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:
    - \( X \in \{1.1, Y \in \{2,2\} \} \times \{1.3\} \)
  - Cannot be used in feedback systems.
- Affine Arithmetic:
  - Account for correlations between the inputs, e.g. \( Y \in [1.1] \times \{1.3\} \)
  - Interval Arithmetic = \( Y \in [1,1] \times \{1,1\} \)
  - Affine Arithmetic = \( Y \in [1,1] \times \{1,1\} \)
  - Can be used to solve for the range of IIR filters (feedback systems).

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 decreases the area the most.
  - Competition continues until user constraint can no longer be met.

Results

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

<table>
<thead>
<tr>
<th>Score</th>
<th>Precision</th>
<th>Hand</th>
<th>Auto</th>
<th>Program</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Test 2</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Test 3</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Test 4</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

Work funded by

| CHREC | NSF Center for High-Performance Reconfigurable Computing |