CMSC 180:
Introduction to Parallel Computing

Jaderick P. Pabico
Institute of Computer Science, College of Arts and Sciences
University of the Philippines Los Baños, College 4031, Laguna

2nd Semester 2020-2021

Dichotomy of Parallel Computing Platforms

- An explicitly parallel program must specify concurrency and interaction between concurrent subtasks.

Dichotomy of Parallel Computing Platforms

- An explicitly parallel program must specify concurrency and interaction between concurrent subtasks.

Control Structure

Dichotomy of Parallel Computing Platforms

- An explicitly parallel program must specify concurrency and interaction between concurrent subtasks.

Control Structure

Communication Model

Control Structure of Parallel Programs

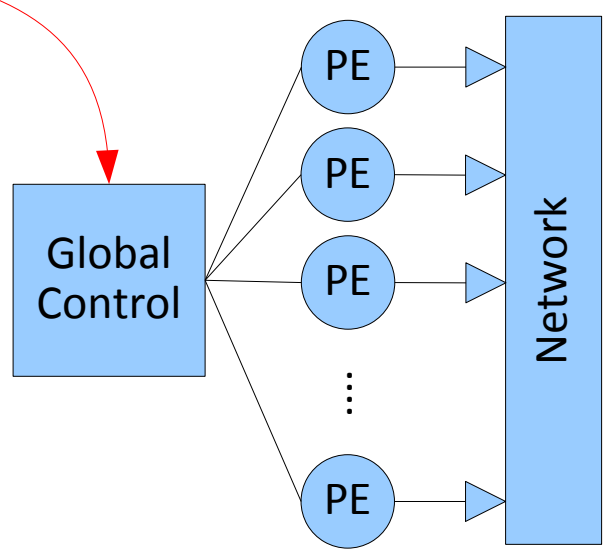
- Parallelism can be expressed at various levels of granularity
  - per line of instruction
  - per block of instructions (e.g., within loops)
  - per subroutines (e.g., functions)
  - per process
- At these granularities, there exists a range of models, along with corresponding architectural support.

Control Structure of Parallel Programs

- Processing elements (PE) in parallel computers either:
  - operate under the centralized control of a single control unit , or
  - work independently.

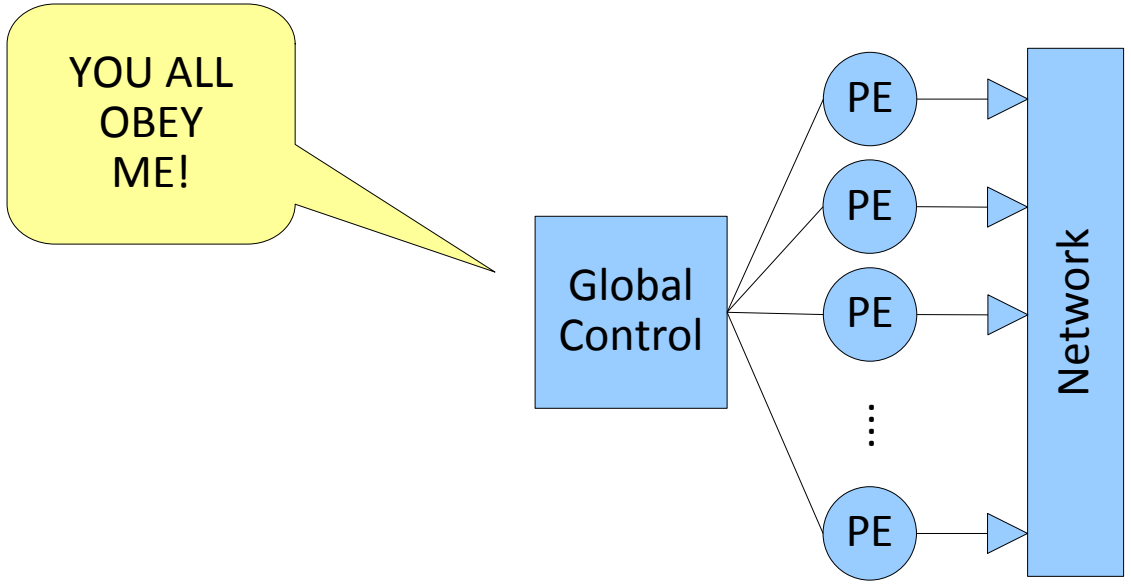
Control Structure of Parallel Programs

- Processing elements (PE) in parallel computers either:
  - operate under the centralized control of a single control unit, or
  - work independently.



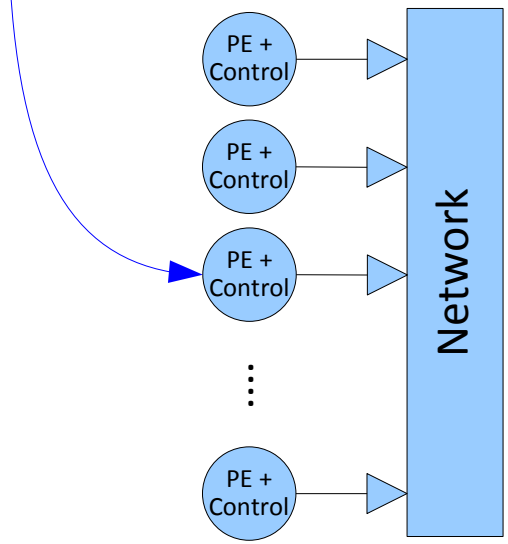
Control Structure of Parallel Programs

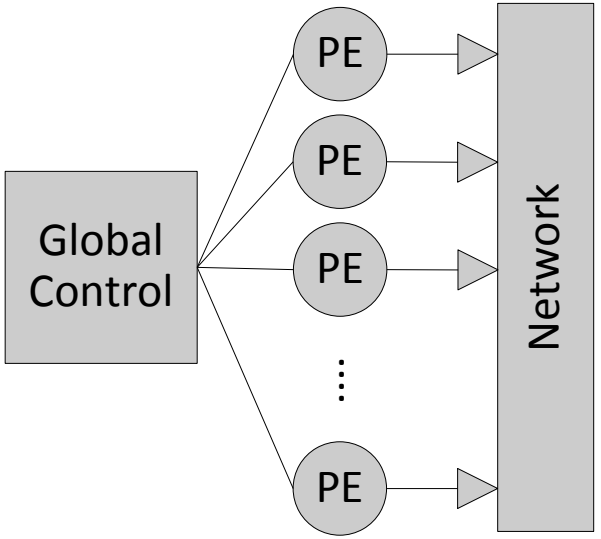
- Processing elements (PE) in parallel computers either:
  - operate under the centralized control of a single control unit, or
  - work independently.



