Prototyping Environment for Reconfigurable Processors

2nd Dagstuhl Seminar on Dynamically Reconfigurable Architectures

Dresden University of Technology
Institute of Computer Engineering

Sergej Sawitzki, Steffen Köhler, Rainer G. Spallek
{sawitzki,stk,rgs}@ite.inf.tu-dresden.de

Dagstuhl, June 2000

Overview

1. Introduction to Reconfigurable Processors
2. Test Environment
   - Hardware
   - Software
3. Prototyping Case Study: CoMPARE
   - Approach
   - Results
4. Summary and Further Work

Reconfigurable Microprocessors: A Short Summary

Processor	Year	Current State
PRISC	1994	Simulation model
nP	1994	Prototype implemented with XC3090 FPGA
Spyder	1996	Prototype implemented with non-commercial multi-FPGA board
OneChip	1996	Prototype implemented with TM1 board (non-commercial)
Garp	1997	Simulation model
CoMPARE	1998	Prototype implemented with FLEX10K20 FPGA
NAPA1000	1998	Project canceled before layout
Proteus	1999	Conceptual approach

Approaches to Processor Prototyping

Dedicated environments: multi-FPGA or CPLD boards with on-board memory and dedicated software. Costs: $ 1000–10000. Examples: TM-1, RPM

Commercial platforms: general purpose solutions for *reconfigurable computing*, multi-FPGA or CPLD boards with on-board memory, driver and monitoring software. Costs: $ 1000–10000. Examples: Hot-Works, Pamette, RP-1000

Evaluation boards: usually a single FPGA or CPLD, no memory, CAD software. Costs: $ 100–1000. Examples: Xilinx Evaluation Board, Altera UP1

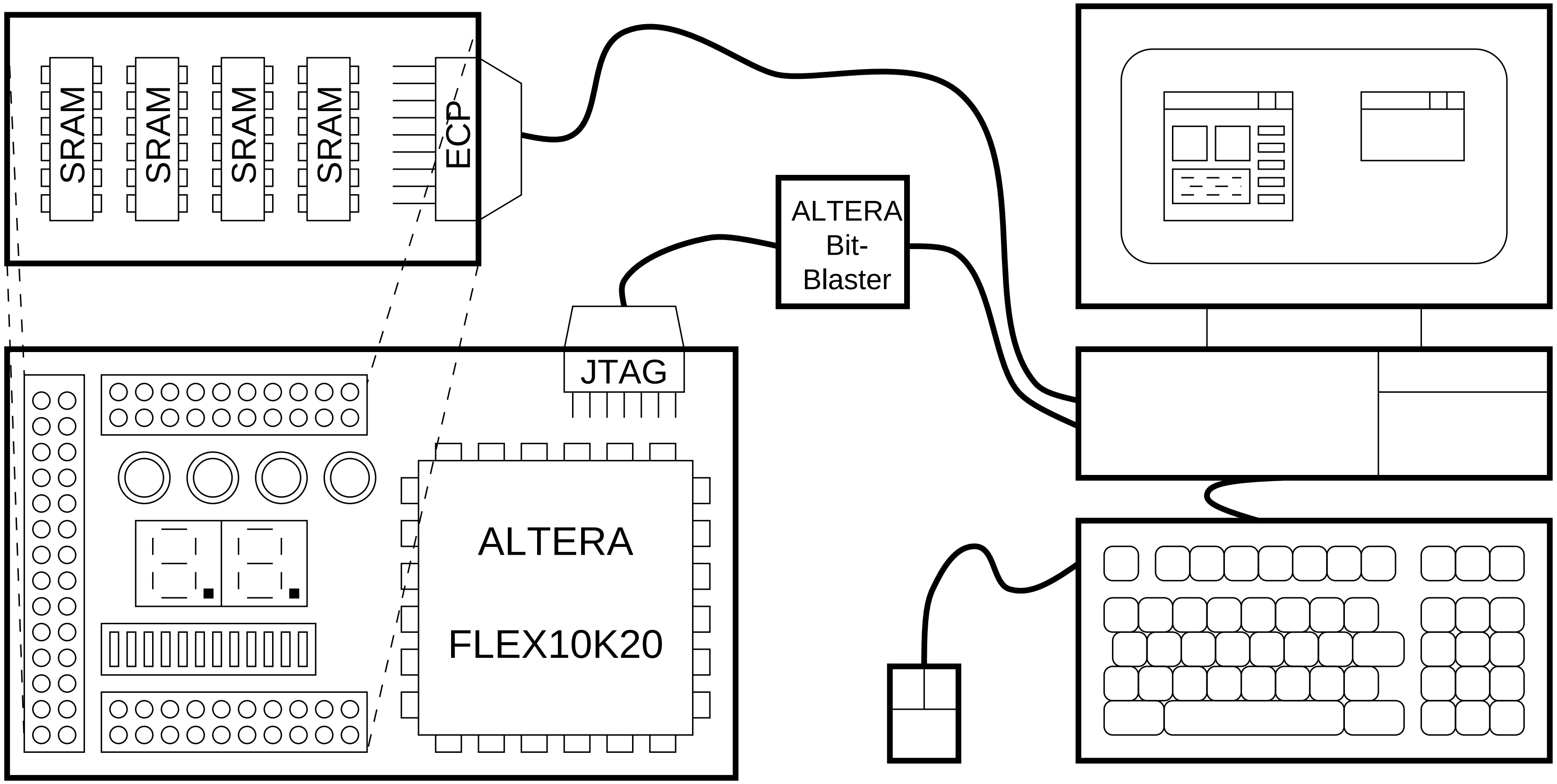
Basic Requirements for the Test Environment

