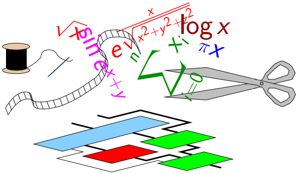
Hardware LTI filters computing just right

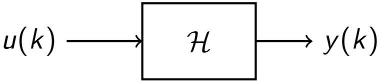
F. de Dinechin,
Th. Hilaire,
M. Istoan,
A. Volkova



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON
Inria informatiques mathématiques
Université de Lyon

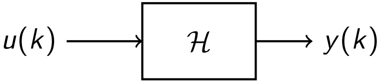
Once upon a time in the maths

… there lived a handsome filter named $\mathcal{H}$



Once upon a time in the maths

... there lived a handsome filter named $\mathcal{H}$



The Matlab fairies had blessed him with the most perfect transfer function:

$$\mathcal{H}(z) = \frac{\sum_{i=0}^{n_b} b_i z^{-i}}{1 + \sum_{i=1}^{n_a} a_i z^{-i}}, \quad \forall z \in \mathbb{C}.$$

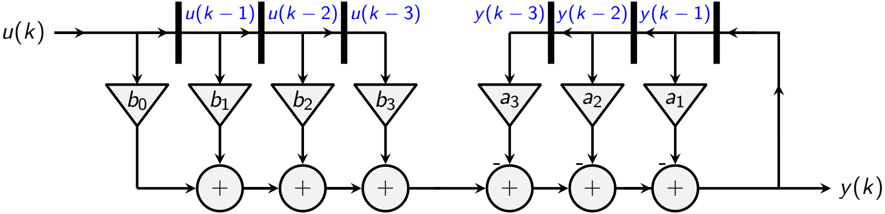
whose coefficients (a_i) and (b_i) were real numbers
so pure they could be computed very accurately in Matlab.

And so $\mathcal{H}$ converged beautifully

using its evaluation formula

$$y(k) = \sum_{i=0}^{n_b} b_i u(k-i) - \sum_{i=1}^{n_a} a_i y(k-i)$$

as long as the poles of $\mathcal{H}$ remained safely within the unit circle.

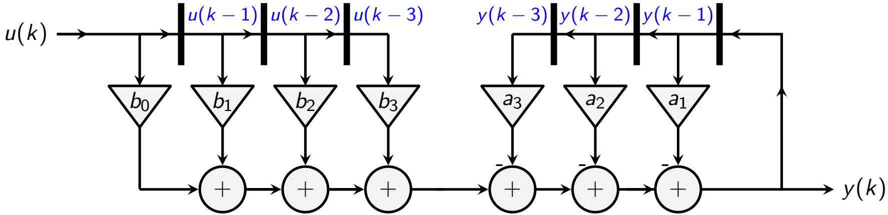


And so $\mathcal{H}$ converged beautifully

using its evaluation formula

$$y(k) = \sum_{i=0}^{n_b} b_i u(k-i) - \sum_{i=1}^{n_a} a_i y(k-i)$$

as long as the poles of $\mathcal{H}$ remained safely within the unit circle.



But the fairies had warned $\mathcal{H}$:

Don't let your poles come close to the unit circle!

For it was a cruel, fixed-point world out there

Although $\mathcal{H}$ had very pure coefficients,

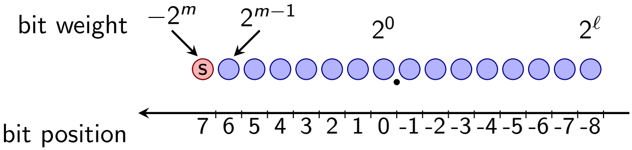
he had to serve in the dark underworld of CorteXLab.

For it was a cruel, fixed-point world out there

Although $\mathcal{H}$ had very pure coefficients,

he had to serve in the dark underworld of CorteXLab.

There, inputs and outputs were no real numbers, but low-resolution numbers in some ugly binary fixed-point (m, l) format.

