TECHNISCHE UNIVERSITÄT DRESDEN

Vortrag zum Beleg

Platzierung und Verdrahtung massiv-paralleler FPGA-Designs am Beispiel eines Many-Core-Prozessors

Michael Lange

Dresden, 16.04.2015

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Gliederung

1 Aufgabenstellung

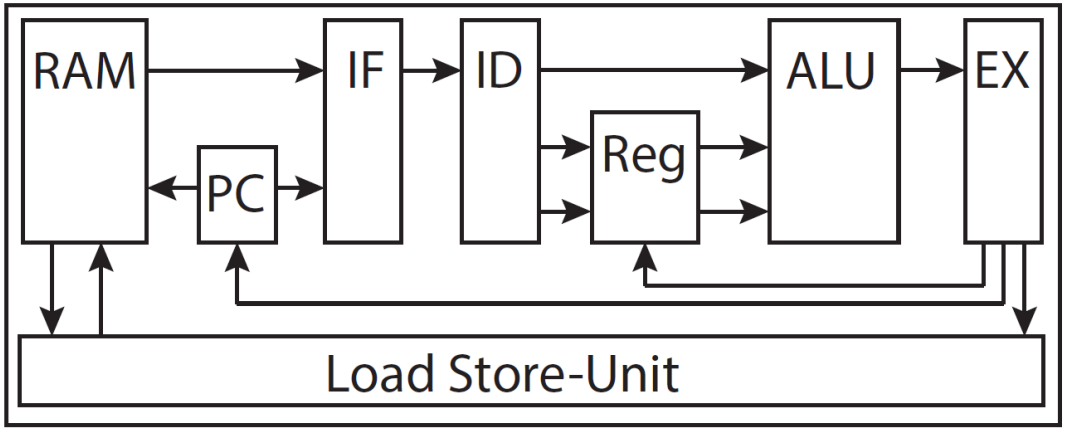
1. Literaturstudium zur geführten Platzierung und Verdrahtung auf FPGAs
2. Auswahl und Analyse von 3 parallelen Algorithmen, die verschiedene Topologien erfordern
3. Implementierung und Test des parallelen Systems für jeden Algorithmus
4. Räumliche Skalierung des jeweiligen Systems unter verschiedenen Vorgaben aus Geometriesicht unter Berücksichtigung der FPGA-Architektur
5. Bewertung der erzielten Skalierbarkeit, Zusammenfassung und Dokumentation der Ergebnisse

2 Voraussetzungen

fertig entwickelter MIPS-Kern auf Basis des MIPS R2000 in VHDL aus dem Komplexpraktikum:

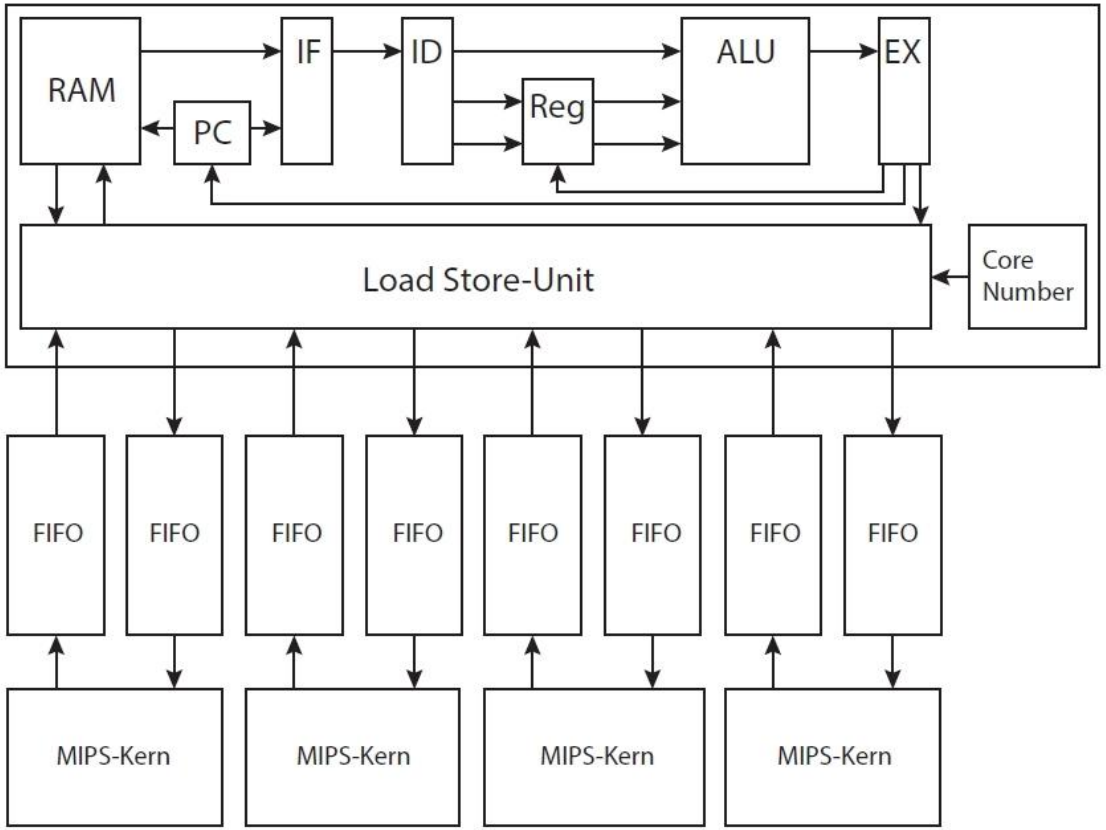
- MIPS-Befehlssatz (RISC, 32 Bit)
- 3-stufige Pipeline
- Dual-Port-RAM, Havard-Architektur

