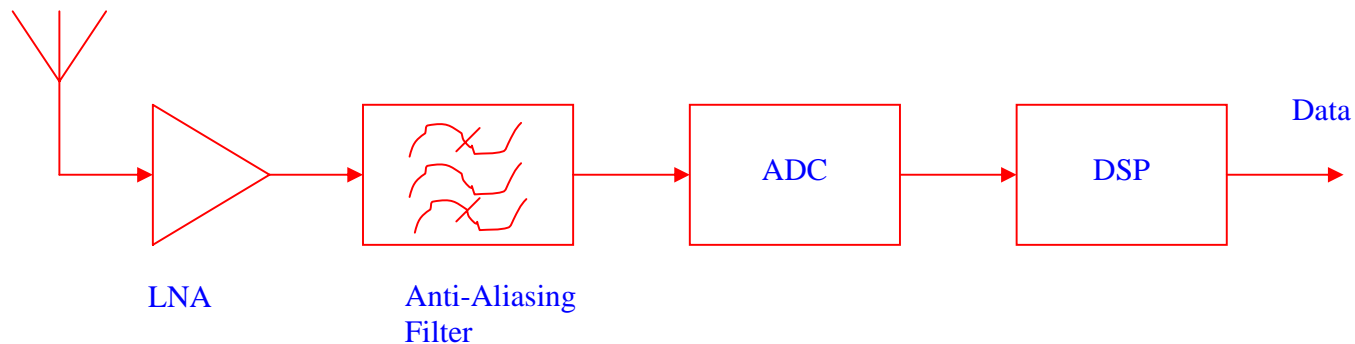


Low Power Reconfigurable Digital Filter Banks for Software Defined Radio Handset

Dr. Vinod A Prasad
Assistant Professor
School of Computer Engineering
Nanyang Technological University
Singapore

Software Radio

- ❖ Software radio is an emerging technology, thought to build flexible radio systems, multiservice, multistandard, multiband, reconfigurable and reprogrammable by software.
- ❖ Expansion of Digital Signal Processing towards the Antenna
- ❖ Single hardware platform which can be reconfigured for all existing and upcoming standards



Issues in Realizing Ideal SDR

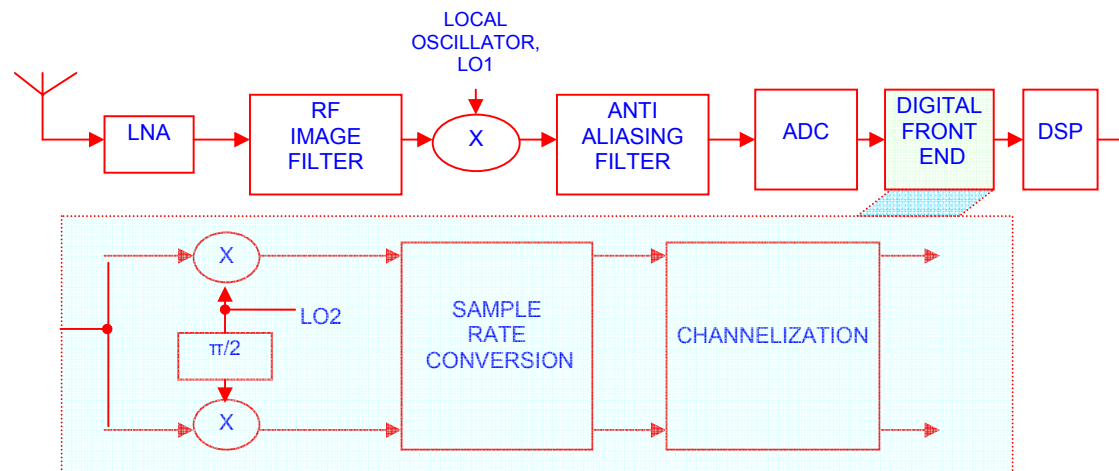
❖ ADC Limitations

- ❖ Speed Constraints - Nyquist Criterion
- ❖ High Dynamic Range of Wireless Signals
- ❖ State of the Art ADC's
 - ❖ 105 Mega samples per sec (MSPS) 14 bit
 - ❖ Still unable to reach the desired level

❖ So we need

- ❖ Analog Processing
- ❖ Intermediate Frequency (IF) Digital Signal Processing

Practical Software Defined Radio Receiver



Current Scenario

- ❖ SDR is employed in base stations only.
- ❖ No constraints on area and power.
- ❖ Reconfigurability is constrained to switching the operation among distinct receivers based on the current mode.

