

# PT12

## 3D Motion Adaptive Video Noise Reducer

### User Manual

Revision 0.1  
8<sup>th</sup> October 2011

## Contents

Contents .....	2
Tables .....	2
Figures .....	2
Introduction .....	3
PT12 Module description .....	5
Signal Interconnections .....	6
Technical Overview .....	9
Register interface .....	12
Register descriptions .....	13

## Tables

Table 1 PT12 Altera FPGA resource requirements .....	3
Table 2 Input/Output signals .....	8
Table 3 Register Descriptions .....	13

## Figures

Figure 1 PT12 noise reduction (left side image is NR off; right side image is medium noise reduction) .....	4
Figure 2 PT12 noise reduction (YPbPr parade waveform - left side image is NR off; right side image is medium noise reduction) .....	4
Figure 3 PT12 Signal Interconnections .....	6
Figure 4 PT12 Block diagram .....	9
Figure 5 Y (Luma) channel noise reduction adaptation tables .....	10
Figure 6 C (Chroma) noise reduction adaptation values .....	11
Figure 7 PT12 Register interface .....	12

## Introduction

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

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

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

$Y_{out}$  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 (or  $Cb_{delay}/Cr_{delay}$ )

k is the feedback factor.

If  $k=1$  the output is a copy of the input. If  $k=0$  the output is the  $Y_{delay}$  value (i.e the output is frozen as the video just circulates through the memory). If  $k=0.5$  the output is half of the input + half of the delayed value. On a static image the output will therefore slowly approach the input value thereby reinforcing any static objects in the image but reducing the amplitude of random objects (e.g. noise).

However moving objects will be blurred by such a filter so the moving objects are detected by differencing the Y channel across the frame delay. The k value 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 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 an encrypted 'black box' design for incorporation into an Altera FPGA, as a netlist for Xilinx FPGAs or as RTL compliant Verilog source code for FPGAs from other vendors or for ASICs. Typical resource usage for an Altera FPGA is shown in Table 1.

Logic Elements	Memory Bits	M9K blocks	9x9 Multipliers	18x18 multipliers
232	81982	11	1	3

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)

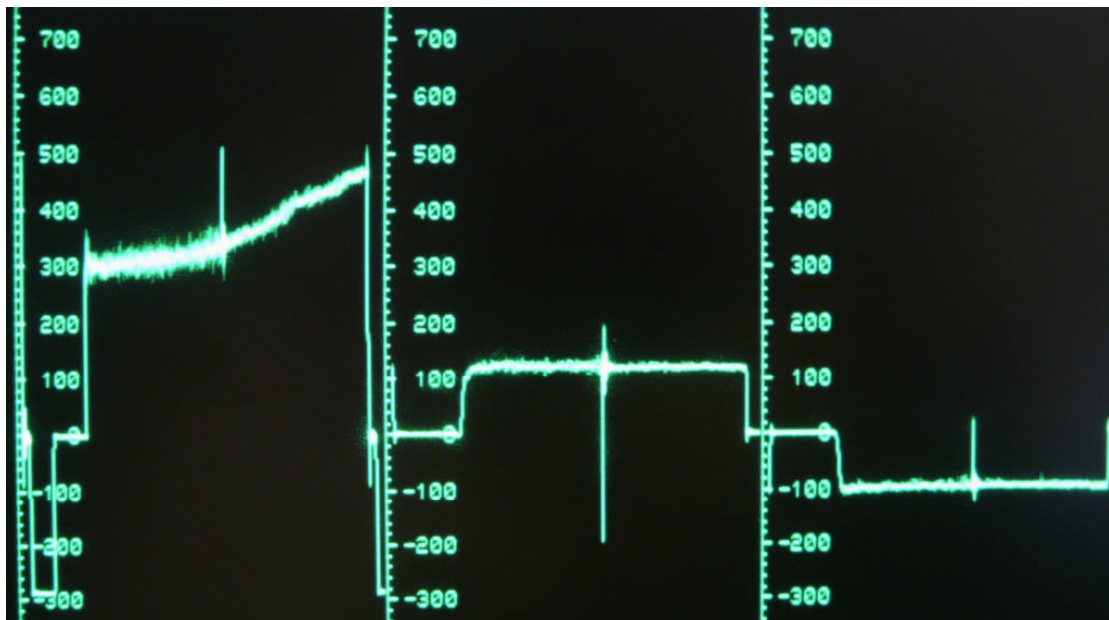


Figure 2 PT12 noise reduction (YPbPr parade waveform - left side image is NR off; right side image is medium noise reduction)

