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.

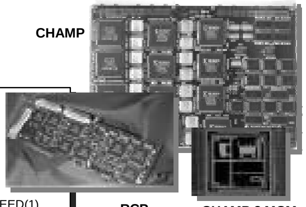
Adaptive Computing Performance Gain

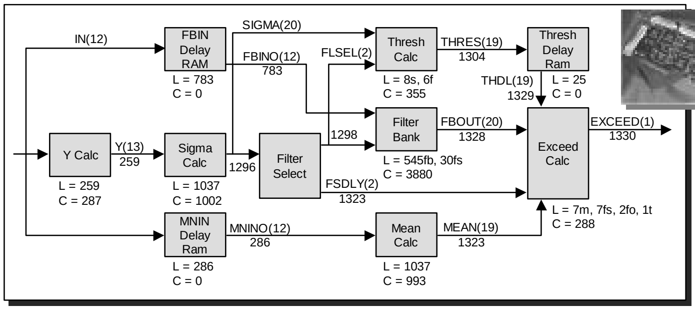
	CHAMP	TMS 320C80
Image Size	256 x 256	256 x 256
Implementation Time	44 Days	28 Days
Frame Rate	305 frames/sec	12 frames/sec
Latency	68 µsec	82,000 µsec
Processing Load	4.7 Bops	0.2 Bops
Utilization	73%	Unknown
Gates	510k	N/A

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

Greater than 10X performance
Design time measured in weeks

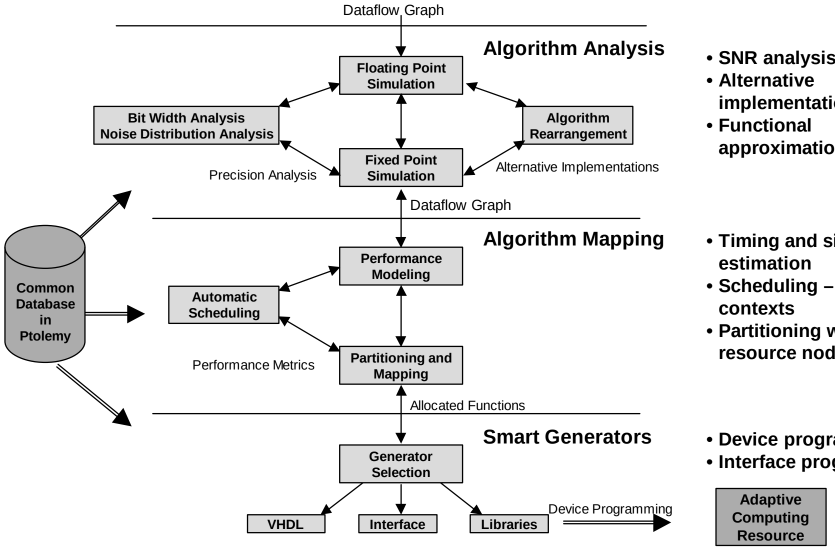
Reconfigurable Architectures





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

Analysis and Mapping in ACS Environment



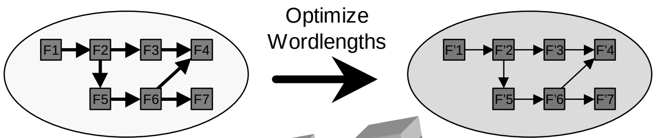
- **SNR analysis**
- **Alternative implementations**
- **Functional approximations**

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

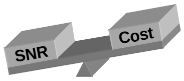
- **Device program**
- **Interface program**

