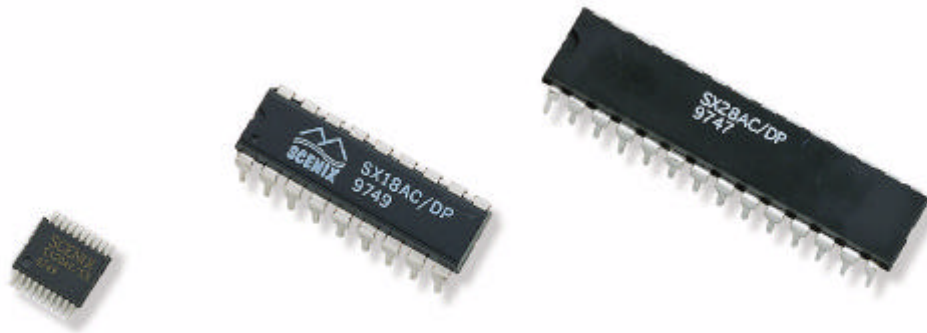


# **μController mit virtueller Peripherie**

vorge stellt in

**Design&Elektronik 6/98**

**DESIGN & ELEKTRONIK**  
PRODUKTE UND KNOW-HOW FÜR DEN ELEKTRONIK-ENTWICKLER



## **Scenix SX - 8 Bit Mikrocontroller mit Virtueller Peripherie und 50 MHz Taktfrequenz**

Im Februar dieses Jahres wurde er in Deutschland auf der Embedded Systems in Sindelfingen vorgestellt: der SX von dem kalifornischen Unternehmen Scenix. Als Entwicklungsumgebung gibt es dazu das „SX-Key Development System“ von der Firma Parallax. Was dieser Controller leisten kann und wo seine Stärken liegen, soll in diesem Beitrag vorgestellt werden.

Bei der Auswahl eines Mikrocontrollers für ein Produkt kommt es auf Leistung und Preis gleichermaßen an. Auf der einen Seite sollen in einem Mikrocontroller möglichst alle wesentlichen Komponenten enthalten sein, die für ein Design benötigt werden. Auf der anderen Seite kosten nicht benötigte Module Chipfläche und damit auch Geld.

Genau hier setzt das Konzept des SX an: Man nehme einen sehr schnellen Rechnerkern mit RISC-ähnlicher Struktur (20 ns Zykluszeit - 50 MIPS) mit schnellem Programmspeicher (12 ns Flash), dazu ein optimiertes Interrupt- und Timersystem sowie ein paar I/O-Ports - fertig! Alle weiteren benötigten Peripheriekomponenten werden mit Software realisiert - daher die Bezeichnung „virtuelle Peripherie“. Durch diese hohe Rechenleistung können in Software viele Aufgaben realisiert werden, für die bisher nur Hardwarekomponenten in Frage kamen - das Silizium wird optimal genutzt! In dem SX-Key Development System sind einige virtuelle Peripheriekomponenten enthalten und auf der Website von Scenix und Parallax gibt es noch mehr davon. Einige Beispiele für virtuelle Peripheriekomponenten sind: UART, SPI, I<sup>2</sup>C, Rechteckgeneratoren, Frequenzmessung, PWM, ADC, Motorsteuerung, Spektralanalyse usw.

Keinesfalls selbstverständlich in dieser Klasse sind die im SX realisierte In-System-Programmierung und In-Circuit-Emulation. Dazu wird der SX-Key (Bild 1) einfach an die beiden Oszillatorpins des zu debuggenden oder zu programmierenden SX angeschlossen. Der Rest wird über die Software des SX-Key Development Systems gesteuert und erlaubt damit ein komfortables Arbeiten.

Externe Programmiergeräte, Emulatoren oder eine dedizierte Debuggerschnittstelle am SX werden nicht benötigt.

Mit diesen Fähigkeiten ausgerüstet, ist der SX mit Preisen von 4-5 \$ pro 100 Stück für den Einsatz in kostensensitiven Applikationen prädestiniert.



