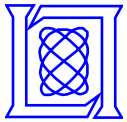
GPU Accelerated Decoding of High Performance Error Correcting Codes

Andrew D. Copeland, Nicholas B. Chang, and Stephen Leung

This work is sponsored by the Department of the Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

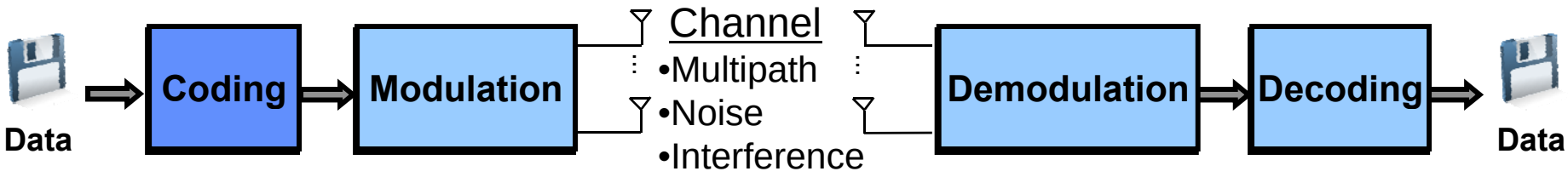


Outline

- **Introduction**
  - Wireless Communications
  - Performance vs. Complexity
  - Candidate Implementations
- Low Density Parity Check (LDPC) Codes
- NVIDIA GPU Architecture
- GPU Implementation
- Results
- Summary