Flexibility: since the design (or the configuration) often changes, the reconfiguration time should be kept short $\longrightarrow$ Constraints considering the bandwidth of the host connection

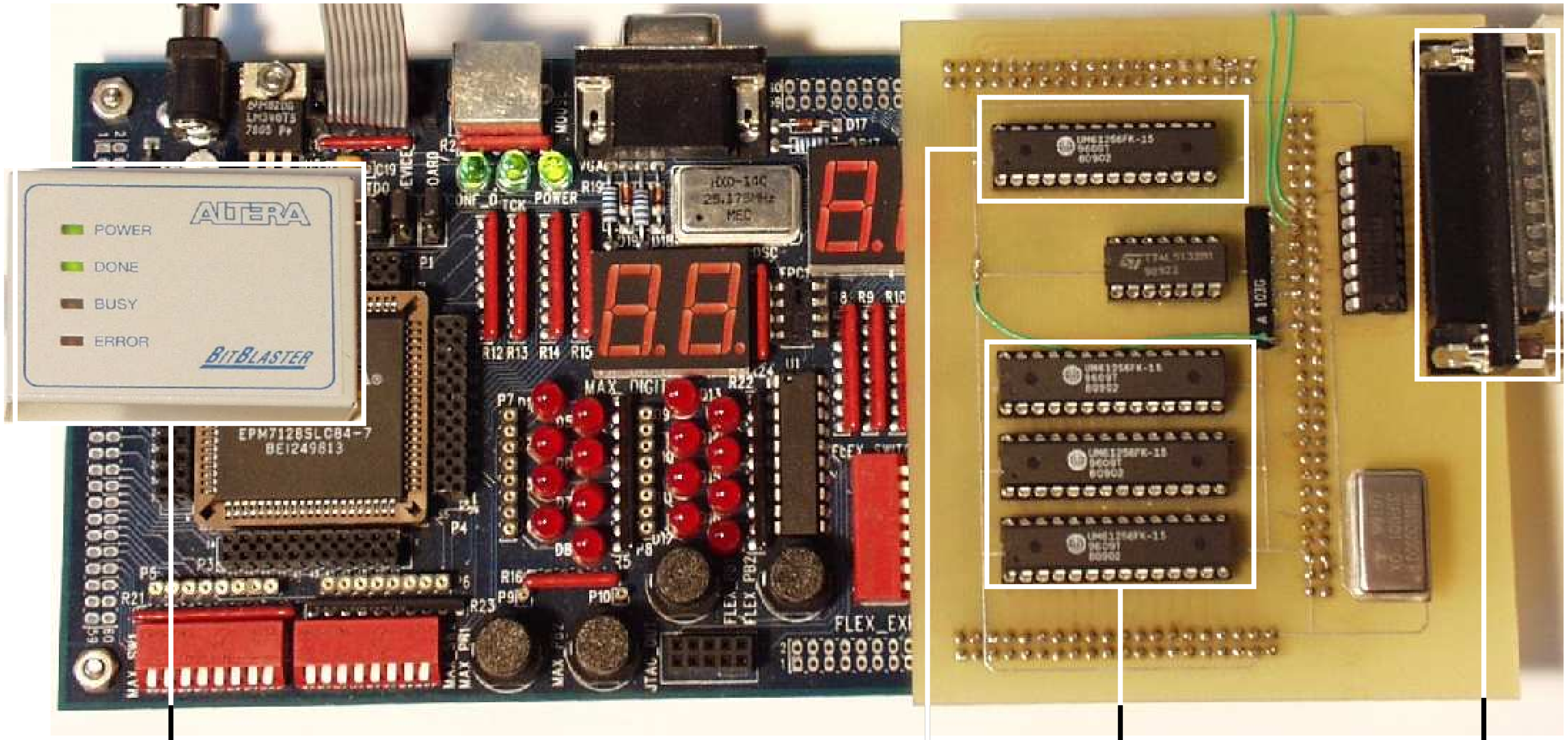
Scalability: wide data paths (32 or 64 bit) do not map satisfactory on the most modern FPGA architectures $\longrightarrow$ Demand for sound performance estimates based on data for narrow (8 or 16 bit) data paths

