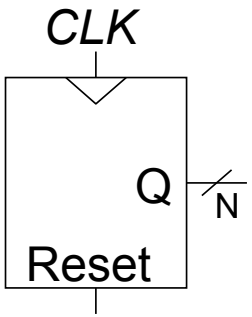
Nikolausfeier

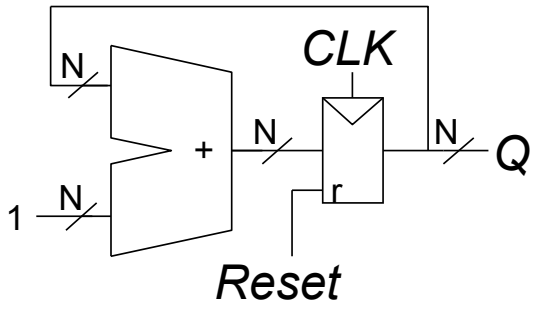
Zähler

- Einfachster Fall: Inkrementieren zu jeder positiven Taktflanke
- Zählen durch einen Zyklus von Werten, Beispiel für 3b Breite
  - 000, 001, 010, 011, 100, 101, 110, 111, 000, 001…
- Beispielanwendungen
  - Digitaluhren
  - Programmzähler: Zeigt auf nächste auszuführende Instruktion

Symbol



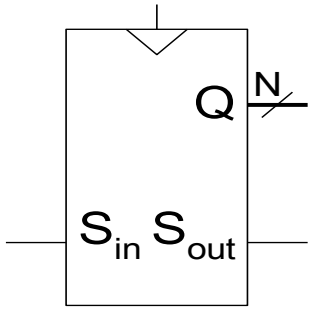
Aufbau



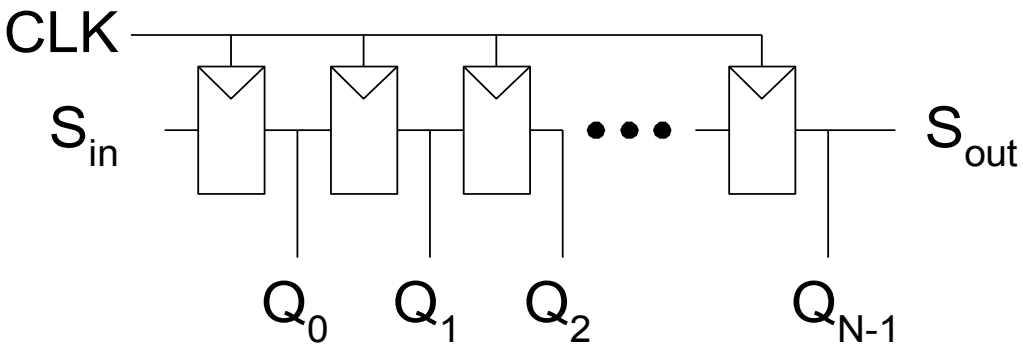
Schieberegister

- Auch: FIFO (*first-in first-out*)
- Schiebe einen neuen Wert jeden Takt ein
- Schiebe einen alten Wert jeden Takt aus
- Kann auch agieren als Seriell-nach-Parallel-Konverter
  - Konvertiert serielle Eingabe (S_{in}) in parallele Ausgabe ($Q_{0:N-1}$)

Symbol:

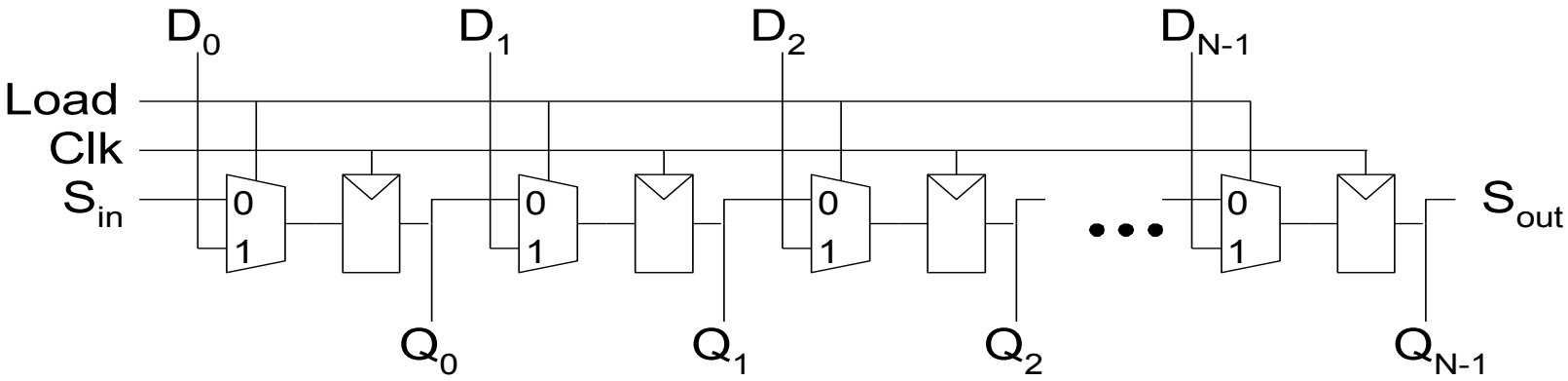


Aufbau:



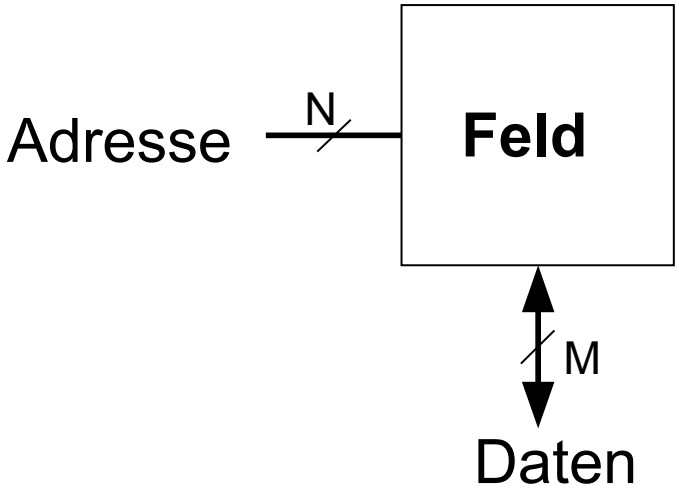
Schieberegister mit parallelem Laden

- Bei *Load* = 1: Agiert als normales *N*-bit Register
- Bei *Load* = 0: Agiert als Schieberegister
- Verwendbar als
  - Seriell-nach-Parallelkonverter (S_{in} nach $Q_{0:N-1}$)
  - Parallel-nach-Seriellkonverter ($D_{0:N-1}$ nach S_{out})



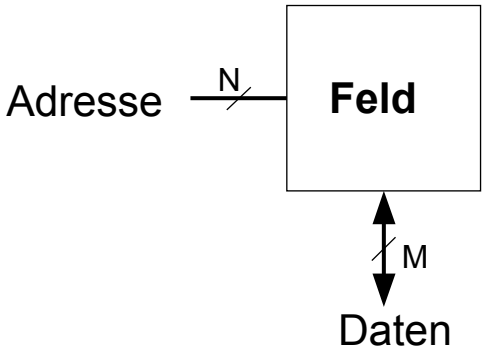
Speicherfelder

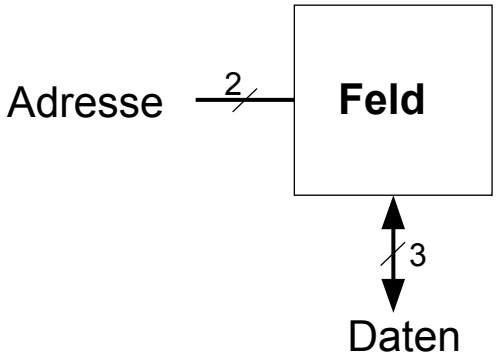
- Können effizient größere Datenmengen speichern
- Drei weitverbreitete Typen:
  - Dynamischer Speicher mit wahlfreiem Zugriff
    - (*Dynamic random access memory,* DRAM)
  - Statischer Speicher mit wahlfreiem Zugriff
    - (*Static random access memory,* SRAM)
  - Nur-Lesespeicher (*Read only memory*, ROM)
- An jede *N*-bit Adresse kann ein *M*-bit breites Datum geschrieben werden



Speicherfelder

- Zweidimensionales Feld von Bit-Zellen
- Jede Bit-Zelle speichert ein Bit
- Feld mit *N* Adressbits und *M* Datenbits:
  - 2^N Zeilen und *M* Spalten
  - **Tiefe:** Anzahl von Zeilen (Anzahl von Worten)
  - **Breite:** Anzahl von Spalten (Bitbreite eines Wortes)
  - **Feldgröße:** Tiefe × Breite = $2^N \times M$





Adresse	Daten		
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

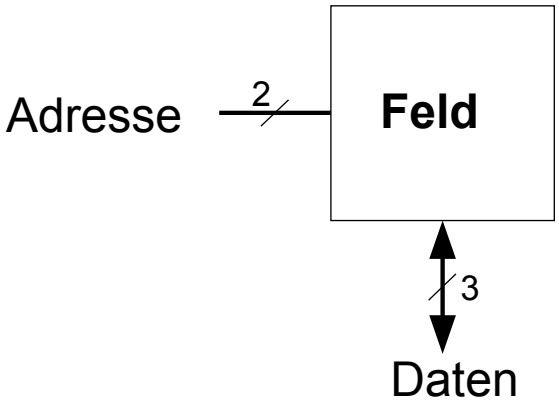
Tiefe

Breite

Beispiel: Speicherfeld

- $2^2 \times$ 3-Bit Feld
- Anzahl Worte: 4
- Wortbreite: 3-Bit
- Beispiel: 3-Bit gespeichert an Adresse 2'b10 ist 3'b100

Beispiel:

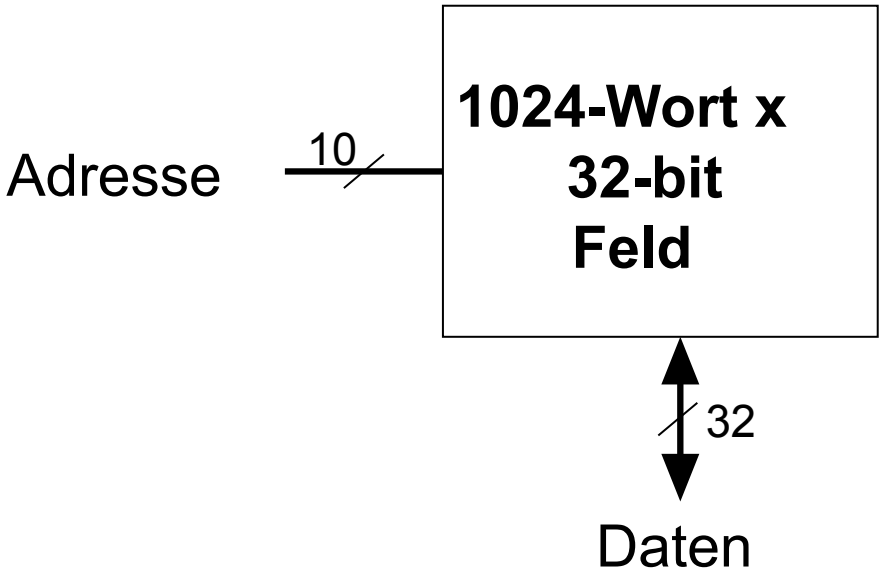


Adresse	Daten		
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

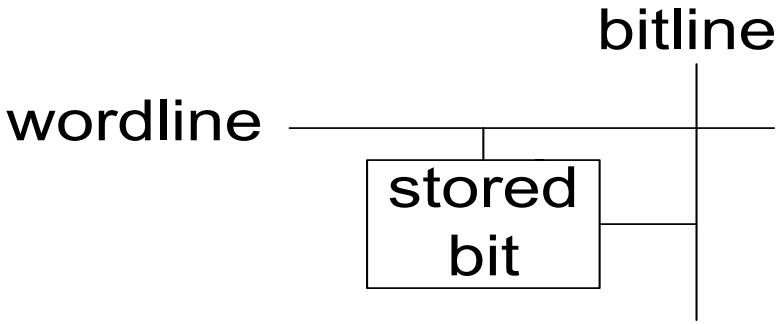
Tiefe

Breite

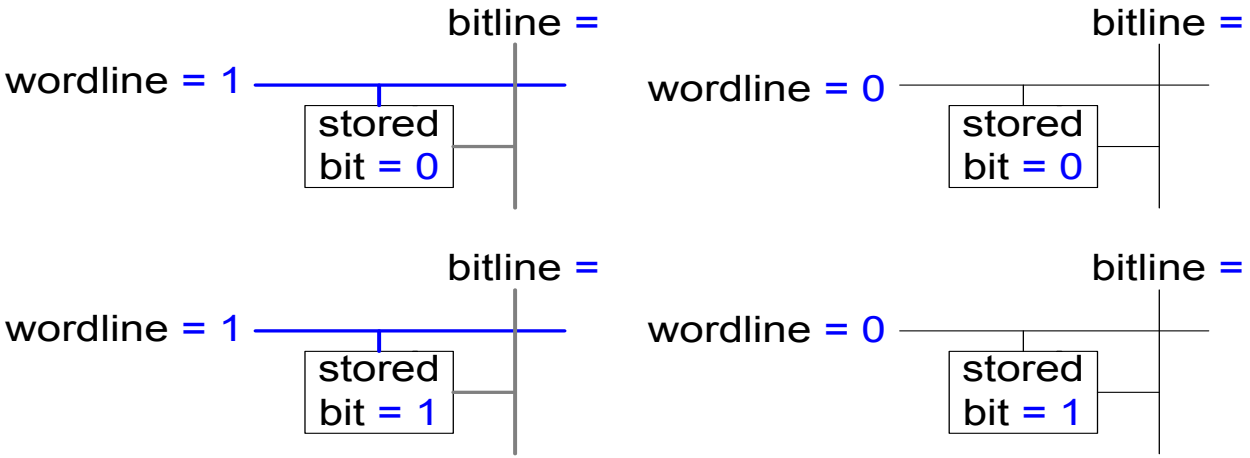
Speicherfelder



Bit-Zellen für Speicherfelder

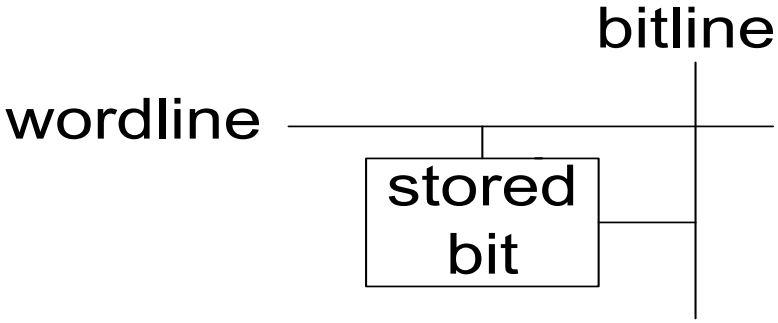


Beispiel:

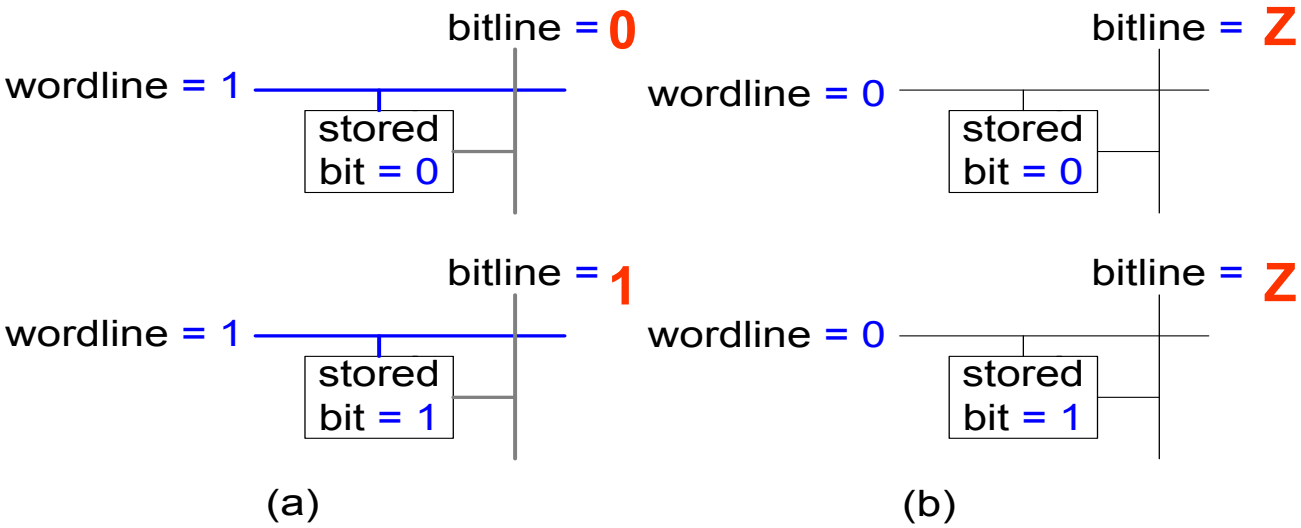


(a) (b)

Aufbau von Speicherfeldern aus Bit-Zellen

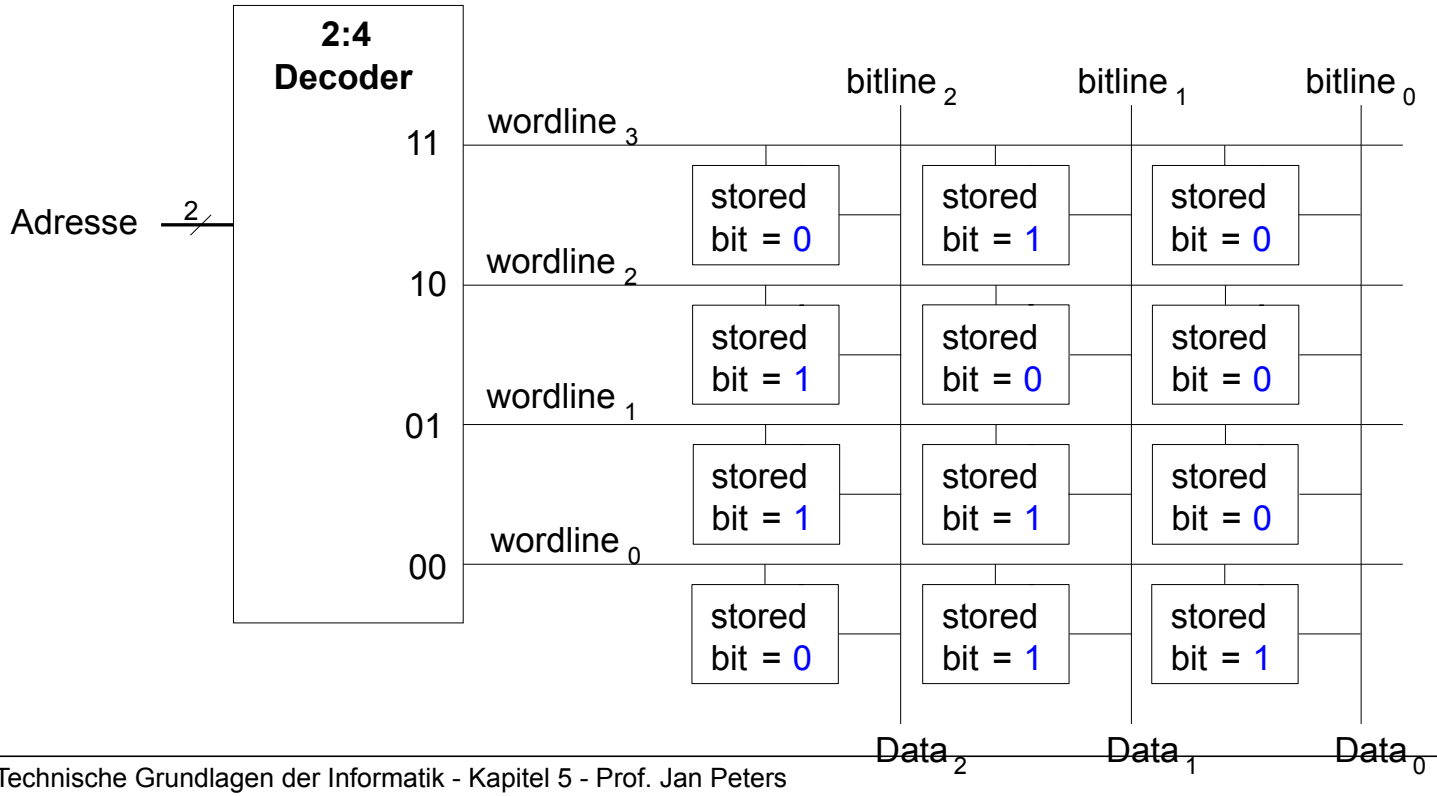


Beispiel:



Aufbau von Speicherfeldern

- **Wordline:**
  - Vergleichbar mit Enable-Signal
  - Erlaubt Zugriff auf eine Zeile des Speichers zum Lesen oder Schreiben
  - Entspricht genau einer eindeutigen Adresse
  - Maximal eine Wordline ist zu jedem Zeitpunkt HIGH



Arten von Speicher: Historische Sicht

TECHNISCHE UNIVERSITÄT DARMSTADT

- Speicher mit wahlfreiem Zugriff (RAM)
- Nur-Lese Speicher (ROM)

