

CAuBS Einsendeaufgaben 1 – Deadline 04.11.2018 23:55 Uhr



Aufgabe 1: 2.1.3.1.2 Rechenwerk: Größer als

In den Beispielen für mögliche Statusbits (Flags) sieht man u.a.

"Waren beide Operanden gleich?" sowie

"War A kleiner als B?".

Die dritte Variante "War A größer als B?" muss jedoch nicht vorhanden sein. Warum kann das Steuerwerk auch ohne dieses Statusbit auskommen?

Die Stati „gleich“ oder „kleiner als“ ergeben, wenn beide nicht zutreffen, automatisch die Lösung „A war größer“.



Aufgabe 2: 2.1.3.2 Speicherwerk: Speicherzelle 3

Zu Beginn des Videos empfängt das Speicherwerk über den Adressbus die Adresse "00000011".

Warum wird damit Speicherzelle 3 angesprochen?

Da die Adressen binär angegeben werden und binär „11“ dezimal „3“ entspricht, wird somit die dritte Speicherzelle angesprochen.



Aufgabe 3: 2.1.3.2 Speicherwerk: Die 4 GiB-Grenze

In der Vergangenheit war immer wieder folgender Satz zu hören: "Ein 32-Bit-Betriebssystem kann maximal 4 GiB Arbeitsspeicher (RAM) verwalten". Gehen wir für unser Speicherwerk davon aus, dass der Adressbus eine Breite von 32 Bit besitzt. Damit können dann auch maximal 4 GiB im Speicherwerk angesprochen werden.

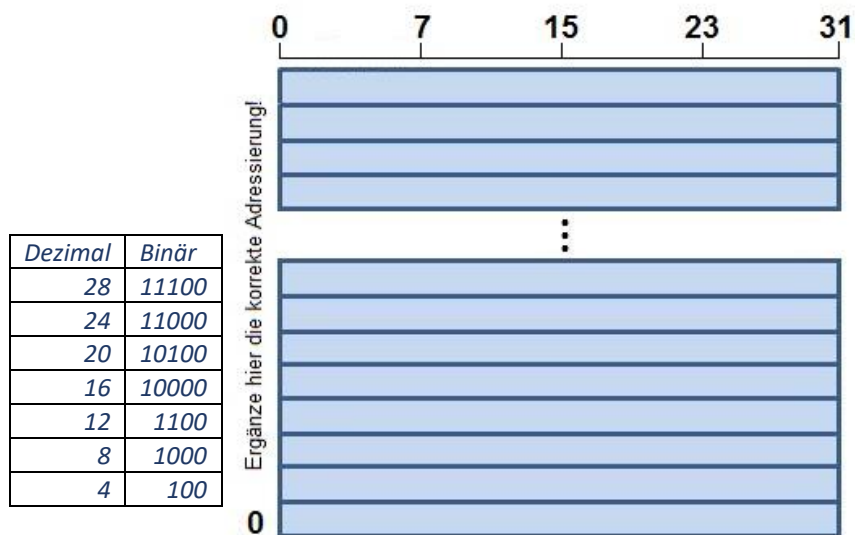
- Wie viele Adressen können mit 32 Bit unterschieden werden? (Das Ergebnis gibt dann gleichzeitig die Anzahl der Speicherzellen im Speicherwerk an.)

Es können 2^{32} Speicherzellen unterschieden werden -> 4.294.967.296

- Wie viele Bit besitzt eine einzige Speicherzelle unter Berücksichtigung der 4 GiB Gesamtspeichermenge und dem Hinweis zur Bedeutung von "GiB"?

