EFFICIENT BOOLEAN SATISFIABILITY USING A MODIFIED NOISE-BASED LOGIC APPROACH

J. KEITH KENEMER

Abstract. A novel method for SAT-solving is presented which uses sets of orthogonal and non-orthogonal binary sequences to encode Boolean variables. SAT statements are encoded into arithmetic operations on sequences–each logic clause is encoded into waveforms which are multiplied (AND-ed) together to encode a the logic statement in conjunctive normal form (CNF). The binary sequences are designed to detect when logic conflict conditions occur for each variable and force the output to zero for each term where this occurs by utilizing orthogonality. Satisfiability can be determined based on whether the output sequence is non-zero (SAT) or zero (UNSAT) over all samples. Encoding the logic statement into a discrete waveform eliminates the need for an explicit expansion of the statement while still producing a discernable result. The SAT solutions can be found in polynomial time and space complexity using this method.

1. Introduction

The proposed method was inspired by waveform-based logic approaches including Noise-based Logic (NBL), Sinusoidal Logic, and also quantum computing. The new method has an important difference, which is that no attempt is made to encode all possible logic minterms into unique basis functions. In NBL-SAT, unique basis functions can be constructed for each minterm, but this requires long correlation times when solving a high-dimensional problem. Sinusoidal Logic is limited by the number of basis functions available and so suffers from basis degeneracy. The proposed method does not suffer from these limitations since the orthogonal binary sequences are used only to detect logic conflicts and force the corresponding output to zero in these cases, and so the constraints on the associated sequences are manageable

2. Boolean Satisfiability(SAT)

Given a logic statement in conjunctive normal form (CNF), the objective of SAT to to determine if there is an assignment of the boolean variables which satisifies the statement(i.e. makes it evaulate to TRUE). SAT is an NP-complete problem with no known polynomial time solution.
Consider the following statement:

$$y = (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge x_2 \tag{1}$$

which is not satisfiable. If we expand the expression, we notice each term in the expansion contains a logical conflict–variables AND-ed with their negation. If all terms in the expansion have such conflicts, the overall statement will not be satisfiable.

$$y = (x_1 \wedge \overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2} \wedge x_2) \vee (x_1 \wedge x_2 \wedge \overline{x_2}) \vee (\overline{x_2} \wedge \overline{x_2} \wedge x_2) \qquad (2)$$

If we replace x_2 in the last clause with x_3, the statement then becomes satisfiable, as new terms are created in the expansion which are free of logical conflict.

3. Encoding of Variables

Each boolean variable x_i and its negation $\overline{x_i}$ are encoded using discrete pseudo-random binary sequences $X_i(n)$ and $\overline{X_i}(n)$ of length L samples which obey the following rules:

(1) $X_i(n) \epsilon \{0,1\}, p = \frac{1}{2}$ at each sample n (random binary sequence)
(2) $\overline{X_i}(n) = 1 - X_i(n)$ for each sample n (point-wise orthogonal)

This encoding effectively forces any term of the form $X_i(n) \cdot \overline{X_i}(n)$ to be zero, regardless of whatever else it is multiplied with. Terms of the form $X_i(n) \cdot X_j(n), i \neq j$ and $X_i(n) \cdot \overline{X_j}(n), i \neq j$ provide non-zero contribution to any product term they are contained within. As seen in later sections, this encoding will allow us to detect when all terms in the expansion have a conflict since this will result in an output sequence consisting entirely of zeros. Encoding the logic expression into a waveform eliminates the need for explicit expansion of the statement but still produces a discernable result.

4. Encoding of Logic Statements

Propositional logic sentences will be transformed into arithmetic expressions involving the discrete sequences described in the previous section. The following table describes the encoding:

Logic Variable/Operator	Arithmetic Variable/Operator
x_i	$X_i(n)$
$\overline{x}_i$	$\overline{X}_i(n)$
$\vee$ (OR)	$+$ (addition)
$\wedge$ (AND)	$\cdot$ (multiplication)

Table 1. Encoding of Logic Statements

This encoding of variables and operators will produce all 0's over the entire output sequence for UNSAT statements. The output for SAT statements will result in at least 1 non-zero value in the output sequence.

5. Summary

TO DO