2017

Improved Subset Generation For The MU-Decoder

Utsav Agarwal
Louisiana State University and Agricultural and Mechanical College

IMPROVED SUBSET GENERATION FOR THE MU-DECODER

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

Electrical Engineering

by
Utsav Agarwal
B.Tech, West Bengal University of Technology, 2012
May 2017

Acknowledgments

I would like to thank my advisor Dr. Ramachandran Vaidyanathan, for his patient guidance and constant support for not only my research but also in making me a better person. I am indeed blessed to research under such a wise pundit. I would also like to thank the committee members Dr. Jerry Trahan and Dr. Konstantin (Costas) Busch for helping me develop the thesis better. I would like to thank the circle of my strength and pride, my family, which includes my parents, brother, grandparents, uncles, aunts and my cousins. I would like to thank my family of friends who made me feel home and supported me throughout this journey. I take this moment to thank my bhalobasha Sonam, for standing by me throughout.

Contents

List of Tables

List of Figures

Abstract

The MU-Decoder is a hardware subset generator that finds use in partial reconfiguration of FPGAs and in numerous other applications. It is capable of generating a set $\mathcal{S}$ of subsets of a large set $\mathbb{Z}_n$ with n elements. If the subsets in $\mathcal{S}$ satisfy the "isomorphic totally-ordered property," then the MU-Decoder works very efficiently to produce a set of u subsets in $O(\log n)$ time and $\Theta(n\sqrt{u}\log n)$ gate cost. In contrast, a naive approach requires $\Theta(un)$ gate cost. We show that this low cost for the MU-Decoder can be achieved without the isomorphism constraint, thereby allowing $\mathcal{S}$ to include a much wider range of subsets. We also show that if additional constraints on the relative sizes of the subsets in $\mathcal{S}$ can be placed, then u subsets can be generated with $\Theta(n\sqrt{u})$ cost. This uses a new hardware enhancement proposed in this thesis. Finally, we show that by properly selecting $\mathcal{S}$ and by using some elements of traditional methods, a set of $u\binom{n}{d}$ subsets can be produced with $\Theta(n\sqrt{u}\log u)$ cost.

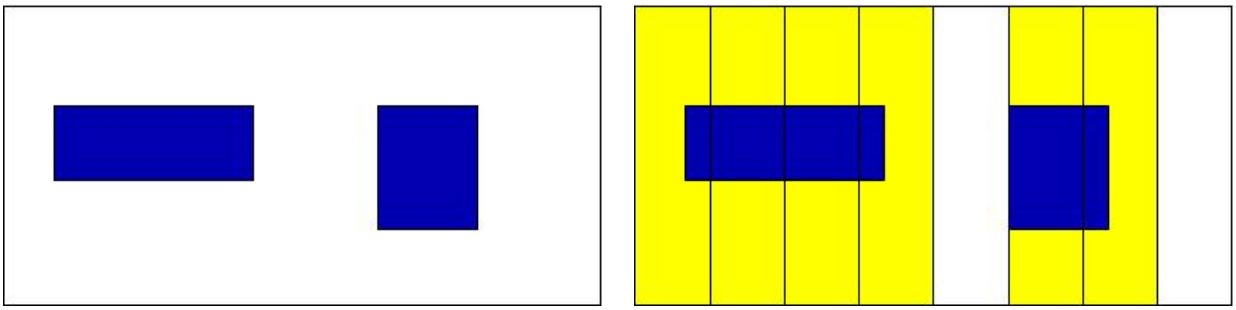
Chapter 1
Introduction

Many modern applications utilize Field Programmable Gate Arrays (FPGAs) [1, 35], which includes intelligent systems [3, 23, 24], scientific applications [8, 13], defense and aerospace systems [19, 25, 27, 34], communication and signal processing [6, 14, 22, 26, 32], instrumentation [10, 30], finance [33].

Partial Reconfiguration [4,5,7,9,18,20,31] (PR) is an important feature of reconfigurable computing that allows a portion of the configuration fabric to be reconfigured at runtime. The feature allows for an efficient use of the chip's real estate. However PR needs to be quick (to be useful at real time). The unit of reconfiguration is typically called a "frame." A frame may contain hundreds or thousands of reconfigurable bits (the entire chip could contain millions). If frames are too large (and granular), a larger portion of the configurable fabric may have to be reconfigured.

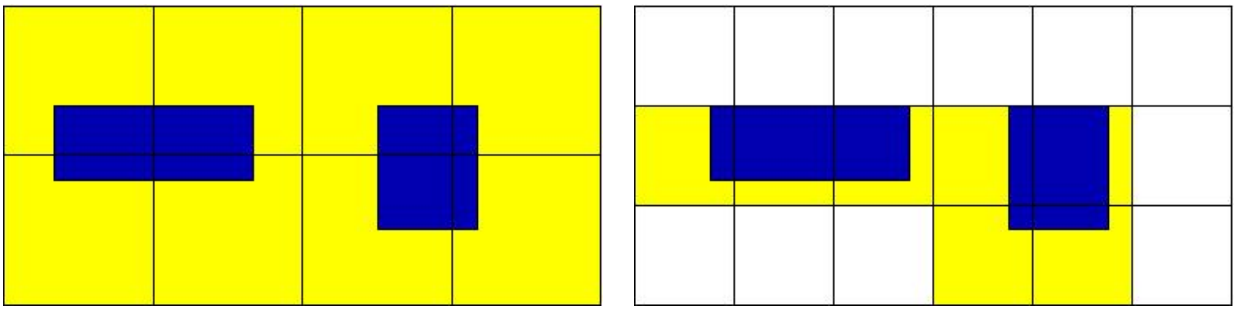
Example 1.1. We consider the illustrative example taken from Ashrafi [2] and Jordan [17]. Figure 1.1 shows us how frame granularity affects the partial configuration of a chip. The blue colored regions represent the area that needs to be reconfigured and the yellow colored regions represent the extra part of the frame that needs to be reconfigured as well during partial configuration. □

This points toward using small frames. However, to configure a frame, it must first be selected and then a data path established to it, before the configuration bits can be input to the frame. Conventional ways to select a frame come down to the use of a 1-hot decoder, that selects one frame at a time (or a 1-element subset of the set of all frames). If we use the illustration of Example 1.1 to make frames small, then there will be many frames to reconfigure and many iterations of using the 1-hot decoder; essentially a time-consuming exercise. The MU-Decoder [16] shows how a subset (of more than 1 frame) can be selected. The scan-path architecture [2] shows how the configuration bits can be delivered to the



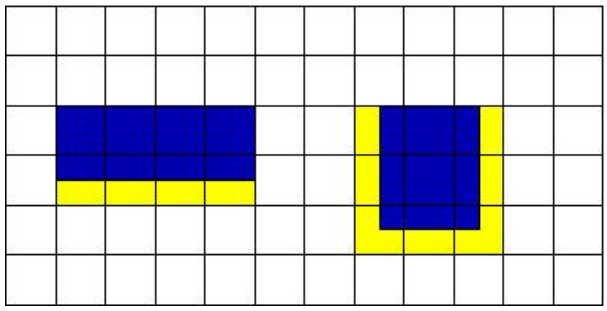
(a)PR area: 15%

(b)Conf area: 75% (6/8 frames)



(c)Conf area: 100% (all 8 frames)

(d)Conf area: 39% (7/18 frames)



(e)Conf area: 24% (17/72 frames)

Figure 1.1: An Illustration of Frame Granularity (a) shows the location of the part needing reconfiguration. It occupies only 15% of the total area. Part (b)-(e) show the impact of various frame size and shapes. For example, in part (c), 6 of the 8 "tall" frames are needed to cover the blue area. So while the partial reconfiguration (PR) area is 15%, the configuration area is 6/8 or 75%. The square frames in part(d) fare worse, requiring a total reconfiguration. With smaller frames, the configuration area reduces.

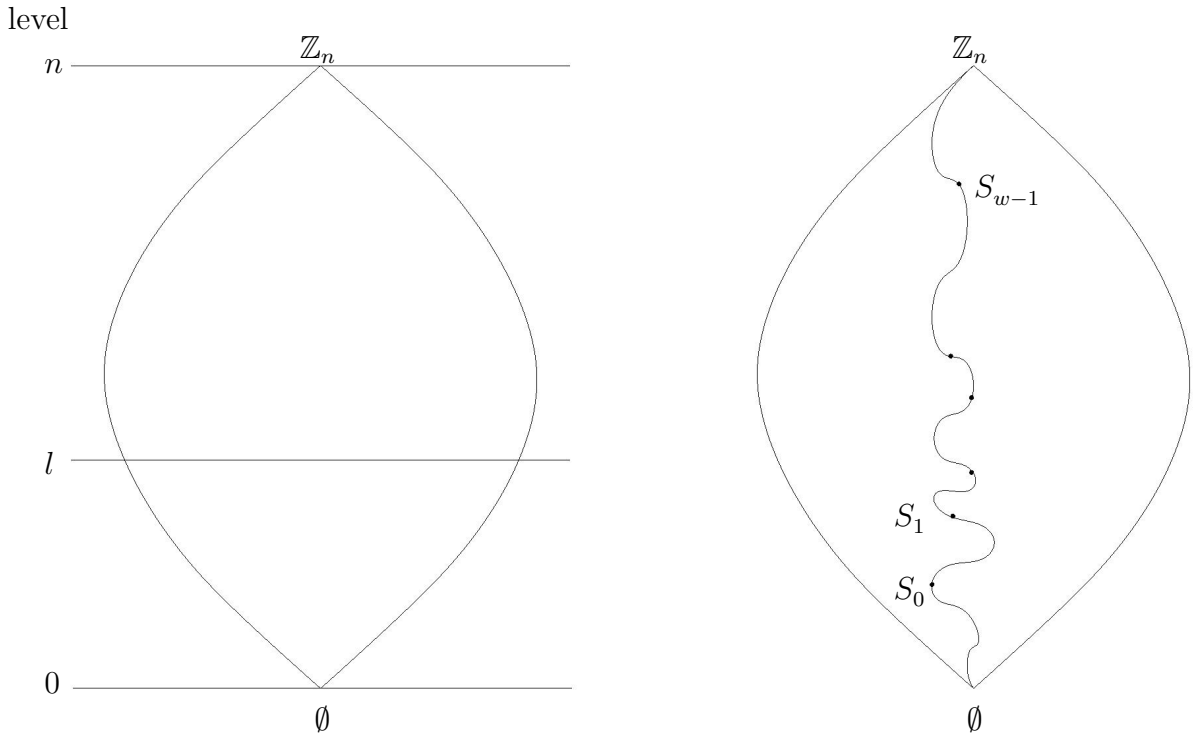
multiple selected frames. The MU-Decoder is, however, efficient only for certain types of subsets. In this thesis, we extend the range of subsets for which the MU-Decoder works. We also construct a framework for generating arbitrary subsets on the MU-Decoder.

The core problem addressed is that of generating a subset. While this is useful in FPGA partial reconfiguration, the application of subset generation is much wider, including wireless and heterogeneous networks [11], vehicle control units [31], database servers [5], image compression [29], bioinformatics [15] and much more.

1.1 Problem Definition

Let $\mathbb{Z}_n = \{0, 1, \cdots, n-1\}$. The goal is to generate a set $\mathcal{S} = \{S_i : 0 \leq i < u, S_i \subseteq \mathbb{Z}_n\}$. In this thesis, we focus on a "totally-ordered" set $\mathcal{S}$ where $S_0 \subset S_1 \subset \cdots \subset S_{u-1}$. The 2^n subsets of $\mathbb{Z}_n$ can be viewed on a Boolean Lattice (Haase diagram) [28] as shown in Figure 1.2(b) (an experiment example where $n = 4$ is in Figure 3.1 on page 14). In this lattice a totally-ordered set is a set of points on a path from $\emptyset$ to $\mathbb{Z}_n$ as shown in Figure 1.2(b) with $S_0, S_1, \cdots, S_{w-1} \in \mathcal{S}$, which is totally-ordered.

It has been shown [17] that $\mathcal{S}$ can be produced using an MU-Decoder with a delay of $O(\log n)$ and gate-cost of $O(n\sqrt{u}\log u)$. Then Jordan and Vaidyanathan [16] extend this to "isomorphic" totally-ordered sets (see Figure 1.3(a)) $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ with $\mathcal{S}_m = \{S_i^m : 0 \leq i < w \text{ and } S_i^m \subseteq \mathbb{Z}_n\}$. Here combinations of the subsets of any two $\mathcal{S}_m, \mathcal{S}_{m'}$ are in one-to-one correspondence. That is, if $\mathcal{S}_m$ contains subsets of size $a_0, a_1, \cdots, a_{w-1}$ then so does $\mathcal{S}_{m'}$. The cost of producing $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ is $O((v+w)n\log w)$. If the uw subsets were to be produced naively by a simple (uw) x n look-up table (LUT) its cost would be $O(uwn)$. So the cost of producing the subsets on the MU-decoder is substantially smaller than that of a brute force "LUT Decoder" [17] method. While the LUT-Decoder can produce arbitrary subsets expensively, the MU-Decoder can produce only a certain type of subsets (for example, isomorphic totally-ordered subsets) inexpensively. In a way, the work of this thesis is to extend the type of subsets that the MU-Decoder can produce effectively. We show that the set $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ can be produced on an MU-Decoder with $O((v+w)n\log(uv))$ (same as

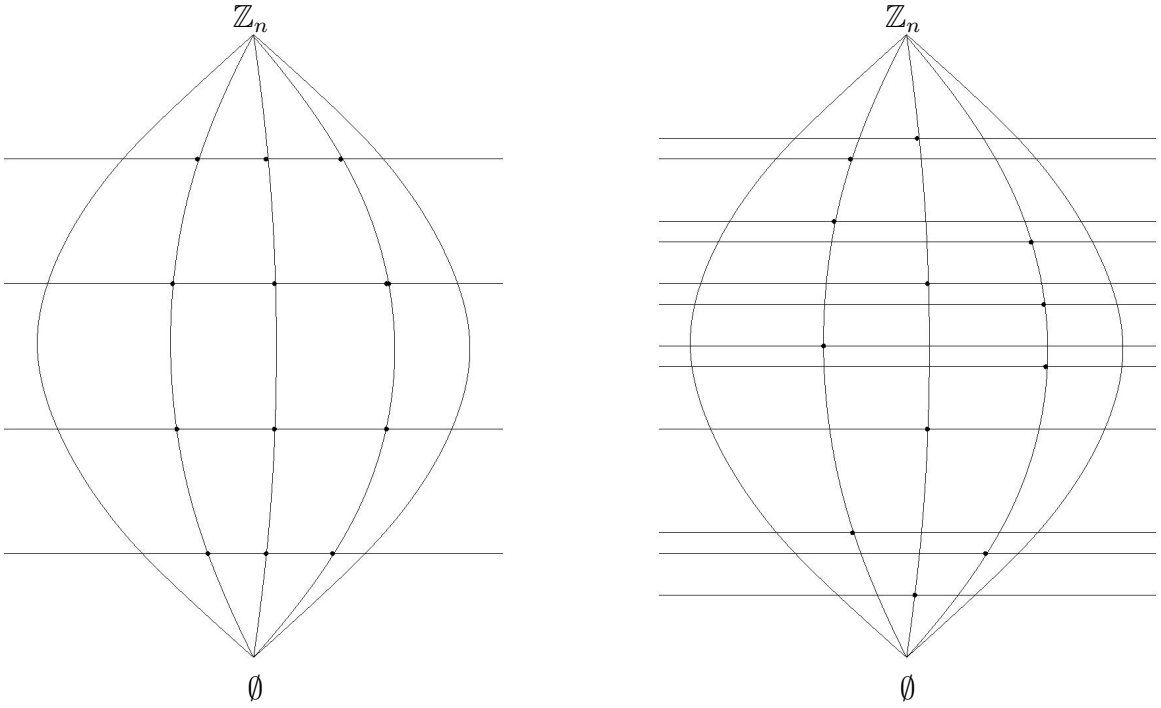


(a)Boolean Lattice with 0 to n levels
level ℓ holds $\binom{n}{\ell}$ subsets of size ℓ

(b)Boolean Lattice with Totally-Ordered
subsets $\mathcal{S} = \{S_0, S_1, \cdots, S_{w-1}\}$

Figure 1.2: Boolean Lattice

before) but without the restriction of isomorphism on the totally-ordered sets $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ (see Figure 1.3(b)). To implement the subset of the type in Figure 1.3(b), the method of Jordan and Vaidyanathan we will need a MU-Decoder of cost $O(vwn \log(uv))$. Considering that v and w could be quite large (in hundreds) the difference in cost could be significant (a factor of tens). Next, we develop properties of totally-ordered sets that allow us to increase the efficiency of the MU-Decoder. For the set $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$, each with w subsets, the cost can be reduced further to $\Theta(n(v+w))$. This is done through a hardware enhancement of the MU-Decoder that does not change its gate-cost significantly. However, it adds some additional conditions on the sets $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$. These conditions, while not as strict as isomorphism, are stricter than just total order. They put constraints on the gaps in the path representing totally-ordered sets (Figure 1.4). The shaded circles in the figure represent the original (generator) subsets of each totally-ordered set and the unshaded circles represent new subsets generated from the generator subsets in the MU-Decoder. Now the restriction



(a)Isomorphic Totally-Ordered Sets (b)Non-Isomorphic Totally-Ordered Sets

Figure 1.3: Totally-Ordered Sets

on the generator subsets is that the smallest of these subsets must have a minimum size. The subsets generated have the condition of being equally spaced in the Boolean Lattice (of equal Hamming Distance [12]) from their corresponding generator subset. The above approach allows us to pick the "best" sets of subsets. $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ such that they are strategically placed close (on the Boolean Lattice) to the subsets that we wish to generate. Together with a 1-hot Decoder, we can now generate $\binom{n}{d} z^2 \log z$ subsets in $O(d \log n)$ time using a MU-Decoder of gate cost $O(zn \log z)$ and delay $O(\log n)$. This is a substantial expansion of the MU-Decoder range for efficient operation. In fact, the method used here can also be ported to the LUT Decoder. However, the cost of the LUT decoder will still be $O(z^2 n \log z)$.