GCC-Compiler mit MIPS-Backend, um C-Code in MIPS-Befehle zu übersetzen



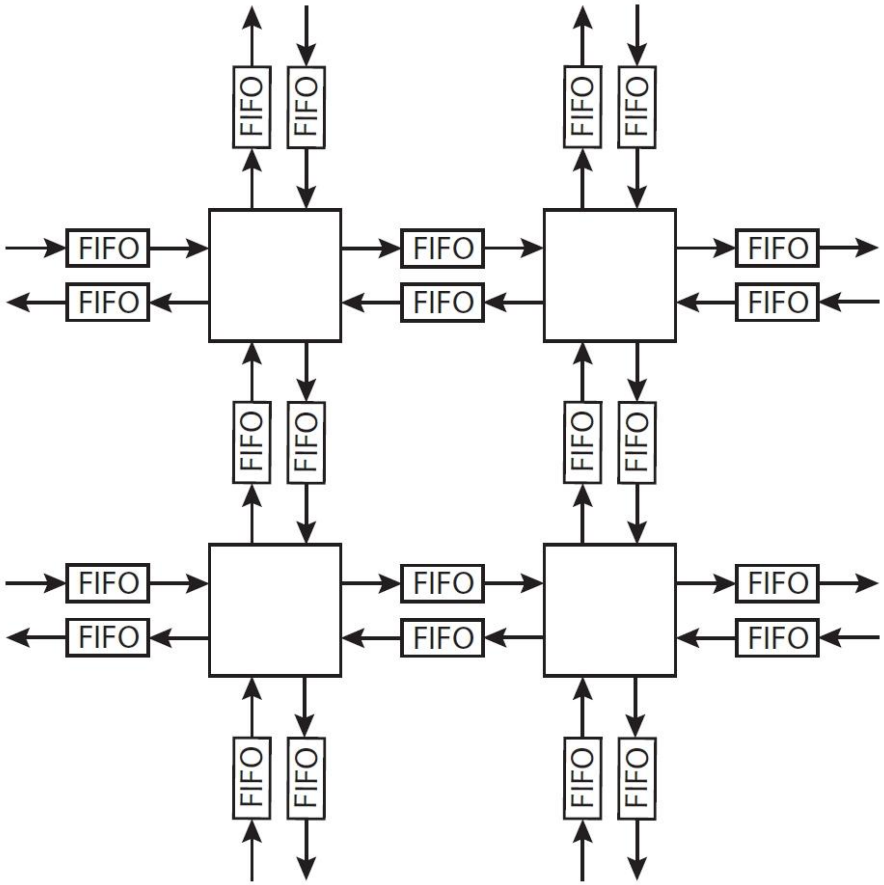
Anpassung des Prozessors

- Hinzufügen von 4 Input- und 4 Output-FIFOs an die Load Store-Einheit
- Memory Mapped I/O für die Core-Nummer und die din-, dout-, empty- und full-Signale der FIFOs



