Canonical Set Theory with Applications from Parallel Matrix Operations and Data Structures to Homomorphic Encryption

Juan Ramírez

July 7, 2023

Jalisco, México
www.binaryprojx.com
jramirez@binaryprojx.com

Abstract

Few questions have resounded through the mind of the mathematician, as much as the simple question of describing the nature of a number. The longstanding consensus is that it does not matter which set theory is used to describe numbers. What matters is that it can be done. It is widely believed that the particular choice of a construction for natural and real numbers is irrelevant for the rest of mathematics. Here, a set theory is proposed as a canonical theory that yields transparent proofs in fundamental areas of mathematics including group theory, discrete mathematics, analysis, data types, and new results tying these areas and addressing Hilbert's 24-th (twenty-fourth) Problem and Benacerraf's Identification Problem. Applications to computer science include a model for Homomorphic Encryption, and a linearly-scaleable circuit that serves for parallel addition and multiplication of scalars, vectors and matrices, with wide implications for Area-Specific Integrated Circuits (ASICs) used in CGI, Neural Networks and AI training, Dimensionality Reduction for Machine Learning at Software and Hardware Level, Digital Signal Processing, among other applications that depend on fast and low-powered vector operations. This low-power In-Situ (In-Memory) computing architecture, based on a patent-pending Simple and Linear Fast Adder (SLFA), is a direct consequence of the proposed set theory. Algebraic invariants are also described with results bringing together set theory, discrete mathematics, number theory and algebraic structures. A canonical block form is defined for the Cayley table of finite groups, in terms of the numeric representation of groups. Automorphisms, and the minimal independent system of equations that define the group are given by the block form, among other information regarding groups' internal and external structure. The proposed construction of natural numbers is generalized to provide a simple and transparent construction of the continuum of real numbers, with a fast approximation for the numeric derivative that can be implemented with the SLFA. Infinite data structures are defined in the most efficient way with the smallest possible data type. A countable sequence of real numbers is coded in a single real number, and an infinite $\infty \times \infty$ real-valued matrix is also coded with a single real number. A real function is coded in a set of real numbers, and a countable sequence of real functions is also coded in a set of real numbers. These codings are meaningful and computable. Mathematical objects of all types are well assigned to tree structures in a proposed hierarchy of types.

Keywords: Structuralism; Set Theory; Fast Adder; Arithmetic Logic Unit; Finite Group; Real Number; Fast Derivative; Data Types; Tree; Complexity; Matrix Multiplication; Fully-Homomorphic Encryption

Introduction

The first section provides appropriate definitions for operation, group, field and linear space that allow simple constructions and proofs in the next sections. It is not a prerequisite for understanding the other sections. The reader who is not motivated by algebraic definitions may skip this section, but Definition 1 and Theorem 1, found in this section, should be understood before proceeding.

Today, every mathematician knows that the natural number zero is the empty set, $\emptyset$. However, most mathematicians in the time of Hilbert (and to this day) did not feel the need to justify the existence of the number 0, and argued that the nature of numbers is irrelevant and only their behavior is important. When Hilbert proposed his famous list of Problems (Paris, 1900), he paid special attention to the philosophical and practical

implications of the Axiomatic Method; abundant material on this subject is found in [Corry(2010)] and other articles and books from this author. Hilbert gave so much thought to this type of problem that the first two and the sixth problem are axiomatic questions. The first is the Continuum Hypothesis, the second is the problem of proving consistency and completeness of arithmetic, and the sixth is the axiomatization of physics. In the decades that followed, a monumental, collective, attempt was made to answer these metamathematical questions through the formalization of different set theories and their logical structures. An explosion of new ideas, concepts and methods was born from these philosophical questions. Physics had been thought to be nearly explained away at the end of the 19-th century, but then relativity and quantum theory came along and pulled the veil. Then, just when everybody thought that a perfect description, if not of physics, but at least of mathematics, was possibly attainable, Gödel tore down these illusions with his incompleteness theorems. He did it in the preconceived world where the Peano Axioms are true. The incompleteness theorems are true if one assumes that natural numbers are objects ruled by Peano's axioms. The Continuum Hypothesis and its negation have both been proven to be consistent with Zermelo-Fraenkel Set Theory, which is also a disturbing fact. It is a general consensus that the axiomatic base of natural numbers chosen, as well as the specific computable coding of numbers and other structures, is irrelevant and all that matters is that it can be done. But still the root of these problems lies somewhere and the foundational definitions are the main suspect, at least among some philosophers of mathematics that take these questions seriously. The question has been raised in the literature of philosophy of mathematics, structuralism, and set theory, if an alternative axiomatic system to Peano Arithmetic could exist that solves, not necessarily all, but some foundational questions of mathematics. Can there exist an alternative set theoretic definition of natural numbers that is specific enough that all mathematical objects are well defined but general enough that the widest possible collection of theorems is provable, all while carefully avoiding syntactic, semantic and logical paradox? These types of questions have taken various forms, and it is difficult enough to ask them in any of their forms. One such question is Benacerraf's Identification Problem [Benacerraf(1965)], whereby he argues that numbers are not sets, but rather sets are defined to have the properties we wish them to have. He says the nature of numbers is projected onto sets, but that does not imply a number *is* a set. He came to this conclusion because there are many ways to define natural numbers and prove all the properties of their structure, using sets. For example, Zermelo-Fraenkel Set Theory says $2 = \{\{\emptyset\}\}$ but Von-Neumann Set Theory defines $2 = \{\emptyset, \{\emptyset\}\}$. Both ways are consistent, and what is more, there are infinitely many other ways of defining natural numbers. Benacerraf came to the conclusion that the number 2 was neither of these sets, and in fact, wasn't any set at all. Hilbert had analyzed this problem in a similar way but with a different conceptual approach. But perhaps he thought it was not prudent to cause mathematicians more confusion than he was already going to cause with problems 1,2 and 6., because the problem he formulated was not included in his famous speech delivered in Paris. Hilbert knew that simply asking these questions was enough of a complication to mathematics and that this next problem, his 24th problem [Thiele(2003)], could wait. *"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs. Develop a theory of the method of proof in mathematics in general. Under a given set of conditions there can be but one simplest proof."* Here, a set theory is proposed that simplifies proofs and provides applications in pure and applied mathematics. The case is made that this is a unique and canonical set theory as part of a broader attempt in proposing an optimal universe for classical mathematics from number theory to analysis [Ramirez(2019)]. The second section describes natural numbers as the set of all hereditarily finite sets, **HFS**. An order $<$ and operations $\oplus, \odot$ are defined on **HFS**, isomorphic to the natural numbers $\mathbb{N}(<, +, \cdot)$. This axiomatization has important consequences in the representation and classification of finite and infinite objects and their functions.

A description for a "Simple and Linear Fast Adder" (SLFA-Patent Pending) sequential circuit with potentially comparable performance and more energy efficient than other fast adders [Uma(2012)],[Singh(2009)] is also discussed in the second section. Appendix "A" is a self-contained patent description of the SLFA. The second section also describes a method for addition of multiple operands that reduces the addition of n-many operands to the addition of two operands, along with a description of the multiple input adder and its implementations in vector operations. This method is implemented in a simple design by connecting multiple SLFAs in parallel. Multiplication of two numbers is possible by using this architecture for addition of partial products, making it a good replacement of Carry Save Adders [Lutz(1994)] also. This multi-operation multi-operand circuit is achieved by connecting b-many SLFAs, of n-bits each, using only parallel connections. It is a rectangular grid of nodes that can perform addition of n-many b-bit numbers, or add b-many pairs of n-bit numbers, or multiply two numbers. The same circuit can also fast approximate numerical derivatives, creating more opportunities for fundamental computing advantages. One of the basic problems with Von Neumann Architecture is that memory and logic

units are separate components with a relative distance between them and have to communicate back and forth via bus. The bandwidth of the bus is usually the bottleneck of the processor. Data migration between memory and logic is very costly and generates huge delay. One solution proposed decades ago was the concept of Computing-In-Memory, but it has several implementation difficulties. The proposed rectangular ALU architecture offers solutions to some of the basic problems for Computing-In-Memory because the logic topology is identical to the memory topology (rectangular grid with parallel connections). Meaning, the architecture can be scaled in a one-to-one fashion with memory in terms of area, delay and topological complexity giving it an advantage in high-energy consuming tasks used in neural network training, processing SHA functions [Sun], etc. This architecture has significant advantages over traditional ALUs (Arithmetic Logic Units) in high-speed implementations of ASIC (Area-Specific Integrated Circuit) because the logic and memory have the same topology and can be layered one on top of the other in a one-to-one manner. Some of the advantages related to Computing-In-Memory are found in [Wang(2023)]. Specific designs are possible for scalar, vector and matrix multiplication pipelines using minimum area, delay and energy consumption than other CPU architectures [Hennessy(1990)]. The circuit design for vector and matrix operations will only be discussed very briefly here, given its potential to generate a competitive edge in the design of ASICs used in Digital Signal Processing, advanced CGI applications, Neural Networks and AI Training among other important applications that heavily rely on processing power for computing large numbers of matrix multiplications, faster and cheaper. Some details will be described here and in future publications which will be made available at the author's homepage, and other details will be reserved for collaboration. If you are interested in this line of research, or to become a financial partner, you can email or visit the author's homepage.

In the third section, a method for coding a finite function as a natural number is detailed. If A, B are two finite sets and $f : A \to B$ a function, then a unique natural number N_f is assigned to the function. A linear order on all finite functions is obtained that is well behaved in several ways. There is a suborder induced on the subset of all finite permutations which is also well behaved in its own ways. Specifically, if η_m, η_n are permutations of $m < n$ many objects, respectively, then $\eta_m < \mathbf{1}_n \leq \eta_n \leq \mathbf{id}_n$ where $\mathbf{1}_n$ is the one-cycle permutation of n objects and $\mathbf{id}_n$ is the identity permutation of n objects. This representation gives a good definition for equivalent functions and permutations. Two finite functions are equivalent if they are represented by the same natural number.

In the fourth section, a formal definition of finite groups is given in terms of natural numbers, where a single natural number is used to code the group in a computable manner. Every finite group G, is well represented with a natural number N_G; if $N_G = N_H$ then H, G are in the same isomorphism class. This defines a linear order on the set of all finite groups, that is well behaved with respect to cardinality. In fact, if H, G are two finite groups such that $|H| = m < n = |G|$, then $H < \mathbb{Z}_n \leq G$. The linear order on groups is

$$\mathbb{Z}_1 < \mathbb{Z}_2 < \mathbb{Z}_3 < \mathbb{Z}_4 < \mathbb{Z}_2^2 < \mathbb{Z}_5 < \mathbb{Z}_6 < D_6 < \mathbb{Z}_7 < \mathbb{Z}_8 < Q_8 < D_8 < \mathbb{Z}_2 \oplus \mathbb{Z}_4 < \mathbb{Z}_2^3 < \mathbb{Z}_9 < \mathbb{Z}_3^2 < \cdots, \quad (1)$$

where D_n is the Dihedral group and Q_8 is the quaternion group. In general, $\mathbb{Z}_n \leq G$ if $|G| = n$ and the order is well behaved with respect to cardinality. The linear order induced on commutative groups, of n objects, also behaves well with respect to factorization of n. Intuitively, if $n = p^k$, then $\mathbb{Z}_{p^k} < \mathbb{Z}_p \oplus \mathbb{Z}_{p^{k-1}} < \mathbb{Z}_p^2 \oplus \mathbb{Z}_{p^{k-2}} < \cdots < \mathbb{Z}_p^k$. For example, $\mathbb{Z}_8 < \mathbb{Z}_2 \oplus \mathbb{Z}_4 < \mathbb{Z}_2^3$, and $\mathbb{Z}_9 < \mathbb{Z}_3^2$. If $n = p_1^{n_1} p_2^{n_2} p_3^{n_3} \cdots p_k^{n_k}$ is the prime factorization of n, then the commutative group $\mathbb{Z}_{p_1}^{n_1} \oplus \mathbb{Z}_{p_2}^{n_2} \oplus \mathbb{Z}_{p_3}^{n_3} \oplus \cdots \oplus \mathbb{Z}_{p_k}^{n_k}$ is the largest commutative group of n objects. For this purpose, a definition of canonical form for a group is given. The canonical form of a finite group is the Cayley table for the group, in a special block form. It reduces the problem of proving two finite groups are isomorphic to finding the canonical table of these groups. In the process of finding the canonical block form, the automorphisms and the minimal set of independent equations that define the group are obtained. An appendix is included where groups of less than ten objects are taken to their canonical block form. The canonical form and all twenty-four automorphisms of the symmetry group Δ_4 are also included in the appendix. A second appendix illustrates the canonical block form defined for finite groups.

The study of real numbers has been reduced to the study of natural numbers. However, the gap (conceptual and practical) between these two kinds of objects is enormous, in most treatments. The proposed set representation of natural numbers allows for the continuum of real numbers to be constructed as a natural extension of the set of natural numbers, without having to build intermediate structures such as $\mathbb{Z}$ or $\mathbb{Q}$. A natural number is a finite subset of **HFS**, while a real number is an infinite subset of **HFS**. Homomorphic Encryption is a very important problem in maximizing efficiency and security in cloud computing. The basic goals of HE can be achieved with a new, simple and foundational approach to this problem. Then, definitions of limits and continuity are given along with a fast derivative algorithm that approximates the numerical derivative of a real function.

Infinite data structures are also optimally defined. Just as a finite group is reduced to a natural number, similar results are true in the infinite case. A real function is a set of real numbers. More surprisingly, a countable sequence of real functions is also a set of real numbers. The complexity of objects is reduced to the minimum possible. In the last section, mathematical objects are well assigned to tree structures. Natural numbers are finite trees (objects of type 0), real numbers are infinite trees (objects of type 1). Sets of real numbers are objects of type 2, and a set of sets of real numbers is an object of type 4. A general description of types is briefly discussed.

Applications to be discussed in separate publications include a new coding of data structures in special memory units; an ALU architecture for optical computing schemes; a finite arithmetic important in the logical approach to artificial intelligence (AI) problems [Lovyagin(2021)]; a theoretical model on the probabilities of nuclear reactions, using a hexagonal grid that codes addition of integer numbers and multiplication of rationals through basic geometric relations; among others. The addition algorithm presented here relates the addition of numbers to the superposition of coupled waves; this is addressed in the conclusions. The lattice for determining nuclear reactions probabilities' with calculations on discrete number systems and the question of whether or not there exists an intrinsic connection between mathematics and physics, will be addressed in a later publication.

1 Groups, Fields and Linear Spaces

Operations are usually defined as a function of the form $(X \times X) \to X$. An alternate approach is taken here, by defining the operation of a group as a function $X \to (X \to X)$, which is known as Currying. A description of fields and linear spaces is also given in this section. The definitions and propositions, of this section, allow trivial proofs in the theory of set numbers of Section 2.

Definition 1. *Let G a non empty set, and $\boldsymbol{Aut}\, G$ the set of bijective functions of the form $G \to G$. A one-to-one function $G \to \boldsymbol{Aut}(G)$ is an operation on G. A set of functions $B \subseteq \boldsymbol{Aut}\, G$ is said to be balanced if $\boldsymbol{id}_G \in B$, and if $x \in B$ implies $x^{-1} \in B$. Let $* : G \to B$ a bijective function, for some balanced set $B \subset \boldsymbol{Aut}\, G$. If*

$$*(x) \circ *(y) = *(*(x)(y)), \tag{2}$$

for every $x, y \in G$, then $*$ is a group structure.

The functions $*(x)$ are called *operation functions of* $*$. The expression $*(x)(y) \in G$ is the image of y under the action of $*(x)$. Thus, $*(*(x)(y)) \in \mathbf{Aut}\, G$ is the image of $*(x)(y) \in G$ under the action of $*$.

Theorem 1. *The definitions of group and group structure are equivalent.*

Proof. Let $*$ a group structure and define an operation on the elements, $x * y = *(x)(y)$. It should be noted that $x*y = *(x)(y)$ is only a convention and depending on the specific function, this convention can vary. For example, an operation can be defined by $x * y = *(y)(x)$. The choice is irrelevant but must be consistent throughout, for each individual operation. Then, the following properties can be verified.

- Identity Element. There exists an object $e \in G$ such that $*(e) = \mathbf{id}_G$. Therefore, $*(e)(x) = x$ for all $x \in G$. This means $e * x = x$ for all $x \in G$. Now it must be shown $x * e = x$. It is true that $*(*(x)(e)) = *(x) \circ *(e) = *(x)$. Since $*$ is injective, it is also true that $*(x)(e) = x$.
- Inverse Element. Let $a \in G$, then there exists a unique $a^{-1} \in G$ such that $*(a^{-1}) = (*(a))^{-1}$ is the inverse function of $*(a)$. This is a direct consequence of the definition of balanced set. It will be proven that $a * a^{-1} = a^{-1} * a = e$. It is enough to prove $a^{-1} * a = e$. It can be verified that $a^{-1} * a = *(a^{-1})(a) = (*(a))^{-1}(a)$. Additionally, $*(a)(e) = a$. Therefore, the inverse function of $*(a)$ applies $(*(a))^{-1}(a) = e$.
- Associativity.

$$\begin{aligned} x * (y * z) &= *(x)(y * z) \\ &= *(x)(*(y)(z)) \\ &= (*(x) \circ *(y))(z) \\ &= *(*(x)(y))(z) \\ &= (*(x)(y)) * z \\ &= (x * y) * z. \end{aligned}$$

For the second part of this proof, it is enough to prove that a group G defines a group structure. The operation functions of the group structure are defined in terms of the cosets xG; define $*(x)$ by $g \mapsto_{*(x)} x * g$. It is easy to verify $*$ is an injective function and it is onto a balanced set. The associative property implies (2). □

The equivalence of groups and group structures is used to find their basic properties.

Theorem 2. *Let $G(*)$ a group with operation $*$. Then,*

1. *Right cancellation; $*(a)(c) = *(b)(c)$ implies $a = b$.*
2. *Left cancellation; $*(c)(a) = *(c)(b)$ implies $a = b$.*
3. *Uniqueness of identity and inverse elements.*
4. *Inverse of inverse; $(x^{-1})^{-1} = x$.*
5. *Existence of unique solutions; given $a, b \in G$ there exists a unique $x \in G$ such that $*(a)(x) = b$, and a unique $y \in G$ such that $*(y)(a) = b$.*

Proof. The first part requires to apply the function $*$, so that $*(*(a)(c)) = *(*(b)(c))$ which implies $*(a) \circ *(c) = *(b) \circ *(c)$. Right cancellation of functions gives $*(a) = *(b)$. It is concluded $a = b$ because $*$ is bijective. The second part can be proven similarly if left cancellation of functions is used.

Let e_1, e_2 be identity elements. Considering e_1 as identity, then $*(e_1)(e_2) = e_2$. If e_2 is the identity, then $*(e_1)(e_2) = e_1$. Therefore $e_1 = e_2$. The uniqueness of the inverse is trivial. If a_1, a_2 are inverse elements of a, then $*a(a_1) = e = *a(a_2)$ implies $a_1 = a_2$ because of left cancellation.

Let $y = x^{-1}$, so that $*(x)$ and $*(y)$ are inverse functions; $(*(x))^{-1} = *(y)$ and $(*(y))^{-1} = *(x)$. The inverse element of $y = x^{-1}$ is the object z such that $*(z)$ is the inverse function of $*(y)$. Therefore, x is the inverse of y and it is concluded $(x^{-1})^{-1} = x$.

For the last part, consider a, b fixed. Since $*(a)$ is a bijective function $G \to G$, there exists a unique $x \in G$ such that $*(a)(x) = b$. On the other hand, a function $*(y)$ that sends a to b needs to be defined. It is easy to see that $b * (a^{-1} * a) = b$, which can be rewritten as $(*(b) \circ *(a^{-1}))(a) = b$. The function $*(b * a^{-1}) = *(*(b)(a^{-1})) = *(b) \circ *(a^{-1})$ sends a to b so that $y = b * a^{-1}$ is the solution. Suppose there exists a second object, w, that satisfies the property of y. Then $*(y)(a) = *(w)(a)$ which implies $y = w$ if right cancellation is used. □

Proposition 1. *A group structure, $*$, defines a new function $\bar{*} : G \to \boldsymbol{Aut}(G)$ such that $\bar{*}(a)(b) = *(b)(a) = b * a$. The function $\bar{*}$ is also a group structure. The two group structures $*, \bar{*}$ are equivalent in the sense that they generate isomorphic groups.*

Proof. First prove $\bar{*}$ is a group structure. It must be shown $\bar{*}$ is a function $\bar{*} : G \to B$, where the image $Im\ \bar{*} = B$ is a balanced subset of $\mathbf{Aut}(G)$. Every object $a \in G$ is assigned a unique function $\bar{*}(a)$, and $\bar{*}(e) = \mathbf{id}_G$ for exactly one object $e \in G$. Next it will be proven $\bar{*}(a)$ is bijective. First of all, it is injective. Take $\bar{*}(a)(x) = \bar{*}(a)(y)$ which is equivalent to the expression $x * a = y * a$, then $x = y$ because of right cancellation. This proves $\bar{*}(a)$ is injective. To prove $\bar{*}(a)$ is onto G, let $b \in G$, then there exists a solution x to the equation $x * a = b$ which is equivalent to $\bar{*}(a)(x) = b$. This proves $\bar{*}(a)$ is a bijection. Now it will be proven the inverse function of $\bar{*}(a)$ is equal to $(\bar{*}(a))^{-1} = \bar{*}(a^{-1}) \in Im(\bar{*})$. By definition, $\bar{*}(a^{-1})(x) = x * a^{-1}$. Also, $\bar{*}(a)$ acts by $\bar{*}(a)(x * a^{-1}) = (x * a^{-1}) * a = x$, which implies the inverse function $(\bar{*}(a))^{-1}$ acts by $(\bar{*}(a))^{-1}(x) = x * a^{-1}$. This proves $\bar{*}(a^{-1}) = (\bar{*}(a))^{-1}$. So far, it has been proven the image of $\bar{*}$ is a balanced set. To prove $\bar{*}$ is injective, take two objects $x, y \in G$ such that $\bar{*}(x) = \bar{*}(y)$. Then, $x = \bar{*}(x)(e) = \bar{*}(y)(e) = y$. Now show $\bar{*}$ satisfies the associative property. For all $a, b \in G$

$$\begin{aligned} \bar{*}(\bar{*}(a)(b))(x) &= \bar{*}(b * a)(x) \\ &= x * (b * a) \\ &= (x * b) * a \\ &= \bar{*}(a)(x * b) \\ &= \bar{*}(a)(\bar{*}(b)(x)) \\ &= (\bar{*}(a) \circ \bar{*}(b))(x), \end{aligned}$$

for all $x \in G$. This proves $\bar{*}$ is a group structure.

Let $G(*)$ be the group generated by $*$ and $G(\bar{*})$ the group generated by $\bar{*}$, then x^{-1} is the same inverse element under both operations. The inverse of $a * b$, under $*$, is equal to $b^{-1} * a^{-1}$. The inverse of $a * b = b\bar{*}a$, under $\bar{*}$, is equal to $a^{-1}\bar{*}b^{-1} = b^{-1} * a^{-1}$. These two groups are isomorphic by $x \mapsto x^{-1}$. To prove, take $\phi(a * b) = (a * b)^{-1} = b^{-1} * a^{-1} = \phi(b) * \phi(a) = \phi(a)\bar{*}\phi(b)$. □

Definition 2. *In general, the functions $*(x)$ and $\bar{*}(x)$ are not equal. When they are equal, the object x is said to commute. A group is abelian if its two generating functions are equal, $* = \bar{*}$.*

Proposition 2. *Let $G(*)$ an operation on the set G. The following are equivalent statements.*

1. *The operation $*$ is associative.*
2. *$*(*(x)(y)) = *(x) \circ *(y)$ for all $x, y \in G$.*
3. *$*(x) \circ \bar{*}(y) = \bar{*}(y) \circ *(x)$ for all $x, y \in G$.*

Proof. The equivalence of 1. and 2. was proven in Theorem 1. Prove the equivalence of 1. and 3. Let $z \in G$, then

$$\begin{aligned}
(*(x) \circ \bar{*}(y))(z) &= *(x)(\bar{*}(y)(z)) \\
&= *(x)(z * y) \\
&= x * (z * y) \\
&= (x * z) * y \\
&= \bar{*}(y)(x * z) \\
&= \bar{*}(y)(*(x)(z)) \\
&= (\bar{*}(y) \circ *(x))(z)
\end{aligned}$$

Suppose 3. holds, then associativity can be proven,

$$\begin{aligned}
x * (z * y) &= *(x)(z * y) \\
&= *(x)(\bar{*}(y)(z)) \\
&= (*(x) \circ \bar{*}(y))(z) \\
&= (\bar{*}(y) \circ *(x))(z) \\
&= \bar{*}(y)(*(x)(z)) \\
&= \bar{*}(y)(x * z) \\
&= (x * z) * y
\end{aligned}$$

□

The following result is useful for consequent sections. It gives a practical means of proving associativity. If the elements of G commute and the operation functions also commute, then the operation is associative.

Proposition 3. *If $*$ is a commutative operation on the set G, and $*(x) \circ *(y) = *(y) \circ *(x)$, for all $x, y \in G$, then $*$ is associative.*

Proof. Given the hypothesis, the equalities $*(x) \circ \bar{*}(y) = *(x) \circ *(y) = *(y) \circ *(x) = \bar{*}(y) \circ *(x)$ hold true. The result follows from 3. and 1. of the last proposition. □

Definition 3. *Let $G(*)$ a group and let $H \subseteq G$ be a subset of the set G. Define $*_H$ as the function $*$ restricted to H. If $*_H$ is a group structure then it is a subgroup of $G(*)$.*

For $H \subset G$ to be a subgroup of G it is necessary that the image of H, under the action of $*_H(h)$, be equal to H, for all $h \in H$. In short, $*_H(h)[H] = H$, for all $h \in H$. This means *H is closed under the operation $*$.*

Definition 4. *Given two groups $G_1(*_1)$ and $G_2(*_2)$, a homomorphism is a function $\phi : G_1(*_1) \to G_2(*_2)$ such that $\phi(*_1(a)(b)) = *_2(\phi(a))(\phi(b))$, for every $a, b \in G_1$. The set of all homomorphisms from $G_1(*_1)$ to $G_2(*_2)$ is represented by the notation $\boldsymbol{Hom}(G_1, G_2)$, when no confusion arises with respect to the operations of each group.*

If the homomorphism is injective as function then it is called a monomorphism, and if it is surjective as function it is called an epimorphism. If the function is bijective it is an isomorphism, or automorphism when $\phi : G \to G$. The set of all automorphisms of $G(*)$ is represented with the notation $\boldsymbol{Aut}\ G(*)$.

The notation $\mathbf{Aut}(G)$ and $\mathbf{Aut}\ G(*)$ is used to differentiate between bijective functions and automorphisms.

Theorem 3. *Let X a set, then the composition operation $\circ$ is a group structure for the set of all bijective functions $\boldsymbol{Aut}\ X$. A subset $B \subseteq \boldsymbol{Aut}\ X$ that is balanced and closed under composition is a subgroup $B(\circ) \subset \boldsymbol{Aut}\ X$.*

A group structure $* : G \to B$, induces an isomorphism $* : G(*) \to B(\circ)$.

The composition operation is a group structure for the set of automorphisms $\boldsymbol{Aut}\ G(*)$. A balanced and closed subset, $\mathcal{B} \subseteq \boldsymbol{Aut}\ G(*)$, is a subgroup $\mathcal{B}(\circ) \subset \boldsymbol{Aut}\ G(*)$.

Proof. For the first part, consider the function $\circ : \mathbf{Aut}\ X \to \mathbf{Aut}(\mathbf{Aut}\ X)$. If $f \in \mathbf{Aut}\ X$, then $\circ(f) : \mathbf{Aut}\ X \to \mathbf{Aut}\ X$ is the function that acts by $\circ(f)(g) = f \circ g$. It will be proven $\circ$ is a bijective function onto a balanced set $Im\ \circ$. Every object in $\mathbf{Aut}\ X$ is assigned a function $\circ(f) \in \mathbf{Aut}(\mathbf{Aut}\ X)$. To see $\circ$ is injective, take two objects $f, g \in \mathbf{Aut}\ X$ and suppose $\circ(f) = \circ(g)$. This implies $f = f \circ \mathbf{id}_X = g \circ \mathbf{id}_X = g$. Now, prove the image of $\circ$ is balanced. The identity of G is mapped to $\circ(\mathbf{id}_G) \in \mathbf{Aut}(\mathbf{Aut}\ X)$ which is the identity of $\mathbf{Aut}(\mathbf{Aut}\ X)$. Also, for every $\circ(f) \in \mathbf{Aut}(\mathbf{Aut}X)$, the inverse function is $(\circ(f))^{-1} = \circ(f^{-1}) \in \mathbf{Aut}(\mathbf{Aut}\ X)$. The associative property is the usual associativity of composition of functions. This proves the first assertion of the first part. The second assertion of the first part is trivial. Take $B(\circ)$ balanced and closed under composition. This makes $B(\circ)$ a group.

For the second part, it must be shown $*$ is an isomorphism. From the first part of this theorem, $B(\circ)$ is a group. It is also known $*$ is a bijection. Definition 4 and associativity, in G, are used to verify $*(*(x)(y)) = *(x) \circ *(y) = \circ(*(x))(*(y))$, for all $x, y \in G$. This proves that the group structure $*$ produces an isomorphism $G(*) \to B(\circ)$, where $B(\circ)$ is the image of $*$ with the operation $\circ$.

The third part of this theorem is proven similarly to the first part. □

The *distributive property* is defined. Rings and fields are also defined.

Definition 5. *Let $K(+)$ a group with identity 0; the set $K - \{0\}$ is represented by K_0. Let $\cdot : K_0 \to \mathcal{C} \subset \boldsymbol{Hom}(K, K)$, an operation. The operation $\cdot$ distributes over $K(+)$, because*

$$\cdot(x)(+(a)(b)) = +(\cdot(x)(a))(\cdot(x)(b)),$$

for every $a, b, x \in K$.

Let $R(+)$ an abelian group, and let $\cdot$ a second operation that distributes over $R(+)$. Suppose $\cdot$ is associative and suppose $\cdot 1 = \boldsymbol{id}_R$ for a unique non trivial element $1 \in R_0$. A ring $R(+, \cdot)$ has two operations, and if $\cdot$ is commutative the ring is abelian.

Let $K(+, \cdot)$ a ring and suppose $Im(\cdot) = \mathcal{C} \subset \boldsymbol{Aut}\ K(+)$ is a balanced set of automoprhisms. Then $K(+, \cdot)$ is a skew field. If the ring $K(+, \cdot)$ is abelian, $K(+, \cdot)$ is a field.

A new notation $*x$ is used for the operation function $*(x)$. The distributive property holds when a group $K(\cdot)$ whose operation functions $\cdot x$, are homomorphisms on the original group $K(+)$. The conditions give the relations $\cdot x(0) = 0$, for all $x \in K$. Define $\cdot 0(x) = 0$. The operation function $\cdot 0$ is the trivial function $\mathbf{0} : K \to \{0\}$.

Corollary 1. *A field is an abelian group $K(+)$ together with a second abelian group $K(\cdot)$ that distributes over $K(+)$.*

Theorems 4 and 5, below, characterize linear spaces and modules. A linear space is an abelian group $V(\oplus)$, together with a field of automorphisms of $V(\oplus)$. Although these two theorems are not explicitly used in the following sections, it is useful for the last section on real numbers. Given an abelian group $V(\oplus)$ a second operation on $\mathbf{Hom}(V, V)$ is given, apart from composition. The operation $\oplus$ of V naturally induces a closed operation on $\mathbf{Hom}(V, V)$. This allows the definition of modules and linear spaces. Define addition of homomorphisms by $(f \oplus g)(x) = f(x) \oplus g(x)$. If $\mathcal{B} \subset \mathbf{Aut}\ V(\oplus)$, then the symbol $\mathcal{B}(\oplus)$ is used to emphasize that the set is being considered with addition, not composition. The trivial function $\mathbf{e} : V \to \{e\}$ acts as an identity object under

addition of homomorphisms, $f = f \oplus \mathbf{e} = \mathbf{e} \oplus f$. Let $f \in \mathbf{Aut}\ V(\oplus)$, and $-f \in \mathbf{Aut}\ V(\oplus)$ the automorphism defined by $-f(x) = -(f(x))$ where $-(f(x))$ is the additive inverse of $f(x)$; the notation $-x$ is used for the inverse of x under $\oplus$. It is easily verified that $f \oplus (-f) = \mathbf{e}$. A set of automorphisms $\mathcal{B}(\oplus)$ is balanced if $\mathbf{e} \in \mathcal{B}(\oplus)$, and if $f \in \mathcal{B}(\oplus)$ implies $-f \in \mathcal{B}(\oplus)$.

Lemma 1. *Let $V(\oplus)$ an abelian group with identity e, and $\mathcal{B}(\oplus) \subset \boldsymbol{Aut}\ V(\oplus)$ a balanced set. If $\mathcal{B}(\oplus)$ is closed under addition of automorphisms, then $\mathcal{B}(\oplus)$ is an abelian group with identity $\boldsymbol{e}$.*

Proof. This result provides an easy way of knowing if $\mathcal{B}(\oplus)$ is a group with addition of functions. It is required that $\mathcal{B}(\oplus)$ be balanced. Under addition of automorphisms, the inverse of f is the function $-f$ that acts by $x \mapsto -(f(x))$. The inverse of $\mathbf{id}_V$ is $-\mathbf{id}_V$ that makes $x \mapsto -x$. Associativity in $V(\oplus)$ implies associativity in $\mathcal{B}(\oplus)$. The commutative property in $\mathcal{B}(\oplus)$ also follows from the commutative property in $V(\oplus)$. □

Theorem 4. *Let $V(\oplus)$ an abelian group and suppose $\mathcal{B}(\circ) \subset \boldsymbol{Aut}\ V(\oplus)$ is a balanced, closed and commutative set of automorphisms with composition. Suppose $\mathcal{B}(\oplus)$ is balanced and closed with addition. Then $\mathcal{B}(\oplus, \circ)$ is a field, and $V(\oplus)$ is a linear space over the field of automorphisms $\mathcal{B}$. The elements of $V(\oplus)$ are called vectors.*

Proof. With respect to composition, it is sufficient to verify $\mathcal{B}(\circ)$ is balanced, closed and abelian. From the third part of Theorem 3, it is concluded $\mathcal{B}(\circ)$ is an abelian subgroup of $\mathbf{Aut}\ V(\oplus)$. If the conditions of the Lemma hold, then $\mathcal{B}(\oplus)$ is a group. Now it will be shown the distributive property holds. This is the simple statement that $\circ f$ is a homomorphism on $\mathcal{B}(\oplus)$, which is expressed by $f \circ (g \oplus h) = (f \circ g) \oplus (f \circ h)$ for every $f, g, h \in \mathcal{B}(\oplus, \circ)$. Let $x \in V$, then

$$\begin{aligned}(f \circ (g \oplus h))(x) &= f(g(x) \oplus h(x)) \\ &= f(g(x)) \oplus f(h(x)) \\ &= (f \circ g)(x) \oplus (f \circ h)(x) \\ &= ((f \circ g) \oplus (f \circ h))(x).\end{aligned}$$

This proves $\mathcal{B}(\oplus, \circ)$ is a field. Now it will be proven the structure of a linear space, in the classic sense, has been defined. The scalar product is simply the application of an automorphism to a vector. Let $f \in \mathcal{B}$, then the scalar product of f, with a vector $v \in V$, is defined as $f \cdot v = f(v)$. First, $(f \circ g)(v) = f(g(v)) = f \cdot (g \cdot v)$ because $\circ$ is the product of the field. Also, $f \cdot (u \oplus v) = (f \cdot u) \oplus (f \cdot v)$ because $f \in \mathbf{Aut}\ V(\oplus)$. By definition of addition of functions, $(f \oplus g) \cdot v = (f \cdot v) \oplus (g \cdot v)$. A linear space is defined by an abelian group V and a set of automorphisms (of V) that form a field. □

Similarly define a module M over a ring.

Theorem 5. *Let $M(\oplus)$ an abelian group and suppose $\mathcal{B}(\circ) \subset \boldsymbol{Hom}(M, M)$ is a closed set of homomorphisms with composition, and $\boldsymbol{id}_M \in \mathcal{B}(\circ)$. Suppose $\mathcal{B}(\oplus)$ is balanced and closed. Then $\mathcal{B}(\oplus, \circ)$ is a ring. The group $M(\oplus)$ is a module over the ring of homomorphisms $\mathcal{B}$. In general, the group $\mathcal{B}(\circ)$ is not abelian.*

2 Finite Sets and Natural Numbers

Finding mathematical objects that satisfy the properties of order and operation for natural and real numbers is not an easy task. This problem was taken up by many mathematicians at the beginning of the last century to formalize arithmetic and analysis. The solution was found that the statements of arithmetic, and later analysis, can be formulated using an elementary concept, *set*. Attempts were made to find set representations of numbers and to model the structure of natural numbers, using sets. Being an elementary concept, a set is not described in terms of other mathematical objects. Rather, mathematical objects are described using the language of sets. A set is a special kind of *collection of objects*. However, in order to avoid paradoxes, the notions of naive set theory had to be formalized. A formal system consists of a formal language (alphabet and grammar), and a deductive system (logical axioms and inference rules). The alphabet is made up of letters $a, b, c, \ldots, A, B, C, \ldots$ used for representing the objects in question, and logical symbols necessary for the deductive system. In formalizing mathematics a logical approach is used, and the definition of formal system provides an abstract space for unintelligent/mechanized manipulation of symbols. A logical system is a formal system together with a set of

non-logical axioms, and where the inference rules are first order logic or higher order logic. A classic example of a logical system is the Peano Arithmetic System. Even though the Peano Arithmetic System appeared to be sound at the beginning, many fatal flaws were pointed out in time. Extreme examples of these are Gödel's theorems. But, a simple and philosophical problem with the Peano axiomatization of natural numbers was that it did not provide answer to the question of what a number is. The entire edifice of mathematical objects is constructed based on the supposition of existence of 0, without saying exactly what object, if any, it is. The entirety of objects have their existence dependent on the existence of the object 0. If 0 exists then everything exists. That is why an even more foundational approach was given to the formalization of mathematics. The formal system of *Collections* which uses the letters $a, b, c, \ldots, A, B, C \ldots$ for collections. There is a single elementary binary relation. The symbol $\in$ is used for the binary relation of contention and the statement that a collection x is *element* of a collection X is represented with the symbol $x \in X$. Suppose the basic definitions for collections such as sub collection, arbitrary union of collections, arbitrary intersection of collections. A *Set Theory* is a logical system formed from the formal system of collections, plus non-logical axioms of set theory. Here, a set theory is proposed as a canonical basis for mathematics, in a non rigorous manner. It is similar to Zermelo-Fraenkel Set Theory but some of the set axioms are presented differently to construct natural numbers alternatively to Peano's Axioms, rendering simpler proofs, constructions and a range of practical applications.

The Zermelo-Fraenkel and Von Neumann constructions of natural numbers are the two most widely used definitions of natural numbers. In both cases, natural numbers are hereditarily finite sets and it can be proven that these sets satisfy Peano's Axioms. The set of all hereditarily finite sets, denoted **HFS**, consists of the sets obtained through the following procedure. The empty set is a member, $\emptyset \in \mathbf{HFS}$. Also, if $x_1, x_2, \ldots, x_n$ are objects in **HFS** then $\{x_1, x_2, \ldots, x_n\} \in \mathbf{HFS}$. Construct sets using these parameters to obtain all hereditarily finite sets. The collection $\{\emptyset\}$ is an object in **HFS**. Since $\emptyset$ and $\{\emptyset\}$ are in **HFS** the collection of these two objects, $\{\emptyset, \{\emptyset\}\}$, is also in **HFS**. Then, take $\emptyset$ and $\{\emptyset, \{\emptyset\}\}$ to find $\{\emptyset, \{\emptyset, \{\emptyset\}\}\} \in \mathbf{HFS}$. The sets $\{\emptyset\}$ and $\{\emptyset, \{\emptyset\}\}$ help construct the set $\{\{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \in \mathbf{HFS}$, etc. The first difficulty to formalize mathematics is to order hereditarily finite sets as natural numbers. The original methods are briefly discussed below, but it has been noted many times in the literature that these proofs and constructions of classic mathematical objects is quite long and cumbersome and involves artificial constructions. This leads to a lack of interest and basic comprehension from most mathematicians towards the axiomatic method and foundations of mathematics. Consequently, most mathematicians have a gap in the understanding of the nature of numbers, the basic blocks of mathematics. The mathematician probably knows that natural numbers can be built up in terms of finite sets, and that real numbers can be built using Cauchy Sequences of rationals, or Dedekind cuts, etc., but the conceptual understanding of these objects called numbers is difficult to grasp and widely considered to lack importance. Numbers are understood in terms of their interpretation as numbers, but hardly as objects in of themselves. The mathematical consensus is that the nature of a number is not important. It does not matter if we use one construction or another, what matters is that these constructions all describe objects whose properties and rules we can verify to satisfy the prerequisites of being a model of numbers. All that matters is that it can be done. It is known that any arithmetic axiomatization in first order logic is semantically incomplete but it is not generally true that the unprovable theorems for one axiomatization will be the same unprovable theorems for another axiomatization. One thing that is certain, however, is that different axiomatizations of numbers lead to different proof complexities for different theorems. In the two traditional constructions of natural numbers (Z-F and VN), discussed below, there are some advantages of one particular construction over the other in some aspects, and vice-versa. Here the argument is made that their exists a canonical set theory that yields new results and transparent proofs in many fundamental areas of mathematics including group theory, discrete mathematics, analysis, data types and computer science, among other areas. Definitions connecting these theories become apparent from the number construction proposed. A computable and meaningful function $\oplus 1 : \mathbf{HFS} \to \mathbf{HFS}$ that defines the order and operations of natural numbers will be the corner stone of our construction of natural numbers. Real numbers, and all other object types will be described in terms of natural numbers in a way that compuatble and meaningful representations and results of these objects can be given using the smallest possible data types.

The solution Zermelo and Fraenkel found is to order a sub collection of **HFS**. Notice it is trivial to order the sets $\emptyset$, $\{\emptyset\}$, $\{\{\emptyset\}\}$, $\{\{\{\emptyset\}\}\}, \ldots$, all of which are elements of **HFS**. If $x \in \mathbf{HFS}$, then $\{x\} \in \mathbf{HFS}$ is the successor. The order of natural numbers is trivially defined for $\mathbb{N}_< = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\{\{\emptyset\}\}\}, \ldots\}$. Addition of these sets has to be defined in such a way that it serves as a model of addition of natural numbers. This simply means the operation of addition has to be defined and its properties proven, which is usually tedious and laborious.

But, the real difficulty arises in understanding the constructions and objects used to describe more complicated structures such as the integer numbers, rational numbers, and real numbers. Integers are described in terms of natural numbers. Rational numbers are described in terms of integers, and real numbers are defined in terms of rational numbers. The last step, in building real numbers, involves objects that are difficult to describe and work with. This leads to a gap in most undergraduate students' learning since most programs do not include these constructions. Even modern day efforts to describe the real number system do not provide an easy way to understand the nature of the object called *real number*.

A second approach in the formal description of natural numbers is due to Von Neumann. He also orders a subset of **HFS**. In particular, the sets $\emptyset < \{\emptyset\} < \{\emptyset, \{\emptyset\}\} < \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} < \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\} \ldots$. If x is a natural number, its successor $x+1$ is the set $\{0, 1, 2, \ldots, x\}$. Here, $x < y$ if $x \in y$ which has some advantage in defining and proving properties of the order and addition. However, when building the later numerical structures, there is a similar situation as in the Zermelo-Fraenkel theory. The greater difficulty arises in building integers, rational numbers and real numbers. The constructions of Z-F and VN, and their technical aspects can be consulted in [Bernays(1991)].

The fact that there is at least two different constructions, gave way to another question, formally referred to as Benaceraff's Identification Problem. It has a great deal to do more with the Philosophy of Mathematics, than the mathematical models in use, but it still has wide implications. The main statement is set forth in a publication titled *"What Numbers Could Not Be"*, [Benacerraf(1965)]. The argument is made that numbers are actually not sets because there is no absolute way of describing them in terms of sets. For example, it cannot be known what object the number 3 is. Zermelo-Fraenkel say $3 = \{\{\{\emptyset\}\}\}$, but Von Neumann says $3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$. Who is to be believed? In fact, there are infinitely many consistent set constructions of natural numbers. Are all these constructions on the same standing? Or are some more convenient than others? Both Z-F and VN provide injective functions $\mathbb{N} \to \mathbf{HFS}$. Ackermann was able to find a bijection $\mathbb{N} \to \mathbf{HFS}$. This is known as BIT-Predicate or Ackermann Coding [Ackermann(1937)], and it has practical implications since mathematical systems can be modeled directly in terms of classic computational processes. It also referred to as BIT-Predicate because it maps the natural number $\sum_i 2^{x_i}$ to the set $\{x_1, x_2, \ldots, x_n\}$. It is important to note that Ackermann coding itself does not give means for adding numbers in any special manner. Although Ackermann coding represents natural numbers as sets, it still treats numbers as binary sequences for purposes of addition and uses the traditional means of operating. Namely, carry over algorithms with its intrinsic time delays. This also makes it difficult to use Ackermann Coding as the basis for an axiomatic set theory. The reader is invited to attempt to find a simple description of the addition operation of natural numbers, in terms of elementary set operations. For example, the reader should try to define a computable and finite process that inputs A, B and outputs the set $A \oplus B$ which is the Ackermann coding for the sum of the numbers corresopnding to A and B? For eaxmple, if $A = \{\emptyset, \{\emptyset\}\} = 3$ and $B = \{\{\emptyset\}\} = 2$, the result of the process should be $5 = \{\emptyset, \{\{\emptyset\}\}\}$, and similarly for any two sets. Solutions to this are not trivial either; remember the goal is to express the operation in terms of elementary set operations (union and intersection of sets). This section is a proposal for a representation of natural numbers using BIT-Predicate, but addition is defined as a finite state machine that reaches stable state in logarithmic time, and the structure of natural numbers is defined by an alternate axiomatic base. The addition operation defined for sets can be easily extended from natural numbers to real numbers. A theory of types enveloping all classic mathematical objects is briefly discussed in the conclusions for later work.

2.1 Motivation

When adding two numbers, natural or real, there is one major difficulty involved. Addition is a special prefix problem which means that each sum bit is dependent on all equal or lower input bits, as noted in [Ladner and Fischer(1980)]. The carrying algorithm can also be consulted in [Metropolis, Rota and Tanny(1980)]. When adding numbers in base 10 (or base $b > 2$), sequences of digits must be used to represent natural numbers. To write a natural number in base b, each digit in the sequence will specify how many times the corresponding power of b is considered; digits will take a value in $\{0, 1, 2, \ldots, b-1\}$. The order of the sequence is important to know how many times each power of b is added. But, with binary representation ($b = 2$), a more elementary language suffices. It is no longer needed to specify how many times a power is added. It is sufficient to specify if a power is considered or not because digits of the sequence take values in $\{0, 1\}$. Essentially, this allows for a natural number to be determined by a *set* of smaller natural numbers; those that appear as power in binary form. For example, the number $7 = 2^0 + 2^1 + 2^2$ is the set $\{0, 1, 2\}$.

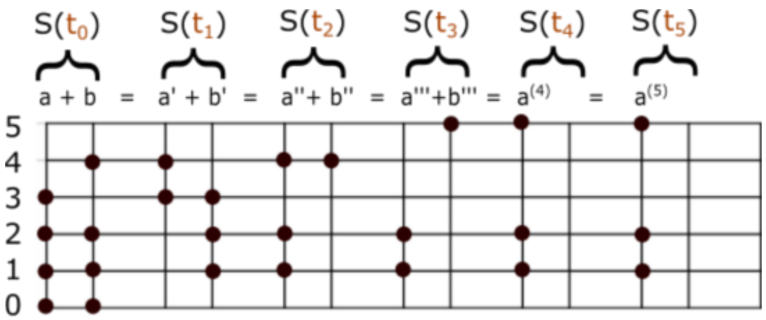

Figure 1: Graphic Representation of $15+23=38$. The sum of two sets is a process that ends in finite steps. The addition is iterated a finite number of times before the system stabilizes. In this example, the system stabilizes after three iterations. Observe that two disjoint set numbers form a stable system. This means $A \oplus B = A \cup B$ if $A \cap B = \emptyset$; the sum of disjoint sets coincides with the union.

In this proposal, addition is treated in terms of sets, and not sequences. The sum $7+13 = (2^0+2^1+2^2)+(2^0+2^2+2^3)$, is the sum of sets $\{0,1,2\} \oplus \{0,2,3\}$. Two new sets are formed - symmetric difference and intersection. The powers that are not repeated $\{1,3\}$, and the powers that repeat $\{0,2\}$. To add a power of 2 with itself (i.e., numbers in the intersection), add "1" to that power, $2^n + 2^n = 2^{n+1}$. The sum is rewritten as $7+13 = (2^1+2^3)+(2^{0+1}+2^{2+1})$. The first term 2^1+2^3 represents symmetric difference $A \triangle B$, while the second term $2^{0+1}+2^{2+1} = (2^0+2^2)+(2^0+2^2)$ represents the intersection. The sum has been reduced to $7+13 = (2^1+2^3)+(2^1+2^3)$. This step is iterated and the result is $7+13 = 2^{1+1}+2^{3+1} = 2^2+2^4 = 20$. The system has reached a stable state because there are no more repeated powers, $\{0,1,2\} \oplus \{0,2,3\} = \{2,4\}$.

This addition of finite sets is isomorphic to addition of natural numbers. To perform the addition of A, B form two new sets $A' = A \triangle B$ and $B' = s(A \cap B)$, where s is the function that adds 1 to the elements of its argument. The addition of these two new sets is the same as the original addition $A \oplus B = A' \oplus B'$ because it is equivalent to a rearrangement of the powers of 2. The terms A, B are rearranged into two new terms. The term A' consists of the non repeated powers (symmetric difference) and the term B' consists of the repeated powers (intersection). It is guaranteed that in a finite number of iterations the intersection $A^{(k)} \cap B^{(k)} = \emptyset$ becomes the empty set. This yields the final answer $A^{(k+1)}$, because $A \oplus B = A^{(k+1)} \oplus B^{(k+1)} = A^{(k+1)} \oplus s(\emptyset) = A^{(k+1)}$.

Apply this reasoning with another example, $15+23=38$, from Figure 1. This is the addition $A \oplus B = \{0,1,2,3\} \oplus \{0,1,2,4\}$ because $15 = 2^0+2^1+2^2+2^3$ and $23 = 2^0+2^1+2^2+2^4$. First find $A' = A \triangle B = \{3,4\}$ and $A \cap B = \{0,1,2\}$, so that $B' = \{0+1, 1+1, 2+1\} = \{1,2,3\}$. Iterate the process with $A'' = A' \triangle B' = \{1,2,4\}$ and $B'' = s(A' \cap B') = \{3+1\} = \{4\}$. Continuing in this manner, a stable state is reached because $A''' \cap B''' = \{1,2\} \cap \{5\} = \emptyset$.

The process described herein is a finite state machine. Each state is composed of two columns. Each column is a finite configuration of energy-levels representing one natural number, as is illustrated in Figure 1. A particle in the basic level "0" is worth 1 unit, and a particle in level "1" is worth 2 units. A particle in level "2" is worth 4 units, and in general a particle in level "n" is worth $2n$ units. A finite configuration of particles in a column represents a set number, so that each state is a pair of natural numbers. As shown in Figure 1, the initial state $S(t_0)$ is given by the inputs A, B. The next state, $S(t_1)$ is given by two new columns. The configuration of the left column is given by the energy levels that were not repeated in state $S(t_0)$. The right column in $S(t_1)$ is given by the objects that repeat but displaced one level up. The configuration of state $S(t_2)$ is defined similarly in terms of state $S(t_1)$. The left column of state $S(t_2)$ is given by the energy levels not repeated in state $S(t_1)$. The configuration in the right column of state $S(t_2)$ is given by the energy levels repeated in state $S(t_1)$ but displaced one level up. In general, the left column of state $S(t_{k+1})$ is given by the energy levels not repeated in state $S(t_k)$. The right column of state $S(t_{k+1})$ is given by a displacement, one level up, of the energy levels repeated in state $S(t_k)$. In a finite number of steps, a stable state is reached, where no particle occupies the right column. The result of the sum is given in the left column.

It should not be difficult for the reader to prove the number of steps to reach stability is bounded above by $\max(A \cup B) + 2$. The addition $A \oplus B = \{0,1,2\} \oplus \{0\}$ is one case that reaches the stable state in four steps (worst case scenario). Adding a unit to the string, $\{0,1,2,\ldots,k\} \oplus \{0\}$, gives the trivial result $\{k+1\}$ in $k+2$

steps. This is the set number expression for the equivalent arithmetical expression $1+(1+2+4+\ldots+2^k)=2^{k+1}$. In general, $\{n,n+1,n+2,\ldots,n+k\}\oplus\{n\}=\{n+k+1\}$ is equivalent to the arithmetical expression $2^n+(2^n+2^{n+1}+2^{n+2}+\ldots+2^{n+k})=2^{n+k+1}$. In fact, this type of string allows us to calculate the iterations for stability given a sum of two numbers. The longest string will give us the total number of iterations before stability. Going back to the bound on the number of iterations, it can be easily seen that it actually does not depend on the maximum value of the set. A more precise bound can be obtained. For example, the number of iterations for calculating $\{0,1\}\oplus\{0\}$ is equal to the number of iterations for calculating $\{5,6\}\oplus\{5\}$. However, the bounds are two very different numbers $\max\{0,1\}$ and $\max\{5,6\}$. To come up with a better bound on the number of iterations, observe that the number of iterations does not need to depend on how large the numbers are. To understand this, build a worst case scenario. Let A,B two sets such that $\#(A)+\#(B)=3$. If the intersection $A\cap B=\emptyset$ is empty, the system is stable from the initial state. To maximize the number of iterations, build a string as above, $A=\{n,n+1\}$ and place the third element in the bottom $B=\{n\}$. This system requires a total of two iterations to stabilize. Any other configuration of three elements will require at most one iteration to stabilize. Now, suppose a total of $k+2$ objects; $\#(A)+\#(B)=k+2$. A string provides a worst case scenario; a string plus the smallest number of the string. The sum $A\oplus B=\{n,n+1,\ldots,n+k\}\oplus\{n\}$ takes a total of $k+2$ iterations to reach stability because of the string. Now, it is easy to see that there is more than one worst case scenario. Change one of the elements from A, to the set B, and the number of iterations will be the same. Doing this with $n+1$, the result is $\{n,n+2,n+3,\ldots,n+k\}\oplus\{n,n+1\}$ which will take $k+2$ iterations to stabilize. This can be done with any of the elements of A, and with more than one. In general, if $A\triangle B=\{n+1,\ldots,n+k\}$ and $A\cap B=\{n\}$, then exactly $k+2$ iterations are needed to stabilize. More generally, two sets with non empty intersection will have at least one string of this form. The longest of such strings will determine the smallest number of iterations needed for stability.