Automated Float to Fixed Point Translation

Floating Point Simulation

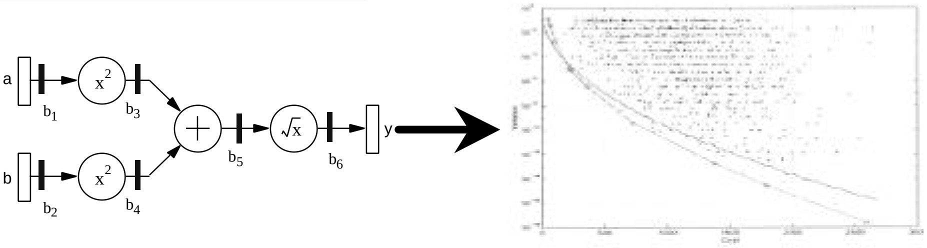


Fixed Point Realization



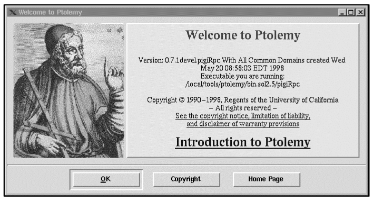
Maximize SNR(<u>b</u>),
subject to Cost(<u>b</u>) " C_0 and <u>b</u> " $\underline{b}_{min}$ " <u>0</u>
or
Minimize Cost(<u>b</u>),
subject to SNR(<u>b</u>) " SNR_0 and <u>b</u> " $\underline{b}_{min}$ " <u>0</u>

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**



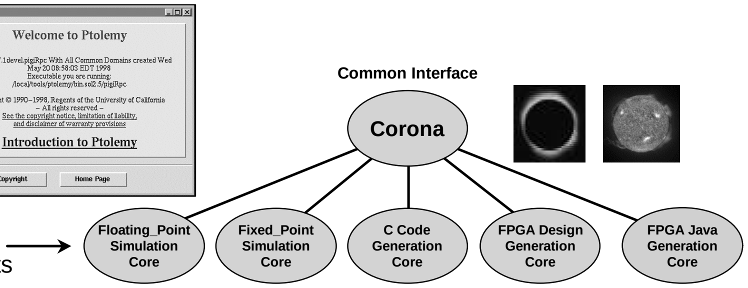
Welcome to Ptolemy

Version: 0.7.1devel.pigiRpc With All Common Domains created Wed May 20 08:58:03 EDT 1998
Executable you are running:
/local/tools/ptolemy/bin.sol2.5/pigiRpc

Copyright © 1990–1998, Regents of the University of California
– All rights reserved –
See the copyright notice, limitation of liability, and disclaimer of warranty provisions

Introduction to Ptolemy

Available cores/targets

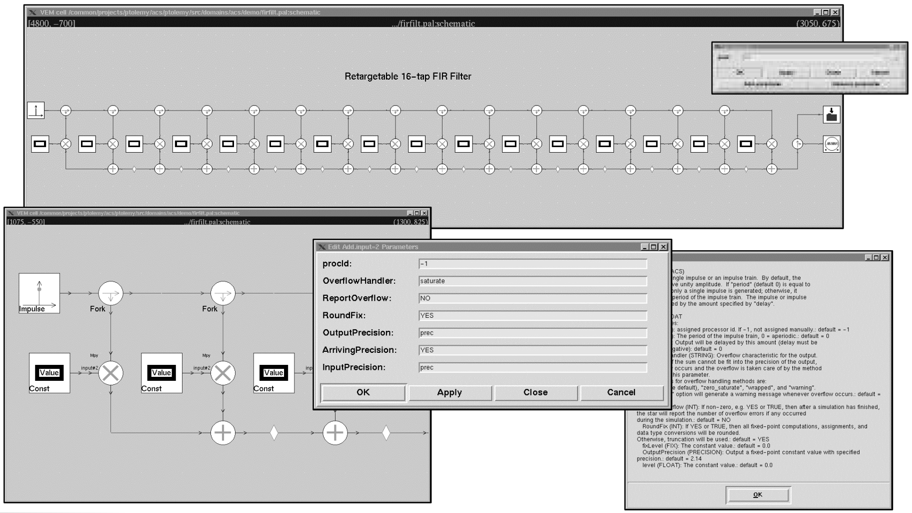

Common Interface
Corona
Floating_Point Simulation Core
Fixed_Point Simulation Core
C Code Generation Core
FPGA Design Generation Core
FPGA Java Generation Core

Retargetable Implementations

UCB BRASS Project

Top Level Example

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


VEM cell /common/projects/ptolemy/acs/ptolemy/src/domains/acs/demo/firfilt.pal:schematic
[4800, -700]
.../firfilt.pal:schematic
(3050, 675)
Retargetable 16-tap FIR Filter
VEM cell /common/projects/ptolemy/acs/ptolemy/src/domains/acs/demo/firfilt.pal:schematic
[1075, -550]
.../firfilt.pal:schematic
(1300, 825)
Impulse
Fork
Fork
Mpy
Mpy
input#2
input#2
Value
Const
Value
Const
Value
Const
Edit Add.input=2 Parameters
OK
Apply
Close
Cancel
OK

procId:	-1
OverflowHandler:	saturate
ReportOverflow:	NO
RoundFix:	YES
OutputPrecision:	prec
ArrivingPrecision:	YES
InputPrecision:	prec

