FPGA acceleration of the Pulsar Search Pipeline

Oliver Sinnen, Krystine Sherwin, and Haomiao Wang

Parallel and Reconfigurable Computing (PARC) Lab,
Department of Electrical, Computer and Software Engineering,
Faculty of Engineering,
University of Auckland

PARALLEL AND RECONFIGURABLE
COMPUTING LAB

Overview

- UoA team actively contributing since 2013
  - Member of NZA
- Work on CSP (Central Signal Processor) package
  - Focus on PSS (Pulsar Search)
  - We are part of TDT team at Uni of Manchester led by Benjamin Stappers
  - Collaborate closely with Prabu Thiagaraj, Raman Research Institute (formerly Uni. of Manchester)
- Investigate and develop PSS pipeline stages on FPGAs
  - FPGAs used as high performance, low-power acceleration devices
- Using OpenCL, a high level (C-based) language
- Targeting high-end FPGAs, initially Intel/Altera, but now also Xilinix

Overview

- UoA team actively contributing since 2013
  - Member of NZA
- Work on CSP (Central Signal Processor) package
  - Focus on PSS (Pulsar Search)
  - We are part of TDT team at Uni of Manchester led by Benjamin Stappers
  - Collaborate closely with Prabu Thiagaraj, Raman Research Institute (formerly Uni. of Manchester)
- Investigate and develop PSS pipeline stages on FPGAs
  - FPGAs used as high performance, low-power acceleration devices
- Using OpenCL, a high level (C-based) language
- Targeting high-end FPGAs, initially Intel/Altera, but now also Xilinix

Overview

- UoA team actively contributing since 2013
  - Member of NZA
- Work on CSP (Central Signal Processor) package
  - Focus on PSS (Pulsar Search)
  - We are part of TDT team at Uni of Manchester led by Benjamin Stappers
  - Collaborate closely with Prabu Thiagaraj, Raman Research Institute (formerly Uni. of Manchester)
- Investigate and develop PSS pipeline stages on FPGAs
  - FPGAs used as high performance, low-power acceleration devices
- Using OpenCL, a high level (C-based) language
- Targeting high-end FPGAs, initially Intel/Altera, but now also Xilinix

Overview

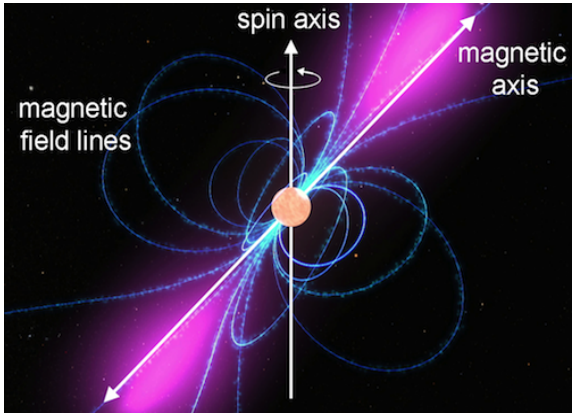
- UoA team actively contributing since 2013
  - Member of NZA
- Work on CSP (Central Signal Processor) package
  - Focus on PSS (Pulsar Search)
  - We are part of TDT team at Uni of Manchester led by Benjamin Stappers
  - Collaborate closely with Prabu Thiagaraj, Raman Research Institute (formerly Uni. of Manchester)
- Investigate and develop PSS pipeline stages on FPGAs
  - FPGAs used as high performance, low-power acceleration devices
- Using OpenCL, a high level (C-based) language
- Targeting high-end FPGAs, initially Intel/Altera, but now also Xilinix

Outline

Outline

What Is A Pulsar?

- Pulsars are rapidly rotating collapsed stars.
- Emit highly directional EM radiation.
- Millisecond pulsars more accurate than (some) atomic clocks.
- Pulsar timing arrays for mapping the universe.
- Binary pulsars for testing general relativity.

