

Herzliche Einladung zu VECTOR INFORMATIK GMBH

23. Mai 2018 in Stuttgart

Wir arbeiten an den Technologien von morgen, und dafür suchen wir
Ideengeber, die Kreativität und Leidenschaft mit Realitätssinn verbinden.



21 Standorte weltweit
> 1.800 Mitarbeiter
414 Mio. € Umsatz (2016)

Lerne uns kennen!

- ▶ Entwicklung elektronischer Innovationen für die Automobilbranche
- ▶ Stark wachsendes Unternehmen
- ▶ Gute Chancen für Hochschul-Absolventen



AKKA

PASSION FOR
TECHNOLOGIES

**EXKURSION
AKKA, MBTECH**

**Standort Flugfeld (Sindelfingen)
23.05.2018**

EINLADUNG EXKURSION ZU AKKA, MBTECH AM 23.05.18 UM 14.00 UHR

ÜBER UNS

Mit über 15.500 Mitarbeitern ist die AKKA Gruppe der europäische Engineeringdienstleister und Technologieberater.

Mit umfassenden Kompetenzen zur gesamten technologischen Produktentwicklung und Systemintegration setzt die Gruppe vor allem in der Automobilentwicklung, im Consulting und in der Digitalisierung Maßstäbe. Ebenfalls ist AKKA mit Aktivitäten in der Luftfahrt, dem Schienenverkehr und vielen weiteren Branchen vertreten.

UNSER PROGRAMM FÜR SIE

Unser Fachbereich Electric/Electronics lädt Sie an den AKKA,MBtech Standort „Flugfeld“ in Sindelfingen ein. Folgendes haben wir für Sie vorbereitet:

- Einblick in unser Unternehmen:
Gesamtfahrzeugentwicklung
- Einblick in den Bereich E/E:
Elektrik/Elektronik-Entwicklung und -Testing für Fahrzeuge: Software – Hardware – Gesamtsystem
- Industriepromotion:
Einsatz künstlicher Intelligenz für die Absicherung von autonomen Fahrzeugen
- HR:
Unsere Einstiegsmöglichkeiten und Benefits
- Rundgang durch unsere Prototypen-Werkstatt
- Get together mit Diskussion (für Snacks und Getränke ist gesorgt ☺)

Übung 01: Informationstechnik (IT)

Marc Weber, Daniel Grimm

Institutsleitung

Prof. Dr.-Ing. Dr. h.c. J. Becker

Prof. Dr.-Ing. E. Sax

Prof. Dr. rer. nat. W. Stork

Institut für Technik der Informationsverarbeitung (ITIV)



Teil 1: Organisatorisches, Einführung & Besprechung Aufgabenblatt 1

Inhalt: Übung 01 – Teil 1

1

- Organisatorisches

2

- Einführung

3

- Besprechung Aufgabenblatt 1

IT-Veranstaltungsübersicht SS2018

- Vorlesung
 - 2 SWS
- Übung
 - 1 SWS
- Ort und Zeit
 - Mittwoch: 09:45 bis 11:15 Uhr, Gebäude 10.21, Benz Hörsaal
 - Donnerstag: 14:00 bis 15:30 Uhr, Gebäude 30.46, Neue Chemie
- Praktikum
 - 2 SWS
 - Genauere Informationen während der Einführungsveranstaltungen
 - Dienstag, 08.05.2018, 17:30 – 19:00 Uhr, Gebäude 50.35 Am Fasanengarten
 - Dienstag, 15.05.2018, 17:30 – 19:00 Uhr, Gebäude 50.35 Am Fasanengarten
- Tutorium
 - Selbststudium
 - Unterlagen inkl. Lösung auf ILIAS

Unterlagen zu Informationstechnik

■ Nutzung der Lernplattform ILIAS

- <https://ilias.studium.kit.edu>
- Registrierung notwendig (falls noch nicht erfolgt)
- Kurs: [2311622] Informationstechnik (SS 2018)
 - Passwort: **it@SS2018** (Groß-/Kleinschreibung beachten!)

■ Inhalt

- Vorlesungsfolien, Übungsblätter, Übungsfolien, Tutoriumsaufgaben, ...

[2311622] Informationstechnik (SS 2018)



[2311622] Informationstechnik (SS 2018)

Anmeldung zur Klausur

- Bachelor: Verwendung der Selbstbedienungsfunktion des Studienbüros
<https://campus.studium.kit.edu/>
- Andere oder nicht online freigeschaltete Studiengänge: Direkte Anmeldung durch Abgabe des blauen Zettels oder Entsprechendem bei einem Betreuer (spätestens eine Woche vor der Prüfung)
- Das Bestehen des Praktikums ist keine Voraussetzung für die Teilnahme an der Klausur
 - Praktikum und Klausur sind getrennte Studienleistungen
- **Anmeldeschluss: 28.09.2018**
- **Abmeldeschluss: 04.10.2018**

Klausur (schriftlich)

- Inhalt: Vorlesung, Übung, Tutorium und Praktikum
 - Für Studenten mit alter SPO wird es keine Klausur ohne Praktikumsanteile geben
- Termin: Donnerstag, **05.10.2018, 08:00 bis 10:00 Uhr**
- Mitzubringen: Fricard oder Lichtbildausweis und Schreibzeug
 - Nicht Bachelorstudenten zusätzlich: Studienzeitsbescheinigung
- Hilfsmittel: 1x A4 Blatt (2x A4 Seiten, handschriftlich, lesbar ohne Hilfsmittel)

- Hörsaalverteilung (nach Nachnamen, bei Doppelnamen ist der erste Nachname maßgebend)
 - Details werden vor der Prüfung über ILIAS bekannt gegeben

