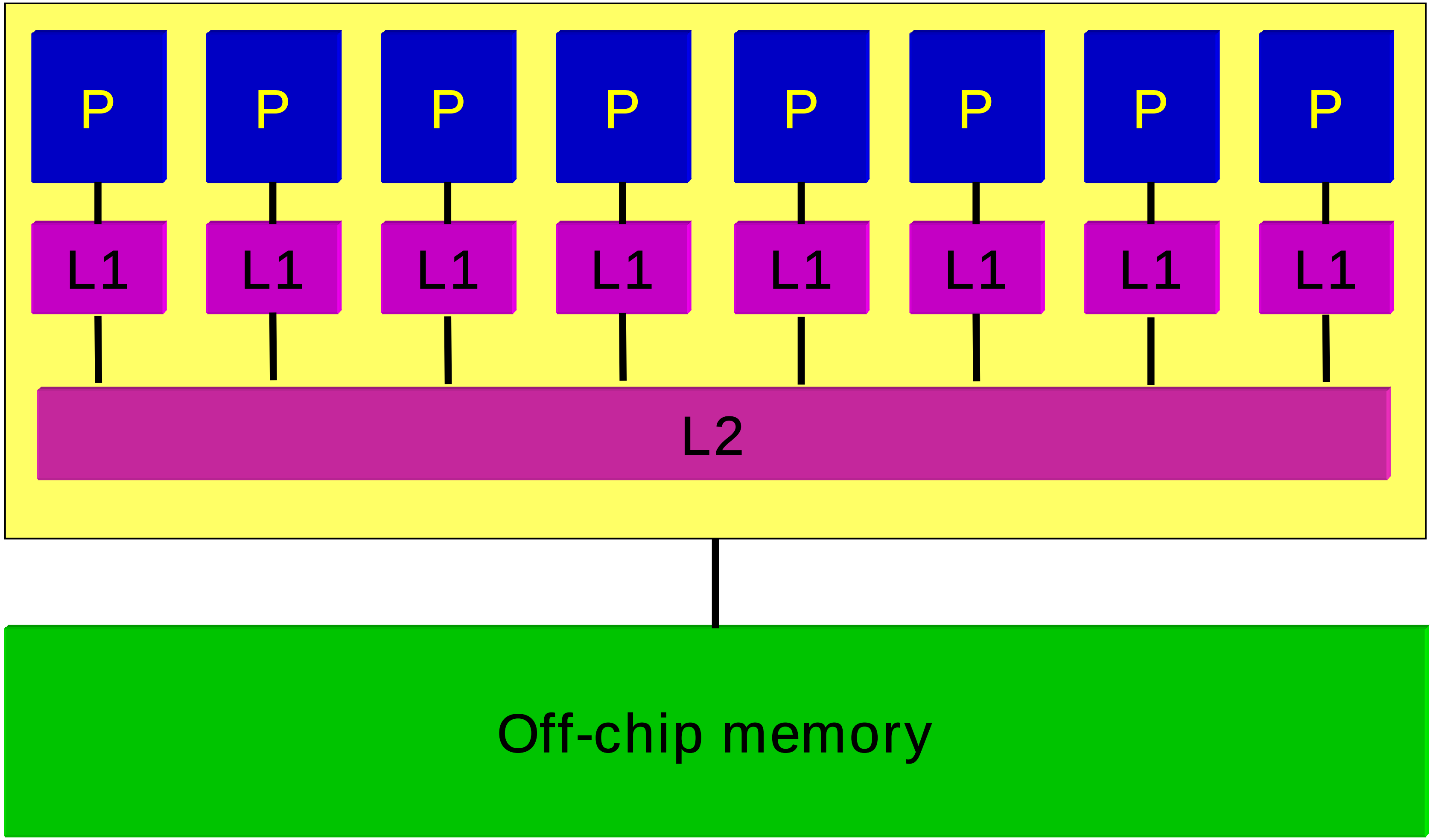
Energy-Aware Computation Duplication for Improving Reliability in Embedded Chip Multiprocessors

G. Chen, M. Kandemir, and F. Li
Department of Computer Science and Engineering
Pennsylvania State University

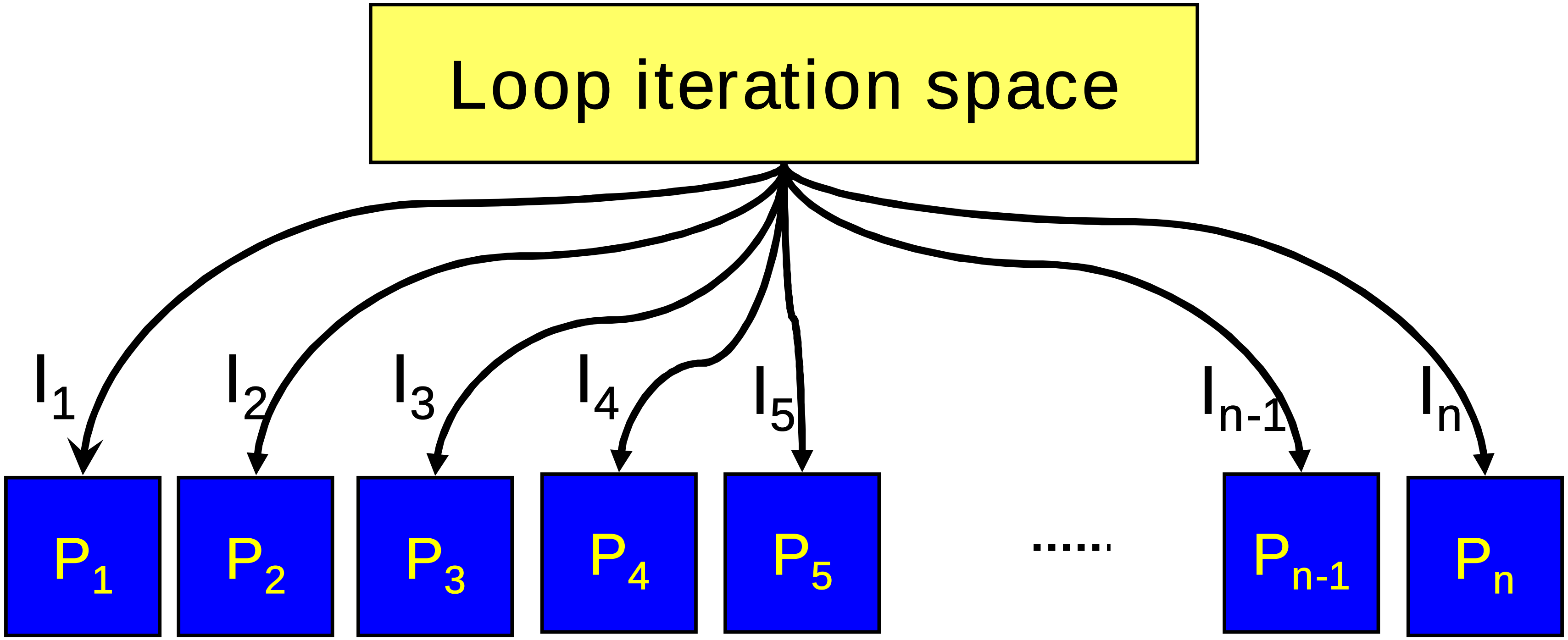
Introduction

- Advantages of chip multiprocessors (CMP)
  - Easy for verification
  - Appropriate for high-level code parallelism
  - Power efficient
- Transient errors
  - Cross-coupling, ground bounce, external terrestrial radiations…
  - Technology scaling and power-saving techniques increases embedded systems' vulnerability to transient errors
- Our goal: Utilizing on-chip parallelism for best tradeoffs between performance, power, and reliability