## PT12 Module description

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

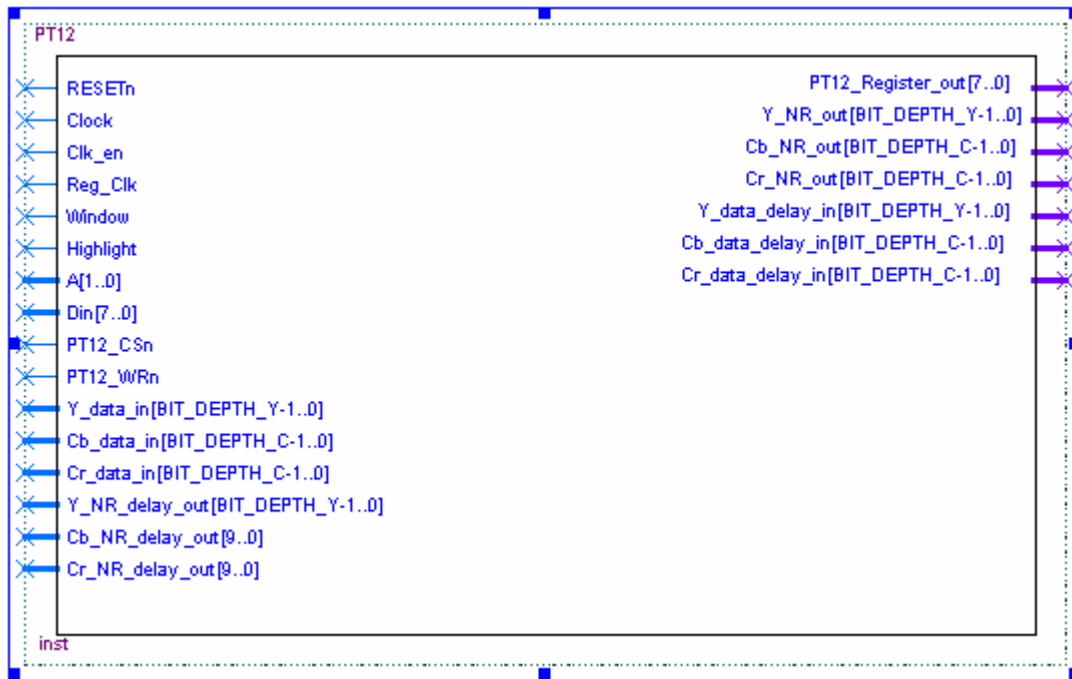
PT12.v  
PT12\_Register\_control.v  
MA\_Y\_ROM.v  
MA\_C\_ROM

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

To save on the memory footprint it is possible to remove the MA\_C\_ROM.v module and re-use the MA\_Y\_ROM.v module instead. The differences are described in the technical description, below.

## Signal Interconnections

The PT12 signal interconnect diagram is shown in Figure 3.



**Figure 3 PT12 Signal Interconnections**

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 and 74.25MHz for HD video. All data inputs should be valid at the rising edge of this clock. 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'.
Reg_Clk	Clock used for writing to the control registers only. If 'Clock' is continuous and stable Reg_clk may be connected to that input. Clk_en has no affect on this input.
Window	If enabled (Control register 2, bit 0 = '1') this input, when logic '1', switches off the noise reduction. This input may be used to switch the noise reduction between its current setting (= '0') and off (= '1') for demonstration or comparison purposes. If unused this input should be set to logic '0'.

