Efficient Hardware Implementations for 2-dimensional Discrete Cosine Transform

Hyesook Lim

Department of Electrical and Computer Engineering

The University of Texas at Austin

Austin, TX 78712

Sep. 18, 1996

Problem Statement

- **Discrete Cosine Transform**
  - The most widely used transform in image processing
  - High computational complexity
  - High speed of processing is required in some applications.
- **Strong need to develop VLSI implementations for DCT**

- **"What is a systolic array?"**

  A computing network possessing

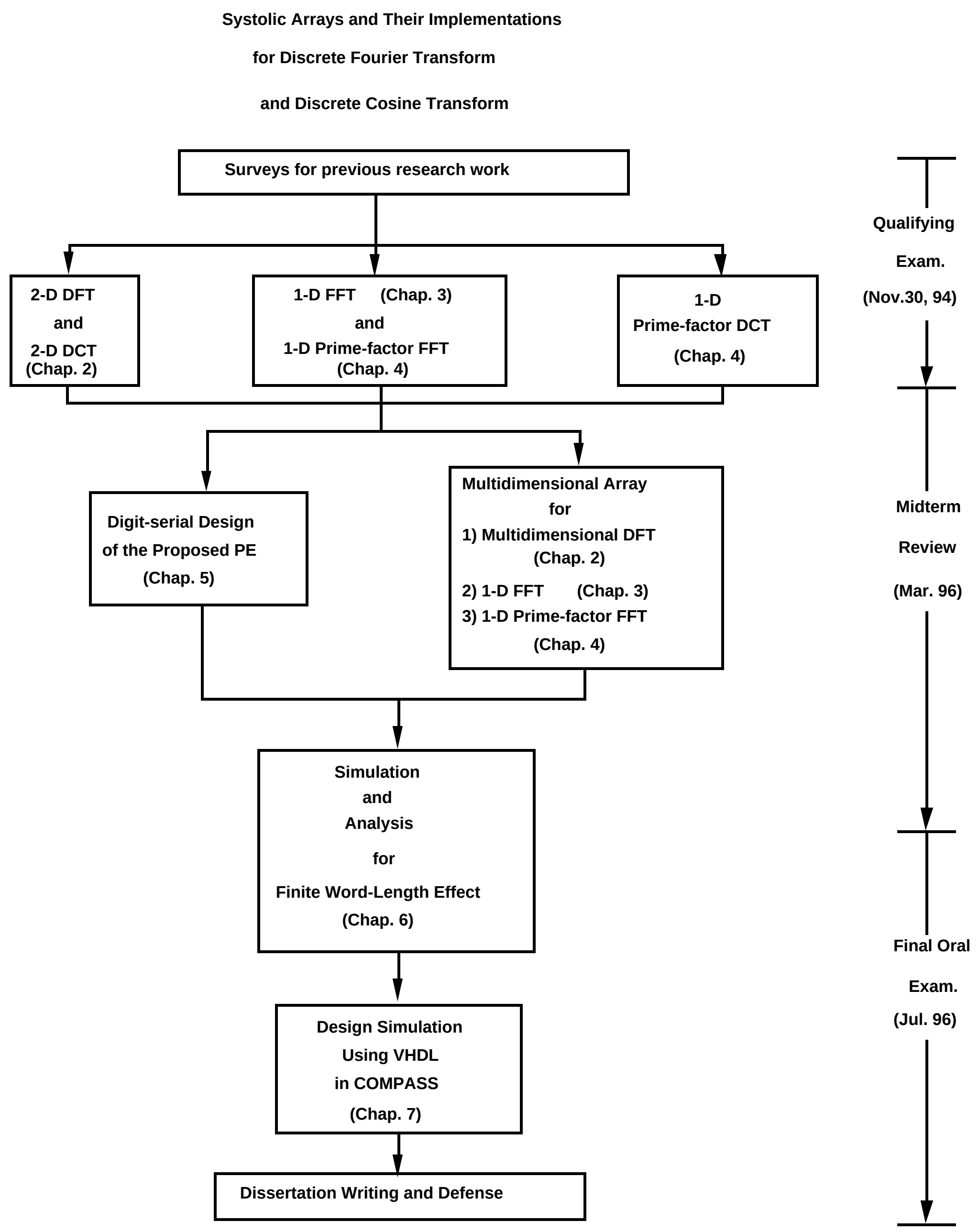
  synchrony, modularity and regularity, locality, and pipelinability.

- **"Why systolic array?"**

  It has excellent features for VLSI implementations.

  (Simple and regular design, local communication, concurrency).

• Research flow


Systolic Arrays and Their Implementations
for Discrete Fourier Transform
and Discrete Cosine Transform
Surveys for previous research work
2-D DFT and 2-D DCT (Chap. 2)
1-D FFT (Chap. 3) and 1-D Prime-factor FFT (Chap. 4)
1-D Prime-factor DCT (Chap. 4)
Digit-serial Design of the Proposed PE (Chap. 5)
Multidimensional Array for 1) Multidimensional DFT (Chap. 2) 2) 1-D FFT (Chap. 3) 3) 1-D Prime-factor FFT (Chap. 4)
Simulation and Analysis for Finite Word-Length Effect (Chap. 6)
Design Simulation Using VHDL in COMPASS (Chap. 7)
Dissertation Writing and Defense
Qualifying Exam. (Nov.30, 94)
Midterm Review (Mar. 96)
Final Oral Exam. (Jul. 96)

Outline

▣ **Introduction**

1) Definition

2) Fast computation algorithms for DCT

3) Applications for 2-D DCT

- A systolic array for 2-D DCT

1) Semi-systolic arrays for matrix multiplication

2) The proposed systolic array for 2-D DCT

3) A 2-D DCT/IDCT processor design

- Conclusions

Definition

- **Discrete Cosine Transform (DCT):**

$$y(k) = \sqrt{\frac{2}{N}}u(k)\sum_{n=0}^{N-1} x(n)\cos\frac{(2n+1)k\pi}{2N} \quad \text{for } k = 0, 1, ..., N\text{-}1$$