Chip Multiprocessor Architecture



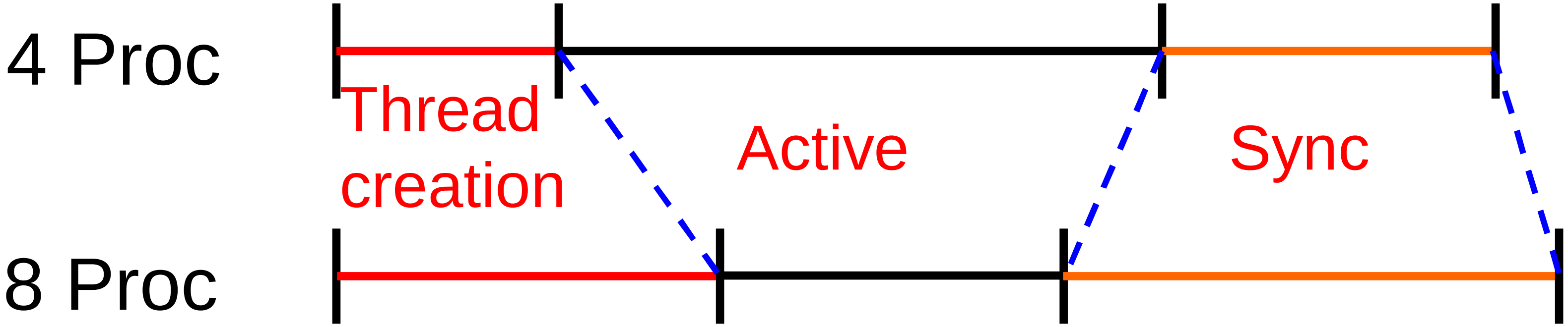
Loop Parallelization



Each processor gets a portion of iterations to execute

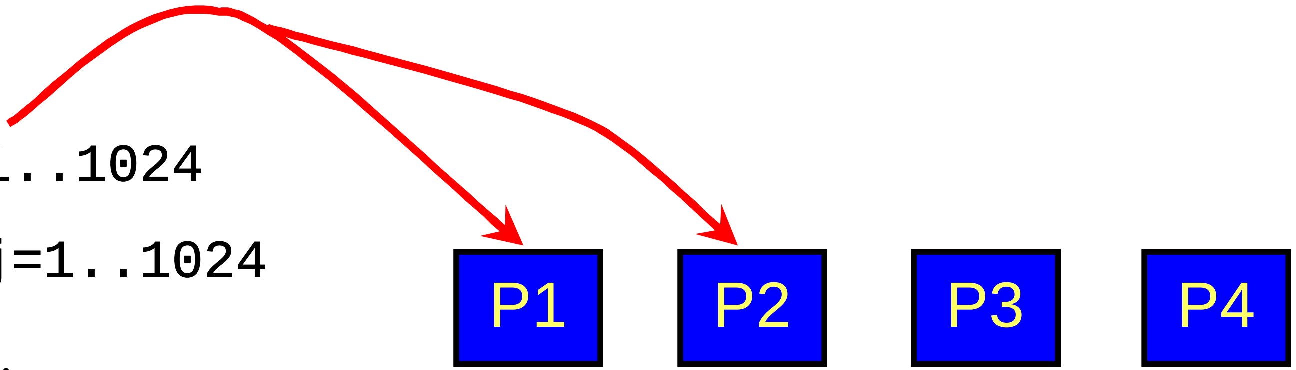
Adaptive Loop Parallelization

- Add more processors can degrade performance of a loop



- Assign the optimum number of processors to each loop

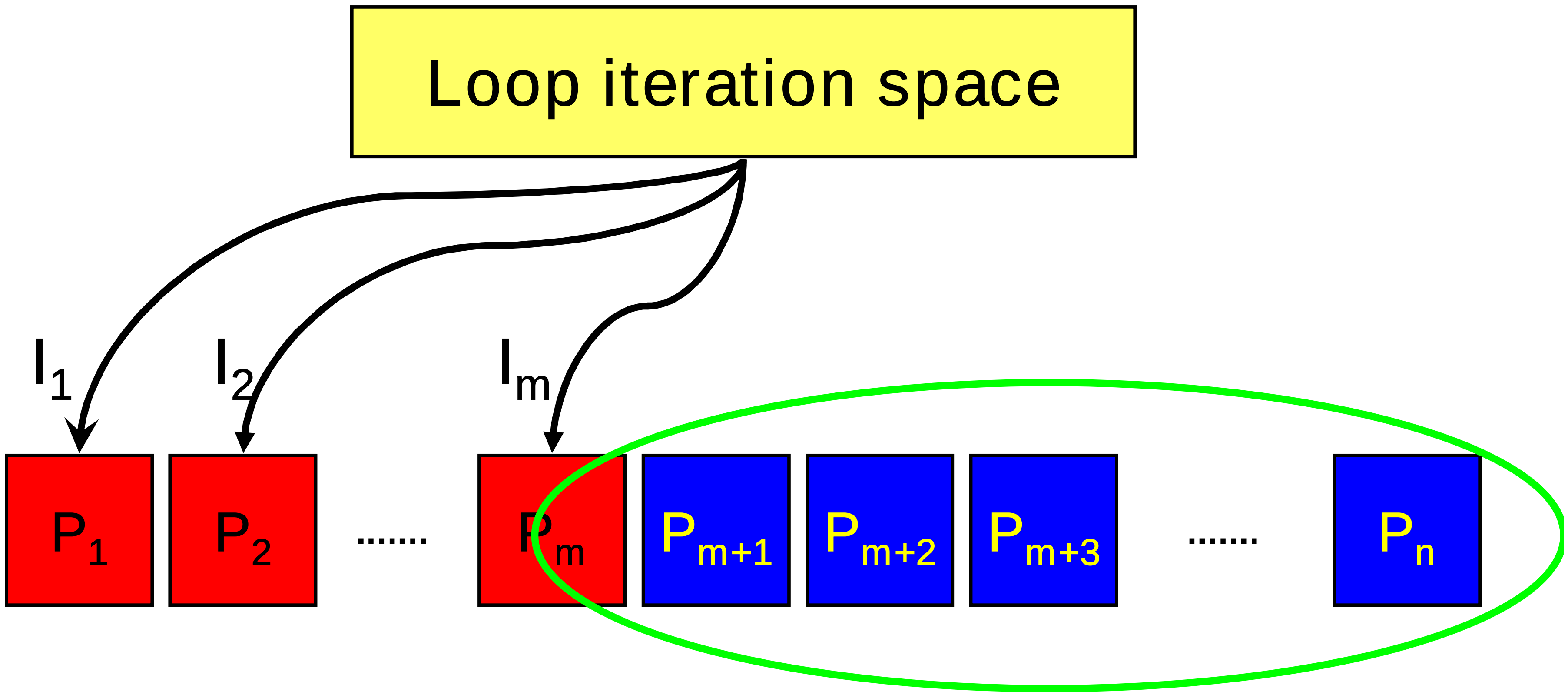
```
for i=1..1024
  for j=1..1024
     …
```



