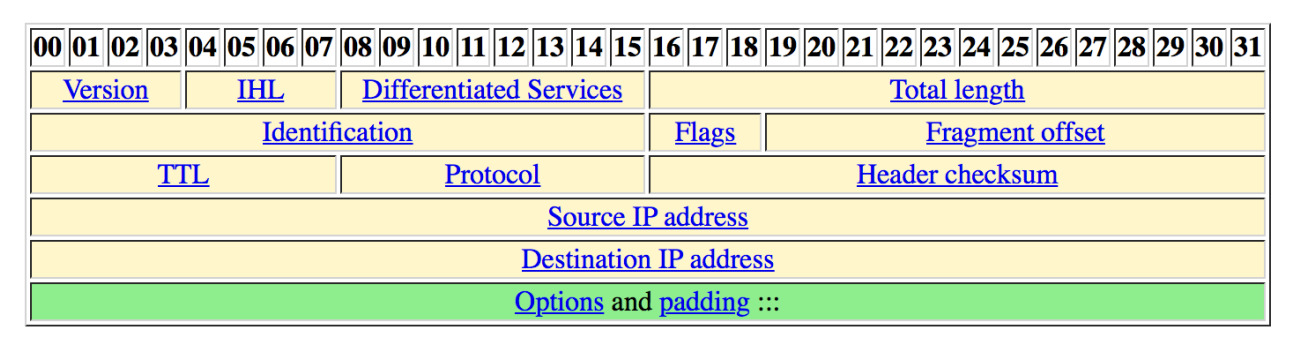
3DIC Cell-Based DataFlow Processor
Communication Cell Formats

IP header:


00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
IHL
Differentiated Services
Total length
Identification
Flags
Fragment offset
TTL
Protocol
Header checksum
Source IP address
Destination IP address
Options and padding :::

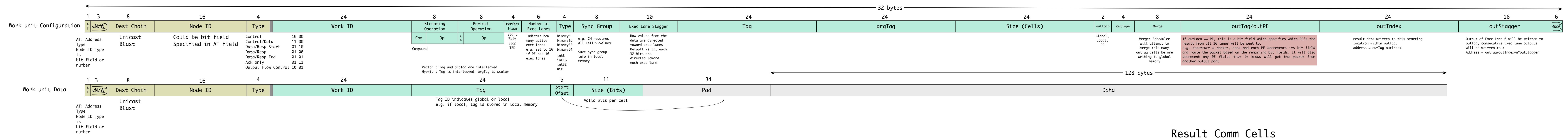
For MCast, use an MCast address which identifies n PE's but use an option that identifies which PE's have been visited.
e.g.option initially set to {64{1'b1}} by source, each PE router maps MCast address to a 64 bit field, then AND's with option field
Each PE will clear its bit in the option field

Rambling

If all columns are allowed to run asynchronously, do we need a means to identify when a WU is complete before another can start.
Perhaps have a 64 element WU complete field in a WU descriptor that each column sets???? rambling

Maybe we should have a global and a local sync, a layer complete would be a global sync where one layer doesn't start untilmthe previous layer is complete.

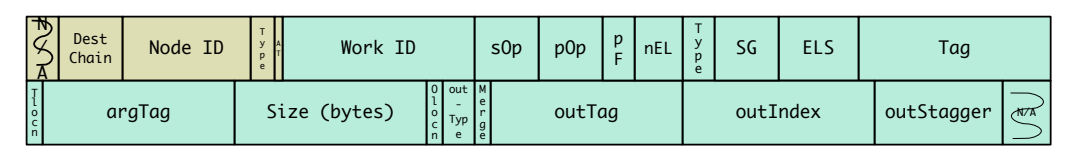
Control/Data Cells : The cell sizes below over worst case sizes. For example, we allocate space for up to 64K nodes on 256 chains.
In practice as shown in the examples, these sizes would be much lower.
In the case of sparsey or ConvNet, they can be ROI's or Kernels.
For Sparsey, we might have 100 MAC's with 1000 cells resulting in 100,000 WID's for layer0.
For ConvNet, we might have 96 kernels each operating on an ROI
Each WID specifies Tag for Cell weight matrix and argTag for the ROI


