High-Performance Correlation Coefficient Calculator

Marcos Vinicius Silva Oliveira Nikita Lazarev Robin Malzac



Summary

Context
Applications
Impmementation
Verification
Performance analysis
Conclusions

Applications

- Radar system
- Detect water leak
- Web filtering
- Risk management
- Telescope image analysis



Original r = 1



Flip r = - 0.057



r = -0.014



S & P (0.02) r = 0.908



Resolution / 2 r = 0.956



S & P (0.2) r = 0.521





r = -0.083

Math & SW vs HW distribution

$$r = \frac{\left[\sum_{m}\sum_{n}(A_{mn} - \overline{A})(B_{mn} - \overline{B})\right]}{\sqrt{\left[\left(\sum_{m}\sum_{n}(A_{mn} - \overline{A})^{2}\right)\right]\left[\left(\sum_{m}\sum_{n}(B_{mn} - \overline{B})^{2}\right)\right]}}$$

$$x = z$$
where $\overline{A} = \text{mean2(A)}$, and $\overline{B} = \text{mean2(B)}$.

SW vs HW distribution



r = Y/sqrt(X*Z)

Corr2 IP interfaces



Custom Arithmetic Logic block simplifications



Custom Arithmetic Logic block



Corr2 IP

 Fifo A receives reference image (A)

 After is loaded, the feedback path is enabled

 Accumulator calculates the sum of elements of the incoming image

•Fifo B receives image to be compared againt A.

 The content is stored until all the elements have been summed up

 4 instances of the CALB block proccess 4 pixels simuntaneously



Corr2 IP Data flow timeline

 Perfomance boost thanks to:

 DMA access to memory

 Reading image A only once for all coeficient calculations

 Loading data, calculating sum, calculating Y, calculating Z, simultaneously

 Processing 4 pixels at the same time







uma_addr).32') 32'h00	001000 (32'h	00001010	(32'h0000102	0) 32'h00001	J30	(32'h00001040) 32'h00001050		32'h00001060	(32'h00001070	
dma_irq	L	1		-								
dma_rx_cnt)11'h000 ()())	₽0000000000000000000000000000000000000	000000000000000.11	000000000	000000000000000000000000000000000000000))()))()11'h010))))(.11'h010		000000000000000000000000000000000000000	()((11'h010
dma_rx_d	٦ <u>ــــــــــــــــــــــــــــــــــــ</u>	1	tuuuuuu		խասա_աստափ			tu_uuuuut				
dma_s	waiting (.) tran	sfering (.)(tr	ansfering ()	(.) transfering) . (transferi	ng ((w)	(.) transfering) .) transfering	. ((waiting)	.) transfering	() transfering)) waiting
FIFOs		İ	+ +		i ‡			ţţ				[]
fifoA_clock		<u>†</u>			, a de la company de la com	<u>, and a second</u>		, and a second second				
fifoA_data	(32'h00	100000000000000000000000000000000000000		<u>)</u>	2'h01010101)	D1010101			1010101	<u> </u>
fifoA_q	(32'h00) 32	h01010101			2'h01010101)) (32'h	D1010101	•		1010101	
fifoA_rdreq	1	Ť	-					t t			[]	
fifoA_sclr	1		-								[]	
fifoA_usedw	(14'h0000))))))		000000000000000000000000000000000000000	10020								<u> </u>
fifoA_wrreq	۰ ــــــ ۱۸۸۷	1	touuuuuu t									
fifoB_clock		المحمد ال <u>محم</u> ة			أستتناكر والأ		Calc	ulatino		Cal	culatin	
fifoB_data).32'h00	100000000000000000000000000000000000000			jaalahaka jalahakakaka jalahakakakaka jalahakakakaka jalahakaka jalahakakakakaka jalahakakakakaka jalahakakakak	######.32'h0			00	_		S
fifoB_q	(32'h00000000			V 22'50101	0101		Z.Y to	or imac		Z.Y f	or ima	
fifoB_rdreq	٦		•	Calcu	lating	X _	_,			,		3
fifoB_sclr	٦	İ	•	Calco	liating			В			С	
fifoB_usedw	(14'h0000	-	• • • • • • • • • • • • • • • • • • •	000000000	200000	<u>)()))()).14'h</u>			0020			20
fifoB_wrreq												
fifo_usedw	(<u>14'h0000</u>)))))))		000000000000000000000000000000000000000	X	00000-000000-	<u> </u>	D##00%15000000	00)()(14'h0020		X	<u>v()</u> 14'h0020
CALB		+						İ			[]	
Amn	(32'h00) 32	h01010101			2'h01010101)	01010101		32'h	1010101	
Bmn	(32'h00)(32	h01010101			2'h01010101)	04030201		32'h	8070605	
XZ	({32) ({32'h	t	000000		32'h00000AB0}		32'	h00000AB0} {32'h	00000AL }	(00)	+	
		00000000} {32'h0	000000}							<u> (321</u>	00000AF0} {32'h	WUUUUADU)
(0)) (32'h0	00000000} {32'h0 0000000			32'h00000AB0) () () () () () () () () () () () () () () () () () ()	00000AB0		(<u>32'</u> n) (32'h)	00000AF0} {32'h	
(0) (1))	00000000} {32'h0 0000000 0000000			32'h00000AB0 32'h00000AB0) 32'l) 1111 (32'l	00000AB0 00000AB0		(32'h) (32'h) (32'h) (32'h) (32'h)	00000AF0} {32'h 00000AF0 00000AB0	
(0) (1) (2)) (32'h0) (32'h0) (32'h0	00000000) {32'h0 0000000 0000000 0000000			32'h00000AB0 32'h00000AB0 32'h00000AB0)))))))))))))))))))	00000AB0 00000AB0 00000AB0		(32'h (32'h (32'h) (32'h) (32'h) (32'h) (32'h)	00000AF0} {32'h 00000AF0 00000AB0 00000AB0	
(0) (1) (2) (3)) (22 m)) (32 h0) (32 h0)) (32 h0) (32 h0)) (32 h	00000000} {32'h0 0000000 0000000 0000000 0000000			2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0)	00000AB0 00000AB0 00000AB0 00000AB0		(32'h 32'h 32'h 32'h 32'h 32'h	00000AF0} {32'h 00000AF0 00000AB0 00000AB0 00000AF0	
(0) (1) (2) (3) Y) (32'h0)) (32'h0)	00000000} {32'h0 0000000 0000000 0000000 00000000 00000	000000000000000000000000000000000000000		2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0		32'h 32'h 32'h 32'h 32'h 32'h 32'h 32'h	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 100000AB0} {321	00000AB/	(321 32'h 32'h 32'h 32'h 32'h 32'h (32'h	00000AF0} {32'h 00000AF0 00000AB0 00000AB0 00000AB0 00000AF0 000010A90} {32'h	10000AB0}
(0) (1) (2) (3) Y (0)) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0) (32'h0)	0000000) {32'h0 0000000 0000000 0000000 0000000 000000	000000000000000000000000000000000000000		22'h00000AB0 22'h00000AB0 22'h00000AB0 22'h00000AB0 32'h00000AB0 32'h00000AB0 22'h00000AB0 22'h00000AB0	Q	32'h 32'h 32'h 32'h 32'h 32'h 32'h 32'h	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB((321 32h 32h 32h 32h 32h 32h 32h 32h 32h 32h	00000AF0} (32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} (32'h 00010A90	0000AB0}
(0) (1) (2) (3) Y (0) (1) (2)	\	0000000) {32'h0 000000 0000000 0000000 0000000 000000	0000003		2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0 2'h00000AB0 2'h00000AB0			00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB/	(32) 32m 32m 32m 32m 32m 32m 32m 32m 32m 32m	00000AF0} (32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} (32'h 00010A90 00000AA0 00000AB0	00000A0)
(0) (1) (2) (3) Y (0) (1) (2) (3)	(12::::::::::::::::::::::::::::::::::::	00000000) {32'h0 0000000 0000000 0000000 00000000 00000	0000003		2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0)	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB((32) 32h 32h 32h 32h 32h 32h 32h 32h	00000AF0} (32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} (32'h0 00010A90 00000AA0 00000AB0 00000AC0	00000AD07.
(0) (1) (2) (3) Y (0) (1) (2) (3) calb_clr		0000000) {32'h0 0000000 0000000 0000000 00000000 {32'h0 0000000 0000000 0000000 0000000			2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0	X		00000AB0 00000AB0 00000AB0 00000AB0 100000AB0 00000AB0 00000AB0 00000AB0	00000AB	(321 321) 321) 321) 321) 321) (321) (321) 321) 321) 321) 321) 321) 321) 321)	00000AF0) {32'h 0000AF0 0000AB0 00000AB0 00000AF0 00010A90) {32'h 00010A90 00000AA0 00000AB0 00000AB0	00000AD07
(0) (1) (2) (3) Y (0) (1) (2) (3) calb_clr calb_crt		0000000) {32'h0 0000000 0000000 0000000 00000000 {32'h0 0000000 0000000 0000000 0000000			2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0	X	321 321 321 321 321 321 321 321 321 321	00000AB0 00000AB0 00000AB0 00000AB0 100000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB	(321 327) 327) 327) 327) (327) (327) 327) 327) 327) 327) 327) 327) 327)	00000AF0} {32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} {32'h 00010A90 00000AA0 00000AA0 00000AB0 00000AC0	
(0) (1) (2) (3) Y (0) (1) (2) (3) calb_clr calb_crt calb_ena	\\	00000000) {32'h0 0000000 0000000 0000000 00000000 00000			2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0		321 321 321 321 321 321 321 321 321 321	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB((321 32m 32m 32m 32m 32m 32m 32m 32m 32m 32m	00000AF0} (32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} (32'h(0010A90 00000AA0 00000AA0 00000AB0 00000AC0	
(0) (1) (2) (3) Y (0) (1) (2) (3) calb_clr calb_crt calb_ena mA	\\	00000000) {32'h0 0000000 0000000 0000000 00000000 00000			22h00000AB0 22h00000AB0 22h00000AB0 22h00000AB0 32h00000AB0 32h00000AB0 22h00000AB0 22h00000AB0 22h00000AB0		321 321 321 321 321 321 321 321 321 321	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB()	(321 32m 32m 32m 32m 32m 32m 32m 32m 32m 32m	00000AF0} (32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} (32'h(00010A90 00000AA0 00000AA0 00000AC0	
(0) (1) (2) (3) Y (0) (1) (2) (3) calb_clr calb_crt calb_ena mA mB		00000000) {32'h0 0000000 0000000 0000000 00000000 00000			22h00000AB0 22h00000AB0 22h00000AB0 22h00000AB0 32h00000AB0 22h00000AB0 22h00000AB0 22h00000AB0 22h00000AB0		321 321 321 321 321 321 321 321 321 321	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB()	(321 321) 321) 321) 321) 321) 321) 321) 3	00000AF0} {32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90} {32'h 00010A90 00000AA0 00000AA0 00000AB0 00000AC0	00000A0)
(0) (1) (2) (3) Y (0) (1) (2) (3) calb_clr calb_crt calb_ena mA mB		00000000) {32'h0 0000000 0000000 0000000 00000000 00000			2'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0 32'h00000AB0 32'h00000AB0 2'h00000AB0 2'h00000AB0 2'h00000AB0		321 321 321 321 321 321 321 321 321 321	00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0 00000AB0	00000AB (8'h12	(321 321) 321) 321) 321) 321) 321) (321) 321) 321) 321) 321) 321) 321) 321)	00000AF0} (32'h 00000AF0 00000AB0 00000AB0 00000AF0 00010A90 {32'h 00010A90 00000AA0 00000AA0 00000AC0	00000A0) 00000AA0)