Bild 1: SX-Key Programmiergerät und Emulator

## SX Architektur

Im Bild 2 ist das Blockschaltbild des SX dargestellt. Um die hohe Rechenleistung von 50 MIPS zu erreichen, wurde eine Harvard Architektur mit separaten Daten- und Programmspeichern und getrennten Bussen eingesetzt. Den Lesern, die bisher schon mit PIC-Mikrocontrollern gearbeitet haben, wird dabei einiges bekannt vorkommen. Der Objektcode ist ein Superset der PIC16C5x-Familie mit einer Anzahl entscheidender Verbesserungen. Die Erweiterungen im Befehlssatz des SX sind optimiert auf das Konzept der Virtuellen Peripherie. Es existiert ein kompatibler Modus, so daß auch für den PIC geschriebene Programme unmittelbar lauffähig sind.

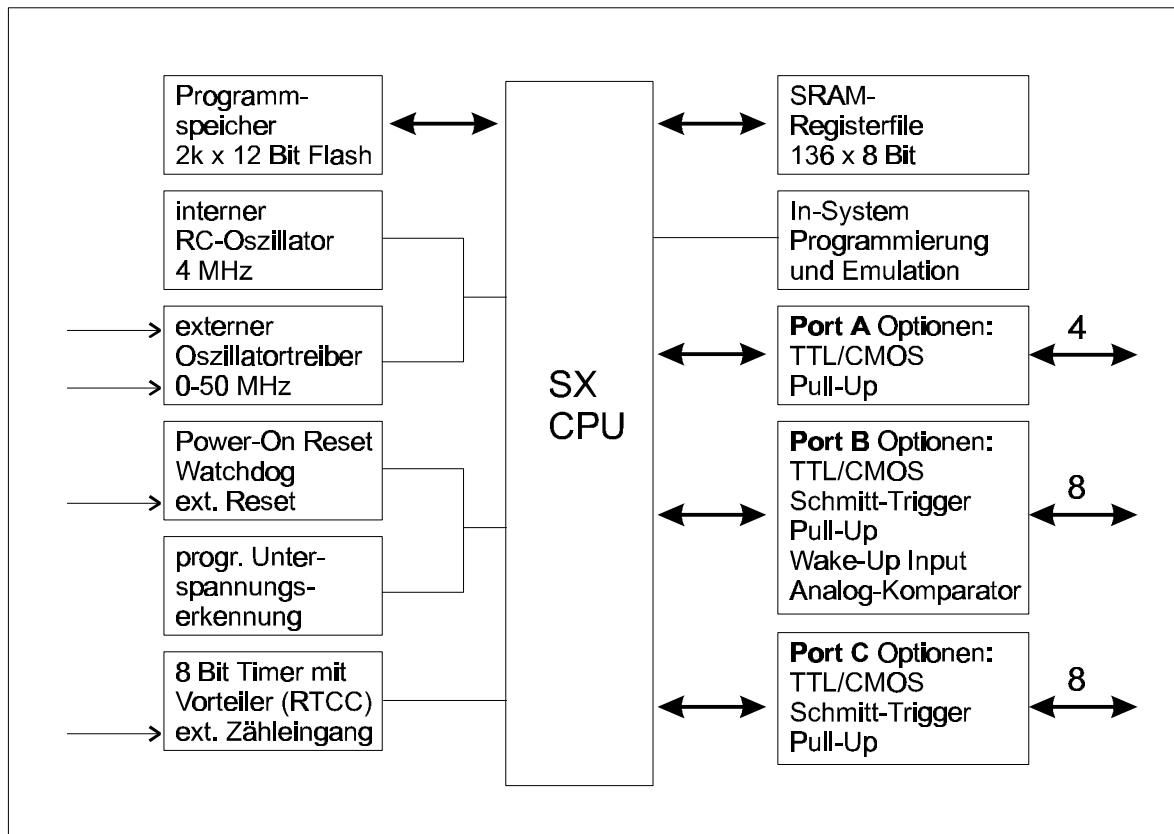


Bild 2: SX Blockschaltbild

Durch die vielfältigen Möglichkeiten hinsichtlich Oszillatorbeschaltung (RC-Glied, Quarze

32 kHz - 50 MHz, Keramikresonatoren), integriertem Power-On Reset, Watchdog, programmierbarer Unterspannungserkennung, programmierbarer Pull-Up Widerstände usw. kommt der SX mit sehr wenigen externen Bauelementen aus.

