USING DSP BLOCKS FOR ROM REPLACEMENT: A NOVEL SYNTHESIS FLOW

Gareth W. Morris, George A. Constantinides, and Peter Y.K. Cheung

Imperial College, Circuits and Systems Group
Department of Electronic and Electrical Engineering
Exhibition Road, London, SW7 2BT

ABSTRACT

This paper describes a method based on polynomial approximation for transferring ROM resources used in FPGA designs to multiplication and addition operations. The technique can be applied to any FPGA architecture containing embedded multiplication, however this paper focuses on using the DSP blocks of Altera Stratix and Stratix II architectures. The transformation is combined with other resource transfers and integrated in a synthesis flow targeting designs implemented on heterogeneous FPGAs. The main advantage of such a system is in handling user constraints on each type of resource: DSP block, LUT and ROM, in addition to timing-related constraints. The flow is based on an extension to the Altera Quartus II synthesis software and Quartus University Interface Program (QUIP) framework. Results are provided for implementations of benchmark algorithms and it is shown through a design-space exploration that the set of achievable designs for the algorithms has been extended by the use of the proposed methods.

1. INTRODUCTION

Contemporary FPGA architectures are moving towards heterogeneity with the embedding of so called 'hard cores' into the device fabric. Many current and previous generation FPGAs include various styles of multiplication elements (e.g. Xilinx Virtex 2 embedded multipliers, Altera Stratix DSP blocks) and RAM in addition to the traditional array of Look Up Tables (LUTs) with routing resources. Conventional synthesis techniques usually require that these heterogeneous elements are instantiated by use of vendor specific syntax or symbols, which can only be mapped to that specific architecture element.

This paper proposes a method of re-mapping resources between ROMs and Altera DSP blocks using a transformation based on modified polynomial approximation with independent coefficient domains. The proposed transfer allows DSP blocks not specifically used in a user's design to be given over to implementing ROM functionality.

To obtain results in real designs the transfer is combined into an alternative synthesis system built on Altera Quartus II tools within the QUIP framework. This system allows re-mapping of functions between heterogeneous elements, specifically ROMs, multipliers or DSP blocks and LUTs as represented in Figure 1.

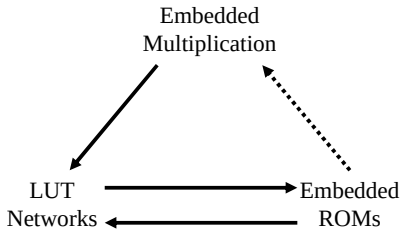

Fig. 1. Diagram showing the transformations between FPGA resources integrated within the proposed synthesis system of Section 4. The dotted line represents the new transformation made possible with the technique proposed in Section 3.

Benchmark designs have been applied to the proposed system, which resulted in a two to fourfold increase in Pareto-optimal[1] implementations over Altera Quartus II. In order to investigate the types of ROM data to which the proposed technique can usefully be applied, synthetically generated data sets have been used. Results show improvements in LUT utilisation over standard techniques of up to 51%.

The rest of this paper is comprised of five further sections. Section 2 provides a summary of related work in this area. Section 3 explains the techniques used for conversion from ROM data into arithmetic operations. Section 4 details the proposed synthesis system. In Section 5 we present and analyse results from applying the synthesis system to different benchmark designs and investigate the types of ROM data that improve LUT usage after applying the techniques of Section 3, over existing methods. Finally, Section 6 con-

[1] A point on the Pareto-optimal design curve represents a design which cannot have improved characteristics in one optimisation criterion, without sacrificing those in another.

cludes the paper and includes plans for future work in this area.

2. BACKGROUND

The authors of [1] examine the problem of uniform piecewise polynomial approximation with variable joints. The authors recognise the need for this in approximating discontinuous functions. A procedure for computing the position of these joints, when given the approximation order, is described. The procedure has guaranteed convergence and minimial approximation error. The alternative transformation scheme, proposed in this paper, achieves equivalent or better results, and in the context of the proposed synthesis system it does not require approximation order as an external input.

A recent analysis of the need for programming models in heterogeneous CPU/FPGA devices is presented in [2]. The article discusses the current high level tools available (SystemC, Handel-C, System Verilog, etc.) and concludes that none of them yet fully support the heterogenous elements of modern FPGAs. The approach described in this paper addresses the issue of using heterogenous elements in designs where they are not specifically instantiated.

