Exact Identities for the Binary Hamming Weight Under Arithmetic and Bitwise Operations

Ricardo Adonis Caraccioli Abrego *

Posted Date: 3 November 2025

doi: 10.20944/preprints202511.0097.v1

Keywords: binary digital sum; Hamming weight; carry propagation; Kummer's theorem; bitwise operations; side-channel

Article

Exact Identities for the Binary Hamming Weight Under Arithmetic and Bitwise Operations

Ricardo Adonis Caraccioli Abrego

Departamento de Ingeniería eléctrica, Universidad Nacional Autónoma de Honduras (UNAH), Campus Cortés; ricardo.caraccioli@unah.edu.hn

Abstract

We collect and prove exact identities for the binary digital sum $S_2(n)$—the Hamming weight $\mathrm{wt}(n)$—under elementary arithmetic and bitwise operations. For $x, y \geq 0$ we derive explicit carry/borrow decompositions of $\mathrm{wt}(x+y)$ and $\mathrm{wt}(x-y)$ in terms of bitwise carries/borrows c_i, b_i (0-based indexing, $c_0 = b_0 = 0$). We restate classical XOR/OR/AND weight identities in a unified notation, give shift–mask lemmas yielding constructive corollaries (e.g., forcing a prescribed Hamming weight), and present a 2-adic reformulation linked to Kummer's theorem. We also discuss algorithmic, hardware, and side-channel applications. Proofs are elementary and self-contained.

Keywords: binary digital sum; Hamming weight; carry propagation; Kummer's theorem; bitwise operations; side-channel

MSC: 11A63, 94B05, 68R15

---

1. Notation and Preliminaries

We work with nonnegative integers in binary. Bits are indexed from 0 (LSB). For $n \geq 0$,

$$\mathrm{wt}(n) = \sum_{i \geq 0} n_i, \quad n = \sum_{i \geq 0} n_i 2^i,\ n_i \in \{0,1\}.$$

Let $x, y \geq 0$. In $x+y$, let $c_i = c_i(x,y) \in \{0,1\}$ be the carry entering bit i (set $c_0 = 0$), and write c_{i+1} for the carry exiting bit i. In $x-y$ with $x \geq y$, define $b_i = b_i(x,y)$ analogously with $b_0 = 0$. For bitwise operations we use $x \,\mathrm{XOR}\, y$, $x \,\mathrm{AND}\, y$, $x \,\mathrm{OR}\, y$, and shifts $x \ll k, x \gg k$.

Proposition 1 (Folklore identities). *For all $x, y \geq 0$,*

$$\mathrm{wt}(x \,\mathrm{XOR}\, y) = \mathrm{wt}(x) + \mathrm{wt}(y) - 2\,\mathrm{wt}(x \,\mathrm{AND}\, y), \tag{1}$$

$$\mathrm{wt}(x \,\mathrm{OR}\, y) = \mathrm{wt}(x) + \mathrm{wt}(y) - \mathrm{wt}(x \,\mathrm{AND}\, y). \tag{2}$$

Proof. Count ones positionwise: XOR marks disagreement; OR marks at-least-one. □

2. Carry and Borrow Decompositions

Fix $m \geq 0$ such that $x, y < 2^{m+1}$. All sums below are taken over $i = 0, \ldots, m$. We also make c_{m+1} and b_{m+1} explicit (final carry/borrow).

Theorem 1 (Addition: exact carry decomposition). *For all $x, y \geq 0$, with $c_0 = 0$,*

$$\boxed{\mathrm{wt}(x+y) \;=\; \mathrm{wt}(x) + \mathrm{wt}(y) \;+ \sum_{i=0}^{m} (c_i - 2c_{i+1}) \;=\; \mathrm{wt}(x) + \mathrm{wt}(y) - \sum_{i=0}^{m} (c_{i+1} - c_i) - c_{m+1}} \tag{3}$$

 doi:10.20944/preprints202511.0097.v1

Proof. At bit i: $x_i + y_i + c_i = s_i + 2c_{i+1}$. Sum $i = 0..m$ and identify $\sum x_i = \mathrm{wt}(x)$, $\sum y_i = \mathrm{wt}(y)$, $\sum s_i = \mathrm{wt}(x+y)$. □

Theorem 2 (Subtraction: exact borrow decomposition). *Let $x \geq y \geq 0$. With $b_0 = 0$,*

$$\boxed{\mathrm{wt}(x-y) \;=\; \mathrm{wt}(x) - \mathrm{wt}(y) \;+ \sum_{i=0}^{m}(b_{i+1} - b_i) + b_{m+1} \;=\; \mathrm{wt}(x) - \mathrm{wt}(y) - \sum_{i=0}^{m}(b_i - 2b_{i+1})} \quad (4)$$

For $x \geq y$ we have $b_{m+1} = 0$.

