Northeastern

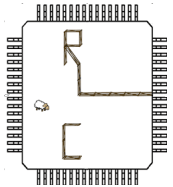
Accelerating Matrix Processing for MIMO Systems

Jieming Xu (now at MathWorks) and Miriam Leeser
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA
mel@coe.neu.edu

MathWorks®

XILINX

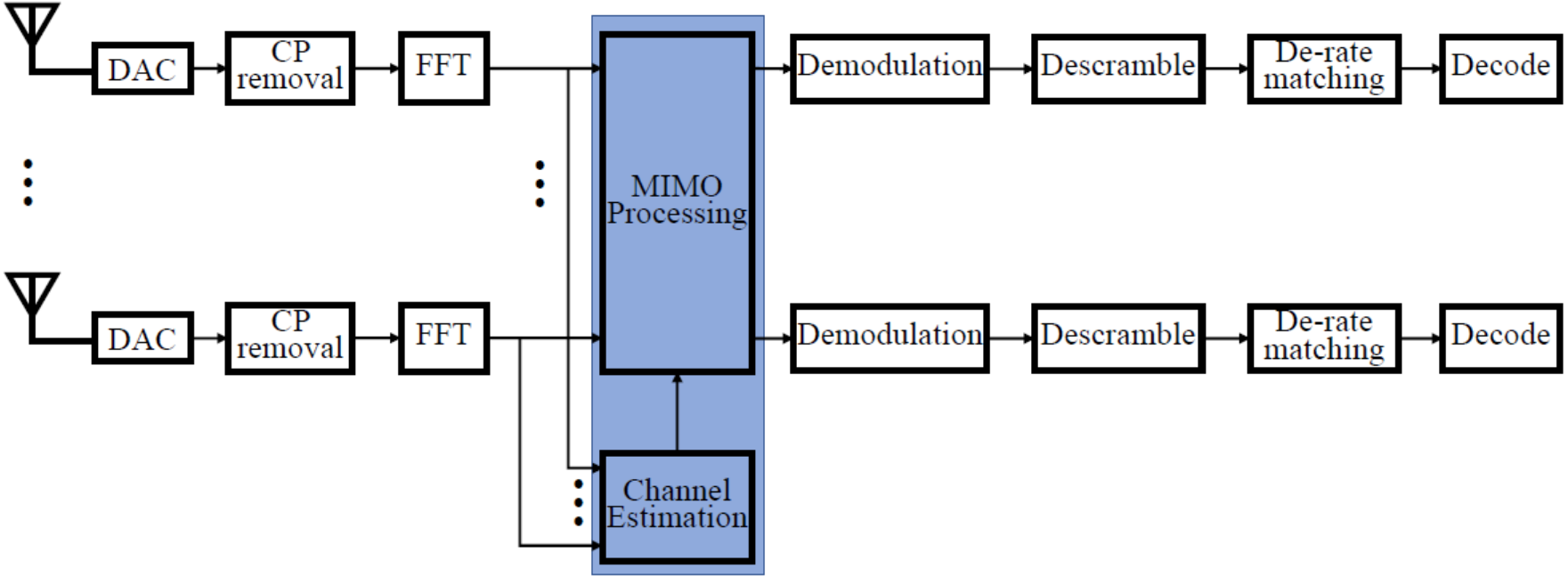
Reconfigurable and GPU Computing Laboratory (RCL)
https://www.northeastern.edu/rcl/



Publications

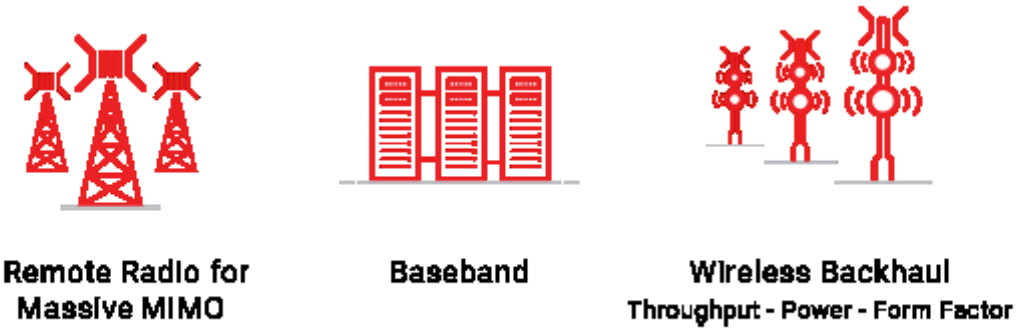
- Jieming Xu's PhD: Accelerating Matrix Processing for MIMO Systems. Available: https://repository.library.northeastern.edu/files/neu:m046v333b
- Mohamed, M., Handagala, S., Xu, J., Leeser, M. and Onabajo, M., 2020. Strategies and demonstration to support multiple wireless protocols with a single RF front-end. *IEEE Wireless Communications*, *27*(3), pp.88-95.
- Xu, J. and Leeser, M., 2018, September. High-level and compact design of cross-channel LTE downlink channel encoder. In *International Conference on Cognitive Radio Oriented Wireless Networks* (pp. 15-24). Springer.

MIMO Receiver Processing (5G)



Hardware: Xilinx RFSoC ZCU 111

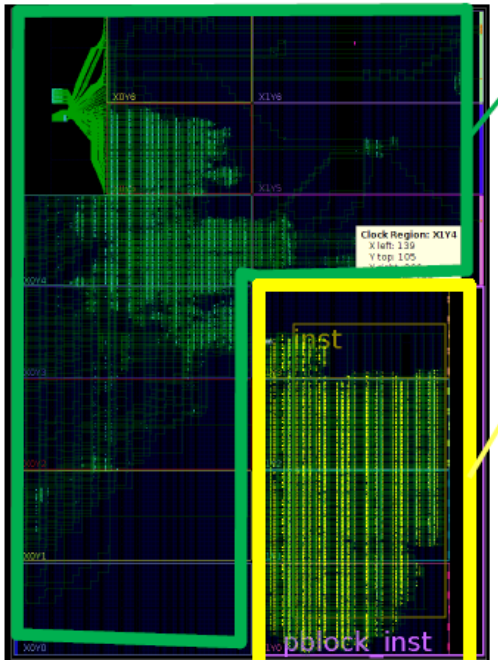
- Integrates multi gigasample RF data converters and soft decision forward error correction (SD FEC) into an SoC architecture.
  - Optimal millimeter wave IF implementations.
  - Device variants with integrated LDPC SD FEC cores and high DSP density for 5G baseband
  - Up to 6GHz of direct RF bandwidth for 5G New Radio (5G NR) support



