



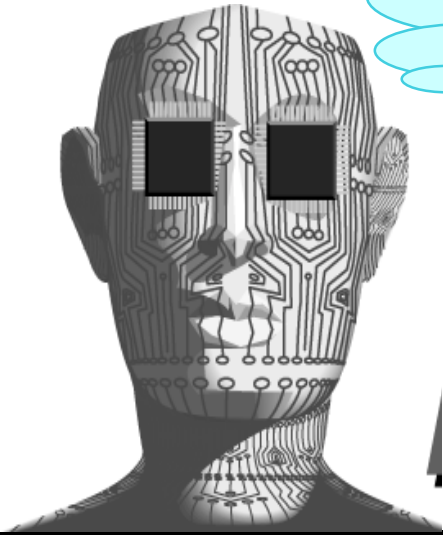
# Modulation and Demodulation Techniques for FPGAs

**DESIGNCON 2000**

**Ray Andraka P.E., president,**

 *the Andraka Consulting Group, Inc.*  
16 Arcadia Drive • North Kingstown, RI 02852-1666 • USA  
401/884-7930 FAX 401/884-7950

You can do *Math*  
in them things???





# Overview

- Introduction
- Digital demodulation for FPGAs
- Filtering in FPGAs
- Comparison to other technologies
- Summary



# Digital Communications

- Historically only base-band processing
- High sample rates for down-converters
- Down-conversion traditionally analog
- Digital down-conversion with specialty chips
- FPGAs can compete



# Why Digital?

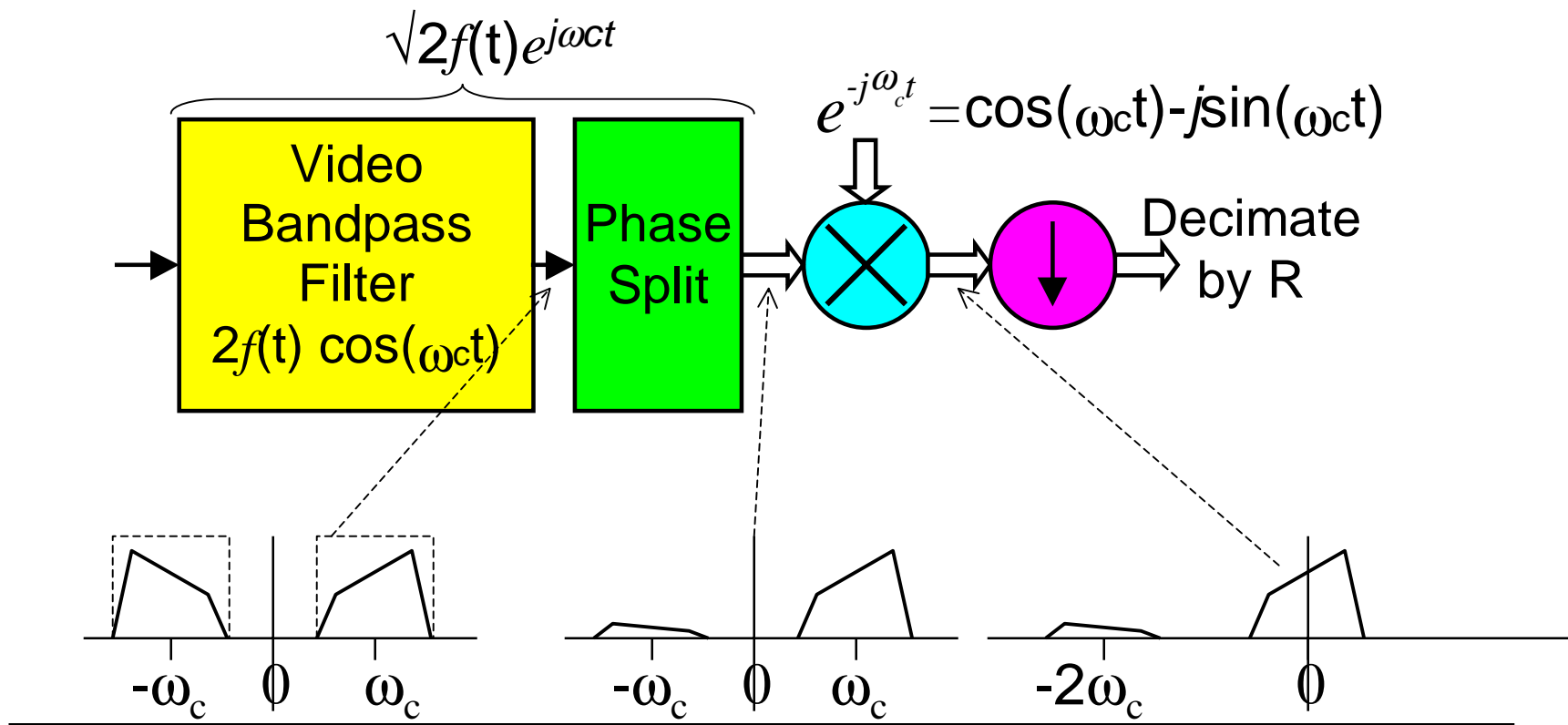
- Frequency agility
- Repeatability
- Cost



