Precision Modeling and Bitwidth Optimization of Floating-Point Applications

Zhihong Zhao

Alternative System Concepts, Inc.

Windham, NH, USA

Miriam Leeser

Northeastern University

Boston, MA, USA

Outline

- Variable Bitwidth Computing
- Precision Modeling Methodology
  - Behavioral Profiling
  - Error Modeling
  - Verification
- Bitwidth Optimization
  - Problem Formulation
  - Optimization Algorithms
  - Optimization Results
- Future Work
- Conclusion

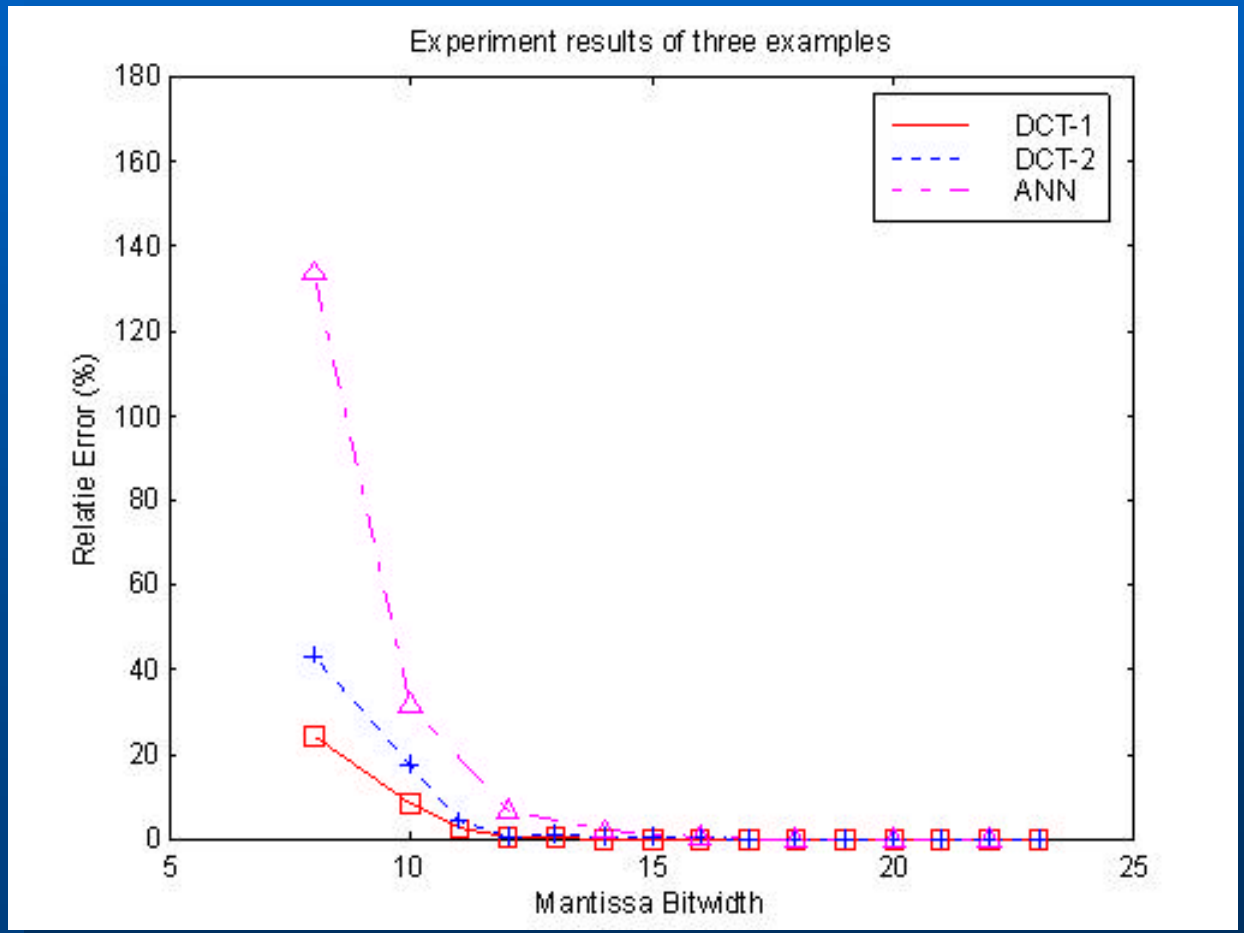
Variable Bitwith Computing

- Why Variable Bitwidth Computing ?
- Obtaining Optimized Bitwidths
  - Other's approach : simulation-based bitwidth searching
  - Our approach: model-based bitwidth optimization
  - System flowchart of the model-based bitwidth optimization

