Aufgabe: Analog-Digitalwandler

G. Kemnitz*, TU Clausthal, Institut für Informatik

19. Juli 2007

Zusammenfassung

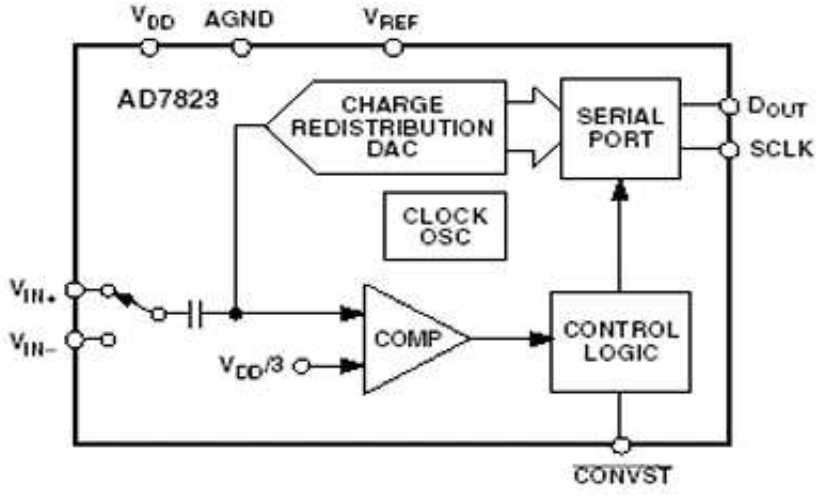
Der ADC (Analog to Digital Converter) AD7823 auf dem angesteckten Zusatzboard AIO1 ist in der Lage, Spannungen an seinen analogen Eingängen in einen digitalen Wert umzuwandeln und über eine synchrone serielle Schnittstelle an einen Rechner oder an unseren zu programmierenden Schaltkreis auszugeben. In der Übung ist eine einfache Schaltung zur Kommunikation mit dem AD7823 vorgegeben. Das Protokoll, über das die Schaltung mit dem ADC kommuniziert, ist mit ChipScope zu untersuchen. Danach sind weitere Funktionen in die Schaltung einzubauen, wobei ChipScope die Plattform für Test und Fehlersuche bleibt.

1 AD7823

Der ADC wird hier nur so genau beschrieben, wie es für die Ansteuerung unbedingt erforderlich ist. Bei Problemen und Fragen bitte Datenblatt aus dem Netz holen und lesen.

Der AD7823 wandelt die Spannungsdifferenz $V_{in+} - V_{in-}$ in einen 8-Bit-Vektor um:

$$MW = \frac{V_{IN+} - V_{IN-}}{V_{REF}}$$



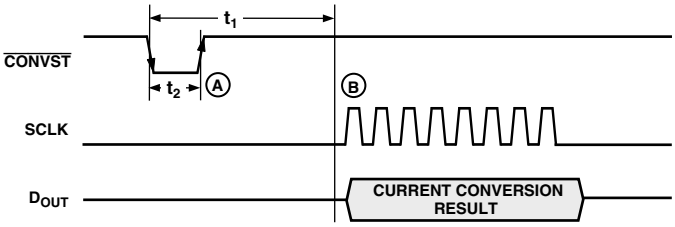
(V_{IN-}, V_{IN+}, V_{REF} - Spannungen zwischen dem gleichnamigen Eingängen und AGND (Masse). Im Versuchsaufbau sind $V_{REF} = 3{,}3\text{V}$, $V_{IN-} = 0\text{V}$ und V_{IN+} über ein kleines Potentiometer zwischen V_{IN-} und V_{REF} einstellbar.

