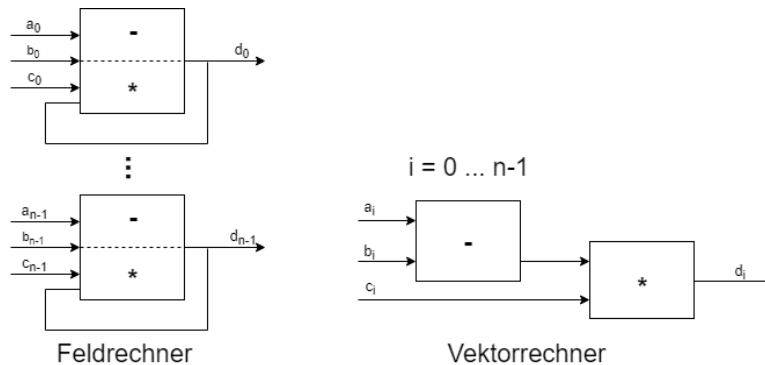


## Lösungen zum Übungszettel 9

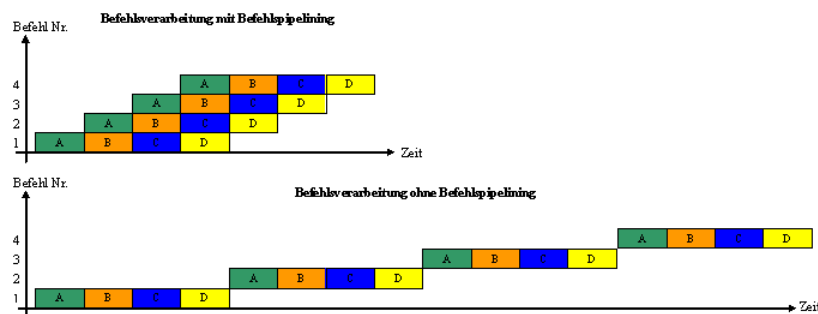
### Aufgabe 1:

a)



Der grundlegende Unterschied dieser beiden Ansätze ist, dass Feldrechner aus  $n$  Rechenelementen (oder Verarbeitungseinheiten) bestehen, die alle auf dem gleichen Instruktionsstrom arbeiten, aber natürlich auf verschiedenen Daten. Im Gegensatz dazu berechnet der klassische Vektorrechner Probleme nach dem Pipeline-Prinzip. Dieses Prinzip der Fließbandverarbeitung ist immer dann anwendbar, wenn komplexe Operationen aus mehreren aufteilbaren Rechenschritten bestehen. Zusätzlich können verschiedene Funktionseinheiten (Addierer, Multiplizierer, ...) aneinandergereiht („chaining“) werden, um auf einem Datenstrom zu arbeiten.

- b) Die Verarbeitung eines Vektors erfolgt nach dem Pipelining-Verfahren. Dafür wird die Operation in möglichst gleichlange Teiloperationen zerlegt, die dann wie in einer Pipeline zeitsequentiell hintereinander in den verschiedenen Stufen bearbeitet werden. Die Ausführung der verschiedenen Teiloperationen überlappen sich dabei für die einzelnen Vektorelemente. Der Gewinn einer Pipelineverarbeitung gegenüber der sequentiellen Verarbeitung ist für lange Vektoren gleich der Stufenzahl der Pipeline [Hockney, Jesshop].



[Von Frank Jacobsen - <http://de.wikipedia.org/wiki/Datei:Befehlspipeline.PNG>, CC BY 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=20551519>]

**Hinweis:** Die Lösungen sollen in PDF-Form, bzw. Code bis zum Montag (10 Uhr) der jeweils folgenden Woche per Mail an [mfriedrichs@techfak.uni-bielefeld.de](mailto:mfriedrichs@techfak.uni-bielefeld.de) abgegeben werden. Zu Beginn des nächsten Übungstermins werden diese in offener Runde vorgestellt und diskutiert.



---

### Aufgabe 2:

- a) LV        V1, RX;            Vektor X laden  
      LV        V2, RY;            Vektor Y laden  
      SUBV     V3, V1, V2;        Vektor Subtraktion  
      LD        F0, a(R0);        Skalar a laden  
      DIVSV    V4, F0, V3;        Skalar-Vektor Division  
      SV        RY, V4;            Resultat in Y speichern
- b) Die Vektorisierung stellt eine horizontale Parallelisierung dar, während die Iteration eine vertikale Parallelisierung darstellt.
- c) Datenunabhängigkeit: Datenflussabhängigkeiten, Antiabhängigkeiten, Ausgabeabhängigkeiten  
      Prozedurale Abhängigkeiten: Fallunterscheidungen, Iterationen)  
      Operationale Abhängigkeiten: Abhängigkeit von der Anzahl der zur Verfügung stehenden Funktionseinheiten

*Daher*: einfache Schleifen, Felder und invariante Ausdrücke auf der rechten Seite und nur Zuweisungen benutzen.

### Aufgabe 3:

- a) Es besteht eine *echte Datenabhängigkeit bzw. Datenflussabhängigkeit* zwischen S1 und S2, eine *Output-Abhängigkeit* zwischen S1 und S4 und eine *Antiabhängigkeit* zwischen aufeinanderfolgenden Iterationen der Anweisungen S2 und S3.
- b) Eine Schleife ist vektorisierbar, wenn nur vorwärts gerichtete Abhängigkeiten bestehen. Nach Vertauschen der Anweisungen S2 und S3 und der Umbenennung der Ausgabevariable von S4 lässt sich die obige Schleife vektorisieren.

Eine Schleife ist parallelisierbar, wenn aufeinanderfolgende Iterationen (Schleifendurchläufe) voneinander unabhängig sind. Zwischen den Anweisungen des Schleifenkörpers, der einen Iterationsschritt darstellt, dürfen Abhängigkeiten bestehen. Das obige Beispiel ist damit nicht parallelisierbar.