

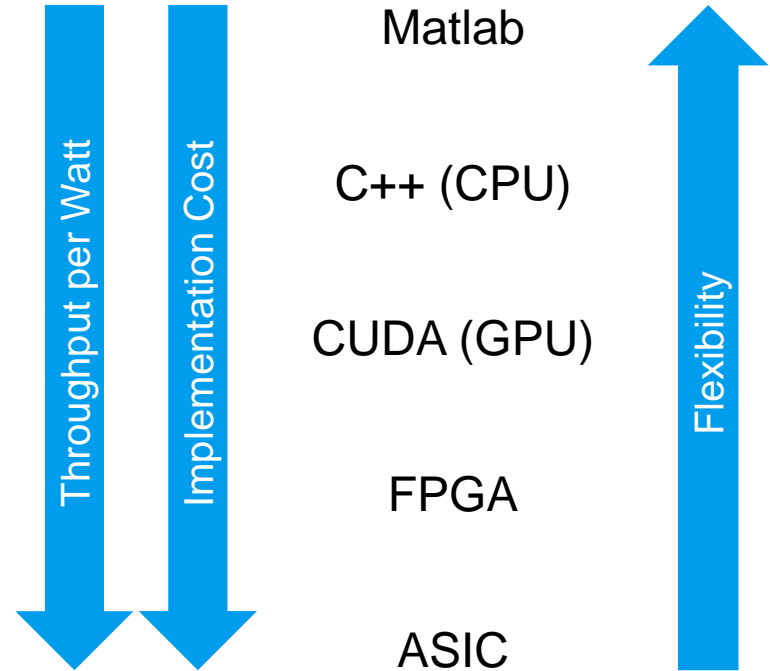
# Decoding on GPUs on the Example of Turbo Product Codes

Stefan Dierks

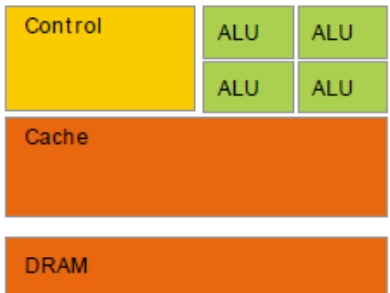
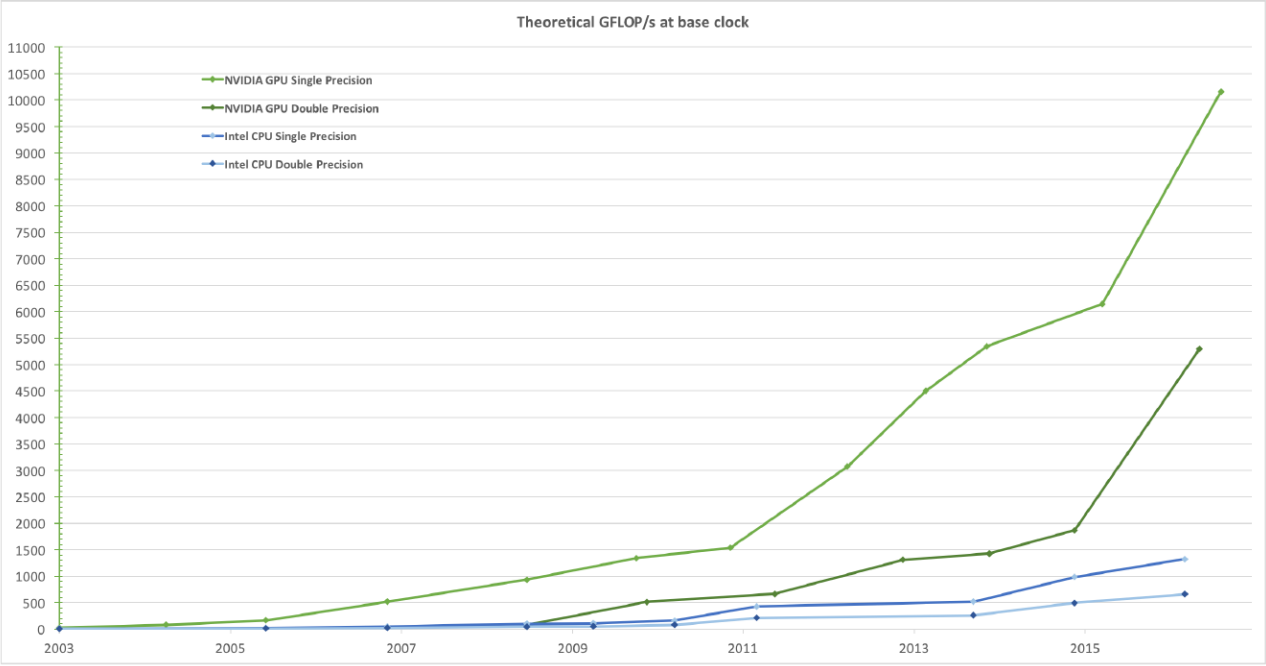
28.02.2019

