Die John-von-Neumann-Architektur mit Johnny entdecken

Material für den Unterricht der Sekundarstufe II (Grund- und Leistungskurs)

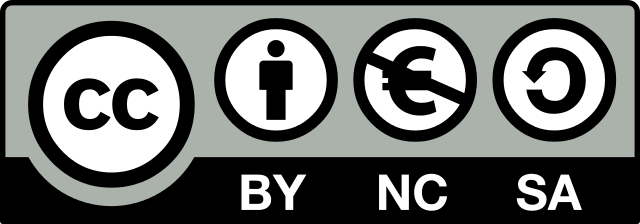
Autor: Dirk Schwenn

neue friedländer gesamtschule  
 dirk.schwenn@web.de

Überarbeitung: Tino Hempel, 2020

Name:

Das Material darf unter der folgenden Lizenz verwendet werden:



## Johnny 1: Lineare Programme

|  |  |  |
| --- | --- | --- |
| Video | Vorkenntnisse | Hilfsmittel |
| 1\_Linear.mp4 | * Aufbau und Bestandteile des von-Neumann-Rechners mithilfe des LB S. 234 ff. * Grundlagen der Arbeitsweise eines von-Neumann-Rechners mit MOPS (Klasse 10) | * Dokumentation Johnny S. 7 bis 10 * Befehlssatzes Johnny * LB Oldenburg * Datei: 01\_1\_Johnny.ram * Datei: 01\_6\_Nachfolgersumme.ram |

1. Informieren Sie sich im Video über die Bedienung des Simulators *Johnny*.   
   Bearbeiten Sie parallel die Aufgabe 2.
2. Implementieren und testen Sie das Programm aus dem Video.

000: TAKE 004  
001: ADD 005  
002: SAVE 006  
003: HLT 000  
004: 012  
005: 008

Geben Sie die Adresse der Speicherzelle der Operanden an. 004 und 005  
Begründen Sie, dass die Adresse 006 die Speicherzelle für das Ergebnis ist.

Das Ergebnis der Addition wird auf Zeile 2 mittels Save-Befehl auf Adresse 6 gespeichert.

1. Ermitteln Sie mithilfe des Befehlssatzes von Johnny die Wirkung folgenden Befehle:

|  |  |
| --- | --- |
| TAKE | Lädt den Inhalt der Adresse in den Akkumulator. |
| SAVE | Speichert den Inhalt aus dem Akkumulator auf die Adresse. |
| ADD  SUB | Addiert/Subtrahiert den Inhalt der Adresse zum/vom Akkumulator. |
| INC  DEC | Erhöht/Vermindert den Inhalt der Adresse mit Hilfe des  Akkumulators um eins. |
| HLT | Beendet das Programm. |

1. Ändern Sie den Befehl auf der Adresse 001auf SUB 005**.**Testen Sie das modifizierte Programm**.**   
   Ermitteln Sie den berechneten Term sowie das erwartete und das berechnete Ergebnis.

|  |  |  |
| --- | --- | --- |
| Term | erwartetes Ergebnis | berechnetes Ergebnis |
| 12 – 8 | 4 | 4 |

1. Modifizieren Sie das Programm so, dass die Operanden vertauscht werden.  
   Ermitteln Sie den berechneten Term sowie das erwartete und das berechnete Ergebnis. Leiten Sie eine Schlussfolgerung ab.

|  |  |  |
| --- | --- | --- |
| Term | erwartetes Ergebnis | berechnetes Ergebnis |
| 8 – 12 | -4 | 0 |

Schlussfolgerung: Das System kann nicht mit negativen Zahlen umgehen. Falls das Ergebnis negativ sein müsste, dann ermittelt das Rechenwerk Null.

1. Begründen Sie, dass das nachfolgende Programm 01\_6\_Nachfolgersumme.ram die Summe aus dem in der Zelle 020 gespeicherten Wert und seinem berechneten Nachfolger bestimmt.

000: TAKE 020  
001: SAVE 021  
002: INC 021  
003: ADD 020  
004: SAVE 021  
005: HLT 000  
020: 007

Begründung: Der Wert in Zelle 020 wird auf Zelle 021 kopiert (Zeilen 0 und 1). Dann wird der Wert der Zelle 21 um eins erhöht (Zeile 2). Damit steht der Nachfolger in Zelle 21. Anschließend erfolgt die Addition des noch im Akku stehenden Nachfolgers zum Ursprungswert in Zelle 20 und seine Speicherung in Zelle 21 (Zeilen 3 und 4).

