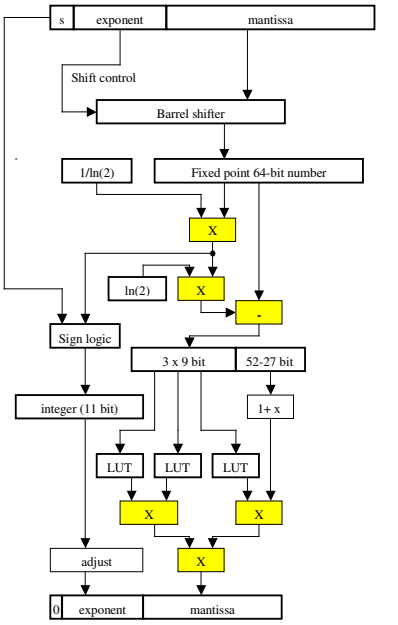
Two Electron Integrals calculation accelerated with Double Precision exp() Hardware Module

Maciej Wielgosz[2], Marcin Pietroń[2], Ernest Jamro[1,2], Paweł Russek[1,2] ,
Kazimierz Wiatr[1,2]
[1] Institute of Electronics AGH, Kraków
[2]ACK „Cyfronet" AGH, Kraków

1. Introduction

FPGA implementation of double precision exponential function module is presented here. The module will be incorporated in the Gaussian system to accelerate the extremely time consuming exponential function evaluation. The exp function is accelerated on SGI RASC board with two Virtex-4 LX200 FPGA. The exp() function alone occupies less than 3% Virtex-4 LX200 FPGA. Exp() arguments are fetched to the FPGA's and results are sent back to processors over the system bus working at speed of NUMAlink 6,4 GB/s. The exponential module reaches the processing speed of 200 MHz, . The external memory interface limits the number of operation (down) to two exp() every clock cycle per a FPGA. The overall end-to-end algorithm execution speedup that authors expect to achieve is 4x as compared to the sequential implementation of the algorithm executed on a single 2 GHz Intel Itanium2 processor.

2. Architecture of exp() module



$$e^x = e^{a*ln2+b} = 2^a.e^b$$
$$e^b = e^{b1}.e^{b2}.e^{b3}.e^p$$
$$e^p = 1 + p + ...$$

The proposed *exp()* module consists of the following sub-modules:
-exceptional states (*inf*, *NaN*) detectioning logic for input data; this unit also converts input data to internal fixed point standard (barrel shifter).
-exponent evaluation module, which separates fractional and integer part (which corresponds to exponent field of the result) - sign migration from fractional to integer part
-LUTs which store fractional elementary values of *exp()*, polynomial approximation and multipliers
-conversion to IEEE-754 standard

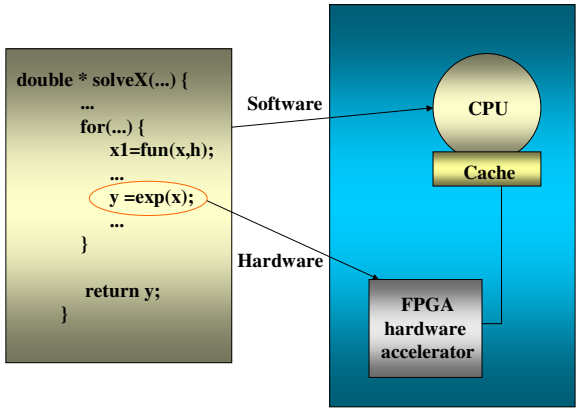
3. Profiling

There are multiple exponential functions in the source code but only several of them are heavily employed in most of the common chemical computation (tasks. For example, while computing benzene molecule the exp() function is executed a few billions times. One of the exp() function hot spots is the subroutine responsible for functional computation in solving the Hartree-Fock equation.
Employment of well known profilers (e.g. gnuprof) was the first step to profile the Gaussian application. Results of using these profilers were not satisfying (e.g. not compatible with Gaussian binaries). Consequently, a new dedicated profiling tool was developed. This tool is able to:

- parametrise functions' monitoring (functions' name, location)
- evoke graphs
- estimate time of function calls
- monitoring of data flow

4. Software – Hardware CoDesign

Automated tool will be developed and extended with additional options that will enable investigation of the source code to find the hot-spots. The tool is to facilitate parameterization of the user environment. We also concentrate on automated extacting of inherent parallelism from the source code.



5. Implementation results

Implementation results for Virtex-4 LX200

implementation	# 4-input LUTs	# flip-flops	# 18-Kb BRAMs	DSP48
1) With DSP48	1293 (0.73%)	105 (0.06%)	6 (1.8%)	71 (74%)
2) Without DSP48	13375 (7.5%)	105 (0.06%)	6 (1.8%)	0
3) optimized multipliers	5025 (3%)	5223 (3%)	6 (1.8%)	0
4) RASC system	14,521 (8%)	20,125 (11%)	29 (8%)	0
5) Single precision (Virtex-2 1000) [6]	1896 (1.044%)	1896 (1.036%)	0	0

Implementation results show that multipliers are the most hardware consuming part of the module and introduce the longest latency, this gave rise to the idea to design the dedicated speed optimized multipliers.

It is worth noticing that there is a large difference between resources absorbed by the standard and optimized multipliers versions. A considerable difference between the number of flip-flops is conspicuous due to pipeline mechanism employed together with optimized multipliers.

Parameters of the module:

- 200 Mhz
- 28 clock cycles latency
- 4% of Virtex 4 LX-200
- max.error 2 ULP