- **Inverse Discrete Cosine Transform (IDCT):**

$$x(n) = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} u(k)y(k)\cos\frac{(2n+1)k\pi}{2N} \quad \text{for } n = 0, 1, ..., N\text{-}1$$

where $x(n)$: time-domain data sequence,

$y(k)$: its transform-domain sequence,

and $u(k) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k=0 \\ 1, & \text{otherwise.} \end{cases}$

Definition in Matrix Form

	DCT	IDCT	Number of Multiplications
1-D Computation	$(y_N) = [C_N](x_N)$	$(x_N) = [C_N]^T (y_N)$	N^2
2-D Computation	$[Z_N] = [C_N][X_N][C_N]^T$	$[X_N] = [C_N]^T [Z_N][C_N]$	$2N^3$

where $(x_N)/(y_N)$: 1-d input/output column vector

$[X_N]/[Z_N]$: 2-d input/output matrix

$[C_N]$: $N \times N$ DCT coefficient matrix

$$[C_N]_{kn} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for} \quad k=0 \\ \cos\frac{(2n+1)k\pi}{2N}, & \text{otherwise.} \end{cases}$$

$$[C_4] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \\ \cos\frac{\pi}{8} & \cos\frac{3\pi}{8} & \cos\frac{5\pi}{8} & \mathrm{c} \\ \cos\frac{2\pi}{8} & \cos\frac{6\pi}{8} & \cos\frac{10\pi}{8} & \mathrm{c} \\ \cos\frac{3\pi}{8} & \cos\frac{9\pi}{8} & \cos\frac{15\pi}{8} & \mathrm{c} \end{bmatrix}$$

Fast Computation Algorithms

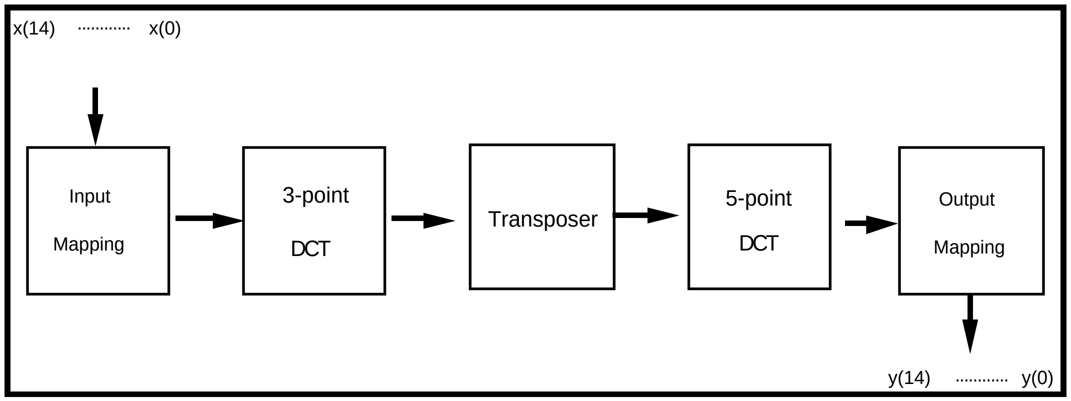
- Principle:

  break one large-point computation into several small-point computations

- Advantage: reduced number of multiplications

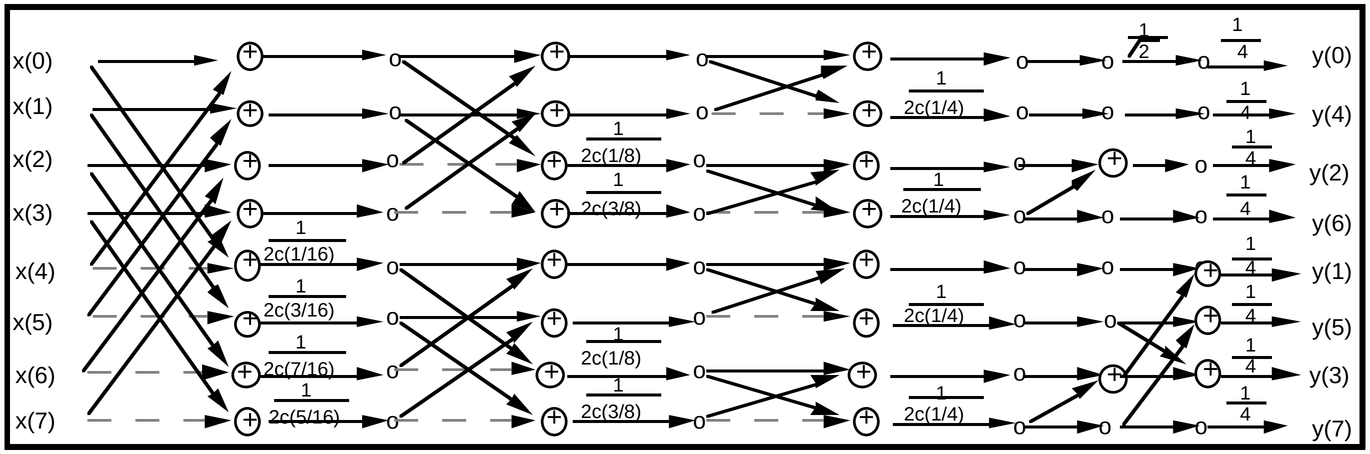
  Disadvantage: complex structure

Prime-factor DCT algorithms: Lee (1994), Yang (1985), Chakrabarti (1990)



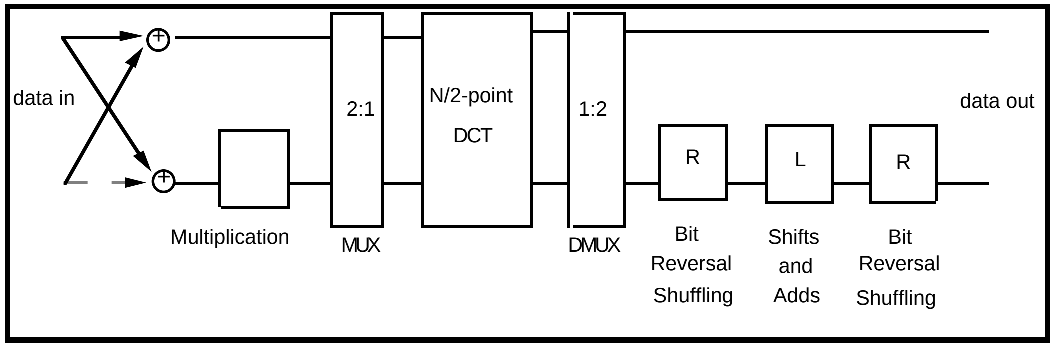
Block diagram for 15-point DCT using prime-factor DCT algorithm