Suppose $A,B\subseteq\{0,1,2,\ldots,N-1\}$ are two random set numbers and let $x\leq N-1$. The probability that $x\in A\triangle B$ is equal to $\frac{2}{4}=\frac{1}{2}$. Then, the probability P that there exist $n,k\in\mathbb{N}$ such that $\{n+1,\ldots,n+k\}\subseteq A\triangle B$, is equal to the probability of k consecutive heads in N fair coin tosses. Therefore, the probability of a N-bit addition taking $k\leq N$ iterations to complete, is equal to $\frac{P}{4}$. On average, it takes $\log_2 N$ iterations to calculate a N-bit addition. The probability of taking more iterations than $\log_2 N$ decreases fast. These coin toss problems are standard. A Simple and Linear Fast Adder (Patent Pending) is described in the first appendix.

In the next subsection addition is formalized for finite sets, and it is isomorphic to addition of natural numbers $\mathbb{N}_+$. In the first section, a definition of operation was given that does not use a cartesian product in the domain. An operation is a function whose image is a space of functions itself. It is a one-to-one function $*:A\to(AfA)$ into the set AfA. The image, AfA, is the set of all one-to-one functions of the form $A\to A$. The operation $\oplus$ that defines addition of sets is defined in terms of its operation functions $\oplus n$ by $\oplus n(x)=n\oplus x$. The function $\oplus 1$ generates the hereditarily finite sets, and it also generates the set of operation functions $\oplus n$. The functions $\oplus n$ are the powers of composition, $\oplus 2=\oplus 1\circ\oplus 1$, $\oplus 3=\oplus 1\circ\oplus 1\circ\oplus 1$, etc. Define two base cases $0=\emptyset$ and $1=\{\emptyset\}$, along with a function $\oplus 1:\mathbf{HFS}\to\mathbf{HFS}$. To add 1 to a set A, apply the function $\oplus 1$ to the set A,

$$\oplus 1(A)=(A\triangle 1)\oplus s(A\cap 1), \tag{3}$$

where $s:\mathbf{HFS}\to\mathbf{HFS}$ sends every set $X=\{x\}_{x\in X}$ to the set $s(X)=\{\oplus 1(x)\}_{x\in X}$. Applying the function s to the set X simply means $\oplus 1$ is applied to every object of X. In the following calculations, use the fact that $s(\emptyset)=\emptyset$. Furthermore, define $A\oplus\emptyset=\emptyset\oplus A=A$ which simply defines $\emptyset$ as the identity element. First, use the definition of the operation to find $\oplus 1(0)=(0\triangle 1)\oplus s(0\cap 1)=1\oplus s(\emptyset)=1\oplus\emptyset=1$. The function $\oplus 1$ generates every element of **HFS** when applied successively.

$$\begin{aligned}
2 &= \oplus 1(1)=(1\triangle 1)\oplus s(1\cap 1)=\emptyset\oplus s(1)=s(1)=\{\oplus 1(0)\}=\{1\}\\
3 &= \oplus 1(2)=(2\triangle 1)\oplus s(2\cap 1)=(\{1\}\triangle\{0\})\oplus s(\{1\}\cap\{0\})=\{0,1\}\oplus s(\emptyset)\\
&= \{0,1\}\oplus\emptyset=\{0,1\}\\
4 &= \oplus 1(3)=(3\triangle 1)\oplus s(3\cap 1)=(\{0,1\}\triangle\{0\})\oplus s(\{0,1\}\cap\{0\})=\{1\}\oplus s(\{0\})\\
&= \{1\}\oplus\{\oplus 1(0)\}=\{1\}\oplus\{1\}
\end{aligned}$$

A suitable definition for $\{1\}\oplus\{1\}$ must be found, and in general a suitable definition for $A\oplus B$ is needed. Extend the definition in the obvious way,

$$A\oplus B=(A\triangle B)\oplus s(A\cap B).$$

Now the number 4 can be found.

$$2 \oplus 2 = (2 \triangle 2) \oplus s(2 \cap 2) = \emptyset \oplus s(2) = s(2) = \{\oplus 1(1)\} = \{2\}.$$

This simply means the set $\{2\} = \{\{1\}\} = \{\{\{\emptyset\}\}\}$ is the object known as *the number 4*. Continue to generate sets, by applying the function $\oplus 1$ to the result.

$$\begin{aligned}
5 &= \oplus 1(4) = (4 \triangle 1) \oplus s(4 \cap 1) = \{0,2\} \oplus s(\emptyset) = \{0,2\} \\
6 &= \oplus 1(5) = (5 \triangle 1) \oplus s(5 \cap 1) = \{2\} \oplus s\{0\} = \{2\} \oplus \{1\} = (\{2\} \triangle \{1\}) \oplus s(\{2\} \cap \{1\}) \\
&= \{1,2\} \oplus s(\emptyset) = \{1,2\} \\
7 &= \oplus 1(6) = (6 \triangle 1) \oplus s(6 \cap 1) = \{0,1,2\} \oplus s(\emptyset) = \{0,1,2\} \\
8 &= \oplus 1(7) = (7 \triangle 1) \oplus s(7 \cap 1) = \{1,2\} \oplus s(\{0\}) = \{1,2\} \oplus \{1\} \\
&= (\{1,2\} \triangle \{1\}) \oplus s(\{1,2\} \cap \{1\}) = \{2\} \oplus s(\{1\}) = \{2\} \oplus \{2\} \\
&= (\{2\} \triangle \{2\}) \oplus s(\{2\} \cap \{2\}) = \emptyset \oplus s\{2\} = s(\{2\}) = \{3\} \\
9 &= \oplus 1(8) = (8 \triangle 1) \oplus s(8 \cap 1) = \{0,3\} \oplus s(\emptyset) = \{0,3\} \\
10 &= \oplus 1(9) = (9 \triangle 1) \oplus s(9 \cap 1) = \{3\} \oplus s(\{0\}) = \{3\} \oplus \{1\} = (\{3\} \triangle \{1\}) \oplus s(\{3\} \cap \{1\}) \\
&= \{1,3\} \oplus s(\emptyset) = \{1,3\}.
\end{aligned}$$

Notice, that the sum of two disjoint sets is the union. When referring to hereditarily finite sets, in this manner, they are called *set numbers*. Let N be a natural number with binary representation $\sum_{i=1}^{n} 2^{a_i}$, then N is the set number $\{a_1, a_2, \ldots, a_n\}$. For example, $5 = \{0,2\}$ because $5 = 2^0 + 2^2$, while $6 = \{1,2\}$ because $6 = 2^1 + 2^2$. The number $11 = \{0,1,3\}$ can easily be found.

$$11 = 5 \oplus 6 = \{0,2\} \oplus \{1,2\} = \{0,1\} \oplus s(\{2\}) = \{0,1\} \oplus \{3\} = \{0,1,3\}.$$

Another way of finding 11 is with the addition

$$11 = 7 \oplus 4 = \{0,1,2\} \oplus \{2\} = \{0,1\} \oplus s(\{2\}) = \{0,1,3\}.$$

2.2 Formalization

The constructions here described are carried out in a slightly modified version of Zermelo-Fraenkel Set Theory. The axioms needed for the constructions of this section are listed. The Axiom of Extensionality which defines equality of sets; two sets are equal if and only if they contain the same elements. The Axioms of Union and Subsets are also included; the Axiom of Subsets allows the construction of the intersection of sets.

To construct all hereditarily finite sets, from the empty set, their is a well defined procedure. Their exists a set $\mathbb{N}$ such that $\emptyset \in \mathbb{N}$, and if $x_1, x_2, \ldots, x_n$ are elements of $\mathbb{N}$ then $\{x_1, x_2, \ldots, x_n\} \in \mathbb{N}$. However, the objects of **HFS** are not generated in a particular order; there is no canonical order in constructing hereditarily finite sets. There are infinite ways of building these sets one by one. For example, once the sets $\emptyset$ and $\{\emptyset\}$ have been found, the sets $\{\{\emptyset\}\}$ and $\{\emptyset, \{\emptyset\}\}$ can be constructed. It is quite clear that 0 should be $\emptyset$ and 1 should be $\{\emptyset\}$. But, which of the two new sets should be the number 2? Ackermann Coding establishes that the number 2 is the set $\{\{\emptyset\}\}$, and 3 is the set $\{\emptyset, \{\emptyset\}\}$. Notice a fundamental difference the Ackermann Coding has with Z-F and VN constructions. Adding one unit to a Z-F or VN natural number is a simple procedure. In the first case, $\{x\}$ is the successor x, and in the second case the successor is $x \cup \{x\}$. With Ackermann Coding the situation is different because the rule for building new sets does not give an order to the sets built. Ackermann Coding builds sets without ordering them. To know the order of the hereditarily finite sets being built (the natural number corresponding to each element of **HFS**), the binary representation of natural numbers has to be known beforehand. The order given to finite sets is known only in hindsight when the binary representations of natural numbers is worked out. This was the difficulty in using Ackermann Coding as an axiomatic base of natural numbers. The problem with this was that there was no simple description of addition in terms of set operations. Numbers had to be operated as binary sequences which takes several layers of set theoretical constructions making the formalization long and

difficult. The function $\oplus 1$ provided here is the first proposal found in the literature of a computable function that determines the Ackermann Coding successor of a hereditarily finite set, in terms of elementary set operations. A recursive set function $\oplus 1$ is proposed that defines addition of natural numbers in BIT-Predicate. This function depends on union and intersection of sets in **HFS** (the symmetric difference of sets can be expressed in terms of union and intersection). The set representation of every natural number is obtained by applying the function $\oplus 1$ to $\emptyset$, a finite number of times. Moreover, the addition of two N-bit numbers is a finite state machine that reaches a stable state in $\log_2 N$ iterations, on average. For practical purposes in the implementation of addition in digital circuits, the Ackermann Coding of numbers is the most convenient and has therefore been referred to as BIT-Predicate. However, two natural numbers given in binary form are added with the carry-over algorithm which treats natural numbers as binary sequences, which introduces an intrinsic carry-over delay. These delays can be overcome with parallel computing algorithms but at a huge area and energy consumption cost, among other problems. An alternative method for parallel addition is given here that can be implemented with a simple and linear circuit with low-energy consumption; the patent-pending SLFA is found in the first appendix.

Definition 6. *Let $0 = \emptyset \in \boldsymbol{HFS}$ and $1 = \{\emptyset\} \in \boldsymbol{HFS}$. Define the set operation $m \oplus n$ with operation functions $\oplus n : \boldsymbol{HFS} \to \boldsymbol{HFS}$ such that $\oplus n(m) = m \oplus n = (m \triangle n) \oplus s(m \cap n)$, where $s(m \cap n) = \{\oplus 1(x)\}_{x \in m \cap n}$. In particular, the function $\oplus 1$ acts on sets by $\oplus 1(m) = (m \triangle \{\emptyset\}) \oplus s(m \cap \{\emptyset\})$.*

Let $n \in \boldsymbol{HFS}$ be the set obtained from $\oplus 1(\oplus 1(\cdots(\oplus 1(0))) = \oplus 1 \circ \oplus 1 \circ \cdots \circ \oplus 1(0)$. This can be expressed as $n = \oplus 1^n(0)$.

An axiom is also needed to ensure natural numbers are infinite. The infinity axiom proposed here is given in terms of a bijection $\oplus 1$. Let $\mathbb{N} = \mathbf{HFS}$, and $\mathbb{N}_1 = \mathbf{HFS}/\{\emptyset\}$ the set of hereditarily finite sets without the empty set. The Infinity Axiom is the statement that the function $\oplus 1 : \mathbb{N} \to \mathbb{N}_1$ is a bijection. This ensures the sets generated are infinite. The object 0 is sent to the object 1. Since 0 is not in the image set, it is not the image of 1. Also 1 cannot be the image of 1 because it is already the image of 0. This means a new object, call it 2, is the image of 1. The argument continues in this manner to prove there are infinitely many natural numbers. The set of arrows of $0 \to 1 \to 2 \to 3 \to \cdots$ are the ordered pairs of the function $\oplus$ that adds one unit, $x \to x \oplus 1$. If this set of arrows is extended to include transitive arrows, then the arrows give the total order of the natural numbers. The functions $\oplus 1^n$ and $\oplus n$ are both assigned to $n = \oplus 1^n(0)$. The equality $\oplus n = \oplus 1^n$ is taken as an axiom, for every $n \in \mathbb{N}$.

It is proven below that the operation functions $\oplus 1^n$ satisfy the properties of commutativity and associativity. To complete the operation of addition in terms of operation functions, the identity function is assigned to the empty set so that $\oplus 0(m) = m$ for every $m \in \mathbf{HFS}$. In the last sub section it has been illustrated how to find $\oplus 1(1)$, $\oplus 1(2), \ldots$. When carrying out the calculations for $3 \oplus 1$, it was recognized that it is necessary to know the value of $\oplus 2(2)$. Continuing to apply $\oplus 1$, more calculations of the form $\oplus n(m)$ are encountered. But, the operation function for $\oplus n$ is explicitly dependent of $\oplus 1$. The functions $\oplus 1^n$ are defined as powers of $\oplus 1$, but to find $\oplus 1$ it is also needed to start finding $\oplus n$. The operation functions $\oplus 1^n$ and $\oplus n$ build each other simultaneously, as has been seen in the calculations of the previous section. The commutative property of $\oplus$ is trivial, using the fact that $f^n \circ f^m = f^m \circ f^n$ for a function f. Addition is $m \oplus n = \oplus n(m) = \oplus n(\oplus m(0)) = \oplus 1^n(\oplus 1^m(0)) = \oplus 1^m(\oplus 1^n(0)) = n \oplus m$. The reader may skip the proofs below, through Proposition 5, to the description of addition of several inputs and multiplication.

The easiest way to prove associative property of set addition is to prove the functions $\oplus m$ and $\bar{\oplus} n$ commute, for every set numbers m, n. Given that commutativity holds, it is true that $\oplus n = \bar{\oplus} n$. Because of Proposition 3, it is sufficient to prove the commutative property holds for operation functions, $\oplus m \circ \oplus n = \oplus n \circ \oplus m$.

Proposition 4. *The associative property holds for $\oplus$.*

Proof. By definition, the function $\oplus n$ is the function $\oplus 1$ applied a total of n times, $\oplus n(a) = \oplus 1^n(a)$. The operation functions $\oplus m$, $\oplus n$ commute,

$$\begin{aligned}
(\oplus n \circ \oplus m)(a) &= \oplus n(\oplus m(a)) \\
&= \oplus 1^n(\oplus 1^m(a)) \\
&= \oplus 1^m(\oplus 1^n(a)) \\
&= \oplus m(\oplus n(a)) \\
&= (\oplus m \circ \oplus n)(a).
\end{aligned}$$

□

A linear order has been given $0 \to_{\oplus 1} 1 \to_{\oplus 1} 2 \to_{\oplus 1} 3 \to_{\oplus 1} 4 \to_{\oplus 1} \cdots$, in terms of addition. The transitive arrows are aggregated to the set of arrows that defines the operation function $\oplus 1$. For example, since $0 \to 1$ is an arrow of $\oplus 1$, and $1 \to 2$ is an arrow of $\oplus 1$, then $0 \to 2$ is a transitive arrow. In the next section it will be specified exactly what is meant by an arrow or an ordered pair.

Let A, B two set numbers, then $A < B$ is true if and only if there exists a set number $m \neq \emptyset$ such that $B = A \oplus m$. Applying $\oplus n$ to B,

$$B \oplus n = \oplus n(A \oplus m) = \oplus n(\oplus m(A)) = \oplus m(\oplus n(A)) = \oplus m(A \oplus n) = (A \oplus n) \oplus m.$$

This implies $A \oplus n < B \oplus n$. That is to say, the operation preserves the order; $A < B$ implies $A \oplus n < B \oplus n$. The order is obviously transitive. Let $B = A \oplus m$ and $C = B \oplus n$. Then $C = (A \oplus m) \oplus n = A \oplus (m \oplus n)$. Since $m \oplus n$ is not the empty set, it is true that $A < C$.

The following result provides a practical way of determining the natural order of **HFS**. Let A, B two distinct natural numbers and consider their symmetric difference $A \triangle B$ which is not empty and is bounded. That is to say, $\max(A \triangle B)$ exists. Furthermore, this maximum is in exactly one of the two sets, not in both. Compare the two sets in terms of this object, $\max(A \triangle B)$. The set that contains this object is the largest of the two. For example, $15 = \{0, 1, 2, 3\} < \{4\} = 16$ because $A \triangle B = \{0, 1, 2, 3, 4\}$ and $\max(A \triangle B) \in \{4\} = 16$. The set number $A = \{1, 5, 6\} = 98$ is smaller than the set number $B = \{0, 7\} = 129$ because $\max(A \triangle B) = 7 \in B$. In the following proof it will be seen that the order is anti symmetric. It will also be seen that every pair of set numbers A, B is comparable; the order is total.

Theorem 6. *Let A, B two set numbers, then $A < B$ if and only if $\max(A \triangle B) \in B$.*

Proof. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set number, and suppose B is a set number such that $A < B$. From (A5), the set number B is obtained by successively adding 1 to the set number A. This means $B = \oplus 1^n(A)$ for some $n \in \mathbb{N}$. It will be proven $\max(A \triangle B) \in B$ for every $B > A$. In this proof, the fact that $A \oplus B = A \cup B$, if $A \cap B = \emptyset$, will be used. Start with $A \oplus 1 = \{a_1, a_2, \ldots, a_n\} \oplus \{0\}$. There are two cases; $0 \notin A$ or $0 \in A$. In the first case, $A \oplus 1 = \{0, a_1, a_2, \ldots\ a_n\}$ which implies $\max(A \triangle (A \oplus 1)) = \max\{0\} = 0 \in A \oplus 1$. Now consider the second case; suppose $a_1 = 0$. Then, $A \oplus 1 = \{0, a_2, \ldots, a_n\} \oplus \{0\} = \{a_2, a_3, \ldots, a_n\} \oplus \{1\}$. There are two sub cases; $1 \notin A$ or $1 \in A$. In the first case, $A \oplus 1 = \{1, a_2, a_3, \ldots, a_n\}$ and the result follows, $\max(A \triangle (A \oplus 1)) = \max\{0, 1\} = 1 \in A \oplus 1$. In the second case, $a_2 = 1$ and this implies $A \oplus 1 = \{a_3, a_4, \ldots, a_n\} \oplus \{2\}$.

More generally, suppose k is the smallest number not in A. Then, $A = \{0, 1, \ldots, k-1, a_{k+1}, a_{k+2}, \ldots, a_n\}$, where $k < a_{k+1} < a_{k+2} < \ldots < a_n$. Applying $\oplus 1$ yields

$$A \oplus 1 = \{k, a_{k+1}, a_{k+2}, \ldots, a_n\}.$$

Then $\max(A \triangle (A \oplus 1)) = k$, which proves $\max(A \triangle (A \oplus 1)) \in A \oplus 1$. Applying $\oplus 1$ again, the result is

$$A \oplus 2 = \{k, a_{k+1}, a_{k+2}, \ldots, a_n\} \oplus \{0\} = \{0, k, a_{k+1}, a_{k+2}, \ldots, a_n\}.$$

This means $A \triangle (A \oplus 2) = \{1, 2, \ldots, k\}$ and the maximum is $k \in A \oplus 2$. Adding a unit again gives $A \oplus 3 = \{1, k, a_{k+1}, a_{k+2}, \ldots, a_n\}$, which then implies the symmetric difference is $A \triangle (A \oplus 3) = \{0, 2, 3, \ldots, k\}$ with maximum in $A \oplus 3$. Then, $A \oplus 4 = \{0, 1, k, a_{k+1}, a_{k+2}, \ldots, a_n\}$ and symmetric difference $A \triangle (A \oplus 4) = \{2, 3, 4, \ldots, k\}$. Continue in this manner, applying $\oplus 1$, until it has been applied a total of $2^k - 1$ times. In each step, the set A will be smaller than. Thus far, it has been proven $\max(A \triangle B) \in B$ if $A < B < A \oplus 2^k$. Applying $\oplus 1$ once more, is simply adding the singleton $2^k = \{k\}$ to the set A. The result is $A \oplus 2^k = \{0, 1, \ldots, k, a_{k+1}, a_{k+2}, \ldots, a_n\}$ because k is the smallest object not in A. This implies $\max(A \triangle (A \oplus 2^k)) = \max\{k\} = k \in A \oplus 2^k$. It is concluded $\max(A \triangle B) \in B$ if $A < B \leq A \oplus 2^k$. The careful reader will notice the elevator argument, or an induction hypothesis is needed to justify this argument. The rest is a repetition of what has been done up to this point. To apply $\oplus 1$ to $A \oplus 2^k$, simply substitute all the elements $0, 1, \ldots, k$ with $k+1$; use $2^{k+1} = 1 + (1 + 2 + 4 + 8 + \cdots + 2^k)$. There are two cases; $k+1 \notin A$ or $k+1 \in A$. In the first case, $\max(A \triangle (A \oplus 2^k \oplus 1)) = k+1 \in A \oplus 2^k \oplus 1$ because $A \oplus 2^k \oplus 1 = \{k+1, a_{k+1}, a_{k+2}, \ldots, a_n\}$. In the second case, $a_{k+1} = k+1$ so that

$$A \oplus 2^k \oplus 1 = \{k+1, a_{k+2}, \ldots, a_n\} \oplus \{k+1\}.$$

Proceed as before, finding the second smallest number not in A. Let $p \in A$ the smallest number in $A-\{k\}$. The numbers k, p are the two smallest numbers not in A, so that $A = \{0, 1, \ldots, k-1, k+1, k+2, \ldots, p-1, a_p, \ldots, a_n\}$. This implies $(A \oplus 2^k) \oplus 1 = \{p, a_p, \ldots, a_n\}$. The symmetric difference with A is $\{0, 1, \ldots, k-1, k+1, \ldots, p\}$. The maximum of the symmetric difference is $p \in (A \oplus 2^k) \oplus 1$. This proves $\max(A \triangle B) \in B$ if $A < B \leq (A \oplus 2^k) \oplus 1$. The symmetric difference of $(A \oplus 2^k \oplus 1) \oplus 1 = \{0, p, a_p, \ldots, a_n\}$ with A, is $\{1, 2, \ldots, k-1, k+1, k+2, \ldots, p-1, p\}$. The maximum of the symmetric difference is $p \in (A \oplus 2^k) \oplus 2$. This proves $\max(A \triangle B) \in B$, if $A < B \leq (A \oplus 2^k) \oplus 2$. Then, $(A \oplus 2^k) \oplus 3 = \{1, p, a_p, \ldots, a_n\}$, which again gives $p = \max(A \triangle (A \oplus 2^k \oplus 3)) \in A \oplus 2^k \oplus 3$. Continue in this manner. Apply $\oplus 1$ to $A \oplus 2^k$ a total of $2^k - 1$ times before reaching

$$(A \oplus 2^k) \oplus 2^k = \{0, 1, \ldots, k-1, p, a_p, \ldots, a_n\}.$$

Here, symmetric difference is $A \triangle ((A \oplus 2^k) \oplus 2^k) = \{k+1, k+2, \ldots, p\}$, and the maximum is $p \in (A \oplus 2^k) \oplus 2^k$. This proves $\max(A \triangle B) \in B$, if $A < B \leq A \oplus 2^k \oplus 2^k$. Adding 1 again, gives $A \oplus 2^k \oplus 2^k \oplus 1 = \{k, p, a_p, \ldots, a_n\}$. The symmetric difference with A is the set $\{k, p\}$. The maximum is $\max\{k, p\} = p \in A \oplus 2^k \oplus 2^k \oplus 1$. Keep adding 1 until $(A \oplus 2^k) \oplus 2^p = \{0, 1, \ldots, q-1, a_{n-q+1}, a_{n-q+2}, \ldots, a_n\}$ has been reached, where $A = \{0, 1, \ldots, k-1, k+1, \ldots, p-1, p+1, \ldots, q-1, a_{q-2}, a_{q-1}, \ldots, a_n\}$ and $q > p$ is the third smallest number not in A. This continues, for all $k, p, q, \ldots, r$ not in A. This proves $\max(A \triangle B) \in B$ if $A < B < A \oplus 2^k \oplus 2^p \cdots \oplus 2^r$. Upon adding 1 to $A \oplus 2^k \oplus 2^p \oplus \cdots \oplus 2^r = \{0, 1, \ldots, a_n\}$, the result is the singleton $\{a_n + 1\}$. It is trivial to prove that the maximum of the symmetric difference is in $A \oplus 2^k \oplus 2^p \cdots \oplus 2^r \oplus 1$, since $\max(A) = \{a_n\} < \{a_n + 1\} = \max(A \oplus 2^k \oplus 2^p \cdots \oplus 2^r \oplus 1)$. Observe that either $\max(X \oplus 1) = \max(X)$ or $\max(X \oplus 1) = \max(X) + 1$. Therefore, the result also holds for any $B > A \oplus 2^k \oplus 2^p \cdots \oplus 2^r \oplus 1$ because

$$\max(B) \geq \max(A \oplus 2^k \oplus 2^p \oplus \cdots \oplus 2^r \oplus 1) > \max(A)$$

.

To prove the second implication, use the following observation. Let $A = \{a_1, a_2, \ldots, a_n\}$ any set number, and let $b \notin A$ the maximum $b = \max(A \triangle B)$. Add 1 to the set number A repeatedly until you get to the set number $R = \{b, a_i 1, a_i 2, \ldots, a_n\}$, where $\{a_i 1, a_i 2, \ldots, a_n\}$ are the elements of A that are greater than b. This means a set N exists such that $R = A \oplus N$. Now, add P to R, where $P = \{b_1, b_2, \ldots, b_j\}$ is the set of objects in B that are smaller than b. The result is $B = P \oplus R = P \oplus (A \oplus N) = A \oplus (N \oplus P)$ which implies $A < B$. □

Let $A = \{2, 5, 6, 8, 9\}$ and $B = \{0, 1, 7, 8, 9\}$. The largest of the two is the set that contains $\max\{0, 1, 2, 5, 6, 7\} = 7$, so that $A < B$. In the next sections the order of set numbers will be given in a specific form. For example, a set number may be written in the form $A = \{\{3, 5\}, \{1, 2\}, \{4, 6\}\} = 2^{2^{2^3+2^5}+2^{2^1+2^2}+2^{2^4+2^6}}$. Compare it with $B = \{\{3, 4\}, \{1, 2\}, \{5, 6\}\} = 2^{2^{2^3+2^4}+2^{2^1+2^2}+2^{2^5+2^6}}$. The order relation is $A < B$ because $\max(A) = \{4, 6\} < \{5, 6\} = \max(B)$.

The operation function $\oplus 1$, of Definition 6, generates all **HFS** when applied successively to 0. The order in which sets are generated is an order of **HFS**, equivalent to the order of natural numbers $\mathbb{N}_{\leq}$. The operation function $\oplus n = \oplus 1^n$ is used to define addition of sets $\oplus 1^n(m) = m \oplus n = (m \triangle n) \oplus s(m \cap n)$.

2.3 Product of Set Numbers

The product is easy to define. Multiplication by 2 has already been defined. In binary representation $2^n + 2^n = 2^{n+1}$, and set numbers have a corresponding rule. To multiply by 2 is to apply the function $\odot 2 = s$ that adds 1 to the elements of the argument. Multiplication by 4 is $s \circ s$ which adds 2 to the elements of the argument. In general, multiplication of B by 2^k is equal to $s^k(B)$. If $B = \{b_1, \ldots, b_n\}$ then $2^k \odot B$ is equal to the set $\{b \oplus k\}_{b \in B} = \{b_1 \oplus k, \ldots, b_n \oplus k\}$. The product of a set number B with 2^k, in our graphic representation, consists of displacing the objects of the set, k units up. The set number $2^k \odot B$ is the *k-displacement of B*. The general product $A \odot B$ is defined in terms of displacements of the *base* B, and the *pivot* A.

$$A \odot B = \bigoplus_{a \in A} \{b \oplus a\}_{b \in B}. \tag{4}$$

Displacements of B are added, one for each object of the pivot A. If $a \in A$ then the a-displacement of B is one of the displacements in our sum. Notice that multiplication by 0 results in the empty set, $0 \odot X = X \odot 0 = 0$.

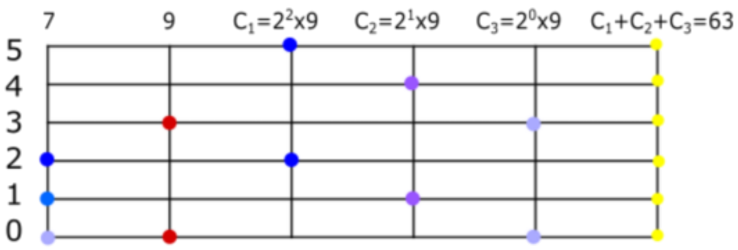

Figure 2: The product $7 \odot 9$. The first and second columns are the pivot and base, respectively. The next three columns correspond to the displacements of the base. The last column is the sum of the displacements. The result is equal to $63 = \{0, 1, 2, 3, 4, 5\}$.

It is also trivial to find $1 \odot X = X \odot 1 = X$. To show that $2 = \{1\}$ is commutative under multiplication,

$$\begin{aligned} \{1\} \odot X &= \{x \oplus 1\}_{x \in X} \\ &= \bigcup_{x \in X} \{x \oplus 1\} \\ &= \bigoplus_{x \in X} \{1 \oplus x\} \\ &= X \odot \{1\}. \end{aligned}$$

This means $2 \odot X = X \odot 2 = X \oplus X$. To find the product $7 \cdot 5 = (2^0 + 2^1 + 2^2)(2^0 + 2^2)$ use distribution to obtain $2^0(2^0 + 2^2) + 2^1(2^0 + 2^2) + 2^2(2^0 + 2^2)$. Then, $(2^{0+0} + 2^{2+0}) + (2^{0+1} + 2^{2+1}) + (2^{0+2} + 2^{2+2}) = (2^0 + 2^2) + (2^1 + 2^3) + (2^2 + 2^4)$. Carrying out the addition gives $7 \cdot 5 = 2^0 + 2^1 + 2^2 + 2^5 = 35$. Before proving general properties, calculate $5 \odot 15 = \{0, 2\} \odot \{0, 1, 2, 3\}$ in two different ways to verify these numbers commute. First make $A = 5$ and $B = 15$. Two displacements of $B = \{0, 1, 2, 3\}$ will be added. The first displacement is $\{0 \oplus 0, 1 \oplus 0, 2 \oplus 0, 3 \oplus 0\} = \{0, 1, 2, 3\}$, and the second displacement is $\{0 \oplus 2, 1 \oplus 2, 2 \oplus 2, 3 \oplus 2\} = \{2, 3, 4, 5\}$. Adding the two, gives $\{0, 1, 2, 3\} \oplus \{2, 3, 4, 5\} = \{0, 1, 3, 6\} = 75$. Now, make $A = 15$ and $B = 5$. Four displacements of $5 = \{0, 2\}$, each corresponding to an element of $15 = \{0, 1, 2, 3\}$. The displacements of 5 are $\{0, 2\}, \{1, 3\}, \{2, 4\}, \{3, 5\}$. Adding these four displacements results in $(\{0, 2\} \oplus \{1, 3\}) \oplus (\{2, 4\} \oplus \{3, 5\}) = \{0, 1, 2, 3\} \oplus \{2, 3, 4, 5\} = 75$, using associativity of addition.

Figure 2 shows the graphic representation of $7 \odot 9$. To formalize this, first verify $\odot 2$ is a morphism for addition of set numbers; verify $s(A \oplus B) = s(A) \oplus s(B)$. Use $X \oplus X = s(X)$ to prove $s(A \oplus B) = (A \oplus B) \oplus (A \oplus B) = (A \oplus A) \oplus (B \oplus B) = s(A) \oplus s(B)$. This implies

$$s^k(A \oplus B) = s^k(A) \oplus s^k(B), \tag{5}$$

for every $k \in \mathbb{N}$. To prove the distributive property use (5) and the commutative and associative properties of addition of sets.

$$\begin{aligned} A \odot (B \oplus C) &= \bigoplus_{a \in A} \{x \oplus a\}_{x \in B \oplus C} \\ &= \bigoplus_{a \in A} s^a(B \oplus C) \\ &= \bigoplus_{a \in A} (s^a(B) \oplus s^a(C)) \\ &= \bigoplus_{a \in A} s^a(B) \oplus \bigoplus_{a \in A} s^a(C) \\ &= (A \odot B) \oplus (A \odot C) \end{aligned}$$

To prove multiplication is commutative, let $a \in A$ fixed. The set $\{b \oplus a\}_{b \in B} = \{b_1 \oplus a, b_2 \oplus a, \ldots, b_n \oplus a\} = \{b_1 \oplus a\} \oplus \{b_2 \oplus a\} \oplus \ldots \oplus \{b_n \oplus a\}$ can be expressed as a sum of disjoint singletons, $\bigoplus_{b \in B} \{b \oplus a\}$. Therefore,

$$\begin{aligned} A \odot B &= \bigoplus_{a\in A}\{b \oplus a\}_{b\in B} \\ &= \bigoplus_{a\in A}\bigoplus_{b\in B}\{b \oplus a\} \\ &= \bigoplus_{b\in B}\bigoplus_{a\in A}\{a \oplus b\} \\ &= \bigoplus_{b\in B}\{a \oplus b\}_{a\in A} \\ &= B \odot A. \end{aligned}$$

The commutative property of addition and multiplication of sets has been proven. Together with the distributive property, these imply

$$(A \oplus B) \odot C = (A \odot C) \oplus (B \odot C). \tag{6}$$

Now it can be proven that the associative property holds for the product of set numbers. Because of Proposition 3, it is sufficient to verify the operation functions of $\odot$ commute. This can easily be done using mathematical induction. It is trivial to verify $\odot A$ and $\odot B$ commute for $N = 1$; it follows from the commutative property $\odot A \odot B(1) = A \odot B = b \odot A = \odot B \odot A(1)$. Suppose $\odot A$ and $\odot B$ commute for arbitrary N, so that $A \odot (B \odot N) = B \odot (A \odot N)$.

$$\begin{aligned} (\odot A \circ \odot B)(N \oplus 1) &= A \odot (B \odot (N \oplus 1)) \\ &= A \odot (B \odot N \oplus B \odot 1) \\ &= A \odot (B \odot N) \oplus A \odot B \\ &= B \odot (A \odot N) \oplus B \odot A \\ &= B \odot (A \odot N \oplus A \odot 1) \\ &= B \odot (A \odot (N \oplus 1)) \\ &= (\odot B \circ \odot A)(N \oplus 1) \end{aligned}$$

This proves associativity of multiplication. The next result characterizes multiplication as a repeated addition.

Proposition 5. *The operation function $\odot N$ acts on sets by $\odot N(X) = \oplus X^N(0)$.*

Proof. This is proven by mathematical induction on N. It is true for $N = 1$, since $1 \odot X = X$. Suppose it is true for N, then using the distributive and commutative properties

$$\begin{aligned} \odot(N \oplus 1)(X) &= \odot N(X) \oplus \odot 1(X) \\ &= \oplus X^N(0) \oplus X \\ &= \oplus X(\oplus X^N(0)) \\ &= \oplus X^{N+1}(0). \end{aligned}$$

□

2.4 Multiplication Array

It has been seen that multiplication is equivalent to the addition of multiple displacements of a given set number, or as the repeated addition of the same number. In ether case, it is the addition of multiple operands. The sub unit responsible for adding the partial products is usually referred to as accumulator of partial products. Some of the current proposals for multiplication are found in [Abrar(2019)], [Emmart(2011)], and [Taib(2020)]. A general method for defining the sum of multiple operands is proposed that has several advantages in hardware implementation. An algorithm is described that reduces the sum of 2^k summands to the sum of $k+1$ summands.

In general, it reduces the sum of n summands to $\max(n)+1$ summands. Consider the sum of 4-many, 8-bit numbers. The summands are $A = a_0a_1\cdots a_7$, $B = b_0b_1\cdots b_7$, $C = c_0c_1\cdots c_7$, $D = d_0d_1\cdots d_7$. This can be represented by the array

$$\begin{array}{cccc} a_7 & b_7 & c_7 & d_7 \\ a_6 & b_6 & c_6 & d_6 \\ a_5 & b_5 & c_5 & d_5 \\ a_4 & b_4 & c_4 & d_4 \\ a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0, \end{array}$$

where each a_i, b_i, c_i, d_i is either 0 or 1. Count the number of 1's in each row. It takes three bits to write in binary form the number of 1's in a single row because $\max(4)+1 = 2+1 = 3$. The number of 1's in each row can be represented in a 8×3 grid,

$$\begin{array}{ccc} a_7' & b_7' & c_7' \\ a_6' & b_6' & c_6' \\ a_5' & b_5' & c_5' \\ a_4' & b_4' & c_4' \\ a_3' & b_3' & c_3' \\ a_2' & b_2' & c_2' \\ a_1' & b_1' & 0 \\ a_0' & 0 & 0 \end{array} \tag{7}$$

The elements a_0', b_1', c_2' will be used to write the number of 1's in row 0. The elements a_1', b_2', c_3' are used to write the number of 1's in row 1, and elements a_2', b_3', c_4' are used to write the number of 1's in row 2, etc. This maintains the representation of energy-levels and their unit value, while avoiding any intervention with totals from one row and the another. The three column grid can be reduced to two columns, by iterating the process. The total number of units in a single row of (7) will be represented with two bits because $\max(3)+1 = 1+1 = 2$. Addition of the two columns

$$\begin{array}{cc} a_7'' & b_7'' \\ a_6'' & b_6'' \\ a_5'' & b_5'' \\ a_4'' & b_4'' \\ a_3'' & b_3'' \\ a_2'' & b_2'' \\ a_1'' & b_1'' \\ a_0'' & 0 \end{array}$$

is equivalent to the original four-input addition. Elements a_0'' and b_1'' represent the total value of the first row in (7). Elements a_1'' and b_2'' represent the total value of the second row, elements a_2'' and b_3'' represent the total value of the third row, etc.

An example is provided, to find the total of $A = 63 = \{0,1,2,3,4,5\}, B = 37 = \{0,2,5\}, C = 21 = \{0,2,4\}, D = 38 = \{1,2,5\}, E = 28 = \{2,3,4\}, F = 13 = \{0,2,3\}, G = 14 = \{1,2,3\}, H = 52 = \{2,4,5\}$. This is given by

$$\begin{array}{cccccccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0. \end{array}$$

Since there can be at most eight objects in each row, only four bits are needed per row. This means the new grid has four columns. There is a total of $4 = \{2\}$ many number 1's in row 0. This is represented by placing the

sequence of digits 0010 in the bottom most diagonal, of the new grid.

$$\begin{array}{cccc} & & & 0 \\ & & 1 & 0 \\ & 0 & 0 & 0 \\ 0 & 0 & 0 & 0. \end{array}$$

Next, there is a total of $3 = \{0, 1\}$ number 1's in row 1. This is represented by placing the sequence of digits 1100 in the next diagonal.

$$\begin{array}{cccc} & & & 0 \\ & & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0. \end{array}$$

Since there are $8 = \{3\}$ number 1's in row 2, the sequence 0001 is placed in the next diagonal.

$$\begin{array}{cccc} & & & 1 \\ & & 0 & 0 \\ & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0. \end{array}$$

Rows 3,4, and 5 have $4 = \{2\}$ number 1's each so that the sequence 0010 is placed in each of the following diagonals.

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0. \end{array}$$

The sum of these four columns can now be reduced to the sum of three columns because three bits are enough for representing a total of four objects per row. Row 0, of the last grid, has a total of 0 number 1's so that the sequence 000 is placed on the bottom diagonal.

$$\begin{array}{ccc} & & 0 \\ & 0 & 0 \\ 0 & 0 & 0. \end{array}$$

There is a total of $1 = \{0\}$ number 1's in Row 1 so that the sequence 100 is placed on the next diagonal.

$$\begin{array}{ccc} & & 0 \\ & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0. \end{array}$$

Continuing in this manner gives

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0. \end{matrix}$$

The addition of three columns is reduced to $\max(3) + 1 = 2$ columns,

$$\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0. \end{matrix}$$

Applying addition of two columns gives $A \oplus B \oplus \cdots \oplus H = 266$,

$$\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0. \end{matrix}$$

The sum of n-many b-bit numbers can be computed with nb many nodes organized in a rectangular grid of size $n \times b$. This grid can be used to add n-many b-bit numbers, and the multiplication algorithm can also be executed. Parallel connections (only nodes from the same row or column are connected) allow for b-many n-bit SLFAs to perform addition of n-many b-bit numbers. The same low-powered circuit also performs parallel-multiplication of two inputs. This modified SLFA, its connections, extensions for vector and matrix multiplication along with case by case analysis and implementation proposals will be described in a separate paper, exclusively found in the author's personal web page "www.binaryprojx.com". To add n-many b-bit numbers, b-many n-bit SLFAs are placed side-by-side in a rectangular grid. Each node has two bits of memory belonging to the SLFA, and requires a third bit of memory that will be referred to as the principal bit. This means each node will consist of a three bit register and a Half Adder, maintaining the gate count and depth very low. The rectangular array requires parallel connections between nodes of the same row or column. The principal bits of the nodes are used for storing the initial inputs. Each SLFA will count the number of 1's in its row; one by one, send signals of the principal bits of that row to the SLFA. This is done in parallel so that all rows are counted at once and they

can each signal process termination individually. Once the elements of a row are counted, the results stored in the SLFA will be sent to their new principal bits. In each iteration the columns with non-zero entries in their principal bits are less. This process is iterated until only two columns are non-zero in the principal bits. Given that this circuit performs addition of multiple inputs, it is also capable of carrying out multiplication of two binary inputs. Additionally, the circuit is able to perform parallel addition of b-many pairs of n-bit numbers, because each SLFA (rows) can be used as an independently-timed SLFA. The circuit is linearly scaleable in terms of bits and inputs, it presents the minimum possible topological complexity (rectangular grid of nodes with parallel connections), and is low-powered due to the gate depth. First, the number of 1's in each row is counted. This is done by sending signals from the principal bits into the the SLFA. When the i-th significant bit of a row is sent to the SLFA, the addition that follows takes at most $\max(i) + 1$ iterations of the SLFA, because the set number i requires $\max(i) + 1$ many digits to express in binary form. The worst case scenario, when adding n inputs, occurs if a row has n-many $1's$ in the principal bits. The number of steps in the worst case scenario is bounded by $\max(2) + \max(3) + \max(4) + \ldots + \max(n) + n$, but is much lower because most of the terms can be bounded by smaller numbers. For example, if i is a multiple of 2, then only one iteration of the SLFA is needed in that step. The only occasions where $\max(i) + 1$ iterations of the SLFA are needed is if i is of the form $2^k - 1$, for some k. Although PASTA adders [Rahman(2013)] are topologically equivalent to the SLFA, it is important to note the PASTA adder is an asynchronous circuit, like most fast-adders [Franklin(1994)], and therefore it lacks the memory units necessary for this addition of multiple inputs. The PASTA adder has multiplexers instead of the registers used in the SLFA, making it inappropriate for implementation in this multi-operand arithmetic architecture.

The relative efficiency of this implementation with respect to other circuits could be done by cases, in terms of the quotient of n and b. For large b and small n, it is easy to see the advantages this circuit would have because it calculates the total number of elements in a row, and it does all rows in parallel. The more rows there are relative to columns the more advantage provided by this method. There is a problem when n gets too big. When the i-th bit of a given row is sent to the SLFA for counting, the SLFA performs $\max(i) + 1$ many iterations. If n is too large this method will present diminishing returns, as i approaches n. However, for this case there is an alternative. The number of summands can be reduced by half in a fixed number of steps. The method reduces the addition of n summands to the sum of $\max(k) + 1$. Thus, 8 summands can be reduced to $\max(8) + 1 = 4$ summands. If the number of summands is a multiple of 8, then this fact can be used to reduce by half the number of summands. There is another way to reduce summands by half because 4 summands are reduced to $\max(4) + 1 = 3$ summands which in turn are reduced to $\max(2) + 1 = 2$ summands. This means that if the number of summands is a multiple of 4, then the number of summands can be reduced to half in this manner. These alternate methods of reduction into half are achieved by rearranging the vertical and horizontal connections of the grid. Depending on the quotient and size of n and b there will be an optimal size for reduction of summands that minimizes time complexity, and the topology of the nodes is unchanged.

There are several benefits in using this architecture for matrix multiplication. Examples of current solutions to matrix multiplication are proposed and referenced in [Zhang(2013)]. In the case that $n = b$ there are advantages to be exploited for matrix multiplication. Suppose you wish to multiply two $n \times n$ matrices and the entries of the matrices are $n/2$-bit numbers. Storing the two matrices requires $2n^3$ memory units in a rectangular arrangement. A logic grid of $2n^3$ nodes with the same rectangular form can be super-positioned on top of the memory elements. This allows calculation of the dot product of one row and one column in the time it takes to multiply two $n/2$-bit numbers, plus the time it takes to add n-many n-bit numbers. In matrix multiplication it can often be the case that the number b of bits of the entries, is smaller than the numbers of rows and columns. Adaptations can be made for these cases also, based on area-specific use and pipeline needs. This rectangular design of subunits and parallel connections solves some of the basic problems with In-Situ computing [Wang(2023)]. Given the fact that memory is hardwired in rectangular grids, but the logic for computing has many complicated patterns and irregular connections, it is difficult to reconcile both designs in the same space. Von Neumann Arhitecture considers memory and the ALU to be two separate parts of a CPU, at the cost of having to transfer data back and forth between memory and logic units. In this proposal, the memory units can be placed on a bottom layer, and the logic circuitry can be placed on a top layer. This superposition of two rectangular grids of equal size provides a solution to some the problems related with Computing-In-Memory. The delay and energy saved by this architecture merits further investigation and comparison [Hennessy(1990)] to other architectures.

2.5 Power as a Generalization of Product

It is easy to see how the relations between the operations of addition and multiplication can be generalized. The composition powers of $\oplus 1$ are the functions $\oplus n$ given by $\oplus n(x) = \oplus 1^n(x)$. The composition powers of $\oplus x$ are the functions $\odot n$ defined by $\odot n(x) = (\oplus x)^n(0)$. The power function will be defined similarly in terms of operation functions $*n$ such that $x^n = *n(x)$. The function $*n$ is defined by $*n(x) = (\odot x)^n(1)$. In [Ramirez(2019)], there is a description of subtraction, division and powers of set numbers. Here, an alternative definition is given for multiplication and powers. Recall that the multiplication of two sets is given by adding all the sets of the form $\{a \oplus b\}$, where $a \in A$ and $b \in B$. Given any $a \in A$ and $b \in B$, consider the function $f : \{0, 1\} \to (A \cup B)$ such that $f(0) = a$ and $f(1) = b$. Then it is true that

$$A \odot B = \bigoplus_{f:\{0,1\}\to(A\cup B)} \{f(0) + f(1)\},$$

where the index f of the sum is taken over every possible function $f : \{0, 1\} \to (A \cup B)$ such that $f(0) \in A$ and $f(1) \in B$. Recall that the set of numbers smaller than a fixed number is $2^n - 1 = \{0, 1, 2, \ldots, n - 1\}$. The generalized product $(A_1 \odot A_2 \odot \cdots \odot A_n)$ can be written as

$$\bigodot_{i\in 2^n-1} A_i = \bigoplus_{f:2^n-1\to A} \left\{ \bigoplus_{i\in 2^n-1} f(i) \right\},$$

where $A = \bigcup_{i\in 2^n-1} A_i$ and the index f is taken over every function such that $f(i) \in A_i$. It is the addition of singletons, and each singleton is the sum of all the objects in the image of some function f of the index. The commutative and associative properties are trivial to prove from this definition. Changing the order for the multiplication of the sets only changes the order in an addition of sets. This equality gives the expected particular cases. It is easy to see that if $A_k = 0$, for some k, the product is 0. This is true because the sum over the index f is empty; there is no function f such that $f(k) \in A_k$. Furthermore, if all the $A_i = X$ are equal to the same number, the result is X^n.

$$X^n = \bigoplus_{f:2^n-1\to X} \left\{ \bigoplus_{i\in 2^n-1} f(i) \right\}.$$

This expression can easily be verified to satisfy the particular cases. For example, if $n = 1$, then $2^n - 1 = 1 = \{0\}$. What are all the functions of the form $f : \{0\} \to X$? Obviously they are the functions of one component, that select the objects of X. Listing them is easy. For every object $x \in X$, the function f_x defined by $f_x(0) = x$ is considered. The sum of the objects in the image is x, for every function f_x. Adding all the sets corresponding to the addition of the image, taken over every function, gives $X = \bigoplus_{x\in X}\{x\}$. It is easy to see that if $X = 1 = \{0\}$ then there is exactly one function $f : 2^n - 1 \to X$, and it is trivially defined by $f(i) = 0$ for every $i \in 2^n - 1$. This means the result is $1^n = \{0\}$. For $X = 2\{1\}$, again there is exactly one function but this time $f(i) = 1$ for every $i \in 2^n - 1$. Therefore, $2^n = \{n\}$. Up until now, 2^n was just a symbol for denoting the set number $\{n\}$, but now it has acquired its traditional meaning. Now, to define X^0 observe two things. The first is that $2^0 = 1$, and the second is that X^0 is undefined with this definition. The number $2^0 - 1 = 0 = \emptyset$ is the empty set so that there are no functions $f : \emptyset \to X$. Therefore it is justified to define $X^0 = 1$.

2.6 Integers

The structure of integers is not necessary to construct the structure of real numbers. However, a construction of $\mathbb{Z}$ is provided because it introduces methods and concepts of previous and later sections. Operation functions and their inverse functions are used to describe integers. A positive integer $\mathbf{n} \in \mathbb{Z}$ is an operation function $\oplus n$, while its negative integer **-n** $\in \mathbb{Z}$ is the inverse function $(\oplus n)^{-1}$. Notice one important fact. Negative integers can easily be distinguished from positive integers. A negative integer is a function of the form **-n** $: \{n, n\oplus 1, n\oplus 2, \ldots\} \to \mathbb{N}$, while a positive integer is a function of the form $\mathbf{n} : \mathbb{N} \to \{n, n \oplus 1, n \oplus 2, \ldots\}$. This will have to be considered when defining addition of integers; it does not represent any difficulty but the reader must be careful. The integer **0** is the identity function of $\mathbb{N}$. The set of negative integers will be represented with the symbol $-\mathbb{N}$. It will be said that $X \subset \mathbb{Z}$ is a *non negative subset of* $\mathbb{Z}$ if $-\mathbb{N} \cap X = \emptyset$, and the like.

The sum of integers is defined in the obvious way, using composition. Let $\mathbf{m} = \oplus m$ and $\mathbf{n} = \oplus n$ positive integers. The composition of these is a positive integer. Define the addition of two positive integers by the relation $\mathbf{m+n} = \oplus m \circ \oplus n$. The sum, **-m-n**, of negative integers $\mathbf{-m} = (\oplus m)^{-1}$ and $\mathbf{-n} = (\oplus n)^{-1}$, is defined as the composition of inverse functions $(\oplus m)^{-1} \circ (\oplus n)^{-1} = (\oplus n \circ \oplus m)^{-1}$. Given commutativity $\oplus n \circ \oplus m = \oplus m \circ \oplus n$, it follows that **-m-n** is equal to the negative integer $(\oplus m \circ \oplus n)^{-1} = \mathbf{-(m+n)}$. The sum of one negative integer **-m** and one positive integer **n** is defined as follows. There are two possible cases. If the corresponding natural numbers satisfy $m < n$, there is a natural number x such that $n = m + x$. Define $\mathbf{-m+n} = \mathbf{x}$, where $\mathbf{x} = \oplus x : \mathbb{N} \to \{n - m, n - m + 1, n - m + 2, \ldots\}$. In the contrary case that the natural numbers satisfy $n < m$, then $m = n + x$ for some natural number x. Define addition of these integers by $\mathbf{-m+n} = \mathbf{-x}$, where $\mathbf{-x} = (\oplus x)^{-1} : \{m - n, m - n + 1, m - n + 2, \ldots\} \to \mathbb{N}$. The order relation between m, n determines if **-m+n** is a positive integer or a negative integer. In both cases, the relation $\mathbf{-m+n} = (\oplus m)^{-1} \circ \oplus n$ holds. But, how is **n-m** defined? Consider the composition $\oplus n \circ (\oplus m)^{-1}$. In both cases, $m < n$ or $n < m$, the composition is $\oplus n \circ (\oplus m)^{-1} : \{m, m + 1, \ldots\} \to \{n, n + 1, \ldots\}$. Although $\oplus n \circ (\oplus m)^{-1}$ is a well defined composition, it is not an integer. The functions $\oplus n \circ (\oplus m)^{-1}$ and $(\oplus m)^{-1} \circ \oplus n$ are not the same function. However, in the intersection of the domains, these compositions are equal functions. Thus, defining the sum of integers as commutative, $\mathbf{n-m} = \mathbf{-m+n}$, is justified. To prove addition of integers is associative, let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ integers. Eight different cases have to be proven. The different combinations for x, y, z being positive or negative. Suppose first, y is positive. Then, $\mathbf{x+y} = \oplus x \circ \oplus y$. Consider two sub cases. If z is positive, the associative property holds for $\mathbf{(x+y)+z} = \mathbf{x+(y+z)}$ because the associative property holds for composition of functions. Suppose z is negative. Then $\mathbf{(x+y)+z} = \mathbf{z+(x+y)} = \mathbf{z+(y+x)} = \mathbf{(z+y)+x} = \mathbf{x+(z+y)} = \mathbf{x+(y+z)}$. Going back to the assumption of y, now suppose y is negative and x is positive. The equality $\mathbf{(x+y)+z} = \mathbf{(y+x)+z} = \mathbf{y+(x+z)} = \mathbf{y+(z+x)} = \mathbf{(y+z)+x} = \mathbf{x+(y+z)}$ holds. If $\mathbf{x}$ and $\mathbf{y}$ are negative, then $\mathbf{x+y} = \oplus x \circ \oplus y$. This implies $\mathbf{(x+y)+z} = (\oplus x \circ \oplus y) \circ \oplus z = \oplus x \circ (\oplus y \circ \oplus z) = \mathbf{x+(y+z)}$. This proves addition of integers is associative. The addition of integers **5-3** is equal to the function $\oplus 2$, while the result of **3-5** is $(\oplus 2)^{-1}$.

