Memory Architectures

Memory

- Memory growing relentlessly
- PCs started with 16K
  - 640K max
  - current 4G norm
- Address bus size?
  - 32 not enough
  - 64 possible
  - 128 is overkill

What is Memory?

Hardware engineer:

> "Memory is a chip in which you can keep bits of data. There are really two kinds: ROM and RAM. These, in turn, come in two varieties each. There is masked programmed ROM and programmable devices, which you can program yourself. RAM may be static, which is easy to use, but has less capacity; dynamic is denser, but needs support circuits."

What is Memory?

Software engineer:

> "Memory is where you run your program. The code and data are read off of the disk into memory, and the program executed. You do not need to worry too much about the size, as virtual memory is effectively unlimited."

What is Memory?

Embedded systems programmer:

> "Memory comes in two varieties: ROM, where you keep code and constants, and RAM, where you keep the variable data [but which contains garbage on startup]."

What is Memory?

C compiler designer:

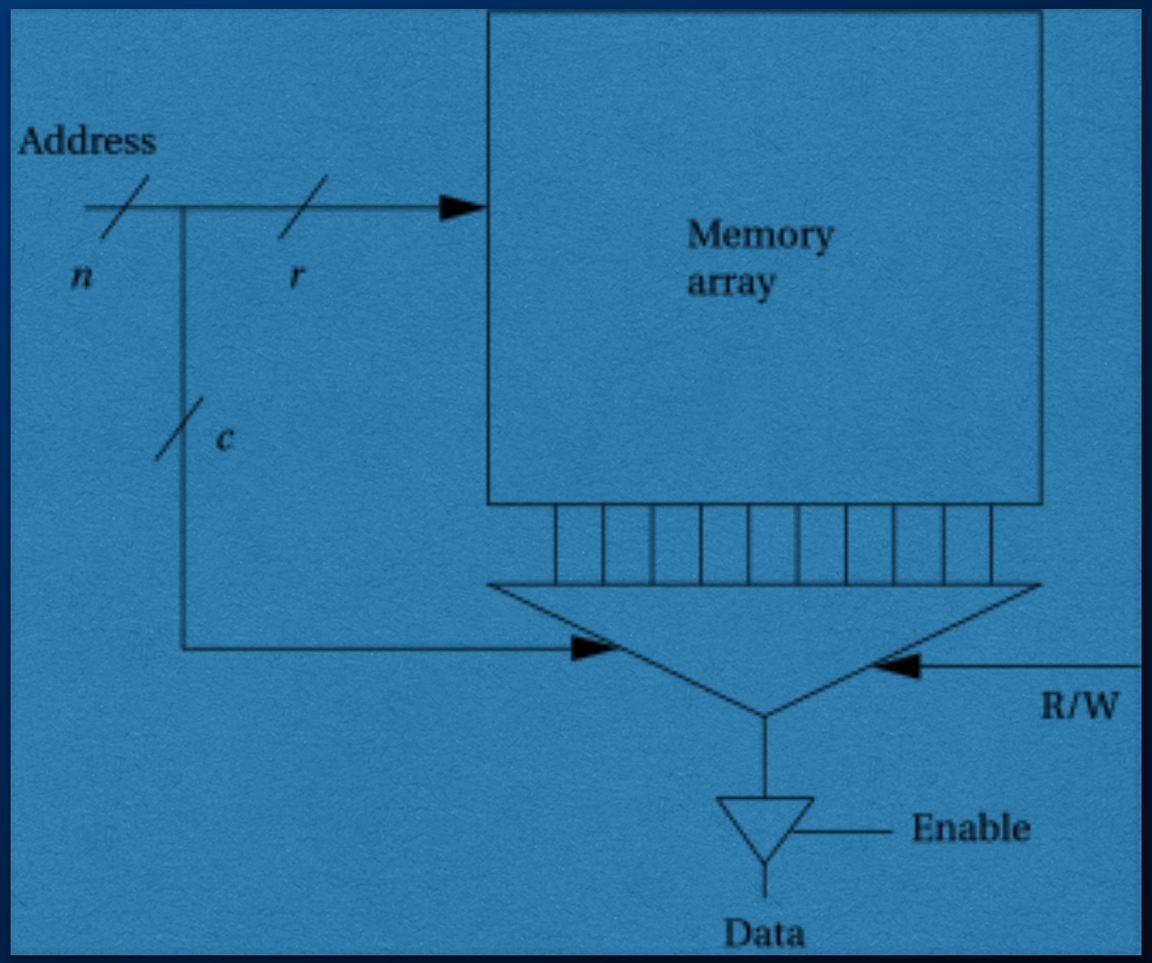
> "There are lots of kinds of memory: there is some for code, variable data, literals, string constants, initialized statics, uninitialized statics, stack, heap, some is really I/O devices, and so forth.

