High Performance Communications Using FPGAs

Dr Chris Dick
Xilinx
IP Development Group
chrisd@xilinx.com

XILINX®

Presentation Outline

- Software radio architecture and justification already covered
  - Significant DSP challenges highlighted
- Why FPGAs for DSP?
- Performance through parallelism
- Some examples
- System design using FPGAs
- What about power?
- EDA and system level tools
- New FPGA architecture - *Virtex* - for DSP

Software Radios - Approaches

- *Speakeasy* I - Multiprocessor using
  - quad TMS320C40 MCMs
  - Application specific processors (ASP)
- Speakeasy II
  - ‘C44s
  - CHAMPS (FPGA)
- MIT *SpectrumWare* - workstation hardware
  - processing done in user space
- Berkeley - Infopad
- Comparatively recently: FPGAs

Software Programmable DSPs

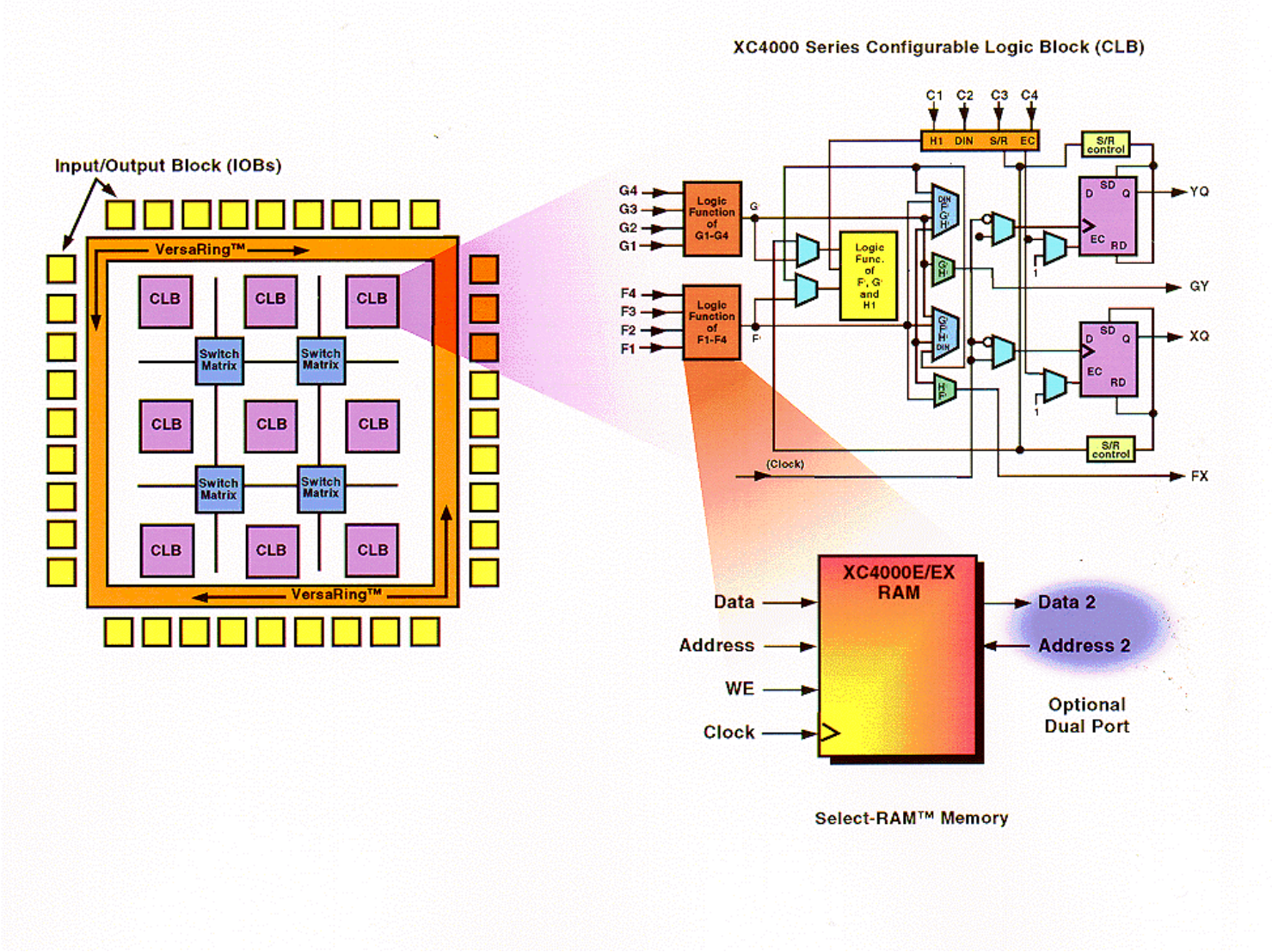
- Datapaths optimized for performing common DSP tasks
  - single-cycle multiply-accumulate (MAC)
  - bit-reversed addressing for Cooley-Tukey FFT
- Advantage
  - software programmable $\Rightarrow$ no silicon NRE
- Disadvantage
  - performance sacrificed

Software Programmable DSPs

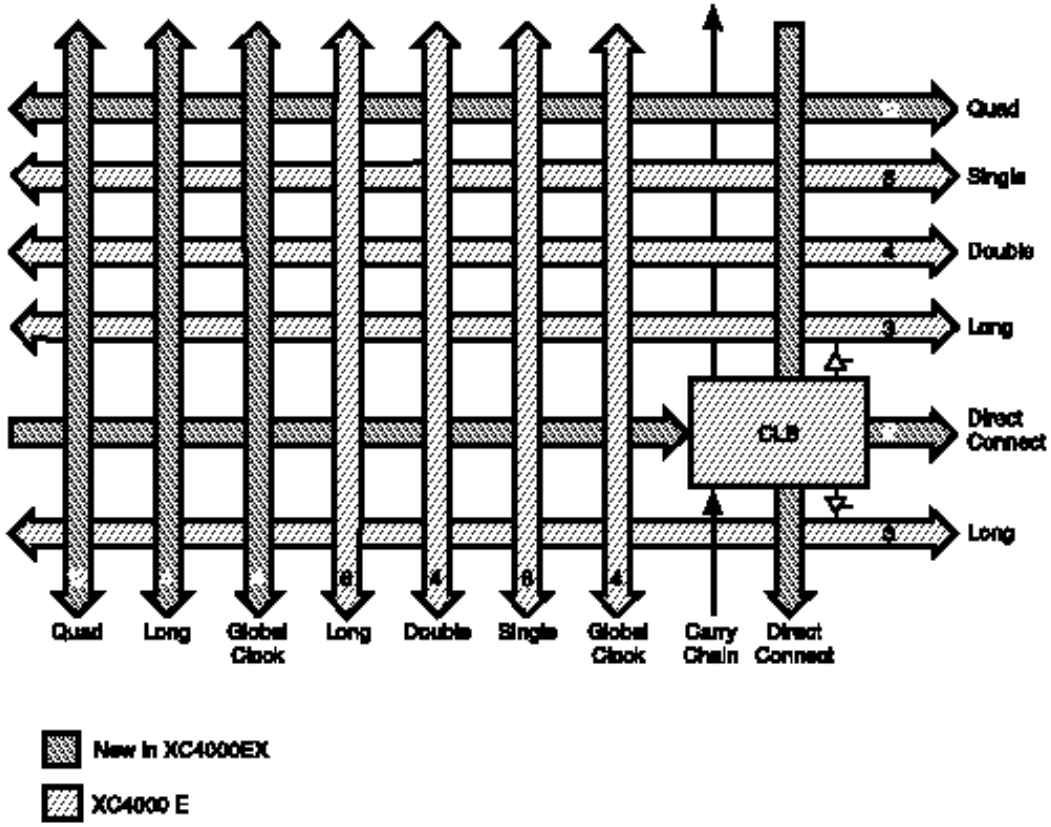
- DSP Instruction set defined by device architects
- Application performance depends on

  1. match between computations required in application and type/number/interconnectivity of microprocessor functional units
  2. how well your algorithm maps onto the instruction set
  3. compiler technology
- Mismatch between computational requirements and processor architecture $\Rightarrow$ WASTED RESOURCES

XC4000 FPGA Architecture



XC4000X Interconnect Hierarchy



FPGA DSP - *It's about time*

- FPGAs are like miniature *silicon foundries* with extremely short turn-around times
- FPGAs open-up the solution space for digital signal processing engineers by decoupling the design process from the preconceptions of the VLSI DSP chip designer
  - Word length - different word lengths are appropriate for different problems or different parts of the same problem
  - Functional unit type and number (parallelism)
  - Functional unit connectivity
  - Memory $\Rightarrow$ depth, width and bandwidth

FPGA DSP

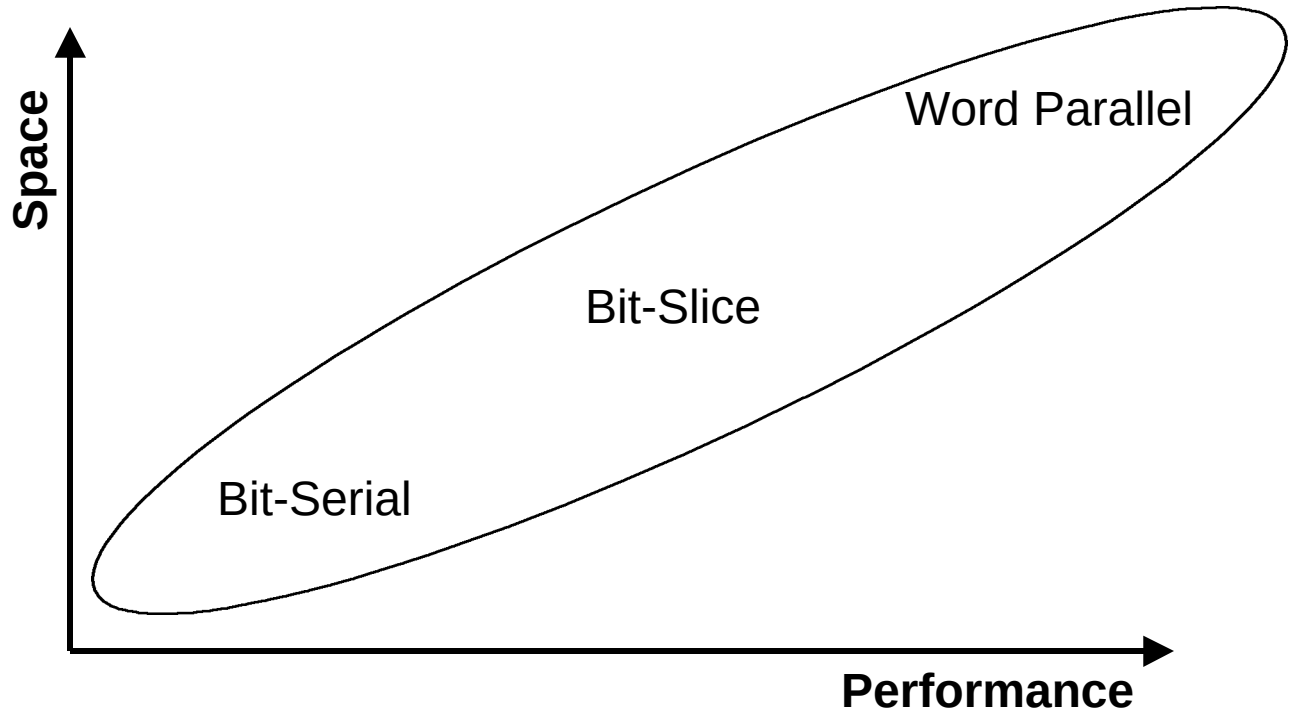
- What digital radio functions can FPGAs be used for?
  - channelizing
  - adaptive equalizer
  - modulation/demodulation
  - trellis encoder/ Viterbi decoder
  - interleaving/deinterleaving
  - Reed-Solomon encoding and decoding
  - encryption/decryption

Why FPGA DSP?

- Using FPGAs brings an entirely new dimension to DSP hardware
- FPGA DSP hardware is like *liquid hardware*
  - functionality defined by downloading configuration bitstream that defines datapath
- Reconfigurable computing technology $\Rightarrow$ *Reconfigurable Receiver*
- ONE PIECE OF HARDWARE CAN BE USED FOR MANY APPLICATIONS

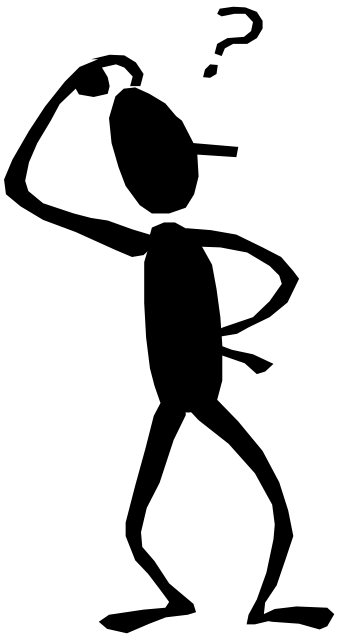
FPGAs Put the Silicon Back in Your Hands

- Designer can make space-time tradeoffs based on system requirements



What do you do when...

the fastest DSP Processor Is Not Fast Enough?