Control Structure of Parallel Programs

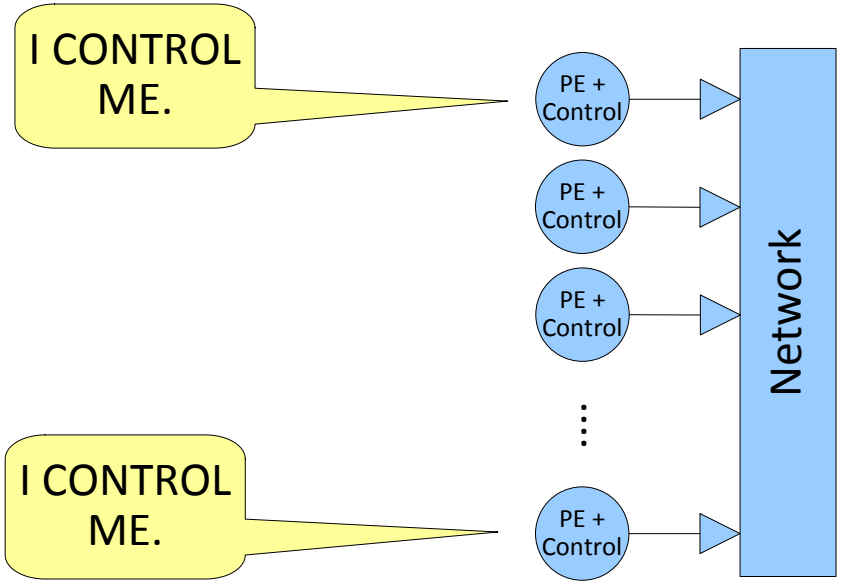
- Processing elements (PE) in parallel computers either:
  - operate under the centralized control of a single control unit, or
  - work independently.

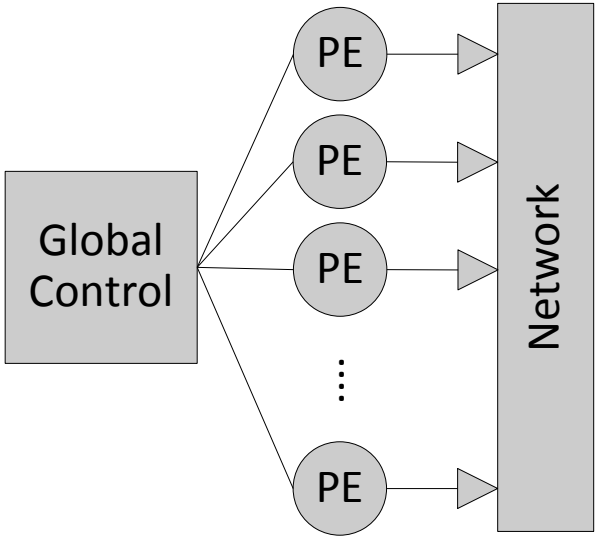




Control Structure of Parallel Programs

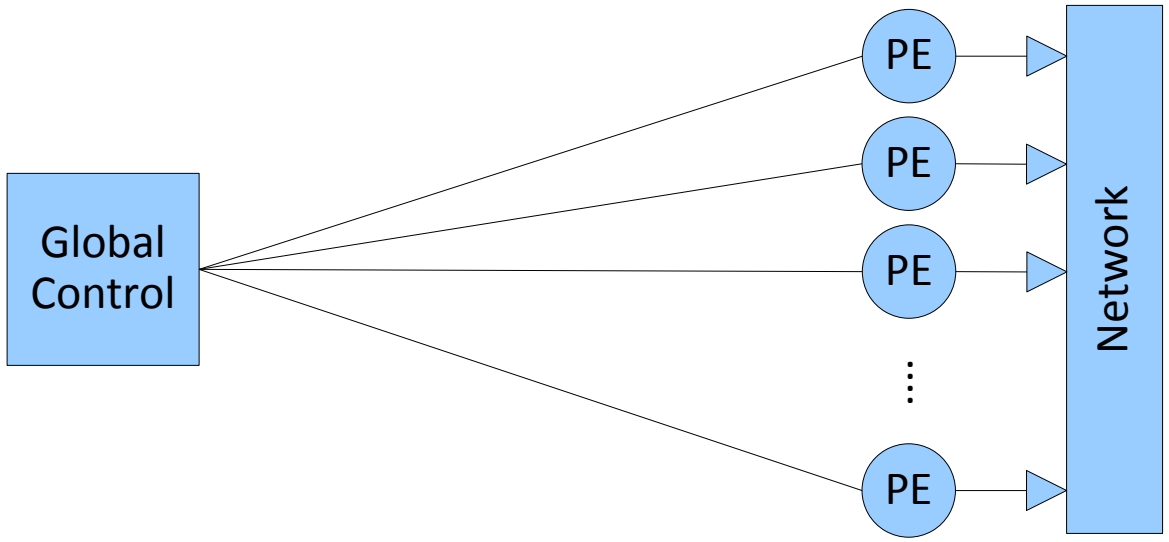
- Processing elements (PE) in parallel computers either:
  - operate under the centralized control of a single control unit, or
  - work independently.