◘ Lee's fast DCT algorithm: Parkhurst (1988), Wu (1993)



Signal flow graph for computing 8-point DCT using Lee's FDCT algorithm

◘ Hou's fast DCT algorithm: Fettweis (1993)



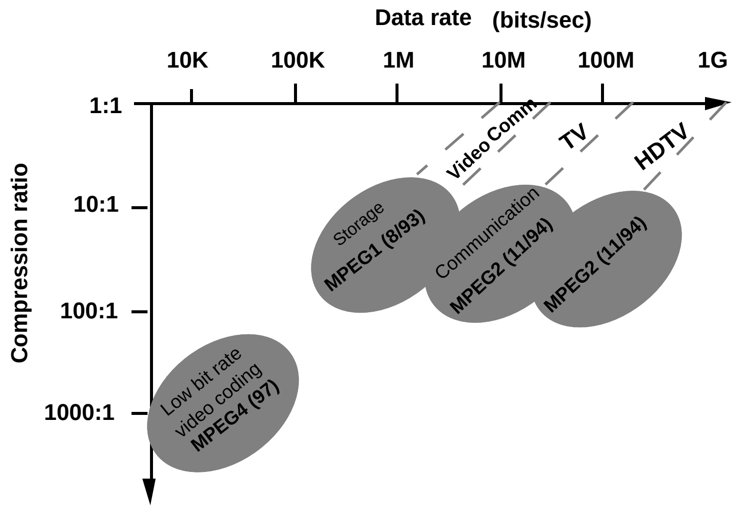
Computation flow for computing DCT using Hou's FDCT algorithm

2-dimensional DCT/IDCT Applications

- Best transform method to compress a digital picture
  - Good energy compactness

- Standard data compression technique for video compression standards
  (H.261, H.263, MPEG-1 and MPEG-2)

- Multimedia World



- MPEG encoding structure

  - Motion compensation (reduce the temporal redundancy)

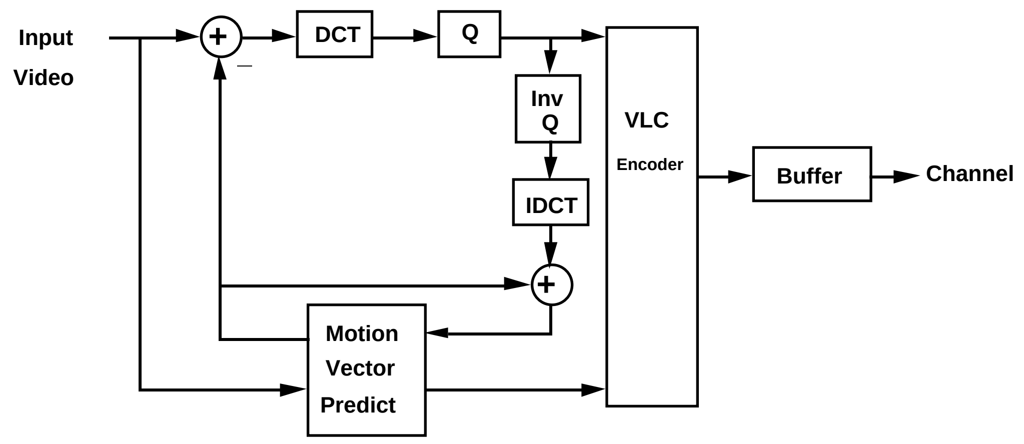
  - The difference signal (prediction error) is compressed using DCT.

    (remove spatial correlation)

  - Quantization

  - Combine motion vectors with DCT information

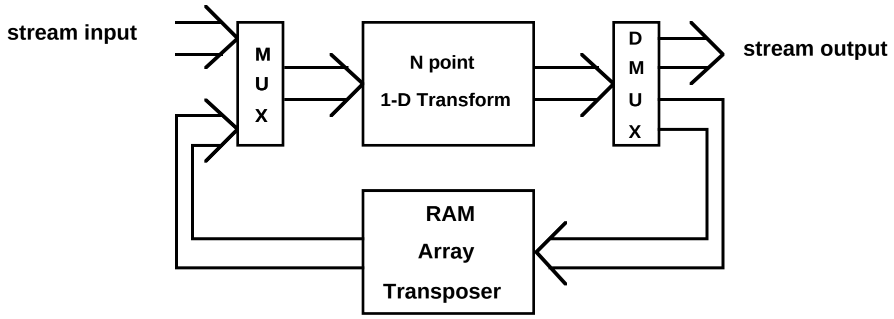
  - Coded using variable length codes (VLC)



- Focusing on the development of VLSI chip implementation of the 2-D DCT

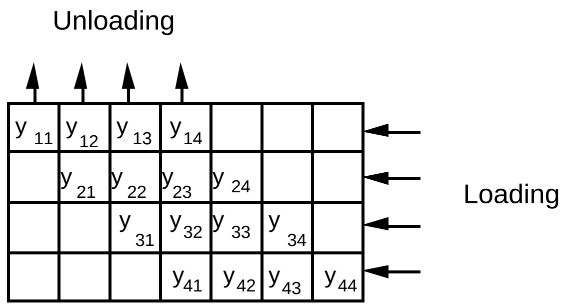
Earlier Implementations for 2-dimensional DCT

◘ Row-column approach for 2-D DCT



◘ RAM Array Transposer

- Must wait for N cycles to unload data
- Limits the pipelined processing
- Area for global connection
- Time for loading and unloading