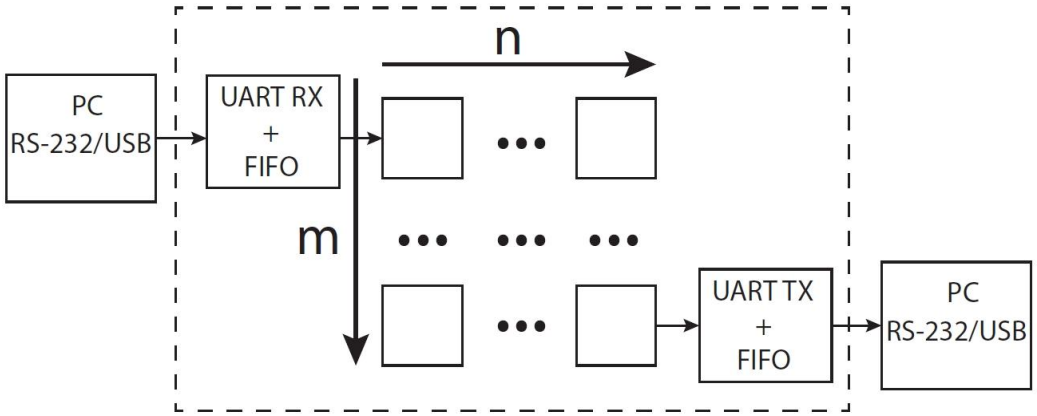
3 Aufbau eines Many-Cores

- Verbindung der Nachbar-Cores mittels FIFOs
- FIFO-Tiefe von 4
- FIFOs gelesen bzw. FIFOs beschrieben nur, wenn sie nicht leer bzw. nicht voll sind
  -> Steuerung durch while-Schleife im C-Code



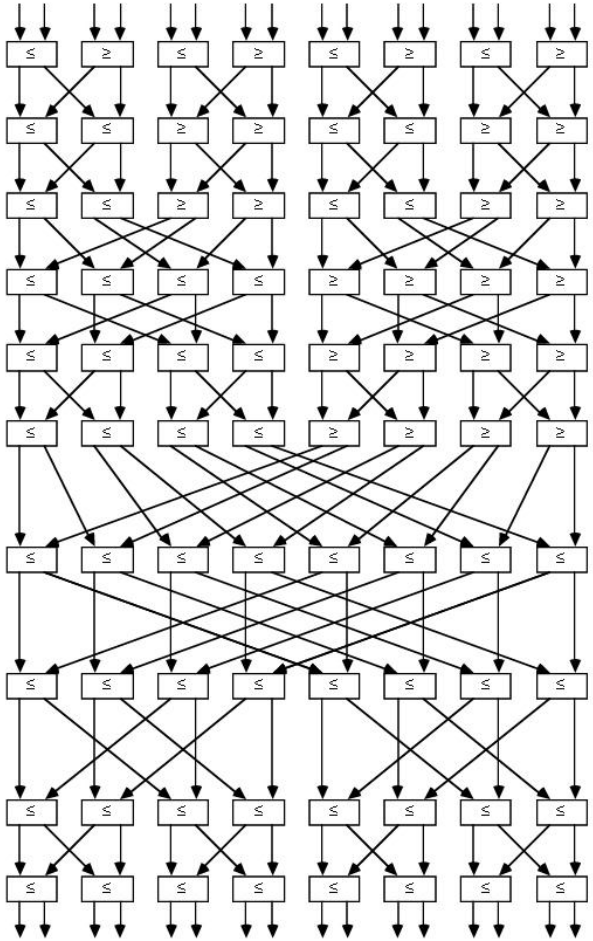
3 Aufbau eines Many-Cores

- Verbindung zwischen PC und FPGA über RS-232 bzw. USB
- RX-Modul der UART erhält Daten, TX-Modul schreibt diese heraus
- RX und TX besitzen große FIFOs, um Eingangs- und Ausgangsdaten zu puffern



