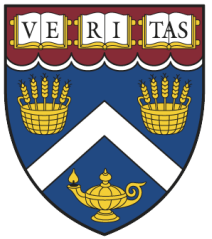
Tracing Garbage Collection



Mutators

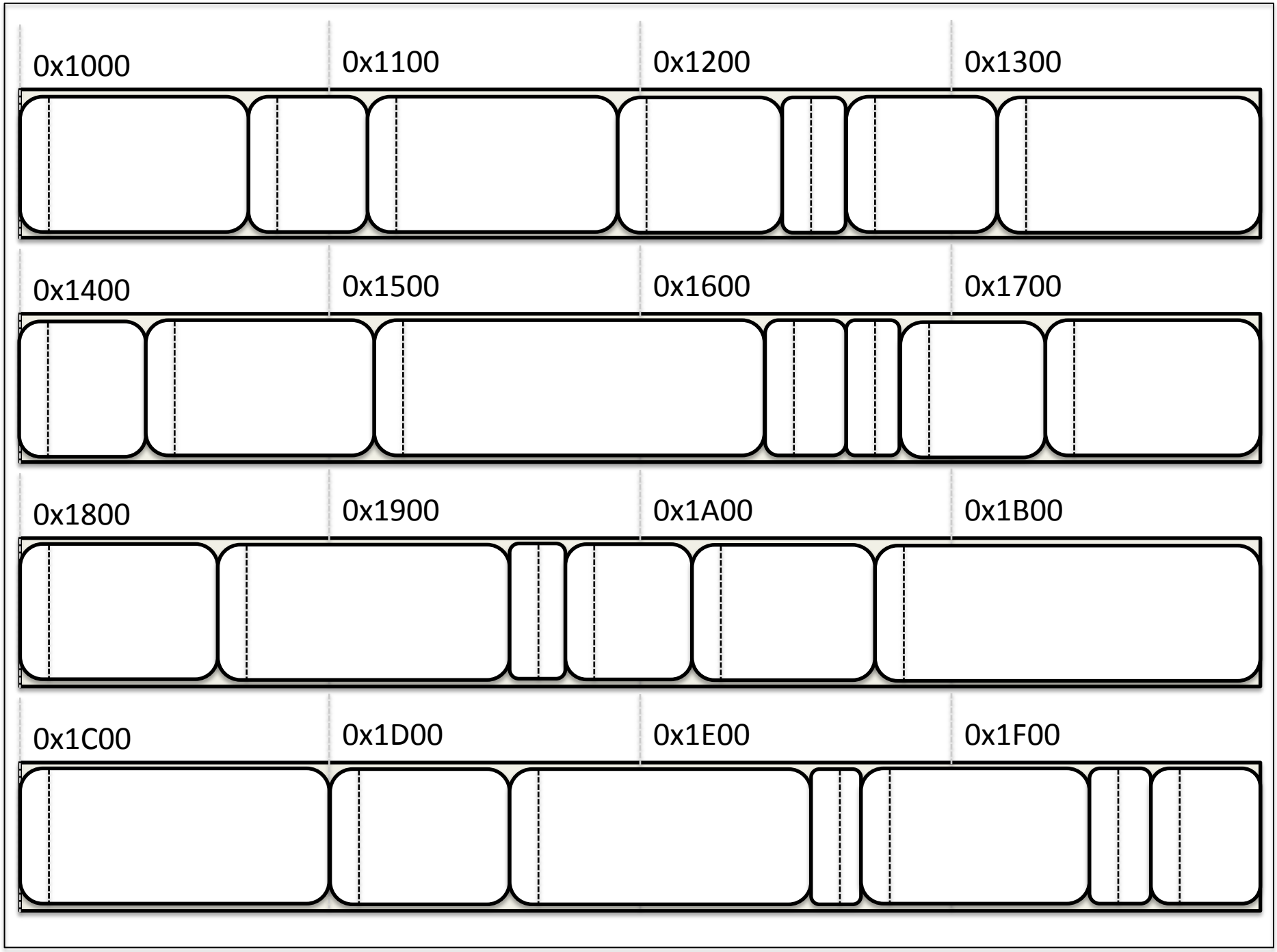
- Anything not doing memory management
  - Application threads
  - JIT compilation
  - Class loading
- Its only purpose is to change the object graph
- How we handle mutators is a major part of GC
  - Pause all non-GC activity
  - Work around a potentially changing graph

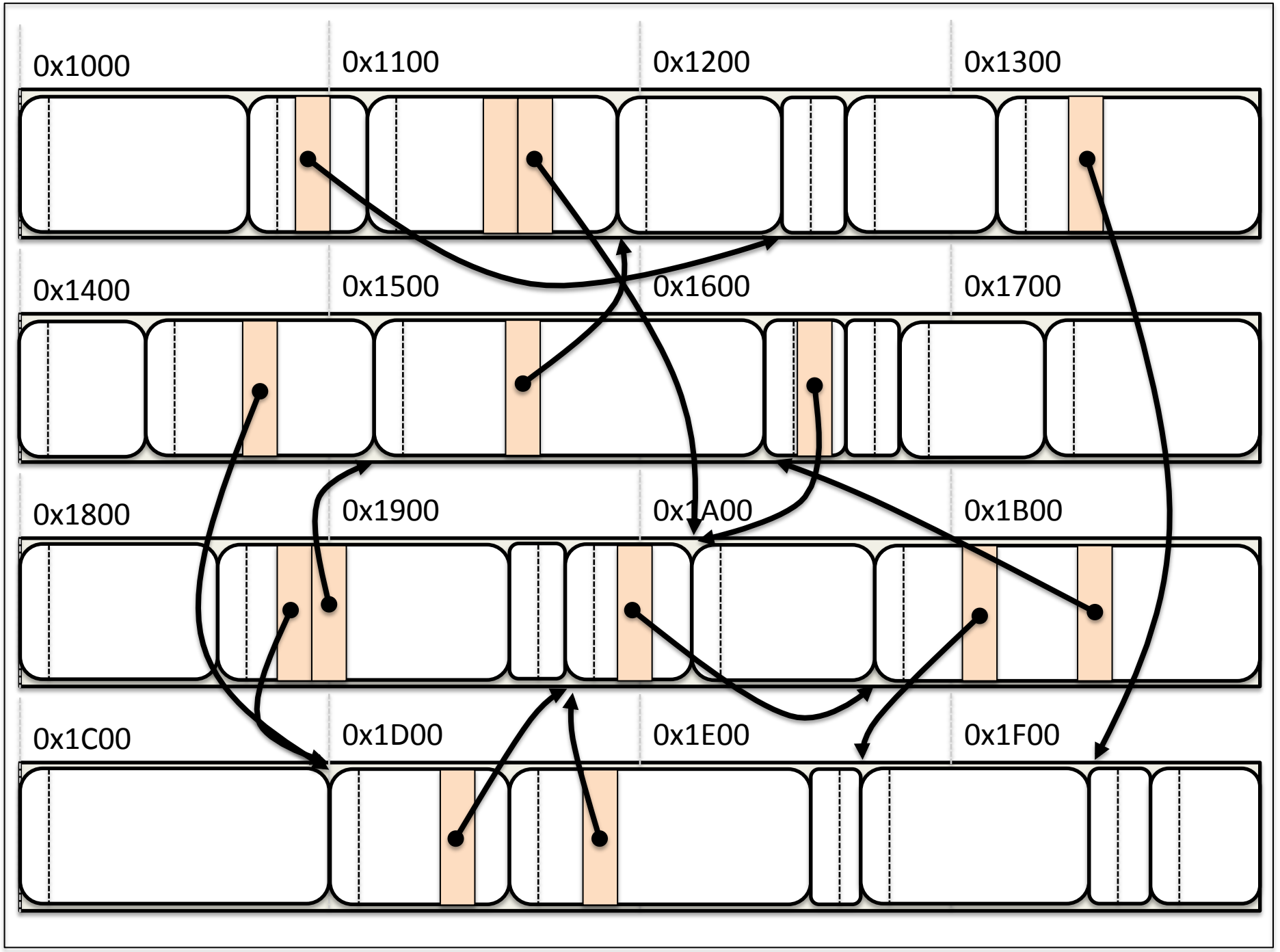
Stop the World Collection

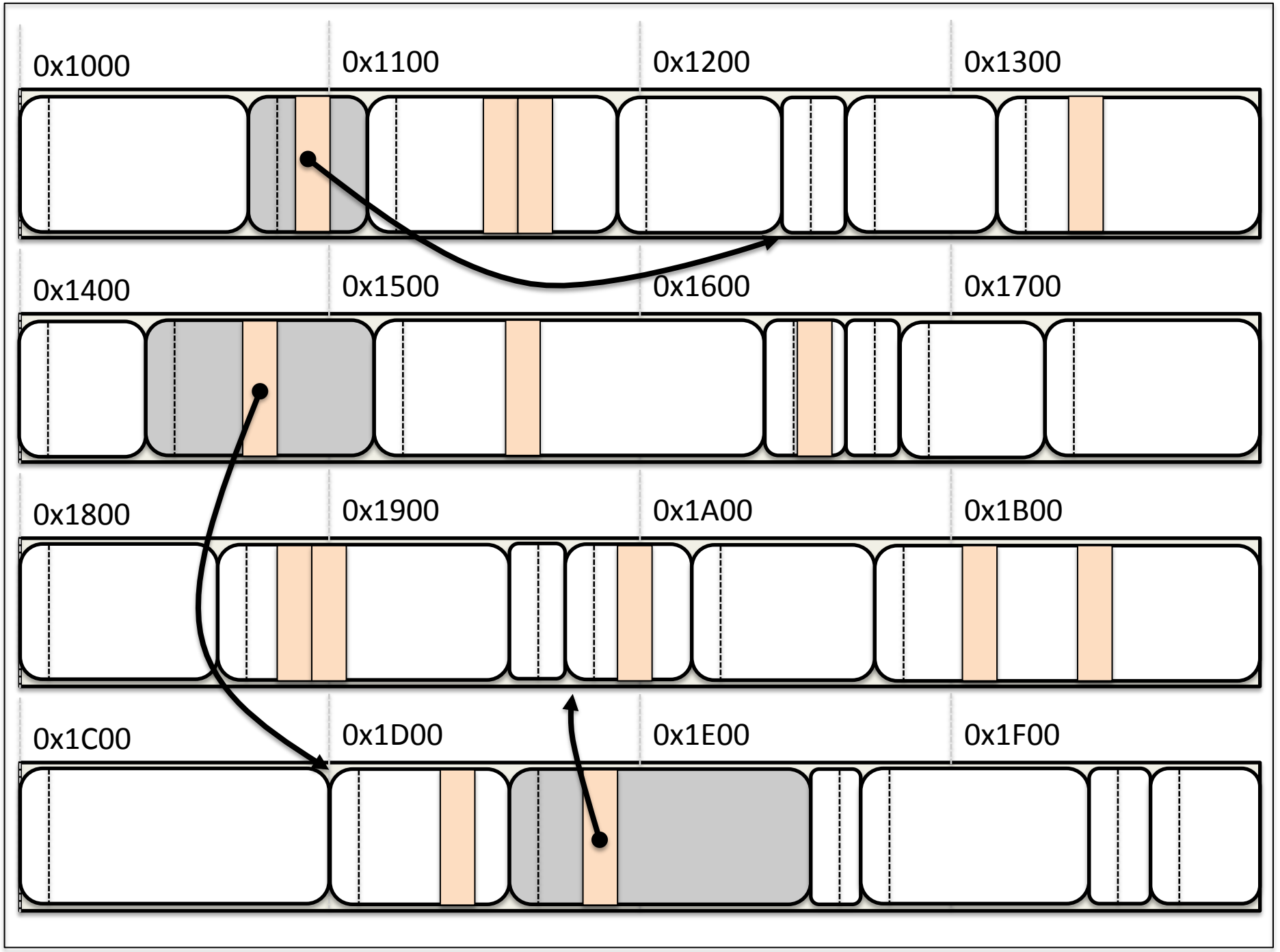
- Look at garbage collection across the whole system
  - Not incremental like reference counting
- Simplest implementation strategy
  - Pause all mutator threads
  - Do garbage collection work in peace
  - Restart all mutator threads
- Useful way to think about algorithms
  - No longer state of the art
  - Pause times can become unbearable for large heaps

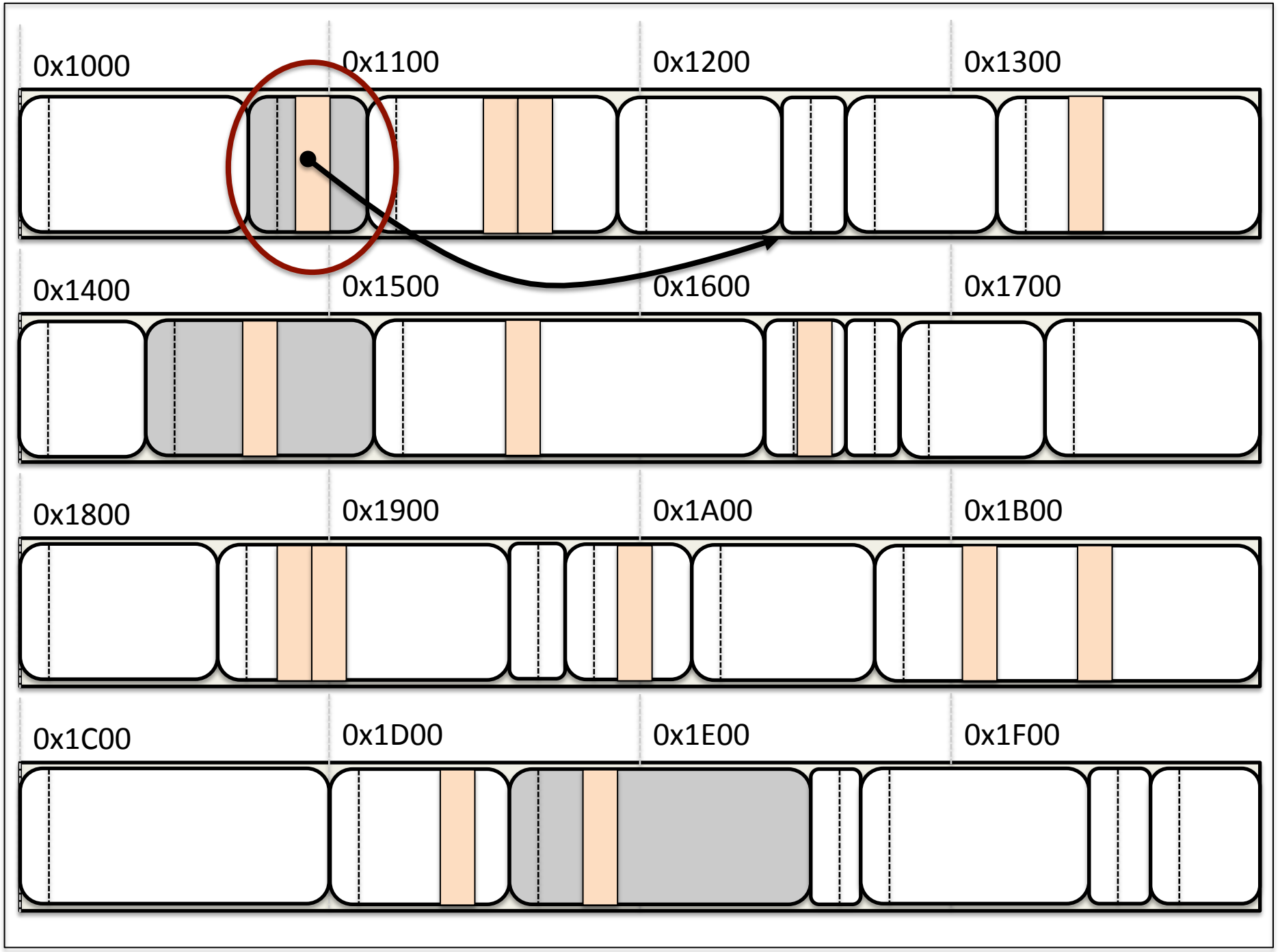
Tracing

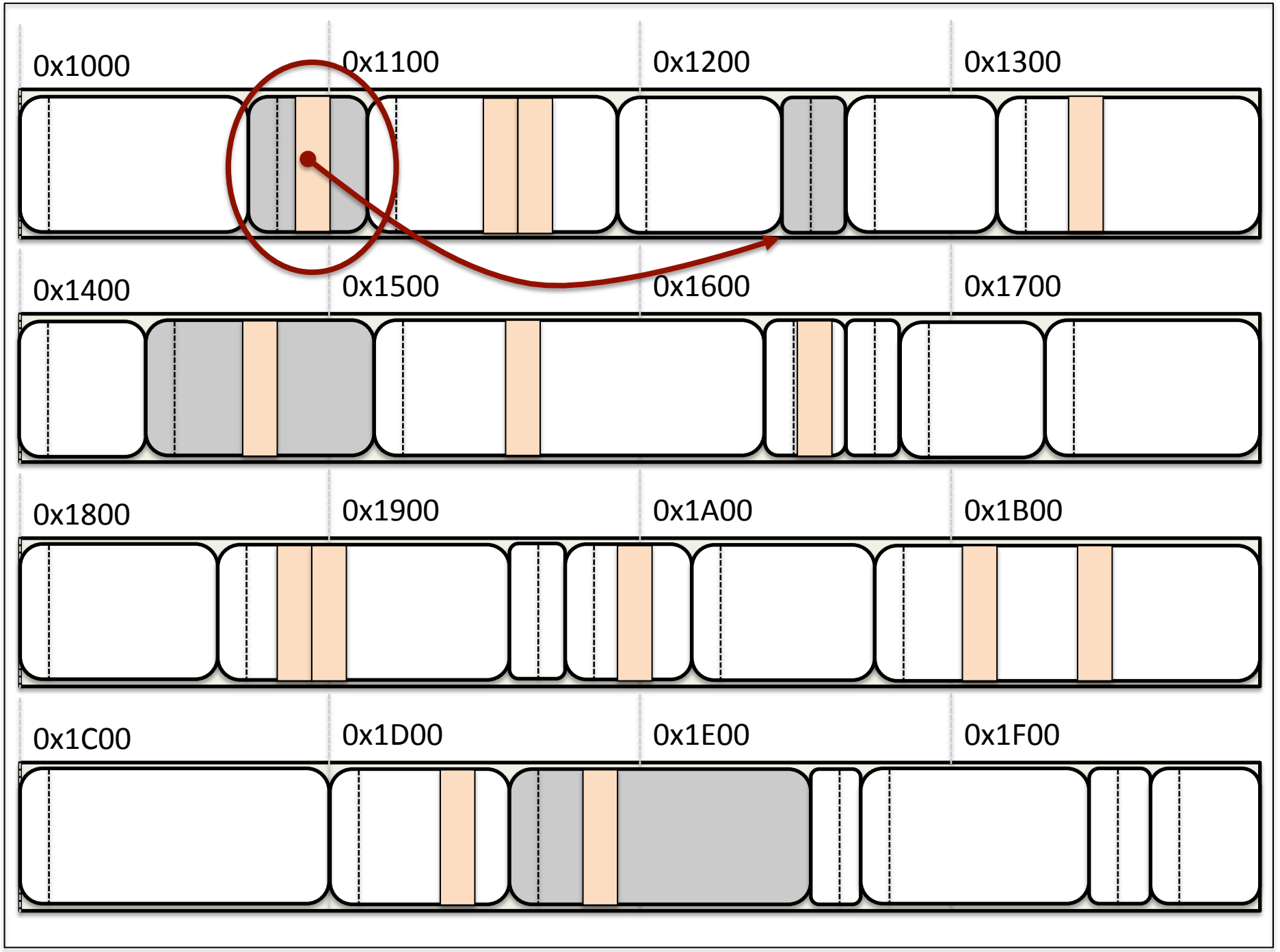
- Tracing lets us find all live objects on the heap
  - Start with objects referred to from the system roots
  - Mark those objects as live
  - Scan those live objects for more references
  - Repeat until all reachable objects are marked
- Garbage cycles not an issue
  - Marked object terminates cycle
- Need perfect type information

