Digital Design: exercise session 3

Weijie Jiang, Ce Ma, Wim Dehaene

January 16, 2025

1 Session introduction

By now, you should be familiar with the design, validation and characterization in hspice of standard gates. One step further, in this session we will target a more complex structure: **the adder of a microprocessor**. Concretely, the Brent-Kung topology depicted in figure 1. A Brent-Kung adder is a traditional prefix adder, whereby a carry-generate and a carry-propagate is generated for specific sections in a logaritmic tree. In this session we will discuss speed improvements to the Brent-Kung adder. The adder is implemented in hspice and stimulated with a .vec file.

However, complex designs also lead to a more complex characterization. In the case of the adder, there are 2 inputs of 4 bits each, meaning 256 different input combinations. Moreover, the output consists of 17 bits that can transition. Therefore, each input combination requires measuring 17 bits.

As designers, before trying any adder optimization technique, we will first try to remove complexity and save our time. To do so, common practices are:

1. **Identify the bottleneck in our design**. It is obvious that speed optimizations should be carried out on the slowest or critical path. Consequently, a smart selection of the input combinations will simplify the analysis and simulation, saving a considerably amount of time. On the other had, power optimizations should target the block/cell with the highest energy consumption.
2. **Embed hspice in any known scripting language (python, matlab...)** to automate the simulation (introducing loops structures, sweeps...). In this session we will use Matlab.

2 Environment setup

1. Boot/login in linux
2. Open a terminal and type the following lines:

   ```
   cd ~/Documents/DigitalDesign/
   cp -rP ~wdehaene/Public/DDIC/Project/DD_Es3 DD_Es3
   cd DD_Es3
   ```

3. Once there, open MATLAB within that folder.

   ```
   matlab -softwareopengl &
   ```

4. Open "Adder.m" file. This file automatically runs the Hspice simulation "spicefiles/Adder16b_BrentKungStandard.tr" file. To plot a hspice voltage signal within matlab type or add in the code:
   plotsig(transientsim,'v__*NodeName0***;v__***NodeName1***;......;v__***NodeNameN***')**

As mentioned before, we will use **Matlab** (although any scripting language is valid) to automate the simulation:

- Look at "Adder16b_BrentKungStandard.m2s". The novelty lies on the the lines starting with the $ symbol. Those lines denote Matlab commands. Matlab is thus used as a preprocessor. We make use of *for loop* statements to easily create the complementary of each input bit (lines 26-32) and to assign an inverter load to each output bit (lines 38-40).
- We encapsulate the Hspice code in the matlab file "Adder16.m". This file runs the Hspice simulation "Adder16b_BrentKungStandard.m2s". Then, it stores all the input/output signals and performs a deeper analysis. There is no need to go in detail on this code.

3 DELAY OPTIMIZATIONS

3.1 Critical path and delay

Before conducting any speed optimization in the adder, we need to identify the critical path, as this path determines the speed limit.

1. Discuss and draw what you think the critical path is on figure 1.

Have a look at the hspice implementation and run the MATLAB file. Look at the results:

2. What is the adder delay?
3. What is the maximum frequency at which the adder could be operated?

Now look at the delay calculation in matlab (lines 35-46 in "Adder16.m") and the input vectors pattern in "Adder16.vec".

4. How are the delays measured? Explain which input combinations are generated, which output transitions are expected and how this information is used in matlab to measure the delay and determine the critical path.
5. Is the correct critical path simulated? If you were the designer, which other approaches could you have used?
6. If now your boss decides to use an 32-bit adder, how would you implement it? How long would it take you?

3.2 Nodal capacitance

The Brent-Kung adder has a very asymmetric structure, compared to other adder topologies. As you can see on figure 1, the adder consists of forward and reverse tree structures.

1. Identify in figure 1 the nodes that are heavily loaded.
2. Verify your thoughts by uncommenting the captab option in "Adder16b_BrentKungStandard.m2s". This command extracts and stores the nodal capacitance values from the simulation in "spicefiles/Adder16b_BrentKungStandard.lis"
3. **To reduce the load of a node, one could insert extra inverters as a buffer mechanism**. Looking at figure 1, which large capacitive nodes would you target? Where would you place the buffers? Re-design the Brent-Kung adder and verify the correct functioning of the adder.
4. Does the critical path stay the same after buffering? If no, where does it move to?
5. Does this improvement help in energy and/or delay?

3.3 Extra improvement

Can you think on a way of optimizing the speed of the adder using the results of Exercises 2 and 5 of previous session 2? (Focus on the cells of the critical path).

4 POWER OPTIMIZATIONS

4.1 Power consumption

In the previous section we discussed the delay/speed of the adder. Another important property of the adder is its power consumption. The software generates two values: "Worst Case Switching Energy" and "DC power consumption".

1. Which one is most relevant for an adder which is operated at 1GHz with an activity of 20% (high performance processor)?
2. What about an adder operated at 10kHz with similar activity (low power embedded processor)?

4.2 Change Vt

Most recent technologies come with two flavors of transistors: the high performance (HP) also know general purpose, and the Low Power (LP) version.

1. What is the difference between these devices?
2. What will be the effect of changing from one type to the other?

In this project, up to now, the LP version was used. The library to be used in Hspice is specified with the *.lib* statement. To use the HP library swap from "tt" to "tt_45HP" in line 18 of "Adder16b_BrentKungStandard.m2s".