32 bytes
Work unit Configuration
1 3 8 16 4 24 8 8 4 6 4 8 10 24 24 24 2 4 8 24 24 16 6
AT
N/A
Dest Chain
Node ID
Type
Work ID
Streaming Operation
Perfect Operation
Perfect Flags
Number of Exec Lanes
Type
Sync Group
Exec Lane Stagger
Tag
argTag
Size (Cells)
outLocn
outType
Merge
outTag/outPE
outIndex
outStagger
N/A
AT: Address Type
Node ID Type is bit field or number
Unicast
BCast
Could be bit field
Specified in AT field
Control 10 00
Control/Data 11 00
Data/Resp Start 01 10
Data/Resp 01 00
Data/Resp End 01 01
Ack only 01 11
Output Flow Control 10 01
Com
Op
AR
Op
Compound
Start
Wait
Stop
TBD
Indicate how many active exec lanes e.g. set to 16 if PE has 16 exec lanes
binary8
binary16
binary32
binary64
int8
int16
int32
Bit
e.g. CM requires all Cell v-values
Save sync group info in local memory
How values from the data are directed toward exec lanes Default is 32, each 32-bits are directed toward each exec lane
Vector : Tag and argTag are interleaved
Hybrid : Tag is interleaved, argTag is scalar
Global, Local, PE
Merge: Scheduler will attempt to merge this many outTag cells before writing to global memory
result data written to this starting location within outTag.
Address = outTag+outIndex
Output of Exec Lane 0 will be written to outTag, consecutive Exec lane outputs will be written to :
Address = outTag+outIndex+n*outStagger
128 bytes
Work unit Data
1 3 8 16 4 24 24 5 11 34
AT
N/A
Dest Chain
Node ID
Type
Work ID
Tag
Start Ofset
Size (Bits)
Pad
Data
Unicast
BCast
Tag ID indicates global or local
e.g. if local, tag is stored in local memory
Valid bits per cell

If outLocn == PE, this is a bit-field which specifies which PE's the result from all 16 lanes will be sent to.
e.g. construct a packet, send and each PE decrements its bit field and route the packet based on the remaining bit fields. It will also decrement any PE fields that it knows will get the packet from another output port.

