Errata Sheet

Application-Specific Arithmetic

Computing Just Right
for the Reconfigurable Computer
and the Dark Silicon Era

Florent de Dinechin and Martin Kumm
2024

Unfortunately, mistakes appear even after several rounds of serious proof-readings by ourself, colleagues and editors. The best way to deal with them is to publish the corrections. This document collects and corrects the errors we made (as far as we know them).

If you find further errors, don't hesitate to tell us, please!

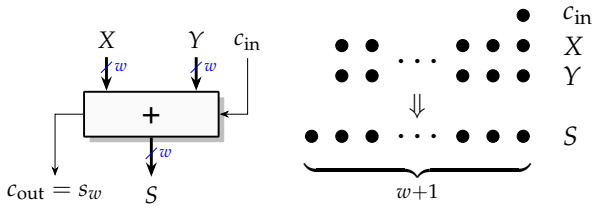
Chapter 7: Sums of Weighted Bits

- Page 180, Table 7.2 has two wrong entries, the correct table is given below

Table 7.2 Resource comparison of the example compressor trees obtained by Dadda (Fig. 19) and RA (Fig. 20)

Method	FA	HA	final adder length	FF (fully pipelined)
Dadda	36	6	14	179
RA	39	7	10	140

- Page 155, Figure 7.5 has an S wrongly placed, the correct figure is given below.



(a) *Integer adder* (b) *Corresponding dot diagram*

Fig. 7.5 The integer adder viewed as a compressor

- Page 184, Table 7.3, some entries where not updated to the updated references and one reference was wrong, changes are marked red below.

Table 7.3 High-efficiency GPCs, counters and adders divided into LUT-based GPCs/-counters (top), LUT- & carry chain-based GPCs/counters targeting AMD FPGAs (middle) and row adders targeting AMD FPGAs (bottom).

GPC/ row adder	Ref.	#LUT6 (k)	Efficiency ($E = \delta/k$)	Delay
(3;2)	[Bru+13]	1	1	$\tau_L \approx \tau$
(6;3)	[Bru+13]	3	1	$\tau_L \approx \tau$
(1,5;3)	[Bru+13]	3	1	$\tau_L \approx \tau$
(7;3)	[Yua+19]	2	**2**	$\tau_L + 2\tau_{CC} \approx \tau$
(2,3;3)	[KZ14a]	2	1	$\tau_L + 2\tau_{CC} \approx \tau$
(1,4,1,5;5)	[KZ14a]	4	1.5	$\tau_L + 4\tau_{CC} \approx \tau$
(1,4,0,7;5)	[Yua+19]	4	1.75	$\tau_L + 4\tau_{CC} \approx \tau$
(1,3,2,5;5)	[KZ14b]	4	1.5	$\tau_L + 4\tau_{CC} \approx \tau$
(6,2,3;5)	[Pre17]	4	1.5	$\tau_L + 4\tau_{CC} \approx \tau$
(6,0,7;5)	[Yua+19]	4	2	$\tau_L + 4\tau_{CC} \approx \tau$
(6,1,5;5)	[Pre17]	4	1.75	$\tau_L + 4\tau_{CC} \approx \tau$
(2,1,1,7;5)	[Yua+19]	4	1.5	$\tau_L + 4\tau_{CC} \approx \tau$
2-input add.		k	1	$\tau_L + k\tau_{CC}$
ternary add.		k	$2 - \frac{2}{k}$	$2\tau_L + \tau_R + k\tau_{CC} \approx 3\tau + k\tau_{CC}$
4:2 compressor	[KZ14a]	k	$2 - \frac{1}{k}$	$\tau_L + k\tau_{CC}$

- Page 185, Fig. 7.23, the figure was not updated with Table 7.3 and partly shows different GPCs. The correct figure is below.

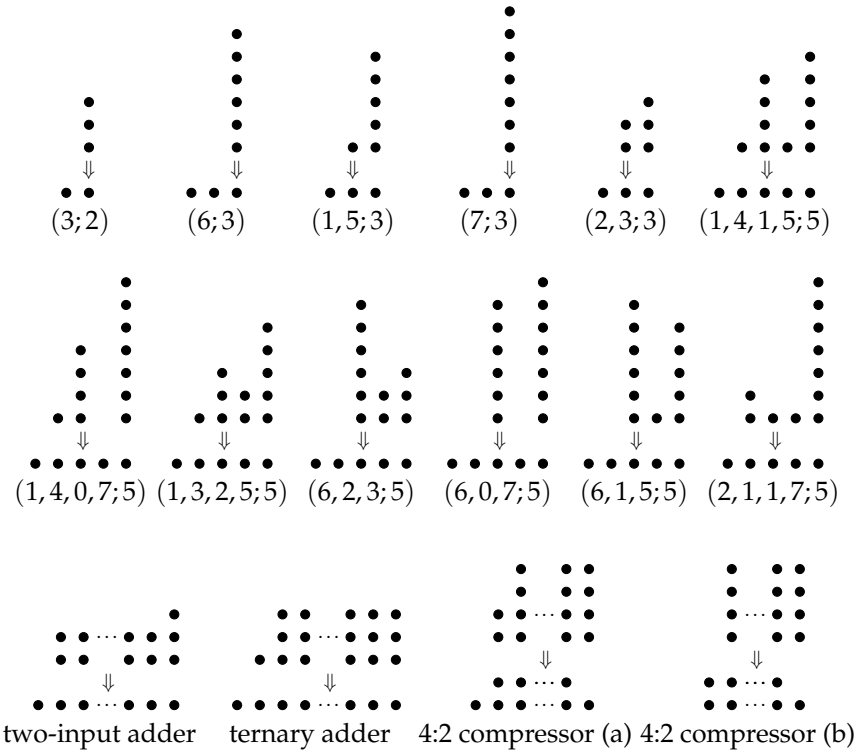


Fig. 7.23 Dot transformations for the GPCs and row adders from Tab. 7.3

Chapter 24: Arithmetic for Deep Learning

- Page 737, last line: "A one-dimensional 2 by 2 convolution" should be "A one-dimensional 3 by 2 convolution".

References

[Bru+13] Nicolas Brunie, Florent de Dinechin, Matei Iştoan, Guillaume Sergent, Kinga Illyes, and Bogdan Popa. "Arithmetic Core Gen-

eration Using Bit Heaps". In: *Field Programmable Logic and Application (FPL)*. IEEE, 2013, pp. 1–8 (cit. on p. 2).

[KZ14a] Martin Kumm and Peter Zipf. "Efficient High Speed Compression Trees on Xilinx FPGAs". In: *Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV)*. 2014, pp. 171–182 (cit. on p. 2).

[KZ14b] Martin Kumm and Peter Zipf. "Pipelined Compressor Tree Optimization Using Integer Linear Programming". In: *International Conference on Field Programmable Logic and Application (FPL)*. IEEE, 2014, pp. 1–8 (cit. on p. 2).

[Pre17] Thomas B. Preußer. "Generic and Universal Parallel Matrix Summation with a Flexible Compression Goal for Xilinx FPGAs". In: *International Conference on Field-Programmable Logic and Applications (FPL)*. IEEE, 2017, pp. 1–7 (cit. on p. 2).

[Yua+19] Yuelai Yuan, Le Tu, Kan Huang, Xiaoqiang Zhang, Tiejun Zhang, Dihu Chen, and Zixin Wang. "Area Optimized Synthesis of Compressor Trees on Xilinx FPGAs Using Generalized Parallel Counters". In: *IEEE Access* 7 (2019), pp. 134815–134827 (cit. on p. 2).