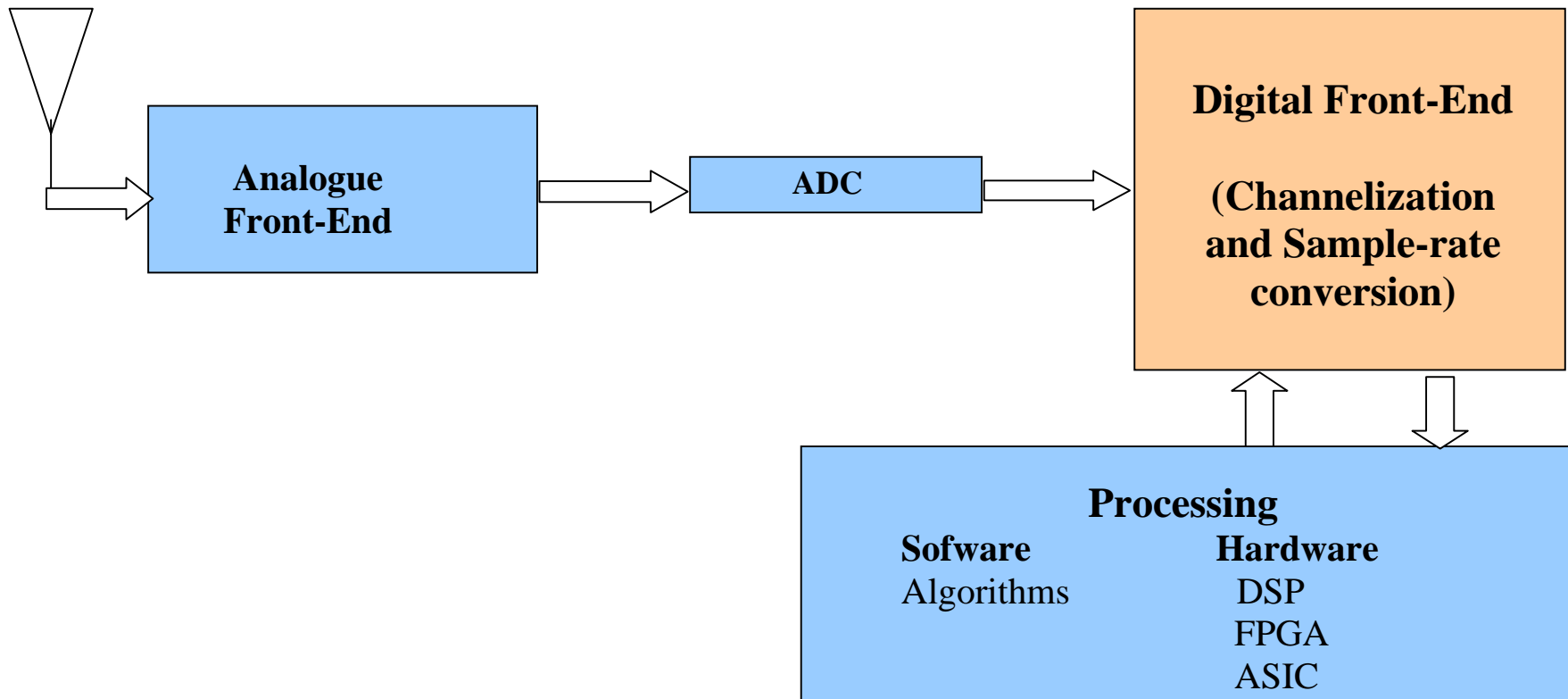
Ultimate Aim of SDR

- ❖ SDR is to be migrated to mobile handsets where its true potential can be realized.
- ❖ Tight constraints on area and power.
- ❖ Need of the same hardware to be reconfigured to operate for a new mode (standard) instead of switching among distinct receivers.

