Energy Optimizations for FPGA-based 2-D FFT Architecture

Ren Chen and Viktor K. Prasanna

Ming Hsieh Department of Electrical Engineering

University of Southern California

Ganges.usc.edu/wiki/TAPAS

Outline

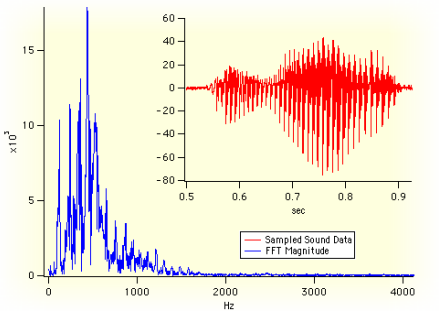
- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

Outline

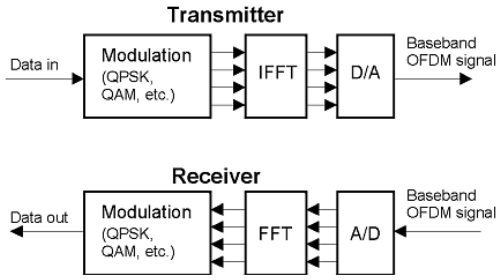
- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

Fast Fourier Transform (FFT) : 1-D and 2-D

Signal Processing

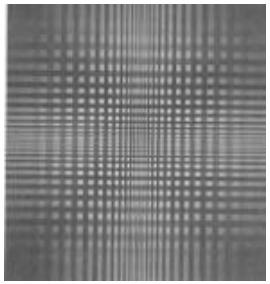


Audio analysis



OFDM System

Image Processing



Frequency domain in images

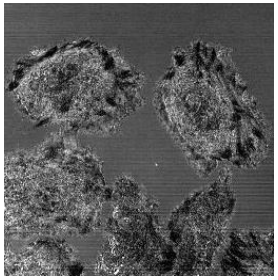
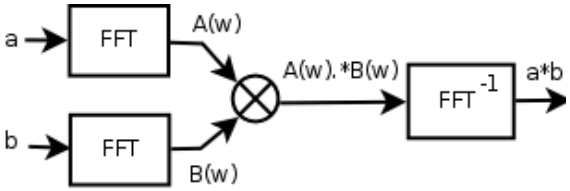
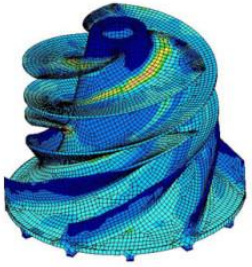


Image filtering

Scientific computing



Multiplication of large integers



Partial differential equations

1-D Fast Fourier Transform (FFT)

- Discrete Fourier Transform
  - Compute an approximation of the Fourier Transform on a discrete set of frequencies from a discrete set of time samples

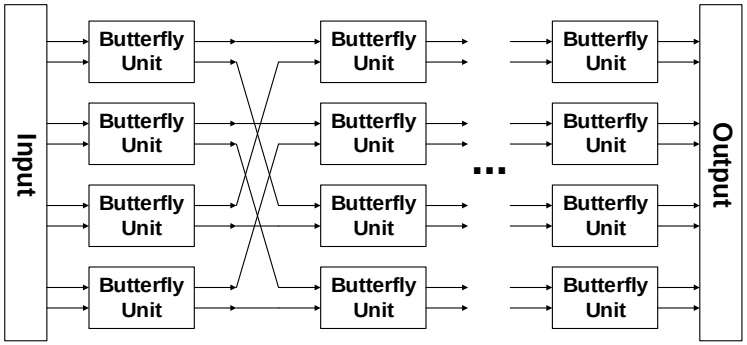
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} \text{ for } k = 0, 1, \cdots, N-1$$

  - Where k is the index of the discrete frequencies and n is the index of time samples

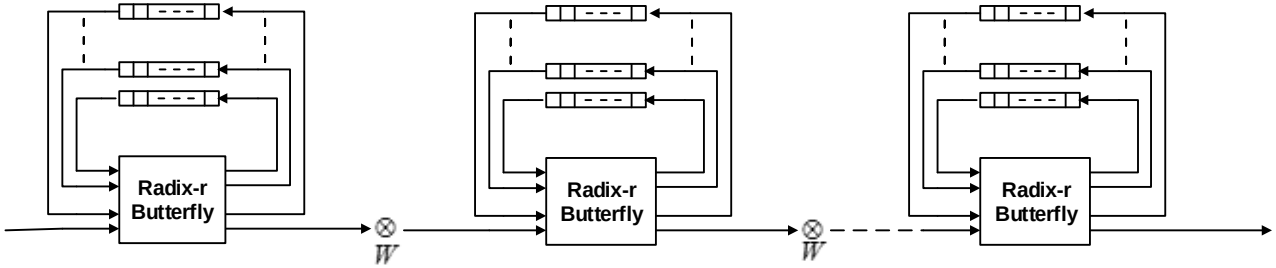
- Fast Fourier Transform
  - Divide and conquer algorithm
  - Much faster with complexity of $O(N \log N)$
  - Well know algorithms: Cooley-Tukey FFT, Prime-factor FFT, Rader's FFT, etc..