Application of polynomial approximation to the problem of emulating ROM functionality with arithmetic operations was the subject of a previous paper by the authors [3]. Overviewed are techniques for ROM to arithmetic conversion using modified polynomial approximation. The work described here expands on [3] and uses the transformation as a basis for providing a resource constrained guided synthesis system.

Use of embedded RAM blocks in modern FPGAs for implementing logic functions has been the subject of some considerable study [4], [5]. The work discusses the mapping of random combinatorial logic into embedded RAM blocks and the effectiveness of this mapping for different FPGA architectures. Logic packing techniques are described, which aim to find the most efficient cut of the design that can be taken to form the ROM contents. An average 69% reduction in LUT utilisation was achieved over 14 benchmark circuits, with a 6% decrease in critical path delay. The work described here complements this approach by adding ROM to multiplier transformations and combining the concepts of ROM to multiplier and LUT to ROM transformations in a unified synthesis approach.

The problem of mapping resources in heterogeneous FPGAs is considered in [6], where a more fine grain view of heterogeneity is taken than [5], and mapping logic to a dual level hierarchy of LUT sizes is examined. A logic packing algorithm is proposed and implemented which reduces routing delays by 6% over Altera's Max+PlusII design suite and with equivalent computation time to other techniques. The methods proposed in this paper go further by examining synthesis in a heterogenous architecture containing additional dimensions to that of LUT size.

3. REPLACING ROMS WITH DSP BLOCKS

This section describes the proposed ROM to multiplication transfer method. It uses a lossless approximation technique based on a modified polynomial evaluation. The general formula for polynomial evaluation is reproduced in (1).

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + ... + c_n x^n \tag{1}$$

Simple polynomial approximation can be applied to any set of ROM data, but to achieve faithful reproduction there is an exponential worst case bound on polynomial order as shown in (2), denoting approximation order by n and ROM input address bus width as a.

$$n \leq 2^a - 1 \tag{2}$$

To mitigate this problem many different coefficients are used for the polynomial evaluation, depending on the value of the input address bus of the ROM being replaced. The coefficients used to evaluate a particular address are selected from a bus derived from the most significant bits of the main address bus. The masking function used is expressed as shown in (3). Each coefficient is selected from the results of separate masking functions, this is shown on the example architecture of Figure 2.

$$m_i = M_i(x) \tag{3}$$

Note that the well known uniform piecewise polynomial (UPP) and simple polynomial approximations [1] are subsets of this technique, corresponding to certain masks.

A modulus may be applied to the input address bus for the purposes of evaluating the polynomial, which has the benefit of limiting the sensitivity of the evaluation to coefficient values thereby reducing coefficient precision requirement. This modulus is restricted to type 2^k, as modulus operations of this type can be easily calculated by simply ignoring most significant bits of the input address bus. The output of the modulus function r has a reduced range compared to x, and is expressed in (4).

$$r = x \bmod 2^k \tag{4}$$

Through combining (1), (3) and (4) the proposed technique can be described with by (5).

$$y = c_0(m_0) + c_1(m_1)r + c_2(m_2)r^2 + c_3(m_3)r^3 + ... + c_n(m_n)r^n \tag{5}$$

An example of the architecture proposed for implementing (5) is presented in Figure 2, with arbitrarily chosen coefficient selection buses. The MSB and LSB maskings of the input address bus, used to select coefficient values and calculate r respectively, are achieved with zero overhead in an FPGA implementation. The coefficient multiplexors are generated in an FPGA with LUT based ROM. The proposed implementation uses an Estrin's method [7] polynomial evaluation, although any other evaluation technique can also be used.

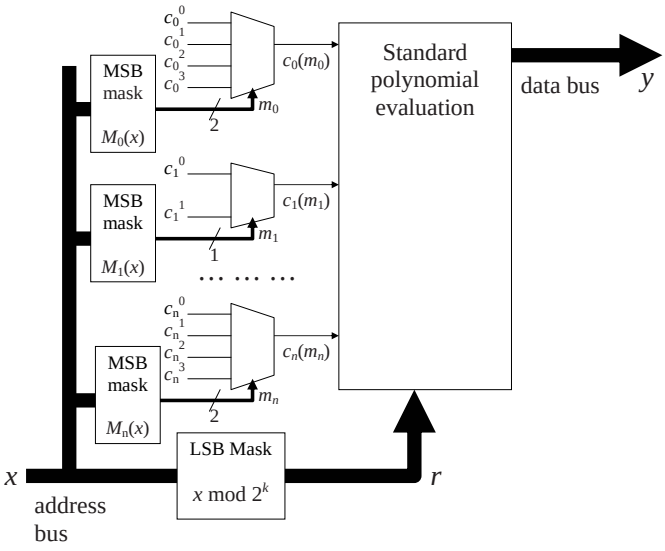

Fig. 2. An example of the architecture proposed for ROM replacement.

