Was ist Pipelining ?

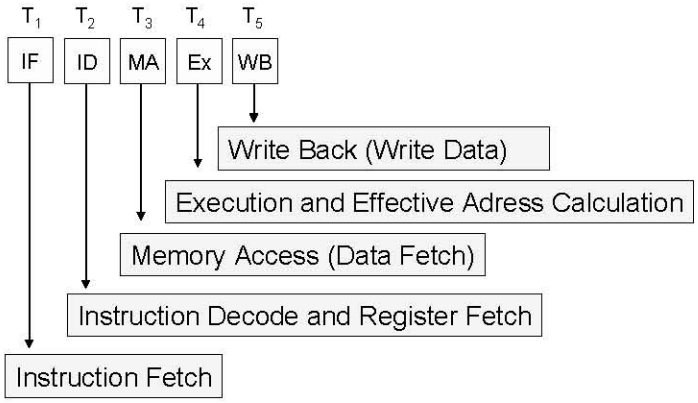
Pipelineverarbeitung ist ein Verfahren der Parallelverarbeitung, bei dem auf mindestens einer Verarbeitungsebene (z.B. Programmanweisungen, Maschinenbefehle) die jeweils durchzuführenden Operationen so unterteilt sind, dass sie in Folge durch unabhängige, spezialisierte Teilwerke taktsynchron verarbeitet werden können.

Zur vollständigen Ausführung einer Operation werden alle Teilwerke sequentiell so durchlaufen, dass sich gleichzeitig mehrere Teiloperationen, zeitlich gegeneinander versetzt, in Bearbeitung befinden.

Brockhaus Enzyklopädie, 19. Auflage, Band 17

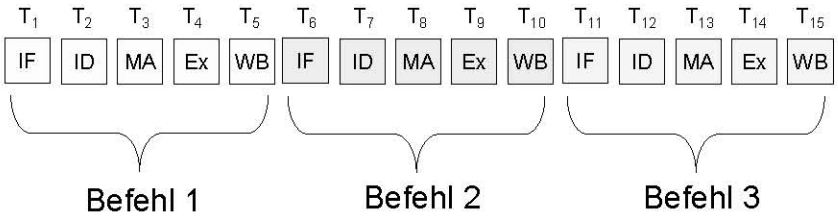
Sequentielle Abarbeitung von Befehlen

Abarbeitung von Befehlen im sequentiellen Modus

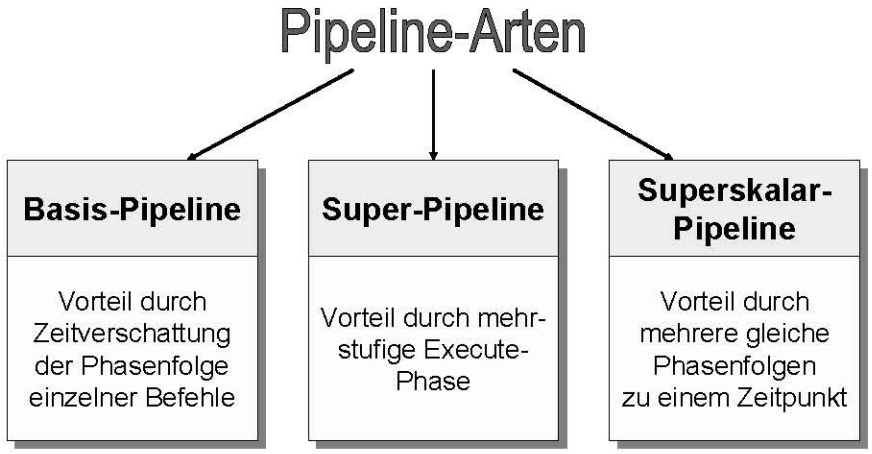


Sequentielle Abarbeitung von Befehlen

Abarbeitung von Befehlen im sequentiellen Modus



Pipeline-Arten



Basis - Pipeline

Abarbeitung von Befehlen in einer Pipeline

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}
IF	ID	MA	Ex	WB	IF	ID	MA	Ex	WB	...
	IF	ID	MA	Ex	WB	IF	ID	MA	Ex	...
		IF	ID	MA	Ex	WB	IF	ID	MA	...
			IF	ID	MA	Ex	WB	IF	ID	...
				IF	ID	MA	Ex	WB	IF	...