Die meisten Befehle werden in einem Taktzyklus abgearbeitet (Befehlswort mit 12 Bit Breite). Der SX nutzt eine vierstufige Pipeline zur Befehlsausführung:

1. Befehl holen
2. Befehl decodieren
3. Befehl ausführen
4. Ergebnis zurückschreiben

Damit wird klar, daß die maximale Verarbeitungsgeschwindigkeit von 50 MIPS nur erreicht wird, wenn die Pipeline gefüllt ist. Sprünge und Unterprogrammaufrufe benötigen 3 Taktzyklen, löschen die Pipeline und verzögern damit die Programmabarbeitung. Günstiger einzusetzen sind daher „Skip“- Befehle wie „vergleiche zwei Operanden und wenn beide gleich sind, überspringe den nächsten Befehl“, da dabei die Pipeline nicht gelöscht wird.

Der SX besitzt weiterhin einen 8 Ebenen Hardwarestack sowie verborgene Schattenregister, die bei Auftreten eines Interrupts den Prozessorstatus retten.

Der Programmspeicher von 2048 x 12 Bit ist in 4 Seiten zu 512 Worten organisiert. Um eine entsprechende Seite anzuwählen, wurde der PAGE Befehl eingeführt.

Der Datenspeicher ist 8 Bit breit und wie in Bild 3 angegeben organisiert:

Von Adresse 00H bis 07H befinden sich die Register, die für Spezialfunktionen des SX benötigt werden. Von Adresse 08H bis 0FH liegt der globale Datenspeicher, auf den immer zugegriffen werden kann. Auf den Datenspeicher bei den Adressen 10H-1FH (Bank 1), 30H-3FH (Bank 2) usw. kann erst nach Auswahl der entsprechenden Bank zugegriffen werden. Der dafür notwendige Befehl ist BANK und modifiziert das File Select Register (FSR). Eine sinnvolle Nutzung der Speicherbänke wären z.B. lokale Variablen für virtuelle Peripheriekomponenten.

Adresse	Bank 0 00H	Bank 1 20H	Bank 2 40H	Bank 3 60H	Bank 4 80H	Bank 5 A0H	Bank 6 C0H	Bank 7 E0H
00H	INDIRECT							
01H	RTCC (Real Time Clock / Counter)							
02H	PC (Program Counter)							
03H	STATUS (Prozessor Flagregister)							
04H	FSR (File Select Register)							
05H	RA (Port A Control Register)							
06H	RB (Port B Control Register)							
07H	RC (Port C Control Register)							
08H	frei für Anwendung							
09H	frei für Anwendung							
0AH	frei für Anwendung							
0BH	frei für Anwendung							
0CH	frei für Anwendung							
0DH	frei für Anwendung							
0EH	frei für Anwendung							
0FH	frei für Anwendung							
10H								
11H								
...bis...	...	...	...	...	...	...	...	...
1EH								
1FH								

Bild 3: Datenspeicher

Weitere Eigenschaften des SX im Überblick:

- 50 MIPS Rechenleistung bei 50 MHz Takt
- 2k x 12 Bit Flash-Programmspeicher mit 10000 Programmierzyklen
- PIC16C5x Superset kompatibel
- In-System Programmierung und Debugging über Oszillatorpins
- I/O-Pin Treiberleistung: 30 mA
- Stromaufnahme: 40 mA bei 50 MHz, 15 mA bei 20 MHz, 10 µA standby
- Gehäuse: 18 pin, 20 pin, 28, pin ; SDIP,SOIC,SSOP
- Multi-Input Wake-Up (z.B. für Tastaturinterrupt)

## SX-Key Development System

Zur Programmentwicklung und Programmierung liefert Parallax kostengünstig das SX-Key Development System.

