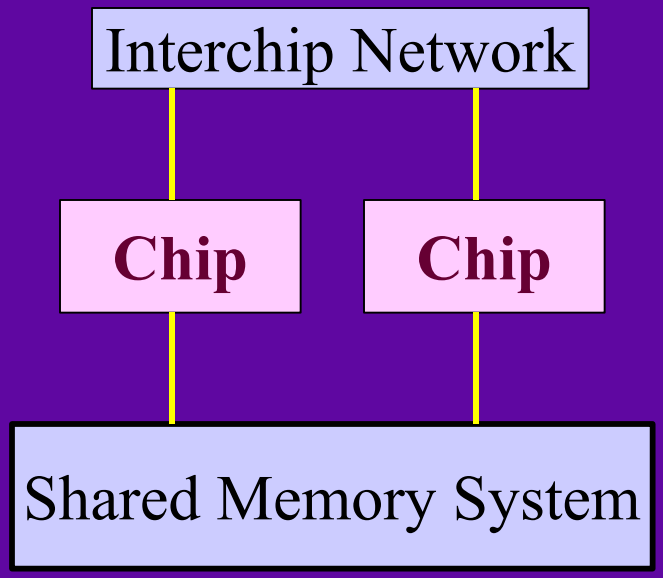
The Fresh Breeze Memory Hierarchy

Jack Dennis
MIT Computer Science
and
Artificial Intelligence Laboratory

A Fresh Breeze System

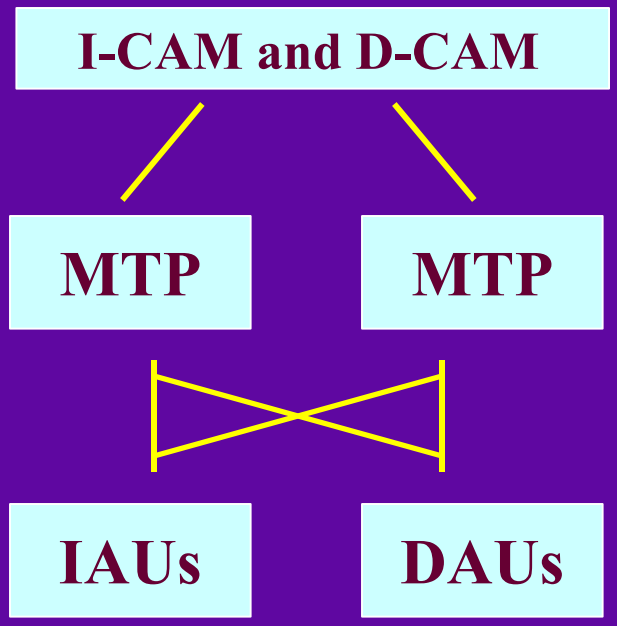


- A scalable multiprocessor system.
- Intended to support the functional programming style for implicit parallelism and programmabiltiy.
- Intended to achieve high performance with low power consumption.

Ideas

- Simultaneous Multithreading: Improves latency tolerance and function unit ultilization.
- Global Shared Memory: Reduces complexity of thread switching and communication; eliminates need for distinction between memory and files.
- Fixed-Size Chunks: Simplifies memory management.
- No Update: Chunks are created, accessed, and released; no multiprocessor consistency problem.
- Cycle-Free Heap: Parallel reference count garbage collection of chunks may be used.

