

How to Use Programmable Logic to Build Fast and Efficient DSP Functions

XUP Workshop Appendix E

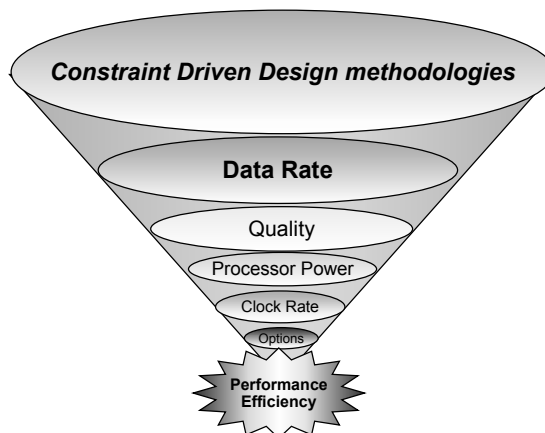
Originally created by: Greg Goslin
Xilinx, Corporate Applications

Using Programmable Logic to Accelerate DSP Functions



Constraint Driven Design Methodology

- **Constraints**
 - System Requirements
 - Hardware Limitations
- **Data Rate**
 - Inputs
 - Outputs
 - Multi-Channel I/O
- **Quality**
 - Number of Bits/Taps
 - Number of Operations
 - Error Tolerance
- **Processor Power**
- **Clock Rate**



Using Programmable Logic to Accelerate DSP Functions



Building Fast and Efficient Filters in FPGAs

■ Efficient Filter Algorithms for FPGAs

- Distributed Arithmetic:

- Bit-Serial
- n-Bit Parallel

■ Using Distributed Arithmetic for Filter Designs

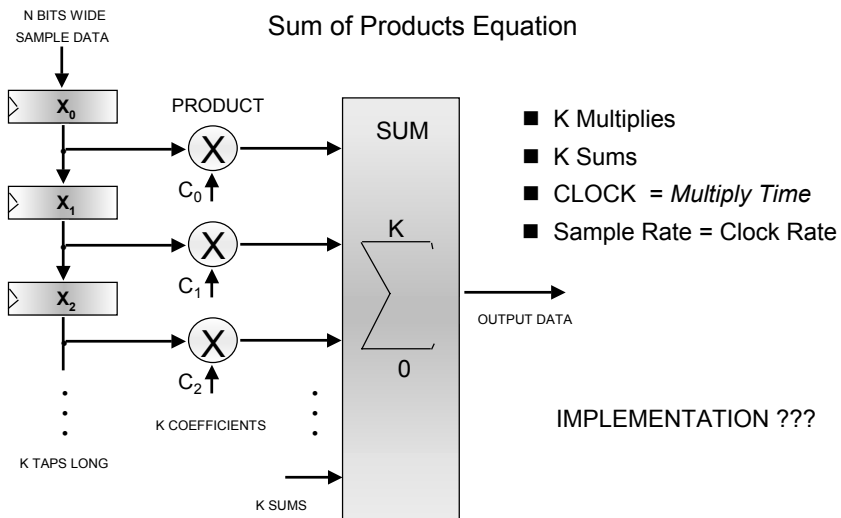
- Serial FIR Filter Example
- Two-Bit Parallel FIR Example
- Full Parallel FIR Example