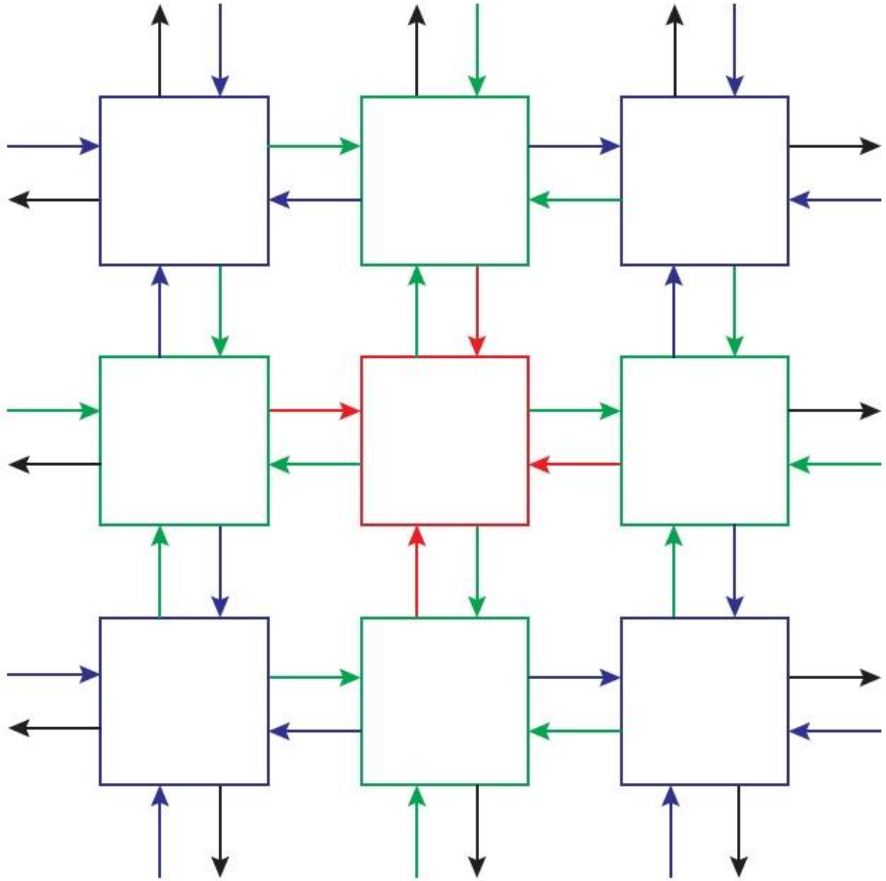
4 Algorithmen – Bitonisches Sortieren

- Folge von Zahlen wird systematisch in immer größere bitonische Folgen sortiert
- jede Stufe wird parallel ausgeführt
- Folge heißt bitonisch, wenn der erste Teil der Folge aufsteigend und der zweite Teil absteigend sortiert ist



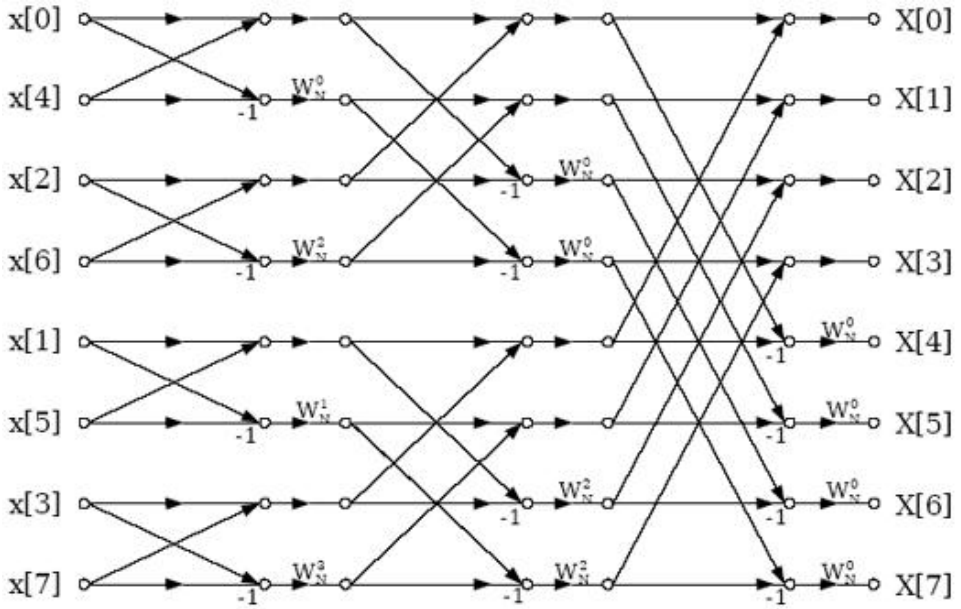
4 Algorithmen – 5-Punkt Stencil-Code

- Berechnungsergebnis eines Kerns von den vier Nachbarn abhängig
- für verschiedenste Algorithmen:
  - partielle Differenzgleichungen
  - lineare Gleichungssysteme
  - Bildverarbeitung
  - Strömungsalgorithmen
  - zellulare Automaten



