ISSN 0105-8517

An Algebraic Model for Bounding Threshold Circuit Depth

Joan Boyar
Gudmund Frandsen
Carl Sturtivant

DAIMI PB – 239
March 1988

Abstract

We define a new structured and general model of computation: circuits using arbitrary fan-in arithmetic gates over the characteristic-two finite fields ($\mathbf{F}_{2^n}$). These circuits have only one input and one output. We show how they correspond naturally to boolean computations with n inputs and n outputs.

We show that if circuit sizes are polynomially related then the arithmetic circuit depth and the threshold circuit depth to compute a given function differ by at most a constant factor.

We use threshold circuits that allow arbitrary integer weights; however, we show that when compared to the usual threshold model, the depth measure of this generalised model only differs by at most a constant factor (at polynomial size).

The fan-in of our arithmetic model is also unbounded in the most generous sense: circuit size is measured as the number of Σ and Π gates; there is no bound on the number of "wires".

We show that these results are provable for any "reasonable" correspondance between bit strings of n-bits and elements of $\mathbf{F}_{2^n}$. And, we find two distinct characterizations of "reasonable". Thus, we have shown that arbitrary fan-in arithmetic computations over $\mathbf{F}_{2^n}$ constitute a precise abstraction of boolean threshold computations with the pleasant property that various algebraic laws have been recovered.

Introduction

The development of arbitrary fan-in boolean circuit complexity within $NC^{(1)}$ has entailed the use of gates with fewer and fewer pleasant algebraic properties [Chandra 84, Furst 81, Razborov 87, Smolensky 87, Hajnal 87]. In particular, the use of threshold gates does not even provide the associative law, and it is not clear that fully arbitrary fan-in can be allowed (cf. parity, where repeated fan-in of the same value gives no additional computational power). Nevertheless, threshold circuits seem to provide the most powerful physically reasonable arbitrary fan-in model of parallel computation.

The aim of this paper is to provide a new model of parallel computation that gives essentially the same complexity measure as threshold circuits, whilst possessing as many pleasant algebraic properties and as few combinatorial restrictions as possible.

Let $\mathbf{F}_{2^n}$ be the finite field with 2^n elements, often implemented as polynomials of degree $< n$ where arithmetic is done modulo a fixed irreducible polynomial of degree n, and all coefficient arithmetic is in $\mathbf{F}_2$ ($\{0,1\}$ with **exclusive-or** and **and** as $+$ and $\times$). For details, see [Lidl 83].

We consider the computation of functions $f : \mathbf{F}_{2^n} \longrightarrow \mathbf{F}_{2^n}$ using arbitrary fan-in sum (Σ) and product (Π) gates and constants. All arithmetic is field operations with unbounded fan-in in the truest sense – to compute an arbitrary power of a value, that value may be fanned-in to a single Π-gate the requisite number of times. The depth of such a circuit is defined in the usual way, and the size by the number of gates. There is no bound on the number of "wires".

The usual implementation of $\mathbf{F}_{2^n}$ described above provides a natural correspondance between field elements and bit strings of n bits and thus between field computations and n input, n output boolean computations. As a boolean model of computation, we choose threshold circuits. Under the above correspondance, we show that arithmetic and threshold depth at polynomial size are essentially the same measure of complexity, differing by at most a constant factor.

We also characterise those implementations of $\mathbf{F}_{2^n}$ under which the above result holds. Thus we have shown that the $\mathbf{F}_{2^n}$ model of parallel computation described above is a precise abstraction of threshold circuits. This is particularly surprising on two counts: first, because it has been widely assumed that field computations could not model parallel computation reasonably owing to "degree difficulties", e.g. [von zur Gathen 86]; second, and more interestingly, because of the provable inability of simple modular operations to compute majority in constant depth and polynomial size [Razborov 87, Smolensky 87].

On the boolean front, several threshold models of computation are possible, the main distinction being between bounded and unbounded weights (see [Hajnal 87] or further on here for details). This corresponds simply to whether or not unbounded fan-in from a single source is allowed, as it is in the $\mathbf{F}_{2^n}$ model. We show that these distinct threshold models are all essentially equivalent.

The interest in threshold complexity, i.e. $\mathbf{F}_{2^n}$ complexity, is manifold. Threshold gates are used in several models of the brain [Rumelhart 86]. Threshold functions clearly provide the most powerful symmetric gates possible: for on the one hand, any symmetric function can be computed by a constant depth threshold circuit of polynomial size [Hajnal 87]; and on the other hand, if we allow circuits using arbitrary symmetric functions, but also allow arbitrary fan-in from a *single* source, then any function may be computed using *one* gate.

Threshold functions can be computed by a fan-in 2 boolean circuit of logarithmic depth (i.e. they are in $NC^{(1)}$). Thus, it is a matter of supreme interest whether all functions in $NC^{(1)}$ can be computed by constant depth, polynomially sized threshold circuits. In an excellent paper, [Barrington 86], showed that the word problem for any fixed non-solvable group is complete for $NC^{(1)}$ via appropriate reductions; whilst conversely, the word problem for any fixed solvable group of order g is reducible to mod-g operations via appropriate reductions. The latter operations are known not to be complete for $NC^{(1)}$ [Smolensky 87]. Since mod-g reduces to majority (via binary count, see [Chandra 84]), the word problem for solvable groups has constant depth polynomial size threshold circuits. Conversely, having majority gates can easily be seen to be equivalent, in this context, to allowing mod-p gates for all p. This follows from the use of integer Chinese remaindering modulo small primes to effect counting.