Eine Speicherzelle enthält 1 Byte/8Bit

- Gehen wir weiter davon aus, dass der Datenbus ebenfalls 32 Bit breit ist. Beim Eintreffen einer Adresse über den Adressbus am Speicherwerk und dem Befehl "Lesen" auf dem Steuerbus wird das Speicherwerk über den Datenbus genau 32 Bit zurücksenden. Der Speicher kann dann in einer Skizze mit Zellen gezeichnet werden, die jeweils eine Breite von 32 Bit besitzen. Wie lautet die korrekte Adressierung dieser 32 Bit breiten Zellen? (Gib die Adressierung sowohl in dezimaler, als auch in binärer Schreibweise an!)



- d. Wenn eine Speicherzelle in der Abbildung eine Breite von 32 Bit besitzt, warum wird oben im Bild nur bis 31 gezählt?

Die Null wird mitgezählt; es sind somit 32 Bit



Aufgabe 4: 2.1.5.1 Aufbau und Arbeitsweise eines Registers: Flip-Flop I+II+III

- a. Im RS-Flip-Flop ist der Wert **Null** gespeichert. Die Eingangsleitungen besitzen folgende Werte:
- Input = 1
 - Select = 1
 - Write = 0
- Welcher Wert wird unter diesen Voraussetzungen für den Ausgang (Output) der Speicherzelle ermittelt?

Am Ausgang muss eine 0 ermittelt werden, der Input ist aufgrund Write = 0 irrelevant

- b. Im RS-Flip-Flop ist der Wert **Eins** gespeichert. Die Eingangsleitungen besitzen folgende Werte:
- Input = 1
 - Select = 1
 - Write = 0
- Welcher Wert wird unter diesen Voraussetzungen für den Ausgang (Output) der Speicherzelle ermittelt?

Am Ausgang muss eine 1 ermittelt werden, der Input ist aufgrund Write = 0 irrelevant

- c. Welchen Wert gibt das RS-Flip-Flop am Ausgang Q aus, wenn beide Eingänge gleich Null sind (S=0, R=0)?

Es wird der Wert ausgegeben, der im Flipflop gespeichert ist.



Aufgabe 5: 2.2.4.4 Registerindirekte Adressierung: Adressierungsarten

Welche Bedeutungen haben die folgenden Befehle:

- **ADD R1, ACC, 8**

Addiere den Wert aus dem „Akkumulator“ zu dem Wert in „Speicherzelle 8“ und schreibe das Ergebnis nach „Register 1“

- **SUB ACC, 10, (R2)**

Ziehe vom Wert von „Speicherzelle 10“ den Inhalt „der Speicherzelle ab, die in Register2 adressiert ist“. Schreibe das Ergebnis in den „Akkumulator“.

- **JUMP (R3)**

Springe im Programm an die im Register3 angegebene Stelle.

- **ADD ACC, (ACC), #5**

Füge dem Wert aus der im Akkumulator angegebenen Speicherzelle 5 hinzu und speichere das Ergebnis im Akkumulator.

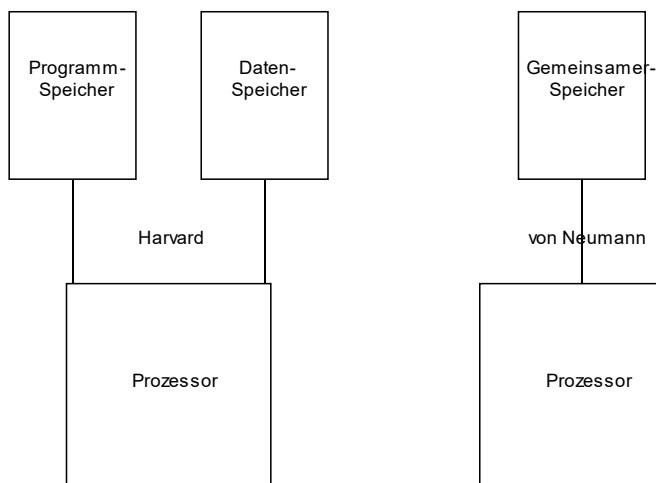
(R1, R2 und R3 bezeichnen Register auf der CPU, siehe Gesamtbild eines Von-Neumann-Rechners.)