ROMable Code

- Code will execute correctly from ROM
  - no copy to RAM necessary
  - but RAM may be faster
- Code and data must not be mixed
  - except for constant data
- Compiler/linker should accommodate these requirements

Memory components

- Several different types of memory:
  - DRAM.
  - SRAM.
  - Flash.
  - MRAM、PCRAM和ReRAM
- Each type of memory comes in varying:
  - Capacities.
  - Widths.



Random-access memory

- Dynamic RAM is dense, requires refresh.
  - Synchronous DRAM is dominant type.
  - SDRAM uses clock to improve performance, pipeline memory accesses.
- Static RAM is faster, less dense, consumes more power.

Read-only memory

- ROM may be programmed at factory.
- Flash is dominant form of field-programmable ROM.
  - Electrically erasable, must be block erased.
  - Random access, but write/erase is much slower than read.
  - NOR flash is more flexible.
  - NAND flash is more dense.

Flash memory

- Non-volatile memory.
  - Flash can be programmed in-circuit.
- Flash can be electrically erased and reprogrammed.
- Light, compact, energy efficient and less expensive.

Flash writing

- Write is much slower than read.
  - 1.6 µs write, 70 ns read.
- Blocks are large (approx. 1 Mb).
- Writing causes wear that eventually destroys the device.
  - Modern lifetime approx. 1 million writes.