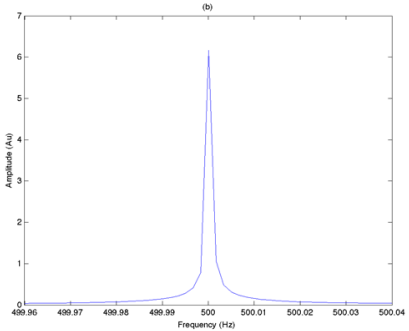


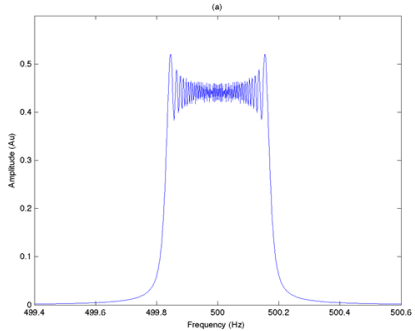
1

[1] Image Credit: NASA/Tod Strohmayer (GSFC)/Dana Berry (Chandra X-Ray Observatory)

Pulsar Search

- Acceleration search: 1) Time-domain 2) Frequency-domain





Pulsar Search challenge

- Unknown position
  - Scan sky
- Unknown distance
  - Trying 1000-2000 different dedispersion measures (DM)
- Unknown period
- Unknown orbital frequency
  - Compensate with matched filtering (currently 85 different filters)
- Faint signal
  - Integrating for ca. 536 secs

=> Brute Force Search

Outline

SKA Technical Reports

Our group contributed to SKA CSP pulsar search sub-element

- SKA1 CSP Pulsar Search Sub-element Prototype Plan document
  - Proto-PS_SKA-TEL-CSP-0000116_2_PulsarSearchProtoPlan_Armour_2016-10-29.docx, contributed to Section 5.4.5
- SKA CSP Pulsar Search Sub-element Detailed Design Document
  - SKA-TEL-CSP-0000082_2_PSSDetDesDoc_Karastergiou_2018-05-29.docx, contributed to Section 16.3
- SKA-TEL-CSP-PSS_0000007_TestReportFTConvolution_01.doc
- SKA-TEL-CSP-PSS_0000008_TestReportHarmonicSum_01.doc
- SKA-TEL-CSP-PSS_0000009_TestReportMedianFiltering_01.doc

Implementations

Contributed OpenCL on FPGA implementations of

- Fourier Domain Acceleration Search
  - Multi-filter FT convolution
  - Harmonic summing
  - Threshold (candidate) detection
- Large-scale median-filtering

For all designs

- Tested and evaluated on real hardware
- Licensed under OpenSource (GPL)

Publications-Journal

- Wang, H., Thiagaraj, P., & Sinnen, O. (2019). FPGA-based Acceleration of FT Convolution for Pulsar Search Using OpenCL. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 11(4), 24.
- Wang, H., Thiagaraj, P., & Sinnen, O. (2018). Harmonic-summing Module of SKA on FPGA--Optimising the Irregular Memory Accesses. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Early Access, 10.1109/TVLSI.2018.2882238.
- Wang, H., Thiagaraj, P., & Sinnen, O. (2019). Combining Multiple Optimised FPGA-based Pulsar Search Modules Using OpenCL. *Journal of Astronomical Instrumentation*. (Accepted)

Publications-Conference

- Wang, H., Sinnen, O. (2019, January). *Design Space Exploration of FPGA-based FIR Filter Group in Fourier Domain.* In 2019 International Conference on Electronics (ICEIC 2019). IEEE.
- Sherwin, K., Stappers, B., Thiagaraj, P., Wang, K., & Sinnen, O. (2018, December). *Investigating how hardware architectures are expressed in high-level languages for an SKA algorithm.* In Field-Programmable Technology (FPT), 2018 International Conference on. IEEE.
- Wang, H., Stappers, B., Thiagaraj, P., & Sinnen, O. (2018, December). *Optimisation of Convolution of Multiple Different Sized Filters in SKA Pulsar Search Engine.* In Field-Programmable Technology (FPT), 2018 International Conference on. IEEE.
- Sherwin, T., Kevin, I., Wang, K., Thiagaraj, P., & Sinnen, O. (2018, August). *Median filtering with very large windows: SKA algorithms for FPGAs.* In 2018 28th International Conference on Field Programmable Logic and Applications (FPL) (pp. 196-1965). IEEE.
- Wang, H., Zhang, M., Thiagaraj, P., & Sinnen, O. (2016, December). *FPGA-based acceleration of FDAS module using OpenCL.* In Field-Programmable Technology (FPT), 2016 International Conference on (pp. 53-60). IEEE.
- Wang, H., Gante, J., Zhang, M., Falcão, G., Sousa, L., & Sinnen, O. (2016, December). *High-Level Designs of Complex FIR Filters on FPGAs for the SKA.* In High Performance Computing and Communications (HPCC), 2016 IEEE 18th International Conference on (pp. 797-804). IEEE.
- Wang, H., & Sinnen, O. (2015, December). *FPGA based acceleration of FDAS module for Pulsar Search.* In Field Programmable Technology (FPT), 2015 International Conference on (pp. 240-243). IEEE.

