Generation of Custom DSP Transform IP Cores: Case Study Walsh-Hadamard Transform

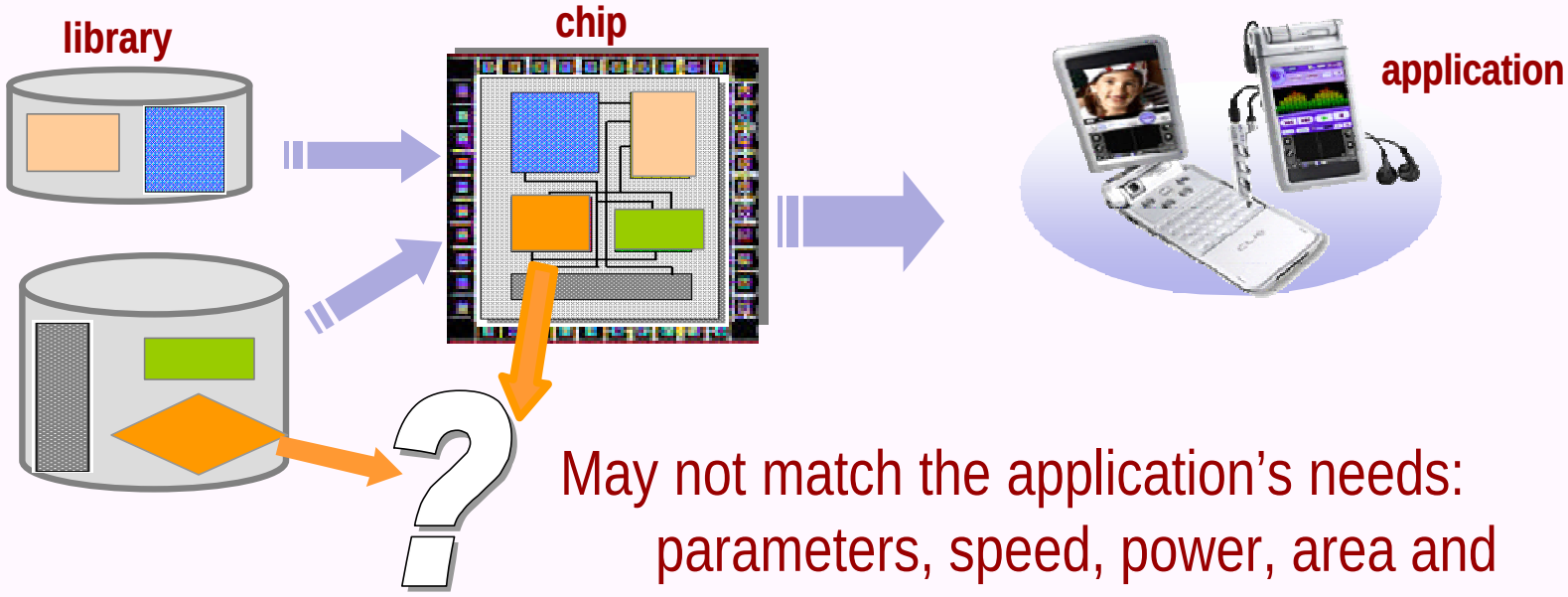
Fang Fang
James C. Hoe
Markus Püschel
Smarahara Misra

Carnegie Mellon University

Conventional Approach: Static IP Cores

- IP cores improve productivity and reduce time-to-market.
- e.g. Xilinx LogiCore library:

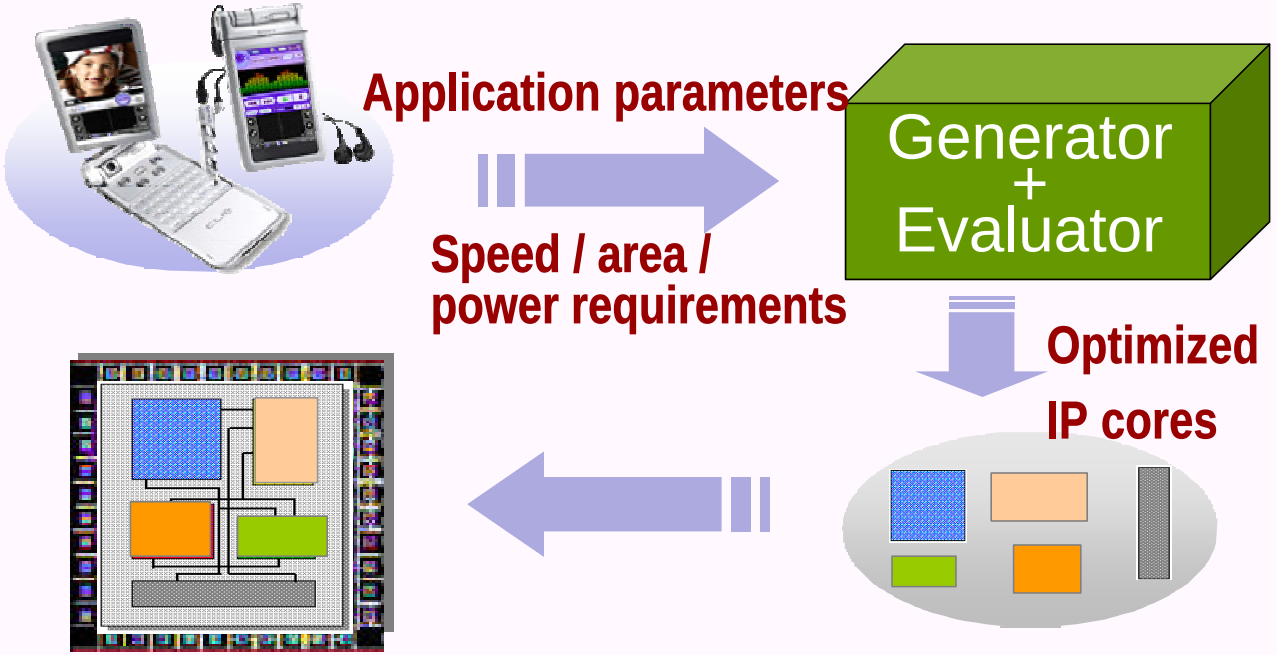
  FFT for N=16, 64, 256 and 1024 on 16-bit complex numbers



May not match the application's needs:
parameters, speed, power, area and their trade-off.

