Implementation eines RISC-Microprozessors auf Spartan-3 FPGAs

Marcel Köhler

s0241122@mail.inf.tu-dresden.de

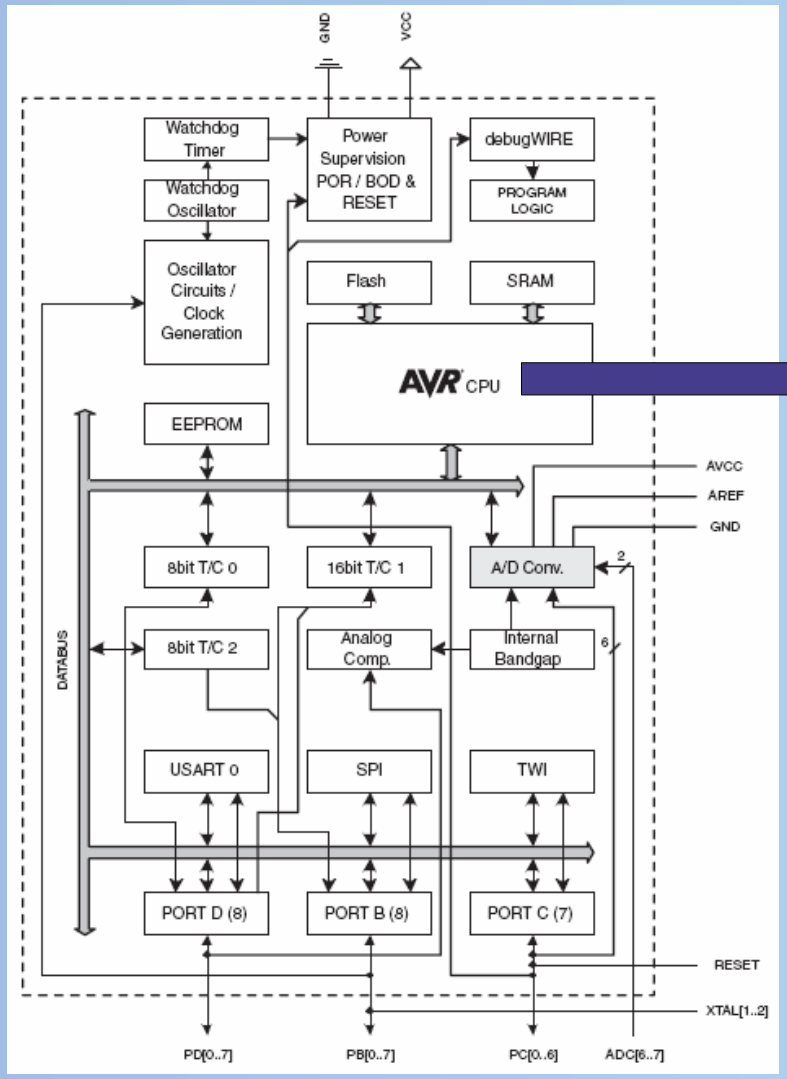
INHALT

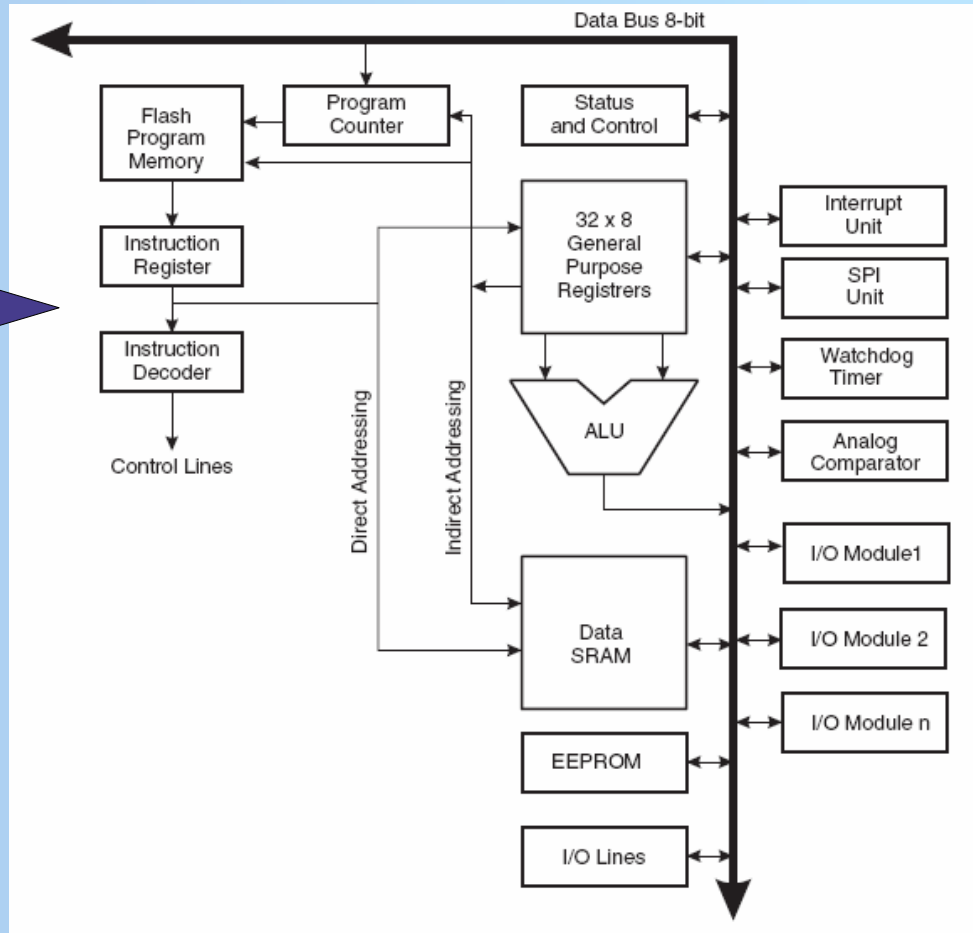
- Motivation
- Übersicht Atmel AVR Architektur
- Topzelle
- Entwurf von Datenpfad, Steuerwerk, DCM
- Performance und Flächenverbrauch
- Ausblick/ Wertung