Die Kommunikation zwischen programmierbarem Logikschaltkreis und ADC erfolgt über 3 Signale:

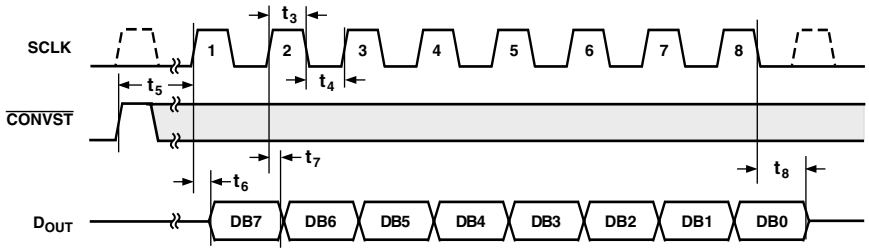
*Tel. 05323/727116

ADC Pin	ucf-Name	Bedeutung
$\overline{\text{CONVST}}$	AD_STRT	Steuersignal (vom FPGA)
SCLK	AD_SCK	Takt (vom FPGA)
D_{OUT}	AD_Dat	Daten (zum FPGA)

Die Umwandlung des analogen Wertes in einen Binärvektor beginnt mit der fallenden Flanke des Signals AD_STRT und ist garantiert nach $t_1 = 5\mu$s abgeschlossen. Anschließend kann das Ergebnis seriell ausgelesen werden.



Auslesen: Mit der steigenden Flanke von AD_STRT wird der Bitzähler des Schnittstellen-Controllers im DAC zurückgesetzt. Anschließend legt der Schnittstellen-Controller mit jeder steigenden Taktflanke beginnend mit dem höchstwertigen Bit ein neues Bit auf die Datenleitung. Mit jeder fallenden Taktflanke sind die Datenbits stabil und können von der selbst zu entwerfenden Schaltung übernommen werden. Die Taktfrequenz von AD_SCK darf maximal 20 MHz betragen. Die Übertragung dauert folglich mindestens $50\text{ns} \cdot 8\,\text{Takte} = 400\text{ns}$ und damit wesentlich weniger Zeit als die Wandlung.



Startet die serielle Übertragung früher als 5 μs nach der letzten fallenden Flanke von AD_STRT, wird das vorletzte Wandlerergebnis aufgegeben, sonst das Ergebnis der letzten Wandlung.

Es soll eine Automat entworfen werden, der kontinuierlich eine neue Messung startet und den Messwert abholt. Dieser Automat muss zyklisch nachfolgende Aktivitäten ausführen. Alle Zeiten sind auf Vielfache von 100 ns aufgerundet:

Zeit in 100ns	Ausgabe	Übernahme
0	AD_Strt<='0'	
30	AD_Strt<='1'	
60	AD_SCK<='1'	
61	AD_SCK<='0'	DB(7)<=AD_SDat
62	AD_SCK<='1'	
63	AD_SCK<='0'	DB(6)<=AD_SDat
$2\,(7-n)+60$	AD_SCK<='1'	
$2\,(7-n)+61$	AD_SCK<='0'	DB(n)<=AD_SDat
74	AD_SCK<='1'	
75	AD_SCK<='0'	DB(0)<=AD_SDat

Der Zyklus Wandlung+Ausgabe dauert etwa 7,5μs.

In VHDL werden zyklische Abläufe durch autonome Automaten (Automaten mit einer Zustandsvariable und ohne Eingang) nachgebildet. Im nachfolgenden Programmausschnitt heißt die Zustandsvariable vZeit:

```
process(Takt10MHz)
 variable vZeit: integer range 0 to 75:=0;
begin
 if Takt10MHz'event and Takt10MHz='1' then
   vZeit:=vZeit+1;
   case vZeit is
     when 0  => AD_Strt<='0';
     when 30 => AD_Strt<='1';
     when 60|62|64|66|68|70|72|74 => AD_SCK<='1';
     when 61|63|65|67|89|71|73 =>
         AD_SCK<='0';
         DB<=DB(6 downto 0) & AD_SDat;
     when 75 =>
         AD_SCK<='0';
         DB<=DB(6 downto 0) & AD_SDat;
         vZeit:=0;
     when others => Null;
   end case;
 end if;
end Process;
```

Das Codefragment unterscheidet sich etwas von dem Musterentwurf, der aus dem Netzt heruntergeladen werden kann. In der Beispieldatei ADU.vhd wird ein etwas langsamerer Takt von (50 / 8) MHz verwendet. Der Automat benötigt dadurch weniger Zustände. Die empfangenen Daten werden zu Testzwecken auf Leuchtdioden ausgegeben. Für die Signale zum und vom ADU sind zusätzliche interne Signale eingeführt. Letzteres ist für die Arbeit mit ChipScope notwendig, weil der integrierten Logikanalysator (ILA) nicht direkt an Schaltkreisanschlüsse, sondern nur an interne Signale angeschlossen werden kann.

