Optimizing Your Flowgraphs for Hardware

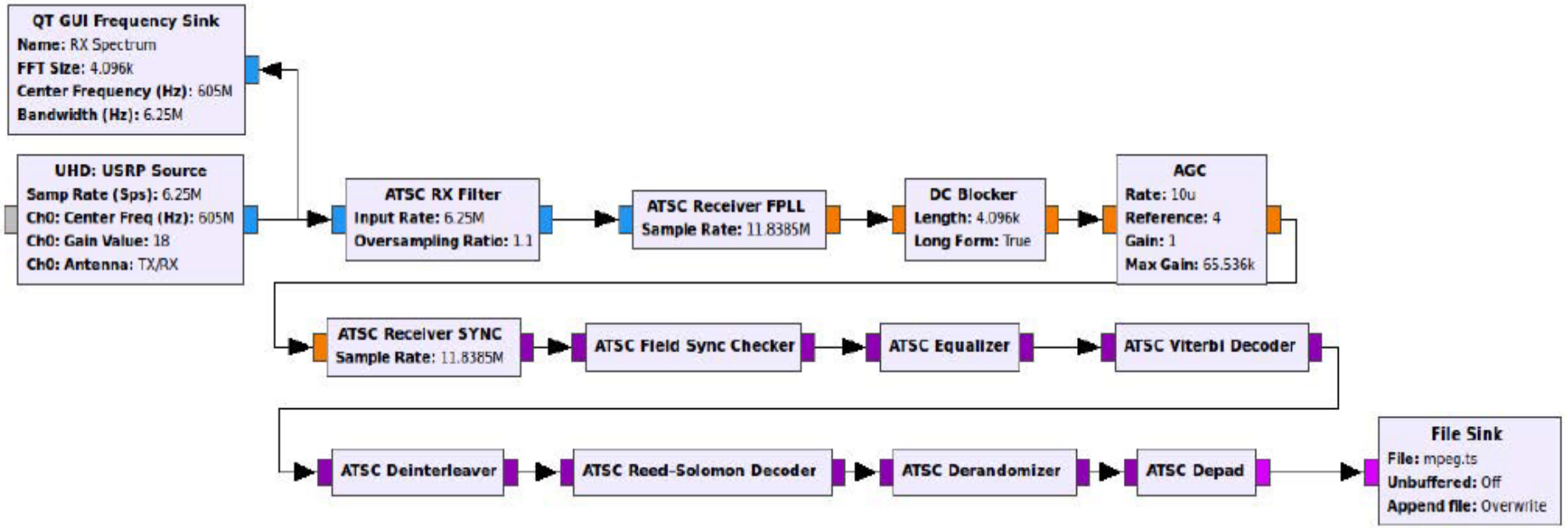
Presented by Greg Scallon

GNU Radio Conference 2017

Tuesday, September 12th 1:45pm

greg@simpleXecutive.com

ATSC PROJECT FLOW GRAPH



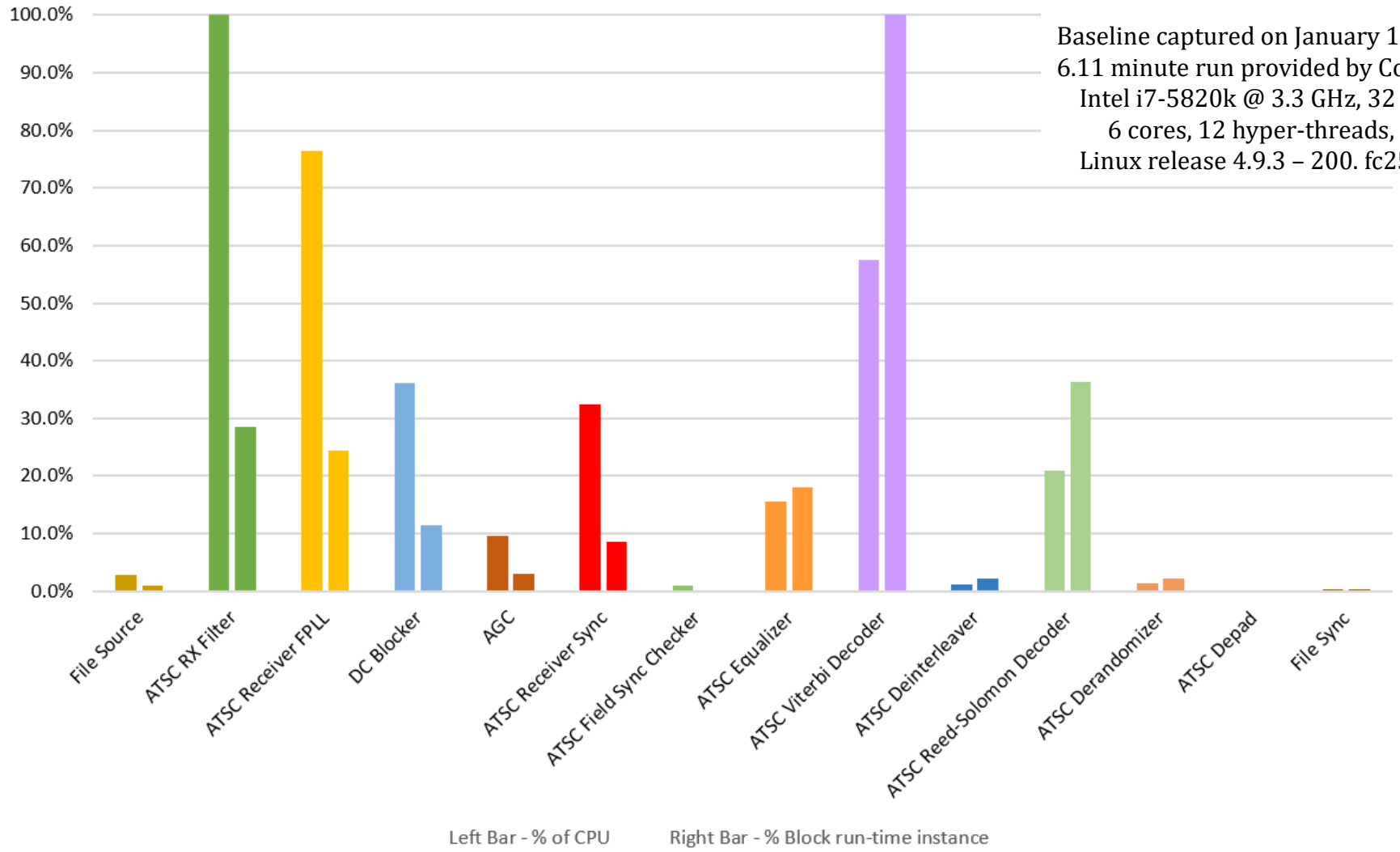
greg@simpleXecutive.com

ATSC MEASUREMENTS

(Baseline Run 1/19/17)						Derived Measurements						
Blockname Units	Average Runtime [ns]	Total Variance [sec]	Total [sec]	Graph #	ATSC Process Name (from ATSC Flowgraph)	Samples per Run	Rate / Sec	Period [us]	Load % GPU	Run [us]	Sigma % avg	Rate In:Out
file_source0	5941.657	0.03788	10.51	1	File Source	1768948	4823	207	2.87%	5.94	0.36%	1:1
pfb_arb_resampler_ccf0	185146.7	8.70045	361.1	2	ATSC RX Filter	1950357	5318	188	**98.45%**	185.15	2.41%	1:1
dtv_atsc_fpll0	157430.1	3.1668	275.9	3	ATSC Receiver FPLL	1752745	4779	209	75.23%	157.43	1.15%	1:1
dc_blocker_ff0	73740.83	1.0931	130.2	4	DC Blocker	1766105	4815	208	35.51%	73.74	0.84%	1:1
agc_ff0	19470.3	0.05124	34.54	5	AGC	1774098	4837	207	9.42%	19.47	0.15%	5:6