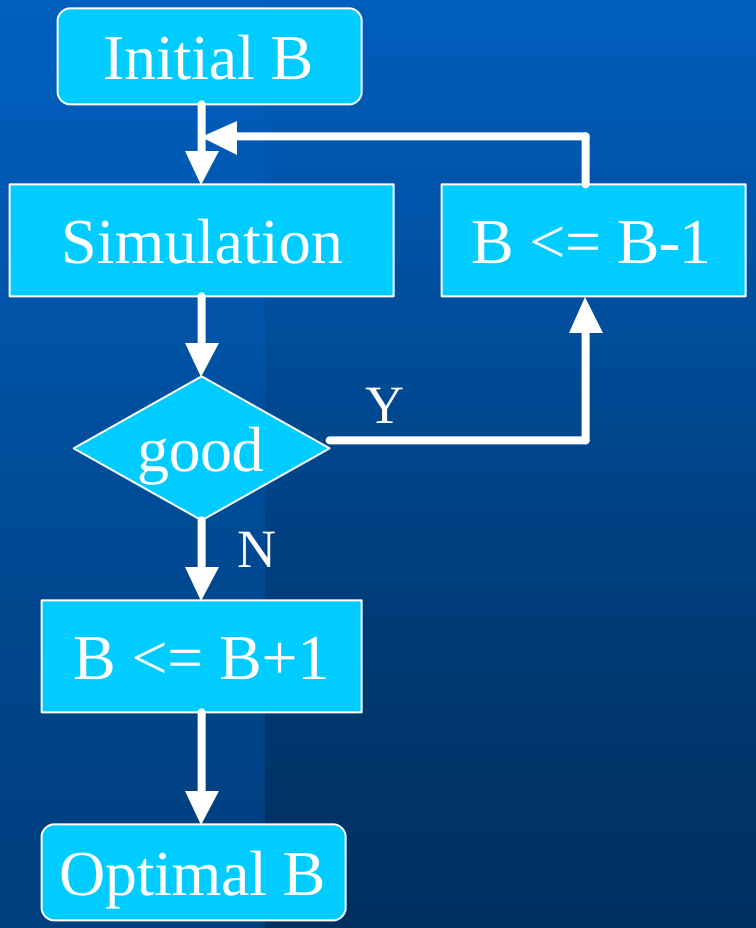
Why Variable Bitwith Computing ?

- Standard FP representation (Single Precision): 32 bits
  - sign: 1 bit
  - exponent: 8 bits
  - mantissa: 23 bits
- Implementation of standard FP operations in custom circuits is expensive
- Mantissa bitwidth can be reduced without compromising precision requirements

Relative Error VS Bitwidth

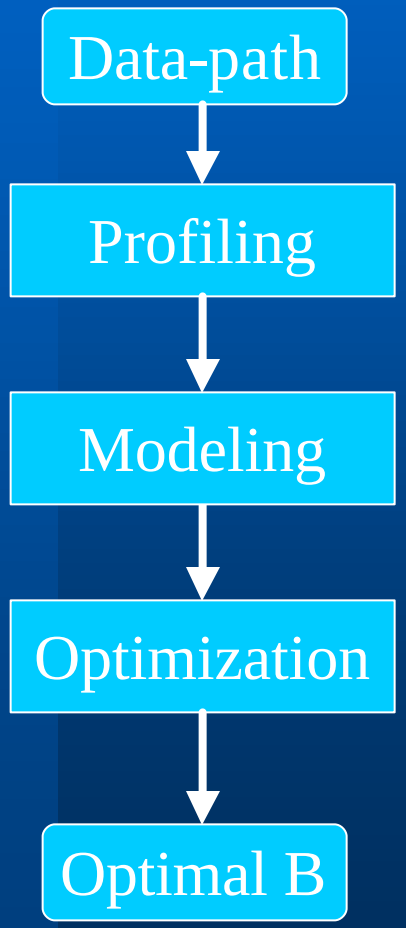


Simulation-based Searching



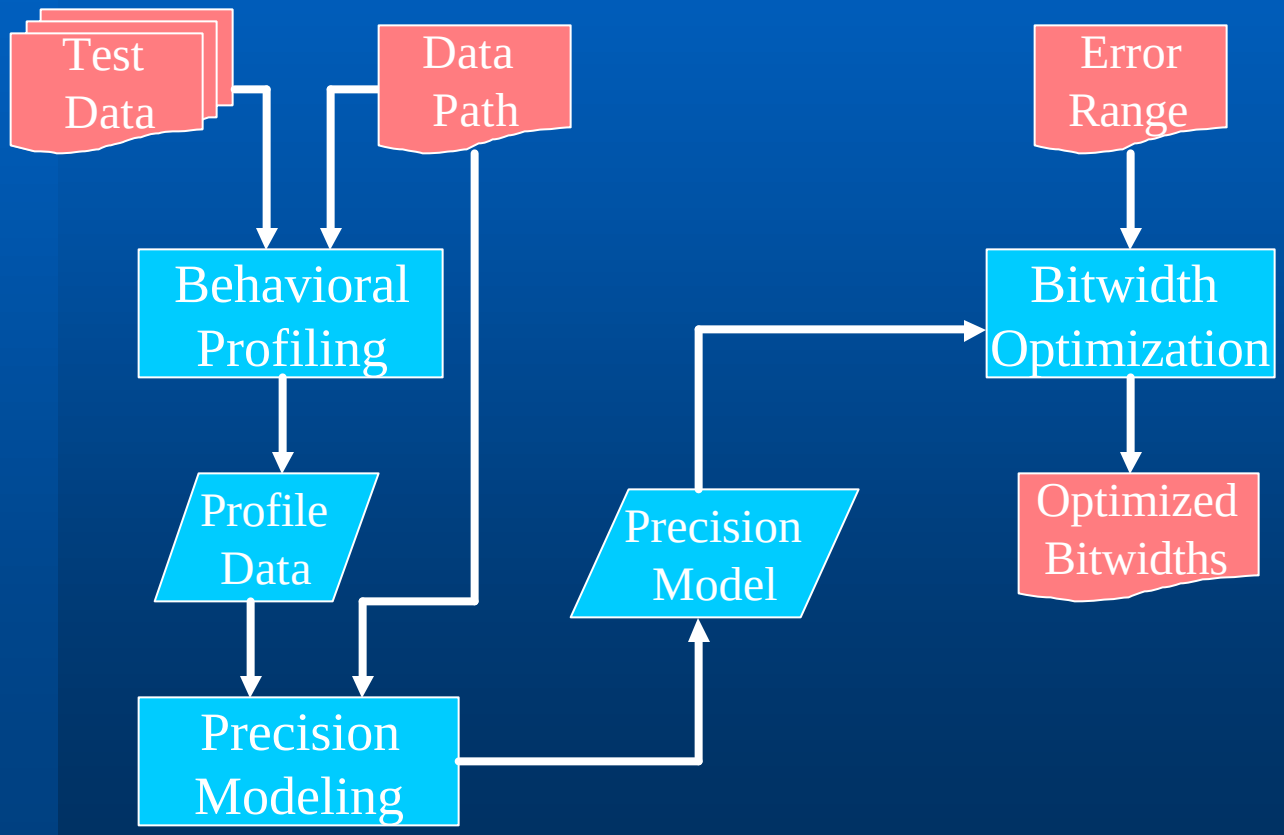
- Iterative simulation
- No explicit arithmetic precision model generated or needed
- Time consuming