Observability: state changes of both reconfigurable and fixed processor components should be observable $\longrightarrow$ Requirements for the communication interface (hardware and protocol)

Prototyping Environment: Hardware (1)



Prototyping Environment: Hardware (2)



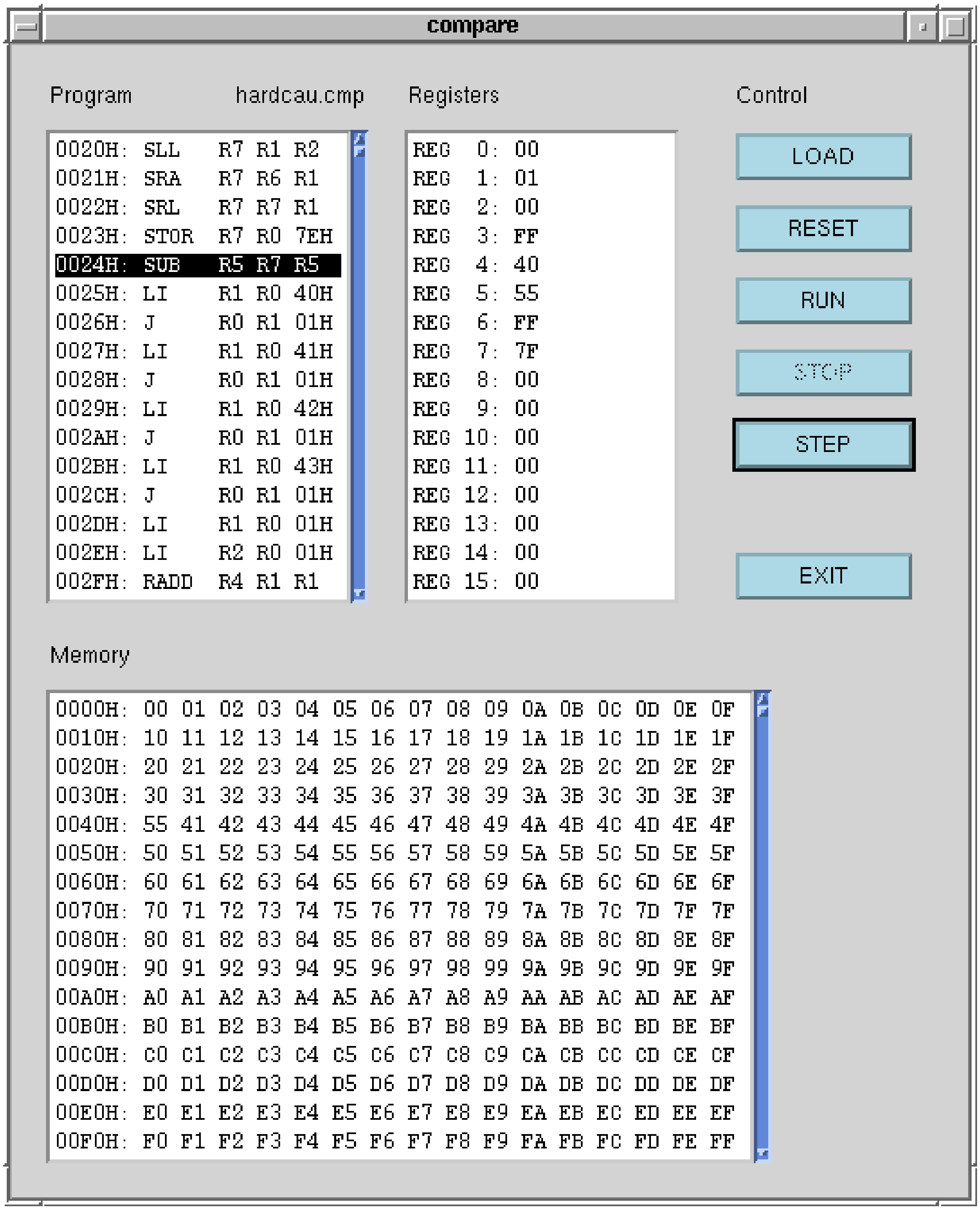
Prototyping Environment: Software(1)