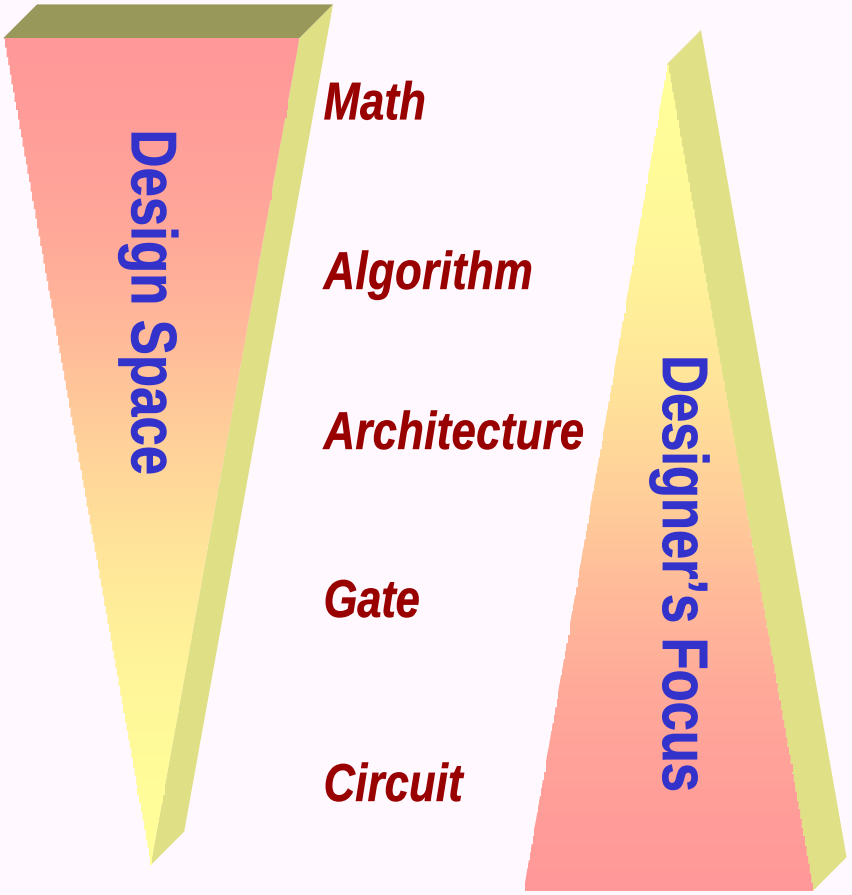
Alternative Approach: IP Core Generation

- **Generate IP cores to match specific application requirements**
  (speed, area, power, numerical accuracy, and I/O bandwidth…)



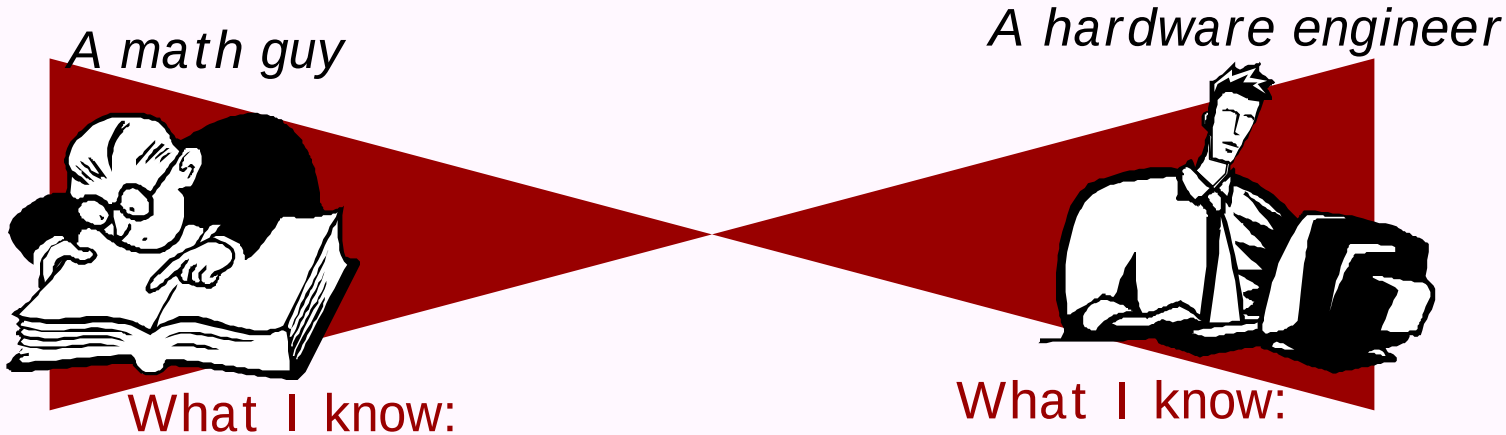
Design space

- DSP transform design can be studied at several levels.
- More math knowledge involved
  - Þ Bigger design space to explore.



Problem

- **Problem: gap between *transform mathematics* and *hardware design***



What I know:

Linear algebra
Digital signal processing
Adaptive filter theory ...

What I know:

Finite state machine
Pipelining
Systolic array ...

Bridge: Formula

- **Solution: - Formula representation of DSP transforms**
  - Automated formula manipulation and mapping

Formula example $DFT_8 = (F_2 \otimes I_4) \cdot D \cdot (I_2 \otimes (I_2 \otimes F_2 \cdots)) \cdot P$

A math guy

Formula
Representation
Manipulation
Mapping

A hardware engineer

What I know:

Linear algebra
Digital signal processing
Adaptive filter theory ...

What I know:

Finite state machine
Pipelining
Systolic array ...

Outline

- Introduction
- Technical Details (illustrated by WHT transform)
  - What are the degrees of design freedom?
  - How do we explore this design space?
- Experimental Results
- Summary and Future work

Walsh-Hadamard Transform

- **Why WHT?**
  - **Typical access pattern for a DSP transform**
  - **Close to 2-power FFT**
  - **Study important construct Ä**
- **Definition**

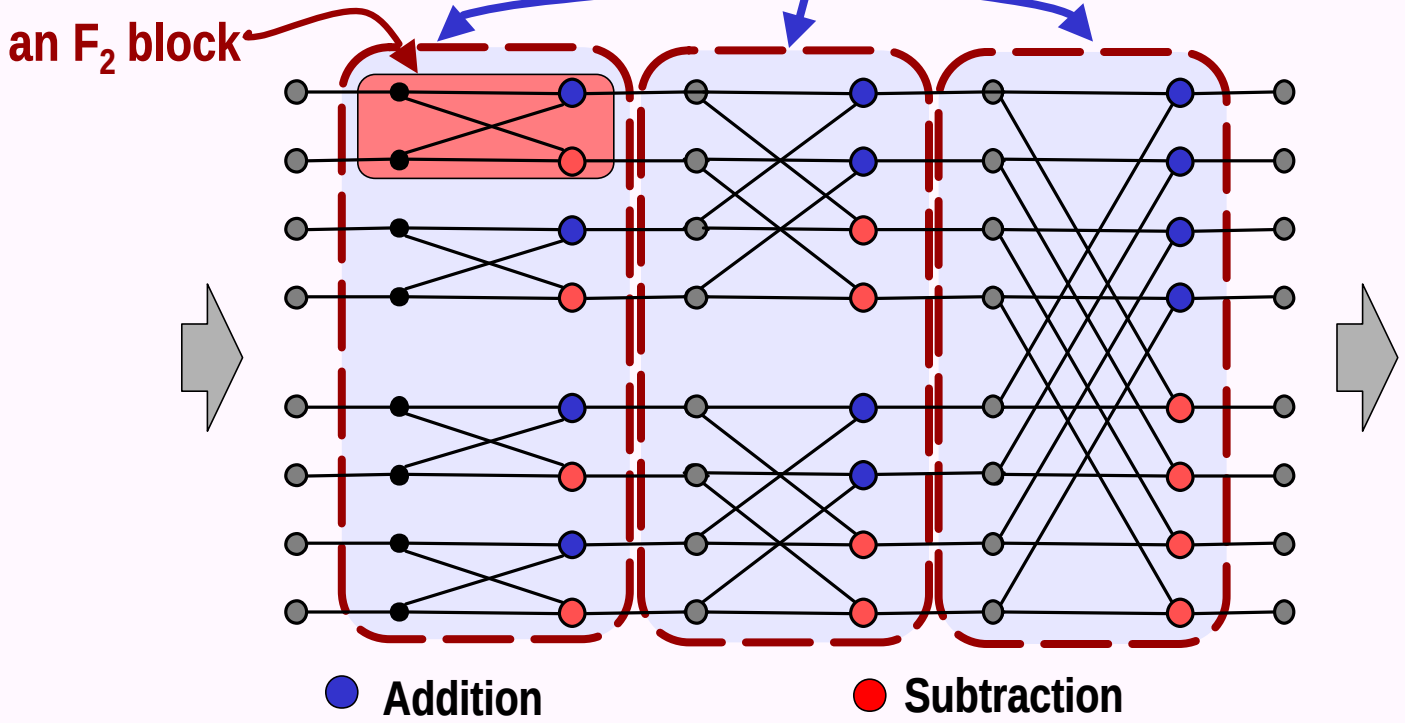
$$WHT_{2^n} = \begin{bmatrix} WHT_{2^{n-1}} & WHT_{2^{n-1}} \\ WHT_{2^{n-1}} & -WHT_{2^{n-1}} \end{bmatrix} \qquad WHT_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$WHT_{2^n} = \underbrace{F_2 \otimes F_2 \otimes \ldots \otimes F_2}_{\text{n fold}} \qquad F_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Tensor product $A \otimes B = [\, a_{k,l} \bullet B \,]$, where $A = [a_{k,l}]$

