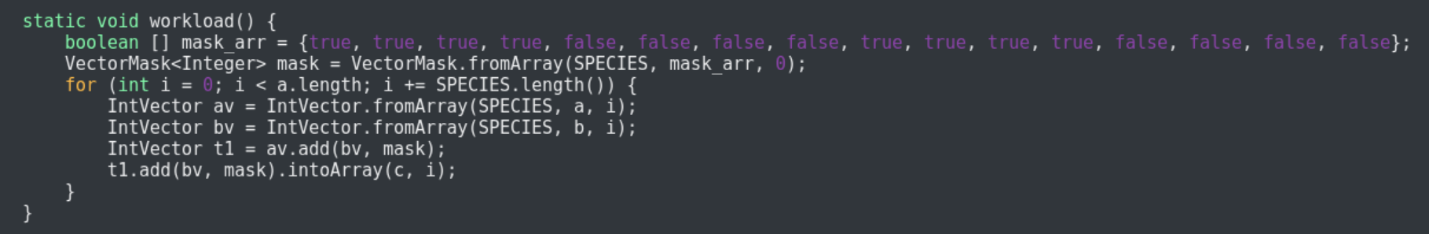
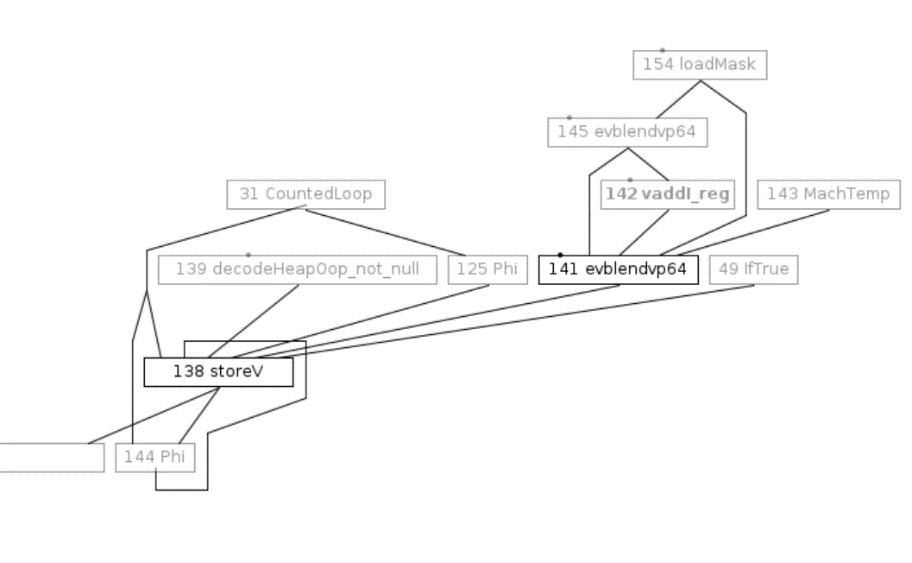
**AVX-512 masked operations optimization and register allocation support for opmask registers**

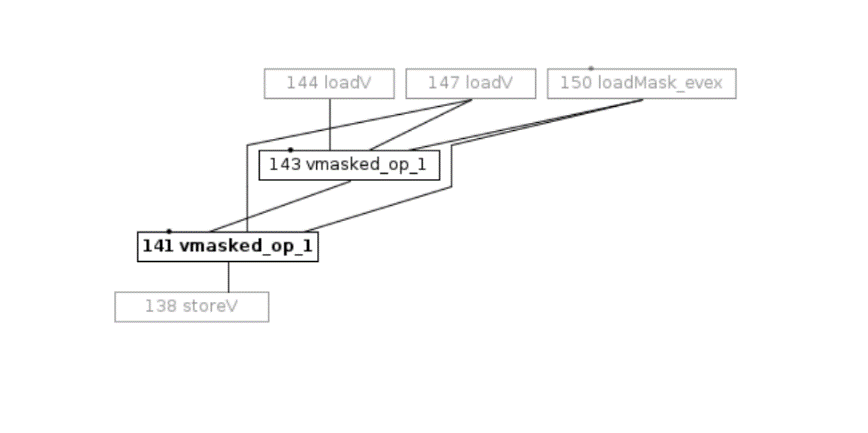
Java Source:



Post matcher original Graph:



Post matcher new Graph:



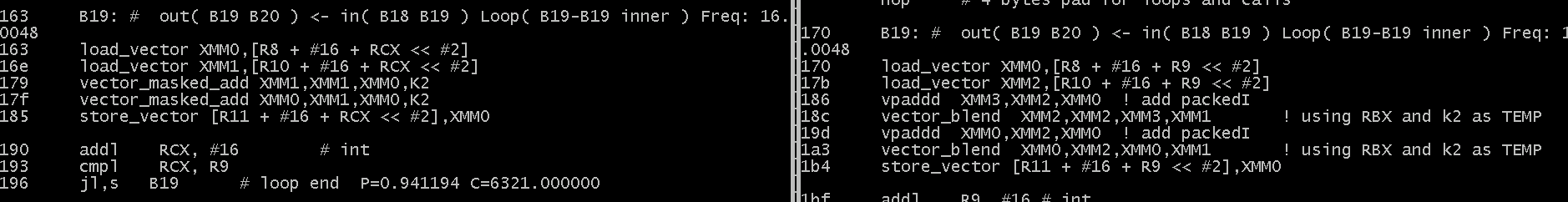
**RA Support (PoC):**

* New register-definitions and register class for opmask registers.
* No new masked type has been introduced, decoupling b/w ideal Type and Ideal\_reg has been exploited.
* ADLC fix for emitting correct predication condition for wider instruction patterns.
* Spilling code extensions.

**Masked operation optimization PoC results:**

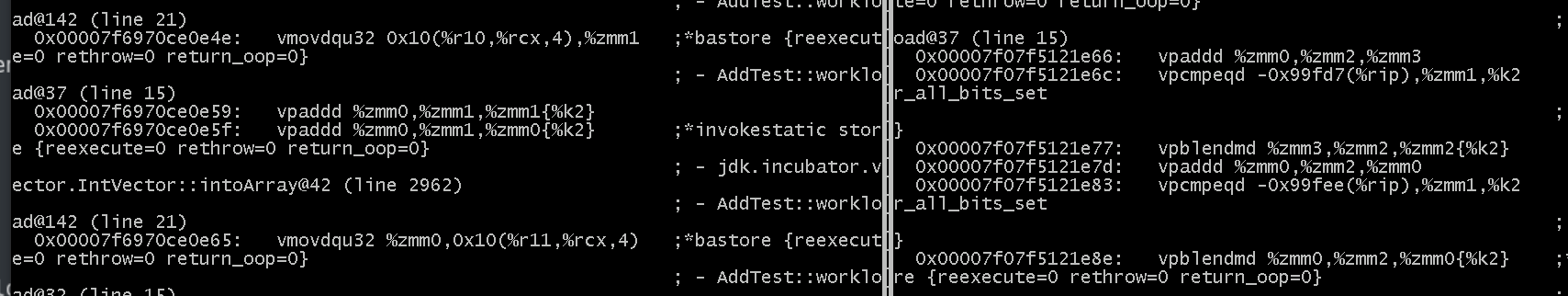
OptoAssembly sequence: Before and After support.

