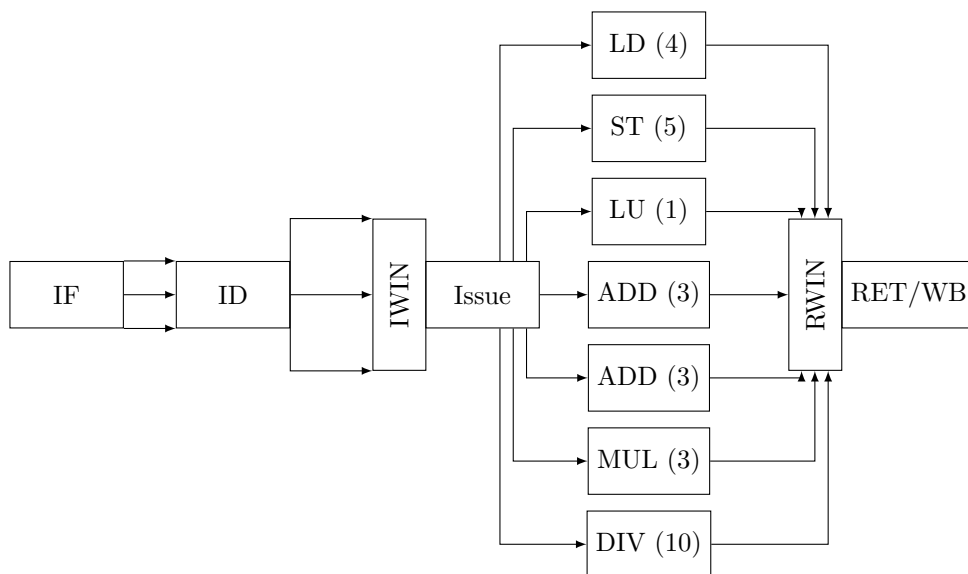


Bitte beachten Sie die allgemeinen Hinweise auf Übungszettel 1

**Abweichende Bewertung:
≥ 2 von 3 Punkten führen zum Bestehen**

Superskalare Pipeline

Gegeben sei folgende superskalare Pipeline:



1. In der ersten Stufe (Instruction Fetch, IF) können pro Takt drei Befehle gleichzeitig geladen werden.
2. Die zweite Stufe (Instruction Decode, ID) kann ebenfalls pro Takt drei Befehle gleichzeitig dekodieren.
3. Dekodierte Befehle befinden sich anschließend architekturell bedingt mindestens einen Takt im Instruction Window (IWIN), welches maximal fünf Instruktionen vorhalten kann. Hier werden sofern nötig auch Operanden parallel geladen (Operand Fetch, OF).
4. Danach werden sie durch die Issue-Einheit auf die entsprechenden (freien) Ausführungseinheiten aufgeteilt. Die Issue-Einheit benötigt dafür keine zusätzliche Zeit und kann bis zu drei Befehle aus IWIN gleichzeitig verteilen (Tripple Issue Unit).

5. Die Pipeline besitzt sieben verschiedene Ausführungs-Einheiten (EXE):
- (a) Load (LD) zum Laden von Daten aus dem Speicher.
 - (b) Store (ST) zum Speichern von Daten in den Speicher.
 - (c) Logical Unit (LU) für logische und binäre Operationen, unter anderem Schiebe-Operationen.
 - (d) Zwei Addierer (ADD) für Addition und Subtraktion.
 - (e) Multiplikator (MUL) für Multiplikationen.
 - (f) Dividierer (DIV) für Divisionen.
6. Anschließend stellt die Retirement/Write-Back-Einheit (RET/WB) die korrekte Reihenfolge der Befehle wieder her und schreibt eventuell das Ergebnis in die Ergebnis-Register. Der dazu nötige Puffer (Retire Window, RWIN) ist ausreichend groß. Einkommende Ergebnisse werden soweit möglich sofort verarbeitet, müssen also nicht zwingend im RWIN warten.
7. Es gibt kein Forwarding/Shortcuts zwischen den Ausführungs-Einheiten. Ein Ergebnis kann allerdings weiter verwendet werden, sobald es mind. 1 Takt in RWIN vorlag.

Die benötigten Laufzeiten (Takte) und Kapazitäten der einzelnen Einheiten lauten wie folgt.

ADD A, B bedeutet im Folgenden $A = A + B$.		Einheit	Takte	Kapazität
		IF	1	max. 3 Befehle / Takt
		ID	1	max. 3 Befehle / Takt
		IWIN	mind. 1	5 Plätze
		Issue	0	3 Befehle / Takt
1 LD A, [0x0AFFE0]		LD	4	1 Einheit
2 SHL B, 3		ST	5	1 Einheit
3 ST B, [0xCAFFEE]		LU	1	1 Einheit
4 ADD E, F		ADD	3	2 Einheiten
5 SHR F, 1		MUL	5	1 Einheit
6 ADD F, 7		DIV	10	1 Einheit
7 SUB B, 3		RWIN	mind. 0	10 Plätze
8 SHL G, 1		RET/WB	1	10 Befehle / Takt
9 MUL G, G				

Aufgaben

- Zeigen Sie eine mögliche Belegung der in-/out-of-order Ausführung des obigen Code-Beispiels anhand nachfolgender Tabelle auf. Wenden Sie pro Takt *first-come, first-served (FCFS)* in der Issue Einheit an. Tragen Sie dazu die Zeilennummern der entsprechenden Befehle ein. Sie können annehmen, dass sowohl IWIN als auch alle EXE Einheiten zu Beginn leer sind.
- Welchen Speed-up haben Sie mit Einführung dieser Pipeline gegenüber einer skalaren Pipeline mit nur einer (mächtigen) EXE Einheit erreicht? Nehmen Sie dazu an, Sie benötigen die gleiche Taktzahl pro Befehl wie bei dieser superskalaren Pipeline.
- Warum ist Ihre Pipeline nicht optimal ausgelastet? Welche weiteren Maßnahmen führen zu einer (signifikant) größeren Beschleunigung?

Bei Fragen zur Tabelle wenden Sie sich bitte an Ihren Tutor.

Übung 11

Rechnerarchitektur

Wintersemester 18/19

Prof. Dr.-Ing. Jochen Schiller

Ausgabe 11.01.2019

Abgabe 25.01.2019, 10:15 (s.t.)