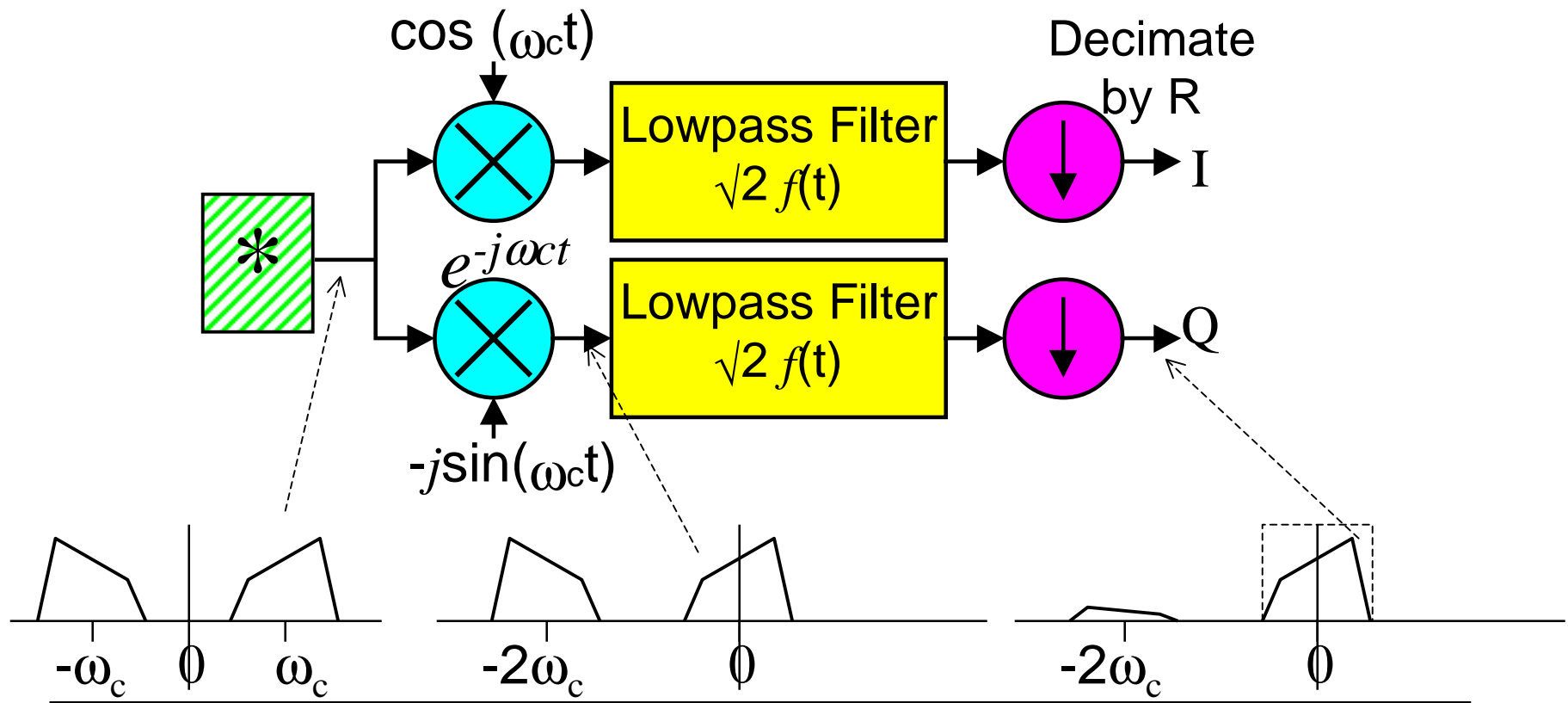
# Digital Challenges

- A to D converter
- High sample rates
- Arithmetic intensive

# Conventional Demodulator

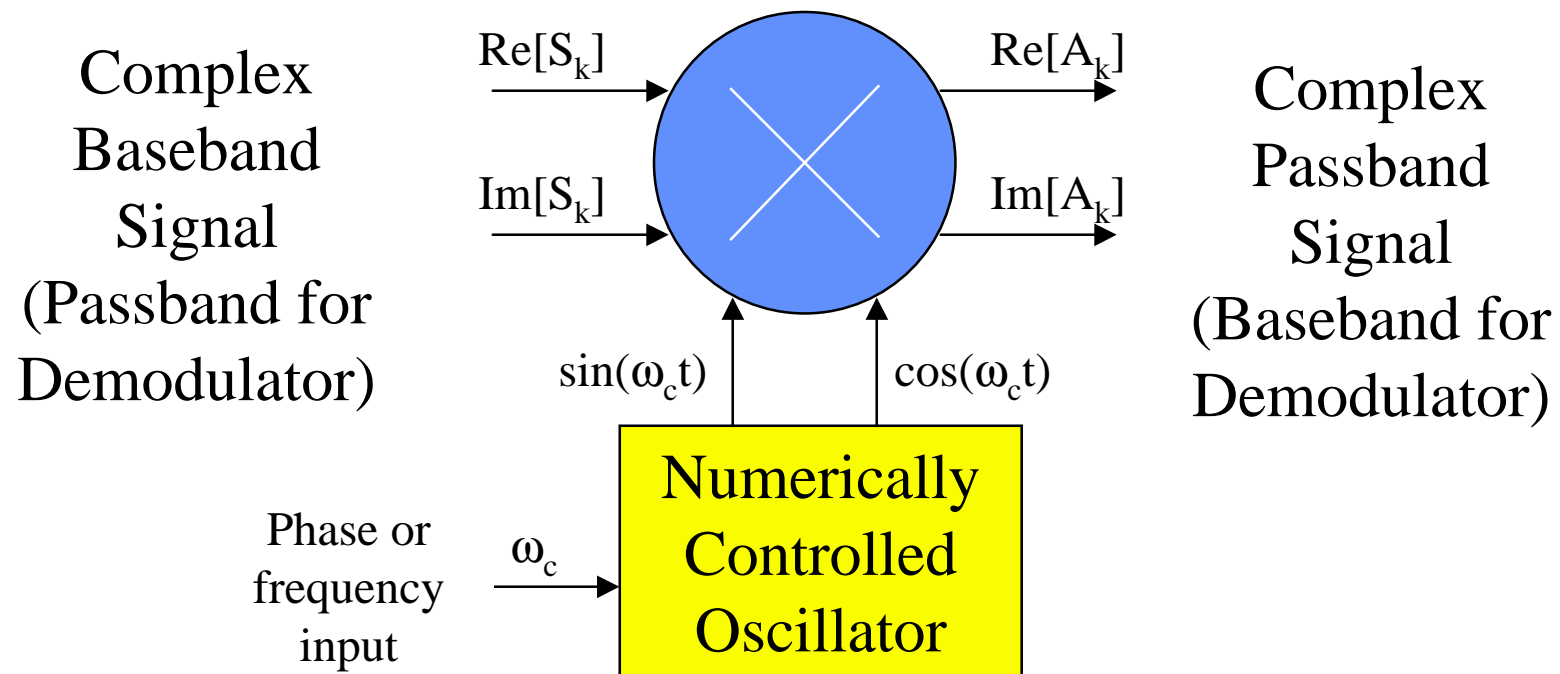


# Re-arranged Demodulator



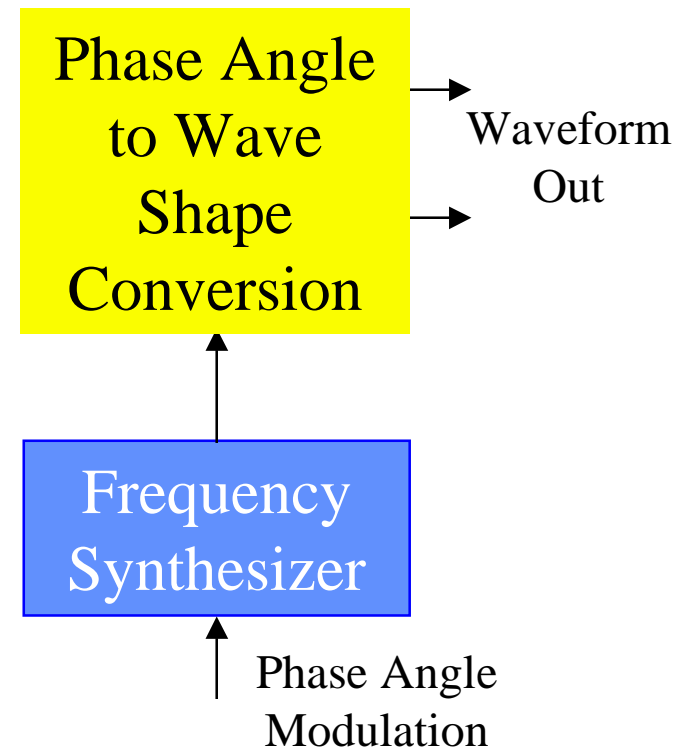


# Complex Mixer



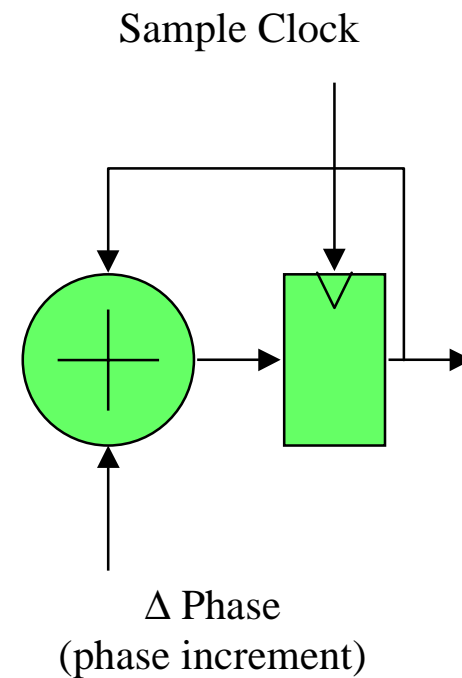
# Waveform Synthesis (NCOs)

- Various Methods
  - Look up table (LUT)
  - Partial products
  - Interpolation
  - Algorithmic
- Most methods use a frequency synthesizer



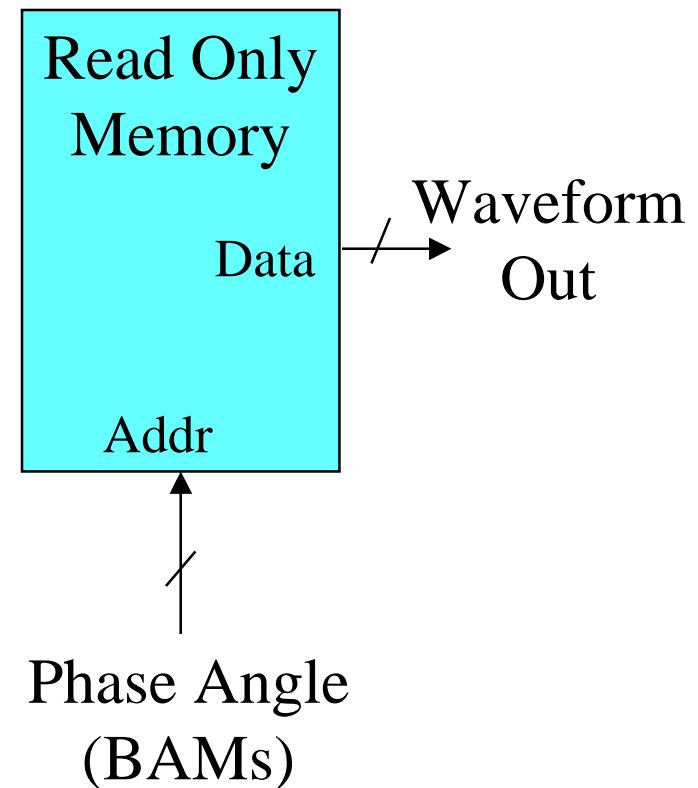
# Phase Accumulator Design

- “Direct Digital Synthesis”
- Essentially integrates phase increment
- Increment value may be modulated
  - Frequency and PSK modulation
- Binary Angular Measure (BAMs)
  - Most significant bit =  $\pi$

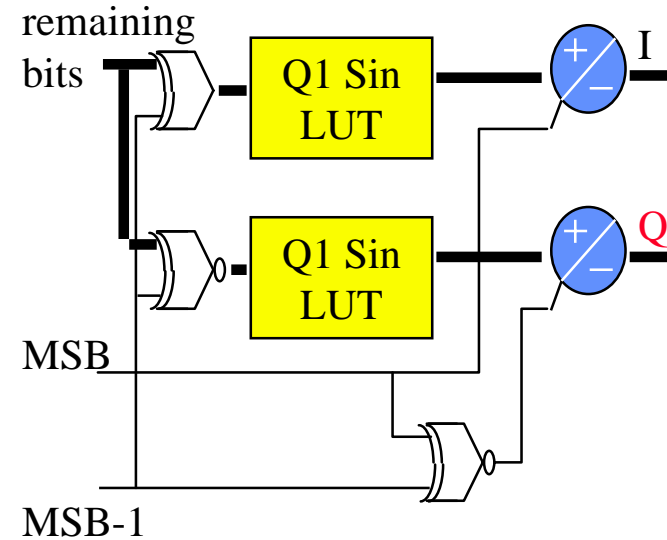
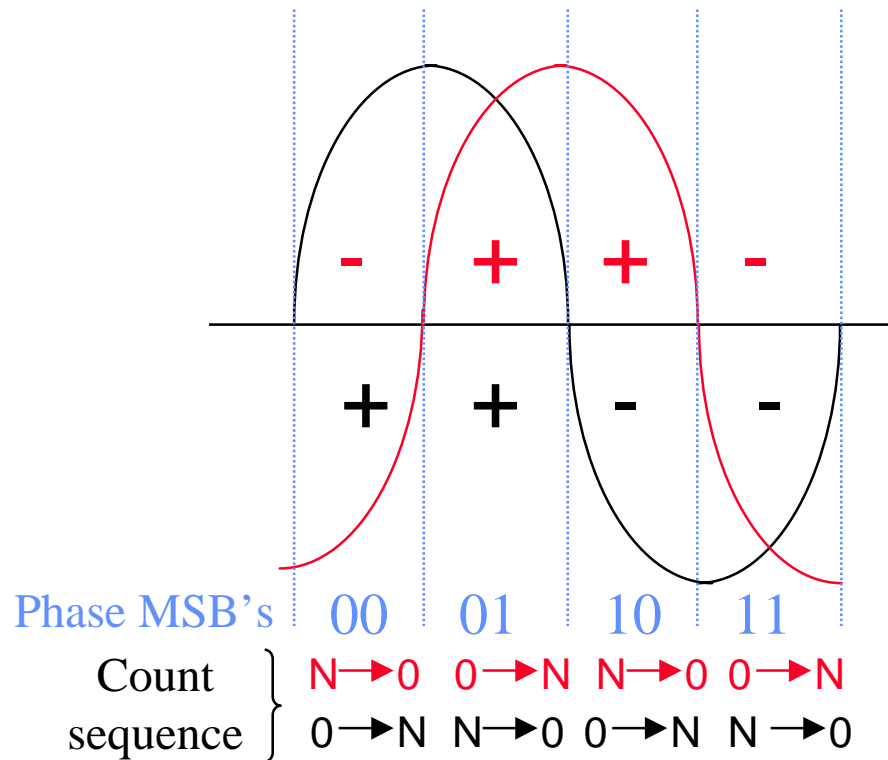


# Waveform Synthesis by LUT

- Phase resolution limited
- Arbitrary waveshapes
- Sampled waveshape must be band-limited
- Complex requires 2 lookups
- $f_{osc} = f_s / 4$  special case

