Boot-Prozess UniDSP56

Gerrit Buhe, DL9GFA, V1.2

1. Einleitung

Dieses Dokument beschreibt die wichtigsten Möglichkeiten, den DSP des UniDSP56 mit Software zu laden und zu starten. Der DSP56309 besitzt im internen ROM einen sehr einfachen kleinen Bootloader (192 Worte), der unmittelbar nach dem Reset den Boot-Mode über die Pins MODA, MODB, MODC und MODD einliest. Da die selben Pins nach der Startphase als Interrupt-Eingänge verwendet werden, schaltet der Multiplexer IC2 den Boot-Mode-Schalter S2 nur während und kurz nach dem Reset an die MODx-Pins. Danach sind diese Eingänge an den Pfostensteckverbinder K9 gelegt. Die Funktionen des internen Boot-Loaders sind in Application Notes des Herstellers beschrieben und werden an dieser Stelle nicht weiter erläutert.

In den meisten Anwendungsfällen des UniDSP56 soll der DSP 8-Bit-breit aus dem Flash-EEPROM IC5 starten, so daß S2 immer in der Stellung „9“ verbleiben, oder ggf. sogar durch Brücken ersetzt werden kann. Im Auslieferzustand des Teilbausatzes ist der Flash-EEPROM bereits mit einem sogenannten Second Stage Bootloader programmiert der es erlaubt, das Board autark („stand alone“) zu betreiben, oder Software komfortabel über die RS232-Schnittstelle herunterzuladen und zu starten.

Im Folgenden wird der Ablauf für beide Startvorgänge beschrieben.

2. Second Stage Boot Loader

Die Bootstrap-Routine des DSP läd nach einem Reset den zweiten, komfortableren Boot-Loader aus dem Flash-EEPROM. Dieser erwartet an den höchstwertigen 4 Bit des Ports B (Host Port Interface, K6) einen Betriebsartenschalter (Hex-Schalter, 16 Zustände auf 4 Bit). Es können also 16 Betriebsarten unterschieden werden, wobei die Zustände „0“ und „F“ (bzw. „1111“, oder wenn kein Schalter installiert ist) dazu führen, daß der RS232-Loader im DSP gestartet wird, der einen PC oder Microcontroller an seiner seriellen Schnittstelle erwartet. Wird eine andere Zahl zwischen 1...14 eingelesen, soll selbständig das entsprechende Programm aus dem Flash-EEPROM in den DSP geladen und gestartet werden. Bild 1 zeigt den beschriebenen Ablauf.