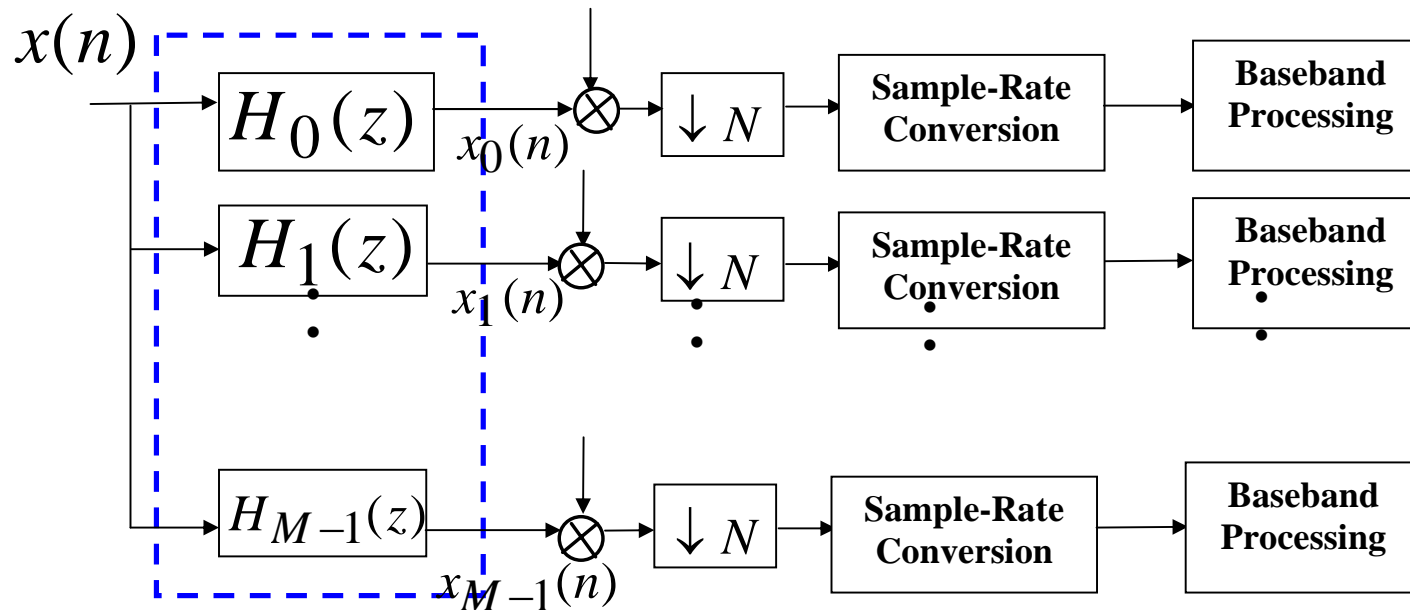
Objectives of our SDR project

- ❖ Design a reconfigurable low complexity SDR receiver to meet the requirements of reconfigurability, high speed and low power of handsets.
- ❖ Design the reconfigurable low complexity digital front end for the SDR and implement in FPGA and ASIC methodology (hybrid).
- ❖ Our current work focuses on:
 1. Low complexity implementation of channel filters and filterbanks (channelizer).
 2. Incorporation of reconfigurability in to these low complexity architectures.

Digital Front End – Our Area of Research



- The most computationally intensive task in the digital front-end is **channelization**.
- Channelization involves the extraction of individual radio channels by bandpass digital filters known as **channel filters**.

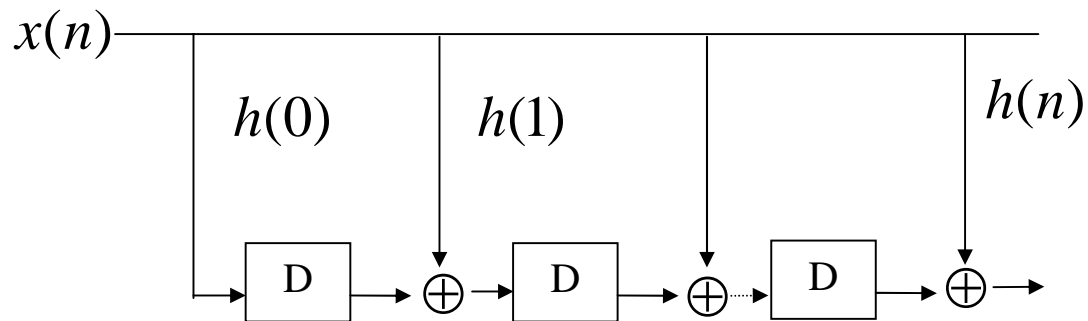


Filter bank channelizer of an SDR

- Most computationally intensive function in Channelizer is **Digital filtering**.
- Accomplished by **FIR filters** (a bank of FIR filters, called channel filters).
- Higher order filters are necessary to meet the stringent adjacent channel attenuation specifications.
- Channel filters need to be implemented with minimum area, high-speed and low power consumption.
- The three major components of a digital filter are **delay, adder, and multiplier** – out of which **multiplier** accounts for the most hardware complexity.
- The number of **adders** (subtractors) needed to realize the coefficient multipliers determines the filter implementation complexity.
- Research on Low Complexity filter realization focus on reducing the number of adders needed to implement the coefficient multipliers.

