EclaireXL - Feature #85

720p support/1080i support

12/20/2019 09:41 PM - foft

Status:	In Progress	**Start date:**	12/20/2019
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Target version:			

Description

Implement 720p60 and 720p50 support, with suitable scaling

History

#1 - 12/20/2019 09:43 PM - foft

- Status changed from New to In Progress

HDMI syncs with 74.25MHz pixel clock. Cyclone V seems surprisingly happy with 742MHz gpio outputs via DDR registers.

#2 - 12/20/2019 09:43 PM - foft

Experimented with filter based scaling. Going with an area based scaler for now instead.

#3 - 12/20/2019 09:50 PM - foft

- File 723978_FULLTEXT01.pdf added

Great paper on scaling algos

#4 - 12/20/2019 09:52 PM - foft

I wrote some octave scripts to try these out. Plugging mister filters in for one set and for the other set doing area scaling with a bunch of different optimization options.

#5 - 12/20/2019 09:53 PM - foft

All the methods seem to need a bunch of multipliers, but fortunately we have some on Cyclone V. 25 DSP blocks on A2 and 66 DSP blocks on A4.
Each DSP block can be 3 9x9, 2 18x18 or 1 27x27 multiplier.
Only using a few already so they are mostly free.

#6 - 12/20/2019 09:56 PM - foft

I guess 1080i is also possible.
Want to note that for 720p60 I need to adjust the frame rate again. Was using 59.94 for the 27MHz option and here its 60.

I wonder if I can get the fractional pll to give an accurate enough 27MHz and 74.25MHz at the same time?

#7 - 12/25/2019 10:31 PM - foft

I implemented winscale and am debugging it!

It seems to work except I have some vertical stripes. I think this is because its actually down-scaling for the x axis (4x 1/2 colour clock width to 1280). While the attached paper says I can downscale up to 50% I think this is incorrect! Since the target pixel will overlap 3 source pixels even if the target pixel is even slightly wider than the source pixel.

#8 - 12/30/2019 08:36 PM - foft

- Subject changed from 720p support to 720p support/1080i support

#9 - 12/30/2019 09:11 PM - foft

The scaler is looking very nice now, a few border/locations bugs to fix.

Looking at the plls, need to have: 27MHz, 135MHz, 74.25MHz and 371.25MHz at the same time. This looks to be possible by feeding 54MHz from the USB pll to the HDMI pll.

It looks like 74.25 will also work for 1080i, so may as well enable that too.

Speaking of other modes, with these two clocks can also support:
1440x576i@50 (also 288 option, but not quite 50Hz)
1440x480i@59.94 (also 240 option, but not quite 59.94Hz)

These are probably the best fit of all, for 4x colour clock modes :-)
For 720p, 480p, 576p only sampling every other pixel at the moment, since can't downscale...
For 1080i and 1440x its either upscaling or exact.
Not that there is any software for 2x or 4x anyway!

#10 - 12/30/2019 10:45 PM - foft

I put in the modelines for 1080i. I need to add interlace support (different vtotal, mid line vsync etc) to the hcnt/vcnt logic. Probably will rewrite it since its not in my style at the moment (_next, _reg, integer based etc).

#11 - 01/05/2020 11:02 PM - foft

Plumbed in a 4x4 block of pixels so I can try out polyphasic filters too.

For now its just using a 2x2 block with the area based scale, but got some vertical bug. Stripey!

Using 16KB of block ram now, which is a bit wasteful. 360*4 pixels/line, 8 lines = 11.25KB. However, addressing is tricker. So using 2048 bytes/line.

#12 - 01/05/2020 11:03 PM - foft

Forgot to mention 1080i working fine as well as 720p.

I was looking at the 3d modes too, but not sure what to use those for ;-)

#13 - 01/15/2020 10:26 PM - foft

I have a polyphasic version working too, I think. So far I only gave it nearest neighbour coefficients. Rebuilding with Lanczos.

#14 - 01/21/2020 09:30 PM - foft

A weeks debugging later, this filter actually really works!

#15 - 01/26/2020 09:58 PM - foft

Adding the i2c wiring to allow these to be controlled from the firmware. So far areascale wired up and working. Now doing crtc, then will do polyphasic. I guess its possible to include both on A4 FPGAs and only one of them on A2 FPGAs.

On that subject I really should try to reduce the resource usage in general, I think some DSP blocks are not shared that should be. That can wait though until its all plumbed in.