RAM: Random-Access Memory

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Flüchtig:** Speicherinhalte gehen bei Verlust der Betriebsspannung verloren
- Kann i.d.R. gleich schnell gelesen und geschrieben werden
- Zugriff auf beliebige Adressen mit ähnlicher Verzögerung möglich
- Hauptspeicher moderner Computer ist dynamisches RAM (DRAM)
  - Aktuell & genauer: DDR3-SDRAM
  - *Double Data Rate 3 - Synchronous Dynamic Random Access Memory*

- Name „RAM“ hat Bedeutung:
  - Auf eine beliebige („Random“) Speicherzelle kann direkt zugegriffen werden.
  - Steht im Gegensatz zu „Sequential Access Memory“ von (früheren) Bandspeichern und Trommelspeichern …

ROM: Read-Only Memory

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Nicht-flüchtig:** Erhält Speicherinhalt auch ohne Betriebsspannung
- Schnell lesbar
- Schreibbar nur sehr langsam (wenn überhaupt)
- Flash-Speicher ist in diesem Sinne ein ROM
  - Kameras
  - Handys
  - MP3-Player

- Auch hier Nomenklatur „ROM“ historisch
  - Auch aus ROMs kann von beliebigen Adressen gelesen werden
  - Es gibt auch schreibbare Arten von ROMs
    - PROMs, EPROMs, EEPROMs, Flash

Arten von RAM

TECHNISCHE UNIVERSITÄT DARMSTADT

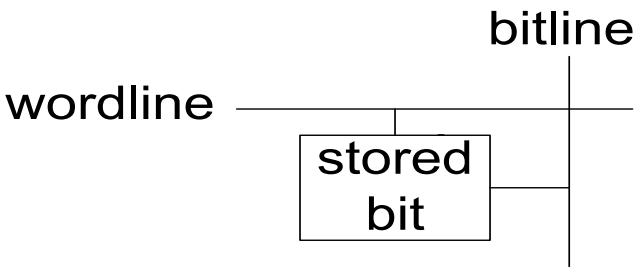
- Zwei wesentliche Typen:
  - Dynamisches RAM (DRAM)
  - Statisches RAM (SRAM)
- Verwenden unterschiedliche Speichertechniken in den Bit-Zellen:
  - DRAM: Kondensator
  - SRAM: Kreuzgekoppelte Inverter

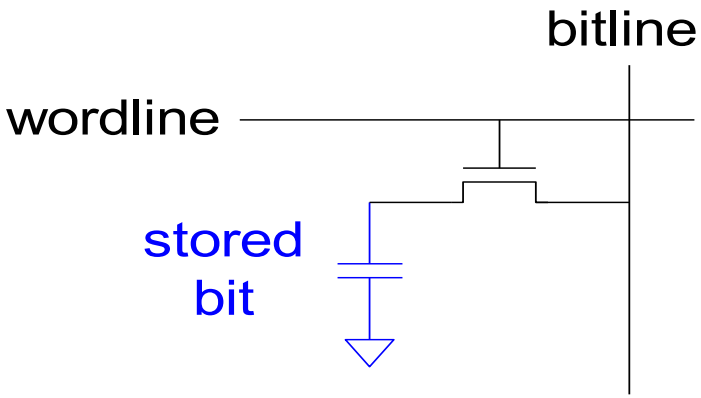
Robert Dennard, 1932 -

- Erfand 1966 bei IBM das DRAM
- Anfangs große Skepsis, ob Technik praktikabel
- Seit Mitte der 1970er Jahre ist DRAM die am weitesten verbreitete Speichertechnik in Computern

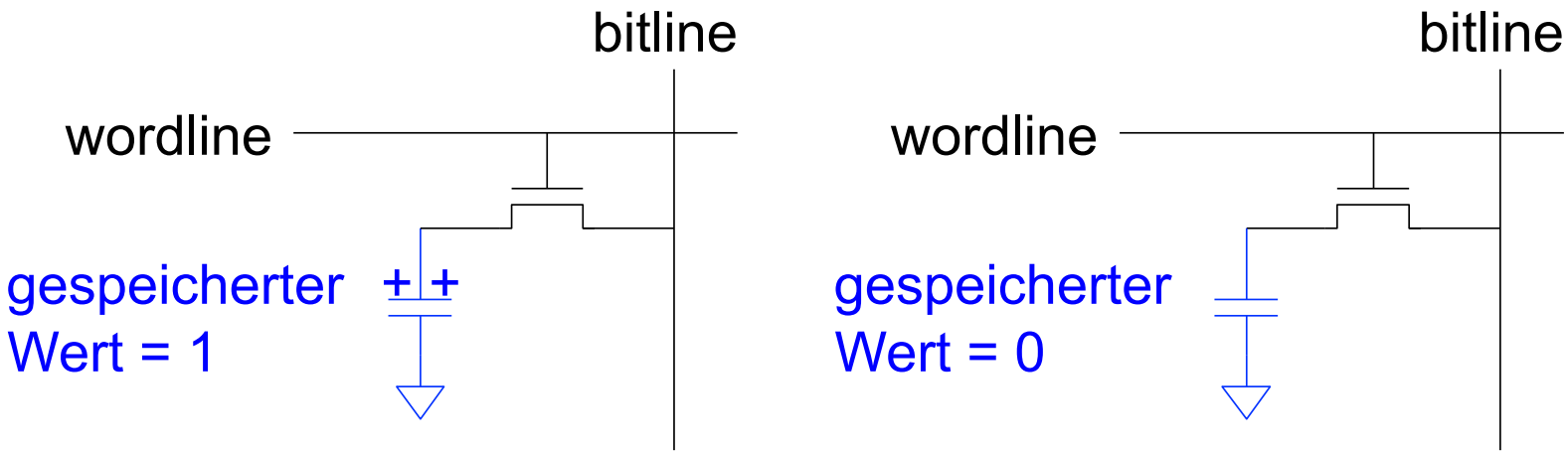
DRAM Bit-Zelle

- Datenbit wird als Ladezustand eines Kondensators gespeichert
- Dynamisch: Der Speicherwert muss periodisch neu geschrieben werden
  - Auffrischung alle paar Millisekunden erforderlich (üblich: 64ms)
  - Kondensator verliert Ladung durch Leckströme
  - … und beim Auslesen



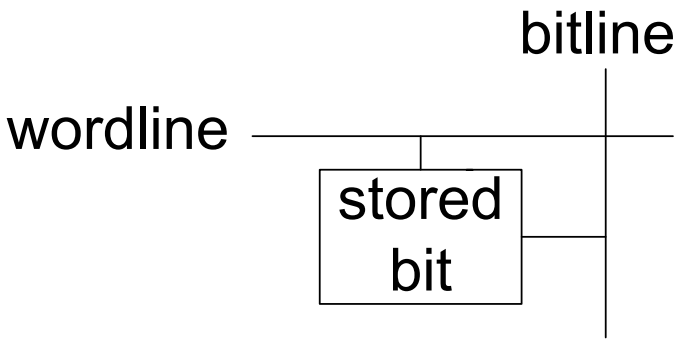


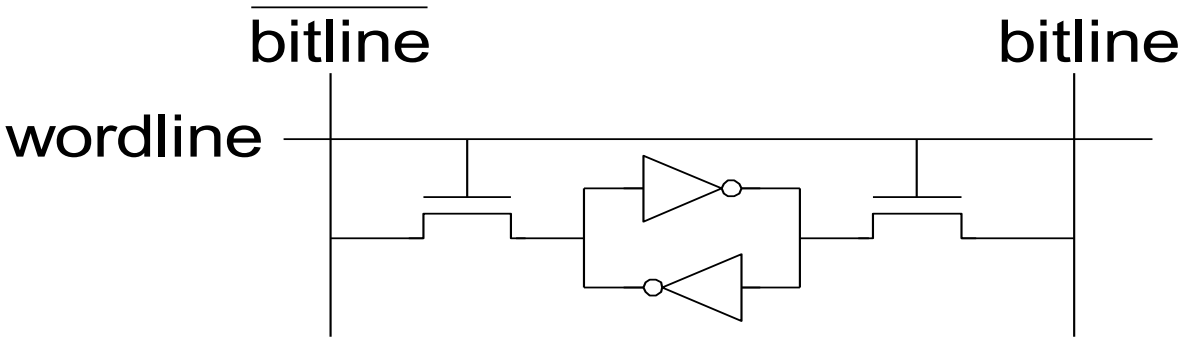
DRAM Bit-Zelle



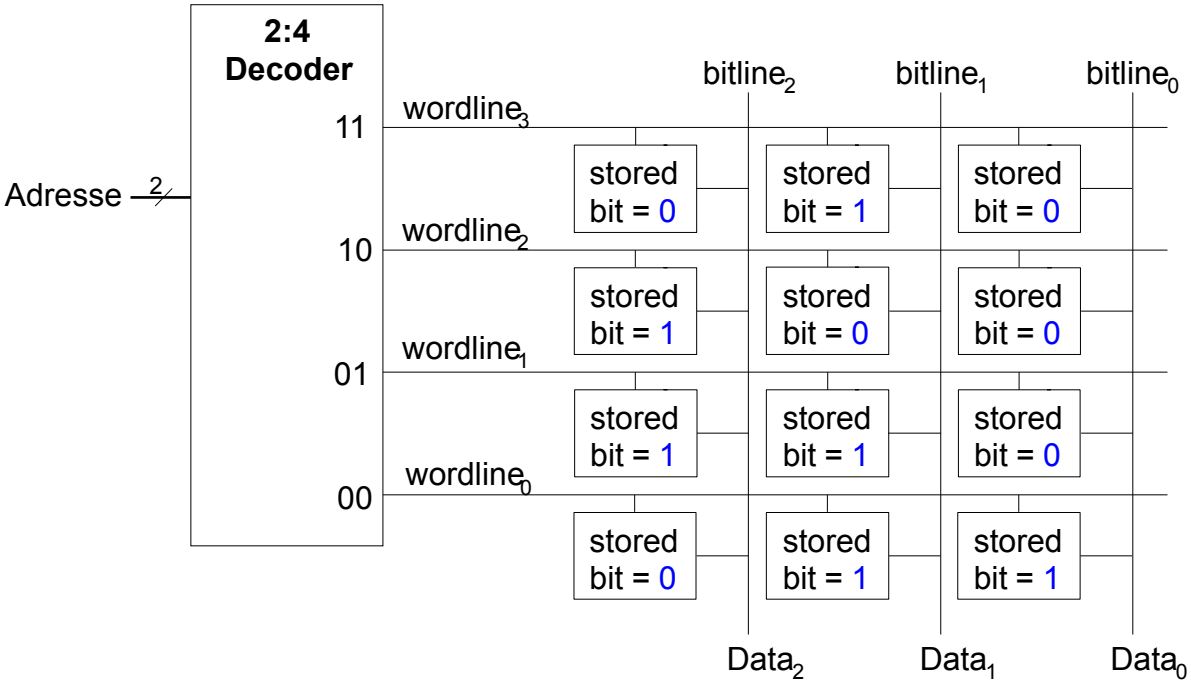
SRAM Bit-Zelle

- Datenbit wird als Zustand von rückgekoppelten Invertern gespeichert
- Statisch: Keine Auffrischung erforderlich
  - Inverter treiben Werte auf gültige Logikpegel

