Mikrocode: Steuersprache für Automaten zum Rechnen und Testen

Dr. Martin Perner, 0/1-SimWare, München

- Teil I: Einführung in den Mikrocode
  - Motivation
  - eLearning-Tool: Mikrocodesimulator „MikroSim“
  - MikroSim als virtuelle Applikation
  - Lernziele und Lehrpotenzial von MikroSim
- Teil II: Applikationsrelevanz
  - Evolution der Rechenautomaten
  - Bedeutung des Mikrocodes: Gestern - und heute“?“
- Teil III: Mikrocode zum Testen von Schaltkreisen
  - Einsatz von Testautomaten (BIST) in der Halbleiterfertigung
  - Mikrosteuercode für BIST-Automaten

Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Motivation: Bedeutung des Mikrocodes

- Programmierung als Schaltersteuerung auf niedrigstem Hardware-Niveau
- Ermöglicht flexible Anpassung von Automatensteuerungen
- Änderung und Optimierung von Steuerabläufen ohne Redesign möglich
- Kompromiss zwischen schneller und fixierter Hardware-Implementierung und langsamerer aber flexibler Steuer-Codeabarbeitung
- Mikrocodesimulator MikroSim vermittelt ein Gefühl dafür …

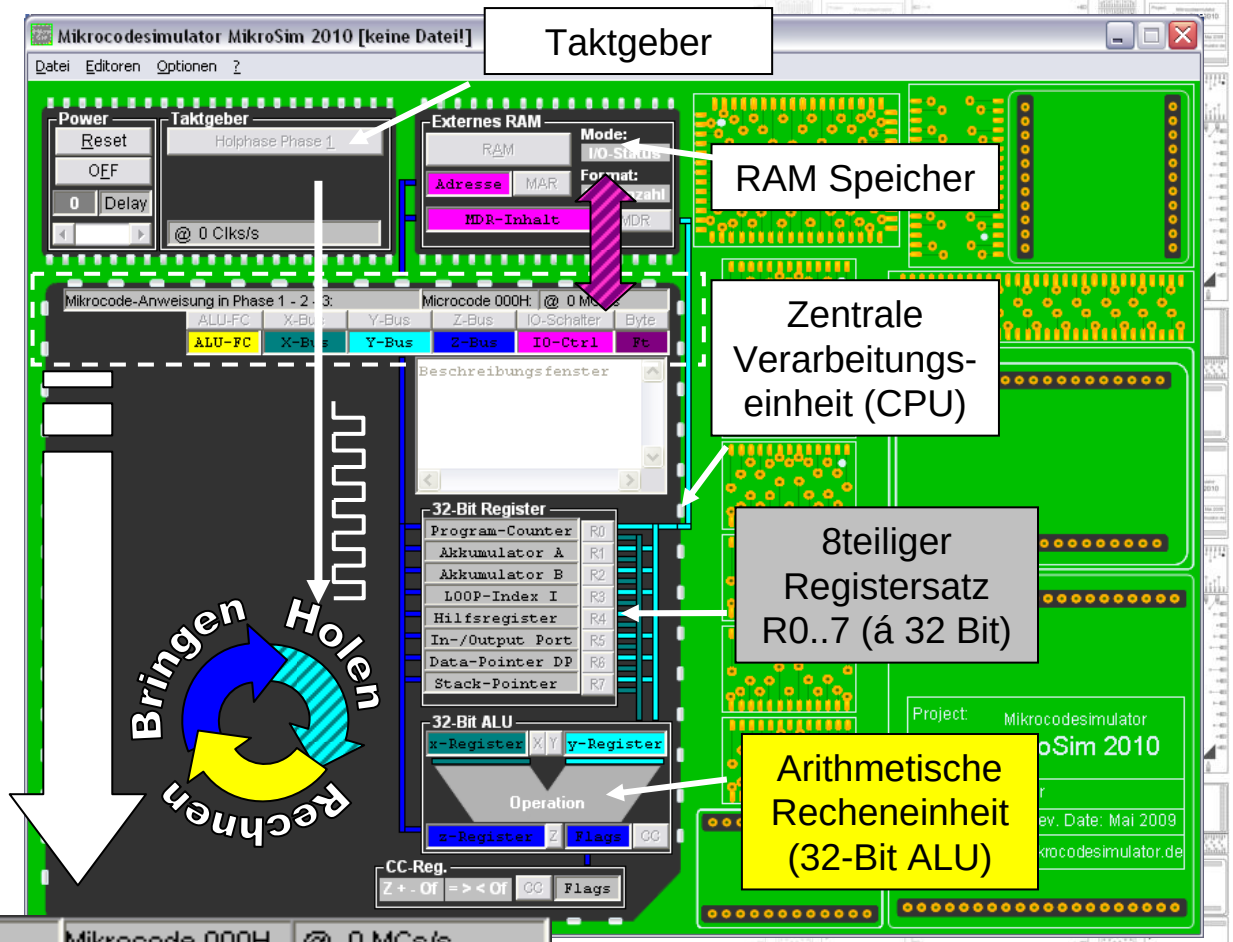
Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Mikrocode für einen virtuellen Rechenautomat

Der virtuelle Rechenautomat
„MikroSim“

… visualisiert eine taktweise und phasenweise Abarbeitung von Schaltvorgängen …

… beschrieben durch seine Steuersprache: den „Mikrocode“.



Beispiel:
„INC R0“ als 49Bit-Mikrocode

Mikrocode-Anweisung in Phase 1: | Mikrocode 000H | @ 0 MCs/s

Md	MCNext	ALU-FC	X-Bus	Y-Bus	Z-Bus	IO-Schalter	Byte
00	000000	0000110	00000000	00000001	00000001	00000000	00

