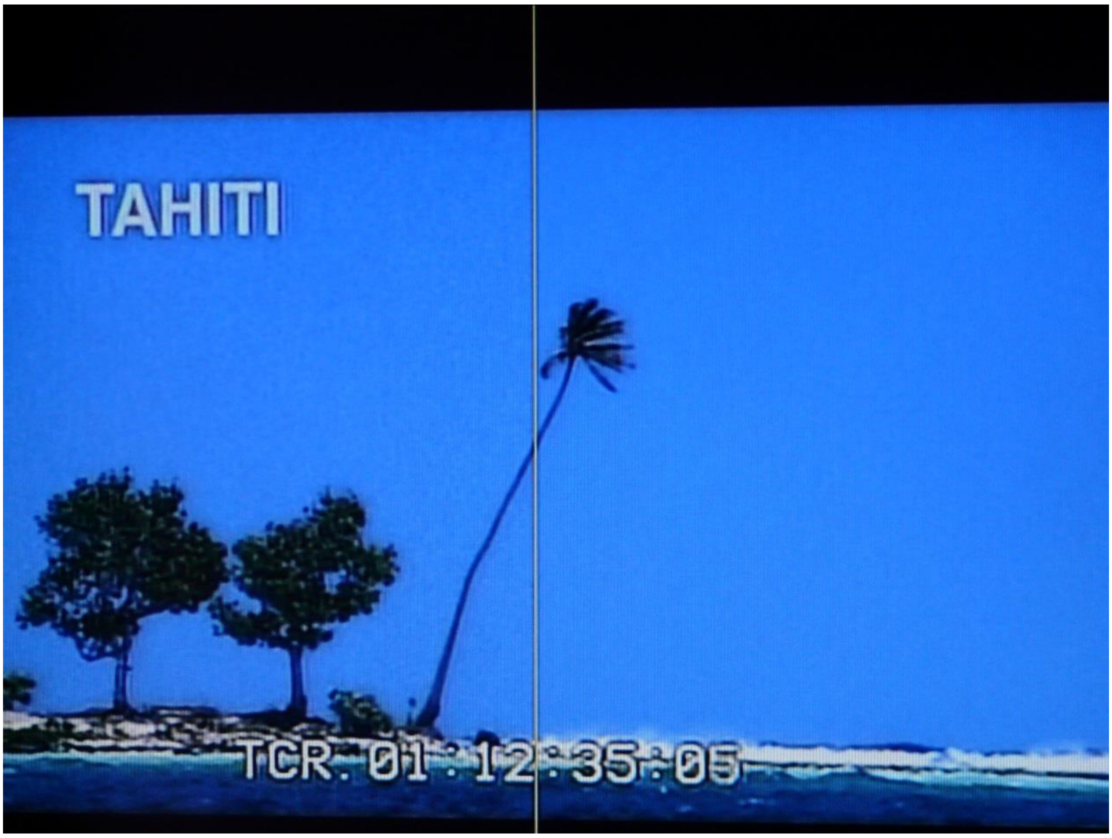

Figure 1 PT12 noise reduction (left side image is NR off; right side image is medium noise reduction)

