

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Bitprozessor hat Harvard-Architektur

Funktionsweise eines von-Neumann Rechners

Beispielprogramm 1

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Im nachfolgenden Modell sind, wegen der Übersichtlichkeit, nicht alle Signale und Register dargestellt.

Der Befehlssatz orientiert sich am 8080/Z80. Die Mnemonic entspricht der des Z80. Auch hier sind nur wenige Befehle beschrieben.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

In diesem Funktionsmodell sind einige Vereinfachungen* gemacht worden, um die Übersichtlichkeit etwas zu verbessern.

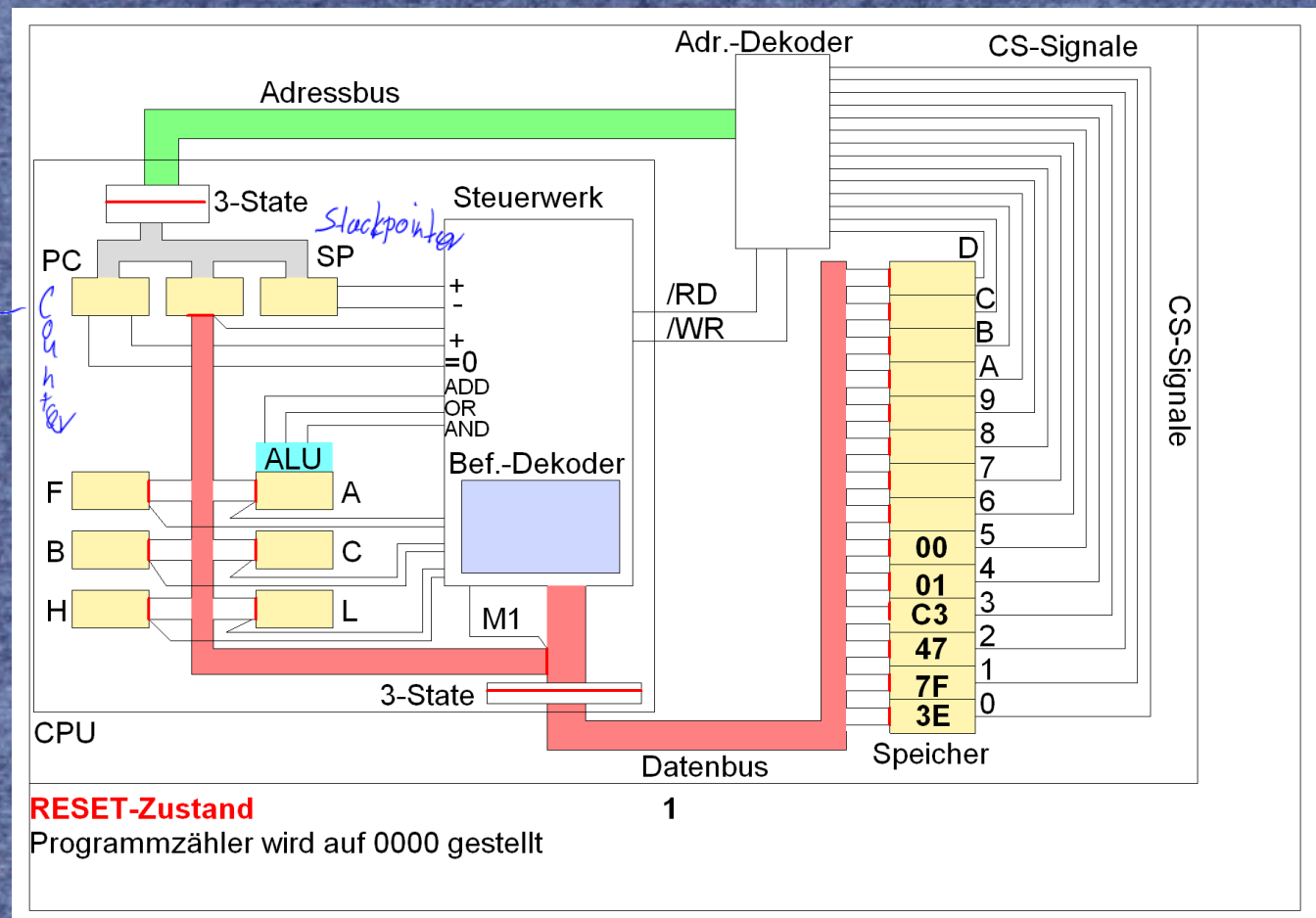
Dies sind u.a.

Der Programmspeicher umfasst nur wenige Adressen

/RD und /WR sind an den Adressdekode geführt anstatt an die Speicherbausteine

Nur wenige logische u. arithmetische Befehle sind angedeutet

Die Anzahl des Register ist stark eingeschränkt



*gegenüber einer realen Z80 CPU

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Kennzeichen eines von-Neumann Rechners:

- gemeinsamer Adressraum für Programm und Daten
(=> ein Adress-, ein Daten- und ein Steuerbus)
- Keine Überlappung von Execute[Bef.1] und Fetch[Bef.2] möglich
- Daten können als Programm interpretiert werden
- leichte Realisierung und Zugriff auf Konstantenlisten

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Abarbeitung eines Befehls

- 1. Befehlholzyklus (Fetch)
Ein Befehl wird in den Befehlsdekoder geladen
- 2. Dekodierzyklus (Decode)
Der Befehl wird analysiert, ggf. weitere Argumente aus dem Speicher geladen
- 3. Ausführzyklus (Execute)
Der Befehl wird ausgeführt. Dabei sind u.U. wieder Speicherzugriffe notwendig.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Das Beispielprogramm 1:

- 0000 3E 7F LD A,7FH
- 0002 47 LD B, A
- 0003 C3 01 00 JP 0001H

In den Akku (A-Register) wird die Konstante 7FH geladen. Diese Konstante steht als Argument direkt beim Befehl. Dann wird der Inhalt von A nach Register B kopiert und zur Adresse 1 gesprungen. Hier steht kein gültiger Befehl sondern das Argument vom ersten Befehl. Dennoch liest der μ P diesen Wert als Befehl (LD A,A) ein und führt ihn auch aus, was hier aber keine neg. Auswirkung hat.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Der Programmablauf startet

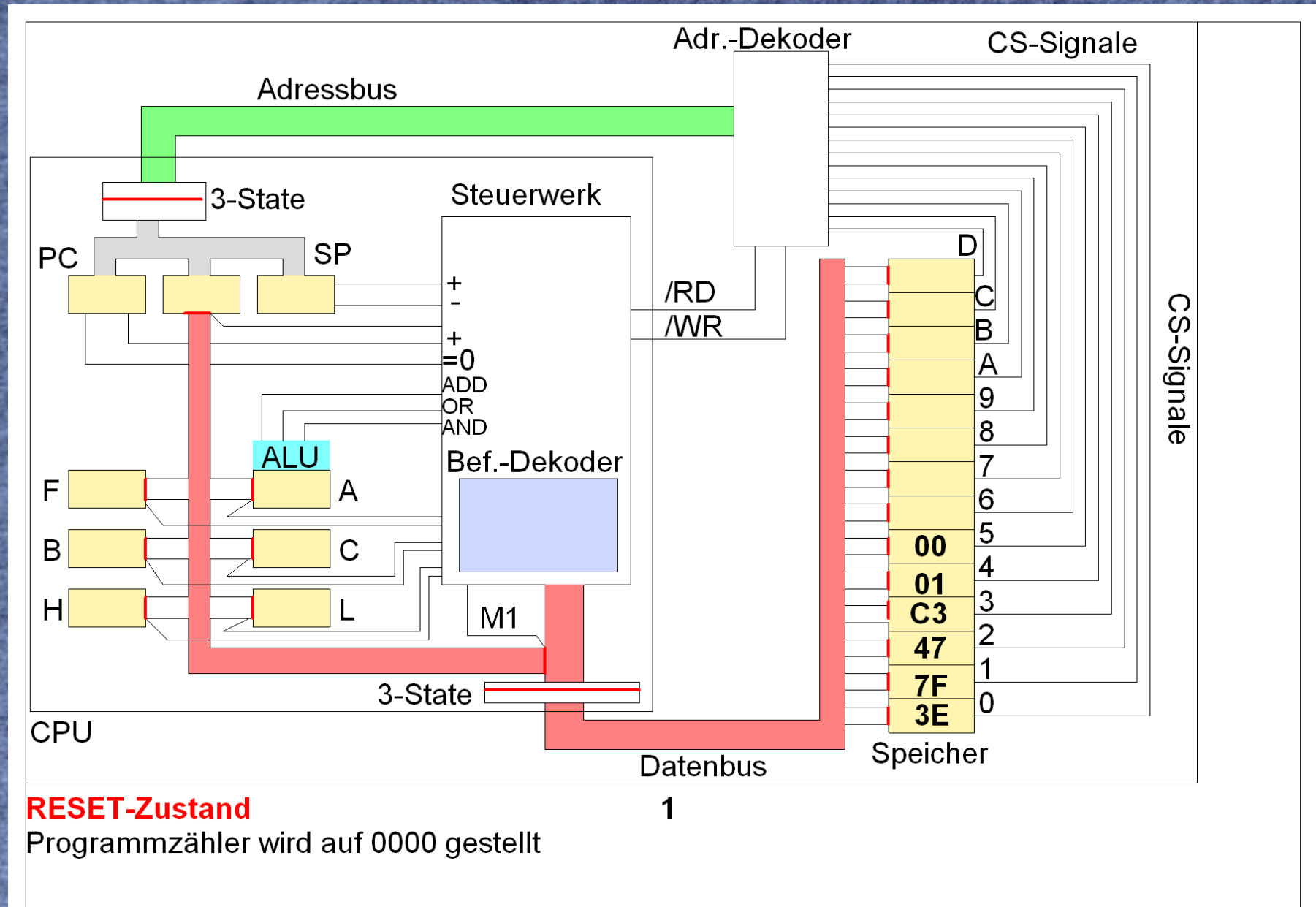
Nach einem RESET steht der Programmzähler auf 0000. Von dort wird der erste Befehl in den Befehls-dekoder geladen, analysiert und ausgeführt.

Da der Befehl ein Argument benötigt, muss nach dem Dekodieren ein weiteres Mal auf den Speicher zugegriffen werden. Damit diese Zugriff auch die richtige Adresse trifft, muss der Programmzähler bereits auf die nächste Adresse zeigen