Processor Number for Optimum Performance

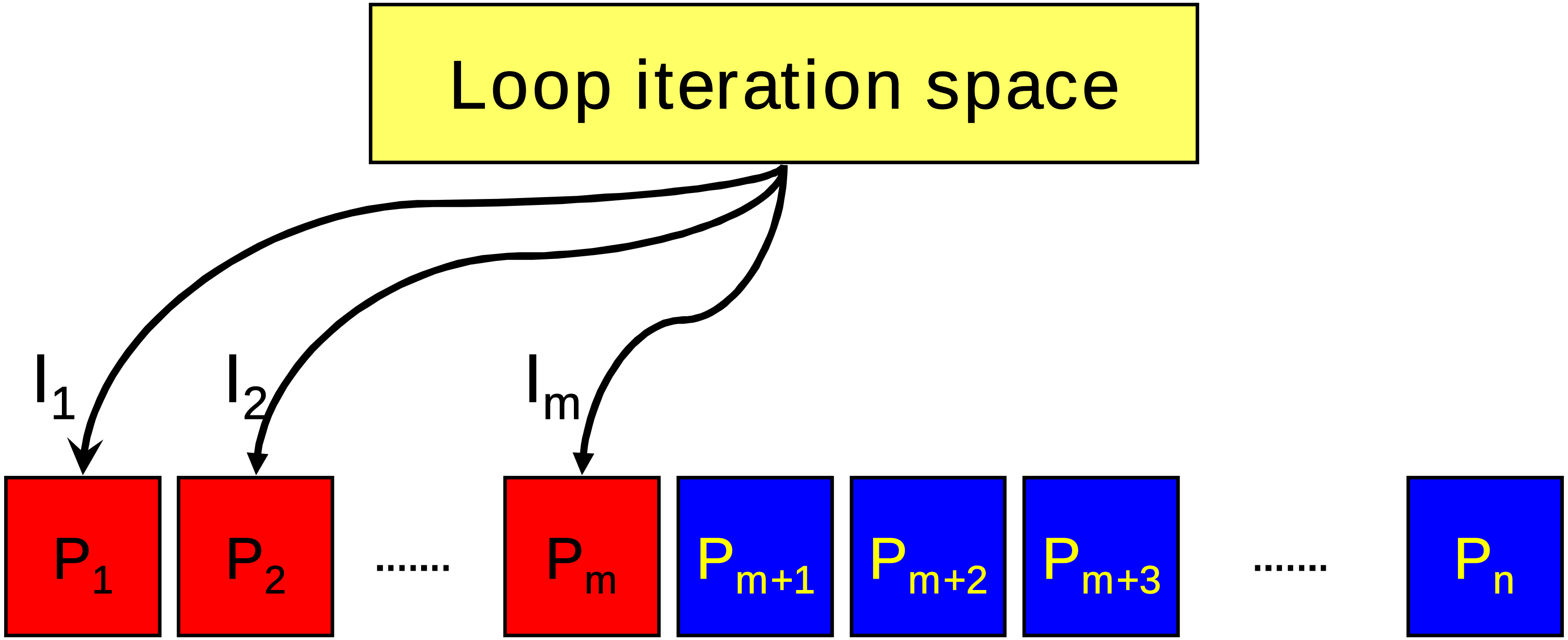
Benchmark	N1	N2	N3	N4	N5	N6	N7
3step-log	1	1	5				
adi	4	5					
btrix	2	1	7	6	1	3	8
eflux	2	3					
full-search	2	2	6				
n-real-updates	4	4	4				
tsf	1	7	2	4			

Adaptive Loop Parallelization



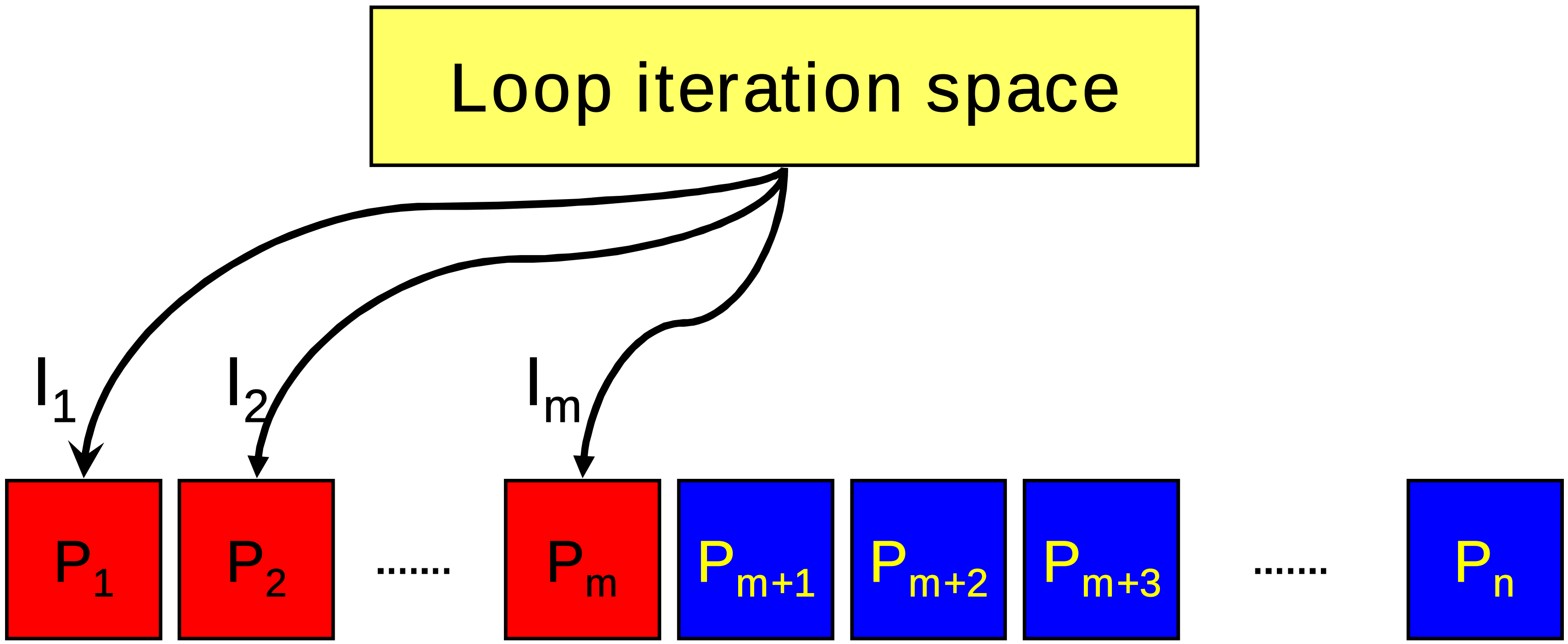
How to utilize idle processors?