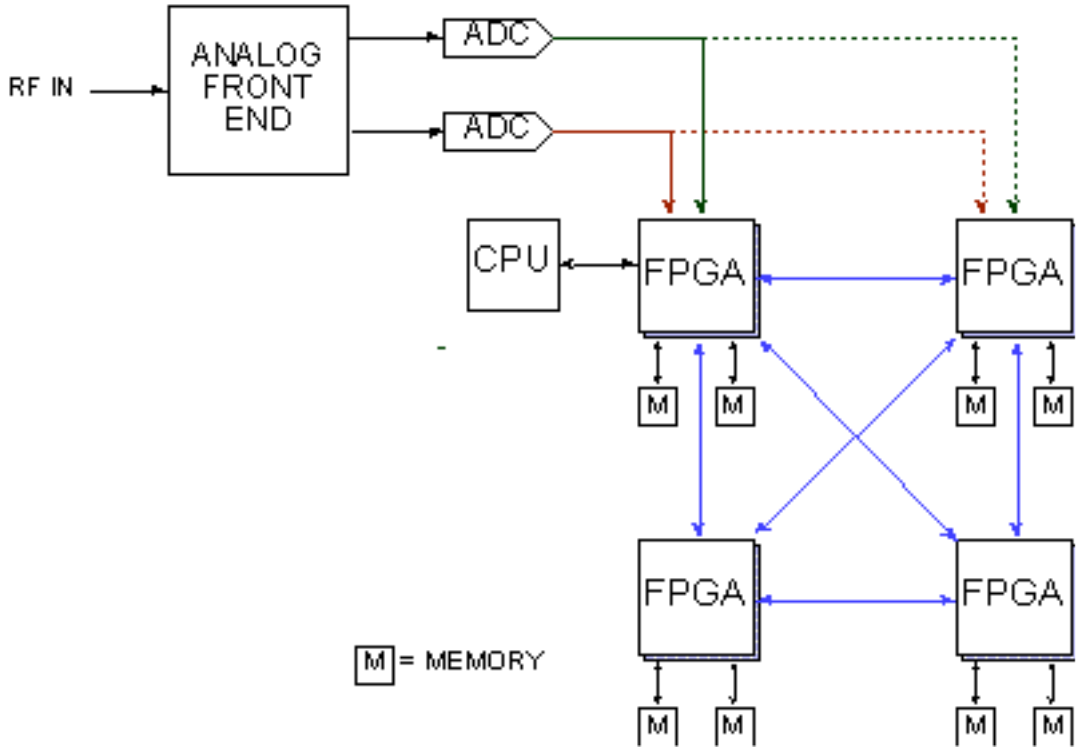
- Design a custom gate array?
- Add more DSP processors?
  - FPGA-based DSP system
  - CPU and FPGA

FPGA DSP System

- FPGA DSP engine consisting of FPGA device(s) only

OR

- FPGA reconfigurable co-processor

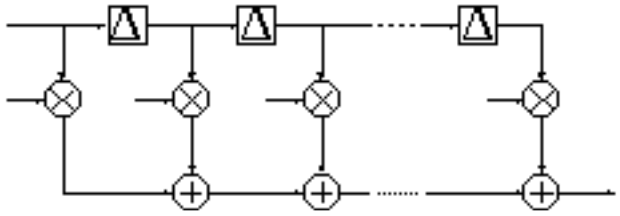


Some Examples

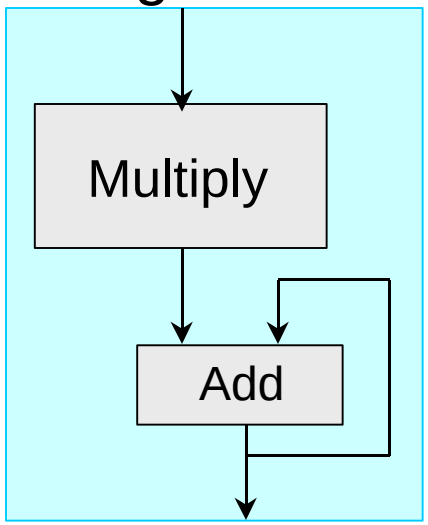
- FIR Filters
- FFT
- Digital receiver
- Adaptive filter

FIR Filters

- Conventional approach using DSP processors



Single MAC

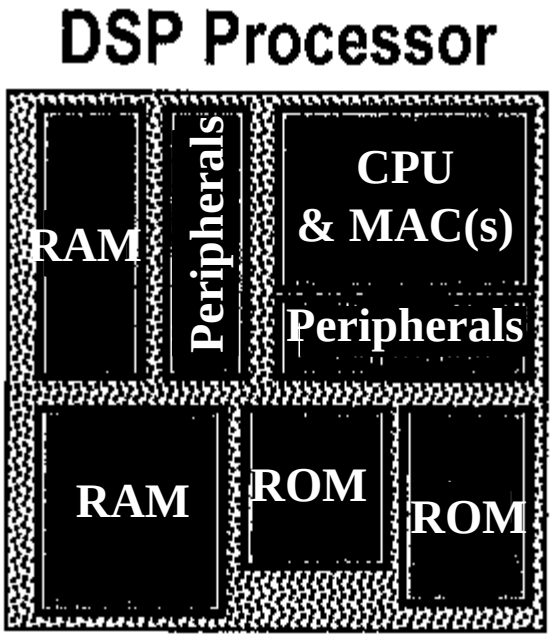


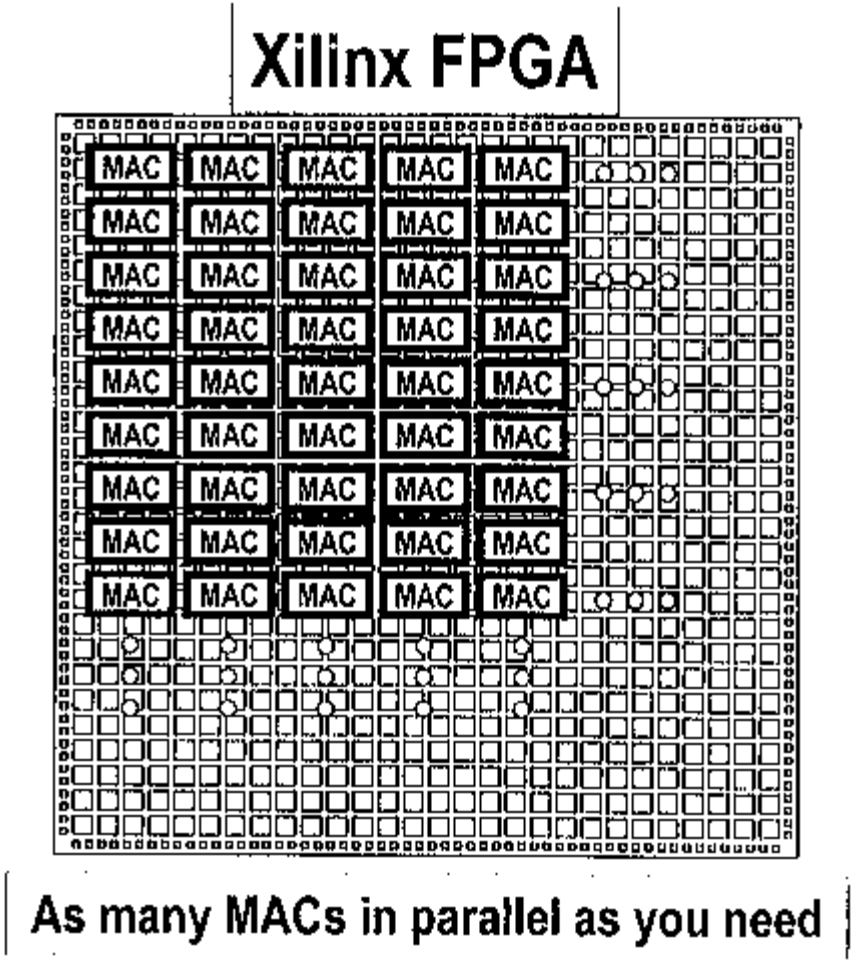
Sequential Processing

+ Programmable
+ Off-the-shelf, standard part
+ Hardware multiplier

– One MAC (Multiply Accumulate)
– Time-Shared
– Performance ceiling

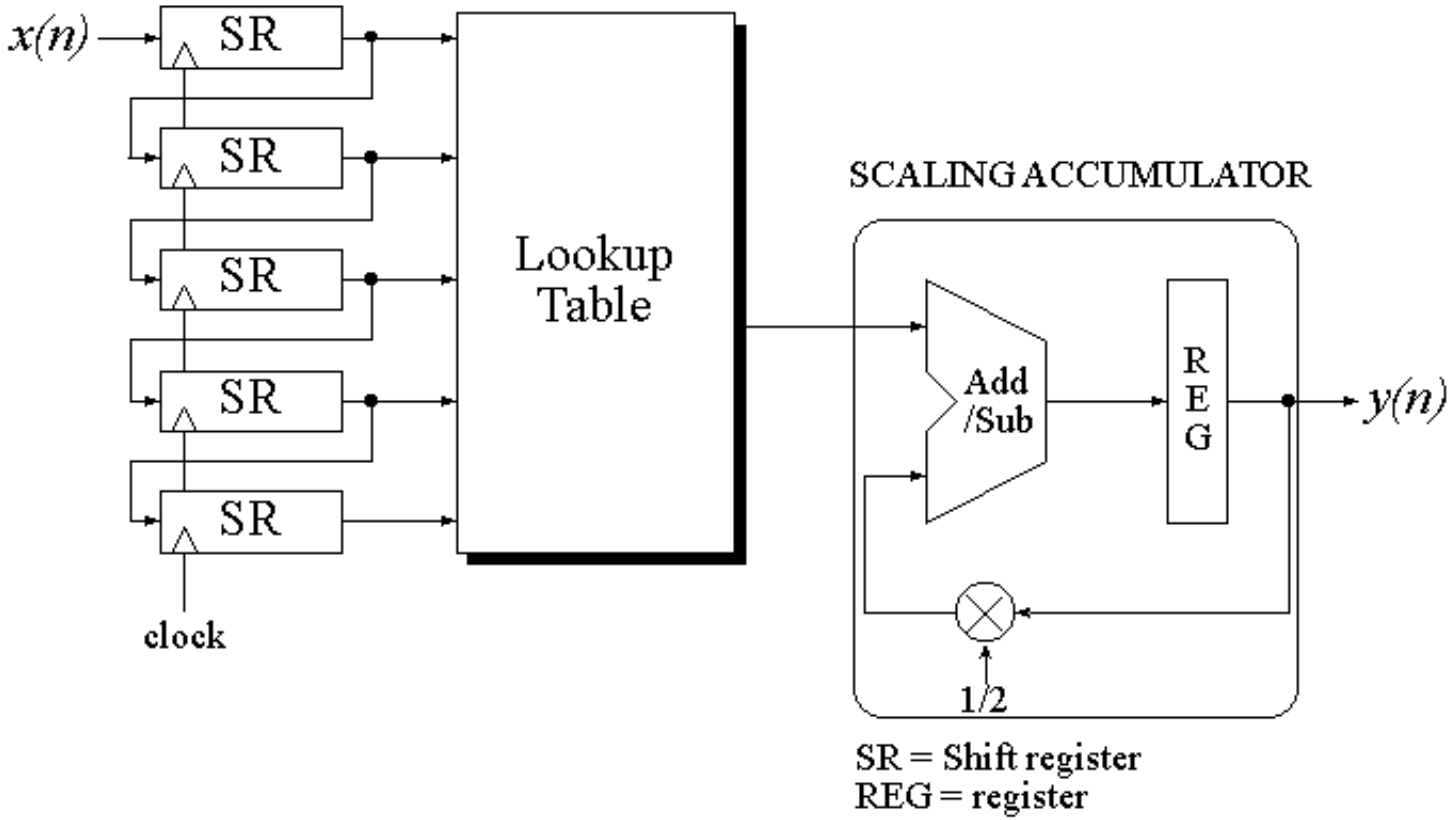
Performance Through Parallel Processing





FIR Filters Using *Distributed Arithmetic (DA)*

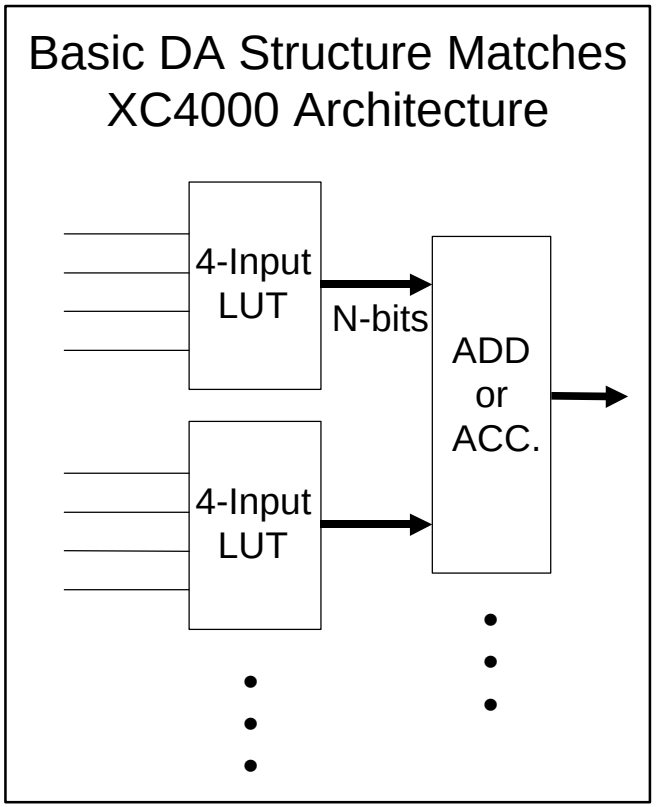
- DA Filter mechanization



Distributed RAM & Distributed Arithmetic: Perfect Match

DA Algorithms:

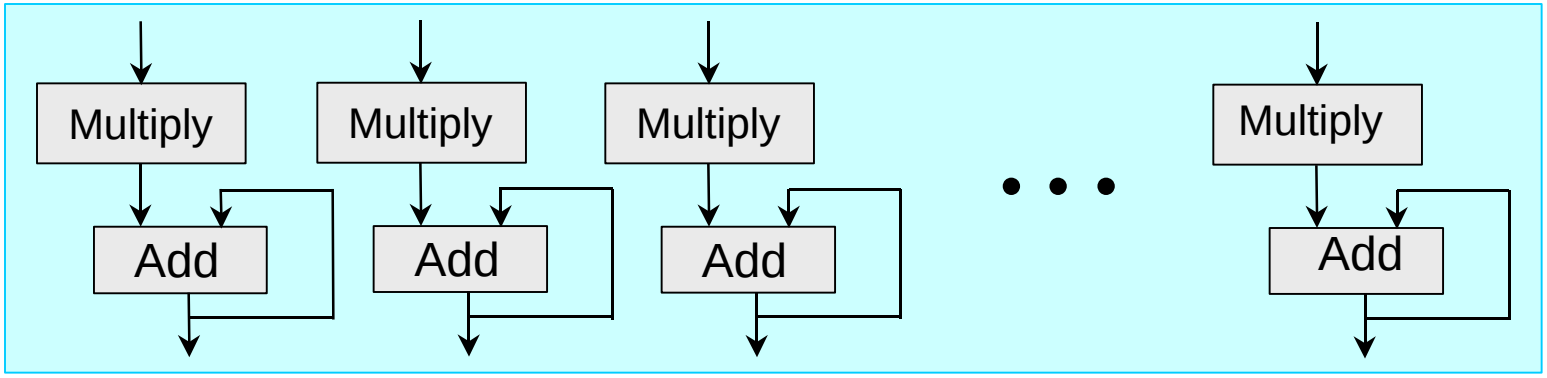
- 4-Input Look-Up-Tables (LUT)
- For higher performance
  Use more LUTs
  = more parallelism
- Efficiency similar to custom solution
  Achievable with LUT logic
  More ASIC gate equivalents
  More cost effective



Xilinx DSP

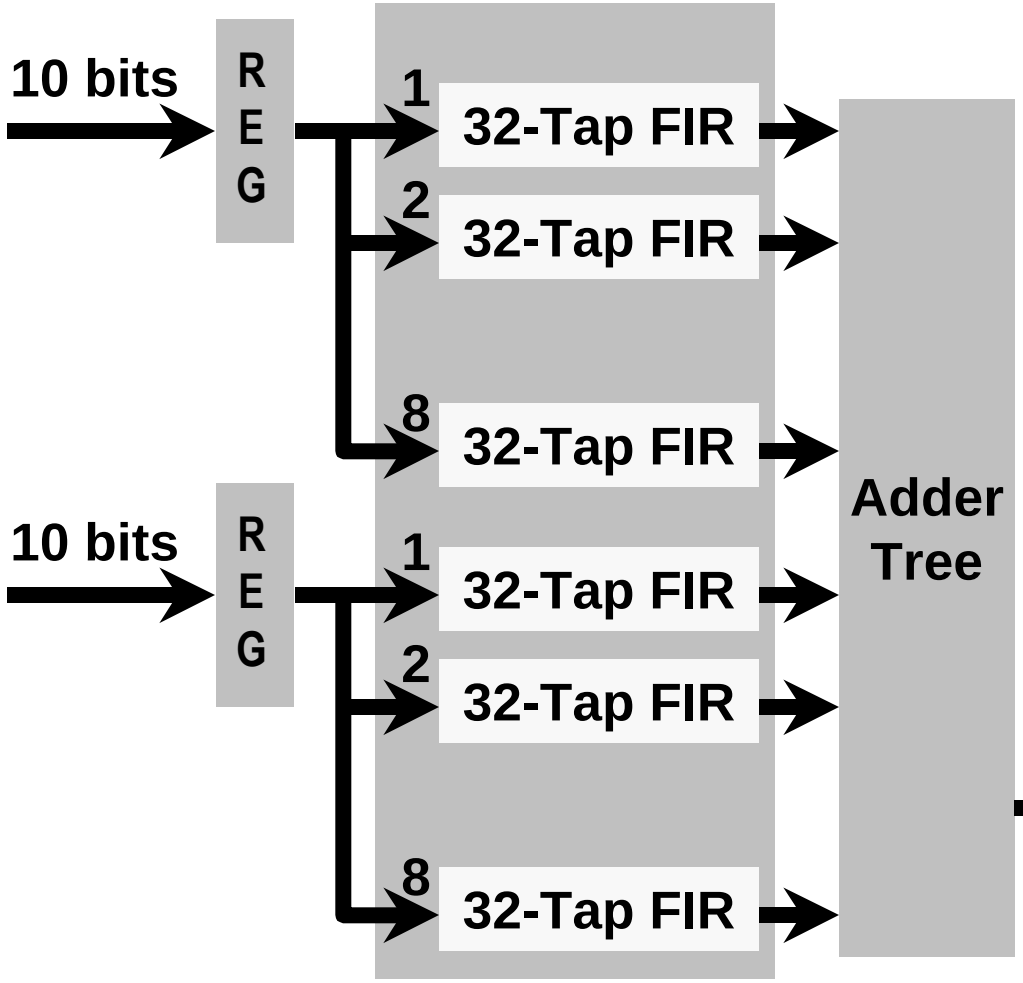
High Performance Alternative - Parallel Processing

+ Programmable
+ Off-the-shelf, standard part
+ Many Multiplies in one clock cycle!
+ Extend the performance of DSP Processors



Multiple MACs, Parallel Processing

An Example: 120 Million Samples per Second 512-Tap Decimating FIR



- 3.8 Billion MACs
  - >10 DSP uPs
- 5,120 Flip-Flops
  - Just for data buffer
- XC4085XL
- 150,000 Gates

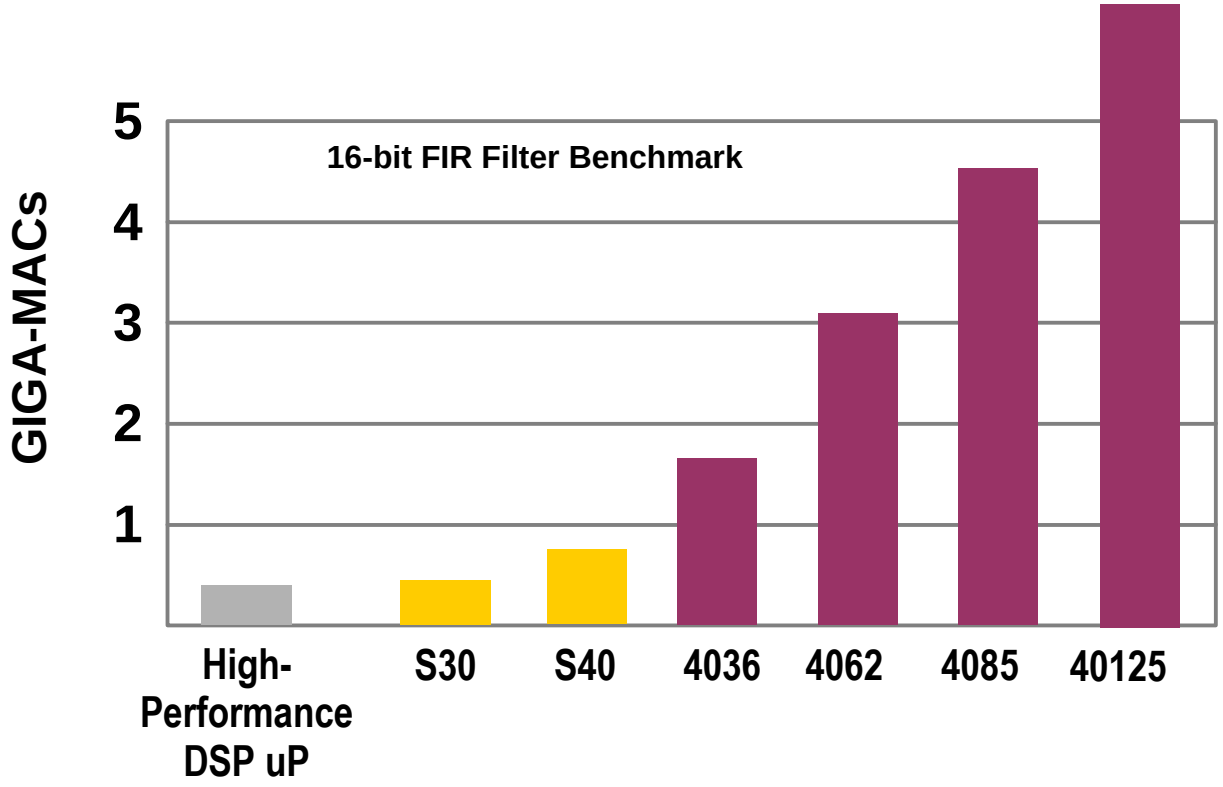
Serial Distributed Arithmetic FIR Filters

Serial Distributed Arithmetic — Data Word = Coefficient Size:

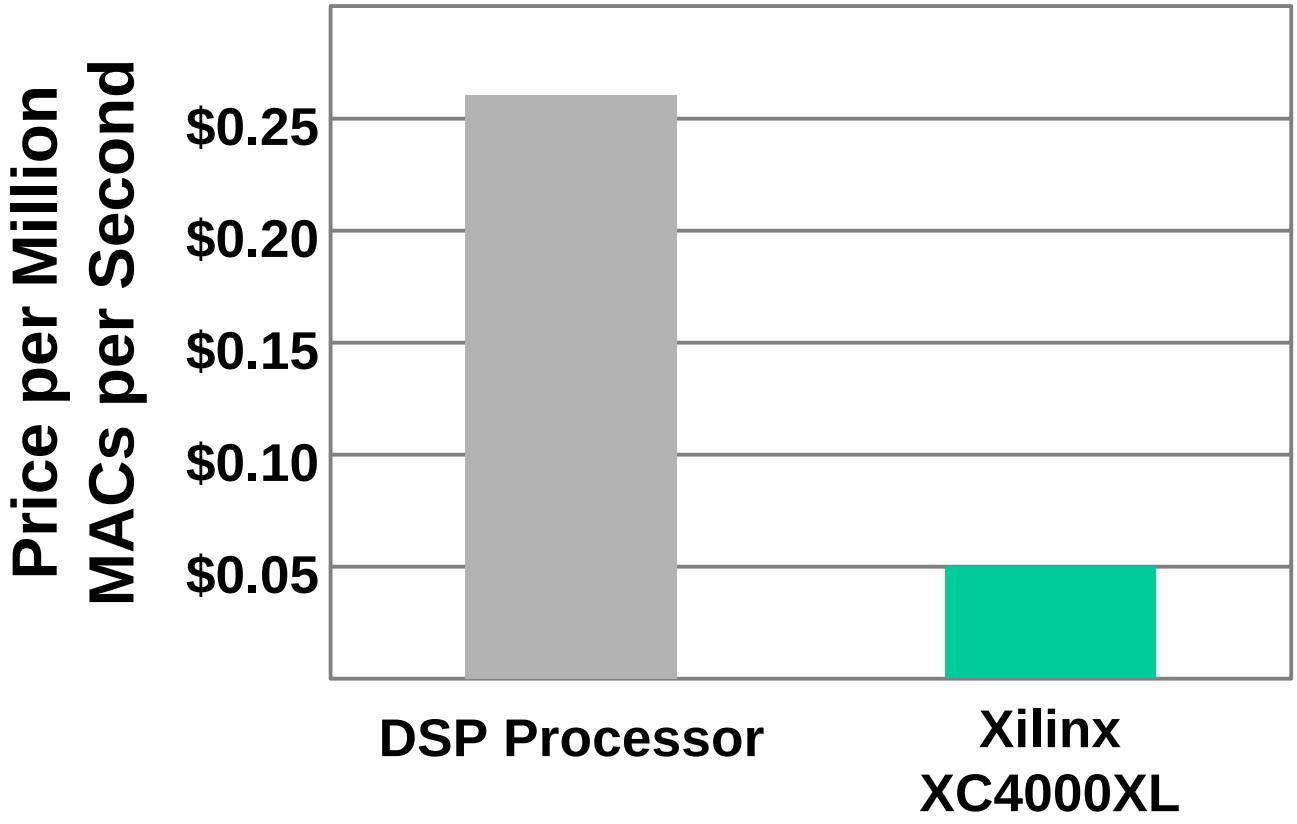
# CLBs		5 bit	8 bit	10 bit	12 bit	14 bit	16 bit	18 bit	20 bit
8 tap	Symm		33	36	39	42	45	52	55
	Non		46	54	59	64	69	77	85
16 tap	Symm	53	61	69	71	76	81	96	102
	Non		80	95	104	112	123	138	142
24 tap	Symm	80	89	101	108	116	127	146	154
	Non		101	114	127	140	153	174	187
32 tap	Symm	93	107	118	126	137	148	175	182
	Non								
40 tap	Symm	116	138	154	165	179	191	226	239
	Non								
48 tap	Symm		158	173	187	202	217	246	261
64 tap	Symm		197	215	233	250	268	305	323
80 tap	Symm		236	257	278	299	320	364	385

