Configurable Floating-Point Unit for the SHMAC platform

Master thesis

Audun L. Indergaard

NTNU

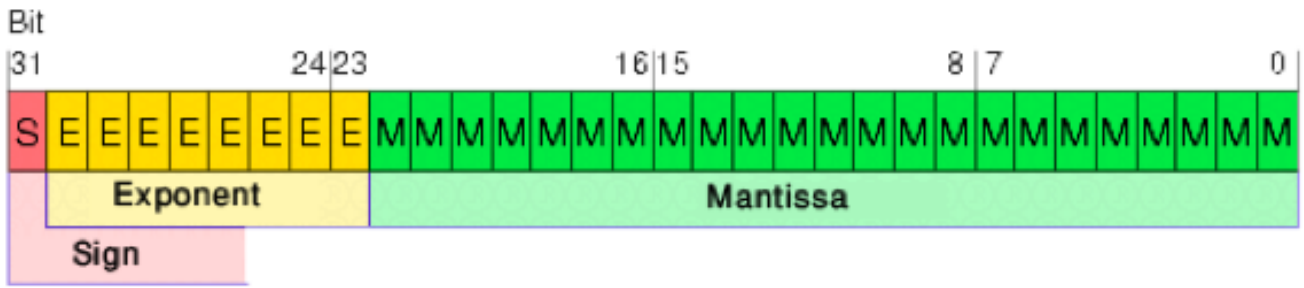
SHMAC Seminar, Monday May 5th 2014

Goals of thesis

- Study the IEEE floating-point standard and its implementation
- Study application specific FPU implementations and in particular any configurable FPU implementations found in the literature.
- Implement a simple FPU for use on the SHMAC platform and test this for selected software applications.
- Study the requirements of the selected software applications and look for FPU optimization possibilities.
- Implement one or more application specific and/or configurable FPUs.
- Evaluate performance and energy gains achieved as well as area results. If time allows, also compare with fixed-point implementations of the software applications.

Floating-Point

- Represent a desimal number
- IEEE Standard 754 for Floating-Point Arithmetic
- Multimedia applications, graphics processing



Figur: Single precision floating point number.

Pros and cons

Floating-Point Unit

+ Flexible
+ Relives the CPU
- Uses more area

Fixed-Point

+ Quick
+ Does not require additional hardware
- Un-flexible

Configurable Floating-Point Unit

How to optimize the FPU?

- Optimize range and precision
- Emulate some arithmetic in software
- Collaps FP operations

Number of FP operations in four SPEC2000fp benchmarks [Def12]:

	+	-	*	/
188.ammp	479	330	930	42
179.art	253	14	247	12
183.equake	127	58	236	18
177.mesa	347	102	586	27

Bit-width Optimisation [GMLC04]

Floating-Point

- U_i represents a floating-point number $(-1)^S \times M \times 2^E$
- m is the bit-width of mantissa
- e is the bit-width of exponent
- ΔU_i is the allowed error
- E_{U_i} is the value of the exponent

$$m \geq E_{U_i} - \lceil \log_2(|\Delta U_i|) \rceil + 1 \quad (1)$$

$$e \geq \lceil \log_2(|max(E_{U_i})/min(E_{U_i})|) \rceil \quad (2)$$

Example for Ammp: $Max(U_i) = 200,000, \Delta U_i = 0.000005$
$Min(U_i) = 0.00001$
$e \geq 5$
$m \geq 28$
$Total\ bit > 34$

Bit-width Optimisation [GMLC04]

Fixed-Point

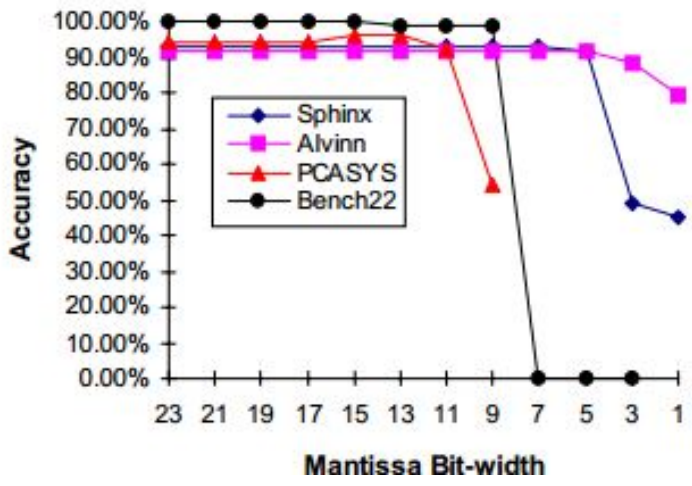
- U_i represents a fixed point number
- k is the bit-width of integer part
- l is the bit-width of fraction part
- ΔU_i is the allowed error

