DSP System Design using Fixed-Point Arithmetic

Gabriel Caffarena Fernández

December 2009

Seminario en “Avances en Sistemas Electrónicos”
E.T.S.I.Telecomuncación, U.P.M., Madrid

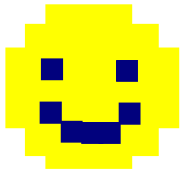
Agenda

- Introduction
- Automatic Quantization Techniques
- Fast quantization
- Non-linear systems
- Implementation issues
- Bibliography

Introduction

- Digital Systems represent numbers using finite precision
  - Microprocessors
  - Custom Hardware: ASIC and FPGAs
- Custom hardware make use of fixed-point arithmetic
  - Low cost
  - Low power
  - High speed
- Fixed-point design is not trivial: Optimization



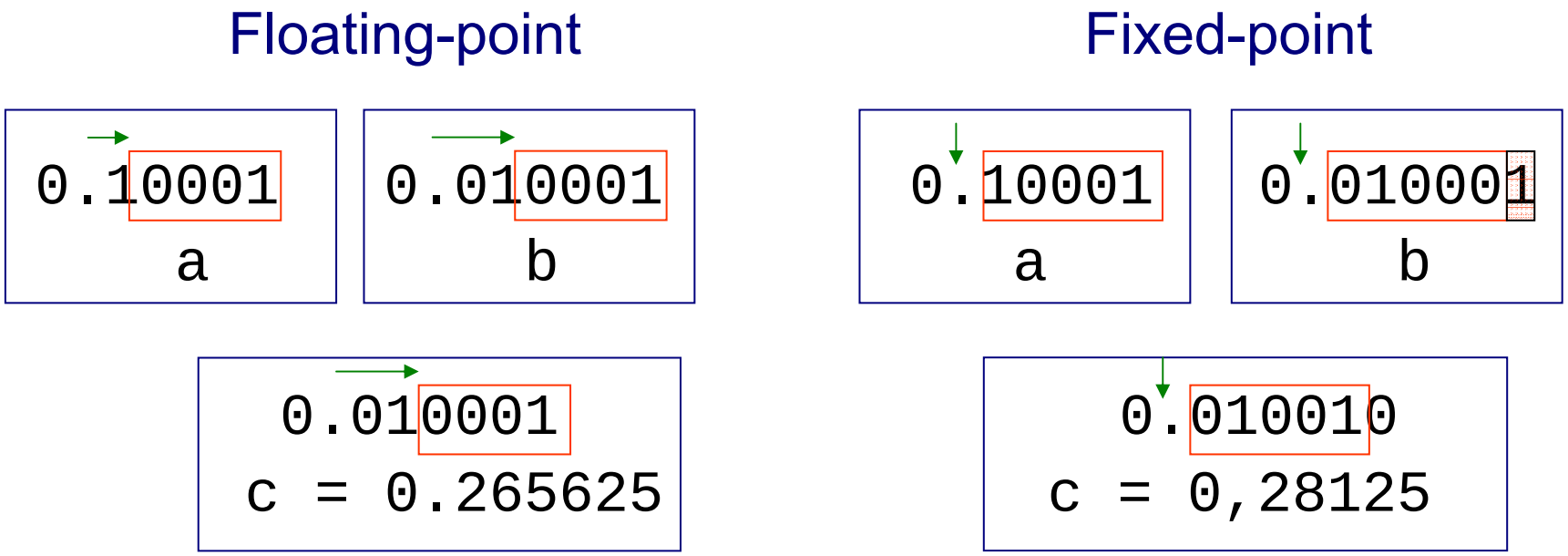


Introduction

Fixed-point Arithmetic

- Floating-point vs. fixed-point

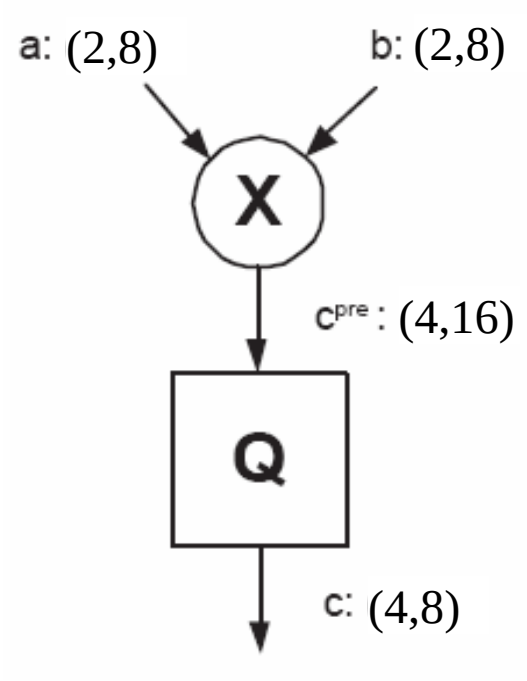
  a=b-c → a=0.53125 b=0.265625

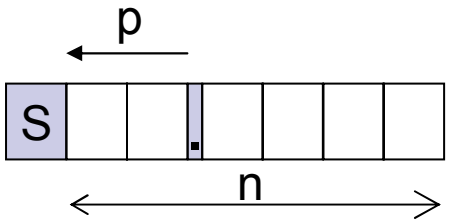


Introduction

Quantization Noise

- Modelled as a uniform white additive noise





(p,n)

$$\sigma_i^2 = \frac{2^{2p_i}}{12}\left(2^{-2n_i}\right)$$
$$\mu_i = 2^{p_i-1}\left(2^{-n_i}\right)$$

∞→8

$$\sigma_i^2 = \frac{2^{2p_i}}{12}\left(2^{-2n_i} - 2^{-2n_i^{pre}}\right)$$
$$\mu_i = 2^{p_i-1}\left(2^{-n_i} - 2^{-n_i^{pre}}\right).$$

16→8

Ref. 1

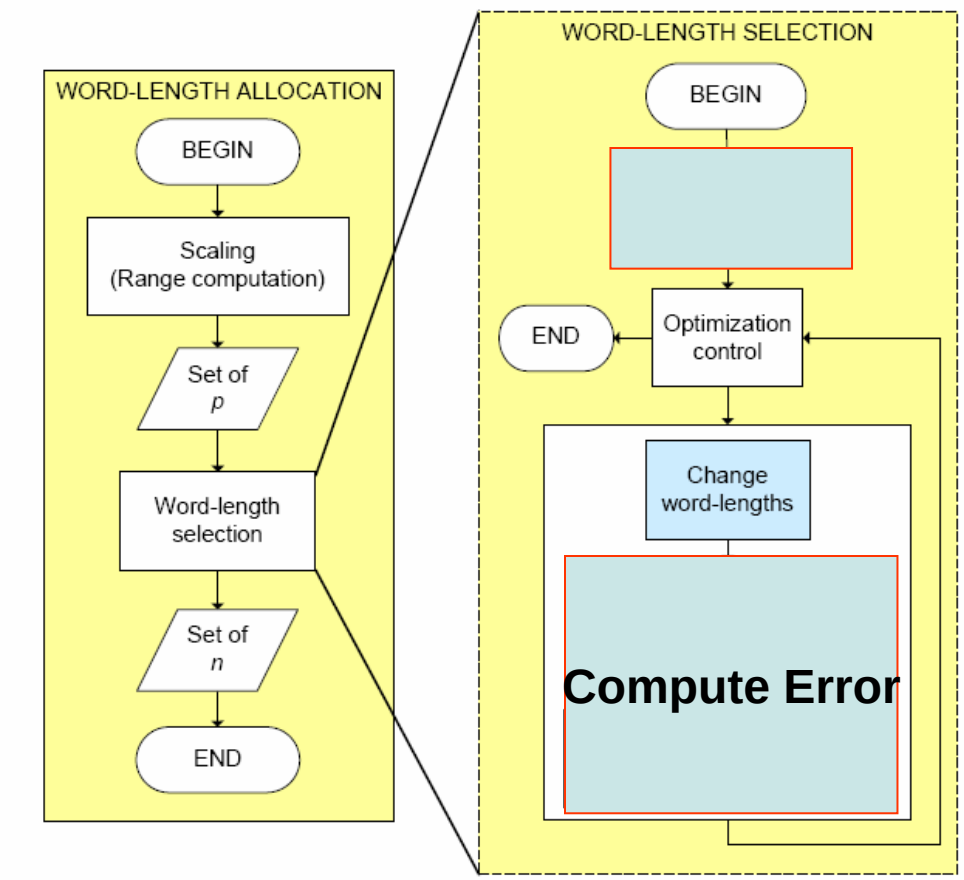
Introduction

Uniform vs. Multiple Quantization

- Uniform word-length paradigm (UWL)
  - All signals/variables have the same “n”
  - Fast Quantization
  - Far from optimal results
- Multiple word-length paradigm (MWL)
  - Custom FxP format for each single signal/variable
  - Complex optimization process
    - Quantization
    - Architecture design
  - High cost reductions (50% compared to UWL)