dtv_atsc_sync0	55546.68	0.76014	117.4	6	ATSC Receiver Sync	2114342	5765	173	32.02%	55.55	0.65%	1:1
dtv_atsc_fs_checker0	1556.729	0.00042	3.395	7	ATSC Field Sync Checker	**2180927**	**5946**	**168**	0.93%	1.56	0.01%	9:2
dtv_atsc_equalizer0	117049.4	0.803	56.35	8	ATSC Equalizer	481400	1313	762	15.36%	117.05	1.43%	3:2
dtv_atsc_viterbi_decoder0	647083.9	5.82213	207.7	9	ATSC Viterbi Decoder	320989	875	1143	56.63%	**647.08**	**2.80%**	1:1
atsc_deinterleaver0	13945.68	0.01538	4.477	10	ATSC Deinterleaver	321011	875	1143	1.22%	13.95	0.34%	1:1
dtv_atsc_rs_decoder0	235631.4	1.96019	75.63	11	ATSC Reed-Solomon Decoder	320987	875	1143	20.62%	235.63	2.59%	1:1
dtv_atsc_derandomizer0	14310.8	0.00624	4.594	12	ATSC Derandomizer	320998	875	1143	1.25%	14.31	0.14%	1:1
atsc_depad0	916.2316	3.5E-05	0.294	13	ATSC Depad	321006	875	1143	0.08%	0.92	0.01%	1:1
file_sink0	2855.984	0.00384	0.917	14	File Sync	320992	875	1143	0.25%	2.86	0.42%	1:1
(366.772 second run)								Total Load	349.8%			
								Per GPU	29.15%			