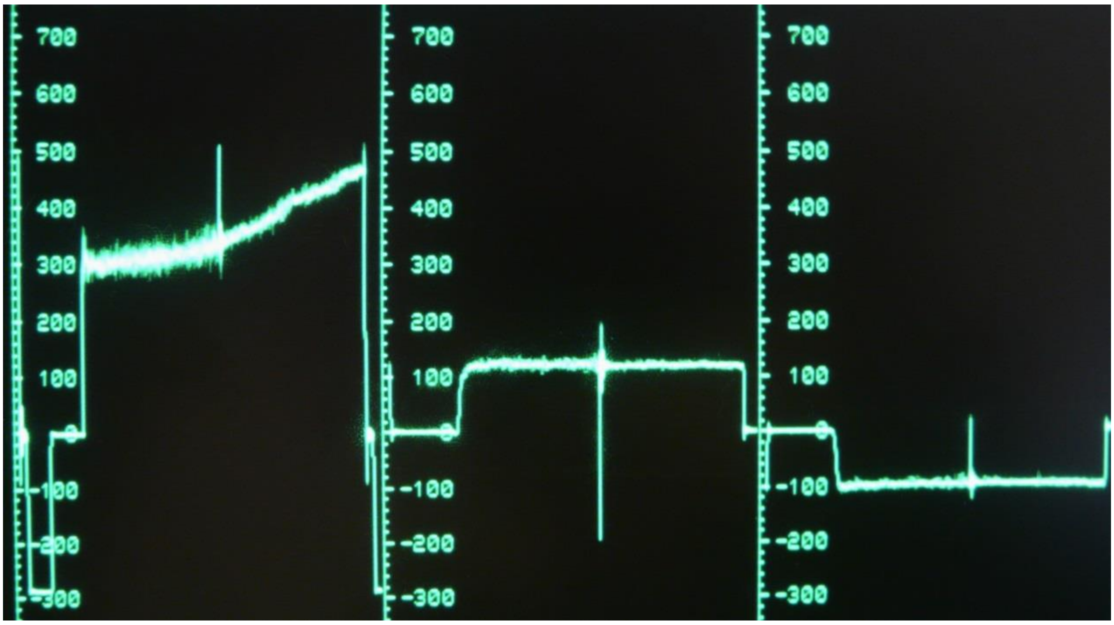

Figure 2 PT12 noise reduction (YPbPr parade waveform - left side of trace is NR off; right side of trace is medium noise reduction)

PT12 Module description

The PT12 video noise reducer IP core comprises 4 Verilog modules:

PT12.v
PT12_Register_control.v
MA_Y_ROM.v
MA_C_ROM.v

PT12.v is the top-level module of the hierarchy and the other modules are instantiated from that.

Signal Interconnections

The PT12 signal interconnect diagram is shown in Figure 3.

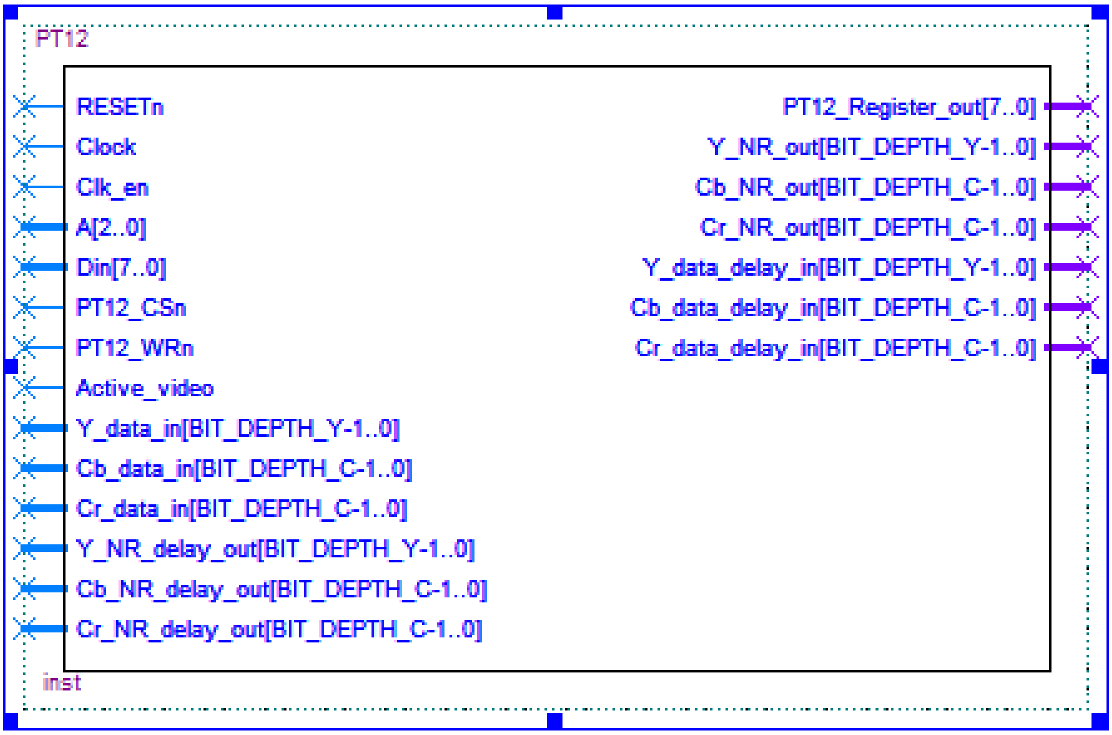
PT12
RESETn
Clock
Clk_en
A[2..0]
Din[7..0]
PT12_CSn
PT12_WRn
Active_video
Y_data_in[BIT_DEPTH_Y-1..0]
Cb_data_in[BIT_DEPTH_C-1..0]
Cr_data_in[BIT_DEPTH_C-1..0]
Y_NR_delay_out[BIT_DEPTH_Y-1..0]
Cb_NR_delay_out[BIT_DEPTH_C-1..0]
Cr_NR_delay_out[BIT_DEPTH_C-1..0]
PT12_Register_out[7..0]
Y_NR_out[BIT_DEPTH_Y-1..0]
Cb_NR_out[BIT_DEPTH_C-1..0]
Cr_NR_out[BIT_DEPTH_C-1..0]
Y_data_delay_in[BIT_DEPTH_Y-1..0]
Cb_data_delay_in[BIT_DEPTH_C-1..0]
Cr_data_delay_in[BIT_DEPTH_C-1..0]
inst

Figure 3 PT12 Block symbol.

