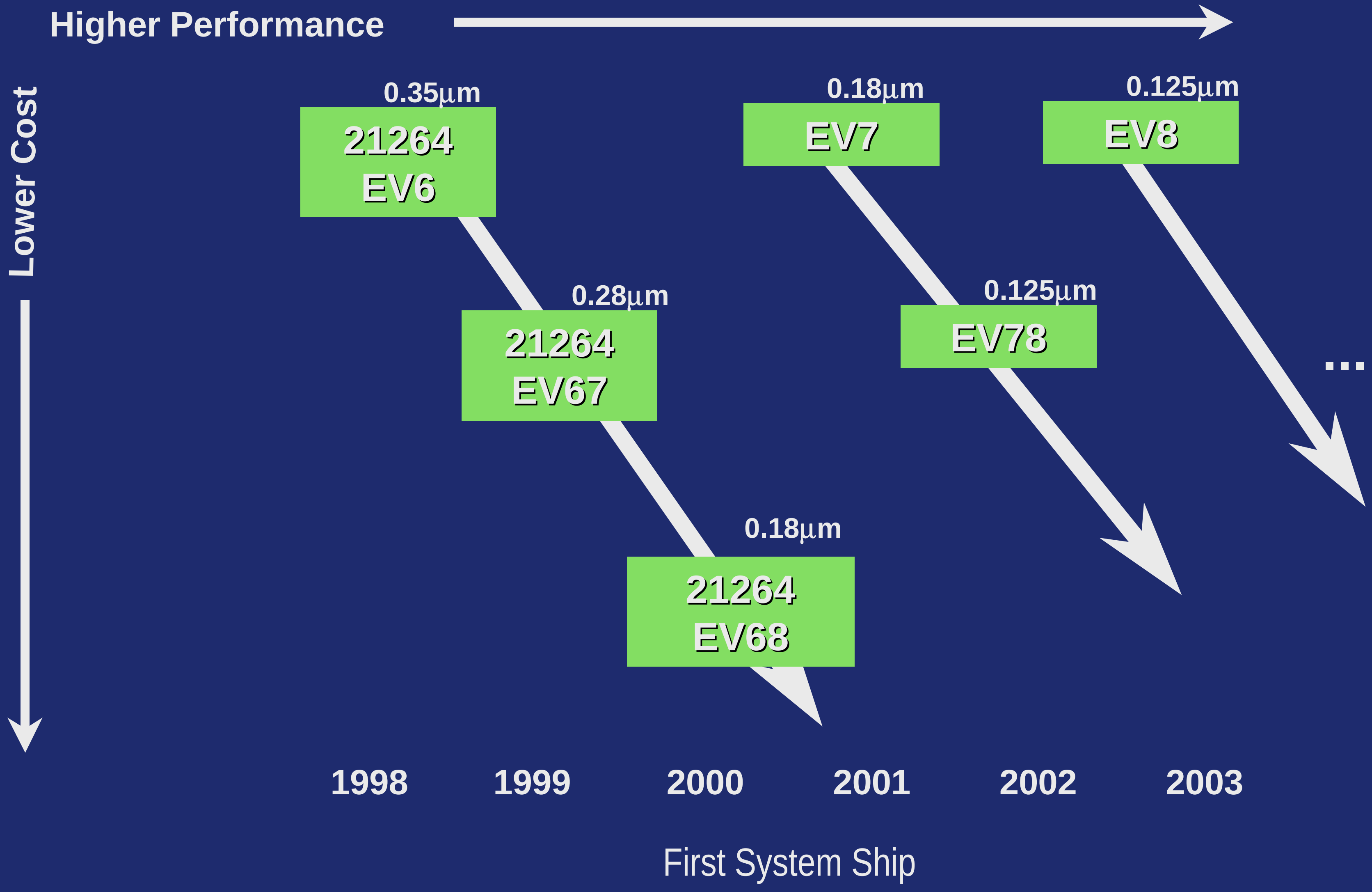
EV8: The Post-Ultimate Alpha

Dr. Joel Emer
Intel Fellow
Intel Architecture Group
Intel Corporation

Alpha Microprocessor Overview



Goals

- Leadership single stream performance

- Extra multistream performance with multithreading
  - Without major architectural changes
  - Without significant additional cost

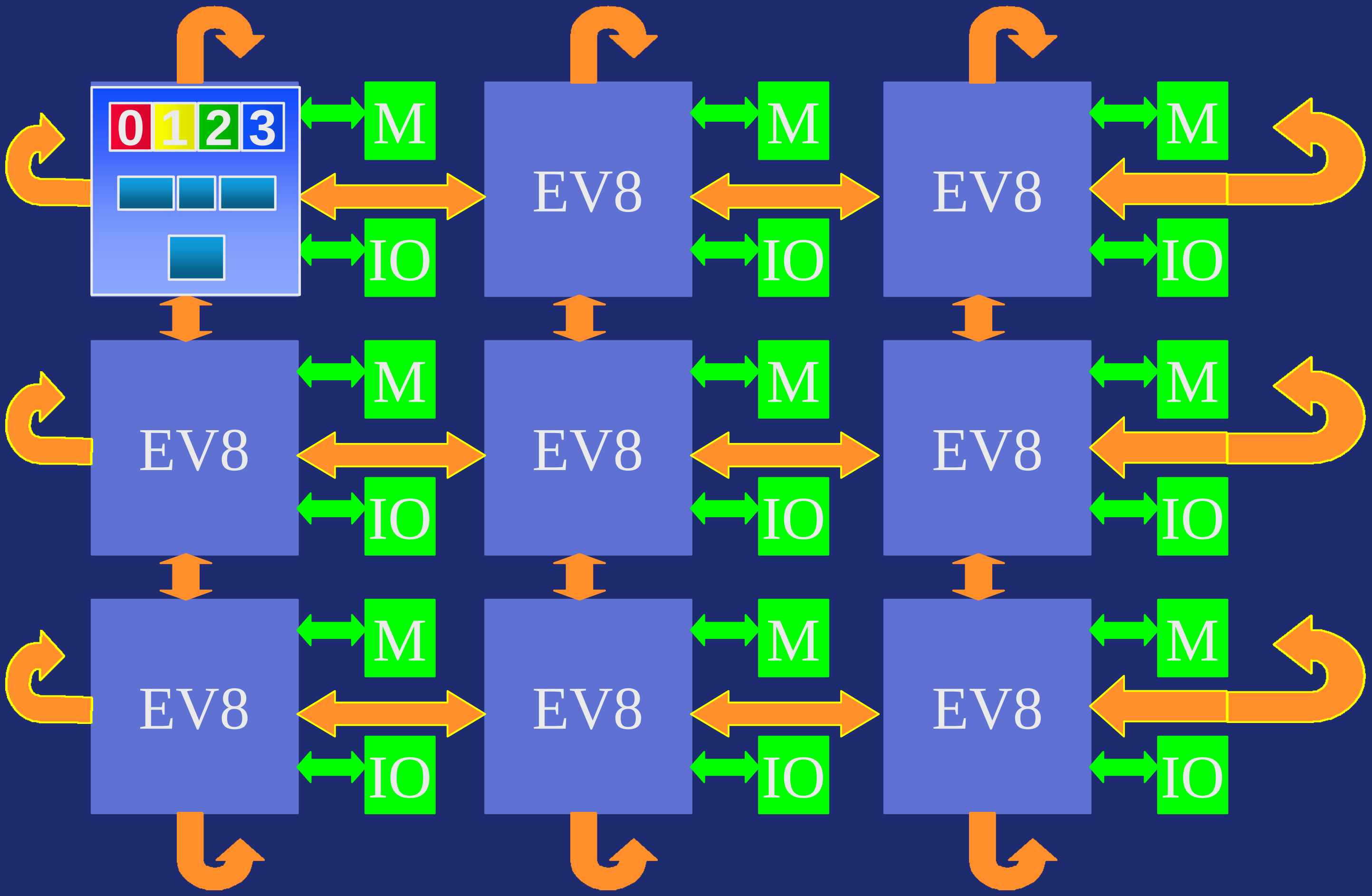
EV8 Architecture Overview

- Aggressive instruction fetch unit
- 8-wide super-scalar execution unit

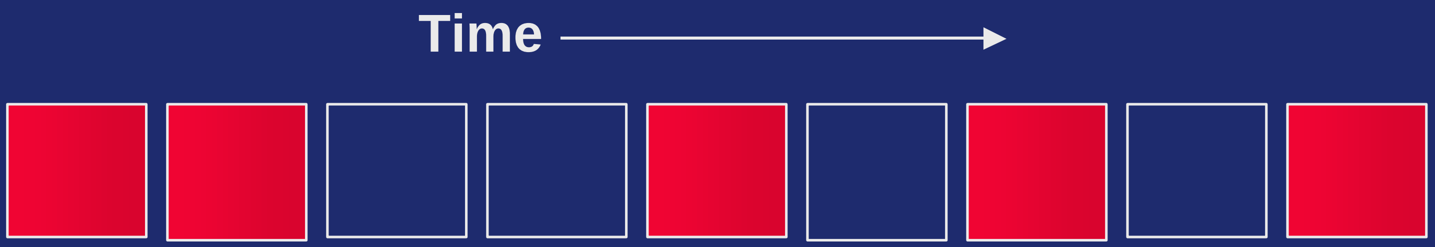
- 4-way simultaneous multithreading (SMT)

- Large on-chip L2 cache
- Direct RAMBUS interface
- On-chip router for system interconnect
  - for glueless, directory-based, ccNUMA
  - with up to 512-way multiprocessing

