Digital Design and Computer Organization

Systolic Array Matrix Multiply

Reetinder Sidhu

Department of Computer Science and Engineering

Digital Design and Computer Organization

Systolic Array Matrix Multiply

Reetinder Sidhu

Department of Computer Science and Engineering

Systolic Array Matrix Multiply

Course Outline

- Digital Design
  - Combinational logic design
  - Sequential logic design
- Computer Organization
  - Architecture (microprocessor instruction set)
  - Microarchitecure (microprocessor operation)
    - ★ **Systolic Array Matrix Multiply**

Concepts covered

- Matrix Multiplication
  - Software
  - Hardware
  - Comparison

Systolic Array Matrix Multiply
Software Matrix Multiplication

Systolic Array Matrix Multiply
Software Matrix Multiplication

PES UNIVERSITY ONLINE

Matrix Multiply Algorithm

```
for (i=0; i<m; ++i) {
  for (j=0; j<n; ++j) {
    for (k=0; k<p; ++k) {
      c[i][j]=c[i][j]+a[i][k]*b[k][j];
    }
  }
}
```

Systolic Array Matrix Multiply
Software Matrix Multiplication

Matrix Multiply Algorithm

```
for (i=0; i<m; ++i) {
  for (j=0; j<n; ++j) {
    for (k=0; k<p; ++k) {
      c[i][j]=c[i][j]+a[i][k]*b[k][j];
    }
  }
}
```

Operations in each Iteration

- Computations each iteration:
  - Loop variable increment and comparison
  - Conditional branch
  - Floating point multiply and add
- On a modern microprocessor, reasonable to assume all three above computations performed in parallel
- So assume time required per iteration is time required to perform floating point multiplication and addition

Systolic Array Matrix Multiply

Software Matrix Multiplication Time

Systolic Array Matrix Multiply
Software Matrix Multiplication Time

Matrix Multiply Algorithm ($m = n = p = 64$)

```
for (i=0; i<64; ++i) {
  for (j=0; j<64; ++j) {
    for (k=0; k<64; ++k) {
      c[i][j]=c[i][j]+a[i][k]*b[k][j];
    }
  }
}
```

Systolic Array Matrix Multiply
Software Matrix Multiplication Time

Matrix Multiply Algorithm ($m = n = p = 64$)

```
for (i=0; i<64; ++i) {
  for (j=0; j<64; ++j) {
    for (k=0; k<64; ++k) {
      c[i][j]=c[i][j]+a[i][k]*b[k][j];
    }
  }
}
```

Performance Estimate

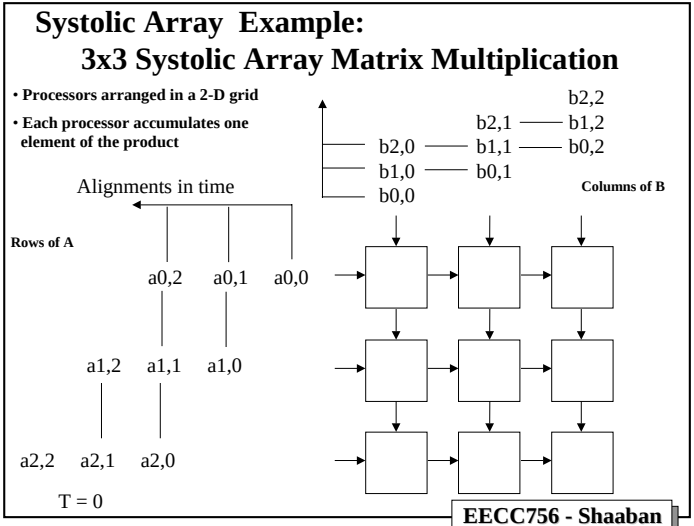
- Assume $m = n = p = 64$
- Assume microprocessor clock speed is 4GHz
  - So clock period is 0.25ns
- Also assume one floating point operation done every clock cycle
- So each loop iteration takes 0.5ns
- Number of loop iterations: $64^3 = 262144$
- Time for 64×64 matrix multiply $= 0.5 \times 262144 = 131072ns$
- Time required for ten 64×64 matrix multiplies on a ten core microprocessor $= 131072ns$

Systolic Array Matrix Multiply

Hardware Implementation of Matrix Multiplication

- To multiply $n \times n$ matrices, use a 2D array of PEs (Processing Elements)
- Each PE can perform a multiply and accumulate in the same clock cycle
- Matrix multiply can be performed in just $3n - 2$ clock cycles