Je nach Länge der Pipeline sind Werte von $\overline{CPI} < 1$ möglich

Pipeline - Probleme

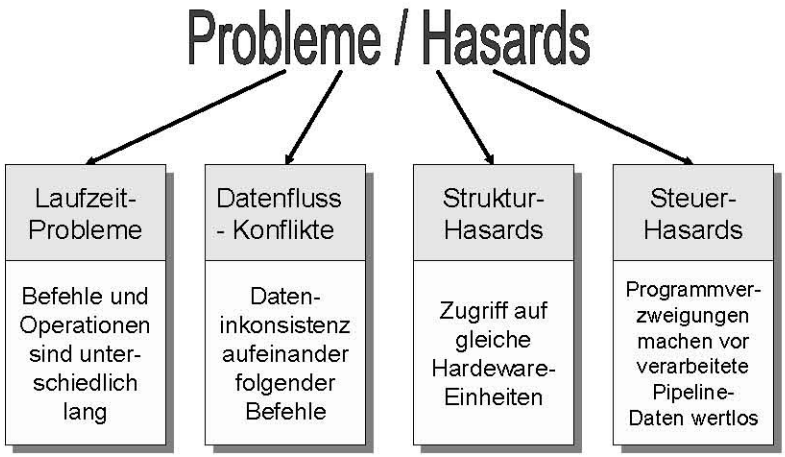
T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
IF	ID	MA	Ex	WB	IF	ID	MA	Ex
	IF	ID	MA	Ex	WB	IF	ID	MA
		IF	ID	MA	Ex	WB	IF	ID
			IF	ID	MA	Ex	WB	IF
				IF	ID	MA	Ex	WB

Probleme

- Unterschiedliche Verarbeitungszeiten
- Mehrfachbelegung von Ressourcen
- Programmfluss-änderungen

! Hasards !

Pipeline – Probleme (Hasards)



Pipelining - Laufzeitprobleme

Problem: Bei komplexen Befehlen, Adressberechnungen, Operationen im Speicher ist die Ausführungszeit unterschiedlich lang.

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}
IF	ID	MA	Ex	**Ex**	WB	stall	IF	ID	MA	...
	IF	ID	MA	stall	Ex	**Ex**	WB	IF	ID	...
		IF	ID	stall	MA	stall	Ex	WB	IF	...
			IF	stall	ID	stall	MA	Ex	WB	...
				stall	IF	stall	ID	MA	Ex	...

Lösung: Pipeline so lange anhalten (Pipeline Stall), bis langsamere Teiloperation abgeschlossen ist; „NOP“ einführen oder mehrere parallele Funktionseinheiten

Pipelining – Datenflusskonflikte

Problem: Bei aufeinander folgenden Befehlen, die über gleiche Speicher / Register die Daten beziehen, entsteht eine Dateninkonsistenz

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	.....
IF	ID	MA	Ex	WB	IF	ID	MA	Ex	WB	...	mov ax,07
	IF	ID	stall	stall	MA	Ex	WB	IF	ID	...	.....
		IF	stall	stall	ID	MA	Ex	WB	IF	...	mov bx,0F
			stall	stall	IF	ID	MA	Ex	WB	...	mov dx,300
				stall	WB	IF	ID	MA	Ex	...	add ax,2
											out ax
											.....
											Es wird **7** und nicht
											9 ausgegeben

Lösung: Pipeline anhalten (Pipeline Stall), bis Ergebnis geschrieben oder Befehlsfolge ändern (statisches Befehls-Scheduling)

Computertechnik
Pipelining

Pipelining – Datenflusskonflikte

Problem: Bei aufeinander folgenden Befehlen, die über gleiche Speicher / Register die Daten beziehen, entsteht eine Dateninkonsistenz