Ref. 1,2,3

Automatic quantization techniques



1. Scaling
   - *p*
2. Word-length selection
   - *n*

Automatic quantization techniques

Scaling and Word-length selection

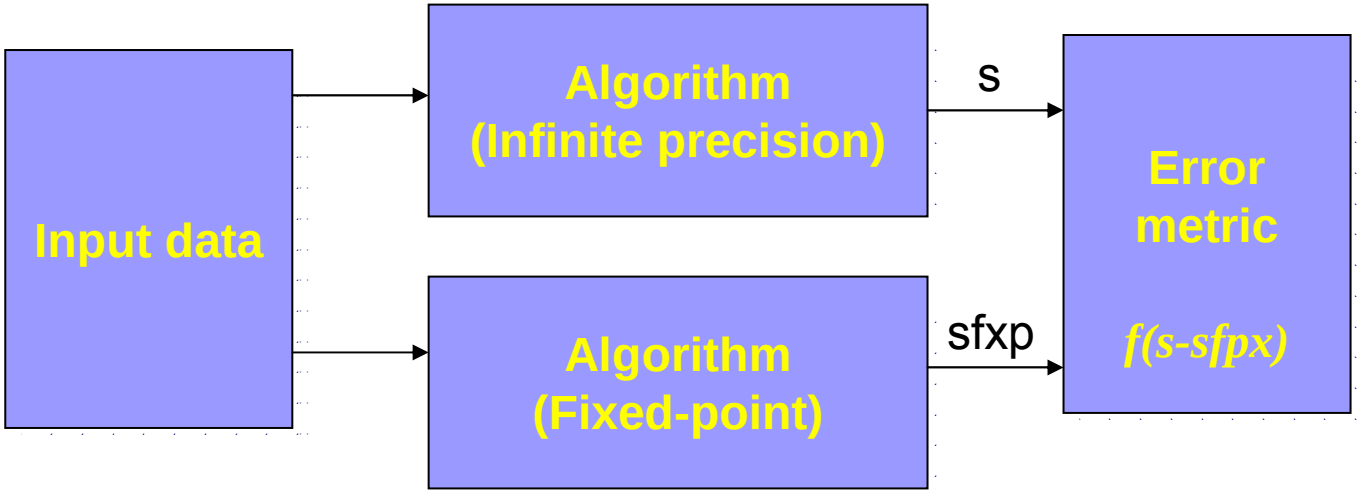
- Scaling
  - Compute maximum absolute value for all the variables
    - Using simulations with collecting capabilities: **C/C++, matlab…**
    - Using semi-analytical techniques: **interval/affine arithmetic…**
  - $$p_i = \left\lfloor log_2 \left(max \left\{ |\overline{range_{s_i}}|, |\underline{range_{s_i}}| \right\} \right) \right\rfloor + 1$$

- Word-length selection
  - Fix value for $\boldsymbol{p=[p_0, p_1, p_2,.., p_{|S|-1}]}$ (|S| signals)
  - "Try" different combinations of $\boldsymbol{n=[n_0, n_1,.., n_{|S|-1}]}$
    - Apply optimization techniques
    - Check noise at the output in each optimization iteration

Ref. 2,3

Automatic quantization techniques

Error computation



- $SQNR = 10log(P_s / P_{fxp\_err})$
- Peak value
  - Confidence interval
- Application specific metrics: BER…

Automatic quantization techniques
Uniform WL Quantization

- Uniform optimization
  1. Our aim is to find a solution that complies with an error constraint
     - SQNR > $SQNR_{goal}$
  2. Find minimum value of n_{global} ($n_i = n_{global}$) that complies with noise constraint
- Binary search

  64,32,16,24,20,22,21
- Fast, but far from optimal

Ref. 2,3

Automatic quantization techniques

Multiple WL Quantization

- Gradient-descent optimization
  1. Our aim is to find a solution that complies with an error constraint (SQNR > $SQNR_{goal}$)
  2. Apply uniform quantization
  3. For each variable *i*
     1. $n_i = n_i$ - 1 bit
     2. Compute error power
     3. $n_i = n_i$ + 1 bit
  4. $n_k = n_k$ + 1 bit, where *k* is the variable that produced the smallest error power that complied with error constraint
  5. Repeat 3 until there is no possible word-length reduction without violating error constraint

Ref. 2,3

Automatic quantization techniques

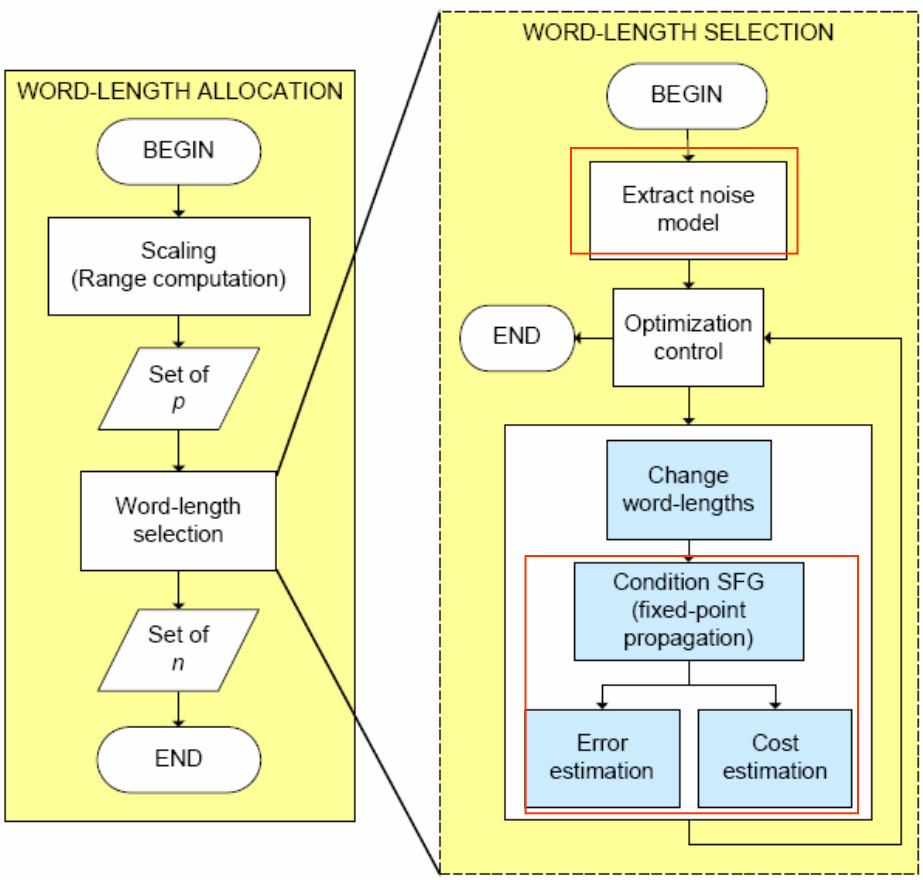
Implementation results:UWL vs MWL

Bench.	σ^2	$GRAD$ (%)	$GRAD_2$ (%)	SA (%)
ITU	10^{-1}	63.97	63.97 / 0.00*	65.33 / 3.78
	10^{-2}	63.41	63.41 / 0.00	64.08 / 1.83
	10^{-3}	63.03	63.03 / 0.00	64.69 / 4.51
LAT_3	10^{-3}	71.66	72.91 / 4.39	72.97 / 4.60
	10^{-4}	63.84	69.65 / 16.37	69.76 / 16.37
	10^{-5}	61.07	64.63 / 9.13	64.63 / 9.13
IIR_4	10^{-3}	64.96	67.11 / 6.16	68.67 / 10.61
	10^{-4}	62.31	62.02 / -0.77	64.07 / 4.67
	10^{-5}	54.63	56.66 / 4.46	58.86 / 9.32
FIR_8	10^{-3}	45.70	61.10 / 28.36	61.56 / 29.21
	10^{-4}	44.09	42.25 / -3.29	48.85 / 8.51
	10^{-5}	33.64	36.11 / 3.73	40.87 / 10.90
All		57.69	60.24 / 5.69	62.03 / 9.45
* with respect to UWL / with respect to $GRAD$				