System Block Diagram

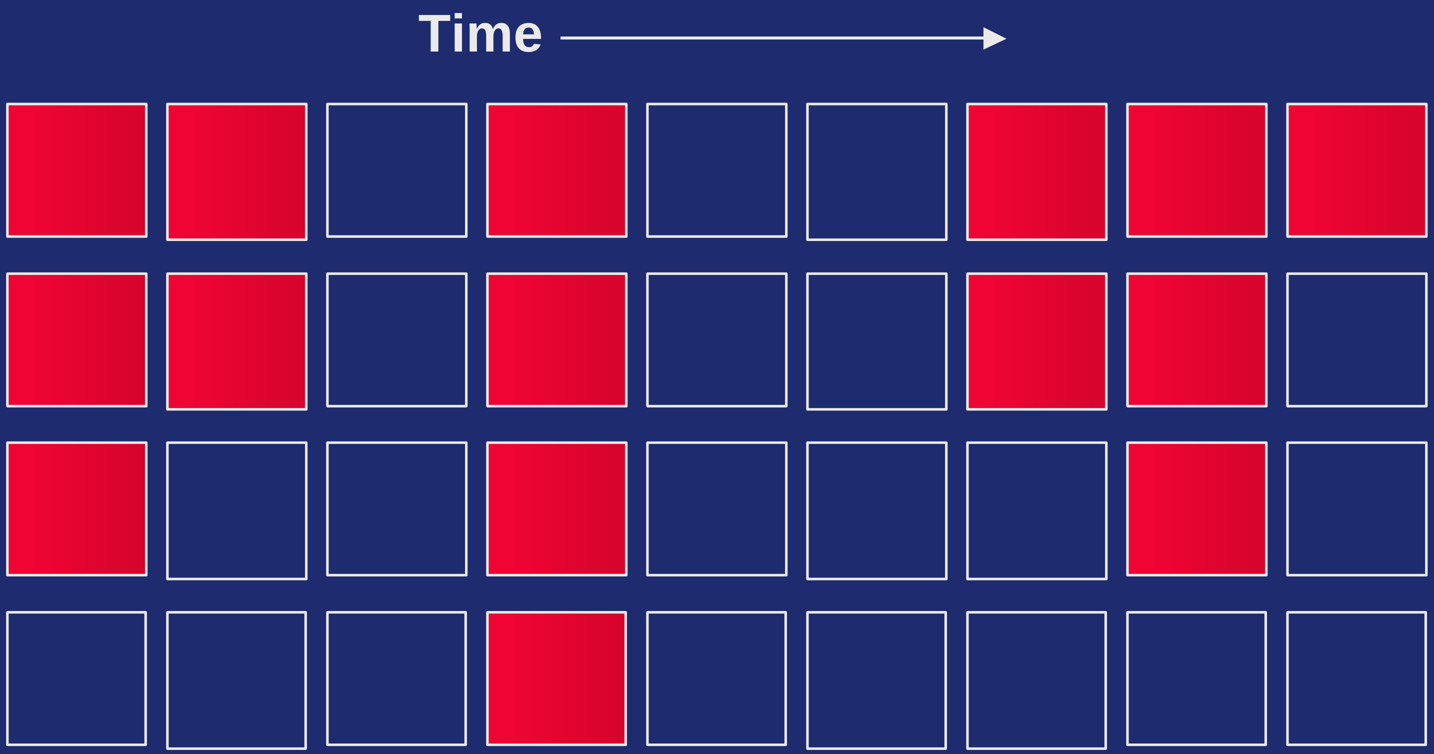


Instruction Issue



Reduced function unit utilization due to dependencies

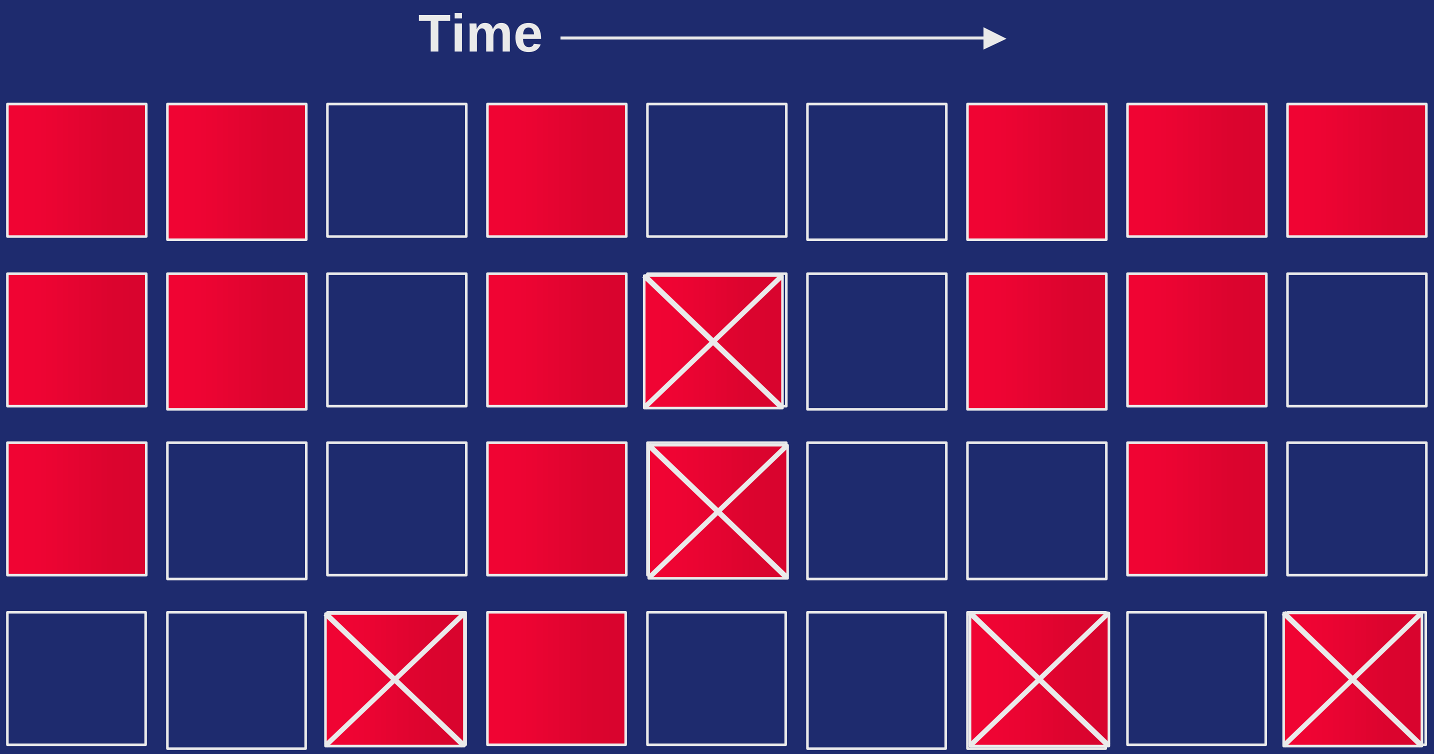
Superscalar Issue



Time

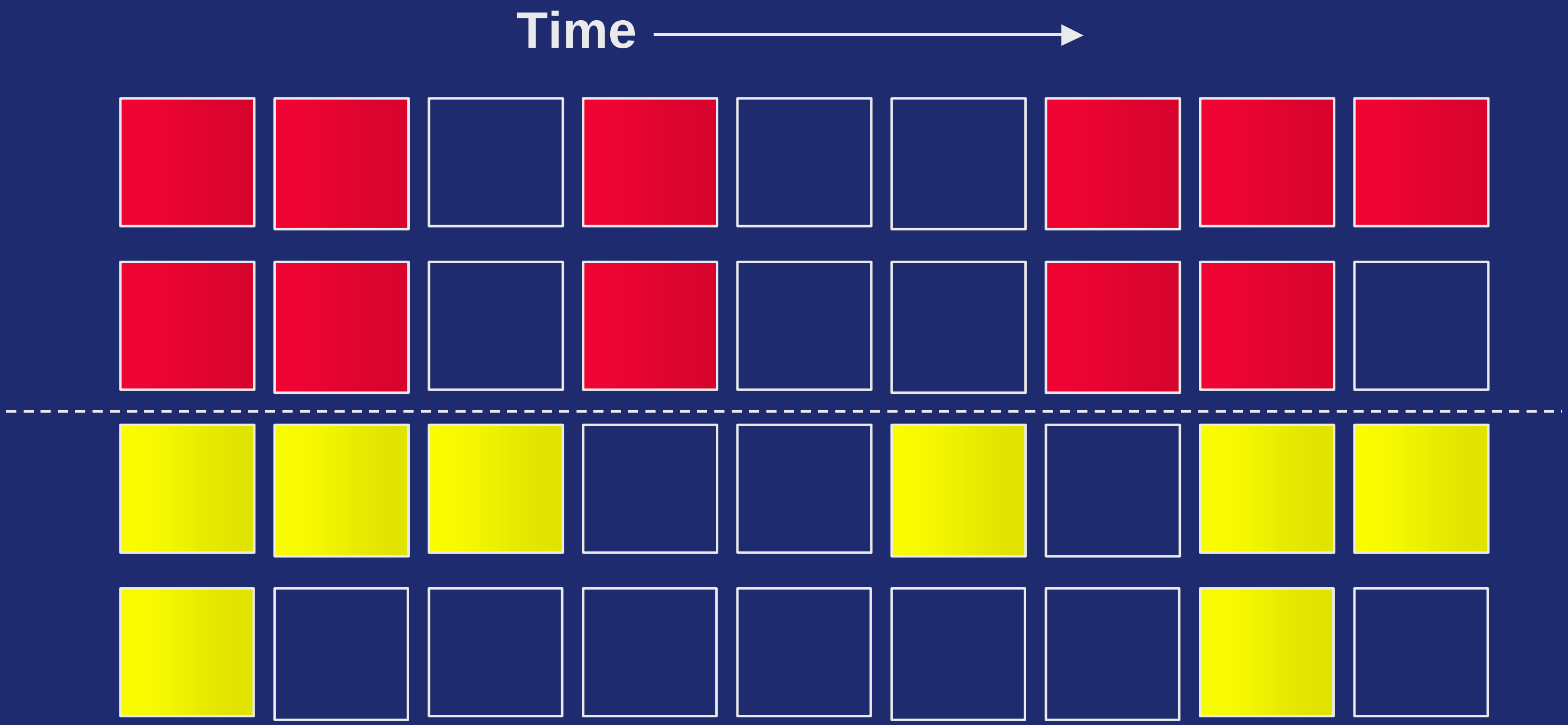
Superscalar leads to more performance, but lower utilization

Predicated Issue



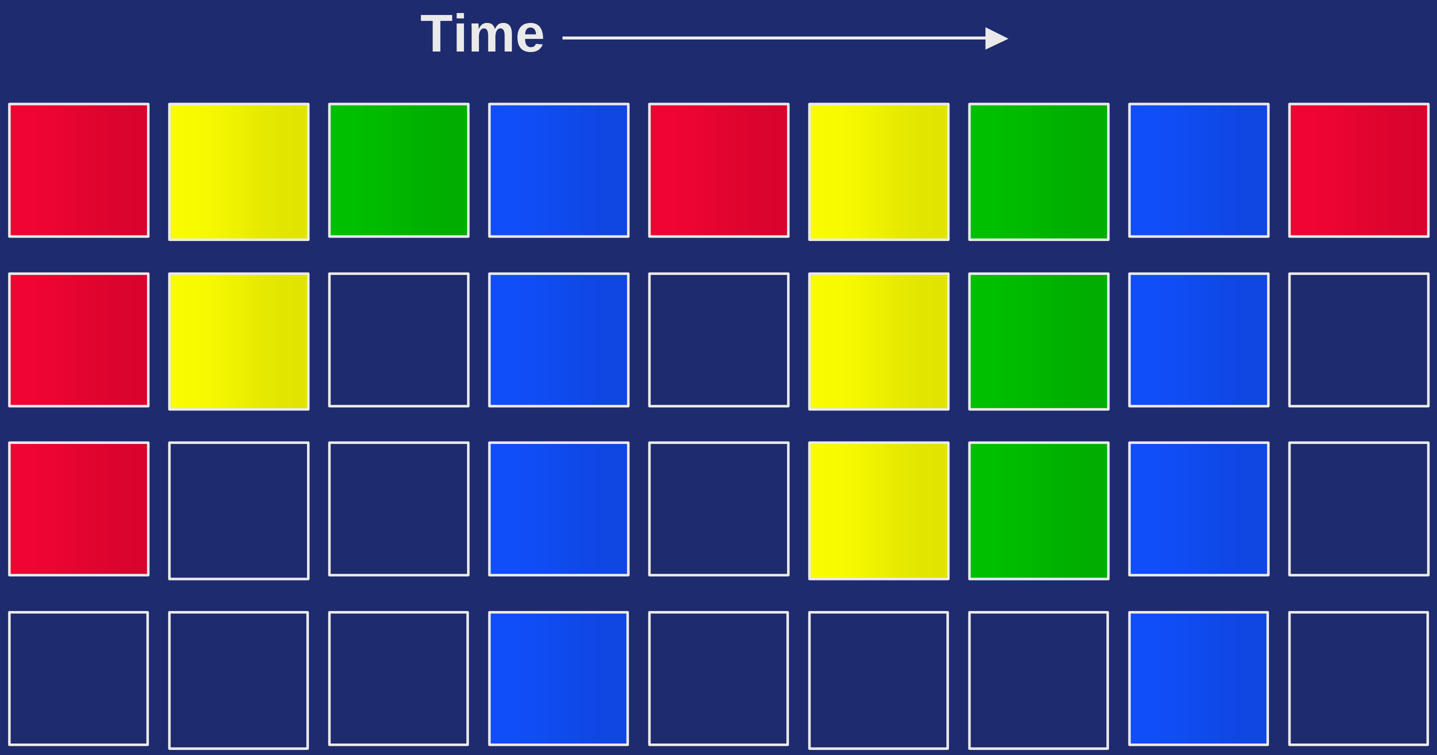
Adds to function unit utilization, but results are thrown away

