

Informatik 3, Ubungsserie 2

Side Notes

Komponenten

- Akkumulator ~ spezielles Register, in welchem Berechnungen der ALU gespeichert werden (oft direkt mit ALU verbunden)
- Befehlszähler ~ spezielles Register, in welchem die Speicheradressen der auszuführenden Befehle stehen. Wird nach Ausführung von Befehlen inkrementiert, so dass der Pointer auf der Speicheradresse liegt, das den nächsten auszuführenden Befehl enthält

Befehle

- Store / Load Befehle = { Arithmetisch-Logische Fkt. (ALU), Speichern und Laden v. Daten in/aus Register (oder Speicher)}
- Sprung-Befehl = {}
 - ALU führt hier einen Vergleich durch (i.d.R.)

1. Gegeben sei ein Prozessor mit einer Taktzykluszeit von 1.25 GHz und einem CPI-Wert von 1.45 (der Prozessor verfügt über keine Pipeline). Ein Programm benötigt zur Ausführung 150'000 Befehle.

a) Wie lang ist die ungefähre Ausführungszeit des Programms? Lösung: (3 Punkte)

$$1.25 \text{ GHz} = 1250000000/\text{s} \Rightarrow 1/1,25/10^9 \text{s} = 8 \cdot 10^{-10} \text{s} = 0,000\ 000\ 000\ 8 \text{s} = 8 \text{ns}$$
$$t = 8 \text{ns} \cdot 150000 \cdot 1,45 = \mathbf{174 \text{ms}}$$

b) Wieso ist der berechnete Wert nur ein Näherungswert? Lösung: (2 Punkte) Weil es sich beim CPI-Wert auch nur um einen Durchschnittswert handelt. Zudem ist es gar nicht möglich, dass jeder der Befehle tatsächlich genau 1.45 Zyklen zur Berechnung benötigt, da für jeden Befehl nur ganze Zyklen verwendet werden können. Die Wahrscheinlichkeit, dass sich die erwartete Ausführungszeit mit der tatsächlichen deckt, konvergiert daher gegen Null (0)! ### c) Der Prozessor wird durch einen leistungsfähigeren Prozessor mit 0.4 ns Taktzykluszeit und einem CPI-Wert von 1.8 ersetzt. Wie lang ist nun die Ausführungszeit des Programms? $t = 0,4 \text{ns} \cdot 150000 \cdot 1,8 = \mathbf{48 \text{ms}}$ ### d) Der Prozessor (von c) wird um 10% übertaktet („overclocking“). Die erzielte Leistungssteigerung beträgt in der Realität aber nur knapp 5%. Wieso? —TDB

2. Gegeben sei ein einfacher Prozessor ohne Pipelining mit einer Wortbreite von 2 Byte (für Daten und Befehle).

a) Welchen Wert beinhaltet der Befehlszähler jeweils nach Ausführung der jeweiligen Befehle der folgenden Befehlssequenz (der Initialwert sei 24 048 für den ersten Befehl): Ladebefehl, Ladebefehl, Addition, unbedingter Sprung um -12, Speicherbefehl, unbedingter Sprung um +8, Addition ... ?

Initialwert -> 24 048
1. Ladebefehl -> 24 050
2. Ladebefehl -> 24 052
3. Addition -> 24 054
4. Sprung -12 -> 24 042
5. Speichern -> 24 044
6. Sprung +8 -> 24 052
7. Addition -> 24 054

b) Was sehen Sie als Informatiker sofort?

Die Befehlssequenz steckt in einer Endlos-Schleife.

3. Gegeben sei ein Prozessor mit 4-stufiger Pipeline (die vier Stufen, wie in der Vorlesung angegeben) und folgender Ausschnitt einer Programm- abfolge: ..., Load, Sprung, Addition, ODER-Operation, Store, Subtraktion, Sprung, AND-Operation,

a) Skizzieren Sie graphisch eine (mögliche) Ausführungsabfolge, unter der Annahme, das beim 1. Sprung zu einer nicht vorhergesehenen Adresse gesprungen wird („branch prediction“ war falsch).