Entity:corr2_tb_Architecture:tb_Date: Mon Jun 12 07:44:41 CEST 2017 Row: 1 Page: 2

** Note: Firmware Version: 0xABCD0001

ModelSim> run 15 us

Model Sim.

MATLAB^{*}

#		Time:	100 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	DMA Start Addr: 0x00001000
#		Time:	220 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	DMA End Addr: 0x00001010
#		Time:	260 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	DMA Burst Count: 0x00000010
#		Time:	300 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	DMA Fifo Threshold: 0x00003FFF
#		Time:	340 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Fifo / Calb register: 0x00000000
#		Time:	400 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	FIF0A Used words: 0x00000000
#		Time:	440 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	FIFOB Used words: 0x00000000
#		Time:	480 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Accumulator result: 0x00000840
#		Time:	2960 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	FIFOA Used words: 0x00000020
#		Time:	3 us Iteration: 1 Instance: /corr2_tb
#	**	Note:	FIFOB Used words: 0x00000000
#		Time:	3040 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Accumulator result: 0x00000840
#		Time:	5620 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	FIF0A Used words: 0x00000020
#		Time:	5660 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	FIFOB Used words: 0x00000020
#		Time:	5700 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Expected XZ[0]=0xAB0 Actual: 0x00000AB0
#		Time:	5760 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Expected XZ[1]=0xAB0 Actual: 0x00000AB0
#		Time:	5800 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Expected XZ[2]=0xAB0 Actual: 0x00000AB0
#		Time:	5840 ns Iteration: 1 Instance: /corr2_tb
#	**	Note:	Expected XZ[3]=0xAB0 Actual: 0x00000AB0
#		Time:	5880 ns Iteration: I Instance: /corr2_tb
#	**	Note:	Accumulator result: 0x00000900
#		Time:	8420 ns iteration: I instance: /corr2_tb
#	**	Note:	FIFUA Used Words: UXUUUUUUUUU
#		Time:	8460 ns iteration: I instance: /corr2_tb
#	**	Note:	FIFUB Used Words: UXUUUUUU2U
#		Time:	8500 ns Iteration: 1 Instance: /corr2_tb