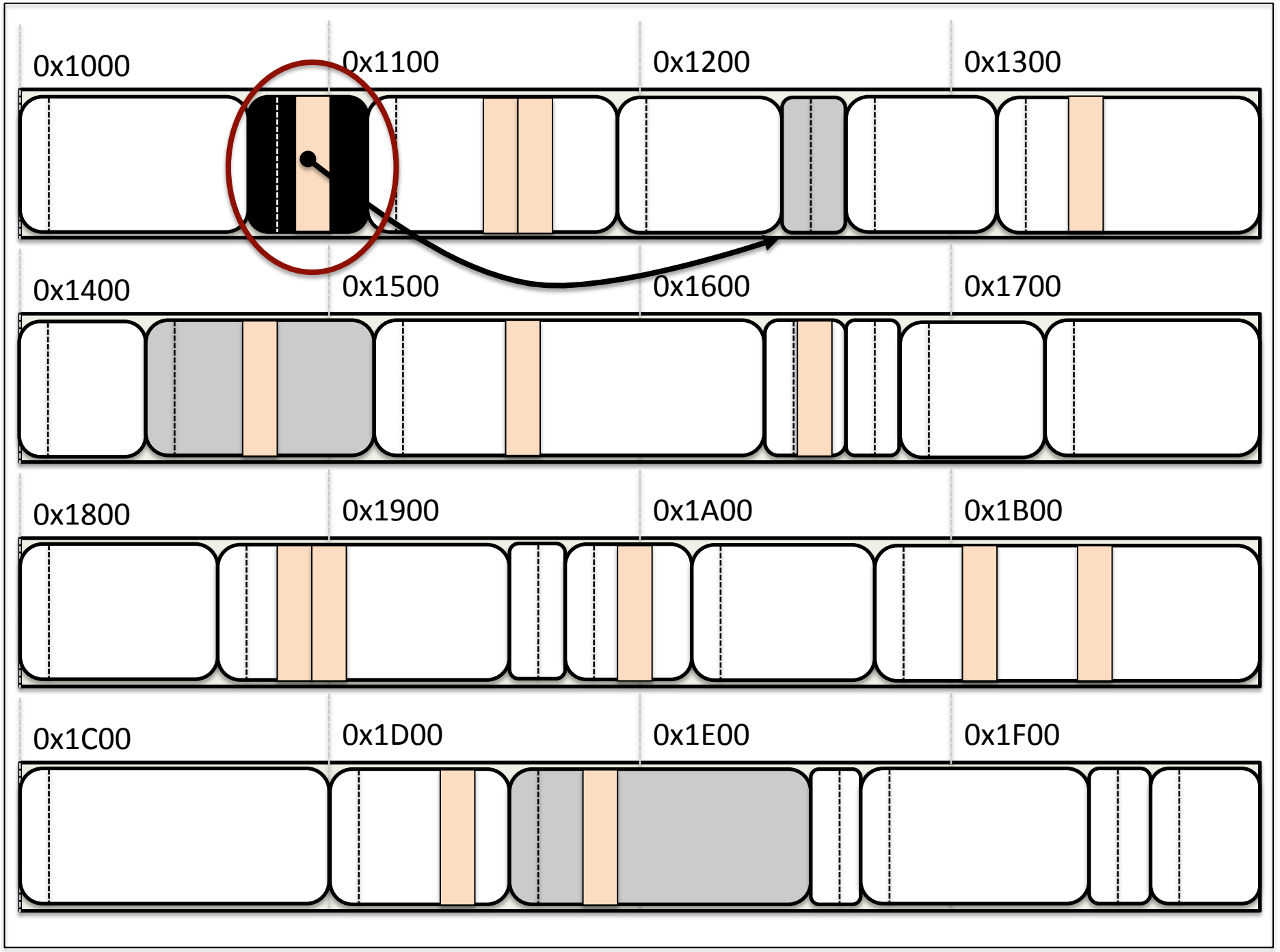

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

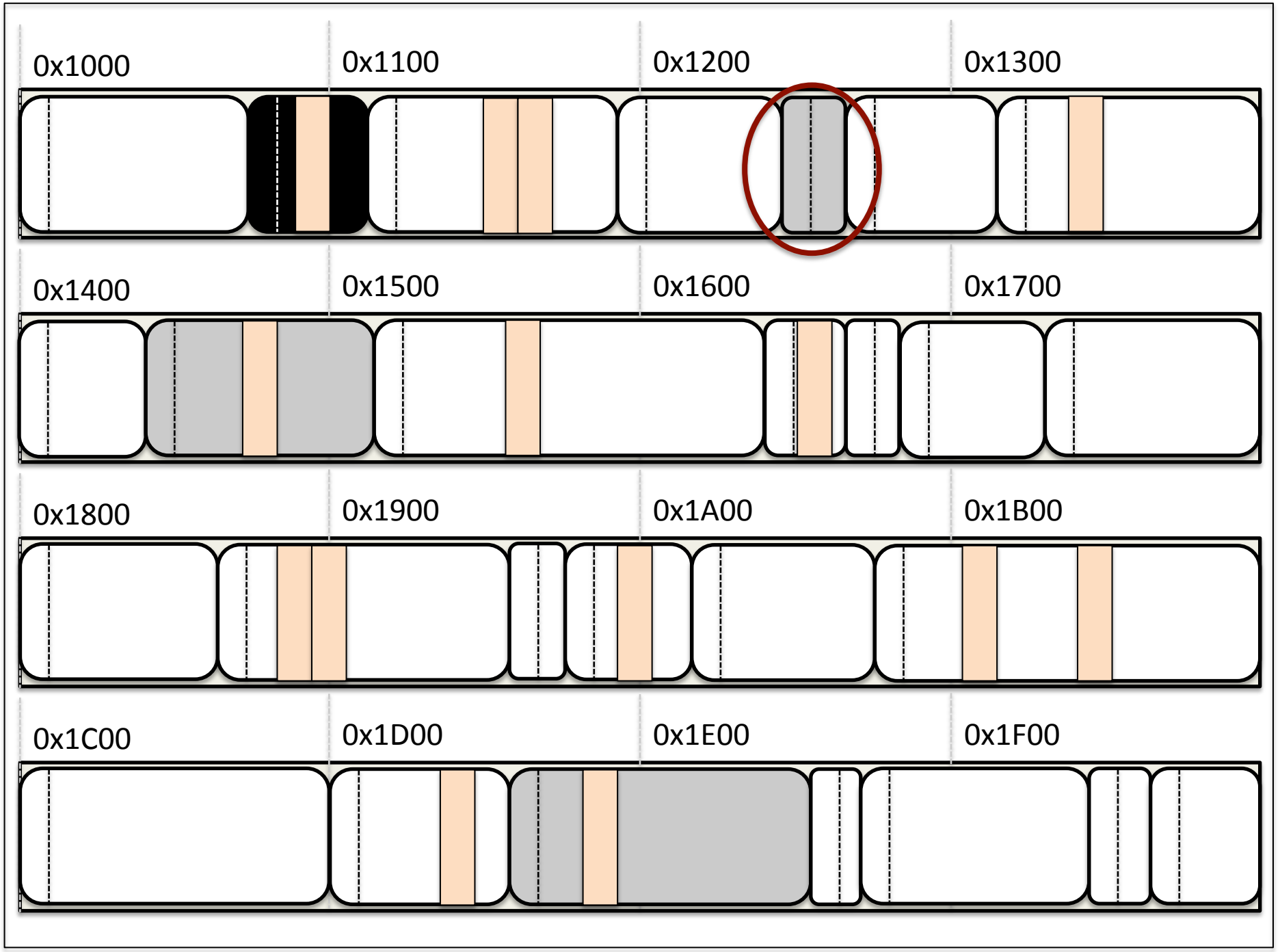

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

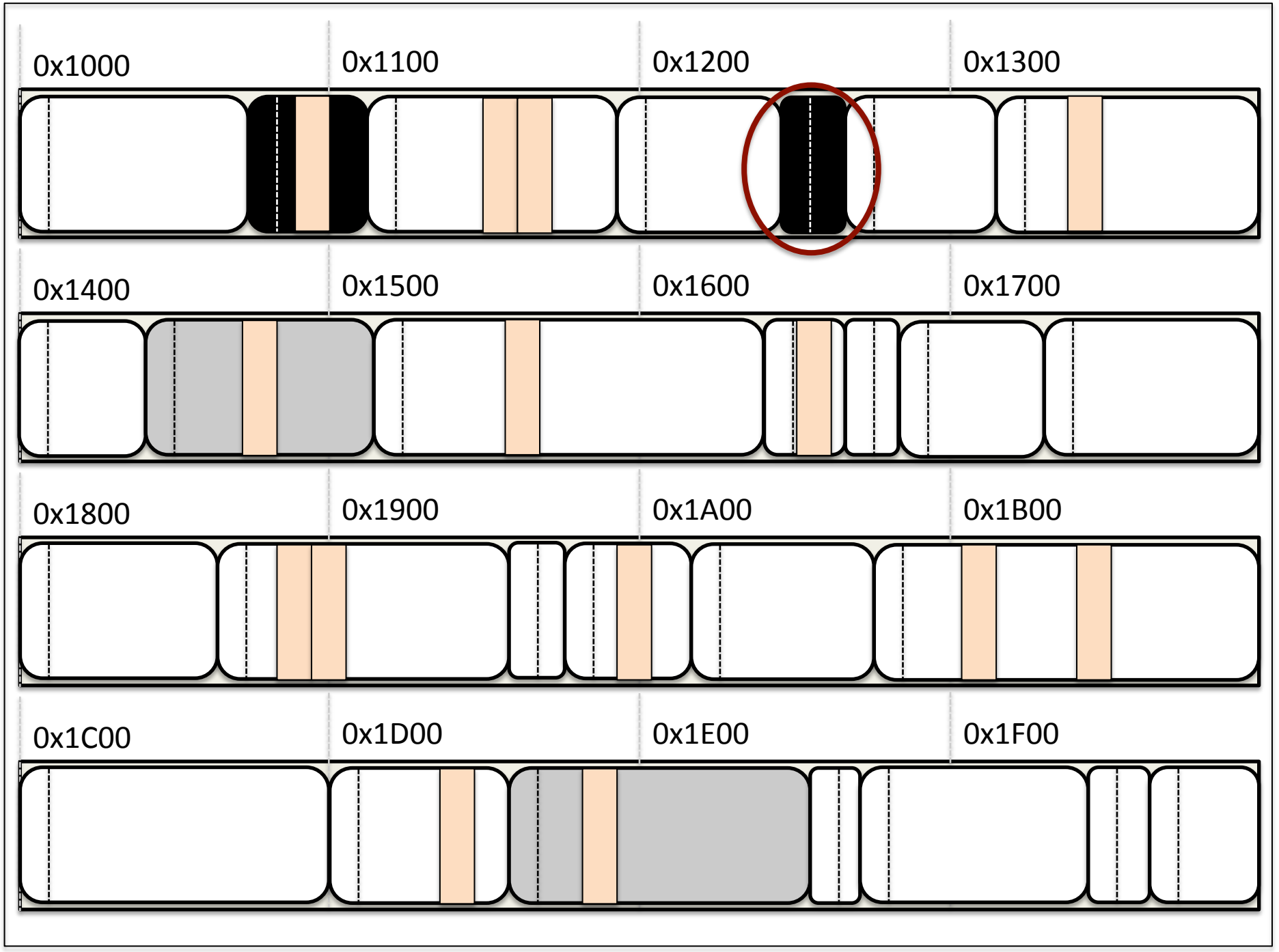

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

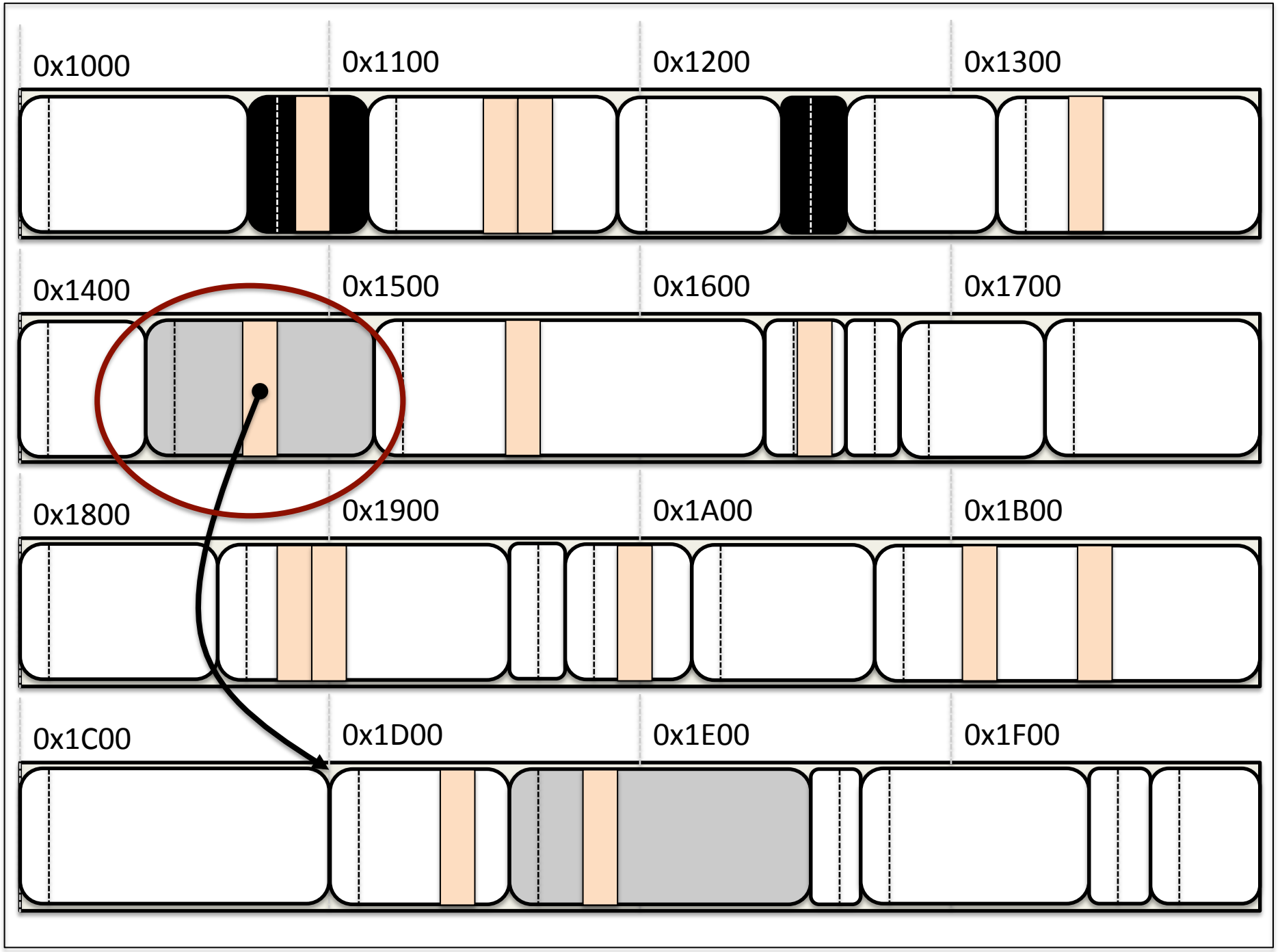

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

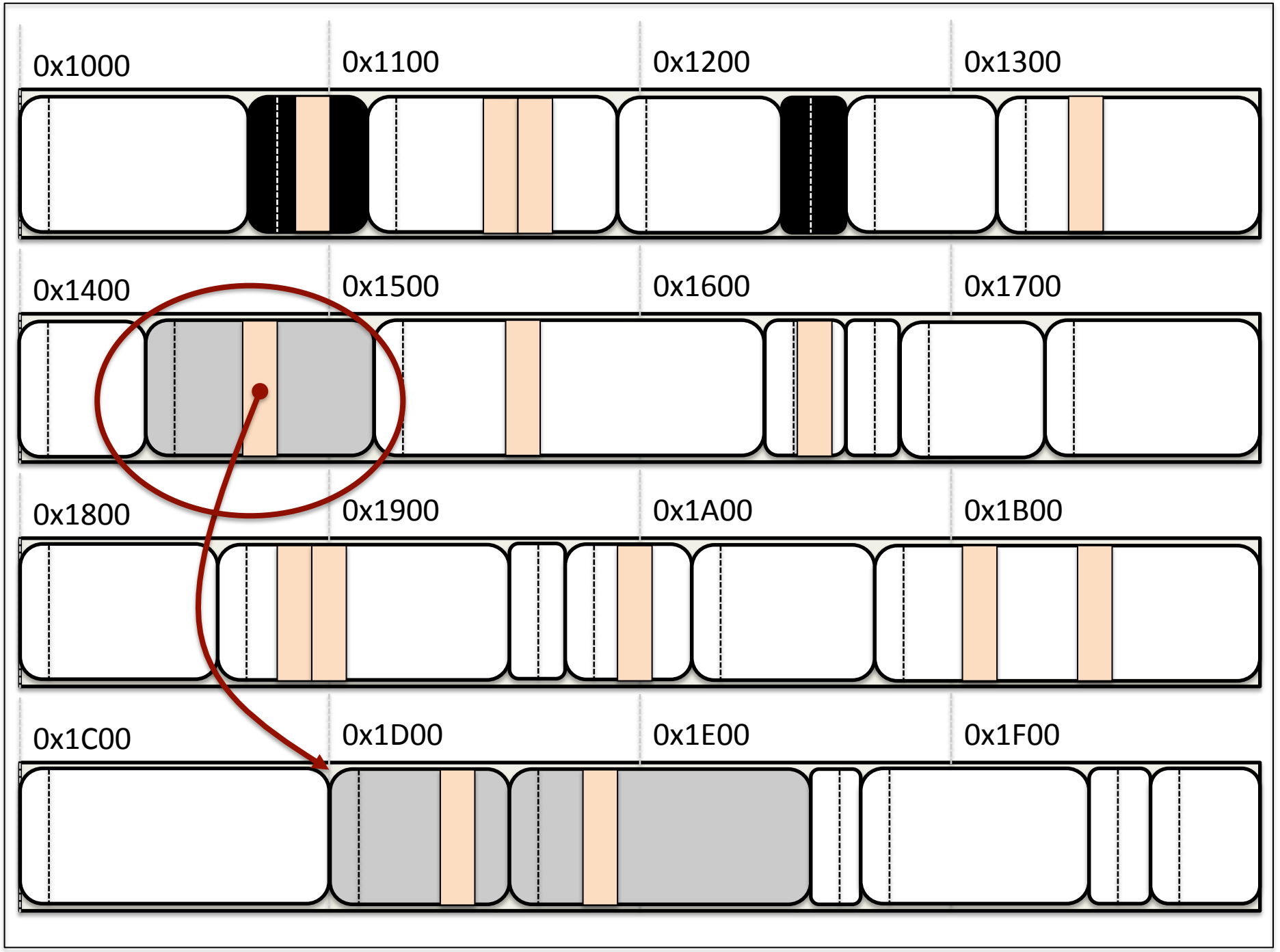

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

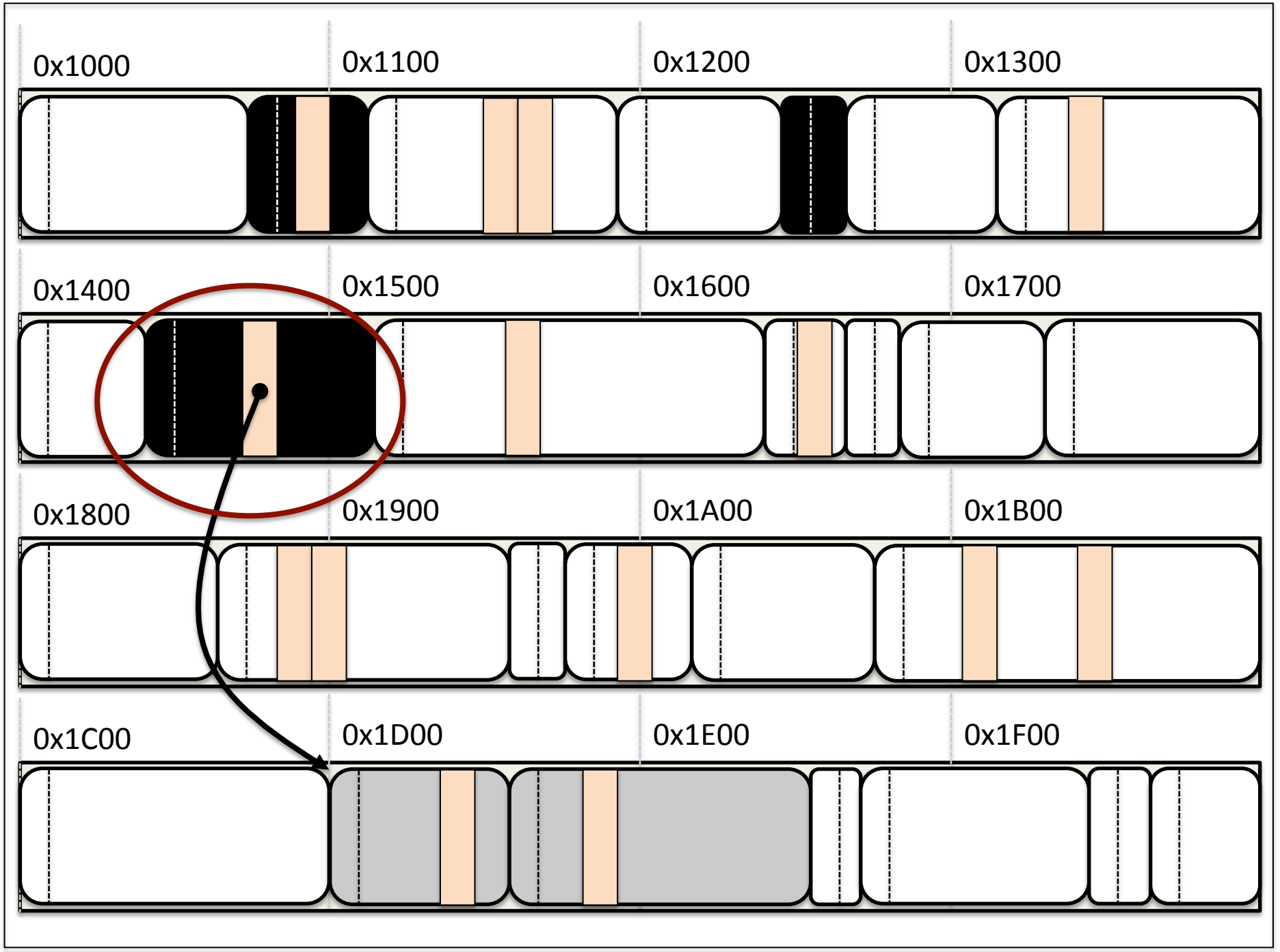

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

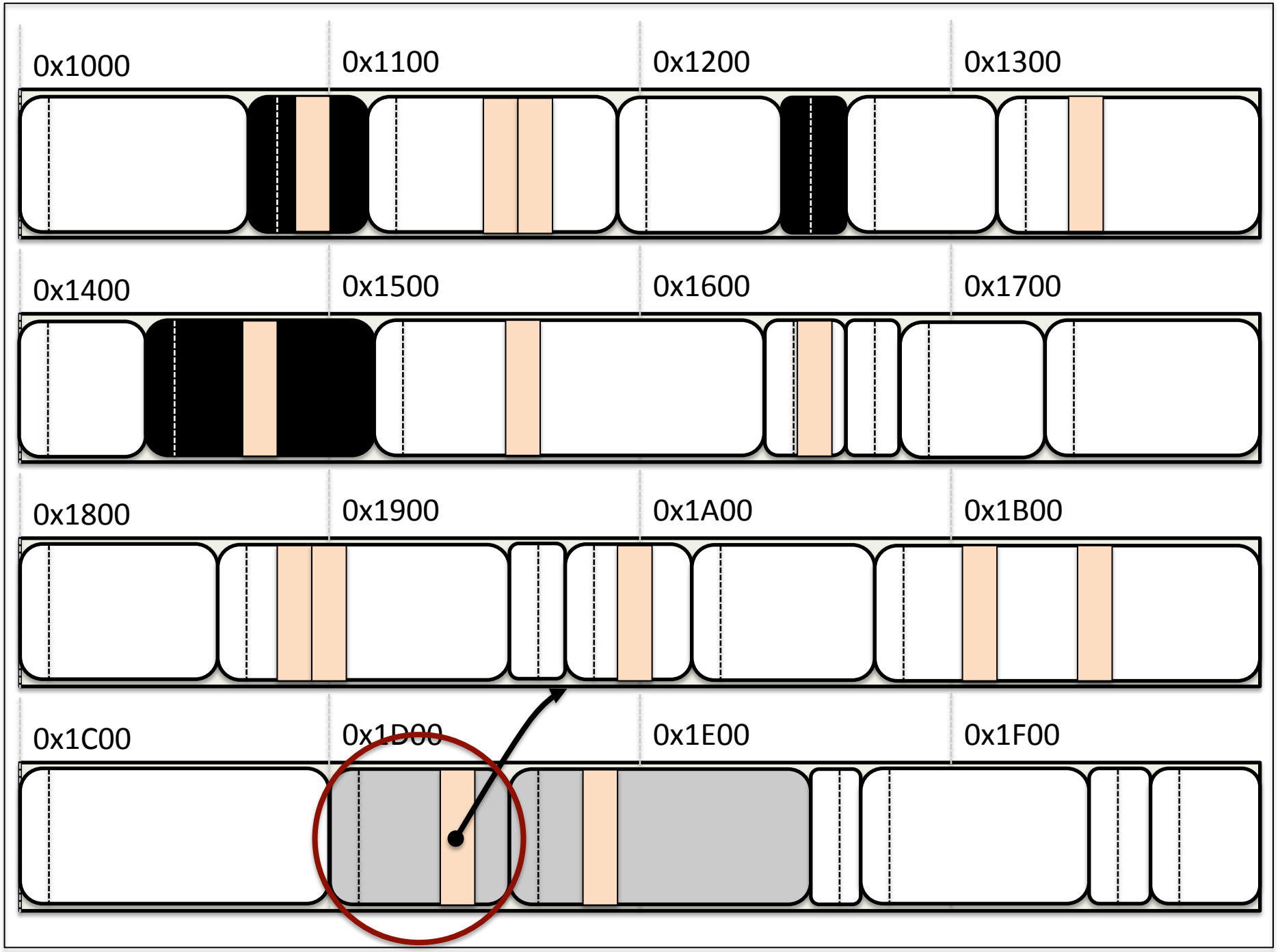

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

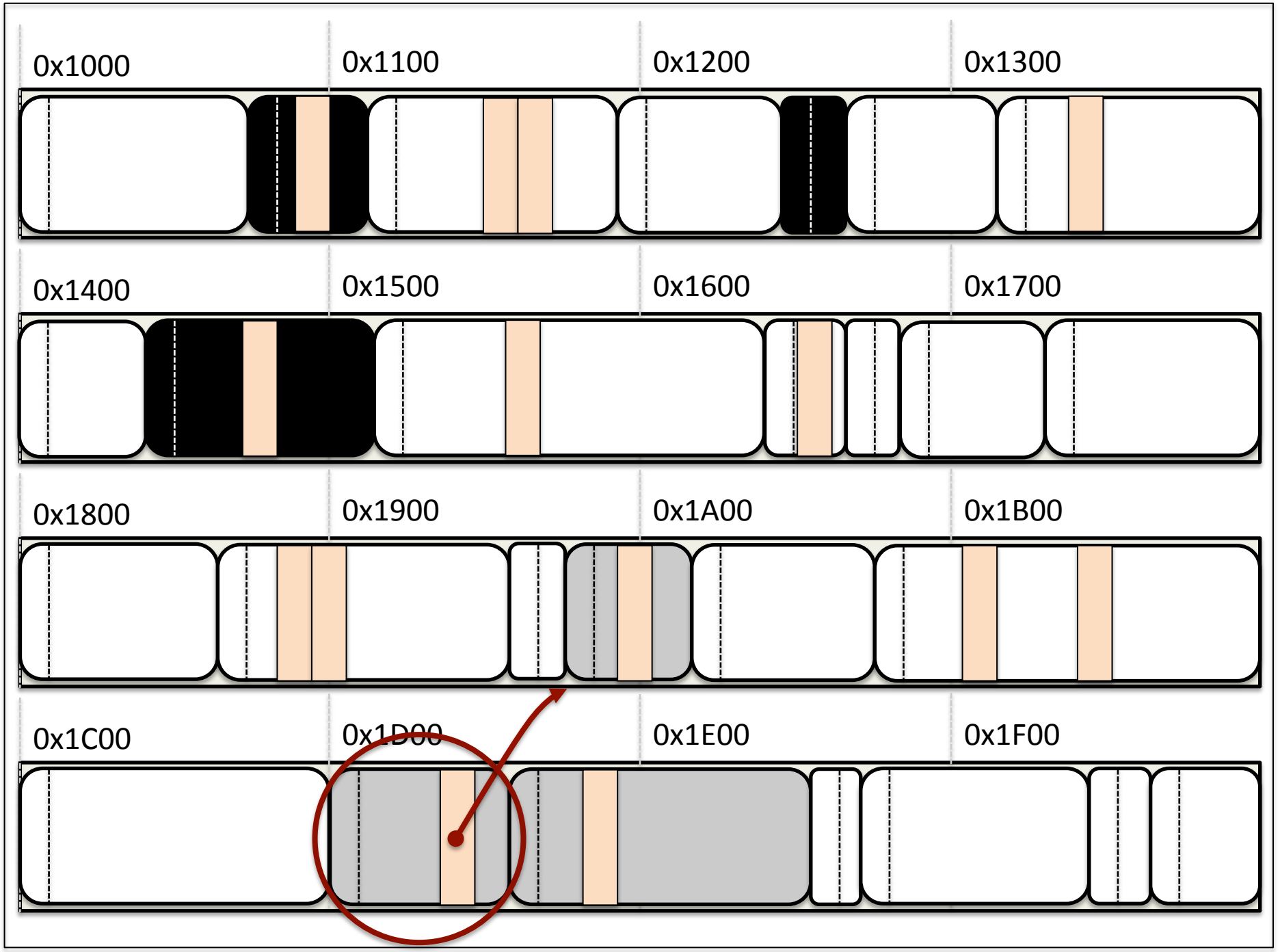

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

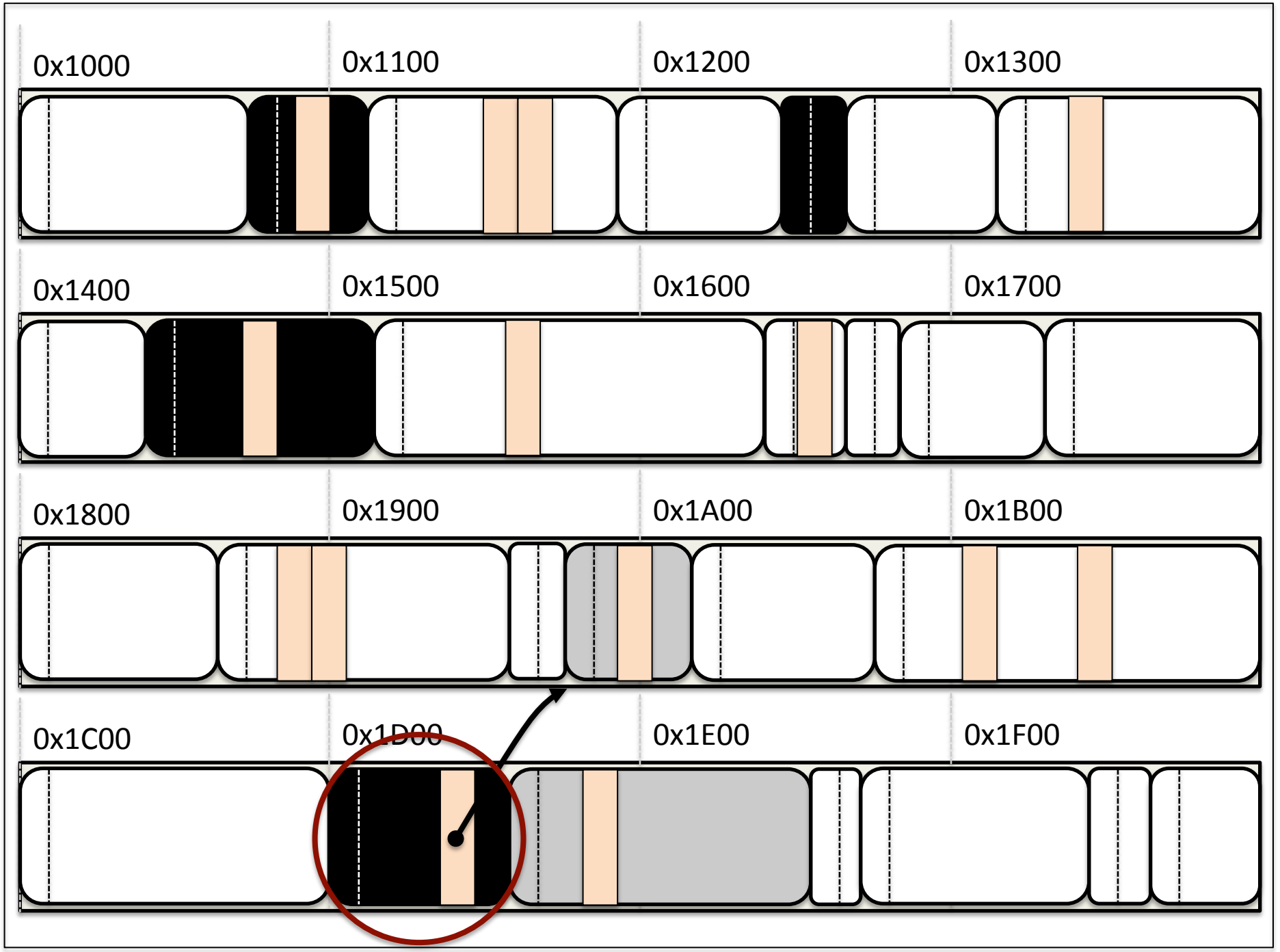

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

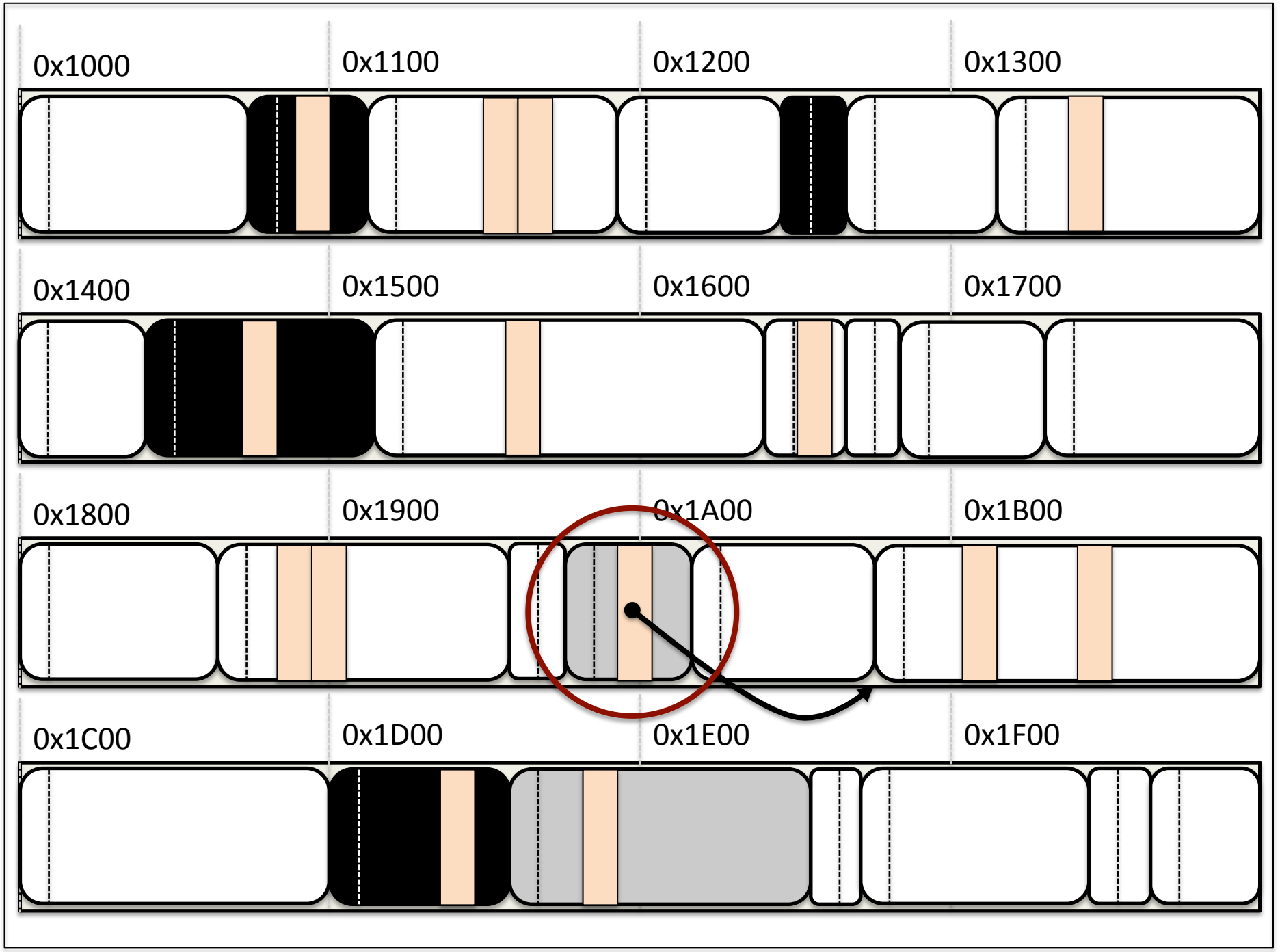

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

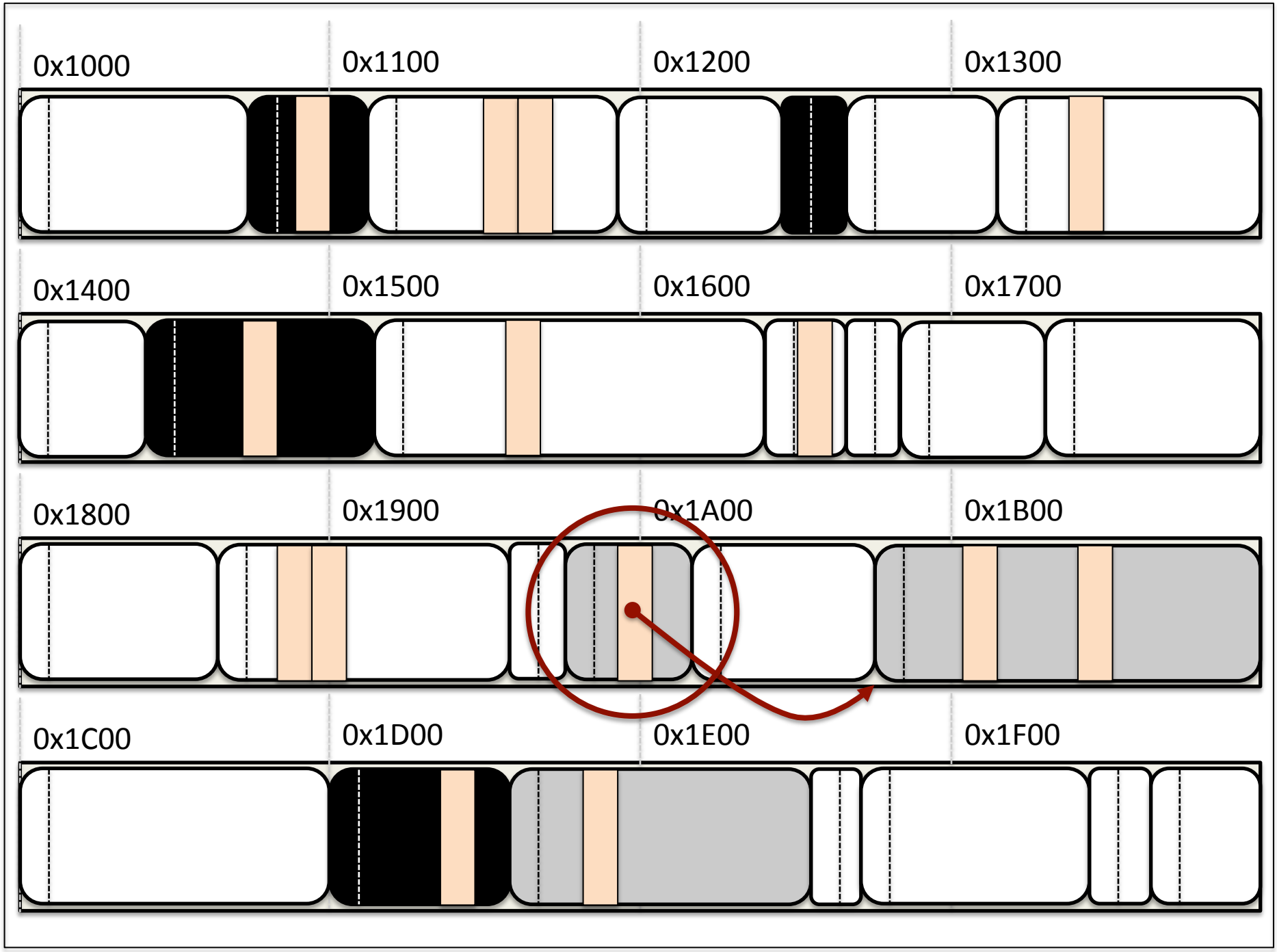

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

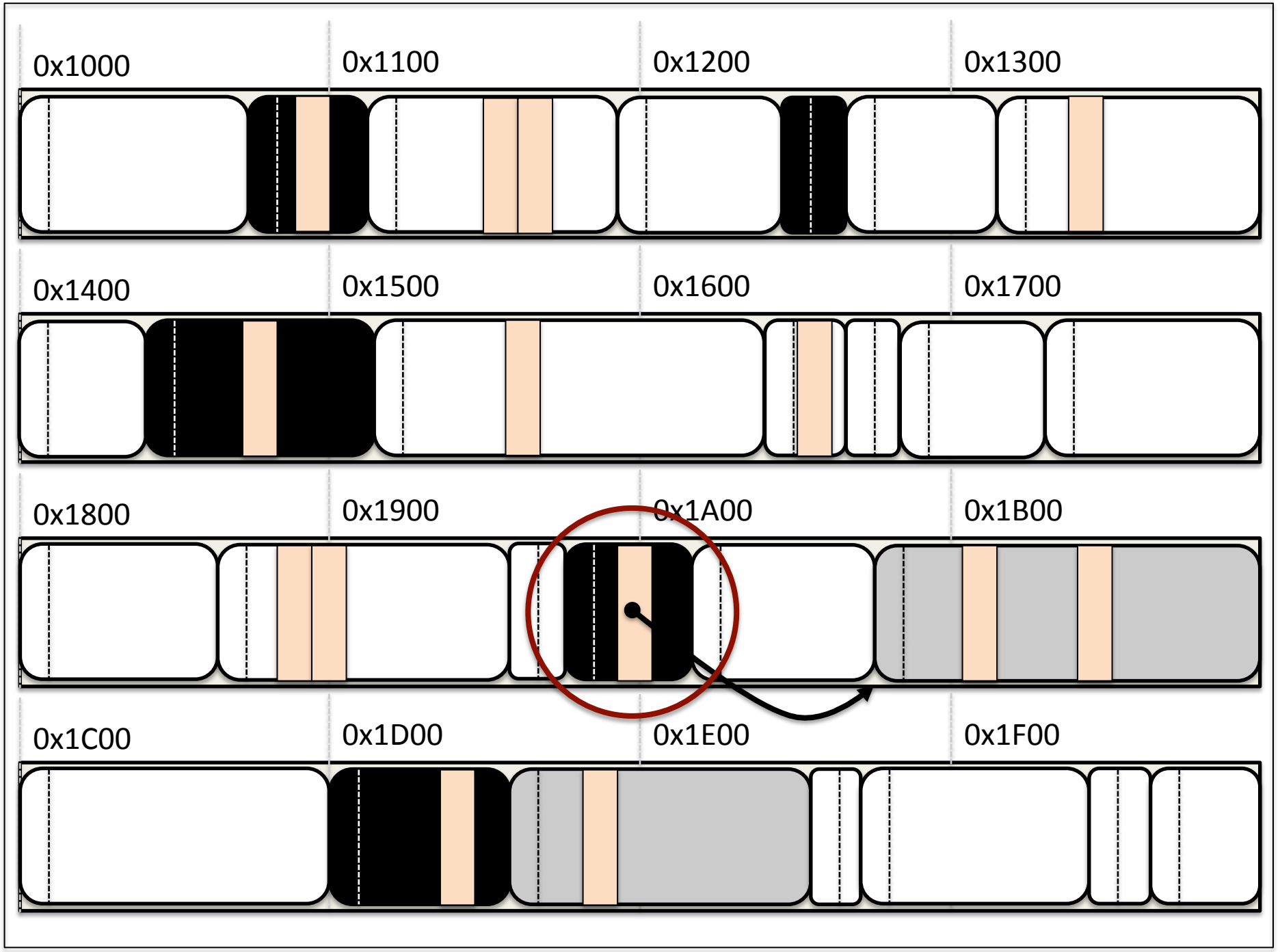

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

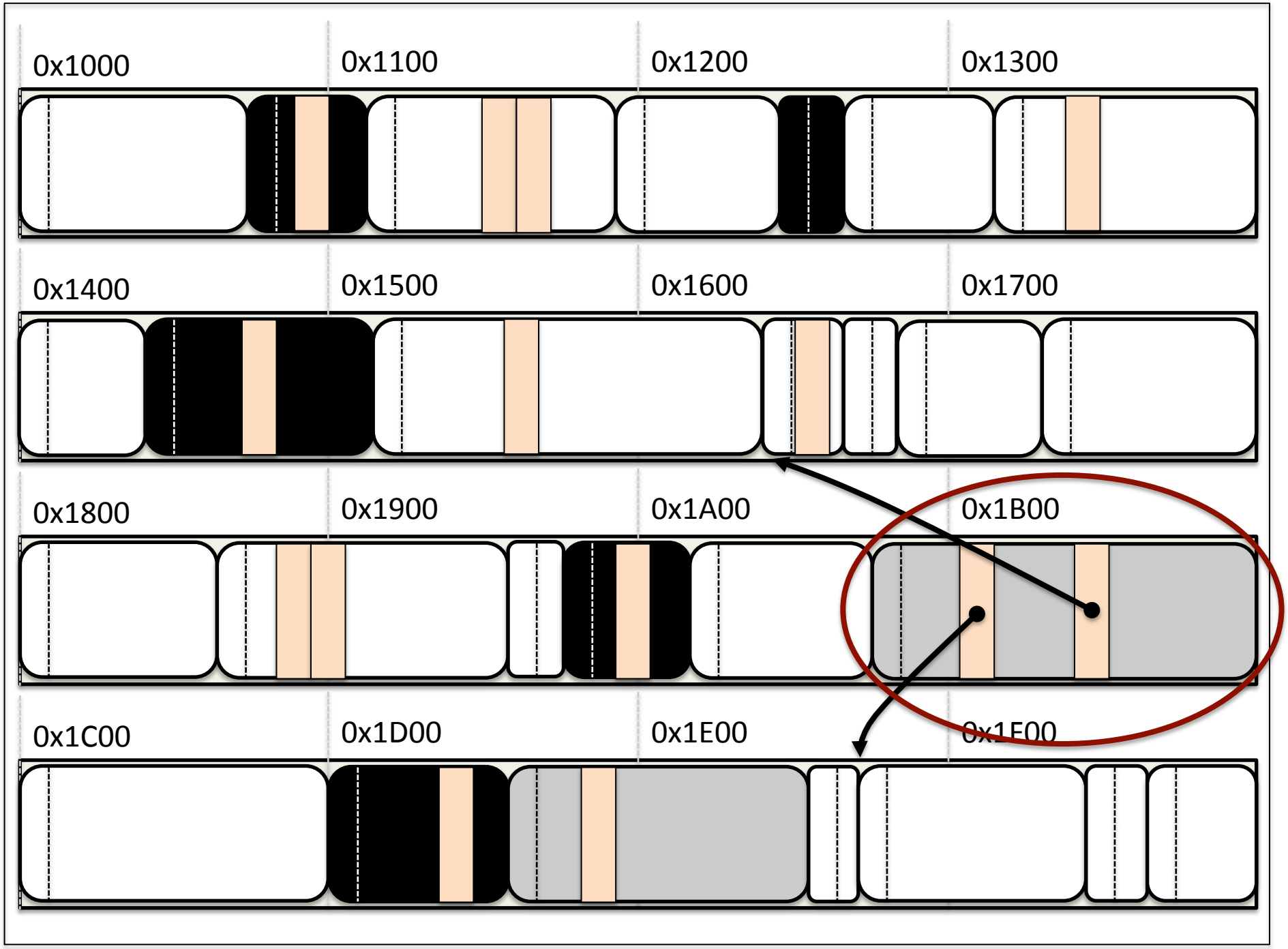

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

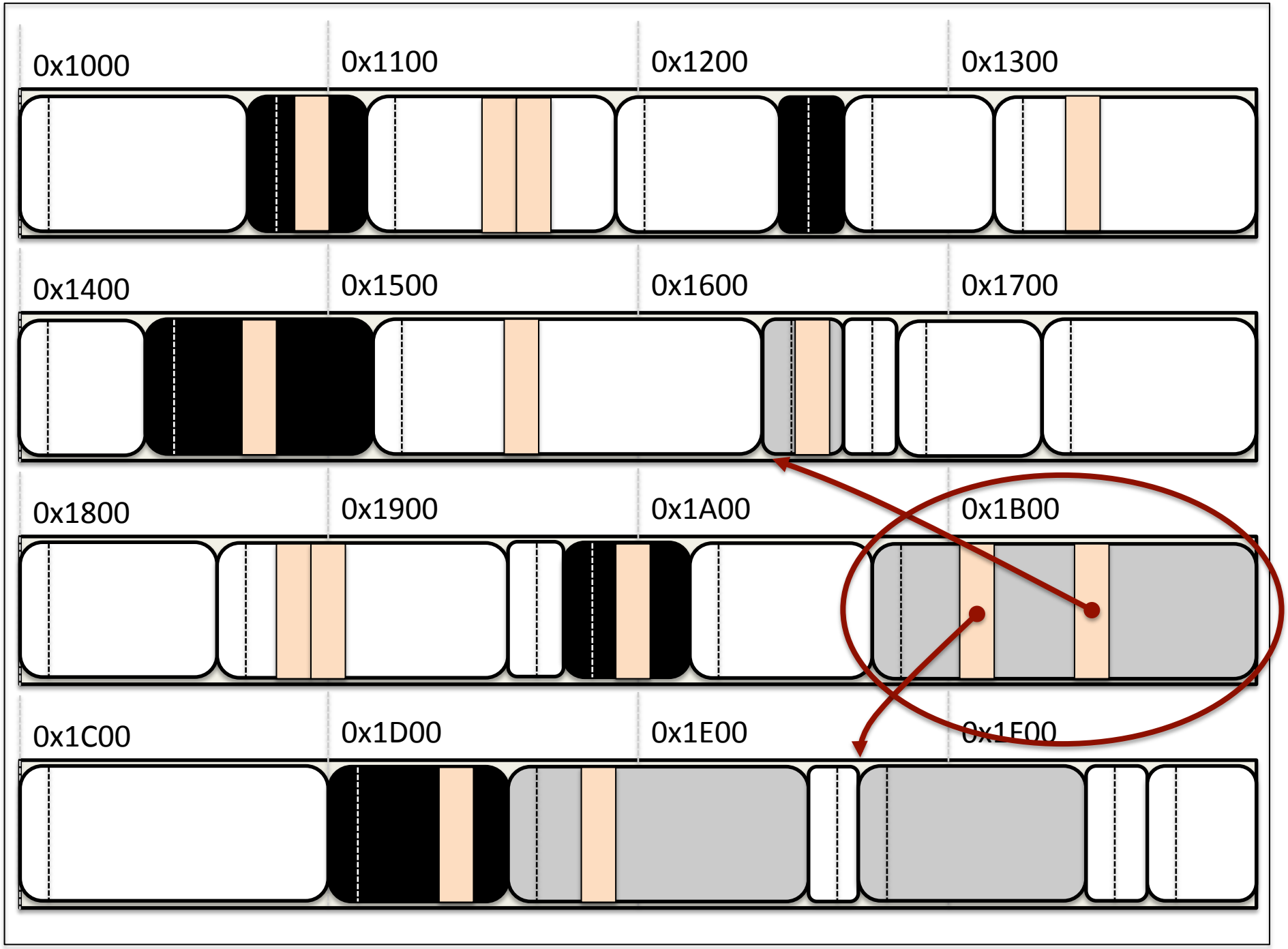

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

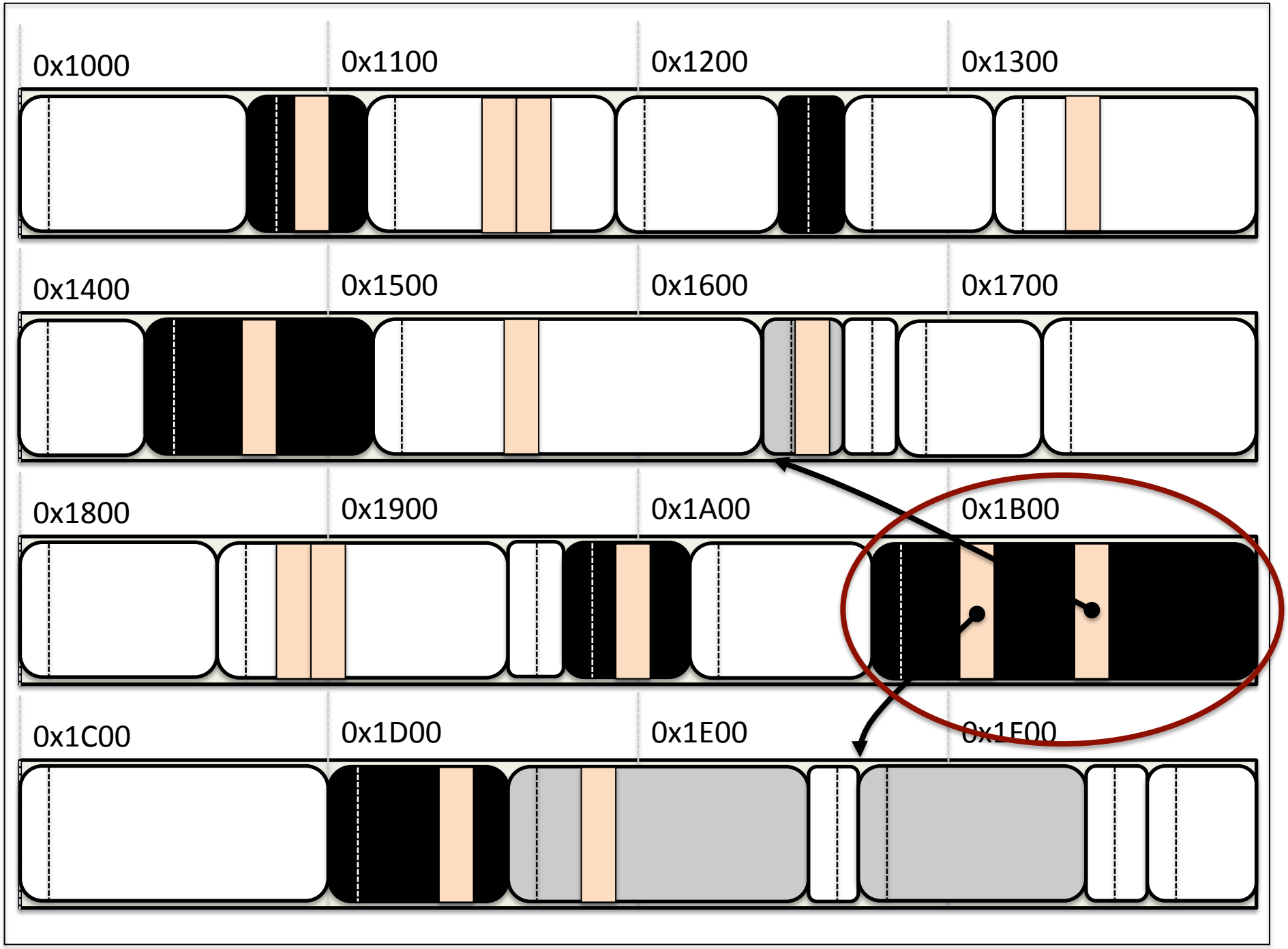

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

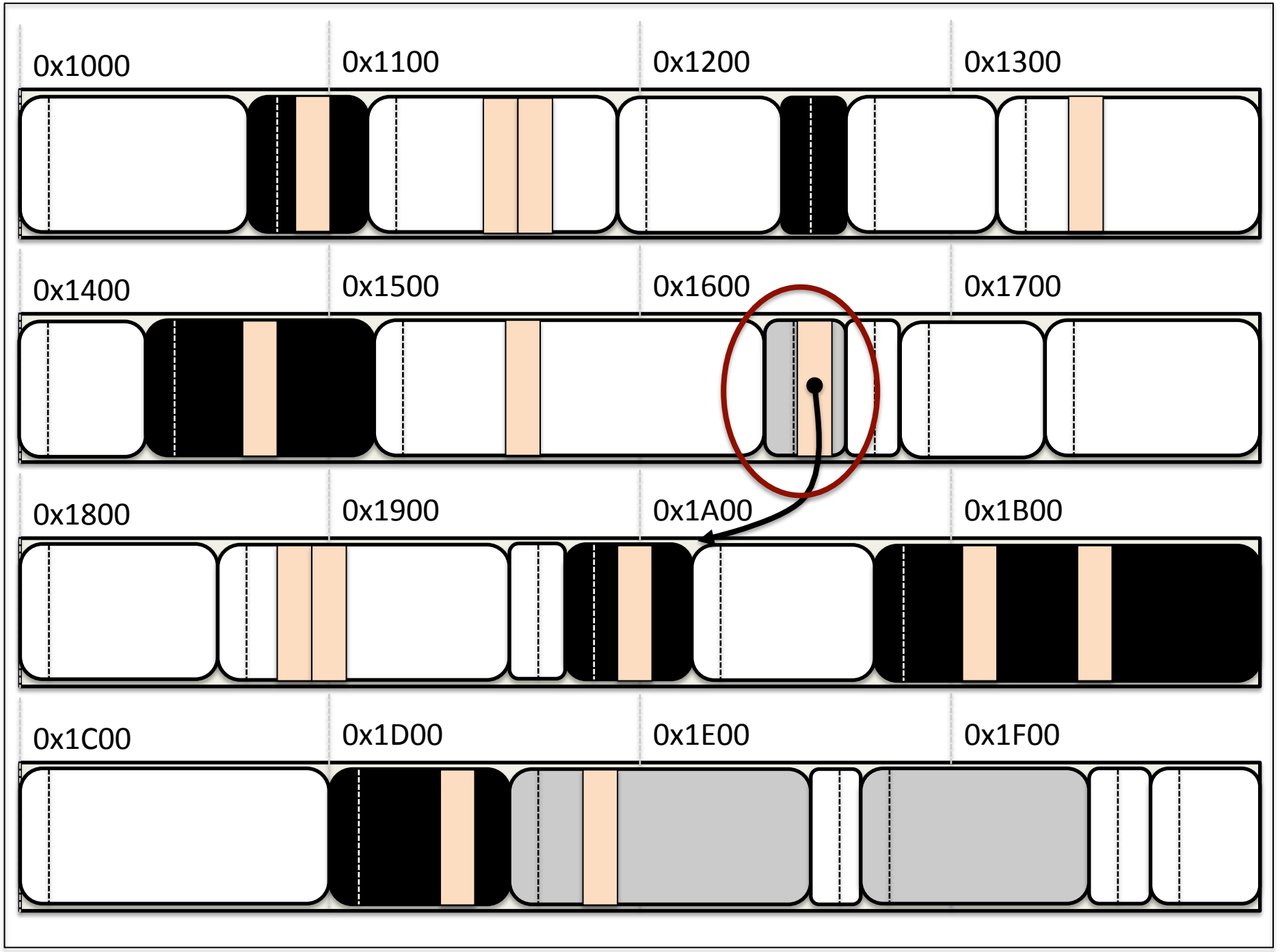

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

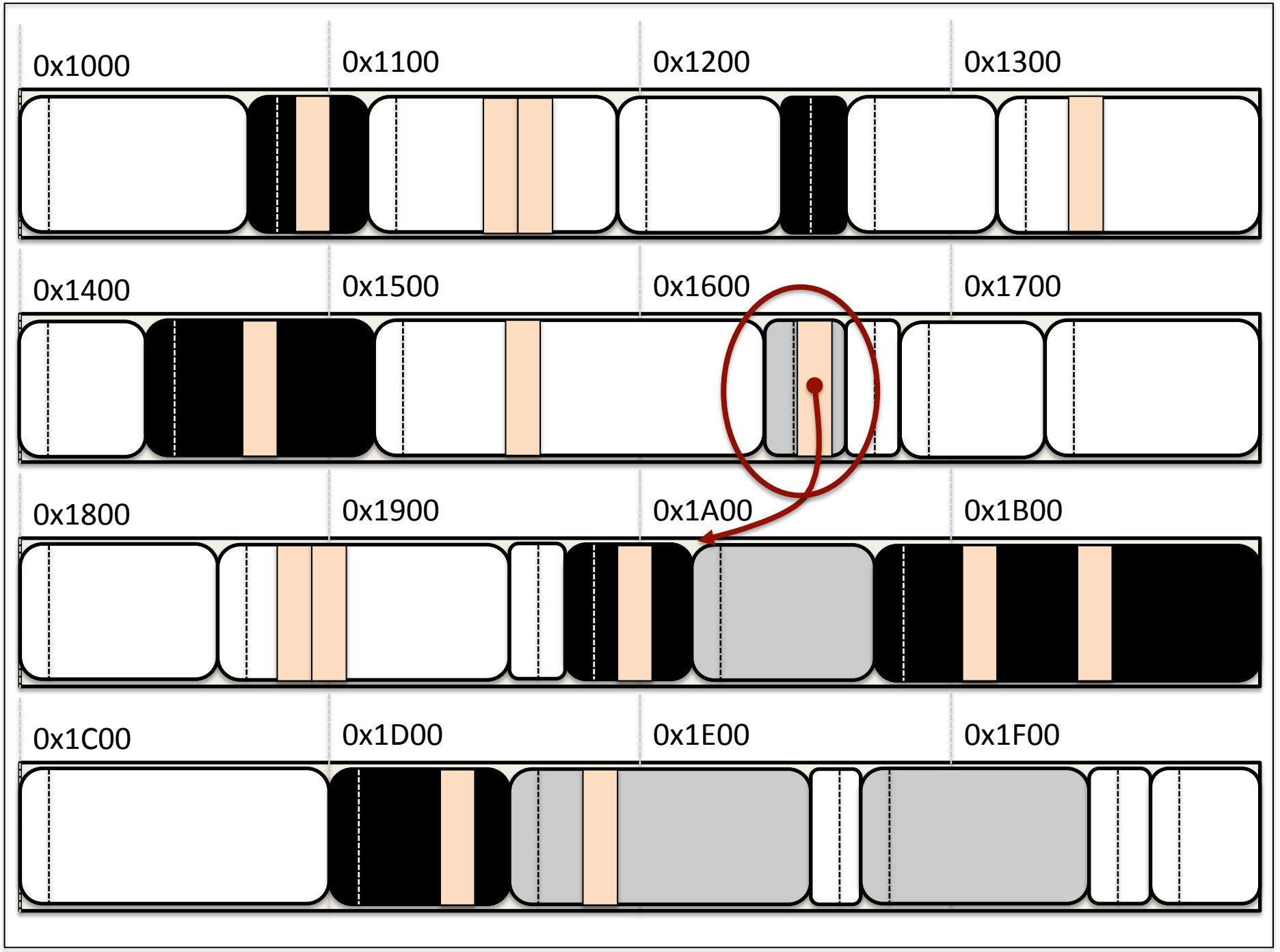