Chip Multiprocessor



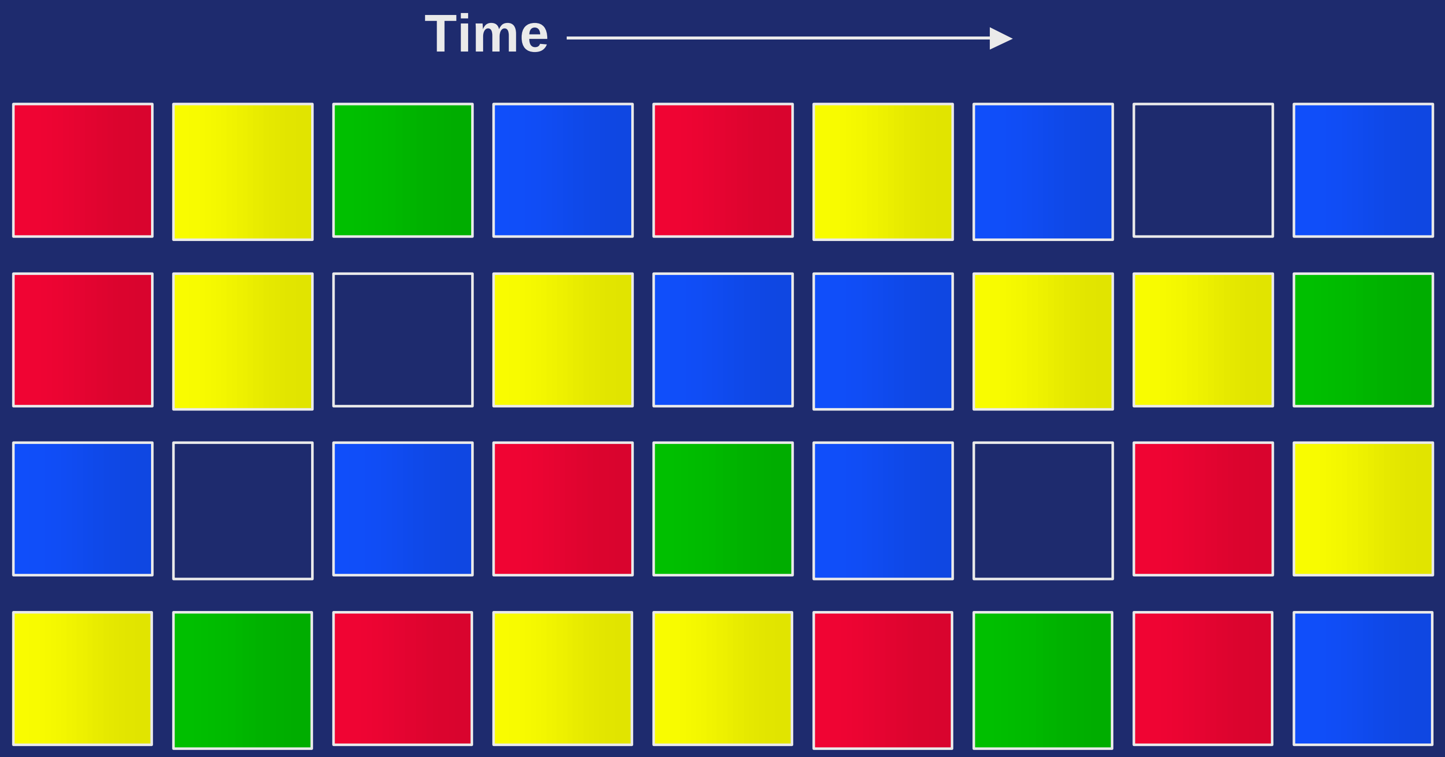
Limited utilization when only running one thread

Fine Grained Multithreading



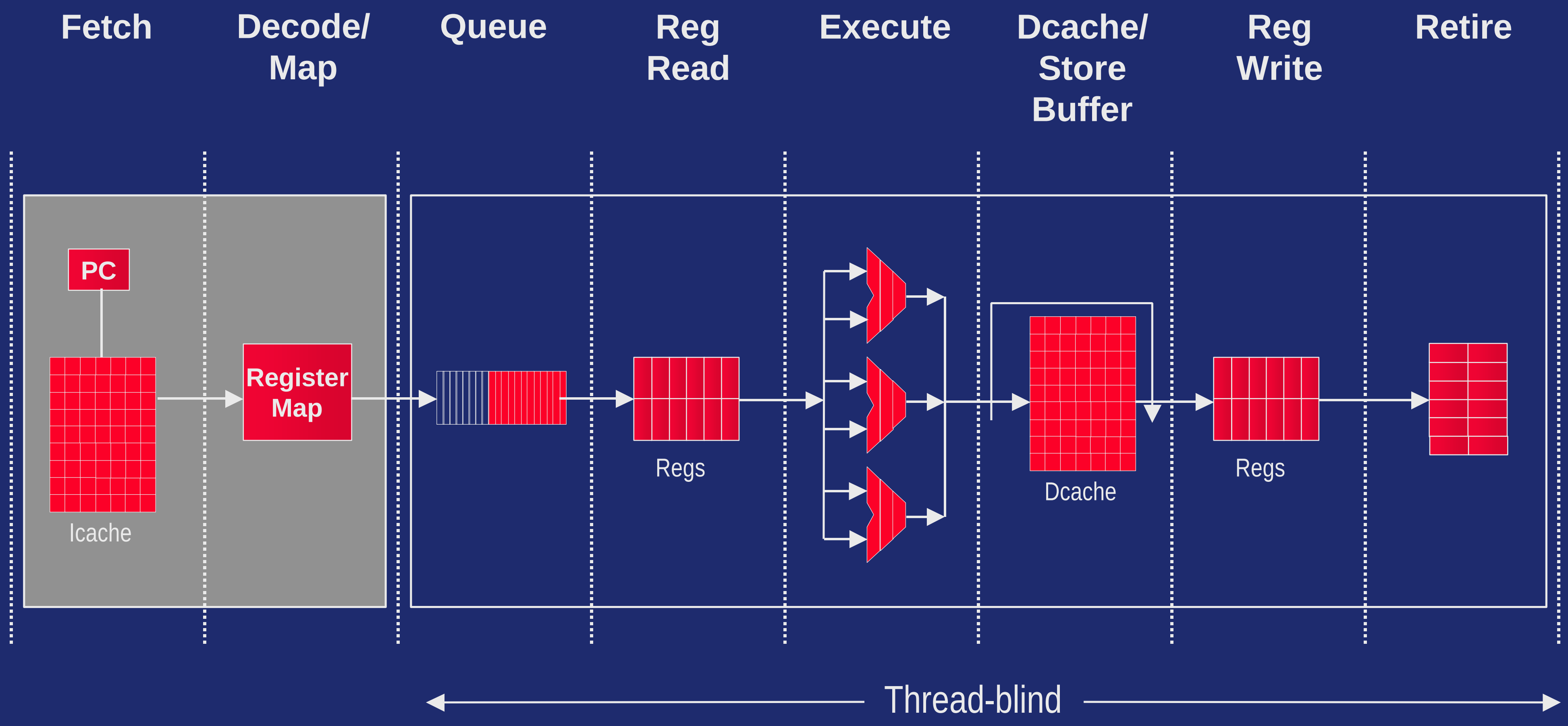
Intra-thread dependencies still limit performance