Using Programmable Logic to Accelerate DSP Functions



FIR FILTER EXAMPLE

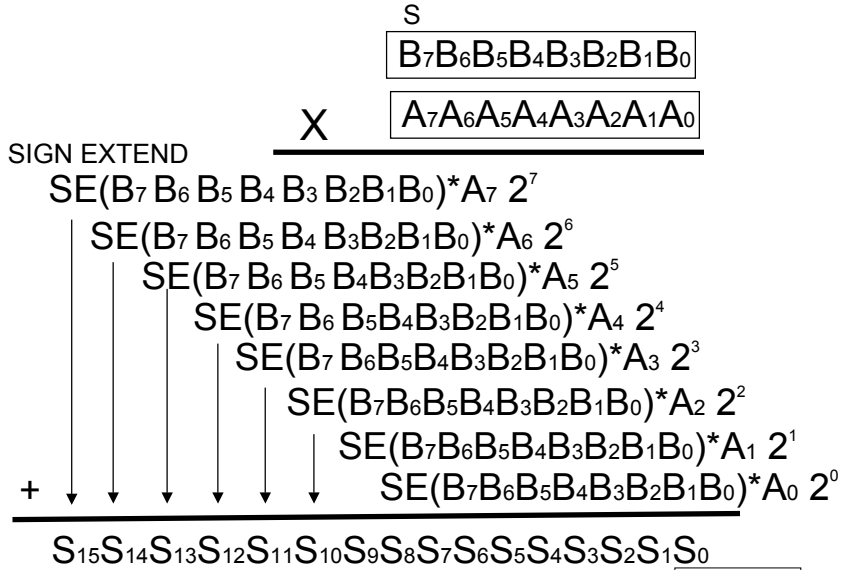
Sum of Products Equation



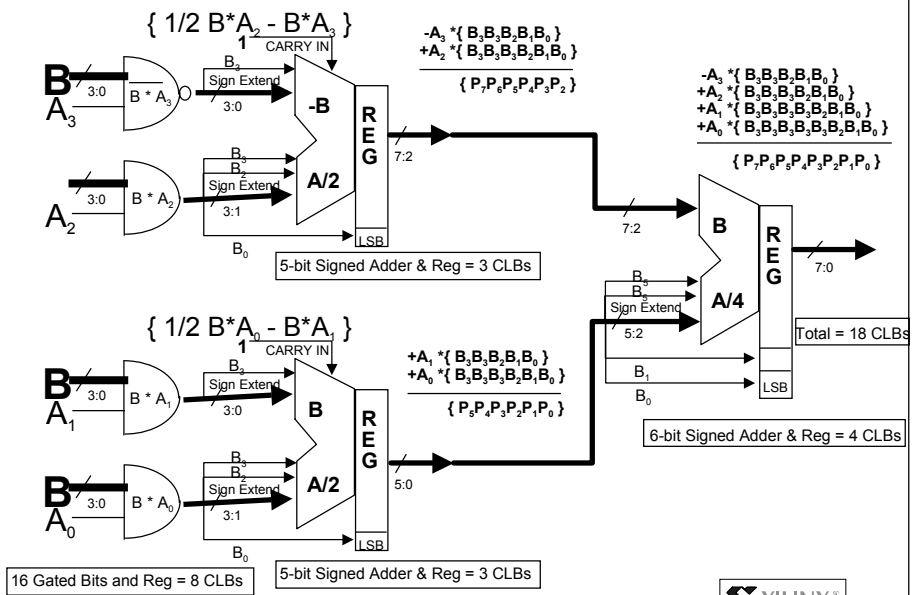
Using Programmable Logic to Accelerate DSP Functions