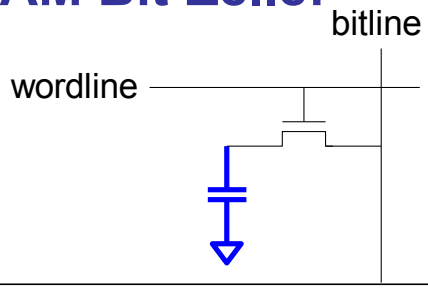




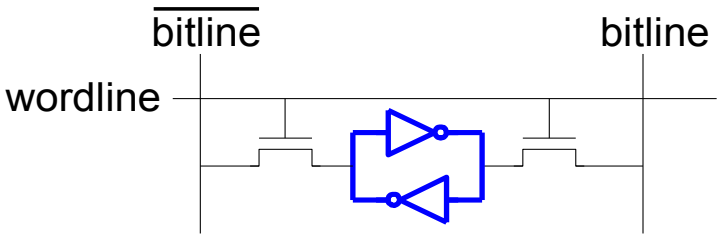
Speicherfelder



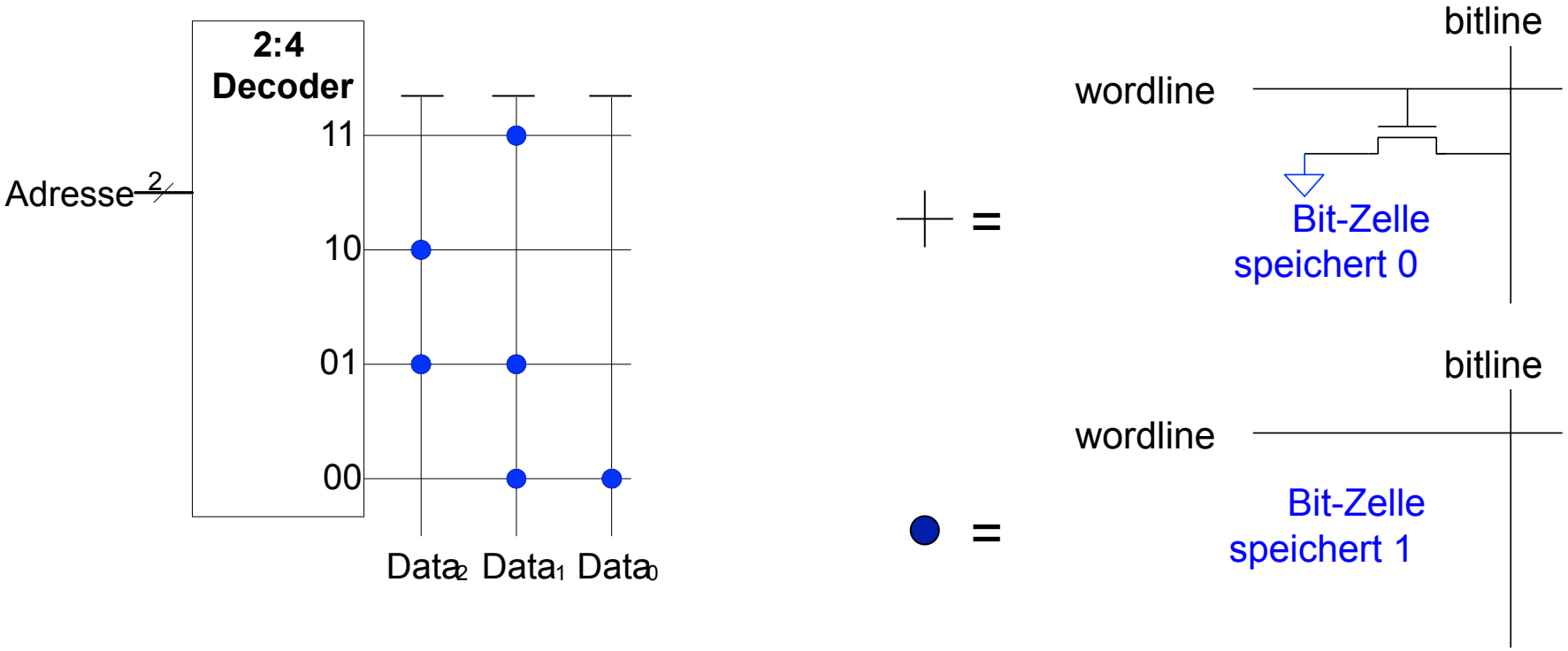
DRAM Bit-Zelle:



SRAM Bit-Zelle:



ROMs: Aufbau der Bit-Zellen

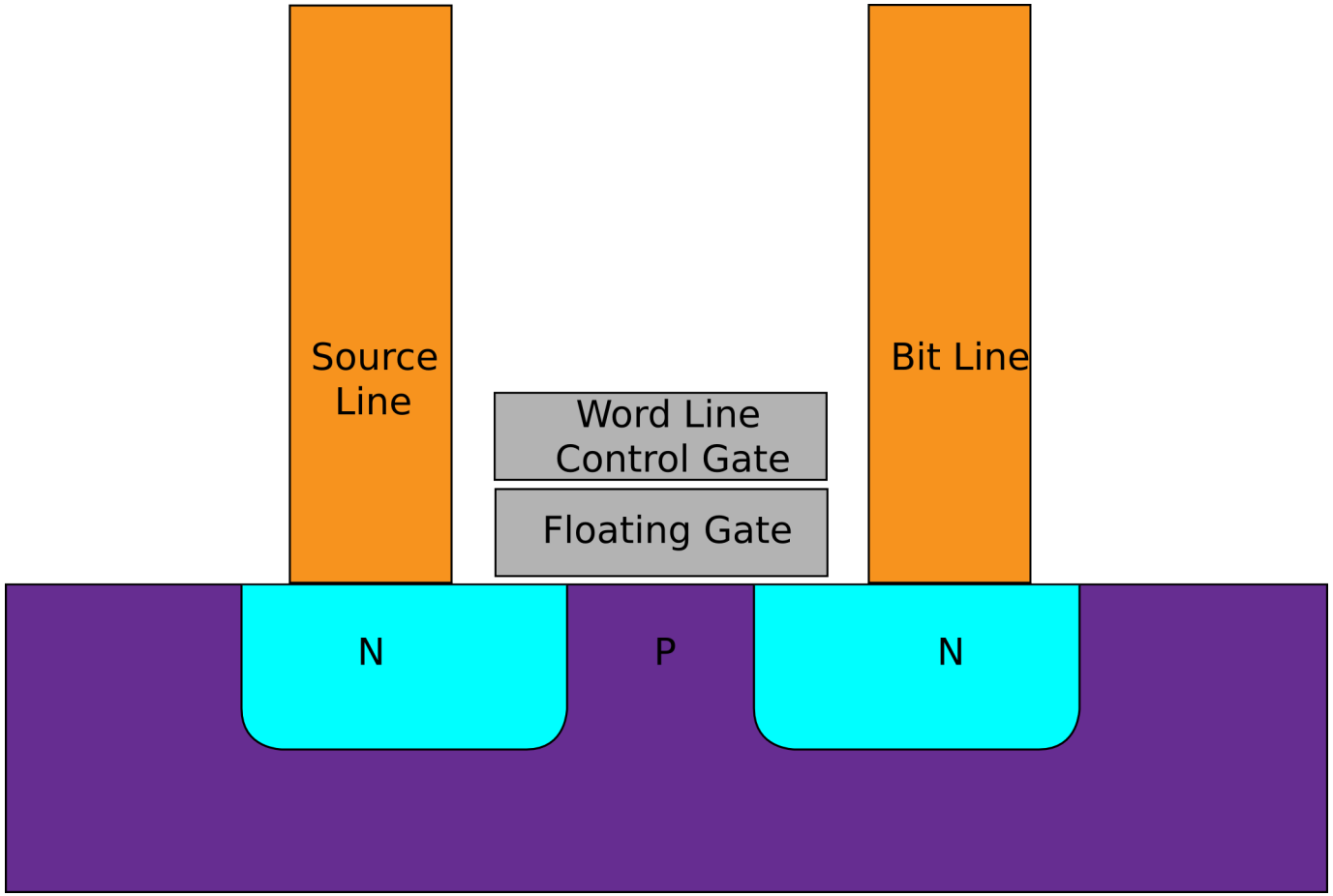


Bitlines sind schwach auf HIGH getrieben

Fujio Masuoka, 1944-

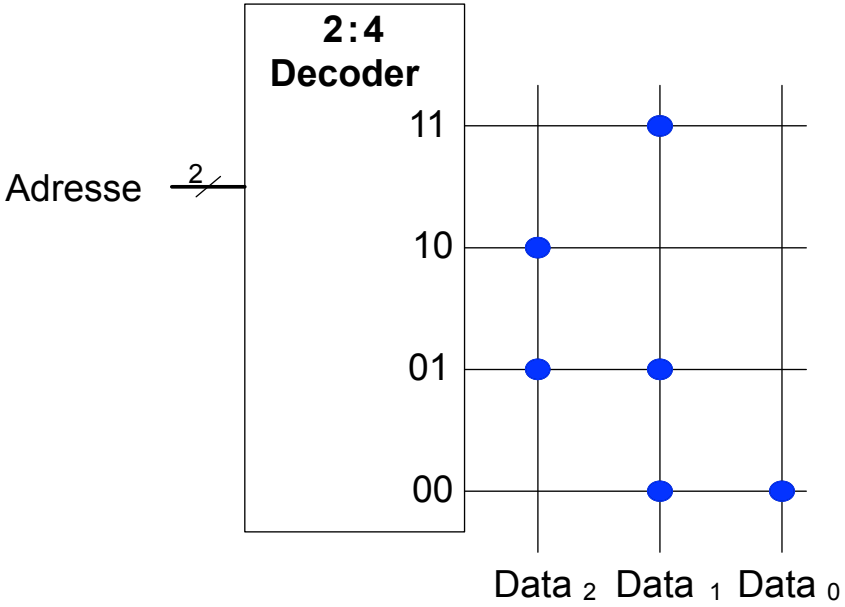
- Entwickelte Speicher und schnelle Schaltungen bei Toshiba von 1971-1994
- Erfand Flash-Speicher als eigenes ungenehmigtes Projekt in den späten 1970ern
  - An Wochenenden und abends
- Löschvorgang erinnerte ihn an Kamerablitz
  - Deshalb Flash-Speicher
- Toshiba kommerzialisierte Technik nur zögerlich
- Erste kommerzielle Chips von Intel in 1988
- Flash-Produkte haben großen Erfolg
  - Derzeit USD 25 Milliarden Umsatz / Jahr

