



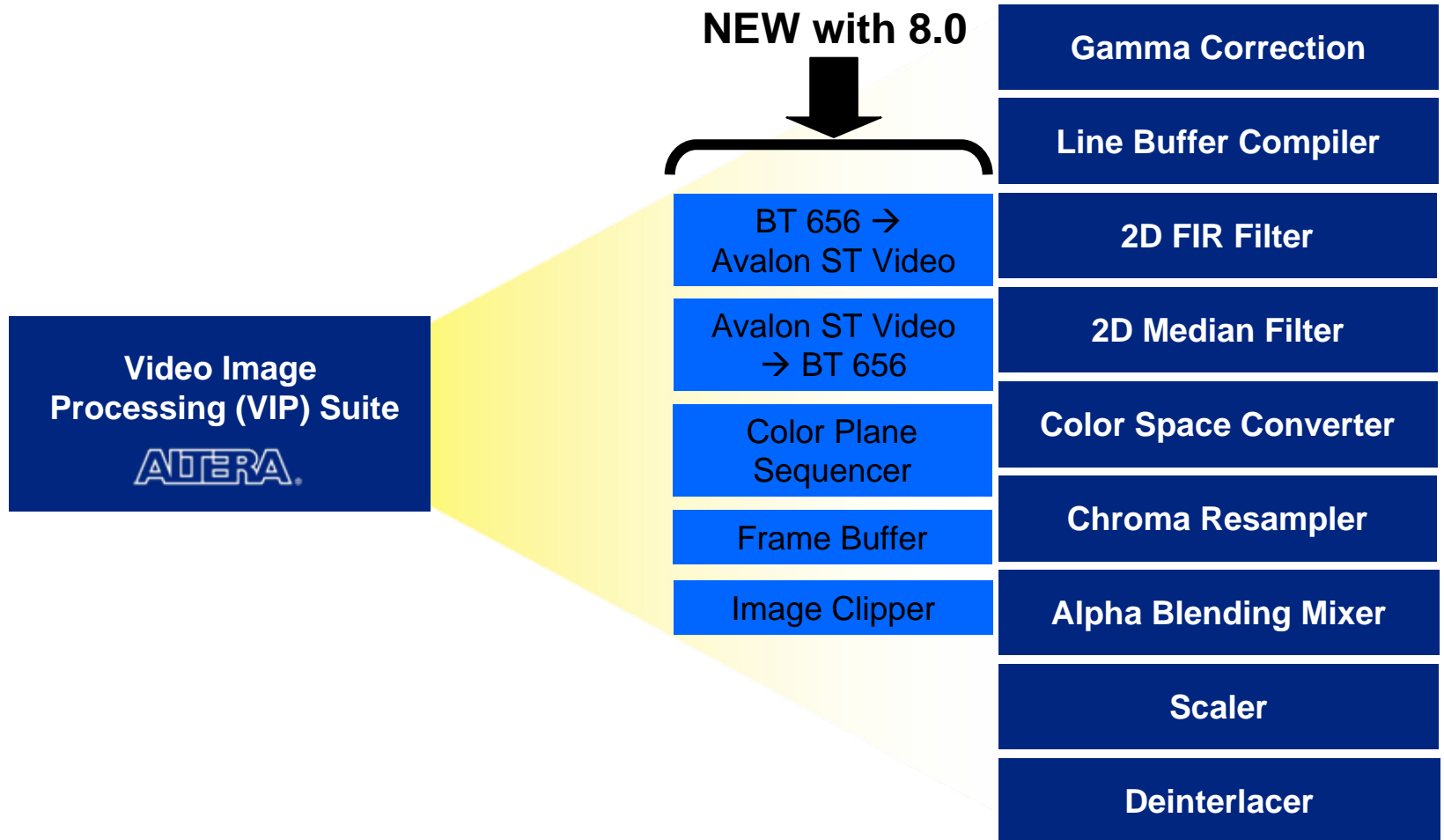
think **AND** not **OR**
Altera @ 40 nm

Altera Video Image Processing (VIP) Solution

Agenda

- What is the VIP Suite of intellectual property (IP) cores?
- Why we built VIP Suite: Typical signal chains implemented using VIP cores
- Overall VIP value proposition
- VIP Suite – core details
- Available VIP resources

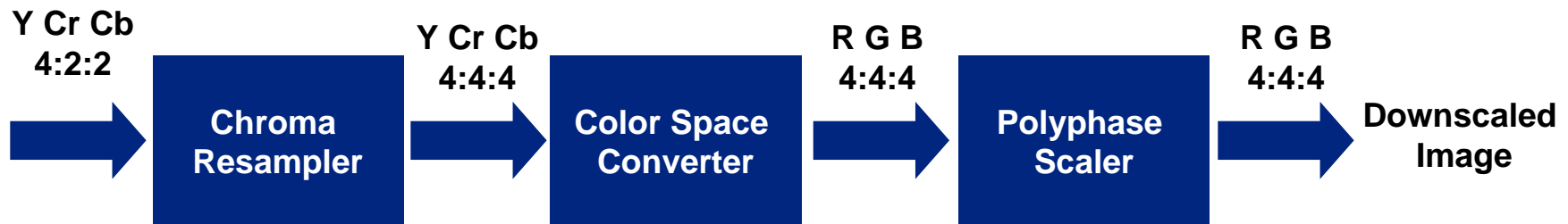
Video Image Processing (VIP)



Suite of building block IP for video processing

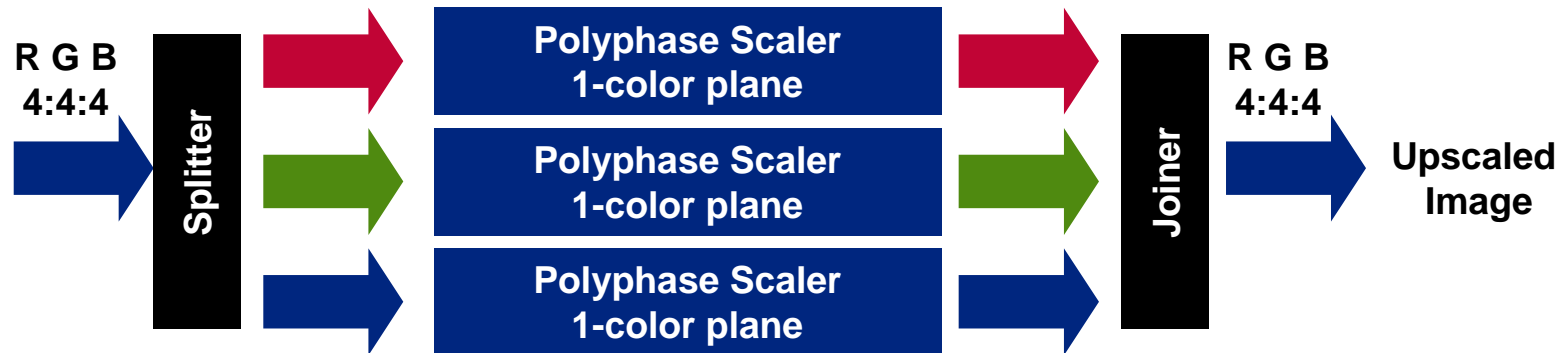
Downscaling – Monitoring Application

- In various kinds of systems, there is a need to downscale and view large video frames on a small display – monitoring signal chain
- VIP cores let you quickly build a high-quality monitoring signal chain by combining the polyphase scaler with other appropriate cores; an example is shown below



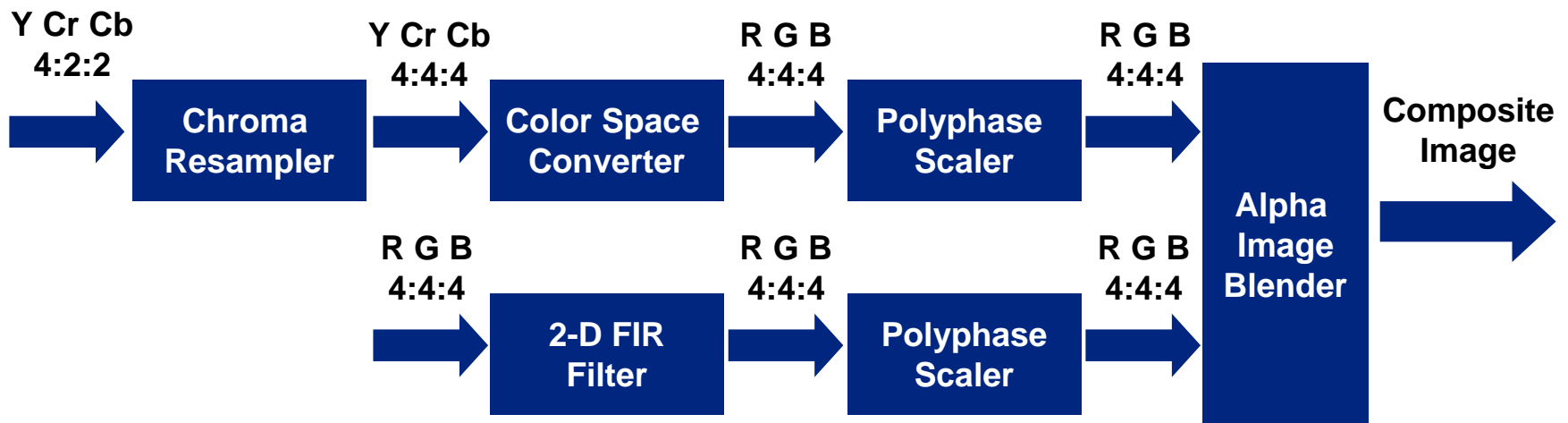
Upscaling – Zooming-In Application

- In various kinds of systems, there is a need to crop a certain section of a video frame and upscale that portion to view on a display
- VIP cores let you quickly build a high-quality upscaler using the built-in crop features within the scaler
- The bicubic scaling option that is built in the VIP scaler offers the best quality/resource trade-off for an upscaling application



Video Mixing Application

- In various kinds of systems, there is a need to mix two streams of video and display on the screen
- The VIP core can let you quickly build multiple custom video processing chains and mix, for example, two streams using the alpha blending core