1. Modifizieren Sie das Programm aus der vorherigen Aufgabe so ab, dass es die Summe aus dem in der Zelle 020 gespeicherten Wert und seinem übernächsten Nachfolger bestimmt. Das Ergebnis soll in der Zelle 021 abgespeichert werden.

|  |  |  |
| --- | --- | --- |
| Adresse | Befehl | Operand |
| 001 | TAKE | 20 |
| 002 | SAVE | 21 |
| 003 | INC | 21 |
| 004 | INC | 21 |
| 005 | ADD | 20 |
| 006 | SAVE | 21 |
| 007 | HLT |  |
| 020 |  | 7 |
|  |  |  |
|  |  |  |

1. Nur für LK: Entwickeln Sie ein Programm, das die Summe aus dem in der Zelle 020 gespeicherten Wert, seinem Vorgänger und seinem Nachfolger bestimmt. Das Ergebnis soll in der Zelle 021 abgespeichert werden.

|  |  |  |
| --- | --- | --- |
| Adresse | Befehl | Operand |
| 000 | DEC | 20 |
| 002 | SAVE | 21 |
| 003 | INC | 21 |
| 004 | ADD | 20 |
| 005 | SAVE | 20 |
| 006 | INC | 21 |
| 007 | ADD | 20 |
| 008 | SAVE | 21 |
| 009 | HLT |  |
| 020 |  | 7 |
|  |  |  |
|  |  |  |

## Johnny 2: Sprunganweisungen

|  |  |  |
| --- | --- | --- |
| Video | Vorkenntnisse | Hilfsmittel |
| 2\_Sprung.mp4 | * Johnny 1 | * Dokumentation Johnny S. 7 bis 10 * Befehlssatzes Johnny * LB Oldenburg * Datei: 02\_1\_max.ram |

1. Informieren Sie sich im Video über die Wirkung von Sprunganweisungen in *Johnny*.   
   Bearbeiten Sie parallel die Aufgabe 2.
2. Implementieren und testen Sie das Programm zur Bestimmung des Maximums der Daten zweier Speicherzellen aus dem Video.

000: TAKE 017  
001: SUB 018  
002: SAVE 021  
003: TST 021  
004: JMP 008  
005: TAKE 018  
006: SAVE 020  
007: HLT 000  
008: TAKE 017  
009: SAVE 020  
010: HLT 000  
017: 055  
018: 093

Begründen Sie, dass die Adresse 020 die Speicherzelle für das Ergebnis ist.

Das Programm holt nach der Entscheidung (Zeile 3) das Maximum entweder aus Zelle 17 oder 18 (Zeilen 5 und 8) und speichert diese Werte in Zelle 20 (Zeilen 6 und 9).

1. Ermitteln Sie mithilfe des Befehlssatzes von Johnny die Wirkung folgenden Befehle:

|  |  |
| --- | --- |
| TST | Vergleicht den Operandenteil der Adresse mit Null. Bei Gleichheit wird der nachfolgende Befehl übersprungen. |
| JMP | Springt zur angegebenen Adresse. |

1. Testen Sie das Programm mit der Bedingung, dass in 017 ein Wert steht, der

|  |  |  |  |
| --- | --- | --- | --- |
|  | Zelleninhalt | | |
| vorher | | nachher |
| 017 | 018 | 020 |
| a) | 99 | 93 | 99 |
| b) | 55 | 93 | 93 |
| c) | 93 | 93 | 93 |

* 1. größer als der in 018 ist.
  2. kleiner als der in 018 ist.
  3. identisch mit dem in 018 ist.

Nutzen Sie die nebenstehende Tabelle. Leiten Sie eine Schlussfolgerung ab.

Schlussfolgerung: Das Programm arbeitet korrekt.

