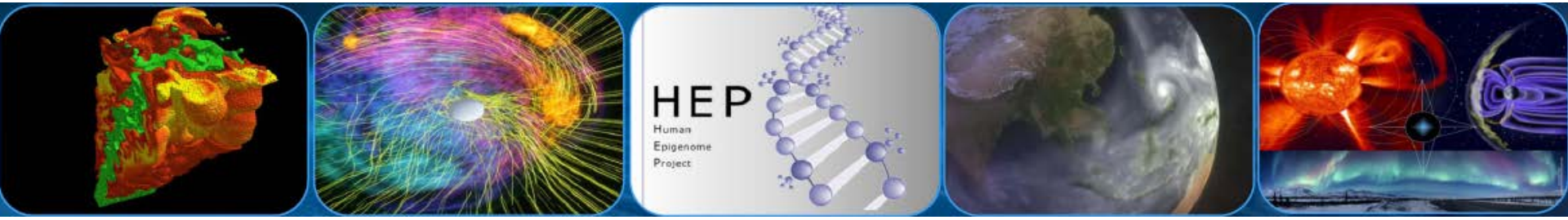
A Coarse-grained Stream Architecture for Cryo-electron Microscopy Images 3D Reconstruction

High Performance Computer Research Center, ICT, CAS, China

Wendi Wang (wangwendi@ncic.ac.cn)
Bo Duan
Wen Tang
Chunming Zhang
Guangming Tan
Peiheng Zhang
Ninghui Sun

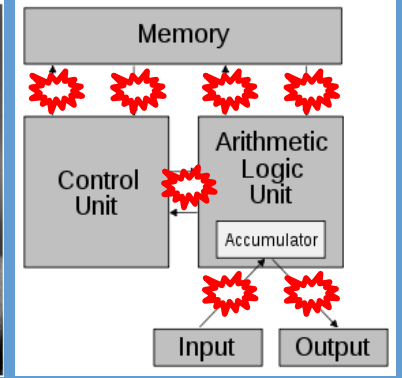
February 24, 2012@Monterey, CA, USA

Apps. getting harder to deal with?



3D Simulation | High energy Phy. | Bioinformatics | Climate | Astronomy

"Von Neumann Bottlenect"

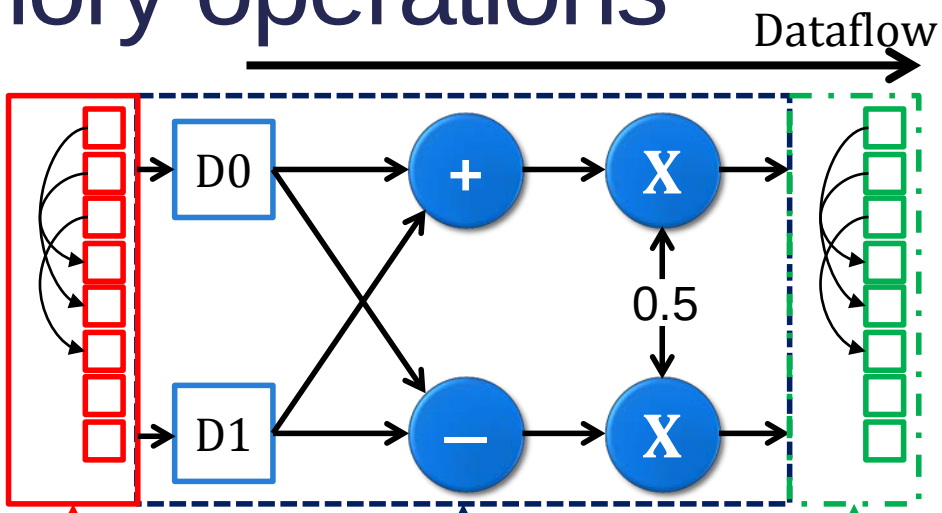


John v. Neumann

Domain-specific Reconfigurable Architectures

Computing & Memory operations

- Mixed up operation flow
  - Address calculation
  - Loop boundary
  - ……

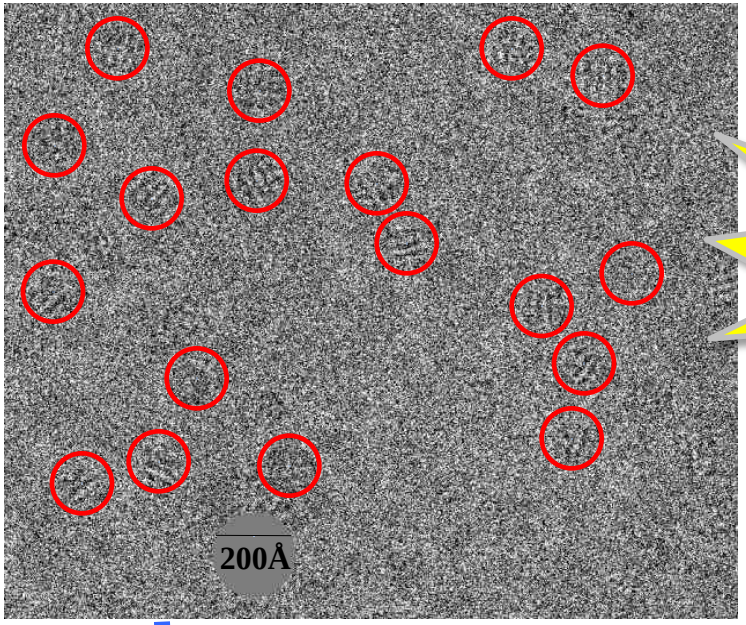



```
for(j=0:1:n)
    for(i=0:1:n)
    {   nj=n*j*2;
        iidx0 = (nj+i)*2;          oidx0 = (nj+j*2+i)*2;
        iidx1 = (nj+n*2-i)*2;      oidx1 = (nj+j*2+n+1)*2;
        m[oidx0]     =(p[iidx0]   + p[iidx1])/2;
        m[oidx1+1]   =(p[iidx1]   - p[iidx0])/2;
        m[oidx1]     =(p[iidx0+1] + p[iidx1+1])/2;
        m[oidx0+1]   =(p[iidx0+1] - p[iidx1+1])/2; }
```