A design based on the architecture described in Figure 2 is defined by the approximation order, the address bus masks selecting each coefficient, the modulus applied to the input address bus, and the ROM contents. These parameters can be set to significantly reduce LUT usage. Completely searching all mask and modulus combinations is realistic only for smaller designs due to exponential computational complexity. For larger designs, a simulated annealing optimisation is used to find the address bus masks and modulus that minimise the number of LUTs used in implementing the design.

The simulating annealing algorithm produces mask sets and a modulus with each iteration. Since polynomials are linear in the coefficients, linear program solution may be used to find the coefficient values. The mask set and modulus are used to formulate a set of linear equations which are solved by a linear program solver to find coefficient values that best reproduce the original ROM.

The linear program solver returns the coefficient values in floating-point format. Before LUT utilisation can be calculated, these values are subject to a word length optimisation scheme. Each coefficient is scaled to a fixed point representation with zero redundancy in the MSBs. The precision of the representation is found by successively increasing the precision of the coefficients separately, retaining full floating-point precision in the other coefficients until faithful rounding is achieved. Next, all coefficient values are rounded, and the error of the resulting approximation is calculated again. If necessary, the precision of all coefficients is then increased by the same quantity until faithful rounding is achieved, to a maximum of 64 bits precision.

The word-length optimised coefficients are used to estimate LUT utilisation by using a high-level model of logic synthesis. The result is then used as a figure of merit by further iterations of the simulated annealing algorithm.

A overview of the technique and the interaction between the ROM data parser, simulated annealing algorithm and linear program solver is provided in Figure 3.

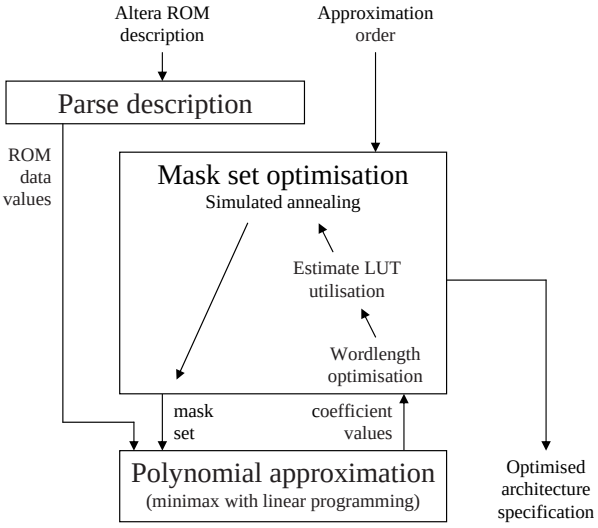

Fig. 3. Diagram representing the ROM to arithmetic conversion technique

4. PROPOSED SYNTHESIS FLOW

The transformation technique described in the previous section has been combined with other resource transformations into a resource constrained guided synthesis system. This system can be used to implement designs on FPGA devices which may not have been possible using existing synthesis techniques.

Firstly the user's design is specified with any usual mechanism (e.g. VHDL, Verilog, schematic capture, Altera DSP builder). This is taken as the input to the synthesis system. The design is technology mapped to a particular architecture, and this mapped design is preserved for later manipulation. As with a usual synthesis flow, the design is then fitted to a particular part and a timing simulation is made. The tool

then parses the Altera timing and fitting reports to compare the synthesised design to user constraints on system clock speed in addition to ROM bit, DSP block and LUT utilisation. If the constraints are met then the programming bit stream is assembled as usual. If the constraints are not met then resources in the design must be transferred depending on the constraint violation. The resources are transferred by substituting replacement components into the previous implementation. The process is repeated until the user constraints are met, or no further optimisation is possible.

A block diagram representing the proposed design-flow and its relationship with the existing Altera Quartus II tools is represented in Figure 4. The existing tools are interfaced within the QUIP framework [8].