Model-based Optimization



- No iterative simulation
- Application-specific arithmetic precision model generated
- One bitwidth value for each operation node

System Flowchart



Precision Modeling Methodology

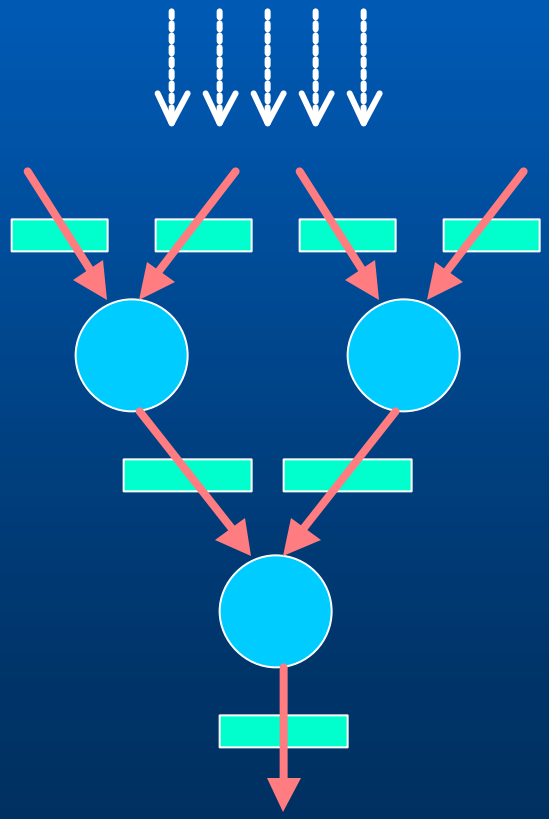
- Behavioral Profiling
- Error Models of FP Operations
  - Rounding Error
  - Propagation Error
- Constructing Precision Model

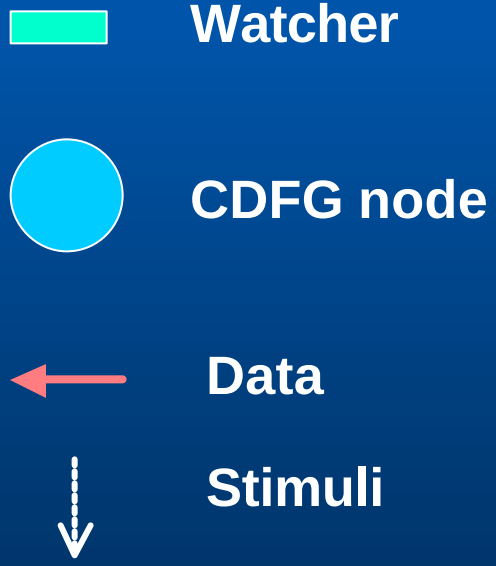
Profile-Driven Modeling

- Behavioral profiling gathers profile data through one-time simulation
  - bit probability
  - statistical values of variables in data-path
- Profiling is performed on a graphical representation (usually CDFG) of the application
- Profile data is used in precision modeling
- Selecting stimuli is important

