ECE/CS 5745/6745: Testing and Verification of Digital Circuits

Fall 2025, Homework # 5
Prepared by *Priyank Kalla*
Due Date: Nov 30, by midnight.

- Everyone should solve Q1 and Q2.
- Solve either [Q3 and Q4 (Hardware design Verification)] OR [Q5 and Q6 (Reduction Algorithm and Gröbner Bases)].
- You can solve both sets for extra credit ☺.

- (Reading Assignment) Please read the book chapter I've provided on Galois fields and hardware design, uploaded on the class website.
- This HW assignment gives you some practice with the following concepts: (i) Some basic concepts of Galois fields (GF); (ii) design of GF circuits; (iii) setting up verification problems in Singular; (iv) the Gröbner basis algorithm; (v) the Weak Nullstellensatz.
- For all computations, you should use the Singular computer algebra tool, of course.
- Singular is installed on the CADE lab machines under `'/usr/local/bin/Singular'`. Notice that the 'S' in Singular is upper-case. Feel free to download the latest version on your own personal computers for use. Details are on our class website.
- Along with this HW, I am uploading some example Singular files, particularly the ones that I used to give you a demo in class.
- On a unix terminal, the way to load a singular script file is as follows:

```
prompt>> Singular
                     SINGULAR                                /
 A Computer Algebra System for Polynomial Computations   /   version 3-1-1
                                                       0<
     by: G.-M. Greuel, G. Pfister, H. Schoenemann        \   Feb 2010
FB Mathematik der Universitaet, D-67653 Kaiserslautern    \
> < "finite-field-demo.sing";
```

Okay, so the HW questions are as follows:

1) **(Understanding Vanishing Polynomials and field containment – a free gift of 10 points)**. In class, we have seen that for any field $\mathbb{F}_q$, we have that $x^q = x$ or $x^q - x = 0$ for all $x \in \mathbb{F}_q$. We call $x^q - x$ as the vanishing polynomial (or the field polynomial) of $\mathbb{F}_q$, as every element of $\mathbb{F}_q$ is a root of $x^q - x$. Note that any element outside of $\mathbb{F}_q$ may not satisfy $x^q - x = 0$. Now, let $\mathbb{F}_{q1}$ and $\mathbb{F}_{q2}$ be two finite fields such that $\mathbb{F}_{q1} \subset \mathbb{F}_{q2}$. Then for all elements $x \in \mathbb{F}_{q1}$, we will have $x^{q1} = x$. Similarly, for all $y \in \mathbb{F}_{q2}$, we will have $y^{q2} = y$. However, we may or may not have $y^{q1} = y$. You will confirm this with the following experiment:
   In my book chapter, consider Ex. 1.1 and 1.2, along with Fig. One.1 in Section IV. Here we construct $\mathbb{F}_{16} = \mathbb{F}_2[x] \pmod{P(x) = x^4 + x^3 + 1}$, with $P(\alpha) = 0$. We found that $\alpha^5, \alpha^{10} \in \mathbb{F}_4 = \mathbb{F}_2[x] \pmod{x^2+x+1}$. Show that α^5, α^{10} satisfy (are the roots of) $x^4 - x$, whereas any element <u>outside</u> of $\mathbb{F}_4 = \{0, 1, \alpha^5, \alpha^{10}\}$ does not satisfy $x^4 - x$, but it satisfies $x^{16} - x$. Use Singular for this demonstration.

