Reconfigurable Computing

1. Introduction

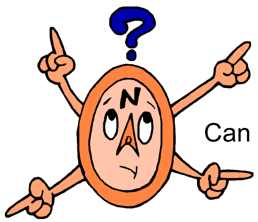
Marco Platzner
Computer Engineering Group

Chapter Overview

1.1 What is Reconfigurable Computing? (1)

General-Purpose Computer Systems

- Classically, computers consist of hardware and software
  - hardware (processor) fixed at fabrication time → hardware is static
  - software (programs) loaded after fabrication time → software is dynamic
- Reconfigurable computers change this
  - reconfigurable computers can adapt their hardware to the application after fabrication time → hardware is now dynamic



Can we make use of this?

What is Reconfigurable Computing? (2)

Embedded Systems

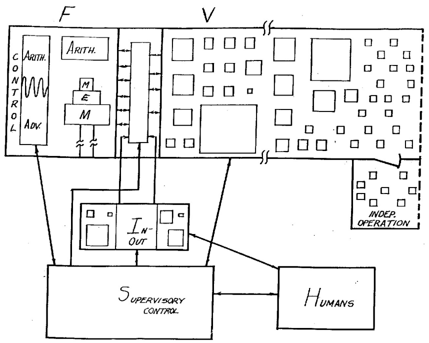
- Fixed-function device (eg. MP3 decoder chip)
  - computes one function which is defined at fabrication time
    - ☺ high performance for this function by customizing the hardware
    - ☹ totally inflexible because function cannot be changed after fabrication
- Programmable device (eg. ARM microprocessor)
  - computes any computable function which is defined after fabrication time
    - ☺ very flexible because we can change to any computable function anytime
    - ☹ low performance because the hardware is not customized to a specific function
- Reconfigurable device combines fixed-function and programmable
  - ☺ high performance for ☺ any computable function anytime by customizing the hardware to the function after fabrication time

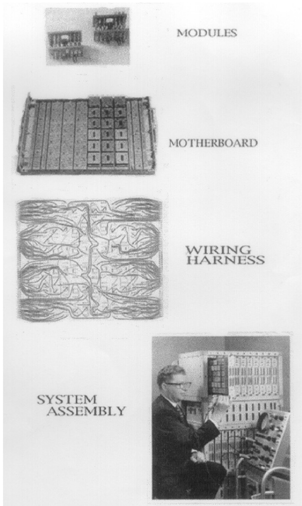
High-performance?

Any function anytime?

Reconfigurable Computing - History (1)

- 1960: G. Estrin at UCLA, “The Fixed Plus Variable Structure Computer”

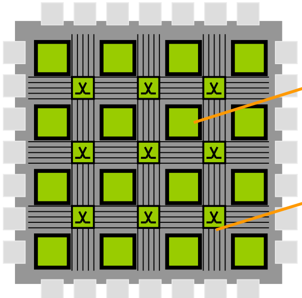


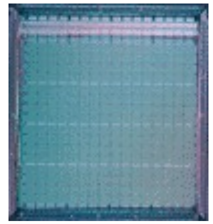


G. Estrin. “Reconfigurable Computer Origins: The UCLA Fixed-Plus-Variable (F+V) Structure Computer”, IEEE Annals of the History of Computing, pages 3-9, Oct-Dec 2000.

Reconfigurable Computing - History (2)

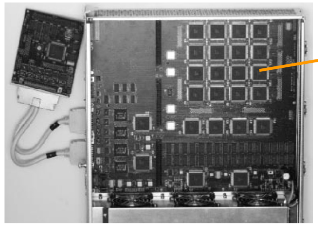
- 1985: Xilinx Inc. introduces the first SRAM-based Field-programmable Gate Array (FPGA)
  - FPGAs are positioned as high-end programmable logic devices
  - they are programmed by writing static RAM cells
  - the resulting volatility was considered a weakness for logic implementation





Reconfigurable Computing - History (3)