1. Modifizieren Sie das Programm so, dass das Minimum der Inhalte der Zellen 017 und 018 in der Zelle 020 gespeichert wird.

|  |  |  |
| --- | --- | --- |
| Adresse | Befehl | Operand |
| 000 | TAKE | 017 |
| 001 | SUB | 018 |
| 002 | SAVE | 021 |
| 003 | TST | 021 |
| 004 | JMP | 008 |
| 005 | TAKE | **017** |
| 006 | SAVE | 020 |
| 007 | HLT | 000 |
| 008 | TAKE | **018** |
| 009 | SAVE | 020 |
| 010 | HLT | 000 |
| 011 |  |  |
| 012 |  |  |
| 013 |  |  |
| 014 |  |  |
| 015 |  |  |
| 016 |  |  |
| 017 |  | 055 |
| 018 |  | 093 |
| 019 |  |  |
| 020 |  |  |
| 021 |  |  |

1. Nur für LK: Modifizieren Sie das Programm so, dass das Maximum der Inhalte in den Zellen 017, 018 und 019 in der Zelle 020 gespeichert wird.

|  |  |  |
| --- | --- | --- |
| Adresse | Befehl | Operand |
| 000 | TAKE | 017 |
| 001 | SUB | 018 |
| 002 | SAVE | 021 |
| 003 | TST | 021 |
| 004 | JMP | 007 |
| 005 | TAKE | 018 |
| 006 | SAVE | 017 |
| 007 | TAKE | 017 |
| 008 | SUB | 019 |
| 009 | SAVE | 021 |
| 010 | TST | 021 |
| 011 | JMP | 014 |
| 012 | TAKE | 019 |
| 013 | SAVE | 017 |
| 014 | TAKE | 017 |
| 015 | SAVE | 020 |
| 016 | HLT | 000 |
| 017 |  | 055 |
| 018 |  | 093 |
| 019 |  | 099 |
| 020 |  |  |
| 021 |  |  |

## Johnny 3: Wiederholungen in der Programmabarbeitung

|  |  |  |
| --- | --- | --- |
| Video | Vorkenntnisse | Hilfsmittel |
| 3\_Schleifen.mp4 | * Johnny 2 | * Dokumentation Johnny S. 7 bis 10 * Befehlssatzes Johnny * LB Oldenburg * Datei: 03\_4\_division.ram |

1. Informieren Sie sich im Video über die Programmierung von Wiederholungen in *Johnny*.   
   Bearbeiten Sie parallel die Aufgabe 2.
2. Implementieren Sie das Programm.  
   Protokollieren Sie die Arbeitsweise des Programms zur Bestimmung des Produkts der Werte in den Speicherzellen 010 und 011 in Zelle 012. Schalten Sie dazu die Sicht auf das Steuerwerk frei.

000: NULL 012   
001: TAKE 012  
002: ADD 010   
003: SAVE 012   
004: DEC 011   
005: TST 011   
006: JMP 001   
007: HLT 000

Programmcounter (in der Form 000 🡪 001 🡪 002 🡪 …):

000 🡪 001 🡪 002 🡪 003 🡪 004 🡪 005 🡪 006 🡪 001 🡪 002 🡪 003 🡪 004 🡪 005 🡪 006 🡪 001 🡪 002 🡪 003 🡪 004 🡪 005 🡪 007

Speicher- und Registerbelegung:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 010: | 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| 011: | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |
| 012: | 0 | 5 | 10 | 15 |  |  |  |  |  |  |  |  |  |
| Akku: | 5 | 2 | 5 | 10 | 1 | 10 | 15 | 0 |  |  |  |  |  |

1. Testen Sie das Programm mit verschiedenen Werten einschließlich Null auf den Adressen 010 bzw. 011.  
   Leiten Sie Schlussfolgerungen zur Arbeitsweise ab.

* 0 \* x wird korrekt, aber ineffizient berechnet.
* x \* 0 wird falsch berechnet und ergibt x.
* Der Algorithmus arbeitet fehlerhaft und teilweise ineffizient.

1. Analysieren Sie das Programm 03\_4\_division.ram. Die Speicherzelle 018 beinhaltet den Dividenden, die Zelle 019 den Divisor. Das Ergebnis wird auf Zelle 020 abgelegt.   
   Protokollieren Sie den Programmablauf.   
   Beschreiben Sie die prinzipielle Arbeitsweise des Algorithmus.