The problems of iterated two-by-two matrix multiplication over $\mathbf{F}_4$ and of iterated three-by-three matrix multiplication over $\mathbf{F}_2$ are thus easily seen to be complete for $NC^{(1)}$. First, they are in $NC^{(1)}$ since matrix multiplication is associative and therefore a tree of constant size, constant depth matrix multiplication circuits will solve such a problem. Second, the multiplicative group in each case is insoluble. In fact, in the case of two-by-two matrices over $\mathbf{F}_4$, the multiplicative group is isomorphic to A_5, which is the smallest insoluble group [Gorenstein 80]. In contrast, iterated two-by-two matrix multiplication over $\mathbf{F}_2$ involves a solvable group. Thus some sufficiently large constant size iterated matrix multiplication gate is the next natural arbitrary fan-in gate to consider after threshold gates when investigating the structure of $NC^{(1)}$. In particular, is there a constant depth polynomial size threshold circuit to simulate such a gate?

Intuitively, threshold circuits catch all abelian problems efficiently and the issue is whether this extends to non-abelian problems. When considered in the light of the earlier remark concerning symmetric functions, these iterated constant size matrix multiplication problems provide the maximum amount of algebraic properties compatible with being complete for $NC^{(1)}$, iff threshold is not complete for $NC^{(1)}$.

In the event that threshold proves to be not complete for $NC^{(1)}$, there is some justification for regarding the classes based upon arbitrary fan-in threshold gates to be the fundamental classes, rather than the classes based on fan-in 2 boolean gates or on arbitrary fan-in iterated matrix multiplication gates. It is reasonable to regard arbitrary fan-in threshold gates as efficiently physically realisable. Presumably, under these circumstances, the iterated matrix multiplication is not efficiently physically realisable. Furthermore, the classes based upon threshold circuits correspond to the simple arithmetic model in $\mathbf{F}_{2^n}$. If it were found that threshold is not complete for $NC^{(1)}$, it would be an unpleasant quirk in the relationship between arbitrary fan-in and fan-in 2 computation.

Conversely, should threshold circuits exist to solve the constant size iterated matrix multiplication problem in constant depth and polynomial size, then the depths of fan-in 2 boolean circuits and threshold or $\mathbf{F}_{2^n}$

circuits to compute a given function would be related by precisely a factor of $\log n$ (ignoring constants and with at most polynomial change in circuit size). Furthermore, it would imply the collapse of the hierarchy of constant depth polynomial size threshold circuits given in [Hajnal 87] to some particular depth. We leave verification of these details to the reader.

The idea of bringing into correspondance finite field arithmetic and boolean computations is not a new one. In [Sturtivant 87], we investigated arithmetic vs. boolean computation in all possible representations of all finite fields. This was for sequential complexity accomplished with fan-in 2 arithmetic, but a connecting result in [Frandsen 88a] shows that this holds equally well for arbitrary fan-in. It also thoroughly investigates the parallel complexity analogue of [Sturtivant 87], which is essentially the generalisation of the present paper to arbitrary characteristic finite fields with more general field representations. In [Bøgestrand 88], tight characteristic 2, fan-in 2, sequential complexity results with a fixed range of representations are described.

In the present paper, we confine our attention to the arbitrary fan-in $\mathbf{F}_{2^n}$ model where representations of field elements are unique and consist of bit-strings of n-bits. All of the results in this paper would hold in the case of computations with more than one input or output and other various or mild generalizations, which we ignore in the interests of simplicity.

Definitions And Results

Definition d1/1

Threshold Functions and Circuits

Threshold functions are of the form

$$T_k^{\underline{\alpha}}(y_1, \ldots, y_m) = 1 \text{ iff } \sum_{i=1}^{m} \alpha_i y_i \geq k$$

where $\underline{\alpha} = (\alpha_1, \ldots, \alpha_m)$ is an arbitrary tuple of integers, and $y_1, \ldots, y_m$ are boolean inputs.

Threshold circuits are defined as circuits in which the gates compute threshold functions of arbitrary fan-in (i.e. in the above definition, m is arbitrarily large). The size of such circuits is defined to be the number of gates. The depth is defined in the usual manner.

Remark: Note that this model of computation is essentially unchanged if we allow only positive integers but also allow negation gates [Hajnal 87], or if we allow $y_1, \ldots, y_m$ to be integers of an appropriately bounded size, since these integers may be represented in binary and simulated by a threshold function of the above type. Neither of these changes alters the measures of depth and size by more than a constant factor.

Definition d2/2

Majority/Negation Circuits

The majority function is of the form:

$$M^{2k}(y_1, \ldots, y_{2k}) = 1 \text{ iff the arithmetic sum } \sum_{i=1}^{2k} y_i \geq k$$

Majority/negation circuits are defined as circuits using unary negation gates and arbitrary fan-in majority gates with the restriction that any two inputs to a majority gate must be either outputs of different gates, distinct inputs, or an input and a gate output.

Remark: This model appears intuitively weaker than the previous model, but in fact, this is not the case, as the following lemmas establish.

Lemma l1/3

Let $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ be an arbitrary tuple of integers. Such a tuple will satisfy various inequalities of the form $\sum_{i \in I_1} \alpha_i \geq (>) \sum_{j \in I_2} \alpha_j$. We regard two tuples as equivalent when they satisfy the same set of inequalities. Any tuple is equivalent to a tuple whose entries have magnitude at most n^{n+1}.

