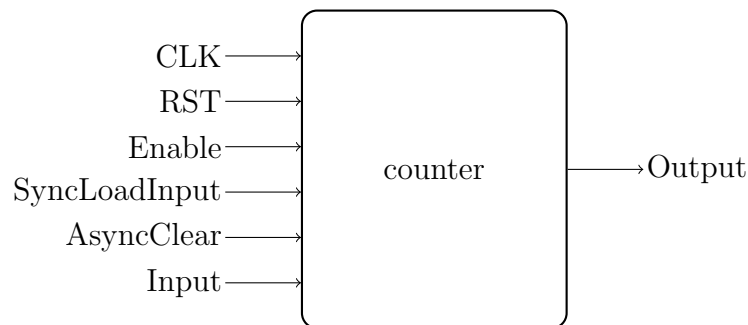


# Counter

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: Enable vom Typ std\_logic
- Eingang: SyncLoadInput vom Typ std\_logic
- Eingang: AsyncClear vom Typ std\_logic
- Eingang: Input vom Typ std\_logic\_vector und der Länge 6
- Ausgang: Output vom Typ std\_logic\_vector und der Länge 6



Verändern Sie die Datei „counter.vhdl“ nicht!

Die „counter“ Entity soll den Ausgangsvektor Output bei einer steigenden Flanke des Taktsignals CLK inkrementieren. Der Eingang RST ist ein synchroner Reset, der Initialwert am Ausgang Output nach dem Reset soll „000000“ 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 AsyncClear auf '1' gesetzt ist, dann soll der Ausgang Output sofort auf „000000“ gesetzt werden. Wenn das Enable Signal auf '1' gesetzt ist, dann soll sich die Entity „counter“ wie oben beschrieben verhalten. Wenn das Enable Signal aber auf '0' gesetzt ist, dann soll die Entity „counter“ auf keinen Eingang reagieren und den aktuellen Ausgangsvektor unverändert lassen.

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.