Ordering integers is natural, in this context. Two integers $\mathbf{x}, \mathbf{y}$ satisfy the inequality $\mathbf{x} < \mathbf{y}$ if and only if $\mathbf{x}(n) < \mathbf{y}(n)$, for any $n \in \mathbb{N}$. For example, $\mathbf{-5} < \mathbf{2}$ because $\mathbf{-5}(5) = 0 < 7 = \mathbf{2}(5)$. Of course, the order is well defined so that there is no natural number n such that $\mathbf{2}(n) < \mathbf{-5}(n)$. To prove $\mathbf{-6} < \mathbf{-3}$ a set number in the domain of **-3** and **-6** is chosen. Say, the number 6. Then, $\mathbf{-6}(6) = 0 < 3 = \mathbf{-3}(6)$.

3 Finite Functions and Permutations

In this section, the set of finite functions on $\mathbb{N}$ is described and an injective function from this set of functions, into the set of natural numbers is defined. The important quality of this representation is that functions are equivalent if and only if they are represented by the same number. Natural numbers are assigned to abstract functions, also. There will be a distinct difference when working with an abstract function or a concrete function. When working with abstract functions, two functions are defined to have the same *structure* if they are assigned the same natural number. Concrete functions, on the other hand, can have the same structure but different numeric representation. For example, consider the functions f, g defined by

$$\begin{array}{ll} f(a) = b & g(a) = a \\ f(b) = a & g(b) = c \\ f(c) = c & g(c) = b. \end{array}$$

These two abstract functions have the same structure, and have the same numeric representation. They will be considered to be the same function. However, if the objects are not abstract, so that $a, b, c \in \mathbb{N}$ take specific values, then f, g are distinct concrete functions and they will be represented by distinct numbers. In the previous example, let $a = 3$, $b = 5$ and $c = 0$. The functions f, g defined by $f(3) = 5$, $f(5) = 3$, $f(0) = 0$, and $g(3) = 3$, $g(5) = 0$, $g(0) = 5$ are different and they will be represented by different numbers.

The set of finite functions of natural numbers is linearly ordered. Then, an equivalence definition for abstract finite functions is given and they are also ordered linearly. A canonical order for the elements of a given abstract finite function is also provided, and it is determined which objects of the function are equivalent. In the example, the objects a, b, are equivalent in the function f, while c is not equivalent to another object. The objects b, c are equivalent in the function g, and a is not equivalent to another object.

3.1 Ordered Pairs

To represent finite functions as natural numbers, first it is necessary to find a way of representing ordered pairs as natural numbers. An ordered pair of natural numbers should be an object (m, n) from which two natural numbers are represented in a predetermined order. This means that (m, n) and (n, m) should not be the same object. The first ordered pair, (m, n), represents two natural numbers in order; first m, then n. The second ordered pair (n, m) means first n, then m. A set of two natural numbers $\{X, Y\}$ is not an ordered pair because it is a set of two objects without a predetermined order; a collection of objects X, Y and they are not ordered.

To solve this, a method of coding an ordered pair of natural numbers using odd/even numbers to represent the first/second component, respectively, is outlined. An odd set number is a set number A with $0 \in A$. An even set number is a set number B such that $0 \notin B$. Any set number X, has associated to it the odd number $s(X) \oplus 1$, and the even number $s(X \oplus 1)$. For example, 0 is associated to the odd number $s(0) \oplus 1 = 1$ and the even number $s(0 \oplus 1) = 2$. The number 1 is associated the odd number $s(1) \oplus 1 = 2 \oplus 1 = 3$ and the even number $s(1 \oplus 1) = s(2) = 4$. In general, the natural number k has odd and even representations $2k + 1$ and $2(k + 1)$, respectively, as shown in Table 1.

X	ODD	EVEN
0	1	2
1	3	4
2	5	6
3	7	8
4	9	10
5	11	12
⋮	⋮	⋮

Table 1: Every natural number is uniquely associated an odd and even number.

To find the odd representation of the set number X, displace the objects of X one unit up, then add the object 0 to $s(X)$ to obtain $s(X) \oplus 1 = s(X) \cup \{0\}$. The even representation, $s(X \oplus 1)$, is obtained by displacing the elements of $x \oplus 1$ one unit up. The ordered pair (m, n) is a set number of one odd and one even number, $\{2m + 1, 2(n + 1)\}$. This allows to differentiate the two components. The odd number represents the first component, while the even number is used for the second component. The set number $P = \{2m + 1, 2(n + 1)\}$ is the ordered pair (m, n). The set number $P = \{2m + 1, 2(n + 1)\}$ represents the ordered pair (m, n). The set number representing $(0, 0)$ is $\{1, 2\} = 2^{2(0)+1} + 2^{2(0+1)} = 6$. The ordered pair $(4, 5)$ is represented by the natural number $2^{2(4)+1} + 2^{2(5+1)} = 2^9 + 2^{12}$. In summary, $P = \{A, B\} \in \mathbb{N}$, with $0 \in A$ and $0 \notin B$, represents the ordered pair (m, n), where $m, n \in \mathbb{N}$ are the unique natural numbers that satisfy $s(m) \oplus 1 = A$ and $s(n \oplus 1) = B$. Solving for m, n gives $m = \frac{A-1}{2}$ and $n = \frac{B}{2} - 1$, where $\frac{X}{2} = s^{-1}(X)$. The function s^{-1} displaces the set number X one unit down, $s^{-1}(X) = \{x - 1\}_{x \in X}$.

Definition 7. *Consider the family of sets*

$$\begin{aligned}
(0,) &= \{6, 18, 66, 258, 1026, \ldots, 2 + 2^{2(n+1)}, \ldots\} \\
(1,) &= \{12, 24, 72, 264, 1032, \ldots, 8 + 2^{2(n+1)}, \ldots\} \\
(2,) &= \{36, 48, 96, 288, 1056, \ldots, 32 + 2^{2(n+1)}, \ldots\} \\
&\vdots \quad \vdots \\
(m,) &= \{2^{2m+1} + 4, 2^{2m+1} + 16, \ldots, 2^{2m+1} + 2^{2(n+1)}, \ldots\}. \qquad (8)
\end{aligned}$$

Any element $x \in \bigcup_i (i,)$, in the above family of sets, is an ordered pair. The ordered pair (m, n) is the $n+1$-st element of the set $(m,)$. A finite relation is a finite subset $R \subset \bigcup_i (i,)$; elements of R are called components.

There are a few important remarks to be made. Every ordered pair of natural numbers is identified with a unique natural number. Two ordered pairs are the same if and only they are represented by the same set number. And, every natural number representing an ordered pair is a multiple of 6 (the converse is obviously not true).

This is a good definition for ordered pairs (m,n). An ordered pair $(0,n)$ is any element of the set $(0,)$. The ordered pair $(0,0)$ is represented by $6 = 2^{2(0)+1} + 2^{2(0+1)}$, and $(0,1)$ is $18 = 2^{2(0)+1} + 2^{2(1+1)}$. The third number of the set $(0,)$ represents the ordered pair $(0,2)$, etc. An element of $(1,)$ is an ordered pair of the form $(1,n)$. The ordered pair $(1,0)$ is represented by $12 = 2^{2(1)+1} + 2^{2(0+1)}$, the first object of $(1,)$. The ordered pair $(1,1)$ is $24 = 2^{2(1)+1} + 2^{2(1+1)}$, the second object of $(1,)$. The third object of $(1,)$ represents the ordered pair $(1,2)$, etc. Now, an important jump can be made, which is a continuation to the last definition. A *relation* will be defined. A finite collection of ordered pairs is a natural number,

$$\{\{A_1,B_1\},\ldots,\{A_n,B_n\}\} = 2^{2^{A_1}+2^{B_1}} + \cdots + 2^{2^{A_n}+2^{B_n}} \tag{9}$$

where A_i are odd and the B_i are even. Under this definition, a set of ordered pairs is a relation, as is usual. The information of a finite relation is stored in a single natural number, and the structure is obtainable from that number. The relation $\{(0,0),(0,1),(0,2),(2,1)\}$ is represented by the set number

$$2^{2^{2(0)+1}+2^{2(0+1)}} + 2^{2^{2(0)+1}+2^{2(1+1)}} + 2^{2^{2(0)+1}+2^{2(2+1)}} + 2^{2^{2(2)+1}+2^{2(1+1)}}.$$

Two finite relations are the same if and only if they are represented by the same natural number. For another example, take the relation $\{(2,1),(2,2),(4,2),(4,4)\}$ given by the set number

$$2^{2^{2(2)+1}+2^{2(1+1)}} + 2^{2^{2(2)+1}+2^{2(2+1)}} + 2^{2^{2(4)+1}+2^{2(2+1)}} + 2^{2^{2(4)+1}+2^{2(4+1)}}.$$

This allows for finite functions to be described as natural numbers.

3.2 Concrete Functions

In this section, the definition of finite relations is used to represent a finite function of natural numbers. Going back to the definition of relation, it is additionally required that no odd number be repeated. A finite function is represented by a set number of the form (9), where all the A_i are distinct.

Definition 8. *A function $f : A \to B$ is a set number $f = \{\{A_1,B_1\},\{A_2,B_2\},\ldots,\{A_n,B_n\}\} = 2^{2^{A_1}+2^{B_1}} + 2^{2^{A_2}+2^{B_2}} + \cdots + 2^{2^{A_n}+2^{B_n}}$, where all the A_i are distinct odd numbers and B_i are even numbers. A function is called bijective if, additionally, all the B_i are distinct. Every element of f is an arrow component. The function f maps $m \mapsto n$ if and only if $2^{2m+1} + 2^{2(n+1)} \in f$.*

A permutation $\{0,1,2,\ldots,n\} \to \{0,1,2,\ldots,n\}$ is particularly easy to identify. It is a set of $n+1$ ordered pairs, in which every element of $\{1,2,3,4,\ldots,2n,2n+1,2(n+1)\}$ appears in exactly one ordered pair. Examples of permutations are

$$\begin{aligned}
\{\{1,2\},\{3,4\}\} &= 2^{2^1+2^2} + 2^{2^3+2^4} \\
\{\{1,4\},\{3,2\}\} &= 2^{2^1+2^4} + 2^{2^3+2^2} \\
\{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\} &= 2^{2^1+2^2} + 2^{2^3+2^4} + 2^{2^5+2^6} + 2^{2^7+2^8} \\
\{\{1,6\},\{3,8\},\{5,2\},\{7,4\}\} &= 2^{2^1+2^6} + 2^{2^3+2^8} + 2^{2^5+2^2} + 2^{2^7+2^4} \\
\{\{1,4\},\{3,10\},\{5,6\},\{7,8\},\{9,2\}\} &= 2^{2^1+2^4} + 2^{2^3+2^{10}} + 2^{2^5+2^6} + 2^{2^7+2^8} + 2^{2^9+2^2} \\
\{\{1,6\},\{3,8\},\{5,2\},\{7,10\},\{9,4\}\} &= 2^{2^1+2^6} + 2^{2^3+2^8} + 2^{2^5+2^2} + 2^{2^7+2^{10}} + 2^{2^9+2^4}
\end{aligned}$$

The first permutation is the identity permutation $(0)(1)$. The second set number is representing the one-cycle permutation $(0,1)$. The third and fourth numbers represent $(0)(1)(2)(3)$ and $(0,2)(1,3)$, respectively. The fifth and sixth permutations are $(0,1,4,)(2)(3)$, and $(0,2)(1,3,4)$. A linear order is provided for the set of finite functions, and in particular permutations. The order is well behaved in several ways. If $f : \{0,1,2,\ldots,m\} \to \{0,1,2,\ldots,m\}$, and $g : \{0,1,2,\ldots,n\} \to \{0,1,2,\ldots,n\}$ are permutations and $m < n$, then the representation of f is smaller than the representation of g. This representation is not very good for measuring how much movement a permutation causes. This manner of assigning natural numbers to functions does not make a distinction between functions with the same structure. For example, the functions f, g defined by $f(0) = 0$ and $g(1) = 1$ have the

same structure but are assigned different numbers. In the following section, this issue is addressed. A natural number is assigned to any abstract finite function, in such a way that two functions are represented by the same number if and only if they have the same structure. This will be taken as definition of equivalent structure for two functions, because it gives a *modulo-structure* representation of concrete functions.

3.3 Abstract Functions

Consider the permutations $(1,2)(3,4)$ and $(1,3)(2,4)$. These will be represented by the numbers $2^{2^3+2^6}+2^{2^5+2^4}+2^{2^7+2^{10}}+2^{2^9+2^8}$ and $2^{2^3+2^8}+2^{2^7+2^4}+2^{2^5+2^{10}}+2^{2^9+2^6}$, respectively. These numbers are different. It would be useful to have a good definition, modulo the structure, for the two functions above, so that they are assigned the same natural number. In other words, it would be advantageous to number finite functions in such a manner that functions with the same structure will be represented by the same natural number. Let $f : A \to B$ a concrete function, where $A, B \in \mathbb{N}$. The first step is to forget the numeric value assigned to the elements of the components. This means that the sets A, B are no longer thought of as set numbers. Think of the elements of A and B as abstract objects with a function defined on them. Every object in $A \cup B$ is given a non-numeric symbol. For example, the function f defined by

$$\begin{aligned} f(2) &= 2 \\ f(5) &= 6 \\ f(6) &= 5 \\ f(8) &= 6 \\ f(10) &= 15 \end{aligned}$$

depends on the distinct objects $2, 5, 6, 8, 10, 15$ and it will be considered an abstract function f^* defined by

$$\begin{aligned} f^*(a) &= a \\ f^*(b) &= c \\ f^*(c) &= b \\ f^*(d) &= c \\ f^*(p) &= q \end{aligned} \tag{10}$$

Now, a way of assigning a natural number N_{f^*} to the abstract function f^*, in a sufficiently reasonable manner, is to be described. To do this, go back to the realm of numeric values. Take a fixed bijection $\eta : \{a, b, c, d, p, q\} \to \{0, 1, 2, 3, 4, 5\}$, and call it a *naming function of* f^*. Using the procedure of the last section, there is an associated representation $N_{f^*}(\eta) \in \mathbb{N}$ that depends on the naming function η and the abstract function f^*. Now consider the set of all representations $\{N_{f^*}(\eta)\}_\eta$; let η variable over all possible naming functions. In the example, there are 6! possibilities.

To find the modulo-structure representation of a concrete function f, first find the abstract function f^* corresponding to f, then proceeded to find all the possible naming functions of f^*. There is a total of $\#(A \cup B)!$ naming functions. Each naming function η provides a representation $N_{f^*}(\eta)$, so that there is a set of representations $\{N_{f^*}(\eta)\}_\eta$.

Definition 9. *Let f be a concrete function and f^* its corresponding abstract function. Their exists at least one naming, ρ, such that $N_{f^*}(\rho)$ is equal to the maximum element of the set $\{N_{f^*}(\eta)\}_\eta$. This maximum is the modulo-structure representation of f, and the symbol $N_{f^*} = N_{f^*}(\rho)$ is used.*

Let f^* and g^* abstract functions such that $N_{f^*}(\eta) = N_{g^*}(\mu)$, for some naming functions η of f^* and μ of g^*. Then $f^* = g^*$, and η, μ *are equivalent naming functions for* f^*. The sets of representations for f^*, g^* are disjoint if f^*, g^* are different functions; $f^* \neq g^*$ implies $\{N_{f^*}(\eta)\}_\eta \cap \{N_{g^*}(\mu)\}_\mu = \emptyset$. This is a good representation of abstract functions as natural numbers, because $\{N_{f^*}(\eta)\}_\eta$ is a natural number and two functions are assigned different numbers if and only if they have different structure. A linear order for finite functions has been defined. The representation of a function is a large natural number because $\#\{N_{f^*}(\eta)\}_\eta = (\#(Dom(f^*) \cup Im(f^*)))!$. If

f^* is a permutation of k objects, the representation of f^* is the sum of $k!$ many powers of 2. The representation of a permutation of 10 objects would be a natural number somewhere close to $10^{10^{230}}$. This representation can be made smaller, and the order of the functions will be invariant. In Definition 9, the fact that sets of representations are disjoint for different functions, is used. The function f is assigned the maximum of the representations, for the following reason. Let $A \cap B = \emptyset$, then the order relation of the maximum elements, $\max(A) < \max(B)$, determines the order relation $A < B$. Assigning to f the set of representations $\{N_{f^*}(\eta)\}_\eta$, or the maximum element, $N_{f^*} = \max\{N_{f^*}(\eta)\}_\eta$, defines the same order on the set of finite functions.

A definition for equivalent objects of a finite function $f : A \to B$ can also be defined. Let $\rho_1, \rho_2 : (A \cup B) \to \{0, 1, 2, \ldots, n-1\}$ two canonical naming functions so that $N_{f^*} = N_{f^*}(\rho_1) = N_{f^*}(\rho_2)$, where $n = \#(A \cup B)$. Suppose the naming functions are not equal, so that $\rho_1(x) \neq \rho_2(x)$, for some $x \in A \cup B$. Naming functions are bijections, so there exists $y \neq x$ such that $\rho_1(y) = \rho_2(x)$. Then x, y are equivalent objects because there are two distinct canonical naming functions ρ_1, ρ_2 that assign the same numerical value to x, y.

Definition 10. *Let $f : A \to B$ a finite function. Two distinct objects $x, y \in A \cup B$ are equivalent if there exist canonical naming functions ρ_1, ρ_2 such that $\rho_1(x) = \rho_2(y)$. This gives an equivalence relation on the set of objects $A \cup B$.*

This method provides two things. The set of all finite functions can be ordered (modulo structure), and a canonical naming function on the objects $Dom(f) \cup Im(f)$ is also obtained. The set of abstract finite permutations can be ordered. Also the elements of any abstract finite permutation can be ordered, and it is easy to know which objects of f are equivalent. Most importantly these orders are well behaved in several ways. Here the focus is on ordering finite permutations, and a general exposition of finite functions is left for future work. Nonetheless, some examples of general functions are given below. The representation of the first finite functions will be found, to get an intuitive grasp of this order.

The first example is of course the trivial function f_0 that sends $a \to a$. This function depends of a single object so use the set $\{0\}$ to name the set of objects $\{a\}$. Recalling the definition of ordered pair, the ordered pair $0 \to 0$ is represented by the number $6 = 2^1 + 2^2$. The odd number is used to represent the preimage and the even number to represent the image; a 0 in the preimage means 1 is an element of the ordered pair and a 0 in the image means 2 is an element of the ordered pair. The function consists of one component. Its only element is 6, so $N_{f_0} = 2^{2^1+2^2}$. To construct all finite functions in order of their representation, the next logical choice is a function f_1 defined by one component, $f_1(a) = b$. In this case, there are two objects so a naming of this function is a bijection $\{a, b\} \to \{0, 1\}$. Choosing the naming $a = 0$ and $b = 1$ gives the representation $2^{2^1+2^4}$. With the naming $a = 1$ and $b = 0$ the representation is $2^{2^3+2^2}$. It is concluded that the canonical representation of f_1 is the number $N_{f_1} = 2^{2^1+2^4}$ corresponding to the first naming function $a = 0$ and $b = 1$ because that is the maximum of the representations. These are the only two possible abstract functions of one component; namely $f_0(0) = 0$ (trivial function) and $f_1(0) = 1$ (one object sent to a different object).

Finite functions are ordered isomorphic to $\mathbb{N}$; every finite function is assigned a unique natural number. There is a set of natural numbers $\{N_f\}_{f:finite\ function}$ representing finite functions; every finite function f is represented by a unique set number N_f. Being a set of natural numbers, the set $\{N_f\}_{f:finite\ function}$ can be ordered $N_0 < N_1 < N_2 < \ldots$. Every finite function f is assigned an index; $N_f = N_k$ for some index k. The first few functions $N_0 < N_1 < N_2 < \ldots$ will be found as examples. The first two functions are the one component functions $N_0 = 2^{2^1+2^2}$ and $N_1 = 2^{2^1+2^4}$ from above. Now consider functions of two components. To find the next function, $f_2 = N_2$, add a component. But, intuitively, our order will also assign a larger representation to a function with more objects, holding fixed the number of components. Consider finite functions of two components, and two objects. There is a total of 3 functions that satisfy this conditions and they are the functions N_2, N_3, N_4. The function N_2 is given by two components that switch the objects in the domain, $f_2(a) = b$ and $f_2(b) = a$. This means the two objects of the function f_2 are equivalent. The naming $a = 0, b = 1$ or the naming $a = 1, b = 0$ give the same representation $N_2 = 2^{2^1+2^4} + 2^{2^3+2^2}$. The next function in order is the identity function on two objects, $f_3(a) = a$ and $f_3(b) = b$. Again, both objects are equivalent and they give the representation $N_3 = 2^{2^1+2^2} + 2^{2^3+2^4}$. The function N_4 is the *trivial function* that sends two objects to one of the two; the components are $f_4(a) = a$ and $f_4(b) = a$. The canonical representation $N_4 = 2^{2^1+2^4} + 2^{2^3+2^4}$ is given by the naming $a = 1$ and $b = 0$. The first summand represents $f_4(0) = 1$ and the second summand represents $f_4(1) = 1$. It is easy to verify the alternative naming function gives a smaller representation. The naming $a = 0$ and $b = 1$, gives the representation $2^{2^1+2^2} + 2^{2^3+2^2}$, of f_4. Notice that the function that seems to cause more movement, f_2, is represented by the smallest number of the three. The function that sends everything to a is the largest of the

three, and the identity is the middle number. This observation will be important in the special case of ordering permutations.

Now consider functions of two components and three objects, the next functions in the order, N_5, N_6, N_7. Each function is given with its canonical naming, and some of the other non canonical representations.

$f_5(a) = b$, $f_5(b) = c$ has canonical naming $a = 1$, $b = 2$, $c = 0$ giving ordered pairs $(1, 2), (2, 0)$ with representation

$$N_5 = 2^{2^3+2^6} + 2^{2^5+2^2}.$$

Other, non canonical, representations are $a = 0$, $b = 1$, $c = 2$ with representation $2^{2^1+2^4} + 2^{2^3+2^6}$; $a = 0$, $b = 2$, $c = 1$ with representation $2^{2^1+2^6} + 2^{2^5+2^4}$; $a = 1$, $b = 0$, $c = 2$ with representation $2^{2^3+2^2} + 2^{2^1+2^6}$; $a = 2$, $b = 1$, $c = 0$ with representation $2^{2^5+2^4} + 2^{2^3+2^2}$; $a = 2$, $b = 0$, $c = 1$ with representation $2^{2^5+2^2} + 2^{2^1+2^4}$, etc.

$f_6(a) = c$, $f_6(b) = c$ has canonical naming $a = 0$, $b = 1$, $c = 2$ giving the ordered pairs $(0, 2), (1, 2)$ with representation

$$N_6 = 2^{2^1+2^6} + 2^{2^3+2^6}.$$

The naming $a = 1$, $b = 0$, $c = 2$ is also canonical; a, b are equivalent objects of f. Other, non canonical, representations are $a = 2$, $b = 1$, $c = 0$ with representation $2^{2^5+2^2} + 2^{2^3+2^2}$; $a = 2$, $b = 0$, $c = 1$ with representation $2^{2^5+2^4} + 2^{2^1+2^4}$. etc.

$f_7(a) = a$, $f_7(b) = c$ has canonical naming $a = 2$, $b = 0$, $c = 1$ giving the ordered pairs $(2, 2), (0, 1)$ with representation

$$N_7 = 2^{2^1+2^4} + 2^{2^5+2^6}.$$

Other, non canonical, naming functions are $a = 2$, $b = 1$, $c = 0$ with representation $2^{2^3+2^2} + 2^{2^5+2^6}$; $a = 0$, $b = 1$, $c = 2$ with representation $2^{2^1+2^2} + 2^{2^3+2^6}$; $a = 0$, $b = 2$, $c = 1$ with representation $2^{2^1+2^2} + 2^{2^5+2^4}$; etc.

So far, the first eight numbers $N_0, N_1, \ldots, N_7$ have been found. To find the next numbers representing functions, in order, one must be careful. There is one function of two components and four objects. However, it is not next in order, because the functions of three components and three objects have smaller representation. Notice that the order of functions is determined first in terms of objects. Let $f : A \to B$ and $g : C \to D$ finite functions and suppose $\#(A \cup B) < \#(C \cup D)$, then $f < g$. If $\#(A \cup B) = \#(C \cup D)$, check the number of components. The function with more components has larger representation; $\#(f) < \#(g)$ implies $f < g$. Let A_n^m a finite function of n objects and m components. The following inequalities hold.

$$A_1^1 < A_2^1 < A_2^2 < A_3^2 < A_3^3 < A_4^2 < A_4^3 < A_4^4 < A_5^3 < A_5^4 < A_5^5 < A_6^3 < A_6^4 < A_6^5 < A_6^6 < \cdots$$

This simply means that apart from being well defined, the order given to finite functions is well behaved in the sense just described. The table below states the number of functions with n objects and m components. There is one function of one object and one component ($a \to a$). There is one function of two objects and one component ($a \to b$). There are three functions of two objects and two components. Three functions of three objects and two components, have also been found. This is shown in Table 2.

Suppose f, g have the same number of objects $\#O(f) = \#O(g)$, and the same number of components $\#C(f) = \#C(g)$. Their canonical representations $N_f, N_g \in \mathbb{N}$ must be found, and the order relation $N_f < N_g$ of the representations determines the order relation $f < g$ of the functions. Therefore, to compare two finite functions, it is sufficient to compute their canonical representations and compare these numbers. To find the index k such that $N_k = N_f$, is slightly more complicated. It is so far known how to find the canonical representation N_f of f. But, to know its position in the order more information than just its canonical representation is needed. The total number of functions there are of less objects, and the total number of functions that have the same number of objects but less components. Then, the canonical representation of all functions with the same number of objects and same number of components has to be found. In the table above, there are seven functions of three components and three objects. These seven functions are listed below. For simplicity of exposition, arrows are used to represent the components of a function. For example, the function defined by the three components $f(a) = f(b) = f(c) = a$ is the set of arrows of the last column. These are shown in Table 3.

Any function of three components and three objects is equivalent to one of these seven. These functions are next in the canonical ordering of finite functions; they are represented by the numbers $N_8, N_9, \ldots, N_{14}$.

Functions	**Objects**	**Components**
1	1	1
1	2	1
3	2	2
3	3	2
7	3	3
1	4	2
9	4	3
	4	4
3	5	3
	5	4
	5	5
1	6	3
	6	4
	6	5
	6	6
$\vdots$	$\vdots$	$\vdots$

Table 2: The first column indicates how many distinct functions of n objects and m components. There is no general way of calculating the number of functions, except to find all possible functions and to determine which ones are equivalent.

N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}
$a \to b$	$a \to b$	$a \to a$	$a \to a$	$a \to a$	$a \to a$	$a \to a$
$b \to c$	$b \to a$	$b \to c$	$b \to b$	$b \to a$	$b \to a$	$b \to a$
$c \to a$	$c \to a$	$c \to b$	$c \to c$	$c \to c$	$c \to b$	$c \to a$

Table 3: There is a total of seven functions of three objects and three components.

To know which of these seven functions is N_8, find the canonical representation of all seven and the one with smallest canonical representation is the function N_8, then the function N_9 is the function with second smallest representation, etc. Of these seven functions, the one with largest representation is the function N_{14}. Here they are given in order from smallest to largest (left to right). It is left as an exercise for the reader, to verify the canonical representations of these functions.

$$\begin{aligned}
N_8 &= 2^{2^1+2^4} + 2^{2^3+2^6} + 2^{2^5+2^2} \\
N_9 &= 2^{2^1+2^6} + 2^{2^3+2^6} + 2^{2^5+2^4} \\
N_{10} &= 2^{2^1+2^4} + 2^{2^3+2^2} + 2^{2^5+2^6} \\
N_{11} &= 2^{2^1+2^2} + 2^{2^3+2^4} + 2^{2^5+2^6} \\
N_{12} &= 2^{2^1+2^2} + 2^{2^3+2^6} + 2^{2^5+2^6} \\
N_{13} &= 2^{2^1+2^4} + 2^{2^3+2^6} + 2^{2^5+2^6} \\
N_{14} &= 2^{2^1+2^6} + 2^{2^3+2^6} + 2^{2^5+2^6}
\end{aligned}$$

To find the canonical representation of N_8, observe the objects are all equivalent. Let $a = 2$, then make $b = 0$ and $c = 1$, to maximize the term where a is image. The naming functions $b = 2, a = 1, c = 0$ and $c = 2, b = 1, a = 0$ also give the canonical representation. The canonical naming function of N_9 is also easy to find. Start by naming $a = 2$, since a is the most frequent object. Then make $b = 1$ because b is the object that has more relations with a. In N_{10} make $a = 2$ because a is a fixed point; this ensures the term $2^5 + 2^6$ is in the function and maximizes the value. The objects b, c are equivalent in the function N_{10} because there are two canonical naming functions $a = 2, b = 1, c = 0$ and $a = 2, b = 0, c = 1$. The rest of the canonical naming

functions are easily found.

Now consider the function of two components and four objects defined by $f_{15}(a) = c$ and $f_{15}(b) = d$. The objects in the image have priority for being assigned larger numbers, so start with naming $c = 3$ because c is in the image. Now, things change between choosing a, b, d for the value 2. Instead of assigning 2 to d, which is also in the image, use the object that is related to $c = 3$. That would be the object $a = 2$. Then, assign the values $d = 1$ and $b = 0$. The components of the function are the ordered pairs $(2, 3)$ and $(0, 1)$ stating $f_{15}(2) = 3$ and $f_{15}(0) = 1$. The set of these ordered pairs is the canonical representation $N_{15} = 2^{2^1+2^4} + 2^{2^5+2^8}$; the summand $2^{2^1+2^4}$ represents the pair $(0, 1)$ and the second summand $2^{2^5+2^8}$ represents the pair $(2, 3)$. The naming function $d = 3$, $b = 2$, $c = 1$, $a = 0$ gives components $(0, 1)$ and $(2, 3)$ so that this is also a canonical naming function. Equivalent objects are those that can be assigned the same numerical value under different canonical naming functions. Therefore, a, b are equivalent and c, d are equivalent.

Next in order are the functions of three components and four objects. Each of these nine functions is represented by one of the numbers $N_{16}, N_{17}, \ldots, N_{24}$. Any function of three components and four objects is equivalent to one of these nine. Table 4 provides these functions.

N_{16}	N_{17}	N_{18}	N_{19}	N_{20}	N_{21}	N_{22}	N_{23}	N_{24}
$a \to c$	$a \to b$	$a \to b$	$a \to c$	$a \to d$	$a \to a$	$a \to a$	$a \to a$	$a \to a$
$b \to a$	$b \to a$	$b \to d$	$b \to c$	$b \to d$	$b \to c$	$b \to d$	$b \to b$	$b \to a$
$c \to d$	$c \to d$	$c \to d$	$c \to d$	$c \to d$	$c \to d$	$c \to d$	$c \to d$	$c \to d$

Table 4: There is a total of nine functions of four objects and three components.

The smallest of these nine functions is $N_{16} = 2^{2^1+2^6} + 2^{2^5+2^8} + 2^{2^7+2^4}$ given by the canonical naming function $a = 2$, $b = 0$, $c = 3$, $d = 1$. The next function is $N_{17} = 2^{2^1+2^4} + 2^{2^5+2^8} + 2^{2^7+2^6}$ with canonical naming function $a = 3$, $b = 2$, $c = 0$, $d = 1$ and a, b are equivalent objects. The third is $N_{18} = 2^{2^1+2^6} + 2^{2^3+2^8} + 2^{2^5+2^8}$ under the naming $a = 0$, $b = 2$, $c = 1$, $d = 3$. Next is the function $N_{19} = 2^{2^3+2^8} + 2^{2^5+2^8} + 2^{2^7+2^2}$ with the naming function $a = 2$, $b = 1$, $c = 3$, $d = 0$ and a, b are equivalent objects. The function $N_{20} = 2^{2^1+2^8} + 2^{2^3+2^8} + 2^{2^5+2^8}$ has naming $a = 2$, $b = 1$, $c = 0$, $d = 3$ and a, b, c are equivalent objects. The function $N_{21} = 2^{2^3+2^6} + 2^{2^5+2^2} + 2^{2^7+2^8}$ is given by $a = 3$, $b = 1$, $c = 2$, $d = 0$. The next function is $N_{22} = 2^{2^1+2^6} + 2^{2^3+2^6} + 2^{2^7+2^8}$ with $a = 3$, $b = 1$, $c = 0$, $d = 2$ and b, c equivalent. The second largest is $N_{23} = 2^{2^1+2^4} + 2^{2^5+2^6} + 2^{2^7+2^8}$ with naming $a = 3$, $b = 2$, $c = 0$, $d = 1$ and a, b equivalent. The largest function is $N_{24} = 2^{2^1+2^4} + 2^{2^5+2^8} + 2^{2^7+2^8}$ with $a = 3$, $b = 2$, $c = 0$, $d = 1$. By now it can be appreciated that it is not trivial to find the canonical naming function of a finite function, in the general case. Careful observations have to be made to calculate the canonical naming functions, without having to find all possible representations. There are two main problems to solve in the general case, and these computational strategies will be treated in future work. 1) Finding the canonical naming function of any finite function, and 2) Finding the total number of distinct abstract functions of n objects and m components. In the next section, the suborder of finite permutations will be discussed and it proves much easier to work with.

The next functions in the order of all finite functions are functions of four objects and four components. The general analysis is left for future work, because finding all the possible distinct functions of four objects and four components is laborious. The functions that come after those are functions of three components and five objects. There is a total of three such functions.

$$\begin{array}{lll} a \to a & a \to b & a \to d \\ b \to d & b \to d & b \to d \\ c \to p & c \to p & c \to p \end{array}$$

There is one function of three components and six objects.

$$\begin{array}{l} a \to d \\ b \to p \\ c \to q \end{array}$$

The representation of f^*, in example (10) is a set of five natural numbers. The canonical naming will have to assign $\eta(a) = 5$. This guarantees $2^{11} + 2^{12} \in N_{f^*}(\rho)$ represents $f^*(a) = a$. No object has a relation with a. Any of the remaining objects b, c, d, p, q can still be assigned the value 4. If $\rho(c) = 4$ the representation is maximized because two components have power 2^{10}; namely, the components $f^*(b) = c$ and $f^*(d) = c$. Choose $\rho(b) = 3$ instead of $\rho(d) = 3$ because b is related to c by two components. This leaves us with $\rho(d) = 2$. Now assign $q = 1$ and $p = 0$. The canonical representation of

$$\begin{aligned} f^*(a) &= a \\ f^*(b) &= c \\ f^*(c) &= b \\ f^*(d) &= c \\ f^*(p) &= q \end{aligned}$$

under the canonical naming ρ is

$$N_{f^*} = 2^{2^1+2^4} + 2^{2^5+2^{10}} + 2^{2^7+2^{10}} + 2^{2^9+2^8} + 2^{2^{11}+2^{12}}$$

and there are no equivalent objects.

3.4 Finite Permutations

The suborder of permutations is easier to find, in part because it is well behaved with respect to cardinality. Let f a permutation on m objects and g a permutation on $n > m$ objects, then $N_f < N_g$. Furthermore, permutations are ordered by complexity. Given permutations f, g of the same size, they can be ordered and the interpretation is that a larger number is a simpler permutation. The identity permutation of size n has larger representation than all other permutations of size n. The one cycle permutation of n objects has the smallest representation. The number of distinct abstract permutations of size n, is equal to the number of additive partitions of n. The order of the first few permutations is given. The unique permutation P_0, of size 1, is the function f_0 of one component, represented by $N_0 = 2^6$. There are two permutations of size 2, the functions N_2 and N_3. There is a total of three permutations of size 3. These are the functions N_8, N_{10}, N_{11}. The smallest of these three numbers, N_8, represents the one cycle permutation. The middle permutation, N_{10}, leaves one object fixed. The largest, N_{11}, represents the identity permutation. Call these first six permutations $P_0 = N_0, P_1 = N_2, P_2 = N_3, P_3 = N_8, P_4 = N_{10}, P_5 = N_{11}$. Let us order the five distinct permutations of size 4. These are given in order in Table 5.

P_6	P_7	P_8	P_9	P_{10}
$a \to b$	$a \to b$	$a \to a$	$a \to a$	$a \to a$
$b \to c$	$b \to a$	$b \to c$	$b \to b$	$b \to b$
$c \to d$	$c \to d$	$c \to d$	$c \to d$	$c \to c$
$d \to a$	$d \to c$	$d \to b$	$d \to c$	$d \to d$

Table 5: There is a total of five permutations of four objects.

If two objects are in the same cycle, then they are equivalent. The converse is not true. For example, all objects of P_{10} are equivalent but they are all in different cycles. Function P_6 has canonical naming function $a = 3$, $b = 1$, $c = 0$, $d = 2$. To find its naming function, observe that all the objects are equivalent. Choose $a = 3$, without loss of generality. Next, the term where a is in the image has to be maximized. To this end, define $d = 2$. Then, to maximize the term where a is in the preimage, make $b = 1$. This implies $c = 0$. In permutation P_7 there are two pairs of equivalent objects a, b and c, d, and a canonical naming function is $a = 3$, $b = 2$, $c = 1$, $d = 0$. Permutation P_8 has canonical naming $a = 3$, $b = 2$, $c = 0$, $d = 1$, and objects b, c, d are equivalent. The canonical naming function of permutation P_9 is $a = 3$, $b = 2$, $c = 1$, $d = 0$, and objects c, d are equivalent. Any naming function of P_{10} is canonical; all objects are equivalent. The fact that all objects are equivalent does not imply every naming is a canonical naming. An example of this is P_6.

$$\begin{aligned}
P_6 &= 2^{2^1+2^6} + 2^{2^3+2^2} + 2^{2^5+2^8} + 2^{2^7+2^4} \\
P_7 &= 2^{2^1+2^4} + 2^{2^3+2^2} + 2^{2^5+2^8} + 2^{2^7+2^6} \\
P_8 &= 2^{2^1+2^4} + 2^{2^3+2^6} + 2^{2^5+2^2} + 2^{2^7+2^8} \\
P_9 &= 2^{2^1+2^4} + 2^{2^3+2^2} + 2^{2^5+2^6} + 2^{2^7+2^8} \\
P_{10} &= 2^{2^1+2^2} + 2^{2^3+2^4} + 2^{2^5+2^6} + 2^{2^7+2^8}
\end{aligned}$$

To find the maximum $N_f(\rho) = \max_\eta\{N_f(\eta)\}$, over all possible naming η, find the canonical naming providing the largest representation. To maximize the set number $\{\{a, f(a)\}, \{b, f(b)\}, \{c, f(c)\}, \{d, f(d)\}\}$. The number $2^7 + 2^8 \in f$ is a component of the canonical naming function if $f(x) = x$ for some $x \in \{a, b, c, d\}$. If there are no fixed points, look for cycles of two objects, and continue looking for the smallest possible cycle, to assign the largest values of the naming. The largest possible set number that can represent an abstract permutation of four elements is $2^{2^1+2^2} + 2^{2^3+2^4} + 2^{2^5+2^6} + 2^{2^7+2^8}$, representing the identity permutation $(0)(1)(2)(3)$. The number N_f measures and compares how much *movement* a permutation causes. If f and g are permutations of the same number of objects and f has less fixed objects than g, then $N_f < N_g$. The more complicated a permutation becomes the smaller its representation becomes (holding fixed the permutated set). Intuitively, assigning larger values to objects in smaller cycles helps to maximize the representation.

One more example of permutations will be given before applying the same method to define groups. Let f be the permutation $(a)(b, c)(p, q, r)$ on the set of abstract objects $\{a, b, c, d, p, q, r\}$. A canonical ordering of its elements, and the canonical representation N_f will be found. It should result in $a = 5$, $b = 4$, $c = 3$, $p = 2$, $q = 0$, $r = 1$, or one of its equivalent numbering functions, and

$$N_f = 2^{2^1+2^4} + 2^{2^3+2^6} + 2^{2^5+2^2} + 2^{2^7+2^{10}} + 2^{2^9+2^8} + 2^{2^{11}+2^{12}}.$$

In all the equivalent numbering functions, $a = 0$. An alternative is $b = 3$ and $c = 4$. The values of the objects in the 3-cycle can also be changed. Make $q = 2$, then $p = 1$ and $r = 0$. If $r = 2$, then $q = 1$ and $p = 0$.

4 Finite Groups

Using the results from the previous sections, finite groups can be represented with natural numbers. A finite group $G(*)$ is a bijection that assigns permutations, of the set G, to objects of G. Operation functions are the elements in the image of $* : G \to \ Aut(G)$. Consider a naming η of the set G. Then the objects of G, and the operation functions of G are set numbers. Thus, $*$ is a function of the form $M \to N$, where $\max(M) < \min(N)$. If the group has k elements, the domain $M = Dom(*)$ is the set number $\{0, 1, 2, \ldots, k-1\}$. The image $N = Im(*)$ is the set number $\{N_{*0}(\eta), N_{*1}(\eta), N_{*2}(\eta), \ldots, N_{*(k-1)}(\eta)\}$, where the operation functions N_{*x} are concrete functions. The operation function $*x$ is represented by a natural number $N_{*x}(\eta)$, given a naming function η. The definition of group satisfies the definition of function. Every finite group is a set number whose elements are ordered pairs. The ordered pairs are sets of two objects; one odd and one even. The first components are odd numbers $2i+1$, for every $i \in \{0, 1, 2, \ldots, k-1\}$. The second components are even numbers representing permutations, $2(N_{*x}(\eta)+1)$. Every naming function η defines a natural number $N_G(\eta)$, that depends on the group and the naming function of that group. There is a finite number of these representations. The maximum representation is the canonical representation $N_G = \max\{N_G(\eta)\}_\eta$ of the group G. This canonical representation gives us a canonical ordering of the elements of G, as well. It behaves much like the representations of permutations. The largest value is assigned to the identity element, $e = k - 1$, in any canonical naming function. A group is a set number of the form

$$\begin{aligned}
&2^{2^{2(k-1)+1}+2^{2\left(2^{(2^1+2^2)}+2^{(2^3+2^4)}+\cdots+2^{(2^{2k-1}+2^{2(k-1+1)})}+1\right)}} + 2^{2^{2(k-2)+1}+2^{2\left(2^{(2^1+2^a)}+2^{(2^3+2^b)}+\cdots+2^{(2^{2k-1}+2^{2(k-2+1)})}+1\right)}} + \\
&+2^{2^{2(k-3)+1}+2^{2\left(2^{(2^1+2^c)}+2^{(2^3+2^d)}+\cdots+2^{(2^{2k-1}+2^{2(k-3+1)})}+1\right)}} + 2^{2^{2(k-4)+1}+2^{2\left(2^{(2^1+2^x)}+2^{(2^3+2^y)}+\cdots+2^{(2^{2k-1}+2^{2(k-4+1)})}+1\right)}} + \\
&\ldots + 2^{2^{2(0)+1}+2^{2\left(2^{(2^1+2^z)}+2^{(2^3+2^w)}+\cdots+2^{(2^{2k-1}+2^{2(0+1)})}+1\right)}},
\end{aligned}$$

where the $k-1$ numbers $a, b, \ldots$ are distinct elements of $\{2, 4, 6, \ldots, 2k-6, 2k-4, 2k\}$, the numbers $c, d, \ldots$ are distinct elements of $\{2, 4, 6, \ldots, 2k-6, 2k-2, 2k\}$, the numbers $x, y, \ldots$ are distinct elements of $\{2, 4, 6, \ldots, 2k-$

$8, 2k-4, 2k-2, 2k\}$, etc. The numbers $z, w, \ldots$ are distinct elements of $\{4, 6, \ldots, 2k\}$. Also, the numbers $a, c, x, z, \ldots$ are all pairwise different and distinct from 2. All the $b, d, y, w, \ldots$ are pairwise different and distinct from 4, etc. The first term, $2^{2^{2(k-1)+1}+2^{2\left(2^{(2^1+2^2)}+2^{(2^3+2^4)}+\cdots+2^{(2^{2k-1}+2^{2(k-1+1)})}+1\right)}}$, indicates that $e = k-1$ is assigned to the identity function. Not all natural numbers of this form are groups. It is additionally required that the associative property holds. Later in this section it will be seen that verifying the associative property is a straightforward process. The composition of functions will be analyzed numerically. With abstract permutations there was a canonical representation, given by a canonical naming of the objects. Here a similar situation arises, now in the context of groups.

Theorem 7. *Let G a finite group of order k, then a naming function $\rho : G \to \{0, 1, 2, \ldots, k-1\}$ exists and a corresponding canonical representation $N_G = \max_{\eta} N_G(\eta) = N_G(\rho)$. The bijection ρ is the canonical ordering of G, and $\rho(e) = k-1$. Two distinct group objects x, y are equivalent if their exists two distinct canonical orderings ρ_1, ρ_2 such that $\rho_1(x) = \rho_2(y)$.*

This gives a well defined linear order on the set of finite groups. Two groups have the same canonical representation if and only if they are isomorphic. This linear order is well behaved with respect to cardinality; $|G| < |H|$ if and only if $N_G < N_H$.

The order of a group element, $|g|$, is the smallest power n such that $g^n = e$. The identity element is assigned to $k-1$, to maximize the representation. Then identify the objects of smallest order, in G; this number is the smallest prime number that divides $|G|$. The number $k-2$ will be assigned to one of these objects. The first groups are constructed to illustrate the procedure of finding canonical representation of a group. Start with the trivial group of one object. The group G_0 is determined by the relation $*a(a) = a$. The trivial naming $a = 0$ is given to it and the operation function N_{*0} is the one component function $P_0 = N_0 = 2^{2^1+2^2}$. Although the numeric value of a group G is used with the notation N_G, a finite group can also be represented with G_k for a natural index k because all finite groups will be generated in a linear order. The first group is the trivial group. The canonical representation is

$$G_0 = 2^{2^{2(0)+1}+2^{2\left(2^{2^1+2^2}+1\right)}} = 2^{2^1+2^{2\left(2^6+1\right)}}.$$

Before continuing on to more groups, the use of a table notation is explained, to represent a set of permutations. The same notation of a column of arrows to represent a single permutation is used, but the arrows are not written. For example, the permutation $(a, b)(c)(d)$ can be written as

$$\begin{matrix} a & b \\ b & a \\ c & c \\ d & d \end{matrix}$$

To represent several permutations of the same size, a single rectangular grid is required. For this, one column is used as a pivot for the rest. For example, the set of permutations $\{(a, b)(c)(d),\ (a, b, c, d),\ (a)(c, b)(d), (a)(b)(c)(d)\}$ can be written as a single rectangular grid of $4+1$ columns. The first (left-most) column serves as pivot by which all other columns are defined. The second column represents the permutation $(a, b)(c)(d)$, the third column represents the permutation (a, b, c, d), the fourth column is $(a)(b, c)(d)$ and the fifth column is $(a)(b)(c)(d)$.

$$\begin{matrix} a & b & b & a & a \\ b & a & c & c & b \\ c & c & d & b & c \\ d & d & a & d & d \end{matrix}$$

In the particular case of groups, the table is square and rows and columns do not repeat objects. Additionally, one column must be equal to the identity permutation, so the following convention is adopted. The left-most column will represent the identity permutation. The identity object will be in the upper left hand corner. The second column is representing the operation function of the second object in the first column. The third column represents the operation function of the third object in the first column. In general, if a is the $k-th$ object in the first column, then the operation function $*a$ is represented in the $k-th$ column of the table. Therefore, an

operation is written in the usual table form, so that the following table has products such as $e*e=e$, $a*e=a$, $b*b=e$, $a*c=e$, and the like.

$$\begin{array}{cccc} e & a & b & c \\ a & b & c & e \\ b & c & e & a \\ c & e & a & b \end{array}$$

This simply means that given any fixed position, the object in that position is expressed in terms of the operation between the first objects of that column and row. This form of writing the operation functions coincides with the multiplication table of the group. If x is the $k-th$ object in the first row, then the $k-th$ column gives the corresponding operation function $*x$. In the process of finding the canonical representation of a group, the associative property will have to be verified frequently. This is given by a simple rule on the group table. Let x be any object in a group G of order n. The element x appears in the table exactly n times; once in each column/row. Each one of the positions where x appears, is an expression for x in terms of two objects; a factorization $x=y*z$. Given a table representing a set of operation functions, the operation satisfies the associative property if and only if $*x=*y\circ *z$, for every factorization $x=y*z$ of every $x\in G$. In the table example above, $b=a*a$ so that $*b=*a\circ *a$ has to be verified. To verify this is true, prove $*b(g)=(*a\circ *a)(g)$ for every $g\in G$. To find $b*c=*b(c)$, the arrows $c\rightarrow_{*a} e\rightarrow_{*a} a$ are used. Also, $b*a=*b(a)$ given by the arrows $a\rightarrow_{*a} b\rightarrow_{*a} c$. For another example, take the product $e=c*a$. It must be proven $*c\circ *a$ is the identity function. First find $(*c\circ *a)(b)$, which is given by the arrows $b\rightarrow_{*a} c\rightarrow_{*c} b$. Also, $(*c\circ *a)(a)$ is given by $a\rightarrow_{*a} b\rightarrow_{*c} a$, etc. The construction of groups can be continued having in mind the above rules. A group of two objects will have a table of the form

$$\begin{array}{cc} e & g_1 \\ g_1 & \end{array}$$

Of course g_1 has an inverse $\neq e$, so that $g_1*g_1=e$

$$\begin{array}{cc} e & g_1 \\ g_1 & e \end{array}$$

The canonical naming function is trivial to find. To maximize the representation, make $e=1$, $g_1=0$. The group has numeric table

$$\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}$$

The group is an operation. This operation is a concrete function of two components. The first component is $*(e)=\mathbf{id}$, that sends $e=1$ to the identity function $(0)(1)$. The second component of the operation is $*(g_1)=(0,1)$, that sends the object $0=g_1$ to the permutation $(0,1)$. The canonical representation has two terms. The first term representing the first component is $2^{2^{2(1)+1}+2^{2^{\left(2^{2^3}+2^{2^4}+2^{2^1}+2^{2^2}+1\right)}}}$. The second component is given by the expression $2^{2^{2(0)+1}+2^{2^{\left(2^{2^3}+2^{2^2}+2^{2^1}+2^{2^4}+1\right)}}}$. The canonical representation, $N_{\mathbb{Z}_2}$, of the group $\mathbb{Z}_2$ is

$$\begin{aligned} G_1 &= 2^{2^{2(1)+1}+2^{2^{\left(2^{2^3}+2^{2^4}+2^{2^1}+2^{2^2}+1\right)}}}+2^{2^{2(0)+1}+2^{2^{\left(2^{2^3}+2^{2^2}+2^{2^1}+2^{2^4}+1\right)}}} \\ &= 2^{2^3+2^{2^{\left(2^6+2^{2^4}+1\right)}}}+2^{2+2^{2^{\left(2^{18}+2^{12}+1\right)}}}. \end{aligned}$$

Why is (11) the canonical representation? The canonical representation is the maximum of the representations. In this case there are two possible representations, one for each naming function. The naming defined by $e=0$ and $g_1=1$, has the representation

$$2^{2^{2(0)+1}+2^{2^{\left(2^{2^3}+2^{2^4}+2^{2^1}+2^{2^2}+1\right)}}}+2^{2^{2(1)+1}+2^{2^{\left(2^{2^3}+2^{2^2}+2^{2^1}+2^{2^4}+1\right)}}}$$

because now 0 is assigned to the identity function, while 1 is assigned the permutation $(0,1)$. This representation is smaller than the canonical representation above. The reader should understand why this is true, before moving

on to the next examples. Remember, the largest number of the naming will be assigned to the identity object because this maximizes the representation. Naming the rest of the objects, to obtain the canonical representation, will be described below.

To make a distinction, a *term* is a number representing a component $x \to_* *x$ of the operation. The upper terms are the numbers representing components of the operation functions; they are called *sub terms*. For example, $2^{2^3+2^2}$ is a sub term of the term $2^{2^{2(0)+1}+2^{2\left(2^{2^3+2^2}+2^{2^1+2^4}+1\right)}}$. Terms are ordered pairs; they are elements of the set $\bigcup_i (i,)$, defined at the beginning of section 4.1. Notice in the second equality, that sub terms are also ordered pairs. For example, the sub term $2^{2^3+2^2}$ and the term $2^3 + 2^{2\left(2^6+2^{2^4}+1\right)}$ are both numbers of the form $2^{2m+1} + 2^{2(n+1)}$. They are both concrete arrows.

