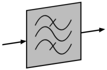
Determining Fixed-Point Formats using the Worst-Case Peak Gain measure

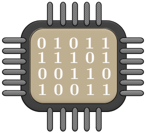
Anastasia Volkova, Thibault Hilaire, Christoph Lauter

Sorbonne Universités, University Pierre and Marie Curie, LIP6, Paris, France

ASILOMAR 49
November 10, 2015

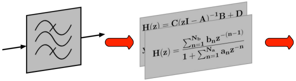
Motivation

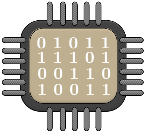




Filter implementation flow:

Motivation

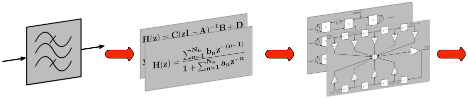


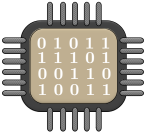


Filter implementation flow:

- Transfer function generation

Motivation

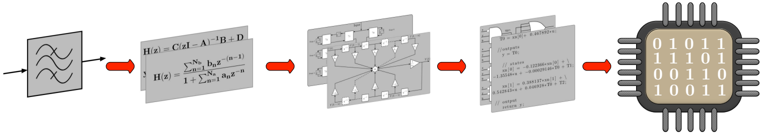




Filter implementation flow:

- Transfer function generation
- State-space, DFI, DFII, ...

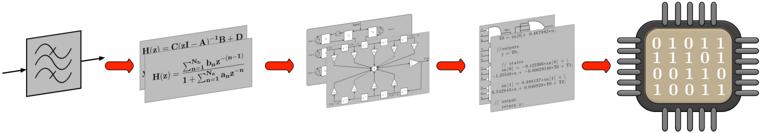
Motivation



Filter implementation flow:

- Transfer function generation
- State-space, DFI, DFII, ...
- Software or Hardware implementation

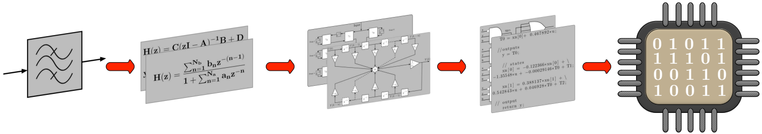
Motivation



Filter implementation flow:

- Transfer function generation
  - ! Coefficient quantization
- State-space, DFI, DFII, ...
- Software or Hardware implementation

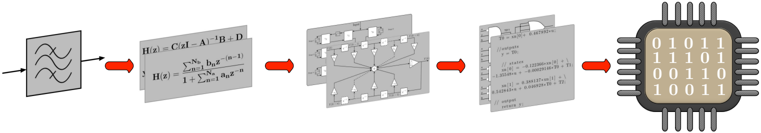
Motivation



Filter implementation flow:

- Transfer function generation
  - ! Coefficient quantization
- State-space, DFI, DFII, ...
  - ! Large variety of structures with no common quality criteria
- Software or Hardware implementation

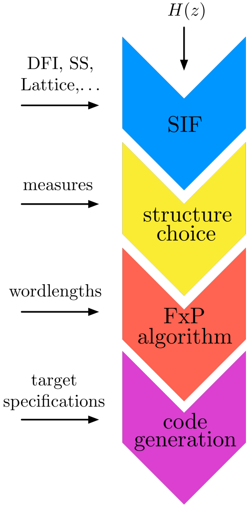
Motivation



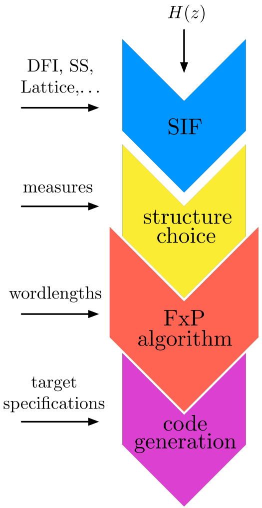
Filter implementation flow:

- Transfer function generation
  - ! Coefficient quantization
- State-space, DFI, DFII, ...
  - ! Large variety of structures with no common quality criteria
- Software or Hardware implementation
  - ! Constraints: power consumption, area, error, speed, etc.

