Engineers Know How to Put Math to Work

Student 1, Student 2
Department of Electrical and Computer Eng.
University of Utah, Salt Lake City, UT-84112
{myemail}@utah.edu

***Abstract*—Galois fields have wide applications in the VLSI domain. This paper describes a new approach to verify VLSI implementations of digital circuits using Gröbner basis methods. The verification problem is modeled as proving the infeasibility of a miter using Hilbert's Nullstellensatz. Buchberger's algorithm is then applied to reason about the variety of ideals corresponding to the VLSI circuits. Experiments demonstrate the superiority of this method against contemporary SAT-based verification when applied to datapath dominated applications.**

I. Introduction

To compile this document, run the following commands:

```
prompt> latex latex-for-class
prompt> bibtex latex-for-class
prompt> latex latex-for-class
prompt> latex latex-for-class
prompt> dvips -o latex-for-class.ps latex-for-class
prompt> ps2pdf latex-for-class.ps
```

This will create the latex-for-class.pdf file. You have to run latex a couple of times to get cross-references resolved.

A. Math Symbols

This is *italics*, **bold font**. This is how we use math mode:

$$A = a_0 + a_1\alpha + a_2\alpha^2 = \sum_{i=0}^{2} a_i \cdot \alpha^i \tag{1}$$

This is also how to use in-line math mode: $\mathbb{F}, \mathbb{R}, \mathbb{Q}, \mathbb{C}, \mathbb{F}_q, \mathbb{F}_{2^k}$, based on my macros.

Let $J = \langle f_1, \ldots, f_s \rangle = J = \langle g_1, \ldots, g_t \rangle \subseteq \mathbb{F}_q[x_1, \ldots, x_n]$ and let $V(J)$ denote the variety V of ideal J. Actually, since variety is needed over $\mathbb{F}_q$ itself, use $V_{\mathbb{F}_q}(J)$.

This is how you write an algorithm and refer to it as Alg. 1, see the caption below the algorithm description in the intro.tex file.

This is how you write the polynomial reduction of $f \pmod G$: $f \xrightarrow{g_1,\ldots,g_t}_+ r$ where $G = \{g_1, \ldots, g_t\}$. Also, there are many ways to write a matrix, two of them are:

$$M = \begin{pmatrix} & x^2y & y^3 & y^2 & y \\ f_4 & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ yf_1 & 2 & 0 & 1 & 0 \\ yf_3 & 0 & 4 & 0 & -1 \end{pmatrix}$$

ALGORITHM 1: Buchberger's Algorithm

```
**Input**: $F = \{f_1, \ldots, f_s\}$
**Output**: $G = \{g_1, \ldots, g_t\}$
$G := F$;
**repeat**
    $G' := G$;
    **for** *each pair* $\{f, g\}, f \neq g$ *in* $G'$ **do**
        $Spoly(f, g) \xrightarrow{G'}_+ r$ ;
        **if** $r \neq 0$ **then**
            $G := G \cup \{r\}$ ;
        **end**
    **end**
**until** $G = G'$;
```

$$M = \begin{array}{c} \\ f_4 \\ yf_1 \\ f_3 \end{array} \begin{array}{c} \begin{array}{cccc} x^2y & y^3 & y^2 & 1 \end{array} \\ \begin{pmatrix} \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 4 & 0 & -1 \end{pmatrix} \end{array}$$

Now, reducing M to a row echelon form using Gaussian elimination gives:

$$M = \begin{array}{c} \\ f_4 \\ h = f_4 - \frac{1}{6}yf_1 \\ r = h - \frac{1}{8}f_3 \end{array} \begin{array}{c} \begin{array}{cccc} x^2y & y^3 & y^2 & 1 \end{array} \\ \begin{pmatrix} \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & -\frac{1}{6} & 0 \\ 0 & 0 & -\frac{1}{6} & \frac{1}{8} \end{pmatrix} \end{array}$$

This is how you provide citation for a journal paper [1], conference paper [2], book [3] or a PhD thesis [4].

Theorem 1.1: This theorem states that Prof. Kalla is indeed the best.

Proof 1: The proof is trivial.

Corollary 1.1: No one is better than Prof. Kalla.

Finally, this is how you include a figure as shown in Fig. 1.

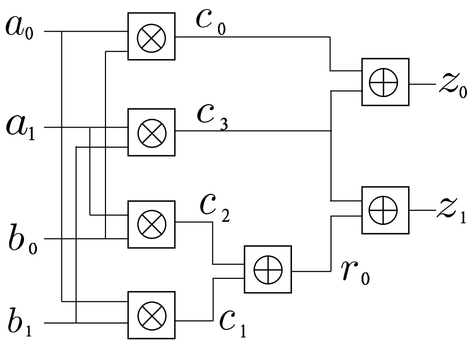

Fig. 1: This is a figure

II. Experimental Results

When you complete your project, your results should compare like this. If not, all of you will fail the course. As shown in Table I, both SINGULAR and our $F4$ approach can verify the correctness of up to 163-bit Mastrovito multipliers – corresponding to the practical NIST-specified Galois field $\mathbb{F}_{2^{163}}$. However, our $F4$-style approach is almost $2.5X$ faster.

TABLE I: Runtime for verifying bug-free and buggy Mastrovito multipliers using our approach. TO = timeout of 10hrs. Time is given in seconds.

Operand size k:	32	64	96	128	160	163
#variables	1155	4355	9603	16899	26243	27224
#polynomials	1091	4227	9411	16643	25923	26989
#terms	7169	28673	64513	114689	179201	185984
Bug-free (Singular)	1.41	112.13	758.82	3054	9361	16170
Bug-free (F_4)	0.83	39.23	243.16	1138	3496	6537
Bugs (Singular)	1.43	114.86	788.65	3061	9384	16368
Bugs (F_4)	0.84	40.01	249.84	1152	3530	6592

References

[1] M. Ciesielski, P. Kalla, and S. Askar, "Taylor Expansion Diagrams: A Canonical Representation for Verification of Data-Flow Designs," *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1188–1201, 2006.

[2] N. Shekhar, P. Kalla, M. B. Meredith, and F. Enescu, "Simulation Bounds for Equivalence Verification of Arithmetic Datapaths with Finite Word-Length Operands," in *Formal Methods in Computer Aided Design*, November 2006, pp. 179–186.

[3] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. New York, USA: Springer, 2007.

[4] B. Buchberger, "Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal," Ph.D. dissertation, University of Innsbruck, 1965.