Optimized FPGA Implementation of Multi-Rate FIR Filters through Thread Decomposition

Jason Zheng[1] Kayla Nguyen[1] Yutao He[1]

[1] Jet Propulsion Laboratory, California Institute of Technology

IEEE Aerospace Conference, 2010

 Government sponsorship acknowledged

Outline

 Government sponsorship acknowledged

Outline

Government sponsorship acknowledged

Finite Impulse Response (FIR) Filter

Digital Signal Processing (DSP) 101

- An FIR filter is a digital filter with finite impulse response.
- Each output is a weighted sum of the previous N inputs.
- $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$
- N is the size, or taps of the filter, h is a coefficient vector.
- N determines the number of multiplications required per output.

Government sponsorship acknowledged

Finite Impulse Response (FIR) Filter

Digital Signal Processing (DSP) 101

- An FIR filter is a digital filter with finite impulse response.
- Each output is a weighted sum of the previous N inputs.
- $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$
- N is the size, or taps of the filter, h is a coefficient vector.
- N determines the number of multiplications required per output.

Government sponsorship acknowledged

Finite Impulse Response (FIR) Filter

Digital Signal Processing (DSP) 101

- An FIR filter is a digital filter with finite impulse response.
- Each output is a weighted sum of the previous N inputs.
- $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$
- N is the size, or taps of the filter, h is a coefficient vector.
- N determines the number of multiplications required per output.

Government sponsorship acknowledged

Finite Impulse Response (FIR) Filter

Digital Signal Processing (DSP) 101

- An FIR filter is a digital filter with finite impulse response.
- Each output is a weighted sum of the previous N inputs.
- $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$
- N is the size, or taps of the filter, h is a coefficient vector.
- N determines the number of multiplications required per output.

Government sponsorship acknowledged

Down- and Upsampling
Digital Signal Processing (DSP) 101

- Downsampling reduces the sampling rate by removing samples.
  - Downsampling by a factor of M means to reduce the sampling rate f to $\frac{f}{M}$
- Upsampling increases the sampling rate by introducting duplicate samples.
  - Upsampling by a factor of L means to increase sampling rate f to Lf
  - Fills the new samples with exisiting sample values
- Downsampling and Upsampling are combined with FIR filters to create Multi-Rate FIRs (MRFIR).

Government sponsorship acknowledged

Down- and Upsampling
Digital Signal Processing (DSP) 101

- Downsampling reduces the sampling rate by removing samples.
  - Downsampling by a factor of M means to reduce the sampling rate f to $\frac{f}{M}$
- Upsampling increases the sampling rate by introducting duplicate samples.
  - Upsampling by a factor of L means to increase sampling rate f to Lf
  - Fills the new samples with exisiting sample values
- Downsampling and Upsampling are combined with FIR filters to create Multi-Rate FIRs (MRFIR).

Government sponsorship acknowledged

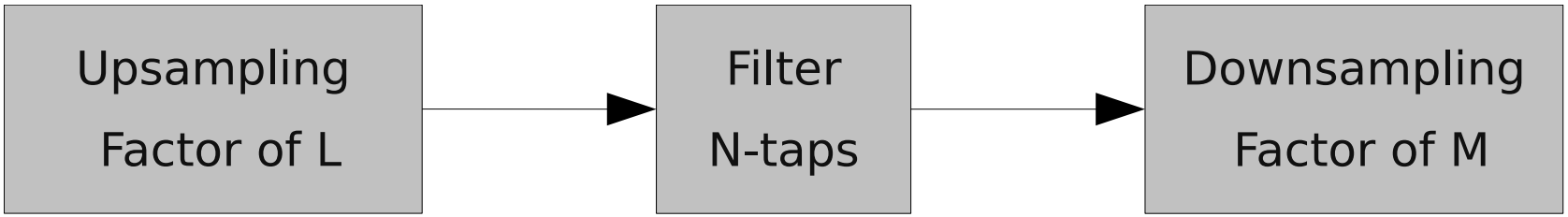
Down- and Upsampling
Digital Signal Processing (DSP) 101

- Downsampling reduces the sampling rate by removing samples.
  - Downsampling by a factor of M means to reduce the sampling rate f to $\frac{f}{M}$
- Upsampling increases the sampling rate by introducting duplicate samples.
  - Upsampling by a factor of L means to increase sampling rate f to Lf
  - Fills the new samples with exisiting sample values
- Downsampling and Upsampling are combined with FIR filters to create Multi-Rate FIRs (MRFIR).