Motivation

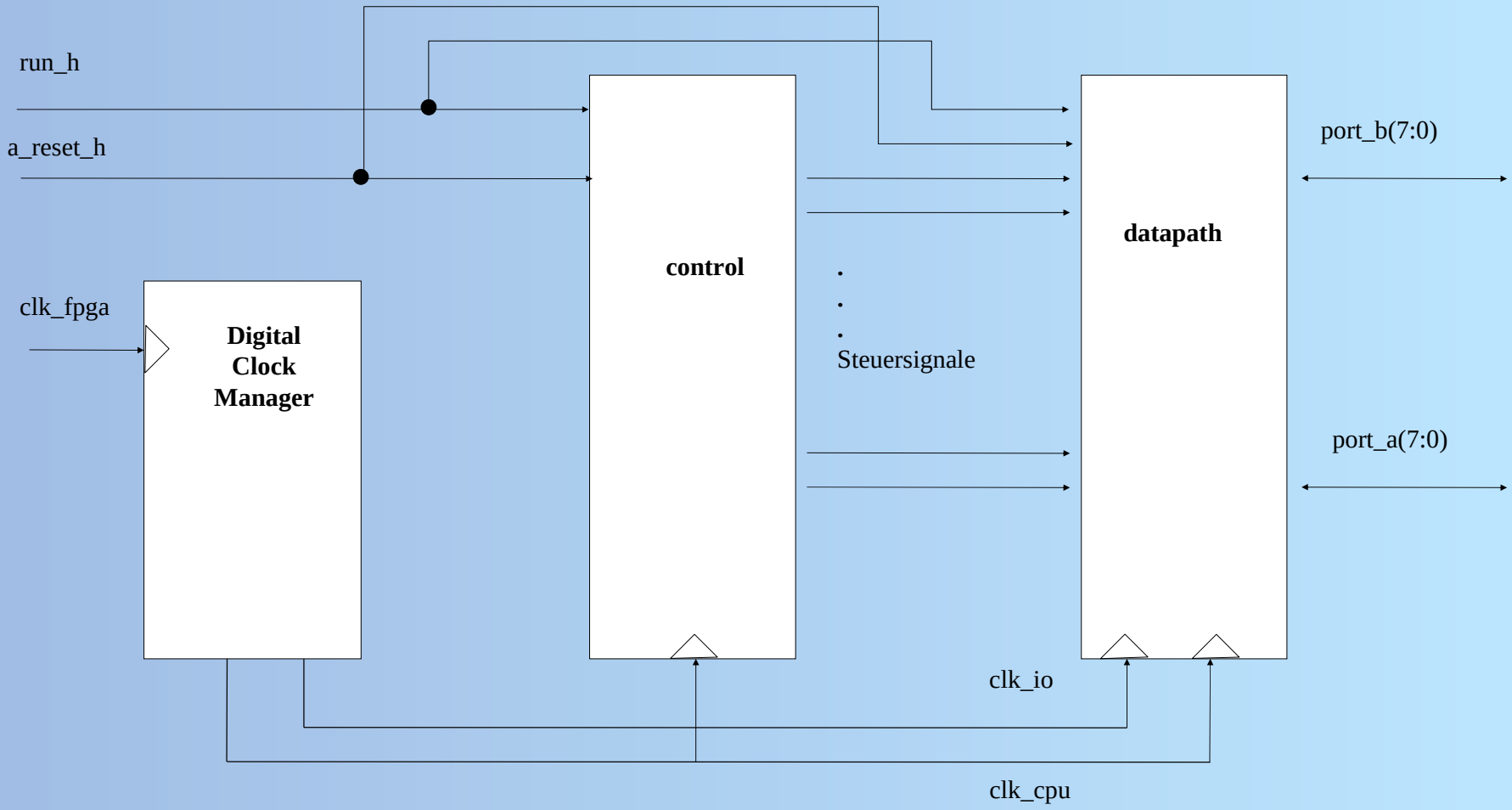
- Kenntnisse über Aufbau und Funktion eines µP vertiefen
- Beschreibung von Hardware-Komponenten mit VHDL
- „Rapid Prototyping“:
  - Zuerst Entwicklung eines FPGA basierten Prototypen
  - Überleitung in einen ASIC, µP …
  - Bei großen Stückzahlen interessant
  - Schaltungsfunktion ist prinzipiell verifiziert