		5 bit	8 bit	10 bit	12 bit	14 bit	16 bit	18 bit	20 bit
Sample Rate	Symm	13.3	8.9	7.3	6.2	5.3	4.7	4.2	3.8
	Non	16.0	10.0	8.0	6.7	5.7	5.0	4.4	4.0
XC4000E-1		MHz	MHz	MHz	MHz	MHz	MHz	MHz	MHz

Greater than 10x DSP uP Performance



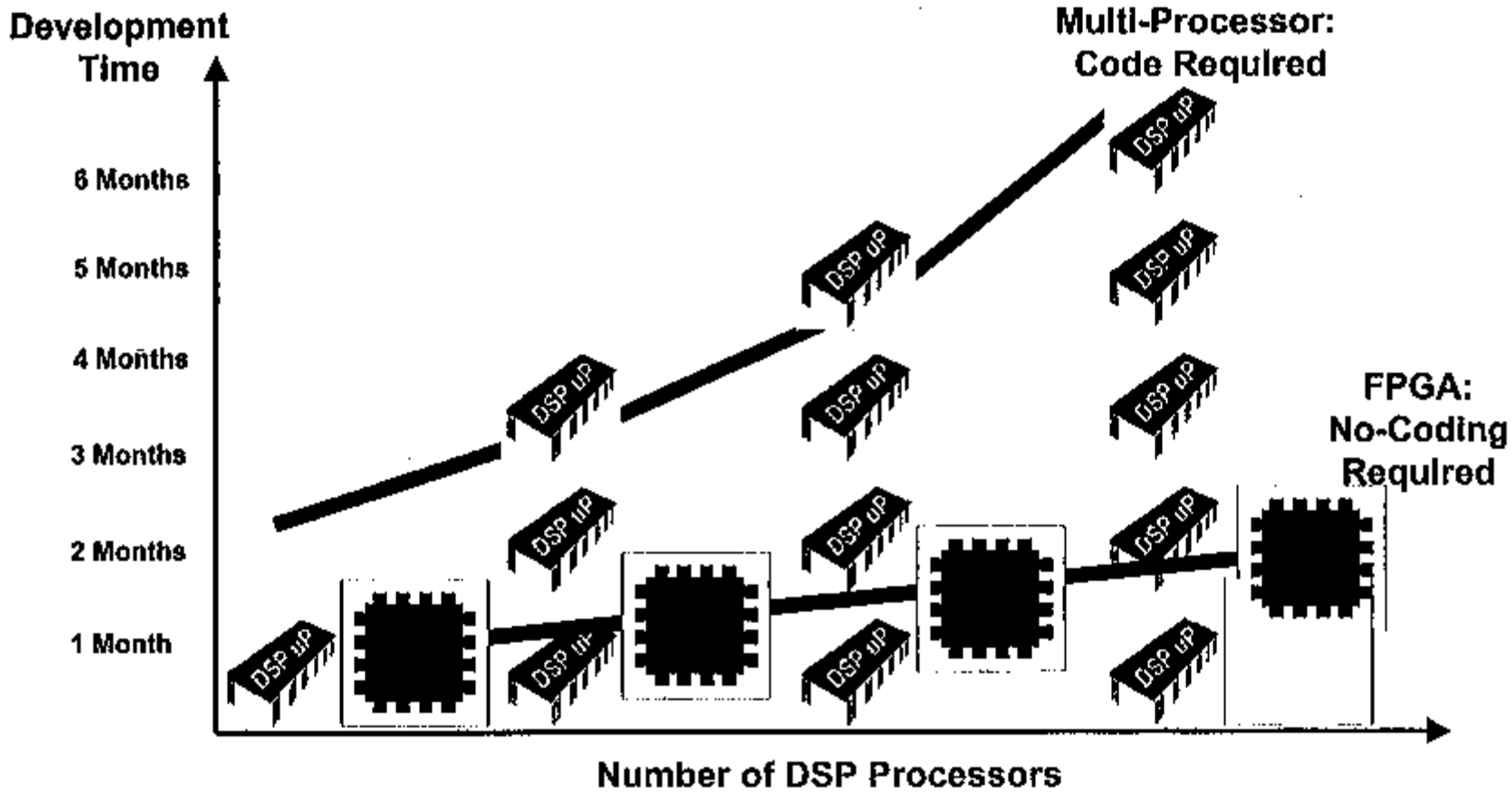
Price


Price per Million MACs per Second
$0.25
$0.20
$0.15
$0.10
$0.05
DSP Processor
Xilinx XC4000XL

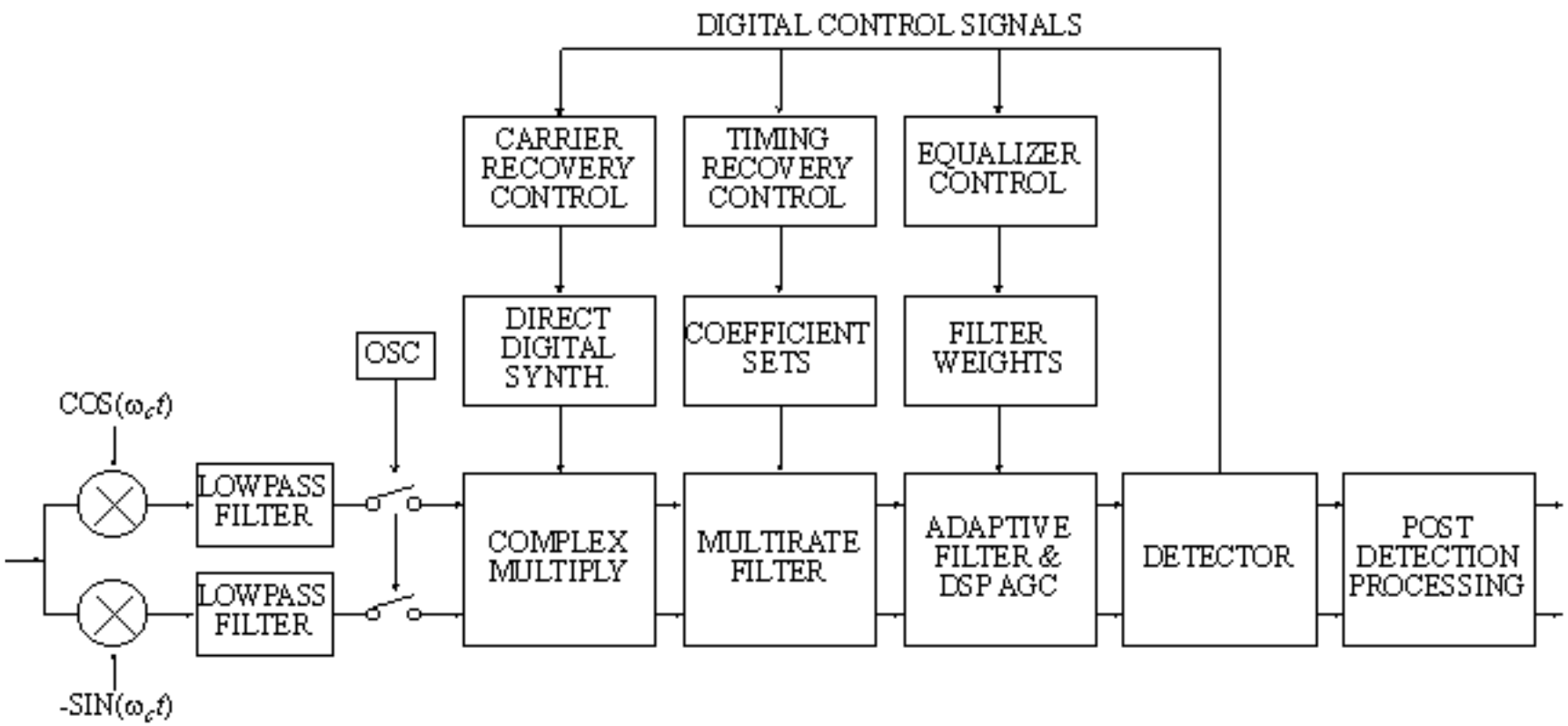
High Performance at a Fraction of the Cost

* Prices based on 50,000 PCS

… And with Faster Time-to-Market



Generic Digital Receiver

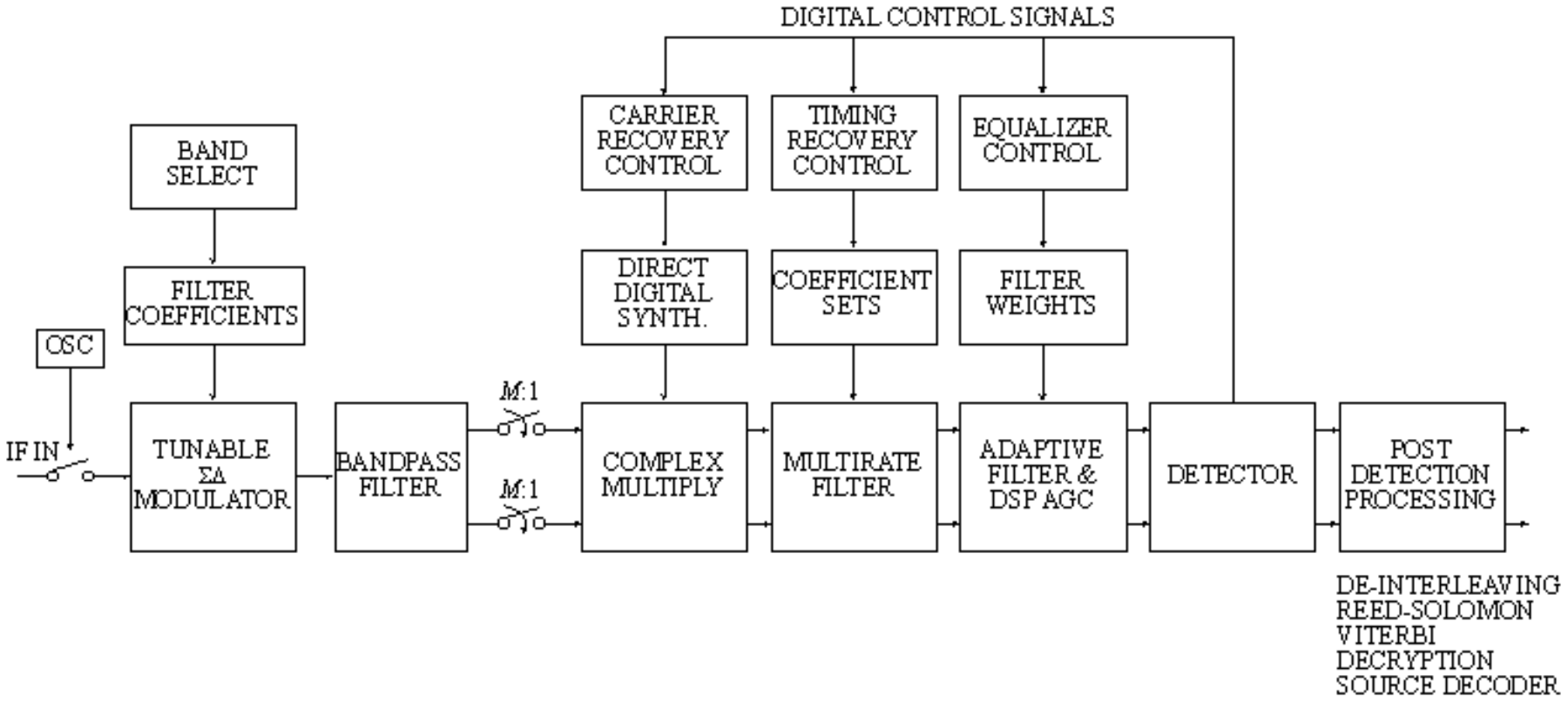


Alternative Approach - ΣΔ Signal Conditioner

- IF sampling collects data at a high rate with expectation that the band selection and bandwidth reduction will in the DSP block
- If output bandwidth is a small fraction of collected bandwidth, can argue that desired signal has been oversampled
- Use tunable ΣΔ modulator to requantize input to a small number of bits while preserving SNR in selected band
- Subsequent processing can proceed with reduced complexity hardware