Chapter 2 presents the preliminary concepts used throughout the thesis.

In Chapter 3, we find ourselves in the midst of defining the conditions for sets being totally-ordered.

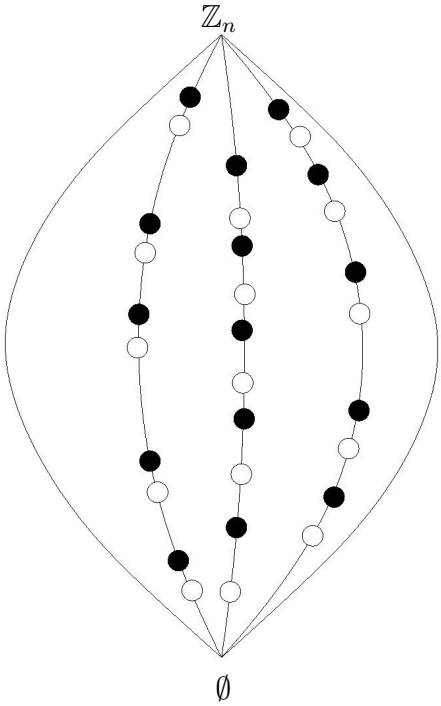


Figure 1.4: Totally Ordered Subset Restriction

In Chapter 4, we prove that isomorphism is not required to produce disjoint output sets which are individually totally-ordered.

In Chapter 5, we discuss hardware enhancements for the MU-Decoder, which increase its productivity by accommodating the concept of *translation* (introduced in this thesis) into the hardware.

In Chapter 6, we use a generic set of subsets and cover an additional d distance on the Boolean Lattice to produce an $\binom{n}{d}$ factor increase in subsets in $O(d \log n)$ time.

Finally, in Chapter 7, we summarize our findings and discuss the future scope of this research and ideas that can be further developed upon.

#	MU-Decoder configuration	Constraint	Cost	Subsets	Equivalent LUT cost	Ref.
1	MD(x, y, z, n)	none	$\Theta(2^x(x+z)$ $+n\log z(2^y$ $+z))$	$\min\{2^x,$ $2^y\lfloor\log z\rfloor\}$	n $\min\{2^x,$ $, 2^y\lfloor\log z\rfloor\}$	[17]
2	MD$(\lceil\log(z-1)\rceil,$ $\lceil\log z\rceil, z, n)$	Totally-ordered, Isomorphic	$\Theta(zn\log z)$	z^2	nz^2	[16]
3	MD$(\lceil\log(z-1)\rceil,$ $\lceil\log z\rceil, z, n)$	Non-Isomorphic	$\Theta(zn\log z)$	z^2	nz^2	[this thesis]
4	MD*$(\lceil\log(z-1)\rceil,$ $\lceil\log(z\log z)\rceil,$ $z, n)$	Translated Partitions	$O(zn\log z)$	$z^2\log z$	$nz^2\log z$	[this thesis]
5	MD*$(\lceil\log(z-1)\rceil,$ $\lceil\log(z\log z)\rceil,$ $z, n)$	Generic Subsets+	$O(zn\log z)$	$z^2\log z\binom{n}{d}$	$nz^2\log z\binom{n}{d}$	[this thesis]

* This MU-Decoder uses a hardware enhancement proposed in this thesis.
+ Subsets with Hamming distance d from total order with $O(d\log n)$ delay

Table 1.1: Results in this thesis

Chapter 2
MU-Decoder

In this chapter, we describe the MU-Decoder and some of its relevant properties. We start the chapter by defining the standard form of a subset when represented as a string of bits, also called the characteristic representation of a subset. We start the first section with a description of the MU-Decoder, along with a diagram and all the terms associated with the same as described by Jordan and Vaidyanathan [16]. We define input and output words and the selector address along with the fact that we use ordered partition in this thesis and not regular partitions to represent the selector module. We describe the casting of a source word into an ordered partition, to produce the output word with the help of an indicator set. We show how we group these words into sets of subsets, moreover even after grouping how the casting of source words to set of ordered partitions still holds, and support it with an example.

We move on to the next section to describe the properties of the MU-Decoder. Since all these properties are discussed in the earlier work, we define them and then cite each of them for proper referencing. In this section, we discuss the gate cost, time delay and subsets produced by an MU-Decoder. We move on to define totally-ordered sets along with the property of isomorphism. Before we proceed we define the characteristic representation of a set.

Definition 2.1. Let $\mathbb{Z}_v = \{0, 1, \cdots, v-1\}$ be a v-element set. Every subset $S \subseteq \mathbb{Z}_v$ can be represented in hardware as a binary string $\langle s_0, s_1, \cdots, s_{v-1} \rangle$ where $s_i = 1$ if and only if $i \in S$. This representation of a subset is called a characteristic representation of a subset. We will, in general, not distinguish a subset from its characteristic representation. Conversely, every v-bit binary string represents a subset of $\mathbb{Z}_v$. □

In this thesis, we will use subsets of different $\mathbb{Z}_v$'s and sets of subsets of these $\mathbb{Z}_v$'s. In general we will use the term "subset" to refer to $S \subseteq \mathbb{Z}_v$ and the term "set" to refer to

$\mathcal{S} = \{S_0, S_1, \cdots\}$, where $S_i \in \mathbb{Z}_v$, or $\mathcal{S} \in \mathcal{P}(\mathbb{Z}_v)$.

2.1 MU-Decoder Structure and Ordered Partitions

The MU-Decoder was proposed by Jordan and Vaidyanathan [16] [17]. It allows for efficient generation of multiple subsets of $\mathbb{Z}_n$. Figure 2.1 shows the general structure of an MU-Decoder. For any x, z, y, n, where $x, y << z << n$, an MU-Decoder, MD(x, z, y, n) has

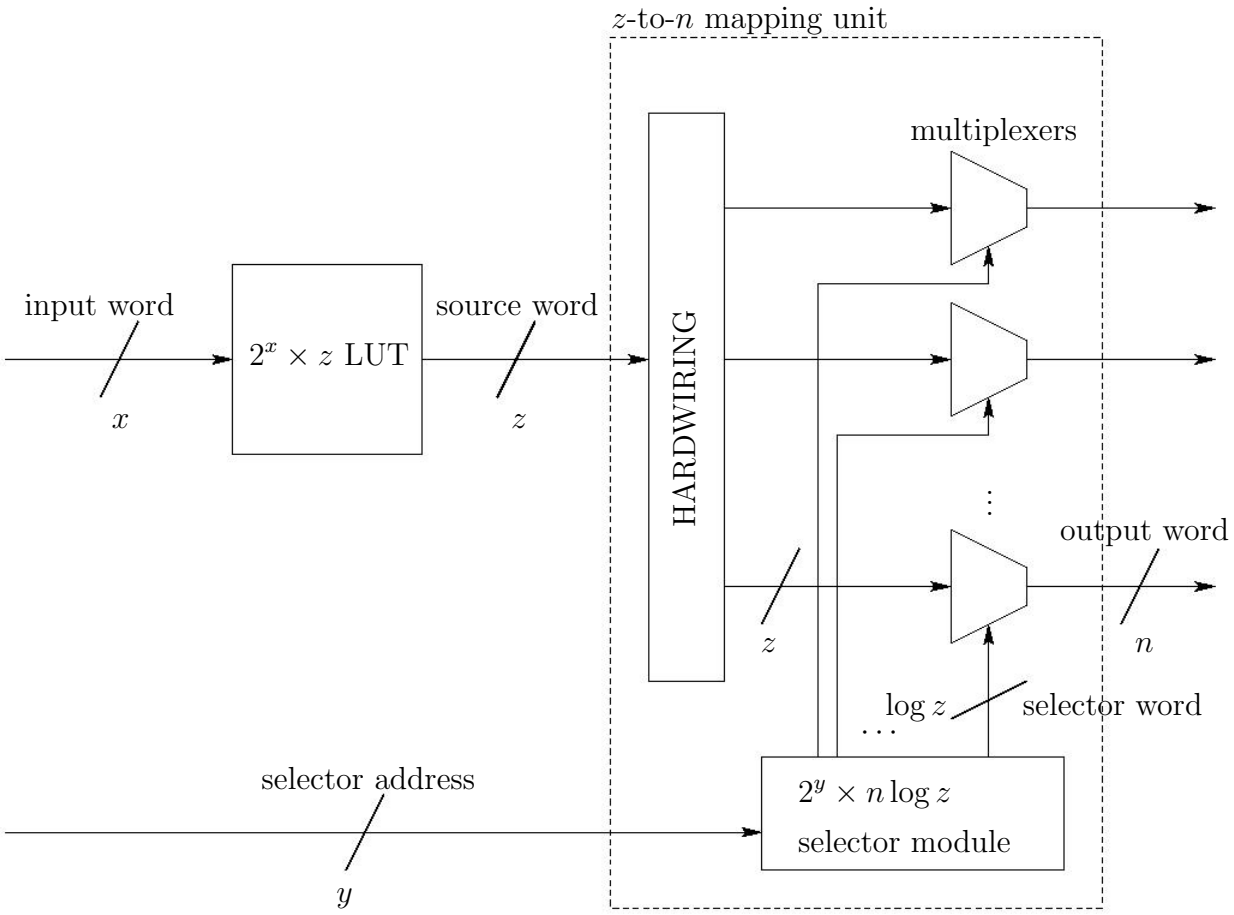


Figure 2.1: x-to-n MU-Decoder MD(x, z, y, n)

$x + y$ external input bits, z internal signals and n output bits. Functionally, the x-bit *input word* selects one of $2^x = X$ locations in the LUT and outputs the corresponding z-bit *source word*. This source word and a separate y-bit *selector address* are input to the Mapping Unit where they are converted into an n-bit output word representing a subset $S \subseteq \mathbb{Z}_n$. The

Mapping Unit multicasts the z-bit source word to the various output word bits based on an *ordered partition* selected by the y-bit selector address. Before we proceed it is important to understand the role of the source word and an ordered partition to generate an output word.

Definition 2.2. An ordered z-partition π of $\mathbb{Z}_n$ is a list $\langle B_0, B_1, \cdots, B_{z-1} \rangle$ pairwise disjoint subsets of $\mathbb{Z}_n$, that cover $\mathbb{Z}_n$. That is for $0 \leq i < j < z$, $B_i \subseteq \mathbb{Z}_n$, $B_i \cap B_j = \emptyset$ and $\bigcup_{i=0}^{z-1} B_i = \mathbb{Z}_n$ □

The difference between an ordered partition and a conventional partition is (a) the blocks B_i of an ordered partition can be empty and (b) the blocks are ordered. In this thesis, all ordered partitions have z-blocks (where z is the source word size). We will use the term ordered partition to mean an ordered z-partition. Let $L = \langle L(i) : 0 \leq i < z \rangle$ be a source word and let $\pi = \langle B_i : 0 \leq i < z \rangle$ be an ordered partition.

Definition 2.3. For any Boolean variable (condition) ν any set S defines the indicator set:

$$[\mathbb{1}(S, \nu)] = \begin{cases} S, & \text{if } \nu = 1 \\ \emptyset, & \text{if } \nu = 0 \end{cases}$$

□

We are now in a position to define how a source word combines with an ordered partition to produce subset S.

Definition 2.4. Define a cast of L into π (denoted by $L \circ \pi$) as a subset $S = \bigcup_{i=0}^{z-1} [\mathbb{1}(B_i, L(i))]$ (where $S \subseteq \mathbb{Z}_n$). It is easy to say that for all $a \in \mathbb{Z}_n$, $a \in S$ if and only if there exists i such that $a \in B_i$ and $L(i) = 1$. □

In the definition above, we remind the reader that there always exists a unique block B_i to which a belongs. The question is simply that of the block number and the corresponding source word bit.

Example 2.1. For $n = 8$, let source words $L_0 = \langle 0111 \rangle$ and $L_1 = \langle 0011 \rangle$, and let ordered partitions $\pi_0 = \langle \{0, 2, 4, 6\}, \{1, 5\}, \{3\}, \{7\} \rangle$ and $\pi_1 = \langle \{0, 1, 2, 3\}, \{4, 5\}, \{6\}, \{7\} \rangle$. Then from definition 2.4 we have $L_0 \circ \pi_0 = S_0^0 = \bigcup_{i=0}^{z-1} [\mathbb{1}(B_i^0, L_0(i))] = [\mathbb{1}(B_0^0, L_0(0))] \cup [\mathbb{1}(B_1^0, L_0(1))] \cup [\mathbb{1}(B_2^0, L_0(2))] \cup [\mathbb{1}(B_3^0, L_0(3))] = \emptyset \cup B_1^0 \cup B_2^0 \cup B_3^0 = \langle 0, 1, 0, 1, 0, 1, 0, 1 \rangle$. Similarly we get $L_1 \circ \pi_0 = S_1^0 = \langle 0, 0, 0, 1, 0, 0, 0, 1 \rangle$, $L_0 \circ \pi_1 = S_0^1 = \langle 0, 0, 0, 0, 1, 1, 1, 1 \rangle$ and $L_1 \circ \pi_1 = S_1^1 = \langle 0, 0, 0, 0, 0, 0, 1, 1 \rangle$ □

In general the LUT of Figure 2.1 contains the $X = 2^x$ source words $L_0, L_1, \cdots L_{X-1}$. The source set is $\mathcal{L} = \{L_i : 0 \leq i < X\}$. Similarly the selector module contains the representation of at least $2^y = Y$ separate ordered partitions. Let $\Pi = \{\pi_j : 0 \leq j < Y\}$ denote the set of ordered partitions. Then we will use notation $\mathcal{L} \circ \pi = \mathcal{S}$ and $\mathcal{L} \circ \Pi = \bigcup_{j=0}^{Y-1} \mathcal{L} \circ \pi_j = \mathcal{S}_j = \mathcal{S}$ to indicate set of subsets of $\mathbb{Z}_n$.

For example 2.1 we can say $\mathcal{L} = \{L_0, L_1\}$ and $\Pi = \{\pi_0, \pi_1\}$, hence the set of subsets $\mathcal{S}_0 = \{S_0^0, S_1^0\} = \mathcal{L} \circ \pi_0$, $\mathcal{S}_1 = \{S_0^1, S_1^1\} = \mathcal{L} \circ \pi_1$. Hence $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 = \mathcal{L} \circ \Pi$

Observe that if the characteristic representation of $S = \langle S(a) : 0 \leq a < n \rangle$ then $S(a) = 1$ if and only if $a \in B_i$ and $L(i) = 1$. To implement this consider a hardwiring of the blocks of π as shown in Figure 2.1, then a cast of L into π is simply multicasting L through this hardwiring. The Mapping Unit implements this multicast by configuring through the selector word each of its n z-to-1 MUXes. It has been shown in theorem 2.1 that $\mathrm{MD}(x, z, y, n)$ has a cost and delay.

2.2 Properties of MU-Decoder

In this section, we list some relevant properties of the MU-Decoder. These are all from Jordan and Vaidyanathan [16]

Theorem 2.1. An MU-Decoder $\mathrm{MD}(x, z, y, n)$ as shown in Figure 2.1 has
Gate Cost=$O(2^x(x + z) + n \log z(z + 2^y))$, Delay=$O(x + \log z + y + \log n)$
producing $\min\{2^x, 2^y \lfloor \log z \rfloor\}$ independent subsets or a many as $O(2^{x+y})$ subsets. □

Now we define a totally-ordered set of subsets.

Definition 2.5. A set $\mathcal{S} = \{S_i : 0 \leq i < u\}$ is totally ordered if and only if there exists an ordering (permutation) of the indices of $\mathbb{Z}_n$ such that for all $0 \leq i < u-1$, $S_{f(i)} \subset S_{f(i+1)}$ □

Later in Section 3.3 (see page 17) we show that we can assume without loss of generality that $f(i) = i$; hence $S_0 \subset S_1 \subset \cdots S_{n-1}$ as stated by Jordan and Vaidyanathan [16].

Definition 2.6. Let $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ be totally-ordered sets. These form an isomorphic set of totally-ordered sets if and only if these subset's cardinalities are in one-to-one correspondence. □

That is if $\mathcal{S}_0$ has elements of cardinality $a_0 < a_1 < \cdots < a_{n-1}$ then so do all $\mathcal{S}_i$'s. Figure 1.3(a) (see page 5), illustrates an isomorphic totally-ordered set. We note that each line represents a totally-ordered set, and these lines contain 4 subsets each. A subset on one totally-ordered set shares the level with a subset from each of the other totally-ordered sets. From Figure 1.2(a) we know that at level ℓ all subsets are of size ℓ. Hence the subsets are isomorphic. This theorem has been proved by Jordan and Vaidyanathan [16].

Chapter 3
Totally-Ordered Sets

In this chapter, we discuss the properties of "totally-ordered sets" as they relate to the MU-Decoder. Jordan and Vaidyanathan [16] showed that totally-ordered sets have an efficient implementation on the MU-Decoder. In this thesis, we expand the scope of this observation by extending the range of sets for which the MU-Decoder is efficient. In this chapter, we formally define totally-ordered sets and derive some properties of the source set (see definition 2.5 on page 12) needed for producing totally-ordered sets. We end the chapter with the definition of a "canonical form" of source words (section 3.3). Then we justify one assumption that the source words are represented in canonical form as just a convenience for studying totally-ordered sets.