Blockschaltbild des AVR





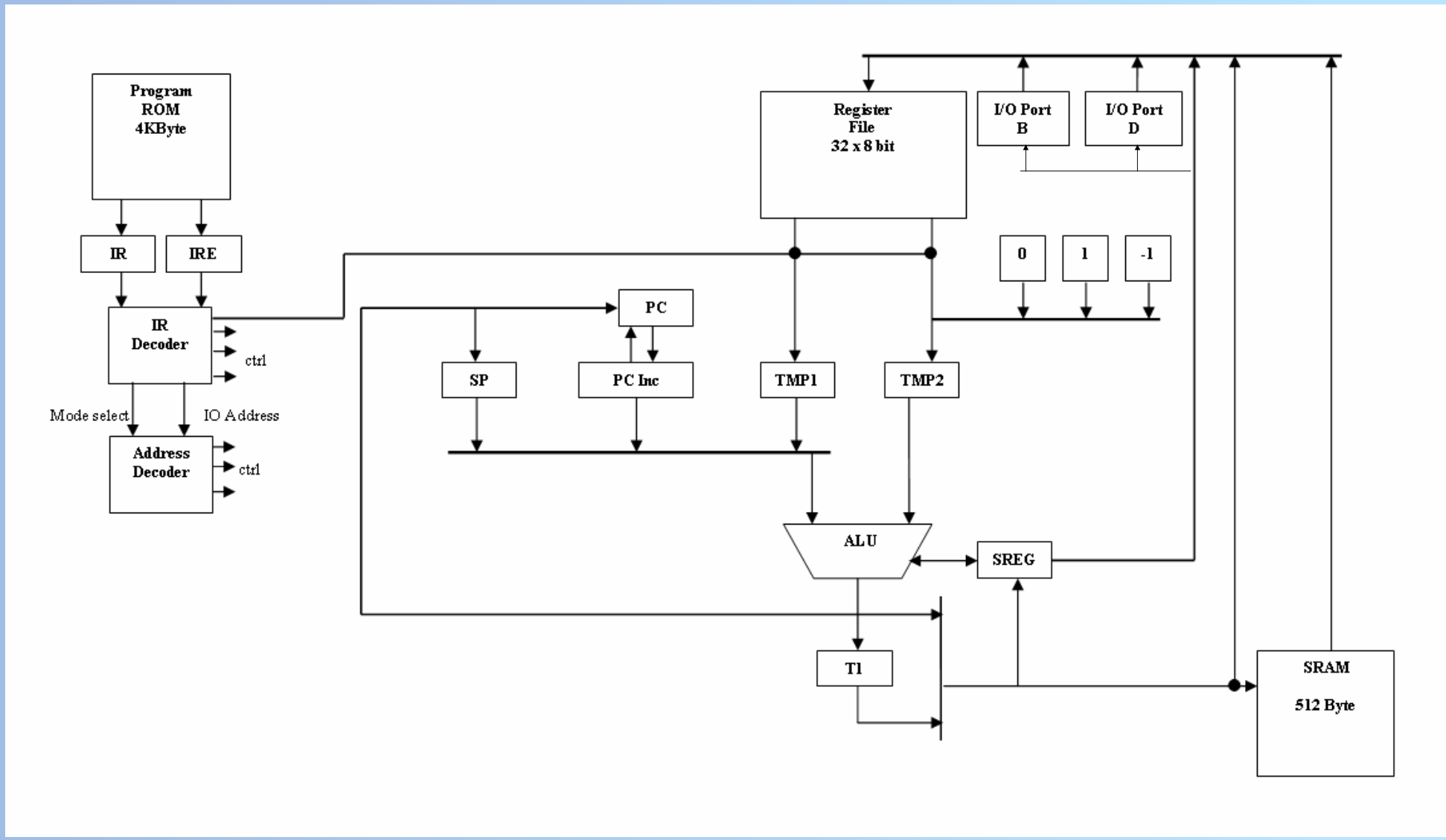
Blockschaltbild Topzelle



Datenpfad

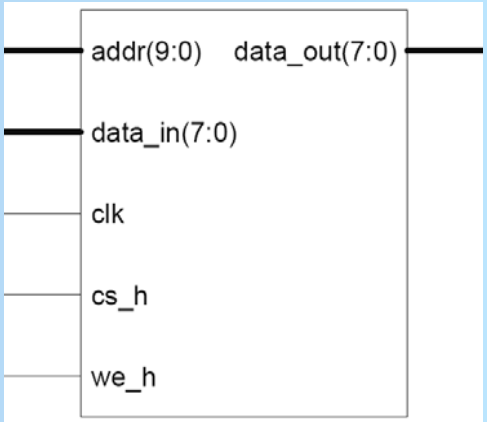
- Speicherrealisierung
- General Purpose Registerfile
- ALU
- Datenregister
- Befehlsdekoder
- Adressdekoder
- IO Port

Datenpfad


Program ROM 4KByte
IR
IRE
IR Decoder
ctrl
Mode select
IO Address
Address Decoder
ctrl
Register File 32 x 8 bit
I/O Port B
I/O Port D
0
1
-1
PC
SP
PC Inc
TMP1
TMP2
ALU
SREG
T1
SRAM 512 Byte

Datenpfad/ Speicherrealisierung

- Programmable ROM
  - Implementation als Block-RAM auf FPGA
  - Größe: 4 KByte
  - Enthält Befehlscode
  - Wird während Synthese über ein textfile initialisiert
