Smaller Circuits for Bit Addition

Mikhail Goncharov, Alexander Kulikov, Georgy Levtsov

STACS 2026

Contributions: One Slide Overview

Theory. Circuit size of bit addition:

$5n - 3m$	[Dadda, 65]
$4.5n - 2m$	[this paper]

Contributions: One Slide Overview

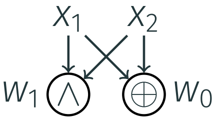
Theory. Circuit size of bit addition:

$5n - 3m$	[Dadda, 65]
$4.5n - 2m$	[this paper]

Practice. Generators: `Cirbo`. IWLS 2024 Contest:

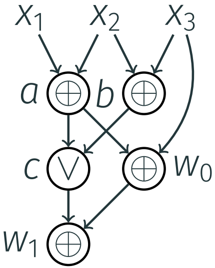
function	2023	team	our	
modulo8	1158	DeepMind	190	83%
sort15	114	TUW	73	35%

Computational Model: Boolean Circuits




```python
def half_adder(x1, x2):
    w0 = x1 ^ x2
    w1 = x1 * x2
    return w0, w1
```

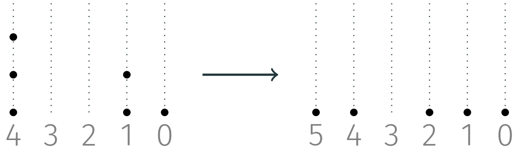
$$\mathrm{HA}(x_1, x_2) = (w_0, w_1)$$
$$x_1 + x_2 = w_0 + 2w_1$$




```python
def full_adder(x1, x2, x3):
    a = x1 ^ x2
    b = x2 ^ x3
    c = a | b
    w0 = a ^ x3
    w1 = c ^ w0
    return w0, w1
```

$$\mathrm{FA}(x_1, x_2, x_3) = (w_0, w_1)$$
$$x_1 + x_2 + x_3 = w_0 + 2w_1$$

Bit Addition and Dot Notation

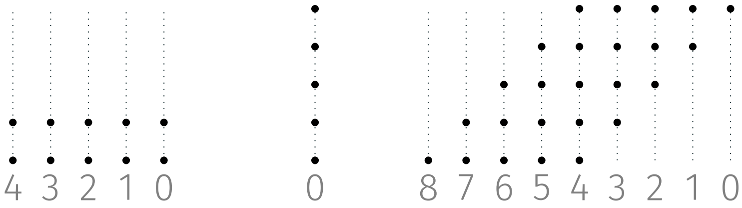


$$\mathrm{BA}_6^{0,1,1,4,4,4} \colon \{0,1\}^6 \to \{0,1\}^5$$

$$\mathrm{BA}_6^{0,1,1,4,4,4}(x_0, x_1, x_2, x_3, x_4, x_5) = (y_0, y_1, y_2, y_4, y_5)$$

$$x_0 + 2x_1 + 2x_2 + 16x_3 + 16x_4 + 16x_5 = y_0 + 2y_1 + 4y_2 + 16y_3 + 32y_5$$

