Салищев Сергей Игоревич

УДК 519.651, 004.315

КУСОЧНО-ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ С СОКРАЩЕННЫМИ ТАБЛИЦАМИ И ГАРАНТИРОВАННОЙ ТОЧНОСТЬЮ

Аннотация

В статье предлагаются улучшения аппаратной реализации кусочно-полиномиальной аппроксимации в полупроводниковых устройствах. Предлагается новая оценка ошибки интерполяции полиномами малой степени с поточечными ограничениями на равноотстоящих узлах. Предлагается метод уменьшения размера таблиц и почти оптимальной квантизации коэффициентов с использованием межсегментных ограничений и смешанного целочисленного программирования, обеспечивающий заданную точность аппроксимации. Демонстрируется 60 % сокращение размера таблиц по сравнению с методом без использования межсегментных ограничений. Результаты логического синтеза полупроводниковой схемы демонстрируют существенное влияние уменьшения размера таблиц на площадь устройства.

Ключевые слова: кусочно-полиномиальная аппроксимация, интерполяция, числа Лебега, логический синтез полупроводниковых схем.

1. INTRODUCTION

Hardware blocks for calculating smooth functions arecommon in many hardware designs related to DSP, 2D and 3D graphics and computer vision. Industrial component libraries contain implementations for such blocks [6; 7]. Piecewise polynomial approximation is an architecture of choice [5; 8] for computation accuracies higher than 18 bits as it providesa good trade-off between latency and area.

On each interval the target function isusually approximated by min-max or orthogonal interpolation polynomials as these polynomials provide strong error bounds for quantized coefficients [1; 4].

It's desirable to apply additional constraints on the values and derivatives of polynomials on the boundaries of adjacent segments. It allows sharing the table data between segments reducing the table size. This complicates the quantization problem as the constrained polynomial may be quite far from min-max.The quantization effects on polynomial coefficients and computations should be accounted analytically, which is a complex mathematical task, or through exhaustive verification, which is impossible for high accuracies.

Strollo et al. [5] propose an effective empirical method for finding optimal table bit-width for piecewise polynomials with constraints on 2 adjacent segments and pointwise constraints on polynomial values on a uniform grid using mixed integer programming. Constraints on adjacent segments allow data sharing between segments and reduce table size by up to 40 %.

In this paper a strong posterior error bound is provided for any polynomial approximation

of a function with limited second derivative on a closed interval with pointwise constraints on polynomial values on a uniform grid. It allows extending method [5] with additional constraints on the polynomial values and derivatives for further table size reduction. The new error bound guarantees method accuracy for conservative data path quantization. So it can be used without exhaustive verification and so it is applicable for arbitrarily high bit-width.

New constraints on polynomial values and derivatives on 4 adjacent segments are proposed. The method has been evaluated using the quadratic piecewise interpolation for a couple of elementary functions. The case study shows that applying both types of constraints can save up to 60 % of table bit-width compared to the unconstrained case. The architecture for one function was synthesized from SystemC to gate level using High-Level Synthesis and RTL synthesis tools. The comparison shows that additional constraints lead to noticeable design area reduction compared to the method with only 2 segment constraints.

Section 2 gives an overview of piecewise polynomial approximation and error analysis. Section 3 provides a description of the proposed algorithm, a posterior error bound and constraints on multiple segments for table size reduction. Section 4 describes implementation results for a trigonometric function and a natural logarithm for accuracies 24–32 bits. Section 5 compares the gate level synthesis results for different constraints and architectures for trigonometric functions. Section 6 summarizes the results and section 7 concludes the paper.

2. PIECEWISE POLYNOMIAL APPROXIMATION

The typical formulation of the function approximation problem is the following: for given $f(x)$ defined on $[a, b]$ we need to build $g(x)$, with the error strictly less than 1 unit in last place of fixed-point binary representation of the result.

$$\|f - g\| < \alpha,\ \alpha = 1\ ulp,\ ulp = 2^{-k}. \qquad (1)$$

Here and below $\|f\| = \|f\|_{\infty,S} = \sup_{x \in S} |f(x)|$.