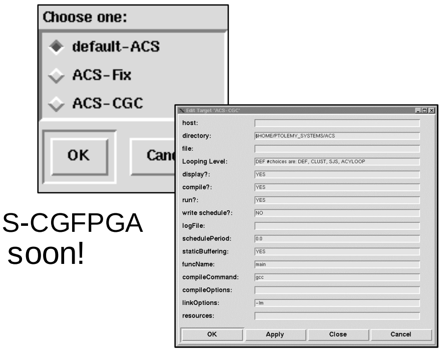
ACS)
ngle impulse or an impulse train. By default, the
ve unity amplitude. If "period" (default 0) is equal to
only a single impulse is generated; otherwise, it
period of the impulse train. The impulse or impulse
ed by the amount specified by "delay".

OAT
es:
): assigned processor id. If -1, not assigned manually.: default = -1
): The period of the impulse train, 0 = aperiodic.: default = 0
Output will be delayed by this amount (delay must be
gative): default = 0
andler (STRING): Overflow characteristic for the output.
f the sum cannot be fit into the precision of the output,
occurs and the overflow is taken care of by the method
his parameter.
s for overflow handling methods are:
e default), "zero_saturate", "wrapped", and "warning".
" option will generate a warning message whenever overflow occurs.: default =

rflow (INT): If non-zero, e.g. YES or TRUE, then after a simulation has finished,
the star will report the number of overflow errors if any occurred
during the simulation.: default = NO
RoundFix (INT): If YES or TRUE, then all fixed-point computations, assignments, and
data type conversions will be rounded.
Otherwise, truncation will be used.: default = YES
fixLevel (FIX): The constant value.: default = 0.0
OutputPrecision (PRECISION): Output a fixed-point constant value with specified
precision.: default = 2.14
level (FLOAT): The constant value.: default = 0.0

Selecting Among Alternative Implementations

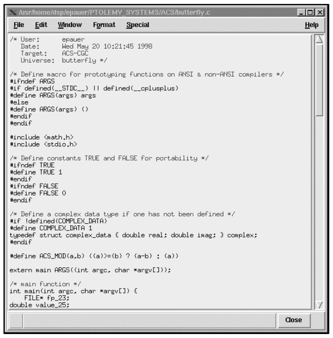
- **Alternative implementations are represented as "targets"**
- **Targets can have parameters**
- **Floating point simulation, fixed point simulation, and C code generation are integrated today. FPGA generation almost ready.**



● default-ACS
◇ ACS-Fix
◇ ACS-CGC

host:	
directory:	$HOME/PTOLEMY_SYSTEMS/ACS
file:	
Looping Level:	DEF #choices are: DEF, CLUST, SJS, ACYLOOP
display?:	YES
compile?:	YES
run?:	YES
write schedule?:	NO
logFile:	
schedulePeriod:	0.0
staticBuffering:	YES
funcName:	main
compileCommand:	gcc
compileOptions:	
linkOptions:	-lm
resources:	

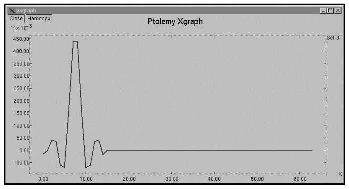
ACS-CGFPGA
soon!

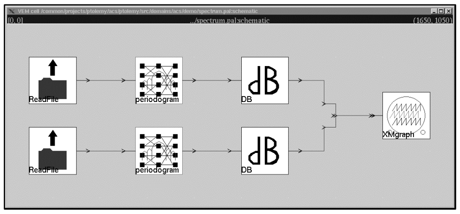


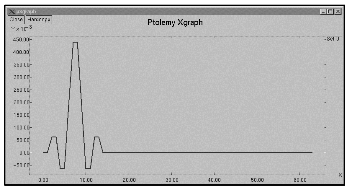
```
/* User:      epauer
   Date:      Wed May 20 10:21:45 1998
   Target:    ACS-CGC
   Universe:  butterfly */

/* Define macro for prototyping functions on ANSI & non-ANSI compilers */
#ifndef ARGS
#if defined(__STDC__) || defined(__cplusplus)
#define ARGS(args) args
#else
#define ARGS(args) ()
#endif
#endif

#include <math.h>
#include <stdio.h>

/* Define constants TRUE and FALSE for portability */
#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif

/* Define a complex data type if one has not been defined */
#if !defined(COMPLEX_DATA)
#define COMPLEX_DATA 1
typedef struct complex_data { double real; double imag; } complex;
#endif

#define ACS_MOD(a,b) ((a)>=(b) ? (a-b) : (a))

extern main ARGS((int argc, char *argv[]));

/* main function */
int main(int argc, char *argv[]) {
    FILE* fp_23;
double value_25;
```

