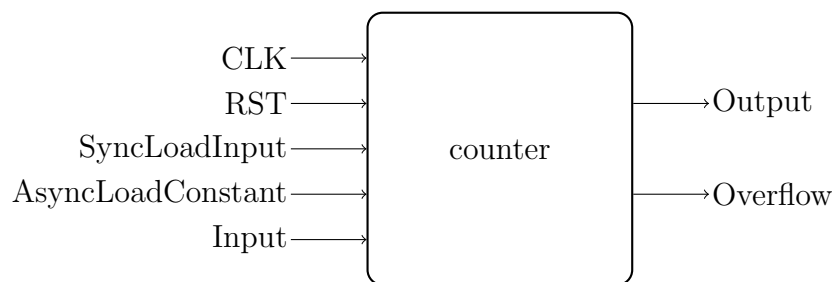


# Counter

Ihre Aufgabe ist es, das Verhalten einer Entity namens „counter“ zu programmieren. Die Entity ist in der angehängten Datei „counter.vhdl“ deklariert und hat folgende Eigenschaften:

- Eingang: CLK vom Typ std\_logic
- Eingang: RST vom Typ std\_logic
- Eingang: SyncLoadInput vom Typ std\_logic
- Eingang: AsyncLoadConstant vom Typ std\_logic
- Eingang: Input vom Typ std\_logic\_vector und der Länge 5
- Ausgang: Overflow vom Typ std\_logic
- Ausgang: Output vom Typ std\_logic\_vector und der Länge 5



Verändern Sie die Datei „counter.vhdl“ nicht!

Die „counter“ Entity soll den Ausgangsvektor Output bei jeder steigenden Flanke des Taktsignals CLK inkrementieren. Der Eingang RST ist ein synchroner Reset, der Initialwert am Ausgang Output nach dem Reset soll „00000“ betragen. Wenn der Eingang SyncLoadInput auf '1' gesetzt ist, dann soll der Ausgang Output bei der steigenden Flanke des Taktes auf den Wert des Eingangsvektors Input gesetzt werden. Wenn der Eingang AsyncLoadConstant auf '1' gesetzt ist, dann soll der Ausgang Output sofort auf den konstanten Wert „01101“ gesetzt werden. Wenn sich der Ausgangsvektor von „11111“ auf „00000“ verändert, dann soll das Overflow Signal bis zur nächsten steigenden Flanke des Takts auf '1' gesetzt werden. In allen anderen Fällen soll das Overflow Signal auf '0' gesetzt sein.

Programmieren Sie dieses Verhalten in der angehängten Datei „counter\_beh.vhdl“.

Um Ihre Lösung abzugeben, senden Sie ein E-Mail mit dem Betreff „Result Task 3“ und Ihrer Datei „counter\_beh.vhdl“ an [vhdl-mc+e384@tuwien.ac.at](mailto:vhdl-mc+e384@tuwien.ac.at).

Viel Erfolg und möge die Macht mit Ihnen sein.