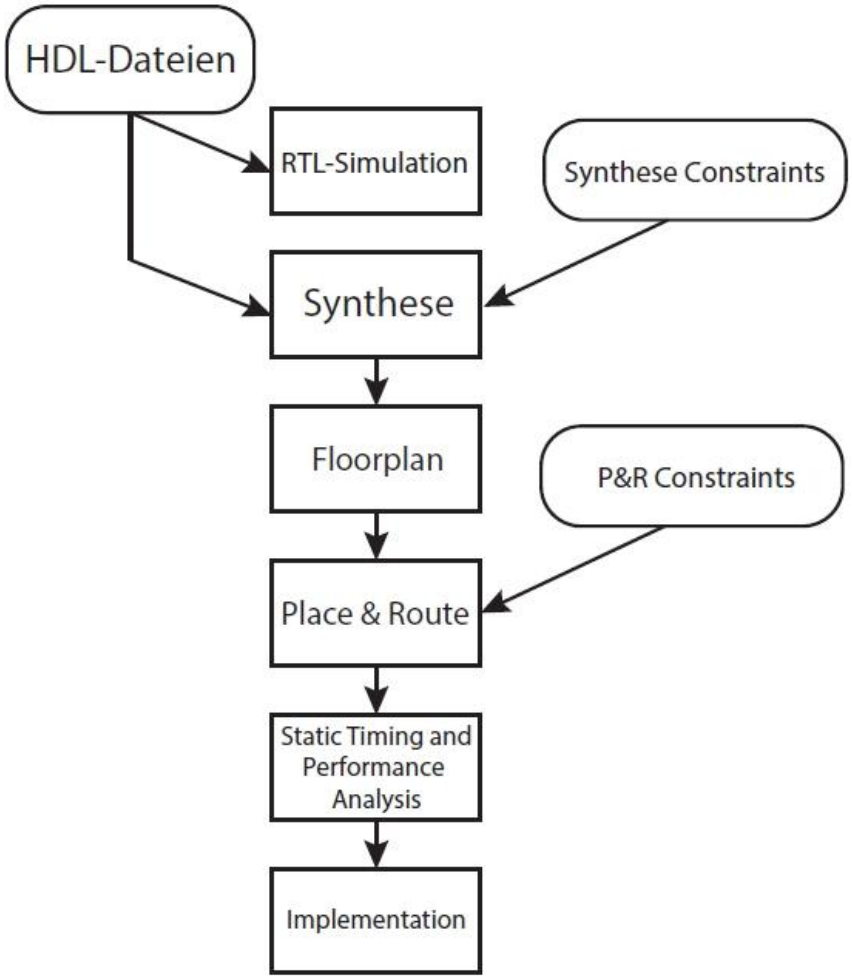
4 Algorithmen – FFT

- parallele Berechnung der Fourier-Transformation mit Hilfe 2^n Samples
- Signal im Zeitbereich wird in den Frequenzbereich überführt
- Angewandt für:
  - Datenkomprimierung
  - Mobilfunk-Datenübertragung



FPGA-Design Flow

- Synthese-Ergebnisse nur grobe Richtwerte
- erst im Place & Route wird das Design auf die Ziel-FPGA gemapped
- danach sind die erreichten Parameter genau und eventuelle Timing-Fehler sichtbar
- P&R-Ergebnisse lassen sich mit Hilfe der Xilinx-Tools manuell verbessern



5 User Constraints für Place & Route

- beeinflusst Abbildung des Designs auf der FPGA
- typische Constraints sind:
  - Platzierungen

    INST “my_ram“ LOC = RAMB36_X0Y0;

    INST “my_mips“ RLOC = X0Y6;