Simultaneous Multithreading



Maximum utilization of function units by independent operations

Basic Out-of-order Pipeline



SMT Pipeline

Fetch

Decode/ Map

Queue

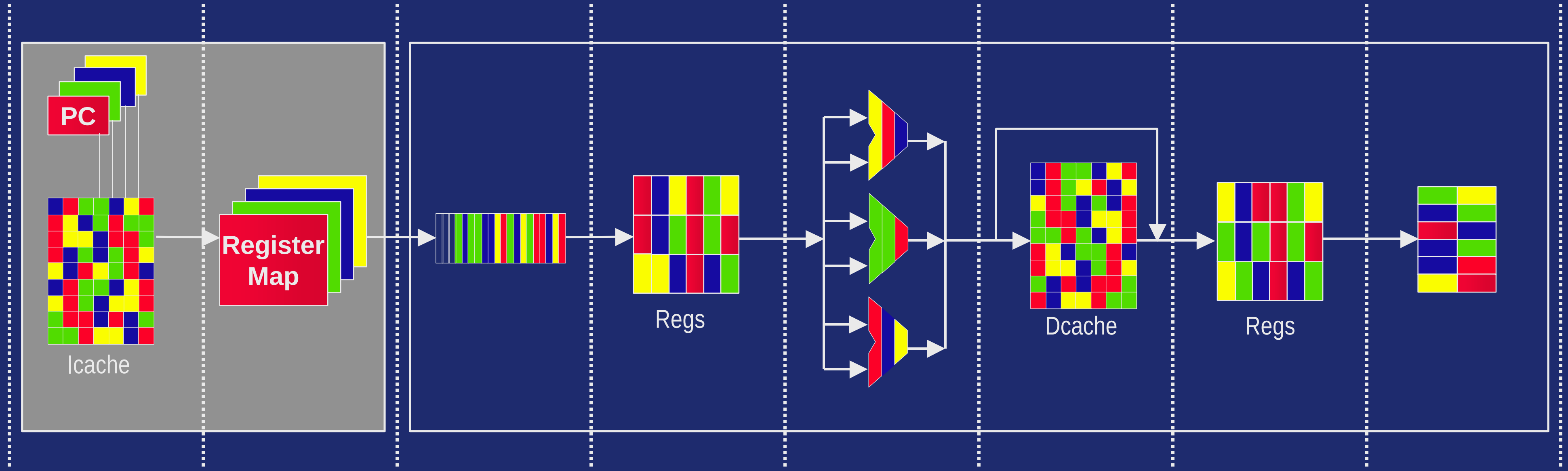
Reg Read

Execute

Dcache/ Store Buffer

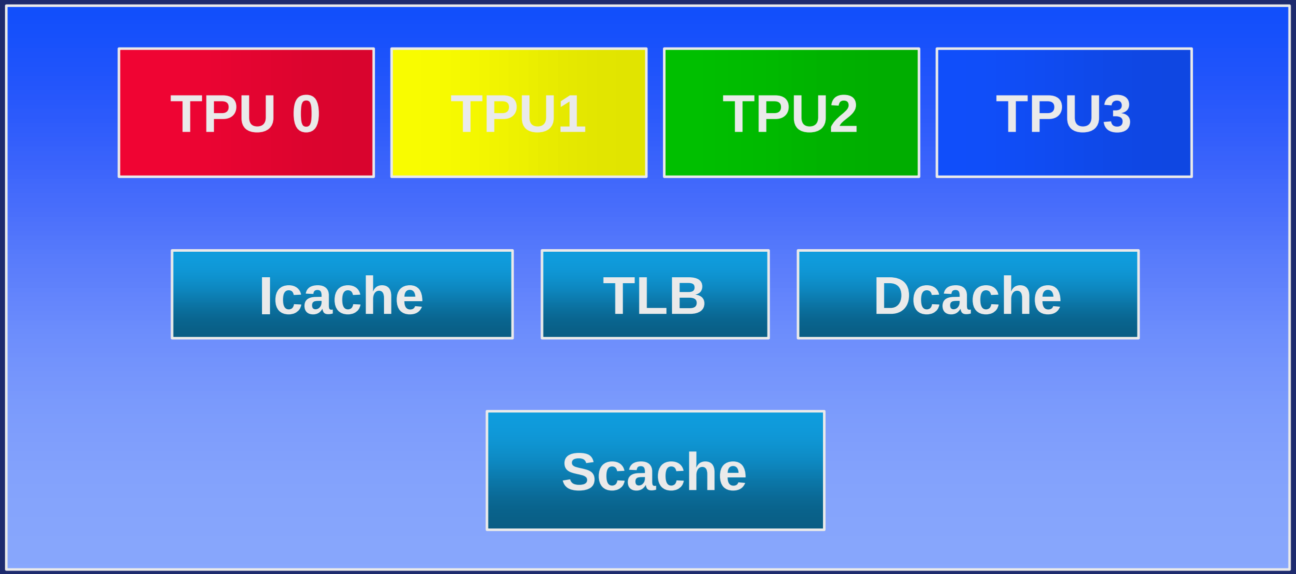
Reg Write

Retire



Architectural Abstraction

- 1 CPU with 4 Thread Processing Units (TPUs)
- Shared hardware resources

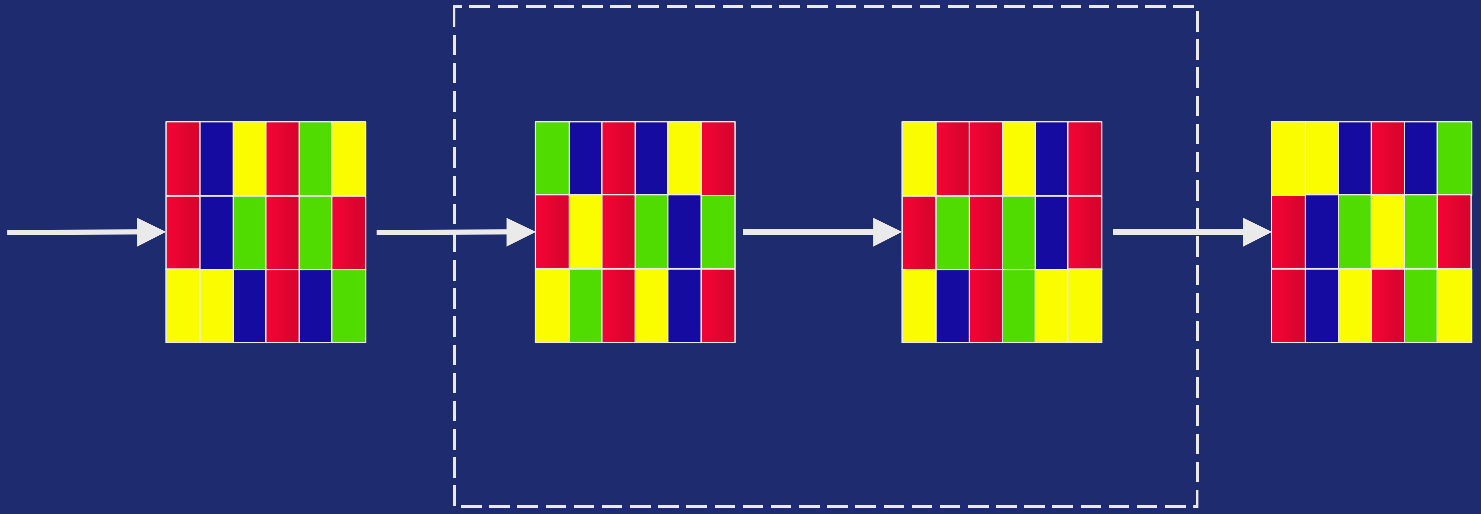


Key Design Principles

- High throughput single stream design
- Enhancements for SMT

Little’s Law

$$\text{Throughput (T)} = \frac{\text{Average Number of Tasks in Region (N)}}{\text{Average Latency in Region (L)}}$$



Little’s Law for Instruction Fetch

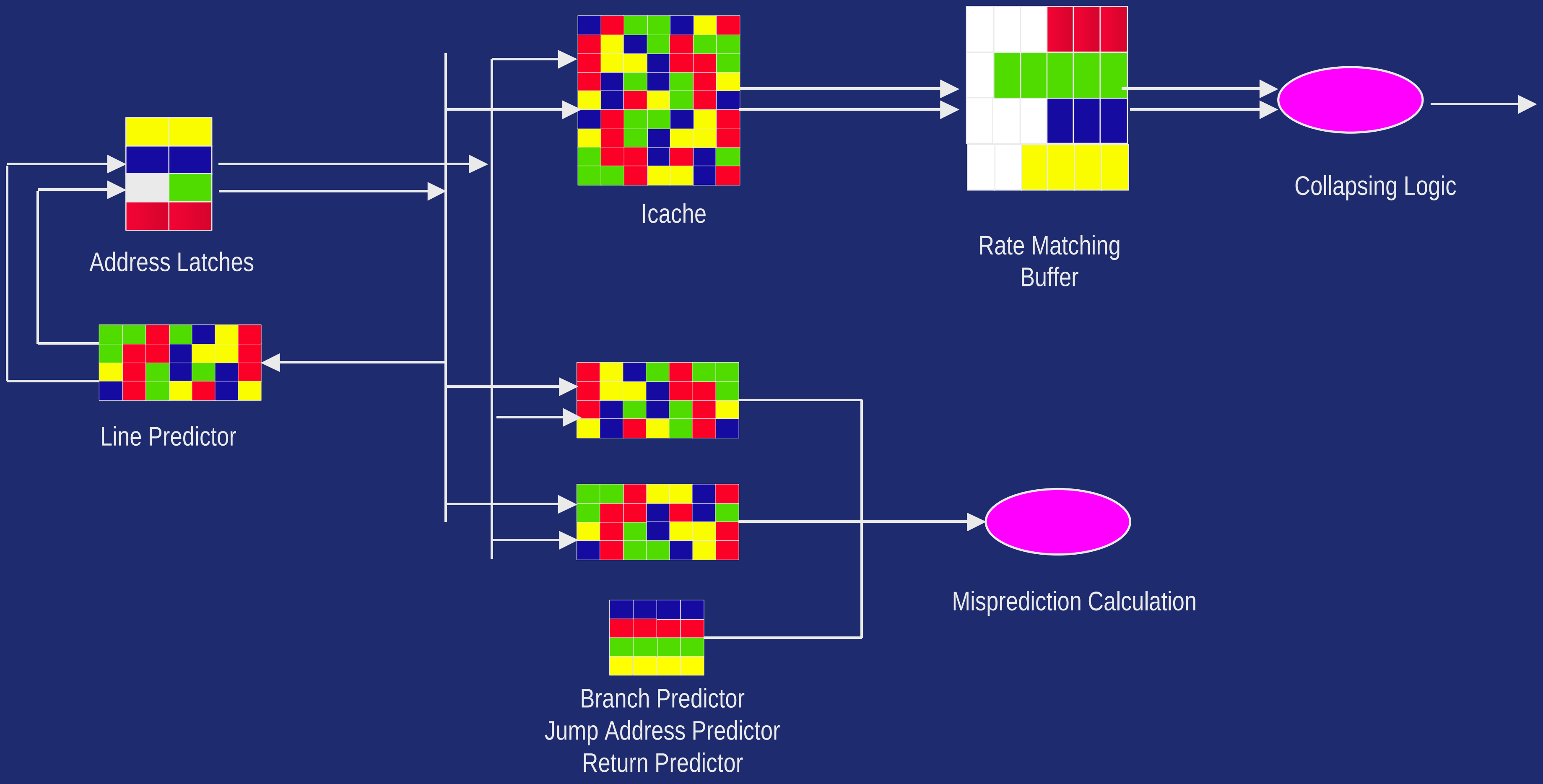
- L = fixed pipe length + average memory latency
- N = number of instructions fetched

$$T \uparrow = \frac{N \uparrow}{L}$$

Instruction Fetch Unit

- Wider fetch
  - Fetch more statically consecutive instructions
  - Limited by “trace” length
- Trace Cache
  - Build sequences of dynamically consecutive instructions
  - Significantly greater complexity
- Double fetch
  - Fetch two non-consecutive blocks of instructions

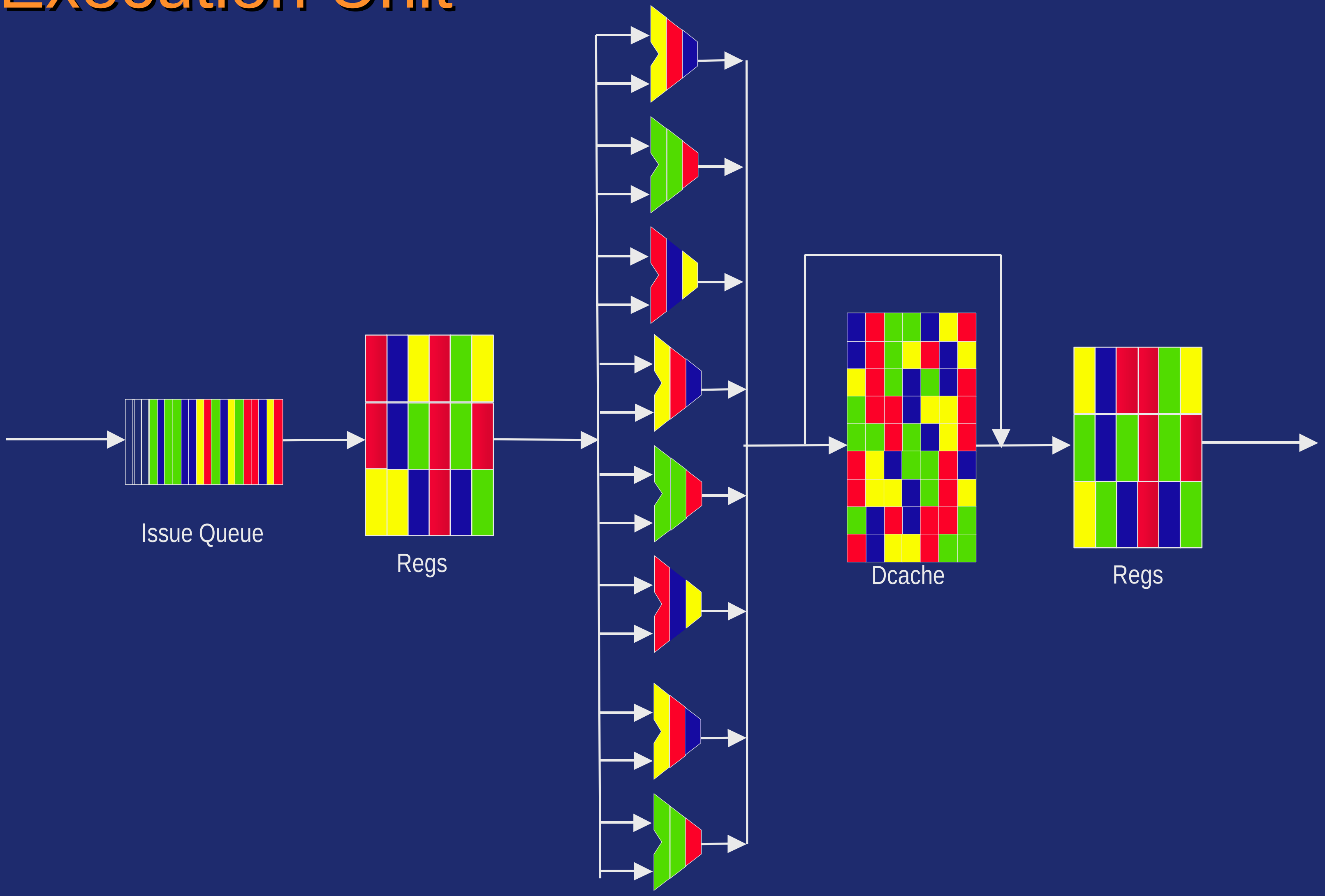
Instruction Fetch Unit



Instruction Fetch Characteristics

- Two 8-instruction fetches per cycle
- 16 branch predictions per cycle
- Jump target prediction
- Return address prediction
- Rate matching buffer of fetched instructions
- Collapse fetched instructions into groups of 8

