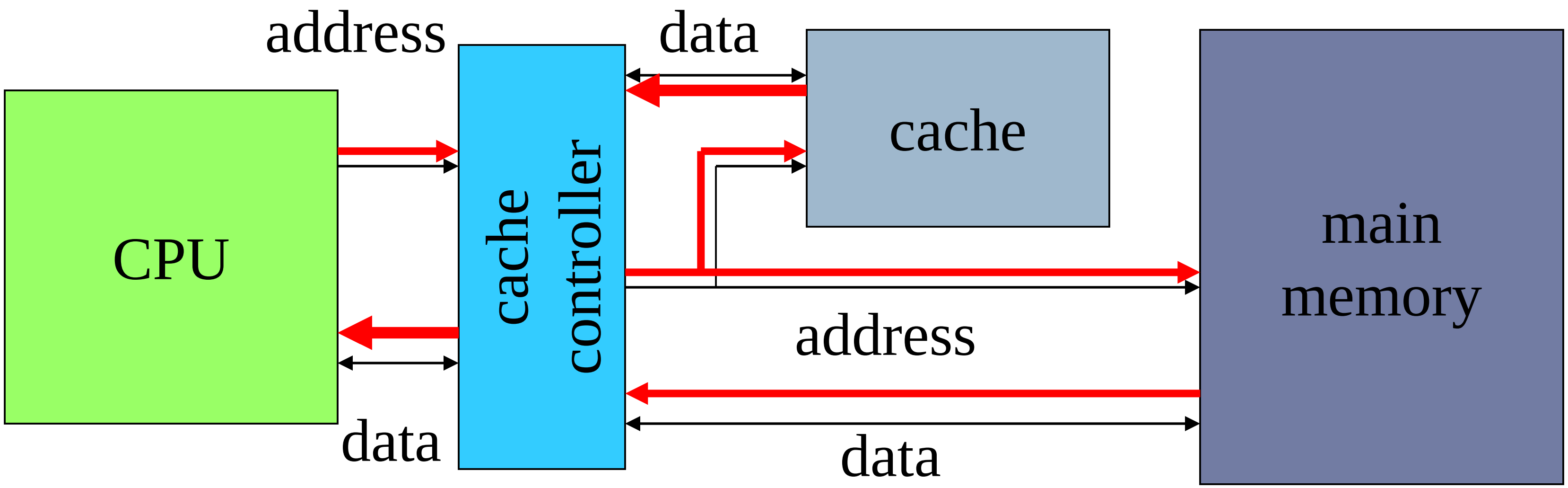
CPUs – Chapter 3.5

Caches.
Memory management.

