
Rechnerarchitektur SS 2016

Exercises: Scoreboarding and Tomasulo's Algorithm

Jian-Jia Chen

TU Dortmund

to be discussed on June, 14, 2016

Scoreboardings

Im nächste Folien finden Sie ein Beispiel zu den gespeicherten Informationen innerhalb der Scoreboarding-Datenstrukturen, für einen aus der Vorlesung bekannten Beispielcode. Konkret ist in den Tabellen folgende Situation festgehalten:

- erste Lade-Instruktion ist vollständig verarbeitet und hat ihr Ergebnis in der *write result*-Phase geschrieben.
- zweite Lade-Instruktion hat die Ausführungsphase abgeschlossen und wartet auf das Schreiben ihres Ergebnisses.
- restliche Instruktionen befinden sich vor bzw. in der *issue*-Phase.

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

Exercise - Scoreboardings

Führen Sie die Programmausführung für die nächsten 10 Takte fort und stellen Sie für jeden Takt den aktuellen Stand der Scoreboarding-Datenstrukturen am Taktende dar. Benutzen Sie hierfür die Vorlage *scoreboarding.ppt* von der Webseite der Übung. Verwenden Sie für diese Aufgabe folgende Latenzzeiten der *execute*-Phase:

- Addition: 2 Taktzyklen
- Multiplikation: 5 Taktzyklen
- Division: 8 Taktzyklen

Die Latenzzeiten sind so zu verstehen, daß die Execute-Phase exakt die angegebene Anzahl Takte benötigt um zu terminieren, d.h. wenn beispielsweise eine Addition in Takt x gestartet wird endet sie am Ende von Takt $x + 1$. *Forwarding* von Ergebnissen wird nicht unterstützt, d.h. Ergebnisse die in der Write-Results-Phase geschrieben werden, können erst im nächsten Takt in der Read-Operands-Phase gelesen werden. Außerdem werden keine Operanden gepuffert, d.h. wenn ein Operand verfügbar ist, ein anderer aber noch nicht, werden beide erst gelesen wenn auch beide verfügbar sind.

Tomasulo-Algorithmus

Konkret ist in den Tabellen (von tomasulo_vorlage_neu.pdf) folgende Situation festgehalten:

- erste Lade-Instruktion ist vollständig verarbeitet und hat ihr Ergebnis in der *write result*-Phase geschrieben.
- zweite Lade-Instruktion hat die Ausführungs-Phase abgeschlossen und wartet auf das Schreiben ihres Ergebnisses.
- restliche Instruktionen befinden sich in der *issue*-Phase.

Exercise - Tomasulo-Algorithmus

Verwenden Sie für diese Aufgabe folgende Latenzzeiten (der *EX*-Phase):

Instruktionstyp	Latenzzeit
Laden	1 Taktzyklus
Addition	1 Taktzyklus
Multiplikation	3 Taktzyklen
Division	6 Taktzyklen

Die Latenzzeiten sind so zu verstehen, daß die Execute-Phase auf der entsprechenden Verarbeitungseinheit exakt die angegebene Anzahl Takte benötigt um zu terminieren, d.h. wenn beispielsweise eine Multiplikation in Takt x gestartet wird endet sie am Ende von Takt $x + 2$. Wenn ein Ergebnis in der *Write*-Phase geschrieben wird, kommt es noch im selben Takt bei allen darauf wartenden Reservation Stations an, und diese fangen im darauf folgenden Takt mit der Abarbeitung der Instruktion an. Pro Takt kann über den *Common Data Bus* nur ein einzelnes Ergebnis übertragen werden. Wenn mehrere Ergebnisse im selben Takt übertragen werden sollen setzt sich immer das Ergebnis durch, dessen Instruktion zuerst die Issue-Phase durchlaufen hat (*in-order Issue*).

Tomasulo-Algorithmus: Takt

▪ Befehls-Status

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

▪ Status der Reservation Stations

<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	<i>A</i>
Load1							
Load2							
Add1							
Add2							
Add3							
Mult1							
Mult2							

▪ Register-Ergebnis-Status

<i>Qi</i>	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>...</i>

Vergleichen Scoreboardings und Tomasulo

Erklären Sie kurz den Unterschied zwischen den erwähnten Verfahren des *Scoreboardings* und dem Verfahren von *Tomasulo*. Welche Unzulänglichkeiten, die beim Scoreboarding auftreten, werden von dem Verfahren von Tomasulo beseitigt? Warum?