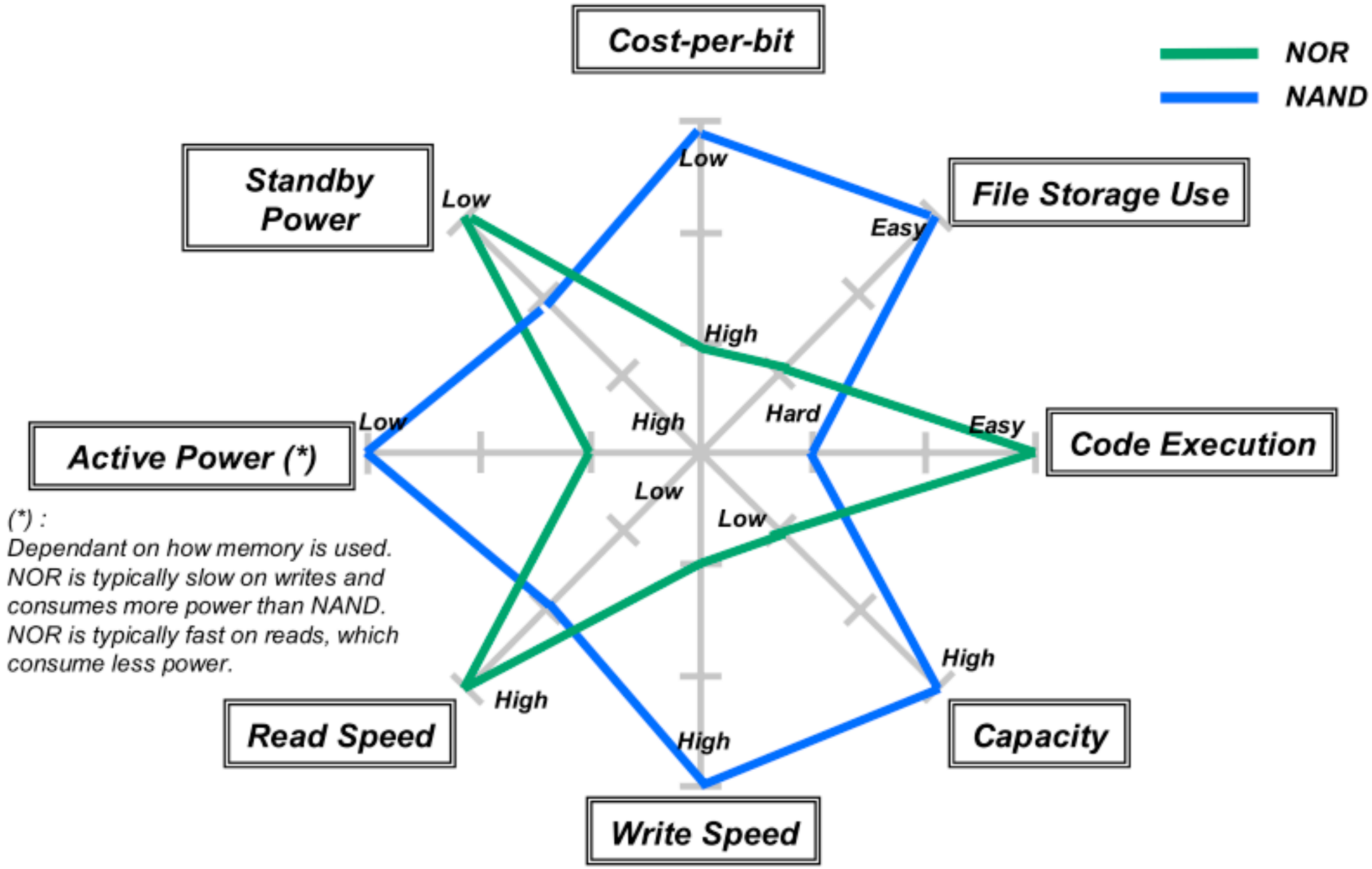
Types of flash

- NOR:
  - Word-accessible read.
  - Erase by blocks.
- NAND:
  - Read by pages (512-4K bytes).
  - Erase by blocks.
- NAND is cheaper, has faster erase, sequential access times.

Distinction between NOR and NAND flash

Attribute	**NAND**	**NOR**
Main Application	File storage	Code execution
Storage capacity	High	Low
Cost per bit	Better	
Active Power	Better	
Standby Power		Better
Write Speed	Good	
Read Speed		Good

https://en.wikipedia.org/wiki/Flash_memory#Distinction_between_NOR_and_NAND_flash

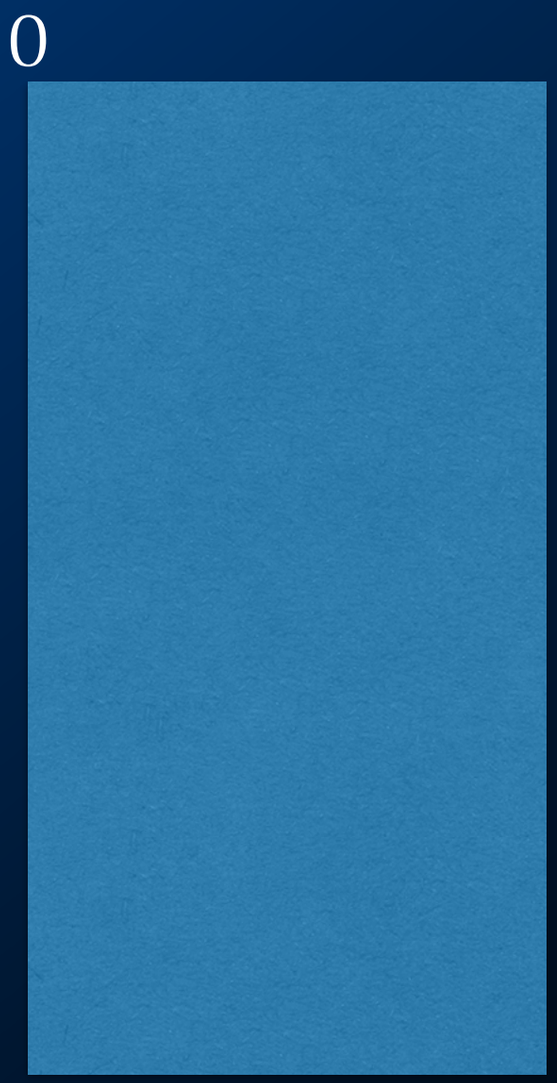

Cost-per-bit
NOR
NAND
Standby Power
Low
Low
File Storage Use
Easy
High
Active Power (*)
Low
High
Hard
Easy
Code Execution
Low
Low
(*) :
Dependant on how memory is used.
NOR is typically slow on writes and
consumes more power than NAND.
NOR is typically fast on reads, which
consume less power.
High
High
Read Speed
High
Capacity
Write Speed