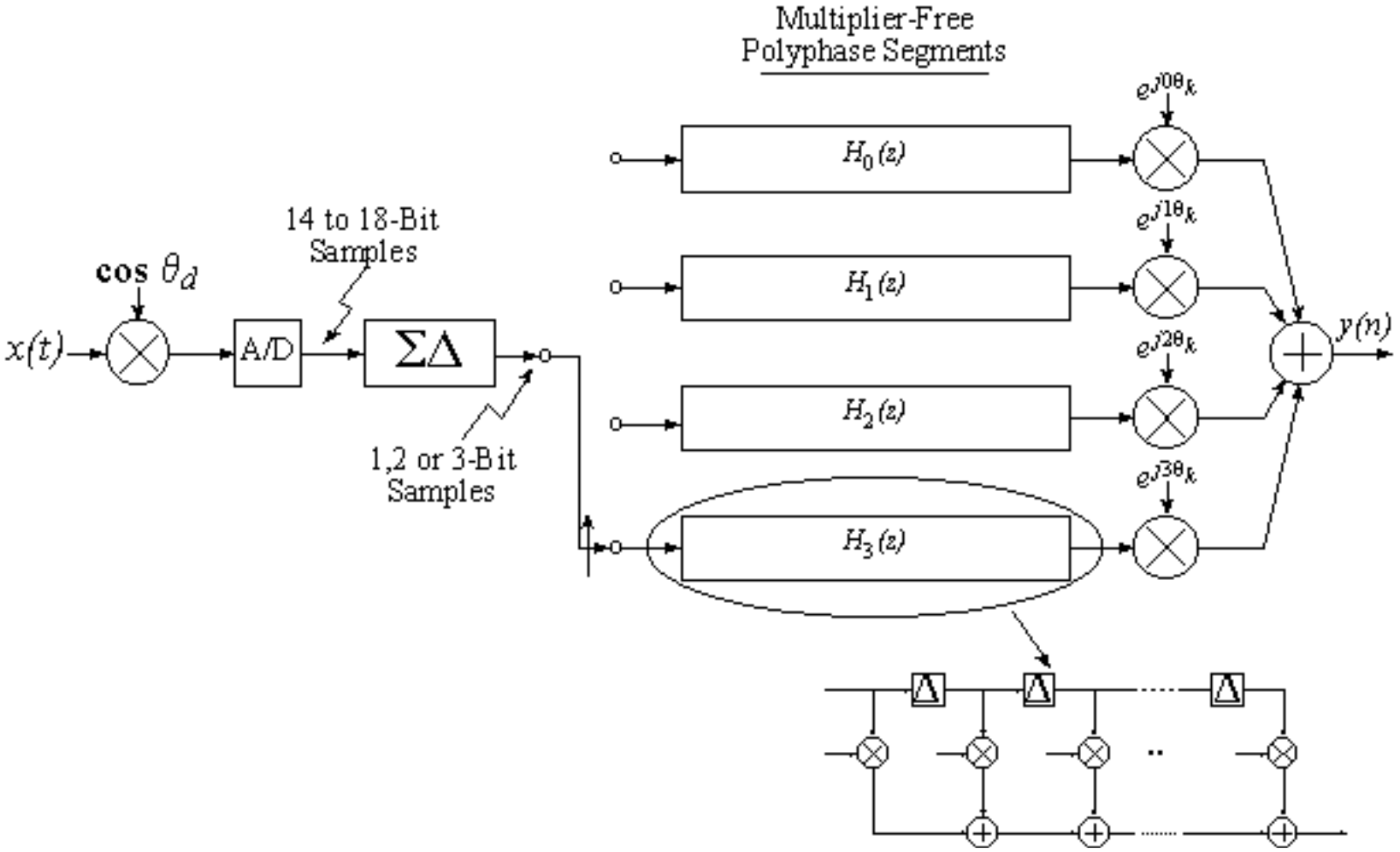
Digital Receiver - ΣΔ Front-End

- This architecture highlights potential offered by FPGAs to realize a solution not available using software programmable DSPs



XILINX®

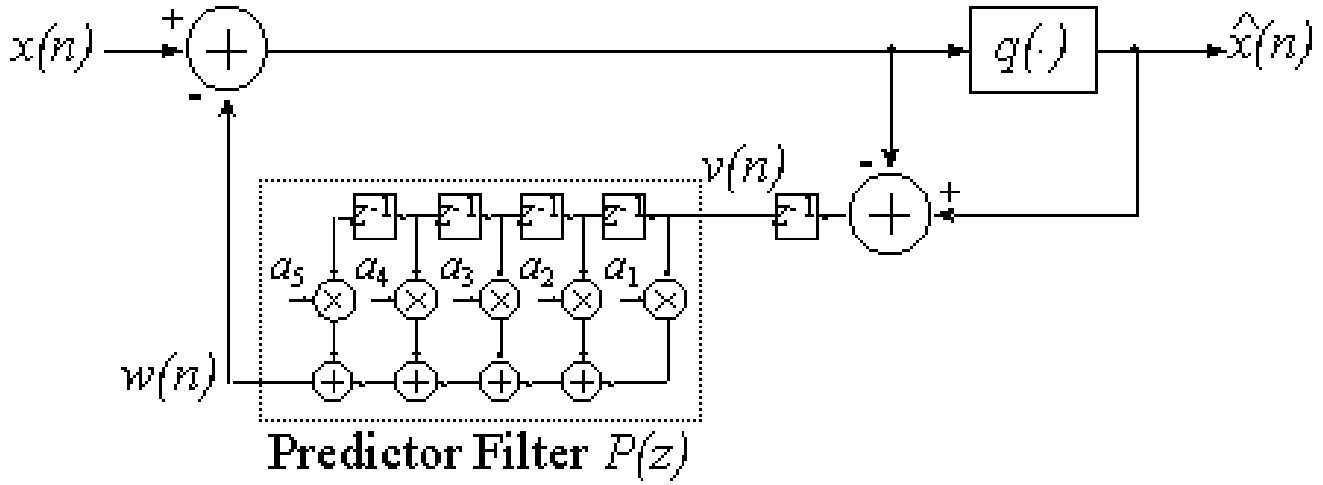
Channelizer with ΣΔ Front-End



XILINX®

ΣΔ Modulator Architecture

- Quantizing input samples to a lower precision before polyphase filter avoids multiplies ⇒ area efficient design



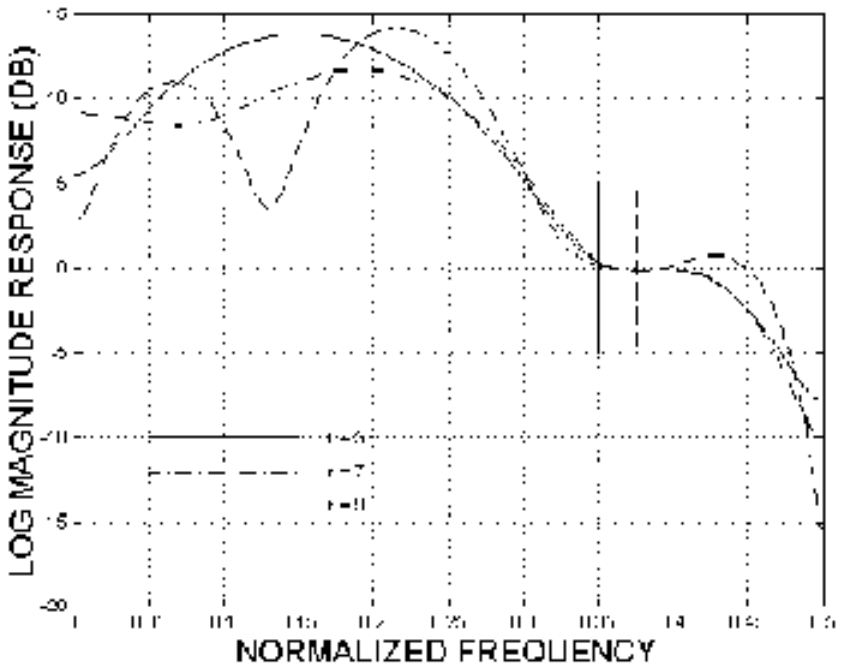
$$\hat{X}(z) = X(z) + Q(z)(1 - P(z))$$

In spectral region of interest $|P(z)| = 1$ and $P(z)$ has leading phase

$\Rightarrow Q(z)(1 - P(z)) = 0$ and $\hat{X}(z) \approx X(z)$ in bandwidth of interest

Predictor Filter Frequency Response

- Center frequency = 0.375 f_s

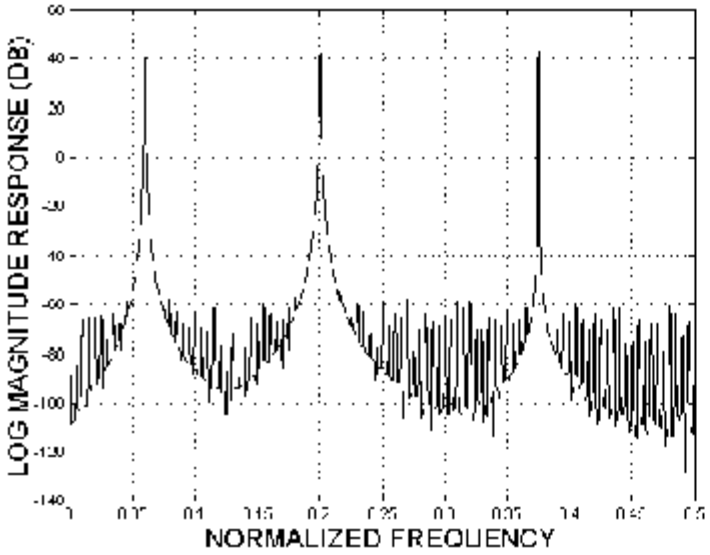


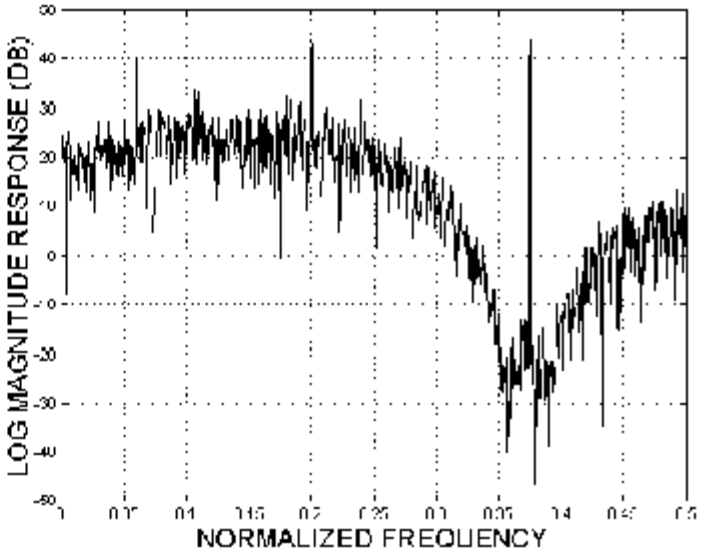
- 0 dB gain in region of interest
- leading phase - predictor

Performance

- Input spectrum - 3 tones
- Required to recover tone at 0.375 f_s

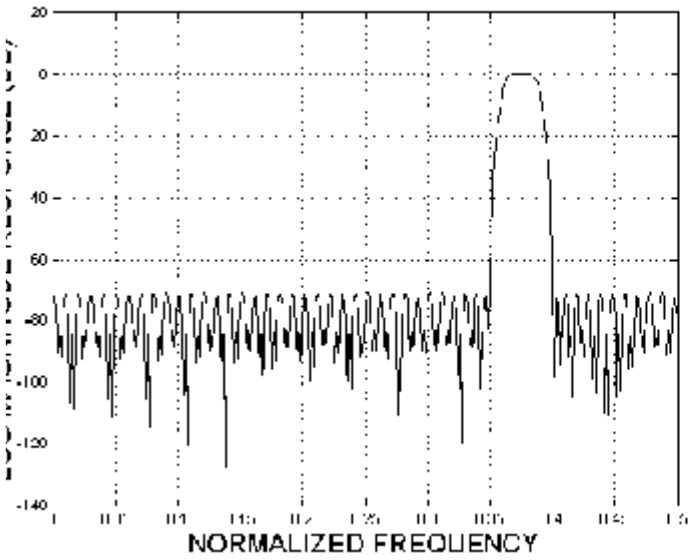
- Re-Quantized Spectrum
  - dynamic range preserved in region of interest and sacrificed in spectrum *don't care region*