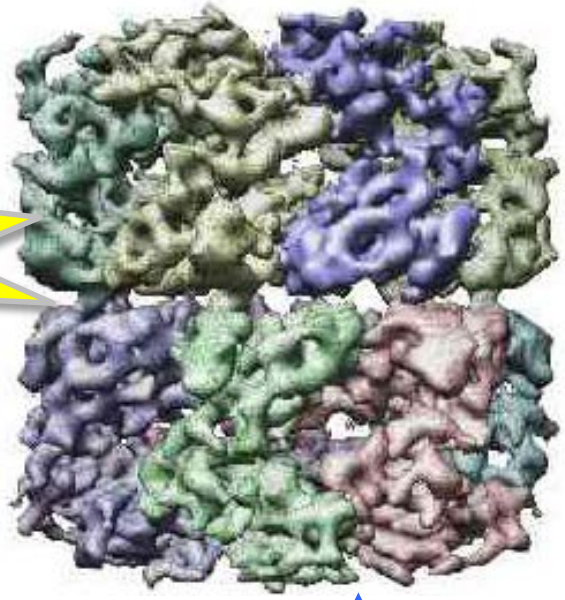
Outline

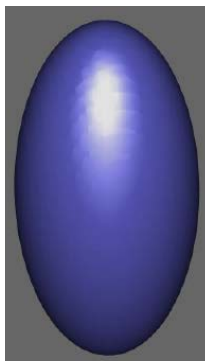
- Motivation
- Single Particle 3D Reconstruction
- Stream Architecture
- Computing Stream Mapping
- Performance Comparison
- Conclusion