Filter Multiplier Complexity Reduction Techniques

- A well-known technique for reducing adder requirement is the **Multiple Constant Multiplications (MCM)**.
- **MCM**: Multiplication of one variable (input signal) with multiple constants (filter coefficients).
- Minimizing adders using **Common Subexpression Elimination (CSE)**.

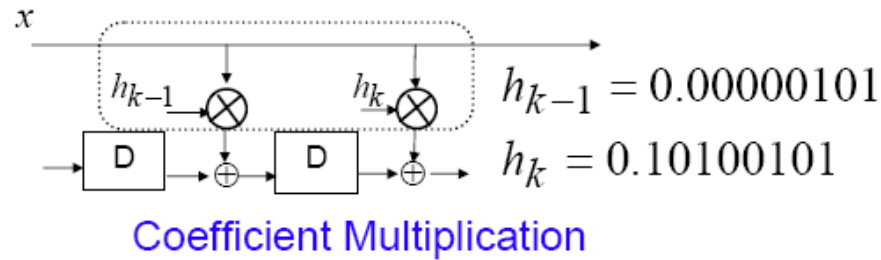


Transposed Direct form FIR Filter Structure

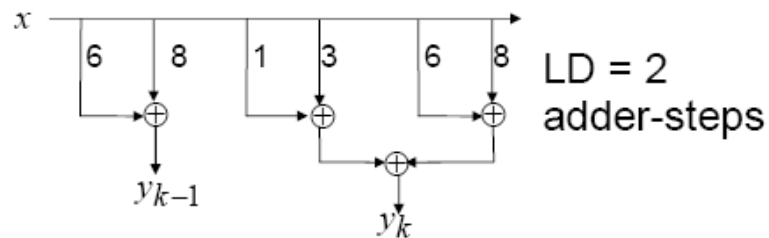
Extract common parts of $h(n)$ s and multiply with $x(n)$, thus eliminate redundant computations.

$$y(n) = x(n) \times \begin{bmatrix} h(0) \\ h(1) \\ \cdot \\ h(n) \end{bmatrix}$$

CSE – Basic Idea



Conventional CSD implementation (Direct implementation)



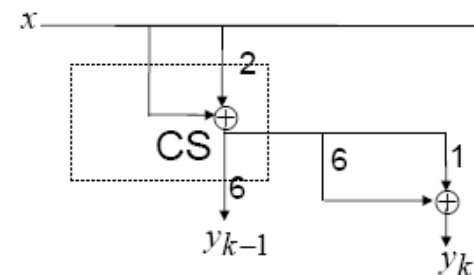
$$y_{k-1} = x[-1] \gg 6 + x[-1] \gg 8$$

$$y_k = x \gg 1 + x \gg 3 + x \gg 6 + x \gg 8$$

4 adders are needed.

CSE implementation

$$x_1 = x + x \gg 2 : \text{Common Subexpression: } (1 \ 0 \ 1)$$



Only 2 adders are needed.

Low Complexity Realizations

1. CSE Optimization by Combining Horizontal and Vertical Common Subexpressions.

- Two classes of Common Subexpressions (CS) – Horizontal and Vertical.

	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
$h(0)$					1	0	-1			1	0	1
$h(1)$									-1	0	1	
$h(2)$			1			-1					1	
$h(3)$			0			0					0	
$h(4)$			1			-1					-1	

Horizontal CS (HCS) points to the circled subexpressions in the first row: (1, 0, -1) and (1, 0, 1).

Vertical CS (VCS) points to the circled subexpressions in the third, fourth, and fifth columns: (1, 0, 1) in column 3, (0, 0, 0) in column 4, and (-1, -1, -1) in column 6.