MIMO Related Calculations

- MIMO channel equalization
  - Power compensation
  - MIMO decoding
  - Beamforming
  - …
- All matrix computations
  - QR, SVD, matrix multiplication

Partial Reconfiguration



Configuration Speed

	PCAP	ICAP
Processor	400MB/s	--
DMA	--	82.1MB/s
BRAM	--	332.1MB/s

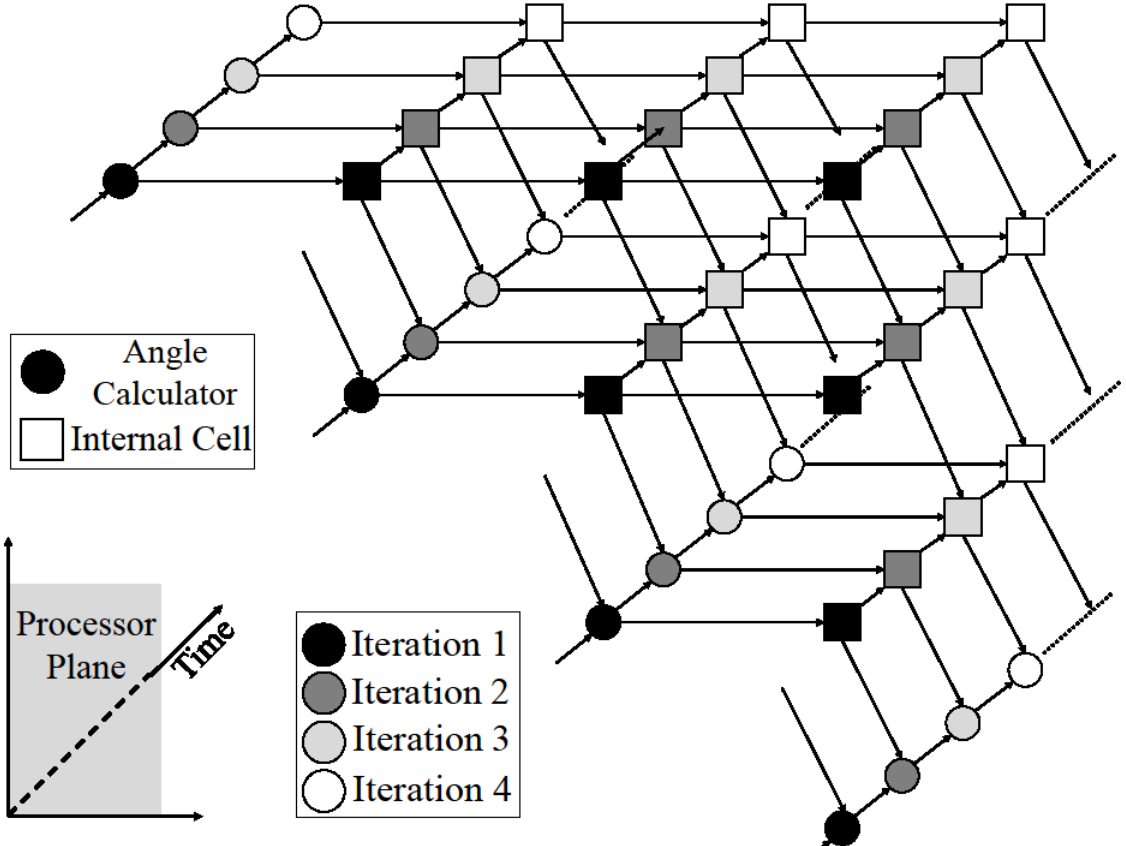
PR_time = 7.5MB / 400 = 18.75ms

Almost 2 frames of LTE signal !!