New Original



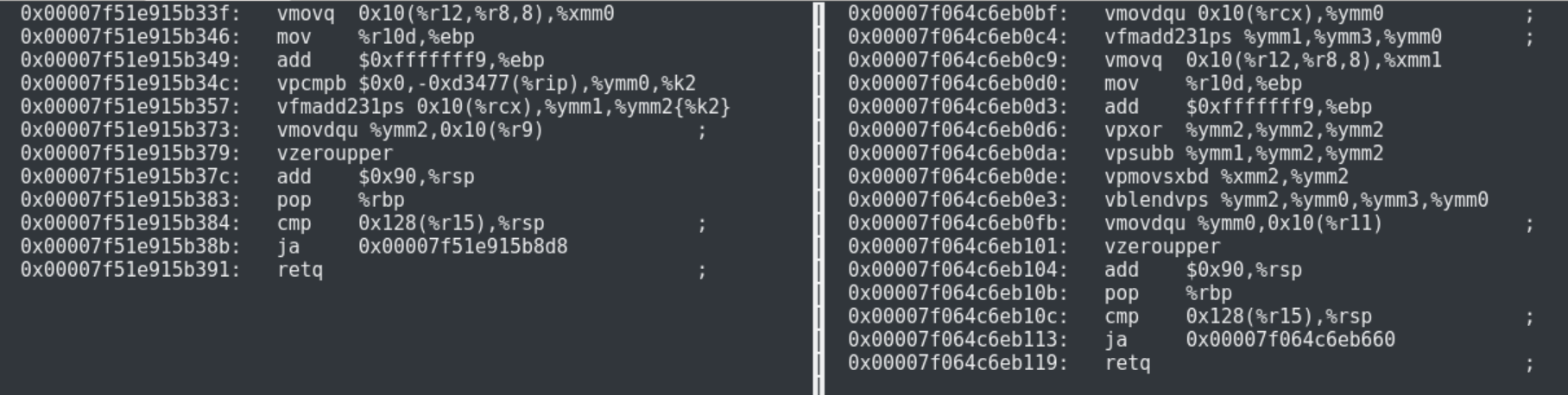
JIT Assembly sequence: Saves extra vector compare and blend per masked operation.

New Original

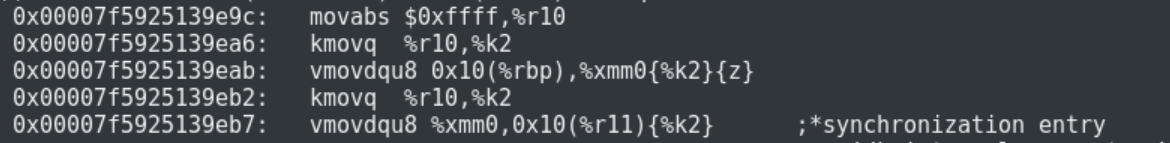


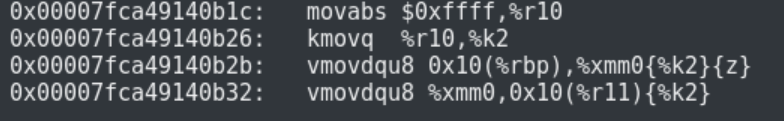
Masking optimization output for Ternary Operation (FMA)

New JIT’ed snippet Original JIT’ed snippet



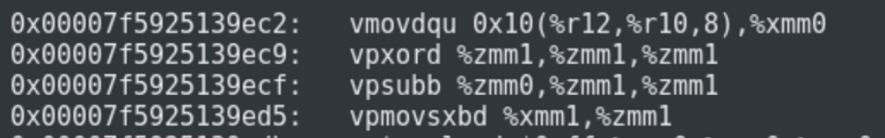
Partial In-lining array copy sequence (elimination of kmovq due to new mask operands):





**Mask loading API VectorMask.fromArray() instruction sequence latency and size:**

**Original:**



Ec9 -> VPXOR Zeroing idiom. (not issued to execution ports).

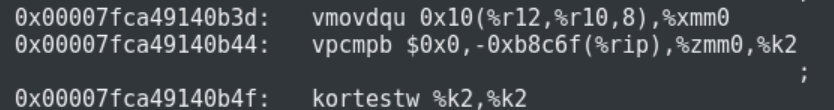
Ecf -> VPSUB 1 cycle

Ed5 -> VPMOVSXBD 3 cycle one port.