- SRAM
  - Implementation als Block-RAM auf FPGA
  - 512 Byte Speicherkapazität



Datenpfad/ General Purpose Register

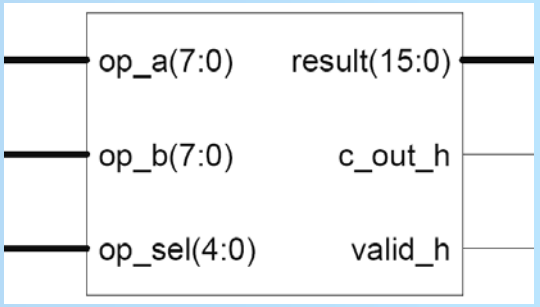
- Implementation als Dual-Port distributed RAM
- Spezieller RAM, aus LUT‘s aufgebaut
- Größe: 32 Register à 8 Bit
- Asynchrones Auslesen einer Adresse möglich
- Gleichzeitiger Lesezugriff auf 2 Register
- Einsparung eines Taktzyklus im Entwurf

- Addierer und Subtrahierer
  - Addieren mit/ ohne Carry
  - Subrahieren mit/ ohne Carry
  - Subtraktion entspricht Addition mit 2-Komplement (Carry-In =1)
- Multiplikation
  - Multiplikation von Zahlen im Format:

    Fractional

    Signed

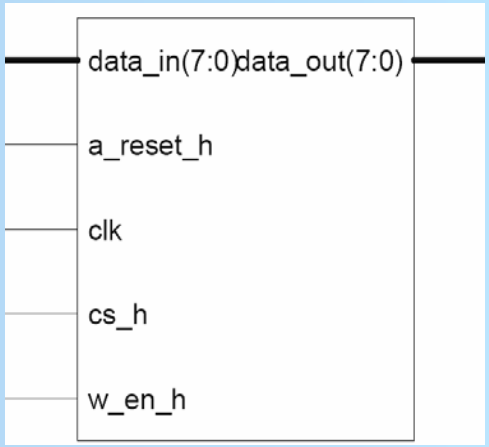
    Unsigned



  - Umsetzung im 18 x 18 bit Hardwaremultiplizer des FPGA
- Bit-/ Logikoperationen

Datenpfad/ Datenregister

- Definition eines allg. 8 Bit Registers
- Instanziierung:
  - ProgramCounter
  - StackPointer
  - Konstantenregister mit Werten -1,0,1
  - Statusregister SREG
  - Temporäre Hilfsregister



Befehlsdekoder und Adressdecoder

- Befehlsdecoder
  - komplexe kombinatorische Logik
  - Dekodierung der Operation + Operanden aus 16-Bit Befehl

- Adressdekoder
  - Mögliche Adressierung:
  - direkte Adressierung → direkte Adresse ist Teil des Opcodes (32-bit Opcode)

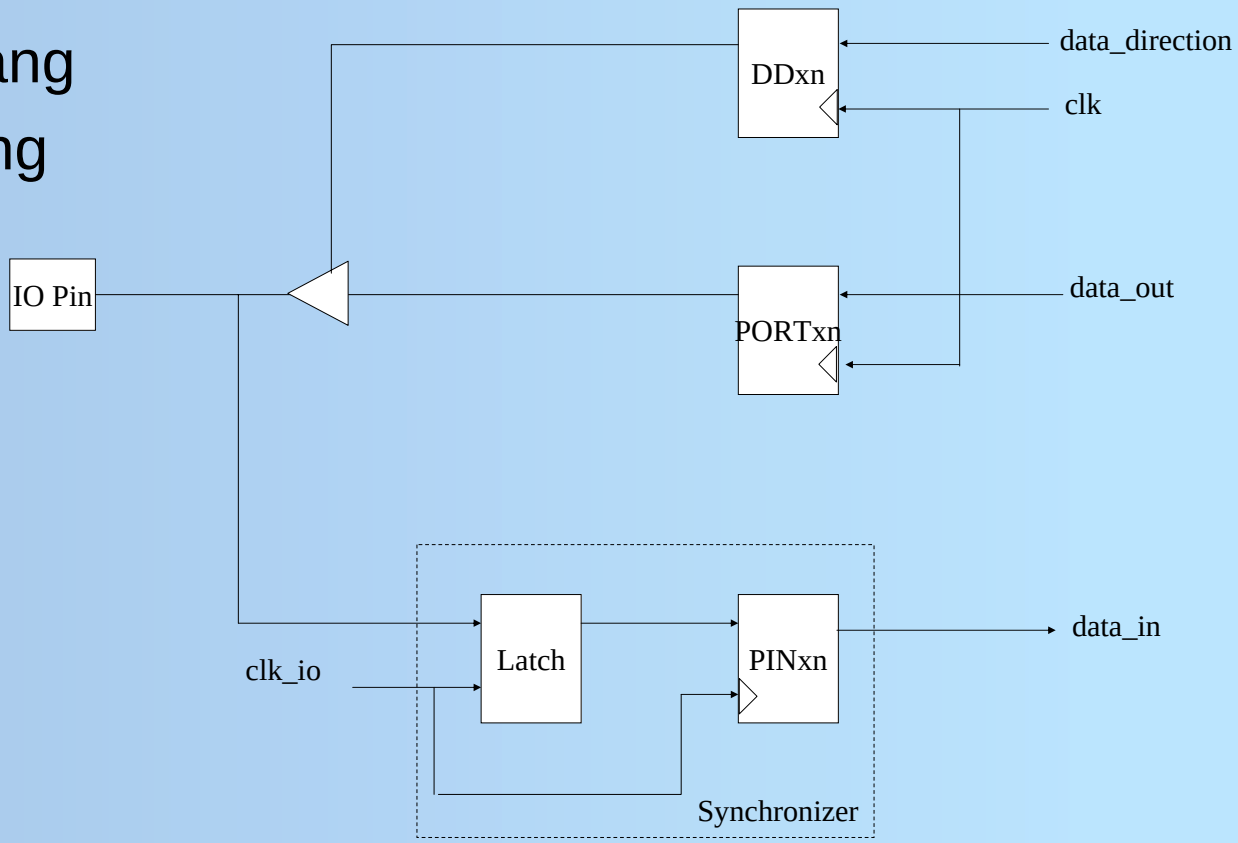
    indirekte Adressierung durch X, Y, Z → Adresse wird durch Indexregister bestimmt

    indirekte Adressierung durch SP → SP beinhaltet Adresse

    Adressierung von IO-Registern

Datenpfad/ Aufbau IO Port

- IO Port stark vereinfacht gg. Spezifikation
  - Sleep-Mode nicht berücksichtigt
  - DDxn für output enable
  - Portxn Datenausgang
  - PINxn Dateneingang