Outline

Hardware

- Terasic DE5-Net with Intel Stratix V x4
- Nallatech 385A with Intel Arria 10 x6
- Xeon+FPGA with Intel Arria 10 (supported by Intel hardware accelerator research program, remote access)
- Xilinx VCU1525 with Virtex UltraScale+ XCVU9P x1
- Xilinx VCU1550 with Virtex UltraScale+ FPGA x1

Hardware

Hardware Specifications

Device	Terasic DE5-Net (**S5**)	Nallatech 385A (**A10**)	Xeon+FPGA (**X+A10**)	Xilinx VCU1525 (**U9**)
Hardware	Intel Stratix V 5SGXA7	Intel Arria 10 GX1150	Intel Arria 10 GX1150	Xilinx Virtex UltraScale+ XCVU9P
Technology	28nm	20nm	20nm	16nm
Compute resource	622,000 LEs	1,506,000 LEs	1,506,000 LEs	2,586,000 LEs
	256 DSP blocks	1,518 DSP blocks	1,518 DSP blocks	6,840 DSP blocks
On-chip memory size	50Mb	53Mb	53Mb	345.9 Mb
Off-chip memory size	2 x 2GB DDR3	2 x 4GB DDR3	–	4 x 16GB DDR4
OpenCL global memory bandwidth	$25,600MB/s$	$35,128MB/s$	$25,000MB/s$	
Max clock frequency	600MHz	1.5GHz	1.5GHz	
Max power consumption	—	75W	–	225W

OpenCL

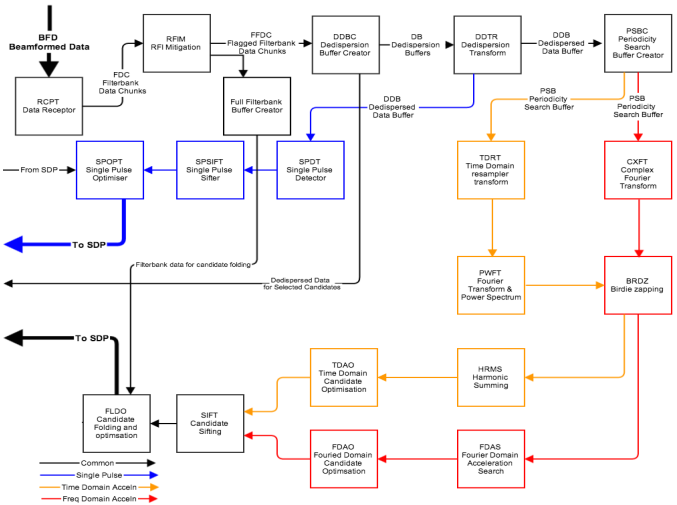
- High level approach to parallel programming.
- Execution portable.
  - Same code, multiple platforms.


```
__kernel
void median(global const PIXEL_TYPE * restrict frame_in,
            global O_TYPE * restrict frame_out)
{
    #pragma unroll UNROLL_SIZE
    for (int col=OTHER_HALF; col<(COLS-OTHER_HALF); col+=(P_SIZE)) {
        // Start with a window full of 0's
```

OpenCL differences: Intel and Xilinx

Optimization Features	Intel (AOC)	Xilinx (XOCC)
Pipe	Channel and Pipe	Pipe
Loop unroll	#pragma unroll	__attribute__((opencl_unroll_hint))
Pipelining	Automatically	__attribute__((xcl_pipeline_loop)) __attribute__((xcl_dataflow))
Kernel Vectorization	__attribute__((num_simd_work_item))	__attribute__((vec_type_hint))
Floating-point operations	balanced tree (–fp-relaxed) rounding operations (–fpc)	No related attributes
Multiple memory banks	-no-interleaving=<global_memory_type>	--max_memory_ports –sp ....
Local memory partition	__attribute__((<memory_type>, ...))	__attribute__((xcl_array_partition))

Pulsar Search Engine (PSS)



Complex system requiring more than 10POps.