Memory Architectures

- Flat single-space
- Segmented
- Bank-switched
- Multiple-space
- Virtual

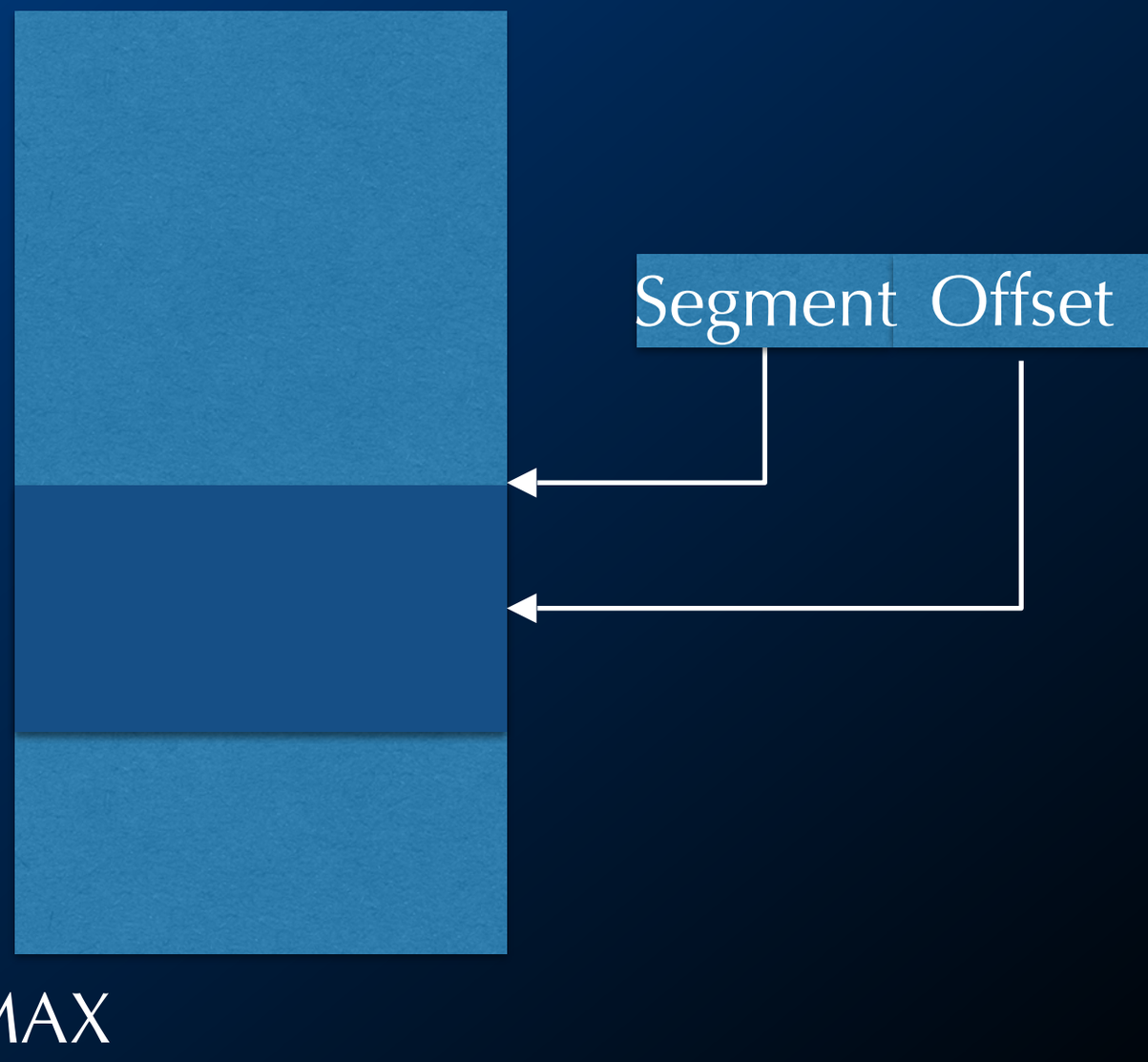
Flat Single-space Memory

- Simple
- Examples: 68K, Z80
- Space may be discontinuous
- Assumed by C
- Care with address 0