Up to 72%

Ref. 2,3

Fast quantization



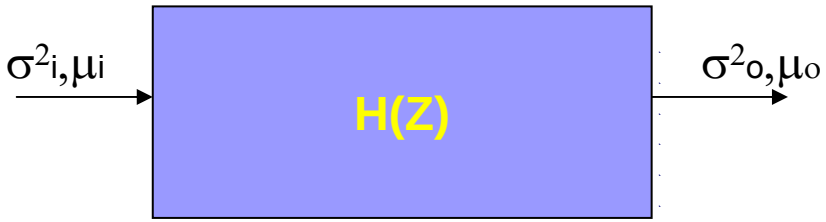
Ref. 3

- Fast estimation of the error
- It requires noise parameterization
- It requires conditioning
  - Propagation

Fast quantization

Noise at the output: LTI systems

- Noise propagation



$$\sigma_o^2 = \sigma_i^2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \mathbf{H}\ e^{j\Omega}) \right|^2 d\Omega$$

$$\mu_o = \mu_i \cdot \mathbf{H}\,(1)$$

$$P_o = \sigma_o^2 + (\mu_o)^2$$

Ref. 3,4,5

Fast quantization

Noise at the output: LTI systems

- Output noise?

$$
\begin{aligned}
\sigma_o^2 &= \sum_{i=0}^{|S|-1} \sigma_i^2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| G_i(e^{j\Omega}) \right|^2 d\Omega \\
&= \sum_{i=0}^{|S|-1} \sigma_i^2 \cdot \sum_{j=0}^{\infty} g_i^2[j] \\
\mu_o &= \sum_{i=0}^{|S|-1} \mu_i \cdot G_i(1) \\
&= \sum_{i=0}^{|S|-1} \mu_i \cdot \sum_{j=0}^{\infty} g_i[j]) \\
P_o &= \sigma_o^2 + (\mu_o)^2
\end{aligned}
$$

- The impulse response of each variable (***g[n]***) is related to the output noise
- Easy to compute
- Isolation between ***g*** and σ, μ
- Automation?
- ***SNR = 10log(Pseñal/Po)***

Fast quantization

Noise at the output: LTI systems

$$
\begin{aligned}
\sigma_o^2 &= \sum_{i=0}^{|S|-1} \sigma_i^2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| G_i(e^{j\Omega}) \right| d\Omega \\
&= \sum_{i=0}^{|S|-1} \sigma_i^2 \cdot \sum_{j=0}^{\infty} g_i^2[j] \\
\mu_o &= \sum_{i=0}^{|S|-1} \mu_i \cdot G_i(1) \\
&= \sum_{i=0}^{|S|-1} \mu_i \cdot \sum_{j=0}^{\infty} g_i[j]) \\
P_o &= \sigma_o^2 + (\mu_o)^2
\end{aligned}
$$

$$
\begin{aligned}
\sigma_o^2 &= \boldsymbol{\sigma^2} \cdot \boldsymbol{v}^T \\
\mu_o &= \boldsymbol{\mu} \cdot \boldsymbol{m}^T \\
P_o &= \boldsymbol{\sigma^2} \cdot \boldsymbol{v}^T + \left(\boldsymbol{\mu} \cdot \boldsymbol{m}^T \right)^2 \\
\boldsymbol{\sigma^2} &\equiv \left\langle \sigma_0^2, \ldots, \sigma_{|S|-1}^2 \right\rangle \\
\boldsymbol{\mu} &\equiv \left\langle \mu_0, \ldots, \mu_{|S|-1} \right\rangle \\
\boldsymbol{v} &\equiv \left\langle \sum_{j=0}^{\infty} g_0^2[j], \ldots, \sum_{j=0}^{\infty} g_{|S|-1}^2[j] \right\rangle \\
\boldsymbol{m} &\equiv \left\langle \sum_{j=0}^{\infty} g_0[j], \ldots, \sum_{j=0}^{\infty} g_{|S|-1}[j] \right\rangle
\end{aligned}
$$

Ref. 3

Fast quantization
Noise parameterization

- For each variable compute the impulse response ***g***
- Compute the summation of ***g[n]^2***
  - We get vector **v**
- Compute the summation of ***|g[n]|***
  - We get vector **m**
- For each value of vector ***n*** compute
  - **σ^2** y μ
  - Now, we can compute the power of the output error
  - 1000 faster than a simulation-based approach

Fast quantization
Estimation results (I)

PERFORMANCE OF THE ESTIMATION METHOD: PRECISION.

Benchmark	Estimation error [120,100)[1] dB (dB)[2]	(%)[3]	[100,80) dB (dB)	(%)	[80,60) dB (dB)	(%)	[60,40] dB (dB)	(%)
RGB	0.11	0.24	0.09	0.09	0.07	0.17	0.07	0.44
$IDCT_8$	0.11	0.11	0.08	0.42	0.21	0.88	0.27	0.68
IIR_2^*	0.04	0.03	0.04	0.04	0.06	0.74	0.04	0.09
LAT_3^*	0.24	0.69	0.18	0.33	0.20	0.15	0.19	0.46
DEL_6^*	0.03	0.01	0.02	0.03	0.03	0.16	0.16	1.16

[1] Error constraint

[2] $|10log(\frac{P_{ref}}{P_{est}})|$ – (max)

[3] $|100(\frac{P_{ref}-P_{est}}{P_{ref}})|$ (average)

* It contains loops

Ref. 3

Fast quantization
Estimation results (II)

PERFORMANCE OF THE ESTIMATION METHOD: COMPUTATION TIME.

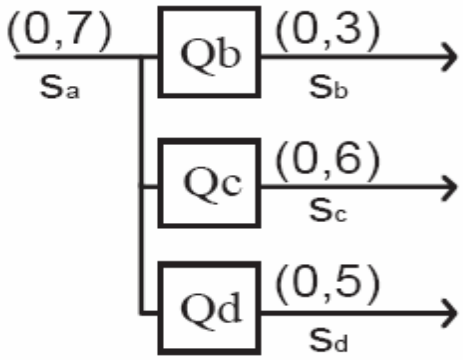
Bench.	FxP Samples	Param. Samples	Param. time (secs)+	No. of estimates (mean)	Estimation-based optim. (secs)+	Simulation-based optim. (secs)+	Speed-up
RGB	20000	1	0.00016	141	0.03	76	×3205
$IDCT_8$	20000	1	0.00031	4575	5.77	13774.81	×2468
IIR_2^*	20000	5000	0.88	19	0.02	4.41	×270
LAT_3^*	20000	20000	10.80	2276	0.74	2381.51	×3222
DEL_6^*	20000	5000	6.31	3930	3.47	11206.08	×3235