Typical Signal Chains: Summary

- There are various other types of signal chains that can be implemented using the VIP cores as well as using VIP cores in conjunction with your custom cores
- VIP cores are designed for easy connectivity, allowing you to start a design using a VIP core and swap it with your custom function
- In many applications the VIP cores are used to do the simple pre-processing and combined with your custom IP to create a complete signal chain



think **AND** not **OR**

Altera @ 40 nm

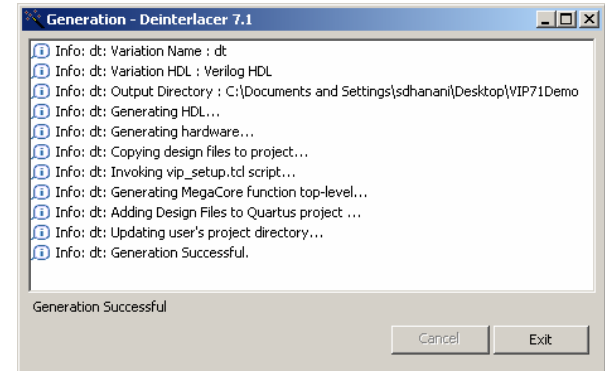
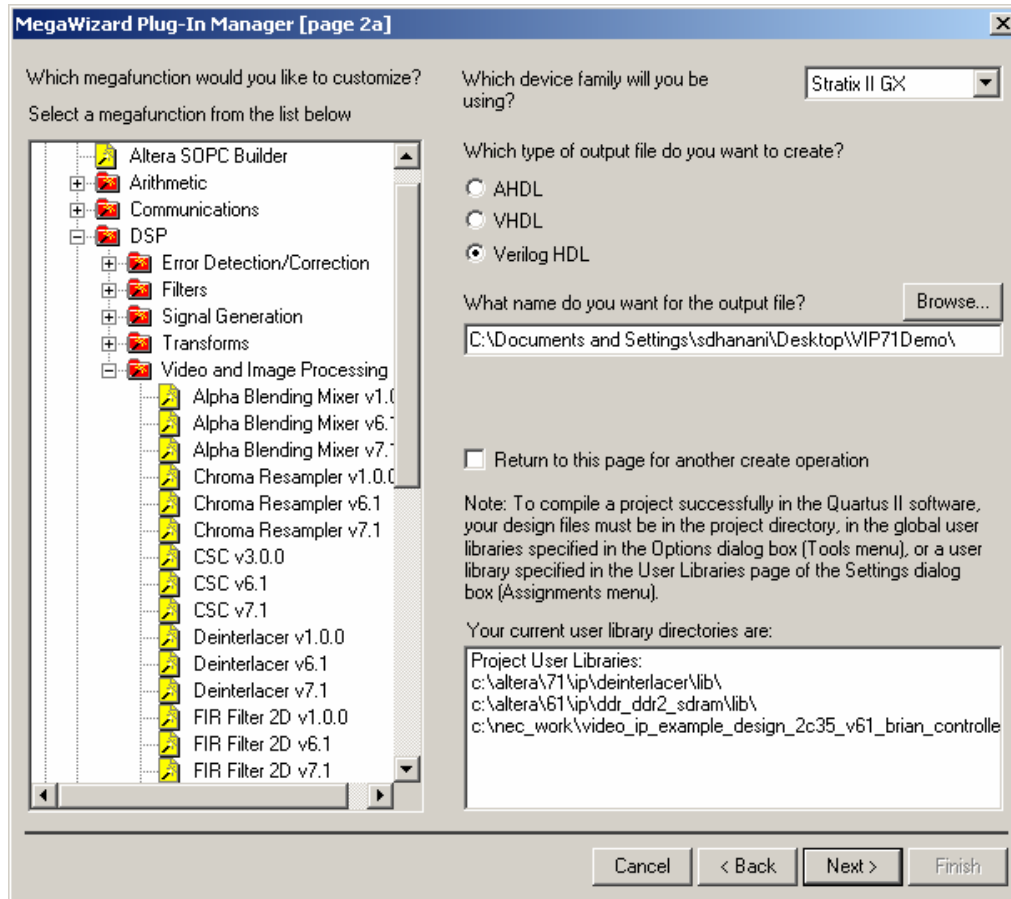
Overall VIP Value Proposition

Video Image Processing with FPGAs

- Video processing that is ***computationally intensive***, fits elegantly in the inherently parallel FPGA architecture i.e.
 - Polyphase video scaling
 - Motion estimation/compensation
 - Motion adaptive deinterlacing
- VIP Suite consists of blocks commonly used to implement complex video/image processing circuits

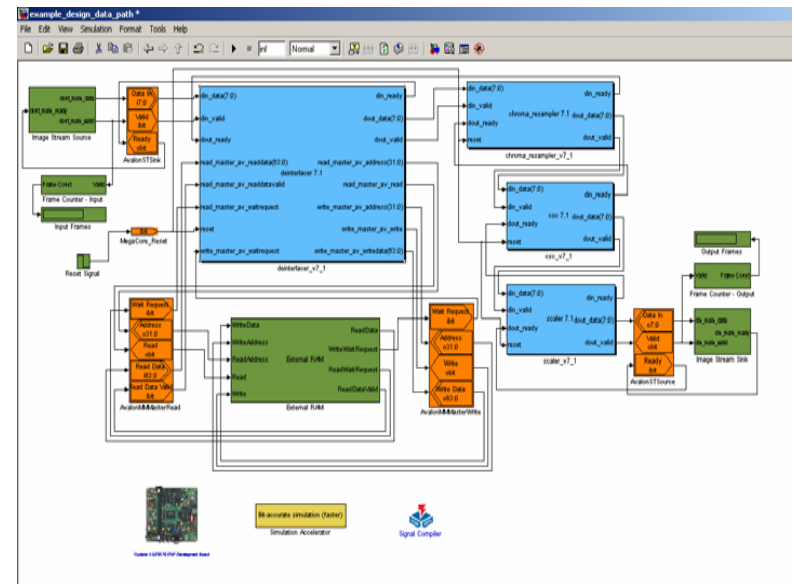
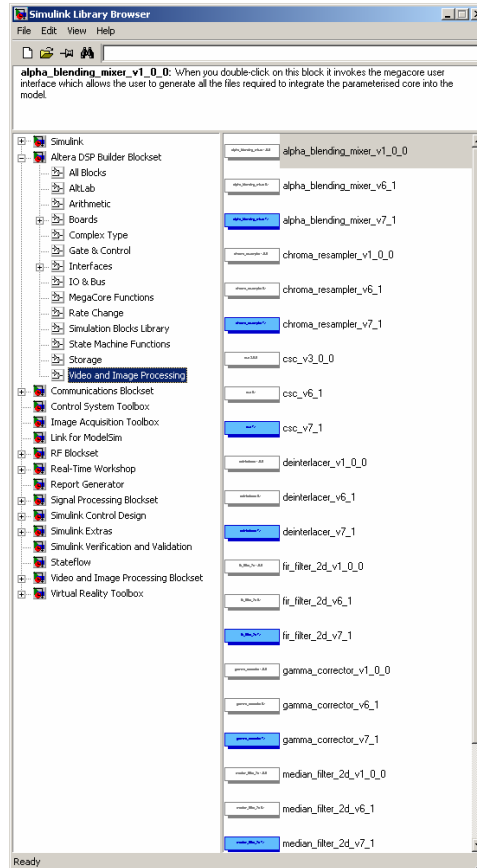
Using Altera VIP Cores: Quartus II Software

- VIP cores are configurable using Quartus® design software

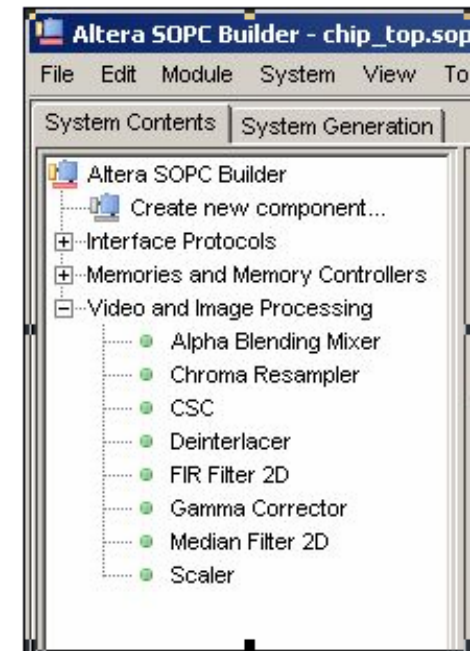
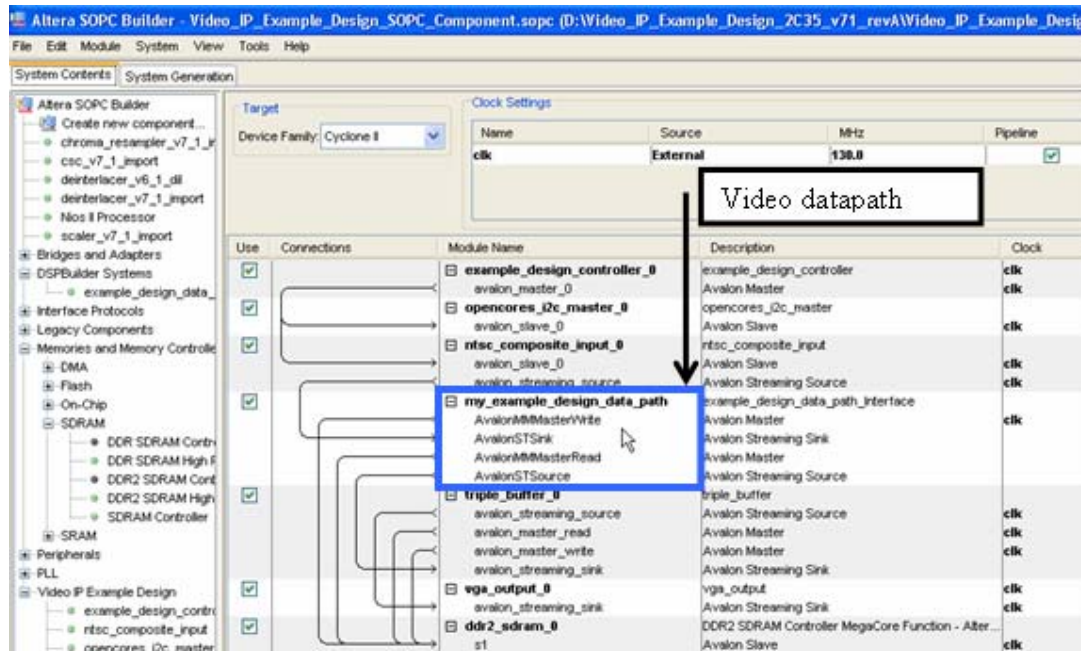