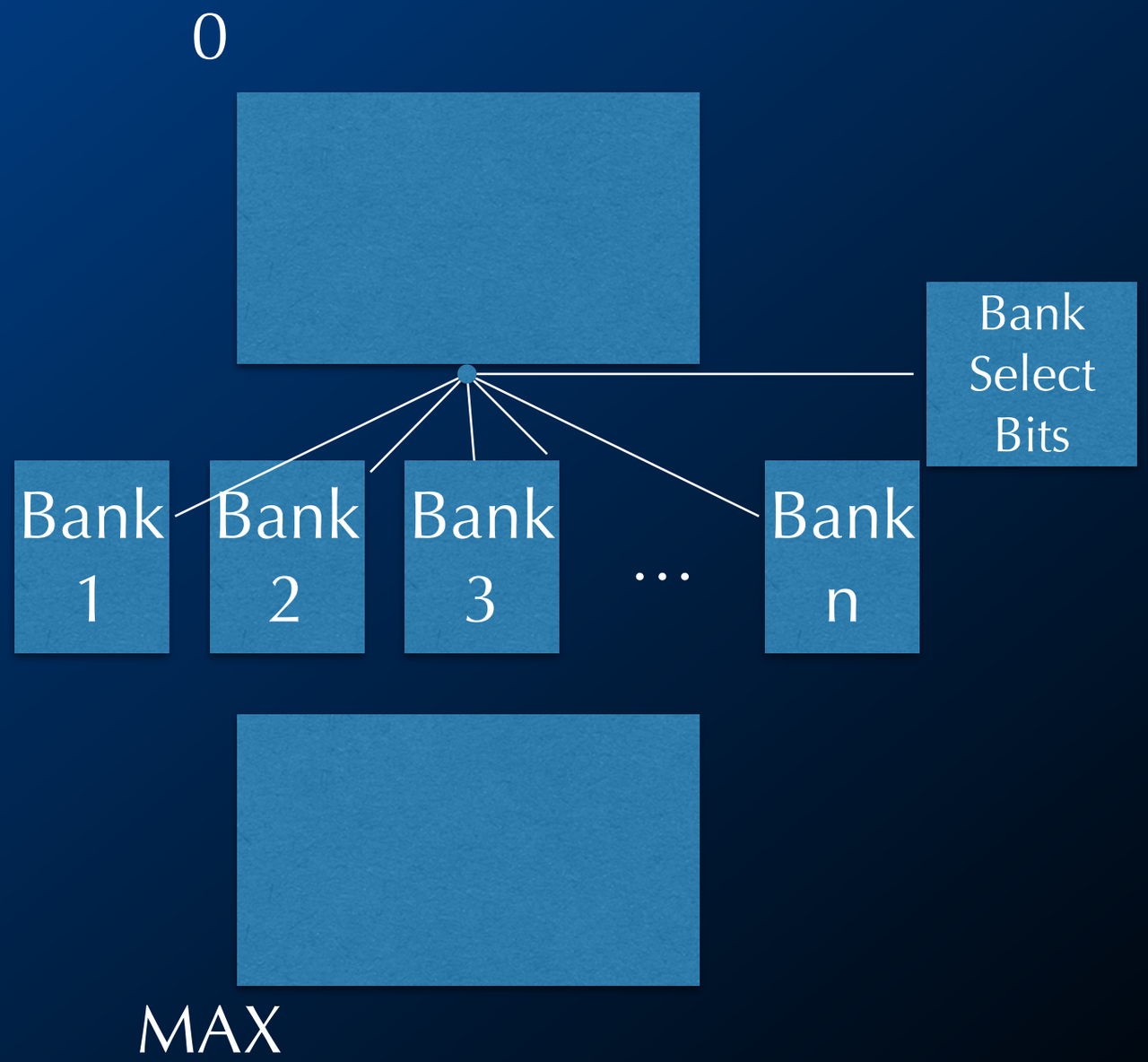
Segmented Memory

- Increased address space
- Example: Intel x86
- 2 part address:
  - segment
  - offset
- Need C extension:
  - near and far

