DS-CDMA Procedures with the Cell Broadband Engine

Project Exam - Group 07gr942

Jes Toft Kristensen
Peter August Simonsen

9^{th} SEMESTER
APPLIED SIGNAL PROCESSING AND IMPLEMENTATION

January 11^{th}, 2008

Outline

Peter

1. Application Analysis
   - Project Purpose
   - Design Methodology
   - CBE
   - DS-CDMA
2. Algorithm and Mapping
   - Signal Model
   - Program Partition
   - Software Design
3. Evaluation
   - System Test
   - Project Conclusion

Jes

4. New Experiments
   - Optimized SIMD for Task 1
   - Performance with no Data Transfers
5. A Revised Algorithm Partitioning
6. Presentation Conclusions

Purpose

Project Motivation

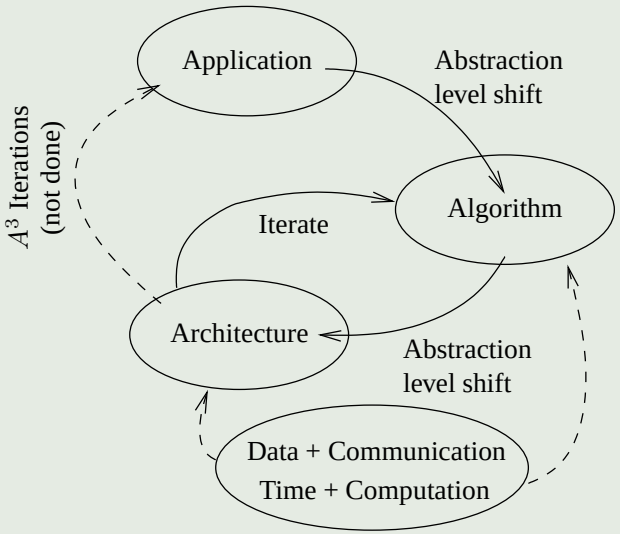
- Investigation of wireless communication algorithms on a parallel processor architecture - the Cell Broadband Engine

Initial Problem

Which factors are important in utilizing the CBE and how does this apply to a CDMA demodulation implementation?

A^3 model based design trajectory

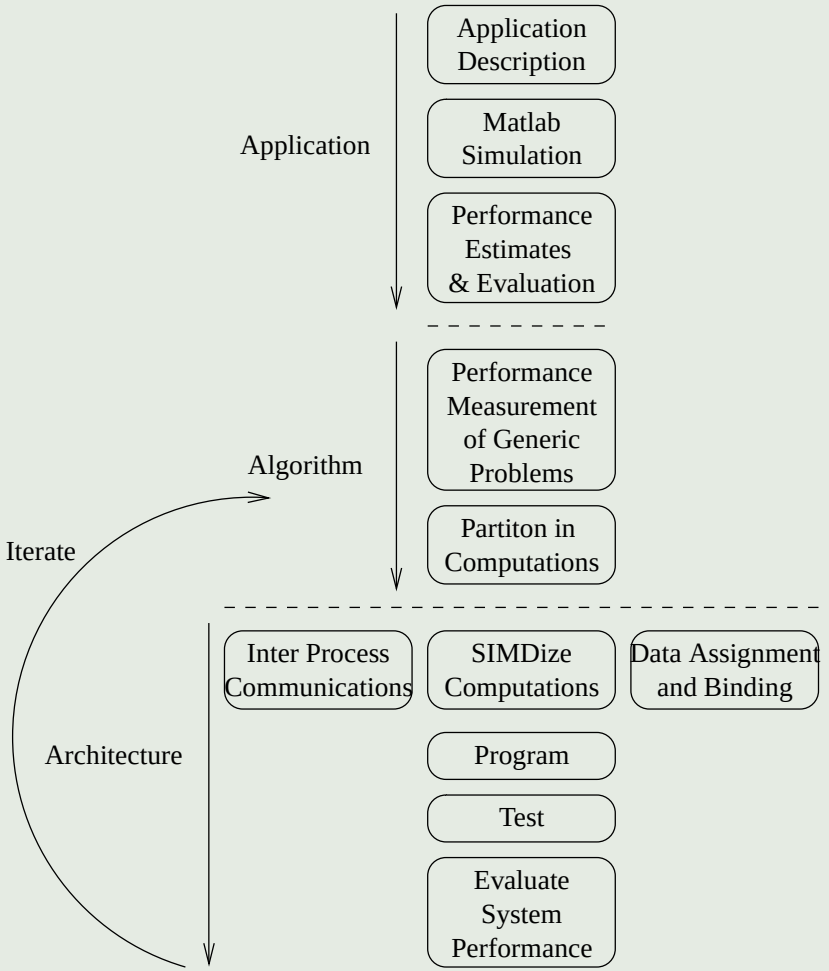
A^3 model



Focus on:

- Application partitioning for platform
- Mapping challenges

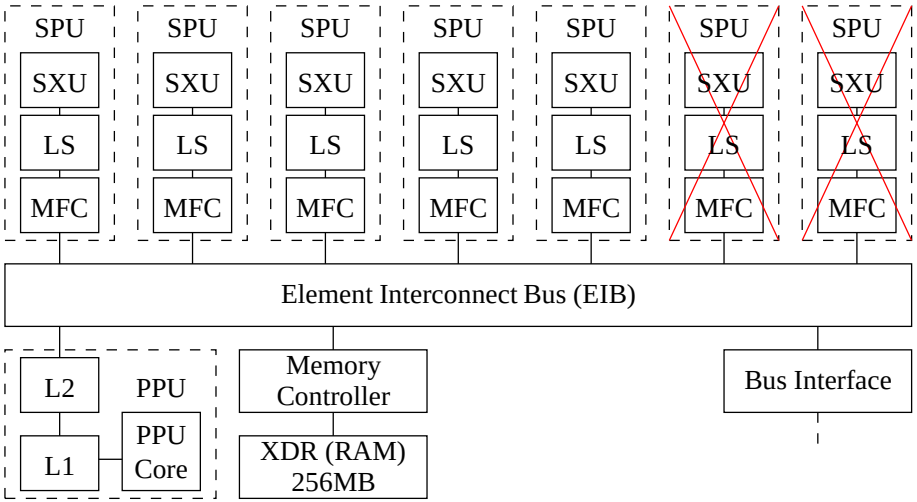
Design Trajectory



Cell Broadband Engine

Heterogenous Multiprocessor Architecture

- 1 PowerPC Unit
- 8 (6) Synergistic Processing Units



Mapping Challenges

Main issues in mapping

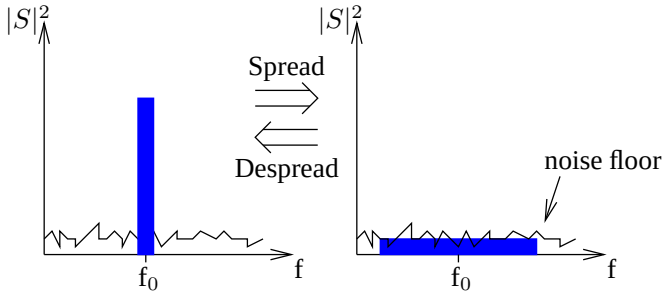
- SIMD instruction utilization
- Eliminate conditional branches
- Concurrent data transfers and data processing

DS-CDMA

Direct Sequence - Code Division Multiple Access

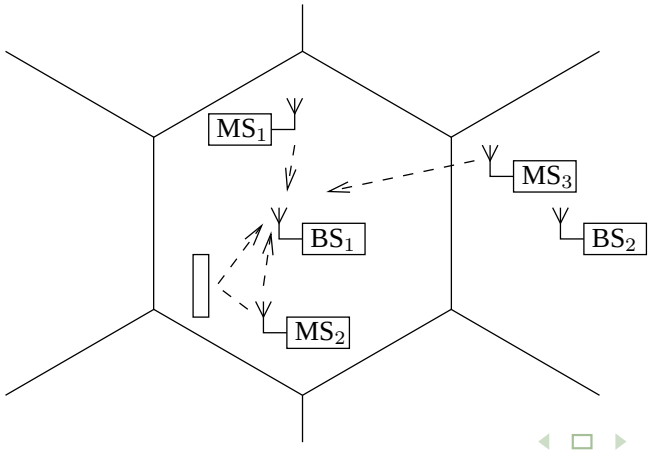
- Spreading

  Separation of users



- Scrambling

  Separation of basestation cells

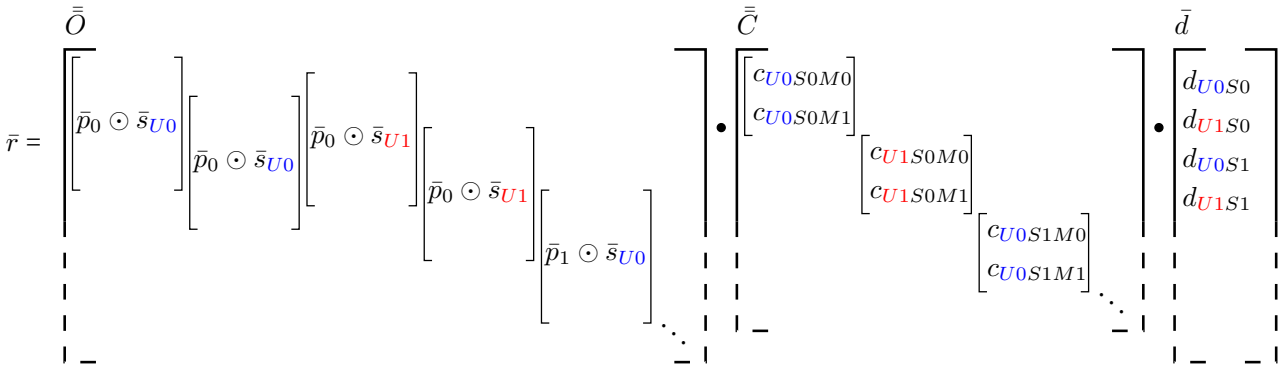


Signal Model

Signal model of received signal ($\bar{r}$) at base station:

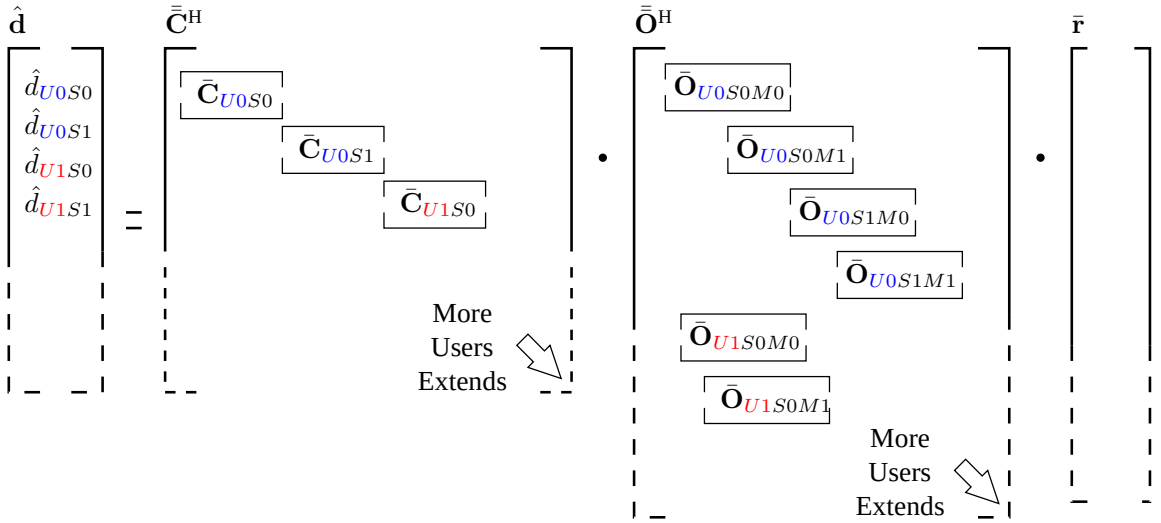
- Asynchronous communication
- Multipath fading

$$\bar{r} = \bar{\bar{O}}\bar{\bar{C}}\bar{d} \Leftrightarrow \hat{d} = \bar{\bar{C}}^H\bar{\bar{O}}^H\bar{r} \tag{1}$$



Program Partitioning

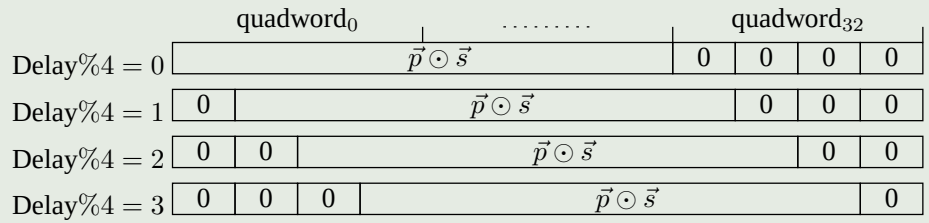
- Right to Left multiplication
  - Reduce complexity of intermediate product
    - SPU task 1: $\bar{r}' = \bar{\bar{O}}^H \bar{r}$
    - SPU task 2: $\hat{d} = \bar{\bar{C}}^H \bar{r}'$
- Division in user
  - Reduction of code generation
  - Work on entire $\hat{r}$
  - Increase algorithm latency