# Why do we use GPUs to decode?

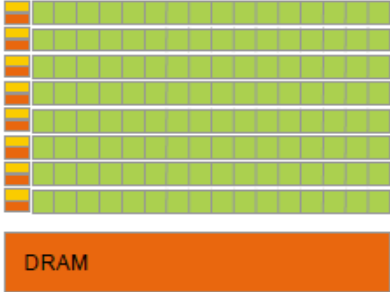
- Software defined receiver
- Many different codes (e.g., turbo product codes, turbo convolutional codes, LDPC, BCH codes)
- Many different code parameters
- Other GPU receiver modules
- Quick development
- Parallel data structures
- Latency is not critical
- Throughput is important
- Small quantity



# GPU: From Graphics Processing to General Purpose Parallel Computing



CPU



GPU



# CUDA (Compute Unified Device Architecture)

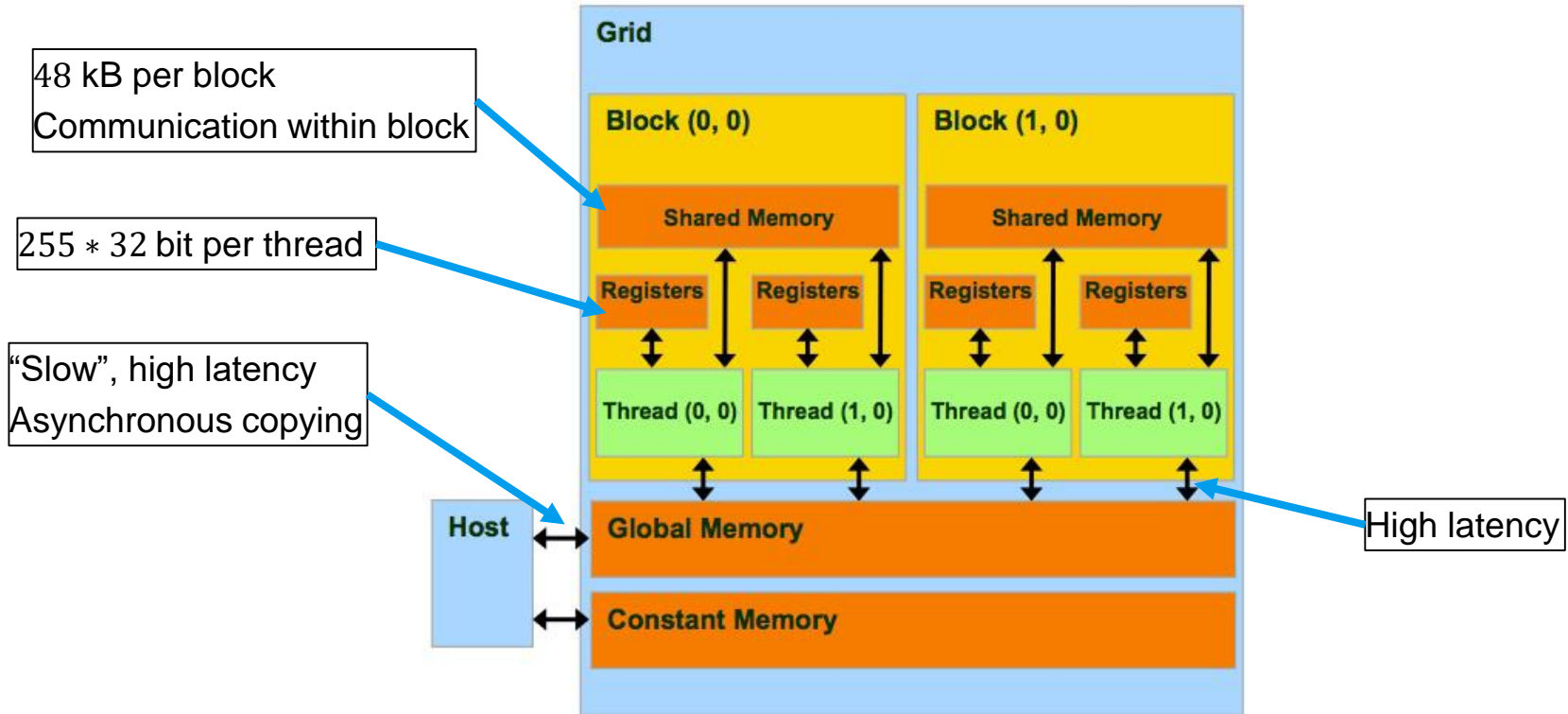
GPU Computing Applications						
Libraries and Middleware						
cuDNN TensorRT	cuFFT, cuBLAS, cuRAND, cuSPARSE	CUDA MAGMA	Thrust NPP	VSIPL, SVM, OpenCurrent	PhysX, OptiX, iRay	MATLAB Mathematica
Programming Languages						
C	C++	Fortran	Java, Python, Wrappers	DirectCompute	Directives (e.g., OpenACC)	
CUDA-enabled NVIDIA GPUs						
Turing Architecture (Compute capabilities 7.x)	DRIVE/JETSON AGX Xavier	GeForce 2000 Series	Quadro RTX Series	Tesla T Series		
Volta Architecture (Compute capabilities 7.x)	DRIVE/JETSON AGX Xavier			Tesla V Series		
Pascal Architecture (Compute capabilities 6.x)	Tegra X2	GeForce 1000 Series	Quadro P Series	Tesla P Series		
Maxwell Architecture (Compute capabilities 5.x)	Tegra X1	GeForce 900 Series	Quadro M Series	Tesla M Series		
Kepler Architecture (Compute capabilities 3.x)	Tegra K1	GeForce 700 Series GeForce 600 Series	Quadro K Series	Tesla K Series		
	EMBEDDED	CONSUMER DESKTOP, LAPTOP	PROFESSIONAL WORKSTATION	DATA CENTER		