Motivation: Automatized filter implementation flow

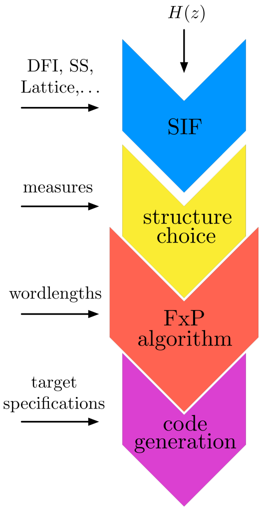


Motivation: Automatized filter implementation flow



Focus on Fixed-Point realization.

Motivation: Automatized filter implementation flow



Focus on Fixed-Point realization.
Optimization of wordlengths:

- Take more - pay more
- Take less - overflow risk

What we want in the end:

- Rigorous algorithm for Fixed-Point Formats (FxPF) determination
- Integration into automatized code generator for filters
- Multiple wordlength paradigm

LTI filters

$\mathcal{H} := (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ is a Bounded Input Bounded Output LTI filter in state-space representation:

$$\mathcal{H} \left\{ \begin{array}{rcl} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) & = & \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \end{array} \right.$$

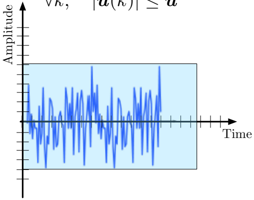
with q inputs, n states and p outputs and state matrices

Basic brick: interval through filter

The Worst-Case Peak Gain theorem

Input $\boldsymbol{u}(k)$

$\forall k, \quad |\boldsymbol{u}(k)| \leq \bar{\boldsymbol{u}}$

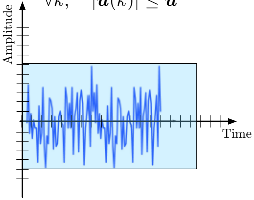


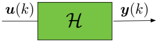
Basic brick: interval through filter

The Worst-Case Peak Gain theorem

Input $\boldsymbol{u}(k)$

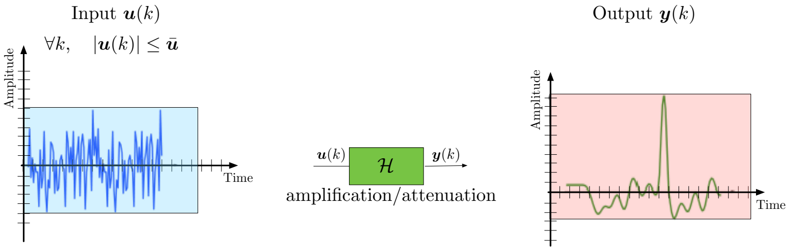
$\forall k, \quad |\boldsymbol{u}(k)| \leq \bar{\boldsymbol{u}}$





amplification/attenuation

The Worst-Case Peak Gain theorem

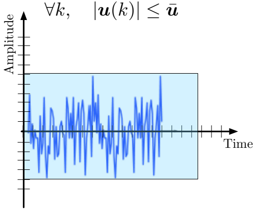


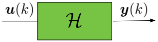
Basic brick: interval through filter

The Worst-Case Peak Gain theorem

Input $\boldsymbol{u}(k)$

$\forall k, \quad |\boldsymbol{u}(k)| \leq \bar{\boldsymbol{u}}$

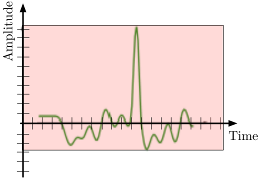




amplification/attenuation

Output $\boldsymbol{y}(k)$

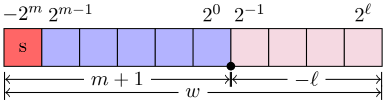
$\forall k, \quad |\boldsymbol{y}(k)| \leq \langle\langle \mathcal{H} \rangle\rangle \, \bar{\boldsymbol{u}}$



Worst-Case Peak Gain

$$\langle\langle \mathcal{H} \rangle\rangle = |\mathbf{D}| + \sum_{k=0}^{\infty} |\mathbf{C}\mathbf{A}^k\mathbf{B}|$$