From Formula to Architecture

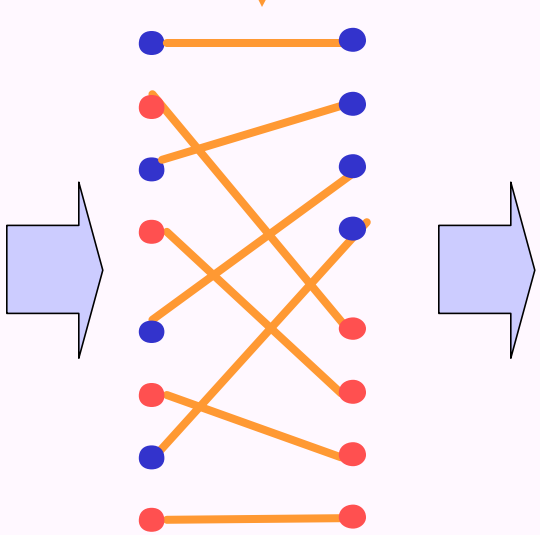
$$WHT_{2^3} = F_2 \otimes F_2 \otimes F_2$$
$$= (F_2 \otimes I_4)(I_2 \otimes (F_2 \otimes I_2))(I_4 \otimes F_2)$$



WHT_2^3

Pease Algorithm

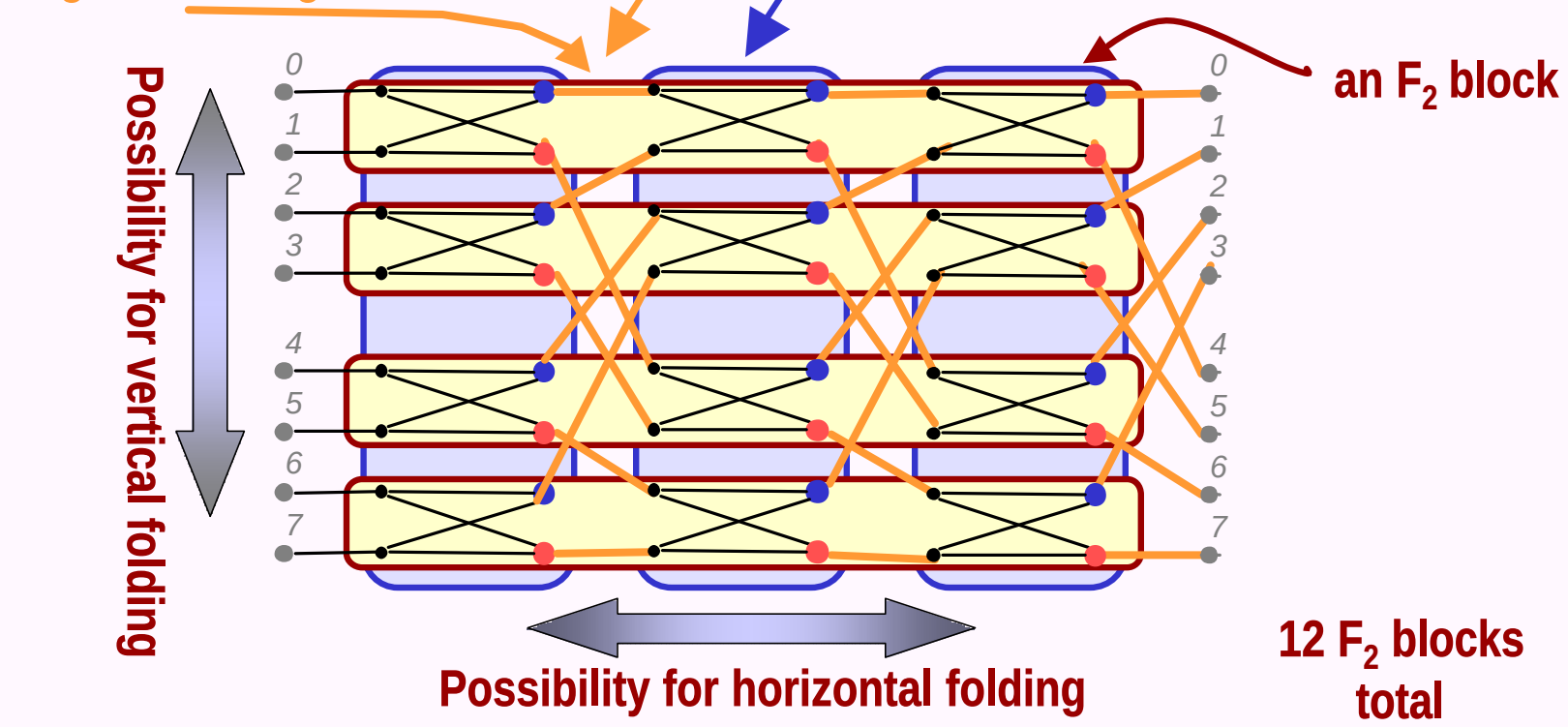
$$WHT_{2^3} = (F_2 \otimes I_4)(I_2 \otimes F_2 \otimes I_2)(I_4 \otimes F_2)$$
$$= L_2^8 (I_4 \otimes F_2) L_2^8 (I_4 \otimes F_2)\ L_2^8 (I_4 \otimes F_2)$$