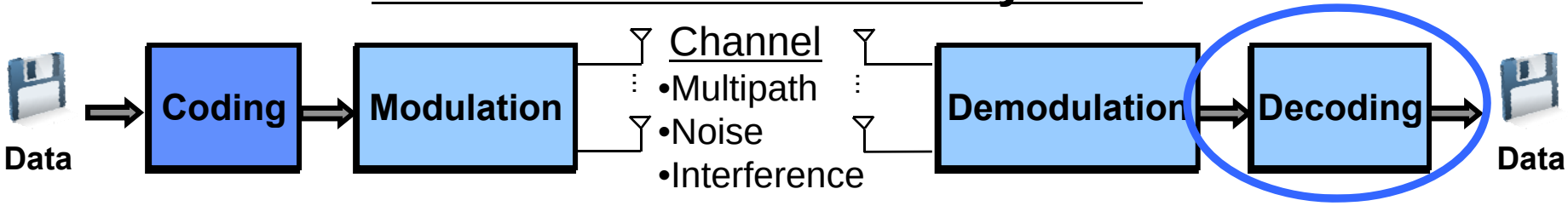
Wireless Communications

Generic Communication System



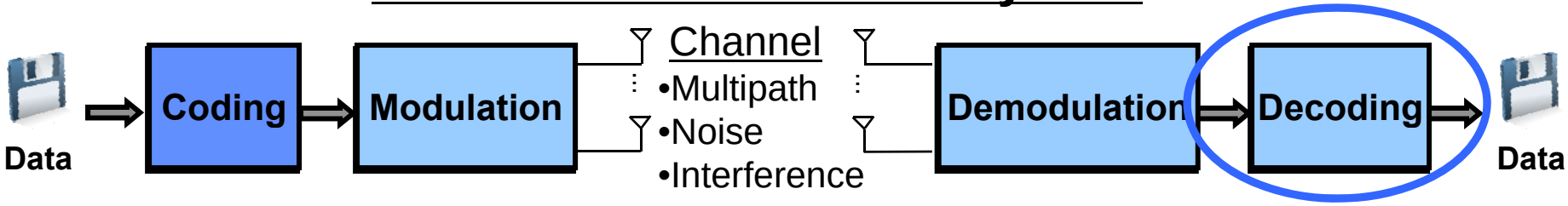
Wireless Communications

Generic Communication System

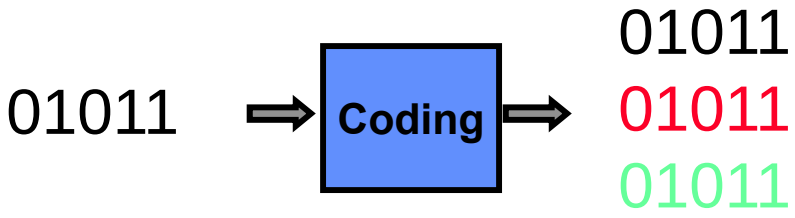


Wireless Communications

Generic Communication System

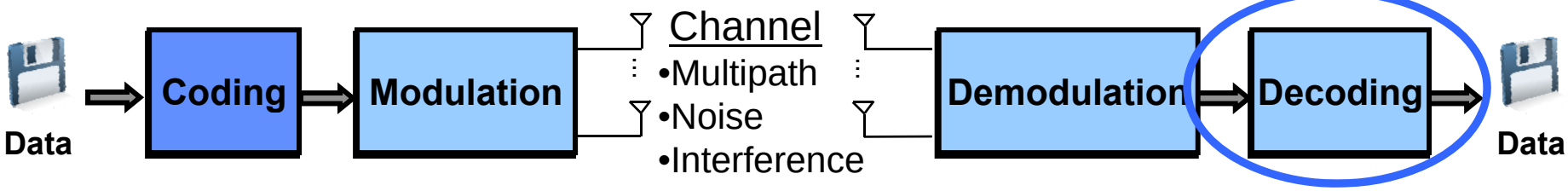


Simple Error Correcting Code

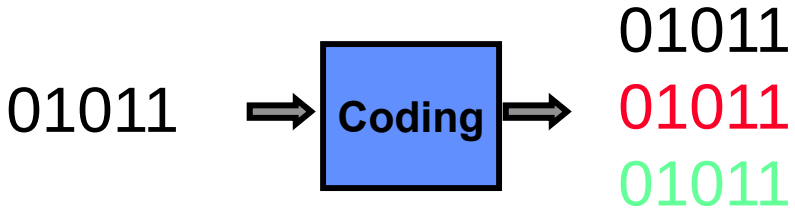


Wireless Communications

Generic Communication System

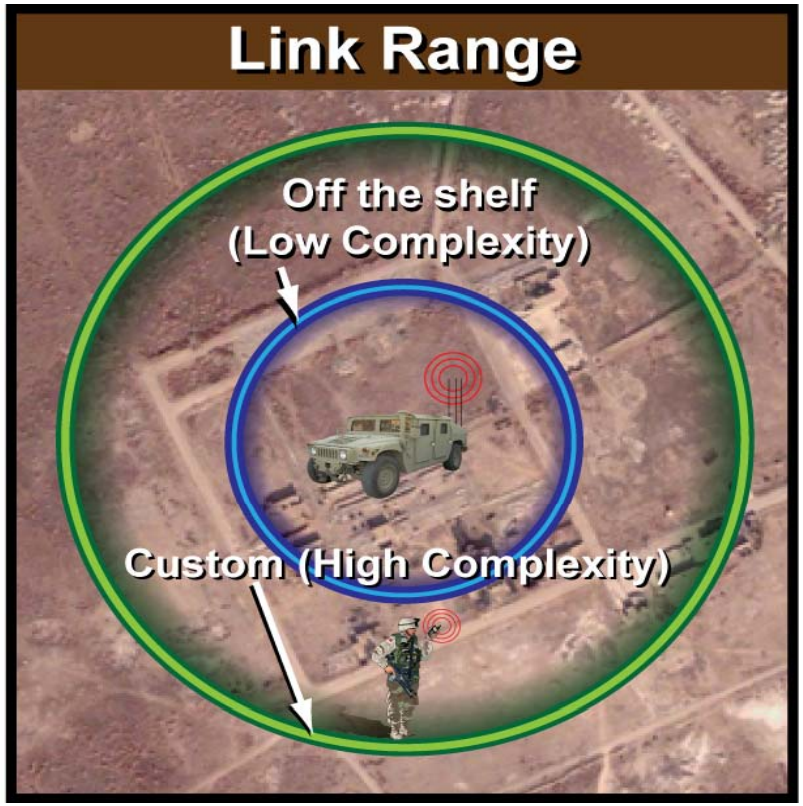


Simple Error Correcting Code



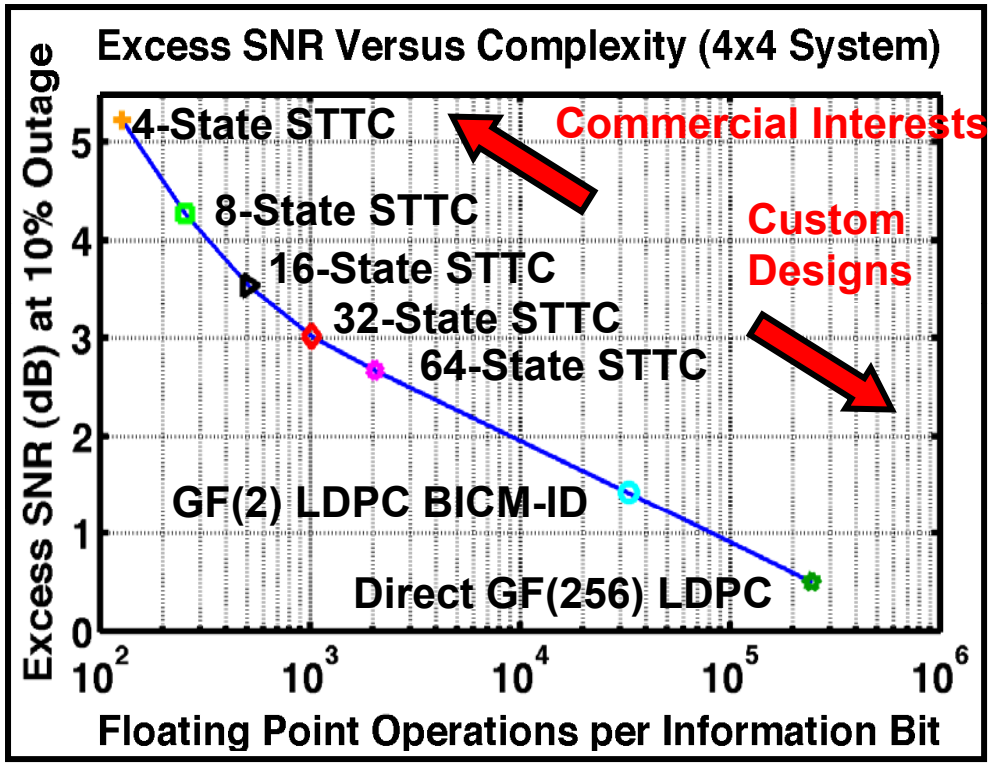
High Performance Codes

- Allow for larger link distance and/or higher data rate
- Superior performance comes with significant increase in complexity



