February 16, 2011
Berkeley, CA

Ptolemy Miniconference

Automated Fixed-Point Analysis in Ptolemy

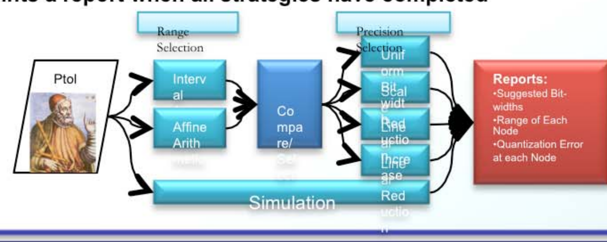
Derrick Gibelyou
Michael Wirthlin

Motivation

- Using a uniform bit-width for FPGAs is inefficient
  - Uniform bit-width selection is useful for DSP and other fixed-width systems
  - Doesn't take advantage of the flexibility of FPGAs
  - Finding the optimal bit-width reduces area and increases clock rate while maintaining the quality of the answer
- Finding the optimal bit-width is difficult
- Many techniques exist for finding near-optimal bit-widths:
  - Statistical Simulation
  - Feed-forward Heuristics
  - SAT / ILP solver
- Ptolemy provides a good infrastructure on which to implement these algorithms

New Ptolemy Director

- Based on SDF director
- Ends when all strategies are finished
  - Strategies are like sub-directors
- Director runs strategies to find range, then runs strategies to find precision
- Prints a report when all strategies have completed

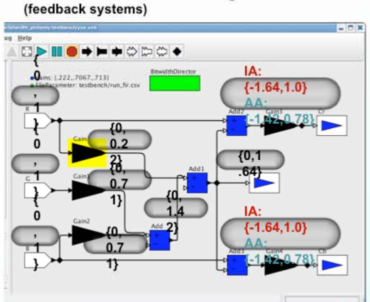


New Tokens

- Implemented new Range Tokens and Simulation Tokens
- Range Tokens inherit from ScalarToken
  - Allows math with ranges and constants
  - Although represented by more than one number, a range token should be treated as a scalar
- Error Tokens
  - Holds two range tokens:
    - Dynamic range
    - Range of Quantization Error
  - Still represents a scalar entity

Range Algorithms

- Interval Arithmetic
  - Simple method for calculating range:
  - X=[-1,1], Y=[-2,2]; X + Y= [-3,3]
  - Cannot be used in feedback systems
- Affine Arithmetic:
  - Accounts for correlations between the inputs, e.g.
    - $X \in [0..1]$
    - Interval Arithmetic $X - X \in [-1..1]$
    - Affine Arithmetic $X - X \in [0]$
  - Can be used to solve for the range of IIR filters (feedback systems)



Selected References
•R. Moore. Interval analysis. Englewood Cliffs, 1966.
•G. A. Constantinides. Word-length optimization for differentiable nonlinear systems. ACM Trans. Des. Autom. Electron. Syst., 11(1):26–43, 2006
•C. F. Fang, R. A. Rutenbar, and T. Chen. Fast, accurate static analysis for Fixed-Point Finite-Precision effects in DSP designs. In Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design, page 275. IEEE Computer Society, 2003.

Precision Algorithms

- Most published techniques involve heuristic competitions to find a near optimal bit-width
- Competition:
  - Once the range is found, the system error can be calculated
  - The winning operator in each iteration is the one that that both:
  - increases the error the least
  - decreases the area the most
  - Competition continues until user constraint can no longer be met

Competition Pseudo Code

```
While(system_error_constraint is met)
  foreach (operator)
      reduce operator width by 1 bit
      measure system error
      score = error * cost_function(operator)
      Save score
      restore operator width
   (Operator with min score).width = width-1
```

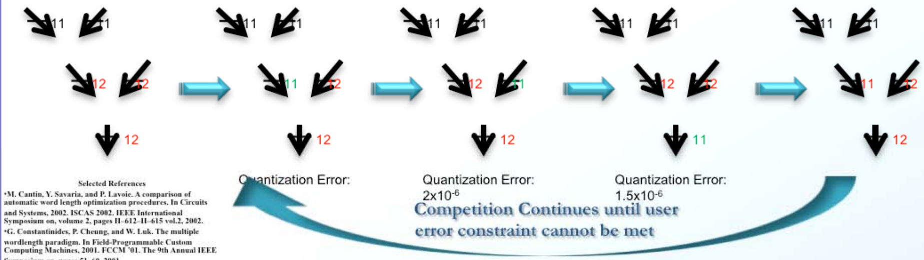
Competitions can begin by finding a uniform bit width

One operator is reduced, and the error measured

Another operator is reduced, and the error measured

Another operator is reduced, and the error measured

The operator introducing the lowest error is chosen



Competition Continues until user error constraint cannot be met

Selected References
•M. Cantin, Y. Savaria, and P. Lavoie. A comparison of automatic word length optimization procedures. In Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on, volume 2, pages II–612–II–615 vol.2, 2002.
•G. Constantinides, P. Cheung, and W. Luk. The multiple wordlength paradigm. In Field-Programmable Custom Computing Machines, 2001. FCCM '01. The 9th Annual IEEE Symposium on, pages 51–60, 2001.
•W. Osborne, R. Cheung, J. Coutinho, W. Luk, and O. Mencer. Automatic Accuracy-Guaranteed Bit-Width optimization for fixed and Floating-Point systems. In Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on, pages 617–620, 2007.

Results

- Several simple test benches have been created:
  - FIR and IIR filters
  - DCT
  - RGB to YUV converter
  - BPSK timing loop
- Results for BPSK timing loop closely match those of human selected values:

Circuit Portion	Operator	Hand Optimized	Program Results	Difference
farrow	Product1	10	6	4
farrow	Product2	10	5	5
farrow	Sum	10	6	4
farrow	Sum10	10	13	-3
farrow	Gain1	10	13	-3
farrow	Sum7	10	4	6
farrow	Sum5	10	6	4
farrow	Sum6	10	6	4
farrow	Sum3	10	6	4
farrow	Sum4	10	6	4
farrow	Sum1	10	7	3
farrow	Sum2	10	6	4
Loop Constant	Loop filter constant	11	1	10
Loop Filter	First Order Loop Filter	11	11	0
NCO	mod	11	12	-1
NCO	Gain	9	14	-5
NCO	Sum9	11	12	-1
NCO	Sum8	11	11	0
ZCTED	Product3	10	14	-4
ZCTED	Subtract1	10	14	-4

Work funded by CHREC
NSF Center for High-Performance Reconfigurable Computing