

Übung zur Einführung in die digitale Signalverarbeitung

Übungseinheit 1

1 Einführung in die Entwicklungsumgebung

- 1.1 Code Composer Studio
- 1.2 Komponenten eines ausführbaren Programms
- 1.3 Programmausführung
- 1.4 Plotten von Daten

2 Hardware

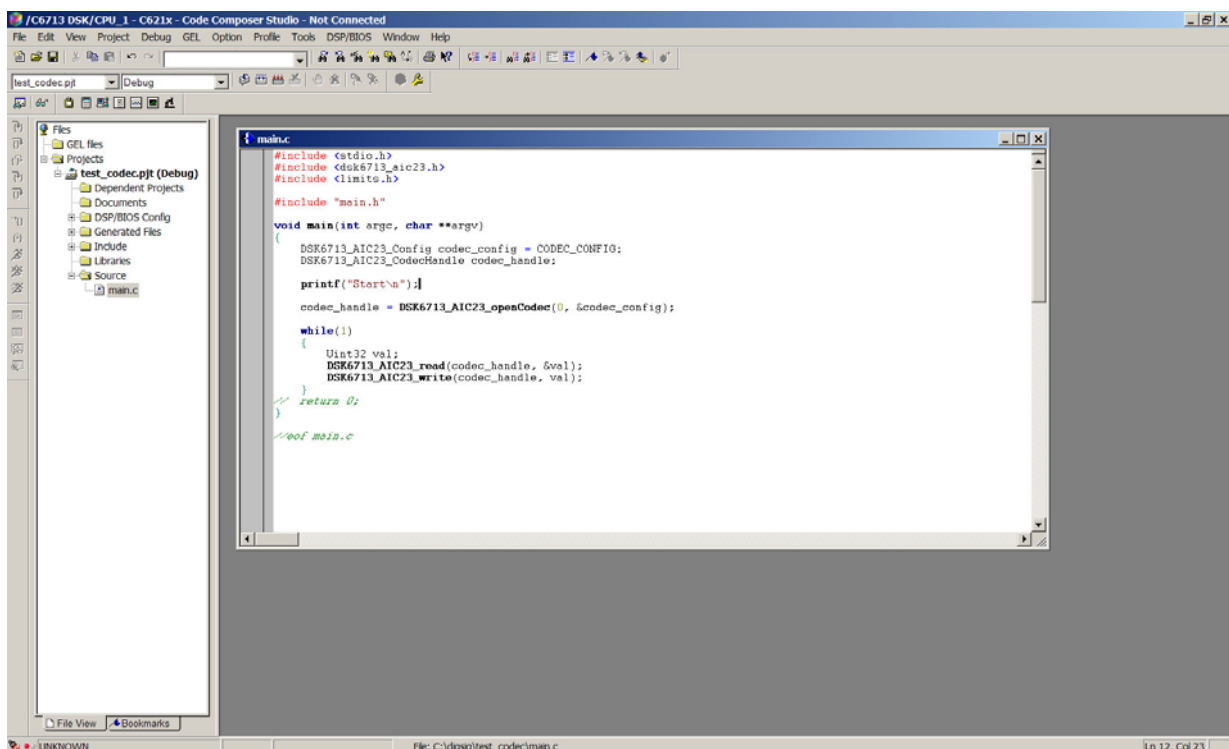
- 2.1 Architektur
- 2.2 Timer
- 2.3 Interrupts

3 Übungen

1 Einführung in die Entwicklungsumgebung

1.1 Code Composer Studio

Das Code Composer Studio (CCS) ist die Entwicklungsumgebung (IDE) für das hier verwendete TI DSK mit dem TMS320C6713 Prozessor. Es bietet Tools für die Codegenerierung wie beispielsweise Editor, Cross-Compiler und Linker für den DSP Prozessor. Das generierte Programm kann mithilfe des CCS auf das DSK Board übertragen und ausgeführt werden. Diese Entwicklungsumgebung stellt zudem Tools zum Debuggen zur Verfügung, wie beispielsweise das Setzen von Breakpoints, Einsehen von Variablen- und Speicherinhalten, graphisches Darstellen von Berechnungsergebnissen und Ausgabe der Ausführungszeit.