** Note: Expected XZ[0]=0xAB0 | Actual: 0x00000AB0 Time: 8560 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected XZ[1]=0xAB0 | Actual: 0x00000AB0 Time: 8600 ns Iteration: 1 Instance: /corr2 th Note: Expected XZ[2]=0xAB0 | Actual: 0x00000AB0 Time: 8640 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected XZ[3]=0xAB0 | Actual: 0x00000AB0 Time: 8680 ns Iteration: 1 Instance: /corr2 th Note: Expected Y[0]=0xAB0 | Actual: 0x00000AB0 Time: 8720 ns Iteration: 1 Instance: /corr2 th Note: Expected Y[1]=0xAB0 | Actual: 0x00000AB0 8760 ns Iteration: 1 Instance: /corr2 th Time: ** Note: Expected Y[2]=0xAB0 | Actual: 0x00000AB0 Time: 8800 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected Y[3]=0xAB0 | Actual: 0x00000AB0 Time: 8840 ns Iteration: 1 Instance: /corr2 tb Note: Accumulator result: 0x00000B00 Time: 11380 ns Iteration: 1 Instance: /corr2 th FIFOA Used words: 0x00000020 Time: 11420 ns Iteration: 1 Instance: /corr2 tb Note: FIFOB Used words: 0x00000020 Time: 11460 ns Iteration: 1 Instance: /corr2 tb Note: Expected XZ[0]=0xAF0 | Actual: 0x00000AF0 Time: 11520 ns Iteration: 1 Instance: /corr2 th Note: Expected XZ[1]=0xAB0 | Actual: 0x00000AB0 Time: 11560 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected XZ[2]=0xAB0 | Actual: 0x00000AB0 Time: 11600 ns Iteration: 1 Instance: /corr2 tb Note: Expected XZ[3]=0xAF0 | Actual: 0x00000AF0 Time: 11640 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected Y[0]=0xA90 | Actual: 0x00010A90 Time: 11680 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected Y[1]=0xAA0 | Actual: 0x00000AA0 Time: 11720 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected Y[2]=0xAB0 | Actual: 0x00000AB0 Time: 11760 ns Iteration: 1 Instance: /corr2 tb ** Note: Expected Y[3]=0xAC0 | Actual: 0x00000AC0 Time: 11800 ns Iteration: 1 Instance:

Top level block diagram

Altera 28-nm Cyclone V 5CSEMA5F31C6N FPGA



Software structure



Performance measurements



Time for loading 32K pixel image into the FIFO vs burst count Not worth using DMA bursts transfers longer than 64 words (for the SoC being investigated in this study)

Performance measurements

32-bit words	Burst count	Time for software (s)	Time for hw (s)	Gain
1	1	0.001363	0.000482	2.827801
2	2	0.001793	0.000494	3.629555
4	4	0.002437	0.000487	5.004107
8	8	0.004009	0.000494	8.115385
16	16	0.006943	0.000496	13.997984
32	32	0.012623	0.000494	25.552632
64	64	0.024511	0.000513	47.779727
128	64	0.046164	0.000542	85.173432
256	64	0.091527	0.000578	158.351211
512	64	0.180916	0.000620	291.800000
1024	64	0.360766	0.000739	488.181326
2048	64	0.708648	0.001019	695.434740
4096	64	1.42377	0.001529	931.177240
8192	64	2.864359	0.002523	1135.298851
16320	64	5.818144	0.004555	1277.309330

The corr2 ip always outperforms the processor-only solution

Performance measurements



Conclusions

- There is a gain of 5X in loading 32K pixel images using DMA burst transfer of 64 words instead of single cycle transfers.
- The corr2 ip always outperforms the processor-only solution for any image size, including the minimum size of 4 pixels.
- The gain increases logaritmically for image sizes between 64 and 4K pixels, ranging from 14X to 488X respectively.
- The gain reaches 1277X for images with size of 64K pixels, which corresponds to the FIFO limit.
- The study presented here shows that the use of hardware aceleration is very interesting for image processing algorithms such as 2D correlation.