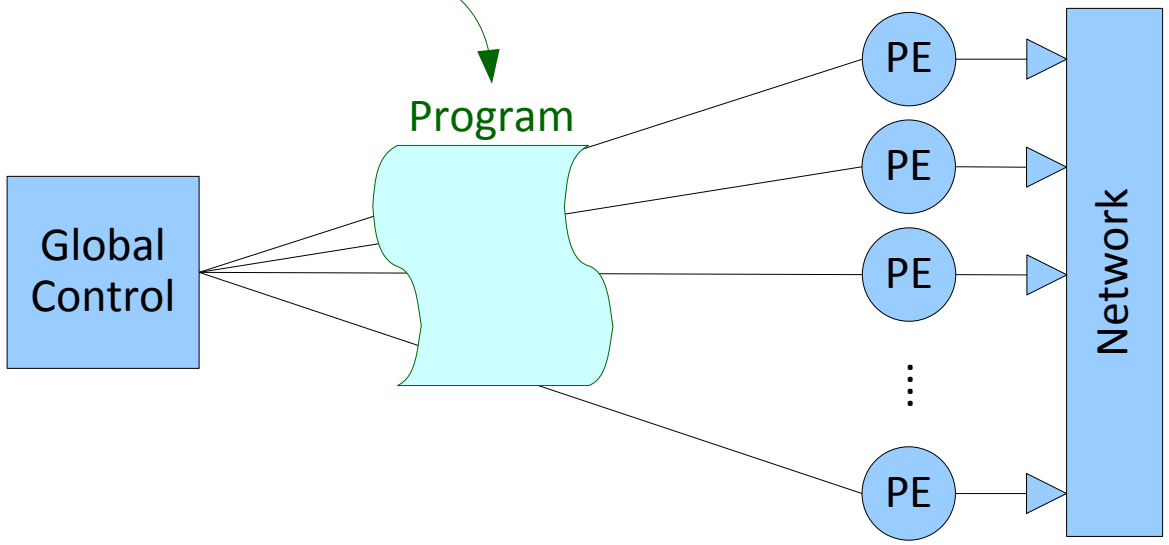
Control Structure of Parallel Programs

- If there is a single control unit that dispatches the same instruction to various processors (that work on different data), the model is referred to as single instruction, multiple data stream (SIMD).



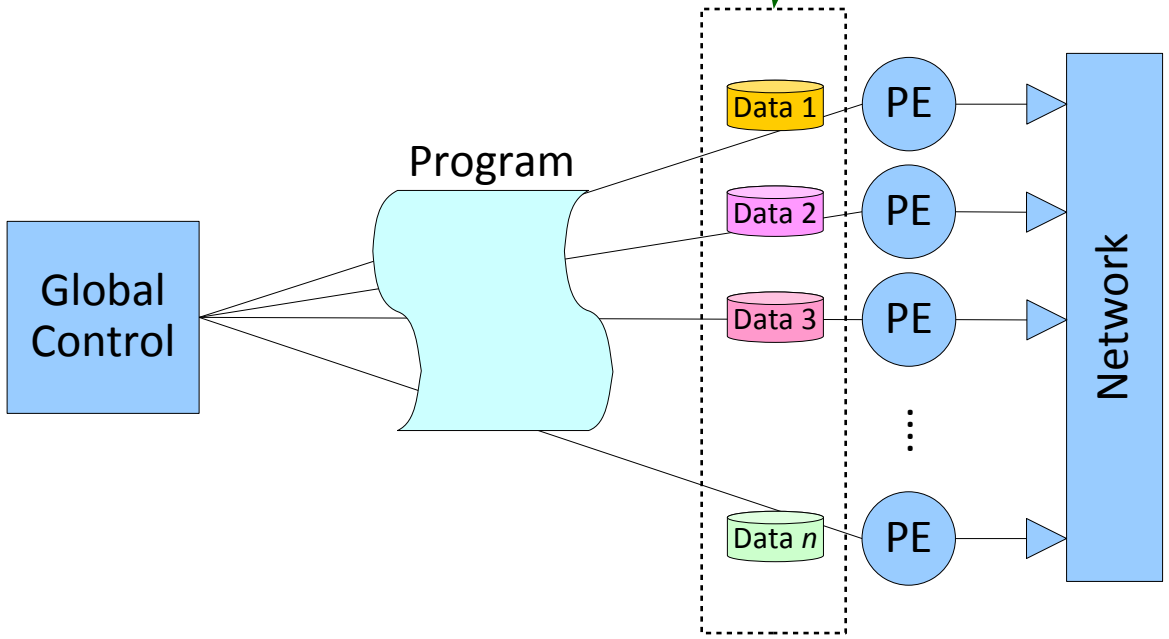
Control Structure of Parallel Programs

- If there is a single control unit that dispatches the same instruction to various processors (that work on different data), the model is referred to as single instruction, multiple data stream (SIMD).



