

Anhand des CISC/RISC basierenden Mikroprozessors mit 4 stufiger Pipeline, in der Siemens 16 bit Mikrocontrollerfamilie, möchte ich jetzt versuchen so eine Pipeline darzustellen:

Befehl_1	Befehl_2	Befehl_3	Befehl_4	Befehl_5	Befehl_6
	Befehl_1	Befehl_2	Befehl_3	Befehl_4	Befehl_5
		Befehl_1	Befehl_2	Befehl_3	Befehl_4
			Befehl_1	Befehl_2	Befehl_3

Die 4 Stufen heißen: **Fetch, Decode, Execute und Write Back**.

Bei 20 MHz Taktfrequenz = 20 MHz CPU Takt dauert eine Stufe 100ns.

Alle Befehle müssen in diesem Echtzeitkorsett abgearbeitet werden, die Verweildauer eines Befehles ist zwar 400ns, aber von außen als Black Box betrachtet, wird alle 100ns (= 1 Maschinenzklus) ein neuer Befehl eingelesen und **alle 100ns verläßt ein Befehl die Pipeline**.

Intern ist es sogar möglich, 32 bit breite Befehle (ca. 7% des Befehls-vorrates) innerhalb 100ns = ein Maschinenzklus abzuarbeiten.

Bei einer 4 stufigen Pipeline ist die Verweildauer eines Befehles also 4 Stufen, d.h. in 1/4 der Zeit wird immer ein neuer Befehl eingelesen und zur gleichen Zeit verläßt ein Befehl die Pipeline, pro Zeiteinheit sind also 4 Befehle gleichzeitig in Bearbeitung.

Würde man so eine Pipeline um einige Stufen erweitern, so nennt man dieses Konzept Superpipelining, d.h. z.Bsp. bei einer 8 stufigen Superpipeline wird in 1/8 der Zeit ein neuer Befehl eingelesen und so weiter.

Ein anderes Konzept möchte ich auch noch erwähnen: Egal ob es sich um CISC oder RISC Architekturen handelt, integriert man auf einen Chip statt einem Mikroprozessor gleich zwei oder mehr vollständige CPUs, so nennt man dieses Konzept superscalar.

Die Steigerung wäre dann eine superscalare superpipelining Architektur, aber jetzt genug davon.

Leider nehmen es viele Halbleiterhersteller mit solchen Definitionen nicht so genau. So findet man am Markt zum Beispiel "RISC"-Architekturen, bei denen man nur eine kleine Einheit des Mikroprozessors doppelt (zum Beispiel eine ALU, um gleichzeitig zwei gleiche Operationen durchführen zu können) aufgebaut hat, und nennt das auch noch scalar, obwohl die CPU nicht zweifach vorhanden ist und dann auch noch RISC. Eine genauerer Umgang mit solchen Ausdrücken wäre angebracht.

Bei der 80C166er Familie wurden auch Verkehrungen für bedingte Sprünge und Loops (Sprungzielberechnung, Jump Cache) getroffen, damit keine Lücken in der Pipeline entstehen.

Die CISC Eigenschaften des Mikroprozessors in der 166er Mikrocontrollerfamilie der Firma Siemens sind die komfortablen Befehle wie sie auch in anderen, CISC basierenden Architekturen existieren (eigentlich sind sie noch viel komfortabler).

Statt dem Flaschenhals "Akkumulator" gibt es je Task 16 Stück 16 bit breite Register mit Akkufunktionalität, d.h. es entfällt das Laden und Entladen von Daten in den Akku.

Es entfallen sämtliche Stack Push und Pop Befehle, da innerhalb eines Maschinenzklus, meistens am Anfang einer neuen Interruptservice-Routine eine komplett neu Registerbank (Context) eingestellt werden kann.

Da in dieser Architektur für jede Interruptquelle ein eigenes Anforderungsflag und eine eigene Interrupteinsprungadresse bereitsteht, entfällt das lästige Abfragen durch Software, welcher Interrupt war es den nun eigentlich, dadurch wird wieder unnötiger Programmcode, der die CPU-Performance sinken läßt, eingespart.

Wird ein Interrupt durch einen der 8 PEC-Kanäle (=quasi DMA) abgefangen, so wird für den Interrupt gar kein Programmcode ausgeführt, es wird auch nicht auf einen Interruptvektor gesprungen, es müssen auch keine Register gesichert werden, das alles beschleunigt die Architektur zu Höchstleistungen.