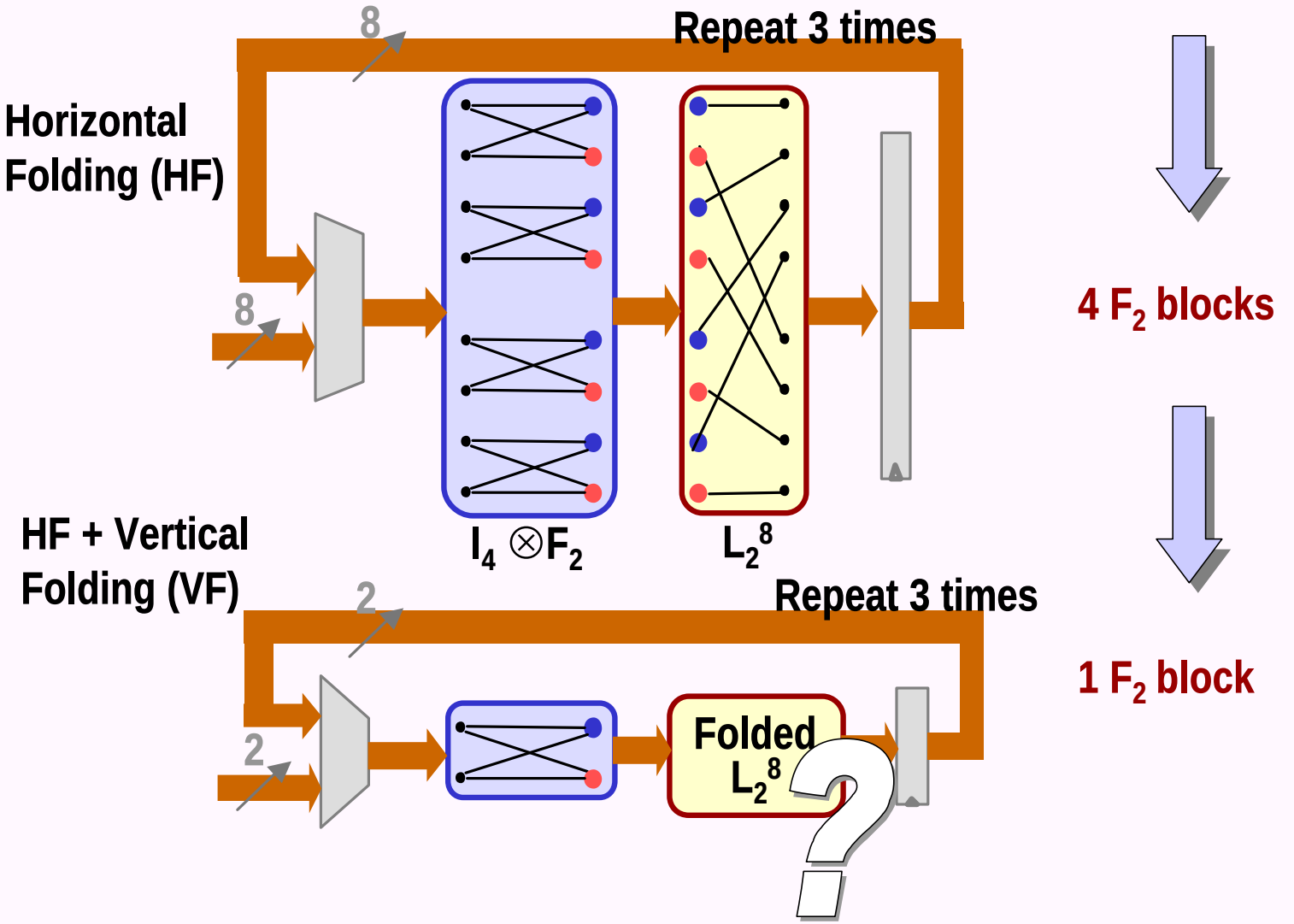
Stride permutation L_2^N

Pease Algorithm

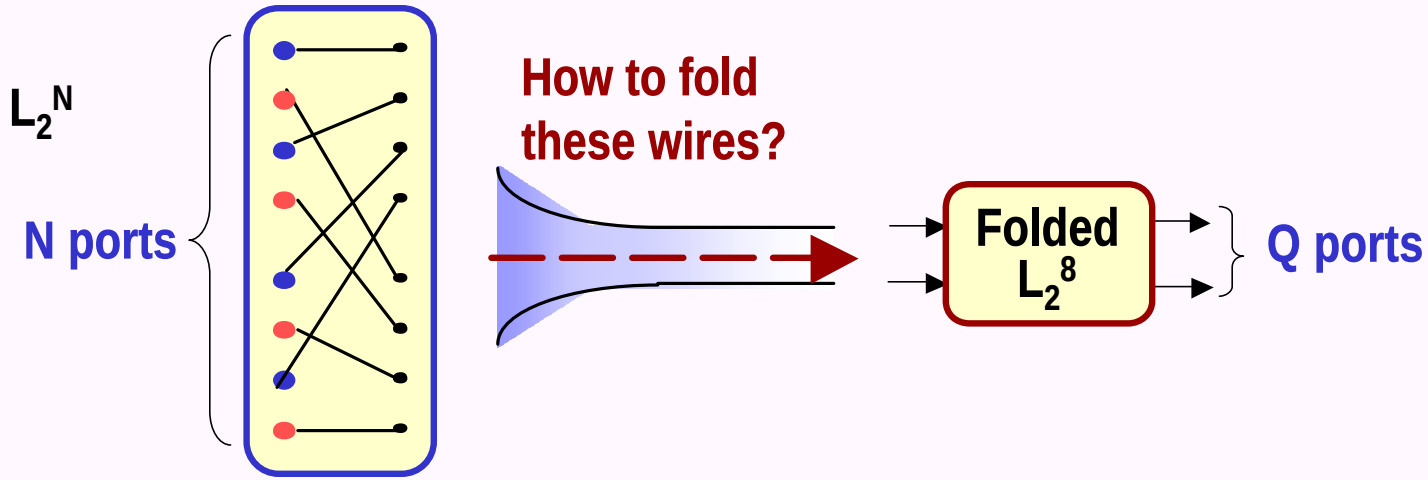
$$WHT_{2^3} = (F_2 \otimes I_4)(I_2 \otimes F_2 \otimes I_2)(I_4 \otimes F_2)$$
$$= L_2^8 (I_4 \otimes F_2) L_2^8 (I_4 \otimes F_2)\ L_2^8 (I_4 \otimes F_2)$$



Folding


Horizontal
Folding (HF)
8
Repeat 3 times
8
$I_4 \otimes F_2$
L_2^8
4 F_2 blocks
HF + Vertical
Folding (VF)
2
Repeat 3 times
2
Folded
L_2^8
1 F_2 block

Challenge in Vertical Folding



- Straightforward approach: Memory-based reordering
  - Extra control logic to reorder address
  - Computation speed is limited by memory speed
- Ad-hoc approach: Register routing
  - Hard to automate the process
- Our approach: formula-based matrix factorization

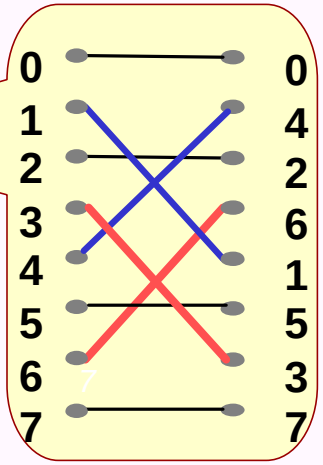
Factorization of Stride Permutation

$$L_2^N = (I_2 \otimes L_2^{N/2}) \bullet J_N$$

$$= (I_2 \otimes ((I_2 \otimes L_2^{N/4}) J_{N/2}) J_N$$

$$= \ldots\ldots$$

$$= (I_{N/Q} \otimes L_2^{Q}) \bullet \prod_{i=0}^{n-q-1} (I_{2^{k-q-i-1}} \otimes J_{2^{q+i+1}})$$

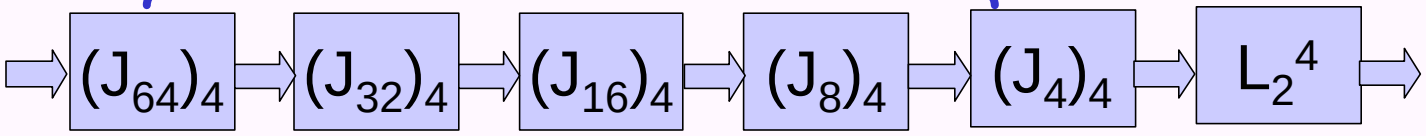


J_8

L_2^Q has Q input ports

Q=2^q, N=2^n

J_N can be easily folded [1]