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

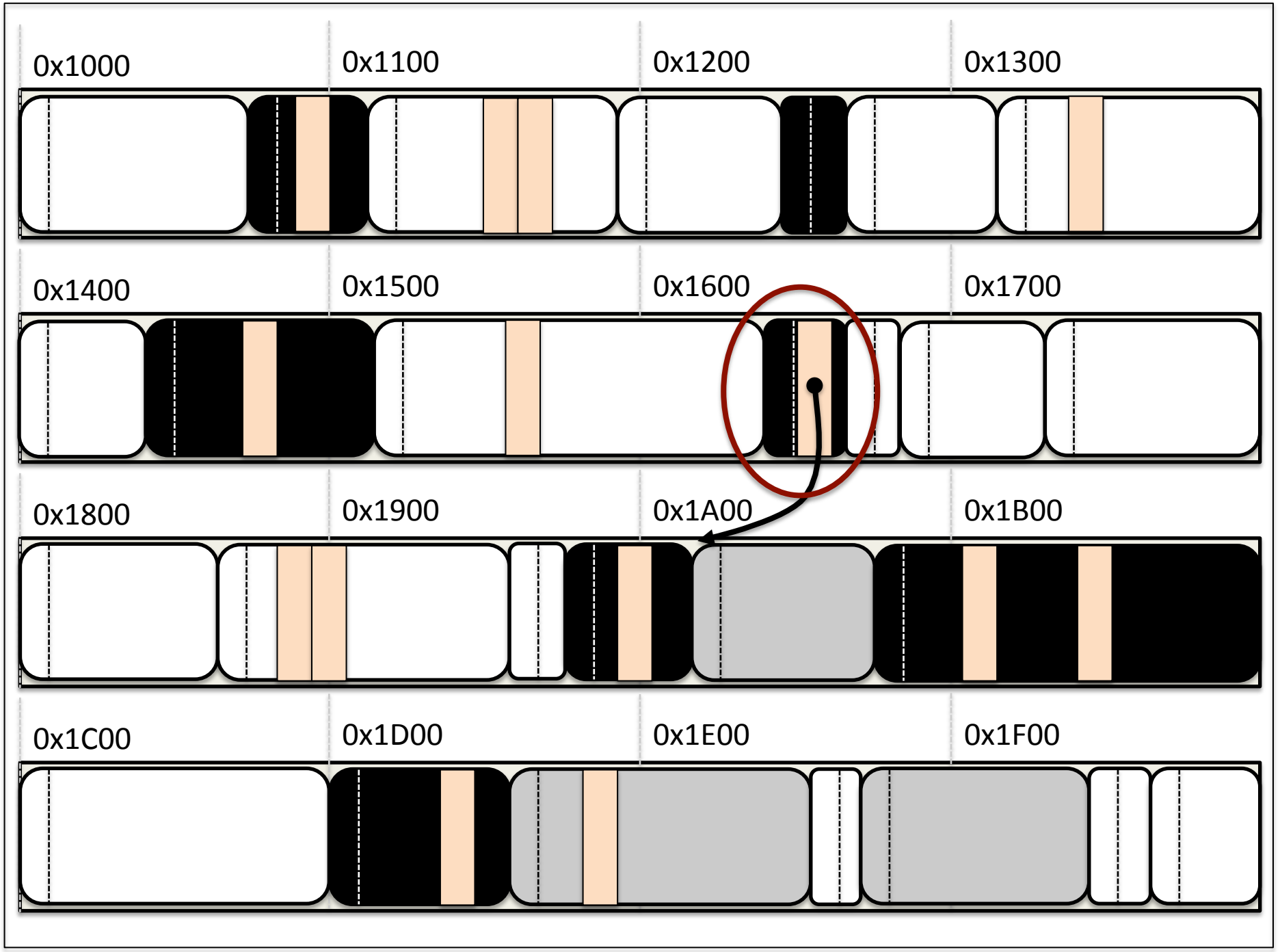

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

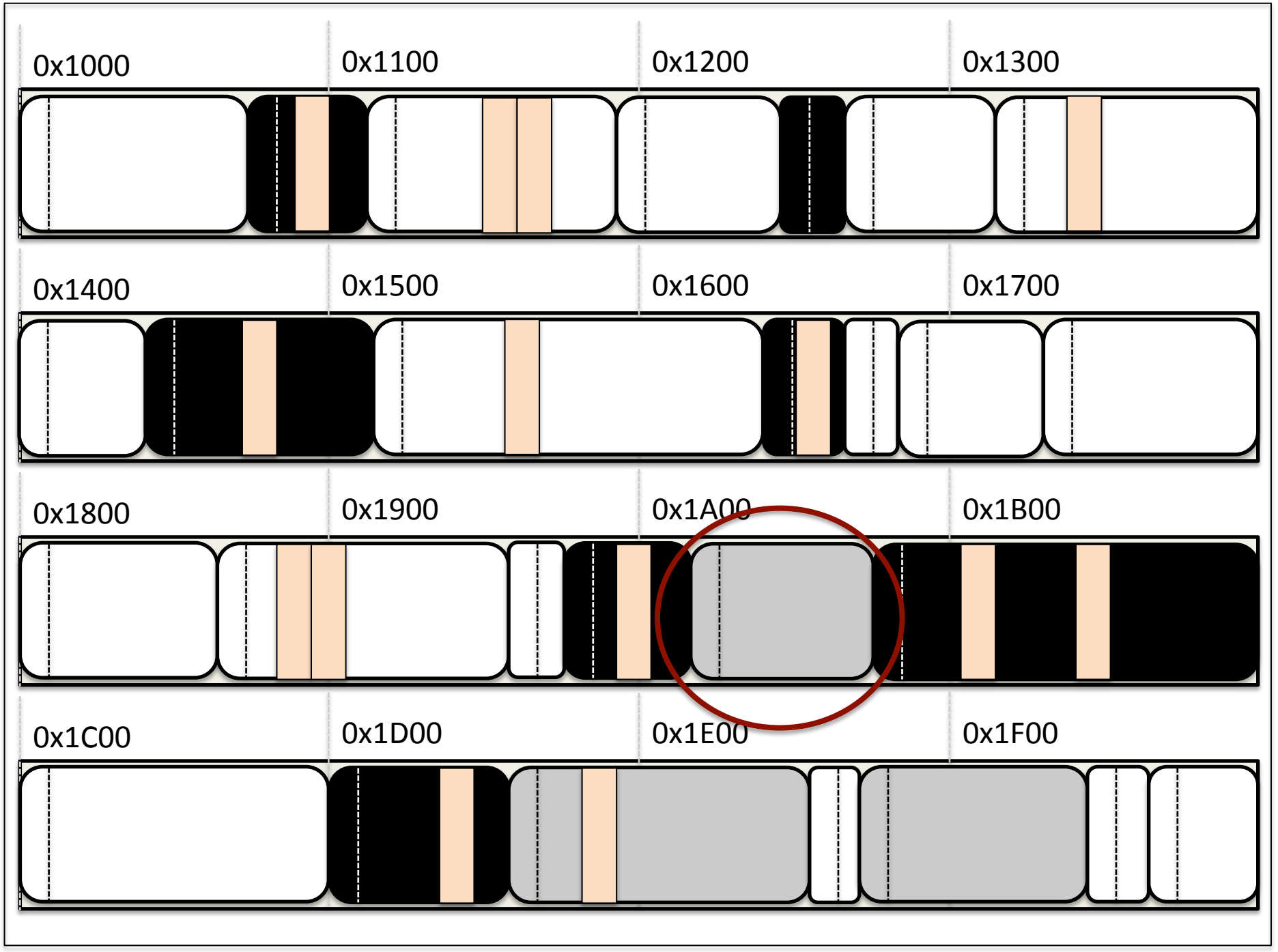

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

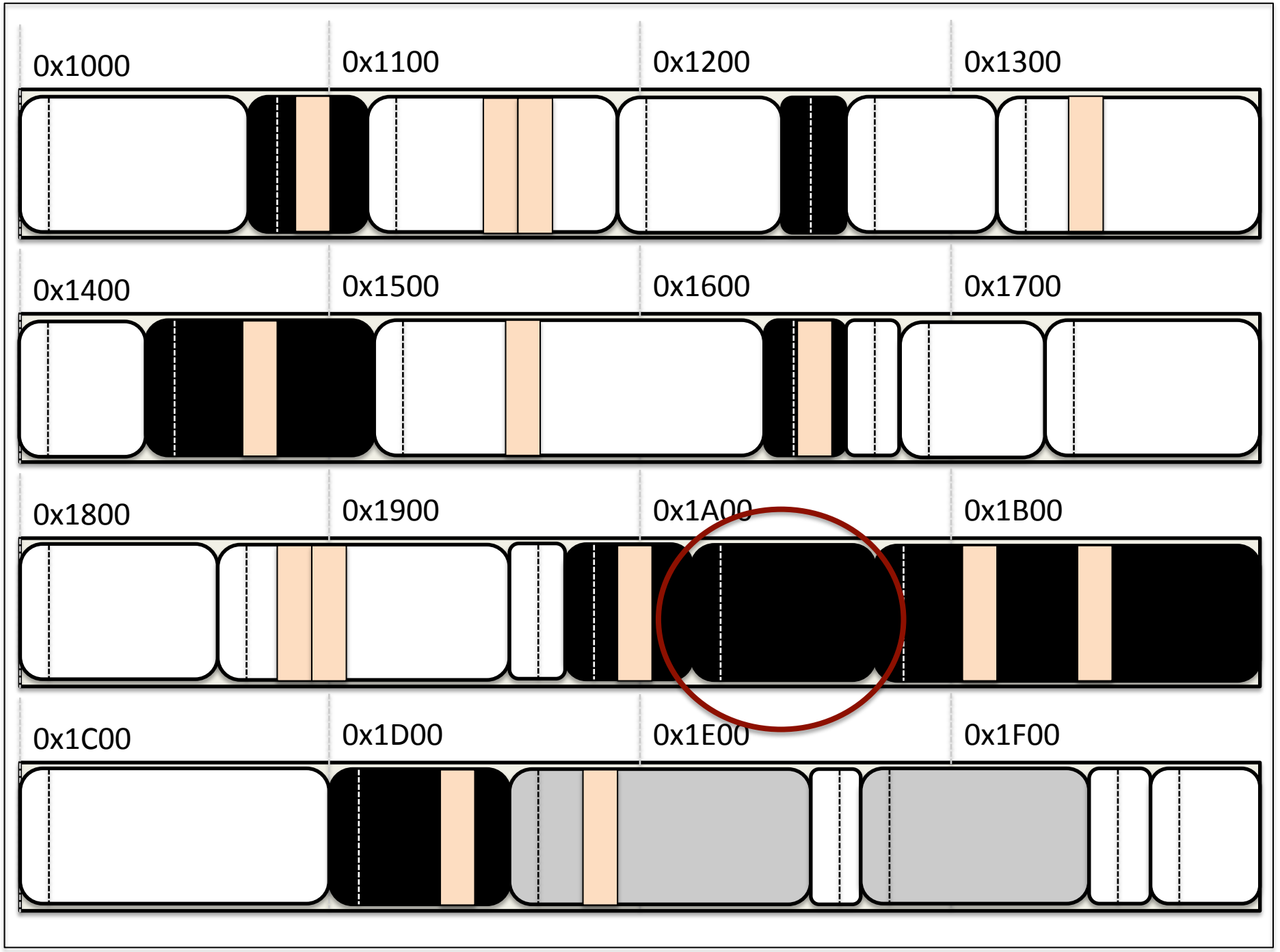

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

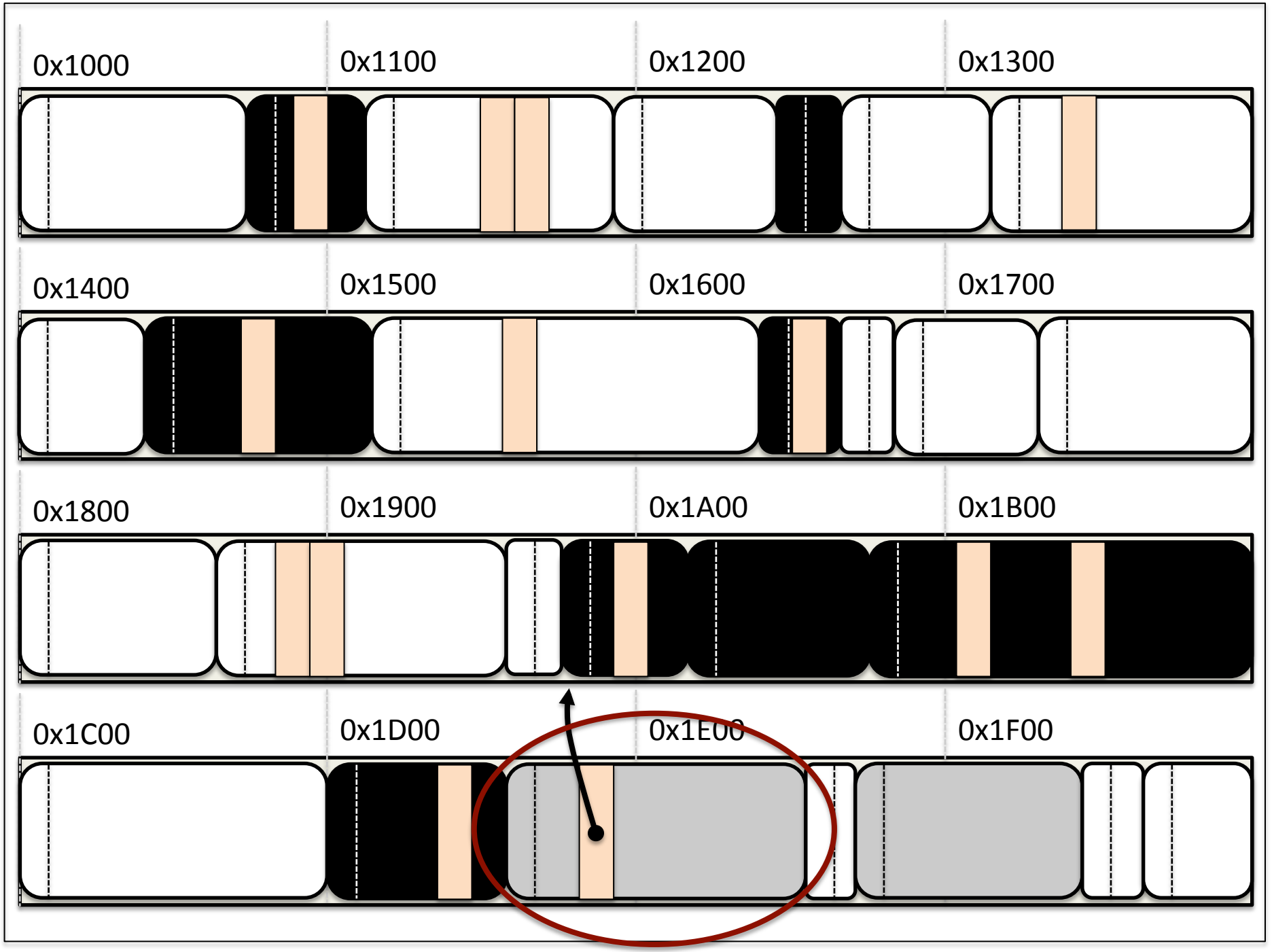

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

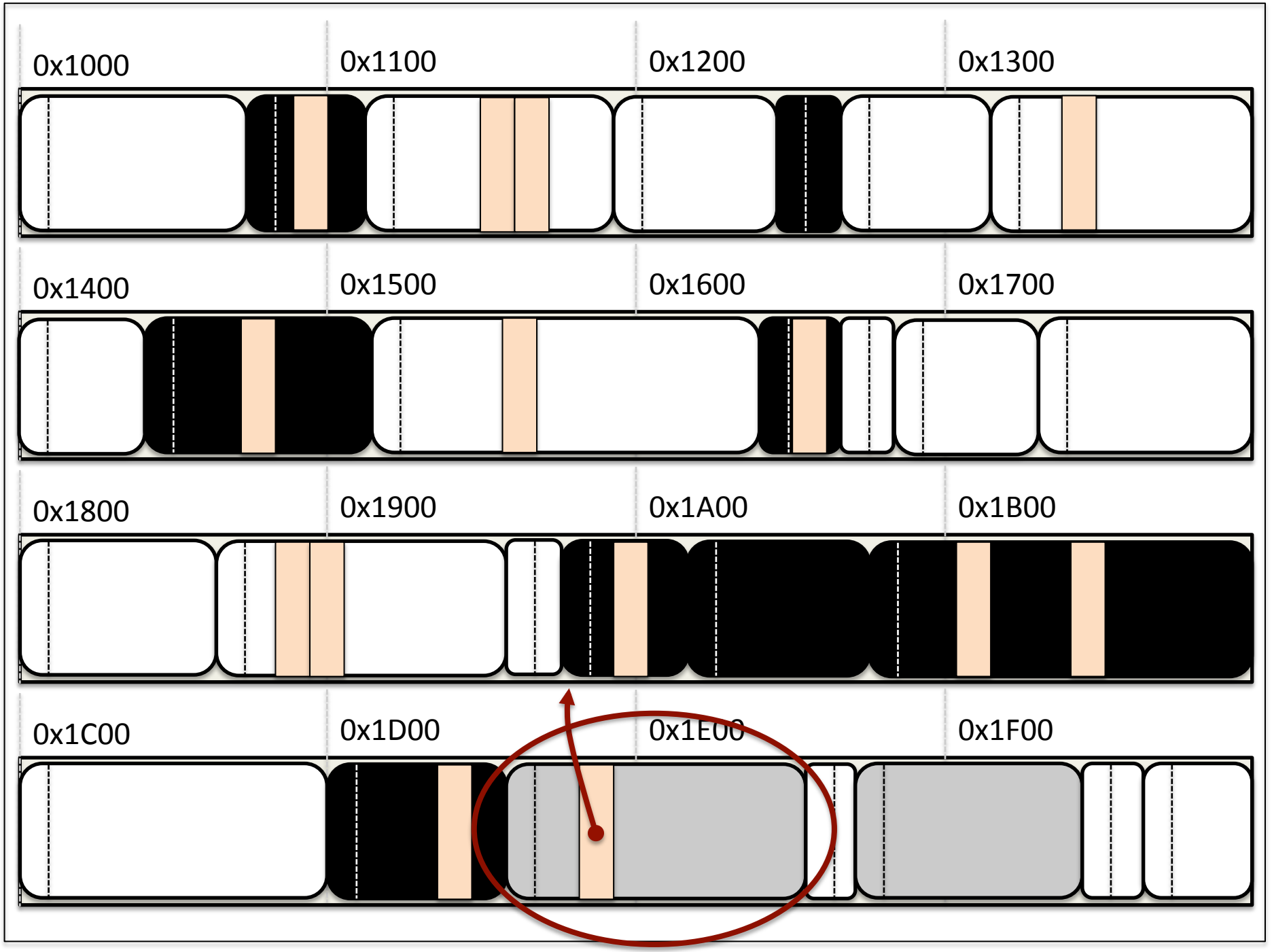

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

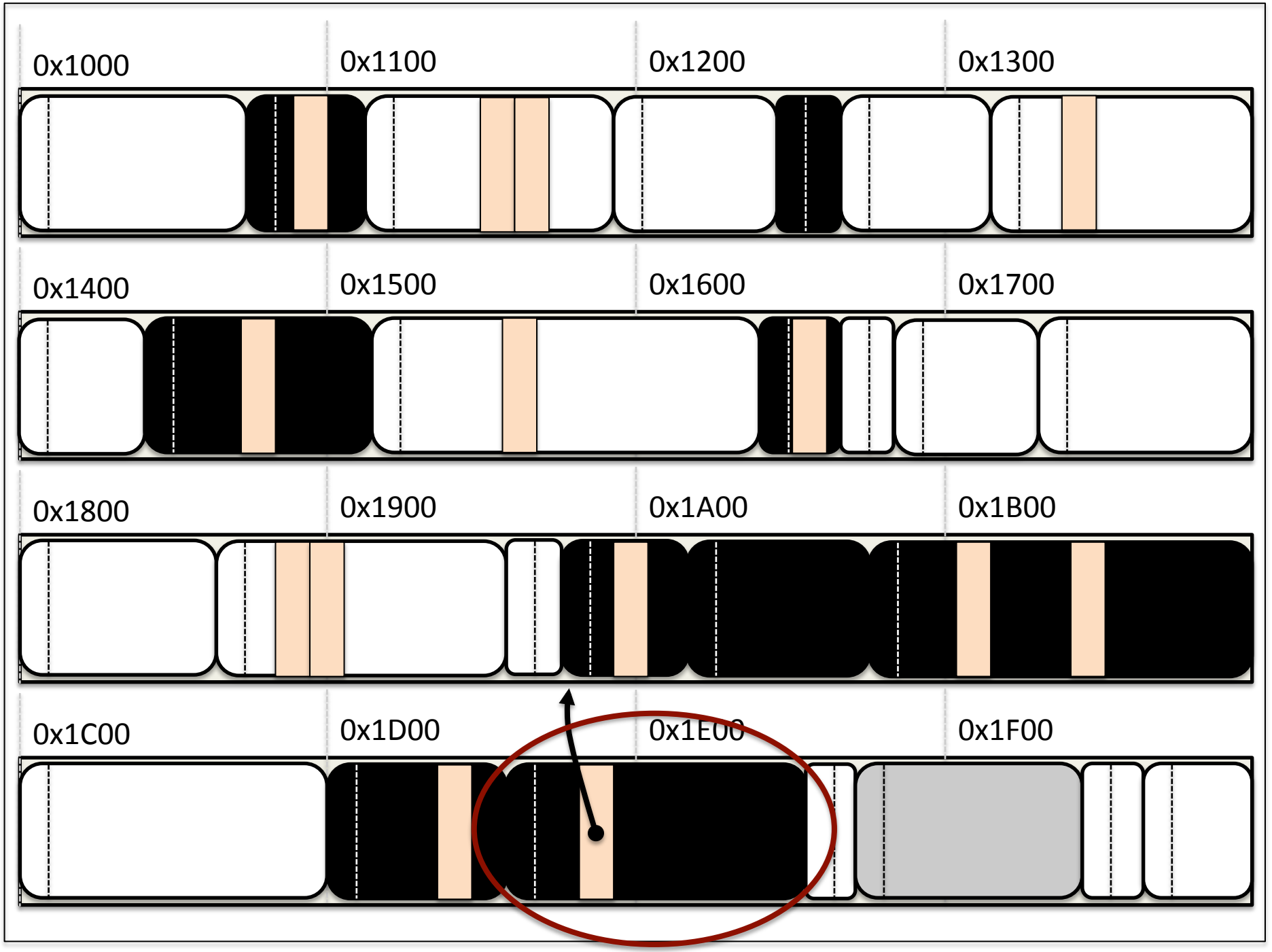

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

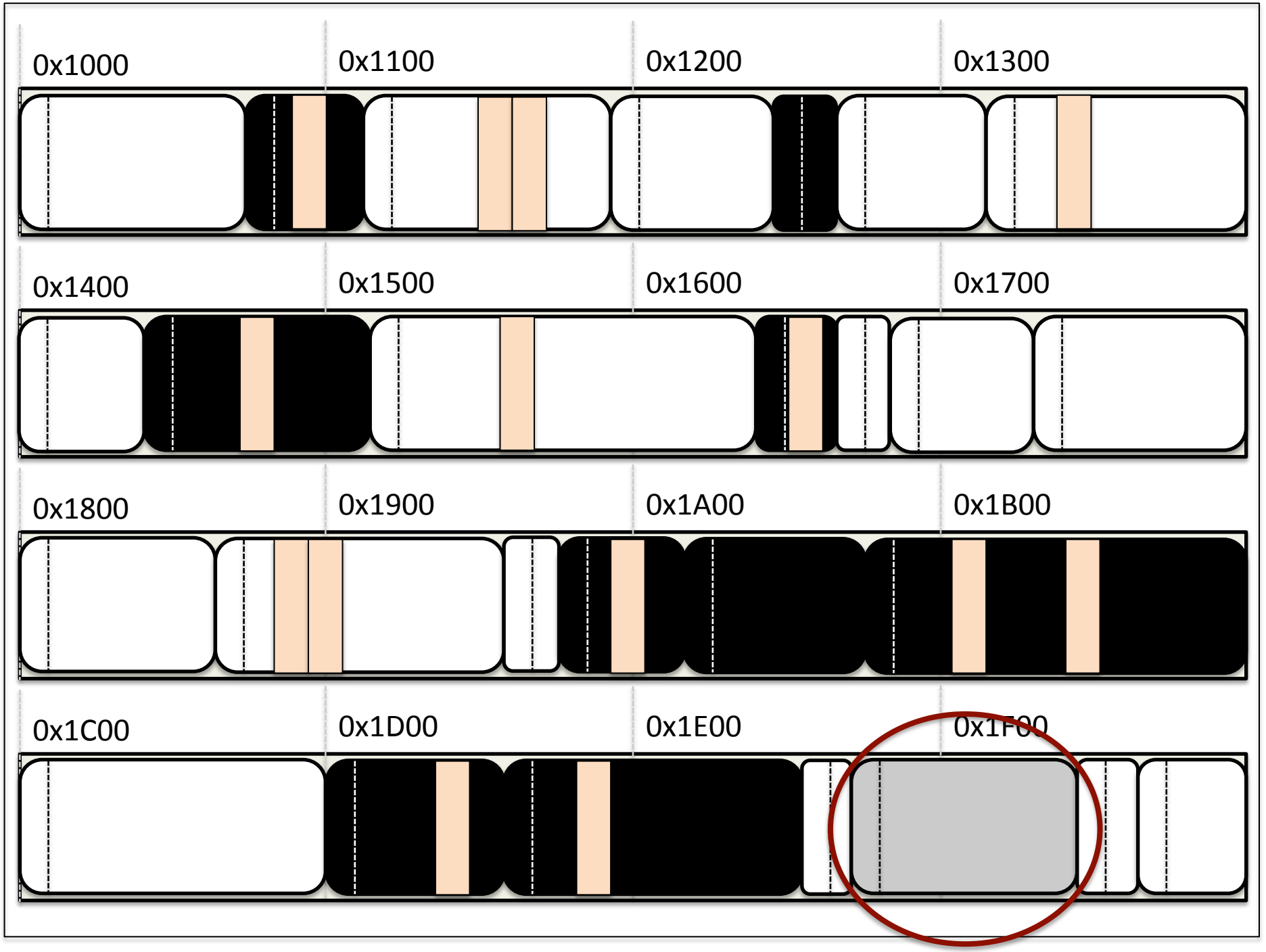

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

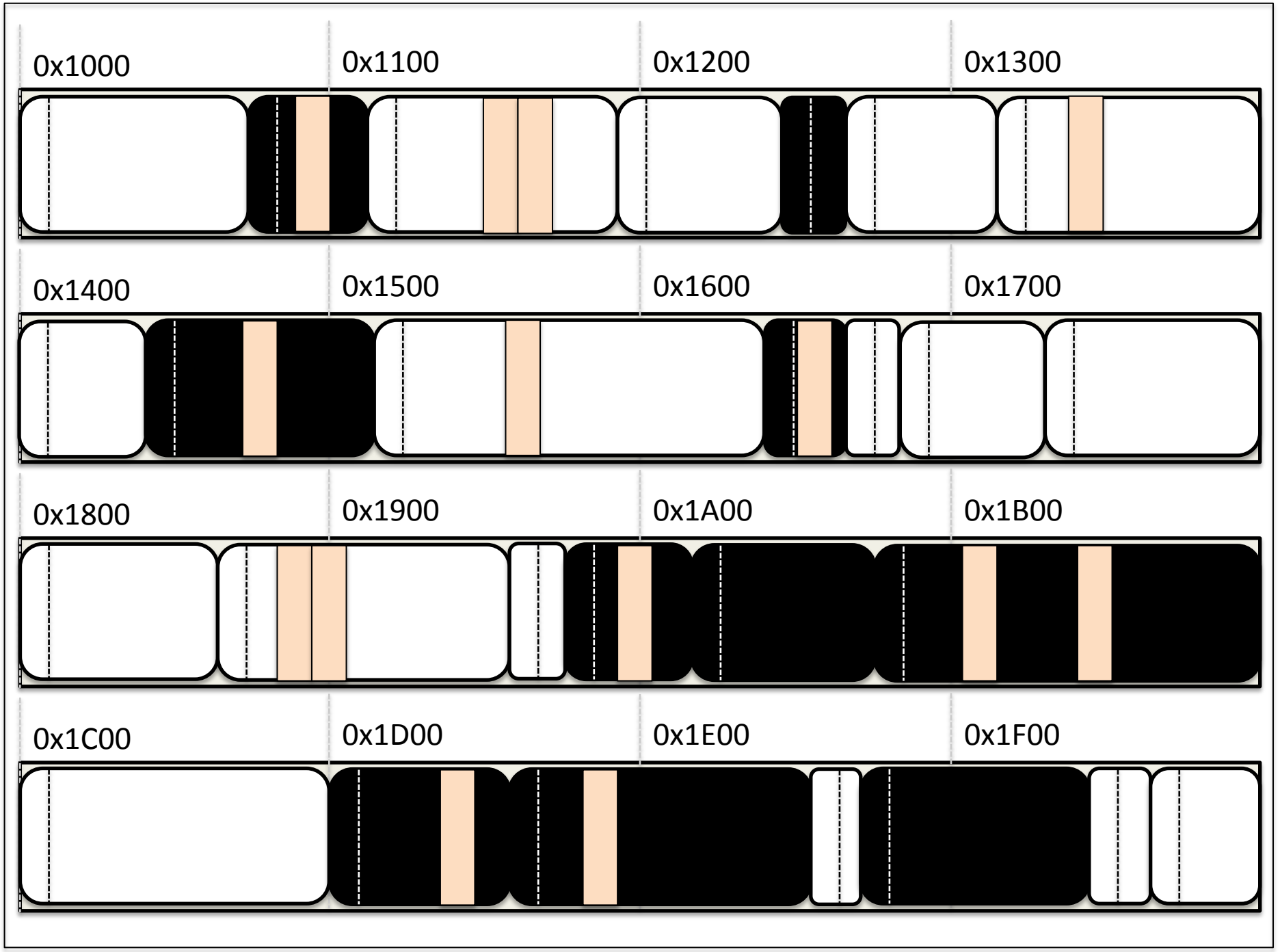

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

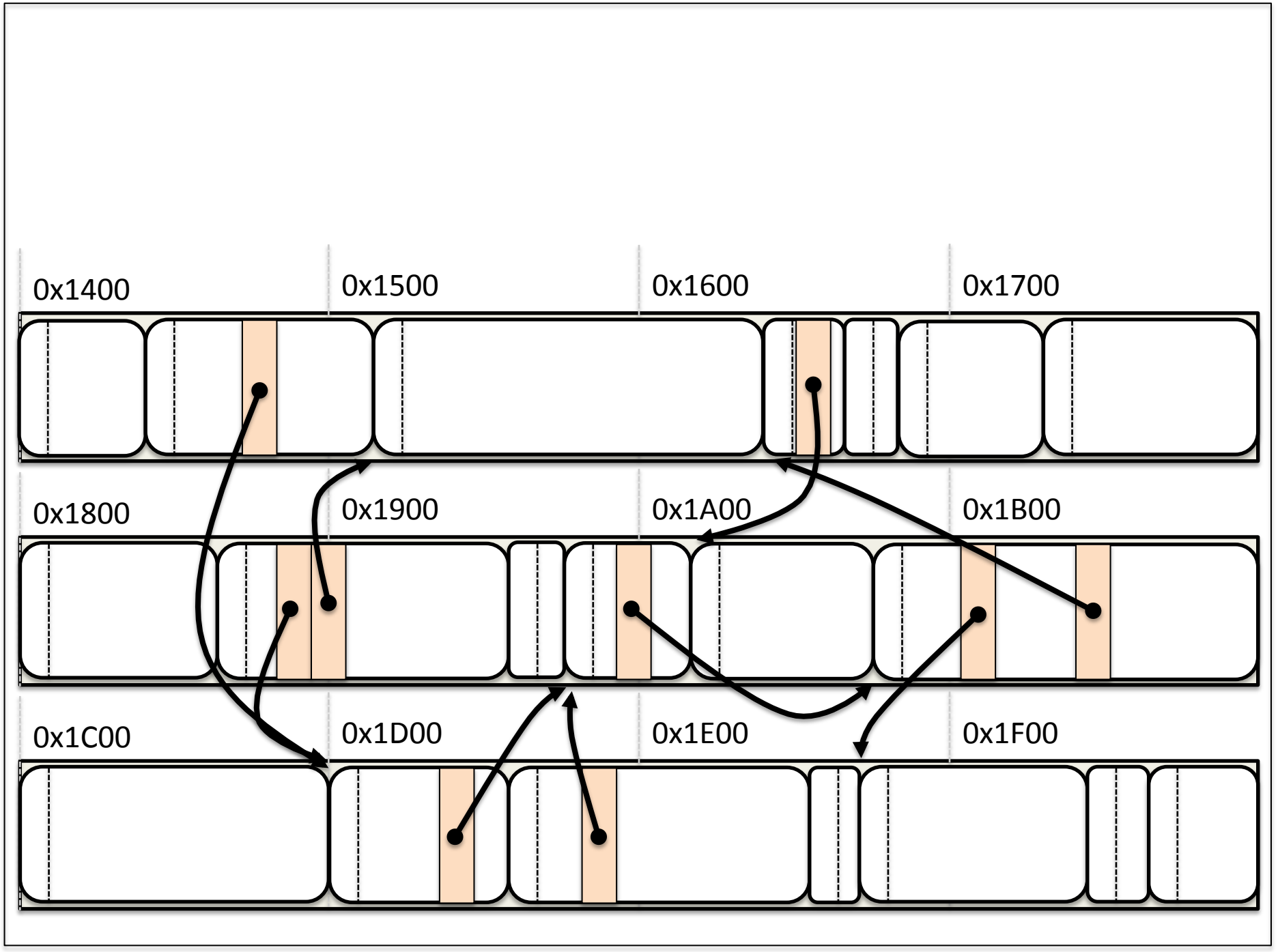

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

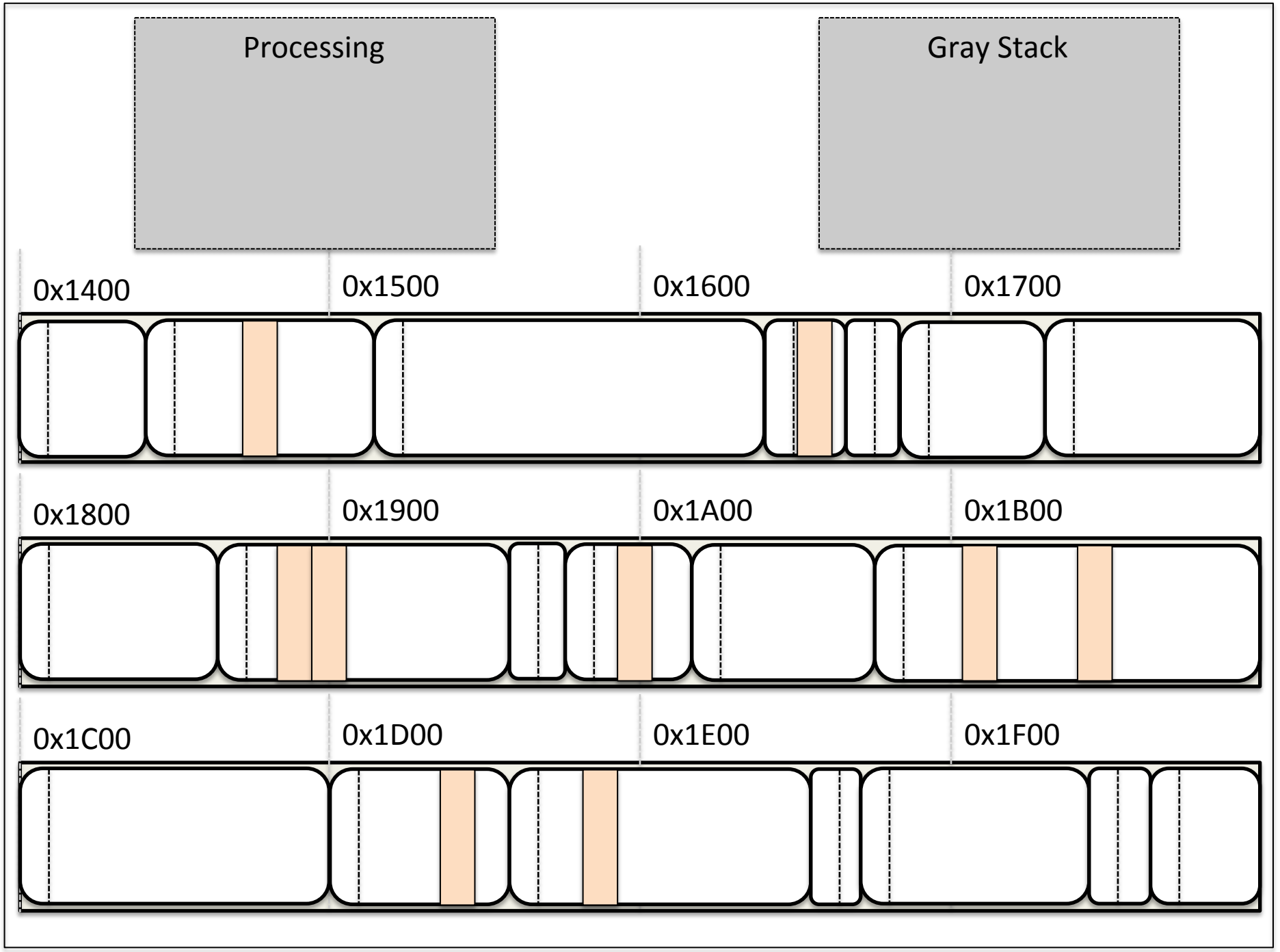
Tricolor Marking

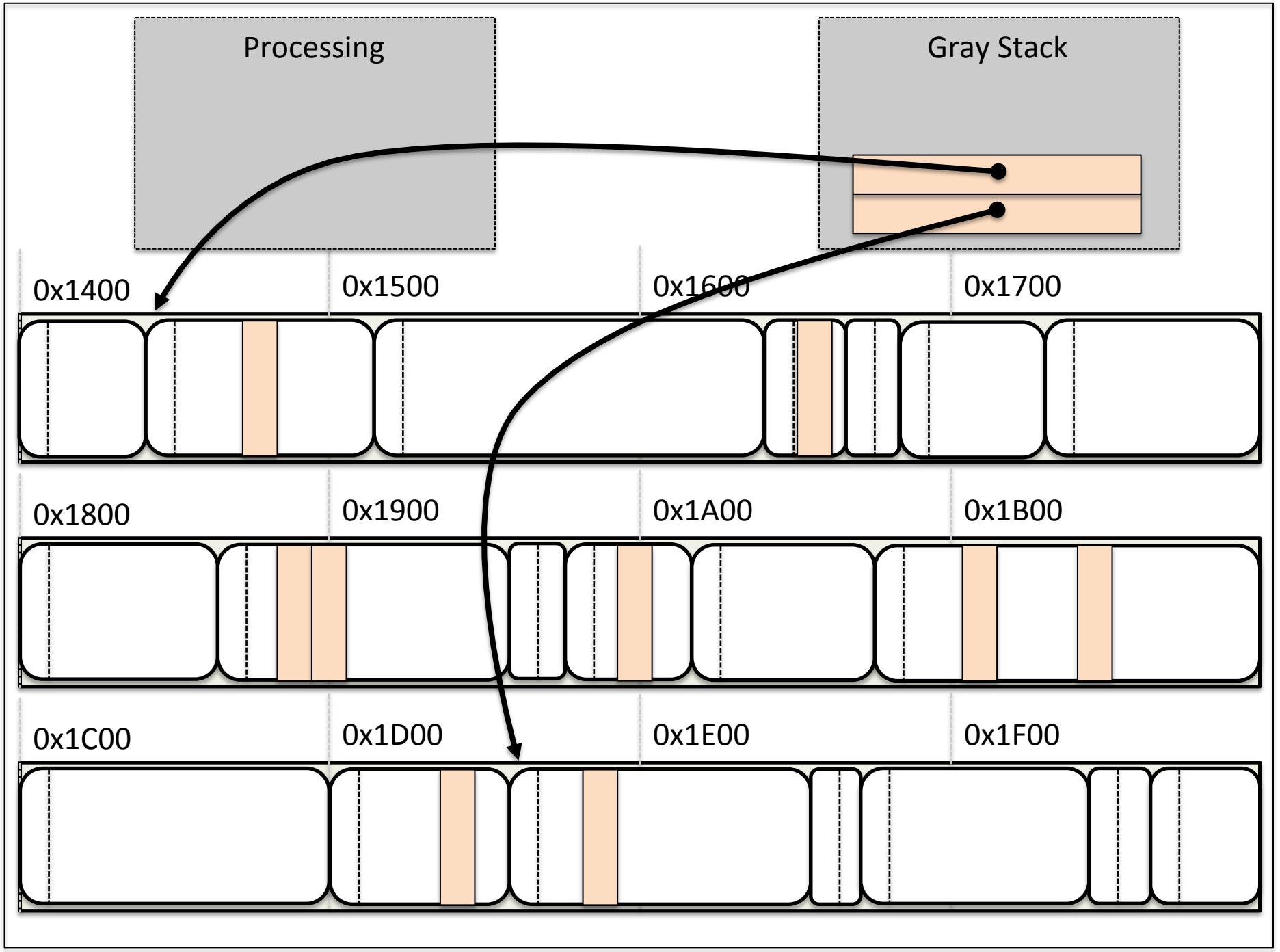
- Convenient way to think about marking
- Objects exist in three possible states
  - White
  - Gray
  - Black
- Maintain invariants between colors
  - Black objects can't point to white
  - Objects cannot go from black or gray to white

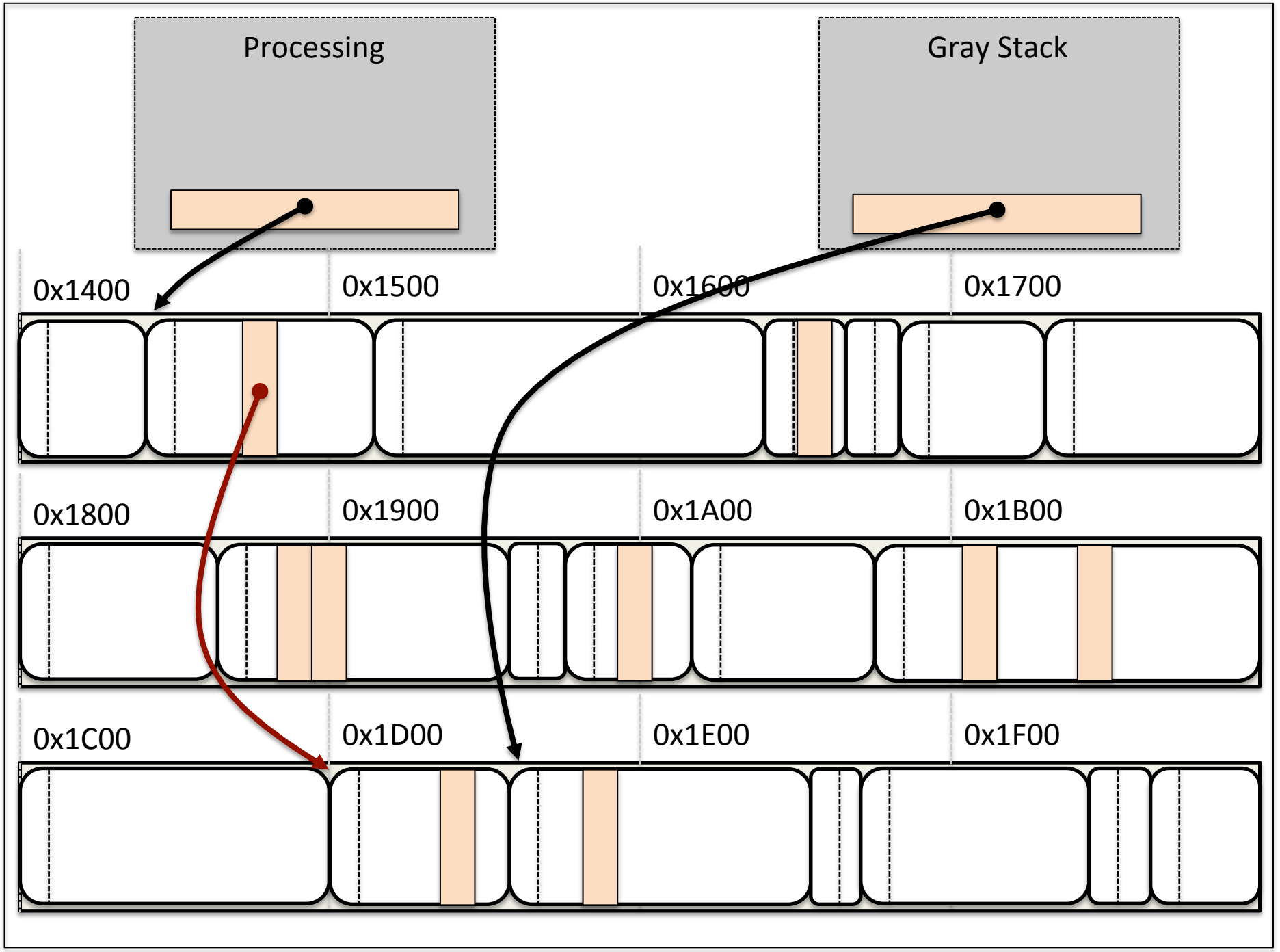
Tracing Implementation

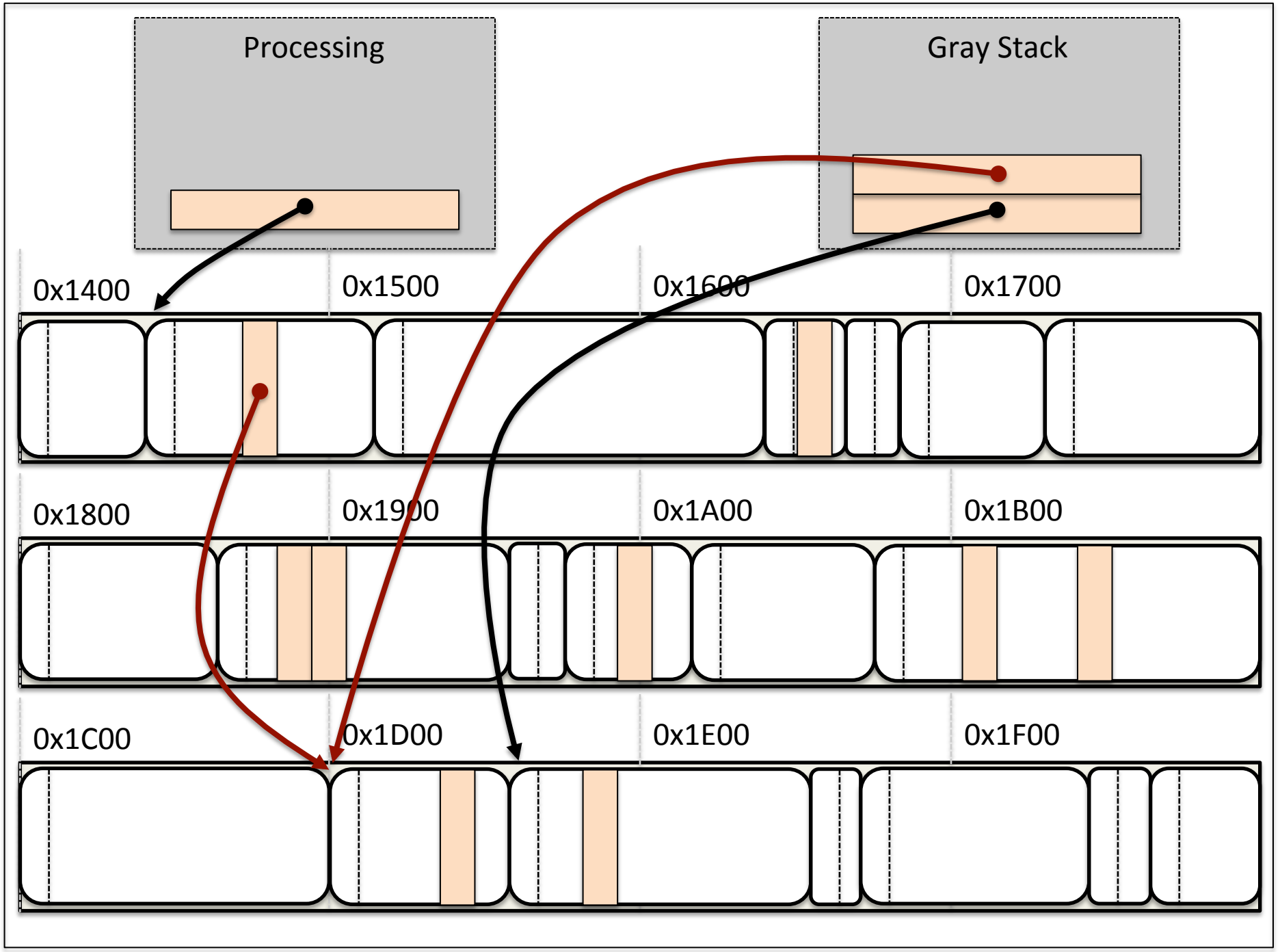
- How do you know the color of objects?
  - Can't look at the whole heap to check
- Track gray objects on a stack
  - Push when an object is set to gray
  - Pop when it is scanned
  - If an object is already marked, don't push it
  - Tracing done when the gray stack is empty