Government sponsorship acknowledged

Multi-Rate FIR (MRFIR)

Digital Signal Processing (DSP) 101



- Decimation filter when L=1.
- Interpolation filter when M=1.
- MRFIRs are widely used as digital frontends for radar, imager, spectrometer, etc.

Government sponsorship acknowledged

Outline

Government sponsorship acknowledged

The Basic Problem

- How to efficiently implement an MRFIR on a Field Programmable Gate Array (FPGA) platform?
- Problem Definition:
  - All coefficients are arbitrary.
  - All inputs samples are arbitrary.
  - Input samples must be processed in real time without buffering.
  - Given N, M, and L, performance is defined as the highest sustainable input rate.

Government sponsorship acknowledged

The Basic Problem

- How to efficiently implement an MRFIR on a Field Programmable Gate Array (FPGA) platform?
- Problem Definition:
  - All coefficients are arbitrary.
  - All inputs samples are arbitrary.
  - Input samples must be processed in real time without buffering.
  - Given N, M, and L, performance is defined as the highest sustainable input rate.

Government sponsorship acknowledged

Observations

- The FPGA platform has a limited number of multipliers available.
- Multiplier count is the bottleneck of MRFIR performance.
- Down- and Upsampling suggests not all computations are necessary.
- Real question: How to design a filter architecture to maximize the performance/multiplier ratio?

Government sponsorship acknowledged

Observations

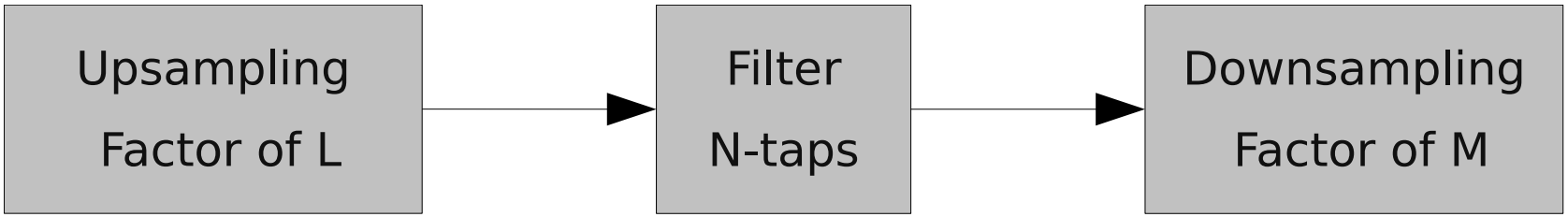
- The FPGA platform has a limited number of multipliers available.
- Multiplier count is the bottleneck of MRFIR performance.
- Down- and Upsampling suggests not all computations are necessary.
- Real question: How to design a filter architecture to maximize the performance/multiplier ratio?