Rechnen mit FC(110b): Z=Y+1

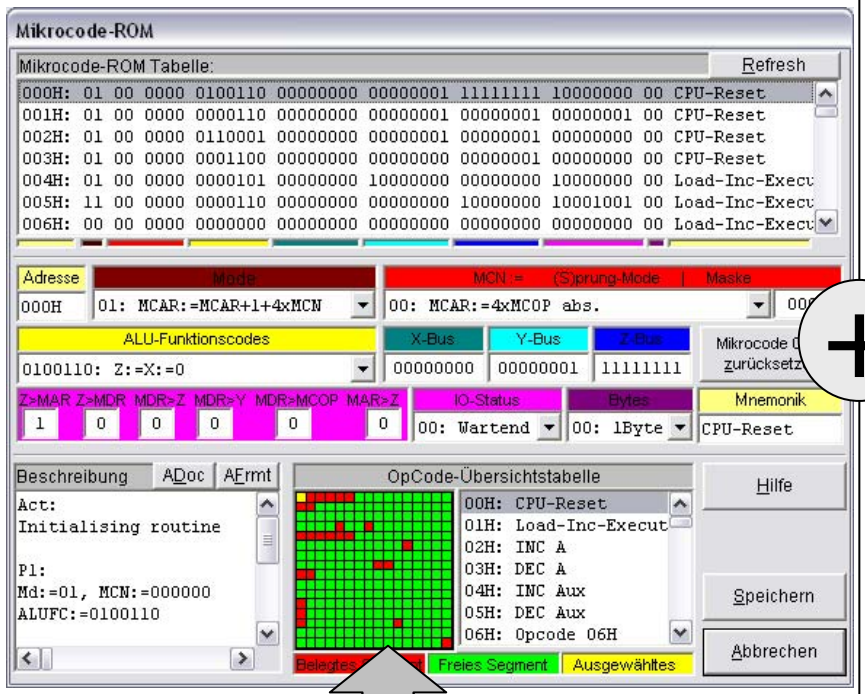
Holen: Y=R0

Bringen: R0=Z

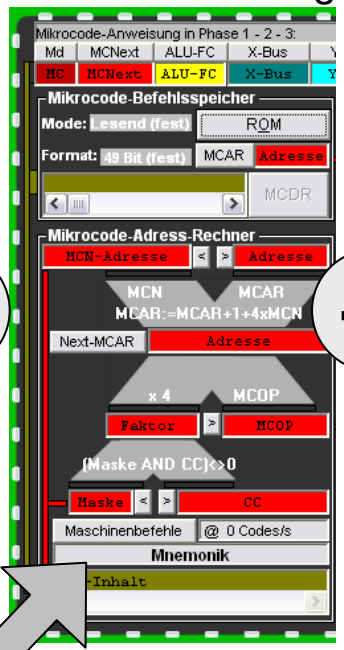
Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Von der Mikrocode-Liste zum Maschinenprogramm

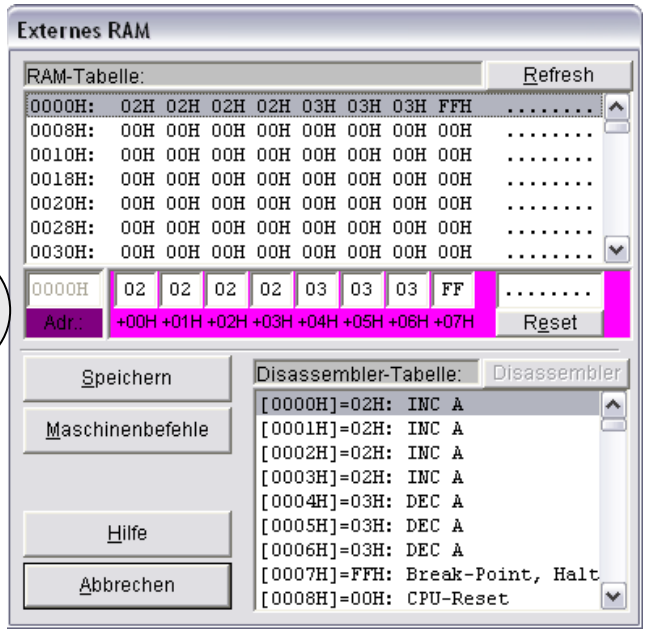
Mikrocode-Liste: 1024 x 49 Bit



Mikrocode-Adresssteuerung



Ablaufplan-Speicher (RAM)