8-Bit X 8-Bit Signed Multiply

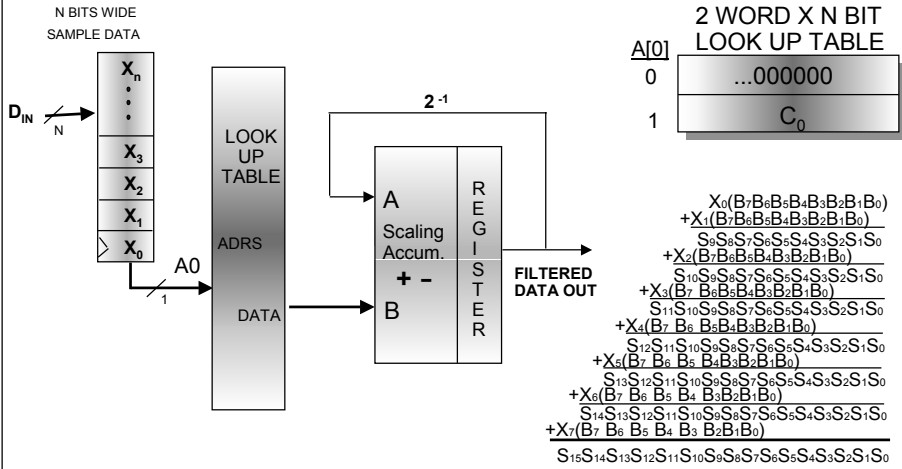


4-Bit Signed Tree Multiplier

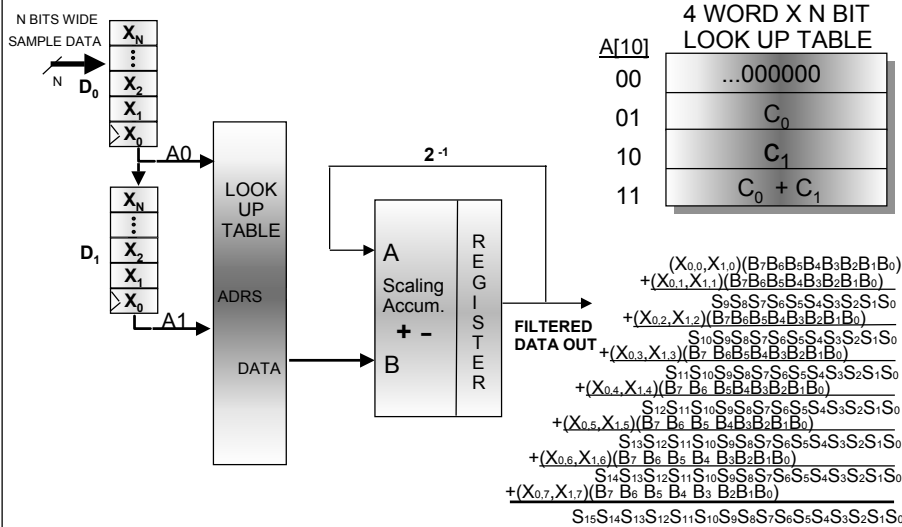


D.A. ONE TAP FIR FILTER = $D_0 C_0$

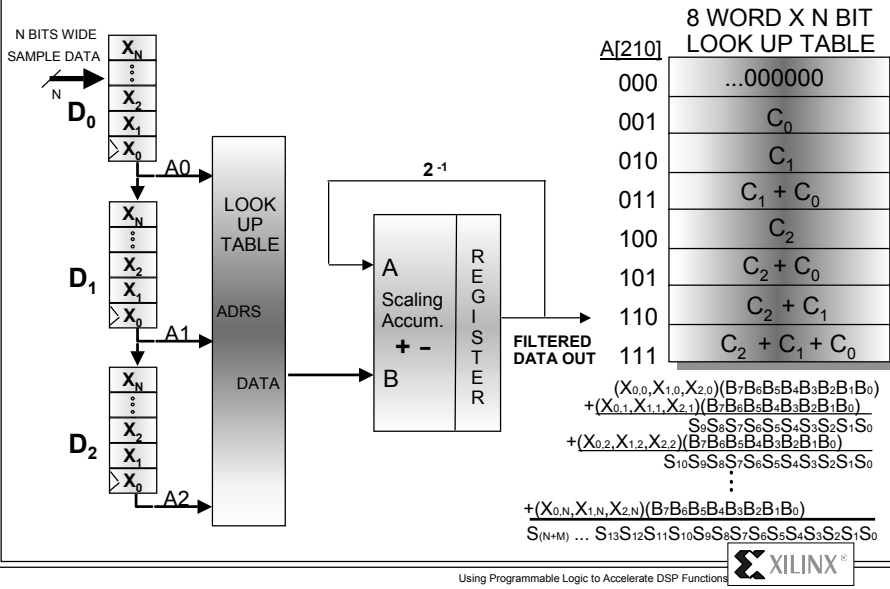
REDUCES TO MULTIPLYING A VARIABLE TIMES A CONSTANT



D.A. TWO TAP FIR FILTER = $D_0 C_0 + D_1 C_1$

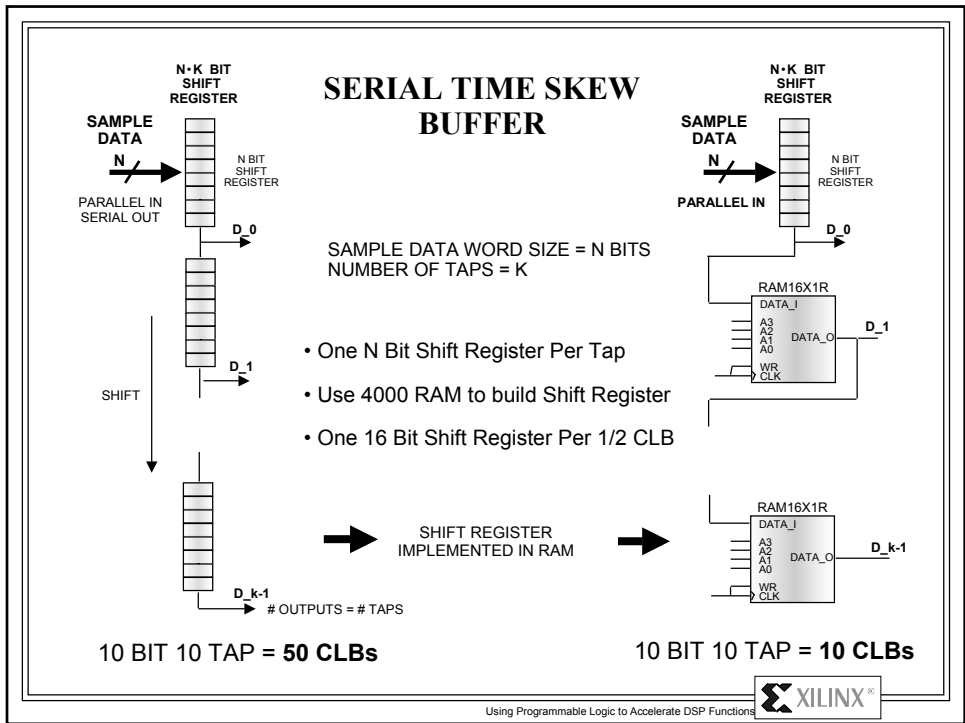
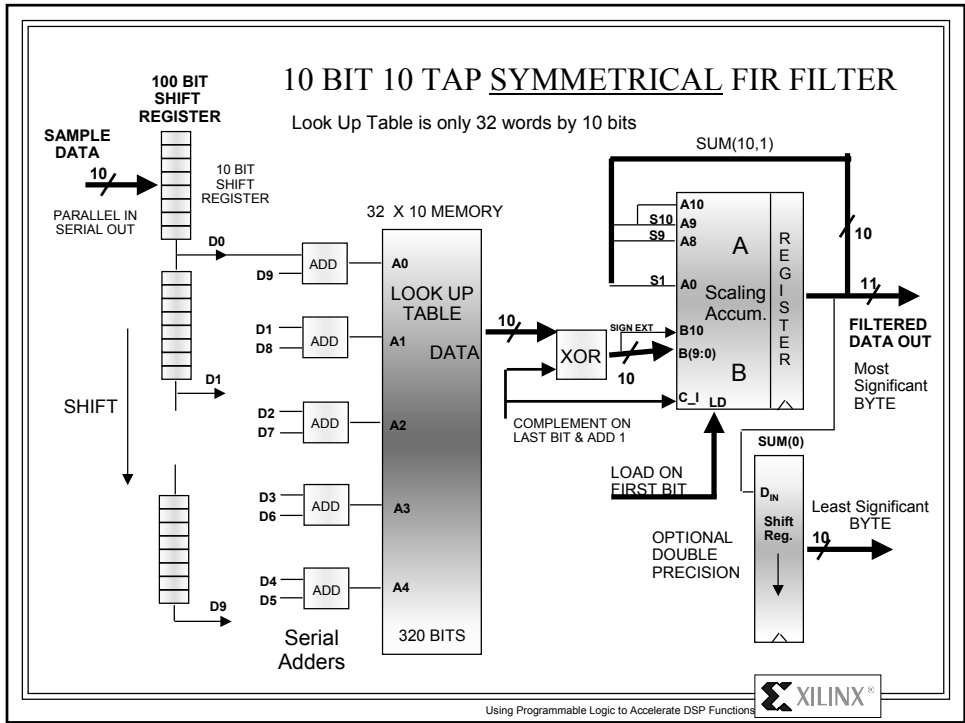