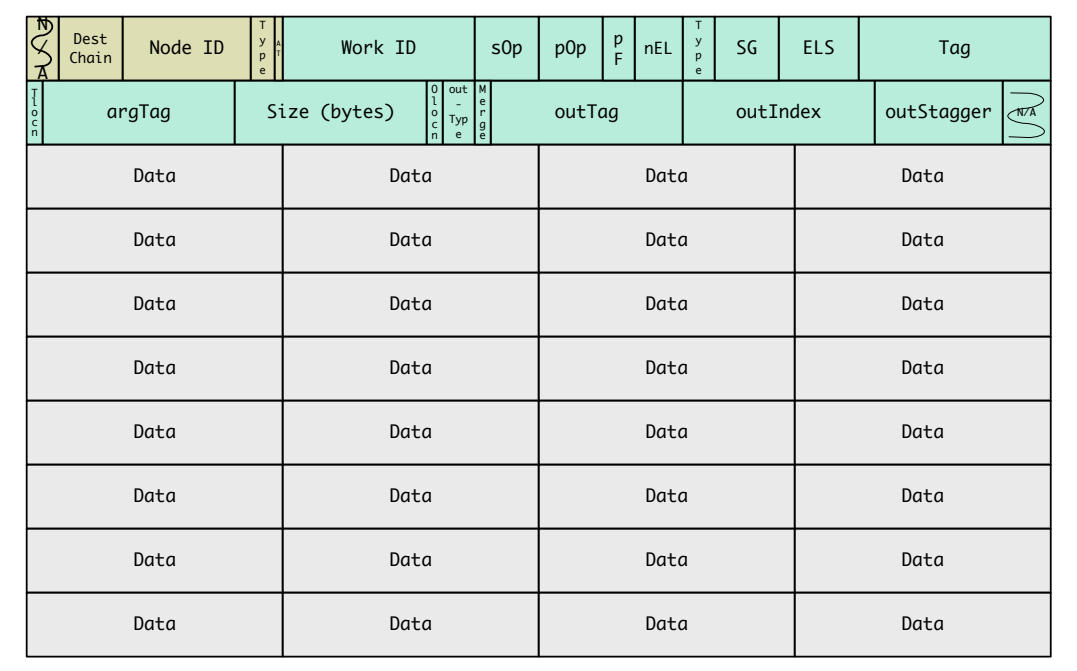
Request Comm Cells

Work-Unit Control
Total Bits = 256/cell



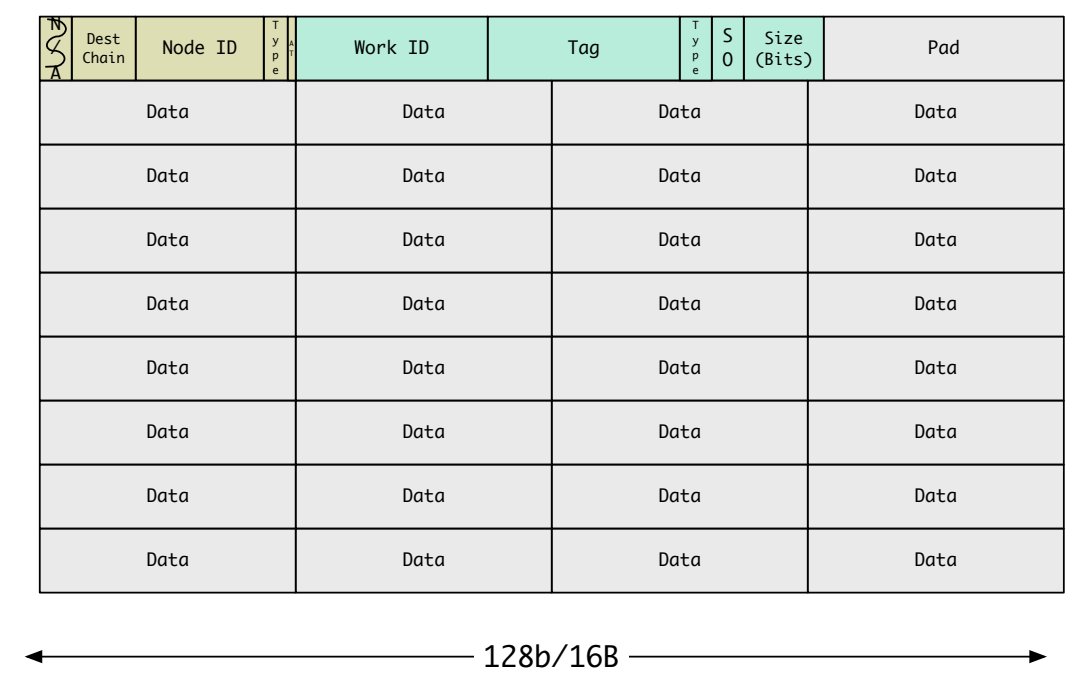
We will likely need to send "sync" commands to PE's with info such as which PE's to sychronize to.
In these cases, a single large cell will be inefficient from a bandwidth perspective. For FIFO's, we have already sized them for data cells.
We need to balance comm logic complexity versus efficiency and latency of control messages.
So perhaps we should balance this by having two lengths, one for control/special and one for data.