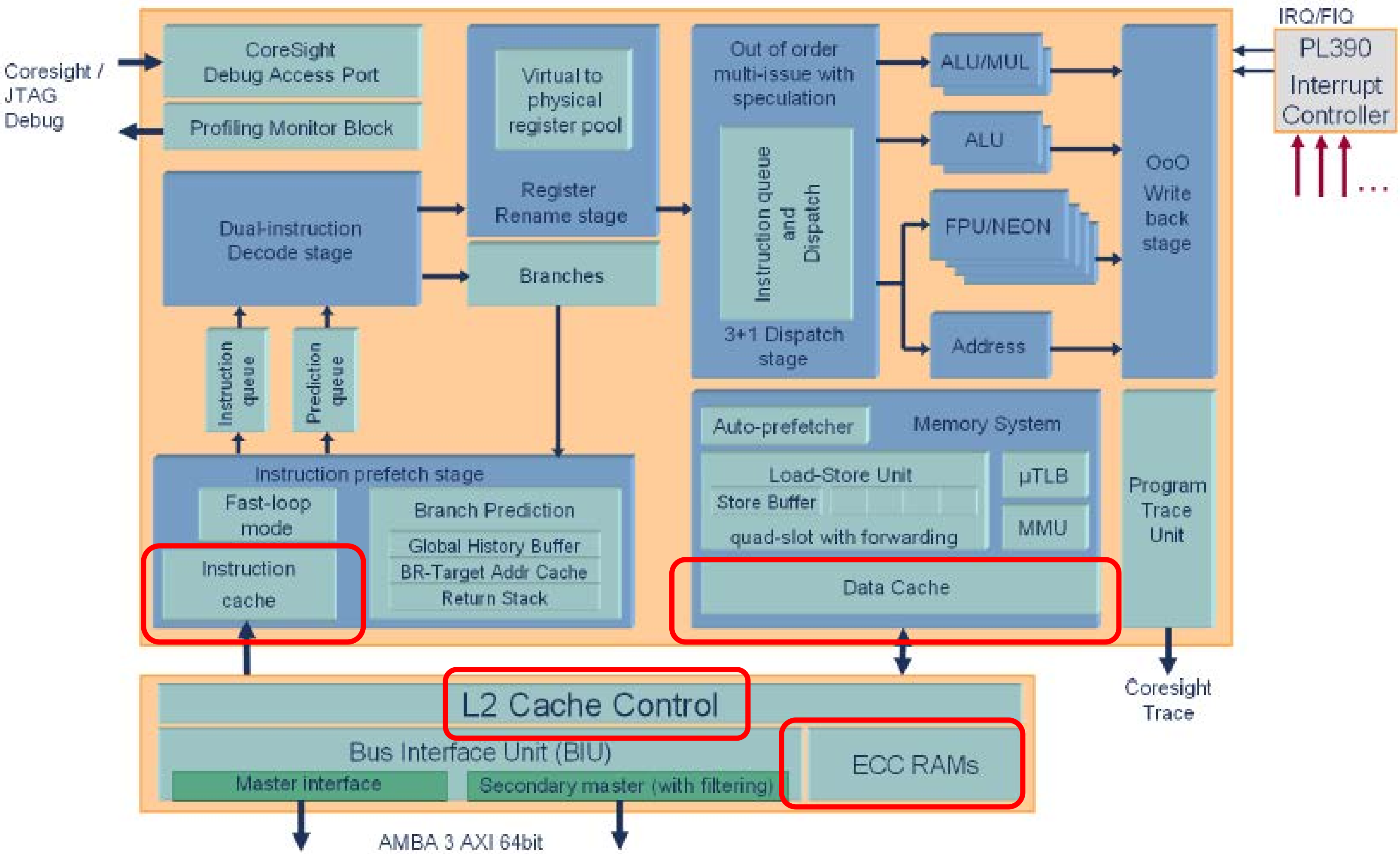
Caches and CPUs



ARM Cortex-A9 Configurations

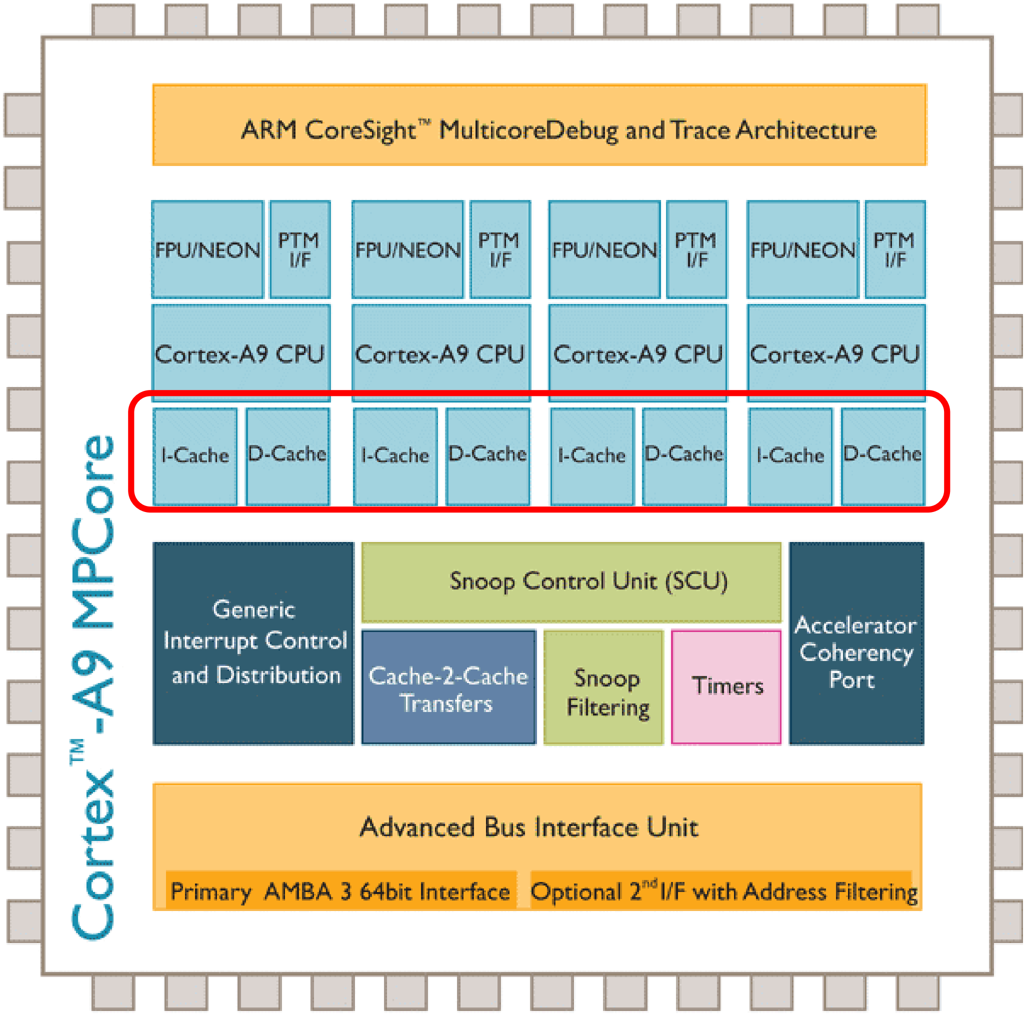
Next-Generation Devices	Typical Cortex-A9 Configuration
Mobile Handsets **Connected Mobile Computers** 	**High-end mobile devices (1500-3000DMIPS)** 2-3 core processor advanced power management 32K Instruction and Data caches, 256-512K shared L2 cache using PL310, partitioned AXI NEON technology-based Media Processing Engine
	Mid-range, cost reduction, (900-1500DMIPS) Single core processor with NEON or FPU 16K or 32K instruction and data caches 128-256K L2 cache using PL310, single AMBA AXI bus
	Feature-rich mass market (600-900DMIPS) Single core processor with FPU 16K instruction and data caches, single AXI
Consumer and Auto-infotainment 	**Consumer: user interactions (800-3000DMIPS)** 1-4 core processors giving design scalability across family of devices 32K instruction and data caches with 0-512K L2 cache NEON technology for advanced media and DSP processing Advanced bus interface unit for high-speed memory transfers between on-chip 3D engines and network interface MACs AMP configurations using separate CPU for real-time RTOS
Networking / Home Gateways 	**Enterprise market (4000-8000DMIPS)** 3-4 core performance optimized implementation 32K+64K instruction and data cache 512K-2MB L2 cache, dual 64 bit AMBA AXI interfaces
	Consumer devices (800-1500DMIPS) 1x or 2x multicore utilizing coherent accelerators 32+32K instruction and data, with 256-512K shared L2 cache NEON or VFP when offering media gateway or services
Embedded 	**Embedded media and imaging (800-2000DMIPS)** 2x multicore utilizing coherent accelerators 32+32K instruction and data with 256K shared L2 cache FPU for postscript and image manipulation and enhancement Code migration through selective AMP/SMP deployments