Behavioral Profiling

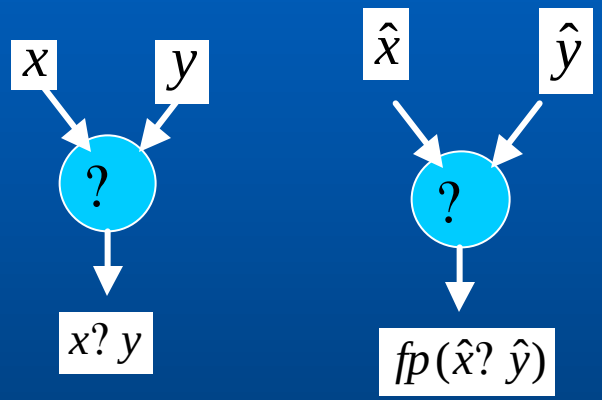
- Watcher insertion & simulation





Error in Floating-Point Model

- The overall error of a FP operation:



$$(x?\ y)\ ?\ \ fp(\hat{x}?\ \hat{y})\ ?\ (x?\ y\ ?\ \hat{x}?\ \hat{y})\ ?\ (\hat{x}?\ \hat{y}\ ?\ \ fp(\hat{x}?\ \hat{y}))$$

- Overall error at the result of an operation :
  Propagation Error(PE) + Rounding Error(RE)

Propagation Error (1)

- Calculation of PE

$$f(x,y) - f(\hat{x},\hat{y}) \approx \frac{\partial f(\hat{x},\hat{y})}{\partial x}(x-\hat{x}) + \frac{\partial f(\hat{x},\hat{y})}{\partial y}(y-\hat{y})$$

$$\varepsilon_f = \frac{|f(x,y) - f(\hat{x},\hat{y})|}{f(x,y)} \approx \frac{\frac{\partial f(\hat{x},\hat{y})}{\partial x}\hat{x}}{f(\hat{x},\hat{y})}\varepsilon_x + \frac{\frac{\partial f(\hat{x},\hat{y})}{\partial y}\hat{y}}{f(\hat{x},\hat{y})}\varepsilon_y \le k_x\varepsilon_x + k_y\varepsilon_y$$

- ***K*** **is the amplification factor, determined based on the operation's type and data**

Propagation Error (2)

Equation:

$$?_{z\_p} ? \frac{|f(x,y) ? f(\hat{x},\hat{y})|}{f(x,y)} ? \frac{f'_x(\hat{x},\hat{y})\hat{x}}{f(\hat{x},\hat{y})}?_x ? \frac{f'_y(\hat{x},\hat{y})}{f(\hat{x},\hat{y})}?_y ? k_x?_x ? k_y?_y$$

MULT:

$$k_x ? \frac{f'_x(\hat{x},\hat{y})\hat{x}}{f(\hat{x},\hat{y})} ? \frac{\hat{x}\hat{y}}{\hat{x}\hat{y}} ? 1.0$$

$$k_y ? \frac{f'_y(\hat{x},\hat{y})\hat{y}}{f(\hat{x},\hat{y})} ? \frac{\hat{x}\hat{y}}{\hat{x}\hat{y}} ? 1.0$$

ADD:

$$k_x ? \frac{f'_x(\hat{x},\hat{y})\hat{x}}{f(\hat{x},\hat{y})} ? \frac{\hat{y}}{\hat{x} ? \hat{y}}$$

$$k_y ? \frac{f'_y(\hat{x},\hat{y})\hat{y}}{f(\hat{x},\hat{y})} ? \frac{\hat{x}}{\hat{x} ? \hat{y}}$$

SQRT:

$$k_x ? \frac{f'(\hat{x})\hat{x}}{f(\hat{x})} ? 0.5$$

Rounding Error

Mantissa:

Precise value:

$$z = (1.0 + a_1 2^{-1} + a_2 2^{-2} + \cdots + a_b 2^{-b} + a_{b+1} 2^{-b-1} + \cdots + a_{22} 2^{-22} + a_{23} 2^{-23}) \times 2^e$$

FP value:

$$\hat{z} = (1.0 + a_1 2^{-1} + a_2 2^{-2} + \cdots + a_b 2^{-b}) \times 2^e$$

Error:

$$-\varepsilon_{z\_r} = \frac{z - \hat{z}}{z} = \frac{(a_{b+1} 2^{-b-1} + \cdots + a_{22} 2^{-22} + a_{23} 2^{-23})}{(1.0 + a_1 2^{-1} + a_2 2^{-2} + \cdots + a_{23} 2^{-23})} - \frac{p_{b+1} 2^{-b-1} + \cdots + p_{22} 2^{-22} + p_{23} 2^{-23}}{1.0 + p_1 2^{-1} + p_2 2^{-2} + \cdots + p_{23} 2^{-23}}$$