#16 - 01/29/2020 10:13 PM - foft

Cleaned up the i2c wiring to be more generic. crtc wiring working too. Can switch from 720p50 and 1080i50 in firmware now (manual code, not menu yet...). Just wired up polyphasic too, except filter params which are currently hardcoded lanczos. Will come back to that.

Next up... clock switching and better plumbing into firmware. So can select 480p/576p,720p and 1080i from a nice menu.

#17 - 02/02/2020 10:30 AM - foft

Clock switching is proving fun, due to a bunch of constraints!

1st board: 50MHz on H16 (CLK11p) - which is connected to FPLL X0_Y38,X54_Y38, but not the other two
2nd board: : 50MHz on H16 (CLK11p) - which is connected to FPLL X0_Y38,X54_Y38, but not the other two AND N16 (clk6p) /H13 (clk10p) to clkgen chip. clk10 is connected to the same plls as clk11, so does not add much (except another input freq). clk6p is just to a 3rd pll X54_y1. None of them go directly to x0_y1.
mini board: 50MHz on H16 (CLK11p) - which is connected to FPLL X0_Y38,X54_Y38, but not the other two

Anyway its possible to route a pll output to the other 2 plls via the global clock network. Though running into some clock network issues, I'm using lots of clkctrl blocks already.

#18 - 02/02/2020 10:38 AM - foft

It seems pretty clear that on the 2nd board can use the reconfigurable clock to provide 27MHz and 74.25MHz hdmi clocks, by reconfiguring it. These can drive clk6p into the 3rd pll, which does x5 for the tmds clk (/2).

For the others... I've not come up with the magic source yet but I'm hopeful!

#19 - 02/09/2020 10:01 AM - foft

The dev of another hdmi library posted it on hacker news. Worth a look.
https://github.com/hdl-util/hdmi

#20 - 02/09/2020 09:25 PM - foft

I have all these modes working properly from the firmware.
NTSC:480i/480p/720p/1080i
PAL:576i/576p/720p/1080i
All in 4:3
480p/576p and 720p all skip a pixel, so 4x gr.0 or gr.8 isn't great.
480i and 1080i do not! Yes, 480i is better than 480p, because its 1440x480i and its 720x480p.

Next up I want to add scalar selection (since we have polyphasic and area) and start to store these in the settings or flash somewhere. All the crtc and scalar settings take up block that is best not wasted.

#21 - 02/09/2020 09:27 PM - foft

Oh and the clkgen chip can drive the video too, for those who want to try custom modes. I've not tried it yet since its statically set up at 30MHz iirc, but should give it a spin.

#22 - 02/13/2020 10:24 PM - foft

I have the crtc/scaler settings now in the flash chip (except scaler filter).
Upside: doesn't waste space in firmware
Downside: people will need to flash with usb blaster, rpd method will brick the video.

#23 - 02/23/2020 10:23 PM - foft

Got this merged down to svn and also v1 and v3 building.

Was hanging on purely internal i2c. Changed master and slave (+glue) to use in/wen instead of inout and working better. Slightly corrupt output on v1. I think this might be due to extra mux on the highest frequency part, will rework that... This is only present on v1 due to pin sharing between hdmi and vga so won't impact the mini.

#24 - 02/26/2020 09:52 PM - foft

On v1 its all working, with 32K block ram as system ram. Once I put it to 64K block ram as system ram (which JUST fits) I never get to basic. After some playing with the logic analyzer (tricky, no block ram!) I can see that at C4F1 the CPU reads D0 (BNE), twice. With 64KB ram the stack pointer increments (wrong), with 32KB it doesn't (correct!).

#25 - 02/26/2020 09:54 PM - foft

I guess related to the opcinfo storage somehow. with 32k get this:
atari800core:atari800|cpu:cpu6502|cpu_65xx:cpu_6502_peter|altsyncram:Mux54_rtl_0|altsyncram_ag91:auto_generated|ALTSYNCRAM AUTO ROM Single Clock 256 1 -- -- yes no -- -- 256 256 1 -- -- 256 1 0 atari800core_eclaireXLv1.atari800core_eclaireXL0.rtl.mif M10K_X38_Y11_N0 Don't care New data New data Off No No - Unsupported Depth 1