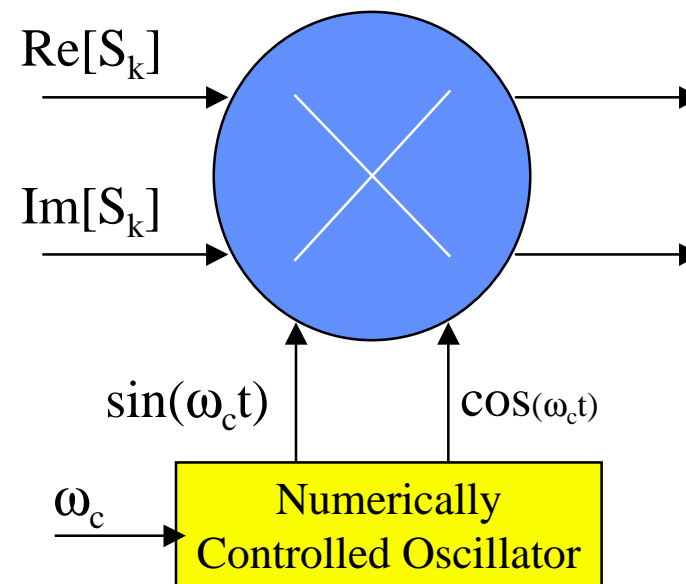


# Using Symmetry to Extend LUT Phase Resolution



# Waveform Synthesizer plus Multiplier

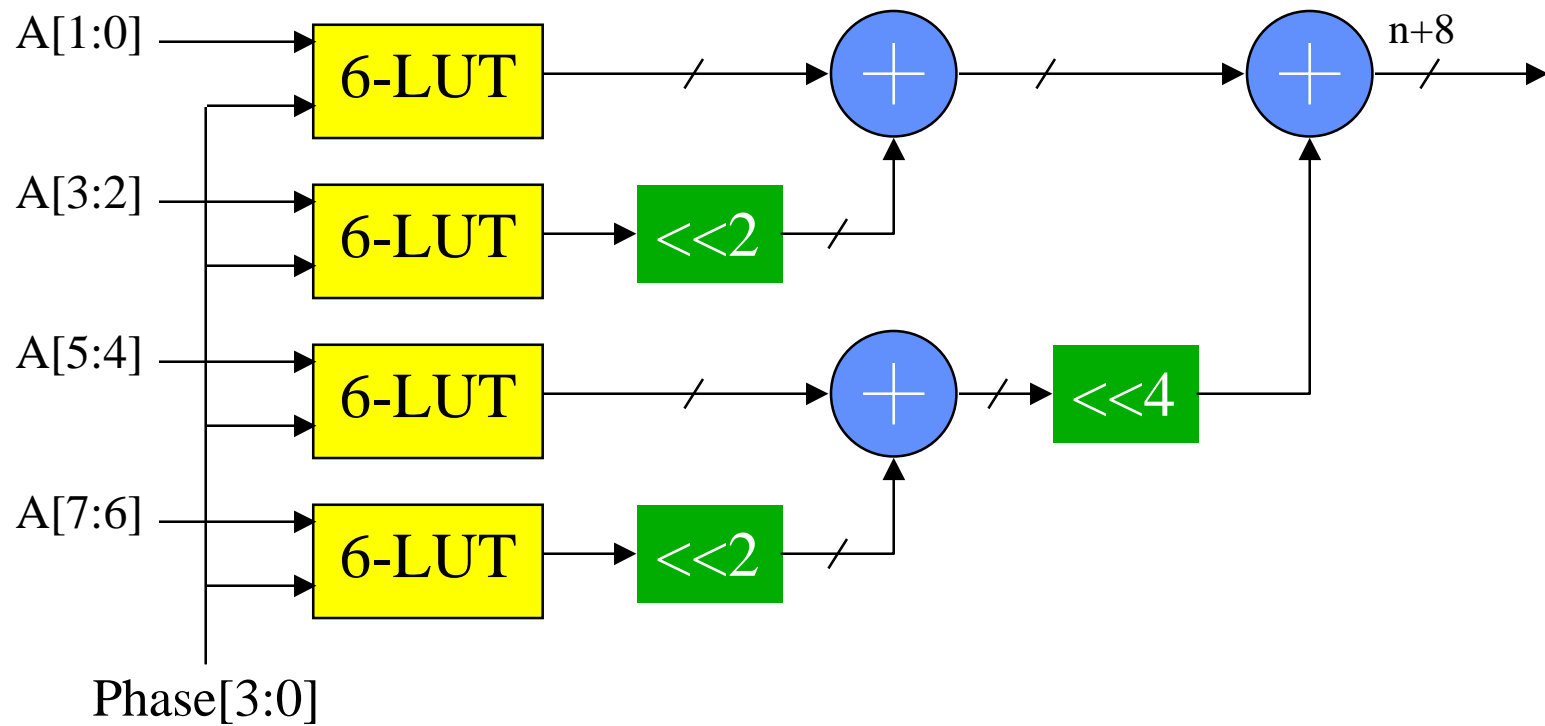
- Obvious Solution
- Separate into functional parts
- Treat each part independently



## Look Up Table Modulator

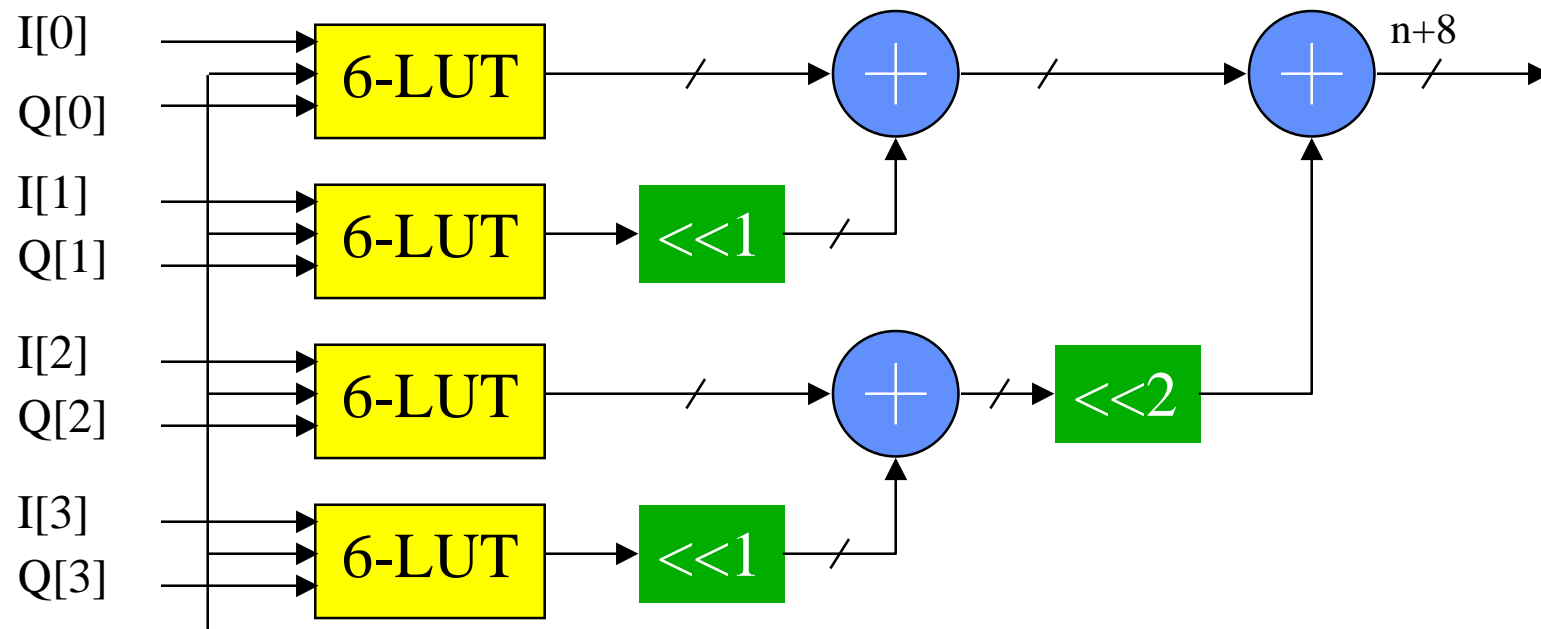
a Cos( $\phi$ )	phase							
signal	000	001	010	011	100	101	110	111
-4	-4	-2.8	0	2.8	-4	2.8	0	-2.8
-3	-3	-2.1	0	2.1	3	2.1	0	-2.1
-2	-2	-1.4	0	1.4	2	1.4	0	-1.4
-1	-1	-0.7	0	0.7	1	0.7	0	-0.7
0	0	0	0	0	0	0	0	0
1	1	0.7	0	-0.7	-1	-0.7	0	0.7
2	2	1.4	0	-1.4	-2	-1.4	0	1.4
3	3	2.1	0	-2.1	-3	-2.1	0	2.1

# Partial Products Modulator



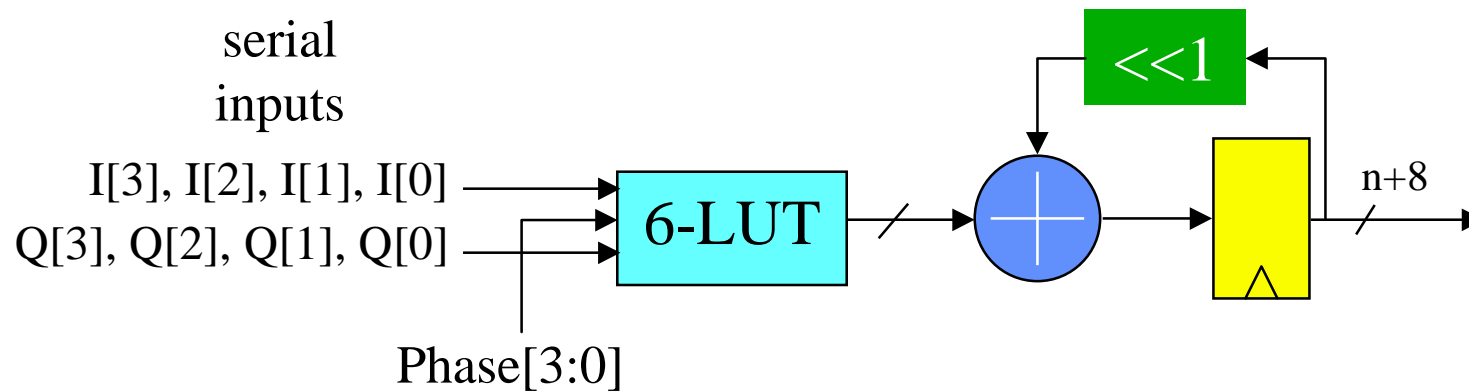


# Distributed Arithmetic Modulator



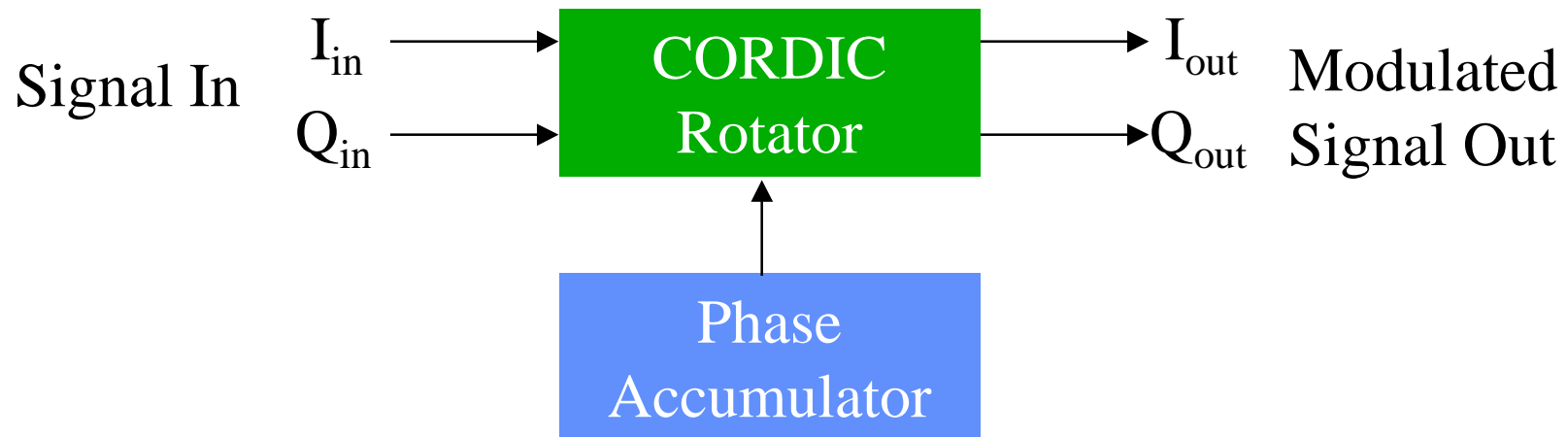
Phase[3:0]