Latency : 4 cycles

Size : 18 bytes

**New:**



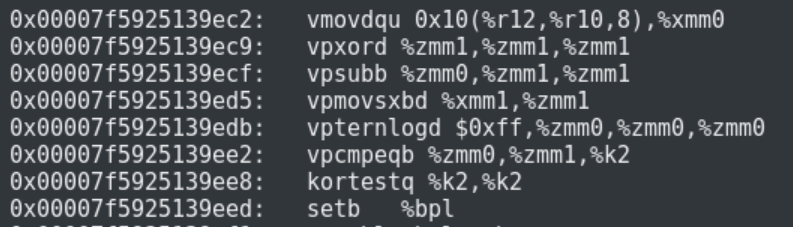
B44 -> VPCMP 3 cycles one port.

Latency : 3 cycles

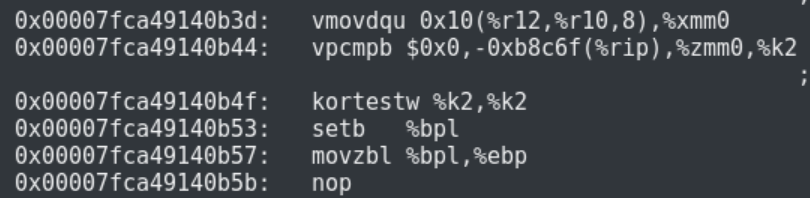
Size : 11 bytes

**VectorAPI masking operations:**

1. ***VectorMask.allTrue() original sequence:***

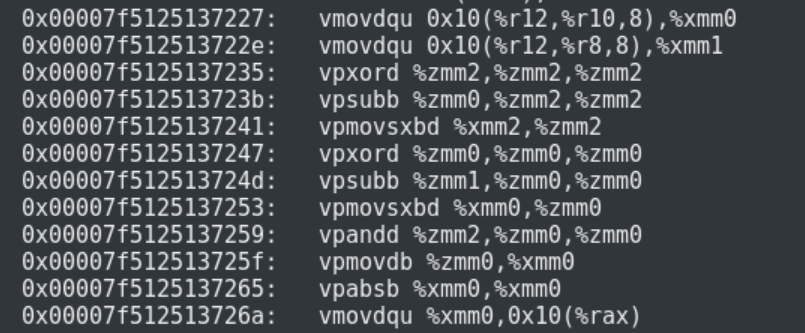


Code Size: 20 bytes

