

Creating Computers from (almost) scratch using FPGAs, VHDL and FORTH

Recreative explorations of the
hardware/software co-design space

Hans Hübner, ZSLUG Meetup, February 7th, 2011

Introduction

- Hacker since 1983
- Wanted to become an electronics Engineer
- Became a systems programmer instead
- Freelance developer



Hans Hübner

Overview

- The quest for a new Lisp machine
- FPGA introduction
- From CADR through SECD to Rekonstrukt
- Conclusions

The quest for a new Lisp Machine

- The VAX running VMS is the blueprint of a beautiful machine
- Lisp machines are even nicer!
- Power from the machine level up to the highest level of application programming
- Can this architecture be preserved?



But RISC prevailed, rightfully so!

- Cheaper
- Faster
- Beauty lies in the eye of the beholder
- The compiler is smarter
- This presentation is about recreation, though!

FPGA Introduction

Introducing FPGAs

- Arrays of logic cells
- Cell behavior and interconnection defined by configuration memory
- Configuration can be changed in the target application

FPGA Architectures

- Cell architecture and interconnect structure are vendor specific
- Cells typically consist of flip flops, memory, lookup table
- High-speed I/O supported by transceiver blocks
- Special purpose blocks are typically available: RAM arrays, multipliers

Scale

- Low-cost FPGAs today are equivalent to 100,000-1,000,000 "standard gates", plus embedded memory.
- Cycle speeds in the order of 100 Mhz are achievable with standard tools
- Small FPGAs in DIY-friendly packages provide for about 100 I/Os, BGA packages with 400 I/Os are common.

Development Environment

- Circuits are described by hardware description language (VHDL, Verilog, System C) or by schematic capture
- Hardware description is converted to a FPGA configuration bitstream with synthesis tool)
- Multiple steps in the process with intermediate representations

Development Environment

- Standardized intermediate netlist representation allows for the integration of third-party IP
- FPGA design inherits from EDA, long turnaround times compared to software development

Development Environment

- VHDL and Verilog are standardized languages
- Vendors supply bitstream generator
- Synthesis Tools and simulators for lower-end chips are free

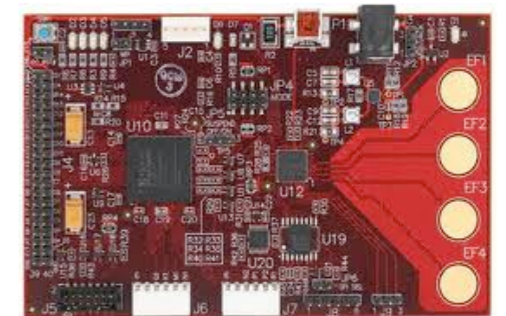
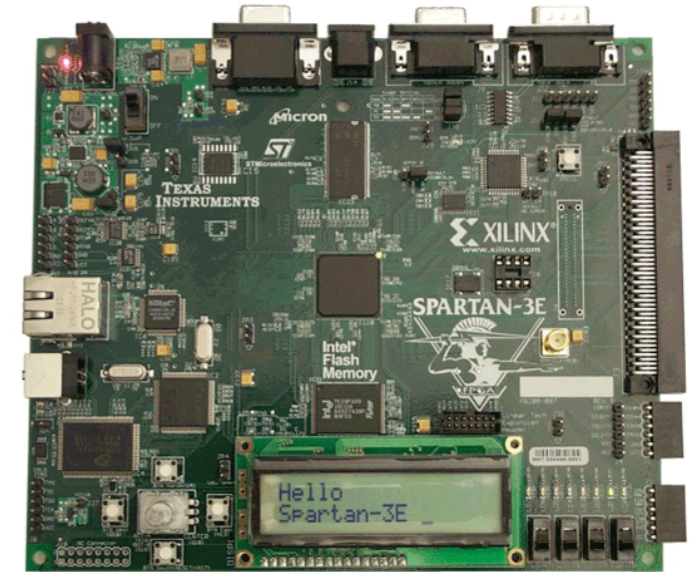
FPGAs in the industry

- Prototyping, circuit verification, circuit simulation
- Low-volume products
High-End communication equipment
copacobana DES cracker
Market data feed analysis
- Fast-turnaround products