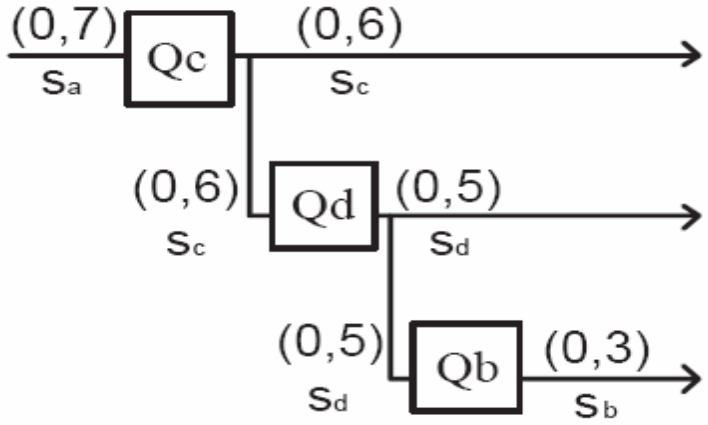
* It contains loops

+ Using 1.66 GHz Intel Core Duo processor and 1 GB of RAM

Fast quantization
Advance issues

- Effect of signal forks





  - Treat each fork output as an different signal
  - Values of *m* and *v* change depending of fixed-point formats

Ref. 2

Non-linear systems

- Impulse response is not time-invariant
- Can be average the “impulse responses”
- Affine arithmetic (AA) provides a means to do that

Ref. 3

Non-linear systems
Affine Arithmetic

- Analysis of error propagation

$$\hat{x} = x_0 + \sum_{i=1}^{N} x_i \epsilon_i$$

$$\begin{aligned} \hat{x} \pm \hat{c} &= x_0 \pm c_0 + \sum_{i=1}^{N} x_i \epsilon_i \\ \hat{x} \pm \hat{y} &= x_0 \pm y_0 + \sum_{i=1}^{N} (x_i \pm y_i) \epsilon_i \\ \hat{c} \cdot \hat{x} &= c_0 x_0 + \sum^{N} c_0 x_i \epsilon_i \end{aligned}$$

$$f(\hat{x}, \hat{y}) \approx f(x_0, y_0) + \sum_{i=1}^{N} \left(\frac{\delta f(x_0, y_0)}{\delta \hat{x}} \cdot x_i + \frac{\delta f(x_0, y_0)}{\delta \hat{y}} \cdot y_i \right) \epsilon_i$$

Ref. 3,4

Non-linear systems

Error estimation

- $\varepsilon_{i,j}$ represents an quantization error introduced by signal I at time step j
- Output Y[n] of an algorithm with |S| signals

$$\hat{Y}[n] = Y_0[n] + \sum_{i=0}^{|S|-1} \sum_{j=0}^{n-1} Y_{i,j}[n]\epsilon'_{i,j}$$

- The quantization error is equal to

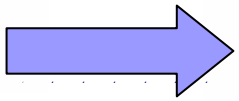
$$\hat{E}rr_Y[n] = Y_0[n] - \hat{Y}[n] = -\sum_{i=0}^{|S|-1} \sum_{j=0}^{n-1} Y_{i,j}[n]\epsilon'_{i,j}$$

- Can we use these error terms to estimate the power of the quantization error?

Ref. 3,4

Non-linear systems
Error estimation

$$
\begin{aligned}
P_o &= \frac{1}{K}\left(\boldsymbol{\sigma^2}\cdot \boldsymbol{v}^T + \boldsymbol{\mu}\cdot \boldsymbol{M}\boldsymbol{\mu}^T\right)\\
\mu_o &= \frac{1}{K}\left(\boldsymbol{\mu}\cdot \boldsymbol{m}^T\right)\\
\sigma_o^2 &= P_o - \mu_o^2
\end{aligned}
$$



$$
\begin{aligned}
\boldsymbol{v} &\equiv \left\langle \sum_{n=0}^{M-1}\sum_{j=0}^{n-1} Y_{0,j}^2[n], \ldots, \sum_{n=0}^{M-1}\sum_{j=0}^{n-1} Y_{|S|-1,j}^2[n] \right\rangle\\
\boldsymbol{m} &\equiv \left\langle \sum_{n=0}^{M-1}\sum_{j=0}^{n-1} Y_{0,j}[n], \ldots, \sum_{n=0}^{M-1}\sum_{j=0}^{n-1} Y_{|S|-1,j}[n] \right\rangle\\
\boldsymbol{M} &\equiv \begin{bmatrix} m_{0,0} & \cdots & m_{|S|-1,0} \\ & \ddots & \\ m_{0,|S|-1} & \cdots & m_{|S|-1,|S|-1} \end{bmatrix}\\
m_{i_1,i_2} &= \sum_{n=0}^{M-1}\left(\sum_{j_1=0}^{n-1} Y_{i_1,j_1}[n]\sum_{j_2=0}^{n-1} Y_{i_2,j_2}[n]\right)
\end{aligned}
$$

Ref. 3

Non-linear Systems
Noise parameterization

- Perform an AA simulation adding a new error term for each variable at each time step during K time steps
- At each time step:
  - Compute and accumulate $\Sigma Y_{i,j}^{2} \rightarrow \boldsymbol{v}$
  - Compute and accumulate $\Sigma Y_{i,j} \rightarrow \boldsymbol{m}$
  - Compute and accumulate $\Sigma Y_{i,j} \times \Sigma Y_{i,j} \rightarrow \boldsymbol{m}$
- Average ***v***, ***m*** and ***M***

Non-linear Systems
Estimation results (I)

PERFORMANCE OF THE ESTIMATION METHOD: PRECISION.

Benchmark	Estimation error [120,100)[1] dB (dB)[2]	(%)[3]	[100,80) dB (dB)	(%)	[80,60) dB (dB)	(%)	[60,40] dB (dB)	(%)
RGB	0.11	0.24	0.09	0.09	0.07	0.17	0.07	0.44
$IDCT_8$	0.11	0.11	0.08	0.42	0.21	0.88	0.27	0.68
IIR_2^*	0.04	0.03	0.04	0.04	0.06	0.74	0.04	0.09
LAT_3^*	0.24	0.69	0.18	0.33	0.20	0.15	0.19	0.46
DEL_6^*	0.03	0.01	0.02	0.03	0.03	0.16	0.16	1.16
$VEC_{3\times3}$	0.07	0.54	0.07	0.11	0.06	0.50	0.09	0.72
$VEC_{8\times8}$	0.05	0.57	0.04	0.40	0.04	0.57	0.13	1.19
EQ	0.27	0.98	0.24	0.71	0.29	0.17	0.18	1.52
POW^*	0.39	5.00	0.17	1.55	0.76	5.96	1.12	12.12
LMS_1^*	0.09	0.41	0.14	0.90	0.16	1.74	0.82	6.96
LMS_2^*	0.09	0.46	0.08	0.24	0.15	0.78	0.92	3.73
LMS_5^*	0.09	0.46	0.08	0.07	0.13	1.08	1.09	5.51
VOL_3^*	1.14	3.33	0.49	1.84	0.81	6.70	1.43	16.67
All	1.14	0.20	0.49	0.09	0.81	1.26	1.43	3.52

[1] Error constraint

[2] $|10log(\frac{P_{ref}}{P_{est}})|$ – (max)

[3] $|100(\frac{P_{ref}-P_{est}}{P_{ref}})|$ (average)

* It contains loops

Ref. 3

Non-linear Systems
Estimation results (II)

PERFORMANCE OF THE ESTIMATION METHOD: COMPUTATION TIME.