# Distributed Arithmetic Modulator (Serial Form)



# CORDIC Modulator

$$\begin{aligned} I_{out} &= I_{in} \cos\phi - Q_{in} \sin\phi \\ Q_{out} &= Q_{in} \cos\phi + I_{in} \sin\phi \end{aligned}$$



# CORDIC Algorithm Explained

- **Coordinate rotation in a plane:**

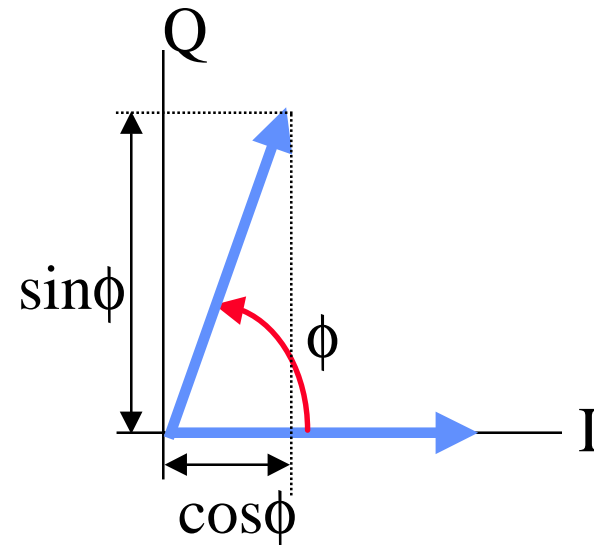
$$x' = x\cos(\phi) - y\sin(\phi)$$

$$y' = y\cos(\phi) + x\sin(\phi)$$

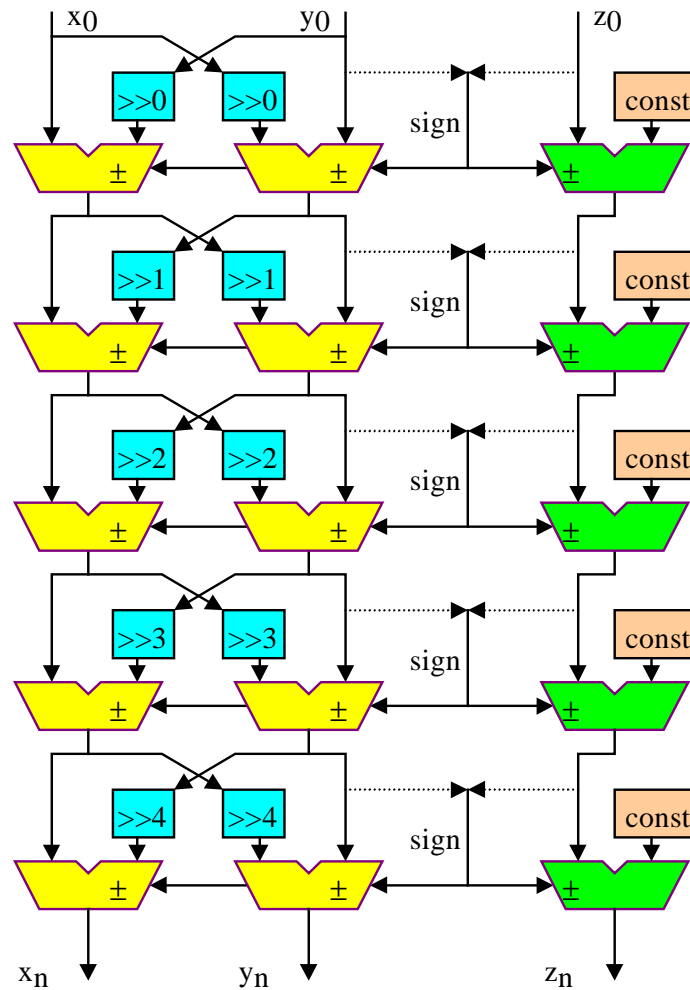
- **Rearranges to:**

$$x' = \cos(\phi) [x - y\tan(\phi)]$$

$$y' = \cos(\phi) [y + x\tan(\phi)]$$

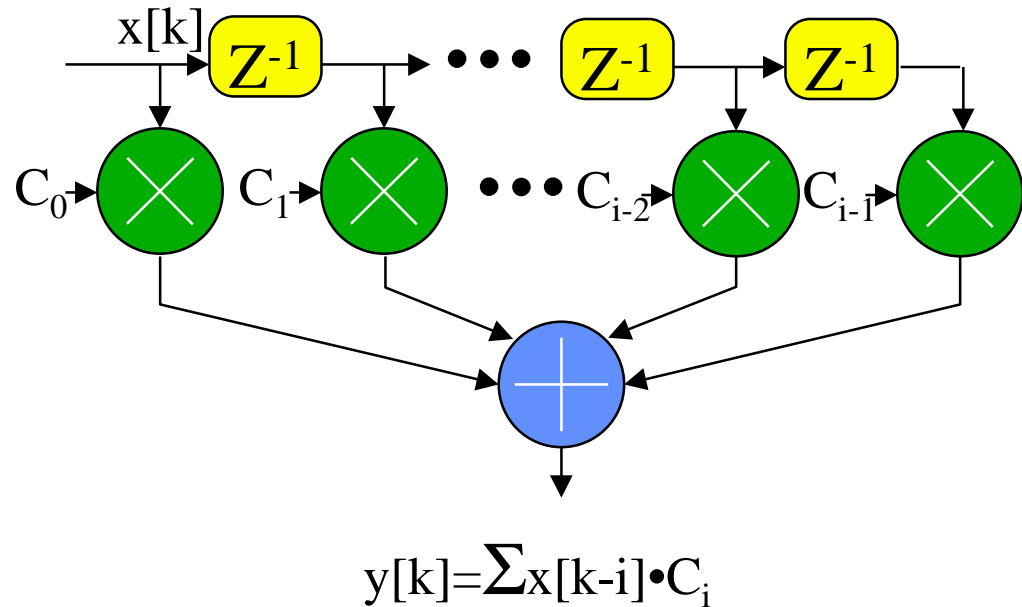


# CORDIC Structure

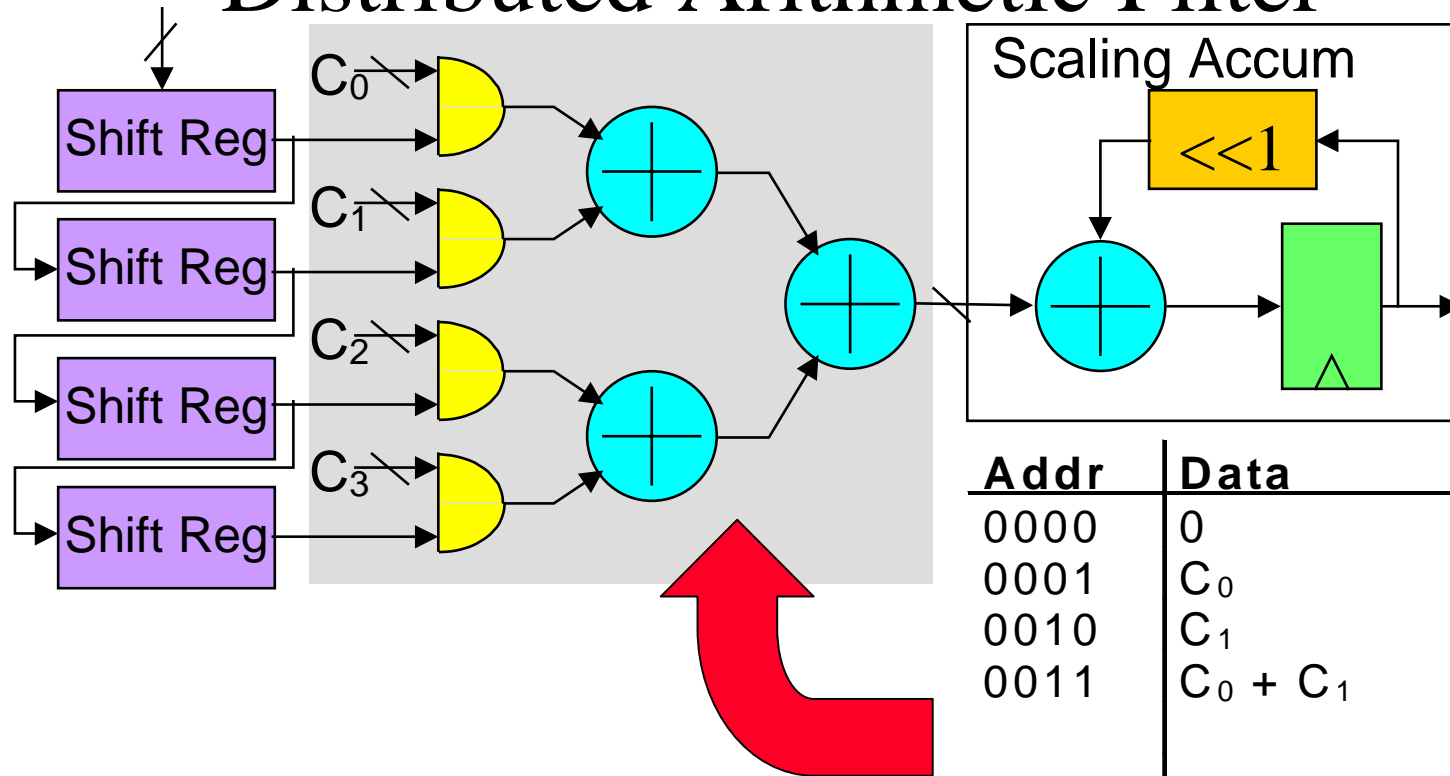


# Digital Filtering

- **Many Constant Multipliers**
- **Delay Queues**
- **Products Summed**
- **Advantages**
  - No tolerance drift
  - Low cost
  - precise characteristic