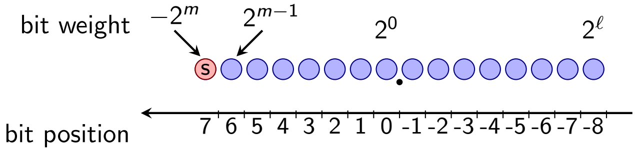


For it was a cruel, fixed-point world out there

Although $\mathcal{H}$ had very pure coefficients,

he had to serve in the dark underworld of CorteXLab.

There, inputs and outputs were no real numbers, but low-resolution numbers in some ugly binary fixed-point (m, l) format.



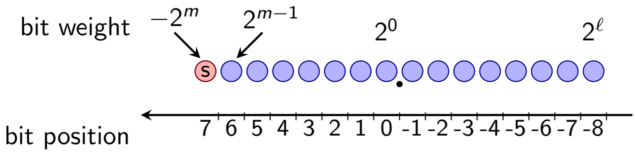
For the input, it was OK, for a fixed-point number is also a real number.

For it was a cruel, fixed-point world out there

Although $\mathcal{H}$ had very pure coefficients,

he had to serve in the dark underworld of CorteXLab.

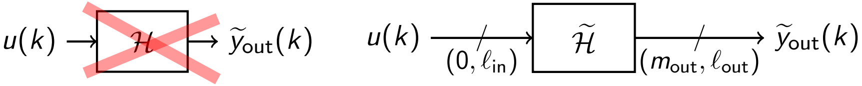
There, inputs and outputs were no real numbers, but low-resolution numbers in some ugly binary fixed-point (m, l) format.



For the input, it was OK, for a fixed-point number is also a real number.

But $\mathcal{H}$ had to round his outputs,

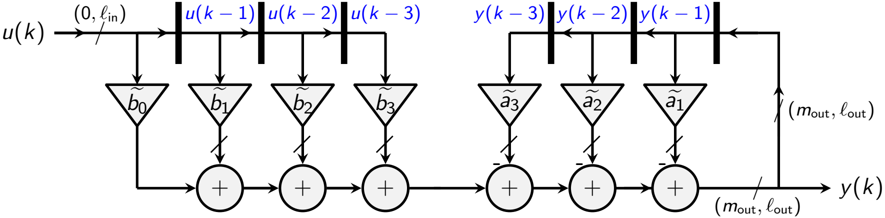
and this transformed him into a vile monster with a tilde.



$\widetilde{\mathcal{H}}$ was a monster, and he diverged

When he rounded his pure coefficients to some fixed-point format,

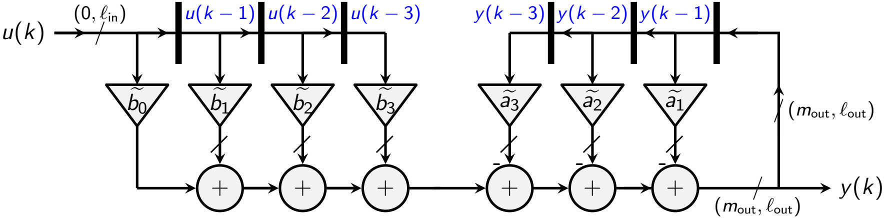
$\widetilde{\mathcal{H}}$ forgot the fairies' advice!



$\widetilde{\mathcal{H}}$ was a monster, and he diverged

When he rounded his pure coefficients to some fixed-point format,

$\widetilde{\mathcal{H}}$ forgot the fairies' advice!



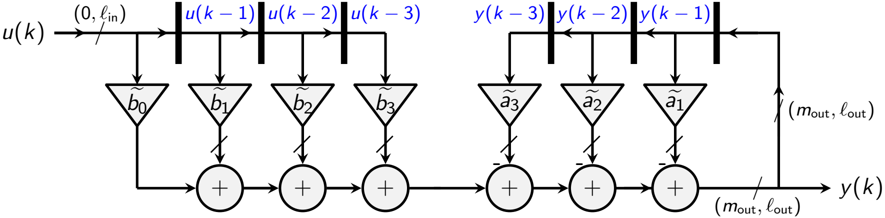
Rounding errors were amplified on the feedback loop,

and $\widetilde{\mathcal{H}}$ diverged miserably.

