# LAB 2: Prep sheet

* **Pls. upload to EMIL before the lab session by every team member and in PDF format only (otherwise no grading),   
  and in addition**
* **give a hardcopy to the Prof/Lab Assistant at the beginning of the lab session**

|  |  |  |
| --- | --- | --- |
| **Team** | if not a joint solution: tick corresponding author | |
| **Participant 1** | Marco Casagrande |  |
| **Participant 2** | Julius Rauscher |  |
| **Participant 3** | Mahmoud Jadaan |  |
| **Participant 4** | Ievgenii Nudga |  |

**PREP TASK 1.1:** Verify that the equations above and the truth table are equivalent.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  As seen, the truth tables below confirm that the table given in the task is CORRECT.  sn = (an←/→bn) ←/→cn     |  |  |  |  |  | | --- | --- | --- | --- | --- | | an | bn | cn | an←/→bn | sn | | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 1 | | 0 | 1 | 0 | 1 | 1 | | 0 | 1 | 1 | 1 | 0 | | 1 | 0 | 0 | 1 | 1 | | 1 | 0 | 1 | 1 | 0 | | 1 | 1 | 0 | 0 | 0 | | 1 | 1 | 1 | 0 | 1 |       cn+1=(an^bn)v(cn^(anvbn))   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | an | bn | cn | an^bn | anvbn | cn^(anvbn) | cn+1 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 1 | 0 | 0 | 1 | 0 | 0 | | 0 | 1 | 1 | 0 | 1 | 1 | 1 | | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | 1 | 0 | 1 | 0 | 1 | 1 | 1 | | 1 | 1 | 0 | 1 | 1 | 0 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |

**PREP TASK 1.2:** Simulate the 1-bit full adder. Does the simulation agree with the truth table in **Ошибка! Источник ссылки не найден.**? Insert the waveforms in the prep sheet.

|  |
| --- |
| Observation:  The adder works as expected |

**PREP TASK 1.2:** Let N=3. Draw the 3-bit adder and label all signals according to the VHDL model in **Ошибка! Источник ссылки не найден.**. Use the following symbol of a 1-bit full adder in your sketch.



|  |
| --- |
|  |

**PREP TASK 1.3:** Write a test bench for the n-bit adder after reading again the text above about good test benches and test cases.

Write down first at least 5 test cases (here: input stimuli to the n-bit adder) and explain why you have chosen those. In the lab, you will later have to measure the maximum propagation delay that the Ripple-Carry Adder can have. Therefore, your test bench should include a test case in which the maximum propagation delay occurs.

**Extend the .do-file provided in EMIL with your test cases and simulate your VHDL model. Insert your .do-file and the simulated waveform in the prep sheet.**

|  |
| --- |
| Test cases:  force A 05 0ns, 3C 50ns, B5 100ns, CD 150ns, FF 200ns -r 250ns  force B F3 0ns, 9E 50ns, 97 100ns, EF 150ns, FF 200ns -r 250ns  force C\_IN 0 0ns  run 500ns  0-50ns : case of no carry  50-100ns: s[0] produces carry  100-150ns: s[1] produces carry  150-250ns: cascaded carry |
| Simulation results: N\_BIT\_ADDER  nbit_adder_wave |

**PREP TASK 3.1:** Use the same test bench as in PREP TASK 1.3 (minor modifications are required since file and entity name have changed) to simulate the arithmetic adder. Compare the results of 1.3 and 3.1 and insert the waveforms in the prep sheet.

|  |
| --- |
| The arithmetic adder always experiences a constant delay of 5ns, while the n\_bit\_adder simulation delay time varies with the inputs and can add up to 25ns.  Simulation: ARITH\_ADDER  arith_adder_wave |