TECHNISCHE UNIVERSITÄT DRESDEN

Analyse verschiedener HLS-Systeme in Hinblick auf ihren Umgang mit der Hochsprachenabstraktion Speicher

Sascha Kath

Dresden, 26.03.2015

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Gliederung

1. Aufgabenstellung
2. HLS-Systeme
    1. LegUP
    2. Vivado HLS
    3. Leap
    4. MyHDL
3. Projektplan

1. AUFGABENSTELLUNG

1. Aufgabenstellung

- Analyse des Umgangs mit der Hochsprachenabstraktion Speicher verschiedener HLS-Systeme
- Benchmark-Funktionen erstellen und implementieren
- Ergebnisse vergleichen

2. HLS-SYSTEME

2.0 HLS-Systeme: Ein Überblick

- high-level-language → intermediate representation → HDL
- Nutze Synthese-Tools (bspw. der FPGA-Hersteller) für die weiteren Schritte

2.1 LegUp

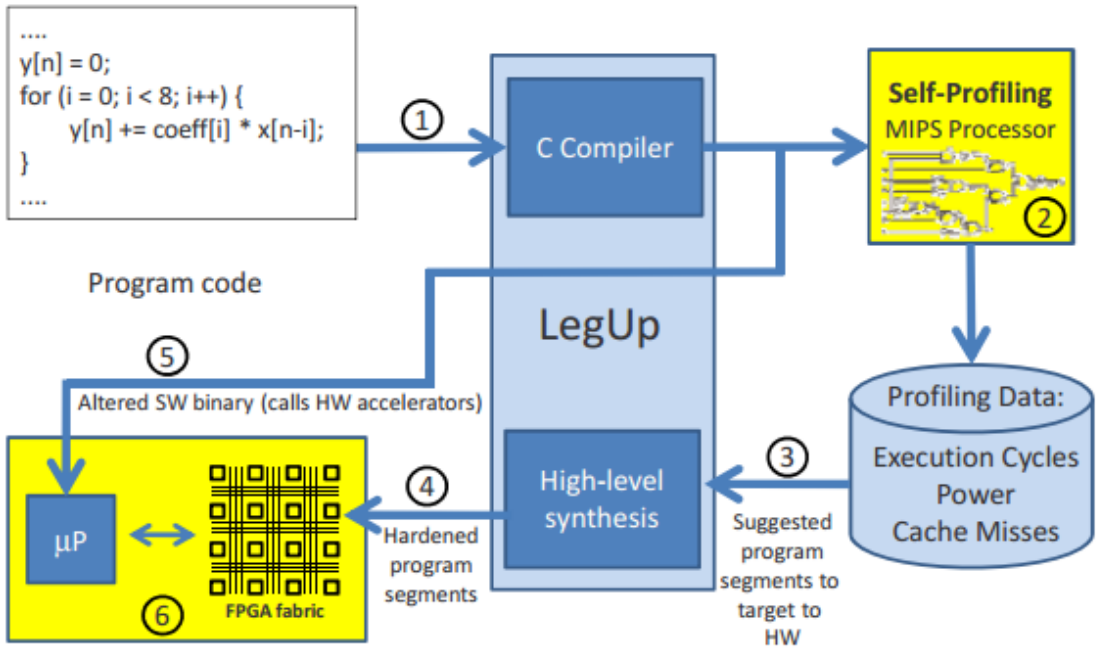


Figure 1: Design flow with LegUp.

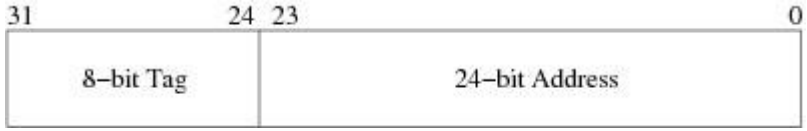
Ziel: vollautomatischer Fluss → selbst-beschleunigender adaptiver Prozessor

2.1 LegUp – vom Code zur Hardware

1. C-Code
2. LLVM als IR
   - LegUp erstellt Ablaufplan → weist Befehle Taktzyklen zu (ASAP scheduling)
3. Verilog
4. Synthese-Werkzeuge

2.1 LegUp - Speicherarchitektur

- LLVM-Speicherarten: stack, globals und heap
- Jede Variable wird in separatem Altsyncram gespeichert → Zugriff über Speichermanager
- 32-bit Adressformat:



- Reservierte Tags:
  - 0x0 → Null-Pointer
  - 0x1 → Prozessorspeicher

2.2 Vivado HLS

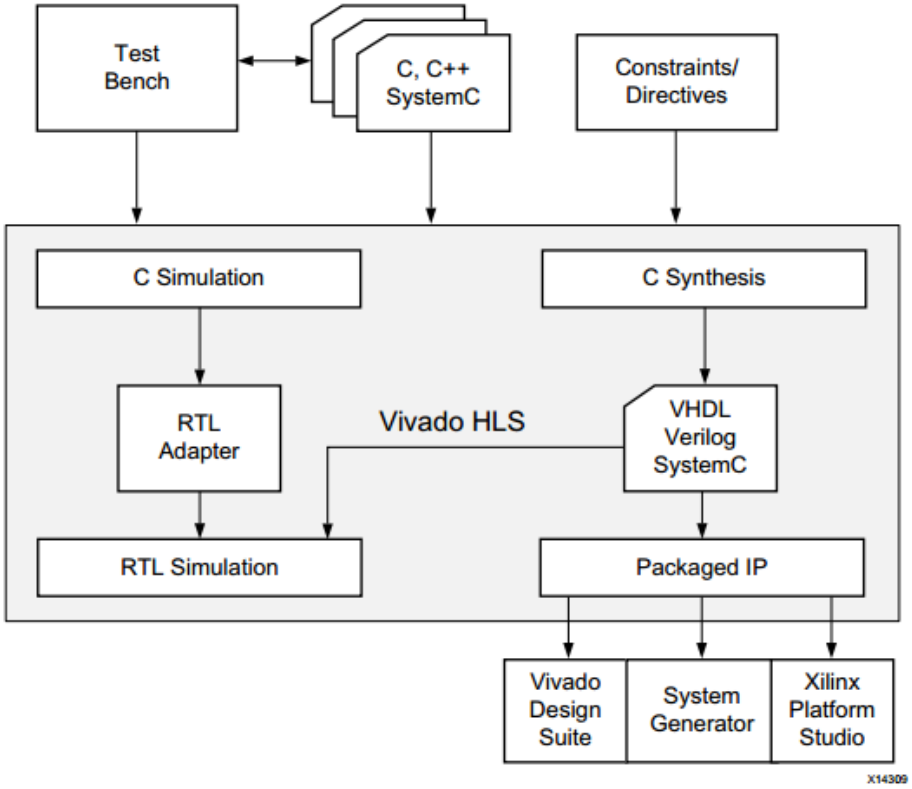


Figure 1-4: **Vivado HLS Design Flow**

2.3 Leap

- LEAP = LINC-based Environment for Application Programming
- Eingabesprache: Bluespec System Verilog
- Grundlegende Konzepte:
  1. Flexibles Intermodul-Kommunikations-Paradigma
     - Programm = Kollektion von latency-insensitive Modulen
     - Intermodul-Kommunikation über latency-insensitive Channels (FIFOs)
  2. Allgemeines Speicher-Paradigma
     - Speicher willkürlicher Größe (private oder shared)

2.3 Leap - Scratchpads

- Partitionierung des on-board RAMs in individuelle Scratchpads (private memory)
- Scratchpad-Controller weist Scratchpads den Speicherbereichen mittels Indirektionstabelle zu
  - Konkatenierung von client ID und Adressanfrage
- Scratchpads haben gleiches Interface wie Speicher auf dem FPGA → nur andere Speichermodule nötig

2.4 MyHDL

- Grundlegende Konzepte:
  1. Generators, um Nebenläufigkeit zu modelieren
  2. Yielded-Werte = Bedingungen auf die Generator wartet bevor er fortfährt. Im Grunde funktionieren Yield-Statements wie Sensitivity-Lists
- Generators sind fortsetzbare Funktionen
  - Ähneln always Blöcken (Verilog) und Prozessen (VHDL)

2.4 MyHDL

- Generators werden zu Verilog- oder VHDL-Konstrukten umgewandelt
- Verilog:
  - Umwandlung in always Blöcke, assignments oder initial Blöcke
- VHDL:
  - Umwandlung in process oder concurrent signal assignments

3. PROJEKTPLAN

3.1 Vorgehen

- Anforderungen an Benchmark-Funktionen:
  - Überschaubare Komplexität
  - Analyseziel klar erkennbar
- Voraussichtliches Ergebnis abschätzen
- Implementierung
- Vergleich/Auswertung

3.2 Benchmark-Funktionen

- Umgang mit …
  - globalen und lokalen Variablen
  - Funktionsparametern
  - Feldern
  - Zeigern
- …mit oder ohne Verzweigungen, Schleifen und Unterprogrammaufrufen

3.3 Ergebnisvergleich

- Ausführungszeit
- Ressourcenverbrauch
- Parallelität der Speicherzugriffe
  → systematische Flaschenhälse