The signal descriptions are shown in Table 2, below.

Inputs	
Signal	**Description**
RESETn	Asynchronous active low reset signal. Asserting this input sets all the control registers to their default value and resets all registers.
Clock	The pixel clock input. The rising edge of the clock is used for the registers. Typically this clock would be 27MHz for SD video, 74.25MHz for 720p/1080p and 148.5MHz for 1080p/50-60Hz. All data inputs should be valid at the rising edge of this clock and all outputs are valid at the rising edge of this clock.
Clk_en	Clock enable input. All register functions are enabled at the rising edge of Clock if Clk_en is valid (logic ‘1’). If Clk_en is ‘0’ no operation is performed. Typically this input might be enabled at 13.5MHz for SD if the clock is 27MHz. If not used this input should be tied to ‘1’.
A[2:0]	Control address bus input used to select the control register to be written to/read from.
Din[7:0]	Control data input bus.
PT12_CSn	Control chip select input, active low. Used in combination with the WRn input to control writing to the control registers. (See Chapter 5.)
PT12_WRn	Active low write enable input. Used in combination with the

	PT12_CSn input to control writing to the control registers. (See Chapter 5).
Active_video	Should be high for the period of the active video (e.g. 720 pixels width if 525i/625i or 1920 pixels width if 1080p standards). This input is only used for the split screen control. If this feature is not used this input may be tied to '0'.
Y_data_in[BIT_DEPTH_Y-1:0]	Y (luma) input to the noise reduction. The data input should be valid at the rising edge of clock if Clk_en is valid (= '1'). The width of the Y input is set with the BIT_DEPTH_Y parameter and should be 8 bits or more; there is no maximum bit depth. The default data width is 10 bits. The input is assumed to be straight binary.
Cb_data_in[BIT_DEPTH_C-1:0]	Cb (chroma) input to the noise reduction. The data input should be valid at the rising edge of clock if Clk_en is valid (= '1'). The width of the Y input is set with the BIT_DEPTH_C parameter and should be 8 bits or more; there is no maximum bit depth. The default data width is 10 bits. The input is assumed to be offset binary (i.e. for 10-bit mode, the black level is value $200).
Cr_data_in[BIT_DEPTH_C-1:0]	Cr (luma) input to the noise reduction. The data input should be valid at the rising edge of clock if Clk_en is valid (= '1'). The width of the Y input is set with the BIT_DEPTH_C parameter and should be 8 bits or more; there is no maximum bit depth. The default data width is 10 bits. The input is assumed to be offset binary (i.e. for 10-bit mode, the black level is value $200).
Y_NR_delay_out[BIT_DEPTH_Y-1:0]	Y (luma) input from the frame delay memory. The data input should be valid at the rising edge of clock if Clk_en is valid (= '1'). The width of the Y input must be the same as the Y_data_in parameter. The default data width is 10 bits.
Cb_NR_delay_out[BIT_DEPTH_C-1:0]	Cb (chroma) input from the frame delay memory. The data input should be valid at the rising edge of clock if Clk_en is valid (= '1'). The width of the Cb input must be the same as the Cb_data_in parameter. The default data width is 10 bits.
Cr_NR_delay_out[BIT_DEPTH_C-1:0]	Cr (chroma) input from the frame delay memory. The data input should be valid at the rising edge of clock if Clk_en is valid (= '1'). The width of the Cr input must be the same as the Cr_data_in parameter. The default data width is 10 bits.
Outputs	
Signal	**Description**
PT12_Register_out[7:0]	Control output data bus. Outputs the control/status register data selected by the A[2:0] bus.
Y_NR_out[BIT_DEPTH_Y-1:0]	Y (luma) output from the noise reduction. The data output is valid at the rising edge of the clock if Clk_en is valid (= '1'). The width of the Y output is the same as the Y_data_in parameter. The default data width is 10 bits.
Cb_NR_out[BIT_DEPTH_C-1:0]	Cb (lchroma) output from the noise reduction. The data output is valid at the rising edge of the clock if Clk_en is valid (= '1'). The width of the Cb output is the same as the Cb_data_in parameter. The default data width is 10 bits.
Cr_NR_out[BIT_DEPTH_C-1:0]	Cr (lchroma) output from the noise reduction. The data output is valid at the rising edge of the clock if Clk_en is valid (= '1'). The width of the Cr output is the same as the Cr_data_in parameter. The default data width is 10 bits.