The piecewise polynomial approximation is a low-order polynomial approximation on multiple segments with different polynomial on each interval. The polynomial coefficients are tabulated. We map each segment on [–1, 1].

$$f_i(x) = f\left(\frac{a_{i+1} + a_i}{2} + x\frac{a_{i+1} - a_i}{2}\right),$$
$$x \in [-1, 1],\ i \in [0..M-1], \qquad (2)$$

$$a_0 = a,\ a_M = b,\ a_{i+1} > a_i. \qquad (3)$$

Here M is a number of segments, $f_i(x)$ is a remapped piece of $f(x)$ on a segment with index i.

We need to implement architecture for calculating polynomial $p_i(x) = \sum_{k=0}^{N} c_{ik} x^k$ approximating $f_i(x)$, $x \in [-1, 1]$, where N is an approximation order. The set of $\{p_i(x)\}$ should be optimal in terms of table size.

The multiplications and additions in data-path for polynomial calculation should be quantized. Let $g_i(x)$ be a quantized polynomial approximation on sub-interval index i, then the error can be represented as

$$\|f_i - g_i\| \le \|f_i - p_i\| + \|p_i - g_i\| = \varepsilon_{meth} + \varepsilon_{quant},$$
$$i \in [0..M-1]. \qquad (4)$$

Here ε_{meth} is the error bound for the approximation method calculated accurately and ε_{quant} is the bound for error introduced by limited precision of implementation. The task of data-path quantization can be solved separately either manually or by an auto-quantization tool. The data-path quantization is conservative if the following inequity holds

$$\varepsilon_{quant} + \varepsilon_{meth} < \alpha. \qquad (5)$$

In this case the resulting architecture will have guaranteed accuracy and the exhaustive verification will not be required. So the approach can be used for arbitrarily high precisions. For the datapath calculating a class of non-constant polynomials the quantization error includes errors for final rounding to 1 *ulp* and intermediate term quantization using l guard bits. Here we assume no quantization of x, so we calculate $g_i(x)$ only in exactly representable input points

$$\varepsilon_{quant} \ge \frac{\alpha}{2}(1 + 2^{-l}). \qquad (6)$$

We need to choose the number of guard bits l and approximation $\{p_i(x)\}$ fulfilling ε_{meth} requirement

$$\varepsilon_{meth} < \frac{\alpha}{2}(1-2^{-l}). \quad (7)$$

Then we need to find the datapath quantization fulfilling (5). As the points with maximum error for $\{p_i(x)\}$ and for data-path quantization rarely coincide, it's possible to break the conservative requirement and further reduce the width of data-path intermediate types. Doing so requires exhaustive verification and is not applicable for high accuracy due to the test bench running time.

3. APPROXIMATION POLYNOMIALS WITH POINTWISE VALUE CONSTRAINTS

Let's consider the polynomial $p(x)=\sum_{k=0}^{N} c_k x^k$ approximating $f(x)$ on the interval [–1, 1] with error ε.

$$\|f-p\| = \varepsilon. \quad (8)$$

In the context of piecewise approximation $p(x)$ is an approximation polynomial for one segment. We would like to represent its coefficients with minimal bit-width. This problem is hard to solve. We consider a simpler problem by applying the constraint only to a limited number of points $P+x$:

$$\max|f(x_i)-p(x_i)| = \varepsilon_0,\ x_0=-1,$$
$$x_P=1,\ x_{i+1}>x_i,\ i\in[0..P]. \quad (9)$$

Section 3.1 describes the way to estimate ε independently of $p(x)$,

$$\varepsilon \le \gamma + \varepsilon_0 = \varepsilon_{meth}. \quad (10)$$

We can provide a constructive procedure for choosing $p(x)$ and ε_0 to fulfill the method requirements (7):

1. Choose piece-wise approximation segment and $f(x)$ for this segment.

2. Find $\varepsilon^* = \min\max|f(x_i)-p^*(x_i)|$, $p^*(x)=\sum_{k=0}^{N} c_k^* x_k$ using Linear Programming (LP).

3. Calculate γ using (28).

4. Choose the number of guard bits l such that $\varepsilon^* < \frac{\alpha}{2}(1-2^{-l})-\gamma$.

5. If cannot complete step 4 increase P or reduce segment size, repeat form step 1.

6. Set $\varepsilon_0 = \frac{\alpha}{2}(1-2^{-l})-\gamma$.

7. Choose $\{d_k\}\subset Z$ fractional word lengths for coefficients $\{c_k\}$.

8. Find $\{c_k\}$ for which $\{2^{d_k}c_k\}\subset Z$ and $|f(x_i)-p(x_i)|\le\varepsilon_0$ using mixed integer programming.