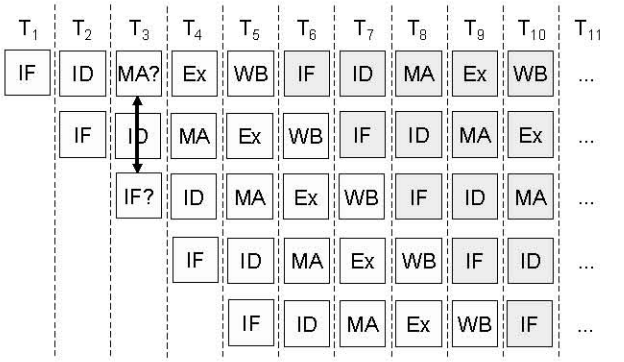
T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	.....
IF	ID	MA	Ex	WB	IF	ID	MA	Ex	WB	...	mov ax,07
	IF	ID	stall	stall	MA	Ex	WB	IF	ID	...	.....
		IF	stall	stall	ID	MA	Ex	WB	IF	...	**add ax,2**
			stall	stall	IF	ID	MA	Ex	WB	...	mov bx,0F
				stall	WB	IF	ID	MA	Ex	...	mov dx,300
											out ax
											.....
											Es wird **7** und nicht
											9 ausgegeben

Lösung: Pipeline anhalten (Pipeline Stall), bis Ergebnis geschrieben oder Befehlsfolge ändern (statisches Befehls-Scheduling)

Prof. Dr.-Ing. W. Schmalwasser *Hochschule Mittweida - Fachbereich: IT & ET*

Pipelining – Strukturkonflikte

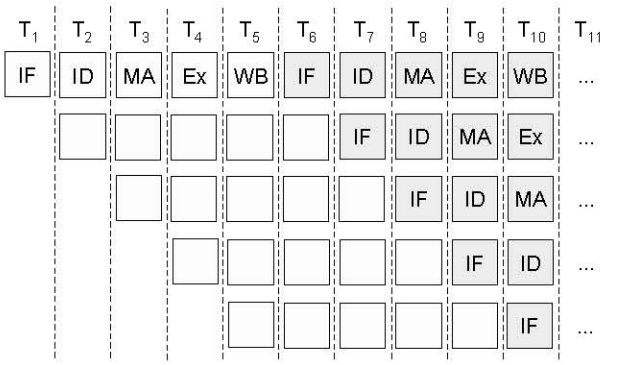
Problem: Bei gleichzeitigem Zugriff mehrerer Verarbeitungsstufen auf gleiche Hardwareeinheiten entstehen Strukturkonflikte (Strukturengpässe).



Lösung: zusätzliche Funktionseinheiten (AGU); zusätzliche Register; Havard-Struktur; getrennte Caches für Daten und Befehle

Pipelining – Steuerkonflikte

Problem: Programmverzweigungen machen vor verarbeitete Pipeline-Daten wertlos.

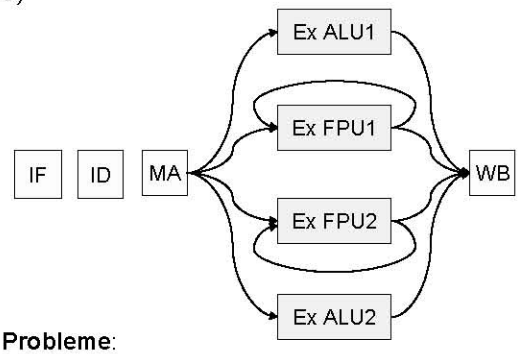


Lösung: static scheduling der Befehlsfolge während der Compilierung (Loop unrolling); unbedingte Sprünge verwenden; spekulative Sprungvorhersage; zwei Pipelines (für beide Wege)

Super-Pipeline

Problem: Unterschiedliche Verweildauer in der EX-Phase

Lösung: Parallele Anordnung mehrerer Ausführungseinheiten (ALU, AGU, FPU)



Auftretende Probleme:

- **Steuerhasards**: Durch mehrfach vorhandene Funktionseinheiten
- **Datenhasards**: Durch unterschiedliche Dauer der einzelnen Befehle

Superscalar-Pipeline

Problem: Verringerung der Ausführungszeiten (kleines CPI-Verhältnis)

Lösung: Parallele Anordnung mehrerer Pipelines

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
IF	ID	MA	Ex	WB	IF	ID	MA	Ex
IF	ID	MA	Ex	WB	IF	ID	MA	Ex
	IF	ID	MA	Ex	WB	IF	ID	MA
	IF	ID	MA	Ex	WB	IF	ID	MA
		IF	ID	MA	Ex	WB	IF	ID
		IF	ID	MA	Ex	WB	IF	ID
			IF	ID	MA	Ex	WB	IF

Wichtig:

- loop unrolling
- Nutzbarmachung von Nebenläufigkeiten