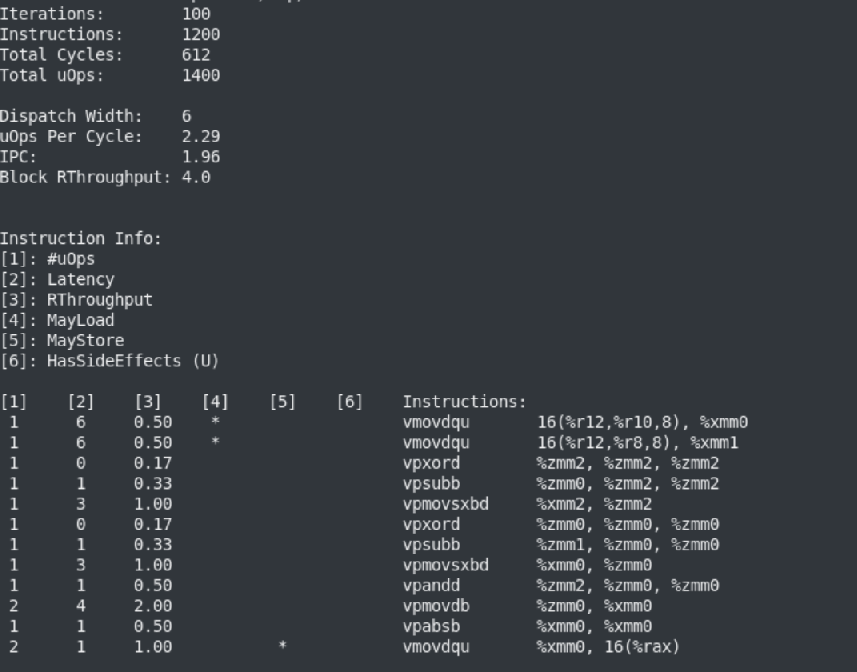


Size : 12 bytes

***VectorMask jdk.incubator.vector.VectorMask.and(VectorMask , VectorMask)***

****

Micro-arch analysis (SKX)

****

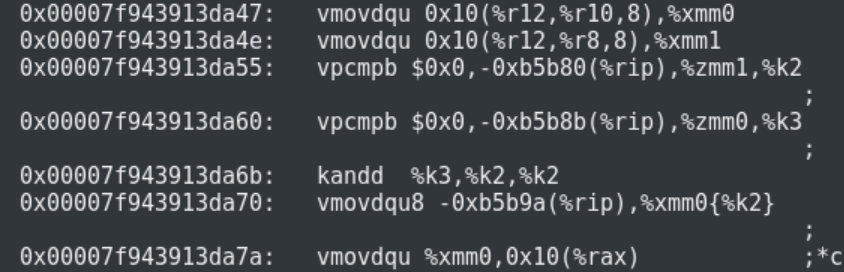
**New Sequence:**

**Mask1 = VectorLoadMask (LoadVector SRC3\_MEM)**

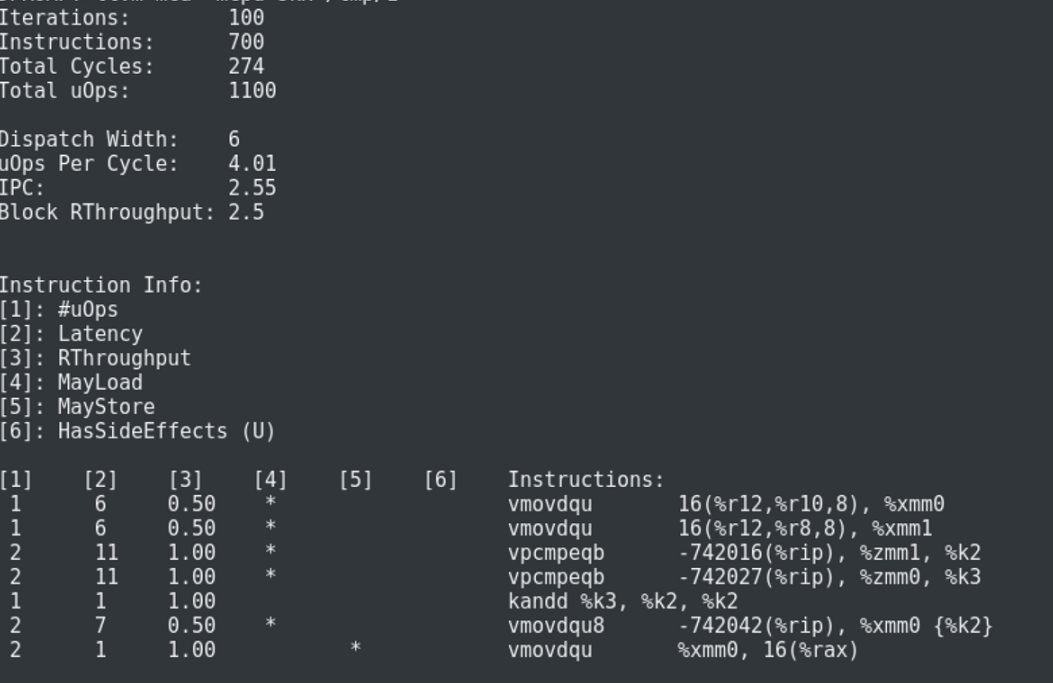
**Mask2 = VectorLoadMask (LoadVector SRC2\_MEM)**