Traditional Fast Fourier Transform Architectures

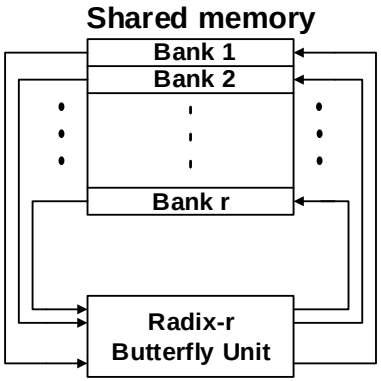
- Pipeline architecture
  - Pipeline ⟹ High performance per area
  - Delay feedback/commutator based
- Parallel architecture:
  - Parallel process ⟹ High throughput
- Shared memory architecture
  - Lowest throughput, highest performance per area



(b) Parallel Architecture



(a) Pipeline Architecture



(c) Shared memory Architecture

Memory Access in 2-D FFT

- Row-column 2-D FFT pseudo code

```
Void fft1d(complex [[*]] u)   {… Sequential FFT …}

Complex [[,*]] a = new complex [[x, N]];
Complex [[,*]] b = new complex [[N, x]];
… Initial values in 'a'

Overall (I = x for :)
    fft1d(a [[I,:]]);
```

→ **Row-wise 1D FFT**

Adlib.remap (b,a);

```
Overall (I = x for :)
    fft1d(b [[:,i]]);
```

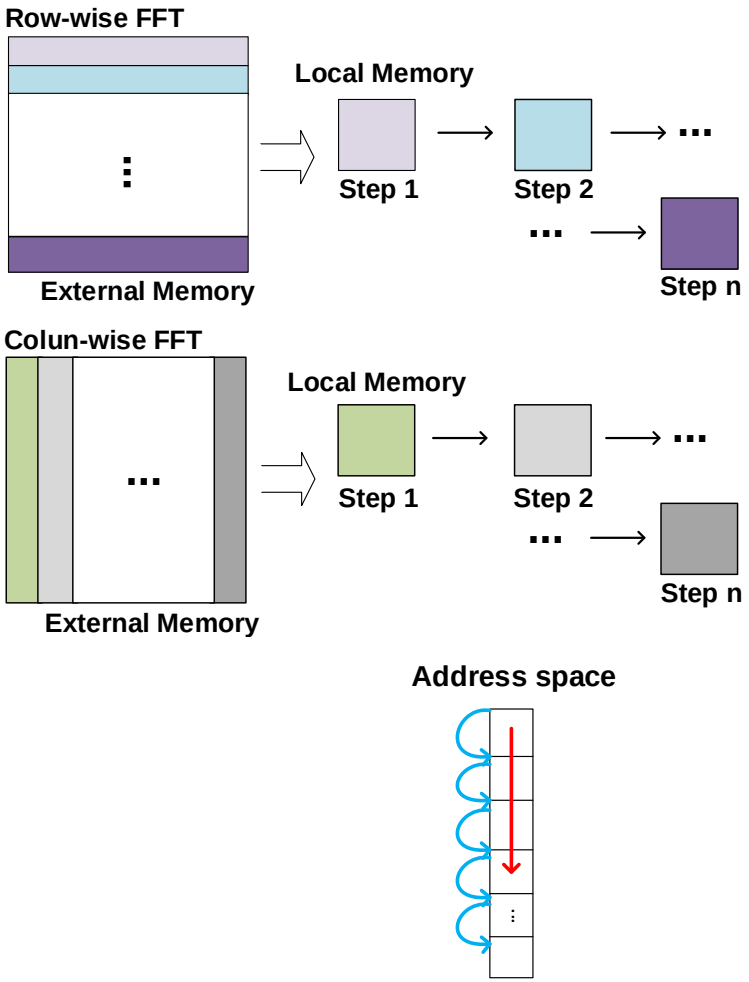
→ **Column-wise 1D FFT**

↓

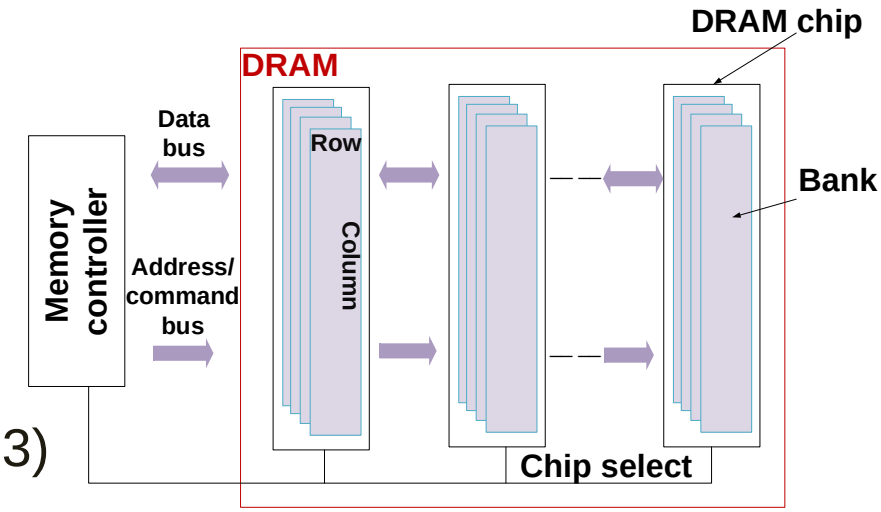
Strided memory access

↓

DRAM row activation



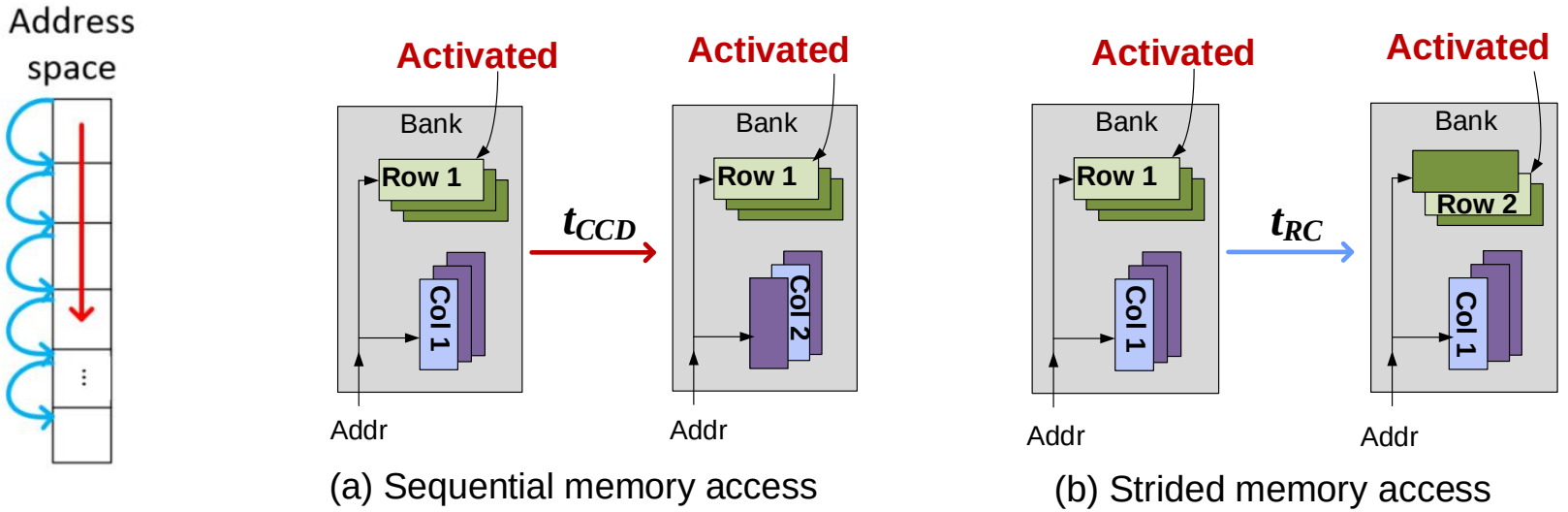
DRAM Organization and Timing



- DDR3 chip organization
  - Usually 8 banks / chip
  - 2 chips / DRAM bus
  - $n = 8K, 16K, 32K$ rows for each bank
  - $m = 1K, 2K$ columns for each row
  - A burst of 8 data transfers per read/write
- DDR3 timing constraints (micron DDR3)
  - t_{RCD} open/active a specific row, $\approx 15\ ns$
  - t_{CCD} minimum time between successive accesses to **the same bank and row**, $\approx 5\ ns$