4.1 $|G| = 3$

Next are groups of three objects. Start with the table

$$\begin{array}{ccc} e & g_1 & g_2 \\ g_1 & & \\ g_2 & & \end{array}$$

All objects of G have to satisfy $|g| \mid 3$, so that $|g| = 3$ for all $g \in G$. This means $g_1^2 \neq e$. Since g_1 is not the identity element, $g_1^2 \neq g_1$. Therefore g_1^2 is a new object g_2, and $g_1 * g_2 = g_1^3 = e$.

$$\begin{array}{ccc} e & g_1 & g_2 \\ g_1 & g_2 & \\ g_2 & e & \end{array}$$

Use the associative rule to find the column of g_2. It is true that $g_2 = g_1^2$, so that $*g_2$ is the function $*g_1 \circ *g_1$. To find $*g_2(g_1)$, follow the arrows $g_1 \to_{*g_1} g_2 \to_{*g_1} e$ so that $g_2 * g_1 = e$. In the same way $g_2 \to_{*g_1} e \to_{*g_1} g_1$, so that $g_2^2 = g_1$.

$$\begin{array}{ccc} e & g_1 & g_2 \\ g_1 & g_2 & e \\ g_2 & e & g_1 \end{array}$$

This is the group $\mathbb{Z}_3$. To find the canonical naming function, start with $e = 2$. One of the non trivial objects g_1, g_2 will have to be assigned the value 1 and the other will be assigned the value 0. Next it is necessary to know which of the two objects will be assigned the value 1 and which will be assigned the value 0. There are two different canonical naming functions. These are $\rho_1 : e = 2$, $g_1 = 1$, $g_2 = 0$, and $\rho_2 : e = 2$, $g_1 = 0$, $g_2 = 1$. Either of these naming functions will give the numerical table

$$\begin{array}{ccc} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 1 \end{array} \tag{11}$$

The canonical representation of the group is a concrete function of three components. Use either of the two canonical naming functions to find it. The first component is the ordered pair that assigns 2 to the identity permutation $(0)(1)(2)$ because $e = 2$ is the identity object. This ordered pair is represented by the number

$$2^{2^{2(2)+1}+2^{2\left(2^{2^5+2^6}+2^{2^3+2^4}+2^{2^1+2^2}+1\right)}}.$$

The second component assigns $g_1 = 1$ to the concrete permutation $(2, 1, 0)$ because this is the permutation represented by the column of 1, in (11). This component is given by

$$2^{2^{2(1)+1}+2^{2\left(2^{2^5+2^4}+2^{2^3+2^2}+2^{2^1+2^6}+1\right)}}.$$

The object $g_2 = 0$ is assigned the permutation $(2, 0, 1)$ because this is the permutation given by the column of 0, in table (11). The third term is the number

$$2^{2^{2(0)+1}+2^{2^{\left(2^{2^5+2^2}+2^{2^3+2^6}+2^{2^1+2^4}+1\right)}}}.$$

The canonical representation is the number

$$2^{2^{2(2)+1}+2^{2^{\left(2^{2^5+2^6}+2^{2^3+2^4}+2^{2^1+2^2}+1\right)}}}+2^{2^{2(1)+1}+2^{2^{\left(2^{2^5+2^4}+2^{2^3+2^2}+2^{2^1+2^6}+1\right)}}}+2^{2^{2(0)+1}+2^{2^{\left(2^{2^5+2^2}+2^{2^3+2^6}+2^{2^1+2^4}+1\right)}}}$$
$$=2^{2^5+2^{2^{\left(2^6+2^{24}+2^{96}+1\right)}}}+2^{2^3+2^{2^{\left(2^{66}+2^{12}+2^{48}+1\right)}}}+2^{2^1+2^{2^{\left(2^{18}+2^{72}+2^{36}+1\right)}}}$$

4.2 $|G| = 4$

Before moving on to finding groups of four objects, one more thing is brought to attention. Given a finite group G, the list of all the operations $a * b$ is a system of equations that defines the group. The procedure used here for finding groups, will provide a *minimal set of independent equations* that determine each group. The group $\mathbb{Z}_2$ is determined by the expression $a^2 = e$ (trivial expressions, $e^2 = e$ and $e * x = x * e = x$ do not have to be written down). So, $\mathbb{Z}_2$ is a group determined by one equation. The group $\mathbb{Z}_3$ is given by the expressions $a^2 = b$ and $a^3 = e$. From these two equations, the complete list of operations of the group can be derived.

Klein 4-Group. Start with a set $\{e, g_1, g_2, g_3\}$. There is at least one object with order equal to the smallest prime divisor of 4; suppose $g_1^2 = e$, without loss of generality.

$$\begin{array}{cccc} e & g_1 & g_2 & g_3 \\ g_1 & e & g_3 & g_2 \\ g_2 & g_3 & & \\ g_3 & g_2 & & \end{array} \tag{12}$$

There are two possibilities $g_2^2 = e$ or $g_2^2 = g_1$. Suppose the first case is true. Then the Klein four-group, $K(4)$, is determined. Any group of of four elements e, g_1, g_2, g_3 such that $e = g_1^2 = g_2^2$, is isomorphic to $K(4)$.

$$\begin{array}{cccc} e & g_1 & g_2 & g_3 \\ g_1 & e & g_3 & g_2 \\ g_2 & g_3 & e & g_1 \\ g_3 & g_2 & g_1 & e \end{array}$$

To find the canonical naming functions start with $e = 3$. there are three remaining objects g_1, g_2, g_3. To find their values start with the list of objects in table form. All the non trivial objects are second order objects whoever $2, 1, 0$ may be.

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2 & 3 & & \\ 1 & & 3 & \\ 0 & & & 3 \end{array}$$

Already, this determines the group.

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 \end{array}$$

This means that any naming function with $e = 3$ is a canonical naming function. The objects g_1, g_2, g_3 are all equivalent, so that $K(4)$ has a total of six canonical naming functions. The object 2 can be chosen from three possible options. The object 1 can be chosen from the remaining two objects and 0 is determined as the remaining object. A naming function will be represented by a sequence. For example, the naming function $e = 3$, $g_1 = 2$, $g_2 = 1$, $g_3 = 0$ is written as $\eta(e, g_1, g_2, g_3)$. The six naming functions are

$$\begin{array}{ccc} \eta(e, g_1, g_2, g_3) & \eta(e, g_2, g_1, g_3) & \eta(e, g_3, g_1, g_2) \\ \eta(e, g_1, g_3, g_2) & \eta(e, g_2, g_3, g_1) & \eta(e, g_3, g_2, g_1) \end{array}$$

Any naming function with $e = 3$, gives the numeric table

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 \end{array}$$

and canonical representation

$$\begin{aligned} N_{K(4)} \;=\; & 2^{2^7+2^{2^{\left(2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} + 2^{2^5+2^{2^{\left(2^{(2^7+2^6)}+2^{(2^5+2^8)}+2^{(2^3+2^2)}+2^{(2^1+2^4)}+1\right)}}} \\ & +2^{2^3+2^{2^{\left(2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}}} + 2^{2^1+2^{2^{\left(2^{(2^7+2^2)}+2^{(2^5+2^4)}+2^{(2^3+2^6)}+2^{(2^1+2^8)}+1\right)}}}. \end{aligned}$$

The first term is the component that sends 3 to the identity function $(0)(1)(2)(3)$, while the second term is the component that sends 2 to the permutation $(0,1)(2,3)$, etc. The group $K(4)$ has a total of six automorphisms, and there are a total of six distinct canonical naming functions. This in not coincidental. Each of these six naming functions determines an automorphism of $K(4)$. Hold one of these fixed as *pivot*. For example, take the pivot $A = \eta(e, g_3, g_1, g_2)$ which will be held fixed. Choose a second canonical naming function $B = \eta(e, g_1, g_3, g_2)$. The function that sends the first component of A to the first component of B, and the second component of A to the second component of B, etc. is called a *component function*. The component function $\phi : A \to B$ is an automorphism, for every canonical naming function B. That is to say, ϕ that acts by $e \mapsto e$, $g_1 \mapsto g_3$, $g_2 \mapsto g_2$, $g_3 \mapsto g_1$ is an automorphism of $K(4)$. Choosing $B = A$ then the identity automorphism is being described. Let each of the of the canonical naming functions defines an automorphism. The initial choice of A is inconsequential; any choice for A gives the same set of functions.

Cyclic group $\mathbb{Z}_4$. Going back to table (12), consider the second case, $g_2^2 = g_1$. This determines the table

$$\begin{array}{cccc} e & g_1 & g_2 & g_3 \\ g_1 & e & g_3 & g_2 \\ g_2 & g_3 & g_1 & e \\ g_3 & g_2 & e & g_1 \end{array}$$

This is the cyclic group $\mathbb{Z}_4$, determined by the equations $g_1^2 = e$ and $g_2^2 = g_1$. To find the canonical representation, be careful to assign values. It is not known which object of $\mathbb{Z}_4$ takes each value of $0, 1, 2$. If the value 2 is assigned to g_2, g_3, the numeric table

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2 & 1 & & \\ 1 & & & \\ 0 & & & \end{array}$$

On the other hand, if 2 is assigned to the second order object, g_1, the table is

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2 & 3 & & \\ 1 & & & \\ 0 & & & \end{array}$$

The latter maximizes the representation. Intuitively, try to assign the larger numbers by giving priority to the left-most columns. Within a column give priority to the objects of upper rows. Place larger numbers further to the left and then further to the top of the table. The rest of the table is determined.

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 0 & 2 & 3 \\ 0 & 1 & 3 & 2 \end{array}$$

Any naming function with $e = 3$, $g_1 = 2$ is a canonical naming function. One canonical naming function is $e = 3$, $g_1 = 2$, $g_2 = 1$, $g_3 = 0$ which is written as $\eta(e, g_1, g_2, g_3)$. The other canonical naming function is $\eta(e, g_1, g_3, g_2)$. This implies g_2, g_3 are equivalent objects. There are two automorphisms. Fix $A = \eta(e, g_1, g_3, g_2)$ then $B = \eta(e, g_1, g_2, g_3)$. This determines the automorphism with components $e \mapsto e$, $g_1 \mapsto g_1$, $g_2 \mapsto g_3$, $g_3 \mapsto g_2$. If $B = A$ the identity automorphism is determined. The canonical representation is

$$\begin{aligned} N_{\mathbb{Z}_4} &= 2^{2^7+2^{2^{\left(2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} + 2^{2^5+2^{2^{\left(2^{(2^7+2^6)}+2^{(2^5+2^8)}+2^{(2^3+2^2)}+2^{(2^1+2^4)}+1\right)}}} \\ &\quad + 2^{2^3+2^{2^{\left(2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^6)}+2^{(2^1+2^8)}+1\right)}}} + 2^{2^1+2^{2^{\left(2^{(2^7+2^2)}+2^{(2^5+2^4)}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}}}. \end{aligned}$$

The groups $K(4)$ and $\mathbb{Z}_4$ are compared in terms of the order of natural numbers. The odd number of the terms do not determine the order because the even numbers, representing the operation functions, are larger than the odd numbers. The group with the largest operation function, that is not in both groups, is the larger of the two. The group $K(4)$ has larger numeric representation than $\mathbb{Z}_4$ because $K(4)$ has the largest operation function that is not in both groups. The cyclic group has the smallest representation, $N_{\mathbb{Z}_4} < N_{K(4)}$, just as the one cycle permutation (a, b, c, d) has smaller representation than $(a, b)(c, d)$. Try to find all groups with less than ten objects. The minimum independent set of equations, the canonical naming functions of its elements, the set of automorphisms, the canonical table and canonical numeric representation are given in the second appendix.

5 Infinite Sets and Real Numbers

In this section, the structure of real numbers is built using the same principles of the construction of natural numbers. The methods are simply extended to the case of infinite sets. First of all, notice that any real number in the unit interval $(0, 1]$ can be given in negative powers of 2. For example, the number $\frac{1}{2} = 2^{-1}$ and $\frac{3}{4} = 2^{-1} + 2^{-2}$.

A second observation is made. Consider the energy level graph of a sum, as in Figure 1. Notice that a vertical displacement of the configuration of points, gives another true statement. What happens if a displacement is made into negative integers? The statement still holds true. See Figure 3. This is true because negative powers of two are still operated with the same rule. The expression $2^n + 2^n = 2^{n+1}$ holds for any integer n, not only positive integers. For example, to add the numbers $\frac{1}{2} + \frac{3}{4}$, the equality is $2^{-1} + 2^{-1} + 2^{-2} = 2^0 + 2^{-2} = 1 + \frac{1}{4}$. This is used in formalizing real numbers.

Natural numbers are represented as hereditarily finite sets, and $\mathbb{N} = \mathbf{HFS}$. The subset axiom implies that any sub collection of $\mathbb{N}$, is a set. In particular, infinite sub collections of $\mathbb{N}$ are sets. In this section it will be proven that these sets are the real numbers. The section is divided in three main parts.

1. **Continuum $[0, 1]$.** Every real number in the unit interval $(0, 1]$ can be uniquely expressed as sum of infinitely many negative powers of 2. Moreover, every infinite set of natural numbers defines a unique real number in the unit interval. Supremum and addition properties are simple to prove.

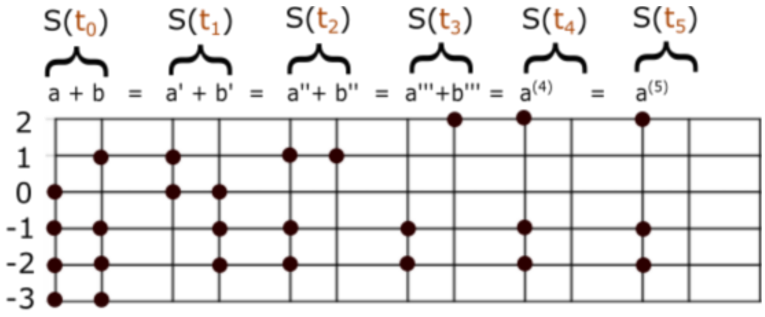

Figure 3: The energy level interpretation can be taken to negative levels. Particles occupying these levels represent negative powers of 2. In Figure 1 this represented $15 + 23 = 38$. Here, the statement is $1.875 + 2.875 = 4.75$.

2. **Real Numbers.** The constructions of $\mathbb{N}$ and $[0,1]$ are generalized to represent positive real numbers as infinite subsets of $\mathbb{Z}$. Then, the set of infinite subsets of $\mathbb{N}$ is given the structure of $\mathbb{R}$ with order and operation properties.

3. **Limits and Continuity.** The concept of limit and continuity has a simple description in terms of these constructions. An initial description of Analysis, in terms of $\mathbb{N}_<$, is provided.

5.1 Continuum $[0,1]$

A real number $x \in (0,1]$ can be expressed as a sum of negative powers of 2, so that $x = \sum_{i \in X} 2^{-i}$ for some set $X \subseteq \mathbb{N}$. The set X is the set number corresponding to x. The set number X can be a finite set (for some rational numbers). However, notice that a number 2^{-k} can be seen as an infinite sum $\left(\sum_{i=k+1}^{\infty} 2^{-i} \right)$. Thus, every $x \in (0,1]$ is represented by a unique infinite set of natural numbers. The symbol $\mathbb{N}_1 = \{1,2,3,\ldots\}$ is used for the set of natural numbers greater than 0. A bijection $\mathbb{N}_{inf} \to (0,1]$, where $\mathbb{N}_{inf}$ is the set of all infinite subsets of $\mathbb{N}_1$, will be given in this section. Infinite subsets of $\mathbb{N}$ are called *infinite set numbers* and they are ordered similarly to finite set numbers, but with one difference. The smaller powers of 2 represent larger numbers. For example, $2^{-5} < 2^{-1}$. Instead of using the maximum of the set difference, now it is the minimum. Therefore, $A < B$ if and only if $\min(A \triangle B) \in B$. Notice that $1 \in \mathbb{R}$ is the set number $\mathbb{N}_1$. Obviously, any two objects in $\mathbb{N}_{inf}$ are comparable in terms of this order relation, $<$, because the symmetric difference is non empty for set numbers $A \neq B$. Then, $\min(A \triangle B)$ exists because of the well order principle.

The order for the continuum has been defined in terms of the minimum of symmetric difference, and not in terms of addition as was the case with the order of natural numbers. This is because addition is not yet defined for infinite set numbers. It is trivial to verify the order is anti symmetric. The order is also transitive. Suppose $A < B$ and $B < C$. There is an object $b_1 \in B/A$ such that $b_1 < a$ for every $a \in A/B$. Also, there exists an object $c_0 \in C/B$ such that $c_0 < b$ for every $b \in B/C$. Suppose there exists $a_2 \in A/C$ such that $a_2 < c$ for every $c \in C/A$. Treat two cases and in each arrive at a contradiction, proving $A < C$.

1. Suppose $a_2 \in B$. But it is also true that $a_2 \notin C$. Then $c_0 < a_2$. Therefore, $c_0 \in A$. It is also true $c_0 \notin B$. This implies $b_1 < c_0$, which in turn means $b_1 \in C$. And, given that $b_1 \notin A$, it follows that $a_2 < b_1$. Use transitivity in $\mathbb{N}$ to find contradiction.

2. Suppose $a_2 \notin B$. This implies $b_1 < a_2$. It is true that $b_1 \notin A$, so that $b_1 \in C$ implies $a_2 < b_1$ which is a contradiction. It must be the case that $b_1 \notin C$. Then, $c_0 < b_1$. For $c_0 < a_2$ to be true, it must be true that $c_0 \in A$. But, this would imply $b_1 < c_0$, again a contradiction.

This proves the order on $\mathbb{N}_{inf}$ is transitive. The collection $\mathbb{N}_{inf}$ has been ordered isomorphic to $(0,1]$. The real number 1 is the set $\mathbb{N}_1$. To include the real number 0, in the order, consider $\mathbb{N}^*_{inf} = \mathbb{N}_{inf} \cup \{\emptyset\}$. This is the

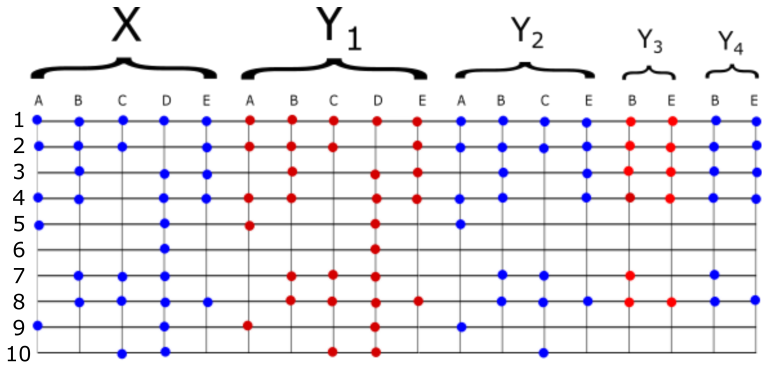

Figure 4: The iterations for finding the supremeum of the family $X = \{A, B, C, D, E\}$ is represented graphically. The elements of X are set numbers in the unit interval. For example, $A = 2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-9} = 0.845703125$.

collection whose objects are the infinite subsets of $\mathbb{N}$, plus the empty set. The order of $[0,1]$ is given in terms of sets, where $0=\emptyset$, every $x\in(0,1]$ is an infinite set of natural numbers and $1=\mathbb{N}_1$. The most important aspect in the order of a continuum is the existence of a least upper bound of any subset, which is constructed explicitly. Let $\mathbf{X}\subseteq\mathbb{N}_{inf}$; every element of $\mathbf{X}$ is an infinite set of natural numbers. Define $x_1=\min(\bigcup\mathbf{X})$ and $Y_1=\{A\in\mathbf{X}|x_1\in A\}$. Let

$$x_{n+1}=\min\left(\bigcup Y_n-\{x_i\}_{i=1}^n\right),$$

where $Y_n=\{A\in Y_{n-1}|x_n\in A\}$. The set number $\{x_i\}_i\in\mathbb{N}_{inf}$ is the supremum of X, by construction. This is shown in Figure 4.

The next step, after defining the order for infinite set numbers, is to define their addition operation. Let $r=s^{-1}$; the inverse function of s. Recall, this function subtracts 1 one unit to the elements of the argument. Given two infinite set numbers $A=\{a_1,a_2,\ldots\}$ and $B=\{b_1,b_2,\ldots\}$, let $A_n=\{a_k\}_{k=1}^n$ and $B_n=\{b_k\}_{k=1}^n$ be the sets of the first n objects. Define

$$A_n\oplus B_n=(A_n\triangle B_n)\oplus r(A_n\cap B_n).$$

The addition $A\oplus B$ is the supremum of the finite sums,

$$A\oplus B=\sup_n(A_n\oplus B_n).$$

5.2 Real Numbers

Based on the observation of Figure 3, the previous constructions of $\mathbb{N}$ and $[0,1]$ can be generalized, into the structure of positive real numbers, $\mathbb{R}_0^+$. But first, it must be proven that integers are sets. Take the integer $1\in\mathbb{Z}$; it is the function $\oplus 1$. A finite function is a finite set of set numbers. Then, the function $\oplus 1$ is the object $\{\{1,4\},\{3,6\},\{5,8\},\{7,10\},\ldots\}$. Thus, the integer $1\in\mathbb{Z}$ is an infinite set number and, in particular, a set. The integer $2\in\mathbb{Z}$ is the infinite set $\{\{1,6\},\{3,8\},\{5,10\},\{7,12\},\ldots\}\in\mathbb{N}_{inf}^*$, etc. The negative integer $-1\in\mathbb{Z}$ is the object $\{\{\{3,2\},\{5,4\},\{7,6\},\{9,8\},\ldots\}$. The negative integer $-2\in\mathbb{Z}$ is the object $\{\{\{5,2\},\{7,4\},\{9,6\},\{11,8\},\ldots\}$, etc. That is to say, integers are a sub collection of the collection $\mathbb{N}_{inf}^*$. To prove the collection of integers is a set, it is sufficient to prove $\mathbb{N}_{inf}^*$ is a set, because of the axiom of subsets. The Axiom of Subsets establishes that the elements of $\mathbb{N}_{inf}^*$ are sets. However, to prove $\mathbb{N}_{inf}^*$ is a set, the Power Set Axiom is needed.

Let $\bar{\mathbb{Z}}\subset P(\mathbb{N}_{inf}^*)$ the set whose objects are subsets of $\mathbb{Z}$, that are bounded above, in terms of the order defined for integers (which is different than the order on infinite set numbers). Put differently, $A\in\bar{\mathbb{Z}}$ if and only if $A\subset\mathbb{Z}$ and $\max(A)$ exists, where $\max(A)$ is the maximum function in terms of the order of integers. The set A is a positive real number well represented by the sum of its integer powers of 2. The non negative integers represent the whole part of the real number, while the negative integers are the fractional part of the real number. This simply means $A\cap\mathbb{N}$ is the whole part of A, and $A\cap-\mathbb{N}$ is the fractional part. The set of all non negative real numbers is $\mathbb{R}_0^+=\bar{\mathbb{Z}}\cup\{\emptyset\}$. Two positive real numbers are order related $A<B$ if and only if $\max(A\triangle B)\in B$. The addition is defined as before by $A\oplus B=(A\triangle B)\oplus s(A\cap B)$. The supremum can also be found in this structure of sets. At this point, negative real numbers can be built using the same technique used to build the negative integers. Every $A\in\mathbb{R}_0^+$ is identified with a function $\oplus A:\mathbb{R}_0^+\to[A,\infty)$. The new set of positive real numbers is the set of bijections $\mathbb{R}_0^+\to[A,\infty)$, for all $A\in\mathbb{R}_0^+$. Negative real numbers are the inverse functions of these. This construction will not be discussed further. A different approach is taken.

An alternative method of building the set of real numbers, $\mathbb{R}$, which depends only on natural numbers is proposed. A construction of the unit interval $(0,1]$ was given, where every number real number in the unit interval was represented as an object in $\mathbb{N}_{inf}^*$. Now, the same set $\mathbb{N}_{inf}^*$ will be used to represent all of the real numbers in a computable coding.

Each of the positive intervals $I_1=(0,1]$, $I_2=(1,2]$,..., and negative intervals $-I_1=(-1,0]$, $-I_2=(-2,-1]$,..., is isomorphic to the interval $(\frac{i}{2^k},\frac{i+1}{2^k}]$, for any $i,k\in\mathbb{N}$. Real number intervals of unit length, I_i, will be compressed into smaller and smaller intervals so as to fit them all in the unit interval as in Figure 5. The interval $I_1\subset\mathbb{R}$ is identified with the interval $(\frac{1}{2},\frac{3}{4}]\subset\mathbb{N}_{inf}$. The interval $I_2\subset\mathbb{R}$ is the interval $(\frac{3}{4},\frac{7}{8}]\subset\mathbb{N}_{inf}$, etc. The negative interval $-I_1\subset\mathbb{R}$ is the interval $(\frac{1}{4},\frac{1}{2}]\subset\mathbb{N}_{inf}$, etc. Let $X\in(\frac{1}{2},\frac{3}{4}]$, then it is an infinite set number such that $1\in X$ and $2\notin X$. A set number $X\in(\frac{3}{4},\frac{7}{8}]$ is an infinite set number such that $1\in X$ and $2\in X$, but $3\notin X$, etc. A set number $X\in(\frac{1}{4},\frac{1}{2}]$ is an infinite set number such that $1\notin X$ and $2\in X$. A set number $X\in(\frac{1}{8},\frac{1}{4}]$

is an infinite set number such that $1, 2 \notin X$ and $3 \in X$, etc. This is easily interpreted in defining the set of all real numbers. Let $X = \{1, 2, 3, \ldots, n, k_1, k_2, k_3, \ldots\} \in \mathbb{N}_{inf}$, where $3 \leq n+2 \leq k_1 < k_2 < k_3 < \cdots$, then X is positive real number. A negative real number is $X = \{n, k_1, k_2, k_3, \ldots\}$ with $3 \leq n+1 \leq k_1 < k_2 < k_3 < \cdots$. This simply means a positive real number with integer part equal to $n-1$ is an infinite set number X with $1, 2, \ldots, n \in X$ and $n+1 \notin X$. A negative real number with integer part equal to $-(n-2)$ is an infinite set number X with $\min(X) = n$. The fractional part will be given by the remaining objects $k_1, k_2, k_3 \ldots$. We can immediately differentiate a positive set number from a negative set number. For example, The set number $\{1, 2, 3, 4, 10, 11, 12, 13, \ldots\}$ is positive with integer part equal to 3. The set number $\{4, 5, 10, 11, 12, 13, \ldots\}$ is negative with integer part equal to -2. The set number $\{1, 2, 6, 7, 8, \ldots\}$ has integer part equal to 1, while $\{6, 8, 9, 10, \ldots\}$ has integer part -4.

The first natural numbers are place holders for identifying the integer part. Let $X \in \mathbb{R}$ an infinite set number and let $\{k_1, k_2, k_3, \ldots\}$ be the fractional part, as just described. The fractional part of X coded as an element of the unit interval is $r^{n+2}(\{k_1, k_2, k_3, \ldots\})$. To code a real number $X \in \mathbb{R}$ as an infinite set number use the first n natural numbers to determine the integer part. Then there are still infinitely many natural numbers left to determine the fractional part. All that has to be done is displace the fractional part $n+2$ places, so that the fractional part and integer part do not interfere. Displacement up, $n+2$ times, is equivalent to applying s^{n+2}. To recover the fractional part, displace the k_i's back $n+2$ times by applying r^{n+2}. A positive real number with integer part n is an infinite set number that contains a string of the first positive integers, and the first integer it leaves out is a cue that the integer part ends there. The integer part is directly determined by the length of the string. A negative real number is an infinite set number that does not contain the first positive integers. For every $X \in \mathbb{R}$, the numbers $\min(X)$ and $\min(X^c)$ are well defined, because of the well ordering principle. Exactly one of these two is equal to 1 and the other is larger than 1. A *positive real number* is an infinite set number with $min(X) = 1$. A *negative real number* is an infinite set number with $min(X^c) = 1$. More specifically, if $0 < X \leq 1$ then $\min(X^c) = 2$, and if $-1 < X \leq 0$ then $\min(X) = 2$. The equality $\min(X^c) = 3$ is equivalent to $1 < X \leq 2$, and $\min(x) = 3$ is equivalent to $-2 < X \leq -1$. If $2 < X \leq 3$ then $\min(X^c) = 4$, and if $-3 < X \leq -2$ then $\min(X) = 4$. In general, $X \in (n-1, n]$ if and only if $\min(X^c) = n+1$, and $X \in (-n, -(n-1)]$ if and only if $\min(X) = n+1$.

For example, the fractional part of π is equal to $.141596\ldots = 2^{-3} + 2^{-6} + 2^{-11} + 2^{-12} + 2^{-13} + 2^{-14} + \cdots$ given by the set $\{3, 6, 11, 12, 13, 14\ldots\}$ Therefore, the numbers π and $-\pi$ are represented by

$$\begin{aligned} \pi &= \{1, 2, 3, 4, 3+5, 6+5, 11+5, 12+5, 13+5, 14+5, \ldots\} \\ &= \{1, 2, 3, 4, 8, 11, 16, 17, 18, 19\ldots\} \\ -\pi &= \{5, 3+5, 6+5, 11+5, 12+5, 13+5, 14+5\ldots\} \\ &= \{5, 8, 11, 16, 17, 18, 19\ldots\} \end{aligned}$$

The set of infinite set numbers $\mathbb{N}^*_{inf}$ is defined as $\mathbb{R}$. Real numbers and natural numbers are different types of sets. A natural number is a finite subset of **HFS**. A real number is an infinite subset of **HFS**. Adequate definitions can be made for addition of real numbers combining addition of natural numbers and addition of defined for the continuum $[0, 1]$.

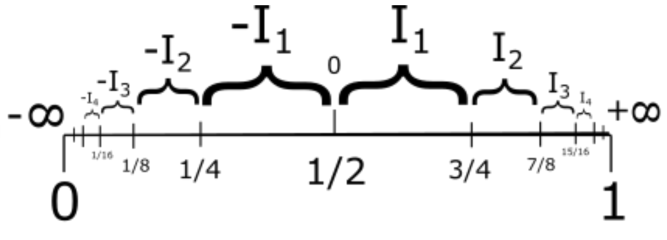

Figure 5: Use the fact that $(0, 1]$ is bijective to any interval $(\frac{n}{2^k}, \frac{n+1}{2^k}]$. Under this representation, the real number $0 \in \mathbb{R}$ is the set $\{2, 3, 4, 5, \ldots\}$. Infinities are the empty set $-\infty = \emptyset$, and the set of positive integers $+\infty = \mathbb{N}_1$.

5.3 Homomorphic Encryption

The science of cryptography is based on the concept principles of relating a message without revealing its contents to unwanted parties. Classical Cryptography attempts to find the optimal solution to the problem of communicating an encrypted message from the first party to a second party, under the restriction that third parties cannot know the contents of the message by any practical means even if the message is intercepted. This is a useful communications scheme in many situations where two parties want to share a message that they wish to remain hidden to any other external actors. We can summarize this model in the following scheme. Participant A wishes to share message M with participant B but nobody else (if a third participant, C, intercepts the message, then he cannot by any practical means know the contents of the message). Notice that Participant B has to know the contents of the message for this scheme to work.

Recent cloud computing applications have called for the implementation of a new type of encryption scheme. For example, you walk into a bank to request a loan. The bank will consult your personal financial data and input it into a program that will make a final decision, based on the output of a mathematical function that will relate your income, expenses, capital, existing debt and a plethora of other personal financial data. In order for the bank to evaluate this mathematical function and make a decision, your data will be exposed to bank employees, management, stock-holders, and other third parties. Now, imagine a scenario where the bank can receive and process all the information needed to make the decision, without ever having to *see the data*. This concept is called Homomorphic Encryption. It is the idea of *computing without knowing*, and its applications to cloud computing are limitless. The mathematical concept of a 'Zero-Knowledge Proof' is the most adequate for describing this communication scheme. It is the process of verifying something without revealing the information that would have to otherwise be revealed.

Two columns represent two numbers. The process and result of adding these numbers is independent of where level 0 is placed, see figures 1 and 3. A true statement holds for vertical displacements of a column representation of addition. This is the conceptual base for a model of FHE because equivalent operations can be carried out in any point of the number line and then later taken back to its original zero-level. The difficulty is to make this scheme sufficiently time and energy efficient, and secure.

5.4 Limits and Continuity

Suitable and practical expressions of the concepts of analysis are proposed. The first concept formalized is *limit point*. Let $P, X \in \mathbb{N}^*_{inf}$ two infinite set numbers. Intuitively, these two objects are close, if their first terms coincide. Take as an example the set numbers $P = \{2, 4, 5, 8, 9, 10, 11, 12, 13, \ldots\} = 2^{-2} + 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-10} + \ldots$ and $X = \{2, 4, 5, 8, 9, 14, 15, 16, 17 \ldots\} = 2^{-2} + 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-14} + \ldots$. They are relatively close because the first terms (the largest terms) coincide. The first elements coinciding is equivalent to the two set numbers being "close". Another way of saying this is that $\min(P \triangle X)$ is a large number. The larger $\min(P \triangle X)$, the larger the elements of $P \triangle X$ become, making the smaller powers (larger terms) coincide. In the part, larger natural numbers represent the smaller terms of the real number.

Let $P \in \mathbb{R}$ an infinite set number, and let $\mathbf{X}$ a set of infinite set numbers. Then, *P is a limit point of $\boldsymbol{X}$* if there exists $X_N \in \mathbf{X}$ such that $\min(P \triangle X_N) > N$, for every $N \in \mathbb{N}$. There is one exception to this definition. The fractional part of some real numbers can be expressed as sum of finite many negative powers of 2. The definition of limit point for these numbers is defined separately. Suppose $P \in \mathbb{N}^*_{inf}$ is an infinite set number that has finite representation $P = \{p_1, p_2, \ldots, p_k\}$, where $p_1 < p_2 < \cdots < p_k$. That is to say, $P = \{p_1, p_2, \ldots, p_{k-1}, p_k + 1, p_k + 2, p_k + 3, \ldots\}$. The last term 2^{-p_k} replaces the terms $2^{-(p_k+1)} + 2^{-(p_k+2)} + 2^{-(p_k+3)} + \ldots$. A set of infinite set numbers is provided such that theses numbers become arbitrarily close to P, from above. Let $X_1 = \{p_1, p_2, \ldots, p_k, p_k + 1\}$, and $X_2 = \{p_1, p_2, \ldots, p_k, p_k + 2\}$. In general define $X_i = \{p_1, p_2, \ldots, p_k, p_k + i\}$. These set numbers X_i are getting closer to P but the previous definition of limit point is not satisfied. The minimum element of the symmetric difference is not getting larger. In fact, it is the constant $\min(P \triangle X_i) = p_k$. Therefore, a different definition is required for this case. Let P an infinite set number with finite representation. The number P is a limit point of $\mathbf{X}$ if for every $N \in \mathbb{N}$ there exists $X_N \in \mathbf{X}$ such that $X_N = \{p_1, p_2, \ldots, p_k, p_k + N, p_{n_1}, p_{n_2}, \ldots\}$, where $p_k + N < p_{n_1} < p_{n_2} < \cdots$. This in turn makes the difference bounded, $|P - X_N| \leq \frac{1}{2^{p_k+N-1}} = \{p_k + N - 1\}$. Let P, X two infinite set numbers with their integer parts equal

and suppose their fractional parts coincide in the first elements,

$$\begin{aligned} P &= 2^{m_1} + 2^{m_2} + \ldots + 2^{m_k} + 2^{n_1} + 2^{n_2} + 2^{n_3} + \ldots + 2^n + 2^{\alpha_1} + 2^{\alpha_2} + 2^{\alpha_3} + \ldots \\ X &= 2^{m_1} + 2^{m_2} + \ldots + 2^{m_k} + 2^{n_1} + 2^{n_2} + 2^{n_3} + \ldots + 2^n + 2^{\beta_1} + 2^{\beta_2} + 2^{\beta_3} + \ldots \end{aligned}$$

where $m_1 < m_2 < \ldots < m_k < n_1 < n_2 < \ldots < n < \alpha_1 < \alpha_2 < \cdots$ and $n < \beta_1 < \beta_2 < \cdots$. The numbers m_i determine the integer part and n_i, n are the elements that coincide in the fractional part (the first negative powers of 2 that coincide). Then, the difference $|P - X| < \frac{1}{2^n}$ is bounded, by $\{n\}$.

Infinite set numbers with finite representations can be handled in another, informal, manner. For simplicity, consider set numbers of $(0, 1] \subseteq \mathbb{N}_{inf}$. For example, $1/2 \in [0, 1]$ has the representations $\{1\} = \{2, 3, 4, \ldots\}$. The set number $P = 1/2$ should be a limit point of the set $\mathbf{X} = \{A_1, A_2, A_3, A_4, \ldots\}$ where the A_i are

$$\begin{aligned} A_1 = &\quad 1 &&= \{1, 2, 3, 4, 5, 6, \ldots\} \\ A_2 = &\quad 3/4 &&= \{1, 3, 4, 5, 6, 7, \ldots\} \\ A_3 = &\quad 5/8 &&= \{1, 4, 5, 6, 7, 8, \ldots\} \\ A_4 = &\quad 9/16 &&= \{1, 5, 6, 7, 8, 9, \ldots\} \\ &\quad \vdots \quad\quad \vdots &&\quad\quad \vdots \end{aligned}$$

If $P = \{2, 3, 4, 5, \ldots\}$ then $\min(P \triangle A_i) = 1$, for every A_i. Using the finite representation $P = \{1\}$, gives the symmetric differences: $P \triangle A_1 = \{2, 3, 4, \ldots\}$, $P \triangle A_2 = \{3, 4, 5, \ldots\}$, $P \triangle A_3 = \{4, 5, 6, \ldots\}$, $P \triangle A_4 = \{5, 6, 7, \ldots\}$,... In effect, satisfying the condition that for every $N \in \mathbb{N}$ there exists $X_N \in \mathbf{X}$ such that $\min(P \triangle X_N) > N$.

If P has finite representation and the set numbers A_i tend to P, from below, the same problem cannot occur. For example, take $P = 1/2 = \{2, 3, 4, 5, \ldots\}$ and the set $\mathbf{X} = \{A_1, A_2, \ldots\}$ defined by

$$\begin{aligned} A_1 = &\quad 3/8 &&= \{2, 4, 5, 6, 7, 8 \ldots\} \\ A_2 = &\quad 7/16 &&= \{2, 3, 5, 6, 7, 8 \ldots\} \\ A_3 = &\quad 15/32 &&= \{2, 3, 4, 6, 7, 8 \ldots\} \\ A_4 = &\quad 31/64 &&= \{2, 3, 4, 5, 7, 8 \ldots\} \\ A_5 = &\quad 63/128 &&= \{2, 3, 4, 5, 6, 8 \ldots\} \\ &\quad \vdots \quad\quad \vdots &&\quad\quad \vdots \end{aligned}$$

It is easily verified that for every $N \in \mathbb{N}$ there exists X_N such that $\min(P \triangle X_N) > N$. The symmetric differences are $P \triangle A_1 = \{3\}$, $P \triangle A_2 = \{4\}$, $P \triangle A_3 = \{5\}$, $P \triangle A_4 = \{6\}$, $P \triangle A_5 = \{7\}, \ldots$.

Continuity is described in terms of the order of natural numbers. In the next section a formal definition for real function is provided. It is used provisionally, for the sake of illustration.

Definition 11. *Let $f : \boldsymbol{A} \subseteq \mathbb{R} \to \boldsymbol{B} \subseteq \mathbb{R}$ a real function, and let p a limit point of the domain $\boldsymbol{A}$. The function f has limit point p, and the limit is equal to q, if and only if for every $N \in \mathbb{N}$ there exists $M \in \mathbb{N}$ such that* $\min(p \triangle x) > M$ *implies* $\min(f(p) \triangle q) > N$.

The function is continuous in p if and only if for every $N \in \mathbb{N}$ there exists $M \in \mathbb{N}$ such that $\min(p \triangle x) > M$ *implies* $\min(f(p) \triangle f(x)) > N$.

The theory of convergence and topological aspects of $\mathbb{R}$ are expressed directly in terms of the order of natural numbers. Using these general indications and the subtraction algorithm, given in [I], it is possible to define the derivative. The derivative can be treated in two ways. The subtraction algorithm allows for the traditional definition of derivative, for finding the numerical value $f'(p)$. However, to prove the *existence* of the derivative, there is an alternative definition of a *discrete derivative*. The quotient of two powers of 2 is obtained by subtracting the powers, $\frac{2^n}{2^m} = 2^{n-m}$. For a fixed $x \in A$ in the domain of f, consider the difference $D_x = \min(fp \triangle fx) - \min(p \triangle x)$. If the derivative, at p, is $f'(p) = 1$ it will have to be true that D_x tends to 0 as x tends to p. If the function has derivative $f'(p) = 0$ it must be true D_x is unbounded as x tends to p. If D_x tends to a positive integer, as x tends to p, then the derivative satisfies the inequality $0 < f'(p) < 1$. The last

case, when D_x tends to a negative number, as x tends to p, means the derivative is $f'(p) > 1$, greater than 1. The fast derivative gives a truncated approximation of the nearest integer power of 2, for the absolute value of the derivative.

The discrete derivative is a criteria for the existence and absolute value of the magnitude of the derivative. In exchange for not knowing the exact numerical value of the derivative, finding the discrete derivative is computationally faster. The quotient $\frac{fp-fx}{p-x}$ of floating point numbers is being substituted with a difference of natural numbers, $\min(fp \triangle fx) - \min(p \triangle x)$. The end result is that instead of having to calculate two subtractions and one division of real numbers, the minimum element for two sets of natural numbers and the difference of these natural numbers is computed.

Alternative constructions of real numbers are found in modern references [A'Campo(2003)],[Arthan(2004)], [De Bruijn(1976)],[Knopfmacher and Knopfmacher(1988)]. These involve complex objects and cumbersome methods for proving the existence of the real number system. The proposed set theory has clear advantages in its simplicity of description of objects and efficient algorithms. That is to say, it stands above other set theories in theoretical and practical terms.

6 Type Theory and Trees

Another application of the proposed set theoretical foundations of numbers is a theory of types that organizes and orders objects of all kinds by coding them in the simplest type possible. An account of representations for general objects of classic mathematics is given. The universe of sets necessary for classic mathematical objects is well represented in terms of trees. A brief description of the theory of types is also outlined in terms of trees.

6.1 Basic Objects In Mathematics

Ordered pairs, and finite sets of ordered pairs, are natural numbers. To define an ordered n-tuple of natural numbers, recall that even and odd natural numbers were used to tell apart the first component from the second. One might initially want to solve in the following manner. To well represent ordered 3-tuples use $\{1, 4, 7, 10, \ldots, 3k-2, \ldots\}$ to represent the first component, then use $\{2, 5, 8, 11, \ldots, 3k-1, \ldots\}$ to represent the second component, and multiples of three, $\{3, 6, 9, 12, \ldots, 3k, \ldots\}$ to represent the third component. This will give a table similar to (8), of ordered pairs. Table 6 allows to describe ordered 3-tuples.

X	$3k-2$	$3k-1$	$3k$
0	1	2	3
1	4	5	6
2	7	8	9
3	10	11	12
4	13	14	15
5	16	17	18
6	19	20	21
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 6: The elements of this table allow us to represent an ordered 3-tuple as a natural number.

The ordered 3-tuple $(0, 0, 0)$ is the set number $2^1 + 2^2 + 2^3 = \{1, 2, 3\}$. Also, $(1, 2, 3)$ is equal to $2^4 + 2^8 + 2^{12} = \{4, 8, 12\}$. To represent 4-tuples, a new Table, 7, is needed.

This manner of defining finite sequences has two big disadvantages that will become clear, when defining a second method for representing ordered n-tuples. The first is quite obvious: it is not possible to define an infinite sequence of natural numbers. The easiest way to solve this is by going back to the definition of ordered pairs. The sets given in (8) are of great importance in the constructions of this section. It is shown again, for reference in Table 8. Here it is used differently. Only the first two rows are needed to define ordered pairs; it will only be necessary to use the first two sets $(0,)$ and $(1,)$. The pair (i, j) will be a set of two numbers; its elements will be the $i+1$-th object of the first row and the $j+1$-th object of the second row.

X	$4k-3$	$4k-2$	$4k-1$	$4k$
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16
4	17	18	19	20
5	21	22	23	24
6	25	26	27	28
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 7: The elements of this table allows to represent an ordered 4-tuple as a natural number.

6	18	66	258	1026	...	$2+2^{2(n+1)}$	...
12	24	72	264	1032	...	$8+2^{2(n+1)}$	...
36	48	96	288	1056	...	$32+2^{2(n+1)}$	...
128	144	192	382	1152	...	$128+2^{2(n+1)}$	...
516	528	576	768	1536	...	$512+2^{2(n+1)}$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	
$2^{2m+1}+4$	$2^{2m+1}+16$	$2^{2m+1}+64$	$2^{2m+1}+256$	$2^{2m+1}+1024$	...	$2^{2m+1}+2^{2(n+1)}$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	

Table 8: The elements of this table allow to represent an ordered pair as a natural number. The elements of the first row are used to represent the first component, while the elements of the second row are used to represent the second component. This table is also used for finding a good representation of sequences of real numbers.

A definition of ordered pair is given, that supersedes the one given before. An ordered pair (i,j) is the set number

$$(i,j) = \{\{1, 2(i+1)\}, \{3, 2(j+1)\}\} = \{2^1 + 2^{2(i+1)}, 2^3 + 2^{2(j+1)}\}. \tag{13}$$

The $i+1$-st element of $(0,)$ is included to show that i is in the first component. Include the $j+1$-st element of $(1,)$ to indicate j is in the second component. For example, the ordered pair $(0,0)$ is the set number $2^6+2^{12} = \{6, 12\}$. The ordered pair $(1,3)$ is $2^{18} + 2^{264} = \{18, 264\}$.

Data types are being defined for different mathematical objects. Different kinds of mathematical objects and relations can be represented as natural and real numbers. An extension of this new representation allows to define *infinite sequence of natural numbers*. Select one element, n_k, from the set $(k,)$, for every $k \in \mathbb{N}$. Then, $2^1 + 2^{2(n_1+1)} \in S$ means n_1 is the first natural number of the sequence. The second number is given by $2^3 + 2^{2(n_2+1)} \in S$, and so on. The set number $\{2^1 + 2^{2(n_1+1)}, 2^3 + 2^{2(n_2+1)}, 2^5 + 2^{2(n_3+1)}, \ldots\}$ represents the sequence $(n_1, n_2, n_3, \ldots)$. For example, the sequence $(1, 3, 2, 5, 4, \ldots)$ is given by

$$\{2 + 2^{2(1+1)}, 8 + 2^{2(3+1)}, 32 + 2^{2(2+1)}, 128 + 2^{2(5+1)}, 512 + 2^{2(4+1)}, \ldots\} = \{18, 264, 96, 4224, 1536, \ldots\}.$$

Of course, to define a finite sequence, a k-tuple, use the first k sets, $(1,), (2,), \ldots (k,)$. A finite sequence of natural numbers is a set number of the form

$$\{2^1 + 2^{2(n_1+1)}, 2^3 + 2^{2(n_2+1)}, 2^5 + 2^{2(n_3+1)}, \ldots, 2^{2k+1} + 2^{2(n_k+1)}\}.$$

A *natural function*, $\mathbb{N} \to \mathbb{N}$, is an infinite set of ordered pairs. A function of this form is a set number

$$\{\{6, B_1\}, \{18, B_2\}, \{66, B_3\}, \{258, B_4\}, \ldots\}$$

where B_i are elements of $(1,)$. If the B_i are all distinct, the function is an injection. If every element of $(1,)$ has an index B_i, the function is onto $\mathbb{N}$. This represents natural functions as real numbers. There exists a bijective function from the set of all natural functions, onto a proper subset of real numbers. Of course, an

infinite sequence of natural and a natural function are two representations of essentially the same thing. This same method of defining a natural function can be used to provide provide a valid construction of an infinite sequence of natural numbers, using the original definition of ordered pair.

How can a *sequence of real numbers* be represented? The same question stated differently, How can a (finite or infinite) sequence of infinite set numbers be well defined? It would be advantageous to find a way of storing and rescuing the information that determines a sequence $\xi = (r_1, r_2, r_3, \ldots)$ where each $r_i = \{n_1^i, n_2^i, n_3^i, \ldots\}$ is a real number. Use the set $(0,)$ to represent the elements of r_1. Use the set $(1,)$ to represent the elements of r_2, etc. Then, $2^{2(i)+1}+2^{2(n_j^i+1)} \in \xi$ if and only if $n_j^i \in r_i$. The infinite sequence of real numbers, $(r_1, r_2, \ldots)$, is represented by the real number $\bigcup_i r_i$. The union of all the r_i's is a real number that represents the infinite sequence $(r_1, r_2, \ldots)$; it is an infinite set number with infinitely many objects from each set $(i,)$. Actually, any set number with infinitely many elements of each $(i,)$ is representing a unique sequence of real numbers. An infinite set $X \subset (0,)$ determines a real number. The set $X = \{2+2^{2(x_1+1)}, 2+2^{2(x_2+1)}, 2+2^{2(x_3+1)}, \ldots\}$ is the coding of the real number $X^* = \{x_1, x_2, x_3, \ldots\}$. In the same manner, $Y = \{8+2^{2(y_1+1)}, 8+2^{2(y_2+1)}, 8+2^{2(y_3+1)}, \ldots\} \subset (1,)$ is the coding of $Y^* = \{y_1, y_2, y_3, \ldots\}$. The infinite set number $X \cup Y$ is a real number, whose objects are in $(0,) \cup (1,)$, and the objects of $(0,)$ are distinguishable from the objects of $(1,)$. The objects in $(0,)$ give the first component, and the second component is given by the elements of $(1,)$. This provides a good representation of the ordered pair of real numbers, (X^*, Y^*), as a single real number $X \cup Y$. To represent ordered 3-tuples of real numbers, use the set $(2,)$, also. Let $Z^* = \{z_1, z_2, z_3, \ldots\} \subset \mathbb{N}$ a real number, then $Z = \{32+2^{2(z_1+1)}, 32+2^{2(z_2+1)}, 32+2^{2(z_3+1)}, \ldots\} \subset (2,)$. And, the ordered 3-tuple (X^*, Y^*, Z^*) is the real number $X \cup Y \cup Z$. An infinite sequence of real numbers $(r_1, r_2, r_3, \ldots)$ is represented by a single real number. A bijective function from the set of all real sequences onto a proper subset of real numbers has been described. A sequence of real numbers is well represented by a single real number. And, it has also been shown that a function $\mathbb{N} \to \mathbb{N}$ is well represented by a real number. Consequently, a *sequence* $(f_1, f_2, \ldots)$*, of functions* $f_i : \mathbb{N} \to \mathbb{N}$, can be represented as a single real number. In summary, a second definition for ordered pairs is given, that is a more powerful definition than the first because it allows to represent an infinite sequence of natural numbers, as a real number. Moreover, if ξ is a countable sequence of real numbers, it is also represented as a real number. Therefore, a good representation of functions $\mathbb{N} \to \mathbb{N}$, and sequences of these functions, is obtained.

A coding of sequences of sequences can also be described. Consider first the simplest kind, a *sequence* $T = (S_1, S_2, \ldots)$ *of sequences,* S_i*, of natural numbers.* Every infinite sequence of natural numbers can be coded as a real number and every sequence of real numbers can be coded as a real number, it follows that a sequence of sequences of natural numbers can be coded as a single real number.

Now, let $\xi_i = (r_1^i, r_2^i, r_3^i, \ldots)$ a sequence of real numbers, for every $i \in \mathbb{N}$, and let $\Xi = (\xi_1, \xi_2, \xi_3, \ldots)$ the sequence of those. It is easy to construct a real number representing this object. This is true because every sequence ξ_i, of real numbers, is represented by a real number. The sequence of real numbers, Ξ, can in turn be reduced to a single real number. A real matrix of infinitely (countable) many columns and rows can be represented by a single real number.

There are more similarities between natural numbers and real numbers. A natural function is a set of natural numbers and each element of the set coded a component $n \to fn$ of the function. Similarly, a real function will be coded by a set of real numbers, and each real number of the set will be coding a component $x \to fx$ of the real function. An ordered pair of real numbers has been defined as a single real number, now a *real function* can be defined. A function is a collection of components $f_x = (x, fx)$, and every ordered pair of real numbers $f_x \in \mathbb{R}$ is a real number. Therefore, the function $f : \mathbb{R} \to \mathbb{R}$ can be represented by a set of real numbers $\{f_x\}_{x \in \mathbb{R}}$. Every real function $\mathbb{R} \to \mathbb{R}$ is the set of real numbers

$$f = \{\{a_1^x, a_2^x, \ldots, b_1^x, b_2^x, \ldots\}\}_{x \in \mathbb{R}},$$

where $x = \{a_i^x\}_i \subset (0,)$ and $f(x) = \{b_i^x\}_i \subset (1,)$. This means $f_x = x \cup f(x) = \{a_1^x, a_2^x, \ldots, b_1^x, b_2^x, \ldots\}$. The function is one-to-one if $f(x) \triangle f(y) \neq \emptyset$ for $x \neq y$. The function f is onto $\mathbb{R}$ if for every infinite subset $A \subset (1,)$, there exists an object $x \in \mathbb{R}$ such that $A = f_x \cap (1,)$. A real function is bijective if for every infinite subset $A \subset (1,)$ there exists exactly one $x \in \mathbb{R}$ such that $A = f_x \cap (1,)$.