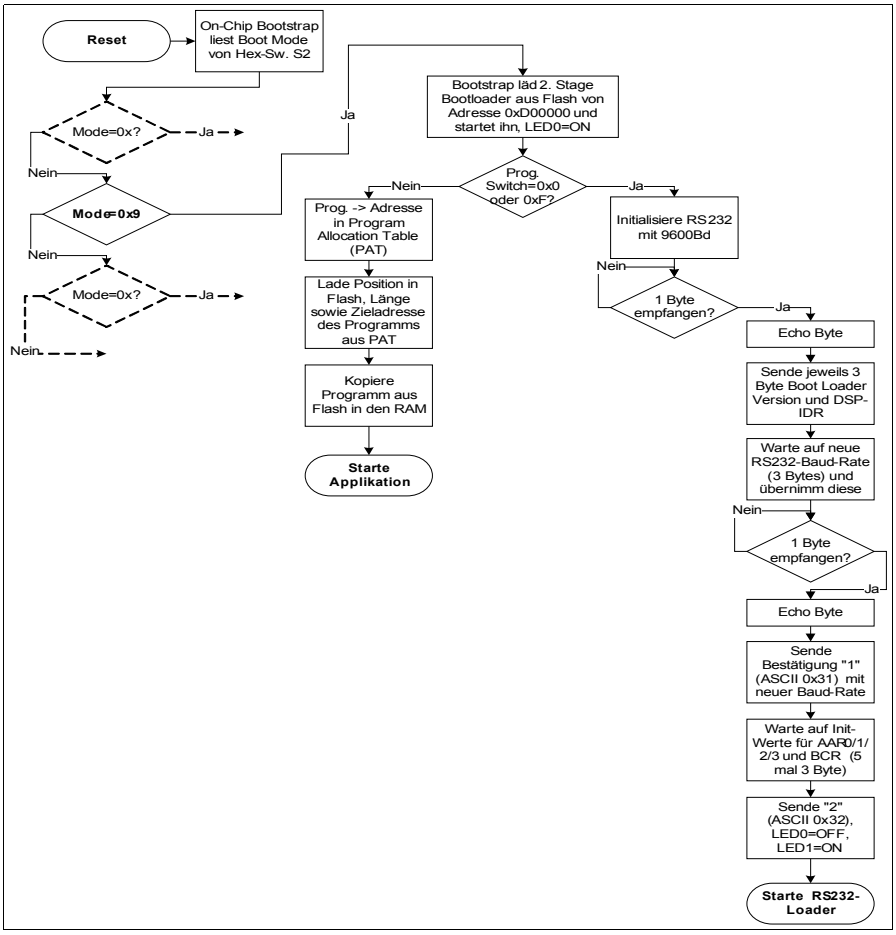

Bild 1: Ablauf des Boot-Vorganges

2.1. Autarkes Laden und Starten von Applikationen

Wird vom Betriebsartenschalter eine Zahl zwischen 1...14 eingelesen, soll eine Applikation aus dem Flash-EEPROM geladen werden. An welcher Adresse das jeweilige Programm im Flash zu finden ist, dessen Länge und Zieladresse im DSP-RAM, sind in der Program Allocation Table (PAT) vermerkt. Da der Flash eine Wortbreite von 8 Bit gleich 1 Byte hat, sind jeweils drei Byte für jede Eintragung nötig. In den ersten 6 Sektoren des Flash-Bausteins befindet sich der Second Stage Bootloader, daher folgt nun die PAT im 7. Sektor. Der Flash selbst wurde in den Adressbereich ab 040000h eingeblendet, so daß die PAT ab 040380h zu finden ist (1 Sektor hat 128 Byte mal 7 macht 380h). Bild 2 gibt eine Übersicht über Adressen und Aufbau der Program Allocation Table.

Adresse	Flash	Datum	
040000h	low byte	2nd Stage Boot Loader	
.	mid byte		
.	high byte		
.	.		
.	.		
.	.		
040380h	low byte	Adresse im Flash	**Applikation #1**
040381h	mid byte		
040382h	high byte		
040383h	low byte	Programm-länge	
040384h	mid byte		
040385h	high byte		
040386h	low byte	Ziel- und Startadresse im DSP	
040387h	mid byte		
040388h	high byte		
040389h	low byte	Adresse im Flash	**Applikation #2**
04038ah	mid byte		
04038bh	high byte		
04038ch	low byte	Programm-länge	
04038dh	mid byte		
04038eh	high byte		
04038fh	low byte	Ziel- und Startadresse im DSP	
040390h	mid byte		
040391h	high byte		
040392h	low byte	Adresse im Flash	**Applikation #3**
040393h	mid byte		
040394h	high byte		
040395h	low byte	Programm-länge	
040396h	mid byte		
040397h	high byte		
040398h	low byte	Ziel- und Startadresse im DSP	
040399h	mid byte		
04039ah	high byte		
.	.	.	.
.	.	.	.
.	.	.	.
0403f5h	low byte	Adresse im Flash	**Applikation #14**
0403f6h	mid byte		
0403f7h	high byte		
0403f8h	low byte	Programm-länge	
0403f9h	mid byte		
0403fah	high byte		
0403fbh	low byte	Ziel- und Startadresse im DSP	
0403fch	mid byte		
0403fdh	high byte		
0403feh	low byte	**16Bit Check-summe PAT**	
0403ffh	high byte		