  - t_{RC} minimum time between issuing two successive activate commands in **a single bank**, $\approx 40\ ns$
  - t_{RRD} minimum time between successive activate commands to different banks, $\approx 8\ ns$
  - t_{RP} precharge the long wires before switching to the next, $\approx 15\ ns$

DRAM Row Activation Cost (1)

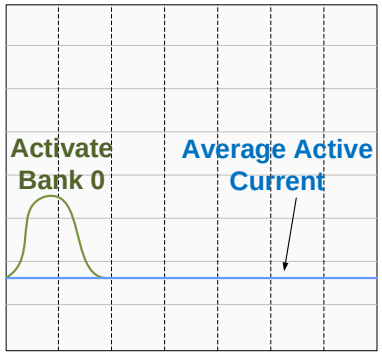
- Latency and throughput
  - Row activation cost determined by
    - $t_{RC} \approx 40\ ns > t_{CCD} \approx 5\ ns$
  - Column-wise 1-D FFT → **strided memory access** → t_{RC} for **DRAM row activation**
  - Also FPGA device needs to be idle waiting for data → lower throughput



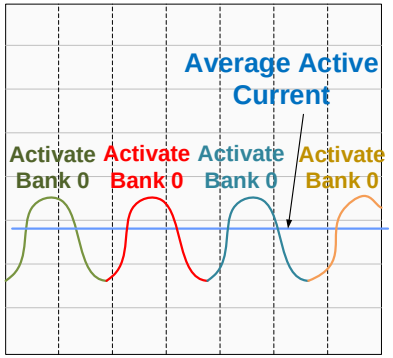
DRAM Row Activation Cost (2)

- Power
  - DRAM Activate power
    - P_{ACT} **DRAM activate power**
  - DRAM RD/WR/Term power (P_{acc})
    - P_{RD} DRAM average read power
    - P_{WR} DRAM average write power
    - P_{rIO} DRAM read I/O termination power
    - P_{wODT} DRAM write on-die termination power
  - **DRAM Background power (P_{bg})**
    - P_{ACT_STBY} DRAM activate standby power
    - P_{PRE_STBY} DRAM precharge standby power
    - P_{ACT_PDN} DRAM activate power down power
    - P_{PRE_PDN} DRAM precharge power down power
    - P_{REF} DRAM refresh power

increase



Open page mode (Active to Active is determined by page hit rate)



