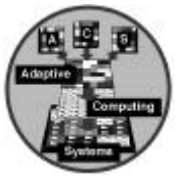


Algorithm Analysis and Mapping Environment for Adaptive Computing Systems

Eric Pauer, Cory Myers, Ken Smith, and Paul Fiore

{pauer,cory,jmsmith,pfiore}@sanders.com

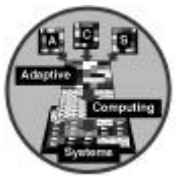
Sanders, a Lockheed Martin Company

Nashua, NH 03061

SANDERS
A Lockheed Martin Company

Statement of the Problem

Reconfigurable computing technology offers leap ahead performance, e.g. 10X ops per watt and/or ops per cubic inch, over general purpose programmable solutions without the need to develop custom hardware. However, today generation of a working implementation requires hardware design expertise and generation of a good implementation requires many slow iterations between an algorithm designer and a hardware developer.



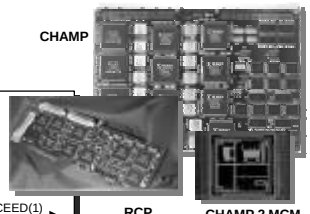
Adaptive Computing Performance Gain

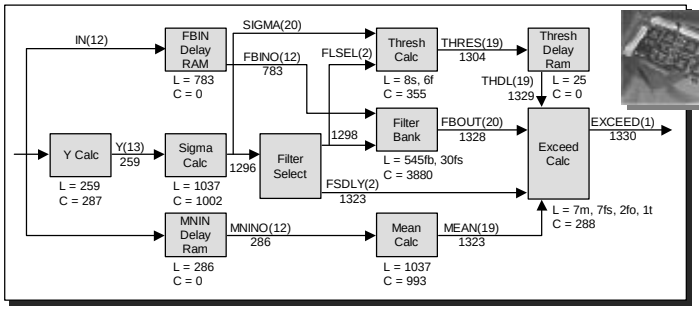
	CHAMP	TMS 320C80
Image Size	256 x 256	256 x 256
Implementation Time	44 Days	28 Days
Frame Rate	305 frames/sec	12 frames/sec
Latency	68 µsec	82,000 µsec
Processing Load	4.7 Bops	0.2 Bops
Utilization	73%	Unknown
Gates	510k	N/A

(Operation count increase by 70% if memory loads and stores are counted)

Greater than 10X performance
Design time measured in weeks

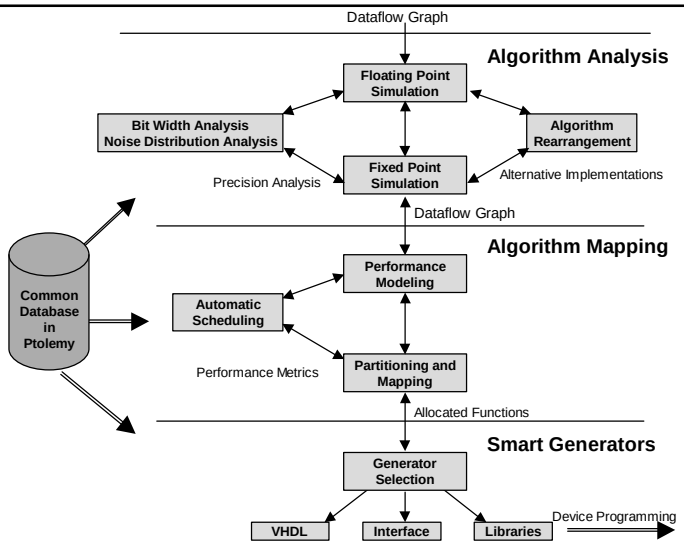
Reconfigurable Architectures





Analysis:
Bit Widths
Latency (L)
Cells Used (C)

Analysis and Mapping in ACS Environment



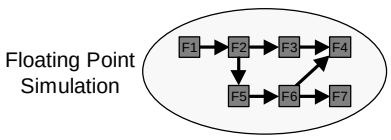
- **SNR analysis**
- **Alternative implementations**
- **Functional approximations**

- **Timing and sizing estimation**
- **Scheduling – FSM and contexts**
- **Partitioning within a resource node**

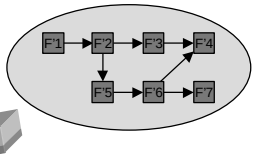
- **Device program**
- **Interface program**

Adaptive Computing Resource

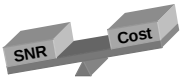
Automated Float to Fixed Point Translation