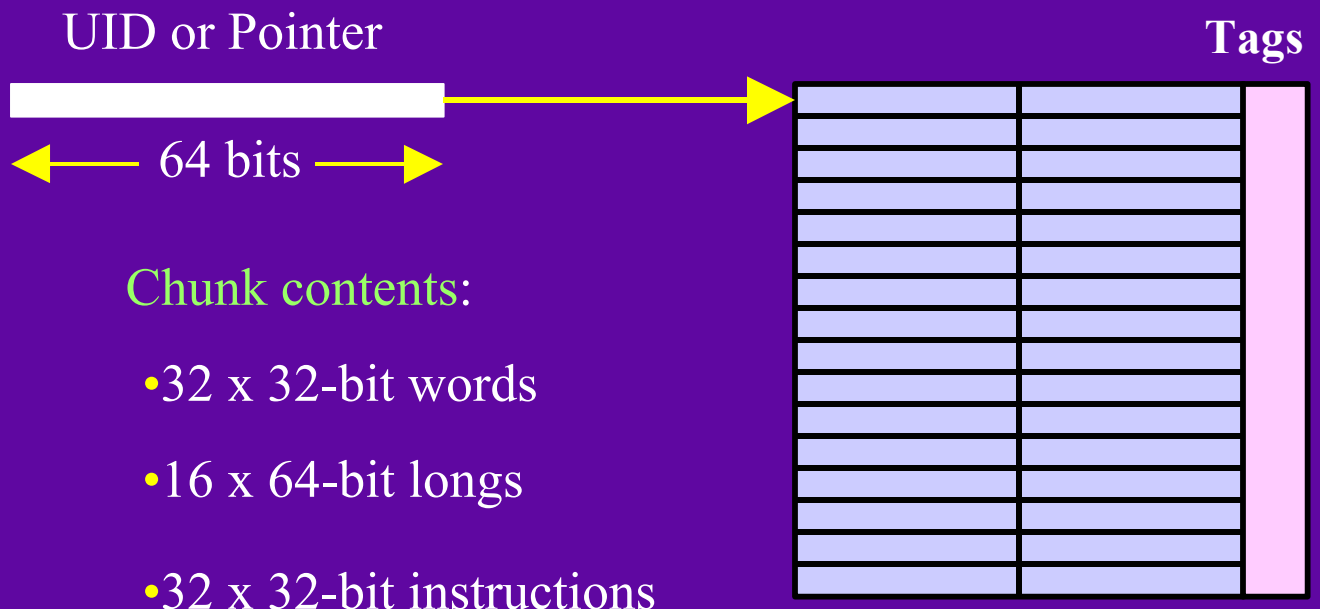
Fresh Breeze Multiprocessor Chip



- Active chunks are held in on-chip IAUs and DAUs. Chunks are analogous to I-cache and D-cache data lines.
- MTPs: Multithreaded Processors.
- I-CAM and D-CAM are associative directories of chunks held in IAUs and DAUs; analogous to cache address lines and TLB, but shared by all MTPs.

Memory Chunks and UIDs

- Chunk: A fixed-size unit of memory allocation. 1024 bits of data;