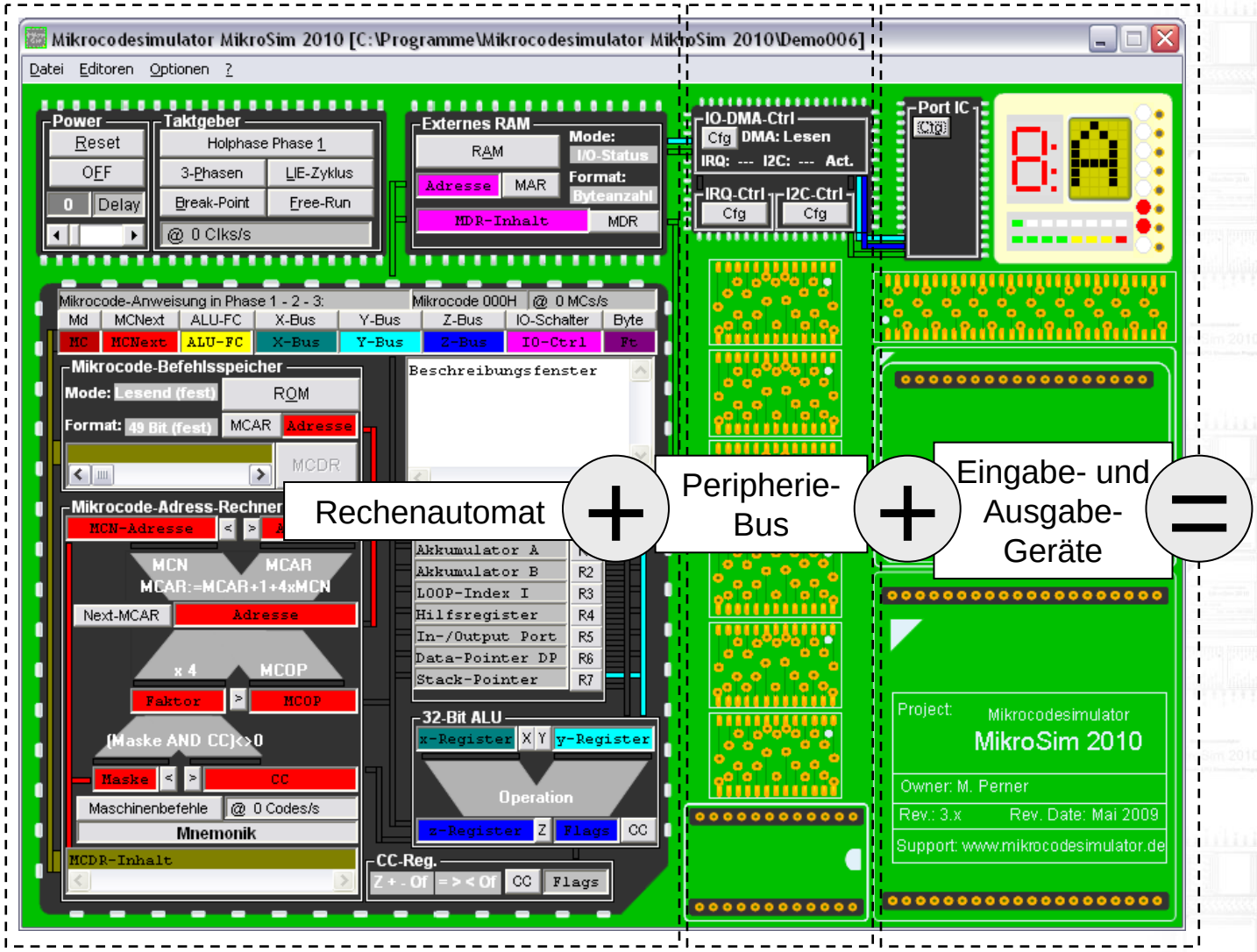


Mikrocode-Strukturierung und -Steuerung:

- Mikrocode-Segmente zu mit je 4Mikrocodes ergeben 256 „Unterprogramme“ (OpCodes)
- OpCode-Ablaufplan im „Externen RAM“ hinterlegt
- Interpreterprogramm in Mikrocode-Sprache für Abarbeitung des Ablaufplans (Maschinenprogramm)

Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Die virtuelle Applikation „MikroSim“



Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Lernziele und Lehrpotenzial von MikroSim

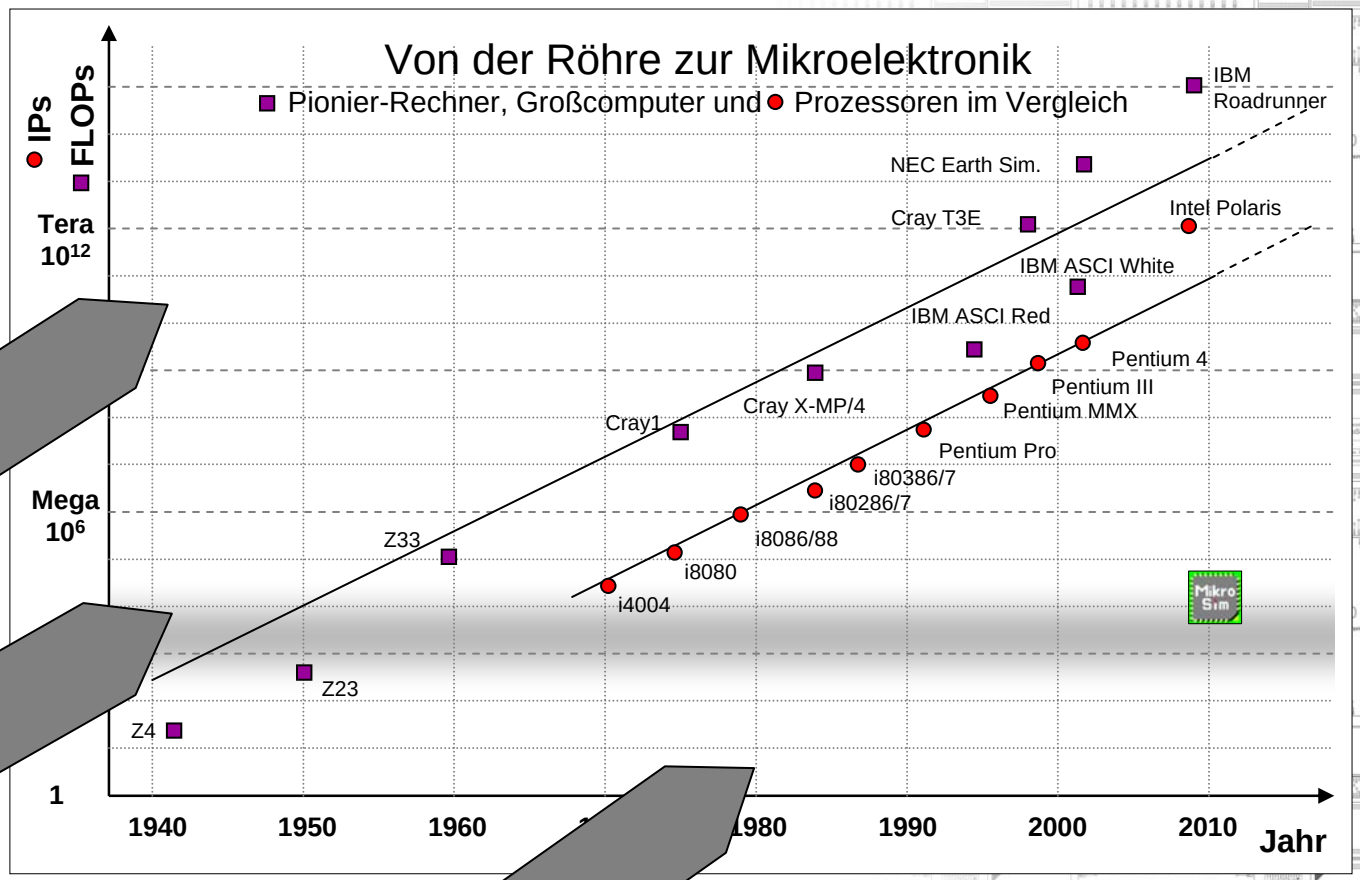
- Gespür für Effektivität der Codeabarbeitung bzgl. Platzbedarf und optimierte Organisation
- Implementierung von Maschinencodes als Mikrocode-Unterprogramme
- Interpretation von Registerinhalten als Schalterstellung, Binär-, Hexadezimal-, Gleitkomma- oder Dezimal-Zahl
- Gefühl für die Mächtigkeit von ALU-Rechenbefehlen für Schiebe-, Bit-, Logik-, Ganzzahl- oder Gleitkomma-Berechnung
- Bedeutung von mathematischen Coprozessoren
- Strukturierung des RAMs für Programm und Daten (Neumann – Harvard)
- Interaktionsmöglichkeiten mit Hardware (DMA, Interrupt, I2C-Bus, …)
- Messen der Abarbeitungsgeschwindigkeit in MIPS und FLOPS verglichen zum reellen PC
- Einordnung der Entwicklung der Rechenautomaten auf der eigenen Plattform verglichen mit MikroSim

Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Evolution der Rechenautomaten

Parallelisierung:
- Rechenoperationsbreite: 4-, 8-, 16-, 32- und 64-Bit
- Aufgabenverteilung mehrerer Rechenpfade einer CPU
- Aufgabenteilung zwischen den Prozessoren
- Aufgabenverarbeitung in mehreren Kernen eines Prozessors

Technische Entwicklung:
- Schalter-Miniaturisierung: Relais – Röhre –Transistor - IC
- Schaltgeschwindigkeit (Takt)
- Strukturverkleinerung (Shrink)
- Bauteilintegration
- …



Automatensteuerung:
- Compiler-Bau: Sprachen für Automaten (Plankalkül Hochsprache)
- Speicherminiaturisierung (Lochstreifen - Ringkern – DRAM)
- Festplatten (Trommelspeicher - Festplatte)
- Cache-Steuerungsmechanismen
- …

Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Bedeutung des Mikrocodes: Gestern - und heute „?“

- Gestern:
  - Lochstreifencodes steuern direkt (‘ 50er und ‘60er)
  - Mikrocode in „CISC“-Rechnern als „BIOS“ für CPUs der ‘70er und ‘80er
  - „RISC“ auf dem Vormarsch in den ‘90ern: Das Interesse am Mikrocode flaut etwas ab
  - Pentium FDIV-Bug bewirkt Revival der flexiblen Mikrocodeprogrammierbarkeit für Bug-Fixes

- Heute:
  - Optimierung der Prozessor-Performance über BIOS
  - Wichtig bei Firmware-Update und Bug-Fix
  - Built-In Self-Test (BIST): Mikrocode-gesteuerte Automaten in Produktion und Test vom Wafer bis zur Applikation

Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Testschritte in der Halbleiterbauelement-Produktion

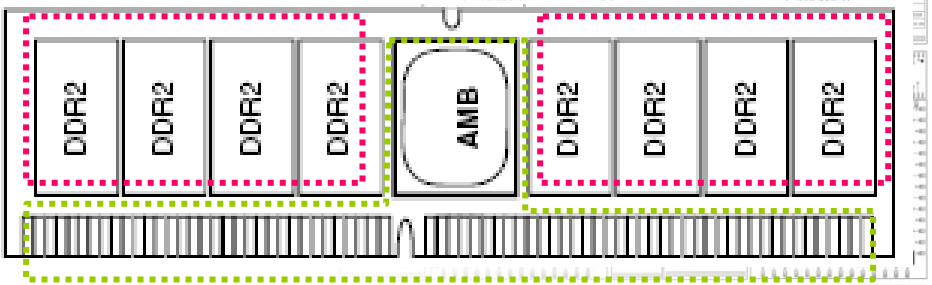
- Testschritte:
  - Wafer-Test
  - Burn-In-Stresstest
  - Baustein-Test
  - Modul-Test
  - Applikations-Test

- Gründe zur Einführung von Test-Automaten auf dem Silizium:
  - Einsparung von Kontakt-Pins zur Erhöhung der Testparallelisierung (Auslastung, Kostensenkung)
  - Nachträgliche Konfigurations- und Reparaturmöglichkeiten (e-Fuse, Ausbeutesteigerung)
  - Verfahrenstechnische Herstellungsgründe und Verwendungszweck (Baustein-Test auf Wafer: Known-Good-Die, Modultest: FBDIMM)

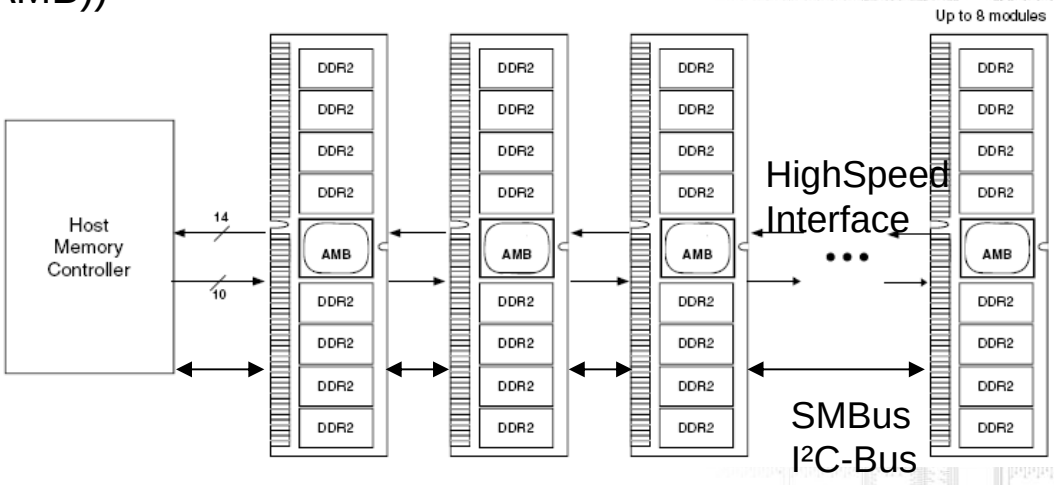
Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

FBDIMM-Speicher-Test mit „Mikrocode“

- Ein Fully Buffered DIMM bestehend aus:
  - DDR2 DRAM **Speicherkomponeten**
  - Module-Platine
  - **Zugriffscontroller-Interface** (Advanced Memory Buffer (AMB))



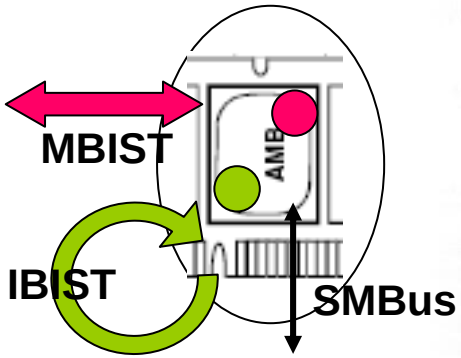
- Applikationsbetrieb:
  - bis zu 8 Module Kanal
  - bis zu 192 Gbyte/System