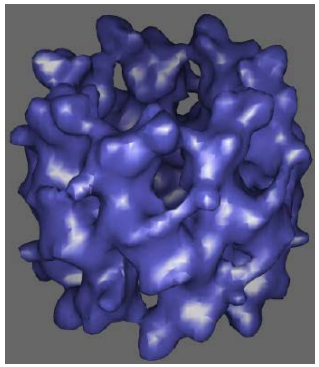
Single Particle 3D Reconstruction

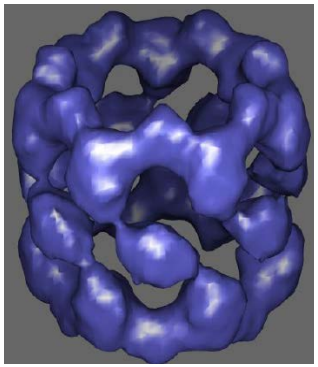


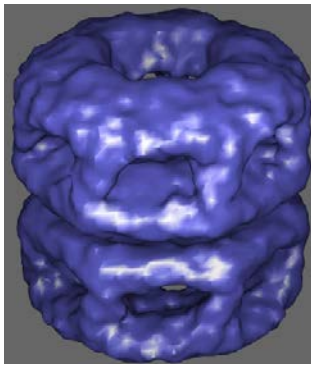
Potential Petaflops application

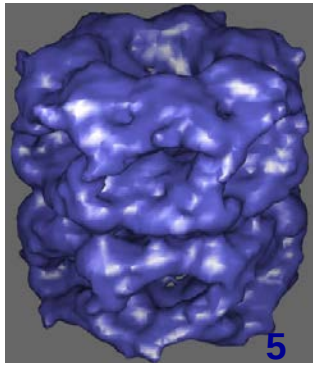




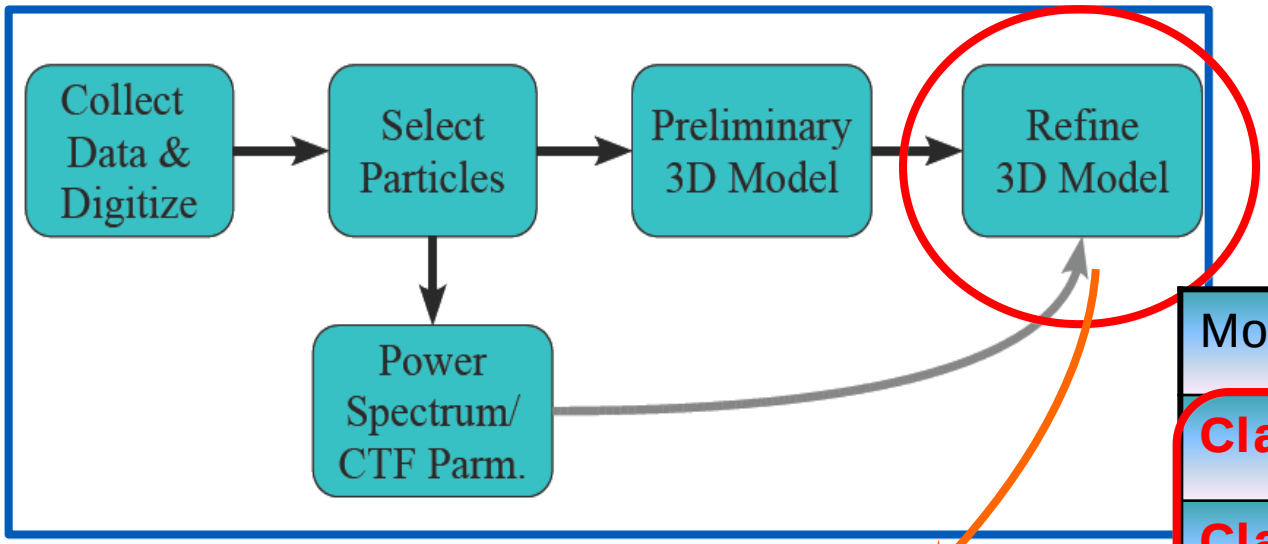


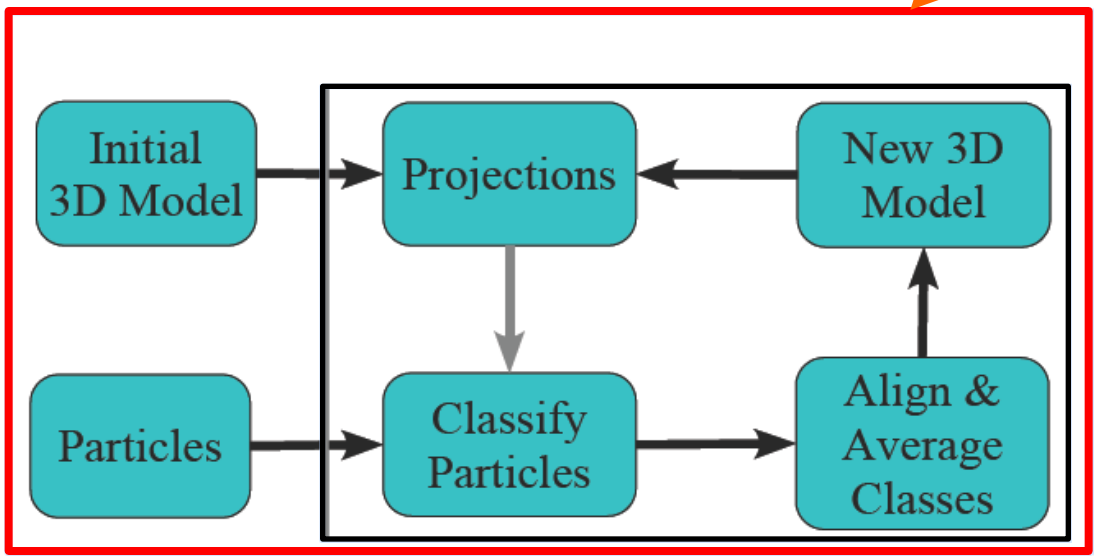






Hotspots of EMAN





97%
(RTAlign)

Module	Ratio
Classalign2	**50.59%**
Classesbymra2	**24.86%**
Classesbymra1	**21.52%**
Make3d	1.61%
Project3d	0.65%
Projtree	0.03%
Proc3d	0.03%

Kernel classification of a complex app.

- Computing Kernels
- Memory Access Patterns: reusable!
- Streams = Computing Kernels + Memory Access Patterns

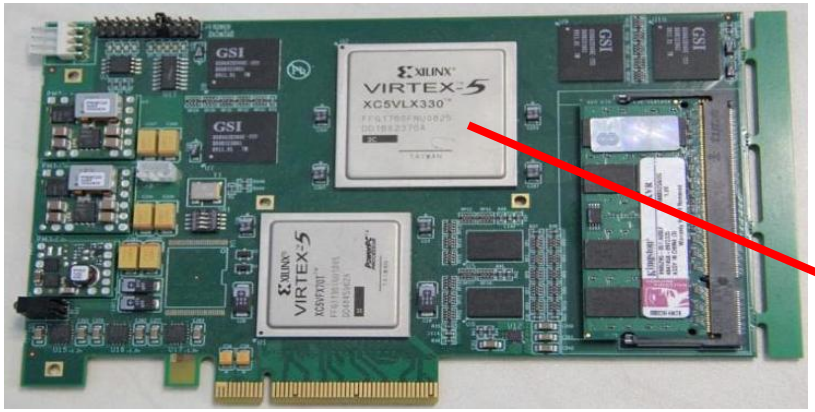
Name	Category	Description
MCF	Computing	Autocorrelation
CCFX	Computing	Cross correlation on x-dimension
CCF	Computing	Cross correlation with reference
Unwrap	Computing	Rectangular coordination to polar one
Rotate	Computing	Rotates with the best rotation angle
Translate	Computing	Translates image with the maximal CCF
DOT	Computing	Scores the rotational&translational alignment
Clip & Zero Padding	Memory	Change the size of images by clipping and padding
Rot180	Memory	Rotate image by 180°
hFlip	Memory	Flip images horizontally
Shift	Memory	Translate image by given 2D offset
Bit Reversal & Transposition	Memory	Used in FFT kernels
Matrix Transposition	Memory	Used in the row-column 2D-FFT kernels
RTFAlign	Stream	Align images using RTAlign and hFlip
RTAlign	Stream	Align images using Rotate, Translate, CCFX, CCF and DOT
MakeRFP	Stream	Calculate the rotating footprint using MCF and Unwrap

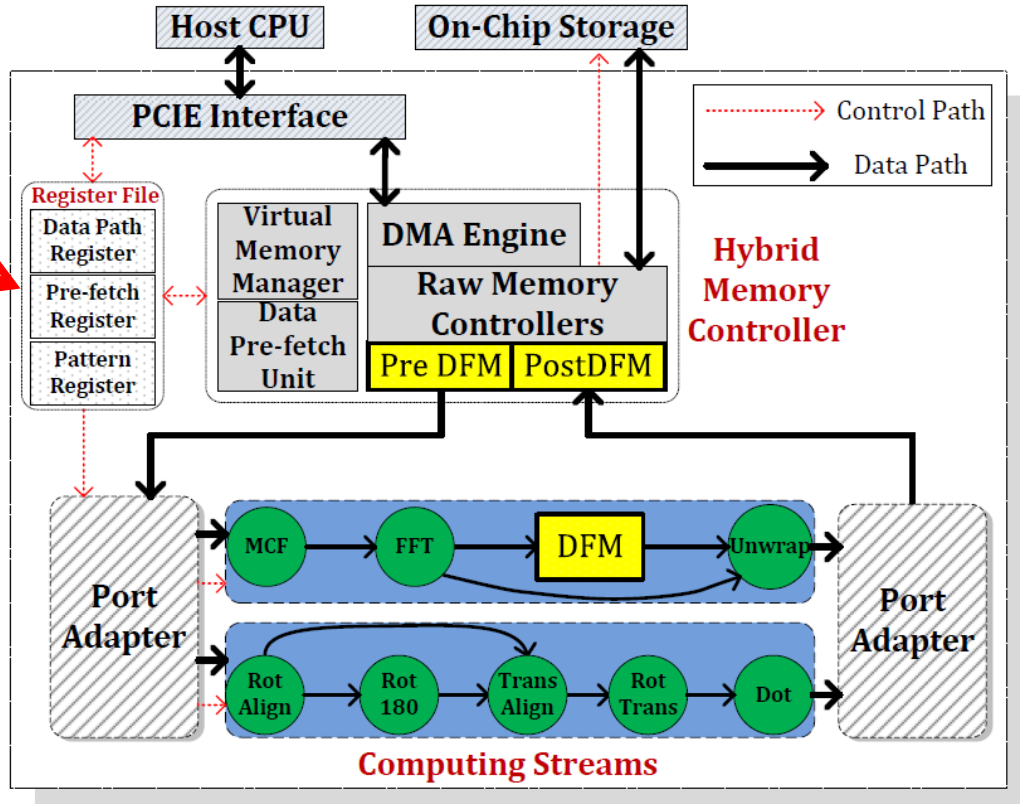
*Some kernels are combined together

Outline

- Motivation
- Single Particle 3D Reconstruction
- Stream Architecture
- Computing Stream Mapping
- Performance Comparison
- Conclusion