Government sponsorship acknowledged

Outline

Government sponsorship acknowledged

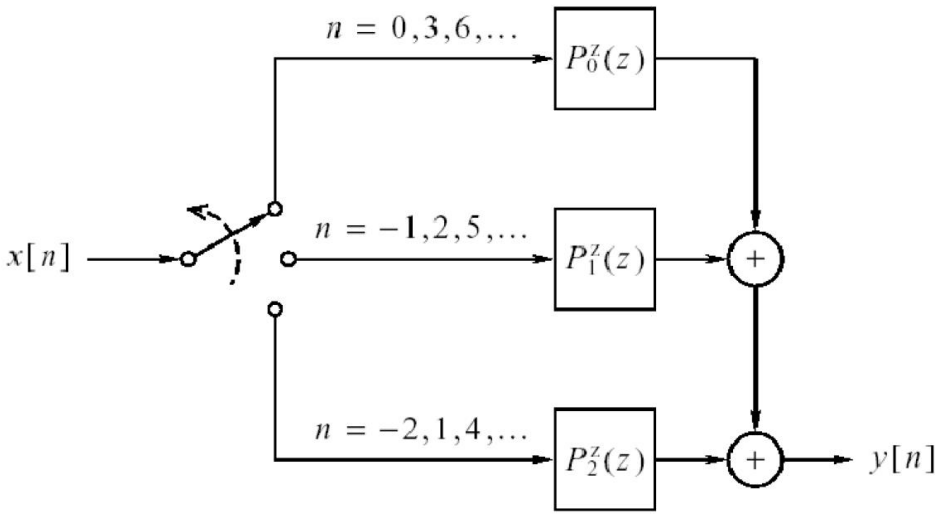
Straight Forward Implementation



- Implemenation discrete FIR, downsample, and upsample blocks.
- No multiplier sharing/optimization.
- Many wasted/repeated computations.

Government sponsorship acknowledged

Polyphase Decomposition

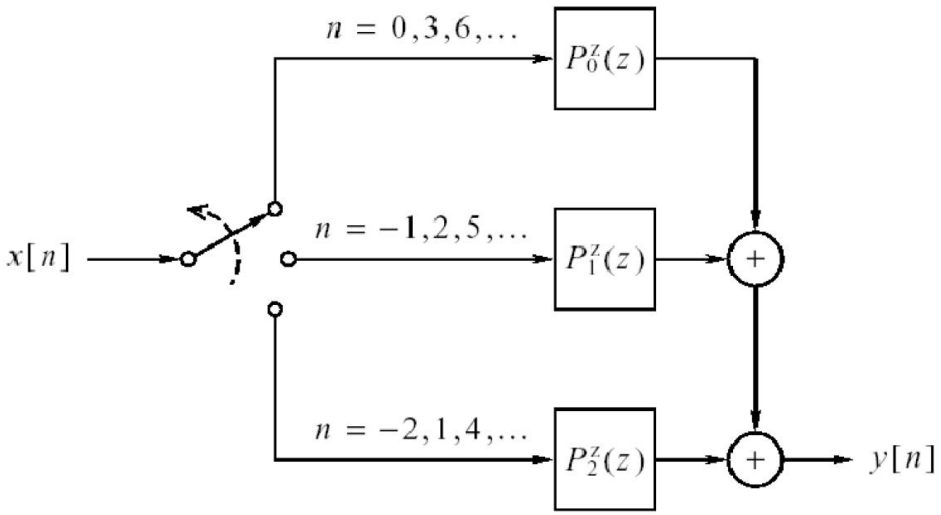


Source: Porat, 1996.

- A decimation filter is split to M sub-filters with tap size of N/M.
- Inputs are multiplexed, outputs are the sum of all sub-filters.
- Difficult to conceptualize multiplier sharing.

Government sponsorship acknowledged

Polyphase Decomposition



Source: Porat, 1996.

- A decimation filter is split to M sub-filters with tap size of N/M.
- Inputs are multiplexed, outputs are the sum of all sub-filters.
- Difficult to conceptualize multiplier sharing.

Government sponsorship acknowledged

Outline

Government sponsorship acknowledged

Each Output is a Thread

Pseudocode of an Output Thread

```
y(n) = 0
for i from n-N+1 to n:
  y(n) = y(n) + x(i)*h(N-i)
```

- Each thread represents the finite convolution for an output.
- Threads spawn and terminate at a fixed output rate.
- Multiplier sharing becomes a static scheduling problem.

Government sponsorship acknowledged

Outline

Government sponsorship acknowledged

Thread Decomposition (TD) Diagram

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
y0	*h4	*h3	*h2	*h1	*h0					
y1		*h4	*h3	*h2	*h1	*h0				
y2			*h4	*h3	*h2	*h1	*h0			
y3				*h4	*h3	*h2	*h1	*h0		
y4					*h4	*h3	*h2	*h1	*h0	
y5						*h4	*h3	*h2	*h1	*h0

- N=5, L=1, M=1
- Columns represent the required multiplications per input cycle.
- Rows represent output threads.

Government sponsorship acknowledged

Decimation TD Diagram

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
y0	*h4	*h3	*h2	*h1	*h0					
y1			*h4	*h3	*h2	*h1	*h0			
y2					*h4	*h3	*h2	*h1	*h0	
y3							*h4	*h3	*h2	*h1
y4									*h4	*h3

- N=5, L=1, M=2.
- Outputs removed by decimation are not shown.
- At most three multiplications needed per input.