#26 - 02/26/2020 10:03 PM - foft

with 64k
atari800core:atari800|cpu:cpu6502|cpu_65xx:cpu_6502_peter|altsyncram:Mux54_rtl_0|altsyncram_ag91:auto_generated|ALTSYNCRAM AUTO ROM Single Clock 256 1 -- -- yes no -- -- 256 256 1 -- -- 256 1 0 atari800core_eclaireXLv1.atari800core_eclaireXL0.rtl.mif M10K_X38_Y13_N0 Don't care New data New data Off No No - Unsupported Depth 1

#27 - 02/26/2020 10:04 PM - foft

M10K is shared:
atari800core:atari800|cpu:cpu6502|cpu_65xx:cpu_6502_peter|altsyncram:Mux54_rtl_0|altsyncram_ag91:auto_generated|ALTSYNCRAM AUTO ROM Single Clock 256 1 -- -- yes no -- -- 256 256 1 -- -- 256 1 0 atari800core_eclaireXLv1.atari800core_eclaireXL0.rtl.mif M10K_X38_Y13_N0 Don't care New data New data Off No No - Unsupported Depth 1
zpu_rom:zpu_rom1|altsyncram:altsyncram_component|altsyncram_od24:auto_generated|ALTSYNCRAM AUTO ROM Single Clock 10240 32 -- -- yes no -- -- 327680 10240 32 -- -- 327680 39 0 zpu_rom.mif M10K_X22_Y14_N0, M10K_X22_Y31_N0, M10K_X22_Y33_N0, M10K_X22_Y18_N0, M10K_X22_Y12_N0, M10K_X30_Y29_N0, M10K_X30_Y26_N0, M10K_X22_Y29_N0, M10K_X22_Y20_N0, M10K_X11_Y12_N0, M10K_X30_Y25_N0, M10K_X22_Y30_N0, M10K_X11_Y19_N0, M10K_X11_Y25_N0, M10K_X3_Y18_N0, M10K_X11_Y20_N0, M10K_X22_Y26_N0, M10K_X3_Y24_N0, M10K_X11_Y14_N0, M10K_X22_Y16_N0, M10K_X3_Y19_N0, M10K_X3_Y22_N0, M10K_X11_Y21_N0, M10K_X11_Y18_N0, M10K_X11_Y15_N0, M10K_X22_Y17_N0, M10K_X22_Y15_N0, M10K_X22_Y19_N0, M10K_X11_Y13_N0, M10K_X22_Y13_N0, M10K_X30_Y31_N0, M10K_X11_Y16_N0, M10K_X22_Y32_N0, M10K_X22_Y28_N0, M10K_X30_Y30_N0, M10K_X3_Y23_N0, M10K_X3_Y20_N0, M10K_X3_Y15_N0, M10K_X38_Y13_N0 Don't care New data New data Off No No - Address Too Wide 2

#28 - 02/26/2020 10:28 PM - foft

with 32k, the block is not shared. I guess something is accessing memory where it shouldn't...

#29 - 02/26/2020 10:57 PM - foft

Its opcodeInfoTable that is getting put into altsyncram. Not sure why its invalid yet. I'm trying the ramstyle attribute but didn't find the right voodoo yet...

#30 - 02/27/2020 10:28 PM - foft

Problem is with E8:
002020100112
bit 20 == high = opcStackUp.

This isn't what is in the constant table, so not clear why quartus is returning that.

Combined with 32KB, so no overlap I see:
002020000112

Where does that bit error come from?

#31 - 02/27/2020 11:10 PM - foft

Tried to upgrade from Quartus 18.0 to 19.1. Same issue.

#32 - 03/01/2020 10:32 PM - foft

- File problem.bmp added

- File problem_not_shared.bmp added

Checked on the ram block input/output when built with 32KB and 64KB.
Same input, same mif, different output...
Check the output of ram_block_1a0 vs the input (myNextOpCode[1-7] and nextOpcInfo[42])
With block shared:
  problem.bmp
With block by itself:
  problem_not_shared.bmp

#33 - 01/14/2021 10:36 AM - foft

There is a new version of Quartus now, I'll try it to see if Intel fixed this bug yet.

#34 - 01/14/2021 10:29 PM - foft

Quartus 20.1.1 still has the bug. Raised with Intel.

Files

723978_FULLTEXT01.pdf	8.72 MB	12/20/2019	foft
problem_not_shared.bmp	1.53 MB	03/01/2020	foft
problem.bmp	2.12 MB	03/01/2020	foft