ARM Cortex A9 Microarchitecture



Main System Memory

ARM Cortex-A9 MPCore



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.
  - Depends on "hits" and "misses"
  - 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.
  - Anticipate what is needed to minimizes misses

Types of misses

- Compulsory (cold): location has never been accessed.
- Capacity: working set is too large.
- Conflict: multiple locations in working set map to same cache entry – fighting for the same cache location

- Cache miss penalty: added time due to a cache miss.

Cache performance benefits

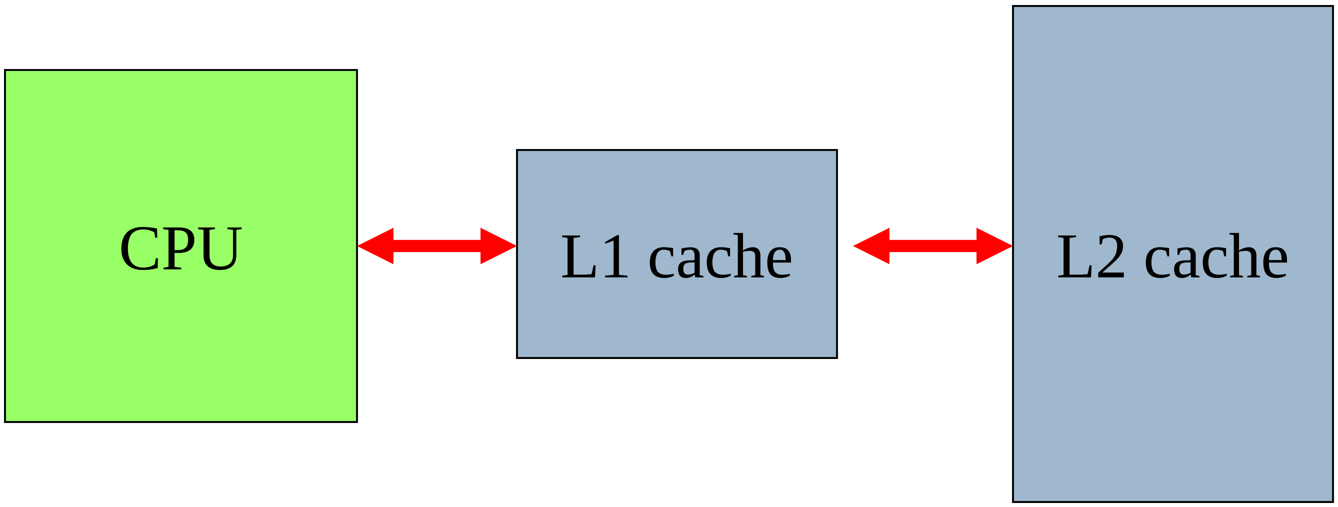
- Keep frequently-accessed locations in fast cache.
- Cache retrieves multiple words at a time from main memory.
  - Sequential accesses are faster after first access.

Memory system performance

- h = cache hit rate; $(1-h)$ = cache miss rate
- t_{cache} = cache access time
- t_{main} = main memory access time

- Average memory access time:
  - $t_{av} = ht_{cache} + (1-h)(t_{cache}+t_{main})$ *look-through cache*
  - $t_{av} = ht_{cache} + (1-h)t_{main}$ *look-aside cache*

Multiple levels of cache



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

Write operations

- Write-through: immediately copy write to main memory.
- Write-back: write to main memory only when location is removed from cache.