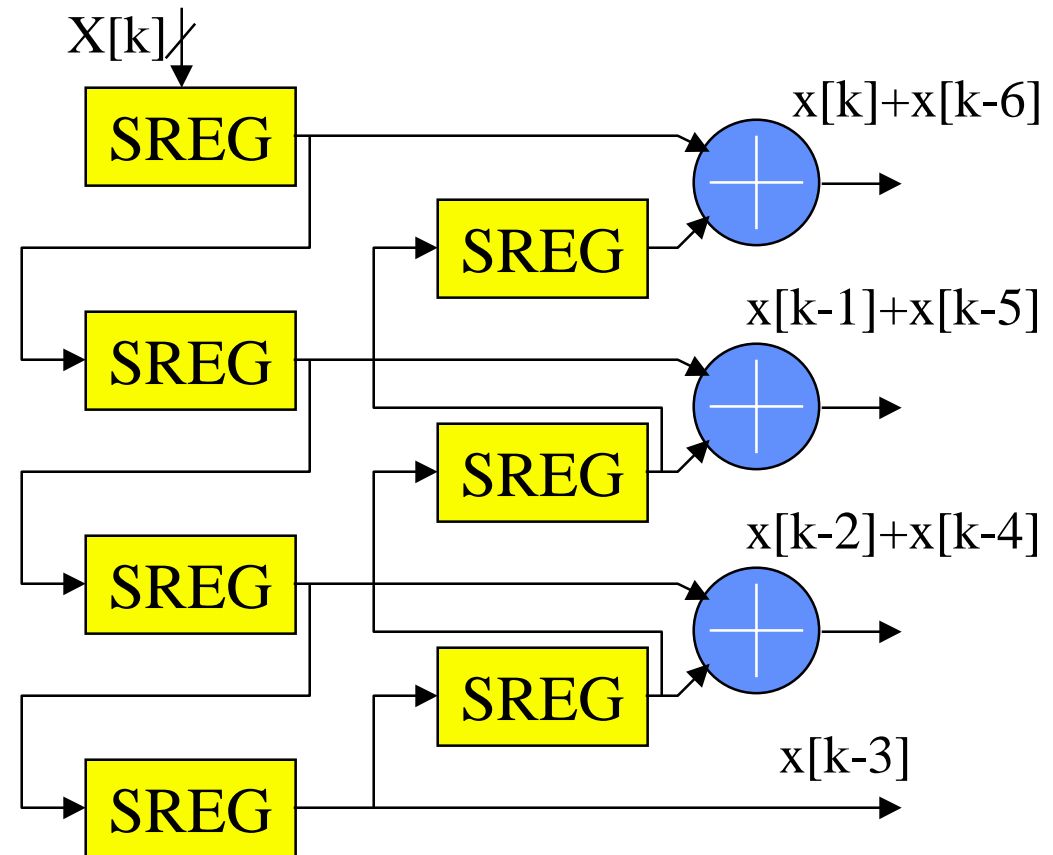
# Distributed Arithmetic Filter



Addr	Data
0000	0
0001	$C_0$
0010	$C_1$
0011	$C_0 + C_1$
1110	$C_1 + C_2 + C_3$
1111	$C_0 + C_1 + C_2 + C_3$

# Take Advantage of Symmetry

- Real filters are symmetric
- Add bits with like coef's before filtering
- Uses Serial Adders
- Halves taps





## Decimating FIR Filters

- Low pass filter then discard samples
- Keep only every 4th output

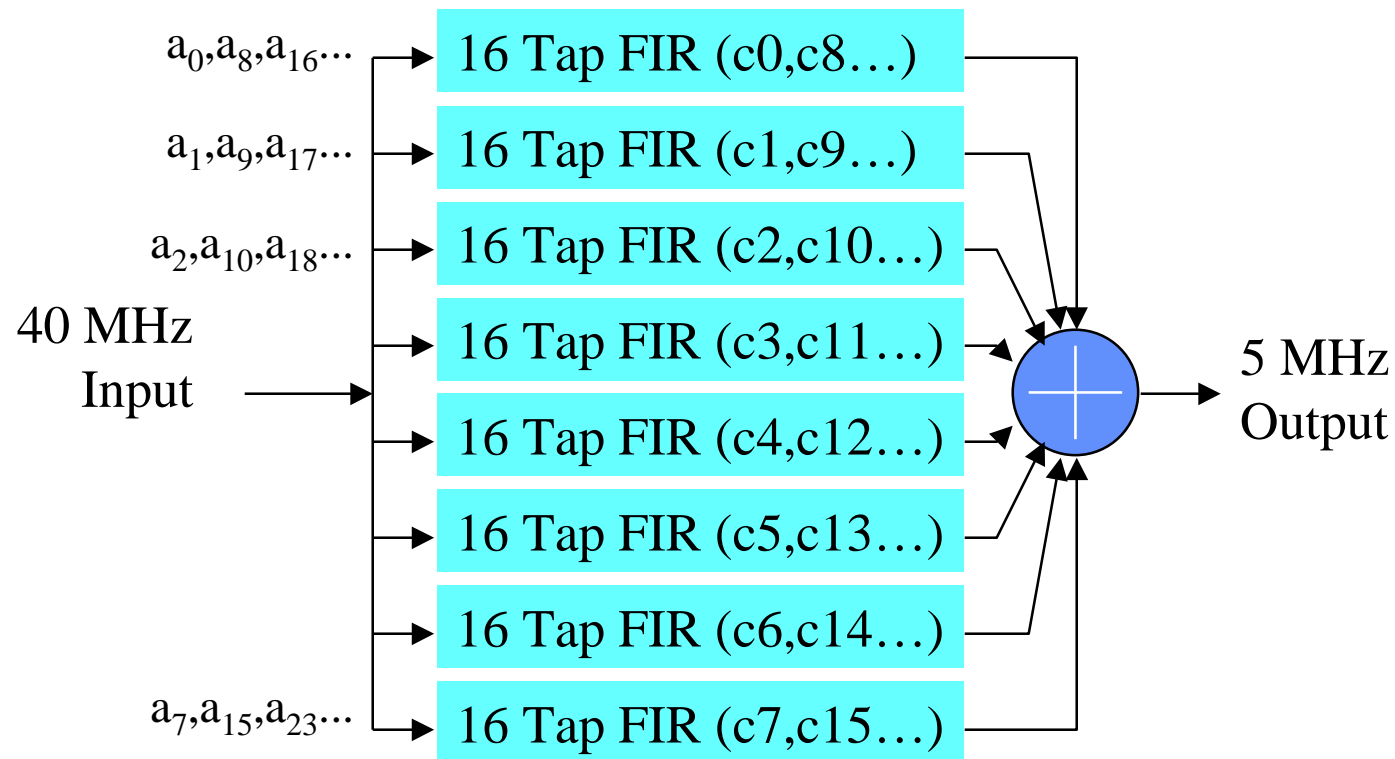
$$Y_{n+0} = a_k C_0 + a_{k-1} C_1 + a_{k-2} C_2 + a_{k-3} C_3 + a_{k-4} C_4 + \dots$$

$$Y_{n+4} = a_{k+4} C_0 + a_{k+3} C_1 + a_{k+2} C_2 + a_{k+1} C_3 + a_k C_4 + \dots$$

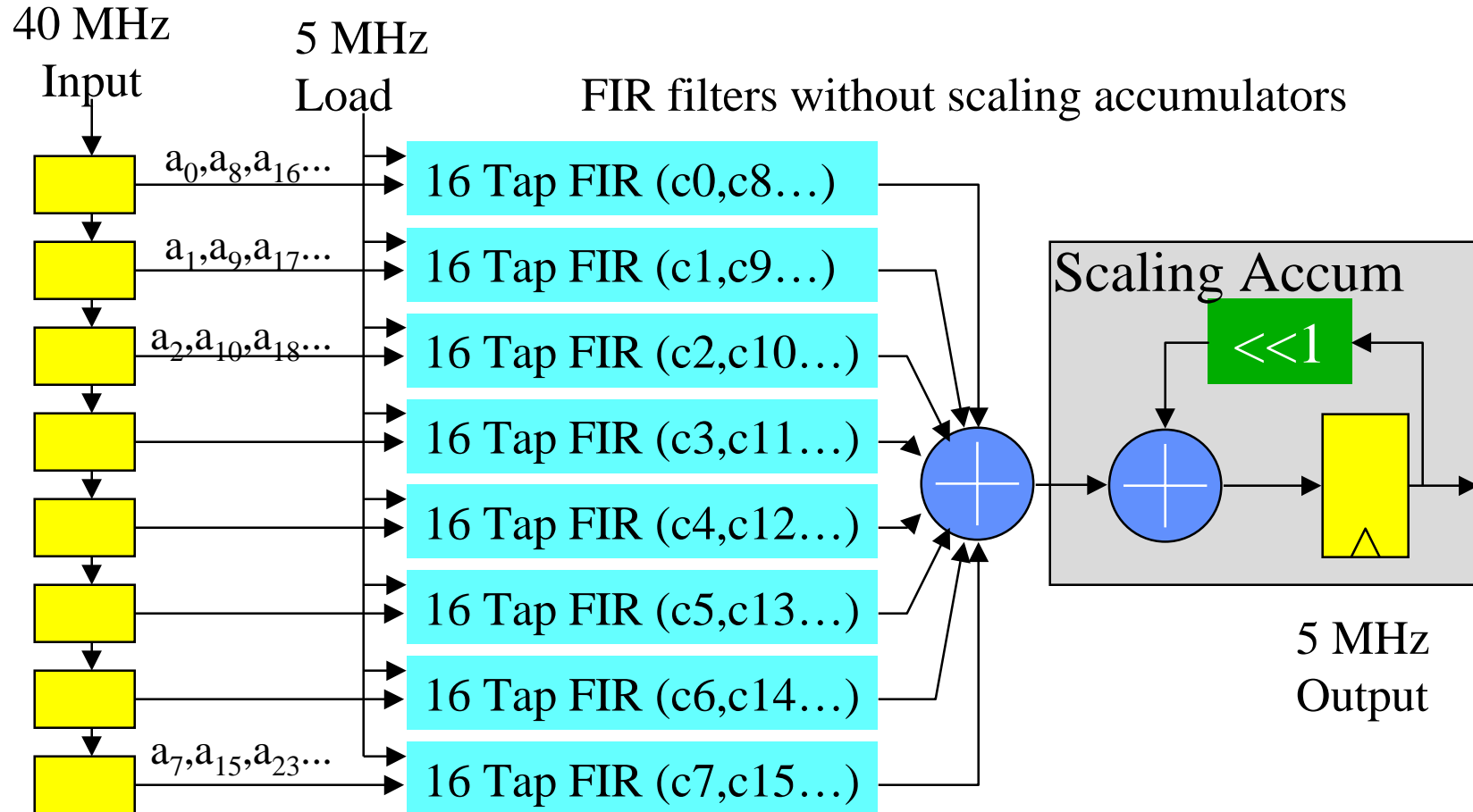
$$Y_{n+8} = a_{k+8} C_0 + a_{k+7} C_1 + a_{k+6} C_2 + a_{k+5} C_3 + a_{k+4} C_4 + \dots$$