- RFIM
- **FLDO**
- **FDAS**

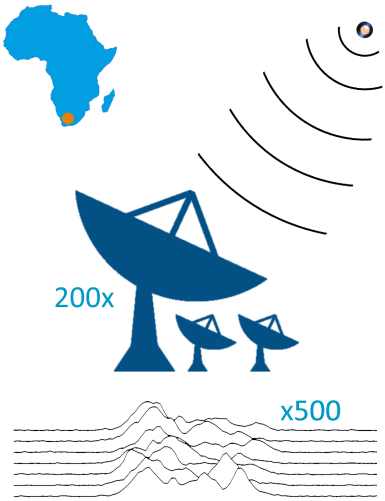
Specifications of Task

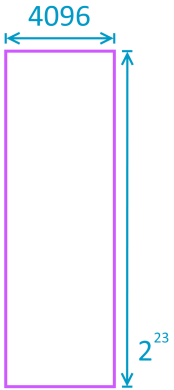
Parameter	Destriiption	Value
N_{beam}	# of beams	$1000 \sim 2000$
T_{obs}	Observation period	$540s$
N_{DM}	# of de-dispersion measure (DM) trails	6000
$N_{channel}$	# of complex samples per group	2^{21}
$N_{template}$	# of templates/filter	85
N_{tap}	# of average template/filter length	211
N_{hm}	# of harmonic planes	8
$N_{candidate}$	# of candidates per harmonic plane	200

Outline

Outline

Input data

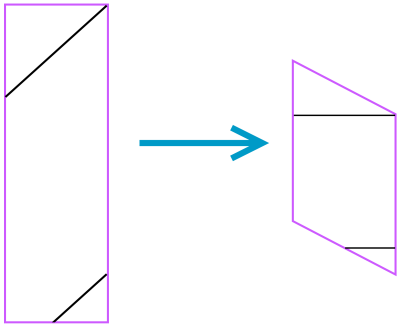




- 536 seconds in 2^{23} time steps.
- 8bit samples from 4096 frequency channels every 64us.
- Up to 500 candidate profiles given for each of two beams.
  - Initial parameters & desired search space.
- 3 parameters to optimise.
- 256 **million trials/second/node**.

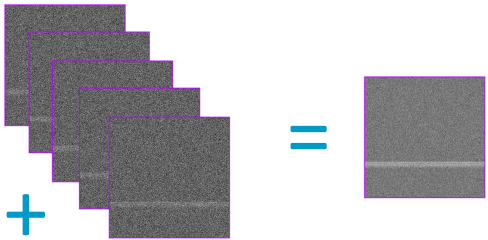
Outline

FOLD - Integration & dedispersion



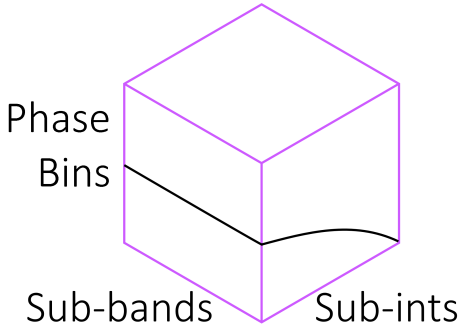
- Input may be integrated, depending on pulsar data.
- Dispersion causes time delays at lower frequencies which must be corrected.
  - Described by Dispersion Measure (DM).
  - Higher frequencies delayed more to compensate.
  - Integer bin shifts based on frequency.
- Frequency channels integrated to ≤ 128 sub-bands for folding.

FOLD - Repeating pulses



- Combine multiple pulses.
- Random background noise cancels out.
- Repeating pulsar signal retains strength.
- Pulsar SNR improves.

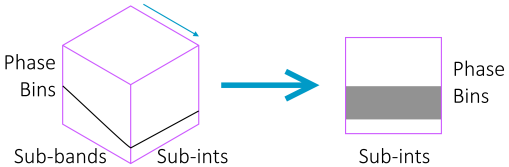
FOLD - 3D data cube



- Up to 64 sub-integrations.
- Each sub-integration containing series of pulses folded together.
- Each folded pulse up to 128 sub-bands across 128 phase bins.

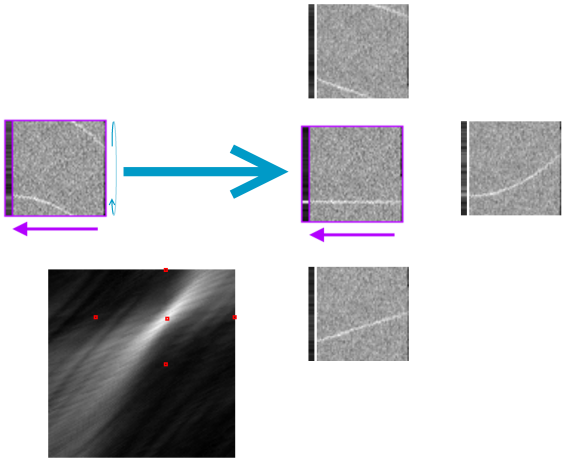
Outline

OPT - Dispersion Measure (DM) trials



- Higher frequency values delayed in time.
- Shifting phase bins across all sub-integrations corrects delay.
- Each trial followed by integration of the sub-bands, giving a flattened 2D array.
  - Period trials independent of sub-bands.
  - Integration reduces necessary computation.

