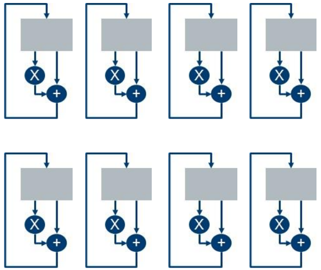
HIGH PERFORMANCE QRD FOR FPGAS

Martin Langhammer
Bogdan Pasca

FPGA2018 – February 27, 2018

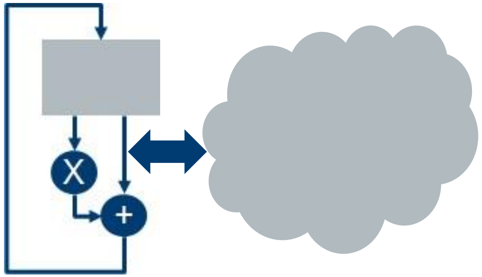
Implementation Targets

GPU



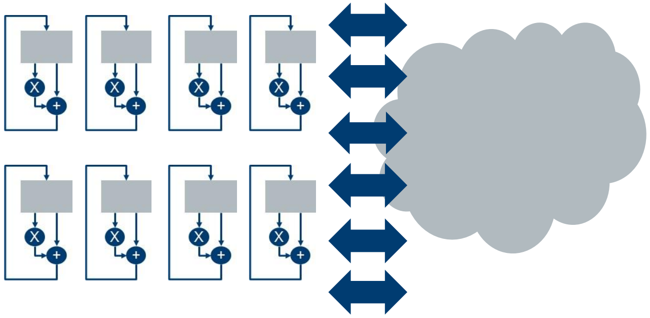
Large number of simple FP load-store units.
If you can map your algorithm to these (SGEMM) it will be fast

CPU



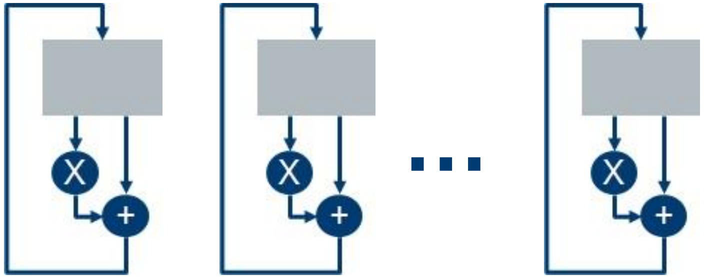
Small number of complex load-store units.
If your algorithm doesn't map directly, there is likely to be a special instruction that can help you

FPGA

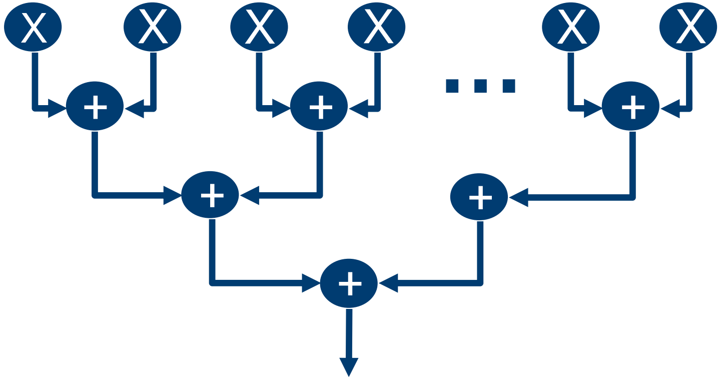


Large number of *configurable* FP load-store units, same TFLOPs as GPU.
Cloud of *configurable* special instructions
Cloud of *configurable* routing for zero overhead data moves

