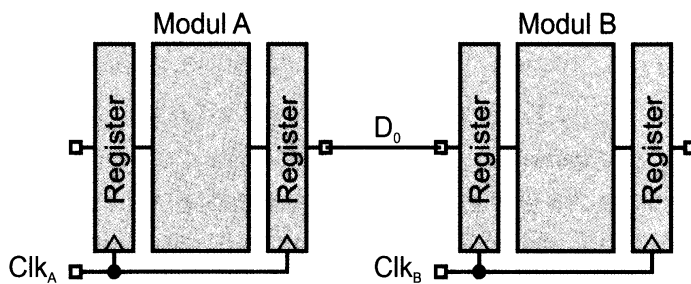


4-6 VHDL-Interfacing

1 Synchroner Entwurf und Taktverteilung

1.1 Synchrone und asynchrone digitale Systeme

Das Problem von digitalen Systemen mit mehreren Modulen lässt sich auf das einfache System entsprechend der Skizze reduzieren.

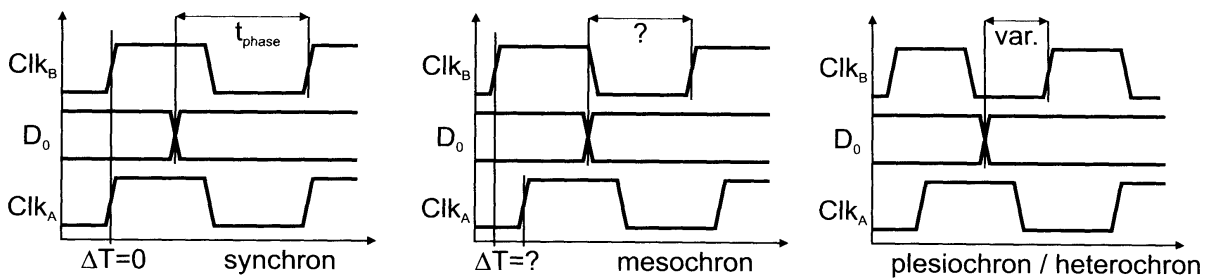


Das Signal D₀ wird von dem mit dem Takt Clk_A getakteten Modul A erzeugt und in das mit einem Takt Clk_B getakteten Modul B übertragen.

Ein Signal ist **isochron**, wenn die Änderungszeitpunkte in einem festen Zeitraster liegen (zB ist jedes periodische Signal isochron), sonst ist es **anisochron**.

Gibt es geringe Schwankungen der Periodendauer, so wird dies **Jitter** genannt.

Die zwei Takte Clk_A und Clk_B können zueinander drei Beziehungen haben:



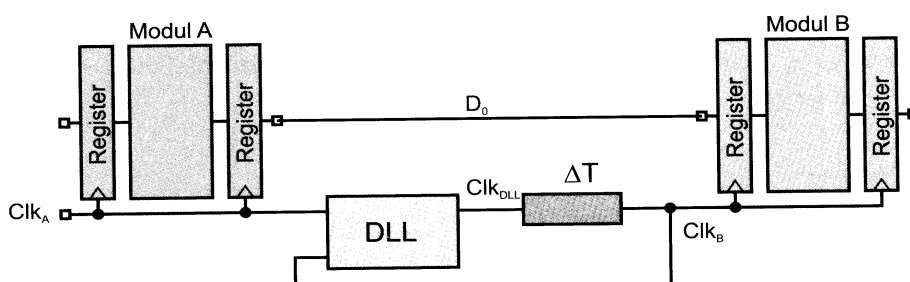
- **synchron**: (griech: zeitgleich) wenn ihre Änderungszeitpunkte übereinstimmen oder wenn sie eine bekannte und konstante Zeitverschiebung aufweisen (**asynchron**: wenn obige Bedingung nicht erfüllt ist)

Wenn Takt Clk_A und Clk_B synchron zueinander sind, so müssen beide exakt die gleiche Frequenz und eine unveränderliche, feste Phasenverschiebung (Zeitdifferenz ΔT) haben. Damit ist erlaubt, dass das Signal direkt durch das FF im Modul B abgetastet wird. Wichtig ist, dass sichergestellt ist, dass die Änderungszeitpunkte des Signals nicht in das Entscheidungsfenster des FFs im Modul B fallen (Gefahr von Metastabilität)

In der Realität gibt es immer einen geringen Taktversatz $\Delta T \neq 0$, wenn dieser konstant ist, spricht man trotzdem von einem synchronen System.