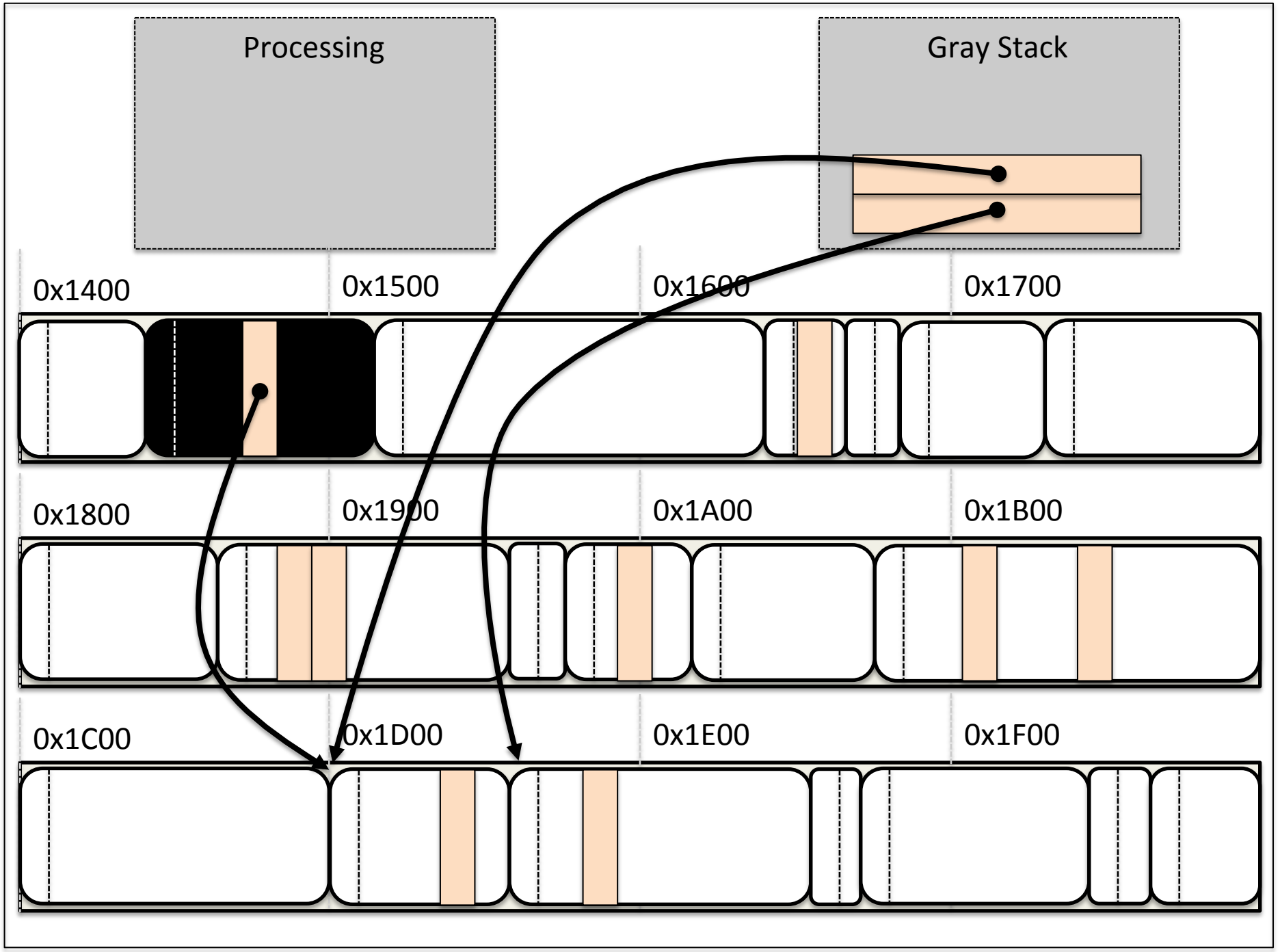

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

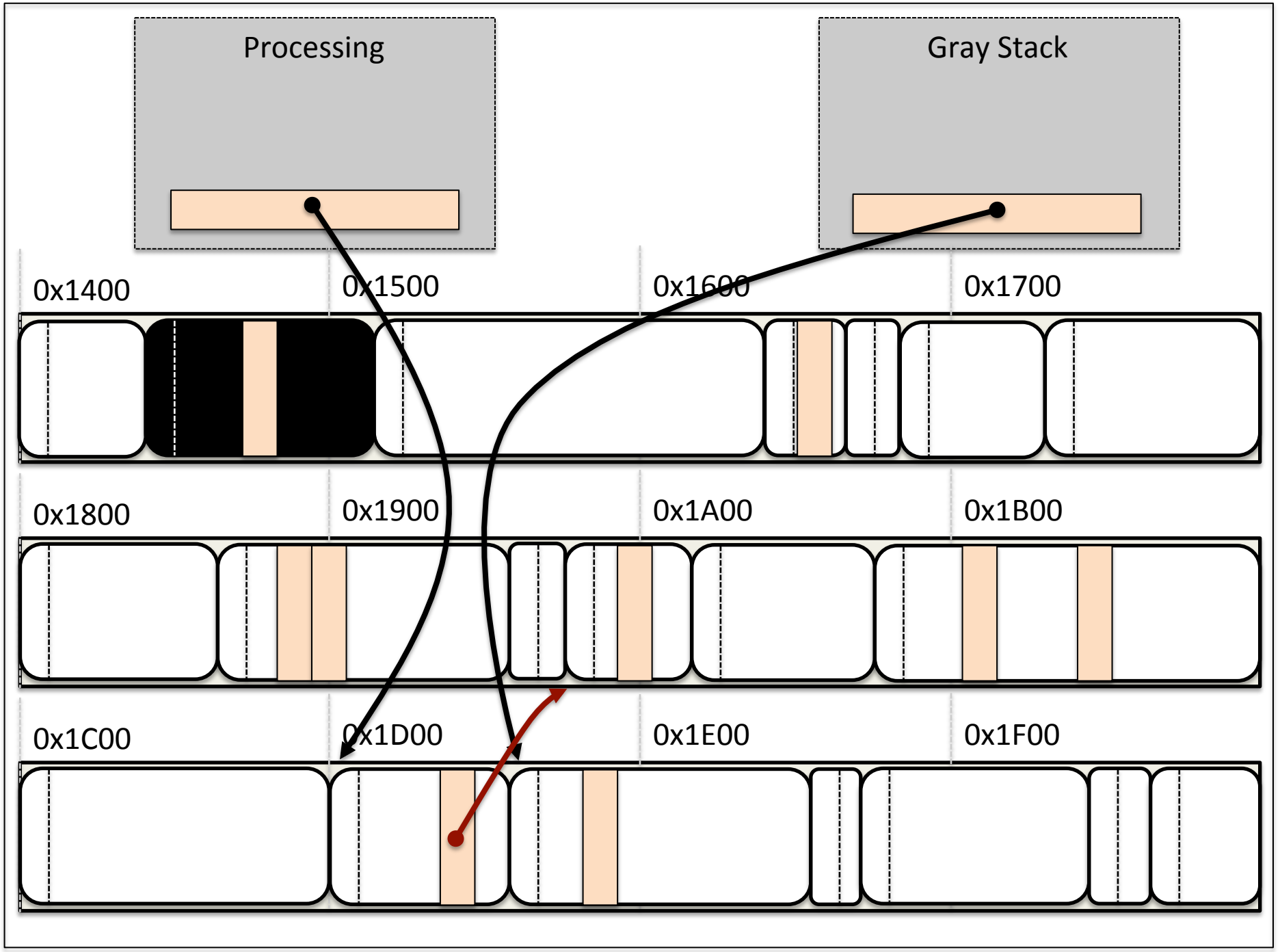

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

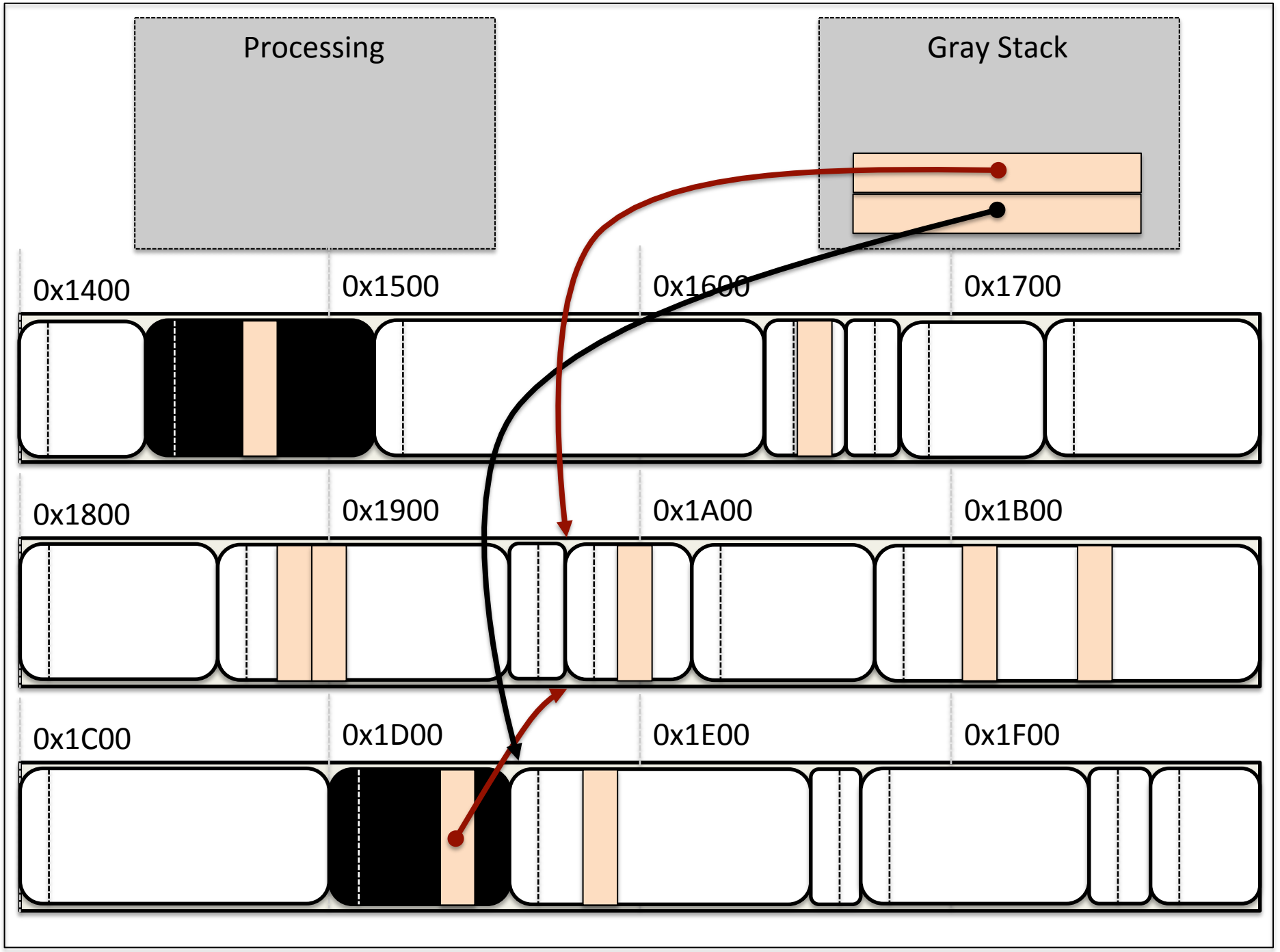

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

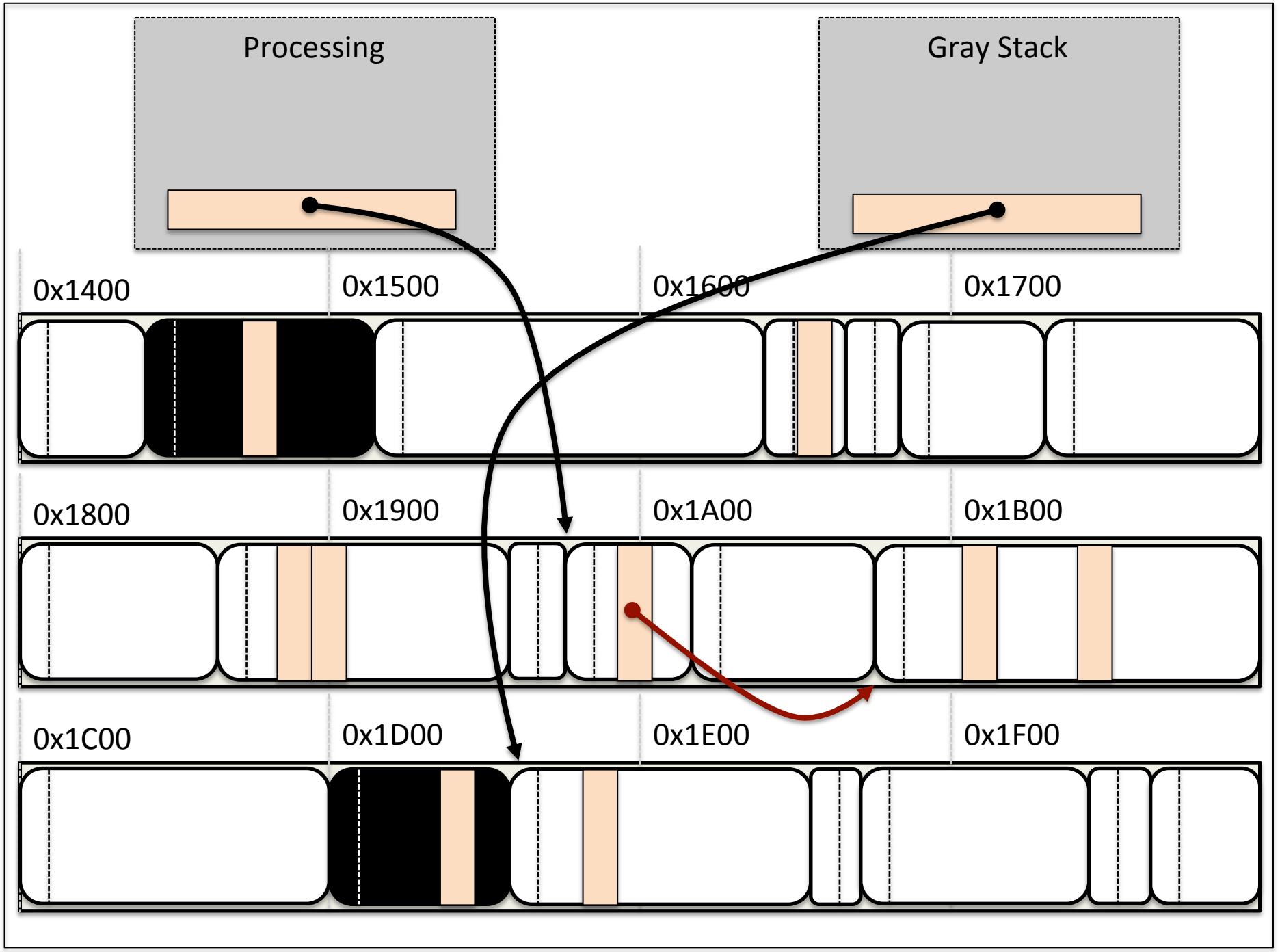

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

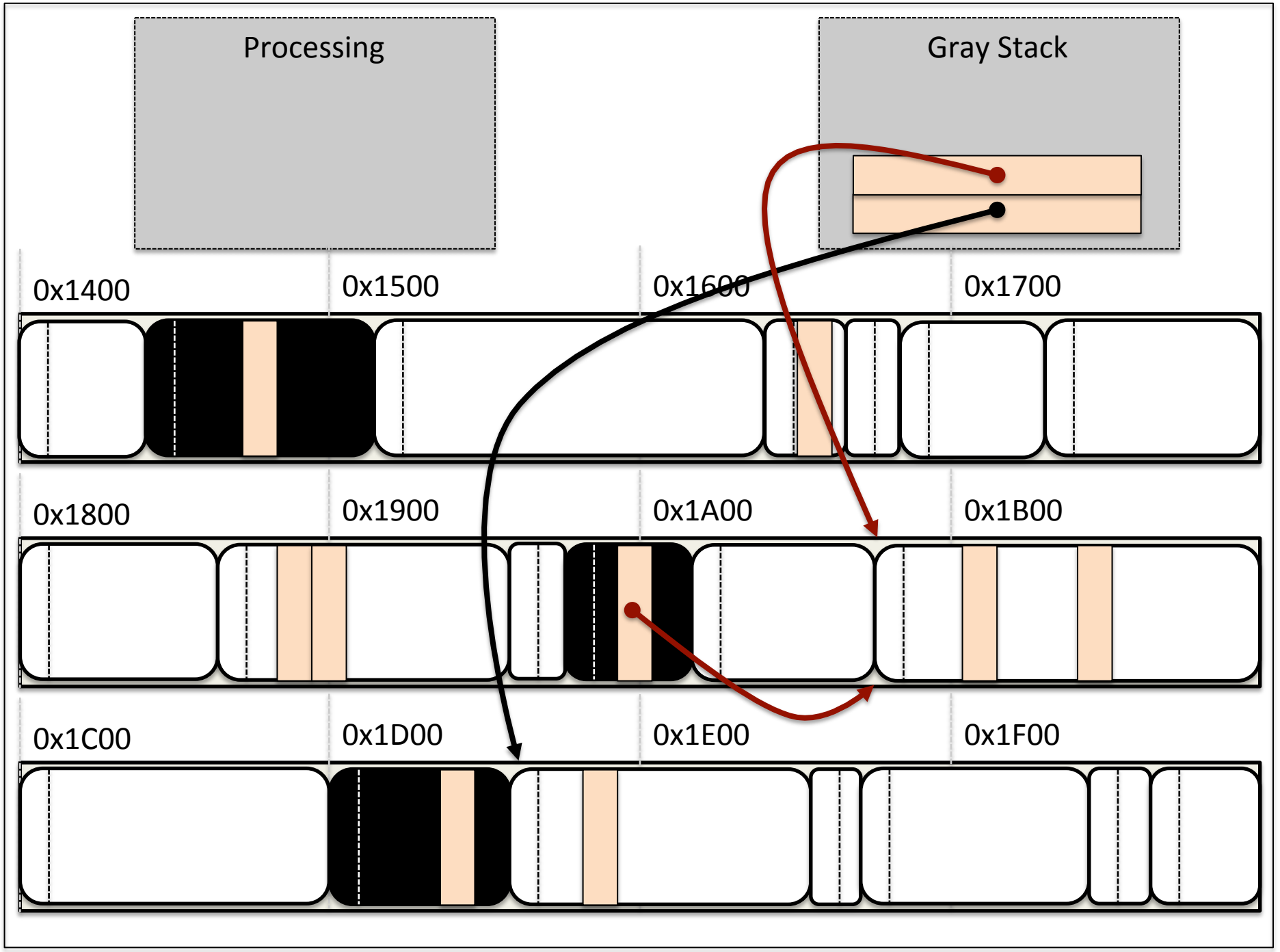

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

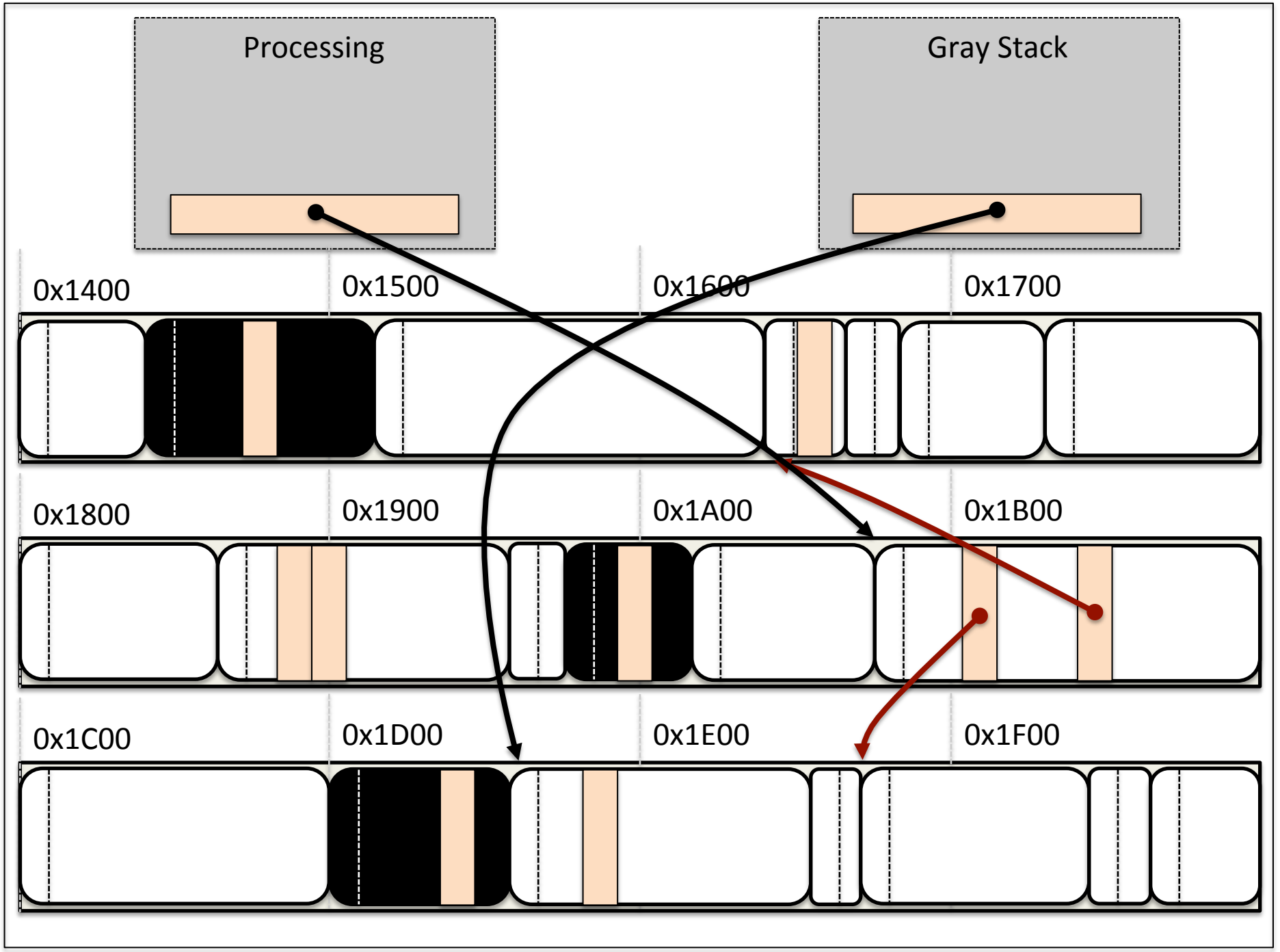

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

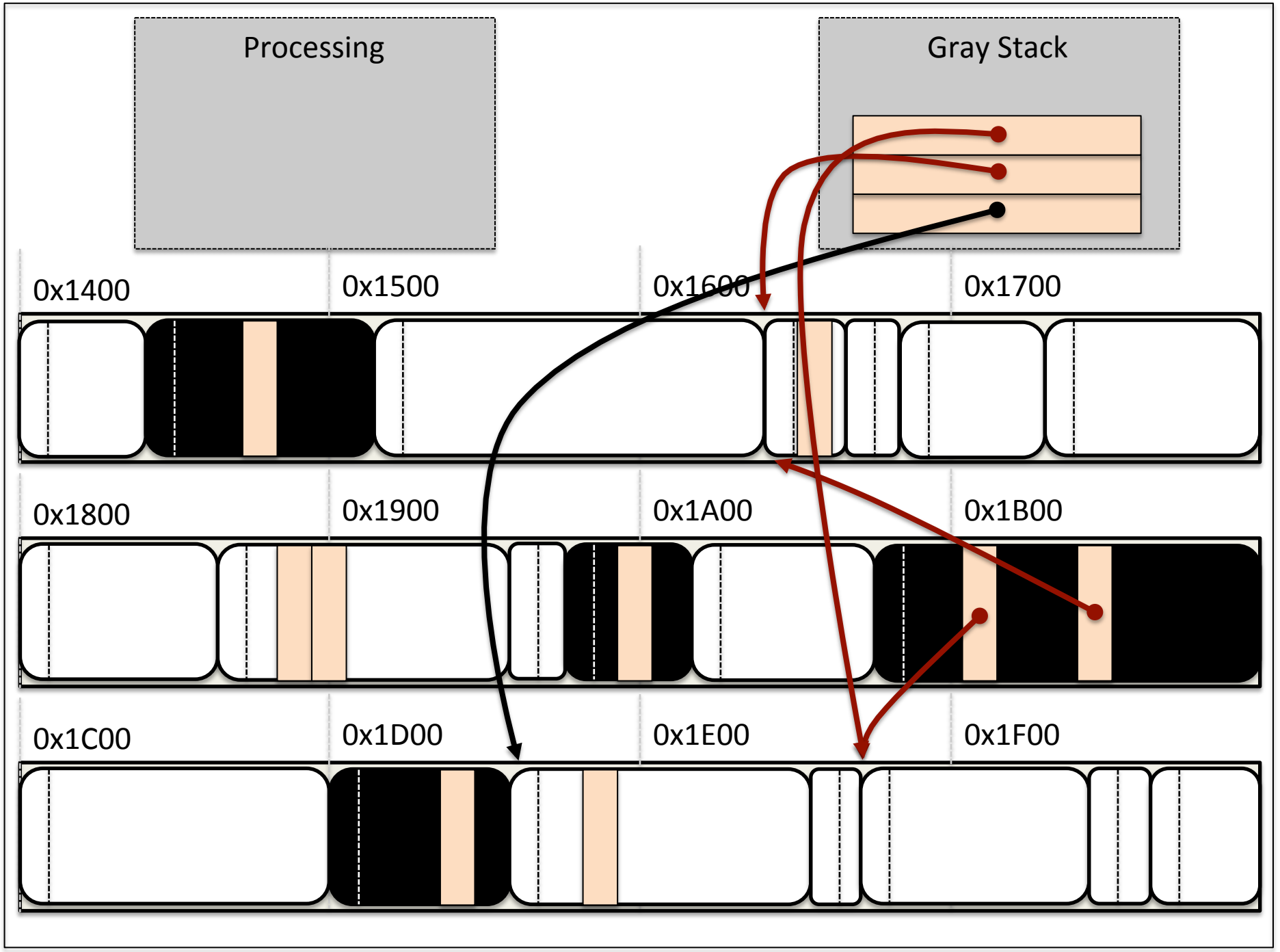

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

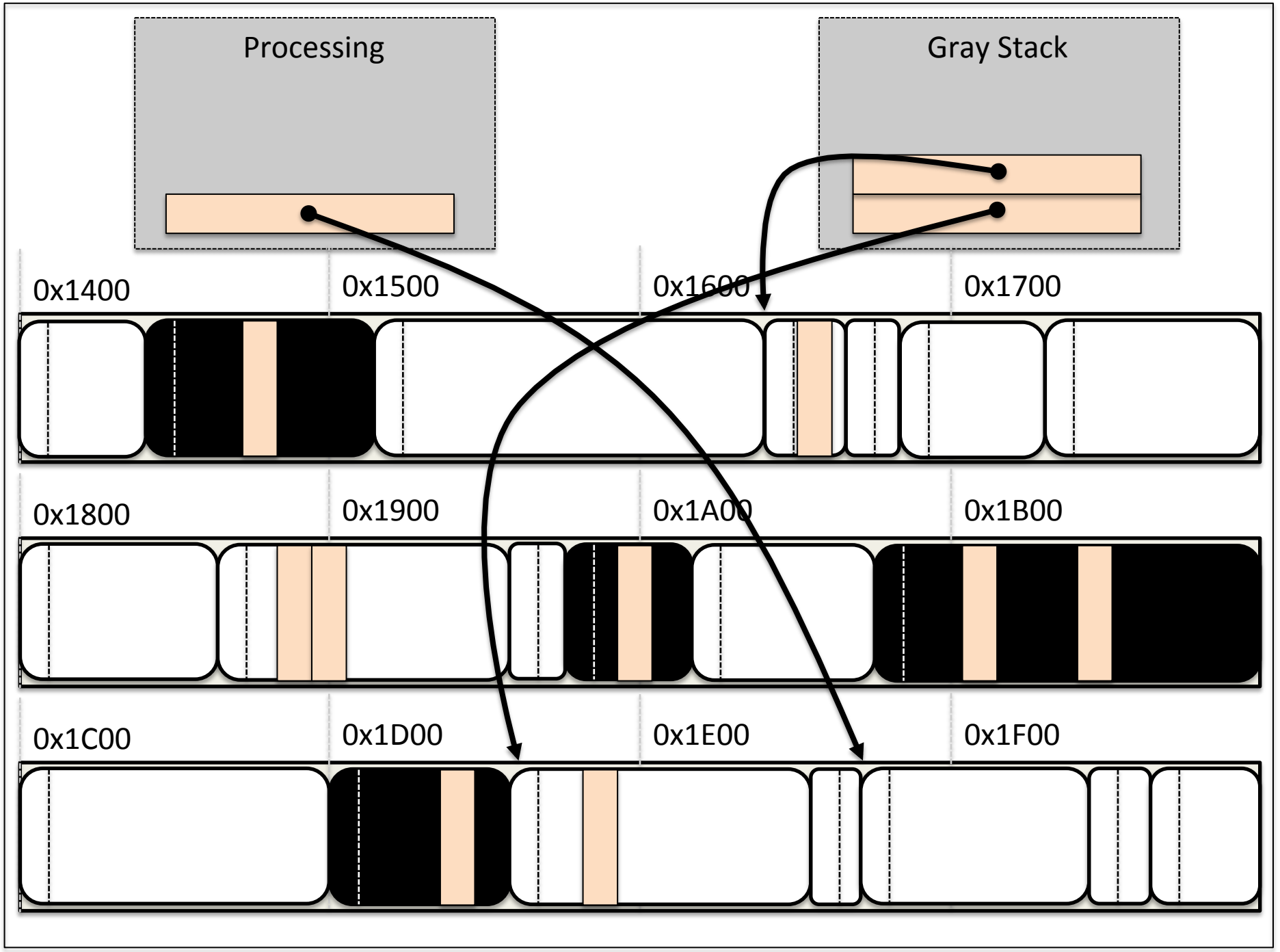

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

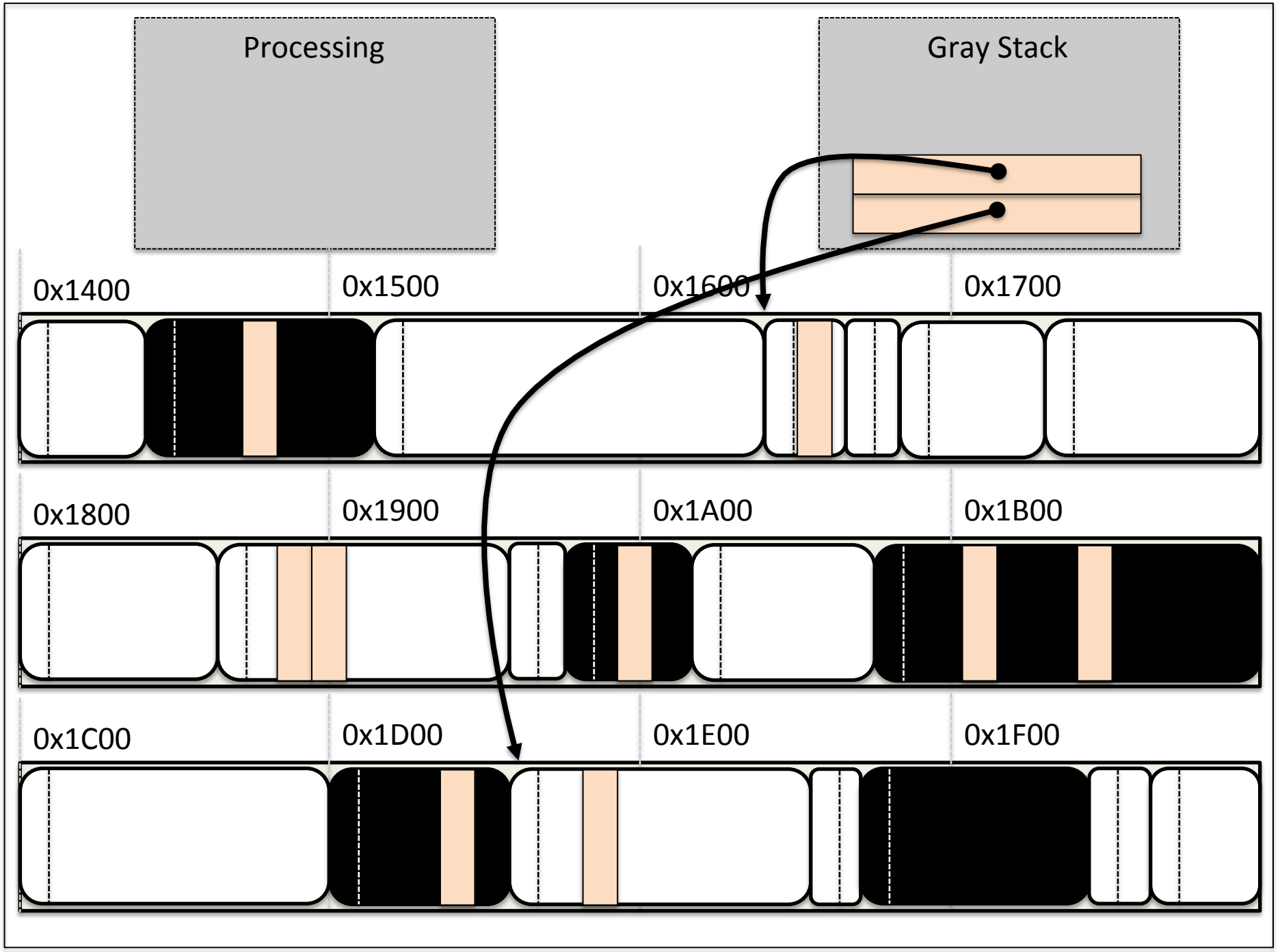

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

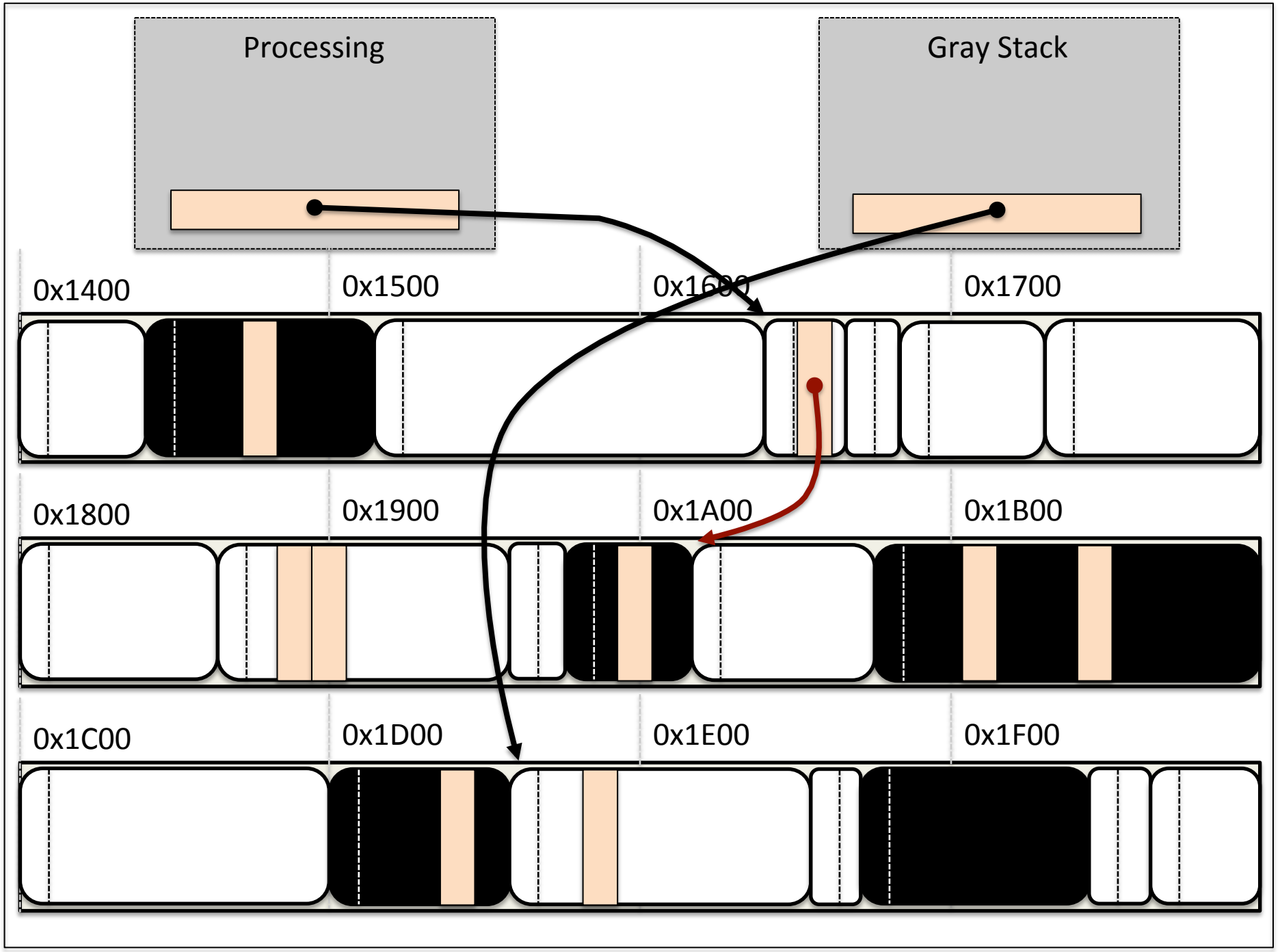

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

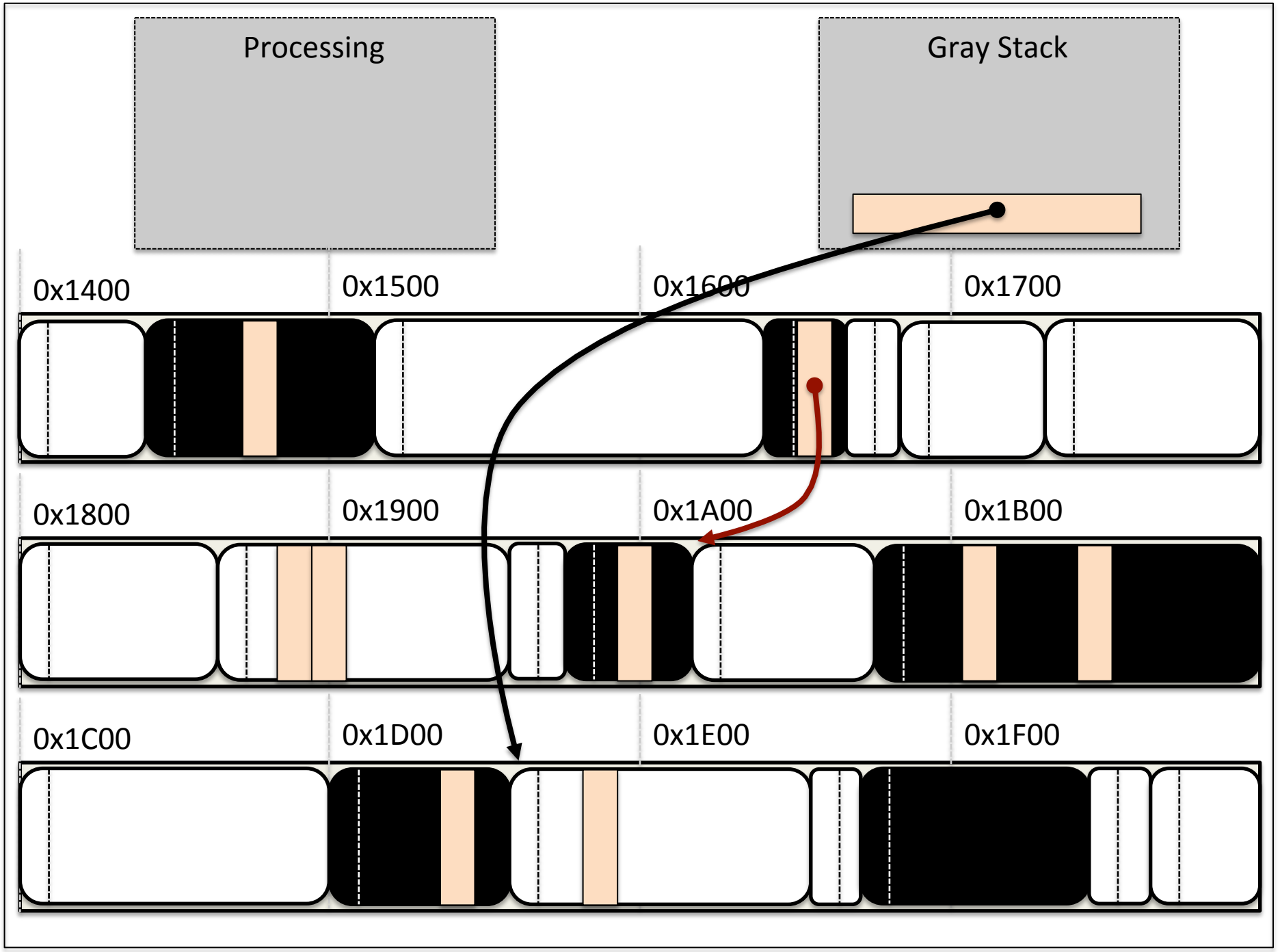

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

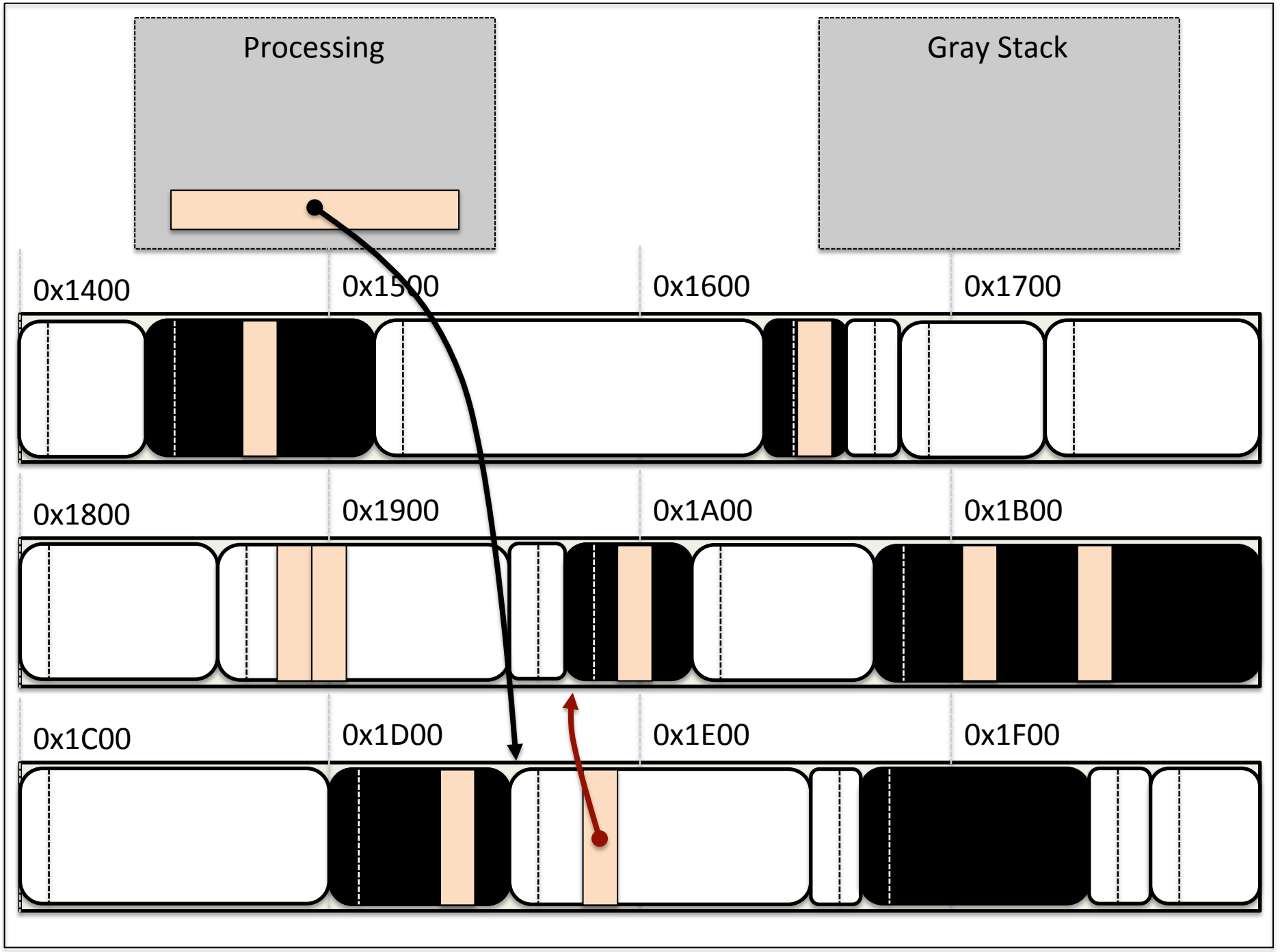

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

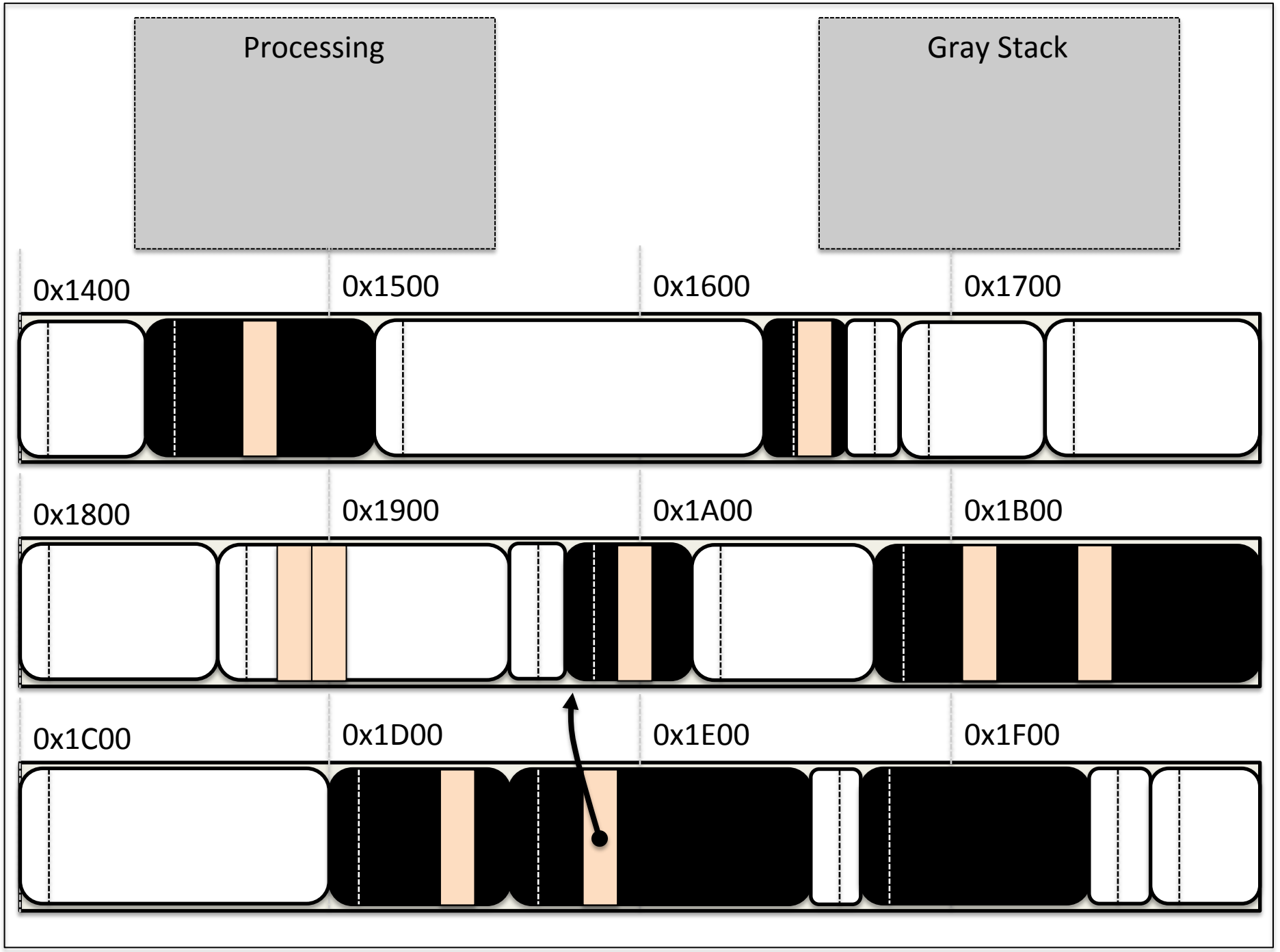

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

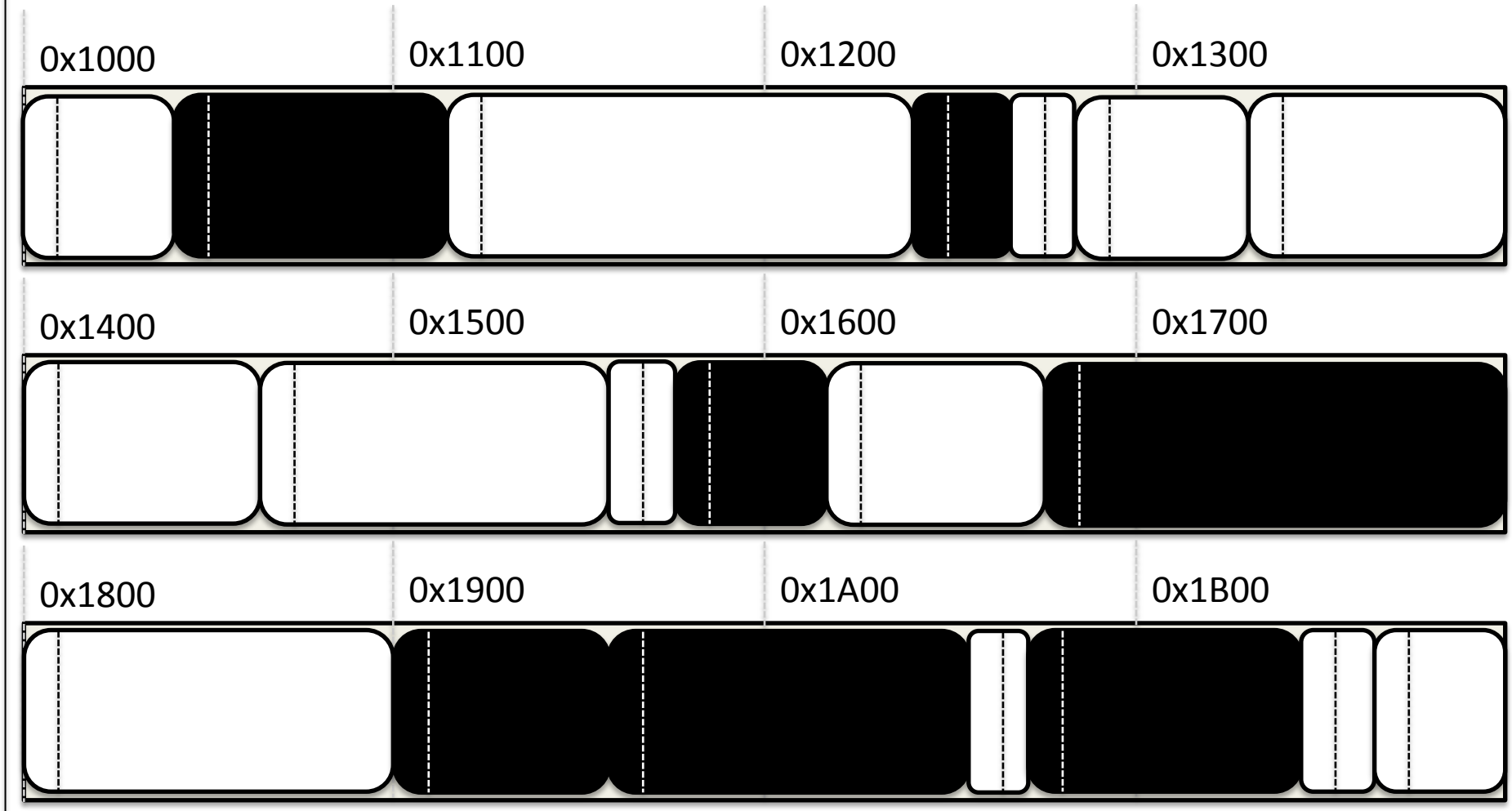

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

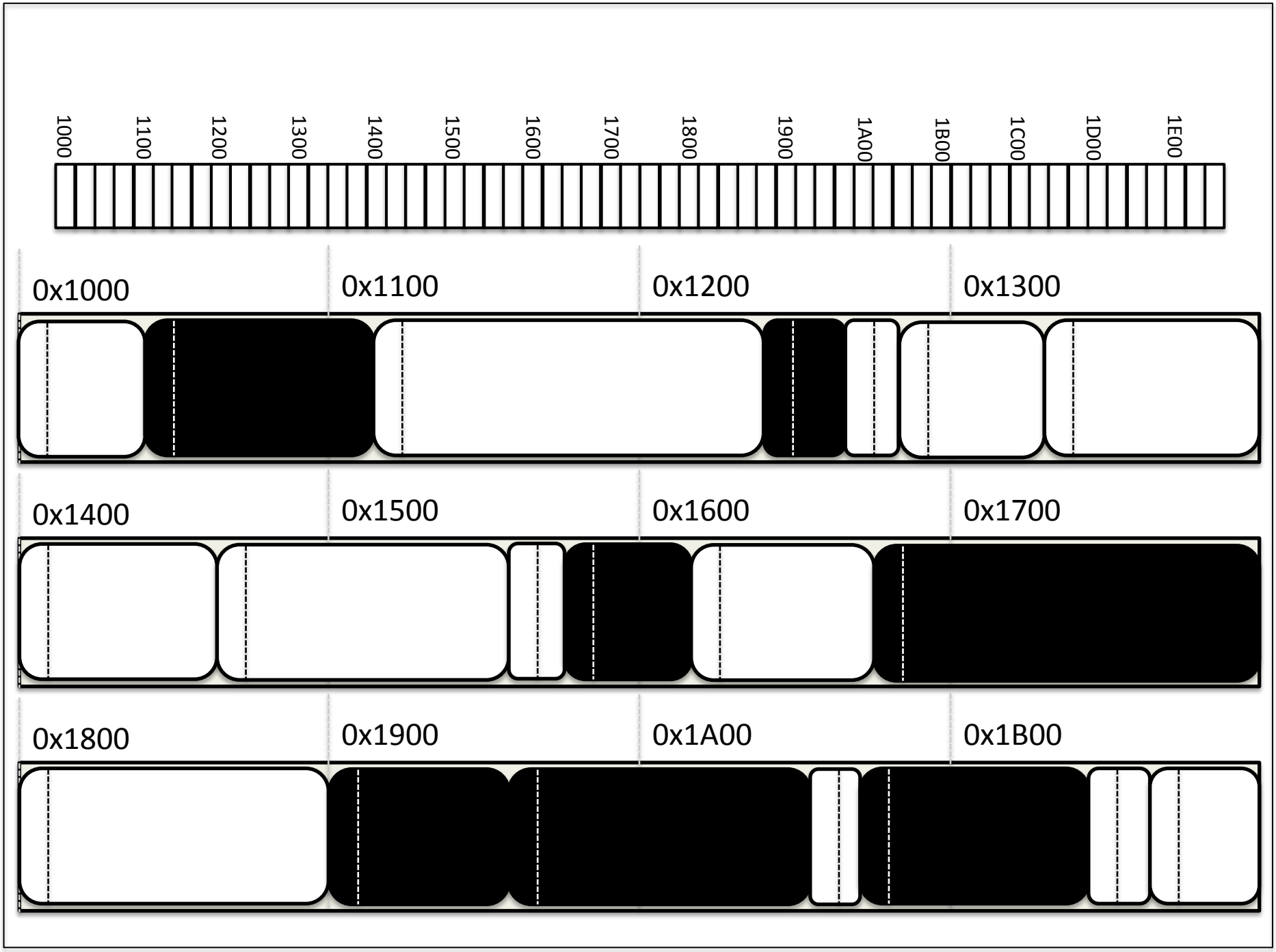

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

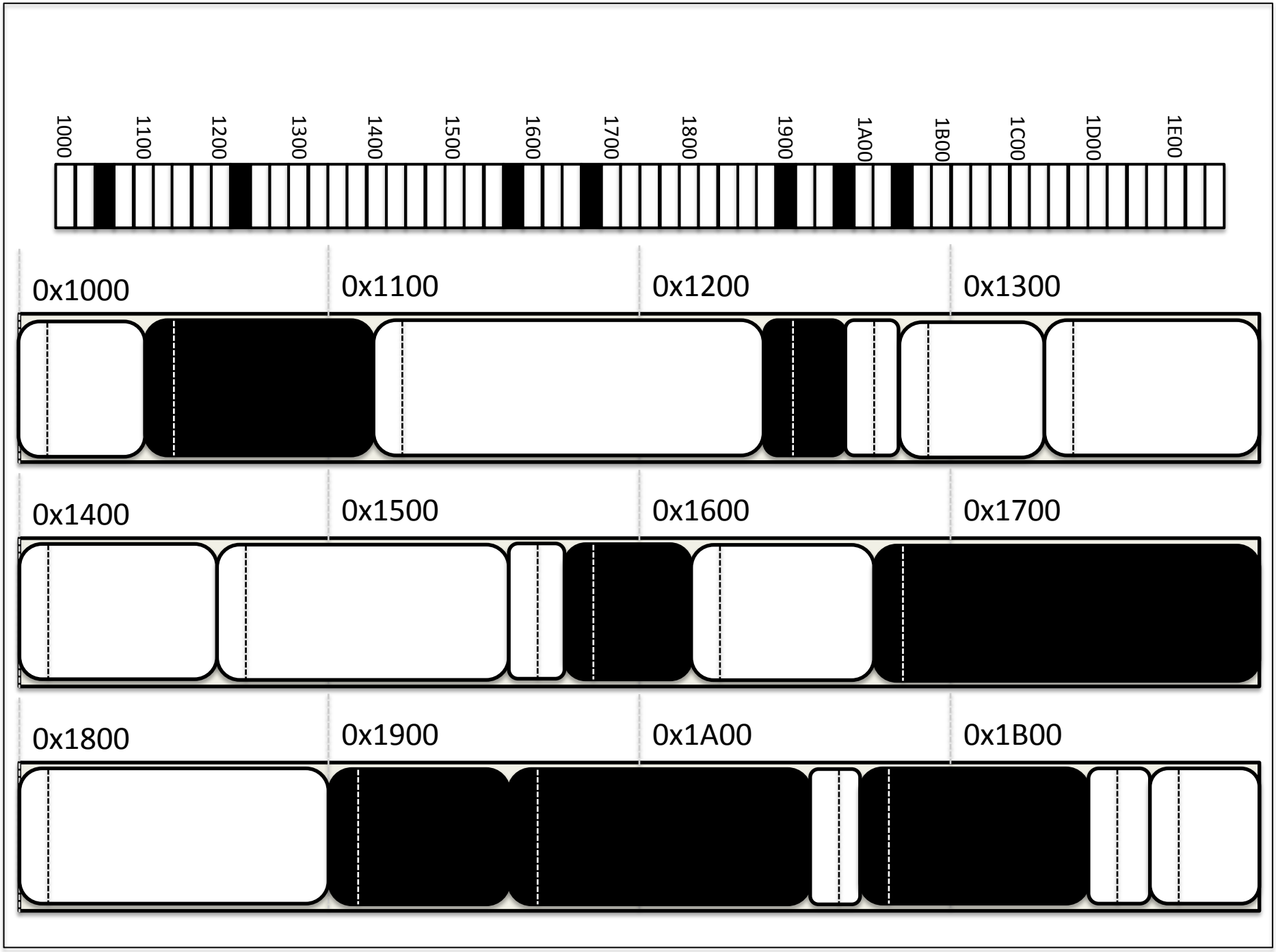

Processing
Gray Stack
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

Mark Bits

- Tracing gives us black and white objects
  - Need to record what objects are black
- Requires at least one bit per object
  - Some algorithms may want more information
- We can use a bit in the object header
  - Recall the bit stealing technique
  - No overflow problem as in reference counting
- Can also store information on the side


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00


1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
1A00
1B00
1C00
1D00
1E00
0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00

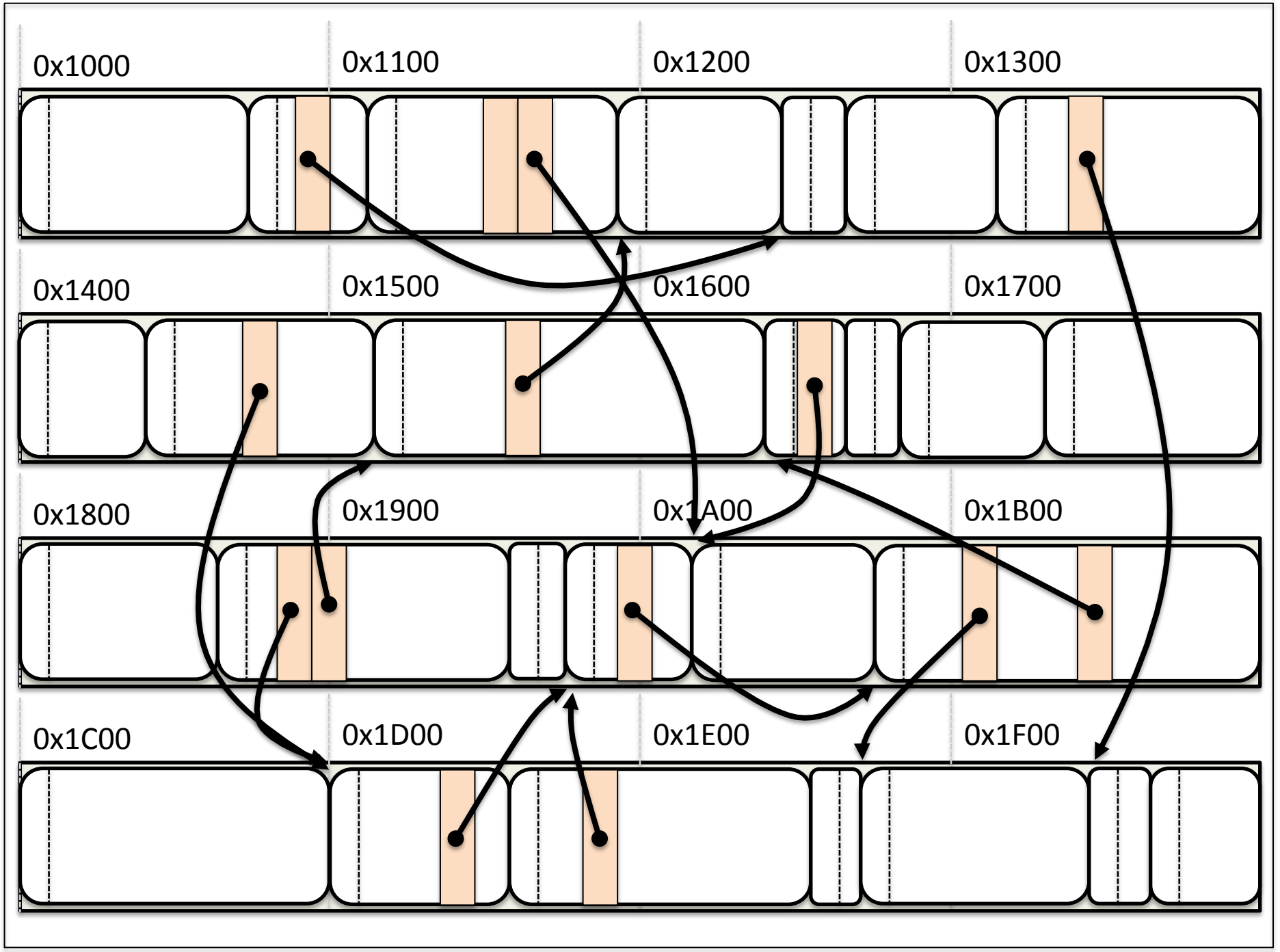

1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
1A00
1B00
1C00
1D00
1E00
0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00

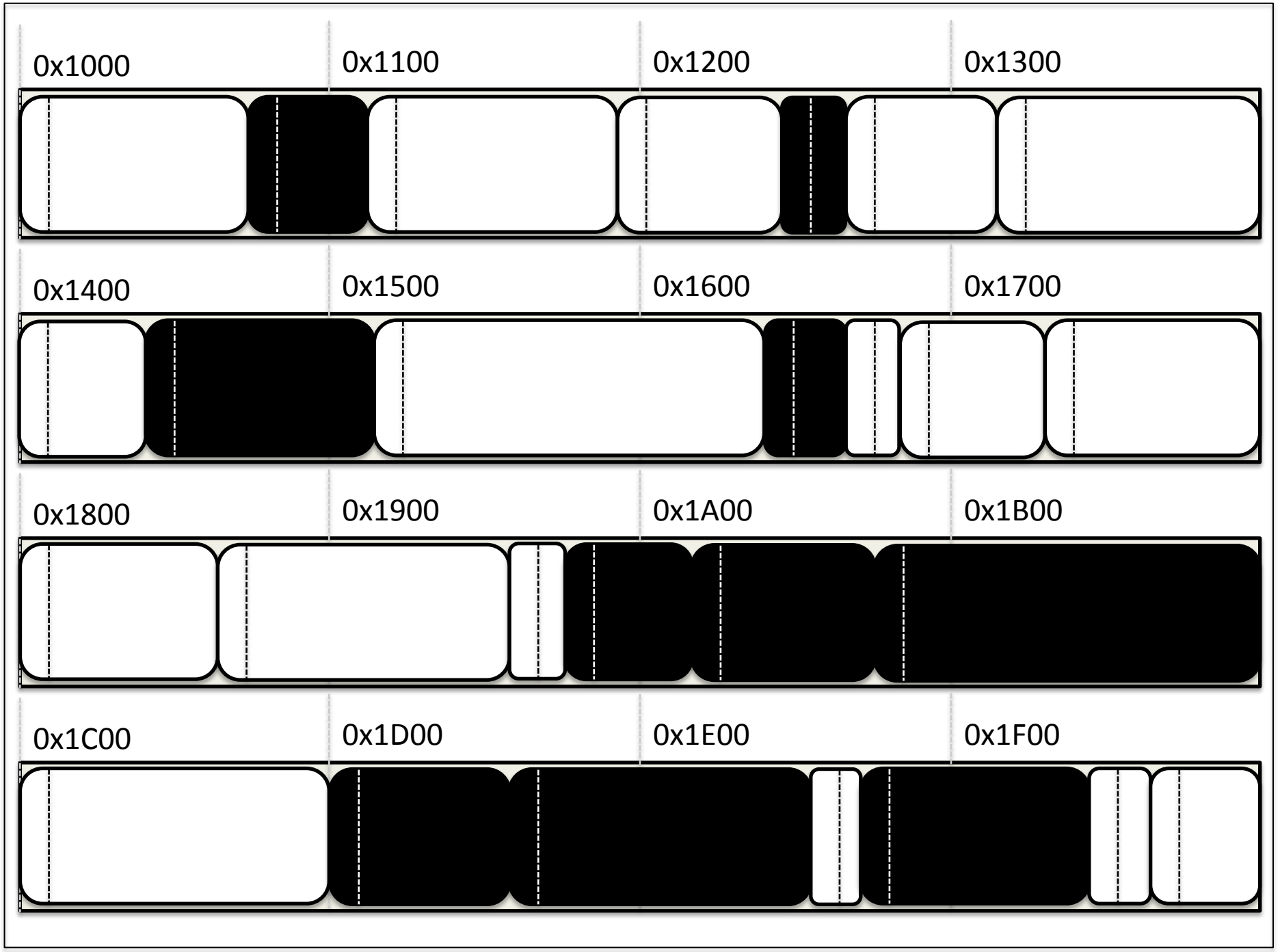
Mark Bitmaps

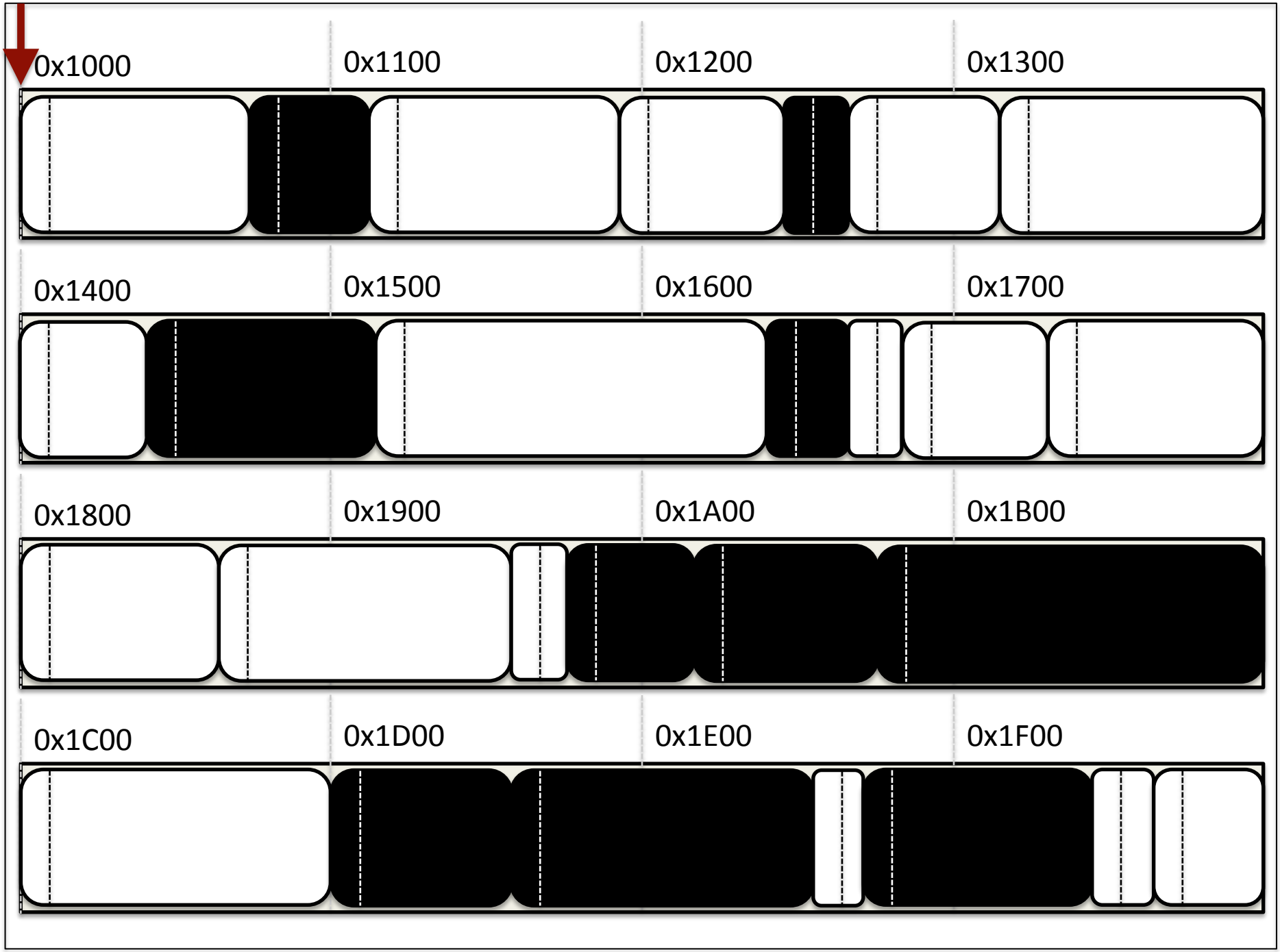
- Alternative to storing mark bits in headers
  - Metadata stored separate from data
  - One bit per fixed size memory range
- Some big advantages
  - Doesn't require masking of stolen bits
  - Good cache behavior
- Some problems
  - May be an extra load per read
  - Non-trivial memory overhead

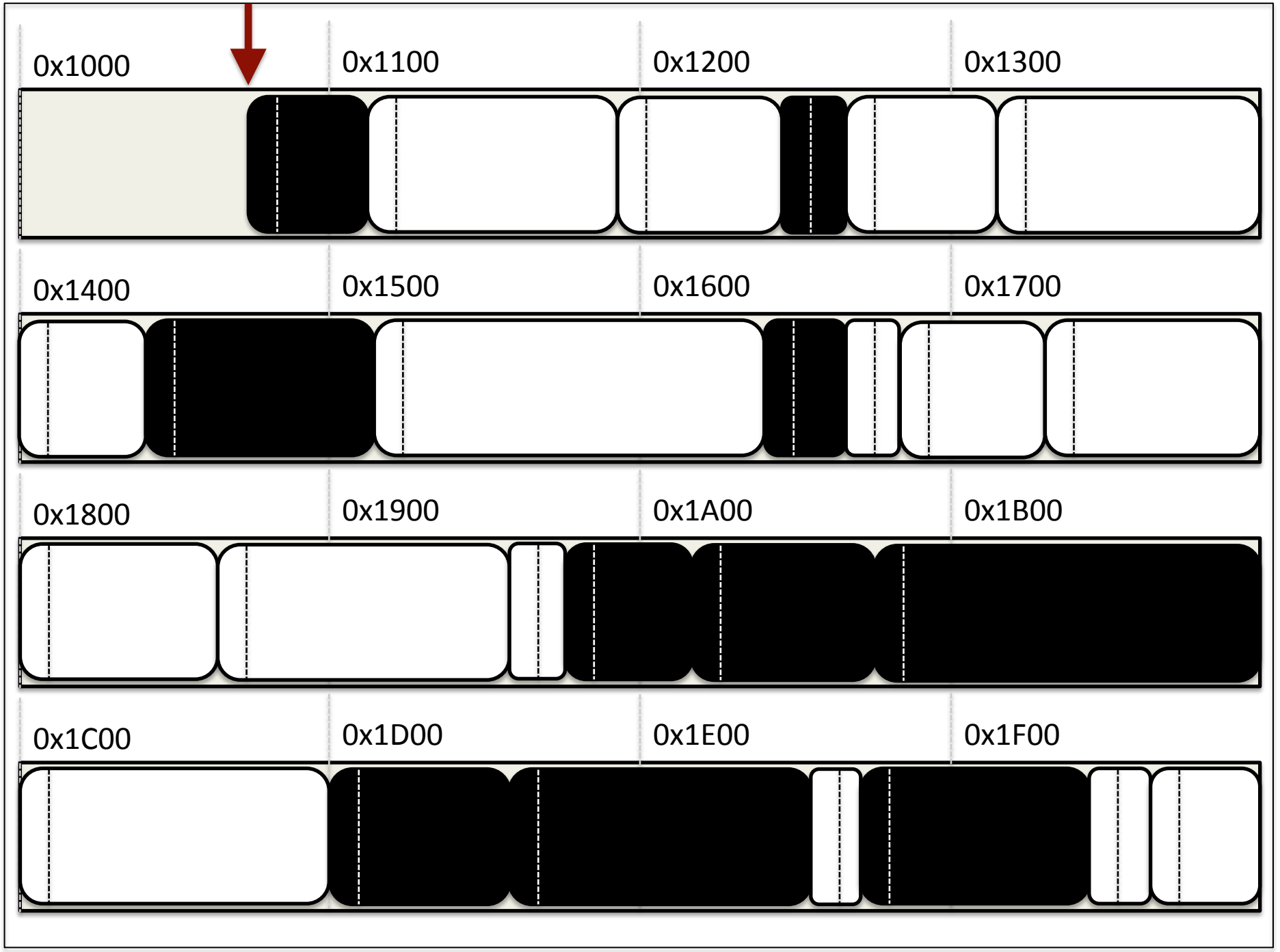
Mark and Sweep Collectors

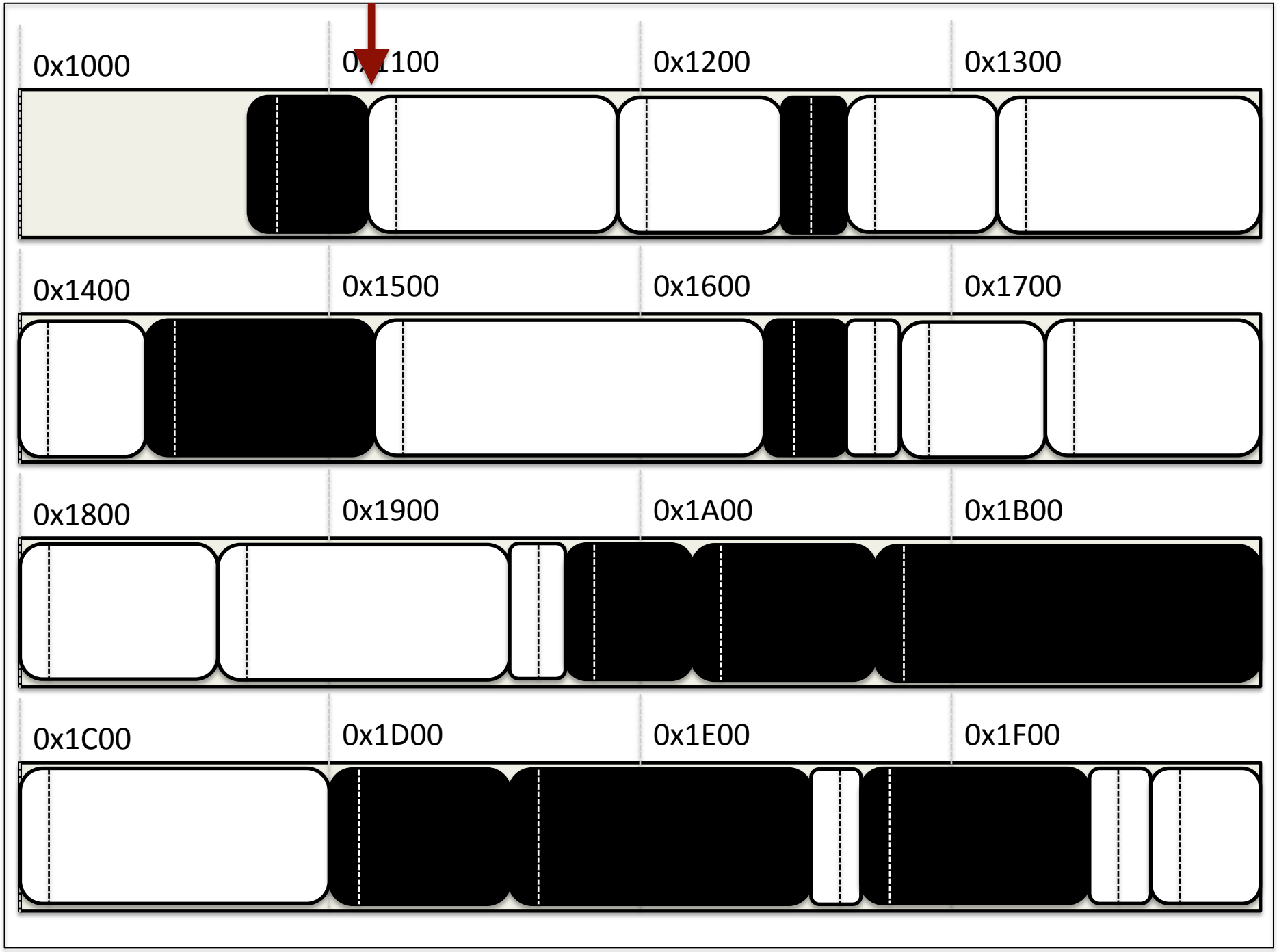
- Two-pass garbage collection algorithm
- Perform a trace across the heap
  - Mark all live objects
  - Everything else is garbage
- Follow up by sweeping over the heap
  - Free any unmarked objects