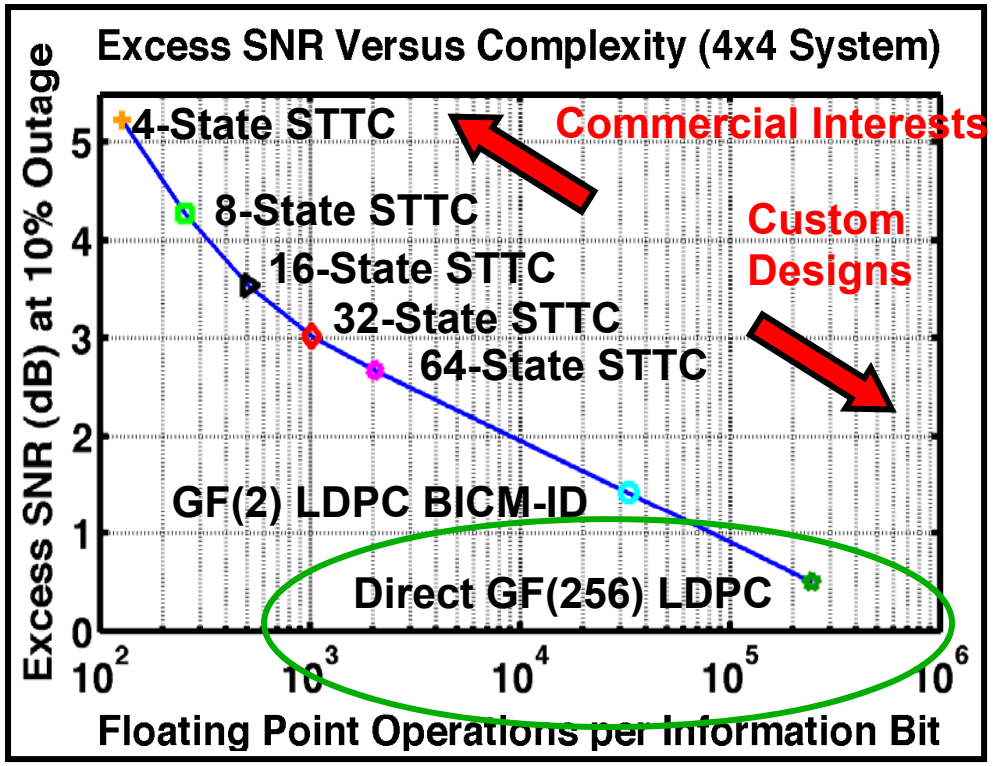
Performance vs. Complexity

- Increased performance requires larger computational cost
- Off the shelf systems require 5dB more power to close link than custom system
- Best custom design only require 0.5dB more power than theoretical minimum



Performance vs. Complexity

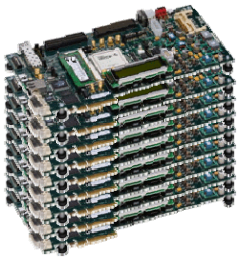
- Increased performance requires larger computational cost
- Off the shelf systems require 5dB more power to close link than custom system
- Best custom design only require 0.5dB more power than theoretical minimum



Our Choice

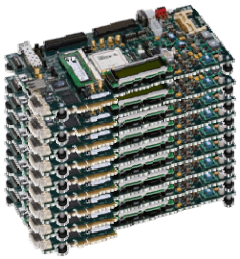
Candidate Implementations

- Half second burst of data
  - Decode complexity 37 TFLOPS
  - Memory transferred during decode 44 TB

	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)				
Performance (decode time)				
Hardware Cost				

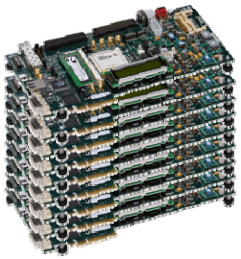
Candidate Implementations

- Half second burst of data
  - Decode complexity 37 TFLOPS
  - Memory transferred during decode 44 TB

	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)	-			
Performance (decode time)	9 days			
Hardware Cost	≈ $2,000			

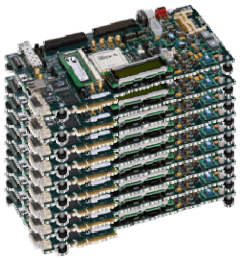
Candidate Implementations

- Half second burst of data
  - Decode complexity 37 TFLOPS
  - Memory transferred during decode 44 TB

	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)	-	1 person year (est.)		
Performance (decode time)	9 days	5.1 minutes (est.)		
Hardware Cost	≈ $2,000	≈ $90,000		

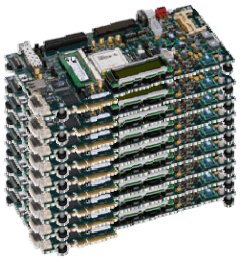
Candidate Implementations

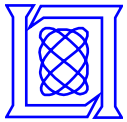
- Half second burst of data
  - Decode complexity 37 TFLOPS
  - Memory transferred during decode 44 TB

	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)	-	1 person year (est.)	1.5 person years (est.) +1 year to Fab.	
Performance (decode time)	9 days	5.1 minutes (est.)	0.5s (if possible)	
Hardware Cost	≈ $2,000	≈ $90,000	≈ $1 million	

Candidate Implementations

- Half second burst of data
  - Decode complexity 37 TFLOPS
  - Memory transferred during decode 44 TB

	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)	-	1 person year (est.)	1.5 person years (est.) +1 year to Fab.	?
Performance (decode time)	9 days	5.1 minutes (est.)	0.5s (if possible)	?
Hardware Cost	≈ $2,000	≈ $90,000	≈ $1 million	≈ $5,000 (includes PC cost)



