Design of digital integrated systems: optimization of a 16 bit Brent-Kung adder

Matthijs Van keirsbilck

November 23, 2016

1 Schematic and optimisation

1. This is a numbered item
2. Another numbered item
3. Same as 1

Cross-referencing items 2 and 3.

1.1 Optimisation

1. Architectural

- To remove inverters from the critical path, as a first attempt, the DotProducts were replaced with equivalent NAND-based operators. This was quite easy to implement, as only one type of DotProduct was needed, and provided a significant increase in performance.
- Even better performance was reached by using AOI and OAI- based DotOperators, where the DotProducts are replaced with DotProductNormal and DotProductInverse. In the rest of the report, that architecture will be described. In the top half of the structure, most DotProducts are full-size (both generate and propagate). Lower down, the propapagate signal is often not use, so some gates can be removed, leading to lower power. As the number of normal/inverting stages is not equal for all paths, some special operators are required as well. DotProductSimpleNormalHighInvertedLow for example is a dotproduct that doesn't generate a propagate signal, and takes as input the P_h and G_h as well as $\overline{G_l}$. In total, there are six different DotOperators (with their abbreviations between brackets):

  (a) DotOperatorNormalIn ('DON')
  (b) DotOperatorInvertedIn ('DOI')
  (c) DotOperatorSimpleNormalIn ('DOSN')
  (d) DotOperatorSimpleInvertedIn ('DOSI')
  (e) DotOperatorSimpleNormalHighInvertedLow ('DOSNHIL')
  (f) DotOperatorSimpleInvertedHighNormalLow ('DOSIHNL')

  See the circuit for the structure 7.
- An efficient 6-transistor implementation of the XOR gate (based on transmission gates) decreases delay by a lot, because of less transistors, faster switching speed and reduced load on the critical path. Because only one consecutive transmission gate stage is used, output isn't degraded much and no buffers are needed (see [1]).

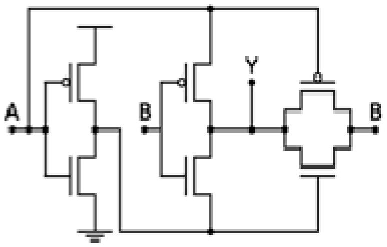


Figure 1: Schematic of the optimised XOR gate

- The building blocks (AOI, OAI, NAND, XOR, NOT) were optimized in size to take into account that some transistors were placed in parallel or in series, respectively increasing or reducing the current in that branch. See Sizing.
- If buffers in the structure buffer a large subcircuit, they will significantly reduce the load on critical path, making it faster. The buffers increase the delay of the path they are placed on, however. This could make those paths the new bottlenecks. The only two minimum-sized buffers that were introduced are situated quite low in the hierarchy so that they don't increase the delays of too many paths (see Figure 8).

2. Sizing

   For achieving maximum speed, widths of some transistors were changed by some scalar factor:

   - pMOSscalar: normally, pMOS devices are sized about twice as large as nMos devices to keep switching symmetrical (this is needed due to lower mobility of holes compared to electron)s. For this adder design, however, lower pMOS width means lower capacitance, and this results in faster switching speeds. Too low values mean that the asymmetricalities will increase delay, too high values increase the capacitive load and reduce delay as well. This trade-off resulted in an optimal pMOS sizing of 1.6 times the nMOS size.
   - seriesScalar: in for example a NAND gate, the 2 nMOS devices are placed in series between output and ground. This results in higher series resistance for the current flow and an extra capacitive node that needs to be drained. Series devices can provide less current than a single device due to these reasons, while devices in parallel can provide double the current. To compensate for these effects, transistors in series configurations are scaled by a factor of seriesScalar.
   - critBasePath: The top right half of the adder (until input7, and DotOperator(7_0)) is very important for all critical paths. This scalar modifies the widths of transistors in this area.
   - critPath1: The most critical path (6 DotOperators), that runs diagonally from input0 to s15. Increasing its size improves speed, but also increases loading.
   - critPath2: The secondmost critical path. If the first critical path is sized large enough, it will no longer have the largest delay, and this second critical path needs to be scaled as well. This increases the load on the first critical path, increasing its delay. A balance needs to be found.
   - XORScalar: some of the paths are more critical than others. For precise adjustments to the path delay, the XOR gates at the end of the path can work faster if they are scaled, without modifying the whole path (which would have a large effect on the loading as well). The most important paths at s15 and s9 were scaled this way.
   - VDD: this reduces the DC power consumtion and the switching energy, but increases the delay. When a good Energy-Delay Product (EDP) is reached, scaling the voltage reduces power consumption. Vdd= 0.93V was the lowest supply voltage where the circuit still fulfilled the specifications.