2 Experiment

Kopieren Sie die Datein ADU.npl, ADU.vhd und Praktikum.ucf in ein Verzeichnis H:\TGP\ADU und öffnen Sie das Projekt "ADU". Die Datei Datei.vhd enthält einen Taktvorteiler und einen Automaten mit der oben skizzierten Funktion, der zyklisch Spannungsmessungen startet und die Messwerte abholt.

Übersetzen Sie das Projekt. Kontrollieren Sie, bevor Sie programmieren, dass das IO1-Board an den B-Steckern der Hauptplatine (rechts) steckt und dass Sie die ucf-Datei für diese Aufgabe

aus dem Netz und keine ucf-Datei, in der die Einträge für das AIO1 noch fehlen, verwenden. Stecken Sie die handgelötete Potentiometerbaugruppe auf das IO1-Board (Betreuer fragen) und probieren Sie die Schaltung aus.

- Welche LED liefert das höchstwertige und welche das niederwertigste Bit des Messwertes?
- Wird der Messwert als positive Zahl (0 - kleinster Wert; x"ff" größter Wert) oder im 2er-Komplement (x"80" kleinster Wert und x"7f" größter Wert) dargestellt?

3 Untersuchung der Daten mit ChipScope

Bauen Sie mit dem ChipScop Core-Inserter den integrierten Logikanalysator ein:

- Add New Source; Typ: "ChipScope Definition ..."; Name: ADU_cs; zugehöriges Source File: ADU; Weiter; Fertig stellen
- Den "ChipScope Core Inserter" mit Doppel-Click auf die neue Quelle "ADU_cs.cdc starten und zweimal "next"
- Reiter "Triggerparameter": Triggerbreite auf 11 erhöhen, alles andere lassen
- Reiter "Capture Parameters": Data Depth auf 512 erhöhen, "Data same as Trigger" und alles andere lassen
- Reiter "Net Connections": über "Modify Connections" "CLOCK PORT CH0" mit "clkDiv_n0000<2> verbinden und "TRIGGER PORTS TRIG0 ..." mit

Port	Signalname	Bedeutung
CH0	ad_SCK_int	Takt für den ADC
CH1	ad_Strt_int	Startsignal für ADC
CH2	ad_SDat_IBUF	Daten vom ADC
CH3	DB<0>	Schieberegister für Datenübernahme
...	...	
CH10	DB<7>	

- "Return to Project Navigator"

Mit diesen Einstellungen zeichnet der integrierte Logikanalysator mit einem Takt von

$$50\text{MHz} \cdot 2^{-3} \approx 6\text{MHz}$$

den zeitlichen Verlauf der Steuer- und Datensignale auf dem seriellen Bus zum ADC sowie den Zustand des Schieberegisters, das die seriellen Daten übernimmt, an.

Um den ChipScope Analyzer" zu starten:

- Auswahl "ADU-behavioral" im "Source-Fenster"
- "Generate Programming File"
- Kontrolle, dass "iMPACT" geschlossen ist
- "Analyze Design Using ChipScope" starten

Wenn der "ChipScop Pro Analyzer" gestartet ist:

- mit rechter Mouse-Taste auf "JTAG-Chain" "Xilinx Parallel Cable" auswählen; "Auto Detect" und hoffen, dass 2 Schaltkreise gefunden werden
- mit rechter Mouse-Taste auf DEV0 (XCV300E); Configure; Select New File; "ADU.bit" auswählen
- Warten bis "Download" abgeschlossen ist

Jetzt erschein das "Trigger"- und das "Wavform"-Fenster

- Mit "File => Import" das zuvor angelegte ChipScope-Definitionsfile "ADU_cs.cdc" importieren
- Trigger-Wert "Value" auf XXX_XXXX_XX0X ändern (Aufzeichnungsbeginn ab Wandlungsstart).