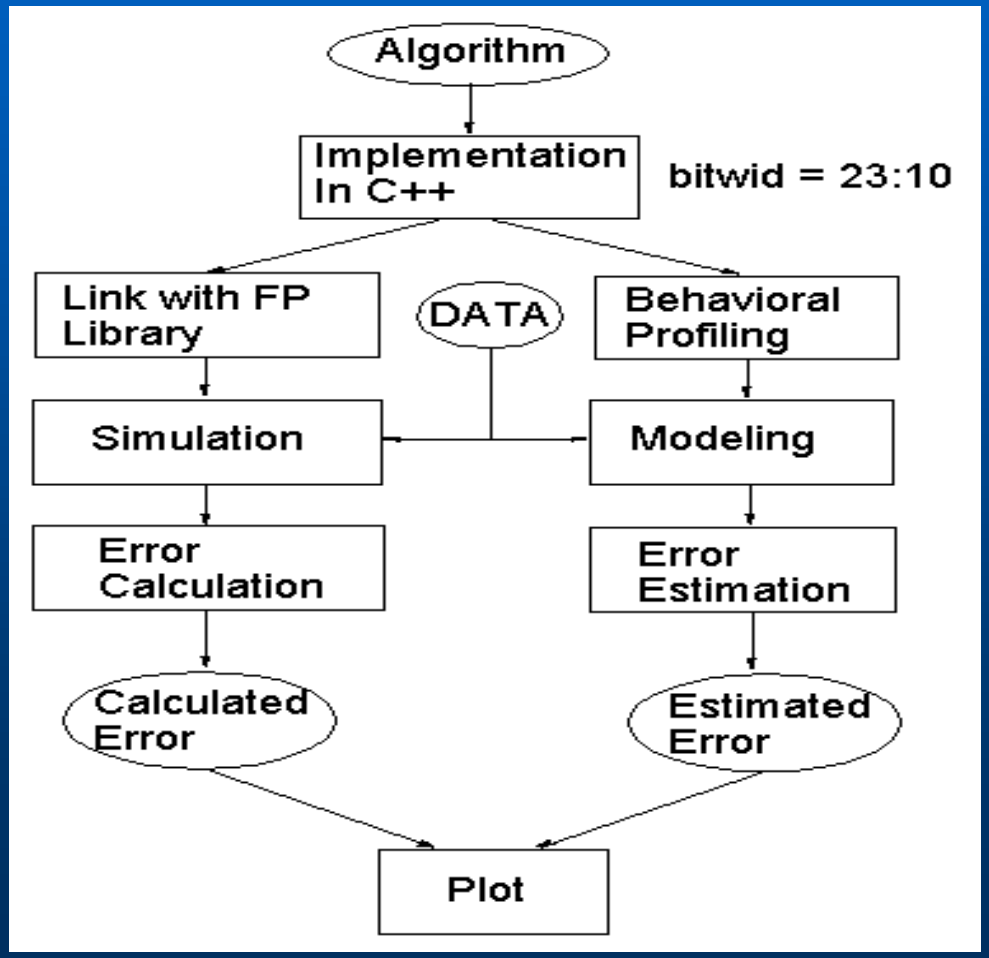
Constructing Precision Model

- Establish error model for each node in data-path using the RE+PE model
- Construct precision model for the application based-on data-path structure
- The precision model is a function of output error of the application in terms of bit-widths in data-path and input error of the application
- The precision model can be used to predict output error and optimize bitwidths

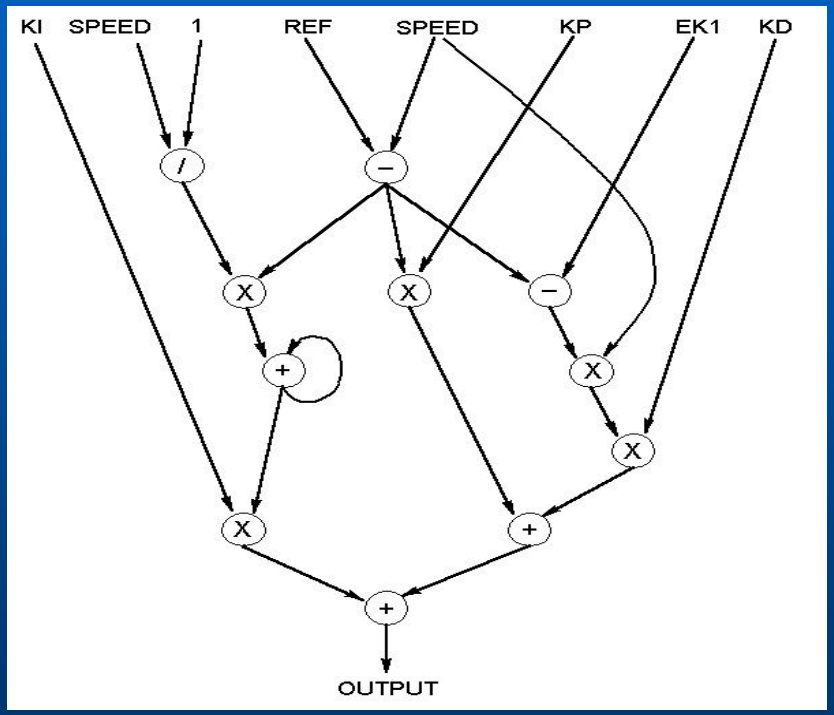
Experimental Results

- Experimental Procedure
- Example Data-paths (CDFGs)
- Comparison of Predicted Error Range and Actual Errors