Optimizing Power Consumption

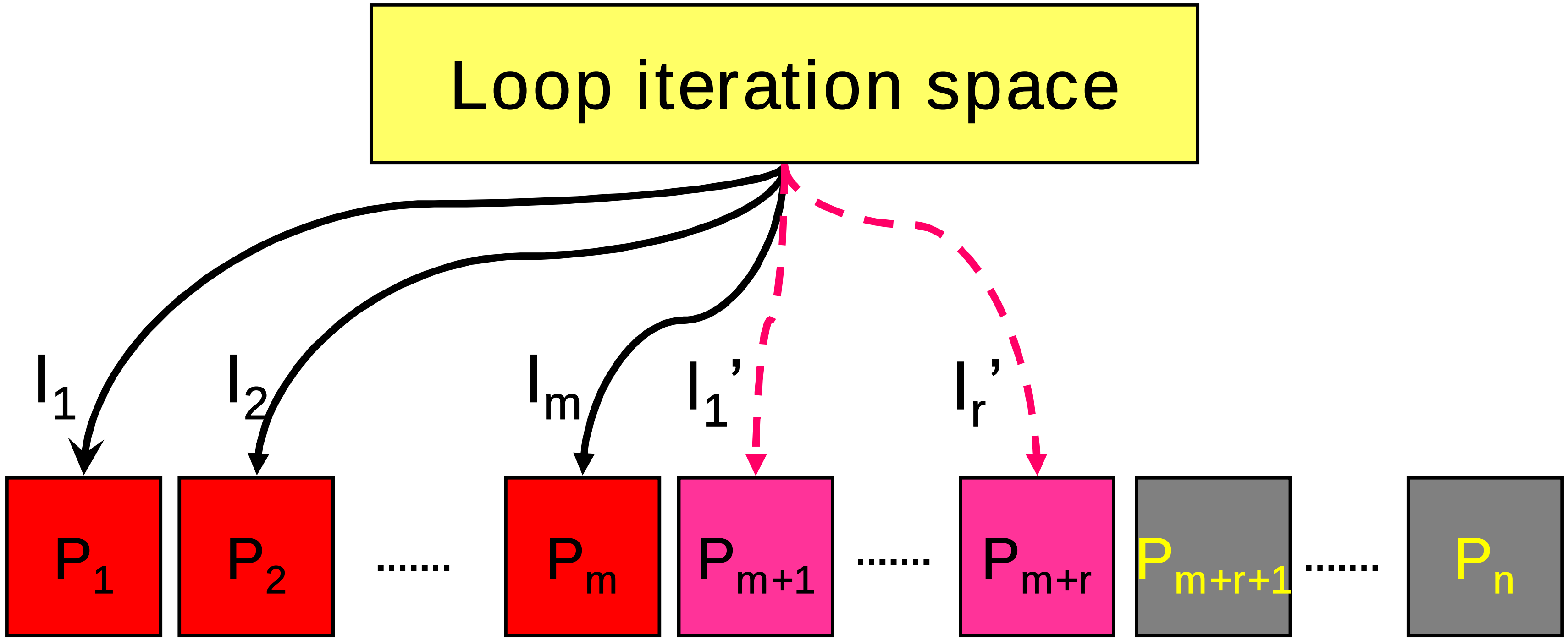


Idle processors are put into low-power mode to save leakage energy

Improving Reliability

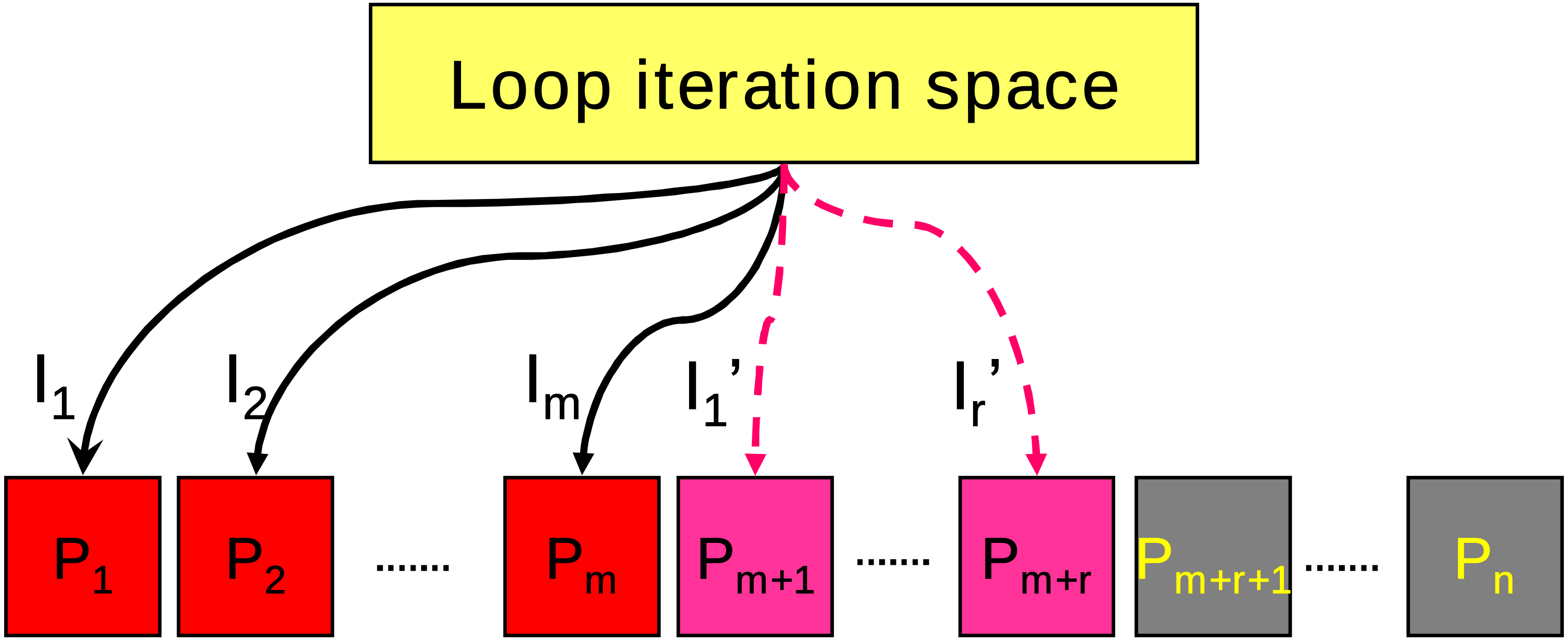


Improving Reliability



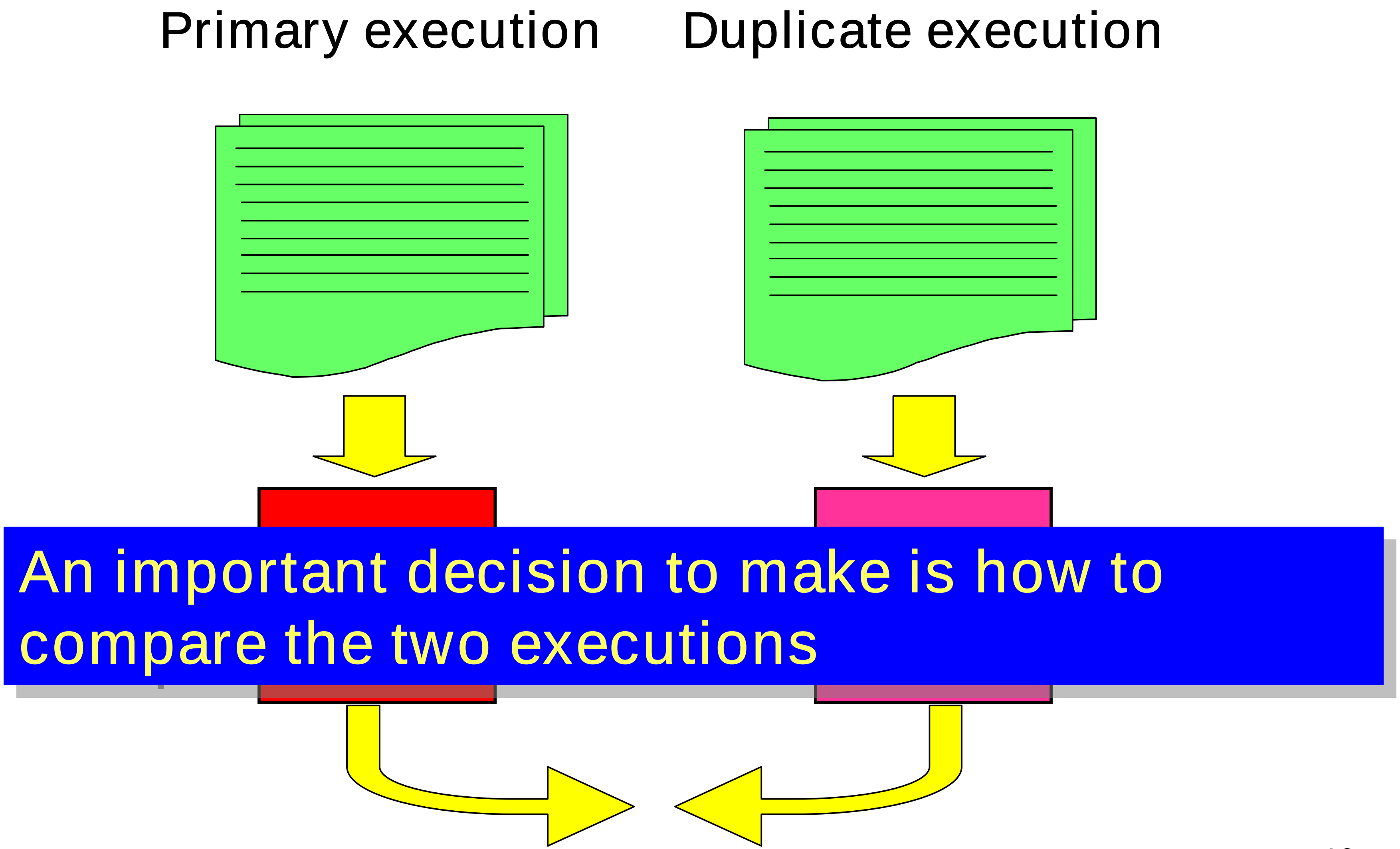
The local iteration space of r processors (P_1 through P_r) are duplicated and executed on processors P_{m+1} through P_{m+r}

