FLOAT TO FIXED

Using Executable Models to Make FPGA Design Easier

Taylor L. Riché, Jim Nagle, Joyce Xu, Don Hubbard
National Instruments
Taylor L. Riché
Principal Product Owner
National Instruments
Manager
Local Arrangements Chair MODELS 2017
Developer
Postdoc (Riché et. al, MODELS 2010)
PhD Student
GOAL:

\[ f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \]

PROBLEM: THIS IS HARD!
SOLUTION:

1. Have algorithm designers model their new algorithms in LabVIEW NXG
2. Use the executable nature of LabVIEW NXG to generate an initial model transform
3. Provide tools to help designers create and apply the remaining transforms
4. Along the way, executability gives constant feedback on “correct enough” by construction
A QUICK INTRO TO G
SO THIS IS G...
AND THIS IS LABVIEW NXG...
A FEW DETAILS

1. Graphical dataflow language
2. Test and Measurement DSL
3. Allows you to model computation
4. Allows you to model hardware configuration
5. Different libraries of mathematical tools
6. Allows creation of EXEs and reusable IP
7. Maps computation to desktop, FPGA, and Realtime

http://www.ni.com/labviewnxg
EXECUTABILITY
IT RUNS
GOAL:

\[ f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \]
GOAL:
THE PROBLEM:

- Throughput constraints require HW
- FPGAs have limited resources
- Floating point takes many resources
- Digital design experts are expensive
THE RESOURCE SOLUTION: FIXED-POINT ARITHMETIC
FIXED POINT

XXXX.XYYYYYYYYYYYYYYYYYY

4 integer bits

12 fractional bits
FIXED POINT CHALLENGES

- Too few integer bits ➔ overflow
- Too few fractional bits ➔ loss of precision
- More bits use more FPGA resources
RESOURCES V. PRECISION

- Pareto Front For a Model and a Device
- Acceptable Region
RESOURCE EXAMPLE

Forward coefficients path

Reverse coefficients path

Filtered X

NATIONAL INSTRUMENTS
OUR SOLUTION:
LABVIEW NXG F2F
DESIGN TENANTS:

- Not trying to make the best F2F tool
- Usability was paramount
  - Don’t hire a FXP or DD expert
  - No spreadsheets!
- Don’t try to encode all constraints
  - Focus on Signal-to-Noise Ratio (SNR)
- Perfect is the enemy of shipping
BUT FIRST, SOME FORMALISMS
\[ M_{\text{golden}} \rightarrow M_{\text{FXP}} \]

\[ T = T_{\text{FXP}1} \cdots T_{\text{FXP}n} \]

\[ T_{\text{FXP}i} = \{ t_1, t_2, \ldots, t_k \} \]

\[ t_j = [DBL \rightarrow (1.15)] \]
$IntMax$

integer bits required for no overflow
FracMin
fractional bits required to meet local SNR
\[ t = \left[ DBL \rightarrow (I.F) \right] \]

\[ F X P_{(I.F)}(\vec{O}_i) \]
\[ \sqrt{\sum_{j=1}^{n} (o_j - F X P_{I.F}(o_j))^2} \]

\[ \| \vec{O}_i - F X P_{I.F}(\vec{O}_i) \| \]
\[ \| \vec{O}_i \| = \sqrt{\sum_{j=1}^{n} o^2_j} \]

\[ SNR_i = \log_{10} \left( \frac{\| \vec{O}_i \|}{\| \vec{O}_i - FXP_{I,F}(\vec{O}_i) \|} \right) \]
FINDING FRACTIONAL BITS

XXXX.XYYYYYYYYYYYYYYYYYY
FINDING FRACTIONAL BITS

XXX.X.YYYYYYYYYYYYYY

SNR

# of bits
FINDING FRACTIONAL BITS

SNR

# of bits

XXX.X.YYYYYYYYYYY
FINDING FRACTIONAL BITS

XXXX.YYYYYYYYYYY
FINDING FRACTIONAL BITS

SNR

# of bits

XXXX.XYYYYYYYY

# of bits
FINDING FRACTIONAL BITS

9 fractional bits

XXX.X.YYYYYYYYY

SNR

# of bits
\[ T = T_{FXP1} \ldots T_{FXPn} \]

\[ T_{FXP1} = \{ t_1, t_2, \ldots, t_k \} \]