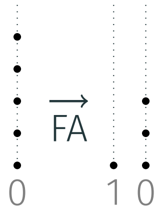
Various Cases



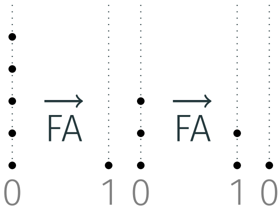
ADD_5 SUM_5 $MULT_5$

Computing the Sum of Five Bits

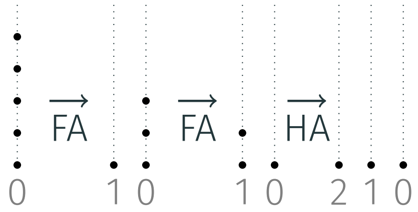
Computing the Sum of Five Bits



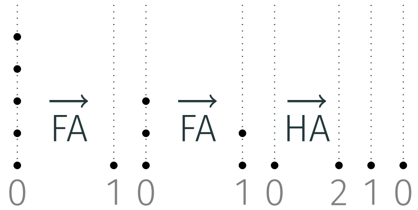
Computing the Sum of Five Bits

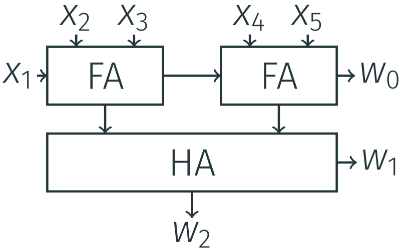


Computing the Sum of Five Bits

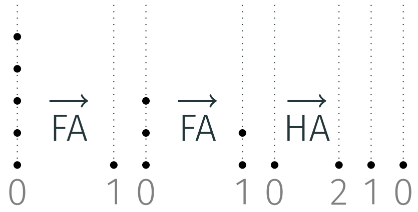


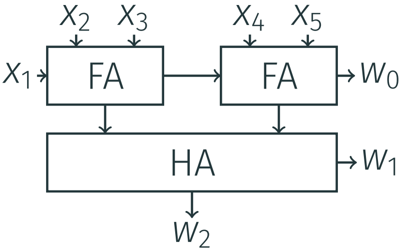
Computing the Sum of Five Bits

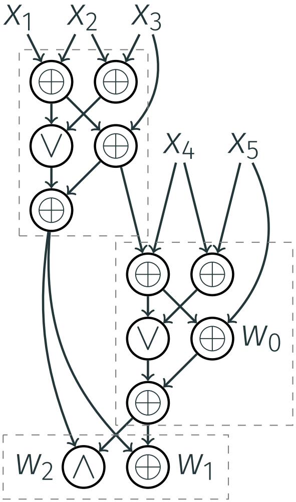




Computing the Sum of Five Bits





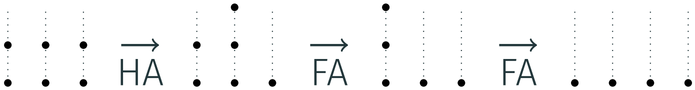


Synthesizing a Circuit for Bit Addition

- On each layer, apply FA's exhaustively, then apply HA if needed. The resulting size is at most $5(n-m)+2m=5n-3m$ [Dadda, 1965]

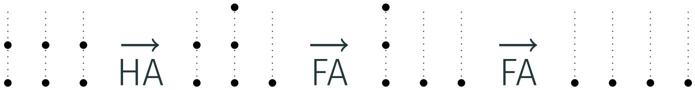
Synthesizing a Circuit for Bit Addition

- On each layer, apply FA's exhaustively, then apply HA if needed. The resulting size is at most $5(n-m)+2m = 5n-3m$ [Dadda, 1965]
- Example: computing the sum of two three-bit integers:



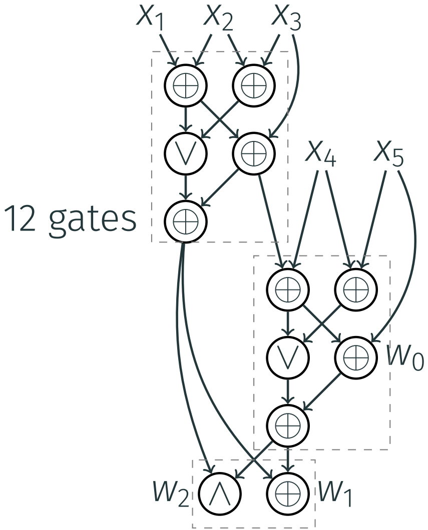
Synthesizing a Circuit for Bit Addition

- On each layer, apply FA's exhaustively, then apply HA if needed. The resulting size is at most $5(n-m)+2m = 5n-3m$ [Dadda, 1965]
- Example: computing the sum of two three-bit integers:

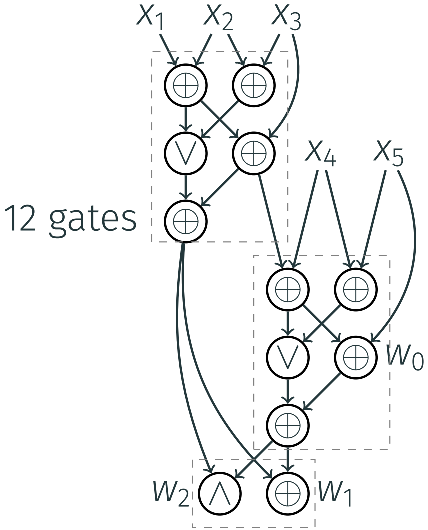


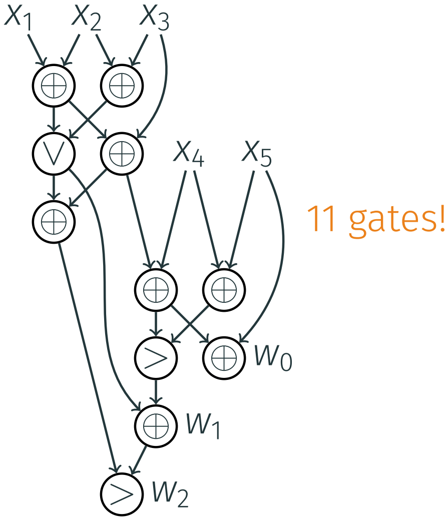
- For adding two n-bit integers, this gives a circuit of size $5n-3$ and it is provably optimal [Red'kin, 1981]