OPT - Period trials



- Both P and $\dot{P}$ varied, sweeping performed as a series of bin shifts.
- One dimension rotates while other fixed.
- Shifting phase bins across all sub-bands approximates refolding with new P and $\dot{P}$.
- Higher subintegrations (later pulses) shifted more.
- Values rotated out the bottom placed at the top.
  - Represents shift from end of one period to start of next.

Output

- Initial data cube sent to SDP for further processing.
- Also outputs metadata including estimated pulse width and SNR for each trial.
- Optimal parameters then determined, as well as relationships such as harmonics.

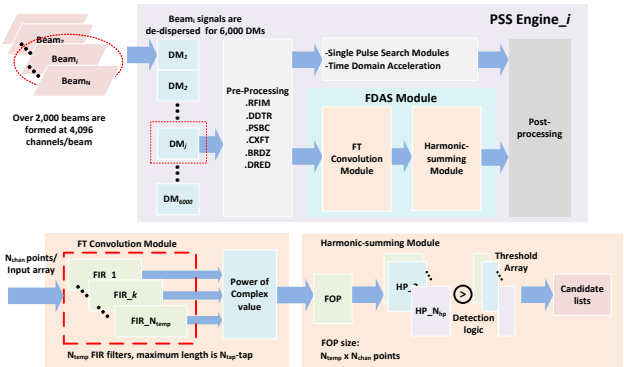
Outline

Outline

Fourier-domain Acceleration Search (FDAS)

FDAS module is applied to search for (binary) pulsars with constant frequency derivatives in Fourier-domain

FT Convolution module + Harmonic-summing module



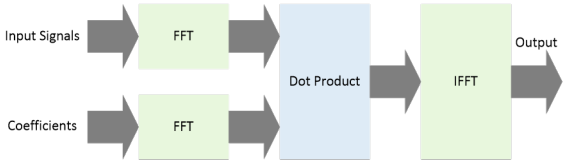
FT Convolution Module

- Time-domain FIR Filter (TDFIR)

$$y_m[i] = \sum_{k=0}^{K-1} x_m[i-k]h_m[k], \text{ for } i = 0, 1, \ldots N-1$$

- Frequency-domain FIR Filter (FDFIR)

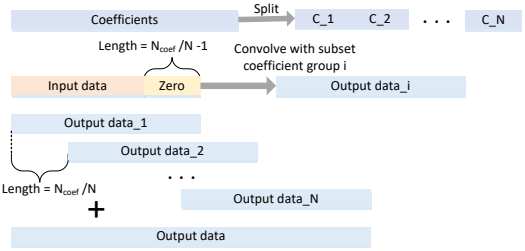
$$\mathscr{F}\{f * h\} = \mathscr{F}\{f\} \cdot \mathscr{F}\{h\}$$



FT Convolution Module

Overlap-add Algorithm

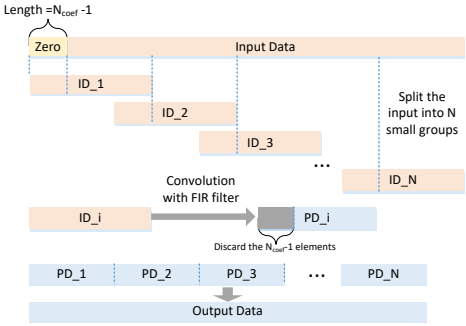
- Split the coefficient array –> **OLA-TD**



