

Floating-point to Fixed-point Conversion for Efficient Implementation of Digital Signal Processing Programs (Short Version for FPGA DSP)

Wonyong Sung

School of Electrical Engineering
Seoul National University

Version 2003. 7. 18

What is fixed-point arithmetic?

❖ Floating-point arithmetic

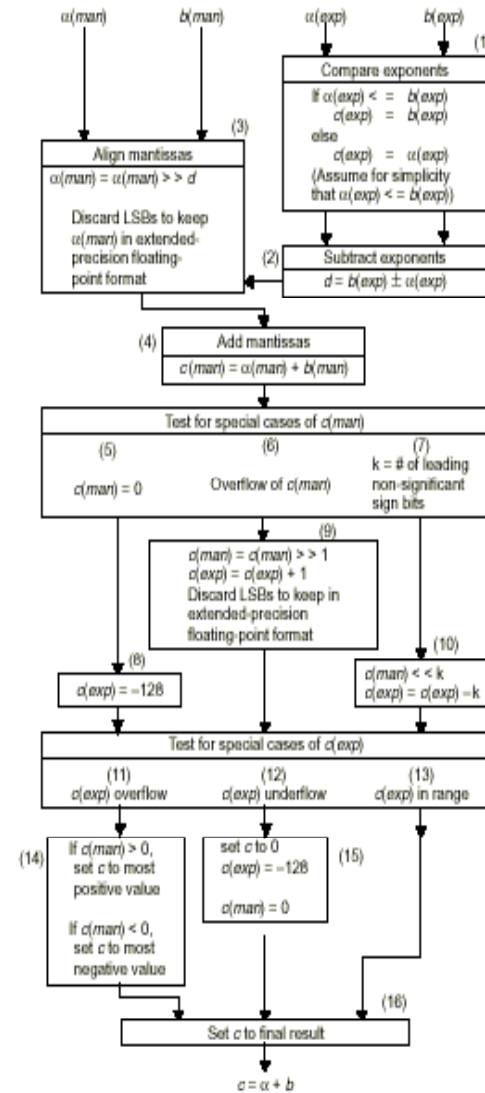
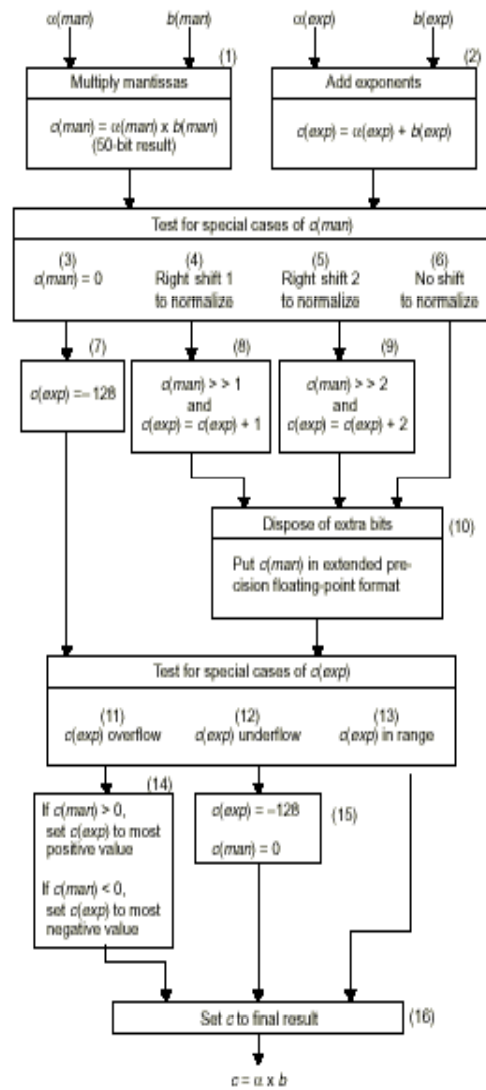
- For ex.) $0.9 * 0.55 = 0.495$, $0.9 * 5500 = 4950$
 $0.9 * 0.5555555 = 0.49999995$

❖ Fixed-point arithmetic

- For ex.) $0.9 * 0.55 = 0.495$ (seems OK)
 $0.9 * 5500 = ?$ (overflow)
 $0.9 * 0.5555555 = 0.499$ (quantization)
- Fixed-point arithmetic
 - Range is limited (scaling needed)
 - Precision limited
 - Less complex hardware

Why fixed-point implementation?

- ❖ **Fixed-point arithmetic (or integer arithmetic) requires less chip area, less delay, and less bus width (for memory).**
 - Leads to ½ to 1/10 chip area reduction
 - Leads to x2 speed increase.
- ❖ **Many embedded processors do not equip floating-point arithmetic unit**
 - A floating-point arithmetic using library requires many (at least a few 10's) cycles.
 - Leads to x10 speed increase or energy reduction.



Why is fixed-point implementation important in DSP?

❖ Digital signal processing

- Requires a large number of arithmetic operations (multiply, add, memory access)
 - 10M ~ 100M operations / sec
- Do not require exact results, in terms of SQNR 40dB ~ 100dB

❖ Embedded systems

- Many do not equip floating-point arithmetic units
- Require different word-length according to the applications
 - Audio: around 20 ~ 24 bits
 - Speech: 12 ~ 20 bits
 - Video: 8 ~ 16 bits
 - Graphics: high precision

Issues in fixed-point optimization

❖ Performance estimation

- By analytical methods (theory)
 - Usually limited to linear systems
- By simulation
 - Requires simulation using a large number of input data (fast simulation required)
 - Easy simulation program generation required

❖ Automatic scaling

- Scales the input and output data ($\times 2^{-n}$)
- $0.9 * 5500 = ? > 0.9 * 0.55 = 0.495 (*10^{+4})$

❖ Word-length optimization

- Smaller word-length degrades the system performance
- Larger word-length requires more chip area and power consumption.

Overview of this talk

1. Fixed-point arithmetic and system design
2. Fixed-point simulation method
3. Autoscaler (floating-point to integer)
4. Fixed-point optimizer (wordlength opt.)

Fixed-point Arithmetic and System Design

School of Electrical Engineering
Seoul National University

1. Number Representation

❖ Floating-point format

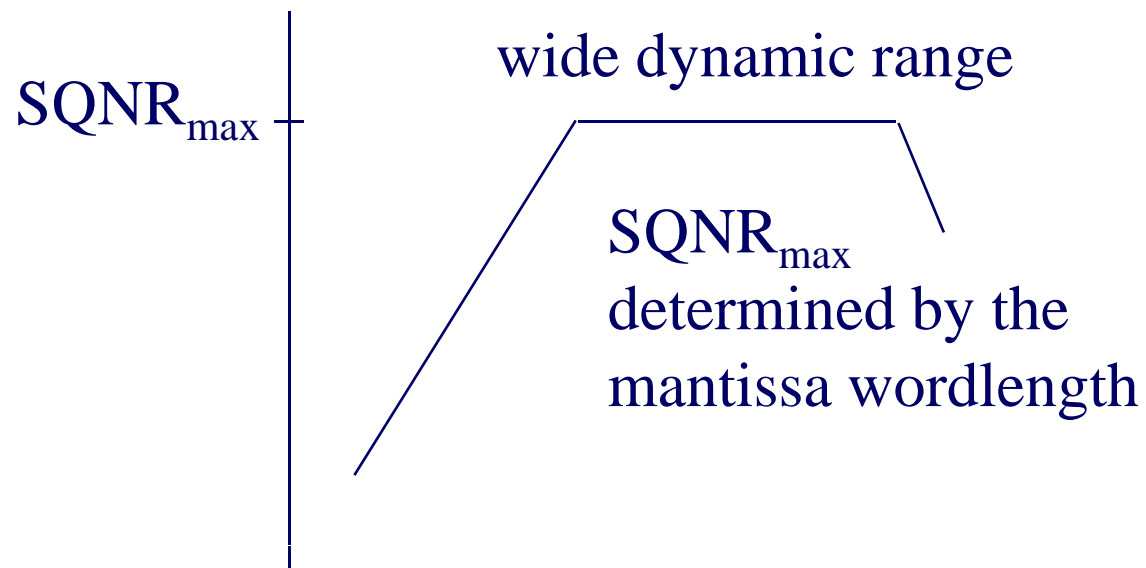
- $x = M(x) 2^{E(x)}$
- wide dynamic range <- no explicit scaling needed
- good for algorithm develop, higher hardware cost