Highlight	If enabled (Control register 2, bit 1 = '1') this input, when logic '1' forces the output to 75% yellow. This input may be used to highlight the border or edge of the window. If unused this input should be set to logic '0'.
A[1..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_CS <sub>n</sub>	Control chip select input, active low. Used in combination with the WR <sub>n</sub> input to control writing to the control registers.
PT12_WR <sub>n</sub>	Active low write enable input. Used in combination with the CS <sub>n</sub> input to control writing to the control registers.
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.
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.
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.
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[1..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**

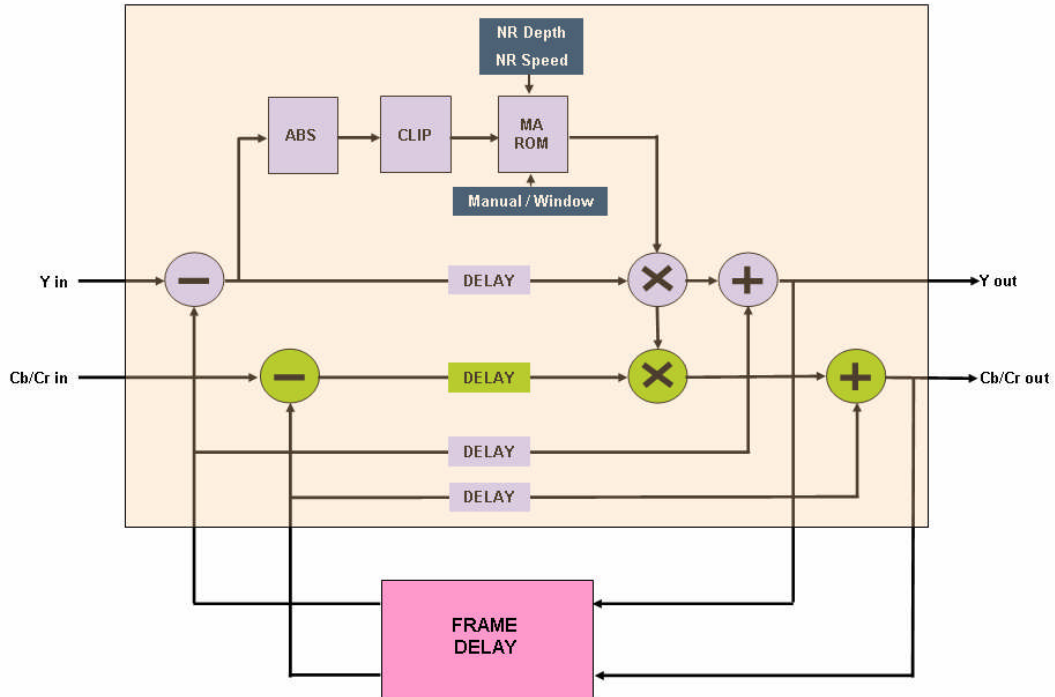
The Verilog instantiation of PT12 is shown below:

**// Instantiate Video noise reducer (PT12)**

```
PT12 #(,BIT_DEPTH_Y(10), ,BIT_DEPTH_C(10)) PT12_inst
(
    .RESETn(RESETn_sig), // input RESETn_sig
    .Clock(Clock_sig), // input Clock_sig
    .Clk_en(Clk_en_sig), // input Clk_en_sig
    .Reg_Clk(Reg_Clk_sig), // input Reg_Clk_sig
    .Window(Window_sig), // input Window_sig
    .Highlight(Highlight_sig), // input Highlight_sig
    .A(A_sig), // input [1:0] 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
    .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 [9:0] Cb_NR_delay_out_sig
    .Cr_NR_delay_out(Cr_NR_delay_out_sig), // input [9:0] 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
);
```

## Technical Overview

A simplified block diagram of the PT12 noise reducer is shown in Figure 2.



**Figure 4 PT12 Block diagram**

The video inputs to the PT12 are separate Y, Cb and Cr channels at programmable bit depths, (supporting 8, 10 and 12 bits).

Each input is subtracted from the frame delayed data: (the frame delay, because of processing delays in the PT12 needs to be 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).

The frame delay memory, because of its size, is usually external to the FPGA/ASIC. A suitable SD controller for an SDRAM is available free of charge on the SingMai website.

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:

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

- Y<sub>out</sub> is the output luma (or Cb/Cr output)
- Y<sub>in</sub> is the input luma (or Cb/Cr input)
- Y<sub>delay</sub> 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). This 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\_depth[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. To save on the memory footprint the luma LUT maybe used for both luma and chroma if required.