Remark 1 (Carries and 2-adic valuation). *The number of carries in adding x and y equals the 2-adic valuation of $\binom{x+y}{x}$:*

$$\#\{i \geq 0 : c_{i+1} = 1\} = \mathrm{v}_2\binom{x+y}{x}. \quad (5)$$

3. Shift–Mask Tools and Constructive Corollaries

Lemma 1 (Shifts). *For $k \geq 0$, $\mathrm{wt}(x \ll k) = \mathrm{wt}(x)$ and $\mathrm{wt}(x \gg k) \leq \mathrm{wt}(x)$.*

Lemma 2 (Mask disjointness). *Let $M = \sum_{j=a}^{a+\ell-1} 2^j$ be a run of ℓ consecutive 1s, disjoint from the support of x. Then $\mathrm{wt}(x \,\mathrm{OR}\, M) = \mathrm{wt}(x) + \ell = \mathrm{wt}(x \,\mathrm{XOR}\, M)$.*

Corollary 1 (Forcing a target Hamming weight). *For any x and any integer $t \geq \mathrm{wt}(x)$, there exists a mask M (a disjoint run of ones) such that $\mathrm{wt}(x \,\mathrm{OR}\, M) = t$.*

4. Applications and Bounds

Write $L = \mathrm{bl}(\max\{x,y\}) = \lfloor \log_2(\max\{x,y\}) \rfloor + 1$ for the active bit-length, and $C = \sum_{i=0}^{m} c_{i+1}$ for the total number of carries (so $C = \mathrm{v}_2\binom{x+y}{x}$ by Remark 1).

Corollary 2 (Bit-length lower bound). *From (3) and $c_i \geq 0$ we have*

$$\mathrm{wt}(x+y) \;\geq\; \mathrm{wt}(x) + \mathrm{wt}(y) \;-\; 2\,C \;\geq\; \mathrm{wt}(x) + \mathrm{wt}(y) \;-\; 2L,$$

since $C \leq L$. This bound is often sharp up to an additive constant.

Corollary 3 (Kummer-based control). *Using $C = \mathrm{v}_2\binom{x+y}{x}$,*

$$\mathrm{wt}(x+y) = \mathrm{wt}(x) + \mathrm{wt}(y) + \sum_{i=0}^{m} c_i - 2\,\mathrm{v}_2\binom{x+y}{x}.$$

Hence any upper bound on $\mathrm{v}_2\binom{x+y}{x}$ (e.g., via binary digit overlaps of x and y) yields a corresponding lower bound on $\mathrm{wt}(x+y)$.

Corollary 4 (Bitwise combinations). *For any x, y,*

$$\mathrm{wt}(x \,\mathrm{XOR}\, y) \leq \mathrm{wt}(x) + \mathrm{wt}(y), \qquad \mathrm{wt}(x \,\mathrm{OR}\, y) \leq \mathrm{wt}(x) + \mathrm{wt}(y),$$

with equality iff $x \,\mathrm{AND}\, y = 0$.

Algorithmic angle

Computing $\mathrm{wt}(x+y)$ directly is $O(L)$ bit-operations (same as addition). The identity (3) expresses $\mathrm{wt}(x+y)$ as a linear form in the carry profile; thus any algorithm or hardware that already exposes carries (e.g., prefix adders like Kogge–Stone/Brent–Kung) gives $\mathrm{wt}(x+y)$ *for free* after addition (one pass of counting). This is useful in:

 doi:10.20944/preprints202511.0097.v1

- **Complexity accounting** in bit-level algorithms: tight bounds on post-addition Hamming weights in dynamic programming or bitset convolution.
- **Word-parallel trickery**: when maintaining population counts under incremental updates, (3) predicts the decrement driven by carry chains.

Hardware angle

In CMOS, dynamic power correlates with bit flips and, in many platforms, with Hamming weights of buses. The carry chain length C (and its distribution) interacts with toggle activity; (3) isolates the linear influence of C on the resulting weight. This connects to prefix adder analyses (Kogge–Stone [6], Brent–Kung [7]).

Side-channel angle

Simple/Differential Power Analysis often models traces via Hamming weight or Hamming distance. Identity (3) provides a *deterministic* link between the weight after addition and the carry chain (which itself depends on operand bit patterns), informing leakage simulators and countermeasures (e.g., operand randomization). See Kocher–Jaffe–Jun [9].