Communication: Connection between the host and the FPGA board via ECP interface (dedicated extendable communication protocol)

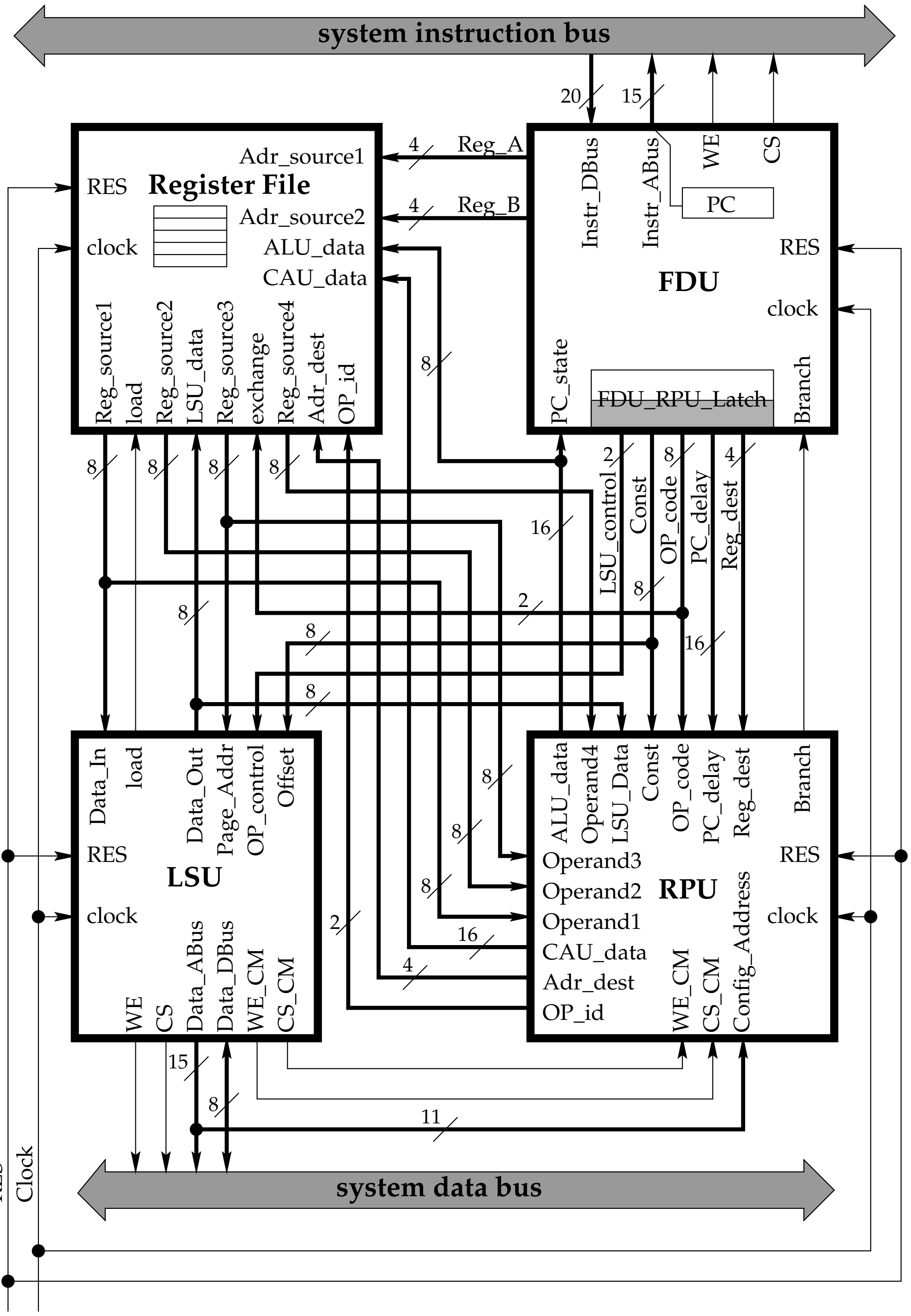
Visualization: Motif GUI for the visual representation of the system state (PC, registers, memory)