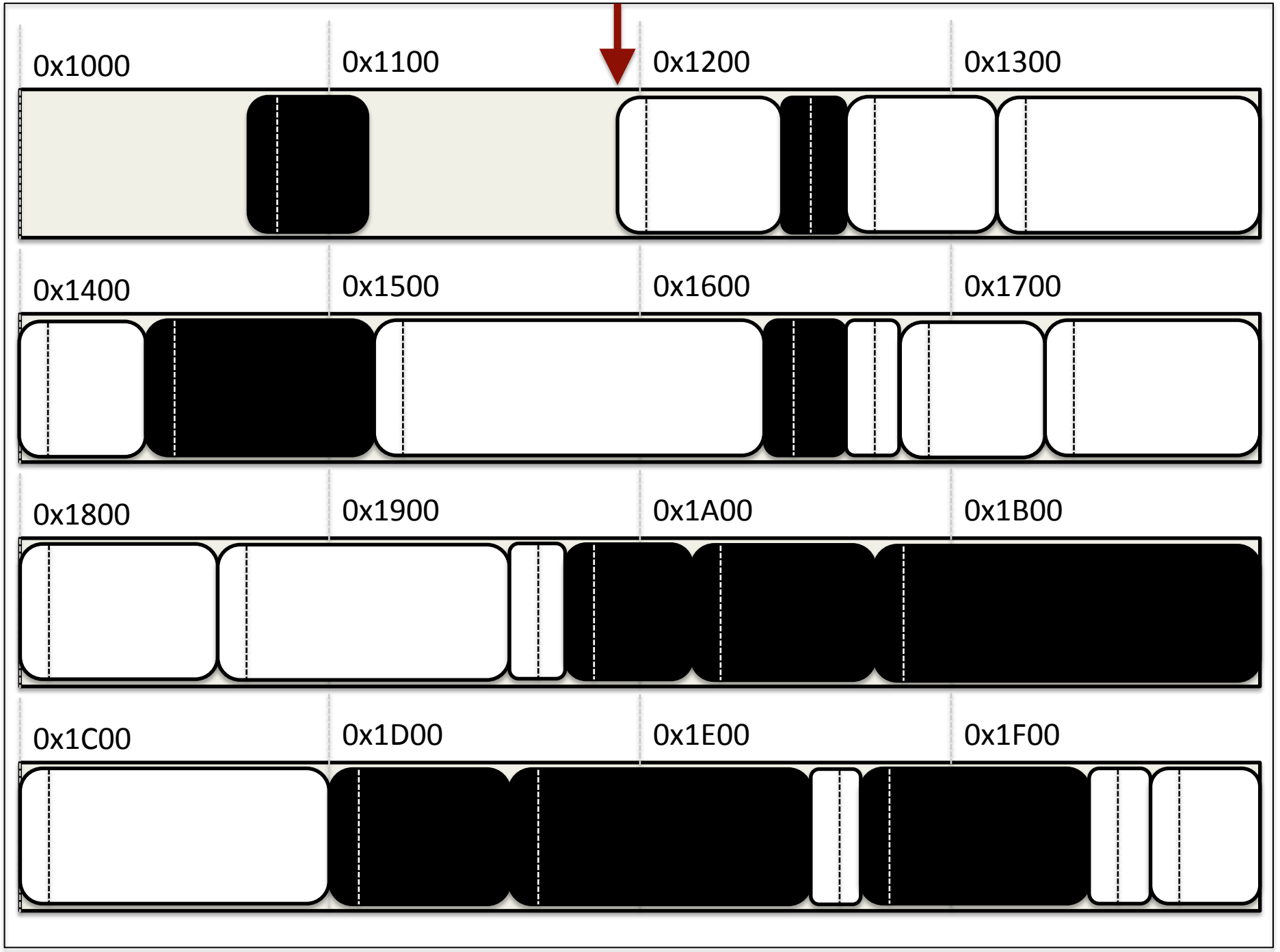

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

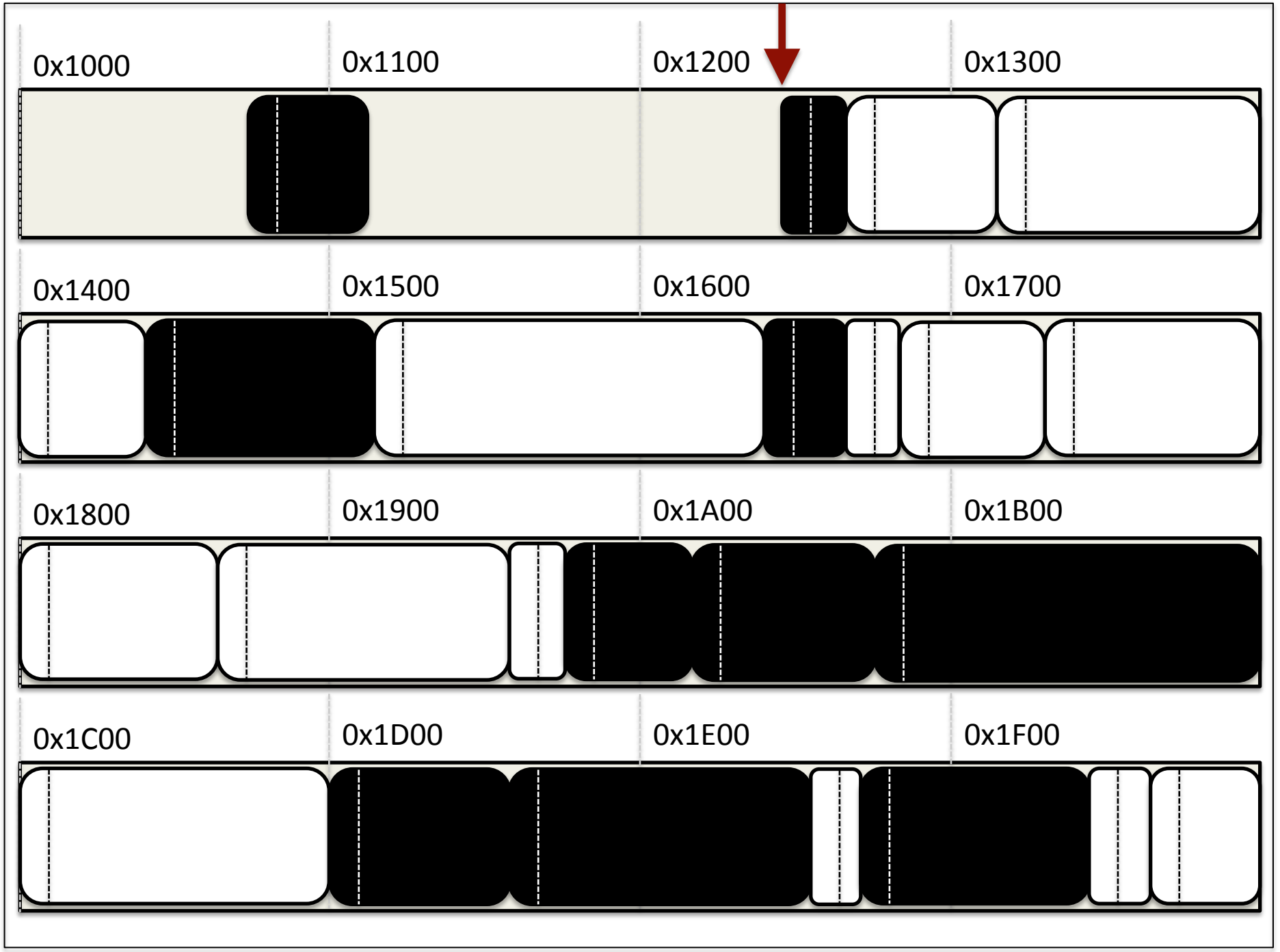

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

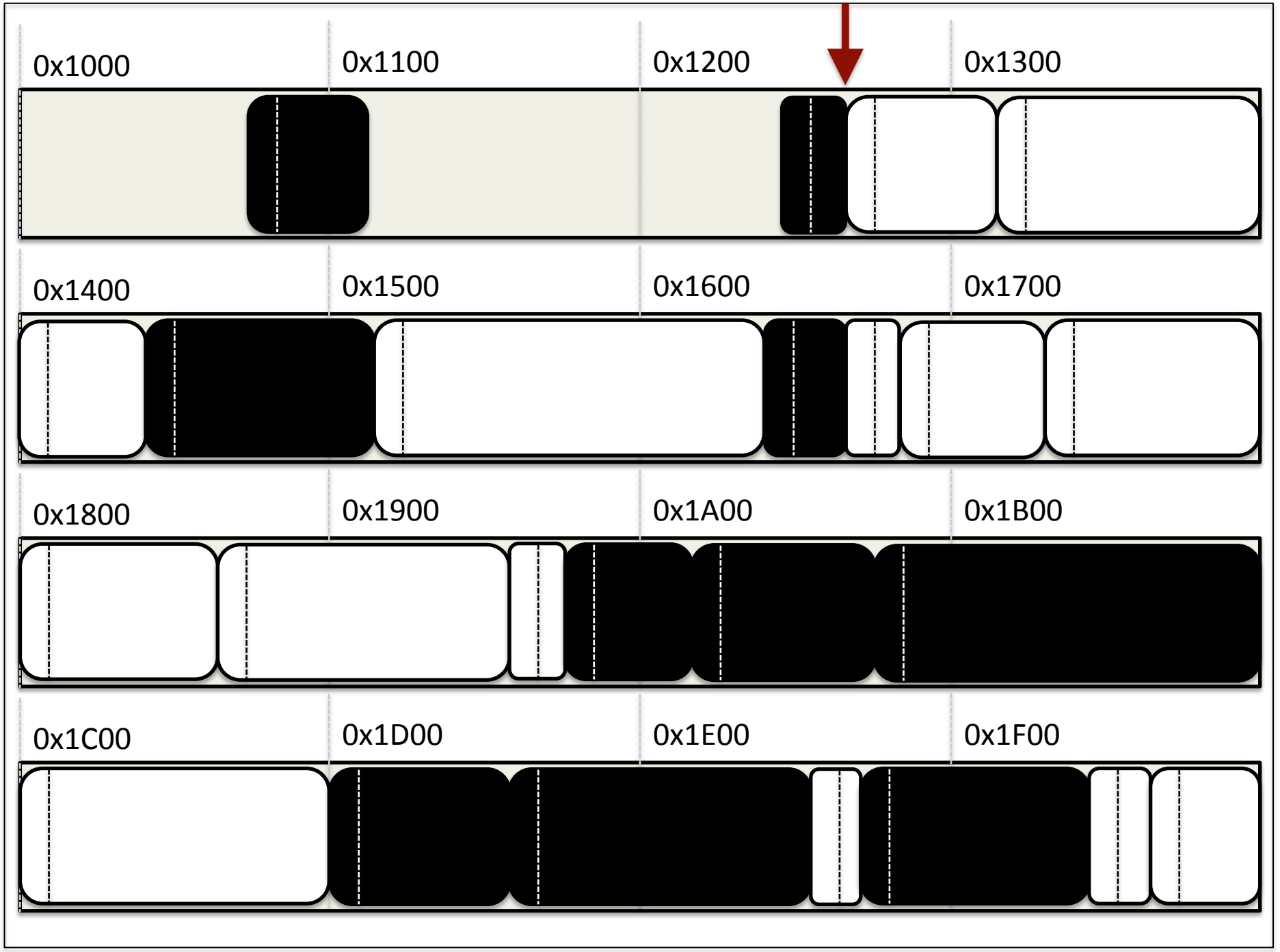

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

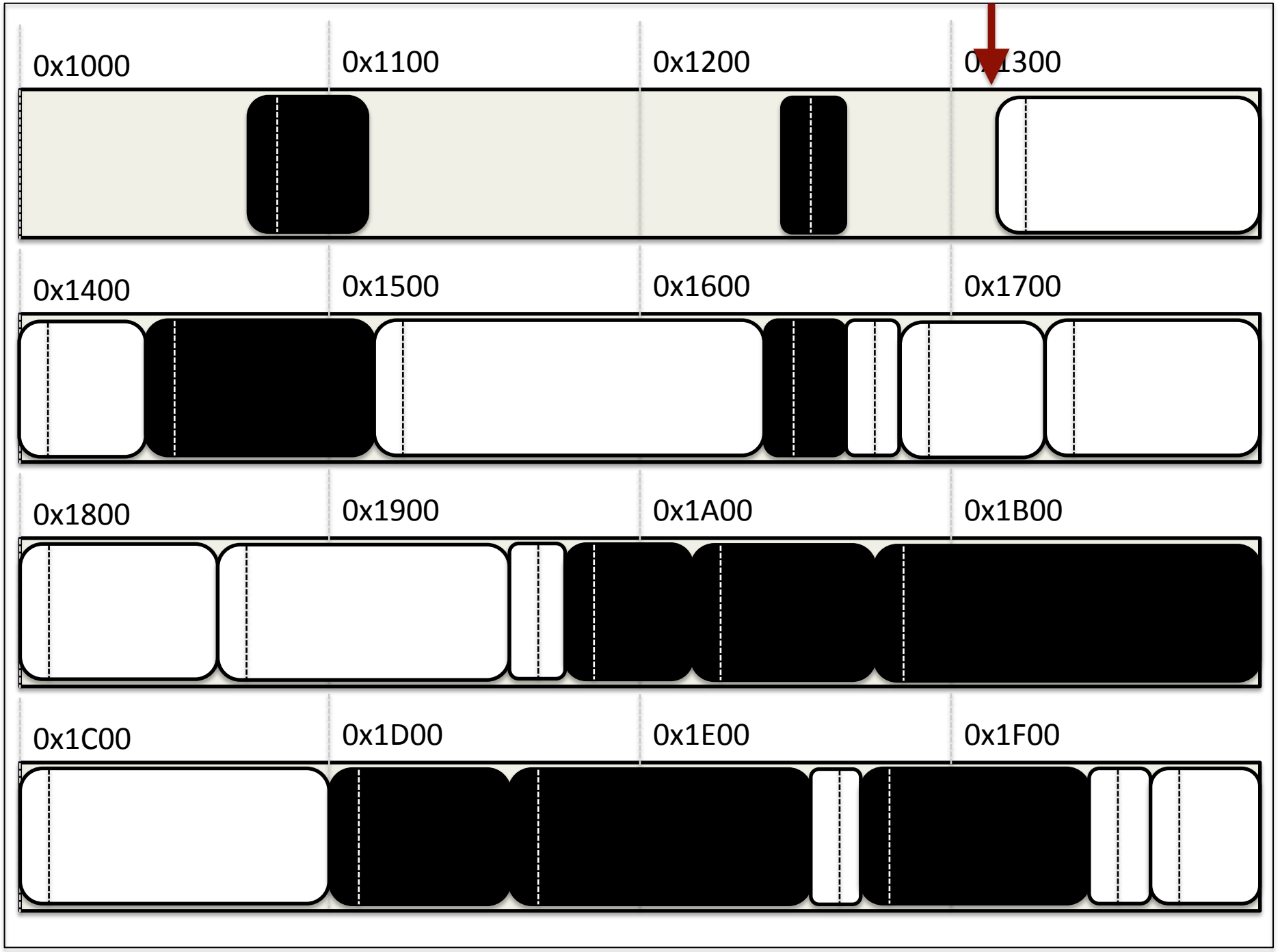

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

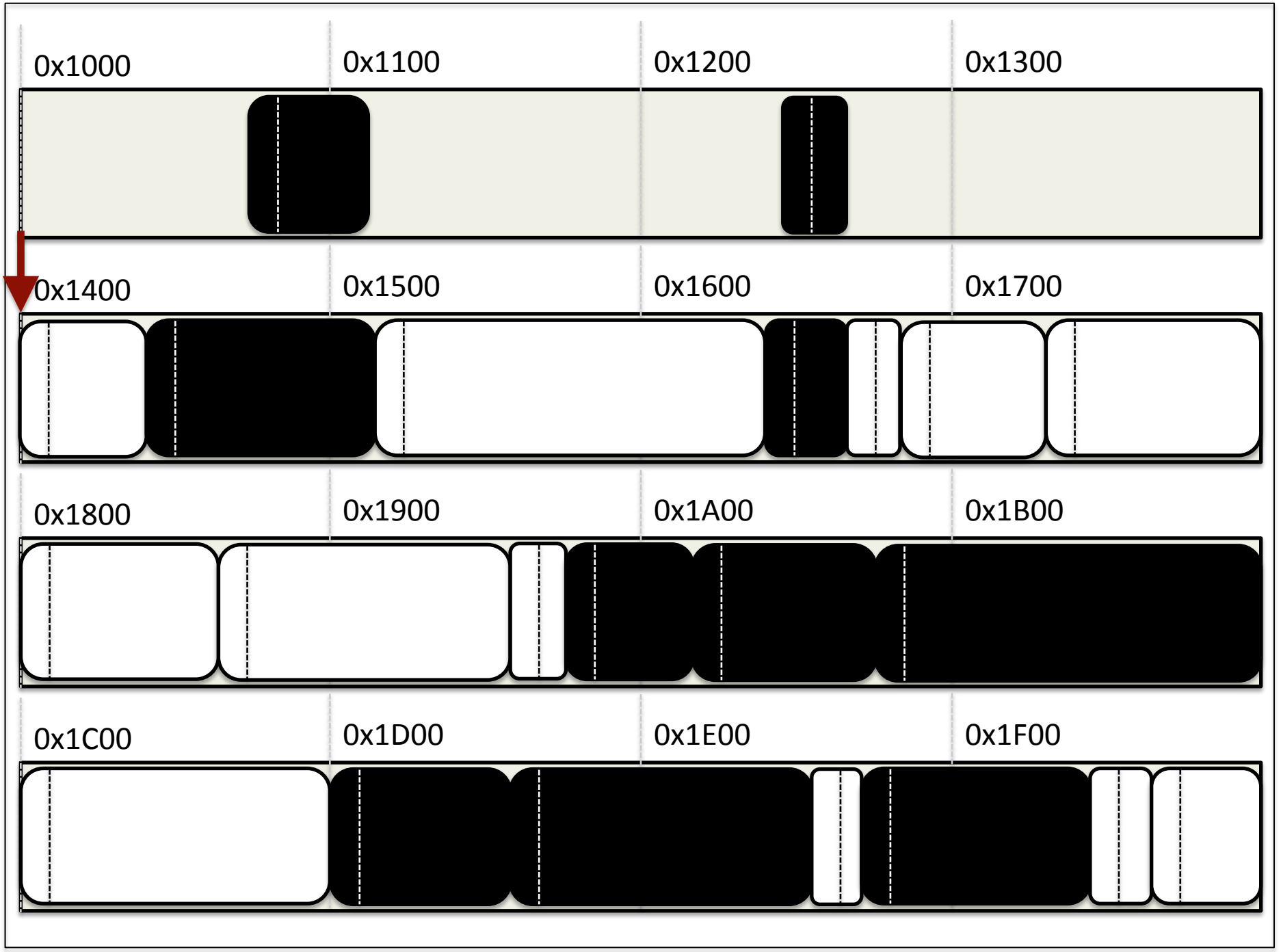

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

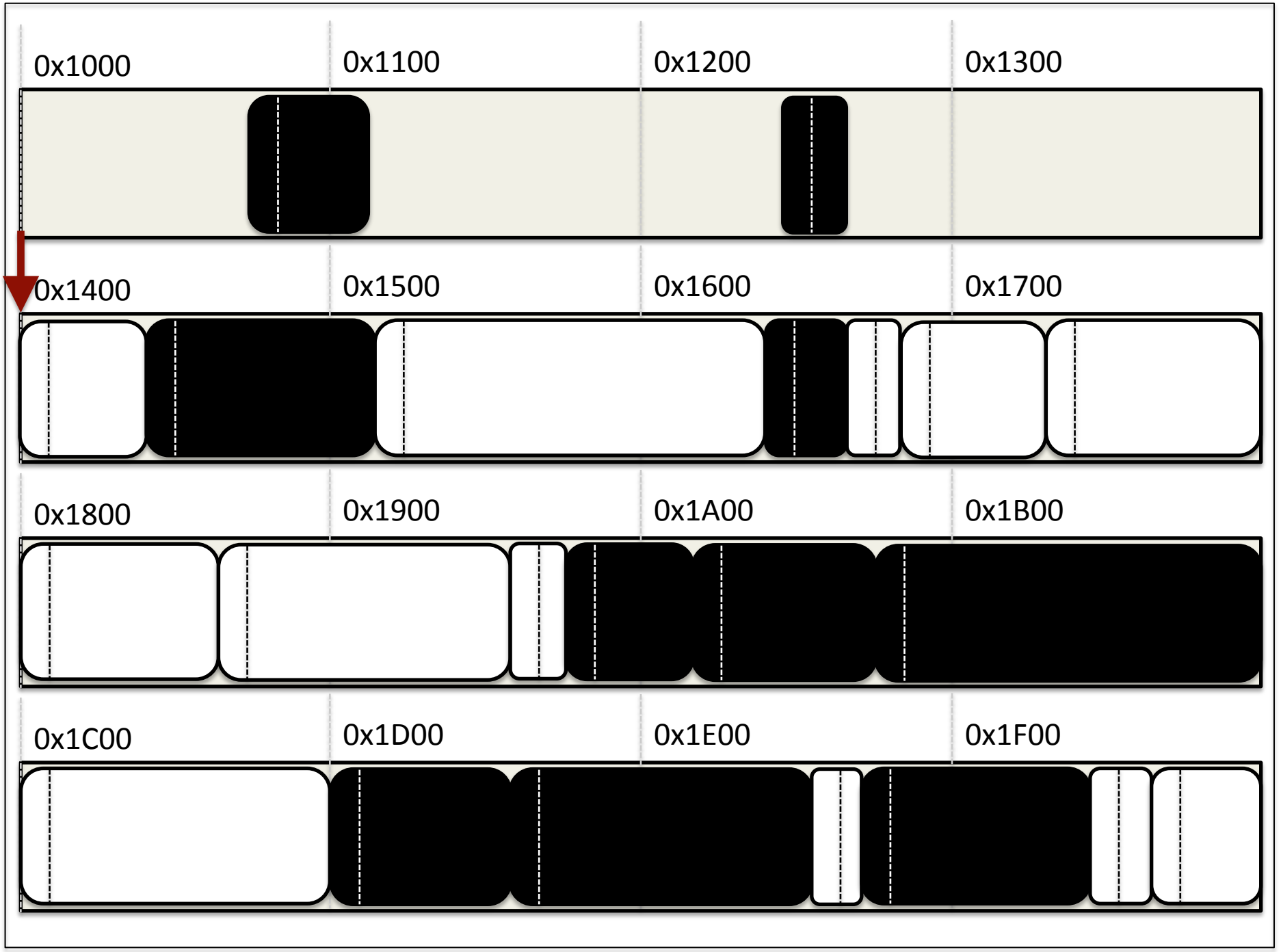

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

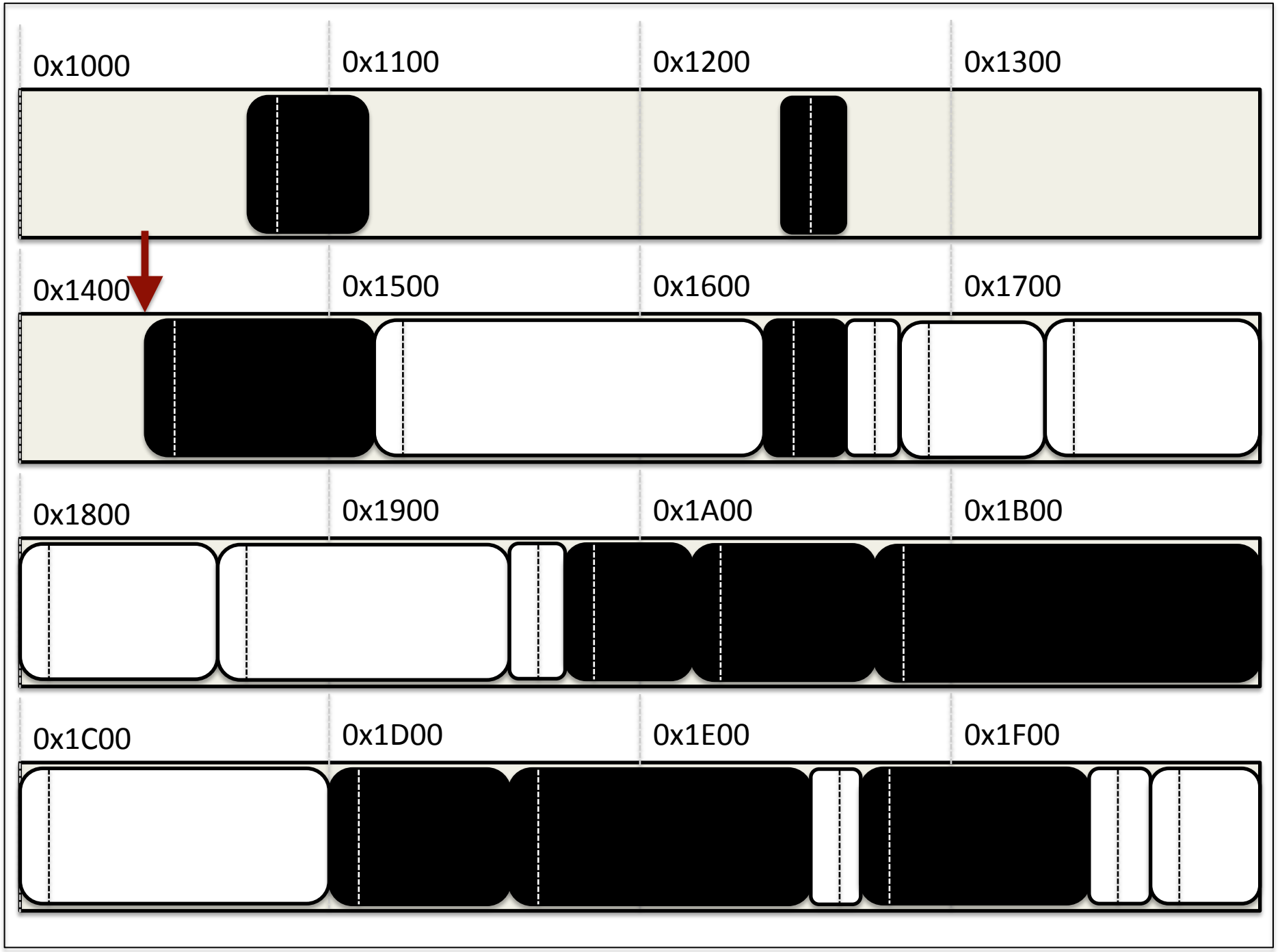

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

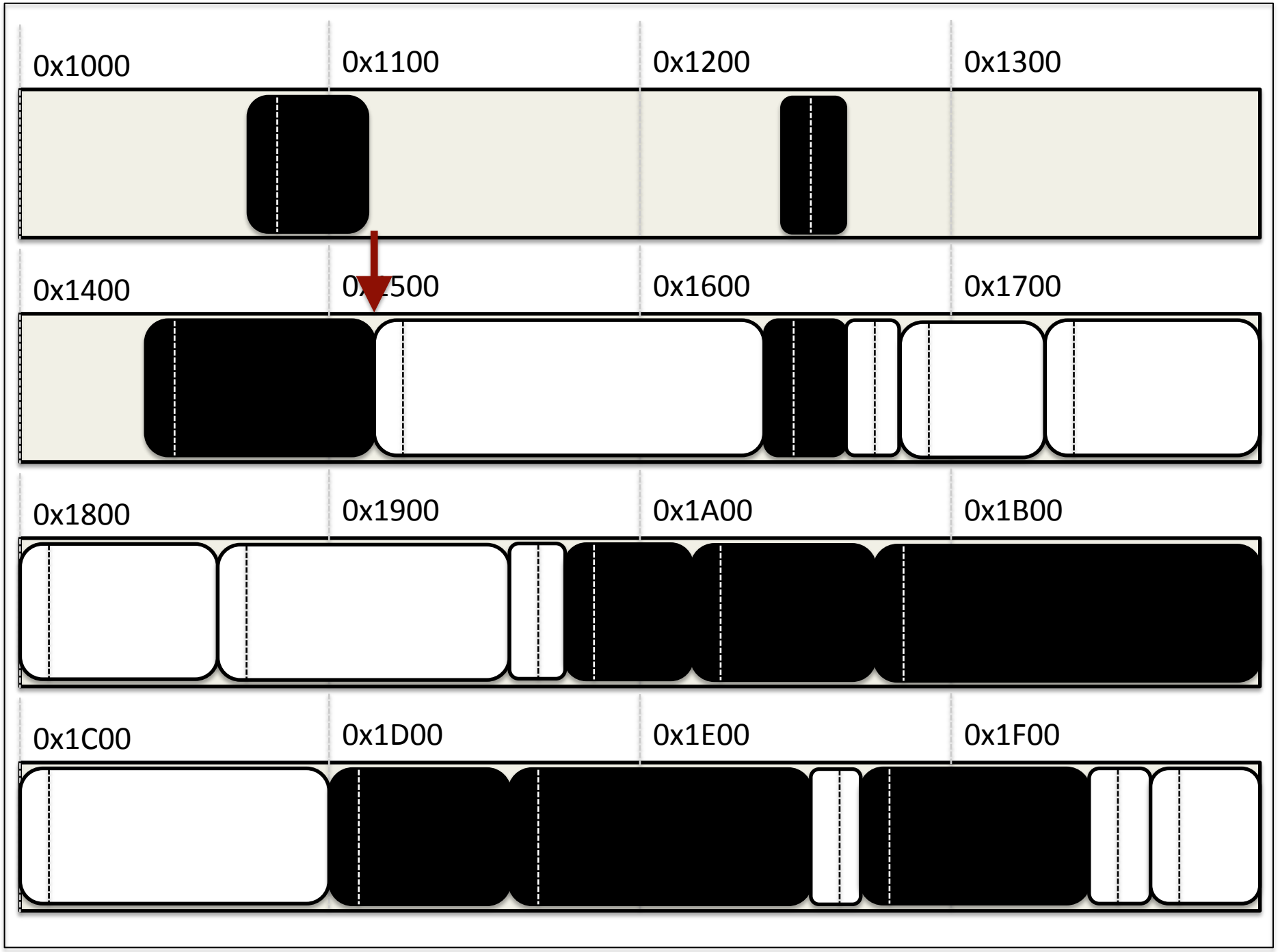

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

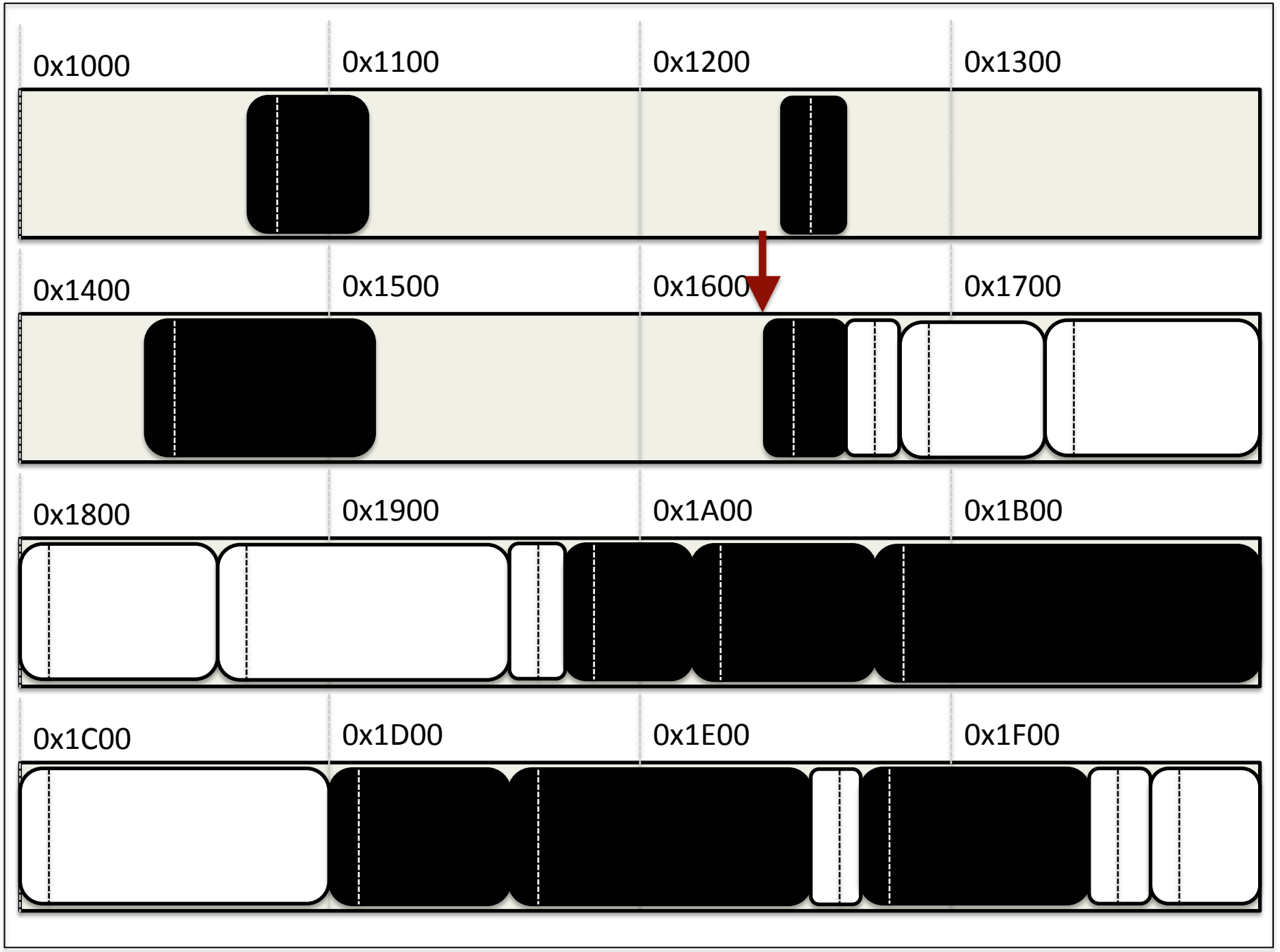
Sweep Phase

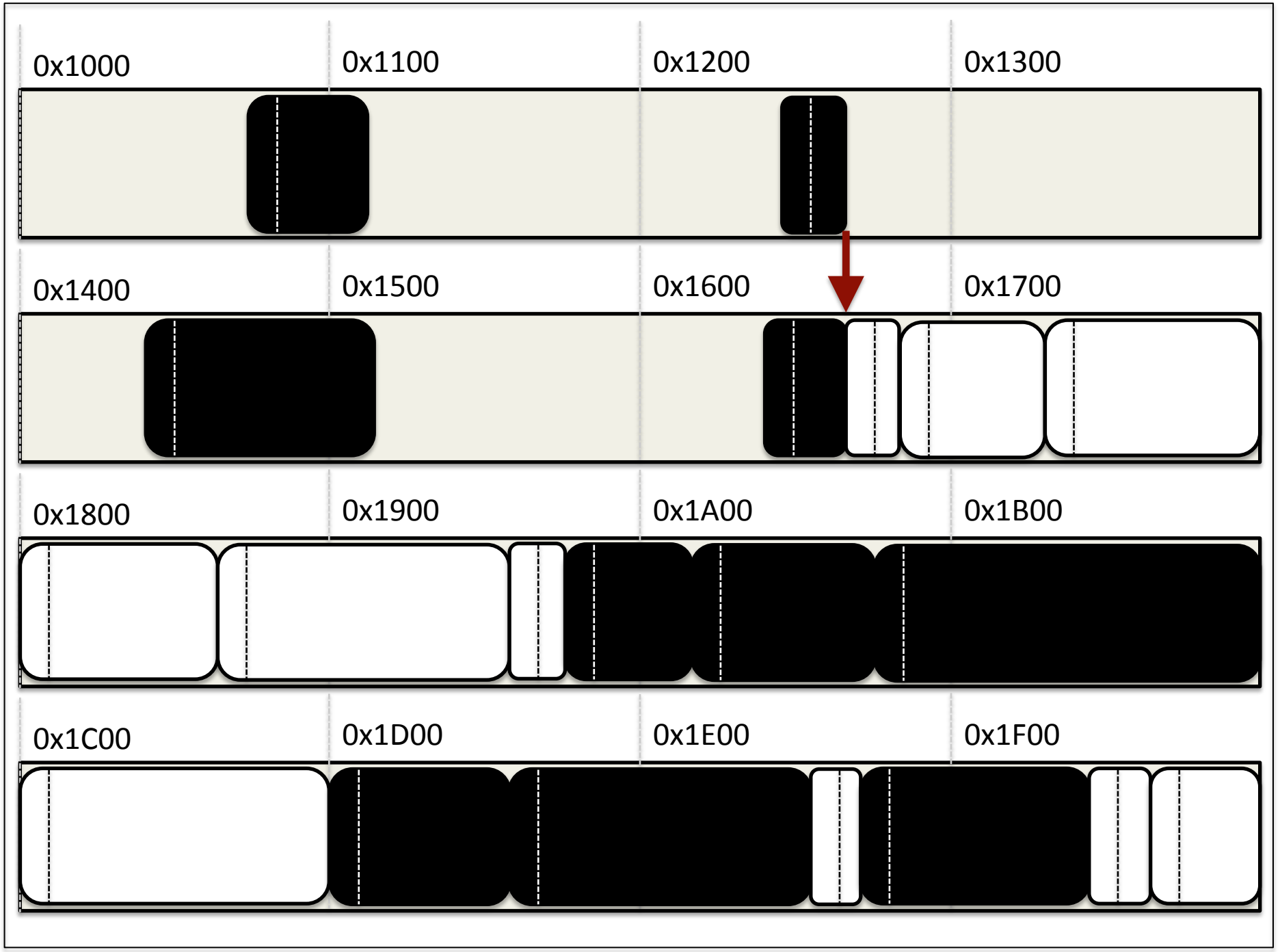
- Sweeping is a linear scan over the heap
  - Start at the first address
  - Calculate the offset of the next object
- Need to be able to scan over gaps

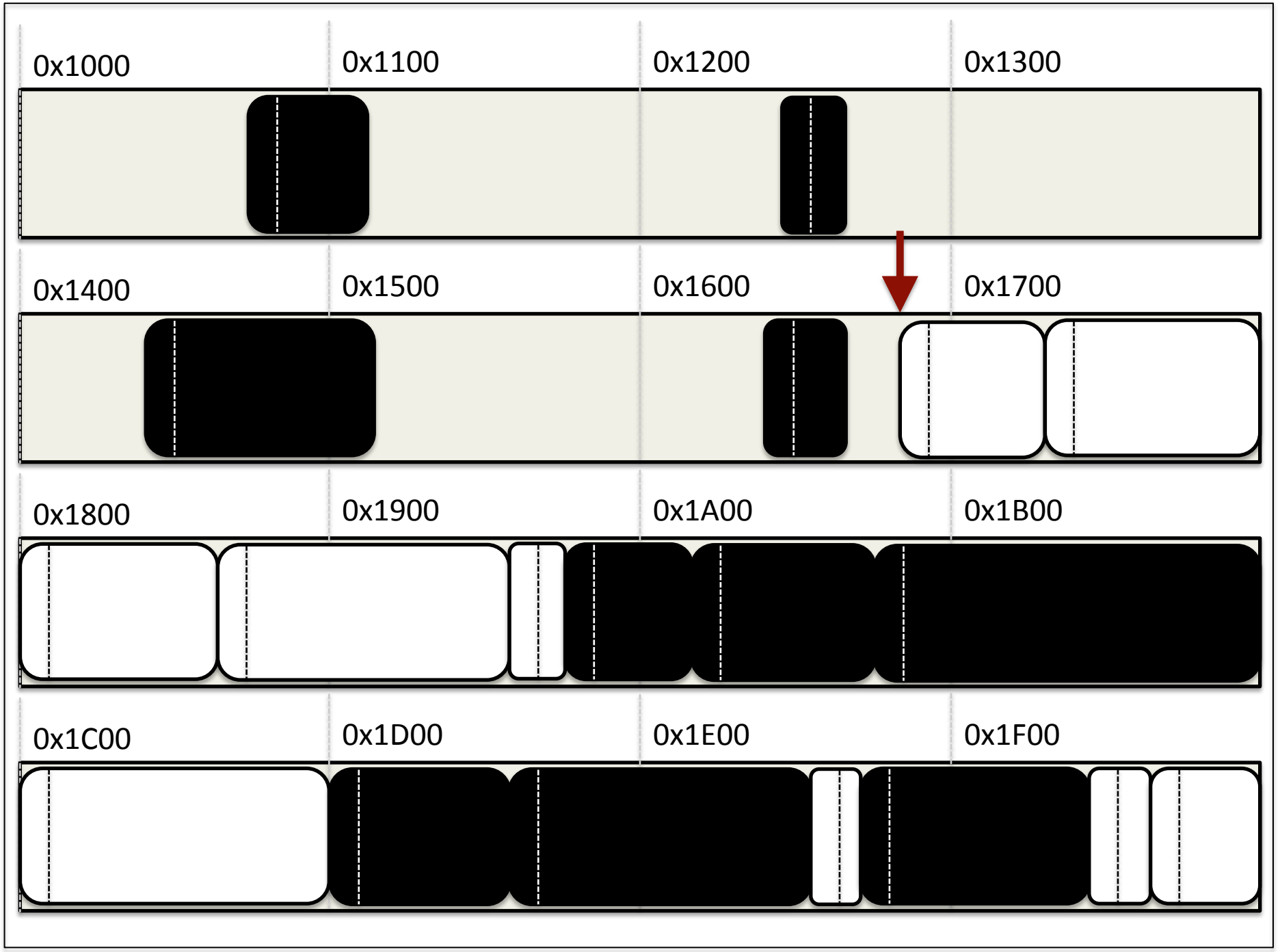

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

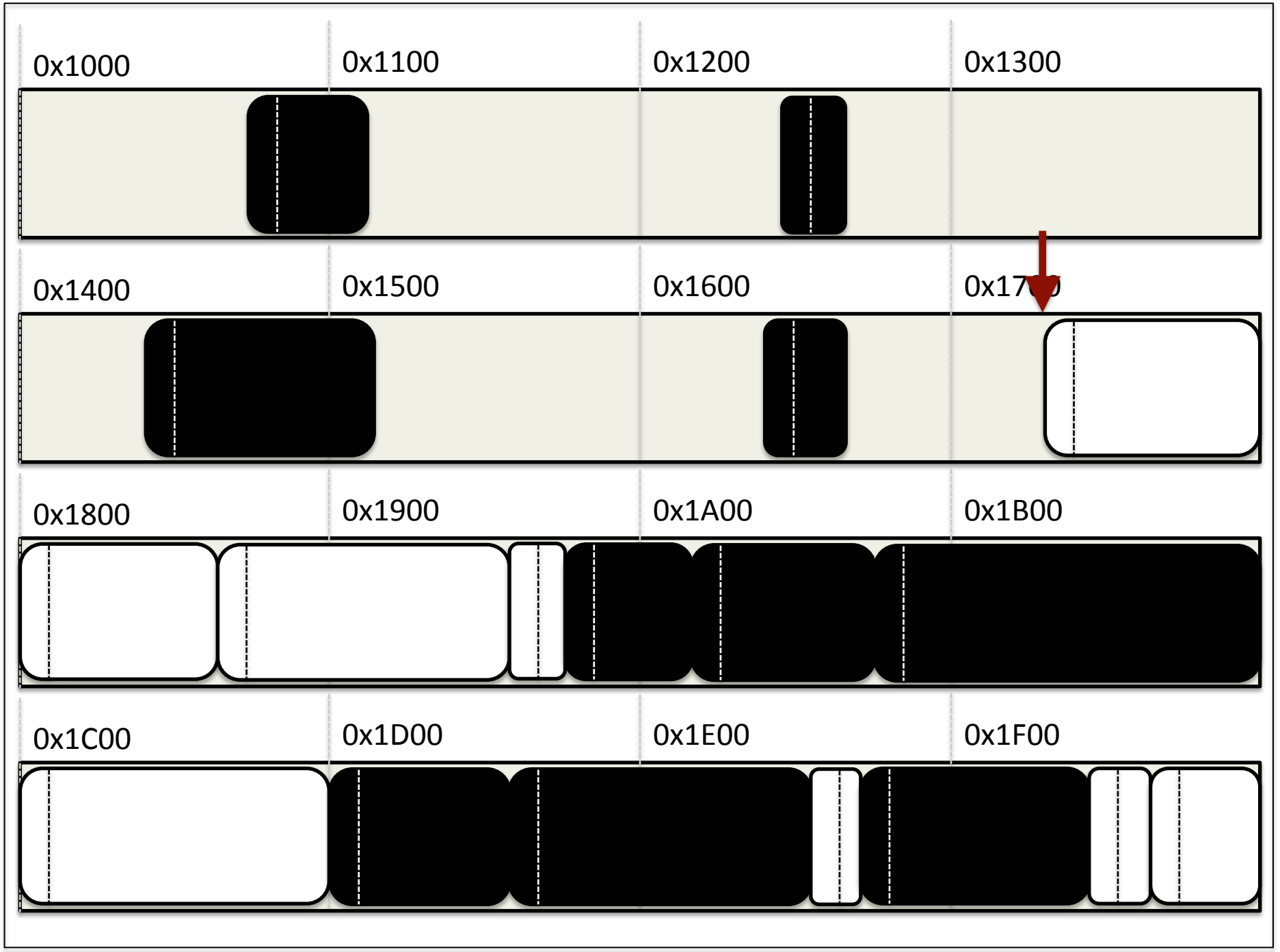

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

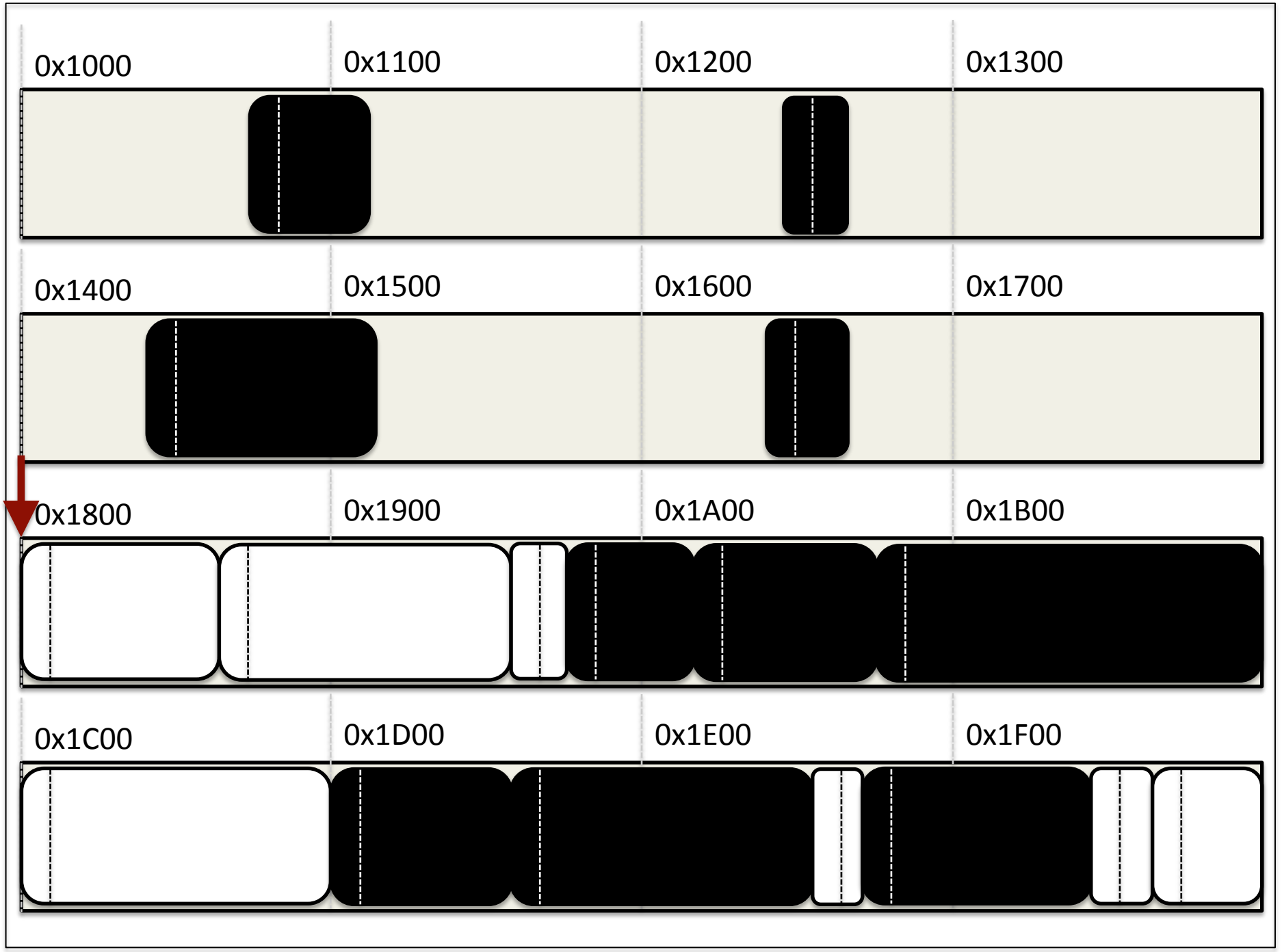

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

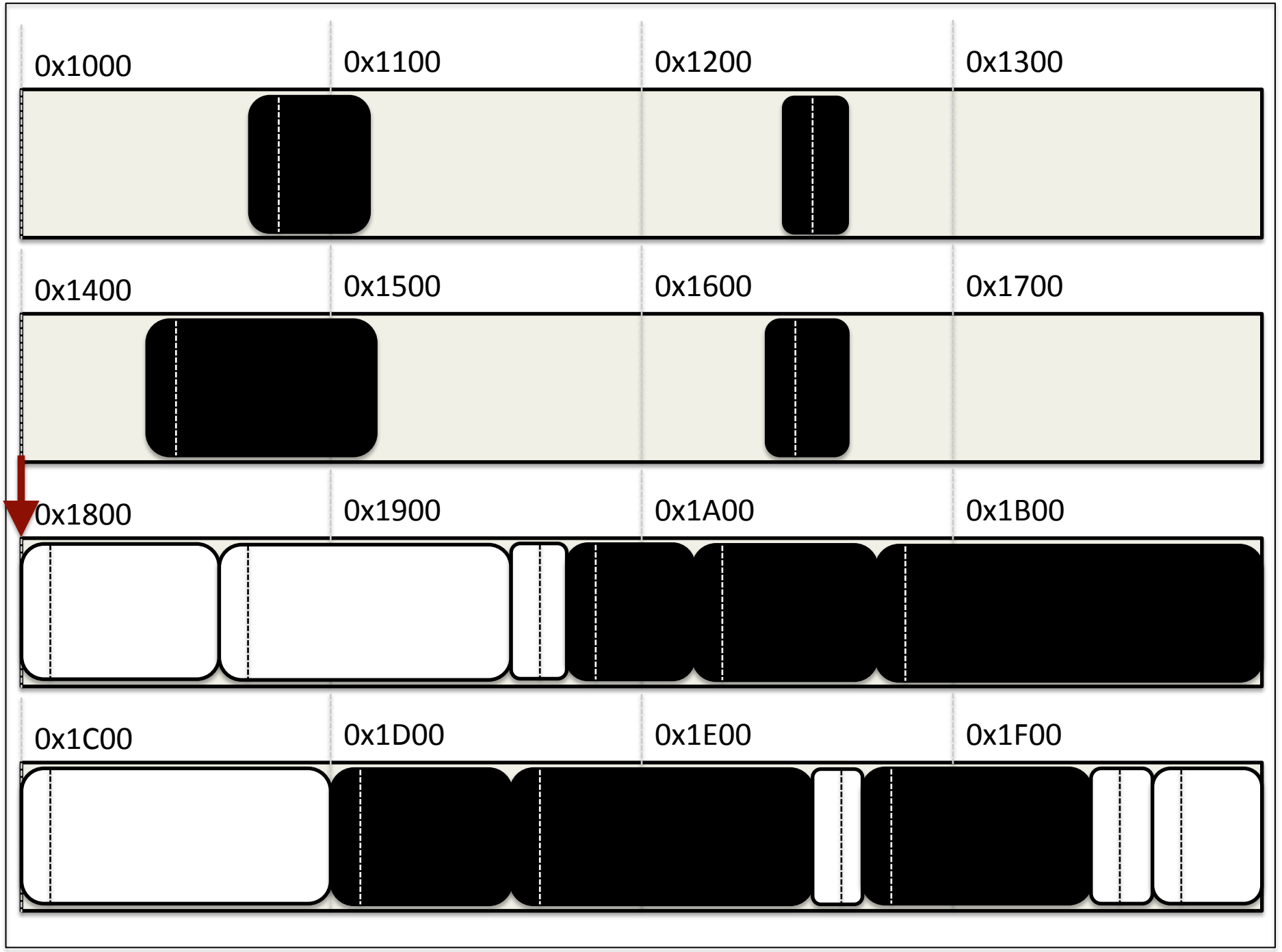

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

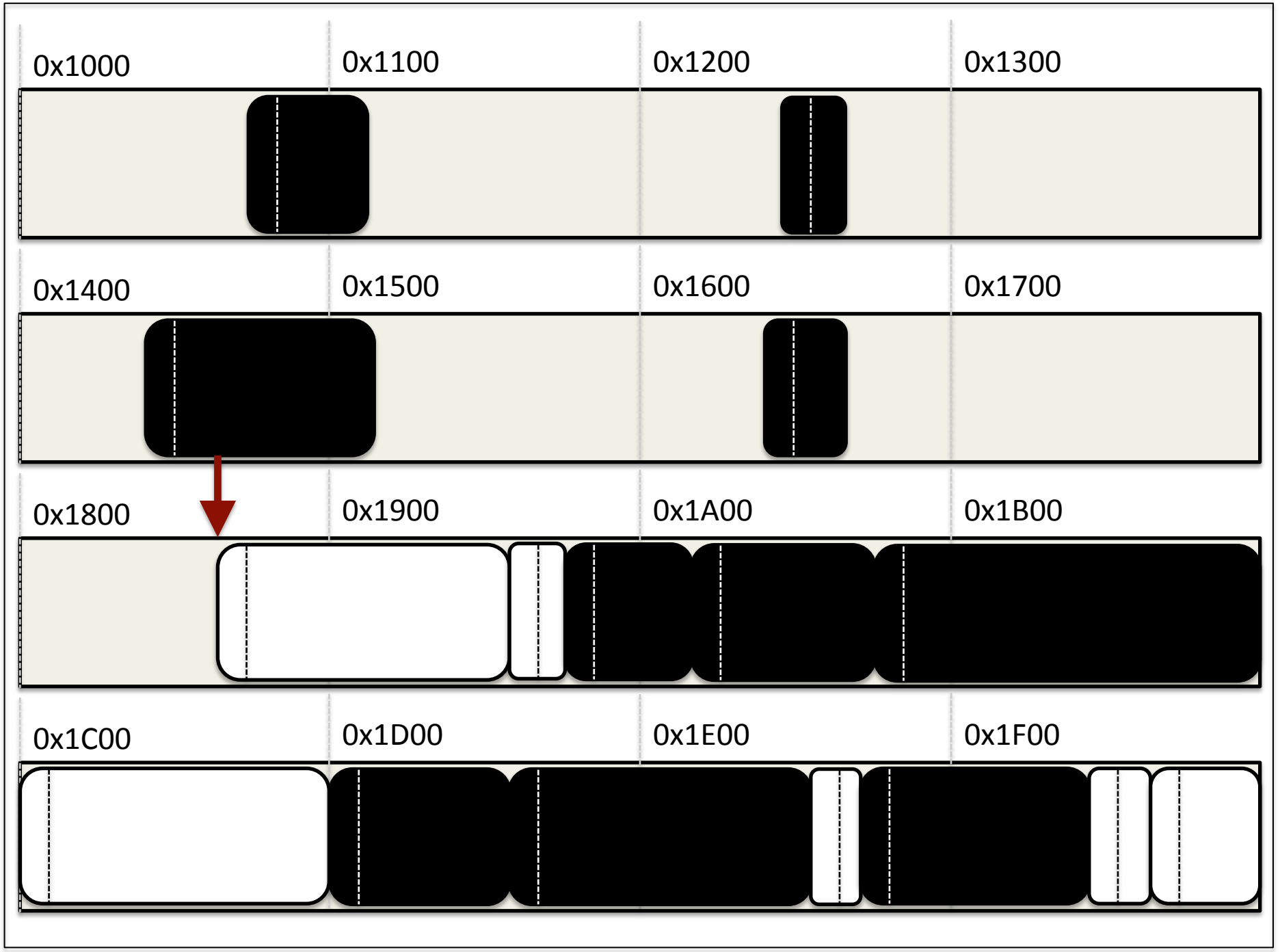

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

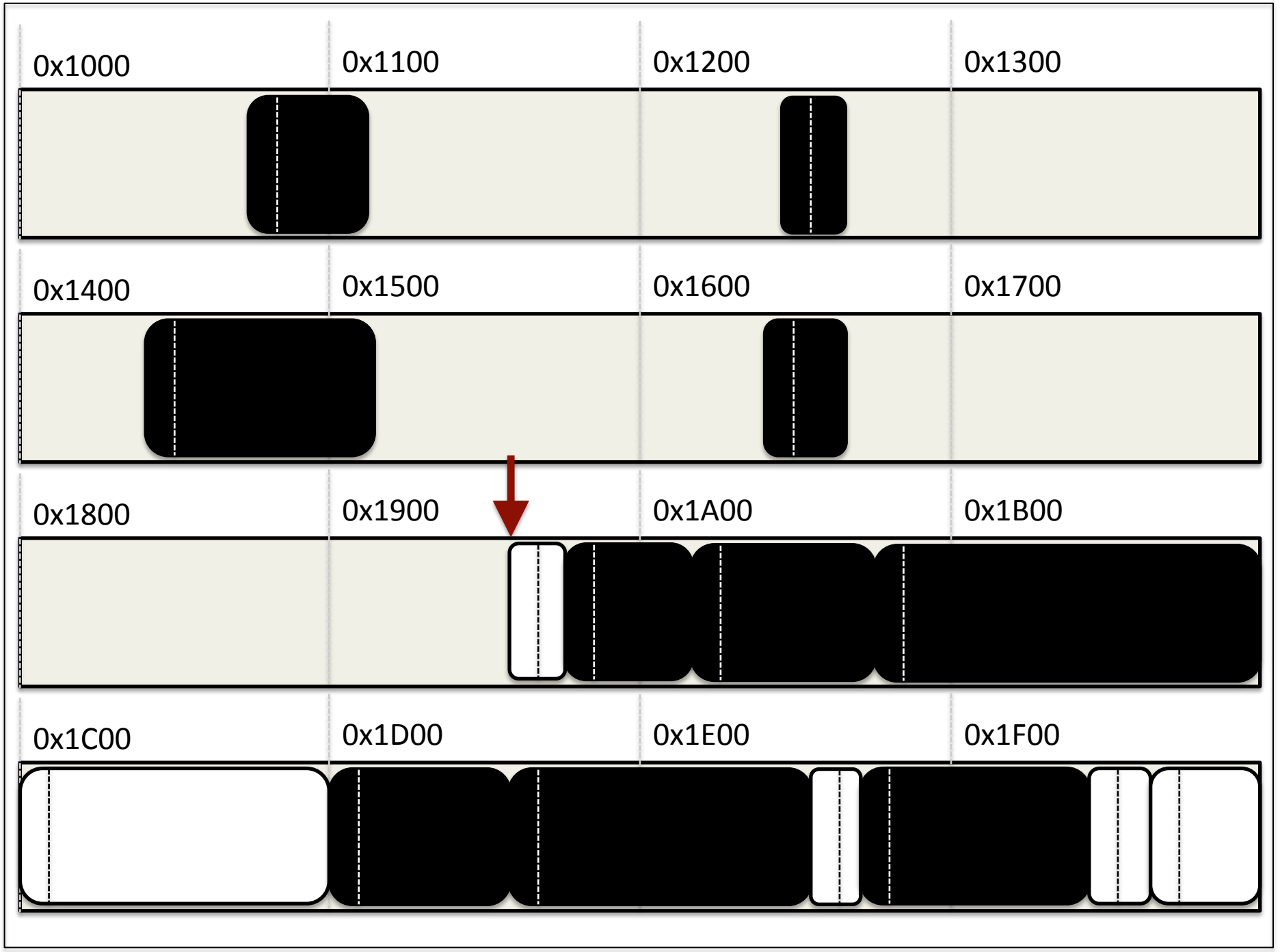

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

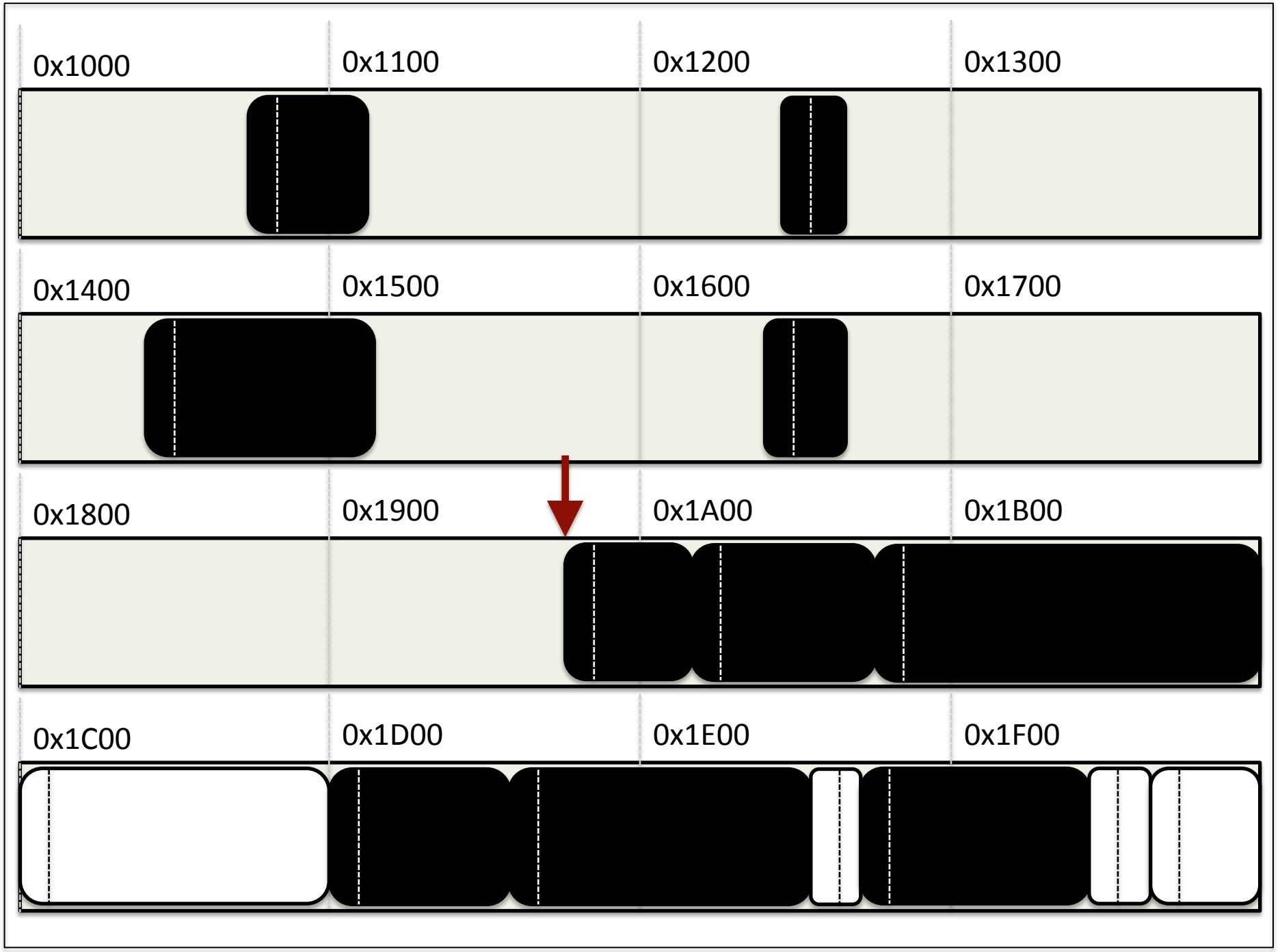
Sweep Phase

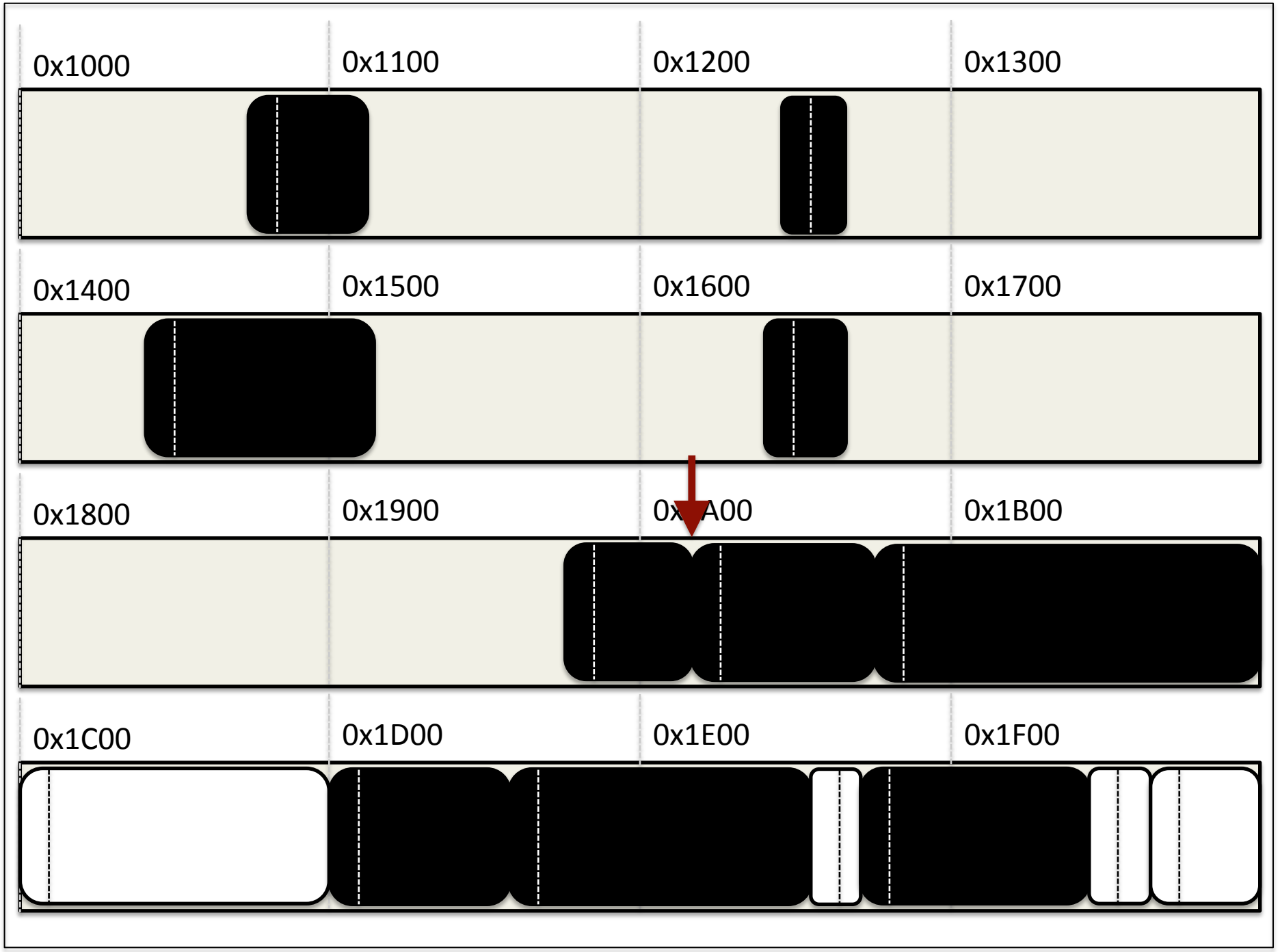
- Sweeping is a linear scan over the heap
  - Start at the first address
  - Calculate the offset of the next object
- Need to be able to scan over gaps
  - Fixed-size blocks
  - Block metadata as in malloc algorithm
  - Create dummy objects in empty spaces

