

Übungen zu
Rechnerarchitektur
– Scoreboarding and Tomasulo's
Algorithm –

Sommersemester 2017

Jian-Jia Chen

Folien von Michael Engel, Björn Bönninghoff,
und Olaf Neugebauer

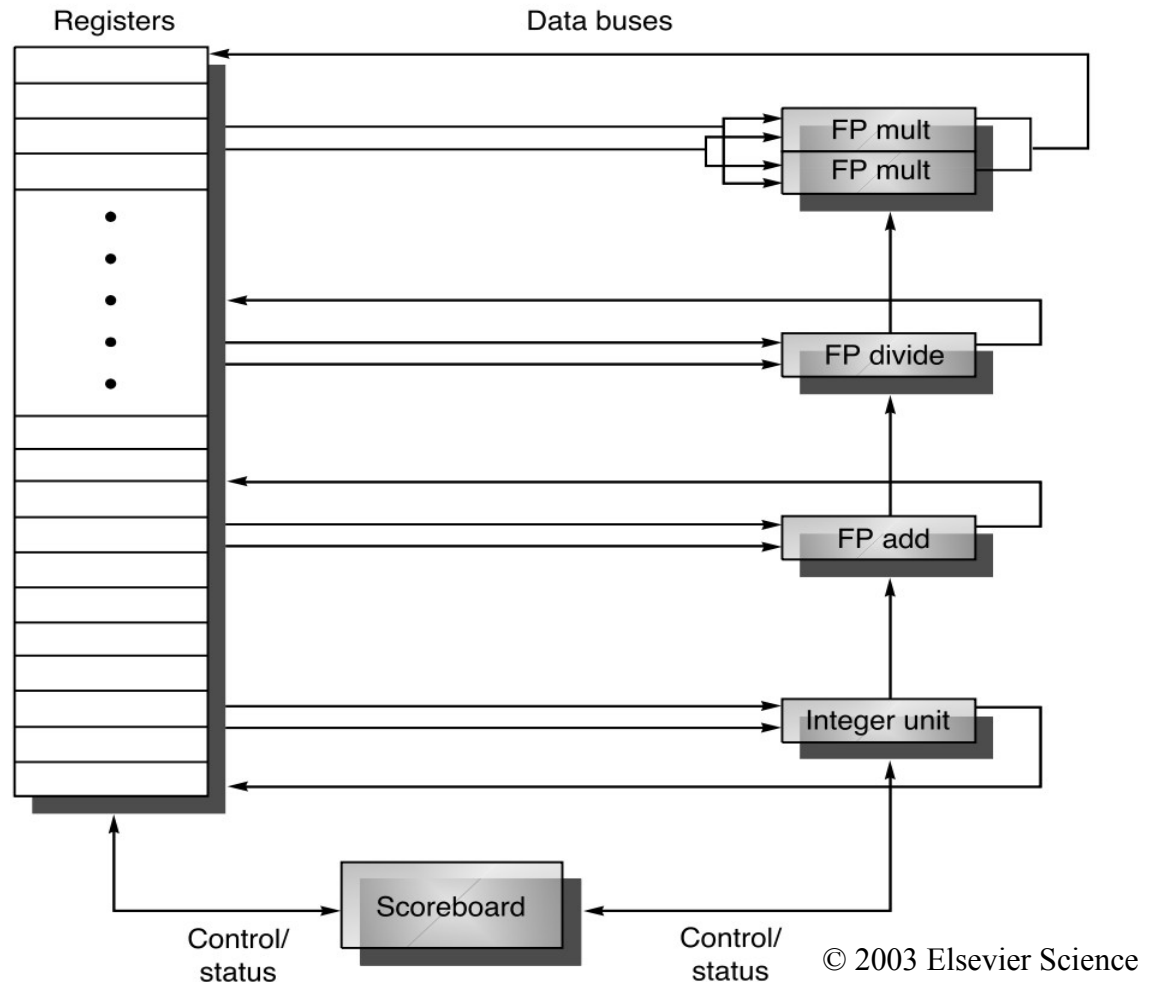
Technische Universität Dortmund

Lehrstuhl Informatik 12

Entwurfsautomatisierung für Eingebettete Systeme

Scoreboarding

- **Struktur einer MIPS-Pipeline mit Scoreboard**
- **Scoreboard kontrolliert funktionelle Einheiten**
- **Bussysteme erforderlich**



Funktion des *Scoreboards*

- **Scoreboard kontrolliert Befehlsausführung ab Befehlsholphase (IF) in 4 Phasen:**
- **1. *Issue***
 - Befehl wird ausgegeben falls
 - benötigte funktionale Einheit ist frei und
 - kein anderer aktiver Befehl (d.h. im *Scoreboard*) schreibt gleiches Zielregister ➡ verhindert WAW-Konflikte
 - Auswertung beginnt nicht, Befehl nur an funktionelle Einheit!
 - Ersetzt Teil der ID-Phase in klassischer Pipeline
 - Anhalten der *Pipeline (stall)* erfolgt:
 - Bei strukturellen Konflikten oder
 - WAW-Konflikt
 - ➡ kein weiterer Befehl wird ausgegeben!