Flash-Speicher: Bit-Zelle



Quelle:Wikipedia

ROMs als Datenspeicher

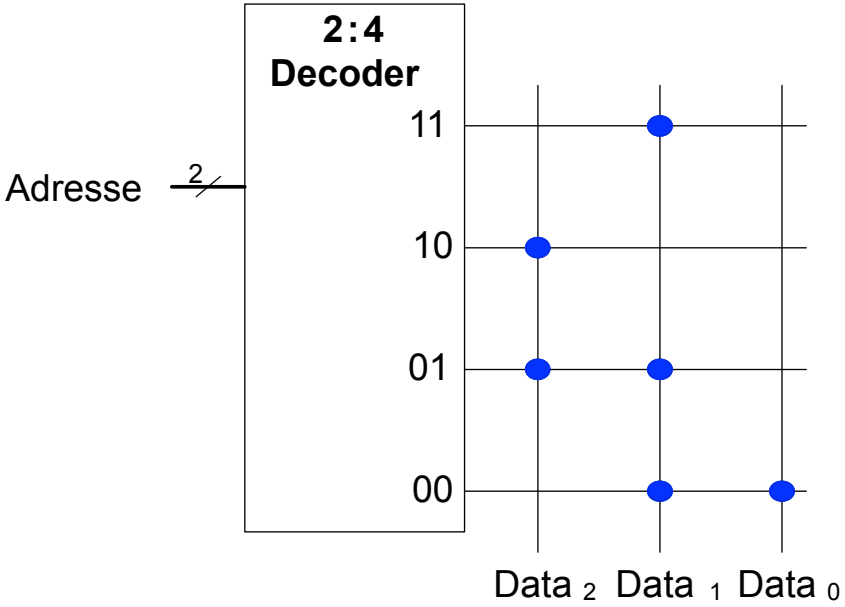


Adresse	Daten		
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

Tiefe

Breite

ROMs als Wertetabellen für boolesche Logik



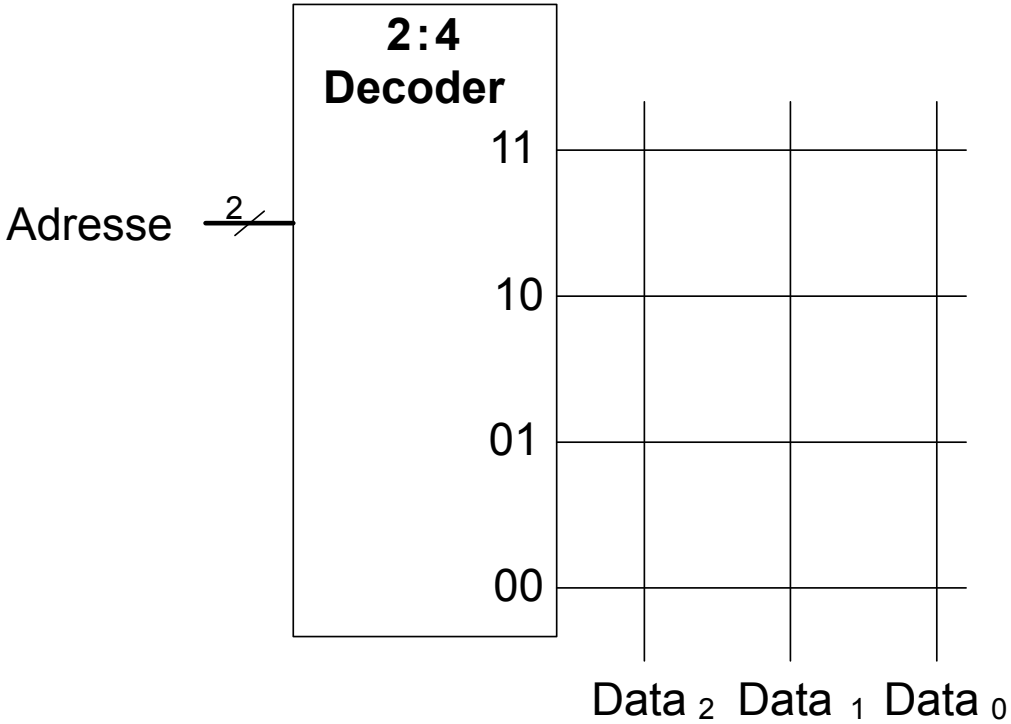
$$Data_2 = A_1 \text{ Å } A_0$$

$$Data_1 = \overline{A_1} + A_0$$

$$Data_0 = \overline{A_1}\,\overline{A_0}$$

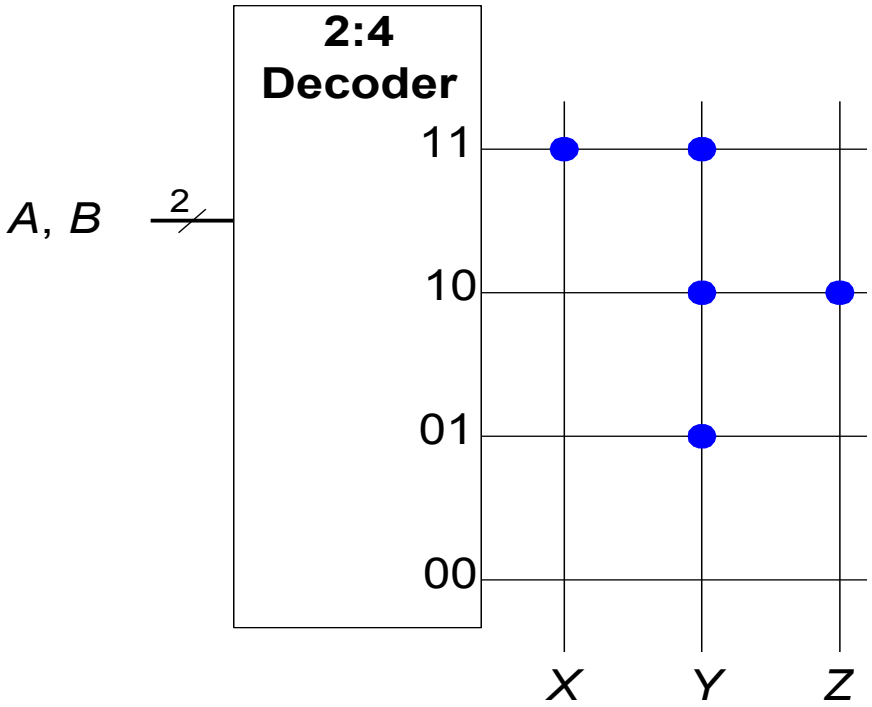
Beispiel: Logik aus ROMs

- Implementierung der folgenden logischen Funktionen durch $2^2 \times 3$-bit ROM:
  - $X = AB$
  - $Y = A + B$
  - $Z = A\overline{B}$

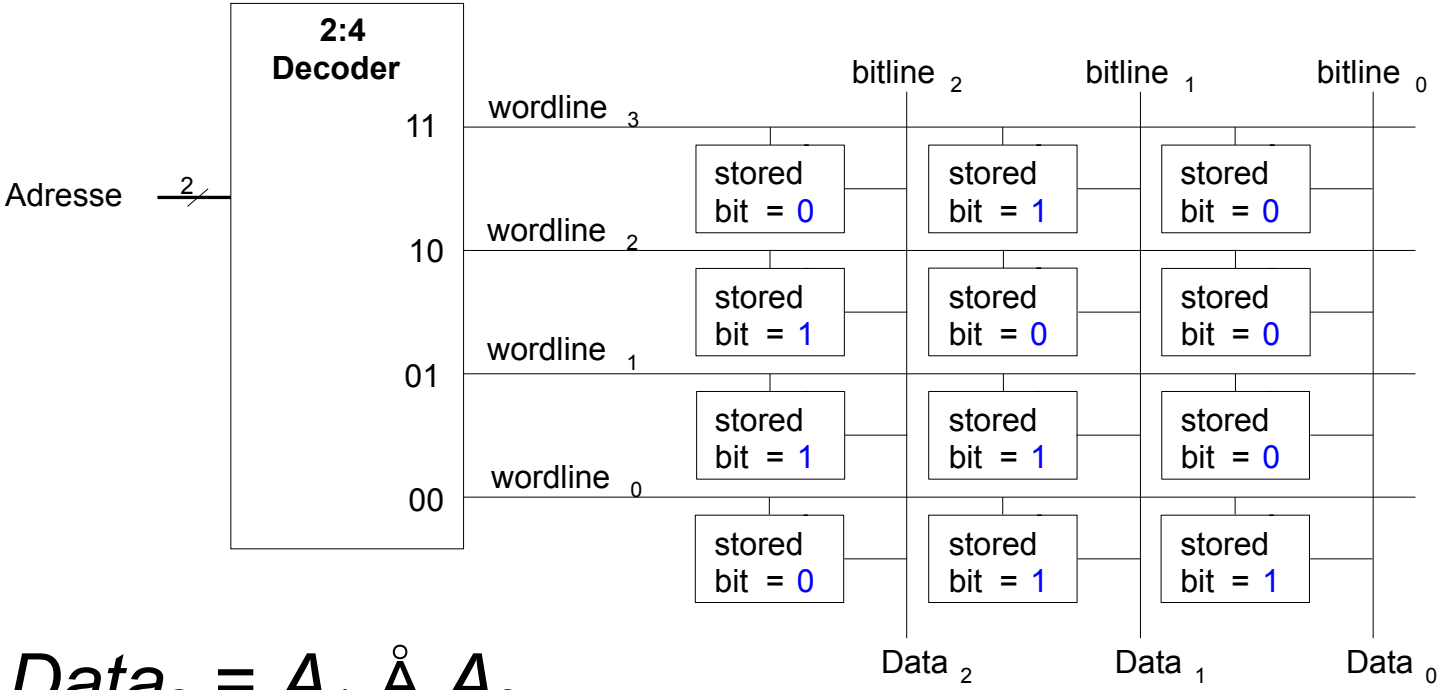


Beispiel: Logik aus ROMs

- Implementierung der folgenden logischen Funktionen durch $2^2 \times 3$-bit ROM:
  - $X = AB$
  - $Y = A + B$
  - $Z = A\overline{B}$