- Organisatorisches



Informationstechnik Übung

Exemplarische Einführung der höheren Programmiersprache C++

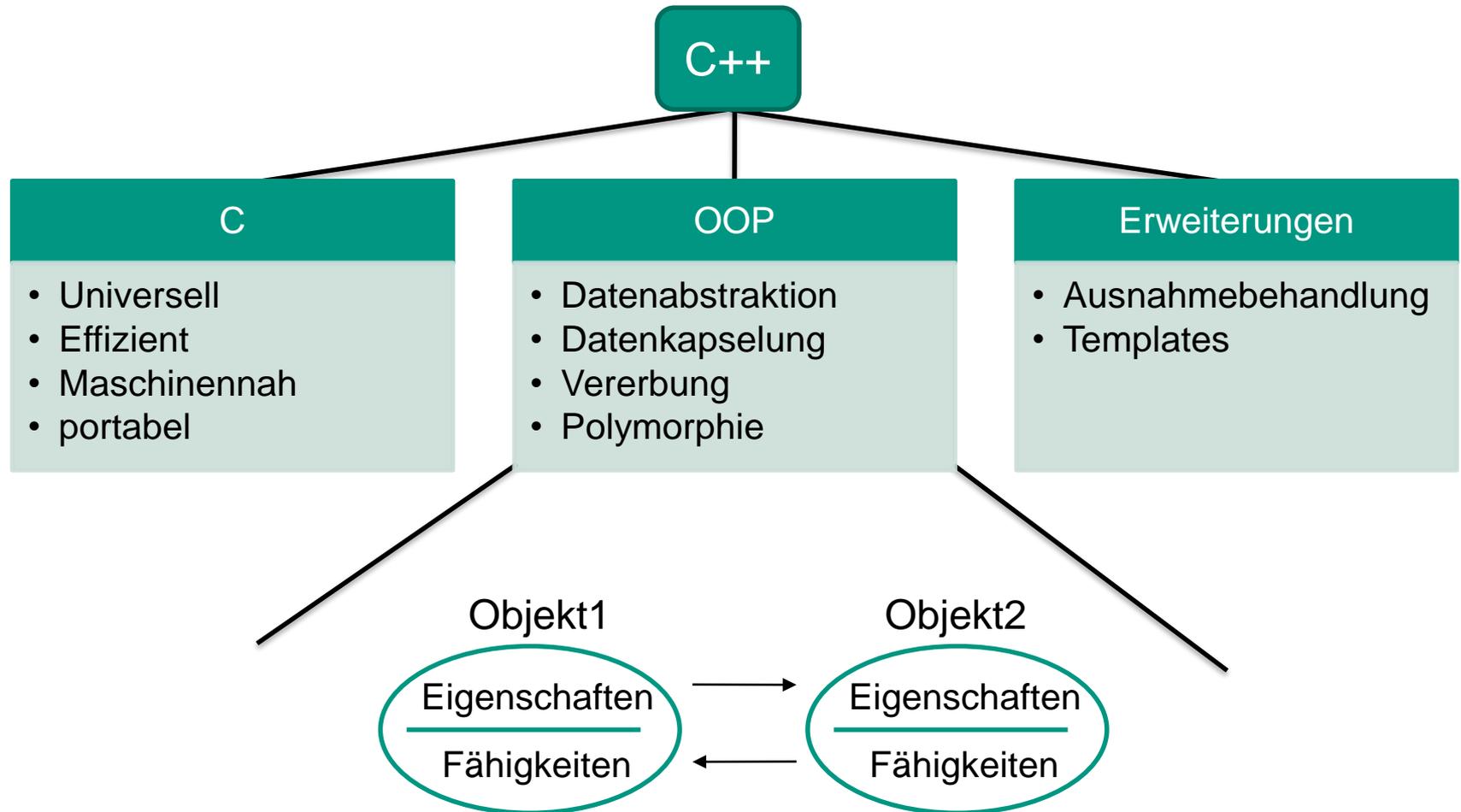
- C und C++ sind die derzeit am weitest verbreiteten Programmiersprachen bezogen auf die Zahl der Anwendungen
- C ist hardwarenah (Treiberprogrammierung)
- C++ umfangreiche Erweiterung von C
- C++ exemplarisch für objektorientierte Programmierung (kommt erst später in der Vorlesung)
- C++ ist eine der mächtigsten Programmiersprachen
- C++ auch Basis für die Hardwarebeschreibungssprache SystemC

Lernziele der Übung

- Der Lernende soll am Ende:
 - Die Grundzüge der Programmiersprache C++ anwenden können
 - Konkrete Problemstellungen mit Hilfe der Informationstechnik lösen können
 - Seine Programme nach den Prinzipien der Objektorientierung strukturiert aufbauen können
 - Algorithmen in unterschiedlichen Darstellungsformen beschreiben können und in lauffähige Programme umsetzen können
 - Qualitätsmerkmale von Algorithmen und Programmen kennen und anwenden und hierzu Tests und Testprogramme erstellen
 - Mit den Grundfunktionen einer Entwicklungsumgebung umgehen können