- Minimale Testbedingung:
  - Spannung (1.8V und 3.3V)
  - Referenztakt (100-400 MHz)
  - Serielles Interface (SMBus/I²C/JTAG)
  - …und Built-In-Self-Test-Automaten (BIST)

  für Speicher (MBIST)

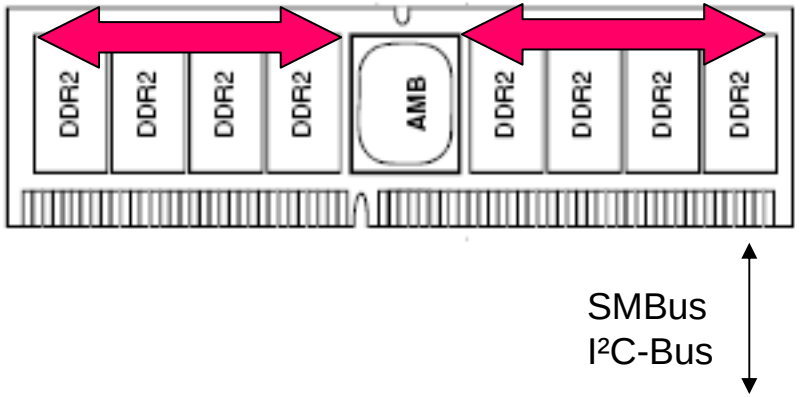
  und Interface (IBIST)



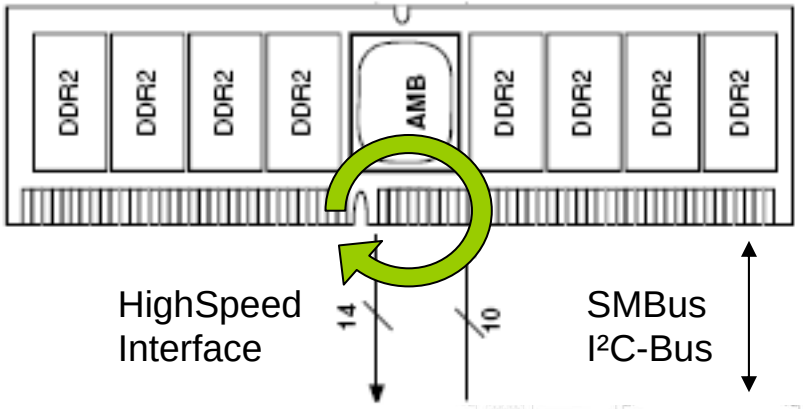
Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

AMB BIST-Automaten für FBDIMM

- Memory BIST (MBIST) beherrscht:
  - AMB Konfiguration
  - DRAM Einzelzugriffe
  - DRAM Konfiguration
  - Datenaugen-Abtastkalibrierung
  - DRAM-Test-Algorithmusauswahl
  - DRAM-Testdurchführung
  - Abfrage des Testergebnis



- Interface BIST(IBIST) beherrscht:
  - AMB Konfiguration
  - IF-Test-Algorithmusauswahl
  - IF-HighSpeed-Testdurchführung
  - Abfrage des Testergebnis

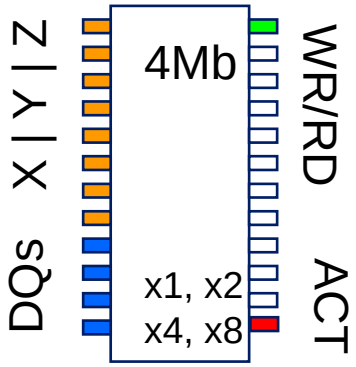


MBIST und IBIST Steuerung erfolgt durch I²C-/SMB-Bus Programmierung Mikrocode-Registern.

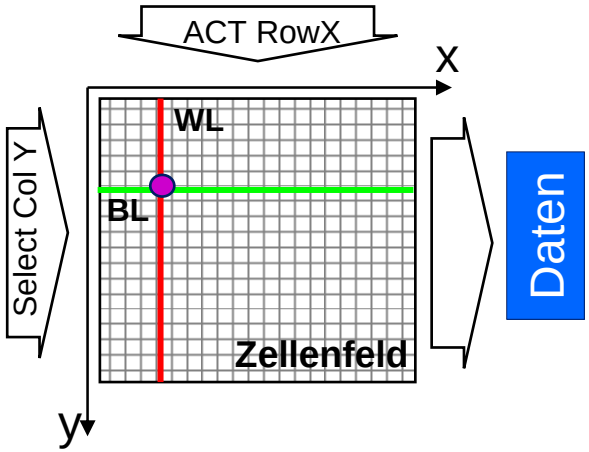
Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Der „Speicher“-Automat