The LUT adaptation values are shown graphically in Figures 3 and 4.

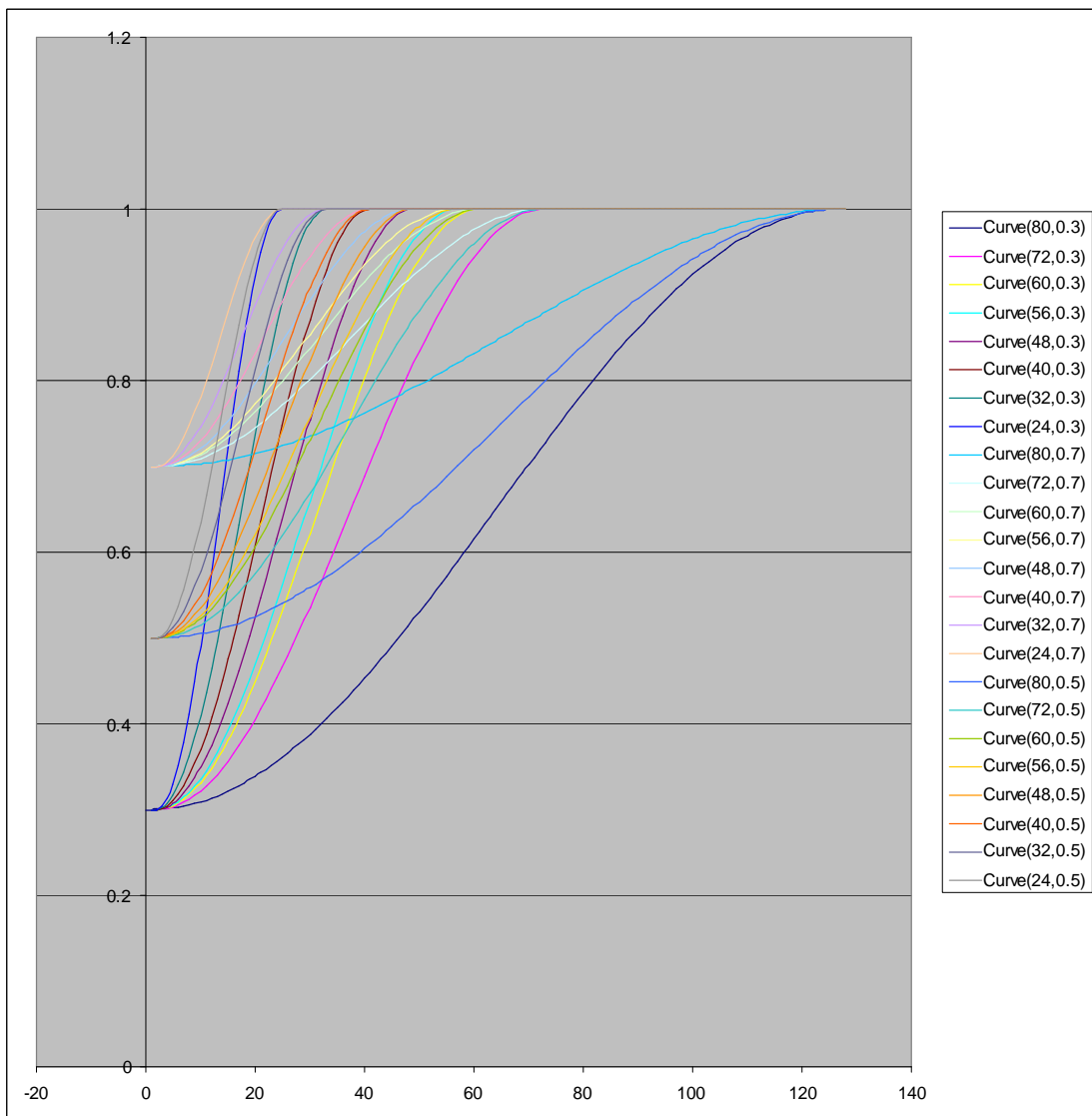
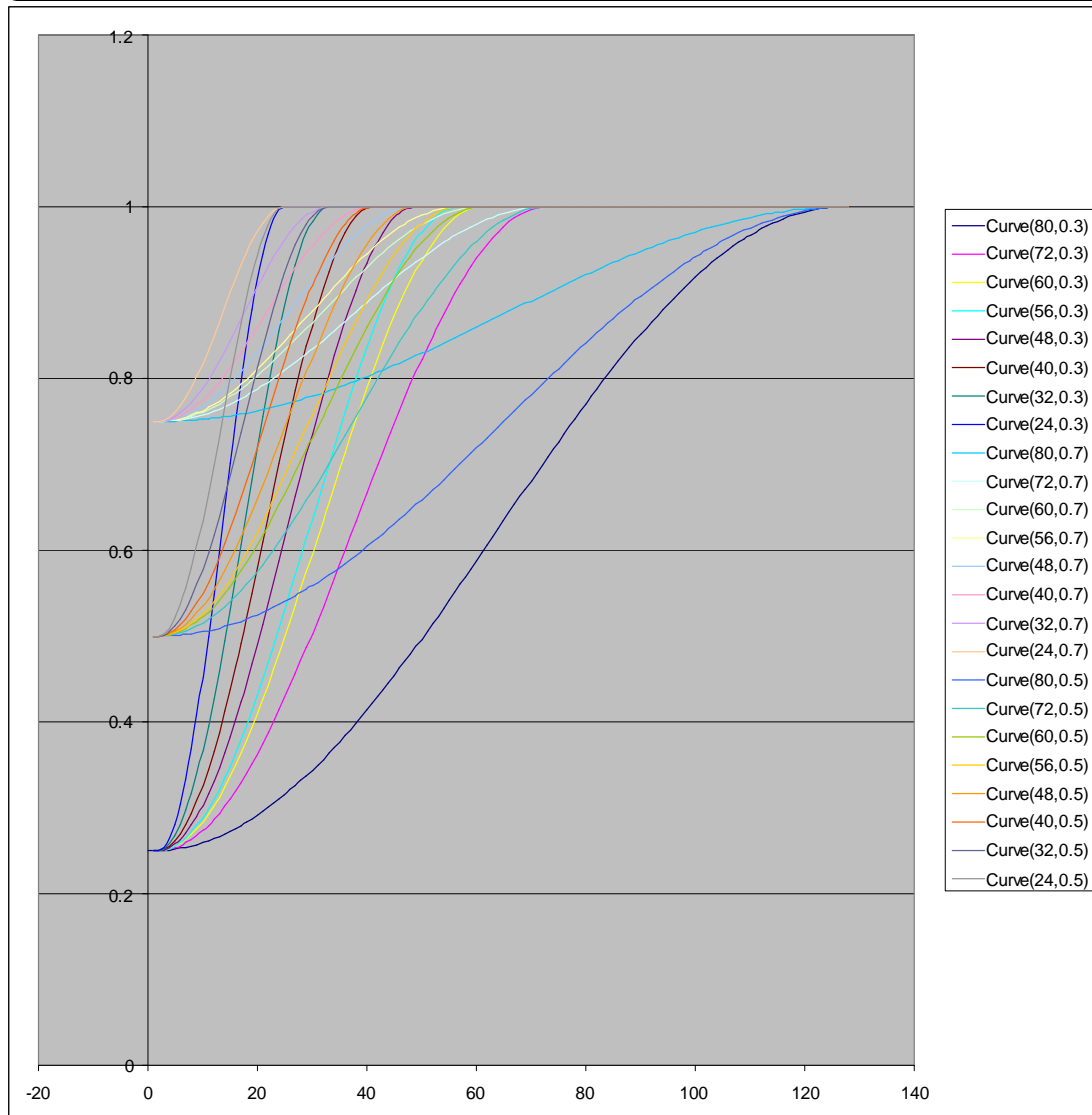