Replacement policies

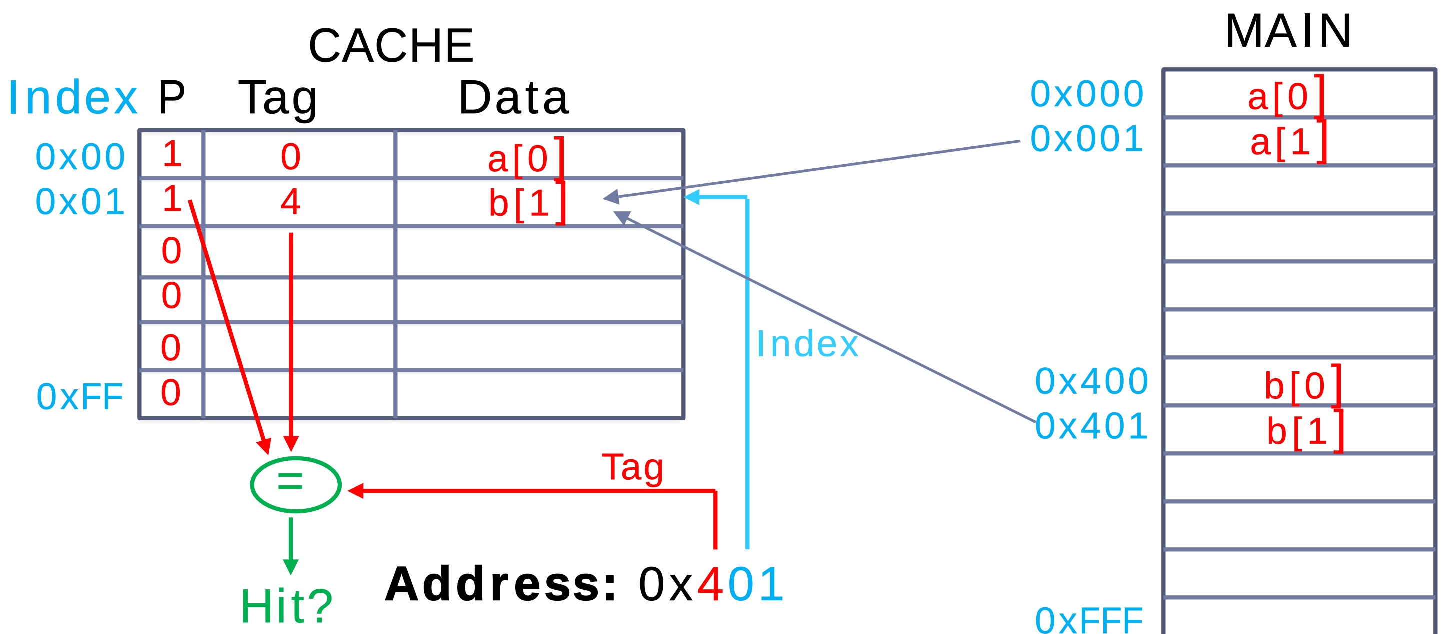
- Replacement policy: strategy for choosing which cache entry to throw out to make room for a new memory location.
- Two popular strategies:
  - Random.
  - Least-recently used (LRU).

Cache organizations

- Fully-associative: any memory location can be stored anywhere in the cache (almost never implemented).
- Direct-mapped: each memory location maps onto exactly one cache entry.
- N-way set-associative: each memory location can go into one of n sets.

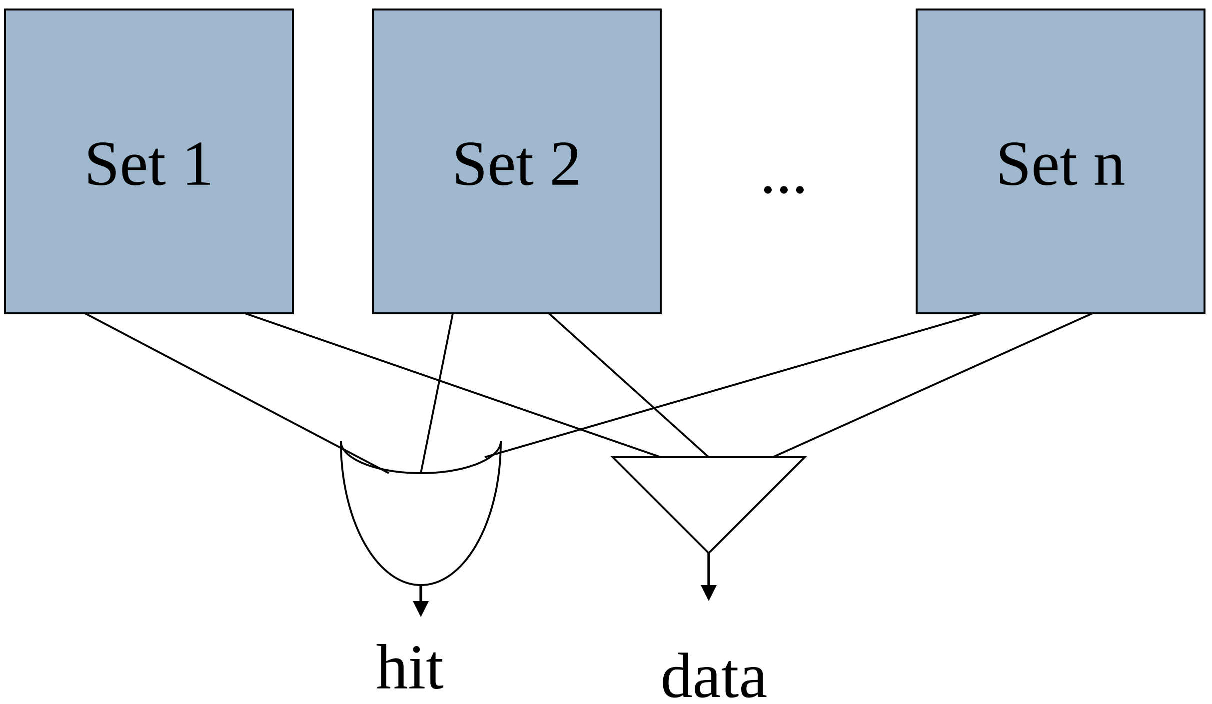
Direct-mapped cache locations

- Many locations map onto the same cache block.
- Conflict misses are easy to generate:
  - Array a[ ] uses locations 0, 1, 2, …
  - Array b[ ] uses loc's 0x400, 0x401, 0x402, …
  - Operation a[i] + b[i] generates conflict misses.



Set-associative cache

- A set of direct-mapped caches:



Example: direct-mapped vs. set-associative

address	data
000	0101
001	1111
010	0000
011	0110
100	1000
101	0001
110	1010
111	0100

Direct-mapped cache behavior

- After 001 access:

block	tag	data
00	-	-
01	0	1111
10	-	-
11	-	-

- After 010 access:

block	tag	data
00	-	-
01	0	1111
10	0	0000
11	-	-

Direct-mapped cache behavior, cont'd.