Accelerator card prototyping

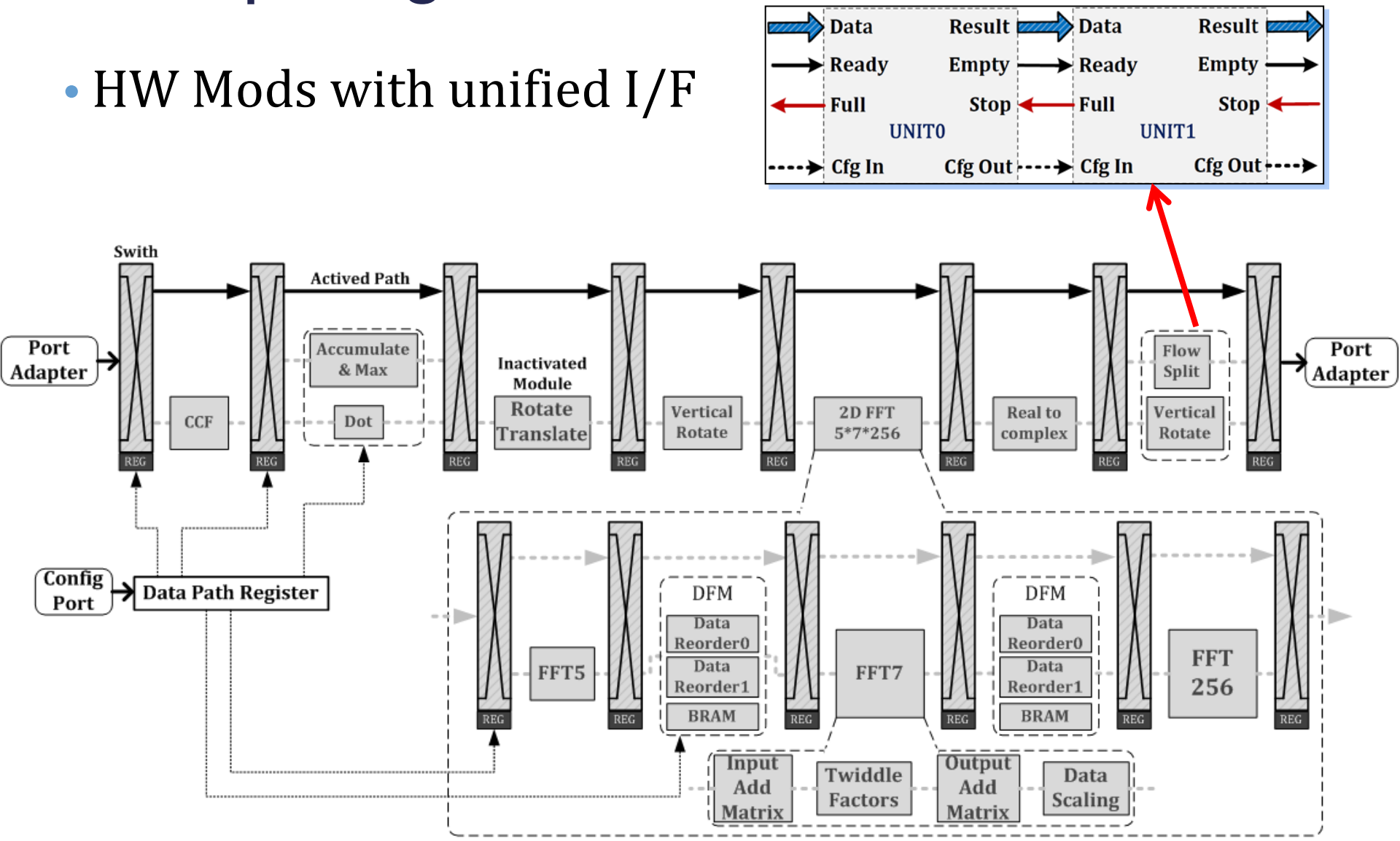




- PCIe Interface~**2GB/s**
- DDR2 SO-DIMM~**3.2GB/s**
- 16MB SRAM~**6.4GB/s**
- V5LX330 for computing
- V5LX70T for housekeeping