Execution Unit



Execution Unit Characteristics

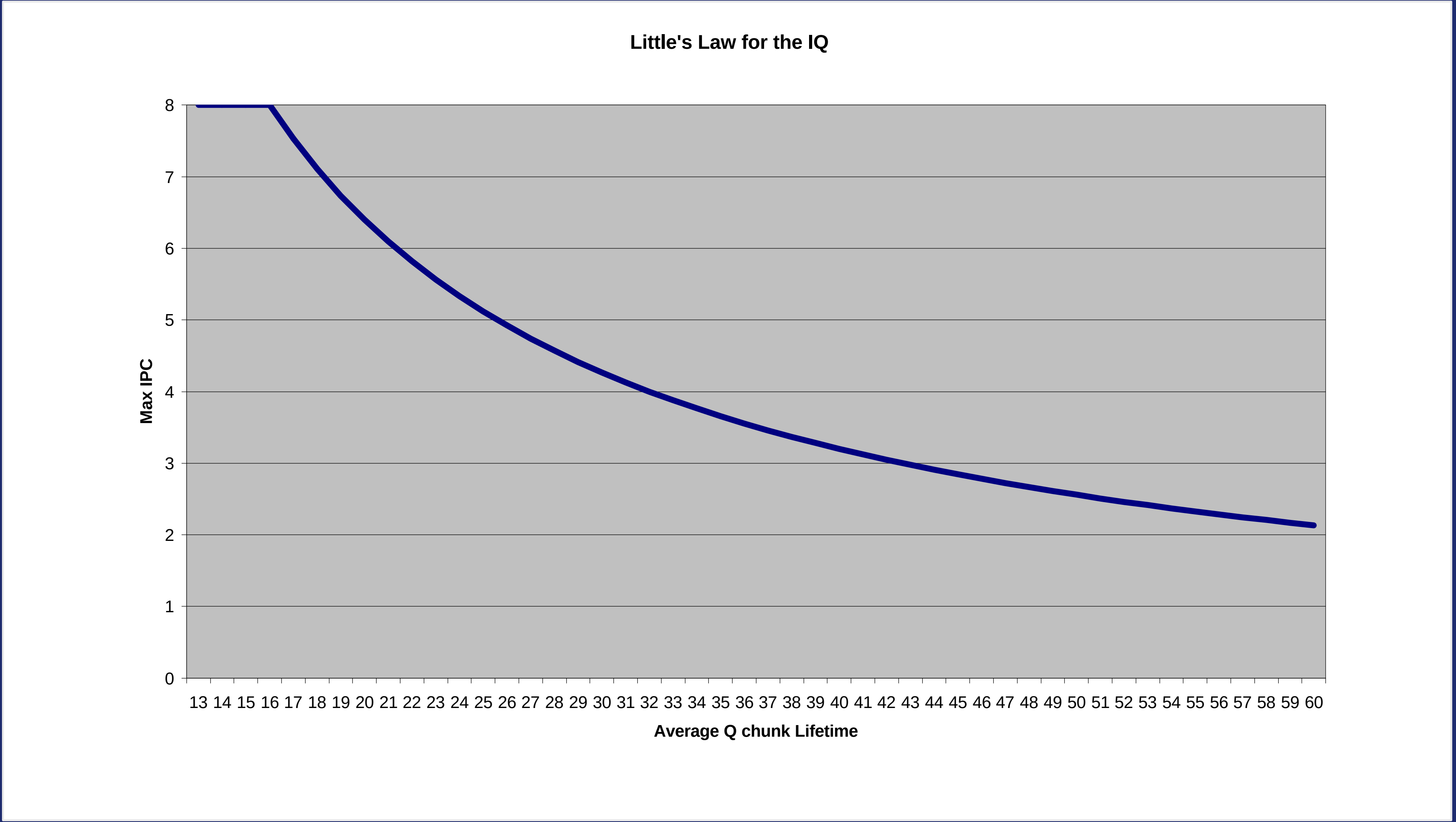
- Single issue queue
  - 8-wide
  - 112+ entries
- Register file
  - 512 registers
  - 16 read/8 write ports
- Function units
  - 8 integer ALUs
  - 4 floating ALUs
  - 4 memory operations (2 read/2 write)

Little’s Law for Execution Unit

- L (min) = Number of cycles in pipe
- T (desired) = Number of desired instructions per cycle (8)

$$8 = \frac{N}{13}$$

Little’s Law for Execution Unit



Key Design Principles

- High throughput single stream design
- Enhancements for SMT

Additions for SMT

- Replication required resources
  - Program counters
  - Register File (architectural space)
  - Register maps
  - …
- Sharable resources
  - Register file (rename space)
  - Instruction queue
  - Branch predictor
  - First and second level caches
  - Translation buffers
  - ….

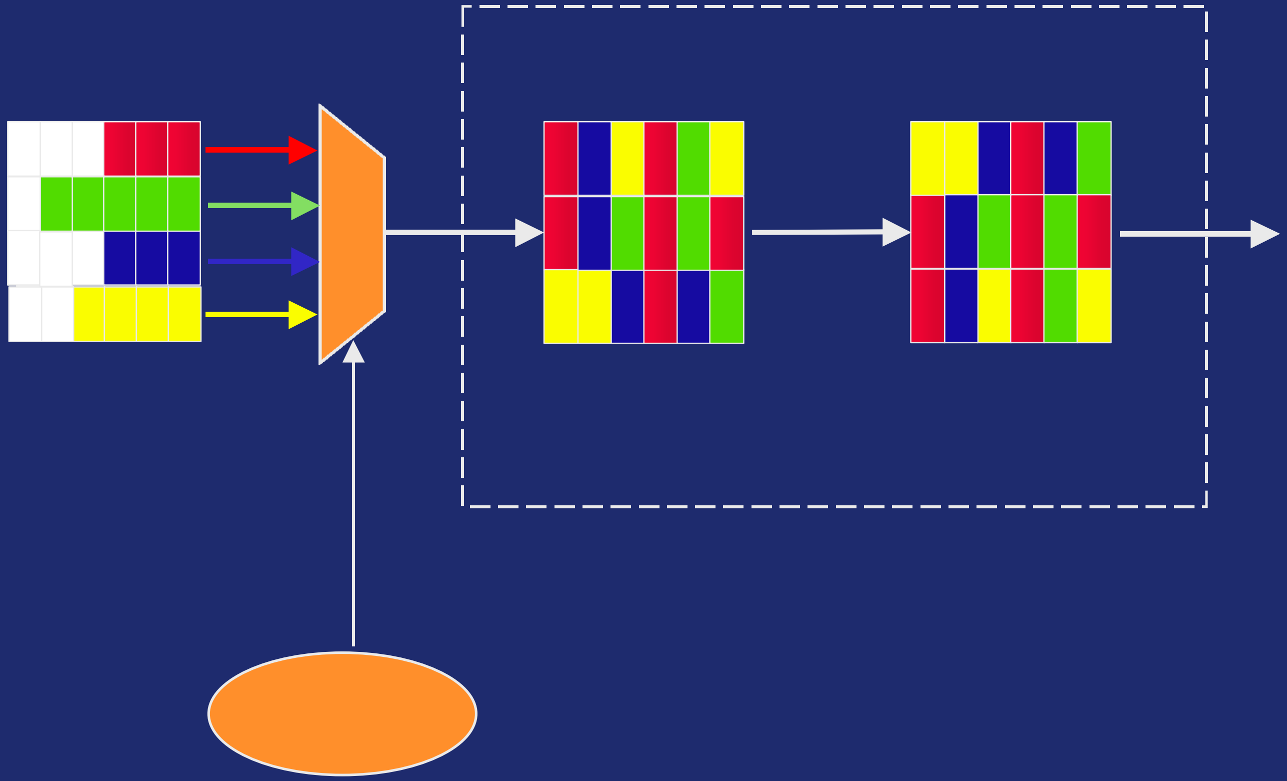
Approaches

- Replicated resources used for…
  - all per TPU state (except register file)
  - some sharable resources where design is easier (*)
    - E.g,, return stack predictor

- Shared resources used for…
  - register file (*)
  - all other sharable resources (*)

* Policy may be needed to make priority decisions

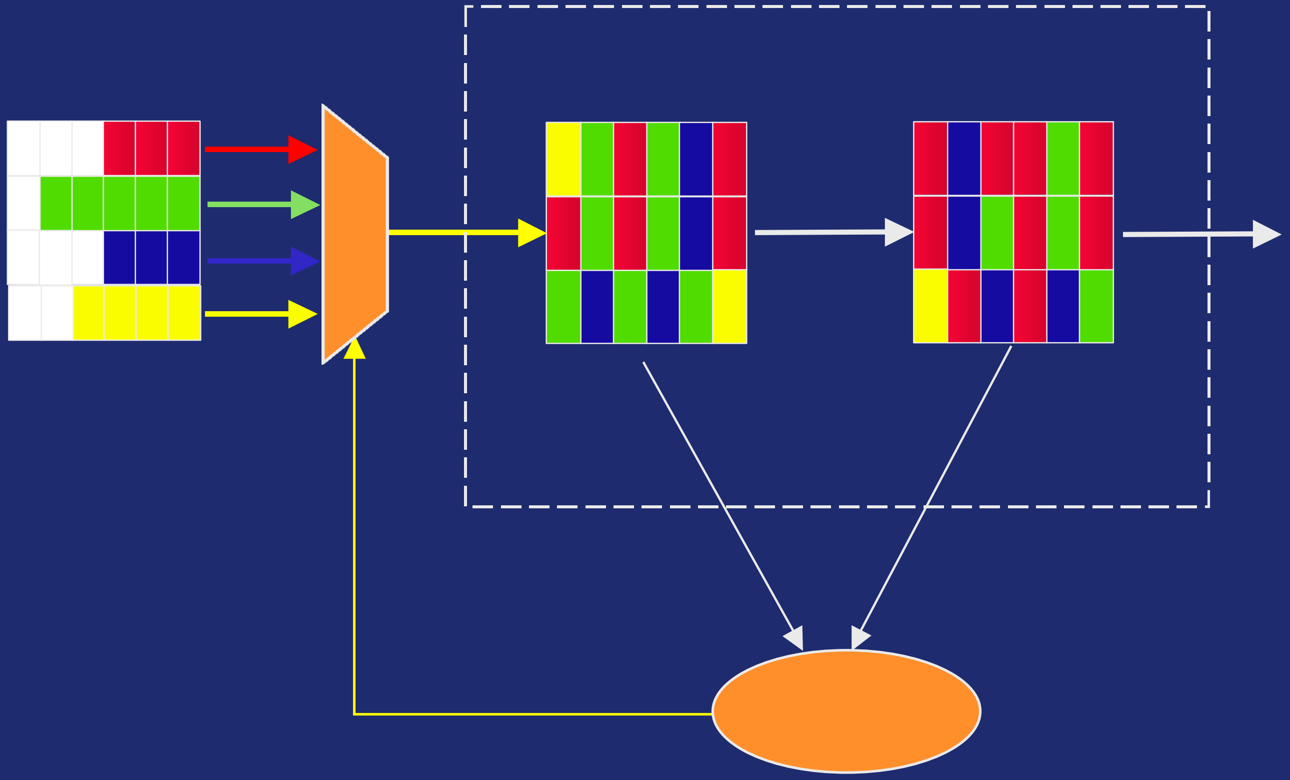
Choosing Policy



Choosing policies

- FIFO – trivial
- Round robin – easy
- Proportional – special case
- Icount-style – fair

Icount Choosing Policy

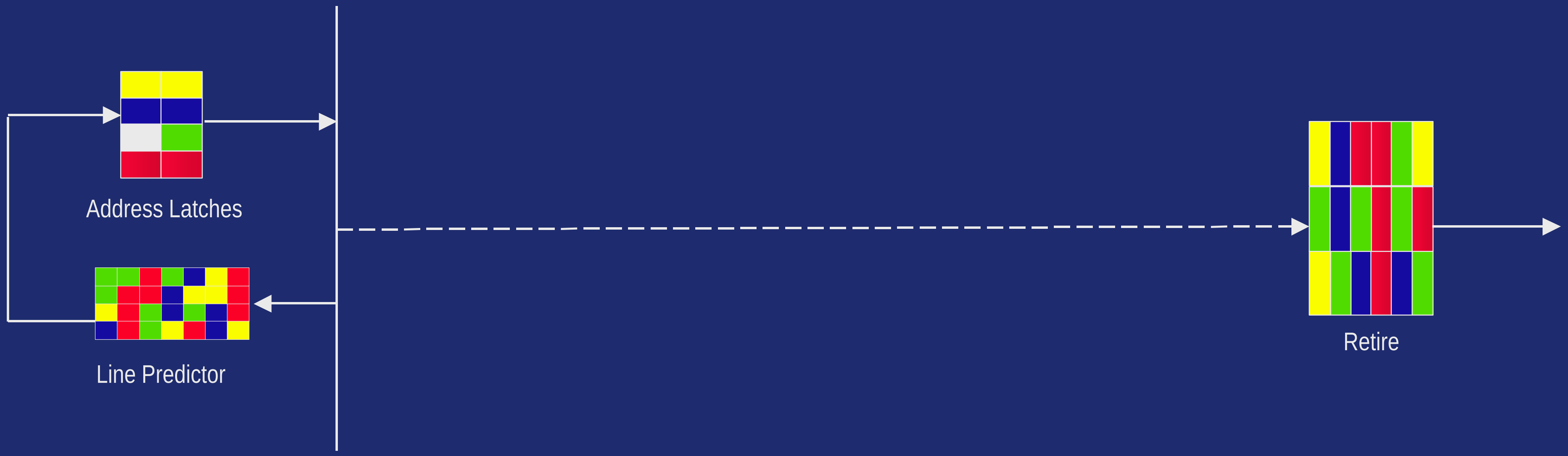


Why Does Icount Make Sense?