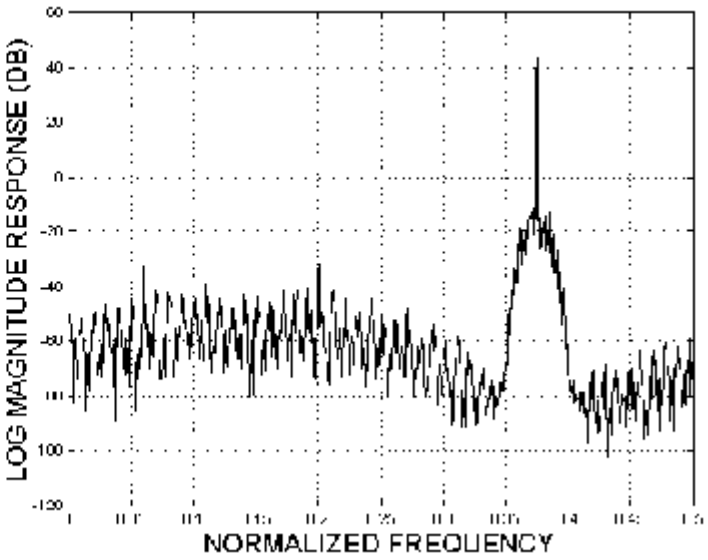


Performance

Filter Response



Filter Output

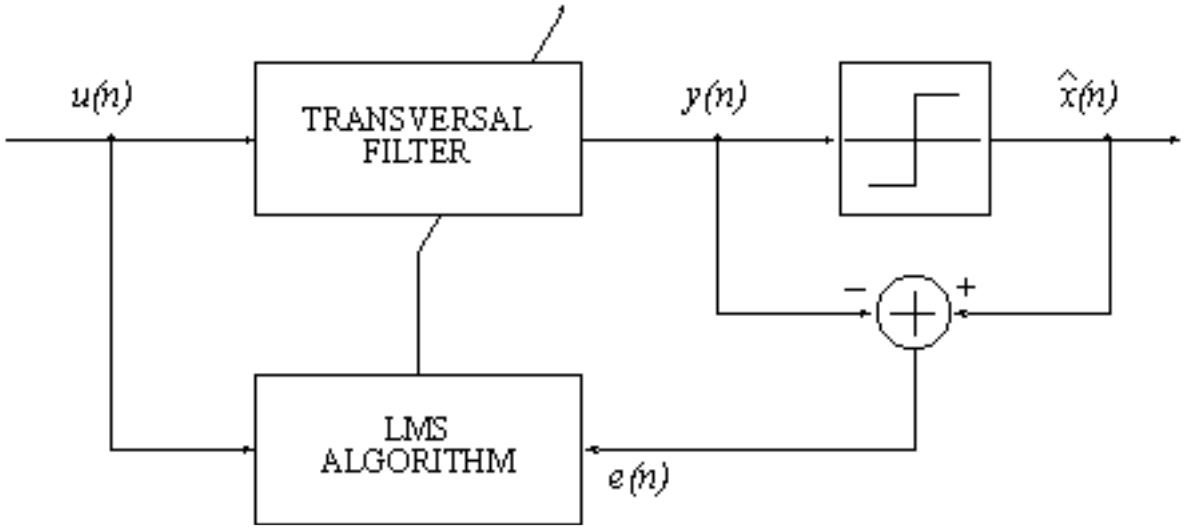


Receiver with ΣΔ Pre-processor

- No need for full multipliers in polyphase filter
- 2 CLBs per tap using bit-serial approach
  - (independent of coefficient precision)
- 12 CLBs per tap using 12b coefficients parallel approach
- Example highlights the architectural tradeoffs possible using FPGA DSP that are not accessible using software DSP

High-Performance FPGA Adaptive Filter

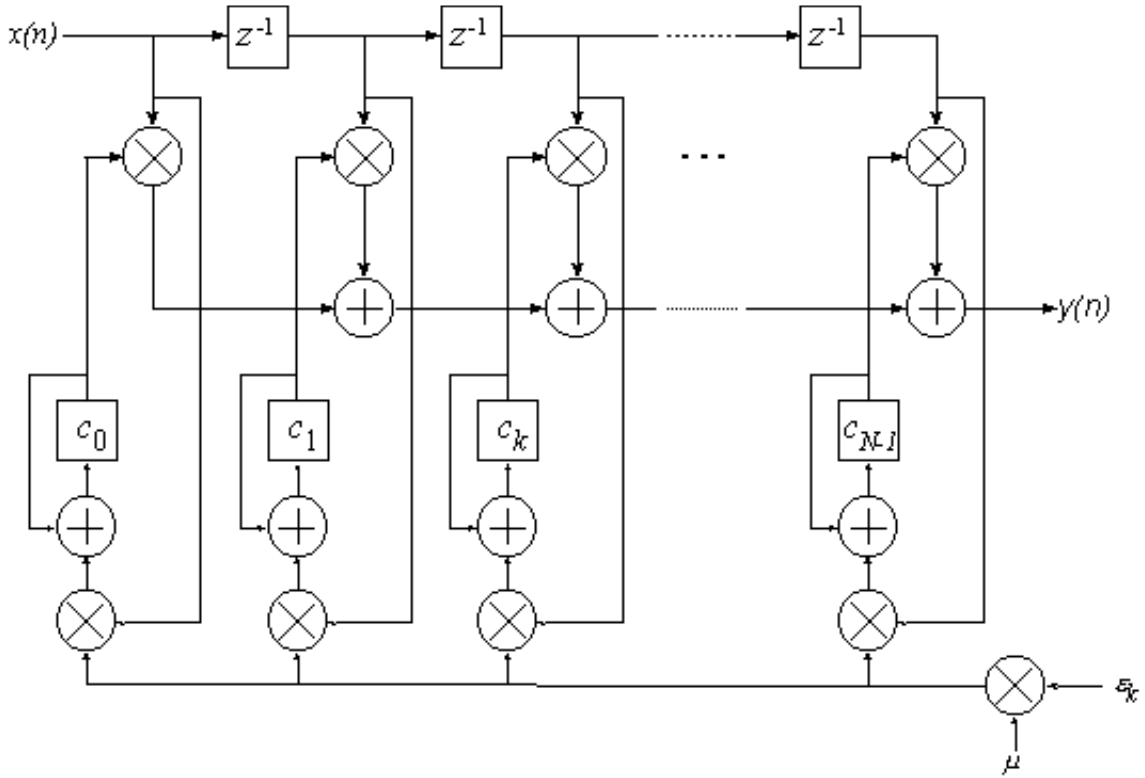
- Decision-directed equalizer



Adaptive Filter

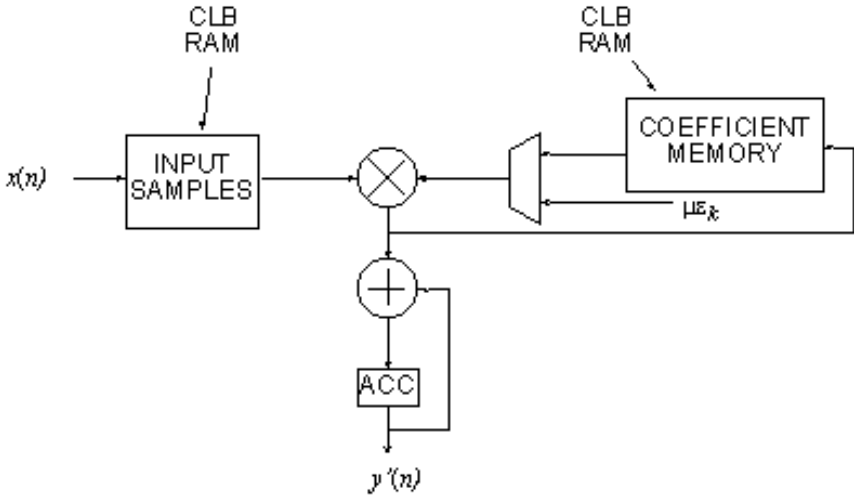
- Consider inner-product and coefficient update

$$c_{k+1} = c_k + me_k x_k$$



Adaptive Filter

- Build adaptive FIR with filter slices
- These can be employed in a parallel arrangement to provide higher performance
- Adder-tree used to combine slice outputs



- 16-b precision slice 246 CLBs
- 12-b precision slice 151 CLBs

Adaptive FIR Performance

- Space-time tradeoffs highlighted
- 16-bit precision, 70 MHz clock

N	# SLICES	f_s (MHz)	CLBS
16	1	2.2	246
	2	4.4	501
	4	8.8	1011
	8	17.6	2024
	16	35.2	4071
32	1	1.1	246
	2	2.2	501
	4	4.4	1011
	8	8.8	2024
	16	17.6	4071

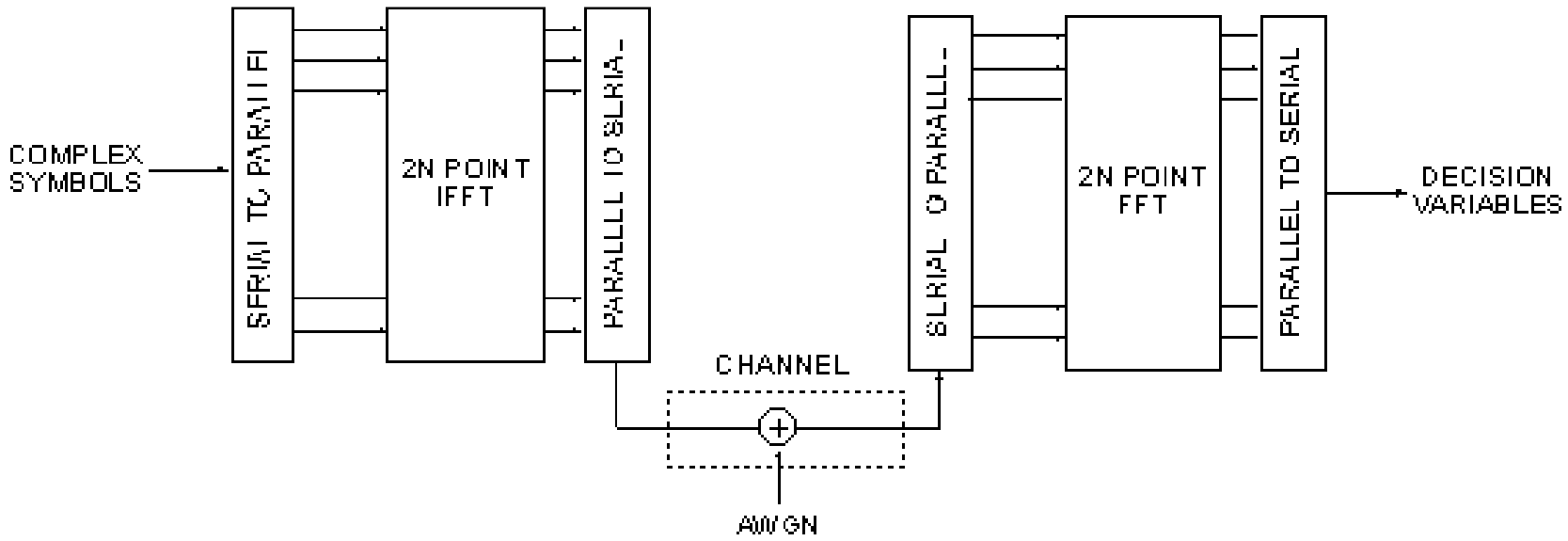
Adaptive FIR Performance

- Space-time tradeoffs highlighted
- 12-bit precision, 80 MHz clock

N	# SLICES	f_s (MHz)	CLBS
16	1	2.5	246
	2	5.0	501
	4	10	1011
	8	20	2024
	16	40.0	4071
32	1	1.25	246
	2	2.5	501
	4	5.0	1011
	8	10.0	2024
	16	20.0	4071

FPGAs, FFTs and OFDM

- OFDM modulation and interpolation implemented by zero-extended FFT



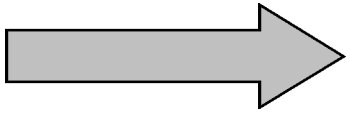
FPGA FFT Performance

- 16-bit complex input/output
- 16-bit phase factors
- Table highlights FPGA capacity for Space-Time tradeoff

N	CLB COUNT / EXECUTION TIME µs					
	CLBs	T	CLBs	T	CLBs	T
16	1512	0.267	1200	0.53		
32	800	2.3				
64	1295	4.2	1545	2.8	750	25
256	1855	18.9				
1024	1495	86	1750	69	3500	35