000: NULL 020   
001: TAKE 018   
002: SUB 019  
003: SAVE 018  
004: INC 020  
005: TST 018  
006: JMP 001  
007: HLT 000  
018: 006  
019: 002  
020: 000

Programmcounter (in der Form 000 🡪 001 🡪 002 🡪 …):

000 🡪 001 🡪 002 🡪 003 🡪 004 🡪 006 🡪 001 🡪 002 🡪 003 🡪 004 🡪 006 🡪 001 🡪 002 🡪 003 🡪 004 🡪 005 🡪 007

Speicher- und Registerbelegung:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 018: | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 0 | 0 | 0 |
| 019: | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 020: | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 |
| Akku: | 0 | 6 | 4 | 4 | 1 | 4 | 2 | 2 | 2 | 0 | 0 | 3 | 0 |

Der Algorithmus arbeitet so, dass vom Dividenden der Divisor solange abgezogen wird, bis dieser Null ist. Die Anzahl des Abziehens wird mithilfe einer inkrementierenden Zelle gezählt.

1. Testen Sie das obige Programm für die Berechnung von 21:7, 22:7 und 20:7. Beschreiben Sie Ihre Beobachtungen. Leiten Sie Schlussfolgerungen ab.

21:7 wird korrekt ermittelt, 22:7 und 20:7 wird falsch berechnet. Da das System nicht mit Dezimalzahlen rechnen kann, sollte zumindest der ganze Anteil korrekt ermittelt werden. Dies ist aber nicht der Fall, der Wert ist jeweils um 1 zu groß.

1. Nur für LK: In der Datei 03\_6\_test.ram befinden sich in den Zellen 000 bis 017 der Algorithmus. Ab der Zelle 021 können fortlaufend Daten größer null eingetragen werden. Eine Null schließt die Datenfolge ab. Das Ergebnis der legt der Algorithmus in der Zelle 020 ab.

000: TAKE 021  
001: SAVE 020   
002: TAKE 021   
003: SUB 020   
004: SAVE 019   
005: TST 019   
006: JMP 008   
007: JMP 010   
008: TAKE 021  
009: SAVE 020  
010: INC 002  
011: INC 008  
012: INC 013  
013: TST 021  
014: JMP 002  
015: HLT 000  
016: TAKE 020  
017: HLT 000  
018:  
019:  
020:  
021: 009  
022: 007  
023: 025

* 1. Protokollieren Sie den Programmlauf für die gegebenen Werte. Beachten Sie dabei, dass sich der Quelltext des Programms während der Laufzeit ändert! Nutzen Sie eine Speicherbelegungsübersicht der Zellen 002, 008, 013 und 020.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 002: | 21 | 21 | 22 | 23 | 24 |  |  |  |  |  |  |  |  |
| 008: | 21 | 21 | 22 | 23 | 24 |  |  |  |  |  |  |  |  |
| 013: | 21 | 21 | 22 | 23 | 24 |  |  |  |  |  |  |  |  |
| 020: | 0 | 9 | 9 | 9 | 25 |  |  |  |  |  |  |  |  |

