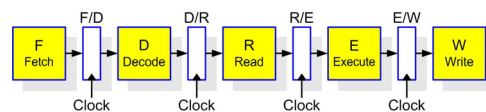
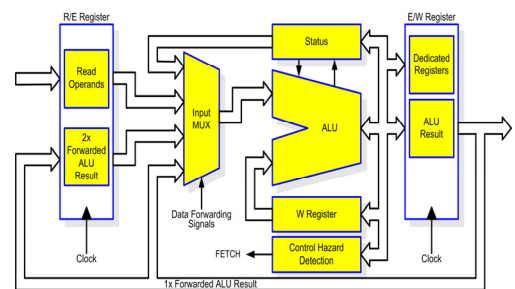
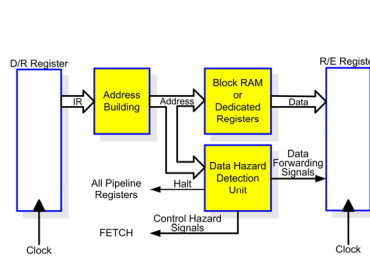
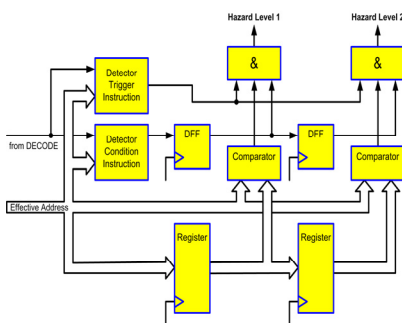


## Entwicklung eines FPGA-basierten Mikrocontrollers mit Pipelineverarbeitung 2007 - 2010

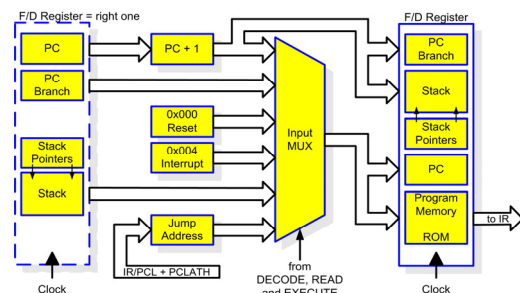
- VHDL-basierter Softcore-Prozessor für System-on-Chip Designs (SoC) in FPGAs
- SW-kompatibel zu Microchip PIC16xxx (Mid-Range Family), basierend auf voll kompatibler, non-pipelined FPGA-Version (100 MHz in Xilinx Spartan 3, 22 Assembler MIPS)
- fünfstufige Pipeline (Fetch, Decode, Read, Execute, Write)



- effiziente Data Hazard Detection Unit in Read steuert Operandenauswahl in Execute



- optimierte Behandlung von unbedingten Verzweigungen, bedingten Sprüngen und Computed Jumps in der Fetch-Stufe (und der Decode- und Read-Stufe) sichert geringe Verluste durch Kontrollflusschasards



- Speed-up gegenüber Version ohne Pipelining zwischen 2,3 und 3,5 bei nur etwa 30% höherem Ressourcenbedarf
- bis zu 70 MHz Taktfrequenz in Xilinx Spartan3
- bis zu 70 Assembler MIPS

Program	Standard (cycles)	Pipelined (cycles)	Ratio	Speedup (x 70/100)
Embedded I/O	1,249,700	250,452	4.99	3.49
Floating Point	3,348	1,007	3.32	2.33

Map report	(total)	Pipelined	Standard	Increase in hardware
Number of slices	(1,920)	772 (40%)	601 (31%)	+28%
Slice Flip Flops	(3,840)	515 (13%)	356 (9%)	+45%
4 input LUTs	(3,840)	1,097 (28%)	864 (22%)	+27%

Ansprechpartner: Prof. Dr.-Ing. Rainer Bernbach (r.bernbach@ostfalia.de, 05331 / 939 - 42620)

Ostfalia Hochschule für angewandte Wissenschaften, Standort Wolfenbüttel, Salzdahlumer Str., 46/48, 38302 Wolfenbüttel