(siehe Bild) ### b) Beschreiben Sie in Ihren Worten, was ein „pipeline flush“ bedeutet. Um die Anzahl der Steuerkonflikte (control hazards) in einer Pipeline zu minimieren, werden Sprunghervorsagen (branch predictions) eingesetzt. Wird ein Sprung falsch geraten, müssen die Befehle, welche in der Zwischenzeit ausgeführt wurden, verworfen werden. Die Pipeline wird also geleert (flushed). Das kostet natürlich Zeit: Je länger die Pipeline, desto mehr Zeit geht verloren.

4. Eine effektive Möglichkeit der Leistungssteigerung bei Prozessoren ist Pipelining.

a) Begründen Sie, warum eine n-stufige Pipeline nicht automatisch zu einer n-fachen Leistungssteigerung führt, selbst wenn es gelingt, die Zykluszeit auf $1/n$ zu reduzieren („perfekte Gleichverteilung“ der Stufen - in der Praxis eigentlich nicht realisierbar).

Dafür gibt es mehrere Gründe: - Längste Pipeline-Stufe bestimmt Zykluszeit der Pipeline (Zykluszeit = $\max(\text{Zeit der Stufen}) + \text{Zusatzaufwand}$) - Es existieren Wartezeiten, weil nicht alle Befehle alle Stufen der Pipeline durchlaufen - Die Komplexität der Hardware ist grösser (d.h. die Pipeline-Stufen sind komplexer und i.d.R. langsamer) - Es gibt eine An- und Auslaufphase, wobei die Pipeline-Stufen nicht immer voll ausgelastet sind - Einzelne Befehle (welche alleine die Pipeline „belegen“) profitieren nicht von einer Pipeline, sondern benötigen sogar länger zur Ausführung als ohne Pipeline - Gewisse Befehle sind auf das Resultat eines anderen Befehls angewiesen (hier muss gewartet werden) - Speicherzugriffe sind gleich schnell, unabhängig ob die zugehörigen Befehle in der Pipeline abgehandelt werden oder nicht

5. Gegeben sei ein Prozessor ohne Pipeline mit der „bekannten“ Befehls- abarbeitung (siehe Vorlesung) und einer Zykluszeit von 20 MHz. Ein Analyse hat ergeben, dass die einzelnen Teilschritte sehr unterschiedliche Zeit erfordern: z. B. „Befehl laden“ 10 ns, „Register lesen“ 3 ns, „Rechenoperation durchführen“ 5 ns, „Speicherzugriff“ 20 ns und „Register schreiben“ 5 ns, ... Sie implementieren denselben Prozessor mit einer 5-stufigen Pipeline (die bisherigen Teilschritte erfordern gleich viel Zeit).

Gegeben

- Zykluszeit = 20 MHz
- Befehl laden 10ns
- Register lesen 3ns
- Rechenoperation durchführen 5ns
- Speicherzugriff 20ns
- Register schreiben 5ns

a) Wie gross ist die Zykluszeit des neuen Prozessors?

$\text{Zykluszeit} = \max(\text{Zeit der Stufen}) + \text{Zusatzaufwand} = 20\text{ns} + \text{Zusatzaufwand}$

b) Um wie viel schneller wird nun ein Befehl maximal ausgeführt?

Lösung: (3 Punkte)

$\text{Zykluszeit vorher} = 10\text{ns} + 3\text{ns} + 5\text{ns} + 20\text{ns} + 5\text{ns} = 43\text{ns}$ Zykluszeit nachher

20ns Zykluszeit vorher / Zykluszeit nachher = $43\text{ns} / 20\text{ns} = \mathbf{2.15}$

c) Um wie viel schneller wird ein Programm maximal ausgeführt?

Unter Idealbedingungen n-Mal Schneller bei einer n-Stufigen Pipeline, also **5-mal** schnelle

d) Wie könnte eine „bessere“ Pipeline-Struktur entwickelt werden?

Durch Umordnen des Codes (Forwarding) oder Vorhersagen (Branch Prediction)

6. (OPTIONAL) Recherchieren Sie die Pipelinestruktur und Kenndaten von zwei aktuellen Mikroprozessoren. Bitte Kenndaten angeben und Struktur skizzieren, inkl. Quellenangabe.