Optimize Wordlengths

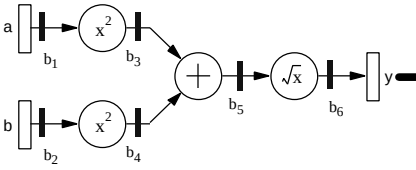


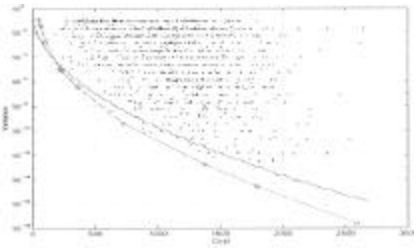
Fixed Point Realization



Maximize SNR(b), subject to Cost(b) " C_0 and b " b_{min} " 0
or
Minimize Cost(b), subject to SNR(b) " SNR_0 and b " b_{min} " 0

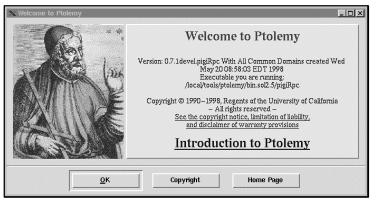
Yields
- Bit Widths for each flow
- Cost estimates (area/complexity)
- Quantization Noise (SNR)





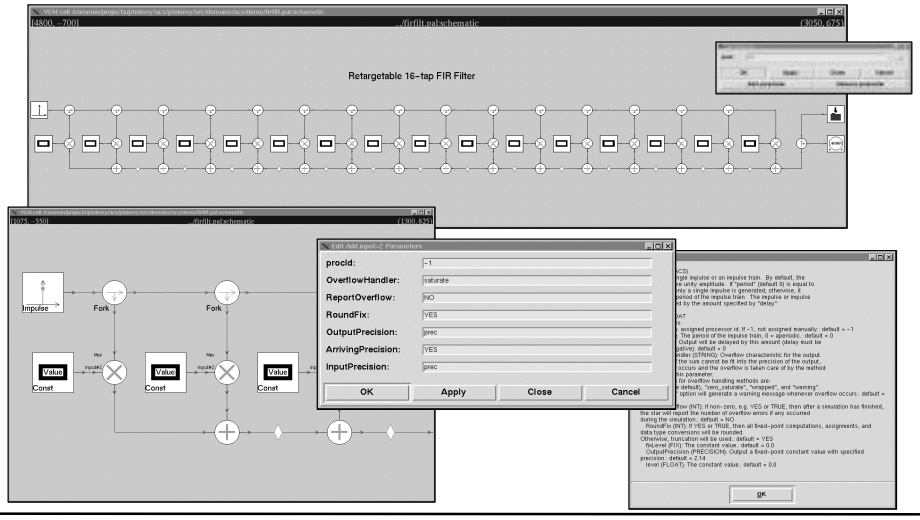
Ptolemy and the ACS Domain

- **Ptolemy - simulation/design environment from the University of California, Berkeley (http://ptolemy.eecs.berkeley.edu)**
- **New ACS domain developed to facilitate movement among simulation and code/design generation (released in 0.7.1, 6/98)**
- **ACS Stars (basic building block) are composed of a Corona (interface) and multiple cores (implementations)**
- **Core (implementation) selection is via targeting mechanism**



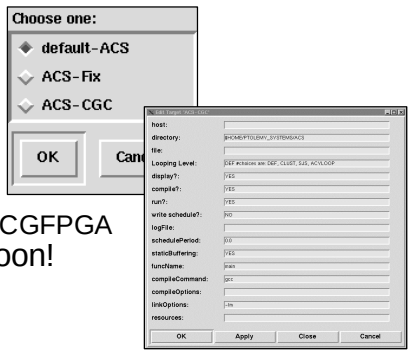
Top Level Example

- **FIR filter to be implemented in both floating point and fixed point simulation**

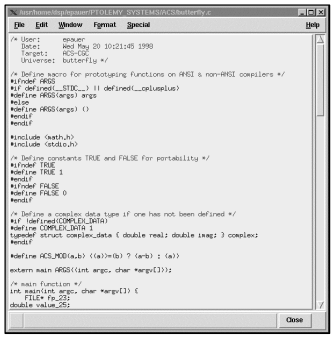


Selecting Among Alternative Implementations

- **Alternative implementations are represented as "targets"**
- **Targets can have parameters**
- **Floating point simulation, fixed point simulation, and C code generation are integrated today. FPGA generation almost ready.**

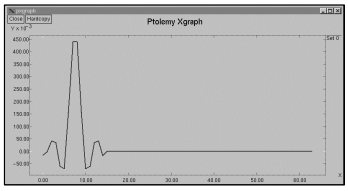


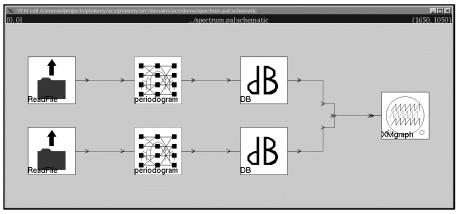
ACS-CGFPGA soon!