Experimental Procedure

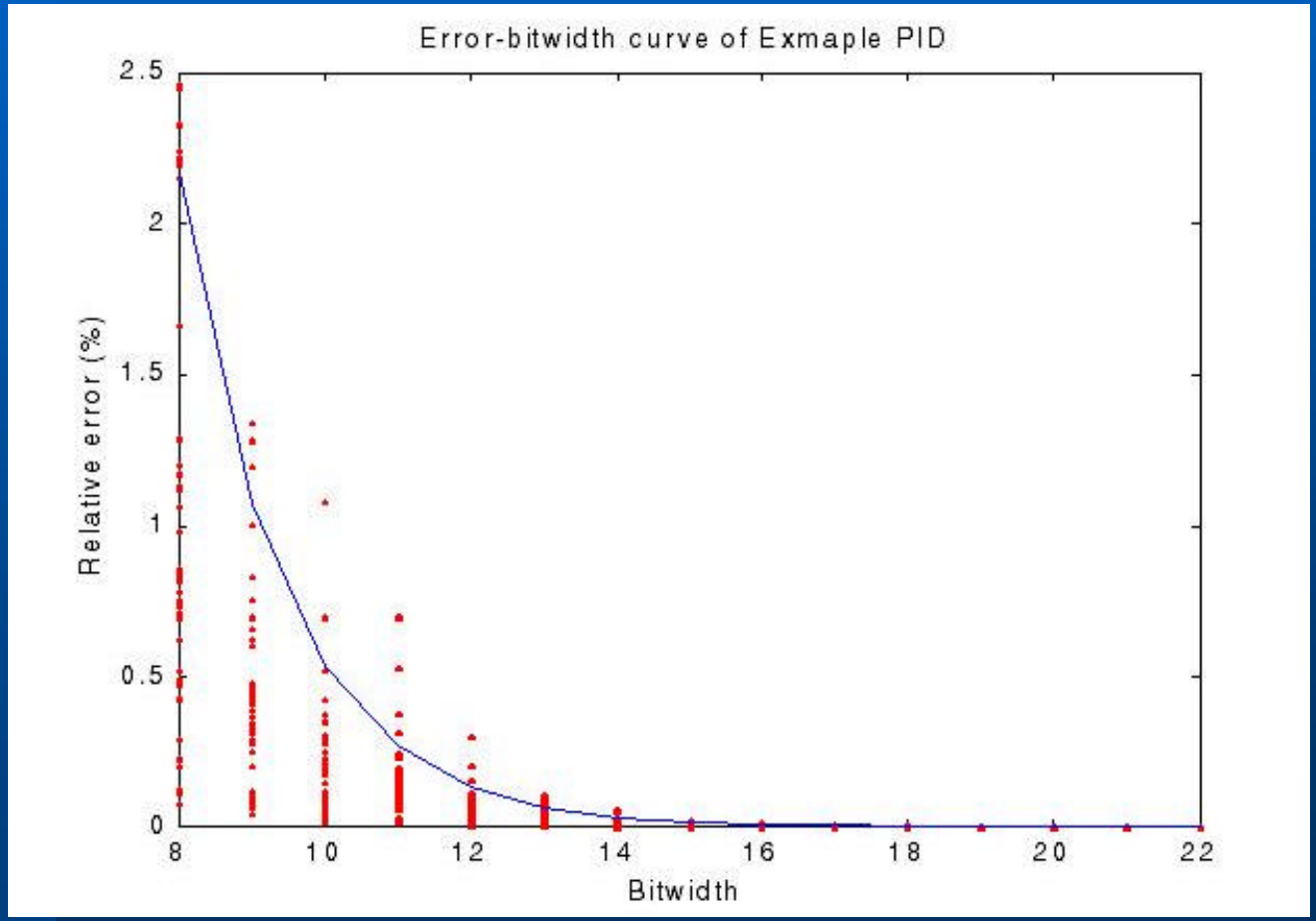


CDFG: PID Controller



$$Irefk ?? (Kp * Ek) ? Ki * ? (Ek * ?T) ? Kd * ? (Ek) / ?T$$

Result: PID Controller



Bitwidth Optimization

- Problem Formulation
- Optimization Methods
  - Grid Steepest Descent (GDS)
  - Accelerated Grid Steepest Descent (GSD-A)
- Optimization Results

Problem Formulation

- **Precision model based on CDFG and profile data:**

$$F = f(b_1, b_2, \ldots, b_N)$$

N: number of nodes in CDFG

- **Objective function:**

$$\min \sum_{1}^{N} b_i$$

b_i: bit-width of node *i*

- **Constraints:**

$$\begin{cases} F \geq P \\ b_i \geq \end{cases}$$

P: accuracy requirements

$$Z \in [1,2,\ldots,23]$$

Z: range for bit-width selection

Optimization Algorithms (1)