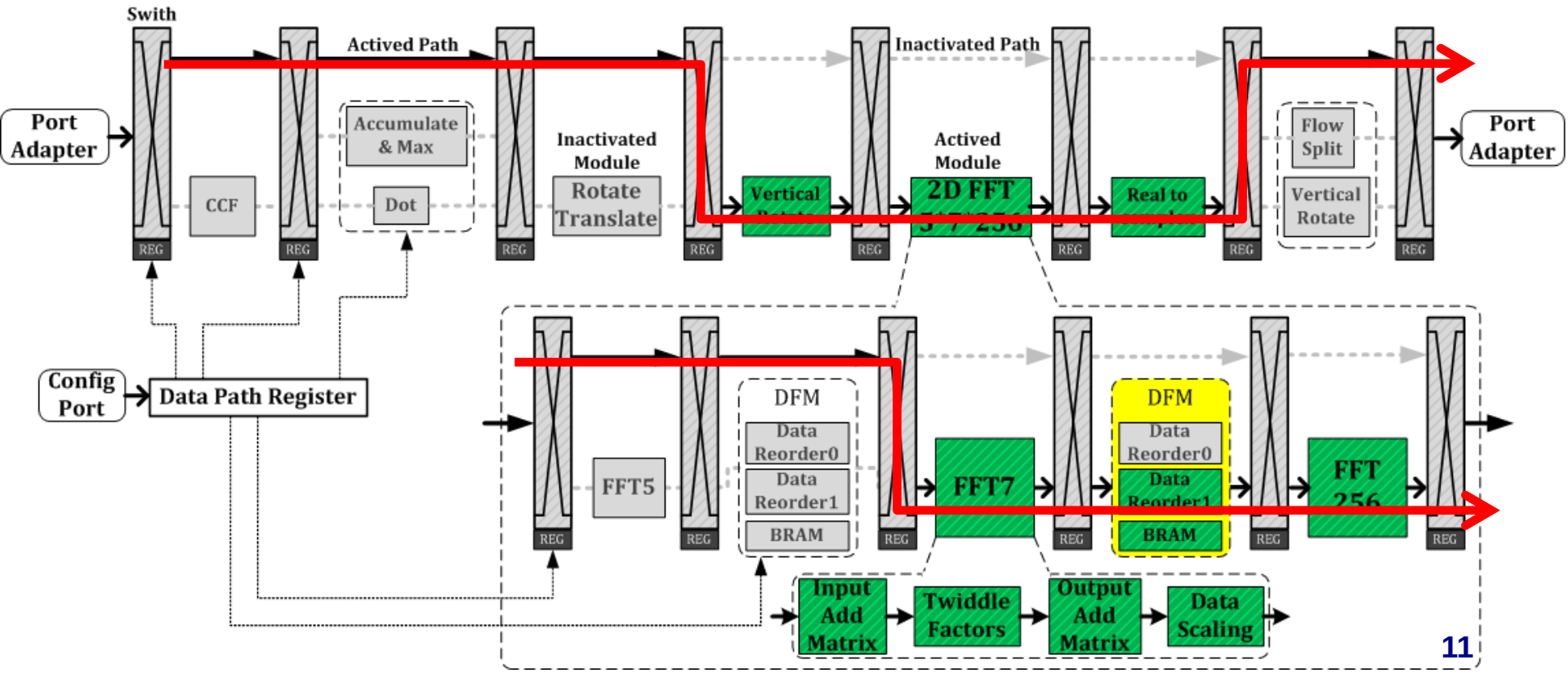
Computing stream

- HW Mods with unified I/F



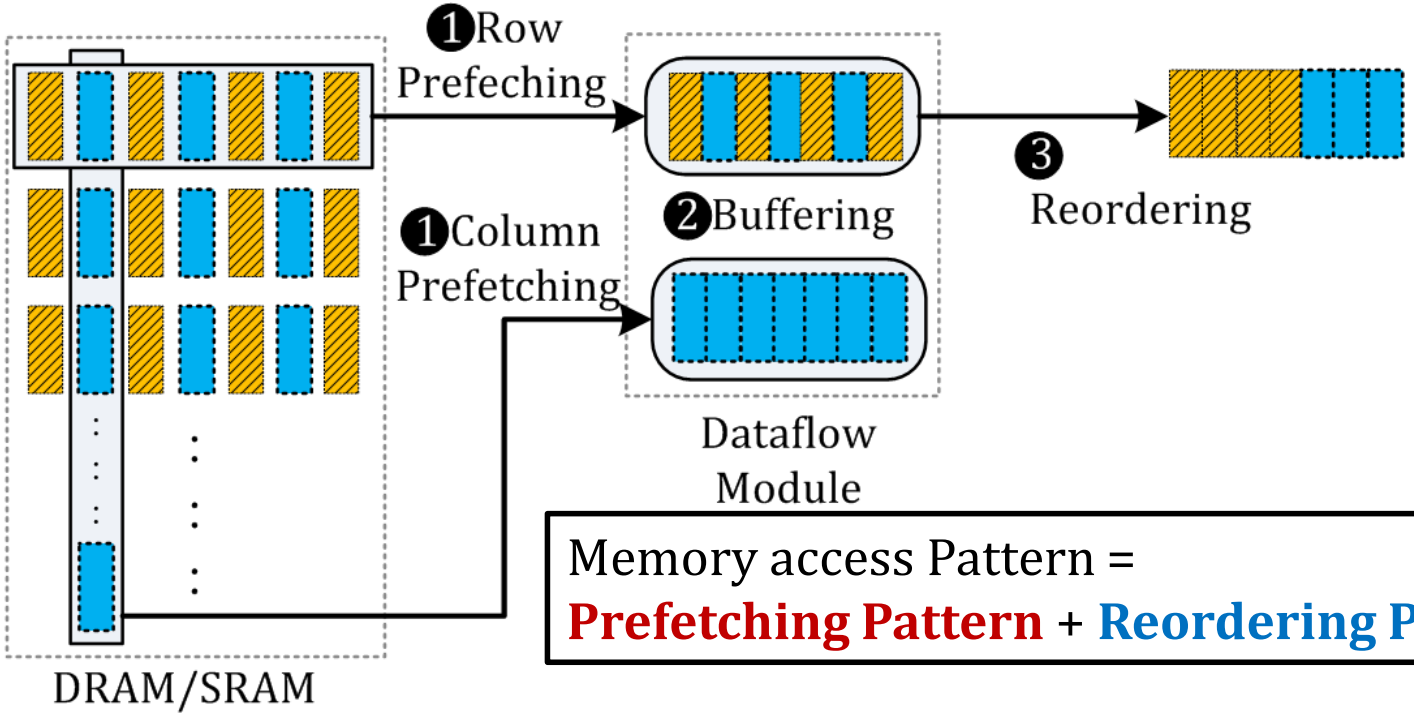
Computing stream

- HW Mods with unified I/F
- Configurable (switch) data path

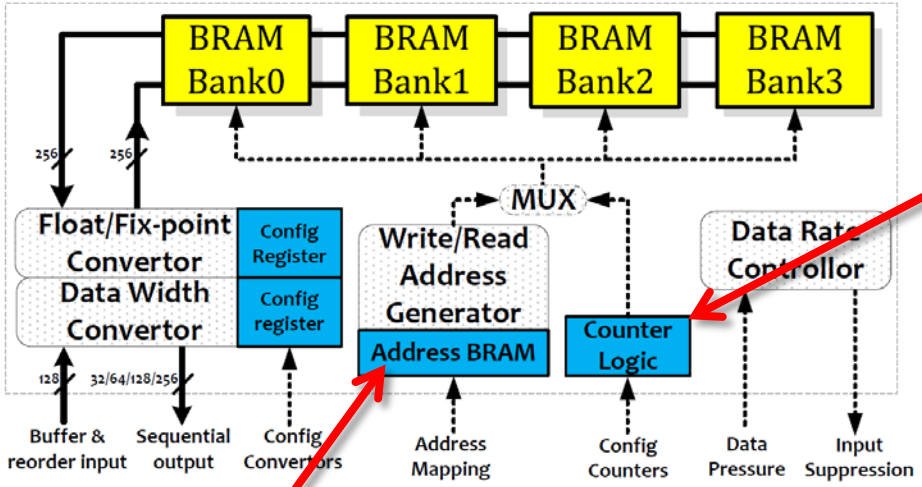


Pattern-based memory access

- 3 steps of memory accessing
  - **Prefetching**: Row(column)-order data prefetching
  - **Buffering**: Storing data in DFM