Bench.	FxP Samples	Param. Samples	Param. time (secs)+	No. of estimates (mean)	Estimation-based optim. (secs)+	Simulation-based optim. (secs)+	Speed-up
RGB	20000	1	0.00016	141	0.03	76	×3205
$IDCT_8$	20000	1	0.00031	4575	5.77	13774.81	×2468
IIR_2^*	20000	5000	0.88	19	0.02	4.41	×270
LAT_3^*	20000	20000	10.80	2276	0.74	2381.51	×3222
DEL_6^*	20000	5000	6.31	3930	3.47	11206.08	×3235
$VEC_{3\times3}$	20000	20000	59	150	0.03	66.86	×2122
$VEC_{8\times8}$	20000	20000	330	1739	1.72	2331.79	×1377
EQ	16000	16000	61.64	231	0.12	105.78	×904
POW^*	20000	20000	546.14	97	0.02	21.93	×1048
LMS_1^*	5000	5000	908.02	712	0.42	163.73	×394
LMS_2^*	5000	5000	592.11	1032	0.94	310.93	×331
LMS_5^*	5000	5000	1646.38	2547	7.26	1611.46	×221
VOL_3^*	5000	5000	212.72	673	0.29	151.13	×526
All	-	-	-	-	-	-	×1486

* It contains loops

\+ Using 1.66 GHz Intel Core Duo processor and 1 GB of RAM

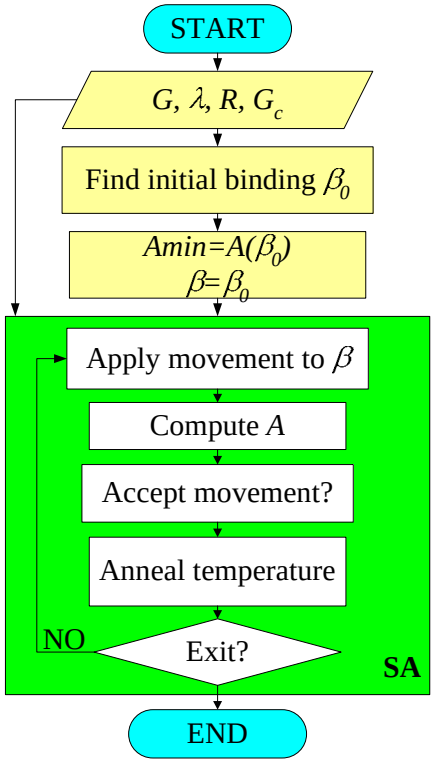
Ref. 3

Implementation Issues

- MWL achieves significant cost reductions
- Resource-sharing and MWL?
- Embedded resources (MULT18x18)?

Implementation Issues

Resource-sharing (I)

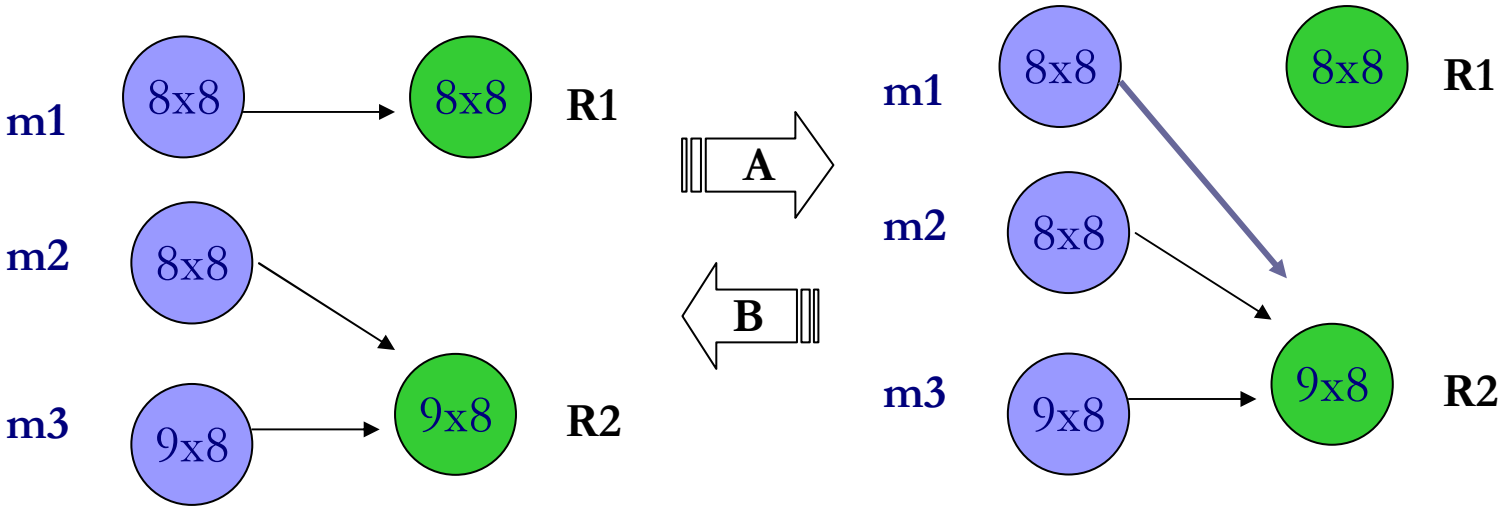


- G Dataflow graph
- λ latency
- R Set of resources
- Gc Compatibility graph

Implementation Issues

Resource-sharing (II)