Wie bereits aus anderen Entwicklungsumgebungen bekannt sein dürfte, gibt es verschiedene Symbolleisten welche die Ausführung von Editor- sowie Programmierwerkzeugen ermöglichen.

Um ein ausführbares Programm zu erstellen, muss ein Projekt angelegt werden. Dies geschieht unter dem Menüpunkt Project → New. Anschließend können neue bzw. bestehende Dateien dem Projekt hinzugefügt werden.

Beim Start des Code Composer Studios sollte das Entwicklungsboard automatisch erkannt werden sofern es angeschlossen und eingeschaltet ist. Um die korrekte Funktionsweise des Boards zu testen kann unter dem Menüpunkt GEL → Check DSK → Quick Test ein Diagnosetool gestartet werden. Dabei wird unter anderem auch eine Zahl ausgegeben, die der Binärcodierung der Positionen der User Switches auf dem DSK Board entspricht.

1.2 Komponenten eines ausführbaren Programms

.c / .cpp / .h / .asm: Die entwickelten Code Dateien des Programms in C/C++ oder Assembler sowie zugehörige Headerdateien.

.cmd: Die Linker Command Datei enthält Anweisungen für den Linker und hauptsächlich Informationen, welche Sektionen des Programms in welchen Speicherbereich übertragen werden sollen.

.lib: Bibliotheken

.out: Ausführbare Datei die vom Linker erstellt wurde und auf das DSK Board geladen und ausgeführt werden kann.

.pj: Projektdatei

vectors.asm: Vektordatei der Interrupts. In dieser Assemblerdatei kann auf die Interrupt Service Routine (ISR) verwiesen werden, die bei einem Auftreten eines Interrupts ausgeführt werden soll.

.gel: General Extension Language. Mit dieser C ähnlichen Interpretersprache kann die Entwicklungsumgebung konfiguriert werden und die CPU initialisiert werden.

1.3 Programmausführung

Ein Programm kann mit der Option Project → Rebuild All erstellt werden. Sofern es keine Fehler bei der Erstellung gegeben hat, erstellt der Linker eine ausführbare .out Datei. Diese kann mit dem Befehl File → Load Program auf das Board geladen werden. Gegebenfalls muss vorher das Board mit dem Befehl Debug → Connect verbunden werden. Sobald das Programm auf das Board geladen wurde, kann es mit dem Befehl Debug → Run ausgeführt werden.

Das Menü Debug enthält zudem Befehle um Breakpoints im Programm zu setzen sowie das Programm zeilenweise auszuführen.

Für die einfache Ein- und Ausgabe können zunächst die Funktionen printf sowie scanf verwendet werden. Die Ausgabe erfolgt im stdout Fenster im unteren Bereich der Entwicklungsumgebung. Mit dem Befehl scanf wird das Programm auf dem Board angehalten, anschließend öffnet sich ein Fenster in der Entwicklungsumgebung mit der Aufforderung eine Eingabe zu tätigen um an das auf dem Board laufende Programm übertragen zu werden.

1.4 Plotten von Daten

Das Code Composer Studio bietet die Möglichkeit, Variablen bzw. Speicherbereiche des Prozessors anzeigen zu lassen. Dabei besteht zunächst die Möglichkeit Variablen zu verfolgen, sobald beispielsweise ein Breakpoint erreicht wird. Dazu wird der Cursor auf der zu verfolgenden Variable platziert und mit dem Menüpunkt View -> Quickwatch den zu beobachtenden Variablen hinzugefügt.