Rechnertechnik I

TINF DHBW Karlsruhe

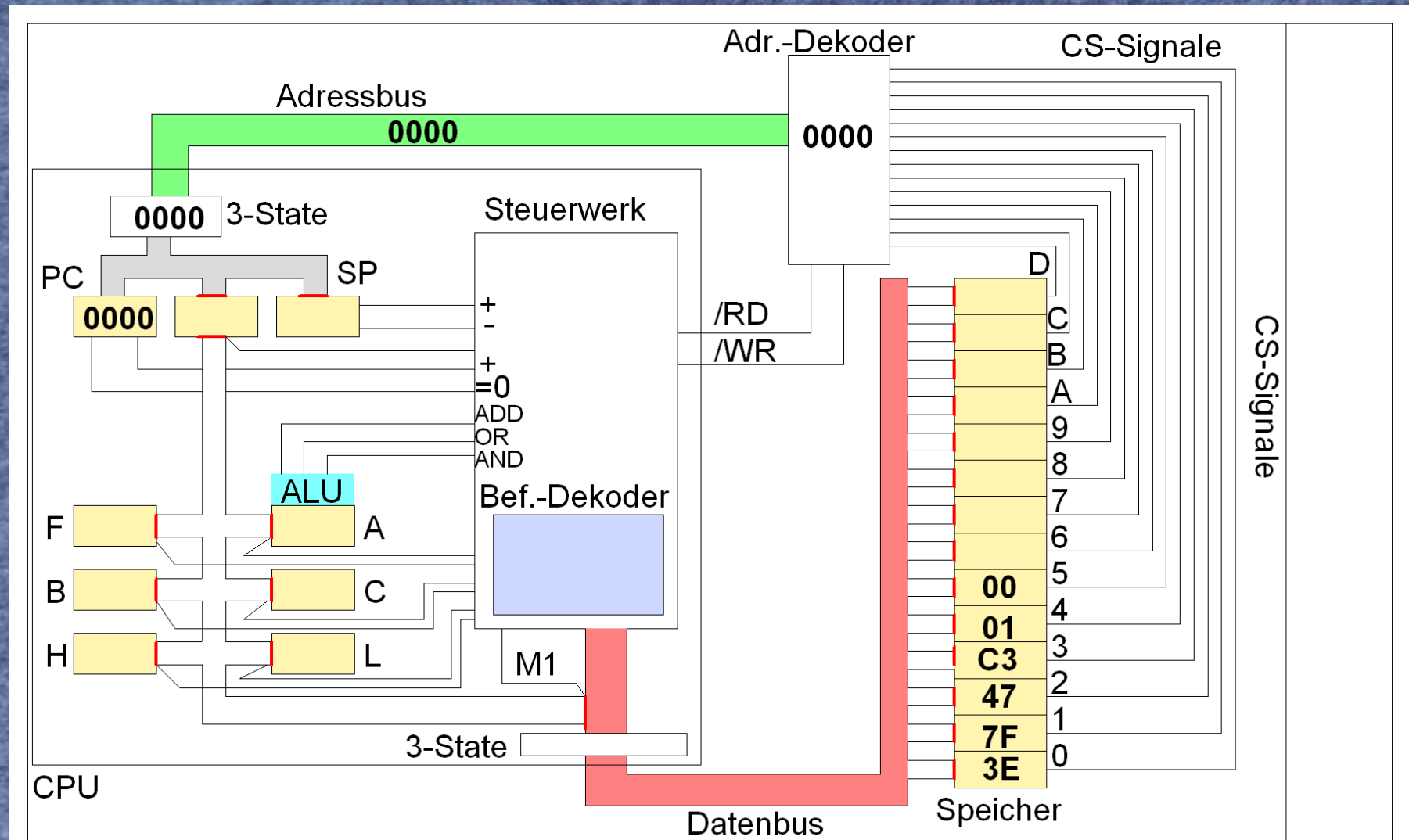
© Stefan Lehmann



Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

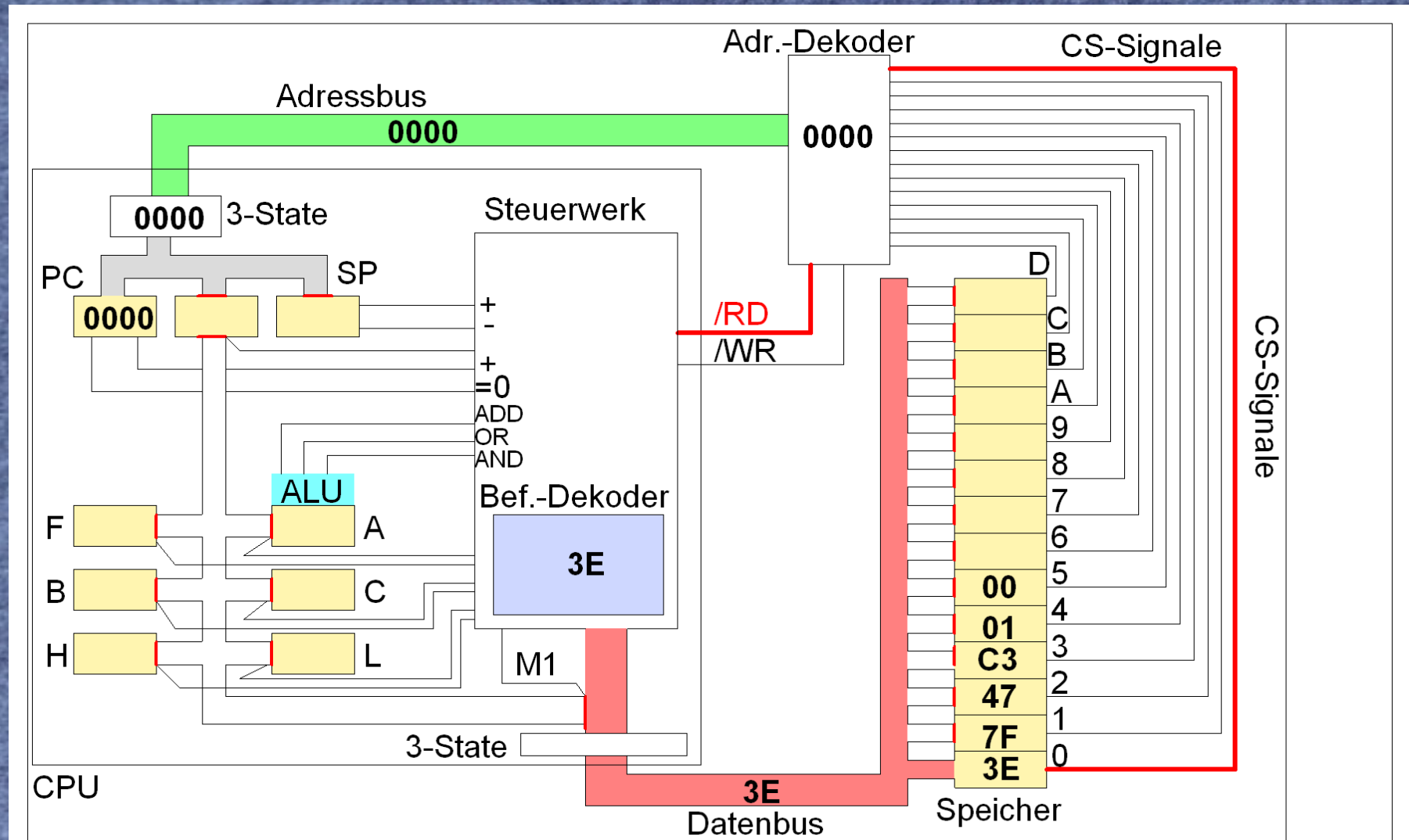
2

Schritt 1: Inhalt des PC auf den Adressbus legen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

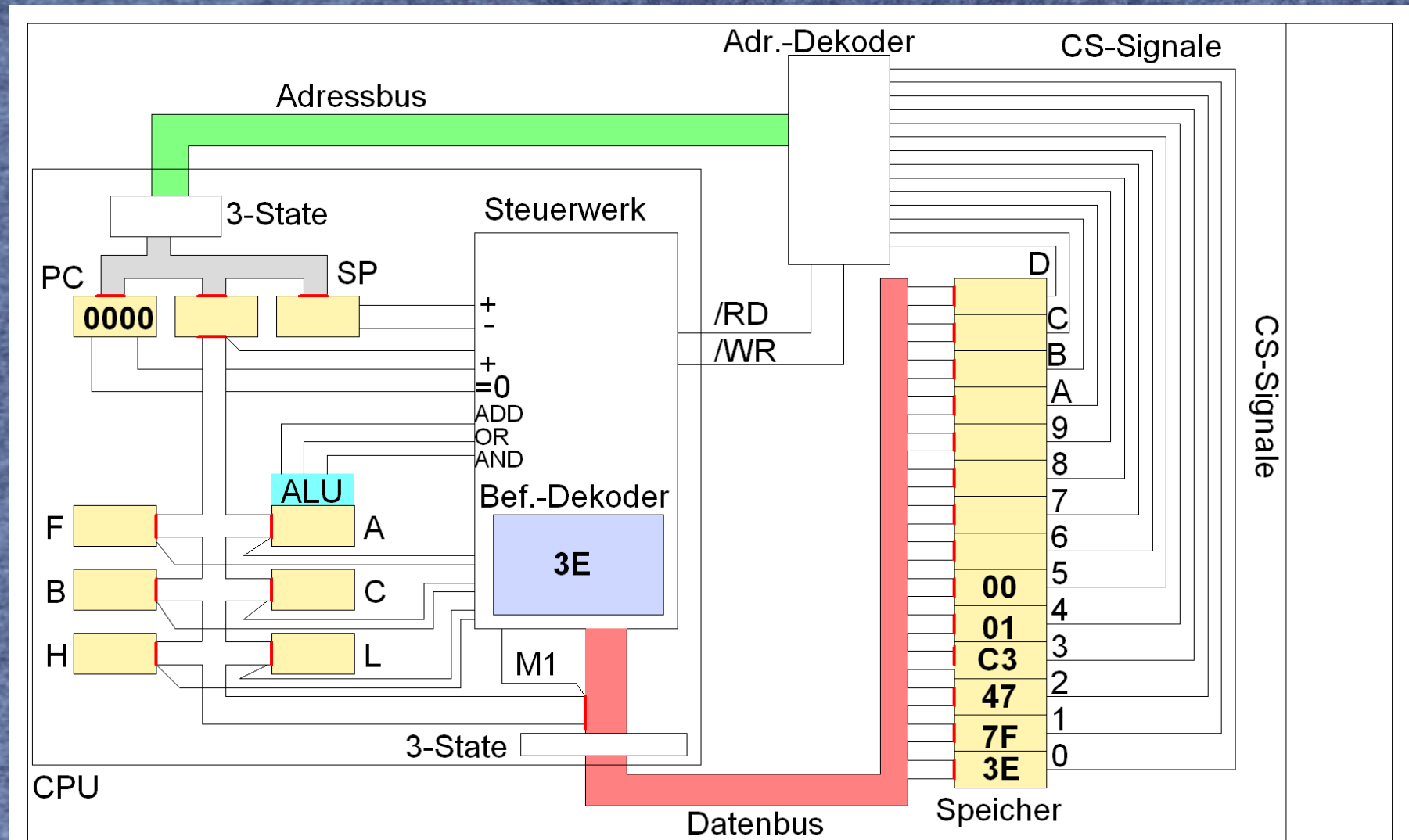
3

Schritt 2: Adr.-Dekoder gibt entsprechende Speicherzelle frei
deren Inhalt gelangt über den Datenbus in den Befehlsdekoder

Rechnertechnik I

TINF DHBW Karlsruhe

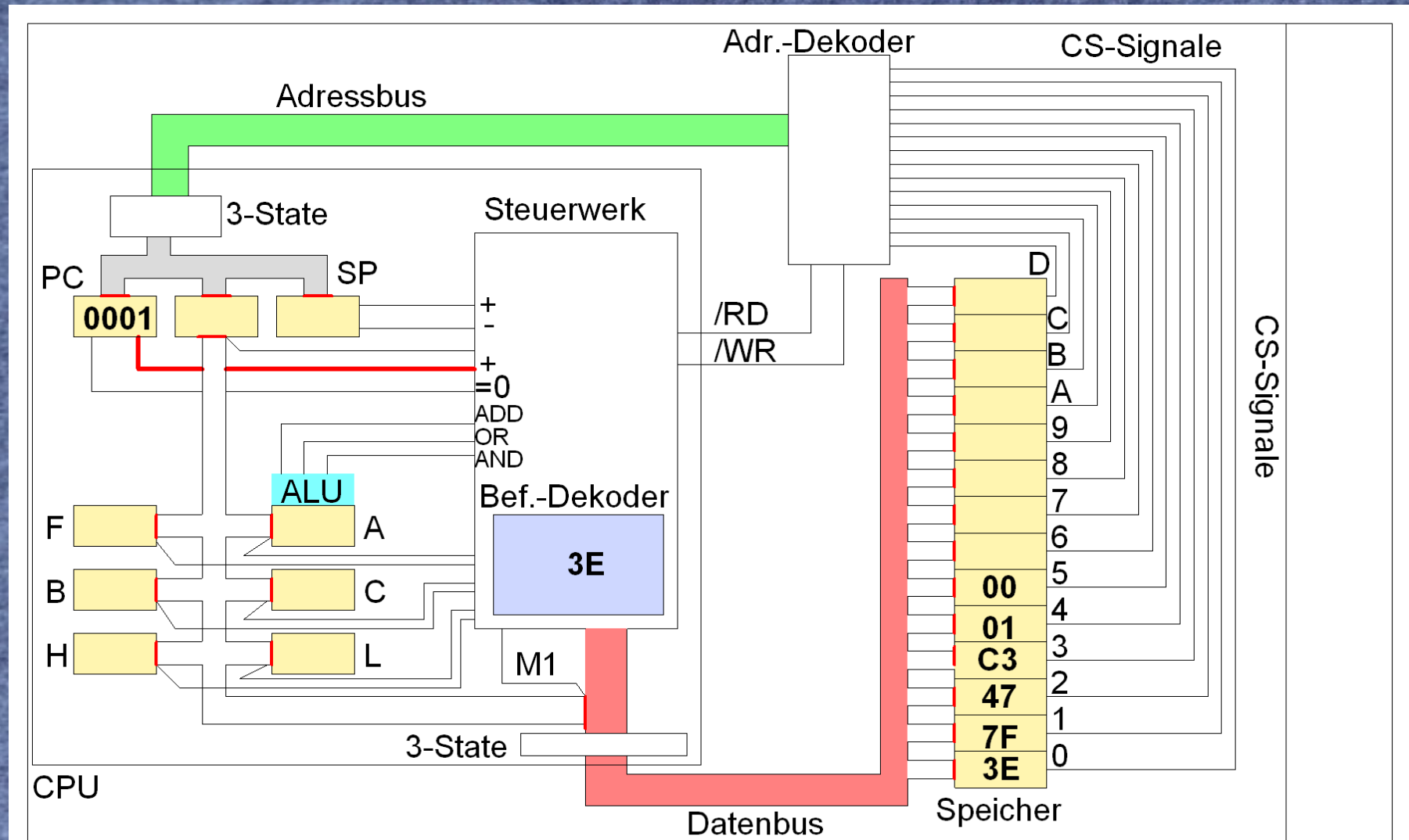
© Stefan Lehmann



Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



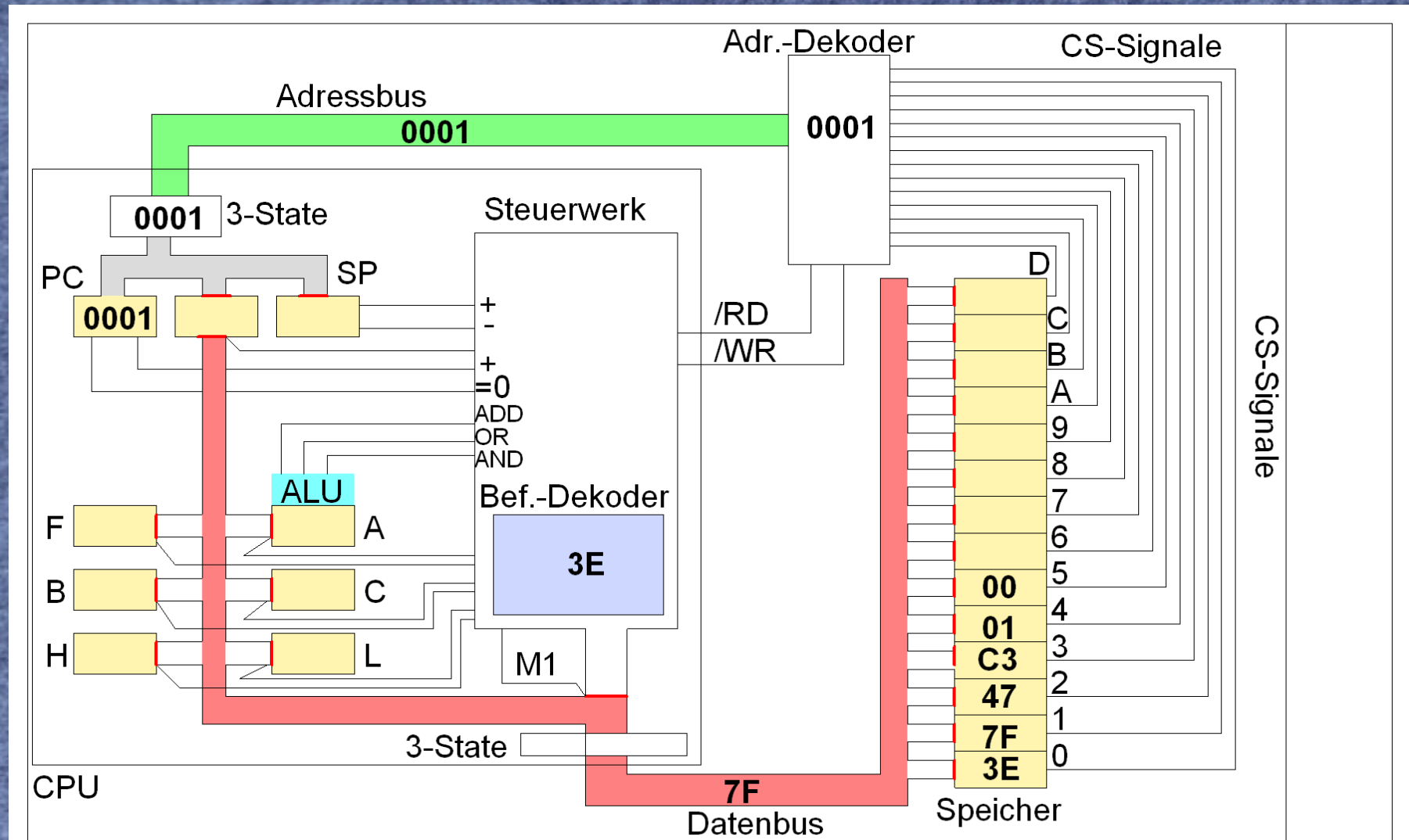
Fetch-Zyklus (M1)

Schritt 4: Programmzähler um 1 erhöhen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Dekode-Zyklus