Logik aus beliebigem Speicherfeld



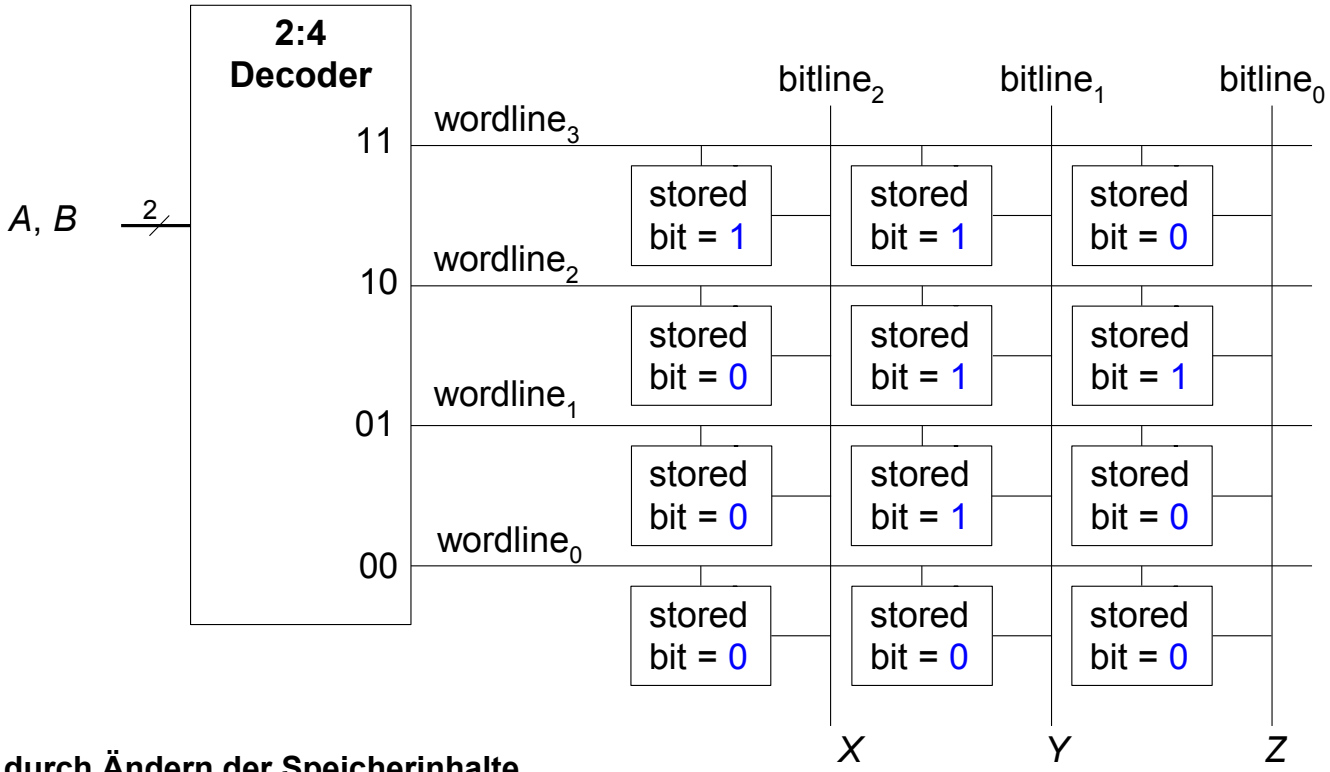
$$Data_2 = A_1 \mathring{A} A_0$$

$$Data_1 = \overline{A_1} + A_0$$

$$Data_0 = \overline{A_1}\,\overline{A_0}$$

Logik aus beliebigem Speicherfeld

- Implementierung der folgenden logischen Funktionen durch $2^2 \times 3$-bit RAM:
  - $X = AB$
  - $Y = A + B$
  - $Z = A\overline{B}$



Andere Funktion nur durch Ändern der Speicherinhalte

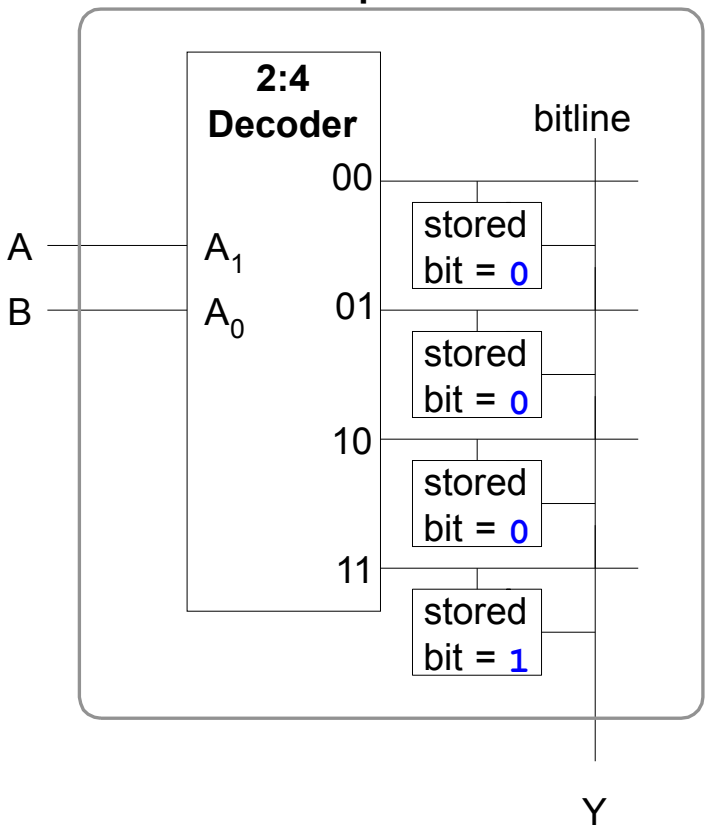
Logik aus beliebigen Speicherfeldern

- Speicherfelder speichern Wertetabellen
  - Lookup-Tables (LUTs)
- Wort aus Eingangsvariablen bildet Adresse
- Für jede Kombination von Eingangsvariablen ist Funktionsergebnis abgespeichert

Werte-tabelle

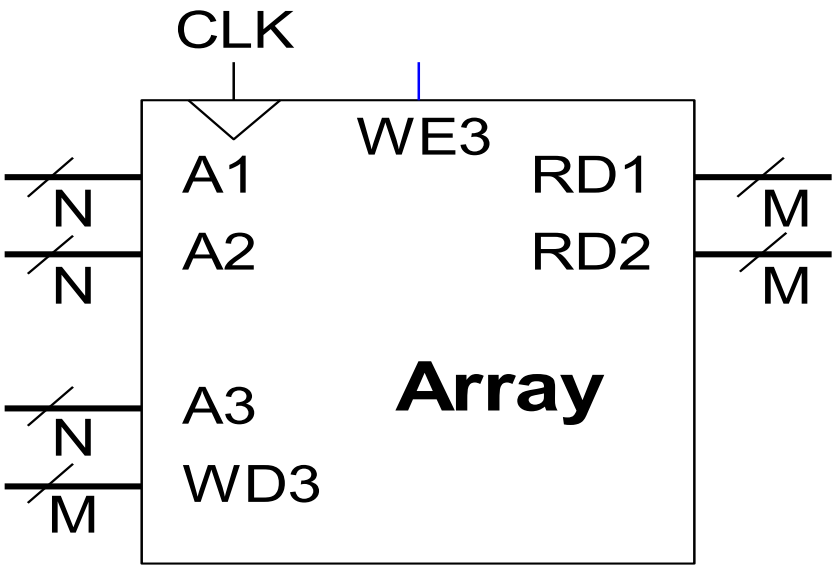
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

4-Wort x 1-bit Speicherfeld



Multi-Port-Speicher

- **Port:** Zusammengehörige Anschlüsse für Adresse und Datum
- Drei-Port Speicher
  - 2 Lese-Ports (A1/RD1, A2/RD2)
  - 1 Schreib-Port (A3/WD3, Signal WE3 löst Schreiben aus)
- Kleine Multi-Port-Speicher werden als Registerfelder bezeichnet
  - Werden z.B. in Prozessoren eingesetzt



Speicherfeld in Verilog

TECHNISCHE UNIVERSITÄT DARMSTADT

```
// 256 x 3b Speicher mit Schreib/Lese-Port
module dmem(   input  logic          clk, we,
               input  logic [7:0]     a
               input  logic [2:0]     wd,
               output logic [2:0]     rd);

  logic [2:0]  RAM[255:0];

  assign rd = RAM[a];
  always_ff @(posedge clk)
      if (we)
          RAM[a] <= wd;
endmodule
```

Logikfelder (*logic arrays*)

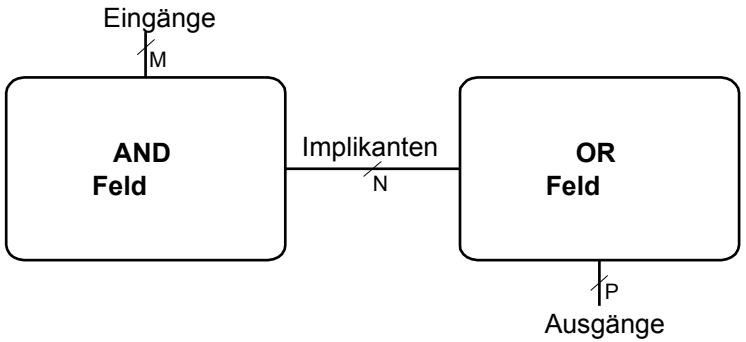
TECHNISCHE UNIVERSITÄT DARMSTADT

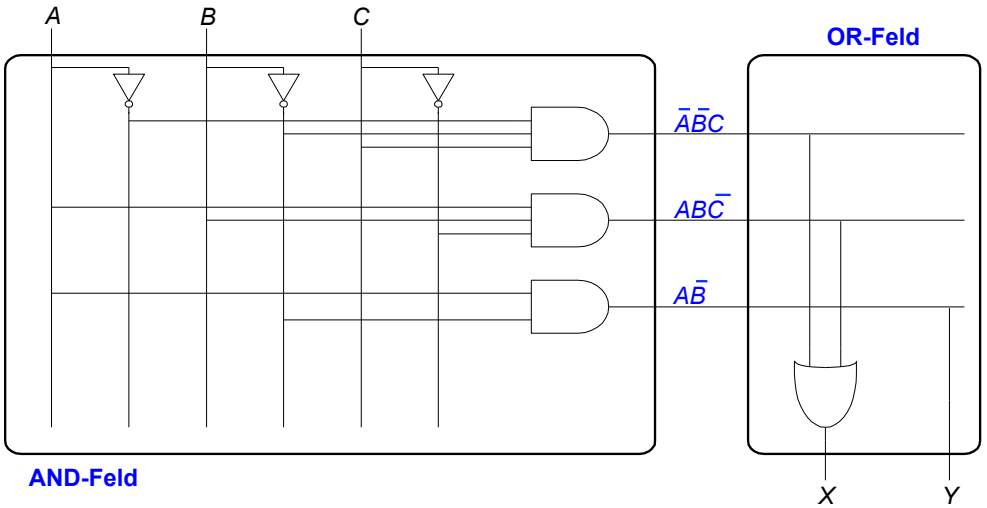
- Programmable Logic Arrays (PLAs)
  - AND Feld gefolgt von OR Feld
  - Kann nur kombinatorische Logik realisieren
  - Feste interne Verbindungen, spezialisiert für DNF (SoP-Form)

- Field Programmable Gate Arrays (FPGAs)
  - Feld von konfigurierbaren Logikblöcken (CLBs)
  - Können kombinatorische und sequentielle Logik realisieren
  - Programmierbare Verbindungsknoten zwischen Schaltungselementen