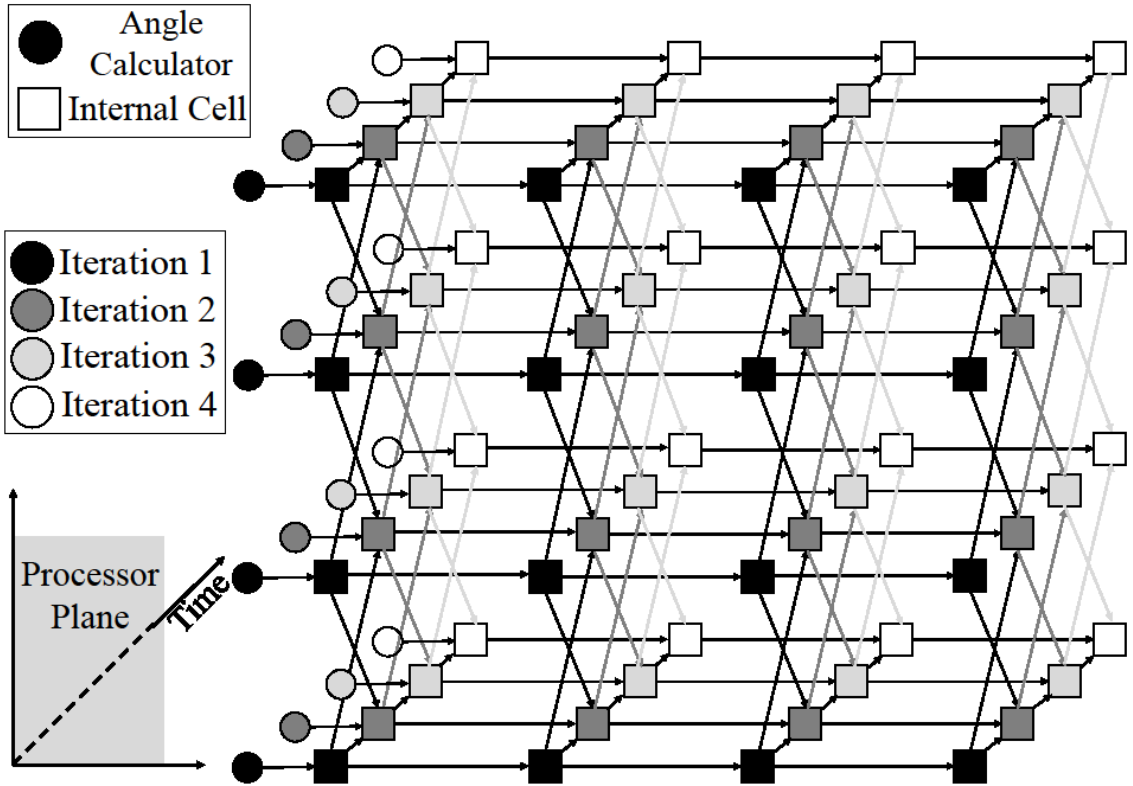
Contributions

- Unified matrix multiplication, QR decomposition, SVD and in the same systolic array
  - Enables large arrays to be supported in hardware
- System is controlled by feeding instructions from an external coprocessor:
  - Low latency for switching between applications -- just change instructions
- Two different arithmetic formats :
  - Floating-point design can process data with a large dynamic range
  - Fixed-point design uses fewer resources and is faster
- System is designed in Simulink
  - Code automation can incorporate floating-point IP cores from any source
- Designs presented have been implemented on Xilinx RFSoC ZCU 111

