

Übung 5

Abgabe bis **Donnerstag, 12. Mai 10:00** via EPIIC: <http://ep.iic.jku.at>.

SystemVerilog Dateien müssen vor der Abgabe in EPIIC ausgeführt werden. Dateien, die nicht kompilieren können nicht abgegeben werden.

1. Conditional Sum Addierer (8 Punkte)

Implementiere den Conditional Sum Addierer für 8 Bit Zahlen als SystemVerilog Modul (strukturelle Beschreibung). Erstelle zusätzlich ein Modul *testbench*, das den Addierer für jede mögliche Eingabe testet, also mit der verhaltensbasierten Beschreibung vergleicht.

2. Carry Lookahead Addierer (8 Punkte)

Implementiere den Carry Lookahead Addierer, so wie in der Vorlesung beschrieben für 8 Bit Zahlen als SystemVerilog Modul (strukturelle Beschreibung). Erstelle zusätzlich ein Modul *testbench*, das den Addierer für jede mögliche Eingabe testet, also mit der verhaltensbasierten Beschreibung vergleicht.

3. Kosten von Addierern (8 Punkte)

Nimm an, dass Addierer nur mit 2-Input Gattern (XOR, AND, OR) realisiert werden sollen und Negationen an den Inputs erlaubt sind.

- Berechne die Tiefe und die Anzahl der Gatter für einen n -Bit Conditional Sum Addierer für $n \in 2, 4, 8, 2^k$.
- Berechne die Tiefe und die Anzahl der Gatter für einen n -Bit Carry Lookahead Addierer für $n \in 2, 4, 8, 2^k$.
- Berechne für welche n der n -Bit Conditional Sum Addierer schneller ist als der n -Bit Carry Lookahead Addierer.

4. (Bonus) Parallele Präfix Berechnung (6 Punkte)

Verwende eine parallele Präfix Berechnung (siehe Wikipedia) mit dem assoziativen Carry-Operator $(G_1, P_1) \circlearrowleft (G_2, P_2) = (G_1 + G_2P_1, P_2P_1)$ um die Kosten des Carry Lookahead Addierers aus 2. zu reduzieren. Implementiere den Addierer als SystemVerilog Modul (strukturelle Beschreibung) und teste das Modul entsprechend. Gib in der SystemVerilog Datei zusätzlich die Tiefe und die Anzahl der Gatter als Kommentar an. Verwende nur Gatter zwei Eingängen (XOR, AND, OR). Die Eingänge dürfen negiert werden.