greg@simpleXecutive.com

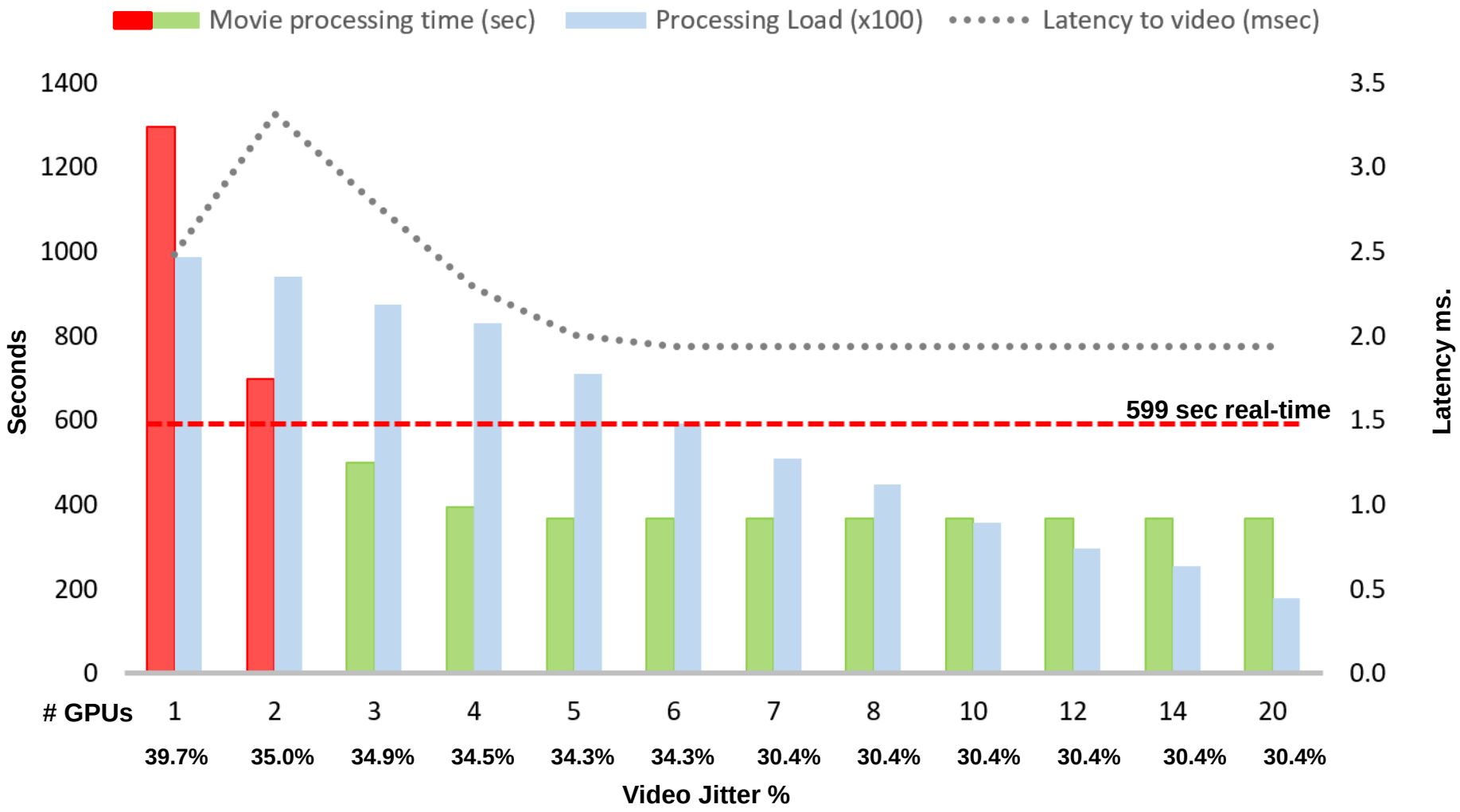
ATSC BLOCK HYPER-THREAD LOADING BASELINE



Baseline captured on January 19 18:07:17 2017
6.11 minute run provided by Corgan Labs
Intel i7-5820k @ 3.3 GHz, 32 GB
6 cores, 12 hyper-threads, 14 ATSC Blocks
Linux release 4.9.3 – 200. fc25x86_64

greg@simpleXecutive.com

ATSC BASELINE TRADES



greg@simpleXecutive.com

ATSC SYSTEM DESIGN EXPERIMENTS

Reduce system output jitter to improve the display by adjusting block priorities

Priority strategies included ordering by:

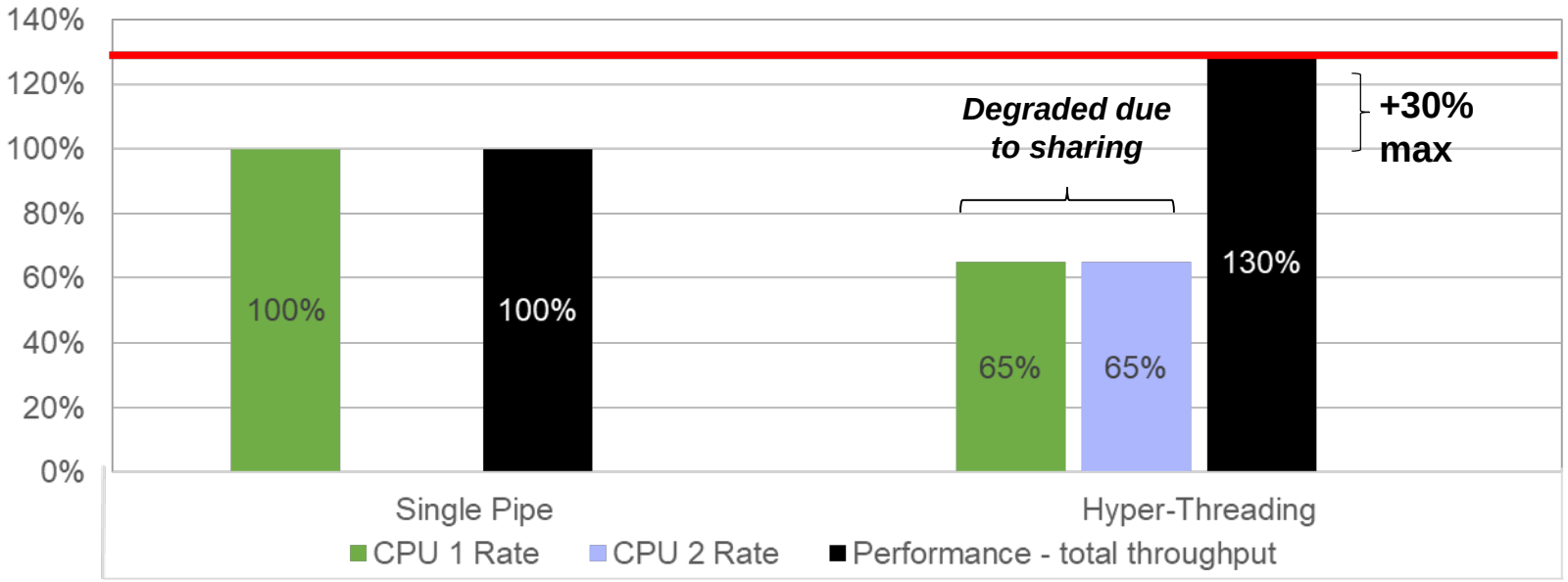
- Block execution order & reverse
- Individual execution duration & reverse
- Total resource consumption & reverse
- Different random assignments