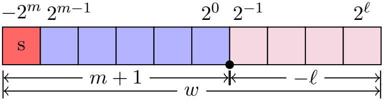
Two's complement Fixed-Point arithmetic



$$t = -2^m t_m + \sum_{i=\ell}^{m-1} 2^i t_i$$

- Wordlength: w
- **M**ost **S**ignificant **B**it position: m
- **L**east **S**ignificant **B**it position: $\ell := m - w + 1$

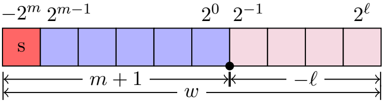
Two's complement Fixed-Point arithmetic



$$t = -2^m t_m + \sum_{i=\ell}^{m-1} 2^i t_i$$

- quantization step: 2^ℓ
- t represented by integer $T = t \cdot 2^\ell$
- $T \in [-2^m; 2^m - 2^\ell] \cap \mathbb{Z}$

Two's complement Fixed-Point arithmetic



$$t = -2^m t_m + \sum_{i=\ell}^{m-1} 2^i t_i$$

- $y(k) \in \mathbb{R}$
- wordlength w bits
- minimal Fixed-Point Format (FPF) is the least m:

$$\forall k, \quad y(k) \in [-2^m; 2^m - 2^{m-w+1}]$$

Problem statement

Let $\mathcal{H} := (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ be a LTI filter:

$$\mathcal{H}\left\{\begin{array}{rcl} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) & = & \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \end{array}\right.$$

Given

- wordlength constraints $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ for each state and output variable
- input domain $\bar{\boldsymbol{u}}$

we need to determine the minimal FPF for all variables of filter $\mathcal{H}$, i.e. find the least $\boldsymbol{m}_x$ and $\boldsymbol{m}_y$ such that

$$\begin{array}{ll} \forall k, & \boldsymbol{x}_i(k) \in [-2^{\boldsymbol{m}_{x_i}}; 2^{\boldsymbol{m}_{x_i}} - 2^{\boldsymbol{m}_{x_i} - \boldsymbol{w}_{x_i} + 1}] \\ \forall k, & \boldsymbol{y}_i(k) \in [-2^{\boldsymbol{m}_{y_i}}; 2^{\boldsymbol{m}_{y_i}} - 2^{\boldsymbol{m}_{y_i} - \boldsymbol{w}_{y_i} + 1}]. \end{array}$$

Modification of $\mathcal{H}$

Let $\boldsymbol{\zeta}(k) := \begin{pmatrix} \boldsymbol{x}(k) \\ \boldsymbol{y}(k) \end{pmatrix}$ be a vertical concatenation of state and output vectors.

Modification of $\mathcal{H}$

Let $\boldsymbol{\zeta}(k) := \begin{pmatrix} \boldsymbol{x}(k) \\ \boldsymbol{y}(k) \end{pmatrix}$ be a vertical concatenation of state and output vectors.

Then the state-space takes the following form:

$$\mathcal{H}_\zeta \begin{cases} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) & + & \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{\zeta}(k) & = & \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{C} \end{pmatrix} \boldsymbol{x}(k) & + & \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{D} \end{pmatrix} \boldsymbol{u}(k) \end{cases}$$

Modification of $\mathcal{H}$

Let $\boldsymbol{\zeta}(k) := \begin{pmatrix} \boldsymbol{x}(k) \\ \boldsymbol{y}(k) \end{pmatrix}$ be a vertical concatenation of state and output vectors.

Then the state-space takes the following form:

$$\mathcal{H}_\zeta \left\{ \begin{array}{rcccc} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) & + & \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{\zeta}(k) & = & \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{C} \end{pmatrix} \boldsymbol{x}(k) & + & \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{D} \end{pmatrix} \boldsymbol{u}(k) \end{array} \right.$$

We seek to determine the least $\boldsymbol{m}_\zeta$ such that

$$\forall k, \quad |\boldsymbol{\zeta}_i(k)| \leqslant 2^{\boldsymbol{m}_{\zeta_i}} - 2^{\boldsymbol{m}_{\zeta_i} - \boldsymbol{w}_i + 1}.$$