A. Map operation to another **shared** resource
B. Map operation to another **non-shared** resource

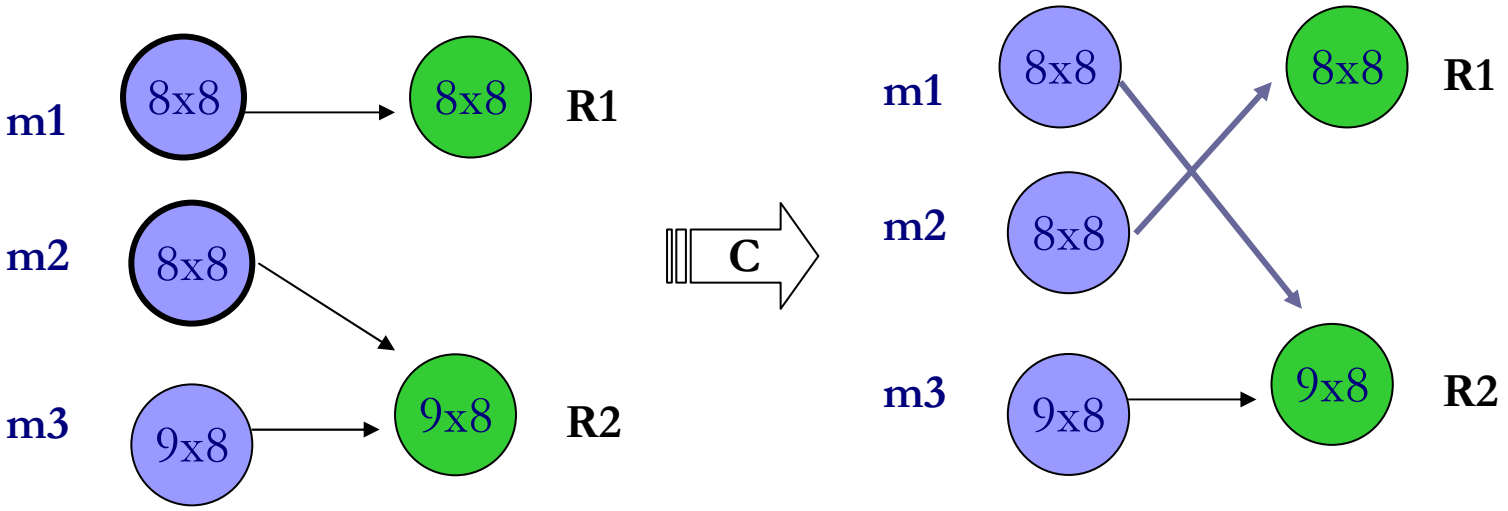


Ref. 3,6

Implementation Issues

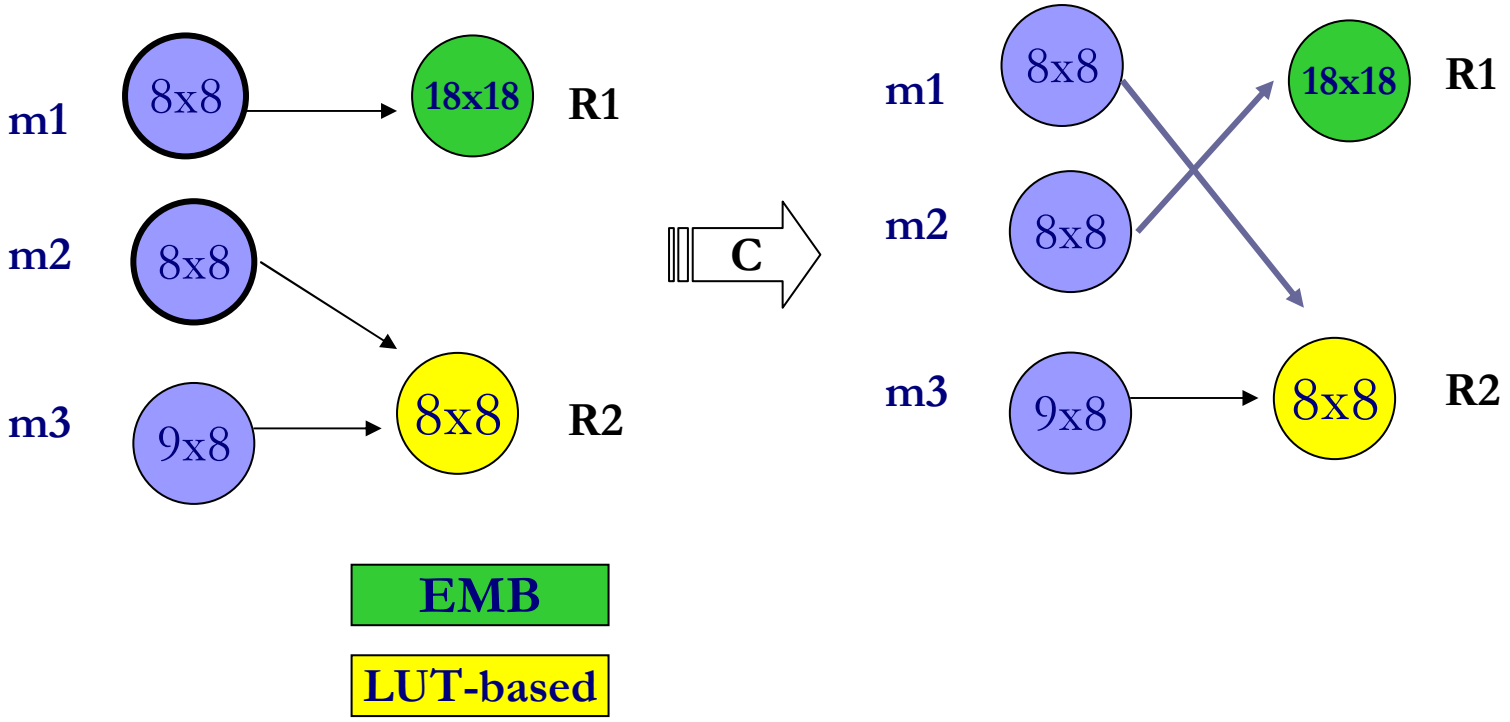
Resource-sharing (III)

C. Swap two operations mapping



Resource-sharing (IV)

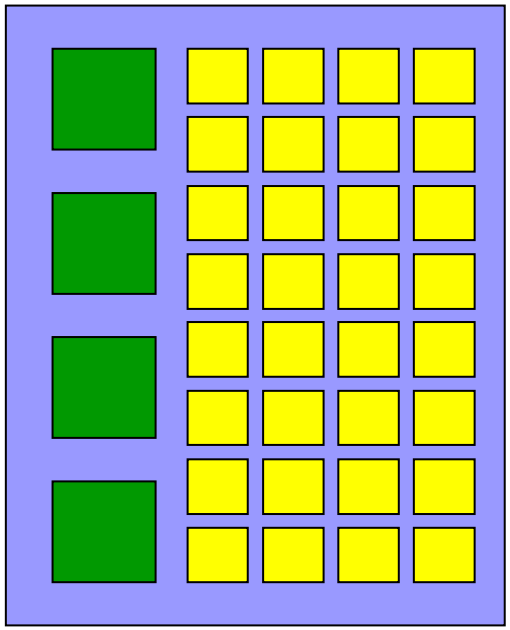
- Heterogeneous implementations



Implementation Issues

Embedded resources (I)

$$\mathbf{A} = (\#\mathrm{LUTs}/\mathrm{LUT}_{\mathrm{MAX}},\ \#\mathrm{MULT}/\mathrm{MULT}_{\mathrm{MAX}})$$



4 MULTs

32 LUTs

Ref. 3,6

∞-norm

$\| A \|_{\infty} = \max(A_i)$ → **Replication**

1-norm

$\| A \|_{1} = \sum A_i$ → **Resource usage**

+-norm

$\| A \|_{+} = K \| A \|_{\infty} + \| A \|_{1}$

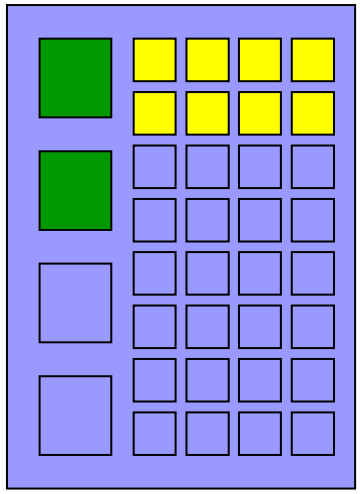
Replication and **Resource usage**

K depends on the number and type of resources

Implementation Issues

Embedded resources (II)