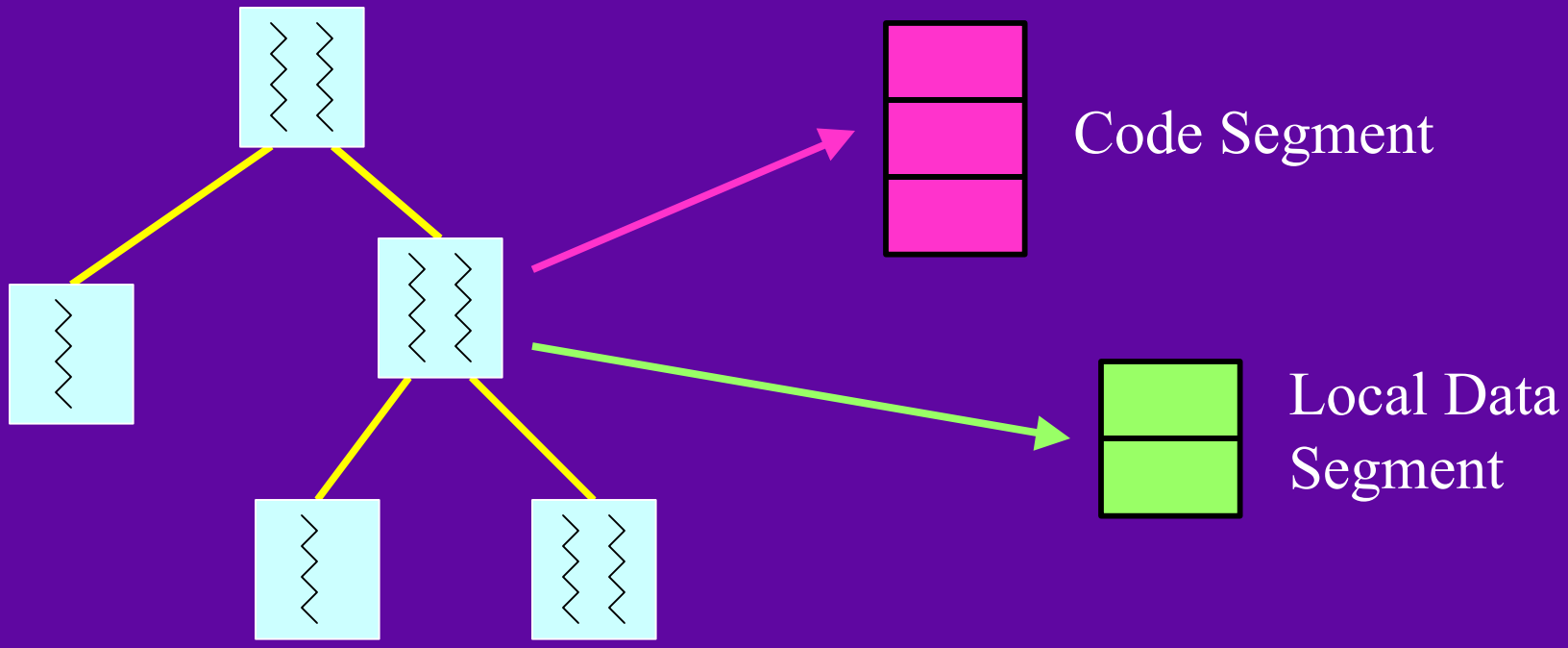


Chunk contents:

- 32 x 32-bit words
- 16 x 64-bit longs
- 32 x 32-bit instructions

Program Execution

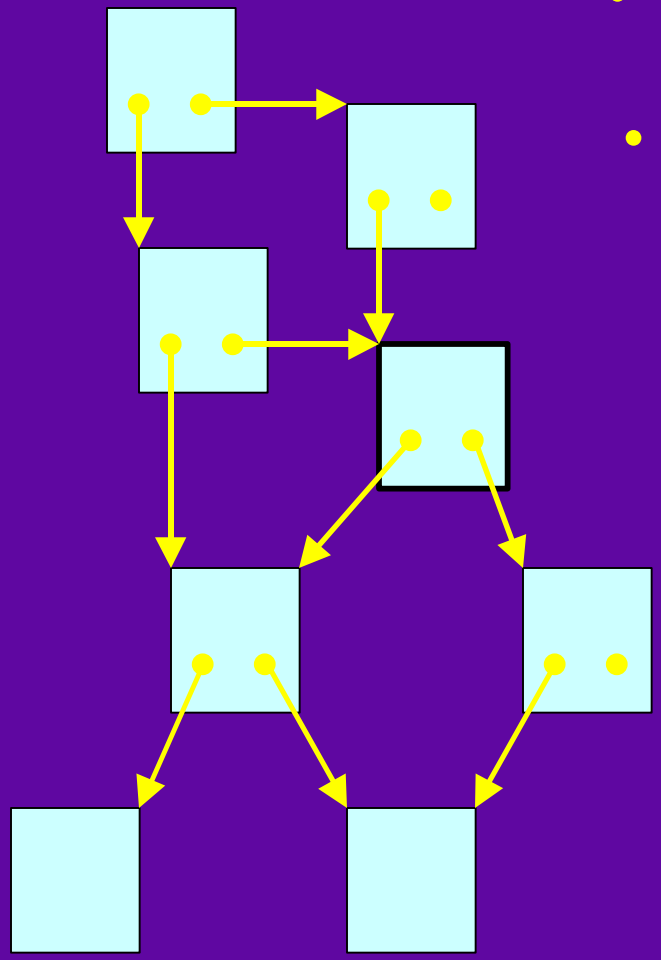
- A program in execution is a tree of method activations (similar to Monsoon).