9. Repeat from step 7 until minimal total bit-widthis $w^* = \min w$, $w=\sum_{k=0}^{N}\left\lceil \log_2 2^{d_k} c_k \right\rceil$ is found using branch-and-bound strategy.

10. If w^* is infeasible for implementation, increase P or reduce segment size, repeat form step 1.

By construction the above algorithm provides near optimal bit-width coefficients $\{c_k\}$ of approximation polynomial for a given α. Supposing small changes of c_k between optimization steps we can achieve good results by optimizing $d=\sum_{k=0}^{N} d_k$ instead of w. Details on applying LP are given in section 3.2. Section 3.3 describes a technique of table size reduction by applying additional constraints to LP.

3.1. ERROR BOUND FOR UNIFORM GRID

To make the previous algorithm work we need to estimate γ which limits the deviation of between known points. Let, f have continuous first and second derivatives.

Lemma 1. *If* $p(x)=\sum_{k=0}^{N} c_k x^k$ *approximates* $f(x)$ *on a set of interpolation nodes* $\{x_k\}_{k=1}^{N+1}$ *on* [–1, 1] *then*

$$\|f''-p''\| \le \frac{2^{N-1}}{(N-2)!}\|f''\| + \lambda_{(N+1),2}(\{x_i\})\max|f(x_i)-p(x_i)|. \quad (11)$$

Here $\{\lambda_{r,v}\}_{v=0}^{r-1}$ *is a set of generalized Lebesgue constants characterizing the set of interpolation nodes* $\{x_k\}_{k=1}^{N+1}$, *the usual Lebesgue constant is* $\lambda_r=\lambda_{r,0}$ *and* $r=N+1$

$$\lambda_{r,v} = \sup_{x\in[-1,1]}\sum_{k=1}^{r}\left|l_{r,k}^{(v)}(x)\right|,$$
$$0\le v<r,\ r\ge 1,\ x\in[-1,1]. \quad (12)$$

$$l_{r,k}(x)=\frac{l_r(x)}{(x-x_k)l_r'(x_k)},$$
$$l_r(x)=\prod_{k=1}^{r}(x-x_k), \qquad (13)$$

Here $\{I_{r,k}(x)\}$ *is a set of fundamental polynomials of Lagrange interpolationon* $\{x_i\}_{i=1}^{N+1}$ *nodes.*

Proof:

The proof is due to equation (10.10) in [4].

Lemma 2. *Derivatives of fundamental polynomials of Lagrange interpolation have a form*

$$l_{r,k}'(x)=\sum_{m=1,m=k}^{r}\frac{l_{(r-1)}^{m},k^{(x)}}{x_m-x_k},$$
$$l_{(r-1),k}^{m}(x)=\frac{(x_m-x_k)}{x-x_m}l_{r,k}(x). \qquad (14)$$

Here $l_{(r-1),k}^{m}(x)$ *is a fundamental polynomial of lower order interpolation with one node removed.*

Proof is by definition of fundamental polynomials (13).

Lemma 3. *For* 3 *and* 4 *equidistant nodes on* [–1, 1] *used in quadratic and cubic interpolation Lebesgue constants for the second derivative have the following upper bounds*

$$\lambda_{3,2}\le 6,\ \lambda_{4,2}\le 162. \qquad (15)$$

Proof:

For the quadratic case compute the second derivatives of fundamental polynomials using lemma 2, by definition

$$\lambda_{3,2}=\sum_{k=1}^{3}\left\|\sum_{m=1,m\neq k}^{3}\sum_{q=1,q\neq m,q\neq k}^{3}\frac{l_{1,1}(x)}{(x_m-x_k)(x_q-x_m)}\right\|. \qquad (16)$$

For 3 equidistant nodes on [–1, 1], $|x_m-x_q|\ge 1$, $m\neq q$ and $l_{1,1}=1$.