Using Altera VIP Cores: Simulink

- VIP Suite can be configured using MATLAB/Simulink



SOPC Builder Support

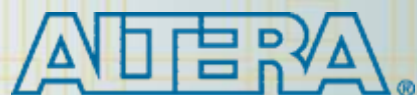


- All the VIP cores are SOPC Builder ready – they can be instantiated and connected using the SOPC Builder environment
- VIP signal chain built and simulated using Simulink can also be exported as a system on a programmable chip (SOPC) component and connected to other system blocks using the SOPC Builder

VIP Value Proposition

- Altera VIP Suite of cores are basic building blocks for video/imaging systems
- VIP IP cores speed up your development cycle by allowing you to focus on IP that is your value add
- VIP cores are designed for easy connectivity, enabling seamless mix-and-match of Altera and proprietary cores

*Accelerating Your
Time to Market*



think **AND** not **OR**
Altera @ 40 nm

VIP Suite – Core Details

VIP Suite: Avalon ST Video → Clocked Video

NEW
with
8.0

Clocked Video Input - my_alt_vip_cti

Clocked Video Input

Parameter Settings

Preset Loader

Preset conversion: HD-SDI 1080i Load values into controls

Avalon-ST-Video Image Data Format

Bits per pixel per color plane: 10 Bits

Number of color planes: 2

Color plane transmission format: Sequence Parallel

General Parameters

Pixel fifo size: 1,920 Pixels

☐ Video in and out use the same clock

☐ Use stop/go control port

Avalon-ST-Video Initial/Default Control Packet

☐ Progressive ☒ Interlaced

Image Width

Progressive / Field 0: 1,920 Pixels

Field 1: 1,920 Pixels

Image Height

Progressive / Field 0: 540 Pixels

Field 1: 540 Pixels

Clocked Video Sync Signals

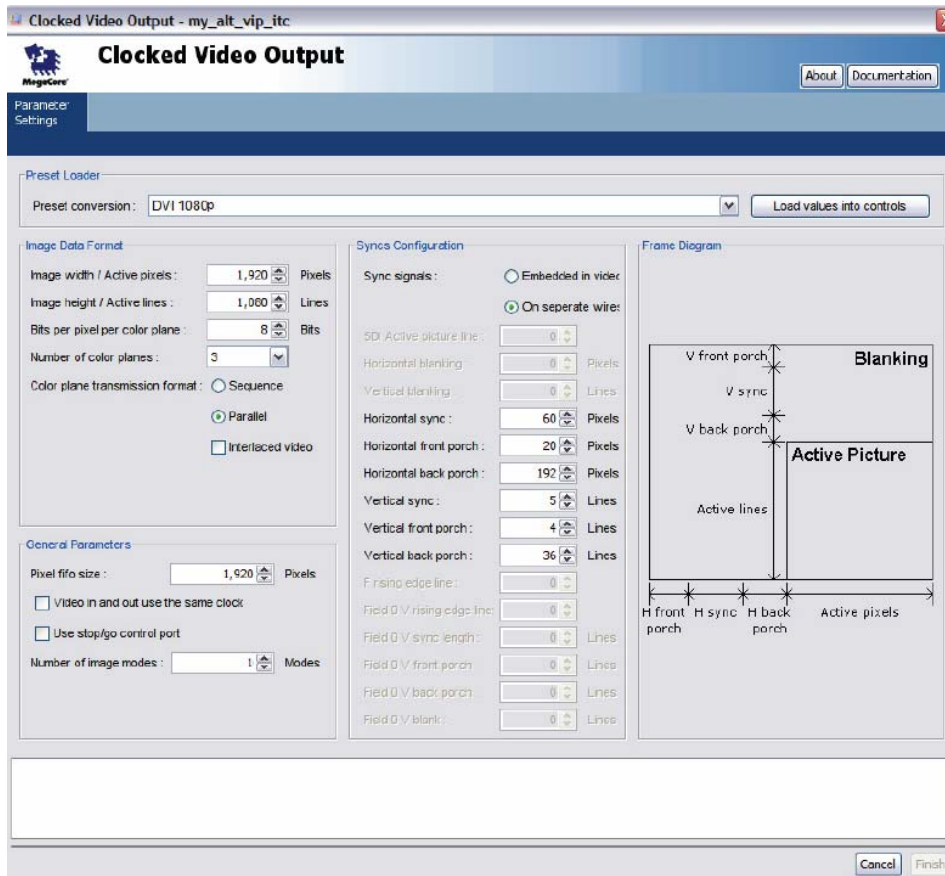
☒ Embedded in video ☐ On separate wires

Cancel Finish

- For example if the input is a HD-SDI 1080i video stream
 - Select that format in the pull down menu
 - Set other parameters such as bits per pixel, # of color planes, image dimensions, etc as shown
 - Automatically convert from SDI to Avalon ST Video format

VIP Suite: Clocked Video → Avalon ST Video

NEW
with
8.0



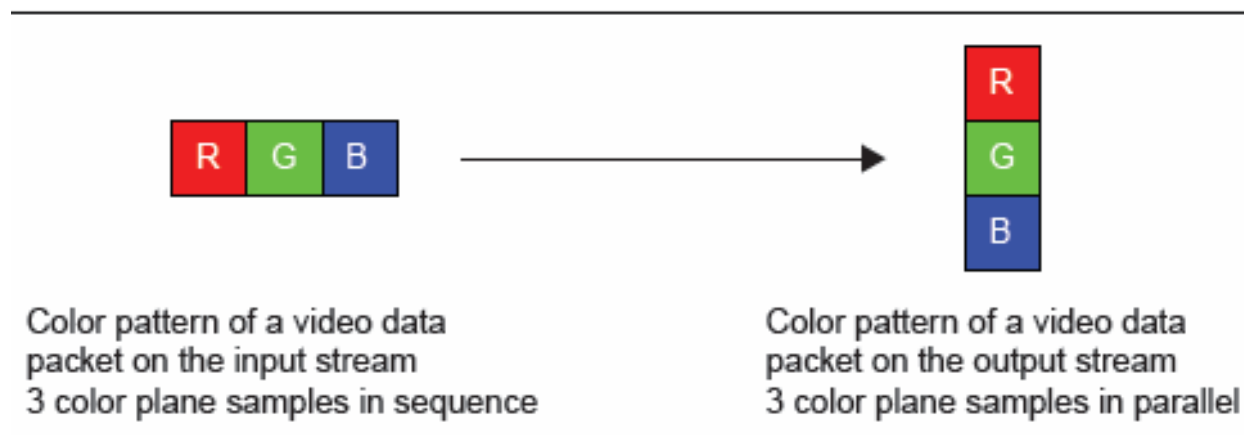
■ For example if you need to create a DVI 1080p video stream

- Select that format in the pull down menu
- Set other parameters such as bits per pixel, # of color planes, image dimensions, etc as shown
- Select the sync parameters
- Automatically convert from Avalon ST Video to the DVI format

VIP Suite: Color Plane Sequencer

NEW
with
8.0

- The Color Plane Sequencer can change how color plane samples are transmitted across the Avalon-ST interface
- The color pattern of a video data packet can be rearranged in any valid combination of channels in sequence and parallel.
- The Color Plane Sequencer can also drop color planes.



VIP Suite: Color Plane Sequencer

NEW
with
8.0

Example shows a Color Plane Sequencer used to split an Avalon-ST video containing 4:2:2 subsampled data (Y'CbCr) into separate luminance (Y') and chrominance (Cb,Cr) streams

MegaWizard Plug-In Manager - Color Plane Sequencer

Color Plane Sequencer

About Documentation

1 Parameter Settings 2 EDA 3 Summary

Port Configuration

Bits per pixel per color plane : 8 Bits

☒ Two pixels per port

Port and Channel Mapping

din0

Color planes in parallel : 1 Color planes in sequence : 4

	Symbol (t+0)	Symbol (t+1)	Symbol (t+2)	Symbol (t+3)
Bits (7-0)	Cb	Y	Cr	Y
inactive				
inactive				

din1

☐ Port enabled

Color planes in parallel : 3 Color planes in sequence : 1

	inactive	inactive	inactive	inactive
inactive				
inactive				
inactive				

dout0

☒ Port enabled

Source non-image packets from port : din 0 ☐ Halve control packet width

Color planes in parallel : 1 Color planes in sequence : 2

	Symbol (t+0)	Symbol (t+1)	inactive	inactive
Bits (7-0)	Cb	Cr		
inactive				
inactive				

dout1

☒ Port enabled

Source non-image packets from port : din 0 ☐ Halve control packet width

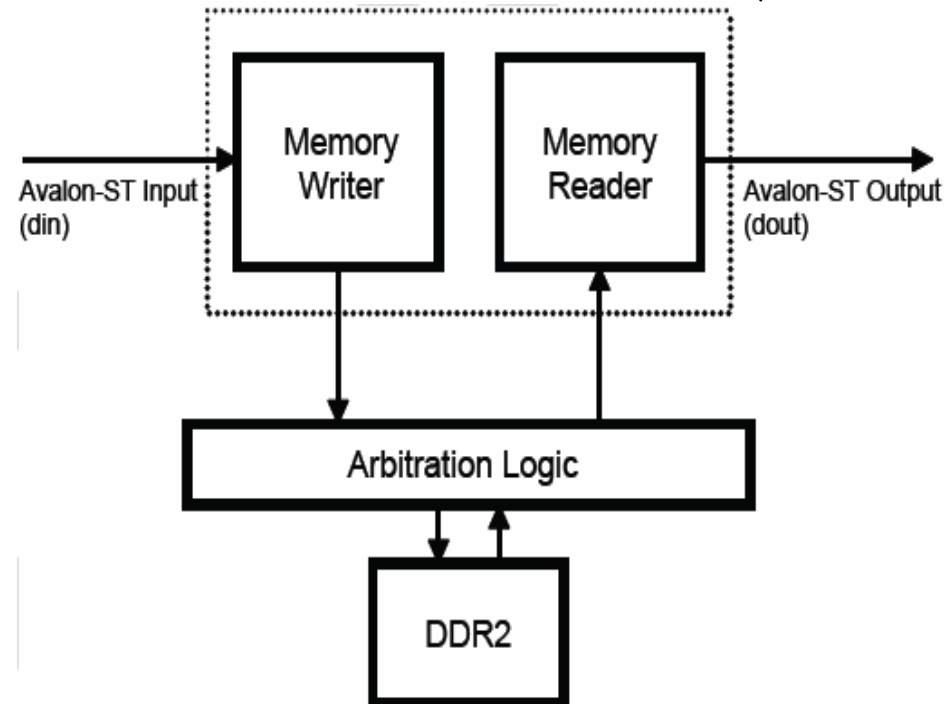
Color planes in parallel : 1 Color planes in sequence : 2