System Design Using FPGAs

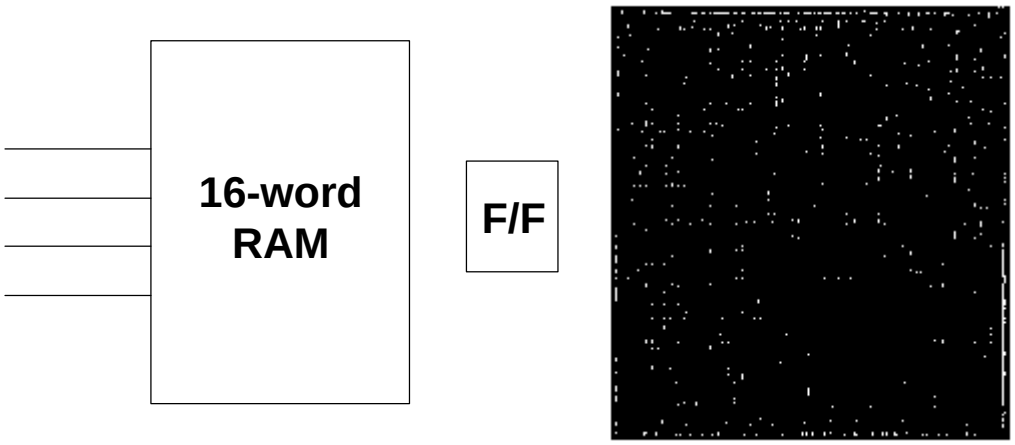
- Schematic capture, HDLs
- High-level modules - *COREs* - available to enhance system design, development, productivity



Decrease time-to-market

- Some DSP CORE examples
  - FFTs
  - FIR filters
  - Reed-Solomon
  - DDS
  - many others

DSP LogiCOREs Exploit FPGA Architecture



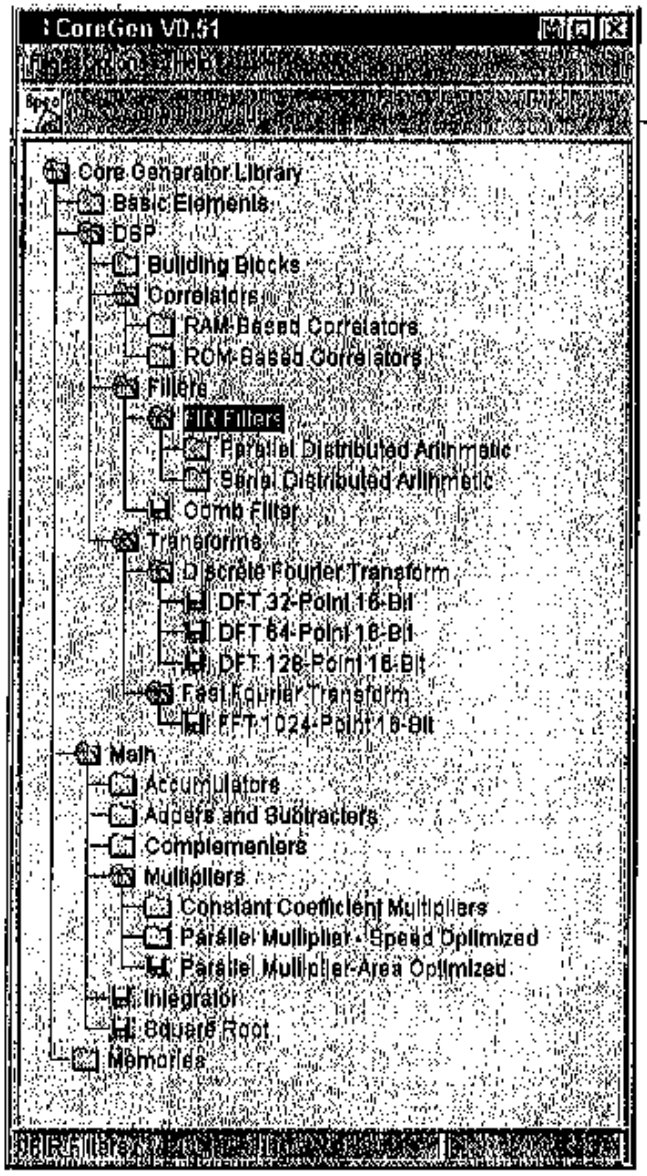
Matrix of 16 by 1 RAM primitives

- Look-up-table logic
- FIFOs, shift-registers, …
- Multiple small memories

10,000 RAM primitives on a chip

Regular, monolithic, scalable structure

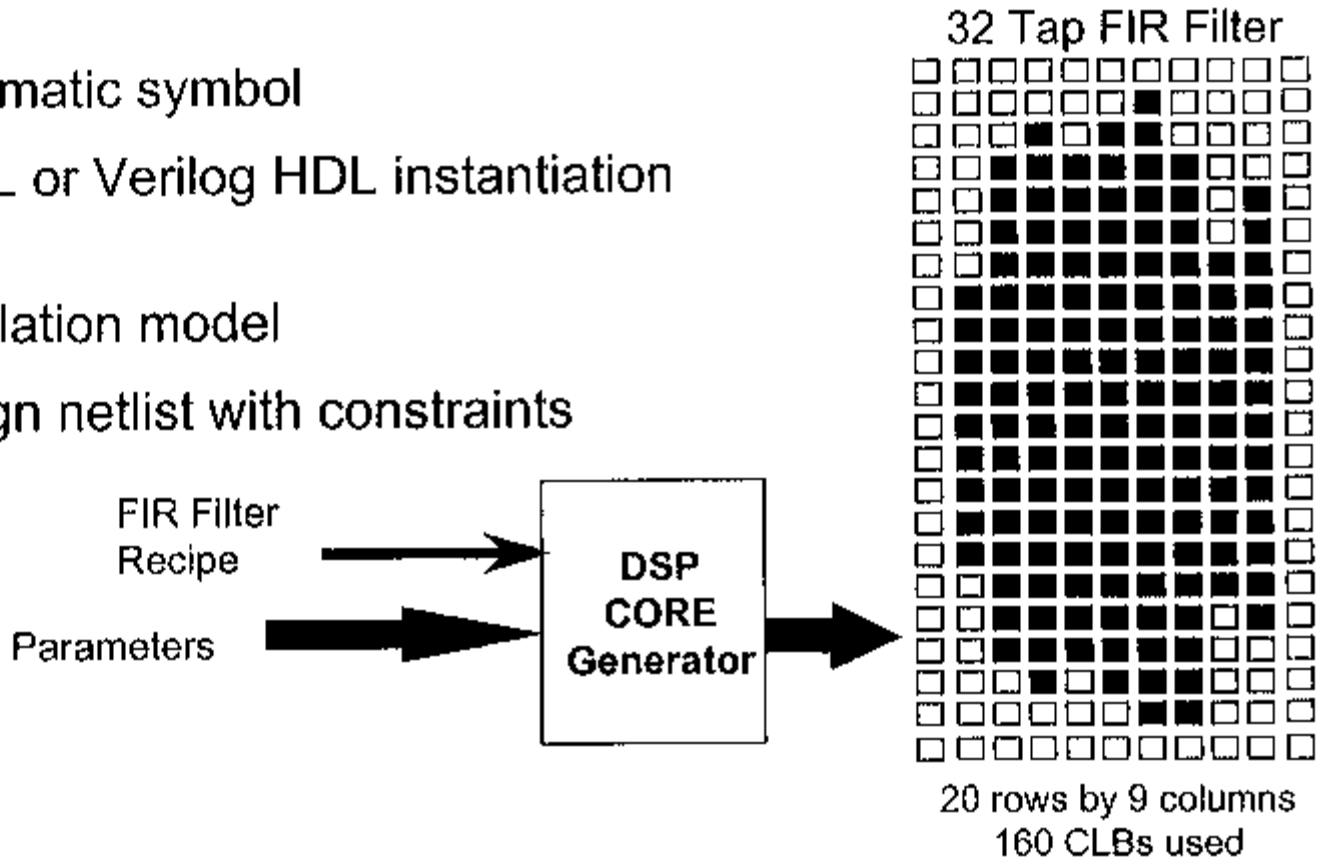
Efficient: 1 - 3 Million MACs per CLB



CORE Generator Design Flow

DSP CORE Generator Outputs

- Schematic symbol
- VHDL or Verilog HDL instantiation code
- Simulation model
- Design netlist with constraints



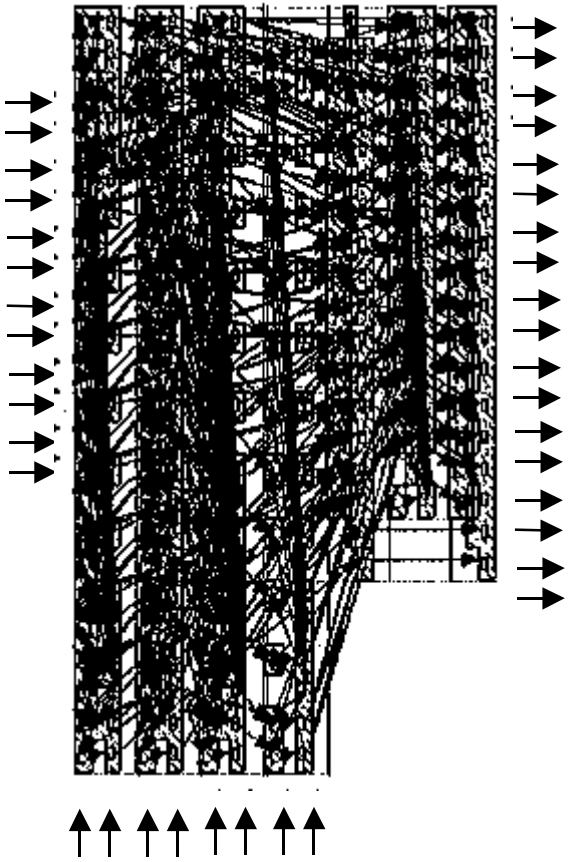
Predictable Performance regardless number of cores

Predictable Size & Performance

- Built for System Performance - Not Benchmarks.
- Generated with RPM (Relationally Placed Macro).

RPM Macro Level Advantages

- Predictable size.
- Close proximity of communicating elements
- Alignment of Critical paths
- Accessible I/O signals
- Improves Density

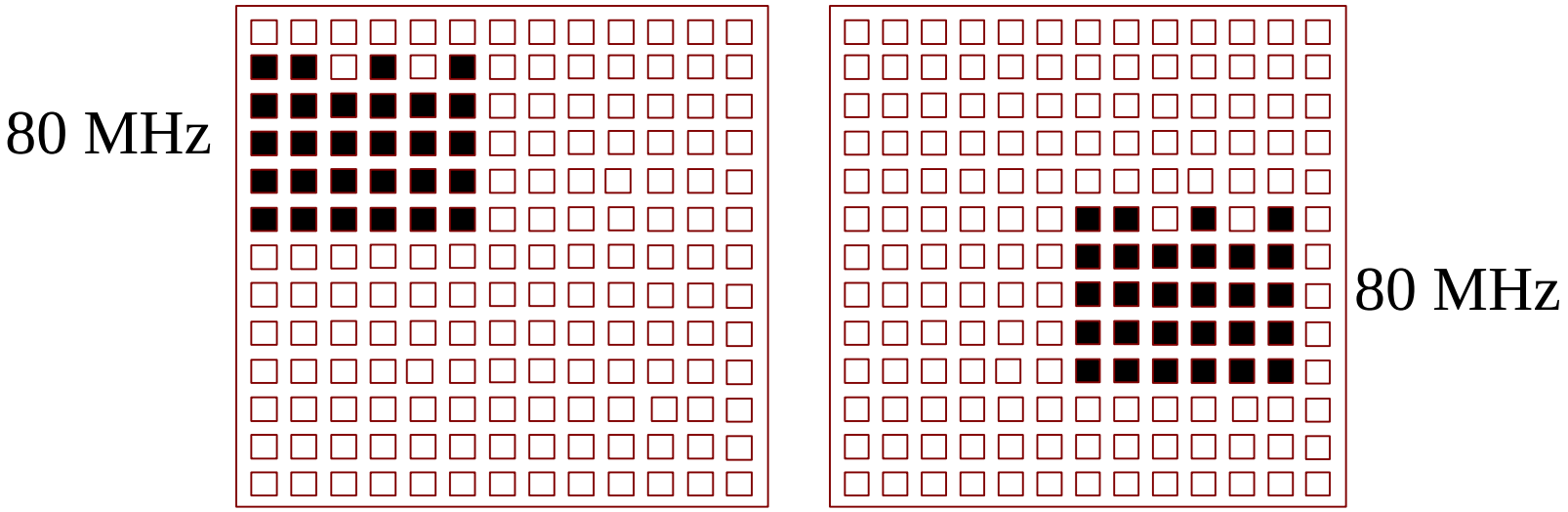


RPM System Level Advantages

- Rapid progress for automatic and manual design methods (1 macro, NOT 100's of elements!)
- Consistent performance anywhere on the die.
- Packing density very high
- Adequate set-up times

