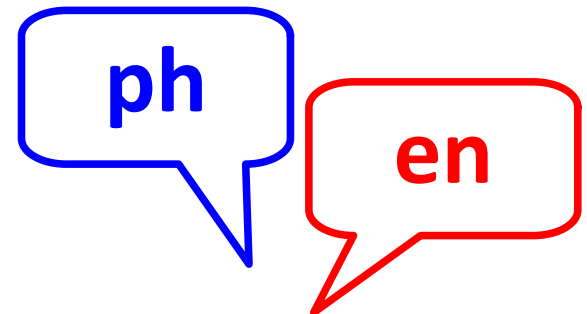


# CMSC 180

## Introduction to Parallel Computing



# OUTLINE

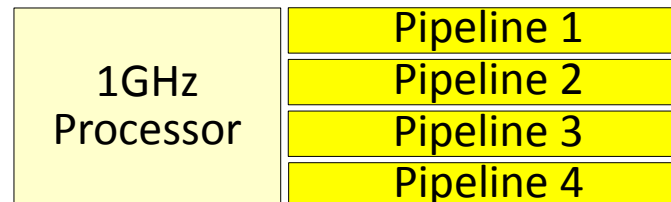
- What is the effect of memory cache on the performance of computations?
  - Example specifications of processor and memory
  - Example computation: Dot product of two vectors
  - Compare :  
theoretical processor rating vs. actual processor rating
  - Practical stuff

# EXAMPLE HARDWARE SPECS

## 1 Processor:

- 1GHz clock speed (1 clock tick per 1 ns)
- Can execute 4 instructions per 1 cycle (1 ns)

WHICH MEANS  
IT HAS  
FOUR PIPELINES.

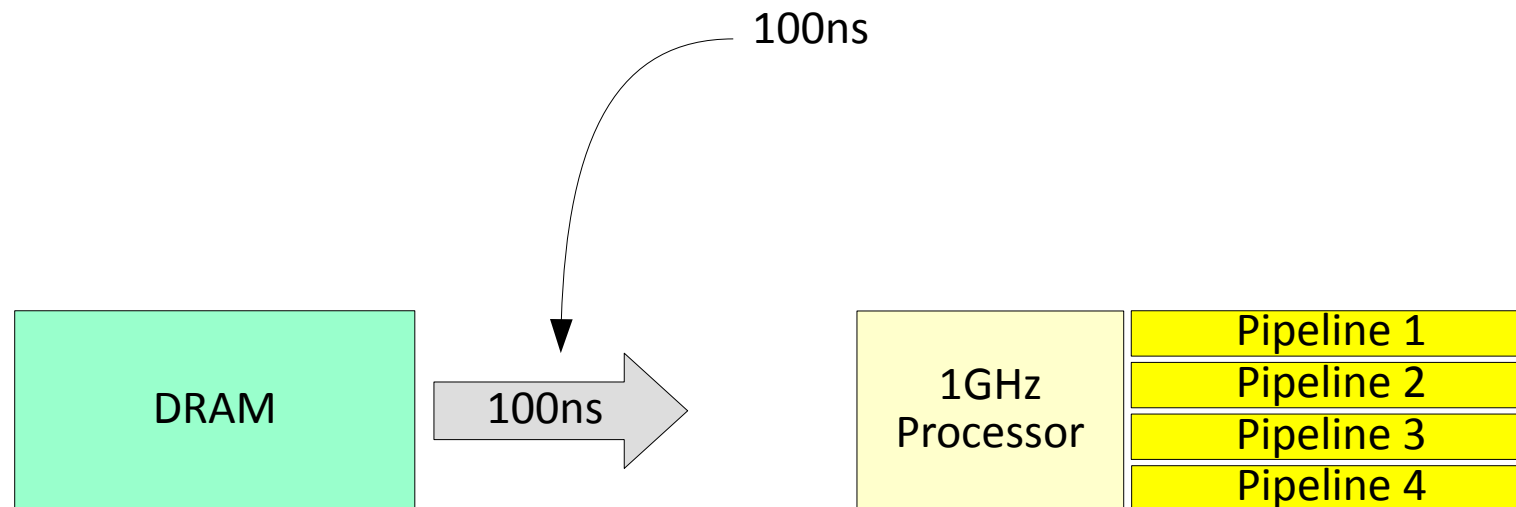


# EXAMPLE HARDWARE SPECS

## 1 Processor:

- 1GHz clock speed (1 clock tick per 1 ns)
- Can execute 4 instructions per 1 cycle (1 ns)

## 2 DRAM: 100ns latency



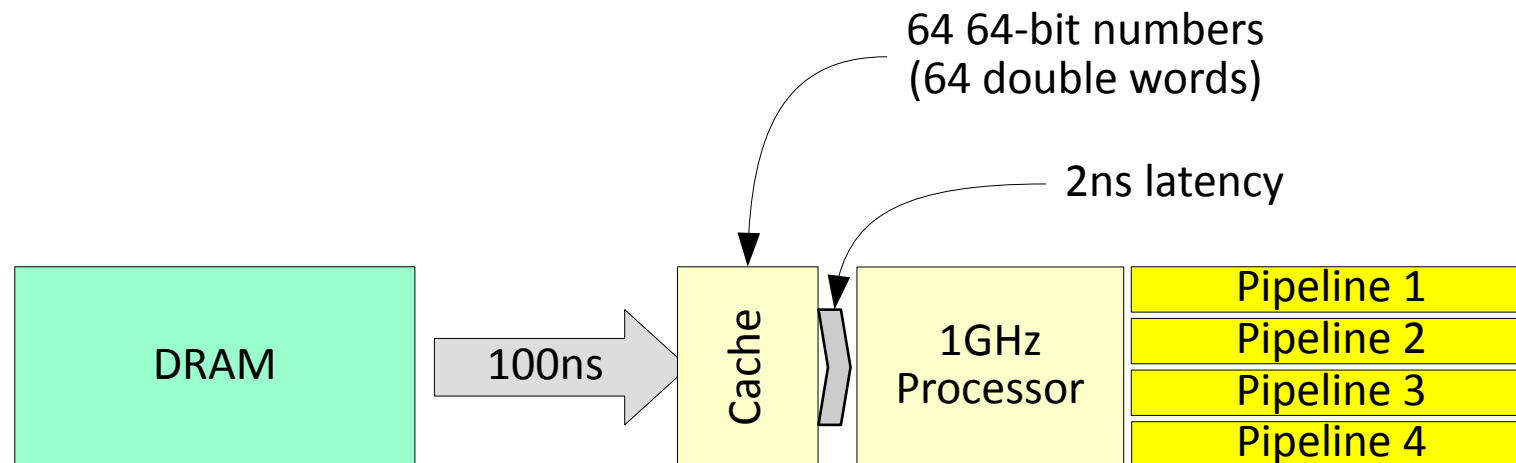
# EXAMPLE HARDWARE SPECS

## 1 Processor:

- 1GHz clock speed (1 clock tick per 1 ns)
- Can execute 4 instructions per 1 cycle (1 ns)

## 2 DRAM: 100ns latency

## 3 Cache: 2ns latency, 64 double words



# EXAMPLE HARDWARE SPECS

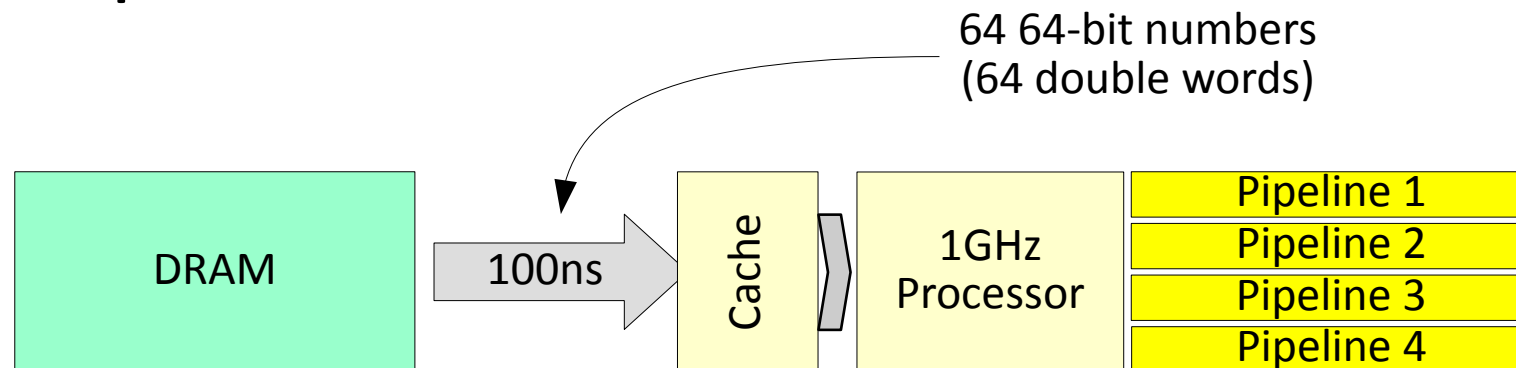
## 1 Processor:

- 1GHz clock speed (1 clock tick per 1 ns)
- Can execute 4 instructions per 1 cycle (1 ns)