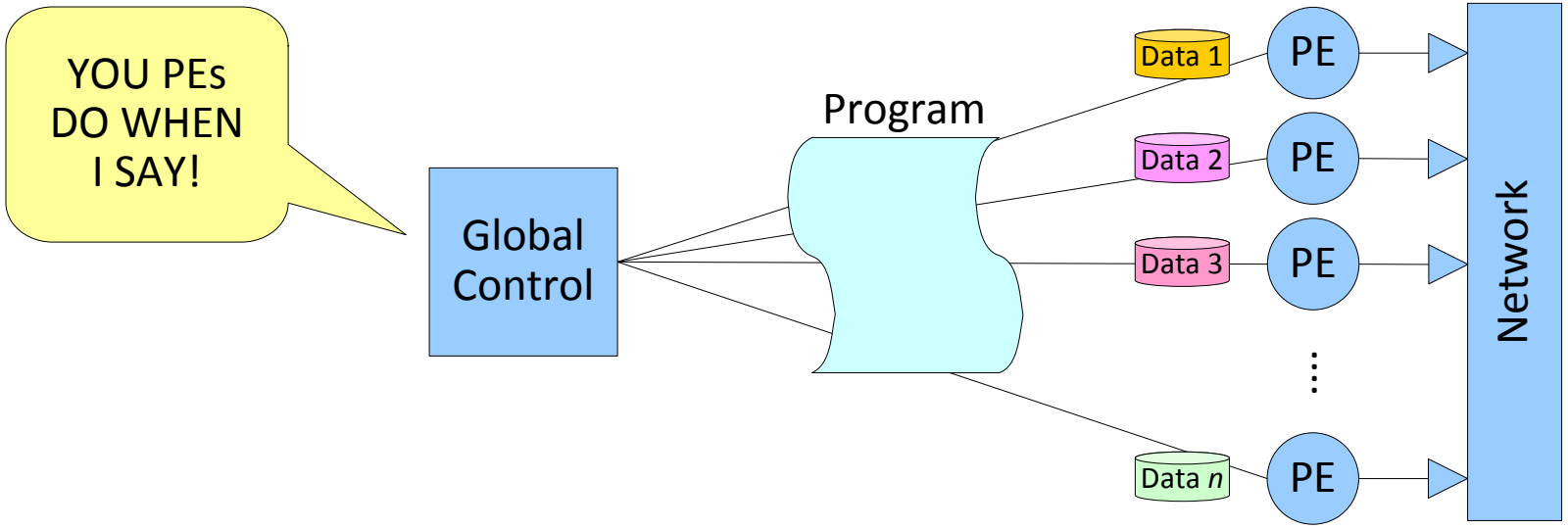
Control Structure of Parallel Programs

- If there is a single control unit that dispatches the same instruction to various processors (that work on different data), the model is referred to as single instruction, multiple data stream (SIMD).



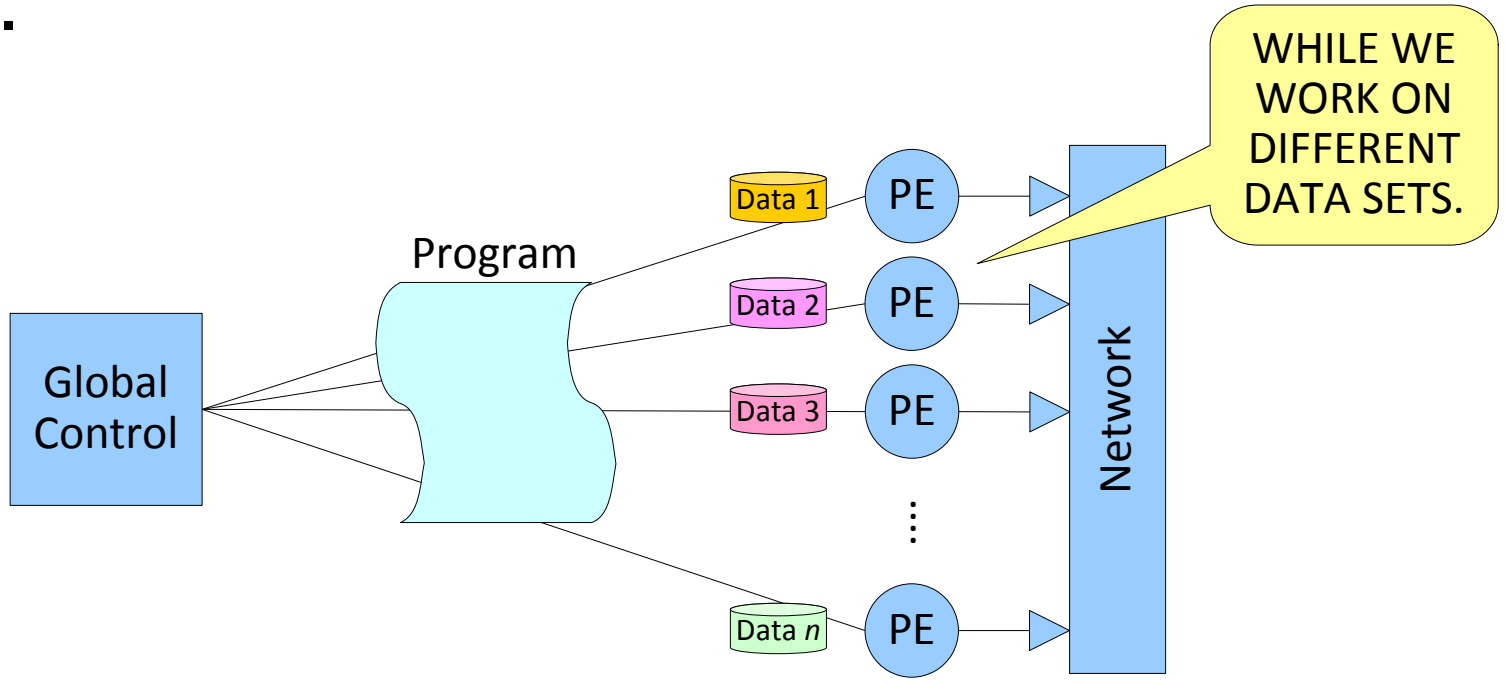
Control Structure of Parallel Programs

- If there is a single control unit that dispatches the same instruction to various processors (that work on different data), the model is referred to as single instruction, multiple data stream (SIMD).



Control Structure of Parallel Programs

- If there is a single control unit that dispatches the same instruction to various processors (that work on different data), the model is referred to as single instruction, multiple data stream (SIMD).



SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

```
1  if (b==0) c=a;
2  else c=a/b;
```

SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

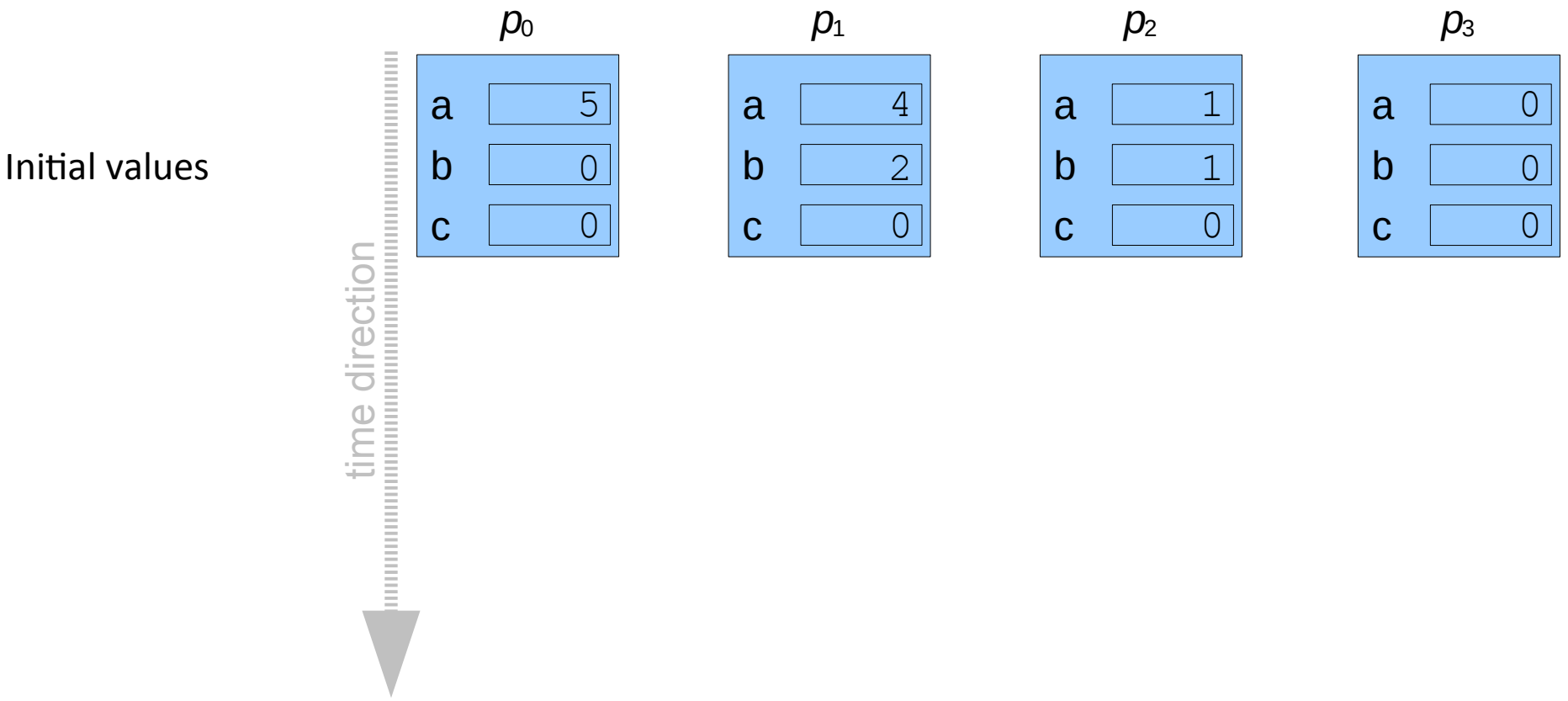
```
1  if (b==0) c=a;
2  else c=a/b;
```

	p_0	p_1	p_2	p_3
Initial values	a 5 b 0 c 0	a 4 b 2 c 0	a 1 b 1 c 0	a 0 b 0 c 0

SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

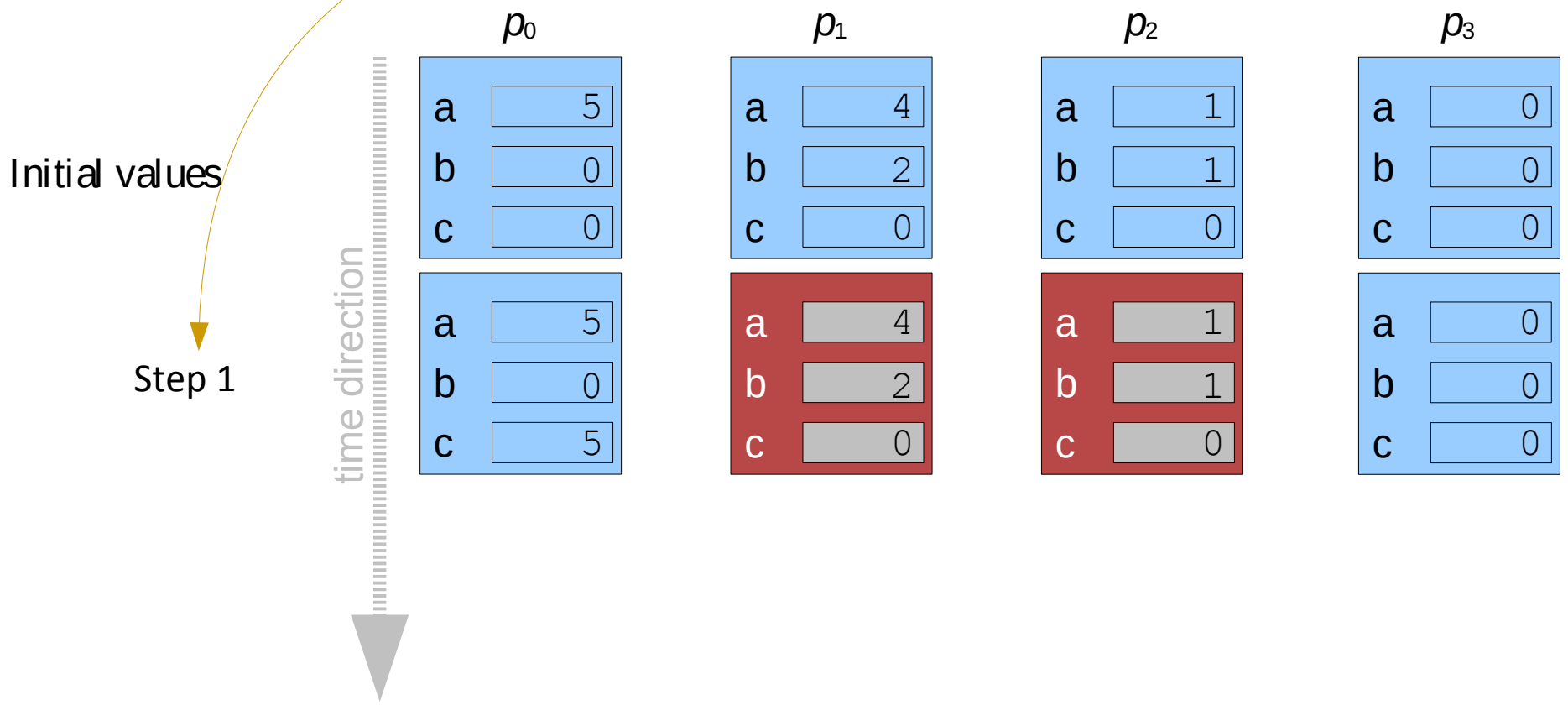
```
1  if (b==0) c=a;
2  else c=a/b;
```



SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

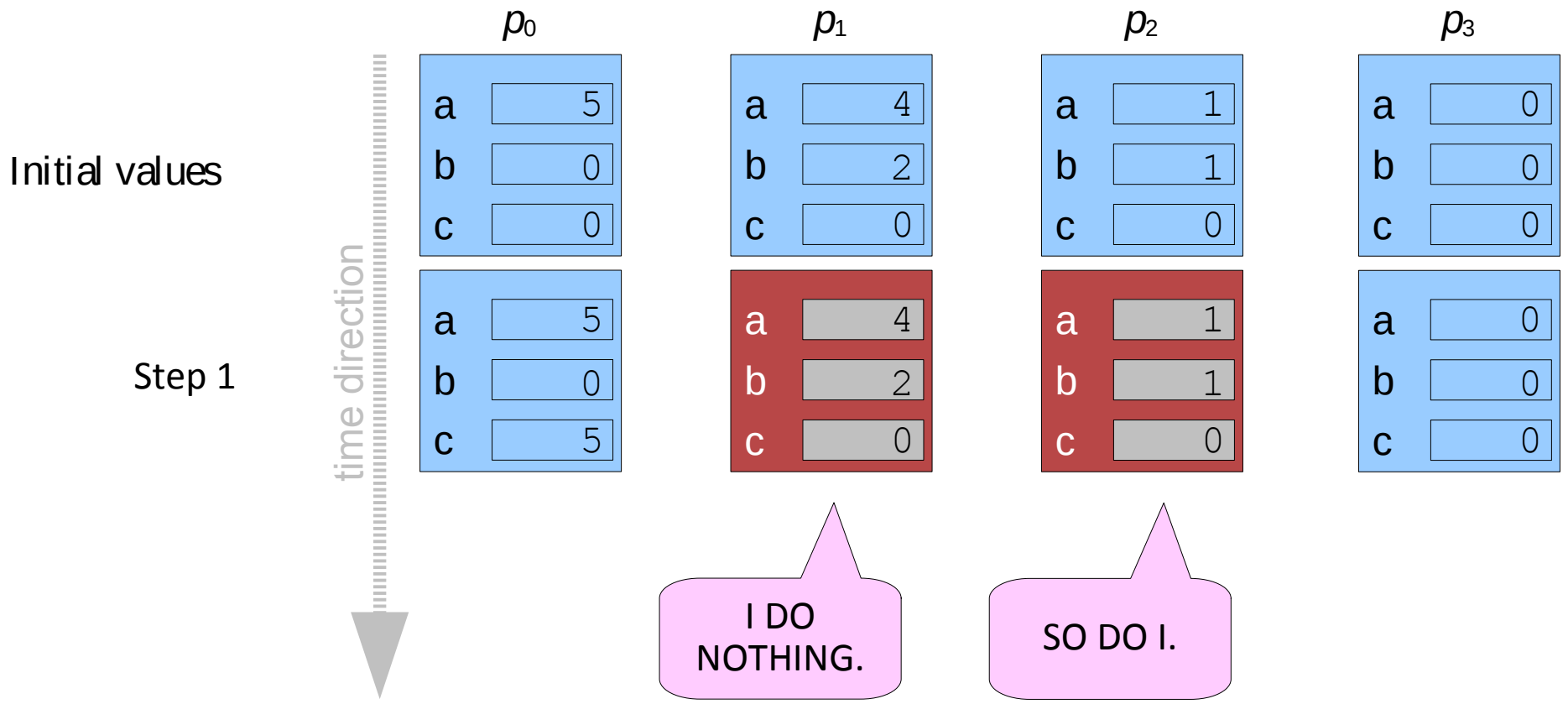
```
1  if (b==0) c=a;
2  else c=a/b;
```



SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

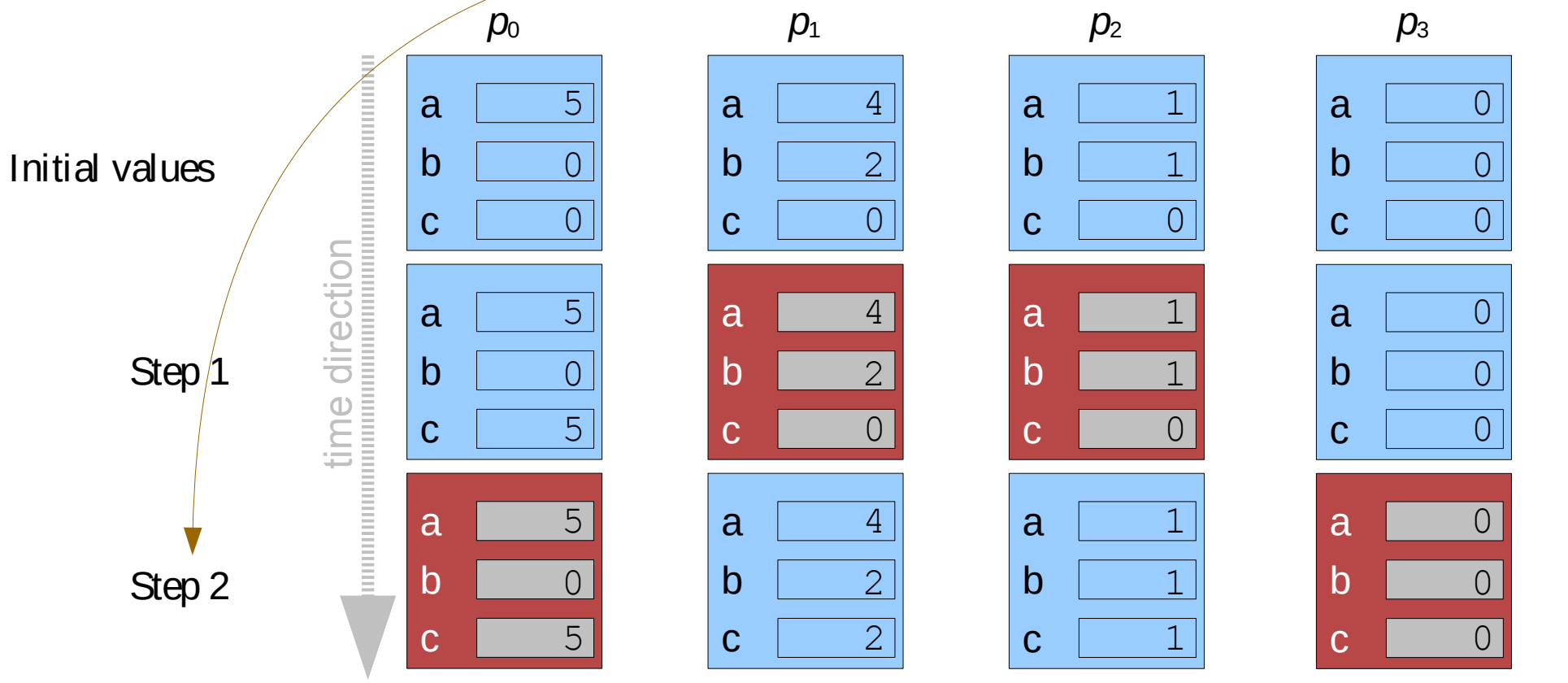
```
1  if (b==0) c=a;
2  else c=a/b;
```



SIMD Processors

- Example: Execution of a conditional statement in a 4-processor SIMD

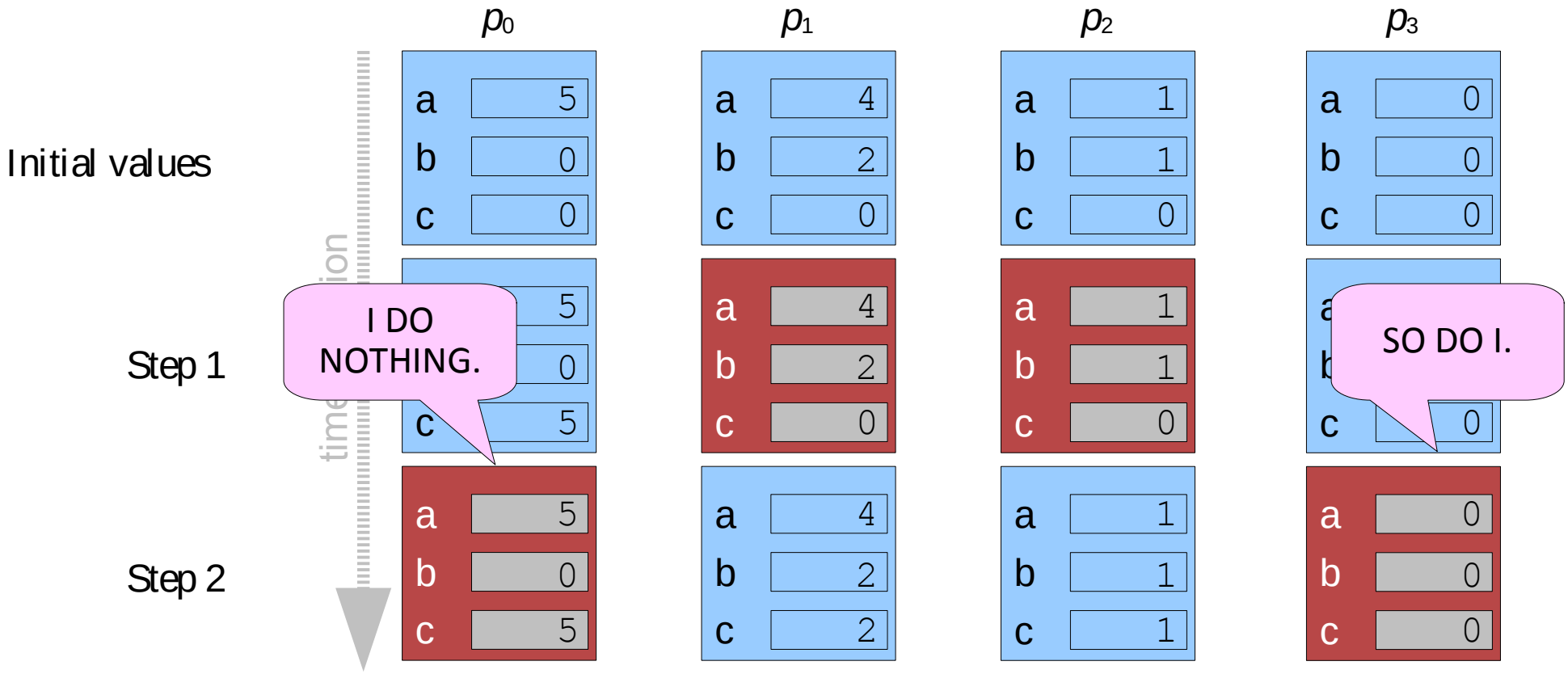
```
1  if (b==0) c=a;
2  else c=a/b;
```