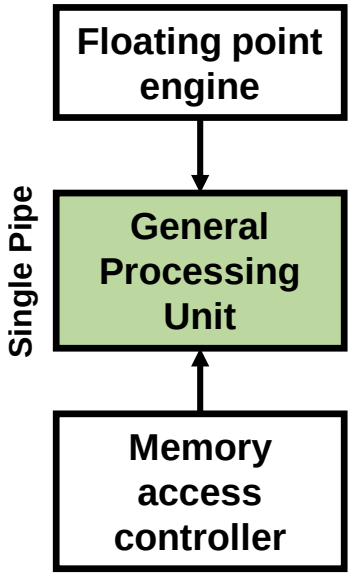
Reduce Rx Filter block execution runtime to improve overall system throughput by:

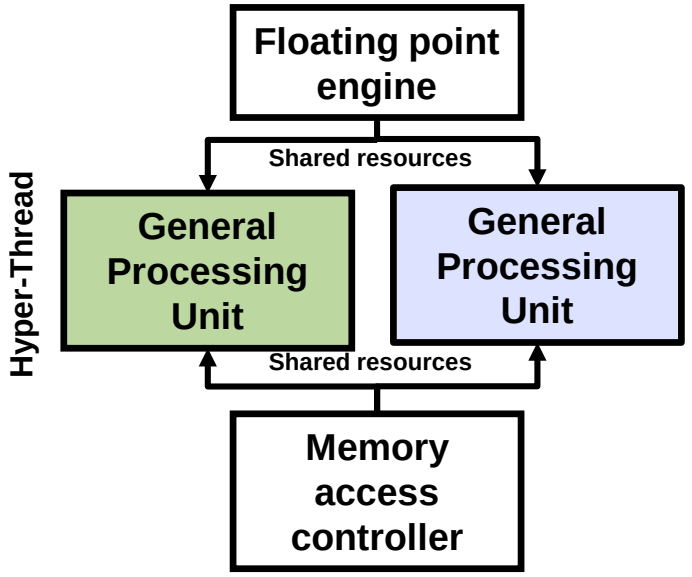
- Overlapping / Parallelizing Rx Filter execution
- Exploiting Intel i7 architectural characteristics