For cubic case computations are slightly more difficult

$$\lambda_{4,2}=\sum_{k=1}^{4}\left\|\sum_{m=1,m\neq k}^{4}\sum_{q=1,q\neq m,q\neq k}^{4}\frac{l_{2,q}^{k,m}(x)}{(x_m-x_k)(x_q-x_m)}\right\|. \qquad (17)$$

Here $l_{2,q}^{k,m}$ is a first order fundamental polynomial with 2 interpolation nodes removed.

$$\frac{1}{|x_k-x_q|}\le\frac{3}{2},\ q\neq k \qquad (18)$$

$$\|l_{2,k}\|=\left\|\frac{x-x_q}{x_k-x_q}\right\|\le 3,\ q\neq k. \qquad (19)$$

Lemma 4. *Lebesgue constants are invariant for linear transformation of variables. Consider linear mapping between* [–1, 1] *and* [*a*, *b*].

$$x=\frac{b-a}{2}s+\frac{b+a}{2}. \qquad (20)$$

The corresponding Lebesgue constant is

$$\hat{\lambda}_{r,v}=\left(\frac{b-a}{2}\right)^{v}\lambda_{r,v}. \qquad (21)$$

Proof:

Directly follows from definition of the generalized Lebesgue constants (12).

Theorem 1. *For* $\{x_i\}$ *which is uniform grid on* [–1, 1] *with step* $\delta=x_{i+1}-x_i$, $x_0=-1$, $x_P=1$, *there exists* γ *fulfilling* (10) *which is independent of* $p(x)$ *and only depends on* $f(x)$ *and* δ*:*

$$\gamma=\frac{2^{N-4}\delta^2}{(N-2)!}\|f''\|+\frac{\delta^2}{8}\varepsilon_0\lambda_{(N+1),2}, \qquad (22)$$

Here $\lambda_{r,v}$ *is the generalized Lebesgue constant* (12).

Proof:

Consider the point the approximation error is maximal, on a compact it always exists:

$$\varepsilon=\|f-p\|=\left\|f(x^*)-p(x^*)\right\|. \qquad (23)$$

There are 2 possible cases, if it lies on the interval boundary $x^*\in\{-1,1\}$, then $\varepsilon=\varepsilon_0$ and $\gamma=0$, as –1, 1 are elements of the grid.

Otherwise $x^*\in(-1,1)$, in this case x^* is the extreme point of a smooth function and so

$$f'(x)-p'(x)=0. \qquad (24)$$

Let x_k be a nearest point in a grid to x^*. So the distance between points is

$$\sigma=|x^*-x_k|\le\frac{\delta}{2}. \qquad (25)$$

Let's consider the first 2 elements of Taylor expansion for x_k in point x^*

$$f(x_k)-p(x_k)=f(x^*)-p(x^*)+\frac{\delta^2}{2}(f''(\psi)-p''(\psi)),\quad \psi\in[x^*,x_k]. \tag{26}$$

By replacing the last term with the estimate from lemma 1 we receive

$$\varepsilon\le\varepsilon_0+\frac{2^{N-4}\delta^2}{(N-2)!}\|f''\|+\frac{\delta^2}{8}\varepsilon_0\lambda_{(N+1),2}(\{x_i\}_{i\in I_0}). \tag{27}$$

Here $I_0\subset[0..P]$, $|I_0|=N+1$ is a subset of indexes defining set of interpolation nodes with minimal Lebesgue constant.

Corollary. *For the uniform grid* $\{x_i\}_{i=0}^{P}$ *on* [–1, 1] *and interpolation order* = 2, 3 *and* $P \bmod N=0$,

$$\varepsilon-\varepsilon_0\le\gamma+\frac{2^{N-4}\delta^2}{(N-2)!}\|f''\|+\frac{\delta^2\varepsilon_0\lambda_{(N+1),2}}{8},\quad \lambda_{3,2}\le 6,\ \lambda_{4,2}\le 162. \tag{28}$$

Proof:

As the grid contains $N+1$ equidistant nodes the proposition directly follows from Theorem 1 and Lemma 3.