Outline

Definition in Matrix Form

	DCT	IDCT	Number of Multiplications
1-D Computation	$(y_N) = [C_N](x_N)$	$(x_N) = [C_N]^T (y_N)$	N^2
2-D Computation	$[Z_N] = [C_N][X_N][C_N]^T$	$[X_N] = [C_N]^T [Z_N][C_N]$	$2N^3$

where (x_N)/(y_N): 1-d input/output column vector

$[X_N]$/$[Z_N]$: 2-d input/output matrix

$[C_N]$: $N \times N$ DCT coefficient matrix

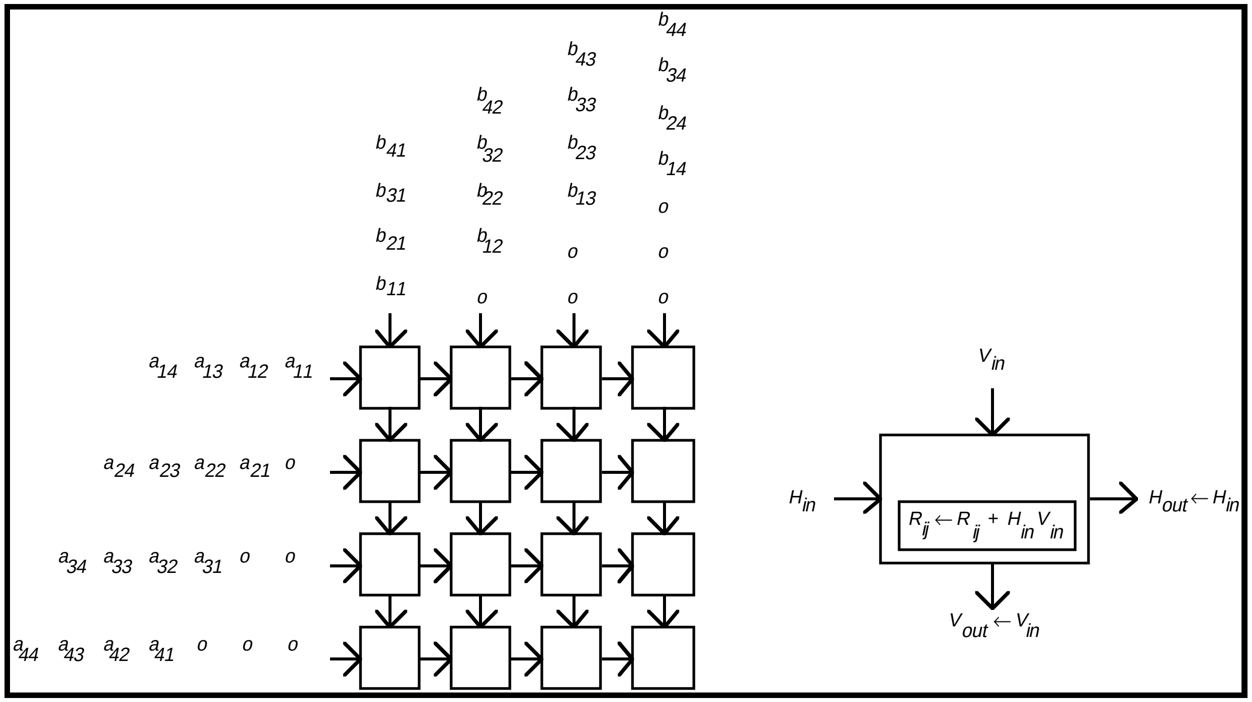
$$[C_N]_{kn} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for} \quad k=0 \\ \cos\frac{(2n+1)k\pi}{2N}, & \text{otherwise.} \end{cases}$$

$$[C_4] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \\ \cos\frac{\pi}{8} & \cos\frac{3\pi}{8} & \cos\frac{5\pi}{8} & \mathrm{c} \\ \cos\frac{2\pi}{8} & \cos\frac{6\pi}{8} & \cos\frac{10\pi}{8} & \mathrm{cc} \\ \cos\frac{3\pi}{8} & \cos\frac{9\pi}{8} & \cos\frac{15\pi}{8} & \mathrm{cc} \end{bmatrix}$$

◘ Semi-systolic array

- Output data is not produced in the boundary cells of the array (Type 1).
- Input data is preloaded into every cell of the array (Type 2).

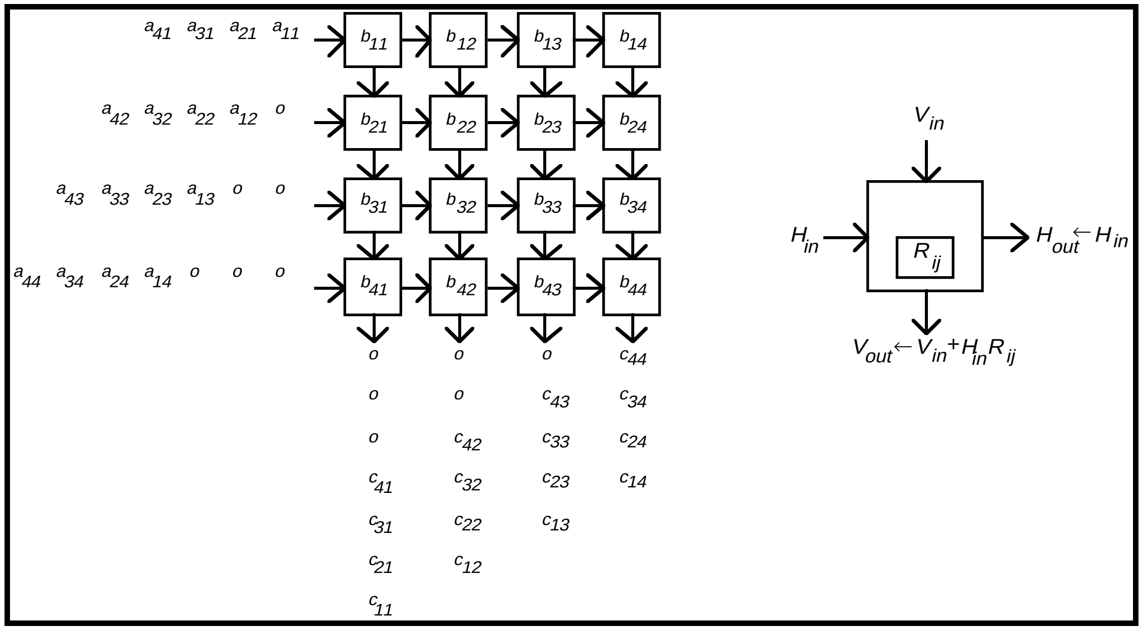
◘ Type 1 semi-systolic array (C = AB, 4 by 4 matrix)