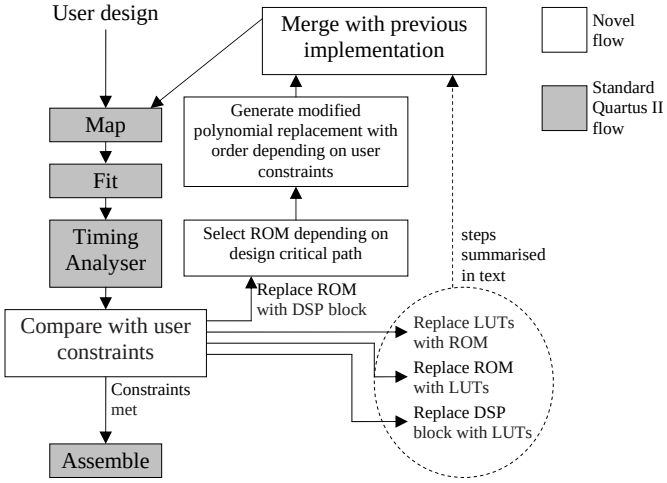

Fig. 4. Block diagram of proposed synthesis system

In the case of a ROM to DSP block transfer, the design is analysed in order to select the ROM furthest away from the critical path for replacement. The parameters of the ROM replacement circuitry described in Section 3 depend on the timing slack from the critical path and the estimated delay of the replacement, which is calculated via a heuristic based on the target architecture's timing specifications.

As the focus of this paper is on the automation of the ROM to DSP block transfer, the remaining transformations shown in the ellipse on Figure 4 are carried out by hand. LUT to ROM and ROM to LUT replacements are performed by tabulation of stored data and forcing the Quartus II synthesis system to implement the tables with ROMs or LUTs as appropriate. DSP block to LUT replacements are performed by substitution of references to DSP block internal multipliers and adders by alternative circuits produced with Synplicity's Synplify design suite. DSP blocks furthest away from the critical path are selected for replacement first.

5. RESULTS

5.1. Benchmark Circuits

To investigate the larger set of possible implementations produced using the proposed techniques for a given design, the synthesis system incorporating the proposed ROM to DSP block transfer has been applied to five benchmark designs taken from various sources.

The first benchmark is an implementation of the CORDIC algorithm producing simultaneous 21 bit sine and cosine outputs. The second benchmark is a fully folded 32 tap FIR filter with 17 bit data bus output and coefficient width. The third benchmark is a hybrid DSP system consisting of signal generation, combination and filtering. The fourth is an FM receiver, which includes a PLL'd oscillator, phase detector and filtering. The final benchmark is an audio synthesis system including oscillators and additive mixing. The complete set of Pareto-optimal results for the tests are shown in Table 1 and are discussed below, the non-Pareto-optimal results that were obtained during the design space exploration have not been reproduced.

In the case of the CORDIC algorithm the proposed system achieves a threefold increase in Pareto-optimal implementations over the point solution provided by the standard Quartus II flow. The additional designs show the potential for completely freeing-up embedded ROM blocks, as well as improvements in system clock frequency of 11 and 13% at the expense of 3 and 22% increases in LUT utilisation respectively. However, this is the only design examined where no ROM to DSP block transfers resulted in new Pareto-optimal designs.

The FIR filter has the largest number of Pareto-optimal implementations. These new implementations give alternatives for increasing all four figures of merit. An increase in LUT usage by 11% has lead to a 27% improvement in clock frequency. A transferral of ROMs into DSP blocks resulted in a total reduction in ROM utilisation, at the expense of using eight times more DSP blocks and 32% more LUTs, this transfer reduced maximum system clock frequency by only 17%. The last implementation comes from a resource transfer from ROM bits to LUTs. A total reduction in used ROM bits has increased the number of LUTs by 52% and also resulted in a 25% improvement in system clock rate.

For the mixed DSP system, a single extra implementation has been discovered as a result of our design space exploration. A reduction in ROM bit usage by 94% with a resource transferal to DSP blocks has increased LUT utilisation by 17% and reduced maximum clock frequency by 77%.

As a result of the design space exploration of the FM receiver, there are two additional implementation possibilities. By increasing the number of LUTs in the design by 12% the clock frequency can be improved by a compara-

Table 1. Table showing the Pareto-optimal results of our investigations, post place and route. The shaded result for each design represents the one produced by the Altera Quartus II system with default synthesis options.