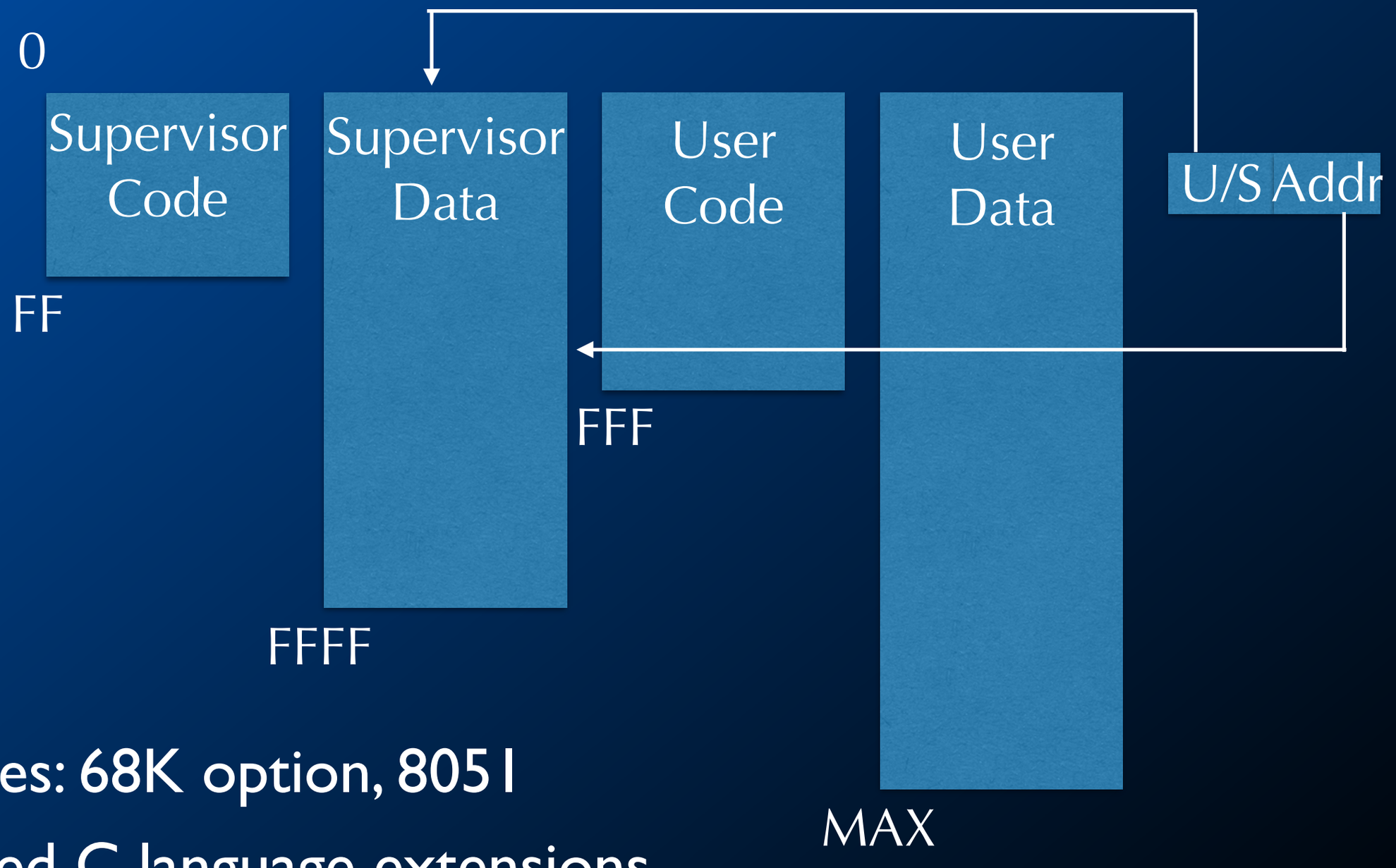


Bank-switched Memory

- Can be added to any processor
- Window into larger address space
- Linker support useful