b_{41} b_{31} b_{21} b_{11}
b_{42} b_{32} b_{22} b_{12} o
b_{43} b_{33} b_{23} b_{13} o o
b_{44} b_{34} b_{24} b_{14} o o o
a_{14} a_{13} a_{12} a_{11}
a_{24} a_{23} a_{22} a_{21} o
a_{34} a_{33} a_{32} a_{31} o o
a_{44} a_{43} a_{42} a_{41} o o o
V_{in}
H_{in}
$R_{ij} \leftarrow R_{ij} + H_{in}V_{in}$
$H_{out} \leftarrow H_{in}$
$V_{out} \leftarrow V_{in}$

$[a_{ik}]$ moves by *x*-axis, $[b_{kj}]$ moves by *y*-axis, and $[c_{ij}]$ is accumulated.

◘ Type 2 semi-systolic array (C = AB, 4 by 4 matrix)



$[a_{ik}]$ moves by *x*-axis, $[b_{kj}]$ stays (preloading), and $[c_{ij}]$ moves by *y*-axis.

Outline

- Introduction

1) Definition

2) Fast computation algorithms for DCT

3) Applications for 2-D DCT

- A systolic array for 2-D DCT

1) Semi-systolic arrays for matrix multiplication

2) The proposed systolic array for 2-D DCT

3) A 2-D DCT/IDCT processor design

- Conclusions

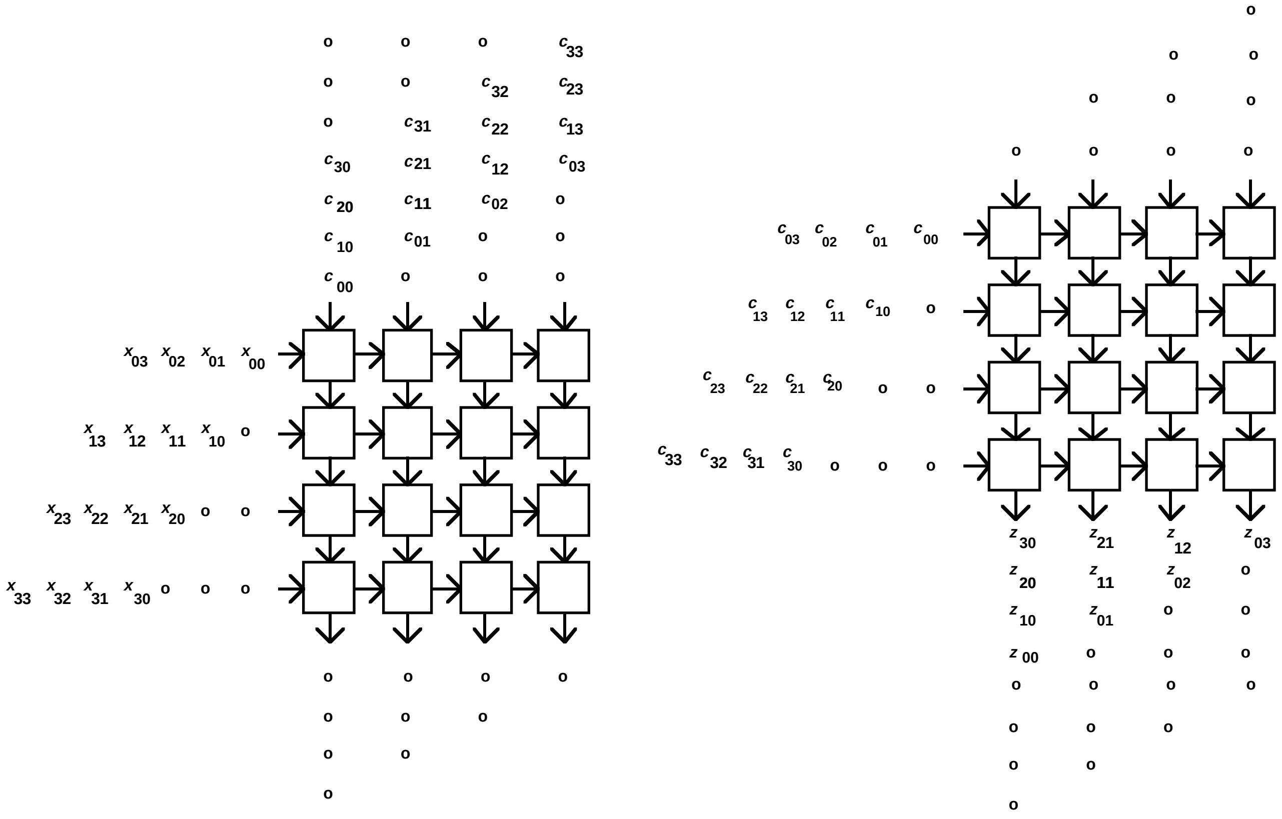
Our Approach for 2-D DCT in Systolic Array

▣ **Basic Principle**

- Combine two types of semi-systolic arrays for matrix multiplication into one array
- Input and output move along axis and intermediate result does not move.
- The resulting array becomes a true systolic array.