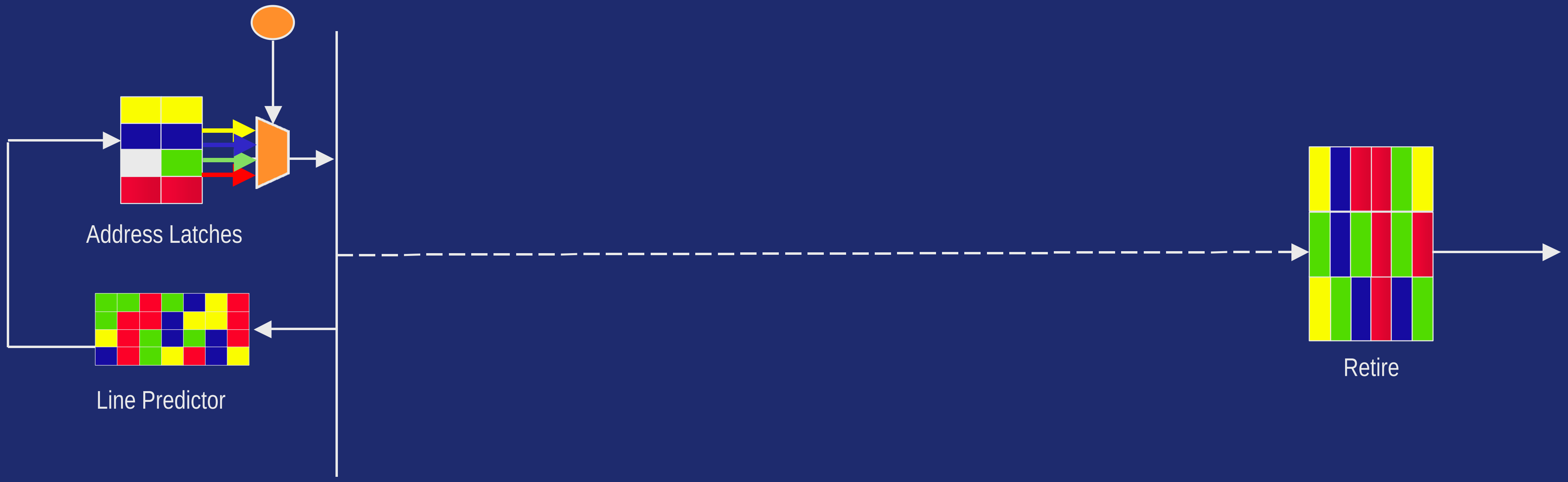
$$T = \frac{N}{L}$$

$$T/4 = \frac{N/4}{L}$$

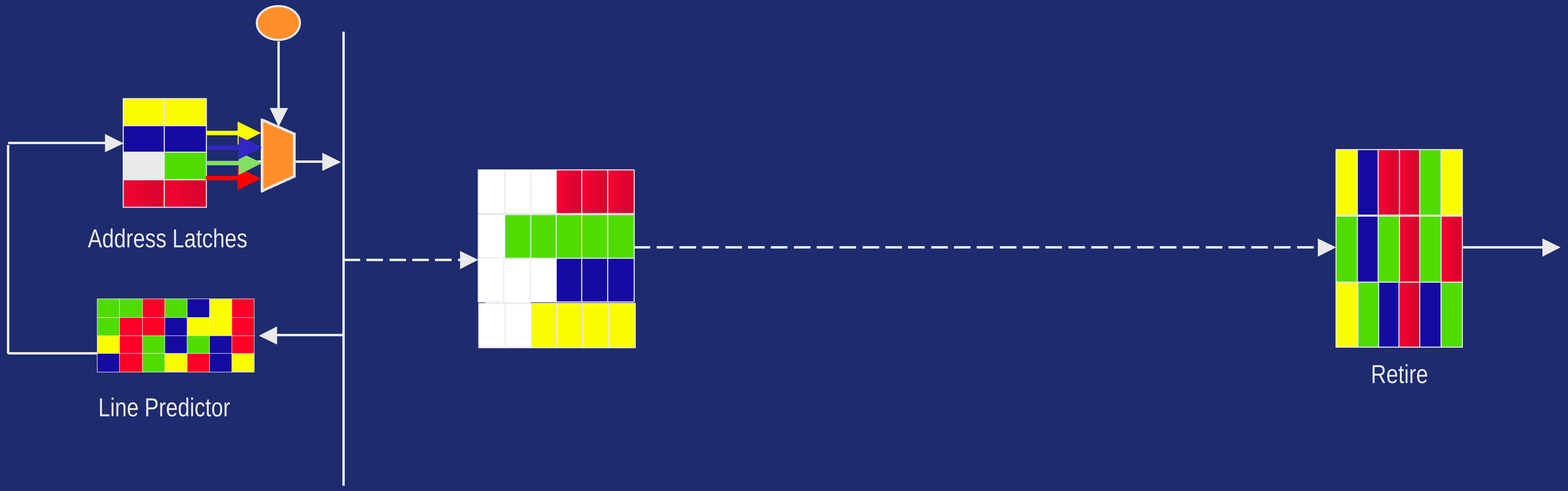
Choosers



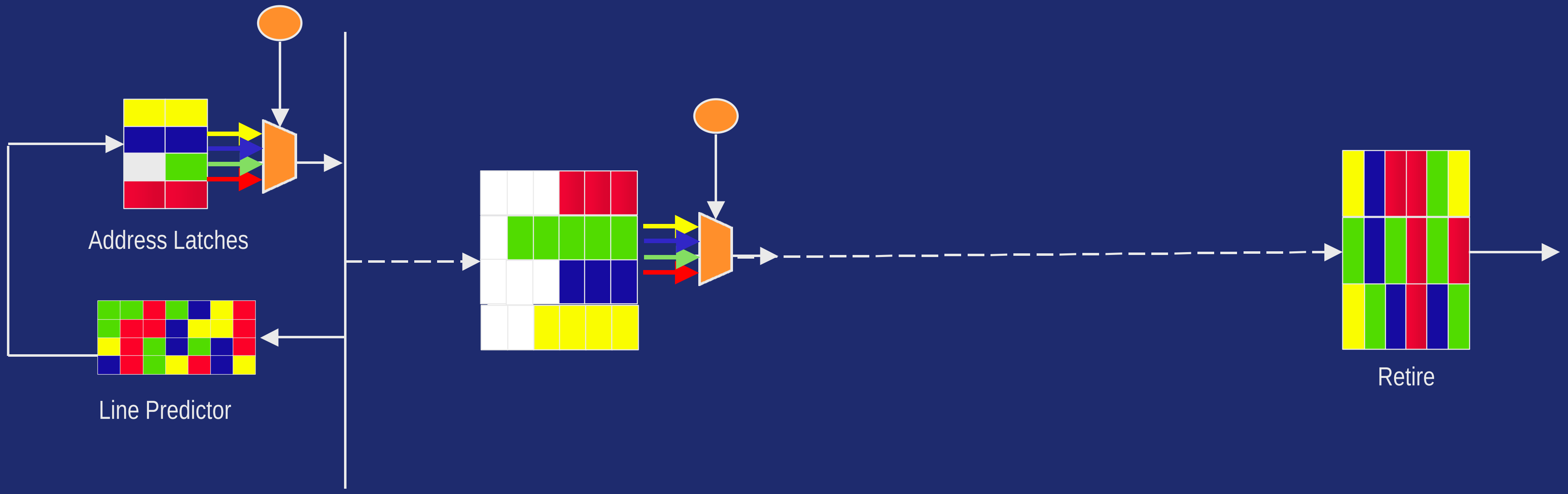
Choosers - Fetch



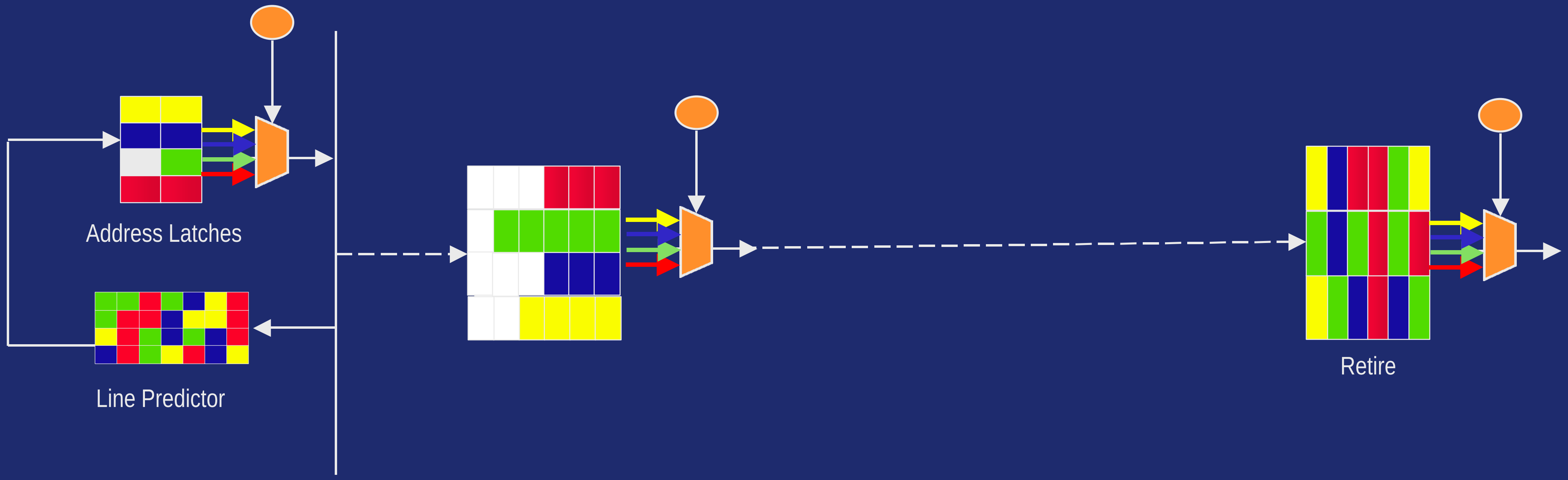
Choosers – Fetch



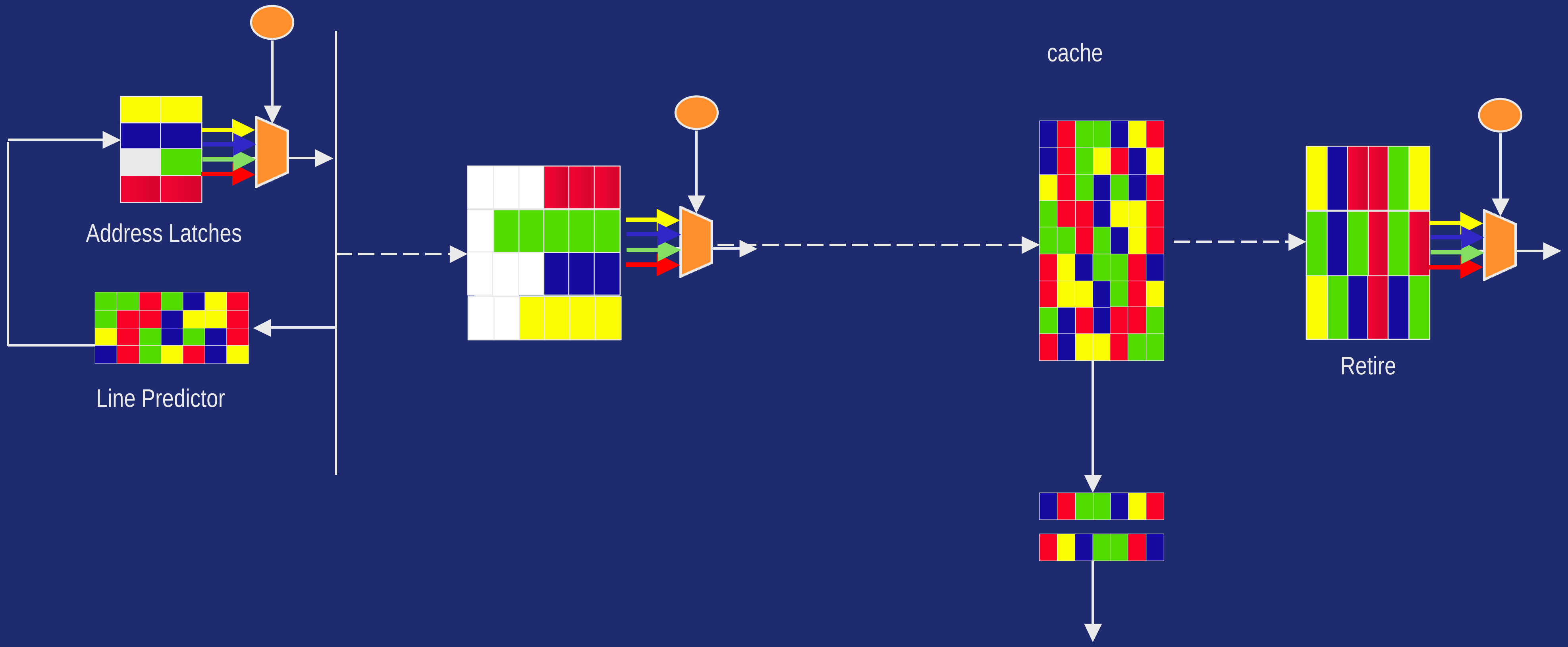
Choosers - Map



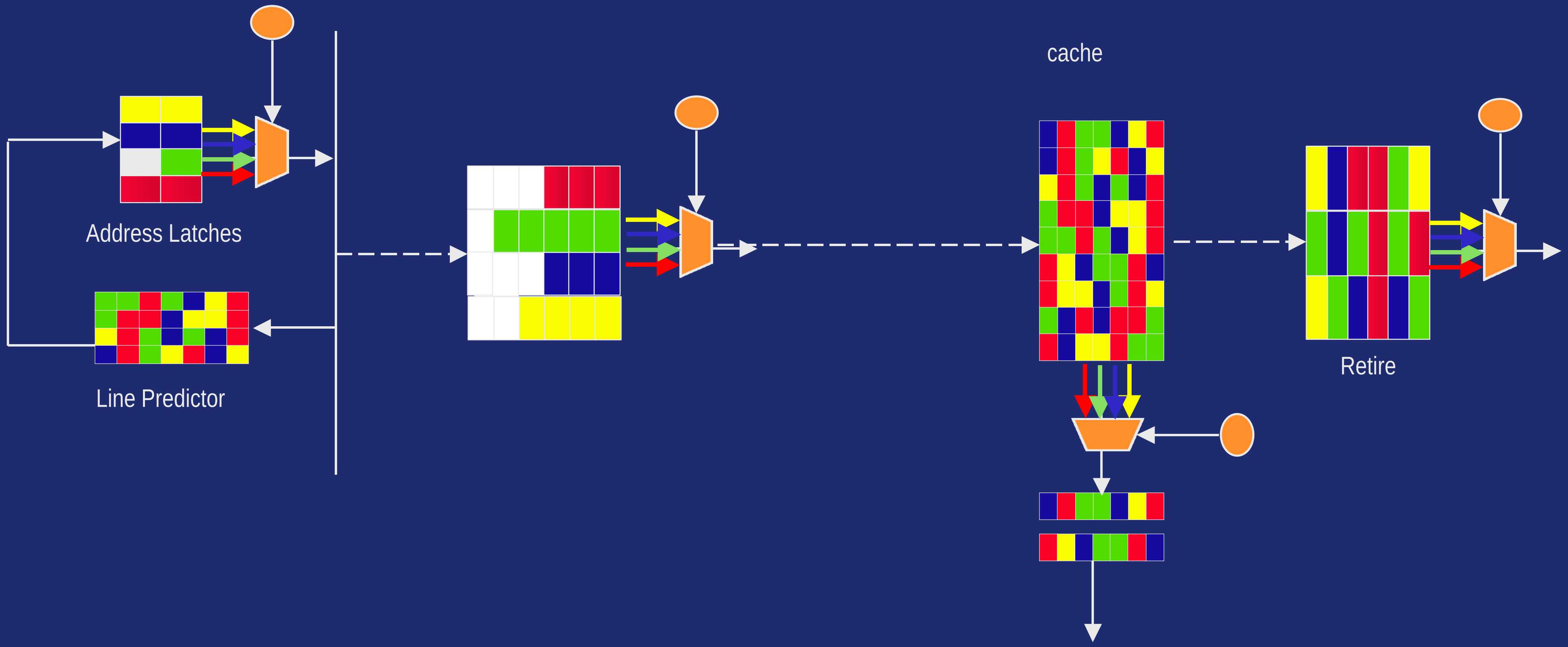
Choosers - Retire