Example of $(L_2^{64})_4$ (N=64, Q = 4)

[1]. J.H.Takala etc., "Multi-Port Interconnection Networks for Radix-R Algorithms", ICASSP01

Freedom in Horizontal Folding

- WHT_2^n has n horizontal stages in the flattened design
  - The divisors of n are all the possible folding degrees
  - *Example*: HF degrees of WHT_2^6 can be 1, 2, 3, 6

- Effects of more horizontal folding degree

Latency (cycle)	Same
Throughput (op / cycle)	Lower
Area	less adders, more muxs & wires
Speed	Not clear

Less pipeline depth Þ *lower throughput*

Freedom in Vertical Folding

- WHT_2^n has 2^n vertical ports in the flattened design
  - 1, 2, 4… 2^{n-1} are all possible folding degrees
  - *Example*: VF degrees of WHT_2^6 could be 1, 2, 4, … 32

- Effects of more vertical folding degree

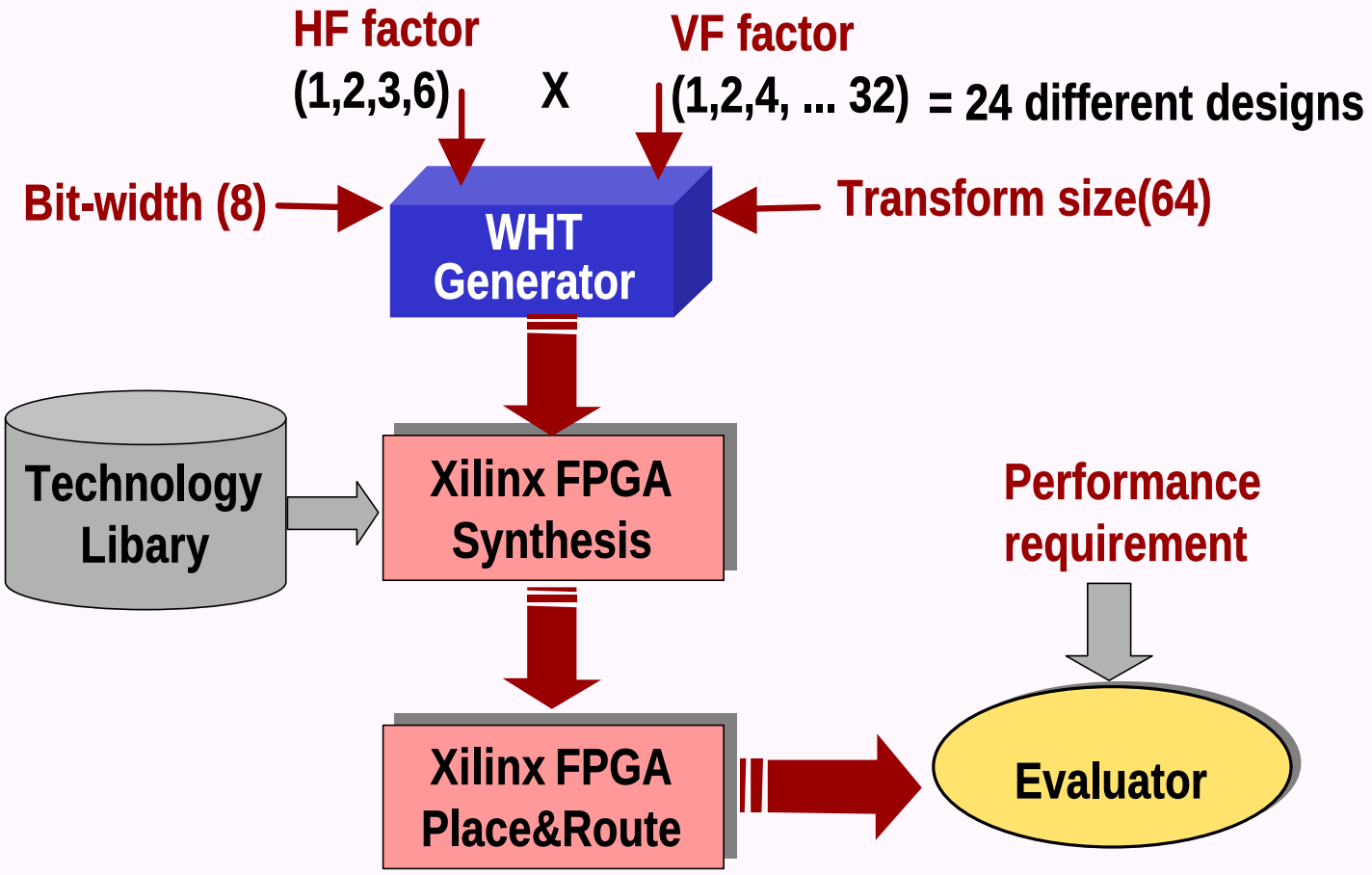
Latency (cycle)	Longer
Throughput (op / cycle)	Lower
Area	less adders, more regs & muxs
Speed	Not clear

Less I/O bandwidth Þ longer computation

Outline

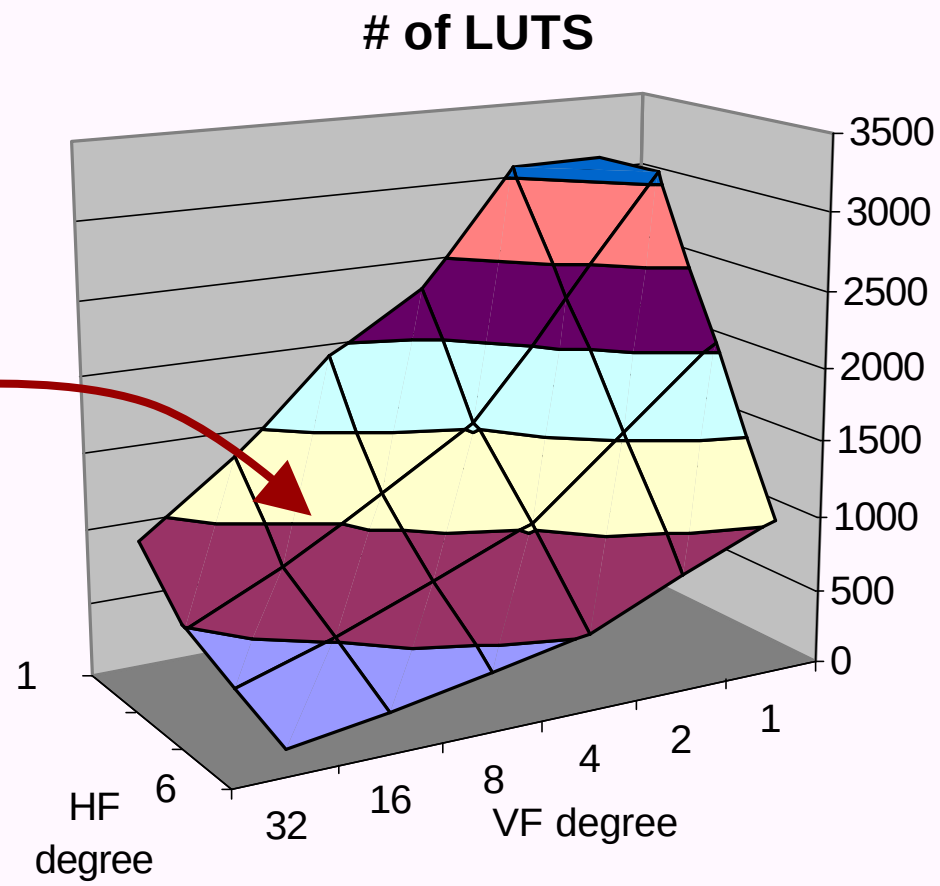
- Introduction
- Technical Details
- Experimental Results
- Summary and Future work