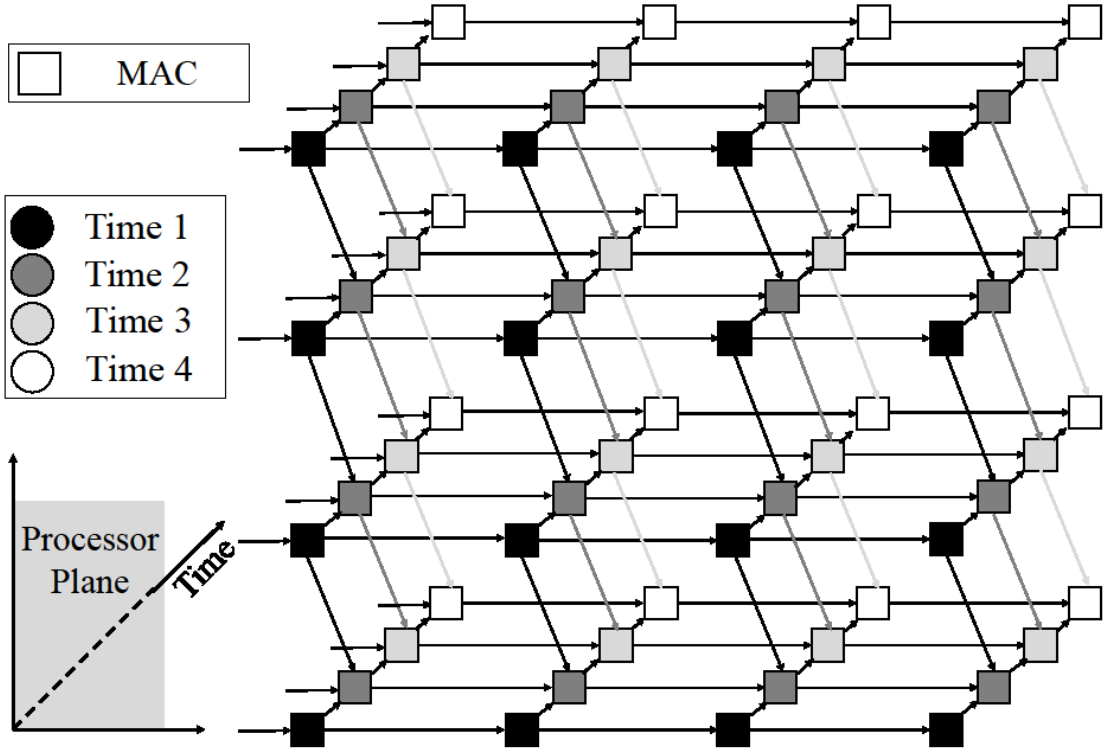
QR Dependence Graph



SVD Dependence Graph



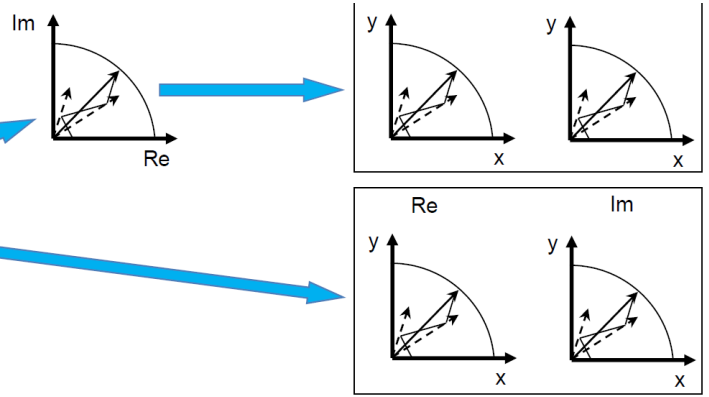
Matrix Multiply DG



Calculation of SVD and QR

- One-sided Jacobi method for SVD
- Givens rotation for QR

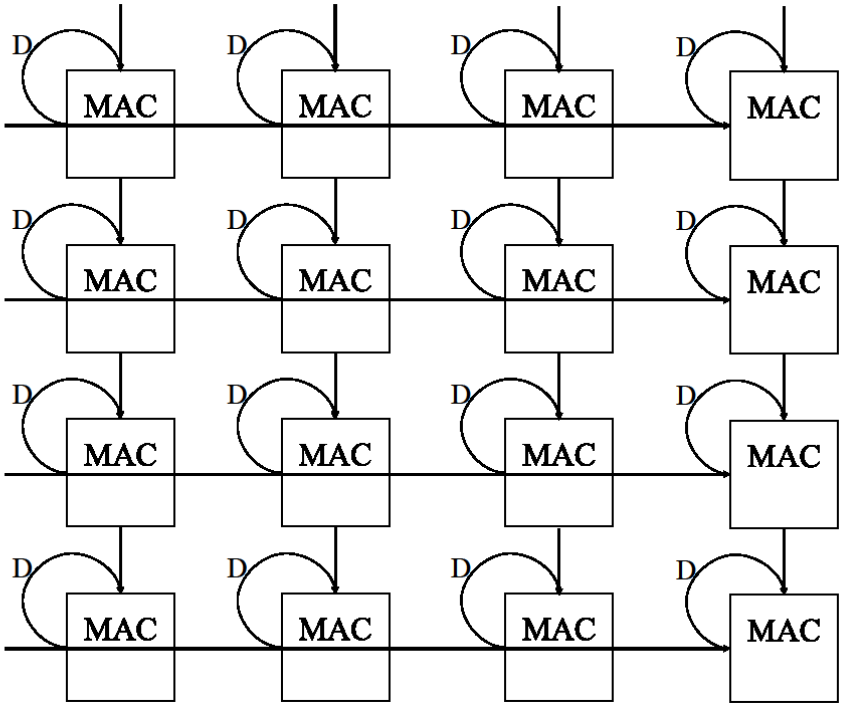
$$\begin{bmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2} \\ -\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}$$



$$\begin{cases} \phi = \text{angle}(c), \theta = \arctan\left(\dfrac{b-a}{|c|}\right) & \text{SVD} \\ \phi = \text{angle}(x_1), \theta = -2\arctan\left(\dfrac{x_1}{y_1}\right) & \text{QR} \end{cases}$$

$$a = \boldsymbol{X}^T\boldsymbol{X}, b = \boldsymbol{Y}^T\boldsymbol{Y}, c = \boldsymbol{X}^T\boldsymbol{Y}$$

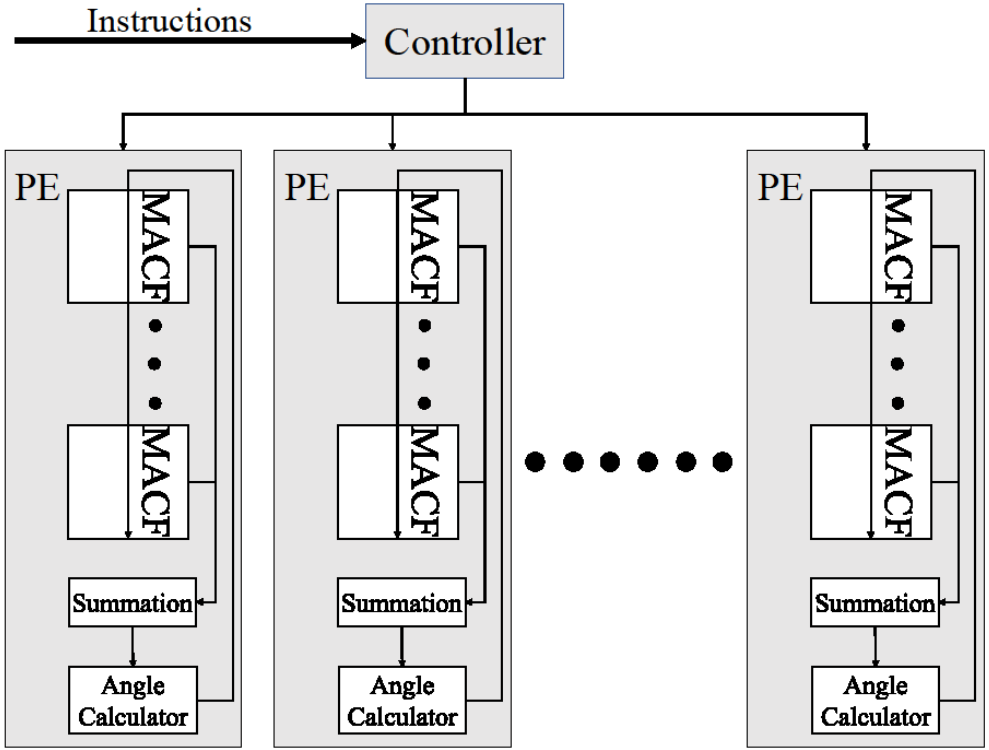
Matrix Multiply(semi-broadcast)



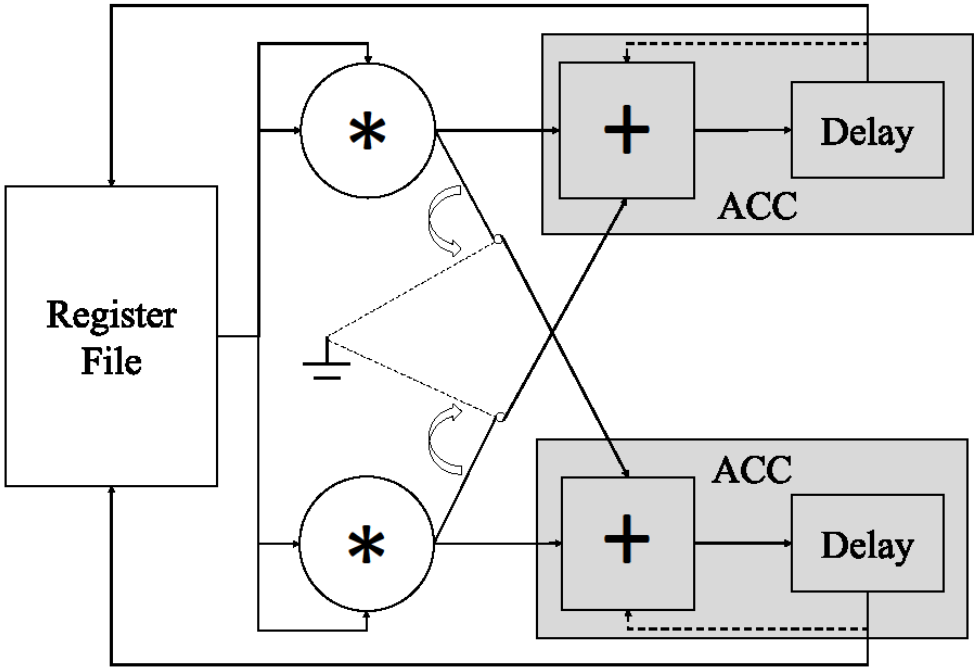
Unified Systolic Array

	Operation	Data Flow Directions	Broadcast Data	Pipeline Data	PE Usage
SVD	Rotate Multiply	Two	Rotation Angle	Matrix Element	Full
QR	Rotate	One	Rotation Element	Matrix Element	Half
MM	Multiply	One	Matrix Element	Matrix Element	Full