❖ Fixed-point format

- only $M(x)$ is used with constant $E(x)$
- minimizing the wordlength is important for economic hardware implementation

Floating-point format

❖ SQNR



- ❖ **32bit IEEE standard format is most widely used (24bit mantissa, 8bit exponent)**

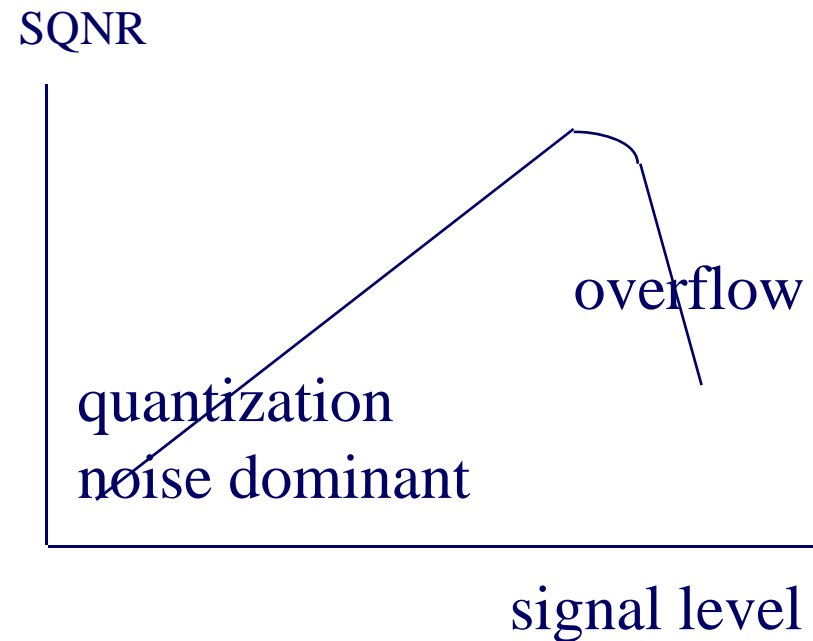
Fixed-point format

❖ SQNR

- Scaling is needed

❖ Specification

- Negative number representation
- Overflow handling
- Word-length reduction
- Conversion to or from real value



Negative number representation

❖ Unsigned

0000: 0, 0001:1, 0111:7, 1111:15

❖ Signed

- Two's complement ($-x = \bar{x} + 1$)

0000:0, 0001:1, 0111:7

1111: -1, 1000: -8

- One's complement ($-x = \bar{x}$)

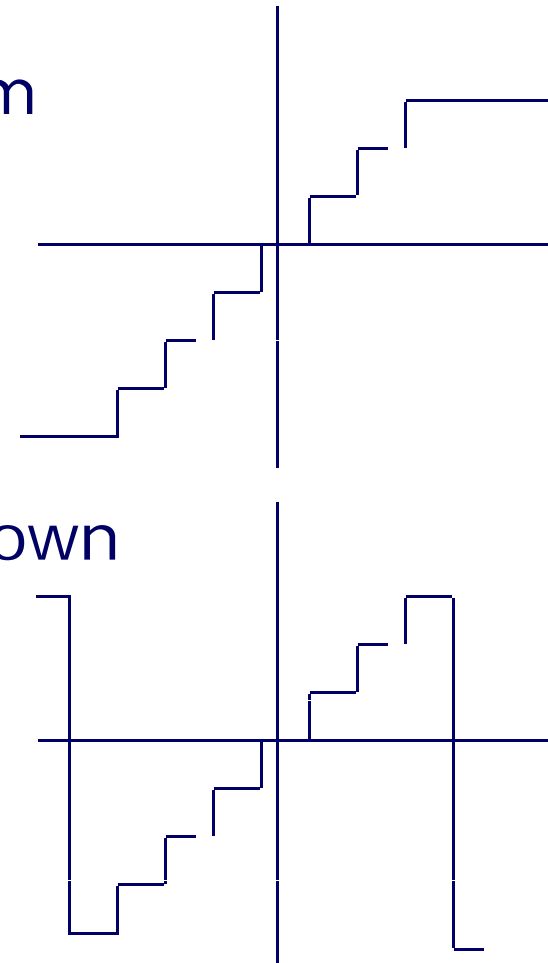
Overflow handling

❖ Saturation

- no sign loss for overflow
- magnitude becomes maximum
- preferred in signal processing

❖ Overflow

- simple implementation
- good when signal range is known



Quantization (Wordlength reduction)

❖ Methods

- Rounding
- Truncation
- Random
- ...

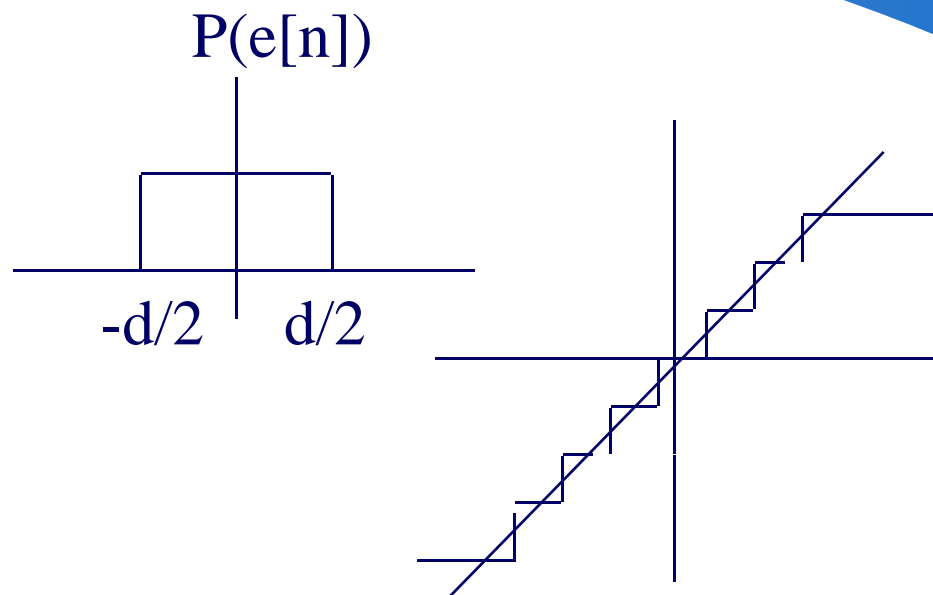
❖ Source of Quantization

- Signal quantization
 - Generate rounding errors that can be modeled by random noise.
- Coefficient quantization
 - Deterministic one, affects the frequency response for the case of filters – this is not random noise.