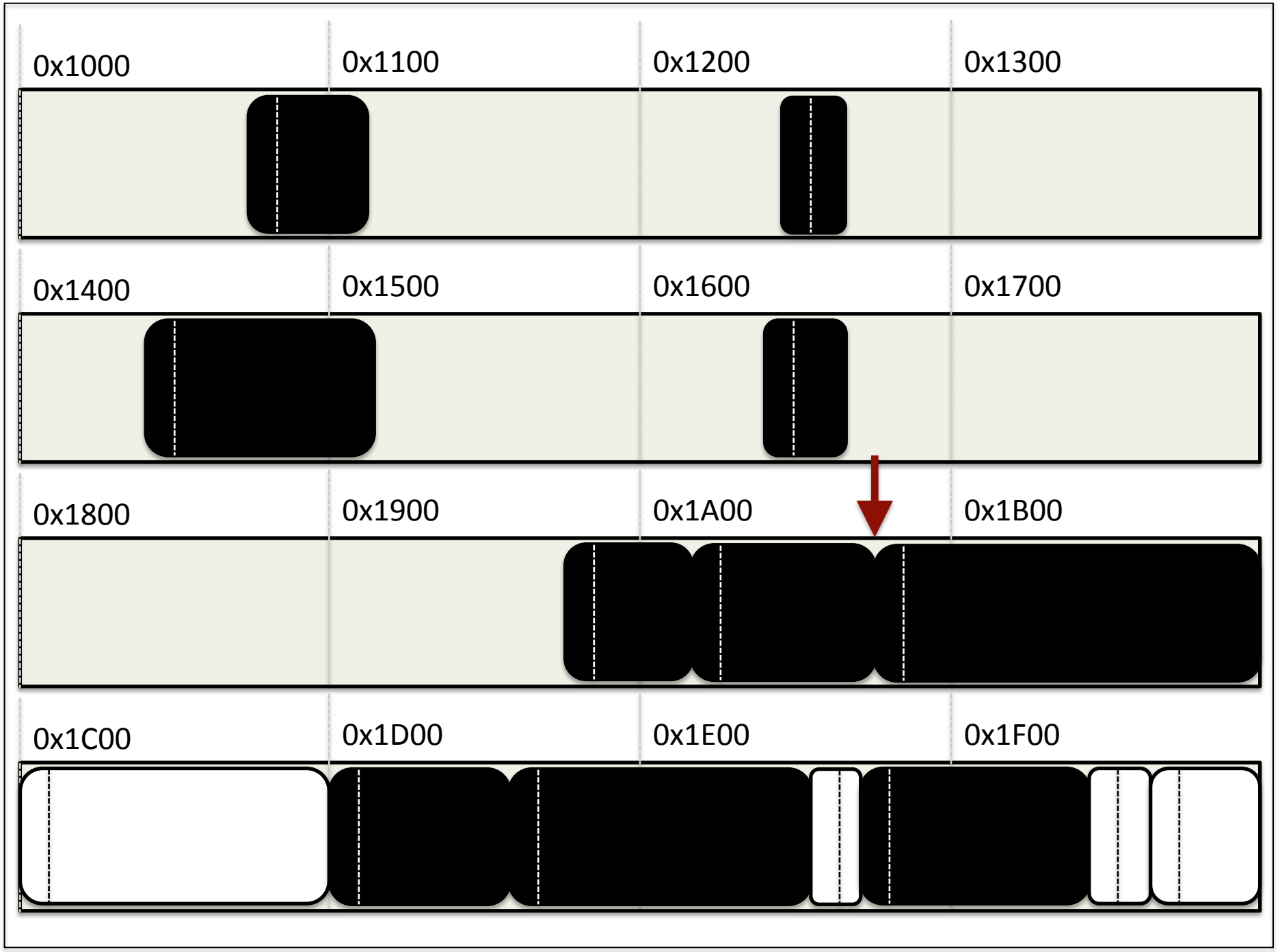

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

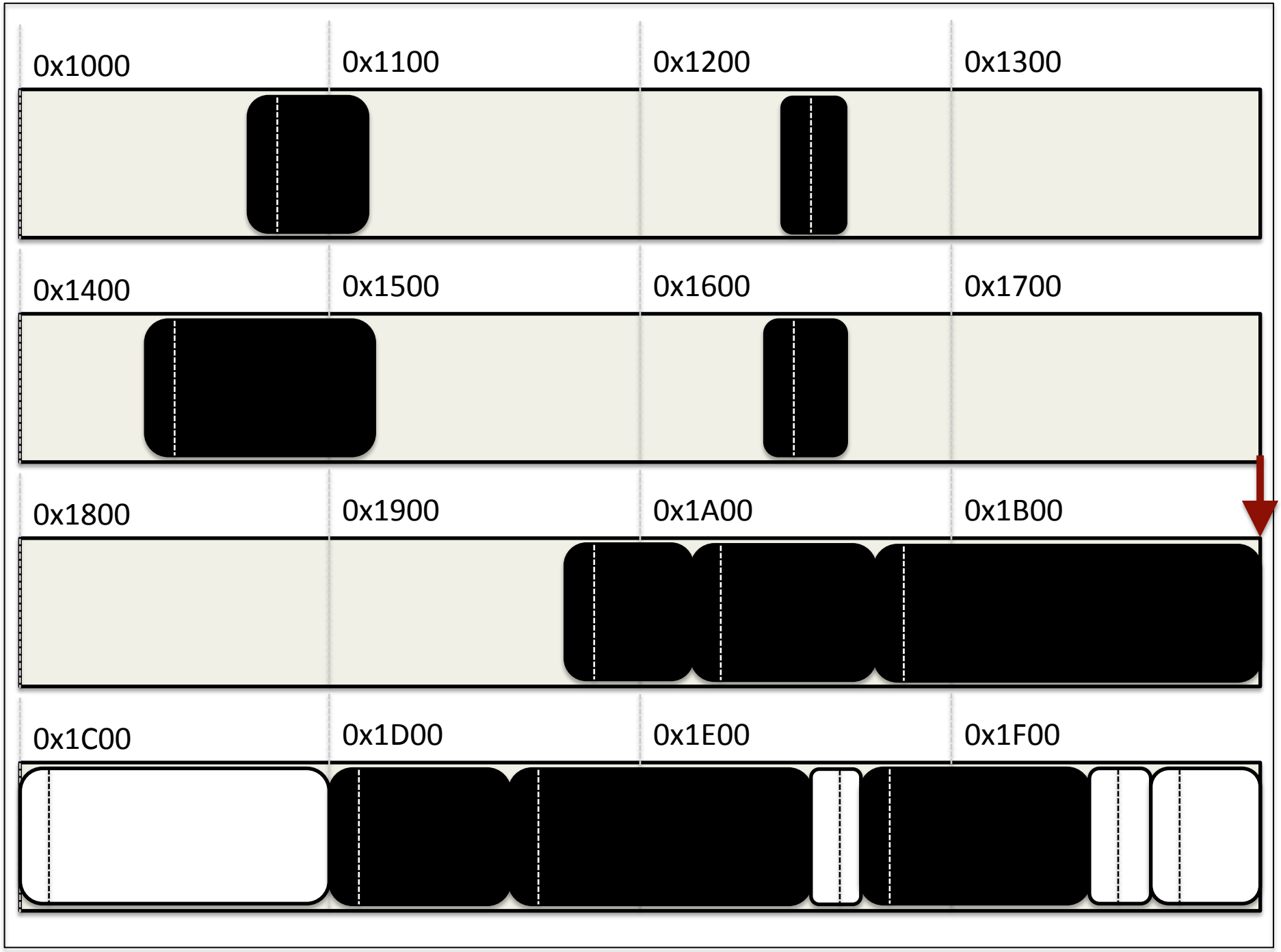

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

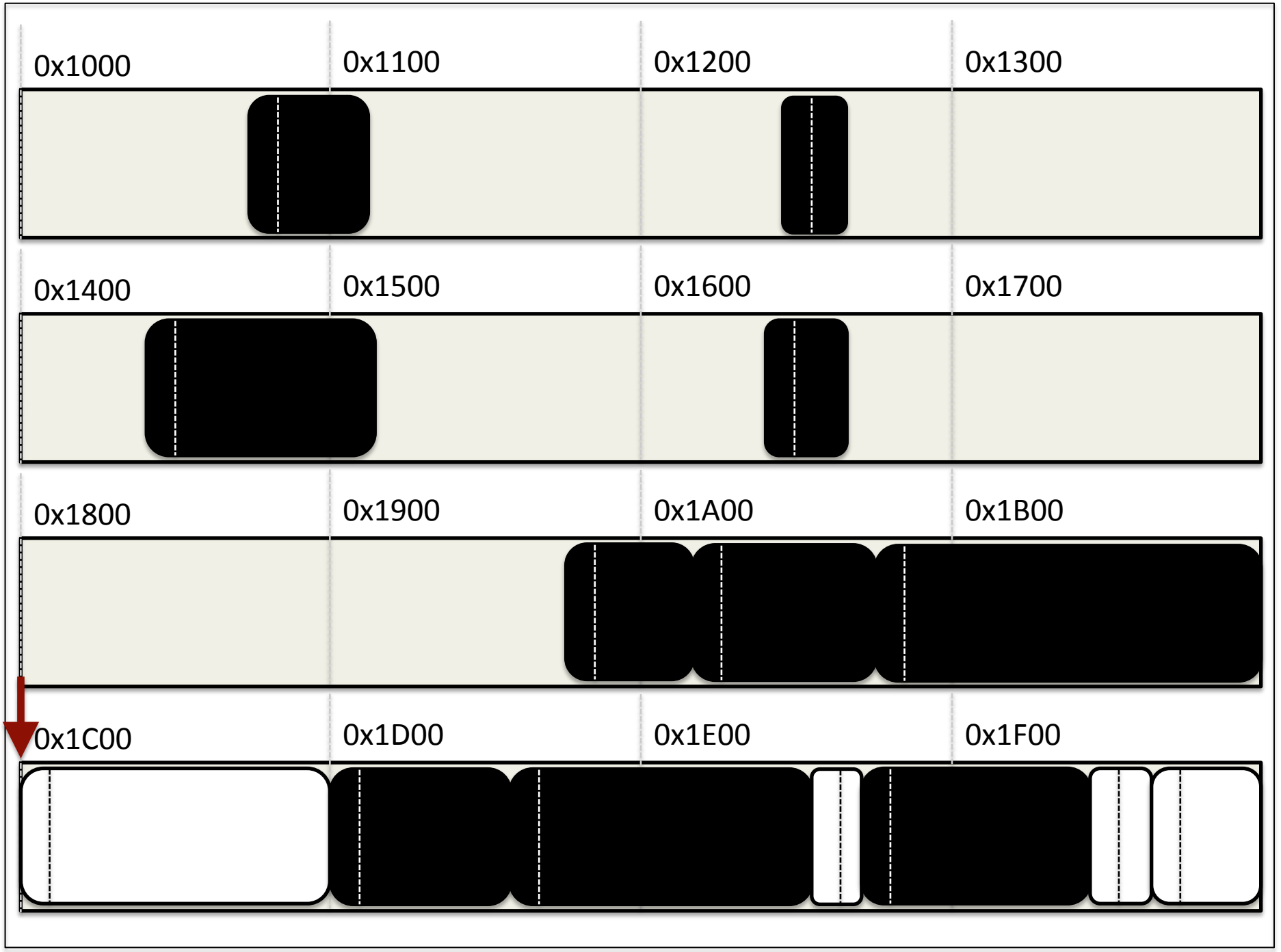

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

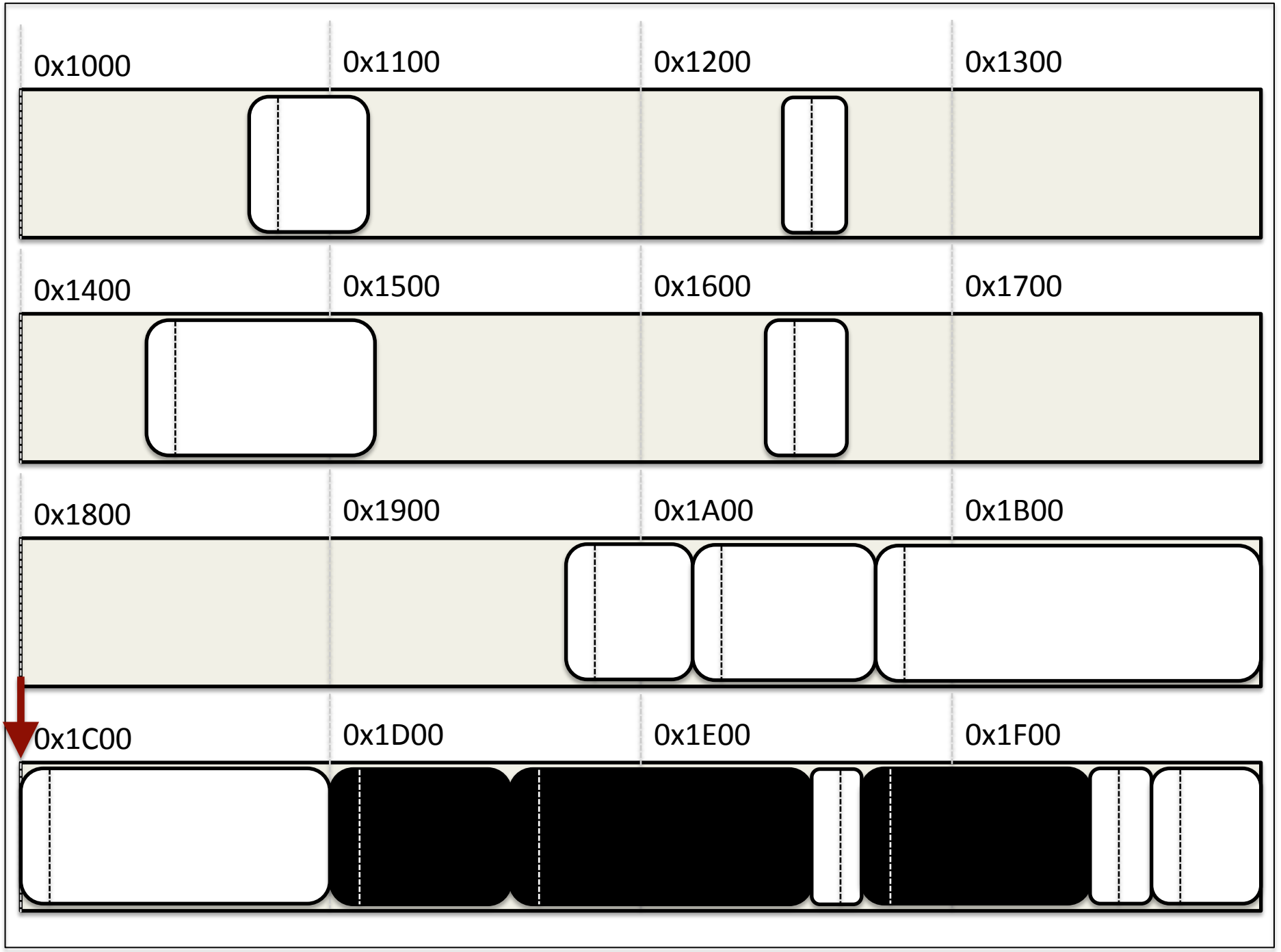

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

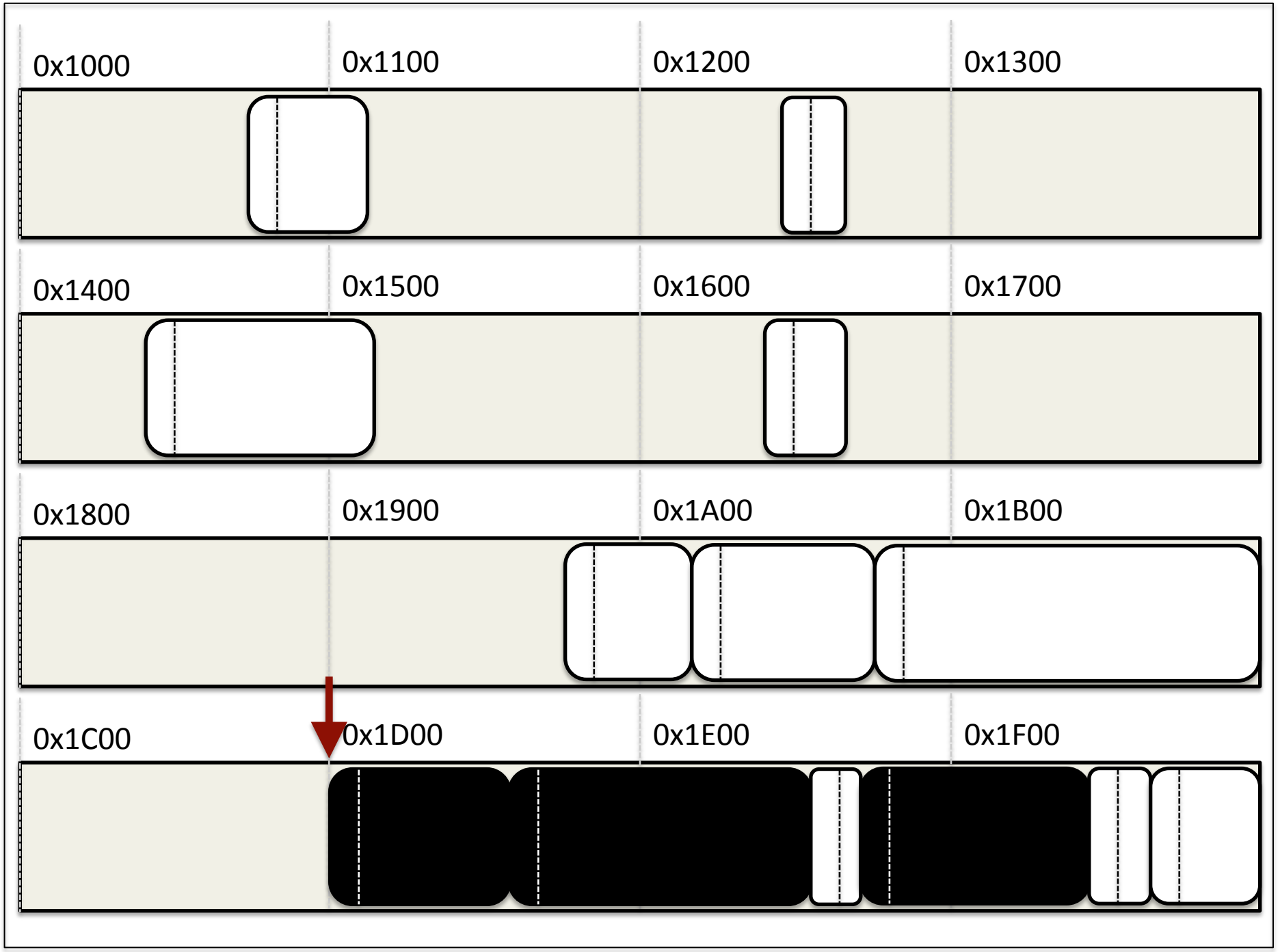

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

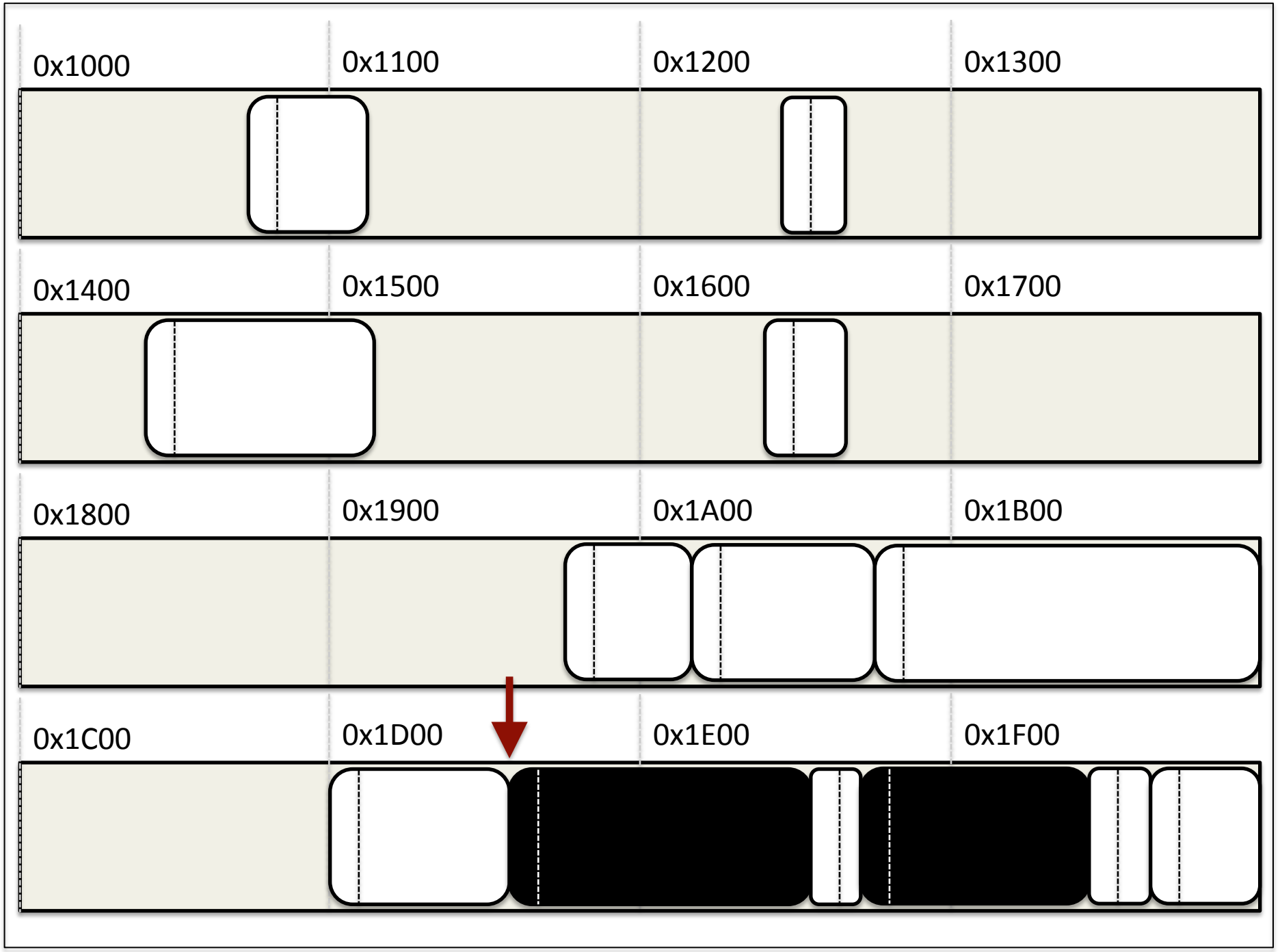

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

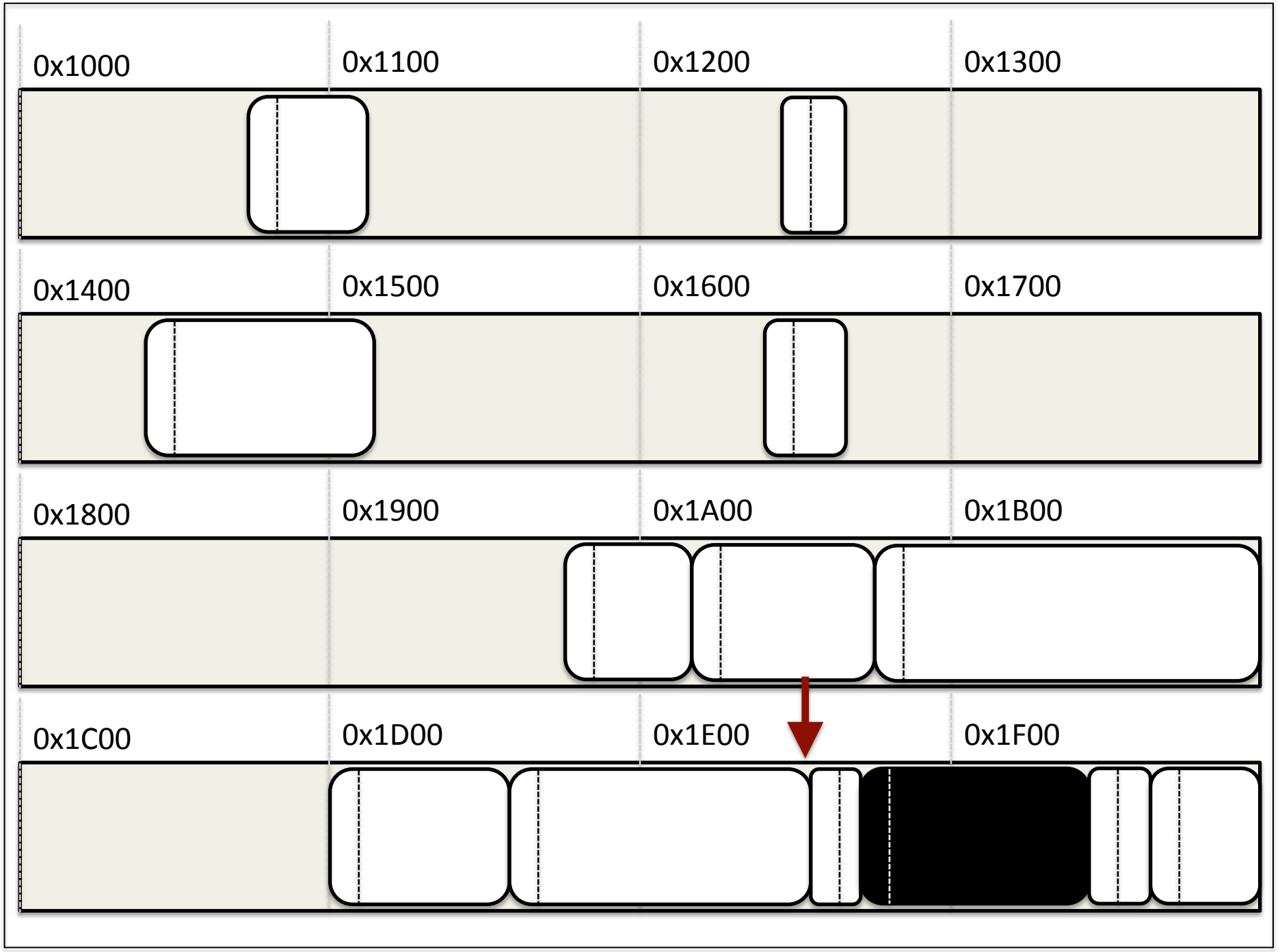
Finishing Up

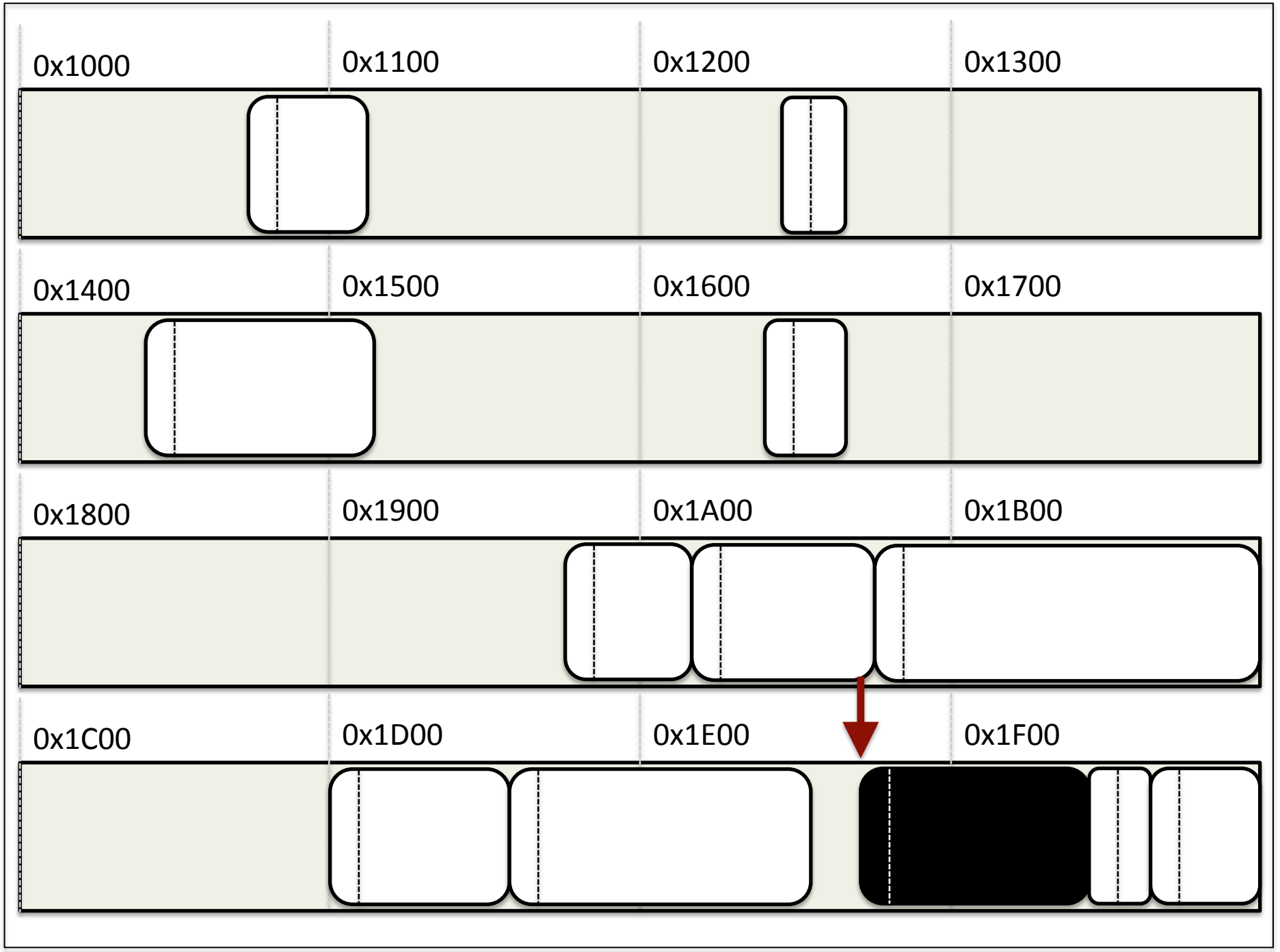
- Clear all mark bits
  - Reset the heap ready for the next collection
- Can alternate the mark bit between GCs
  - Allocate objects with mark bit set
  - Introduces complexity if you are using bit stealing
- Can clear as you go with the sweep phase
  - Once the pointer passes an object it is done with

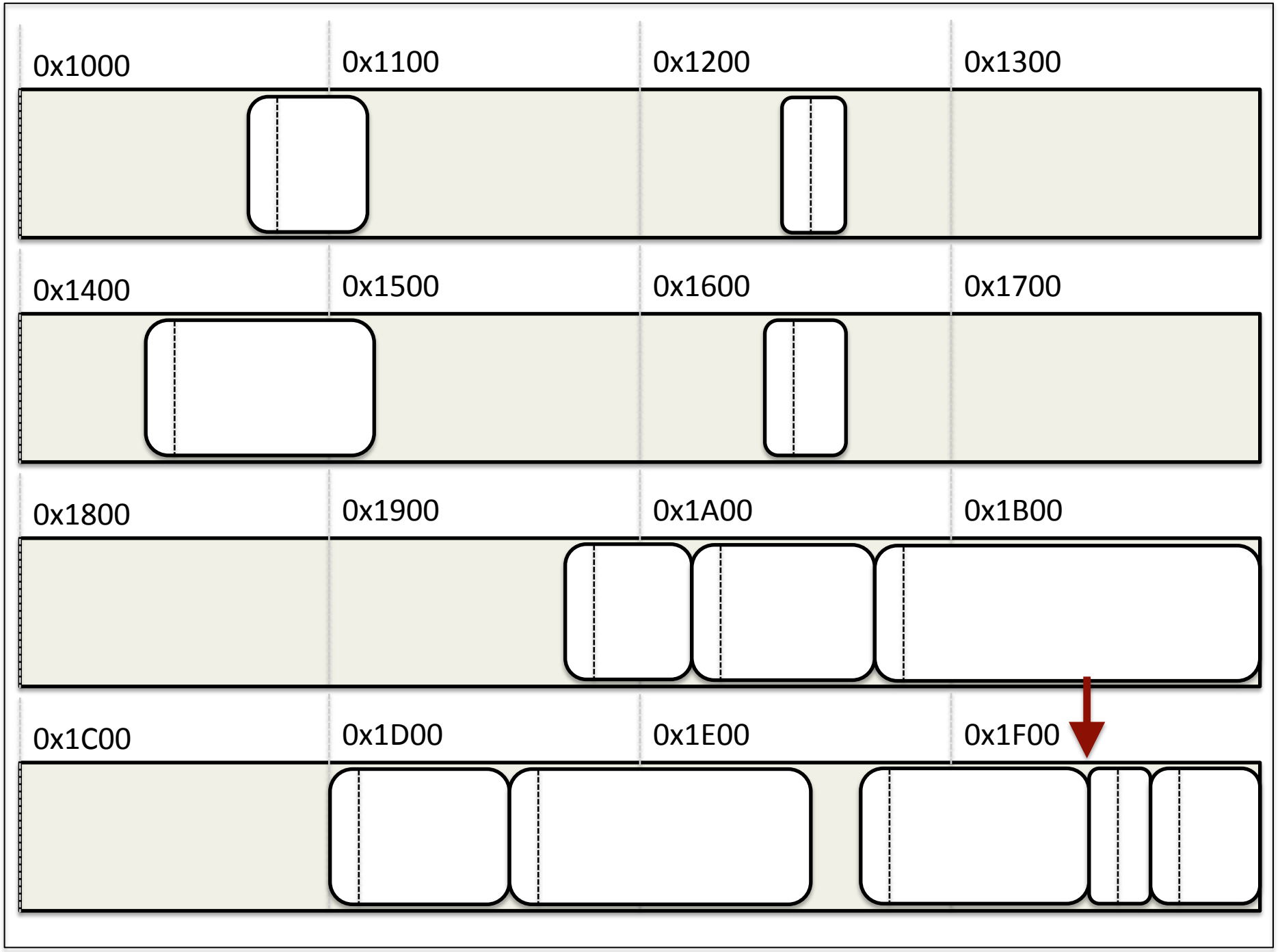

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

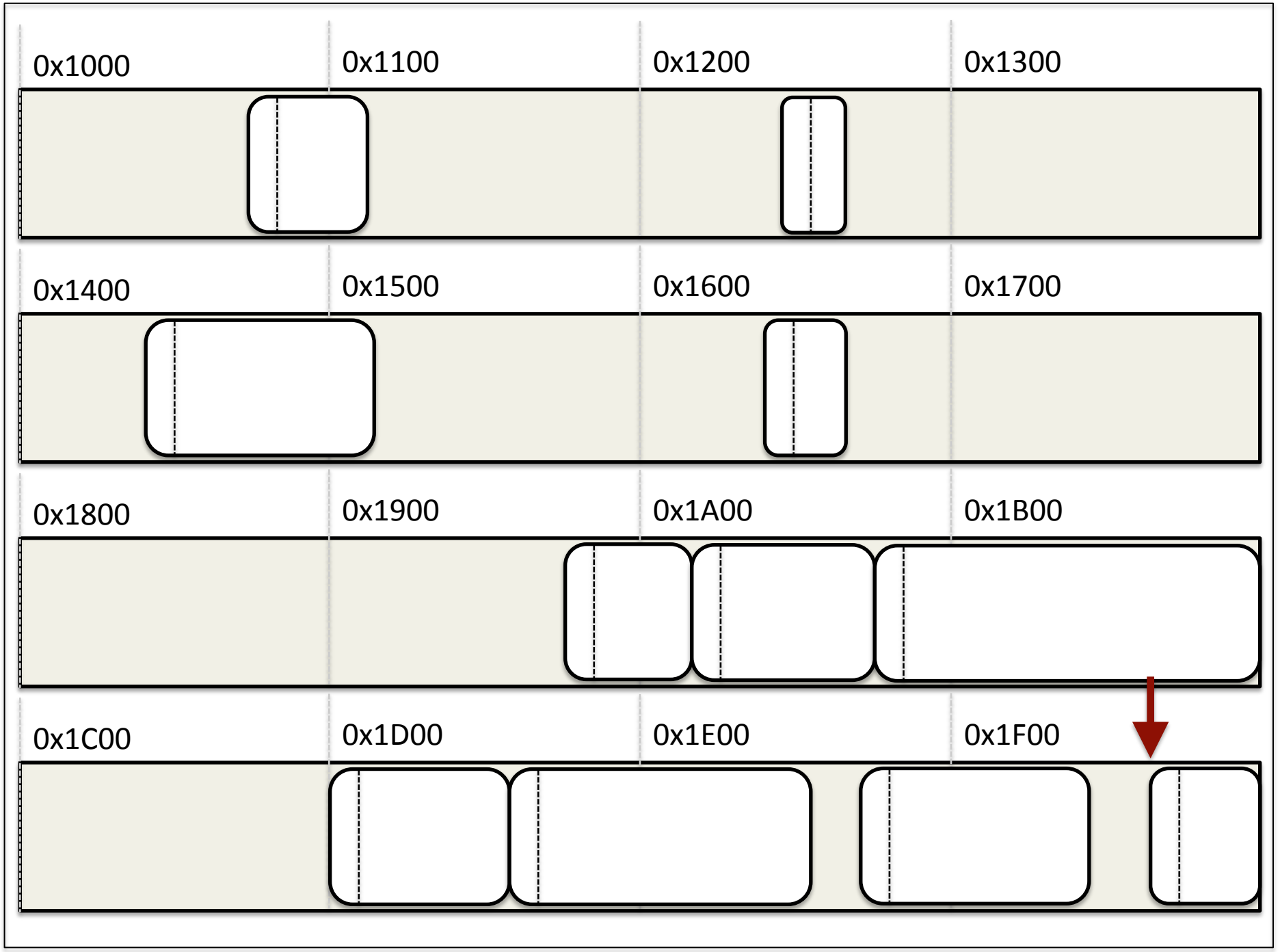

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

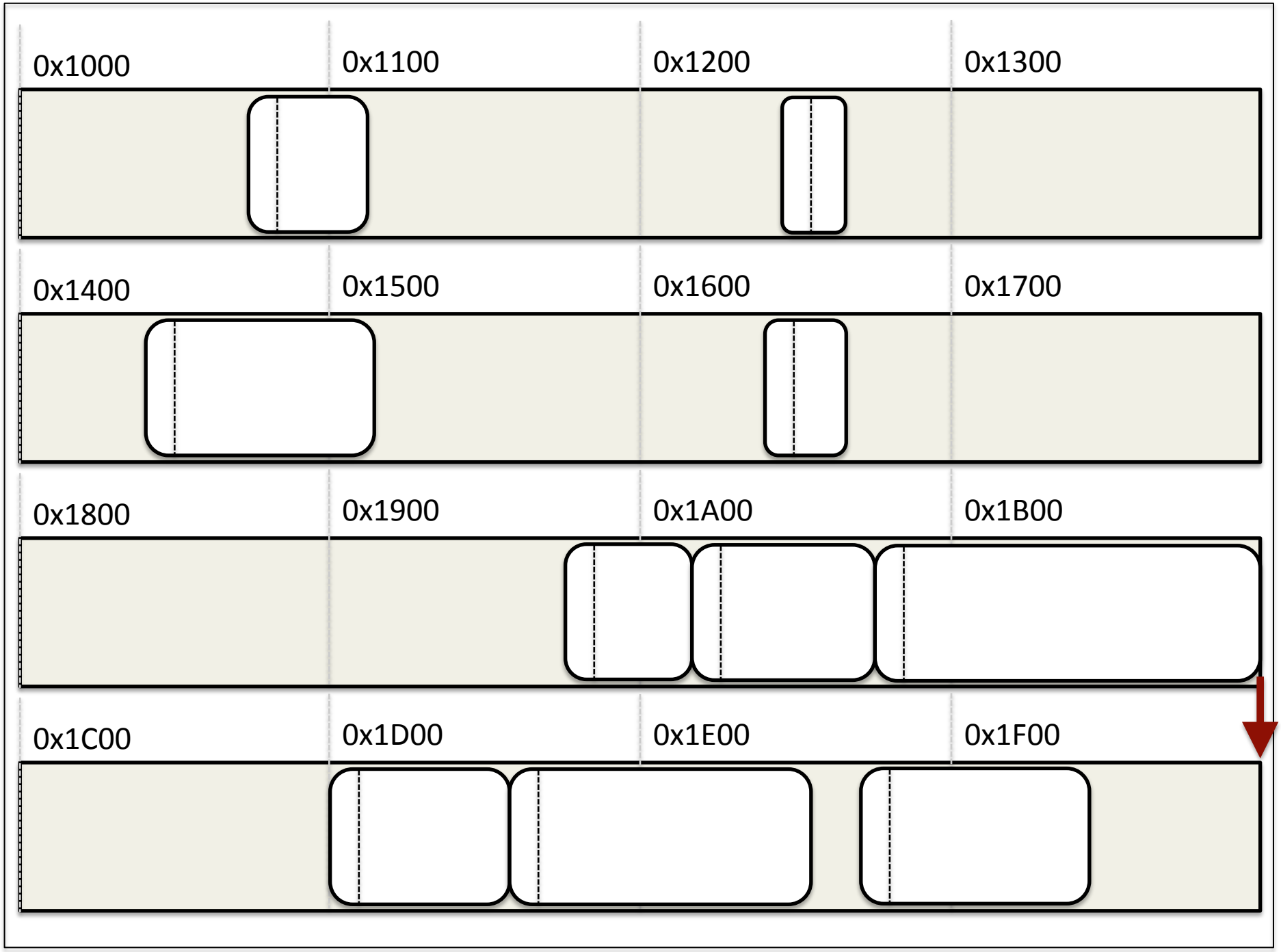

0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1000
0x1100
0x1200
0x1300
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

Performance Characteristics

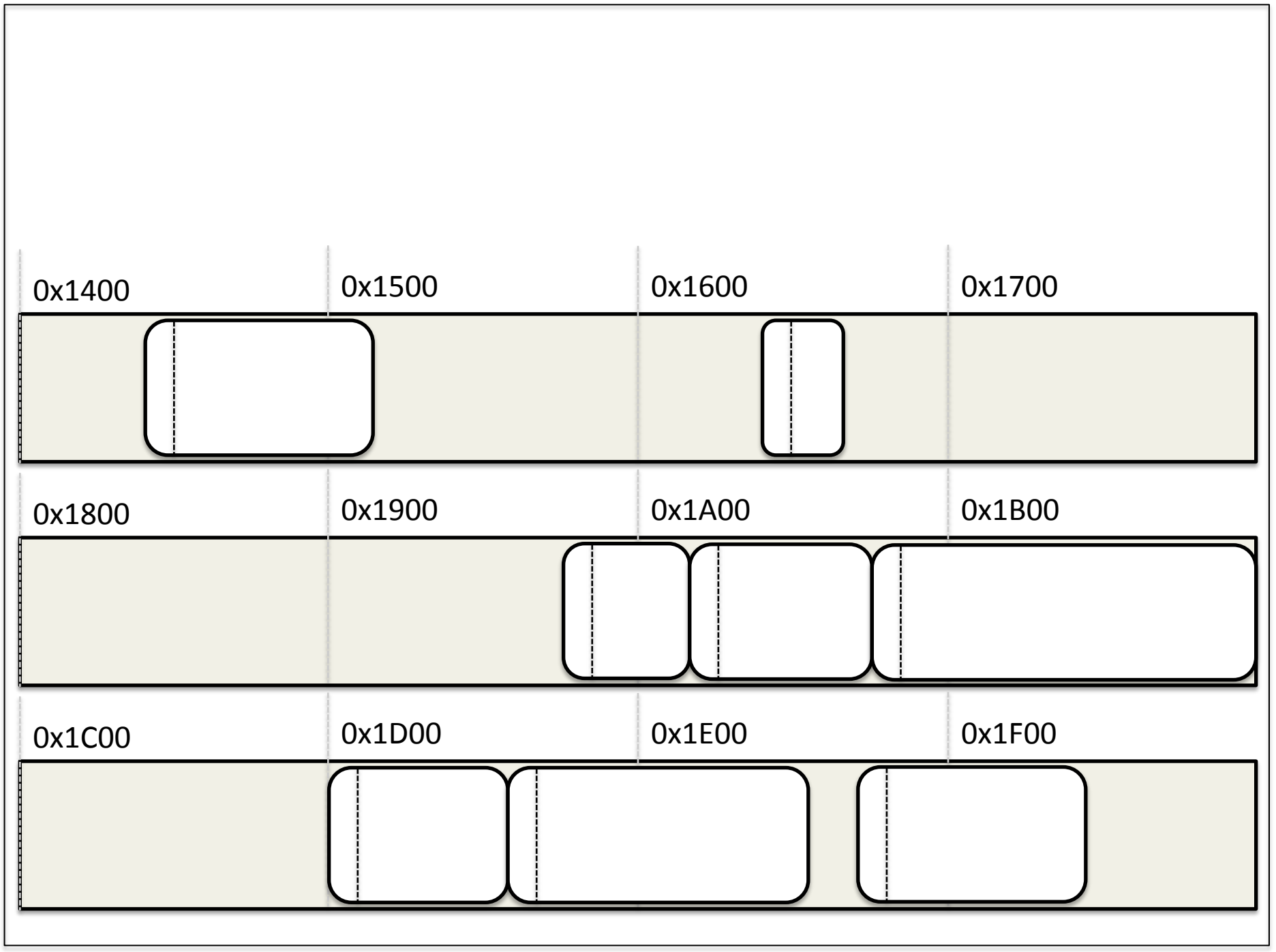
- Two passes over the heap
  - First phase random(ish) access
  - Second phase linear
- Not good for the cache
  - Second phase can be mitigated by prefetching
  - Mutator's cache completely blown

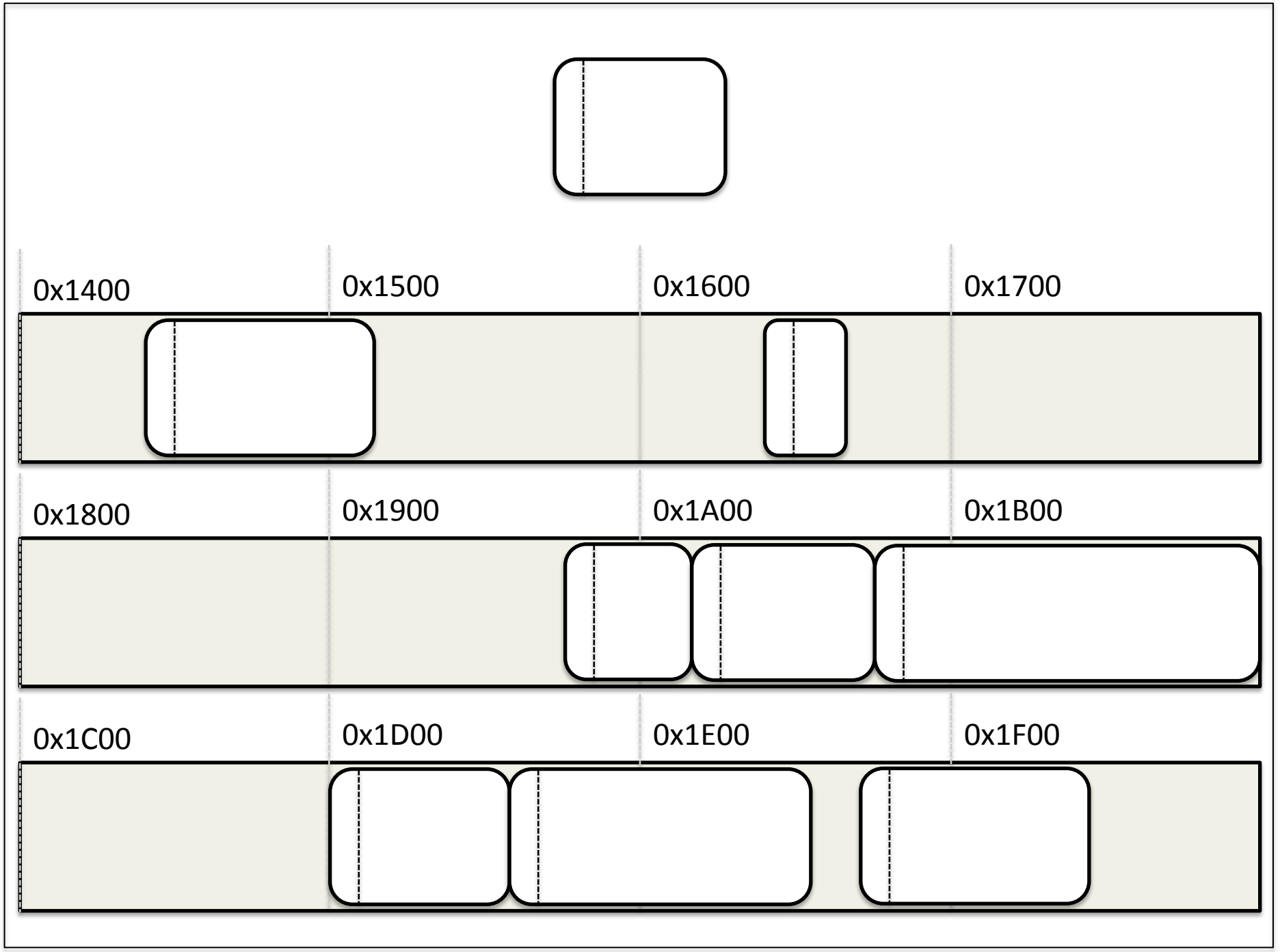
Mark and Sweep Allocation

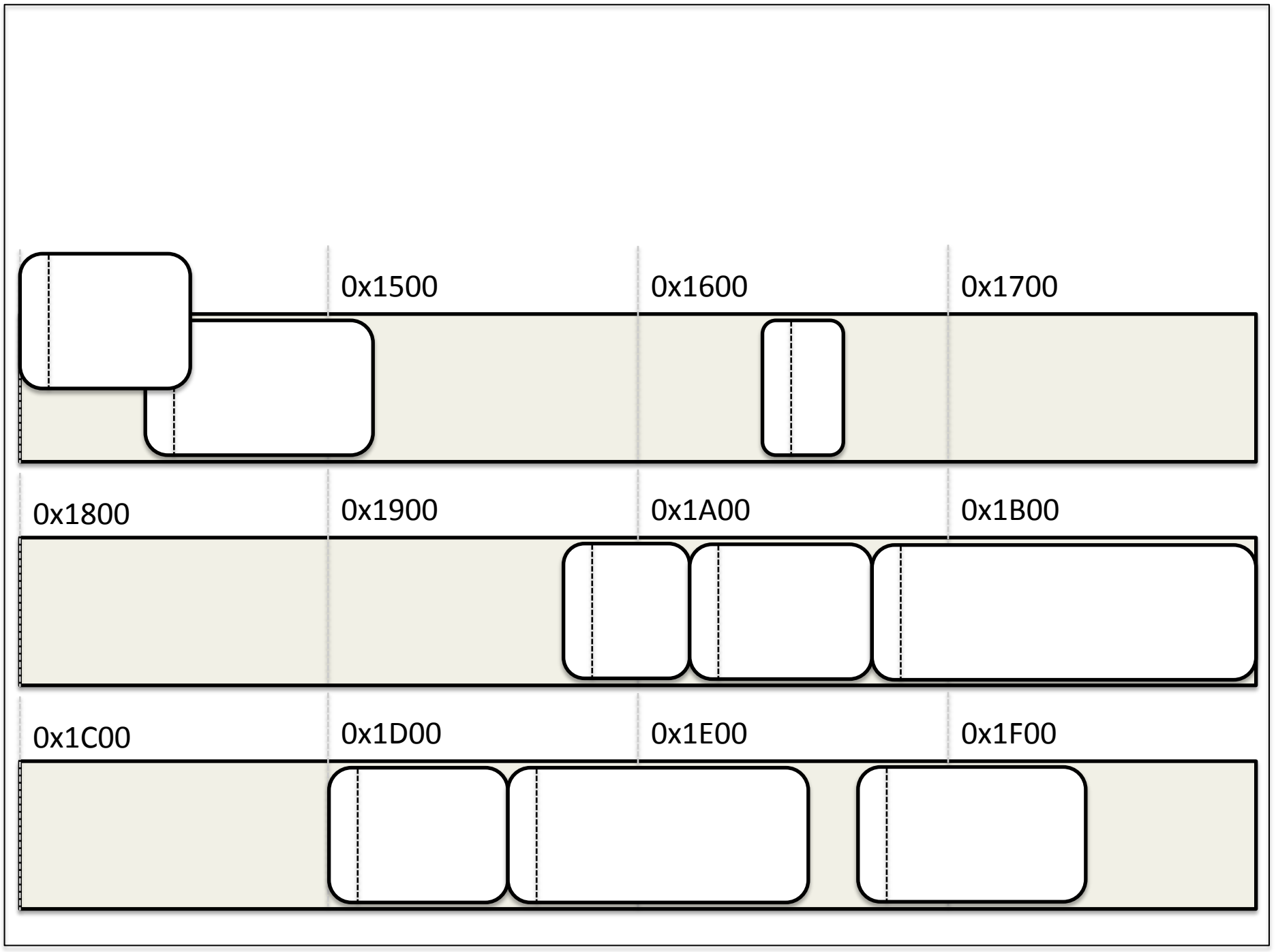
- Sweep phase creates holes in the memory space
  - Known as fragmentation
- Allocator needs to fit objects to those holes
- Danger of wasting space if no object fits