SIMD Processors

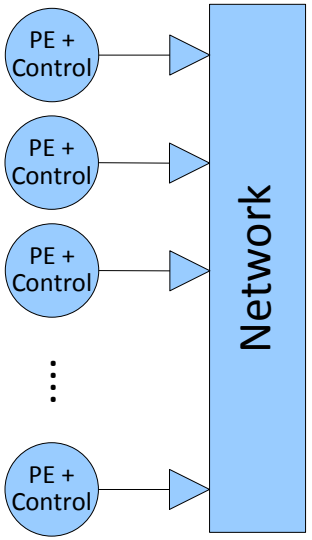
- Example: Execution of a conditional statement in a 4-processor SIMD

```
1  if (b==0) c=a;
2  else c=a/b;
```



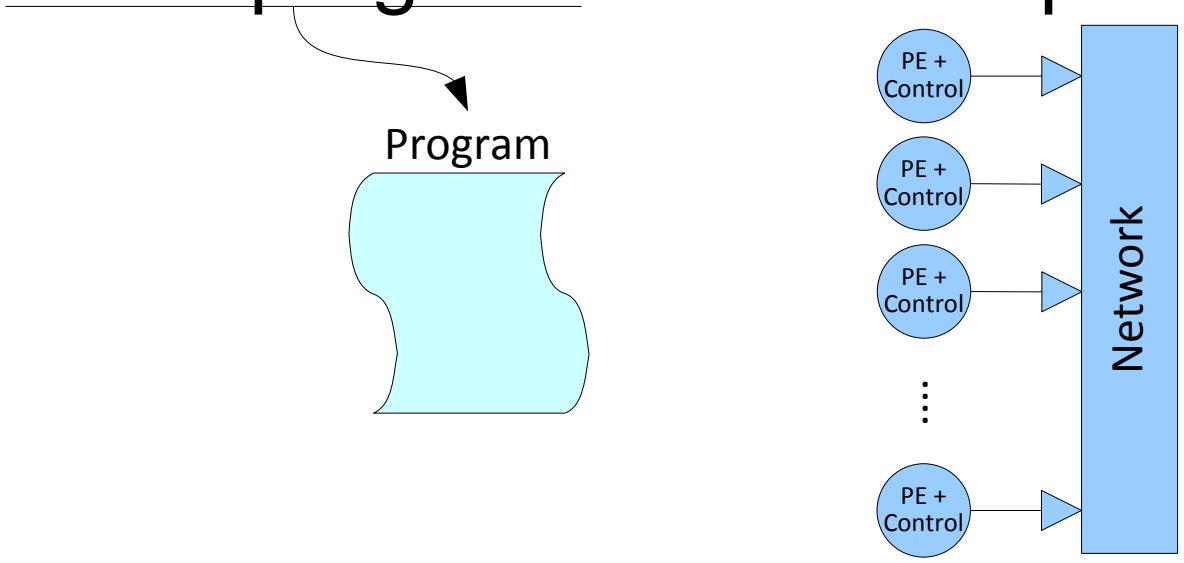
MIMD Processors

- In contrast to SIMD processor, MIMD processors can execute different programs on different processors.
- A special case of this, called single program multiple data stream (SPMD) executes the same program on different processors.



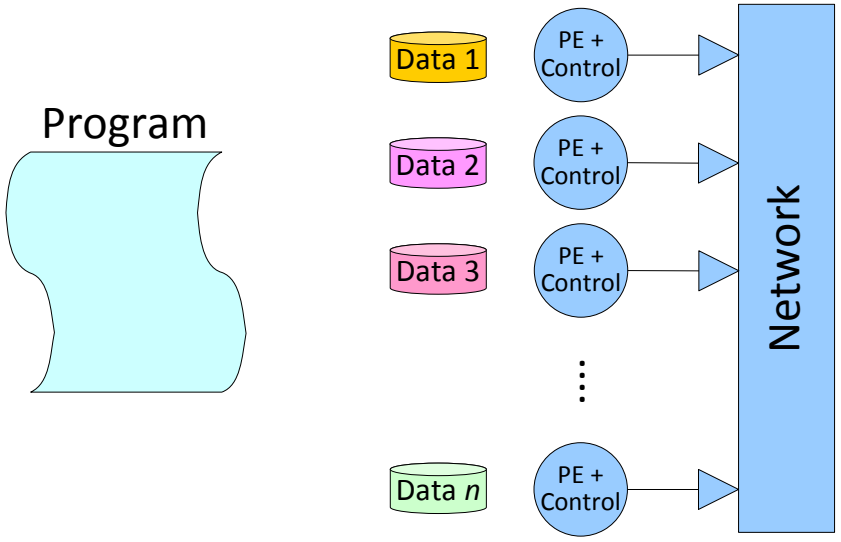
MIMD Processors

- In contrast to SIMD processor, MIMD processors can execute different programs on different processors.
- A special case of this, called single program multiple data stream (SPMD) executes the same program on different processors.



MIMD Processors

- In contrast to SIMD processor, MIMD processors can execute different programs on different processors.
- A special case of this, called single program multiple data stream (SPMD) executes the same program on different processors.



MIMD Processors

- In contrast to SIMD processor, MIMD processors can execute different programs on different processors.
- A special case of this, called single program multiple data stream (SPMD) executes the same program on different processors.
- SPMD and MIMD are closely related in terms of programming flexibility and underlying architectural support.

SIMD vs. MIMD

- SIMD require less hardware than MIMD (hint: single control unit)
- However, since SIMD are specially designed, they tend to be expensive and have long design cycles.
- Not all applications are naturally suited to SIMDs.
- In contrast, platforms supporting the SPMD model can be built from inexpensive off-the-shelf components with relatively little effort in a short amount of time.
  - Example: commodity clusters

Next Discussion

- Implicit parallelism
- Limitations of the performance of memory systems
- Dichotomy of platforms
- Communication model
- Physical organization
- Communication costs
- Messaging cost models and routing
- Mapping techniques