- reduces to n parallel filters fed every nth sample
- sub-filter results summed to get result

# 128 tap 8:1 Decimating FIR Filter

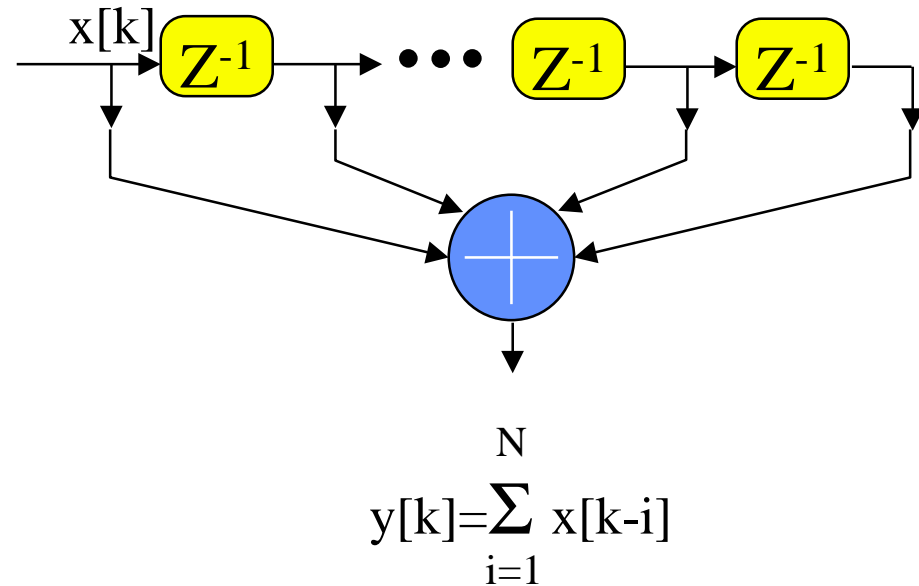
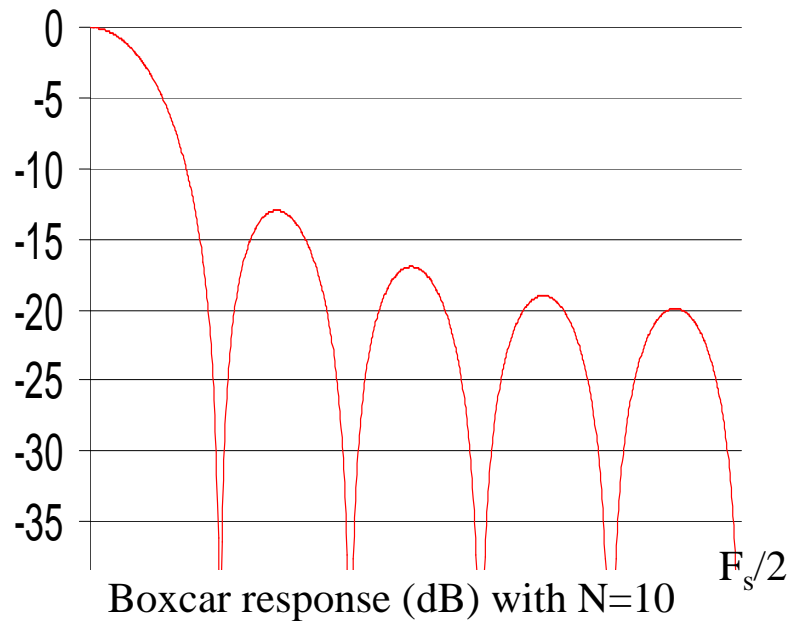


# Decimating FIR Filter Reduction



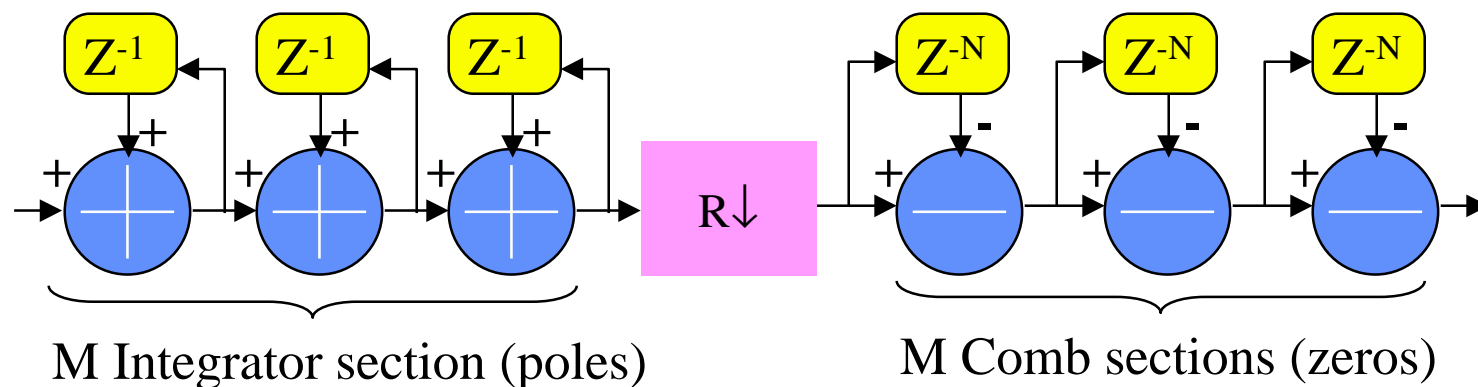
# Multiplier-less Filtering

- Boxcar or Moving Average filter
- FIR with unity coefficients

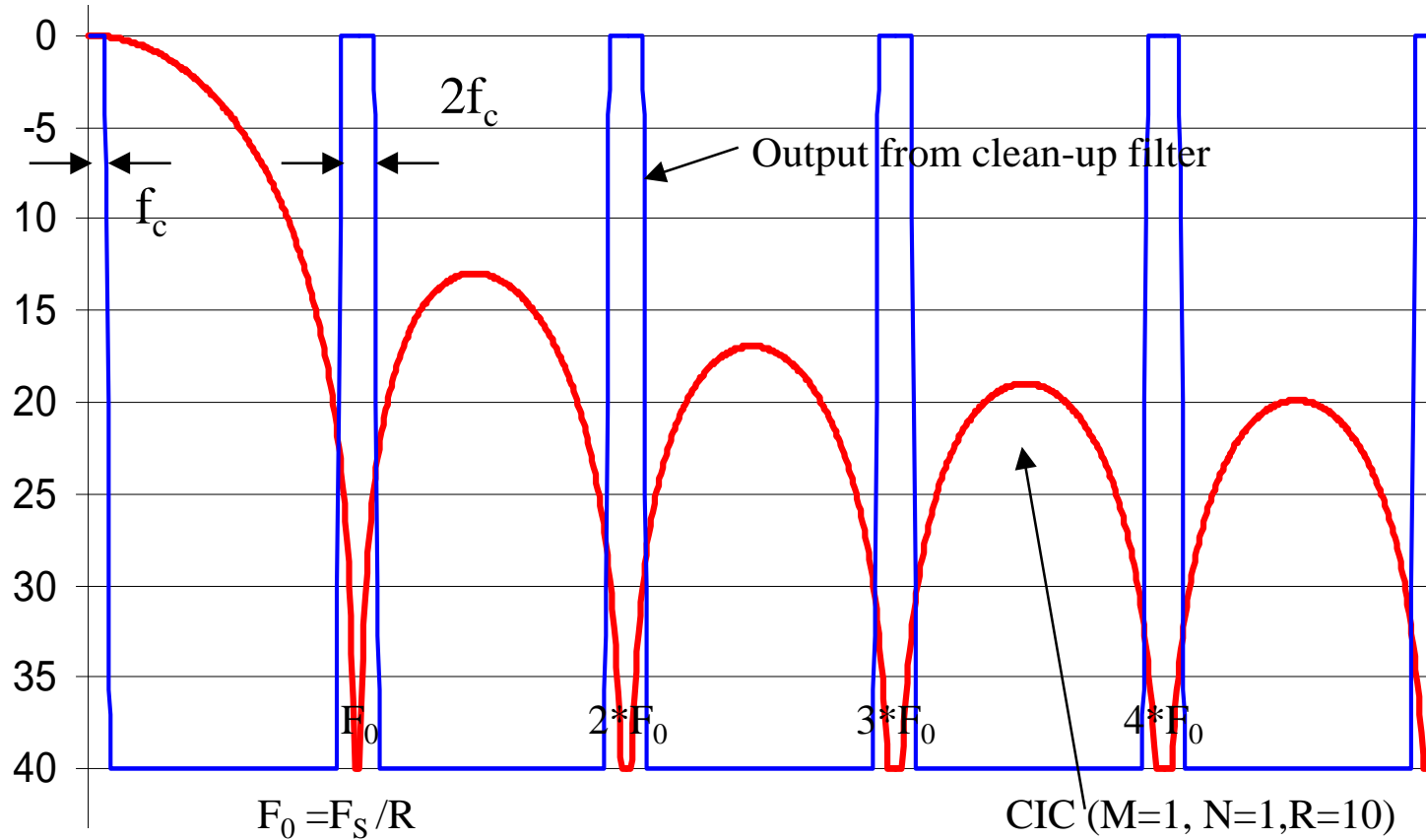


# Cascaded Integrator-Comb Filters

- Response same as  $M$  cascaded  $N \cdot R$  boxcar filters
- High order multiplier-less interpolation or decimation
- Constant response relative to decimated sample rate
- Use small FIR to shape response