$\widetilde{\mathcal{H}}$ was a monster, and he diverged

When he rounded his pure coefficients to some fixed-point format,

$\widetilde{\mathcal{H}}$ forgot the fairies' advice!



Rounding errors were amplified on the feedback loop,

and $\widetilde{\mathcal{H}}$ diverged miserably.

And so Nokia engineers abandonned small and handsome $\mathcal{H}$,

preferring a plain 128-tap FIR filter which was bulky, but stable.

So $\mathcal{H}$ was crying, alone and forgotten

So $\mathcal{H}$ was crying, alone and forgotten

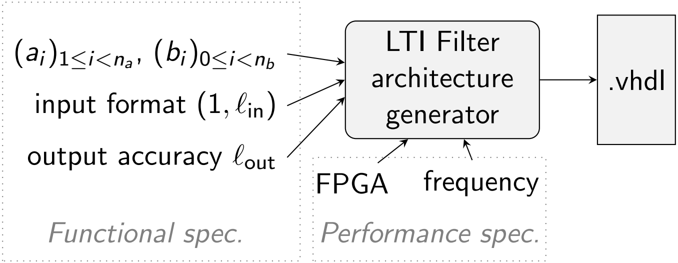
... when the good old witch FloPoCo heard his complaint.

Looking at him, she said:

you're not that evil, you are just poorly specified.

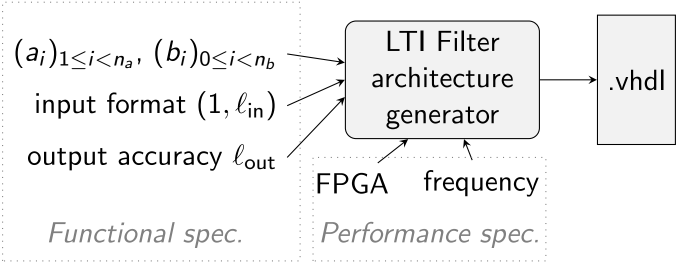
So $\mathcal{H}$ was crying, alone and forgotten

... when the good old witch FloPoCo heard his complaint.
Looking at him, she said:
you're not that evil, you are just poorly specified.
And in a whip of her magical TikZ cursor, she designed him
a new interface:



So $\mathcal{H}$ was crying, alone and forgotten

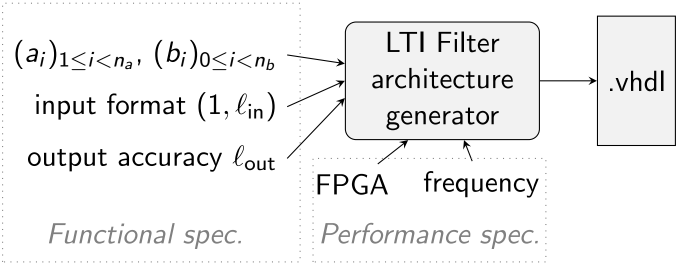
... when the good old witch FloPoCo heard his complaint.
Looking at him, she said:
you're not that evil, you are just poorly specified.
And in a whip of her magical TikZ cursor, she designed him
a new interface:



These a_i and b_i were reals! The very real coefficients!
$\mathcal{H}$ suddenly felt much lighter than he had ever felt.

So $\mathcal{H}$ was crying, alone and forgotten

... when the good old witch FloPoCo heard his complaint.
Looking at him, she said:
you're not that evil, you are just poorly specified.
And in a whip of her magical TikZ cursor, she designed him
a new interface:



These a_i and b_i were reals! The very real coefficients!

$\mathcal{H}$ suddenly felt much lighter than he had ever felt.

But... but...

But you forgot to provide me a m_{out}, cried $\mathcal{H}$

But you forgot to provide me a m_{out}, cried $\mathcal{H}$

No, said FloPoCo, for I have, somewhere in my library, a spell that can compute it out of your coefficients.

But you forgot to provide me a m_{out}, cried $\mathcal{H}$

No, said FloPoCo, for I have, somewhere in my library, a spell that can compute it out of your coefficients. (wait a moment, where is it? It was written by poor princess Anastasia Volkova during her captivity in the caves of the mighty sorcerers Lauter and Hilaire...)