We begin with a definition of a totally-ordered set (of subsets of $\mathbb{Z}_n$), taken from Jordan and Vaidyanathan [16].

Definition 3.1. A set $\mathcal{S} = \{S_0, S_1, \cdots, S_{u-1}\} \subseteq \mathcal{P}(\mathbb{Z}_n)$ is totally-ordered if and only if there exists an ordering of elements of $\mathcal{S}$, such that $S_0 \subset S_1 \subset \cdots \subset S_{u-1}$. □

Example 3.1. Let $n = 8, u = 5$. Then with $S_0 = \{0, 6\}$, $S_1 = \{0, 1, 6\}$, $S_2 = \{0, 1, 5, 6\}$, $S_3 = \{0, 1, 5, 6, 7\}$ and $S_4 = \{0, 1, 2, 5, 6, 7\}$, the set $\mathcal{S} = \{S_0, S_1, \cdots, S_4\} \subseteq \mathcal{P}(8)$ is totally-ordered, with $S_0 \subset S_1 \subset S_2 \subset S_3 \subset S_4$. In terms of the characteristic string of a set we have $S_0 = \langle 10000010 \rangle$, $S_1 = \langle 11000010 \rangle$, $S_2 = \langle 11000110 \rangle$, $S_3 = \langle 11000111 \rangle$ and $S_4 = \langle 11100111 \rangle$. Recall that $\overrightarrow{S}$ is a characteristic string of set S. □

3.1 Totally-Ordered Sets in the Boolean Lattice

For a n element set $\mathbb{Z}_n$, its power set $\mathcal{P}(\mathbb{Z}_n)$ (with 2^n elements) can be represented as a boolean lattice [28]. The most common expression of this lattice is as a Hasse diagram [28]. This is a graph $\mathcal{G}_n$ with 2^n nodes (one per element of $\mathcal{P}(\mathbb{Z}_n)$). Two nodes S_i, S_j with characteristic strings σ_i, σ_j have an edge between them if and only if σ_i, σ_j differ in exactly one bit, that is S_i, S_j are different by exactly one element of $\mathbb{Z}_n$. Typically nodes of $\mathcal{G}_n$ are

arranged in $n+1$ levels (numbered $0, 1, \cdots, n$) such that level ℓ contains $\binom{n}{\ell}$ nodes (subsets of $\mathbb{Z}_n$) with ℓ elements. Therefore, the empty set $\emptyset$ is the only one at level 0 and the set $\mathbb{Z}_n$ is the only one at level n. Figure 3.1 shows the representation of Boolean Lattice $\mathscr{G}_4$ with 2^4 nodes and 5 levels. We remember from Figure 1.2(a) (see page 4) that ℓ holds $\binom{n}{\ell}$ subsets

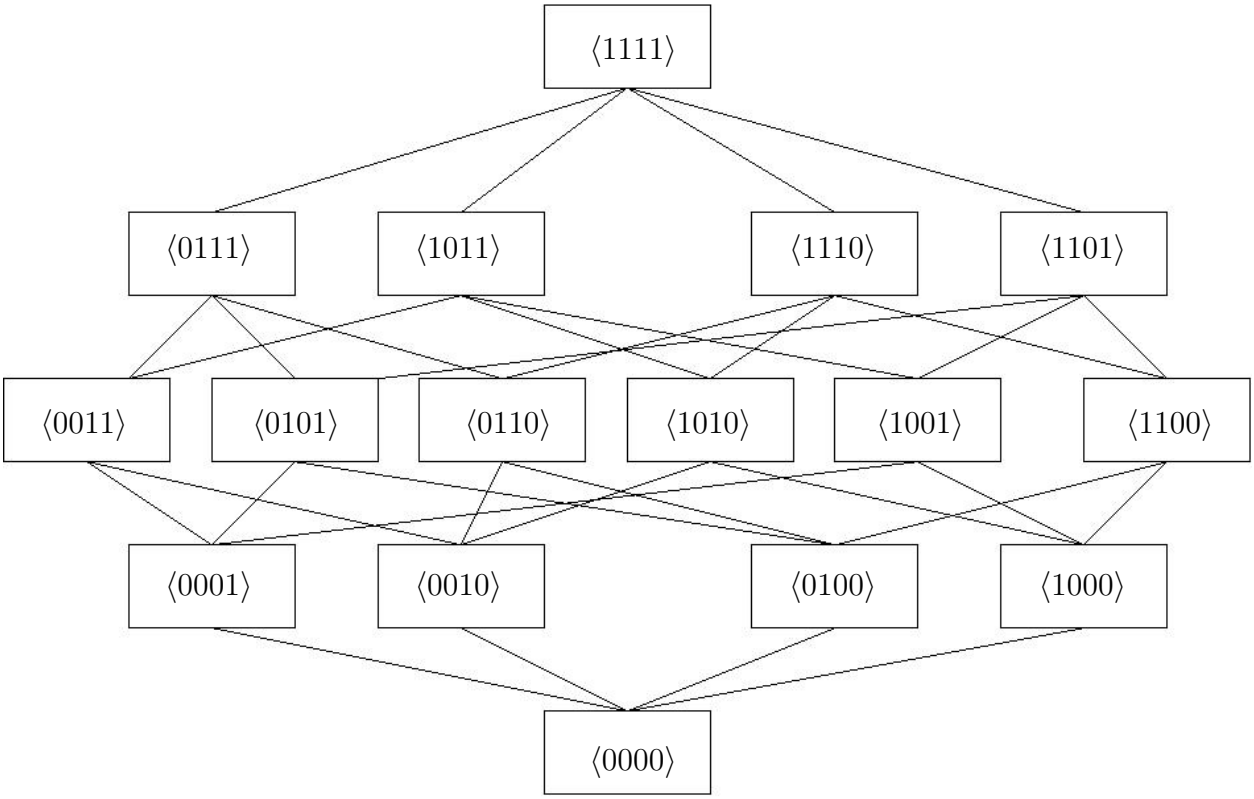


Figure 3.1: Boolean Lattice $\mathscr{G}_4$

of size ℓ. Observe that if there is an edge in $\mathscr{G}_n$ between S_i and S_j then they must be in adjacent levels, say ℓ and $\ell+1$. Without loss of generality let S_i have ℓ elements and S_j have $\ell+1$ elements. Then $S_j = S_i \cup \{a\}$, where a the only element of S_j that is not in S_i. Therefore, $S_i \subset S_j$. We view $\mathscr{G}_n$ as a directed (acyclic) graph where each undirected edge between S_i and S_j with $S_i \subset S_j$ is viewed as directed edge from S_i to S_j. In this view, all the edges of Figure 3.1 are directed towards the top. The following observation is now optional.

Lemma 3.1. If $S_0 \subset S_1 \subset \cdots \subset S_{u-1}$ then the totally-ordered set $\mathcal{S} = \{S_0, S_1, \cdots, S_{u-1}\}$ corresponds to a path in $\mathscr{G}_n$ traversing $S_0, S_1, \cdots, S_{u-1}$ in that order. □

In Definition 2.4 (see page 2.4) we described how a source set $\mathcal{L}$ and a set of ordered partition s Π produce an output set $\mathcal{S} = \mathcal{L} \circ \pi$. Now in Section 3.2 we further study the relationship between the source set $\mathcal{L}$ and the output set $\mathcal{S}$, given that one of them is a totally-ordered set. In Section 3.3 we first define a canonical form for the source words in a totally-ordered source set. Then we show that every totally-ordered source set can be assumed to be in canonical form. This representation makes many of the proofs starting from Chapter 4 more concise.

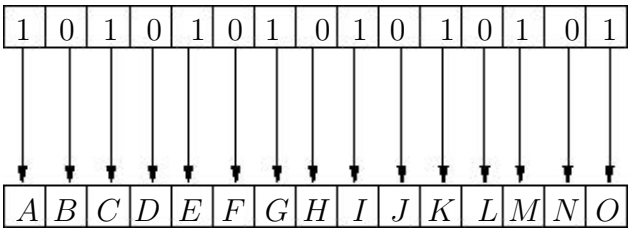
3.2 Totally-Ordered Source and Output Sets

Let $\mathcal{L} = \{L_0, L_1, \cdots, L_{X-1}\}$ be a set of z bit source words and let $\pi = \langle B_0, B_1, \cdots, B_{z-1} \rangle$ be an ordered z ordered partition of $\mathbb{Z}_n$. Let $\mathcal{S} = \mathcal{L} \circ \pi = \{S_0, S_1, \cdots, S_{u-1}\}$ be a set of u subsets of $\mathbb{Z}_n$, where $u \leq X$.

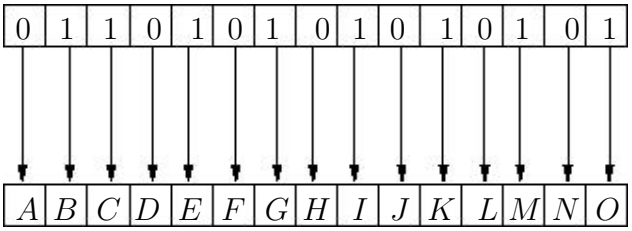
Lemma 3.2. If $\mathcal{L}$ is a totally-ordered set then $\mathcal{S} = \mathcal{L} \circ \pi$ is a totally-ordered set. □

Proof. We proceed in the contrapositive direction. Suppose $\mathcal{S}$ is not a totally-ordered set. This implies that there exist different subsets $S_i, S_j \in \mathcal{S}$ such that $S_i \not\subset S_j$ and $S_j \not\subset S_i$. This further implies that there exist elements $a, b \in \mathbb{Z}_n$ such that $a \in S_i$, $a \notin S_j$ and $b \in S_j$, $b \notin S_i$. For blocks B_q, B_r of ordered partition π, let $a \in B_q \in \pi$ and $b \in B_r \in \pi$; observe that $q \neq r$, otherwise $a, b \in B_q \in \pi$ and $a \in S_i$ if and only if $b \in S_i$. For some source words $L_i, L_j \in \mathcal{L}$, let $S_i = L_i \circ \pi$, and $S_j = L_j \circ \pi$. Now $a \in S_i$, and $b \notin S_i$ implies that $L_i(q) = 1$ and $L_i(r) = 0$ (see definition 2.4 on page 10) or $q \in L_i$ and $r \notin L_i$. Similarly $b \in S_j$, and $a \notin S_j$ implies that $L_j(r) = 1$ and $L_j(q) = 0$ (chapter 2) or $r \in L_j$ and $q \notin L_j$. Thus $L_i \not\subseteq L_j$ and $L_j \not\subseteq L_i$, implying that $\mathcal{L}$ is also not a totally-ordered set. □

Example 3.2. Let $\mathcal{S}_i = \{S_0^i, S_1^i, \cdots, S_{X-1}^i\}$ not be a totally-ordered set, and $S_0^i =$



$S_1^i =$



We can say from the above values that $A \in S_0^i, A \notin S_1^i$, similarly $B \notin S_0^i, B \in S_1^i$. Now assuming the output set is a result of the following computation $\mathcal{L} \circ \pi_i = \mathcal{S}_i$ (definition 2.4), let the ordered partition $\pi_i = \{\{B\}, \{A\}, \{C, E, G, I, K, M, O\}, \{D, F, H, J, L, N\}\}$, and hence the corresponding $\mathcal{L} = \{L_0, L_1, \cdots, L_{X-1}\}$,$L_0 = 0110$ and $L_1 = 1010$ where $L_0 \circ \pi_i = S_0^i$ and $L_1 \circ \pi_i = S_1^i$. Hence $L_0 \not\subseteq L_1$ and $L_1 \not\subseteq L_0$ implying $\mathcal{L}$ is not a totally-ordered set. □

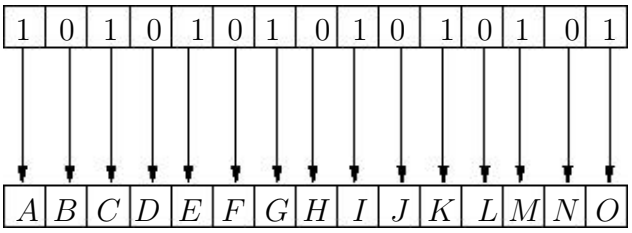
We now proceed to a similar result in the opposite direction.

Lemma 3.3. Let $\mathcal{L} \circ \pi = \mathcal{S}$, let π be an ordered partition with no empty blocks. If $\mathcal{S}$ is a totally-ordered set then $\mathcal{L}$ is a totally-ordered set. □

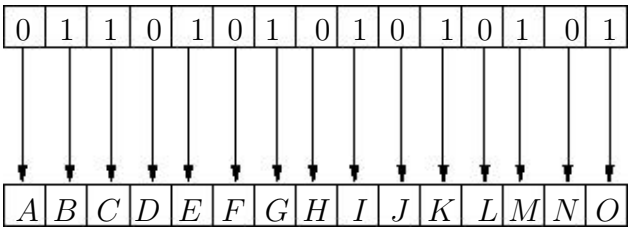
Proof. Again we proceed in the contrapositive direction. Suppose that $\mathcal{L}$ is not totally-ordered. This implies that there exist distinct source words $L_i, L_j \in \mathcal{L}$ such that $L_i \not\subset L_j$ and $L_j \not\subset L_i$. This further implies that there exist elements $q, r \in \mathbb{Z}_z$ such that $q \in L_i$, $q \notin L_j$ and $r \in L_j$, $r \notin L_i$. We can further say that $L_i(q) = 1, L_i(r) = 0$ and $L_j(r) = 1, L_j(q) = 0$. Let $a \in B_q \in \pi$ and $b \in B_r \in \pi$. (The existence of a, b is assured as $B_q, B_r \neq \emptyset$.) This further implies that $a \in S_i = L_i \circ \pi$, $a \notin S_j = L_j \circ \pi$ and $b \in S_j = L_j \circ \pi$, $b \notin S_i = L_i \circ \pi$. Hence we can conclude that $\mathcal{S}$ is not a totally-ordered set. □

Example 3.3. Let $\mathcal{L} = \{L_0, L_1, \cdots, L_{X-1}\}$,$L_0 = 0110$ and $L_1 = 1010$ where $L_0 \circ \pi_i = S_0^i$ and $L_1 \circ \pi_i = S_1^i$. Now $L_0(2) = 1, L_0(3) = 0$ and $L_1(3) = 1, L_1(2) = 0$. Let $\pi_i =$

$\{\{B\},\{A\},\{C,E,G,I,K,M,O\},\{D,F,H,J,L,N\}\}$. Assuming $\mathcal{S}_i = \{S_0^i, S_1^i, \cdots, S_{X-1}^i\}$ the corresponding $S_0^i =$



$S_1^i =$



Hence $S_0^i \not\subseteq S_1^i$ and $S_1^i \not\subseteq S_0^i$ implying $\mathcal{S}_i$ is not a totally-ordered Set. □

3.3 Canonical Form of Source Set

We first define a reverse sorted string.

Definition 3.2. A binary string $\langle a_0, a_1, \cdots, a_{m-1}\rangle$ is reverse sorted if and only if every 1 in the string precedes any 0. □

Example 3.4. For $m = 8$, $\overrightarrow{L_1} = \langle 11100000\rangle$ is reverse sorted whereas $\overrightarrow{L_2} = \langle 10110000\rangle$ and $\overrightarrow{L_3} = \langle 0000011\rangle$ are not.

Recall from Definition 2.1 (see page 8) that the characteristic string of a set $L \subseteq \mathbb{Z}_z$ depends on the characteristic order $\overrightarrow{c}$ assumed on $\mathbb{Z}_z$.

Definition 3.3. Let $\mathcal{L} \subseteq \mathcal{P}(\mathbb{Z}_z)$, and let $\overrightarrow{c} = \langle c_0, c_1, \cdots, c_{z-1}\rangle$ be a characteristic order of $\mathbb{Z}_z$. The characteristic order $\overrightarrow{c}$ is in canonical form if and only if the characteristic string $\overrightarrow{L}$ is reverse sorted for every $L \in \mathcal{L}$. □

Example 3.5. For $z = 8$ and $\mathcal{L} = \{L_0, L_1\}$, let $L_0 = \{1,3,5\}$ and $L_1 = \{1,2,3,4,5\}$. Consider the characteristic order $\overrightarrow{c_1} = \langle 0,1,2,3,4,5,6,7\rangle$, $\overrightarrow{c_2} = \langle 6,1,2,3,4,5,0,7\rangle$, $\overrightarrow{c_3} =$

$\langle 1,3,5,2,4,0,6,7\rangle$ and $\overrightarrow{c_4} = \langle 5,1,3,4,2,6,0,7\rangle$. Under $\overrightarrow{c_1}$ we have $L_0 = \langle 01010100\rangle$ and $L_1 = \langle 01111100\rangle$. Now characteristic order $\overrightarrow{c_1}$ gives the exact same characteristic representation of L_0, L_1. For characteristic order $\overrightarrow{c_2}$ we have rearranged the positions of elements of $\mathbb{Z}_n$. However the characteristic string of L_0 and L_1 are still the same as under $\overrightarrow{c_1}$. Now for $\overrightarrow{c_3}$ we see that the positions are rearranged in such a way that the resulting characteristic string produces $L_0 = \langle 11100000\rangle$ and $L_1 = \langle 11111000\rangle$. Here all 1's precede all the 0's, which is the reverse sorted order. For $\overrightarrow{c_4}$, we see that the characteristic strings of L_0 and L_1 are the same as under $\overrightarrow{c_3}$ and hence reverse sorted order. Thus $\overrightarrow{c_3}$ and $\overrightarrow{c_4}$ are in canonical form whereas $\overrightarrow{c_1}$ and $\overrightarrow{c_2}$ are not. We note $\overrightarrow{c_3} \neq \overrightarrow{c_4}$. □

We now show that there is no loss of generality in assuming a canonical order for the source set $\mathcal{L}$.