Computing the MSB using the WCPG theorem

Applying the WCPG theorem on filter $\mathcal{H}_\zeta$ gives

$$\forall k, \quad |\boldsymbol{\zeta}_i(k)| \leqslant \left(\langle\langle H_\zeta \rangle\rangle \, \bar{\boldsymbol{u}}\right)_i$$

Computing the MSB using the WCPG theorem

Applying the WCPG theorem on filter $\mathcal{H}_\zeta$ gives

$$\forall k, \quad |\boldsymbol{\zeta}_i(k)| \leqslant \left(\langle\langle H_\zeta \rangle\rangle \, \bar{\boldsymbol{u}}\right)_i$$

Therefore, the smallest $\boldsymbol{m}_{\zeta_i}$ satisfying

$$\left(\langle\langle H_\zeta \rangle\rangle \, \bar{\boldsymbol{u}}\right)_i \leqslant 2^{\boldsymbol{m}_{\zeta_i}} - 2^{\boldsymbol{m}_{\zeta_i} - \boldsymbol{w}_i + 1},$$

will satisfy the wordlength constraints.

Computing the MSB using the WCPG theorem

Applying the WCPG theorem on filter $\mathcal{H}_\zeta$ gives

$$\forall k, \quad |\boldsymbol{\zeta}_i(k)| \leqslant \left(\langle\!\langle H_\zeta \rangle\!\rangle \, \bar{\boldsymbol{u}}\right)_i$$

Therefore, the smallest $\boldsymbol{m}_{\zeta_i}$ satisfying

$$\left(\langle\!\langle H_\zeta \rangle\!\rangle \, \bar{\boldsymbol{u}}\right)_i \leqslant 2^{\boldsymbol{m}_{\zeta_i}} - 2^{\boldsymbol{m}_{\zeta_i} - \boldsymbol{w}_i + 1},$$

will satisfy the wordlength constraints.

We can compute $\boldsymbol{m}_{\zeta_i}$ with

$$\boldsymbol{m}_{\zeta_i} = \left\lceil \log_2 \left(\langle\!\langle H_\zeta \rangle\!\rangle \, \bar{\boldsymbol{u}}\right)_i - \log_2 \left(1 - 2^{1 - \boldsymbol{w}_{\zeta_i}}\right) \right\rceil.$$

Taking the quantization errors into account

The exact filter $\mathcal{H}_\zeta$ is:

$$\mathcal{H}_\zeta \quad \left\{ \begin{array}{rcl} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{\zeta}(k) & = & \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{C} \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{D} \end{pmatrix} \boldsymbol{u}(k) \end{array} \right.$$

Taking the quantization errors into account

The actually implemented filter $\mathcal{H}_\zeta^\Diamond$ is:

$$\mathcal{H}_\zeta{}^\Diamond \begin{cases} \boldsymbol{x}^\Diamond(k+1) & = & \Diamond_{\boldsymbol{m}_x}\left(\boldsymbol{A}\boldsymbol{x}^\Diamond(k) + \boldsymbol{B}\boldsymbol{u}(k)\right) \\ \boldsymbol{\zeta}^\Diamond(k) & = & \Diamond_{\boldsymbol{m}_\zeta}\left(\begin{pmatrix}\boldsymbol{I}\\ \boldsymbol{C}\end{pmatrix}\boldsymbol{x}^\Diamond(k) + \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{D}\end{pmatrix}\boldsymbol{u}(k)\right) \end{cases}$$

where $\Diamond_m$ is some operator ensuring faithful rounding:

$$|\Diamond_m(x) - x| \leqslant 2^{m-w+1}.$$

Taking the quantization errors into account

The actually implemented filter $\mathcal{H}_\zeta^\Diamond$ is:

$$\mathcal{H}_\zeta{}^\Diamond \left\{ \begin{array}{rcl} \boldsymbol{x}^\Diamond(k+1) & = & \Diamond_{\boldsymbol{m}_x}\left(\boldsymbol{A}\boldsymbol{x}^\Diamond(k) + \boldsymbol{B}\boldsymbol{u}(k)\right) \\ \boldsymbol{\zeta}^\Diamond(k) & = & \Diamond_{\boldsymbol{m}_\zeta}\left(\begin{pmatrix}\boldsymbol{I}\\ \boldsymbol{C}\end{pmatrix}\boldsymbol{x}^\Diamond(k) + \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{D}\end{pmatrix}\boldsymbol{u}(k)\right) \end{array}\right.$$