But you forgot to provide me a m_{out}, cried $\mathcal{H}$

No, said FloPoCo, for I have, somewhere in my library, a spell that can compute it out of your coefficients. (wait a moment, where is it? It was written by poor princess Anastasia Volkova during her captivity in the caves of the mighty sorcerers Lauter and Hilaire...) (I hope it still compiles...)

But you forgot to provide me a m_{out}, cried $\mathcal{H}$

No, said FloPoCo, for I have, somewhere in my library, a spell that can compute it out of your coefficients. (wait a moment, where is it? It was written by poor princess Anastasia Volkova during her captivity in the caves of the mighty sorcerers Lauter and Hilaire...) (I hope it still compiles...) Ha, here you go:

Definition: *Worst-Case Peak Gain* $\langle\langle\mathcal{H}\rangle\rangle$ of a filter $\mathcal{H}$

$$\langle\langle\mathcal{H}\rangle\rangle = \max_{||u||_\infty = 1} ||y||_\infty$$

where $||u||_\infty$ is defined as $||u||_\infty = \max_k |u(k)|$.

Then of course,

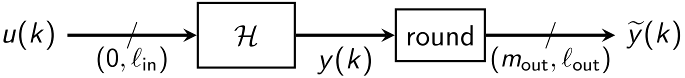
$$m_{\text{out}} = \lceil \log_2 \langle\langle\mathcal{H}\rangle\rangle \rceil \ .$$

But how will this save me from diverging? cried $\mathcal{H}$

- Remember: you are $\mathcal{H}$, answered the good witch
  and $\mathcal{H}$ doesn't diverge in the pure mathematical world

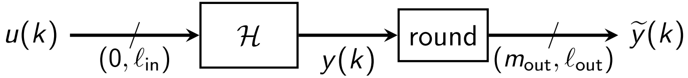
But how will this save me from diverging? cried $\mathcal{H}$

- Remember: you are $\mathcal{H}$, answered the good witch
  and $\mathcal{H}$ doesn't diverge in the pure mathematical world
- Let me cast this spell on your architecture:
  $\widetilde{\mathcal{H}}$ shall return a result that is that of $\mathcal{H}$, rounded only once.
- Then, your alter ego $\widetilde{\mathcal{H}}$ won't diverge.



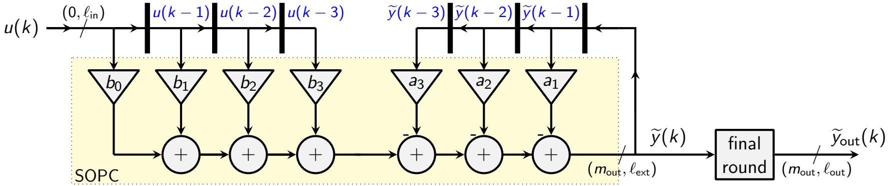
But how will this save me from diverging? cried $\mathcal{H}$

- Remember: you are $\mathcal{H}$, answered the good witch
  and $\mathcal{H}$ doesn't diverge in the pure mathematical world
- Let me cast this spell on your architecture:
  $\widetilde{\mathcal{H}}$ shall return a result that is that of $\mathcal{H}$, rounded only once.
- Then, your alter ego $\widetilde{\mathcal{H}}$ won't diverge.

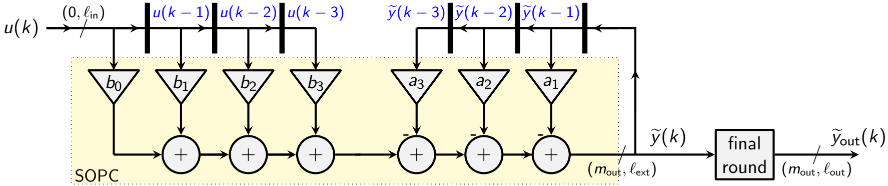


And FloPoCo invoked his two most crafted gremlins, Istoan and de Dinechin, to code this spell, with the help of Princess Anastasia who had managed to escape her tormentors.