Quantization methods

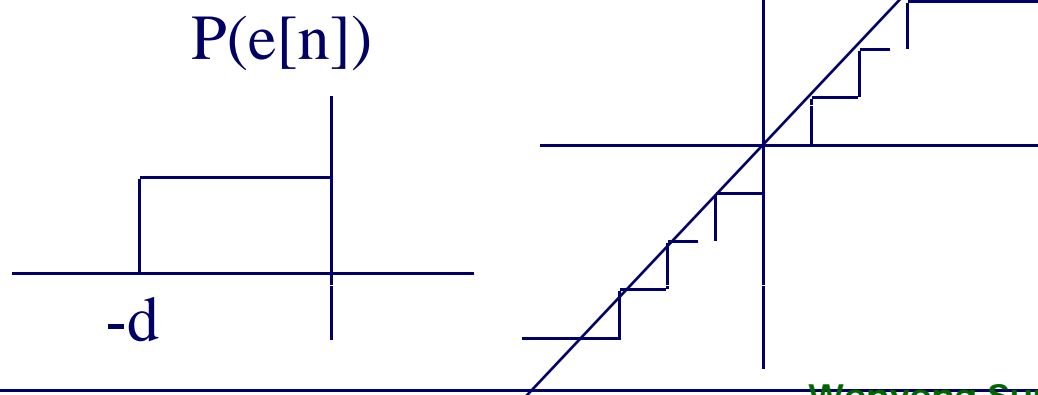
❖ Rounding

- max. quant. error is $d/2$



❖ Truncation

- max. quant. error is d ,
- D.C. bias (fatal when accumulated)



2. Fixed-point Data Format

❖ Integer format

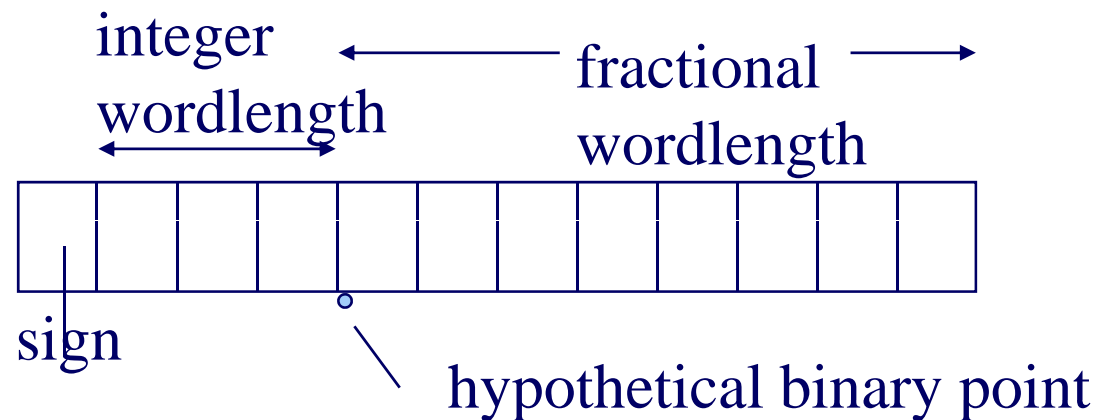
- all data is interpreted as an integer
- the quantization level is large (1)

❖ Fractional format

- all data is between -1 to 1
- the quantization level is $2^{-(B-1)}$

❖ Generalized fixed-point format

- allow both integer and fractional wordlengths
- integer wordlength determines the maximum signal range, and the fractional wordlength determines the quantization level.



Generalized fixed-point format

❖ SPW format

- $\langle \text{wordlength, integer wordlength, sign} \rangle \langle 12, 3, t \rangle$
- signal range: $-2^3 \sim +2^3$, quantization level: 2^{-8}

❖ Silage (DFL) format

- fix $\langle \text{wordlength, fractional wld} \rangle$; always two's compl.
- fix $\langle 12, 8 \rangle$

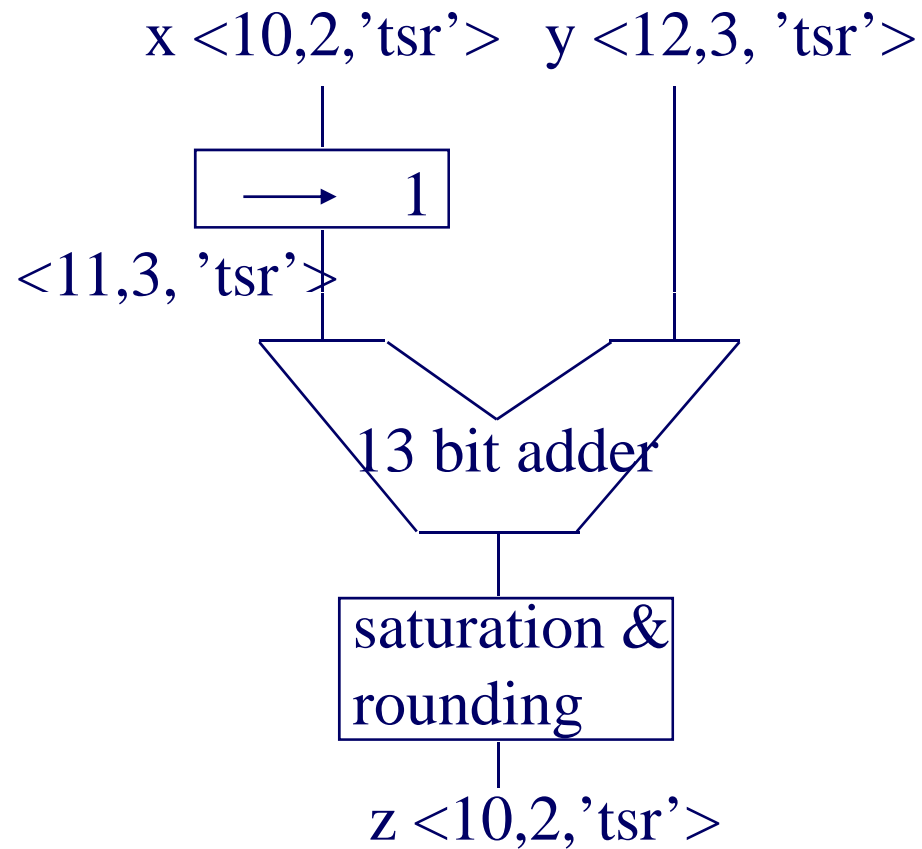
❖ SNU Fixed-point Simulator format

- $\langle \text{wordlength, integer wordlength, sign_overflow_quantization mode} \rangle$
- $\langle 12, 3, \text{'tsr'} \rangle$

Generalized fixed-point format

- ❖ **Arithmetic shift left by n bit**
 - decreases the IWL by n (shift is used for scaling, not for cost effective multiplication)
 - * this is not the case that shift is used instead of multiplication with 2^n
- ❖ **Arithmetic shift right by n bit**
 - increases the IWL by n
- ❖ **Addition**
 - the IWL of both operands should be the same
- ❖ **Multiplication ($z = x * y$)**
 - $IWL(z) = IWL(x) + IWL(y) - 1$; in 2's compl.