(a) OLA

Overlap-save Algorithm

- Split the input array –> **OLS-FD**

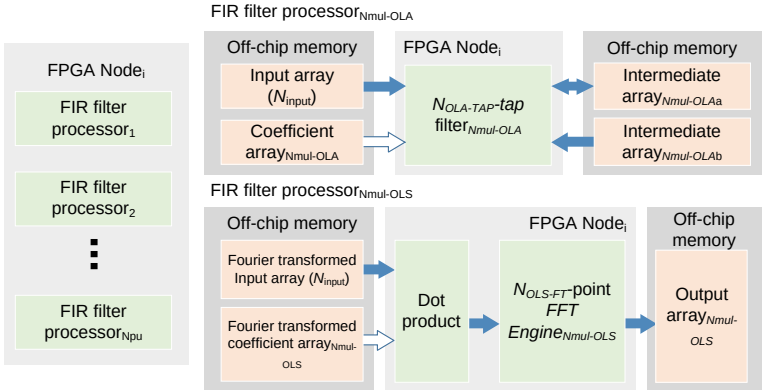


(b) OLS

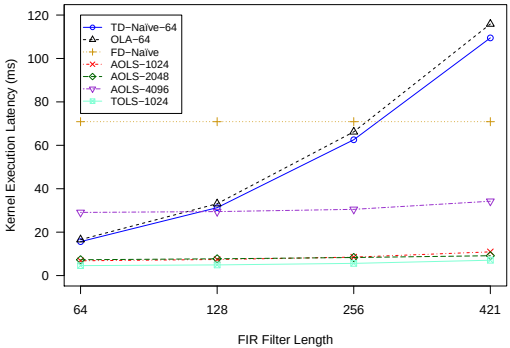
FT Convolution Module

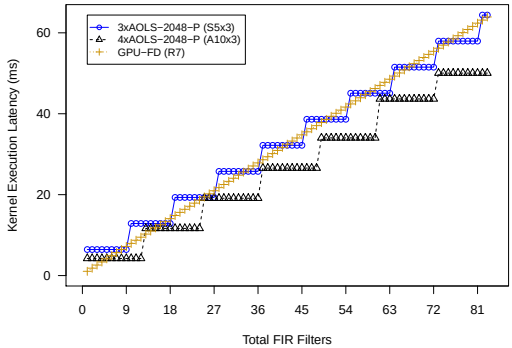
For general implementation, all filters are extended to the same length as the longest filter.

- Multiple processing units can be implemented in one FPGA image
- All processing units share the same input array or Fourier transformed input array



FT Convolution Module-Evaluation



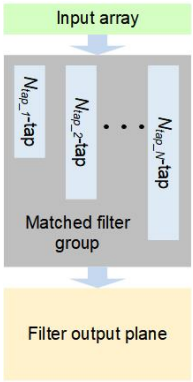


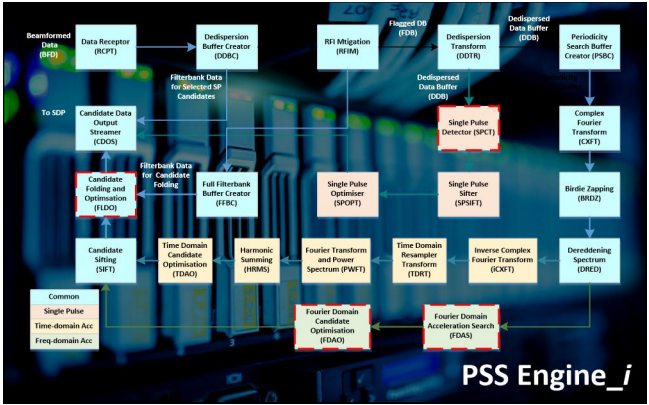
Outline

Matched Filter Group Optimisation

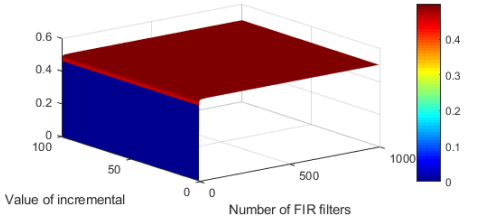
$MF - (N_{filter},\ N_{inc},\ Tap_1)$: A group of FIR filters with different sizes