- **mesochron**: (griech: zeitlich gemittelt) wenn die durchschnittlichen Bitraten gleich sind.

Dieser Fall tritt ein, wenn die Takte gleiche Frequenz aber eine unbekannte und beliebige Phasenlage (ΔT) zueinander haben.



Das Signal D₀ darf nicht ohne zusätzliche Maßnahmen mit dem Takt Clk_B abgetastet werden. Zur Justierung der Phasenlage gibt es in FPGAs sog. DLL (delay locked loop). Die DLL vergleicht den Clk_B mit einem Referenztakt (hier Clk_A) und justiert eine interne Verzögerungsstrecke so, dass

die Gesamtphasenverschiebung zwischen den beiden Takten immer $\Delta T = T$ (und damit $= 0$) wird.

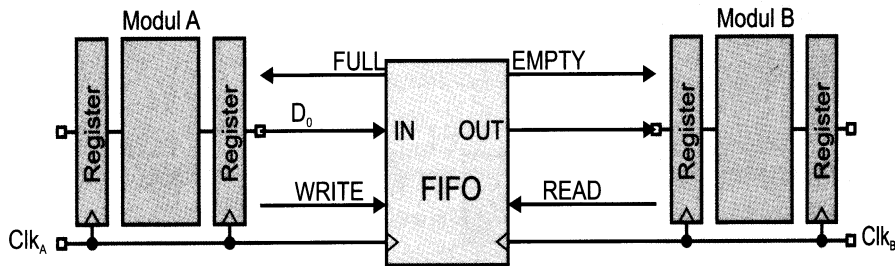
Zur Synchronisierung könnte auch eine PLL (phase locked loop) verwendet werden. Dabei wird ein VCO

(voltage controlled oscillator) von einem Phasenvergleicher so angesteuert, dass der von ihm erzeugte Takt phasenstarr zum Referenztakt ist. Durch Frequenzteiler lassen sich auch gebrochen rationale Taktfrequenzverhältnisse erreichen.

- **plesiochron**: (griech: nahezu) wenn die Bitraten nominell gleich sind.

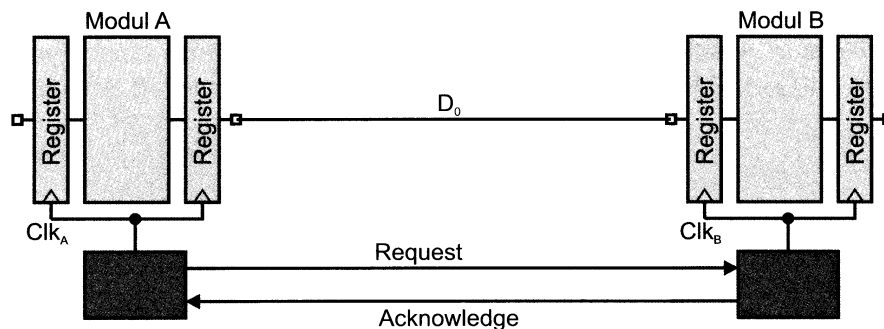
Dieser Fall entspricht der Situation, dass die beiden Takte ClkA und ClkB von zwei Taktoszillatoren mit nominell gleicher Frequenz erzeugt werden (die faktisch einen kleinen Frequenzunterschied aufweisen). Durch den Frequenzunterschied variiert die Phasenlage der beiden Takte zueinander ständig.

Ist auch die nominelle Bitrate nicht gleich, so spricht man von **heterochromen** Signalen.



Zur Synchronisierung bei plesiochronen Signalen werden meist FIFOs verwendet. Über zusätzliche Verriegelungssignale (EMPTY, FULL) muss das Schreib-Leseverhalten geregelt werden (Modul A sendet schneller, als Modul B lesen kann → FIFO wird voll; Modul A sendet langsamer als Modul B lesen will: FIFO wird leer.)

Es gibt auch Systeme ohne Takte: heißen ebenfalls **asynchrone** Systeme



Es gibt auch eine Mischung aus synchron und asynchron: GALS (globally asynchronous locally synchronous) wo mehrere intern jeweils synchrone Module über asynchrone Signale (+ Handshakeleitungen) miteinander kommunizieren.