Generalized fixed-point format



3. Scaling Method

❖ Purpose of Scaling

- prevent overflows or waste of bits by proper shifting (arithmetic shift)

❖ Scaling implementation

- overflow prevention: shift right (IWL increase)
- save extra-bits: shift left (IWL decrease)

❖ Minimum integer wordlength

- $IWL_{\min}(x) = \lceil \log_2 |R(x)| \rceil$
- $R(x)$ is the range of a signal

Range estimation methods

❖ Analytical method

- L1 norm based is most widely used
- L1 norm based: very conservative estimate
- applicable to linear or simple systems

❖ Simulation based

- requires computing power
- optimum estimate, but input signal dependent
- applicable to non-linear and time-varying systems

L1 norm based scaling



$$|y[n]| \leq x_{\max} \sum |h[n]|$$

This means that the output can be higher than the input level by L1 norm times

L1 norm computation

- by using analytical method
- by computing the unit pulse response, and then summing-up the absolute value of the response

L2 norm based scaling

$$\text{L2 norm} = \sum |h[n]|^2$$

- L2 norm is an upperbound for the signal energy
- it is less pessimistic than the L1 norm
- it is optimum estimation when the input is a white random input

L_{∞} norm based scaling

❖ $L_{\infty} = \text{Max } |H(j\omega)|$

- is an upper bound when the input is sinusoid
- good when the input signal is very highly correlated
- least pessimistic norm

Simulation based range estimation

- collect information of sum, squared_sum, absolute_max, and the number of update during the floating-point simulation
- derive mean and standard deviation for a signal after the simulation
- $R(x) = \max\{|m(x)| + n \sigma(x), AMax\}$, where n is between 4 to 16

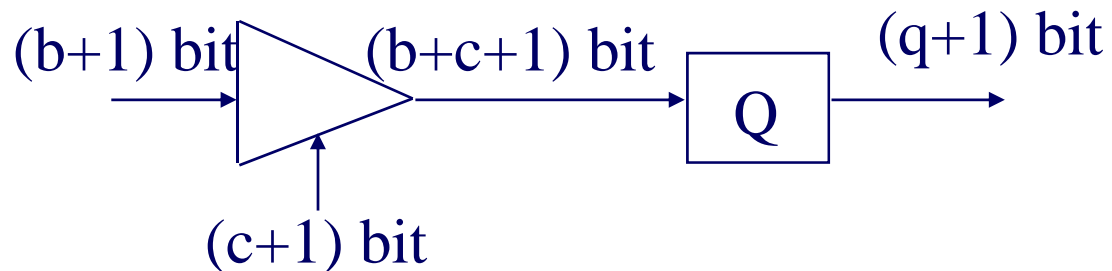
Simulation based range estimation

- ❖ **Comparison for a 2nd order IIR filter for speech application (WL = 16 bit)**
 - IWL determined by the simulation based method is 4 bit smaller than that of the L1 norm based method
 - SQNR difference of 24 dB
- ❖ **ADPCM implementation**
 - needs 4 different speech files for obtaining a reliable range data

4. Wordlength Determination

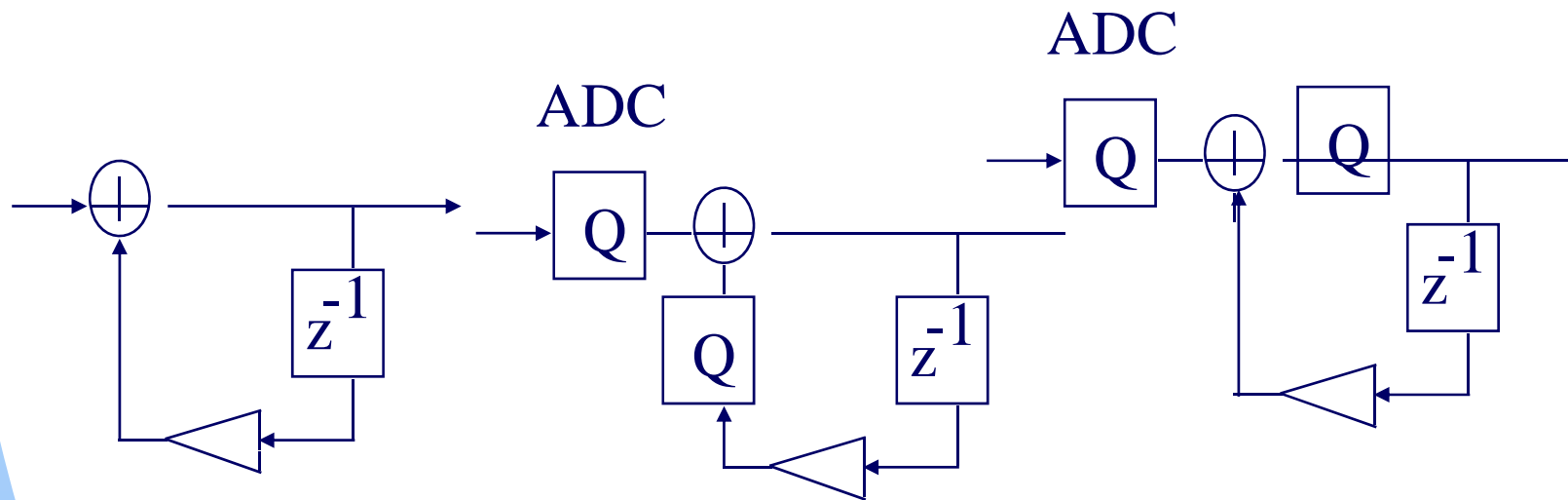
❖ Wordlength reduction

- reduce the wordlength, at an ADC or multiplier output, to minimize the hardware cost while keeping the signal accuracy acceptable

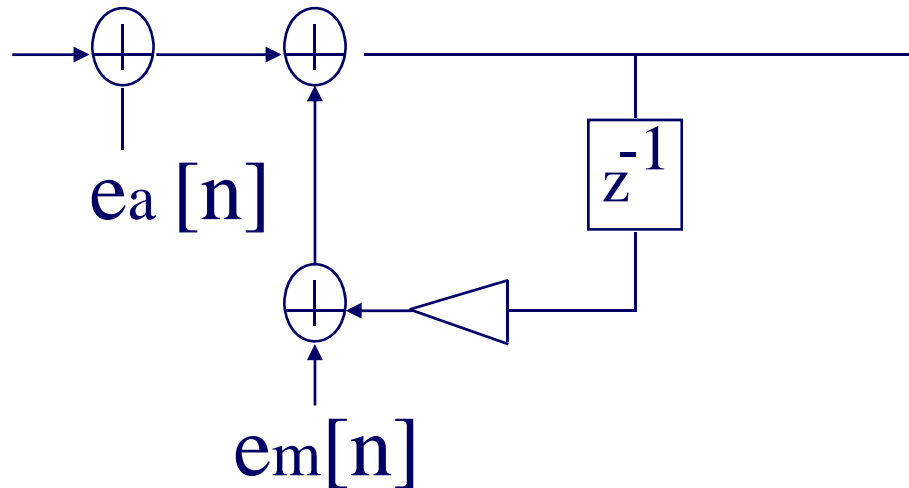


Modeling of quantization

- ❖ Ideal system - for floating-point simulation
- ❖ Non-linear model
- for fixed-point simulation
- ❖ Linearized model - for analytical method



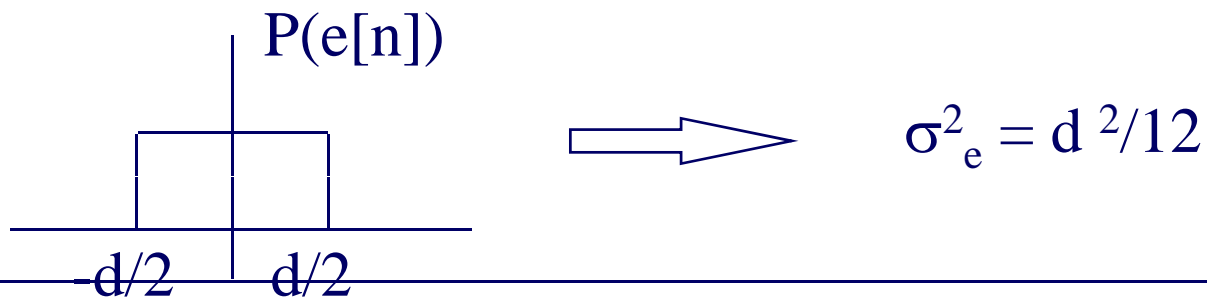
Modeling of quantization - continue



- ❖ the magnitude of $e_a[n]$ and $e_m[n]$ is dependent on the fractional word-length

