PMPP 2015/16

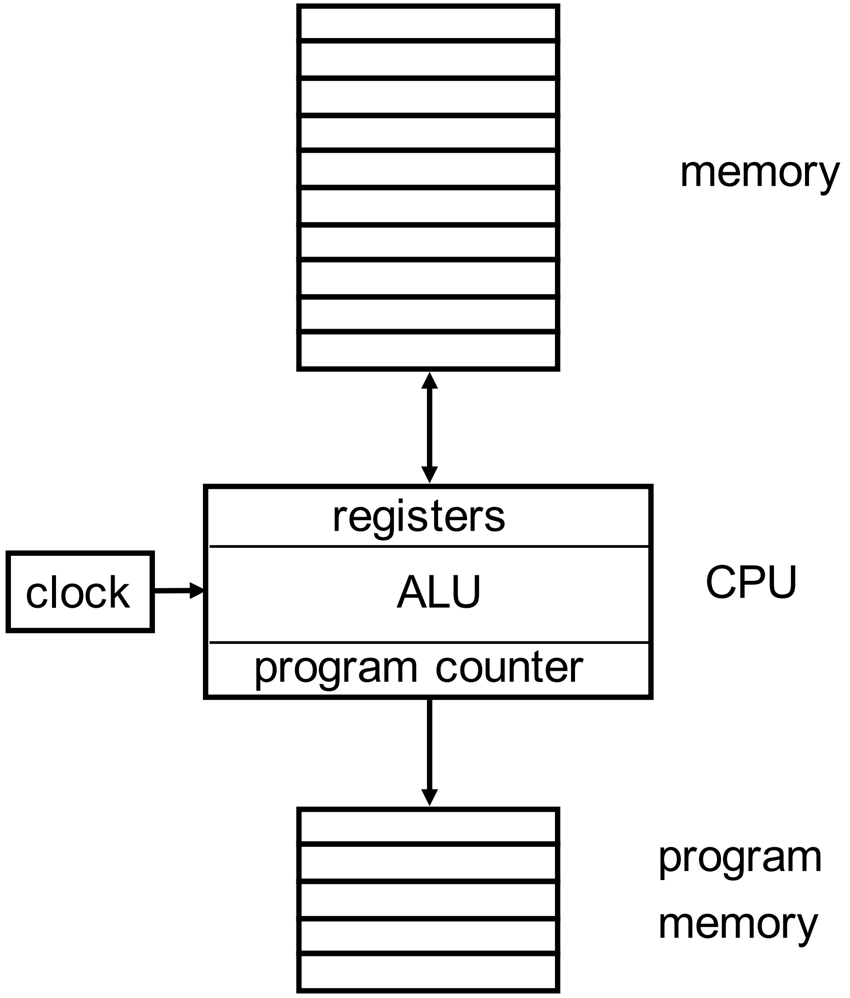
PRAM

Random Access Machine (RAM)

- von Neumann model
- program is sequence of instructions
  - memory access instructions
  - arithmetic logical instructions
  - conditional instructions

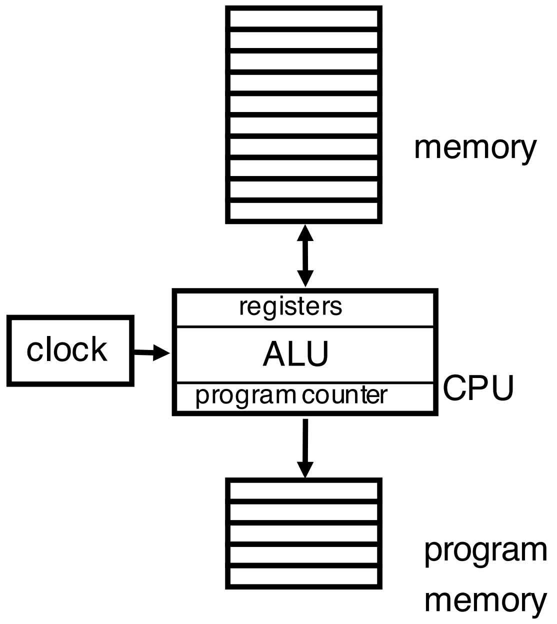
➔sequential computation

- time complexity
  - number of operations as function of size of input (in bits, words, …)