Extending previous results, any sequence of real functions, $(f_1, f_2, \ldots)$, is a set of real numbers. Just as $(0,)$ and $(1,)$ are used to define a function $f_1 : \mathbb{R} \to \mathbb{R}$, the set $(2,)$ and $(3,)$ can be used to define a function $f_2 : \mathbb{R} \to \mathbb{R}$. In the same way $(4,)$ and $(5,)$ are used to define a function $f_3 : \mathbb{R} \to \mathbb{R}$, etc.

There is another consequence of coding a real function $\mathbb{R} \to \mathbb{R}$ as a set of real numbers. A real operation $\mathbb{R} \to (\mathbb{R} \to \mathbb{R})$ can be coded as a set of real numbers. An object $\mathbb{R} \to (\mathbb{R} \to (\mathbb{R} \to \cdots (\mathbb{R} \to \mathbb{R}) \cdots))$ is coded as a

set of real numbers for finite iterations of the image. In the next subsection, type theory is described using trees.

6.2 Trees

It has been shown that natural numbers are the finite sets that can be built recursively with the function $\oplus 1$. These sets can be well represented by finite tree structures. Trees are used to represent natural numbers first, then all types of objects. Sets and trees are equivalent. A finite set number is a set of finitely many smaller set numbers. The definition of trees is equivalent. A tree is a *trunk* (the principle node); the set X. Every branch of the trunk is an element of the set X. For example, a single trunk with no branches is the set number 0. Suppose the tree of X is known. How is the tree corresponding to $X \oplus 1$ found? Add a branch that is a 0-tree (add 1 unit). The set number 1 is a trunk with one 0-branch; $1 = \{0\}$. This in illustrated in Figure 6.

A tree is a graph of nodes and edges such that (i) A *trunk* can be identified: a principle edge with a finite number of *branches* attached to one of its nodes. All branches are attached to the same node of the trunk. (ii) Each branch on the tree is a tree. (iii) A single edge is a tree; the 0-tree. The successor of a tree is obtained by adding a single edge to the trunk; attach a 0-tree to the trunk. Adding an edge to the 0-tree gives its successor, the 1-tree, which is two edges joined together at one node. Adding an edge to the 1-tree, yields its successor, the 2-tree, etc.

An extra rule for defining an equivalence class on finite trees is needed. If a tree has two identical branches, substitute these two identical branches with a single branch, the successor. This process is called *reduction*. If a tree can be reduced to obtain another tree, they are in the same equivalence class. An irreducible tree is said to be in canonical form. Reducing the 2-tree, gives the canonical form. To reduce the 2-tree, substitute the two identical 0-trees with a single 1-tree. Adding a single edge to the result of that, results in the canonical form of the 3-tree because it contains no identical branches. If an edge is added to the 3-tree, reduction of branches will have to be applied two times before reaching the canonical form of the 4-tree. First take away the two 0-trees and add a 1-tree. But, there is already another 1-tree; there are two identical 1-trees. Replace those trees with a single 2-tree. Every natural number is associated an equivalence class of finite trees, and a single canonical tree. Every branch on the canonical tree of a set number X corresponds to a natural number $k \in X$. Every tree is made up of smaller trees, and a well defined method of building trees is provided. The canonical tree associated to the set number X, has $\#(X)$ many branches. Each branch is defined in the same way. A natural number is defined by its cardinality; and the cardinality of its elements; and the cardinality of the elements of its' elements; etc. Trees are used to represent real numbers, also. A real number is a tree with infinitely many different branches, each branch a natural number. A set of real numbers is a tree with infinitely many branches, each a real number. The next subsection is a formalization of the concept of types. The concept of number is generalized. A number is a tree, and every mathematical object is a number in this sense.

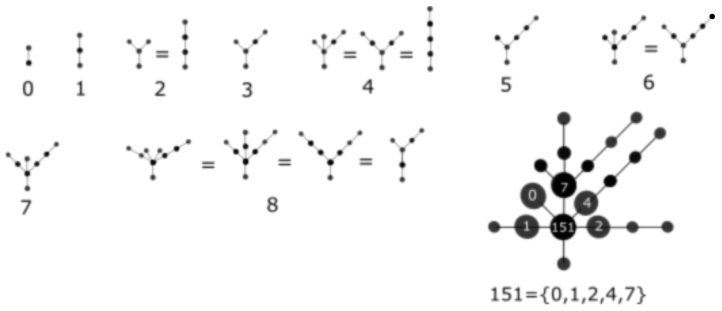

Figure 6: Canonical trees can be built easily, given a set number. The tree representation of $6 = \{1, 2\}$ is a tree with two branches; a 1-tree and a 2-tree. The canonical tree for $7 = \{0, 1, 2\}$ has three branches. One branch is the 0-tree, the second branch is the 1-tree and the third branch is the 2-tree. The canonical tree of $8 = \{3\}$ is a trunk with one branch, which is the 3-tree. The canonical tree of $151 = \{0, 1, 2, 4, 7\}$ has five branches: $0, 1, 2, 4, 7$-trees.

6.3 Type Theory

Finite trees are *objects of Type-0*. Trees of infinite branches with each branch being an object of type-0 are called *objects of Type-1*. For example, a natural number is an object of Type-0 and a real number is an object of Type-1. A tree whose branches are all objects of Type-1 is an *object of Type-2*. An example of an object of Type-2 is a set of real numbers.

Use the Replacement Axiom to build a tree with branches of Type-0 and Type-1. An object with elements of these two types is an *object of Type-3*. A set consisting of natural and real numbers is an object of Type-3. The power set of a Type-2 object is an object of Type-4; a tree whose branches are Type-2 objects. A set of sets of real numbers is an *object of Type-4*. The set of integers $\mathbb{Z}$ is an object of Type-2, while the set of positive real numbers $\bar{\mathbb{Z}}$ is an object of Type-4. That is why it can be better to code the real numbers as infinite set numbers; infinite set numbers are objects of Type 1. It will ultimately depend, of course, of the application.

Define Type-n objects in a manner analogous to the definition of natural numbers. The power set axiom is required for the existence of $P(\mathbb{N}), P(P(\mathbb{N})), P(P(P(\mathbb{N}))), P(P(P(P(\mathbb{N})))), \ldots, A, P(A), \ldots$ which are sets of type $2, 4, 16, 2^{16}, \ldots, n, 2^n, \ldots$, respectively. A Type-8 object is a tree whose branches are all Type-3 objects. The power set of a Type-4 object is a Type-16 object, etc. Other types are found using the replacement axiom to combine subsets of these power sets. A Type-7 object consists of objects of three different types, $0, 1, 2$. In Section 6, real sequences are coded as Type-1 objects, and real functions are coded as Type-2 objects.

The next step in classifying types is to consider trees with infinite many types of branches. A tree with branches of Type-n_1, Type-n_2, Type-$n_3 \ldots$ for infinite many types is called an *object of infinite Type-1,0*. An *object of infinite Type-1,1* is a tree that only has branches of infinite Type-1,0. An *object of Type-1,2* is a tree that has only branches of Type-1,1. A tree with branches of both Type-1,0 and infinite Type-1,1 is an *object of infinite Type-1,3*. If all the branches of a tree are objects of Type-1,2, it is an *object of infinite Type-1,4*. All infinite Types-$1, k$ are constructed in a manner analogous to natural numbers.

Consider a tree whose branches are all objects of infinite type; the elements of the tree are objects of Type-$1, n_1$, Type-$1, n_2$, Type-$1, n_3, \ldots$ for infinitely many infinite types. This tree is an object of infinite Type-$2, 0$. Trees whose objects are only objects of Type-$2, 0$ are called *objects of Type*-$2, 1$. A tree whose objects are all of Type-$2, 1$ is an *object of Type*-$2, 2$. A tree with objects of Type-$2, 0$ and Type-$2, 1$ is an *object of Type*-$2, 3$, etc.

An object of infinite Type-$3, 0$ is a tree that has branches of finite type and infinite Type-$1, k$. A tree with objects of Type-3,0 is an object of Type-3,1, and so on. An object of infinite Type-4,0 is a tree with infinite many types of objects of Type-$2, k$. An object of Type-$4, 0$ has objects of Type-$2, n_1$, Type-$2, n_2$, Type-$2, n_3 \ldots$ for infinite many Type-$2, k$ objects. An object of Type-$4, 1$ is a tree with branches of Type-$4, 0$, etc. An infinite Type-5,0 object consists of branches of finite type and infinite types $2, k$. A Type-$6, 0$ object consists of objects of types $1, k$ and types $2, k$, etc. Continue in this manner until all objects of Type-m, n, for every $m, n \in \mathbb{N}$, have been described. Higher hierarchy types are left for future analysis.

7 Conclusions

The importance of the axiomatic base is usually undermined because it does not bring any new results or methods into most practical areas of mathematics. Instead, the axiomatic base of mathematics is seen as a *stone in the path*; an obstacle to be dealt with and forgotten. The natural number system proposed allows for natural constructions of classic structures of mathematics. Finite groups are described using natural numbers. Finding all finite groups of n objects is still not trivial but a better notion of attacking this problem is acquired. A minimum set of independent equations that defines each group is obtained in the process. Two groups are isomorphic if their canonical block forms are identical. The set of all finite groups is totally and linearly ordered. This linear order on finite groups is well behaved with respect to cardinality and other aspects. In particular, the commutative group $\mathbb{Z}_n$ is the smallest group of n objects; $\mathbb{Z}_n < G$ for every group G such that $|G| = n$. If $n = p_1^{n_1} p_2^{n_2} p_3^{n_3} \cdots p_k^{n_k}$ is the prime factorization of n, then the commutative group $\mathbb{Z}_{p_1}^{n_1} \oplus \mathbb{Z}_{p_2}^{n_2} \oplus \mathbb{Z}_{p_3}^{n_3} \oplus \cdots \oplus \mathbb{Z}_{p_k}^{n_k}$ is the largest commutative group of n objects. This last behavior was not treated with detail, and is left for future work. Finite groups are also ordered internally. The elements of any finite group are ordered through the canonical naming functions. A criteria for defining equivalent objects of a fixed finite group is obtained, that provides the automorphisms of the group. The set theory for natural numbers was extended to describe infinite mathematical objects such as real numbers, real functions, real valued matrices, sets of real numbers, and structures derived from those, etc. Results pursued in future work can include a thorough description of groups,

rings, fields and linear spaces, in the finite and infinite cases separately. Another line of work will include a more comprehensive description of the calculus of real numbers. The theory of types and the Continuum Hypothesis can be considered for future work. There are a variety of ways for coding the information of mathematical structures. Natural data types for the basic structures have been provided, although this library of types must be completed. Trees are used to represent any type of mathematical object. The general procedure for expressing mathematical objects using the smallest type possible is described.

The computational aspects can also be treated with detail, focusing on physical models to represent the arithmetic of Energy Levels. In [Magidor], the author mentions the possibility that *"...we will be able to compare between different Set Theories according to what type of mathematical hinterland they provide for theoretical Physics."* Aside from classic computational schemes that can be improved, such as the one proposed for a simple and linear fast adder, modern computational schemes can also be explored. Encoding and storing mathematical objects (structures of information), is an option to be considered for future work. On the other hand, the linear sum of two waves, in phase, with equal wavelength and frequency, is equal a wave with double the amplitude. The linear superposition of constructive interference from two coherent sources satisfies the numeric principle for addition, $2^n + 2^n = 2^{n+1}$. Thus, measurements on the amplitude of waves can be used as a computational arithmetic model. This could provide a valid approach, for a linear optical computing scheme. Most recently, in [Miscuglio(2020)], it has been noted that *"...the wave nature of light and related inherent operations such as interference and diffraction, can play a major role in enhancing computational throughput..."* And that *"In this view, photons are an ideal match for computing node-distributed networks."* An implementation of the finite-state machine of addition can be a system of coherent wave sources.

References

[Corry(2010)] Leo Corry. David Hilbert and the Axiomatization of Physics (1898–1918): From Grundlagen der Geometrie to Grundlagen der Physik. *Springer Netherlands,* ***2010***

[Benacerraf(1965)] Benacerraf, Paul. What Numbers Could Not Be; *Philos. Rev.* ***1965****, 74.*

[Thiele(2003)] Rüdiger Thiele. Hilbert's Twenty-Fourth Problem. *The American Mathematical Monthly, 110:1, 1-24,* ***2003****.* DOI: 10.1080/00029890.2003.11919933

[Ramirez(2019)] Ramírez, J.P. A New Set Theory for Analysis; *Axioms* ***2019****, 8, 31.*

[Uma(2012)] R. Uma, Vidya Vijayan, M. Mohanapriya, Sharon Paul. 2012. Area, Delay and Power Comparison of Adder Topologies. *International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1,* ***February 2012****.*

[Singh(2009)] R.P.P. Singh, Parveen Kumar, Balwinder Singh. Performance Analysis Of Fast Adders Using VHDL. *2009 International Conference on Advances in Recent Technologies in Communication and Computing. IEEE Computer Society.*

[Lutz(1994)] D. R. Lutz, D. N. Jayasimha. The Power of Carry Save Addition. *Department of Computer and Information Science, The Ohio State University.* ***1994****.*

[Sun] Yiqiu Sun, Haichao Yang, et. al. ASIC Design for Bitcoin Mining. *University of Michigan.*

[Wang(2023)] Chenyu Wang, Ge Shi, Fei Qiao, Rubin Lin, Shien Wu and Zenan Hu. Research Progress in Architecture and Application of RRAM with Computing-In-Memory. *Nanoscale Adv.,* ***2023****, 5, 1559-1573.*

[Hennessy(1990)] Hennessy, J.L. and Patterson, D.A. Computer Architecture: A Quantitative Approach.*Morgan Kaufmann, Waltham.****1990****.*

[Lovyagin(2021)] Lovyagin, Yuri N., and Lovyagin, Nikita Yu. Finite Arithmetic Axiomatization for the Basis of Hyperrational Non-Standard Analysis; *Axioms 10, no. 4: 263.* ***2021****.* https://doi.org/10.3390/axioms10040263

[Bernays(1991)] Bernays, Paul. Axiomatic Set Theory; *Dover: New York, NY, USA,* ***1991****.*

[Ackermann(1937)] Ackermann, W. Die Widerspruchsfreiheit der allgemeinen Mengenlehre. *Math. Ann. 114, 305–315.*

[Ladner and Fischer(1980)] R. E. Ladner and M. J. Fischer. Parallel Prefix Computation; *Journal of the ACM, 27(4), pp. 831-838, October* ***1980****.*

[Metropolis, Rota and Tanny(1980)] Metropolis, N.; Rota, G.C.; Tanny, S. Significance Arithmetic: The Carrying Algorithm; *Journal of Combinatorial Theory, Series A,* ***1973****, 14, 386–421.*

[Abrar(2019)] Abrar, M., Elahi, H., Ahmad, B.A. et al. An area-optimized N-bit multiplication technique using N/2-bit multiplication algorithm. *SN Appl. Sci. 1, 1348* ***(2019)****.*

https://doi.org/10.1007/s42452-019-1367-6

[Emmart(2011)] Niall Emmart and Charles C. Weems. High Precision Integer Multiplication with a GPU Using Strassen's Algorithm with Multiple FFT Sizes. *Parallel Processing Letters, Vol.21, No. 03, pp. 359-375* ***(2011)****.* https://doi.org/10.1142/S0129626411000266

[Taib(2020)] Muhammad Ikmal Mohd Taib, Muhammad Najmi Zikry Nazri, et. al (2020). Design of Multiplication and Division Operation for 16 Bit Arithmetic Logic Unit (ALU). *JOURNAL OF ELECTRONIC VOLTAGE AND APPLICATION VOL. 1 NO. 2* ***(2020),*** *46-54.* DOI: https://doi.org/10.30880/jeva.2020.01.02.006

[Rahman(2013)] Mohammed Ziaur Rahman. Parallel Self-Timer Adder (PASTA). *United States Patent Application,* ***May 9, 2013****.*

[Franklin(1994)] Franklin, M.A. and Pan, T. (1994) Performance Comparison of Asynchronous Adders. *Proceedings of IEEE Symposium on Advanced Research in Asynchronous Circuits and Systems, Salt Lake City, 3-5 November* ***1994****, 117-125.* https://doi.org/10.1109/ASYNC.1994.656299

[Zhang(2013)] Ting Zhang, Cheng Xu, Tao Li, Yunchuan Qin and Min Nie. An Optimized Floating-Point Matrix Multiplication on FPGA. *Information Technology Journal, 12:* ***2013*** *1832-1838.* DOI: 10.3923/itj.2013.1832.1838

[A'Campo(2003)] A'Campo, N. A Natural Construction for the Real Numbers. *arXiv,* ***2003****;* arXiv:math.GN/0301015 v1.

[Arthan(2004)] Arthan, R.D. The Eudoxus Real Numbers. *arXiv,* ***2004****;* arXiv:math/0405454.

[De Bruijn(1976)] De Bruijn, N.G. Definig Reals Without the Use of Rationals; *Koninkl. Nederl. Akademie Van Wetenschappen: Amsterdam, The Netherlands,* ***1976****.*

[Knopfmacher and Knopfmacher(1988)] Knopfmacher, A.; Knopfmacher, J. Two Concrete New Constructions of the Real Numbers. *Rocky Mt. J. Math.* ***1988****, 18, 813–824.*

[Magidor] Magidor, Menachem. Some Set Theories are More Equal. Preliminary Draft.

[Miscuglio(2020)] Miscuglio, Mario. Photonic Tensor Cores for Machine Learning. *Appl. Phys. Rev.* 7, 031404 (2020); https://doi.org/10.1063/5.0001942

Funding

This research received no external funding.

Conflicts of Interest

The author declares no conflict of interest.

Acknowledgments

Special Thanks to my Professors at undergraduate school. I am specifically thankful to Ms. Sofia Ortega Castillo who has helped me to prepare and organize this material, for talks given at the 52nd National Congress of Mathematics (2019, Monterrey, México), and encouraged me to pursue publication of the material. I am indebted to my professor in group theory, Alonso Castillo Ramírez who has assisted me with all kinds of questions that came up during the time I wrote some of the details on groups. And, to my professor in analysis and probability theory, Victor Pérez Abreu Carreón who has always been a great teacher and friend, whose conversations and classes have inspired a great deal of the work I have tried to carry out. Any corrections or changes to be made are sole responsibility of the author.

A Simple and Linear Fast Adder (Patent Pending)

Disclosed herein is a fast adder design based on a novel axiomatization of mathematics, of natural and real numbers, by the author. Addition is a Finite State Machine that, on an average, takes $\log_2 n$ iterations to calculate a n-bit addition. Further, for the proposed fast adder, the probability of a n-bit addition taking $k \leq n$ iterations to complete, is equal to the probability of k consecutive heads in n fair coin tosses. The circuitry is linear and simple, in the sense that adding bits to the inputs does not complicate the circuit topology. The growth is linear, and the instruction set is constant, and hardware based. The Figures presented in this sections are numbered 1-11, and should not be confused with the figures of the previous sections which were numbered 1-6. The figures pertaining the patent, will be referenced by the abbreviation "FIG.", followed by the number of the figure.

BACKGROUND

[0001] The subject matter of the present invention is related to a general-purpose fast adder, which is designed in the form of a sequential logic circuit. Particularly, the present invention proposes a fast adder defined in terms of a finite state machine that replaces traditional carry-over algorithms of addition, based on a novel axiomatization of mathematics, by the author. The adder constitutes a direct application of this foundation of mathematics which serves as supporting material for several aspects, including further applications, of the Simple and Linear Fast Adder.

[0002] Efficient and inexpensive Central Processing Units (CPUs) or processing units with low dissipation are an ever growing priority. One of the crucial subunits of the CPUs is an Arithmetic Logic Unit (ALU). Typically, the ALU is responsible for performing the actual arithmetic and logical operations in the CPUs. The efficiency and performance of the ALU generally depends on specific components of the ALU, namely, the adder and the bit shift component.

[0003] One of the basic problems with an existing adder, such as a Ripple Carry Adder, is the propagation delay. A traditional solution to overcome this is to use a parallel adder. Further, other solutions such as a Carry Look-Ahead (CLA), Carry Select, Carry Skip, and Carry Increment adders face their own problems. For example, in the case of CLA, if the number of bits is increased, the area and complexity of the circuit increases considerably. Therefore, the CLA fast adders of more than four bits are generally built using parallel 4-bit adders. This multi-level structure adds up to the propagation time delays.

[0004] In view of the above limitations in the existing adders, it would be advantageous to have an adder that offers linear growth and complexity irrespective of the increase in the number of bits.

[0005] The information disclosed in this background of the disclosure section is only for enhancement of understanding of the general background of the invention and should not be taken as an acknowledgement or any form of suggestion that this information forms the prior art already known to a person skilled in the art.

SUMMARY

[0006] It is an objective of the present invention to provide a general-purpose fast adder having a small count of 'AND' and 'XOR' logic gates, setting a new standard in the design and manufacture of ALU by providing efficiency that is comparable to parallel adders, while having a reduced material, production, and energy costs.

[0007] It is a further objective of the invention to design a fast adder that is implemented based on an arithmetic and real number model and which can be implemented for operation on signed and rational approximations to real numbers, with a few minor modifications.

[0008] It is a further objective of the invention to provide a universal fast adder of linear area, with logarithmic time delay.

[0009] In view of the foregoing, an embodiment of the present disclosure relates a general purpose fast adder

that is in the form of a sequential logic circuit, based on a finite state machine that is not time constant. On an average, it takes $\log_2 n$ iterations to complete addition of two n-bit numbers. The proposed fast adder has the advantages of linear growth and complexity, in the sense that adding one bit of input requires adding a subunit consisting of four registers and five logical gates, and the subunits are connected in series. The instruction set does not increase when the number of bits is increased. The performance of the adder is potentially comparable to the existing fast adders, while using five logical gates (one XOR, and four AND) and four registers of memory, per bit of input. In an implementation according to the present invention, the four bit adder presented here uses sixteen AND gates, four XOR gates and sixteen one bit registers. This adder has linear area and complexity, and logarithmic delay. The power dissipation of the adder is theoretically constant, due to constant gate depth. Instruction set is also constant and independent of the number of bits of input.

[0010] In an implementation, the proposed invention is flexible and compatible with different signed representations. The proposed ALU architecture is able to support operands for integer and rational approximations to real numbers. As an example, the operations can include, without limiting to, left/right shift (multiplication/division by 2), addition, signed operations, and other operations derived thereof.

[0011] In an embodiment of the present disclosure, the four-bit adder component is configured to support integer type data and rational approximations to real number type data.

[0012] In another embodiment of the present disclosure, the four-bit adder component is configured to perform operations comprising at least one of left shift operation, right shift operation, addition, signed operations and one or more derived operations. In an embodiment, performing the operations comprises representing the numbers in a binary form in corresponding set of natural numbers, such that, each number is a set of smaller natural numbers, wherein elements of the set of smaller numbers are denoted in powers of 2 in a binary representation.

[0013] In another embodiment of the present disclosure, the linear fast adder comprises determining a symmetric difference corresponding to the operations performed at the four-bit adder component, the determining comprising saving an initial state of the operations in at least one one-bit registers in the four-bit adder component, directing output of each of the one-bit registers in two disjoint paths and computing the symmetric difference and intersection in the output of each of the one-bit registers. In an embodiment, the bit configurations saved in the one-bit registers are passed through at least one XOR gate in the four-bit adder component for yielding the symmetric difference. The bit configurations saved in the one-bit registers are passed through at least one AND gate in the four-bit adder component for determining intersection in the output.

[0014] In another embodiment of the present disclosure, to represent a rational approximation of non-negative real numbers, a fraction of the bits is used for the rational part and the remaining bits are used for the integer part.

[0015] In another embodiment of the present disclosure, adding a single bit to the operands requires adding of a sub-unit of four bits and five logic gates in a linear manner to the four-bit adder component.

[0016] In another embodiment of the present disclosure, the time taken by the linear fast adder is equal to the sum of the two gate delays, and the reading and writing process.

[0017] In another embodiment of the present disclosure, clock cycles for the linear fast adder remain shorter depending on the gate depth and constant instructions, such that an increase in speed of memory writing process results in a compounded reduction of time.

[0018] In another embodiment of the present disclosure, an instruction set associated with the linear fast adder is constant and is independent of the number of bits of input provided to the linear fast adder.

[0019] In another embodiment of the present disclosure, the operation of the linear fast adder is controlled based on an arithmetic model that defines addition operations in terms of a finite state machine. Here, each state of the finite state machine comprises two columns and each column represents a finite configuration of energy levels representing one natural number. In a subsequent state of the finite state machine, the finite configuration

on the left column of the two columns represents the energy levels that are not repeated in the preceding state and the finite configuration on a right column of the two columns represents objects that are repeated from the preceding state.

[0020] The foregoing summary is illustrative only and is not intended to be in any way limiting. In addition to the illustrative aspects, embodiments, and features described above, further aspects, embodiments, and features will become apparent by reference to the drawings and the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, explain the disclosed principles. In the figures, the leftmost digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and regarding the accompanying figures, in which:

[0022] FIGURE 1 shows a graphical representation of an exemplary operation $(15 + 23 = 38)$, in accordance with some embodiments of the present disclosure.

[0023] FIGURE 2 shows an external view of one-bit data register, in accordance with some embodiments of the present disclosure.

[0024] FIGURE 3 illustrates use of exemplary registers RA/RA' and RB/RB' with input "i", output "o" and logic gates, in accordance with some embodiments of the present disclosure.

[0025] FIGURE 4 shows a full view of a four bit adder along with "enable" and "set" and connections to the control unit, in accordance with some embodiments of the present disclosure.

[0026] FIGURE 5 shows a flow diagram for the instruction set, where the instruction set is constant and independent of the bit length of inputs, in accordance with some embodiments of the present disclosure.

[0027] FIGURE 6 shows addition of rational and real numbers as an extension of the natural number arithmetic proposed, in accordance with some embodiments of the present disclosure.

[0028] FIGURE 7 shows a complete structure of the fast adder, compatible with multiplication and division, in accordance with some embodiments of the present disclosure.

[0029] FIGURE 8 shows an exemplary control unit and its internal parts, in accordance with some embodiments of the present disclosure.

[0030] FIGURE 9 shows an alternative arrangement of the fast adder including double edge triggered flip flops, in accordance with some embodiments of the present disclosure.

[0031] FIGURE 10 shows a basic subunit for the fast adder, such that one of these subunits handles one bit of input and is connected in series to give an n-bit adder, in accordance with some embodiments of the present disclosure.

[0032] FIGURE 11 shows a modification of the subunit, which is modified to handle three inputs, in accordance with some embodiments of the present disclosure. The method depicted by this figure is less convenient, in most aspects, and different from the multi-operand rectangular grid defined in section 2.

[0033] It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appre-

ciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes which may be substantially represented in computer readable medium and executed by a computer or processor, whether such computer or processor is explicitly shown.

DETAILED DESCRIPTION

[0034] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0035] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however, that it is not intended to limit the disclosure to the specific forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternatives falling within the scope of the disclosure.

[0036] The terms "comprises", "comprising", "includes", or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device, or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a system or apparatus preceded by "comprises... a" does not, without more constraints, preclude the existence of other elements or additional elements in the system or method.

[0037] In the following detailed description of the embodiments of the disclosure, reference is made to the accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure, and it is to be understood that other embodiments may be utilized and that changes may be made without departing from the scope of the present disclosure. The following description is, therefore, not to be taken in a limiting sense.

[0038] An overview of the proposed invention:

[0039] For a better understanding of the proposed invention, the following paragraphs provide an introduction to the simple mathematical background of the arithmetic logic and provide a general overview of the invention. In an embodiment, the numbers are written in binary form. However, instead of treating numbers as a sequence of binary symbols, they are treated as sets of natural numbers. For example, the integer seven, $7 = 111$ in binary form, would be represented as the set of natural numbers $\{0, 1, 2\}$. The number twelve, $12 = 1100$, is represented by the set $\{2, 3\}$. The number $21 = 10101$ is represented by the set $\{0, 2, 4\}$. Each natural number is a set of smaller natural numbers, and the elements of the set are the powers of 2 in binary representation.

[0040] Similarly, addition is also treated in terms of sets, and not sequences. For example, consider the sum $7+13 = (2^0+2^1+2^2)+(2^0+2^2+2^3)$, which is the sum of sets $\{0, 1, 2\} \oplus \{0, 2, 3\}$. Here, two new sets are formed - symmetric difference and intersection. That is, the powers that are not repeated $\{1, 3\}$, and the powers that repeat $\{0, 2\}$. To add a power of 2 with itself (i.e., numbers in the intersection), simply add "1" to that power, $2^n + 2^n = 2^{n+1}$. Therefore, the sum can be rewritten as $7 + 13 = (2^1 + 2^3) + (2^{0+1} + 2^{2+1})$. The first term, $2^1 + 2^3$, represents the symmetric difference $A \triangle B$, while the second term $2^{0+1} + 2^{2+1} = (2^0 + 2^2) + (2^0 + 2^2)$ represents the intersection. The sum has been reduced to $7 + 13 = (2^1 + 2^3) + (2^1 + 2^3)$. Iterating, there is no symmetric difference. And, adding "1" to the repeated powers gives $7 + 13 = 2^{1+1} + 2^{3+1} = 2^2 + 2^4 = 20$.

[0041] If A, B are two finite sets of natural numbers, they can be added using the same method. Form two new sets $A' = A \triangle B$ and $B = s(A \cap B)$, where s is the function that adds one unit, to the elements of $A \cap B$. Then $A + B = A' + B'$. It is guaranteed that, in a finite number of iterations, the intersection $A^{(k)} \cap B^{(k)} = \emptyset$ becomes the empty set. This yields the final answer $A^{(k+1)}$, because

$$\begin{aligned} A+B &= A^{(k+1)}+B^{(k+1)} \\ &= A^{(k+1)}+s(\emptyset) \\ &= A^{(k+1)} \end{aligned}$$

[0042] A second example is $15+23=38$ of FIG. 1. The operands in the initial state are $A=\{0,1,2,3\}$, and $B=\{0,1,2,4\}$ because $15=2^0+2^1+2^2+2^3$ and $23=2^0+2^1+2^2+2^4$. The second state is $A'=A\triangle B=\{3,4\}$, and $B'=s(A\cap B)=\{0+1,1+1,2+1\}=\{1,2,\}$. The next state is given by $A''=A'\triangle B'=\{1,2,4\}$ and $B''=s(A'\cap B')=\{3+1\}=\{4\}$. Iterating again gives $A'''=\{1,2\}$ and $B'''=\{4+1\}=\{5\}$. Iterating once more, a stable state is reached; $A^{(4)}=\{1,2,5\}$ and $B^{(4)}=0$.

[0043] The process described herein is a finite state machine. Each state is composed of two columns. Each column is a finite configuration of energy-levels representing one natural number, as is illustrated in FIG. 1. A particle in the basic level "0" is worth 1 unit, and a particle in level "1" is worth 2 units. A particle in level "2" is worth 4 units, and in general a particle in level "n" is worth 2^n units. A finite configuration of particles in a column represents a set number, so that each state is a pair of natural numbers. As shown in FIG. 1, the initial state $S(t_0)$ is given by the inputs A,B. The next state, $S(t_1)$ is given by two new columns. The configuration of the left column is given by the energy levels that were not repeated in state $S(t_0)$. The right column in $S(t_1)$ is given by the repeated objects, displaced one level up. The configuration of state $S(t_2)$ is defined similarly in terms of state $S(t_1)$. The left column of state $S(t_2)$ is given by the energy levels not repeated in state $S(t_1)$. The configuration in the right column of state $S(t_2)$ is given by the energy levels repeated in state $S(t_1)$ but displaced one level up. In general, the left column of state $S(t_{k+1})$ is given by the energy levels not repeated in state $S(t_k)$. The right column of state $S(t_{k+1})$ is given by a displacement, one level up, of the energy levels repeated in state $S(t_k)$. In a finite number of steps, a stable state is reached, where no particle occupies the right column. The result of the sum is given in the left column.

[0044] In an embodiment, the basic idea behind the circuit implementation of this addition algorithm is to receive two inputs A,B and output two new numbers $A'=(A\triangle B)$ and $B'=s(A\cap B)$. These two new numbers will satisfy $A'+B'=A+B$. Iterate the process using A',B' as new inputs, to obtain A'',B'' which satisfies $A''+B''=A+B$. In a finite number of iterations $B^{(k)}$ becomes zero. For a finite integer k, it is true that $B^{(k)}=0$ and the sum is $A^{(k)}=A+B$. This process will take, on average, $\log_2 n$ steps, where n is the number of bits. It takes at most n steps to terminate, and the probability for the process to end in $k\leq n$ steps is the probability of k successive heads in n coin tosses.

[0045] In an embodiment, to add two n bit numbers, four n bit registers, RA, RA' and RB, RB' are required. For example, RB' is the register of bits $RB'0, RB'1, \ldots, RB'(n-1)$. Registers will have "set" and "enable" connections for read and write functions, respectively. When registers RA and RB are on "set", registers RA' and RB' are on "enable". Similarly, when registers RA and RB are on "enable", registers RA' and RB' are on "set".

[0046] In an embodiment, the initial state $S(t_0)$ is saved in the RA and RB registers. These registers output their stored memory which will go through two different paths. One path will treat symmetric difference and the other will handle the intersection. The bit configuration saved in RA, RB is enabled to go through XOR gates, yielding symmetric differences. The definition of symmetric difference is equivalent to the truth table of the XOR gate. The output of each XOR gate will be saved in the same significant bit of the RA' register. On the second path, intersection is determined by AND gates. The output of each AND gate will be saved in the next significant bit of the RB' register. The intersection is displaced one level up, and this is reflected with the bit shift. At this point, state $S(t_1)$ is stored in registers RA', RB'. This represents the first iteration of our finite state machine. The bits stored in registers RA', RB' will be enabled to move through the XOR and AND gates. The result will be saved in the RA, RB registers, storing state $S(t_2)$ in registers RA, RB. Continue to move back and forth in this manner until the stopping condition is met. The stopping condition is that the output of RB/RB' (whichever is enabled) is equal to the zero vector.

[0047] The following components are needed. Four n bit registers, RA, RA', RB, RB'. A total of n XOR gates, and $4n$ AND gates with bit shift. The XOR determines symmetric difference and stores the results in the same significant bit. The AND gates provide the intersection, and the bit shift represents the rule $2^k + 2^k = 2^{k+1}$ applied to the objects of the intersection. Additionally, a Zero Flag "Z" checks for the stopping condition. Namely, that the right column, RB/RB' is off. The Zero Flag will take the value "Z=1" if any of the outputs from register RB/RB' are "1". It will take the value "Z=0" if and only if all of the outputs of register RB/RB' are "0". When the Zero Flag turns off, the Sum "S" is the set of signals S_0, S_1, S_2, S_3, which are output from register RA/RA'.

[0048] A bit shift requires three iterations to complete. Multiplication by 2 is the addition $s(A) = A \oplus A = 2 \odot A$. Find $A' = A \triangle A = 0$ and $B' = s(A \cap A) = s(A)$. The result is a displacement of A, one unit up, saved in register RB'. One more iteration gives $A'' = A' \triangle B' = 0 \triangle s(A) = s(A)$ and $B'' = s(A' \cap B') = s(0 \cap s(A)) = s(0) = 0$. In the third and final iteration, the stopping condition is met, because register RB outputs the zero vector. The sum is the output "S", of register RA.

[0049] Configuration and operation of the depackaging assembly:

[0050] In an embodiment, the functioning of each individual register is explained in detail in the following paragraphs. There is one data input "i" and one data output "o". Additionally, two more input signals are included. A set signal "s" to write, and an enable signal "e" to read. If "s" is a high signal "1", the data input "i" is stored in memory. If "e" is high, then the last input saved on memory is the data output "o" of the register. The external view of the data latch is shown in FIG. 2. The process described here will never have "e" and "s" on at the same time (nor will "e′" and "s′" be on at the same time). When one is on the other is off, so that the bits will never read and write simultaneously to avoid error. Only "s,e′" are on at the same time, as are "e,s′". This same function can be described using different read and write processes. The first model presented here, for illustrative purposes, is a level triggered version. A more efficient alternative is later described in this document using dual edge triggered flip flops which require a much more simple CU.

[0051] In an embodiment, implementation of the n-bit ALU requires four n bit registers, RA, RB, RA', RB'. This is shown in FIG. 3. Registers are arranged so that XOR and AND gates are placed in between the two columns of registers RA/RA' on the left and RB/RB' on the right. The output of the XOR gates is directed into registers RA/RA', while the output of the AND gates is directed into registers RB/RB' with a bit shift. Symmetric difference of the two columns will be saved in the left column RA/RA', and the intersection with a bit shift will be saved in the right column RB/RB'. For every bit of input, a subunit of two gates and four bits of memory is required.

[0052] The data inputs "$i = A_0, A_1, A_2, A_3$" and "$i = B_0, B_1, B_2, B_3$" are only activated at the beginning of the instruction set. At the same time, a high set signal "s" is activated. The result is that the initial state $S(t_0)$ is stored in registers RA, RB. The Zero Flag "Z", and Sum "S" are also shown in FIG. 3. The connections "Z" and "S" are outputs of the registers; inputs to the CU. The Zero Flag determines if the stopping condition is met, "Z=0". Namely, that the output from register RB/RB' is zero, 0000. The "S" connections coming from register RA/RA' will represent the resulting sum, when the stopping condition is met. A Carry Flag "CF" connection is included.

[0053] The Input/Output connections and logic gates are placed on the top layer, while "Z" and "S" are on a second layer, below the latter. This is shown in FIG. 4. In an embodiment, the set and enable connections of FIG. 4, "s,e,s′,e′" are each on their own layer so they do not intersect with each other, nor with the top two layers. The four layers of set and enable connections are represented by four thin lines that do not intersect. They function in the following manner. If "s′" (write RA', RB') is on, then "e" (read RA, RB) is on simultaneously. Similarly, if "e′" (read RA', RB') is on, then "s" (write RA, RB) is on.

[0054] A total of six layers of connections are needed. Four bottom layers for set and enable connections, and the two top layers for "i", "o" and "Z", "S". Three different line thicknesses are used in FIG. 4 to reflect this. Thin lines are used for the set "s" and enable "e" connections and they are placed at the bottom. Thick lines are placed on top of the four layers of thin lines and are used for "Z" and "S". Medium thickness lines are placed at the top layer and are used for input "i" and output "o".

[0055] The first step in the process is to write the data input signals $i = A_0$, B_0, A_1, B_1, A_2, B_2, A_3, B_3 in the registers $RA0, RB0, RA1, RB1, RA2, RB2, RA3, RB3$, respectively. The data connections appear at the bottom of the Control Unit in FIG. 4. This first step is achieved by activating the data input "i" signals, along with the set "s" signal. The input signals are on low "0" or high "1" according to the inputs A, B being represented. The input connections are activated only once at the beginning of the instruction set. Simultaneously, the set signal "s" is high "1". There is one exception. After the initial data input into $RB0$, the bit shift requires a "0" input into $RB'0$, then it will require low "0" to be input into $RB0$. This continues in an alternate manner until the stopping condition is met. This is specified in the instruction set. A "0" signal is sent to $RB'0$, the first time "s′" is activated, and every iteration after that "0" is sent to $RB0/RB'0$ in an alternate manner as explained.

[0056] In an embodiment, the second step is to output the data signal "o" of the RA, RB registers. This is achieved with a high enable "e" signal. The data outputs of RA, RB will go through the XOR and AND gates. At the same time "s′" is also on, so that RA', RB' registers write the output of the gates. The result is that the second state of the finite state machine is saved in the RA', RB' registers. The next iteration is to output the bits stored in RA', RB' and write the result on the RA, RB bits. This is achieved by turning on "e′" and "s" simultaneously. The third state of the system is stored in memory, in the RA, RB registers. Continuing in this manner for a finite number of iterations leads to a stable state; the output of register RB/RB' will be 0000 in a finite number of states. The result is the Sum "S" output of register RA'.

[0057] FIG. 5 shows a flow diagram for the instruction set, where the instruction set is constant and independent of the bit length of inputs, in accordance with some embodiments of the present disclosure. The instruction set for the flow diagram is given below:

1. Load data inputs $i = A_i, B_i$ to registers RAi, RBi, and activate Set "s=1".
2. Activate Enable "e=1" and Set "s′=1". Load data input "i=0" to $RB'0$ bit.
3. Read Zero Flag "Z"

   If "Z=0", Get "S";

   Else "Z=1", Activate Enable "e′=1" and Set "s=1". Load data input "i=0" to $RB0$ bit.
4. Read Zero Flag "Z"

   If "Z=0", Get "S";

   Else "Z=1", Go to II.

These instructions can be carried out largely by Hardware. This will be explained later in the document.

[0058] An example is illustrated in the following paragraphs. Let $A = 6 = 0110$ and $B = 3 = 0011$. The corresponding instructions are listed below:

I. First instruction will load inputs $A = 0110$ and $B = 0011$ to registers RA, RB. That is, $RA0 = 0, RA1 = 1, RA2 = 1, RA3 = 0$, and $RB0 = 1, RB1 = 1, RB2 = 0, RB3 = 0$.

II. Subsequent instruction will read the contents of registers RA, RB. These contents are directed to the XOR and AND gates. The output of these is written on the RA', RB' registers. In our example, the outputs of $RA0, RB0$ are "0,1", respectively. These outputs will then be directed to the XOR0 gate, and input "1" into $RA'0$. Simultaneously, the same "0,1" outputs, from $RA0, RB0$, will also be directed into the AND0 gate which will input "0" into $RB'1$. In an embodiment, the output of registers $RA1 = 1, RB1 = 1$ will input "0" into $RA'1$ and "1" into $RB'2$, after going through gates XOR1 and AND1, respectively. The outputs of $RA2 = 1, RB2 = 0$ will write "1" into $RA'2$ and "0" into $RB'3$ after passing through XOR2 and AND2. Also, $RA3 = RB3 = 0$ will write "0" into $RA'3$. The bit-shift requires the CU to input "0" into $RB'0$. While these outputs go through the gates and the results are written, the output of the RA, RB registers will be sent to the CU in the form of "$S_0 = 0, S_1 = 1, S_2 = 1, S_3 = 0$" and "Z=1", respectively.

III. The subsequent instruction will read "Z=1". The action path is to read RA', RB' and write the results on registers RA, RB. The results are $RA0 = 1, RB0 = 0, RA1 = 0, RB1 = 0, RA2 = 0, RB2 = 0, RA3 = 0, RB3 = 1$. Again, the bit-shift requires a "0" input into $RB0$. At the same time, the output of RA and RB has been sent to the CU in the form of "$S_0 = 1, S_1 = 0, S_2 = 1, S_3 = 0$" and "Z=1", respectively.

IV. The subsequent instruction will read "Z=1". Then, go to Instruction II.

II'. Outputs the memory of RA, RB into RA', RB'. The result will be $RA'0 = 1, RA'1 = 0, RA'2 = 0, RA'3 = 1$, and $RB'0 = RB'1 = RB'2 = RB'3 = 0$. At the same time, the output of RA and RB has been sent to the CU in the form of "$S_0 = 1, S_1 = 0, S_2 = 0, S_3 = 0$" and "Z=1", respectively.

III'. will read "Z=1". The action path is to read RA', RB' and write the results in RA, RB. At the same time, the output of RA', RB' is sent to the CU as "$S_0 = 1, S_1 = 0, S_2 = 0, S_3 = 1$" and "Z=0", respectively. This concludes the program, with "S" being the result of addition of the original inputs; $6 + 3 = 9$.

[0059] To represent a rational approximation of a non-negative real number, a fraction of the bits is used for the rational part and the remaining bits are used for the integer part. This gives us operation for fixed point rational numbers. The examples given are of fixed point nature. However, this ALU architecture is compatible with floating point representation and operations.

[0060] Negative energy levels are identified with negative powers of 2. Therefore, a set of negative integers will give a unique number in the unit interval $[0, 1]$. For example, the set $\{-1\}$ is the number $\frac{1}{2} = 2^{-1}$. The set representation of $\frac{3}{4} = 2^{-1} + 2^{-2}$ is the set $\{-1, -2\}$. Consider the finite state machine of FIG. 1. Notice that changing the labels on the energy levels gives a new expression. For example, making the bottom level equal to 3, instead of 0. This means 3 is added to every element of a set number. Instead of $15+23 = \{0, 1, 2, 3\} \oplus \{0, 1, 2, 4\}$, the new addition is $\{0 + 3, 1 + 3, 2 + 3, 3 + 3\} \oplus \{0 + 3, 1 + 3, 2 + 3, 4 + 3\} = \{3, 4, 5, 6\} \oplus \{3, 4, 5, 7\} = 120 + 184$. The new result is obtained by adding 3 to all the elements of the original result, $\{1{+}3, 2{+}3, 5{+}3\} = \{4, 5, 8\} = 304$.

[0061] In an embodiment, if the energy levels are displaced into negative integers, the results still hold a true expression. In FIG. 6, an example of this is provided. The addition of sets with negative integers in its elements is the same as before. The addition $\frac{1}{4} + \frac{1}{4} = \{-2\} \oplus \{-2\}$ is equal to $\frac{1}{2} = \{-1\}$. The set addition is $(\{-2\} \triangle \{-2\}) \oplus s(\{-2\} \cap \{-2\})$; first term is the empty set, and the second term is $s(\{-2\}) = \{-2{+}1\} = \{-1\}$.

[0062] The circuit for adding numbers whose elements include negative integers does not require any additional components. The circuit of FIG. 4 suffices. However, to divide numbers by 2, a second bit shift is needed. This is easily achieved by adding two enabling AND gates to each AND gate of the ALU. One gate will Enable Multiply "EM" and the other will Enable Divide "ED". Only one of these can be on at a time and must remain on during the entire time of the operation. Carry flags for multiplication "MCF" and division "DCF" are also included. This is illustrated in FIG. 7. A connection for input in $RB'3$ is also included for division, just as an input connection is included for $RB'0$.

[0063] One more component should be included in the description of the ALU. Once an addition is performed and new data inputs are to be loaded, the registers will be receiving signals from the CU and the XOR gates

because when "s" is on, so is "e′". To solve this, an AND gate is placed after each XOR gate. This enables the symmetric difference just as the "ED" and "EM" gates enable the intersection. These gates will be called "EXOR", and one of its inputs is connected to the output of its corresponding XOR gate and the other is a high signal whenever"EM" or "ED" are a high signal. This is a viable solution because it also gives a two gate depth to the XOR path, as in the case of the AND path. The XOR and AND paths have equal gate depth.

[0064] In an embodiment, the control logic is designed simple enough to show in a diagram. FIG. 8 is an internal view of the Control Unit. Step I requires set connection "s" to be on. This is achieved with a high signal in "e′/s". Simultaneously, the data inputs are also sent to the registers. Step II Requires for "s′/e=1" to be turned on. This will read registers RA and RB and write on registers RA' and RB'. The output of register RB will, at the same time, be directed to the Zero Flag "Z". A "Switch Unit" is included to perform the following function. The first time "Switch Unit" receives high input "Z=1", it will output a low signal to "s′/e". The next time the switch unit receives a high signal "Z=1", it will output a high signal. That high signal will go to gate "sw" so that now a low signal is directed to "e′/s". This continues in alternating manner so that the output of the switch unit moves between high and low, starting with a low signal. When the switch unit receives a low signal, there is no output because the stopping condition has been met.

[0065] In an embodiment, the control unit has an input "D/M". If addition is to be performed, "D/M=1" should be on. Bit shift equivalent to multiplication by 2 is performed if both inputs are equal. If the signal is low "D/M=0", then "ED=1" and the operation performed is division by 2 when both inputs are equal. The carry out connections "DCF" and "MCF" are shown again. The "EXOR" signal is given by gates "B1" and "B2"; it is on whenever "ED" or "EM" are on. A flag "F" is included for internal use of the CU. The flag is on when the stopping condition is met; the flag turns on when "Z" turns off. It can be used to save the Sum "S" in memory once the addition is completed. It is also used to indicate when new data inputs are loaded to the registers, and it shuts off "ED", "EM" and "EXOR". The flag is also used for outputting a zero value to $RB0/RB'0$. In the case of division, the bits $RB3/RB'3$ take their place.

[0066] In an embodiment, increasing the number of input bits increases the area linearly. This is a box-car architecture, where adding a bit to the operands requires to add a sub unit of four bits and five logic gates in linear manner (add a box car). A n-bit adder requires $4n$ many bits of registers, n many XOR gates, and $4n$ many AND gates. Compared to other fast architectures, this represents a significant reduction in material resources and area. The requirements are the same number of registers, and reduced gate count, area, and complexity. Furthermore, the instruction set remains the same, for any number of bits.

[0067] The finite state machine is not time constant. Calculation time is constant for equal inputs, but differs for different inputs. Let t, the time length for one iteration, then $t \cdot \log_2 n$ is the average time to complete an addition, and the longest time is $t \cdot n$. The circuit has a fixed gate depth of two logic gates, plus the lengthiest micro steps of reading and writing memory. This allows easy calculation of the time it takes to perform one iteration. It is equal to the sum of the two gate delays, plus the reading and writing process.

[0068] In an embodiment, the circuit is designed to have easy synchronization, independent of the choice of logic, clock speeds and register type. Particular solutions abound and are routine. The general principal of modeling the finite state machine through a logical circuit is being described. In terms of area, the CLA has area of order $O(n \log_2 n)$, while the area of the fast adder here proposed is of linear order $O(n)$. The ratio of these two orders is $O(\log_2 n)$. Approximately $\log_2 n$ many fast adders, of the type here proposed, may fit in the same space of one CLA of equal bits, as n gets bigger. Also, CLA performs in one clock cycle, while the proposed adder considers a positive number of iterations, on average $\log_2 n$ many iterations. It is concluded that performance is expected to be comparable in terms of area and speed, as the number of bits, n, and the number of operations performed, grow. This is true if the clock cycles of the compared adders are of equal time length. It must be considered that the clock cycles for the proposed adder will be shorter because the gate depth is a small constant and instructions are constant. This effect will potentially give better performance than other fast parallel adders. This design is likely to operate at higher than conventional clock speeds. If the memory writing process can be sped up, then the whole process will have a compounded reduction of time, and possibly outperform other fast adders, bit for bit.

[0069] Another advantage of the present invention would be power consumption, because the control logic and gate depths are a small constant number. The design can be adapted to specific applications such as general purpose, graphics, scientific, etc.

[0070] A second example is provided, illustrating the internal process of the adder and the control unit. This example will illustrate a bit shift to the right, which is equivalent to division by 2. In this case, let $A = B = 0111$. The result of the operation is $A + B = 0011$, and the carry flag "DCF" will be activated in the process. Load the data inputs; the set connection "s" is set to high. Simultaneously, the data inputs of A and B are activated. On every iteration that follows, the "D/M" input in the CU will be set to low so that the division gates "ED0-ED3" are enabled in the ALU. Thus, bits are carried to the right, not the left. Next, the "s′/e" input of the CU is turned on. This will enable reading of the RA, RB registers, and writing on RA', RB' registers. Specifically, registers $RA0 = 1, RB0 = 1$ will both input a high signal to gate "AND0", turning it on. Therefore, gate "ED0" will be turned on and it will be sending a signal to the division carry flag "DCF". This will simply indicate that a carry over to the right is taking place in the first bit. At the same time, the corresponding process takes place for the other bits. registers $RA1 = 1$ and $RB1 = 1$ turn gate "AND1" on. This turns gate "ED1" on, sending a high signal to register $RB'0$. A similar situation happens with the next bit, $RA2 = RB2 = 1$. These send a high signal to $RB'1$. The last bit, $RA3 = RB3 = 0$ will send a low signal to $RB'2$. Also, a low signal is sent to $RB'3$ as part of the instruction set. At the end of this process the configuration of the registers is $RA' = 0000$ and $RB' = 0011$. While this is taking place, the outputs of register RA are also sent to the Zero Flag. Since at least one of these bits is on, "Z" is on.

[0071] The high signal of "Z" will go into the switch unit which will output a low signal, initiating the second iteration. Registers $RA' = 0000$, and $RB' = 0011$ are read and then pushed through the XOR and AND gates. The output of the gates is written on registers $RA = 0011$, and $RB = 0000$, respectively. The output of the Zero Flag is "1", so a high signal goes into the switch unit which will output a high signal. This will read registers RA and RB. The output of register RB is the zero vector, so that the output of the zero flag is a low signal "Z=0". This signals RA to be saved in memory or operate where it is needed. At the same time, it will signal for the "ED", "EM", and "EXOR" to be off in the next clock cycle so that new data can be input to the registers without error. This example suggests that it could be convenient to have a Zero Flag for register RA, also. This last implementation would subtract one iteration from the bit shift.

[0072] Two sets of registers RA, RB, RA', RB' were used because of the racing problem. The outputs of the XOR and AND gates are looped back to the registers. That is why a rudimentary master-slave solution has been illustrated. However, that solution is not the most efficient. An alternative solution is presented. A variation of the proposed ALU can be implemented using edge triggered registers. If the registers are replaced for memory bits capable of handling inputs/outputs independently and without error of feedback, then the number of memory registers is reduced. A total of two n bit registers will suffice. Edge triggered registers offer a solution. A register with three connections is used: clock "CLK", input "D", and output "Q". Each register will have to read on the positive edge and write on the negative edge of the clock cycle. This is shown in FIG. 9. Enable divide, enable multiply, and enable EXOR gates are not shown.

[0073] A three operand version is possible and comparable to Carry Save Adder. The proposed unit should yield better performance when adding positive numbers. The expected sign difficulties of CSA are still present for signed operations of three operands. A comparison is given between the gate topology for a two operand unit in FIG. 10, and a three operand unit in FIG.11. This is not the same multi-operand principle of connecting parallel SLFAs in a rectangular grid defined in section 2. They are two distinct methods based on the same arithmetic principles .