Proposed CS Sharing Method:

- Efficiently combines the most common HCS and VCS using a look-ahead algorithm.
- Compatible VCS are chosen to completely exploit FIR filter coefficient symmetry.
- In the case of HCSE method, since all the bits forming an HCS exist within the coefficient, its symmetric counter-part can be easily implemented using delays (additional adders are not required).
- However in VCSE method, the bits occur across the coefficients and hence the symmetry is destroyed when the bits are of opposite sign - additional MBAs are required to obtain the symmetric part.

Experimental Results

FIR filter

$N=26$,

Pass-band edge: 0.2π

Stop-band edge: 0.25π

Method	8-bit CSD	Reduction rate (%)	16-bit CSD	Reduction rate (%)
VCSE	37	17.7	84	29.4
HCSE	35	22.2	72	39.4
Proposed Method	32	28.9	70	41.2

IEE Electronics Letters, January 2003

2. CSE Optimization by Super-subexpression Elimination

- Extract 2 bit CS
- Examine for multiple occurrences for identical shifts with non-zero bit or another CS
- Form Super-Subexpressions (SSs)

	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16
$h(0)$	1	0	1		-1			1	0	1			1	0	-1	
$h(1)$		1		-1	0	1		1	0	1			1	0	1	-1

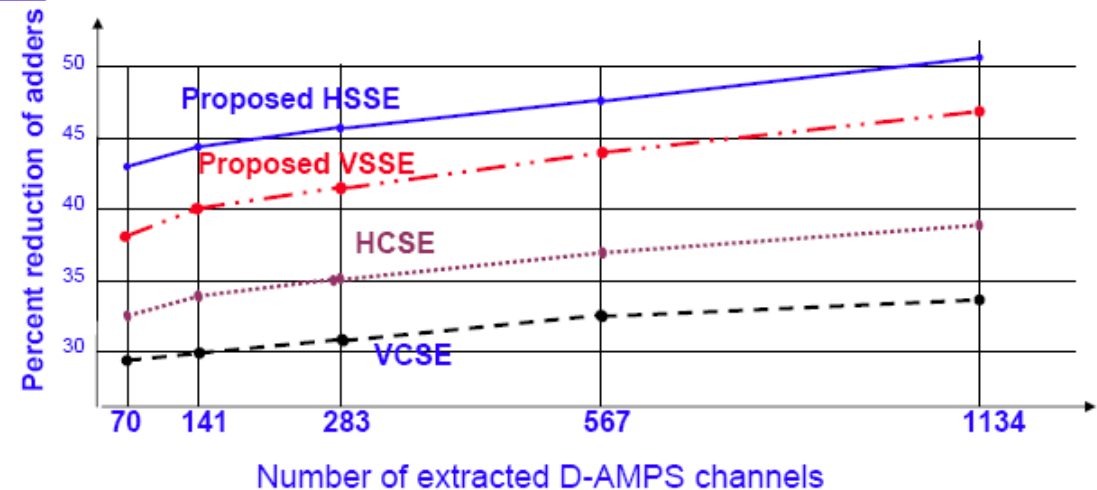
- Several such SSs are observed if filters have large number of taps.
- Investigation of 100 to 1200 tap filters with different stop band attenuation specifications and different word lengths revealed that following 3 bit SS and their negated version occurred very frequently (70%):
[1 0 1 0 1], [1 0 1 0 -1], [-1 0 1 0 1], [-1 0 1 0 -1] and their negated versions.

Experimental Results

Filter bank Channelizer for D-AMPS standard

- Sampling rate of input signal – 32.04MHz
- BW of channels – 30KHz
- PB edge – 30KHz, SB edge- 30.5KHz
- PB ripple 0.1 dB
- Filter coefficients (CSD) – 12 and bits

- PDC Channelizer was also examined.



Reduction of adders to implement the D-AMPS channel filters for different number of channels extracted.

- **Average reduction of complexity compared to best known method: 30%**

(IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems, Feb. 2005)

Coefficient-Partitioning Method

- The CSE algorithms evaluate the complexity of adders in terms of the number of adders used in the multipliers – do not analyze the complexity of each adder.
- We analyzed the complexity of each adder in terms of the number of full adders (FAs) needed and proposed an efficient coefficient-partitioning method to minimize the number of full adders.
- The complexity of each adder used in the multiplier is significant in practical implementations, as it determines the actual cost of the adder.
- The area, power and speed of an adder depend on its complexity.
- An adder that adds two n bit numbers needs at the most $(n+1)$ full adders to compute the sum. (A ripple carry adder is assumed here on account of its low-power consumption).
- Efforts to optimize area, power and speed should focus on minimizing the number of FAs required to implement the multipliers.