Jetzt können Sie sich das serielle Protokoll des ADCs und die Funktion des Schieberegisters, das die Daten übernimmt, anschauen (Klick auf das grüne Dreieck).

- Notieren Sie (auf Papier) die zeitlichen Signalverläufe auf dem Bus zum ADC in Tabellenform und vergleichen Sie die gemessenen Werte mit der Protokollbeschreibung aus Abschnitt 1.

Takt	Zeit	ad_SCK_int	ad_Strt_int	ad_SDat_IBUF
0	0μs		10-Flanke	
...	...	...	...	...

- Wieviele Messungen führt die Schaltung pro Sekunde genau aus?
- Wo gibt es noch Zeitreserven, um schneller messen zu können und wie wäre der Automat zu verändern? Test Sie Ihre geänderten Automaten.

4 Ausgabe des Messwertes auf der 7-Segmentanzeige

Ändern Sie die Schaltung so, dass der Messwert als 2-stelliger Hexadezimalwert auf der 7-Segmentanzeige dargestellt wird. Die beiden ungenutzten Ziffern sollen nicht leuchten.

5 Zusatzaufgabe

Wandeln Sie den 8-stelligen hexadezimalen Messwert vor der Ausgabe in eine 3-stellige Dezimalzahl um. Vorschlag für den Algorithmus:

Schritt 1: Subtrahiere vom Messwert 200 dezimal. Ist die Differenz positiv, setze Ergebnisbit 9 gleich Eins und Rest gleich Differenz. Sonst setze Ergebnisbit 9 gleich 0 und Rest gleich Messwert.

Schritt n: Subtrahiere vom Messwert Subtr(m) entsprechend nachfolgender Tabelle. Ist die Differenz positiv, setze Ergebnisbit m gleich Eins und Rest gleich Differenz. Sonst setze Ergebnisbit m gleich 0 und lasse Rest unverändert.

n (Berechnungsschritt)	2	3	4	5	6
m (berechntes Ergebnisbit)	8	7	6	5	4
Subtr(m) Subtrahend als Dezimalwert	100	80	40	20	10

Schritt 7: Zuweisunge Ergebnis(3 downto 0)<= Rest(3 downto 0);

Diese 7 Schritte werden im einfachsten Fall in den Automaten, der den ADC kontrolliert, mit eingebaut. Bei einem Automaten dieser Größe ist es in jedem Fall zu empfehlen, mit dem eingebauten Logikanalysator (ila) oder dem Simulator an 2 bis 5 Beispielen die Abarbeitung Schritt für Schritt zu kontrollieren. Im Beispiel sind insbesondere die Signalverläufe für den Rest, die Differenz und das Ergebnis während dieser Tests interessant. Dazu muss die Triggerbreite des ILAs erhöht und die zusätzlich zu beobachtenden Signale angeschlossen werden (Doppel-Click auf ADU_cs und Vornahme der Änderungen).

Für eine Kontrolle/Fehlersuche mit dem Simulator sollten Rest, Differenz und Ergebnis als zusätzliche Ausgänge aus der Schaltung geführt werden (ENTITY temporär ändern). Spätestens wenn Sie dann eine Testbench-Waveform erzeugen wollen, stoßen Sie jedoch auf das Problem, dass sie die vom ADC zu liefernden Daten (das gesamte serielle Protokoll) per Hand vorgeben müssen. Das ist eine zeitaufwendige Arbeit, bei der sich auch Fehler einschleichen können. ChipScope zur Auswertung und ADC als Testsignalquelle ist bei dieser Aufgabe deutlich einfacher.

6 Aufräumen

- Über Menüpunkt "Project, Cleanup Project Files" automatisch generierte Design-Files löschen.
- Netzteil zur Spannungsversorgung aus der Steckdose ziehen.
- Modelsim und Projektnavigator beenden.