3.2. LP AND ILP METHODS FOR FINDING POLYNOMIAL COEFFICIENTS

For the sake of simplicity we will consider that the approximation segment is mapped to [0, 1]. Due to Lemma 4 all the above results still apply. We need to solve the following optimization problem.

$$\begin{cases}\max|f(x_i)-p(x_1)|=\varepsilon,\\ x_0=0,x_P=1,x_{i+1}>x_i,i\in[0..P]\\ \varepsilon\to\min,\ p(x)=\sum_{k=0}^{N}c_kx^k\end{cases} \tag{29}$$

For using the linear programming solver we need to convert it to the canonical form.

Let V be a Vandermonde matrix of order N of points $\{x_i\}$.

$$V=\{x_i^k\}_{i=0,k=0}^{N,P}. \tag{30}$$

The vector of function values is

$$f=\{f(x_0),...,f(x_P)\}. \tag{31}$$

The vector of variables is

$$x=\{c_0,...,c_N,\varepsilon\}=\{c,\varepsilon\}. \tag{32}$$

The minimized function is lx, where $l=\{0,...,0,1\}$. In these terms the canonical linear problem is

$$\begin{cases}\begin{pmatrix}V & -1\\ -V & -1\end{pmatrix}x\le\begin{pmatrix}f\\ -f\end{pmatrix}.\\ lx\to\min\end{cases} \tag{33}$$

To implement the main algorithm we need to solve the mixed integer programming [2] problem with the additional integer constraint $\{2^{d_i}c_i\}\subset Z$. As we know the target ε we don't need to solve the optimization problem and only need to find a base solution satisfying the constraints. So the usual branch-and-bound method degrades to depth-first-search, which is substantially faster.

3.3. USING LINEAR CONSTRAINTS FOR TABLE SIZE REDUCTION

Strollo et al. [5] show that it's possible to share the table data between 2 adjacent approximation segments by exploiting the smoothness of the approximated function.

We consider an $f(x)$, $x\in[-1,1]$. It can be decomposed into 2 halves.

$$f(x)=\begin{cases}f_R(x),x\in[0,1]\\ f_L(-x),x\in[-1,0]\end{cases}. \tag{34}$$

Both f_L, f_R are defined on [0, 1]. Then we consider polynomial approximations for f_L, f_R of order N.

$$p(x)=\sum_{i=0}^{N}p_ix^i,\ q(x)=\sum_{i=0}^{N}q_ix^i. \tag{35}$$

We are interested in $p(x)$, $q(x)$ which share some coefficients. It means the corresponding derivatives at $x=0$ are equal. Case study in [5] shows that for quadratic and cubic case it's possible to share all the coefficients except one of the highest order.

$$p^{(v)}(0)=q^{(v)}(0),\ 0\le v<N. \tag{36}$$

It is equivalent to

$$c_{L,v}=(-1)^vc_{R,v},\ 0\le v<N. \tag{37}$$

We employ the mixed integer programming to find the coefficientsof $p(x)$, $q(x)$. The canonical for the problem is as follows

$$\begin{cases} \begin{pmatrix} V & 0 & -1 \\ 0 & V & -1 \\ -V & 0 & -1 \\ 0 & -V & -1 \end{pmatrix} \begin{pmatrix} c_L \\ c_R \\ \varepsilon \end{pmatrix} \le \begin{pmatrix} f_L \\ f_R \\ -f_L \\ -f_R \end{pmatrix} \\ K \begin{pmatrix} c_L \\ c_R \\ \varepsilon \end{pmatrix} = 0 \\ (0,0,1) \begin{pmatrix} c_L \\ c_R \\ \varepsilon \end{pmatrix} \to \min \end{cases} . \quad (38)$$