2-dimensional DCT by 2-dimensional Systolic Array

- **2-D DCT:** $[Z_N]=[C_N][X_N][C_N]^T$ ($[C_N]$: coefficient matrix, $[X_N]$: data matrix)

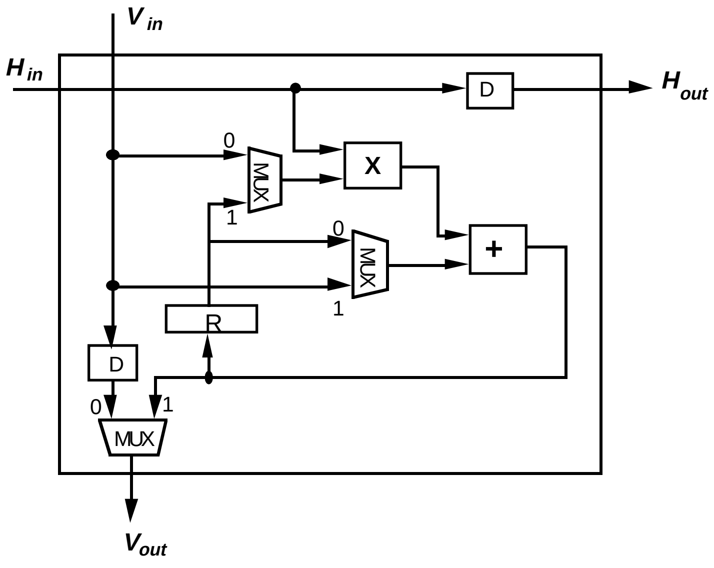


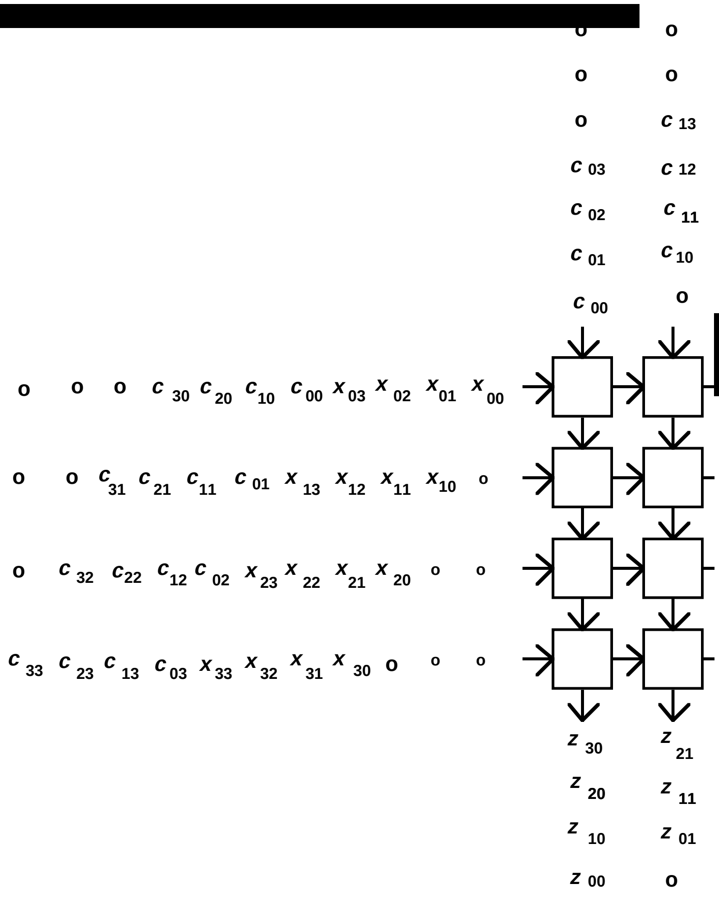
Semi-systolic arrays for 1-dimensional DCTs

(a) $\boxed{[Y_N]=[X_N][C_N]^T}$ in Type-1 semi array (b) $\boxed{[Z_N]=[C_N][Y_N]}$ in Type-2 semi array

Proposed Systolic Array and PE Structure for 2-D DCT

2-D DCT:





o o o c13 c12 c11 c10 o
o o o c03 c02 c01 c00
o o o c30 c20 c10 c00 x03 x02 x01 x00
o o c31 c21 c11 c01 x13 x12 x11 x10 o
o c32 c22 c12 c02 x23 x22 x21 x20 o o
c33 c23 c13 c03 x33 x32 x31 x30 o o o
z30 z20 z10 z00
z21 z11 z01 o

$$[Z_N] = [C_N][X_N][$$

Complexity Estimation for VLSI Implementation

- **Throughput (time):** number of cycles to compute each point of the transform

- **Area:** depends on the number of multipliers for 2-dimensional array

  (Each multiplier has $O(\log N)$ time and $O(\log N)$ area.)

- **Measure of Complexity : Area⊠Time²** (Lower bound : $O(N^4 \log^2 N)$)

- **Our proposed systolic array**

  - Area×Time² = $O(N^4 \log^3 N)$ for 2-dimensional transform
  - within a $O(\log N)$ factor of lower bound

Outline

- Introduction

1) Definition

2) Fast computation algorithms for DCT

3) Applications for 2-D DCT

- A systolic array for 2-D DCT

1) Semi-systolic arrays for matrix multiplication

2) The proposed systolic array for 2-D DCT

3) A 2-D DCT/IDCT processor design

- Conclusions

Earlier 2-D DCT/IDCT Processors

- Madisetti and Wilson (1995), Jang, et. al (1994), Wu and Chiou (1993) Sheu, et. al (1993), Jain, et. al (1993, 1992)

- Fast DCT (or IDCT) algorithms (Lee, Chen) are used.
- Fast algorithms require complex control unit and data reordering unit. (It affects the design cycle since design and testing become complex.)