  - **Reordering**: Data reordering with predefined patterns



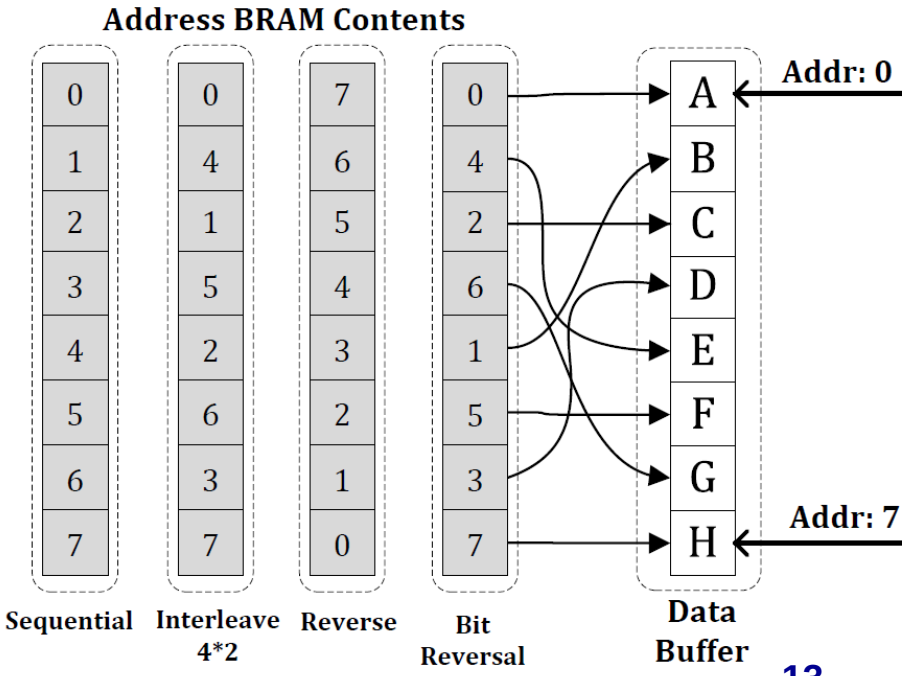
Dataflow Module & Reordering Patterns



1 Configurable counter

2 Indirect memory access

$$Dout[i] = Din[\mathbf{Addr[i]}]$$

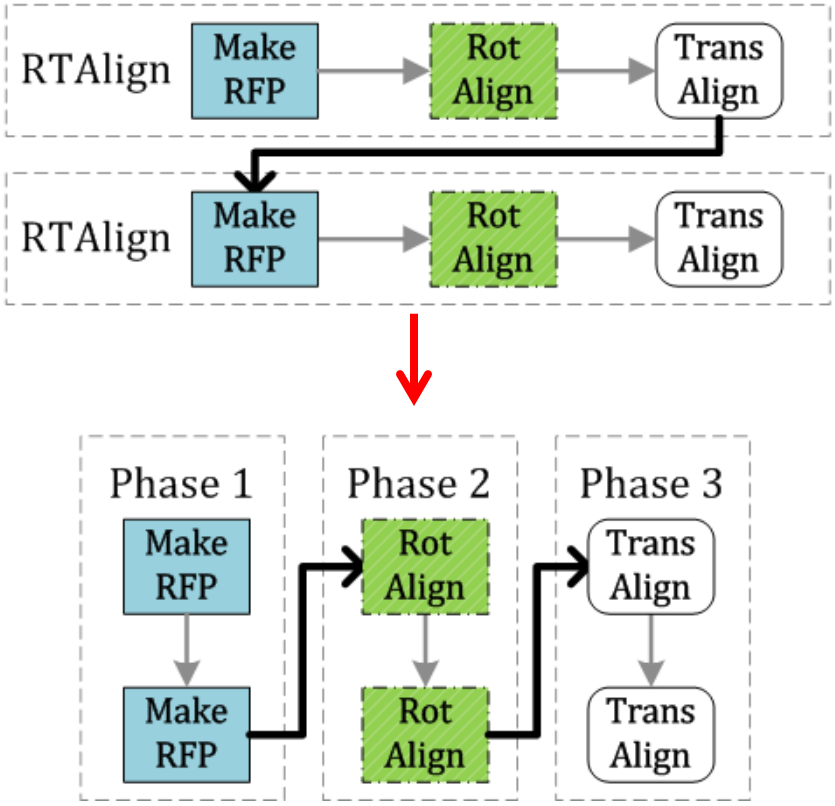


Outline

- Motivation
- Single Particle 3D Reconstruction
- Stream Architecture
- Computing Stream Mapping
- Performance Comparison
- Conclusion