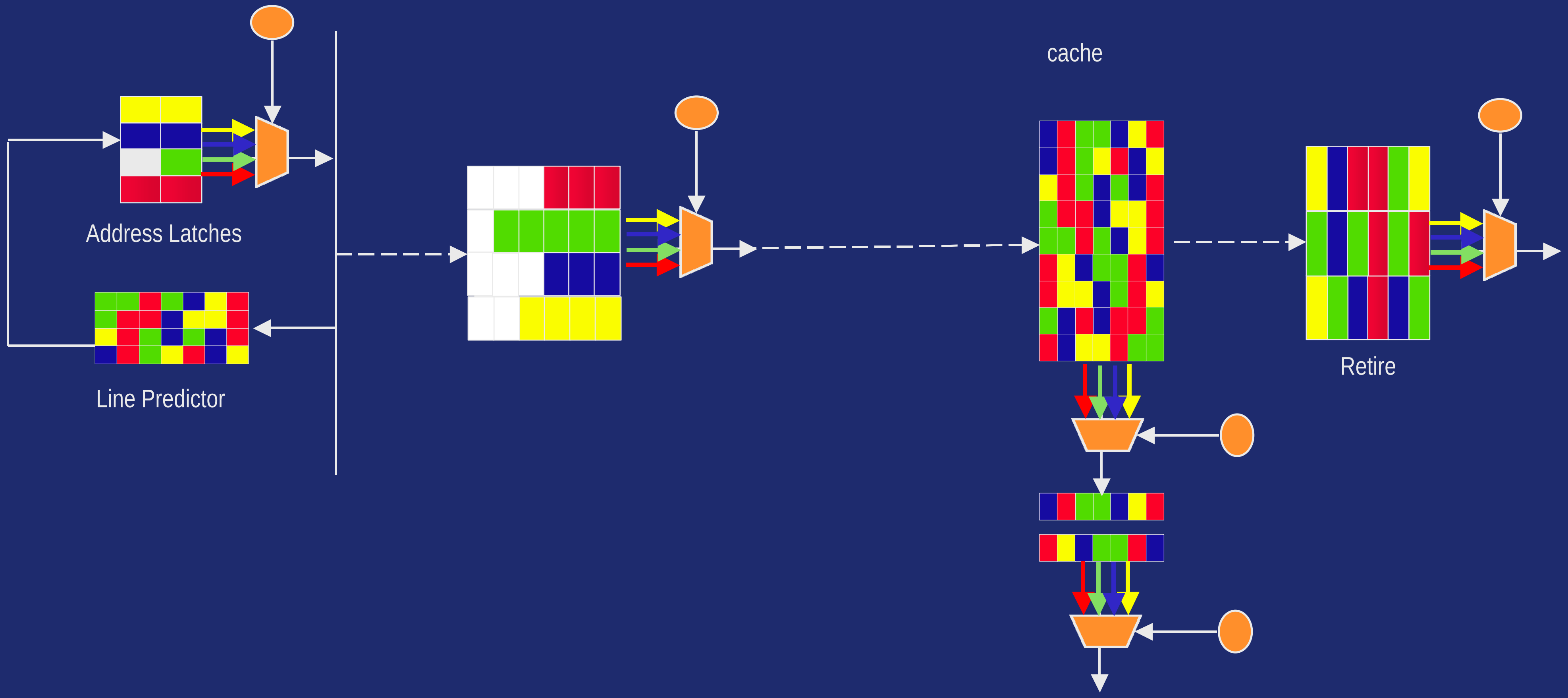
Choosers – LD/ST numbers



Choosers – LD/ST numbers



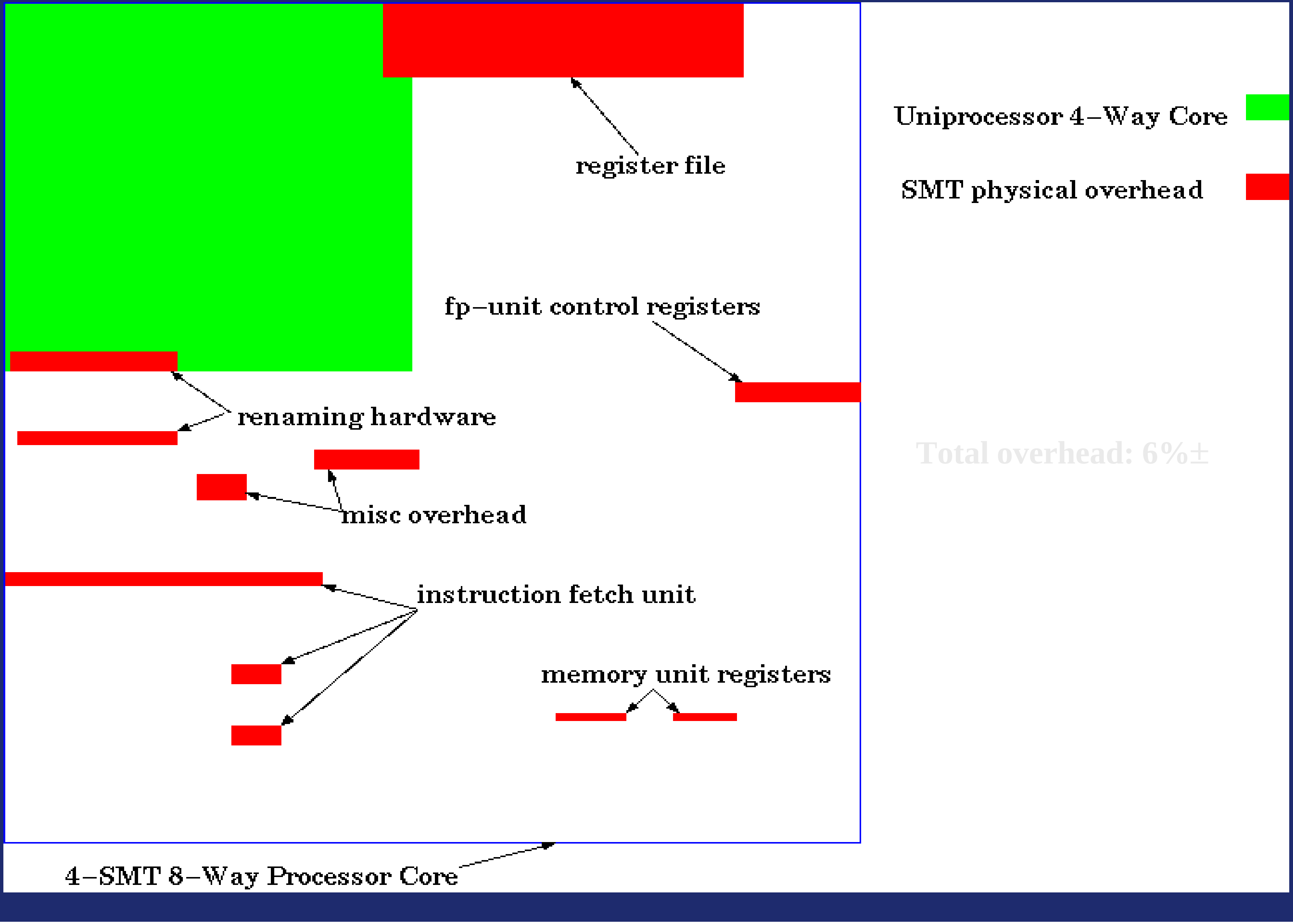
Choosers – Miss/Store



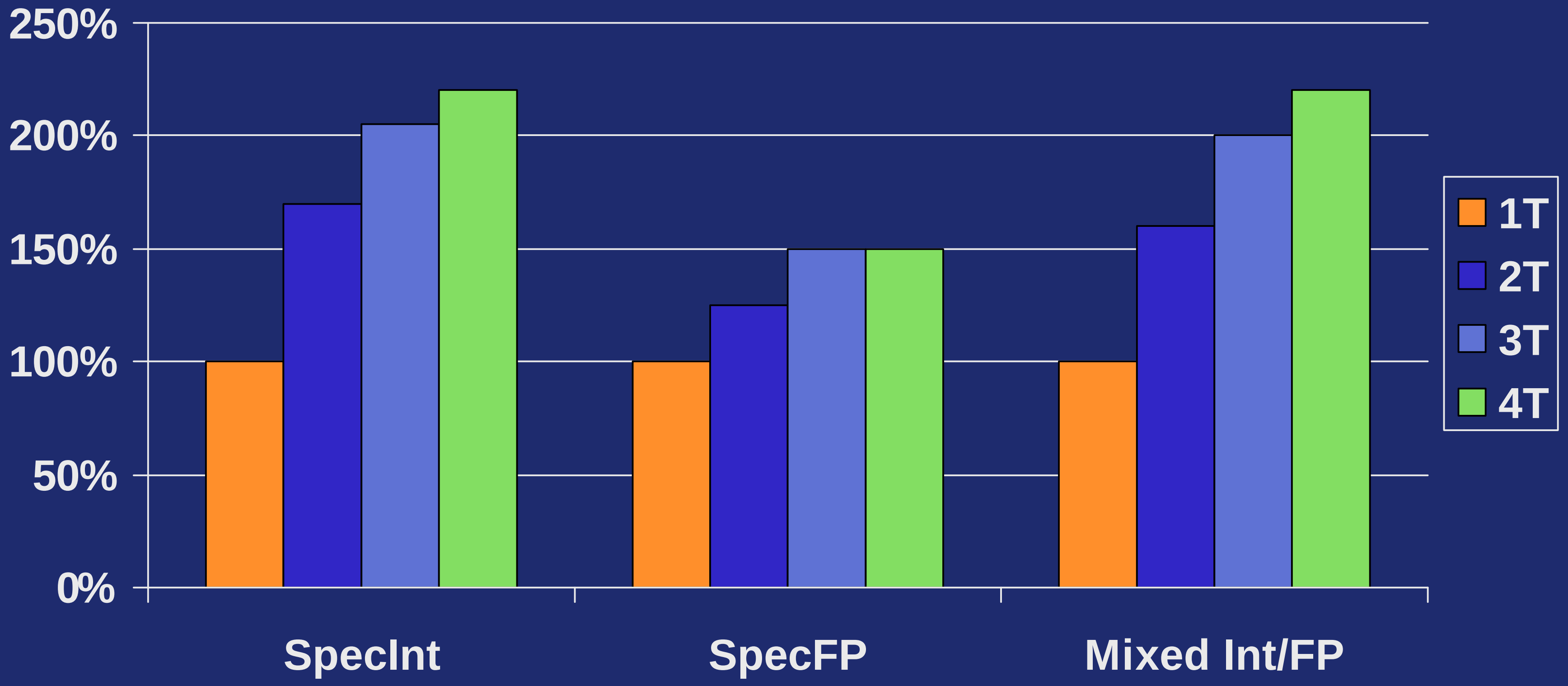
Choosers

- Fetch Chooser - Icount
- Map Chooser - Icount
- LD/ST Number Chooser - Proportional
- Retire Chooser – Round Robin
- Load miss Chooser – Round Robin
- Store Buffer Chooser - FIFO