Funktion des *Scoreboards*

■ 2. *Read Operands (RO)*

- *Scoreboard* beobachtet Verfügbarkeit von Quelloperanden
 - ☞ sind verfügbar, wenn kein früher ausgegebener Befehl, der noch aktiv ist, diese schreiben wird (Datenflussprinzip) □
- Sobald Operanden verfügbar: *Scoreboard* signalisiert betreffender fkt. Einheit Beginn der Ausführungsphase
- RAW-Konflikte (d.h. echte Datenabhängigkeiten) werden hier dynamisch aufgelöst
- Befehle werden ggf. *out-of-order* zur Ausführung gebracht (*Issue* erfolgt nicht strikt *in-order*!) □
- *Scoreboarding* verwendet kein *Forwarding* (Komplexität!)
 - ☞ Operanden werden bei Verfügbarkeit aus Register gelesen

Funktion des *Scoreboards*

- **3. *Execution***
 - Funktionelle Einheit beginnt Befehlsausführung nach Erhalt der Operanden
 - *Scoreboard* wird informiert, sobald Ergebnis vorliegt (noch in funktioneller Einheit, d.h. noch nicht im Register!) □
 - Entspricht „klassischer“ EX-Phase, ggf. >1 Zyklen lang
- **4. *Write Results (WR)***
 - Ausführungseinheit meldet abgeschlossene Berechnung an *Scoreboard*
 - *Scoreboard* erteilt Berechtigung zum Abspeichern, wenn Zielregister nicht von aktiver Operation (in RO) noch benötigt ☞ Lösung von WAR-Konflikten
- **Abspeichern macht ggf. Befehl ausführungsbereit**
- **WR und RO dürfen nicht überlappen!**

Datenstrukturen

1. **Befehlsstatus:** Phase (*Issue, ... Write Results*), in der sich Befehl befindet
2. **Status der funktionelle Einheiten**, unterteilt in:
 - *Busy* : Einheit arbeitend oder nicht
 - *Op* : Aktuelle Operation
 - *Fi* : Zielregister
 - *Fj, Fk* : Quellregister
 - *Qj, Qk* : fkt. Einheiten, die ggf. Quellregisterinhalt erzeugen
 - *Rj, Rk* : *Flag*, ob Quelloperanden verfügbar aber, noch nicht in *Read Operands* gelesen
5. **Register-Ergebnis-Status**
Welche funktionelle Einheit verfügbare Register schreibt (ggf. leer)

Scoreboarding: Takt +0

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	
MUL.D F0,F2,F4	+			
SUB.D F8,F6,F2	+			
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2				

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	ja	Load	F2	R3				nein	
Mult1	ja	Mult	F0	F2	F4	Int.		nein	ja
Mult2	nein								
Add	ja	Sub	F8	F6	F2		Int.	ja	nein
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1	Int			Add	Div

Scoreboarding: Takt +1

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+			
SUB.D F8,F6,F2	+			
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2				

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein	-	-	-				-	
Mult1	ja	Mult	F0	F2	F4	-		ja	ja
Mult2	nein								
Add	ja	Sub	F8	F6	F2		-	ja	ja
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1	-			Add	Div

Scoreboarding: Takt +2

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+		
SUB.D F8,F6,F2	+	+		
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2				

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	ja	Mult	F0	F2	F4			nein	nein
Mult2	nein								
Add	ja	Sub	F8	F6	F2			nein	nein
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1				Add	Div

Scoreboarding: Takt +3

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+	1	
SUB.D F8,F6,F2	+	+	1	
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2				

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	ja	Mult	F0	F2	F4			nein	nein
Mult2	nein								
Add	ja	Sub	F8	F6	F2			nein	nein
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1				Add	Div

Scoreboarding: Takt +4

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+	2	
SUB.D F8,F6,F2	+	+	+	
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2				

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	ja	Mult	F0	F2	F4			nein	nein
Mult2	nein								
Add	ja	Sub	F8	F6	F2			nein	nein
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1				Add	Div

Scoreboarding: Takt +5

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+	3	
SUB.D F8,F6,F2	+	+	+	+
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2				