Work-Unit Control/Data
Total Bits = 1280/cell

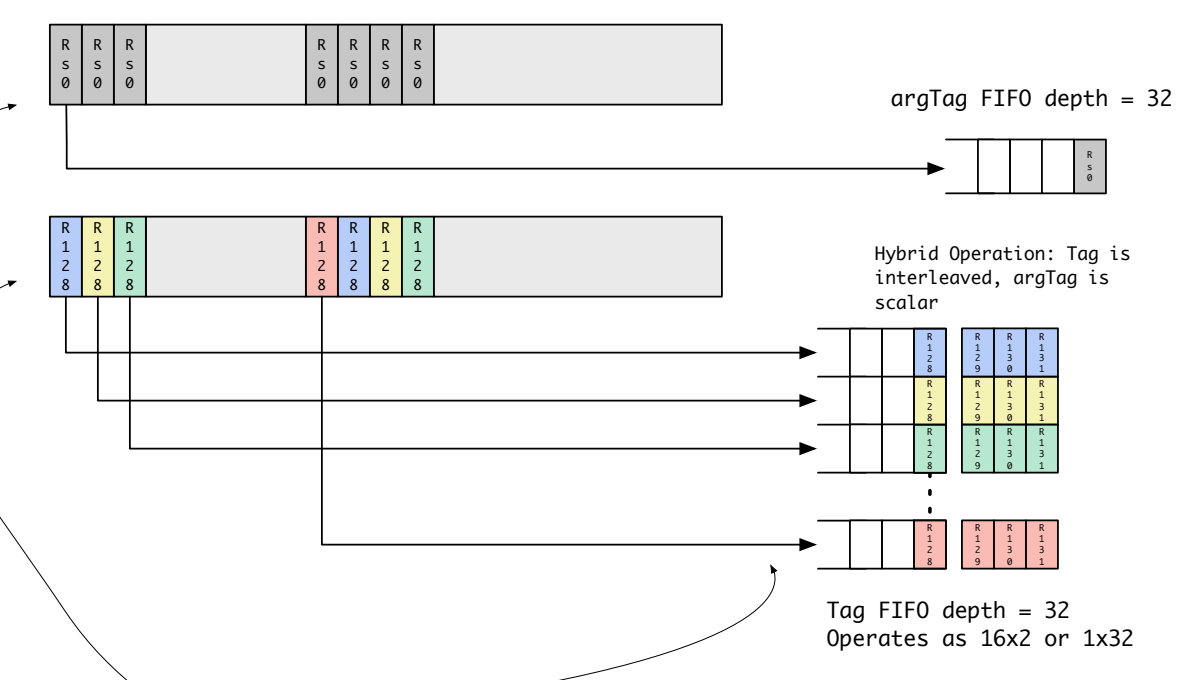


Used if :
a) Tag data wholly contained in cell. argTag can still be streaming if sOp is valid
b) sOp is write to shmem (2-words of data saved to each Exec Lanes shared memory)

