



Beamer mit AMBA

Inhaltsverzeichnis

1	Einführung	1
2	Start	1
3	Ziele	2
3.0.1	Todo	2
3.1	Ansteuerung von Zeichnungspunktspeichern	2
3.1.1	Todo	3
3.2	Geschwindigkeit des Zeichnens	3
3.2.1	Todo	3
3.3	Testmodus des Interpolationsrechners	3
3.3.1	Todo	3

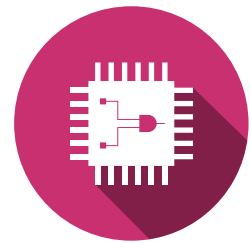
1 Einführung

Dieses Labor greift die Interpolationsschaltung auf und das Ziel ist es, sie an den AMBA-Bus anzubinden. Die verwendete Schaltung zeigt eine Möglichkeit, eine generische Busschnittstelle zu realisieren, die sich leicht an andere Systeme als den AMBA-Lite-Bus anpassen lässt, der in dieser Laborreihe als Beispiel herangezogen wurde. Die Schaltung könnte leicht an eine andere Art von parallelem Bus und sogar an einen seriellen Bus angepasst werden.

2 Start

Die Schaltung befindet sich in der Bibliothek **AhbLiteComponents**. Der Prüfstand in der Bibliothek **AhbLiteComponents_test**.

*Erinnerung: Das Modellierungsprogramm muss über die Datei **systemOnChip.bat** gestartet werden.*



3 Ziele

Das Gerät bietet zwei modulierte Ausgänge $\Sigma\Delta$, mit denen eine Kurve auf einem Oszilloskop gezeichnet werden kann.

Die Register, auf die der Mikroprozessor schreibend zugreift, sind:

- Adresse 00: Steuerregister
 - Bit 0: **run**, bestimmt, ob eine Kurve gezeichnet wird oder ob das System angehalten wird.
 - Bit 1: **updatePattern**, definiert, ob neue Punkte in die Speicher geladen werden oder ob man sich im Zeichenmodus befindet.
 - Bit 2: **interpolateLinear**, legt fest, ob die Verbindungslinie zwischen zwei aufeinanderfolgenden Punkten eine Gerade oder ein Spline ist.
 - Bits mit der höchsten Wertigkeit (MSB): **patternSize**, linksbündig, gibt die Anzahl der Punkte auf der zu zeichnenden Kurve an.
- Adresse 01: Geschwindigkeitsregister, gibt die Zeitspanne zwischen der Berechnung von zwei aufeinanderfolgenden Punkten der Zeichnung an.
- Adresse 02: Register der Punkte der X-Kurve
- Adresse 03: Register der Punkte der Y-Kurve

Die Register, auf die lesend zugegriffen wird, sind:

- Adresse 00: Befehlsregister, um die zuvor geschriebenen Werte erneut zu lesen
- Adresse 01: Geschwindigkeitsregister, ermöglicht es, den zuvor geschriebenen Wert erneut zu lesen

3.0.1 Todo

- Schreibe den VHDL-Code des Adressdecoders **periphAddressDecoder**.
- Schreiben Sie den VHDL-Code für die Steuer- und Geschwindigkeitsregister des Geräts **ahbBeamer**.

3.1 Ansteuerung von Zeichnungspunktspeichern

Die Speicher, die die Liste der zu zeichnenden Punkte enthalten, werden in speziellen Speicherblöcken innerhalb der programmierbaren Schaltung verwendet. Die Steuerung der Speicher erfolgt in zwei verschiedenen Modi:

- im Modus **updatePattern** löst jeder neue Schreibzugriff auf das betreffende Register einen Schreibvorgang im Speicher und eine Inkrementierung des Adresszählers aus
- im Kurvenzeichenmodus bewirkt jede neue Anfrage nach einem Punkt durch das Signal **newPolynom**, dass der Adresszähler inkrementiert wird

Wenn das Signal **updatePattern** auf '1' wechselt, wird der Adresszähler auf Null zurückgesetzt.

Im Kurvenzeichnungsmodus wird der Adresszähler zurückgesetzt, wenn der Zähler den Wert **patternLength-1** erreicht. Dies ist im Modus **updatePattern** nicht der Fall.



3.1.1 Todo

- Schreiben Sie den VHDL-Code für die Steuerblöcke der Speicher für die Punkte der X- und Y-Kurven des Geräts **ahbBeamer**.

3.2 Geschwindigkeit des Zeichnens

Die Geschwindigkeitsregistrierung ermöglicht es dem Gerät, mit einer Geschwindigkeit zu arbeiten, die langsamer ist als die Taktfrequenz.

3.2.1 Todo

- Schreiben Sie den VHDL-Code für den Zähler, der die Impulse **interpolateEn** im **periph-SpeedController**-Block erzeugt.

3.3 Testmodus des Interpolationsrechners

Mit der Schaltfläche **selSinCos** können Sie einen Kreis im X-Y-Modus zeichnen, unabhängig von einem Wert, der in ein Register des Geräts geschrieben wurde.

3.3.1 Todo

- Schreiben Sie den VHDL-Code für die Tabelle, die die Sinus- und Cosinuswerte erzeugt, in den Block **sinCosTable**.
- Simulieren Sie mithilfe der zur Verfügung stehenden Testbank das System und überprüfen Sie, ob das Gerät **ahbBeamer** richtig funktioniert.