    AREA_GROUP “my_mips“ RANGE = SLICE_X0Y325:SLICE_X37Y349
  - Pin-Zuweisungen

    NET “clk“ LOC = P38;
  - Takt-Spezifikation

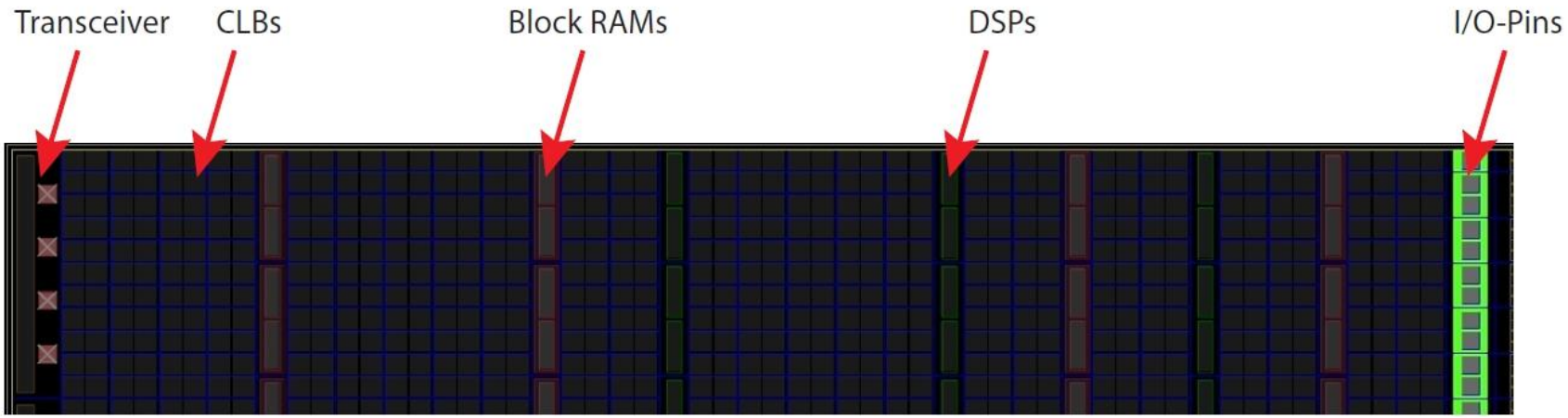
    NET “clk“ TMN_NET = “clk“;

    TIMESPEC TS_clk = PERIOD “clk“ 10 ns HIGH 50%;

5 User Constraints-Dateien

- Altera:
  - .qsf-Datei (Quartus Settings File) für Placement-Constraints
  - .sdc-Datei (Synopsys Design Constraints) für Timing-Constraints
  - manuell oder per GUI eingebbar:
    - Quartus II (Placement-Constraints)
    - Quartus TimeQuest Timing Analyzer (Timing-Constraints)
- Xilinx:
  - .xcf-Datei (XST Constraint File) für Synthese-Constraints
  - .ucf-Datei (User Constraints File) für Place & Route-Constraints
  - Manuell oder per GUI eingebbar
    - Xilinx ISE (Timing-Constraints)
    - Xilinx PlanAhead (Placement-Constraints)

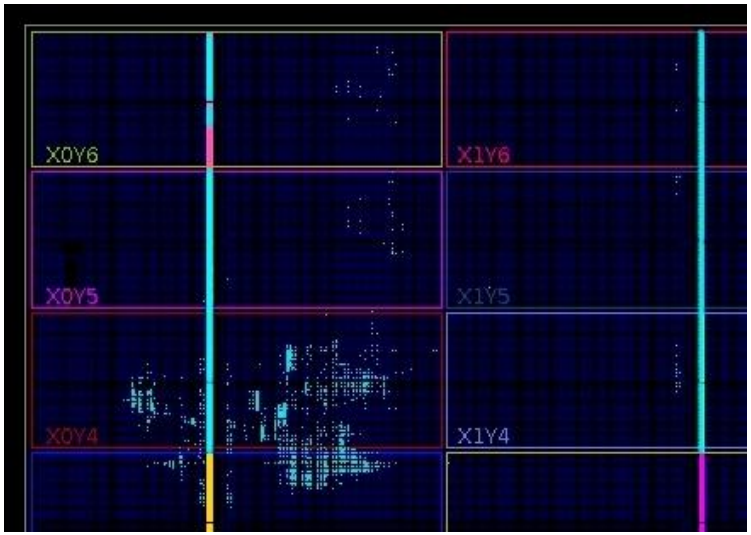
FPGA-Aufbau – am Beispiel des Virtex 7



- FPGA besteht aus mehreren Clock Regions
- X/Y-Matrix aus diesen Elementen, beginnend in der linken unteren Ecke
- außerdem Clock Manager, PCIE-IP-Cores und ADCs vorhanden