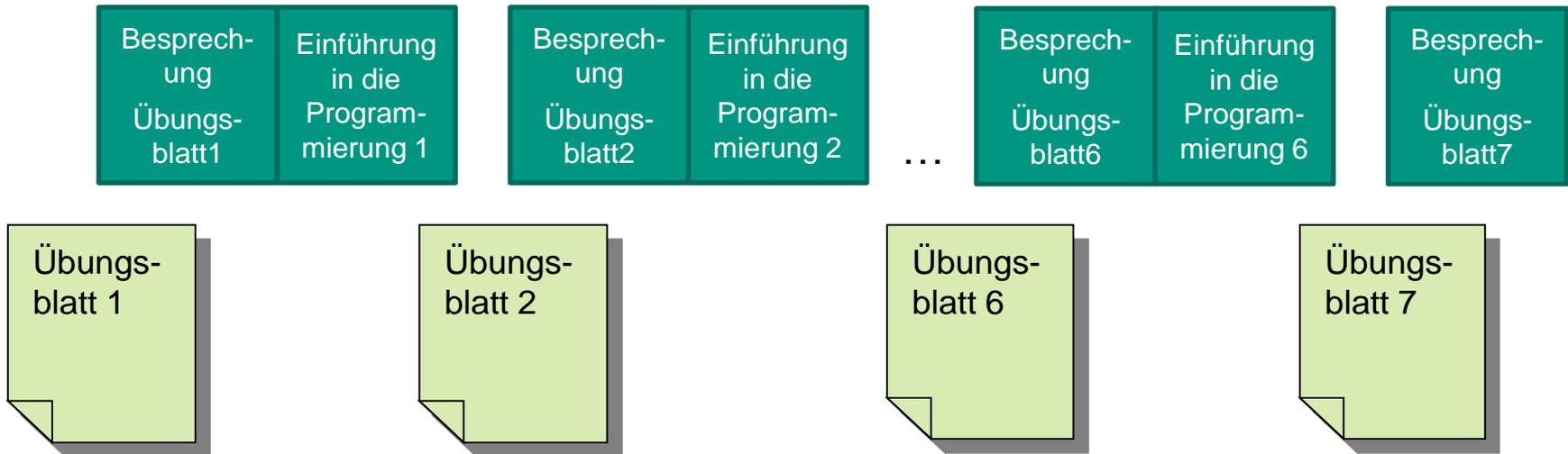


Warum C++?



Teilung der Übung

- Übung besteht im Allgemeinen aus 2 Teilen
 - 1. Teil: Besprechung der Übungsaufgaben
 - 2. Teil: Einführung in die Programmierung



Aktives Lernen

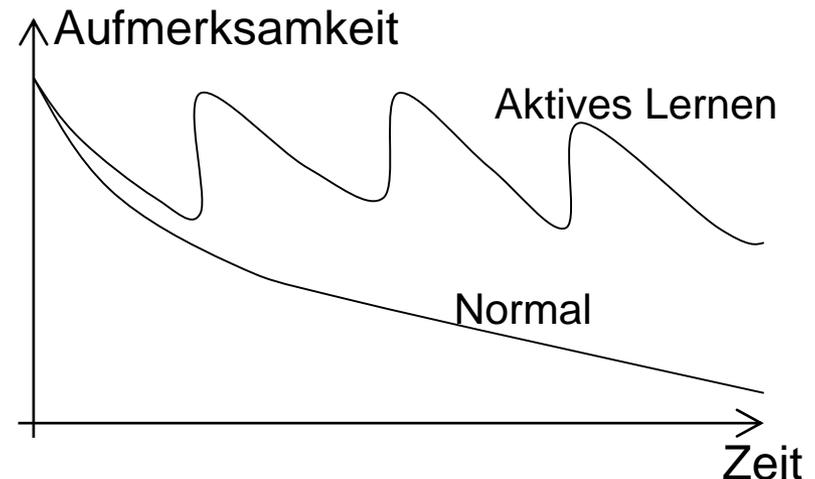
■ Aktive / Direkte Mitarbeit der Studenten

- Kurze Aufgaben während der Übung zum direkten Lösen
- Max. 2 Minuten
- In Gruppenarbeit
- Direkte Lösungsanfrage an Gruppen
- Themenwechsel



■ Ziele

- Schnelleres Lernen
- Weniger Nacharbeit
- Direkteres Verstehen
- Erhöhte Aufmerksamkeit



Inhaltsverzeichnis Übung

1

- Variablen & Gültigkeitsbereich, Operatoren, Arrays, Kontrollstrukturen

2

- Funktionen, Zeiger, Strings

3

- Header, Programmstrukturen

4

- Objektorientierung, Vererbung & Klassendiagramm

5

- Dynamische Speicherverwaltung & Polymorphie, STL

6

- Sortieralgorithmen, Tiefensuche

7

- Optimierungsalgorithmen, Q&A

- Einführung



Aufg. 1.01: Verständnisfragen Lsg. (1)

- a) Der Cache kompensiert die Latenz zwischen Prozessor und Hauptspeicher.
- b) Definieren Sie den Begriff "Write through" in Bezug auf Caches und nennen Sie einen Vor- und einen Nachteil.

Jeder Schreibzugriff wird auch im Arbeitsspeicher durchgeführt. Jeder Schreibzugriff bedingt einen Zugriff auf den Systembus.

Nachteile:

1. Durch Belastung des Systembusses, wird der Zugriff anderer Komponenten auf den Arbeitsspeicher erheblich behindert
2. Längere Ausführungszeit

Vorteile:

1. Datenkonsistenz ist gesichert, d.h. es ist gewährleistet, dass Datum im Cache und Arbeitsspeicher stets den gleichen Wert hat
2. Erheblich einfachere Steuerung

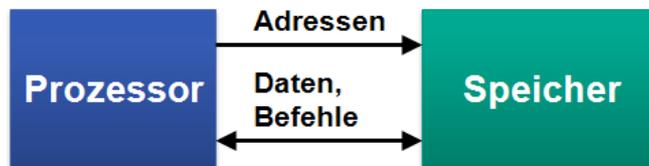
Aufg. 1.01: Verständnisfragen Lsg. (2)

- c) Techniken zur Beschleunigung der Befehlsausführung sind Pipelining, Superpipelining, Superskalar, Multi-Core.
- d) Bei der Interpretation eines Befehls im Speicher wird unterschieden, ob das niedrigstwertige Byte an der höchsten Adresse („big endian“, z.B. Apple), oder niedrigsten Adresse („little endian“, z.B. Intel x86) gespeichert wird.

Aufg. 1.01: Verständnisfragen Lsg. (3)

e) Was ist der Hauptunterschied zwischen der von-Neumann- und der Harvard-Architektur? Hinweis: Sie können Ihre Lösung auch graphisch darstellen.

■ Von Neumann-Architektur:



Ein Speicher sowohl für Programm-Daten als auch Daten-Daten

■ Harvard-Architektur:



Getrennte Speicher für Programm-Daten und Daten-Daten / gleichzeitiger Zugriff möglich

Aufg. 1.01: Verständnisfragen Lsg. (4)

- f) Eine Rechnerarchitektur kann grundlegend in interne und externe Architektur unterteilt werden.
- g) Eine von-Neumann-Rechnerarchitektur besteht aus den Hauptkomponenten Rechenwerk, Steuerwerk, Speicherwerk und Ein-/Ausgabewerk.
- h) Erläutern Sie folgende Adressierungsmodi einer CPU:
- Register Mode: Der Operand wird aus einem Register bezogen.
 - Direct Mode: Die Speicheradresse des Operanden ist in der Instruktion gespeichert.
 - Immediate Mode: Der Operand ist Teil der Instruktion.
 - Register Indirect Mode: Ein Register enthält die Speicheradresse, an der der Operand gespeichert ist (Pointer).