Proof

Let $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ be an arbitrary tuple of integers. For every two (disjoint) subsets I_1, I_2 of $\{1, 2, \ldots, n\}$ precisely two of the four statements that follow are valid:

$$\begin{aligned}
&\textstyle\sum_{i \in I_1} \alpha_i - \sum_{i \in I_2} \alpha_i \geq 0 \\
&\textstyle\sum_{i \in I_2} \alpha_i - \sum_{i \in I_1} \alpha_i \geq 1 \\
&\textstyle\sum_{i \in I_1} \alpha_i - \sum_{i \in I_2} \alpha_i \geq 1 \\
&\textstyle\sum_{i \in I_2} \alpha_i - \sum_{i \in I_1} \alpha_i \geq 0
\end{aligned}$$

The resulting collection of inequalities may be put into matrix form: $\mathbf{A}\underline{\beta} \geq \mathbf{b}$, where $\underline{\beta} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^t$.

$\mathbf{A}$ is a $3^n \times n$ matrix with $-1, 0, 1$ entries (and rank n) and $\mathbf{b}$ is a 3^n-dimensional column vector with $0, 1$ entries.

Consequently the lemma is proved if we can modify an arbitrary integer solution $\mathbf{x} = \underline{\beta}$ to $\mathbf{Ax} \geq \mathbf{b}$ in a way that preserves the inequality, but diminishes the absolute values of $\underline{\beta}$'s entries sufficiently.

In the following, let m_k denote the maximal absolute value of a $k \times k$ determinant with $-1, 0, 1$ entries.

We use the following procedure to diminish absolute values of solution $\mathbf{x} = \underline{\beta}$ to $\mathbf{A}x \geq \mathbf{b}$.

<u>Input:</u>	$\mathbf{A}, \underline{\beta}, \mathbf{b}$ such that $\mathbf{A}\underline{\beta} \geq \mathbf{b}$.
	$\mathbf{A}$ has rank n, is $3^n \times n$, has $-1, 0, 1$ entries
	$\mathbf{b}$ is $3^n \times 1$, has $0, 1$ entries
	$\underline{\beta}$ is $n \times 1$, has integer entries

<u>Output</u> $\underline{\beta}'$ such that $\mathbf{A}\underline{\beta}' \geq \mathbf{b}$ and $|\beta'_i| \leq n^2 \cdot m_{n-1}^2$ for all i.

Method:

```
$\mathbf{x} := \underline{\beta}$;
while $(\mathbf{Ax})_i \geq 2$ for all $i$ such that the $i$'th row of $\mathbf{A}$ is non-
   singular do decrease the absolute value of some $x_j$ by 1;

for $k := 1$ to $n-1$ do
   | INVARIANT: $\mathbf{Ax} \geq \mathbf{b}$,
   | There exists a $k \times n$ submatrix $\mathbf{B'}$ of $\mathbf{A}$ such that $\mathbf{B'}$ has
   | rank $k$ and all entries in the vector $\mathbf{B'}\,\mathbf{x}$ lie in the interval
   | $[0, n \cdot m_{n-1}]$.
   Choose a $\mathbf{B'}$ that fulfils the invariant above; We want to find
   a nonzero vector $\mathbf{d}$ in the nullspace of $\mathbf{B'}$. We can assume
   without loss of generality that the $k$ leftmost columns of $\mathbf{B'}$
   are linearly independent and constitute a square matrix $\mathbf{B}$.
   Let $\mathbf{B}_i$ be $\mathbf{B}$ where the $i$'th column has been replaced by
   the last column of $\mathbf{B'}$, and form the vector

      $\mathbf{d} = (det\mathbf{B}_1, det\mathbf{B}_2, \ldots, det\mathbf{B}_k, 0, 0, \ldots, 0, -det\mathbf{B})^t$

   that fulfils $\mathbf{Bd} = 0$ by Cramer's rule.
   Since $\mathbf{A}$ has rank $n > k$, there exists a row vector in $\mathbf{A}$
   linearly independent from the rows in $\mathbf{B'}$ and such that
   $\mathbf{a} \cdot \mathbf{d} \neq 0$. Furthermore the sum of the elements in $\mathbf{d}$ are
   bunded by $(k+1) \cdot m_k \leq n \cdot m_{n-1}$. These facts assure that
   the loop beneath stops yielding a vector $\mathbf{x}$ that is still a
   solution to $A\ \mathbf{x} \geq \mathbf{b}$:

   while [for all row vectors $\mathbf{a}$ in $\mathbf{A}$ which are linearly indepen-
   dent of the rows in $\mathbf{B'}$ it is the case that $\mathbf{a} \cdot \mathbf{x} \geq n \cdot m_{n-1} + 1$]
   do
   $\mathbf{x} := \mathbf{x} \pm \mathbf{d}$ (where +/- is selected in order to decrease the
      minimal value of $\mathbf{a} \cdot \mathbf{x}$ as in the condition);
{end of for-loop}
```

$\underline{\beta}' := \mathbf{x}$;

<u>INVARIANT</u>: $\mathbf{A}\underline{\beta}' \geq \mathbf{b}$,
There exists an $n \times n$ submatrix $\mathbf{B}$ of $\mathbf{A}$ such that $\mathbf{B}$ is nonsingular and all entries in the vector $\mathbf{B}\underline{\beta}' = \mathbf{c}$ lie in the interval $[0, n \cdot m_{n-1}]$.

Choose $\mathbf{B}$, $\mathbf{c}$ as in the invariant. According to Cramer's rule $\underline{\beta}' = \frac{1}{det\mathbf{B}} \cdot (det\mathbf{B}_1, det\mathbf{B}_2, \ldots, det\mathbf{B}_n)$, where $\mathbf{B}_i$ is $\mathbf{B}$ with the i'th column replaced by $\mathbf{c}$. Consequently, $|\beta_i'| \leq |det\mathbf{B}_i| \leq m_{n-1} \sum_{i=1}^{n} c_i \leq m_{n-1}\, n \cdot n \cdot m_{n-1} = n^2 \cdot m_{n-1}^2$.

