cranach SoC [FACT SHEET]

Netzwerkfähige 'netpp on chip' Prozessorlösungen für (kleine) FPGAs

Das cranach System on Chip (SoC) Design gehört zur Familie der *cCAP* (Configureable-Custom Application Processor). Diese zeichnet sich durch hohe Konfigurierbarkeit wie auch geringen Resourcenbedarf auf FPGA-Bausteinen aus. Unabhängig von der eingesetzten FPGA-Technologie können Peripheriegeräte und IP-Cores von Registeradressen bis Anzahl der Peripherie-Instanzen wie z.B. UARTs oder SPI-Interfaces beliebig auf spezielle Anwendungen zugeschnitten werden.

Die cCAP-Familie wurde speziell entwickelt, um spezifischen Anforderungen an beweisbarer, robuster Funktionalität gerecht zu werden, wie auch Anwendungsprogrammierern die Nutzung von FPGA-Technologien zu erleichtern. Die kompakte Befehlsstruktur der Kernarchitektur ermöglicht höchst platzsparende Programme wie auch kundenspezifische Mikrocode-Erweiterungen mit DSP-Befehlen.

1 Funktionsübersicht 'cranach'

- 32 Bit-Processor 'ZPUng' v1.1, dreistufige Pipeline
- Wishbone-Bus für Adress-Decoder und Peripherie
- DMA für Ethernet RX/TX
- Programmierbar in C (GCC)
- Hardware-Debugger (JTAG), GDB
- Ethernet, UDP/IP und **netpp** (Network Property Protocol) Stack

Technische Daten der Standardkonfiguration siehe Tabelle 1.

Mit dem vorinstallierten *netpp*-Betriebssystem werden Eigenschaften wie Betriebsparameter per Gerätebeschreibung (siehe Abb. 1) definiert und ins SPI Flash geladen. Die Low-Level-Treiber für die betreffenden Interfaces sind Teil der Firmware. Die Ansteuerung von aussen kann per Python-Kommando, netpp client oder grafischem Interface wie auch Webserver erfolgen. Weitere Sensoren können somit ohne Programmierung per Registerbeschreibung angesprochen werden.

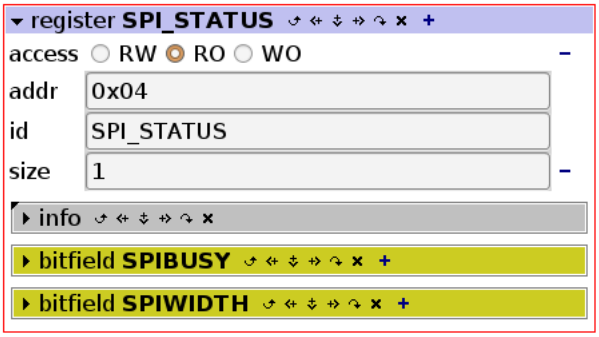

Fig. 1: SPI Register-Beschreibung

2 Simulation

Der komplette SoC kann mit der Anwendung unter realen Szenarien in einer virtuellen Maschine co-simuliert werden. Vorteile:

- zyklengenaue Verifikation der korrekten Funktion
- Effiziente visuelle Fehlersuche in Wellenformen (Abb. 2), optionales on-chip Trace Debugging

3 Beispielanwendungen

- Intelligente Fernkontrolle von FPGA-Logik per Python-Script, etc.
- Konfigurationsprozessor für High-Speed Serdes-Interfaces
- Echtzeit-fähiger PWM-Controller mit garantierten Netzwerk-Latenzzeiten
- Rechenbeschleuniger
- Redundante Systeme (Multi-Core)

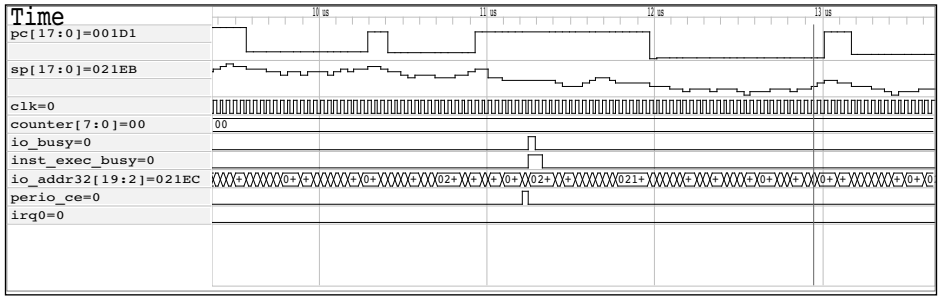

Fig. 2: Visualisierung Wellenform

Technische Daten		
	Speicher	
ROM	256kB	Programmspeicher, nur Lesen
L1RAM	16kB	Programm und Daten-Segment
L1CACHE	16kB	Programm/Daten oder ROM-Cache
SRAM	2kB	Dediziertes Stack-Memory
SPAD	1kB	ScratchPad RAM für DMA/Daten
	Peripherie	
SCACHE	1	SPI flash cache
SIC	1	System Interrupt Controller mit 4-Peripherie-Kanälen, priorisierbar
DMA	2	Zwei simultane DMA-Kanäle für Ethernet Send/Receive Packet Queues, interruptfähig
UART	1	UART-Console
PWM	3	PWMPlus für zyklengenaue Echtzeitsteuerungen, Systemtimer (TIMER0)
GPIO	2x16	Generische I/O Pins für Bit-Banging
MAC	1	Media Access Control Core (RGMII/MDIO), 100 MBit
SYSEXT	1	CRC-Beschleuniger

Tab. 1: cranach Standardkonfiguration

Python-Beispiel

```
import netpp

fpga = netpp.connect("UDP:192.168.0.5:2016")
r = fpga.sync()
if r.Status.get() == 1:
    r.LED.Green.set(1)
```

4 Referenz-Plattformen

Der cranach-SoC ist für folgende Referenzplattformen als Netzliste verfügbar:

- Lattice ECP5 Versa Eval board
- Trenz TE-0600 Core-Modul
- In Planung: 'netpp node' Evaluationskit auf Spartan6 Basis

Portierungen auf weitere Plattformen sind auf Anfrage möglich.

Der Resourcenbedarf ist den untenstehenden Tabellen zu entnehmen.

Lattice ECP5	
SLICE	3370
Registers	3100
LUT4	4590
EBR	28

Tab. 2: Resource usage ECP5

Xilinx Spartan6	
Slice Registers	2395
Slice LUT	5780
LUT-FF	2364
BlockRAM	26

Tab. 3: Resource usage Spartan6

5 Weiterführende Informationen

Web-Übersicht zur cCAP-Familie
http://section5.ch/index.php/dienstleistungen/custom-cpu-solutions/

cranach Hardware-Referenz `soc-cranach.pdf`
Auf Anfrage