Let $\mathcal{L} = \{L_0, L_1, \cdots, L_{X-1}\}$ be totally-ordered with $L_0 \subset L_1 \subset \cdots \subset L_{X-1}$. We now construct a characteristic order $\overrightarrow{c} = \langle c_0, c_1, \cdots, c_{z-1}\rangle$ as described below. Let $|L_0| = m_0$, $|L_i - L_{i-1}| = m_i$ for $0 < i < X$ and let $n_i = m_0 + m_1 + \cdots + m_i = |L_i|$. In constructing $\overrightarrow{c}$ we enumerate the m_0 elements of L_0 first; that is $L_0 = \{c_0, c_1, \cdots, c_{m_0-1}\}$. The relative order of these m_0 elements is irrelevant. Next for $0 < i < X$, we assign the m_i elements of $L_i - L_{i-1}$ to elements of $\{c_{n_{i-1}}, c_{n_{i-1}+1}, \cdots, c_{n_i-1}\}$. We will call a characteristic order $\overrightarrow{c}$ a standard characteristic order.

Theorem 3.1. *Every standard characteristic order is canonical for totally-ordered* $\mathcal{L}$

Proof. Without loss of generality let $\mathcal{L} = \{L_0, L_1, \cdots, L_{X-1}\}$ with $L_0 \subset L_1 \subset \cdots \subset L_{X-1}$ and $|L_i| = n_i$, for $0 \leq i < X$. Let $\overrightarrow{c} = \langle c_0, c_1, \cdots, c_{z-1}\rangle$ be a standard characteristic order. From our construction we can say as follows:

$$\overrightarrow{c} = \Big\langle \underbrace{c_0, c_1, \cdots, c_{n_0-1}}_{L_0}, \underbrace{c_{n_0}, c_{n_0+1}, \cdots, c_{n_1-1}}_{L_1-L_0}, \cdots, \underbrace{c_{n_{i-1}}, c_{n_{i-1}+1}, \cdots, c_{n_i-1}}_{L_i-L_{i-1}}, \cdots \\ , \underbrace{c_{n_{X-2}}, c_{n_{X-2}+1}, \cdots, c_{n_{X-1}-1}}_{L_{X-1}-L_{X-2}} \Big\rangle \quad (3.1)$$

Hence the representation of L_i under $\overrightarrow{c}$ has the following form:

$$\overrightarrow{c} = \Bigg\langle \underbrace{\underbrace{1,1,\cdots,1}_{L_0}, \underbrace{1,1,\cdots,1}_{L_1-L_0}, \cdots, \underbrace{1,1,\cdots,1}_{L_i-L_{i-1}}}_{L_i}, \underbrace{0,0,\cdots,0}_{L_{i+1}-L_i}, \cdots, \underbrace{0,0,\cdots,0}_{L_{X-1}-L_{X-2}} \Bigg\rangle$$

which is reverse sorted (definition 3.2). □

Thus we may, without any loss of generality assume that every totally ordered source set $\mathcal{L}$ is represented in canonical order. This assumption does not change $\mathcal{L}$, it only makes our proof in subsequent chapters much easier.

Chapter 4
Generating a Given Totally Ordered Set

Let $\mathcal{S} = \mathcal{L} \circ \pi$ be any set of output subsets that can be produced from a source set $\mathcal{L}$ and a single ordered partition π. In this chapter, we will first produce multiple ordered partition s $\pi_0, \pi_1, \cdots, \pi_{y-1}$ for totally-ordered $\mathcal{S}$. This result is important as the entire result of Jordan and Vaidyanathan [16] applies only to a relatively small set $\mathcal{S}$ with at most $X = z - 1$ elements. Further, we show that any decomposition of $\mathcal{S}$ into $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{k-1}$ works, as long as $|\mathcal{S}_i| \leq X$.

In a separate result, Jordan and Vaidyanathan [16] showed that a set $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{k-1}$ of isomorphic totally-ordered sets $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{k-1}$ with $|\mathcal{S}_i| \leq X$ can be implemented as $\mathrm{MU}(x, x+1, \log k, n)$. We extend this to work for any set $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{k-1}$ of totally-ordered sets (not necessarily isomorphic).

4.1 Generating a Large Totally-Ordered Set

We begin with the case where a totally-ordered set $\mathcal{S}$ is small. Recall that $\mathcal{L}$ is the source set. Suppose that $|\mathcal{L}| \geq |\mathcal{S}|$. In Lemma 3.3 we showed that if $\mathcal{S} = \mathcal{L} \circ \pi$ then if $\mathcal{S}$ is totally-ordered, then so is $\mathcal{L}$ (assuming π has no empty block). The following lemma, in a way, works in the opposite direction.

Lemma 4.1. For any given $\mathcal{L}$ and $\mathcal{S}$ that are both totally-ordered sets, with $|\mathcal{L}| \geq |\mathcal{S}|$ there exists an ordered partition π such that $\mathcal{L} \circ \pi = \mathcal{S}$. □

Proof. Let $\mathcal{L} = \{L_0, L_1, \cdots, L_{X-1}\}$ with $L_i \subset L_{i+1}$ for $0 \leq i < X-1$ and $\mathcal{S} = \{S_0, S_1, \cdots, S_{u-1}\}$ with $S_i \subset S_i + 1$ for $0 \leq i < u - 1$. Here $u \leq X$. Recall that z is the source word length for L_i to be distinct. $X \leq z - 1$, assuming $L_i \neq \emptyset$ or $\mathbb{Z}_z$. Without loss of generality assume that $\mathcal{L}$ is in canonical form. Then it is easy to see that $L_i = \left\langle \underbrace{111...111}_{i+1} \underbrace{000...000}_{z-(i+1)} \right\rangle$. Construct ordered partition $\pi = \{B_i : 0 \leq i < z\}$ as follows: block $B_0 = S_0$ and for $0 < i < u$,

$B_i = S_i - S_{i-1}$. Elements of $\mathbb{Z}_n - S_{u-1}$ can be placed in any manner among blocks B_u to B_{z-1}. Observe that $\bigcup_{j=0}^{i} B_j = S_i$. Consider any source word $L_i = \left\langle \underbrace{111...111}_{i+1} \underbrace{000...000}_{z-(i+1)} \right\rangle$ where $0 \leq i < u$. $a \in L_i \circ \pi$ if and only if $a \in B_0 \cup B_1 \cup \cdots \cup B_i = S_i$. That is $\mathcal{S}_i = L_i \circ \pi$. Therefore $\mathcal{S} = \mathcal{L} \circ \pi$. □

The above result also appears in Jordan and Vaidyanathan [16], although expressed less formally. This result extends in a simple manner to large totally-ordered sets $\mathcal{S}$ with greater than $|\mathcal{L}|$ elements. Ordered partition $\mathcal{S}$ into non-empty sets $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{v-1}$ such that $\bigcup_{j=0}^{v-1} \mathcal{S}_j = \mathcal{S}$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$. Any such ordered partition suffices since $\mathcal{S}$ is totally-ordered, clearly each S_i is totally-ordered.

Let $|\mathcal{S}_i| \leq |\mathcal{L}|$ so $v \geq \left\lceil \frac{u}{X} \right\rceil$. For each $\mathcal{S}_i$, applying Lemma 2.4 gives us an ordered partition π_i such that $\mathcal{S}_i = \mathcal{L} \circ \pi_i$. Thus with $\Pi = \{\pi_i : 0 \leq i < v\}$, we get $\mathcal{S} = \mathcal{L} \circ \Pi$.

Lemma 4.2. For any totally-ordered output set $\mathcal{S}$ and a totally-ordered source set $\mathcal{L}$ there exists a set of v ordered partitions π such that $\mathcal{S} = \mathcal{L} \circ \Pi$ where $v \geq \left\lceil \frac{|\mathcal{S}|}{|\mathcal{L}|} \right\rceil$ □

Theorem 4.1. A MU$(x, 2^x + 1, y, n)$ can produce a totally-ordered set of at most 2^{x+y} subsets. □

Proof. Setting $|\mathcal{L}| = 2^x$ and $v = 2^y$ in Lemma 4.2, we have $|\mathcal{S}| \leq v|\mathcal{L}| = 2^{x+y}$. □

This result extends the idea in Jordan and Vaidyanathan [16] to large totally-ordered sets. The above theorem shows a method to implement a large totally-ordered set $\mathcal{S}$ or a MU-Decoder. In doing so, we partitioned $\mathcal{S}$ into 2^x blocks. Clearly, it is useful to have each block of $\mathcal{S}$ contain approximately the same number of elements, so that the number of blocks is reduced. Thus $\mathcal{S} = \bigcup_{i=0}^{v-1} \mathcal{S}_i$ where $v = \left\lceil \frac{|\mathcal{S}|}{|\mathcal{L}|} \right\rceil$ and $|\mathcal{S}_i| \leq \left\lceil \frac{|\mathcal{S}|}{v} \right\rceil \leq |\mathcal{L}|$.

However, which elements of $\mathcal{S}$ are in $\mathcal{S}_i$ is not clear. For the purpose of the results in this chapter, this question is irrelevant. However, in Chapter 5 we will show that a particular way of constructing $\mathcal{S}_i$ is advantageous.

4.2 Generating a Set of Non-isomorphic Totally Ordered Sets

Let $\mathcal{S} = \bigcup_{i=0}^{v-1} \mathcal{S}_i$ be a set of output sets with $\mathcal{S}_i \neq \emptyset$, $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$ and $\mathcal{S}_i$ is totally-ordered. Then recall Definition 2.6 that $\mathcal{S}$ is a set of isomorphic totally-ordered sets if and only if for each $\mathcal{S}_i, \mathcal{S}_j$ and any $S^i \in \mathcal{S}_i$ there exists an $S^j \in \mathcal{S}_j$ such that $|S^i| = |S^j|$. Jordan and Vaidyanathan [16] proved that if $\mathcal{S}$ is a set of isomorphic totally-ordered sets and if $|\mathcal{S}_i| \leq |\mathcal{L}|$ then an MU-Decoder can generate $\mathcal{S}$ as in Definition 2.6.

The main contribution of this definition is to leverage a common $\mathcal{L}$ for all $v = 2^y$ totally-ordered sets. We now show that the isomorphic restriction is not needed and that the range of ordered partition could exceed v.

Theorem 4.2. Let $\mathcal{S} = \bigcup_{i=0}^{v-1} \mathcal{S}_i$ be a set of output sets where $\mathcal{S}_i$ is non-empty totally-ordered and pairwise disjoint. Then for any totally-ordered source set $\mathcal{L}$, there exists a set Π of at most $v + \frac{|\mathcal{S}|}{|\mathcal{L}|}$ ordered partitions such that $\mathcal{S} = \mathcal{L} \circ \Pi$. □

Proof. By Lemma 2.4 each $\mathcal{S}_i$ can be generated by $\mathcal{L}$ and an ordered partition set π_i (so $\mathcal{S}_i = \mathcal{L} \circ \pi_i$). Here $|\pi_i| = \left\lceil \frac{|\mathcal{S}_i|}{|\mathcal{L}|} \right\rceil$. Therefore the total number of ordered partition s needed for $\mathcal{S} = \bigcup_{i=0}^{v-1} \mathcal{S}_i$ is at most

$$\begin{aligned} \sum_{i=0}^{v-1} |\pi_i| &= \sum_{i=0}^{v-1} \left\lceil \frac{|\mathcal{S}_i|}{|\mathcal{L}|} \right\rceil \\ < \sum_{i=0}^{v-1} 1 + \frac{|\mathcal{S}_i|}{|\mathcal{L}|} &= v + \frac{1}{|\mathcal{L}|} \sum_{i=0}^{v-1} |\mathcal{S}_i| = v + \frac{|\mathcal{S}|}{|\mathcal{L}|}. \end{aligned}$$

The above is used later for Generic Subset Generation. □

Chapter 5
Hardware Enhancement for Partition Generation

In Chapter 2 we discussed the structure of the MU-Decoder (Figure 2.1, Page 9). In this chapter, we particularly focus on the selector module of the MU-Decoder. The selector module accepts a selector address and selects the corresponding ordered partition to be used with the MUXes for mapping the source words. Originally each selector address produces one ordered partition. In this chapter, we propose hardware enhancement for the selector module that allows each selector address to produce multiple ordered partitions.

The original selector module produces 2^y ordered z-partitions with a hardware cost of $O(2^y n \log z)$, where n is the size of the output word representing an output subset of $\mathbb{Z}_n = \{0, 1, \cdots, n-1\}$. With the enhancements, we will be able to produce $k2^y$ ordered z-partitions with a cost of $O(2^y n \log h)$; that is with the same cost as before, we can produce a factor of k more ordered partitions.

In the next section we provide an overview of the new hardware. Section 5.1 is devoted to the idea of *"Translation"* of one ordered partition to another. This operation is essential to our hardware enhancement

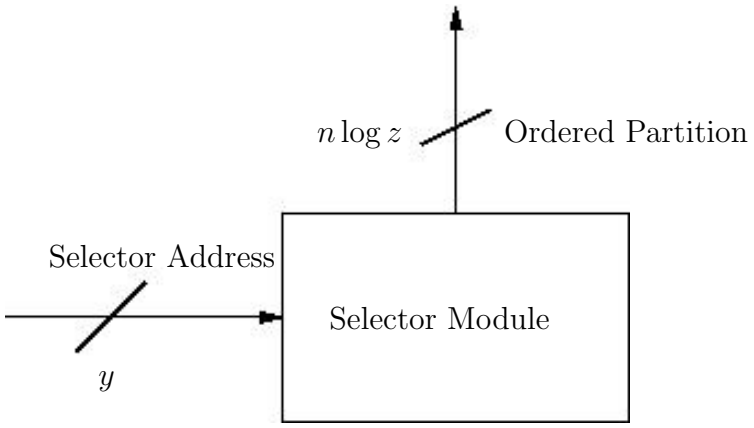


Figure 5.1: Current Selector Module

5.1 Ordered Partition Translation

In this section we describe an operation called translation on an ordered partition π_0 that produces another ordered partition π_1. Central to this operation is a *"Translation Vector"* $\overrightarrow{C}$ that guides the translation of π_0 to π_1. Subsequently we will develop the proposition of translation.

Definition 5.1. Let ordered partition $\pi_0 = \langle B_m^0 : 0 \leq m < z \rangle$ be an ordered partition on $\mathbb{Z}_n$, with $|B_m^0| = \ell_m^0$. For each $0 \leq m < z$, let $C_m \subseteq B_m^0$ with $C_{z-1} = \emptyset$. Denote $|C_m| = c_m$. Let vector $\overrightarrow{C} = \langle C_m : 0 \leq m < z \rangle$. Ordered partition $\pi_1 = \langle B_m^1 : 0 \leq m < z \rangle$ is a unit-translation of π_0 with respect to vector $\overrightarrow{C}$ (or $\pi_0 \xrightarrow{\overrightarrow{C}} \pi_1$) if and only if $B_0^1 = B_0^0 - C_0$ and for all $0 < m < z$, $B_m^1 = (B_m^0 - C_m) \cup C_{m-1}$. □

Example 5.1. Let ordered partition $\pi_0 = \langle \{3,4,11,12\}, \{2,5,10,13\}, \{1,6,9,14\}, \{0,7,8,15\} \rangle$ and ordered partition $\pi_1 = \langle \{3,4\}, \{11,12,2,5,10\}, \{13,1\}, \{6,9,14,0,7,8,15\} \rangle$. Then $\pi_0 \xrightarrow{\overrightarrow{C}} \pi_1$, that is π_1 is a unit-translation of π_0 with respect to vector $\overrightarrow{C} = \langle \{11,12\}, \{13\}, \{6,9,14\}, \{\} \rangle$. Let ordered partition $\pi_2 = \langle \{3,4,11,13\}, \{2,5,10,12\}, \{1,6,9,15\}, \{0,7,8,14\} \rangle$ then $\pi_0 \not\xrightarrow{\overrightarrow{D}} \pi_2$, because $B_0^2 \neq B_0^0 - D_0$ for any $D_0 \subseteq B_0^0$ we have $B_0^2 \not\subseteq B_0^0$. Moreover though $B_1^2 = (B_1^0 - D_1) \cup D_0$ where $D_0 = \langle 12 \rangle$, $D_1 = \langle 13 \rangle$. But $B_2^2 \neq (B_2^0 - D_2) \cup D_1$, $B_3^2 \neq (B_3^0 - D_3) \cup D_1$ and $D_3 \neq \emptyset$.

Let ordered partition $\pi_3 = \langle \{3,4\}, \{11,12,2\}, \{5,10\}, \{13,1,6,9,14,0,7,8,15\} \rangle$, then $\pi_1 \xrightarrow{\overrightarrow{E}} \pi_3$, that is π_3 is a unit-translation of π_1 with respect to vector $\overrightarrow{E} = \langle \{\}, \{5,10\}, \{13,1\}, \{\} \rangle$. We also note that $\pi_0 \not\xrightarrow{\overrightarrow{F}} \pi_3$, because $\langle 13 \rangle \in B_1^0$, so it should be in $B_1^3 = (B_1^0 - F_1) \cup F_0$ or $B_2^3 = (B_2^0 - F_2) \cup F_1$ but $\langle 13 \rangle \in B_3^3$ which is not possible through unit-translation. □

We observe that when π_0 is translated to π_1, $c_m \leq |B_m^0|$ elements of $B_m^0 \in \pi_0$ move to $B_{m+1}^1 \in \pi_1$. All the remaining elements remain in the same block, that is the rest of the elements of $B_m^0 \in \pi_0$ move to $B_m^1 \in \pi_1$ which is basically the same block number across different partitions.