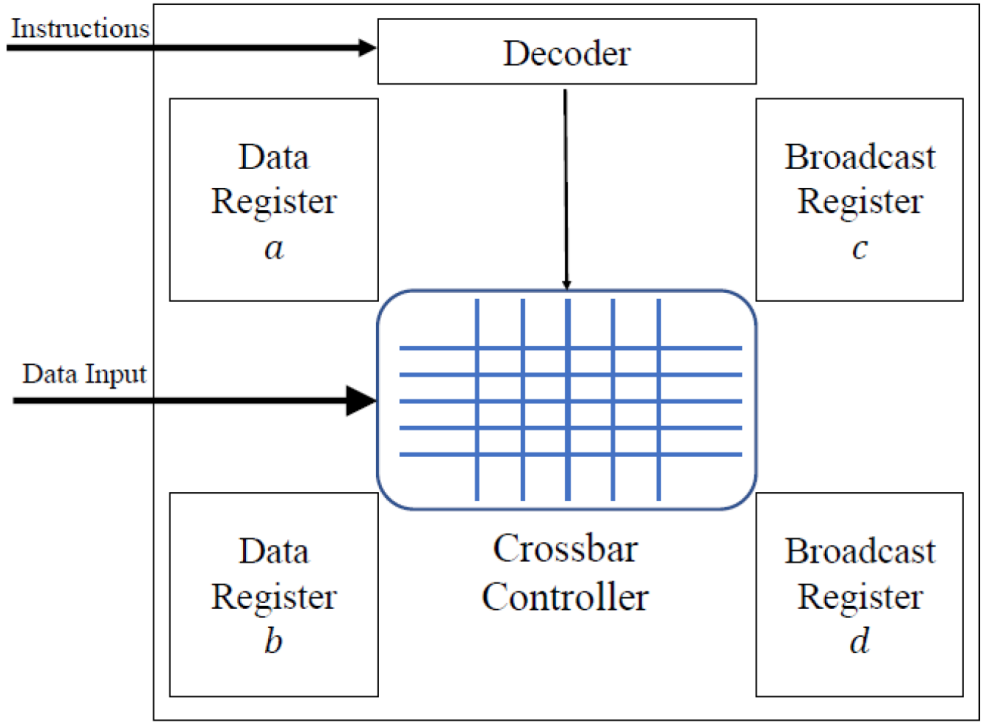
System Structure



MAC unit with register file



Register File



- Matrix multiplication
- Vector projection
- Rotation

Commonly Used Instructions

Load Operation	Load Data Registers
	Load Broadcast Register
Arithmetic Operation	Matrix Multiplication
	Plane Rotation
	Diagonal Multiplication
Data Flow Operation	Matrix Multiplication Flow
	SVD Flow
	QR Decomposition Flow

Example Program: One-sided Jacobi SVD

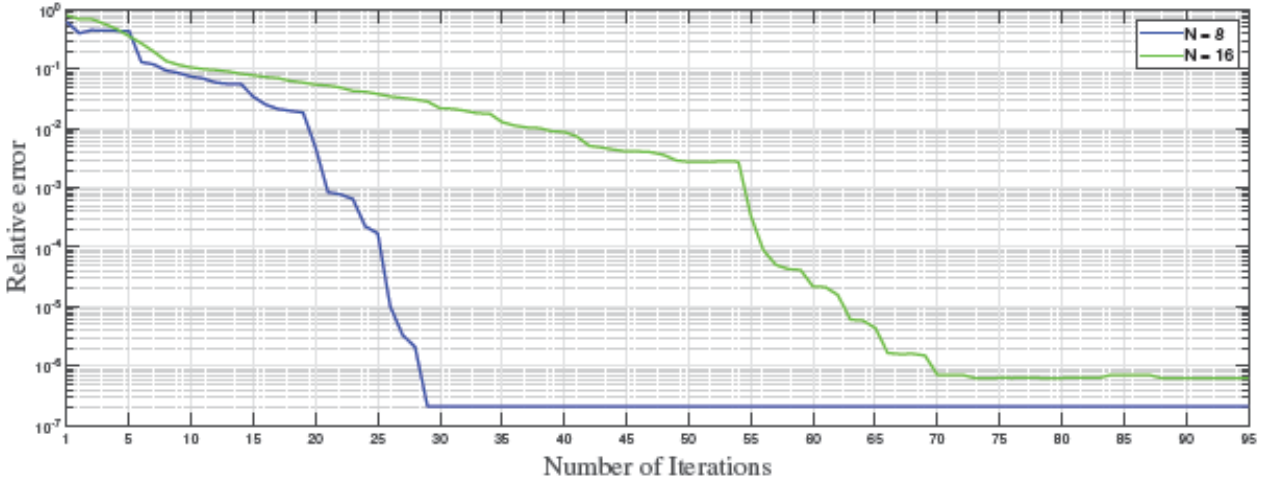
```
        LOAD_DATA           # load matrix to data reg
…
        LOAD_BROADCAST      # data to broadcast reg
        IP_CAL              # compute inner product
LP:     LOAD_BROADCAST      # data to broadcast reg
        CP_CAL              # compute cross product

wait for angle calculator …   completion signal triggered

        LOAD_BROADCAST      # load complex angle value
        ROTC_CAL            # complex angle rotation
        LOAD_BROADCAST      # load Givens rotation value
        ROT0_CAL             # rotate the first row
        ROT1_CAL            # rotate the second row
        LOAD_DATA           # load rotated data
        INT_DATA            # swap data between PEs
        JP LP               # jump to LP … until convergence
```