Darin enthalten sind:

- SX-Key (Bild 1) mit Kabel zum Anschluß an die serielle Schnittstelle des PC
- SX-Key Demoboard (Bild 4)
- 2 Stück 28-pin SX Chips
- das SX-Key Development Software mit Assembler, Debugger und Programmer
- Handbuch

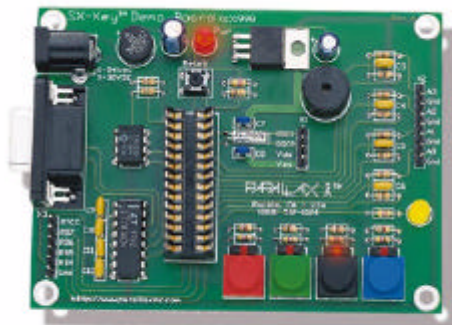


Bild 4: SX-Key Demoboard

Zuerst ist der SX-Key an den PC und das Demoboard anzuschließen (OSC1, OSC2, VDD, VSS). Danach kann die SX-Key Development Software gestartet werden. Die Bedienung ist sehr einfach: Source öffnen und editieren, Assembler starten, Programmieren - fertig ! Die Optionen für die Programmierung können alternativ im Sourcefile oder im Menü Device Setup eingestellt werden. Für die Programmierung des Beispielprogramms mit 364 Worten wurden 45 s benötigt. Eine Erhöhung der Programmiergeschwindigkeit ist allerdings bereits angekündigt.

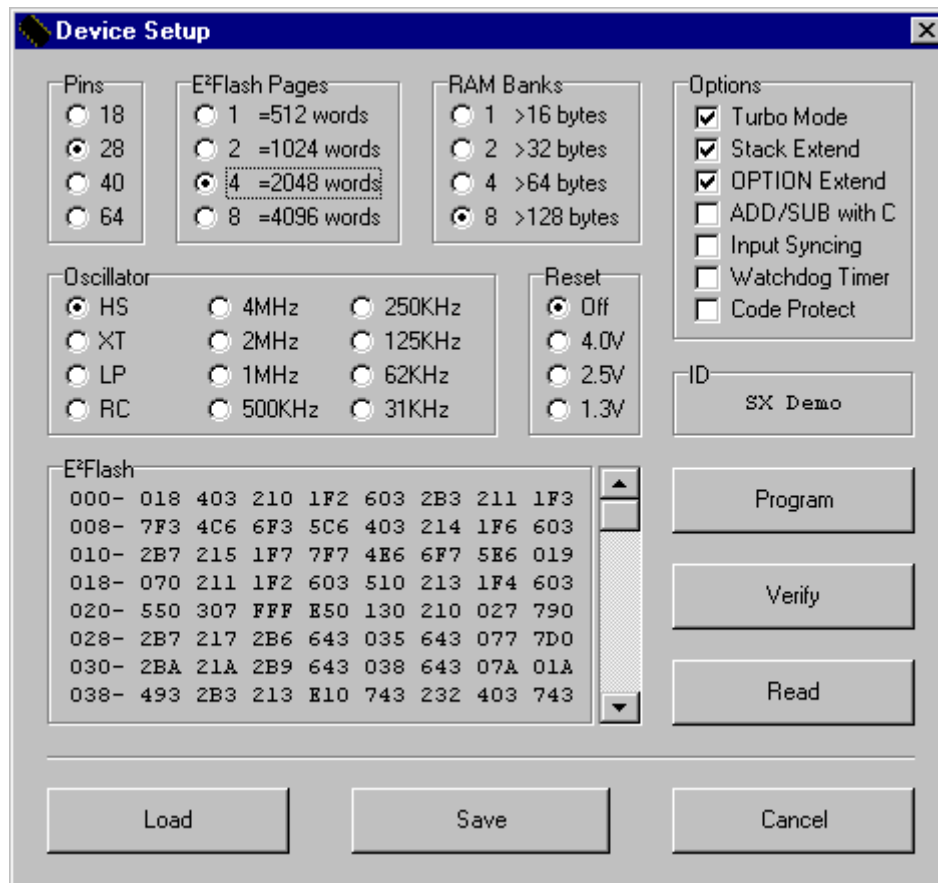


Bild 5: Device Setup

Gelegentlich empfiehlt sich ein Blick in das Assembler Listing.