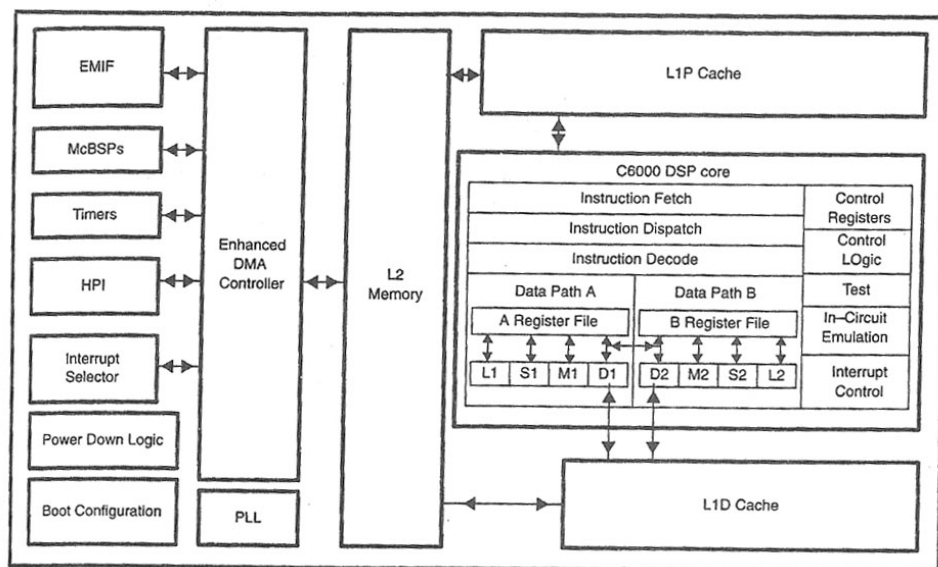
Außerdem besteht die Möglichkeit, eine Variable oder Array kontinuierlich mit einer bestimmten Abtastfrequenz zu beobachten und die Werte in einem Graphen darzustellen.

Dazu kann der Menüpunkt View -> Graph -> Time/Frequency benutzt werden. Im folgenden Menü kann dann das zu beobachtende Array sowie weitere Einstellungen vorgenommen werden. Das Signal kann dann im Zeit- oder im Frequenzbereich dargestellt werden.

2 Hardware

2.1 Architektur

In dieser Übung wird ein DSK (DSP Starter Kit) von Texas Instruments mit einem TMS320C6713 DSP Prozessor verwendet, der mit einer maximalen Geschwindigkeit von 225 MHz läuft. Dies ist ein floating-point Prozessor mit einer VLIW (very long instruction word) Architektur. Ähnlich wie bei der Harvard-Struktur gibt es dabei eine Aufteilung des Speichers in Programm- und Datenspeicher. Der Datenspeicher ist zudem in zwei Bänke aufgeteilt. Dies ermöglicht parallele Speicherzugriffe sofern die Daten in verschiedenen Bänken vorliegen.



Die Prozessoren der C6000er Serie bestehen aus zwei Datenpfaden um auf die zwei Datenspeicherbänke zuzugreifen. Dabei hat die CPU acht unabhängige funktionale Einheiten für Multiplikationen (.M), logische und arithmetische Operationen (.L), Verzweigungen, Bitmanipulationen und arithmetische Operationen (.S) und Datentransfers (.D).

Der Prozessor verfügt über eine zweistufige Cache Architektur, wobei der L1 Cache in einen jeweils vier KByte großen Programm- und Datencache unterteilt ist. Das Prozessorinterne RAM bzw. der L2 Cache hat eine Größe von 256 KByte. Weiterhin besteht eine direkte Anbindung zum externen, 16 MByte großen SDRAM Speicher. Dieser ist in den Speicherbereich des Prozessors eingebunden.

Der Prozessor besitzt verschiedenen Komponenten, darunter zwei Timer und zwei McBSP (multichannel buffered serial ports) Schnittstellen um den Audiocodec auf dem Board anzusprechen.

2.2 Timer

Der Prozessor besitzt zwei Timer Einheiten. Diese können unter anderem dazu verwendet werden, periodisch Interrupts auszulösen. Die Steuerung der Timer erfolgt dabei über drei Hardware-Register.

Im CTL (timer control register) Register können durch das Setzen von bestimmten Bits verschiedene Einstellungen wie beispielsweise Zählfrequenz vorgenommen, sowie der Timer gestartet werden.

Das PRD (timer period register) Register dient der Einstellung der Frequenz des Timers.

Das CNT (timer counter register) Register enthält den Wert um den der Zähler inkrementiert wird.

Die Timer können ziemlich bequem durch die im CCS vorhandenen Bibliotheken der CSL (chip support library) konfiguriert und angesprochen werden. Weitere Informationen über Funktionen zur Bedienung eines Timers finden sie in der Online Hilfe.

2.3 Interrupts

Ein Interrupt unterbricht den aktuellen Programmfluss und es wird eine ISR (interrupt service routine) abgearbeitet. Dabei werden die Register der CPU gesichert und nach der Ausführung der ISR wieder hergestellt.

