Systolic Arrays

Presentation at UCF

by

Jason HandUber

February 12, 2003

Presentation Overview

- Introduction
  - Abstract Intro to Systolic Arrays
  - Importance of Systolic Arrays
  - Necessary Review – VLSI, definitions, matrix multiplication
- Systolic Arrays
  - Hardware & Network Interconnections
  - Matrix-Vector Multiplication
  - Beyond M.V. Multiplication
  - Applications and Extensions of covered topics
- Summary

Introduction - Scenario

- Your boss approaches you at work and notifies you that the company has a chance at landing an obscenely lucrative government contract.
- He asks you to put together a proposal and indicates that for you to keep making the $130,000 / year that you make you should be able to secure the contract.
- Lastly, he informs you that the government contract is concerned with one of the topics on the following slide, that you have essentially limitless funding, and that the contract specifies that the final run-time of the algorithm must be linear.

Signal Processing Algorithms

Introduction – “Why?”

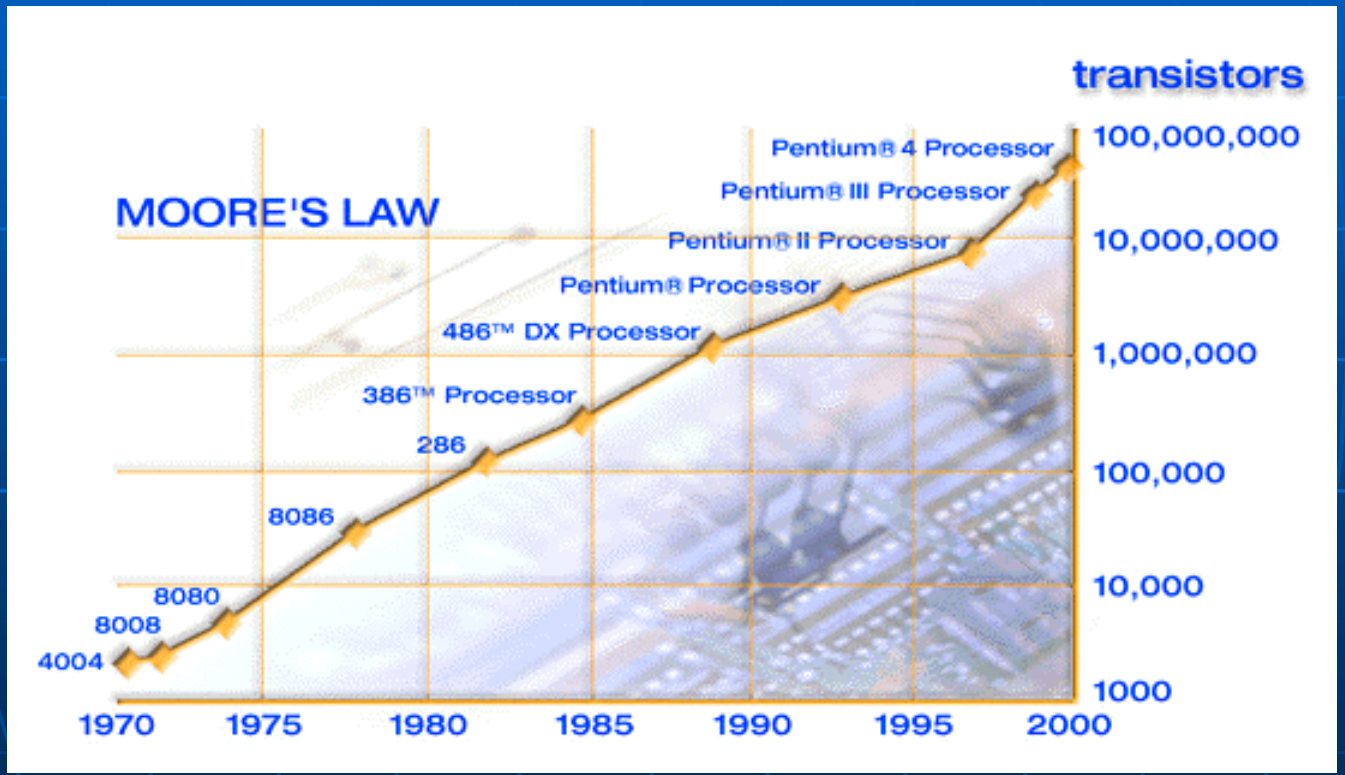
- What is the main commercial point of Computer Architecture?

  Essentially → Moore’s Law

- To that end, what are two main points computer architects have been focusing on in recent years?