Arithmetic

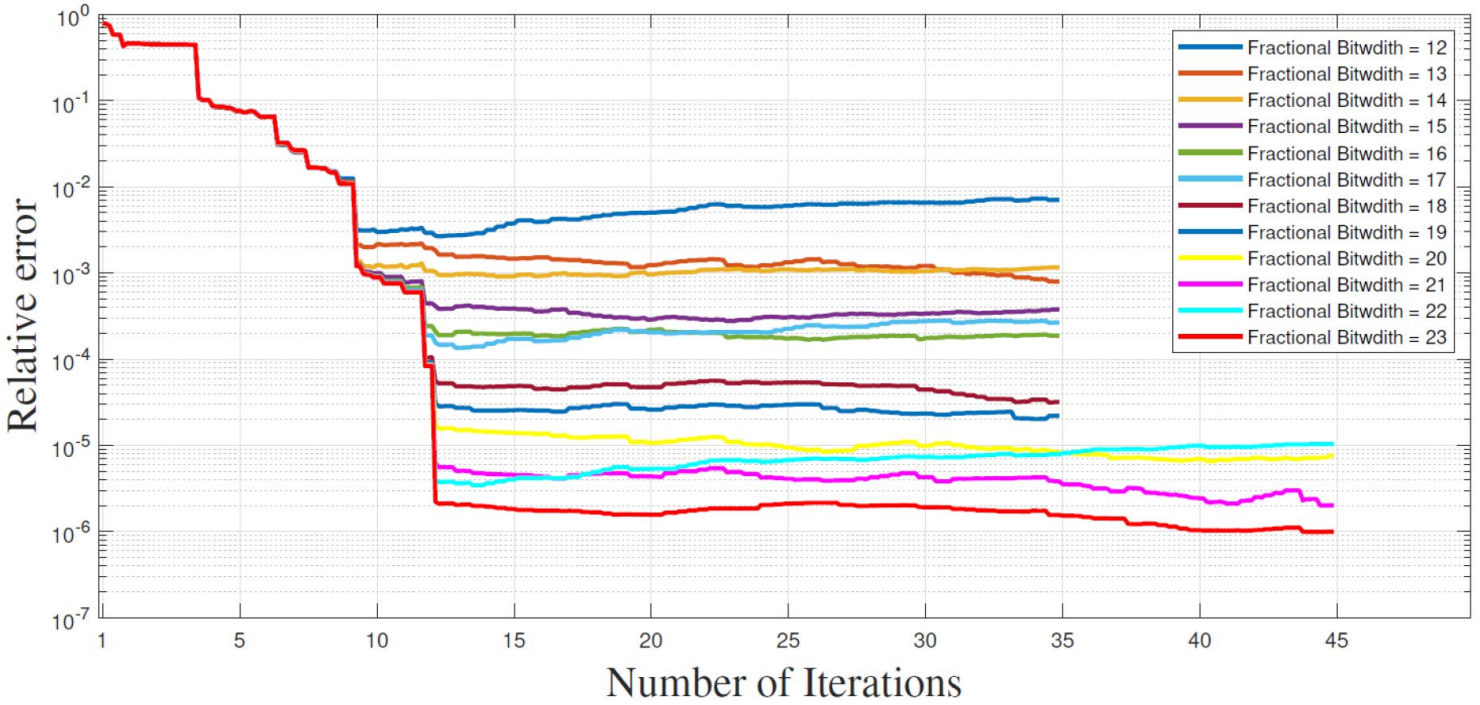
- Choice of floating point or fixed point when building the hardware design
- Floating Point:
  - IEEE Single precision using Xilinx IP cores
  - Easy to substitute another IP library
- Fixed point:
  - 8 × 8 version uses 25 bits, 18 fractional
  - 16 × 16 version uses 27-bit fixed-point

Relative Error Floating Point

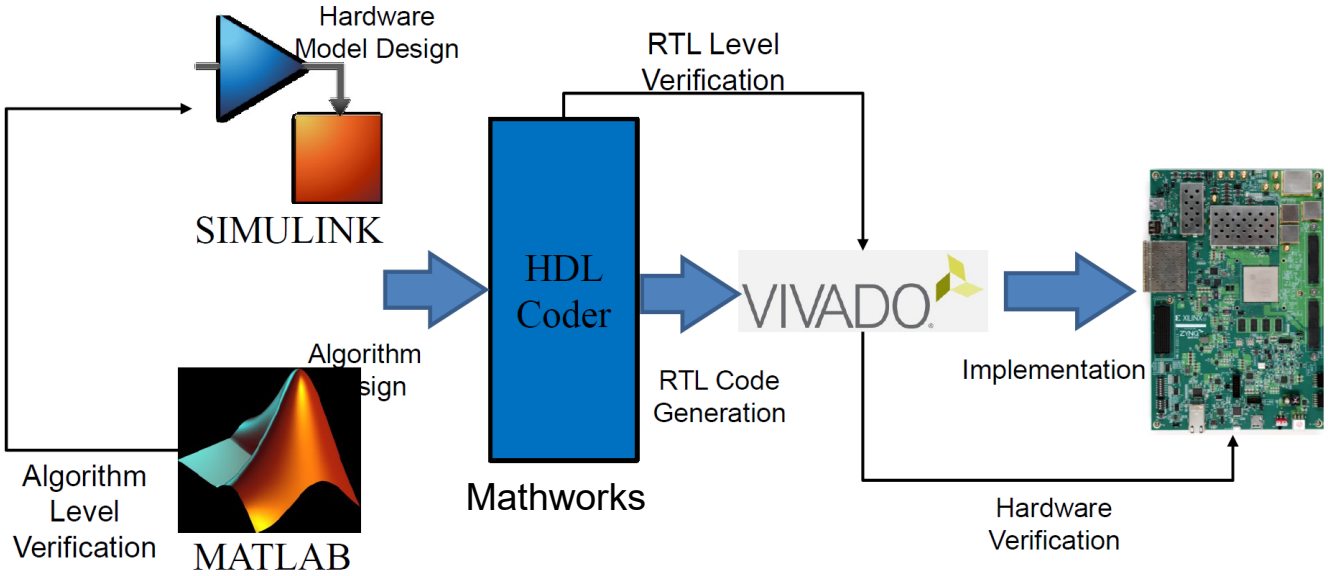
- SVD 8 x 8 and 16 x 16 designs
  - Compared to MATLAB SVD (double precision floats)