Design Space Exploration

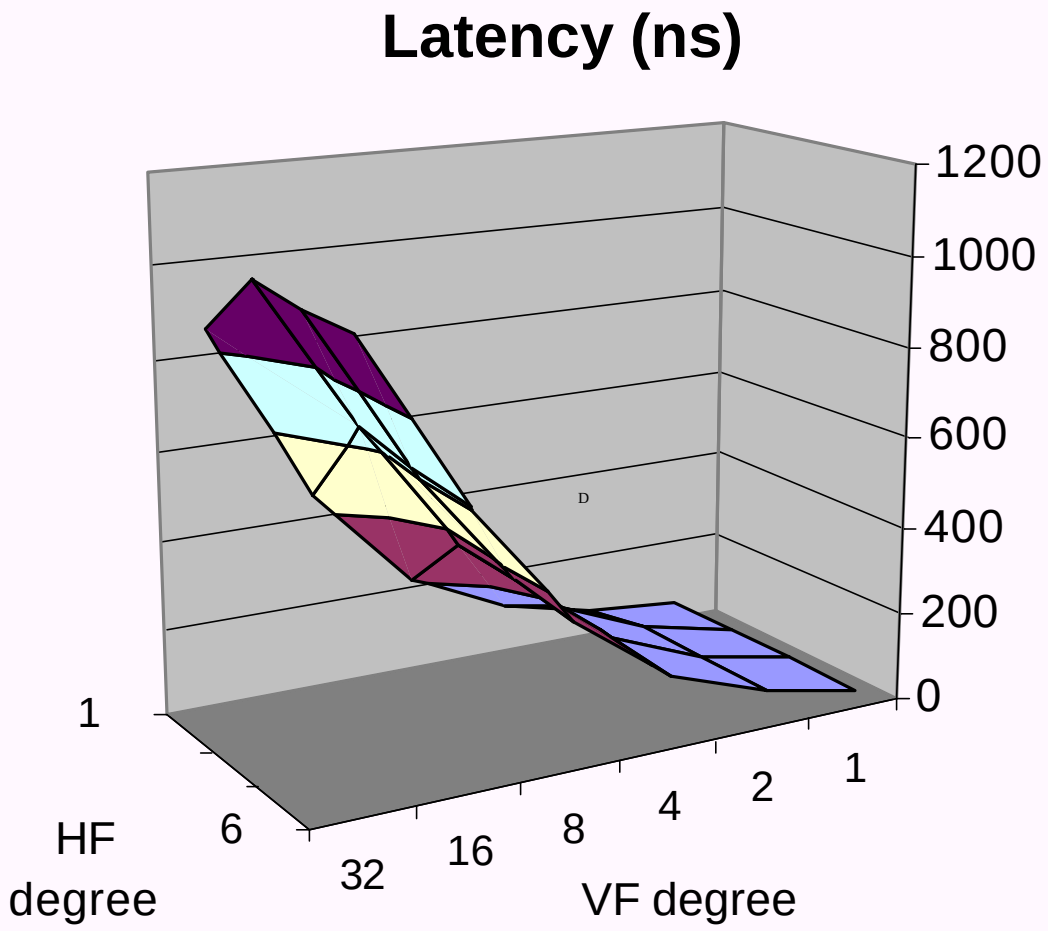


Area vs. Folding Degrees

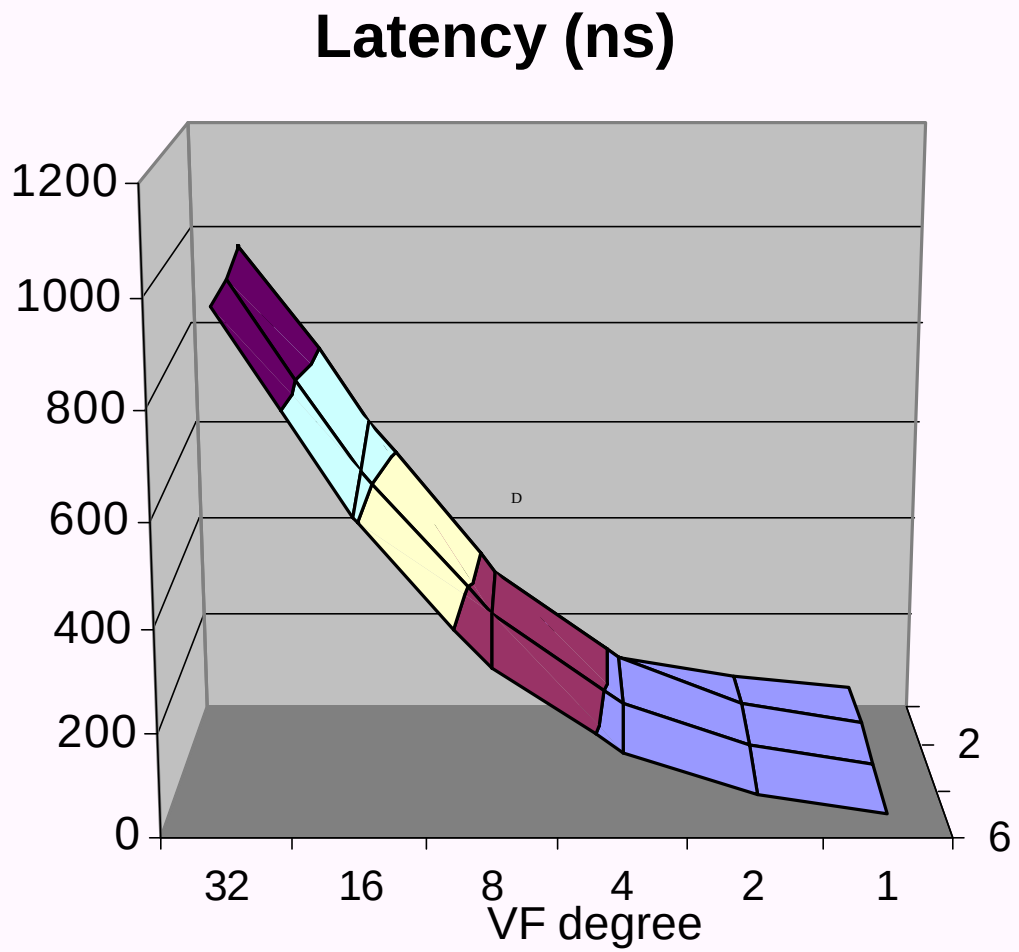
To achieve the same area, multiple folding options are available.



