CONTACT NAME
Przemyslaw Tredak: ptrendx@gmail.com

Efficient implementation of Mersenne Twister MT19937 Random Number Generator on the GPU

Przemysław Trędak[1] and Cliff Woolley[2]
[1] Faculty of Physics, University of Warsaw, Poland
[2] NVIDIA Corporation

Motivation

- Mersenne Twister MT19937 Pseudo Random Number Generator is reliable and widely used PRNG for scientific and commercial purposes
- This PRNG is not easily parallelizable - there is no efficient implementation of it for GPUs nor manycore CPUs
- To address that, authors of MT19937 invented different version that is more easily suited for GPUs - MTGP, Mersenne Twister for Graphics Processors
- MTGP produces different results than MT19937 and its statistical properties were not checked as extensively as those of MT19937
- Therefore there is a need for efficient implementation of MT19937 for GPUs

Basic Idea

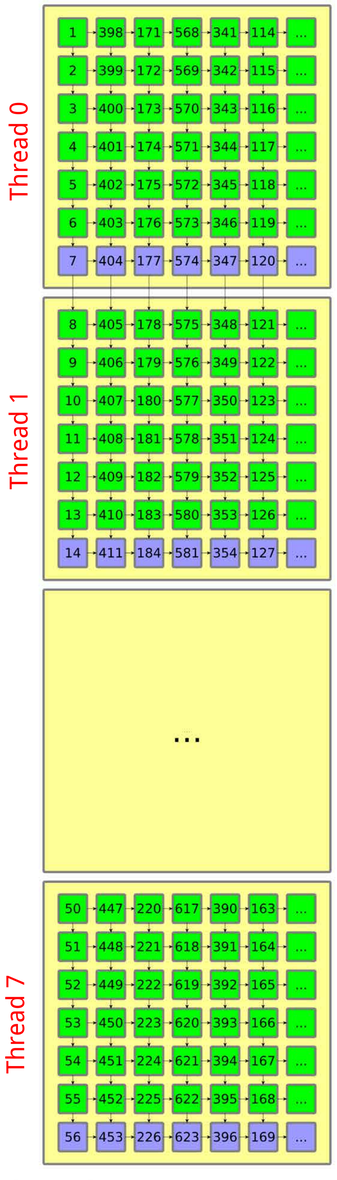
- Mersenne Twister MT19937 is based on recursion

$$x_{i+624} = f\left(x_i, x_{i+1}, x_{i+397}\right)$$

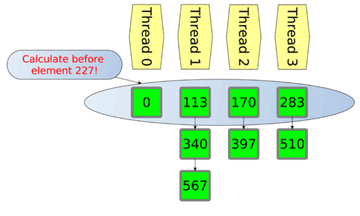
- Parallelism in the single MT19937 generator is limited - need of many generators
- MT19937 PRNG has very big state vector - 624 4-byte words or nearly 2.5 KB of memory
- Amount of fast shared memory on current devices is 48 KB per single Streaming Multiprocessor - fits only 19 generators per SM
- Much bigger storage is possible when using registers for keeping generator's state - 128 KB on Fermi, 256 KB on Kepler, but the communication is difficult!
- Need to find a way to assign state vector elements to device threads to minimize communication overhead

Algorithm

- Single generator is operated by 8 threads
- Each thread keeps 77 state vector elements in registers

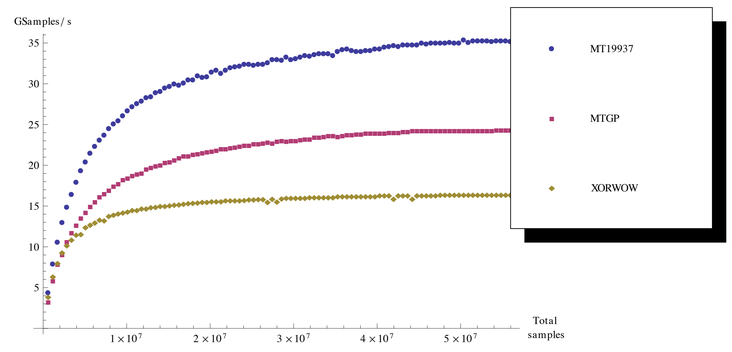


- State vector elements are assigned to threads to minimize the need for communication - 66 out of 77 of them can be computed without any communication
- Communication pattern for the rest of elements is simple - thread i needs data from thread (i+1) mod 8 - possibility of using warp shuffle instruction instead of shared memory load/stores
- 8 extra elements remain and need to be treated separately by 4 threads:



- 64 generators per Streaming Multiprocessor - over 3 times more than would fit in shared memory
- More generators obtained using jumping ahead in a sequence of single MT19937 generator by large constant length

Performance



- On NVIDIA Tesla K20X, resulting performance is **2.2x** higher than the CURAND XORWOW generator and **1.5x** higher than MTGP generator
- Peak performance of **37.8** GSamples/s

Conclusion

- We presented novel implementation of MT19937 Pseudo-Random Number Generator on the GPUs
- Presented implementation achieves major speedup over other generators used on the GPUs, like XORWOW and MTGP
- This work is being incorporated into an upcoming release of cuRAND

References

- Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. M. Matsumoto, T. Nishimura
- Variants of Mersenne Twister suitable for Graphics Processors. M. Saito, M. Matsumoto
- Efficient Jump Ahead for F2-Linear Random Number Generators. H. Haramoto, M. Matsumoto, T. Nishimura, F. Panneton, P. L'Ecuyer.