Management: initialization, code and configuration download, run and single-step execution, debugging

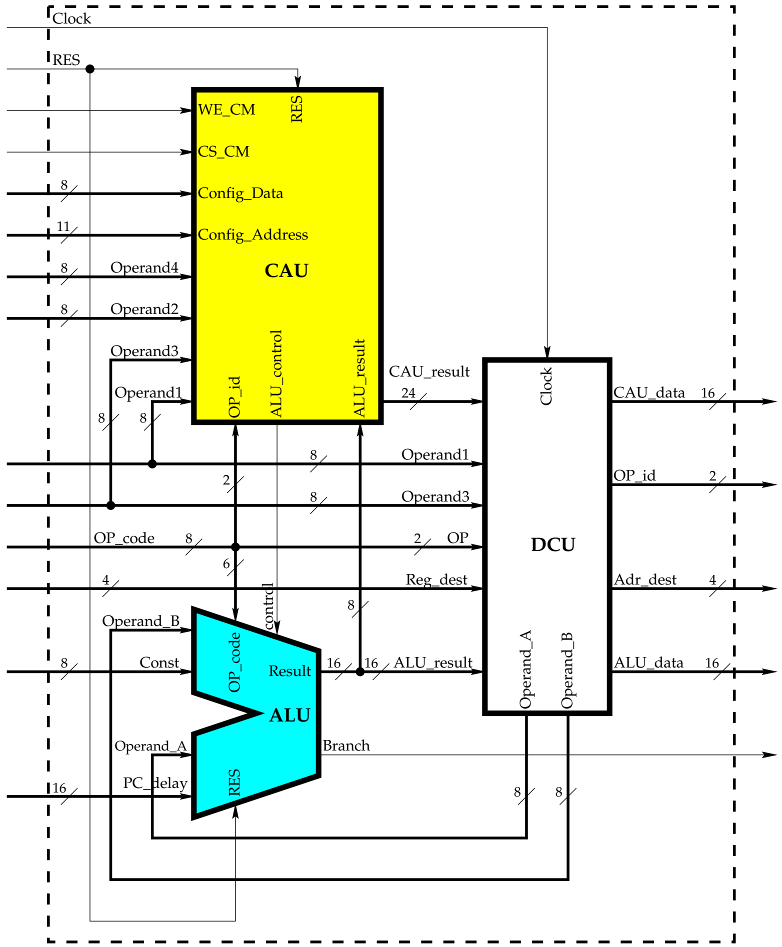
Prototyping Environment: Software(2)