Improving Reliability

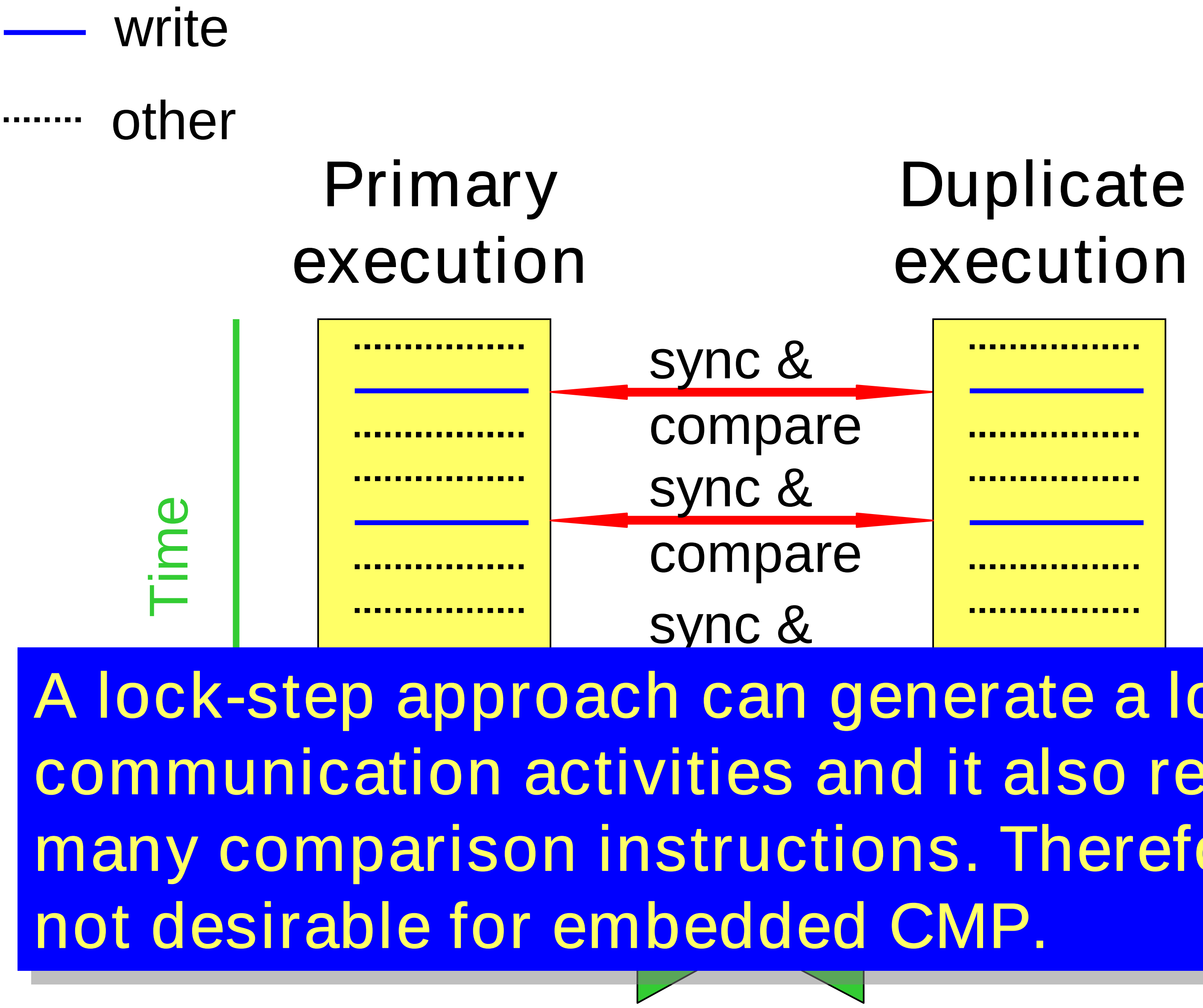


r/m is called the duplication percentage. Different duplication percentages represent different tradeoff points between performance, energy, and reliability