Mit etwas Überraschung wird man beim ersten Mal zur Kenntnis nehmen, daß der Assembler komplexere Befehle wie „Lade Register 1 mit Register 2“ in zwei Befehle unterteilt:

1. Lade Arbeitsregister W mit Register 2
2. Lade Register 1 mit Arbeitsregister W

Damit sind Parallax Assembler Befehlssatz und SX Befehlssatz nicht ganz identisch!

Leider wird das Sourcefenster immer nach dem Öffnen des Listings geschlossen. In Anbetracht der momentan vorliegenden Softwareversion 0.3 kann man aber sicher noch einige Verbesserungen in Richtung Bedienerfreundlichkeit erwarten.

Wenn der scharfe Blick in das Listing nicht mehr genügt, ist es Zeit, den Debugger zu starten. Im Debugger wird der Status des SX angezeigt. Es sind folgende Aktionen möglich:

- Step (Ausführung einer Instruktion mit Aktualisierung des Debug Fensters)
- Walk (Ausführung mehrerer Instruktionen mit Aktualisierung des Debug Fensters )
- Run (Ausführung des Programms mit voller Geschwindigkeit)
- Stop
- Reset
- Modifizierung von Registerinhalten während der Emulation
- Setzen von Breakpoints und Watchpoints

Geänderte Werte werden im Display farblich hervorgehoben.

In der vorliegenden Programmversion 0.3 ist das Debugging noch nicht möglich. Das Debugging wird allerdings mit Auslieferung des angekündigten Updates verfügbar sein. Als sinnvolle Ergänzung werden Hochsprachen wie BASIC (Parallax) und C (Bytcraft) angeboten.

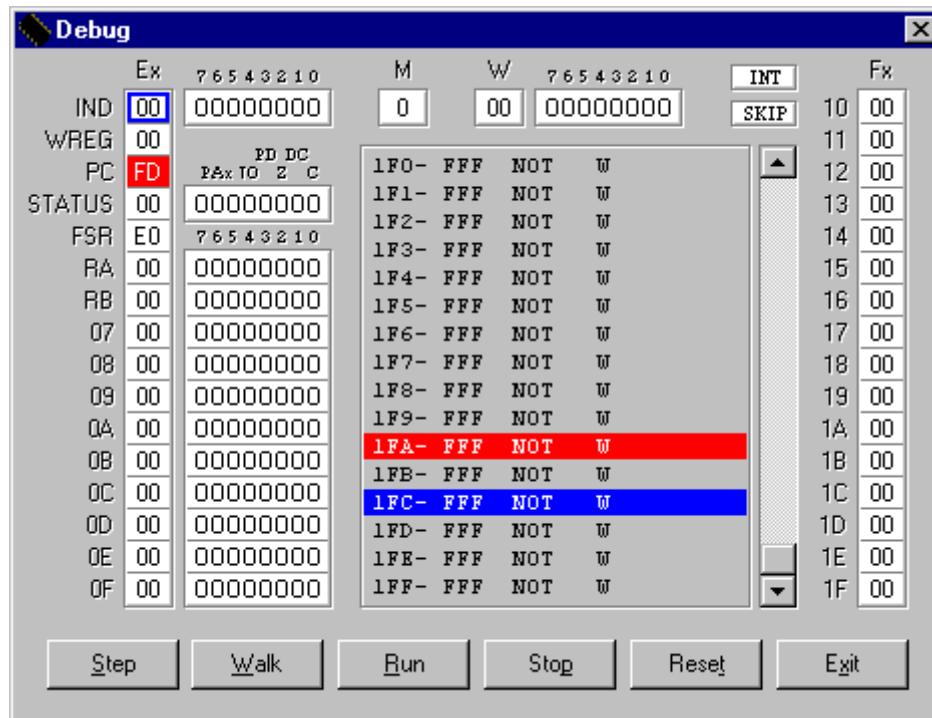


Bild 6: Debug Fenster

## Zeitverhalten mit Virtueller Peripherie

Um ein Gefühl für das Zeitverhalten des SX bei dem Einsatz der virtuellen Peripherie zu bekommen, wird im folgenden eine Beispielimplementation von Parallax vorgestellt, die die Leistungsfähigkeit dieses Konzepts demonstrieren soll.