where $\Diamond_m$ is some operator ensuring faithful rounding:

$$|\Diamond_m(x) - x| \leqslant 2^{m-w+1}.$$

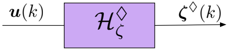
It holds

$$\mathcal{H}_\zeta^\Diamond \left\{ \begin{array}{rcl} \boldsymbol{x}^\Diamond(k+1) & = & \boldsymbol{A}\boldsymbol{x}^\Diamond(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{\varepsilon}_x(k) \\ \boldsymbol{\zeta}^\Diamond(k) & = & \begin{pmatrix}\boldsymbol{I}\\ \boldsymbol{C}\end{pmatrix}\boldsymbol{x}^\Diamond(k) + \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{D}\end{pmatrix}\boldsymbol{u}(k) + \begin{pmatrix}\boldsymbol{\varepsilon}_x(k)\\ \boldsymbol{\varepsilon}_y(k)\end{pmatrix} \end{array}\right.$$

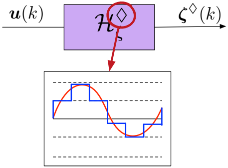
with

$$|\boldsymbol{\varepsilon}_x(k)| \leqslant 2^{\boldsymbol{m}_x - \boldsymbol{w}_x + 1} \quad \text{and} \quad |\boldsymbol{\varepsilon}_y(k)| \leqslant 2^{\boldsymbol{m}_y - \boldsymbol{w}_y + 1}.$$

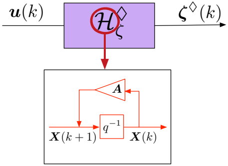
Implemented filter decomposition



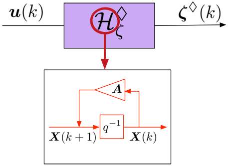
Implemented filter decomposition

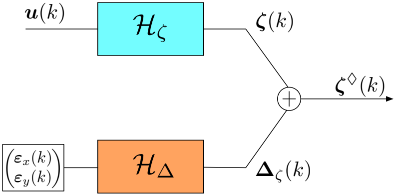


Implemented filter decomposition

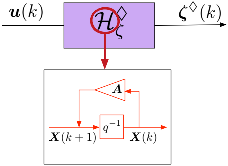


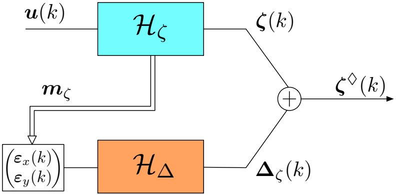
Implemented filter decomposition





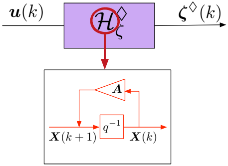
Implemented filter decomposition

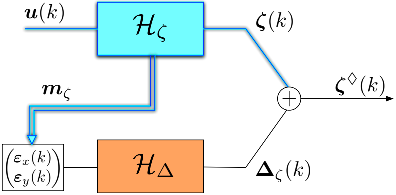




! Filter $\mathcal{H}_\Delta$ depends on the MSBs of filter $\mathcal{H}_\zeta$

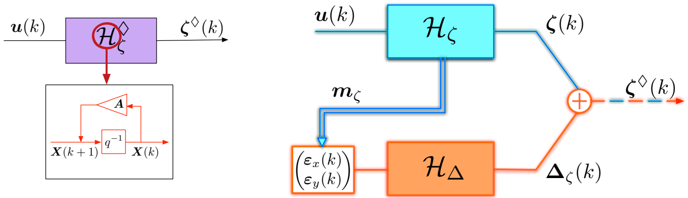
Implemented filter decomposition





! Filter $\mathcal{H}_\Delta$ depends on the MSBs of filter $\mathcal{H}_\zeta$

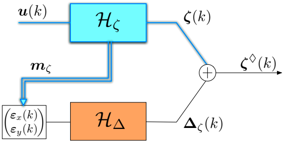
Implemented filter decomposition



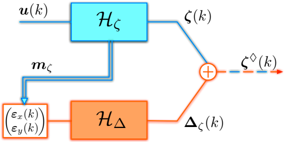
! Filter $\mathcal{H}_\Delta$ depends on the MSBs of filter $\mathcal{H}_\zeta$