Mapping streams to multi-step process

- Divide CFG into steps
- Computing in batch mode



12 steps of the RTAlign stream

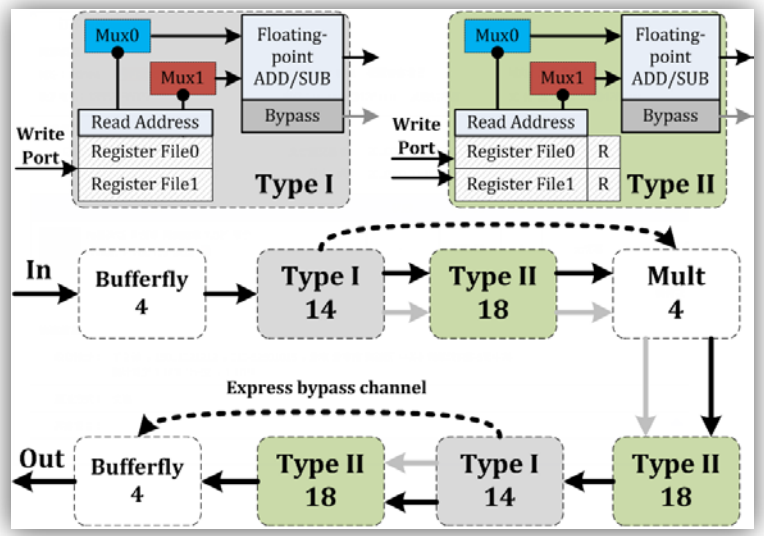
#	Function	Activated Modules	Memory Controller[1]
1	1D FFT	2D FFT	I: interleave/ O: deinterleave
2	1D IFFT/ CCFX/Acc.	2D FFT/CCF/ Acc. & Max	I: interleave/2-Op.[2]
3	Rotate	RotateTranslate	I: clip/random access
4	1D FFT	2D FFT	I: interleave/ O: column write/ deinterleave
5	1D FFT	2D FFT/ Post-Vertical Rotate	O: column write/ deinterleave
6	CCF	CCF	I: interleave/2-Op.[2] O: column write
7	1D IFFT	2D FFT/ Pre-Vertical Rotate	O: column write
8	1D IFFT	R2C	I: interleave
9	1D IFFT	2D FFT/Flow Split	
10	MAX	Acc. & Max	I: clip
11	translate		I: clip
12	Dot	Dot	I: interleave/2-Op.[2]

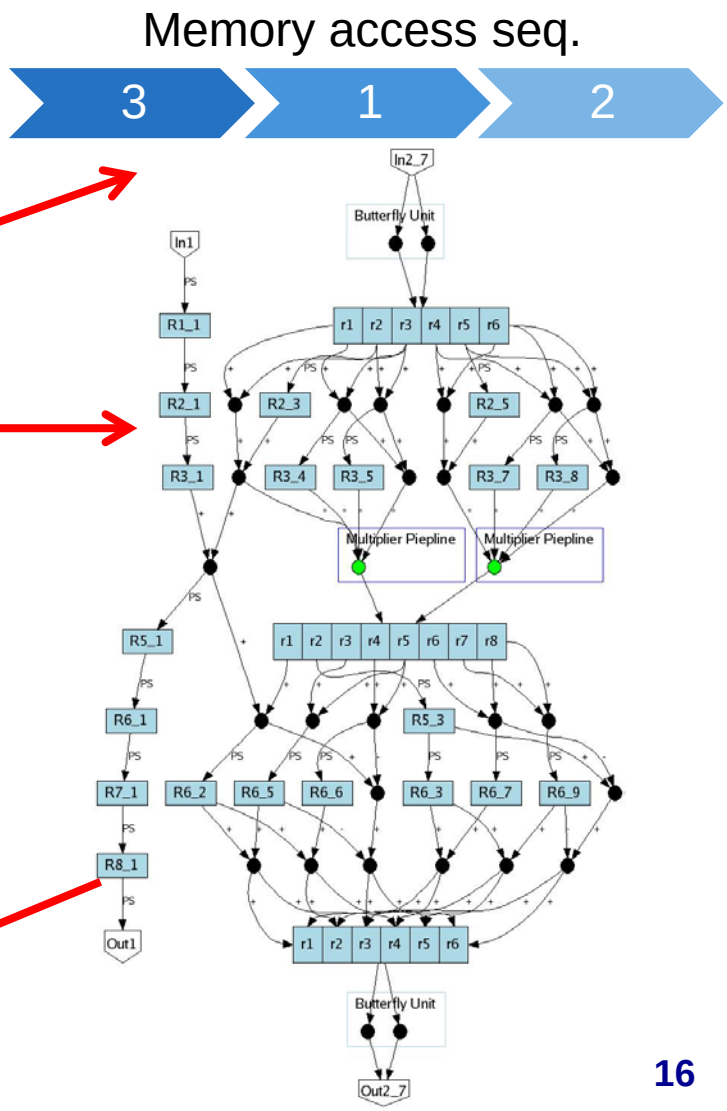
[1]I for input, O for output memory access patterns.
[2]Reading two operands from two separated addresses.

Kernel implementation

- HDL module generator
  - Mapping CFG to HDL code template
- Reserving memory access seq.

```
for(i=0:1:n)
{
   ......
}
```