Latency vs. Folding Degrees (WHT_{64})

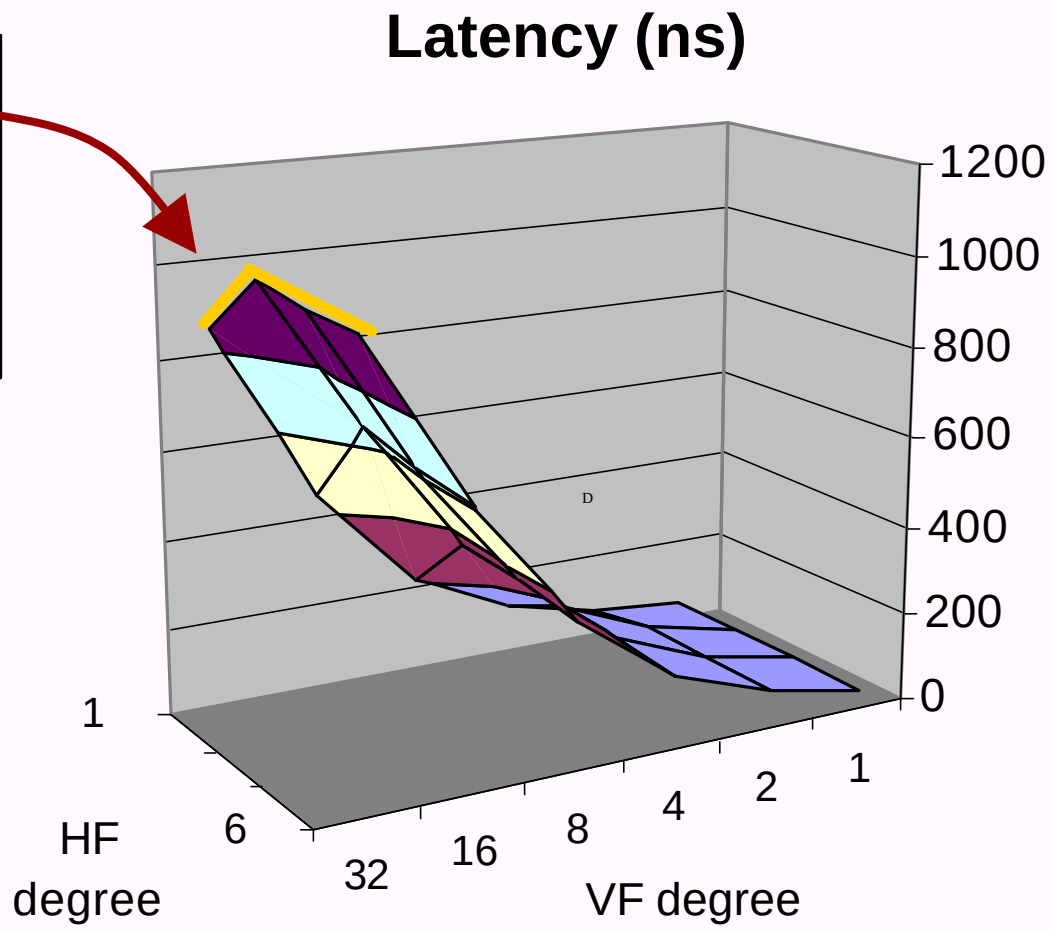


Latency vs. Folding Degrees (WHT_{64})



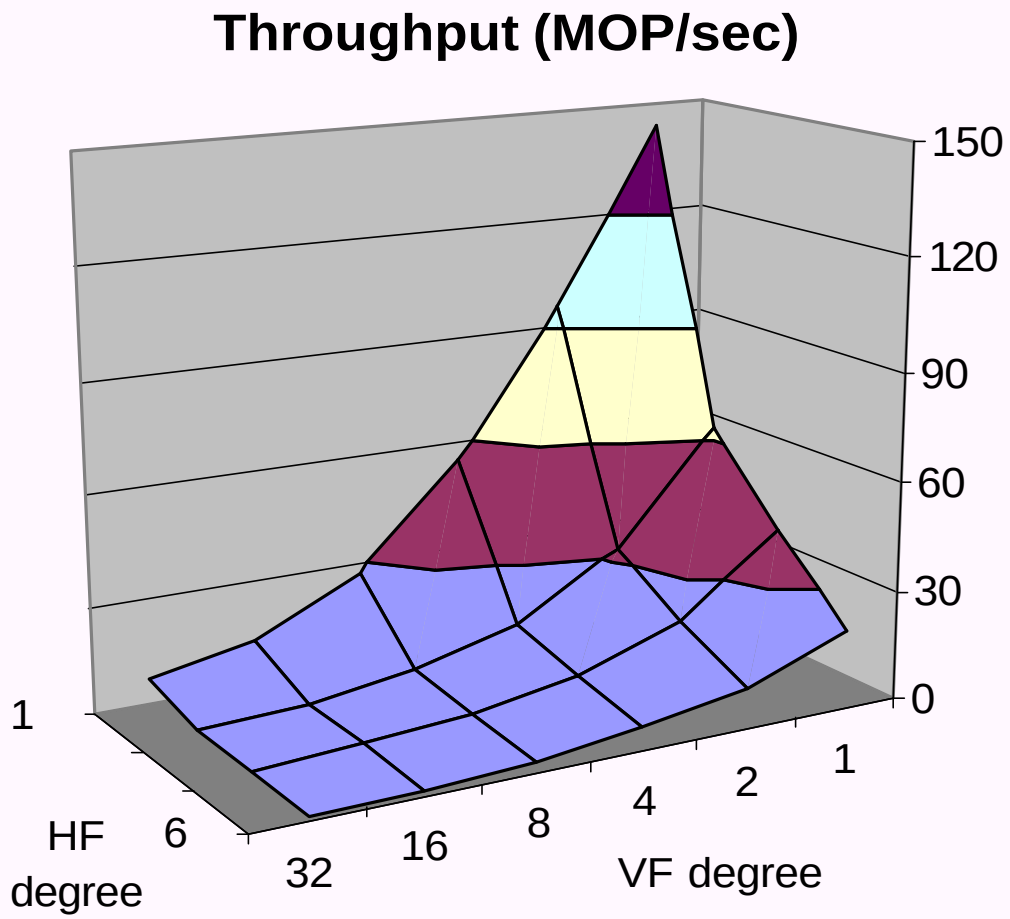
Latency vs. Folding Degrees (WHT_{64})

Latency is almost unaffected by HF, except comparing flattened design with folded design