- After 011 access:

block	tag	data
00	-	-
01	0	1111
10	0	0000
11	0	0110

- After 100 access:

block	tag	data
00	1	1000
01	0	1111
10	0	0000
11	0	0110

Direct-mapped cache behavior, cont'd.

- After 101 access:

block	tag	data
00	1	1000
01	1	0001
10	0	0000
11	0	0110

- After 111 access:

block	tag	data
00	1	1000
01	1	0001
10	0	0000
11	1	0100

2-way set-associtive cache behavior

- Final state of cache (twice as big as direct-mapped):

set	blk 0 tag	blk 0 data	blk 1 tag	blk 1 data
001		1000	-	-
010		1111	1	0001
100		0000	-	-
110		0110	1	0100

2-way set-associative cache behavior

- Final state of cache (same size as direct-mapped):

set	blk 0 tag	blk 0 data	blk 1 tag	blk 1 data
0	01	0000	10	1000
1	10	0111	11	0100

ARM Cortex-A9 Configurations

Next-Generation Devices	Typical Cortex-A9 Configuration
Mobile Handsets **Connected Mobile Computers** 	**High-end mobile devices (1500-3000DMIPS)** 2-3 core processor advanced power management 32K Instruction and Data caches, 256-512K shared L2 cache using PL310, partitioned AXI NEON technology-based Media Processing Engine
	Mid-range, cost reduction, (900-1500DMIPS) Single core processor with NEON or FPU 16K or 32K instruction and data caches 128-256K L2 cache using PL310, single AMBA AXI bus
	Feature-rich mass market (600-900DMIPS) Single core processor with FPU 16K instruction and data caches, single AXI
Consumer and Auto-infotainment 	**Consumer: user interactions (800-3000DMIPS)** 1-4 core processors giving design scalability across family of devices 32K instruction and data caches with 0-512K L2 cache NEON technology for advanced media and DSP processing Advanced bus interface unit for high-speed memory transfers between on-chip 3D engines and network interface MACs AMP configurations using separate CPU for real-time RTOS
Networking / Home Gateways 	**Enterprise market (4000-8000DMIPS)** 3-4 core performance optimized implementation 32K+64K instruction and data cache 512K-2MB L2 cache, dual 64 bit AMBA AXI interfaces
	Consumer devices (800-1500DMIPS) 1x or 2x multicore utilizing coherent accelerators 32+32K instruction and data, with 256-512K shared L2 cache NEON or VFP when offering media gateway or services
Embedded 	**Embedded media and imaging (800-2000DMIPS)** 2x multicore utilizing coherent accelerators 32+32K instruction and data with 256K shared L2 cache FPU for postscript and image manipulation and enhancement Code migration through selective AMP/SMP deployments

Example caches

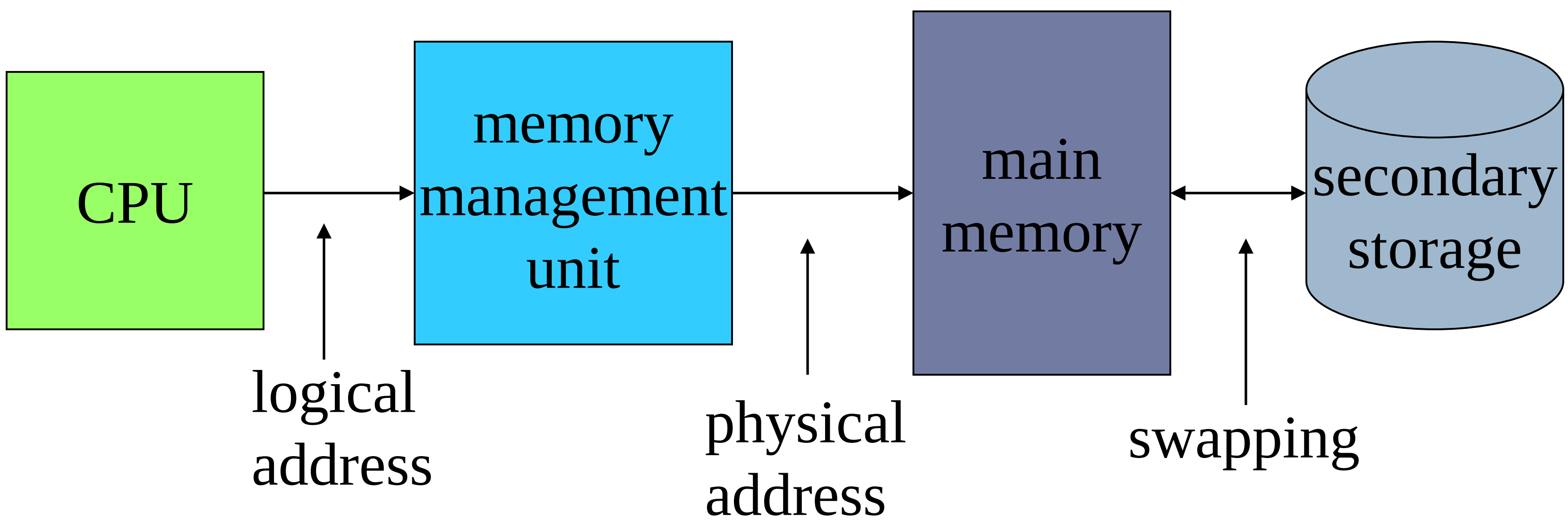
- StrongARM:
  - 16 Kbyte, 32-way, 32-byte block instruction cache.
  - 16 Kbyte, 32-way, 32-byte block data cache (write-back).
- C55x:
  - Various models have 16KB, 24KB cache.
  - Can be used as scratch pad memory.