Two step approach

Step 1 Determine the MSBs $\boldsymbol{m}_\zeta$ for the exact filter $\mathcal{H}_\zeta$, applying the WCPG theorem;



Step 2 Compute the error-filter, induced by the format $\boldsymbol{m}_\zeta$ and deduce the FPF of the implemented filter $\mathcal{H}_\zeta^\diamond$



MSB computation error analysis

$$\boldsymbol{m}_{\zeta_i} = \left\lceil \log_2 \left(\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle \, \bar{\boldsymbol{u}} \right)_i + \log_2 \left(1 - 2^{1-\boldsymbol{w}_i} \right) \right\rceil$$

MSB computation error analysis

$$\boldsymbol{m}_{\zeta_i} = \left\lceil \log_2 \big(\underbrace{\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle}_{\widehat{\langle\langle \mathcal{H}_\zeta \rangle\rangle} + \boldsymbol{\varepsilon}_{WCPG}} \bar{\boldsymbol{u}} \big)_i + \log_2 \left(1 - 2^{1-\boldsymbol{w}_i} \right) \right\rceil$$

MSB computation error analysis

$$\boldsymbol{m}_{\zeta_i} = \left\lceil \log_2 \left(\underbrace{\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle}_{\widehat{\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle} + \boldsymbol{\varepsilon}_{WCPG}} \bar{\boldsymbol{u}} \right)_i + \log_2 \left(1 - 2^{1-\boldsymbol{w}_i} \right) \right\rceil$$

$$\widehat{\boldsymbol{m}}_{\zeta_i} = \left\lceil \log_2 \left(\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle \bar{\boldsymbol{u}} \right)_i + \log_2 \left(1 - 2^{1-\boldsymbol{w}_i} \right) + \log_2 \left(1 + \frac{\left(\boldsymbol{\varepsilon}_{WCPG} \cdot \bar{\boldsymbol{u}} \right)_i}{\left(\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle \bar{\boldsymbol{u}} \right)_i} \right) \right\rceil$$

MSB computation error analysis

$$\boldsymbol{m}_{\zeta_i} = \left\lceil \log_2 \underbrace{\left(\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle\right.}_{\widehat{\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle} + \boldsymbol{\varepsilon}_{WCPG}} \left. \bar{\boldsymbol{u}}\right)_i + \log_2\left(1 - 2^{1-\boldsymbol{w}_i}\right)\right\rceil$$

$$\begin{aligned}\widehat{\boldsymbol{m}}_{\zeta_i} = &\left\lceil \log_2 \left(\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle \bar{\boldsymbol{u}}\right)_i + \log_2\left(1 - 2^{1-\boldsymbol{w}_i}\right)\right. \\ &\left. + \log_2\left(1 + \frac{\left(\boldsymbol{\varepsilon}_{WCPG} \cdot \bar{\boldsymbol{u}}\right)_i}{\left(\langle\!\langle \mathcal{H}_\zeta \rangle\!\rangle \bar{\boldsymbol{u}}\right)_i}\right)\right\rceil \\ = &\; \boldsymbol{m}_{\zeta_i} + \underbrace{\lceil \ldots \rceil}_{\in\{0,1\}}\end{aligned}$$

MSB computation error analysis

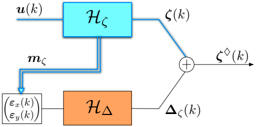
$$\boldsymbol{m}_{\zeta_i} = \left\lceil \log_2 \Big(\underbrace{\langle\langle \mathcal{H}_\zeta \rangle\rangle}_{\widehat{\langle\langle \mathcal{H}_\zeta \rangle\rangle} + \boldsymbol{\varepsilon}_{WCPG}} \bar{\boldsymbol{u}} \Big)_i + \log_2 \left(1 - 2^{1-\boldsymbol{w}_i} \right) \right\rceil$$