Definition 5.2. For each element $a \in \mathbb{Z}_n$, let $r(a)$ be a unique rank (number) from $\mathbb{Z}_n$.

The value a and its rank $r(a)$ are not related. A set of ranks r is consistent with an ordered partition $\pi = \langle B_0, B_1, \cdots, B_{z-2}, B_{z-1} \rangle$ if and only if the following condition holds. For all $a \in B_i$ and $b \in B_j$, if $i < j$ then $r(a) < r(b)$ and if $a, b \in B_i^0$ still $r(a) \neq r(b)$. □

In Example 5.1, $\pi_0 \xrightarrow{\vec{C}} \pi_1 \xrightarrow{\vec{E}} \pi_3$ is a 2-transition with ordered partition $\pi_0 = \langle\{3,4,11,12\},\{2,5,10,13\},\{1,6,9,14\},\{0,7,8,15\}\rangle$ and the rank of the ordered partition $r(\pi_0) = \langle\{0,1,2,3\},\{4,5,6,7\},\{8,9,10,11\},\{12,13,14,15\}\rangle$.

Definition 5.3. Let $\pi_0 \xrightarrow{\vec{C^0}} \pi_1 \xrightarrow{\vec{C^1}} \pi_2 \cdots \pi_{k-1} \xrightarrow{\vec{C^{k-1}}} \pi_k$ be a series of unit-translations. This sequence of k unit-translations from π_0 to π_k will be called a k-translation if and only if there exists a rank r for each element of $\mathbb{Z}_n$ that is consistent with every ordered partition π_i $(0 \leq i \leq k)$. □

This implies that the C_m elements of B_m^0 that move into B_{m+1}^1 are the highest ranked elements of B_m^0.

Lemma 5.1. Suppose $\pi_0 \xrightarrow{\vec{C}} \pi_1$. Then for all $a \in \mathbb{Z}_n$, if $a \in B_m^0$ then $a \in B_{m'}^1$ where $m' \in \{m, m+1\}$. □

Proof. Let position $a \in B_m^0$. Now every block in ordered partition π_1 is $B_m^1 = (B_m^0 - C_m) \cup C_{m-1}$; where $C_{-1} = \emptyset$. Since $a \in B_m^0$, $a \notin B_{m-1}^0$ if it exists (the blocks are disjoint). That is $a \notin C_{m-1} \in B_{m-1}^0$. Now we consider two cases, first if $a \in C_m$, then $a \notin B_m^1$ but $a \in B_{m+1}^1 = (B_{m+1}^0 - C_{m+1}) \cup C_m$. Second if $a \notin C_m$ then $a \in B_m^1 = (B_m^0 - C_m) \cup C_{m-1}$.

□

Remark: The block number of any element of block B_m^0 increases by at most 1 as we translate from π_0 to π_1.

Theorem 5.1. Let $\mathcal{L}$ be a source set and π_0, π_1 be ordered z-partitions such that $\pi_0 \xrightarrow{\vec{C}} \pi_1$. Let output sets $\mathcal{S}_0 = \mathcal{L} \circ \pi_0$ and $\mathcal{S}_1 = \mathcal{L} \circ \pi_1$. If $\mathcal{L}$ is totally-ordered, then $\mathcal{S}_0 \cup \mathcal{S}_1$ is totally-ordered. □

Proof. Without loss of generality, we assume that the source set $\mathcal{L}$ is in canonical form (see Section 3.3 on page 17). Clearly, by Lemma 3.2 (Page 15), $\mathcal{S}_0$ and $\mathcal{S}_1$ are independently totally-ordered, as $\mathcal{L}$ is totally-ordered. Suppose that $\mathcal{S}_0 \cup \mathcal{S}_1$ is not totally-ordered. Then there exist subsets $S_i^0 \in \mathcal{S}_0$ and $S_j^1 \in \mathcal{S}_1$ such that $S_i^0 \not\subseteq S_j^1$ and $S_j^1 \not\subseteq S_i^0$. This implies that there are elements $a, b \in \mathbb{Z}_n$ such that $a \in S_i^0, b \notin S_i^0$ and $a \notin S_j^1, b \in S_j^1$. This implies that a, b are in different blocks of π_0; similarly they are in different blocks of π_1. Let $a \in B_{\hat{i}}^0$, $b \in B_{\tilde{i}}^0$ where $\hat{i} \neq \tilde{i}$ and let $a \in B_{\hat{j}}^1$ and $b \in B_{\tilde{j}}^1$ where $\hat{j} \neq \tilde{j}$. Let $S_i^0 = L_{i'} \circ \pi_0$ and $S_j^0 = L_{j'} \circ \pi_1$. Since $a \in S_i^0$, $a \in B_{\hat{i}}^0$ and $S_i^0 = L_{i'} \circ \pi_0$, we have $L_{i'}(\hat{i}) = 1$. Similarly $L_{i'}(\tilde{i}) = 0$,$L_{j'}(\hat{j}) = 0$ and $L_{j'}(\tilde{j}) = 1$. Now since source set $\mathcal{L}$ is in canonical form, from $L_{i'}(\hat{i}) = 1$ and $L_{i'}(\tilde{i}) = 0$ we can say

$$\hat{i} < \tilde{i} \tag{5.1}$$

$$\text{similarly} \qquad \hat{j} > \tilde{j} \tag{5.2}$$

Now since $a \in B_{\hat{i}}^0$ and $a \in B_{\hat{j}}^1$, a might have moved to a new block in translation.

$$\text{by Lemma 5.1} \qquad \hat{i} \leq \hat{j} \leq \hat{i} + 1 \tag{5.3}$$

$$\text{Similarly} \qquad \tilde{i} \leq \tilde{j} \leq \tilde{i} + 1 \tag{5.4}$$

Hence we can say $\tilde{i} \overset{(5.4)}{\leq} \tilde{j} \overset{(5.2)}{\leq} \hat{i} \overset{(5.3)}{\leq} \hat{j} \overset{(5.3)}{\leq} \hat{i} + 1 \overset{(5.1)}{\leq} \tilde{i}$ which is a contradiction. Hence the lemma. □

Lemma 5.2. Let $G_{j+1}^1 = B_0^1 \cup B_1^1 \cup \cdots \cup B_{j+1}^1$ and $G_{j+1}^0 = B_0^0 \cup B_1^0 \cup \cdots \cup B_{j+1}^0$. Then $G_{j+1}^1 = G_{j+1}^0 - C_{j+1}$, where $0 \leq j + 1 < z - 1$. □

Proof. As assumed $G_{j+1}^1 = B_0^1 \cup B_1^1 \cup \cdots \cup B_{j+1}^1$ and $G_{j+1}^0 = B_0^0 \cup B_1^0 \cup \cdots \cup B_{j+1}^0$. From Definition 5.1 we know that $B_0^1 = B_0^0 - C_0$ and for all $0 < j < z$, $B_j^1 = (B_j^0 - C_j) \cup C_{j-1}$

where $C_{z-1} = \emptyset$.

$$\begin{aligned}
\text{Hence} \qquad G^1_{j+1} &= B^1_0 \cup B^1_1 \cup \cdots \cup B^1_{j+1} \\
&= \{(B^0_0 - C_0)\} \cup \{(B^0_1 - C_1) \cup C_0\} \cup \cdots \cup \{(B^0_{j+1} - C_{j+1}) \cup C_j\} \\
&= B^0_0 \cup B^0_1 \cup \cdots \cup (B^0_{j+1} - C_{j+1}) \\
&= G^0_j \cup (B^0_{j+1} - C_{j+1})
\end{aligned}$$

Since all blocks are disjoint and $C_{j+1} \subseteq B^0_{j+1}$

$$\begin{aligned}
\text{Hence:} \qquad G^1_{j+1} &= G^0_{j+1} - C_{j+1} \\
\text{moreover} \qquad G^1_{j+1} &\subset G^0_{j+1}
\end{aligned}$$

Now since $C_{z-1} = \emptyset$ and union of all the blocks gives us $\mathbb{Z}_n$

$$\text{we have:} \qquad G^1_{z-1} = G^0_{z-1} = \mathbb{Z}_n$$

Hence the equation

$$G^1_{j+1} = G^0_{j+1} - C_{j+1} \tag{5.5}$$

proves the lemma. □

Let us again consider the translation $\pi_0 \xrightarrow{\vec{C}} \pi_1$, which yields the output set $\mathcal{S}_0 = \mathcal{L} \circ \pi_0$ and $\mathcal{S}_1 = \mathcal{L} \circ \pi_1$. From Theorem 5.1 we know that if source set $\mathcal{L}$ is totally-ordered then $\mathcal{S}_0 \cup \mathcal{S}_1$ is totally-ordered as well. We now derive the circumstances under which $\mathcal{S}_0$ and $\mathcal{S}_1$ are disjoint.

Lemma 5.3. Let $\pi_0 \xrightarrow{\vec{C}} \pi_1$ and for totally-ordered source set $\mathcal{L}$, let $\mathcal{S}_0 = \mathcal{L} \circ \pi_0$ and $\mathcal{S}_1 = \mathcal{L} \circ \pi_1$. Then if $0 < c_i < |B^0_i|$ for all $0 \leq i < z$ we have $\mathcal{S}_0 \cap \mathcal{S}_1 = \emptyset$, where $C_{z-1} = \emptyset$. □

Proof. We first observe that since $0 < C_i < |B^0_i|$ for $0 \leq i < z-1$, we have non empty blocks for π_0. We proceed in the contrapositive direction. Let $S \in \mathcal{S}_0 \cap \mathcal{S}_1$. Then there exists some $0 \leq u', v' < X$ such that $S = L_{u'} \circ \pi_0 = L_{v'} \circ \pi_1$. We can, without loss of generality, assume the canonical form for source set $\mathcal{L}$ and say $L_{u'} = \left\langle \underbrace{111...111}_{u} \underbrace{000...000}_{z-u} \right\rangle$ and $L_{v'} = \left\langle \underbrace{111...111}_{v} \underbrace{000...000}_{z-v} \right\rangle$. Using Definition 2.4 we can say $L_{u'} \circ \pi_0 = \bigcup_{i=0}^{z-1} [\mathbb{1}(B^0_i, L_{u'}(i))] =$

$\{B_0^0 \cup B_1^0 \cup \cdots \cup B_u^0\} = G_u^0$ (Lemma 5.2). Similarly
$L_{v'} \circ \pi_1 = \bigcup_{i=0}^{z-1} [\mathbb{1}(B_i^1, L_{v'}(i))] = \{B_0^1 \cup B_1^1 \cup \cdots \cup B_v^1\} = G_v^1$. Hence $S = G_u^0 = G_v^1$.

Hence	$S = G_u^0 = G_v^1$	(5.6)
using Equation (5.5) where	$G_v^1 = G_v^0 - C_v$	
we get	$G_u^0 = G_v^0 - C_v$	(5.7)

We now consider a few cases:

i If $u = v$ then we can state equations (5.5),(5.6) as $G_v^0 \overset{(5.6)}{=} G_v^1 \overset{(5.5)}{=} G_v^0 - C_v$. Hence $C_v = \emptyset$. This is not possible as $c_v > 0$.

ii For $u > v$ let the union of u blocks $G_u^0 = \{B_0^0 \cup B_1^0 \cup \cdots \cup B_v^0 \cup B_{v+1}^0 \cup \cdots \cup B_u^0\} = G_v^0 \cup \{B_{v+1}^0 \cup B_{v+2}^0 \cup \cdots \cup B_u^0\}$. Now $B_i^0 \neq \emptyset$. Hence $G_u^0 \supset G_v^0 \overset{(5.5)}{\supset} G_v^1 \overset{(5.6)}{=} G_u^0$, which is the necessary contradiction.

iii For $u < v$, let $u + i = v$ where $i > 0$ is an integer.

$$\begin{aligned} G_u^0 &= G_v^0 - C_v && \text{Equation (5.7)} \\ &= G_{u+i}^0 - C_{u+i} \\ &= \left[G_u^0 \cup \underbrace{\left(B_{u+1}^0 \cup B_{u+2}^0 \cup \cdots \cup B_{u+i}^0 \right)}_{H \text{ (let)}} \right] - C_{u+i} \\ &= (G_u^0 - C_{u+i}) \cup (H - C_{u+i}) \end{aligned}$$

Now since $C_{u+i} \subseteq B_{u+i}^0 \notin G_u^0$we can say

$$G_u^0 = G_u^0 \cup (H - C_{u+i})$$

Hence $H - C_{u+i} \subseteq G_u^0$

This is possible only if $H - C_{u+i} = \emptyset$ because all the blocks of an ordered partition are disjoint, meaning $H \cap G_u^0 = C_{u+i} \cap G_u^0 = \emptyset$

From $H - C_{u+i} = \emptyset$

we can say $H \subseteq C_{u+i}$

Now since all the blocks are non empty and $H = \left(B_{u+1}^0 \cup B_{u+2}^0 \cup \cdots \cup B_{u+i}^0 \right)$ we can say

$$B_{u+i}^0 \subset H \subseteq C_{u+i}$$

Therefore $|B_{u+i}^0| < c_{u+i}$,

which contradicts our assumption that $c_{u+i} < |B_{u+i}^0|$.

□

Lemma 5.4. Let $\pi_0 \xrightarrow{\vec{C}} \pi_1$ and for totally-ordered source set $\mathcal{L}$, let $\mathcal{S}_0 = \mathcal{L} \circ \pi_0$ and $\mathcal{S}_1 = \mathcal{L} \circ \pi_1$. Then $\mathcal{S}_0 \cap \mathcal{S}_1 = \emptyset$ implies for all $0 < m < z - 1$, $0 < c_m < |B_m^0|$. □

Proof. Suppose the conclusion is false. That is for all $0 < m < z-1$, $0 < c_m < |B_m^0|$ is false. This implies that there exists m such that $c_m = |B_m^0|$ or $c_m = 0 \neq |B_m^0|$.

i For $c_m = |B^0_m|$, that is $C_m = B^0_m$

$$\begin{array}{lllll} \text{Now} & G^1_m & = & G^0_m - C_m & \text{equation (5.5)} \\ & & = & G^0_m - B^0_m & \text{using } C_m = B^0_m \\ & & = & (B^0_0 \cup B^0_1 \cup \cdots \cup B^0_m) - B^0_m & \\ & & = & B^0_0 \cup B^0_1 \cup \cdots \cup B^0_{m-1} & \text{because the Blocks } B^0_i \text{ are pairwise disjoint.} \\ \text{or} & G^1_m & = & G^0_{m-1} & \text{implies that the union of blocks receiving} \\ & & & & \text{value 1 in } \pi_1 \text{ and } \pi_0 \text{ are equal.} \end{array}$$

Hence the result:

$$\implies \mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$$

i For $c_m = 0 \neq |B^0_m|$, that is $C_m = \emptyset$

$$\begin{array}{lllll} \text{Now} & G^1_m & = & G^0_m - C_m & \text{equation (5.5)} \\ \text{or} & G^1_v & = & G^0_v & \text{implies that the union of blocks receiving value-} \\ & & & & 1 \text{ in } \pi_1 \text{ and } \pi_0 \text{ are equal. Hence the result:} \end{array}$$

$$\implies \mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$$

Hence the lemma. □

Lemma 5.5. Let $\pi_0 \xrightarrow{\vec{C}} \pi_1$ and for totally-ordered source set $\mathcal{L}$, let $\mathcal{S}_0 = \mathcal{L} \circ \pi_0$ and $\mathcal{S}_1 = \mathcal{L} \circ \pi_1$. Then for all $0 < m < z-1$ we have $0 < c_m < |B^0_m|$ if and only if $\mathcal{S}_0 \cap \mathcal{S}_1 = \emptyset$. □

Proof. Evident from the results of the two Lemmas 5.3 and 5.4. □

We now consider a particular class of translation $\pi_0 \xrightarrow{\vec{C}} \pi_1$, in which for all $0 \leq m < z-1$, $|B^0_v| \neq \emptyset$ implies $0 < c_m = c \leq |B^0_m|$. That is c lower bounds the size of the smallest block of π_0. We will call such a translation as a *C-uniform unit-translation* or simply a *C-uniform translation*, where the context is clear. Before we proceed let us consider the following example.