FPGAs for Hackers

Entry is cheap

- FPGA evaluation boards start around \$100
- Development software is available at no cost (limited)
- Solder-friendly modules are available
- Windows is pretty much mandatory



FPGA Hacking in the wild

- Copyright-bit killer (elektor 1997)
Not strictly an FPGA project (used a
Altera MAX 7000 CPLD)
Removed copyright bit from digital
audio streams

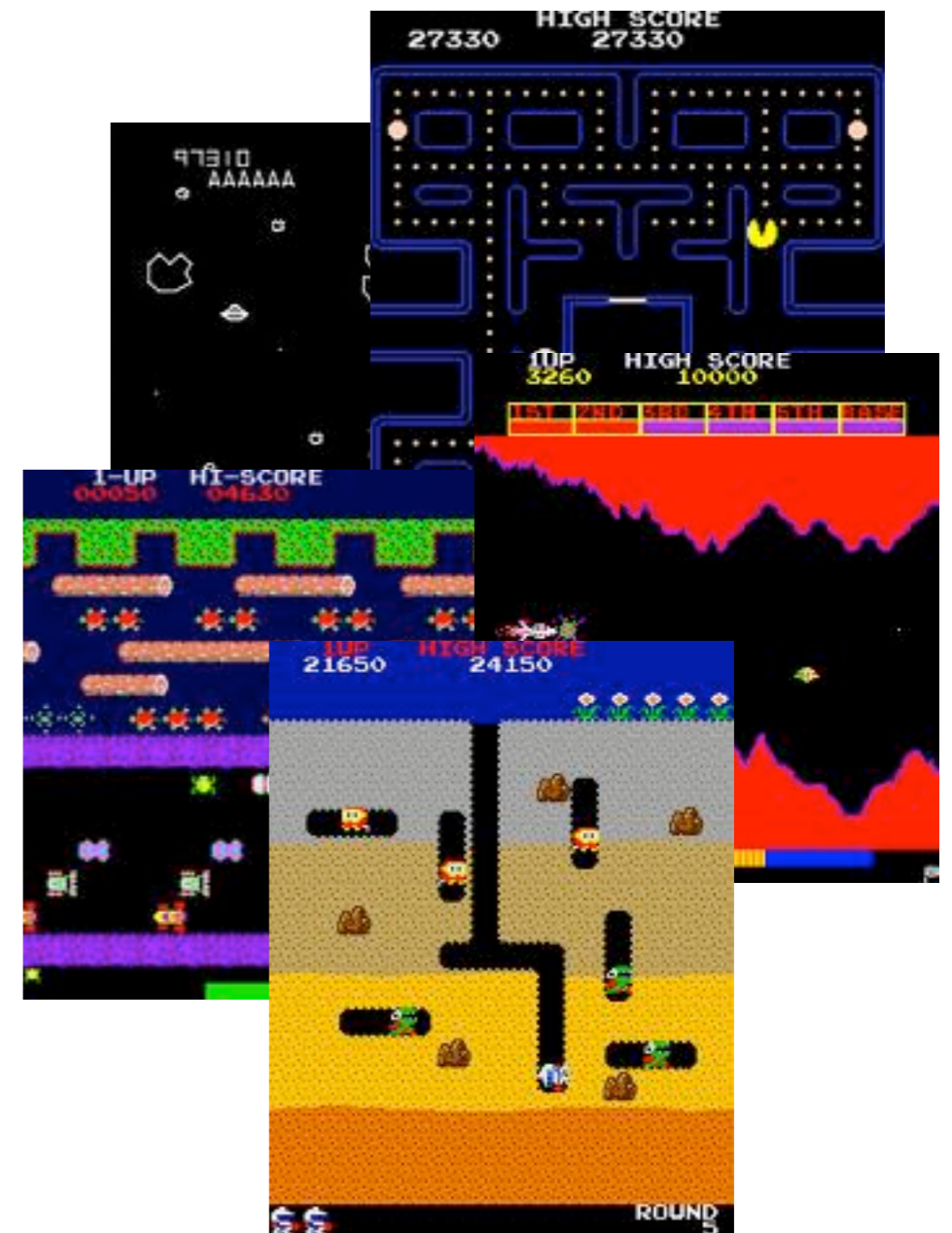
FPGA Hacking in the wild

- Game emulations

Re-create original arcade game hardware in FPGA

Run unmodified

original game code



FPGA Hacking in the wild

- Milkymist

VJ Platform built
around MIOS II and
RTEMS

Started on Xilinx
ML401 board, now has
its own hardware
GPL V3, supported
product