Computation Duplication

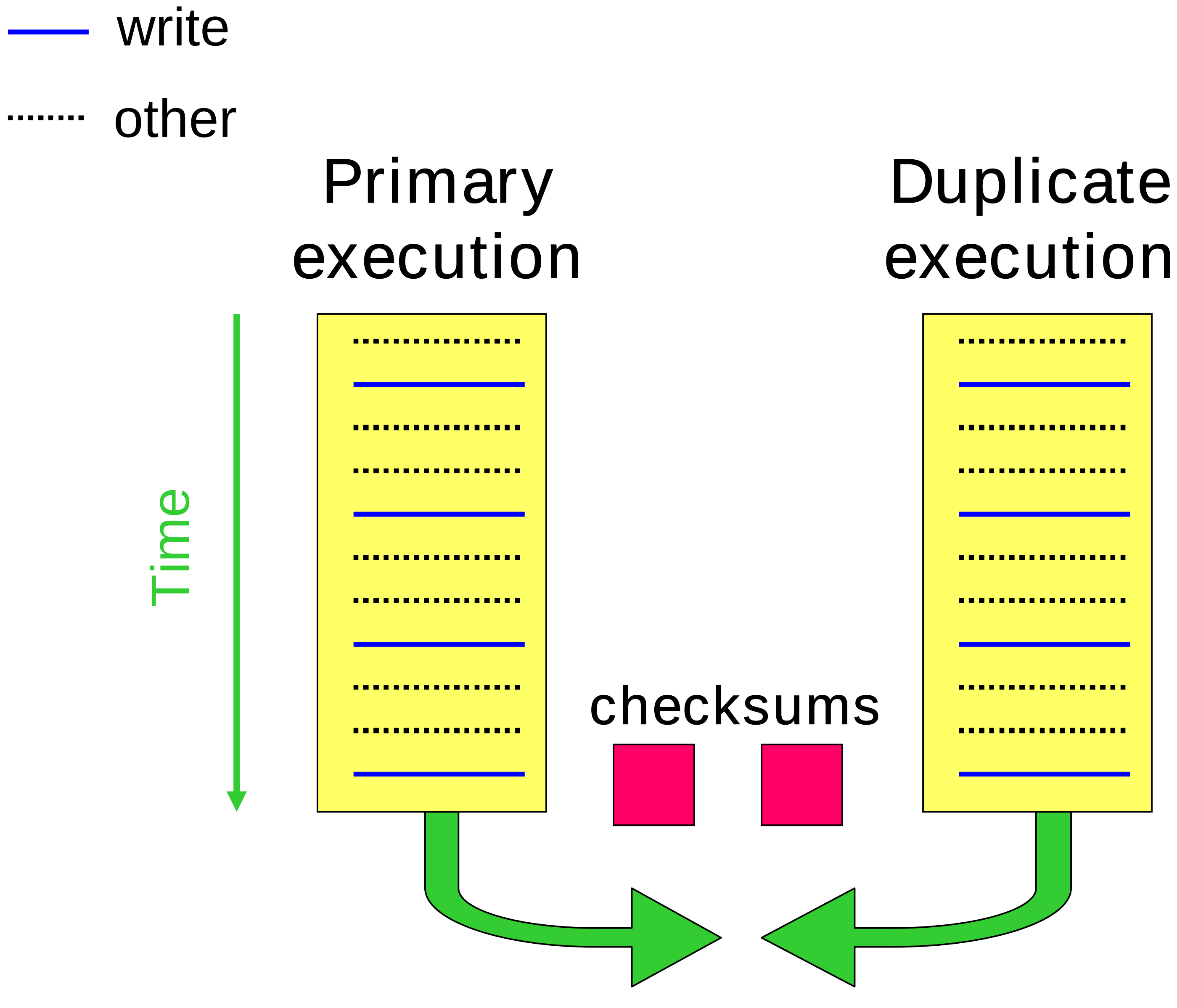


Lock-step Approach

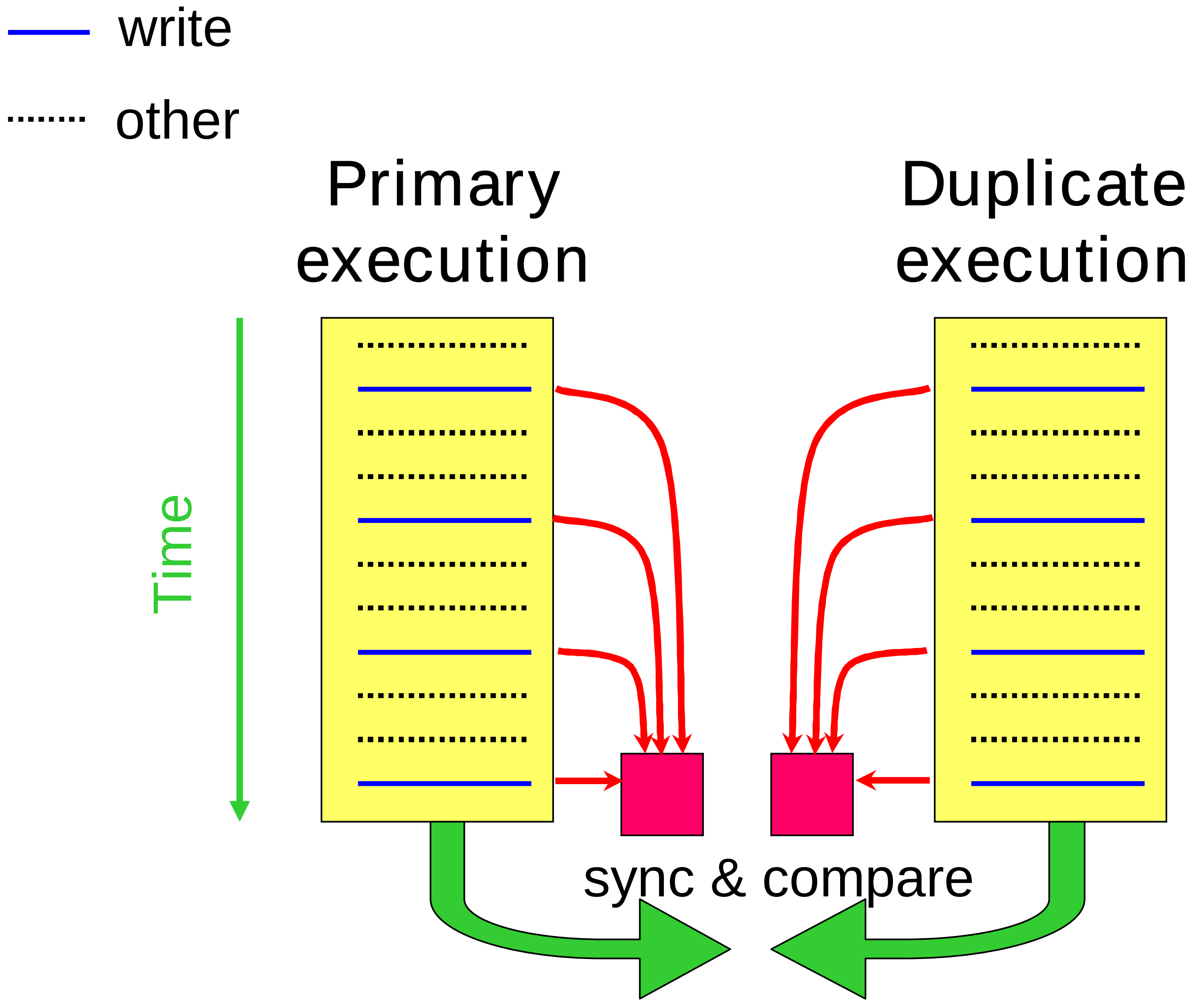


A lock-step approach can generate a lot of communication activities and it also requires many comparison instructions. Therefore, it is not desirable for embedded CMP.

Checksum-based Execution Comparison



Checksum-based Execution Comparison



Example Code

Original loop body

```
A[i]=C[i+1]*D[i]+E[i];
B[i]=C[i-1]-D[i-1];
```

Checksum for each processor

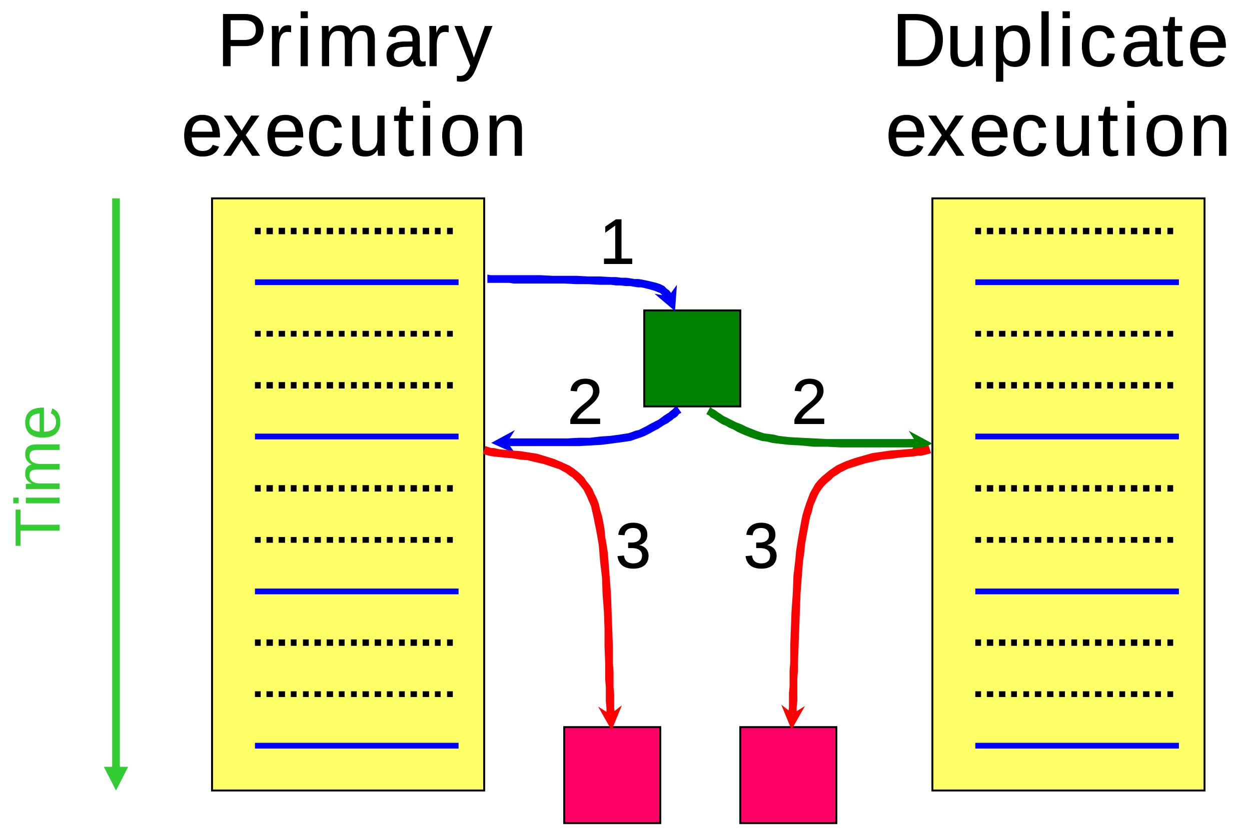
Primary execution

```
A[i]=C[i+1]*D[i]+E[i];
CHECK[prid]+=A[i];
B[i]=C[i-1]-D[i-1];
CHECK[prid]+=B[i];
```