Systolic Array Matrix Multiply

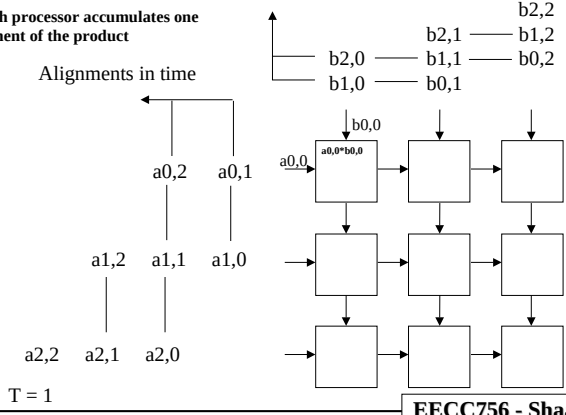


Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply

Systolic Array Example:
3x3 Systolic Array Matrix Multiplication

- **Processors arranged in a 2-D grid**
- **Each processor accumulates one element of the product**

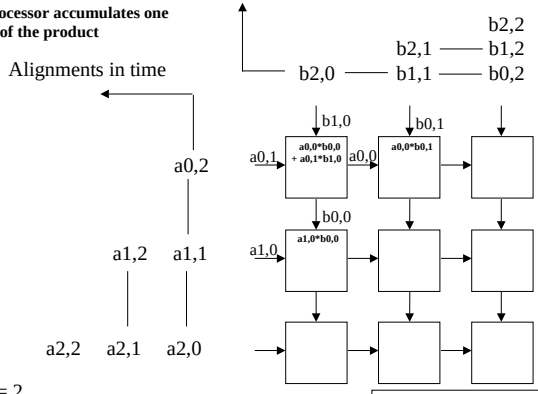


Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply

Systolic Array Example:
3x3 Systolic Array Matrix Multiplication

- **Processors arranged in a 2-D grid**
- **Each processor accumulates one element of the product**

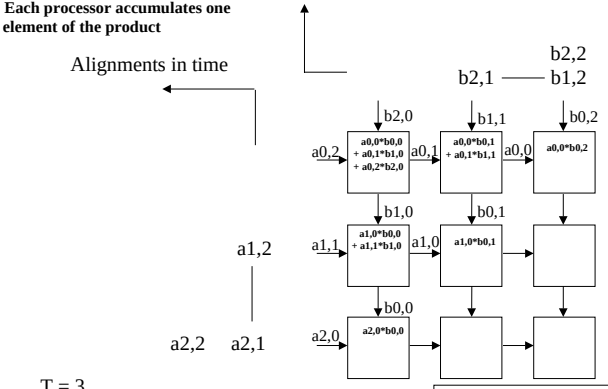


Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply

Systolic Array Example:
3x3 Systolic Array Matrix Multiplication

- **Processors arranged in a 2-D grid**
- **Each processor accumulates one element of the product**

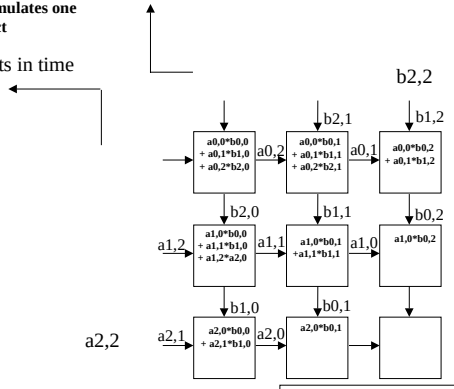


Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply

Systolic Array Example: 3x3 Systolic Array Matrix Multiplication

- Processors arranged in a 2-D grid
- Each processor accumulates one element of the product



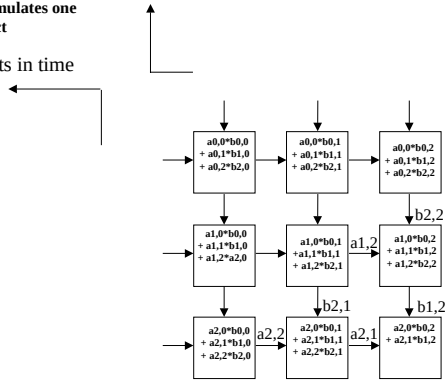
T = 4

Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply

Systolic Array Example:
3x3 Systolic Array Matrix Multiplication

- **Processors arranged in a 2-D grid**
- **Each processor accumulates one element of the product**



T = 6

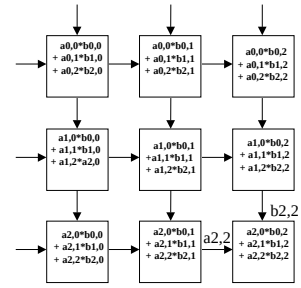
Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply

Systolic Array Example:
3x3 Systolic Array Matrix Multiplication

- **Processors arranged in a 2-D grid**
- **Each processor accumulates one element of the product**

Alignments in time



Done

T = 7

Example source: http://www.cs.hmc.edu/courses/2001/spring/cs156/

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

PES UNIVERSITY ONLINE

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

- Hardware implementation clock cycle
  - ASIC clock cycle $\approx$ 1.5 ns
    - Application Specific Integrated Circuit
  - FPGA clock cycle $\approx$ 4 ns
    - Field Programmable Gate Array