## Entwicklung eines FPGA-basierten Mikrocontrollers mit Pipelineverarbeitung

Rainer Bermbach

Ostfalia Hochschule für angewandte Wissenschaften  
Fakultät Elektrotechnik  
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel  
r.bermbach@ostfalia.de

In modernen Elektronikdesigns werden Mikrocontroller und andere Schaltungsteile zunehmend durch sog. IPs (Intellectual Property) in FPGA- oder Gate-Array-Implementationen ersetzt. Diese System-on-Chip Designs (SoC) erlauben durch ihre hohe Flexibilität eine effektive Entwicklung und interessante Systemimplementierungen. Im Rahmen einer Machbarkeitsstudie hatte das Labor Datentechnik der Fakultät Elektrotechnik an der Ostfalia für solche SoC Designs eine voll kompatible VHDL-Version der bekannten Mikrocontrollerfamilie PIC16xxx entwickelt, die mit einer Taktfrequenz von bis zu 100 MHz auf einem Xilinx Spartan 3 FPGA etwa 22 Assembler MIPS in praktischen Programme erreicht.

Um die Leistungsfähigkeit noch weiter zu steigern, wurde eine Version mit einer mehrstufigen Pipeline entwickelt. Hardwaregrundlage war erneut ein Xilinx Spartan 3, dessen Block RAMs für die Implementierung des Programmspeichers, der General Purpose Register und des Hardware Stacks Verwendung fanden. Eine erste Designaufgabe realisierte die eigentliche Pipeline durch Zwischenschalten von Pipelineregistern zwischen die einzelnen Verarbeitungsstufen, die jeweils die Ergebnisse der Vorstufe an die nächste Verarbeitungseinheit weiter reichen, aber auch Arbeiten wie das Anhalten der Pipeline oder das Löschen von Pipelineregistern übernehmen. Als Grundstruktur wurde eine fünfstufige Pipeline gewählt mit den Stufen Fetch, Decode, Read, Execute und Write, die sich an den Verarbeitungsschritten der ursprünglichen Architektur orientieren.

Um den bei Pipelinearchitekturen auftretenden Problemen wie Datenhasards und Kontrollflusshasards begegnen zu können, sind aufwändige Zusatzschaltungen notwendig. Eine leistungsfähige Data Hazard Detection Unit erkennt potentiell kritische Befehle und merkt sich ihr Auftreten. Kommt es tatsächlich zu Datenhasards durch Folgebefehle, steuert sie den Eingang der Execute-Stufe, der dann noch nicht gespeicherte Ergebnisse vorheriger Berechnungen erhält (Data Forwarding) und vermeidet so Taktverluste. Nur in wenigen Fällen (z.B. bei I/O-Zugriffen) ist ein Anhalten der Pipeline nötig, um Datenhasards aufzulösen.

Kontrollflusshasards, verursacht durch alle Arten von Verzweigungen, haben großen Einfluss auf die Leistungsfähigkeit von Pipelinearchitekturen und müssen entsprechend behandelt werden. Die gewählte Implementierung führt unbedingte Sprünge, Unterprogrammaufruf und -rückkehr durch entsprechendes Design der Fetch-Stufe ohne Verluste aus. Für Skip-Befehle (bedingte Sprünge) wird angenommen, dass sie den Folgebefehl ausführen und nicht überspringen – eine Art statische Sprungvorhersage. Versuche zeigten, dass dies für 90% der Schleifendurchläufe und etwa 50% der sonstigen Programmstrukturen zutrifft. Computed Jumps und Interrupts verursachen nur zwei Takte Verlust.

Die Pipelineverarbeitung benötigt nur 28% mehr Slices im FPGA und erreicht dabei derzeit (noch unoptimiert) eine Taktfrequenz von 70 MHz bei annähernd 1 CPI, d.h. fast 70 Assembler MIPS. Benchmarks ergaben eine Beschleunigung zwischen min. 2,3 (Floating-Point-Berechnung mit extrem vielen Verzweigungen) und 3,5 (typische Embedded Programme) gegenüber der non-pipelined Version und eine Leistungssteigerung gegenüber Standardprodukten um etwa den Faktor 14.