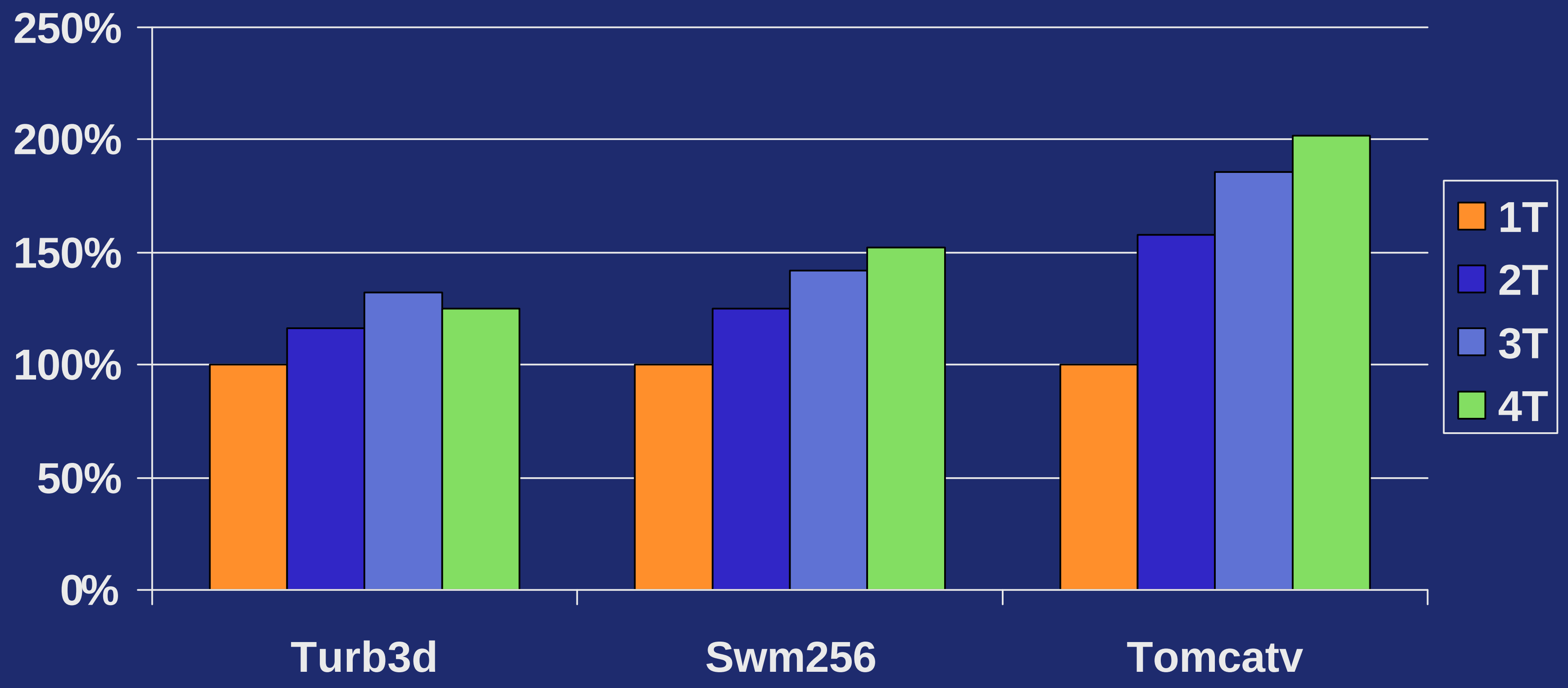
Area Cost of SMT Support



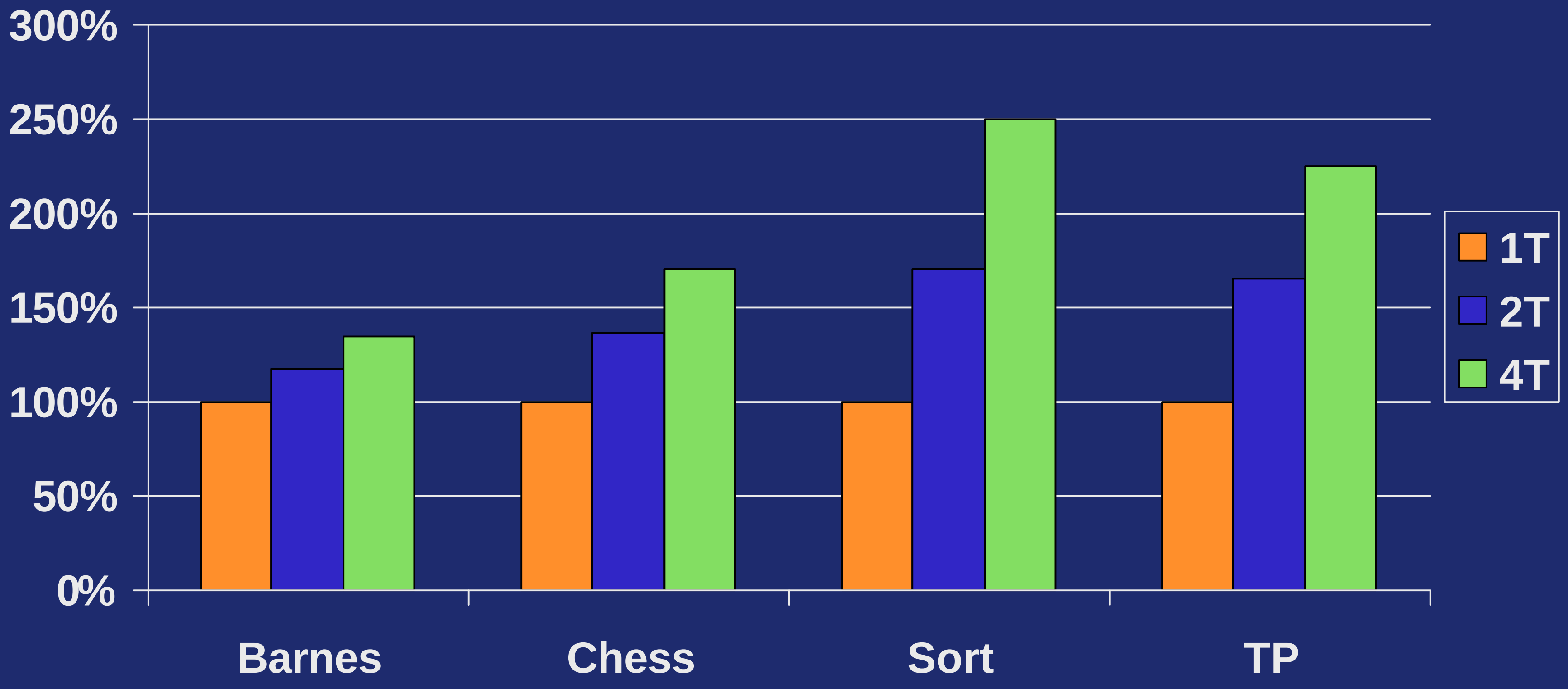
Multiprogrammed workload



Decomposed SPEC95 Applications



Multithreaded Applications



Acknowledgements

- Tryggve Fossum
  - Chuan-Hua Chang
  - George Chrysos
  - Steve Felix
  - Chris Gianos
  - Partha Kundu
  - Jud Leonard
  - Matt Mattina
  - Matt Reilly