FPGA Building blocks

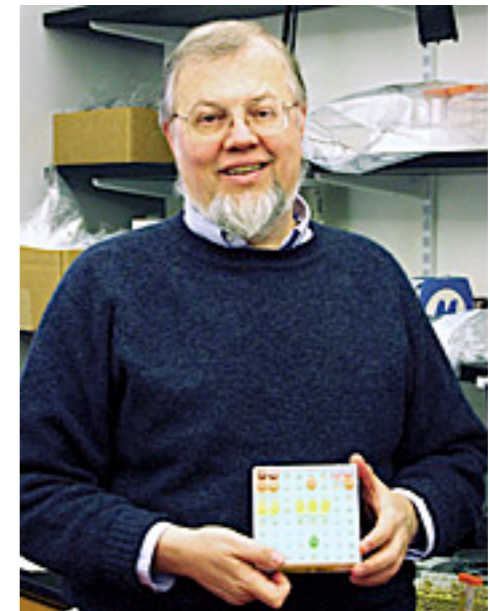
- Free implementations for 8bit-CPU's are available
- 16 and 32 bit CPU's exist, but are less common in the hacker world
- Other IP cores are available (opencores.org)



From CADR through
SECD to rekonstrukt

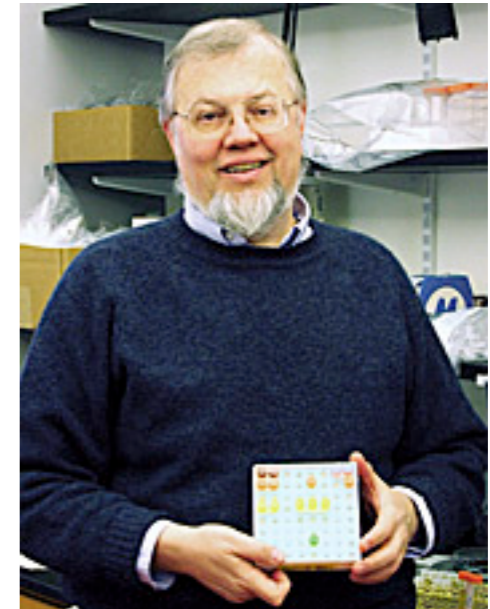
Resurrecting the MIT CADR

- Early Lisp machine design by Tom Knight
- TTL based, schematics available, software available - possible?
- Some discussions and partial results with collaborators



Resurrecting the MIT CADR

- Needs redesign for FPGA (synchronous operation)
- lispμ Microcode is 100 bits wide, needs loads of memory



Pick a smaller target

The SECD chip (1989)

- SECD is a virtual machine for functional programming languages
- PhD thesis project, formal specification and verification of a SECD CPU chip
- Thesis and proof published, including the microcode
- Small enough to fit into XC3S1000 FPGA

Reimplementing the SECD Chip from artifacts

- Disassembled binary representation of the microcode
- RTL description and graphics
- SECD instruction set documentation