**Mask3 = AndV Mask1 Mask2**

**StroreVector (VectorStoreMask Mask3 XMM) RES\_MEM**

****

Micro-arch analysis (SKX)



Following mask Generating Nodes directly produce an opmask operand:

VectorLoadMask

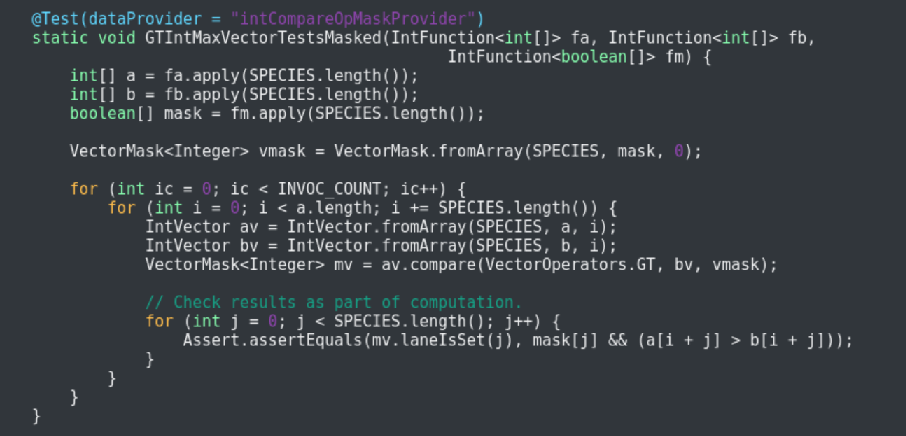
VectorMaskCmp

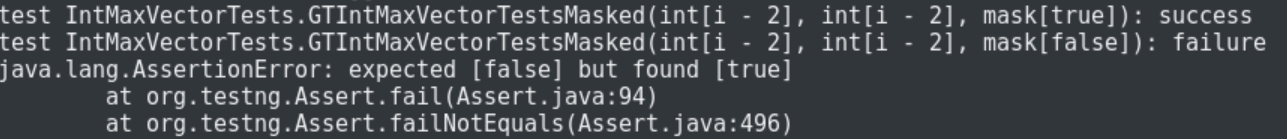
VectorMaskGen

Other mask generating nodes (mainly maskAll) must be followed by VectorStoreMask + VectorLoadMask

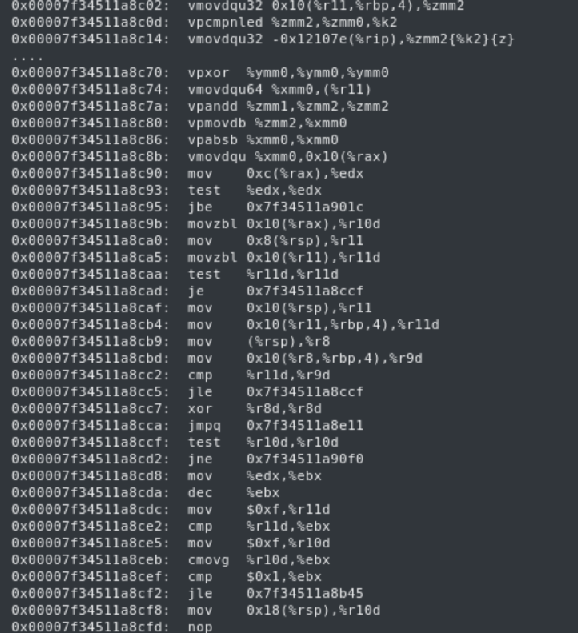
since in EVEX mode mask propagation should be though opmask register. This is already being done during Idealization of VectorUnbox node.