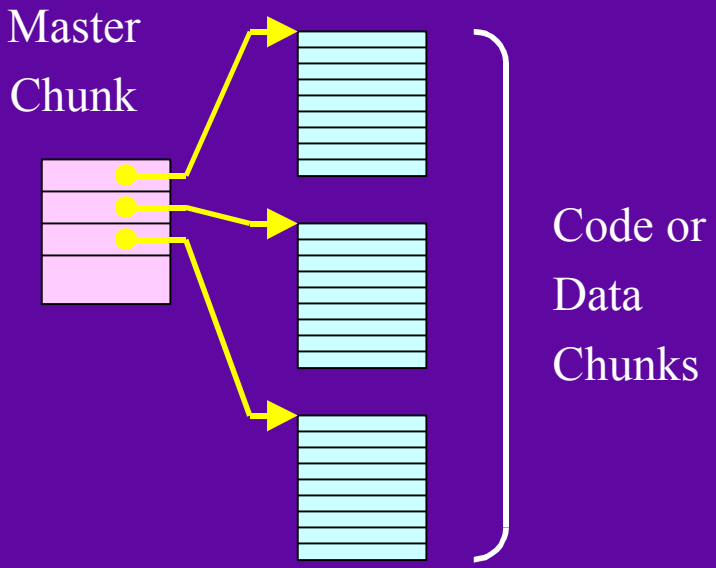
Several threads may be active in each method activation.

Data Structures

- Cycle-Free Heap



- Fan-out as large as 16
- **Code Segment or Local Data Segment**



- Arrays: Three levels yields 4096 elements (longs)