# NVIDIA Tesla V100

Streaming Multiprocessors	80
Cores per SM	2*32
Cores	5120
GPU Boost Clock	1530 MHz
Peak FP32 TFLOPS	15.7
Memory Size	16 GB
TDP	300 Watts
Transistors	21.1 billion
GPU Die Size	815 mm <sup>2</sup>
Manufacturing Process	12 nm FFN
Cost	≈ 7000€



# GPU Memory Hierarchy



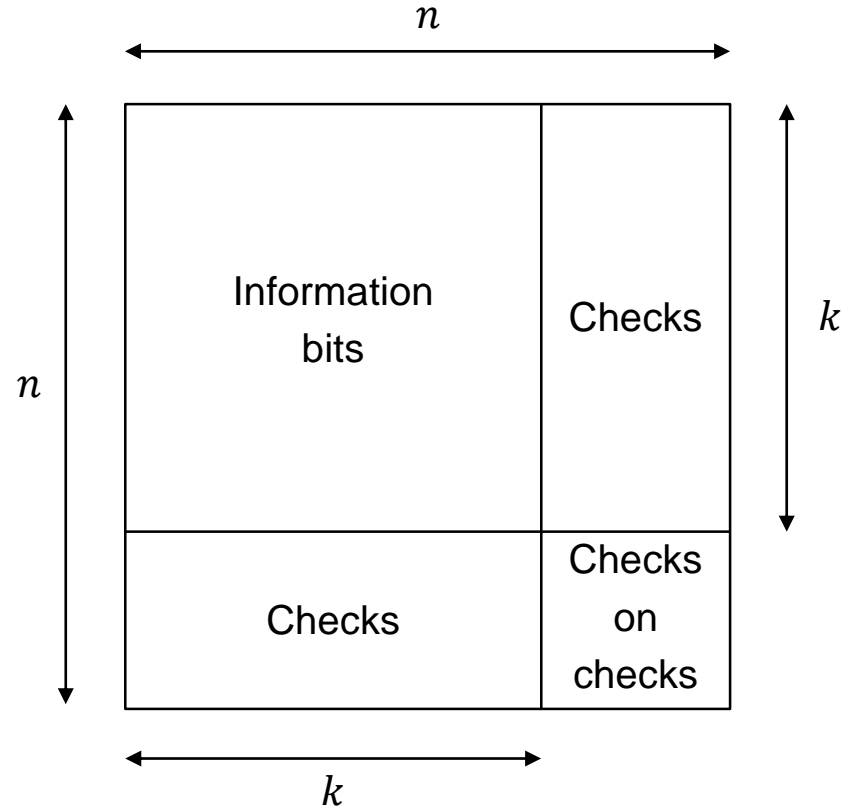
# Turbo Product Code (TPC)