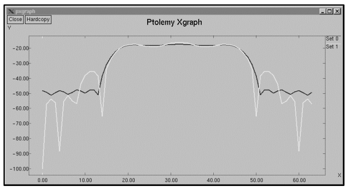
Comparing Implementations

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

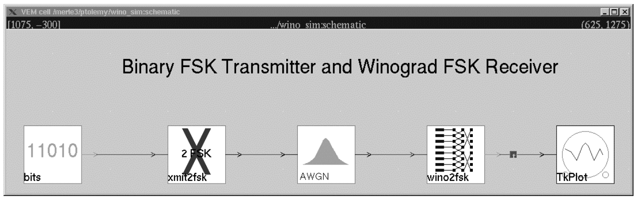


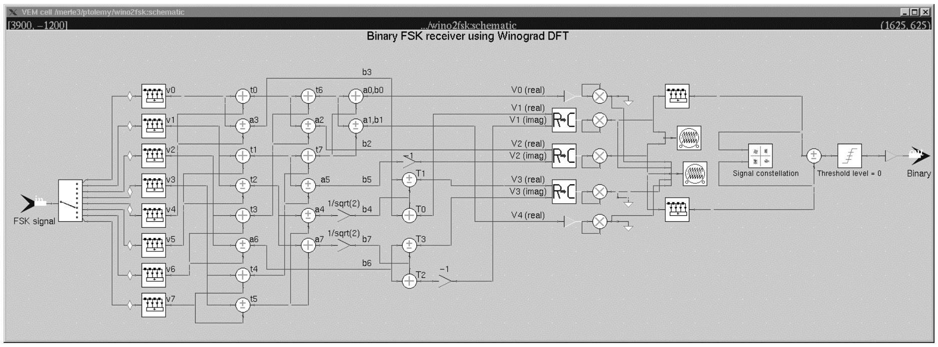

VEM cell /common/projects/ptolemy/acs/ptolemy/src/domains/acs/demo/spectrum.pal:schematic
[0, 0]
.../spectrum.pal:schematic
(1650, 1050)
ReadFile
periodogram
DB
XMgraph
ReadFile
periodogram
DB





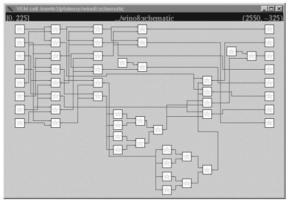
Winograd-based FSK Receiver


VEM cell /merle3/ptolemy/wino_sim:schematic
[1075, -300]
.../wino_sim:schematic
(625, 1275)
Binary FSK Transmitter and Winograd FSK Receiver
11010
bits
2 FSK
xmit2fsk
AWGN
wino2fsk
TkPlot

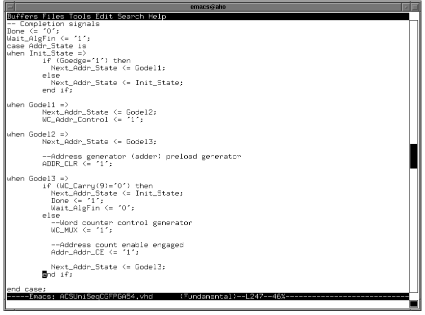


ACS Domain - CGFPGA Target

Winograd dataflow (ACS domain)

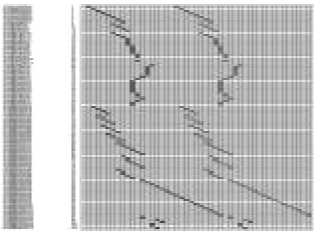


VHDL design (generated)