Y_data_delay_in[BIT_DEPTH_Y-1:0]	Y (luma) output to the frame delay memory. The data input is valid at the rising edge of the clock if Clk_en is valid (= ‘1’). The width of the Y input must be the same as the Y_data_in parameter. The default data width is 10 bits.
Cb_data_delay_in[BIT_DEPTH_C-1:0]	Cb (chroma) output to the frame delay memory. The data input is valid at the rising edge of the clock if Clk_en is valid (= ‘1’). The width of the Cb input must be the same as the Cb_data_in parameter. The default data width is 10 bits.
Cr_data_delay_in[BIT_DEPTH_C-1:0]	Cr (chroma) output to the frame delay memory. The data input is valid at the rising edge of the clock if Clk_en is valid (= ‘1’). The width of the Cr input must be the same as the Cr_data_in parameter. The default data width is 10 bits.

Table 2 Input/Output signals

The Verilog instantiation of PT12 is shown below:

```
PT12 PT12_inst
(
.RESETn(RESETn_sig) ,                                   // input  RESETn_sig
.Clock(Clock_sig) ,                                     // input  Clock_sig
.Clk_en(Clk_en_sig) ,                                   // input  Clk_en_sig
.A(A_sig) ,                                             // input [2:0] A_sig
.Din(Din_sig) ,                                         // input [7:0] Din_sig
.PT12_CSn(PT12_CSn_sig) ,                               // input  PT12_CSn_sig
.PT12_WRn(PT12_WRn_sig) ,                               // input  PT12_WRn_sig
.Active_video(Active_video_sig) ,                       // input  Active_video_sig
.Y_data_in(Y_data_in_sig) ,                             // input [BIT_DEPTH_Y-1:0] Y_data_in_sig
.Cb_data_in(Cb_data_in_sig) ,                           // input [BIT_DEPTH_C-1:0] Cb_data_in_sig
.Cr_data_in(Cr_data_in_sig) ,                           // input [BIT_DEPTH_C-1:0] Cr_data_in_sig
.Y_NR_delay_out(Y_NR_delay_out_sig) ,                   // input [BIT_DEPTH_Y-1:0] Y_NR_delay_out_sig
.Cb_NR_delay_out(Cb_NR_delay_out_sig) ,                 // input [BIT_DEPTH_C-1:0] Cb_NR_delay_out_sig
.Cr_NR_delay_out(Cr_NR_delay_out_sig) ,                 // input [BIT_DEPTH_C-1:0] Cr_NR_delay_out_sig

.PT12_Register_out(PT12_Register_out_sig) ,             // output [7:0] PT12_Register_out_sig
.Y_NR_out(Y_NR_out_sig) ,                               // output [BIT_DEPTH_Y-1:0] Y_NR_out_sig
.Cb_NR_out(Cb_NR_out_sig) ,                             // output [BIT_DEPTH_C-1:0] Cb_NR_out_sig
.Cr_NR_out(Cr_NR_out_sig) ,                             // output [BIT_DEPTH_C-1:0] Cr_NR_out_sig
.Y_data_delay_in(Y_data_delay_in_sig) ,                 // output [BIT_DEPTH_Y-1:0] Y_data_delay_in_sig
.Cb_data_delay_in(Cb_data_delay_in_sig) ,               // output [BIT_DEPTH_C-1:0] Cb_data_delay_in_sig
.Cr_data_delay_in(Cr_data_delay_in_sig)                 // output [BIT_DEPTH_C-1:0] Cr_data_delay_in_sig
);

defparam PT12_inst.BIT_DEPTH_Y = 10;
defparam PT12_inst.BIT_DEPTH_C = 10;
```

Technical Overview

A simplified block diagram of the PT12 noise reducer is shown in Figure 4.