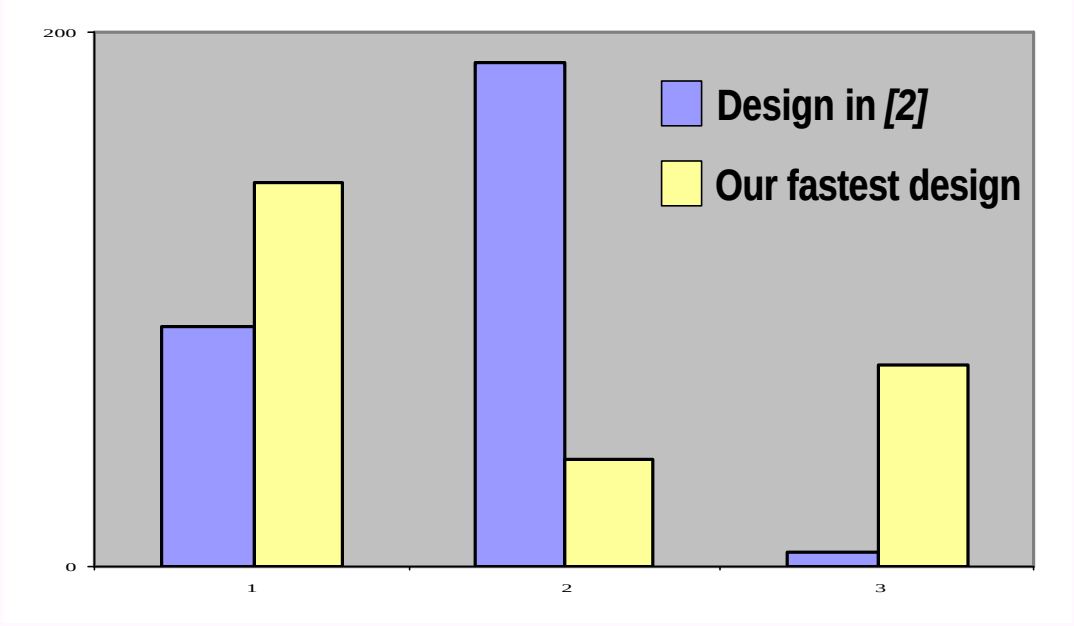
Throughput vs. Folding Degrees

Folding always lowers throughput



Comparison with an Existing Design

- WHT_8
  - 8 bit fixed-point
  - FPGA: Xilinx Virtex xcv1000e-fg680 Speed grade: -8
  - Compare our fastest generated designs against results reported by Amira, et al. *[2]*



Area (#of slices) Latency(ns) Throughput(MOP/s)

60% more area

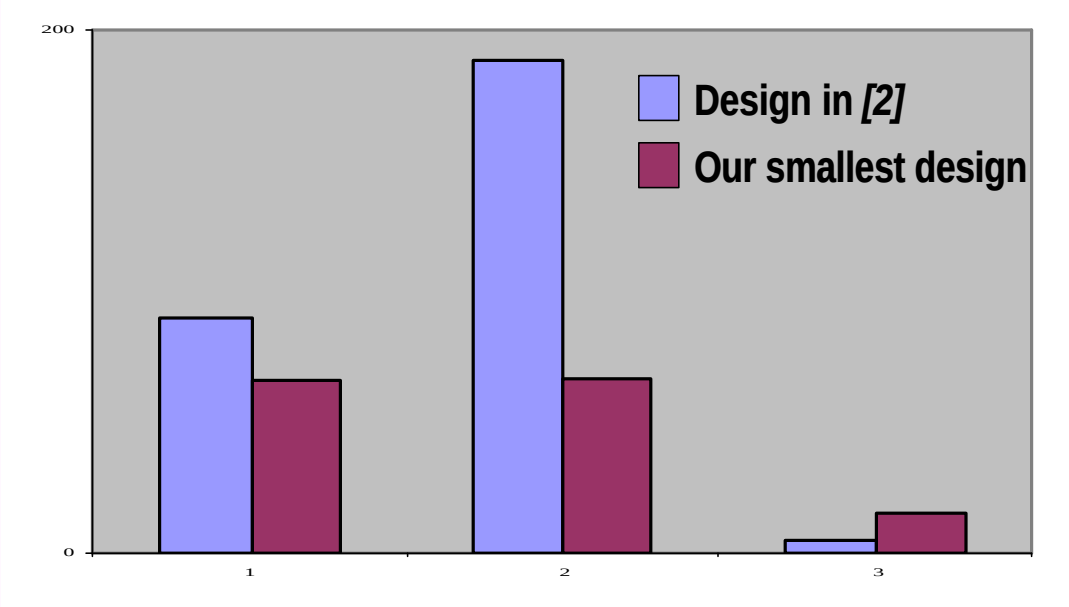
80% reduction in latency

13 times higher throughput

[2] A.Amira et al., "Novel FPGA Implementations of Walsh-Hardamard Transforms for Signal Processing", Visior Image and Signal Processing, IEE Proceedings- , Volume: 148 Issue: 6 , Dec. 2001

Comparison with an Existing Design

- WHT_8
  - 8 bit fixed-point
  - FPGA: Xilinx Virtex xcv1000e-fg680 Speed grade: -8
  - Compare our smallest generated designs against results reported by Amira, et al. *[2]*



Area (#of slices) Latency(ns) Throughput(MOP/s)

Less area

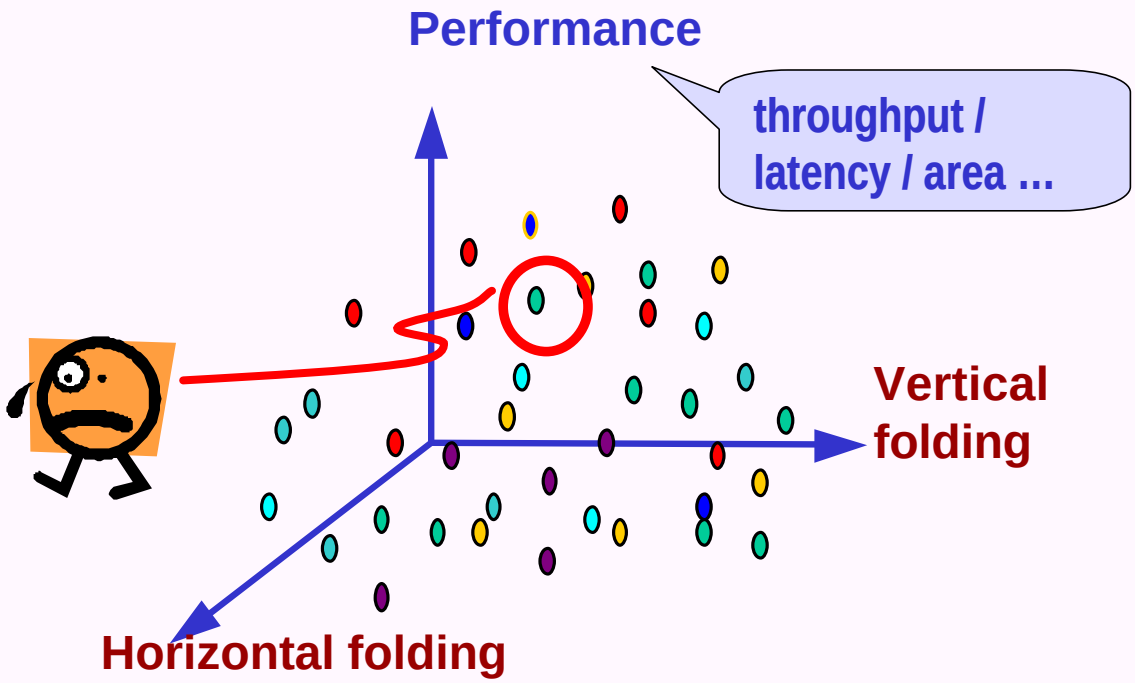
Shorter latency

Higher throughput

[2] A.Amira et al., "Novel FPGA Implementations of Walsh-Hardamard Transforms for Signal Processing", Visior Image and Signal Processing, IEE Proceedings- , Volume: 148 Issue: 6 , Dec. 2001

Summary

- Large performance variations over the design space of horizontal and vertical folding
- Automatic design space exploration through formula manipulation and mapping can find the best trade-off



Future work

More DSP transform

Formula

Representation
Manipulation
Mapping

More design decisions

DFT
DCT
DST
DWT
…

Pipelining
Systolic array
Distributed Arithmetic
Fix-point vs. Floating-point
…

Thank you !

Contact: Fang Fang

Email: ffang@cmu.edu

URL: www.ece.cmu.edu/~ffang