Request Data
Total Bits = 1152/cell

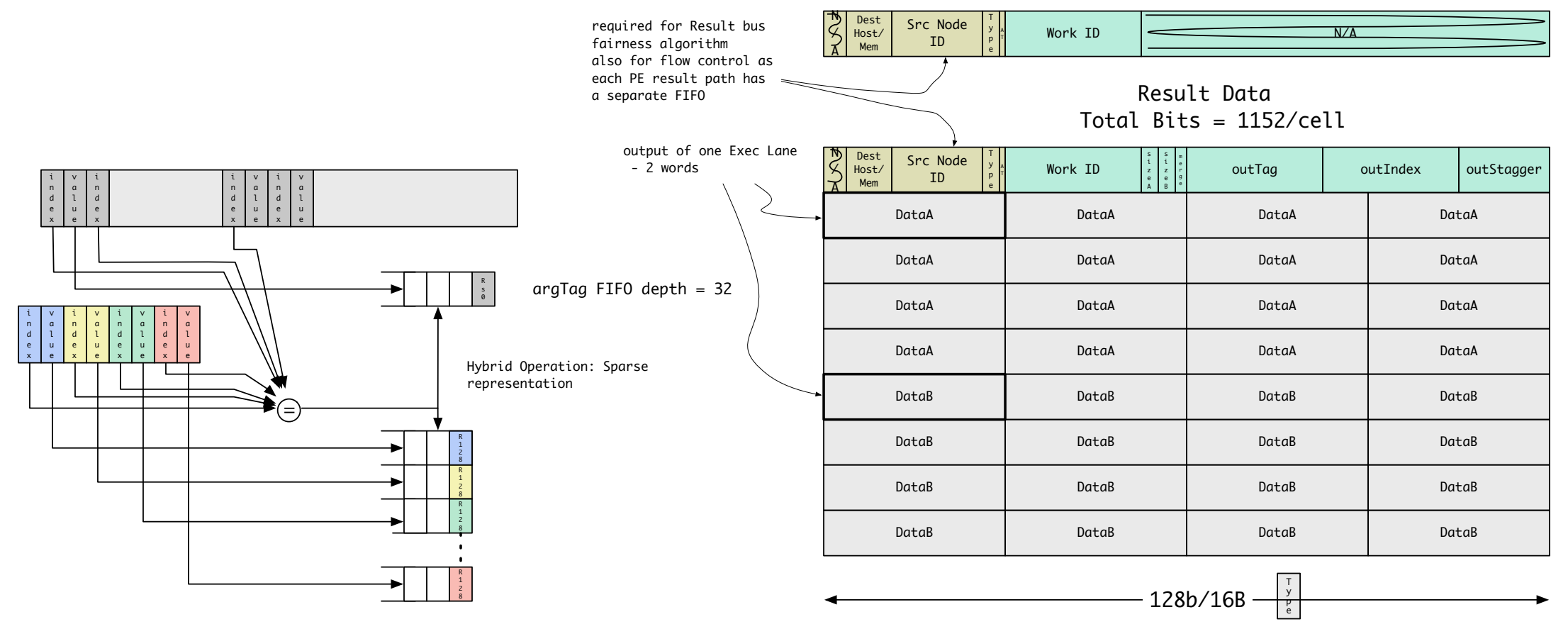


Each data cell has sufficient data for 2-quantums for 16-exec lanes, 1-quantum for 32-exec lanes



Result Comm Cells

Ack Only


N/A
Dest Host/Mem
Src Node ID
Type
Work ID
N/A
required for Result bus fairness algorithm also for flow control as each PE result path has a separate FIFO
output of one Exec Lane - 2 words
argTag FIFO depth = 32
Hybrid Operation: Sparse representation
Result Data
Total Bits = 1152/cell
outTag
outIndex
outStagger
DataA
DataB
128b/16B
Type

Each data cell has sufficient data for 2-quantums for 16-exec lanes.
For result data cells, the two returned words must be consecutive in memory

If PE returns multiple result cells, then last cell has Type=Payload End

Some result cells the two data fields can contain {value, offset}, in which case the outStagger field is ignored and all the lane offset fields are wrt outIndex

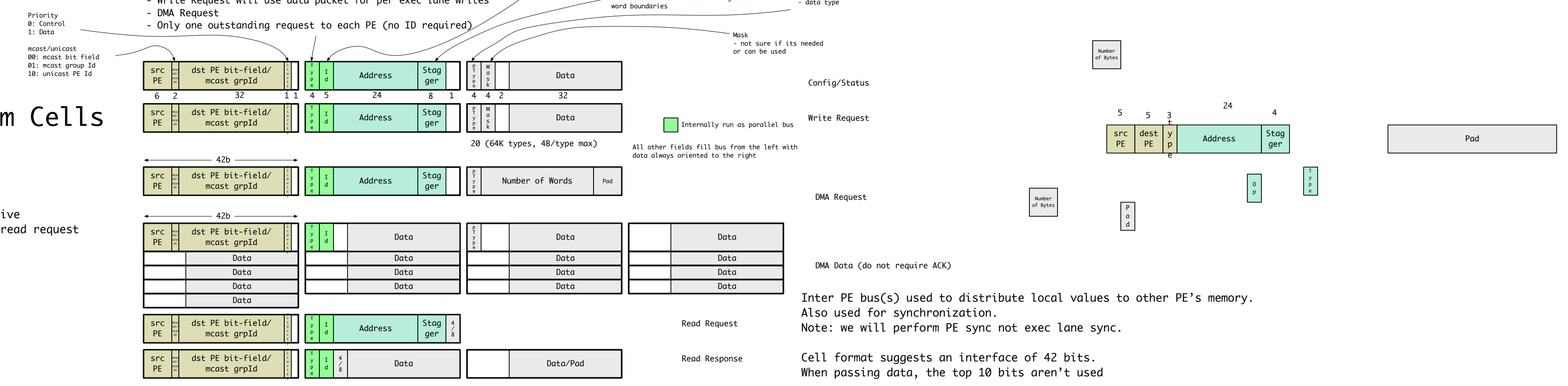
Result fields options:
{value, value}
{value, offset}