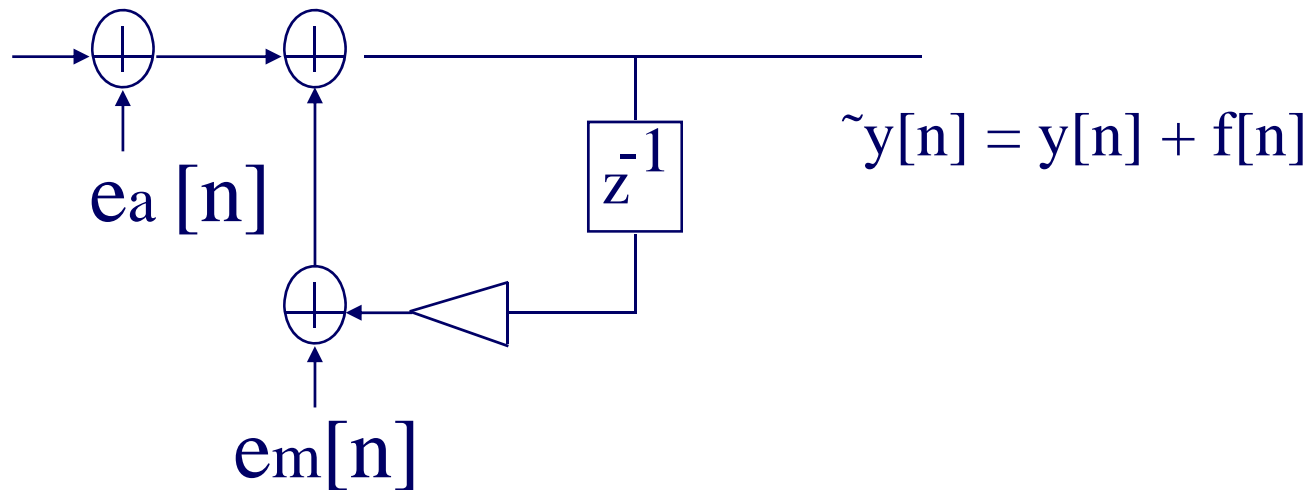
Linearized modeling of signal quantization

- ❖ Add statistically equivalent quantization error signal instead of the quantizer
- ❖ Quantization noise
 - wide-sense stationary white noise
 - uniform distribution between $-d/2$ to $d/2$
 - different noise sources are uncorrelated



Modeling of signal quantization

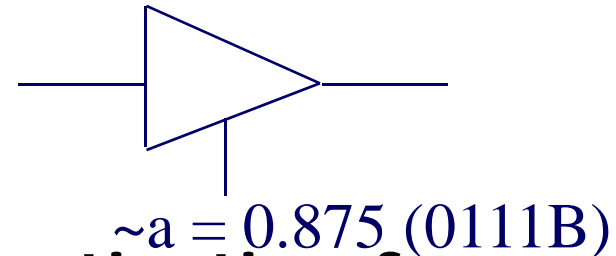
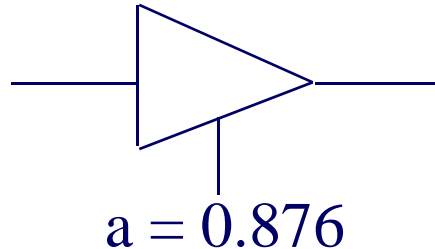
❖ Computation of output noise power



$$\begin{aligned}\sigma_f^2 &= (\sigma_a^2 + \sigma_m^2) (1/2\pi) \int |H(e^{j\omega})|^2 d\omega \\ &= (\sigma_a^2 + \sigma_m^2) \sum |h[n]|^2\end{aligned}$$

Coefficient quantization

- ❖ Word-length reduction in filter coefficients or (usually) constant system parameters



- ❖ Effects of coefficients quantization for FIR and IIR digital filters
 - deterministic
 - easily known by obtaining the frequency response from the quantized coefficients
 - optimization program for minimum hardware cost

5. Fixed-point Performance

- ❖ **Effects of quantization noise in digital signal processing systems**
 - Linear digital filters:
 - additive quantization noise at the output
 - Constant filter coefficients:
 - transfer function is modified
 - Adaptive digital filter:
 - quantization noise changes the adaptation performance or the mean squared error (mse) after the convergence <- distortion

❖ Performance measure

- Linear digital filters
 - SQNR
- Adaptive digital filters
 - mean squared error after some time-off period
- Speech coder (waveform coder)
 - the SQNR between the original speech and the reconstructed speech after compensating the filtering or time-delay (but this does not apply to LPC or CELP, model based coder)
- Communication system(Tx-Rx system)
 - the bit-error rate is the best measure, but requires much simulation time
 - the peak error at the eye diagram can be a measure

Overview of this talk

1. Fixed-point arithmetic and system design
2. Fixed-point simulation method
3. Autoscaler (floating-point to integer)
4. Fixed-point optimizer (wordlength opt.)

Word-Length Optimization for Digital Signal Processing Algorithms

W. Sung and K. Kum, "Simulation-based Word-length Optimization Method for Fixed-point Digital Signal Processing Systems," *IEEE Trans. Signal Processing*,