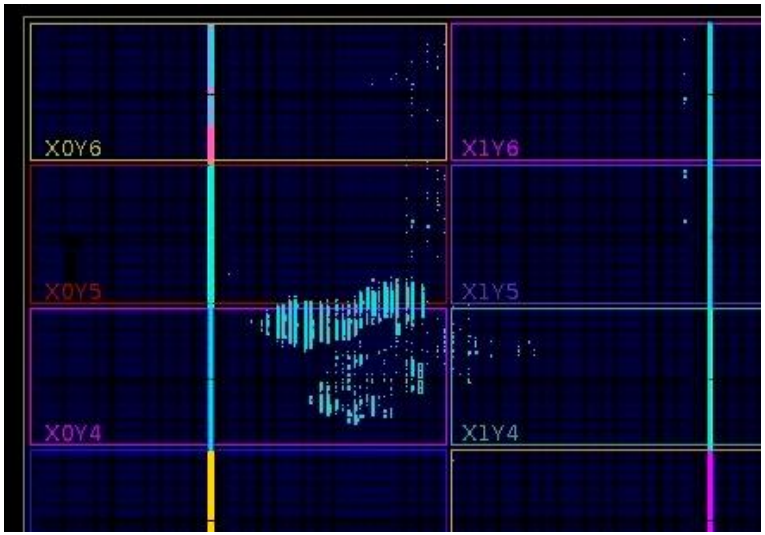
6 Beispiel Place & Route – Timing-Constraint

- Design wird automatisch mittig platziert, um geringe Pfadlänge zu erreichen



automatisches Place & Route ohne Timing Constraint

→ Design wird nahezu willkürlich platziert

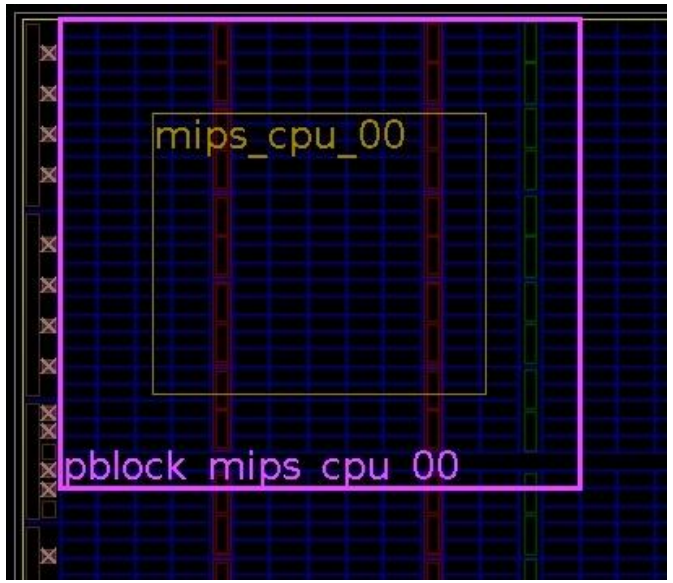


automatisches Place & Route mit scharfem Timing Constraint

→ P&R optimiert enorm auf Zeitverhalten, Pfadlängen geringer

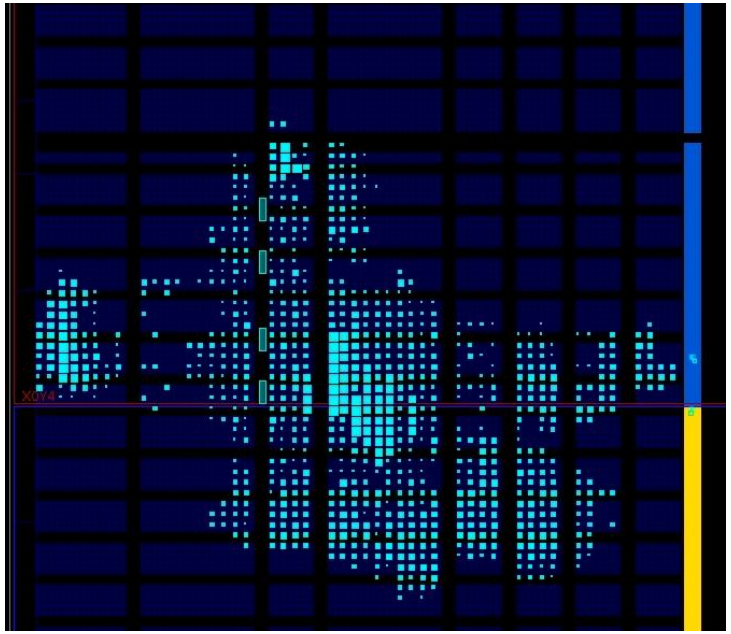
6 Beispiel Place & Route – Placement-Constraint