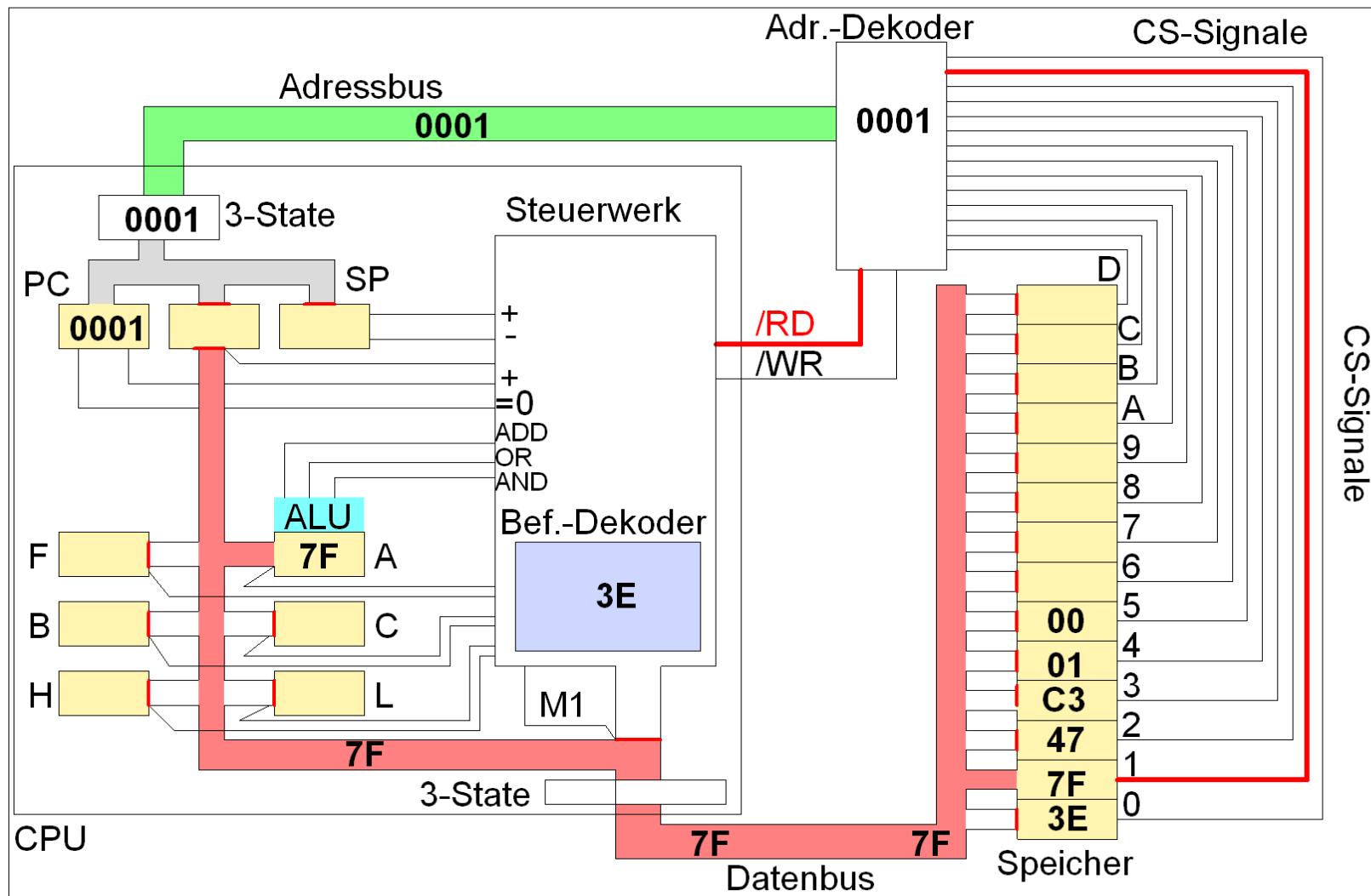
6

Steuerwerk erkennt, dass ein Argument eingelesen werden muss
deshalb wird erneut vom Speicher (Adresse 0001) gelesen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Execute-Zyklus

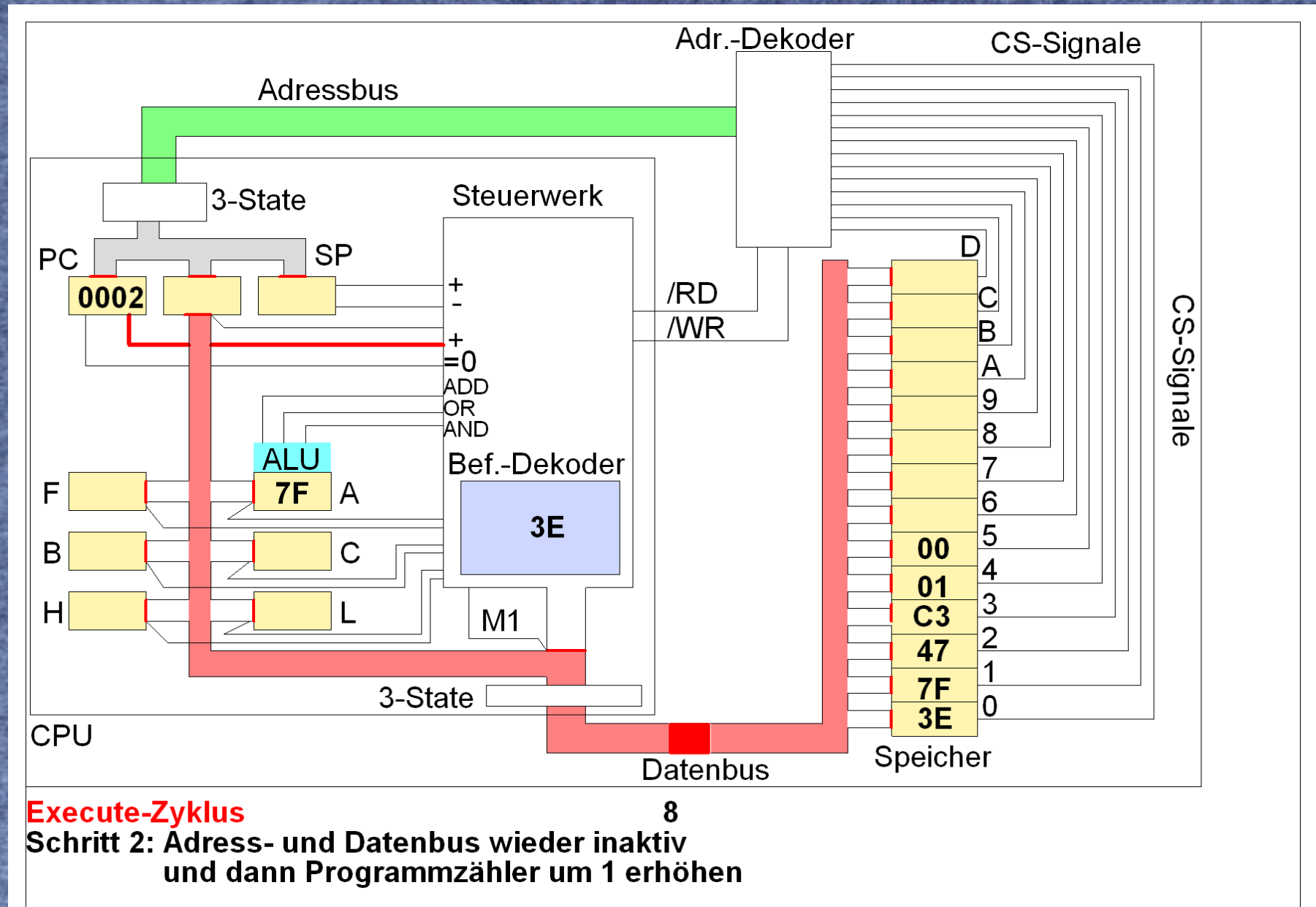
7

Schritt 1: Adressierung und Lesen der Speicherstelle 0001
der gelesene Wert kommt ins A-Register

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Damit wäre der erste Befehl abgearbeitet.

Die Auswirkungen waren:

- Programmzähler steht jetzt auf 0002
- Im Akku steht der Wert 7FH
- ab jetzt weiß der μ P nicht mehr, dass sein letzter Befehl LD A, 7FH war

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Nun wird der nächste Befehl eingelesen

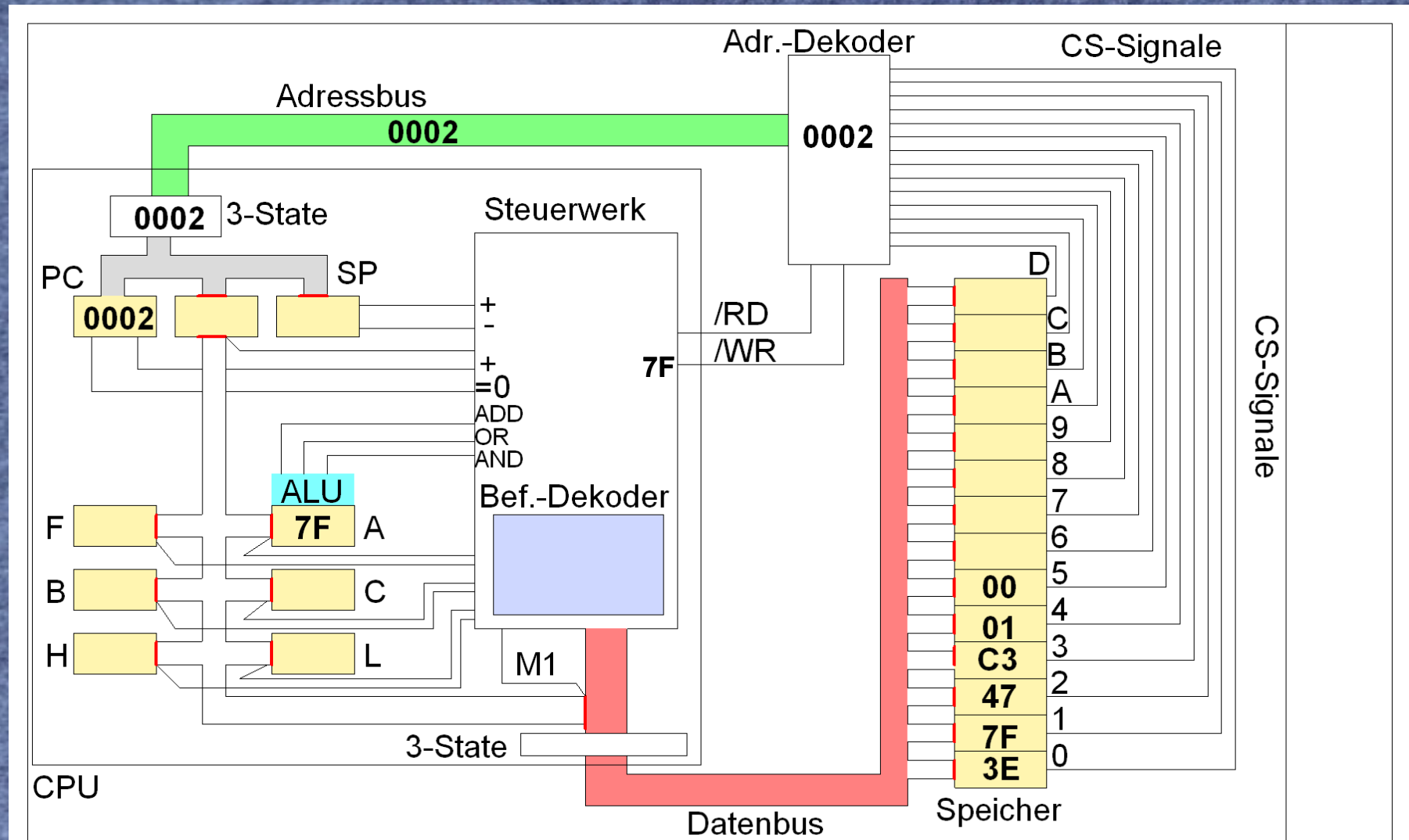
Dieser Befehl besteht nur aus dem Befehlscode, hat also keine weiteren Argumente

LD B, A kopiert den Inhalt von A ins B-Register

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



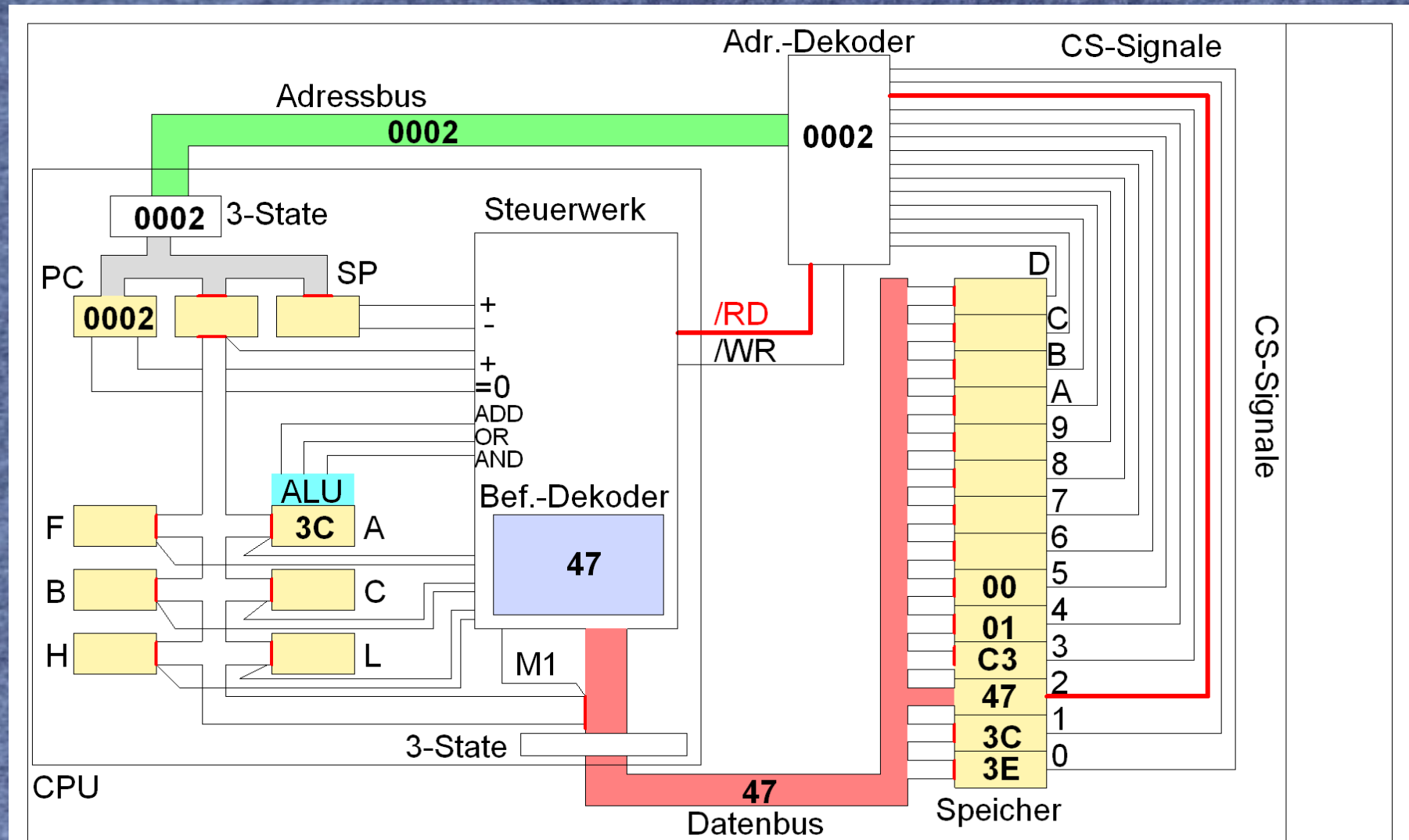
Fetch-Zyklus (M1)