Closed Page, with Bank interleaving (Active to Active time = t_{RRD})

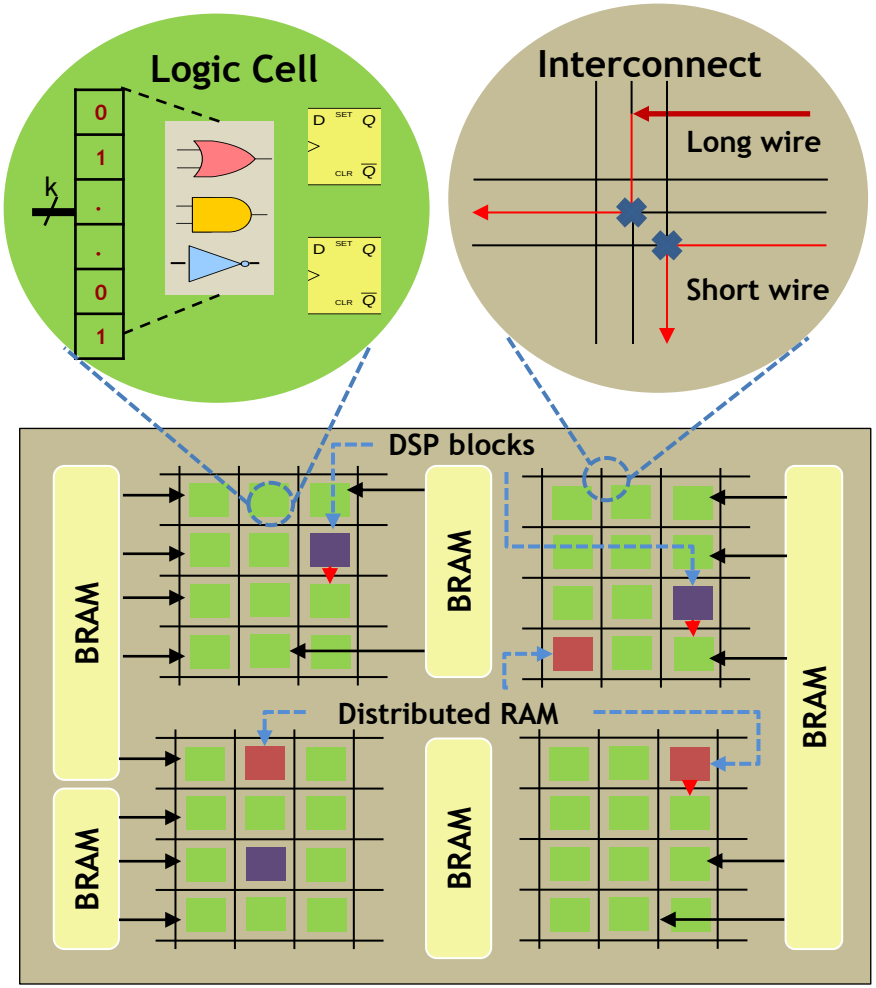
Conclusion: DRAM row activations is the dominating factor in energy consumption and performance of the external memory

Outline

- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

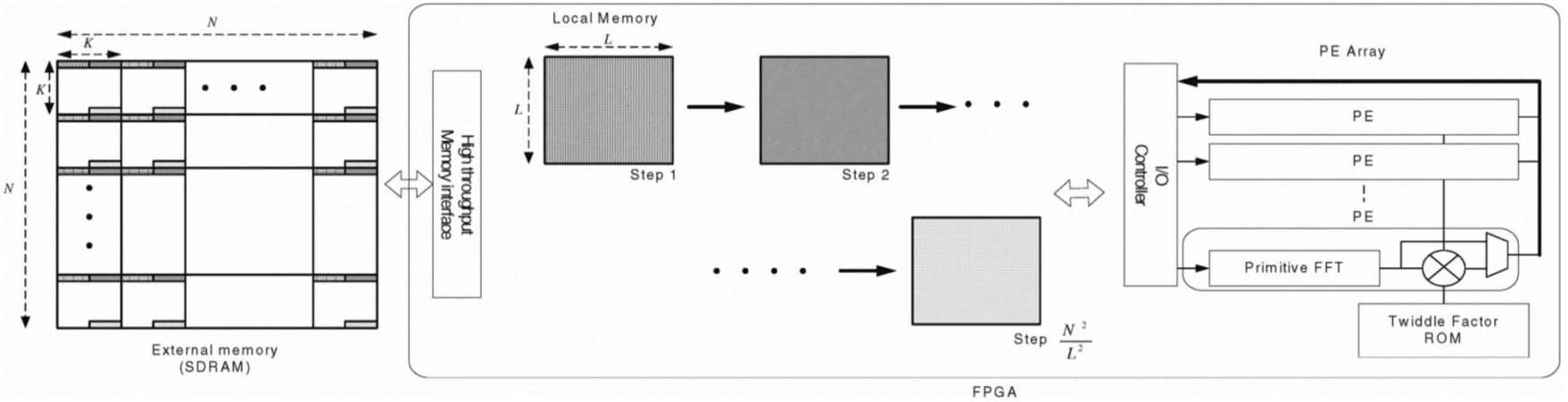
Platforms- FPGA

- Field Programmable Gate Arrays
  - Configurable logic blocks
  - Programmable interconnect
  - Programmable on-chip memory
- Logic block functionality
  - Pure logic (AND, OR, etc.)
  - Shift register
- Memory hierarchy
  - LUT-based distributed RAM
  - Block RAM
  - External memory through I/O
  - Huge on-chip bandwidth (2.8Tbps)