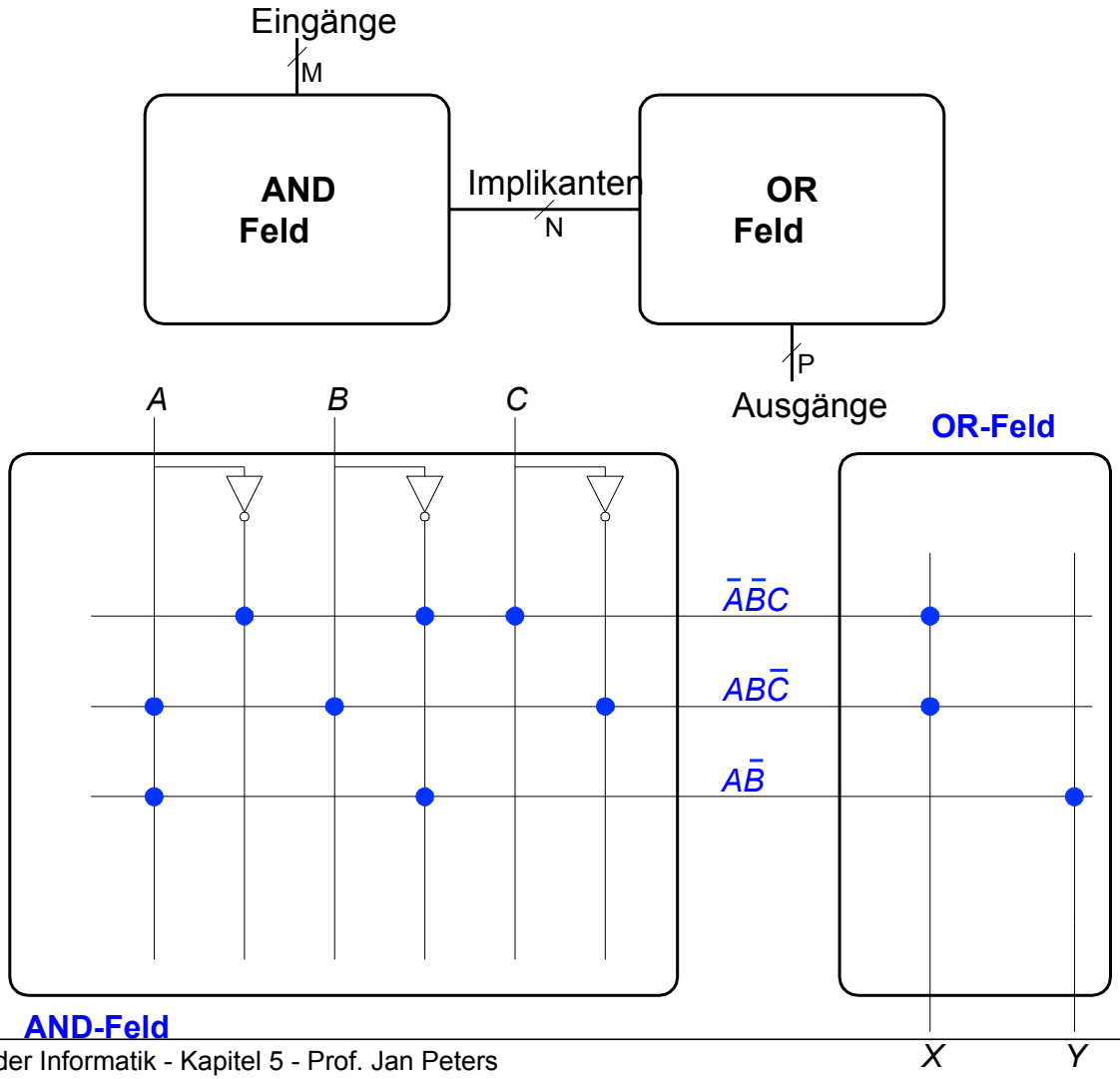
Boole‘sche Funktionen mit PLAs: Idee

- $X = \overline{A}\,\overline{B}C + AB\overline{C}$
- $Y = A\overline{B}$





PLAs: Vereinfachte Schreibweise



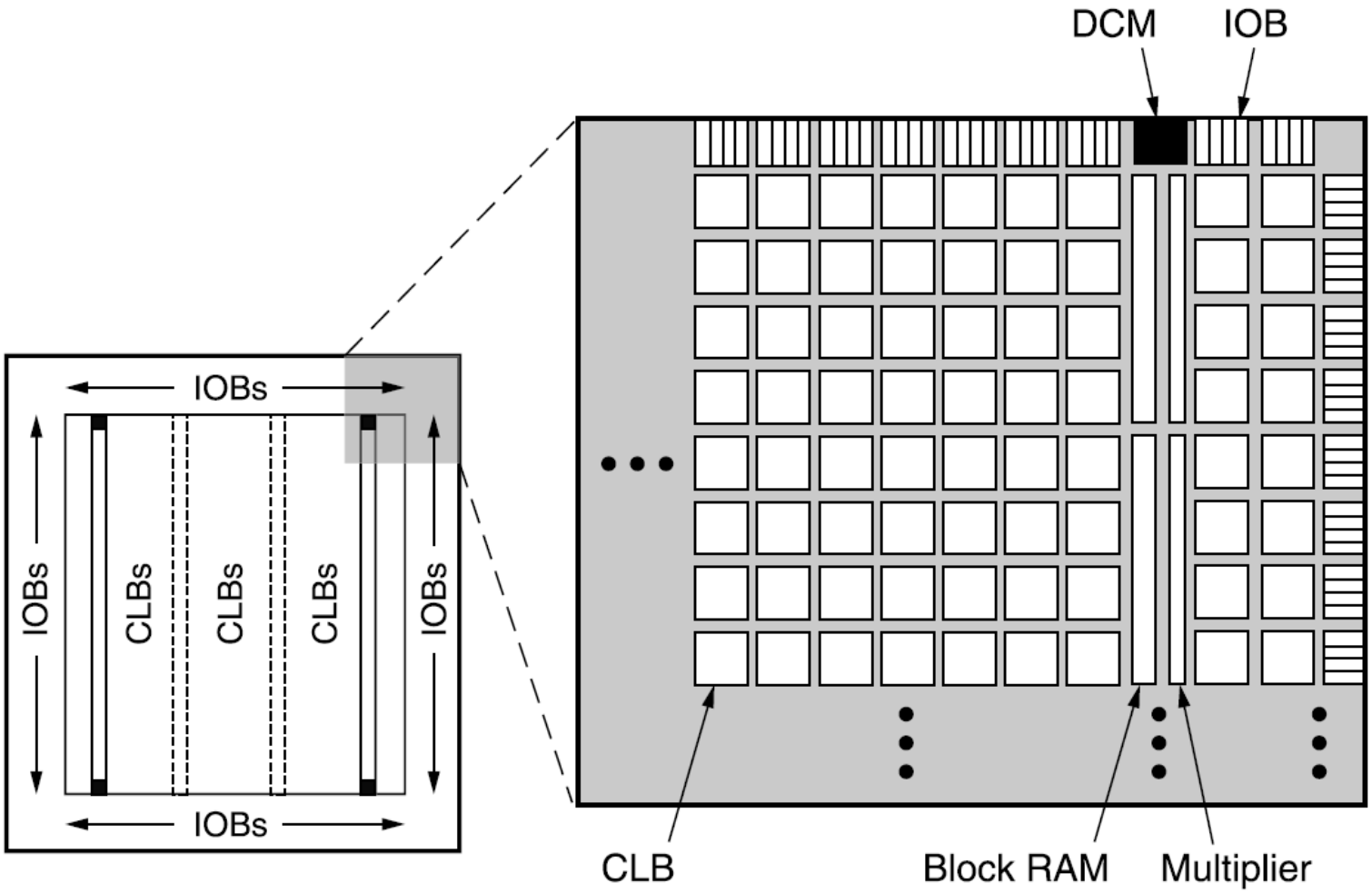
FPGAs: Field Programmable Gate Arrays

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bestehen grundsätzlich aus:
  - CLBs (Configurable Logic Blocks): Realisieren kombinatorische und sequentielle Logik
    - Konfigurierbare Logikblöcke
  - IOBs (Input/Output Blocks): Schnittstelle vom Chip zur Außenwelt
    - Ein-/Ausgabeblöcke
  - Programmierbares Verbindungsnetz: verbindet CLBs und IOBs
    - Kann flexibel Verbindungen je nach Bedarf der aktuellen Schaltung herstellen

  - Reale FPGAs enthalten oftmals noch weitere Arten von Blöcken
    - RAM
    - Multiplizierer
    - Manipulation von Taktsignalen (DCM)
    - Sehr schnelle serielle Verbindungen (11 Gb/s)
    - Komplette Mikroprozessoren
    - …

Struktur eines Xilinx Spartan 3 FPGA

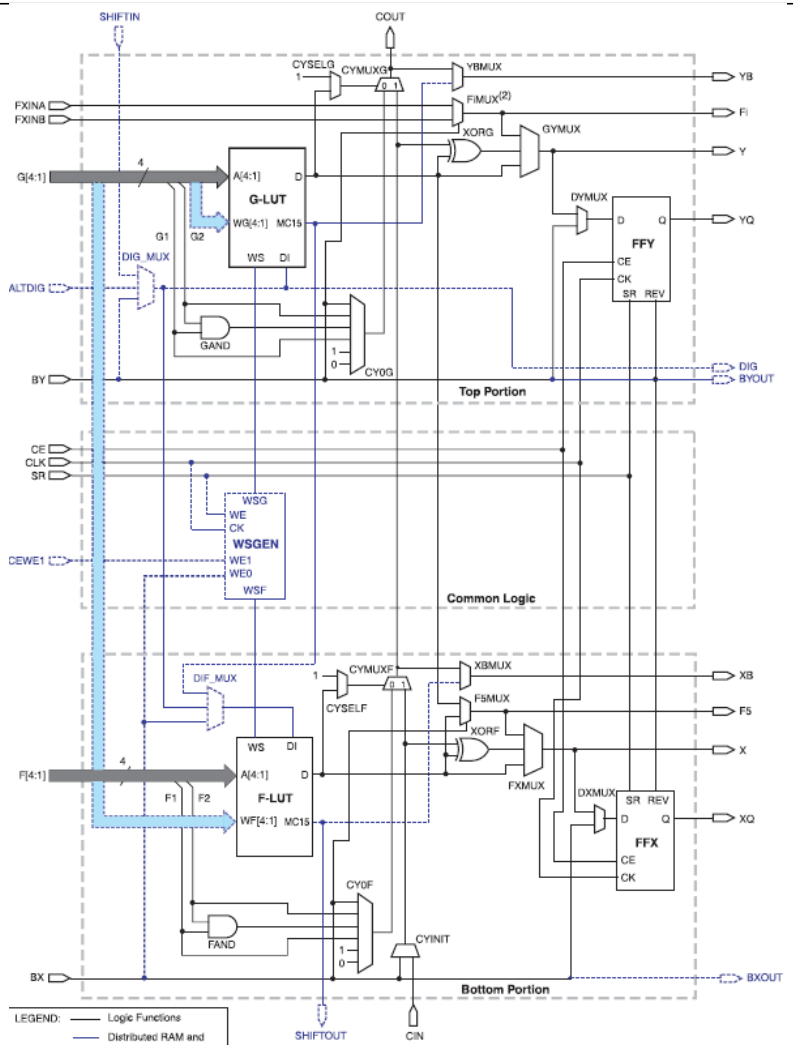


Konfigurierbare Logikblöcke (CLBs)