Schritt 1: Inhalt des PC auf den Adressbus legen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

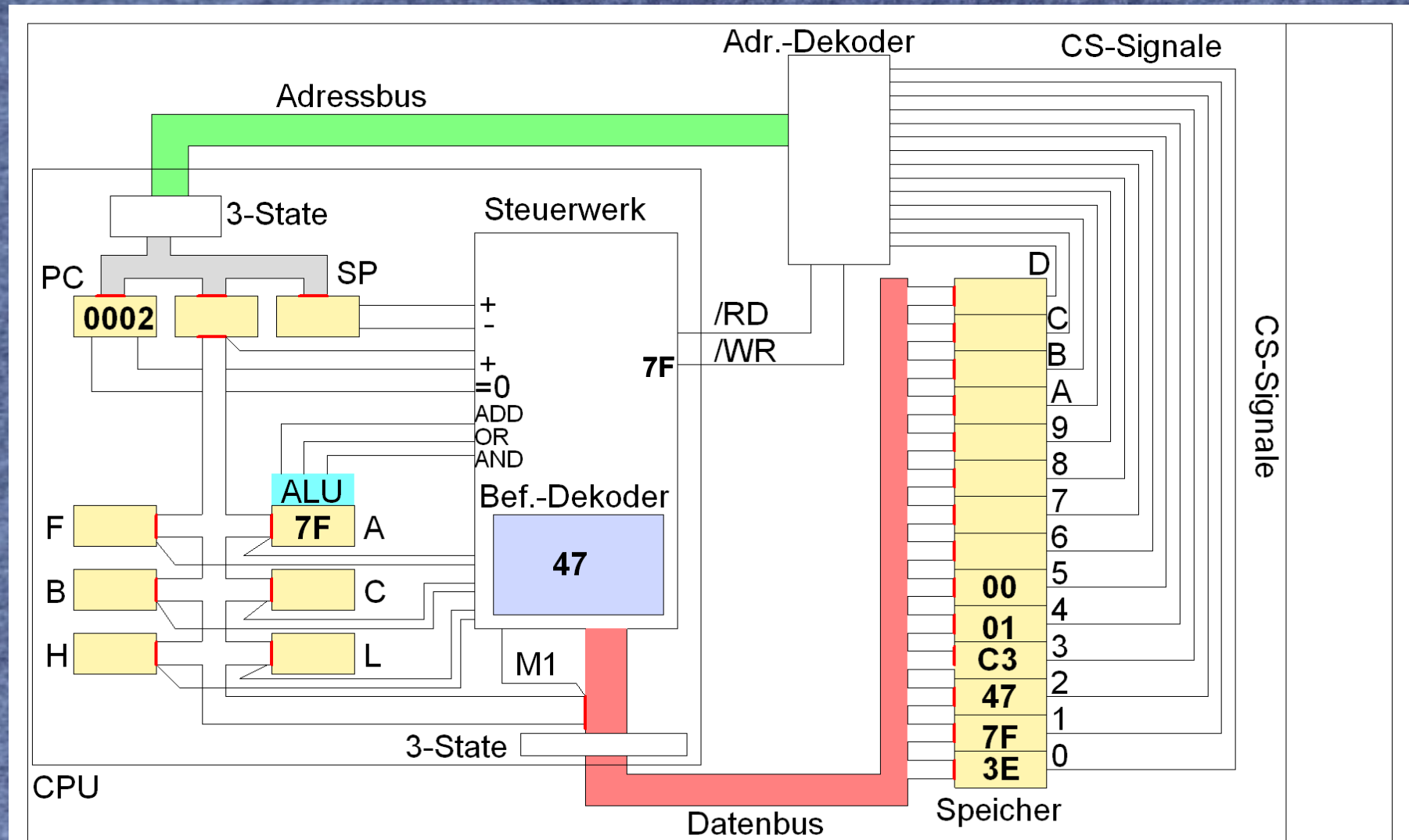
10

Schritt 2: Adr.-Dekoder gibt entsprechende Speicherzelle frei
deren Inhalt gelangt über den Datenbus in den Befehlsdekoder

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

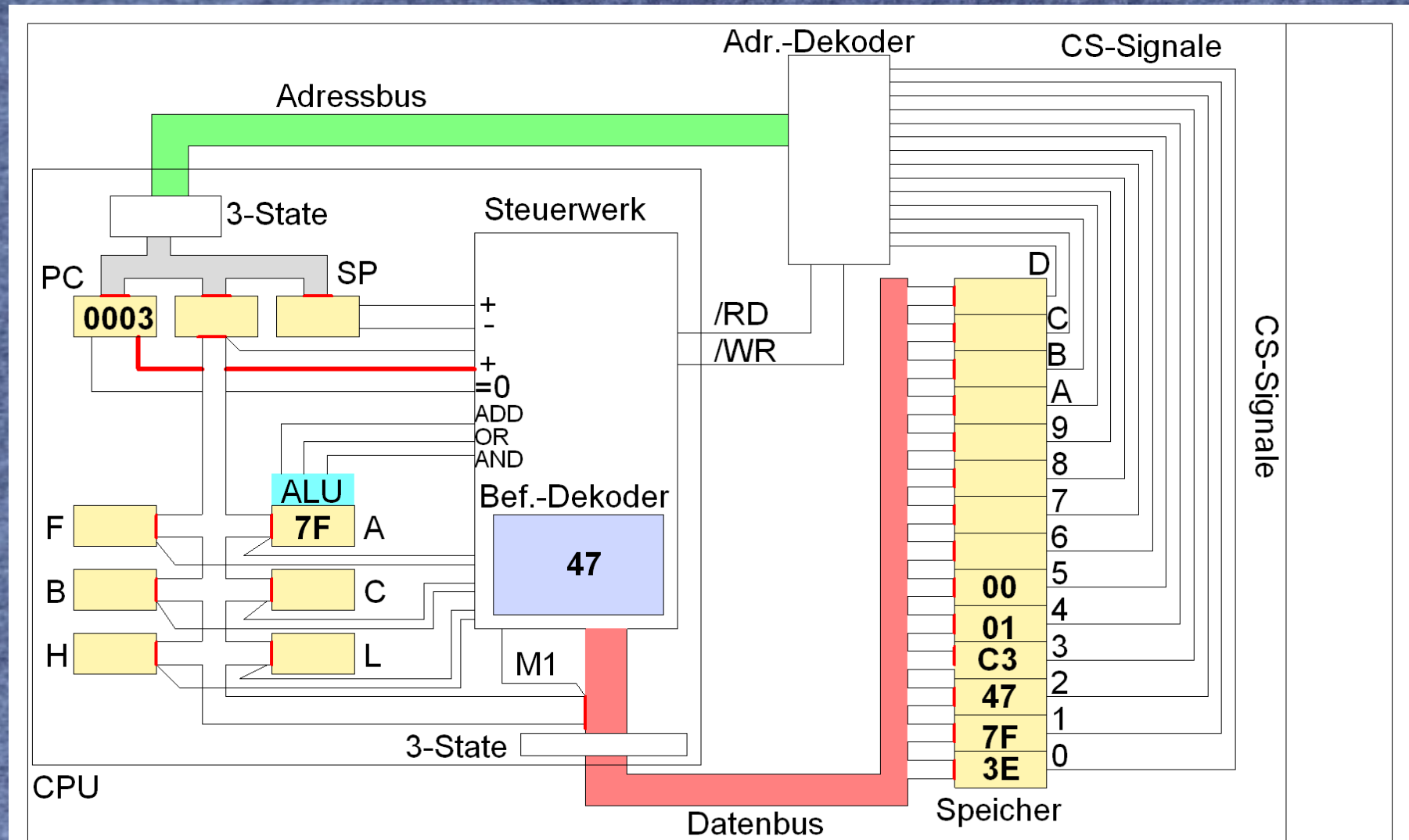
11

Schritt 3: Adress- und Datenbus wieder inaktiv

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



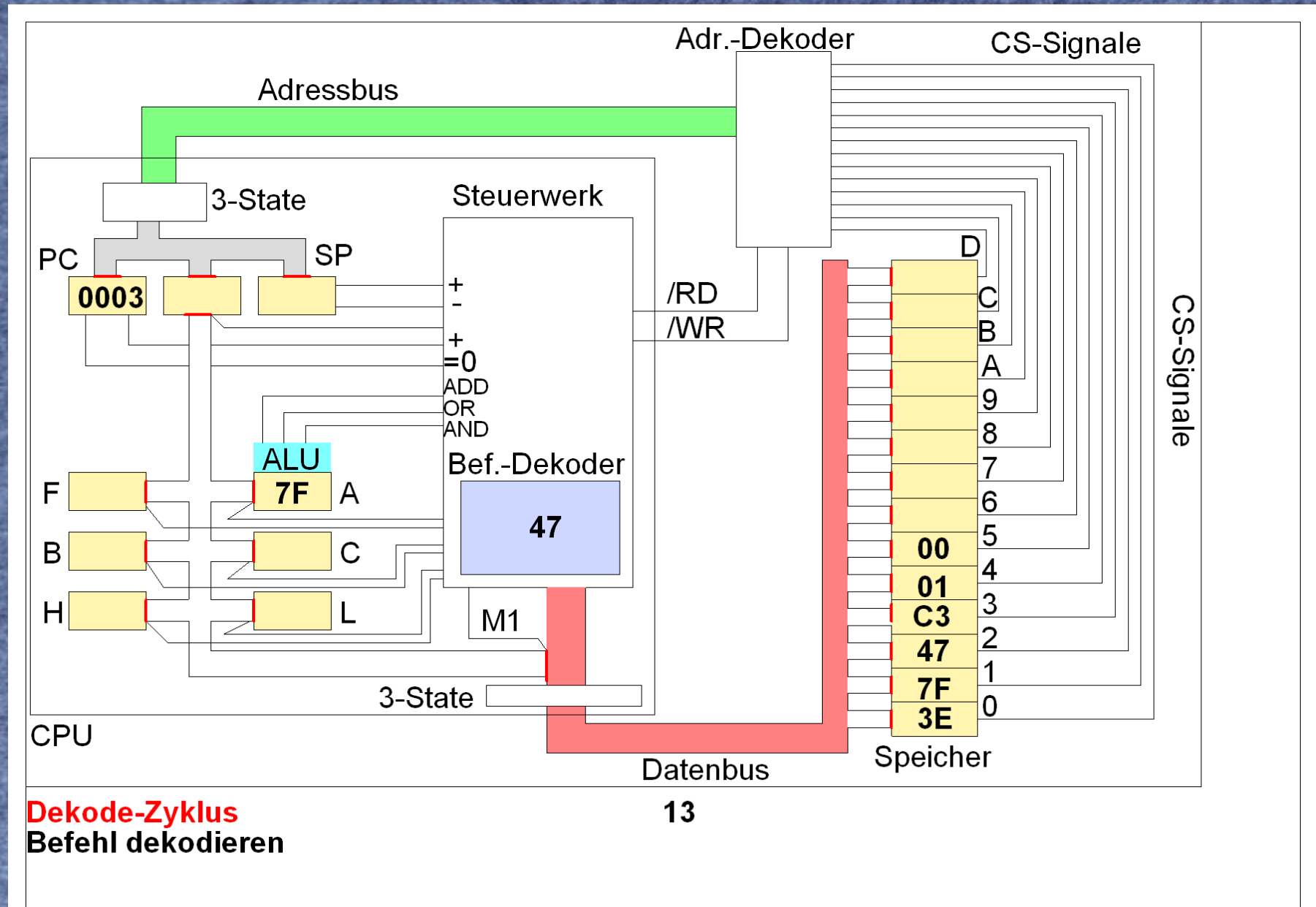
Fetch-Zyklus (M1)

Schritt 4: Programmzähler um 1 erhöhen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

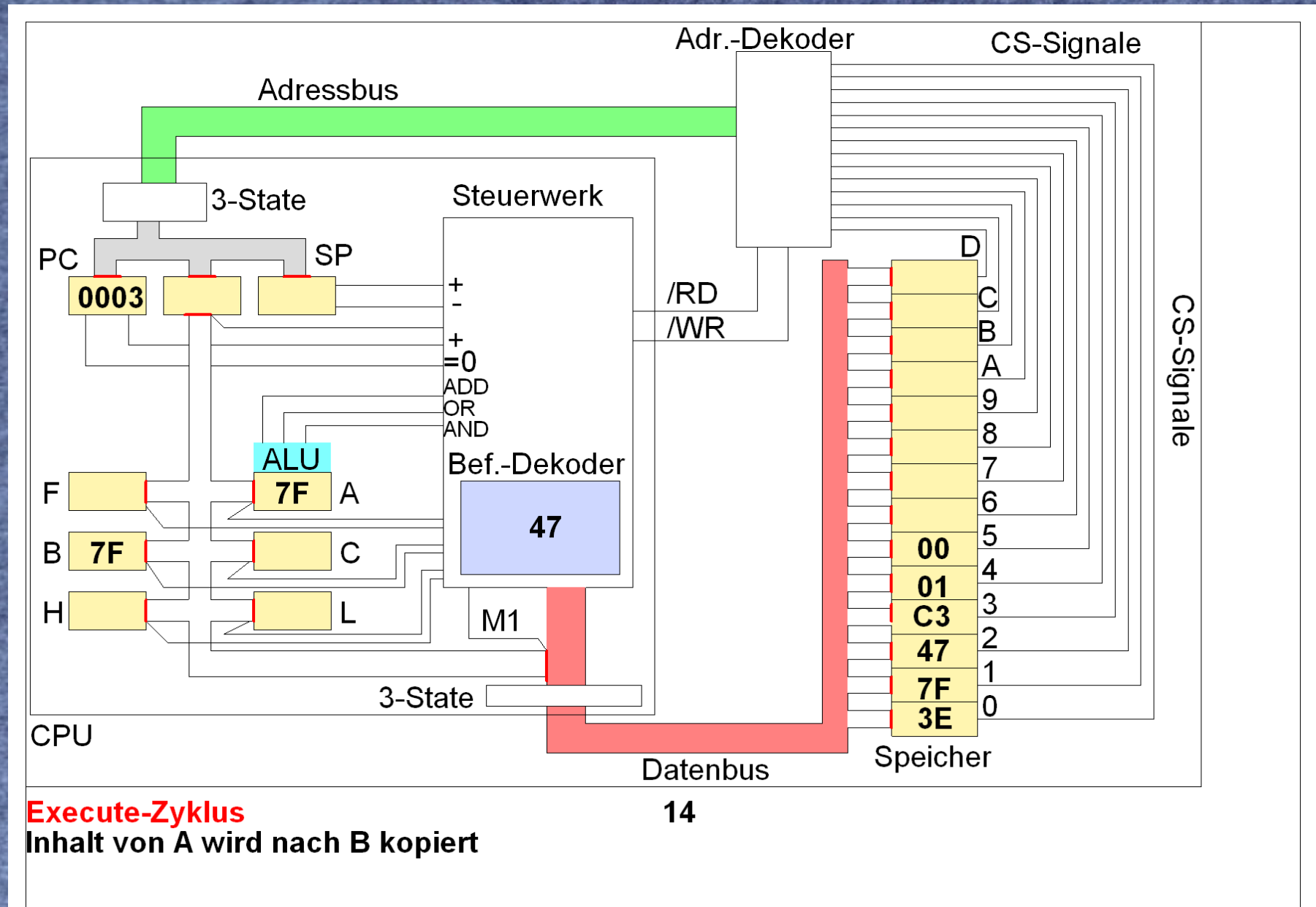


Dekode-Zyklus
Befehl dekodieren

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Jetzt ist auch der zweite Befehl abgearbeitet.