	Symbol (t+0)	Symbol (t+1)	inactive	inactive
Bits (7-0)	Y	Y		
inactive				
inactive				

VIP Suite: Frame Buffer

NEW
with
8.0

- The Frame Buffer core is used to buffer video frames into external RAM
- It supports double or triple-buffering with a range of options for frame dropping and repeating
- The Frame Buffer is built with two basic blocks
 - A writer which stores input pixels into memory
 - A reader which retrieves video frames from the memory and outputs them



VIP Suite: Frame Buffer

NEW
with
8.0

- Frame Buffer settings to allow for triple-buffering of a 480×720 R'G'B' video stream transmitted in parallel

Frame Buffer - my_alt_vip_vfb

Frame Buffer

Parameter Settings

Image Data Format

Maximum image width : 480 Pixels

Maximum image height : 270 Pixels

Bits per pixel per color plane : 8 Bits

Number of color planes in sequence : 1 Planes

Number of color planes in parallel : 3 Planes

Non-image data handling

Number of packets buffered per frame : 1

Maximum packet length : 10

Behavior

External memory port width : 256

Base address of frame buffers : 0x00000000

☐ Runtime control for the writer thread

☒ Frame dropping

Write-only master interface FIFO depth : 64

Write-only master interface burst target : 32

☐ Runtime control for the reader thread

☒ Frame repetition

Read-only master interface FIFO depth : 64

Read-only master interface burst target : 32

3 frame buffers are required, a total of 1216 kilobytes.

VIP Suite: Clipper



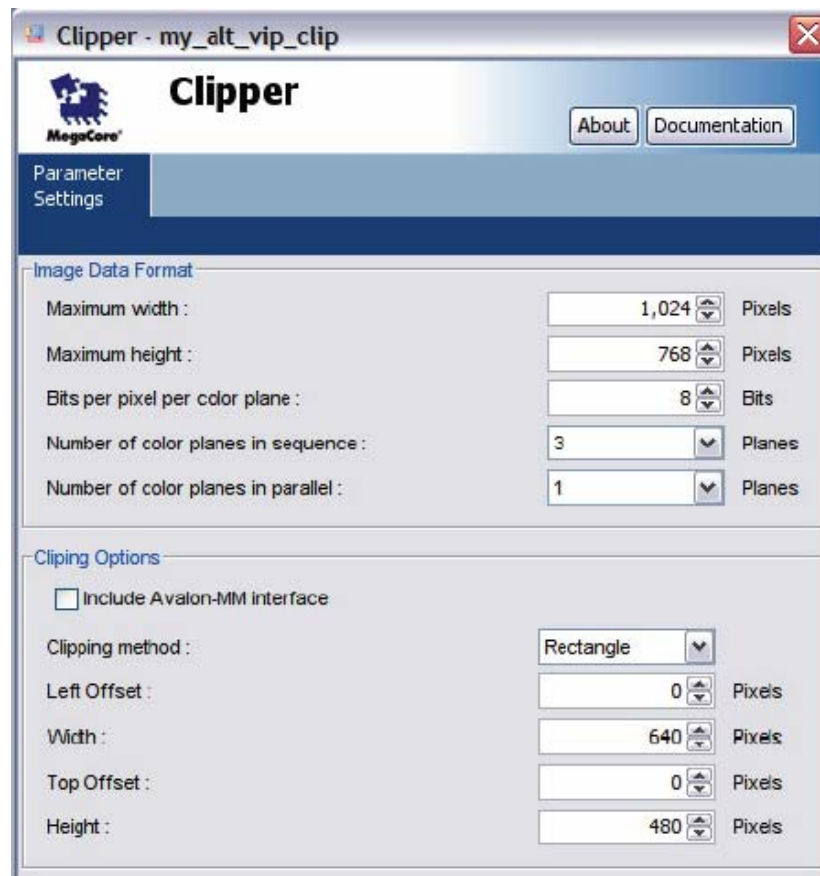
NEW
with
8.0

- The Clipper MegaCore function provides a means to select an active area from a video stream and discard the remainder
- The active region can be specified by either providing the offsets from each border, or by providing a point to be the top-left corner of the active region along with the region's width and height
- The Clipper can deal with changing input resolutions by reading Avalon-ST Video control packets
- An optional Avalon-MM interface allows the clipping settings to be changed at runtime

VIP Suite: Clipper

NEW
with
8.0

- Shown is a Clipper core parameterized to crop the upper-left 640×480 pixels of a 1024×768 video stream.



VIP Suite: Run-time Configurable Polyphase Scaling



D1/SDTV: 720x480



HDTV 1080p: 1920x1080

- High-quality multi-tap scaler with use models in monitoring (downscaling) and zooming (upscaling) applications

VIP Suite: Run-time Configurable Polyphase Scaling

Scaler - my_alt_vip_scl

Scaler

Parameter Settings
Resolution

Scaling from arbitrary input image size to arbitrary output image size

☐ Run-time control of image size

Input image width : 1,024 Pixels
Input image height : 768 Pixels
Output image width : 640 Pixels
Output image height : 480 Pixels
Bits per pixel per color plane : 8 Bits
Number of color planes : 3 Planes
Color plane transmission format : Sequence

Color planes can be input in parallel or in sequence

Cancel < Back Next > Finish

Scaler - my_alt_vip_scl

Scaler

Parameter Settings
Resolution

Set the number of taps & phases in the horizontal and vertical dimension

Scaling algorithm : Polyphase

Number of vertical taps : 4 Number of vertical phases : 16
Number of horizontal taps : 4 Number of horizontal phases : 16

Vertical Coefficient Precision

Number of bits used in vertical coefficients:
☒ Signed Integer bits : 1 Fraction bits : 7
The total word length of each vertical coefficient is 9 bits

Horizontal Coefficient Precision

Number of bits to preserve between vertical and horizontal filtering : 9 bits
Number of bits used in horizontal coefficients:
☒ Signed Integer bits : 1 Fraction bits : 7
The total word length of each horizontal coefficient is 9 bits

Set the co-efficient precision

Cancel < Back Next > Finish

VIP Suite: Run-time Configurable Polyphase Scaling

The screenshot shows the 'Scaler' application window with the 'Parameter Settings' tab selected. The 'Coefficient Data' section is active, showing options for 'Load coefficient' (unchecked) and 'Vertical Coefficient Data'. Under 'Vertical Coefficient Data', 'Memory banks' is set to 2 and 'Filter function' is set to 'Lanczos 2'. A 'Preview coefficients' button is visible. Below this, the 'Horizontal Coefficient Data' section is also visible with similar settings. At the bottom, there are 'Cancel', '< Back', 'Next >', and 'Finish' buttons.

Choose coefficients generated from 12 pre-defined Lanczos functions

OR choose to import custom coefficients

Preview the coefficients

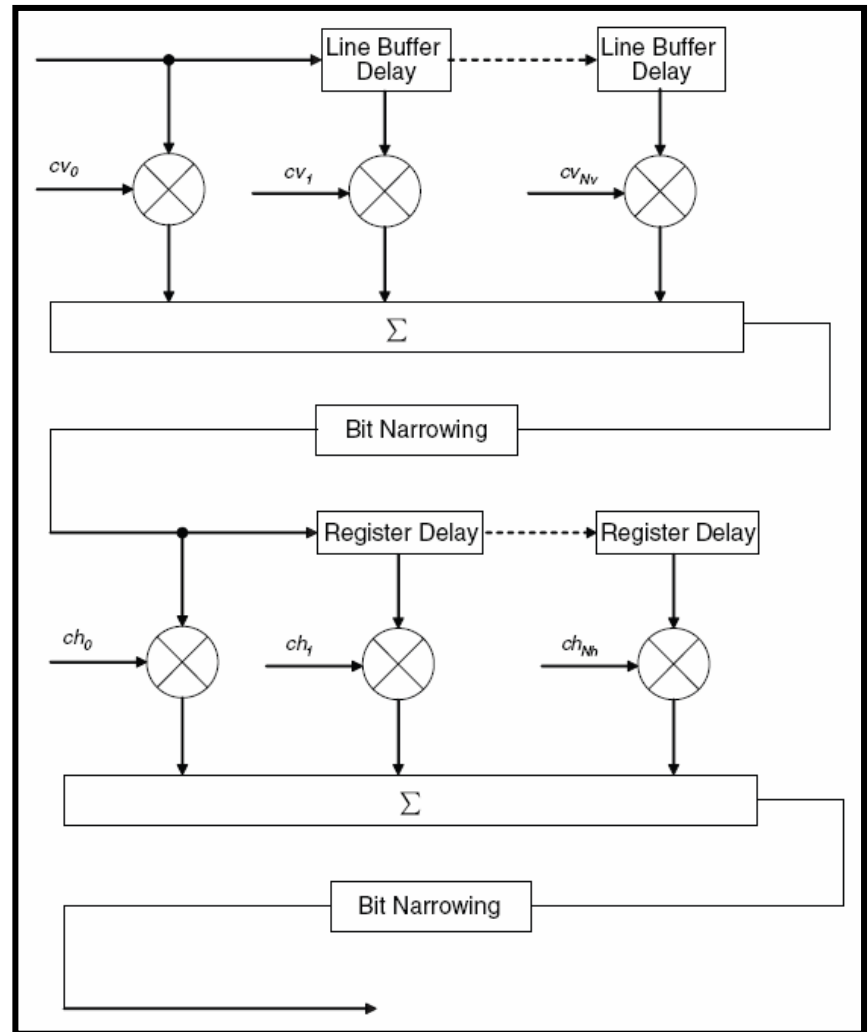
The screenshot shows the 'Scaler Vertical Coefficient Preview' window, which displays a table of coefficients for 16 phases. The table has five columns: Phase, Coeff 0, Coeff 1, Coeff 2, and Coeff 3. The values are as follows:

Phase	Coeff 0	Coeff 1	Coeff 2	Coeff 3
0	0	128	0	0
1	-4	126	6	0
2	-8	124	13	-1
3	-10	119	20	-1
4	-11	111	30	-2
5	-11	103	40	-4
6	-10	93	50	-5
7	-9	82	61	-6
8	-8	72	72	-8
9	-6	61	82	-9
10	-5	50	93	-10
11	-4	40	103	-11
12	-2	30	111	-11
13	-1	20	119	-10
14	-1	13	124	-8
15	0	6	126	-4

A 'Close' button is located at the bottom right of the window.