greg@simpleXecutive.com

INTEL HYPER-THREAD DEFINITION

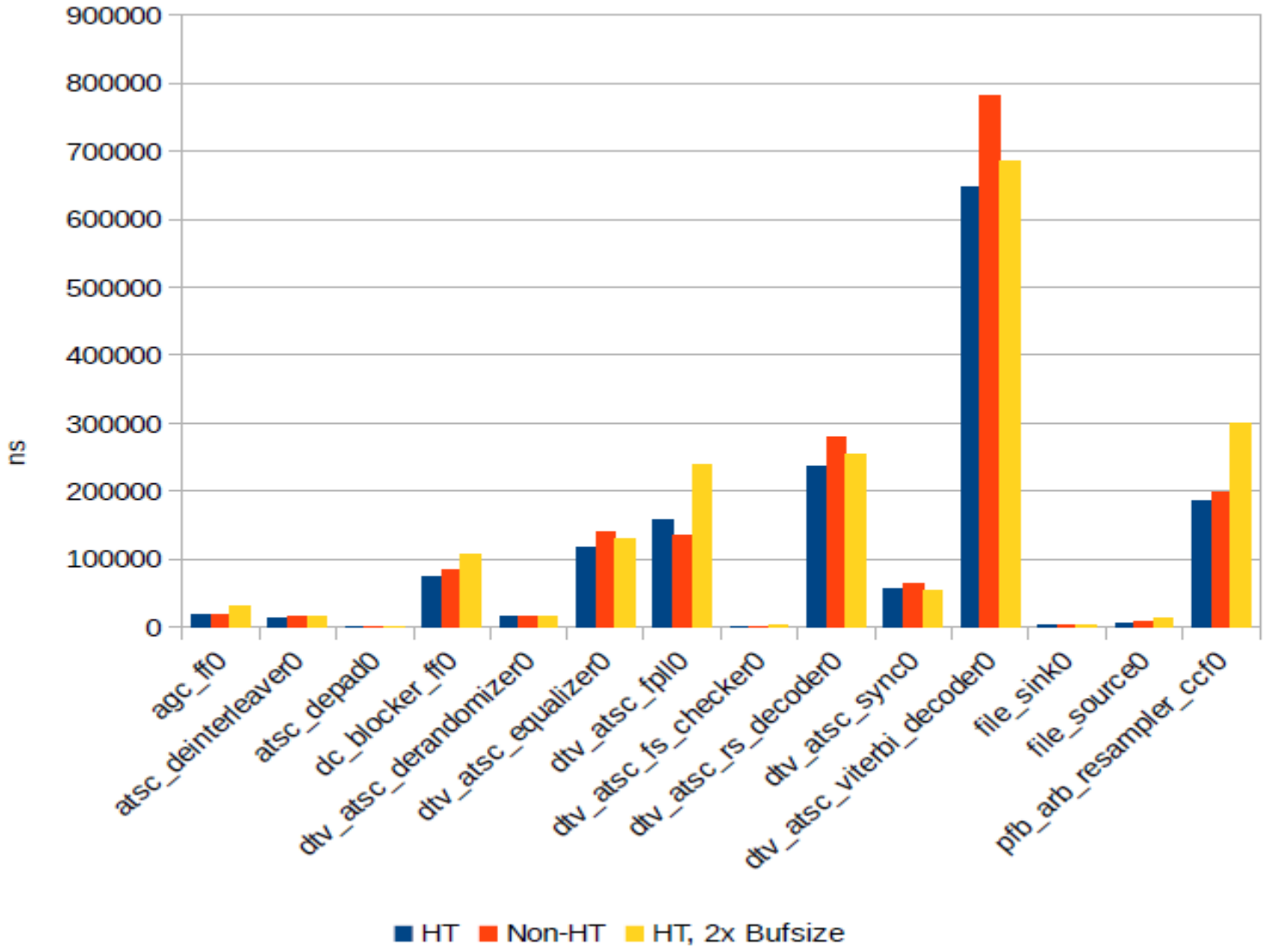






greg@simpleXecutive.com

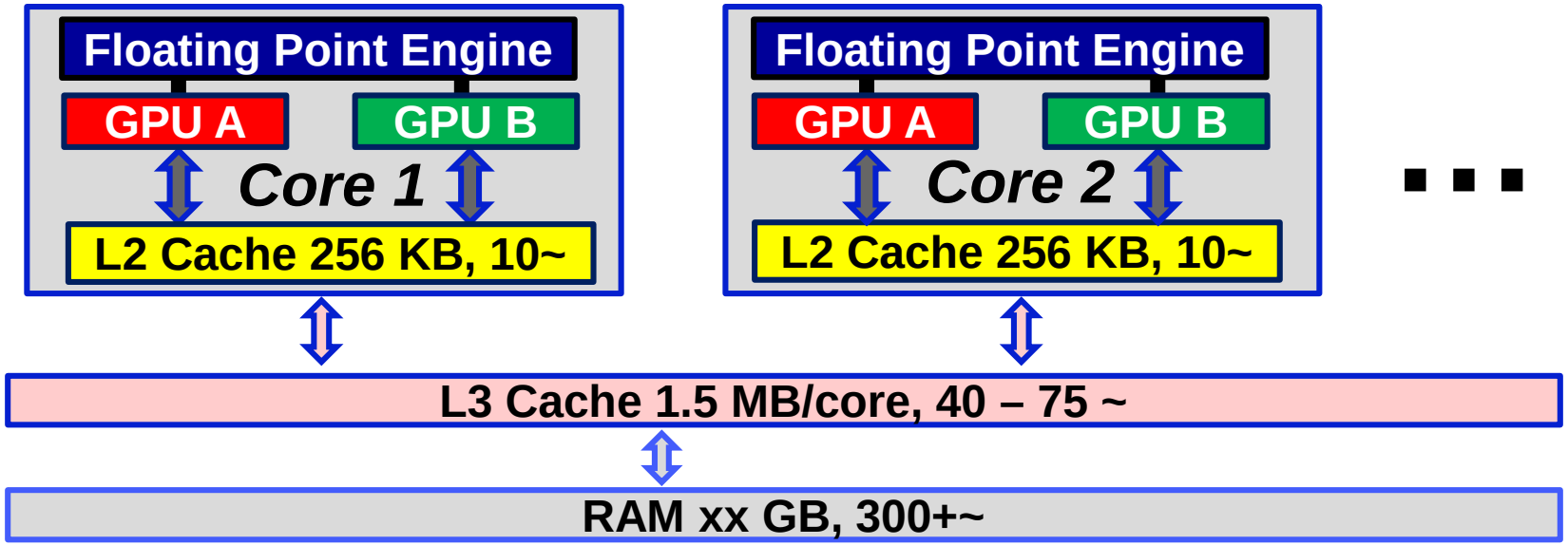
MEASURED WORK TIMES PER BLOCK