JTREG test (IntMaxVectorTests.java)

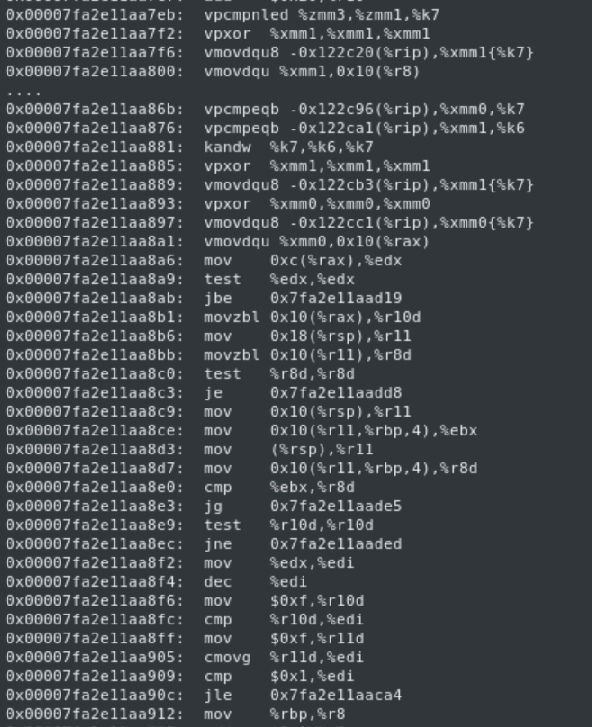




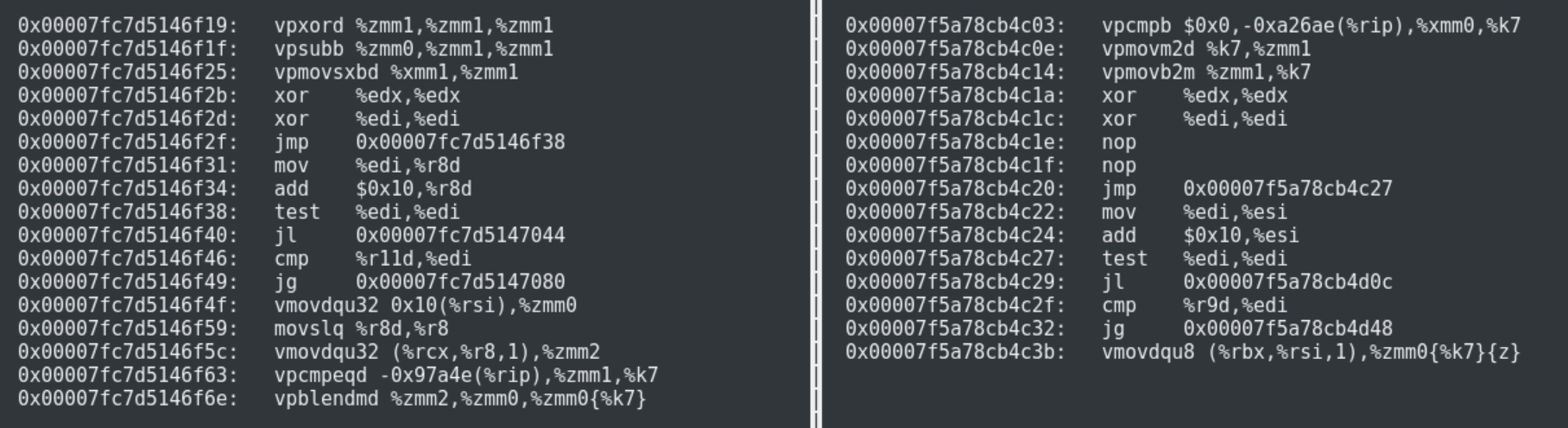
Earlier JIT sequence :



New JIT Sequence:



Masked Memory Operation (Load/Store) from/into [ByteArray/ByteBuffer]



Mask Cast Optimization AVX-512:

