Introduction to reconfigurable systems

- **Reconfigurable system (RS)** = any system whose sub-system configurations can be changed or modified after fabrication.

- **Reconfigurable computing (RC)** is commonly used to designate computers whose processing elements, memory units, and/or interconnections can change function and/or (spatial) configuration after fabrication, prior or during the run-time of a particular program or part of a program.

- **Classical computer system** deals with software running on hardware,
- **The new branch of CS** deals with *flowware* running on hardware, as with *configware* and *flowware* “running” on *morphware*. 
Books

1. Gokhale, Maya, B., Graham, Paul S.: **Reconfigurable Computing**
   Accelerating Computation with Field-Programmable Gate Arrays
   2005, 238 p., Springer Netherland, Hardcover

2. Bobda Ch.: **Introduction to Reconfigurable Computing**
   Architectures, Algorithms, and Applications, Springer Netherlands,

3. Papers on the web page of the course Reconfigurable Circuits:

   1. **Configware and morphware going mainstream**
   2. **A Decade of Reconfigurable Computing: a Visionary Retrospective**
   3. **Software Technologies for Reconfigurable Systems**
   4. **Programovatelné logické obvody**
   6. **Seeking Solutions in Configurable Computing**

* [http://www.arl.wustl.edu/~lockwood/class/cse566-f04/](http://www.arl.wustl.edu/~lockwood/class/cse566-f04/)
* [http://www.cs.cmu.edu/~seth/15828/](http://www.cs.cmu.edu/~seth/15828/)
* [http://csdl2.computer.org/comp/proceedings/fccm/2002/1801/00/18010307df](http://csdl2.computer.org/comp/proceedings/fccm/2002/1801/00/18010307df)
Hardware

Architectural components of a configurable computer (Fig.1)

RS use FPGAs, CPLD or other programmable hardware to accelerate algorithm execution by mapping compute-intensive calculations to the reconfigurable substrate. These hw resources are frequently coupled with a general-purpose microprocessor that is responsible for controlling the reconfigurable logic and executing program code that cannot be efficiently accelerated.

Fig.1 Mangione-Smith, Et al. (1997)
There are **two primary methods in conventional computing** for the execution of algorithms.

- The first is to use hardwired technology, either an **Application Specific Integrated Circuit (ASIC)** or a group of individual components forming a board-level solution, to perform the operations in hardware. ASICs are designed specifically to perform a given computation, and thus they are very fast and efficient. However, the circuit cannot be altered after fabrication.

- The second method is to use **software-programmed microprocessors** — execute a set of instructions to perform a computation. By changing the software instructions, the functionality of the system is altered without changing the hardware. The processor must read each instruction from memory, decode its meaning, and only then execute it.
**Reconfigurable computing** is intended to fill the gap between hardware and software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware.

Reconfigurable devices, including field-programmable gate arrays (FPGAs), contain an array of computational elements whose functionality is determined through multiple programmable configuration bits. These elements, sometimes known as logic blocks, are connected using a set of routing resources that are also programmable.

In this way, custom digital circuits can be mapped to the reconfigurable hardware by computing the logic functions of the circuit within the logic blocks, and using the configurable routing to connect the blocks together to form the necessary circuit.
The core component of a reconfigurable system is **programmable hardware that can be temporarily (partly) customized for a specific program or part of a program**.

This enables very effective computations due to the efficient implementation of applications or program-specific operations directly in the programmable hardware. Such implementations can exploit various sorts of parallelism through extensive use of the “computing-in-space” paradigm (versus the “computing-in-time” paradigm used in sequential “von Neumann-type” processors). Specifically, reconfigurable systems are capable of exploiting the loop-level parallelism and bit-level parallelism which are inherent in particular applications, algorithms or programs.

**Applications fields**, such as pattern recognition, image processing, signal processing, cryptography, symbolic computations, set theoretical computations, and logical computations, it is important to exploit the capabilities of reconfigurable systems in these areas.
Proximity of the CPU to the programmable hardware.

First-generation systems typically used peripheral buses like the Sparc SBus to provide a coprocessor-like structure. Recently, some researchers have argued that the programmable hardware must be much closer to the processor, perhaps even on the datapath, fed by processor registers.

Fig. 2 Different levels of coupling in a reconfigurable system; (Reconfigurable logic is shaded). Compton, Hauck (1992)
Reconfigurable architectures

FPGAs form the processing building blocks of reconfigurable computers. The most common RC configuration is an accelerator board that plugs into the I/O slot of a microprocessor. The plug-in board typically contains

- one or more FPGAs,
- interface logic for the FPGAs to communicate with the conventional computer’s I/O bus,
- memory local to the RC board, often double or quad data rate (DDR or QDR) Static Random Access Memory (SRAM) and/or higher capacity Synchronous Dynamic RAM (SDRAM),
- A/D interfaces or other serial communication links to acquire data or communicate over a network.
As an accelerator, the RC is used in one of two scenarios:

1. The FPGAs can be used to process high bandwidth data streams from the external serial I/O interface, perform data reduction, and send processed data at a lower volume to the host processor. Alternatively, the host sends archival data to the RC memory subsystem.

2. A second RC scenario is as an acceleration component of a cluster supercomputer. (In this configuration, the I/O interface on the FPGA board communicates with the interconnection network of the supercomputer, and gives the board access to the supercomputer’s global memory address space. An FPGA communicates over the interconnection network with processors.)

Commercial FPGA Companies (Flash and Antifuse FPGA+SRAM FPGA): Xilinx, Altera, Lattice, Actel, Other
The main hardware implementation technology of (re-)configurable systems RS is field programmable logic: Field Programmable Gate Arrays (FPGAs), and Complex Programmable Logic Devices (CPLDs).

The main reasons for using FPGAs and CPLDs were:

- the high hardware development and fabrication costs of ASICs and other hard-wired implementations,
- the high reliability of FPGAs and CPLDs,
- the short time-to-market when using programmable devices,
- the independence of system developers and IC manufacturers.

Most fine-grained (re-)configurable systems and many coarse-grained ones are based on look-up table (LUT) FPGA technology.

Implementing an application by software means procedural programming, i.e. programming in time. But implementing an algorithm on morphware (hardware) means structural programming by configware for structural reconfiguration of the morphware device and flowware for scheduling the data streams. The acronym field-programmable gate array (FPGA) indicates, that structural re-programming can be practiced anywhere, like e.g. at the customers site. An important commercial aspect.
Granularity of programmable hardware.

Reconfigurable systems can be divided into fine-grained and coarse-grained. In fine-grained systems configuration is at the level of individual bits or sub-words, while coarse-grained configurations are at the word level. Coarse-grained architectures require less configuration resources than fine-grained ones and have more area-efficient interconnection routing switches.

Software

- Application development involves the use of traditional CAD tools, which were developed for ASICs. The input to the multi-FPGA mapping software may be a description in a hardware description language such as Verilog or VHDL, a software programming language such as C or C++, or perhaps a structural circuit description.

- Steps: (see fig. next page)

- When reconfigurable systems use more than one FPGA the design must first be partitioned into the different FPGA chips. The first mapping step is logic restructuring, which alters the input netlist so that it can be mapped to the multi-FPGA system. The process of assigning partitions to specific FPGAs (global placement) in the system, is combined with the partitioning stage. After this global routing handles the routing of inter-FPGA signals and Pin assignment then decides which specific I/O pins on each of the FPGAs will carry the inter-FPGA signals.

The next steps are the technology mapping, placement and routing of the individual FPGAs in the system and the download of all files.
Hauck-Agarval (1996)
Details of the Xilinx CLB (left) and switchbox (top right) [Xilinx94]. The multiplexers, LUTs, and latches in the CLB are configured by SRAM bits. Diamonds in the switchbox represent six individual connections (bottom right), allowing any permutation of connections among the four signals incident to the diamond.
Topologies (e.g.)

Mesh (left) and crossbar (right) topologies. In the crossbar, chips A-D are routing-only, while W-Z hold all the logic in the system.

Figure 13. A hierarchy of crossbars. FPGAs M-T hold all the logic in the system. Chips E-H and I-J form two first-level crossbars, and chips A-D form a second-level crossbar.
The Splash 2 topology [Arnold92]. The linear array of FPGAs (A-H) is augmented by a routing-only crossbar (R). Note that the real topology has 16 FPGAs in the linear array. Hauck-Argaval 1996

The G800 board [Giga95]. The base board has two FPGAs and four sockets. The socket at left holds four computing module boards (the maximum allowed in a socket), while the socket at right has none.
One of the benefits of reconfigurable computing is the ability to execute multiple operations in parallel. In cases where circuits are specified using a structural hardware description language, the user specifies all structures and timing, and therefore either implicitly or explicitly specifies any parallel operation.

For behavioral and HLL descriptions, there are two methods to incorporate parallelism: manual parallelization through special instructions or compiler directives, and automatic parallelization by the compiler.
A programming bit for SRAM-based FPGAs [Xilinx 1994] (left) and a programmable routing connection (right).

D flip-flop with optional bypass (left) and a 3-input LUT (right).
A basic logic block, with a 4-input LUT, carry chain, and a D-type flip-flop with bypass.

A generic island-style FPGA routing architecture.

[Xilinx 1994]
The functional unit from a Xilinx 6200 cell [Xilinx 1996].

Segmented (left) and hierarchical (right) routing structures. The white boxes are logic blocks, while the dark boxes are connection switches.
Technology mapping

Figure 2. Example of 5-input LUT technology mapping. The input circuit (left) is restructured and grouped together into 5-input functions (right). The gray loops at right indicate individual LUTs. The numbers on the gates are for identification.

Hauck, Argaval 1996
Figure 3. Placement of the circuit from Figure 2. The numbers are the number of the LUT (from the technology mapping) assigned to each logic block, while the letters are the assignment of input signals to I/O blocks.
the performance of the circuit. By running both algorithms together, a complete routing solution can be created.

**Figure 5.** Routing of the placement from Figure 3. The small black squares are the configuration points used for this mapping.

• Philip Garcia, Katherine Compton, Michael Schulte, Emily Blem, and Wenyin Fu: An Overview of Reconfigurable Hardware in Embedded Systems (web)

• Thank you for your attention