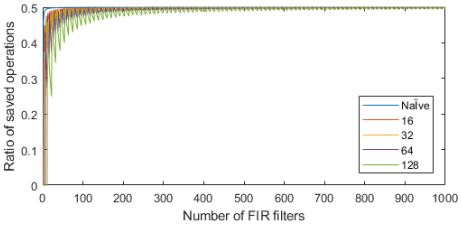
$$Tap_{i+1} = Tap_i + N_{inc}$$

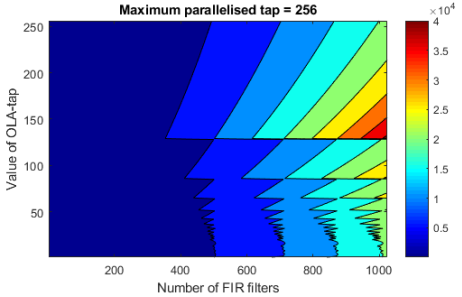




Matched Filter Group Optimisation-TD



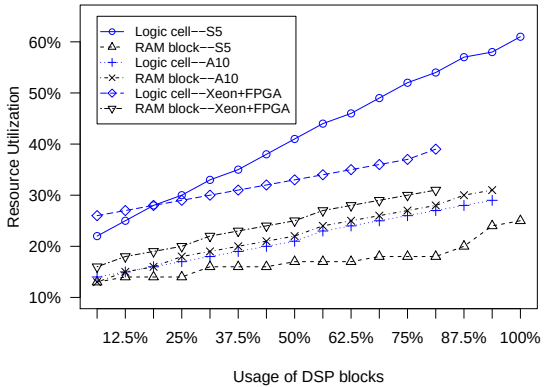


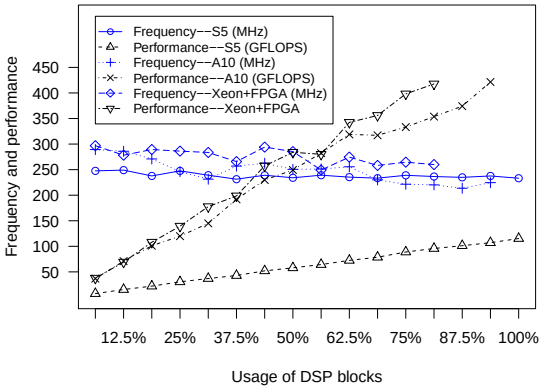


- The smaller the number of launch times the faster the design
- Search for the optimal $N_{OLA-tap}$ for the OLA-TD based implementation on a specific FPGA

Matched Filter Group Optimisation-TD

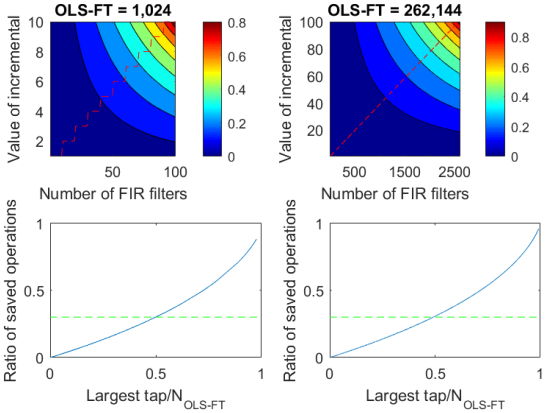
Main factors: 1) DSP blocks and 2) off-chip memory bandwidth





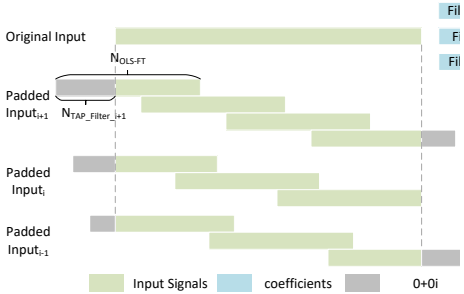
Load balancing–longest processing time rule (LPT)

Matched Filter Group Optimisation-FD



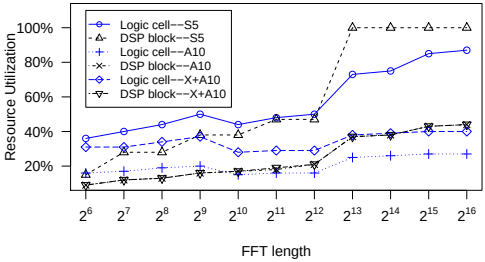
- The higher the value of $Tap_{N_{filter}}/N_{OLS-FT}$, the more operations can be saved (when $Tap_{N_{filter}}/N_{OLS-FT} = 0.5$, up to 30% operations can be saved)
- The larger the value of N_{OLS-FT}, the more logic resource is costs
- For a specific N_{OLS-FT}, N_{pu} is decided by the logic resources on an FPGA

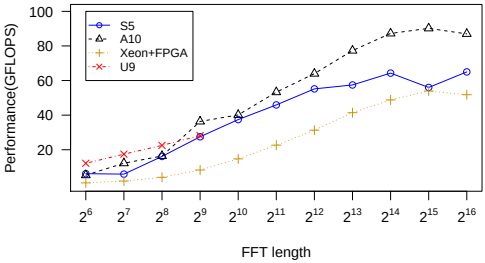
Matched Filter Group Optimisation-FD



- Padded input group for small filters cannot be used for large FIR filters
- The higher $N_{TAP-filter}/N_{OLS-FT}$, the larger the size of padded input group
- Multiple padded input groups lead to the increase of needed off-chip memory
- The goal is find the suitable N_{OLS-FT} and the total number of padded input group