D.A. THREE TAP FIR FILTER

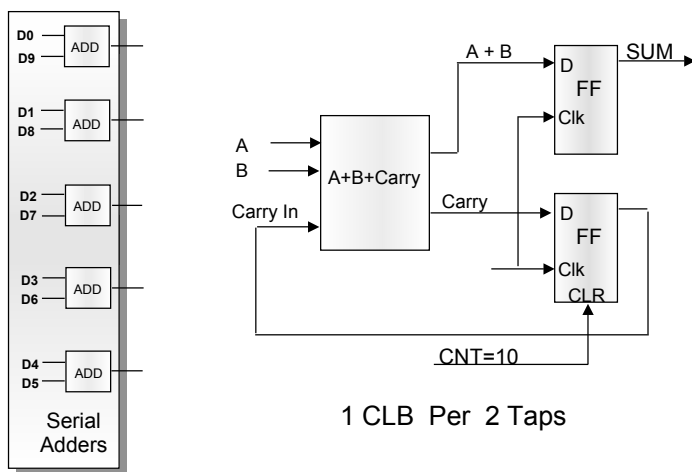


The Development of a Distributed Arithmetic FIR Filter

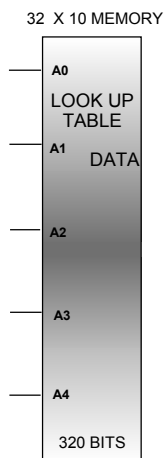
10-Bit 10-Tap - XC4000 Family Example



Serial Adder

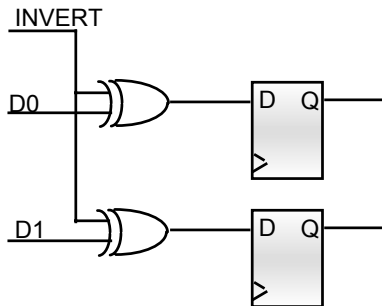


DISTRIBUTED ARITHMETIC LOOK-UP TABLE



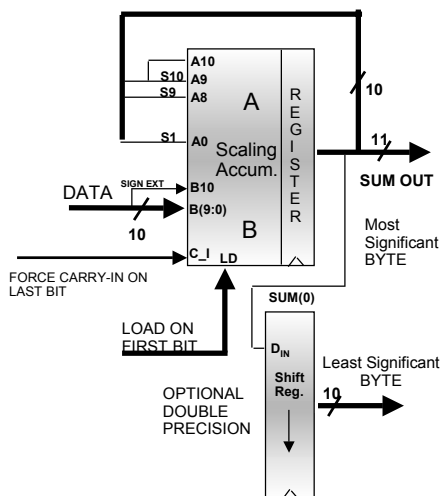
- HOLDS ALL PARTIAL PRODUCTS
- LUT IS AS WIDE AS COEFF
- CAN USE MEMGEN TO BUILD LUT

1's COMPLEMENTER

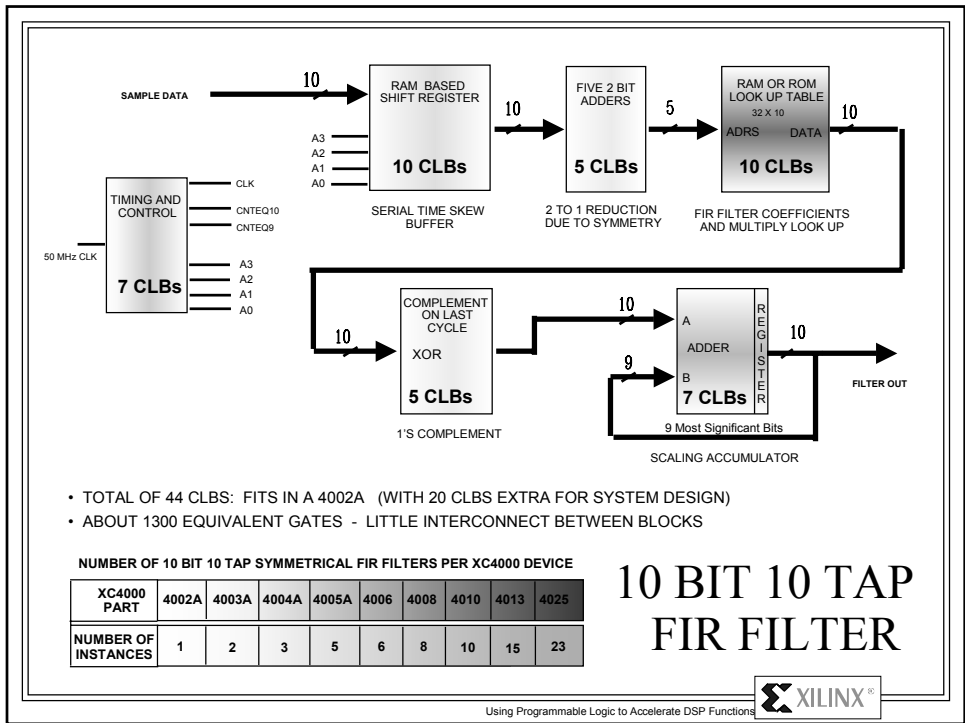
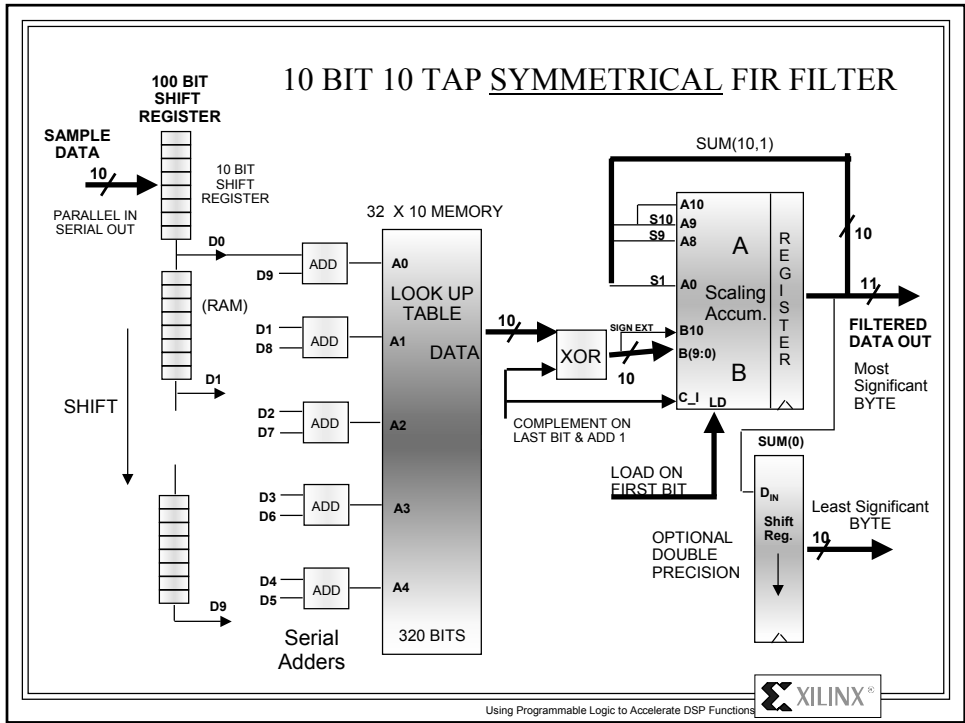