Aufg. 1.01: Verständnisfragen Lsg. (5)

- i) Das Taktsignal einer CPU gibt maßgeblich die Arbeitsgeschwindigkeit vor und dient zur Synchronisation.
- j) Erläutern Sie die beiden Begriffe Pipelining und Superskalar:
- Pipelining: Zeitliche Überlappung der Befehlsausführungsphasen zur Durchsatzerhöhung.
 - Superskalar: Wie Pipelining und zusätzlich mehrerer Ausführungseinheiten und deren dynamische Befehlszuweisung zur parallelen Ausführung mehrerer Befehle – im Idealfall Vervielfachung der Leistung.

Aufg. 1.01: Verständnisfragen Lsg. (6)

- k) Nennen und erläutern Sie die 5 Phasen einer typischen 5-stufigen DLX Pipeline in der korrekten Reihenfolge.
- 1) Instruction Fetch (IF): Befehl holen; Program counter (PC) erhöhen.
 - 2) Instruction Decode (ID): Instruktion dekodieren und Werte aus Registern lesen.
 - 3) Execute (EX): Ausführung der Operation/Adressberechnung in der ALU.
 - 4) Memory Access (MEM): Hauptspeicherzugriff; nur für Load-, Store- und Branch-Operationen.
 - 5) Write Back (WB): Ergebnis in Registersatz zurückschreiben.

Aufg. 1.01: Verständnisfragen Lsg. (7)

- l) Erklären Sie den Unterschied zwischen einem Interrupt und Polling.
- a) Interrupt: Kurzfristige Unterbrechung des Programms durch Interrupt Request (IRQ), in Hardware gesteuert durch Interrupt Controller. Abarbeiten von zeitkritischen Aufgaben in Interrupt Service Routine (ISR).
→ Laufzeiteffizient, aber zusätzliche Hardware nötig.
 - b) Polling: Programmierte zyklische Abfrage eines Status.
→ Einfach, keine zusätzliche Hardware nötig, belegt aber ständig CPU-Leistung.

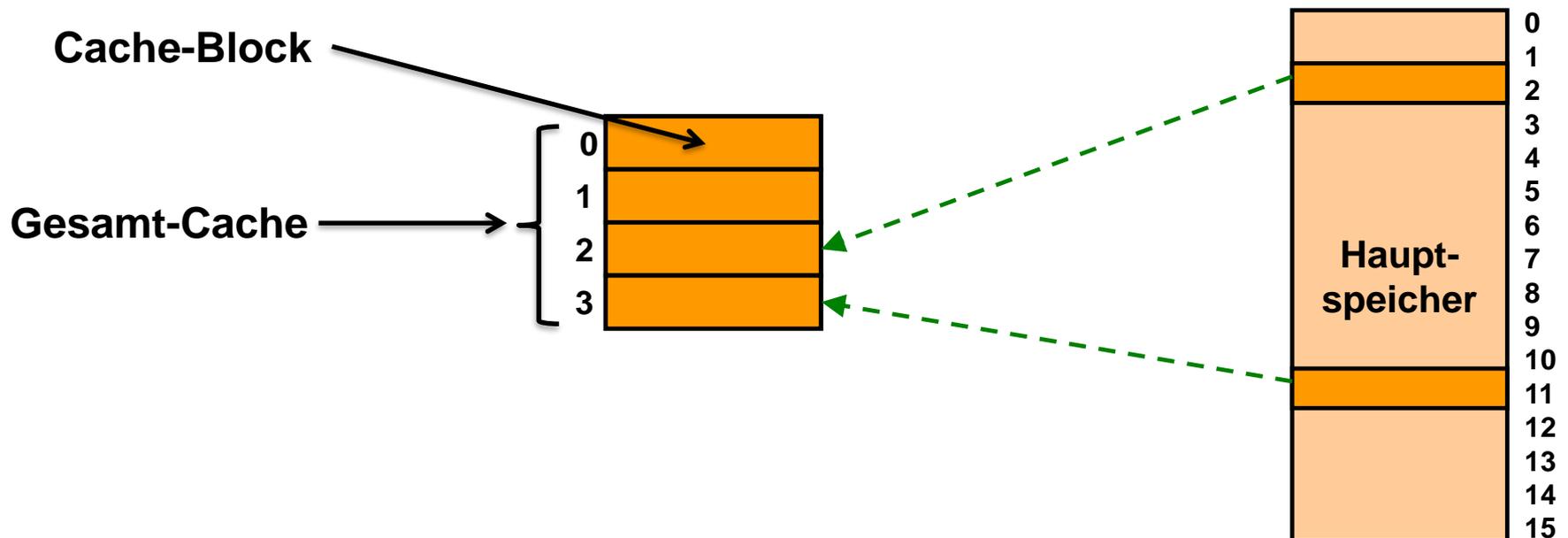
- Aufg. 1.01: Verständnisfragen



Aufg. 1.02: Cacheorganisation Lsg. (1)

- a) Nennen Sie die drei Möglichkeiten der Cacheorganisation. Welche Vor- und Nachteile haben diese?

Direct Mapped: Ein Block kann nur an einer Adresse mod m ($m = \text{Cache-Größe}$, hier: 4) gespeichert werden.