3. How does this influence speed, switching energy and static power consumption?
4. For what kind of circuits would you advise the LP version or the HP version?

4.3 Voltage scaling

To lower power consumption even further, voltage scaling could be applied. The dynamic energy has a quadratic dependency on the supply voltage:

$$E_{dyn} = \alpha \times C \times V_{dd}^2 \tag{1}$$

Whereas the static power consumption has the following dependency:

$$P_{leak} = I_{leak} \times V_{dd} \tag{2}$$

Now, your boss asks you to design a 16b Brent-Kung adder operating at 500MHz with minimum power consumption. Considering previous equations, a straightforward way to achieve it is lowering Vdd. Run the SweepVdd script to sweep the adder's supply voltage from 0.75V to 1V.

1. What is the lowest supply voltage you can achieve to satisfy the speed requirement?
2. What is the total energy consumption? What is the most dominant factor?

4.4 Extra improvement

Your boss asks you to further reduce the consumption. This time, we opt for identifying the bottleneck in the circuit. Remove MYAOI gates from vdd (observed/measured) to vdd2 (unobserved) and look at the differences in the consumption. Repeat the same with XOR gates. Can you think on a way of optimizing the power of the adder using the results of exercise 5 of previous session 2?

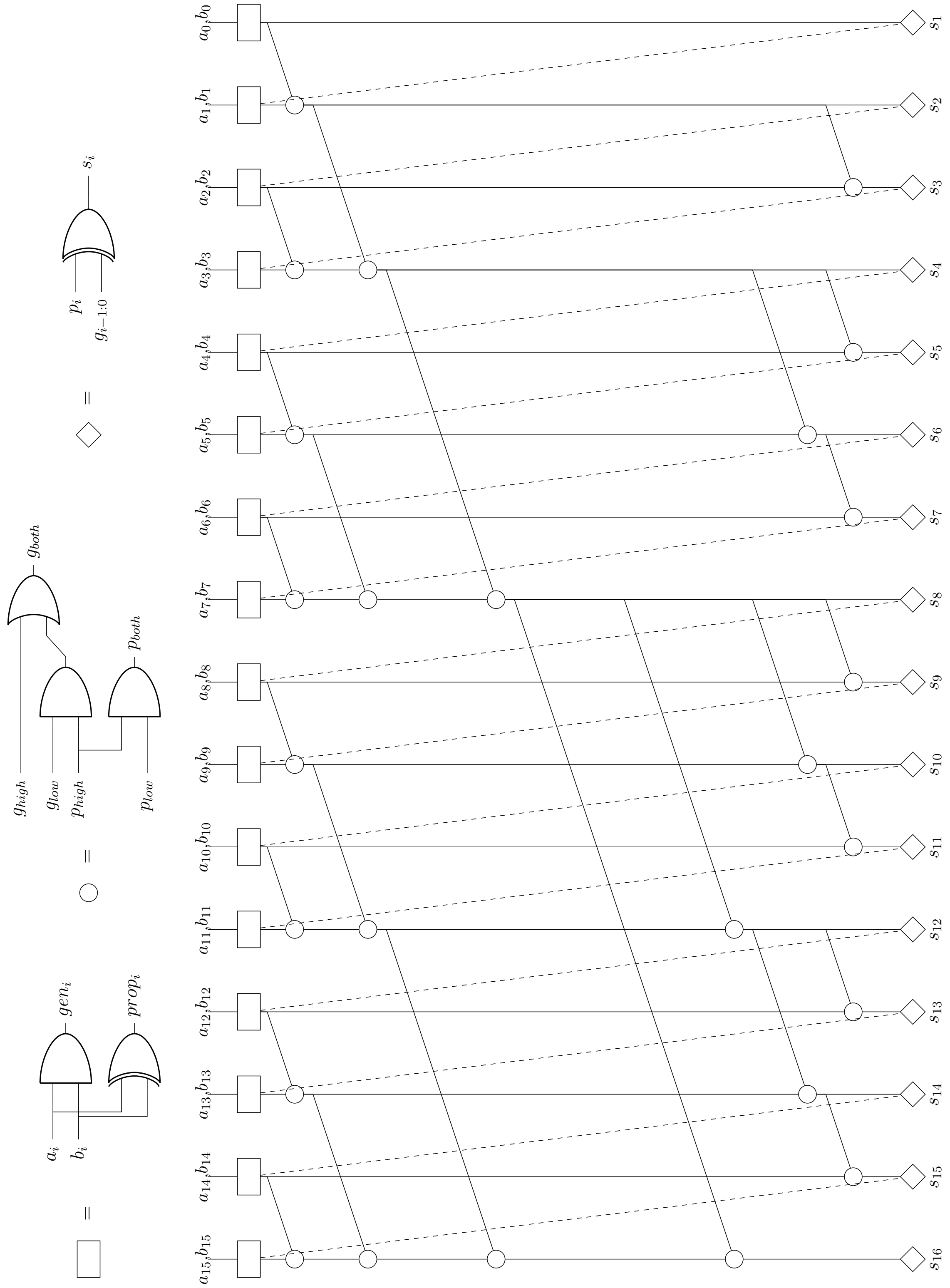

Figure 1: Schematic of the 16 bit Brent-Kung adder