Outline

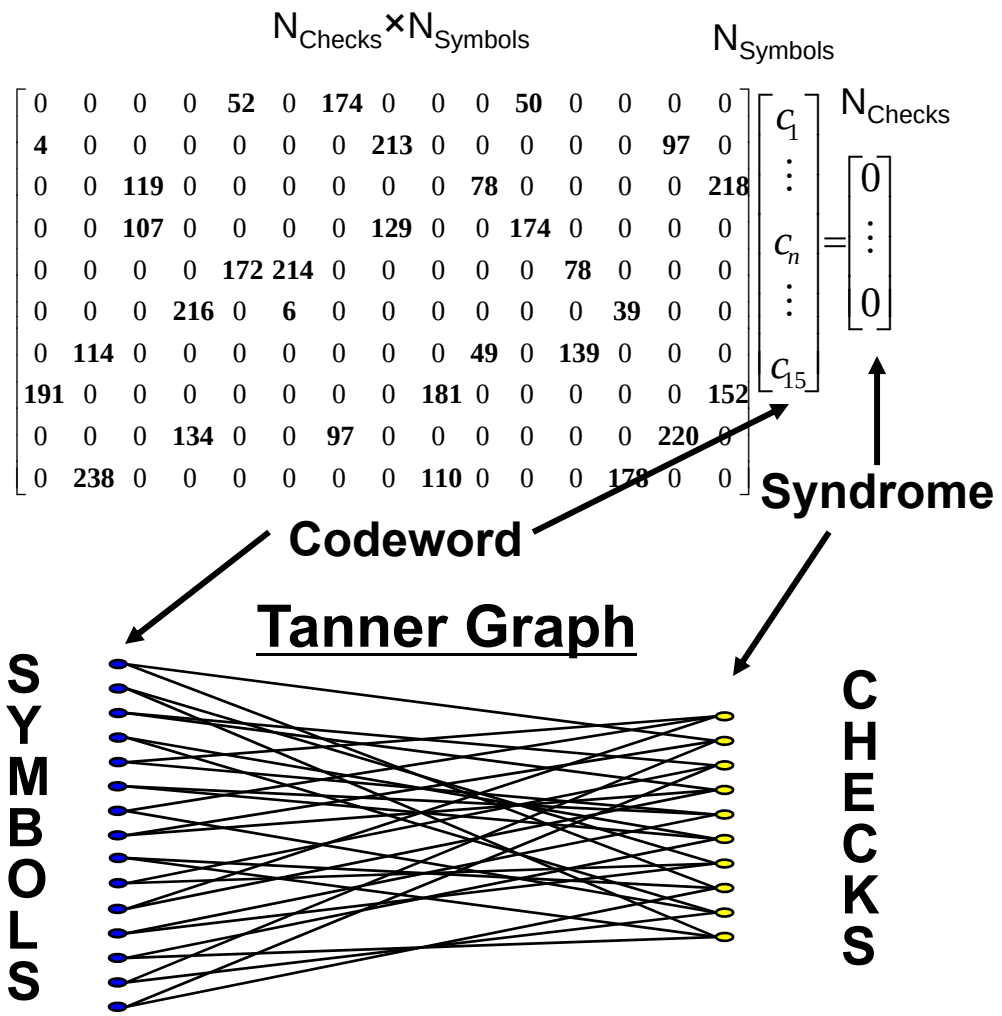
- Introduction
- **Low Density Parity Check (LDPC) Codes**
  - **Decoding LDPC GF(256) Codes**
  - **Algorithmic Demands of LDPC Decoder**
- NVIDIA GPU Architecture
- GPU Implementation
- Results
- Summary

Low Density Parity Check (LDPC) Codes

LDPC Codes

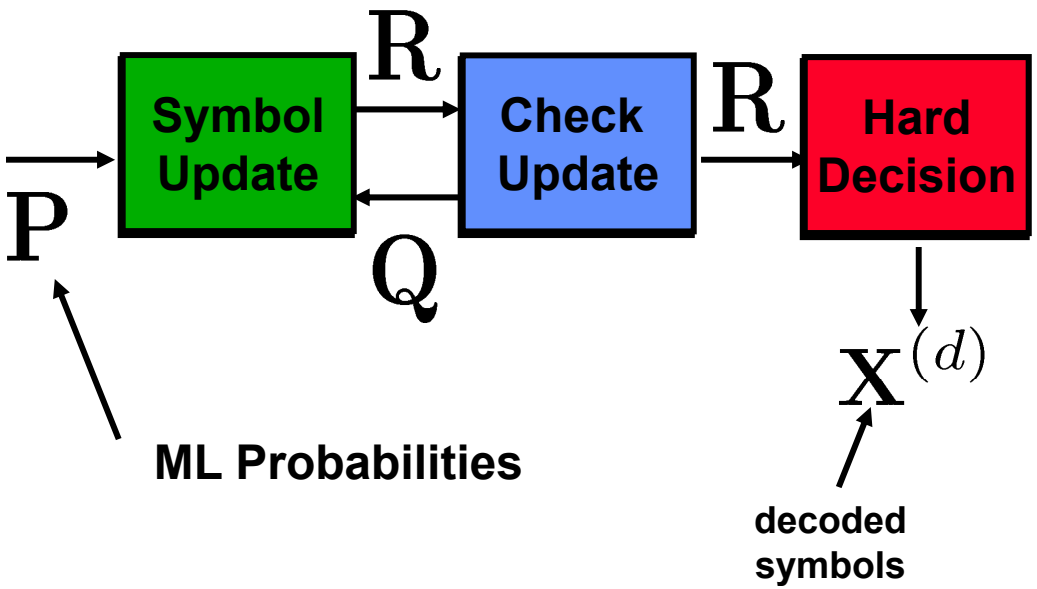
- Proposed by Gallager (1962)
  - codes were binary
- Not used until mid 1990s
- GF(256) Davey and McKay (1998)
  - symbols in [0, 1, 2, …, 255]
- Parity check matrix defines graph
  - Symbols connected to checks nodes
- Has error state feedback
  - Syndrome = 0 (no errors)
  - Potential to re-transmit