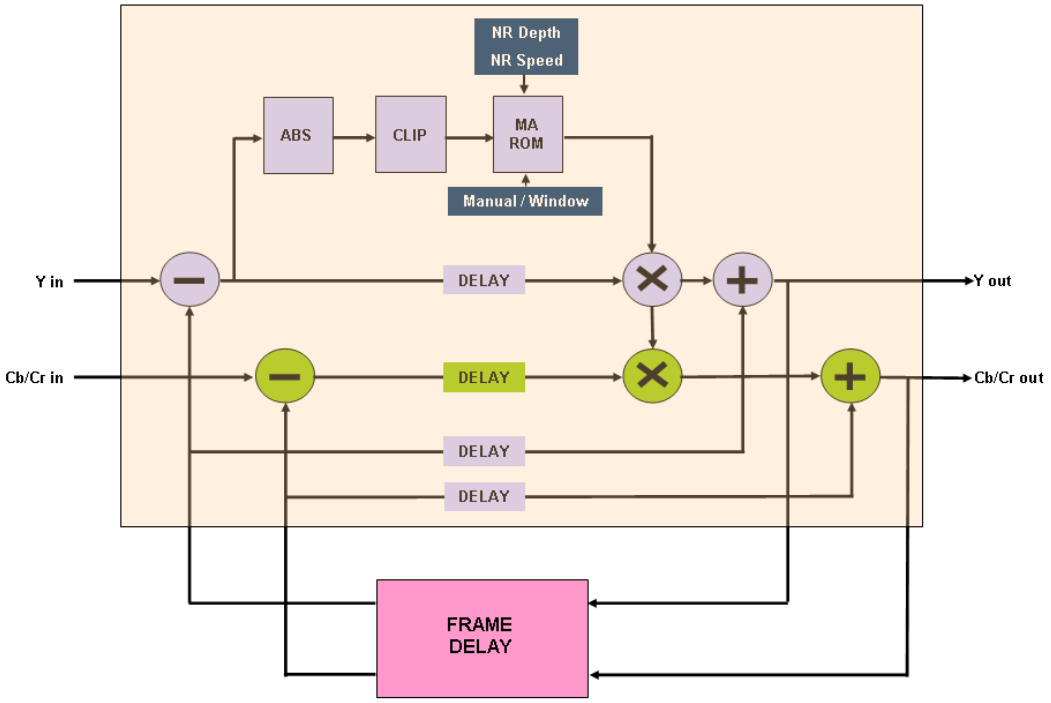

Figure 4 PT12 Block diagram

The video inputs to the PT12 are separate Y, Cb and Cr channels at programmable bit depths, (minimum of 8 bits, no maximum bit depth).

Each input is subtracted from the frame delayed data: (the frame delay, because of processing delays in the PT12 needs to be exactly 1 frame – 6 pixels. For example, for 525 line SD video at 13.5 MHz the frame needs to be 858 [pixels/line] x 525 [line/frame] = 450450 pixels – 6 = 450444 pixels. For HD operation e.g. 1080p/30Hz, the delay should be 2200 [pixels/line] x 1125 [line/frame] = 2475000 pixels – 6 = 2474994 pixels).

The frame delay memory, because of its size, is usually external to the FPGA/ASIC. It can be implemented using SD/DDR RAM or custom video FIFOs, such as the Averlogic AL460. If DDR memory already exists in the system, it can be shared with the PT12. The PT12 requires one read-before-write operation.

The subtracted data is then multiplied by the k factor before being added to the delayed data to form the noise reduced output (which also is the input to the frame delay).

This architecture is a rearrangement of the following equation:

$Y_{out} = k*Y_{in} + (1-k)*Y_{delay}$ where:

Y_{out} is the output luma (or Cb/Cr output)
Y_{in} is the input luma (or Cb/Cr input)

Y_{delay} is the output from the frame delay
k is the feedback factor.

The k factor sets the degree of noise reduction. However temporal noise reduction such as this leaves trails on moving objects. To prevent this the frame difference from the first subtractor (Y channel only) is used to detect motion and reduce the noise reduction where motion occurs.

The absolute value of the difference is calculated and this value is then clipped, (all differences above value 127 are clipped to 127). The clipped differences value is then used to address a lookup table along with two controls, the depth and the speed. These fixed controls (via control register 1) are used to set the degree of noise reduction (NR_Ydepth[1:0] and NR_Cdepth[1:0]) and the speed that the noise reducer responds to motion (NR_speed[2:0] - the level of the difference value that starts a reduction in the noise reduction. Two lookup tables are used, one for the Y value and one for the Cb/Cr values because the chroma trails are less visible to the eye so a higher degree of noise reduction may be applied to them.
The LUT adaptation values are shown graphically in Figures 5 and 6. The degree of noise reduction is shown on the vertical axis (value 1 is no noise reduction) and the horizontal axis if the value of the luma difference. So, for example, with no motion (difference value = 0) the k value is set to 0.3, 0.5 or 0.7 depending on the depth control register setting. As the difference value increases then. Depending on the NR_speed setting (either value 24, 32, 40, 48, 56, 60, 72 or 80) the k value is then adjusted according to the curves shown in the diagram. An NR_speed setting of ‘111’ (80) is the slowest to adapt to motion and the value ‘000’ (24) is the fastest to adapt to motion.