- INVERTS DATA ON LAST CYCLE
- 2 BITS PER CLB

SCALING ACCUMULATOR



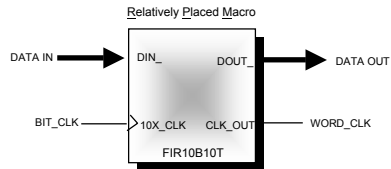
- ADDS DATA TO $(1/2) * (\text{SUMOUT})$
- 2 BITS PER CLB
- NEED N+1 BITS
- DOUBLE PRECISION WITH SR
- CAN USE XBLOX FOR RPM



PERFORMANCE

- FIR10B10T MACRO CAN BE CLOCKED AT 66 MHZ @XC4000E-3
- 10 BIT WORD REQUIRES 11 CLOCKS
- 10 BIT SAMPLE WORD RATE IS 6 MHZ
- 8 BIT WORD REQUIRES 9 CLOCKS, ETC
- 8 BIT SAMPLE WORD RATE IS 8 MHZ

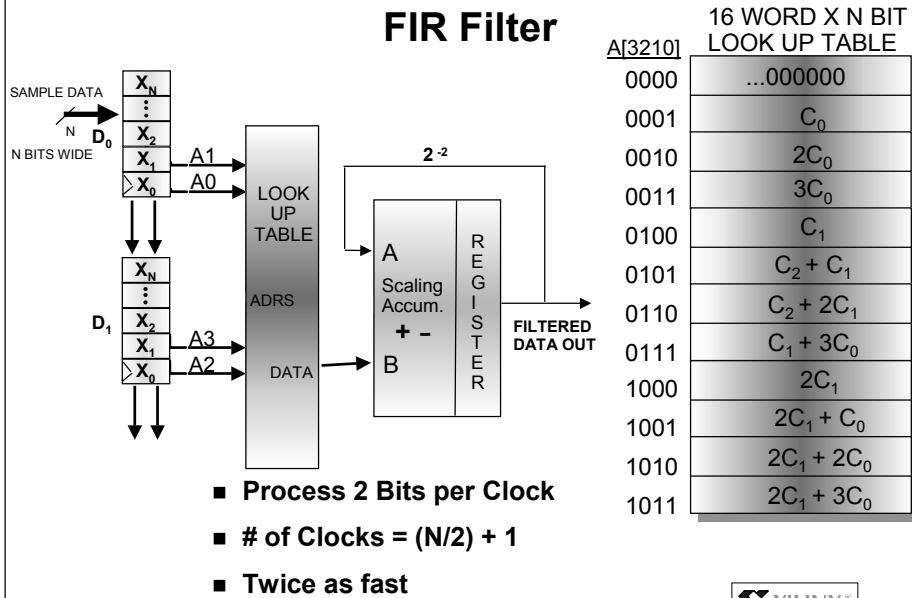
FIR Filter Macro



WORD SIZE	6	8	10	12	14	16	BITS
SAMPLE RATE	11.1	7.4	6.1	5.1	4.4	3.9	MSPS

Double-Rate DA FIR Filters

Two Bit Parallel Distributed Arithmetic FIR Filter



Double Sample Rate D.A. FIR Filters

- Twice the I/O Data Sample Rate
- Two Taps Requires 4 Input LUT without Symmetry
- Four Taps Requires 4 Input LUT with Symmetrical FIR
- Time Skew Buffer uses Twice as many CLBs
- LUTs are the same, if equal bit weights are used to address the LUTs.
- 2-Bit PDA Performance, Clocked at 66 MHz @XC4000E-3

WORD SIZE	6	8	10	12	14	16	BITS
SAMPLE RATE	22.2	14.8	12.2	10.2	8.8	7.8	MSPS (Double Precision)

Full Parallel D.A. FIR Filters

- One 8-Bit Tap Requires two 4 Input LUTs and an ADDER with an offset for bit weighting.
- Time Skew Buffer must use REGs
- Maximum I/O Data Sample Rate
- Full PDA Performance, in a XC4000E-3/-2, 50-70 MHz.
 - Pipelining can further increase sample rate
- LUTs are the same, if equal bit weights are used to address the 4-Coefficients in the LUT.