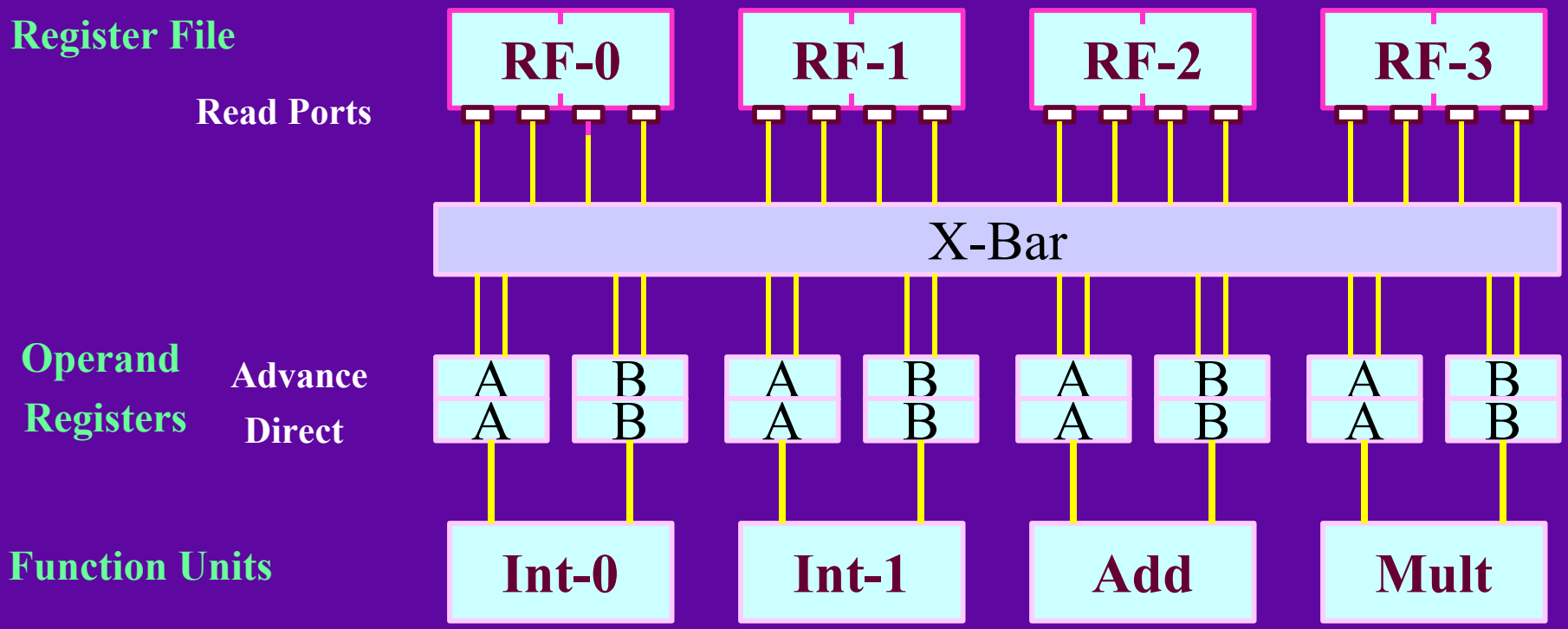
Memory Hierarchy

- Register File:
  - 32 x 32-bit words per Activity
- Code and Data Access Units:
  - 1024-bit Code and Data Chunks shared by all MTPs on chip. About 2^{14} chunks. Like cache lines.
- Shared Memory System:
  - Repository for code and data chunks. Up to 2^{64} chunks.

Register File

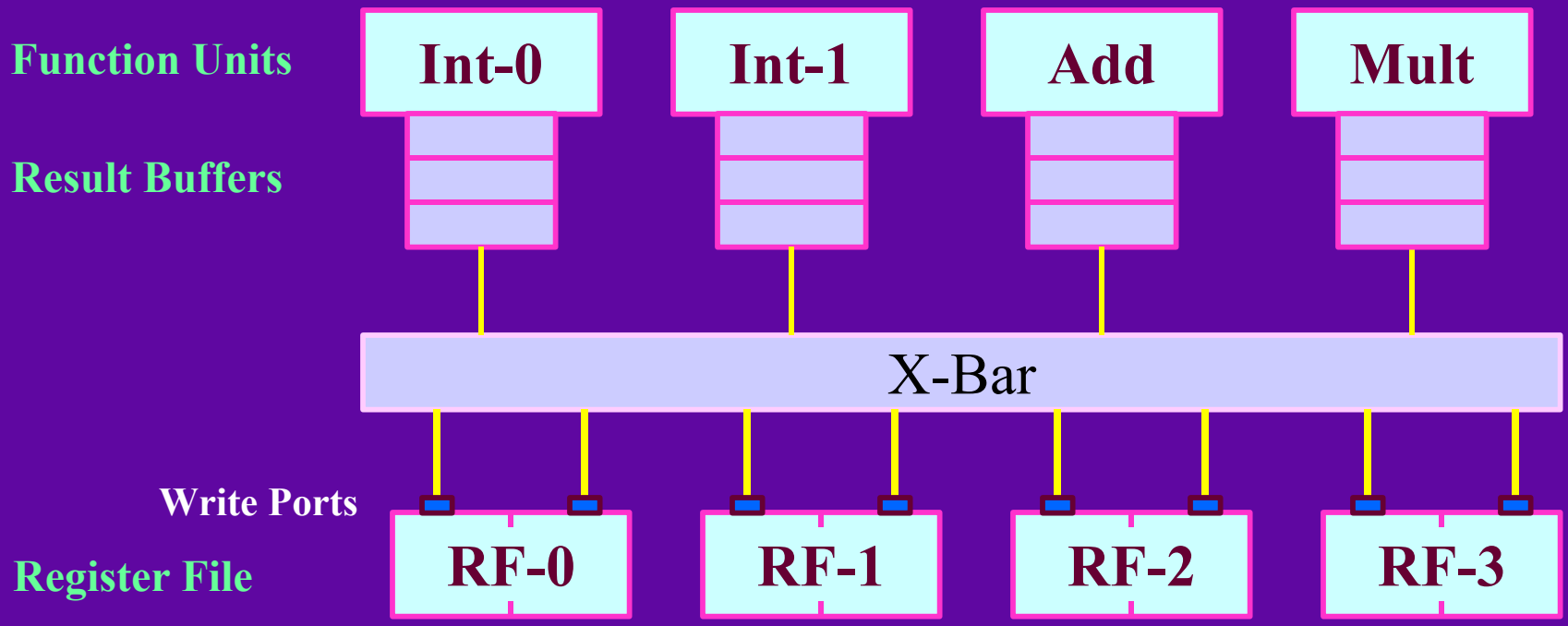
- 32 x 32-bit words per Activity
- Words are paired for longs or UIDs
- Four sections, each having two banks for left and right (even/odd) words.
- Total of 16 words per bank (four per activity).
- Two read ports, one write port per bank.
- Total bandwidth is 4 x 2 x 2 = 16 reads (32-bit) per cycle; 8 (32-bit) writes per cycle.

