

Entwicklung und Implementierung eines digitalen Funktionsgenerators in VHDL

Markus Hartlage

FH-Bielefeld

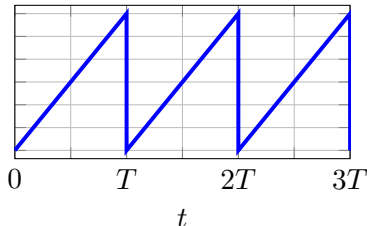
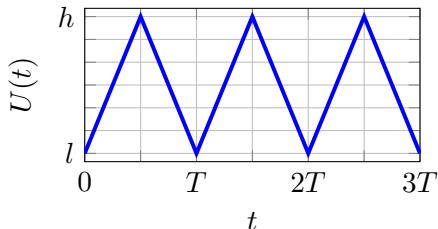
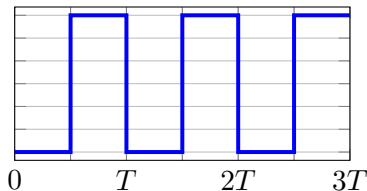
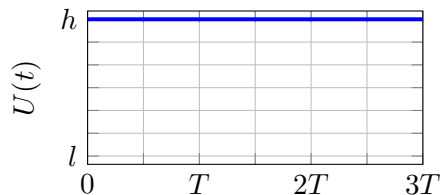
April 1, 2022



FH Bielefeld
University of
Applied Sciences

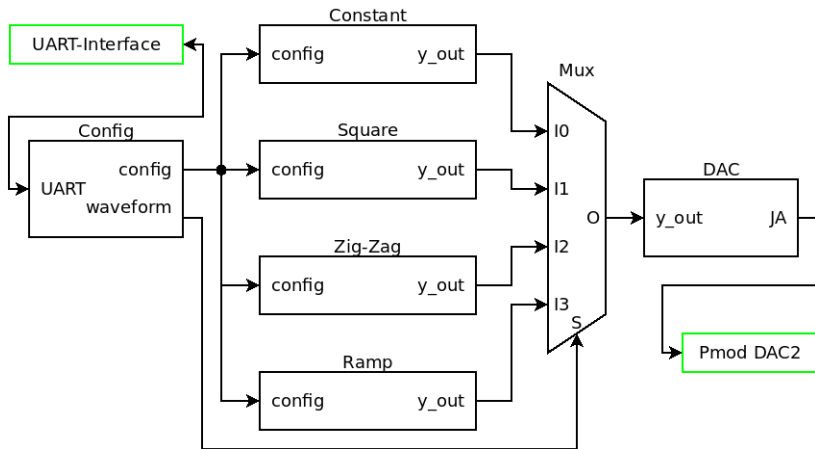
Konzept - Anforderungen

- ▶ Ausgabe vier verschiedener Funktionsverläufe
 - ▶ Konstante, Rechteck, Zick-Zack, Rampe
- ▶ Konfiguration per UART-Schnittstelle



Konzept - Aufbau

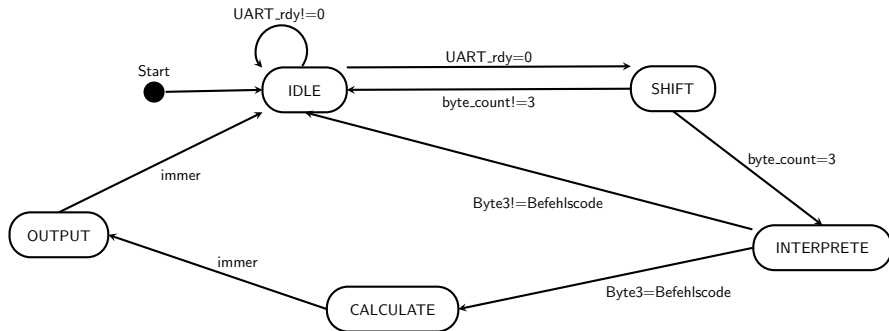
- ▶ Aufbau aus Konfigurations-, Funktions- und DAC Komponente
- ▶ zusätzliche Hardware: Uart-Interface und digital-analog Konverter