Multiple-space Memory



- Examples: 68K option, 8051
- May need C language extensions

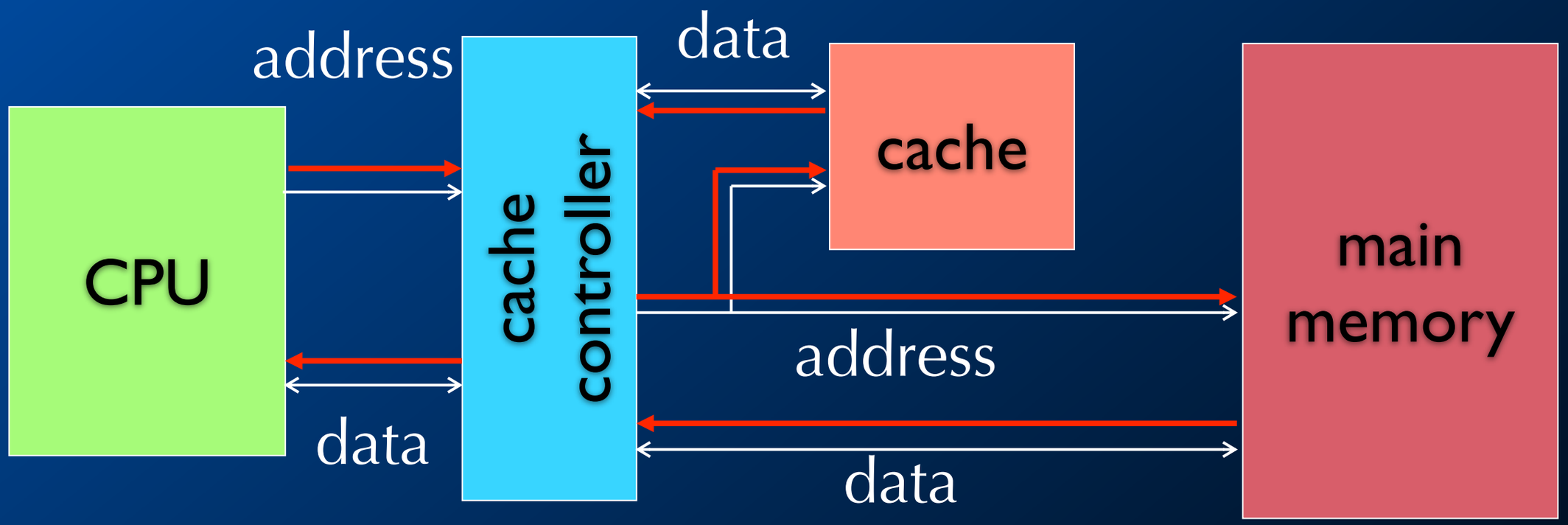
Virtual Memory

- Increase apparent memory size
- Swap data on/off of disk
- Not real time

Cache Memory

- Not strictly a memory architecture
- May often be ignored by programmers
- Optimization key to effective use

Caches and CPUs



Cache operation

- Many main memory locations are mapped onto one cache entry.
- May have caches for:
  - instructions;
  - data;
  - data + instructions (unified).
- Memory access time is no longer deterministic.