Alignment for SIMD

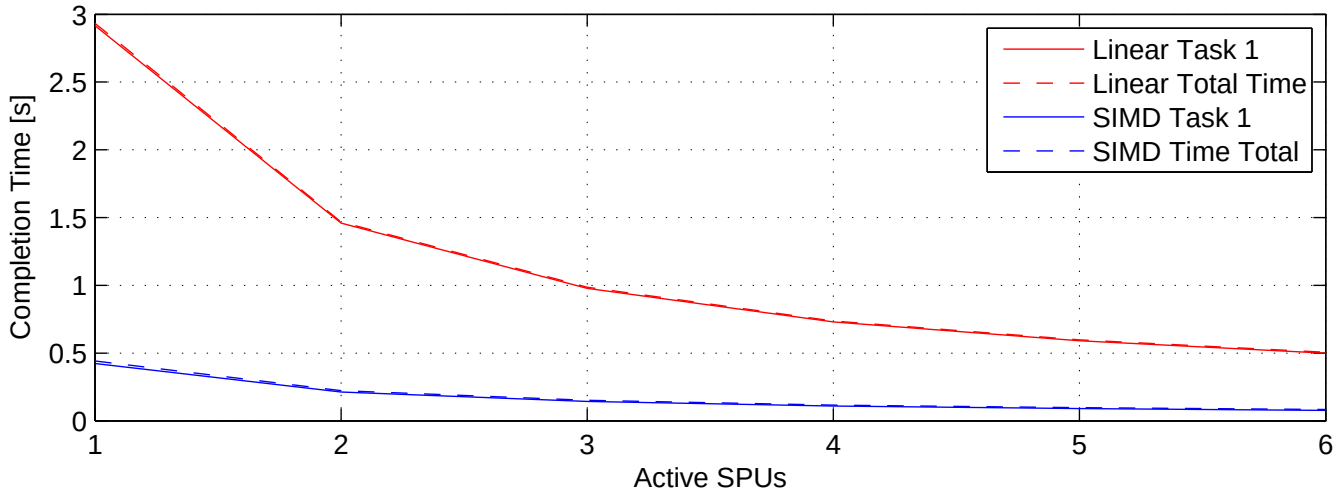
Vector alignment

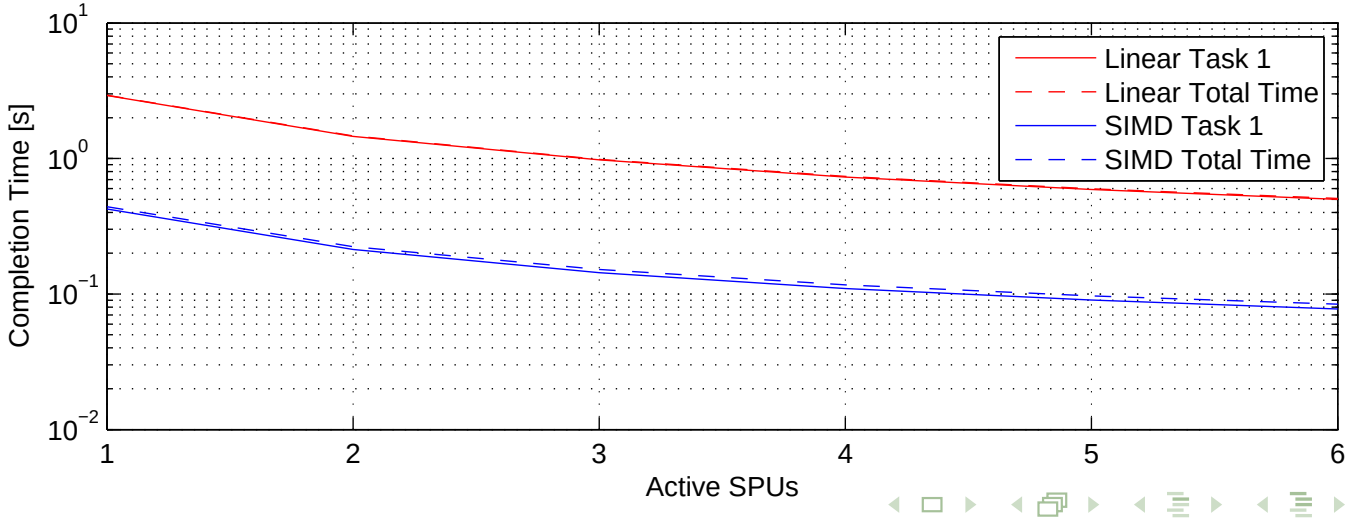
- Handling random delays by copying rows of $\bar{\bar{O}}^H$:



System Test

- Performance measured as program completion time





Results Discussion

- 10 ms completion requirement
  - 84.2 ms achieved
  - 10.3% SPU utilization
- Problems origin:
  - Inefficient pipeline utilization
  - Suboptimal compiler scheduling

Conclusion

Initial Problem

Which factors are important in utilizing the CBE and how does this apply to a CDMA demodulation implementation?

- Programming for the CBE:
  - Concurrent data transfer and processing
  - No control structures in SPU program
  - SIMD utilization with one operand constant
- CDMA application:
  - High level of inherent concurrency
  - Low SPU utilization
  - Partitioning needs revision

Outline

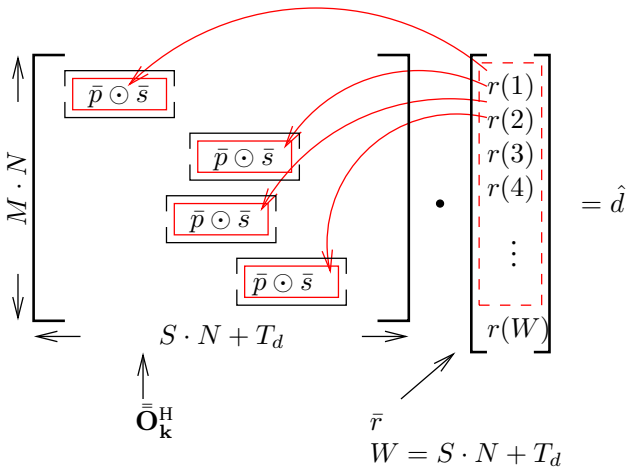
Peter

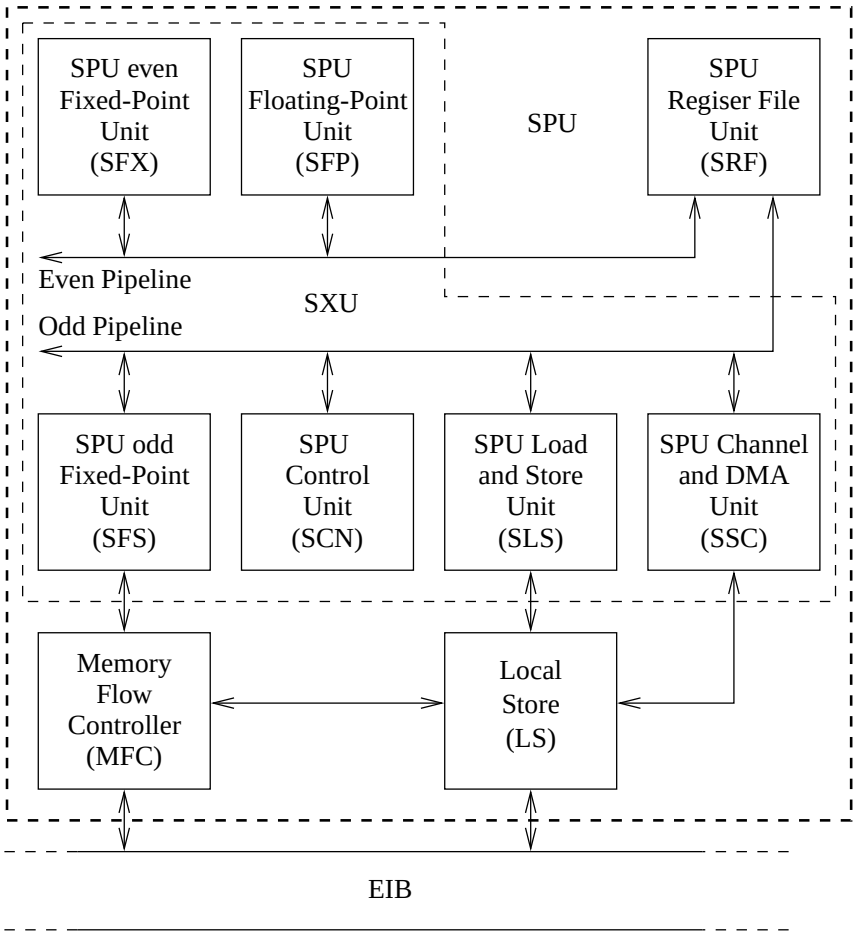
1. Application Analysis
   - Project Purpose
   - Design Methodology
   - CBE
   - DS-CDMA
2. Algorithm and Mapping
   - Signal Model
   - Program Partition
   - Software Design
3. Evaluation
   - System Test
   - Project Conclusion

