ISS-SP-027

GPU における多倍長演算アルゴリズムと その高速化に関する研究

千葉 光剛† 尾崎 操一†† 高木 信裕†† 武居 知哉†† 和田 幸一†† 藤本 典幸†††

†法政大学院理工学研究科応用情報工学専攻 ††法政大学理工学部応用情報工学科

†††大阪府立大学大学院工学研究科電気・情報系専攻

1. はじめに

GPUは画像処理を担当する主要な部品の一つである.GPUは CPU に比べて高い並列処理能力を持っているため,現在GPU を用いた計算処理の高速化が検討されている[1].
本研究では,32 ビットや 64 ビットを超えるような大きな値を扱う場合に用いられる多倍長整数[2]に関する演算の高速化を図った.今回扱った多倍長演算は多倍長加算と多倍長乗算である.多倍長加算に関しては 3 つのアルゴリズムを, 多倍長乗算に関しては 5 つのアルゴリズムを実装した.実装したアルゴリズムは多倍長演算を行うライブラリの一つである MPIR[2]と比較を行った.

2. 多倍長加算と多倍長乗算のアルゴリズム

多倍長整数は配列を用いて表現されており,配列の 1 つの要素を 1bit とする場合と配列の 1 つの要素を 32bit とする場合を実装した.つまり$nbit$の多倍長整数は1要素 1bitの場合は要素数n個の配列で表現されており,1 要素 32bit の場合は要素数$n/32$個の配列で表現されている.

多倍長加算アルゴリズムとして,桁上先見加算アルゴリズム,桁上保存加算アルゴリズムと冗長 2 進表現を利用した加算アルゴリズム[3]を用いた.多倍長乗算アルゴリズムは, 被乗数と乗数から部分積を求め, その部分積の加算を繰り返すことで積を求める.

多倍長加算を行うアルゴリズムは, 配列の 1 つの要素を 1bitとして計算する桁上先見加算アルゴリズムと冗長 2 進表現を利用した加算アルゴリズムの 2 つを実装し,配列の 1 つの要素を 32bitとして計算する冗長 2 進表現を利用した加算アルゴリズムを実装した.多倍長乗算の部分積の加算を行うアルゴリズムは, 配列の 1 つの要素を 1bitとして加算を桁上先見加算アルゴリズム,冗長 2 進表現を利用した加算アルゴリズムの 2 つを用いた多倍長乗算を実装した.また配列の 1 つの要素を 32bitとして計算する桁上先見加算アルゴリズム,桁上保存加算アルゴリズムと冗長 2 進表現を利用した加算アルゴリズムを用いていた多倍長乗算アルゴリズム 3 つも実装した.

3.実行時間

GPU 上で実装したアルゴリズムの実行時間は CPU-GPU 間の通信時間を含まない GPU 上でプログラムが実行されている時間である.多倍長加算の実行時間の比較を表 1 に多倍長乗算の実行時間の比較を表 2 に示す.冗長 2 進表現を利用した加算,乗算は冗長 2 進表現で答えが求まるまでの時間,桁上保存加算アルゴリズムを利用した加算は 2 数が求まる時間,乗算は部分積の 32 bitのズレがあるグループと無いグループで 2 数ずつ計 4 つが求まるまでの時間,桁上先見加算アルゴリズムは加算,乗算ともに答えが求まる時間である.

実行結果は,CPU は「Intel Core i7-3090X」,GPU は「GeForce GTX TITAN」を使用している.

表 1　多倍長加算の実行時間の比較(単位は ms)

bit	先見 加算 (1bit)	先見 加算 (32bit)	冗長 2 進 (1bit)	冗長 2 進 (32bit)	MPIR
262144	3.742	0.469	0.182	0.105	0.588
524288	7.206	0.813	0.218	0.112	0.088
1048576	15.06	1.509	0.343	0.129	0.178
2097152	31.35	3.217	0.461	0.136	0.355
4194304	64.38	7.543	0.768	0.178	0.722

表 2 多倍長乗算の実行時間の比較(単位は ms)

bit	桁上 先見 (1bit)	桁上 先見 (32bit)	桁上 保存 (32bit)	冗長 2 進 (1bit)	冗長 2 進 (32bit)	MPIR
2048	120.545	1.031	0.256	5.205	0.477	0.039
4096	479.772	2.530	0.315	20.166	0.545	0.134
8192	1899.739	8.911	0.642	83.814	0.705	0.352
16384		36.448	1.013	320.72	1.024	0.792
32768		151.04	1.453		2.037	2.145

4. 結果

多倍長加算に関しては MPIR と比較を行うと配列の 1 つの要素を 32 bitとした冗長 2 進表現を利用した加算アルゴリズムの実行時間が MPIR よりも高速だった.多倍長乗算に関しては MPIR と比較を行うと部分積の加算として配列の 1 要素を 32 bitとした桁上保存加算アルゴリズムを用いた乗算アルゴリズムの実行結果が MPIR よりも高速で計算できる可能性があることがわかった.

参考文献

[1]青木尊之,額田彰."はじめての CUDA プログラミング", 光学社(2009)
[2]T.Granlund, the GMP Development Team, W. Hart,and the MPIR Team. MPIR 2.7.2 "*MPIR* 2.7.2", November 2015. http://mpir.org/downloads.html.
[3] 高木直史,安浦寛人,矢島脩三"冗長 2 進加算木を用いた VLSI 向き高速乗算器",電気通信学会論文誌 J67-D(4), (1984)