Example 5.2. For $n = 8$, that is $\mathbb{Z}_n = \{0,1,2,3,4,5,6,7\}$, let $z = 3$ and let $\pi_0 = \langle\{0,5,7\},\{2,3\},\{1,4,6\}\rangle$. Here the block number for elements 0,5,7 is 0, that of 2,3 is 1 and that of 1,4,6 is 2. Let $\vec{C} = \langle\{5,7\},\{2,3\}\rangle$, then for $\pi_0 \xrightarrow{\vec{C}} \pi_1$ we have $\pi_1 = \langle\{0\},\{5,7\},\{2,3,1,4,6\}\rangle$. The block number for 0 is still 0, however the block number of 5,7 is now 1. This Translation causes the block number of the elements of $\mathbb{Z}_n$ to change. Notice that the number of elements entering and leaving each block is uniform ($c = 2 \leq |B_m^0|$) except the first and last blocks. Hence a C-uniform translation. □

Lemma 5.6. There exists a C-uniform translation $\pi_0 \xrightarrow{\vec{C}} \pi_1$, if and only if no block B_m^0 $(0 \leq m < z-1)$ has $0 < c' < c$ elements, that is every block has $\geq c$ elements. □

Proof. If every block has αc $(\alpha > 0)$ elements then consider any $\vec{C} = \langle C_0, C_1, \cdots, C_{z-2}\rangle$ with $c_m = c$. Since $|B_m^0| \geq c$, $c_m \leq B_m^0$ is always true. On the contrary, if some block has $0 < |B_m^0| = c' < c$, then c_m must be equal to c for a C-uniform translation. However $c_m \leq |B_m^0| < c$. Thus the C-uniform translation is not possible. □

Lemma 5.7. If $\pi_0 \xrightarrow{\vec{C}} \pi_1$ is a C-uniform translation with respect to $\vec{C}$, and if every block B_m^0 of ordered partition π_0 (for $0 \leq m < z-1$) has $\geq c$ elements, then every block B_m^1 of ordered partition π_1 has equal number of elements, that is $|B_m^1| = |B_m^0|$. □

Proof. Recall from Definition 5.1 that $B_m^1 = (B_m^0 - C_m) \cup C_{m-1}$ and $C_m \subseteq B_m^0$. Since $C_{m-1} \cap B_m^0 = \emptyset$, we have $|B_m^1| = |B_m^0| - |C_m| + |C_{m-1}| = |B_m^0| - c_m + c_{m-1}$. Now $c_m = c_{m-1} = c$, hence $|B_m^1| = |B_m^0| - c + c = |B_m^0|$. □

Putting Lemmas 5.6, 5.7 together we can say that if the number of elements in every block (except possibly the last block) of π_0 is greater than or equal to c, then we can construct C-uniform translations

$$\pi_0 \xrightarrow{\vec{C^0}} \pi_1 \xrightarrow{\vec{C^1}} \pi_2 \cdots \pi_{k-1} \xrightarrow{\overrightarrow{C^{k-1}}} \pi_k$$

This is because the block size remains the same through the translations which is $\geq c$.
We now consider such a *C-uniform k translation* in which from each block B_m^i, c elements shift out to the corresponding block $B_{m+1}^{i+1} \in \pi_{i+1}$.

Example 5.3. For $n = 16$, that is $\mathbb{Z}_n = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$, let $z = 4$ and let $\pi_0 = \left\langle \underbrace{\{0, 5, 7, 8, 9, 14\}}_{\text{block 0}}, \underbrace{\{2, 3, 13, 15\}}_{\text{block 1}}, \underbrace{\{1, 4, 6, 12\}}_{\text{block 2}} \underbrace{\{10, 11\}}_{\text{block 3}} \right\rangle$. Let $\vec{C^0} = \langle \{9, 14\}, \{13, 15\}, \{6, 12\} \rangle$, then for $\pi_0 \xrightarrow{\vec{C^0}} \pi_1$ we have $\pi_1 = \left\langle \underbrace{\{0, 5, 7, 8\}}_{\text{block 0}}, \underbrace{\{9, 14, 2, 3\}}_{\text{block 1}}, \underbrace{\{13, 15, 1, 4\}}_{\text{block 2}} \underbrace{\{6, 12, 10, 11\}}_{\text{block 3}} \right\rangle$. Let $\vec{C^1} = \langle \{7, 8\}, \{2, 3\}, \{1, 4\} \rangle$, then for $\pi_1 \xrightarrow{\vec{C^1}} \pi_2$ we have $\pi_2 = \left\langle \underbrace{\{0, 5\}}_{\text{block 0}}, \underbrace{\{7, 8, 9, 14\}}_{\text{block 1}}, \underbrace{\{2, 3, 13, 15\}}_{\text{block 2}} \underbrace{\{1, 4, 6, 12, 10, 11\}}_{\text{block 3}} \right\rangle$. The block number for 0,5 is still 0, however the block number of 9,14 is now 1. These translations cause the block number of the elements of $\mathbb{Z}_n$ to change. Notice that the number of elements entering and leaving each block is uniform ($c = 2 \leq |B_m^0|$), hence the block size remains constant (except the first and last blocks). Hence a C-uniform 2-translation. The extent of change in block numbers is tracked later in the chapter. □

We recall from Example 5.3 that as we translate from one ordered partition to another, block number of elements of $\mathbb{Z}_n$ change. We now track the extent of change to the block numbers as we progress through a C-uniform k translation $\pi_0 \xrightarrow{\vec{C^0}} \pi_1 \xrightarrow{\vec{C^1}} \pi_2 \cdots \pi_{k-1} \xrightarrow{\overrightarrow{C^{k-1}}} \pi_k$. Specifically, we consider any element $a \in B_m^0$ whose block number in π_0 is m. As we move from π_0 to $\pi_1, \pi_2, \cdots \pi_k$, the block number of a could increase. This increase is calculated in following paragraphs.

As defined in Definition 5.2 we can say that there is a rank r of the elements of $\mathbb{Z}_n$, that is constant with π_i $(0 \leq i \leq k)$, where π_i is produced from C-uniform k translation of π_0 through π_k. Considering $\pi_0 \xrightarrow{\vec{C^0}} \pi_1$, let some element $a \in B_m^0$ move to block B_{m+1}^1; recal that $a \in B_m^1$ or B_{m+1}^1. Thus the maximum increase in the block number of element a is 1. If $g(k)$ denotes the maximum increase in the block number of an element as we translate from π_0 to π_k, then $g(1) = 1$. Figure 5.2 shows the contents of a block B_{m+1}^1 with $\geq c$ elements. Clearly $a \in B_m^0$ that moved to B_{m+1}^1 is one of the lowest ranked c elements of B_{m+1}^1. Let $\left\lfloor \frac{|B_{m+1}^1| - c}{c} \right\rfloor = \left\lfloor \frac{|B_{m+1}^1|}{c} \right\rfloor - 1 = \ell_1$ (where $|B_{m+1}^1| \geq c$). After another ℓ_1 translation $a \in B_{m+1}^1$ moves to B_{m+2}^2 only if it is one of the largest c ranked elements of B_{m+1}^1. If $\ell_1 \geq 1$, then

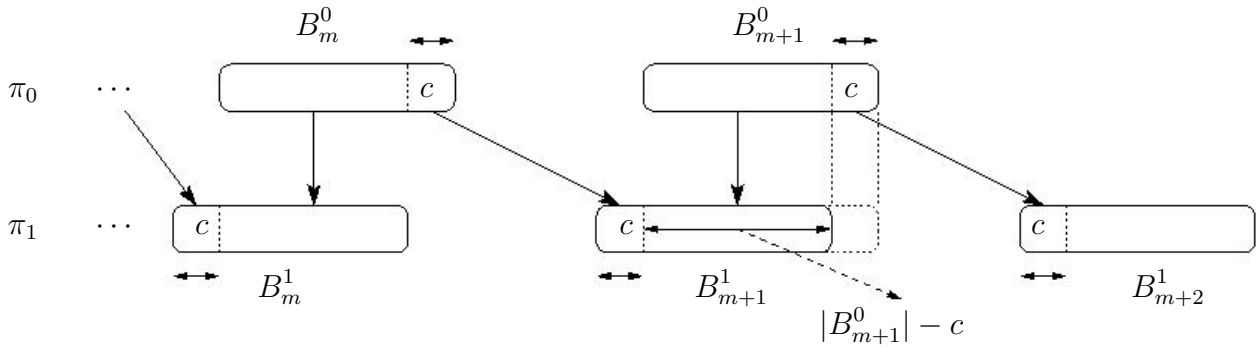


Figure 5.2: C-uniform translation

$|B^1_{m+1}| \geq 2c$ and a will not move out of B^1_{m+1} to B^2_{m+2}. It will remain in B^1_{m+1}, that is block number of a will not increase. In fact, it is easy to show that the block number of a will increase only after ℓ_1 additional translations. So for all $1 \leq \ell' \leq \ell_1$, $a \in B^{\ell'}_{m+1}$. It's possible that $a \in B^{\ell'+1}_{m+2}$. Table 5.1 extends this argument to show the number of translations needed for each block number increase. We now restrict our attention to a particular case where

Block Number	Number of Translations
m	0(initial)
$m+1$	$1 = k_1$
$m+2$	$k_1 + \left\lfloor \frac{\lvert B^{k_1}_{m+1}\rvert}{c} \right\rfloor = k_2$
$m+3$	$k_2 + \left\lfloor \frac{\lvert B^{k_2}_{m+2}\rvert}{c} \right\rfloor = k_3$
$\vdots$	$\vdots$
$m+h$	$k_{h-1} + \left\lfloor \frac{\lvert B^{k_{h-1}}_{m+h-1}\rvert}{c} \right\rfloor = k_h$

Table 5.1: Block Number of $a \in B^0_m$ over translation

$|B^0_0| \geq kc$. We call this a *non-depleting C-uniform k-translation.* It is easy to verify that each block $B^i_0 \geq 0$ $(0 \leq i < k)$, since $|B^0_m| \geq c$ for all $0 < m < z-1$, hence $B^i_m \geq c$. This is because for $m > 0$, a block receives c elements and gives up c elements. It is only B^i_0 that does not receive any element for non-depleting C-uniform k-translation.

Hence: $$|B^i_0| = |B^0_0| - ic \qquad \text{for } 0 \leq i < k \tag{5.8}$$

and $$|B^i_m| = |B^0_m| \qquad \text{for } 0 < m < z-1 \tag{5.9}$$

In Table 5.1 we see that $k_h > k_{h-1} \geq 1$. Therefore using Equation 5.9, we can say $|B_m^{k_h}| = |B_m^0|$. Thus from the table, we can say

$$\begin{aligned}
k_h &= k_{h-1} + \left\lfloor \frac{|B_{m+h-1}^{k_{h-1}}|}{c} \right\rfloor \\
&= k_{h-1} + \left\lfloor \frac{|B_{m+h-1}^{0}|}{c} \right\rfloor \\
&= k_{h-1} + \left\lfloor \frac{\ell_{m+h-1}^{0}}{c} \right\rfloor \\
&= k_{h-2} + \left\lfloor \frac{\ell_{m+h-2}^{0}}{c} \right\rfloor + \left\lfloor \frac{\ell_{m+h-1}^{0}}{c} \right\rfloor \\
&= k_{h-3} + \left\lfloor \frac{\ell_{m+h-3}^{0}}{c} \right\rfloor + \left\lfloor \frac{\ell_{m+h-2}^{0}}{c} \right\rfloor + \left\lfloor \frac{\ell_{m+h-1}^{0}}{c} \right\rfloor \\
&\vdots \\
&= k_1 + \left\lfloor \frac{\ell_{m+1}^{0}}{c} \right\rfloor + \left\lfloor \frac{\ell_{m+2}^{0}}{c} \right\rfloor + \cdots + \left\lfloor \frac{\ell_{m+h-1}^{0}}{c} \right\rfloor \\
&= 1 + \sum_{i=1}^{h-1} \left\lfloor \frac{\ell_{m+i}^{0}}{c} \right\rfloor
\end{aligned}$$

An unintended consequence of this is the following Lemma.

Lemma 5.8. If for every $0 < m < z-1$, $|B_m^0| = b$ then $k_h = 1 + (h-1) * \left\lfloor \frac{b}{c} \right\rfloor$ □

Since $g(k_h) = h$ we can have as many as $1 + (h-1) * \left\lfloor \frac{b}{c} \right\rfloor$ non-depleting C-uniform k-translation and still have each element's block number increase by at most h.

Suppose we have an ordered partition $\pi_0 = \langle B_m^0 : 0 \leq m < z \rangle$ with z blocks in which B_0^0 has at least $k_h c$ elements and each of $B_1^0, B_2^0, \cdots, B_{z-1}^0$ has at least c elements each. Then π_0 can be translated to π_h (according to a series of translations $\pi_0 \xrightarrow{\vec{C^0}} \pi_1 \xrightarrow{\vec{C^1}} \pi_2 \cdots \pi_{k-1} \xrightarrow{\vec{C^{k-1}}} \pi_k$) ensuring that the block number of any element of $\mathbb{Z}_n$ increases (corresponding to that of π_0) by at most h.

Observe that since we are using a C-uniform translation with each $\vec{C^i} = \langle C_0^i, C_1^i, \cdots, C_{z-2}^i, \rangle$ has $|C_m^i| = c$.

Example 5.4. For $n = 16$, that is $\mathbb{Z}_n = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$, let $z = 4$, $c = 2$ and let $\pi_0 = \Big\langle \underbrace{\{11,0,5,7,8,9,14\}}_{\text{block 0}}, \underbrace{\{2,3,13,15\}}_{\text{block 1}}, \underbrace{\{1,4,6,12\}}_{\text{block 2}} \underbrace{\{10\}}_{\text{block 3}} \Big\rangle$. Let $\vec{C^0} = \langle \{9,14\}, \{13,15\}, \{6,12\} \rangle$, then for $\pi_0 \xrightarrow{\vec{C^0}} \pi_1$ we have

$\pi_1 = \left\langle \underbrace{\{11,0,5,7,8\}}_{\text{block 0}}, \underbrace{\{9,14,2,3\}}_{\text{block 1}}, \underbrace{\{13,15,1,4\}}_{\text{block 2}} \underbrace{\{6,12,10\}}_{\text{block 3}} \right\rangle$. Let $\vec{C^1} = \langle\{7,8\},\{2,3\},\{1,4\}\rangle$, then for $\pi_1 \xrightarrow{\vec{C^1}} \pi_2$ we have $\pi_2 = \left\langle \underbrace{\{11,0,5\}}_{\text{block 0}}, \underbrace{\{7,8,9,14\}}_{\text{block 1}}, \underbrace{\{2,3,13,15\}}_{\text{block 2}} \underbrace{\{1,4,6,12,10\}}_{\text{block 3}} \right\rangle$. Let $\vec{C^2} = \langle\{0,5\},\{9,14\},\{13,15\}\rangle$, then for $\pi_2 \xrightarrow{\vec{C^2}} \pi_3$ we have $\pi_3 = \left\langle \underbrace{\{11\}}_{\text{block 0}}, \underbrace{\{0,5,7,8\}}_{\text{block 1}}, \underbrace{\{9,14,2,3\}}_{\text{block 2}} \underbrace{\{13,15,1,4,6,12,10\}}_{\text{block 3}} \right\rangle$.
These 3 translations cause the block number of the elements of $\mathbb{Z}_n$ to change by up to 2. Notice that the size of the first block is greater than the product of the number of translations and shift size, moreover the number of elements entering and leaving each block is uniform ($c = 2 \leq |B_m^0|$), hence the block size remains constant (except the first and last blocks). Hence a non-depleting C-uniform 3-translation ($\pi_0 \xrightarrow{\vec{C^0}} \pi_1 \xrightarrow{\vec{C^1}} \pi_2 \xrightarrow{\vec{C^2}} \pi_3$). □

The block number for elements $\{11,0,5,7,8,9,14,2,3,13,15,1,4,6,12,10\}$ changed from $0,0,0,0,0,0,0,1,1,1,1,2,2,2,2,3$ to $0,1,1,1,1,2,2,2,2,3,3,3,3,3,3,3$ respectively. Hence the change in block number for the elements are $0,1,1,1,1,2,2,1,1,2,2,1,1,1,1,0$, we call this as the increment vector for $\pi_0 \longrightarrow \pi_h$. In general the increment vector $\langle d_0, d_1, \cdots, d_{n-1}\rangle$ gives d_a, the increase in block number for element $a \in \mathbb{Z}_n$ with $0 \leq d_a \leq h$.

Recalling that an ordered partition $\pi = \langle B_0, B_1, \cdots, B_{z-1}\rangle$ is represented in the address generator module (see Figure 5.3) as a sequence $\langle e_0, e_1, \cdots, e_a, \cdots, e_{n-1}\rangle$, where for all $a \in \mathbb{Z}_n$, $e_a = m$ if and only if $a \in B_m$. Clearly $0 \leq e_a < z$. Let us call the sequence $\langle e_a : a \in \mathbb{Z}_n\rangle$ as the partition vector. Given any partition vector $\vec{e^0} = \langle e_a^0 : a \in \mathbb{Z}_n\rangle$ (corresponding to ordered partition π_0) and an increment vector $\vec{d} = \langle d_a : a \in \mathbb{Z}_n\rangle$ for $\pi_0 \longrightarrow \pi_q$, where $q = k$. One can generate the partition vector $\vec{e^q} = \langle e_a^q : a \in \mathbb{Z}_n\rangle$ as $e_a^q = e_a^0 + d_a$. Notice that $0 \leq d_a \leq h$. Figure 5.3 shows how π_q can be generated from π_0.

