



# Parallel Vector & Signal Processing

Mark Mitchell

CodeSourcery, LLC

April 3, 2003

# Deliverables

---

- **VSIPL++ Specification**
  - CodeSourcery et. al.
- **VSIPL++ Reference Implementation**
  - CodeSourcery.
- **High-Performance Implementations**
  - Various vendors.

# Objectives

---

- Parallelism
- Performance
- Portability
  - ANSI/ISO C++
- Compatibility with VSIPL
  - On a conceptual level.
  - Allow VSIPL to be used to implement VSIPL++.

# Major Components

---

- Data distribution:
  - Mapping from logical arrays to physical processors.
- High-performance serial computation:
  - Expression templates:
    - Intuitive syntax.
    - Loop fusion, elimination of temporaries.
  - VSIPL
    - Can be used to implement VSIPL++.

# Data Distribution Model

---

- Views
  - Lightweight access to data.
- Blocks
  - Logically rectangular arrays of data.
- Partitions
  - Block-cyclic data distribution determines subblocks.
- Processor Map
  - Map from from subblocks to physical processors.

# Computational Model

---

Data driven computation:

- Computation occurs on the processor that owns the data being updated.

Parallel computation is divided into chunks that can be performed efficiently in serial.

# Implementation of zherk

---

A rank-k update of Hermitian matrix C:

$$C \leftarrow \alpha * A * \text{conjug}(A)^t + \beta * C$$

```
CMatrix A(10,15);  
CMatrix C(10,10);  
C = alpha * prodh(A,A)  
      + beta * C;
```



# Parallel Vector & Signal Processing

Mark Mitchell

CodeSourcery, LLC

April 3, 2003