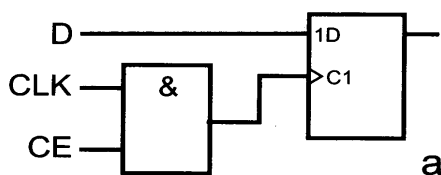
Zusammenfassung der möglichen Taktbeziehungen:

Bezeichnung		Phasendifferenz $\Delta T/T$	Frequenzdifferenz Δf
synchron		const, bekannt	gleich 0
asynchron	mesochron	unbekannt	gleich 0
asynchron	plesiochron	nicht const	ungleich 0, gering
asynchron	heterochron	nicht const	ungleich 0, groß

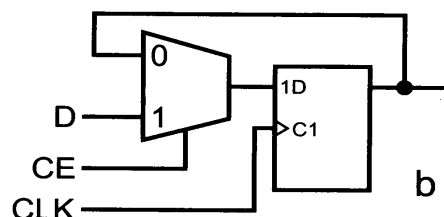
1.2 Clock Gating

nochmals zu Wiederholung: Clock Gating ist in einer synchronen Schaltung **verboten**

NEIN, nicht so



sondern so !



1.3 Digital Clock Management

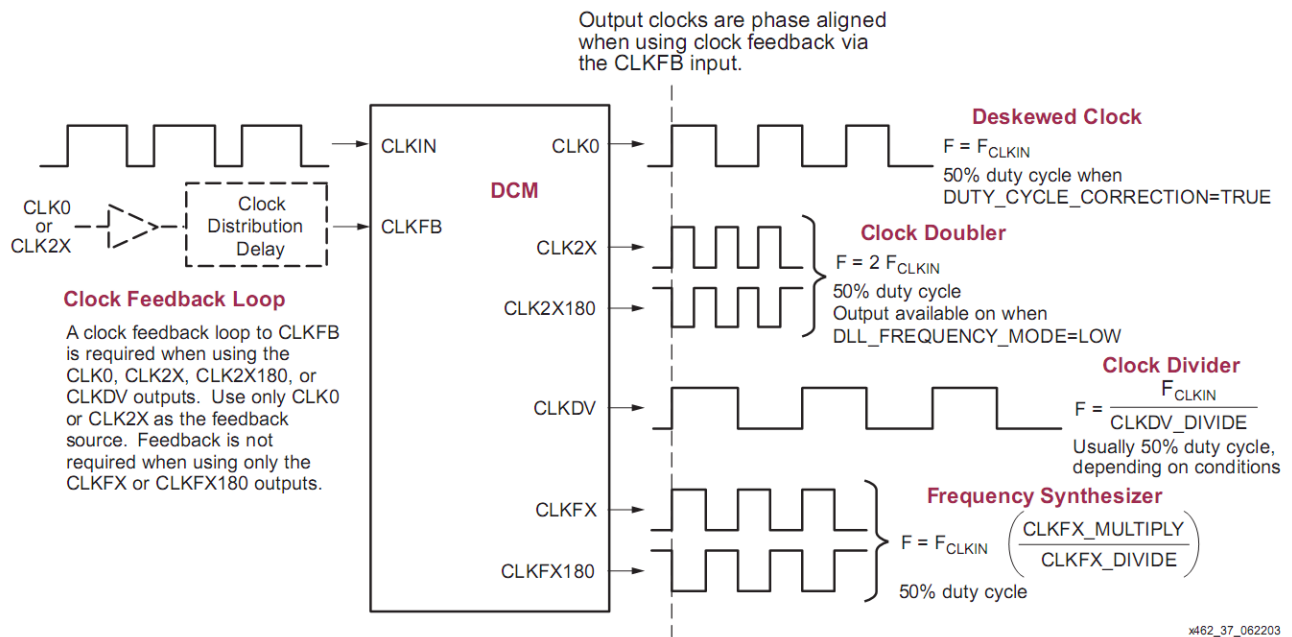
Quelle: Spartan 3E FPGA Family Data Sheet

DCM (Digital Clock Manager)

können in einem FPGA für verschiedene Zwecke eingesetzt werden (Quelle: XAPP462.pdf Xilinx):

- Synchronisation zweier Taktsignale indem Verzögerungszeiten aufgehoben werden (Deskewing)
- Vervielfachung der Frequenz von Taktsignalen
- Frequenzsynthese von Taktsignalen