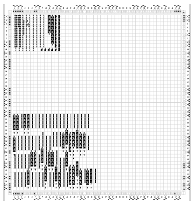
```
-- Completion signals
Done <= '0';
Wait_AlgFin <= '1';
case Addr_State is
when Init_State =>
        if (Goedge='1') then
          Next_Addr_State <= Godel1;
        else
          Next_Addr_State <= Init_State;
        end if;

when Godel1 =>
        Next_Addr_State <= Godel2;
        WC_Addr_Control <= '1';

when Godel2 =>
        Next_Addr_State <= Godel3;

        --Address generator (adder) preload generator
        ADDR_CLR <= '1';

when Godel3 =>
        if (WC_Carry(9)='0') then
          Next_Addr_State <= Init_State;
          Done <= '1';
          Wait_AlgFin <= '0';
        else
          --Word counter control generator
          WC_MUX <= '1';

          --Address count enable engaged
          Addr_Addr_CE <= '1';

          Next_Addr_State <= Godel3;
        end if;

end case;
-----Emacs: ACSUniSeqCGFPGA54.vhd      (Fundamental)--L247--46%------------------------
```

CGFPGA target yields: VHDL design and schedule

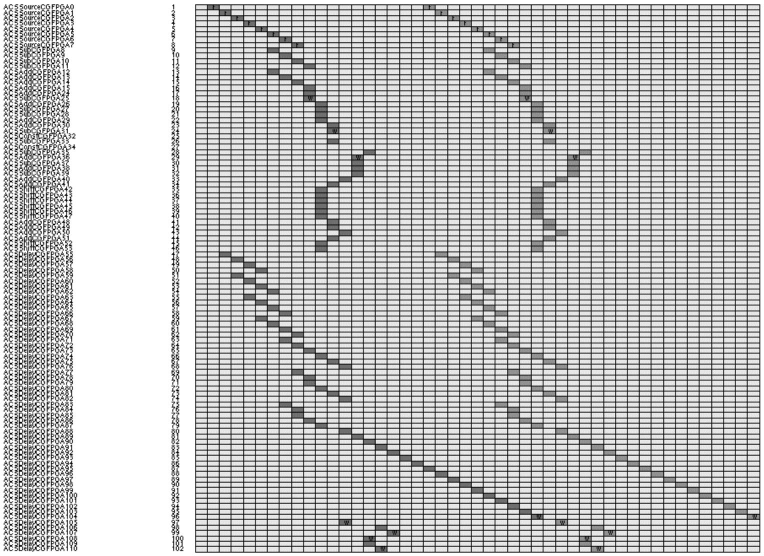


Dataflow/Hardware schedule

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



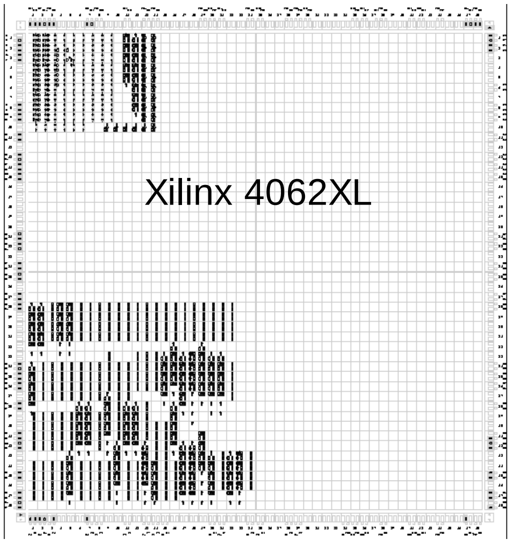
Winograd schedule



FPGA-based Winograd DFT

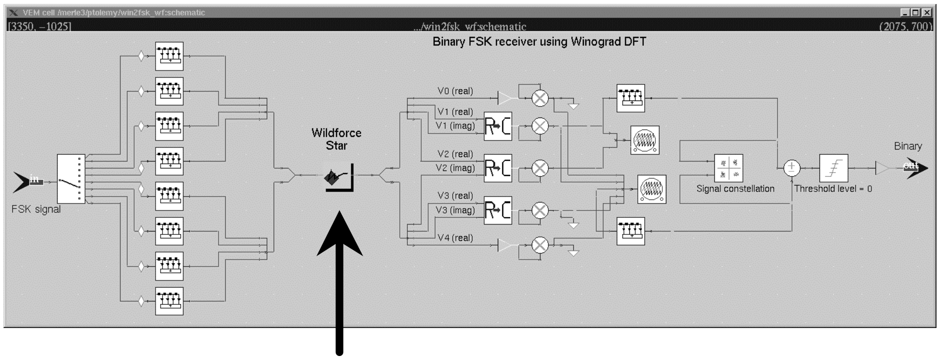
Address generator
Word counter
Data multiplexer
Sequencer/State Machine