Related Work (1)

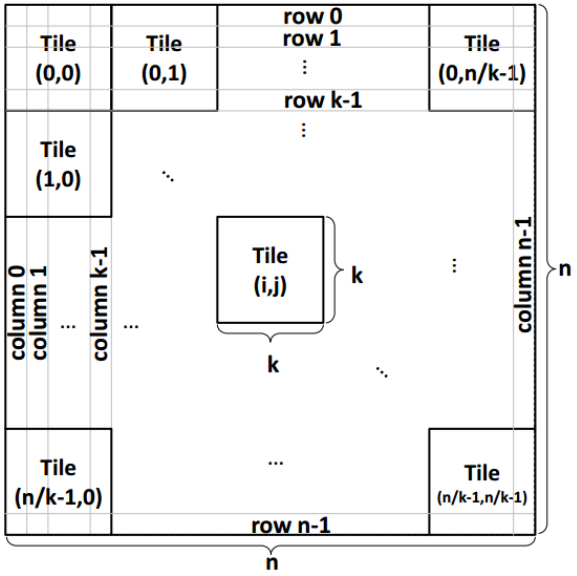
- Minimized # of DRAM row activations
- # of ops increases from $2N^2 \log N$ to $2N^2(1 + \log N)$ → more power and latency



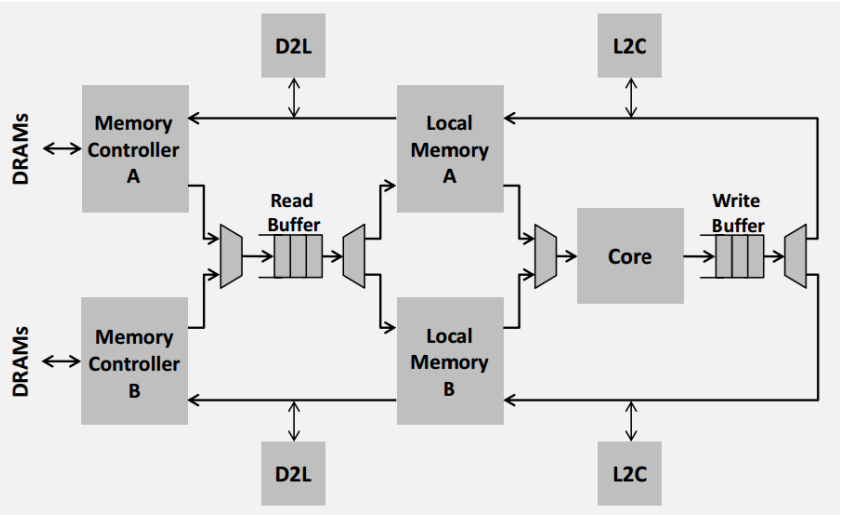
J. S. Kim, C.-L. Yu, L. Deng, S. Kestur, V. Narayanan, and C. Chakrabarti, "FPGA architecture for 2D Discrete Fourier Transform based on 2D decomposition for large-sized data," in Proc. of IEEE Workshop on Signal Processing Systems, Oct 2009, pp. 121–126.

Related Work (2)

- Improves the external memory bandwidth utilization
- Energy performance is not considered



Data layout in DRAM



Architecture Overview

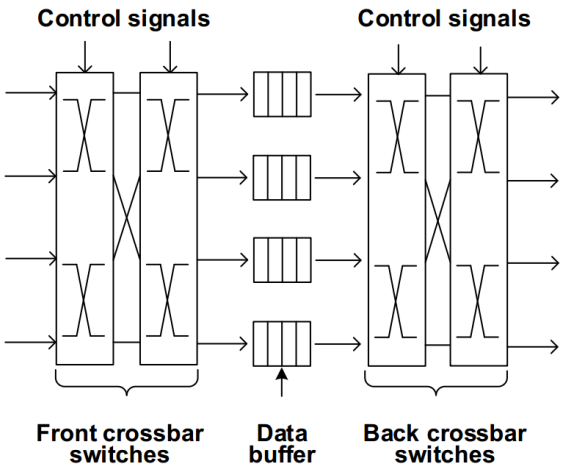
B. Akin, P. Milder, F. Franchetti, and J. Hoe, “Memory bandwidth efficient two-dimensional fast fourier transform algorithm and implementation for large problem sizes,” FCCM ’12, April 2012, pp. 188–191.