Data Memory : Data Memory can be a combination of linear or interleaved blocks of data. For sparsey and ConvNet, the ROI is a common block of data that is used in multiple Work Units. In this case, the block of data allocated for ROI will be linear.
For weight matrices/kernels, we will interveave the Kernel to allow the individual Kernels to be spread across execution lanes

Inter-PE Comm Cells

Inter PE Data Packet :
1) as a result of a WU directive
2) as a result of a write or read request
3) DMA request

Control Packet :
- Config/Status used to provide status between
- Read Request
- Write Request will use data packet for per exec lane writes
- DMA Request
- Only one outstanding request to each PE (no ID required)



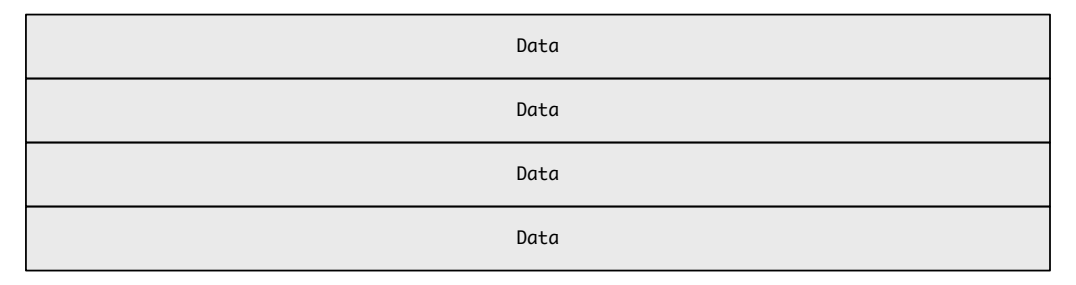
Inter PE bus(s) used to distribute local values to other PE's memory.
Also used for synchronization.
Note: we will perform PE sync not exec lane sync.

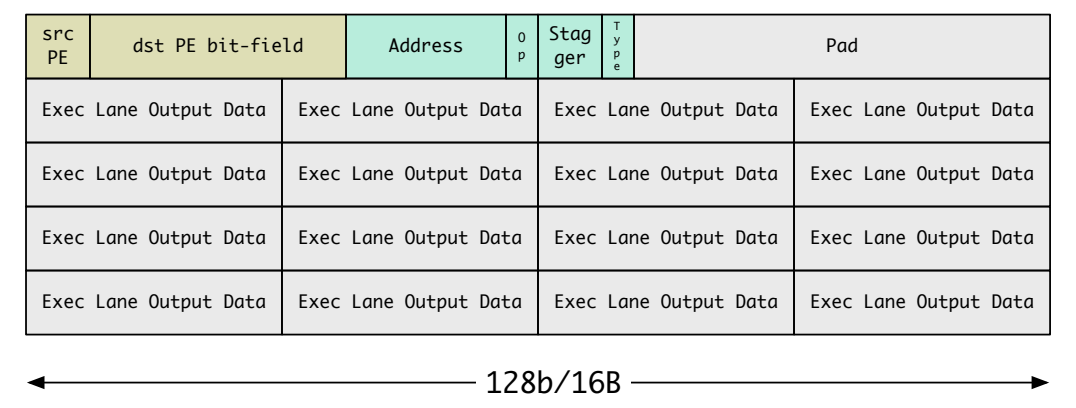
Cell format suggests an interface of 42 bits.
When passing data, the top 10 bits aren't used

External NoC Interface
- 42 bits
- Supports {Type (4), laneId(4), streamId(1), Address(24), Stagger(8), 4/8(1)}

Internal NoC interface
- Type (4)
- laneId (4-bits: 0..15)
- streamId (1-bit: 0,1)
- Data (40-bits: Supports {srcPe(5), mcast(2), mcastBitMaskAddr(32)},{Address(24), Stagger(8)})

If we need to use the inter-PE bus to communicate status such as synchronization, then we might need to make the inter-PE cell size variable