Assume a symmetric TPC:

- Same row and column block code
- Symmetric block-code  $(n, k, \delta)$
- $(n, k, \delta) \times (n, k, \delta) = (n^2, k^2, \delta^2)$  TPC

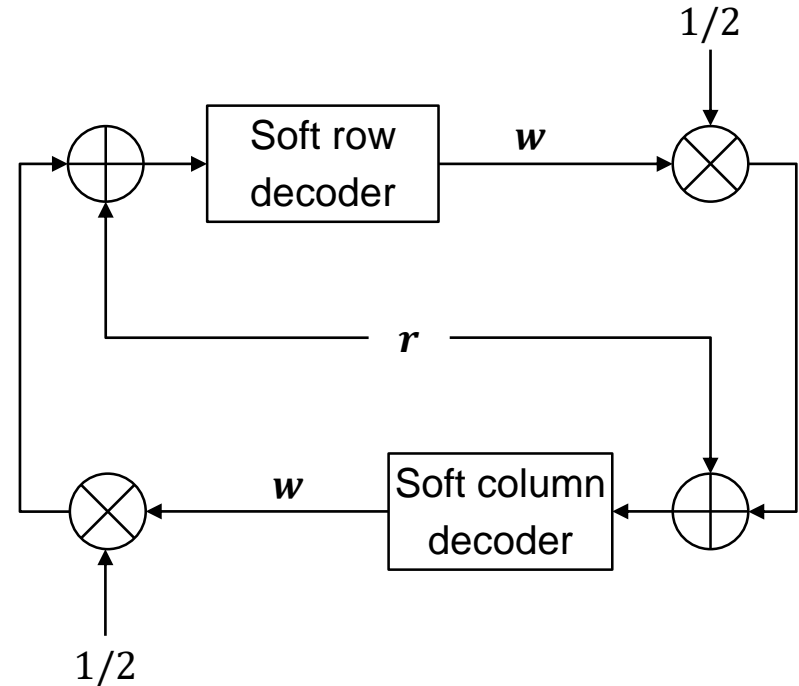
Application

- Satellite communication (e.g., VSAT)
- ...



# TPC Decoding

- Hard-in hard-out (HiHo) decoder
  - Closed-chain errors remain
- Soft-in soft-out (SiSo) decoder
  - Chase-Pyndiah decoder [Pyndiah, R. M., IEEE Trans. Commun., 1998]
  - “A Low Complexity Decoding Algorithm for Extended Turbo Product Codes” [Xu, C.; et al., IEEE Trans. Wireless Commun., 2008]
  - ...
- Hybrid approaches





# Low Complexity Soft Block Decoder

1. Input: LLRs  $\bar{\mathbf{r}} = \mathbf{r} + \frac{1}{2}\mathbf{w}$
2. ML sequence :  $\mathbf{y}_j = \begin{cases} 0, & \text{where } \bar{r}_j > 0 \\ 1, & \text{where } \bar{r}_j \leq 0 \end{cases}$
3. Determine  $p$  positions with minimal  $|\bar{r}_j|$
4. Generate  $2^p$  test pattern  $\mathbf{t}^{(1)}, \dots, \mathbf{t}^{(2^p)}$  (all permutations of  $p$  positions)
5. Perturbed sequences:  $\mathbf{z}^{(i)} = \mathbf{y} \oplus \mathbf{t}^{(i)}$
6. Hard decode  $\mathbf{z}^{(i)} \rightarrow \mathbf{c}^{(i)}$
7. Analog weight:  $d^{(i)} = -\sum(\mathbf{c}_j^{(i)} \oplus \mathbf{y}_j) |\bar{r}_j|$
8. ML code word:  $\mathbf{c}^{(\hat{i})}$  where  $\hat{i} = \arg \max_i d^{(i)}$
9. Estimate extrinsic information  $\mathbf{w}$  of  $\mathbf{c}^{(\hat{i})}$

Estimate extrinsic information of  $\mathbf{c}_j^{(\hat{i})}$ :

- Find largest  $d^{(\hat{i})}$  where  $\mathbf{c}_j^{(i)} \neq \mathbf{c}_j^{(\hat{i})}$

$$\mathbf{w}_j = (d^{(i)} - d^{(\hat{i})}) (2\mathbf{c}_j^{(\hat{i})} - 1) - \bar{r}_j$$

- If  $\mathbf{c}_j^{(i)} = \mathbf{c}_j^{(\hat{i})}$  for  $i \in [1, \dots, 2^p]$

$$\mathbf{w}_j = \frac{1}{p} \left( \min_i (d^{(i)}) - d^{(\hat{i})} \right) (2\mathbf{c}_j^{(\hat{i})} - 1)$$