2) **(Fundamentals of the Weak Nullstellensatz, and counting the number of solutions for zero-dimensional ideals using standard monomials of an ideal – 30 points)**. Consider the finite field of 5 elements $\mathbb{F}_5 = \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$. Let $\overline{\mathbb{Z}_5}$ be its algebraic closure. Let polynomials $f_1 = x^2 + y^2 + 1, f_2 = x^2y + 2xy + x$, where $f_1, f_2 \in \mathbb{Z}_5[x, y]$. Answer the following questions.
   a) Describe in your own words: what is the algebraic closure ($\overline{\mathbb{Z}_5}$) of $\mathbb{Z}_5$?
   b) Does the system of polynomial equations $\{f_1 = 0, f_2 = 0\}$ have common solutions over the closure $\overline{\mathbb{Z}_5}$? In other words, is the variety $V_{\overline{\mathbb{Z}_5}}(f_1, f_2) = \emptyset$? Answer this question without actually solving for the roots of f_1, f_2.
   c) If the system of polynomial equations $\{f_1 = 0, f_2 = 0\}$ does have solutions over $\overline{\mathbb{Z}_5}$, is the number of solutions finite or infinite? If the number of solutions is finite, then count the number of solutions over $\overline{\mathbb{Z}_5}$. Once again, you have to apply Nullstellensatz concepts without actually solving for the roots.
   d) You are now asked to find if the system of polynomial equations $\{f_1 = 0, f_2 = 0\}$ does have solutions over the field $\mathbb{Z}_5$ itself? In other words, is $V_{\mathbb{Z}_5}(f_1, f_2)$ empty or non-empty? If non-empty, count the number of solutions in $\mathbb{Z}_5$, i.e. what is $|V_{\mathbb{Z}_5}(f_1, f_2)|$?
   e) Explain how you arrived at your answers. Attach your (cleaned-up and well commented) Singular experiments to your solutions.

3) **(GF multiplier design, Miter construction, and Equivalence Check – 30 points)** In my slides and in my book chapter on finite fields, I have shown you how to design a Mastrovito multiplier circuit that performs multiplication $Z = A \cdot B \pmod{P(x)}$, where $A = \{a_{k-1}, \ldots, a_0\}, B = \{b_{k-1}, \ldots, b_0\}$ are 2 k-bit inputs, $Z = \{z_{k-1}, \ldots, z_0\}$ is the k-bit output and $P(x)$ is the given primitive polynomial. In the slides, I have given you a design of a 4-bit circuit, as well as that of a 2-bit circuit. In addition, the book chapter also shows a circuit schematic for a 4-bit Mastrovito multiplier. Study these multiplier design concepts carefully, and then solve the following:

   a) Design a 3-bit Mastrovito multiplier over the Galois field $\mathbb{F}_8 = \mathbb{F}_2[x] \pmod{P(x)}$ using $P(x) = x^3 + x + 1$.

   b) In the lecture slides, I have also shown you how to construct a miter between a polynomial *Spec* and a circuit implementation. Moreover, on the class webpage, along with this HW, I have also uploaded a file '2-bit-multiplier.sing' that shows: i) how to create/write an algebraic miter in Singular; ii) declare the ideal J generated by the miter's polynomials; iii) generate ideal $J_0 = \langle x_i^2 - x_i, A^q - A, B^q - B, Z^q - Z, t^q - t \rangle$ where x_i's are the bit-level variables and A, B, Z, t are (word-level) variables that take values in the field; and iv) compute the Gröbner basis 'G = groebner(J + J_0)'. Study this file and execute it in Singular to interpret the result.

   c) Create a similar 'algebraic miter' between your 3-bit GF multiplier and the *Spec* polynomial $f_{spec} : Z + A \cdot B$. Describe the ideal J generated by the miter's polynomials and J_0 as the ideal of vanishing polynomials. Describe this ideal $J + J_0$ corresponding to the miter as a Singular file.

   d) Using this Singular file, and the Weak Nullstellensatz, formulate the equivalence check on this miter to ascertain whether your design correctly implements the *Spec* . Briefly describe your problem formulation, and the outcome of this experiment.

   e) Now purposely introduce a bug in the design – say, by changing some gate – and then prove that there is indeed a bug in the design.

   f) Submit your circuit schematic, the polynomial ideal, the singular file, its output, and the conclusion from your experiments.

4) **(Verification of an arbitrary circuit using the Weak Nullstellensatz over finite fields – 30 points)** Consider

the function (mapping) $f : \mathbb{B}^3 \to \mathbb{B}^3$ shown in the truth-table below.