Remark 2 (Towards multiplicative operations)**.** *For multiplication,* $xy = \sum_i x_i (y \ll i)$ *is a sum of* $\mathrm{wt}(x)$ *shifted copies of* y*; hence*

$$\mathrm{wt}(xy) \;\le\; \mathrm{wt}(x)\ \mathrm{bl}(y)$$

by a naive union bound (collisions and carries can reduce the weight). A refined analysis requires carry bookkeeping across the partial-product convolution; we leave a tight identity as future work.

5. Worked Examples and Sanity Checks

Bits are indexed from 0. We verify Theorems 1 and 2 on small pairs.

Example A (Addition)

Let $x = 29 = (11101)_2$, $y = 23 = (10111)_2$. Then $x + y = 52 = (110100)_2$, $\mathrm{wt}(52) = 3$. Active carries:

$$c_0 = 0,\ \ c_1 = c_2 = c_3 = c_4 = c_5 = 1,\ \ c_6 = 0.$$

Thus $\sum_{i=0}^{5}(c_i - 2c_{i+1}) = (0-2) + (1-2) + (1-2) + (1-2) + (1-2) + (1-0) = -5$. By (3):

$$\mathrm{wt}(x+y) = \mathrm{wt}(x) + \mathrm{wt}(y) + \sum_{i=0}^{5}(c_i - 2c_{i+1}) = 4 + 4 - 5 = 3.$$

Example B (Subtraction)

Let $x = 53 = (110101)_2$, $y = 19 = (10011)_2$. Then $x - y = 34 = (100010)_2$ has wt $= 2$. A borrow trace gives an active window with $\sum(b_{i+1} - b_i) = 0$, hence $\mathrm{wt}(x-y) = \mathrm{wt}(x) - \mathrm{wt}(y) = 4 - 3 = 1$ plus a single bit created by the borrow cascade, totaling 2, matching (4).

A small table:

Table 1. Sanity checks consistent with (3) and (4).

x	y	$\mathrm{wt}(x)$	$\mathrm{wt}(y)$	$x+y$	$\mathrm{wt}(x+y)$	$x-y\ (\ge 0)$	$\mathrm{wt}(x-y)$
5	3	2	2	8	1	2	1
29	23	4	4	52	3	6	2
37	14	3	3	51	3	23	4

 doi:10.20944/preprints202511.0097.v1

6. 2-Adic Perspective and Kummer Linkage

Kummer's theorem states that the exponent of a prime p dividing $\binom{u}{v}$ equals the number of carries when adding v and $u-v$ in base p. For $p=2$,

$$\mathrm{v}_2\binom{x+y}{x} = \#(\text{carries in adding } x \text{ and } y).$$

Combining with Theorem 1 yields an explicit reformulation of $\mathrm{wt}(x+y)$ in terms of $\mathrm{v}_2\binom{x+y}{x}$ and $\{c_i\}$.

7. Positioning, Prior Art, and Contributions

Digital sums in base 2 go back to Delange and Coquet; Allouche–Shallit systematized many properties via automata. The identities here are elementary but presented as a consolidated, explicit package with constructive corollaries and a 2-adic bridge. Connections to prefix adders and leakage models highlight practical relevance.

8. Conclusions

We provided exact weight identities under addition, subtraction, and bitwise operations; shift–mask tools; bounds in terms of bl and Kummer; and application pointers (algorithms, hardware, side-channels). Future work: tight multiplicative identities and carry-profile statistics.

References

1. H. Delange, Sur la fonction sommatoire de la fonction somme des chiffres, *Enseign. Math.* (2) **21** (1975), 31–47.
2. J. Coquet, A summation formula related to the binary digits, *Invent. Math.* **73** (1983), 107–115.
3. J.-P. Allouche and J. Shallit, *Automatic Sequences: Theory, Applications, Generalizations*, Cambridge Univ. Press, 2003.
4. E. E. Kummer, Über die ergänzungssätze zu den allgemeinen Reciprocitätsgesetzen, *J. Reine Angew. Math.* **44** (1852), 93–146.
5. P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge Univ. Press, 2009.
6. P. M. Kogge and H. S. Stone, A parallel algorithm for the efficient solution of a general class of recurrences, *IEEE Trans. Comput.* **C-22** (8) (1973), 786–793.
7. R. P. Brent and H. T. Kung, A regular layout for parallel adders, *IEEE Trans. Comput.* **C-31** (3) (1982), 260–264.
8. D. E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 3rd ed., Addison-Wesley, 1998.
9. P. Kocher, J. Jaffe, and B. Jun, Differential Power Analysis, *CRYPTO 1999*, LNCS 1666, 388–397.