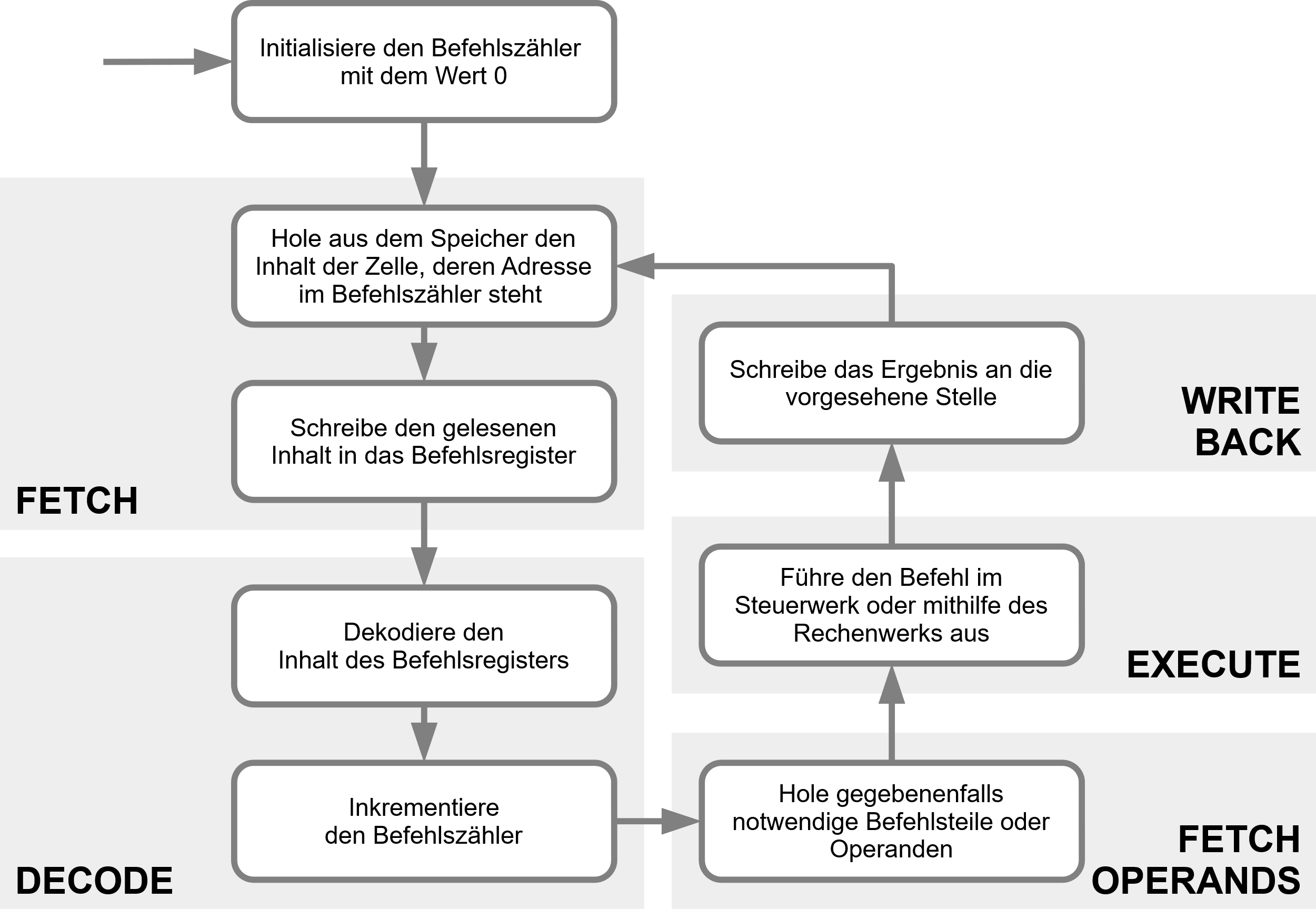
* 1. Testen Sie das Programm für verschiedene Datenreihen. Beachten Sie dabei, dass das Programm für jede Datenreihe neu zu laden und die Daten ab Zelle 021 anzupassen sind. Beschreiben Sie die Wirkung des Programms.

Der Algorithmus ermittelt den größten Wert (ungleich Null) in einer beliebig langen Liste von Daten.

## Johnny 4: Von-Neumann-Zyklus und Erweiterungsvorschläge

|  |  |  |
| --- | --- | --- |
| Video | Vorkenntnisse | Hilfsmittel |
| 4\_Zyklus.mp4 | * Johnny 2 (GK) * Johnny 3 (LK) | * Dokumentation Johnny S. 7 bis 10 * Befehlssatzes Johnny * LB Oldenburg |

1. Informieren Sie sich im Video über den von-Neumann-Zyklus. Stellen Sie den Zyklus grafisch darf. Erfassen Sie dabei die genauen Aufgaben in den Phasen. Nutzen Sie Ihre Kenntnisse aus den vorherigen Lektionen zum Steuerwerk.



1. Nur für LK: Dokumentieren Sie den Programmablauf für ein Programm aus Johnny 2.
2. Nur für LK: Bearbeiten Sie zwei von drei Aufgaben.
   1. Erweitern Sie das Programm 03\_2\_multiplikation.ram so, dass es stehts korrekt rechnet und möglichst effizient rechnet.

Das Programm basiert auf dem Multiplikationsprogramm 03\_2. Es wurde zu Beginn um das Testen der beiden Faktoren auf null ergänzt.

* 1. Erweitern Sie das Programm 03\_4\_division.ram so, dass es stehts korrekt rechnet und ggf. zusätzlich den Rest der Division bestimmt.
  2. Implementieren Sie ein Programm zur Berechnung der Potenz a2.

Das Programm basiert auf dem Multiplikationsprogramm 03\_2. Es wurde zu Beginn um das Kopieren der Basis als zweiter Faktor auf die Zelle 21 ergänzt.