Journal-Style Draft: MSD Ternary Arithmetic on a Dual-Panel Black–Gray–White LCD Device

Abstract

This manuscript-style draft presents a journal-oriented description of arithmetic processing in Modified Signed Digit (MSD) representation on a dual-panel dye-doped LCD architecture that realizes three optical states: black, gray, and white.[cite:131] [cite:141] The proposed mapping associates the ternary optical states with the balanced digit set $\{-1, 0, +1\}$, enabling carry-free MSD addition, multiplication through partial-product generation and accumulation, and matrix–vector multiplication through repeated dot-product evaluation on a spatially parallel optical substrate.[cite:142] [cite:146] [cite:154] Because MSD arithmetic removes long carry propagation, the computational schedule is governed mainly by a fixed number of optical-electrical update frames and by panel response time rather than by operand word length.[cite:138] [cite:153]

Introduction

Redundant signed-digit number systems are attractive in arithmetic hardware because they replace long carry chains with local transformations, making addition and accumulation inherently parallel.[cite:150] [cite:153] In MSD arithmetic, the digit alphabet is chosen so that multiple representations exist for the same value, and this redundancy is exploited to design carry-free or low-carry adders and accumulators.[cite:142] [cite:146] Optical and optoelectronic implementations are particularly relevant because a two-dimensional spatial light modulator can apply the same local rule to many digits simultaneously, which aligns naturally with the locality of MSD transforms.[cite:154] [cite:161]

The device considered here is a dual-panel LCD system in which two dye-doped liquid-crystal panels are cascaded between an input polarizer and an output analyzer.[cite:131] [cite:145] Prior material and tolerance studies are assumed to have already established stable and repeatable electro-optical operating windows for three luminance states—black, gray, and white—so the focus of this manuscript is not on panel fabrication but on arithmetic mapping and algorithmic use of the device.[cite:135] [cite:141] [cite:145] The central objective is to show that an MSD computational model can be expressed directly on this ternary optical substrate, and that the same framework extends from scalar addition to multiplication and matrix–vector products.[cite:142] [cite:154]

Device Model and Optical Ternary Encoding

Dual-panel architecture

The optical stack consists of a diffuse light source, an input linear polarizer, the first dye-doped LCD panel, an aligned second dye-doped LCD panel, and an output polarizer.[cite:126] [cite:131] Dye-doped liquid-crystal systems rely on anisotropic absorption, so the transmitted intensity depends on the field-controlled liquid-crystal orientation, the dye alignment, and the projection through the analyzer.[cite:131] [cite:145] For the present arithmetic model, the detailed electro-optical physics are abstracted into an effective per-pixel transmittance law

$$T(x,y) = T_{\mathrm{pol}}\, T_1\big(V_1(x,y)\big)\, T_2\big(V_2(x,y)\big)$$

where T_{pol} is the polarizer transmission factor and T_1, T_2 are the panel responses under the applied drive voltages.[cite:126] [cite:131]

Ternary black–gray–white code

The two-panel transfer function is configured so that the optical output falls into three robust luminance regions associated with black, gray, and white.[cite:135] [cite:141] These three states are mapped to a balanced ternary digit set as follows: black $\to -1$, gray $\to 0$, and white $\to +1$.[cite:142] [cite:154] This coding is convenient because the optical order of luminance is consistent with the algebraic order of the balanced MSD symbols, which simplifies thresholding and readout.[cite:154] [cite:161]

A practical detector or downstream classifier therefore needs only two thresholds to recover the optical digit:

$$d(x,y) = \begin{cases} -1, & T(x,y) < \tau_1 \\ 0, & \tau_1 \le T(x,y) \le \tau_2 \\ +1, & T(x,y) > \tau_2 \end{cases}$$

with τ_1 and τ_2 chosen from the already characterized safe margins of the device.[cite:135] [cite:145]

MSD Number Representation

An integer X in balanced MSD form is represented by

$$X = \sum_{i=0}^{n-1} d_i 2^i, \qquad d_i \in \{-1, 0, +1\}$$

where redundancy permits more than one valid digit sequence for the same numeric value.[cite:150] [cite:153] The value of this redundancy is that arithmetic transforms can be designed so that each output digit depends only on a small neighborhood of input digits instead of on a carry chain extending across the full word.[cite:142] [cite:146] On the optical device, a word is placed along one spatial dimension of the display, so each pixel or subregion stores one digit in black, gray, or white form.[cite:154] [cite:161]

MSD Addition on the Dual-Panel Device

Principle of carry-free local transforms

Parallel three-step MSD addition has been studied precisely because it replaces serial carry propagation with a fixed sequence of local recoding operations.[cite:138] [cite:142] In journal terms, the key result is that the addition of two MSD operands can be decomposed into a bounded sequence of transforms—often described in the literature by intermediate recoding stages such as C, P, and R—that operate digitwise or near-digitwise.[cite:142] [cite:146] This property makes the algorithm compatible with a 2D optoelectronic array, since the same transformation is applied simultaneously to all digit positions.[cite:154] [cite:161]