Komponenten - Konfiguration

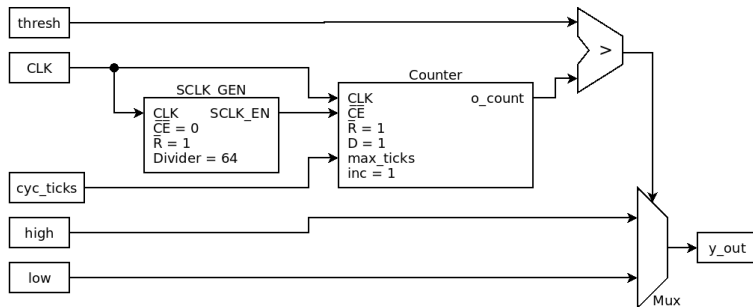
Aufbau als state machine:

- ▶ Byteweises Einlesen der UART Rx Signale
- ▶ Interpretation von vier Bytes als Befehl (Befehlscode + Argumente)
- ▶ Berechnung und Speicherung der neuen Konfiguration



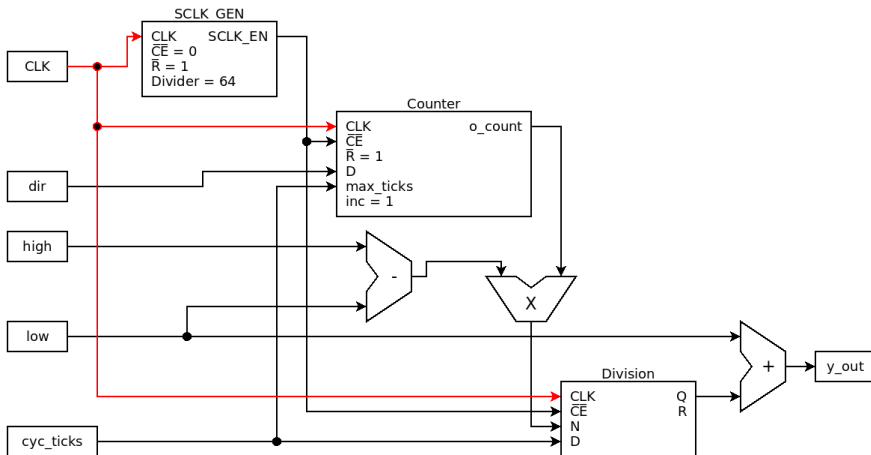
Komponenten - Square

- ▶ nach 64 steigenden Flanken *Counter* aktivieren
- ▶ Ausgang auf *low* wenn $o_count > thresh$, sonst *high*