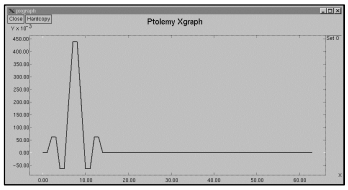


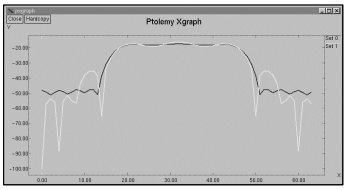
Comparing Implementations

- **Comparison of floating point and fixed point implementations**

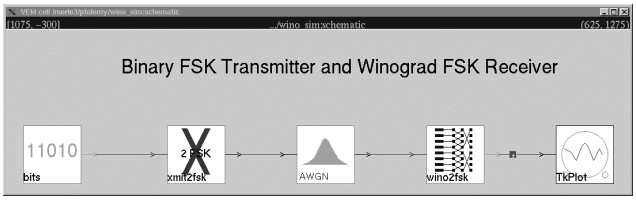


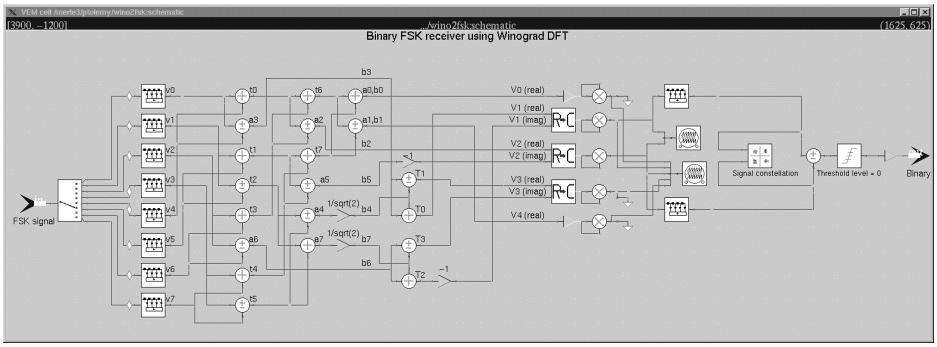






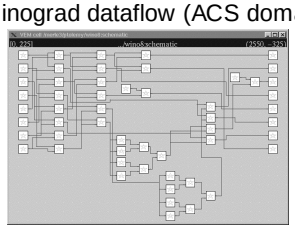
Winograd-based FSK Receiver





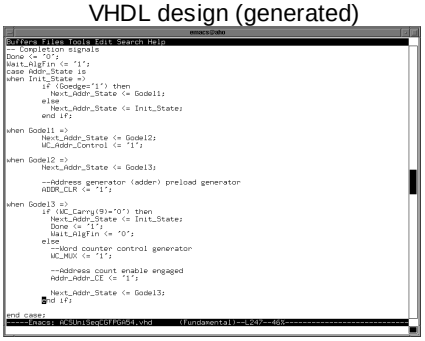
ACS Domain - CGFPGA Target

Winograd dataflow (ACS domain)



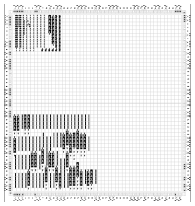
CGFPGA target yields: VHDL design and schedule

VHDL design (generated)

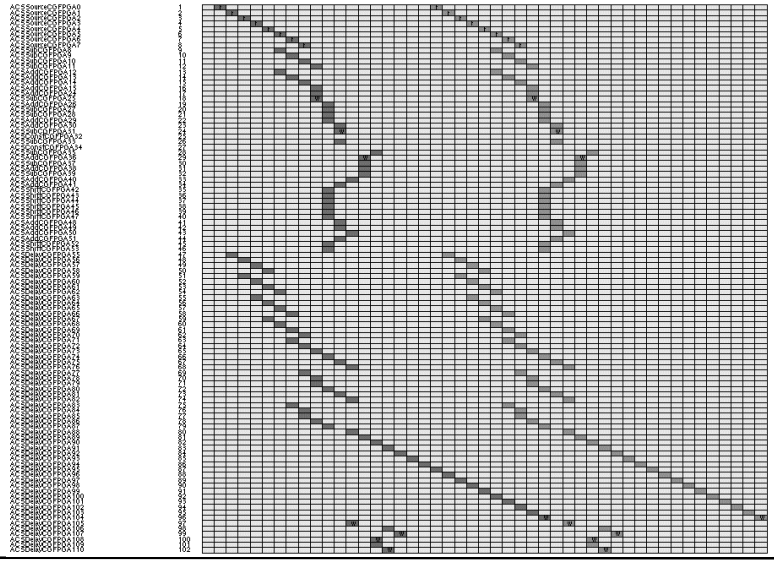


Dataflow/Hardware schedule

The results are sent to synthesis and place/route, yielding complete FPGA implementation!