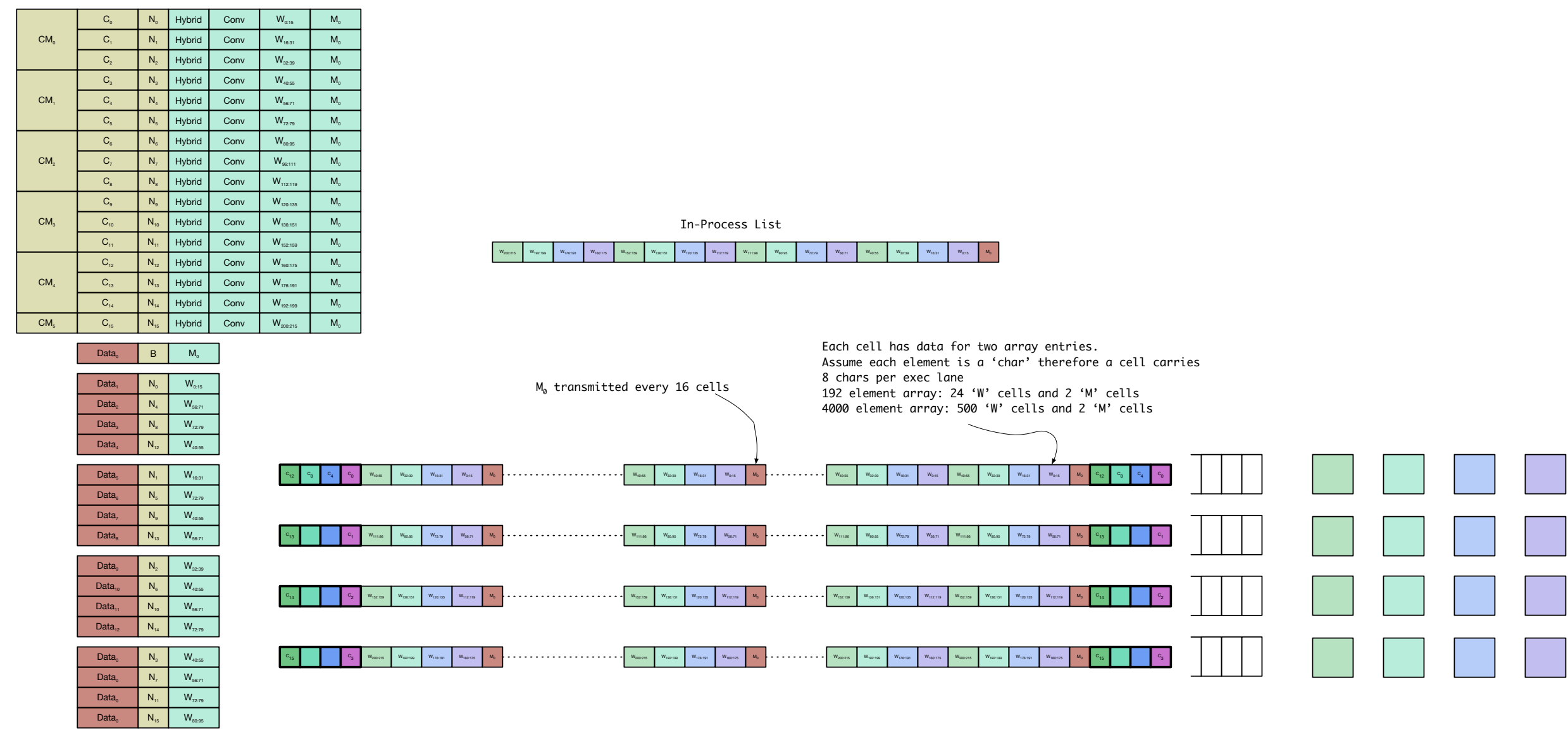


Need to construct a cell from the results of all exec lanes to be transmitted across the PE comm interface

Cell will be ATM-like, with VPI/VCI being a bit field.

Use a 32-bit bus width??

Need to think about a sync mechanism to ensure all local memory is updated
- processing steps controlled by main controller?
- all's PE's acknowledge step completion (using write to memory semaphore)?
- with cct switched the comm path is fixed, so send sync message as last packet



Layers are constructed from a list of work elements.
When all the work elements are complete, the layer is complete.