- These processors require transpose memory for 2-D computation.
  - limits the pipelined processing
  - requires time for loading and unloading
  - requires area for global connection

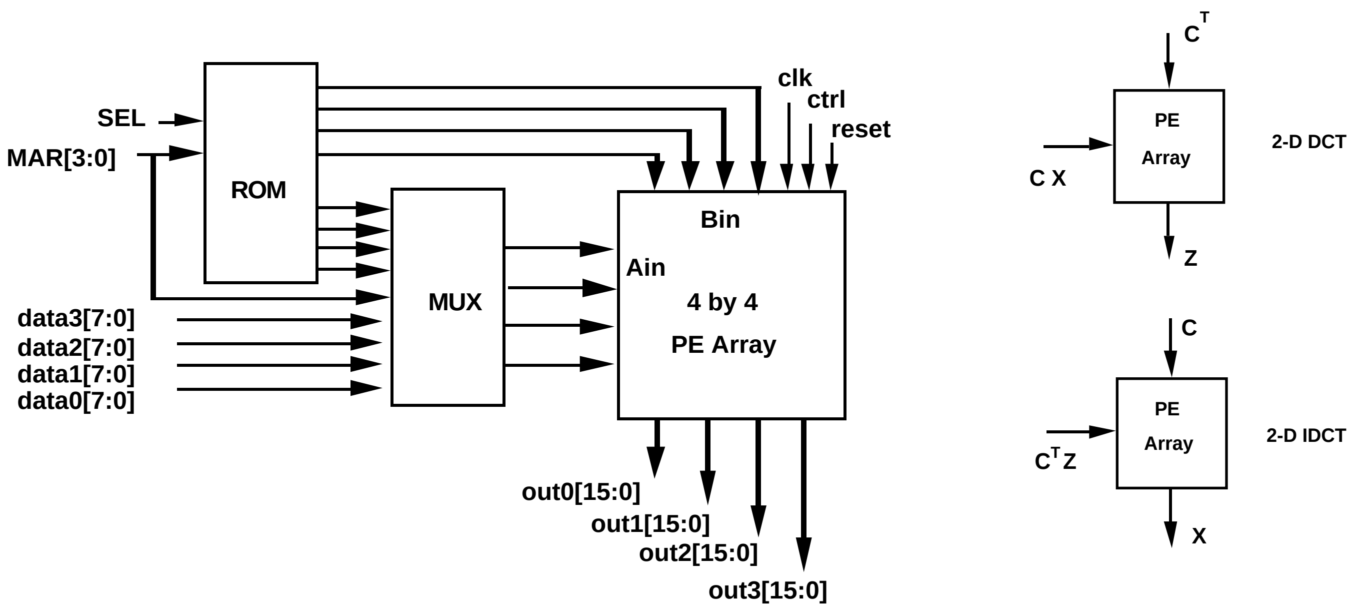
▣ **The proposed 2-D DCT/IDCT processor**

- does not require transpose memory.
- modular and regular structure and local interconnection
- design and testing is much simpler

▣ **We modeled the processor and evaluated the performance in behavior level using VHDL in COMPASS.**

▣ **We synthesized each functional block and PE .**

Architecture of Our 2-D DCT/IDCT Processor



- ROM block: provides the coefficient inputs, C^t or C
- MUX block: switches data inputs and coefficient inputs
- SEL signal: selects DCT computation or IDCT computation
- Processing is unified for 2-D DCT or 2-D IDCT computation.

Simulation for Finite Word-length on $\boxed{8\times 8}$ IDCT

- **IEEE Specifications for Implementation of $\boxed{8\times 8}$ IDCT (for $-256 \leq$ input < 256)**

  1) For any pixel location, the peak error (PPE) ≤ 1.

  2) For any pixel location, the mean square error (PMSE) ≤ 0.06.

  3) Overall, the mean square error (OMSE) ≤ 0.02.

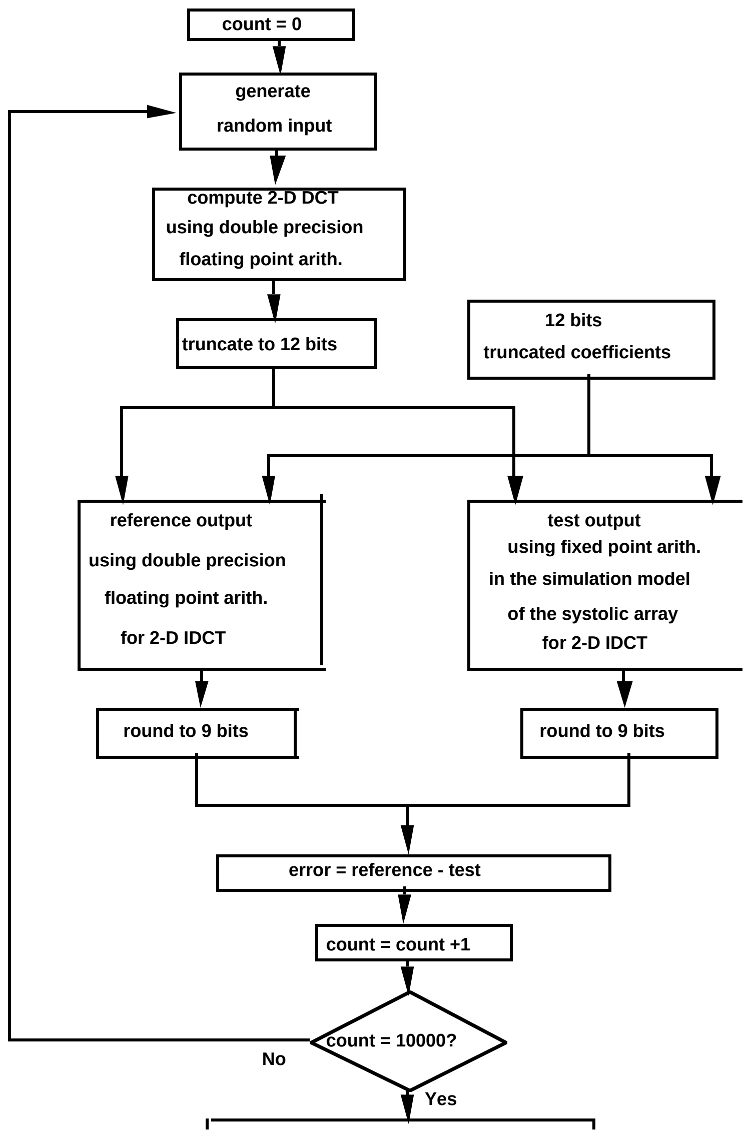
  4) For any pixel location, the mean error (PME) ≤ 0.015

  5) Overall, the mean error (OME) ≤ 0.0015.