$$k \geq \lceil \log_2(|max(U_i)/min(U_i)|) \rceil \quad (3)$$

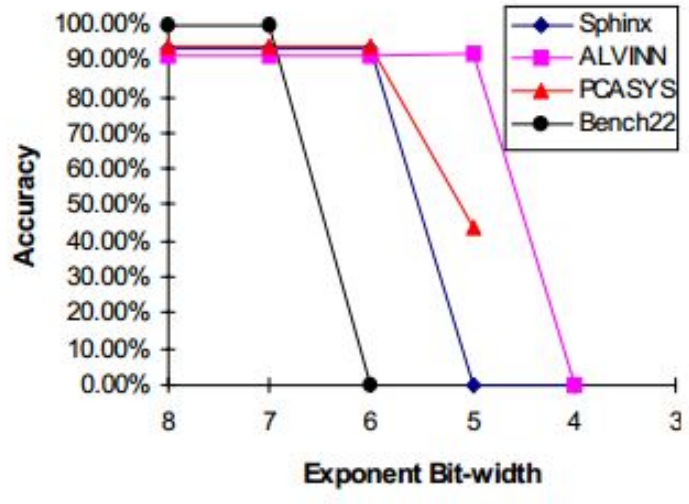
$$l \geq \lceil \log_2(|\Delta U_i|) \rceil + 1 \quad (4)$$

Example for Ammp: $Max(U_i) = 200,000, \Delta U_i = 0.000005$
$Min(U_i) = 0.00001$
$k \geq 18$
$l \geq 18$
$Total\ bit = 36$

Bit-width Optimisation [TRN98]