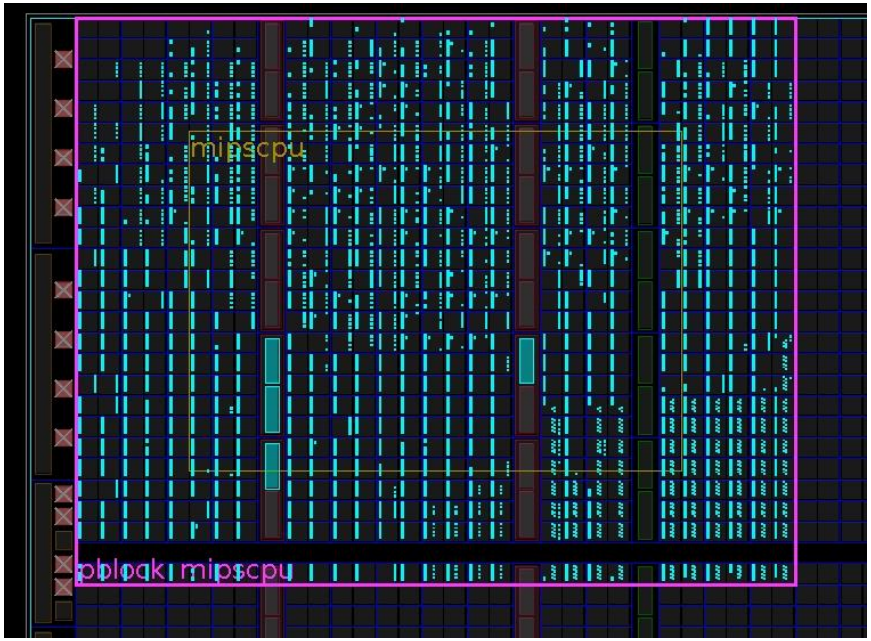
- testweise Platzierung eines Kerns am linken oberen Rand
- die dazugehörigen Constraints werden in die .ucf-Datei geschrieben




```
INST "mipscpu" AREA_GROUP = "pblock_mipscpu";
AREA_GROUP "pblock_mipscpu" RANGE=SLICE_X0Y325:SLICE_X37Y349;
AREA_GROUP "pblock_mipscpu" RANGE=DSP48_X0Y130:DSP48_X1Y139;
AREA_GROUP "pblock_mipscpu" RANGE=RAMB18_X0Y130:RAMB18_X2Y139;
AREA_GROUP "pblock_mipscpu" RANGE=RAMB36_X0Y65:RAMB36_X2Y69;
```

6 Beispiel Place & Route – Placement-Constraint





- ohne Placement-Constraint
- mit Placement-Constraint

Ziel der Arbeit:

- Auswahl einer jeweils idealen Prozessor- und Verbindungs-Struktur für die vorgegebenen Algorithmen
- Erstellung skalierfähiger Designs
- Untersuchung der Granularität bei Placement-Constraints
- Bewertung und eventuelle Verbesserung des Timing und Placement
- Vergleich mit automatischem Place & Route
- Finden allgemeingültiger Aussagen zur Implementierung auf FPGAs hinsichtlich des Placements und des Routings

Literatur

Frank Kesel, Ruben Bartholomä:
Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs - Oldenbourg Verlag 2013

Steve Kilts:
Advanced FPGA Design - John Wiley & Sons 2007

Andreas Schwill:
Bitonisches Sortieren auf Parallelrechnern - Universität Potsdam 1999

Shah M. Faizur Rahman, Quig Yi, Apan Quasem:
Understanding Stencil Code Performance on MultiCore Architectures - University San Antonio 2011

Jan Metzner, Alexander Schaal:
Schnelle Fourier-Transformation – FH München 2002

Xilinx Constraints Guide v13.4
Xilinx PlanAhead Software Tutorial v12.3
Xilinx Timing Constraints User Guide v12.3
Xilinx Floorplanning Methodology Guide v14.1
Xilinx FPGA Editor Guide v2.1