Es sind hier im Detail die folgenden Peripheriekomponenten zu realisieren:

- 2 separate Signalgeneratoren für Rechteckimpulse (mit 16 Bit-Timern)
- 2 separate 8 Bit Digital/Analog-Wandler mit PWM
- 2 separate 8 Bit Analog/Digital -Wandler als Sigma/Delta Wandler mit 256-fachem Oversampling (Samplingrate ca.1,2 kHz)
- serielle Schnittstelle (RS-232), voll duplex mit 19,2 kBit

Für die Bedienung der virtuellen Peripherie wird kontinuierlich ein Interrupt ausgelöst, der dann die virtuellen Peripheriekomponenten nacheinander bedient. Es wird eine zyklische Abarbeitungsrate von ca. 306 kHz erreicht. Dabei ist zu beachten, daß nicht immer alle Komponenten bei jedem Interrupt ein Update erfahren (z.B. die serielle Schnittstelle erst bei jedem 16. Interrupt).

Die Ergebnisse werden in Flags bzw. im RAM abgelegt. Die Weiterverarbeitung der Ergebnisse erfolgt dann später im Hauptprogramm.

Um die Interrupts jitterfrei erzeugen zu können, wird als Interruptquelle der 8 Bit Real-Time-Clock/Counter (RTCC) benutzt, der bei jedem Befehlszyklus inkrementiert wird. Bei jedem Überlauf des RTCC wird ein Interrupt ausgelöst. Dabei werden automatisch das Arbeitsregister, Statusregister und File Select Register in Schattenregister gerettet, der Program Counter auf dem Stack abgelegt, weitere Interrupts verboten und die Interruptroutine ab Adresse 000H angesprungen. Bis zum Beginn der Interruptroutine dauern diese Aktionen 3 Zyklen (60ns) aufgrund der Pipeline.