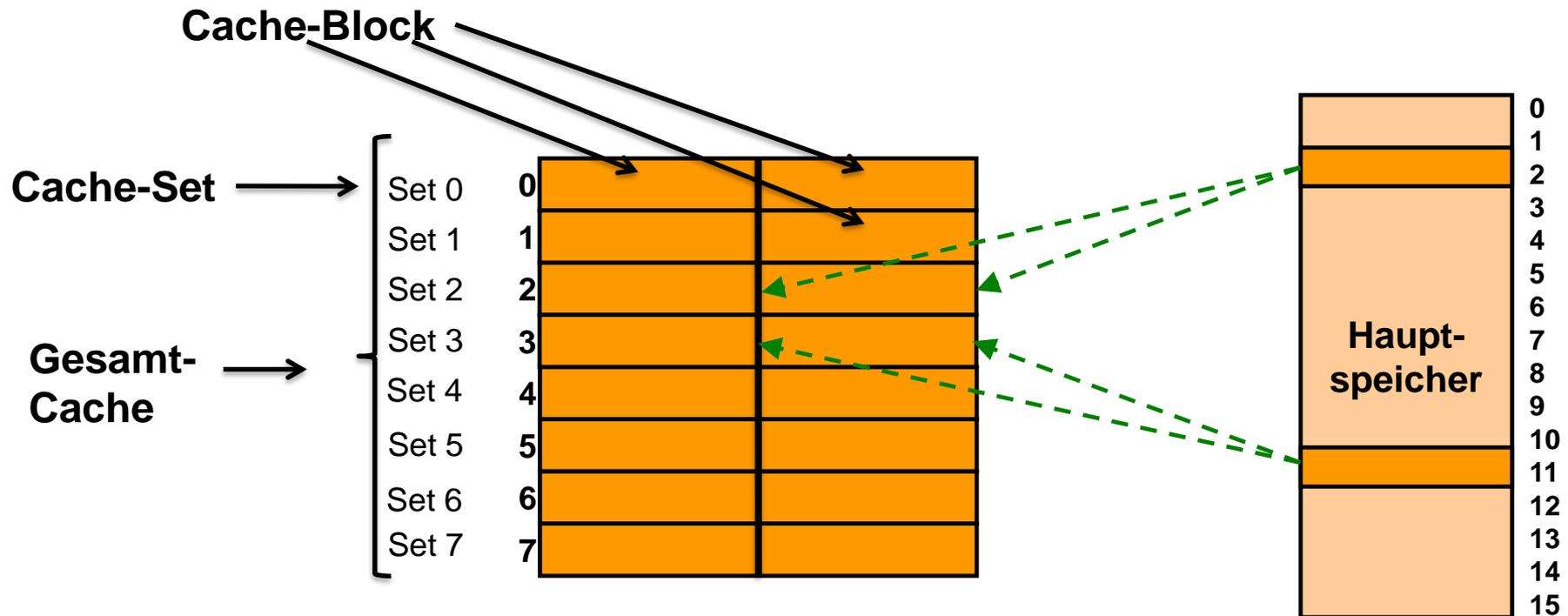


Aufg. 1.02: Cacheorganisation Lsg. (2)

N -Wege assoziativ: Ein Block kann in einem von N Cache-Blöcken gespeichert werden, bei denen die Hauptspeicheradresse mod S gleich der Cache-Adresse mod S ist (S = Anzahl der Cache-Sets).

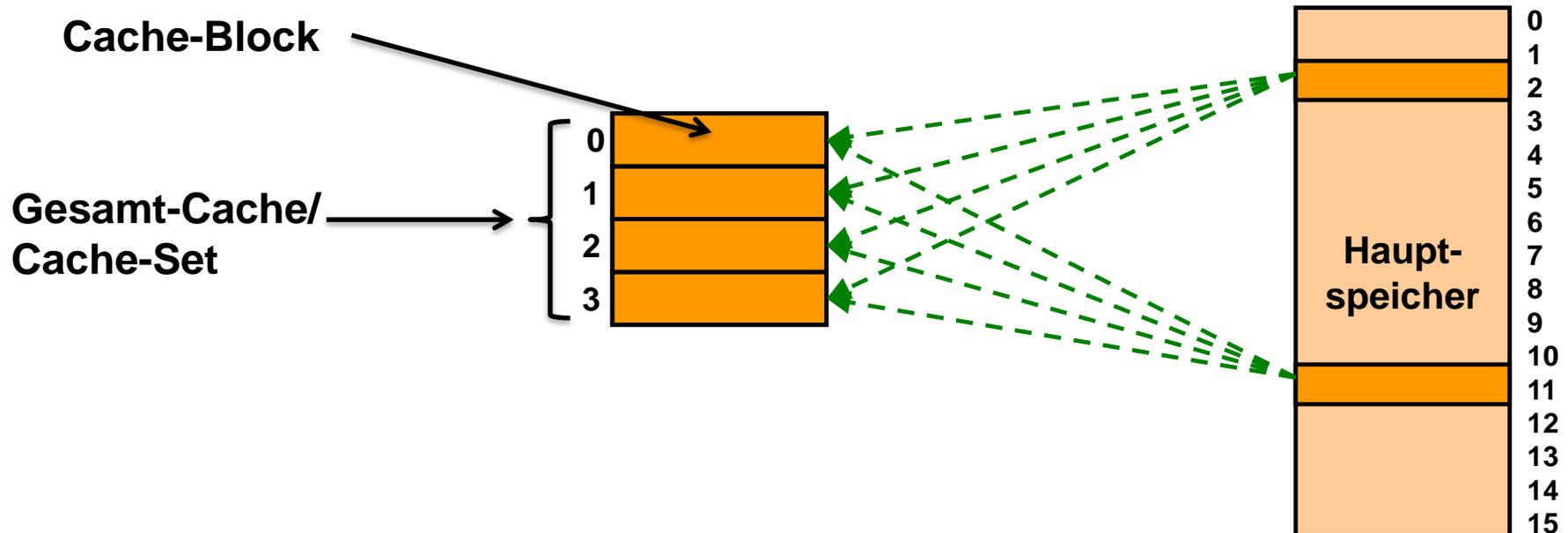
Cache-Größe $m = S * N$.

Beispiel: 2-Wege assoziativ



Aufg. 1.02: Cacheorganisation Lsg. (3)

Voll-assoziativ: Jeder Block kann an allen Adressen gespeichert werden. Voll-assoziativ ist N -Wege assoziativ mit $N = m = \text{Anzahl der Cache-Blöcke}$.