[0074] In an embodiment, FIG. 10 shows the basic sub unit that allows for the iterative process on one-bit. It consists of a half adder where the output of AND is connected to the register $RB(i + 1)$ of the next bit, representing the new configuration in the right column of the finite state machine. The output of XOR is directed back to the RAi register of the same bit, representing the left column.

[0075] FIG. 11 is an adaptation of the adder, for three inputs. First, there are XOR and AND gates "2" and "3". When all three inputs A, B, C are a high signal, then gates "2" and "3" both output a high signal so that the output of gate "4" is a high signal. This amounts to a high signal being sent to the RB register of the next bit. That is, a carry over. Simultaneously, a high signal is sent back to the RA register of the same bit. A unit remains in the same bit. The case when all the inputs of a bit are "1" result in a carry over and unit in the same bit. This is the only case in which gate "4" is on, so no more attention is paid to it.

[0076] In an embodiment, if only two of the three inputs are a high signal, then gates "2" and "3" will both be off. Specifically, the fact that gate "3" is off, implies that gate "5" is on. Simultaneously, two of the three gates "6, 7, 8" are on. This means gate "9" is receiving two high signals, so that its output is a high signal "1". There is a carry over and no unit remains in that bit. The next case is when only one of three inputs is on. Gate "2" is off, and gate "3" is on; a unit remains in that bit, and no carry over is generated.

[0077] In an embodiment, if all inputs are off then gate "5" will be on, but gates "6, 7, 8" will be off so that gate "9" is also off. No carry overtakes place and there is no unit remaining, all cases have now been covered. The circuit whose elements are gates "2-9" is a one bit adder for three operands. Several bits are connected in series, to iterate until the system stabilizes. The control logic will remain the same.

[0078] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0079] The terms "including", "comprising", "having" and variations thereof mean "including but not limited to", unless expressly specified otherwise.

[0080] The enumerated listing of items does not imply that any or all the items are mutually exclusive, unless expressly specified otherwise. The terms "a", "an" and "the" mean "one or more", unless expressly specified otherwise.

[0081] While various aspects and embodiments have been disclosed herein, other aspects and embodiments will be apparent to those skilled in the art. The various aspects and embodiments disclosed herein are for purposes of illustration and are not intended to be limiting, with the true spirit being indicated by the following claims.

WHAT IS CLAIMED IS:

1. A linear fast adder for an Arithmetic Logic Unit (ALU), the adder comprising:

    a) a four-bit adder component comprising a plurality of logic gates comprising at least sixteen AND gates, four XOR gates; and

    b) a plurality of one-bit registers; wherein the four-bit adder is configured with a linear area, linear complexity and a logarithmic delay; and wherein the four-bit adder has a constant gate depth thereby resulting in constant power dissipation.

2. The linear fast adder of claim 1, wherein the four-bit adder component is configured to support a plurality of operands for integer type data and rational approximations to real number type data.

3. The linear fast adder of claim 1, wherein the four-bit adder component is configured to perform operations comprising at least one of left shift operation, right shift operation, addition, signed operations and one or more derived operations.

4. The linear fast adder of claim 3, wherein performing the operations comprises: representing the numbers in a binary form in corresponding set of natural numbers, such that each number is a set of smaller natural

numbers, wherein elements of the set of smaller numbers are denoted in powers of 2 in a binary representation.

5. The linear fast adder of claim 1 further comprises determining a symmetric difference corresponding to the operations performed at the four-bit adder component, the determining comprising:

   a) saving an initial state of the operations in at least one one-bit registers in the four- bit adder component;

   b) directing output of each of the one-bit registers in two disjoint paths; and

   c) computing the symmetric difference and intersection in the output of each of the one-bit registers.

6. The linear fast adder of claim 5, wherein the bit configurations saved in the one-bit registers is passed through at least one XOR gate in the four-bit adder component for yielding the symmetric difference.

7. The linear fast adder of claim 5, wherein the bit configurations saved in the one-bit registers is passed through at least one AND gate in the four-bit adder component for determining an intersection in the output.

8. The linear fast adder of claim 1, wherein to represent a rational approximation of non- negative real number, a fraction of the bits is used for the rational part and the remaining bits are used for the integer part.

9. The linear fast adder of claim 1, wherein adding a single bit to the operands requires adding of a sub-unit of four bits and five logic gates in a linear manner to the four-bit adder component.

10. The linear fast adder of claim 1, wherein the time taken by the linear fast adder is equal to sum of the two gate delays, and the reading and writing process.

11. The linear fast adder of claim 1, wherein clock cycles for the linear fast adder remains shorter depending on the gate depth and constant instructions, such that an increase in speed of memory writing process results in a compounded reduction of time.

12. The linear fast adder of claim 1, wherein an instruction set associated with the linear fast adder is constant and is independent of the number of bits of input provided to the linear fast adder.

13. The linear fast adder of claim 1, wherein the operation of the linear fast adder is controlled based on an arithmetic model that defines addition operations in terms of a finite state machine.

14. The linear fast adder of claim 13, wherein each state of the finite state machine comprises two columns and each column represents a finite configuration of energy levels representing one natural number.

15. The linear fast adder of claim 14, wherein in a subsequent set of the finite state machine, the finite configuration on a left column of the two columns represents the energy levels that are not repeated in the preceding state and the finite configuration on a right column of the two columns represents objects that are repeated from the preceding state.

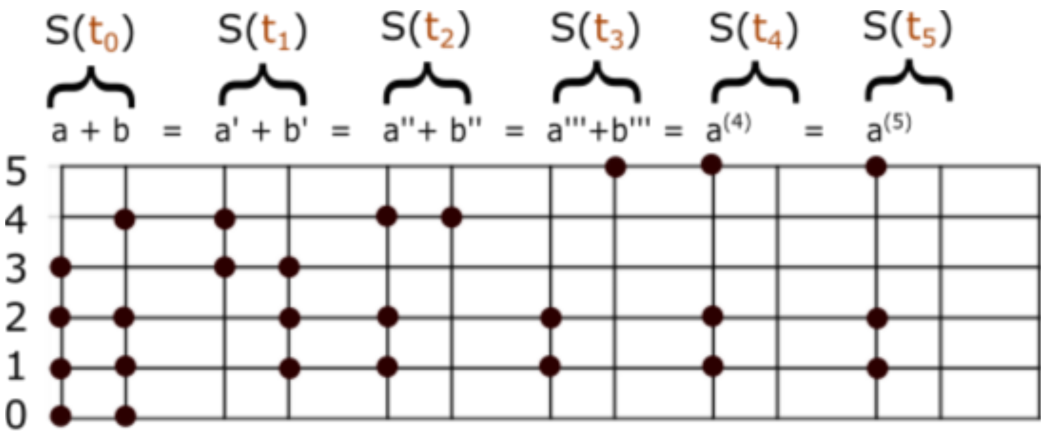

FIG. 1

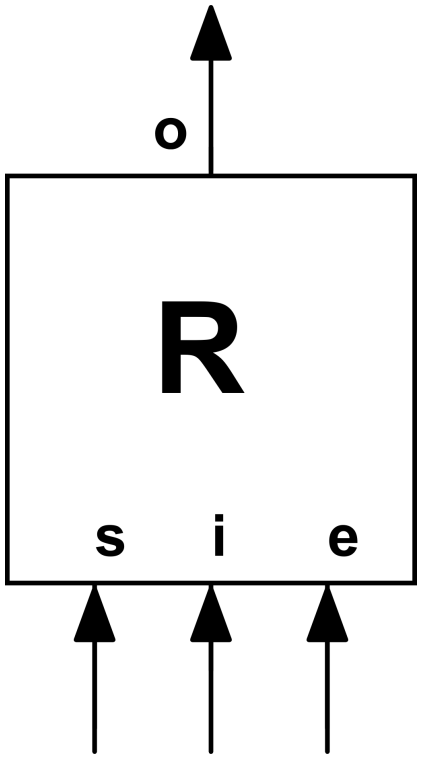

FIG. 2

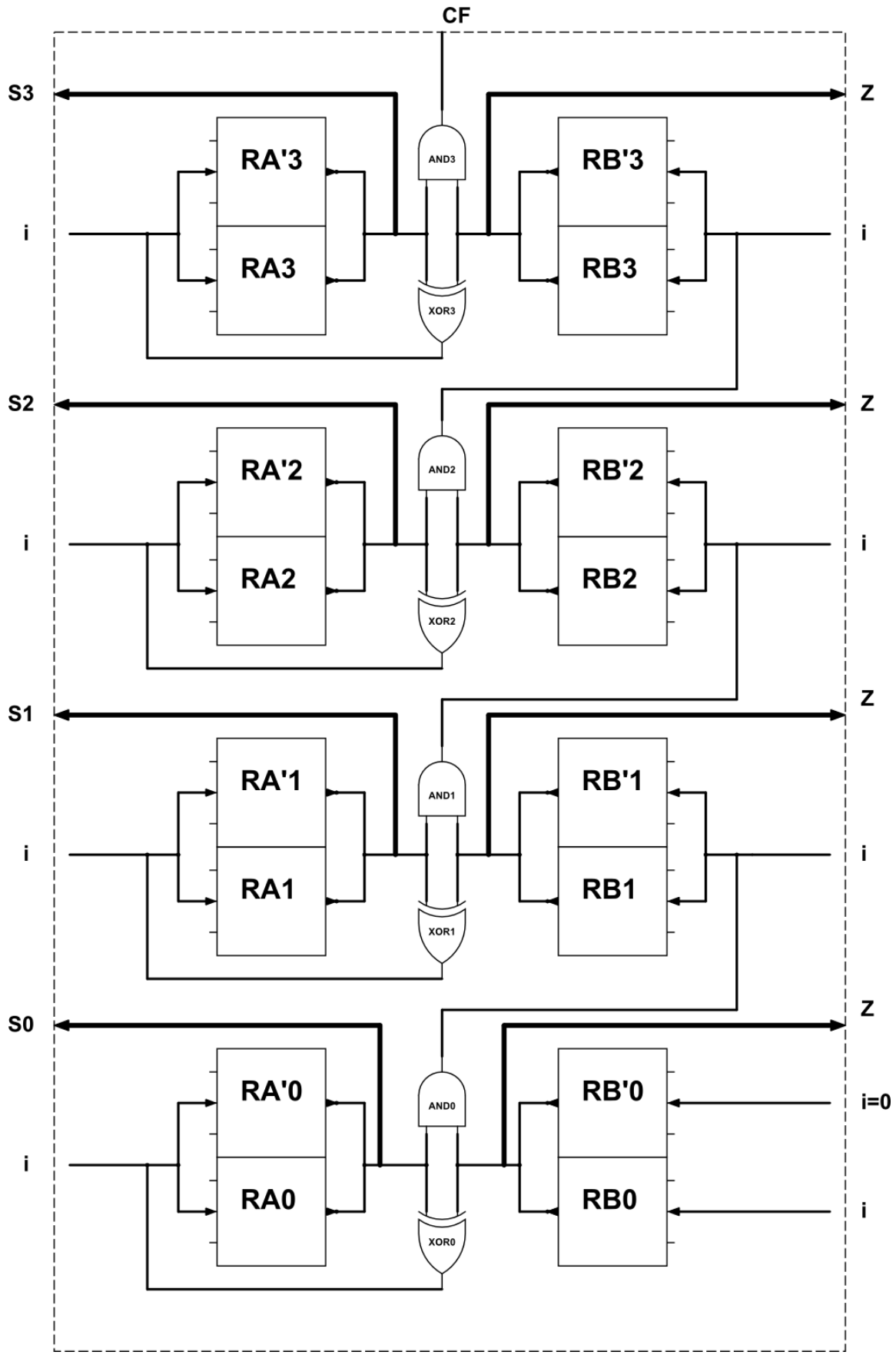
CF
S3
Z
RA'3
AND3
RB'3
i
RA3
XOR3
RB3
i
S2
Z
RA'2
AND2
RB'2
i
RA2
XOR2
RB2
i
S1
Z
RA'1
AND1
RB'1
i
RA1
XOR1
RB1
i
S0
Z
RA'0
AND0
RB'0
i=0
i
RA0
XOR0
RB0
i

FIG. 3

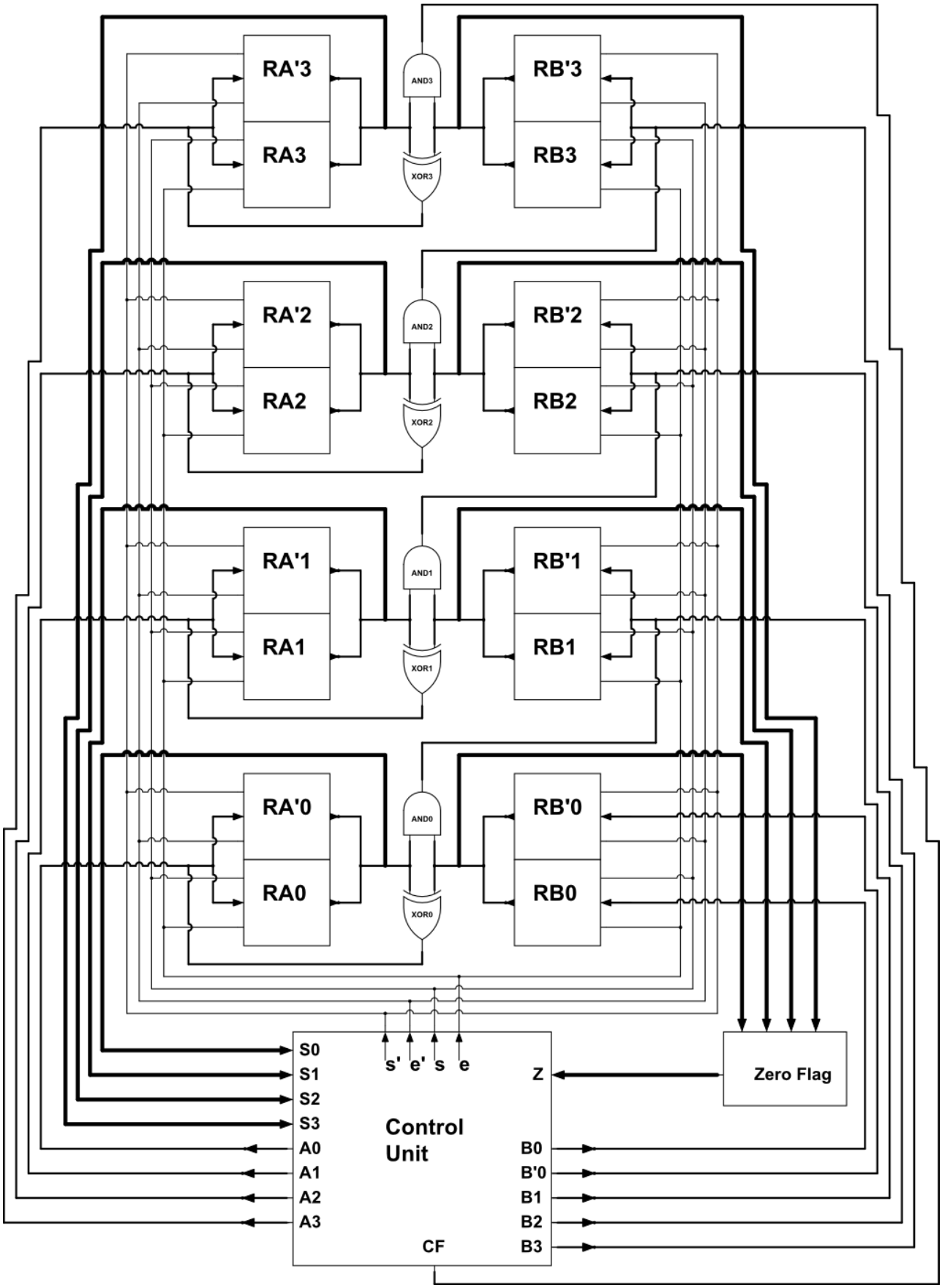
RA'3
RA3
AND3
XOR3
RB'3
RB3
RA'2
RA2
AND2
XOR2
RB'2
RB2
RA'1
RA1
AND1
XOR1
RB'1
RB1
RA'0
RA0
AND0
XOR0
RB'0
RB0
s'
e'
s
e
S0
S1
S2
S3
A0
A1
A2
A3
Control
Unit
CF
Z
Zero Flag
B0
B'0
B1
B2
B3

FIG. 4

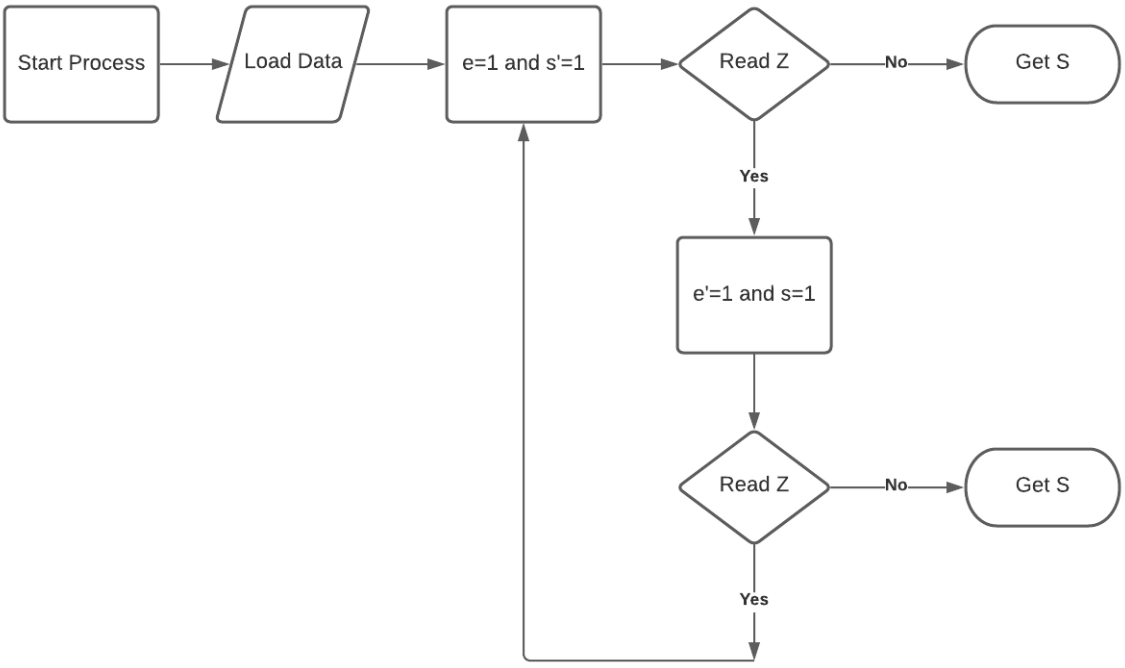
Start Process
Load Data
e=1 and s'=1
Read Z
No
Get S
Yes
e'=1 and s=1
Read Z
No
Get S
Yes

FIG. 5

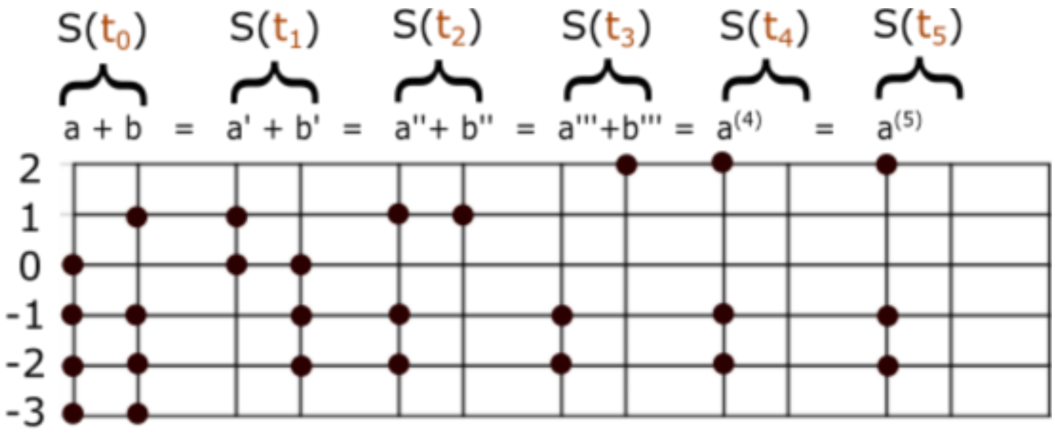
S(t0)
S(t1)
S(t2)
S(t3)
S(t4)
S(t5)
a + b = a' + b' = a''+ b'' = a'''+b''' = a(4) = a(5)
2
1
0
-1
-2
-3

FIG. 6

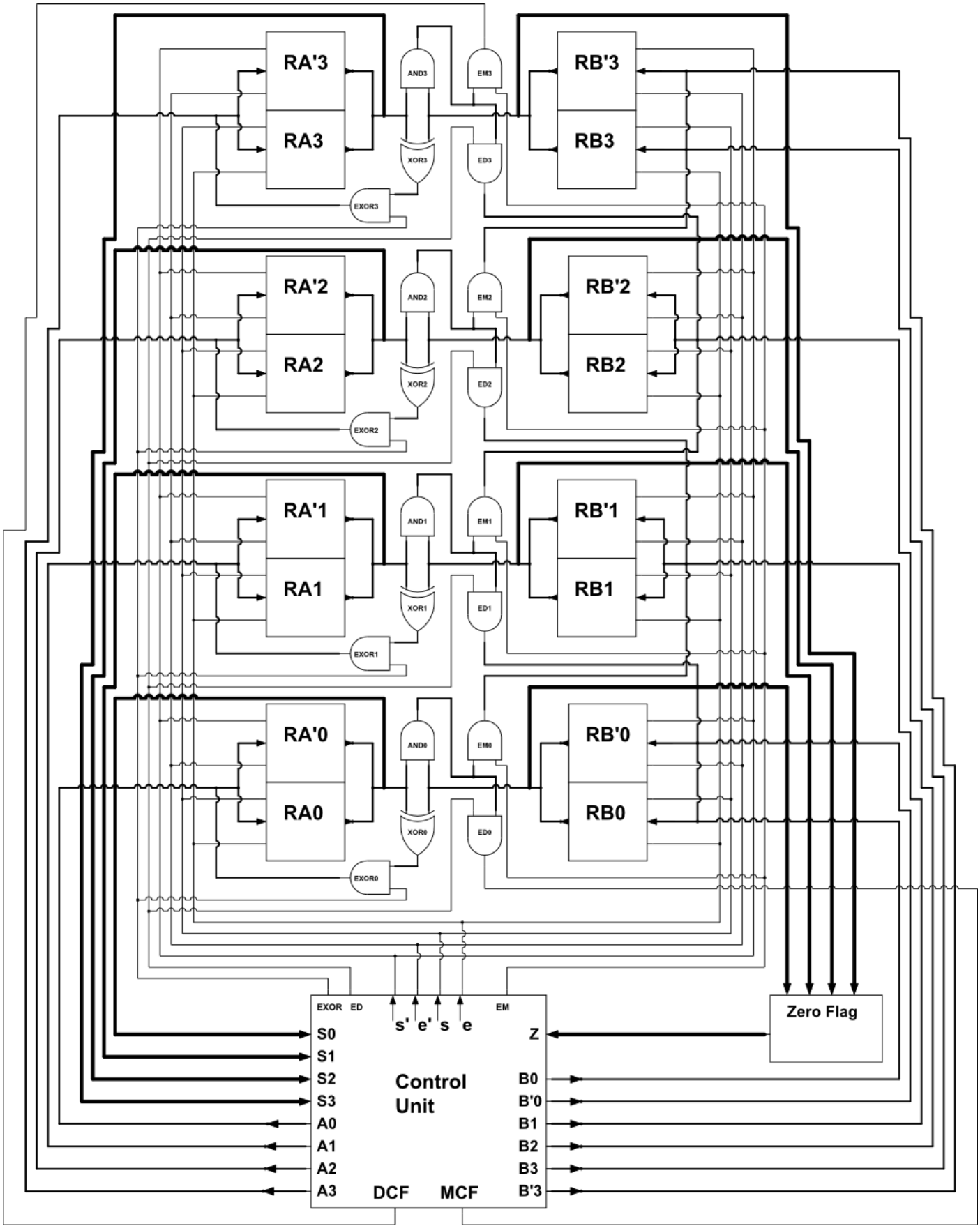
RA'3
RA3
AND3
EM3
RB'3
RB3
XOR3
ED3
EXOR3
RA'2
RA2
AND2
EM2
RB'2
RB2
XOR2
ED2
EXOR2
RA'1
RA1
AND1
EM1
RB'1
RB1
XOR1
ED1
EXOR1
RA'0
RA0
AND0
EM0
RB'0
RB0
XOR0
ED0
EXOR0
EXOR ED
EM
s' e' s e
S0
S1
S2
S3
A0
A1
A2
A3
Control Unit
DCF MCF
Z
B0
B'0
B1
B2
B3
B'3
Zero Flag

FIG. 7

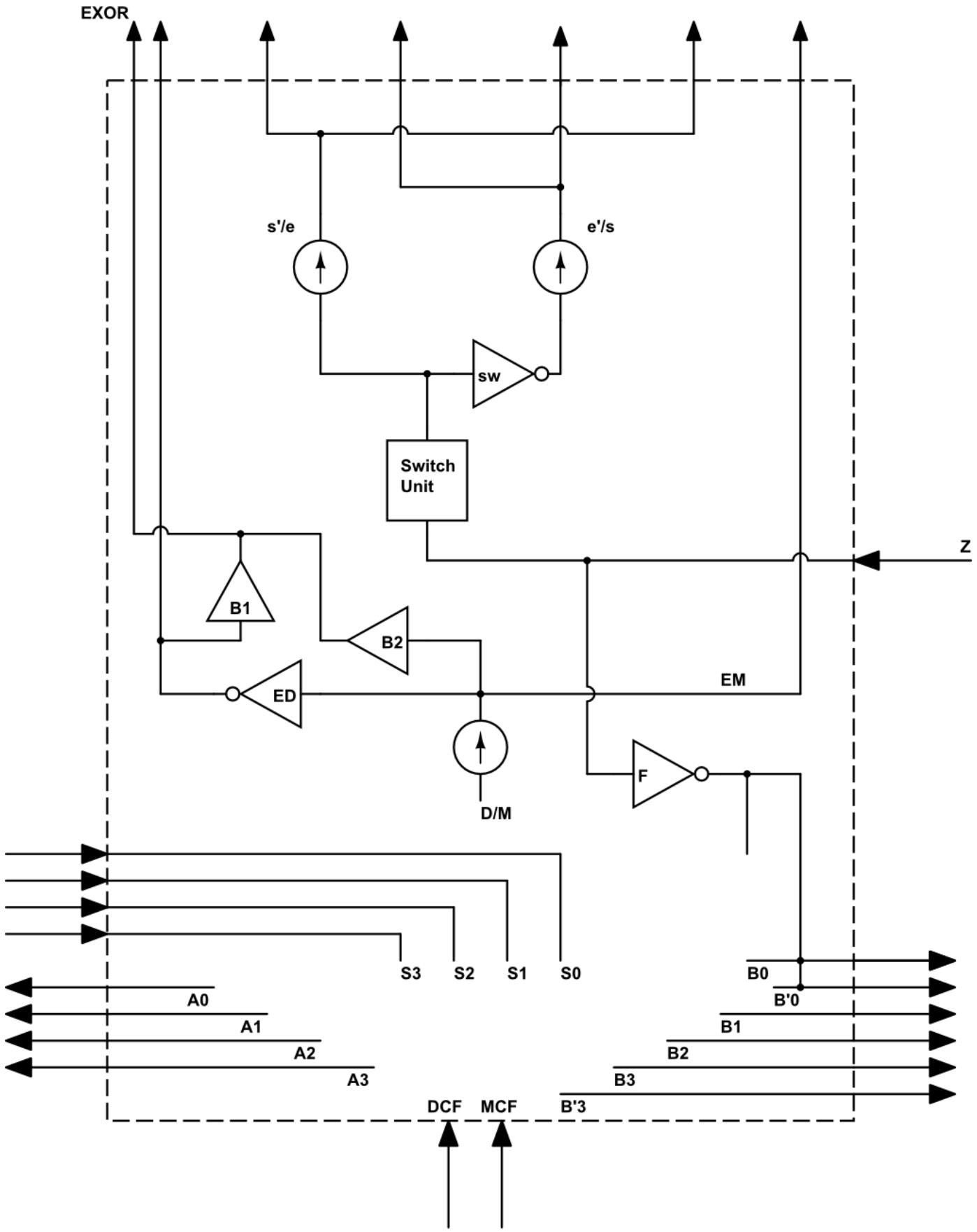
EXOR
s'/e
e'/s
sw
Switch Unit
Z
B1
B2
EM
ED
D/M
F
S3
S2
S1
S0
B0
B'0
A0
A1
A2
A3
B1
B2
B3
B'3
DCF
MCF

FIG. 8

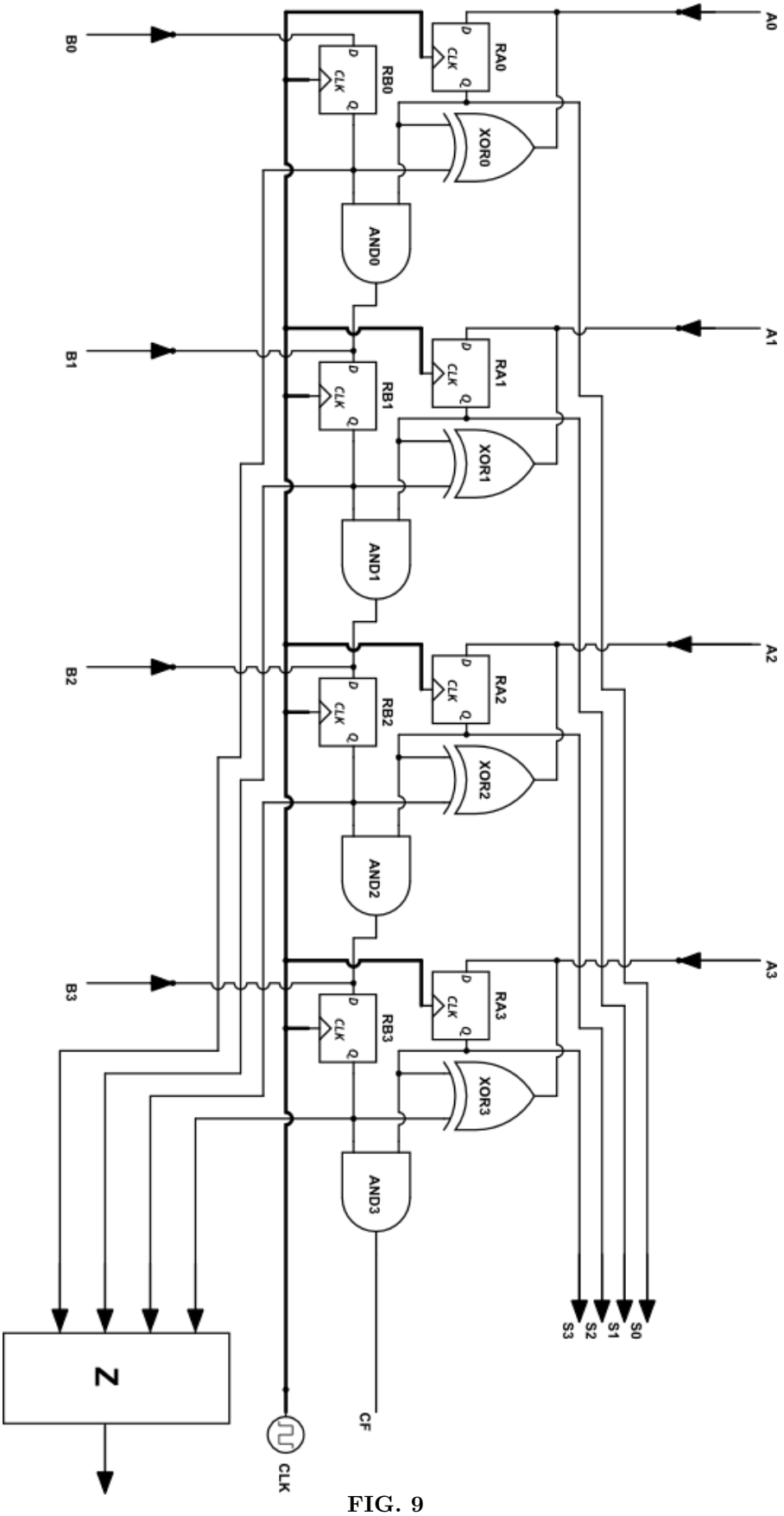
A0
B0
RA0
RB0
XOR0
AND0
A1
B1
RA1
RB1
XOR1
AND1
A2
B2
RA2
RB2
XOR2
AND2
A3
B3
RA3
RB3
XOR3
AND3
S3
S2
S1
S0
Z
CF
CLK
D
CLK
Q

FIG. 9

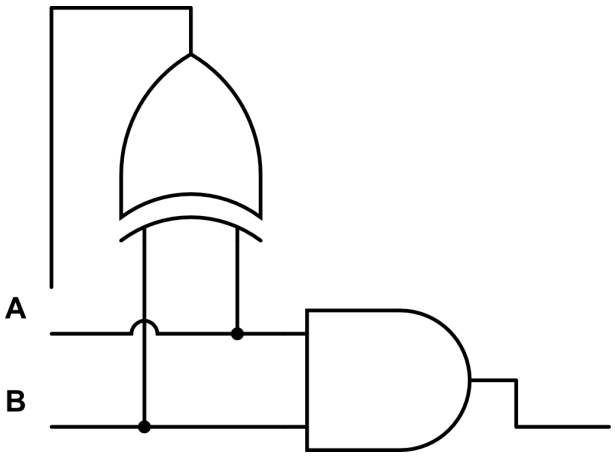

FIG. 10

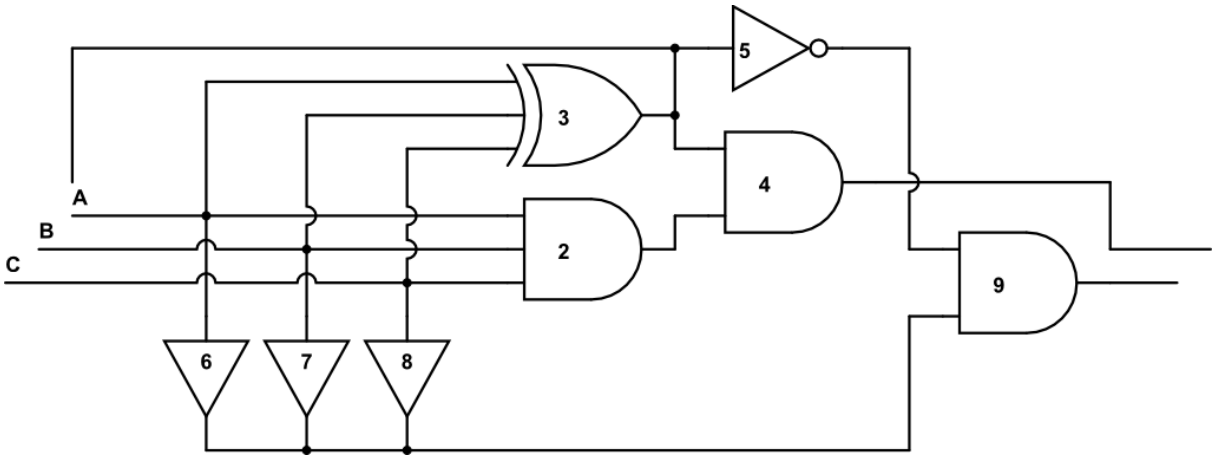

FIG. 11

B Canonical Block Forms

Examples are given, to illustrate the procedure for finding all groups of n elements along with their automorphisms. The canonical block form of the symmetry group Δ_4 is provided along with its automorphisms.

B.1 $|G| = 5$

If G is a group with five objects, then all non trivial objects satisfy $|g| \,|\, 5$. This implies $|g| = 5$, for all non trivial $g \in G$. Without loss of generality, choose any object g_1. Then, g_1^2 is a non trivial object, g_2. Also, $g_1 * g_2 = g_1^3$ is a new non trivial object, g_3, etc.

e	g_1	g_2	g_3	g_4
g_1	g_2			
g_2	g_3			
g_3	g_4			
g_4	e			

Now, use the associative property to find the operation function of g_2, and it will be placed in the second column. It is true $g_2 = g_1 * g_1$, so that it must also be true that $*g_2 = *g_1 \circ *g_1$. This means $*g_2(g_1)$ is found by $g_1 \to_{*g_1} g_2 \to_{*g_1} g_3$. Also, $*g_2(*g_2)$ because $g_2 \to_{*g_1} g_3 \to_{*g_1} g_4$, etc.

e	g_1	g_2	g_3	g_4
g_1	g_2	g_3		
g_2	g_3	g_4		
g_3	g_4	g_5		
g_4	e	g_1		

Do the same with the column of $g_3 = g_1 * g_2$ and $g_4 = g_1 * g_3$, so that $*g_3 = *g_1 \circ *g_2$ and $*g_4 = *g_1 \circ *g_3$. For example, $g_3 * g_1 = *g_3(g_1)$ is given by the arrows $g_1 \to_{*g_2} g_3 \to_{*g_1} g_4$, etc.

e	g_1	g_2	g_3	g_4
g_1	g_2	g_3	g_4	e
g_2	g_3	g_4	e	g_1
g_3	g_4	e	g_1	g_2
g_4	e	g_1	g_2	g_3

The group is defined by the number of objects, so that there exists only one group, $\mathbb{Z}_5$, of five objects. To find the canonical naming functions, make $e = 4$ and $a = 3$ for some object $a \in \mathbb{Z}_5$ such that $|a| = 5$. However, all non trivial objects have order 5, so that a can be any non trivial object. To maximize the representation, the object $b = a^2$ has to be assigned the numerical value 2.

4	3	2	1	0
3	2			
2				
1				
0				

The new object $c = a * b = a^3$ is assigned value 1, and $d = a^4$ is assigned value 0.

4	3	2	1	0
3	2			
2	1			
1	0			
0	4			

The rest of the table can be found, using the associative property.

4	3	2	1	0
3	2	1	0	4
2	1	0	4	3
1	0	4	3	2
0	4	3	2	1

This numerical table is given by four different naming functions. Consider the naming function that has $e = 4$ and $g_4 = 3$. Then $g_4^2 = g_3 = 2$, and $g_4^3 = g_2 = 1$, and $g_4^4 = g_1 = 0$. This naming functions is represented by the sequence (e, g_4, g_3, g_2, g_1). The four canonical naming functions are

$$\begin{array}{c}(e, g_1, g_2, g_3, g_4)\\(e, g_2, g_4, g_1, g_3)\\(e, g_3, g_1, g_4, g_2)\\(e, g_4, g_3, g_2, g_1).\end{array}$$

These four canonical naming functions are actually the automorphisms of $\mathbb{Z}_5$, in disguise. Fix any one of these naming functions, say $A = (e, g_3, g_1, g_4, g_2)$. Let B any other canonical naming function, say $B = (e, g_2, g_4, g_1, g_3)$. The bijective function defined below is an automorphism.

$$\begin{array}{c}e \mapsto e\\g_1 \mapsto g_4\\g_2 \mapsto g_3\\g_3 \mapsto g_2\\g_4 \mapsto g_1\end{array}$$

Let B any other canonical naming functions, say $B = (e, g_4, g_3, g_2, g_1)$. A second automorphism has been determined,

$$\begin{array}{c}e \mapsto e\\g_1 \mapsto g_3\\g_2 \mapsto g_1\\g_3 \mapsto g_4\\g_4 \mapsto g_2.\end{array}$$

Four automorphisms of $\mathbb{Z}_5$ are determined using the four canonical naming functions. The canonical representation is

$$\begin{aligned}N_{\mathbb{Z}_5} \;=\;& 2^{2^9+2^{2^{\left(2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} + 2^{2^7+2^{2^{\left(2^{(2^9+2^8)}+2^{(2^7+2^6)}+2^{(2^5+2^4)}+2^{(2^3+2^2)}+2^{(2^1+2^{10})}+1\right)}}}\\ &+2^{2^5+2^{2^{\left(2^{(2^9+2^6)}+2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^{10})}+2^{(2^1+2^8)}+1\right)}}} + 2^{2^3+2^{2^{\left(2^{(2^9+2^4)}+2^{(2^7+2^2)}+2^{(2^5+2^{10})}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}}}\\ &+2^{2^1+2^{2^{\left(2^{(2^9+2^2)}+2^{(2^7+2^{10})}+2^{(2^5+2^8)}+2^{(2^3+2^6)}+2^{(2^1+2^4)}+1\right)}}}.\end{aligned}$$

B.2 $|G| = 6$

Dihedral Group D_6. Begin as usual, with the list of objects.

e	g_1	g_2	g_3	g_4	g_5
g_1					
g_2					
g_3					
g_4					
g_5					

There exists at least one element of order equal to the smallest prime divisor of 6; there is at least one object of order 2. Since 3 is a prime divisor of 6, the group has at least one object of order 3, as well. In fact, there has to be a multiple of $\phi(3) = 2$ many objects of order 3. Therefore, any group of six objects will have either two, or four, objects of order 3. First, consider the case with two objects of order 3, and three objects of order

2. Suppose, without loss of generality, $g_1^2 = g_2$ and $g_1 * g_2 = e$.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & g_2 & e & & & \\ g_2 & e & g_1 & & & \\ g_3 & & & e & & \\ g_4 & & & & e & \\ g_5 & & & & & e \end{array}$$

The object $g_1 * g_3$ is a new object, g_4, and the column of g_1 is determined. Then, find the column of g_2 by means of the composition $*g_1 \circ *g_1$.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & g_2 & e & & & \\ g_2 & e & g_1 & & & \\ g_3 & g_4 & g_5 & e & & \\ g_4 & g_5 & g_3 & & e & \\ g_5 & g_3 & g_4 & & & e \end{array}$$

Then, use $|g_3| = 2$ to find

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & g_2 & e & & & \\ g_2 & e & g_1 & & & \\ g_3 & g_4 & g_5 & e & & \\ g_4 & g_5 & g_3 & g_2 & & \\ g_5 & g_3 & g_4 & g_1 & & \end{array} .$$

Use $|g_3| = 2$ again, now to find

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & g_2 & e & g_5 & & \\ g_2 & e & g_1 & g_4 & & \\ g_3 & g_4 & g_5 & e & & \\ g_4 & g_5 & g_3 & g_2 & & \\ g_5 & g_3 & g_4 & g_1 & & \end{array} .$$

It is trivial to find the columns of g_4, g_5 in terms of the rest of the columns, using associativity as usual.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & g_2 & e & g_5 & g_3 & g_4 \\ g_2 & e & g_1 & g_4 & g_5 & g_3 \\ g_3 & g_4 & g_5 & e & g_1 & g_2 \\ g_4 & g_5 & g_3 & g_2 & e & g_1 \\ g_5 & g_3 & g_4 & g_1 & g_2 & e \end{array} \tag{14}$$

This is the dihedral group D_6. It is determined by the equations

$$\begin{aligned} g_1^2 &= g_2 \\ g_1 * g_2 = g_3^2 = g_4^2 = g_5^2 &= e. \end{aligned}$$

Letters $a, b, c, \ldots$ and $x_1, x_2, x_3, \ldots$ are auxiliary variables in finding the canonical naming of groups. The first numerical value assigned is $e = 5$. Recall, the strategy is to assign the larger numbers first, by giving priority to the left-most columns. Within a column, priority is given to the objects of upper rows. Observe there are three objects of second order. One of these three objects, call it a, will be assigned the value 4. Then, whatever object may be chosen for b, there is a fourth object $a * b = x_1$. And, since $|a| = 2$ it is also true $a * x_1 = b$.

$$\begin{array}{cccc} e & a & b & x_1 \\ a & e & & \\ b & x_1 & & \\ x_1 & b & & \end{array}$$

To maximize the representation, name $b = 3$ and $a * b = x_1 = 2$. That yields the numeric table

$$\begin{array}{cccc} 5 & 4 & 3 & 2 \\ 4 & 5 & & \\ 3 & 2 & & \\ 2 & 3 & & \end{array}.$$

So far, the only thing known about the canonical naming functions, is that $a = 4$ is one of the second order objects. If there were an object that commutes with a, it would be used as the object b. But, from (14) it is clear there is no choice of b that commutes with a. That is to say, the second order objects of D_6 do not commute with any non trivial object. Therefore, new objects $c = b * a$ and $x_2 = a * c$, are added to the table as shown below

$$\begin{array}{cccccc} e & a & b & x_1 & c & x_2 \\ a & e & c & & & \\ b & x_1 & & & & \\ x_1 & b & & & & \\ c & x_2 & & & & \\ x_2 & c & & & & \end{array}.$$

Make $|b| = 2$, to maximize the representation. Now it is known a canonical naming function of this group must have $a = 4$ and $b = 3$ for two second order objects a, b. Also make $c = 1$ and $x_2 = 0$, to maximize the representation. Use $|b| = 2$ to find the rest of the column of b. The rest of the table is determined using associativity.

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 4 & 5 & 1 & 0 & 3 & 2 \\ 3 & 2 & 5 & 4 & 0 & 1 \\ 2 & 3 & 0 & 1 & 5 & 4 \\ 1 & 0 & 4 & 5 & 2 & 3 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

To obtain a canonical naming function make $a = 4$, $b = 3$ for two objects of order 2. Obviously, g_3, g_4, g_5 are equivalent objects; two of these have to be chosen to take the values of 4 and 3. This implies g_3, g_4, g_5 are equivalent. The object $a * b$ is assigned the value $x_1 = 2$. Then $b * a = c = 1$ and $a * c = x_2 = 0$. The objects g_2, g_3 are equivalent. There is a total of six possible canonical naming functions.

$$\begin{array}{lll} (e, g_3, g_4, g_2, g_1, g_5) & (e, g_4, g_3, g_1, g_2, g_5) & (e, g_5, g_3, g_2, g_1, g_4) \\ (e, g_3, g_5, g_1, g_2, g_4) & (e, g_4, g_5, g_2, g_1, g_3) & (e, g_5, g_4, g_1, g_2, g_3) \end{array}$$

The group has a total of six automorphisms, given by the six canonical naming functions. The group is shown in block form, but the blocks are not cosets of a normal subgroup (even though D_6 has a normal subgroup). The canonical block form is composed of four 3×3 blocks, and there are two types of blocks. The first type of block has objects in $A = \{1, 2, 3, 4, 5\}$ while the second type of block has objects in $B = \{0, 1, 2, 3, 4\}$. Blocks A_1 and A_2 are located in the upper left corner and lower right corner, respectively. Blocks B_1 and B_2 are in the upper right hand and lower left hand, respectively. This group has canonical representation

$$\begin{aligned} N_{D_6} \;=\; & 2^{2^{11}+2^{2^{\left(2^{(2^{11}+2^{12})}+2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} \\ & +2^{2^9+2^{2^{\left(2^{(2^{11}+2^{10})}+2^{(2^9+2^{12})}+2^{(2^7+2^6)}+2^{(2^5+2^8)}+2^{(2^3+2^2)}+2^{(2^1+2^4)}+1\right)}}} \\ & +2^{2^7+2^{2^{\left(2^{(2^{11}+2^8)}+2^{(2^9+2^4)}+2^{(2^7+2^{12})}+2^{(2^5+2^2)}+2^{(2^3+2^{10})}+2^{(2^1+2^6)}+1\right)}}} \\ & +2^{2^5+2^{2^{\left(2^{(2^{11}+2^6)}+2^{(2^9+2^2)}+2^{(2^7+2^{10})}+2^{(2^5+2^4)}+2^{(2^3+2^{12})}+2^{(2^1+2^8)}+1\right)}}} \\ & +2^{2^3+2^{2^{\left(2^{(2^{11}+2^4)}+2^{(2^9+2^8)}+2^{(2^7+2^2)}+2^{(2^5+2^{12})}+2^{(2^3+2^6)}+2^{(2^1+2^{10})}+1\right)}}} \\ & +2^{2^1+2^{2^{\left(2^{(2^{11}+2^2)}+2^{(2^9+2^6)}+2^{(2^7+2^4)}+2^{(2^5+2^{10})}+2^{(2^3+2^8)}+2^{(2^1+2^{12})}+1\right)}}}. \end{aligned}$$

Cyclic Group $\mathbb{Z}_6$. Now consider the case with four objects of order 4, and one object of order 2.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & e & & & & \\ g_2 & & g_3 & e & & \\ g_3 & & e & g_2 & & \\ g_4 & & & & & \\ g_5 & & & & & \end{array}$$

Without loss of generality, make $g_1 * g_2 = g_4$.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & e & & & & \\ g_2 & g_4 & g_3 & e & & \\ g_3 & g_5 & e & g_2 & & \\ g_4 & & & & & \\ g_5 & & & & & \end{array}$$

Using $|g_1| = 2$, one finds

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & e & & & & \\ g_2 & g_4 & g_3 & e & & \\ g_3 & g_5 & e & g_2 & & \\ g_4 & g_2 & & & & \\ g_5 & g_3 & & & & \end{array} .$$

Now use $|g_2| = |g_3| = 3$ to find

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & e & & & & \\ g_2 & g_4 & g_3 & e & g_5 & g_1 \\ g_3 & g_5 & e & g_2 & g_1 & g_4 \\ g_4 & g_2 & & & & \\ g_5 & g_3 & & & & \end{array} .$$

It is the case that $|g_4| = |g_5| \neq 2$, so the only option is $g_4^2 = g_3$ and $g_5^2 = g_2$.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & e & & & & \\ g_2 & g_4 & g_3 & e & g_5 & g_1 \\ g_3 & g_5 & e & g_2 & g_1 & g_4 \\ g_4 & g_2 & & & g_3 & \\ g_5 & g_3 & & & & g_2 \end{array}$$

It is easy to see that $|g_4| = |g_5| = 6$. It is concluded there is no group $|G| = 4$ with four objects of order 2. The table is determined and the cyclic group is obtained.

$$\begin{array}{cccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 \\ g_1 & e & g_4 & g_5 & g_2 & g_3 \\ g_2 & g_4 & g_3 & e & g_5 & g_1 \\ g_3 & g_5 & e & g_2 & g_1 & g_4 \\ g_4 & g_2 & g_5 & g_1 & g_3 & e \\ g_5 & g_3 & g_1 & g_4 & e & g_2 \end{array} \tag{15}$$

The cyclic group $\mathbb{Z}_6$ is determined by $|G| = 6$ and

$$\begin{aligned} g_1^2 &= e \\ g_2^2 = g_3^2 &= g_4 \\ g_1 * g_2 &= g_3 \\ g_4^2 &= g_2. \end{aligned}$$

To find the canonical naming functions, use $a, b, c, \ldots$ and $x_1, x_2, x_3, \ldots$ as auxiliary variables. Start naming $e = 5$. There is only one second order object, so $g_1 = a = 4$. Add an object $b \neq g_1$. Whatever object is chosen for b, there is another object $a * b = x_1$. The group is commutative, so $b * a = x_1$. Since $|a| = 2$, it can be verified that $a * x_1 = b$. Commutativity gives $x_1 * a = b$.

$$\begin{array}{cccc} e & a & b & x_1 \\ a & e & x_1 & b \\ b & x_1 & & \\ x_1 & b & & \end{array}$$

In order to maximize the representation, name $b = 3$ and $a * b = x_1 = 2$. But, it is still not known what object of the group will be assigned to $b = 3$. The possible naming functions are

$$\begin{array}{cccc} (e, g_1, g_2, g_4, g_3, g_5) & (e, g_1, g_3, g_5, g_2, g_4) & (e, g_1, g_4, g_2, g_3, g_5) & (e, g_1, g_5, g_3, g_2, g_4) \\ (e, g_1, g_2, g_4, g_5, g_3) & (e, g_1, g_3, g_5, g_4, g_2) & (e, g_1, g_4, g_2, g_5, g_3) & (e, g_1, g_5, g_3, g_4, g_2). \end{array}$$

Whatever object b may be, b^2 is a new object c. Then, the operation $a * c$ is a new object x_2. Consequently, $b * x_1 = x_1 * b = x_2$ and $x_1^2 = c$. To maximize the representation, make $c = 1$ and $x_2 = 0$.

$$\begin{array}{cccccc} e & a & b & x_1 & c & x_2 \\ a & e & x_1 & b & x_2 & c \\ b & x_1 & c & x_2 & & \\ x_1 & b & x_2 & c & & \\ c & x_2 & & & & \\ x_2 & c & & & & \end{array}$$

Some of the naming functions can be eliminated. Keep only those such that the square of the third component is equal to the fifth component ($b^2 = c$). Notice, only the objects g_2, g_3 are the square of some other object. The naming functions that satisfy these conditions are reduced to four.

$$\begin{array}{c} (e, g_1, g_2, g_4, g_3, g_5) \\ (e, g_1, g_3, g_5, g_2, g_4) \\ (e, g_1, g_4, g_2, g_3, g_5) \\ (e, g_1, g_5, g_3, g_2, g_4) \end{array}$$

Any of the naming functions above, gives the table below.

$$\begin{array}{cccccc} e & a & b & x_1 & c & x_2 \\ a & e & x_1 & b & x_2 & c \\ b & x_1 & c & x_2 & & \\ x_1 & b & x_2 & c & & \\ c & x_2 & & & & \\ x_2 & c & & & & \end{array}$$

No more operations can be determined with the information available. A choice must be made for b, c, so that $b * c = e$ or $b * c = a$. Choosing them so $b * c = e$ maximizes the representation. The four candidate naming functions above satisfy this condition, so the candidate naming functions have not been reduced by this. However, the table is now

$$\begin{array}{cccccc} e & a & b & x_1 & c & x_2 \\ a & e & x_1 & b & x_2 & c \\ b & x_1 & c & x_2 & e & a \\ x_1 & b & x_2 & c & a & e \\ c & x_2 & e & a & & \\ x_2 & c & a & e & & \end{array}.$$

Next, focus on c^2. Notice that two of the four naming functions satisfy $c^2 = x_1$. The other two naming functions satisfy $c^2 = b$. The latter two maximize the representation. The two canonical naming functions

are $(e, g_1, g_2, g_4, g_3, g_5)$ and $(e, g_1, g_3, g_5, g_2, g_4)$. The cyclic group $\mathbb{Z}_6$ has a total of two automorphisms. Take $A = (e, g_1, g_3, g_5, g_2, g_4)$. The non trivial automorphism is the function ϕ with components $e \mapsto e$, $g_1 \mapsto g_1$, $g_2 \mapsto g_3$, $g_3 \mapsto g_2$, $g_4 \mapsto g_5$, $g_5 \mapsto g_4$. Taking $A = (e, g_1, g_2, g_4, g_3, g_5)$ and $B = (e, g_1, g_3, g_5, g_2, g_4)$, gives the same non trivial automorphism ϕ. The numeric table givrn by these naming functions is

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 4 & 5 & 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 & 5 & 4 \\ 2 & 3 & 0 & 1 & 4 & 5 \\ 1 & 0 & 5 & 4 & 3 & 2 \\ 0 & 1 & 4 & 5 & 2 & 3 \end{array}.$$

The 2×2 block on the upper left hand corner is the normal subgroup $N = \mathbb{Z}_2$. The table is made up of nine 2×2 blocks that are the cosets N, bN and b^2N, for $b \in \{g_2, g_3\}$. These coset blocks form the group $\mathbb{Z}_3$. The canonical table above is written as

$$\begin{array}{ccc} N & bN & b^2N \\ bN & b^2N & N \\ b^2N & N & bN \end{array}.$$

The canonical naming table gives the additional information that $\mathbb{Z}_6/\mathbb{Z}_2 = \mathbb{Z}_3$. The canonical representation of the cyclic group is

$$\begin{aligned} N_{\mathbb{Z}_6} &= 2^{2^{11}+2^{2^{\left(2^{(2^{11}+2^{12})}+2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} \\ &\quad +2^{2^9+2^{2^{\left(2^{(2^{11}+2^{10})}+2^{(2^9+2^{12})}+2^{(2^7+2^6)}+2^{(2^5+2^8)}+2^{(2^3+2^2)}+2^{(2^1+2^4)}+1\right)}}} \\ &\quad +2^{2^7+2^{2^{\left(2^{(2^{11}+2^8)}+2^{(2^9+2^6)}+2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^{12})}+2^{(2^1+2^{10})}+1\right)}}} \\ &\quad +2^{2^5+2^{2^{\left(2^{(2^{11}+2^6)}+2^{(2^9+2^8)}+2^{(2^7+2^2)}+2^{(2^5+2^4)}+2^{(2^3+2^{10})}+2^{(2^1+2^{12})}+1\right)}}} \\ &\quad +2^{2^3+2^{2^{\left(2^{(2^{11}+2^4)}+2^{(2^9+2^2)}+2^{(2^7+2^{12})}+2^{(2^5+2^{10})}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}}} \\ &\quad +2^{2^1+2^{2^{\left(2^{(2^{11}+2^2)}+2^{(2^9+2^4)}+2^{(2^7+2^{10})}+2^{(2^5+2^{12})}+2^{(2^3+2^6)}+2^{(2^1+2^8)}+1\right)}}}. \end{aligned}$$

Up to this point, there has not been any difficulty in finding the canonical naming and representation of groups. The first groups are ordered

$$\begin{aligned} G_0 &= \mathbb{Z}_1 \\ G_1 &= \mathbb{Z}_2 \\ G_2 &= \mathbb{Z}_3 \\ G_3 &= \mathbb{Z}_4 \\ G_4 &= \mathbb{Z}_2 \oplus \mathbb{Z}_2 \\ G_5 &= \mathbb{Z}_5 \\ G_6 &= \mathbb{Z}_6 \\ G_7 &= D_6. \end{aligned}$$

The first step in finding the canonical naming function is to identify the objects of smallest order. By now it is easy to find the canonical table and representation of $\mathbb{Z}_7$ (see $\mathbb{Z}_5$). This is the next group in order, $G_8 = \mathbb{Z}_7$.