Wordlength optimization

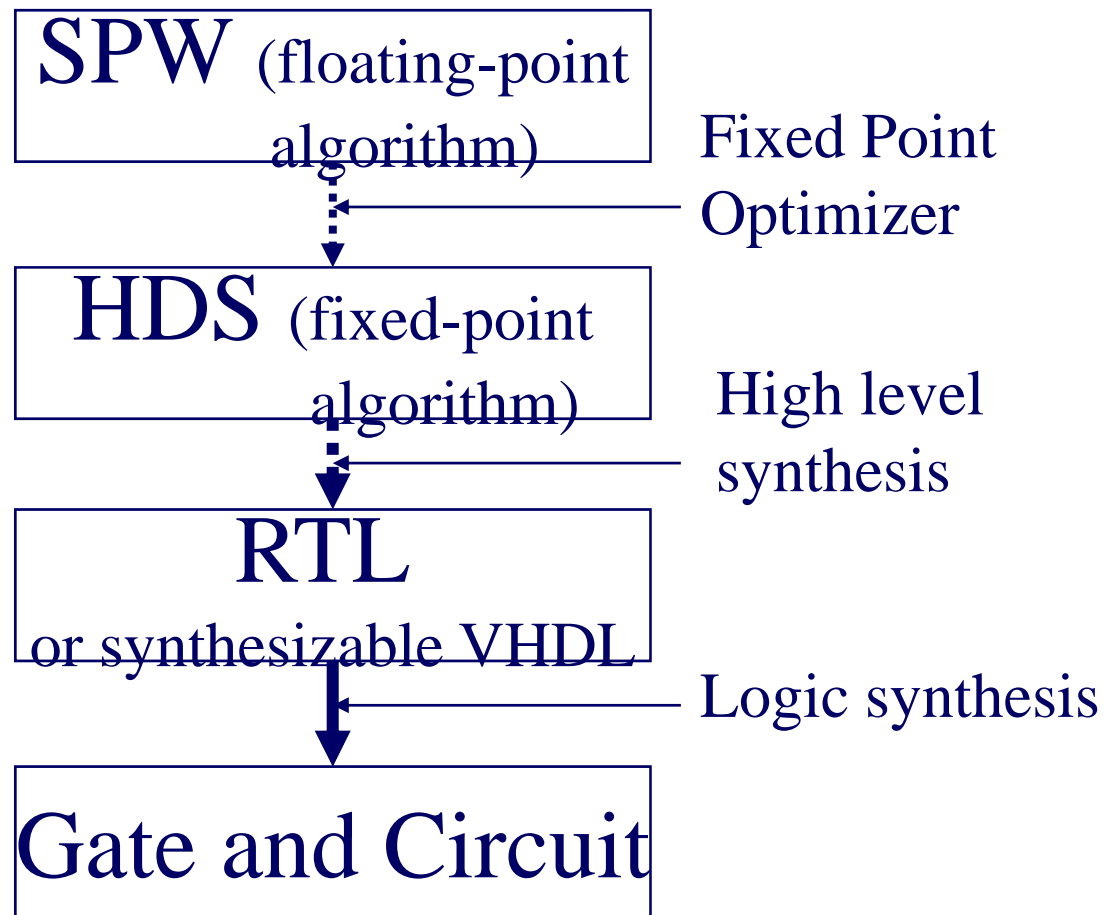
❖ IWL determination

- It can be done relatively easily by range estimation

❖ FWL (or WL) determination

- Should know the effects of quantization noise at each quantizer (usually inserted at the output of a multiplier or a summing adder)
- In the simulation based method, it requires many iterative simulations

Overall flow – SPW



Fixed Point Optimizer Flow

- ❖ **Signal flow graph**
- ❖ **Define the performance measure and insert the measurement block**
- ❖ **Insert quantizer if needed**
- ❖ **Grouping of signal flow graph**
- ❖ **Determine the integer wordlength by range estimation**
- ❖ **Determine the fractional wordlength**
 - Assign one fractional wordlength for each group
 - Determination of the minimum wordlength for each group
 - Hardware cost model
 - Exhaustive search or heuristic search

Goals of fixed-point optimization

- ❖ **Start with signal flow graph in SPW or widely used GUI**
- ❖ **Define the performance**
 - SQNR, bit error rate, convergence time, ...
- ❖ **Minimize the hardware cost**
 - Should know the HW cost for each arithmetic and storage block, time-multiplexing ratio
- ❖ **Conduct it at a short time, if possible**

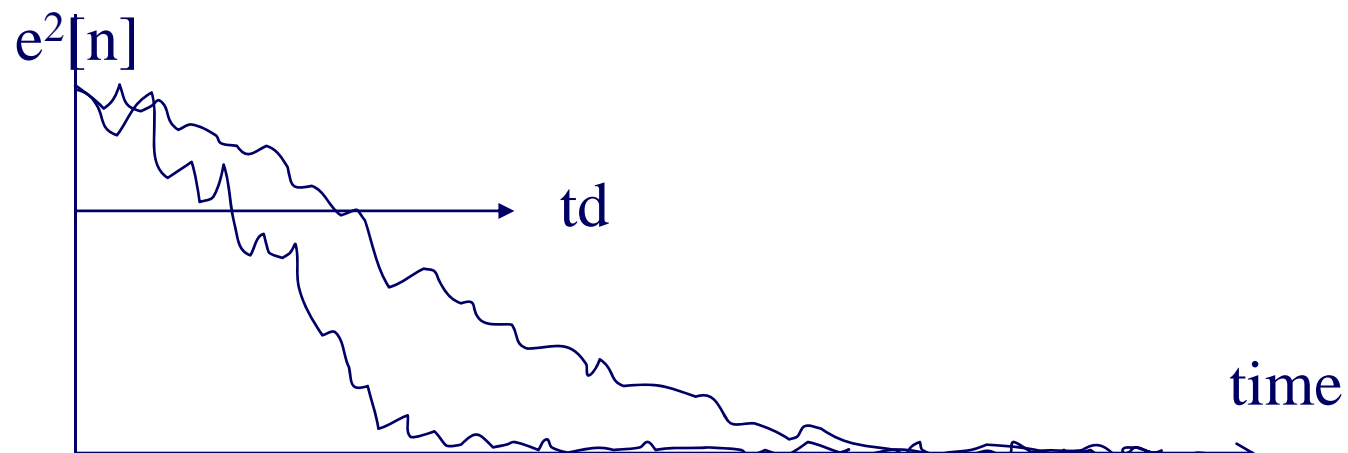
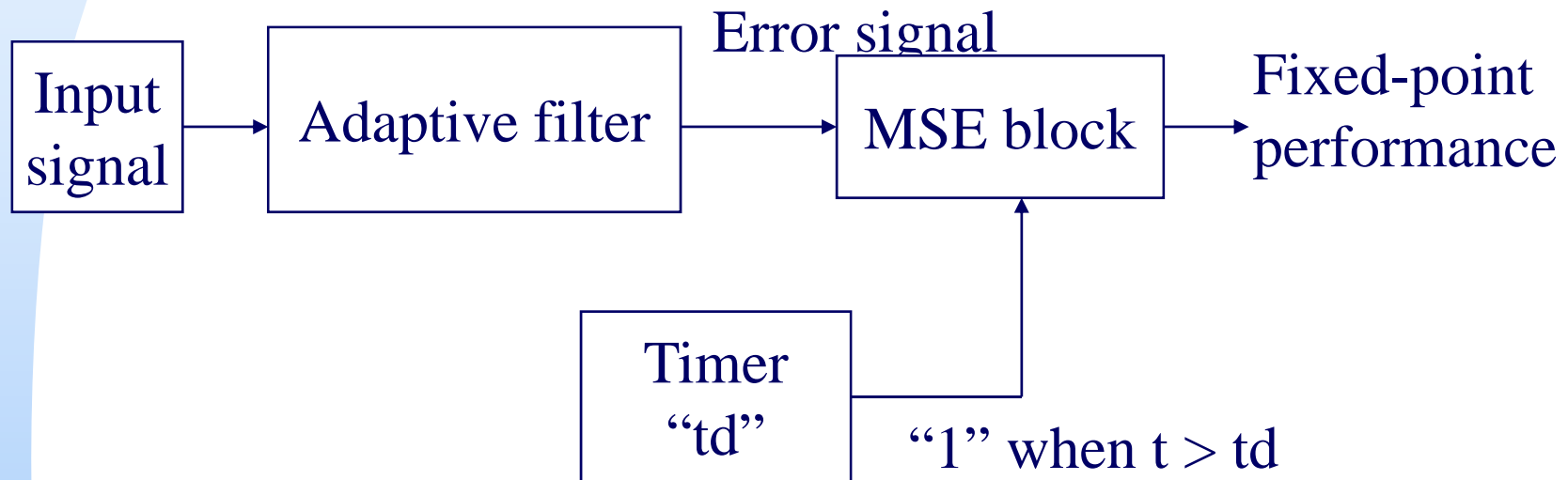
Performance measures

❖ SQNR (Signal to Quantization Noise Ratio):

- widely used for linear systems or waveform coders (ADM, ADPCM)
- but inadequate for non-linear and perceptual quality based coders

❖ System specific performance measures are needed

- Adaptive filters: smaller wordlength for coefficients leads to slow adaptation. -
> MSE after some fixed delay (adaptation time)
- Receivers: MSE of the decision point (this determines the bit error rate)