Optical scheduling

Let $A = (a_{n-1}, \dots, a_0)$ and $B = (b_{n-1}, \dots, b_0)$ be two MSD operands encoded in black–gray–white states.[cite:142] The addition is executed as a short sequence of display frames:

1. **Frame 1: initial recoding.** The controller maps the local pairs (a_i, b_i) to an intermediate ternary state through a precomputed voltage lookup table on the two panels.[cite:142] [cite:146]
2. **Frame 2: correction stage.** The intermediate pattern is transformed according to the second MSD rule set, again in parallel over all digit sites.[cite:138] [cite:142]
3. **Frame 3: normalization.** The result is returned to canonical balanced-digit form so that the output is once again represented only by black, gray, and white.[cite:142] [cite:146]

Because all digits are processed in parallel, the latency is dominated by the fixed frame count and panel dynamics rather than by operand length.[cite:138] [cite:153] This is the primary computational argument for using MSD on the device.[cite:142] [cite:154]

Pseudocode for addition

```
Algorithm 1: Three-frame MSD addition on a dual-panel ternary LCD
Input: A = (a_{n-1},...,a_0), B = (b_{n-1},...,b_0), a_i,b_i in {-1,0,+1}
Output: S = A + B in canonical MSD form

1: Encode A and B as black/gray/white optical words
2: for all digit positions i in parallel do
3:     u_i <- LUT1(a_i, b_i)
4: end for
5: Display u on panel schedule F1
6: for all digit positions i in parallel do
7:     v_i <- LUT2(u_i, neighborhood(u,i))
8: end for
9: Display v on panel schedule F2
10: for all digit positions i in parallel do
11:     s_i <- LUT3(v_i, neighborhood(v,i))
12: end for
13: Display s on panel schedule F3
14: return S = (s_{n-1},...,s_0)
```

The lookup tables are determined by the chosen MSD recoding scheme, while the three displayed frame schedules are the physical implementation of those logic transforms on the dual-panel device.[cite:138] [cite:142] [cite:146]

MSD Multiplication

Partial-product generation

Multiplication in MSD arithmetic retains the classical partial-product structure, but both the elementary digit products and the accumulation are carried out in the balanced signed-digit domain.[cite:149] [cite:154] For digits $a_i, b_j \in \{-1, 0, +1\}$, the elementary product belongs to the same set:

$$p_{ij} = a_i b_j \in \{-1, 0, +1\}$$

and therefore maps directly back to black, gray, and white optical states.[cite:149] [cite:150] This is especially convenient for the proposed device because the output alphabet of a digit multiplier matches the native optical alphabet of the display.[cite:154]

Spatial interpretation on the display

A practical implementation assigns one operand to a row-wise or region-wise encoding and the second operand to another region or time-multiplexed pattern.[cite:154] [cite:155] For a given shift index j, the device can generate all products $a_i b_j$ in parallel across the full operand width, then place those products at weight 2^{i+j} either through spatial shifting on the panel or through logical offset management in the controller.[cite:154] In either case, the architecture turns multiplication into a repeated cycle of **(i)** optical product generation and **(ii)** MSD accumulation of aligned partial products.[cite:153] [cite:154]

Pseudocode for multiplication

```
Algorithm 2: MSD multiplication on a dual-panel ternary LCD
Input: A = (a_{n-1},...,a_0), B = (b_{m-1},...,b_0)
Output: P = A * B in MSD form

1: Initialize accumulator P <- 0 in MSD form
2: for j = 0 to m-1 do
3:     for all i in parallel do
4:         q_i <- a_i * b_j   // q_i in {-1,0,+1}
5:     end for
6:     Shift q by j positions to form partial product Q^(j)
7:     P <- MSD_ADD(P, Q^(j))
8: end for
9: return P
```

The call `MSD_ADD` refers to the three-frame addition primitive of Algorithm 1, so multiplication is obtained by composing a native ternary digit-product stage with a carry-free MSD accumulation stage.[cite:149] [cite:154]

Matrix–Vector Multiplication in MSD Form

Mathematical structure

For a matrix $M \in \mathbb{Z}^{r \times c}$ and a vector $x \in \mathbb{Z}^{c}$, both represented in MSD form, the matrix–vector product is

$$y = Mx, \qquad y_i = \sum_{j=1}^{c} M_{ij} x_j.$$

Each output component is therefore a dot product, which can be decomposed into repeated signed-digit multiplications and MSD additions.[cite:154] [cite:157] This decomposition is well aligned with optical matrix processors, where many pairwise products can be formed simultaneously over a two-dimensional array.[cite:154] [cite:161]