Die Auswirkungen waren:

- Programmzähler steht jetzt auf 0003
- Im Akku und im B-Register steht der Wert 7FH
- ab jetzt weiß der μ P nicht mehr, dass sein letzter Befehl LD B, A war, geschweige denn was der vorletzte Befehl war.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Es folgt die Bearbeitung des 3. Befehls.

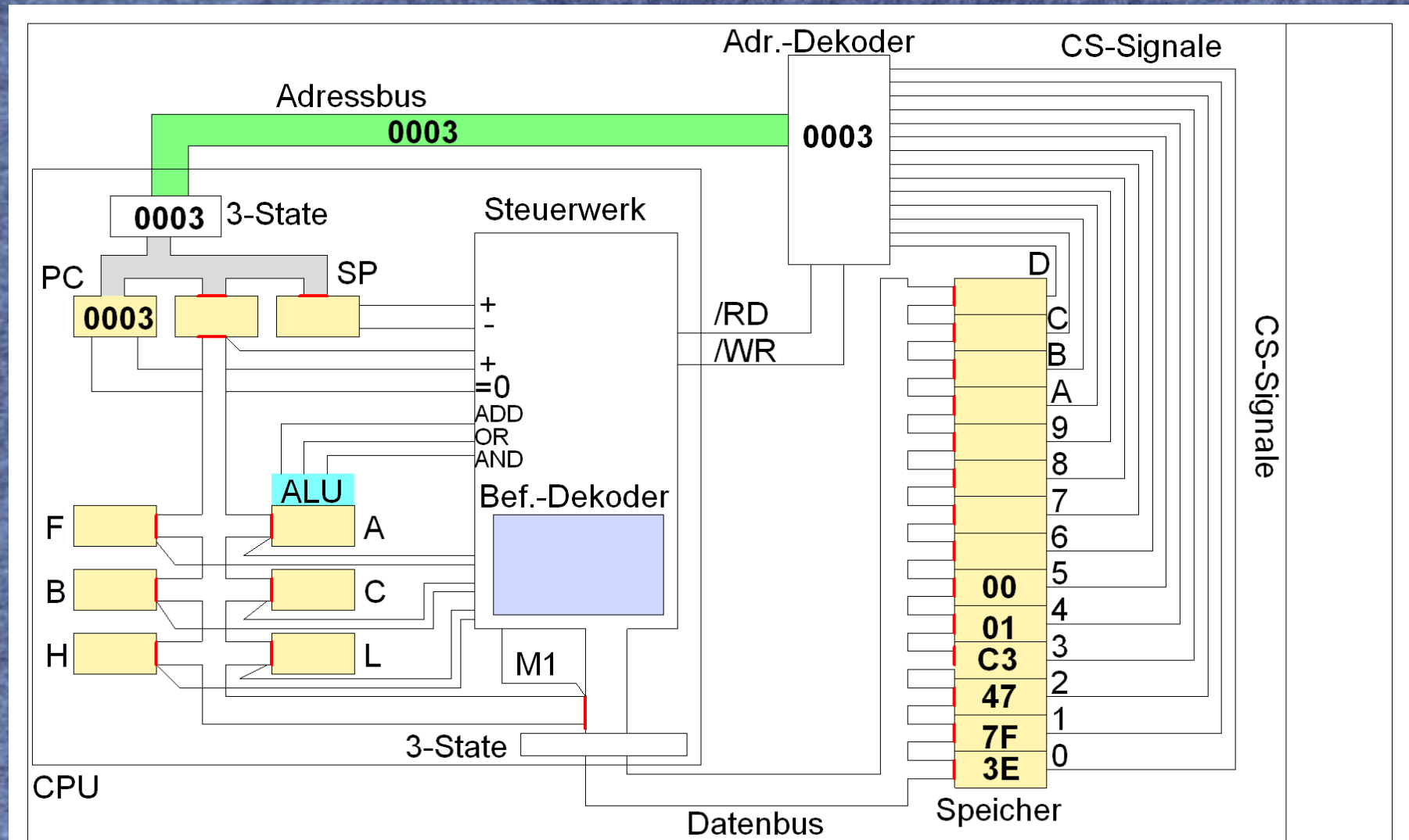
Dieser Befehl besteht aus dem Befehlscode und zusätzlich zwei weiteren Bytes.

JP 0001 führt einen Sprung an die Programm-
adresse 0001H aus.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

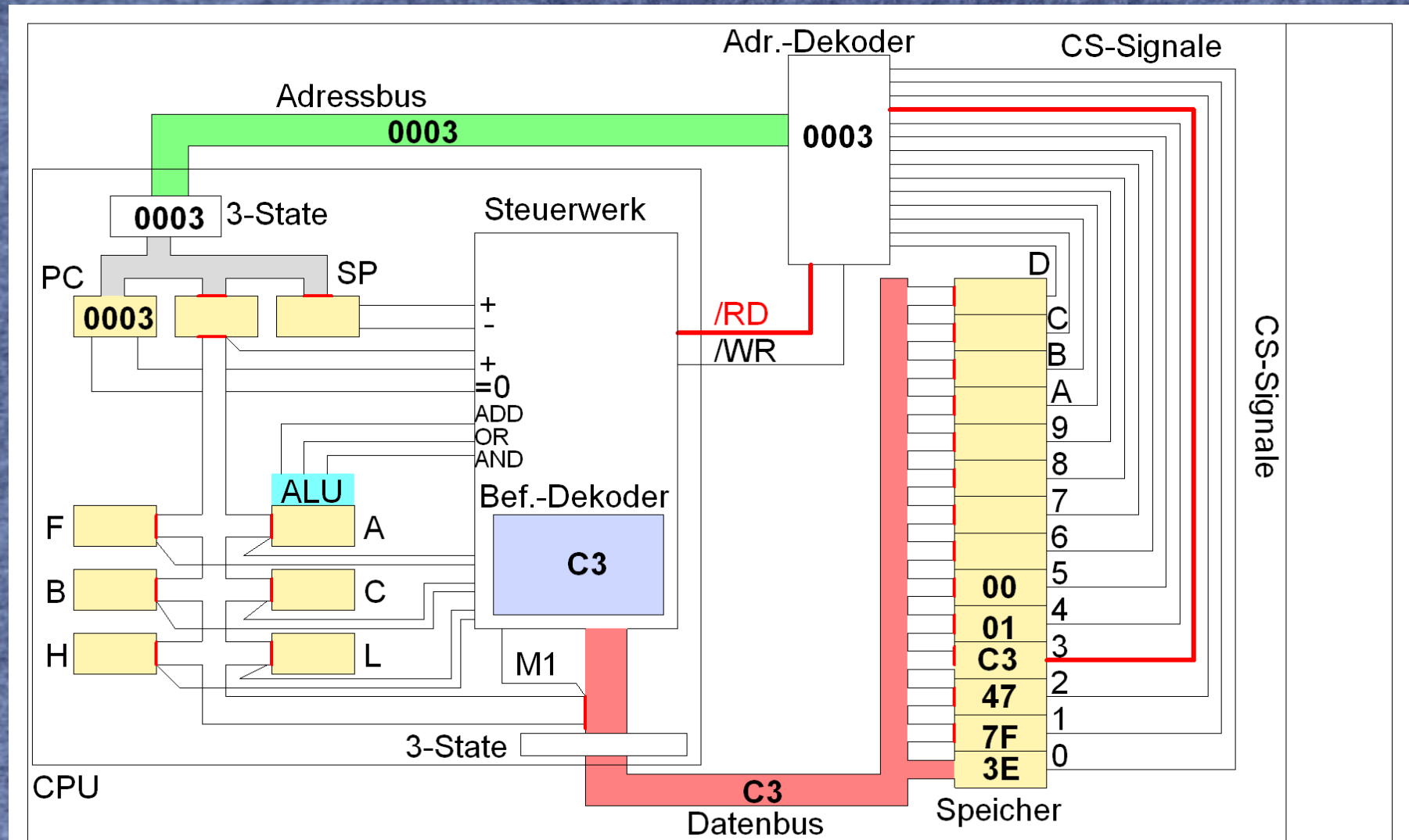
15

Schritt 1: Inhalt des PC auf den Adressbus legen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

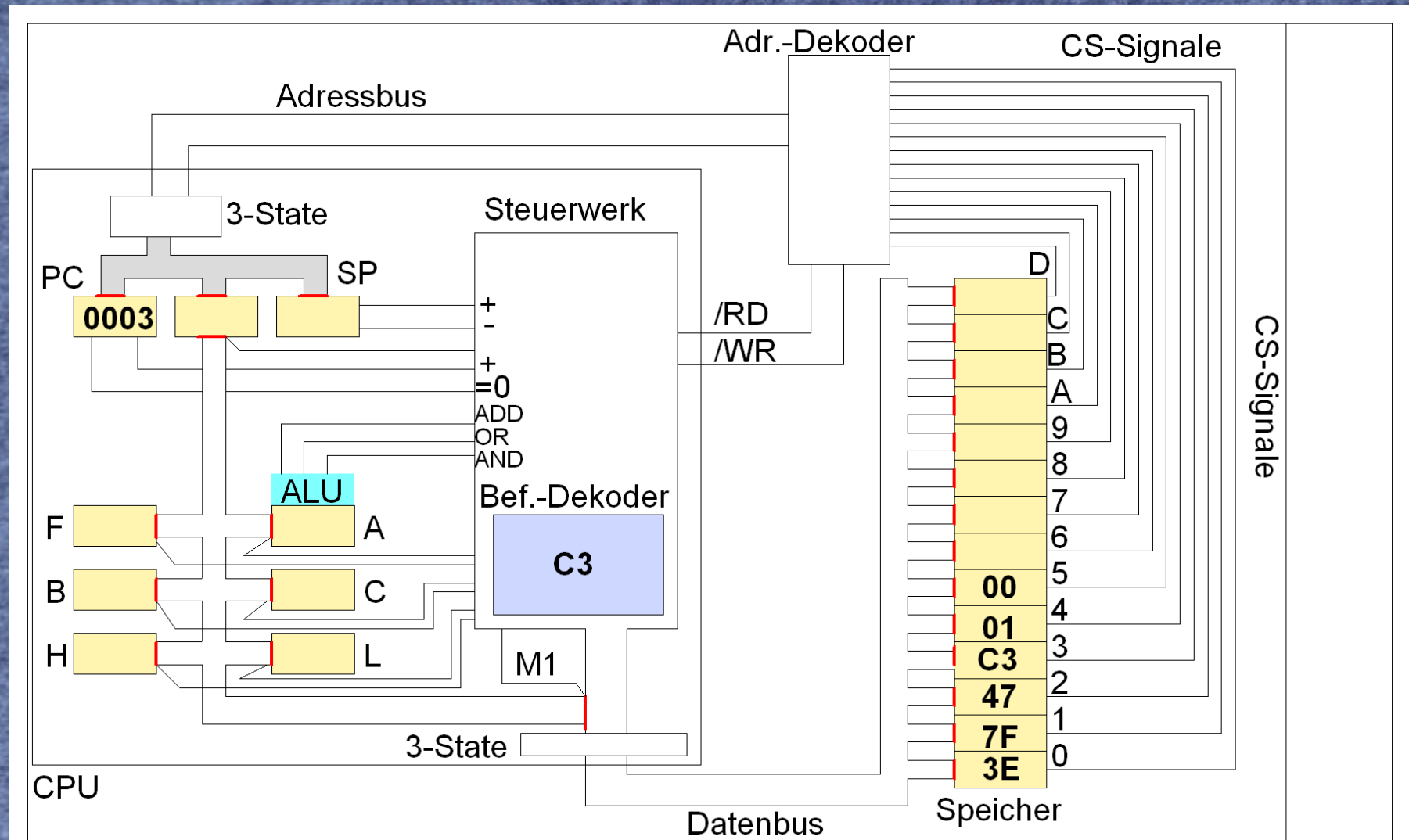
16

Schritt 2: Adr.-Dekoder gibt entsprechende Speicherzelle frei
deren Inhalt gelangt über den Datenbus in den Befehlsdekode