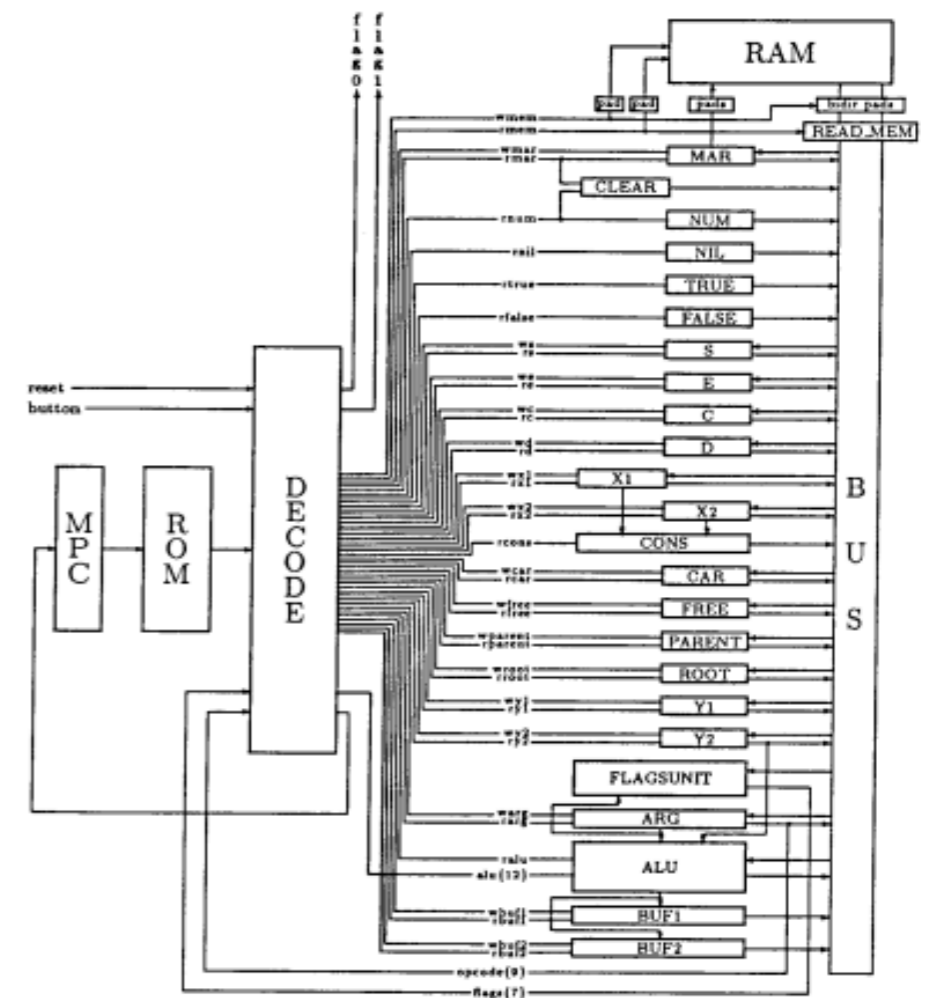


Figure 4.2: Register Transfer Level View of SECD Machine

Done, to a degree

- Bugs in RAP instruction and GC fixed (proof wtf?)
- Runs in simulation, and yes, it is slow
- Requires a frontend processor
- Debugged in simulator, runs test programs and compiler

Implementing the

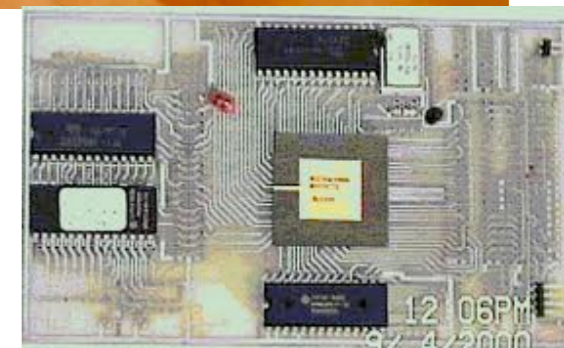
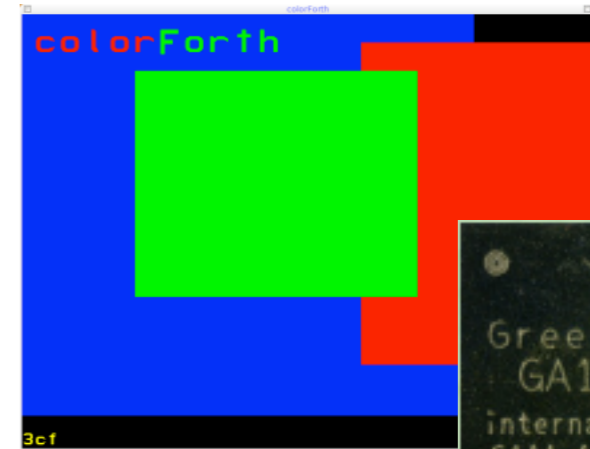
- SECD is a functional machine, no side effects
- SECD front end is responsible for all I/O
- Fills SECD memory, starts program, reads result
- MC6809 based, schematics available
- System09 to the rescue

Stumbling over the memory interface

- Hardware defect made the dual-port SRAM that sits between SECD and 6809 non-functional
- What to do with this working 6809 system?