Beispiele für die komfortablen Befehle:

Indirekte, indizierte Adressierung, indirect mit post increment oder pre decrement, usw.. Schiebe, Rotier und Normalisierbefehle bis zu 16 bits innerhalb eines Maschinenzklus = 100 ns. Zugriff auf ein Byte eines Wortes mit logisch Und (um bits gezielt zu löschen) und logisch Oder (um bits gezielt zu setzen) innerhalb von 100ns. Analog ist auch der Zugriff auf SFRs.

SFR (Special Function Register) sind Register, die die Schnittstelle Mikroprozessor-On Chip Peripherie darstellen.

Durch Ändern des Inhaltes eines SFR kann zum Beispiel die Betriebsart des On Chip AD-Wandlers geändert werden (zum Bsp.: von "ein Kanal einmal wandeln" auf "ein Kanal dauernd wandeln"), oder es kann eine serielle Schnittstelle initialisiert werden.

Parallele Ports: Einlesen eines parallelen Ports, Operation, Ausgabe des neuen Wertes auf den gleichen Port) dauert 100ns.

Wie schon gesagt, **der Mikrocontroller muß schnell auf die Umgebung reagieren können und der Befehlssatz ist für diese Aufgaben optimiert.**

Die größte Kopfarbeit war es, so komfortable Befehle (CISC-typische) so in Hardware zu gießen, damit diese auch in das Pipelining Schema, also in das Echtzeitkorsett passen, und dadurch RISC-typisch abgearbeitet werden können.

Bei der Siemens 16 bit Mikrocontrollerfamilie ist das gelungen.

Durch dieses Konzept wird eine Rechenleistung der 16bit CPU erreicht, die weit über der Rechenleistung von 32 bit CISC basierenden Prozessoren liegt.

Das entscheidende Kriterium für die Performance eines Mikrocontrollers ist die Architektur (die Taktfrequenz ist von der verfügbaren Technologie des Halbleiterherstellers abhängig).

Damit die Pipeline des CISC/RISC (CISC bedeutet also komfortabler Befehlssatz, RISC bedeutet Abarbeitung der komfortablen Befehle in einer 4 stufigen Pipeline) basierenden Mikroprozessors im Mikrocontroller immer mit Befehlen gefüllt wird und keine Lücken auftreten, um also einen Flaschenhals in der Architektur des Bausteins zu vermeiden, hat man **on Chip 5 parallele Bussysteme** (bis zu 32 bit breit) implementiert, auf denen parallel Peripherals, Interrupt-Controller, PEC und CPU zugreifen können, auch ein paralleler Zugriff auf die gleiche Speicherzelle im interne RAM ist möglich.

Hinweis: bei anderen, am Markt erhältlichen Mikrocontrollern müssen alle Einheiten über einen einzigen Bus Daten austauschen, für diese Architekturen trifft der Begriff "Flaschenhals" zu.

Die Leistungsfähigkeit des gesamten Single Chip Computers ist dadurch begrenzt.

Der **Buscontroller** der 166er Familie erlaubt eine flexible Anpassung an alle mögliche **externen Busse**.

Adreßraum: **64 kByte bis 16 MByte** (Architektur erlaubt größeren Adreßbereich, Mikrocontroller sollen jedoch klein sein, so wirkt die Anzahl der zur Verfügung gestellten Pins limitierend für den physikalischen Adreßraum.

Datenleitungen:

8 bit oder 16 bit gemultiplext oder nicht gemultiplext.

Zusätzliche Einstellmöglichkeiten im 50ns Raster von Waitstates, Tri-State-Time, Memory Cycle Time, Read/Write Delay und ALE-Verlängerung

Chipselectleitungen, Ready-Leitungen, Bootstrap-Loader und Mehrprozessorsystem mit HOLD, HLDA und BREQ Leitungen.

Die **Segmentierung** des Adreßraumes verhindert zeitraubende zusätzliche Zugriffe auf den externen Bus, dies wäre beim mitschleppen absoluter Adressen der Fall und steht dem Gedanken, der Mikrocontroller soll so schnell wie möglich **auf die Umgebung reagieren**, entgegen.

□