Algorithm	ROM bits	DSP blocks	LUTs	System clock (MHz)
Cordic	544	0	526	163
	0	0	543	182
	0	0	641	184
FIR Filter	544	2	168	133
	0	18	221	110
	544	2	187	169
	0	2	255	166
DSP System	1768	0	2024	150
	104	34	2370	34
FM Receiver	8192	0	546	63
	8192	0	613	72
	0	2	1000	68
Audio Synthesis	256	0	507	210
	0	2	529	200
	0	0	567	194

ble 14%. Complete transfer of ROM resources to other elements is possible, this uses 2 DSP blocks and increases LUT utilisation by 83%. This transfer resulted in an 8% improvement in clock frequency over the standard Quartus implementation.

The final benchmark, the audio synthesis engine, has also had two additional design space points highlighted as a result of the design space exploration. The first of these transfers all the ROM resources into 2 DSP blocks, with a small increase of 4% of LUTs utilisation and a decrease in clock frequency of 5%. The second design maps all the ROM resources to LUTs. The LUT utilisation is also increased by 4%, however the clock frequency is decreased 8% in this case.

The results from the design space exploration illustrate a two to fourfold increase in possible Pareto-optimal implementations over the single point solution provided by the Quartus II synthesis system. These results illustrate the applicability of the ROM to DSP block transfer in real designs.

5.2. Generated Data Sets

While the results of Section 5.1 demonstrate the efficiency of the proposed ROM to DSP block technique in real designs, this section investigates the limits of the technique in terms of what types of ROM data result in successful transfers.

The total number of possible combinations for data present in a given ROM is 2^{d2^a} where d represents data bus width. In the general case any technique based on exploiting trends within the data will produce new Pareto-optimal designs in a minority of cases, although in practical applications the technique performs well, as the results in Section 5.1 have shown. In the past UPP approximation [1] has been used for function evaluation, and it could conceivably be used in place of the proposed technique in the synthesis system of Section 4.

To compare the types of data set which may be successfully applied to the proposed ROM to DSP block transfer technique with UPP, synthetically generated data has been used. To produce these artificial ROM data sets, third order versions of the architecture represented in Figure 2 were generated with three different types of randomised coefficient selection masking, in addition to randomised address bus modulus and coefficient values. The mask types used for generating the data sets were standard UPP approximation type masks and masks including most significant bits only. The ROMs generated had a 10 bit address bus and 16 bit data bus. Data sets formed from the output of 100 architectures of each of these mask types and 100 random data sets were applied to the technique of Section 3 and an optimal UPP approximation. Both types of approximation were allowed to range from zeroth to fourth order, the actual order was selected to reduce LUT usage after transfer for a particular data set. The comparison of the results are shown in Figure 5 in terms of estimated LUT usage after transferring ROM resources.

The data generated using random UPP mask sets showed improvements in LUT utilisation using the proposed transfer architecture over standard UPP approximation in 43% of cases. In these cases a 17% improvement in estimated LUT utilisation after performing the ROM to DSP Block transfer was observed. The UUP sub-graph of graph of Figure 5 shows clear stratification of the results, this due to the discrete orders available to the ROM replacement. The MSB mask sets show an improvement in LUT utilisation using the proposed technique over UPP in 41% of the data sets, a similar figure to UPP mask sets. For these designs an average 19% improvement in LUT usage has been achieved, again similar to data derived from UPP mask sets.

Random data sets behave as expected, with neither technique performing well. In no case does the proposed technique out perform UPP approximation.

Overall the proposed ROM to DSP block transferal technique obtains LUT savings over UPP of up to 51% in the test designs. These results bear out the findings of Section 5.1 that the ROM to DSP block transfer technique is applicable to a guided synthesis system for practical designs, where the stored data is at least partially correlated.

6. CONCLUSIONS AND FURTHER WORK

This paper has presented a ROM to DSP block resource transferal technique using a hybrid polynomial approximation based approach. This method has further been used in a synthesis system which manages resource utilisation within a design, allowing heterogeneous resource based synthesis constraints. This synthesis method was implemented with a modified Quartus II design-flow using the QUIP framework.

Three benchmark designs were applied to the system and the multi-dimensioned LUT, ROM bit, clock period, DSP block design space was expanded in each case, with significant reductions in ROM usage. An average threefold increase over the solutions provided by the standard Quartus II flow was found by using benchmark algorithms. In tests with synthetic ROM contents the proposed ROM to DSP block technique has been shown to offer LUT savings over UPP approximation of up to 51%.