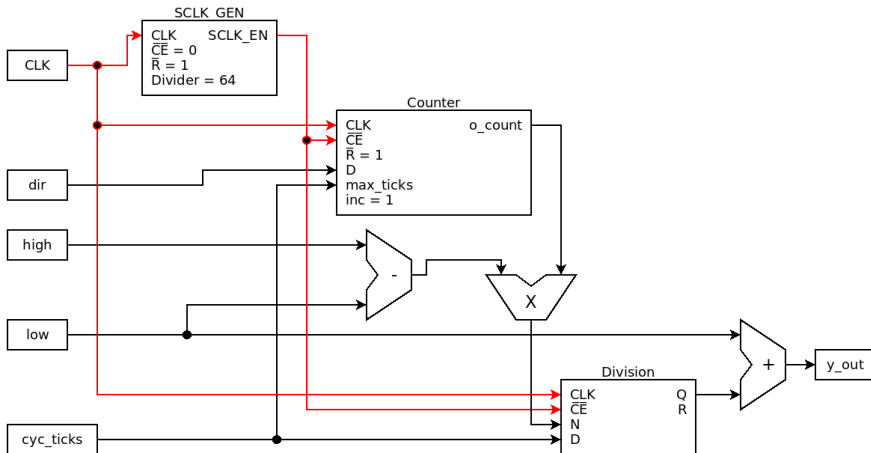
Komponenten - Rampe

► Abzählen von 64 Taktzyklen



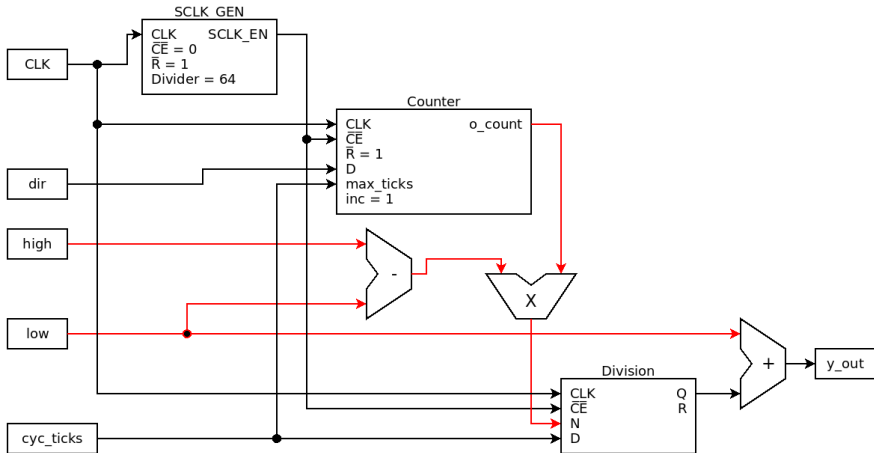
Komponenten - Rampe

► Aktivieren des Zählers



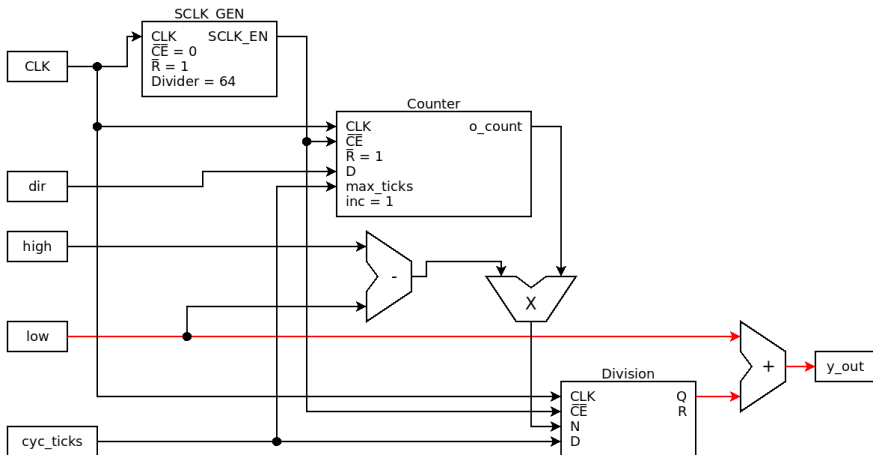
Komponenten - Rampe

- ▶ neuer Zählstand wird mit Amplitude multipliziert und *Division* beginnt zu teilen



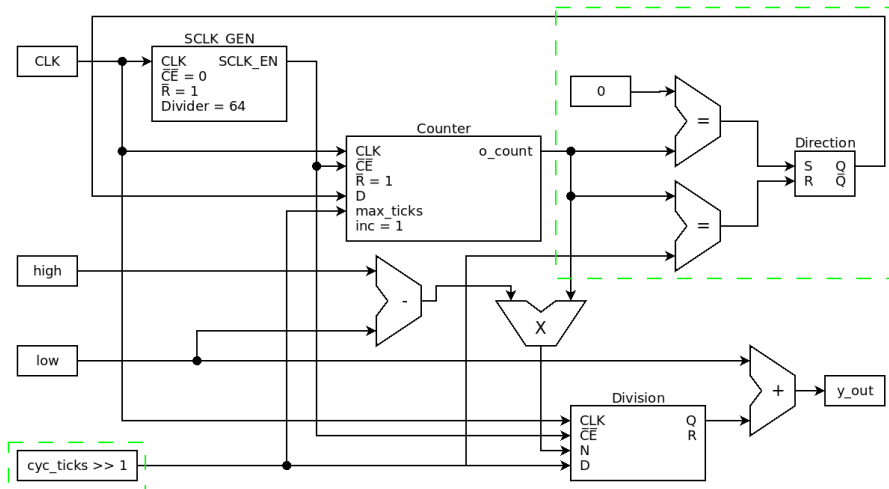
Komponenten - Rampe

- ▶ untere zwölf Bits von Q plus low ergeben y_{out}



Komponenten - Zick-Zack

- ▶ Zählrichtung wird mit Komparatoren geregelt
- ▶ halbe Zykluszeit (*cyc_ticks* um ein Bit nach rechts geschoben)



Komponenten - Funktionsbausteine

Sources