## 2 DRAM: 100ns latency

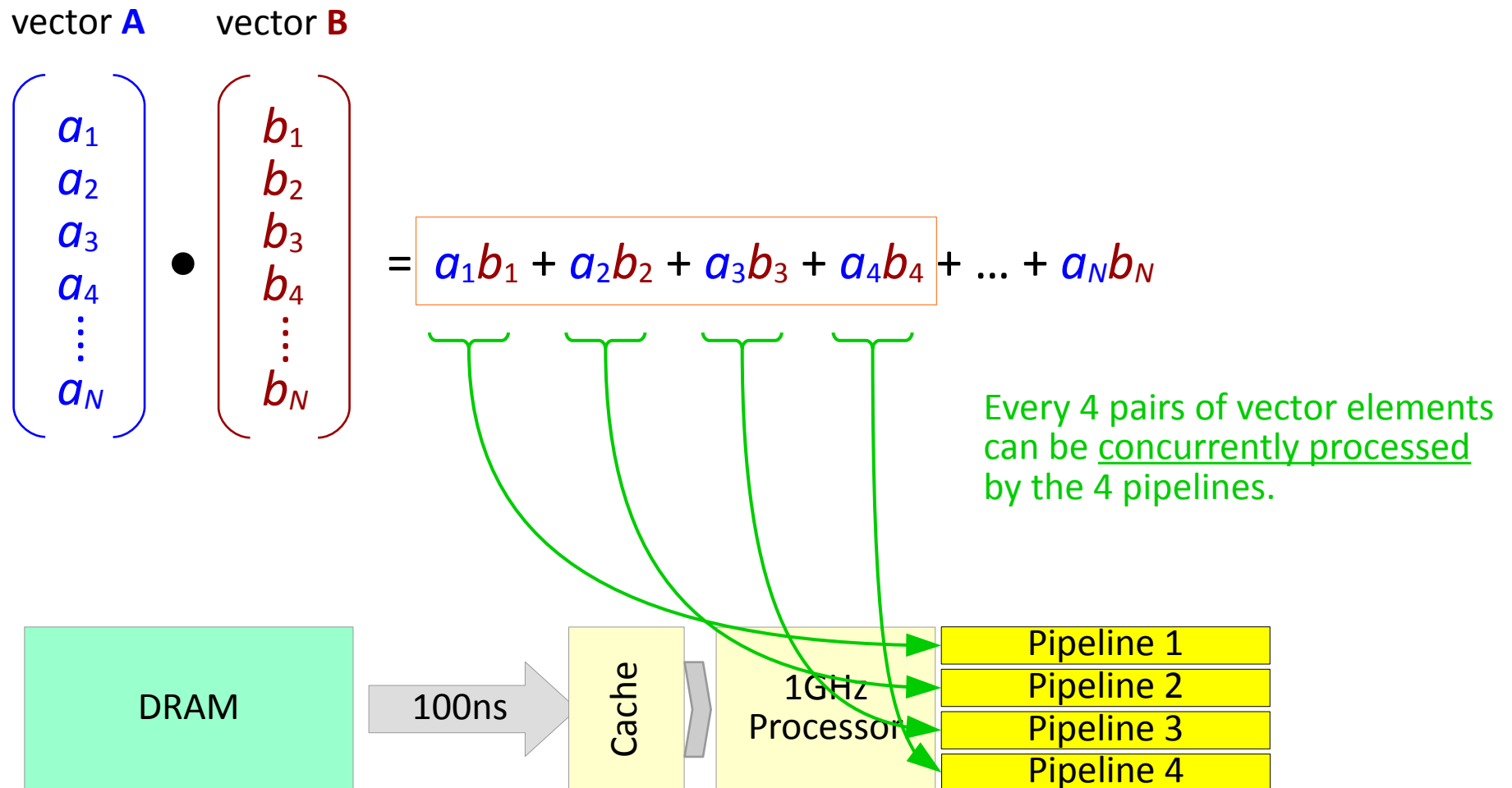
## 3 Cache: 2ns latency, 64 double words

## 4 Datapath: 64 double words



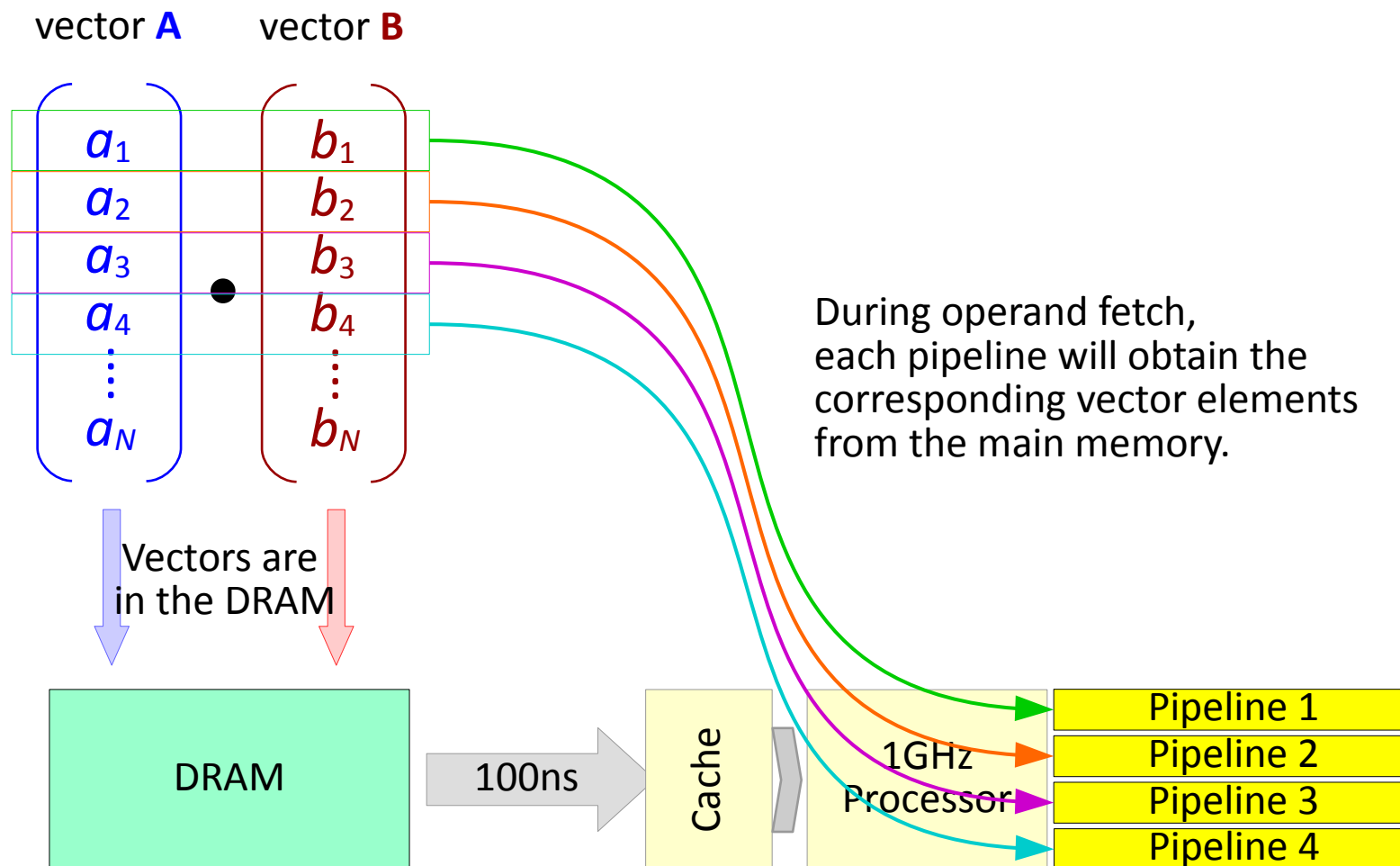
# EXAMPLE APPLICATION

- Dot-product of two vectors: **A** and **B**



# EXAMPLE APPLICATION


- Dot-product of two vectors: **A** and **B**





# PROCESSOR RATING

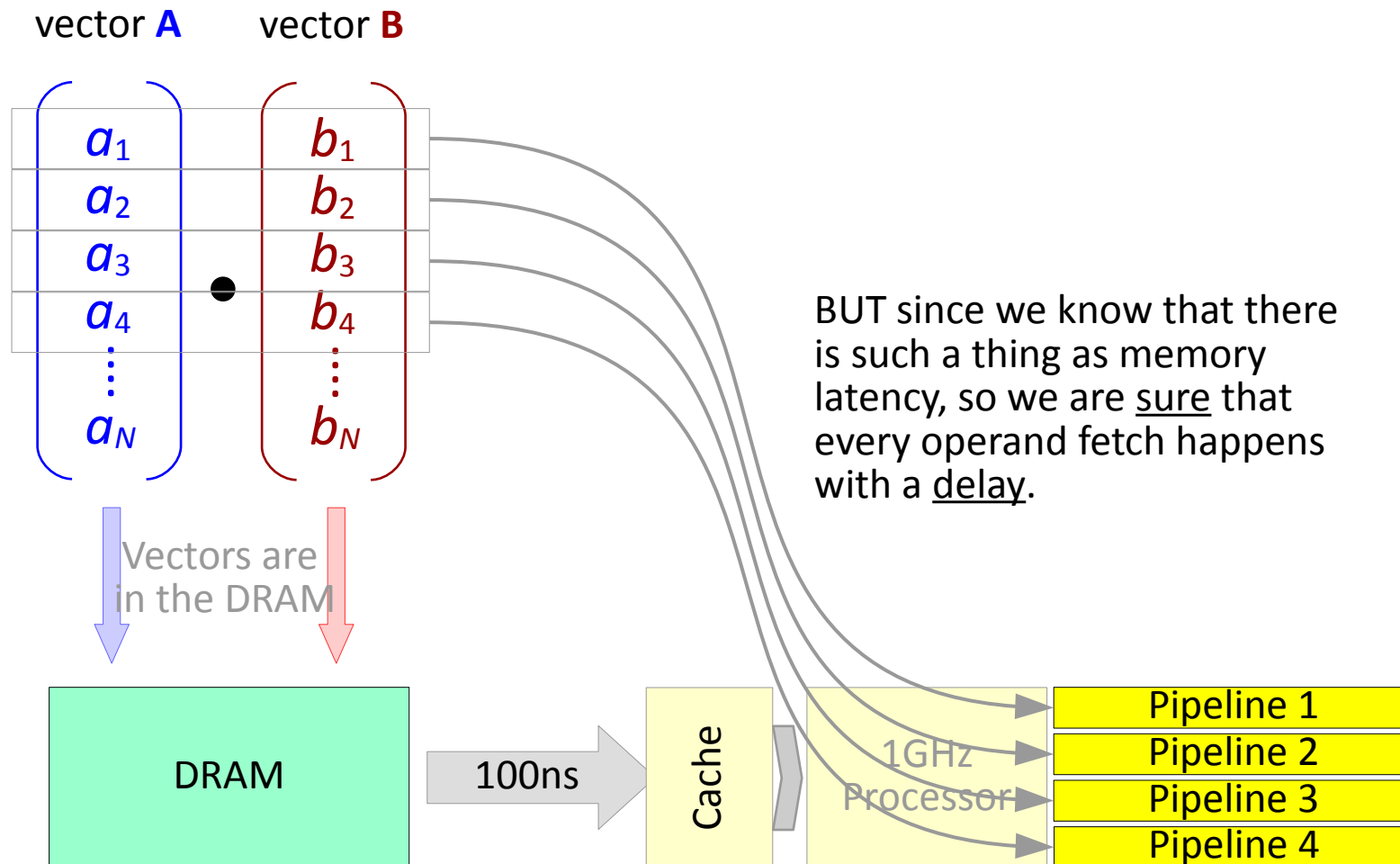
- Theoretical processor rating:
  - ▷ Four billion multiply-add operations in a second
  - ▷ Four billion floating-point operations per second
  - ▷ 4GFLOPS



THAT'S AN AWESOME  
4 GIGAFLOPS  
RATING FOR A  
1GHz PROCESSOR  
SPEED.