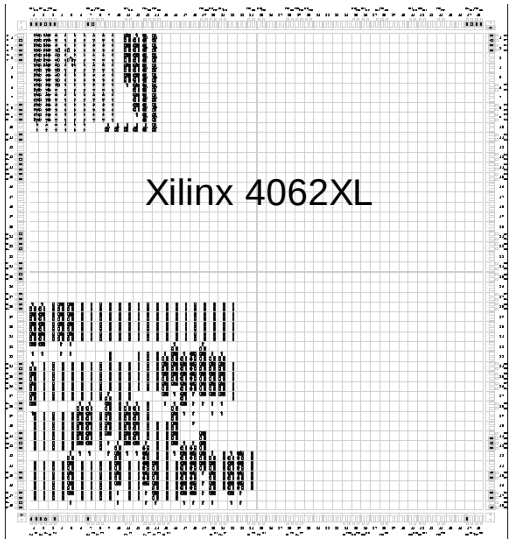


Winograd schedule



FPGA-based Winograd DFT

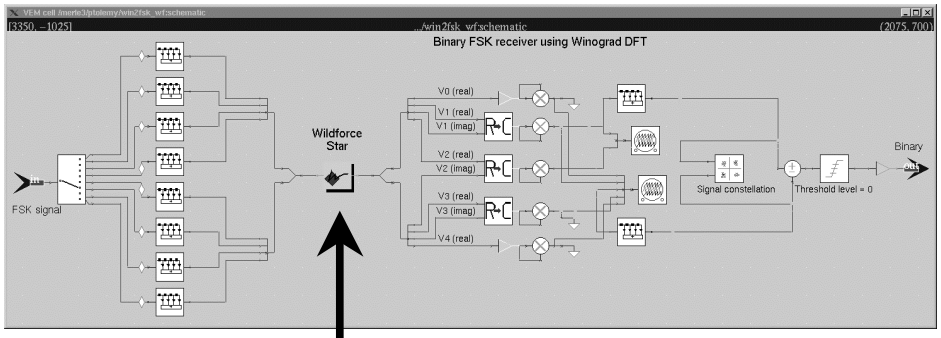
Address generator
Word counter
Data multiplexer
Sequencer/State Machine



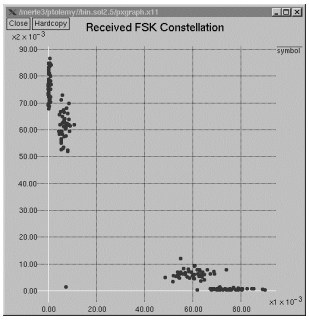
Winograd DFT computations

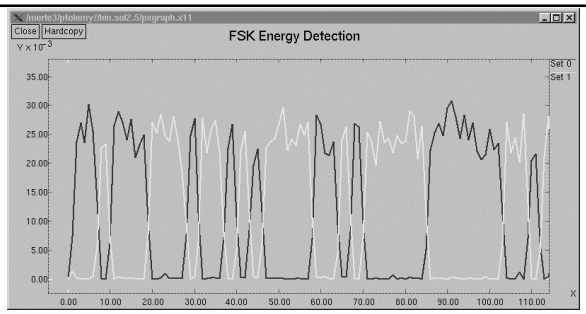
Hardware-in-the-loop

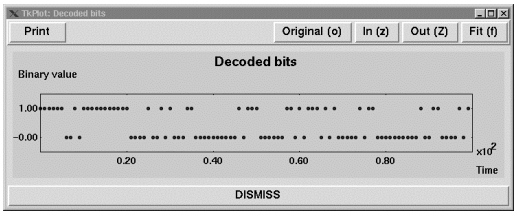
SDF Galaxy

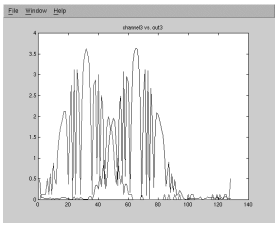


SDF Wildforce star executes complete FPGA design in hardware on Annapolis Wildforce FPGA board









Sim vs. Hardware

Related ACS Work at Sanders