* Report received from Marcus Mueller February 14, 2017

greg@simpleXecutive.com

INTEL i7 MEMORY HIERARCHY



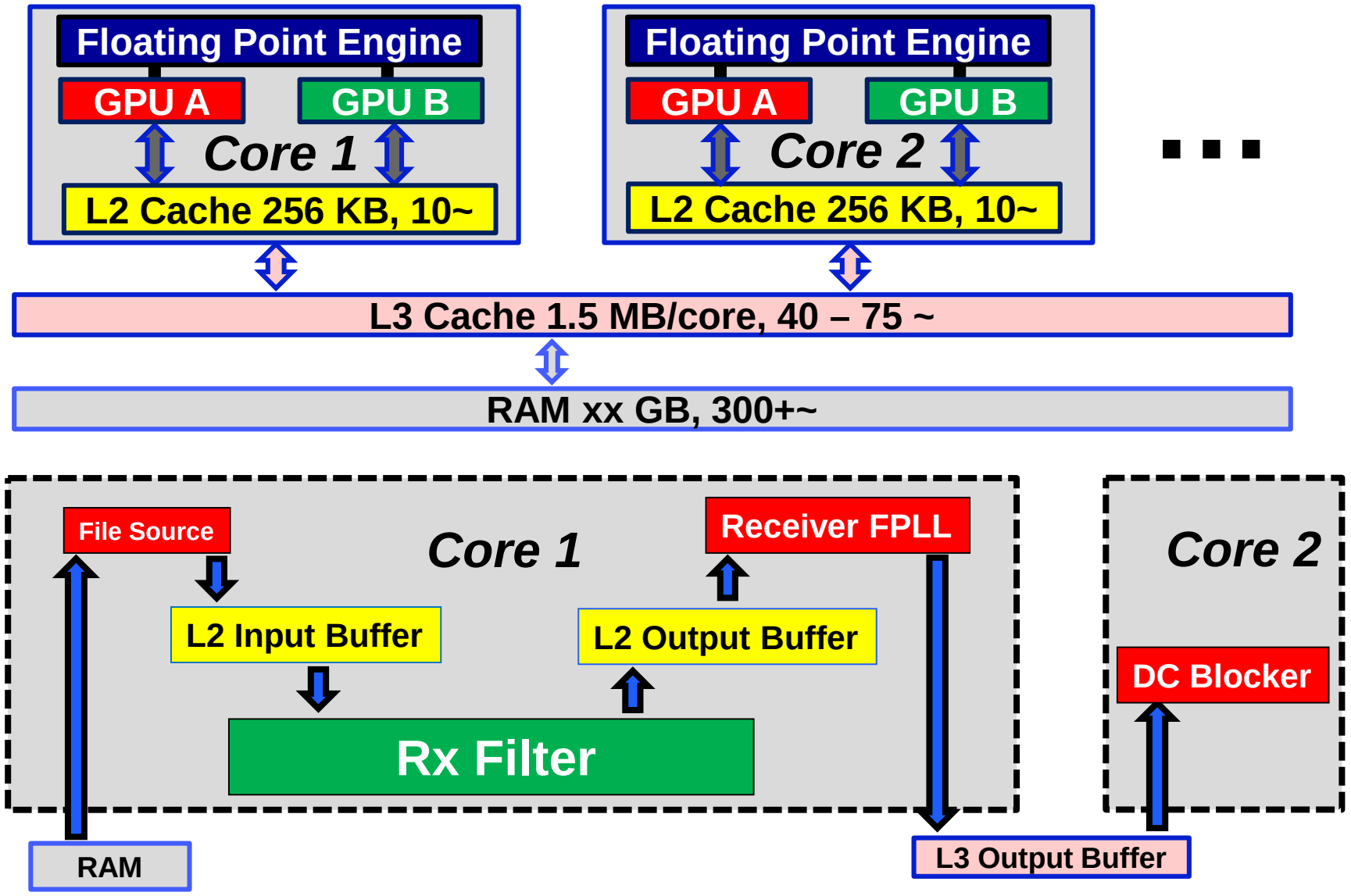
An L2 Cache access costs ~ 3 nanoseconds