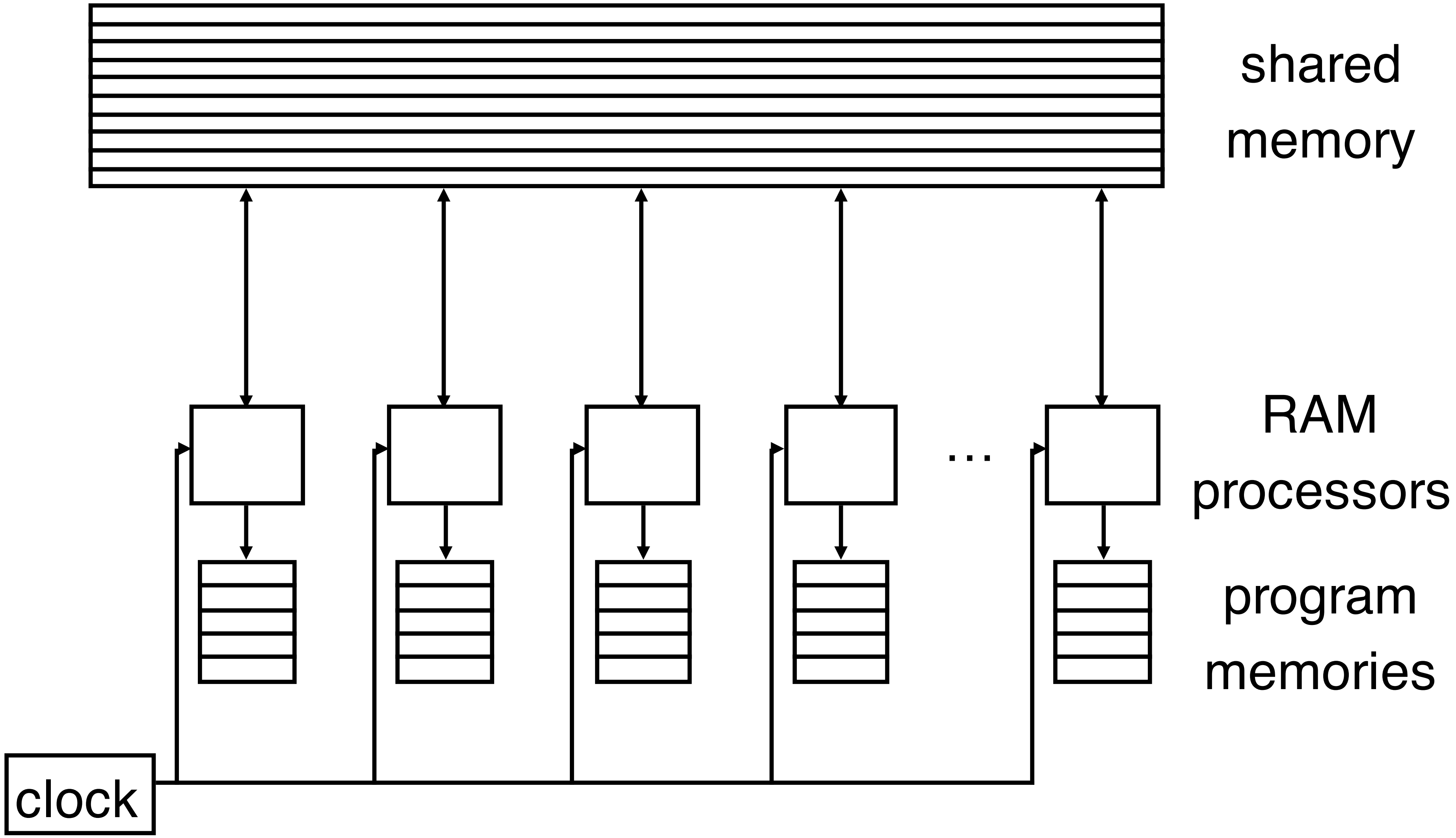


Random Access Machine (RAM)