- Recently, we proposed a coefficient-partitioning method (CPM) which reduces the complexity of each adder in the coefficient multiplier - First approach in literature which focuses on filter complexity reduction at full adder level.

Illustration

$$h_k = 0.0000101001010101 = 2^{-5} + 2^{-7} + 2^{-10} + 2^{-12} + 2^{-14} + 2^{-16}.$$

HCS: $x_2 = x_1 + x_1 \gg 2,$ **Output:** $y_k = 2^{-5}x_2 + 2^{-10}x_2 + 2^{-14}x_2$

- 59 FAs are needed to implement the multiplier block using CSE.

Proposed Coefficient-Partitioning:

Pseudo-Floating-Point (PFP) form of CSE: $2^{-5}(x_2 + 2^{-5}x_2 + 2^{-9}x_2)$

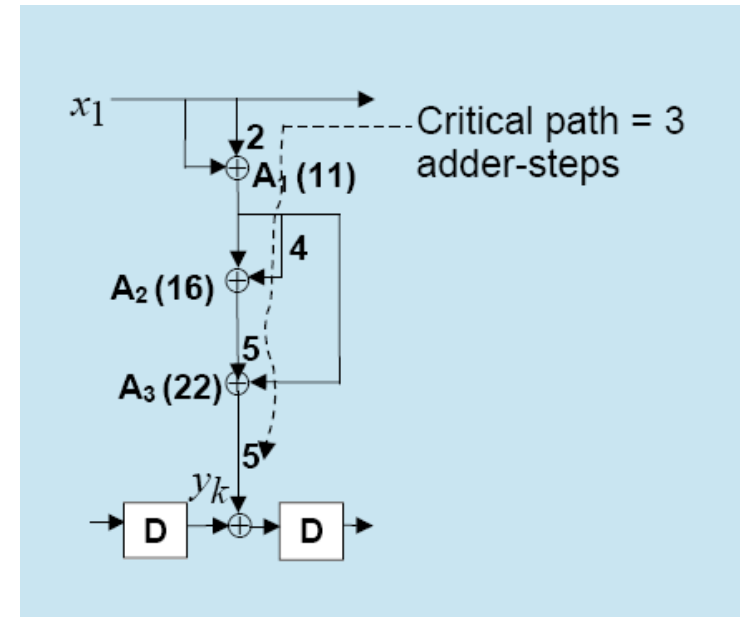
Partitioning the span part into two sub-coefficients, $h_1(n)$ and $h_2(n)$, we have:

$$h_1(n) = x_2 \text{ and } h_2(n) = 2^{-5}x_2 + 2^{-9}x_2$$

PFM of the later sub-coefficient: $h_2(n) = 2^{-5}(x_2 + 2^{-4}x_2)$

CPM Filter Multiplier

- Only 49 FAs are needed (reduction of 17% over CSE).
- Critical path = 3 adder-steps in both methods, i.e., identical delay.



- **Key idea of our Coefficient-Partitioning:** Reduce the ranges of the operands so that the adder width can be reduced; this in turn minimizes the number of FAs.

A.P.Vinod and E.M-K.Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," *IEEE Transactions on Wireless Communications*, July 2006.

A.P.Vinod and E.M-K.Lai, "An efficient coefficient-partitioning algorithm for realizing low complexity digital filters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Dec 2005.

Experimental Results

FIR filters (10 to 400 taps), stop-band ripple specs (20 dB to 95 dB) at various sampling frequencies for GSM, W-CDMA, PDC and DAMPS channelizers.

Reduction of full adders over the direct method in designing the FIR filter with coefficient wordlength of 16 bits, for different number of filter taps.