Polyphase Scaling Algorithm

- Data from multiple lines of the input image are assembled into line buffers – one for each vertical tap
- These data are then fed into parallel multipliers, before summation and possible loss of precision
- The results are gathered into registers – one for each horizontal tap
- These are again multiplied and summed before precision loss down to the output data bit width



Polyphase Scaling Algorithm

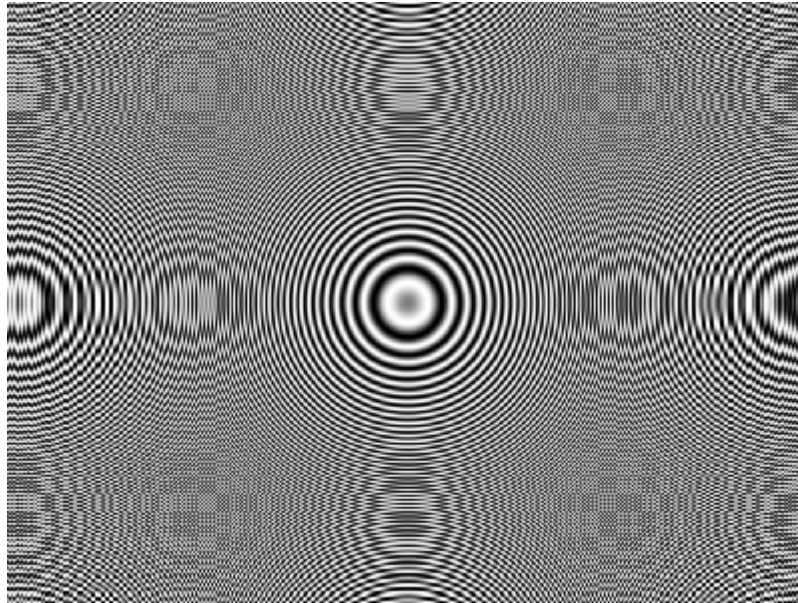
- In polyphase mode, the parameters for the scaler function must be chosen carefully to get the best image quality
- Incorrect parameters can cause a decrease in image quality even as the resource usage increases
- The parameters which have the largest effect are the number of taps and the filter function chosen to provide the coefficients
- The number of phases and number of bits of precision used are less important to the image quality

Recommended parameters for the scaler function				
Scaling problem	Taps	Phases	Precision	Coefficients
Scaling up with any input/output resolution	4	16	Signed, 1 integer bit, 7 fraction bits	Lanczos-2, or Bicubic
Scaling down from M pixels to N pixels	$\frac{M \times 4}{N}$	16	Signed, 1 integer bit, 7 fraction bits	Lanczos-2
Scaling down from M pixels to N pixels (lower quality)	$\frac{M \times 2}{N}$	16	Signed, 1 integer bit, 7 fraction bits	Lanczos-1

ORIGINAL IMAGE TO BE DOWNSCALED

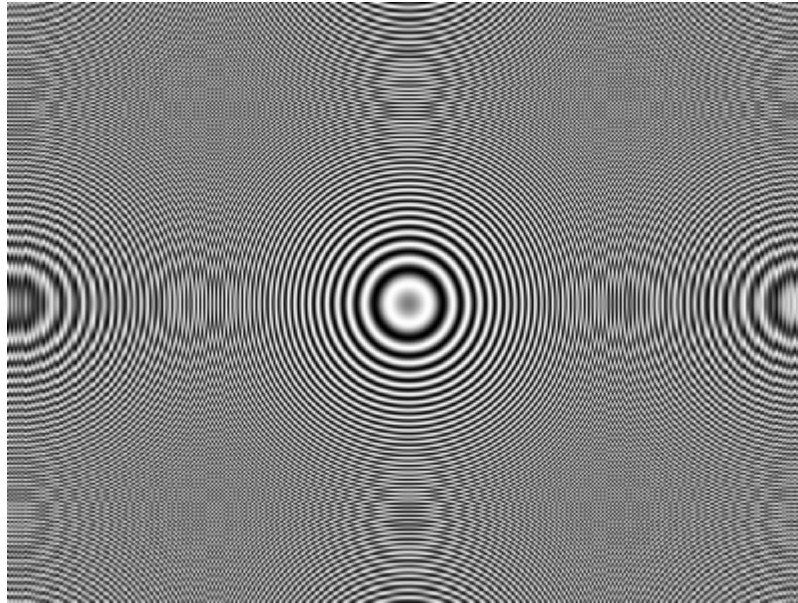


Downscaling Using Nearest Neighbor Interpolation



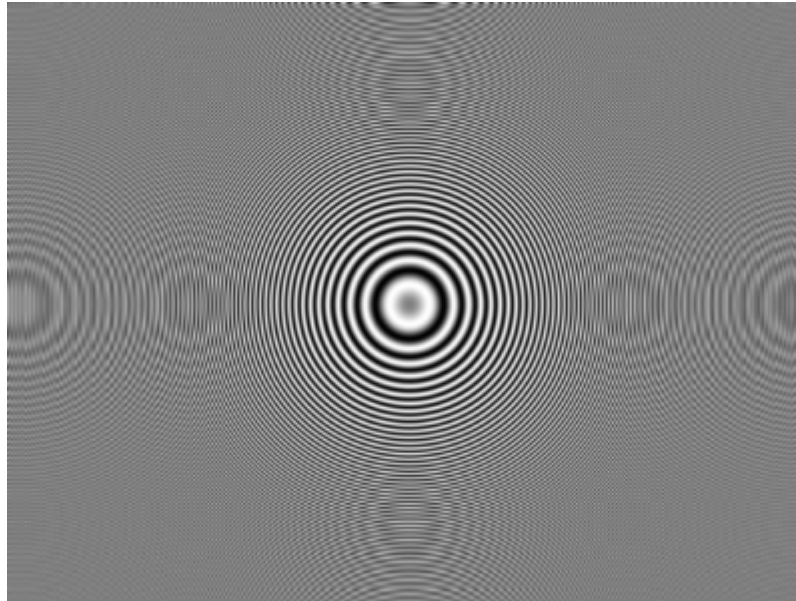
Nearest neighbor algorithm uses one neighboring pixel to interpolate.
This results in severe scaling artefacts.

Downscaling Using Bilinear Interpolation



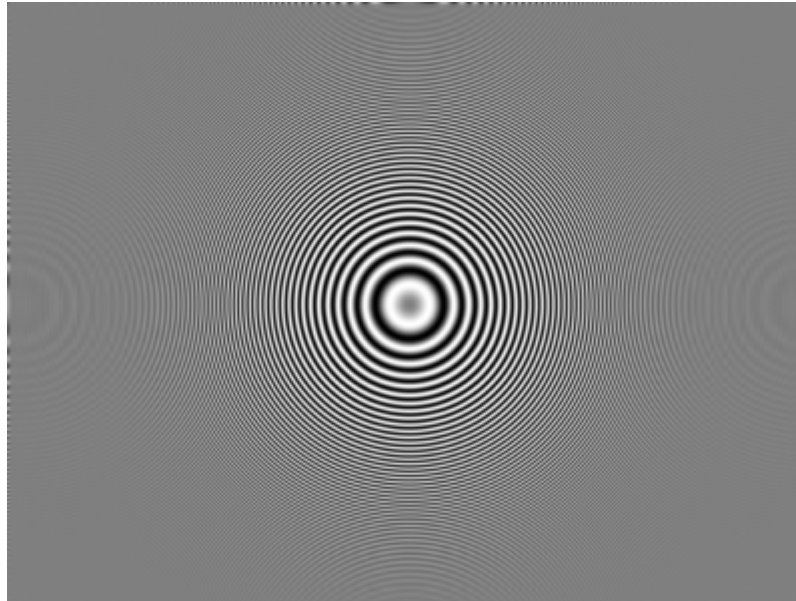
Bilinear algorithm uses a 2x2 matrix of pixels to interpolate.
This results in fewer (though significant) scaling artefacts.

Downscaling Using 5-Tap Lanczos-1 Interpolation



A 5-tap Lanczos-1 algorithm uses a 5x5 matrix of pixels to interpolate. This results in significantly reduced amount of scaling artefacts.

Downscaling Using 9-Tap Lanczos-2 Interpolation



A 9-tap Lanczos-2 algorithm uses a 9x9 matrix of pixels to interpolate. This results in very high-quality scaling – at the expense of resources.

VIP Suite: Deinterlacer With Motion Adaptive Algorithm

Deinterlacer - my_alt_vip_dil

Deinterlacer

Parameter Settings

Maximum image width : 720 Pixels

Maximum image height : 486 Pixels

Bits per pixel per color plane : 8 Bits

Number of color planes in sequence : 2 Planes

Number of color planes in parallel : 1 Planes

Default initial field : ☒ F0 ☐ F1

Non-image data handling

Number of packets buffered per field : 1

Maximum packet length : 10

Deinterlacing method : Weave

Output frame rate : As input frame rate (F1 synchronized)

Frame buffering mode : Double buffering

Avalon-MM master ports width : 16

Base address of frame buffers : 0x00001000

Read-only master(s) interface FIFO depth : 64

Read-only master(s) interface burst target : 32

Write-only master(s) interface FIFO depth : 64

Write-only master(s) interface burst target : 32

The amount of memory which must be free at this address is 1367 kilobytes

Cancel Finish

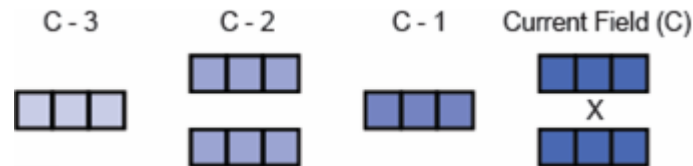
Select the input image height and width

Choose the deinterlacing algorithm – bob, weave, motion adaptive

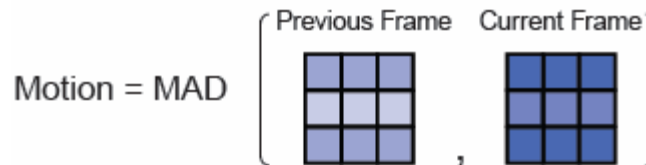
The weave and motion adaptive algorithms require external frame buffering