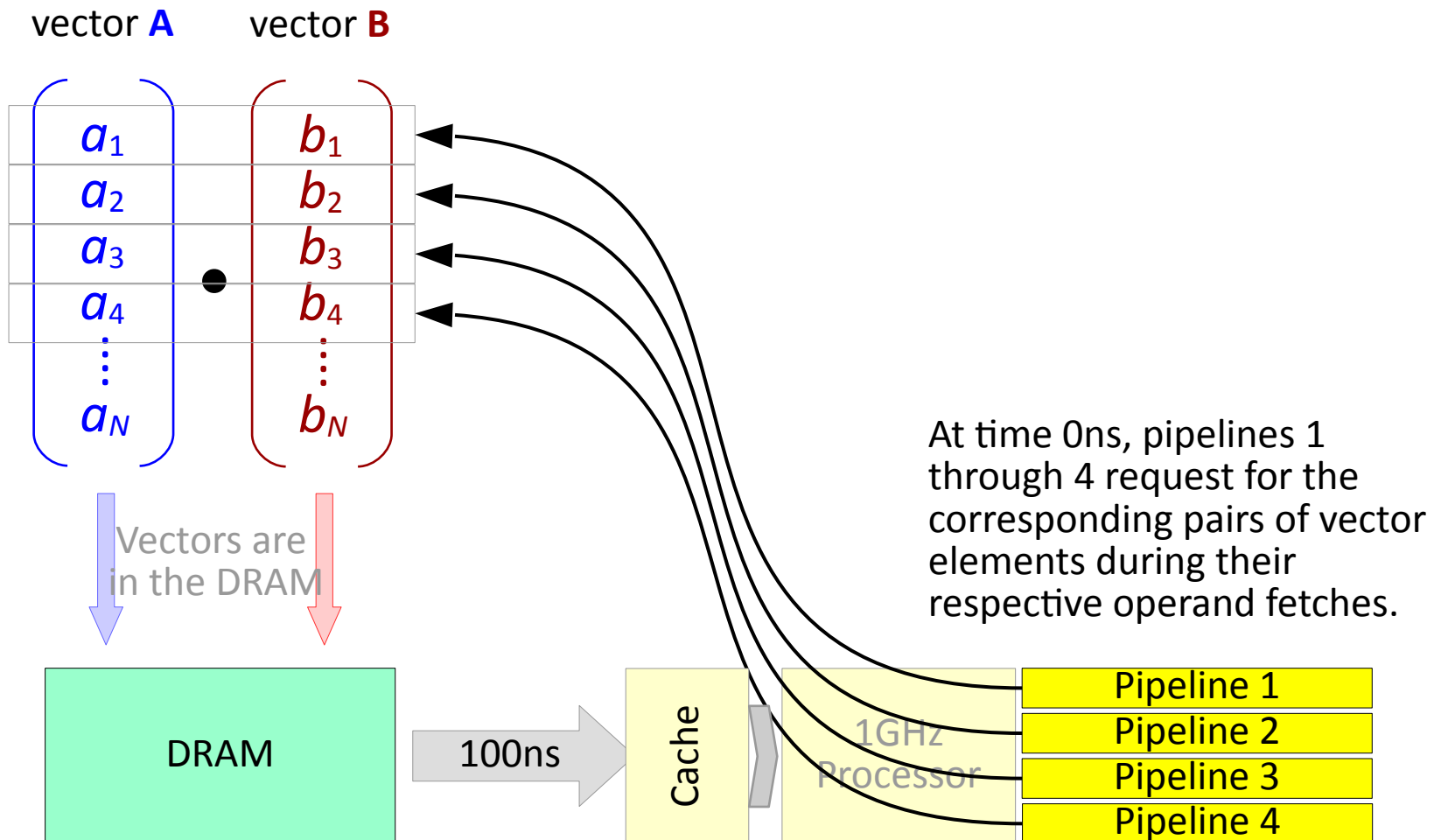
# PROCESSOR RATING

- Actual processor rating:



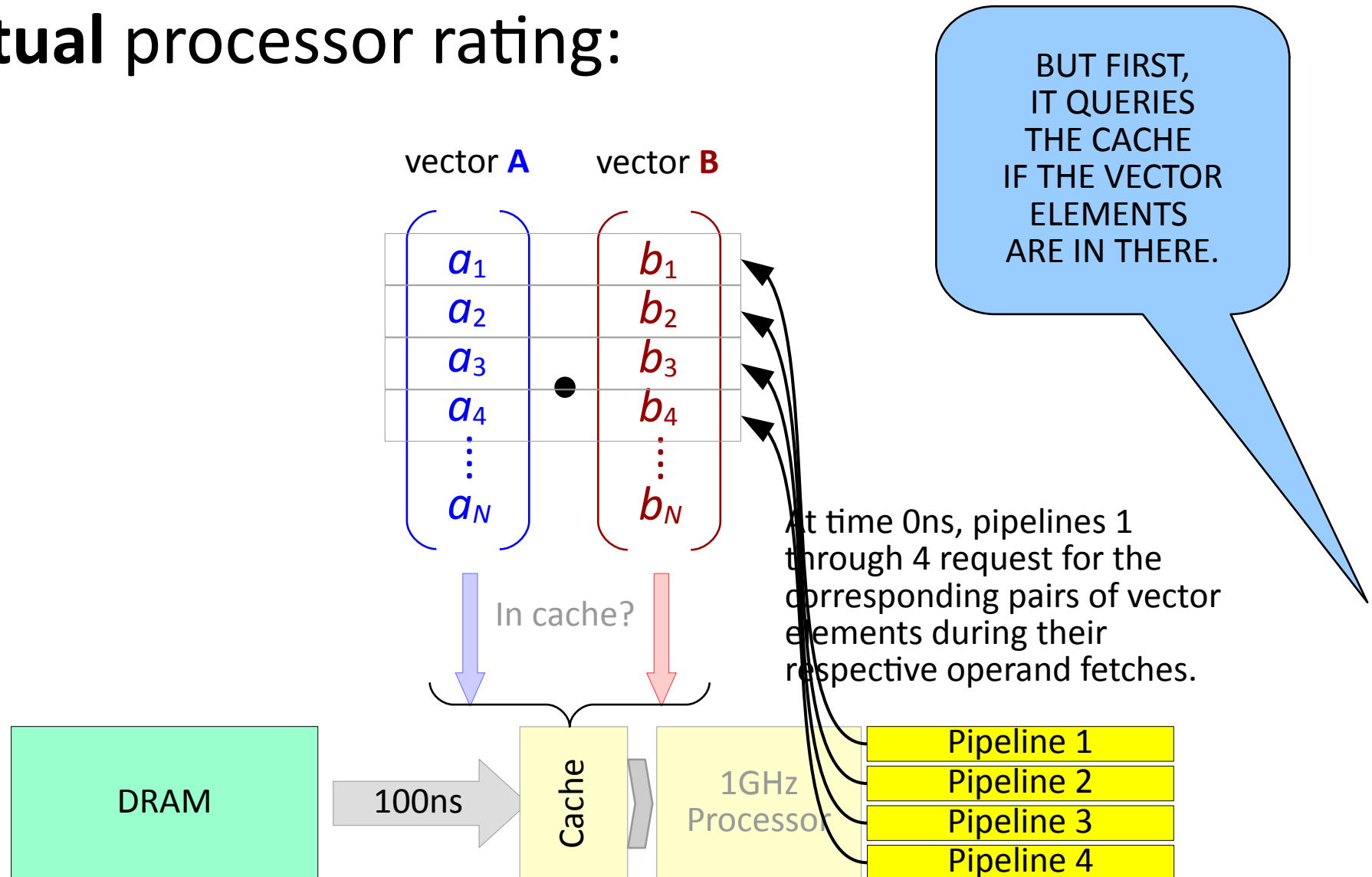
# PROCESSOR RATING

- Actual processor rating:



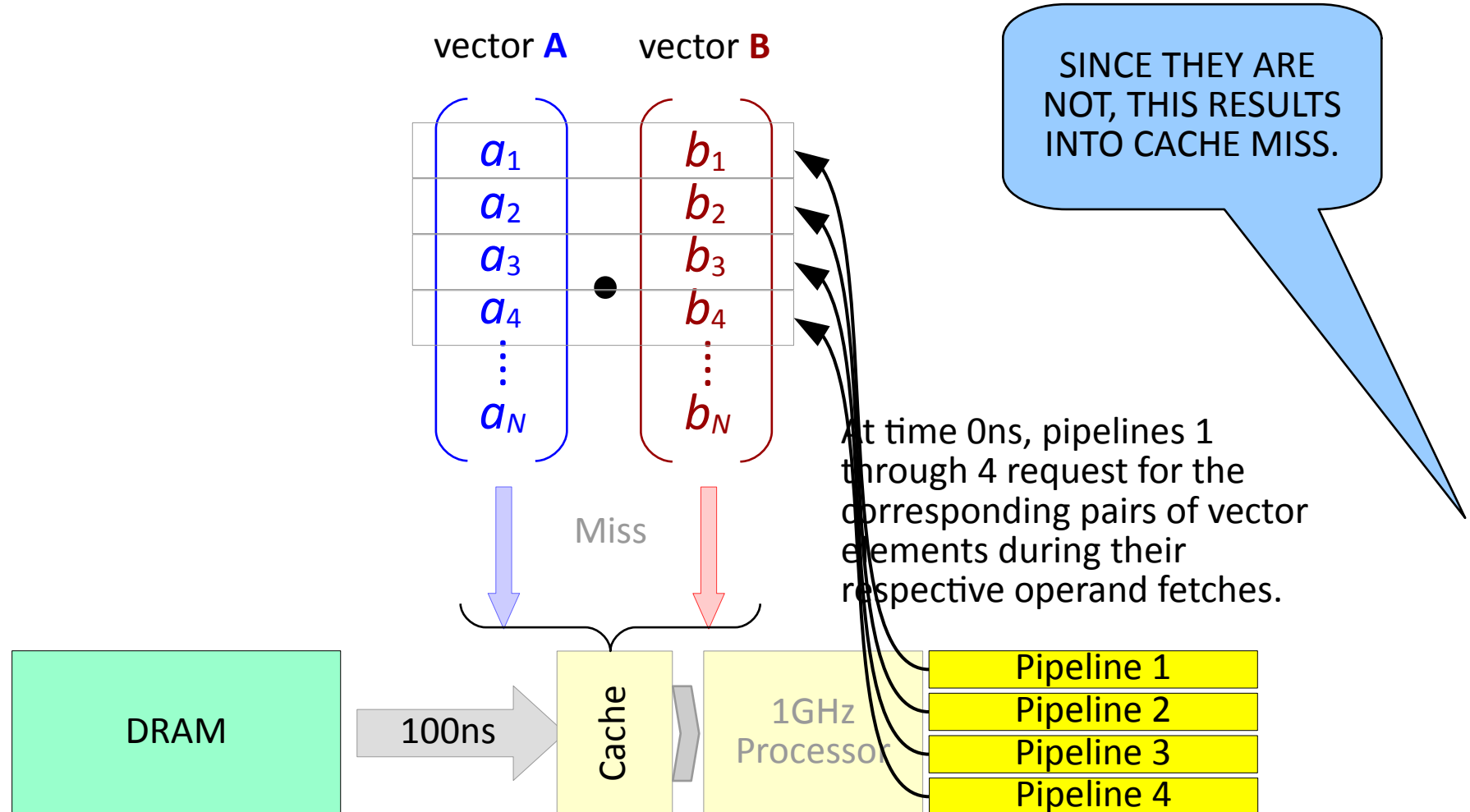
# PROCESSOR RATING

- Actual processor rating:



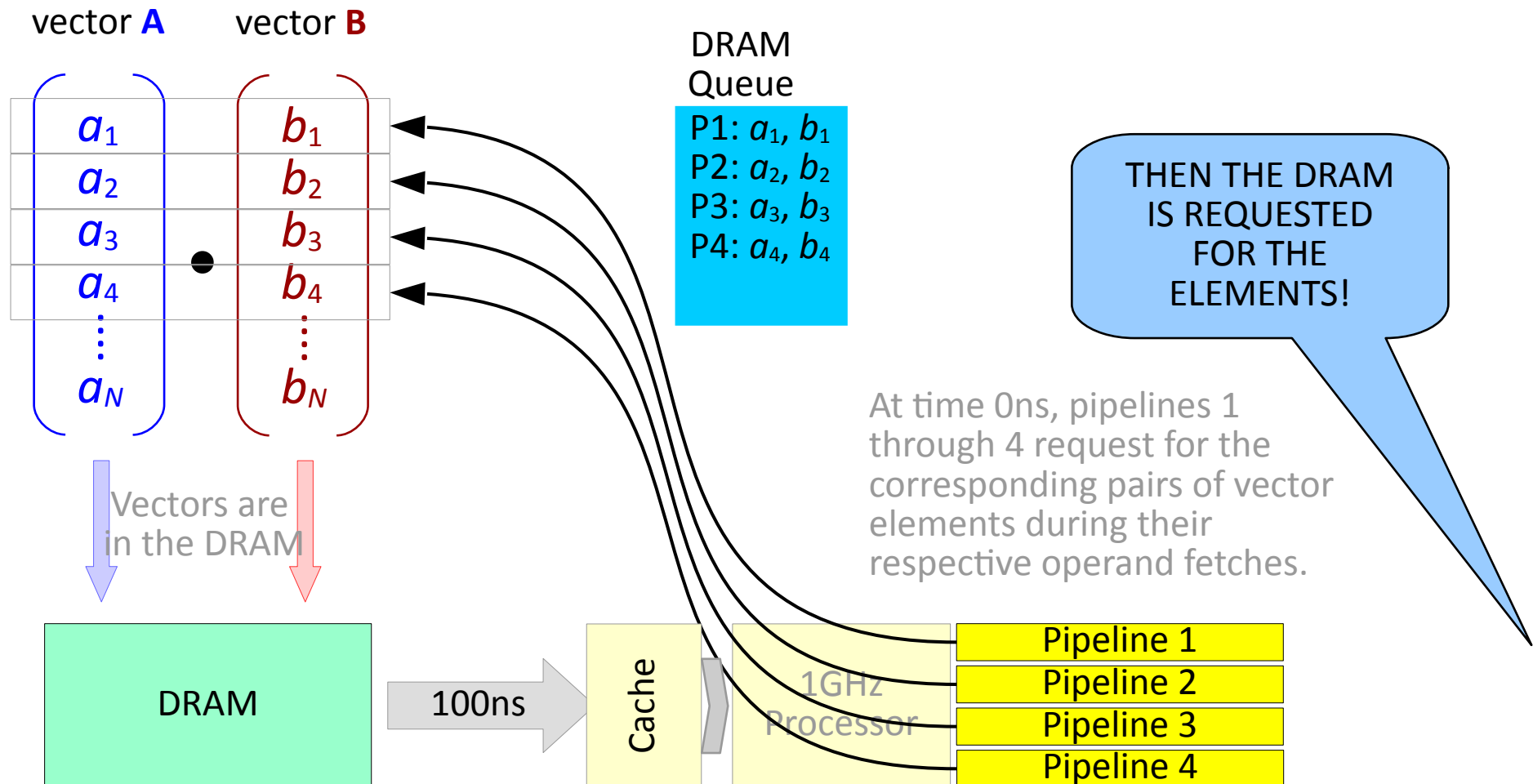
# PROCESSOR RATING

- Actual processor rating:



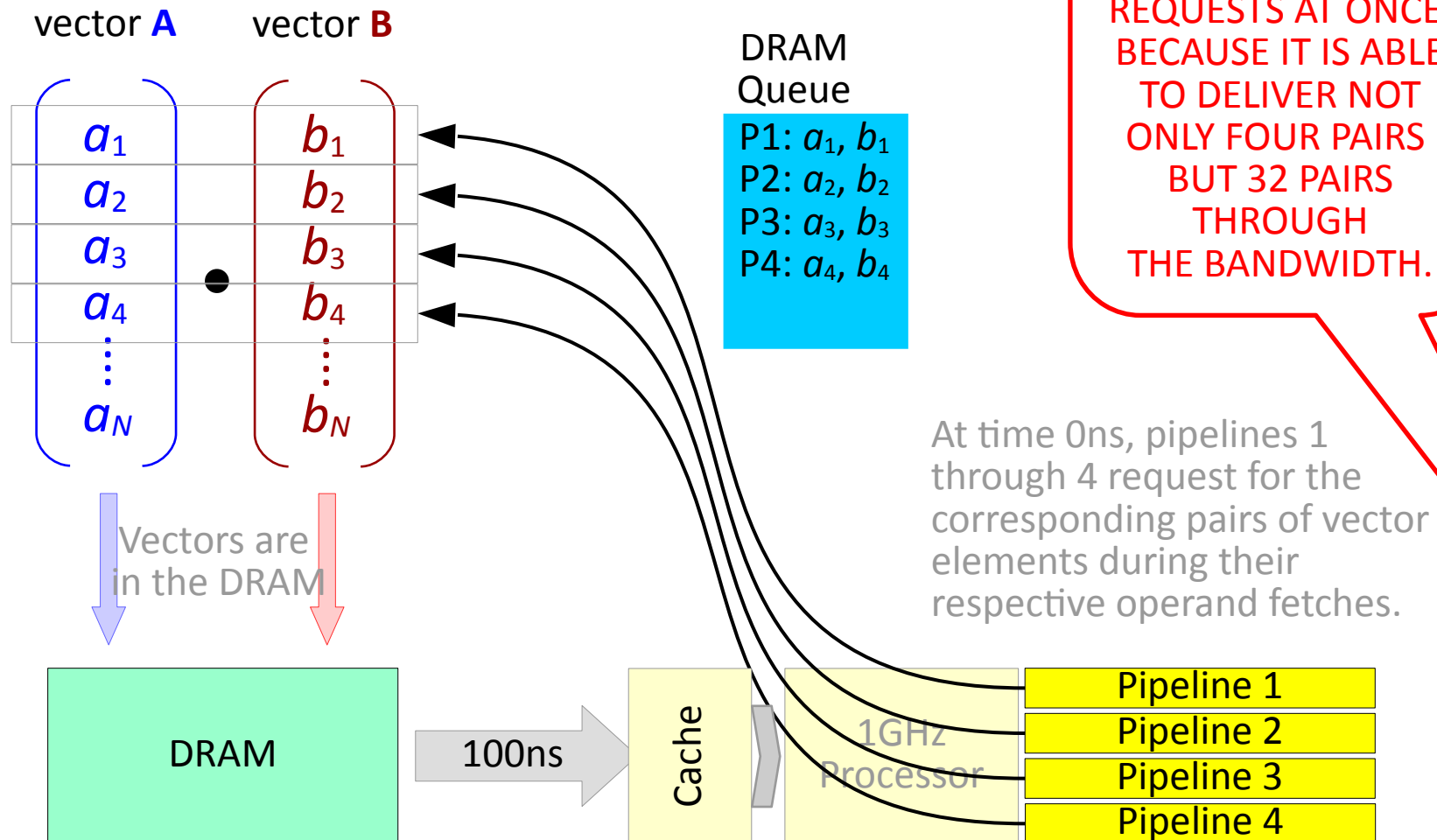
# PROCESSOR RATING

- Actual processor rating:



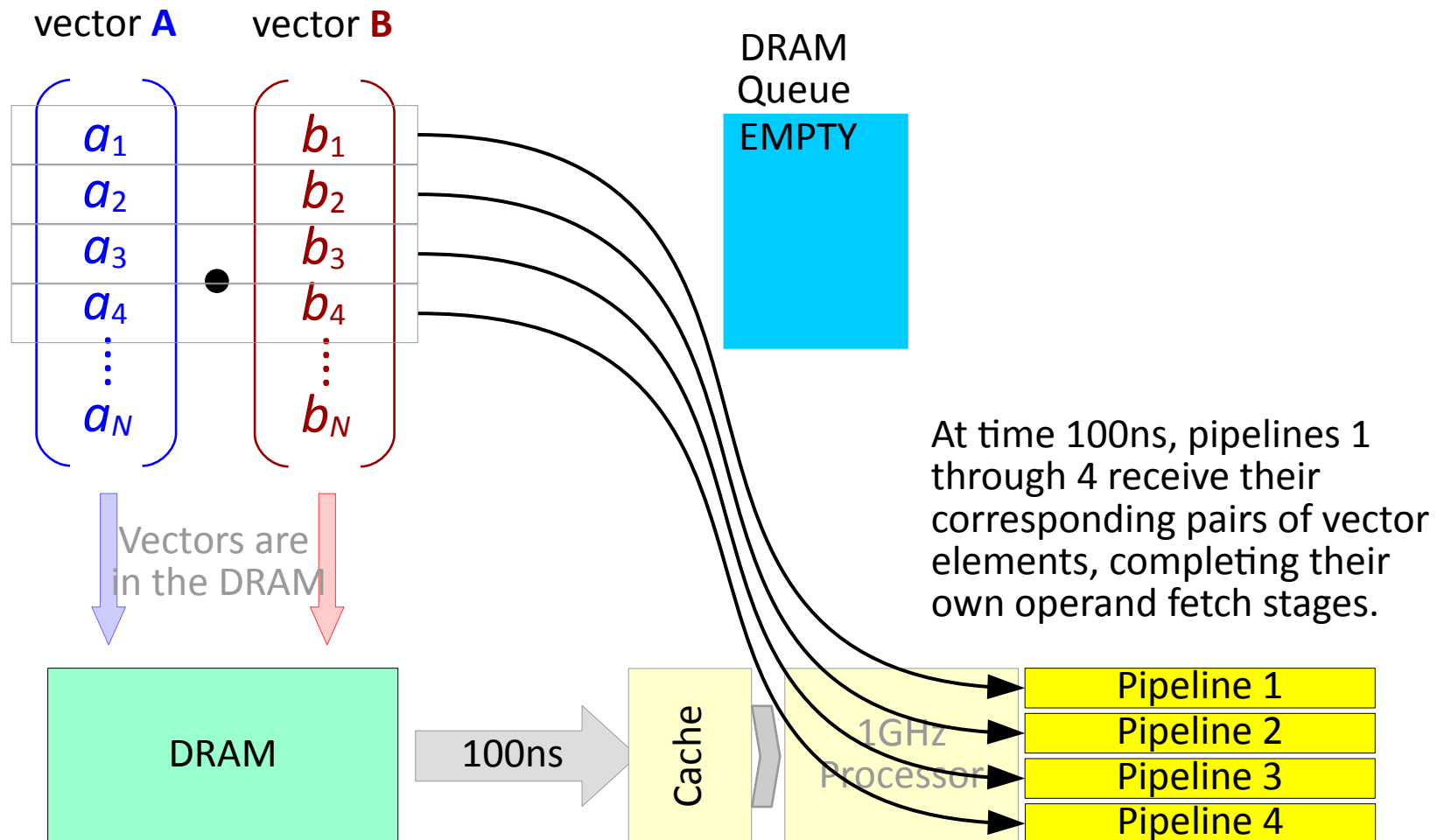
# PROCESSOR RATING

- Actual processor rating:



# PROCESSOR RATING

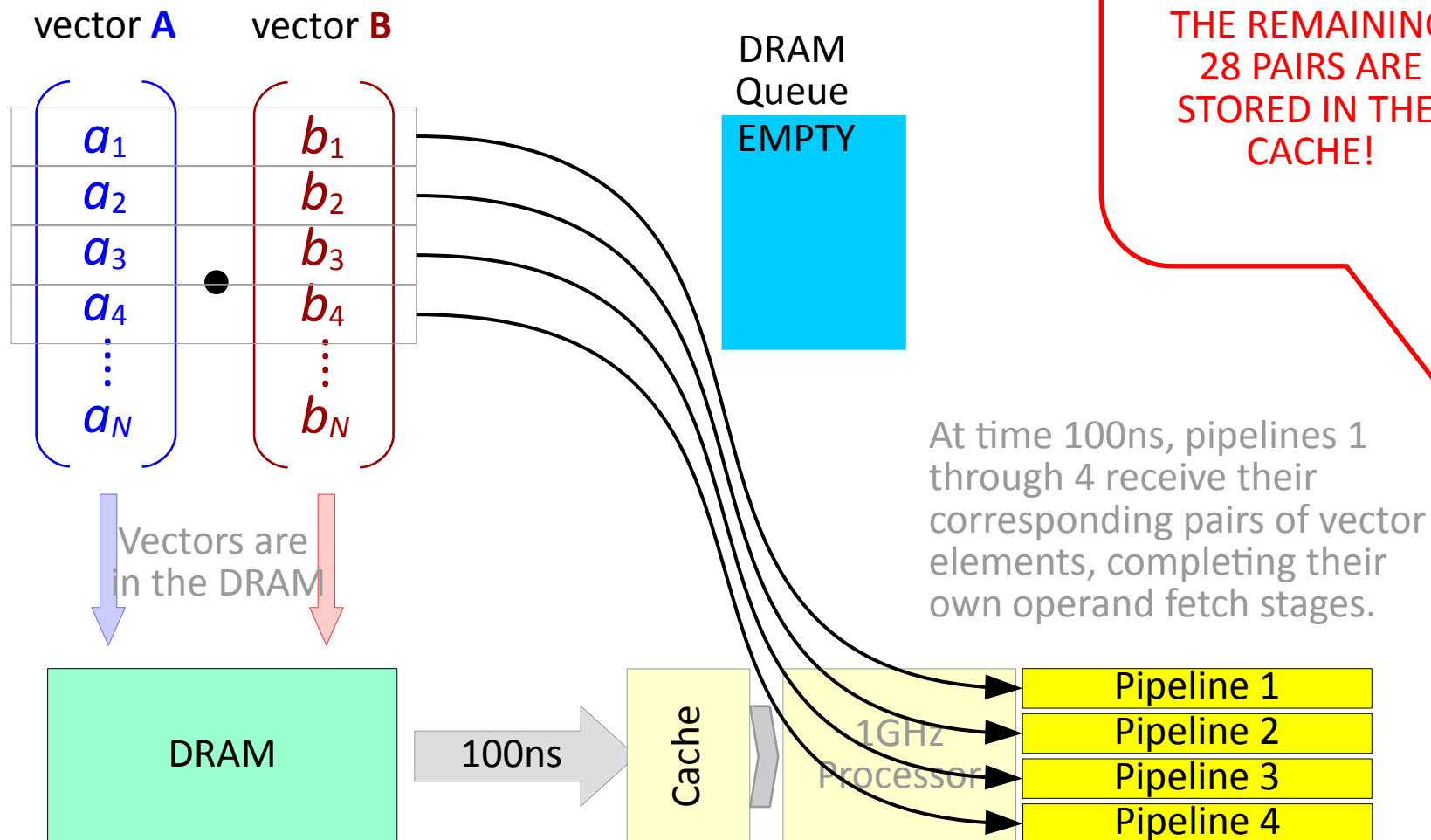
- Actual processor rating:





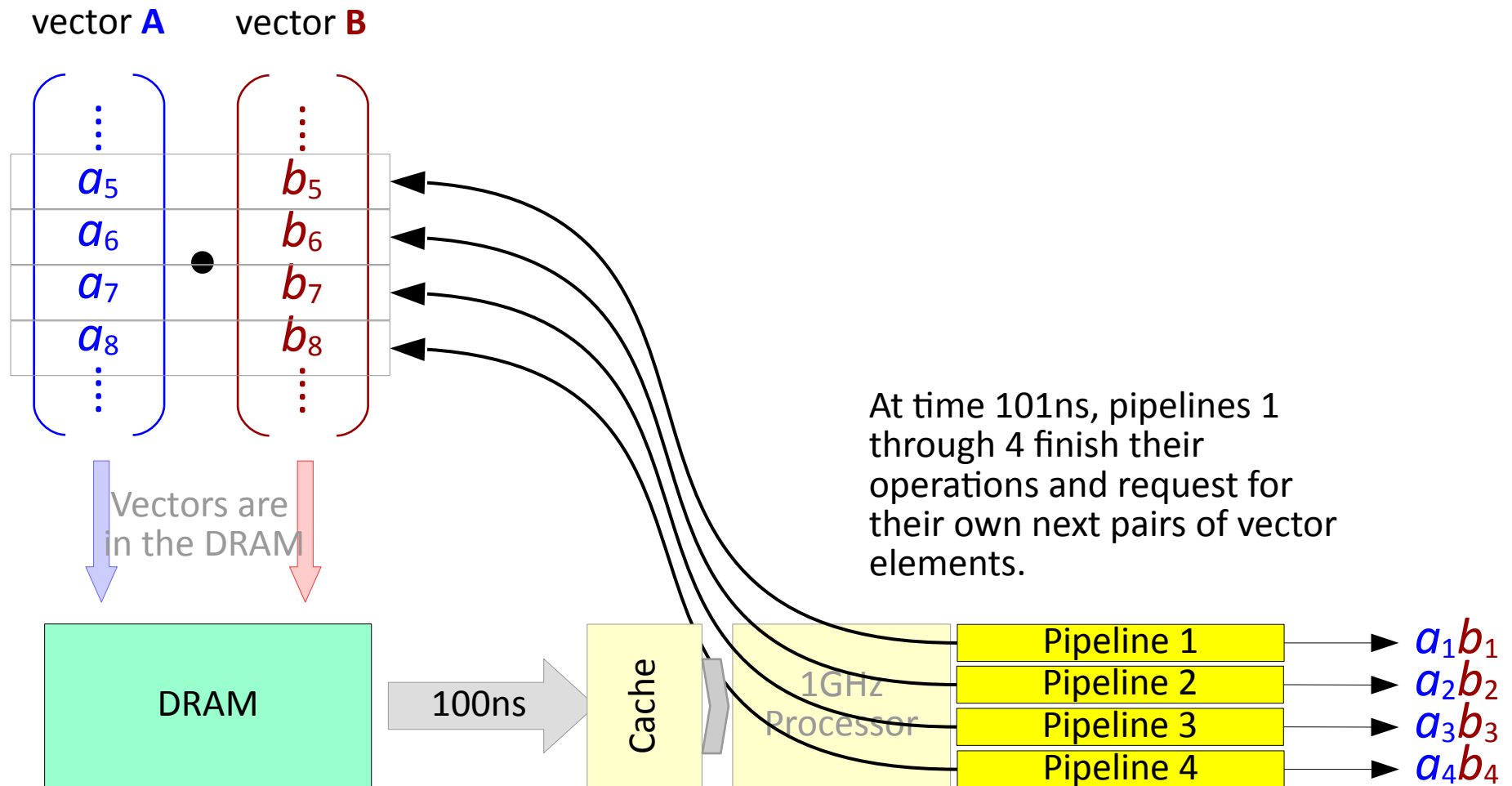
# PROCESSOR RATING

- Actual processor rating:



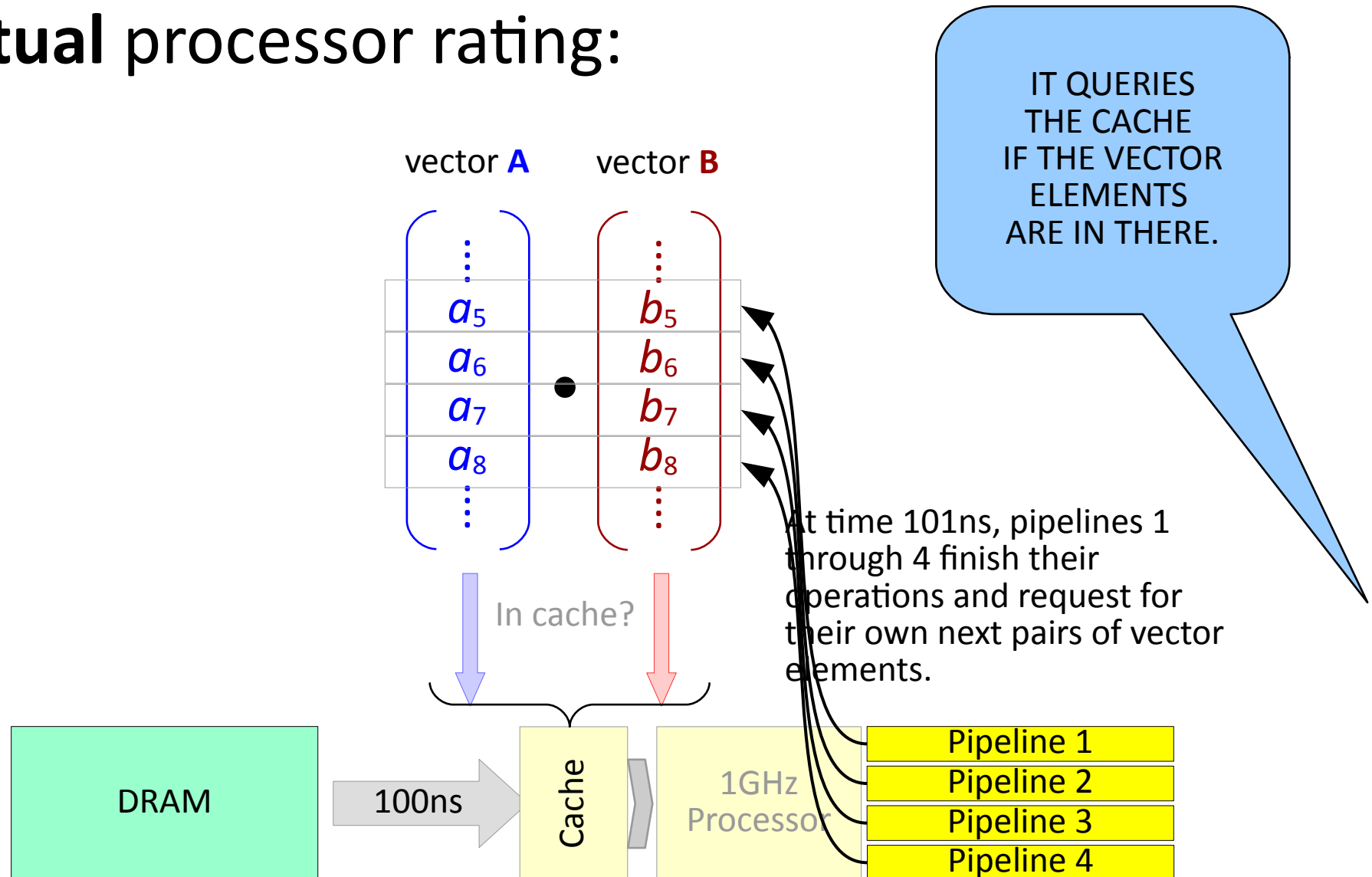
# PROCESSOR RATING

- Actual processor rating:



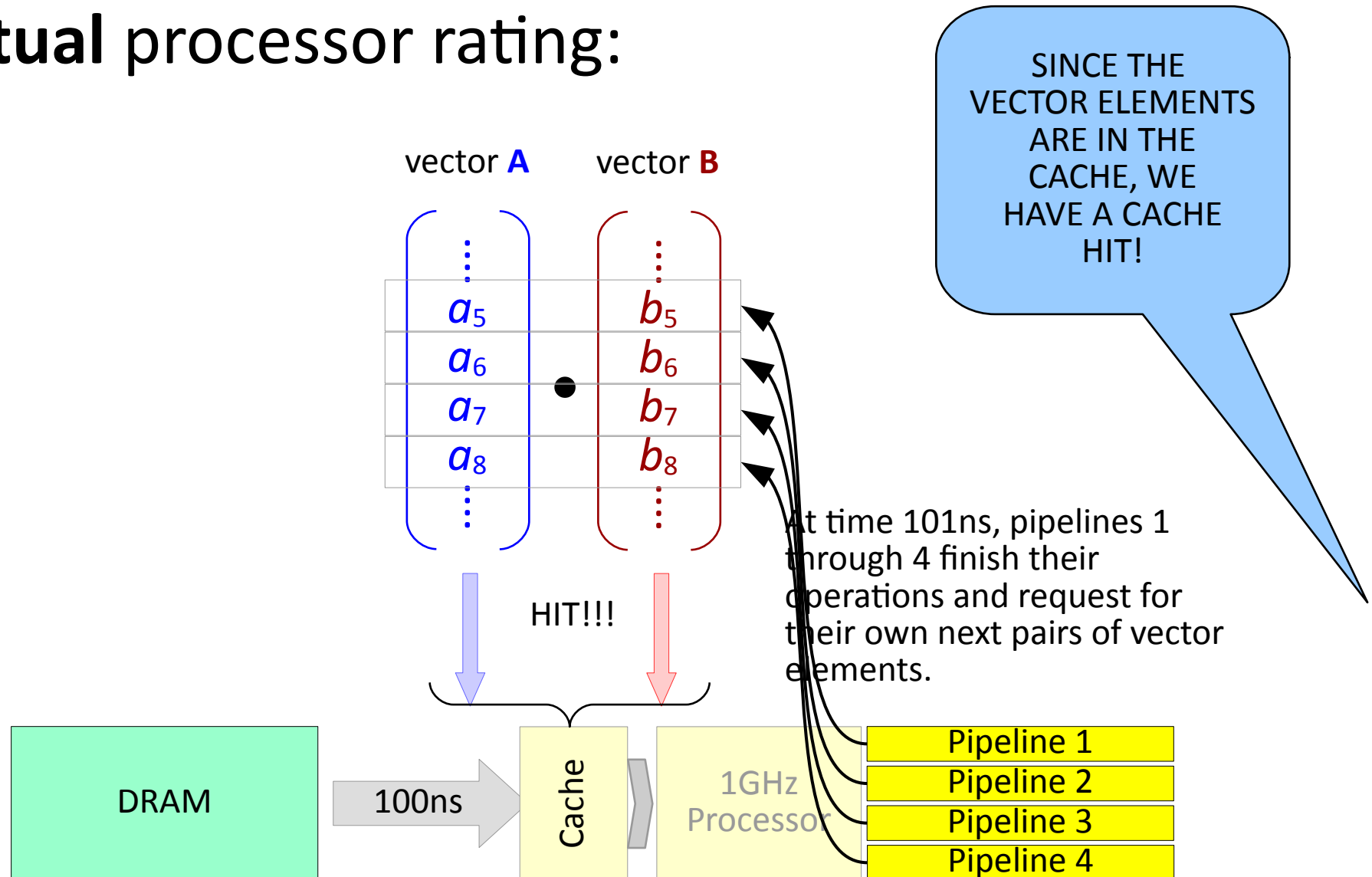
# PROCESSOR RATING

- Actual processor rating:



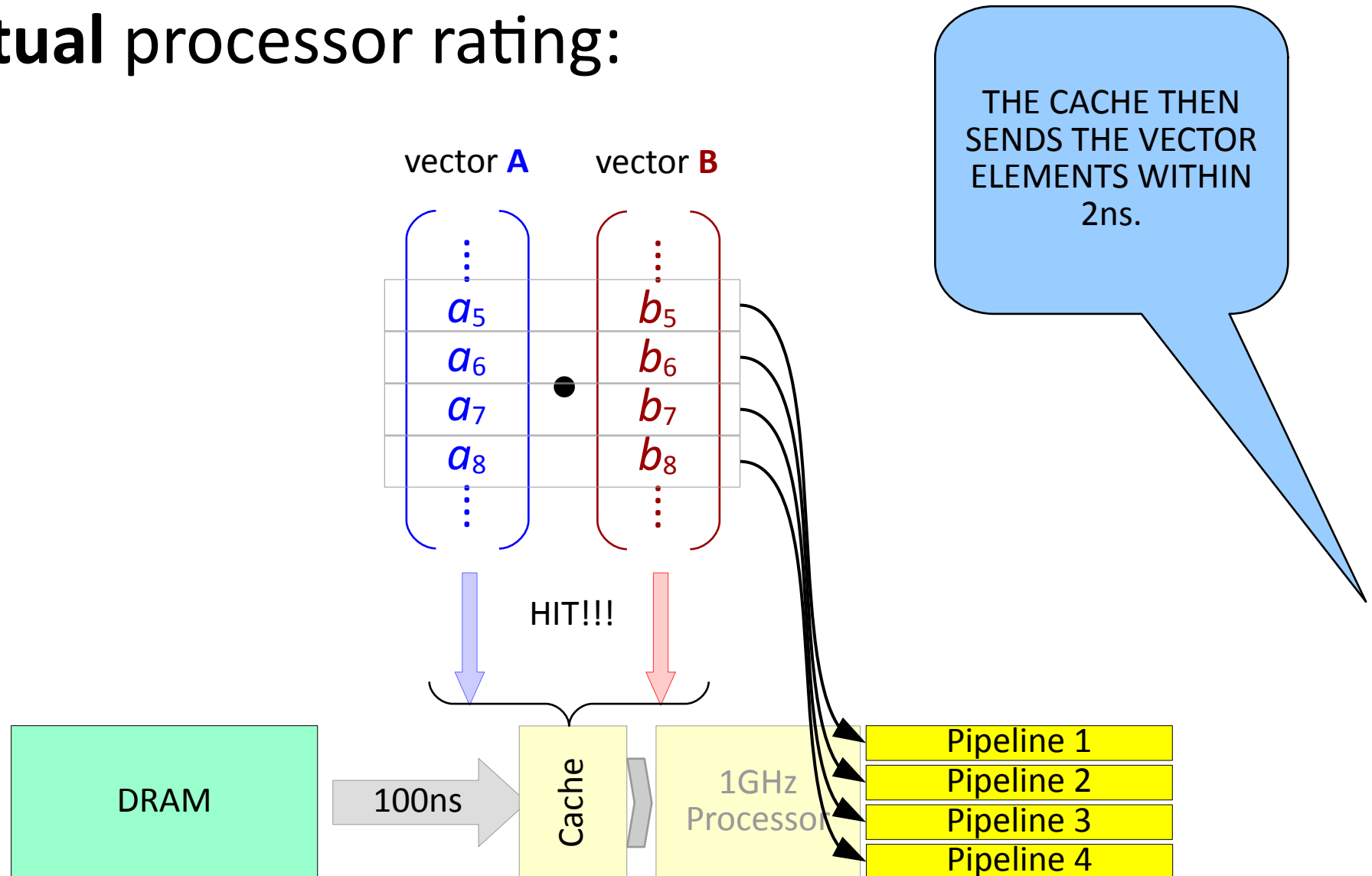
# PROCESSOR RATING

- Actual processor rating:



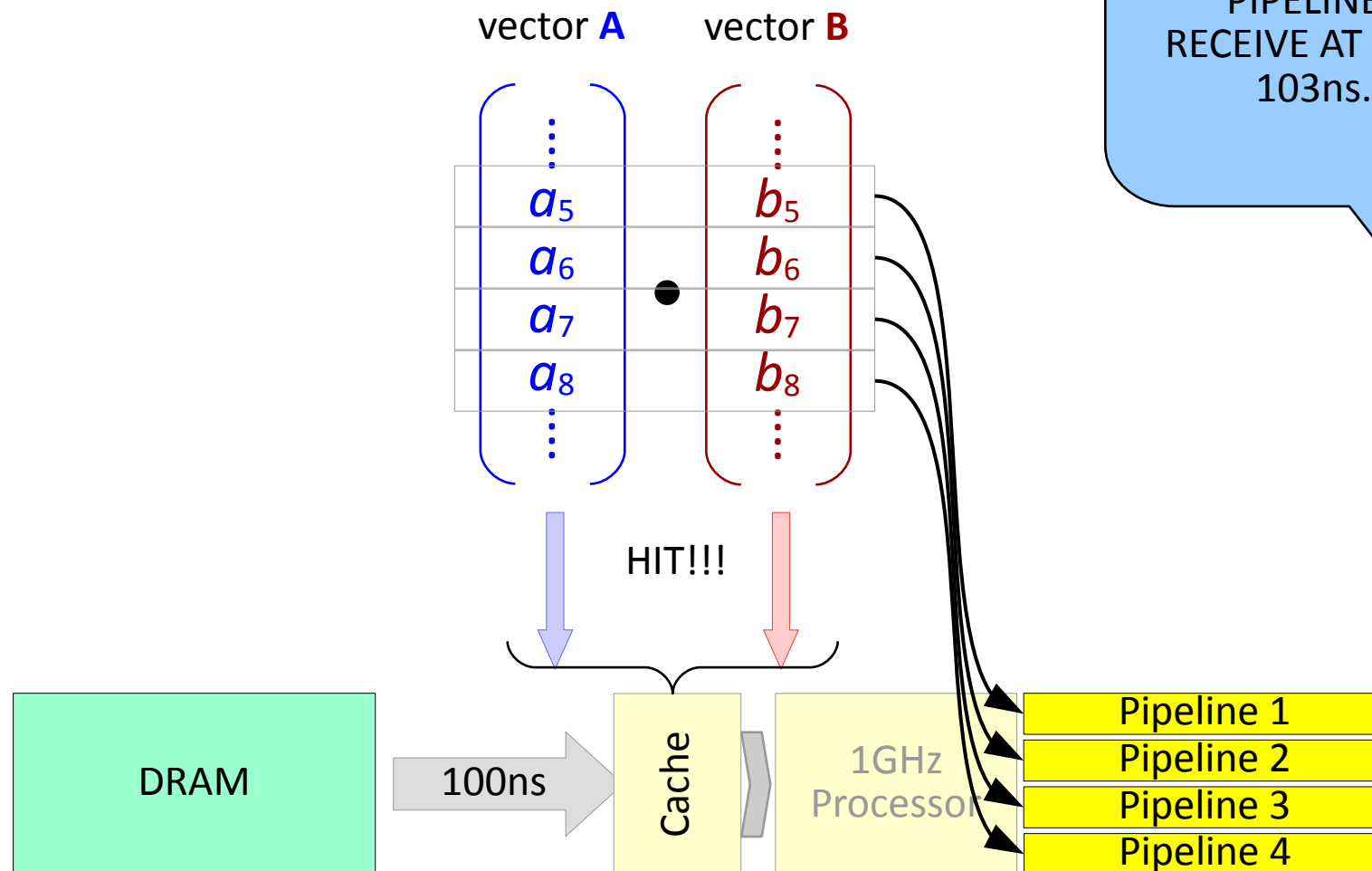
# PROCESSOR RATING

- Actual processor rating:



# PROCESSOR RATING

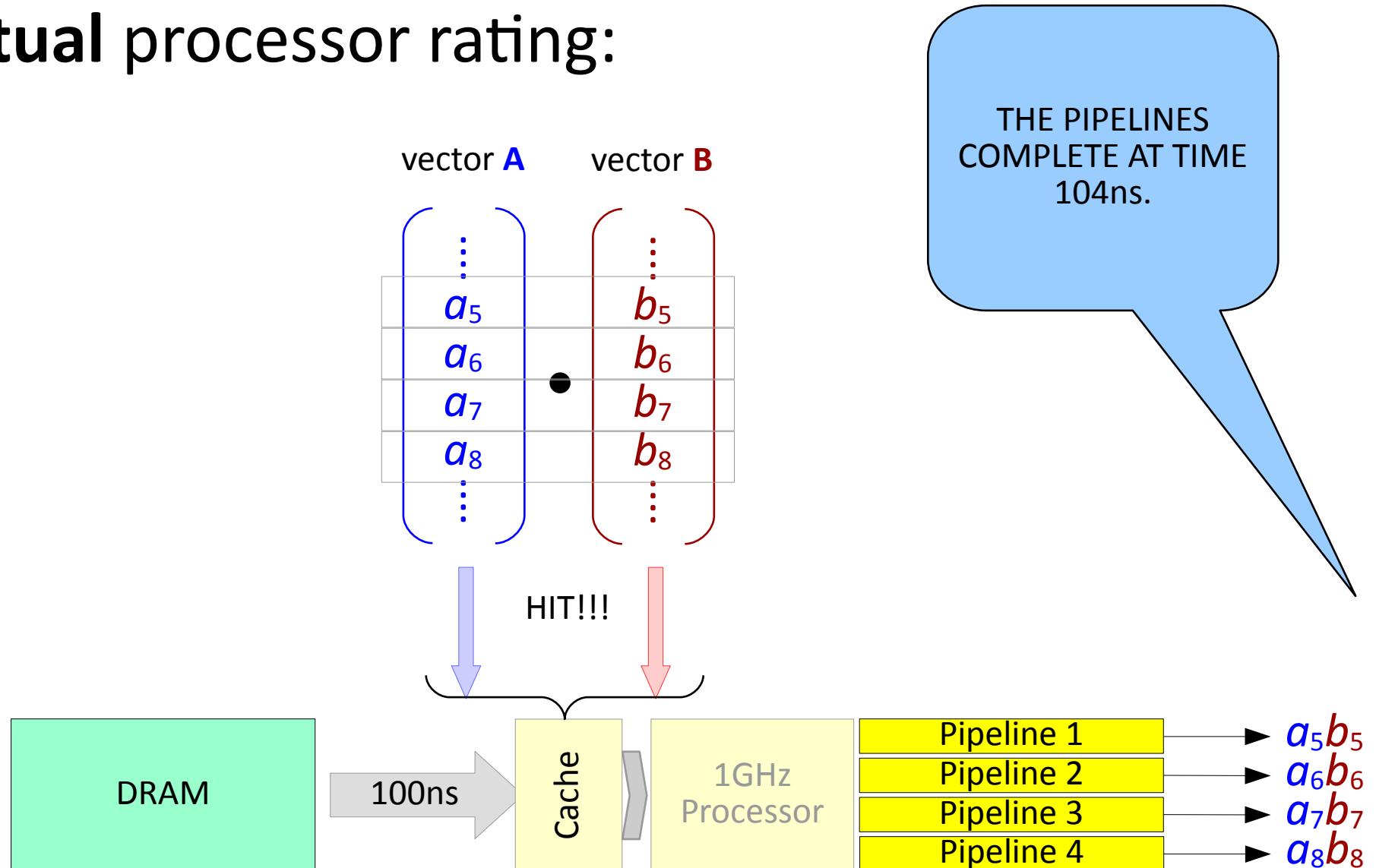
- Actual processor rating:



WHICH THE  
PIPELINES  
RECEIVE AT TIME  
103ns.