2 Results

Two different architectures are shown, one that targets maximum possible speed, and another that targets minimum power with a delay constraint of 650ps. To measure these results, one test file was run for each path. Propagation of an input change through all important critical paths were simulated throuth a '.vec' file for each path (thanks to Bob Vanhoof for providing some basic test files). The shown delay, switching energy, and EDP are the worst (maximum) delay of all test cases (not necessarily the same one). The DC power obviously remains constant for a given architecture.

2.1 Max speed

scalar	value
pMOSscalar	1.6
seriesScalar	2.6
critBase	1.4
critPath1	1.8
critPath2	1.1

Table 1: Size Scaling factors for maximum speed

Supply	1 V
Worst Case delay	511 ps
Worst Case Switching energy	135 fJ
Worst Case DC power	1.73 nW
Worst Case EDP	69045 ps*fJ

Table 2: Performance of the circuit tuned for maximum speed

2.2 Minimum Power @ 650ps

scalar	value
pMOSscalar	1
seriesScalar	2.6
critBase	1
critPath1	1.5
critPath2	1

Table 3: scaling factors for minimum power

2.3 performance with pMOS width = 1.0

For the maximum speed circuit, the delay goes up when pMOSscalar is lowered too much, but this is not the case for the minimum power circuit (the only difference between the circuits is the sizing of the critical paths). For the minimum power circuit, it is very surprising that the extremely low width of pMOS transistors doesn't increase delay (since it introduces asymmetricalities), and seems to improve the circuit by a large margin (EDP of 55k while it was > 70k before). The reason the simulation gave these results might be because the test cases don't cover every possible switching event. Therefore, the simulation above, with more conservative pMOS sizes obtained from the maximum speed optimizations is considered to show the actual performance of the circuit. In order to reach DC power < 1nW, a few parameters were tuned. See Adder16b_BrentKung_Power_Pmos1.m2s .

Supply	0.93 V
Worst Case delay	648 ps
Worst Case Switching energy	113 fJ
Worst Case DC power	1.936 nW
Worst Case EDP	733578 ps*fJ

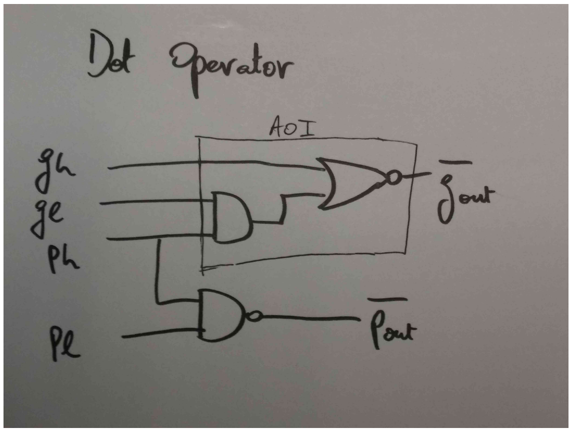
Table 4: Final performance of the circuit tuned for minimum power and 650ps delay

Supply	0.932 V
Worst Case delay	648.1732 ps
Worst Case Switching energy	85.224 fJ
Worst Case DC power	0.999 nW
Worst Case EDP	55240 ps*fJfJ