$A = \{a_2 a_1 a_0\}$	$\mapsto$	$Z = \{z_2 z_1 z_0\}$
000	$\mapsto$	000
001	$\mapsto$	001
010	$\mapsto$	111
011	$\mapsto$	111
100	$\mapsto$	101
101	$\mapsto$	011
110	$\mapsto$	101
111	$\mapsto$	101

- Using Karnaugh-maps (or any other method) design a Boolean logic circuit that implements the function. Draw the circuit schematic and give Boolean equations for the outputs.
- Now you will verify that the implemented circuit is equivalent to the truth-table specification using Gröbner bases over Galois fields. Interpret this function $f : \mathbb{B}^3 \to \mathbb{B}^3$ as a function $f : \mathbb{F}_{2^3} \to \mathbb{F}_{2^3}$. Recall from the lecture slides on finite fields (also see Sec VI in my book chapter on finite fields), any function $f : \mathbb{F}_q \to \mathbb{F}_q$ is a polynomial function; i.e. there exists a polynomial $Z = \mathcal{F}(A)$ that describes this function.
- Using the Lagrange's interpolation formula (Eqn. One.5, Sec VI in my book chapter), derive a unique, minimal, canonical polynomial representation of the function as $Z = \mathcal{F}(A)$ over $\mathbb{F}_{2^3}$.
- Subsequently, using the **Weak Nullstellensatz over the finite field** $\mathbb{F}_{2^3}$, prove the equivalence of the polynomial function (specification) against the circuit that you synthesized (implementation). For this, you have to once again create a miter between the interpolated *Spec* and the synthesized implementation and solve it using Gröbner bases.
- Once again, describe **clearly and completely** your mathematical problem formulation and the algorithmic solution employed.

5) **Multivariate Division: (20 points)**

a) Implement the multi-variate division algorithm that performs the reduction $f \xrightarrow{f_1,\ldots,f_s}_+ r$, where the polynomials are from $\mathbb{F}[x_1, \ldots, x_n]$, where n the number of variables is fixed (given) and $\mathbb{F}$ is any field. You might wish to implement the algorithm as a function/subroutine/procedure in Singular.

b) Let $f = x^3 - x^2y - x^2z + x,\ f_1 = x^2y - z,\ f_2 = xy - 1$. Impose a deglex order, $x > y > z$ on

$\mathbb{Q}[x, y, z]$. Using your algorithm, compute r_1 = remainder of division by (f_1, f_2), and r_2 = remainder of division by (f_2, f_1). What do you notice?

6) **The Gröbner Basis Algorithm: (40 points)**

a) Now you are asked to implement the Buchberger's algorithm to compute a Gröbner basis G for a given ideal $J = \langle f_1, \ldots, f_s \rangle$.

b) Your algorithm should take a set of polynomials as input ($F = \{f_1, \ldots, f_s\}$) and produce the Gröbner basis ($G = \{g_1, \ldots, g_t\}$) as output.

c) You may use the multi-variate division (reduction) subroutine that you implemented in the above assignment to perform division. Otherwise, you may use the `reduce()` command of Singular for the same. Keep in mind that the reduce command requires that the divisor polynomials be given as an 'ideal'. Also, feel free to use the "teachstd.lib" which implements the `spoly()` function. Also feel free to use any data-structures of Singular – list, array, ideal, etc.

d) Let $f_1 = x^2y - y + x$, $f_2 = xy^2 - x$. Determine a Gröbner basis G of $I = \langle f_1, f_2 \rangle \subset \mathbb{Q}[x, y]$ using your program. First use the lex order with $x > y$, and then repeat the experiment using deglex and degrevlex orders. Compare your result with Singular's `groebner()` command.

e) Using your programs, determine whether or not $f \in \langle f_1, f_2 \rangle$, where $f = x^4y - 2x^5 + 2x^2y^2 - 2x^3y - 2x^4 - 2y^3 + 4xy^2 - 3x^2y + 2x^3 - y + 2x$; where f_1, f_2 are as given above. Use any monomial order.

f) Please write your code modularly, making use of procedural constructs of Singular. Preserve your code, don't lose it.