$$\begin{aligned} \widehat{\boldsymbol{m}}_{\zeta_i} &= \Big\lceil \log_2 \left(\langle\langle \mathcal{H}_\zeta \rangle\rangle \bar{\boldsymbol{u}} \right)_i + \log_2 \left(1 - 2^{1-\boldsymbol{w}_i} \right) \\ &\qquad + \log_2 \left(1 + \frac{\left(\boldsymbol{\varepsilon}_{WCPG} \cdot \bar{\boldsymbol{u}}\right)_i}{\left(\langle\langle \mathcal{H}_\zeta \rangle\rangle \bar{\boldsymbol{u}} \right)_i} \right) \Big\rceil \\ &= \boldsymbol{m}_{\zeta_i} + \underbrace{\lceil \ldots \rceil}_{\in \{0,1\}} \end{aligned}$$

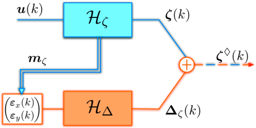
Adjust error term ε_{WCPG} in order to be at most **off by one**.

Algorithm

Step 1 Determine the MSBs $\boldsymbol{m}_\zeta$ for the exact filter $\mathcal{H}_\zeta$, applying the WCPG theorem;



Step 2 Compute the error-filter $\mathcal{H}_\Delta$, induced by the format $\boldsymbol{m}_\zeta$ and deduce the MSB $\boldsymbol{m}_\zeta^\Diamond$ of the implemented filter;



Step 3 If $\boldsymbol{m}_{\zeta_i}^\Diamond == \boldsymbol{m}_{\zeta_i}$ then return $\boldsymbol{m}_{\zeta_i}^\Diamond$ otherwise $\boldsymbol{m}_{\zeta_i} \leftarrow \boldsymbol{m}_{\zeta_i} + 1$ and go to Step 2.

Numerical examples

Example:

- Random filter with 6 states, 1 input, 3 outputs
- $\bar{u} = 3.7776$, wordlengths set to 7 bits

$$m_\zeta = (4,\ \ 4,\ \ 4,\ \ 4,\ \ 2,\ \ 3,\ \ 6,\ \ 5,\ \ 5)$$

$$m_\zeta^\diamondsuit = (4,\ \ 5,\ \ 4,\ \ 4,\ \ 2,\ \ 3,\ \ 6,\ \ 5,\ \ 5)$$

Numerical examples

Example:

- Random filter with 6 states, 1 input, 3 outputs
- $\bar{u} = 3.7776$, wordlengths set to 7 bits

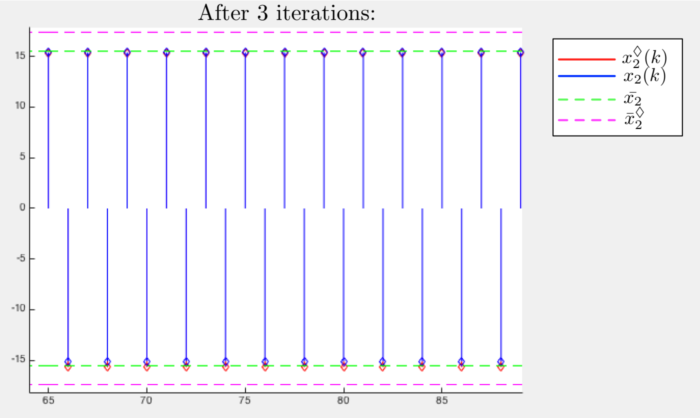
$$m_\zeta = (4,\ 4,\ 4,\ 4,\ 2,\ 3,\ 6,\ 5,\ 5)$$

$$m_\zeta^\Diamond = (4,\ 5,\ 4,\ 4,\ 2,\ 3,\ 6,\ 5,\ 5)$$

After 3 iterations:

$$m_\zeta^\Diamond = (4,\ 5,\ 5,\ 4,\ 3,\ 4,\ 6,\ 5,\ 5)$$

Numerical examples



Conclusion and Perspectives

Conclusion

- Rigorous procedure for Fixed-Point Formats determination
- Filter computation errors are taken into account, ensuring that no overflow occurs
- Multiple-wordlength paradigm

Perspectives

- Integrate into optimization procedures in automatic workflow
- Solve off-by-one problem

Thank you!
Questions?