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

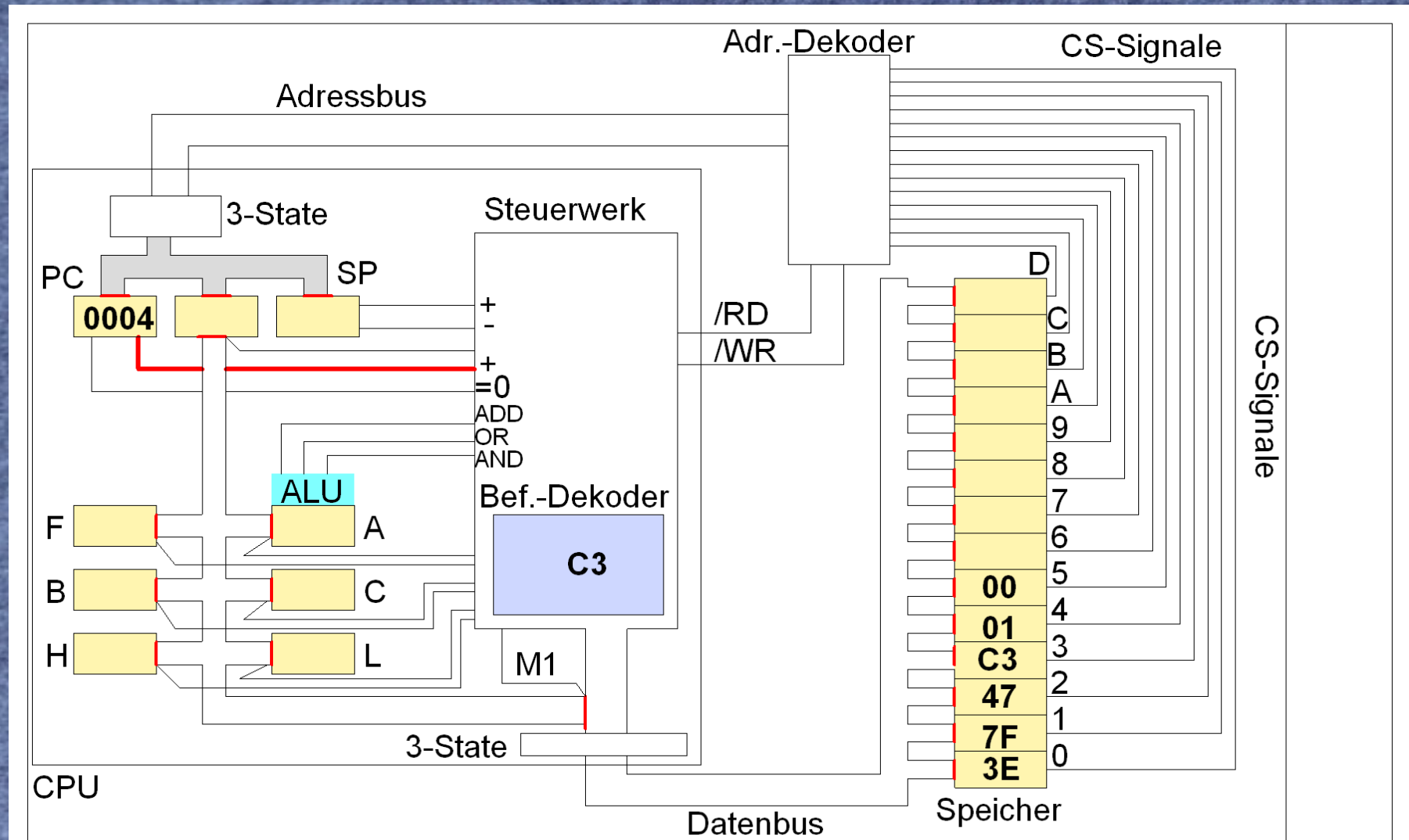
17

Schritt 3: Adress- und Datenbus wieder inaktiv

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



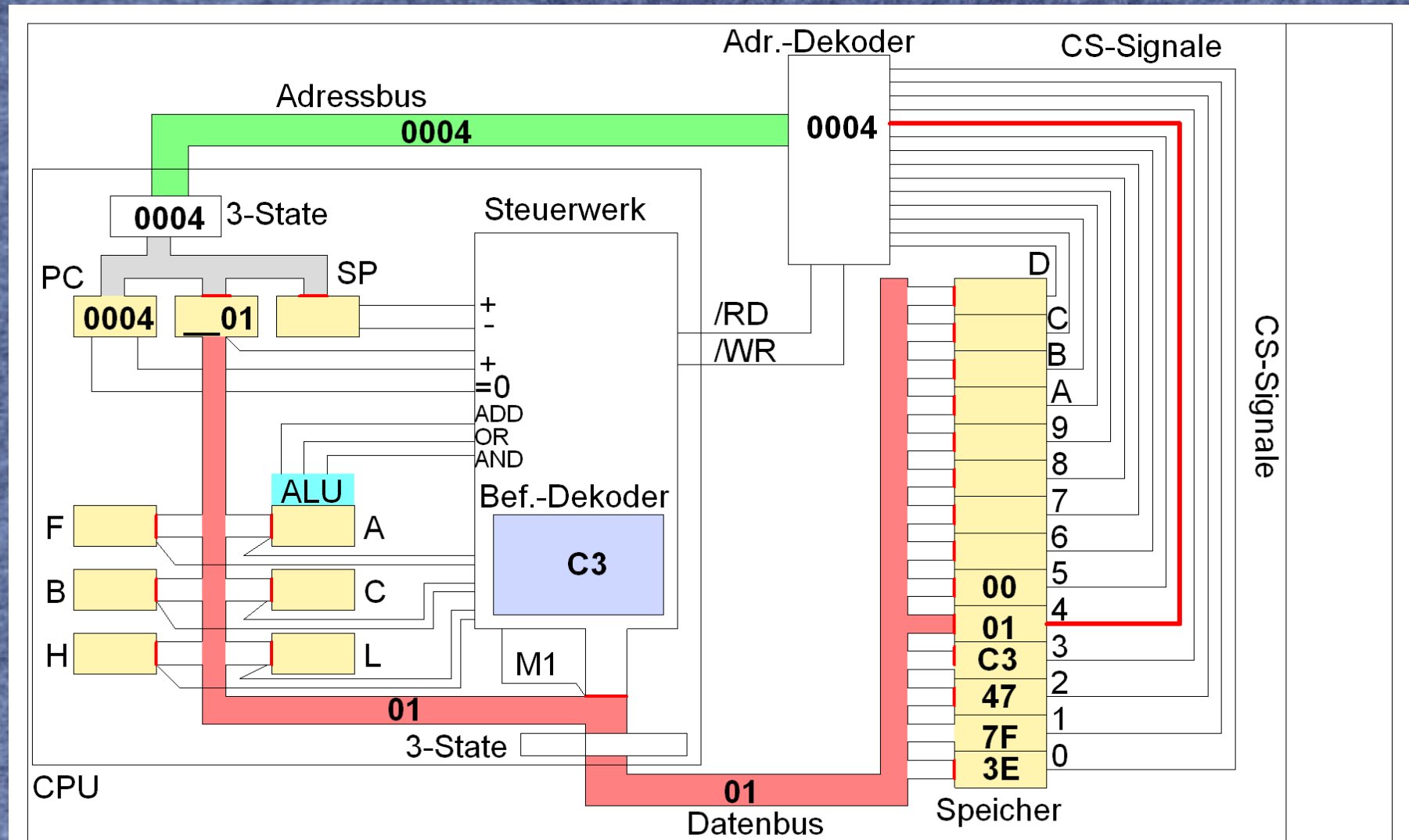
Fetch-Zyklus (M1)

Schritt 4: Programmzähler um 1 erhöhen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Dekode-Zyklus

19

Schritt 1: Steuerwerk erkennt, dass zwei Argumente eingelesen werden müssen deshalb wird erneut vom Speicher (Adresse 0004) gelesen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

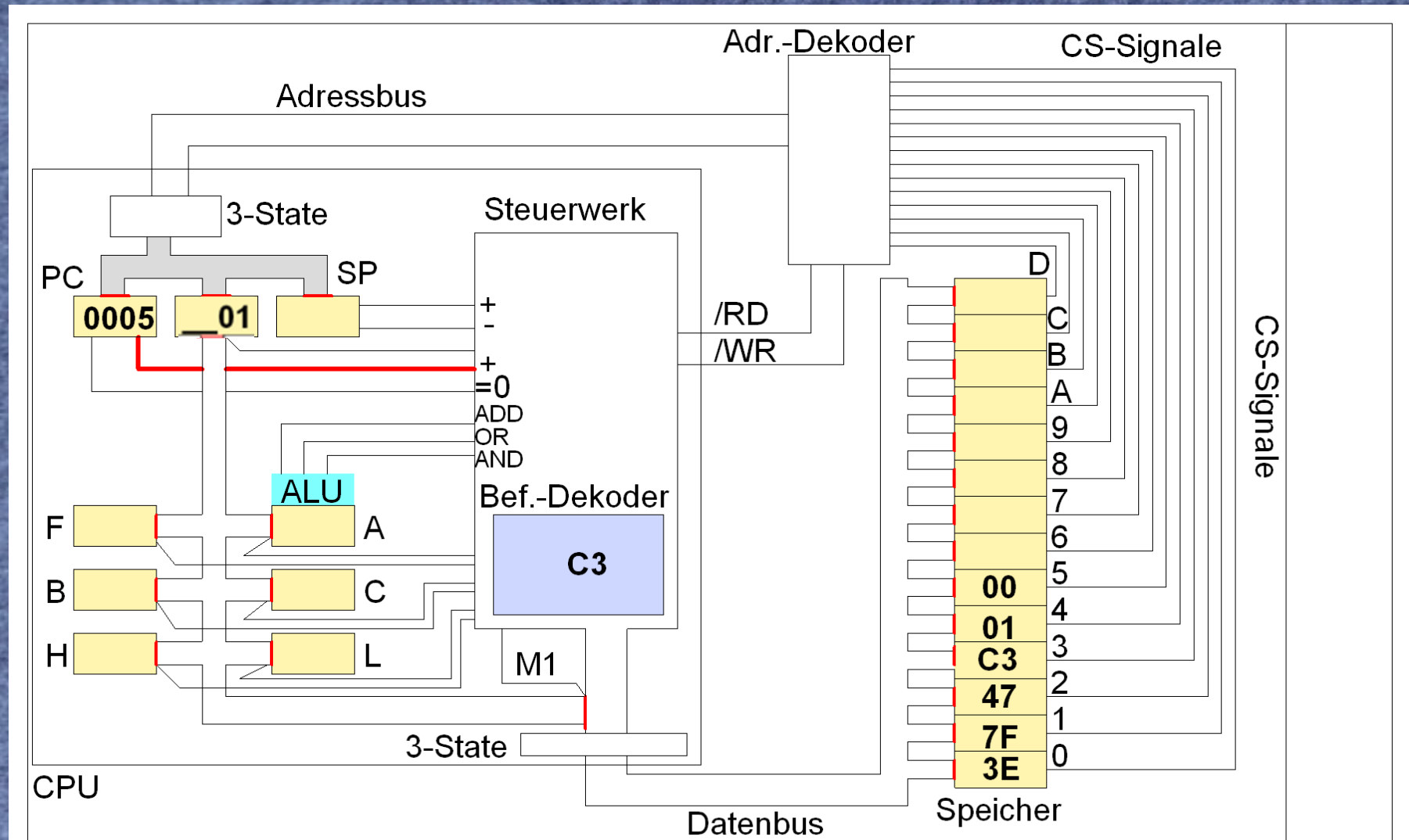
Da zwei Argumente eingelesen werden müssen, erfolgt der Speicherzugriff auch unter zwei Mal. Die Adresse steht aktuell im PC, weil die Argumente die Bytes nach dem Befehlscode sind.

Die einzelnen Argumente müssen deshalb zuerst in einen Zwischenspeicher geladen werden, damit der Programmzähler auch für den zweiten Speicherzugriff erhalten bleibt.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