  Pipelining & Parallelism

Moore’s Law


transistors
MOORE'S LAW
Pentium® 4 Processor
Pentium® III Processor
Pentium® II Processor
Pentium® Processor
486™ DX Processor
386™ Processor
286
8086
8080
8008
4004
100,000,000
10,000,000
1,000,000
100,000
10,000
1000
1970
1975
1980
1985
1990
1995
2000

Introduction – Pipelining & Parallelism

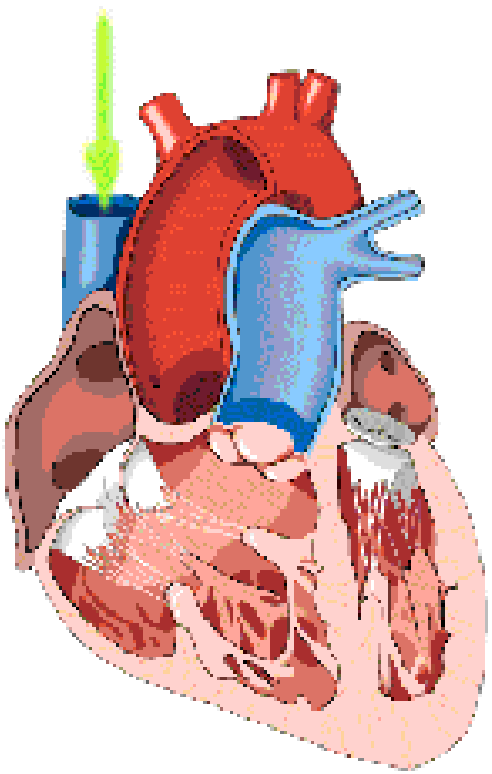
Processor Pipelining

Ideally at least one new instruction completes every time cycle.

Parallelism

Multiple jobs are allowed to perform simultaneously

Introduction – “Systolic”



- Right Atrium
- Tricuspid Valve
- Right Ventricle
- Pulmonic Valve
- Pulmonary Arteries
- Pulmonic Veins
- Left Atrium
- Mitral Valve
- Left Ventricle
- Aortic Valve
- Aorta

Introduction – Systolic Definition

- “Imagine n simple processors arranged in a row or an array and connected in such a manner that each processor may exchange information with only its neighbors to the right and left. The processors at either end of the row are used for input and output. Such a machine constitutes the simplest example of a systolic array.”[1]

Introduction – Systolic Definition (2)

- “Systolic Arrays are regular arrays of simple finite state machines, where each finite state machine in the array is identical...A systolic algorithm relies on data from different directions arriving at cells in the array at regular intervals and being combined.” [2]

Systolic Arrays

- “By pipelining, processing may proceed concurrently with input and output, and consequently overall execution time is minimized. Pipelining plus multiprocessing at each stage of a pipeline should lead to the **best-possible** performance.”[3]

Introduction – Review - VLSI

- VLSI – Very Large Scale Integration
- VLSI is low-cost, high-density, high-speed.
- "VLSI technology is especially suitable for designs which are *regular, repeatable,* and with high *localized* communications."
- "A systolic array is a design style for VLSI." [3]

Introduction – Review - Matrix Multiplication

- Consider multiplying a 3x2 X 2x1 matrix:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$y_1 = (a_{11} * x_1) + (a_{12} * x_2)$$
$$y_2 = (a_{21} * x_1) + (a_{22} * x_2)$$
$$y_3 = (a_{31} * x_1) + (a_{32} * x_2)$$

Introduction – Review

Systolic Cell – basic workhorse (processor) of a systolic array.

- Few Fast Registers
- ALU
- Simple I/O

- Multiple CPUs on one machine
- Parallel Execution

Systolic Advantages
How they work

A systolic array has multiple cells networked together to form an array.

Speed – register to register transfer of data. Data is not destroyed until it has been completely used.