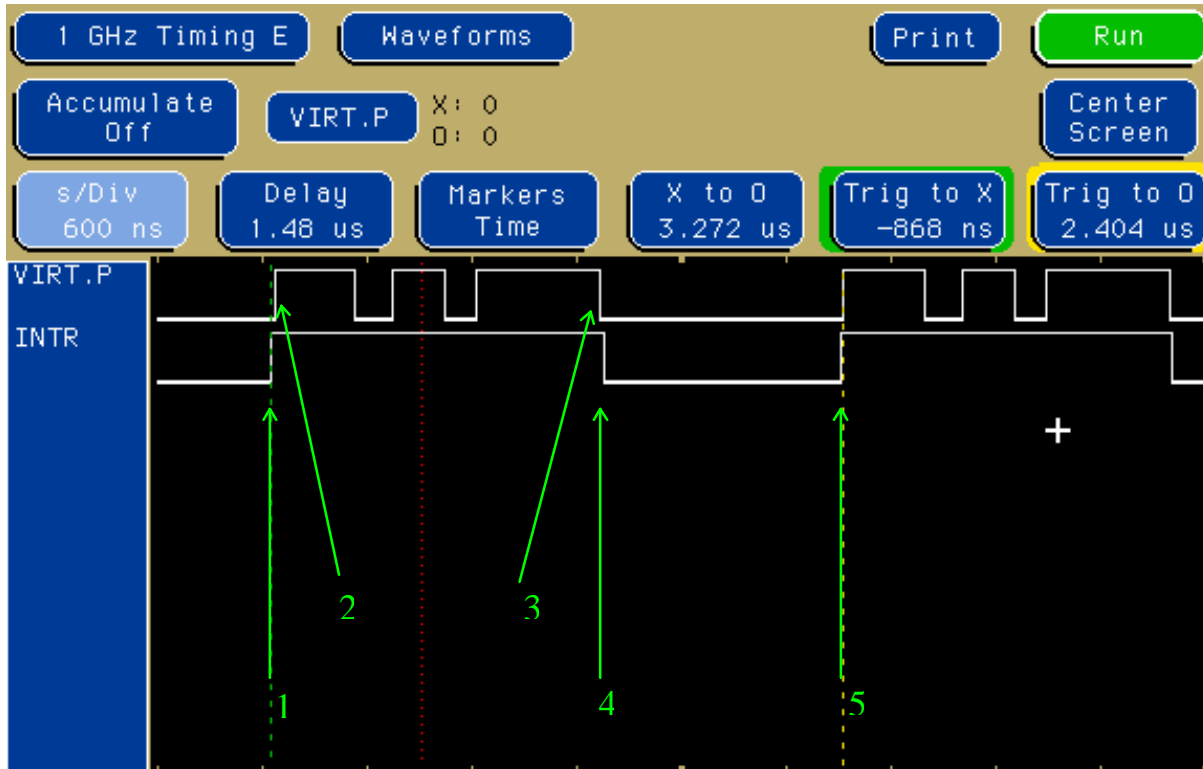
Es existiert u.a. eine „Return from Interrupt“-Instruktion, die nach 3 Zyklen (60ns) die geretteten Register wiederherstellt, neue Interrupts erlaubt und den RTCC gleichzeitig um einen bestimmten Wert „vorstellen“ kann, so daß der Zeitpunkt des nächsten RTCC-Interrupts ohne Zusatzaufwand wählbar ist. Durch diese Hardwareunterstützung ist die Latenzzeit minimal.

Um das Zeitverhalten transparent darstellen zu können, wurden zwei Pins des SX nur zum Tracing benutzt. Dazu sind bei Beginn und Ende des Interrupts und der einzelnen Routinen der virtuellen Peripheriekomponenten Instrumentierungen (Kontrollmeldungen) eingefügt worden. Der Code der Interruptroutine belegt samt Instrumentierung 102 Worte. Das Pin „INTR“ signalisiert mit High-Pegel, daß die Interruptroutine aktiv ist und das Pin „VIRT.P“ kennzeichnet die Rechenzeit für die jeweiligen Peripheriekomponenten.