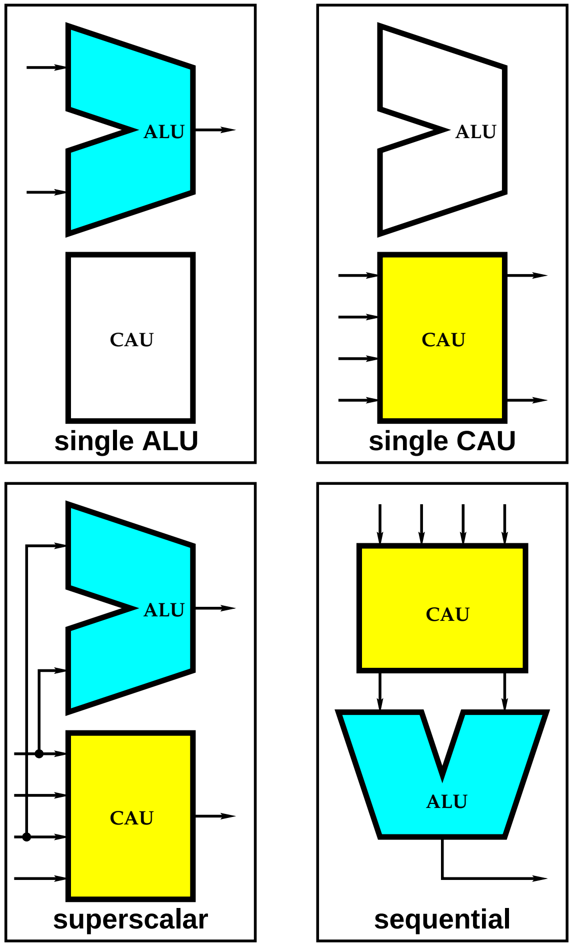


CoMPARE: Top Level View

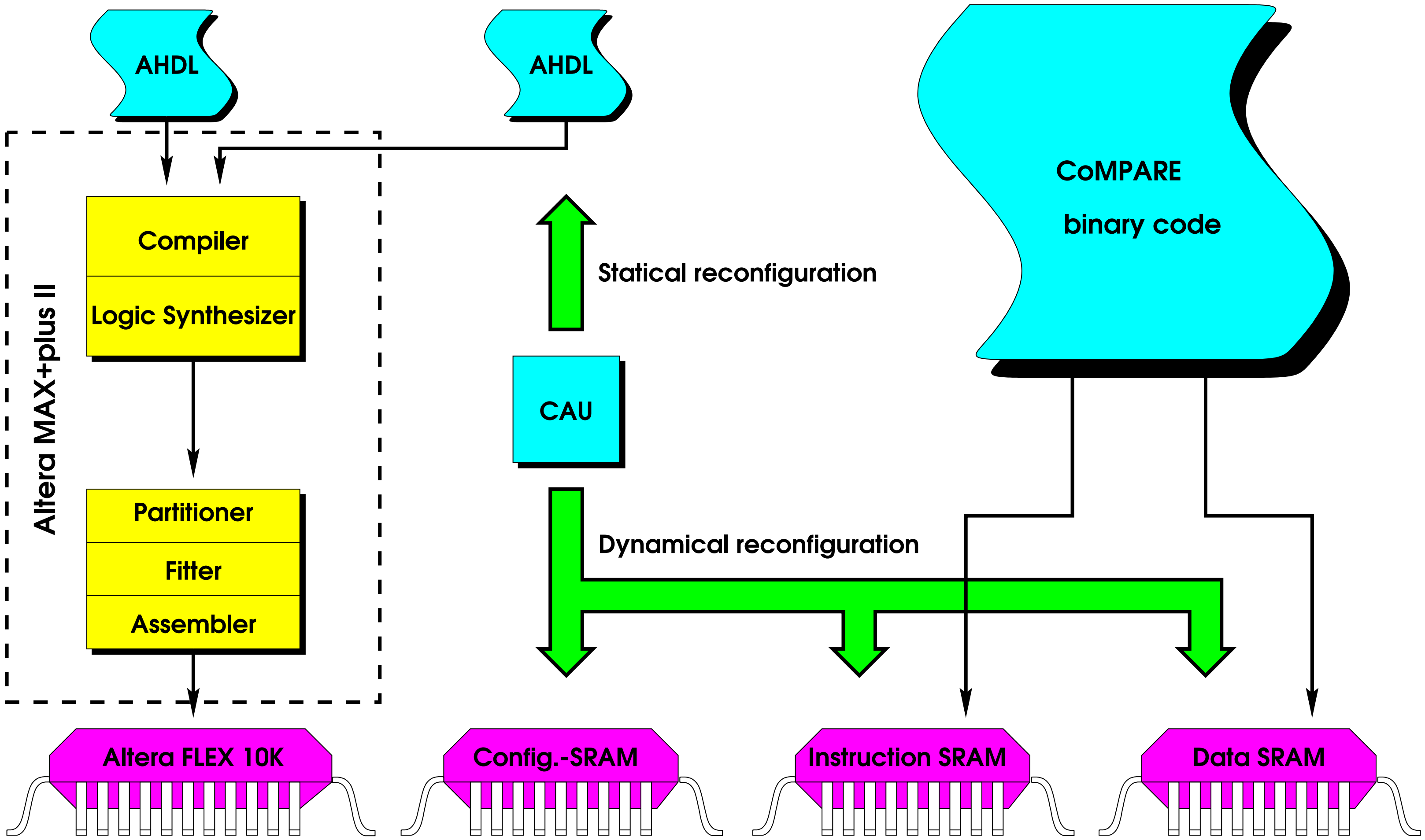


RPU Structure





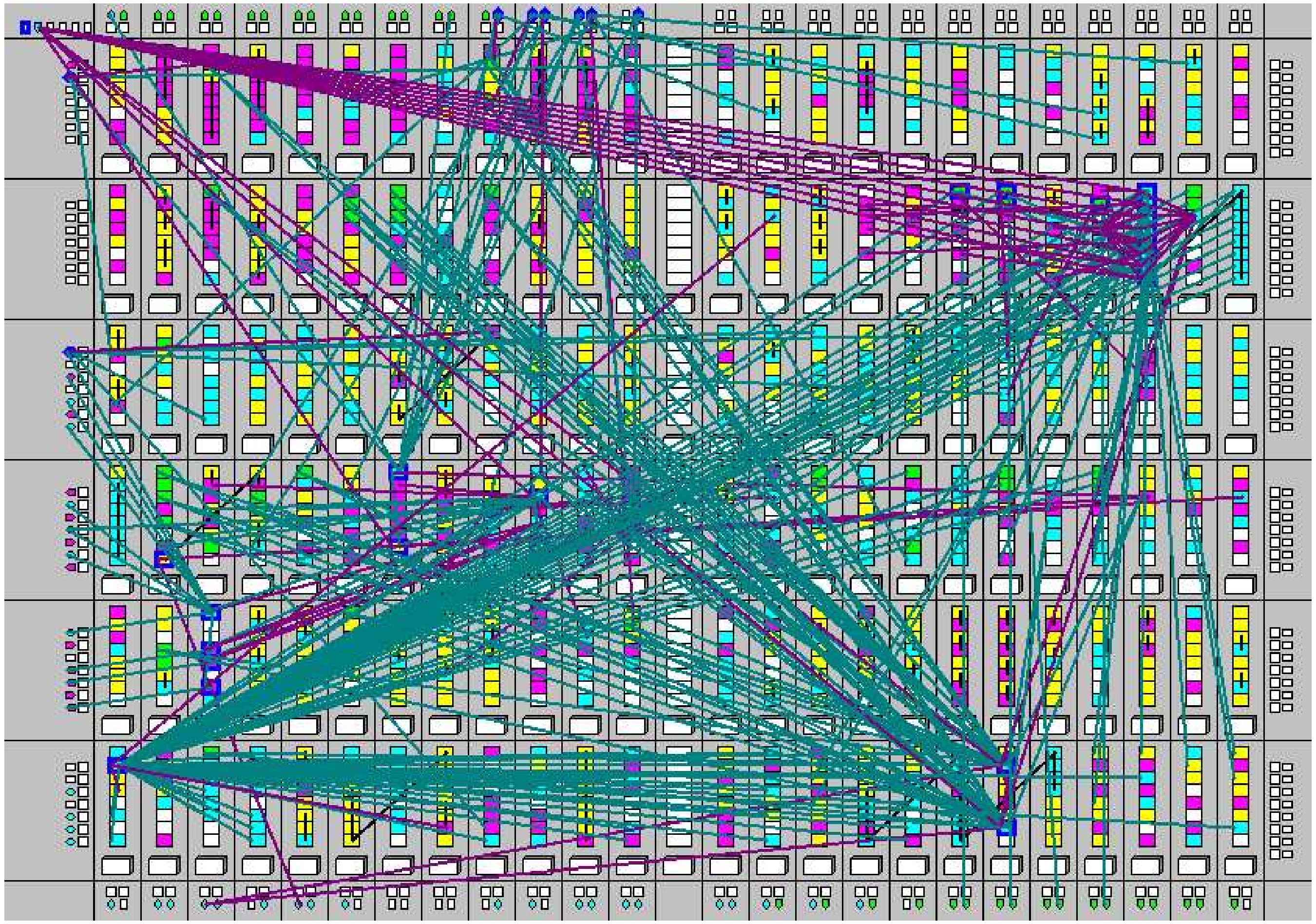
Design and Evaluation Flow



AHDL Design Example

```
SUBDESIGN lsu
(Data_In[7..0], OP_control[1..0],
 Page_Addr[7..0], Offset[7..0],
 Data_Bus_In[7..0], RES, clock  : INPUT;
 Data_MRead, Data_MWrite, WE_CM,
 load, Data_Out[7..0], Data_ABus[14..0],
 Data_Bus_Out[7..0]             : OUTPUT;)
VARIABLE loadff          : DFF;
BEGIN
 Data_Out[]        = Data_Bus_In[]; % Read %
 Data_MRead        = OP_control0;
 loadff.d          = !OP_control1 & OP_control0;
 loadff.clrn       = !RES;
 loadff.clk        = clock;
 load              = loadff.q;
 Data_ABus[7..0]   = Offset[];