B.3 $|G| = 8$

To find groups of eight objects, observe the possible orders of the objects are the divisors of 8. Particularly, there exists at least one object of order 2. There can be a multiple of two, $2i$, objects of order 4, and a multiple of four,

$4j$, many objects of order 8. All four groups of eight objects will be found. Each group has a canonical naming function, obtained from the numeric table, a minimal independent set of equations that defines the group, and canonical representation. Then, the canonical representations of these groups are compared to find the order $G_9 < G_{10} < G_{11} < G_{12} < G_{13}$.

Direct Product $\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$. Take the simplest case first, and then it will be clear how things can be complicated little by little. The simplest case is to consider all objects of order 2 (make $i = j = 0$). Additionally, suppose the group is commutative. With maximization in mind, this gives the table

$$\begin{array}{cccccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 \\ g_1 & e & g_3 & g_2 & g_5 & g_4 & g_7 & g_6 \\ g_2 & g_3 & e & g_1 & & & & \\ g_3 & g_2 & g_1 & e & & & & \\ g_4 & g_5 & & & e & g_1 & & \\ g_5 & g_4 & & & g_1 & e & & \\ g_6 & g_7 & & & & & e & g_1 \\ g_7 & g_6 & & & & & g_1 & e \end{array} .$$

It is easy to see $g_2 * g_4 \notin \{e, g_1, g_2, g_3, g_4, g_5\}$. Suppose, without loss of generality, $g_2 * g_4 = g_6$. This determines the rest of the column of g_2. Then, use $g_2 * g_4 = g_4 * g_2$ and $g_2 * g_6 = g_6 * g_2$ to find the third row. Then it is possible to find the fourth column and fourth row, using the associative property. For example, use $g_6 * g_2 = g_4$ to find $g_4 * g_3 = g_6 * (g_2 * g_3) = g_6 * g_1 = g_7$. Finally, use $g_3 * g_5 = g_6$ to find $g_6 * g_4 = g_3 * (g_5 * g_4) = g_3 * g_1 = g_2$.

$$\begin{array}{cccccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 \\ g_1 & e & g_3 & g_2 & g_5 & g_4 & g_7 & g_6 \\ g_2 & g_3 & e & g_1 & g_6 & g_7 & g_4 & g_5 \\ g_3 & g_2 & g_1 & e & g_7 & g_6 & g_5 & g_4 \\ g_4 & g_5 & g_6 & g_7 & e & g_1 & g_2 & g_3 \\ g_5 & g_4 & g_7 & g_6 & g_1 & e & g_3 & g_2 \\ g_6 & g_7 & g_4 & g_5 & g_2 & g_3 & e & g_1 \\ g_7 & g_6 & g_5 & g_4 & g_3 & g_2 & g_1 & e \end{array} \tag{16}$$

This determines the group $\mathbb{Z}_2^3 = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$. This group table is already in canonical block form. Again, the special block form of cosets of $N_1 = \mathbb{Z}_2$ is seen. The expression $\mathbb{Z}_8/N = \mathbb{Z}_4$ is given in the table, because there are sixteen 2×2 blocks, $N_1, g_2N_1, g_4N_1, g_6N_1$, that form $\mathbb{Z}_4$.

$$\begin{array}{cccc} N_1 & g_2N_1 & g_4N_1 & g_6N_1 \\ g_2N_1 & N_1 & g_6N_1 & g_4N_1 \\ g_4N_1 & g_6N_1 & N_1 & g_2N_1 \\ g_6N_1 & g_4N_1 & g_2N_1 & N_1 \end{array}$$

The group $\mathbb{Z}_4$ has normal subgroup $N_2 = \mathbb{Z}_2$, and $\mathbb{Z}_4/\mathbb{Z}_2 = \mathbb{Z}_2$. A third expression of group quotients can be observed. Table (16) also gives the expression $\mathbb{Z}_8/\mathbb{Z}_4 = \mathbb{Z}_2$ because there are four 4×4 blocks forming the group $\mathbb{Z}_2$; these blocks are the cosets of $N_3 = \mathbb{Z}_4$,

$$\begin{array}{cc} N_3 & cN_3 \\ cN_3 & N_3 \end{array} .$$

To determine this group, seven objects of order 2 are required. Additionally, a commutative object, g_1, is needed. The following equations determine the group.

$$\begin{aligned} & e = g_1^2 = g_2^2 = g_3^2 = g_4^2 = g_5^2 = g_6^2 = g_7^2 \\ & g_1 * g = g * g_1, \qquad\qquad g \in \mathbb{Z}_2^3. \end{aligned}$$

There is a total of one hundred and sixty eight distinct automorphisms of $\mathbb{Z}_2^3$, given by the canonical naming functions. Assign $e = 7$ for the identity element, and $a = 6$ for any non trivial element of the group. Then, assign $b = 5$ to a second non trivial element. To maximize representation, assign $a * b = x_1 = 4$. Next, choose a third

object to assign to $c = 3$, and $a * c = x_2 = 2$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a				
x_1	b	a	e				
c	x_2			e	a		
x_2	c			a	e		
d	x_3					e	a
x_3	d					a	e

Finally, make $b * c = d = 1$ and $a * d = x_3 = 0$. This determines the rest of the table. The first object, a, is chosen from seven different possible choices. The object b is chosen from a total of six options. Finally, c is taken from a total of four different options. The numeric table, given by these canonical naming functions is

7	6	5	4	3	2	1	0
6	7	4	5	2	3	0	1
5	4	7	6	1	0	3	2
4	5	6	7	0	1	2	3
3	2	1	0	7	6	5	4
2	3	0	1	6	7	4	5
1	0	3	2	5	4	7	6
0	1	2	3	4	5	6	7

and the canonical representation is

$$
\begin{aligned}
N_{\mathbb{Z}_2^3} \;=\;& 2^{2^{15}+2^{2^{\left(2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} \\
&+2^{2^{13}+2^{2^{\left(2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^{10})}+2^{(2^9+2^{12})}+2^{(2^7+2^6)}+2^{(2^5+2^8)}+2^{(2^3+2^2)}+2^{(2^1+2^4)}+1\right)}}} \\
&+2^{2^{11}+2^{2^{\left(2^{(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^{16})}+2^{(2^9+2^{14})}+2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}}} \\
&+2^{2^9+2^{2^{\left(2^{(2^{15}+2^{10})}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^{14})}+2^{(2^9+2^{16})}+2^{(2^7+2^2)}+2^{(2^5+2^4)}+2^{(2^3+2^6)}+2^{(2^1+2^8)}+1\right)}}} \\
&+2^{2^7+2^{2^{\left(2^{(2^{15}+2^8)}+2^{(2^{13}+2^6)}+2^{(2^{11}+2^4)}+2^{(2^9+2^2)}+2^{(2^7+2^{16})}+2^{(2^5+2^{14})}+2^{(2^3+2^{12})}+2^{(2^1+2^{10})}+1\right)}}} \\
&+2^{2^5+2^{2^{\left(2^{(2^{15}+2^6)}+2^{(2^{13}+2^8)}+2^{(2^{11}+2^2)}+2^{(2^9+2^4)}+2^{(2^7+2^{14})}+2^{(2^5+2^{16})}+2^{(2^3+2^{10})}+2^{(2^1+2^{12})}+1\right)}}} \\
&+2^{2^3+2^{2^{\left(2^{(2^{15}+2^4)}+2^{(2^{13}+2^2)}+2^{(2^{11}+2^8)}+2^{(2^9+2^6)}+2^{(2^7+2^{12})}+2^{(2^5+2^{10})}+2^{(2^3+2^{16})}+2^{(2^1+2^{14})}+1\right)}}} \\
&+2^{2^1+2^{2^{\left(2^{(2^{15}+2^2)}+2^{(2^{13}+2^4)}+2^{(2^{11}+2^6)}+2^{(2^9+2^8)}+2^{(2^7+2^{10})}+2^{(2^5+2^{12})}+2^{(2^3+2^{14})}+2^{(2^1+2^{16})}+1\right)}}}.
\end{aligned}
$$

A non abelian group with all objects of order 2 does not exist. If $c * a = d$, then $d * a = c$ because $|a| = 2$. Since $|d| = 2$ it is also true $d * c = a$. On the other hand, $a * c = x_2$ and $|c| = 2$ imply $x_2 * c = a$. This is a contradiction with the definition of group.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	d	x_3	c	x_2
b	x_1	e	a				
x_1	b	a	e				
c	x_2			e	a	a	
x_2	c				e		
d	x_3					e	
x_3	d						e

The contradiction does not depend on the first four objects e, a, b, x_1. This means that any non abelian group of eight objects, must also have objects of order 4 or 8. In particular, the group $\mathbb{Z}_2^3$, above, is the only group of eight objects with all non trivial elements of order 2. The commutative condition determined the group. Commutativity of one non trivial element implies commutativity on all the elements of the group.

Dihedral Group D_8. Now consider groups with elements of orders 2 and 4. Only a multiple of $2 = \phi(4)$, many objects of order 4 are possible. First, consider the case with two objects of order 4, and five objects of second order. Let a, b, x_1, c, x_2 be the objects of order 2, and let d, x_3 the objects of order 4.

e	a	b	x_1	c	x_2	d	x_3
a	e						
b	x_1	e					
x_1	b		e				
c	x_2			e			
x_2	c				e		
d	x_3					a	e
x_3	d					e	a

Since $|b| = |x_1| = |c| = |x_2| = 2$, then $x_1 * b = b * x_1 = x_2 * c = c * x_2 = a$.

e	a	b	x_1	c	x_2	d	x_3
a	e						
b	x_1	e	a				
x_1	b	a	e				
c	x_2			e	a		
x_2	c			a	e		
d	x_3					a	e
x_3	d					e	a

Now, $|b| = |x_1| = |c| = |x_2| = 2$ implies $b * a = x_1$, $x_1 * a = b$, $c * a = x_2$, $x_2 * a = c$, respectively. Then, $d * a = x_3$ and $x_3 * a = d$. Notice a block form is starting to appear in the table for $K(4)$. There are 2×2 blocks forming the Klein group. Suppose, without loss of generality, $b * c = d$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a				
x_1	b	a	e				
c	x_2	d	x_3	e	a		
x_2	c	x_3	d	a	e		
d	x_3					a	e
x_3	d					e	a

The rest of the table is determined. Find $b * d = c$, and $d * c = b$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a				
x_1	b	a	e				
c	x_2	d	x_3	e	a	b	x_1
x_2	c	x_3	d	a	e	x_1	b
d	x_3	c	x_2			a	e
x_3	d	x_2	c			e	a

Next use $c = b * d$ to find $c * d = b * (d * d) = b * a = x_1$.

$$\begin{array}{cccccccc}
e & a & b & x_1 & c & x_2 & d & x_3 \\
a & e & x_1 & b & x_2 & c & x_3 & d \\
b & x_1 & e & a & & & & \\
x_1 & b & a & e & & & & \\
c & x_2 & d & x_3 & e & a & b & x_1 \\
x_2 & c & x_3 & d & a & e & x_1 & b \\
d & x_3 & c & x_2 & x_1 & b & a & e \\
x_3 & d & x_2 & c & b & x_1 & e & a
\end{array}$$

The rest of the table is determined similarly. For example, $|c| = 2$ implies $c * b = x_3$.

$$\begin{array}{cccccccc}
e & a & b & x_1 & c & x_2 & d & x_3 \\
a & e & x_1 & b & x_2 & c & x_3 & d \\
b & x_1 & e & a & x_3 & d & c & x_2 \\
x_1 & b & a & e & d & x_3 & x_2 & c \\
c & x_2 & d & x_3 & e & a & b & x_1 \\
x_2 & c & x_3 & d & a & e & x_1 & b \\
d & x_3 & c & x_2 & x_1 & b & a & e \\
x_3 & d & x_2 & c & b & x_1 & e & a
\end{array}$$

This is the Dihedral group, D_8, defined by $|G| = 8$ and the set of equations

$$\begin{aligned}
&e = a^2 = b^2 = x_1^2 = c^2 = x_2^2 \\
&d^2 = a.
\end{aligned}$$

It is the only group $|G| = 8$, with exactly two objects of order 4, and five objects of order 2. Notice that the set of equations only mentions seven different objects. The eighth object is $a * d$, and it is of order 4. To find the canonical naming function of this group, write the group with non generic symbols.

$$\begin{array}{cccccccc}
e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 \\
g_1 & e & g_3 & g_2 & g_5 & g_4 & g_7 & g_6 \\
g_2 & g_3 & g_1 & e & g_7 & g_6 & g_4 & g_5 \\
g_3 & g_2 & e & g_1 & g_6 & g_7 & g_5 & g_4 \\
g_4 & g_5 & g_6 & g_7 & e & g_1 & g_2 & g_3 \\
g_5 & g_4 & g_7 & g_6 & g_1 & e & g_3 & g_2 \\
g_6 & g_7 & g_5 & g_4 & g_3 & g_2 & e & g_1 \\
g_7 & g_6 & g_4 & g_5 & g_2 & g_3 & g_1 & e
\end{array} \tag{17}$$

Now, use the letters $a, b, \ldots, x_1, x_2, \ldots$ as auxiliary variables to find the canonical naming. Avoid confusion with the fact that the same symbols $a, b, \ldots, x_1, x_2, \ldots$ have just been used as auxiliary variables to find the group. Table (17) will be the reference for D_8. Start with $e = 7$, and an arbitrary object, $a = 6$, of order 2. Add an object $b = 5$. To maximize the representation, make $x_1 = a * b = b * a = 4$. Put simply, one must choose two objects, a, b, that satisfy $e = a^2$ and $a * b = b * a$. There are ten options of ordered pairs of D_8 that satisfy these relations. For example, g_7 is a second order object and it commutes with g_6. Also, g_1 is a second order object and it commutes with g_2. Furthermore, an element b can be found such that $|b| = 2$, maximizing the representation. There are six options to do this. The objects g_4, g_5 commute, as do g_6, g_7 and they are all second order objects. Also, notice g_1 commutes with each of them. Therefore, any of these objects can take the place of a, for now. Add another object to the table, say $c = 3$, and consequently $x_2 = a * c = 2$. Then, add another

object $d = 1$, for the product $b * c = d$, and $x_3 = a * d = 0$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b				
b	x_1	e	a				
x_1	b	a	e				
c	x_2	d	x_3				
x_2	c	x_3	d				
d	x_3	c	x_2				
x_3	d	x_2	c				

In order to maximize the representation, find c that commutes with a. This implies a also commutes with $x_2 = a * c$. The only object that commutes with at least four objects is g_1. Therefore, $a = g_1$. The possible canonical naming functions have been reduced to a total of sixteen possible naming functions. Choose b from four different objects, $\{g_4, g_5, g_6, g_7\}$, and then c can be chosen from four different objects.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a				
x_1	b	a	e				
c	x_2	d	x_3				
x_2	c	x_3	d				
d	x_3	c	x_2				
x_3	d	x_2	c				

None of the choices of naming functions will have $b * c = c * b$. This is because the four options for assigning b, which are g_4, g_5, g_6, g_7, only commute with g_1 and $a * b = x_1$. For example, g_4 only commutes with g_1 and $g_5 = g_1 * g_4$, etc. So far, the canonical block form is

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a	x_3	d		
x_1	b	a	e	d	x_3		
c	x_2	d	x_3				
x_2	c	x_3	d				
d	x_3	c	x_2				
x_3	d	x_2	c				

Notice that in eight of the sixteen possible naming functions, there are eight that satisfy $c^2 = e$. The other eight functions assign c to g_2 or g_3. To maximize the representation choose the first eight naming functions; assign c to a second order object.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a	x_3	d		
x_1	b	a	e	d	x_3		
c	x_2	d	x_3	e	a		
x_2	c	x_3	d	a	e		
d	x_3						
x_3	d						

The rest of the table is determined. There is a total of eight canonical naming functions. Choose b from the list $\{g_4, g_5, g_6, g_7\}$, and make $x_1 = g_1 * b$. Then choose c from the remaining two objects of that list. The objects g_4, g_5 are equivalent, and g_6, g_7 are equivalent. The order 4 objects g_2, g_3 are equivalent. The canonical naming functions are

$(e, g_1, g_4, g_5, g_6, g_7, g_3, g_2)$	$(e, g_1, g_5, g_4, g_6, g_7, g_2, g_3)$	$(e, g_1, g_6, g_7, g_4, g_5, g_2, g_3)$	$(e, g_1, g_7, g_6, g_4, g_5, g_3, g_2)$
$(e, g_1, g_4, g_5, g_7, g_6, g_2, g_3)$	$(e, g_1, g_5, g_4, g_7, g_6, g_3, g_2)$	$(e, g_1, g_6, g_7, g_5, g_4, g_3, g_2)$	$(e, g_1, g_7, g_6, g_5, g_4, g_2, g_3)$

The numeric table is

7	6	5	4	3	2	1	0
6	7	4	5	2	3	0	1
5	4	7	6	0	1	2	3
4	5	6	7	1	0	3	2
3	2	1	0	7	6	5	4
2	3	0	1	6	7	4	5
1	0	3	2	4	5	6	7
0	1	2	3	5	4	7	6

and the canonical representation is

$$
\begin{aligned}
N_{D_8} \;=\;& 2^{2^{15}+2^{2^{\left(2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^{9}+2^{10})}+2^{(2^{7}+2^{8})}+2^{(2^{5}+2^{6})}+2^{(2^{3}+2^{4})}+2^{(2^{1}+2^{2})}+1\right)}} \\
&+2^{2^{13}+2^{2^{\left(2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^{10})}+2^{(2^{9}+2^{12})}+2^{(2^{7}+2^{6})}+2^{(2^{5}+2^{8})}+2^{(2^{3}+2^{2})}+2^{(2^{1}+2^{4})}+1\right)}} \\
&+2^{2^{11}+2^{2^{\left(2^{(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^{16})}+2^{(2^{9}+2^{14})}+2^{(2^{7}+2^{4})}+2^{(2^{5}+2^{2})}+2^{(2^{3}+2^{8})}+2^{(2^{1}+2^{6})}+1\right)}} \\
&+2^{2^{9}+2^{2^{\left(2^{(2^{15}+2^{10})}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^{14})}+2^{(2^{9}+2^{16})}+2^{(2^{7}+2^{2})}+2^{(2^{5}+2^{4})}+2^{(2^{3}+2^{6})}+2^{(2^{1}+2^{8})}+1\right)}} \\
&+2^{2^{7}+2^{2^{\left(2^{(2^{15}+2^{8})}+2^{(2^{13}+2^{6})}+2^{(2^{11}+2^{2})}+2^{(2^{9}+2^{4})}+2^{(2^{7}+2^{16})}+2^{(2^{5}+2^{14})}+2^{(2^{3}+2^{10})}+2^{(2^{1}+2^{12})}+1\right)}} \\
&+2^{2^{5}+2^{2^{\left(2^{(2^{15}+2^{6})}+2^{(2^{13}+2^{8})}+2^{(2^{11}+2^{4})}+2^{(2^{9}+2^{2})}+2^{(2^{7}+2^{14})}+2^{(2^{5}+2^{16})}+2^{(2^{3}+2^{12})}+2^{(2^{1}+2^{10})}+1\right)}} \\
&+2^{2^{3}+2^{2^{\left(2^{(2^{15}+2^{4})}+2^{(2^{13}+2^{2})}+2^{(2^{11}+2^{6})}+2^{(2^{9}+2^{8})}+2^{(2^{7}+2^{12})}+2^{(2^{5}+2^{10})}+2^{(2^{3}+2^{14})}+2^{(2^{1}+2^{16})}+1\right)}} \\
&+2^{2^{1}+2^{2^{\left(2^{(2^{15}+2^{2})}+2^{(2^{13}+2^{4})}+2^{(2^{11}+2^{8})}+2^{(2^{9}+2^{6})}+2^{(2^{7}+2^{10})}+2^{(2^{5}+2^{12})}+2^{(2^{3}+2^{16})}+2^{(2^{1}+2^{14})}+1\right)}}.
\end{aligned}
$$

Direct Product $\mathbb{Z}_2 \oplus \mathbb{Z}_4$. Now consider groups with four objects of order 4, and three objects of order 2. Let a any object of order 4, so that $a^2 = b$ is a new object, as is $a^3 = c$.

e	a	x_1	x_2	b	x_3	x_4	x_5
a	x_1	x_2	e				
x_1	x_2	e	a				
x_2	e	a	x_1				
b							
x_3							
x_4							
x_5							

The column of a can be completed. Let b an object not in $\{e, a, x_1, x_2\}$. The expression $a * b = x_3$ holds for an element x_3 that is not in $\{e, a, x_1, x_2, b\}$. The columns of x_1 and x_2 can be completed.

e	a	x_1	x_2	b	x_3	x_4	x_5
a	x_1	x_2	e				
x_1	x_2	e	a				
x_2	e	a	x_1				
b	x_3	x_4	x_5				
x_3	x_4	x_5	b				
x_4	x_5	b	x_3				
x_5	b	x_3	x_4				

Choose b such that it has order $|b| = 2$. This gives the row of b.

$$\begin{array}{cccccccc} e & a & x_1 & x_2 & b & x_3 & x_4 & x_5 \\ a & x_1 & x_2 & e & & & & \\ x_1 & x_2 & e & a & & & & \\ x_2 & e & a & x_1 & & & & \\ b & x_3 & x_4 & x_5 & e & a & x_1 & x_2 \\ x_3 & x_4 & x_5 & b & & & & \\ x_4 & x_5 & b & x_3 & & & & \\ x_5 & b & x_3 & x_4 & & & & \end{array}$$

Consider two different cases; the cases where b, x_3 commute or not. Supposing they do not commute leads to contradiction. It is a good exercise to find the contradiction in the least number of steps. For the commutative case, the table is determined.

$$\begin{array}{cccccccc} e & a & x_1 & x_2 & b & x_3 & x_4 & x_5 \\ a & x_1 & x_2 & e & x_3 & x_4 & x_5 & b \\ x_1 & x_2 & e & a & x_4 & x_5 & b & x_3 \\ x_2 & e & a & x_1 & x_5 & b & x_3 & x_4 \\ b & x_3 & x_4 & x_5 & e & a & x_1 & x_2 \\ x_3 & x_4 & x_5 & b & a & x_1 & x_2 & e \\ x_4 & x_5 & b & x_3 & x_1 & x_2 & e & a \\ x_5 & b & x_3 & x_4 & x_2 & e & a & x_1 \end{array}$$

This is the direct product group $\mathbb{Z}_2 \oplus \mathbb{Z}_4$. To define this group, an object of order 4 is required, as given by the equations $a^2 = x_1$, $a * x_1 = x_2$, $a * x_2 = e$. Then, a second order object, b, that commutes with $x_3 = a * b$. These conditions form the system of equations

$$\begin{aligned} a^2 &= x_1 \\ a^3 &= x_2 \\ a * x_2 = b^2 &= e \\ a * b &= x_3 \\ b * x_3 &= x_3 * b \end{aligned}$$

To find the canonical naming functions, write the table in terms of g_i.

$$\begin{array}{cccccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 \\ g_1 & g_2 & g_3 & e & g_5 & g_6 & g_7 & g_4 \\ g_2 & g_3 & e & g_1 & g_6 & g_7 & g_4 & g_5 \\ g_3 & e & g_1 & g_2 & g_7 & g_4 & g_5 & g_6 \\ g_4 & g_5 & g_6 & g_7 & e & g_1 & g_2 & g_3 \\ g_5 & g_6 & g_7 & g_4 & g_1 & g_2 & g_3 & e \\ g_6 & g_7 & g_4 & g_5 & g_2 & g_3 & e & g_1 \\ g_7 & g_4 & g_5 & g_6 & g_3 & e & g_1 & g_2 \end{array} \tag{18}$$

Begin by assigning $e = 7$, and $a = 6$ for some second order object a. Choose an arbitrary object $b = 5$, and make $a * b = x_1 = 4$. The group is commutative. Choose b of second order. This gives the table of $K(4)$. Maximization of the representation is guaranteed thus far.

$$\begin{array}{cccc} e & a & b & x_1 \\ a & e & x_1 & b \\ b & x_1 & e & a \\ x_1 & b & a & e \end{array} .$$

Add another arbitrary object from the remaining elements; $c = 3$, and $a * c = x_2 = 2$. The objects that c can be chosen from are g_1, g_3, g_5, g_7. Then, the value of 1 is assigned to the element $b * c = c * b = d = 1$, and the

smallest value is assigned to $x_3 = a * d = 0$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	e	a	d	x_3	c	x_2
x_1	b	a	e	x_3	d	x_2	c
c	x_2	d	x_3				
x_2	c	x_3	d				
d	x_3	c	x_2				
x_3	d	x_2	c				

To continue maximizing the representation, choose a so that it is the square of some object, $c^2 = a$. Therefore, $g_2 = a = 7$ because g_2 is the only non trivial element that is square of another group element. This determines the table. To find a canonical naming function choose b from two possible choices, g_4, g_6. Then choose c from four possible choices, g_1, g_3, g_5, g_7. There is a total of eight canonical naming functions defining eight automorphisms of $\mathbb{Z}_2 \oplus \mathbb{Z}_4$.

$(e, g_2, g_4, g_6, g_1, g_3, g_5, g_7)$ $(e, g_2, g_4, g_6, g_3, g_1, g_7, g_5)$ $(e, g_2, g_4, g_6, g_5, g_7, g_1, g_3)$ $(e, g_2, g_4, g_6, g_7, g_5, g_3, g_1)$
$(e, g_2, g_6, g_4, g_1, g_3, g_7, g_5)$ $(e, g_2, g_6, g_4, g_3, g_1, g_5, g_7)$ $(e, g_2, g_6, g_4, g_5, g_7, g_3, g_1)$ $(e, g_2, g_6, g_4, g_7, g_5, g_1, g_3)$

These naming functions give the numeric table

7	6	5	4	3	2	1	0
6	7	4	5	2	3	0	1
5	4	7	6	1	0	3	2
4	5	6	7	0	1	2	3
3	2	1	0	6	7	4	5
2	3	0	1	7	6	5	4
1	0	3	2	4	5	6	7
0	1	2	3	5	4	7	6

with canonical representation -0cm

$$
\begin{aligned}
N_{\mathbb{Z}_2\oplus\mathbb{Z}_4} \;=\;& 2^{2^{15}+2^{2^{\left(2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^{9}+2^{10})}+2^{(2^{7}+2^{8})}+2^{(2^{5}+2^{6})}+2^{(2^{3}+2^{4})}+2^{(2^{1}+2^{2})}+1\right)}}} \\
&+2^{2^{13}+2^{2^{\left(2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^{10})}+2^{(2^{9}+2^{12})}+2^{(2^{7}+2^{6})}+2^{(2^{5}+2^{8})}+2^{(2^{3}+2^{2})}+2^{(2^{1}+2^{4})}+1\right)}}} \\
&+2^{2^{11}+2^{2^{\left(2^{(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^{16})}+2^{(2^{9}+2^{14})}+2^{(2^{7}+2^{4})}+2^{(2^{5}+2^{2})}+2^{(2^{3}+2^{8})}+2^{(2^{1}+2^{6})}+1\right)}}} \\
&+2^{2^{9}+2^{2^{\left(2^{(2^{15}+2^{10})}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^{14})}+2^{(2^{9}+2^{16})}+2^{(2^{7}+2^{2})}+2^{(2^{5}+2^{4})}+2^{(2^{3}+2^{6})}+2^{(2^{1}+2^{8})}+1\right)}}} \\
&+2^{2^{7}+2^{2^{\left(2^{(2^{15}+2^{8})}+2^{(2^{13}+2^{6})}+2^{(2^{11}+2^{4})}+2^{(2^{9}+2^{2})}+2^{(2^{7}+2^{14})}+2^{(2^{5}+2^{16})}+2^{(2^{3}+2^{10})}+2^{(2^{1}+2^{12})}+1\right)}}} \\
&+2^{2^{5}+2^{2^{\left(2^{(2^{15}+2^{6})}+2^{(2^{13}+2^{8})}+2^{(2^{11}+2^{2})}+2^{(2^{9}+2^{4})}+2^{(2^{7}+2^{16})}+2^{(2^{5}+2^{14})}+2^{(2^{3}+2^{12})}+2^{(2^{1}+2^{10})}+1\right)}}} \\
&+2^{2^{3}+2^{2^{\left(2^{(2^{15}+2^{4})}+2^{(2^{13}+2^{2})}+2^{(2^{11}+2^{8})}+2^{(2^{9}+2^{6})}+2^{(2^{7}+2^{10})}+2^{(2^{5}+2^{12})}+2^{(2^{3}+2^{14})}+2^{(2^{1}+2^{16})}+1\right)}}} \\
&+2^{2^{1}+2^{2^{\left(2^{(2^{15}+2^{2})}+2^{(2^{13}+2^{4})}+2^{(2^{11}+2^{6})}+2^{(2^{9}+2^{8})}+2^{(2^{7}+2^{12})}+2^{(2^{5}+2^{10})}+2^{(2^{3}+2^{16})}+2^{(2^{1}+2^{14})}+1\right)}}} .
\end{aligned}
$$

Quaternion Group Q_8. Consider G with six objects of order 4. Let a the only object of second order.

e	a	b	x_1	c	x_2	d	x_3
a	e						
b	x_1	a					
x_1	b		a				
c	x_2			a			
x_2	c				a		
d	x_3					a	
x_3	d						a

Find $x_1 * b = e$ and $b * x_1 = e$, using associativity. It can also be verified that $b * a = b^3 = a * b$. Using associativity it can be found $x_1 * a = b$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b				
b	x_1	a	e				
x_1	b	e	a				
c	x_2			a			
x_2	c				a		
d	x_3					a	
x_3	d						a

Suppose, without loss of generality, $b * c = d$.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b				
b	x_1	a	e				
x_1	b	e	a				
c	x_2	d	x_3	a			
x_2	c	x_3	d		a		
d	x_3	x_2	c			a	
x_3	d	c	x_2				a

Use associativity, as usual, to find

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b				
b	x_1	a	e				
x_1	b	e	a				
c	x_2	d	x_3	a	e		
x_2	c	x_3	d	e	a		
d	x_3	x_2	c			a	e
x_3	d	c	x_2			e	a

.

It is true $c * a = c^3 = a * c$; then, use associativity to find $x_2 * a$. In a similar fashion, $d * a = d^3 = a * d$ and $x_3 * a = d$ can be found.

e	a	b	x_1	c	x_2	d	x_3
a	e	x_1	b	x_2	c	x_3	d
b	x_1	a	e				
x_1	b	e	a				
c	x_2	d	x_3	a	e		
x_2	c	x_3	d	e	a		
d	x_3	x_2	c			a	e
x_3	d	c	x_2			e	a

Next use $c = x_1 * d$ to find $c * d = x_1 * (d * d) = x_1 * a = b$. The rest of the table is determined as usual. Written in generic variables $g_1, g_2, g_3, g_4, g_5, g_6, g_7$,

$$\begin{array}{cccccccc} e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 \\ g_1 & e & g_3 & g_2 & g_5 & g_4 & g_7 & g_6 \\ g_2 & g_3 & g_1 & e & g_7 & g_6 & g_4 & g_5 \\ g_3 & g_2 & e & g_1 & g_6 & g_7 & g_5 & g_4 \\ g_4 & g_5 & g_6 & g_7 & g_1 & e & g_3 & g_2 \\ g_5 & g_4 & g_7 & g_6 & e & g_1 & g_2 & g_3 \\ g_6 & g_7 & g_5 & g_4 & g_2 & g_3 & g_1 & e \\ g_7 & g_6 & g_4 & g_5 & g_3 & g_2 & e & g_1 \end{array}.$$

This group was determined by the conditions of having one second order object, g_1, and $g_1 = g_2^2 = g_3^2 = g_4^2 = g_5^2 = g_6^2$. Thus, the system of equations

$$\begin{aligned} & g_1^2 = e \\ & g_2^2 = g_4^2 = (g_1 * g_2)^2 = (g_1 * g_4)^2 = (g_2 * g_4)^2 = (g_4 * g_2)^2 = g_1. \end{aligned}$$

determines the quaternion group Q_8. To find the canonical naming functions, start with $e = 7$ and $g_1 = a = 6$ because g_1 is the only second order object. Then, choose a fourth order object to take the numerical value $b = 5$, and $x_1 = a * b = 4$. This object, b, is chosen from six possible objects of fourth order. Then, choose another object $c = 3$, and $a * c = x_2 = 2$, $b * c = d = 1$, $a * d = x_3 = 0$. The element c is chosen from four remaining group elements. All the fourth order objects are equivalent and there is a total of twenty four canonical naming functions and automorphisms. The numeric table given by the canonical naming functions is

$$\begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 6 & 7 & 4 & 5 & 2 & 3 & 0 & 1 \\ 5 & 4 & 6 & 7 & 0 & 1 & 3 & 2 \\ 4 & 5 & 7 & 6 & 1 & 0 & 2 & 3 \\ 3 & 2 & 1 & 0 & 6 & 7 & 4 & 5 \\ 2 & 3 & 0 & 1 & 7 & 6 & 5 & 4 \\ 1 & 0 & 2 & 3 & 5 & 4 & 6 & 7 \\ 0 & 1 & 3 & 2 & 4 & 5 & 7 & 6 \end{array}.$$

The canonical representation is

$$\begin{aligned} N_{Q_8} \quad = \quad & 2^{2^{15}+2^{2\left(2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}} \\ & +2^{2^{13}+2^{2\left(2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^{10})}+2^{(2^9+2^{12})}+2^{(2^7+2^6)}+2^{(2^5+2^8)}+2^{(2^3+2^2)}+2^{(2^1+2^4)}+1\right)}} \\ & +2^{2^{11}+2^{2\left(2^{(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^{14})}+2^{(2^9+2^{16})}+2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^6)}+2^{(2^1+2^8)}+1\right)}} \\ & +2^{2^9+2^{2\left(2^{(2^{15}+2^{10})}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^{16})}+2^{(2^9+2^{14})}+2^{(2^7+2^2)}+2^{(2^5+2^4)}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}} \\ & +2^{2^7+2^{2\left(2^{(2^{15}+2^8)}+2^{(2^{13}+2^6)}+2^{(2^{11}+2^2)}+2^{(2^9+2^4)}+2^{(2^7+2^{14})}+2^{(2^5+2^{16})}+2^{(2^3+2^{12})}+2^{(2^1+2^{10})}+1\right)}} \\ & +2^{2^5+2^{2\left(2^{(2^{15}+2^6)}+2^{(2^{13}+2^8)}+2^{(2^{11}+2^4)}+2^{(2^9+2^2)}+2^{(2^7+2^{16})}+2^{(2^5+2^{14})}+2^{(2^3+2^{10})}+2^{(2^1+2^{12})}+1\right)}} \\ & +2^{2^3+2^{2\left(2^{(2^{15}+2^4)}+2^{(2^{13}+2^2)}+2^{(2^{11}+2^8)}+2^{(2^9+2^6)}+2^{(2^7+2^{10})}+2^{(2^5+2^{12})}+2^{(2^3+2^{14})}+2^{(2^1+2^{16})}+1\right)}} \\ & +2^{2^1+2^{2\left(2^{(2^{15}+2^2)}+2^{(2^{13}+2^4)}+2^{(2^{11}+2^6)}+2^{(2^9+2^8)}+2^{(2^7+2^{12})}+2^{(2^5+2^{10})}+2^{(2^3+2^{16})}+2^{(2^1+2^{14})}+1\right)}}. \end{aligned}$$

Cyclic Group $\mathbb{Z}_8$. Finding the cyclic group is trivial, and it is defined by the equations $a^2 = b$, $b^2 = c$, $c^2 = e$. It has numeric table

7	6	5	4	3	2	1	0
6	5	4	3	2	1	0	7
5	4	3	2	1	0	7	6
4	3	2	1	0	7	6	5
3	2	1	0	7	6	5	4
2	1	0	7	6	5	4	3
1	0	7	6	5	4	3	2
0	7	6	5	4	3	2	1

The canonical representation of this group being

$$
\begin{aligned}
N_{\mathbb{Z}_8} \;=\;& 2^{2^{15}+2^{2^{\left(2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^{9}+2^{10})}+2^{(2^{7}+2^{8})}+2^{(2^{5}+2^{6})}+2^{(2^{3}+2^{4})}+2^{(2^{1}+2^{2})}+1\right)}}} \\
&+2^{2^{13}+2^{2^{\left(2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^{10})}+2^{(2^{9}+2^{8})}+2^{(2^{7}+2^{6})}+2^{(2^{5}+2^{4})}+2^{(2^{3}+2^{2})}+2^{(2^{1}+2^{16})}+1\right)}}} \\
&+2^{2^{11}+2^{2^{\left(2^{(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^{8})}+2^{(2^{9}+2^{6})}+2^{(2^{7}+2^{4})}+2^{(2^{5}+2^{2})}+2^{(2^{3}+2^{16})}+2^{(2^{1}+2^{14})}+1\right)}}} \\
&+2^{2^{9}+2^{2^{\left(2^{(2^{15}+2^{10})}+2^{(2^{13}+2^{8})}+2^{(2^{11}+2^{6})}+2^{(2^{9}+2^{4})}+2^{(2^{7}+2^{2})}+2^{(2^{5}+2^{16})}+2^{(2^{3}+2^{14})}+2^{(2^{1}+2^{12})}+1\right)}}} \\
&+2^{2^{7}+2^{2^{\left(2^{(2^{15}+2^{8})}+2^{(2^{13}+2^{6})}+2^{(2^{11}+2^{4})}+2^{(2^{9}+2^{2})}+2^{(2^{7}+2^{16})}+2^{(2^{5}+2^{14})}+2^{(2^{3}+2^{12})}+2^{(2^{1}+2^{10})}+1\right)}}} \\
&+2^{2^{5}+2^{2^{\left(2^{(2^{15}+2^{6})}+2^{(2^{13}+2^{4})}+2^{(2^{11}+2^{2})}+2^{(2^{9}+2^{16})}+2^{(2^{7}+2^{14})}+2^{(2^{5}+2^{12})}+2^{(2^{3}+2^{10})}+2^{(2^{1}+2^{8})}+1\right)}}} \\
&+2^{2^{3}+2^{2^{\left(2^{(2^{15}+2^{4})}+2^{(2^{13}+2^{2})}+2^{(2^{11}+2^{16})}+2^{(2^{9}+2^{14})}+2^{(2^{7}+2^{12})}+2^{(2^{5}+2^{10})}+2^{(2^{3}+2^{8})}+2^{(2^{1}+2^{6})}+1\right)}}} \\
&+2^{2^{1}+2^{2^{\left(2^{(2^{15}+2^{2})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^{14})}+2^{(2^{9}+2^{12})}+2^{(2^{7}+2^{10})}+2^{(2^{5}+2^{8})}+2^{(2^{3}+2^{6})}+2^{(2^{1}+2^{4})}+1\right)}}}.
\end{aligned}
$$

Groups of eight elements have been ordered; $\mathbb{Z}_8 < Q_8 < D_8 < \mathbb{Z}_2 \oplus \mathbb{Z}_4 < \mathbb{Z}_2^3$.

B.4 $|G| = 9$

Direct Product $\mathbb{Z}_3 \oplus \mathbb{Z}_3$. If $|G| = 9$ then $|g| = 3$ or $|g| = 9$ for any $g \in G$. Start by searching for groups with all objects of order 3. The function $*x_1$ is equal to the composition $*a \circ *a$. Since $|b| = 3$, it is true $b^2 \neq a$. If $b^2 = a$, then $b * a = e$ which is a contradiction. Suppose $b^2 = c$, without loss of generality.

e	a	x_1	b	x_2	x_3	c	x_4	x_5
a	x_1	e						
x_1	e	a						
b	x_2	x_3	c					
x_2	x_3	b						
x_3	b	x_2						
c	x_4	x_5	e					
x_4	x_5	c						
x_5	c	x_4						

The rest of the rows of b and c are found using associativity.

e	a	x_1	b	x_2	x_3	c	x_4	x_5
a	x_1	e						
x_1	e	a						
b	x_2	x_3	c	x_4	x_5	e	a	x_1
x_2	x_3	b						
x_3	b	x_2						
c	x_4	x_5	e	a	x_1	b	x_2	x_3
x_4	x_5	c						
x_5	c	x_4						

Check for a non abelian group. The first option for this is $b * a = x_3$, which implies $x_2 * a = b$. There is a contradiction because $|x_2| = 3$ implies $x_2 * x_4 = a$. There is also a contradiction if $b * a = x_4$. This relation implies $x_3 * a = c$. The last expression together with $|x_3| = 3$ implies $x_3 * x_1 = a$, which is a contradiction with $x_1^2 = a$. The supposition $b * a = x_5$ implies $x_2 * a = c$. This, together with $x_2 * c = a$ is a contradiction with the fact that $|x_2| = 3$. It can be concluded $b * a = x_2$. Then it can be found $x_2 * a = x_3$ and $x_3 * a = b$. Use the last expression to find $b * x_1 = x_3$.

e	a	x_1	b	x_2	x_3	c	x_4	x_5
a	x_1	e	x_2	x_3	b			
x_1	e	a	x_3	b	x_2			
b	x_2	x_3	c	x_4	x_5	e	a	x_1
x_2	x_3	b						
x_3	b	x_2						
c	x_4	x_5	e	a	x_1	b	x_2	x_3
x_4	x_5	c						
x_5	c	x_4						

Use the expression $b = x_2 * x_1$ to find $b * x_2 = x_4$. Then use $b = x_2 * x_1$ to find $b * x_4 = a$.

e	a	x_1	b	x_2	x_3	c	x_4	x_5
a	x_1	e	x_2	x_3	b			
x_1	e	a	x_3	b	x_2			
b	x_2	x_3	c	x_4	x_5	e	a	x_1
x_2	x_3	b	x_4	x_5	c			
x_3	b	x_2	x_5	c	x_4			
c	x_4	x_5	e	a	x_1	b	x_2	x_3
x_4	x_5	c	a	x_1	e			
x_5	c	x_4	x_1	e	a			

Finding the rest of the table is trivial. The result is the direct product group $\mathbb{Z}_3 \oplus \mathbb{Z}_3$. The system of equations that defines this group is given by the relations

$$\begin{aligned} a^2 &= x_1 \\ b^2 &= c \\ a^3 = b^3 &= e \\ a * b &= b * a. \end{aligned}$$

This group is determined by two commuting third order elements such that neither is the square of the other.

To find the canonical naming function of this group begin by expressing the table in generic variables g_i.

$$\begin{array}{ccccccccc}
e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 & g_8 \\
g_1 & g_2 & e & g_4 & g_5 & g_3 & g_7 & g_8 & g_6 \\
g_2 & e & g_1 & g_5 & g_3 & g_4 & g_8 & g_6 & g_7 \\
g_3 & g_4 & g_5 & g_6 & g_7 & g_8 & e & g_1 & g_2 \\
g_4 & g_5 & g_3 & g_7 & g_8 & g_6 & g_1 & g_2 & e \\
g_5 & g_3 & g_4 & g_8 & g_6 & g_7 & g_2 & e & g_1 \\
g_6 & g_7 & g_8 & e & g_1 & g_2 & g_3 & g_4 & g_5 \\
g_7 & g_8 & g_6 & g_1 & g_2 & e & g_4 & g_5 & g_3 \\
g_8 & g_6 & g_7 & g_2 & e & g_1 & g_5 & g_3 & g_4
\end{array}$$

Observe that the group is commutative. The group naming will be determined by choosing any two objects a, b that satisfy the conditions mentioned above. Let $e = 8$ and choose an arbitrary $a = 7$. Then $a^2 = 6$ maximizes the representation. The object a is chosen from eight possible choices. Next choose a second object $b = 5$, and assign the name $a * b = x_2 = 4$ and $a * x_2 = 3$. This object b can be chosen from six remaining objects. Finally, assign the values $b^2 = c = 2$, $a * c = x_4 = 1$, $a * x_4 = x_5 = 0$. This has determined a total of forty eight canonical naming functions and automorphisms of $\mathbb{Z}_3^2$. Choosing the two objects a, b determines the rest of the naming values. This gives the numeric table

$$\begin{array}{ccccccccc}
8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
7 & 6 & 8 & 4 & 3 & 5 & 1 & 0 & 2 \\
6 & 8 & 7 & 3 & 5 & 4 & 0 & 2 & 1 \\
5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 \\
4 & 3 & 5 & 1 & 0 & 2 & 7 & 6 & 8 \\
3 & 5 & 4 & 0 & 2 & 1 & 6 & 8 & 7 \\
2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 \\
1 & 0 & 2 & 7 & 6 & 8 & 4 & 3 & 5 \\
0 & 2 & 1 & 6 & 8 & 7 & 3 & 5 & 4
\end{array}.$$

The canonical representation of this group is the number

$$\begin{aligned}
N_{\mathbb{Z}_3^2} \;=\;& 2^{2^{17}+2^{2^{\left(2^{(2^{17}+2^{18})}+2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} \\
&+2^{2^{15}+2^{2^{\left(2^{(2^{17}+2^{16})}+2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{18})}+2^{(2^{11}+2^{10})}+2^{(2^9+2^8)}+2^{(2^7+2^{12})}+2^{(2^5+2^4)}+2^{(2^3+2^2)}+2^{(2^1+2^6)}+1\right)}}} \\
&+2^{2^{13}+2^{2^{\left(2^{(2^{17}+2^{14})}+2^{(2^{15}+2^{18})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^8)}+2^{(2^9+2^{12})}+2^{(2^7+2^{10})}+2^{(2^5+2^2)}+2^{(2^3+2^6)}+2^{(2^1+2^4)}+1\right)}}} \\
&+2^{2^{11}+2^{2^{\left(2^{(2^{17}+2^{12})}+2^{(2^{15}+2^{10})}+2^{(2^{13}+2^8)}+2^{(2^{11}+2^6)}+2^{(2^9+2^4)}+2^{(2^7+2^2)}+2^{(2^5+2^{18})}+2^{(2^3+2^{16})}+2^{(2^1+2^{14})}+1\right)}}} \\
&+2^{2^9+2^{2^{\left(2^{(2^{17}+2^{10})}+2^{(2^{15}+2^8)}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^4)}+2^{(2^9+2^2)}+2^{(2^7+2^6)}+2^{(2^5+2^{16})}+2^{(2^3+2^{14})}+2^{(2^1+2^{18})}+1\right)}}} \\
&+2^{2^7+2^{2^{\left(2^{2^{(2^{17}+2^8)}+(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^2)}+2^{(2^9+2^6)}+2^{(2^7+2^4)}+2^{(2^5+2^{14})}+2^{(2^3+2^{18})}+2^{(2^1+2^{16})}+1\right)}}} \\
&+2^{2^5+2^{2^{\left(2^{(2^{17}+2^6)}+2^{(2^{15}+2^4)}+2^{(2^{13}+2^2)}+2^{(2^{11}+2^{18})}+2^{(2^9+2^{16})}+2^{(2^7+2^{14})}+2^{(2^5+2^{12})}+2^{(2^3+2^{10})}+2^{(2^1+2^8)}+1\right)}}} \\
&+2^{2^3+2^{2^{\left(2^{(2^{17}+2^4)}+2^{(2^{15}+2^2)}+2^{(2^{13}+2^6)}+2^{(2^{11}+2^{16})}+2^{(2^9+2^{14})}+2^{(2^7+2^{18})}+2^{(2^5+2^{10})}+2^{(2^3+2^8)}+2^{(2^1+2^{12})}+1\right)}}} \\
&+2^{2^1+2^{2^{\left(2^{(2^{17}+2^2)}+2^{(2^{15}+2^6)}+2^{(2^{13}+2^4)}+2^{(2^{11}+2^{14})}+2^{(2^9+2^{18})}+2^{(2^7+2^{16})}+2^{(2^5+2^8)}+2^{(2^3+2^{12})}+2^{(2^1+2^{10})}+1\right)}}}.
\end{aligned}$$

Cyclic Group $\mathbb{Z}_9$. If $|G| = 9$, the objects of G have order 3 or 9. The only group with all objects of order three has been found. Now consider the case where there exists at least one object of order 9. But, since the group has nine objects, this must be the cyclic group, $\mathbb{Z}_9$. The cyclic group of nine objects is trivially given by

the numeric table

$$\begin{array}{ccccccccc} 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 \\ 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 \\ 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 \\ 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 \\ 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 \\ 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{array} .$$

The canonical representation is

$$\begin{aligned} N_{\mathbb{Z}_9} \;=\; & 2^{2^{17}+2^{2^{\left(2^{(2^{17}+2^{18})}+2^{(2^{15}+2^{16})}+2^{(2^{13}+2^{14})}+2^{(2^{11}+2^{12})}+2^{(2^9+2^{10})}+2^{(2^7+2^8)}+2^{(2^5+2^6)}+2^{(2^3+2^4)}+2^{(2^1+2^2)}+1\right)}}} \\ & +2^{2^{15}+2^{2^{\left(2^{(2^{17}+2^{16})}+2^{(2^{15}+2^{14})}+2^{(2^{13}+2^{12})}+2^{(2^{11}+2^{10})}+2^{(2^9+2^8)}+2^{(2^7+2^6)}+2^{(2^5+2^4)}+2^{(2^3+2^2)}+2^{(2^1+2^{18})}+1\right)}}} \\ & +2^{2^{13}+2^{2^{\left(2^{(2^{17}+2^{14})}+2^{(2^{15}+2^{12})}+2^{(2^{13}+2^{10})}+2^{(2^{11}+2^8)}+2^{(2^9+2^6)}+2^{(2^7+2^4)}+2^{(2^5+2^2)}+2^{(2^3+2^{18})}+2^{(2^1+2^{16})}+1\right)}}} \\ & +2^{2^{11}+2^{2^{\left(2^{(2^{17}+2^{12})}+2^{(2^{15}+2^{10})}+2^{(2^{13}+2^8)}+2^{(2^{11}+2^6)}+2^{(2^9+2^4)}+2^{(2^7+2^2)}+2^{(2^5+2^{18})}+2^{(2^3+2^{16})}+2^{(2^1+2^{14})}+1\right)}}} \\ & +2^{2^9+2^{2^{\left(2^{(2^{17}+2^{10})}+2^{(2^{15}+2^8)}+2^{(2^{13}+2^6)}+2^{(2^{11}+2^4)}+2^{(2^9+2^2)}+2^{(2^7+2^{18})}+2^{(2^5+2^{16})}+2^{(2^3+2^{14})}+2^{(2^1+2^{12})}+1\right)}}} \\ & +2^{2^7+2^{2^{\left(2^{2^{(2^{17}+2^8)}+(2^{15}+2^6)}+2^{(2^{13}+2^4)}+2^{(2^{11}+2^2)}+2^{(2^9+2^{18})}+2^{(2^7+2^{16})}+2^{(2^5+2^{14})}+2^{(2^3+2^{12})}+2^{(2^1+2^{10})}+1\right)}}} \\ & +2^{2^5+2^{2^{\left(2^{(2^{17}+2^6)}+2^{(2^{15}+2^4)}+2^{(2^{13}+2^2)}+2^{(2^{11}+2^{18})}+2^{(2^9+2^{16})}+2^{(2^7+2^{14})}+2^{(2^5+2^{12})}+2^{(2^3+2^{10})}+2^{(2^1+2^8)}+1\right)}}} \\ & +2^{2^3+2^{2^{\left(2^{(2^{17}+2^4)}+2^{(2^{15}+2^2)}+2^{(2^{13}+2^{18})}+2^{(2^{11}+2^{16})}+2^{(2^9+2^{14})}+2^{(2^7+2^{12})}+2^{(2^5+2^{10})}+2^{(2^3+2^8)}+2^{(2^1+2^6)}+1\right)}}} \\ & +2^{2^1+2^{2^{\left(2^{(2^{17}+2^2)}+2^{(2^{15}+2^{18})}+2^{(2^{13}+2^{16})}+2^{(2^{11}+2^{14})}+2^{(2^9+2^{12})}+2^{(2^7+2^{10})}+2^{(2^5+2^8)}+2^{(2^3+2^6)}+2^{(2^1+2^4)}+1\right)}}} . \end{aligned}$$

Comparing the two numbers, it is verified $\mathbb{Z}_9 < \mathbb{Z}_3^2$. It is becoming more clear how to find the canonical representation, without having to calculate all the representations. But there are still several considerations before attacking the general case.