WORD SIZE	6	8	10	12	14	16	BITS
SAMPLE RATE	70	70	70	70	66	66	MSPS (Double Precision)

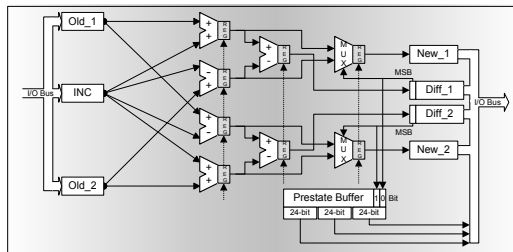


Using Programmable Logic to Accelerate DSP Functions

FPGA-Based DSP Coprocessor Design Implementation

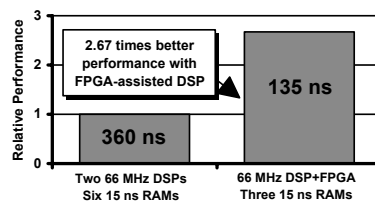
Performance

- Programmable DSP (DSP56300)
 - 24 clock cycles
 - 360 nsec @ 66 MHz
- FPGA-Based Coprocessor
 - 9 clock cycles
 - 135 nsec @ 66 MHz

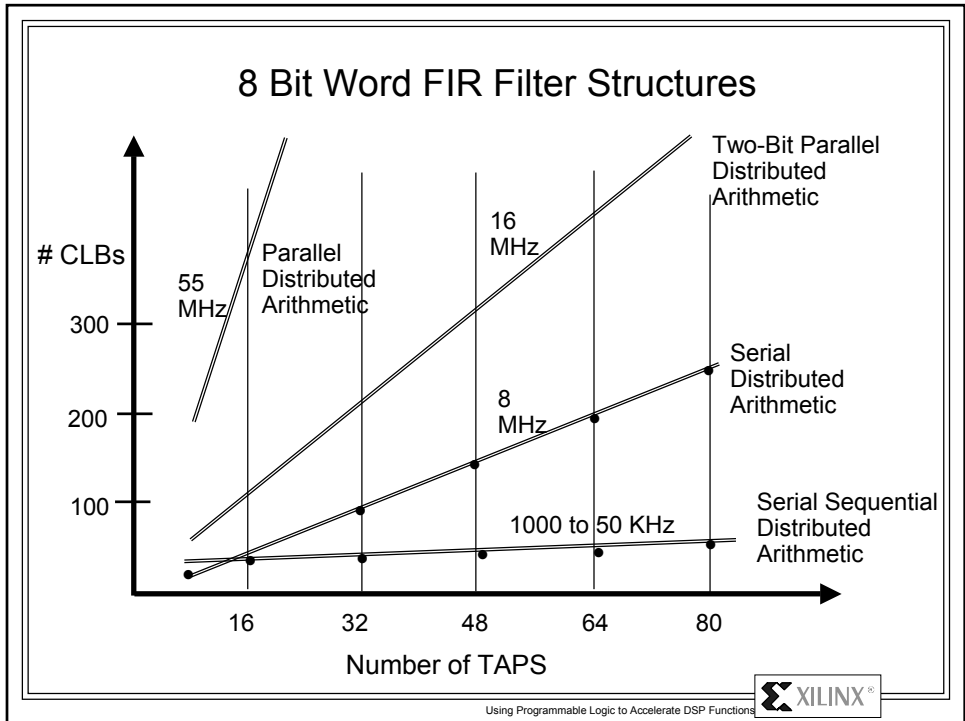


Results:

- 37.5% of original processing time
- 2.67X Increase in throughput
- System Requirements:
 - Before: 4-DSPs, 12-RAMs
 - After: 2-DSPs, 6-RAMs, 1-XC4013E



Using Programmable Logic to Accelerate DSP Functions



FIR Filter Implementation Options

8 Bit Word Example

	Serial* Sequential	Serial* Distributed Arithmetic	Parallel* Distributed Arithmetic
8 Taps	36 CLBs 1.08 MHz	44 CLBs 8.1 MHz	250 CLBs 60 MHz
16 Taps	36 CLBs 0.46 MHz	70 CLBs 8.1 MHz	400 CLBs 55 MHz
32 Taps	44 CLBs 0.23 MHz	122 CLBs 8.1 MHz	
48 Taps	62 CLBs 0.15 MHz	178 CLBs 8.1 MHz	
64 Taps	70 CLBs 0.11 MHz	228 CLBs 8.1 MHz	