How about some FORTH?

- FORTH has a tradition as low-level language
- FORTH CPUs exist, too
- FUN FORTH IS OK.

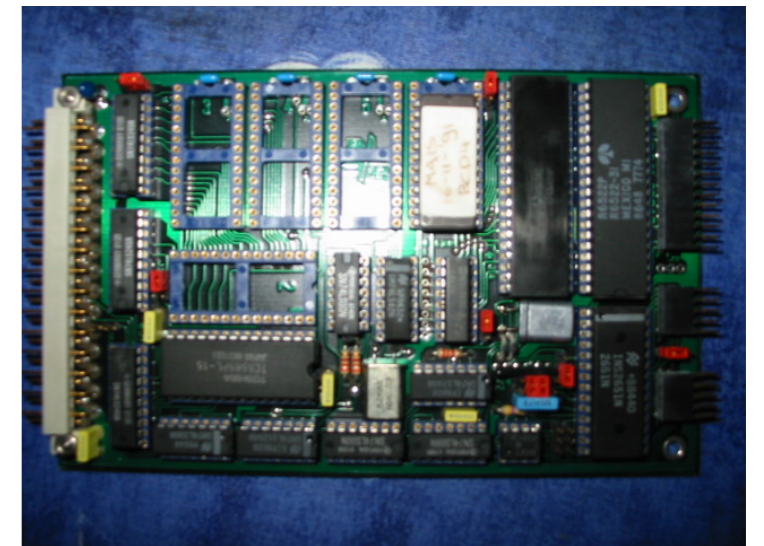


How about some FORTH?

- Play around with new hardware designs interactively
- 6809 architecture well suited to FORTH implementation (dual stacks)
- Existing FORTH implementation should be easy to port

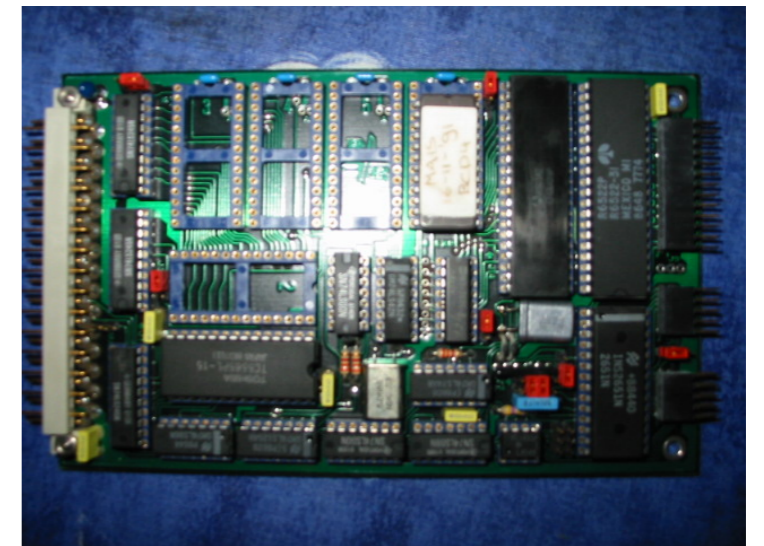
Porting MAIS FORTH to System09

- Written for simple, straightforward MC6809 hardware
- Complete ANS-like FORTH implementation



Porting MAIS FORTH

- Porting from two sides:
 - Change MAIS FORTHs UART driver to match System09 UART
 - Change System09 memory layout to match MAIS Kastje's



Packaging up

- System09 + MAIS Forth + I/O cores == rekonstrukt
- Beginnings of audio VHDL cores
- Available for download at <http://rekonstrukt.googlecode.com/>

Conclusions

- Hacking FPGAs is fun, and instructive
- Entry is cheap, but you really want a logic analyzer
- Trivial stuff can be done without simulation, but simulation will always save time

Conclusions

- Resurrecting the original Lisp machine will require specialized hardware
- If you require a microprocessor, use a microprocessor
- There is no reason not to try without :)

Thank you for
listening!