Aufg. 1.02: Cacheorganisation Lsg. (4)

Vor- und Nachteile: N -Wege assoziativ hat den Nachteil, dass $N-1$ mehr Komparatoren als direkt-abbildende Caches benötigt werden, was die Hardware komplexer macht und die benötigte Chipfläche erhöht. Vorteil ist aber die höhere Flexibilität beim Zuweisen von Daten aus dem Hauptspeicher in den Cache, da die Daten N verschiedenen Cache-Sets zugewiesen werden können. Dies führt zu einer höheren Cache-Hit-Rate als bei direkt-abbildenden Caches, da bei letzteren jede Adresse im Hauptspeicher genau einem Block im Cache zugewiesen ist.

Aufg. 1.02: Cacheorganisation Lsg. (5)

b) Berechnen Sie für jeden Typ die Gesamtanzahl der Cache-Blöcke und die Anzahl der Cache-Sets.
Hinweis: Beim Cachetyp, bei dem eine Variable N festgelegt werden soll, nehmen Sie den Wert 4 an.

1. Direkt-Abbildend (direct mapped: DM)

$$\underline{\text{Anzahl Cache-Blöcke} = 512 \text{ KB} / 8 \text{ B} = 65536 = 2^{16}}$$

$$\underline{\text{Anzahl Cache-Sets} = n = 65536 = 2^{16}}$$

2. 4- Weg assoziativ (set associative: SA)

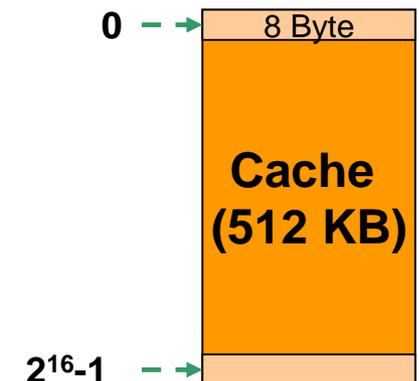
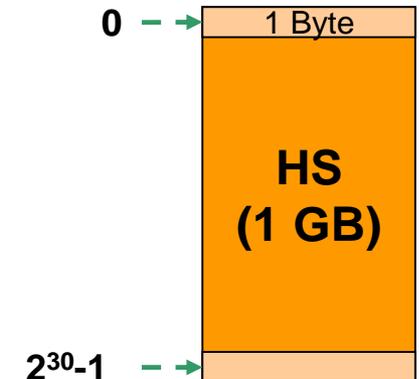
$$\underline{\text{Anzahl Cache-Blöcke} = 512 \text{ KB} / 8 \text{ B} = 65536 = 2^{16}}$$

$$\underline{\text{Anzahl Cache-Sets} = n = 4}$$

3. Voll-assoziativ (fully associative: FA)

$$\underline{\text{Anzahl Cache-Blöcke} = 512 \text{ KB} / 8 \text{ B} = 65536 = 2^{16}}$$

$$\underline{\text{Anzahl Cache-Sets} = 1}$$

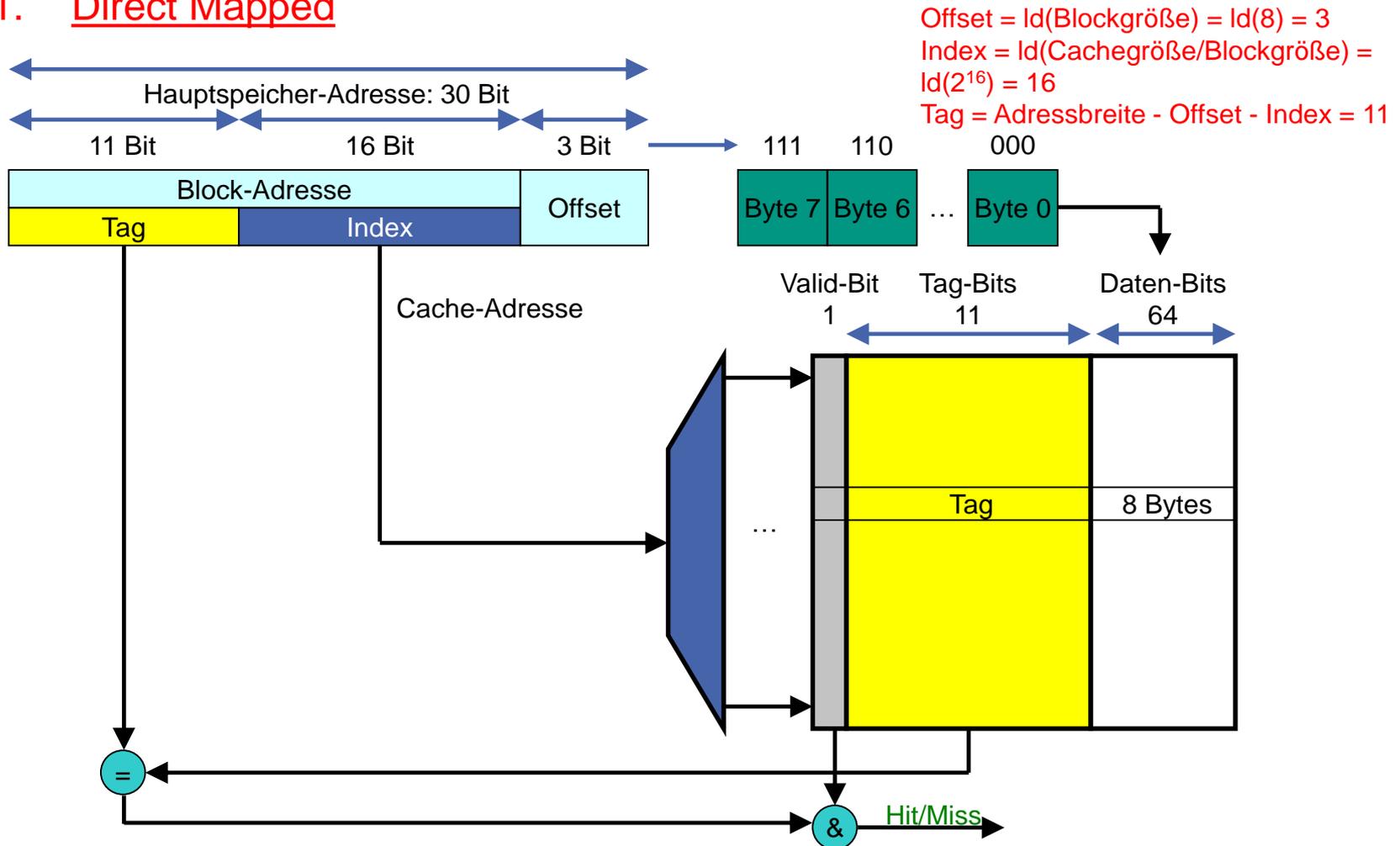


Aufg. 1.02: Cacheorganisation Lsg. (6)