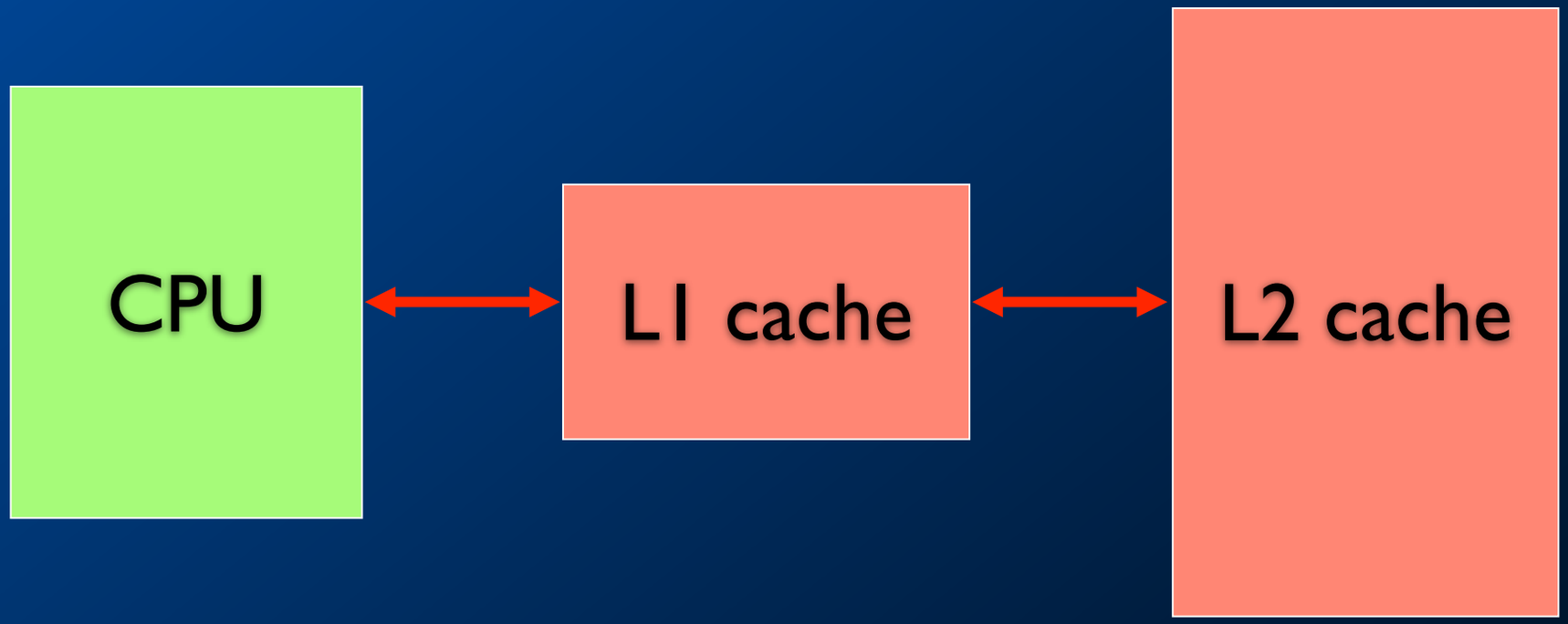
Terms

- Cache hit: required location is in cache.
- Cache miss: required location is not in cache.
- Working set: set of locations used by program in a time interval.

Memory system performance

- h = cache hit rate.
- t_{cache} = cache access time, t_{main} = main memory access time.
- Average memory access time:
- $t_{av} = ht_{cache} + (1-h)t_{main}$

Multiple levels of cache



Multi-level cache access time

- h_1 = cache hit rate.
- h_2 = hit rate on L2.
- Average memory access time:
  - $t_{av} = h_1 t_{L1} + h_2 t_{L2} + (1 - h_2 - h_1) t_{main}$

Replacement policies

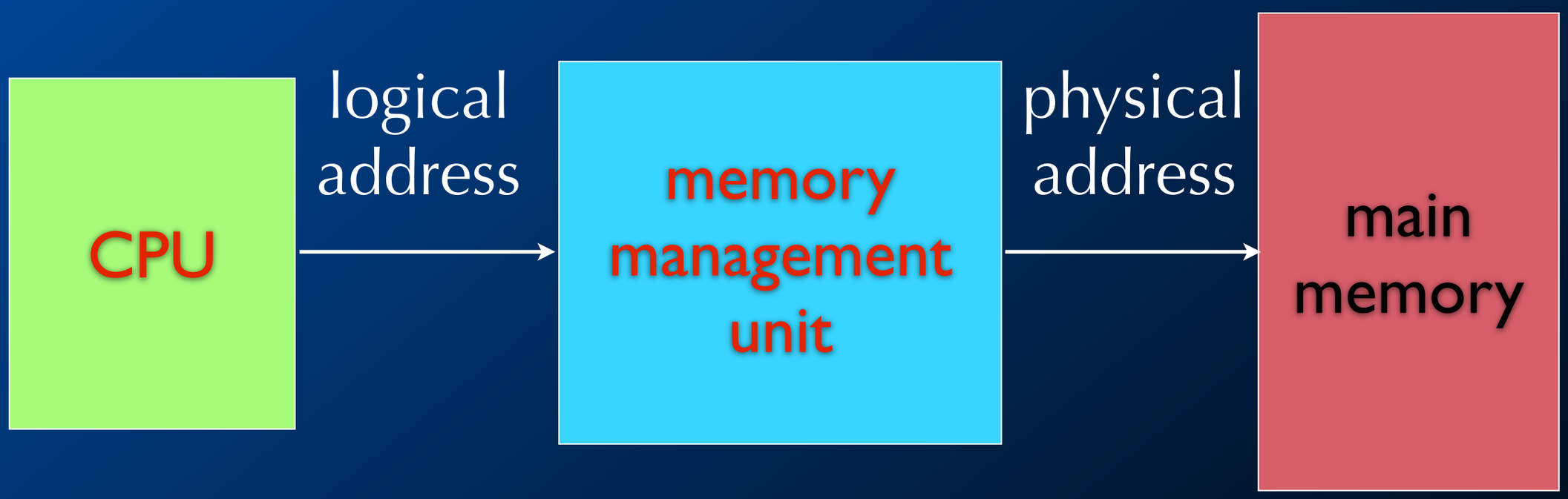
- Replacement policy: strategy for choosing which cache entry to throw out to make room for a new memory location.
- Two popular strategies:
  - Random.
  - Least-recently used (LRU).

Cache performance benefits

- Keep frequently-accessed locations in fast cache.
- Cache retrieves more than one word at a time.
  - Sequential accesses are faster after first access.

Memory management units

- Memory management unit (MMU) translates addresses:



Memory management tasks

- Allows programs to move in physical memory during execution.
- Allows virtual memory:
  - memory images kept in secondary storage;
  - images returned to main memory on demand during execution.
- Page fault: request for location not resident in memory.

Thank you!