Winograd DFT
computations

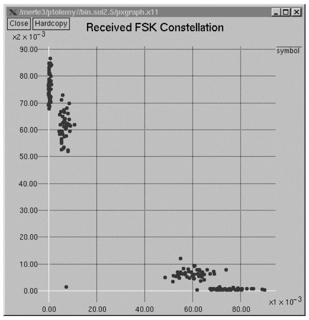


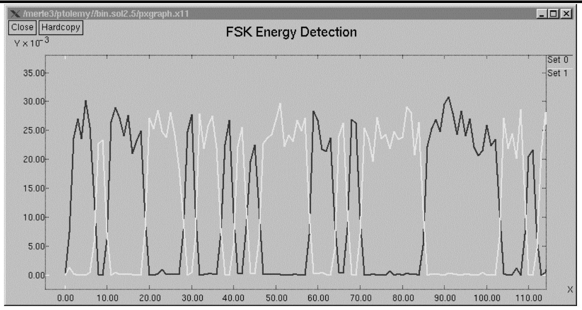
Hardware-in-the-loop

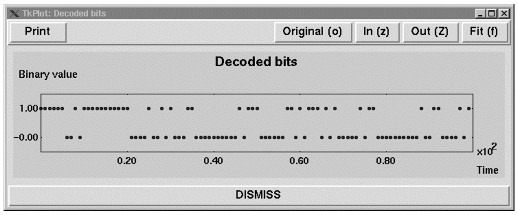
SDF Galaxy

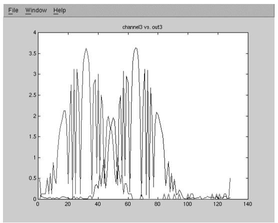


SDF Wildforce star executes complete FPGA design in hardware on Annapolis Wildforce FPGA board



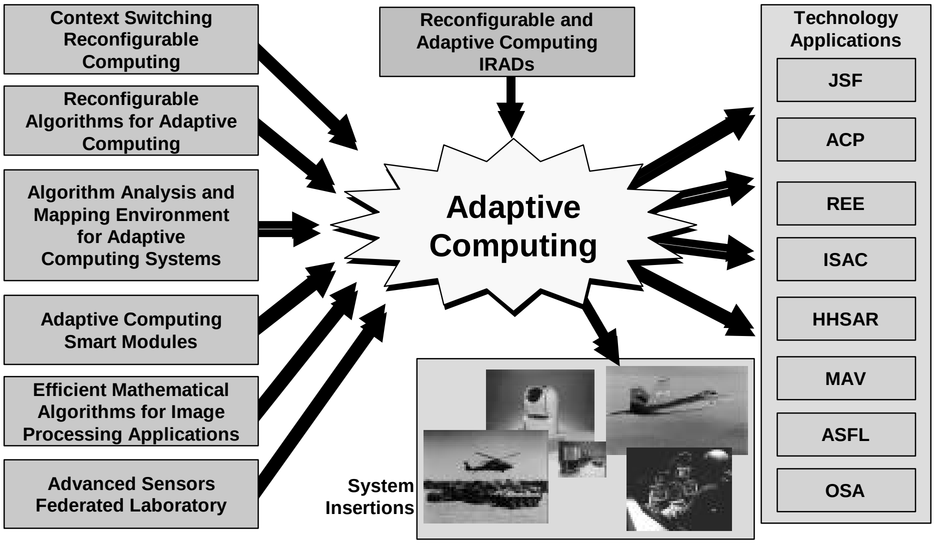






Sim vs. Hardware

Related ACS Work at Sanders


Context Switching Reconfigurable Computing
Reconfigurable Algorithms for Adaptive Computing
Algorithm Analysis and Mapping Environment for Adaptive Computing Systems
Adaptive Computing Smart Modules
Efficient Mathematical Algorithms for Image Processing Applications
Advanced Sensors Federated Laboratory
Reconfigurable and Adaptive Computing IRADs
Adaptive Computing
System Insertions
Technology Applications
JSF
ACP
REE
ISAC
HHSAR
MAV
ASFL
OSA