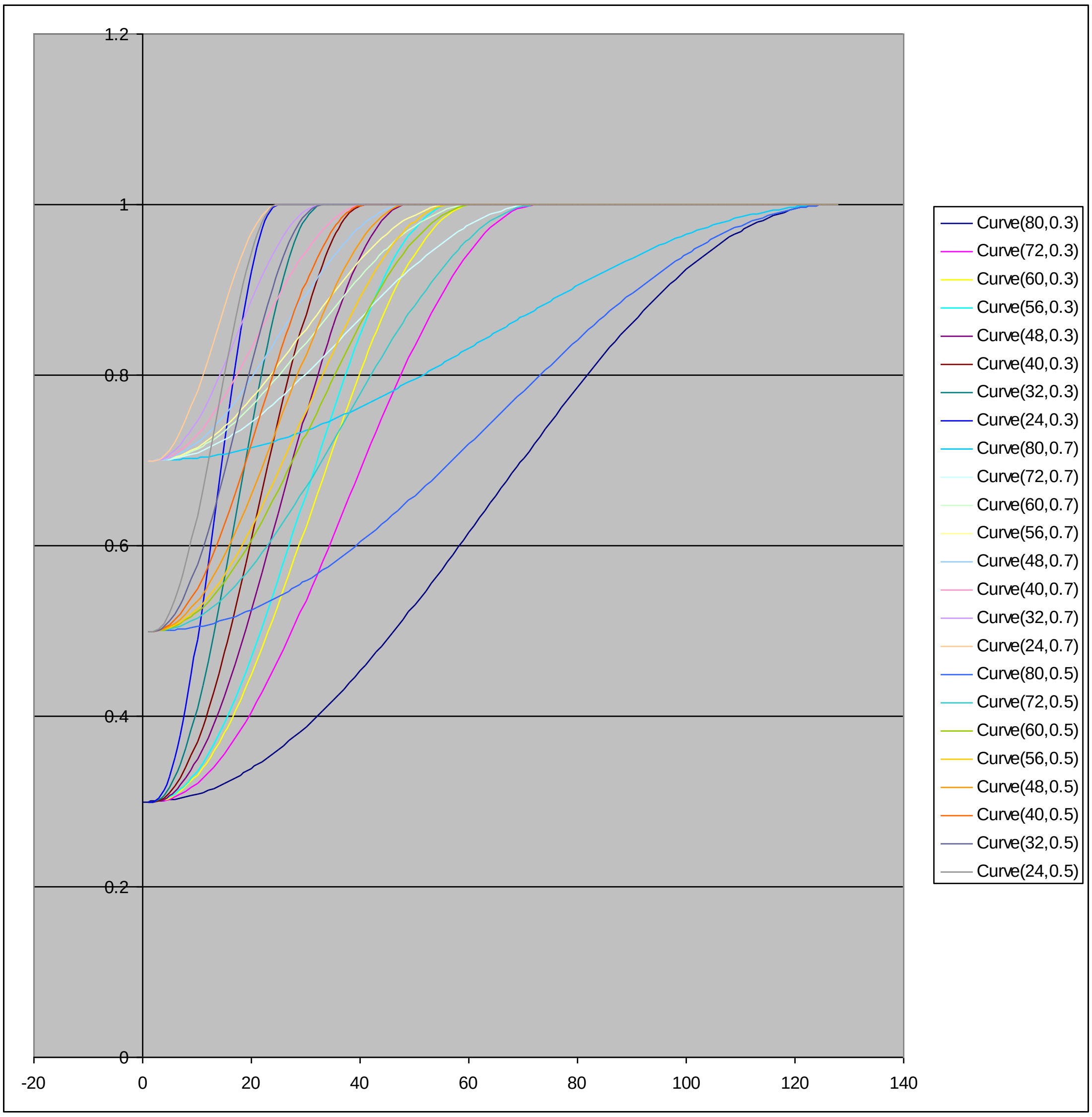

Figure 5 Y (Luma) channel noise reduction adaptation tables

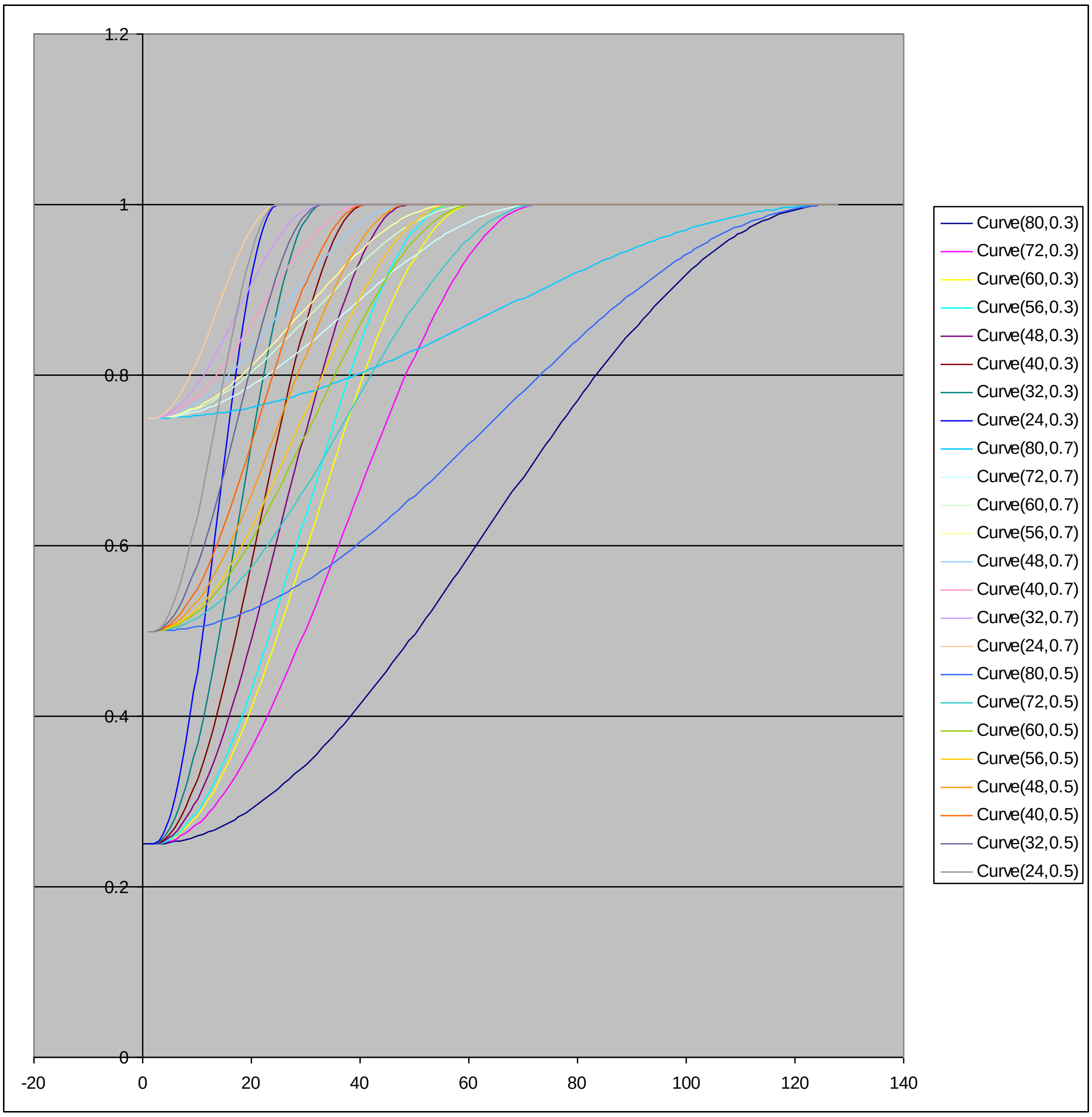

Figure 6 C (Chroma) noise reduction adaptation values.

For conditions where the noise is very high and the motion adaptation is not required a manual K factor may be programmed setting register $01 bits 0 (for Y) or bit 1 (for Cb/Cr) to ‘1’. The manual setting is then controlled with registers $02 and $03. A manual value of ‘0’ sets the noise reduction to off, and higher values increase the degree of noise reduction.

The expected degree of noise reduction (Gaussian noise) for static objects is 4.8dB (k=0.5), 8.5dB (k-0.25) and 11.8dB (k=0.125).

The ‘Split_enable’ register bit (register $01, bit 6) may be used to vertically split the screen between noise reduction off and the currently set value of noise reduction. The split position may be highlighted if register $01, bit 7 is set to a ‘1’. The split position may be adjusted relative to the ‘Active_video’ input using registers $04 and $05.

The PT12 is controlled using a simple 8 bit register interface which is described in the next section.

Register interface

Figure 7 shows the timing diagram for the register interface; it is a conventional microprocessor interface. Each register is selected via a 3-bit address bus. Writes to unused register locations are ignored.

To write to the selected register the PT12_CSn (chip select) input must be asserted low, the A[2:0] assigned to the required register address and the data for this register set up. The PT12_WRn input must then be driven low and high again: On the rising edge of this pulse the data is latched into the address selected. The PT12_CSn should then be returned high.

For the write to occur reliably the address (A[2:0]) and data (Din[7:0]) must be stable and valid during the low to high transition of the PT12_WRn pulse.

The address input also selects the register data that is presented on the PT12_Register_out[7:0] bus. This output is independent of the PT12_CSn or PT12_WRn inputs.