An L3 Cache access is about 5 times longer than L2

RAM accesses are about 6 times longer than L3

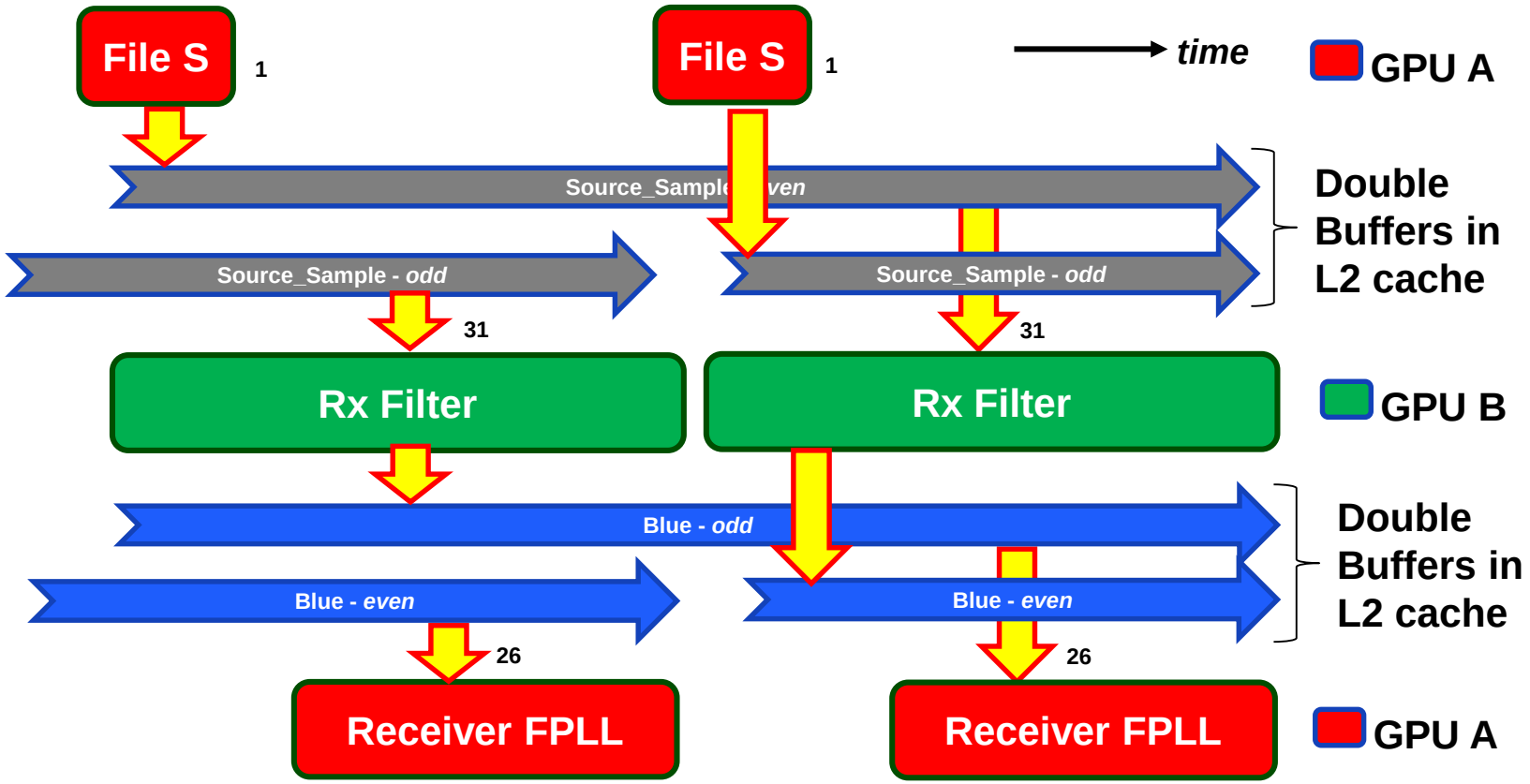
greg@simpleXecutive.com

L2 SIBLING COMMUNICATION



greg@simpleXecutive.com

EXPLOITING L2 SIBLING ALLOCATION

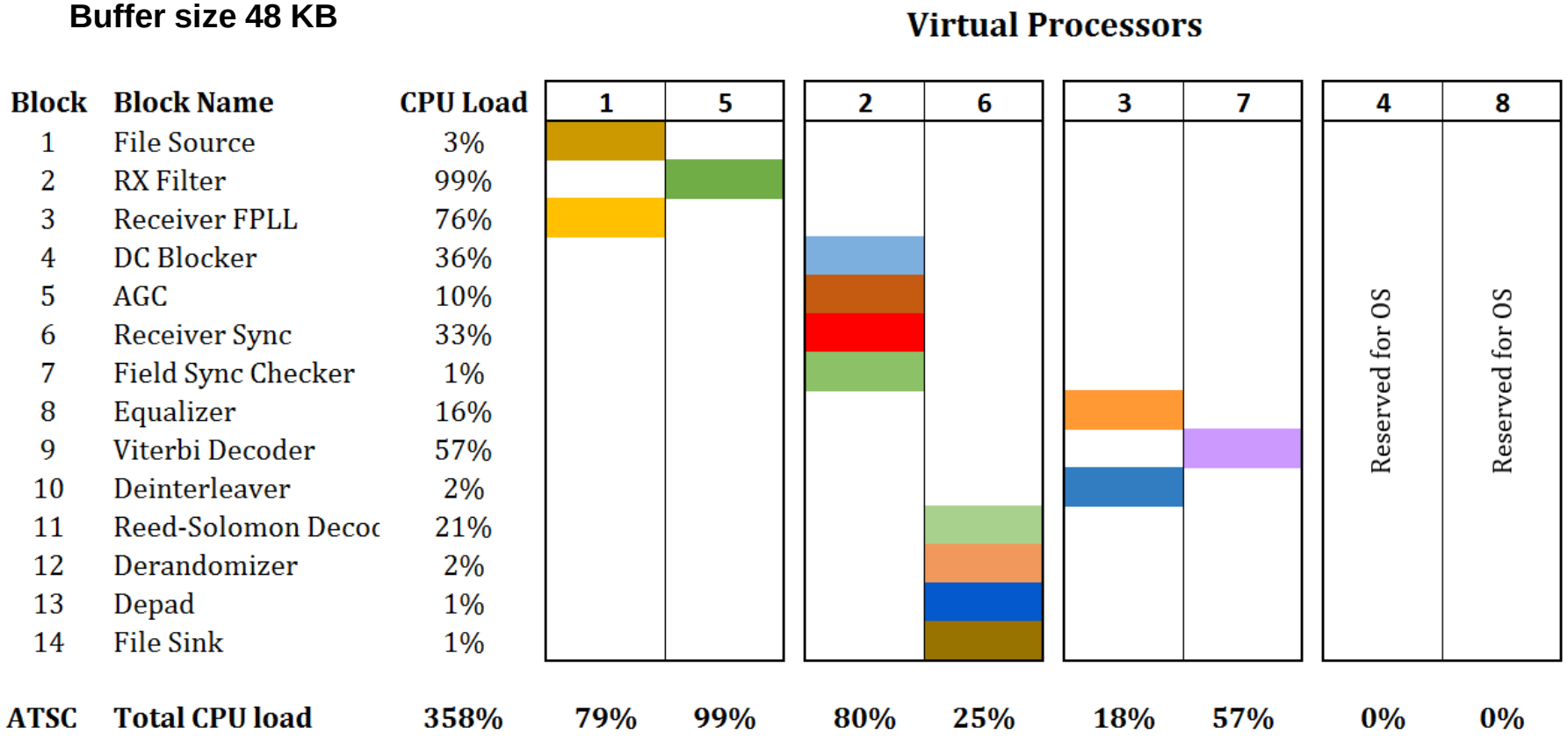


Result: 18.1% performance improvement measured

greg@simpleXecutive.com

EFFECTIVE CORE ALLOCATION FOR an i7 MACHINE

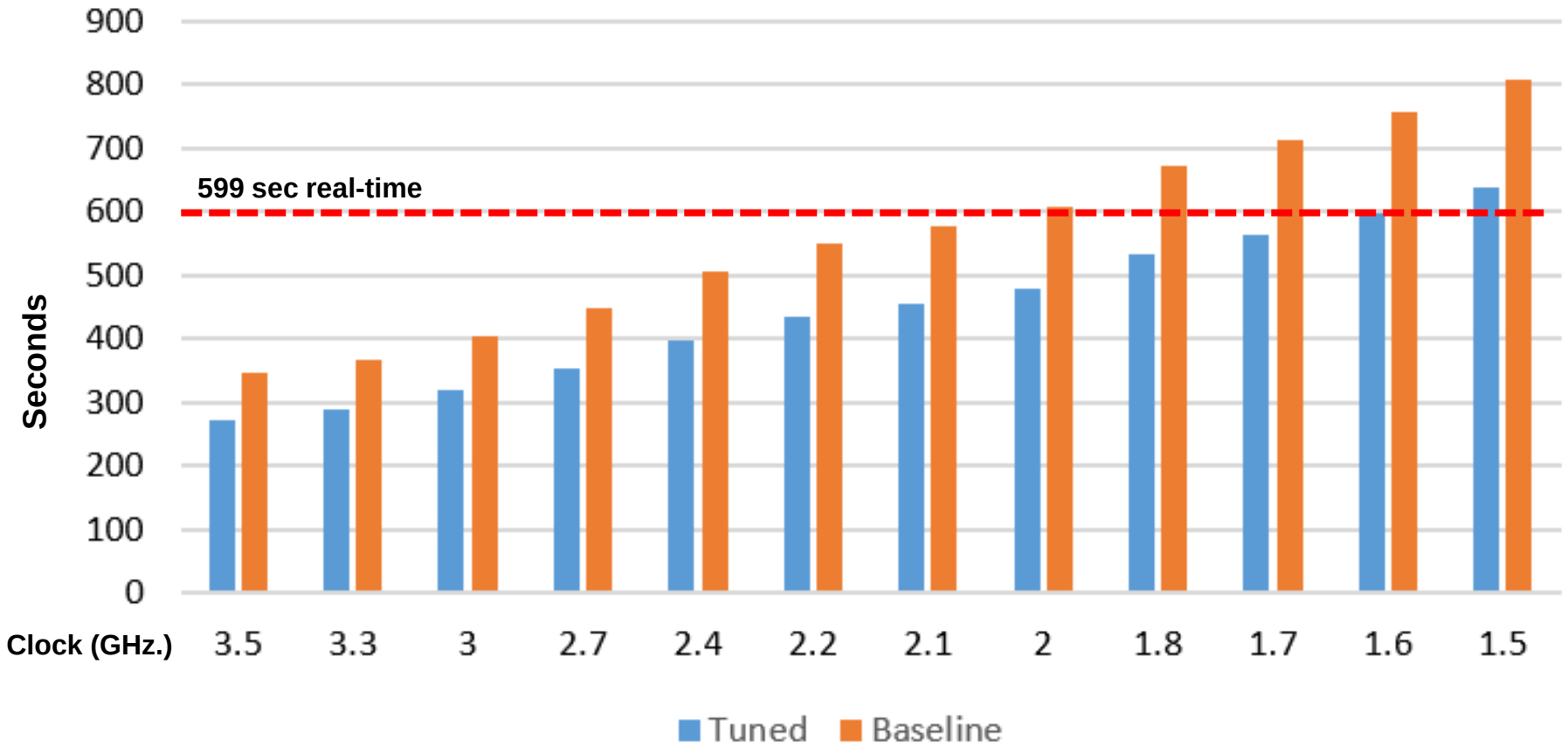
Configuration:
4 cores
Buffer size 48 KB



Block	Block Name	CPU Load	1	5	2	6	3	7	4	8
1	File Source	3%								
2	RX Filter	99%								
3	Receiver FPLL	76%								
4	DC Blocker	36%								
5	AGC	10%								
6	Receiver Sync	33%								
7	Field Sync Checker	1%								
8	Equalizer	16%								
9	Viterbi Decoder	57%								
10	Deinterleaver	2%								
11	Reed-Solomon Decoc	21%								
12	Derandomizer	2%								
13	Depad	1%								
14	File Sink	1%								
ATSC	**Total CPU load**	**358%**	**79%**	**99%**	**80%**	**25%**	**18%**	**57%**	**0%**	**0%**

greg@simpleXecutive.com

CPU CLOCK TRADE



greg@simpleXecutive.com

Questions?

Please join me Thursday, September 14th 10am

Presenting:

A Case Study in Optimizing GNU Radio's ATSC Flowgraphs

Contact: greg@simpleXecutive.com