Further work in this area will be to examine the design space of more benchmark algorithms using synthesis system, and also to fully automate all possible transformations. In addition the technique could be extended to constrain other heterogeneous resource types.

7. REFERENCES

[1] T. Pavlidis and A. Maika, "Uniform piecewise polynomial approximation with variable joints," *Journal of Approximation Theory*, vol. 12, pp. 61–69, 1974.

[2] D. Andrews, D. Niehaus, and P. Ashenden, "Programming models for hybrid CPU/FPGA chips," *IEEE Computer*, vol. 37, no. 1, pp. 118–120, 2004.

[3] G. W. Morris, G. A. Constantinides, and P. Y. K. Chenug, "Migrating functionality from ROMs to embedded multipliers," in *Proc. IEEE Symposium on Field Programmable Custom Computing Machines*, 2004.

[4] S. Wilton, "Heterogeneous technology mapping for area reduction in FPGAs with embedded memory arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 1, pp. 56–68, January 2000.

[5] S. Wilton, "Implementing logic in FPGA memory arrays: Heterogeneous memory architectures," in *Proc. IEEE International Conference on Field-Programmable Technology*, 2002.

[6] J. Cong and S. Xu, "Performance-driven technology mapping for heterogeneous FPGAs," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 19, no. 11, pp. 1268–1281, 2000.

[7] J. Aravena and S. Soh, "Architectures for polynomial evaluation," in *Proc. 21st Southeastern Symposium on System Theory*, 1989.

[8] Altera Corporation, *Quartus II University Program (QUIP) Version 2.1*, Altera, 2004.

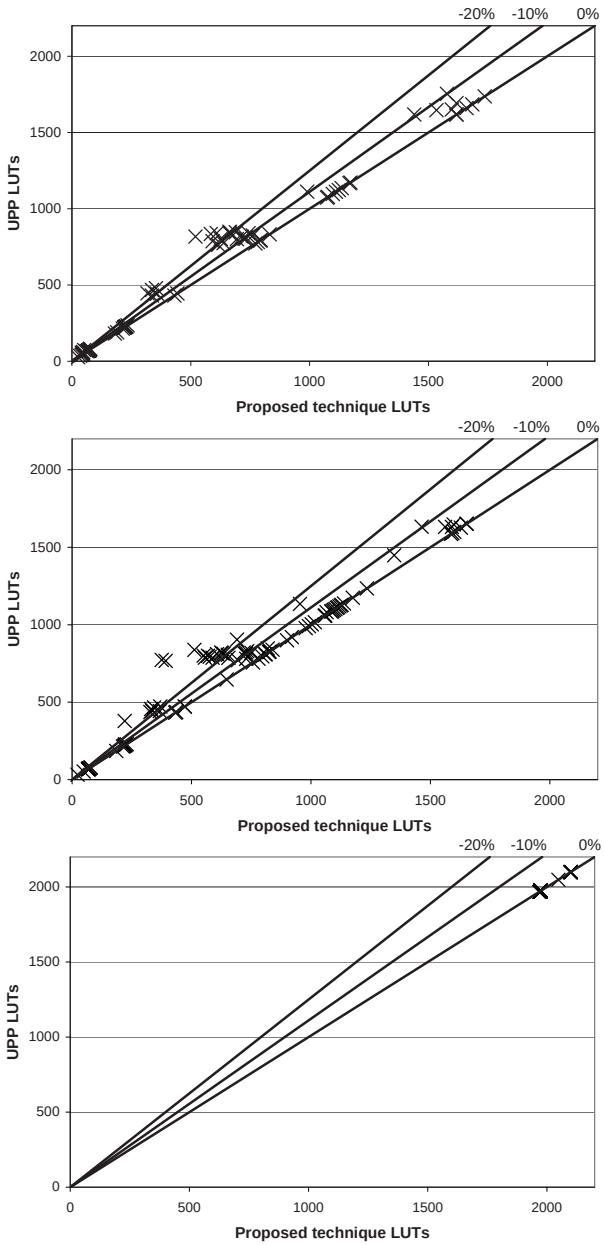

Fig. 5. Comparison of UPP approximation with the proposed polynomial based technique for different data sets. The data sets compared to, from top to bottom, are UPP masked, MSB masked, and pure random. The overlay represents improvements relative to UPP approximation.