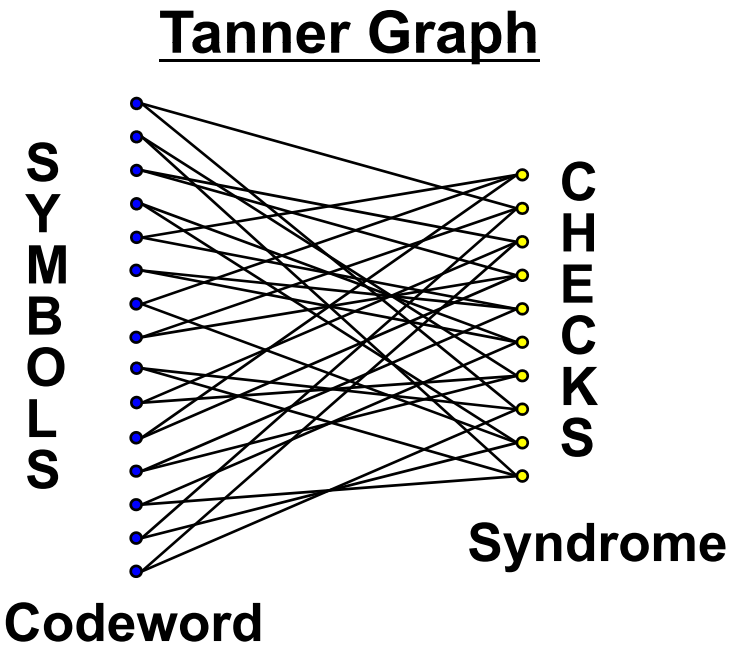
LDPC GF(256) Parity-Check Matrix



Decoding LDPC GF(256) Codes

- Solved using LDPC Sum-Product Decoding Algorithm
  - A belief propagation algorithm
- Iterative Algorithm
  - Number of iterations depends on SNR





Algorithmic Demands of LDPC Decoder

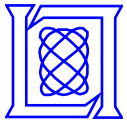
Computational Complexity

- Decoding of a entire sequence **37 TFLOPS**
  - Single frame 922 MFLOPS
  - Check update 95% of total

Memory Requirements

- Data moved between device memory and multiprocessors **44.2 TB**
  - Single codeword 1.1 GB
- Data moved within Walsh Transform and permutations (part of check update) **177 TB**
  - Single codeword 4.4 GB

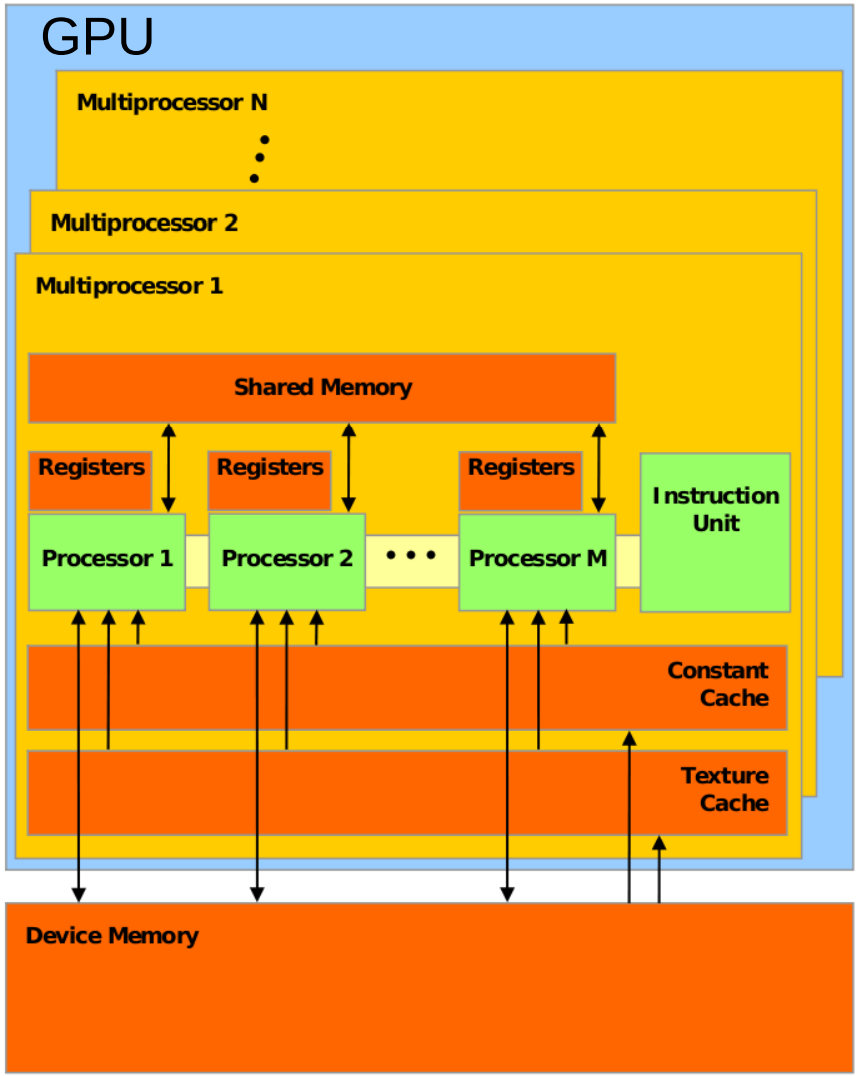
Decoder requires architectures with high computational and memory bandwidths