Jes

4. New Experiments
   - Optimized SIMD for Task 1
   - Performance with no Data Transfers
5. A Revised Algorithm Partitioning
6. Presentation Conclusions

Project SIMD for Task 1

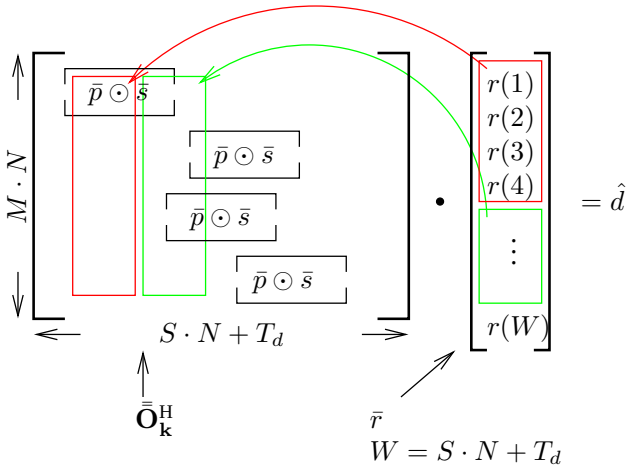


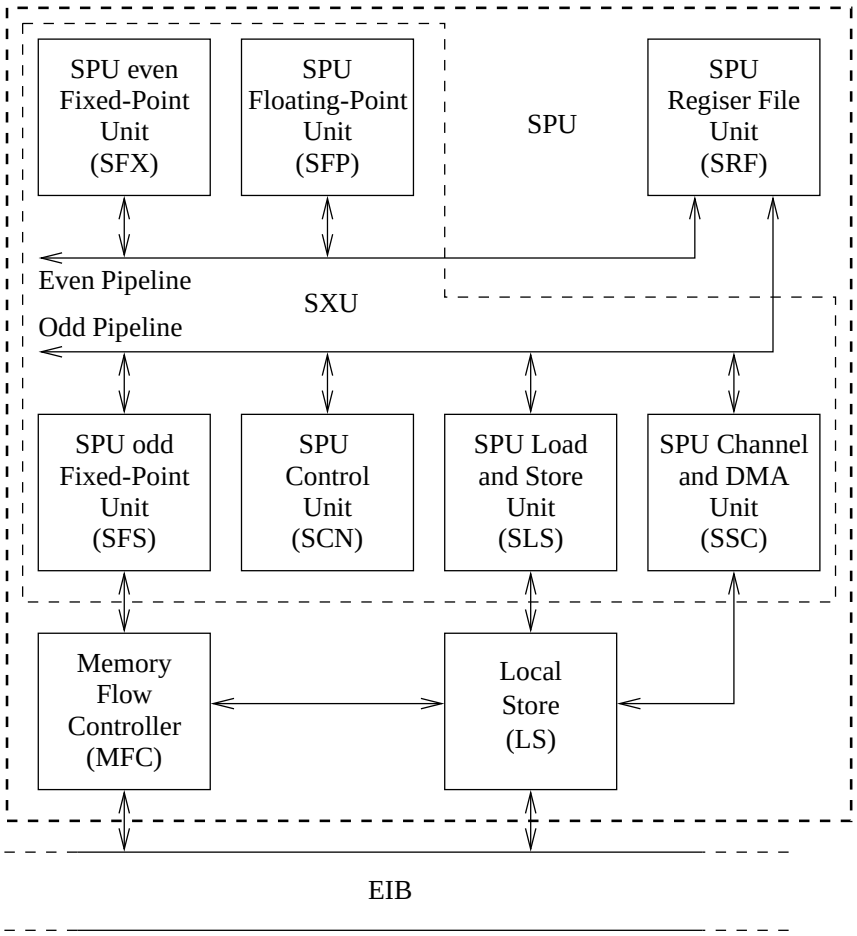


Operation

```
Loop:
   load operand 1   (odd)
   load operand 2   (odd)
   multiply         (even)
   store            (odd)
```