Superscalar Operation I
Operand Access



- Port 0 supplies Operand A; Port 1 supplies Operand B.
- Port 3 is used to read registers for store transfers.

Superscalar Operation II
Result Writeback

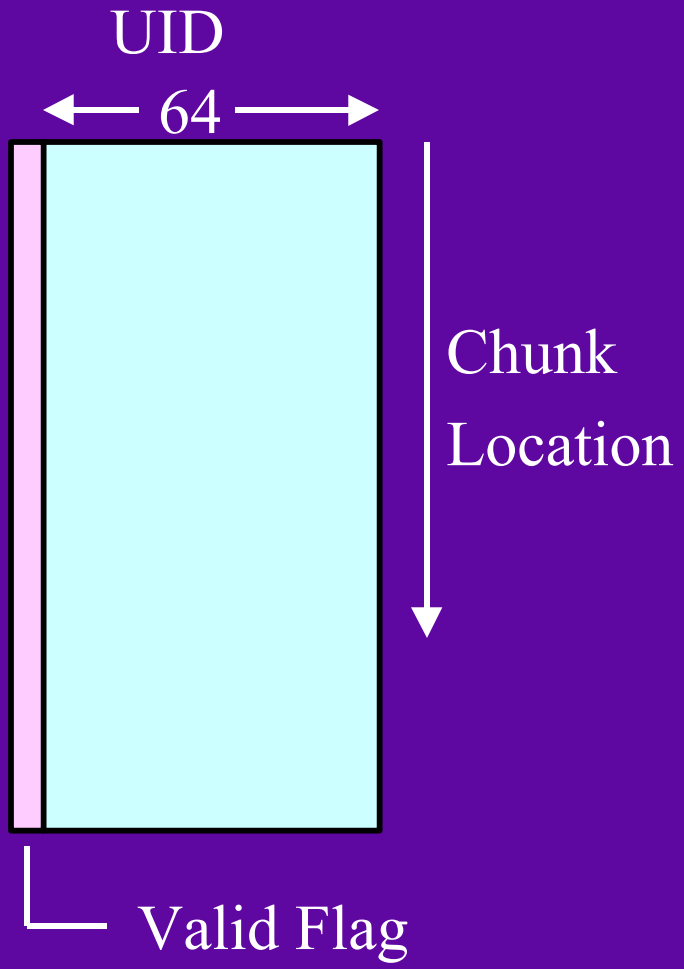


- Bypass Operands may be selected from the result buffers.
- Load transfers share the write ports of all Register File sections and have priority over writebacks