Das Bild zeigt die Anschlussmöglichkeiten des DCM in einem Spartan III FPGA (Xilinx)



Taktteilung (CLKDIV)

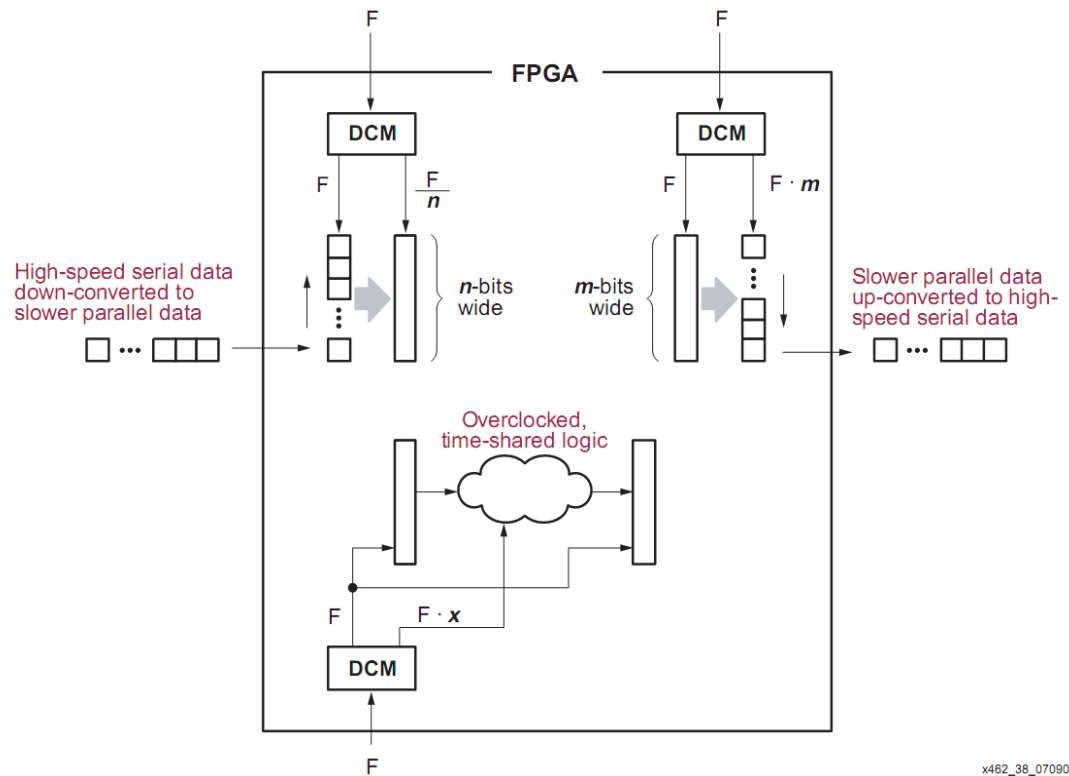
CLKDV_DIVIDE	Selects the constant used to divide the CLKIN input frequency to generate the CLKDV output frequency	1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6.0, 6.5, 7.0, 7.5, 8, 9, 10, 11, 12, 13, 14, 15, and 16
--------------	--	---

Taktfrequenzteilung und -vervielfachung

In einem FPGA kann es sinnvoll sein, mehrere Taktdomänen mit ganzzahligen Frequenzverhältnissen zu verwenden:

- Übergang von seriellen Schnittstellen zu interner paralleler Signalverarbeitung ($f \rightarrow f / n$)
- Übergang von interner paralleler Signalverarbeitung zu seriellen Schnittstellen ($f \rightarrow f * n$)
- Logik, die als HW nur einmal vorhanden ist, aber im Zeitmultiplex mehrfach verwendet wird ($f \rightarrow f * x$)

Beispiel für mehrere Taktdomänen



2 Ein-Ausgabe

2.1 Logikfamilien – Kompatibilität

Bei Vcco Spannungsversorgung mit 3.3V sind sowohl Ein- als auch Ausgänge mit LVTTTL (3.3V) und LVCMOS3.3 kompatibel; die LVCMOS Familien mit kleinerer Versorgungsspannung sind nur inputkompatibel. Die Treiberstärken sind zwischen 2 mA und 16 mA einstellbar (DRIVE Constraint)

2.2 Constraints

Quelle: „Constraints Guide“ zur aktuellen Webpack Version (cgd.pdf)

Achtung: Bezeichner in .UCF Files sind case-sensitive, dh Bezeichner müssen im UCF-File GENAU SO geschrieben werden wie in den Designfiles (obwohl VHDL nicht case sensitive ist).