- Late ’80s / early ’90s
  - researchers realize that the volatility of SRAM-based FPGAs is key to FPGA-based computing systems
  - first FPGA-based custom computing machines
    - PAM @ DEC Paris Research Lab, Splash @ Supercomputing Research Center
    - beat supercomputers in signal processing, en/decryption, pattern matching
  - workshop on field-programmable logic founded in 1991 (FPL)
  - conference on FPGA-based computing machines founded in 1993 (FCCM)

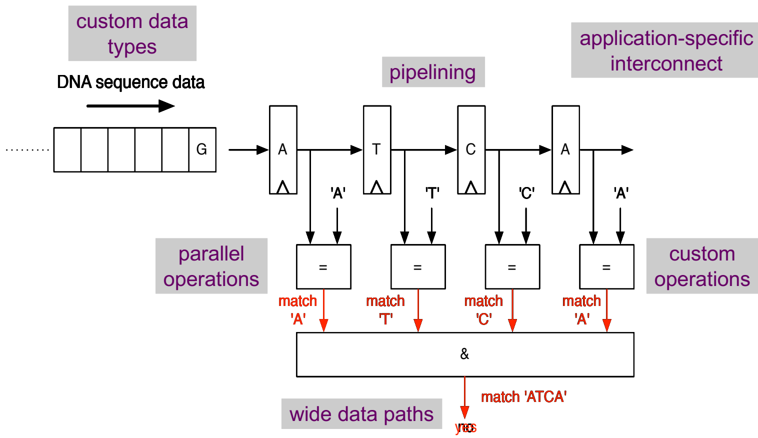


DECPeRLe-1 system

J.E. Vuillemin, P. Bertin, D. Roncin, M. Shand, H.H. Touati, and P. Boucard. “Programmable active memories: reconfigurable systems come of age”, IEEE Transactions on VLSI Systems, (4)1, 1996.

Potential of Computing in Hardware

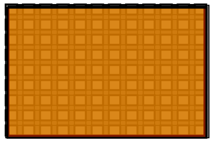
- Bioinformatics example: substring search in genome sequences



Reconfigurable Computing – History (4)

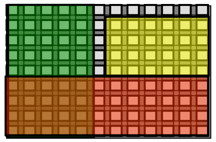
- ~1995: dynamic reconfiguration techniques: change hardware functionality at runtime

full dynamic reconfiguration



FPGA

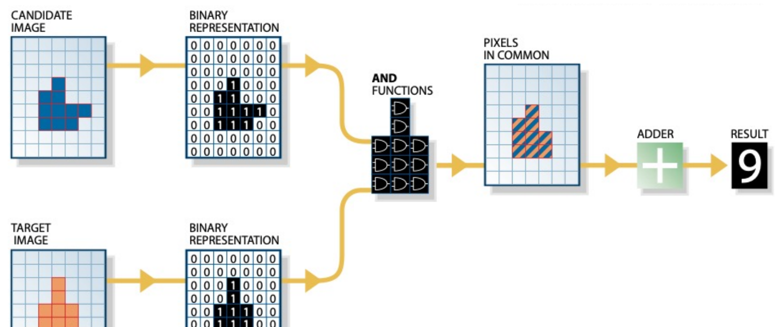
partial dynamic reconfiguration



FPGA

Reconfigurable Computing – History (5)

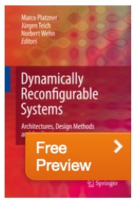
- 1997-2003: DARPA program "Adaptive Computing Systems"



J. Villasenor and W. H. Mangione-Smith, "Configurable Computing", Scientific American, pages 66-71, June 1997

Reconfigurable Computing – History (6)

- 2003-2009: Priority Program "Rekonfigurierbare Rechensysteme" of the German Research Foundation



M. Platzner, J. Teich, and N. Wehn (Eds.). "Dynamically Reconfigurable Systems Architectures, Design Methods and Applications" Springer 2010.