Outline

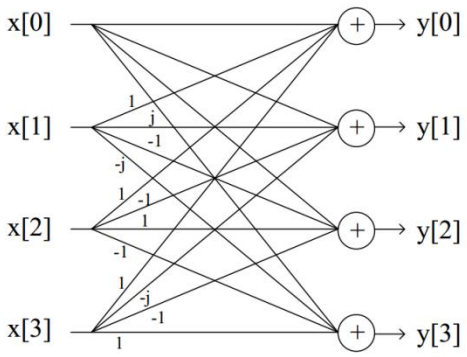
- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

Key Architectural Components

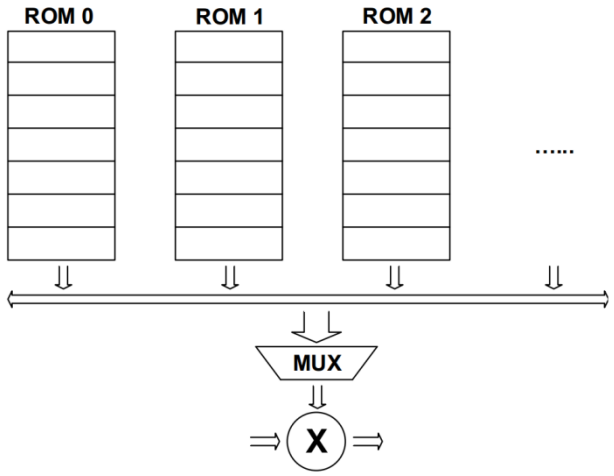
- Used for building a complete FFT design
- Each component responsible for a key task



(a) Data path permutation unit



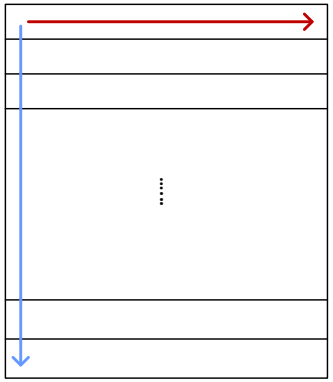
(b) Radix-4 block



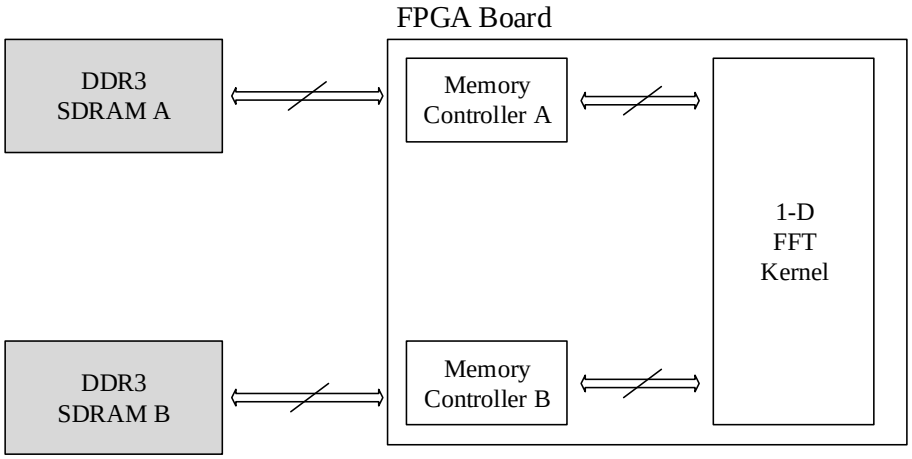
(c) TWC (twiddle factor computation) unit

2-D FFT Baseline Architecture

- 1-D FFT kernel with flexible data parallelism
- Based on row-column algorithm
- Two DDR3 chips used for overlapping the access latency



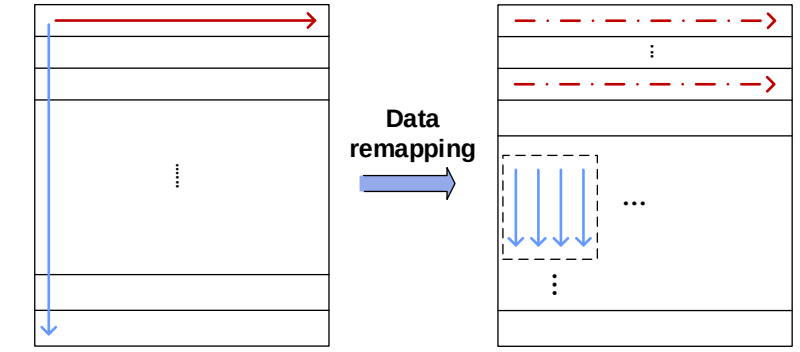
DRAM access pattern



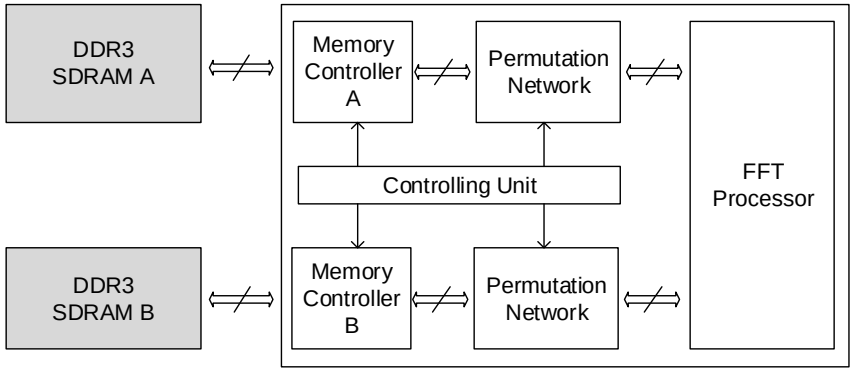
Proposed DRAM-based 2-D FFT architecture