Relative Error: 8x8 Matrix Mult



Tool Flow


Hardware
Model Design
SIMULINK
RTL Level
Verification
HDL
Coder
VIVADO
Implementation
Algorithm
Level
Verification
MATLAB
RTL Code
Generation
Mathworks
Hardware
Verification

Hardware Performance

		SVD Iteration Clock Cycles	QR Rotation Clock Cycles	Matrix Mul Clock Cycles
8x8	Fixed Point	159	159	3
	Floating Point	383	265	11
16x16	Fixed Point	169	169	3
	Floating Point	399	265	11

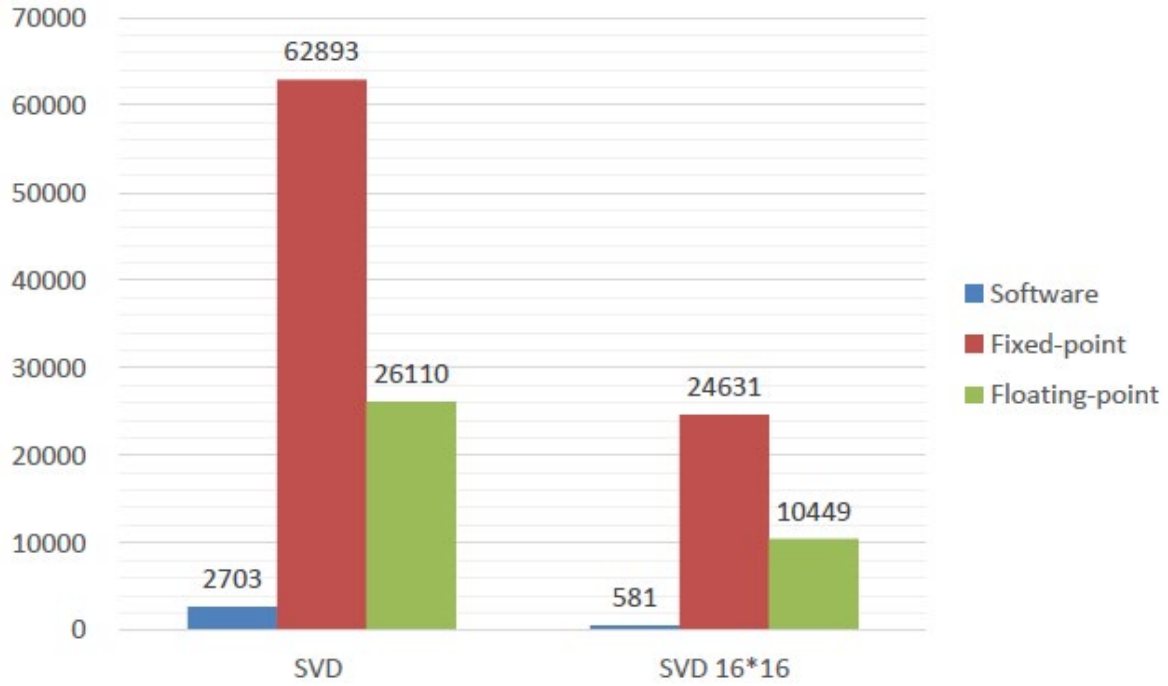
Resources and Clock Speed

		LUT	FF	LUTRAM	DSP	Clock (MHz)
8x8	Fixed Point	9.75%	5.8%	0.1%	6.2%	380
	Floating Point	21.5%	18%	2.3%	17.6%	374
16x16	Fixed Point	33.7%	20.6%	0.2%	24.7%	250
	Floating Point	72.5%	61.5%	9.9%	65.5%	250

HW vs SW time in seconds

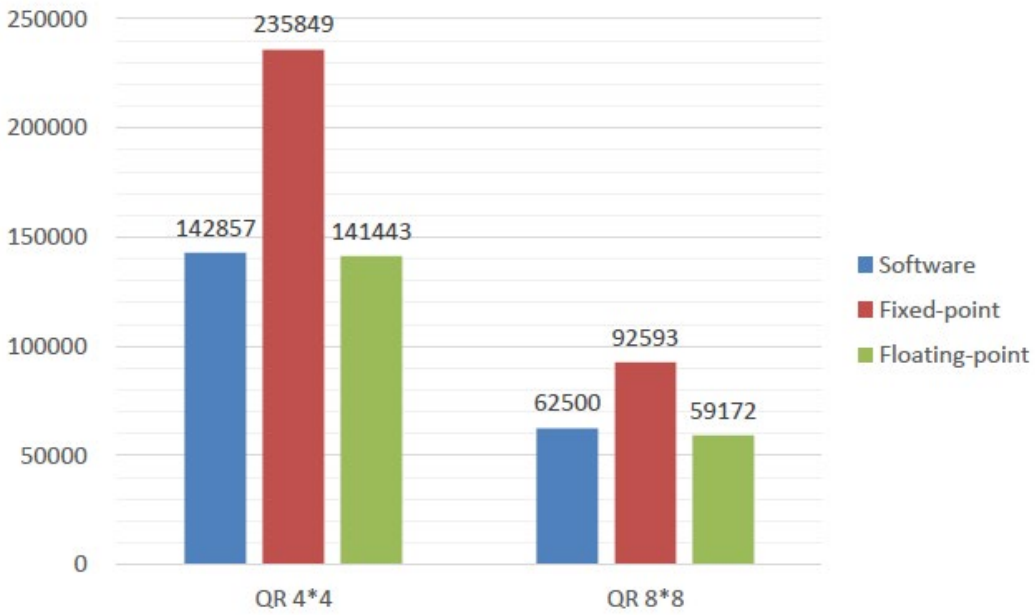
	SW	HW	
		Fixed	Float
SVD 8x8	3.7×10^{-4}	1.6×10^{-5}	3.8×10^{-5}
SVD 16x16	1.7×10^{-3}	4×10^{-5}	9.6×10^{-5}
QR 4x4	7×10^{-6}	4.25×10^{-6}	7.1×10^{-6}
QR 8x8	1.6×10^{-5}	1.1×10^{-5}	1.6×10^{-5}