{end of algorithm}

To prove the lemma, we need only verify that $m_{n-1}^2 \leq n^{n-1}$. A $k \times k$ determinant with $-1, 0, 1$ entries can be regarded as the volume of a hyperparallelipiped each of whose edges are bounded in length by $\sqrt{k}$, implying that $m_k \leq \sqrt{k}^k$.

□

Remark: The reader will note that this is a slight modification of the technique used to prove integer linear programming to be in NP [Hopcroft 79].

Lemma 12/4

A threshold gate computing $T_k^{\underline{\alpha}}(y_1, \ldots, y_n)$ can be simulated by a majority/negation circuit of constant depth and size polynomial in n, irrespective of the size of the integers $\underline{\alpha}$, k.

Proof

Given $\underline{\alpha}$,k, we know from 11/3 that there exists an equivalent tuple $\underline{\beta}$ each of whose integers require at most $(n+1)\log n$ bits to describe (ignoring the sign). Since $\underline{\beta}$ satisfies the same subset inequalities as $\underline{\alpha}$, it is clear that there exists a new threshold $k' \leq \sum_{i=1}^{n} |\beta_i|$ such that:

$$T_k^{\underline{\alpha}}(y_1, \ldots, y_n) = T_{k'}^{\underline{\beta}}(y_1, \ldots, y_n)$$

Thus k' requires at most $(n+2)\log n$ bits to describe. We then have

$$T_{k'}^{\underline{\beta}}(y_1, \ldots, y_n) = 1 \text{ iff } \sum_{i=1}^{n} \beta_i y_i \geq k'$$

Thus $T_{k'}^{\underline{\beta}}$ may be computed by evaluating the sum and making the comparison explicitly. That all of these operations can be computed in constant depth and polynomial size using majority and negation gates follows immediately from the reductions in [Chandra 84]. In fact, the negative weights may be eliminated first by negating the corresponding y_i and adjusting the threshold value appropriately, as noted in [Hajnal 87].

$\square$

Remark: In fact, the above lemma may be extended to the case where the weights and threshold are arbitrary real numbers, see [Frandsen 88b].

Theorem th1/5

If there is a threshold circuit of size C_T and depth d_T to compute a given n-input function, then there is a majority/negation circuit of size polynomial in C_T and n and depth at most $c \cdot d_T$ where c is a constant.

A similar (converse) statement holds for threshold circuits simulating majority/negation circuits. Thus, the depth measure of complexity at bounded size is essentially the same for both models.

Proof

The second part follows easily, since $M^{2k}(y_1, \ldots, y_{2k}) = T_k^{\underline{\alpha}}(y_1, \ldots, y_{2k})$ when $\underline{\alpha} = (1, 1, 1, \ldots, 1)$.

The first part follows from Lemma l 2/4 and the observation that the maximum fan-in of a threshold gates is about $C_T + n$.

□

Definition d3/6

Trigger Circuits

We regard the threshold circuits from definition d1/1 and remarks and the majority/negation circuits from definition d2/2 (and any other simple variations) as a single class of models of computation. These are equivalent up to constant depth and polynomial size. We refer to this class of models and any representative member of it as *trigger circuits.*

Definition d4/7

An $\mathbf{F}_{2^n}$ Σ, Π Circuit

This is an arithmetic circuit in the field $\mathbf{F}_{2^n}$ using unbounded fan-in sum (Σ) and product (Π) gates. The size is defined to be the number of gates and the depth is defined in the usual manner.

Remark: Note that the fan-in to a Π-gate may be enormous since to compute a very high power of some value it is only necessary to use a single Π-gate.

Definition d5/8

A representation of $\mathbf{F}_{2^n}$

This is a bijection $\phi_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$. This defines which element is associated with which bit string.

Definition d6/9

A good representation of $\mathbf{F}_{2^n}$

This is a family of bijections $\phi_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$ and $n+1$ families of trigger circuits $(1 \leq j < n)$

$$\begin{aligned} S_n &: \{0,1\}^{n^2} \rightarrow \{0,1\}^n \\ P_n &: \{0,1\}^{n^2} \rightarrow \{0,1\}^n \\ C_n^j &: \{0,1\}^n \rightarrow \{0,1\}^n \end{aligned}$$

all of polynomially bounded size in n and constant depth, satisfying

$$\begin{aligned} S_n(\phi_n(Z_1), \ldots, \phi_n(Z_n)) &= \phi_n(\sum_{i=1}^{n} Z_i) \\ P_n(\phi_n(Z_1), \ldots, \phi_n(Z_n)) &= \phi_n(\prod_{i=1}^{n} Z_i) \\ C_n^j(\phi_n(Z_1)) &= \phi_n(Z_1^{2^j}) \end{aligned}$$

for all assignments of $\mathbf{F}_{2^n}$ elements to $Z_1, \ldots, Z_n$. Thus, ϕ_n defines the semantics of the representation, i.e. which bit string represents which field element (up to automorphism), and S_n, P_n and C_n^j provide constant-depth, polynomial-size implementations of n-input field Σ and Π gates, and gates computing the j^{th} conjugate, respectively.

Remarks: It is not clear either that such a representation exists or that such a representation is able to simulate an $\mathbf{F}_{2^n}$ Σ, Π circuit with only a constant change in depth and a polynomial increase in size. Conjugation is simply an automorphism of $\mathbf{F}_{2^n}$ (see [Lidl 83]), and is a necessary operation for efficient simulation of arbitrary fan-in $\mathbf{F}_{2^n}$ circuits as will be shown later.

