### 1 Introduction

Numbers in digital logic computations are commonly represented in one of two ways: Binary or Stochastic. In 2016, a Deterministic approach was introduced as another alternative representation [1]. The benefits and downsides of each of these approaches is summarized below.

<table>
<thead>
<tr>
<th>Compact</th>
<th>Fault Tolerant</th>
<th>Simple Generation</th>
<th>Exactly Accurate</th>
<th>Simple Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Stochastic</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

In response to these results, this research builds upon the achievements of the Deterministic approach and addresses the remaining issue of long latencies associated with a non-compact representation. It does so by introducing a hybrid representation that includes both positional and uniform aspects. After developing a representation, hardware to perform simple arithmetic operations was implemented and analyzed.

### 2 Representation

The Unary Positional system uses $k$ uniform bit streams of length $n$ to represent numbers in base-$n$ over the range $[0, n^k]$. The position of bits within a stream is insignificant, but the positions of each stream are weighted by increasing powers of $n$. An example is presented below with $n = 8$ and $k = 3$.

$$\begin{align*}
00011111 & \quad 00000011 & \quad 00011111 \\
(5 \times 8^3) = 320 & + (2 \times 8^3) = 16 & + (4 \times 8^3) = 4 = 340
\end{align*}$$

### 3 Computational Hardware

With a representation in place, basic arithmetic operations including addition and multiplication were explored. Accounting for carryover is the most prominent challenge in performing either of these computations in any positional representation. Intuitively, carryover represents full groups of $n$ bits that can be represented more compactly by using the weighting of the next higher position. Carryover in the Unary Positional representation is handled with a shift register and a collection of control signals as shown below. One of these Carry Units is used at each position of the computation.

### 4 Performance Evaluation

The representation and computational hardware described were implemented in Verilog and simulated to ensure correctness and evaluate performance. Evaluations were made in five main areas of interest including: compactness, cost of conversion/generation, fault tolerance, and time-space complexity. Comparisons made with respect to Deterministic and Binary computations were most thoroughly investigated. Results are presented below. In these results, $n$ represents the base number or bit stream length, and $k$ represents the number of positions in the Unary Positional number.

### 5 Conclusion

Referring back to the evaluation criteria in the Introduction, Unary Positional computation has produced promising results. The representation boasts simple generation, high fault tolerance, and exact accuracy. As desired, the Unary Positional representation did indeed boast exponentially shorter latencies than the Deterministic approach. Although this goal was achieved at a cost of more complex hardware, the overall time-space complexity is still highly desirable over the Deterministic approach. Moving forward, additional research is needed to identify what applications would benefit from this hybrid approach.