Duplicated execution

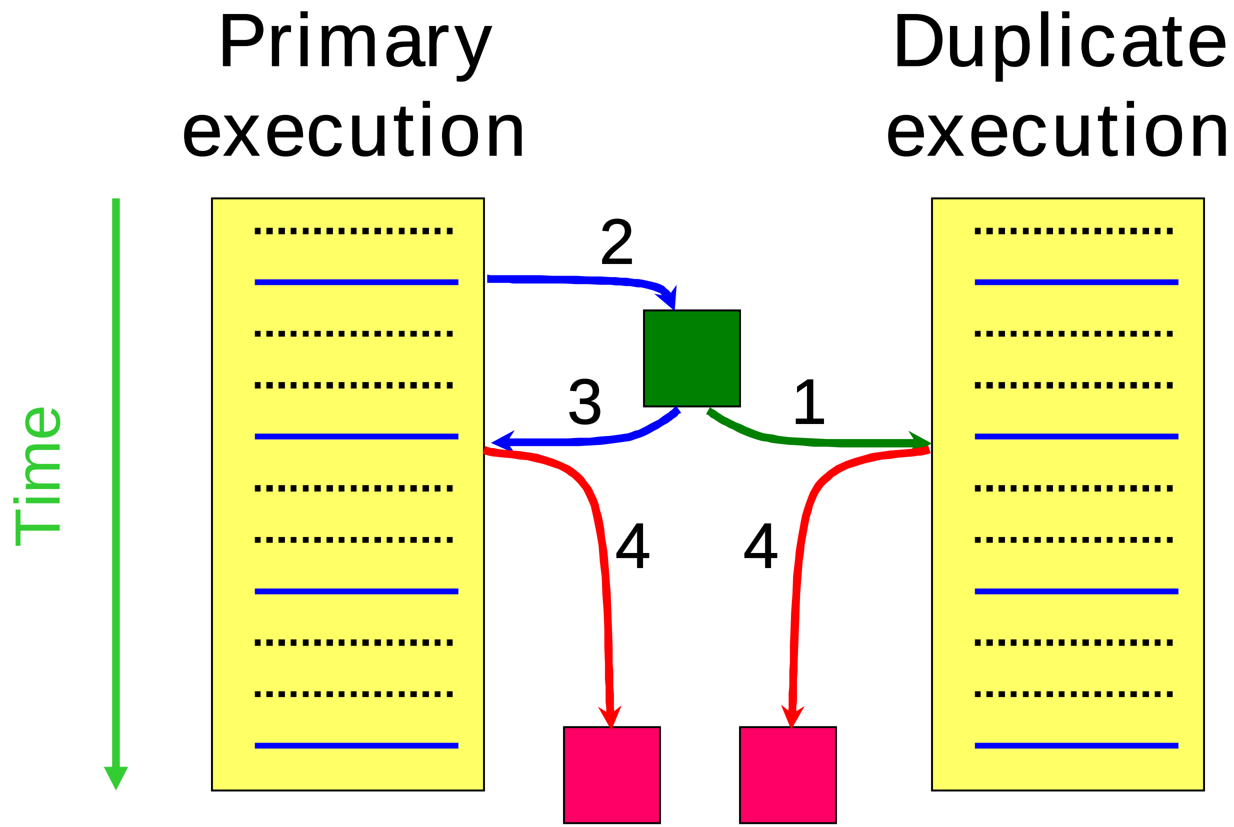
```
CHECK[prid]+=
     C[i+1]*D[i]+E[i];
CHECK[prid]+=
     C[i-1]-D[i-1];
```

Shared Data Problem



Desired execution order

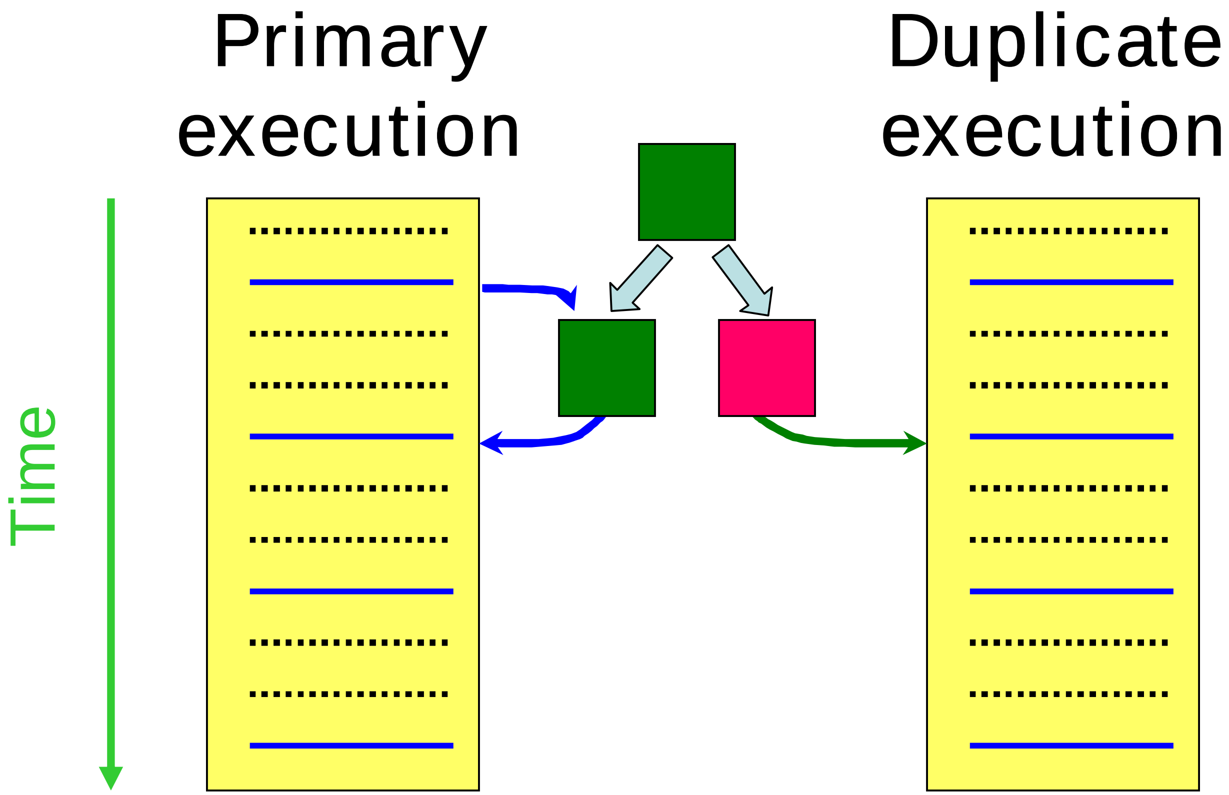
Shared Data Problem



Undesirable execution sequence

Race condition might happen if an array element is both read and written in the iteration space

Solution to Shared Data Problem



Example Code

Original loop body

```
A[i]=B[i]+C[i];
B[i]=C[i]-10;
```

```
A[i]=B[i]+C[i];
CHECK[prid]+=A[i];
B[i]=C[i]-10;
CHECK[prid]+=B[i];
```

Primary execution

```
CHECK[prid]+=
     B’[i]+C[i];
CHECK[prid]+=
     C[i]-10;
```