Improved SIMD for Task 1

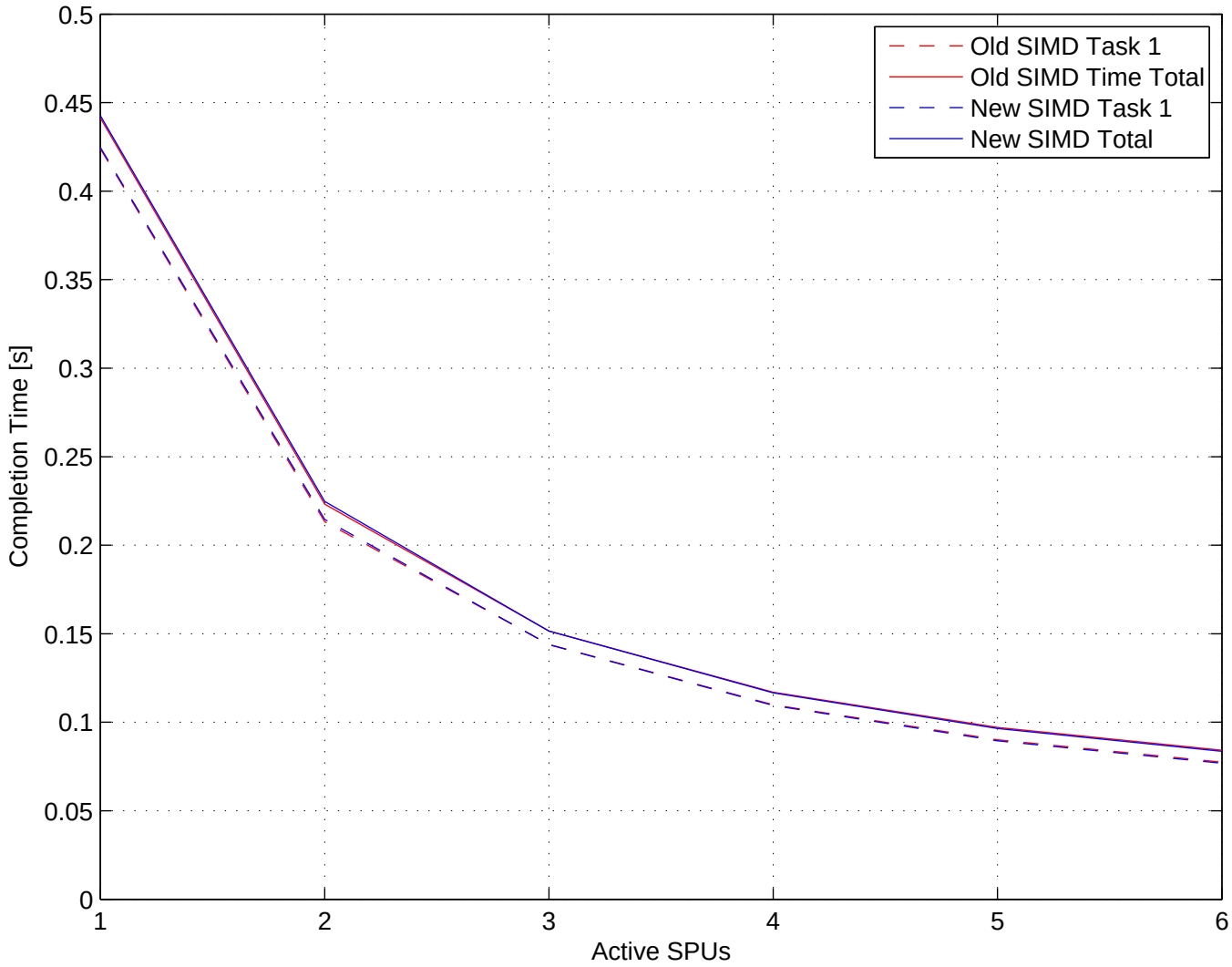




Operation

```
load operand 1   (odd)
Loop:
    load operand 2   (odd)
    multiply         (even)
    store            (odd)
```