- practical model of sequential computation given "reasonable" assumptions
  - e.g., amount of information storable in a memory cell or register
  - total memory size
- crucial assumption that each instruction is executed in unit time
  - close link between RAM and reduced instruction set computer (RISC)
  - may, e.g., be violated by cached memory
- von Neumann bottleneck
  - requires faster processors or concurrent processing



Parallel Random Access Machine (PRAM)



Parallel Random Access Machine (PRAM)

- natural generalization of RAM model

- unspecified number of identical processing units
  - unique processor ID of each processing unit
  - all processors have access to a common shared memory
  - processors are controlled by a common clock
  - synchronous execution (one instruction per time step executed by all processors)

- processors may have different programs and perform different instructions at each time step
  - MIMD model

Parallel Random Access Machine (PRAM)

- for theoretical purposes often assumption of unlimited number of processors
  - still only a finite number of processors can be active at any given time
- alternatively bounded number of processors
  - e.g., p(n)-processor PRAM, n is function of problem size
  - it can be shown that a p-processor PRAM with fixed number of processors can simulate a p(n)-processor PRAM
- sometimes shared memory is also bounded
  - (p;m)-processor PRAM is a p-processor PRAM with m shared memory cells

Parallel Random Access Machine (PRAM)

- models for concurrent memory access

- exclusive write, exclusive read (EREW)
  - no two processors may read nor write to the same memory cell within the same time step

- concurrent read, exclusive write (CREW)
  - multiple processors can read content of memory cell concurrently
  - no simultaneous writing to the same cell

- concurrent read, concurrent write (CRCW)
  - reading and writing of a cell at the same time step allowed
  - special mechanisms for concurrent write

Parallel Random Access Machine (PRAM)

- mechanisms to resolve concurrent write conflicts
- weak
  - concurrent writing only allowed for special value (e.g., 0)
- common
  - concurrent writes must all write the same value
- arbitrary
  - only one of the concurrently written values stored, others are lost
- priority
  - value of processor with highest priority (based on ID) is stored
- combining
  - concurrently written values are combined and then stored

Parallel Random Access Machine (PRAM)

- two main abstractions in the model

1. the processors work synchronously under the control of a common clock
2. all processors have access to a shared memory where they can each read or write a cell in each time step

- allows to study the maximum amount of parallelism in a problem
- shared memory main obstacle for building a PRAM in hardware

Parallel Random Access Machine (PRAM)

- some hardware approximations of a PRAM have been built, e.g. SB-PRAM, CUDA architecture