Lemma l3/10

Let $\psi_n : \mathbf{F}_{2^n} \rightarrow \mathbf{F}_{2^n}$ be an $\mathbf{F}_{2^n}$ Σ, Π circuit of depth $d_{(n)}$ and size $C_{(n)}$. Then there exists a corresponding trigger circuit $\tau_n : \{0,1\}^n \rightarrow \{0,1\}^n$ of depth $O(d_{(n)})$ and size $(n \cdot C_{(n)})^{O(1)}$, with every good representation $\phi_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$ such that $\tau_n(\phi_n(z)) = \phi_n(\psi_n(z))$.

Proof

We simulate the Σ and Π gates in the $\mathbf{F}_{2^n}$ circuit in this representation in such a way as to fulfil the statement of the lemma.

A basic difficulty is the unbounded fan-in of the gates. This is dealt with easily in the case of Σ gates, where multiple fan-in of a single value only contributes that value once or not at all to the output of the gate. Therefore the effective fan-in to a Σ-gate is at most about $n + C_{(n)}$.

In the case of a Π-gate, a fan-in of a single value u, k times contributes a factor of u^k to the gate's output. Here k may be bounded by 2^n since any value u satisfies $u^{2^n} = u$ within $\mathbf{F}_{2^n}$ [Lidl 83]. Next, we observe that the conjugates $u, u^2, u^4, \ldots, u^{2^{n-1}}$ may be used to compute any power of u using a Π-gate of fan-in at most n. Since there are at most about $n + C_{(n)}$ *distinct* inputs to any Π-gate, if we introduce a new kind of gate that computes the conjugates, we ensure that the fan-in to any Π-gate is at most about $n(n + C_{(n)})$, whilst at most doubling the depth of the original circuit.

We now assume that the original $\mathbf{F}_{2^n}$ Σ, Π circuit is transformed in the above ways in order to control gate fan-in. The three kinds of gates Σ, Π and conjugation are implemented in the good representation in constant depth and size polynomial in the fan-in by using the circuits defined to exist in d6/9. The resulting circuit fulfils the statement of the lemma.

$\square$

Definition d7/11

A standard representation of $\mathbf{F}_{2^n}$

Let $f_n \in \mathbf{F}_2[\mathbf{x}]$ be an irreducible polynomial of degree n. Then $\mathbf{F}_2[\mathbf{x}]/(f_n)$ is isomorphic to $\mathbf{F}_{2^n}$ [Lidl 83]. Thus, $\mathbf{F}_{2^n}$ can be represented as all polynomials over $\mathbf{F}_2$ of degree less than n, with arithmetic being simulated by polynomial arithmetic modulo f_n.

Choosing $\phi_n(z)$ to be the coefficient tuple of the polynomial representing the field element z defines the semantic function of a standard representation of $\mathbf{F}_{2^n}$.

Remark: It is not clear at this juncture whether this is a *good* representation.

Definition d8/12

A strong representation of $\mathbf{F}_{2^n}$

Here we regard $\{0,1\}$ both as boolean values and as members of $\mathbf{F}_{2^n}$. A strong representation is a semantic bijection $\phi_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$ with the property that there are constant depth, polynomially bounded size (in n) arithmetic circuits

$$\begin{aligned} i_n &: \{0,1\}^n \rightarrow \mathbf{F}_{2^n} \\ o_n &: \mathbf{F}_{2^n} \rightarrow \{0,1\}^n \end{aligned}$$

satisfying

$$\begin{aligned} o_n(z) &= \phi_n(z) \\ i_n(\phi_n(z)) &= z \end{aligned}$$

for all field elements z.

Remark: Intuitively, a strong representation is one in which entry to or exit from the representation can be accomplished very efficiently with arithmetic.

Definition d9/13

Equivalent Representations

A representation $\phi_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$ *translates* into a representation $\theta_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$ (written $\phi_n \leq \theta_n$) iff there exists a constant depth, polynomially bounded size (in n) trigger circuit $T_n : \{0,1\}^n \to \{0,1\}^n$ satisfying $T_n(\phi_n(z)) = \theta_n(z)$ for all $z \in \mathbf{F}_{2^n}$. We say that ϕ_n and θ_n are *equivalent* representations (written $\phi_n \equiv \theta_n$) iff $\phi_n \leq \theta_n$ and $\theta_n \leq \phi_n$.

Lemma 14/14

If ϕ_n is a strong representation of $\mathbf{F}_{2^n}$ and θ_n is a good representation of $\mathbf{F}_{2^n}$, then $\phi_n \equiv \theta_n$. Thus both ϕ_n and θ_n are good and strong.

Proof

$\underline{\phi_n \leq \theta_n}$
If ϕ_n is strong then a constant depth polynomial size arithmetic circuit, i_n, exists for ϕ_n. If i_n is implemented using representation θ_n, this gives a trigger circuit that takes the θ_n representation of the ϕ_n bit string representing a field element, and produces the θ_n representation of that element. This circuit is constant depth, polynomial size because the representation θ_n is good. All that remains is to choose the two bit strings $a_n = \theta_n(0)$ and $b_n = \theta_n(1)$, and to prefix each input to the new circuit, where an θ_n representation of zero or one is required, with a small circuit that takes a boolean zero or one and switches into the old input either a_n or b_n as appropriate.

$\underline{\theta_n \leq \phi_n}$
ϕ_n is strong; therefore a polynomial, constant depth circuit o_n exists for ϕ_n^{-1}. If o_n is implemented in representation θ_n, the input is now the θ_n representation of a field element, and the output is the θ_n representation of the bit-string that is the ϕ_n representation of the same field element. Now take a small circuit that will recognise the θ_n representations of zero and one and output the corresponding boolean value. Appending one of these to each output where a zero or one in representation θ_n will appear achieves the desired result.

$\square$

Lemma 15/15

Any standard representation is strong.