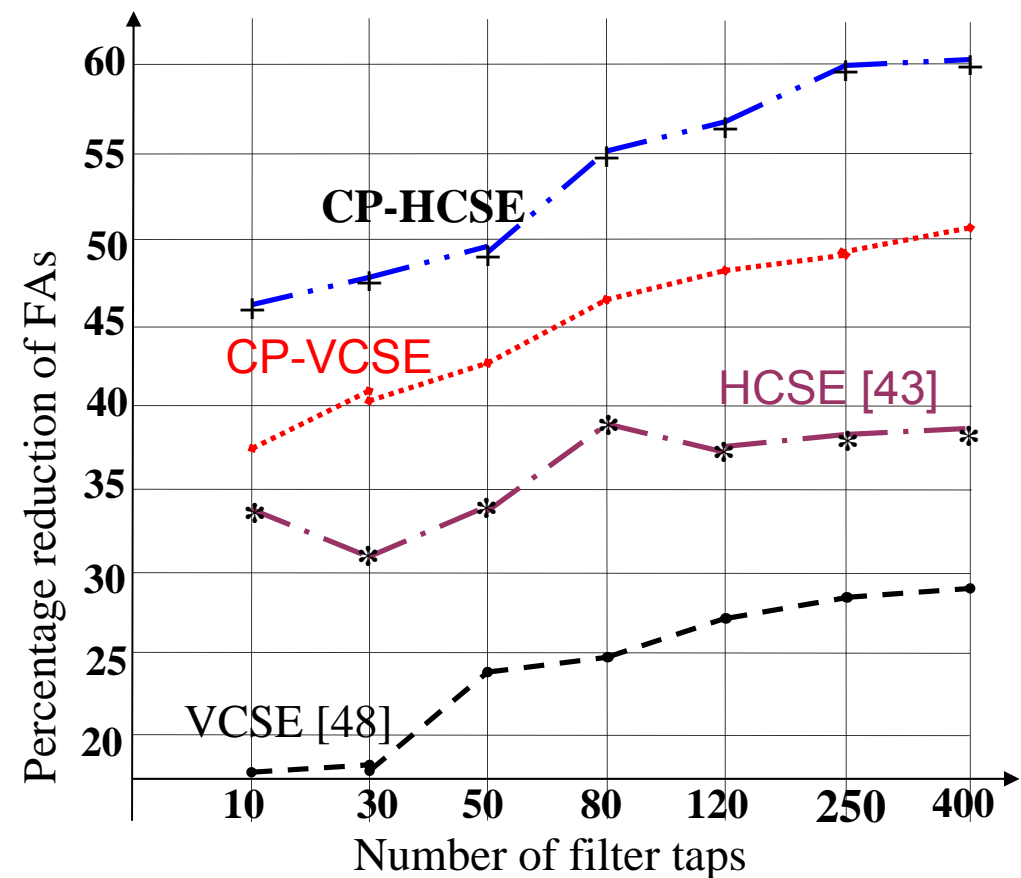
Average reduction:

- **Our CP-HCSE: 54%**
- **Our CP-VCSE: 44.2%**
- **HCSE: 36.4%**
- **VCSE: 22.4%**

Higher reduction as #taps increases.

Reduction of CP-HCSE over HCSE

- **10 taps: 11.3%, 400 taps: 22%**



Drawbacks of Existing CSE Algorithms

- ❖ Existing CSE algorithms are CSD based, hence can't be easily extended to reconfigurability because of subtraction operation involving negative bits.
- ❖ In general, none of the CSE algorithms in literature deal with higher-order filters.

Proposed Binary Subexpression Elimination

Analysis: Why Binary works better than CSD?

- ❖ In general, the number of LOs, $N_{LO} \approx (0.2345 \times N_{nz} - 0.6643 \times N_{CS} + 4.0487 \times N_{up})$,

N_{nz} = The number of non-zero digits before the application of CSE technique

N_{CS} = The number of CSs

N_{UP} = The number of un-paired digits (bits) which do not form CSs.

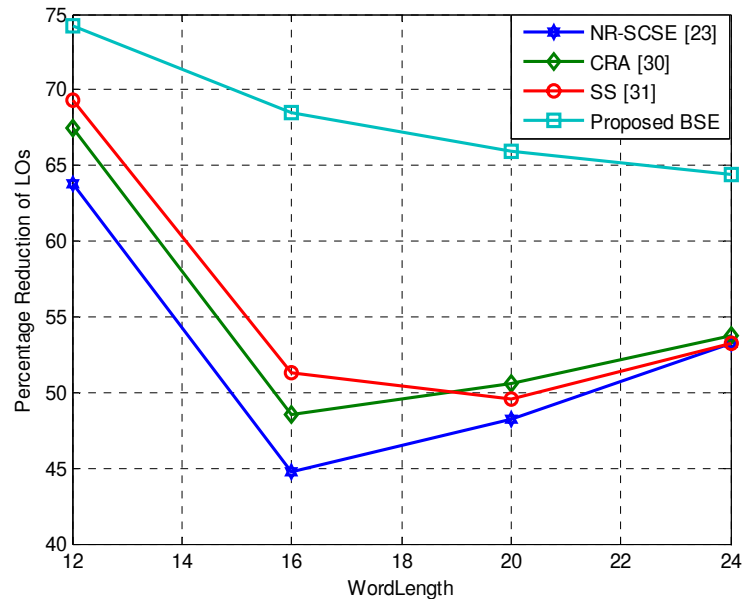
- ❖ Binary has 27.7% increase of N_{nz} and 6.7% increase of N_{CS} over CSD. On the other hand, the N_{up} of binary is less by 67% compared to CSD.
- ❖ Conclusion: Binary will offer better adder reductions than CSD.

Binary Subexpression Elimination (BSE)

Experimental Results (D-AMPS Standard)

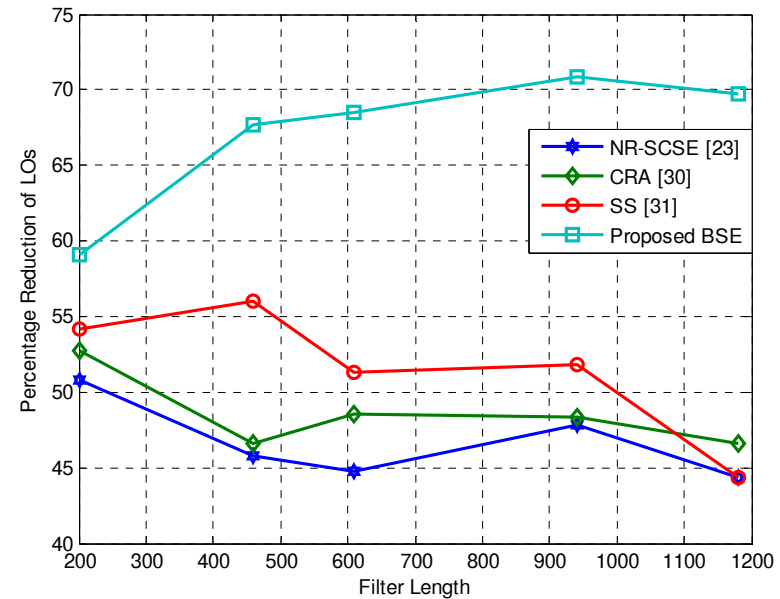
FIR Filter Specifications $w_p = 0.6173\pi$ and $w_s = 0.6276\pi$

Reduction of LOs in designing the filter with 610 taps over direct method.



Proposed BSE offers percentage reductions of 16% over NR-SCSE [23], 14% over CRA [30] and 11% over SS [31].

Reduction of LOs in designing the filter with 16-bit coefficient word length



Proposed BSE offers percentage reductions of 26% over NR-SCSE [23], 23% over CRA [30] and 20% over SS [31].

Over-all Comparison

	NRSCSE	CRA	HCUB	SS
LO	24%	18%	7%	17%
LD	-7.14%	12.5%	53.5%	14%

- ❖ Reconfigurability is another key requirement of channel filters in SDRs.
- ❖ BSE employs binary representation of filter coefficients. Hence can be easily applied to reconfigurable filters.

-
- NRSCSE:** M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. on Circuits and Syst. II*, vol. 49, no. 3, pp. 196-203, Mar. 2002.
- CRA:** F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution algorithm for common subexpression elimination in digital filter design," *IEEE Trans. on Circuits and Syst. II*, vol. 52, pp.695-700, Oct. 2005.
- HCUB:** Y. Voronenko and M. Pushcel, "Multiplierless Multiple Constant Multiplication," to appear in *ACM Transactions on Algorithms*.
- SS:** A.P. Vinod and M-K Lai, "On the implementation of efficient channel filters for wideband receivers by optimizing common subexpression elimination methods", *IEEE Trans. on Computer-Aided Design*, vol. 24, pp. 295-304, Feb. 2005.

R.Mahesh and A.P.Vinod, "A new common subexpression elimination algorithm for realizing low complexity higher order digital filters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, Accepted for publication (to appear in January 2008).