  6) For all zero input, the proposed IDCT shall generate all-zero output.

- **Find out the minimum word length of our systolic array which satisfies these requirements**

◘ Simulation flow for finding the minimum word-length

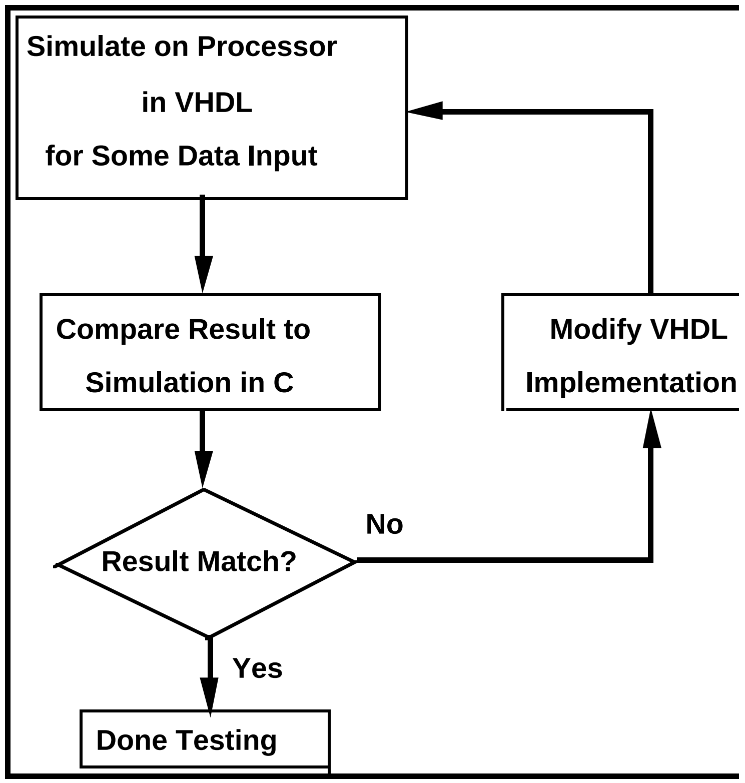

count = 0
generate random input
compute 2-D DCT using double precision floating point arith.
truncate to 12 bits
12 bits truncated coefficients
reference output using double precision floating point arith. for 2-D IDCT
test output using fixed point arith. in the simulation model of the systolic array for 2-D IDCT
round to 9 bits
round to 9 bits
error = reference - test
count = count +1
count = 10000?
No
Yes

◘ The minimum word-length to satisfy the IEEE specifications

Word-length	15	16	17	18	19	20
PPE≤1	1.0	1.0	1.0	1.0	1.0	1.0
PMSE≤0.06	0.0120	0.0068	0.0044	0.0037	0.0026	0.0008
OMSE≤0.02	0.0089	0.0046	0.0025	0.0014	0.0007	0.0003
PME≤0.015	0.0108	0.0064	0.0044	0.0037	0.0026	0.0008
OME≤0.0015	0.0078	0.0041	0.0022	0.0012	0.0007	0.0003

Hardware Simulation

- Simulation flow in VHDL



- Test cases were written.
- 2-D transform result becomes available after 2*N* clock cycles.
- Clock cycle time is 61.25 ns (16.3 MHz) for .4μ technology.

Synthesized Result

▣ **Number of gates equivalent for PEs and each block**

	4×4 block transform	8×8 block transform
peb.vhd	3113	4071
pe1.vhd	3148	4125
MUX	152	357
ROM	174	904

Summary of 2-D DCT/IDCT Processor

▣ A signal processor which performs 2-D DCT or 2-D IDCT computation.

- The processor provides a common data path shared by both DCT and IDCT.
- The processor is modeled and the performance is evaluated in behavior level using VHDL.
- Each block and PEs are synthesized.

Conclusions

- **New systolic approach for computing 2-D discrete cosine transform**
- **A processor including the systolic array has been designed and the performance has been evaluated.**

- **Excellent features of our proposed systolic arrays:**

1) Simple and regular processing elements (multiply-add type).

2) Simple and regular connections between the PEs

3) no networks for the transposition of intermediate spectrum

(the area complexity for matrix transposer, O(N^4), dominates the area.)

4) The *area*time*^{2} complexity is within a O(log$N$) factor of the lower bound.

5) Easily scaled by extending the size of PE array

6) Can be used for any multidimensional orthogonal transforms

(DFT, DST, DHT)

Publications

1. H. S. Lim and E. E. Swartzlander, Jr., "A Systolic Array for 2-D DFT and 2-D DCT," *International Conference on Application-Specific Array Processors*, Aug. 1994.

2. H. S. Lim and E. E. Swartzlander, Jr., "An Efficient Systolic Array for DCT Based on Prime-Factor Decomposition," *International Conference on Computer Design*, Oct. 1995.

3. H. S. Lim and E. E. Swartzlander, Jr., "Efficient Systolic Arrays for FFT Algorithms," *Twenty-Ninth Annual Asilomar Conference on Signals, Systems, and Computers*, Oct. 1995.

4. H. S. Lim, "Multidimensional Systolic Arrays for Multidimensional DFTs," *UT Student Research Conference*, *Graduate Engineering Council in the University of Texas at Austin*, Nov. 1995.

5. H. S. Lim and E. E. Swartzlander, Jr., "Multidimensional Systolic Arrays for Multidimensional DFTs," *IEEE International Conference on Acoustics, Speech & Signal Processing,* May 1996.

6. H. S. Lim, C. Yim, and E. E. Swartzlander, Jr., "Finite Word-Length Effects of a 2-D DCT Systolic Array," *International Conference on Application-Specific Array Processors*, Aug. 1996

7. H. S. Lim, “Multidimensional Systolic Arrays for Computing Discrete Fourier Transform and Discrete Cosine Transform,” Chapter 6 in *Application Specific Processors* (E. E. Swartzlander, Jr., ed.), Kluwer Academic Publishers, Boston, pp. 161-195, 1996 (In press).