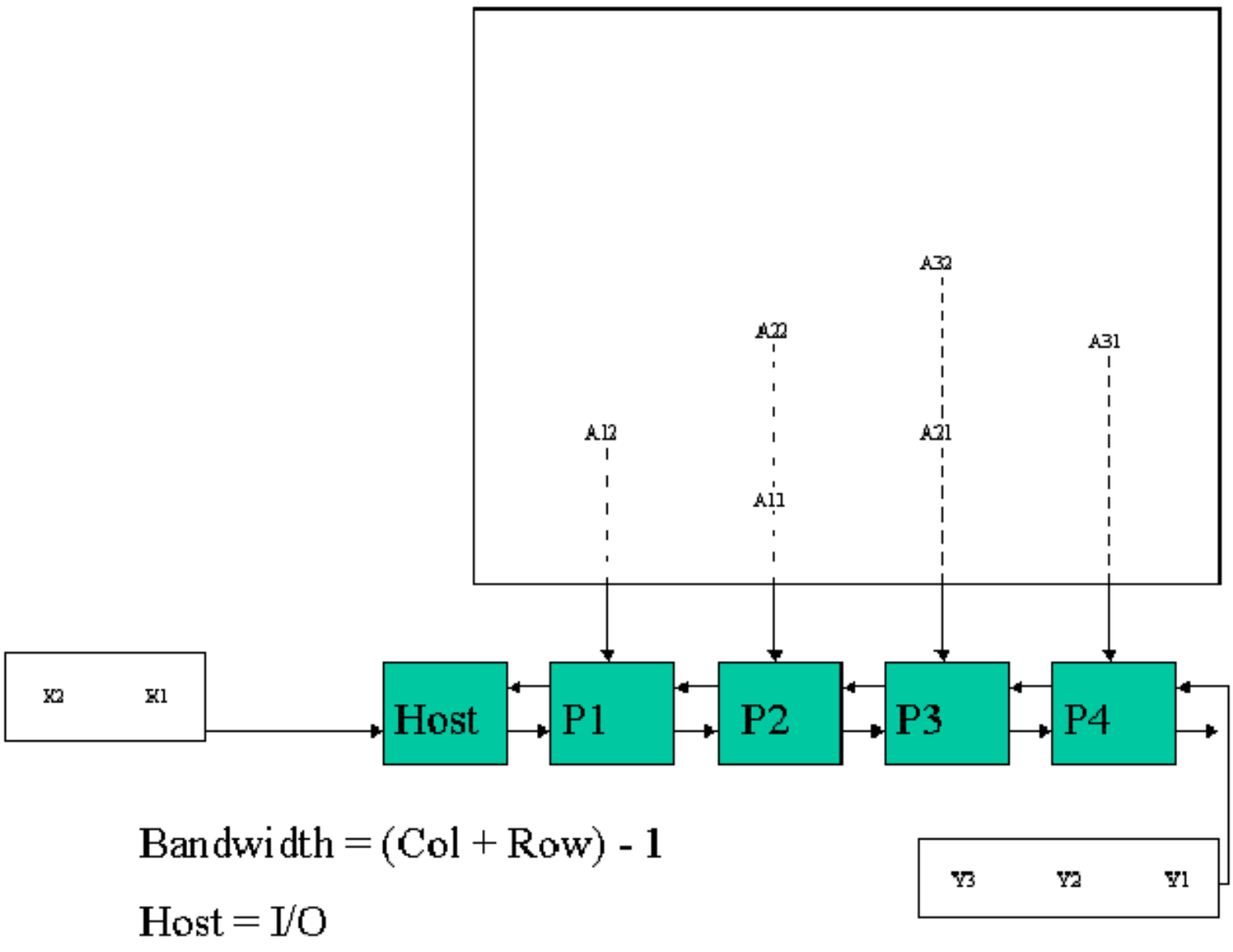
Synchronization – All cells run off of a central clock.

Host Data Entry – All cells (including boundary cells) are I/O capable.

Example of Linear Systolic Array

- Breakdown of data into 3 parts
  - Input matrix 1
  - Input Matrix 2
  - Output matrix
- What are the different parts to an array?
- What is bandwidth?