SVDs per second



Northeastern

QRs per second



- Not as good as SVD. Only using half the array

Conclusions

- SVD, QR and Matrix Multiply implemented on a single FPGA design
- Change function by changing instructions
  - Low latency switching between tasks
- Can support many different wireless and MIMO tasks on the same FPGA
- Partial reconfiguration is NOT an option:
  - Bitstream for 8x8 design is 33.5 Mb, 60 ms to reconfigure

Future Directions

- Currently, hardware is implemented on the FPGA fabric of an RFSoc
- Next steps
  - Connect to the RF Frontend
  - Demonstrate MIMO applications and processing chain

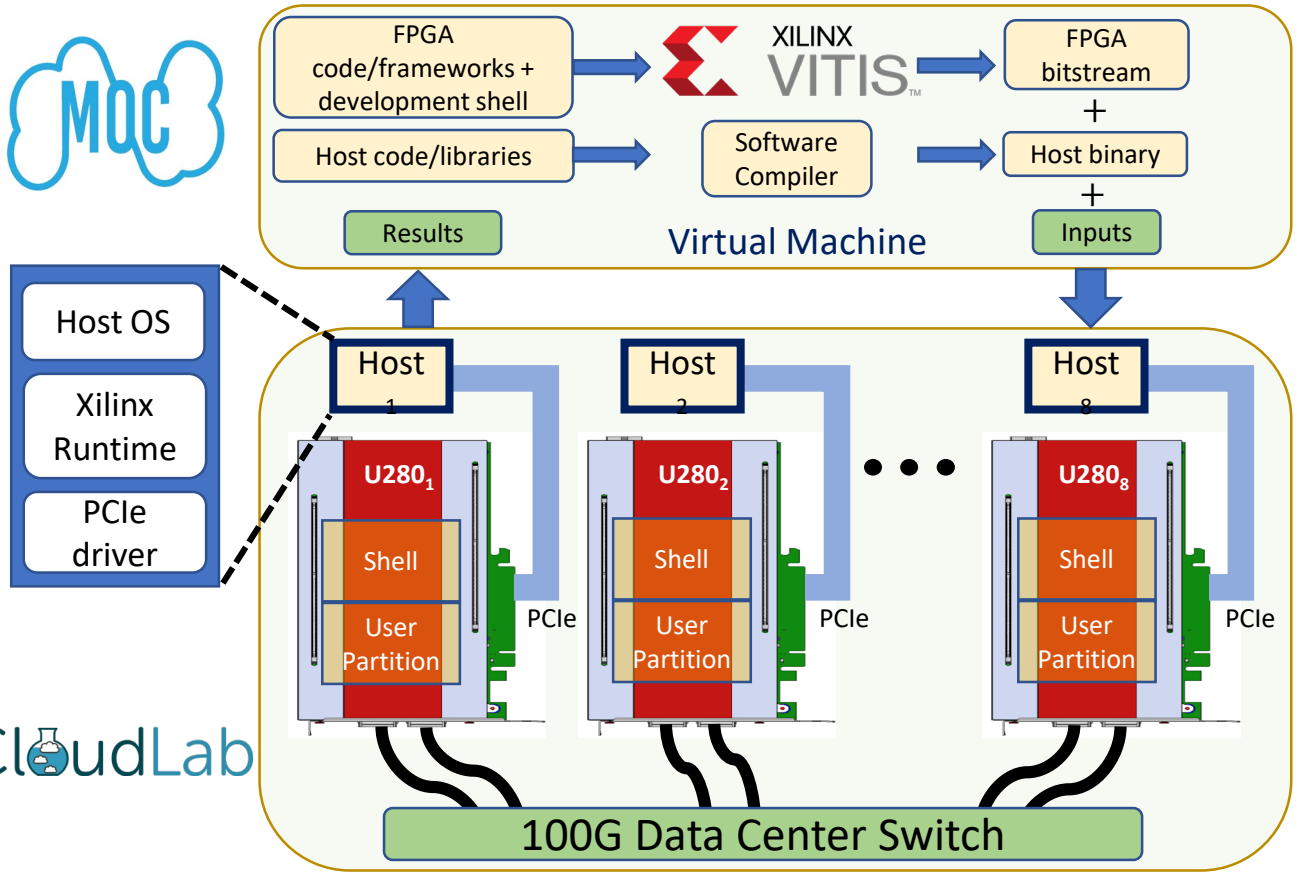
Publications and Thank You

- Jieming Xu's PhD: Accelerating Matrix Processing for MIMO Systems. Available: https://repository.library.northeastern.edu/files/neu:m046v333b
- Mohamed, M., Handagala, S., Xu, J., Leeser, M. and Onabajo, M., 2020. Strategies and demonstration to support multiple wireless protocols with a single RF front-end. *IEEE Wireless Communications*, *27*(3), pp.88-95.
- Xu, J. and Leeser, M., 2018, September. High-level and compact design of cross-channel LTE downlink channel encoder. In *International Conference on Cognitive Radio Oriented Wireless Networks* (pp. 15-24). Springer, Cham.

https://www.northeastern.edu/rcl/

- Thank you:

Open Cloud Testbed: Edge to Cloud



- Tools run on MOC
- Eight Alveo U280s in Cloudlab
- Both MOC and Cloudlab are in MGHPCC
- User accounts will be available later this summer

NSF