Bild 2: Speicherorganisation Flash-EEPROM mit PAT

2.2. Initialisierung des RS232-Loaders

Ist kein Betriebsartenschalter an K6 installiert, oder wird eine „0“ oder „F“ gewählt, soll der serielle Programmlader gestartet werden. Dies wird durch Aufleuchten der LED0 angezeigt. Bild 1 gibt im rechten Zweig den Ablauf bis zum eigentlichen Loader wider. Zuerst wird die RS232 mit 9600Bd (1 Startbit, 8 Datenbits, 1 Stopbit, keine Parität) initialisiert und zur Synchronisation der Kommunikation auf ein erstes empfangenes Byte gewartet. Wird dieses Byte empfangen, erfolgt ein Echo desselben zur Bestätigung und es werden jeweils drei Byte (1 Wort des DSP) Boot-Loader-Version und DSP-Identifikation gesendet (low, mid, high). Dies dient der Rückwärtskompatibilität bei möglichen Folgeversionen von Hardware und Boot-Loader. Nun erwartet das UniDSP56 eine neue Baudrate für die serielle Kommunikation in Form von drei Bytes, die den Inhalt des Teilerregisters SCCR angeben. Hierbei wird zur Flußkontrolle jedes Byte direkt zurückgesendet. Der Wert für das Register berechnet sich wie folgt: SCCR_VAL=floor(100e6/(64*Baudrate)-0.5). Die Baudrate ist damit bis hoch zu 500kBit/s einstellbar. Beim Programm *spush.exe* wird immer auf die höchste Übertragungsrate des PCs von 115200Bit/s geschaltet. Anschließend wird auf ein beliebiges Byte des PCs mit der neuen Baudrate gewartet, welches zur Bestätigung sogleich zurück gesendet wird, gefolgt von einer „1“ (ASCII 0x31), deren korrekter Empfang auf der Host-Seite (z.B. PC) die Geschwindigkeitsumschaltung abschließt.

Da die Kommunikation nun mit einer hohen Baudrate funktioniert, erfolgt jetzt das Einstellen des Mappings sowie der Wait-States der externen Speicher des DSPs, damit auch diese in der Folge mit Daten beladen werden können. Dazu werden die Initialisierungswerte für die Adress-Attribut-Register 0...3 und das Bus-Control-Register an den DSP übertragen. Es handelt sich also um fünf 24Bit-Worte mit je 3 Byte (low, mid, high). Es wird wiederum jedes Byte zur Flußkontrolle vom DSP zurückgesendet. Ist dies erfolgreich beendet, antwortet der DSP mit einer „2“ (ASCII 0x32) und die LED1 leuchtet (LED0 aus). Nun wird der eigentliche RS232-Loader gestartet.

3. Der eigentliche RS232-Loader

Die nun gestartete Laderoutine gestattet das iterative Beschreiben der verschiedenen internen und externen Speicherbereiche, sowie das Starten eines Programmes von einer vorgegebenen Adresse. Es stehen dazu fünf Kommandos zur Verfügung. Die verschiedenen Kommandos und der Ablauf sind in Bild 3 dargestellt. Während die ersten drei Worte (3x3Byte, Kommando, Länge, Zieladresse) vom DSP durch Zurücksenden eines jeden Bytes bestätigt werden, erfolgt keinerlei Wiederholung der übertragenden Speicherdaten. Diese werden *wortweise* (24 Bit!) zu einer Check-Summe aufaddiert und nach dem letzten Datenwort an den Host übertragen, um die Datenintegrität zu bestimmen (low, mid, high).

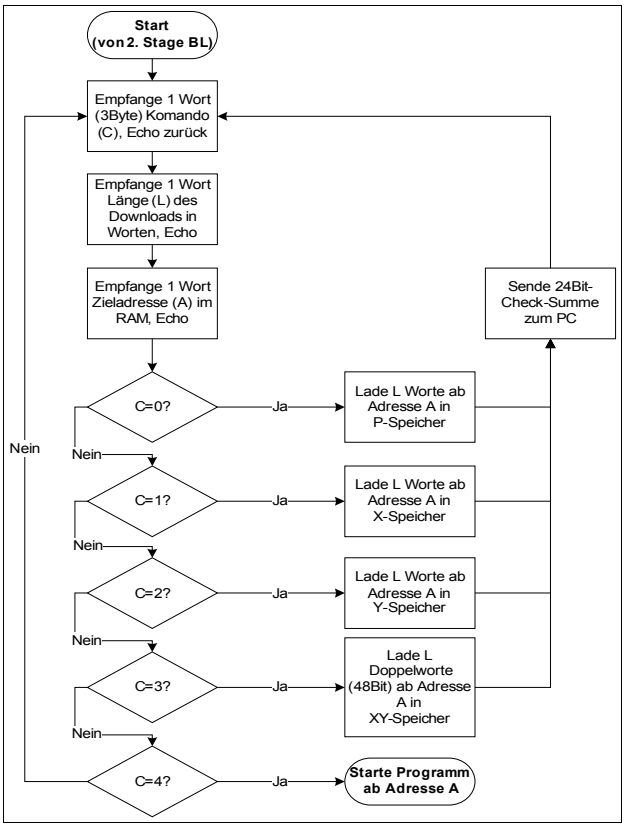

Bild 3: Ablaufdiagramm des RS232-Loaders