Result of SIMD Change

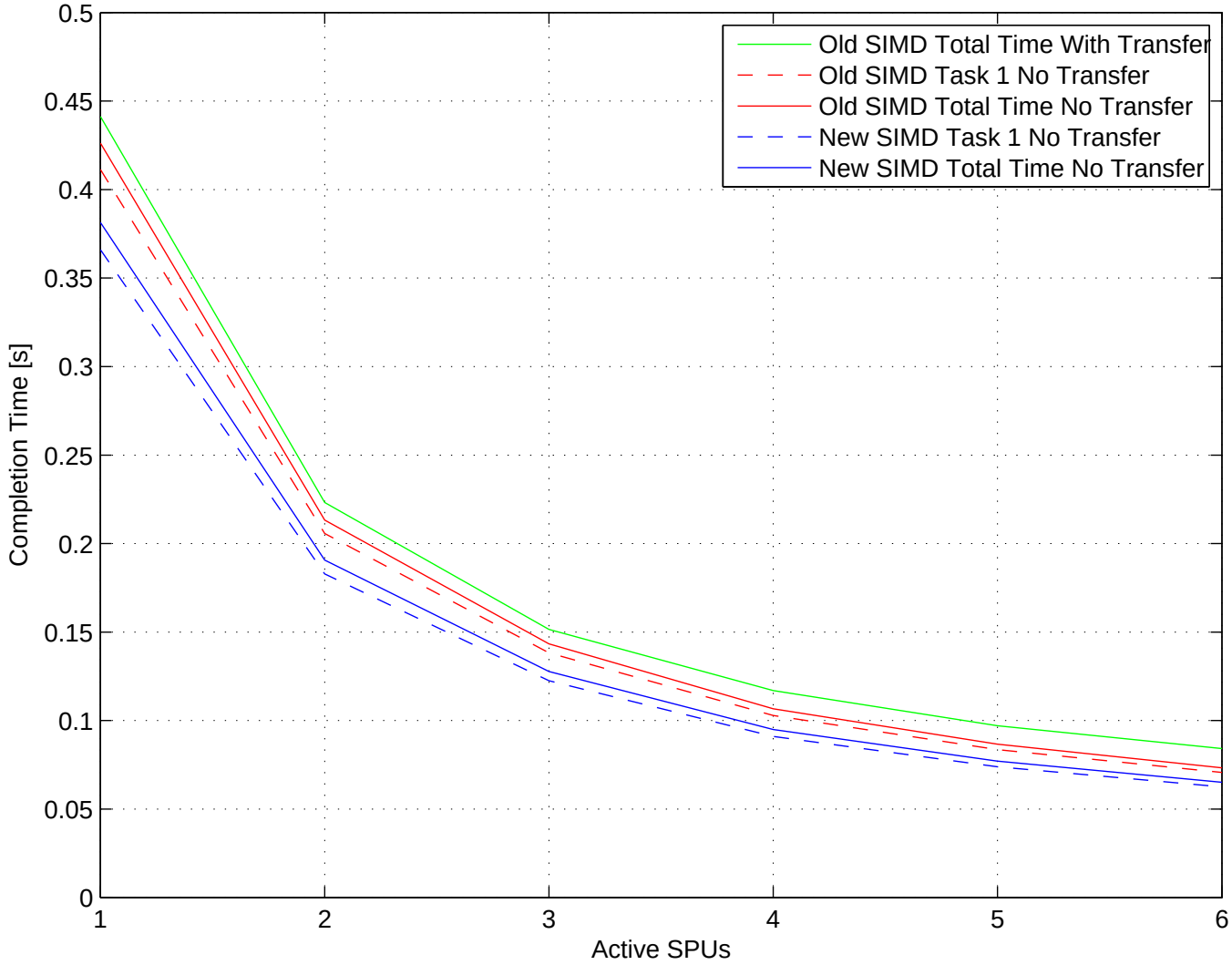


Performance Measure with no Data Transfers

Measurement

- Removed all data transfers from code, except:
  - Initializations
  - Transfer of $\bar{D}$ for each user
- Does this yield a significant improvement?
- Does this have any effect on the improved SIMD implementation?

No Transfer Result



The Problem in Relation with the CBE

Needed Calculations

For 1 communication burst:

$$O_{total} = \mu \cdot 665\,856\,000 + \alpha \cdot 646\,195\,200 \quad (2)$$

$$n_{FLOPS} = \frac{665\,856\,000}{10ms} = 66.6\,\text{GFLOPS} \quad (3)$$

$$p_{utilization} = \frac{66.6\,\text{GFLOPS}}{76.8\,\text{GFLOPS}} = 87\% \quad (4)$$

The Problem in Relation with the CBE

Needed Calculations

For 1 communication burst:

$$O_{total} = \mu \cdot 665\,856\,000 + \alpha \cdot 646\,195\,200 \quad (2)$$

$$n_{FLOPS} = \frac{665\,856\,000}{10ms} = 66.6\,\text{GFLOPS} \quad (3)$$

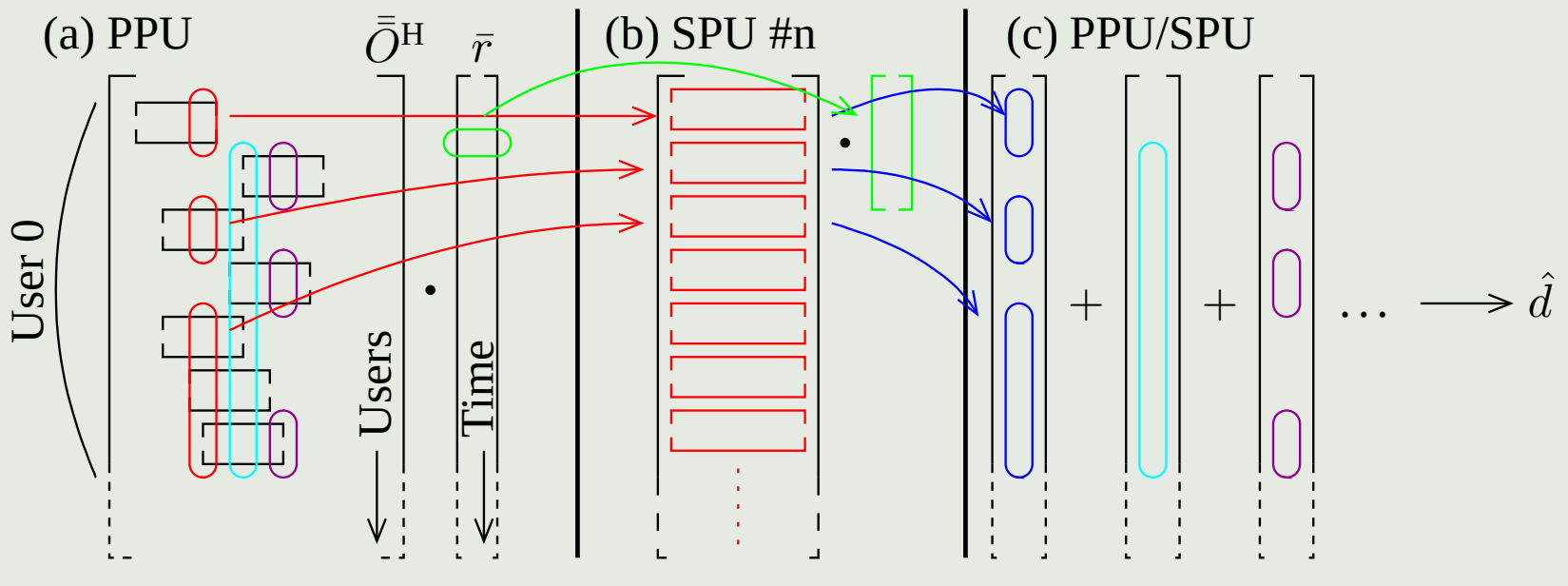
$$p_{utilization} = \frac{66.6\,\text{GFLOPS}}{76.8\,\text{GFLOPS}} = 87\% \quad (4)$$