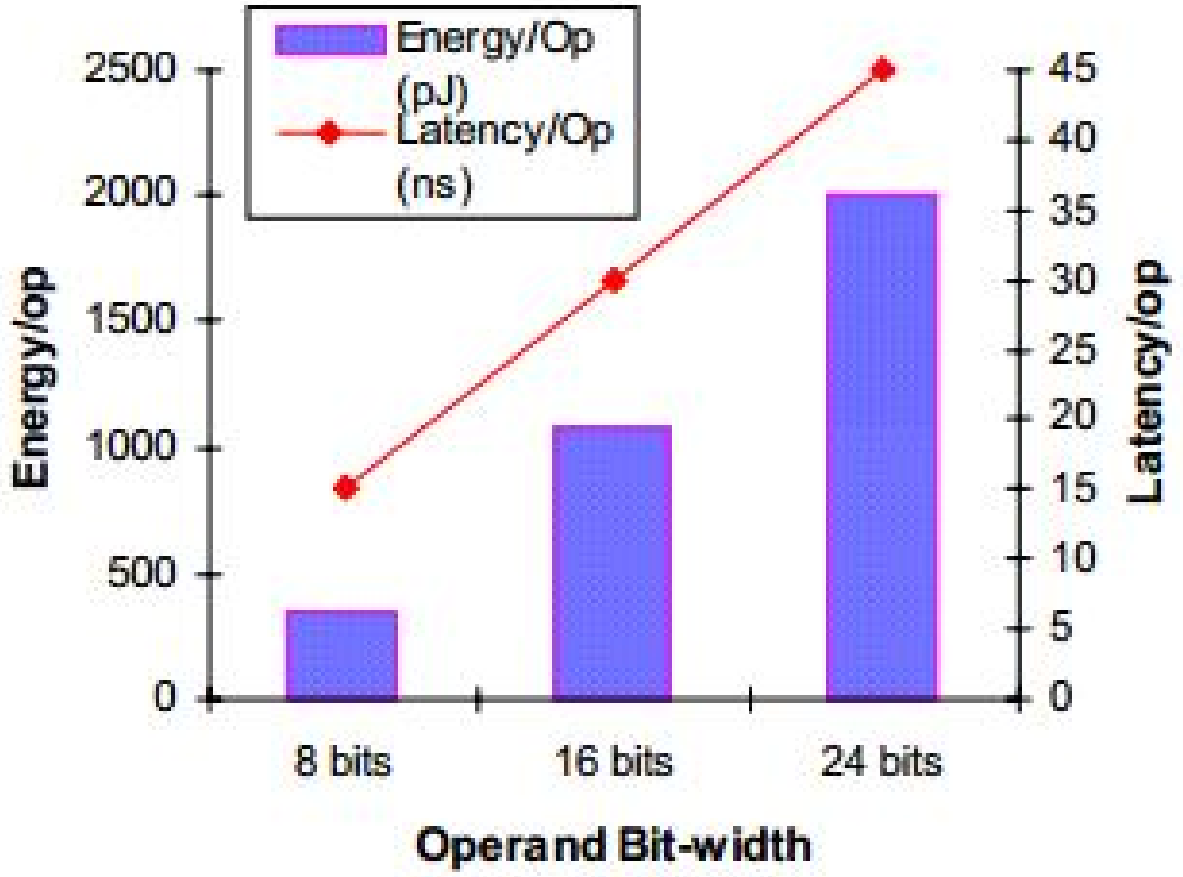


Figur: Program Accuracy across Various Mantissa Bit-widths



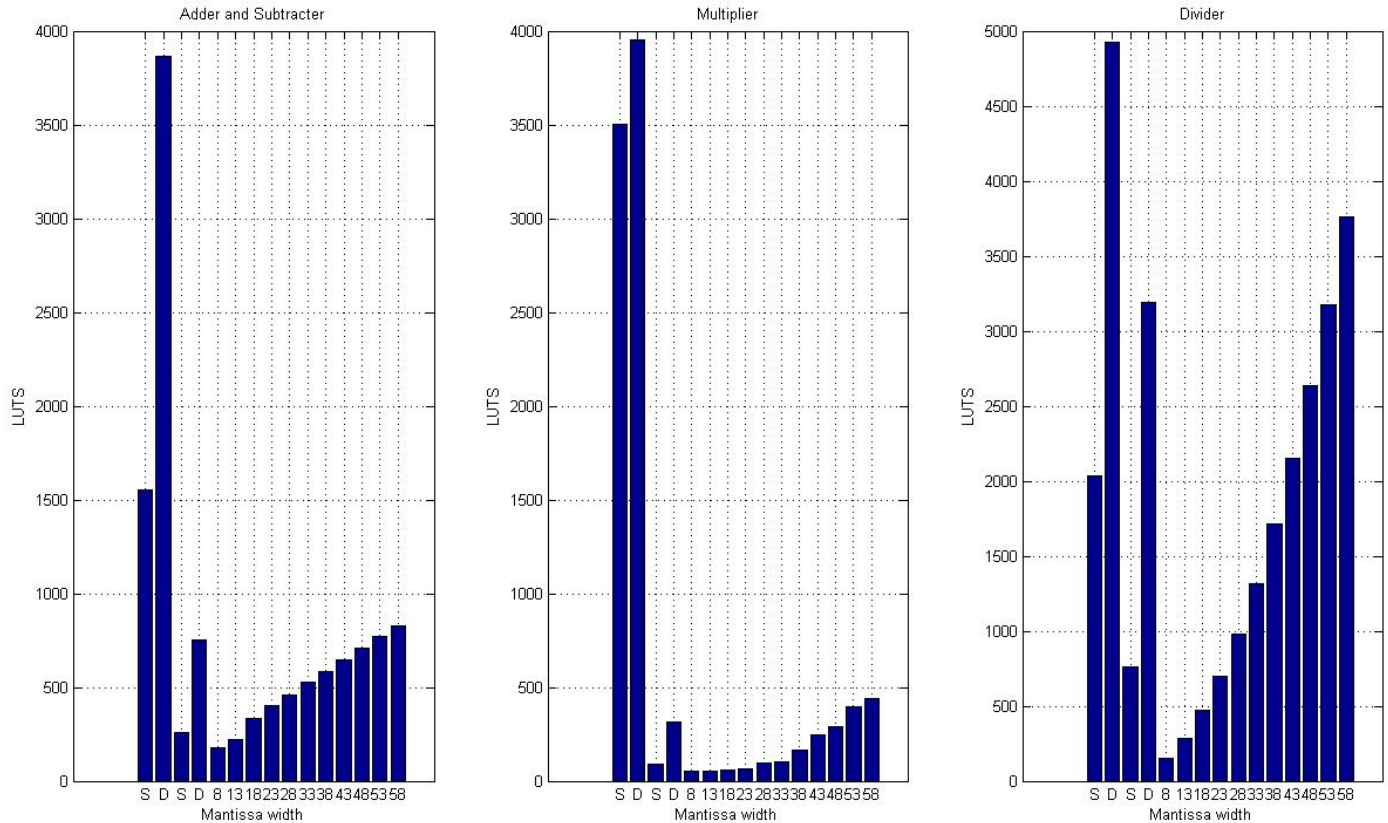
Figur: Program Accuracy across Various Exponent Bit-widths

Bit-width Optimisation [TRN98]



Figur: Energy and performance with different bit-widths

Size of FP Arithmetic [Ind14]



Figur: Area of adder and subtracter with mantissa varying from 8 to 58 bit

Q&A

Bibliography


David Defour, *Collapsing floating-point operations*, Tech. report, Universite de Perpignan, 2012.


Altaf Abdul Gaffar, Oskar Mencer, Wayne Luk, and Peter Y.K. Cheung, *Unifying Bit-width Optimisation for Fixed-point and Floating-point Designs*, Tech. report, IEEE, 2004.


Audun Lie Indergaard, *Configurable Floating-Point Unit for the SHMAC platform*, Tech. report, Norwegian University of Science and Technology, 2014.


Ying Fai Tong, Rob A. Rutenbar, , and David F. Nagle, *Minimizing Floating-Point Power Dissipation Via Bit-Width Reduction*, Tech. report, Carnegie Mellon University, 1998.