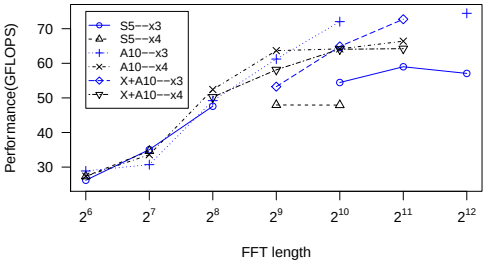
Matched Filter Group Optimisation-FD

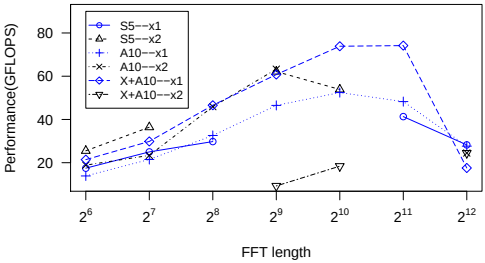




- processing multiple points in parallel such as 4, 8, and 16 (8 in Figure)
- Resource usage drops when FFT length increased from 2^9 to 2^{10} and increases dramatically when FFT length is larger than 2^{12}.
- The larger the FFT length, the higher the performance from 2^6 to 2^{12}

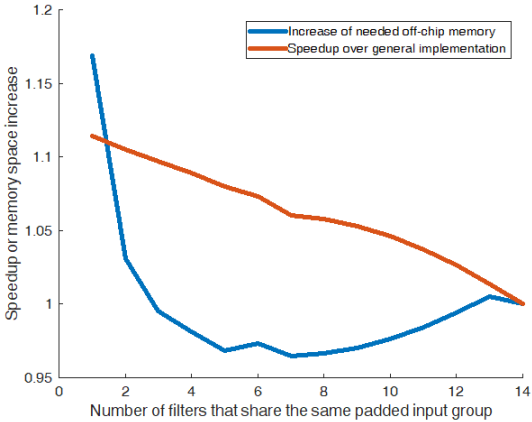
Matched Filter Group Optimisation-FD

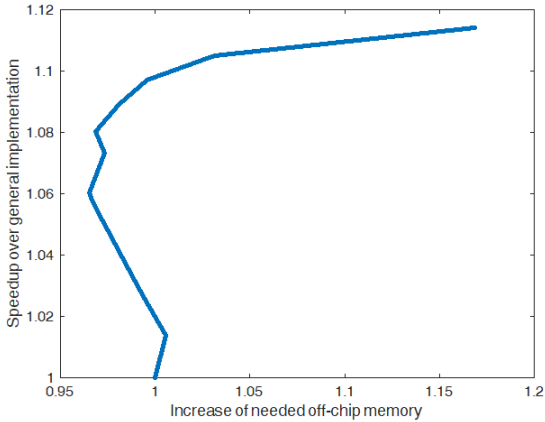




- 4-point FFT engine $N_{pu} = 3$ and $N_{pu} = 4$ are tested
  - S5: $N_{OLS-FT} = 2048$, $N_{pu} = 3$
  - A10: $N_{OLS-FT} = 4096$, $N_{pu} = 3$
  - X+A10: $N_{OLS-FT} = 2048$, $N_{pu} = 4$
- 8-point FFT engine $N_{pu} = 1$ and $N_{pu} = 2$ are tested
  - S5: $N_{OLS-FT} = 1024$, $N_{pu} = 2$
  - A10: $N_{OLS-FT} = 512$, $N_{pu} = 2$
  - X+A10: $N_{OLS-FT} = 2048$, $N_{pu} = 1$

Matched Filter Group Optimisation-FD





Summary

- Investigated the optimisation of matched filter group implementation for the pulsar search engine.
- For the optimisation in time-domain, the main task is to balance the size of $N_{OLA-tap}$ and off-chip memory bandwidth
- For the optimisation in Fourier-domain, the main task is to balance the off-chip memory cost and the speedup in execution latency
- Duplicating more processing units cannot improve the performance when the required bandwidth outnumbers the device off-chip memory bandwidth

Conclusions and future work

- Investigated high-performance implementations of PSS pipelines on FPGA
  - Meeting requirements using 'portable' high-level approach OpenCL

Future

- Investigate other stages for implementation
- Investigate the Candidate sifting module (SIFT) that process the output from the Fourier Domain Candidate Optimisation (FDAO) module

OpenMP conference

International OpenMP conferences in Auckland in September 2019

- OpenMPCon – developer conference 9-10 Sept. – www.openmpcon.org

- Tutorial day 11 Sept.
- IWOMP - academic conference 12- 13 Sept. – parallel.auckland.ac.nz/iwomp2019