The canonical block form of the symmetric group Δ_4 is calculated, as another example.

B.5 Δ_4

The multiplication table of Δ_4 is exhibited for reference. The symbols g_i are used for the elements of order 2, and h_i for the rest of the objects.

$$
\begin{array}{cccccccccccccccccccccccc}
e & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 & g_8 & g_9 & h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 & h_{10} & h_{11} & h_{12} & h_{13} & h_{14} \\
g_1 & e & g_4 & g_5 & g_2 & g_3 & h_2 & h_1 & h_4 & h_3 & g_7 & g_6 & g_9 & g_8 & h_6 & h_5 & h_{12} & h_{11} & h_{14} & h_{13} & h_8 & h_7 & h_{10} & h_9 \\
g_2 & g_4 & e & h_6 & g_1 & h_5 & h_1 & h_2 & g_9 & g_8 & g_6 & g_7 & h_4 & h_3 & g_5 & g_3 & h_{11} & h_{12} & h_{10} & h_9 & h_7 & h_8 & h_{14} & h_{13} \\
g_3 & g_5 & h_5 & e & h_6 & g_1 & h_{10} & h_9 & h_8 & h_7 & h_{14} & h_{13} & h_{12} & h_{11} & g_2 & g_4 & g_9 & g_8 & g_7 & g_6 & h_4 & h_3 & h_2 & h_1 \\
g_4 & g_2 & g_1 & h_5 & e & h_6 & g_7 & g_6 & h_3 & h_4 & h_2 & h_1 & g_8 & g_9 & g_3 & g_5 & h_8 & h_7 & h_{13} & h_{14} & h_{12} & h_{11} & h_9 & h_{10} \\
g_5 & g_3 & h_6 & g_1 & h_5 & e & h_{13} & h_{14} & h_{11} & h_{12} & h_9 & h_{10} & h_7 & h_8 & g_4 & g_2 & h_3 & h_4 & h_1 & h_2 & g_8 & g_9 & g_6 & g_7 \\
g_6 & h_1 & h_2 & h_7 & g_7 & h_{11} & e & g_4 & h_{13} & h_{10} & g_1 & g_2 & h_9 & h_{14} & h_8 & h_{12} & g_3 & h_5 & h_3 & g_9 & g_5 & h_6 & g_8 & h_4 \\
g_7 & h_2 & h_1 & h_8 & g_6 & h_{12} & g_4 & e & h_9 & h_{14} & g_2 & g_1 & h_{13} & h_{10} & h_7 & h_{11} & h_5 & g_3 & g_8 & h_4 & h_6 & g_5 & h_3 & g_9 \\
g_8 & h_3 & g_9 & h_9 & h_4 & h_{13} & h_{11} & h_8 & e & g_2 & h_7 & h_{12} & g_1 & g_4 & h_{14} & h_{10} & h_1 & g_7 & g_3 & h_6 & g_6 & h_2 & g_5 & h_5 \\
g_9 & h_4 & g_8 & h_{10} & h_3 & h_{14} & h_7 & h_{12} & g_2 & e & h_{11} & h_8 & g_4 & g_1 & h_{13} & h_9 & g_6 & h_2 & h_6 & g_3 & h_1 & g_7 & h_5 & g_5 \\
h_1 & g_6 & g_7 & h_{11} & h_2 & h_7 & g_2 & g_1 & h_{14} & h_9 & g_4 & e & h_{10} & h_{13} & h_{12} & h_8 & h_6 & g_5 & h_4 & g_8 & h_5 & g_3 & g_9 & h_3 \\
h_2 & g_7 & g_6 & h_{12} & h_1 & h_8 & g_1 & g_2 & h_{10} & h_{13} & e & g_4 & h_{14} & h_9 & h_{11} & h_7 & g_5 & h_6 & g_9 & h_3 & g_3 & h_5 & h_4 & g_8 \\
h_3 & g_8 & h_4 & h_{13} & g_9 & h_9 & h_{12} & h_7 & g_4 & g_1 & h_8 & h_{11} & g_2 & e & h_{10} & h_{14} & h_2 & g_6 & h_5 & g_5 & g_7 & h_1 & h_6 & g_3 \\
h_4 & g_9 & h_3 & h_{14} & g_8 & h_{10} & h_8 & h_{11} & g_1 & g_4 & h_{12} & h_7 & e & g_2 & h_9 & h_{13} & g_7 & h_1 & g_5 & h_5 & h_2 & g_6 & g_3 & h_6 \\
h_5 & h_6 & g_3 & g_4 & g_5 & g_2 & h_{14} & h_{13} & h_7 & h_8 & h_{10} & h_9 & h_{11} & h_{12} & g_1 & e & h_4 & h_3 & g_6 & g_7 & g_9 & g_8 & h_1 & h_2 \\
h_6 & h_5 & g_5 & g_2 & g_3 & g_4 & h_9 & h_{10} & h_{12} & h_{11} & h_{13} & h_{14} & h_8 & h_7 & e & g_1 & g_8 & g_9 & h_2 & h_1 & h_3 & h_4 & g_7 & g_6 \\
h_7 & h_{11} & h_8 & g_6 & h_{12} & h_1 & g_9 & h_3 & h_5 & g_3 & h_4 & g_8 & h_6 & g_5 & h_2 & g_7 & h_{10} & h_{13} & g_4 & e & h_{14} & h_9 & g_2 & g_1 \\
h_8 & h_{12} & h_7 & g_7 & h_{11} & h_2 & h_4 & g_8 & g_3 & h_5 & g_9 & h_3 & g_5 & h_6 & h_1 & g_6 & h_{14} & h_9 & e & g_4 & h_{10} & h_{13} & g_1 & g_2 \\
h_9 & h_{13} & h_{14} & g_8 & h_{10} & h_3 & h_6 & g_3 & g_7 & h_1 & h_5 & g_5 & h_2 & g_6 & g_9 & h_4 & g_2 & e & h_8 & h_{11} & g_4 & g_1 & h_{12} & h_7 \\
h_{10} & h_{14} & h_{13} & g_9 & h_9 & h_4 & g_3 & h_6 & h_2 & g_6 & g_5 & h_5 & g_7 & h_1 & g_8 & h_3 & e & g_2 & h_{12} & h_7 & g_1 & g_4 & h_8 & h_{11} \\
h_{11} & h_7 & h_{12} & h_1 & h_8 & g_6 & g_8 & h_4 & g_5 & h_6 & h_3 & g_9 & g_3 & h_5 & g_7 & h_2 & h_9 & h_{14} & g_1 & g_2 & h_{13} & h_{10} & e & g_4 \\
h_{12} & h_8 & h_{11} & h_2 & h_7 & g_7 & h_3 & g_9 & h_6 & g_5 & g_8 & h_4 & h_5 & g_3 & g_6 & h_1 & h_{13} & h_{10} & g_2 & g_1 & h_9 & h_{14} & g_4 & e \\
h_{13} & h_9 & h_{10} & h_3 & h_{14} & g_8 & g_5 & h_5 & g_6 & h_2 & g_3 & h_6 & h_1 & g_7 & h_4 & g_9 & g_1 & g_4 & h_7 & h_{12} & e & g_2 & h_{11} & h_8 \\
h_{14} & h_{10} & h_9 & h_4 & h_{13} & g_9 & h_5 & g_5 & h_1 & g_7 & h_6 & g_3 & g_6 & h_2 & h_3 & g_8 & g_4 & g_1 & h_{11} & h_8 & g_2 & e & h_7 & h_{12}
\end{array}
\tag{19}
$$

It is noted, ahead of time, the canonical naming function does not assign the ten highest values to the set $\{e, g_1, g_2, \ldots, g_9\}$. Some h_i objects will have a higher numeric value than some g_i. The smallest order of any non trivial object is 2. It is obvious $e = 23$, $23 = a$, $22 = b$ for two second order objects, a, b, that commute. The possible pairs are:

$$
\begin{array}{ll}
\{g_1, g_2\} & \{g_4, g_6\} \\
\{g_1, g_4\} & \{g_4, g_7\} \\
\{g_2, g_4\} & \{g_6, g_7\} \\
\{g_1, g_3\} & \{g_2, g_8\} \\
\{g_1, g_5\} & \{g_2, g_9\} \\
\{g_3, g_5\} & \{g_8, g_9\}
\end{array}
\tag{20}
$$

Let a, b any of these pairs; the pairs can be used in either order because they are not ordered pairs. For example, a naming can be $a = g_1$, $b = g_2$, or $a = g_2$, $b = g_1$. Any of the pairs above determine the naming function $e = 23$, $a = 22$, $b = 21$, $x_1 = 20$ with table

$$
\begin{array}{cccc}
e & a & b & x_1 \\
a & e & x_1 & b \\
b & x_1 & e & a \\
x_1 & b & a & e
\end{array}
\quad .
$$

To maximize the representation, find a, b, x_1 such that $\{e, a, b, x_1\}$ forms the Klein 4-group. In fact, the triads

$$
\{g_1, g_2, g_4\} \qquad \{g_1, g_3, g_5\} \qquad \{g_4, g_6, g_7\} \qquad \{g_2, g_8, g_9\} \tag{21}
$$

form the Klein 4-group. Given any one of these triads, it is unknown which objects will be a and b. For example, using $\{g_1, g_2, g_4\}$, who should be defined as a, b, x_1? All the non trivial objects of $K(4)$ are equivalent, so this can not be decided yet.

Add a new object c_1 and $x_2 = a * c_1$.

$$\begin{array}{cccccc} e & a & b & x_1 & c_1 & x_2 \\ a & e & x_1 & b & & \\ b & x_1 & e & a & & \\ x_1 & b & a & e & & \\ c_1 & x_2 & & & & \\ x_2 & c_1 & & & & \end{array}$$

Another new object is needed also, $c_2 = b * c_1$, then $x_3 = a * c_2$.

$$\begin{array}{cccccccc} e & a & b & x_1 & c_1 & x_2 & c_2 & x_3 \\ a & e & x_1 & b & & & & \\ b & x_1 & e & a & & & & \\ x_1 & b & a & e & & & & \\ c_1 & x_2 & c_2 & x_3 & & & & \\ x_2 & c_1 & x_3 & c_2 & & & & \\ c_2 & x_3 & c_1 & x_2 & & & & \\ x_3 & c_2 & x_2 & c_1 & & & & \end{array}$$

In summary, the canonical naming function will involve one of the Klein 4-subgroups $\{e, a, b, x_1\}$ of Δ_4, and an object c_1 that commutes with a, if it should exist. This maximizes the table. There are several options to do this. In fact, all the candidate naming functions admit an object c_1 that commutes with a. This gives the table

$$\begin{array}{cccccccc} e & a & b & x_1 & c_1 & x_2 & c_2 & x_3 \\ a & e & x_1 & b & x_2 & c_1 & & \\ b & x_1 & e & a & & & & \\ x_1 & b & a & e & & & & \\ c_1 & x_2 & c_2 & x_3 & & & & \\ x_2 & c_1 & x_3 & c_2 & & & & \\ c_2 & x_3 & c_1 & x_2 & & & & \\ x_3 & c_2 & x_2 & c_1 & & & & \end{array}$$

determined by the equations

$$\begin{aligned} e &= a^2 = b^2 \\ a * b &= b * a \\ a * c_1 &= c_1 * a. \end{aligned}$$

For the triads in (21), it is necessary to find an object c_1 that commutes with a. For example, all the objects in $\{g_1, g_2, g_4\}$ commute with at least one object not in that set. In the case of $\{g_2, g_8, g_9\}$, only g_2 commutes with objects not in that list. This means if the triad $\{g_2, g_8, g_9\}$ is chosen, then $a = g_2$. For $\{g_1, g_3, g_5\}$ it must be true $a = g_1$, and for $\{g_4, g_6, g_7\}$ the naming $a = g_4$ must be given. The objects that commute with each second order object g_i are listed.

$$\begin{array}{ccc} Comm(g_1) = \{g_2, g_4, g_3, g_5, h_5, h_6\} & Comm(g_2) = \{g_1, g_4, g_8, g_9, h_3, h_4\} & Comm(g_3) = \{g_1, g_5\} \\ Comm(g_4) = \{g_1, g_2, g_6, g_7, h_1, h_2\} & Comm(g_5) = \{g_1, g_3\} & Comm(g_6) = \{g_4, g_7\} \\ Comm(g_7) = \{g_4, g_6\} & Comm(g_8) = \{g_2, g_9\} & Comm(g_9) = \{g_2, g_8\} \end{array} \tag{22}$$

This information reduces the possible naming functions because a little more is known about a. The possible naming functions are more than would be practical to list, but they are easy to describe. An object $a \in \{g_1, g_2, g_4\}$ is needed, along with a second order object b that together determine the subgroup $K(4)$. For example, if $a = g_4$, then one can choose $b \in \{g_1, g_2, g_6, g_7\}$; find a second order object that commutes with $a = g_4$. In the case of $a = g_1$, one must choose $b \in \{g_2, g_4, g_3, g_5\}$. If $a = g_2$ then $b \in \{g_1, g_4, g_8, g_9\}$. After determining the subgroup $K(4)$, an object c_1 that commutes with a is needed. The expressions of (22) determine which combinations allow c_1. Start representing naming functions with finite sequences; in the form $(a, b, x_1, c_1, x_2, c_2, x_3)$. For example,

the naming function $a = g_4$, $b = g_2$, $x_1 = a * b = g_1$, $c_1 = g_7$, $x_2 = a * c_1 = g_6$, $c_2 = b * c_1 = h_1$, $x_3 = a * c_2 = h_2$ is given by the expression $(g_4, g_2, g_1, g_7, g_6, h_1, h_2)$. It is already known $a \in g_1, g_2, g_4$ for any of the triads giving $K(4)$. Choose a second order object, b, that commutes with a, and then choose an object c_1 that also commutes with a. All possible naming functions are listed below.

$$
\begin{array}{lll}
(g_1,g_2,g_4,g_3,g_5,h_5,h_6) & (g_2,g_1,g_4,g_8,g_9,h_3,h_4) & (g_4,g_1,g_2,g_6,g_7,h_1,h_2) \\
(g_1,g_2,g_4,g_5,g_3,h_6,h_5) & (g_2,g_1,g_4,g_9,g_8,h_4,h_3) & (g_4,g_1,g_2,g_7,g_6,h_2,h_1) \\
(g_1,g_2,g_4,h_5,h_6,g_3,g_5) & (g_2,g_1,g_4,h_3,h_4,g_8,g_9) & (g_4,g_1,g_2,h_1,h_2,g_6,g_7) \\
(g_1,g_2,g_4,h_6,h_5,g_5,g_3) & (g_2,g_1,g_4,h_4,h_3,g_9,g_8) & (g_4,g_1,g_2,h_2,h_1,g_7,g_6) \\
 & & \\
(g_1,g_4,g_2,g_3,g_5,h_6,h_5) & (g_2,g_4,g_1,g_8,g_9,h_4,h_3) & (g_4,g_2,g_1,g_6,g_7,h_2,h_1) \\
(g_1,g_4,g_2,g_5,g_3,h_5,h_6) & (g_2,g_4,g_1,g_9,g_8,h_3,h_4) & (g_4,g_2,g_1,g_7,g_6,h_1,h_2) \\
(g_1,g_4,g_2,h_5,h_6,g_5,g_3) & (g_2,g_4,g_1,h_3,h_4,g_9,g_8) & (g_4,g_2,g_1,h_1,h_2,g_7,g_6) \\
(g_1,g_4,g_2,h_6,h_5,g_3,g_5) & (g_2,g_4,g_1,h_4,h_3,g_8,g_9) & (g_4,g_2,g_1,h_2,h_1,g_6,g_7) \\
 & & \\
(g_1,g_3,g_5,g_2,g_4,h_6,h_5) & (g_2,g_8,g_9,g_1,g_4,h_4,h_3) & (g_4,g_6,g_7,g_1,g_2,h_2,h_1) \\
(g_1,g_3,g_5,g_4,g_2,h_5,h_6) & (g_2,g_8,g_9,g_4,g_1,h_3,h_4) & (g_4,g_6,g_7,g_2,g_1,h_1,h_2) \\
(g_1,g_3,g_5,h_5,h_6,g_4,g_2) & (g_2,g_8,g_9,h_3,h_4,g_4,g_1) & (g_4,g_6,g_7,h_1,h_2,g_2,g_1) \\
(g_1,g_3,g_5,h_6,h_5,g_2,g_4) & (g_2,g_8,g_9,h_4,h_3,g_1,g_4) & (g_4,g_6,g_7,h_2,h_1,g_1,g_2) \\
 & & \\
(g_1,g_5,g_3,g_2,g_4,h_5,h_6) & (g_2,g_9,g_8,g_1,g_4,h_3,h_4) & (g_4,g_7,g_6,g_1,g_4,h_1,h_2) \\
(g_1,g_5,g_3,g_4,g_2,h_6,h_5) & (g_2,g_9,g_8,g_4,g_1,h_4,h_3) & (g_4,g_7,g_6,g_2,g_1,h_2,h_1) \\
(g_1,g_5,g_3,h_5,h_6,g_2,g_4) & (g_2,g_9,g_8,h_3,h_4,g_1,g_4) & (g_4,g_7,g_6,h_1,h_2,g_1,g_2) \\
(g_1,g_5,g_3,h_6,h_5,g_4,g_2) & (g_2,g_9,g_8,h_4,h_3,g_4,g_1) & (g_4,g_7,g_6,h_2,h_1,g_2,g_1)
\end{array}
\tag{23}
$$

In (22) it can also be observed that given any choice of $K(4) = \{e, a, b, x_1\}$, there is no group element $g \notin K(4)$ that commutes with both a and b. That is to say, $x_1 = a * b$ is the only element of Δ_4 that commutes with a and b. None of the candidate triads satisfy $a * c_1 = c_1 * a$ and $b * c_1 = c_1 * b$ simultaneously. The highest valued object that can be in the position of $c_1 * b$, is x_3. Each of the finite sequences above satisfies $a * c_1 = c_1 * a$ and $x_3 = c_1 * b$. Any one of the naming functions in (23) will give the table

$$
\begin{array}{cccccccc}
e & a & b & x_1 & c_1 & x_2 & c_2 & x_3 \\
a & e & x_1 & b & x_2 & c_1 & & \\
b & x_1 & e & a & x_3 & c_2 & & \\
x_1 & b & a & e & c_2 & x_3 & & \\
c_1 & x_2 & c_2 & x_3 & & & & \\
x_2 & c_1 & x_3 & c_2 & & & & \\
c_2 & x_3 & c_1 & x_2 & & & & \\
x_3 & c_2 & x_2 & c_1 & & & &
\end{array}
\quad .
$$

It is possible to choose c_1 with the additional restraint $|c_1| = 2$, maximizing the representation. The naming functions

$$
\begin{array}{lll}
(g_1,g_2,g_4,g_3,g_5,h_5,h_6) & (g_2,g_1,g_4,g_8,g_9,h_3,h_4) & (g_4,g_1,g_2,g_6,g_7,h_1,h_2) \\
(g_1,g_2,g_4,g_5,g_3,h_6,h_5) & (g_2,g_1,g_4,g_9,g_8,h_4,h_3) & (g_4,g_1,g_2,g_7,g_6,h_2,h_1) \\
 & & \\
(g_1,g_4,g_2,g_3,g_5,h_6,h_5) & (g_2,g_4,g_1,g_8,g_9,h_4,h_3) & (g_4,g_2,g_1,g_6,g_7,h_2,h_1) \\
(g_1,g_4,g_2,g_5,g_3,h_5,h_6) & (g_2,g_4,g_1,g_9,g_8,h_3,h_4) & (g_4,g_2,g_1,g_7,g_6,h_1,h_2) \\
 & & \\
(g_1,g_3,g_5,g_2,g_4,h_6,h_5) & (g_2,g_8,g_9,g_1,g_4,h_4,h_3) & (g_4,g_6,g_7,g_1,g_2,h_2,h_1) \\
(g_1,g_3,g_5,g_4,g_2,h_5,h_6) & (g_2,g_8,g_9,g_4,g_1,h_3,h_4) & (g_4,g_6,g_7,g_2,g_1,h_1,h_2) \\
 & & \\
(g_1,g_5,g_3,g_2,g_4,h_5,h_6) & (g_2,g_9,g_8,g_1,g_4,h_3,h_4) & (g_4,g_7,g_6,g_1,g_4,h_1,h_2) \\
(g_1,g_5,g_3,g_4,g_2,h_6,h_5) & (g_2,g_9,g_8,g_4,g_1,h_4,h_3) & (g_4,g_7,g_6,g_2,g_1,h_2,h_1)
\end{array}
\tag{24}
$$

give the Dihedral Group

$$\begin{array}{cccccccc} e & a & b & x_1 & c_1 & x_2 & c_2 & x_3 \\ a & e & x_1 & b & x_2 & c_1 & x_3 & c_2 \\ b & x_1 & e & a & x_3 & c_2 & x_2 & c_1 \\ x_1 & b & a & e & c_2 & x_3 & c_1 & x_2 \\ c_1 & x_2 & c_2 & x_3 & e & a & b & x_1 \\ x_2 & c_1 & x_3 & c_2 & a & e & x_1 & b \\ c_2 & x_3 & c_1 & x_2 & x_1 & b & a & e \\ x_3 & c_2 & x_2 & c_1 & b & x_1 & e & a \end{array} .$$

Add a new object d_1 to the table above. The operation $b * d_1$ is a new object, d_2. The operation $c_1 * d_1$ is also a new object, p_1. Finally, $p_2 = b * p_1$ to maximize the representation.

$$\begin{array}{cccccccc|cccccccc} e & a & b & x_1 & c_1 & x_2 & c_2 & x_3 & d_1 & x_4 & d_2 & x_5 & p_1 & x_6 & p_2 & x_7 \\ a & e & x_1 & b & x_2 & c_1 & x_3 & c_2 & & & & & & & & \\ b & x_1 & e & a & x_3 & c_2 & x_2 & c_1 & & & & & & & & \\ x_1 & b & a & e & c_2 & x_3 & c_1 & x_2 & & & & & & & & \\ c_1 & x_2 & c_2 & x_3 & e & a & b & x_1 & & & & & & & & \\ x_2 & c_1 & x_3 & c_2 & a & e & x_1 & b & & & & & & & & \\ c_2 & x_3 & c_1 & x_2 & x_1 & b & a & e & & & & & & & & \\ x_3 & c_2 & x_2 & c_1 & b & x_1 & e & a & & & & & & & & \\ \hline d_1 & x_4 & d_2 & x_5 & p_1 & x_6 & p_2 & x_7 & & & & & & & & \\ x_4 & d_1 & x_5 & d_2 & x_6 & p_1 & x_7 & p_2 & & & & & & & & \\ d_2 & x_5 & d_1 & x_4 & x_7 & p_2 & x_6 & p_1 & & & & & & & & \\ x_5 & d_2 & x_4 & d_1 & p_2 & x_7 & p_1 & x_6 & & & & & & & & \\ p_1 & x_6 & p_2 & x_7 & d_1 & x_4 & d_2 & x_5 & & & & & & & & \\ x_6 & p_1 & x_7 & p_2 & x_4 & d_1 & x_5 & d_2 & & & & & & & & \\ p_2 & x_7 & p_1 & x_6 & x_5 & d_2 & x_4 & d_1 & & & & & & & & \\ x_7 & p_2 & x_6 & p_1 & d_2 & x_5 & d_1 & x_4 & & & & & & & & \end{array}$$

How is d_1 chosen? If there is another object(s) that commutes with a, it would be candidate to d_1. However, in each of the naming functions, there are no more objects that commute with a. The next largest value that can be placed in $d_1 * a$, is $d_2 = b * d_1$. Observe that only some of the naming functions above will satisfy this condition. For example, the naming function $a = g_1$, $b = g_3$ is disqualified from being a canonical naming function because there is no element $d_1 \notin D_8$ such that $d_1 * a = b * d_1$. The only cases when such an object d_1 exists is if $a, b \in \{g_1, g_2, g_4\}$. The easiest way to find the candidates for d_1, is to compare the row of a and the column of b. If the i-th object in the row of a coincides with the i-th object in the column of b, then the i-th object on the first row (or first column) is a candidate for d_1. For example, with the naming function $(g_1, g_2, g_4, g_3, g_5, h_5, h_6)$, the candidates for d_1 are the objects $g_6, g_7, h_1, h_2, h_9, h_{10}, h_{13}, h_{14}$. The candidates for d_1 are determined by a, b. If $a = g_1$ and $b = g_3$ there is no candidate for d_1. If $a = g_1$ and $b = g_4$ the candidates for d_1 are $g_8, g_9, h_3, h_4, h_7, h_8, h_{11}, h_{12}$, etc. The naming functions that satisfy this condition are those that have a, b in g_1, g_2, g_4. More is known about the canonical naming function. The Klein group $K(4) = \{e, g_1, g_2, g_4\}$, in any order, gives the first four objects of the naming function. Then, a second order object c_1 that commutes with a must be chosen. Then choose d_1 so that $b * d_1 = d_1 * a$. Below, twelve naming functions are given. Each of these has eight possible candidates for d_1. There is a total of ninety-six possible naming functions.

$$\begin{array}{lll} (g_1, g_2, g_4, g_3, g_5, h_5, h_6, d_1, \ldots, x_7) & (g_2, g_1, g_4, g_8, g_9, h_3, h_4, d_1, \ldots, x_7) & (g_4, g_1, g_2, g_6, g_7, h_1, h_2, d_1, \ldots, x_7) \\ (g_1, g_2, g_4, g_5, g_3, h_6, h_5, d_1, \ldots, x_7) & (g_2, g_1, g_4, g_9, g_8, h_4, h_3, d_1, \ldots, x_7) & (g_4, g_1, g_2, g_7, g_6, h_2, h_1, d_1, \ldots, x_7) \\ (g_1, g_4, g_2, g_3, g_5, h_6, h_5, d_1, \ldots, x_7) & (g_2, g_4, g_1, g_8, g_9, h_4, h_3, d_1, \ldots, x_7) & (g_4, g_2, g_1, g_6, g_7, h_2, h_1, d_1, \ldots, x_7) \\ (g_1, g_4, g_2, g_5, g_3, h_5, h_6, d_1, \ldots, x_7) & (g_2, g_4, g_1, g_9, g_8, h_3, h_4, d_1, \ldots, x_7) & (g_4, g_2, g_1, g_7, g_6, h_1, h_2, d_1, \ldots, x_7) \end{array} \tag{25}$$

It is possible to reduce the naming functions, further. Some of the candidate naming functions (not all) satisfy $d_1 * b = a * d_1$, which maximizes the representation. Keep the naming functions that satisfy $b * d_1 = d_1 * a$ and

$d_1 * b = a * d_1$, simultaneously. In the case of $a = g_1$, $b = g_2$ the candidates for d_1 are reduced to g_6, g_7, h_1, h_2.

e	a	b	x_1	c_1	x_2	c_2	x_3	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7
a	e	x_1	b	x_2	c_1	x_3	c_2	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1
b	x_1	e	a	x_3	c_2	x_2	c_1	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2
x_1	b	a	e	c_2	x_3	c_1	x_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6
c_1	x_2	c_2	x_3	e	a	b	x_1								
x_2	c_1	x_3	c_2	a	e	x_1	b								
c_2	x_3	c_1	x_2	x_1	b	a	e								
x_3	c_2	x_2	c_1	b	x_1	e	a								
d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7								
x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2								
d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1								
x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6								
p_1	x_6	p_2	x_7	d_1	x_4	d_2	x_5								
x_6	p_1	x_7	p_2	x_4	d_1	x_5	d_2								
p_2	x_7	p_1	x_6	x_5	d_2	x_4	d_1								
x_7	p_2	x_6	p_1	d_2	x_5	d_1	x_4								

The naming functions $(a, b, x_1, c_1, x_2, c_2, x_3, d_1, x_4, d_2, x_5, p_1, x_6, p_2, x_7)$ are given below.

$(g_1, g_2, g_4, g_3, g_5, h_5, h_6, g_6, h_1, h_2, g_7, h_7, h_{11}, h_8, h_{12})$
$(g_1, g_2, g_4, g_3, g_5, h_5, h_6, g_7, h_2, h_1, g_6, h_8, h_{12}, h_7, h_{11})$
$(g_1, g_2, g_4, g_3, g_5, h_5, h_6, h_1, g_6, g_7, h_2, h_{11}, h_7, h_8, h_{12})$
$(g_1, g_2, g_4, g_3, g_5, h_5, h_6, h_2, g_7, g_6, h_1, h_{12}, h_8, h_7, h_{11})$

$(g_2, g_1, g_4, g_8, g_9, h_3, h_4, g_6, h_2, h_1, g_7, h_{13}, h_{10}, h_9, h_{14})$
$(g_2, g_1, g_4, g_8, g_9, h_3, h_4, g_7, h_1, h_2, g_6, h_9, h_{14}, h_{13}, h_{10})$
$(g_2, g_1, g_4, g_8, g_9, h_3, h_4, h_1, g_7, g_6, h_2, h_{14}, h_9, h_{13}, h_{10})$
$(g_2, g_1, g_4, g_8, g_9, h_3, h_4, h_2, g_6, g_7, h_1, h_{10}, h_{13}, h_{14}, h_9)$

$(g_1, g_2, g_4, g_5, g_3, h_6, h_5, g_6, h_1, h_2, g_7, h_{11}, h_7, h_{12}, h_8)$
$(g_1, g_2, g_4, g_5, g_3, h_6, h_5, g_7, h_2, h_1, g_6, h_{12}, h_8, h_7, h_{11})$
$(g_1, g_2, g_4, g_5, g_3, h_6, h_5, h_1, g_6, g_7, h_2, h_7, h_{11}, h_8, h_{12})$
$(g_1, g_2, g_4, g_5, g_3, h_6, h_5, h_2, g_7, g_6, h_1, h_8, h_{12}, h_7, h_{11})$

$(g_2, g_1, g_4, g_9, g_8, h_4, h_3, g_6, h_2, h_1, g_7, h_{10}, h_{13}, h_{14}, h_9)$
$(g_2, g_1, g_4, g_9, g_8, h_4, h_3, g_7, h_1, h_2, g_6, h_{14}, h_9, h_{10}, h_{13})$
$(g_2, g_1, g_4, g_9, g_8, h_4, h_3, h_1, g_7, g_6, h_2, h_9, h_{14}, h_{13}, h_{10})$
$(g_2, g_1, g_4, g_9, g_8, h_4, h_3, h_2, g_6, g_7, h_1, h_{13}, h_{10}, h_9, h_{14})$

$(g_1, g_4, g_2, g_3, g_5, h_5, h_6, g_8, h_3, h_4, g_9, h_9, h_{13}, h_{10}, h_{14})$
$(g_1, g_4, g_2, g_3, g_5, h_5, h_6, g_9, h_4, h_3, g_8, h_{10}, h_{14}, h_9, h_{13})$
$(g_1, g_4, g_2, g_3, g_5, h_5, h_6, h_3, g_8, g_9, h_4, h_{13}, h_9, h_{14}, h_{10})$
$(g_1, g_4, g_2, g_3, g_5, h_5, h_6, h_4, g_9, g_8, h_3, h_{14}, h_{10}, h_9, h_{13})$

$(g_2, g_4, g_1, g_8, g_9, h_3, h_4, g_3, h_5, h_6, g_5, h_8, h_7, h_{11}, h_{12})$
$(g_2, g_4, g_1, g_8, g_9, h_3, h_4, g_5, h_6, h_5, g_3, h_{11}, h_{12}, h_8, h_7)$
$(g_2, g_4, g_1, g_8, g_9, h_3, h_4, h_5, g_3, g_5, h_6, h_7, h_8, h_{12}, h_{11})$
$(g_2, g_4, g_1, g_8, g_9, h_3, h_4, h_6, g_5, g_3, h_5, h_{12}, h_{11}, h_7, h_8)$

$(g_1, g_4, g_2, g_5, g_3, h_6, h_5, g_8, h_3, h_4, g_9, h_{13}, h_9, h_{14}, h_{10})$
$(g_1, g_4, g_2, g_5, g_3, h_6, h_5, g_9, h_4, h_3, g_8, h_{14}, h_{10}, h_{13}, h_9)$
$(g_1, g_4, g_2, g_5, g_3, h_6, h_5, h_3, g_8, g_9, h_4, h_9, h_{13}, h_{10}, h_{14})$
$(g_1, g_4, g_2, g_5, g_3, h_6, h_5, h_4, g_9, g_8, h_3, h_{10}, h_{14}, h_9, h_{13})$

$(g_2, g_4, g_1, g_9, g_8, h_4, h_3, g_3, h_5, h_6, g_5, h_7, h_8, h_{12}, h_{11})$
$(g_2, g_4, g_1, g_9, g_8, h_4, h_3, g_5, h_6, h_5, g_3, h_{12}, h_{11}, h_7, h_8)$
$(g_2, g_4, g_1, g_9, g_8, h_4, h_3, h_5, g_3, g_5, h_6, h_8, h_7, h_{11}, h_{12})$
$(g_2, g_4, g_1, g_9, g_8, h_4, h_3, h_6, g_5, g_3, h_5, h_{11}, h_{12}, h_8, h_7)$

$$
\begin{array}{c}
(g_4, g_1, g_2, g_6, g_7, h_1, h_2, g_8, h_4, h_3, g_9, h_{11}, h_8, h_7, h_{12}) \\
(g_4, g_1, g_2, g_6, g_7, h_1, h_2, g_9, h_3, h_4, g_8, h_7, h_{12}, h_{11}, h_8) \\
(g_4, g_1, g_2, g_6, g_7, h_1, h_2, h_3, g_9, g_8, h_4, h_{12}, h_7, h_8, h_{11}) \\
(g_4, g_1, g_2, g_6, g_7, h_1, h_2, h_4, g_8, g_9, h_3, h_8, h_{11}, h_{12}, h_7) \\
\\
(g_4, g_1, g_2, g_7, g_6, h_2, h_1, g_8, h_4, h_3, g_9, h_8, h_{11}, h_{12}, h_7) \\
(g_4, g_1, g_2, g_7, g_6, h_2, h_1, g_9, h_3, h_4, g_8, h_{12}, h_7, h_8, h_{11}) \\
(g_4, g_1, g_2, g_7, g_6, h_2, h_1, h_3, g_9, g_8, h_4, h_7, h_{12}, h_{11}, h_8) \\
(g_4, g_1, g_2, g_7, g_6, h_2, h_1, h_4, g_8, g_9, h_3, h_{11}, h_8, h_7, h_{12}) \\
\\
(g_4, g_2, g_1, g_6, g_7, h_1, h_2, g_3, h_6, h_5, g_5, h_{10}, h_9, h_{13}, h_{14}) \\
(g_4, g_2, g_1, g_6, g_7, h_1, h_2, g_5, h_5, h_6, g_3, h_{13}, h_{14}, h_{10}, h_9) \\
(g_4, g_2, g_1, g_6, g_7, h_1, h_2, h_5, g_5, g_3, h_6, h_{14}, h_{13}, h_9, h_{10}) \\
(g_4, g_2, g_1, g_6, g_7, h_1, h_2, h_6, g_3, g_5, h_5, h_9, h_{10}, h_{14}, h_{13}) \\
\\
(g_4, g_2, g_1, g_7, g_6, h_2, h_1, g_3, h_6, h_5, g_5, h_9, h_{10}, h_{14}, h_{13}) \\
(g_4, g_2, g_1, g_7, g_6, h_2, h_1, g_5, h_5, h_6, g_3, h_{14}, h_{13}, h_9, h_{10}) \\
(g_4, g_2, g_1, g_7, g_6, h_2, h_1, h_5, g_5, g_3, h_6, h_{13}, h_{14}, h_{10}, h_9) \\
(g_4, g_2, g_1, g_7, g_6, h_2, h_1, h_6, g_3, g_5, h_5, h_{10}, h_9, h_{13}, h_{14})
\end{array}
\tag{26}
$$

One can start to suspect what objects might turn out to be equivalent. For example, it is quite clear g_1, g_2, g_4 might probably be equivalent, and also g_3, g_5, and g_6, g_7, and h_1, h_2, etc. The naming functions of (26) all give a new object $d_1 * c_1$. For example, in the naming function $(g_1, g_2, g_4, g_3, g_5, h_5, h_6, g_6, h_1, h_2, g_7)$, there is a new object $d_1 * c_1 = g_6 * g_3 = h_{10}$. Including this new object, $q_1 = d_1 * c_1$, gives another new object $r_1 = c_1 * q_1$. The object $r_1 = c_1 * q_1$ must be added, along with $x_{10} = a * r_1$, $r_2 = b * r_1$, $x_{11} = a * r_2$.

e	a	b	x_1	c_1	x_2	c_2	x_3	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7	q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}
a	e	x_1	b	x_2	c_1	x_3	c_2	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1	q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1
b	x_1	e	a	x_3	c_2	x_2	c_1	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2	x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}
x_1	b	a	e	c_2	x_3	c_1	x_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6	x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2
c_1	x_2	c_2	x_3	e	a	b	x_1	q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7
x_2	c_1	x_3	c_2	a	e	x_1	b	q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1
c_2	x_3	c_1	x_2	x_1	b	a	e	x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6
x_3	c_2	x_2	c_1	b	x_1	e	a	x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2
d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7																
x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2																
d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1																
x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6																
p_1	x_6	p_2	x_7	d_1	x_4	d_2	x_5																
x_6	p_1	x_7	p_2	x_4	d_1	x_5	d_2																
p_2	x_7	p_1	x_6	x_5	d_2	x_4	d_1																
x_7	p_2	x_6	p_1	d_2	x_5	d_1	x_4																
q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}																
x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2																
q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1																
x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}																
r_1	x_{10}	r_2	x_{11}	q_1	x_8	q_2	x_9																
x_{10}	r_1	x_{11}	r_2	x_8	q_1	x_9	q_2																
r_2	x_{11}	r_1	x_{10}	x_9	q_2	x_8	q_1																
x_{11}	r_2	x_{10}	r_1	q_2	x_9	q_1	x_8																

To maximize the representation, take the naming functions that satisfy $|d_1| = 2$.

$$(g_1, g_2, g_4, g_3, g_5, h_5, h_6, g_6, h_1, h_2, g_7, h_7, h_{11}, h_8, h_{12}, h_{10}, h_{14}, h_{13}, h_9, g_9, h_4, g_8, h_3)$$
$$(g_1, g_2, g_4, g_3, g_5, h_5, h_6, g_7, h_2, h_1, g_6, h_8, h_{12}, h_7, h_{11}, h_9, h_{13}, h_{14}, h_{10}, g_8, h_3, g_9, h_4)$$
$$(g_1, g_2, g_4, g_5, g_3, h_6, h_5, g_6, h_1, h_2, g_7, h_{11}, h_7, h_{12}, h_8, h_{13}, h_9, h_{10}, h_{14}, g_8, h_3, g_9, h_4)$$
$$(g_1, g_2, g_4, g_5, g_3, h_6, h_5, g_7, h_2, h_1, g_6, h_{12}, h_8, h_7, h_{11}, h_{14}, h_{10}, h_{13}, h_9, g_9, h_4, g_8, h_3)$$
$$(g_1, g_4, g_2, g_3, g_5, h_5, h_6, g_8, h_3, h_4, g_9, h_9, h_{13}, h_{10}, h_{14}, h_8, h_{12}, h_{11}, h_7, g_7, h_2, g_6, h_1)$$
$$(g_1, g_4, g_2, g_3, g_5, h_5, h_6, g_9, h_4, h_3, g_8, h_{10}, h_{14}, h_9, h_{13}, h_7, h_{11}, h_{12}, h_8, g_6, h_1, g_7, h_2)$$
$$(g_1, g_4, g_2, g_5, g_3, h_6, h_5, g_8, h_3, h_4, g_9, h_{13}, h_9, h_{14}, h_{10}, h_{11}, h_7, h_8, h_{12}, g_6, h_1, g_7, h_2)$$
$$(g_1, g_4, g_2, g_5, g_3, h_6, h_5, g_9, h_4, h_3, g_8, h_{14}, h_{13}, h_{10}, h_9, h_{12}, h_8, h_7, h_{11}, g_7, h_2, g_6, h_1)$$

$$(g_2, g_1, g_4, g_8, g_9, h_3, h_4, g_6, h_2, h_1, g_7, h_{13}, h_{10}, h_9, h_{14}, h_{11}, h_{12}, h_7, h_8, g_5, h_6, g_3, h_5)$$
$$(g_2, g_1, g_4, g_8, g_9, h_3, h_4, g_7, h_1, h_2, g_6, h_9, h_{14}, h_{13}, h_{10}, h_8, h_7, h_{12}, h_{11}, g_3, h_5, g_5, h_6)$$
$$(g_2, g_1, g_4, g_9, g_8, h_4, h_3, g_6, h_2, h_1, g_7, h_{10}, h_{13}, h_{14}, h_9, h_7, h_8, h_{11}, h_{12}, g_3, h_5, g_5, h_6)$$
$$(g_2, g_1, g_4, g_9, g_8, h_4, h_3, g_7, h_1, h_2, g_6, h_{14}, h_9, h_{10}, h_{13}, h_{12}, h_{11}, h_8, h_7, g_5, h_6, g_3, h_5)$$
$$(g_2, g_4, g_1, g_8, g_9, h_3, h_4, g_3, h_5, h_6, g_5, h_8, h_7, h_{11}, h_{12}, h_9, h_{14}, h_{10}, h_{13}, g_7, h_1, g_6, h_2)$$
$$(g_2, g_4, g_1, g_8, g_9, h_3, h_4, g_5, h_6, h_5, g_3, h_{11}, h_{12}, h_8, h_7, h_{13}, h_{10}, h_{14}, h_9, g_6, h_2, g_7, h_1)$$
$$(g_2, g_4, g_1, g_9, g_8, h_4, h_3, g_3, h_5, h_6, g_5, h_7, h_8, h_{12}, h_{11}, h_{10}, h_{13}, h_9, h_{14}, g_6, h_2, g_7, h_1)$$
$$(g_2, g_4, g_1, g_9, g_8, h_4, h_3, g_5, h_6, h_5, g_3, h_{12}, h_{11}, h_7, h_8, h_{14}, h_9, h_{13}, h_{10}, g_7, h_1, g_6, h_2)$$

$$(g_4, g_1, g_2, g_6, g_7, h_1, h_2, g_8, h_4, h_3, g_9, h_{11}, h_8, h_7, h_{12}, h_{13}, h_{14}, h_9, h_{10}, g_5, h_5, g_3, h_6)$$
$$(g_4, g_1, g_2, g_6, g_7, h_1, h_2, g_9, h_3, h_4, g_8, h_7, h_{12}, h_{11}, h_8, h_{10}, h_9, h_{14}, h_{13}, g_3, h_6, g_5, h_5)$$
$$(g_4, g_1, g_2, g_7, g_6, h_2, h_1, g_8, h_4, h_3, g_9, h_8, h_{11}, h_{12}, h_7, h_9, h_{10}, h_{13}, h_{14}, g_3, h_6, g_5, h_5)$$
$$(g_4, g_1, g_2, g_7, g_6, h_2, h_1, g_9, h_3, h_4, g_8, h_{12}, h_7, h_8, h_{11}, h_{14}, h_{13}, h_{10}, h_9, g_5, h_5, g_3, h_6)$$
$$(g_4, g_2, g_1, g_6, g_7, h_1, h_2, g_3, h_6, h_5, g_5, h_{10}, h_9, h_{13}, h_{14}, h_7, h_{12}, h_8, h_{11}, g_9, h_3, g_8, h_4)$$
$$(g_4, g_2, g_1, g_6, g_7, h_1, h_2, g_5, h_5, h_6, g_3, h_{13}, h_{14}, h_{10}, h_9, h_{11}, h_8, h_7, h_{12}, g_8, h_4, g_9, h_3)$$
$$(g_4, g_2, g_1, g_7, g_6, h_2, h_1, g_3, h_6, h_5, g_5, h_9, h_{10}, h_{14}, h_{13}, h_8, h_{11}, h_7, h_{14}, g_8, h_4, g_9, h_3)$$
$$(g_4, g_2, g_1, g_7, g_6, h_2, h_1, g_5, h_5, h_6, g_3, h_{14}, h_{13}, h_9, h_{10}, h_{12}, h_7, h_{11}, h_8, g_9, h_3, g_8, h_4)$$

These naming functions give the table

e	a	b	x_1	c_1	x_2	c_2	x_3	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7	q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}
a	e	x_1	b	x_2	c_1	x_3	c_2	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1	q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1
b	x_1	e	a	x_3	c_2	x_2	c_1	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2	x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}
x_1	b	a	e	c_2	x_3	c_1	x_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6	x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2
c_1	x_2	c_2	x_3	e	a	b	x_1	q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7
x_2	c_1	x_3	c_2	a	e	x_1	b	q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1
c_2	x_3	c_1	x_2	x_1	b	a	e	x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6
x_3	c_2	x_2	c_1	b	x_1	e	a	x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2
d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7	e	a	b	x_1	c_1	x_2	c_2	x_3								
x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2	b	x_1	e	a	x_3	c_2	x_2	c_1								
d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1	a	e	x_1	b	x_2	c_1	x_3	c_2								
x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6	x_1	b	a	e	c_2	x_3	c_1	x_2								
p_1	x_6	p_2	x_7	d_1	x_4	d_2	x_5									e	a	b	x_1	c_1	x_2	c_2	x_3
x_6	p_1	x_7	p_2	x_4	d_1	x_5	d_2									b	x_1	e	a	x_3	c_2	x_2	c_1
p_2	x_7	p_1	x_6	x_5	d_2	x_4	d_1									x_1	b	a	e	c_2	x_3	c_1	x_2
x_7	p_2	x_6	p_1	d_2	x_5	d_1	x_4									a	e	x_1	b	x_2	c_1	x_3	c_2
q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}	c_1	x_2	c_2	x_3	e	a	b	x_1								
x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2	c_2	x_3	c_1	x_2	x_1	b	a	e								
q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1	x_2	c_1	x_3	c_2	a	e	x_1	b								
x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}	x_3	c_2	x_2	c_1	b	x_1	e	a								
r_1	x_{10}	r_2	x_{11}	q_1	x_8	q_2	x_9									c_1	x_2	c_2	c_1	e	a	b	x_1
x_{10}	r_1	x_{11}	r_2	x_8	q_1	x_9	q_2									c_2	x_3	c_1	x_2	x_1	b	a	e
r_2	x_{11}	r_1	x_{10}	x_9	q_2	x_8	q_1									x_3	c_2	x_2	c_1	b	x_1	e	a
x_{11}	r_2	x_{10}	r_1	q_2	x_9	q_1	x_8									x_2	c_1	x_3	c_2	a	e	x_1	b

At this point, No more information can be added to the table. An observation is made to complete the table. All the naming functions satisfy $d_1 * p_1 = r_1$. With this, the table is finished.

e	a	b	x_1	c_1	x_2	c_2	x_3	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7	q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}
a	e	x_1	b	x_2	c_1	x_3	c_2	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1	q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1
b	x_1	e	a	x_3	c_2	x_2	c_1	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2	x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}
x_1	b	a	e	c_2	x_3	c_1	x_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6	x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2
c_1	x_2	c_2	x_3	e	a	b	x_1	q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}	d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7
x_2	c_1	x_3	c_2	a	e	x_1	b	q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1	d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1
c_2	x_3	c_1	x_2	x_1	b	a	e	x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2	x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6
x_3	c_2	x_2	c_1	b	x_1	e	a	x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}	x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2
d_1	x_4	d_2	x_5	p_1	x_6	p_2	x_7	e	a	b	x_1	c_1	x_2	c_2	x_3	r_1	x_{10}	r_2	x_{11}	q_1	x_8	q_2	x_9
x_4	d_1	x_5	d_2	x_6	p_1	x_7	p_2	b	x_1	e	a	x_3	c_2	x_2	c_1	r_2	x_{11}	r_1	x_{10}	x_9	q_2	x_8	q_1
d_2	x_5	d_1	x_4	x_7	p_2	x_6	p_1	a	e	x_1	b	x_2	c_1	x_3	c_2	x_{11}	r_2	x_{10}	r_1	q_2	x_9	q_1	x_8
x_5	d_2	x_4	d_1	p_2	x_7	p_1	x_6	x_1	b	a	e	c_2	x_3	c_1	x_2	x_{10}	r_1	x_{11}	r_2	x_8	q_1	x_9	q_2
p_1	x_6	p_2	x_7	d_1	x_4	d_2	x_5	r_1	x_{10}	r_2	x_{11}	q_1	x_8	q_2	x_9	e	a	b	x_1	c_1	x_2	c_2	x_3
x_6	p_1	x_7	p_2	x_4	d_1	x_5	d_2	r_2	x_{11}	r_1	x_{10}	x_9	q_2	x_8	q_1	b	x_1	e	a	x_3	c_2	x_2	c_1
p_2	x_7	p_1	x_6	x_5	d_2	x_4	d_1	x_{10}	r_1	x_{11}	r_2	x_8	q_1	x_9	q_2	x_1	b	a	e	c_2	x_3	c_1	x_2
x_7	p_2	x_6	p_1	d_2	x_5	d_1	x_4	x_{11}	r_2	x_{10}	r_1	q_2	x_9	q_1	x_8	a	e	x_1	b	x_2	c_1	x_3	c_2
q_1	x_8	q_2	x_9	r_1	x_{10}	r_2	x_{11}	c_1	x_2	c_2	x_3	e	a	b	x_1	p_1	x_6	p_2	x_7	d_1	x_4	d_2	x_5
x_8	q_1	x_9	q_2	x_{10}	r_1	x_{11}	r_2	c_2	x_3	c_1	x_2	x_1	b	a	e	p_2	x_7	p_1	x_6	x_5	d_2	x_4	d_1
q_2	x_9	q_1	x_8	x_{11}	r_2	x_{10}	r_1	x_2	c_1	x_3	c_2	a	e	x_1	b	x_7	p_2	x_6	p_1	d_2	x_5	d_1	x_4
x_9	q_2	x_8	q_1	r_2	x_{11}	r_1	x_{10}	x_3	c_2	x_2	c_1	b	x_1	e	a	x_6	p_1	x_7	p_2	x_4	d_1	x_5	d_2
r_1	x_{10}	r_2	x_{11}	q_1	x_8	q_2	x_9	p_1	x_6	p_2	x_7	d_1	x_4	d_2	x_5	c_1	x_2	c_2	x_3	e	a	b	x_1
x_{10}	r_1	x_{11}	r_2	x_8	q_1	x_9	q_2	p_2	x_7	p_1	x_6	x_5	d_2	x_4	d_1	c_2	x_3	c_1	x_2	x_1	b	a	e
r_2	x_{11}	r_1	x_{10}	x_9	q_2	x_8	q_1	x_6	p_1	x_7	p_2	x_4	d_1	x_5	d_2	x_3	c_2	x_2	c_1	b	x_1	e	a
x_{11}	r_2	x_{10}	r_1	q_2	x_9	q_1	x_8	x_7	p_2	x_6	p_1	d_2	x_5	d_1	x_4	x_2	c_1	x_3	c_2	a	e	x_1	b

The canonical naming functions of the symmetry group Δ_4 have been found, and they provide the twenty four automorphisms of Δ_4. Equivalent objects of the group are identified. The equivalence classes of objects are

$$\{e\}$$
$$\{g_1, g_2, g_4\}$$
$$\{g_3, g_5, g_6, g_7, g_8, g_9\}$$
$$\{h_1, h_2, h_3, h_4, h_5, h_6\}$$
$$\{h_7, h_8, h_9, h_{10}, h_{11}, h_{12}, h_{13}, h_{14}\}$$

The canonical representation is easily obtained from this table in numeric form. To verify isomorphism of two groups, find the numeric tables and these have to coincide.