Duplicated execution

Experimental Setup

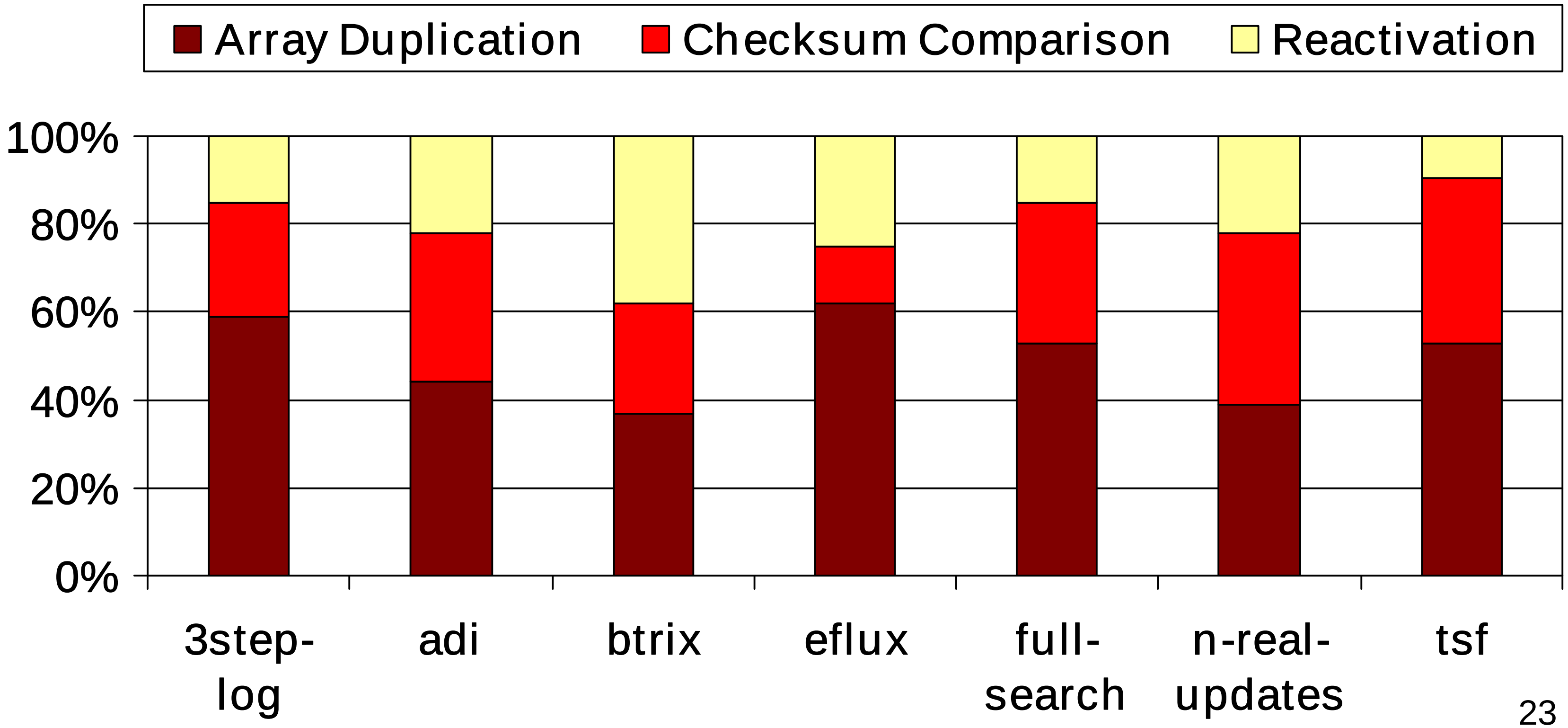
- Simics for CMP simulator
- 8 processors
- 8KB L1 I-cache, 8KB L1 D-cache, 1MB L2 cache
- Seven benchmarks from Perfect Club, Livermore, DSPStone

Energy-Delay-Fallibility (EDF) Product

- EDF = energy * (execution cycles) / Fallibility
  - Fallibility = 1/reliability
  - Reliability is the percentage of primary processors that have duplicates
- EDF is a good metric for evaluating the tradeoffs between energy, performance, and reliability
  - We want EDF to be as small as possible

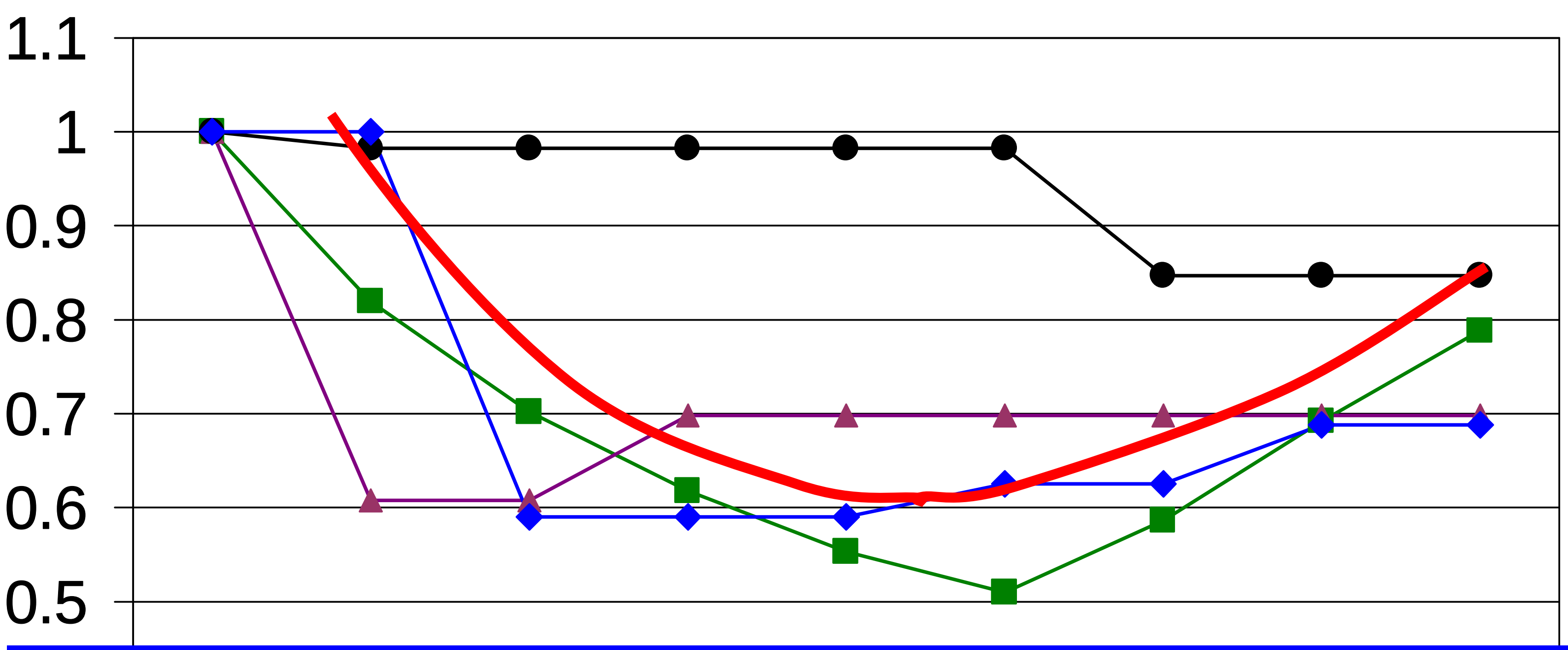
Performance Overhead

- Less than 2% overhead when averaged over all the benchmark codes
- Performance overhead breakdown



EDF with Different Percentage Usage of Idle

As percentage of duplicates increases, EDF increases since reliability increases.



As we use more processors for duplication, the benefits coming from increased reliability can be offset by increased energy consumption.

Conclusion

- On-chip parallelism of CMP can be used for improving reliability
- Single metric based compilation strategies are not sufficient for current embedded systems, where multiple constraints are important
- EDF can be used for evaluate the tradeoffs of power, performance, and reliability