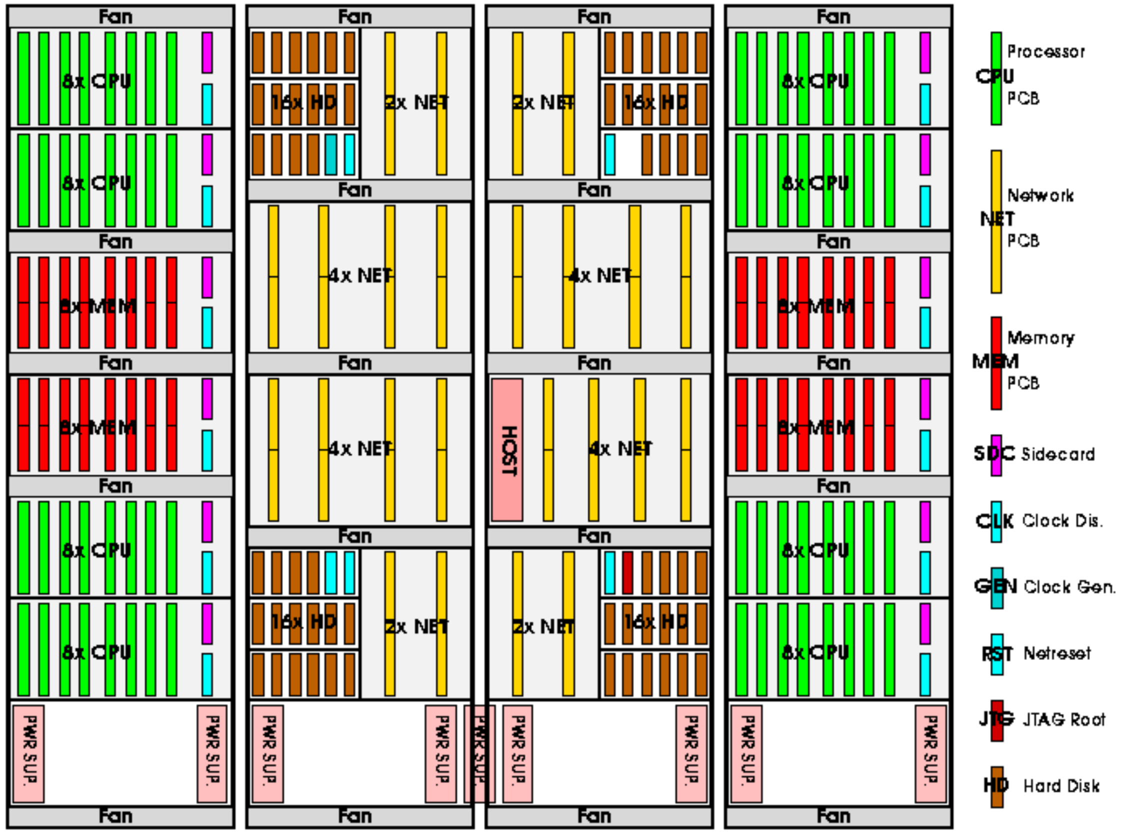
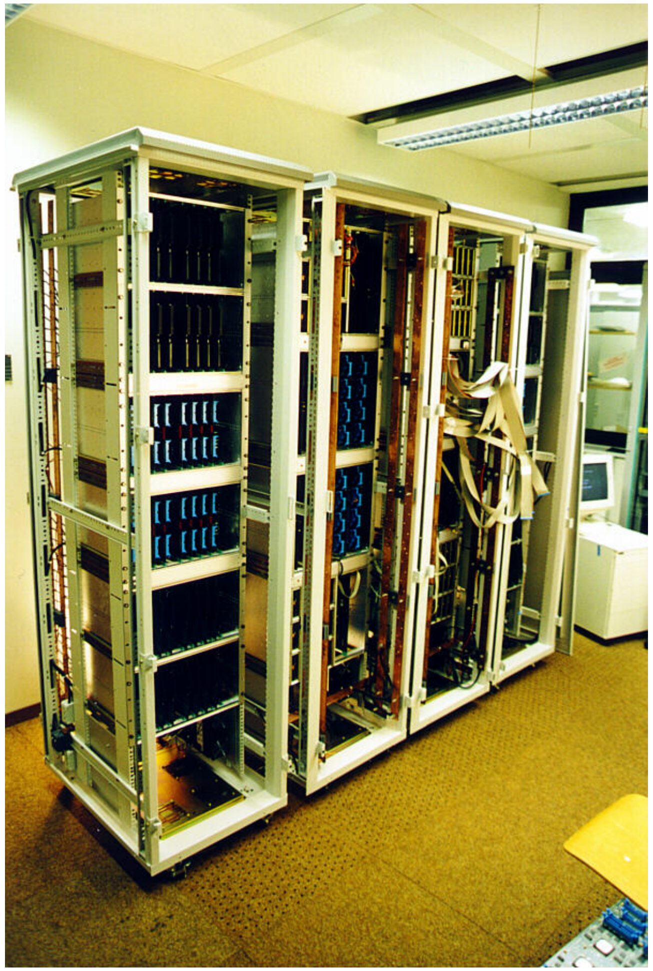


image from SB-PRAM web page

Parallel Random Access Machine (PRAM)