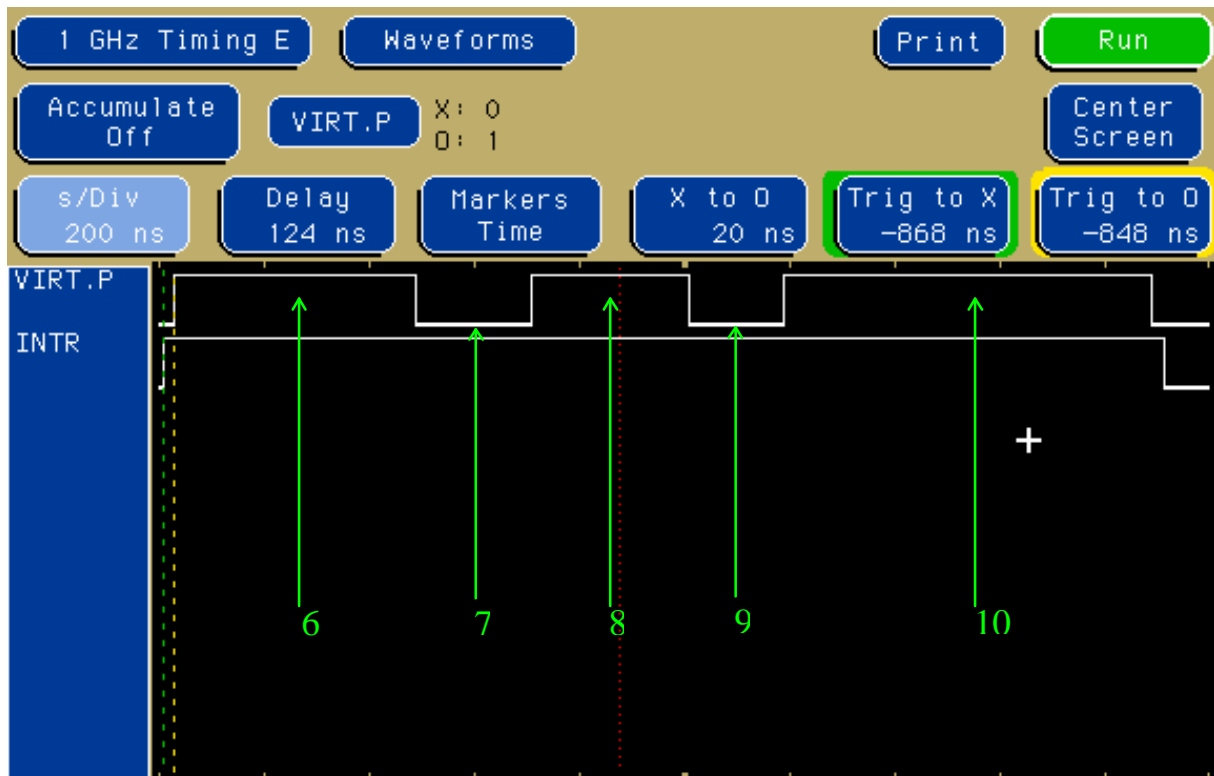
Im folgenden Screenshot des Logikanalysators sind zwei Interrupts zu erkennen. Der Zeitpunkt (1) kennzeichnet den Beginn und (4) das Ende der Interruptroutine. Die Interruptroutine dauert ca. 1,9µs. Zum Vergleich: Ein Fließkomma DSP z.B. TI TMS320C31 rechnet bei ebenfalls 50 MHz Takt an einem Bandpaßfilter 2. Ordnung ca. 1,6µs !

Der nächste Interrupt (5) erfolgt  $3,27\mu\text{s}$  nach Beginn des letzten. Die verbleibende Rechenzeit zwischen (4) und (5) von  $1,37\mu\text{s}$  steht dann der Anwendung zur Verfügung. Das sind noch 37 % der Rechenleistung des SX und immerhin (rechnerisch) 18,5 MIPS.

Gut zu erkennen ist auch die geringe Verzögerung von 20ns beim sequentiellen Schalten der beiden Tracingsignale an (3) und (4).



Im nächsten Screenshot des Logikanalysators soll das Timing für einen Interrupt dargestellt werden.



Die folgende Tabelle faßt die Meßergebnisse zusammen:

Virtuelle Peripheriekomponente	Realisierung	Nr. im Screenshot	Rechenzeit
LED	16 Bit-Timer (Freq. Generator)		
Lautsprecher	16 Bit-Timer (Freq. Generator)	(6)	460 ns
Analog Ausgabe 1	8 Bit PWM		
Analog Ausgabe 2	8 Bit PWM	(7)	220 ns
Analog Eingabe 1	8 Bit ADC und Port Updates Kanäle 1 + 2	(8)	300 ns
Analog Eingabe 2	8 Bit ADC	(9)	180 ns
serielle Schnittstelle	19.2 kBit RS-232	(10)	700 ns

Autorenkontakt:

Thomas Wurlitzer  
c/o MCT Lange & Thamm  
Hohe Str. 9-13  
04107 Leipzig

eMail: [tw@elektronikladen.de](mailto:tw@elektronikladen.de)  
Fax: 0341-2118355

Vertrieb in Deutschland:

Elektronikladen Mikrocomputer GmbH  
Tel.: 05232-8171  
Fax: 05232-86197

Internetlinks:

[www.parallaxinc.com](http://www.parallaxinc.com)  
[www.scenix.com](http://www.scenix.com)  
[www.mrt.htwk-leipzig.de](http://www.mrt.htwk-leipzig.de)  
[www.elektronikladen.de/sx.html](http://www.elektronikladen.de/sx.html)