 Data_ABus[14..8]  = Page_Addr[6..0]; % Write%
 Data_MWrite       = OP_control1 & !OP_control0;
 WE_CM             = !Page_Addr7 & Page_Addr6 &
                     OP_control1 & OP_control0;
 Data_Bus_Out[]    = Data_In[];
END;
```

Mapping of the Datapath



Synthesis Results

Functional Unit	**Cell Count (LC)**	**Cell Count in % of FLEX10K20**
Fetch and Decode Unit (FDU)	119	10
Register File	364	32
Load and Store Unit (LSU)	35	3
Arithmetic and Logic Unit (ALU)	192	17
Datapath Control Unit (DCU)	54	5
Configurable Array Unit (CAU)	max. 226	max. 19
Debugging-Interface	162	14

Clock frequency $\approx$ 10 MHz, **Power dissipation** $\approx$ 2 W

Coding Example

```
# '#' indicates comment
# Instruction memory section
# Format: I<i-mem address>:<instr-code>
# Preloading some immediates
I0000:05005    # LI R5,5
I0001:06006    # LI R6,6
I0002:07007    # LI R7,7
# ADD: R1 = 07 + 07 = 0E
I0003:80177    # ADD R1,R7,R7
# AND: R2 = 06 and 05 = 04
I0004:88265    # AND R2,R6,R5
# Data memory section
# Format: D<d-mem address>:<data word>
D0040:55
D4001:09
D4101:0A
D4201:09
D4301:09
```

Summary and Further Work

- cost optimal solution for the prototyping of reconfigurable systems
- CoMPARE prototyping study confirms the simulation results
- timing optimization, IP generation
- completion of the software framework (assembler, compiler)
- research work on reconfigurable architectures for DSP applications
- efficiency comparisons with other architectures (reconfigurable meshes, systolic arrays)