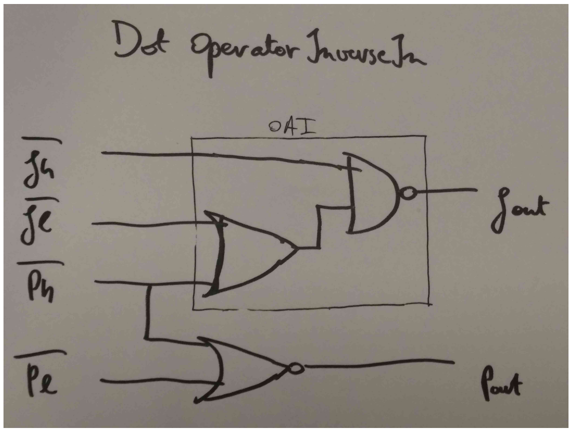
Table 5: Performance tuned for minimum power and 650ps delay, with pMOSscalar = 1.0

2.4 Schematic

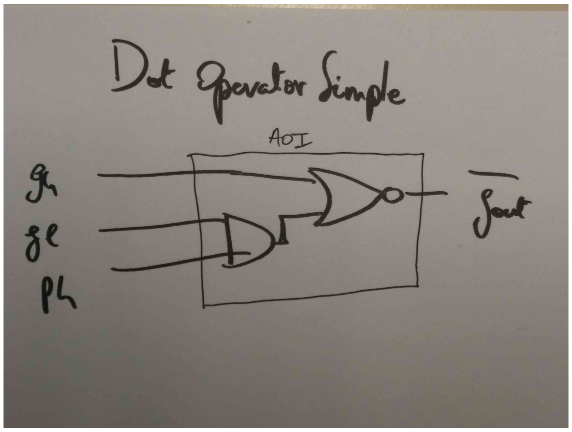
2.4.1 DotOperators



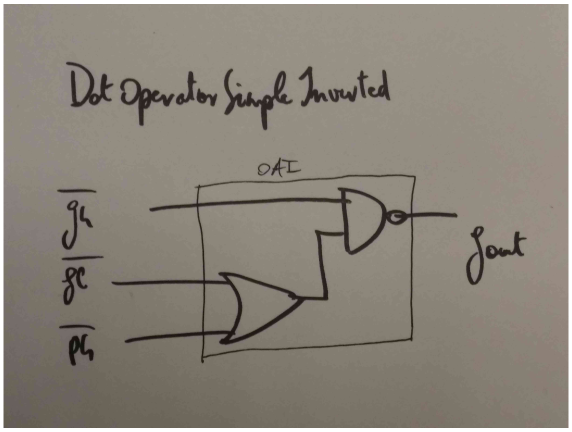
(a) Schematic of the Dot Operator



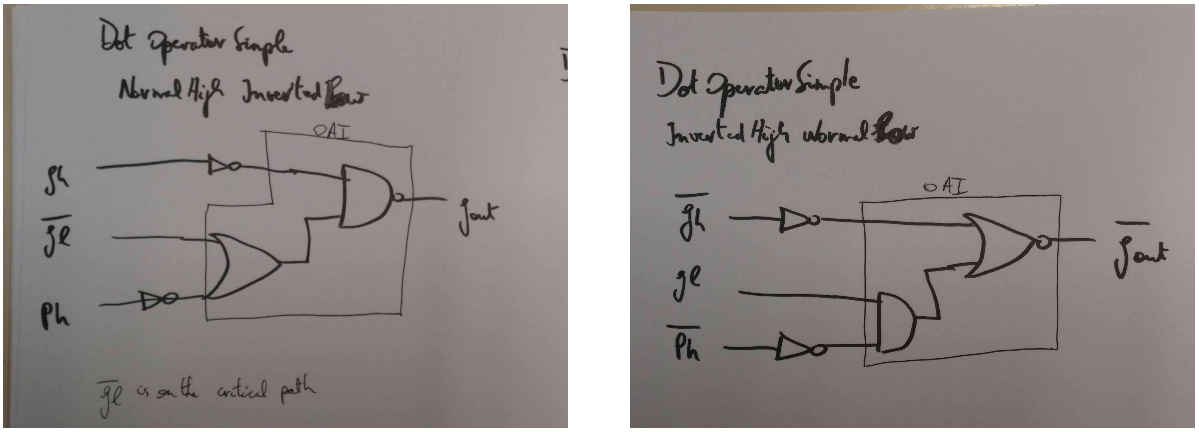
(b) Schematic of the Dot Operator with inverted inputs