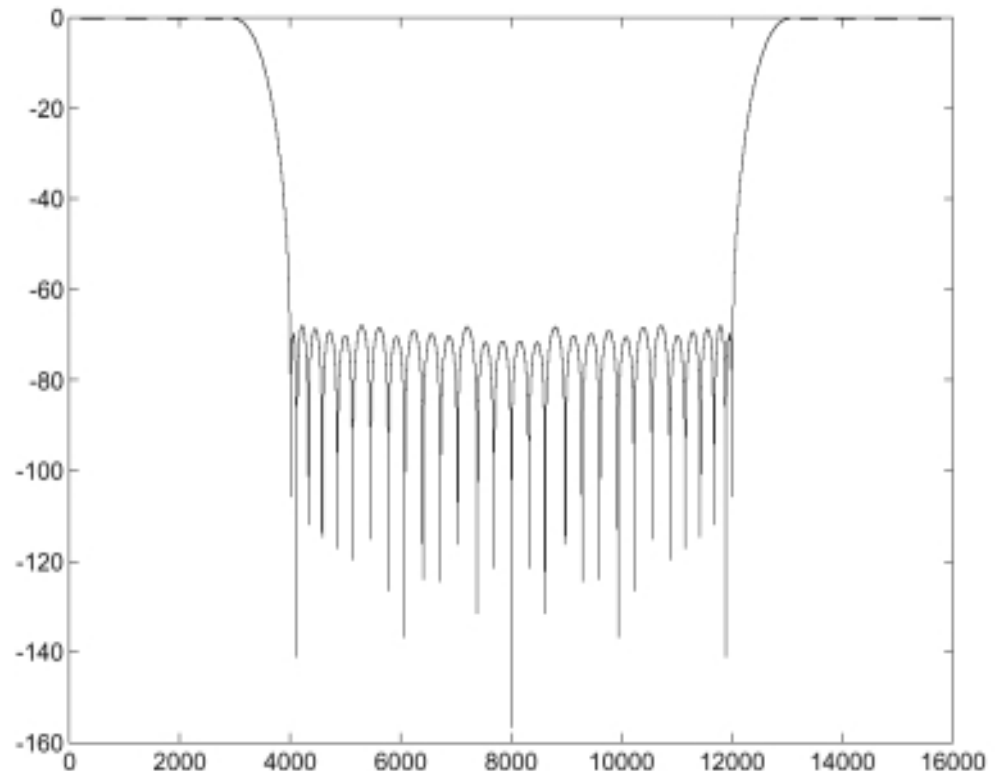


# Frequency Sampling with CIC



## Half Band Filters

- Special case of FIR filter
- Nearly half of the coefficients are zero
- Response is anti-symmetric about  $F_s/4$
- 15th order half-band filter is only 5 taps



$$C_i = [-15, 0, 84, 0, -296, 0, 1251, 2048, 1251, 0, -296, 0, 84, 0, -15]$$



# Comparison to dedicated digital modulator chips

- **Performance similar to dedicated chips**
- **Can be tailored to exact requirements**
- **Other logic can be integrated into chip**
- **Filtering in dedicated chips sometimes better**
- **Some development required**
- **FPGA generally cheaper than Digital Modulator Chips**





## Comparison to DSP Micros

- **Higher performance than DSP micro**
- **Higher integration**
- **Cost is comparable considering performance**
- **Hardware vs. software development**



# Design Considerations

- **Floorplanning required for performance and density**
- **Macro generator tools simplifying design task**
- **Use instantiation in logic synthesis**



## Other Considerations

- **Transmitter**
  - filter needed to minimize ISI
  - Carrier can be coordinated with symbol rate
- **Receiver**
  - filter for noise rejection, correction of channel distortion
  - Accurate phase reference for carrier needed
  - Timing normally recovered from signal
  - Coordinate IF, sample frequency, symbol frequency



# Summary

- **Approach depends on performance, resolution and size**
- **Competes with dedicated digital modulator chips**
- **Can be tailored to exact requirements**
- **Each design has potential for hardware shortcuts**



# Resources

- **FPGA Vendor Application Notes**
  - **Xilinx web page <http://www.xilinx.com>**
  - **DSP applications notes**
- **Tools**
  - **DSP toolbox for Xilinx**
  - **Vendor DSP Macro Generators**
- **Consulting services, Training, etc.**
  - **Andraka Consulting Group, Inc**
  - **401/884-7930**
  - **web page: <http://users.ids.net/~randraka>**



## References

- M. Frerking, “Digital Signal Processing in Communication Systems”, Kluwer Academic Publishers, 1994
- R. Andraka, “A survey of CORDIC algorithms for FPGA based computers”, ACM, 1998
- E.B. Hogenauer, “An Economical Class of Digital Filters for Decimation and Interpolation”, IEEE Trans on ACSSP vol ASSP-29 no.2 April 1981
- A. Peled & B. Liu, “A New Hardware Realization of Digital Filters”, IEEE trans on ACSSP, vol. ASSP-22 no. 6, December 1974.

*ASSP and other transactions are a goldmine of hardware implementations*

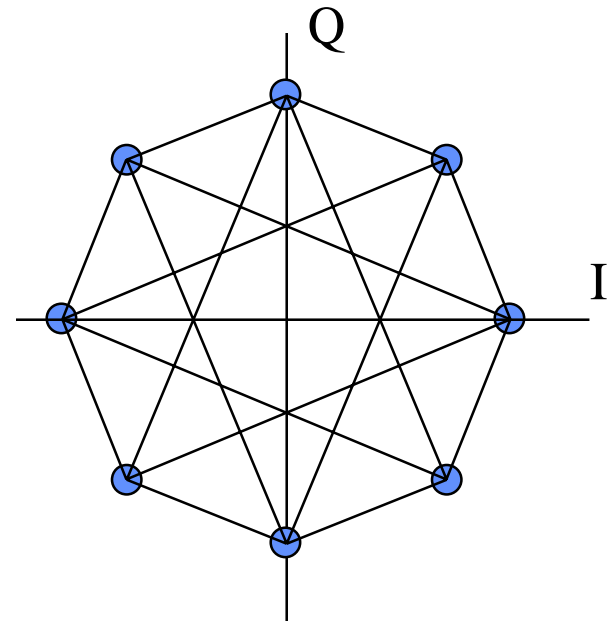


**Example:**

**North American  
TDMA Digital Cellular  
 $\pi/4$  DQPSK Modem**

# Specifications

- Differential phase encoding
  - $\pm\pi/4$  or  $\pm3\pi/4$  phase offset
- Non-coherent receiver
- 28.6 kbps



$\pi/4$  DQPSK Constellation



# Transmitter

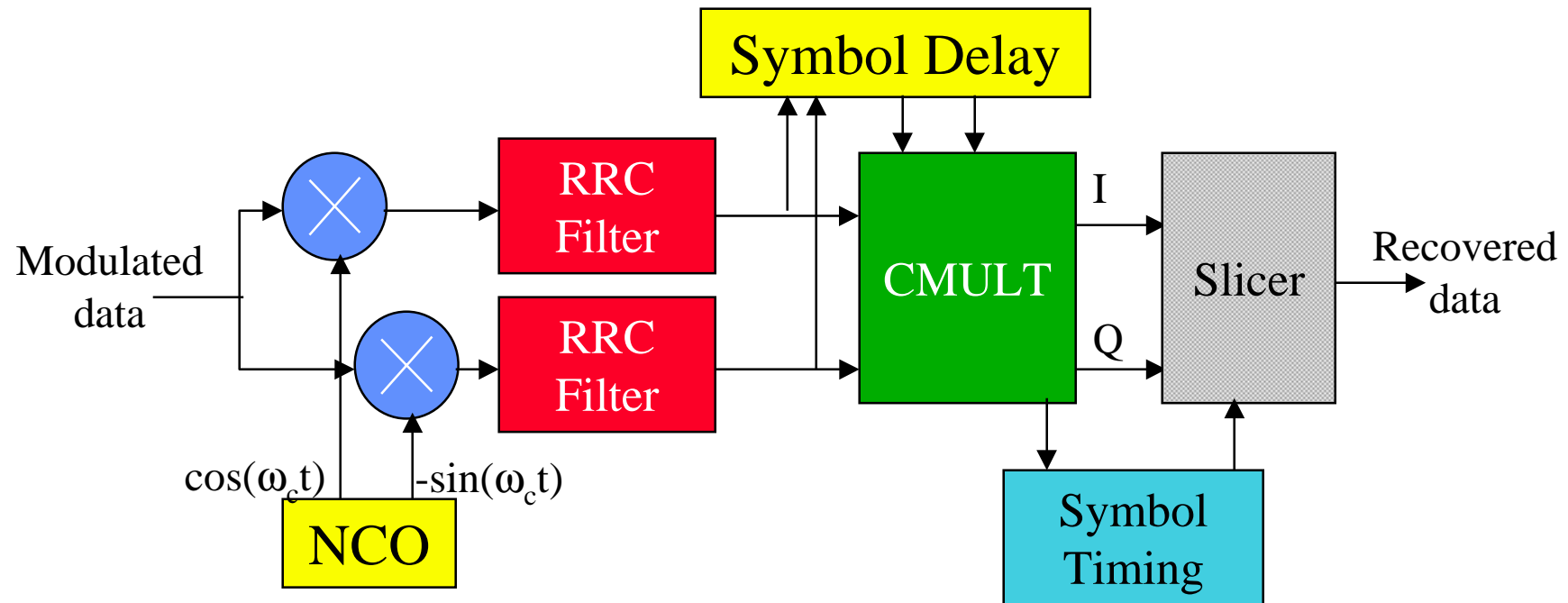
- **Differential coding**
- **Uses 2 bits per symbol**
- **I and Q take values  $0, \pm 1, \pm 1/\sqrt{2}$**
- **Filter interpolates to upsample**
- **Similar to QAM modulator example**



# Receiver

- **Non-coherent differential detection**
- **Digital demodulation from IF to baseband**
- **Equalizers left out of example**
- **Nyquist filter is RRC w/ 35% excess BW**
- **48.6 kbps (24.3 kbaud)**

# $\pi/4$ DQPSK Demodulator



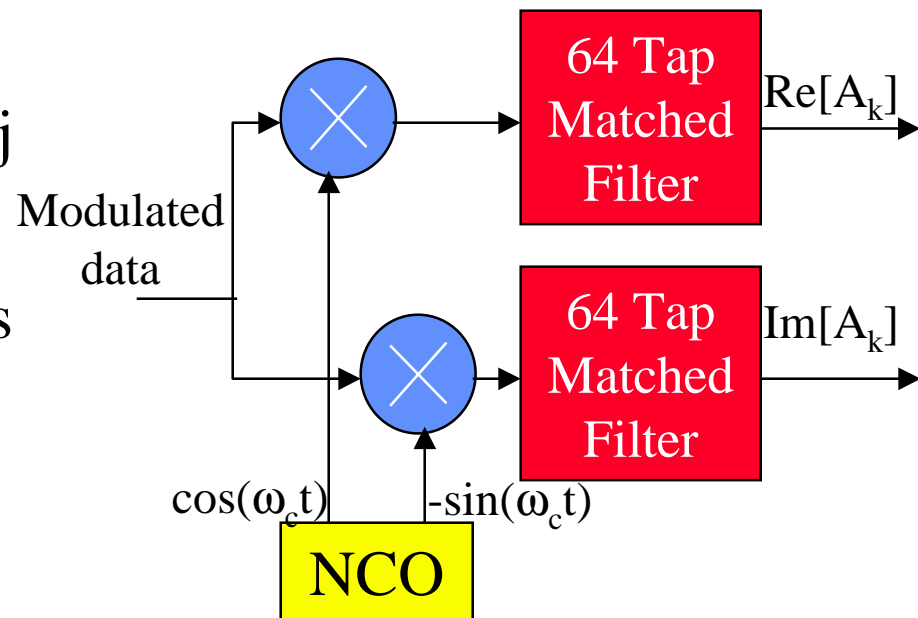


# Receiver

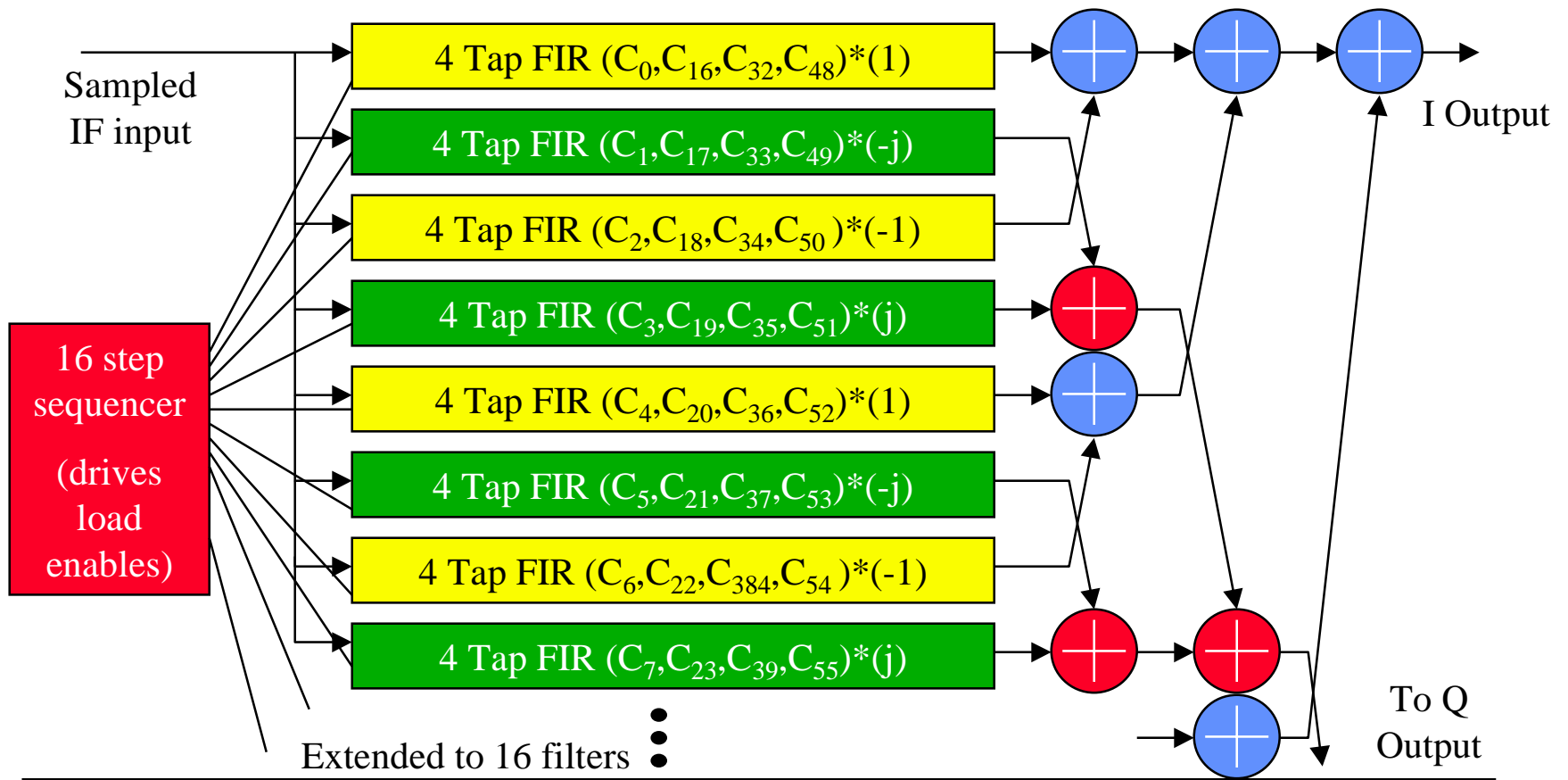
- **Sample baseband at 4x baud rate = 97.2Khz**
- **Bit serial detection logic**
  - 16 bit baseband I and Q = 16x bit clock
- **Sample IF using bit clock then decimate**
- **IF center frequency = bit clock/4 = 16\*B**
  - simplifies mixer

## Down-Converter and Filters

- DAC sampled at bit rate
- IF at bit rate/4
- NCO sequence is  $1, -j, -1, j$
- Decimate 16:1
- Decimating filter reduces to 16 parallel filters
- Use 4 tap filters
  - Net filter is 64 taps



# Reduced Down-Converter and Filters



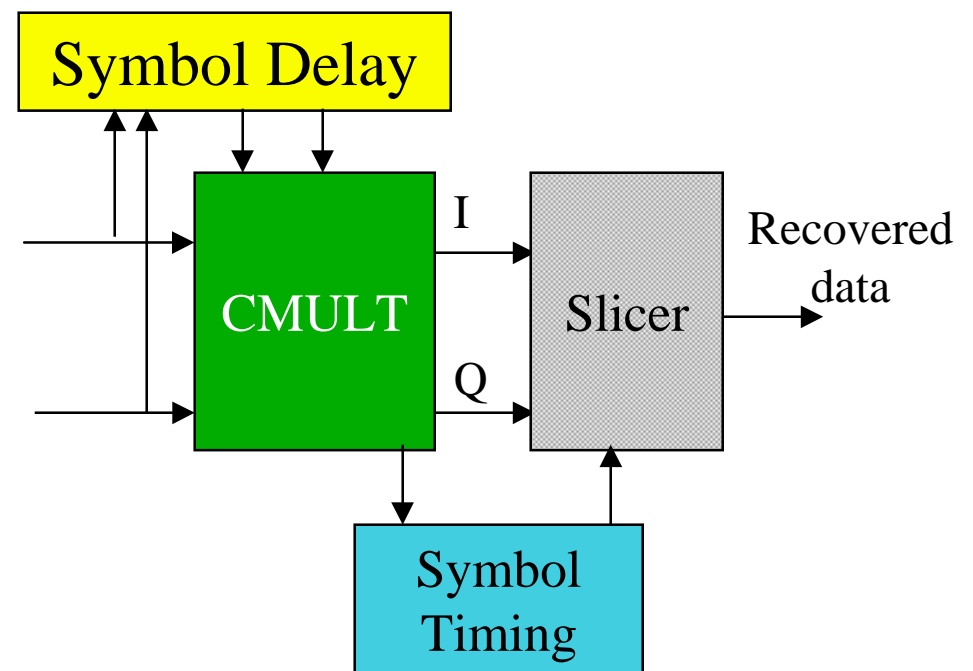


## Further Filter Reduction

- **Each 4 tap filter is 4 LUT and Scaling Acc**
- **Move SA to end of adder tree**
  - **Each filter is a 4 LUT with delay queue**
- **Mixer and 64 tap filter (I and Q) is 152 CLBs**
- **Filter bit rate is a low 1.55 MHz**
- **Present data LSB first**
  - **Allows serial LSB first output from filter**

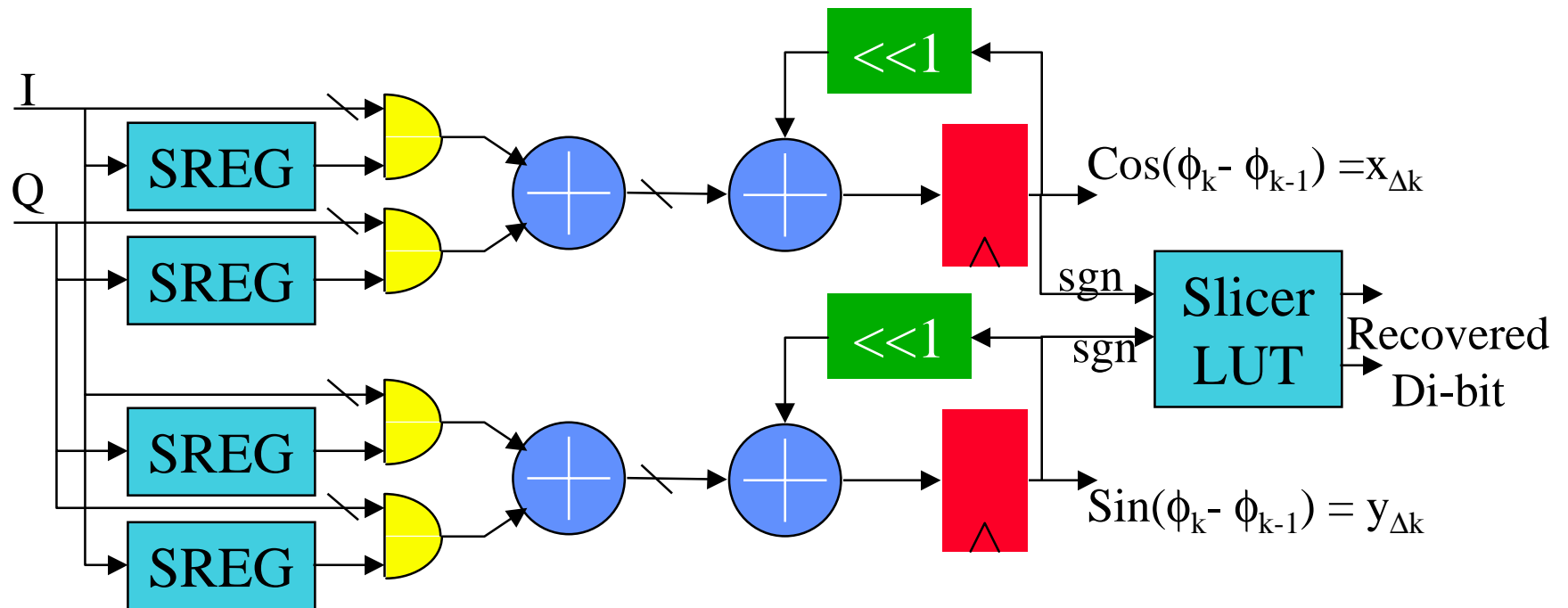
# Detector

- Differential Detection
- $x[k] \bullet x^*[k-1]$  yields  $\sin(\phi_k - \phi_{k-1}), \cos(\phi_k - \phi_{k-1})$
- Implement CMULT with scaling accumulators
- 4 Samples per symbol
  - Delay is 64 clocks fixed
- Symbol timing selects which sample to output





# Complex Multiply





# Symbol Timing

- **Error statistic controls state machine**
- $e = x_{\Delta k}^2 + y_{\Delta k}^2$ 
  - proportional to magnitude squared
  - Use larger + 1/2 smaller approx
- Compute for each sample
- Compare to previous sample
  - if difference is less than a threshold no change to sample point
  - otherwise if current larger, advance sample point
  - or if previous larger, retard sample point
- 29 CLBs



## Modem Implementation

- **1.55 MHz master clock (sloowww)**
- **Entire design in 3/4 XCS30-3 (or 4013E-4)**
- **Device cost under \$10**
- **Performance compares favorably with heavily loaded TMS320C50**
  - **TI DSP can only implement 20 Tap filters,**
  - **Tx and Rx not concurrent in TI DSP**
  - **TI DSP design in/out is complex baseband, not IF**
  - **TI DSP design can't handle equalizer or viterbi decoder**