Government sponsorship acknowledged

Interpolation TD Diagram

	x0	x0	x1	x1	x2	x2	x3	x3	x4	x4
y0	*h4	*h3	*h2	*h1	*h0					
y1		*h4	*h3	*h2	*h1	*h0				
y2			*h4	*h3	*h2	*h1	*h0			
y3				*h4	*h3	*h2	*h1	*h0		
y4					*h4	*h3	*h2	*h1	*h0	
y5						*h4	*h3	*h2	*h1	*h0

- N=5, L=2, M=1.
- Two columns represent one input cycle.
- N unique multiplications per input cycle.

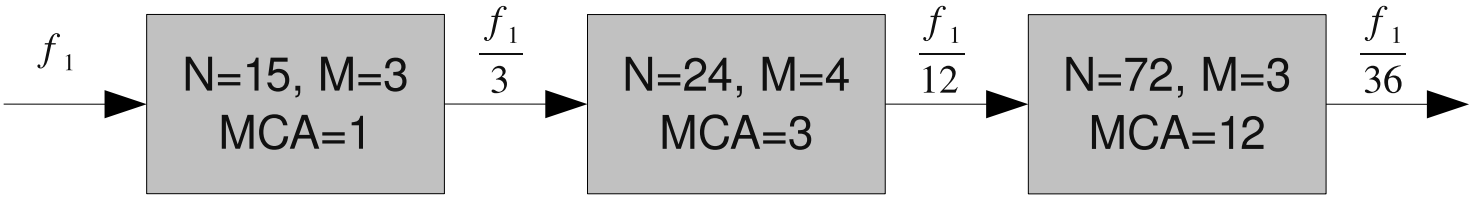
Government sponsorship acknowledged

Optimal Multiplier Sharing

- Multiplications per input cycle N_{mpi} are clearly indicated in the columns of TD diagrams.
- Final factor to consider: how fast is the multiplier clock f_{mult} relative to the input rate f ?
- Min. multiplier required: $N_{mult} = N_{mpi} \frac{f}{f_{mult}}$
- $\frac{f_{mult}}{f}$: Multiplier Clock Advantage (MCA).

Government sponsorship acknowledged

More on Multiplier Clock Advantage (MCA)



- MCA exploitation leads to very efficient designs in multi-staged MRFIR filter banks (see above).
- All multipliers run at clock rate $f_{mult} = f_1$
- The first stage needs $\frac{15}{3} = 5$ multipliers (MCA=1).
- With an MCA of 3, the second stage needs 2 multipliers
- Despite being a 72-tap filter, the last stage only needs 2 multipliers.

Government sponsorship acknowledged

Outline

Government sponsorship acknowledged

Advantages

- Thread Decomposition (TD) facilitates global, arbitrary multiplier sharing.
- Polyphase Decomposition (PD) facilitates sharing by multiplexing sub-filters, or locally within sub-filters.
- TD exploits the MCA to the fullest extent, while PD is tightly coupled with M and L.
- The MCA effect is more substantial for cascaded MRFIR filter banks.
- Result: higher throughput (performance) per multiplier.

Government sponsorship acknowledged

Advantages

- Thread Decomposition (TD) facilitates global, arbitrary multiplier sharing.
- Polyphase Decomposition (PD) facilitates sharing by multiplexing sub-filters, or locally within sub-filters.
- TD exploits the MCA to the fullest extent, while PD is tightly coupled with M and L.
- The MCA effect is more substantial for cascaded MRFIR filter banks.
- Result: higher throughput (performance) per multiplier.

Government sponsorship acknowledged

Summary

- The Thread Decomposition Diagram translates an MRFIR design into a static scheduling problem of the multipliers.
- The minimum number of multipliers is determined by the number of multiplications per input (N_{mpi}) and the multiplier clock advantage ($\frac{f_{mult}}{f}$).
- Thread Decomposition is designed to accomplish the goal of minimum multiplier count.

Government sponsorship acknowledged

Acknowledgements

- The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.
- We would like to thank the following JPL engineers for their contributions: Charles Le, Ian Tan, Sunant Katanyoutanant, Martin Le.

 Government sponsorship acknowledged