Surprise: $\text{size}(\text{SUM}_5) = 11$

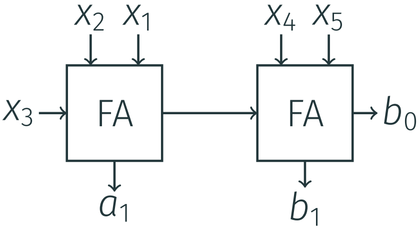


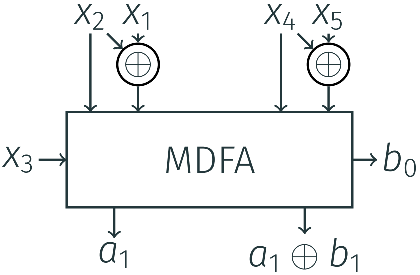
Surprise: $\text{size}(\text{SUM}_5) = 11$

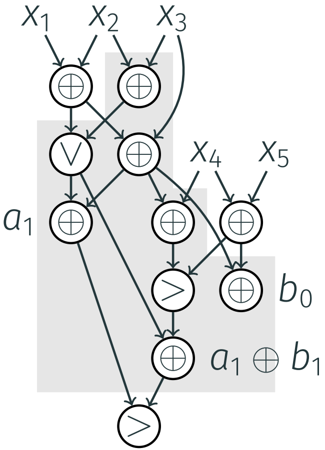




Modified Double Full Adder







Theory: New Upper and Lower Bounds

Upper bounds: $\text{size}(\mathrm{BA}_n^S) \leq 4.5n - 2m$

This generalizes $\text{size}(\mathrm{SUM}_n) \leq 4.5n$ [Demenkov et al. 2010] and improves $\text{size}(\mathrm{BA}_n^S) \leq 5n - 3m$ [Dadda 1965]

Theory: New Upper and Lower Bounds

Upper bounds: $\text{size}(\text{BA}_n^S) \leq 4.5n - 2m$

This generalizes $\text{size}(\text{SUM}_n) \leq 4.5n$ [Demenkov et al. 2010] and improves $\text{size}(\text{BA}_n^S) \leq 5n - 3m$ [Dadda 1965]

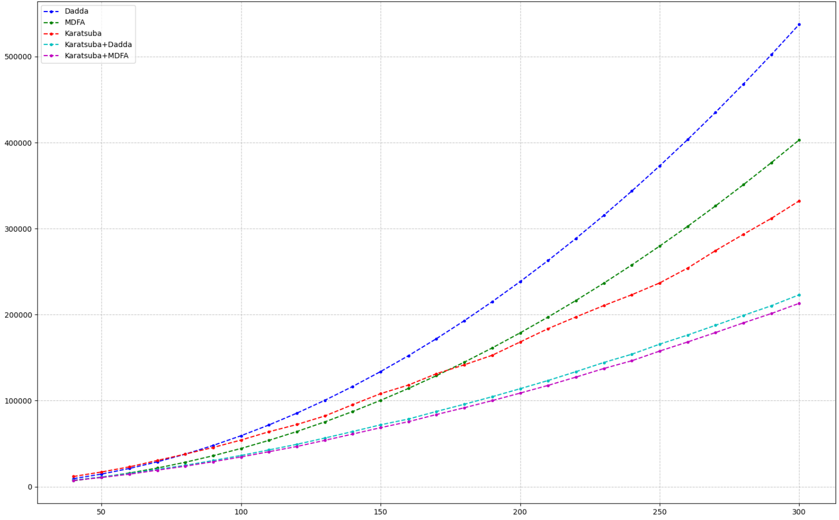
Lower bounds: $\text{size}(\text{INC}_n) = 2n \pm c$

Hence, $\text{size}(\text{BA}_n^S) \not\leq 5n - 3.1m$ and $\text{size}(\text{BA}_n^S) \not\leq 4.5n - 2.6m$

Practice: IWLS 2024 Contest on Logic Synthesis

function	2023	team	our	
modulo8	1158	DeepMind	190	83.59%
square12	1284	DeepMind	324	74.77%
div8	306	DeepMind	142	53.59%
sqrt16	226	DeepMind	136	39.82%
sort15	114	TUW	73	35.96%
maj15	68	TUW	48	29.41%

Practice: Generators for $MULT_n$



Contributions and Open Problems

Contributions

- **Upper bounds:** $\mathsf{size}(\mathrm{BA}_n^s) \leq 4.5n - 2m$
- **Lower bounds:** $\mathsf{size}(\mathrm{INC}_n) = 2n \pm c$
- **Software:** `Cirbo` package

Open problems

- Close the gap $2.5n \leq \mathsf{size}(\mathrm{SUM}_n) \leq 4.5n$!
- Improve $4.5n - 2m$ further!