Problem Size

$$O_{total} = K \cdot N \cdot \Big[\mu \cdot (M(S+T_d)+S+T_d)+\ldots$$
$$\ldots \quad \alpha \cdot ((M-1)(S+T_d)+S+T_d-1)\Big] \quad (5)$$

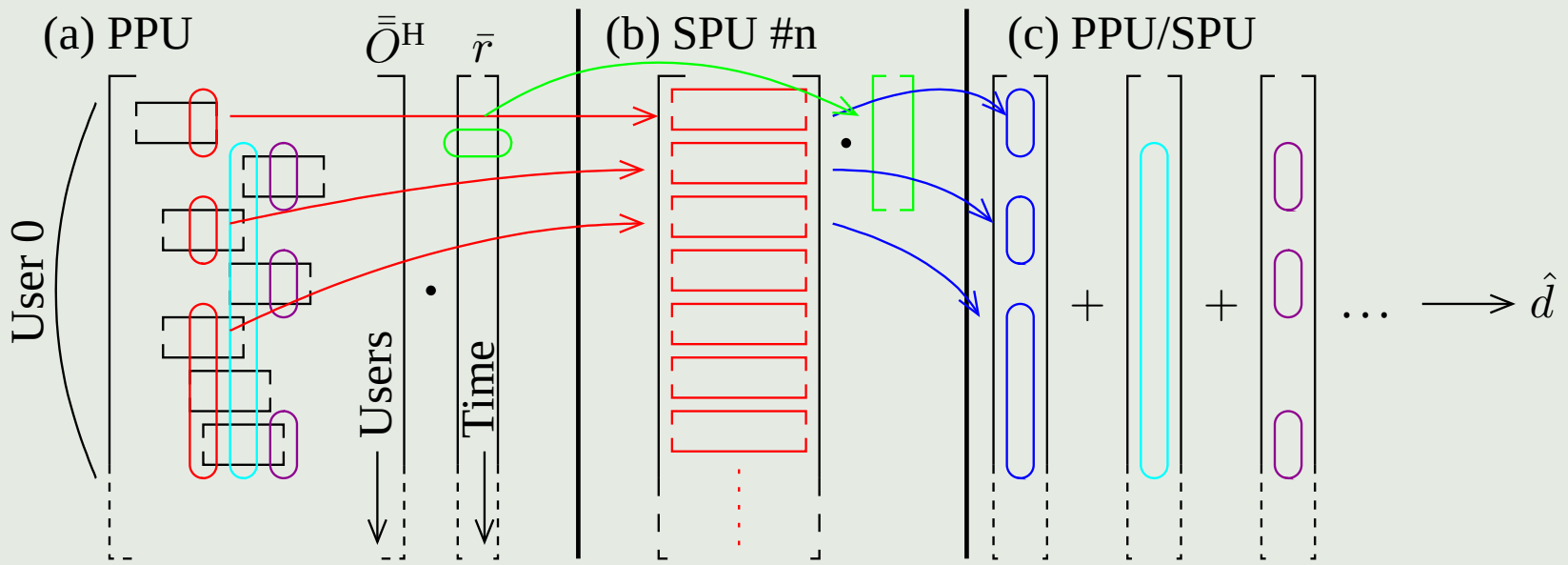
New Proposed Partition

Graphical Representation



New Proposed Partition

Graphical Representation



Outcome

- Higher SPU utilization with fewer multiply by zeroes
- More work done by PPU but more data transferred to and from SPUs

Achievements

In the Project

1. Mapped real problem to parallel architecture
2. Concurrency successfully extracted from problem
3. SIMD instructions are used
4. SPU architecture not fully utilized

Achievements

In the Project

1. Mapped real problem to parallel architecture
2. Concurrency successfully extracted from problem
3. SIMD instructions are used
4. SPU architecture not fully utilized

In the Presentation

5. Performed further tests of
   - Better utilization of the SPU architecture
   - Performance without data transfers
6. A revised algorithm partitioning
   - Handle delays on PPU
   - Utilize SPU architecture as in experiment
   - Process data as it arrives

End

Thank you for your time.

Questions?