Motion Adaptive Deinterlacing

- Simple ***motion adaptive*** algorithm
- Pixels are collected from the current field and the three preceding it



- These pixels are assembled into two 3×3 groups of pixels and the minimum absolute difference (MAD) of the two groups is calculated



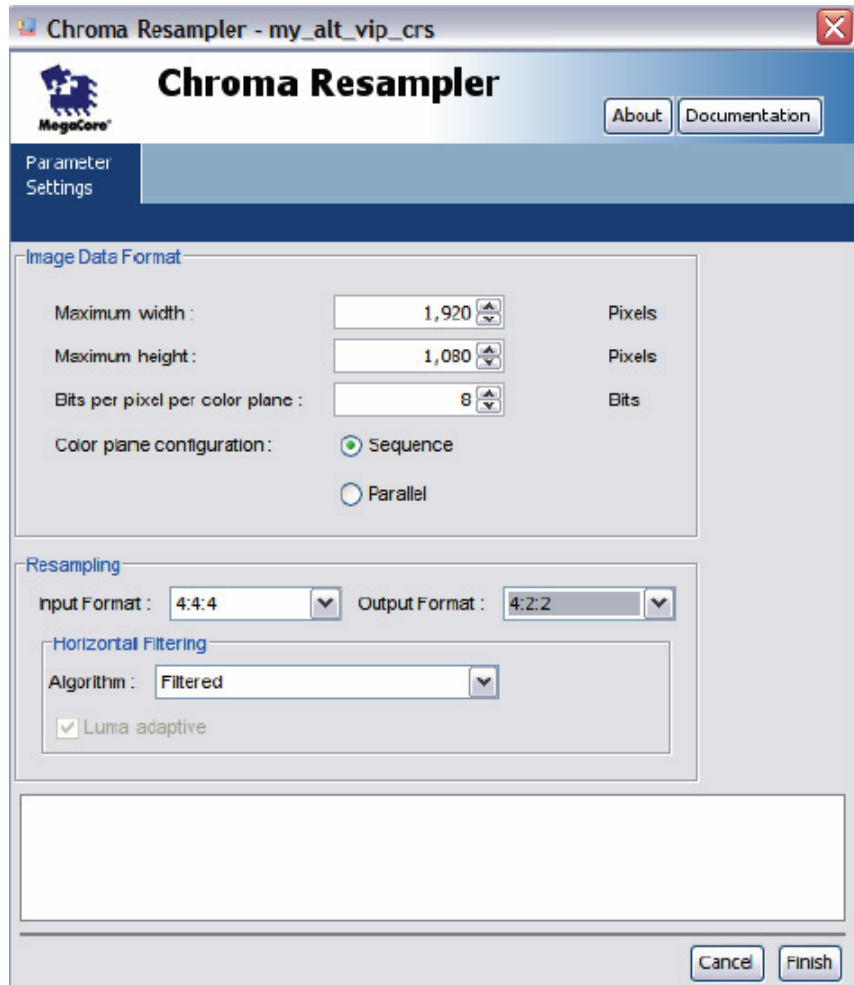
- Compare the MAD value with the previous MAD
 - Either the new value is kept or a mean of the new and old value is used

Chroma Downsampling Reduces the Bits/Frame

Image Size	Frame Size: (Total # of Pixels)	Frame Size: (Assume 10 bits per pixel and 4:4:4)	Frame Size: (Assume 10 bits per pixel and 4:2:2)	Frame Size: (Assume 10 bits per pixel and 4:2:0)
1920X1080p	1920 x 1080 = 2M pixels	60 Mbits	40 Mbits	30 Mbits
1920X1080i	1920 x 1080 x 0.5 = 1M pixels	30 Mbits	20 Mbits	15Mbits
1280X720p	1280 x 720 = 900K pixels	27 Mbits	18 Mbits	13.5 Mbits
SD 720x480i	720 x 480 x 0.5 = 173K pixels	5.19 Mbits	3.46 Mbits	2.595 Mbits

The numbers are not strictly accurate since HSYNC and VSYNC are not taken into account, but are meant to show the difference between a 4:4:4 and a 4:2:0 representation

VIP Suite: Multi-tap Filtered Chroma Resampler



- The Chroma Resampler core allows you to change between 4:4:4, 4:2:2 and 4:2:0 sampling rates
- 4:4:4 → 4:2:2
 - The filtered algorithm for down-sampling uses a 9-tap filter with a fixed set of Lanczos-2 coefficients
- 4:2:2 → 4:4:4
 - The filtered algorithm for up-sampling uses a 4-tap filter with a fixed set of Lanczos-2 coefficients
- For both up-sampling and down-sampling, the vertical resampling (4:2:0) algorithm is fixed at nearest neighbor

VIP Suite: Alpha Blending Mixer

Alpha Blending Mixer - my_alt_vip_mix

Alpha Blending Mixer

About Documentation

Parameter Settings

Image Data Format

Maximum layer width: 720 Pixels

Maximum layer height: 576 Pixels

Bits per pixel per color plane: 8 Bits

Number of color planes in sequence: 3 Planes

Number of color planes in parallel: 1 Planes

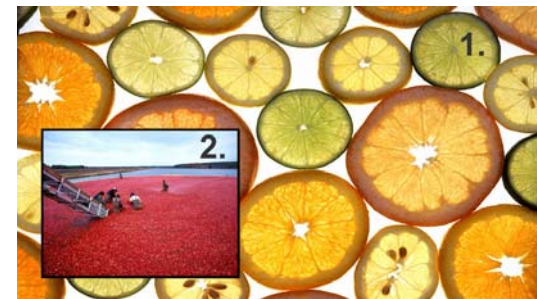
Mixing Options

Number of layers being mixed: 2

☒ Alpha blending Alpha bits per pixel: 4 Bits

Cancel Finish

Select the size of the background image layer



Select the number of layers to be mixed

The level of blending is controlled by the number of alpha bits (up to 8)

VIP Suite: Color Space Conversion

CSC - my_alt_vip_csc

CSC

About Documentation

1 Parameter Settings

General Operands

Image Format

Image width : 720 Pixels

Image height : 576 Pixels

Color Plane Configuration

☒ Three color planes in sequence

☐ Three color planes in parallel

Input Data Type

Bits per pixel per color plane : 8 Bits

Datatype : Unsigned

☒ Guard bands Max : 240 Min : 16

Output Data Type

Bits per pixel per color plane : 8 Bits

Datatype : Unsigned

☐ Guard bands Max : 255 Min : 0

Result to Output Data Type Conversion Procedure

1) The results are in the range 16.00 to 233.00 (to 2 decimal places)
The results have 8 fraction bits

The scaled results have 8 fraction bits

Remove fraction bits by : Truncate values to integer

4) The scaled, integer, sign handled results are in the range 16 to 233
The selected output data type has a range of 0 to 255
The results are within the range of the output data type

Specify the input image sizes in terms of pixels as well as the number of bits/plane/pixel

Input guard bands ensure that the core will never receive data in the guard bands

In the output data, the guard bands option is turned off because the full output range of 0 to 255 is required

VIP Suite: Color Space Conversion

MegaWizard Plug-In Manager - CSC

CSC
Version 7.1

About Documentation

1 Parameter Settings 2 Simulation Model 3 Summary

General Operands

Compile Time Operands

Color model conversion : Y'CbCr: HDTV to Computer R'G'B'

Din and dout refer to the input and output channels respectively.
 $dout_0 = (A0 * din_0) + (B0 * din_1) + (C0 * din_2) + S0$
 $dout_1 = (A1 * din_0) + (B1 * din_1) + (C1 * din_2) + S1$
 $dout_2 = (A2 * din_0) + (B2 * din_1) + (C2 * din_2) + S2$

Key
Desired Value
Actual Value

Coefficients

	A	B	C
0	1.164	0.0	1.793
	1.1640625	0.0	1.79296875
1	1.164	-0.213	-0.534
	1.1640625	0.0	0.0
2	1.164	2.115	0.0
	1.1640625	1.99609375	0.0

☐ Signed Integer bits : 1 Bits

The total coefficient length is 9 bits

Summands

	S
0	-992.512
	0.0
1	307.968
	255.99609375
2	-1157.3760000000002
	0.0

☐ Signed Integer bits : 8 Bits

The total summand length is 16 bits

Coefficient and summand fraction bits : 8 Bits

Cancel < Back Next > Finish

Specify the integer bits and signed option to get higher co-efficient precision

VIP Suite: 2D-FIR

- Various video/image processing signal chains have to filter the input signals to
 - Remove noise
 - Smoothen the image
 - Sharpen the image
 - Implement custom filtering
- The 2D-FIR filter function implements various custom filtering efficiently and quickly
- This core performs 2D convolution, using matrices of 3×3, 5×5, and 7×7 constant coefficients
- With suitable coefficients, the core can perform several operations including, but not limited to: sharpening, smoothing, and edge detection

VIP Suite: 2D-FIR

- An output pixel is calculated by
 - Multiplying input pixels in the kernel by the corresponding coefficient
 - Summing the values
- The output pixel value
 - Has its fractional bits removed
 - Is constrained to the output bit range
- The position of the output pixel corresponds to the mid point of the kernel

FIR Filter 2D - my_alt_vip_fir

FIR Filter 2D

Parameter Settings | Coefficients

Image Format

Image width: 720 Pixels
Image height: 576 Pixels

Color Plane Configuration

Number of color planes in sequence: 3 Planes

Input Data Type

Bits per pixel per color plane: 8 Bits
Data type: Unsigned
☐ Guard bands Max: 0 Min: 0

Output Data Type

Bits per pixel per color plane: 8 Bits
Data type: Unsigned
☐ Guard bands Max: 0 Min: 0

Result to Output Data Type Conversion Procedure

1) The results are in the range -510.00 to 510.00 (to 2 decimal places)
The results have 10 fraction bits
Move binary point right: 1 Places

2) The scaled results are in the range -510.00 to 510.00 (to 2 decimal places)
The scaled results have 10 fraction bits
Remove fraction bits by: Truncate values to integer

3) The scaled, integer results of the are in the range -510 to 510
The selected output data type is unsigned
Convert from signed to unsigned by: Replacing negative with absolute value

4) The scaled, integer, sign handled results are in the range 0 to 255
The selected output data type has a range of 0 to 255
The results will be saturated to the minimum and maximum values of the output data type