Proof

Let $u = \Sigma_{i=0}^{n-1} u_i \theta^i$ be the relationship between a field element u and its bits in a standard representation $(u_o, \ldots, u_{n-1})$. Then $u^{2^j} = \Sigma_{i=0}^{n-1} u_i (\theta^{2^j})^i$ since $(x+y)^2 = x^2 + y^2$ in $\mathbf{F}_{2^n}$. Thus

$$\begin{bmatrix} u \\ u^2 \\ u^4 \\ \vdots \\ u^{2^{n-1}} \end{bmatrix} = \begin{bmatrix} 1 & \theta & \theta^2 & \theta^3 & \ldots \\ 1 & \theta^2 & \theta^4 & \theta^6 & \ldots \\ 1 & \theta^4 & \theta^8 & \theta^{12} & \ldots \\ \vdots & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{bmatrix}$$

As observed in [Sturtivant 87], the constant matrix is invertable since the conjugates $\theta, \theta^2, \theta^4, \ldots, \theta^{2^{k-1}}$ are distinct. Thus $(u_0, \ldots, u_{n-1})$ are computable from u in arithmetic depth 2 and polynomial size. (Or, we can merely observe that traces can be used to compute u_i directly [Lidl 83].)

Conversely, $u = \Sigma_{i=0}^{n-1} u_i \theta^i$ immediately yields a depth 2 arithmetic circuit of polynomial size to compute u from its bits.

$\square$

Remark: Note that the existence of a good representation implies that the standard representation is good by 14/14 and 15/15. We have not, however, shown the existence of a good representation. Clearly, a good representation exists iff a standard representation is good.

Lemma 16/16

There exists a constant depth polynomially bounded size in n majority/negation circuit to compute the n-input parity function (modulo-2 sum).

Proof

See e.g. [Hajnal 87].

□

Remark: Note that it is just as simple to compute the so-called mod-p function defined in [Smolensky 87]. Equally n-input **and** or **or** gates can be very easily simulated.

Lemma 17/17

An N-input majority gate (for $N < 2^n$) can be simulated by a constant-depth, $N^{O(1)}$ size $\mathbf{F}_{2^n}$ Σ, Π circuit, regarding $\{0,1\}$ as field elements.

Proof

Let $x_1, \ldots, x_N$ be the N-inputs that will be $\{0,1\}$ field values and whose majority is to be computed. Each input, x_i, enters a small circuit that on input 0, outputs 1; and on input 1, outputs g, where g is a fixed primitive element [Lidl 83] in the field $\mathbf{F}_{2^n}$. All of these computations occur in depth 2, and all of their results are the input to a single Π-gate.

Thus, the function so far computed is given by the expression:

$$\prod_{i=1}^{N} g^{x_i} = g^{\sum_{i=1}^{N} x_i}$$

Since g is a generator for the multiplicative group of $\mathbf{F}_{2^n}$, the $N+1$ possible powers of g that can arise from this expression are all distinct. As the majority function only depends upon the number of ones in the input, the majority function may be computed from this result by table lookup. This is achieved in $\mathbf{F}_{2^n}$ by the fact that $b(1-(x-a)^{2^n-1})$ computes b when x is a and zero otherwise [Lidl 83].

□

Theorem th2/18

Let $\tau_n : \{0,1\}^n \to \{0,1\}^n$ be a trigger circuit of depth $d_{(n)}$and size $C_{(n)}$. Then there exists a corresponding $\mathbf{F}_{2^n}$ Σ, Π circuit $\psi_n : \mathbf{F}_{2^n} \to \mathbf{F}_{2^n}$ of depth $O(d_{(n)})$ and size $(n \cdot C_{(n)})^{O(1)}$, with every strong representation $\phi_n : \mathbf{F}_{2^n} \to \{0,1\}^n$ such that $\tau_n(\phi_n(z)) = \phi_n(\psi_n(z))$. In particular, this follows if ϕ_n is a standard representation.

Proof

This follows from Theorem th1/5 and Lemma l7/17. The last part follows from Lemma l5/15.

$\square$

Lemma l8/19

There exist constant-depth polynomial size majority/negation circuits for:

(a) the addition of n n-bit numbers;

(b) the addition of n degree-n polynomials over $\mathbf{F}_2$;

(c) the multiplication of 2 degree n polynomials over $\mathbf{F}_2$, given that one of them is fixed;

(d) the polynomial modulo operation with a fixed modulus;

(e) the integer modulo operation with a fixed modulus.

Proof

(a) [Chandra 84].

(b) Follows from Lemma l6/16.

(c) By long multiplication, this is just repeated addition, and thus this follows from (b).

(d) Also follows from (b).

(e) Follows from [Chandra 84].

Lemma l9/20

The number of irreducible polynomials over $\mathbf{F}_2[x]$ of degree $\leq \delta$ is at least $2^\delta/\delta$.

Proof

Consider the field $\mathbf{F}_{2^\delta}$. An element of this field is a root of an irreducible polynomial of degree less than or equal to δ, and its conjugates are the other roots of this polynomial [Lidl 83]. Since an element may have at most δ distinct conjugates and the field has 2^δ elements, the result follows.

$\square$

Lemma l10/21

(Chinese Remaindering for polynomials in $\mathbf{Z}_2[x]$)
Let $p_1, \ldots, p_k$ be irreducible polynomials in $\mathbf{F}_2[x]$ and let $f_1, \ldots, f_k$ be any polynomials in $\mathbf{F}_2[x]$ satisfying $deg\ f_i < deg\ p_i$. Then there exists a unique polynomial f satisfying the conditions $deg\ f < \sum_i deg\ p_i$ and $f \equiv f_i \ mod\ p_i$ for each i. Furthermore, $f \equiv \sum_{i=1}^{k} b_i f_i \ mod\ \prod_{i=1}^{k} p_i$ where the polynomials b_i only depend upon the polynomials $p_1, \ldots, p_k$ and satisfy the constraint $deg\ b_i < \sum_i deg\ p_i$.