FPGA FP Arithmetic Structures

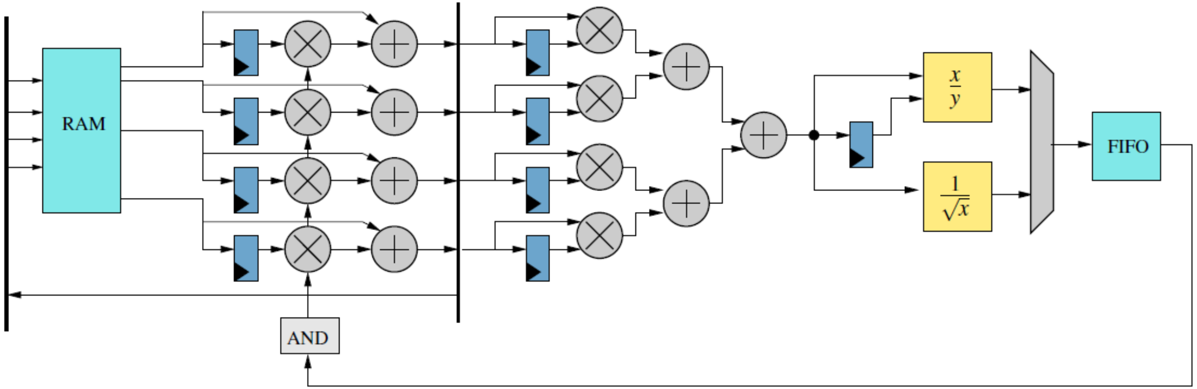


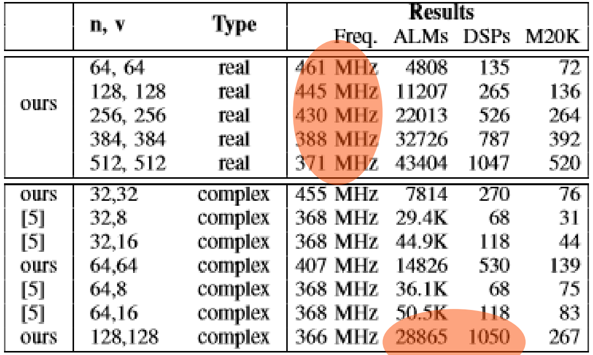
Scalar (plus load-store)



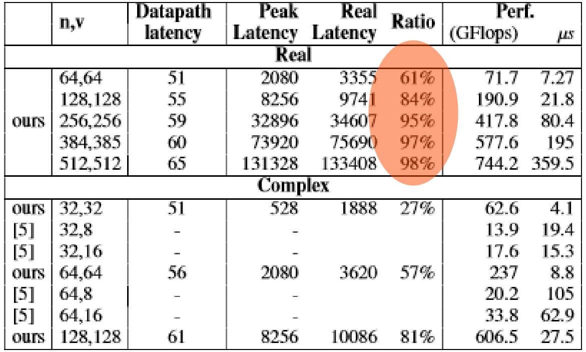
Vector

Results





	n, v	Type	Results			
			Freq.	ALMs	DSPs	M20K
ours	64, 64	real	461 MHz	4808	135	72
	128, 128	real	445 MHz	11207	265	136
	256, 256	real	430 MHz	22013	526	264
	384, 384	real	388 MHz	32726	787	392
	512, 512	real	371 MHz	43404	1047	520
ours	32,32	complex	455 MHz	7814	270	76
[5]	32,8	complex	368 MHz	29.4K	68	31
[5]	32,16	complex	368 MHz	44.9K	118	44
ours	64,64	complex	407 MHz	14826	530	139
[5]	64,8	complex	368 MHz	36.1K	68	75
[5]	64,16	complex	368 MHz	50.5K	118	83
ours	128,128	complex	366 MHz	28865	1050	267



	n,v	Datapath latency	Peak Latency	Real Latency	Ratio	Perf. (GFlops)	μs
Real							
ours	64,64	51	2080	3355	61%	71.7	7.27
	128,128	55	8256	9741	84%	190.9	21.8
	256,256	59	32896	34607	95%	417.8	80.4
	384,385	60	73920	75690	97%	577.6	195
	512,512	65	131328	133408	98%	744.2	359.5
Complex							
ours	32,32	51	528	1888	27%	62.6	4.1
[5]	32,8	-	-			13.9	19.4
[5]	32,16	-	-			17.6	15.3
ours	64,64	56	2080	3620	57%	237	8.8
[5]	64,8	-	-			20.2	105
[5]	64,16	-	-			33.8	62.9
ours	128,128	61	8256	10086	81%	606.5	27.5