Cancel < Back Next > Finish

VIP Suite: 2D Median Filter

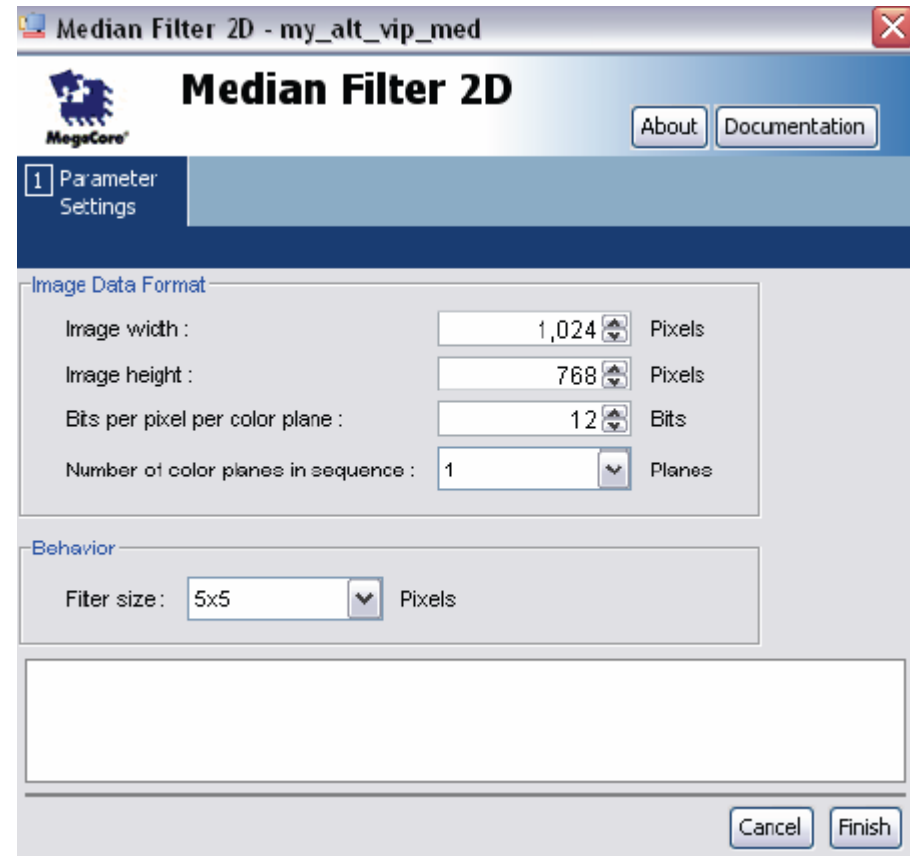
- Noise gets introduced into video data set via any electrical system used for storage, transmission, and/or processing

Median Filtering Is a Simple and Very Effective Noise Removal Filtering Process

- Median filtering:
 - Each pixel is determined by the median value of all pixels in a selected neighborhood (mask, template, window)
 - The median value **m** of a population (set of pixels in a neighborhood) is that value in which half of the population has smaller values than **m**, and the other half has larger values than **m**

VIP Suite: 2D Median Filter

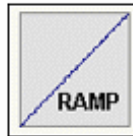
- The 2D median filter function provides a means to perform 2D median filtering operations using matrices of 3x3, 5x5, or 7x7 kernels
- Each output pixel is the median of the input pixels found in a 3x3, 5x5, or 7x7 kernel centered on the corresponding input pixel
- Larger kernel sizes require many more comparisons to perform the median filtering function; they also require correspondingly large increases in the number of logic elements used
- Larger sizes have a stronger effect, removing more noise but also potentially removing more detail



VIP Suite: Gamma Correction



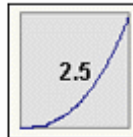
Sample Input to Monitor



Graph of Input

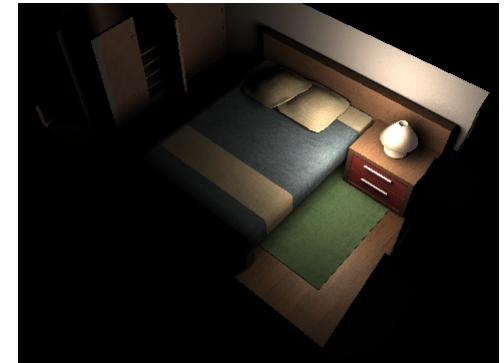


Output from Monitor

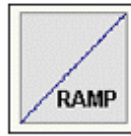


Graph of Output $L = V ^ 2.5$

NO Gamma Correction



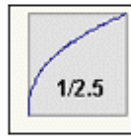
Sample Input



Graph of Input



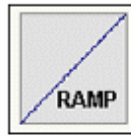
Gamma Corrected Input



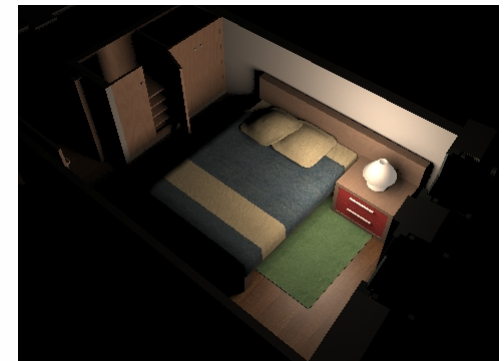
Graph of Correction $L' = L ^ (1/2.5)$



Monitor Output



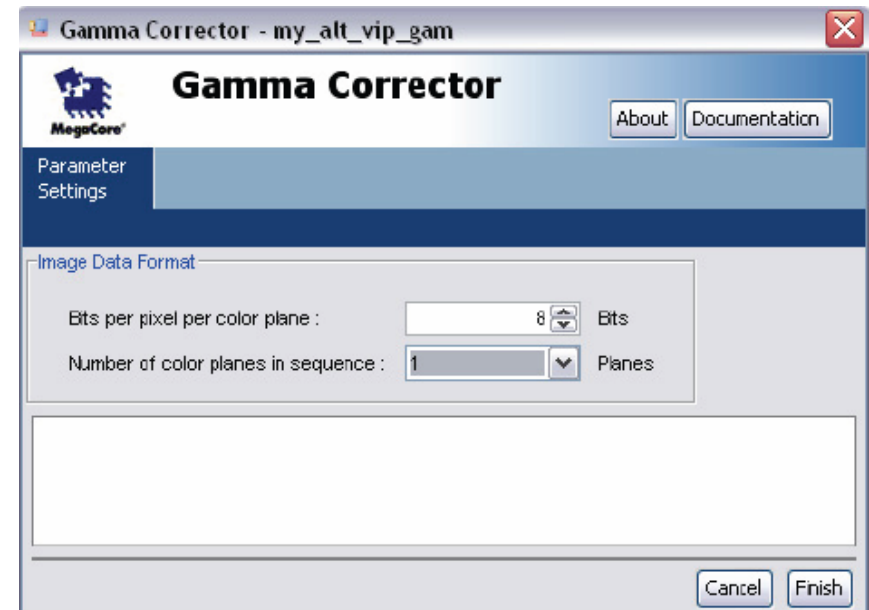
Graph of Output



Gamma Correction 2.5

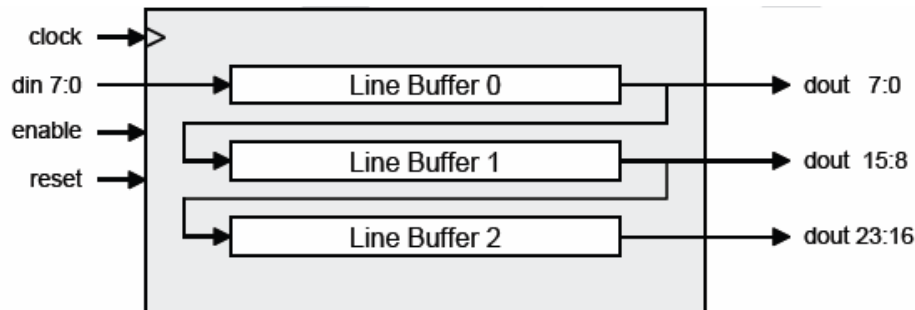
VIP Suite: Gamma Correction

- The gamma corrector function provides a look-up table (LUT) accessed through an Avalon-MM slave port
- The gamma values can be entered into the LUT by external hardware using this interface



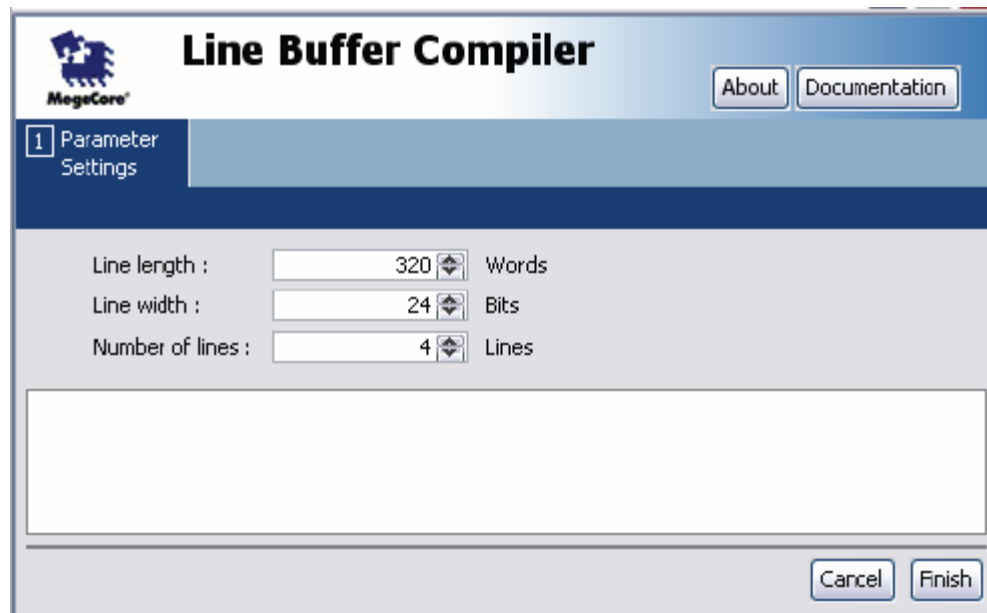
VIP Suite: Line Buffer Compiler

- FPGA memory is a valuable resource for many video and imaging applications
 - Particularly when developing HD systems and implementing high order accuracy algorithm
- The Line Buffer Compiler provides an efficient means to map line buffers on to Altera on-chip memories