Scratch pad memories

- Alternative to cache:
  - Software determines what is stored in scratch pad.
- Provides predictable behavior at the cost of software control.
- C55x cache can be configured as scratch pad.

Memory management units (3.5.2)

- Memory management unit (MMU) translates addresses:



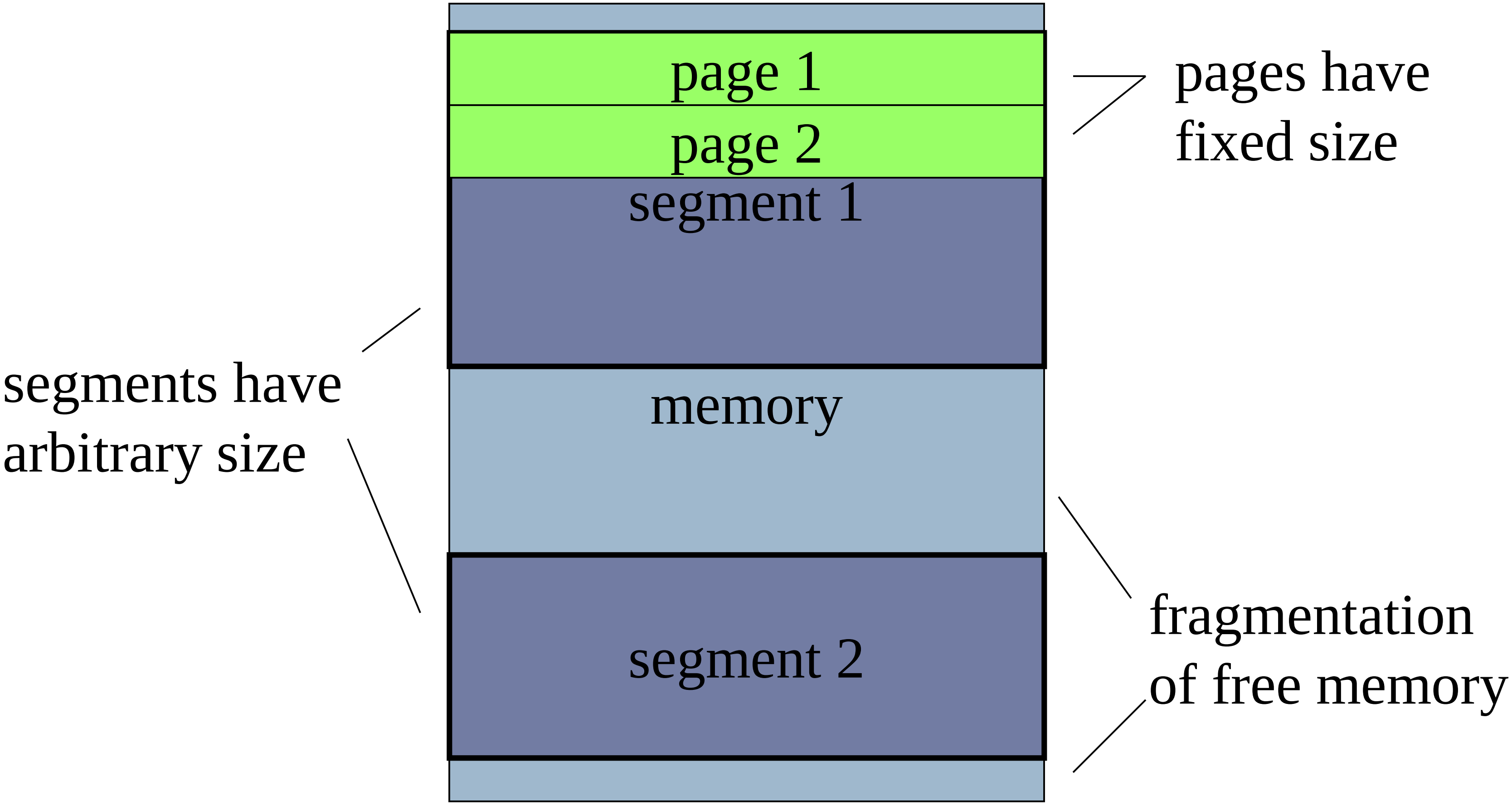
Memory management tasks

- 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.