Actual architecture



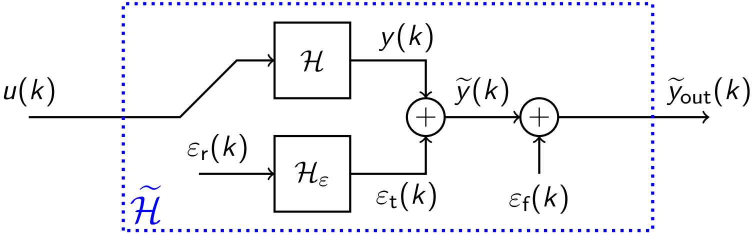
Actual architecture



Another point of view:
When $\ell_{ext} \to -\infty$ (which means: as the internal accuracy increase), at some point the computation should become accurate enough for $\widetilde{\mathcal{H}}$ to converge.

Amplification of errors on the feedback loop

Here should come 3 pages of runes which end in the following figure:



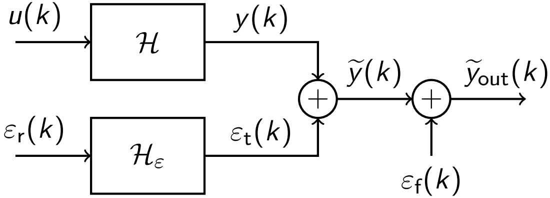
- $\overline{\varepsilon}_{\text{r}}$ is the sum of all rounding errors

$$\varepsilon_{\text{r}}(k) = \widetilde{y}(k) - \left(\sum_{i=0}^{n_b} b_i u(k-i) - \sum_{i=1}^{n_a} a_i \widetilde{y}(k-i) \right)$$

- $\mathcal{H}_\varepsilon$ is the virtual filter that captures the error amplification on the feedback loop:

$$\overline{\varepsilon}_{\text{t}} = \langle\!\langle \mathcal{H}_\varepsilon \rangle\!\rangle \, \overline{\varepsilon}_{\text{r}} \ .$$

Errors are captured, let us chain them in the basement



Rounding errors depend on the architecture

Example: An architecture optimized for LUT-based FPGAs:

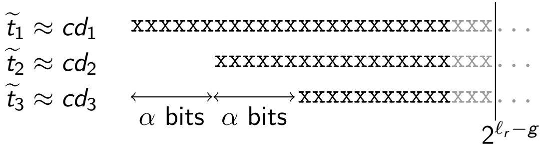
- Split an input x into D chunks of α bits (e.g. $\alpha = 4$: hexadecimal).

$$x \;=\; \sum_{k=1}^{D} 2^{-k\alpha} d_k \quad \text{where} \quad d_k \in \{0, ..., 2^{\alpha} - 1\}$$

- Then cx becomes

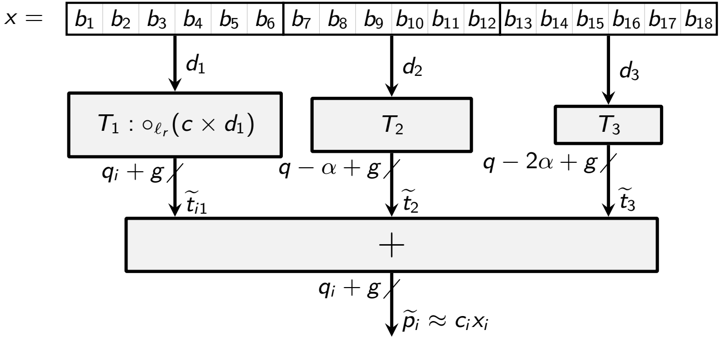
$$cx \;=\; \sum_{k=1}^{D} 2^{-k\alpha} c d_k$$