2-D FFT Optimized Architecture

- Balance the throughput by choosing the data parallelism of the 1-D FFT kernel
- Employ a permutation network
  - Enable local data transposition
  - Generate correct data order for 1-D FFT kernel



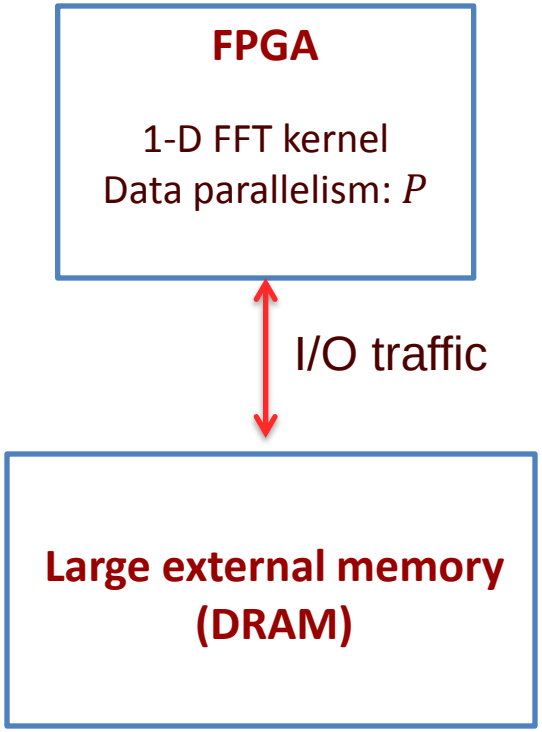
DRAM access in baseline **DRAM access in opt. arch.**



Proposed DRAM-based 2-D FFT architecture

Throughput Balancing between FPGA and DRAM

- FPGA
  - Operating frequency: 200 MHz
  - On-chip computation time $\geq \frac{N \log N}{200 \times P}$
- DRAM
  - Each DRAM chip: 16-bit data pins
  - Maximum operating frequency: 800MHz
  - I/O time $\geq \frac{16 \times N}{800}$
- Throughput balance
  - Choose P to increase DRAM bandwidth utilization
  - Reduce latency cost brought by DRAM row activation



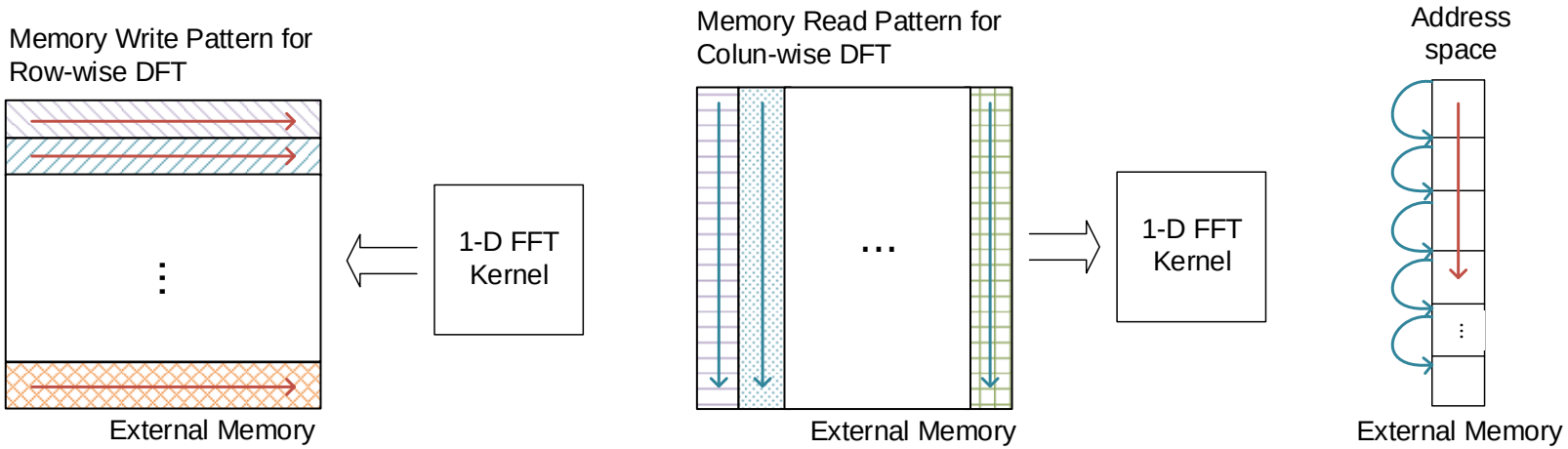
Outline

- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

Data Remapping on DRAM (1)

- Key ideas
  - Decompose the problem by data remapping: large stride → small stride
  - Permutation network: permute data on-chip
  - Permutation network: communication in 1-D FFT