# TPC Parameters

## ■ TPC with extended Hamming codes

- (32, 26, 4) x (32, 26, 4)      Code rate:  $\frac{676}{1024} = 0.66$       1 \* 32 threads per block
- (64, 57, 4) x (64, 57, 4)      Code rate:  $\frac{3249}{4096} = 0.79$       2 \* 32 threads per block
- (128, 120, 4) x (128, 120, 4)      Code rate:  $\frac{14400}{16384} = 0.88$       4 \* 32 threads per block

## ■ TPC extrinsic information matrix fits into shared memory (as 16bit half precision floats)

## ■ Received LLRs fit into registers (as 16bit half precision floats)

## ■ 4 turbo iterations

## ■ HiHo decoder parameters:

- $4 * 2 * n = 8n$  hard block decoder calls

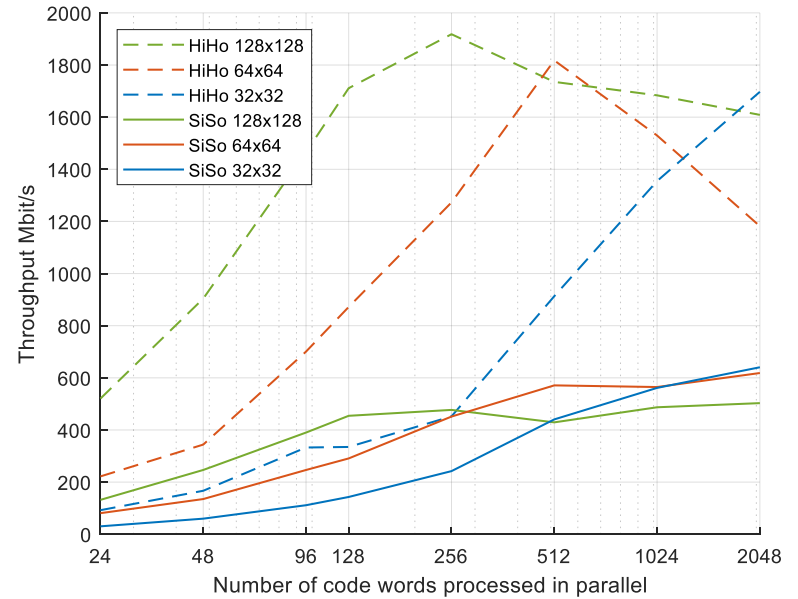
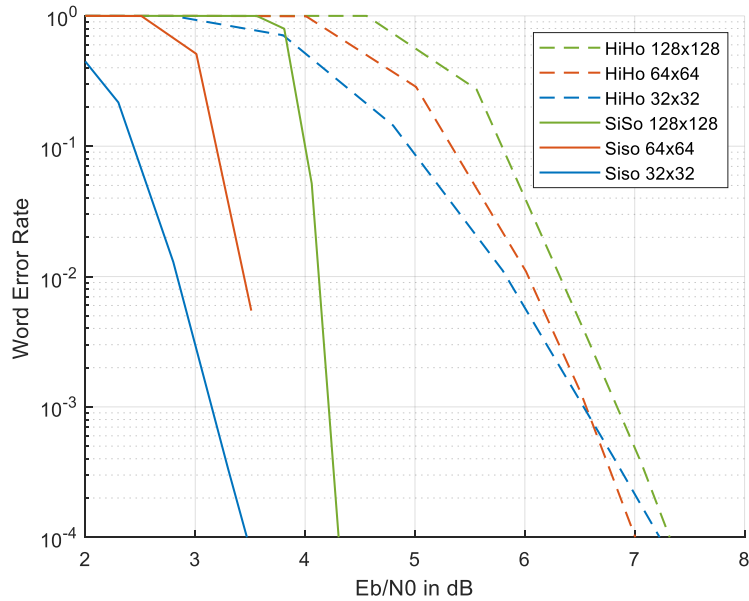
## ■ SiSo decoder parameters:

- $p = 3$  bit positions
- $4 * 2 * 2^p * n = 64n$  hard block decoder calls

- Hard block decoder should be fast:
  - Use hardware intrinsic where possible
  - Provide parameters at compile time



# Performance Results



# Other GPU Modules

- Turbo convolutional decoder
- LDPC decoder
- BCH decoder
- CRC
- Detector
- Time offset estimator
- Frequency/phase offset estimator
- Resampler