Aufgabe 6: 2.2.7 Aufgaben & Co. zu Prozessoren: VNA vs. Harvard

Erkundige dich nach den Unterschieden zwischen der **Von-Neumann-Architektur** und der **Harvard-Architektur**!

- a. Stelle die Harvard-Architektur und die Von-Neumann-Architektur in einer Skizze gegenüber und erläutere die Unterschiede! Nutze dafür nicht nur Wikipedia, auch Brinkschulte et.al. 2010 und Böttcher 2006 geben Erläuterungen dazu.



Während der von-Neumann Rechner nur einen Bus für Befehle und Daten verwendet, nutzt die Harvard-Architektur je einen Bus für Daten und Programm.

Die Harvard Architektur trennt Daten und Programm wodurch die Programm/Datenspeicherübertragung schneller erfolgen kann als bei der von Neumann Architektur, bei der es nur einen gemeinsamen Speicher gibt.

Desweiteren ist Programmcode bei der Harvard-Architektur gegen Veränderung bei fehlerhaftem Datenzugriff geschützt.

Gegenüber der von Neumann Architektur hat die Harvard Architektur den Nachteil, dass sowohl ein Programmspeicher als auch ein Datenspeicher in ausreichender Größe vorhanden sein müssen, da freier Speicher nicht gemeinsam genutzt werden kann.

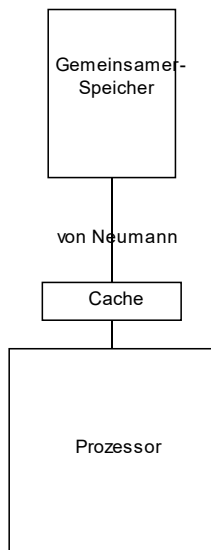
- b. Was meint beispielsweise Böttcher 2006 wenn er in Kapitel 7.2.1 von einer "Pseudo-Harvard-Architektur" schreibt?

Er bezieht sich darauf, dass der L1 Cache in einen Programm- und einen Datenteil unterteilt sind während der eigentliche Speicher nach wie vor auf Basis der klassischen von Neumann Architektur arbeitet.

- c. Und welchen Vorteil bieten die in Kapitel 7.2 von Böttcher 2006 beschriebenen Cache-Speicher?

Der Cache Speicher ist zwar sehr klein und teuer aber um bis zu Faktor 100 schneller als der Hauptspeicher

- d. Integriere die Cache-Speicher in deine Skizze der Von-Neumann-Architektur!



Aufgabe 7: 2.3.4 Interrupt-Controller: Schätze die Folgen ab!

Beschreibe ein Szenario, bei dem jeweils eine der vier Bedingungen einer präzisen Unterbrechung nicht erfüllt ist.

- a. Was sind die Folgen, wenn Bedingung 1 nicht erfüllt ist?

Die CPU findet nach der Unterbrechung nicht zurück ins Programm – in der Regel sollte das einfach zum Stillstand führen, via zufälliger Basisregister/Stackregister könnte aber auch ein zufälliger Befehl getroffen werden, ab dem der dort liegende Programmcode weiter läuft und Daten zerstören. Die Folge ist somit, dass das Verhalten des Computers nicht vorhersehbar ist.

- b. Was sind die Folgen, wenn Bedingung 2 nicht erfüllt ist?

Sind die Befehle nicht vollständig abgearbeitet, ist es sowohl möglich, dass nichts schief geht (z.B. eine Bedingung (die zufällig passt) hätte ohnehin zum anliegenden Programmzähler geführt) als auch dass mit falschen Daten gearbeitet wird oder Daten zur Abarbeitung fehlen. Die Folge ist somit auch hier, dass das Verhalten des Computers nicht vorhersehbar ist.

- c. Was sind die Folgen, wenn Bedingung 3 nicht erfüllt ist?