- Today, FPGAs …
  - are one of the fastest growing sectors of the semiconductor market
  - are being used for computing tasks in both embedded systems and high-performance computers
- In the last 30 years, many startups entered (and left) the market with new types of devices and tools

Reconfigurable Computing - History (7)

- Research community
  - has rapidly grown over the last 30 years, includes microelectronics, computer science & engineering, and many application domains
  - conference/workshop series
    - FPL Field-programmable Logic and Applications
    - FCCM Field-programmable Custom Computing Machines
    - FPGA Field-programmable Gate Arrays
    - FPT Field-Programmable Technology
    - and many more …
  - journals (started 2007/08)
    - Transactions on Reconfigurable Technology and Systems (ACM)
    - International Journal on Reconfigurable Computing (Hindawi)

Nomen Est Omen

Adaptive Computing

Custom Computing

FPGA-based Computing

Configurable Computing

Reconfigurable Computing

Computing with Virtual Hardware

1.2 For Whom is this Course

- Goals
  - introduce to the field of Reconfigurable Computing
  - provide an overview over FPGA architectures and design tools
  - give first practical experience in programming FPGAs
  - serve as a starting point for research activities
- Addressed study programs
  - Computer Science (CS) master students
    - elective module in focus area “Computer Systems”
  - Computer Engineering (CE) master students
    - elective module in focus areas “Computer Systems” and “Embedded Systems”
- Prerequisites
  - this course covers a wide range of topics from micro/nano-electronics to algorithms, the lab includes programming of hardware and software
  - no formal prerequisites w.r.t other Master-level courses
  - BUT: you need solid Bachelor-level knowledge in digital design, algorithms and programming

1.3 Course Content & Organization (1)

- Lecture – Chapters (tentatively)
  1. Introduction
  2. Evolution of programmable logic devices
  3. Computer-aided design for FPGAs
  4. FPGA architectures
  5. High-level languages for programming FPGAs
  6. Comparison of devices and technologies
  7. Reconfigurable systems
  8. (Selected) research topics

Course Content & Organization (2)

- Lecture uses the “inverted classroom model”
  - elements of lecture and post-processing are swapped (inverted)
  - learning activities that students can do well on their own are shifted to a preparation phase
  - the common attendance time is used for an active discussion
- Advantage
  - instead of frontal teaching, a time window is created for joint discussion and deepening of understanding
  - the lecture material can be studied at any time, as often as desired, and from anywhere
- Implementation
  - I make material available in PANDA (slides, screencast+audio, exercises)
  - You prepare independently for the classroom sessions
  - We use the common attendance times for
    - clarification of specific questions, discussions
    - exercises, examples, quizzes
    - reflection of the learning process

Course Content & Organization (3)

- Classroom times
  - Wednesday 14:15 - 15:45, room D2
  - contact: Marco Platzner
    O3.207, ☎ 60 5250, platzner@upb.de
- Lab sessions and times
  - lab dates/times will be announced
  - there will be two lab groups with physical presence (recommended), however, it will be possible to attend the lab remotely
  - contact: Christian Lienen (lab coordinator)
    O3.116, ☎ 60 4447, christian.lienen@upb.de

    Heinrich Riebler
    O3.152, ☎ 60 5382, heinrich.riebler@uni-paderborn.de

Course Content & Organization (4)

- Lab modules
  - module #1: FPGA design tool flow (Xilinx Vivado)
    → required course achievement (!)
  - module #2: Hw/sw codesign and high-level synthesis (Xilinx PYNQ)
    → bonus: one grade step improvement in the final grade
  - module #3: High-performance computing with oneAPI
    → bonus: one grade step improvement in the final grade
- Grading
  - passing lab module #1 is a required course achievement
  - passing lab module #2 improves grade by 1 grade step (if exam passed)
  - passing lab module #3 improves grade by 1 grade step (if exam passed)
  - oral exam (~45') covering lecture + exercises + lab