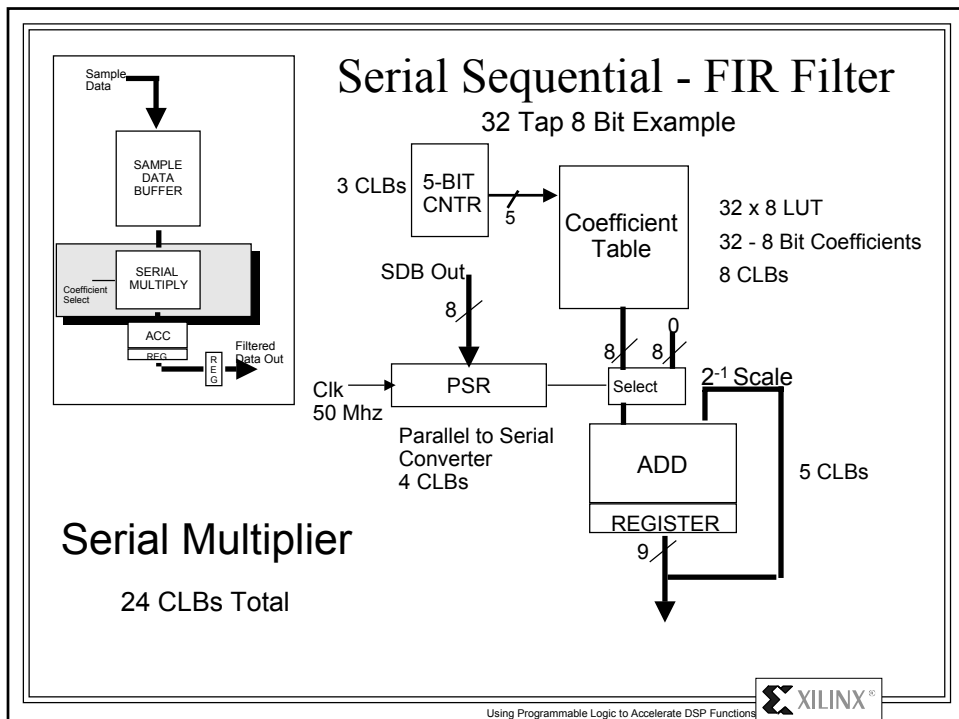
* Note: These designs are NOT Pipelined

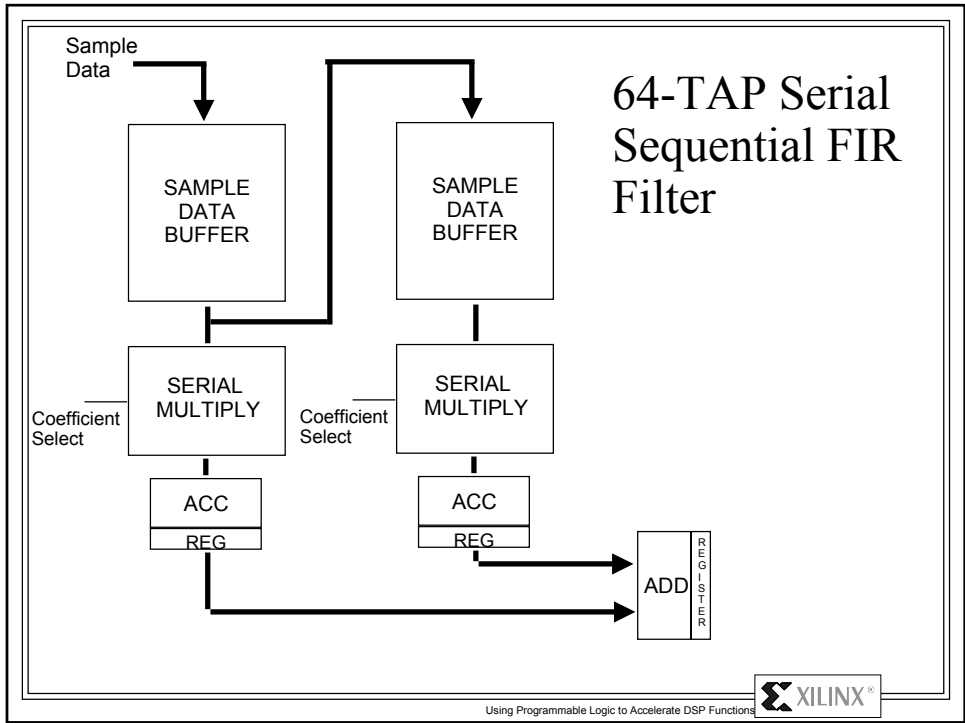
Using Programmable Logic to Accelerate DSP Functions

Lower Sample Rate Applications:

- Efficient CLB Counts
- Large Number of TAPs
- Moderate Sample Rates
- Non Symmetrical FIR OK

Serial Sequential Architecture





Serial Sequential - FIR Filter

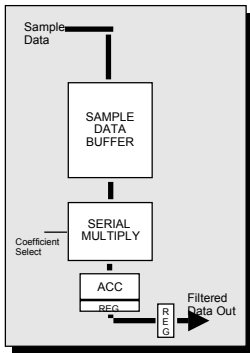
Number CLBs vs. Taps / Word Size

	8 Bit	10 Bit	12 Bit	14 Bit	16 Bit
8 Tap	36	43	50	57	64
16 Tap	36	43	50	57	64
32 Tap	44	53	62	71	80
48 Tap	62	77	92	107	122
64 Tap	70	85	100	115	130
80 Tap	97	115	133	151	169
96 Tap	97	115	133	151	169
128 Tap	112	137	162	187	212