Proof

The uniqueness follows from [Hungerford 74, p. 131]. It is easily verified that choosing $b_i = c_i d_i$ where $c_i = \prod_{j \neq i} p_j$ and $d_i = c_i^{-1}$ in the field $\mathbf{Z}_2[x]/(p_i)$ yields the correct solution. This is entirely analogous to the integer Chinese Remaindering Theorem.

□

Lemma l11/22

There exist constant depth polynomial size majority/negation circuits for the multiplication of n degree n polynomials over $\mathbf{F}_2$.

Proof

In this proof, we use the phrase "table lookup" to mean computing a function directly from its disjunctive normal form in constant depth.

The following is analogous to [Beame 86] where an iterated product of integers is considered. To compute the product of polynomials, $\prod_{i=1}^{n} b^{(i)}$, we take the first k irreducible polynomials $p^{(1)}, \ldots, p^{(k)}$ from $\mathbf{F}_2[x]$ and compute in parallel all the residues $b^{(i)}$ *mod* $p^{(j)}$ using Lemma 8/19 part (d).

We then compute in parallel $B^{(j)} = (\prod_{i=1}^{n} (b^{(i)} \ mod \ p^{(j)}) \ mod \ p^{(j)})$ in $\mathbf{F}_2[x]/(p^{(j)})$, which is a *field* since $p^{(j)}$ is irreducible.

$B^{(j)}$ is computed by parallel table lookup of the discrete logarithms of $b^{(i)}$ *mod* $p^{(j)}$ in the fields $\mathbf{F}_2[x]/(p^{(j)}) = \mathbf{F}_{2^{\alpha_j}}$. Let $l_i^{(j)} = \log_j(b^{(i)} \ mod \ p^{(j)})$. Then $\log_j B^{(j)} = \sum_{i=1}^{n} l_i^{(j)} \ mod \ (2^{\alpha_j} - 1)$, where integer addition is employed.

This sum is computed using Lemma l8/19 part (a) and the modulo operation using Lemma l8/19 part (e). (Some optimization is possible here, since the modulus simply gives rise to cyclic carrying, but we ignore this since it merely gains a constant factor in depth.) $B^{(j)}$ is recovered from its logarithm using table lookup.

The final step in the computation is to recover the product $B = \prod_{i=1}^{n} b^{(i)}$ from the $B^{(j)}$ values where $B^{(j)} = B \ mod \ p^{(j)}$. This is accomplished in constant depth using Lemma l10/21, and the circuits to implement Lemma l10/21 are given in Lemma l8/19 parts (c), (b) and (d).

It remains to show that the table lookups are polynomial size circuits. This follows from Lemma l8/19, which ensures that the maximum degree of the $p^{(j)}$ polynomials is exponentially small.

□

Theorem th3/23

Any standard representation is a good representation.

Proof

Let $(b_0, \ldots, b_n)$ be the bit string representing the field element b in some standard representation of $\mathbf{F}_{2^n}$ given by the field identity $b = \sum_{i=0}^{n-1} b_i\theta^i$. Here a fixed element θ has been chosen for each field, with $(1, \theta, \theta^2, \ldots, \theta^{n-1})$ a basis over $\mathbf{F}_2$ and $f(\theta) = 0$ where $f(x) = \sum_{i=0}^{n} a_i x^i$ is irreducible. Arithmetic on a field element b is then simulated by polynomial arithmetic over $\mathbf{F}_2$ on $\sum_{i=0}^{n-1} b_i x^i$ modulo f [Lidl 83].

A Σ-gate is simulated in a standard representation simply by using Lemma l8/19 part (b).

A Π-gate is simulated in a standard representation (following the remark in the opening paragraph) in two stages: First, by simulating multiple polynomial multiplication using Lemma l11/22; and second by simulating a modulo $f(x)$ operation on the result using Lemma l8/19 part (d).

A conjugation gate is simulated in a standard representation as follows. If $b = \sum_{i=0}^{n-1} b_i\theta^i$, then $b^{2^\alpha} = \sum_{i=0}^{n-1} b_i(\theta^{2^\alpha})^i$. But $(\theta^{2^\alpha})^i$ has some bit representation

$$(\theta^{2^\alpha})^i = \sum_{j=0}^{n-1} M_{ji}^{(\alpha)}\theta^j$$

where $M_{ji}^{(\alpha)}$ is an element of a bit matrix. Thus

$$b^{2^\alpha} = \sum_{j=0}^{n-1}(\sum_{i=0}^{n-1} M_{ji}^{(\alpha)}b_i)\theta^j$$

Therefore the representation of b^{2^α} is a bit vector *linear* in the bits representing b, and can be computed in parallel using a number of parity computations. These are implemented as in Lemma 16/16.

$\square$

Theorem th4/24

All good representations of $\mathbf{F}_{2^n}$ are strong and vice-versa, and are equivalent to a standard representation.

Proof

Follows from th3/23 and lemmas 14/14 and 15/15.

$\square$

Remark: This shows that the only representations of interest are equivalent to a standard representation. Thus we may refer to $\mathbf{F}_{2^n}$ computations without reference to any particular representation, the assumption being that a good, strong representation is being used to define a correspondance between field elements and bit strings. In this sense, $\mathbf{F}_{2^n}$ arithmetic computations are an abstraction of bit-computations.