Data Movement

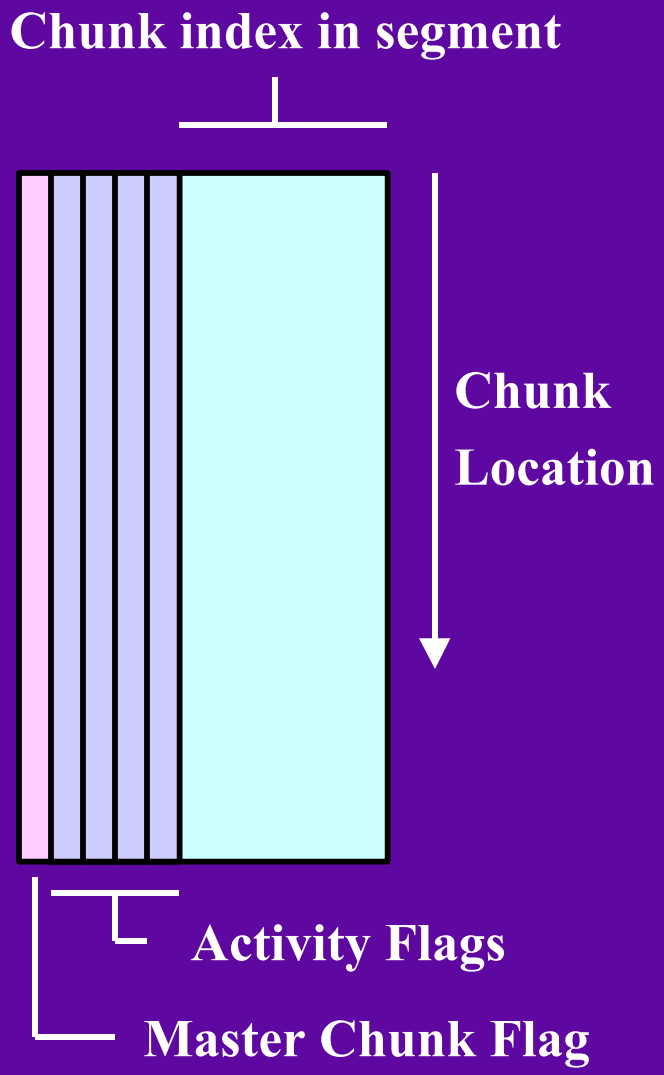
- Between Registers and Access Units
  - Programmed as multi-word loads and stores
  - X-Bar switch permits one transfer per MTP simultaneously, if no conflicts
- Between Access Units and the Shared Memory System
  - Complete Chunk is the unit of transfer.
  - Demand “Paging” using built-in LRU replacement of chunks in Access Units.
  - Supported by on-chip Activity (Thread) Scheduler.

Chunk Directories
I-CAM and D-CAM



- Content-Addressed memory; 2^{14} entries.
- Shared by all MTPs on Chip using an arbitrated bus.
- Short-cuts needed to minimize number of accesses and decrease average latency of **access**.

Mapping Short-Cuts



- A Mapping table is maintained separately by each MTP for the chunks of the code segment and local data segment of each Activity.
- The Map is consulted before requesting service from the I-CAM or D-CAM.
- Each UID entry in the Register File has an associated location auxiliary field.
- After the first reference using a UID from a register, subsequent references are made without consulting the D-CAM.

Conclusion

- The Fresh Breeze project combines of a lot of ideas, old and new.
- We believe it addresses the challenges of today's technological and usage environment.
- Many design choices have been made with little ability to anticipate their effects on performance and programmability.
- We look forward with excitement and trepidation to the results of simulation trials programming experiments.

It should be fun!