- Tabulate each cd_k sub-product in an α-input table indexed by d_k



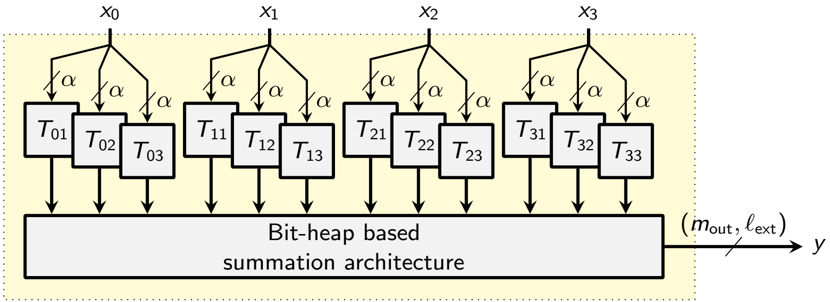
- Remark: c is a real number here, no need to quantize it!

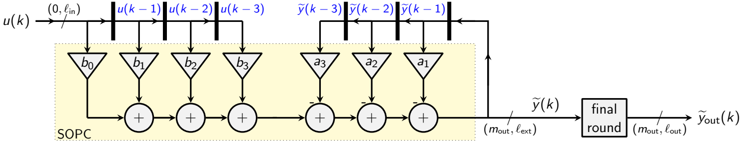
A LUT-based architecture



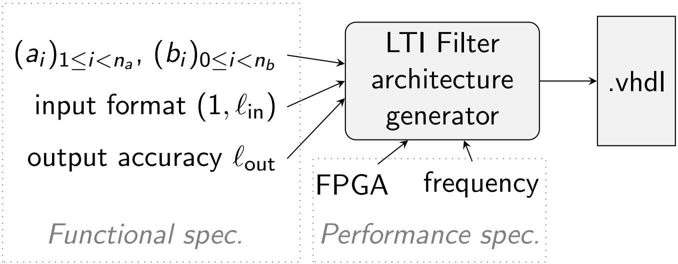
The error is proportional to 2^{-g}, so can made as small as needed by increasing g.

Overall architecture





You're not evil, you're just poorly specified



- a_i and b_i: real numbers
  - high-precision numbers from Matlab
  - mathematical formulae such as `sin(3*pi/8)`
- ℓ_{in} and ℓ_{out}: integers denoting the weight of the least significant bits of the input and of the result.

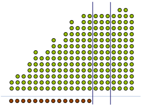
Computing just right (TM):

ℓ_{out} specifies output precision, but also output accuracy.

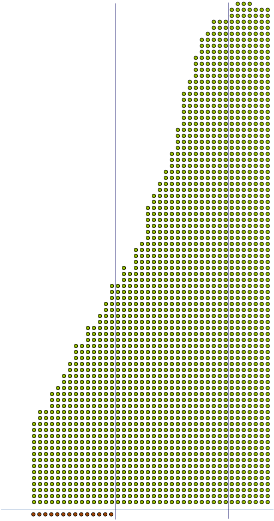
A demo?

- some Butterworth filters
- a radar filter submitted to Thibault a few years ago

Bit heaps for 12-bit Butterworth filters



Order 4, $g = 4$
because $-\log_2 \langle\!\langle \mathcal{H}_\varepsilon \rangle\!\rangle = 3$



Order20, $g = 7$
because $-\log_2 \langle\!\langle \mathcal{H}_\varepsilon \rangle\!\rangle = 19$

So everybody lived happily ever after...

- A point of view on filter design that is universal
  - don't compute useless bits: output format specifies output accuracy
  - complete error analysis (coefficient quantization + architectural rounding errors)
  - error amplification captured by a safe implementation of the WCPG

So everybody lived happily ever after...

- A point of view on filter design that is universal
  - don't compute useless bits: output format specifies output accuracy
  - complete error analysis (coefficient quantization + architectural rounding errors)
  - error amplification captured by a safe implementation of the WCPG
- A finely tuned implementation that uses FPGA-specific arithmetic

... And they had a lot of children

This is just a basic block on the way to more interesting filter structures.

- implementation space: state space, SIF
- clean rule of the game: enables comparison of functionally equivalent architectures

(to be continued)

The End

- Try me in FloPoCo v. 4.1.3
- Read more on HAL:
  Towards Hardware IIR Filters Computing Just Right:
  Direct Form I Case Study