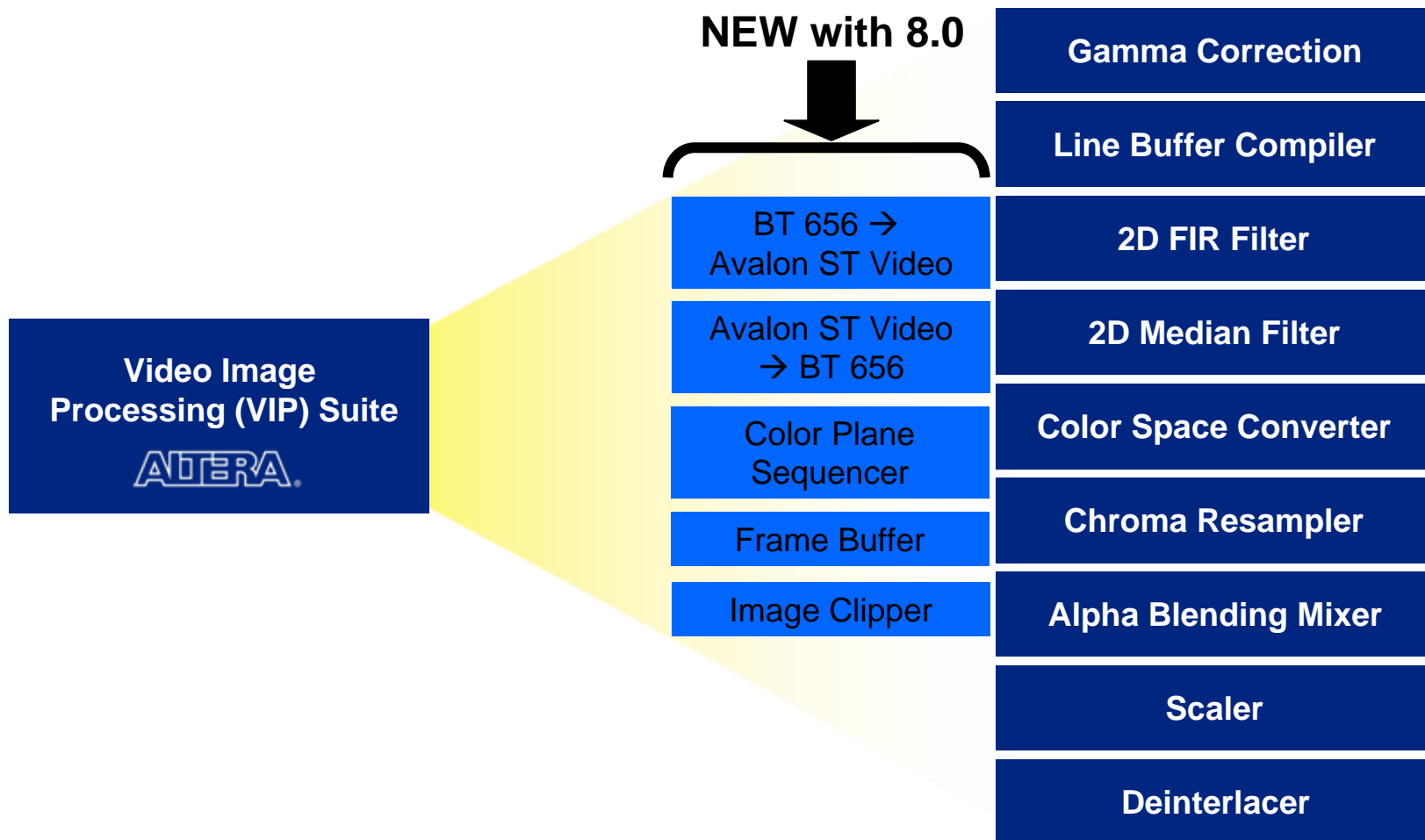


VIP Suite: Line Buffer Compiler

- To parameterize your Line Buffer Compiler function for a set of four line buffers each capable of holding 320 24-bit words...



Video Image Processing (VIP)



Suite of building block IP for video processing

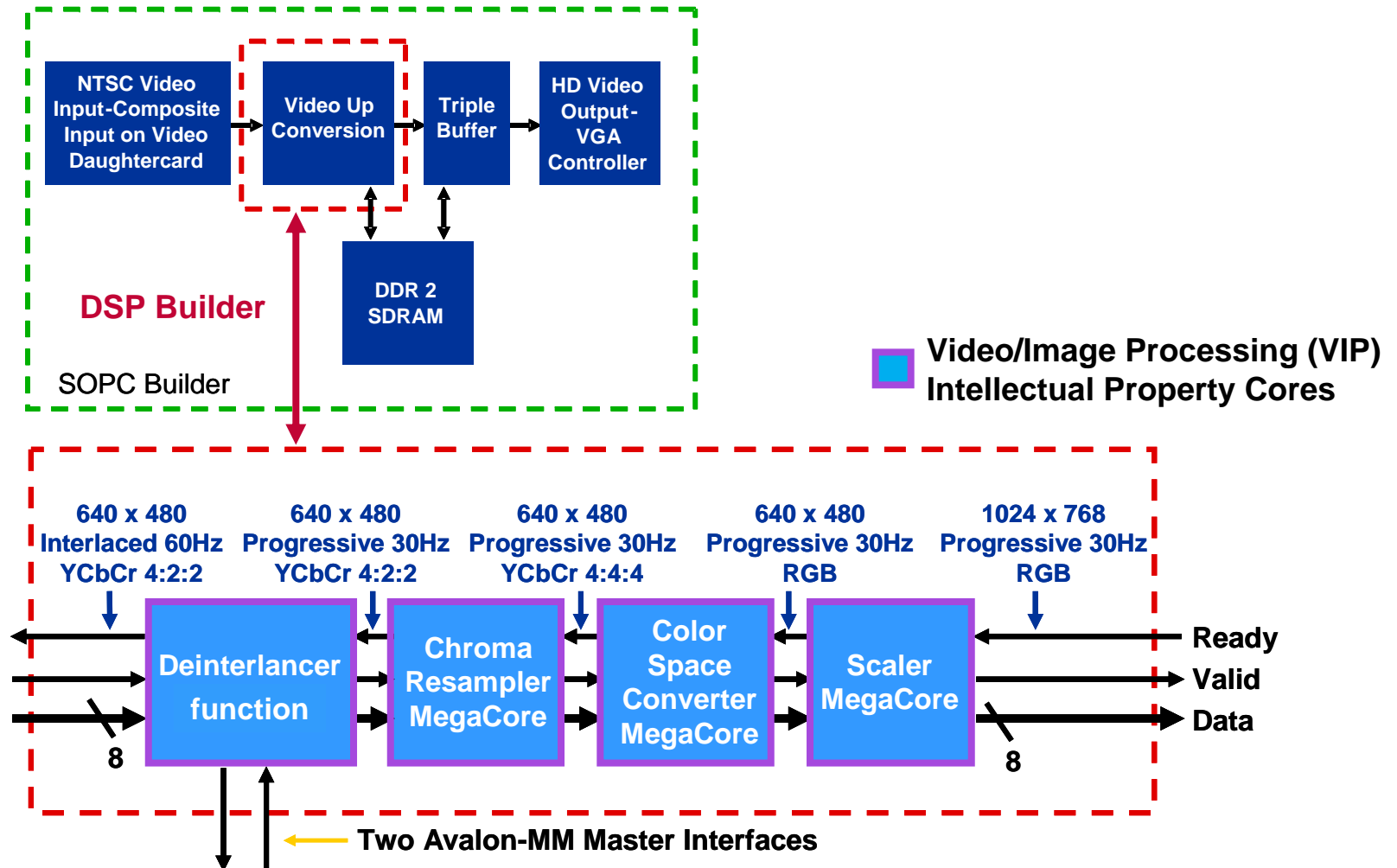


think **AND** not **OR**

**VIP Resources Available from
Altera**

Altera @ 40 nm

Altera Video Reference Design




















<http://www.altera.com/end-markets/refdesigns/sys-sol/broadcast/ref-post-processing.html>

Video Design Examples

DSP Design Examples

Table 1 contains digital signal processing (DSP) design examples for use in designs for Altera® devices. To see the design example, choose the corresponding icon in the Design Entry Method column.

<i>Table 1. DSP Design Examples—Functions and Design Entry Methods</i>	
Function	Design Entry Method
Deinterlacer Using Weave Mode 	
Deinterlacer Using Bob Mode 	
Gamma Correction 	
YCbCr to RGB Color Space Conversion 	
Image Frame Resizing Using Scaler 	
Salt and Pepper Noise Removal Using 2D Median Filter 	
Video Picture in Picture (PIP) Mixing Using Alpha Blending Mixer 	
Chroma Resampling Upconversion	
2D Sharpening Finite Impulse Response (FIR) Filter 	

<http://www.altera.com/support/examples/dsp/exm-dsp.html>

Cyclone III Video Kit



- Altera EP3C120F780 development board
- Bitec HSMC Quad Video daughter card
 - 8 composite or 4 s-video inputs
 - 1 HD (1080p) DVI Output port or
 - 1 TV (PAL/NTSC) output with resolutions to 1024x768 and support for composite, s-video or SCART (RGB) outputs
- Bitec HSMC DVI daughter card
 - 1 HD (1080p) DVI Output port (HDMI with external adaptor)
 - 1 HD (1080p) DVI Input port (HDMI with external adaptor)
- Interfaces directly to the Altera Video and Image Processing (VIP) Suite

http://www.bitec.ltd.uk/ciii_video_dev_kit.html

Stratix II GX Video Kit



- Stratix II GX video development board with a 2SGX90
- Video interfaces
 - Digital Video Interface (DVI) inputs/outputs
 - Four (4) standard definition (SD)/high definition (HD) SDI inputs/outputs, including Dual-Link SDI support
 - Asynchronous Serial Interface (ASI) inputs/outputs
- Audio interfaces
 - AES3
 - Sony/Phillips digital interface (S/PDIF)
- External memory
 - DDR2 DIMM (72 bit at 266 MHz)
 - 2-Mbyte SRAM
 - 16-Mbyte flash (configuration)

<http://www.altera.com/products/devkits/altera/kit-dsp-professional.html>

Summary

- Altera's video and image processing solution
 - Allows you to focus on your core competency
 - Eliminates the need to design, test, and debug standard video IP
 - Speeds up your development cycle
- Altera offers a complete programmable solution for your video application
 - IP (VIP Suite)
 - Development kits
 - Reference designs
 - Training