

This is a summary of a paper presented at the 70th European Association of Geoscientists and Engineers (EAGE) Conference, Rome, Italy, June 2008.

About the Authors

Haohuan Fu, William Osborne are at Imperial College London.

Bob Clapp is with the Center for Computational Earth & Environmental Science (CEES) at Stanford University.

Oliver Pell is at Maxeler Technologies.

Summary

Reducing the time for converting collected data into seismic images is becoming critical for the production cycle of the oil industry. Conventional seismic processing is performed on a CPU with 32 or 64-bit precision for all operations. In certain cases, using a reduced precision produces equivalent result within acceptable tolerances. CPUs do not support configurable bit-widths thus reducing precision brings no performance benefit. In contrast, FPGAs enable application-specific number representations and reducing the bit-width used can greatly increase performance with more concurrent processing cores per device.

In this paper, we use a complex exponential in downward continued based migration application as our case study. We have developed a tool to help us evaluate trade-offs between precision and performance, which we use to analyze the most appropriate number formats for the complex exponential step. The design consists of three parts, square root calculation (SQRT), sine/cosine evaluation (SINE), and wavefield complex multiplication (WMUL).

During our exploration of fixed-point formats, as the range of variables in the three parts are quite different from each other, we apply a different bit-

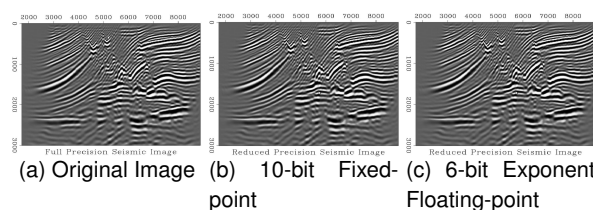


Figure 1: Seismic images for different fixed-point and floating-point precision.

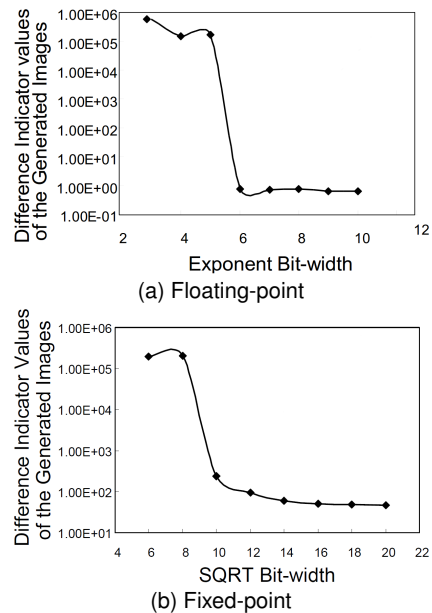


Figure 2: Difference indicator value for differences in resultant image given different data precision.

width in each of the three parts. We keep two of the widths constant, and change the other width gradually to find out its effect on the accuracy of the application. Figure 2b shows the results of an example where we keep SINE and WMUL bit-widths as 20 and 30, and vary the SQRT bit-width between 6 and 20. In our example, when SQRT bit-width increases to 10 bits, the difference indicator (DI) value changed from 10^5 to 10^2 .

Applying a similar method to other parts, we determine the values of 12 and 16 for SINE and WMUL bit-widths. We automatically enumerate nearby cases and determine the minimum widths to provide acceptable image quality are 12, 16, 16 for SQRT, SINE and WMUL.

For floating point designs, we explored combinations of different exponents and significand widths. In Figure 2a, we used a significand width of 24 and gradually increase the exponent width from 3 to 10 bits. There is a clear change at a width of 6 bits, where the DI value drops from 10^5 to 10. Using these values as an initial estimate, we again apply enumerations for values nearby, and determine that the minimum exponent and significand bit-widths are 6 and 16 for a floating-point design. Figure 1 shows the resultant images for the reduced-precision implementations and the original single-precision floating point image.