Principle figure for matrix–vector multiplication

The figure below summarizes the intended dataflow.

```
Figure 1. Principle of MSD matrix-vector multiplication on a dual-panel ternary LCD

          Input polarizer
                |
        +-------------------+
        |   LCD Panel 1     |  <- matrix coefficients M_ij encoded as B/G/W cells
        +-------------------+
                |
        +-------------------+
        |   LCD Panel 2     |  <- replicated vector entries x_j and control frames
        +-------------------+
                |
          Output polarizer
                |
      Optical ternary products M_ij * x_j
                |
   +----------------------------------------+
   | Row-wise MSD accumulation engine       |
   | (three-frame carry-free addition)      |
   +----------------------------------------+
                |
         Output vector y_i in MSD form
```

This figure captures the central abstraction: panel 1 holds the spatially distributed matrix data, panel 2 injects the vector components and recoding schedule, and the row-wise outputs are interpreted as ternary partial products that are then accumulated through the same fixed MSD-addition primitive.[cite:154] [cite:161]

Optical mapping strategy

A journal-friendly way to describe the mapping is the following. The entries M_{ij} are arranged on the display so that each matrix row occupies a spatial row band, while each vector component x_j is replicated or broadcast over the corresponding column band.[cite:154] At each multiplication step, the overlap of the encoded coefficient and the encoded vector entry yields the ternary elementary product $M_{ij}x_j$, again represented optically as black, gray, or white.[cite:154][cite:155] The set of products for a fixed j is therefore generated over all rows simultaneously, after which a row-wise MSD accumulation produces the updated partial sums for all y_i in parallel.[cite:154][cite:161]

Pseudocode for matrix–vector multiplication

```
Algorithm 3: MSD matrix-vector multiplication on a dual-panel ternary LCD
Input: M in MSD form, x in MSD form
Output: y = Mx in MSD form

1: Initialize y_i <- 0 for all rows i
2: for j = 1 to c do
3:     Broadcast x_j across the display columns associated with M_:j
4:     for all matrix rows i in parallel do
5:         g_i <- MSD_MUL(M_ij, x_j) over all digit positions
6:     end for
7:     for all rows i in parallel do
8:         y_i <- MSD_ADD(y_i, g_i)
9:     end for
10: end for
11: return y
```

Here `MSD_MUL` denotes the partial-product generation and local recoding process of Algorithm 2, while `MSD_ADD` denotes the three-frame carry-free addition primitive.[cite:149][cite:154][cite:161]

Complexity and Architectural Interpretation

The main architectural advantage of the approach is that the number of optical-electrical phases is tied to the arithmetic transform schedule rather than to the carry length of a binary representation.[cite:138][cite:153] Addition uses a fixed number of recoding frames, multiplication requires repeated partial-product steps plus the same fixed-form MSD addition, and matrix–vector multiplication computes all row updates in parallel for each broadcast vector element.[cite:142][cite:154][cite:161] Consequently, the method trades temporal carry propagation for spatial resource usage on the panel, which is exactly the exchange that makes sense on a high-resolution two-dimensional modulator.[cite:154][cite:161]

The ternary black–gray–white coding also has an implementation benefit: the same optical alphabet is preserved from stored operands to intermediate results and final outputs.[cite:141][cite:154] That avoids costly symbol-conversion stages and allows one arithmetic primitive to be reused hierarchically, from scalar addition to matrix–vector products.[cite:142][cite:154]

Discussion

From a journal perspective, the most defensible claim is not that the LCD material system performs arithmetic by itself, but that the dual-panel device provides a physically meaningful ternary substrate on which MSD arithmetic can be scheduled efficiently.[cite:131] [cite:145] [cite:154] The arithmetic benefit arises from MSD recoding theory, while the hardware benefit arises from the spatial parallelism and three-level optical symbol set of the black–gray–white device.[cite:142] [cite:153] This division is useful in writing because it separates previously completed electro-optical characterization from the present algorithmic contribution. [cite:135] [cite:145]

A second important discussion point is extensibility. Once the three-level coding and three-frame addition primitive are in place, multiplication and matrix–vector multiplication follow compositionally: multiplication is partial-product generation plus MSD accumulation, and matrix–vector multiplication is repeated row-parallel accumulation of those products. [cite:149] [cite:154] [cite:161] This makes the framework suitable for broader linear-algebra kernels, especially where the throughput advantage of simultaneous row processing is more important than single-operation latency.[cite:154] [cite:157]

Conclusion

A dual-panel dye-doped LCD capable of stable black, gray, and white operation can be interpreted as a native ternary computational plane for balanced MSD arithmetic.[cite:131] [cite:141] On this substrate, MSD addition is implemented as a fixed sequence of local recoding frames, MSD multiplication is expressed as ternary partial-product generation followed by carry-free accumulation, and matrix–vector multiplication is obtained by broadcasting vector components against spatially encoded matrix rows and accumulating row outputs in MSD form.[cite:138] [cite:142] [cite:149] [cite:154] The resulting journal narrative is therefore coherent: the material platform supplies a robust ternary optical alphabet, while the MSD algorithms supply an arithmetic formalism whose locality matches the spatial parallelism of the device. [cite:145] [cite:161]