The Streaming Hybrid Architecture Vector Engine (SHAVE) processor micro-architecture is an eight-slot very long instruction word processor capable of performing multiple 128-bit vector operations in parallel with multiple load/store, scalar ﬂoating-point, integer, and control-ﬂow operations in a single clock cycle.

1. Shave eight functional units:
   1. Predicated execution unit (PEU)
   2. Branch and repeat unit (BRU)
   3. Two 64-bit load-store units (LSU0 and LSU1)
   4. 128-bit vector arithmetic unit (VAU)
   5. 32-bit vector/scalar arithmetic unit (SAU)
   6. 32-bit integer arithmetic unit (IAU)
   7. 128-bit compare move unit (CMU)
2. 32 entry 128-bit vector register file (VRF)
3. 32 entry 32-bit vector/scalar register file (IRF)
4. SIMD arithmetic support
5. Trigonometric function support
6. Debug and control unit (DCU)
7. 2kB two-way instruction cache
8. 1kB data cache
9. Low latency access to CMX memory
10. Translation lookaside buﬀer (TLB)

**函数传参**

**Registers:**1) *Stack pointer register* – IRF register I19 is reserved as a dedicated stack pointer and should never be  
modified except to allocate and free space for stack variables.  
2) *Link Register* – IRF register I30 is used both to store the address to which control flow should jump when a function is called, and the return address to which control flow should jump after a function has  
completed execution.  
3) *Preserved IRF registers* (i20,...,i29,i31) and *VRF registers* (v24,...,v31)

Up to 8 scalar parameters may be passed via the IRF, using registers i11, i12, ..., i18. Register i18 is also used to store the return value for a method call.

Up to 8 vector parameters may be passed via the VRF, using registers v16, v17, ..., v23. Register v23 is also used to store the return value for a method call if the return value is one of the above types.

循环：

You may use BRU.RPL (a variable number of times) or BRU.RPS (a constant number of times) to create a loop. Or you may use different jump instructions (BRU.BRA , BRU.JMP) with a condition or counter to  
create a conditional loop.

分支Branching:

cmu.cmii.i32 i0,i1  
peu.pc1c EQ || bru.bra L\_branch1  
peu.pc1c NEQ || bru.bra L\_branch2

指令循环执行：

CNN模块

Supports multichannel convolution to a depth of 2048 channel.

Can process all GoogLeNet stages up to Sofmax entirely in Hardware

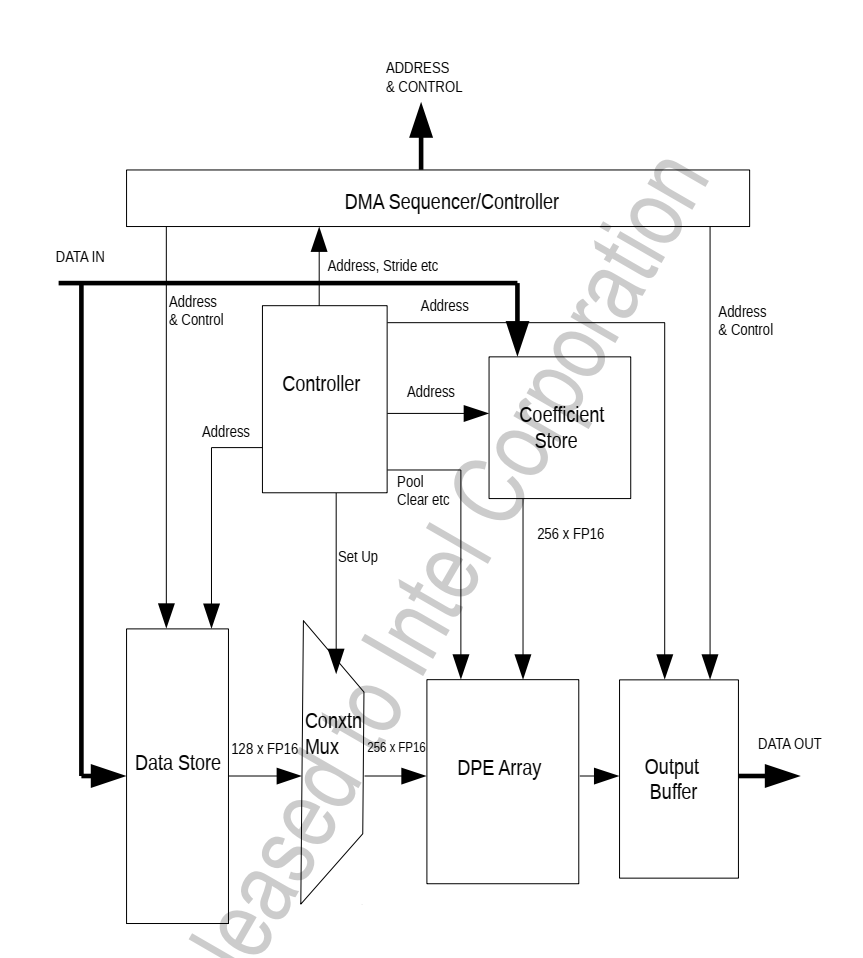
Convolution Kernel size from 1X1 up to 15X15

Convolution/Pooling Stride: 1-8

Compute throughput of 256 MACs/CNN Block/Cycle

Accumulate Precision is hybrid fixed point 24-bit

FP16 precision for input, output data and coefficients



There are 256 DPEs in total. All can perform Convolution, Pooling and PreLU depending on setup