Here K is the derivative equality constraint between adjacent segments. For example $K_{2,1}$ for sharing the first 2 coefficients in quadratic approximation

$$K_{2,1} = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \quad (39)$$

We can exploit the smoothness of $f(x)$ even further by grouping 4 adjacent intervals. Let's consider $f(x)$, $x \in [0, 4]$. We can divide it into 4 segments in the following way

$$f(x) = \begin{cases} f_0(1-x), & x \in [0,1] \\ f_1(x-1), & x \in [1, 2] \\ f_2(3-x), & x \in [2, 3] \\ f_3(x-3), & x \in [3, 4] \end{cases} . \quad (40)$$

For these functions we consider approximation polynomials p_0, p_1, p_2, p_3. We can employ the following constraints

$$\begin{cases} p_0^{(v)}(0) = p_1^{(v)}(0) \\ p_2^{(v)}(0) = p_3^{(v)}(0) \\ \left| p_2^{(v)}(1) - p_1^{(v)}(1) \right| \le \delta_v \\ \sum \delta_v \to \min \end{cases}, \quad 0 \le v < N . \quad (41)$$

Improvement by applying the 4 segment constraints vs. 2 segment constraints is smaller compared to 2 segment constraints vs. no constraint. First of all $p^{(v)}(1)$ contains more than 1 summand. So exploiting the data sharing requires adders in the data path. Also δ_v need to be tabulated taking some additional bits. As the highest order coefficients are not changed it's expected that the data sharing will have a minor effect on the critical path timing. It will be shown in the case study that the data sharing is beneficial for architecture area.

4. IMPLEMENTATION

For the implementation the functions resembling Synopsis DesignWare IP blocks DW_sincos [6] and DW_ln [7] were chosen with fractional accuracy ranging from 24 to 32. The first is a $\sin(\pi x)$ and $\cos(\pi x)$ approximation for $x \in [-1,1)$. The second is a natural logarithm $\ln(x+1)$ approximation for $x \in [0,1)$.

Due to symmetry in trigonometric functions it's only needed to approximate $\sin(\pi x)$ for $x \in [0,1/2)$.

For both cases the piecewise quadratic approximation was used with 2 segment constraints,with 4 segment constraints proposed above,and without constraints.A Matlab script was built to generate the tables for both cases.

The following tables show the growth of bit-width of tabulated values with increased accuracy (table 1).

The average bit-width reduction per segment for $\sin x$ for 2 segment constraints is 40 %, for 4 segment constraints is 57 % (table 2).

The average bit-width reduction per segment $\ln x$ for 2 segment constraints is 40 %, and for 4 segment constraints is 60 %, when compared to the unconstrained case.

5. COMPARISON

For the comparison a block compatible with Synopsys DesignWareDW_sincos [6] was implemented using the quadratic piecewise polynomials on optimal number of segments with 4 segment derivative constraints, 2 segment derivative constraints and cubic piecewise polynomials on 64 segments as described in the Synopsys DesignWare trigonometric architecture overview [8]. In all cases the polynomial values were computed directly without using Horner scheme. A conservative data-path quantization has been used.For quadratic methods the SystemC code was synthesized to gate level RTL (table 3, 4).

Table 1. Sin approximation

Accuracy	Guard bits	Segments	Bits per segmentunconstrained	Bits per segment 2 seg. constraints	Bits per segment 4 seg. constraints
24	2	128	57	34.5	25
25	4	128	66	40.5	29.5
26	2	256	60	36	24.5
27	2	256	63	38	27.25
28	4	256	72	44	31.75
29	2	512	66	39.5	26.75
30	3	512	72	43.5	30.5
31	4	512	78	47.5	34
32	2	1024	72	43	29.75

Table 2. Ln approximation

Accuracy	Guard bits	Segments	Bits per segment unconstrained	Bits per segment 2 seg. constraints	Bits per segment 4 seg. constraints
24	2	128	54	32	21.5
25	3	128	60	36	24.5
26	4	128	66	40	27.25
27	2	256	60	35.5	23.25
28	3	256	66	39.5	26.25
29	4	256	72	43.5	29.5
30	2	512	66	39	26.5
31	3	512	72	43	28.5
32	4	512	78	47	32.75

The area comparison in Table 3 shows that the table size has serious impact on the resulting design and that approximations with 4 segment constraints have better area in practice compared to 2 segment constraint designs and cubic design despite additional adders to exploit the data sharing. Table 4 shows that the timing of quadratic polynomial is also smaller by 22–30 % vs. cubic. It is due to one less multiplier on critical path. These numbers exhibit significant variability as they are highly sensitive to low level optimizations applied during gate level synthesis.

6. RESULTS

This paper provides a new error bound for the method of finding the piecewise polynomial approximation with finite precision coefficients of optimal bit width and linear constraints on derivatives for cross-segment data sharing proposed in [5]. The error bound allows guaranteed accuracy independent of the additional linear constraints. So the method can be extended with additional constraints and applied to arbitrary accuracy without exhaustive verification.

Table 3. Design area, clk 5ns

Accuracy (bit)	Quadratic 4 seg.	Quadratic 2 seg.	Cubic no constraint
24	55%	72%	100%
32	68%	84%	100%

Table 4. Best timing

Accuracy (bit)	Quadratic 4 seg.	Quadratic 2 seg.	Cubic no constraint
24	70%	74%	100%
32	78%	74%	100%

New linear constraints on derivatives for 4 adjacent segments are proposed.The case study shows that applying additional constraints substantially reduces the design area compared to the 2 segment constraints case and the cubic interpolation case.

7. SUMMARY

Piecewise polynomial approximation is the method of choice for hardware blocks computing smooth functions with fractional accuracies higher than 18 bits due to balance between performance and design complexity. It is used in multiple research papers and in industry strength component libraries.

The main result of this paper is a practical method of building piecewise polynomial approximation with an optimal table bitwidth for given constraints with a guaranteed accuracy based on solving Integer Linear Programming problem.

In addition to the first and second derivative constraints on 2 adjacent segments, constraints for 4 adjacent segments were added leading to table reduction compared to [5].

For now, only a limited case study has been performed. The results show that the table reduction positively affects the area of the design without noticeable impact on timing.

The effect of datapath quantization hasnot beeninvestigated yet. The manual backward error propagation method was used leading to conservative quantization. It is expected that more aggressive quantization might save a couple of bits from multiplier widths reducing the design area even further.

Bibliography/References

1. *Cheney E., Light W.* A Course in Approximation Theory, New York: Chelsea, 1999.

2. *Gärtner B., Matoušek J.* Understanding and Using Linear Programming, Berlin: Springer, 2006.

3. *Günttner R.* On asymptotics for the uniform norms of the Lagrange interpolation polynomials corresponding to extended Chebyshev nodes, SIAM J. Numer. Anal. Vol. 25 (1988). P. 461–469.

4. *Lokuzievsky O., Gavrikov M.* Numerical Analysis Essentials[in Russian], Moscow: Janus, 1995.

5. *Strollo A.G.M., De Caro, D., Petra, N.* Elementary Functions Hardware Implementation Using Constrained Piecewise-Polynomial Approximations, Computers, IEEE Transactions on. Vol. 60, № 3. P. 418–432, March 2011.

6. Synopsys DesignWareDW_sincos http://www.synopsys.com/dw/doc.php/doc/dwf/datasheets/dw_sincos.pdf (date 30.10.2012).

7. Synopsys DesignWareDW_ln http://www.synopsys.com/dw/doc.php/doc/dwf/datasheets/dw_ln.pdf (date 30.10.2012).

8. Synopsys DesignWare trigonometry overview http://www.synopsys.com/dw/doc.php/doc/dwf/datasheets/trig_overview.pdf (date 30.10.2012).

Abstract

An improvement to the piecewise polynomial approximation in hardware is proposed. A new error bound is given for the low-order polynomial interpolation with pointwise constraints ona uniform grid. A method of table size reduction and near optimal quantization of coefficients using intersegment constraints and mixed integer programmingwith guaranteed accuracy is proposed.A case study shows up to 60% table size reduction compared to unconstrained polynomials. Gate level RTL synthesis shows that table reduction has noticeable impact on the design area.

Keywords: piecewise polynomial approximation, interpolation, Lebesgue numbers, RTL synthesis.

Салищев Сергей Игоревич,
старший преподаватель кафедры информатики СПбГУ,
инженер лаборатории Intel,
sergey.i.salishev@gmail.com.