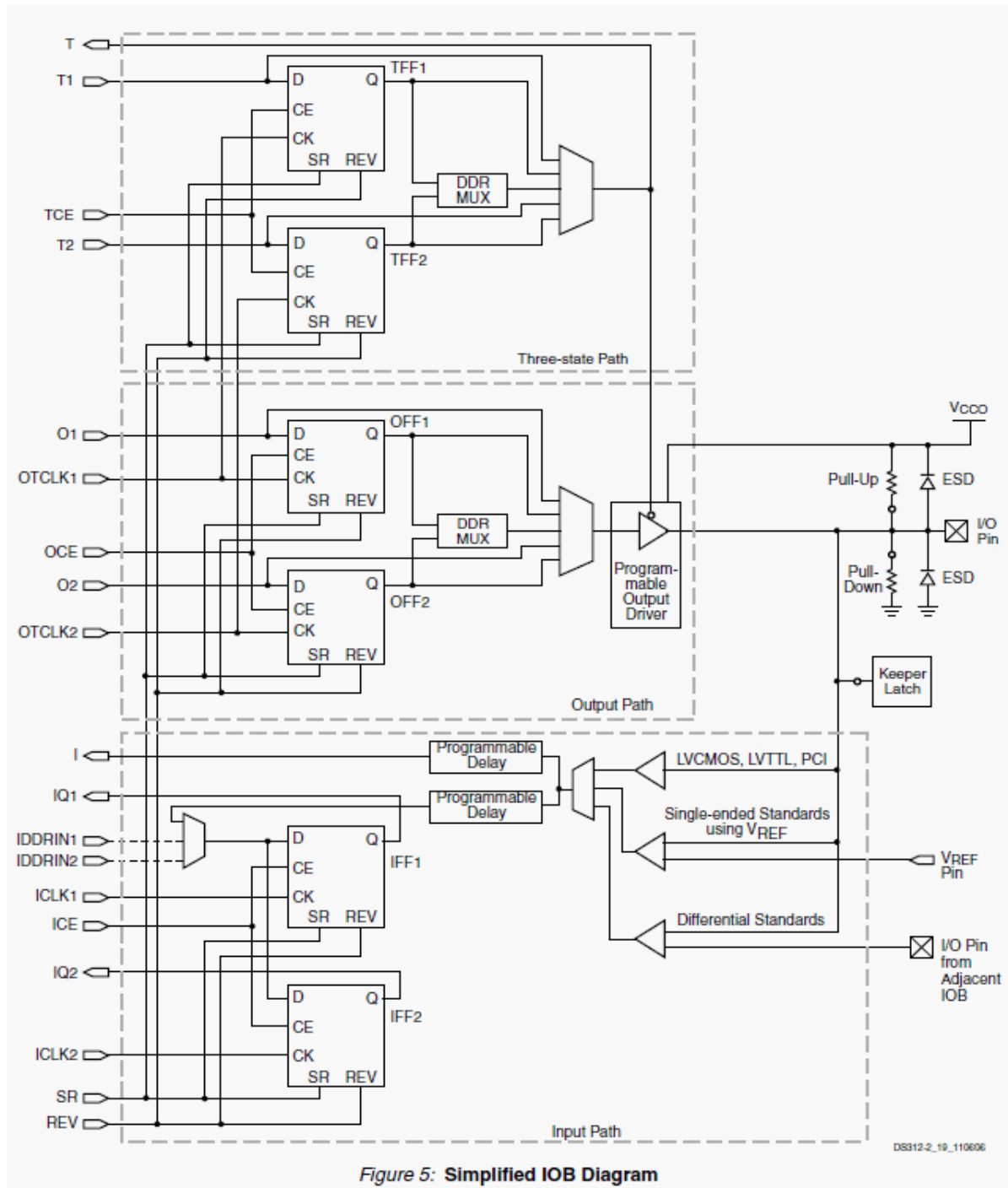
Wichtigste Constraints:

- **PULLUP / PULLDOWN:**
- **LOC**
- **DRIVE**
- **TIMESPEC**
- **SLEW**

NET "SDA" LOC = "J3" | DRIVE={2|4|6| 8|12|16 |24}; #12 mA is the default

Defines the slew rate (rate of transition) behavior of each individual output to the device

2.3 IO-Block bei Spartan 3E

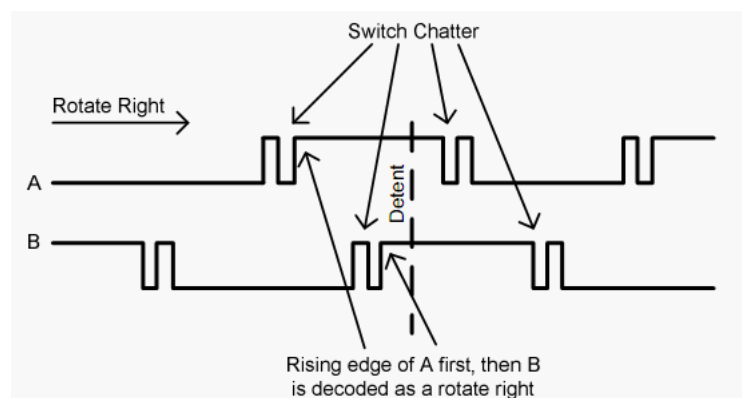


3 Interfacing

3.1 Buttons

Entprellen (Debouncing)

Entitiy: entpreller.vhd



3.2 Rotary encoder

Interface_Entity: EncoderFFF

File: EncoderFFF.vhd

Demoprojekt: dac

Beschreibung: Up-down-counter 0 – fff mit Überlauf

Interface_Entity: EncoderFF

File: EncoderFF.vhd

Demoprojekt: tonerzeugung (CPLD, 8 Mhz Takt)