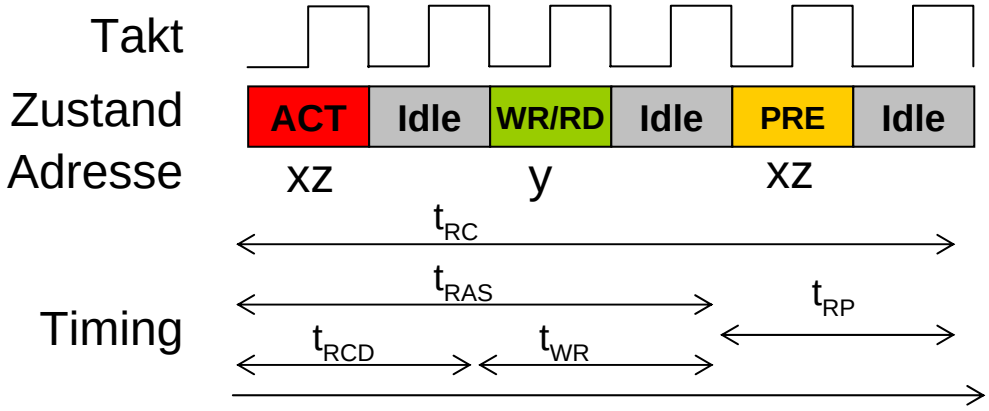
Speicherbaustein



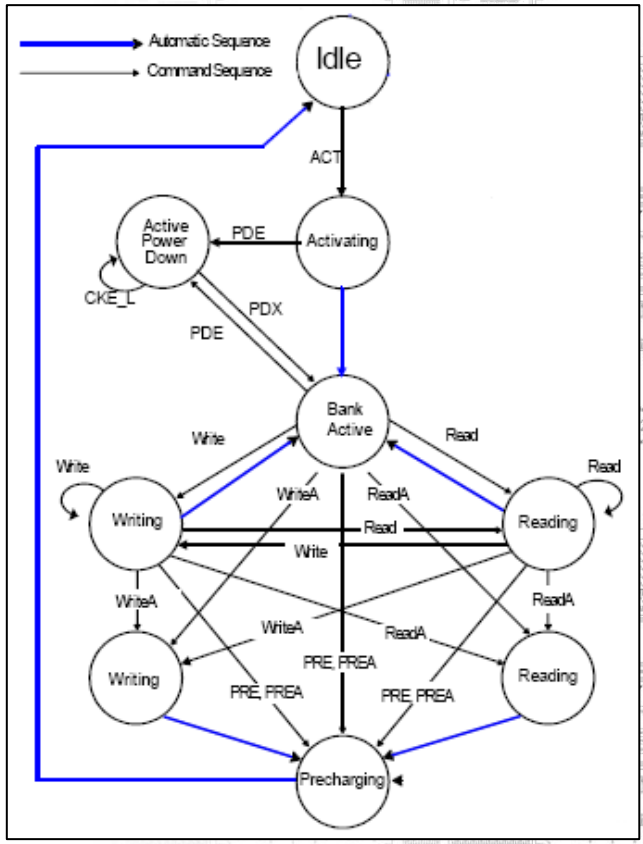
Speicherzugriff



Vereinfachter Speicherzugriffsverlauf

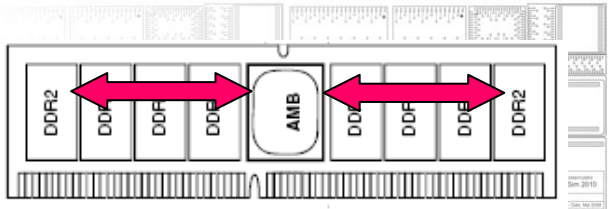


Zustandsdiagramm erlaubter Speicherzugriffe:

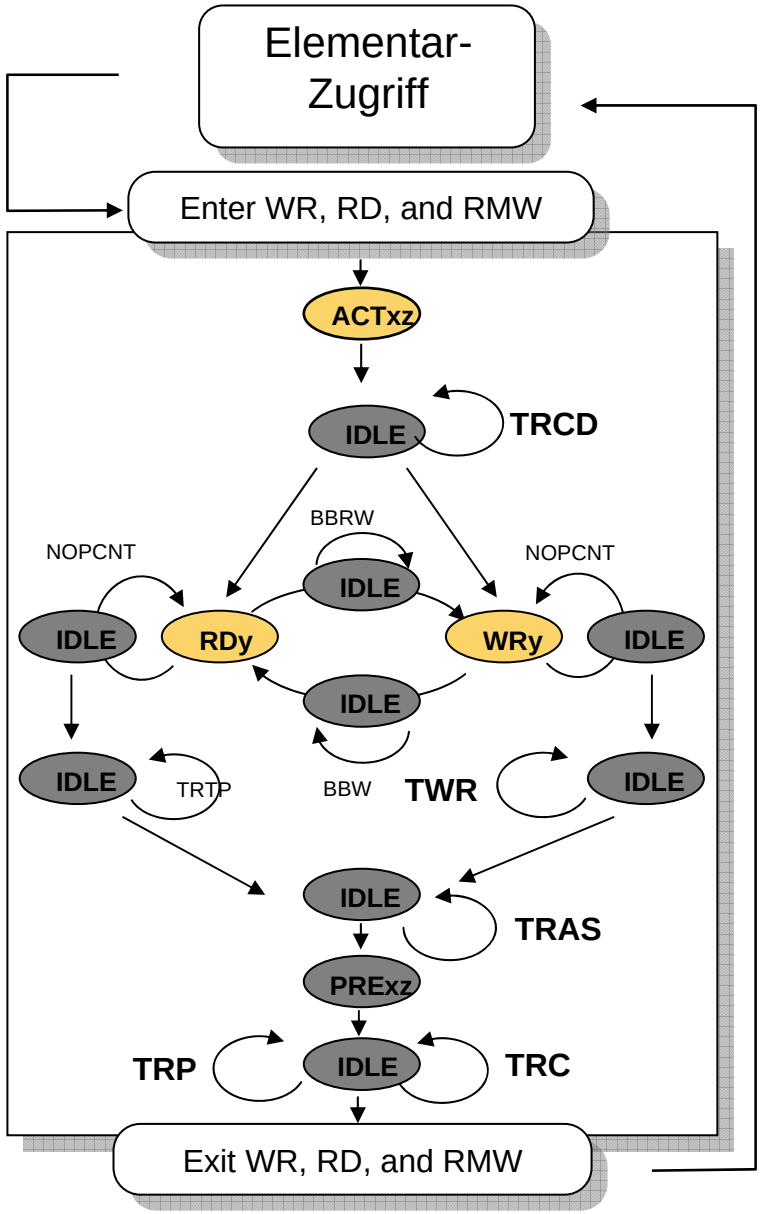


Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

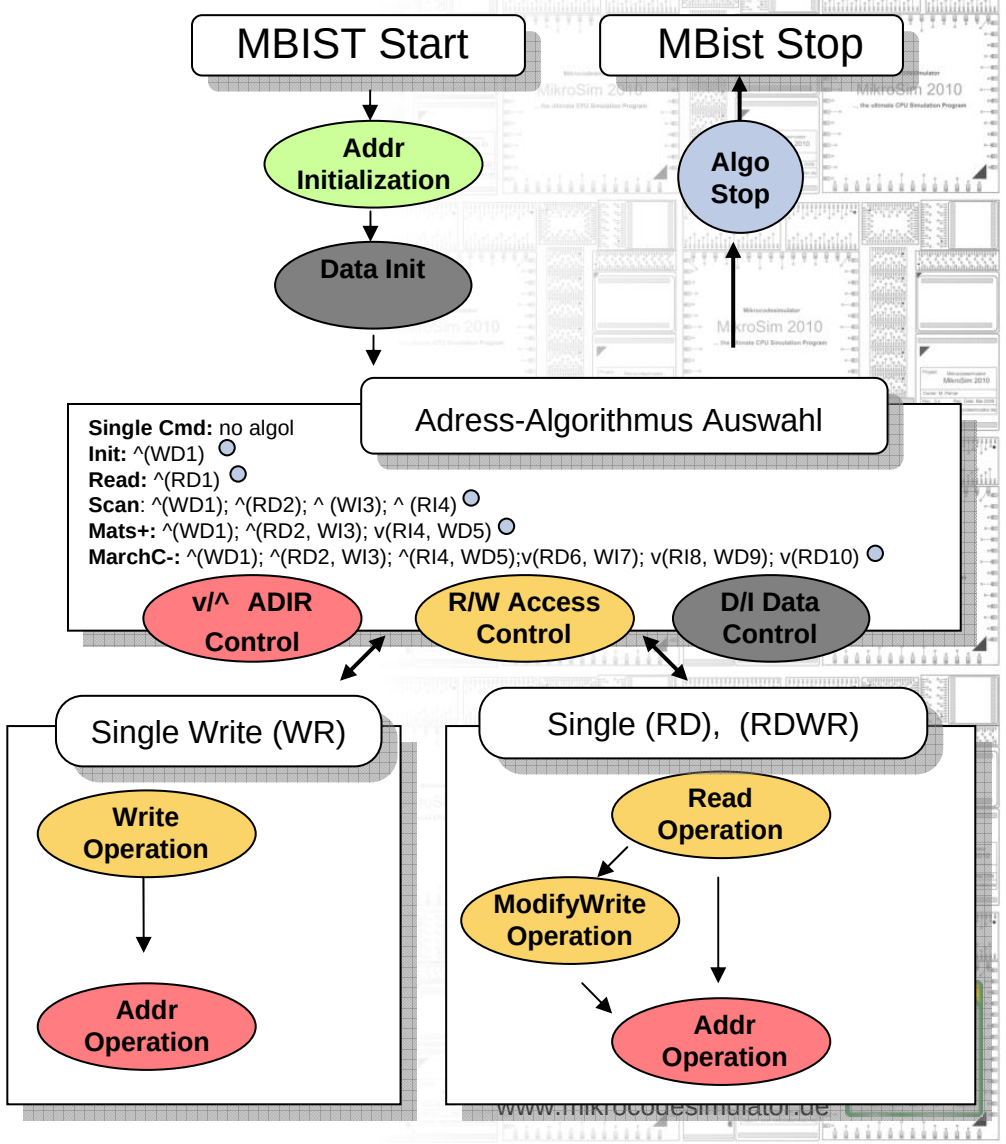
Der Speichertest-Automat von MBIST



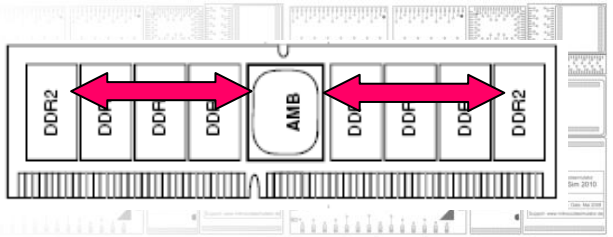
Einzelzugriff-Steuerung



Adess-Sequenz-Steuerung

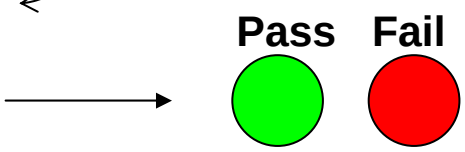


Mikrocode-Programmierung und Test mit MBIST

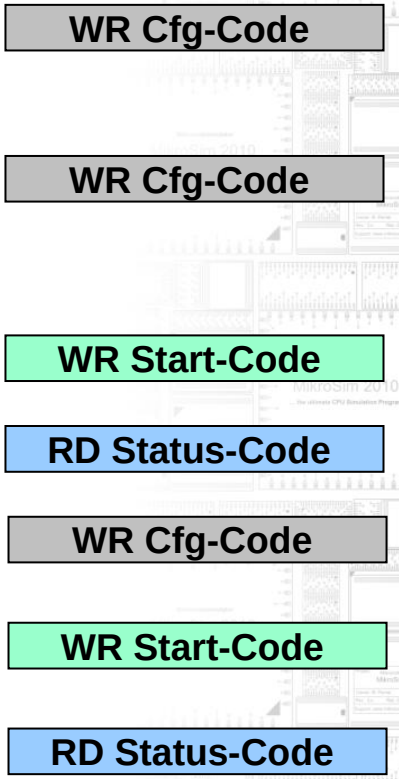


Mikrocode-Programmierschritte

1. Modul-Initalisierung
   @ Speicher: Power-Up und Bausteinkonfiguration
   @ AMB: Einstellung des Speichertyps
2. Timing-Set-Programmierung @ AMB
3. Speicherkanal-Kalibrierung @ AMB
   z.B. Datenaugenabtastung
4. Test-Algorithmus Vorbereitung @ AMB
   Datenmuster, Address-Sequenz, etc.
5. MBIST Start @ AMB
6. Testabfrage @ AMB



Mikrocodeprogrammierung über I2C-Bus

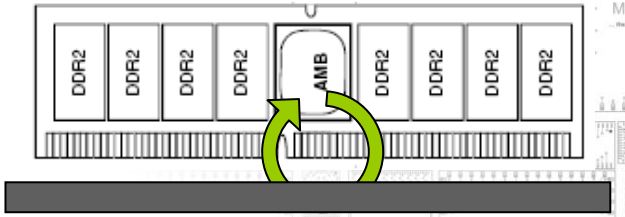


Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Mikrocode-Programmierung und Interface-Test mit IBIST