Implementation 1



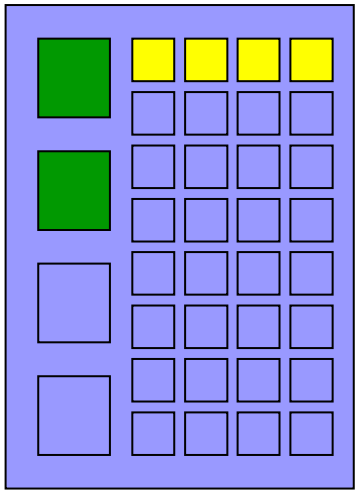
$A = (0.25, 0.5)$

$\| A \|_{\infty} = 0.5 \rightarrow 2$

$\| A \|_{1} = 0.75$

$\| A \|_{+} = 15.75 \quad (K=30)$

Implementation 2



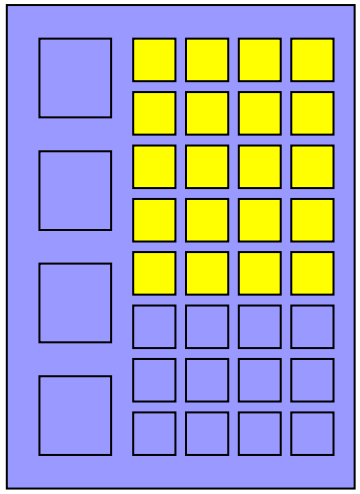
$A = (0.125, 0.5)$

$\| A \|_{\infty} = 0.5 \rightarrow 2$

$\| A \|_{1} = 0.625$

$\| A \|_{+} = 15.5$

Implementation 3



$A = (0.625, 0.0)$

$\| A \|_{\infty} = 0.625 \rightarrow 1.6$

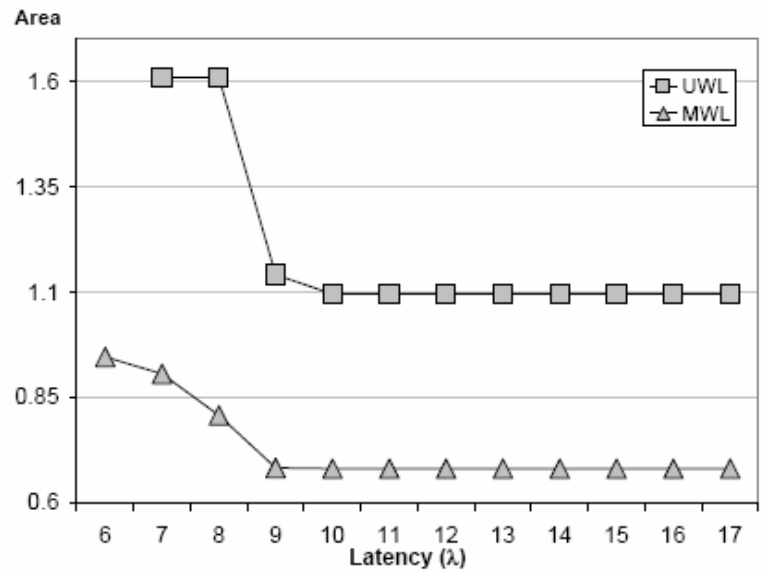
$\| A \|_{1} = 0.625$

$\| A \|_{+} = 19.375$

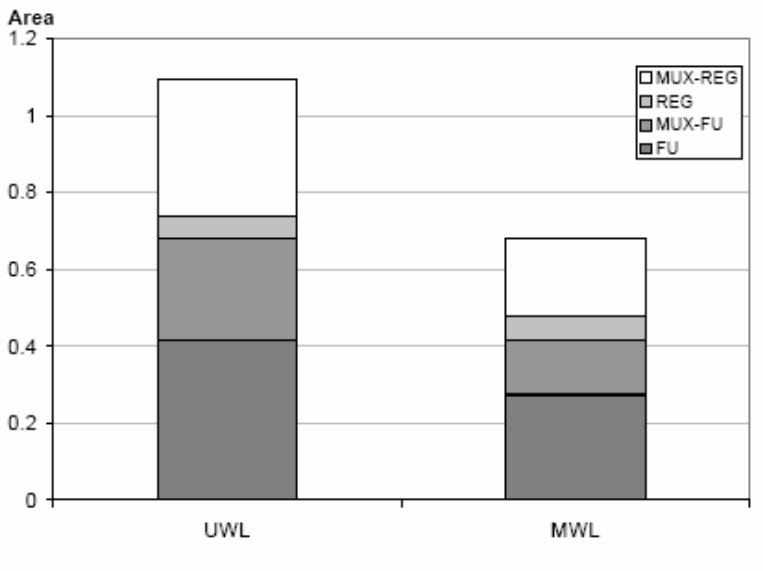
Implementation Issues

Results: Homogeneous implementations

Particular example: Bi-quad IIR,
$\sigma^2=10^{-3}$



(a) IIR_4, $\sigma^2 = 10^{-3}$



(b) IIR_4, $\sigma^2 = 10^{-3}$, $\lambda = 17$

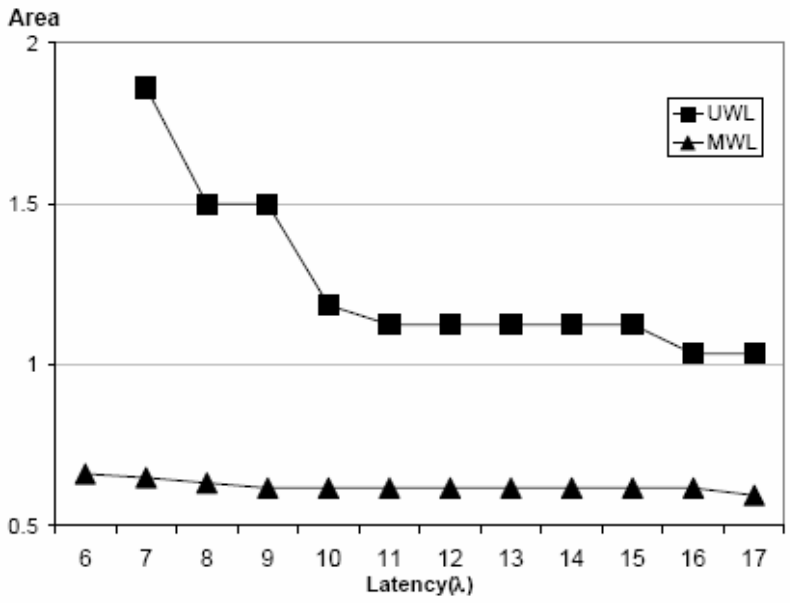
Ref. 3,6

Implementation Issues

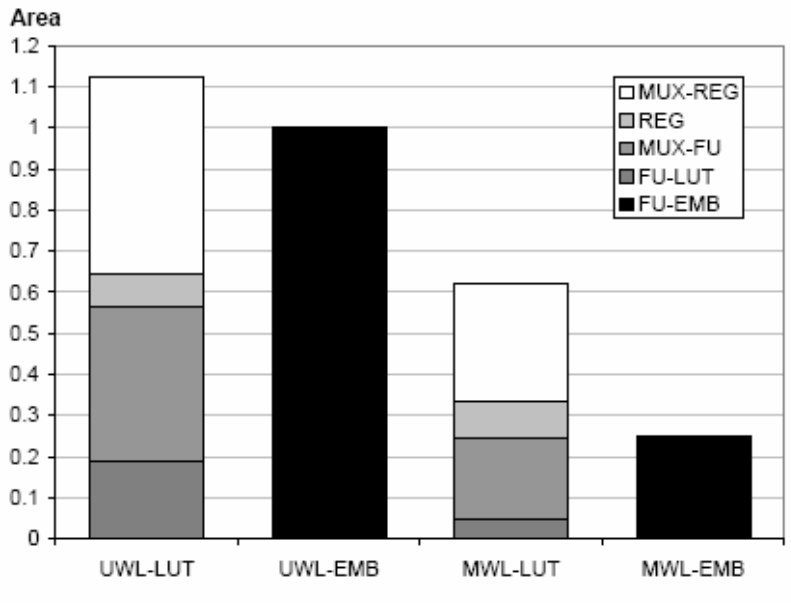
Results: Heterogeneous implementations

Particular example: Bi-quad IIR,

$\sigma^2=10^{-3}$



(a) IIR_4, $\sigma^2 = 10^{-3}$

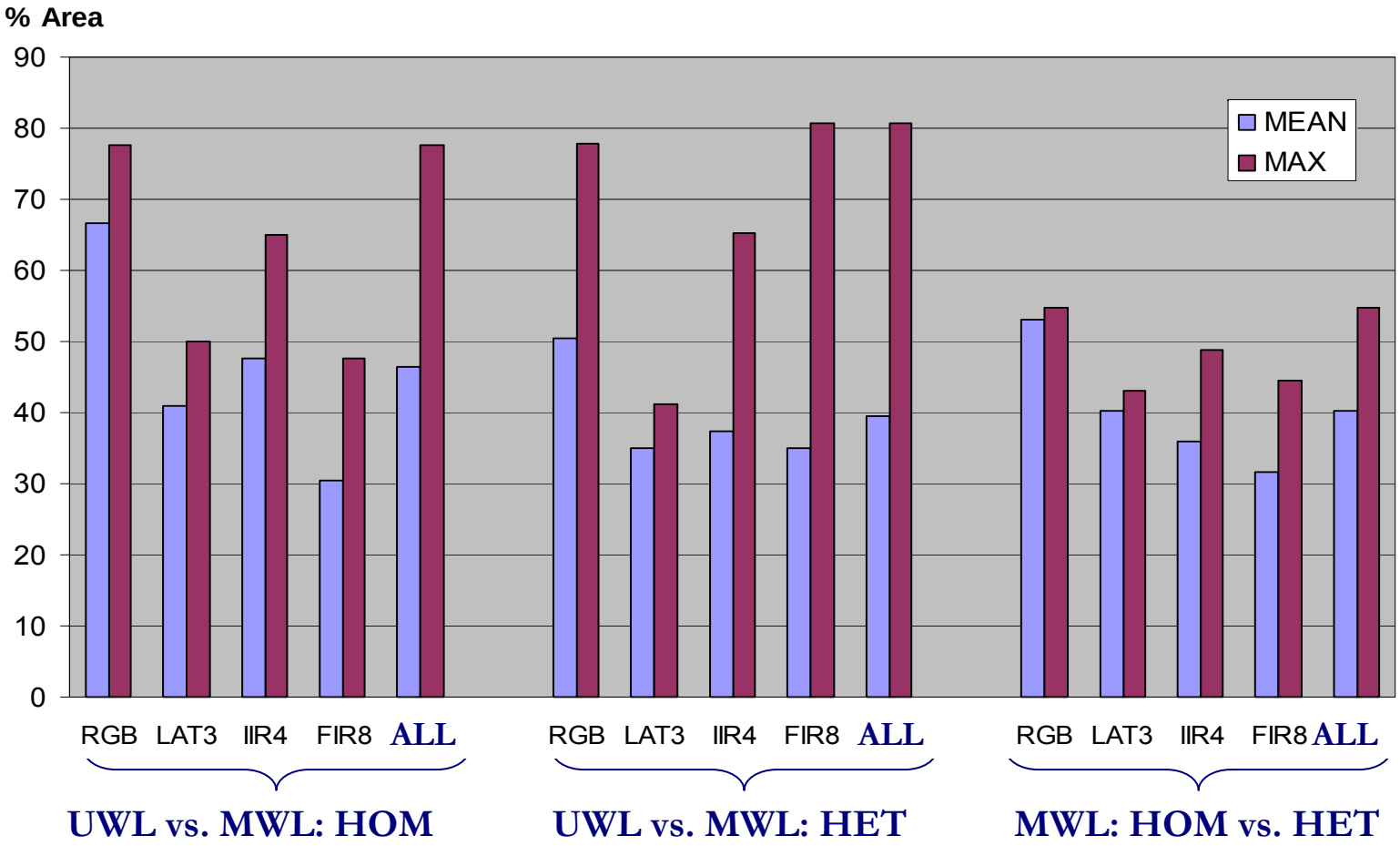


(b) IIR_4, $\sigma^2 = 10^{-3}$, $\lambda = 17$

Ref. 3,6

Implementation Issues

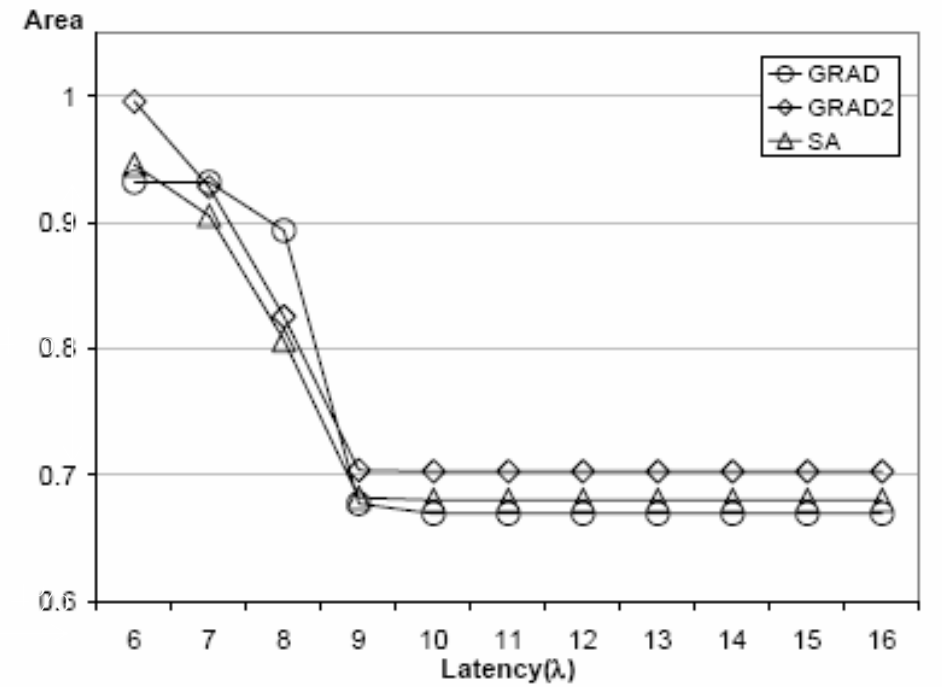
Overall Results



Ref. 3,6

Implementation Issues

Results: Effect of WL selection