\[ t_j = [DBL \rightarrow (IntMax_j \cdot FracMin_j)] \]
THE WORKFLOW
<table>
<thead>
<tr>
<th>File Name</th>
<th>Object</th>
<th>Type</th>
<th>(Initial Suggestion)</th>
<th>SNR</th>
<th>Overflow</th>
<th>Underflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>X</td>
<td>DBL</td>
<td>(I4&lt;2.2&gt;)</td>
<td>(12.41)</td>
<td>(0%)</td>
<td>(23%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Reverse Coefficients</td>
<td>DBL</td>
<td>(I3&lt;2.1&gt;)</td>
<td>(11.16)</td>
<td>(0%)</td>
<td>(66.7%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Forward Coefficients</td>
<td>DBL</td>
<td>(U2&lt;-1.3&gt;)</td>
<td>(10.57)</td>
<td>(0%)</td>
<td>(0%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Numeric Constant</td>
<td>DBL</td>
<td>(U1&lt;0.1&gt;)</td>
<td>(+Inf)</td>
<td>(0%)</td>
<td>(0%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Numeric Constant</td>
<td>DBL</td>
<td>(U1&lt;0.1&gt;)</td>
<td>(+Inf)</td>
<td>(0%)</td>
<td>(0%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>DBL</td>
<td>(I3&lt;-1.4&gt;)</td>
<td>(11.55)</td>
<td>(0%)</td>
<td>(25.4%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>DBL</td>
<td>(I3&lt;-1.4&gt;)</td>
<td>(11.55)</td>
<td>(0%)</td>
<td>(25.4%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>DBL</td>
<td>(I3&lt;-1.4&gt;)</td>
<td>(11.55)</td>
<td>(0%)</td>
<td>(25.4%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>DBL</td>
<td>(I3&lt;-1.4&gt;)</td>
<td>(11.55)</td>
<td>(0%)</td>
<td>(25.4%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Add</td>
<td>DBL</td>
<td>(I4&lt;1.3&gt;)</td>
<td>(12.46)</td>
<td>(0%)</td>
<td>(24.5%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Add</td>
<td>DBL</td>
<td>(I4&lt;1.3&gt;)</td>
<td>(13.2)</td>
<td>(0%)</td>
<td>(24.9%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Subtract</td>
<td>DBL</td>
<td>(I5&lt;2.3&gt;)</td>
<td>(14.8)</td>
<td>(0%)</td>
<td>(30.5%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>DBL</td>
<td>(I4&lt;0.4&gt;)</td>
<td>(10.71)</td>
<td>(0%)</td>
<td>(43.2%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>DBL</td>
<td>(I4&lt;-1.5&gt;)</td>
<td>(13.95)</td>
<td>(0%)</td>
<td>(32.8%)</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Add</td>
<td>DBL</td>
<td>(I5&lt;0.5&gt;)</td>
<td>(14.22)</td>
<td>(0%)</td>
<td>(32.5%)</td>
</tr>
</tbody>
</table>
LOCAL ERROR V.
REAL ERROR
CORRECT ENOUGH
BY CONSTRUCTION
<table>
<thead>
<tr>
<th>File Name</th>
<th>Object</th>
<th>Type</th>
<th>(Initial Suggestion)</th>
<th>SNR</th>
<th>Overflow</th>
<th>Underflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>X</td>
<td>I4&lt;2.2&gt;</td>
<td></td>
<td>18.1</td>
<td>0%</td>
<td>11.4%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Reverse Coefficients</td>
<td>I3&lt;2.1&gt;</td>
<td></td>
<td>11.16</td>
<td>0%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Forward Coefficients</td>
<td>U2&lt; -1.3&gt;</td>
<td></td>
<td>16.36</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Numeric Constant</td>
<td>U1&lt;0.1&gt;</td>
<td>+Inf</td>
<td></td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Numeric Constant</td>
<td>U1&lt;0.1&gt;</td>
<td>+Inf</td>
<td></td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>I3&lt;-1.4&gt;</td>
<td></td>
<td>15.55</td>
<td>5.92%</td>
<td>11.4%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>I3&lt;0.3&gt;</td>
<td></td>
<td>12.96</td>
<td>0%</td>
<td>11.4%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>I3&lt;-1.4&gt;</td>
<td></td>
<td>15.55</td>
<td>5.92%</td>
<td>11.4%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Add</td>
<td>I4&lt;-1.3&gt;</td>
<td></td>
<td>11.3</td>
<td>0%</td>
<td>21.9%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Add</td>
<td>I4&lt;-1.3&gt;</td>
<td></td>
<td>10.51</td>
<td>0%</td>
<td>20%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Subtract</td>
<td>I5&lt;2.3&gt;</td>
<td></td>
<td>3.56</td>
<td>0%</td>
<td>17.6%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>I4&lt;-0.4&gt;</td>
<td></td>
<td>-1.92</td>
<td>3.8%</td>
<td>17.6%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Multiply</td>
<td>I4&lt;-1.5&gt;</td>
<td></td>
<td>0</td>
<td>0%</td>
<td>99.9%</td>
</tr>
<tr>
<td>Second Order IIR Filter - FXP.gvi</td>
<td>Add</td>
<td>I5&lt;0.5&gt;</td>
<td></td>
<td>-7.81</td>
<td>0%</td>
<td>17.6%</td>
</tr>
</tbody>
</table>
Reverse coefficients path

\[ l4 < 0.4 > : -1.92 \]

\[ l4 < -1.5 > : 0 \]

\[ l5 < 0.5 > : -7.81 \]

\[ l5 < 2.3 > : 3.56 \]
\[ T = T_{FXP1}, T_{FXP2}, \ldots, T_{FXPn} \]
EXECUTABILITY ALLOWS FOR CONTINUAL FEEDBACK
CONCLUSION

Our F2F tool in LabVIEW NXG FPGA:

1. Lets algorithm designers model their new algorithms in LabVIEW
2. Uses the executable nature of LabVIEW to generate an initial model transform
3. Provides tools to help designers create and apply the remaining transforms
4. Gives constant feedback on “correct enough” by construction