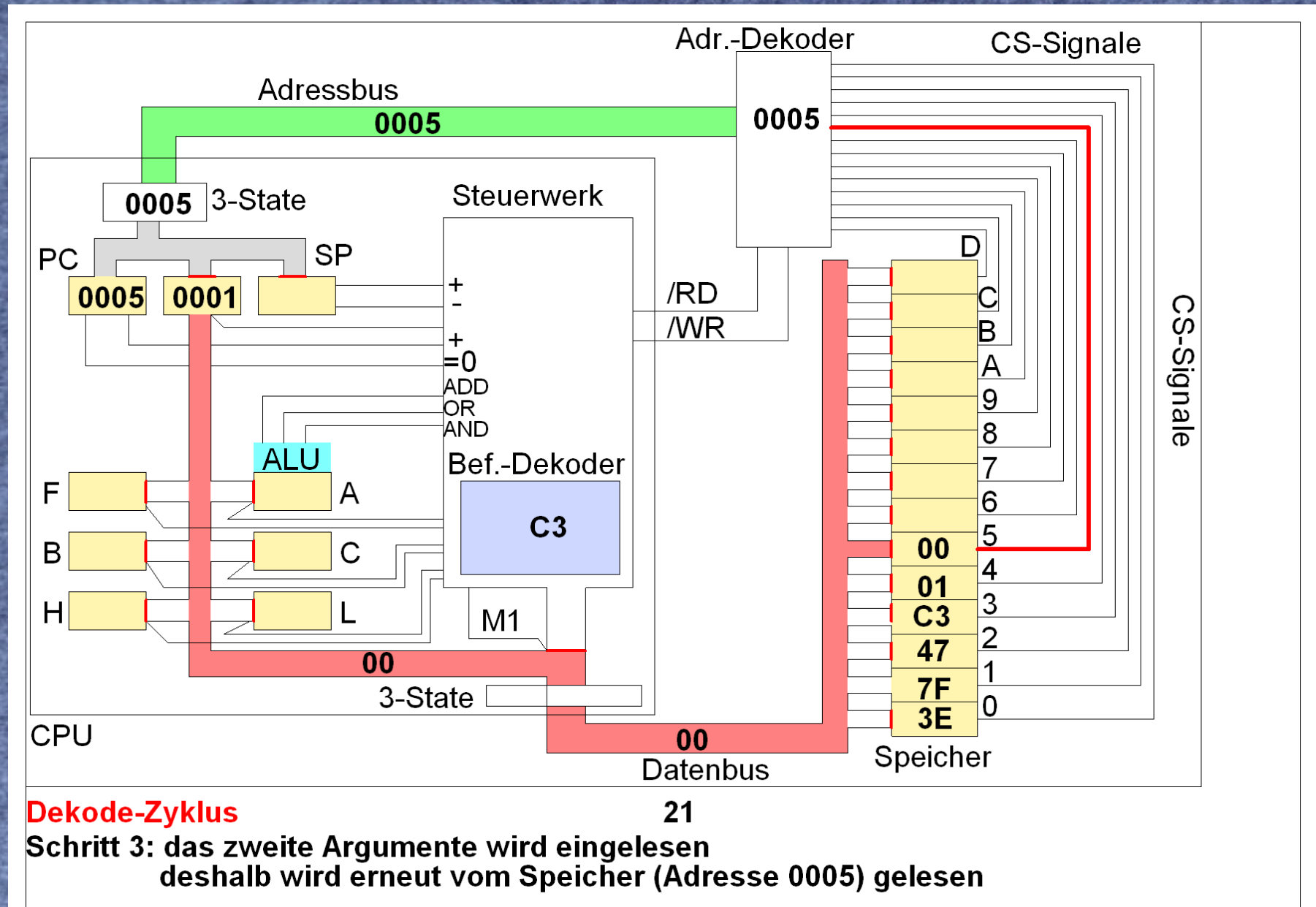
Dekode-Zyklus

Schritt 2: Programmzähler um 1 erhöhen

Rechnertechnik I

TINF DHBW Karlsruhe

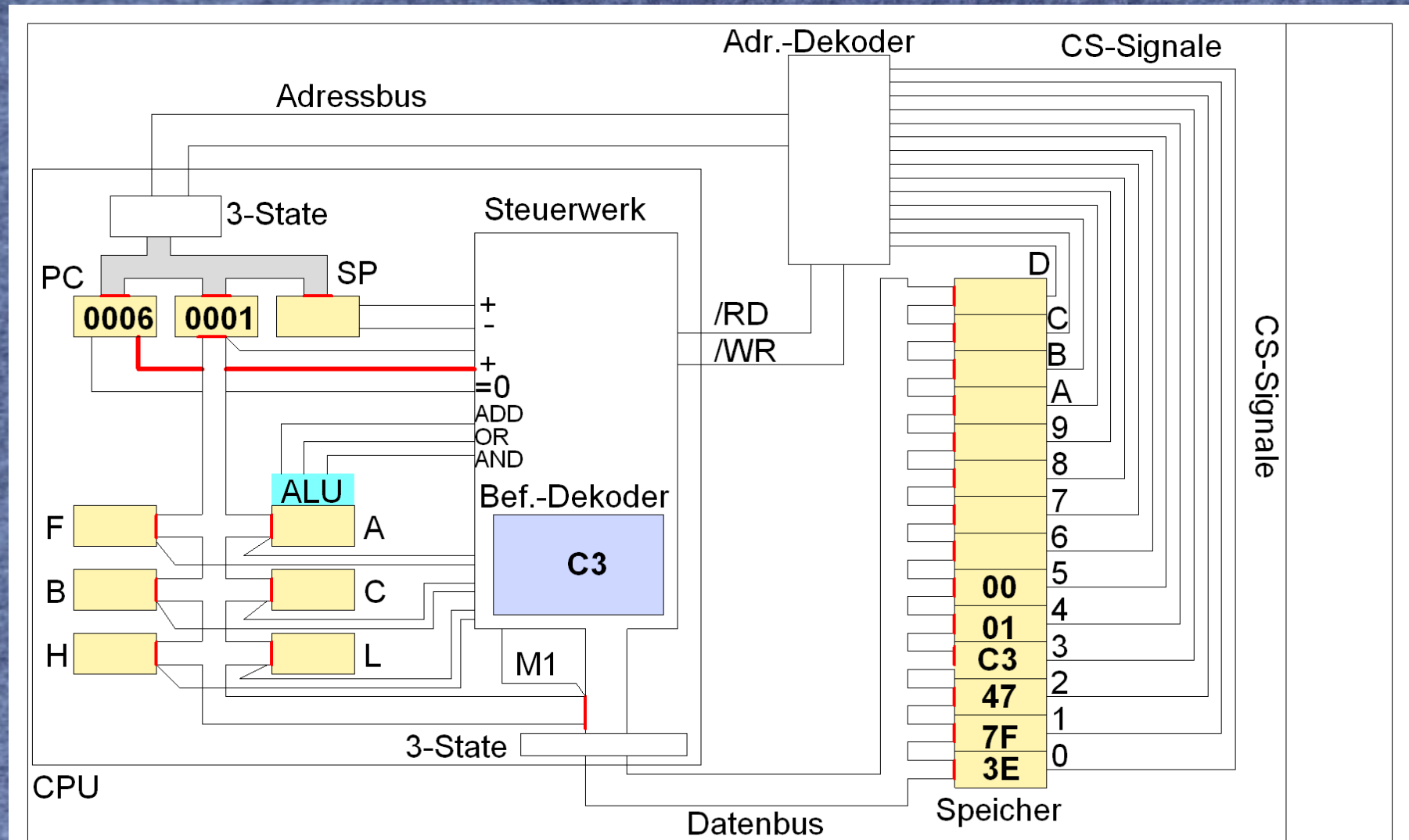
© Stefan Lehmann



Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



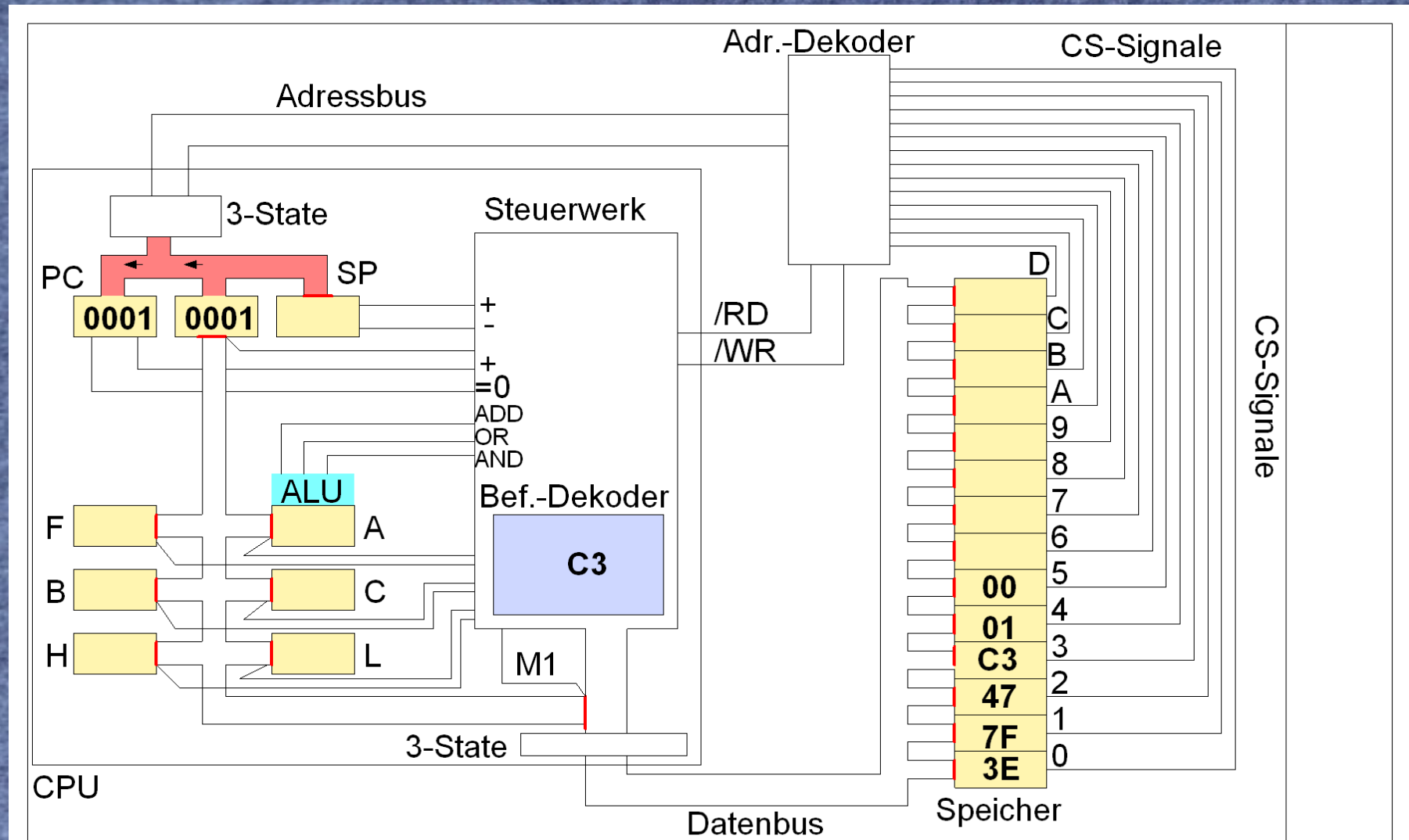
Dekode-Zyklus

Schritt 4: Programmzähler um 1 erhöhen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Execute-Zyklus

23

Inhalt des Zwischenspeichers in den Programmzähler kopieren

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Ein Sprungbefehl an eine absolute Adresse bedeutet somit nichts anderes, als dass der Programmzähler mit einer 16-Bit Adresse überschrieben wird.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Damit ist der dritte Befehl auch abgearbeitet.

Die Auswirkungen waren:

- Programmzähler steht jetzt auf 0001

-

ACHTUNG!

- Der Sprung auf Adresse 0001 bedeutet, dass als nächstes das Argument des 1. Befehls als neuer Befehl interpretiert wird. Das ist jedoch ein Fehler!

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Es folgt die Bearbeitung des 4. Befehls, der aber eigentlich das Argument (7FH) des 1. Befehls ist.

Wird dieses Argument als Befehl dekodiert, kann dies unvorhersehbare Auswirkungen haben. In diesem Fall bedeutet 7FH als Befehl

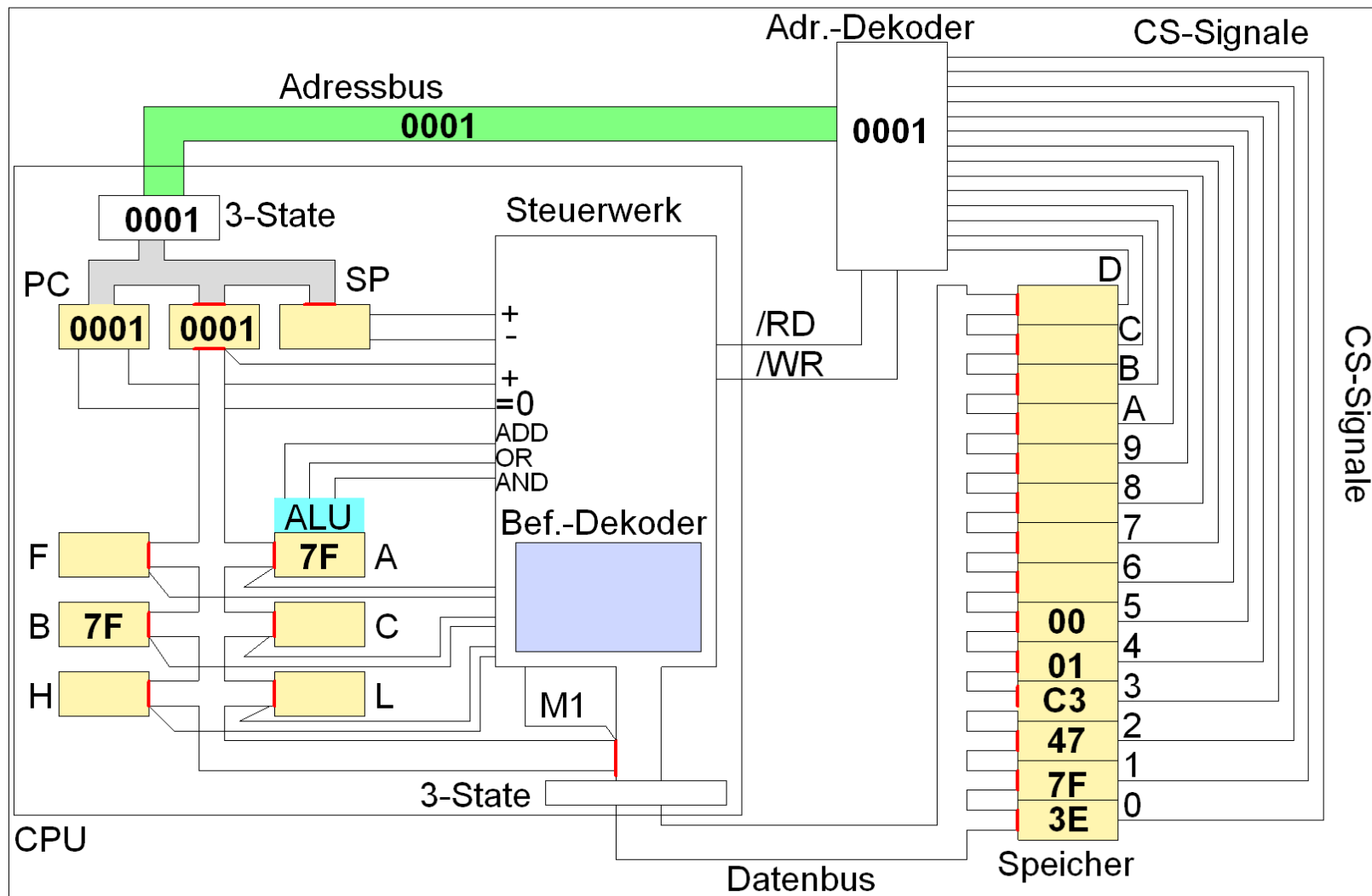
LD A,A

was in diesem Fall kein Problem darstellen würde.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