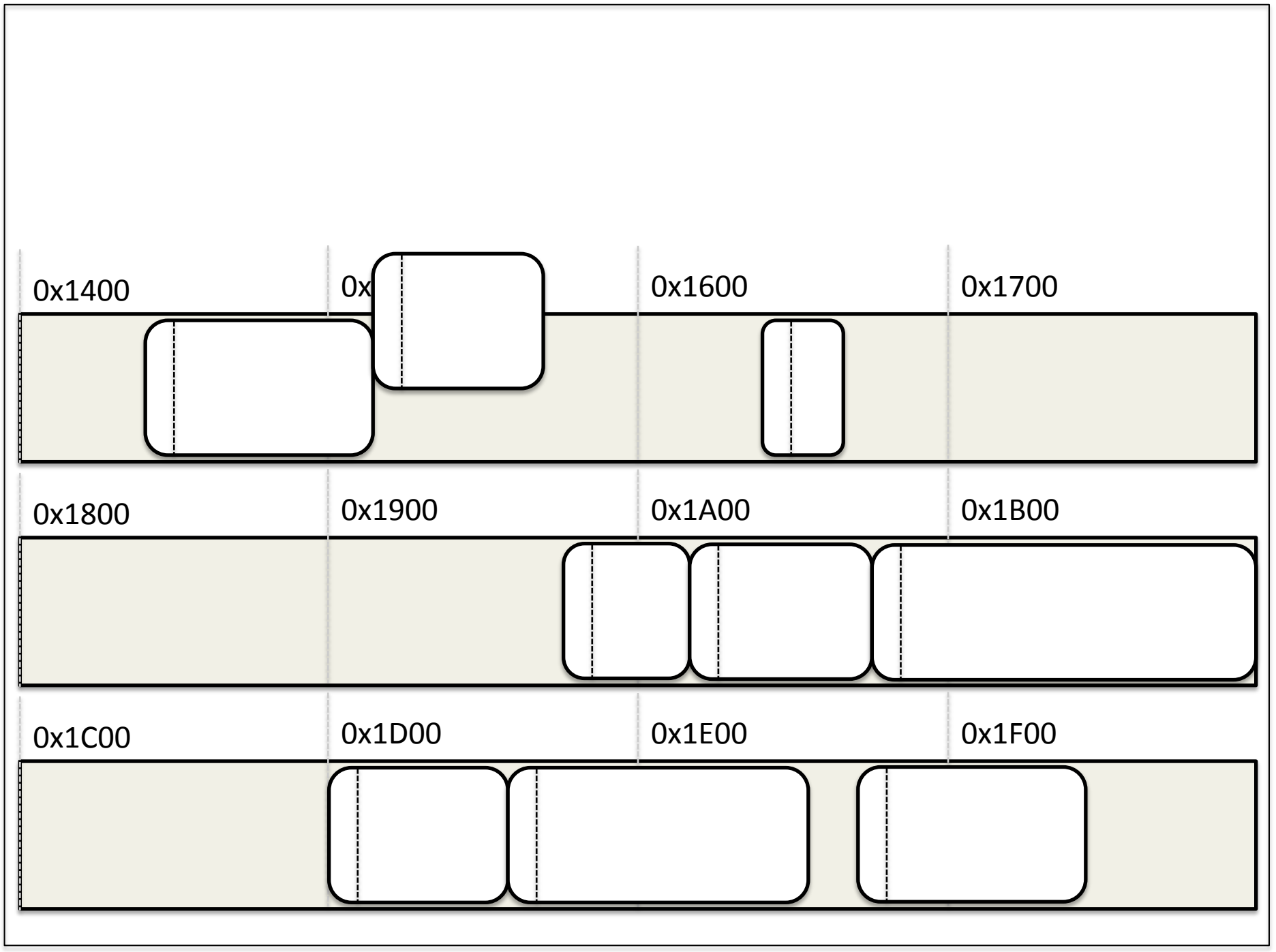
Allocation Strategies

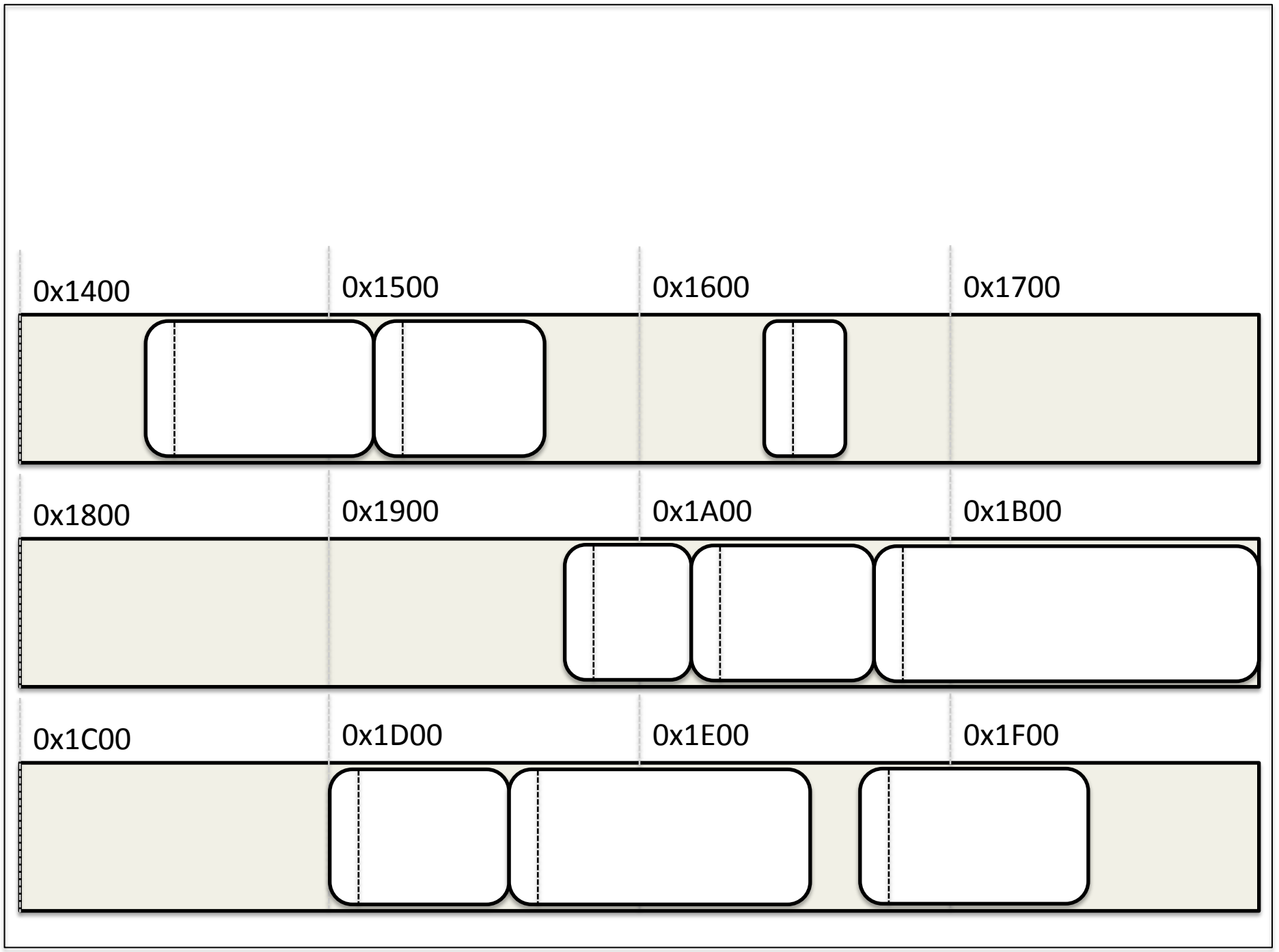
- First Fit
  - Put new object in the first hole that's large enough


0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

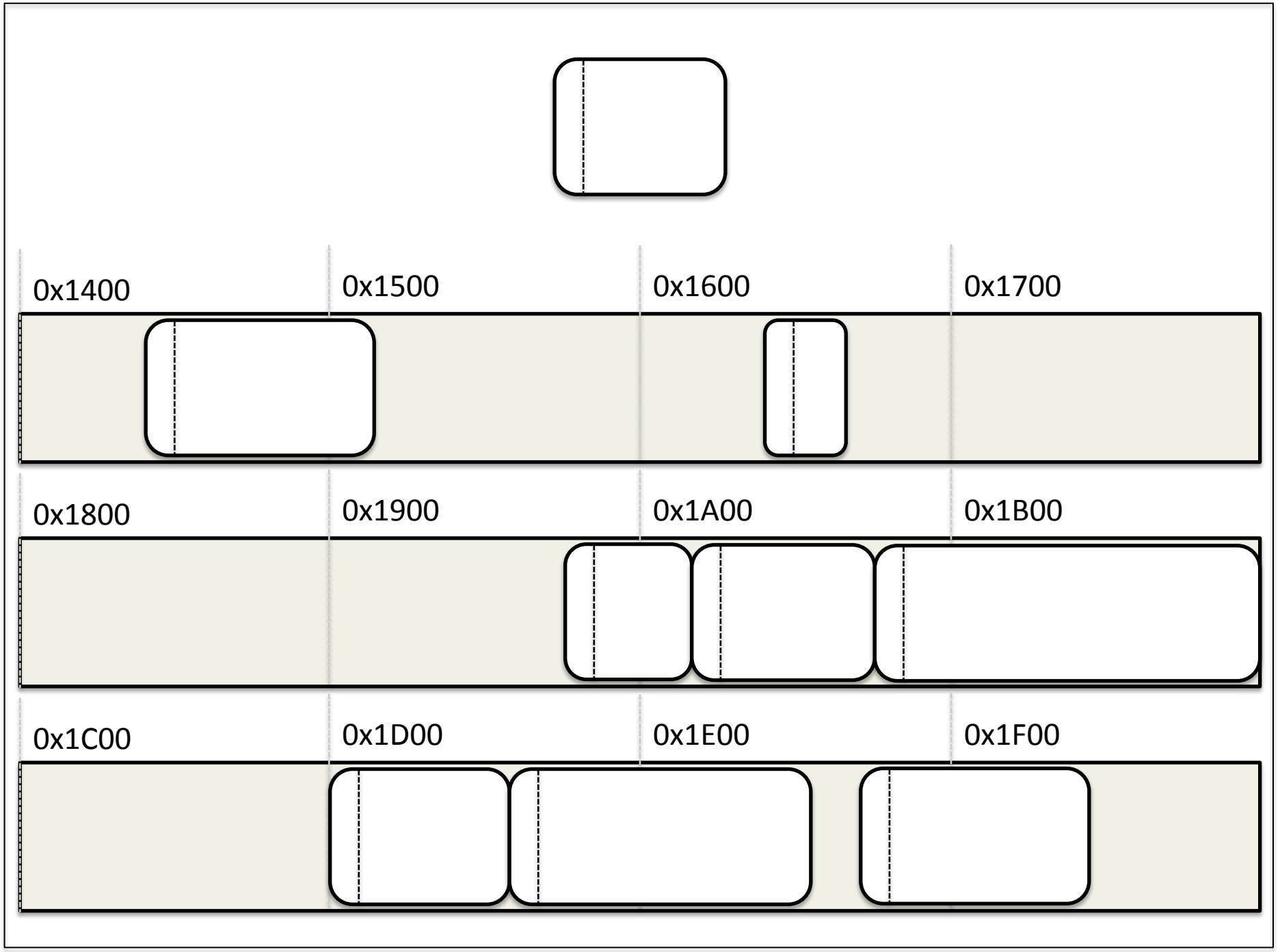

0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

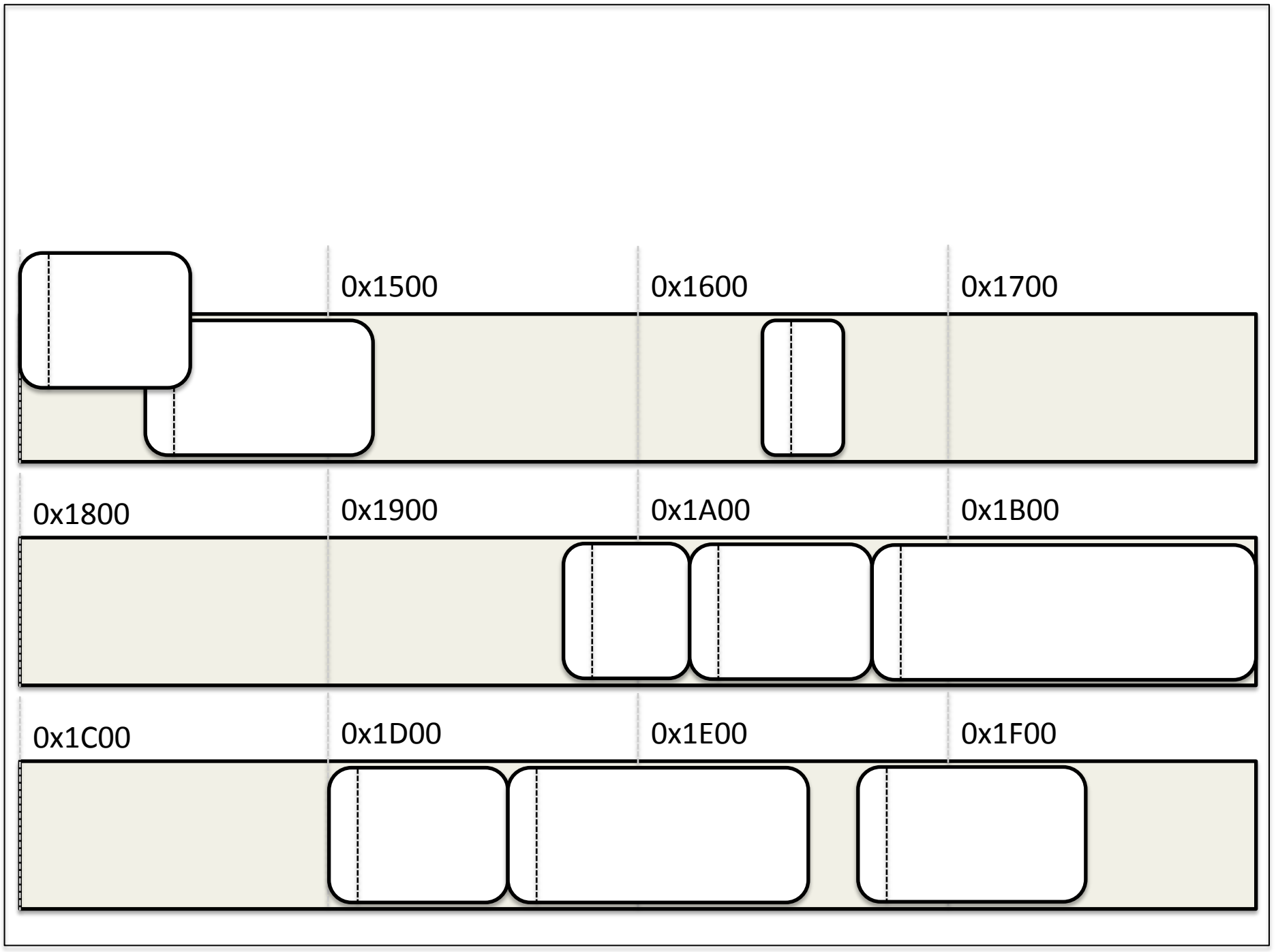

0x1400
0x
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

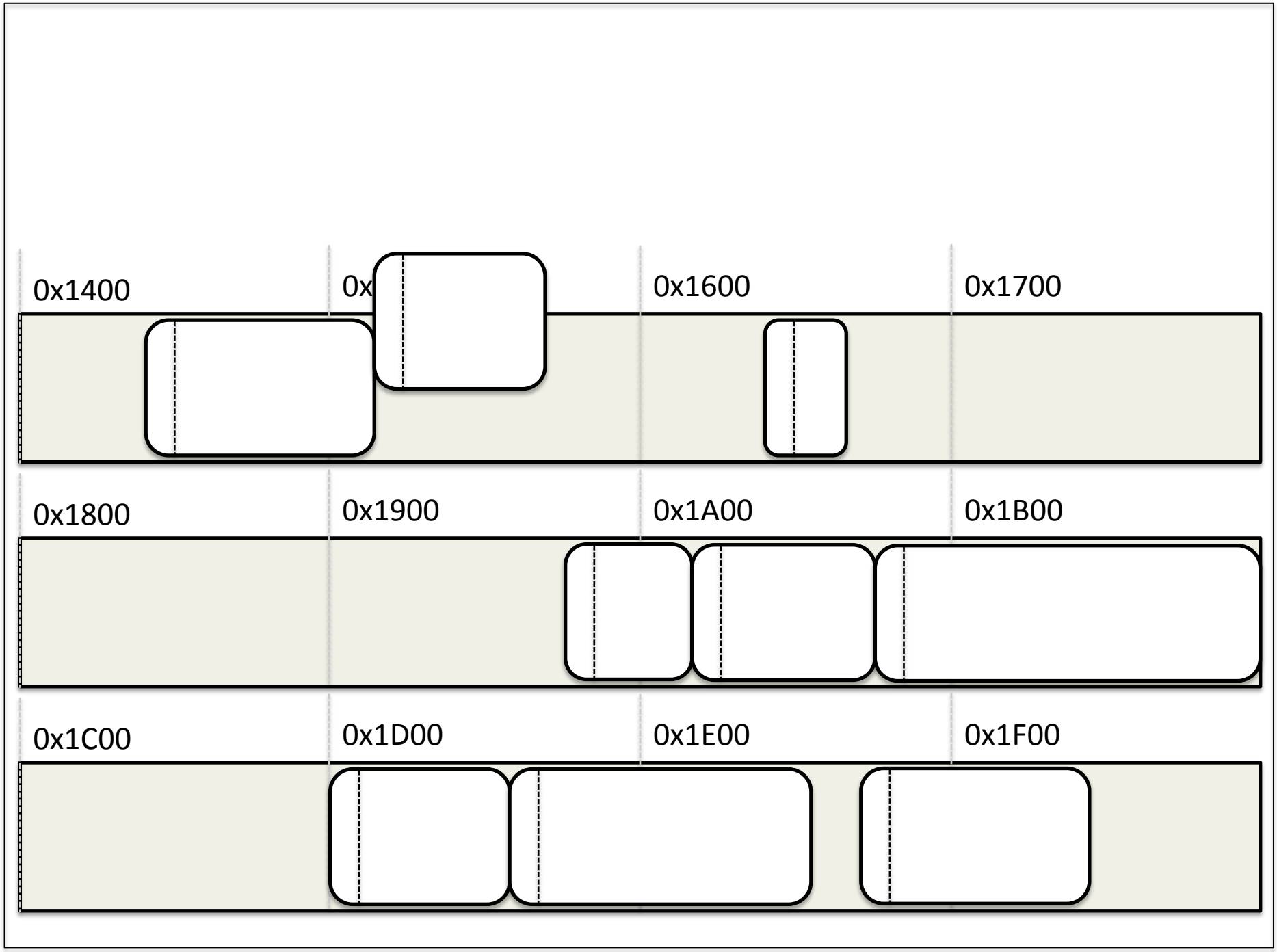

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

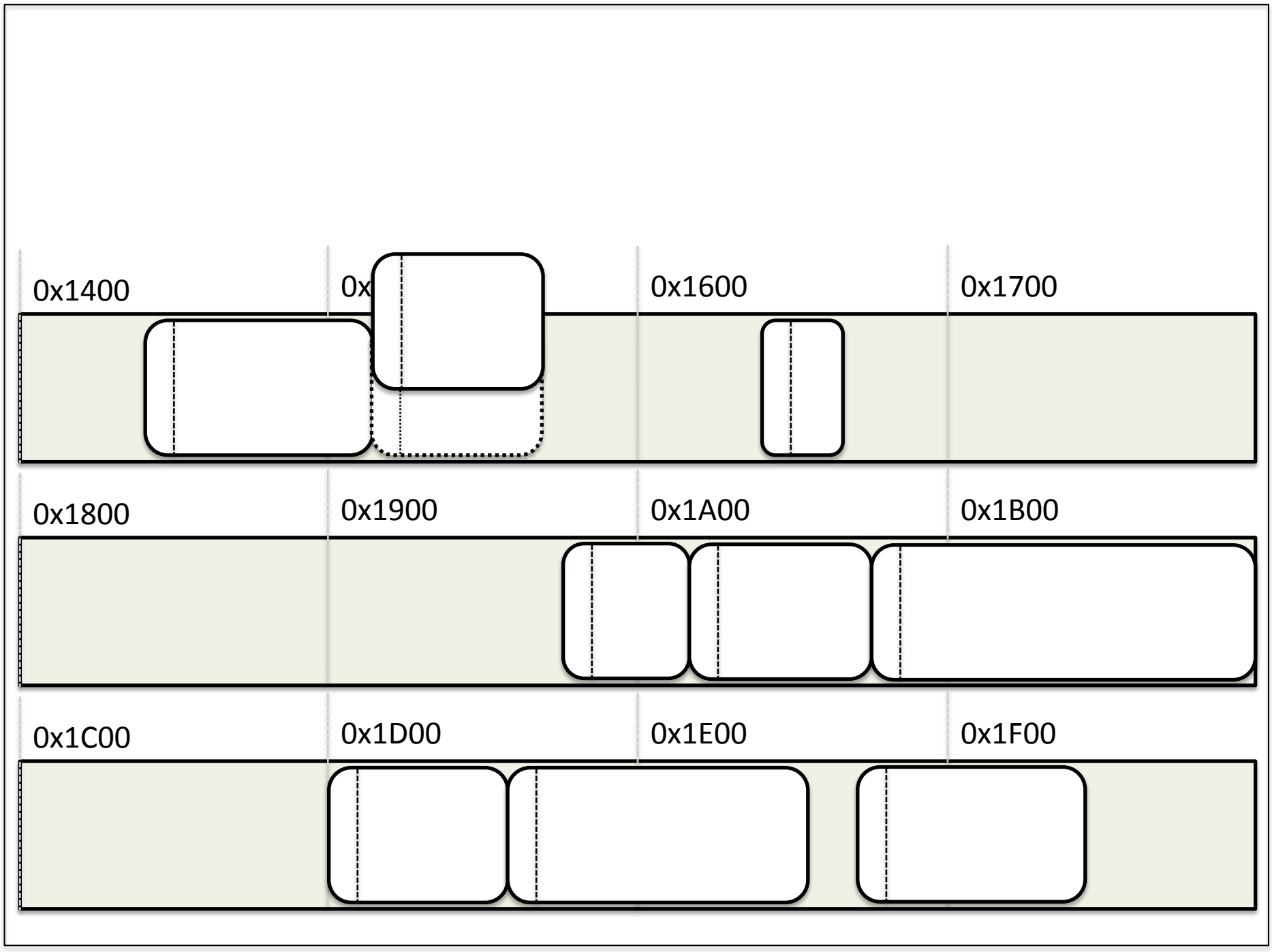
Allocation Strategies

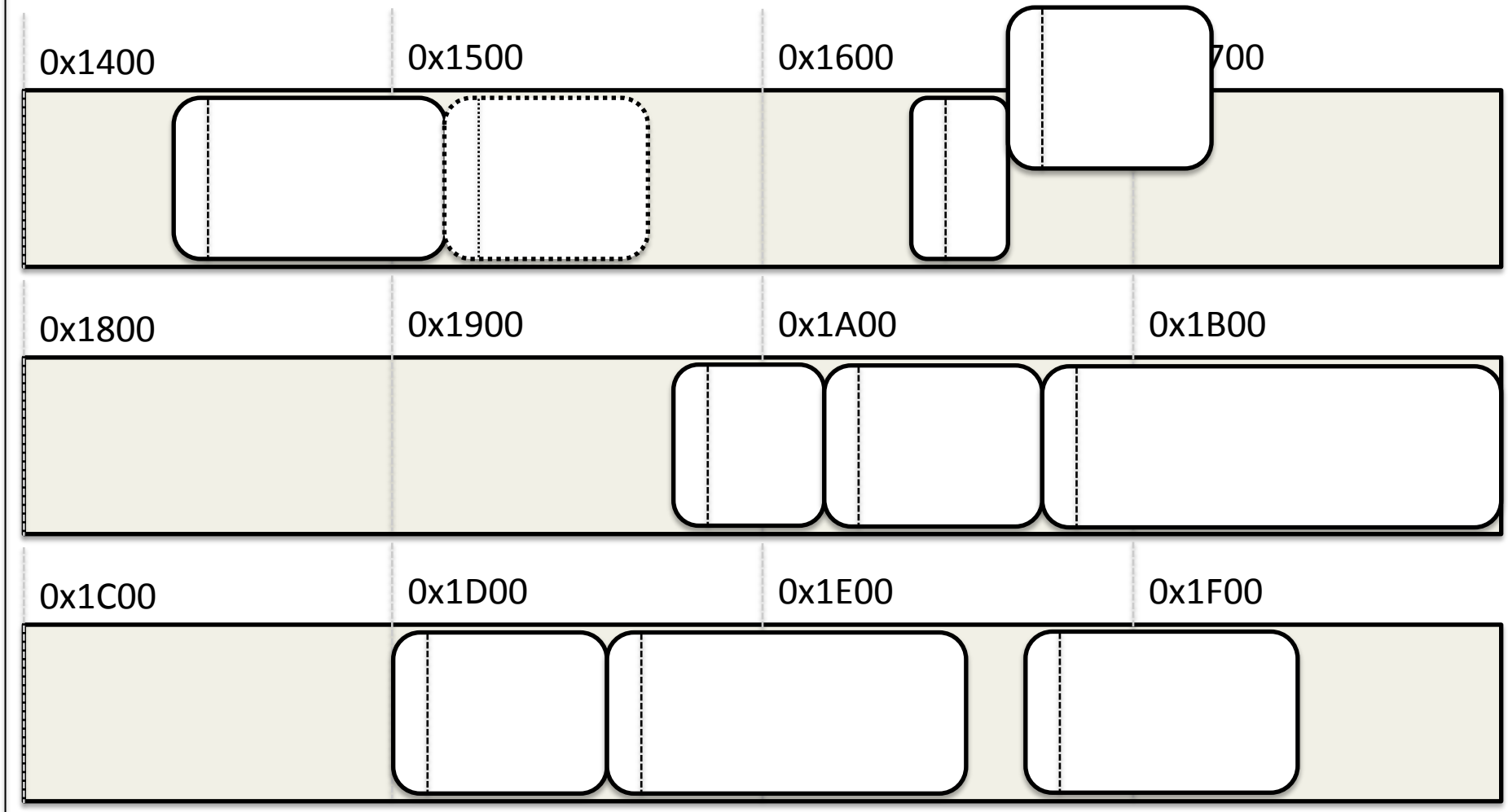
- First Fit
  - Put new object in the first hole that's large enough
- Best Fit
  - Scan the heap to find the hole that fits best