Figure 5 Y (Luma) channel noise reduction adaptation tables



**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 using register \$02.

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 Window input to the PT12 may be used to switch the noise reduction between its current value and off. This can be useful for comparison or demonstration purposes. The window position (e.g a horizontal split) may be highlighted (in yellow).

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

## Register interface

Figure 5 shows the timing diagram for the register interface; it is a conventional microprocessor interface. Each register is selected via an 8 bit address bus. Writes to unused register locations are ignored.

To write to the selected register the PT12\_CS<sub>n</sub> (chip select) input must be asserted low. Whilst this is low the PT12\_WR<sub>n</sub> must be taken low. An internal write enable pulse is created at the next rising edge of the Reg\_Clk clock and writing occurs at the next clock edge following that enable.

For the write to occur reliably the address (A[2:0]) and data (Din[7:0]) must be stable and valid during the PT12\_WR<sub>n</sub> pulse. The minimum width of the PT12\_WR<sub>n</sub> pulse is 80ns for a 27MHz clock.

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 other control signals or the Reg\_Clk clock.

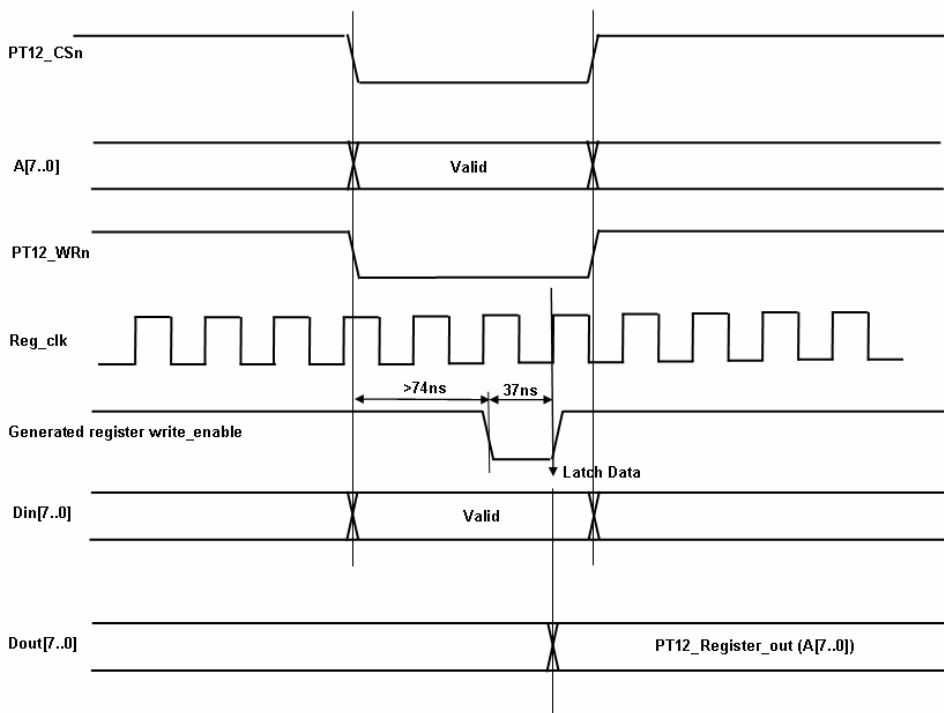


Figure 7 PT12 Register interface.

## 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	<b>Control 1</b>	R/W		<b>Noise reduction control</b>
			7	Not used
			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	127
			3:2	NR Y (luma) depth control
			Bits [3:2]	Y 'k' Factor
			00	1.0 (NR off)
			01	0.7
			10	0.5
11	0.3			
1:0	NR C (chroma) depth control			
Bits [3:2]	C 'k' Factor			
00	1.0 (NR off)			
01	0.75			
10	0.5			
11	0.25			
\$01	<b>Control 2</b>	R/W		<b>Noise reduction control</b>
			7:3	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.
			1	If '1' the highlight external input is enable, else if '0' the input is ignored.
0	If '1' the window external input is enable, else if '0' the input is ignored.			
\$02	<b>Manual k value</b>	R/W	7	Not used
			6:0	Adjusts the manual value of the 'k' feedback factor (if register \$00 = \$00 and register \$01 bit 2 is set). Value 0 is a 'k' factor of 1.0 (NR off) and value 127 is a 'k' factor of 0.006 (maximum noise reduction)
\$03	<b>Version No.</b>	R	7:0	PT12 IP core version number.

**Table 3 Register Descriptions**