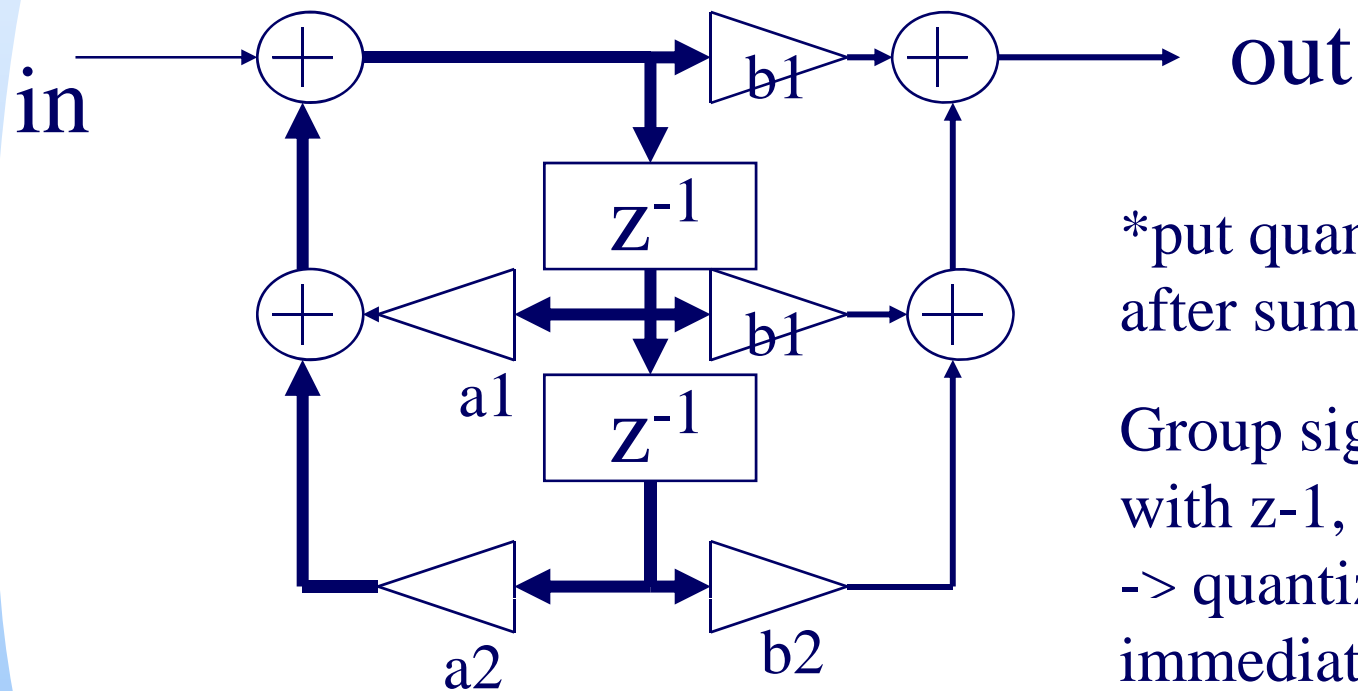


Fast optimization

- ❖ **Range estimation: conducts in one simulation for each input vector**
- ❖ **Fixed-point optimization: requires simulations for all different wordlengths for each signal – many many combinations**
 - Reduce the number of signals that can have different word-lengths
 - Using signal flow graph (Signal Grouping)
 - Using HW binding information (needs to be combined with high-level synthesis)
 - Find the upper and lower bound of the wordlengths for search

Signal grouping

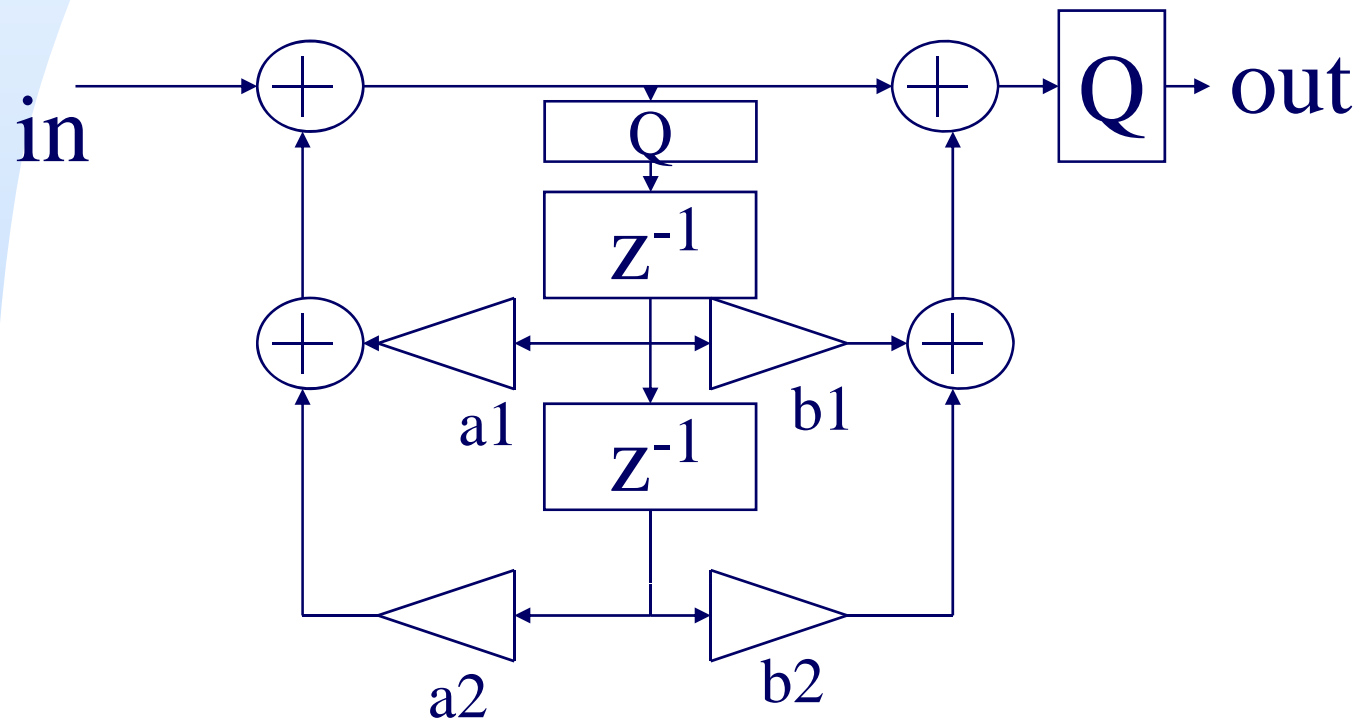
- ❖ Needed for reducing the number of signals having different wordlengths. This reduces the optimization time.
- ❖ Assigns the same wordlength to a block connected by
 - Delay (z^{-1})
 - Adders
 - Mux
- ❖ Give different wordlengths to blocks connected by
 - Quantizers
 - Multipliers
- ❖ May put some quantizers to separate signals because too small number of groups will lead to higher quantization noise



*put quantizer
after summing tree

Group signals connected
with z^{-1} , adders
-> quantize signal
immediately after mult.

$$W_{mult} = W_{adder}$$

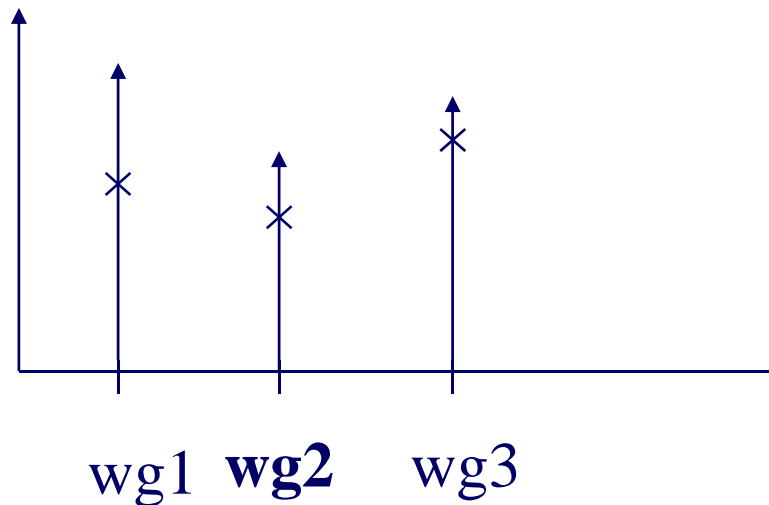


Assigning the same wordlength to all mult inputs

Assigning the same wordlength to all adders

$W_{adder} > W_{mult}$

Search range reduction – minimum wordlength for each group



Minimum wordlength for each group: there is a lower bound of signal wordlength for a group to satisfy the given performance.