- c) Erläutern Sie zu jedem Typ die Speicherorganisation des Caches bezüglich im Cache abzulegende Daten und Adressierung
Hinweis: Verwenden Sie als Hilfsmittel die Abbildungen zur Cacheorganisation aus der Vorlesung.

Aufg. 1.02: Cacheorganisation Lsg. (7)

1. Direct Mapped



Cacheorganisation

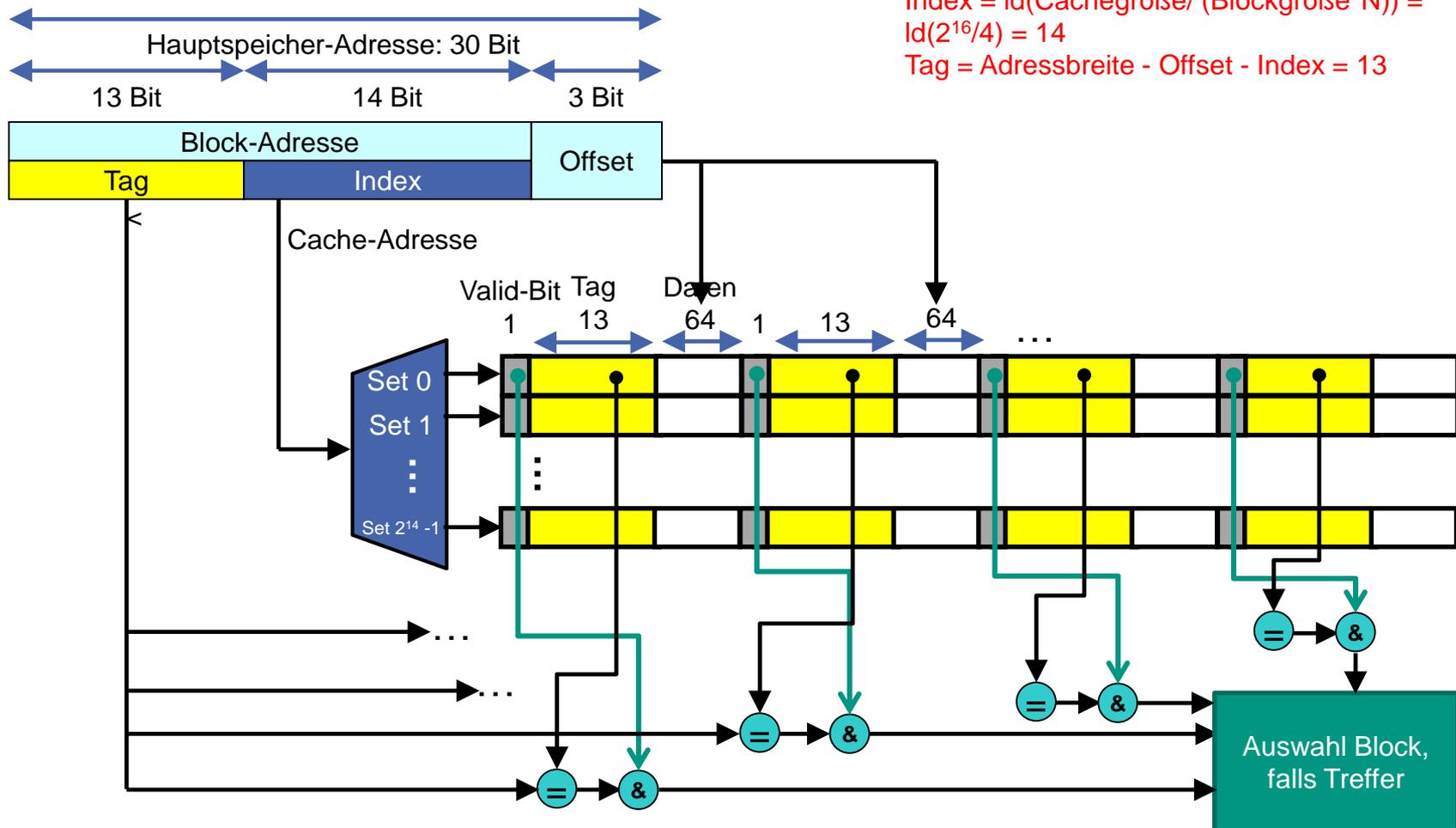
■ Beispiel: Direct Mapped

- Cache ist in Form einer Tabelle mit mehreren Datenworten pro Zeile (Block) organisiert
- Hauptspeicheradresse von insgesamt 30 Bit Breite setzt sich zusammen aus:
 - Offset (3 Bits): Adressiert ein 8-Bit Datenwort innerhalb eines Blockes, wobei jeder Block z.B. $2^3 = 8$ Datenworte enthalten kann.
 - Index(16 Bit): Adressiert einen Block innerhalb des Caches, die Gesamtzahl der Blöcke beträgt $2^{16} = 65536$.
 - Tag (11 Bit): Dient als Referenz im Cache, um die Gültigkeit des aktuellen Eintrages festzustellen. Ist dies nicht der Fall, werden die alten Werte im Cache bei Bedarf in den Hauptspeicher übertragen und die aktuell referenzierten Daten aus dem Hauptspeicher in den Cache geschrieben.
- Das Valid-Bit gibt an, ob in dem betreffenden Block die Daten gültig sind, d.h. ob der entsprechende Block in Benutzung ist, oder ob der Block frei ist und einfach überschrieben werden kann
- Die gesamte Größe des Caches errechnet sich aus der Anzahl der Zeilen, Anzahl der Datenworte pro Zeile und der Größe der Datenworte: $65536 * (8 * 1) = 512$ KBytes, hinzu kommen die Verwaltungsbits