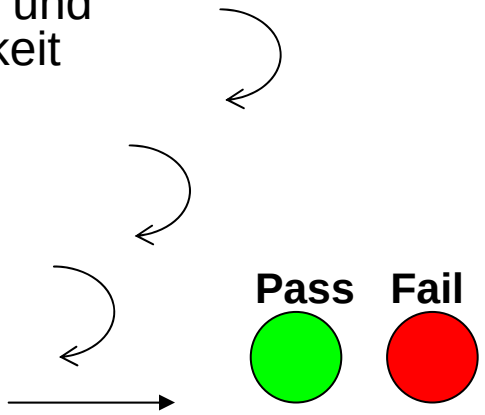
Loop-Back-Test-Methode:

- „Kurzschluss-Sockel“-Test
- Testen des Initialisierungsprotokolls und Selbsttest mit definierten Datenpaketen
- Läuft bei Applikationsgeschwindigkeit (2.4-4.8 Gbit/s)

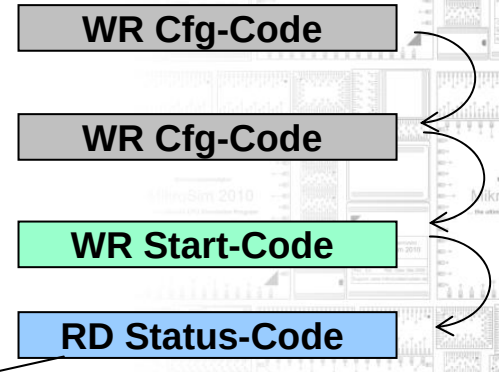


Mikrocode-Programmierschritte

1. Konfiguration des Datenmusters
2. Konf. der Treiberstärke und Empfängerempfindlichkeit
3. Kalibrierung
4. Starte Datengenerator
5. Warte ~ 100 ms



Mikrocodeprogrammierung über I2C-Bus



Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de

Zusammenfassung und Resümee

Teil I: Didaktische Einführung in den Mikrocode

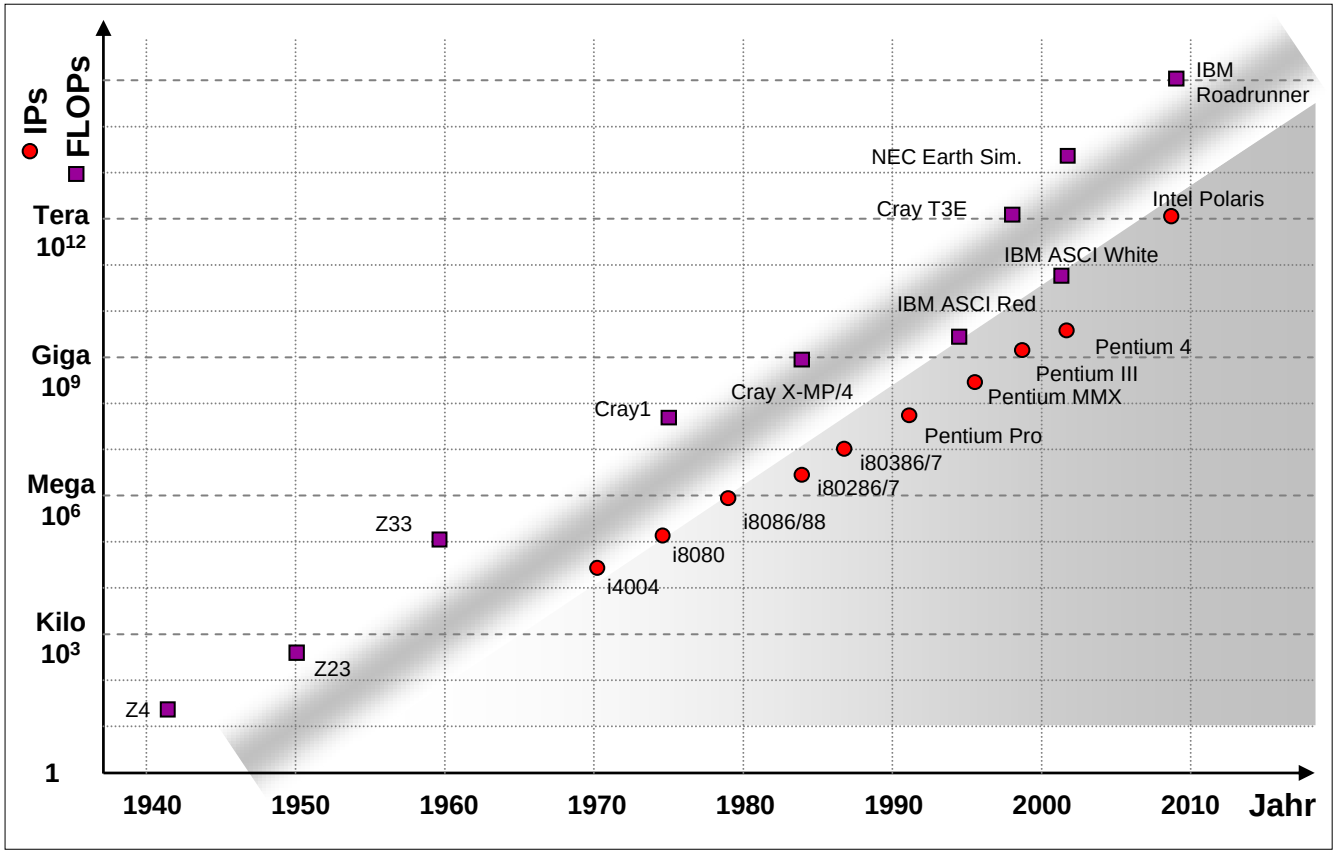
Motivation, Lernziele und Lehrpotenzial des e-Learning-Tools „MikroSim“

Teil II: Applikationsrelevanz des Mikrocodes

Evolution der Rechenautomaten, Bedeutung gestern - und heute “!“

Teil III: Mikrocode zum Testen von Schaltkreisen

Am Beispiel von FBDIMM



Mikrocode als flexible Steuerung von Automaten hat eine öffentlich „wahrgenommene“ Bedeutung für Prozessoren und Großrechner.

„Unbemerkt“ nehmen jedoch unzählige Mikrocode-gesteuerte Automaten im Hintergrund unscheinbaren Einfluss auf Produktion, Test und Anwendung integrierter elektronischer „Bauelemente“.

Doch wir bemerken sie oft erst, wenn sie nicht funktionieren!

Dr. Martin Perner
0/1-SimWare
info@mikrocodesimulator.de
www.mikrocodesimulator.de