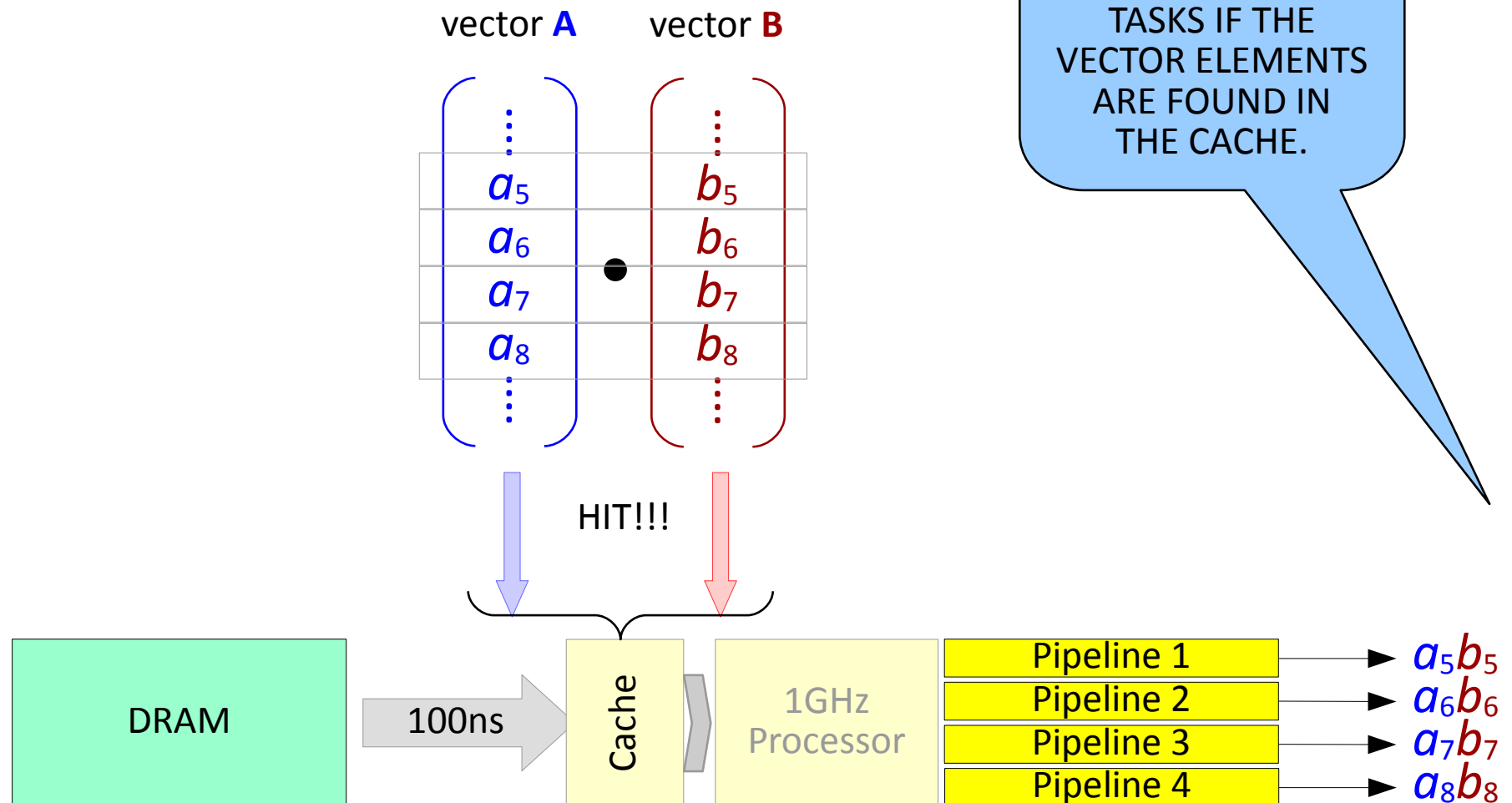
# PROCESSOR RATING

- Actual processor rating:



# PROCESSOR RATING

- Actual processor rating:



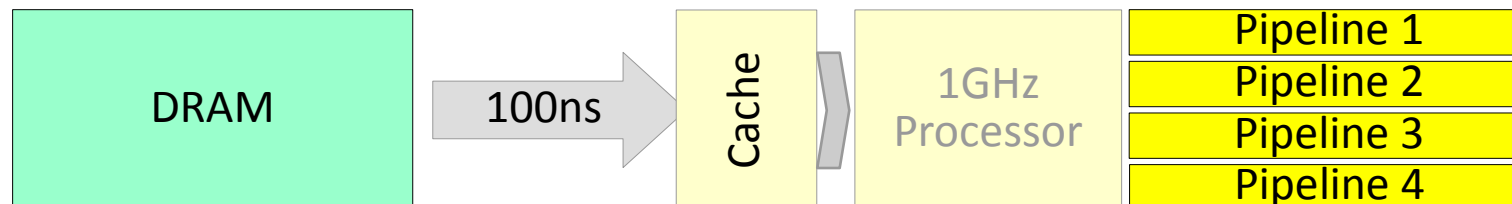


# PROCESSOR RATING

- Actual processor rating:

<u>Vector pairs</u>	<u>Cache</u>	<u>Delivery</u>	<u>Compute</u>	
1-4	Miss	100ns	1ns	110ns
5-8		2ns	1ns	
9-12		2ns	1ns	
13-16		2ns	1ns	
17-20	Miss	100ns	1ns	110ns
21-24		2ns	1ns	
25-28		2ns	1ns	
29-32		2ns	1ns	

SIXTEEN FLOATING  
POINT OPERATIONS  
EVERY 110ns.



# PROCESSOR RATING

- **Actual** processor rating:
  - ▷ So, we can finish 16 pairs of vector elements every about 110ns.
  - ▷ OR, 16 floating point operations every 110ns.
  - ▷ OR, one floating point operation every 6.875ns.
  - ▷ >145 million floating point operations in a sec.
  - ▷ >145 MFLOPS

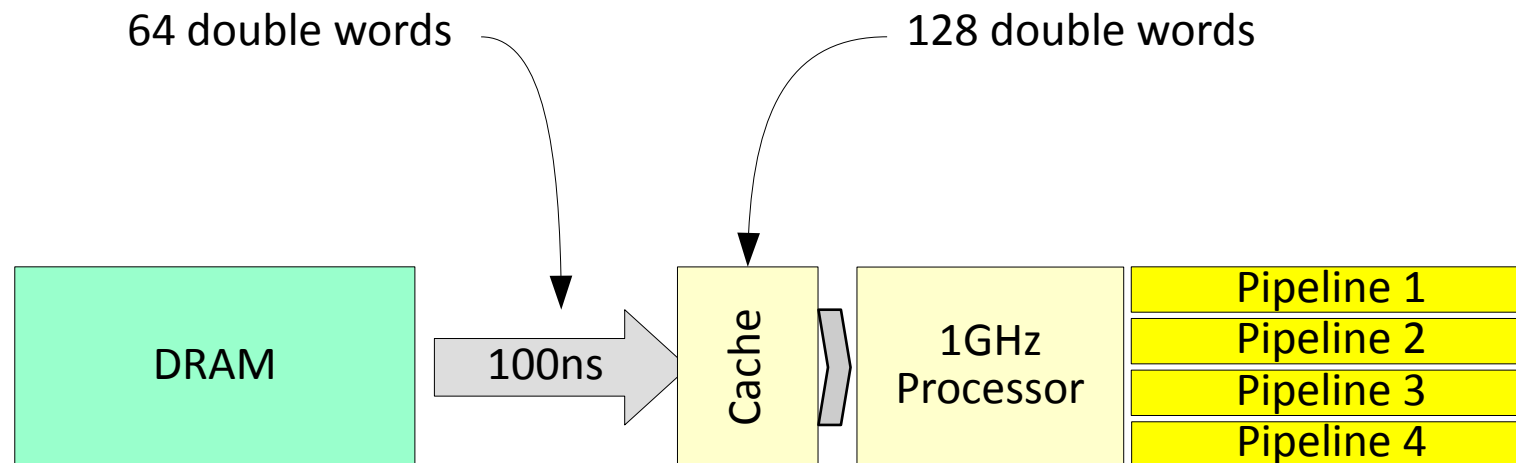
# PROCESSOR RATING

- **Comparison:**

- ▷ Theoretical/Peak Rating: 4GFLOPS
- ▷ Actual Rating: 145MFLOPS
- ▷ Only 36.25% of what we expect
- ▷ Makes the 4 pipelines useful

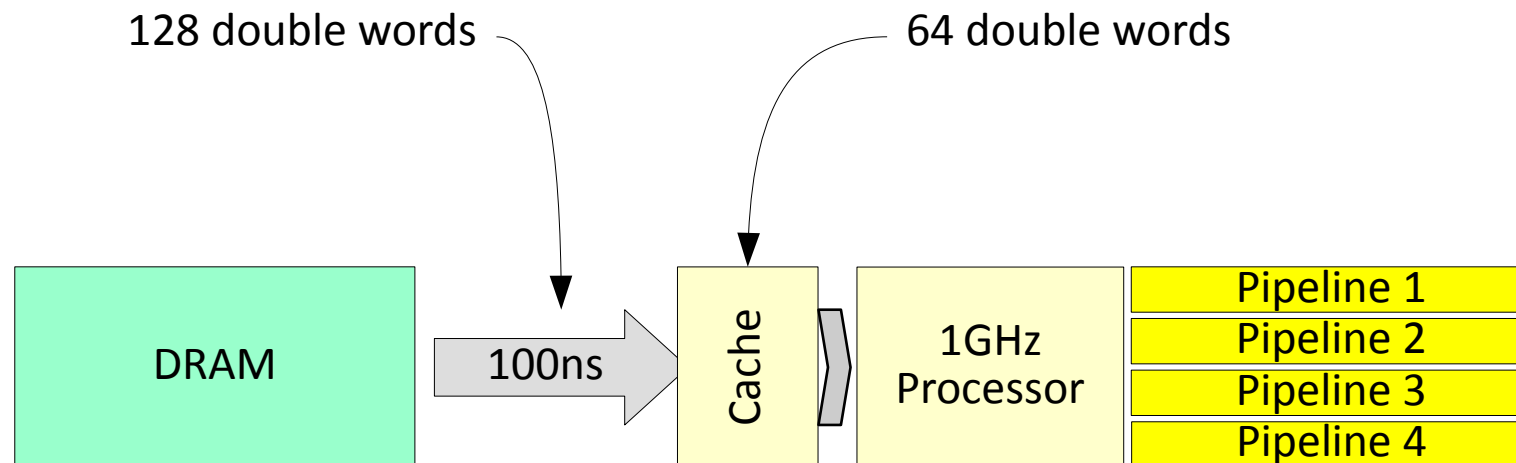
# PRACTICAL STUFF

- Double the cache but not the datapath bandwidth



# PRACTICAL STUFF

- Double the cache but not the datapath bandwidth
- Double the datapath bandwidth but not the cache



**NEXT DISCUSSION...**

**Hiding Memory Latency**