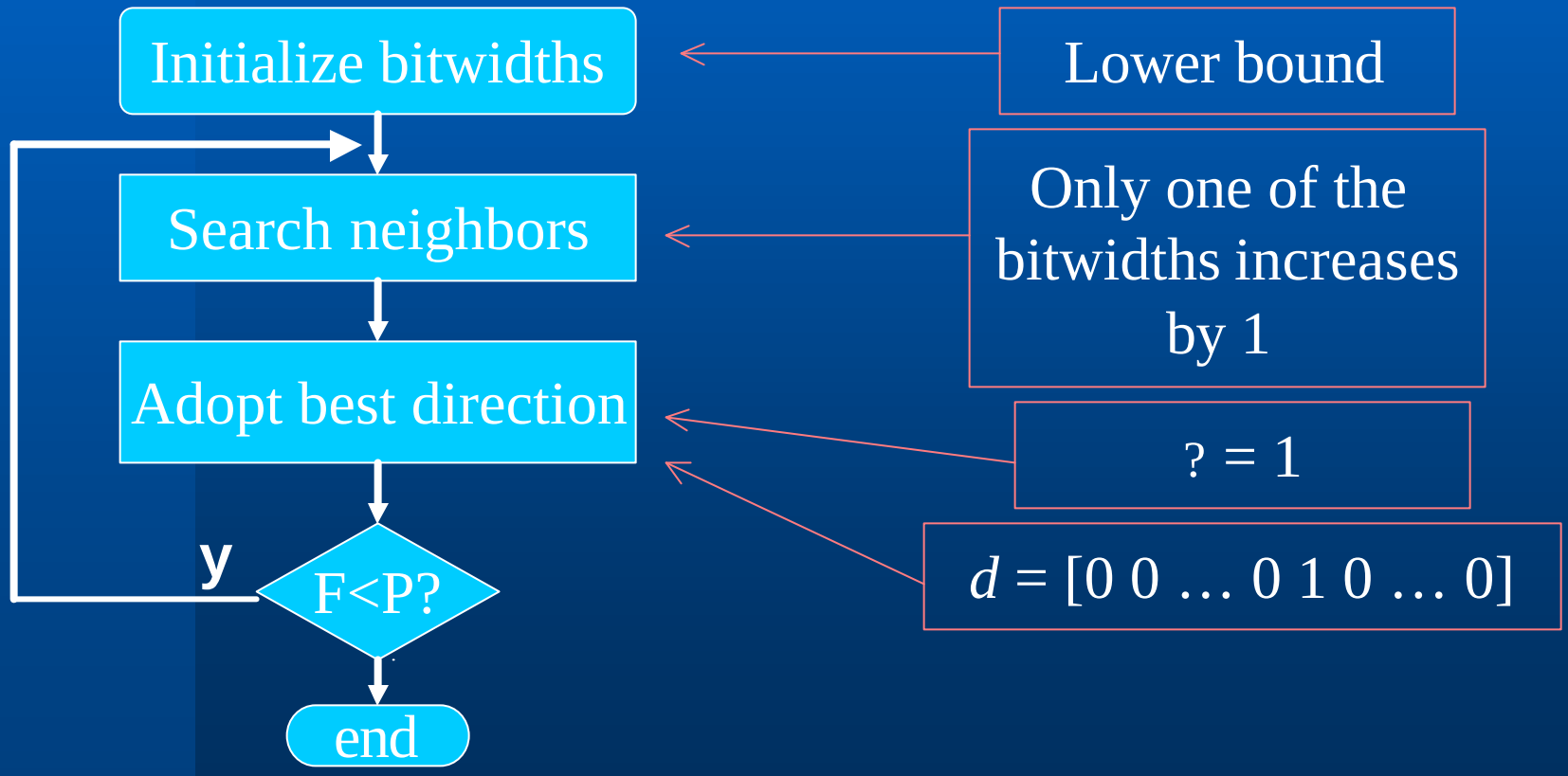
$$\begin{cases} \min f(x) \\ x_{k+1} = x_k + \lambda_k d_k \\ \lambda_k \geq 0 \end{cases}$$

x: vector of bitwidths
k: search step

- **Regular Steepest Decent**
  - **In each step, direction is calculated, step length is determined by searching in the direction**
- **Grid Steepest Decent (GSD)**
  - **In each step, step length is fixed (?=1), direction is determined by searching**

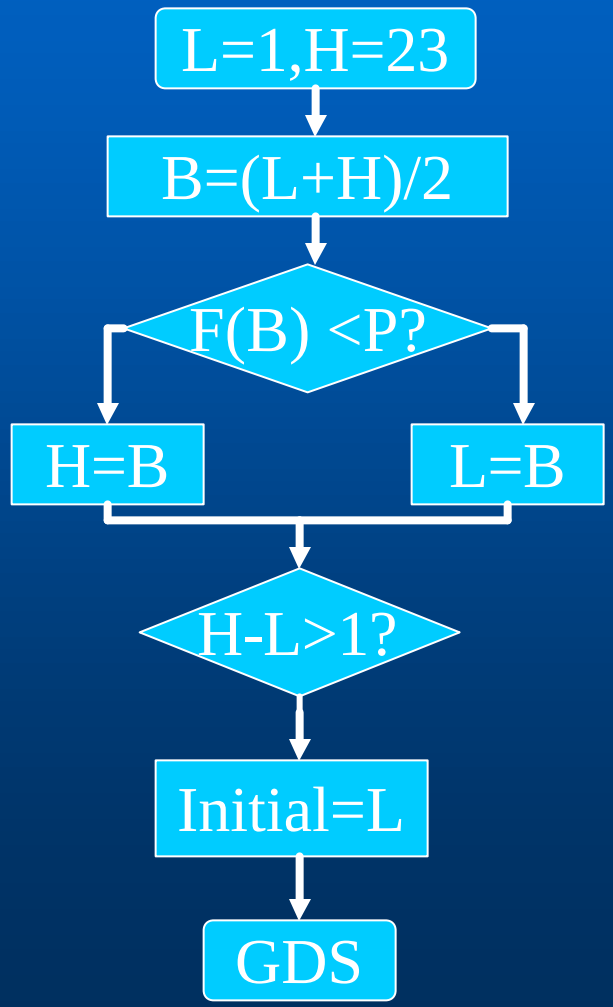
Optimization Algorithms (2)