Outline

- Introduction
- Low Density Parity Check (LDPC) Codes
- **NVIDIA GPU Architecture**
  - **Dispatching Parallel Jobs**
- GPU Implementation
- Results
- Summary

NVIDIA GPU Architecture

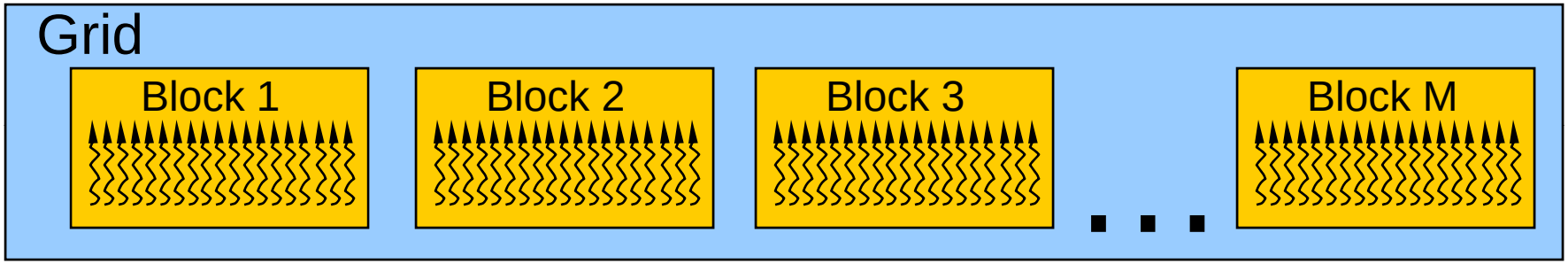


- GPU contains N multiprocessors
  - Each Performs Single Instruction Multiple Thread (SIMT) operations
- Constant and Texture caches speed up certain memory access patterns
- NVIDIA GTX 295 (2 GPUs)
  - N = 30 multiprocessors
  - M = 8 scalar processors
  - Shared memory 16KB
  - Device memory 896MB
  - Up to 512 threads on each multiprocessor
  - Capable of 930 GFLOPs
  - Capable of 112 GB/sec

NVIDIA CUDA Programming Guide Version 2.1

Dispatching Parallel Jobs

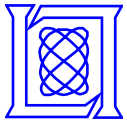
- **Program decomposed into blocks of threads**
  - **Thread blocks are executed on individual multiprocessors**
  - **Threads within blocks coordinate through shared memory**
  - **Threads optimized for coalesced memory access**



- **Calling parallel jobs (kernels) within C function is simple**
- **Scheduling is transparent and managed by GPU and API**

```
grid.x = M;  threads.x = N;
kernel_call<<<grid, threads>>>(variables);
```

Thread blocks execute independently of one another
Threads allow faster computation and better use of memory bandwidth
Threads work together using shared memory



Outline

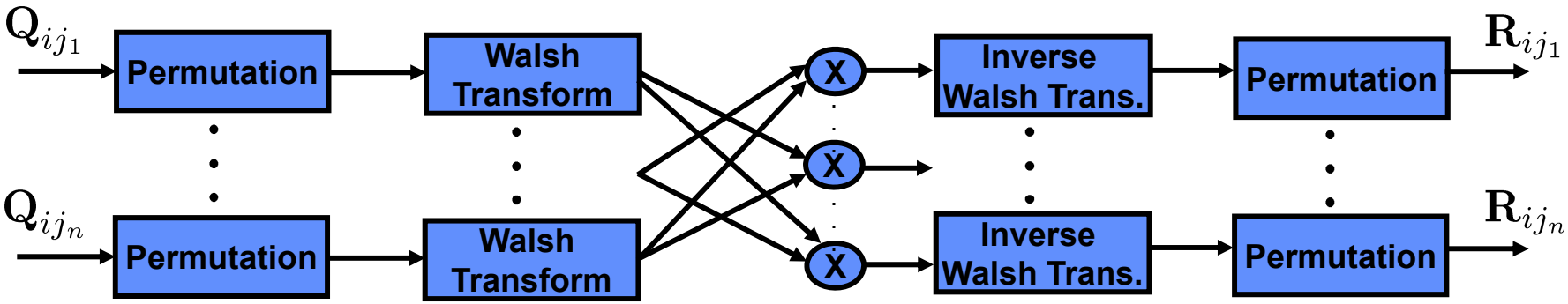
- Introduction
- Low Density Parity Check (LDPC) Codes
- NVIDIA GPU Architecture
- **GPU Implementation**
  - **Check Update Implementation**
- Results
- Summary