In the method of Jordan and Vaidyanathan [16] the address generator stored the vector for every ordered partition, it needs to use. This required a $2^y \times n \log z$ LUT with gate cost $O(2^y n \log z)$. Now we only need to store the increment vector, each of size $n \log h$ (as opposed to $n \log z$ for the entire ordered partition). If 2^{y_1} increment vectors are stored then

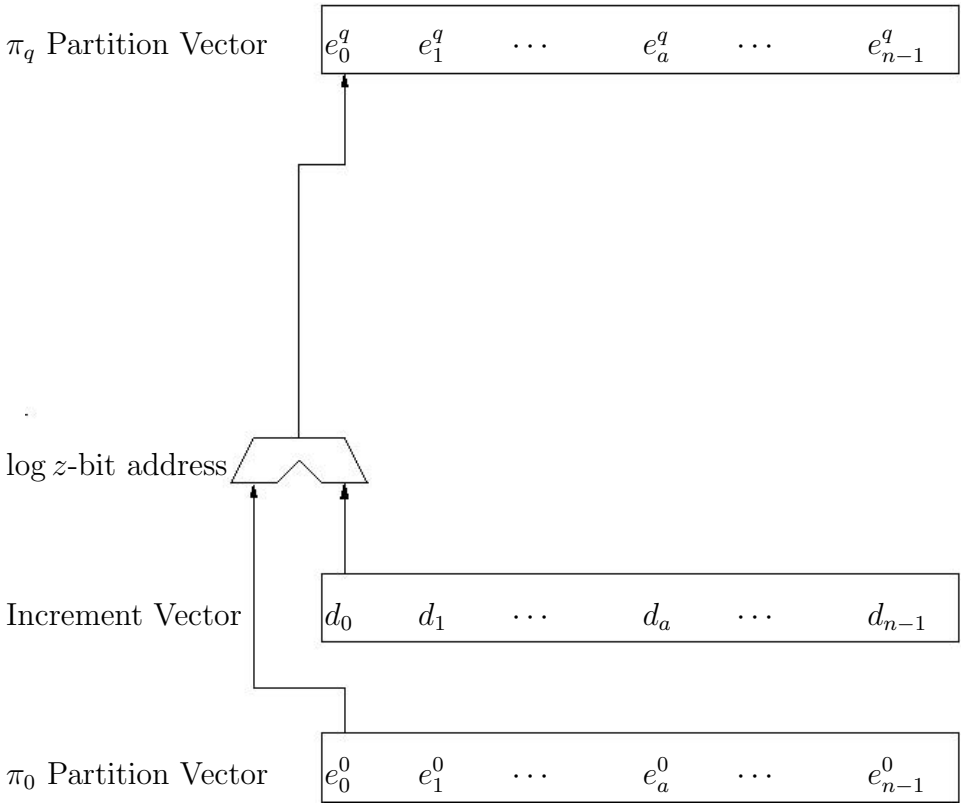


Figure 5.3: Address Generator

the cost is $O(2^{y_1} n \log h)$ for the LUT, $2n \log z$ for the input and output partition vectors. The additions even if performed by a ripple carry adder have $O(n \log z)$ cost. Thus the total cost is $O(n \log z + 2^{y_1} n \log h)$. If we set $2^{y_1} \log h = 2^y \log z$ then $y_1 = y \log \left(\frac{\log z}{\log h}\right)$. Now if h is a constant then $y_1 = y \log \log z$. Thus with the same cost of $O(2^y n \log z)$, the enhanced MU-Decoder can now produce 2^{y_1} ordered partitions and 2^{y_1+x} subsets rather than the 2^{y+x} subsets in the original solution. Further generalizing this, if we use 2^y ordered partitions each capable of handling 2^{y_1} increment vectors, then with a cost of $O(2^y n \log z + 2^{y_1} n \log h)$ one could produce 2^{y+y_1+x} subsets. If $2^y \log z + 2^{y_1} \log h = O(2^y \log z)$ then we need $2^y = O\left(2^{y_1} \frac{\log h}{\log z}\right)$. This gives the hardware structure of Figure 5.4.

In general it is more efficient to decrease y (to even 0), however, a larger y allows for independent ordered partition generation. We will explore the $x = 0$ case further in Chapter 6.

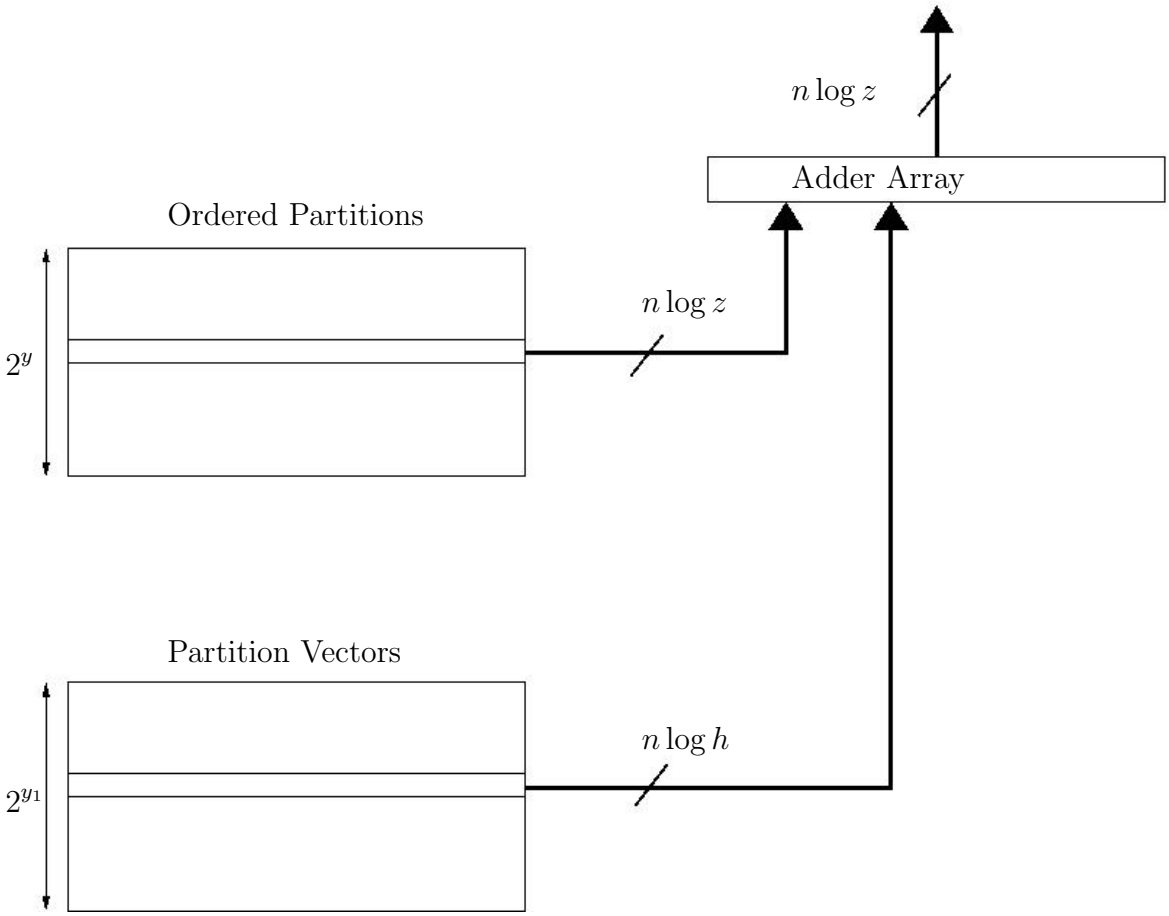


Figure 5.4: Selector Module Hardware Structure

Chapter 6
Generic Subsets

Until now we have looked at generating a given totally-ordered set. In this chapter, we generate a generic set of subsets of $\mathbb{Z}_n$, from which other subsets can be discovered.

6.1 Traversing the Boolean Lattice

Given a subset $S \subseteq \mathbb{Z}_n$, supersets of S can be generated by adding one element of $\mathbb{Z}_n - S$ at a time to S. This amounts to moving up the Boolean lattice, one level at a time. In fact, a one-hot decoder generates subsets this way.

Example 6.1. On a 1-hot decoder the subset $S_2 = \{0, 1, 2\}$ may be generated as $S_0 = \{0\}$, $S_1 = S_0 \cup \{1\}$ and $S_2 = S_1 \cup \{2\}$. □

Similarly, subsets can be formed by removing one element at a time, that is moving down the Boolean lattice. Thus the path length of moving from subsets $\hat{S} = (S_1 - S_2) \cup S_3$ (where $S_1 \cap S_2, S_1 \cap S_3$ are empty) is $|S_2| + |S_3|$ which is a good reflection of the cost of generating $\hat{S}$ from S_1.

The overall idea is to generate S_1 in $O(\log n)$ time and then spend another $O(d \log n)$ time to generate S from S_1. The set S_1 is called a primary set and S (derived from S_1) a secondary set. Now given a set $\mathcal{S} = \langle S_i : 0 \leq i < u \rangle$ of primary sets (produced by an MU-Decoder), we try to produce a set $\hat{\mathcal{S}} = \{\hat{S}_i\} \supseteq \mathcal{S}$ of secondary subsets that can be produced within a distance d of $\mathcal{S}$.

6.1.1 Single Total Order $\mathcal{S}$

Let $\mathcal{S} = \{S_i : 0 \leq i < u\}$ be totally-ordered. Using Lemma 4.1, $\mathcal{S}$ can be produced using an MU-Decoder in $O(\log n)$ time. Select $\mathcal{S}$ so that $S_i \subseteq S_{i+1}$ and $|S_{i+1}| - |S_i| \geq 2d$ This implies that the totally-ordered set $\mathcal{S}$ occupies a path in the Boolean lattice in which individual elements of $\mathcal{S}$ are at least $2d$ apart (see Figure 6.1). For each primary set S_i reaching in $O(\log n)$ time using an MU-Decoder, secondary set at distance d can be reached. Since Hamming distance between any S_i and $S_j \geq 2d$, a d distance set can be reached from

only one primary set.

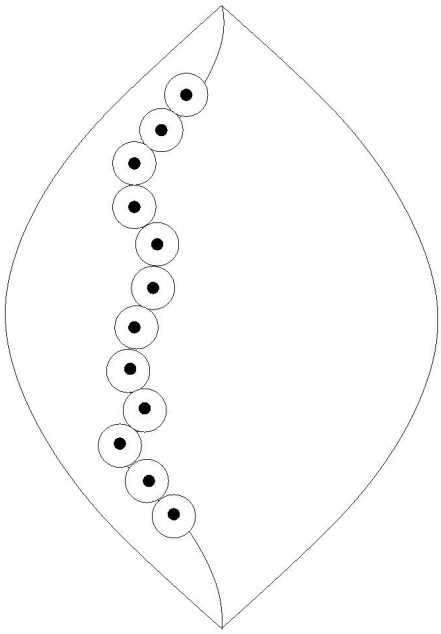


Figure 6.1: $2d$ spaced totally-ordered subsets

Lemma 6.1. The number of d-distance secondary sets reachable from any primary set $S_i \subseteq n$ is $\binom{n}{d}$. □

Proof. Out of the n possible elements in (or not in) S_i, select d to be not in (or in) $\hat{S}_i$ (a secondary set). □

Example 6.2. For set $\{0, 1, 2\}$, a 1-hot decoder generates $\{0\}$, then generates $\{1\}$ and accumulates this with $\{0\}$ to get $\{0, 1\}$ and finally generates $\{3\}$ to accumulate with prior result to get $\{0, 1, 2\}$. □

In general, the above accumulation is a move up from a point in the Boolean Lattice. A move down can be done in a similar way (using a active 0 1-hot decoder).

Given any set S, another set S' that differs from S in d elements (Hamming distance d) can be produced from S in d iteration. Here we create a set of primary subsets $\mathcal{S}$, such that each pair of elements of $\mathcal{S}$ has a Hamming distance $> 2d$, see Figure 6.1.

Lemma 6.2. If primary set of subsets is totally ordered and if it has a pairwise Hamming distance of at least $2d$, then $\binom{n}{d}z^2 \log z$ subsets can be generated in $O(d \log n)$ time by an MD($(\lceil \log(z-1) \rceil, 0, z, n)$). □

Proof. This MU-Decoder represents a single line with $2^x = z$ elements. Therefore each source word corresponds to one primary subset and there is only one partition being used ($y = 0$). The Totally-orderedfor a single line is proved by Jordan and Vaidyanathan [17] and each subset in the totally-orderedline can be used to reach $\binom{n}{d}$ subsets (Lemma 6.1) □

The following theorem is deduced in a similar way to Lemma 6.1 using Theorem 4.1 (page 21).

Theorem 6.1. Now if we have $2^y \leq \frac{n}{2d}$ $(d \geq 1)$ primary sets, where all the subsets have a pairwise Hamming distance $\geq 2d$, then $\binom{n}{d}z^2 \log z 2^y$ totally-orderedsubsets can be generated in $O(d \log n)$ time by an MD($(\lceil \log(z-1) \rceil, \lceil \log(z \log z) \rceil, z, n)$). □

Proof. $2^y \leq \frac{n}{2d}$ primary subsets can be spaced as Figure 6.1. □

To see the significance of this result, we observe that for $d = 1$ we have $nz^2 \log z$ subsets at $O(nz \log z)$ cost and $O(\log n)$ delay. However with $z = 2$, there is no significant change to the delay and cost, but the number of subsets produces increases to approximately $\frac{n^2}{2} nz^2 \log z$ (an increase by an $\frac{n^2}{2}$ factor). Given that n is much larger than z, this is a substantial increase in the number of subsets produced. More generally, for about the same costa and delay, with any constant d we will have a polynomial increase in the number of subsets produced.

Let $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \mathcal{S}_{Y-1}$, where the $\mathcal{S}_i$'s are pairwise disjoint and equal in size. Suppose $\mathcal{S}_i = \mathcal{L} \circ \pi_i$. Then $\mathcal{S}$ can be generated as shown in Definition 2.4. Now we look into the fact whether all these π_i ordered partitions can be generated from a single ordered partition π_0 as shown in Lemma 5.8 (see page 34). For us to translate k ordered partitions from ordered partition π_0, we need $|B_0^0| > kc$ and each block in π_0 to have at least c elements (except the last one). We pick any $\mathcal{S}$ (path of XY elements) where $X = |\mathcal{L}|$ the source set size and

Y is the number of partitions, including the primary and the translated ones. Let elements of $\mathcal{S}$ be subsets uniformly distributed on the path. That is, $|S_i - S_{i-1}| \geq c \geq 2d$ for each i (see Figure 6.2). Now let us divide the XY totally-ordered subsets from Figure 6.2 into Y sets of subsets as below

$$\begin{aligned}
\mathcal{S}_0 &= \{S_{Y-1}, S_{2Y-1}, \cdots, S_{iY-1}, \cdots, S_{XY-1}\} \\
\mathcal{S}_1 &= \{S_{Y-2}, S_{2Y-2}, \cdots, S_{iY-2}, \cdots, S_{XY-2}\} \\
&\vdots \\
\mathcal{S}_j &= \{S_{Y-(j+1)}, S_{2Y-(j+1)}, \cdots, S_{iY-(j+1)}, \cdots, S_{XY-(j+1)}\} \\
&\vdots \\
\mathcal{S}_Y - 1 &= \{S_0, S_Y, \cdots, S_{(i-1)Y}, \cdots, S_{(X-1)Y}\}
\end{aligned}$$

Now $\mathcal{S}_j = \{S_0^j, S_1^j, \cdots, S_i^j, \cdots, S_{X-1}^j\}$, where $S_i^j = S_{(i+1)Y-(j+1)} = L_i \circ \pi_j$ and $S_0^j = S_{Y-(j+1)}$. For $\pi_j = \langle B_0^j, B_0^j, \cdots, B_{z-1}^j \rangle$, we know $B_0^j = S_0^j$, $B_i^j = S_i^j - S_{i-1}^j$ and $B_X^j = \mathbb{Z}_n - S_{X-1}^j$. We used $X = 2^x = z - 1$ from Jordan and Vaidyanathan [16].

Lemma 6.3. Every block of ordered partition π_j is non-empty, except block B_X^j which is possibly empty. □

Proof. $B_0^0 - B_0^j = S_0^0 - S_0^j = S_{Y-1} - S_{Y-(j+1)}$. Since $Y - 1 - (Y - (j+1)) = j$, we can say $|S_{Y-1}| - |S_{Y-(j+1)}| = cj$. Clearly $S_0^0 \supseteq S_0^j$ so is $B_0^0 \supseteq B_0^j$.
In general,

$$\begin{aligned}
B_i^0 - B_i^j &= (S_i^0 - S_{i-1}^0) - (S_i^j - S_{i-1}^j) \\
&= (S_{(i+1)Y-1} - S_{(i)Y-1}) - (S_{(i+1)Y-(j+1)} - S_{(i)Y-(j+1)}) \\
\text{Now} \quad & (i+1)Y - 1 \geq (i+1)Y - (j+1) \geq iY - 1 \geq iY - (j+1) \\
\text{or} \quad & (S_{(i+1)Y-1} \supset (S_{(i+1)Y-(j+1)} \supset S_{(i)Y-1}) \supset S_{(i)Y-(j+1)}) \\
\text{Hence} \quad B_i^0 - B_i^j &= S_{(i+1)Y-1} - S_{(i+1)Y-(j+1)} \\
\text{or} \quad |B_i^0 - B_i^j| &= jc
\end{aligned}$$

□

Hence we can say $\pi_0 \rightarrow \pi_j$ is a non depleting C-uniform k-translation. We can say that if element a moves into B_i^j from B_{i-1}^{j-1}, then a never moves out of B_i^j. Hence $\mathcal{S}^0 = \mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{Y-1}$ can be generated from a single ordered partition π_0. Therefore for each totally-ordered set, the number of generator partitions is only 1 which is π_0 and the rest are generated using the increment vector. Hence $y = y_1 + y_2 = y_2$.