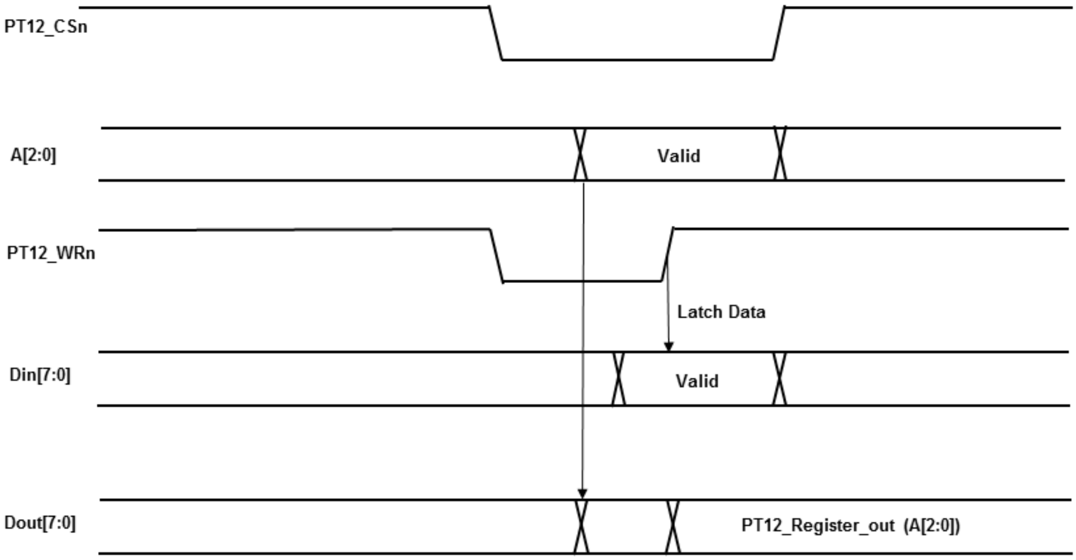

Figure 7 PT12 Register interface.

Register descriptions

Table 3 lists all of the control and status registers. All of the registers are 8-bit; unused register bits read back as zeros.

Please note that some registers can be set to values that are illegal and will produce invalid outputs.

Asserting the RESETn input sets the PT12 registers to their default values.

Register Offset	Register Name	R/W	Bit Value	Description	
$00	**Control 1**	R/W			
			7	Not used	
	NR speed		6:4	NR speed adaptation.	
				Bits [6:4]	Maximum difference value for NR adaptation
				000	24
				001	32
				010	40
				011	48
				100	56
				101	60
				110	72
				111	80
	NR C depth		3:2	NR C (luma) depth control	
				Bits [3:2]	C ‘k’ Factor
				00	1.0 (NR off)
				01	0.7
				10	0.5
				11	0.25 (maximum noise reduction)
	NR Y depth		1:0	NR Y (chroma) depth control	
				Bits [1:0]	Y ‘k’ Factor
				00	1.0 (NR off)
				01	0.75
				10	0.5
				11	0.3 (maximum noise reduction)
$01	**Control 2**	R/W			
	Highlight enable		7	If set to ‘1’ and ‘Split enable’ (bit 6) is also a ‘1’, the split position of the screen is highlighted with a light grey bar.	
	Split enable		6	If set to ‘1’ the noise reduction is disabled for the left side of the screen and enabled (at the level set by register $01) on the right side of the screen. This allows a comparative assessment of the noise reduction.	
			4:3	Not used	
			2	If ‘1’ enables manual control of k feedback factor (via register $02). Note that Register $01 must also be set value $00. If ‘0’ the PT12 motion adapts the k factor automatically.	
	Manual C noise reduction		1	If set to ‘1’ the C noise reduction is set to manual mode with no motion adaptation. The manual noise reduction value is set by register $03.	
	Manual Y noise reduction		0	If set to ‘1’ the Y noise reduction is set to manual mode with no motion adaptation. The manual noise reduction value is set by register $02.	
$02	**Manual Y k value**	R/W			
			7:0	Adjusts the manual value of the ‘k’ feedback factor (if register $01 bit 0 is set) for the Y channel. Value 0 is a ‘k’ factor of 1.0 (NR off) and value 255 is a ‘k’ factor of 0.004 (maximum noise reduction)	
$03	**Manual C k value**	R/W			
			7:0	Adjusts the manual value of the ‘k’ feedback factor (if register $01 bit 0 is set) for the Cb/Cr channels. Value 0 is a ‘k’ factor of 1.0 (NR off) and value 255 is a ‘k’ factor of 0.004 (maximum noise reduction)	
$04	**Split position (LSB)**	R/W	7:0	Adjusts the position of the split screen (Register $01, bit 6). The split	

Register Offset	Register Name	R/W	Bit Value	Description
$05	**Split position (MSB)**	R/W	3:0	position is a 12 bit word = ({Register$05[3:0],Register$04[7:0]). The value of the split is the number of pixels from the beginning of the active video control input. For example, for 525i or 625i (NTSC/PAL) operation, to position the split halfway across the screen the registers would be programmed with value 360_{10} (720 active pixels/line divide by 2). i.e. register $05 programmed with 1 (256) and register $04 programmed with 104_{10}.

Table 3 Register Descriptions