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	ja	Mult	F0	F2	F4			nein	nein
Mult2	nein								
Add	nein	-	-	-	-			-	-
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1				-	Div

Scoreboarding: Takt +6

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+	4	
SUB.D F8,F6,F2	+	+	+	+
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2	+			

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	ja	Mult	F0	F2	F4			nein	nein
Mult2	nein								
Add	ja	Add	F6	F8	F2			ja	ja
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1			Add		Div

Scoreboarding: Takt +7

Befehls-Status

<i>Befehl</i>	<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+	+	
SUB.D F8,F6,F2	+	+	+	+
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2	+	+		

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	ja	Mult	F0	F2	F4			nein	nein
Mult2	nein								
Add	ja	Add	F6	F8	F2			nein	nein
Divide	ja	Div	F10	F0	F6	Mult1		nein	ja

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>	Mult1			Add		Div

Scoreboarding: Takt +8

Befehls-Status

Befehl	Issue	Read Op.	Exec. complete	Write
L.D F6,32(R2)	+	+	+	+
L.D F2,96(R3)	+	+	+	+
MUL.D F0,F2,F4	+	+	+	+
SUB.D F8,F6,F2	+	+	+	+
DIV.D F10,F0,F6	+			
ADD.D F6,F8,F2	+	+	1	

Status der Funktionseinheiten

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	nein								
Mult1	nein	-	-	-	-			-	-
Mult2	nein								
Add	ja	Add	F6	F8	F2			nein	nein
Divide	ja	Div	F10	F0	F6	-		ja	ja

Register-Ergebnis-Status

	F0	F2	F4	F6	F8	F10
Einheit	-			Add		Div

Scoreboarding: Takt +9

Befehls-Status

Befehl		Issue	Read Op.	Exec. complete	Write
L.D	F6,32(R2)	+	+	+	+
L.D	F2,96(R3)	+	+	+	+
MUL.D	F0,F2,F4	+	+	+	+
SUB.D	F8,F6,F2	+	+	+	+
DIV.D	F10,F0,F6	+	+		
ADD.D	F6,F8,F2	+	+	+	

Status der Funktionseinheiten

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	nein								
Mult1	nein	-	-	-	-			-	-
Mult2	nein								
Add	ja	Add	F6	F8	F2			nein	nein
Divide	ja	Div	F10	F0	F6			nein	nein

Register-Ergebnis-Status

	F0	F2	F4	F6	F8	F10
Einheit	-			Add		Div

Scoreboarding: Takt +10

Befehls-Status

<i>Befehl</i>		<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D	F6,32(R2)	+	+	+	+
L.D	F2,96(R3)	+	+	+	+
MUL.D	F0,F2,F4	+	+	+	+
SUB.D	F8,F6,F2	+	+	+	+
DIV.D	F10,F0,F6	+	+	1	
ADD.D	F6,F8,F2	+	+	+	+

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	nein								
Mult2	nein								
Add	nein	-	-	-	-			-	-
Divide	ja	Div	F10	F0	F6			nein	nein

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>				-		Div

Scoreboarding: Takt +11

Befehls-Status

<i>Befehl</i>		<i>Issue</i>	<i>Read Op.</i>	<i>Exec. complete</i>	<i>Write</i>
L.D	F6,32(R2)	+	+	+	+
L.D	F2,96(R3)	+	+	+	+
MUL.D	F0,F2,F4	+	+	+	+
SUB.D	F8,F6,F2	+	+	+	+
DIV.D	F10,F0,F6	+	+	2	
ADD.D	F6,F8,F2	+	+	+	+

Status der Funktionseinheiten

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
Integer	nein								
Mult1	nein								
Mult2	nein								
Add	nein								
Divide	ja	Div	F10	F0	F6			nein	nein

Register-Ergebnis-Status

	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>
<i>Einheit</i>						Div

Scoreboarding: folgende Takte *(nur DIV wird ausgeführt)*

Befehls-Status

Befehl		Issue	Read Op.	Exec. complete	Write
L.D	F6,32(R2)	+	+	+	+
L.D	F2,96(R3)	+	+	+	+
MUL.D	F0,F2,F4	+	+	+	+
SUB.D	F8,F6,F2	+	+	+	+
DIV.D	F10,F0,F6	+	+	+	
ADD.D	F6,F8,F2	+	+	+	+

Status der Funktionseinheiten

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	nein								
Mult1	nein								
Mult2	nein								
Add	nein								
Divide	ja	Div	F10	F0	F6			nein	nein

Register-Ergebnis-Status

	F0	F2	F4	F6	F8	F10
Einheit						Div