Lemma 6.4. $\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_{Y-1}$ are disjoint. □

Proof. We can say from the construction of Figure 6.1 that since $|S_i - S_{i-1}| \geq c$, hence each output subset is at least c distance apart from the nearest subset and hence disjoint. □

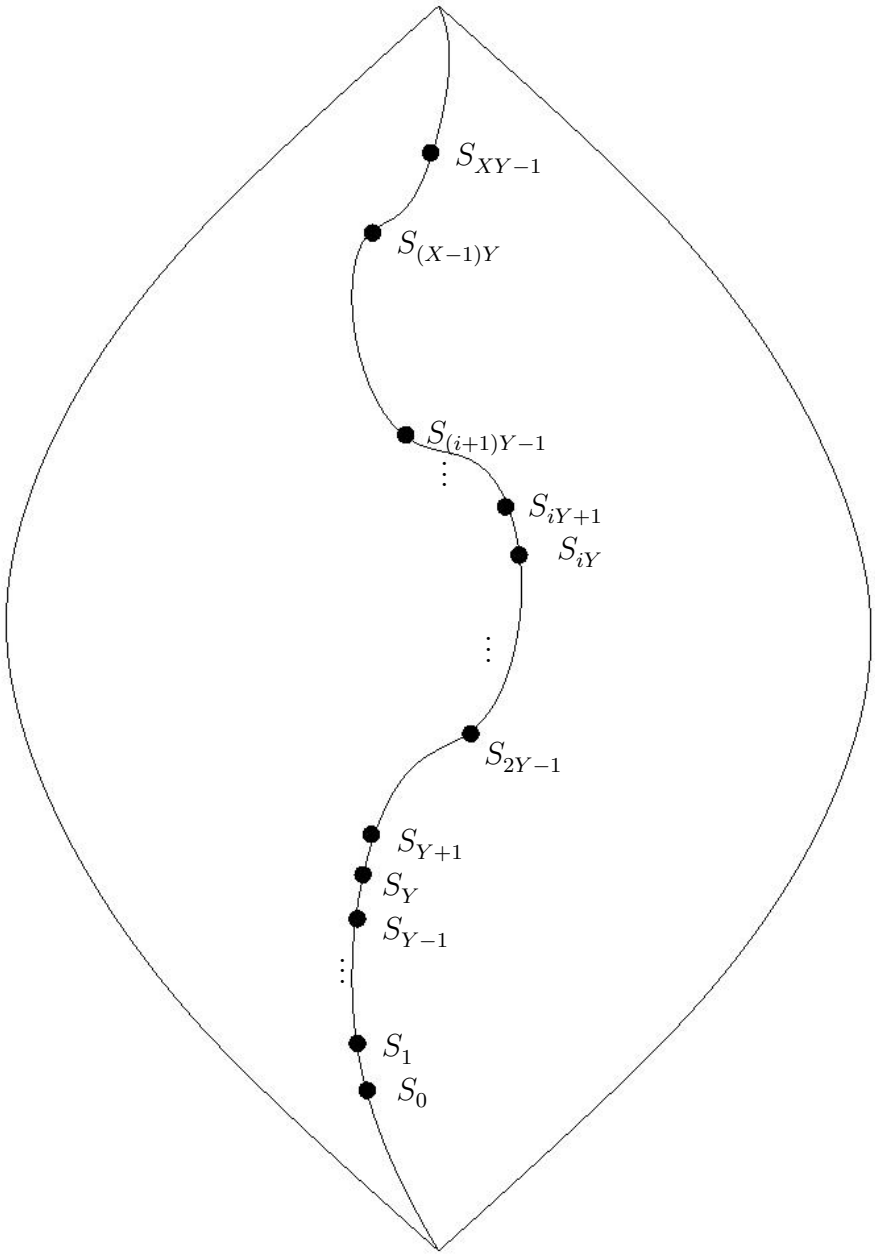


Figure 6.2: XY Totally-Ordered subsets, each circle has radius d from the primary subset

Chapter 7
Conclusion

7.1 Work Covered

In this thesis, we proved that for any source set and any output set we can produce an ordered partition which will map the given source set to the output, as long as the source set and output set are totally-ordered. We further expand on this to say that for a given set of output sets, which are individually totally-ordered, they can be produced from any given single totally-ordered source set we produce the required set of ordered partitions for this output set (for the same source set). This alleviates the limitations due to isomorphism (see Jordan and Vaidyanathan [16]). This allows us to produce different sets of individually totally-ordered output sets from a single totally-ordered source set, hence effectively reducing the cost of using a large LUT-Decoder. A set of subsets as represented in Figure 1.3(b) (see page 5), can be produced with the same cost as a regular MU-Decoder, but with the isomorphism constraint, the cost would rise significantly.

We further worked on hardware enhancements for generating additional ordered partitions from existing ordered partitions, inside the MU-Decoder itself, leading to an increase in the number of subsets produced for about the same hardware cost. We enhance the existing hardware of the MU-Decoder to produce these additional subsets. This additional hardware doesn't affect the complexity of previously assumed cost of the MU-Decoder and generates $\Theta(\log z)$ additional subsets. Producing these additional subsets has certain limitations, the generator and generated subsets together are totally-ordered. So far we generated, for π_0, the ordered partitions $\pi_1, \pi_2, \cdots, \pi_k$. The subsets $\mathcal{S}_i = \mathcal{L} \circ \pi_i$ are totally-ordered. In fact we assert that $\bigcup_i \mathcal{S}_i$ is totally-ordered. However, our method does not require this. All we impose is that $\mathcal{S}_0 \cup \mathcal{S}_i$ must be totally-ordered (as in Figure 7.1). That is, while we have proved our method for $\bigcup_i \mathcal{S}_i$ being totally-ordered, we have strong evidence that is all we require for $\mathcal{S}_0 \cup \mathcal{S}_i$ to be totally-ordered. In this thesis, we focused our work producing subsets which

can be produced cheaply, and using these subsets to approach a given set of subsets at a distance of up to $\log n$ from the produced subset. In this thesis, we focused our work producing the "best" sets of subsets. Together with the 1-hot Decoder, we generate $\binom{n}{d}2^y z^2 \log z$ subsets in $O(d \log n)$ time using a MU-Decoder MD($\lceil \log(z-1) \rceil, \lceil \log(z \log z) \rceil, z, n$) of gate cost $O(zn \log z)$ and delay $O(\log n)$ where $2^y \leq \frac{n}{2d}$ and $(d \geq 1)$. This is a substantial expansion of the MU-Decoder range for efficient operation.

7.2 Future Work

This work opens up several possible directions for future work.

Is the translation algorithm the best possible? Are similar algorithms (to generate ordered partition) possible for non-totally-ordered sets?

The cost of enhancement is driven by the cost $O(n2^y)$ of the increment vector. We assume that these vectors are independent. Could one be derived from the other?

The MU-Decoder itself is a reconfigurable device being reconfigured through its LUTs. Can the MU-Decoder be used as a LUT? For example the increment vector π as big as a subset of $\mathbb{Z}_n$.

So far the MU-Decoder used is "universal" [17] in which all z source bits are sent to all n muxes. We want to figure if there is a "good" set of subsets (as in Chapter 6) for which each MUX receives only a subset of source bits. This would drastically reduce the cost of the interconnects in the mapping unit and the address selector LUT size. (It has been shown by Kongari [21] that these interconnect is a large contributor to the MU-Decoder area.)

Can multiple MU-Decoder be used to produce different ranges of subsets of $\mathbb{Z}_n$?

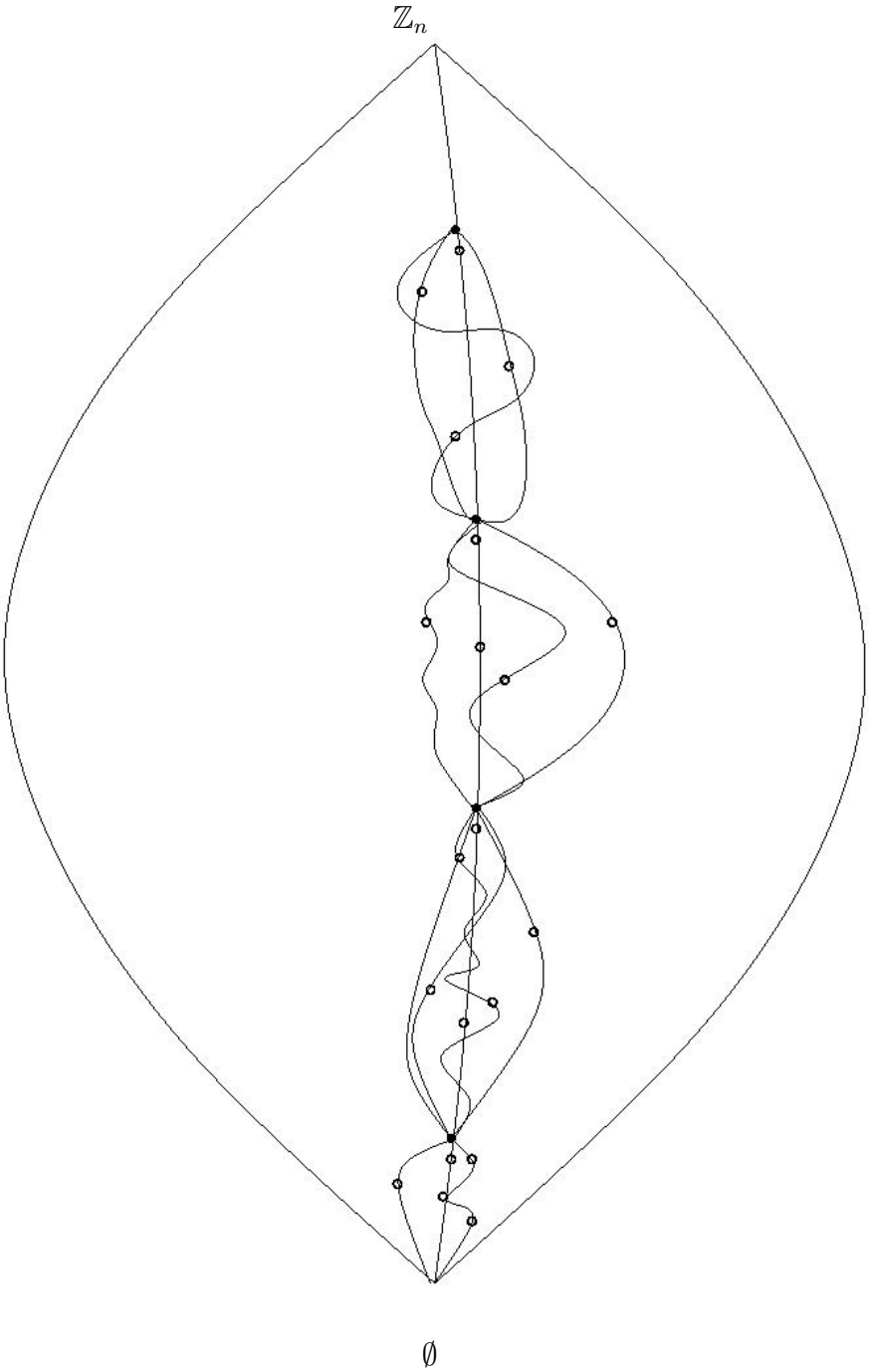


Figure 7.1: Multiple Totally-Ordered paths produced from one Totally-Ordered set of subsets

Bibliography

[1] Intel FPGAs, Solutions, 2017.

[2] Arash Ashrafi. An architecture for configuring an efficient scan path for a subset of elements. Master's thesis, Louisiana State University, 2016.

[3] P. N. Bachate and S. M. Mahamuni. FPGA based robots for industrial security and application. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 1757–1760, May 2016.

[4] S. Banerjee, E. Bozorgzadeh, and N. D. Dutt. Integrating Physical Constraints in HW-SW Partitioning for Architectures with Partial Dynamic Reconfiguration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(11):1189–1202, Nov 2006.

[5] A. Becher, F. Bauer, D. Ziener, and J. Teich. Energy-aware SQL query acceleration through FPGA-based dynamic partial reconfiguration. In *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–8, Sept 2014.

[6] SL Bishop, Suresh Rai, B Gunturk, Jerry L Trahan, and Ramachandran Vaidyanathan. Reconfigurable implementation of wavelet integer lifting transforms for image compression. In *ReConFig 2006.*, pages 1–9. IEEE, 2006.

[7] Christophe Bobda. *Introduction to reconfigurable computing: architectures, algorithms, and applications*. Springer Science & Business Media, 2007.

[8] F. Chekired, A. Mellit, S.A. Kalogirou, and C. Larbes. Intelligent maximum power point trackers for photovoltaic applications using FPGA chip: A comparative study. *Solar Energy*, 101:83 – 99, 2014.

[9] S. Corbetta, M. Morandi, M. Novati, M. D. Santambrogio, D. Sciuto, and P. Spoletini. Internal and External Bitstream Relocation for Partial Dynamic Reconfiguration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 17(11):1650–1654, Nov 2009.

[10] V. R. Deskar, M. P. V. Kumar, P. Kumar, and M. S. Gururaj. Design of net meter using FPGA. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 322–326, May 2016.

[11] Ghada Dessouky and Ahmad-Reza Sadeghi. Poster: Exploiting dynamic partial reconfiguration for improved resistance against power analysis attacks on FPGAs. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, pages 223–224, New York, NY, USA, 2016. ACM.

[12] G. Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12(2):125–131, Apr 1966.

[13] A. Gregerson, A. Farmahini-Farahani, B. Buchli, S. Naumov, M. Bachtis, K. Compton, M. Schulte, W. H. Smith, and S. Dasu. FPGA design analysis of the clustering algorithm for the CERN Large Hadron Collider. In *2009 17th IEEE Symposium on Field Programmable Custom Computing Machines*, pages 19–26, April 2009.

[14] P. Greisen, M. Runo, P. Guillet, S. Heinzle, A. Smolic, H. Kaeslin, and M. Gross. Evaluation and FPGA implementation of sparse linear solvers for video processing applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(8):1402–1407, Aug 2013.

[15] H. M. Hussain, K. Benkrid, and H. Seker. Dynamic partial reconfiguration implementation of the svm/knn multi-classifier on FPGA for bioinformatics application. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7667–7670, Aug 2015.

[16] M. C. Jordan and R. Vaidyanathan. MU-Decoders: A class of fast and efficient configurable decoders. In *2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*, pages 1–4, April 2010.

[17] Matthew Collin Jordan. A configurable decoder for pin-limited applications. Master's thesis, Louisiana State University, 2006.

[18] Cindy Kao. Benefits of partial reconfiguration. *Xcell journal*, 55:65–67, 2005.

[19] P. N. Karthik and K. Suresh. Devise and establishment of property specification language to verify the complex behaviour of FPGA ethernet IP core. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 763–768, May 2016.

[20] Hirak Kashyap and Ricardo Chaves. Compact and On-the-Fly Secure Dynamic Reconfiguration for Volatile FPGAs. *ACM Trans. Reconfigurable Technol. Syst.*, 9(2):11:1–11:22, January 2016.

[21] Raghavendra Kongari. Cost and performance modeling of the MU-Decoder. Master's thesis, Louisiana State University, 2011, 2011.

[22] B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, and D. Jao. Post-quantum cryptography on FPGA based on isogenies on elliptic curves. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(1):86–99, Jan 2017.

[23] J. Kumar, Shanmukha, Murali, J. Kumar, and R. Bhakthavatchalu. Design and implementation of hodgkin and huxley spiking neuron model on FPGA. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 1483–1487, May 2016.

[24] Yufei Ma, N. Suda, Yu Cao, J. S. Seo, and S. Vrudhula. Scalable and modularized RTL compilation of convolutional neural networks onto FPGA. In *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–8, Aug 2016.

[25] M. R. Maheshwarappa, M. D. J. Bowyer, and C. P. Bridges. Improvements in CPU and FPGA performance for small satellite SDR applications. *IEEE Transactions on Aerospace and Electronic Systems*, PP(99):1–1, 2017.

[26] P. Narapureddy, C. M. Ananda, B. P. Kumar, and E. P. J. Kumar. Design and implementation of fiber channel based high speed serial transmitter for data protocol on FPGA. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 926–931, May 2016.

[27] David Ratter. FPGAs on Mars. *Xcell J*, 50:8–11, 2004.

[28] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill Higher Education, 7th edition, 2012.

[29] J. Sarkhawas, P. Khandekar, and A. Kulkarni. Variable quality factor jpeg image compression using dynamic partial reconfiguration and microblaze. In *2015 International Conference on Computing Communication Control and Automation*, pages 620–624, Feb 2015.

[30] A. C. Shettar, K. M. Sudarshan, and S. Rehman. FPGA design and implementation of digital pwm technique for DC-DC converters. In *IEEE International Conf. on Recent Trends in Electronics, Information Communication Technology*, pages 918–920, May 2016.

[31] S. Shreejith, K. Vipin, S. A. Fahmy, and M. Lukasiewycz. An approach for redundancy in flexray networks using FPGA partial reconfiguration. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 721–724, March 2013.

[32] Nitin Srivastava, Jerry L Trahan, Ramachandran Vaidyanathan, and Suresh Rai. Adaptive image filtering using run-time reconfiguration. In *Reconfigurable Architectures Workshop, International Parallel and Distributed Processing Symposium*, 2003.

[33] J. K. Toft and A. Nannarelli. Energy efficient FPGA based hardware accelerators for financial applications. In *2014 NORCHIP*, pages 1–6, Oct 2014.

[34] Y. Wang, Q. Liu, and A. E. Fathy. Cw and pulse doppler radar processing based on FPGA for human sensing applications. *IEEE Trans. Geoscience and Remote Sensing*, 51(5):3097–3107, 2013.

[35] Xilinx Inc., Applications, 2017.

Vita

Utsav Agarwal was born on March 3 1989, in Kolkata India. He received his Elementary education from Abhinav Bharati High School, following which he received the rest of his schooling from The Assemply of God Church School. He joined Meghnad Saha Institute of Technology in the field of Electronics and Communication Engineering and graduated as a Bachelor of Technology in May 2012. After working for a couple of years India, he joined Louisina State University, USA as a graduate student in Electrical and Computer Engineering Department. He is expected to receive his Master of Science in Electrical Engineering in May 2017.