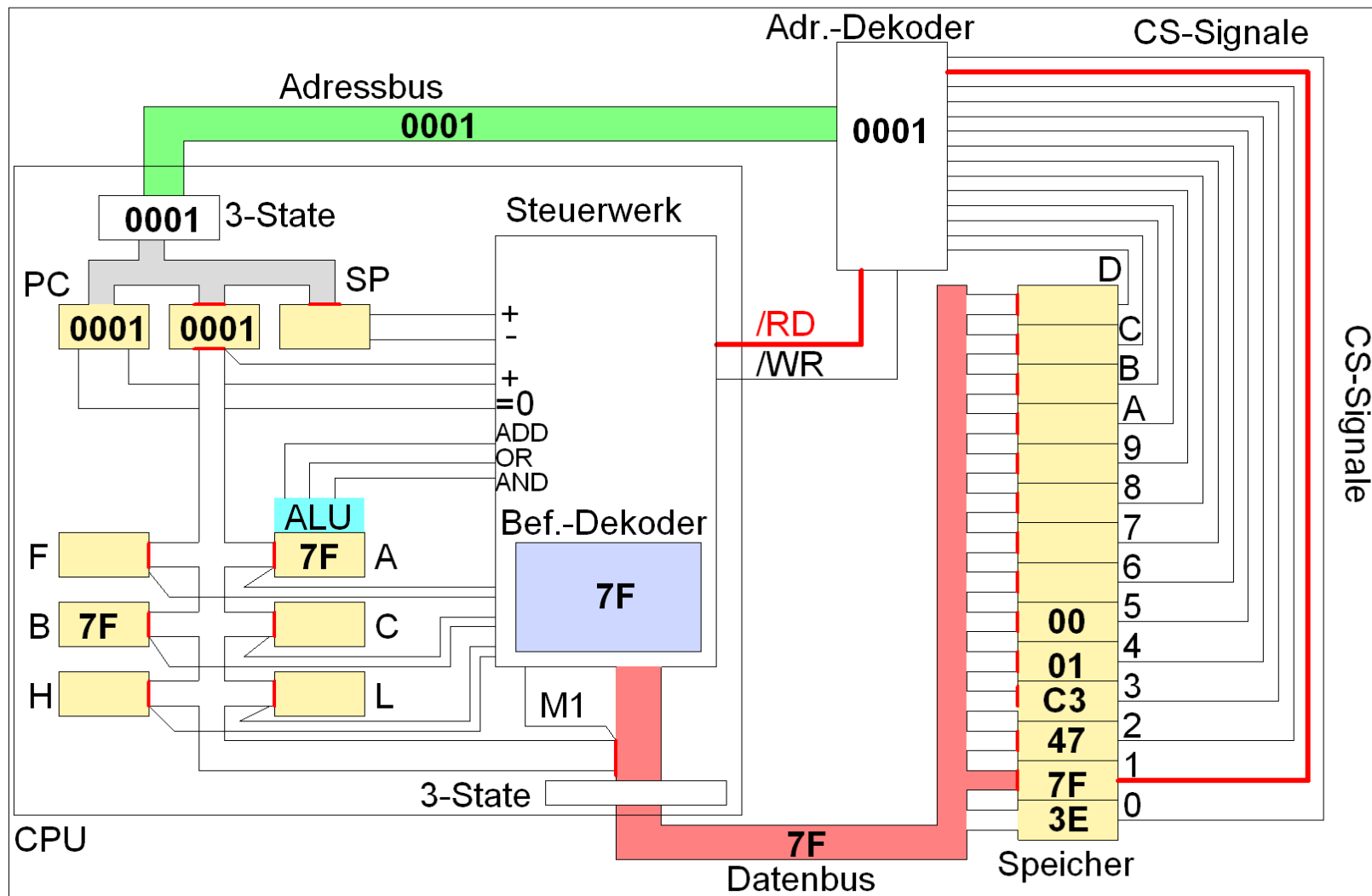
24

Schritt 1: Inhalt des PC auf den Adressbus legen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

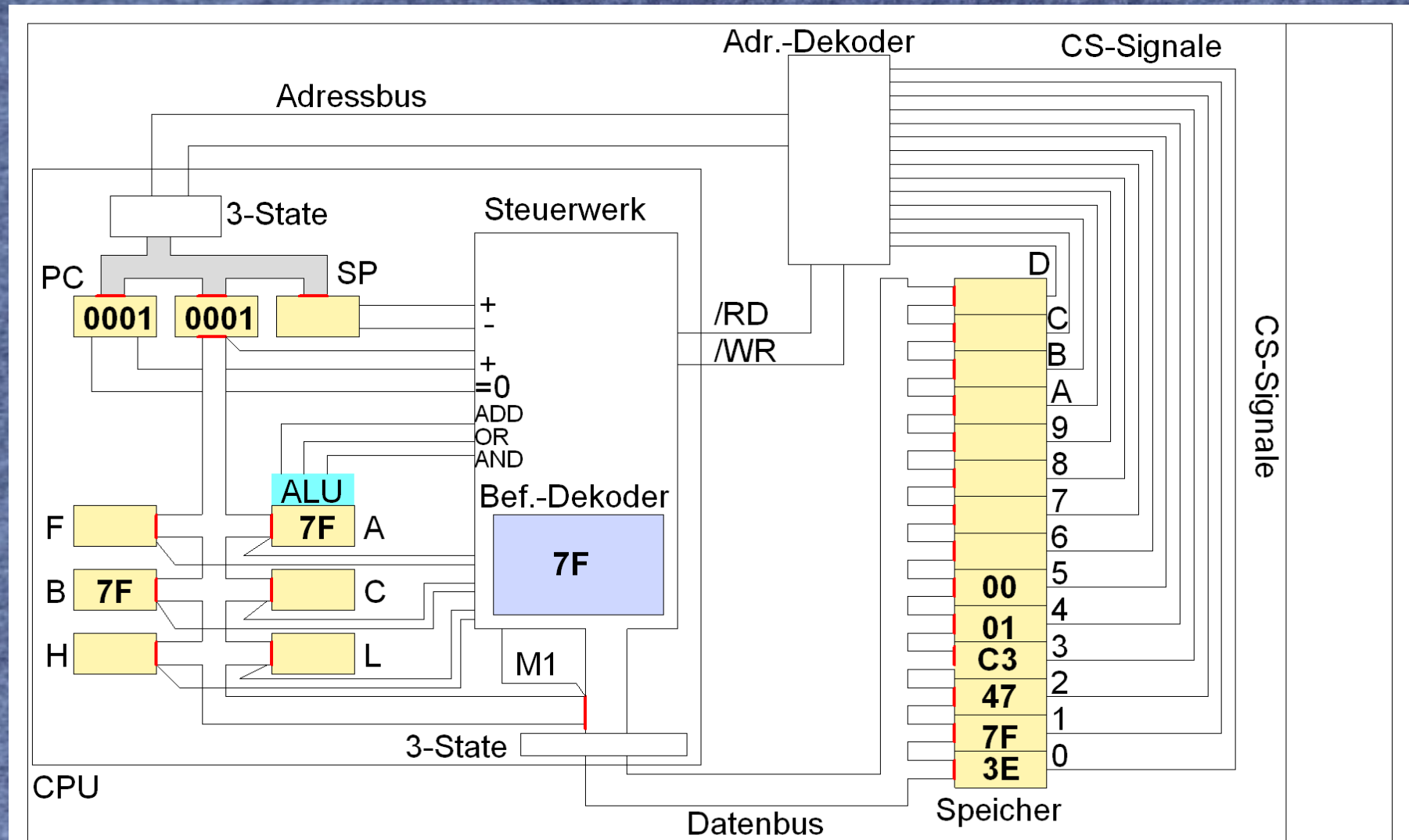
25

**Schritt 2: Adr.-Dekoder gibt entsprechende Speicherzelle frei
deren Inhalt gelangt über den Datenbus in den Befehlsdekoder**

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Fetch-Zyklus (M1)

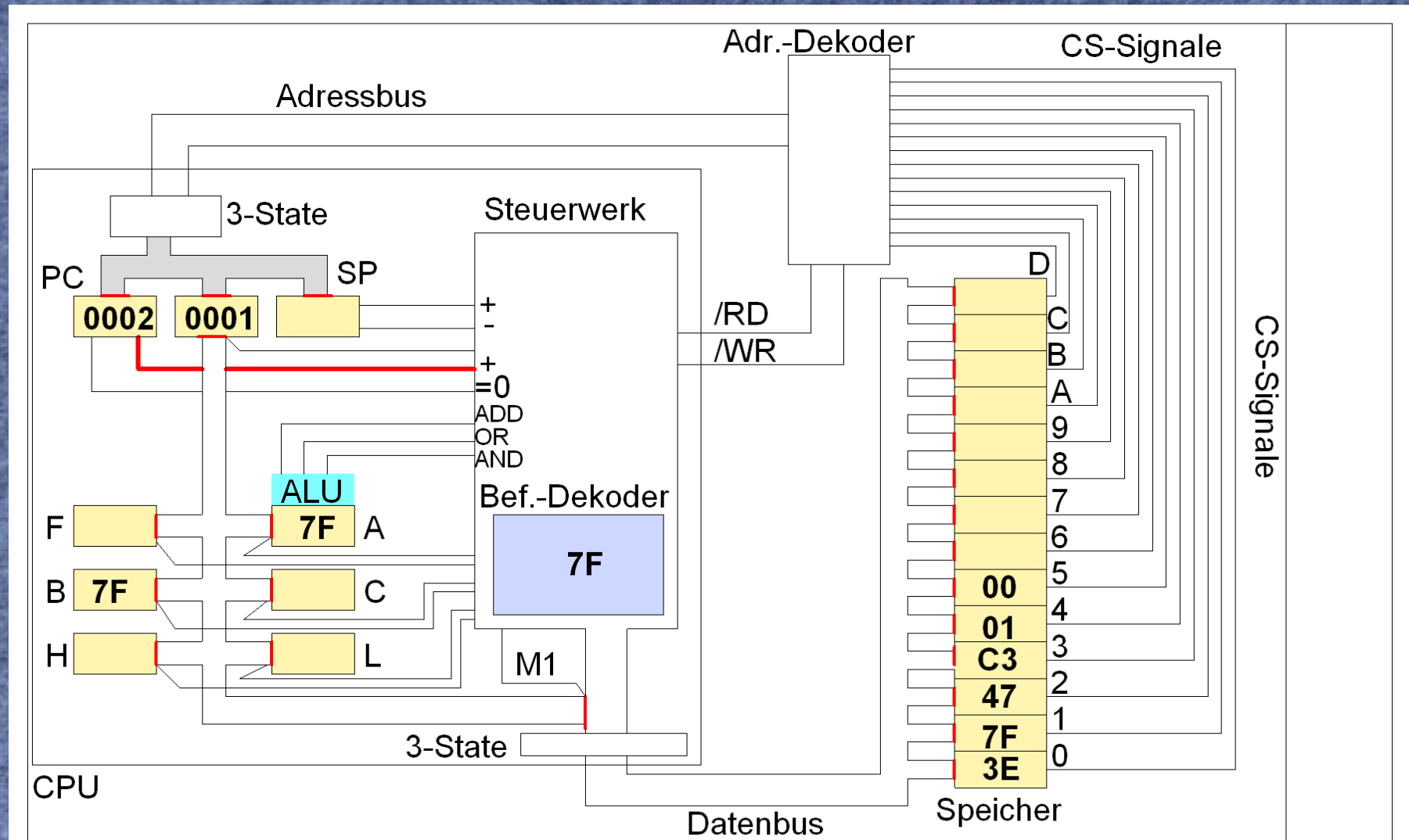
26

Schritt 3: Adress- und Datenbus wieder inaktiv

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



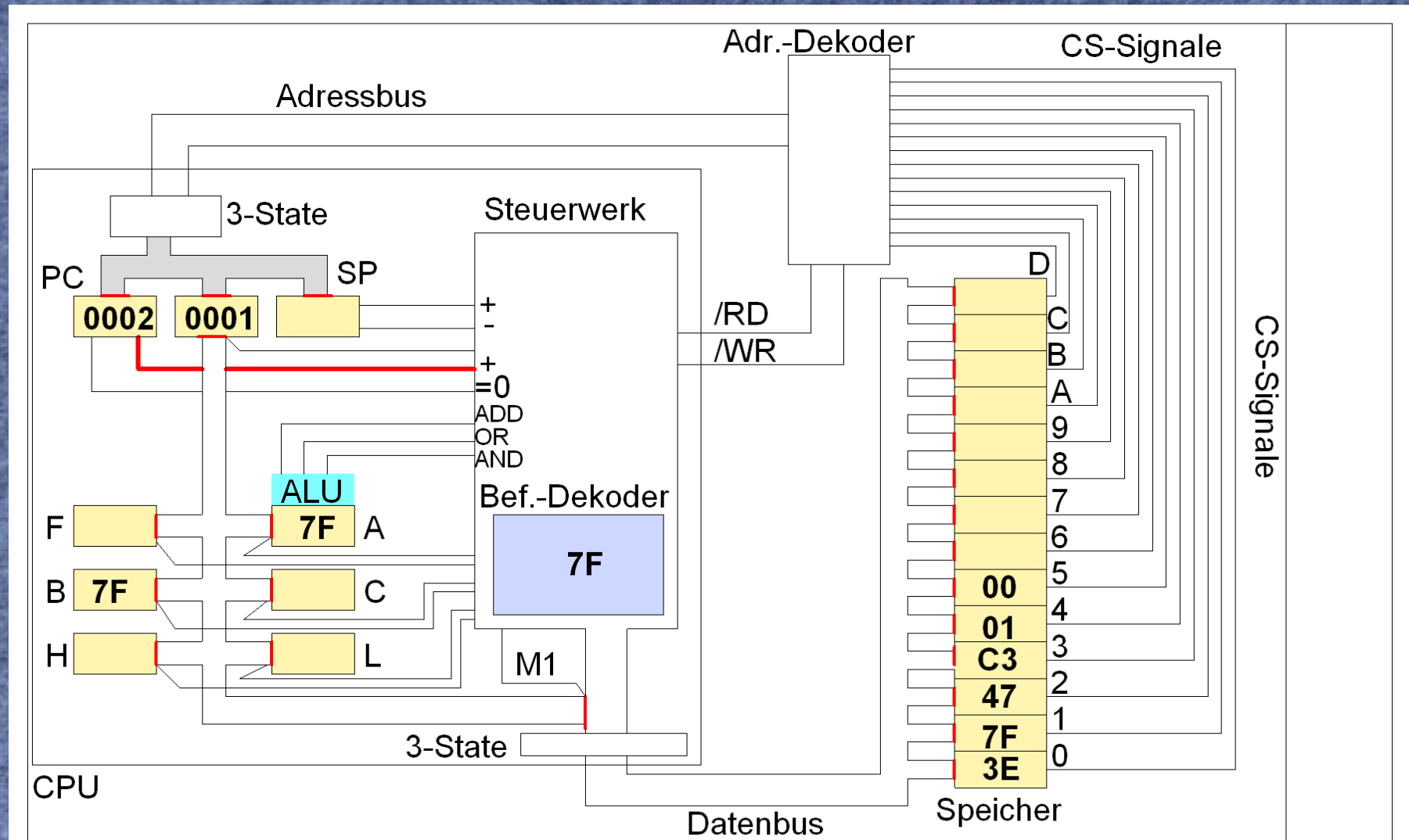
Fetch-Zyklus (M1)

Schritt 4: Programmzähler um 1 erhöhen

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann



Execute-Zyklus

Befehl ausführen (`LD A, A`)

trotz dieses fehlerhaften Sprungs, passiert an den Registerinhalten nichts weiter

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

An diesem Beispiel sieht man sehr gut, dass fehlerhafter Code nicht unbedingt zu einem Fehlverhalten des Programms führt.

Es zeigt aber auch, dass man solche Fehler nur sehr schwer finden kann, denn das Programm funktioniert ja wie gewünscht.

Wäre der Sprung an die Adresse 0000 aber richtig gewesen, kann es sein, dass das Programm so lange richtig funktioniert, solange in A und B der Wert 7FH steht.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

In der PDF „Neumann_Modell_P2.PDF“ wird das Programm nur an der Programmadresse geändert.

Statt des Arguments 7FH wird das Argument 3CH in den Akku geladen.

3CH als Befehl bedeutet INC A

- Schauen Sie sich dort das entsprechende Verhalten der CPU an.

Rechnertechnik I

TINF DHBW Karlsruhe

© Stefan Lehmann

Anhang:

Danke an die aufmerksamen Studenten, die den einen oder anderen Fehler noch entdeckt haben.

Herr Schinka, Herr Buchmann