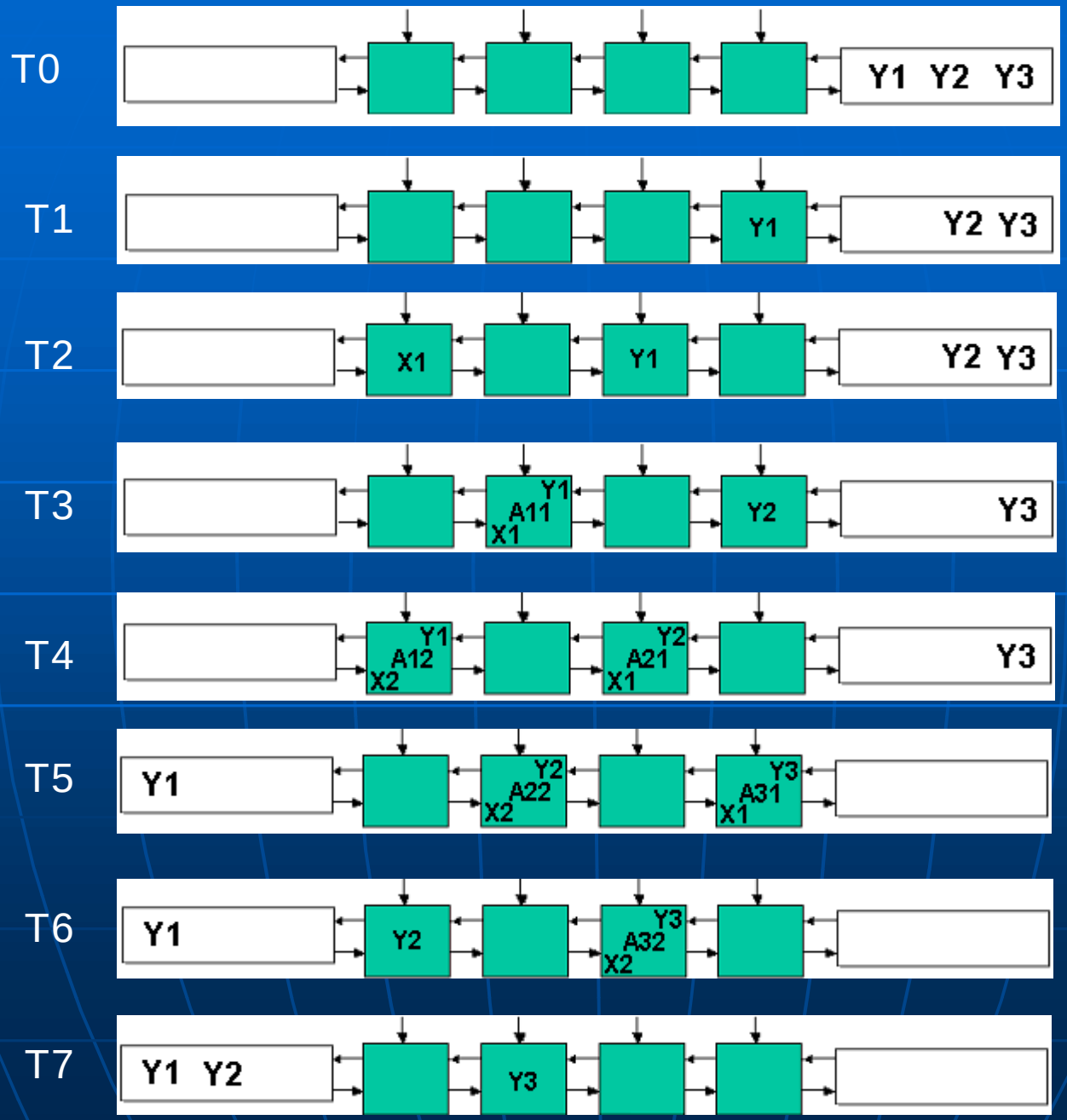
Systolic Arrays



Bandwidth = (Col + Row) - 1

Host = I/O

Y values goes left, X values go right, A values fan in



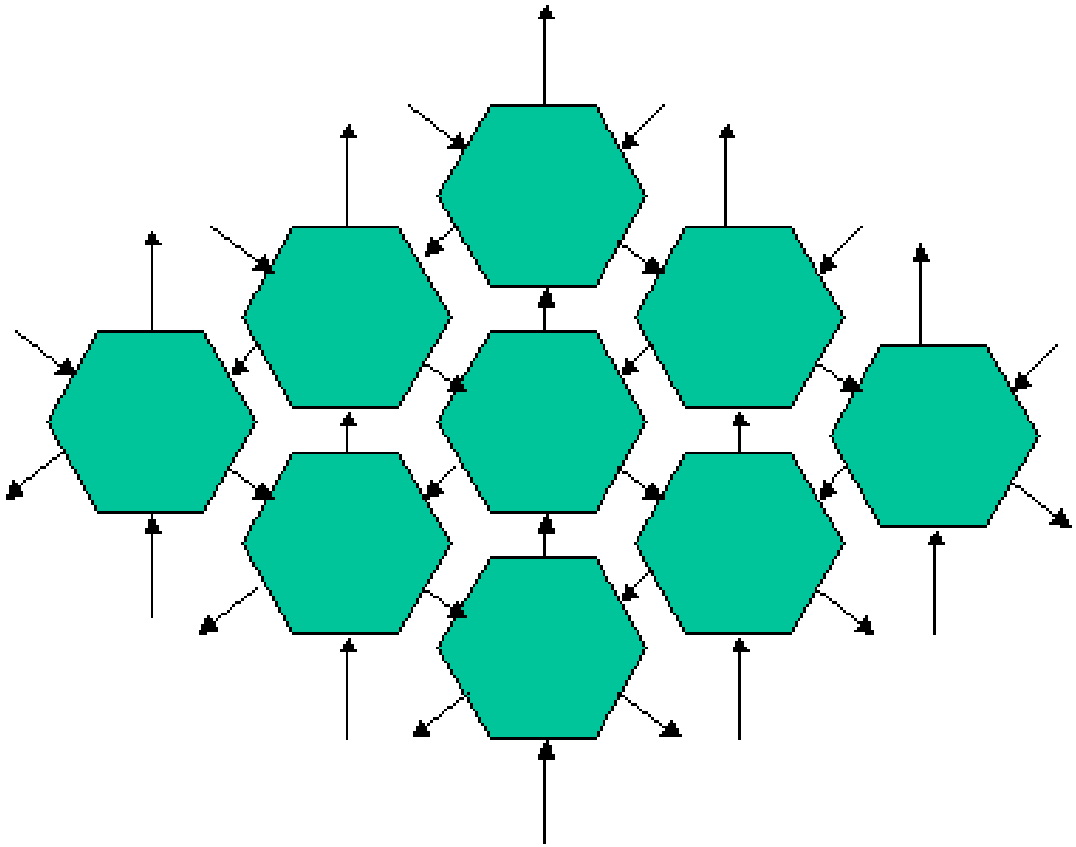
Systolic Arrays
Linear Systolic Arrays

- **PIPELINING**
  - Multiple CPUs Pipelined Together
  - Basic Architecture
  - Speed Up
    - O(wn) → Exponential
    - 2n + w → Linear!!

Systolic Arrays

- The concepts used in Matrix-Vector multiplication can be easily extended to compute more complex functions.
  - Some of these functions were introduced in the introduction during the flash presentation and include the multiplication of multiple matrices and n-dimensional applications.

Systolic Arrays



HEXAGONAL SYSTOLIC ARRAY

THINK: MULTIPLE MATRIX MULTIPLICATION

3N + MIN (W1, W2)

Pipelining Vs. Systolic Array

- Input data is not consumed
- Input data streams can flow in different directions
- Modules may be organized in a two dimensional (or higher) configuration
- Configurable – Different array configurations available for different processing purposes.

Systolic Advantages and How they work.

- Systolic Array – A network of systolic Cells.
  - Systolic Cell – An independent operating environment with processor, registers and ALU.
- Scalable – Easily extend the architecture to many more processors.
- Capable of supporting SIMD organizations for vector operations and MIMD for non-homogeneous parallelism.
- Allow extremely high throughput w/multi-dimensional arrays.

Systolic Disadvantages

- Complicated – Both in Hardware and Software.
  - In fact entire volumes exist outlining systolic array verification.
- Expensive in comparison to uni-processor systems, although much faster.

Presentation Summary

- Systolic Arrays offer a way to take certain exponential algorithms and use hardware to make them linear.