Consider a discrete logarithm in $\mathbf{F}_{2^n}$ being a representation $l_n : \mathbf{F}_{2^n} \rightarrow \{0,1\}^n$ (the log of 0 being defined so as to make l_n a bijection). By slackening the notions of good and strong to merely involve size constraints, it is possible using an analogue of th4/24 to show that the complexity of computing the discrete logarithm is essentially the same as the complexity of field addition in the logarithmic representation. This follows since field multiplication and conjugation are easy in the logarithmic representation.

Theorem th5/25

Regard the elements of $\mathbf{F}_{2^n}$ as bit strings (by virtue of some strong/good representation). Then the $\mathbf{F}_{2^n}$ Σ, Π depth and the trigger circuit depth of some n-input, n-output boolean function are related by at most a constant factor with polynomially bounded difference in circuit size.

Proof

Follows immediately from all the preceding theorems.

$\square$

Remark: This is an extremely robust statement as it is invariant under a wide range of changes of representation and implementation as outlined in the preceding definitions.

Essentially, both parallel time upper and lower bounds coincide in the two models. Thus this theorem provides the large body of theory which is known about finite fields and which can now be applied to the theory of parallel computation.

Conclusion

We have shown that:

- Threshold circuits of unbounded integer weights can be simulated in constant depth and polynomial size by majority and negation.
- There is a natural family of boolean representations of $\mathbf{F}_{2^n}$. These representations form a single computationally interesting equivalence class. Thus, one may regard $\mathbf{F}_{2^n}$ as an abstraction of the booleans.
- Unbounded fan-in $\mathbf{F}_{2^n}$ computations can be simulated with only constant increase in depth and polynomial change in size by majority and negation.
- An n input, n output majority/negation circuit can be simulated by a one-input, one-output $\mathbf{F}_{2^n}$ circuit with only constant increase in depth and polynomial change in size.
- The threshold, majority/negation, and $\mathbf{F}_{2^n}$ models of unbounded fan-in computation yield essentially equivalent bounds for parallel computation.

These findings suggest that:

- Trigger circuits form a robust class of computational models.
- Existential (and thus non-uniform) statements about boolean circuits may occasionally be easier to make than constructive ones, e.g. the use of very large integer constants in the unbounded weighted threshold model.
- The pursuit of parallel-lower bound technology should use the arithmetic model $\mathbf{F}_{2^n}$ since there already exists a wide body of known mathematics concerning it and provides the most structure, elegance, and parsimony as a model of computation.

References

[Barrington 86] BARRINGTON, D.A., Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1. *Proceedings 18th ACM Symp. on Theory of Computing, pp. 1-5, ACM, New York 1986.*

[Beame 86] BEAME, P.W., COOK, S.A., and HOOVER, H.J., Log Depth Circuits for Division and Related Problems. *SIAM J. Comput.* **15** *(1986), pp. 994-1003.*

[Bøgestrand 88] BØGESTRAND, K. and LUND, C., Computations in Finite Fields. *Technical Report DAIMI IR-71, Computer Science Department, Aarhus University, Denmark, 1988.*

[Chandra 84] CHANDRA, A.K., STOCKMEYER, L. and VISHKIN, U., Constant Depth Reducibility. *SIAM Journal on Computing 13* (1984), pp. 423-439.

[Frandsen 88a] FRANDSEN, G.S. and STURTIVANT, C., The Depth Efficacy of Unbounded Characteristic Finite Field Arithmetic. *Technical Report DAIMI PB-240, Computer Science Department, Aarhus University, Denmark, 1988.*

[Frandsen 88b] FRANDSEN, G.S. and STURTIVANT, C.S., An exact and efficient implementation of Threshold gates with arbitrary real weights. *Technical Report DAIMI PB-241, Computer Science Department, Aarhus University, Denmark, 1988.*

[Furst 81] FURST, M., SAXE, J.B. and SIPSER, M., Parity, Circuits and the Polynomial Time Hierarchy. *Proc. 22nd IEEE Symp. on Foundations of Computer Science, pp. 260-270. IEEE Computer Society, Los Angeles, 1981.*

[Gorenstein 80] GORENSTEIN, D., *Finite Groups.* Chelsea Publ. Comp., New York, 1980.

[Hajnal 87] HAJNAL, A., MAASS, W., PUDLÁK, P., SZEGEDY, M. and TURÁN, G., Threshold circuits of bounded depth. *Proceedings 28th IEEE Symp. on Foundations of Computer Science*, pp. 99-110. IEEE Computer Society, Los Angeles, 1987.

[Hopcroft 79] HOPCROFT, J.E. and ULLMAN, J.D. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[Hungerford 74] HUNGERFORD, T.W., *Algebra*, Springer Verlag, 1974.

[Lidl 83] LIDL, R. and NIEDERREITER, H., *Finite Fields.* Encyclopedia of Mathematics and its Applications, **20**. Addison-Wesley, Reading, Mass., 1983.

[Razborov 87] RAZBOROV, A.A., Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR, 41* (1987) pp. 333-338.

[Rumelhart 86] RUMELHART, D.E., McCLELLAND, J.L. and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1*, MIT Press, 1986.

[Smolensky 87] SMOLENSKY, R. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. *Proc. 19th ACM STOC, (1987), pp. 77-82.*

[Sturtivant 87] STURTIVANT, C. and FRANDSEN, G., The Computational Efficacy of Finite Field Arithmetic. *Technical Report DAIMI PB-227, Computer Science Department, Aarhus University, Denmark, (1987).*

[von zur Gathen 86] von zur GATHEN, J. and SEROUSSI, G., Boolean circuits versus arithmetic circuits. *Proceedings 6th Int. Conf. in Computer Science, Santiago, Chile, 1986, pp. 171-184.*