Ein Interrupt kann vom Prozessor intern oder extern von einer Hardwarekomponente auf dem Board ausgelöst werden. Es gibt 16 Interruptquellen, davon zwei Timer-Interrupts, vier externe, vier McBSP und vier DMA Interrupts. Die Interrupts können mit verschiedenen Prioritäten für deren Ausführung versehen werden. Der Reset Interrupt hat die höchste Priorität, gefolgt von dem NMI (Non Maskable Interrupt). Wird der NMI deaktiviert, so werden zudem alle anderen Interrupts, außer dem Reset Interrupt deaktiviert.

Um Interrupts zu konfigurieren stehen verschiedene Interrupt - Steuerregister zur Verfügung. Dabei müssen folgende Schritte ausgeführt werden:

- Eintragen der ISR in der Datei vectors.asm
- Setzen des Verweises auf die Vektoren
- Aktivierung der Interrupts, im speziellen des NMI Interrupts
- Mappen eines Interrupts auf das Event einer Hardware-Komponente
- Reset sowie erneute Aktivierung des Interrupts

Diese Schritte können durch Setzen der entsprechenden Bits in den Steuerregistern vorgenommen werden, aber es gibt wiederum bequeme Funktionen in der CSL Bibliothek um die Konfiguration vorzunehmen. Informationen dazu finden sie in der Online Hilfe.

3 Übungen

- 1) Schließen sie das DSK Board an PC und Stromversorgung an, starten sie die Code Composer Studio Entwicklungsumgebung und stellen sie eine Verbindung mit dem Board her. Führen sie anschließend den Selbsttest aus um zu testen ob das Board ordnungsgemäß funktioniert.
- 2) Erstellen sie ein neues Projekt, schreiben sie ein einfaches „Hallo Welt“ Programm und führen sie es auf dem DSP Board aus. Erweitern sie anschließend ihr Programm um die user leds und dip switches auf dem Board anzusteuern. Inkludieren sie dazu die beiden Header-Dateien dsk6713_dip.h und dsk6713_led.h. Wird ein dip switch gedrückt so soll die entsprechende LED leuchten. Verwenden sie dazu eine Endlosschleife, in der die jeweiligen Zustände der switches abgefragt und entsprechend die LED's ein- oder ausgeschaltet werden.
- 3) Erstellen sie mithilfe von MATLAB eine Sinustabelle mit folgenden Eigenschaften
 - 1500 Hz
 - 128 Datenpunkte
 - Amplitude 1000Schreiben sie die Tabelle in eine für C verwendbare Headerdatei und erstellen sie die Tabelle dabei als short-Array (16 bit signed integer). Nützliche MATLAB Befehle sind dabei round, fopen, fprintf, fclose.
- 4) Erstellen sie ein neues Projekt. Verwenden sie ein Timer Modul um einen CPU Interrupt mit einer bestimmten Frequenz auszulösen. Beschreiben sie in diesem Interrupt einen 256 Elemente fassenden Buffer zirkulär mit den Werten aus der in Aufgabe 3 erstellten Sinustabelle. Lassen sie den Sinusgraphen im CCS plotten indem sie den Buffer anzeigen lassen. Kontrollieren sie außerdem die Frequenz des Sinussignals, indem sie auch den Frequenzbereich (FFT) des Signals plotten. Passen sie die Einstellungen an ihr Programm an, wie beispielsweise Variablenname, Buffergröße und Datentyp. Stellen sie zudem die Abtastfrequenz auf den Wert ein, den sie bei der Erzeugung in MATLAB verwendet haben.

Tipps:

- Vergessen sie nicht in Übungsaufgabe 2 die Suchpfade für die Header- und Bibliotheksdateien anzugeben. Benötigt wird neben den oben genannten Header-Dateien die Bibliothek dsk6713bsl.lib. Fügen sie dem Projekt weiters eine DSP/BIOS Konfiguration hinzu (Menüpunkt File -> New)
- Verwenden sie in Aufgabe 4 die CSL (chip support library) um den Timer und die Interrupts zu konfigurieren. Informationen zur Verwendung der CSL finden sie in den pdf Dateien C6000 API reference sowie C6000 periphery guide. Vergessen sie nicht die Funktion CSL_init vor der Verwendung der CSL Bibliothek auszuführen.
- Tragen sie in der von TI vorgegebene vectors.asm Datei ihren Interrupt ein.