- They are expensive and complex but yield enormous throughput.
- Any Questions?

References

[1] Bayoumi, Magdy. Ling, Nam. Specification and Verification of Systolic Arrays. World Scientific Publishing Co. Pte. Ltd. Singapore. 1999.

[2] Brown, Andrew. VLSI Circuits and Systems in Silicon. McGraw-Hill Book Company. London. 1991.

[3] Dewdney, A.K. The (New) Turing Omnibus. Henry Holt and Company. New York. 1993.

[4] Fisher, J. “Very long instruction word architectures and the ELI-512”. In proc. 10th International Symposium on Computer Architecture. June 1983, pp. 140-150.

[5] Hennessey, J. Patterson, D. Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers. San Francisco.

[6] Kung, H.T. Leiserson, C.E. “Algorithms for VLSI processor arrays.” C. Mead and L. Conway, editors, Introduction to VLSI Systems, Ch. 8.3. Addison-Wesley. 1980.

[7] McCanny, John. McWhirter, John. Swartzlander, Earl. Systolic Arrays Processors. Prentice Hall International Ltd. Hertfordshire. 1989.

[8] Moore, Will. McCabe, Andrew. Urquhart, Roddy. Systolic Arrays. J W Arrowsmith Ltd. Bristol. 1987.

Post-Presentation Thoughts

The comment made in class that it is difficult for the programmer to know when to implement parallelism isn't really a fair statement. Certain newer architectures and compilers determine all code dependencies and make it easy for a CPU scheduler to efficiently divide such tasks with help from the code itself during compile-time. For more information on some of these "newer" architectures look up VLIW (Very Long Instruction Words) to see how smart compilers in conjunction with specific schedulers can package instruction words so that they easily take full advantage of multiple processors with minimal or no delay to compute the division of such tasks.