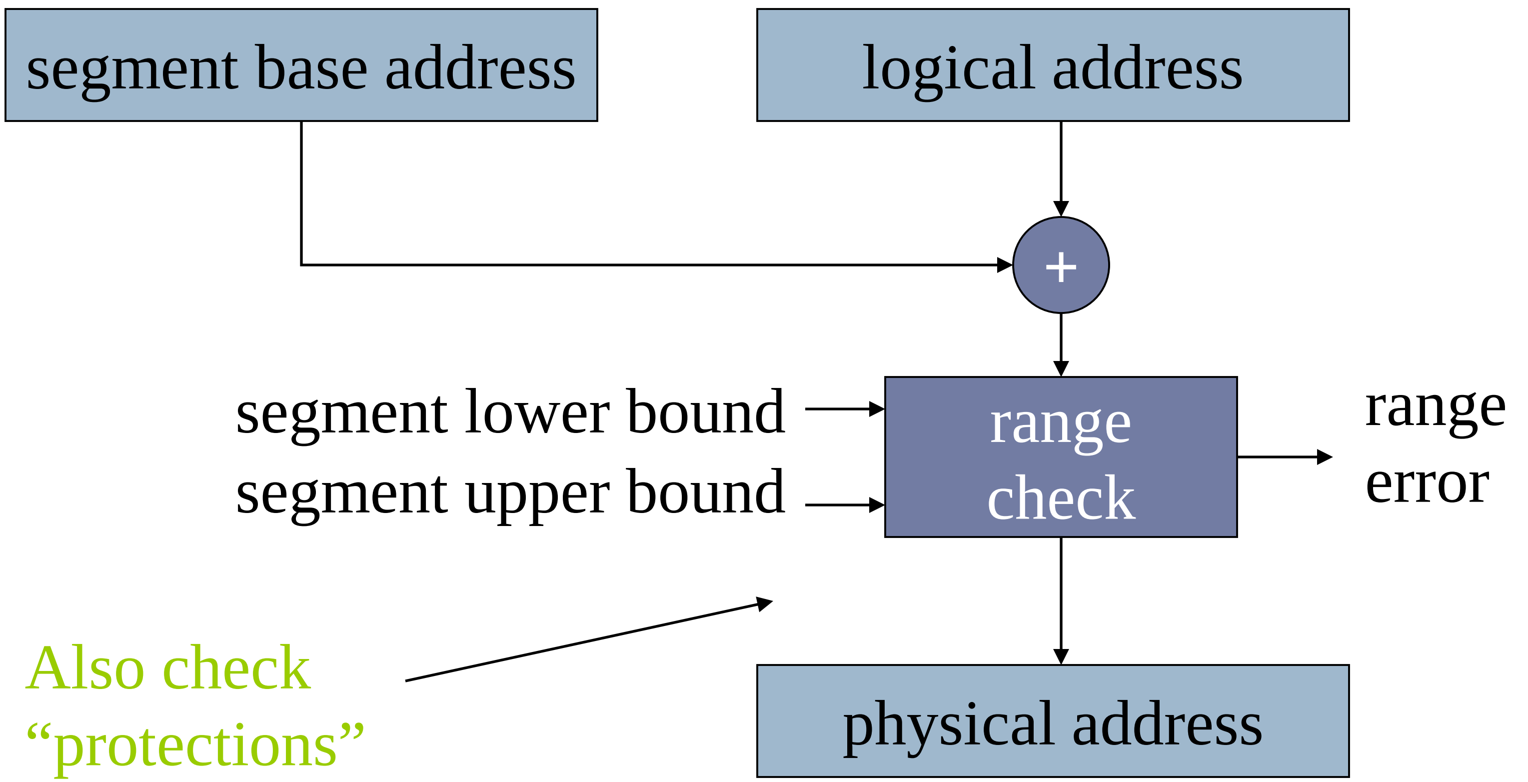
Address translation

- Requires some sort of register/table to allow arbitrary mappings of logical to physical addresses.
- Two basic schemes:
  - segmented;
  - paged.
- Segmentation and paging can be combined (x86, PowerPC).