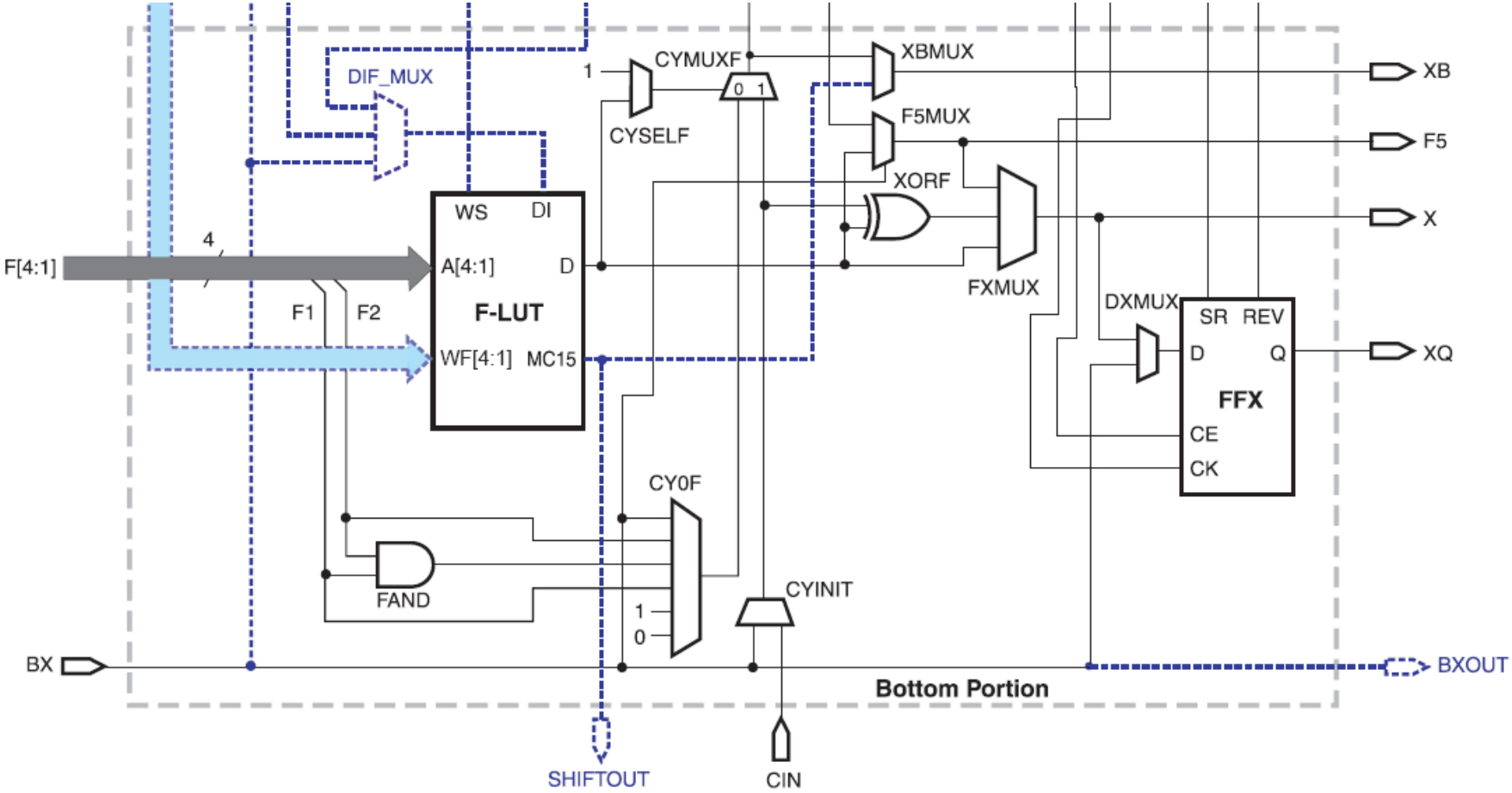
TECHNISCHE UNIVERSITÄT DARMSTADT

- Bestehen im wesentlichen aus:
  - LUTs (lookup tables): realisieren kombinatorische Funktionen
  - Flip-Flops: realisieren sequentielle Funktionen
  - Multiplexern: Verbinden LUTs und Flip-Flops

Xilinx Spartan 3 CLB



Xilinx Spartan CLB



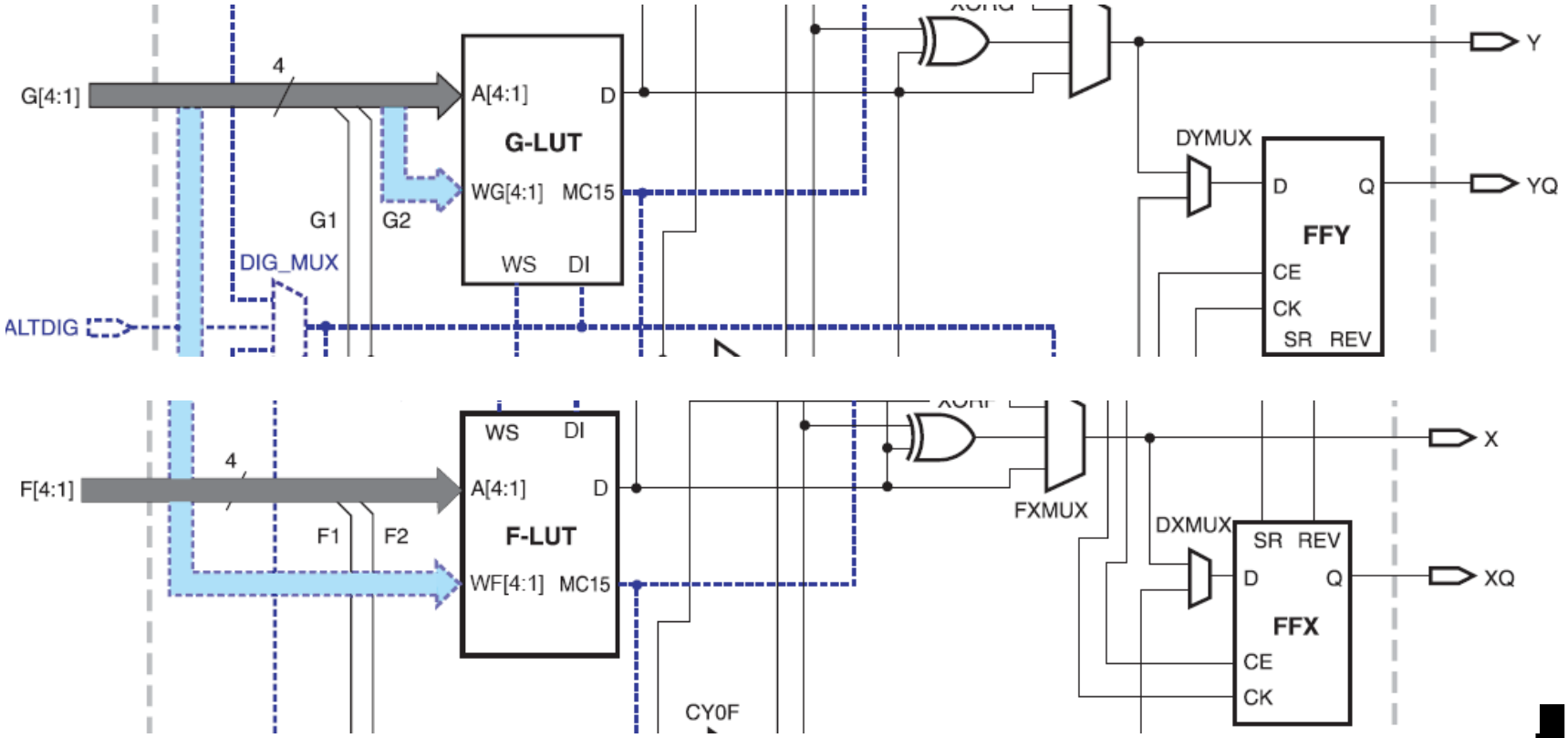
Xilinx Spartan 3 CLB

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ein Spartan 3 CLB enthält:
  - 2 LUTs:
    - F-LUT (2^4 x 1-bit LUT)
    - G-LUT (2^4 x 1-bit LUT)
  - 2 sequentielle Ausgänge:
    - *XQ*
    - *YQ*
  - 2 kombinatorische Ausgänge:
    - *X*
    - *Y*

Beispiel: Kombinatorische Logik mit CLBs

- Berechnung der folgenden Funktionen mit dem Spartan 3 CLB
  - $X = \overline{A}\,\overline{B}C + AB\overline{C}$
  - $Y = A\overline{B}$



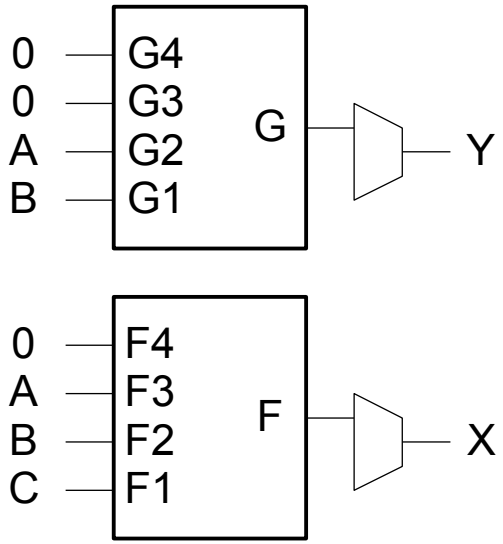
Beispiel: Kombinatorische Logik mit CLBs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Berechnung der folgenden Funktionen mit dem Spartan 3 CLB
  - $X = \overline{A}\,\overline{B}C + AB\overline{C}$
  - $Y = A\overline{B}$

F4	(A) F3	(B) F2	(C) F1	(X) F
X	0	0	0	0
X	0	0	1	1
X	0	1	0	0
X	0	1	1	0
X	1	0	0	0
X	1	0	1	0
X	1	1	0	1
x	1	1	1	0

G4	G3	(A) G2	(B) G1	(Y) G
X	X	0	0	0
X	X	0	1	0
X	X	1	0	1
X	X	1	1	0



Entwurfsfluß für FPGAs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wird in der Regel durch Entwurfswerkzeuge unterstützt
  - Beispiel: Xilinx ISE
- Ist in der Regel ein iterativer Prozess
  - Planen
  - Implementieren
  - Simulieren
  - Wiederhole …
- Entwickler denkt nach
- Entwickler gibt Entwurf als Schaltplan oder HDL-Beschreibung ein
- Entwickler wertet Simulationergebnisse aus
- Wenn Simulation zufriedenstellend: Synthetisiere Entwurf in Netzliste
- Bilde Netzliste auf FPGA-Konfiguration ab (CLBs, IOBs, Verbindungsnetz)
- Lade Konfigurationsdaten (*bit stream*) auf FPGA
- Teste Schaltung nun in realer Hardware