Aufg. 1.02: Cacheorganisation Lsg. (8)

2. 4-Wege assoziativ

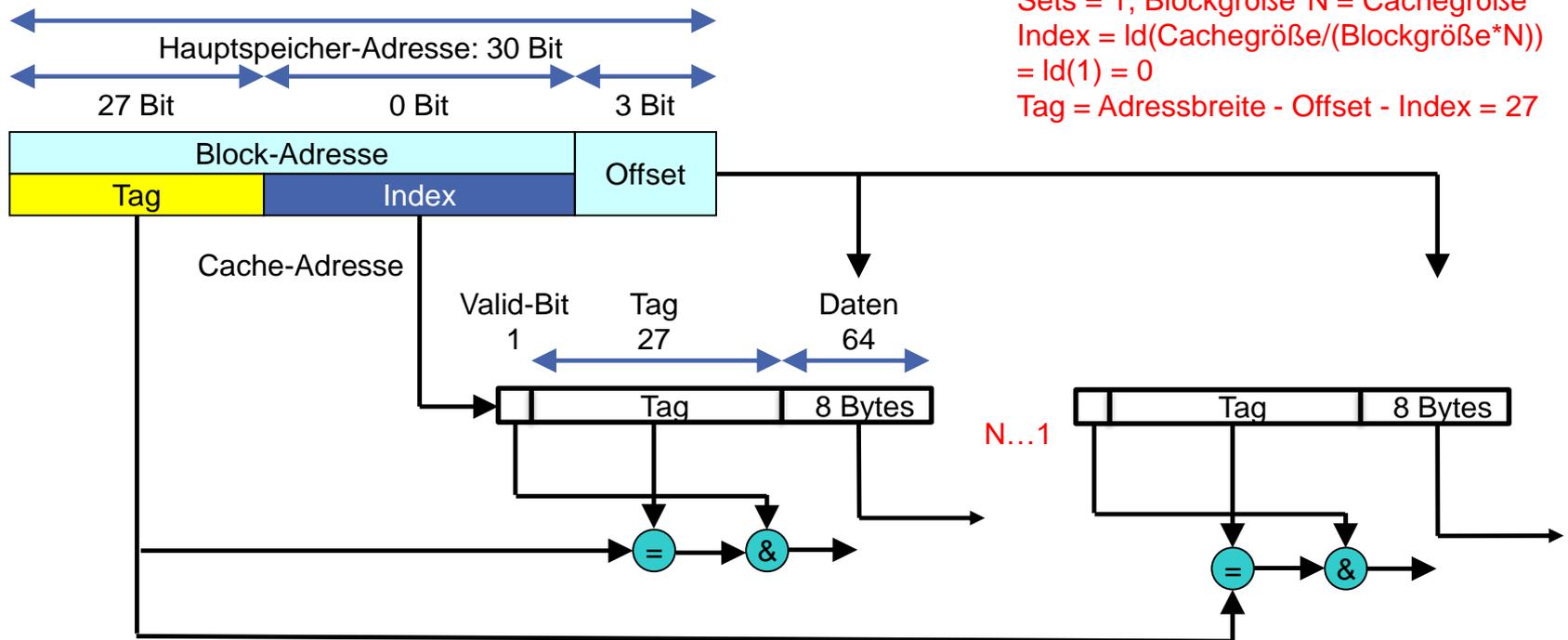
$\text{Offset} = \text{ld}(\text{Blockgröße}) = \text{ld}(8) = 3$
 $\text{Index} = \text{ld}(\text{Cachegröße} / (\text{Blockgröße} * N)) = \text{ld}(2^{16}/4) = 14$
 $\text{Tag} = \text{Adressbreite} - \text{Offset} - \text{Index} = 13$



Aufg. 1.02: Cacheorganisation Lsg. (9)

3. Voll-assoziativ

$\text{Offset} = \text{ld}(\text{Blockgröße}) = \text{ld}(8) = 3$
 $\text{Sets} = 1, \text{Blockgröße} * N = \text{Cachegröße}$
 $\text{Index} = \text{ld}(\text{Cachegröße} / (\text{Blockgröße} * N)) = \text{ld}(1) = 0$
 $\text{Tag} = \text{Adressbreite} - \text{Offset} - \text{Index} = 27$



Aufg. 1.02: Cacheorganisation Lsg. (10)

- d) Berechnen Sie für jeden Typ die in Wirklichkeit notwendige Speicherkapazität für den Cache Baustein

Speichergröße = Datenspeicher + Speicher für Tags + Speicher für Valid-Bits

Direkt-Abbildend (direct mapped: DM)

Speichergröße = 512 KB + (12 Bit * 2¹⁶) / (8 Bit/Byte) = 512 KB + 96 KB = 608 KB

4-Wege assoziativ (set associative: SA)

Speichergröße = 512 KB + (14 Bit * 2¹⁶) / (8 Bit/Byte) = 512 KB + 112 KB = 624 KB

Voll-assoziativ (fully associative: FA)

Speichergröße = 512 KB + (28 Bit * 2¹⁶) / (8 Bit/Byte) = 512 KB + 224 KB = 736 KB

Aufg. 1.02: Cacheorganisation Lsg. (11)

Der benötigte Speicher ist offensichtlich nicht der ausschlaggebende Faktor bei der Auswahl! Was sonst?

Trade-off: Anzahl der benötigten Komparatoren (Logik-Platz)

vs.

Höhere Flexibilität beim Mapping (== höhere Cache-Hit-Rate)

- Aufg. 1.02: Cacheorganisation



Fragen?

Vielen Dank für Ihre Aufmerksamkeit



Marc Weber
Karlsruher Institut für Technologie (KIT) – ITIV
marc.weber3@kit.edu