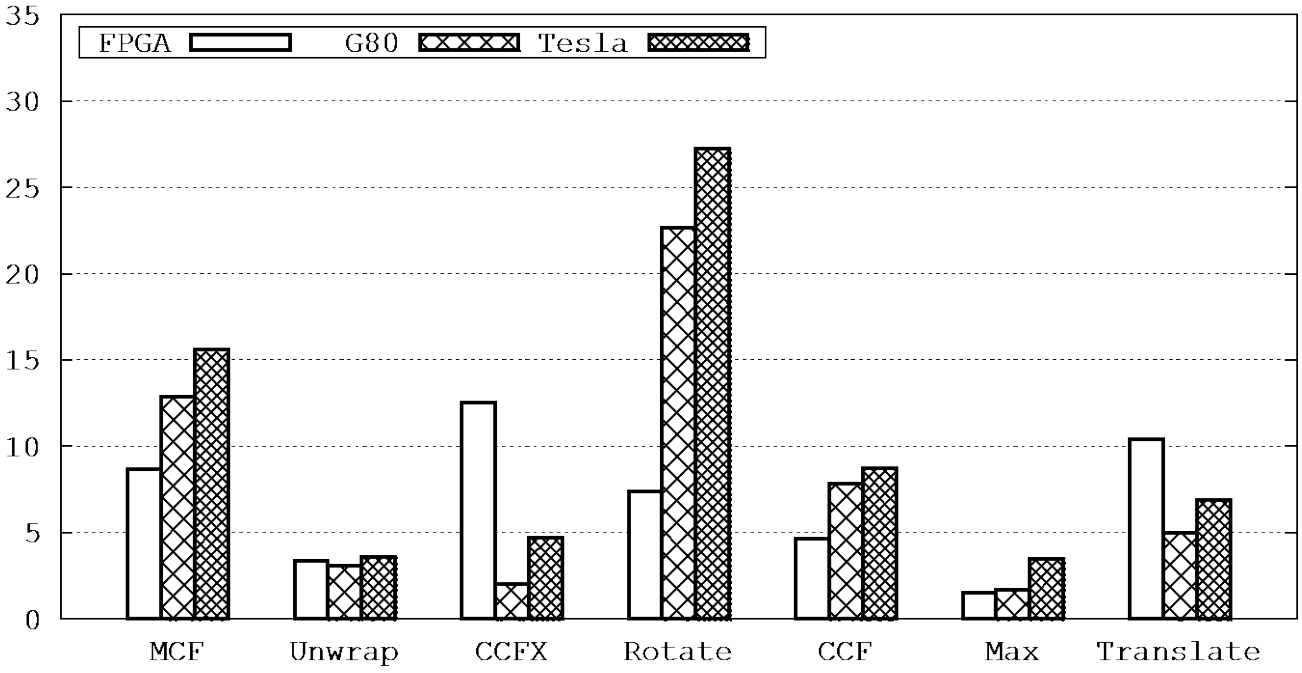
[Reconfig10, FPT11]

Outline

- Motivation
- Single Particle 3D Reconstruction
- Stream Architecture
- Computing Stream Mapping
- Performance Comparison
- Conclusion

Kernel Performance

- GPUs can be 5 times faster than FPGAs



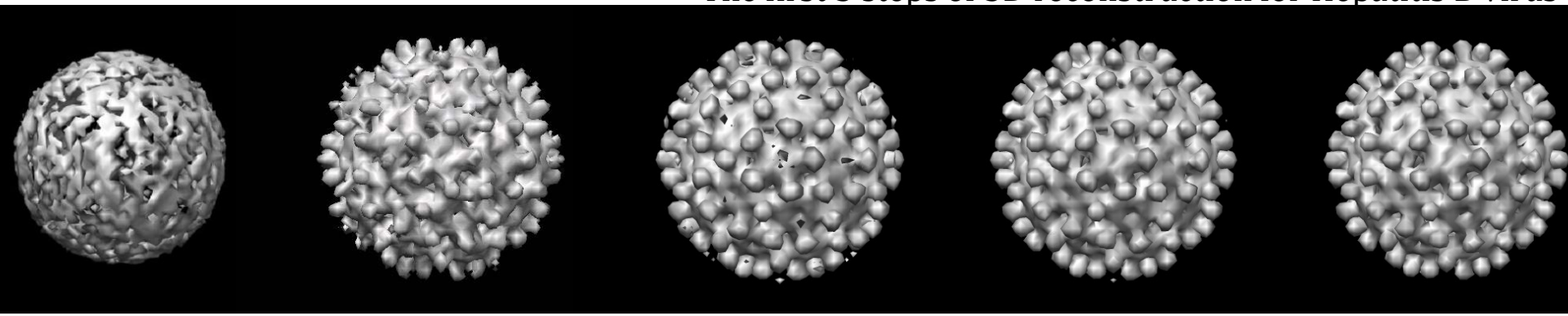
Nvidia Tesla GTX480 40nm

Intel Xeon E5520 45nm

Xilinx Virtex5 LX330 65nm

Overall performance

The first 5 steps of 3D reconstruction for Hepatitis B virus



- **3x** faster than a 4-cores CPU (Xeon E5520)
  - GPUs are 8x faster, due to the high GDDR5 bandwidth and massively parallel processing
- Slower than GPUs (**2x~4x**)
  - Limited by off-chip data bandwidth
- FPGAs are power efficient(**3x~4x**)

Conclusion

- Folding complex computing streams on FPGAs
  - Spt. memory access patterns from computing flow
  - Arch. support of pattern-based memory access
- 3 times faster than a 4-cores CPU
- Slower than GPUs but more power efficient

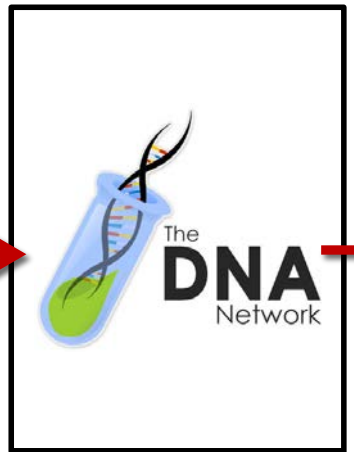
Future work

- In-socket co-processor based on Intel QPI
- Job scheduling on a heterogeneous cluster with FPGA-based accelerators [ICS09, HPDC11]

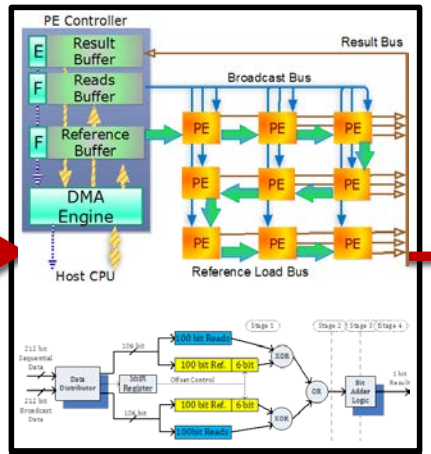
Future work

- In-socket co-processor based on Intel QPI
- Job scheduling on a heterogeneous cluster with FPGA-based accelerators [ICS09, HPDC11]
- Exploiting new application domains [IPDPSW12]

Architecture Scalability



Next Gen. Sequencing



Systolic Array

100x+ Speedup

Thanks for your attention!