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

- Hardware implementation clock cycle
  - ASIC clock cycle $\approx$ 1.5 ns
    - Application Specific Integrated Circuit
  - FPGA clock cycle $\approx$ 4 ns
    - Field Programmable Gate Array
- For $n = 64$, number of clock cycles
  $= 3 \times 64 - 2 = 191$

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

- Hardware implementation clock cycle
  - ASIC clock cycle $\approx$ 1.5 ns
    - Application Specific Integrated Circuit
  - FPGA clock cycle $\approx$ 4 ns
    - Field Programmable Gate Array
- For $n = 64$, number of clock cycles $= 3 \times 64 - 2 = 191$
- Time for 64×64 matrix multiply $= 191 \times 4 = 764ns$

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

- Hardware implementation clock cycle
  - ASIC clock cycle $\approx$ 1.5 ns
    - Application Specific Integrated Circuit
  - FPGA clock cycle $\approx$ 4 ns
    - Field Programmable Gate Array
- For $n = 64$, number of clock cycles $= 3 \times 64 - 2 = 191$
- Time for 64×64 matrix multiply $= 191 \times 4 = 764ns$
- Time for ten 64×64 matrix multiplies $= 7640ns$

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

- Hardware implementation clock cycle
  - ASIC clock cycle $\approx$ 1.5 ns
    - Application Specific Integrated Circuit
  - FPGA clock cycle $\approx$ 4 ns
    - Field Programmable Gate Array
- For $n = 64$, number of clock cycles $= 3 \times 64 - 2 = 191$
- Time for 64×64 matrix multiply $= 191 \times 4 = 764ns$
- Time for ten 64×64 matrix multiplies $= 7640ns$

Hardware Speedup Over Software

- Speedup for ten 64×64 matrix multiplies

$$= \frac{\text{Software time}}{\text{Hardware time}} = \frac{131072}{7640} \approx \mathbf{17}$$

Systolic Array Matrix Multiply
Hardware Matrix Multiplication Time

- Hardware implementation clock cycle
  - ASIC clock cycle $\approx$ 1.5 ns
    - Application Specific Integrated Circuit
  - FPGA clock cycle $\approx$ 4 ns
    - Field Programmable Gate Array
- For $n = 64$, number of clock cycles $= 3 \times 64 - 2 = 191$
- Time for 64×64 matrix multiply $= 191 \times 4 = 764ns$
- Time for ten 64×64 matrix multiplies $= 7640ns$

Hardware Speedup Over Software

- Speedup for ten 64×64 matrix multiplies

$$= \frac{\text{Software time}}{\text{Hardware time}} = \frac{131072}{7640} \approx \mathbf{17}$$

Hardware implementations of algorithms can be many times faster than software

Systolic Array Matrix Multiply
Google TPU (Tensor Processing Unit)

PES UNIVERSITY ONLINE

Systolic Array Matrix Multiply
Google TPU (Tensor Processing Unit)

Need for Hardware Acceleration

- As mentioned in the beginning of the course, Moore's Law is slowing down

Systolic Array Matrix Multiply
Google TPU (Tensor Processing Unit)

Need for Hardware Acceleration

- As mentioned in the beginning of the course, Moore's Law is slowing down
- Google makes software (Tensorflow) for ML/AI

Systolic Array Matrix Multiply
Google TPU (Tensor Processing Unit)

Need for Hardware Acceleration

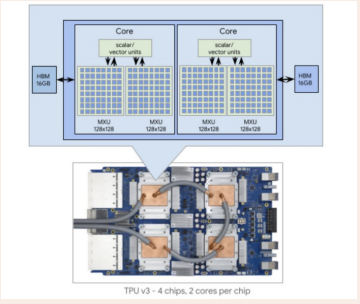
- As mentioned in the beginning of the course, Moore's Law is slowing down
- Google makes software (Tensorflow) for ML/AI
- Google also makes hardware (Tensor Processing Units) for ML/AI

Systolic Array Matrix Multiply
Google TPU (Tensor Processing Unit)

Need for Hardware Acceleration

- As mentioned in the beginning of the course, Moore's Law is slowing down
- Google makes software (Tensorflow) for ML/AI
- Google also makes hardware (Tensor Processing Units) for ML/AI

Google Tensor Processing Unit



Source: https://cloud.google.com/tpu

- 128×128 systolic array matrix multipliers
- 700MHz (1.4ns clock period)

Systolic Array Matrix Multiply

Think About It

- Systolic array $n \times n$ matrix multiplication takes $3n - 2$ clock cycles
- But each PE computes only for n clock cycles
- Can a new matrix multiplication be started before the previous one is complete?
- How many matrix multiplies can be active in the systolic array at any given time?