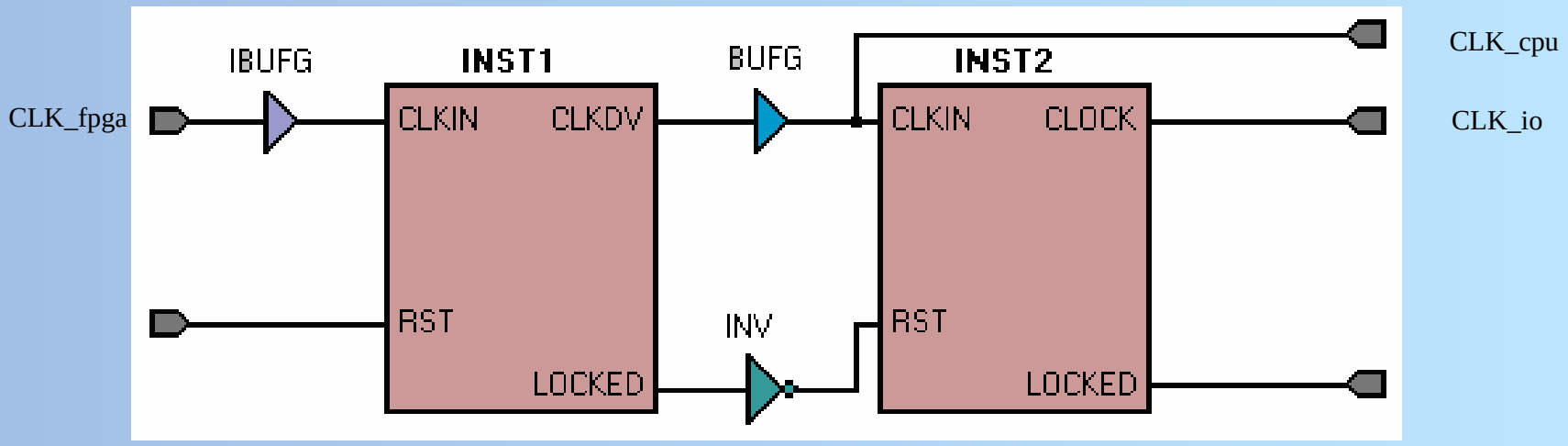


Steuerwerk

- Im Steuerwerk FSM integriert
  - 50 Zustände
- Erhält auszuführende Instruktion vom Befehlsdekoder
  - Zustand bestimmt Ausgabe der Steuersignale für Datenpfad
  - Bedingte Zustandsübergänge durch Statusregister-Flags

Digital Clock Manager

- Kaskadenschaltung mit IP-Core Generator
- 50 MHz FPGA Takt auf 12.5 MHz geteilt
- 12.5 auf 4 MHz geteilt
- I/O-Pad enthält Synchronizer für Datenregister → clk_io



Performance und Flächenverbrauch

- Optimierung auf Chipfläche bei Synthese
- Finale Ressourcenauslastung:
  - Slices: 597 (31%)
  - Slices FF‘s: 259 (6%)
  - LUT‘s: 1043 (27%)
- Hohe Komplexität an kombinatorischer Logik für Steuerwerk und Befehlsdekoder
- Änderung des Zielsystems:
  - größerer HW-Aufwand, wegen Nutzung HW-spezifischer Elemente (Block RAM, HW-Multiplier)
- Getestet mit Speed: 4 MHz
- Theoretisch 30 MHz möglich, Steigerung durch Speedoptimierung bei Synthese

Ausblick/ Wertung

- Verfeinerung durch weitere Peripherie
  - Watchdog
  - IO Ports
- Nachträgliches Laden des Programmspeichers
  (momentan nur während Synthese möglich)
- Übersichtlichkeit des Datenpfads → Entwurf verbesserungswürdig