GPU Implementation of LDPC Decoder

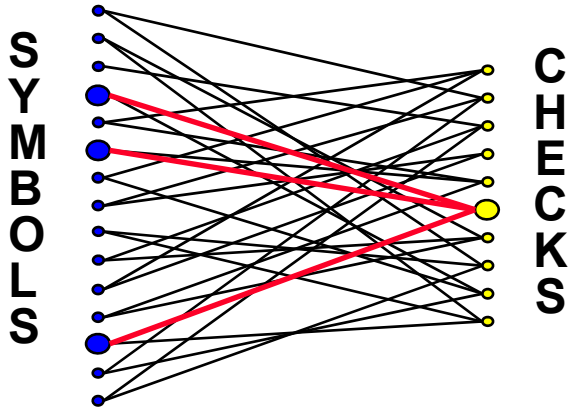
- Quad GPU Workstation
  - 2 NVIDIA GTX 295s
  - 3.20GHz Intel Core i7 965
  - 12GB of 1330MHz DDR3 memory
  - Costs $5000

- Software written using NVIDIA's Compute Unified Device Architecture (CUDA)
- Replaced Matlab functions with CUDA MEX functions
- Utilized static variables
  - GPU Memory allocated first time function is called
  - Constants used to decode many codewords only loaded once
- Codewords distributed across the 4 GPUs

Check Update Implementation

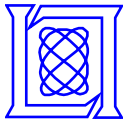


- **Thread blocks assigned to each check *independently* of all other checks**
- **Q_{ij} and R_{ij} are 256 element vectors**
  - **Each thread is assigned to element of GF(256)**
  - **Coalesced memory reads**
  - **Threads coordinate using shared memory**




```
grid.x = num_checks;
threads.x = 256;
check_update<<<grid, threads>>>(Q,R);
```

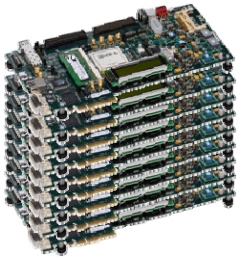
Structure of algorithm exploitable on GPU



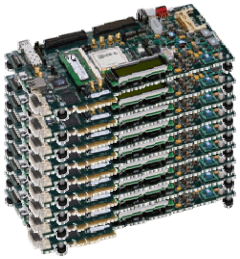
Outline

- Introduction
- Low Density Parity Check (LDPC) Codes
- NVIDIA GPU Architecture
- GPU Implementation
- **Results**
- Summary

Candidate Implementations

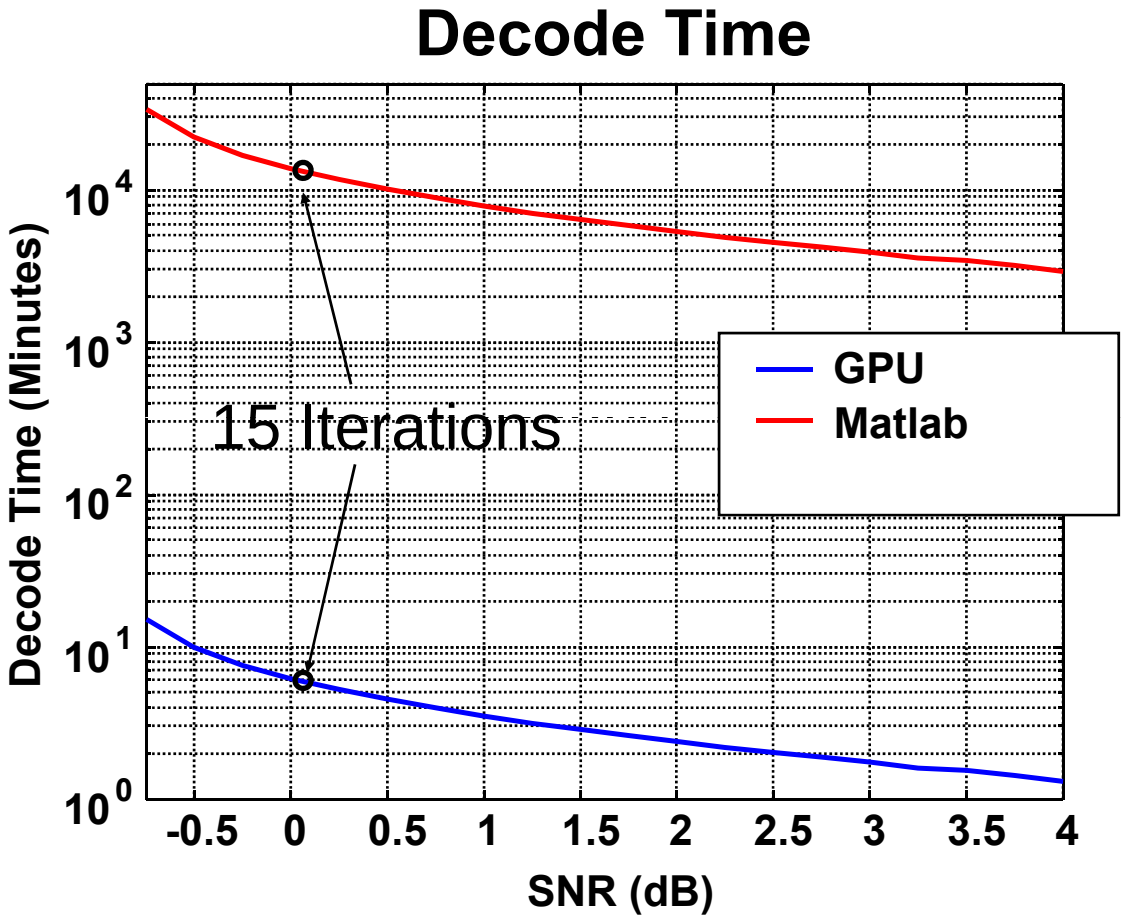
	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)	-	1 person year (est.)	1.5 person years (est.) +1 year to Fab.	?
Performance (time to decode half second of data)	9 days	5.1 minutes	0.5s (if possible)	?
Hardware Cost	≈ $2,000	≈ $90,000	≈ $1 million	≈ $5,000 (includes PC cost)