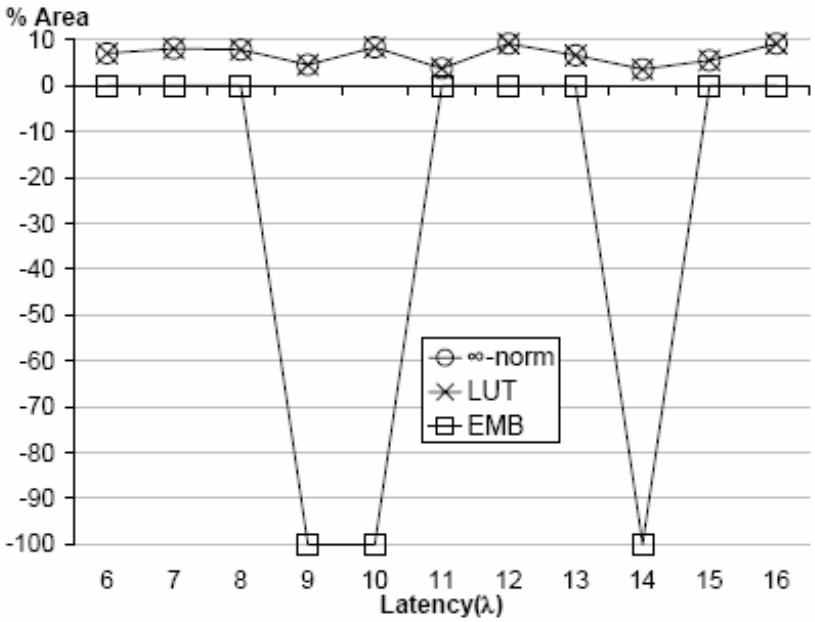


(c) IIR_4, $\sigma^2 = 10^{-3}$

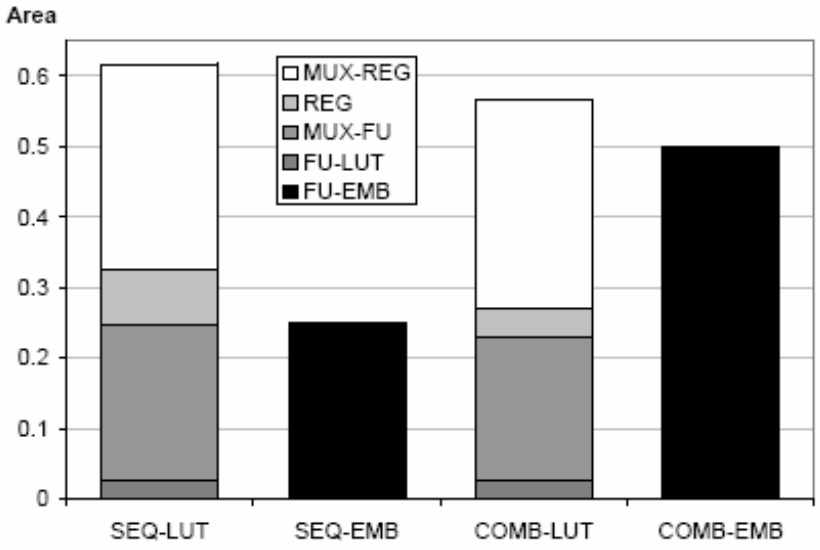
Ref. 3

Implementation Issues

Combined WL Selection and Resource Sharing



(a) IIR_4, $\sigma^2 = 10^{-5}$



(b) IIR_4, $\sigma^2 = 10^{-5}$, $\lambda = 10$

Ref. 3

Bibliography

1. G.A. Constantinides et al, “Truncation Noise in Fixed-Point SFGs. IEE Electronics Letters, 35(23):2012–2014, 1999.
2. G. A. Constantinides et al, “Wordlength Optimization for Linear Digital Signal Processing”. IEEE Trans. Computer-Aided Design, 22(10):1432–1442, October 2003
3. G. Caffarena, “Combined Word-Length Allocation and High-Level Synthesis of DSP Circuits”, PhD Thesis, UPM, 2008, http://oa.upm.es/1822/
4. J.A. Lopez et al, “Fast and Accurate Computation of the Round-Off Noise of LTI Systems”, IET Circuits, Devices and Systems, vol. 2(4), pp. 393-408, 2008.
5. D. Menard, R. Rocher, P. Scalart, and O. Sentieys. “SQNR Determination in Non-Linear and Non-Recursive Fixed-Point”, Systems. In Proc. European Signal Processing Conference EUSIPCO’04, pages 1349–1352, 2004.
6. G. Caffarena et al, “Architectural Synthesis of Fixed-Point DSP Datapaths using FPGAs”, International Journal of Reconfigurable Computing, in press, 2009.

THANKS!!!!