Beschreibung: Up-down-counter 0 – fff mit Überlauf

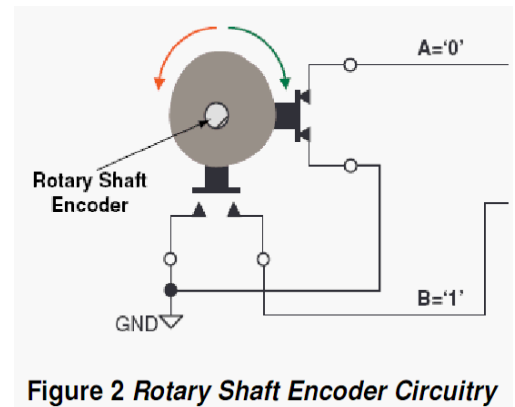


Figure 2 Rotary Shaft Encoder Circuitry

3.3 2-Kanal DAC mit Microwire Interface

Interface_Entity: DA2RefComp

File: DA2RefComp.vhdl

Demoprojekt: dac

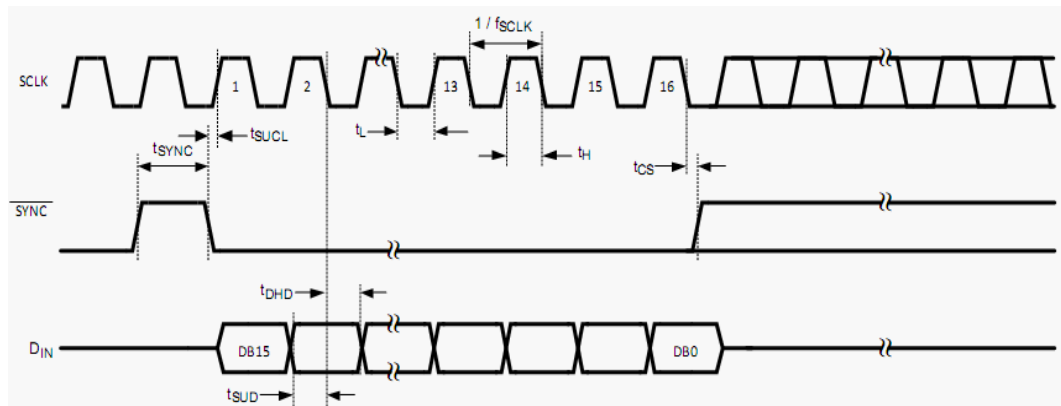


Figure 2. DAC121S101 Timing

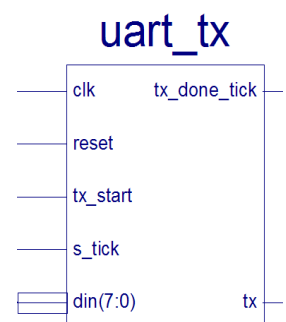
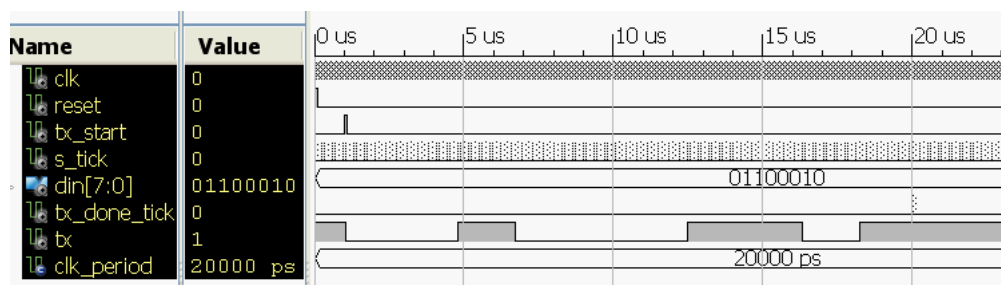
3.4 UART

Interface_Entity: uart_tx

File: uart_tx.vhd (118 Z.)

Demoprojekt: Klima

usb_uart



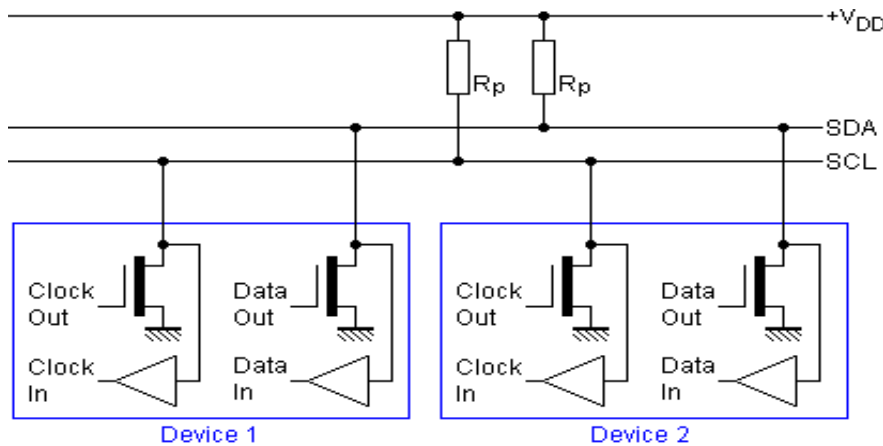
3.5 I2C

Interface_Entity: i2c

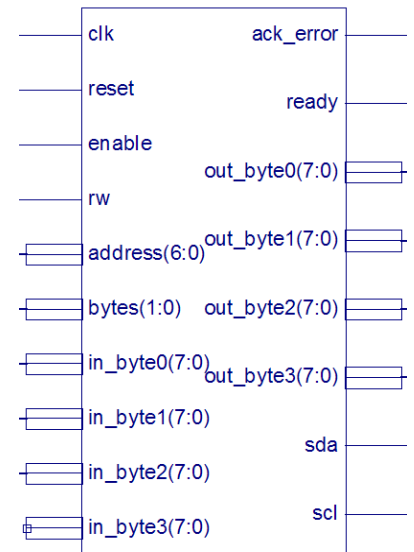
File: i2c_master.vhd (395 Z.)

Demoprojekt: Klima

Beschreibung: bis zu 4 Byte rx und tx; Takterzeugung intern



i2c



Inhaltsverzeichnis

4-6	VHDL-Interfacing.....	2
1	Synchroner Entwurf und Taktverteilung.....	2
1.1	Synchrone und asynchrone digitale Systeme.....	2
1.2	Clock Gating.....	3
1.3	Digital Clock Management.....	4
2	Ein-Ausgabe.....	5
2.1	Logikfamilien – Kompatibilität.....	5
2.2	Constraints.....	5
2.3	IO-Block bei Spartan 3E.....	6
3	Interfacing.....	7
3.1	Buttons.....	7
3.2	Rotary encoder.....	7
3.3	2-Kanal DAC mit Microwire Interface.....	7
3.4	UART.....	8
3.5	I2C.....	8