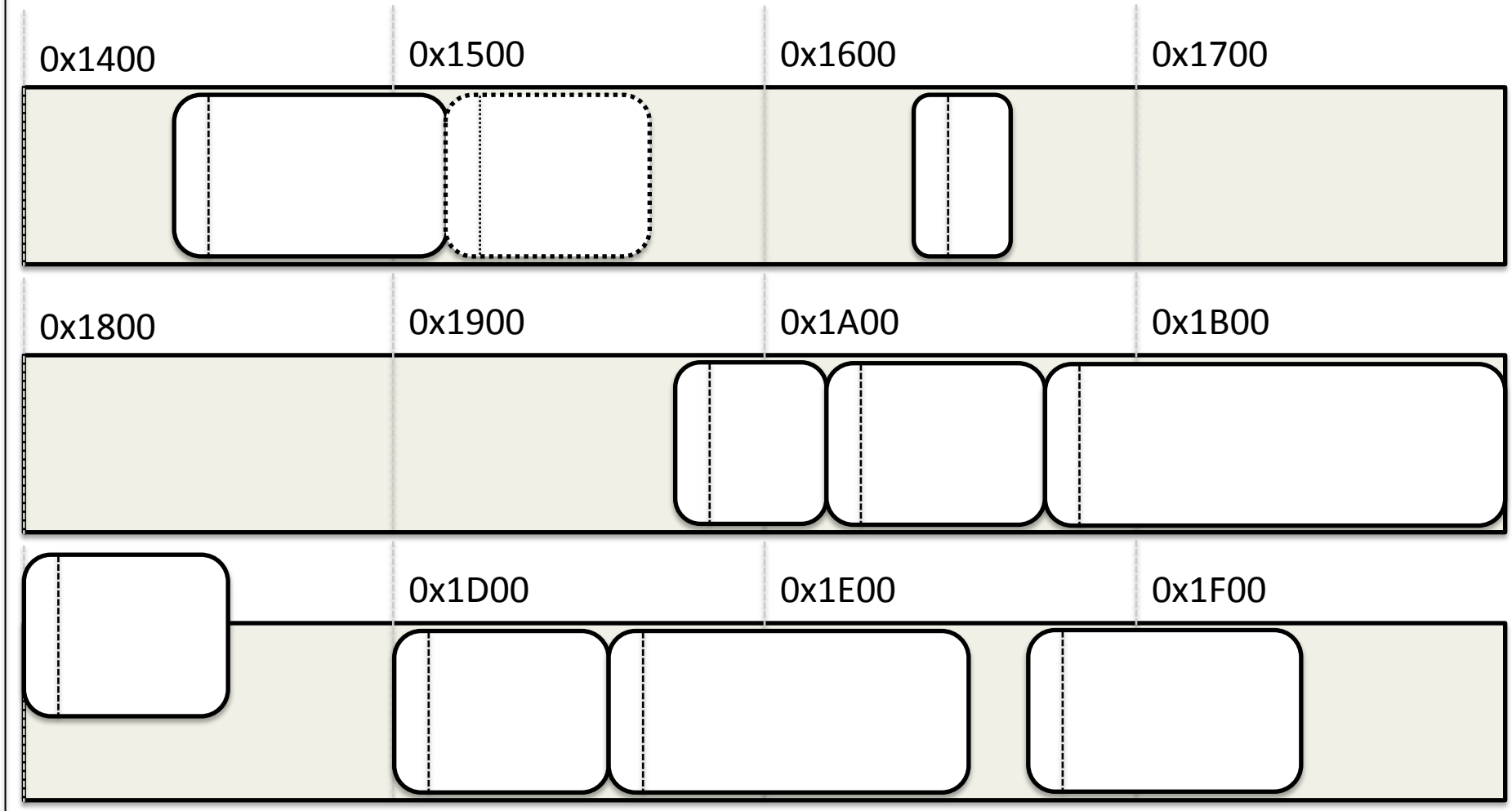

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

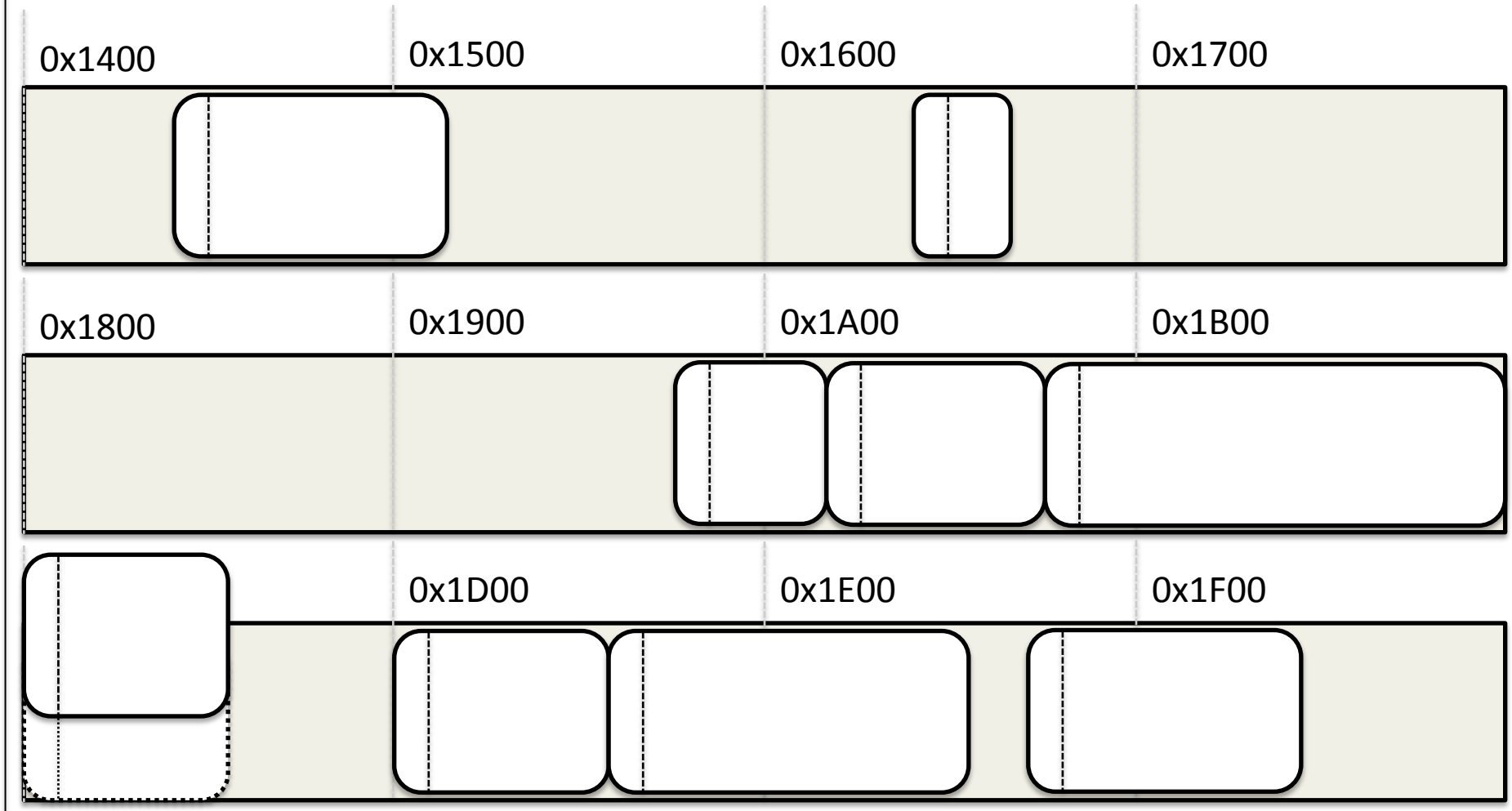

0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

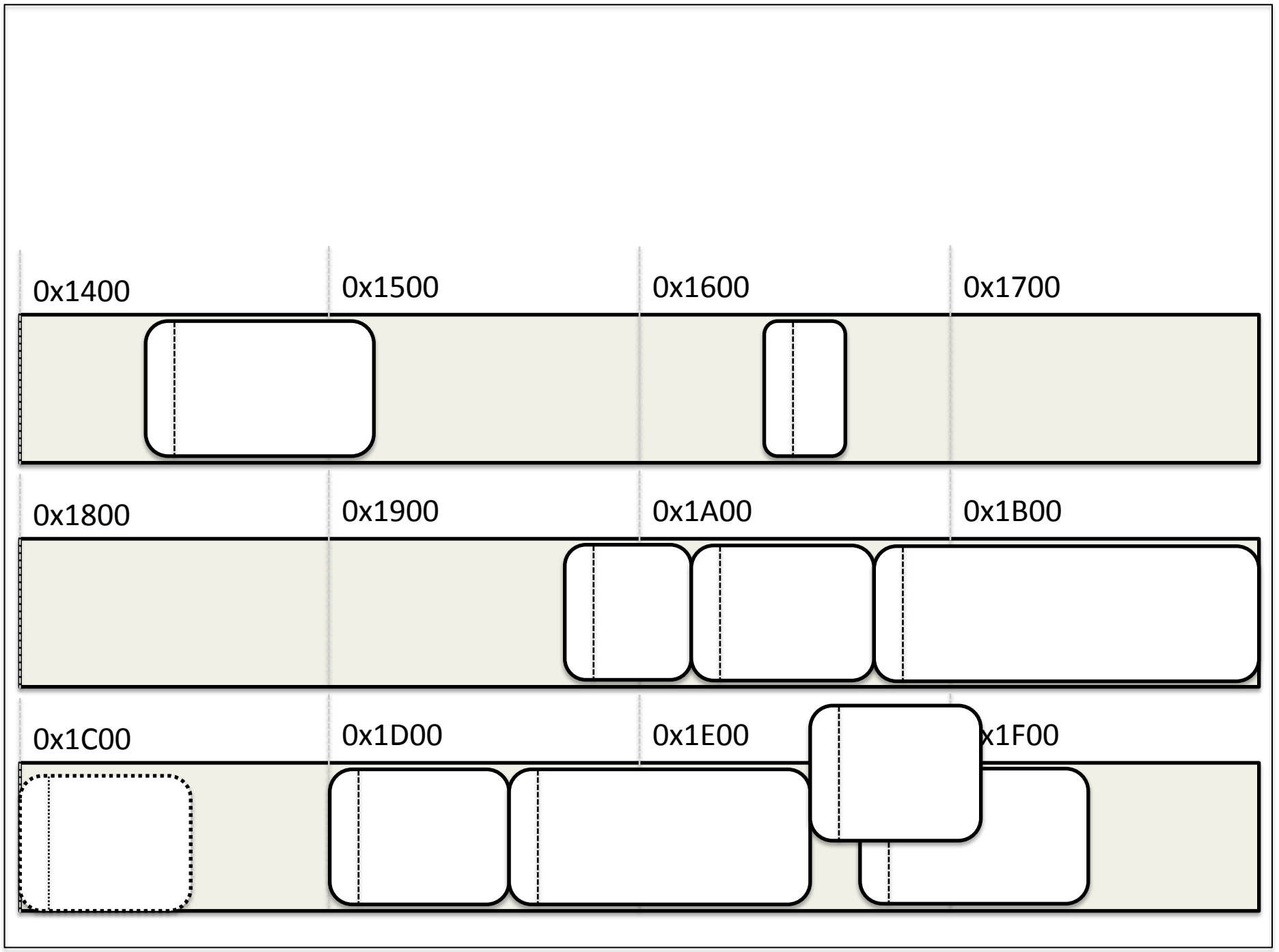

0x1400
0x
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

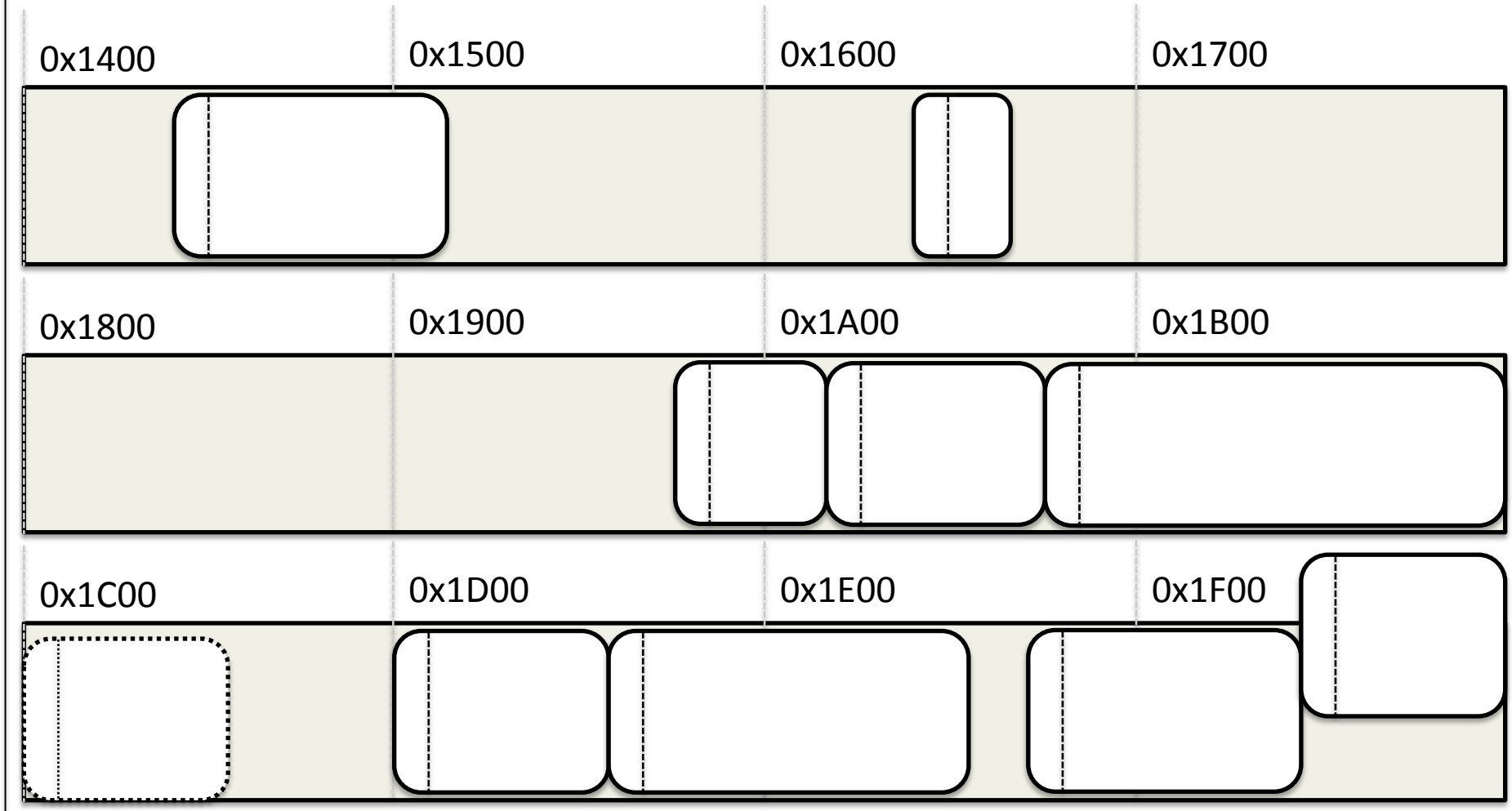

0x1400
0x
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

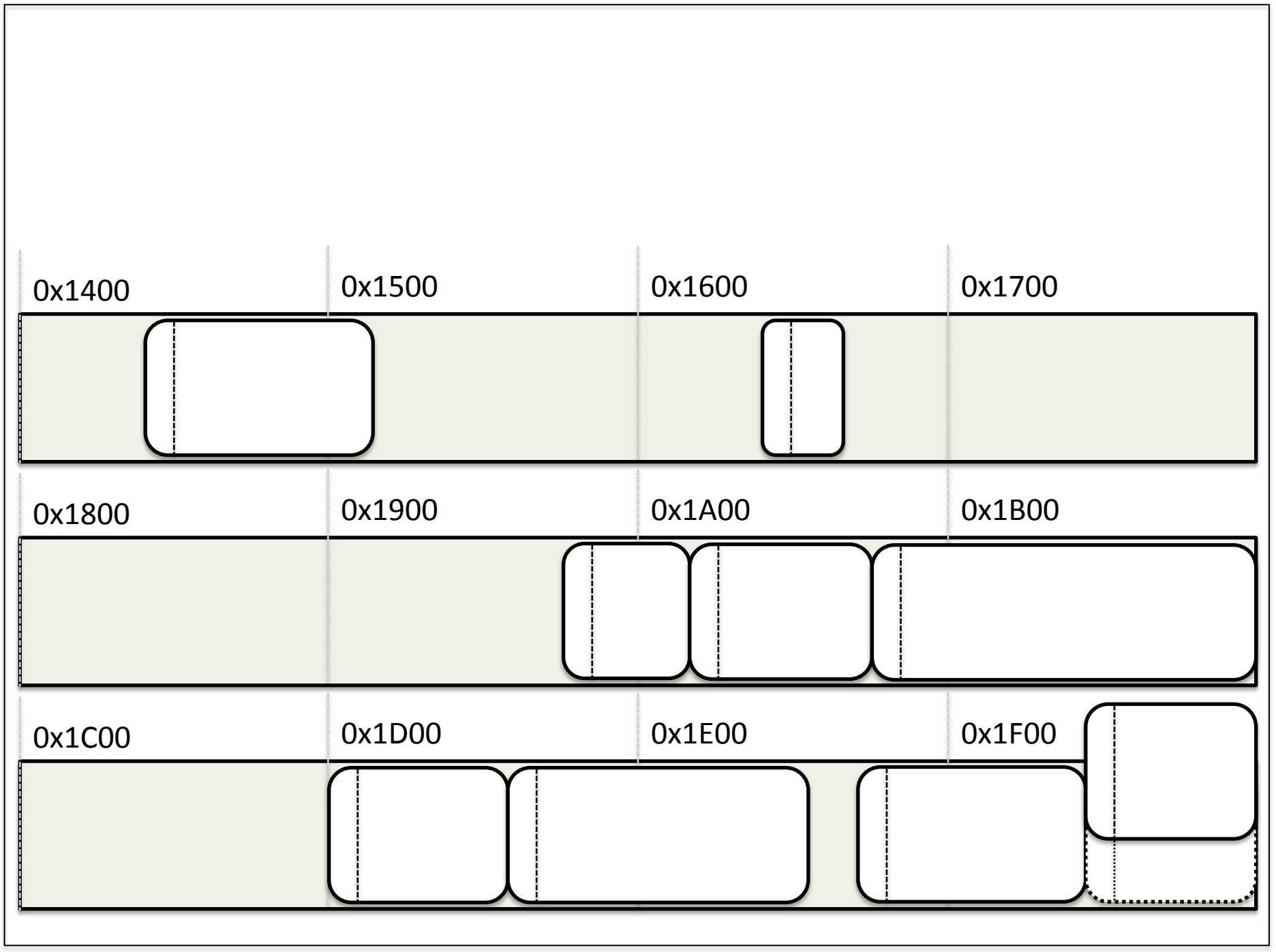

0x1400
0x1500
0x1600
700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

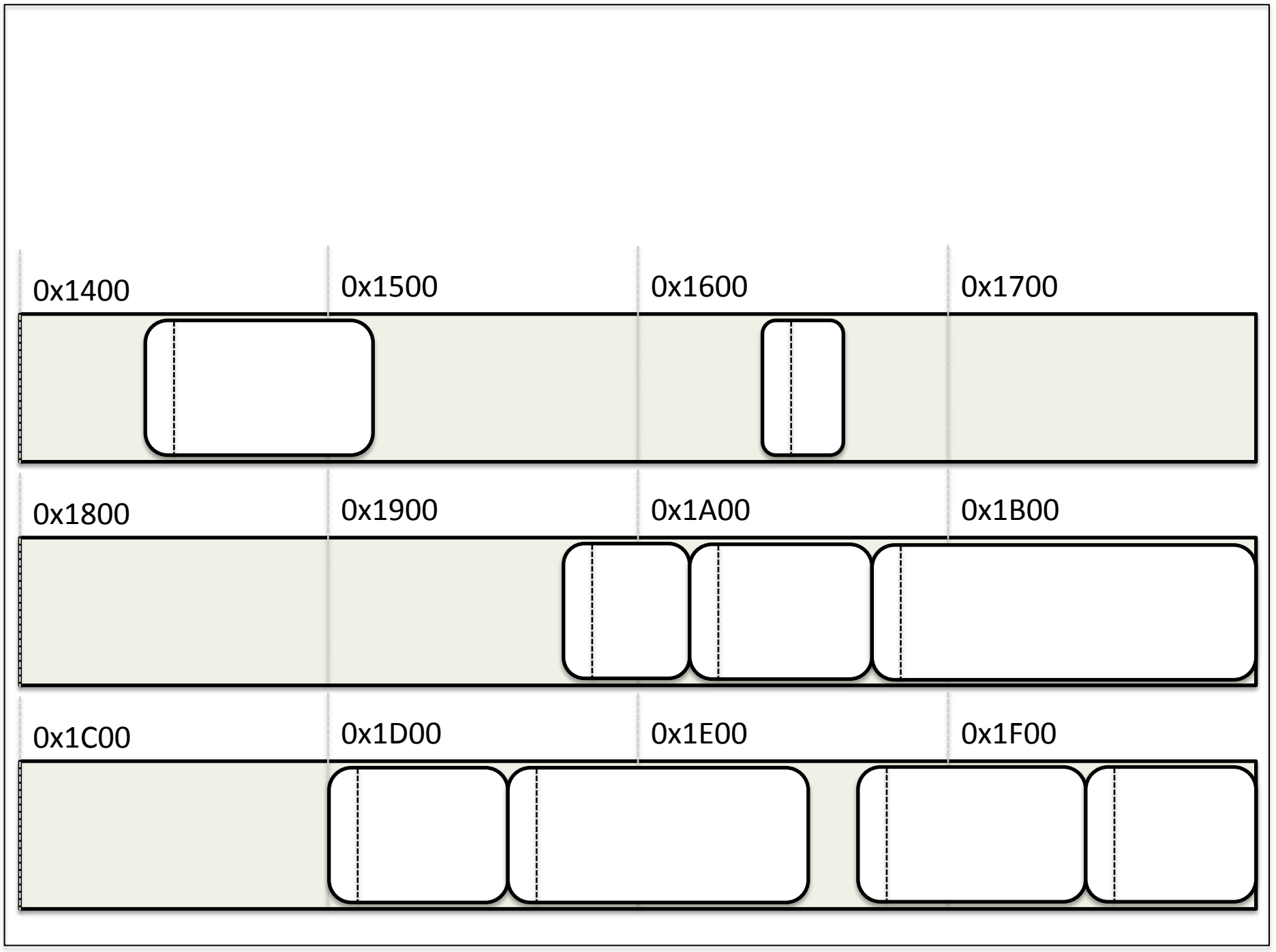

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1D00
0x1E00
0x1F00


0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1D00
0x1E00
0x1F00


0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
x1F00

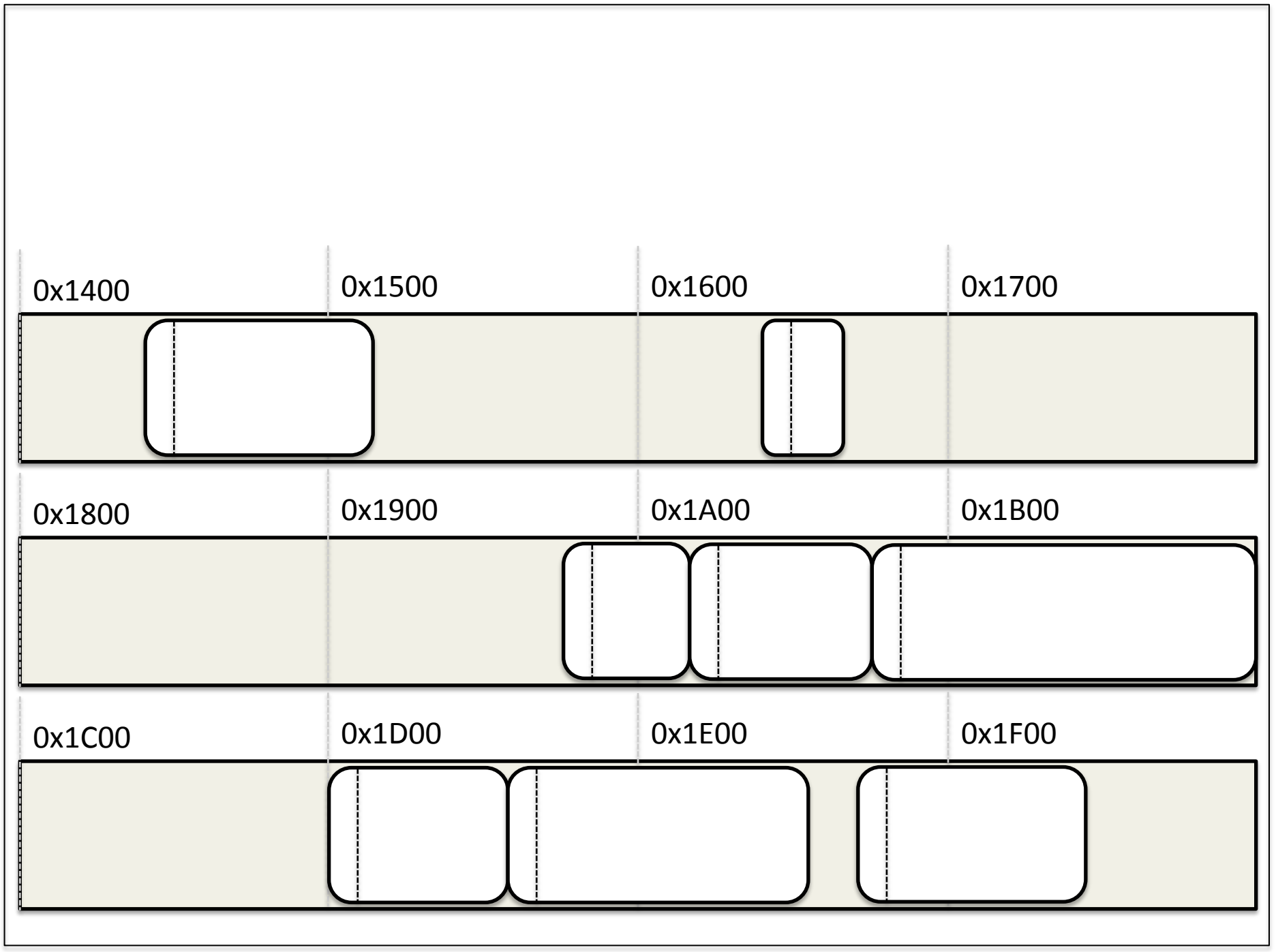

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

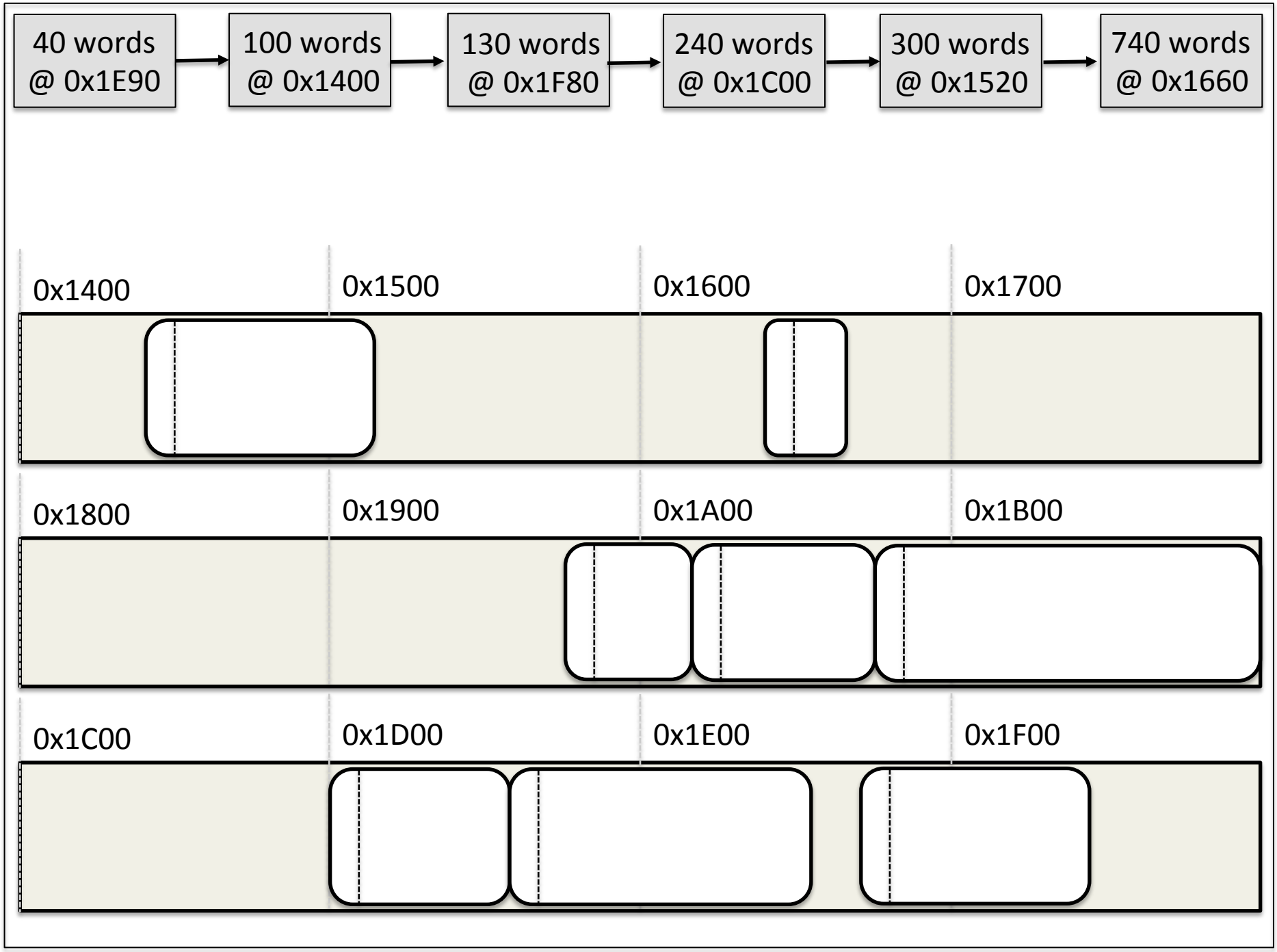

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

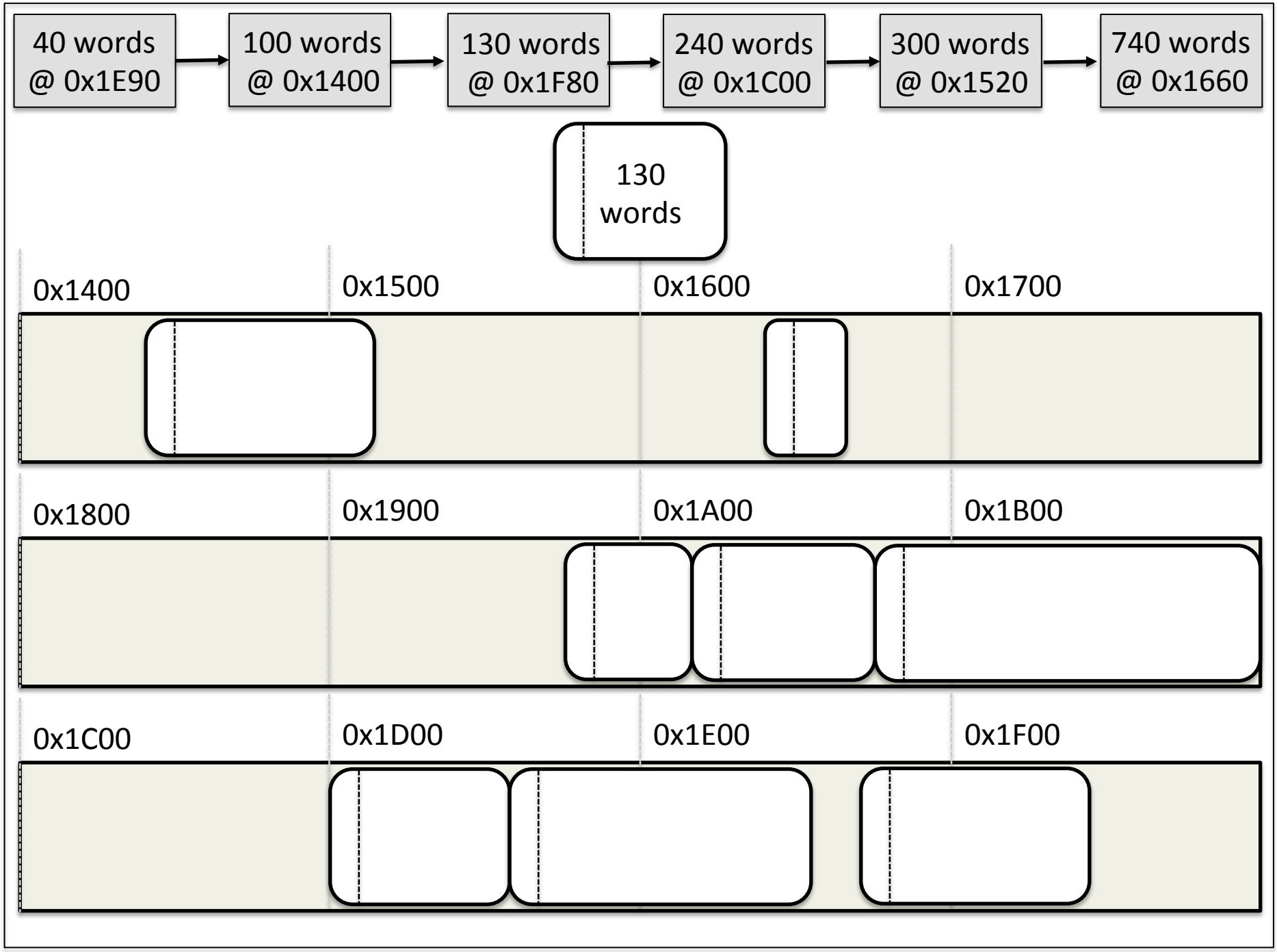

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

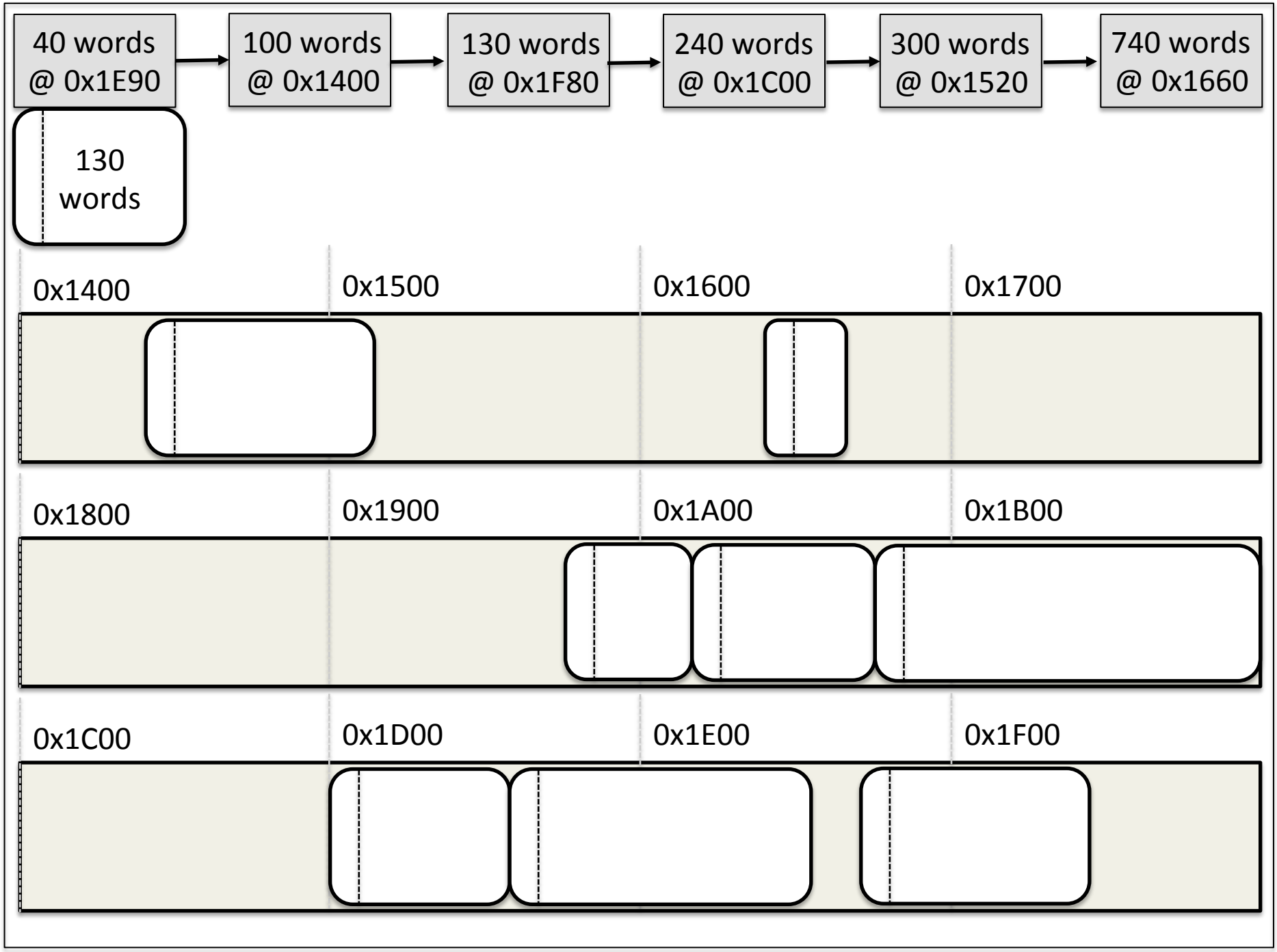
Allocation Strategies

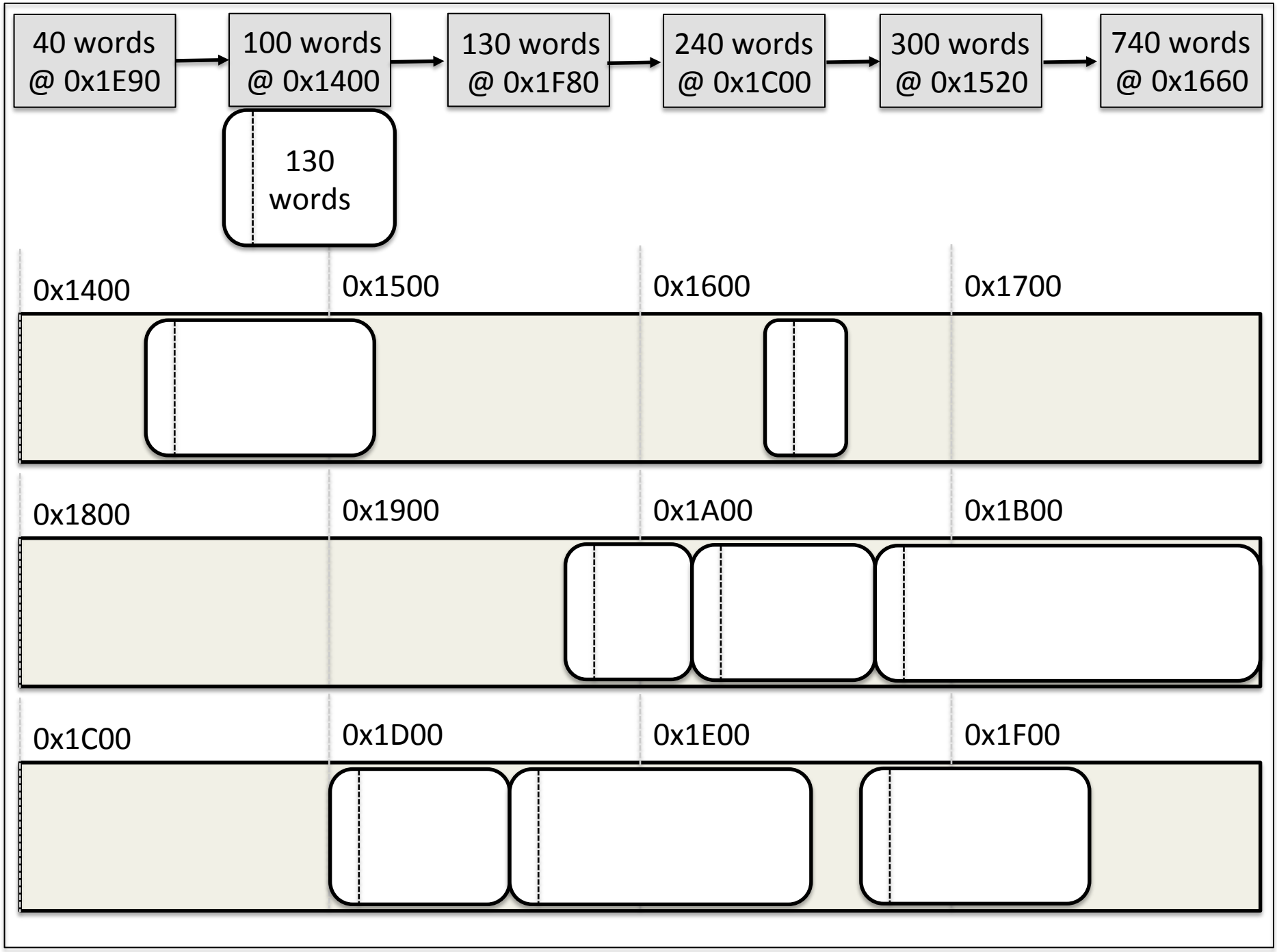
- First Fit
  - Put new object in the first hole that's large enough
- Best Fit
  - Scan the heap to find the hole that fits best
- Free List
  - Maintain a sorted list of all holes

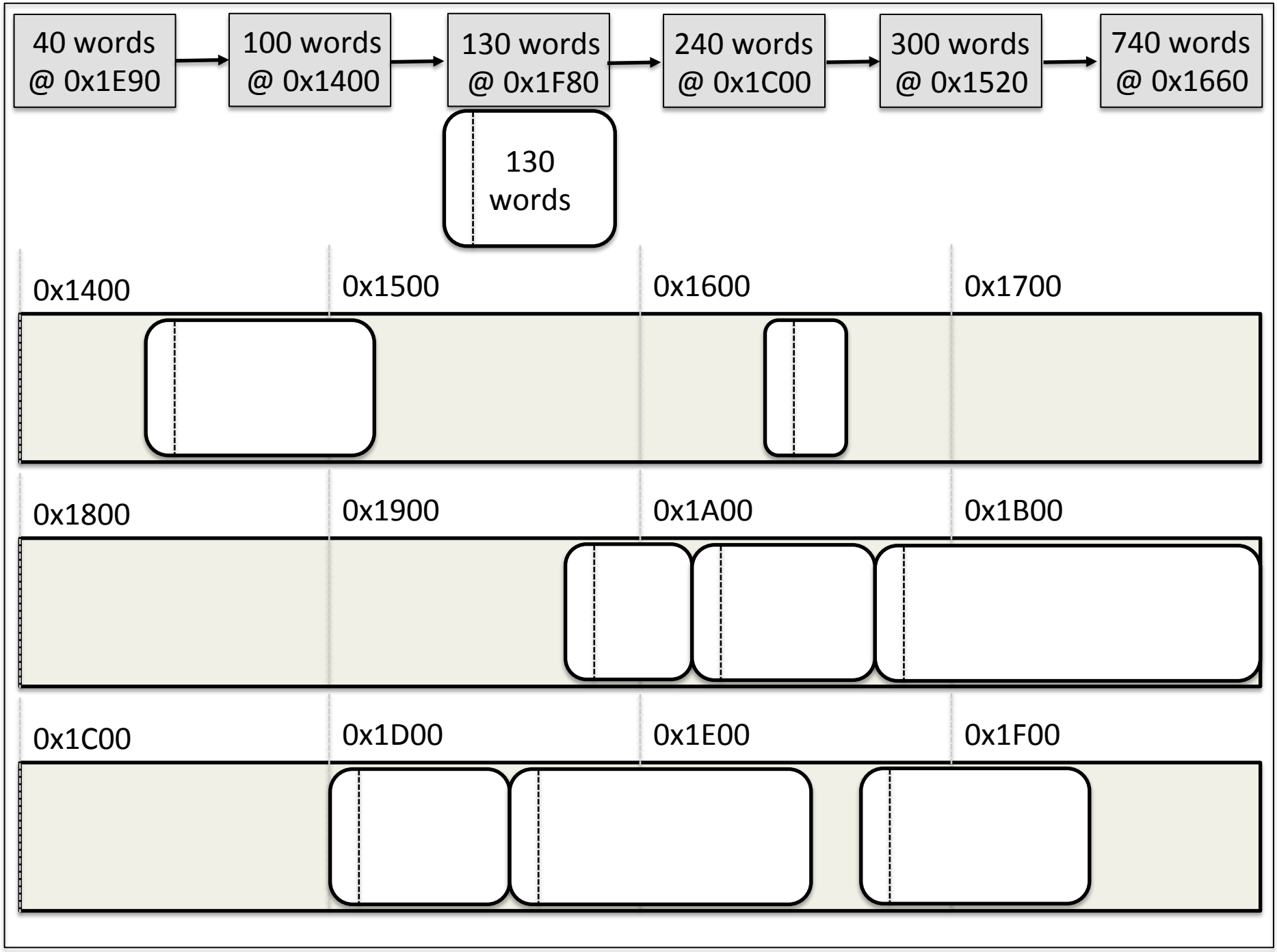

0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

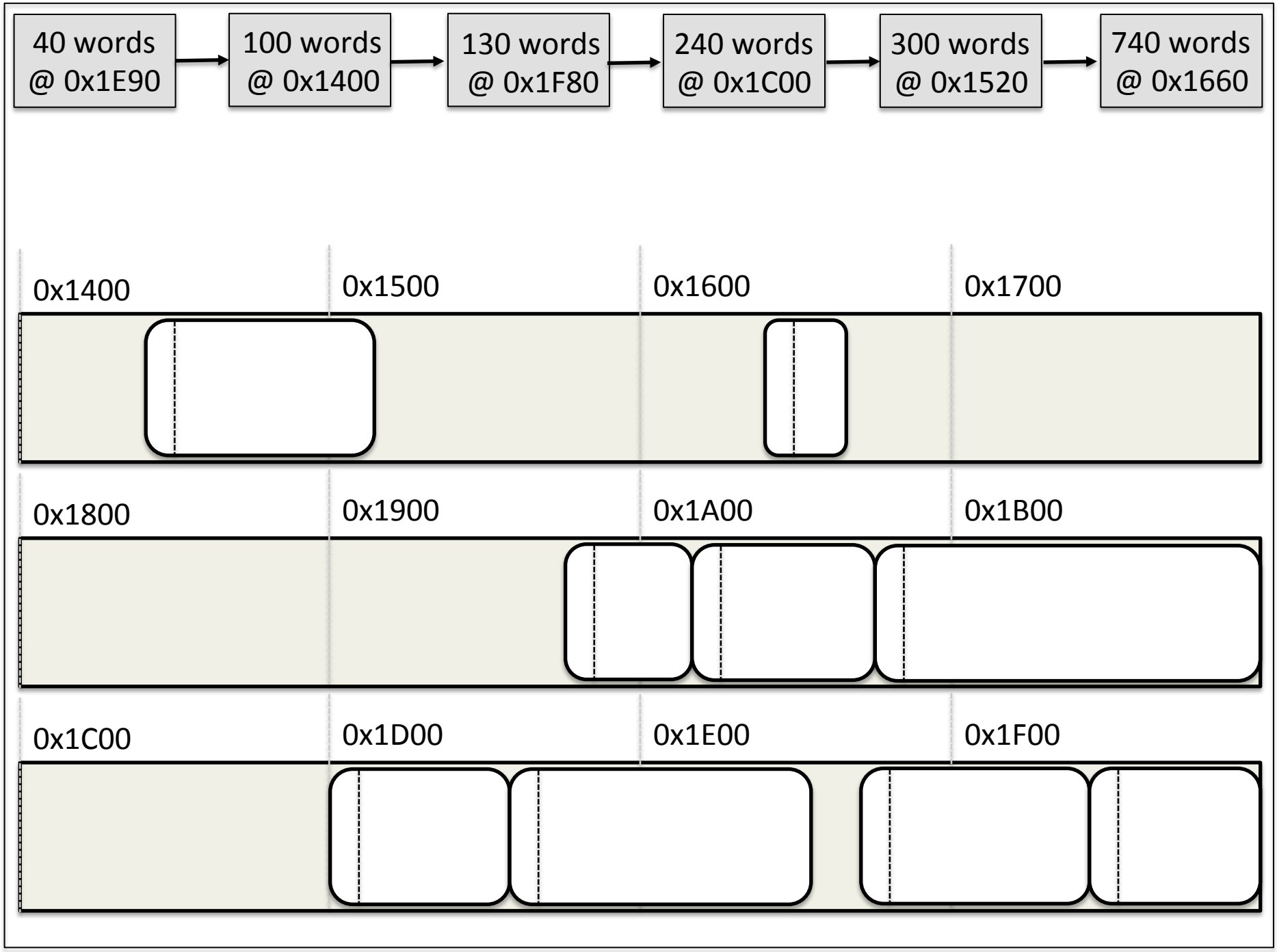

40 words
@ 0x1E90
100 words
@ 0x1400
130 words
@ 0x1F80
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

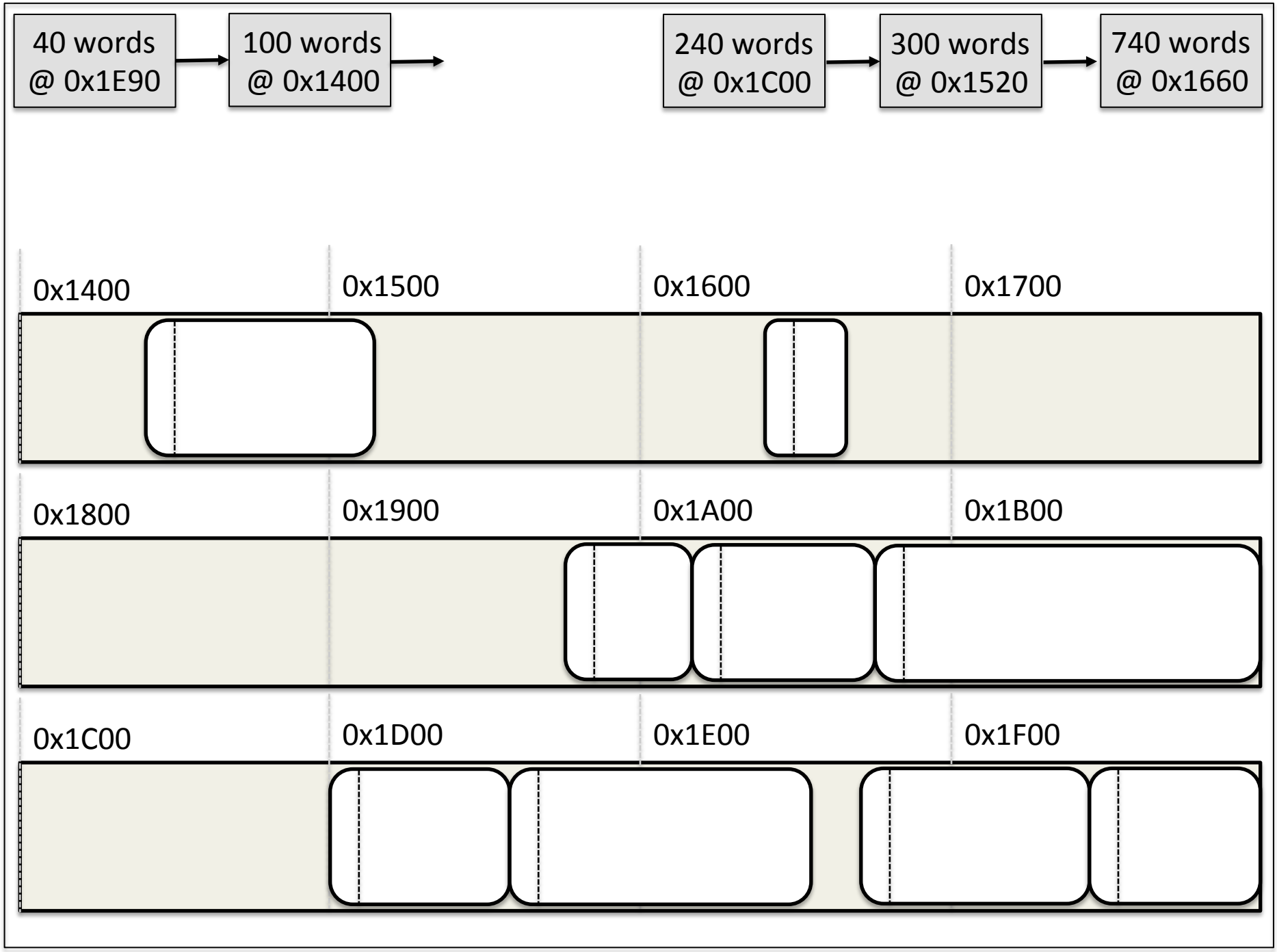

40 words @ 0x1E90
100 words @ 0x1400
130 words @ 0x1F80
240 words @ 0x1C00
300 words @ 0x1520
740 words @ 0x1660
130 words
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

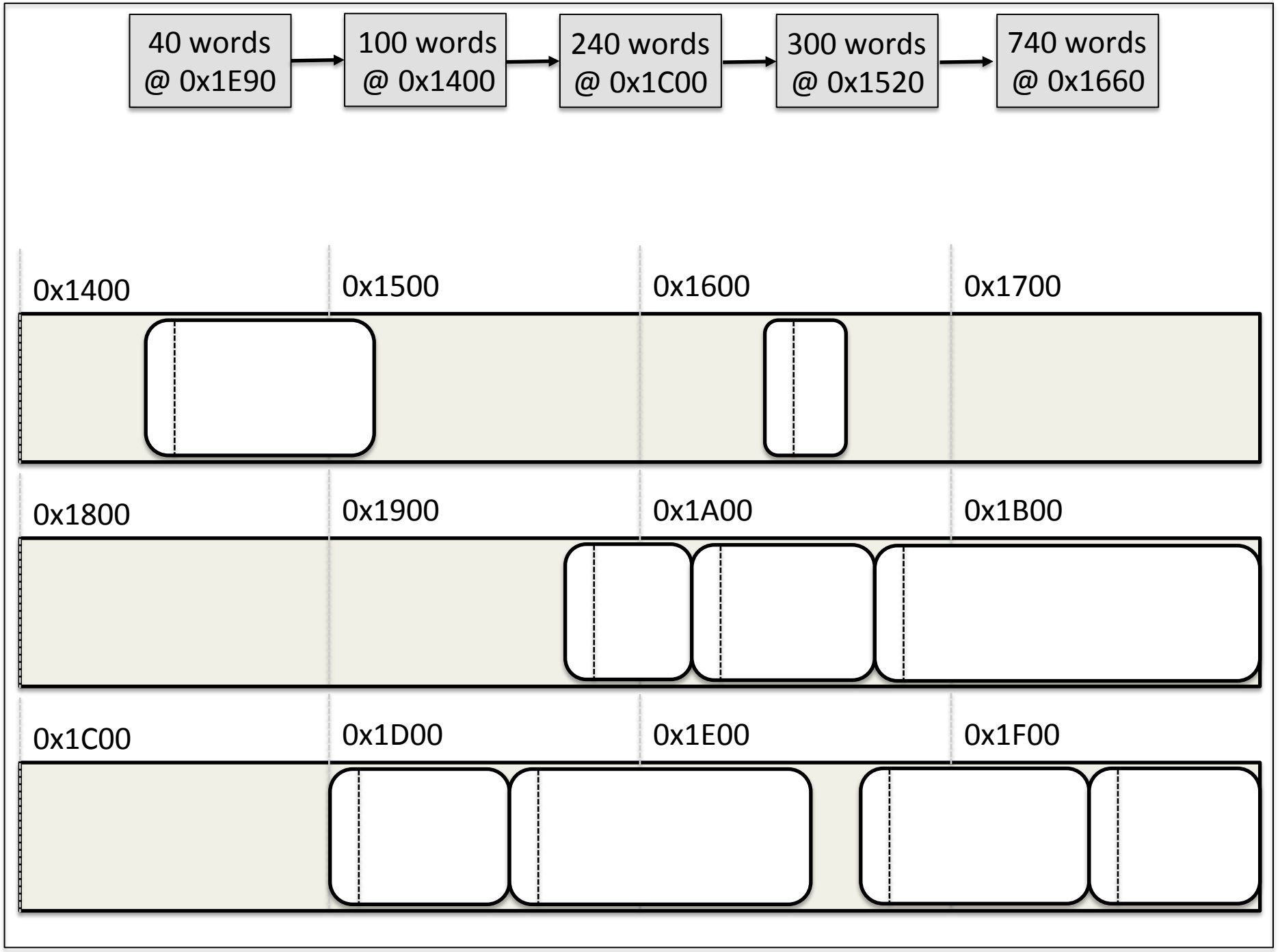

40 words
@ 0x1E90
100 words
@ 0x1400
130 words
@ 0x1F80
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
130
words
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


40 words
@ 0x1E90
100 words
@ 0x1400
130 words
@ 0x1F80
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
130
words
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00


40 words
@ 0x1E90
100 words
@ 0x1400
130 words
@ 0x1F80
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
130
words
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

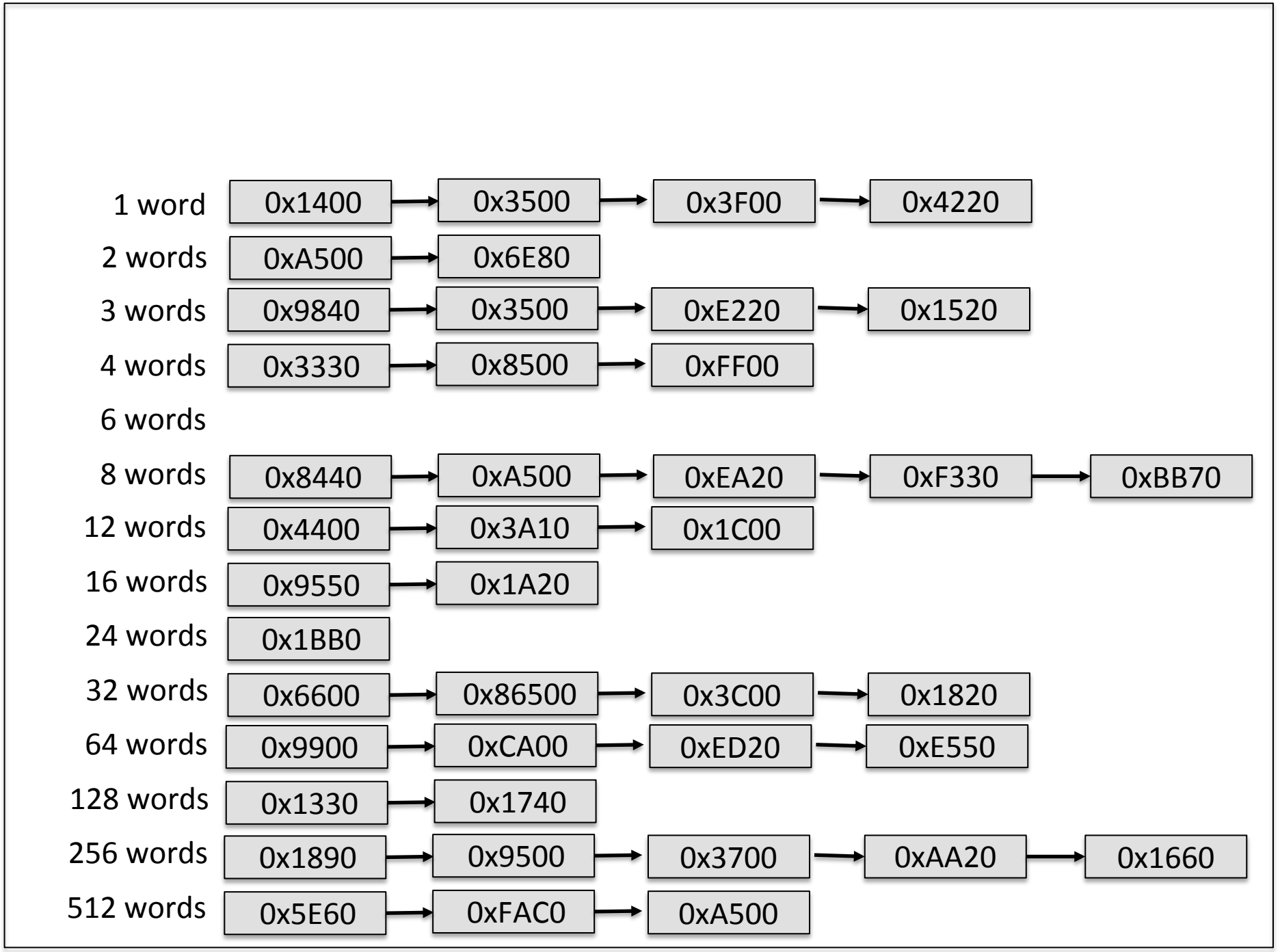

40 words
@ 0x1E90
100 words
@ 0x1400
130 words
@ 0x1F80
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

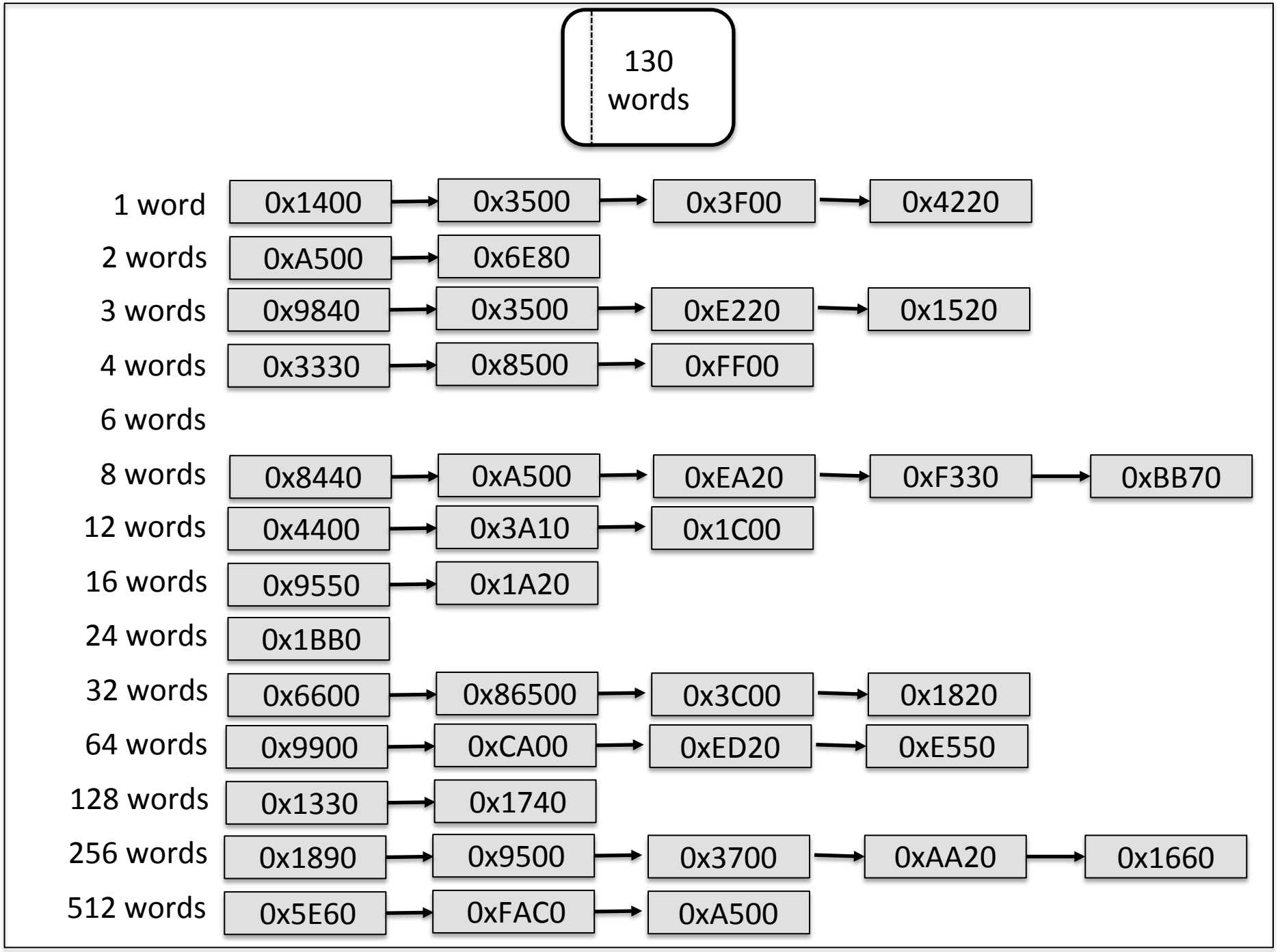

40 words
@ 0x1E90
100 words
@ 0x1400
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

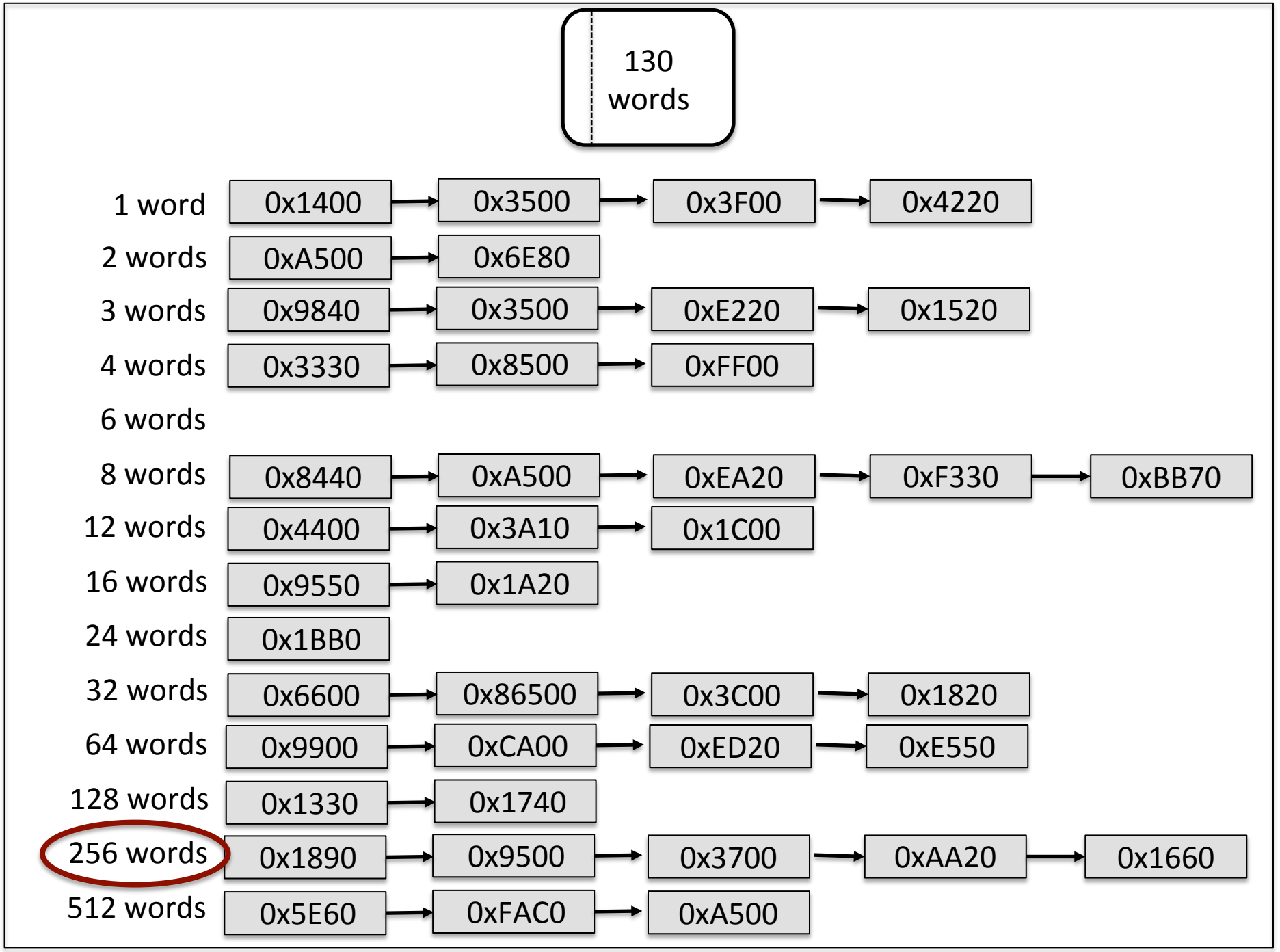

40 words
@ 0x1E90
100 words
@ 0x1400
240 words
@ 0x1C00
300 words
@ 0x1520
740 words
@ 0x1660
0x1400
0x1500
0x1600
0x1700
0x1800
0x1900
0x1A00
0x1B00
0x1C00
0x1D00
0x1E00
0x1F00

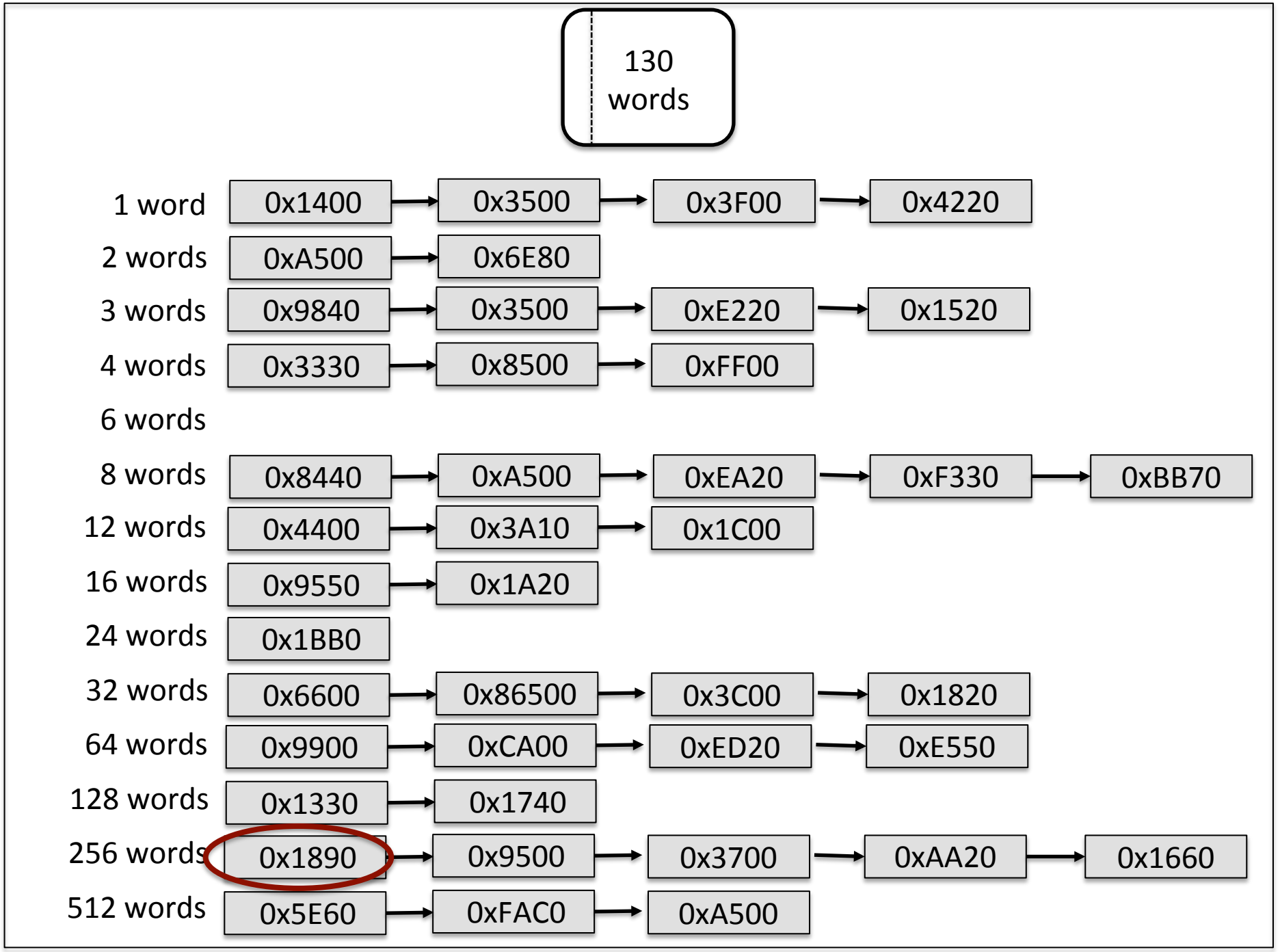
Allocation Strategies

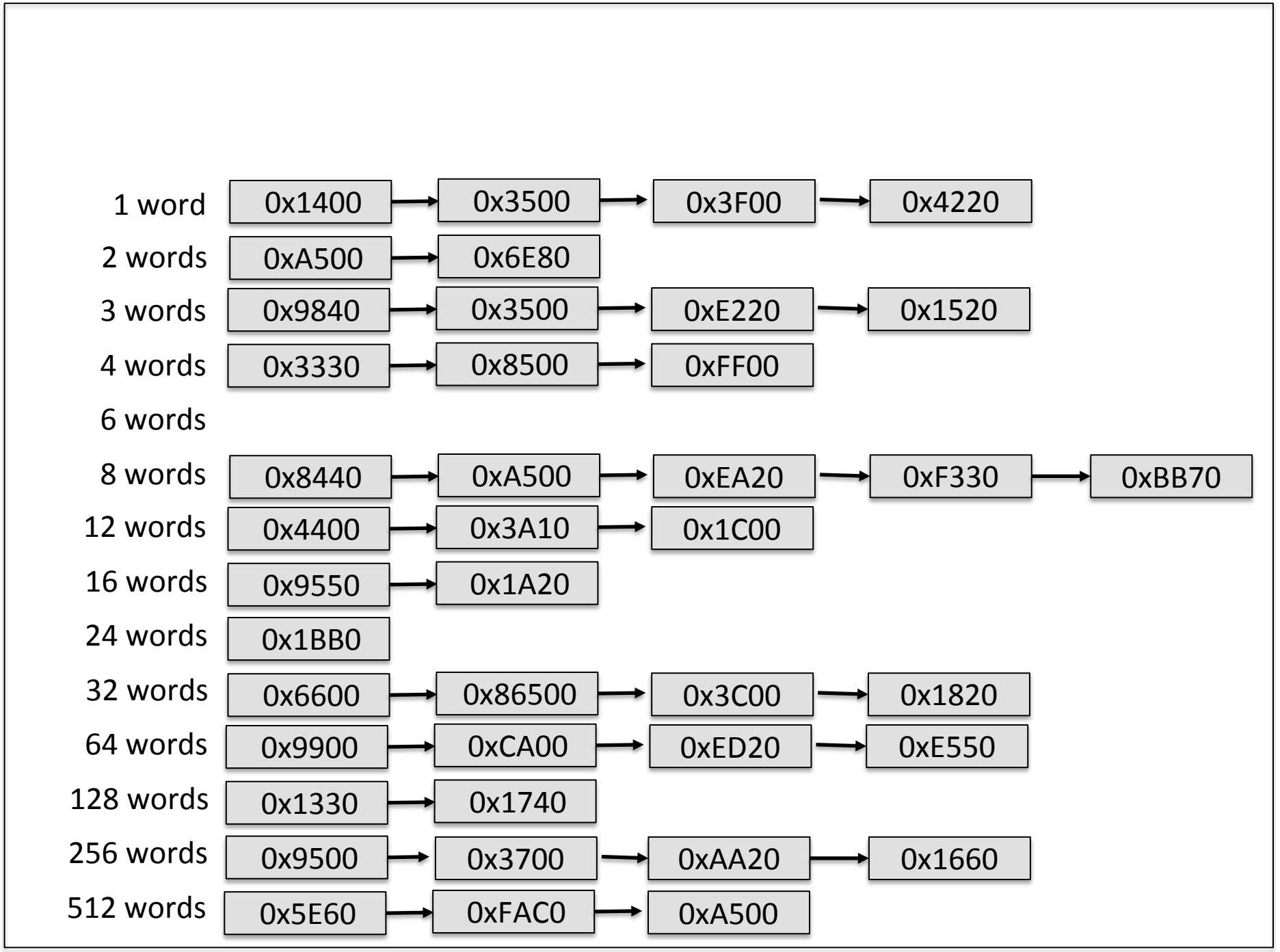
- First Fit
  - Put new object in the first hole that's large enough
- Best Fit
  - Scan the heap to find the hole that fits best
- Free List
  - Maintain a sorted list of all holes
- Segregated Free List
  - Maintain multiple free lists for different sized holes


1 word
0x1400
0x3500
0x3F00
0x4220
2 words
0xA500
0x6E80
3 words
0x9840
0x3500
0xE220
0x1520
4 words
0x3330
0x8500
0xFF00
6 words
8 words
0x8440
0xA500
0xEA20
0xF330
0xBB70
12 words
0x4400
0x3A10
0x1C00
16 words
0x9550
0x1A20
24 words
0x1BB0
32 words
0x6600
0x86500
0x3C00
0x1820
64 words
0x9900
0xCA00
0xED20
0xE550
128 words
0x1330
0x1740
256 words
0x1890
0x9500
0x3700
0xAA20
0x1660
512 words
0x5E60
0xFAC0
0xA500


130 words
1 word
0x1400
0x3500
0x3F00
0x4220
2 words
0xA500
0x6E80
3 words
0x9840
0x3500
0xE220
0x1520
4 words
0x3330
0x8500
0xFF00
6 words
8 words
0x8440
0xA500
0xEA20
0xF330
0xBB70
12 words
0x4400
0x3A10
0x1C00
16 words
0x9550
0x1A20
24 words
0x1BB0
32 words
0x6600
0x86500
0x3C00
0x1820
64 words
0x9900
0xCA00
0xED20
0xE550
128 words
0x1330
0x1740
256 words
0x1890
0x9500
0x3700
0xAA20
0x1660
512 words
0x5E60
0xFAC0
0xA500


130 words
1 word
0x1400
0x3500
0x3F00
0x4220
2 words
0xA500
0x6E80
3 words
0x9840
0x3500
0xE220
0x1520
4 words
0x3330
0x8500
0xFF00
6 words
8 words
0x8440
0xA500
0xEA20
0xF330
0xBB70
12 words
0x4400
0x3A10
0x1C00
16 words
0x9550
0x1A20
24 words
0x1BB0
32 words
0x6600
0x86500
0x3C00
0x1820
64 words
0x9900
0xCA00
0xED20
0xE550
128 words
0x1330
0x1740
256 words
0x1890
0x9500
0x3700
0xAA20
0x1660
512 words
0x5E60
0xFAC0
0xA500


130
words
1 word
0x1400
0x3500
0x3F00
0x4220
2 words
0xA500
0x6E80
3 words
0x9840
0x3500
0xE220
0x1520
4 words
0x3330
0x8500
0xFF00
6 words
8 words
0x8440
0xA500
0xEA20
0xF330
0xBB70
12 words
0x4400
0x3A10
0x1C00
16 words
0x9550
0x1A20
24 words
0x1BB0
32 words
0x6600
0x86500
0x3C00
0x1820
64 words
0x9900
0xCA00
0xED20
0xE550
128 words
0x1330
0x1740
256 words
0x1890
0x9500
0x3700
0xAA20
0x1660
512 words
0x5E60
0xFAC0
0xA500


1 word
0x1400
0x3500
0x3F00
0x4220
2 words
0xA500
0x6E80
3 words
0x9840
0x3500
0xE220
0x1520
4 words
0x3330
0x8500
0xFF00
6 words
8 words
0x8440
0xA500
0xEA20
0xF330
0xBB70
12 words
0x4400
0x3A10
0x1C00
16 words
0x9550
0x1A20
24 words
0x1BB0
32 words
0x6600
0x86500
0x3C00
0x1820
64 words
0x9900
0xCA00
0xED20
0xE550
128 words
0x1330
0x1740
256 words
0x9500
0x3700
0xAA20
0x1660
512 words
0x5E60
0xFAC0
0xA500

Avoiding Fragmentation

- Segregated free lists help
  - Approximates best fit across the whole heap
  - Introduce some of wasted space
- Fragmentation a danger when objects are deleted
- Need to compact the heap to eliminate it