Es gibt keine Folgen wenn z.B. der Nachfolgende Befehl Daten (nochmal) einliest – wenn aber gerechnet / geschrieben wird, wird u.U. mit falschen Daten gearbeitet. (z.B. $a=a+3$ ergibt beim zweiten Mal einen anderen Wert). Die Folge ist somit auch hier, dass das Verhalten des Computers nicht vorhersehbar ist.

- d. Was sind die Folgen, wenn Bedingung 4 nicht erfüllt ist?

Wenn nicht klar ist, ob der Befehl schon ausgeführt worden ist oder nicht, ist es nicht möglich zu entscheiden, ob der Befehl nochmals oder erstmals oder gar nicht ausgeführt wird.



Aufgabe 8: 2.3.4.3 Quasi-gleichzeitige Ausführung mehrerer Prozesse: Betriebssystem und Prozesse im Wechsel

Nach dem Start eines Rechners mit integriertem Hardware-Taktgeber werden insgesamt drei Prozesse gestartet:

1. Prozess BS: Das Betriebssystem.
2. Prozess X: Eine beliebige Anwendung.
3. Prozess Y: Eine weitere beliebige Anwendung.

Prozess BS beginnt ab Speicherstelle 0 im Hauptspeicher, damit wird das Betriebssystem zuerst gestartet.

Erläutere was passiert, wenn Prozess BS sich nun um die quasi-gleichzeitige Ausführung aller Prozesse auf der CPU kümmert. In welcher Weise wechseln sich BS, X und Y auf der CPU ab? Wodurch wird dieser Wechsel veranlasst?

Ein Taktgenerator erzeugt ein Signal, das für Zeitslots genutzt wird. Zuerst wird das Betriebssystem aus Speicher 0 gestartet und ruft im Verlauf die Programme X und Y auf. Durch den Taktgenerator wird regelmäßig eine Interruptbehandlungsroutine ausgelöst. Die als Teil des Betriebssystems den Programmen X und Y unterschiedlich große Zeitslots zur Verfügung stellen kann



Aufgabe 9: 2.3.4.4.1 Allgemeiner Aufbau eines Controllers: Lesen und/oder schreiben?

Auf das Datenregister eines Controllers kann die CPU sowohl lesend, als auch schreibend zugreifen. Wie verhält es sich aber beim Steuer- und beim Zustandsregister? Ist der Zugriff hier *nur lesend*, *nur schreibend* oder *lesend und schreibend* möglich?

Auf das Statusregister greift die CPU nur lesend zu (Bereit / Beschäftigt). Das Steuerregister wird von der CPU nur mit Arbeitsanweisungen beschrieben, erhält aber keine.



Aufgabe 10: 2.3.5 DMA-Controller: DMA und CPU

Der Einsatz eines DMA-Controllers hat nicht nur Vorteile. Ein Nachteil ist beispielsweise, dass Datenbus und Adressbus während eines Datentransfers per DMA nicht der CPU zur Verfügung stehen.

- a. Welche Auswirkungen hat dies auf die Arbeitsweise der CPU?

Die CPU erteilt dem DMA Controller nur noch die Anweisung, was zu transferieren ist und erhält irgendwann ein „fertig“ per Interrupt statt selbst die Daten zu laden.

- b. Was kann die CPU während der Zeit des DMA-Transfers erledigen, und was geht nicht mehr?

Die CPU kann in der Zeit nur mit Daten aus Cache und Stack arbeiten. Der DMA Controller in der Datenübertragung performanter als die CPU selbst sodass die Übertragung schneller abgeschlossen werden kann. Dem DMA Controller können einzelne Zyklen zugestanden werden, was bei einer Übertragung von einer langsamen Festplatte zum schnellen Speicher Sinn macht, wodurch die CPU auch „während“ der Übertragung immer Daten zwischenholen kann. Bei Burst-Übertragungen großer Blöcke gerät die CPU in den Stillstand.