- **Grid Steepest Decent (GSD)**



Optimization Algorithms (3)

- Accelerated GSD (GSD-A)
  - "Smart" Initial Point
  - Binary search to locate initial point
  - Total search time is reduced to a fraction
  - Initial Point: All bitwidths have the same initial value
  - GSD: Each bitwidth is calculated individually



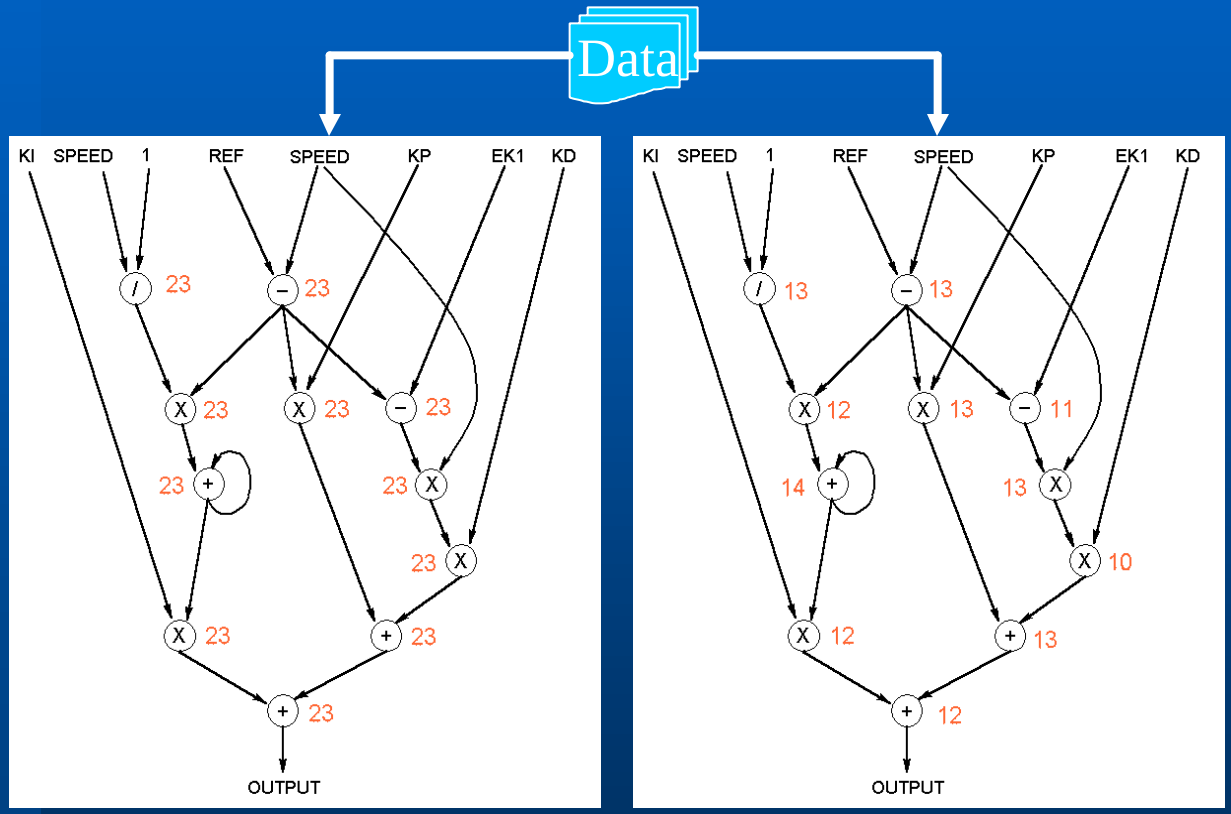
Optimization Results (1)

Result Comparison (P = 5%)

EXAMPLE	IEEE BITWIDTH	OPTIMIZED BITWIDTH	RGSD STEPS	RGSD-A STEPS
DIFFEQ	230	132	98	9
PID	253	136	87	8
Three MULT	96	37	59	2

Optimization Result (2)

- IEEE Format
- Total Bitwidth **253**
- Output Error **0**



- Variable Bitwidth
- Total Bitwidth **136**
- Output Error **5%**

Future Work

- Validate the optimized bitwidths
  - *C++ floating-point library* supporting variable bitwidth
  - *VHDL Variable precision floating-point component library*, developed by Rapid Prototyping Lab at Northeastern University, available under GPL at http://www.ece.neu.edu/groups/rpl/projects/floatingpoint/
- Improve error models
  - Propagation error models and rounding error models
  - Singularity issues
- Integrate in high level synthesis flow
  - IEEE 1076.3 working group: variable bitwidth floating-point for synthesis

Conclusion (1)

- Variable bitwidth FP computing is viable
- Model-based bitwidth optimization has advantages over simulation-based searching
- A methodology of FP precision modeling has been developed
- The precision model predicts output error and can be used for bit-width optimization

Conclusion (2)

- A customized optimization algorithm, Grid Steepest Descent (GSD), has been developed
- Search acceleration techniques have been applied to GSD
- Optimized bitwidths for a given precision target can be found quickly
- Sum of the optimized bitwidths is significantly smaller than that of standard IEEE format