Work memory : a list of work elements.
In the case of sparsey or ConvNet, they can be ROI's or Kernels.
For Sparsey, we might have 100 MAC's with 1000 cells resulting in 100,000 WID's for layer0.
For ConvNet, we might have 96 kernels each operating on an ROI
Each WID specifies Tag for Cell weight matrix and argTag for the ROI

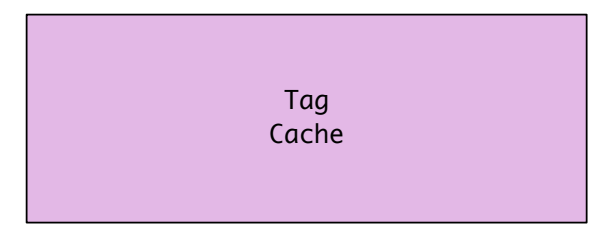
Sparsey: 100,000 MAC's.
BU Input ~ 192b = 24
H Input ~ 5000b = 156.25B

Example Sizes

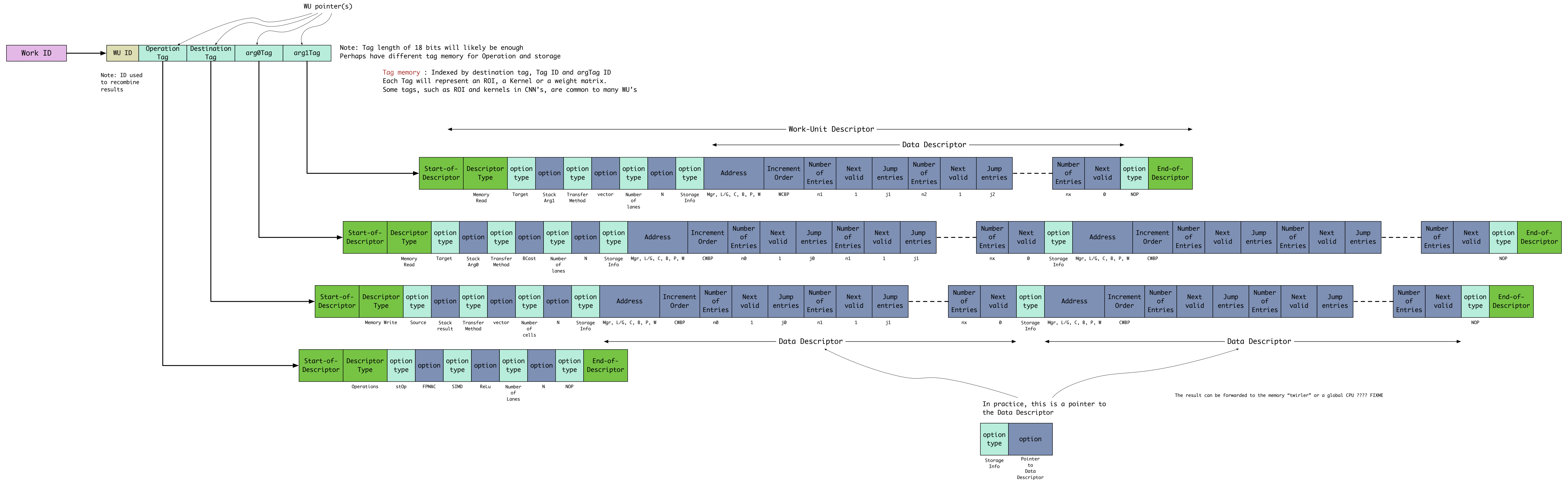
Number of Chains	4
Dest Chain Size	2
Number of nodes	16
Nodes per chain	4
Node ID Size	2
Tag Address Size	17
Out Index Size	14

128Kx85

Scheduler Command



128Kx56



Note: Tag length of 18 bits will likely be enough
Perhaps have different tag memory for Operation and storage

Tag memory : Indexed by destination tag, Tag ID and argTag ID
Each Tag will represent an ROI, a Kernel or a weight matrix.
Some tags, such as ROI and kernels in CNN's, are common to many WU's

The result can be forwarded to the memory "twirler" or a global CPU ???? FIXME