- 4002 = 64 CLBs
- 4005 = 196 CLBs
- 4013 = 576 CLBs
- 4025 = 1024 CLBs

Using Programmable Logic to Accelerate DSP Functions

Serial Sequential - FIR Filter



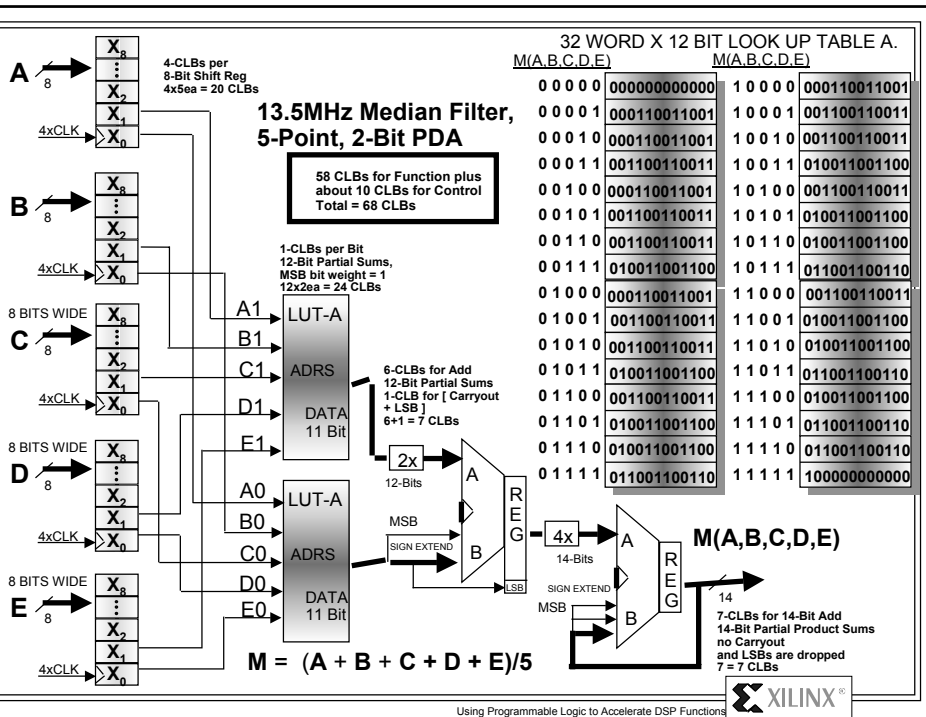
Maximum Sample Rate / Word Size

TAPS	8 Bit	10 Bit	16 Bit
8 Tap	781Khz	625Khz	390Khz
16 Tap	390Khz	312Khz	195Khz
32 Tap	195Khz	156Khz	97Khz
48 Tap	130Khz	104Khz	65Khz
64 Tap	97Khz	78Khz	48Khz
80 Tap	78Khz	62Khz	39Khz
96 Tap	65Khz	52Khz	32Khz
128 Tap	48Khz	39Khz	24Khz

- Serial Mult. Limitations
- Can Use Multiple 16 Tap Building Blocks
- 8X Faster at 128 Taps



Using Programmable Logic to Accelerate DSP Functions



Using Programmable Logic to Accelerate DSP Functions

Design the following Application:

Equations:

- $Y(R,G,B) = 0.299*R + 0.587*G + 0.114*B$
 - $U(R,G,B) = -0.169*R - 0.331*G + 0.500*B$
 - $V(R,G,B) = 0.500*R - 0.419*G - 0.081*B$

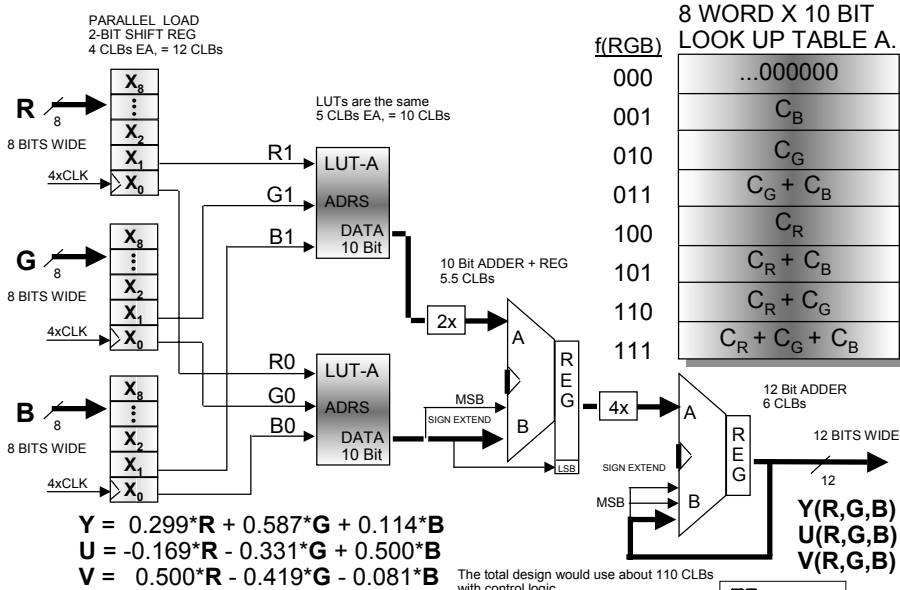
 - R, G, B Data is 8-Bits at 13.5 MHz. The circuit already has a 2x Clk (27 MHz).

 - Draw a functional schematic diagram of the circuit.
- How do you implement the three multipliers or MACs?
 What is the estimated size of the final design?
 What is the estimated speed of the final design?
 How long would it take to turn over this design?



Using Programmable Logic to Accelerate DSP Functions

Video Coding Application with 4x Clock



Using Programmable Logic to Accelerate DSP Functions

Video Coding Application with 2x Clock