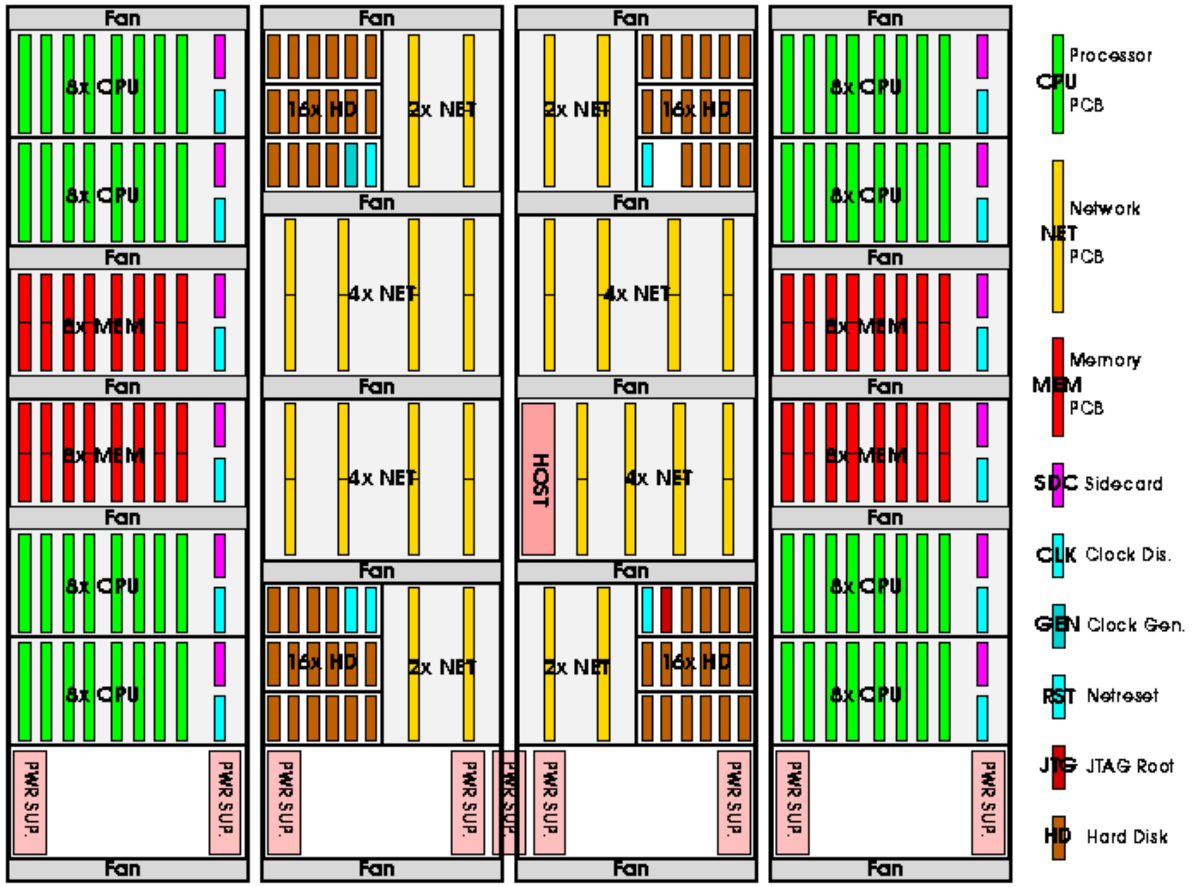


image from SB-PRAM web page

Parallel Random Access Machine (PRAM)

Some hardware approximations of the PRAM do exist, and some experience with the implementation of PRAM algorithms has been gathered, but the exact practical value of the PRAM is still a subject of controversy.

Joerg Keller, Christoph Kessler, Jesper Larsson Traeff

Practical PRAM Programming, 2001