Resource reuse ⟶ High efficiency

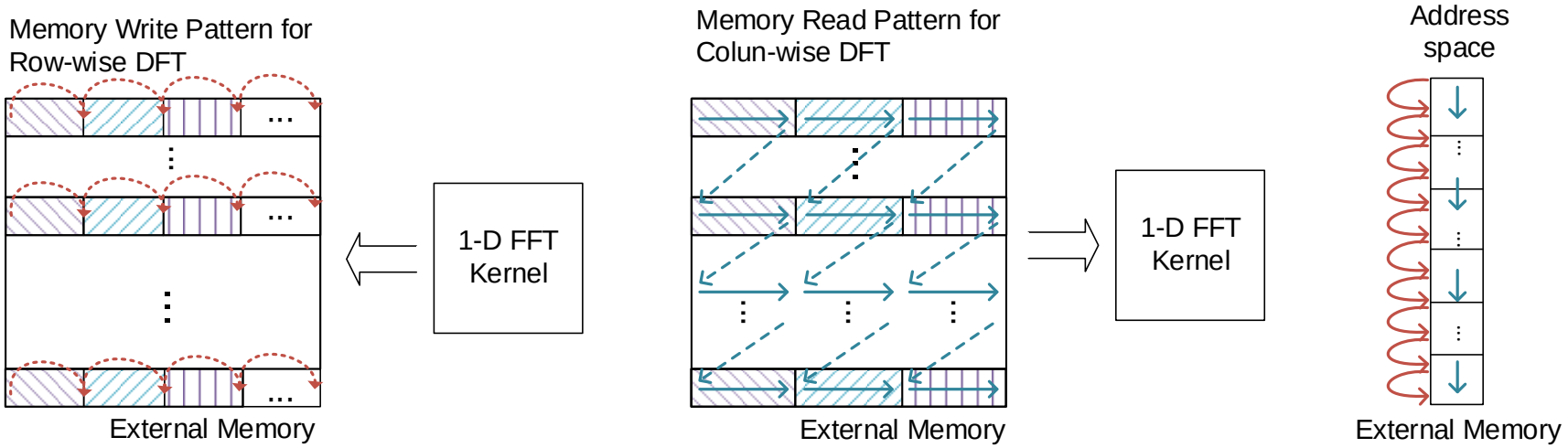


Before data remapping

Data Remapping on DRAM (2)

- Key ideas
  - Decompose the problem by data remapping: large stride → small stride
  - Permutation network: permute data on-chip
  - Permutation network: communication in 1-D FFT

Resource reuse ⟶ High efficiency



After data remapping

DRAM Energy Performance

	1024 x 1024 2-D FFT	4096 x 4096 2-D FFT	8192 x 8192 2-D FFT
Energy per read for column-wise FFT (Baseline)	3.43 nJ	5.48 nJ	6.42 nJ
Energy per read for column-wise FFT (Optimized)	1.76 nJ	2.31 nJ	2.62 nJ
Reduction percentage (Energy per read)	48.6%	57.8%	59.2%

Outline

- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

Experimental Setup

- FPGA Artix 7 XC7VX200T with -2L speed grade
- Vivado 2013.4
- Vivado Power Analysis Tool
- Micron DDR3 SDRAM System-Power Calculator

Performance Metric

- Energy Efficiency
  - Energy efficiency = $\dfrac{\text{number of real operations}}{\text{energy consumed by the design}}$
  - Energy consumed by the design
    = time taken by the design × average power dissipation of the design
    = $T \times P$
  - For N-point Radix-4 FFT,
    - number of real operations = $2N \log_2 N + \frac{9}{4} N \log_2 N$
    - P is obtained through simulation

Experimental Result (1)

	Baseline architecture (1-D FFT kernel)			Optimized architecture (1-D FFT kernel)			
FFT Size	Power (W)	Energy per point (nJ)	Energy efficiency (GFLOPS/W)	Power (W)	Energy per point (nJ)	Energy efficiency (GFLOPS/W)	Energy efficiency improvement
1024	2.56	6.4	5.9	0.71	1.9	19.6	3.3x
4096	3.42	8.6	5.4	1.04	2.6	17.9	3.3x
8192	4.31	10.8	4.7	1.39	3.5	15.8	3.4x

Experimental Result (2)

	Baseline architecture (2-D FFT architecture)			Optimized architecture (2-D FFT architecture)			
FFT Size	Power (W)	Energy per point (nJ)	Energy efficiency (GFLOPS/W)	Power (W)	Energy per point (nJ)	Energy efficiency (GFLOPS/W)	Energy efficiency improvement
1024 x 1024	7.28	18.2	2.09	1.83	4.6	8.31	3.9x
4096 x 4096	9.82	24.5	1.89	2.32	5.8	8.01	4.2x
8192 x 8192	11.28	28.2	1.79	2.51	6.3	8.06	4.5x

Outline

- Introduction
- Background and Related Works
- Design and Implementation
- Minimizing DRAM Energy
- Experimental Results and Analysis
- Conclusion and Future Work

Conclusion

- Conclusion
  - Throughput-balanced 2-D FFT architecture
    - Reduces energy consumption per point by up to 77.7%
    - Achieves up to 4.5x improvement in energy efficiency
  - Data remapping enabling on-chip local transposition
    - Reduces DRAM energy consumption by up to 59.2%
    - Achieves high DRAM page hit rate
- Future work
  - Design framework for automatic energy optimizations on FPGA

Thanks!

Questions?

Ganges.usc.edu/wiki/TAPAS