Candidate Implementations

	PC	9x FPGA	ASIC	4x GPU
				
Implementation Time (beyond development)	-	1 person year (est.)	1.5 person years (est.) +1 year to Fab.	6 person weeks
Performance (time to decode half second of data)	9 days	5.1 minutes	0.5s (if possible)	5.6 minutes
Hardware Cost	≈ $2,000	≈ $90,000	≈ $1 million	≈ $5,000 (includes PC cost)

Implementation Results

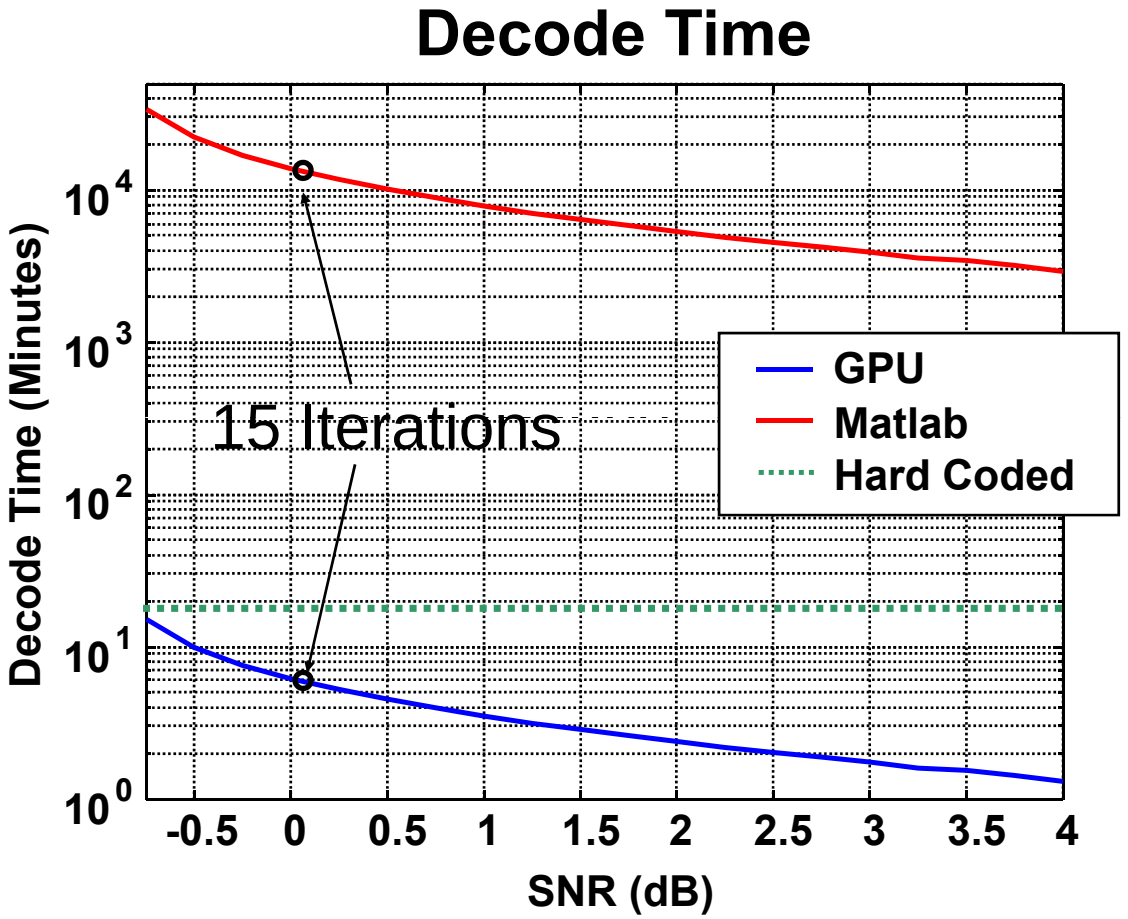
- Decoded test data in 5.6 minutes instead of 9 days
  - A 2336x speedup



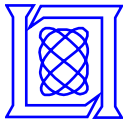
GPU implementation supports analysis, testing, and demonstration activities

Implementation Results

- Decoded test data in 5.6 minutes instead of 9 days
  - A 2336x speedup
- Hard coded version runs fixed number of iterations
  - Likely used on FPGA or ASIC versions

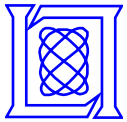


GPU implementation supports analysis, testing, and demonstration activities



Summary

- GPU architecture can provide significant performance improvement with limited hardware cost and implementation effort
- GPU implementation is a good fit for algorithms with parallel steps that require systems with large computational and memory bandwidths
  - LDPC GF(256) decoding is a well-suited algorithm
- Implementation allows reasonable algorithm development cycle
  - New algorithms can be tested and demonstrated in minutes instead of days



Backup

Experimental Setup

Test System Parameters

- Data rate 1.3 Gb/s
- Half second burst
  - 40,000 codewords (6000 symbols)
- Parity check matrix 4000 checks × 6000 symbols
  - Corresponds to rate 1/3 code
- Performance numbers based on 15 iterations of decoding algorithm