This is obtained by assigning full precision (double floating-p) to other wordlengths and only reduces the wordlength of that group until the performance is just satisfied.

$$p((w_1, w_2, \dots, w_i, \dots, w_N)) \geq p((w_1, w_2, \dots, w_i-1, \dots, w_N))$$

Minimum HW cost wordlength search

- ❖ After obtaining the minimum wordlength vector, the final wordlength vector requiring the minimum hardware cost is found by search
- ❖ HW cost model: # of gates for the implementation of arithmetic and memory units
 - Should consider time-sharing ratio

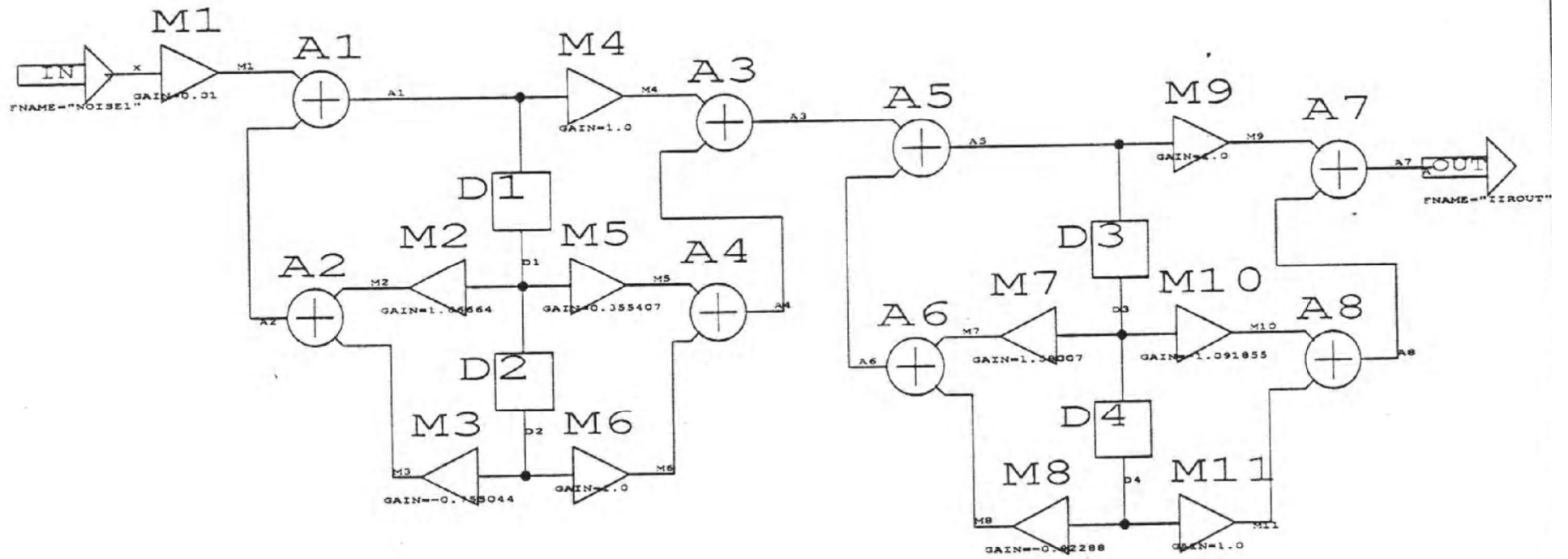
Search methods

❖ Exhaustive search algorithms

- Sort all the possible wordlength vector (starting from the minimum wordlength vector) by the HW cost
- Try from the smallest HW cost vector to the next, ... until the performance is satisfied.
- The number of searches can be very excessive.

❖ Heuristic search

- For the given minimum wordlength vector, if the simulation results do not meet the specification, all the wordlengths are increased by one bit, if not two bits, and then the wordlength of a signal whose cost is the most expensive is reduced one by one.
- The number of search: usually one or two + the number of groups $\sim = N_{\text{group}}$



4차 IIR 필터
 4th order IIR filter

Signal grouping 신호의 그룹핑

그룹	신호
1	X
2	M1, A1, A2, M2, M3, D1, D2
3	M4, M5, M6, A3, A4, A5, A6, M7, M8, D3, D4
4	M9, M10, M11, A7, A8

Range of each group 그룹의 범위

그룹	범위
1	2.372
2	0.192
3	1.781
4	1.128

신호들의 범위 Range

signal	mean	variance	range
A1	4.558032e-03	7.616390e-04	0.115
A2	4.172500e-03	7.310290e-04	0.112
A3	1.068781e-02	3.904840e-03	0.261
A4	6.129794e-03	1.371268e-03	0.154
A5	3.156875e-02	7.517205e-02	1.128
A6	2.088093e-02	4.977613e-02	0.915
A7	2.823217e-02	4.344898e-02	0.861
A8	-3.336580e-03	1.358935e-02	0.465
D1	4.548226e-03	7.616802e-04	0.115
D2	4.513321e-03	7.613873e-04	0.111
D3	3.109902e-02	7.509117e-02	1.127
D4	3.061905e-02	7.500560e-02	1.121
M1	3.855313e-04	3.404527e-05	0.02
M10	-3.395561e-02	8.951978e-02	1.16
M11	3.061905e-02	7.500560e-02	1.12
M2	7.580257e-03	2.115712e-03	0.19
M3	-3.407754e-03	4.340605e-04	0.08
M4	4.558032e-03	7.616390e-04	0.11
M5	1.616471e-03	9.621104e-05	0.04
M6	4.513321e-03	7.613873e-04	0.11
M7	4.913862e-02	1.874743e-01	1.78
M8	-2.825770e-02	6.388284e-02	0.98
M9	3.156875e-02	7.517205e-02	1.12
X	3.855312e-02	3.404529e-01	2.37

Hw cost and sqnr according to wordlength vector

Minimum wordlength for each group and signal to quant. noise

그룹별 최소 단어길이 및 그때의 신호대 잡음비

그룹	단어길이	신호대 잡음비
1	9	42.37
2	13	45.21
3	12	40.27
4	10	40.59

단어길이에 따른 하드웨어 비용 및 신호대 잡음비

wld w	cost $c(w)$	sqnr $s(w), \text{dB}$
(9, 13, 12, 10)	8230	36.42
(9, 13, 12, 11)	8270	37.75
(10, 13, 12, 10)	8280	36.56
(9, 13, 12, 12)	8310	38.81
(10, 13, 12, 11)	8320	38.44
(11, 13, 12, 10)	8330	36.75
...		
(9, 13, 13, 12)	8626	38.37
(10, 13, 13, 11)	8636	40.14
...		

Added Features in Fixed Point Optimizer

- ❖ **Support of both automatic and manual grouping functions**
 - Manual grouping functions may use the hardware sharing information
- ❖ **Support of predetermined wordlength**
 - Scaling only
- ❖ **Architecture driven wordlength optimization**
 - Uniform wordlength optimization for bit serial implementations