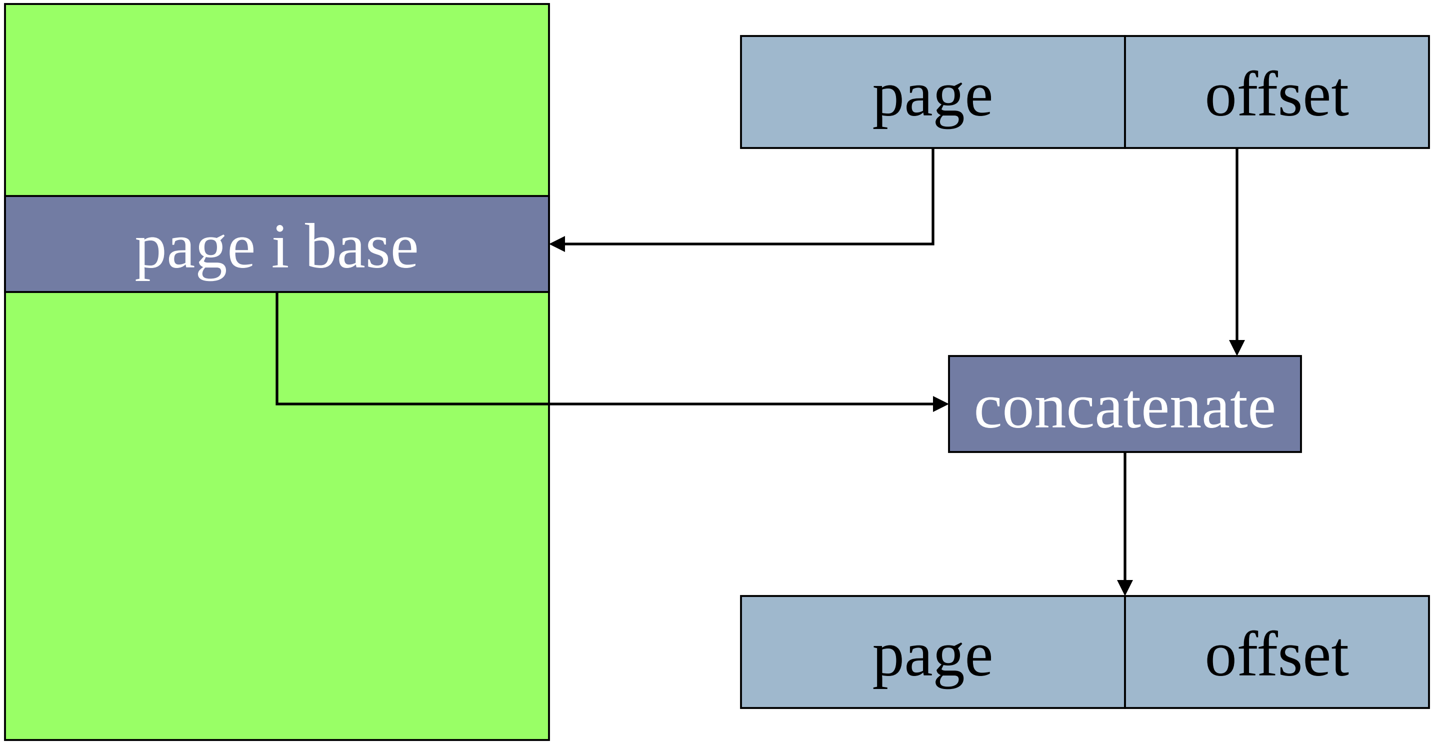
Segments and pages



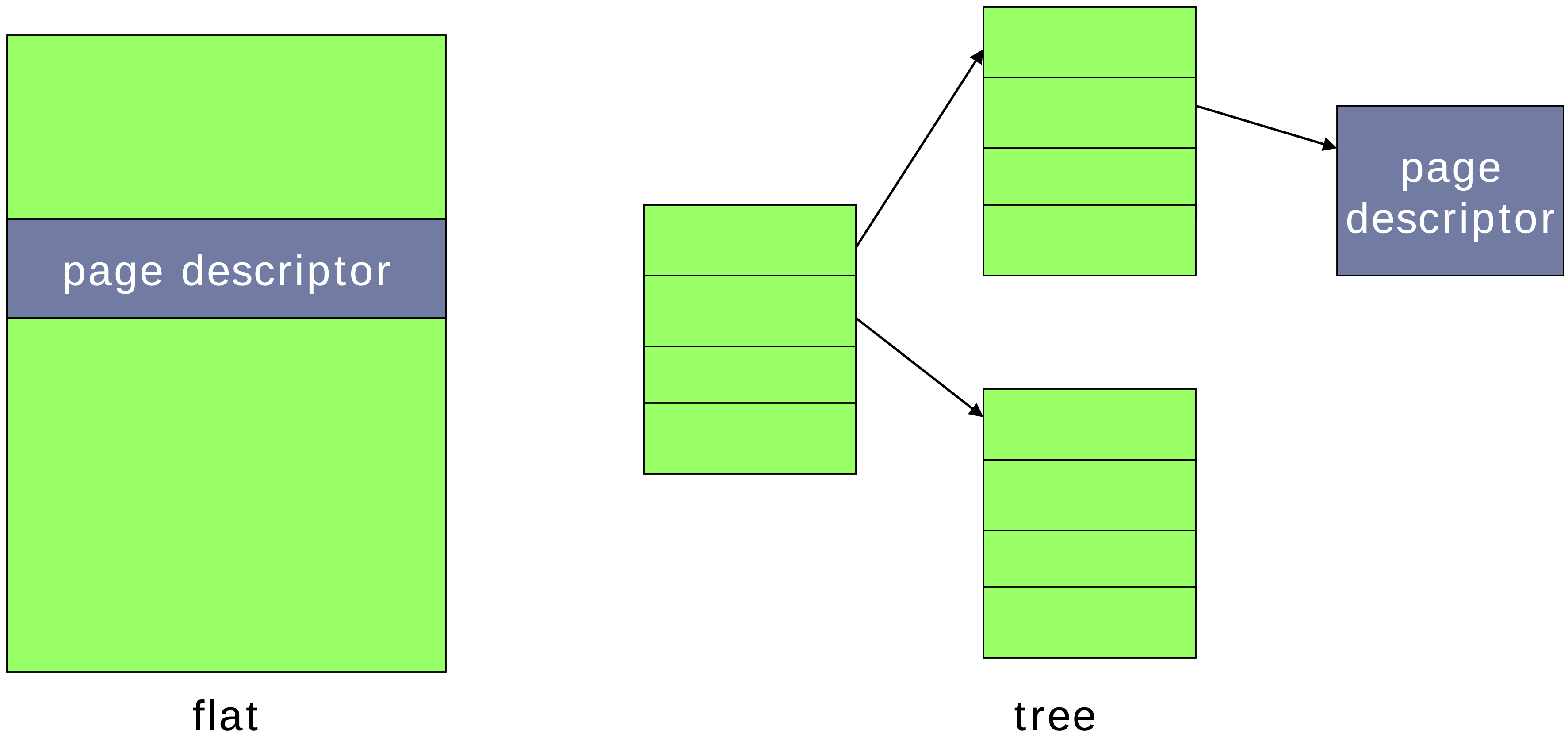
Segment address translation



Page address translation



Page table organizations



Caching address translations

- Large translation tables require main memory access.
- TLB (translation lookaside buffer): cache for address translation.
  - Typically small.

ARM memory management (optional)

- Memory region types:
  - section: 1 Mbyte block;
  - large page: 64 kbytes;
  - small page: 4 kbytes.
- An address is marked as section-mapped or page-mapped.
- Two-level translation scheme.

ARM address translation