(a) Schematic of the Dot Operator Simple (without propagate generation)

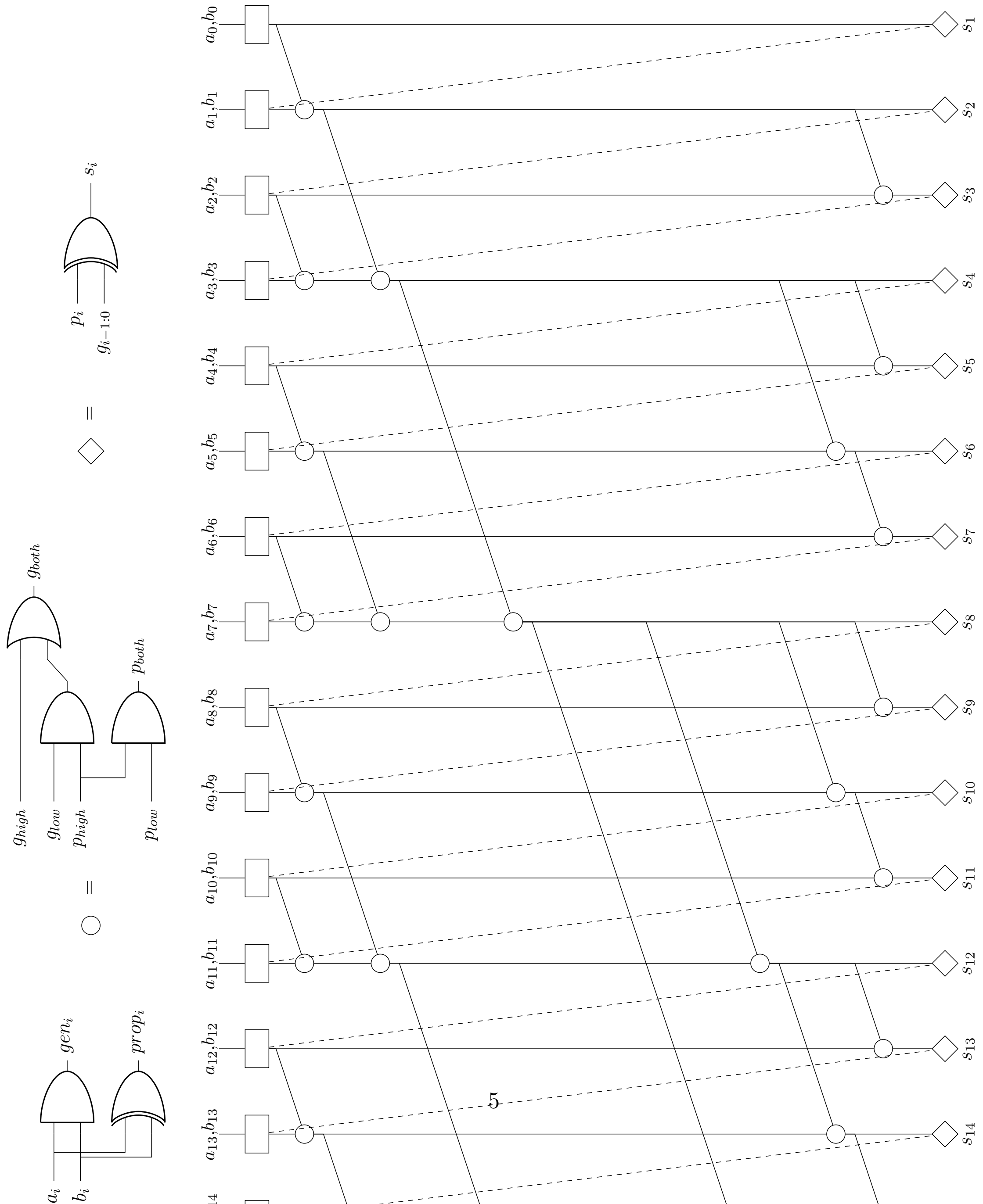


(b) Schematic of the Dot Operator Simple, Inverted inputs



(a) Schematic of the Dot Operator Simple Normal High inputs, Inverted Low inputs

(b) Schematic of the Dot Operator Simple Inverted High inputs, Normal Low inputs



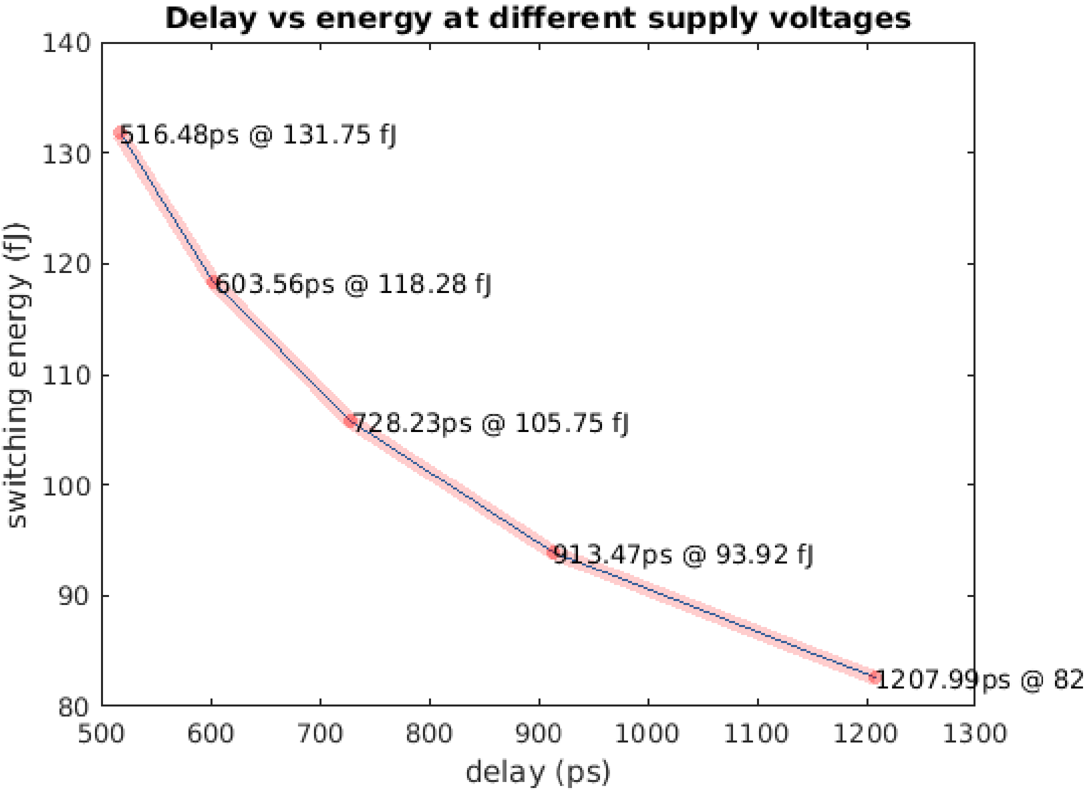


Figure 6: EDP of the optimised adder

References

[1] Universitat Hamburg *CMOS transmission-gate XOR gate* accessed: 28/10/2016
`https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/05-switched/40-cmos/xor-tgate.html`

[2] Noel Daniel Gundi *Implementation of 32 bit Brent-Kung adder using complementary pass transistor logic* accessed: 28/10/2016
`https://shareok.org/bitstream/handle/11244/25747/Gundi_okstate_0664M_13905.pdf?sequence=1`