Filling a device with Xilinx Cores does not reduce performance

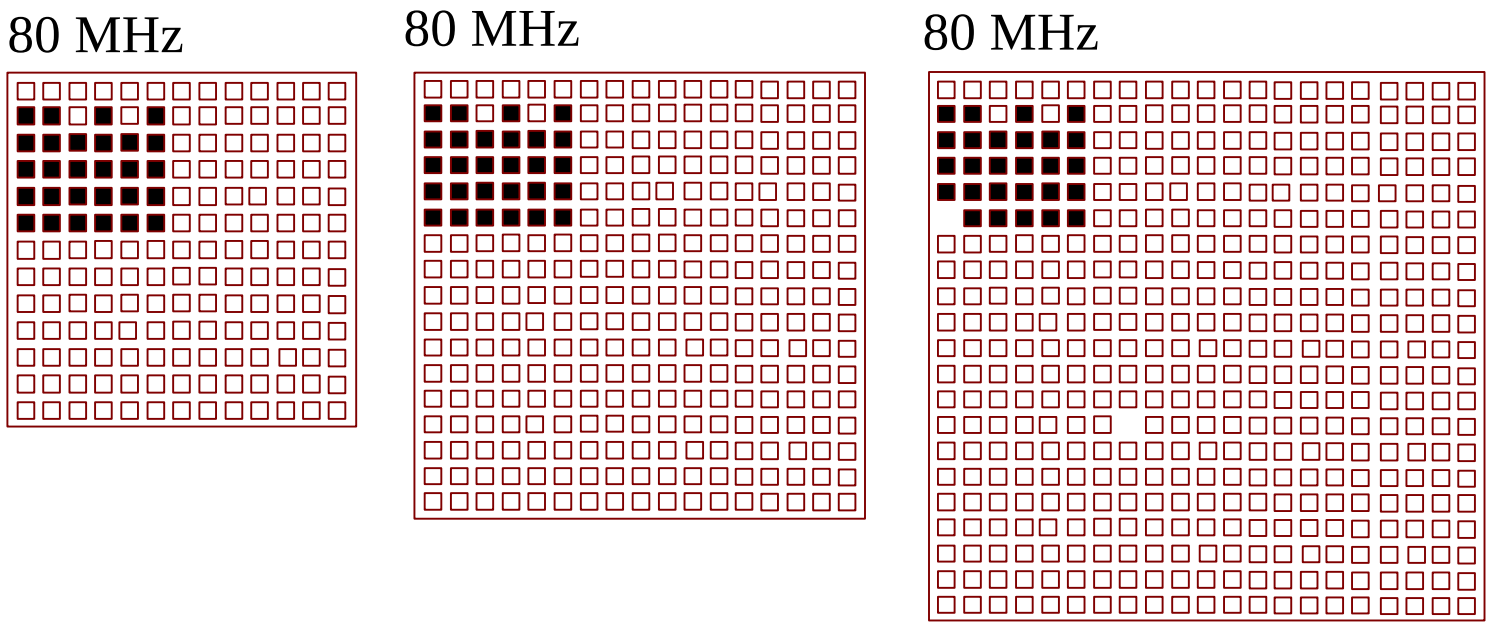
Performance Independent of core location



Same core installed in different locations

- Xilinx LogiCOREs deliver the same performance for any placement
- Non-segmented routing FPGAs can’t do this

Performance Independent of Device Size

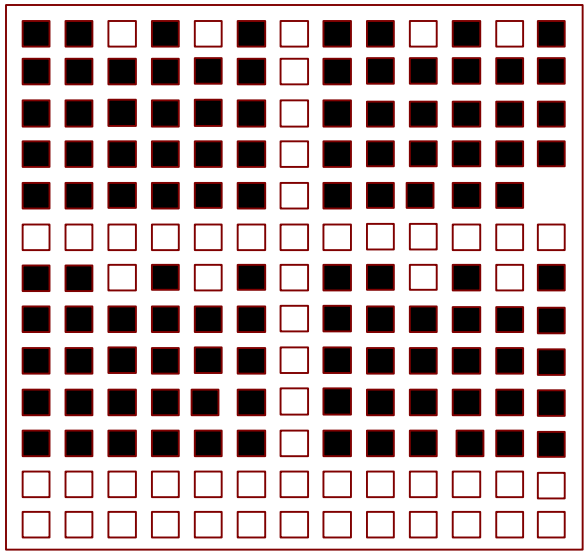


- Same performance for a 4005 or 4085
- Non-segmented routing FPGAs can’t do this

Performance Independent of Device Utilization

80 MHz

80 MHz

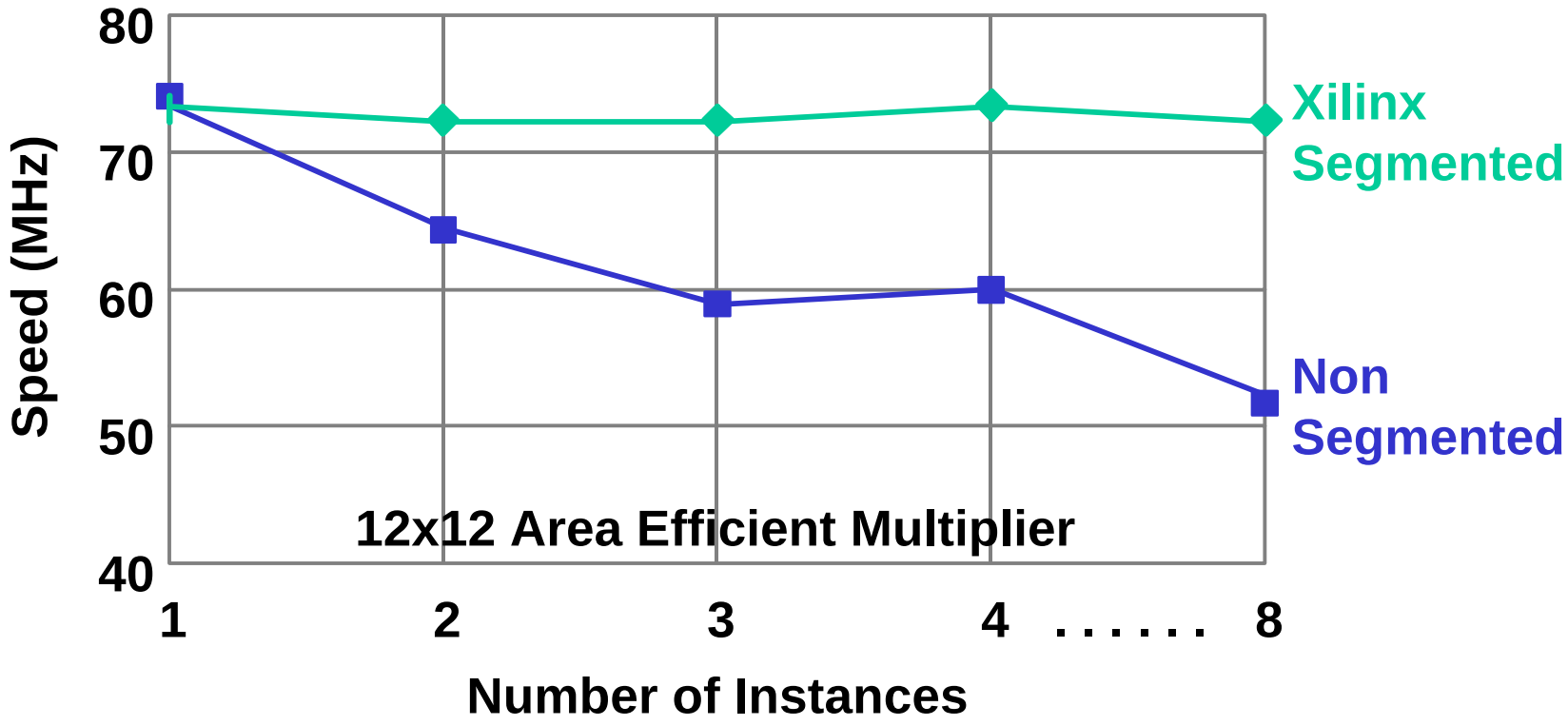


80 MHz

80 MHz

- Xilinx has performance independent of the number of cores added
- Non-segmented routing FPGAs can't do this

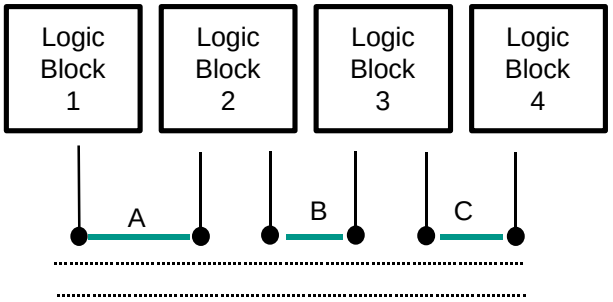
Segmented Vs Non-Segmented Routing



Segmented = More Predictable and Repeatable

Segmented Routing = Low Power

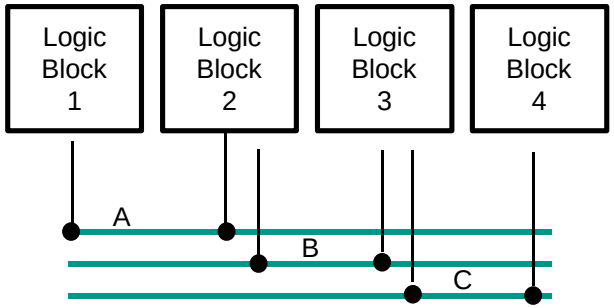
Xilinx FPGA Architecture



"Segmented" Interconnect Lines

- Lower capacitance on short lines means lower power

Other Architecture

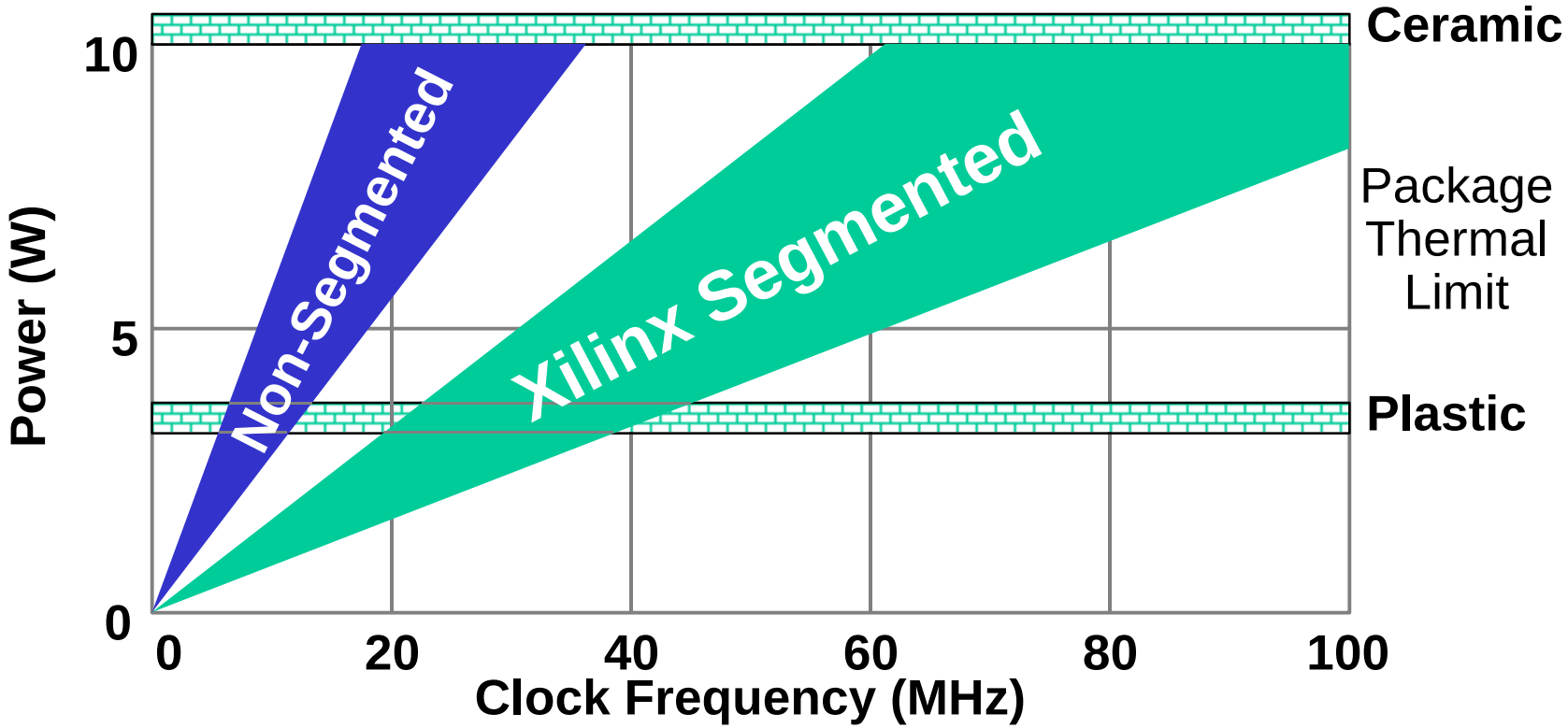


"Non-Segmented" Interconnect Lines

- Higher capacitance on each net means higher power

Xilinx **routing is less than 2/3 non-segmented FPGA power - assuming same process**

Segmented Interconnect Yields Lower Power

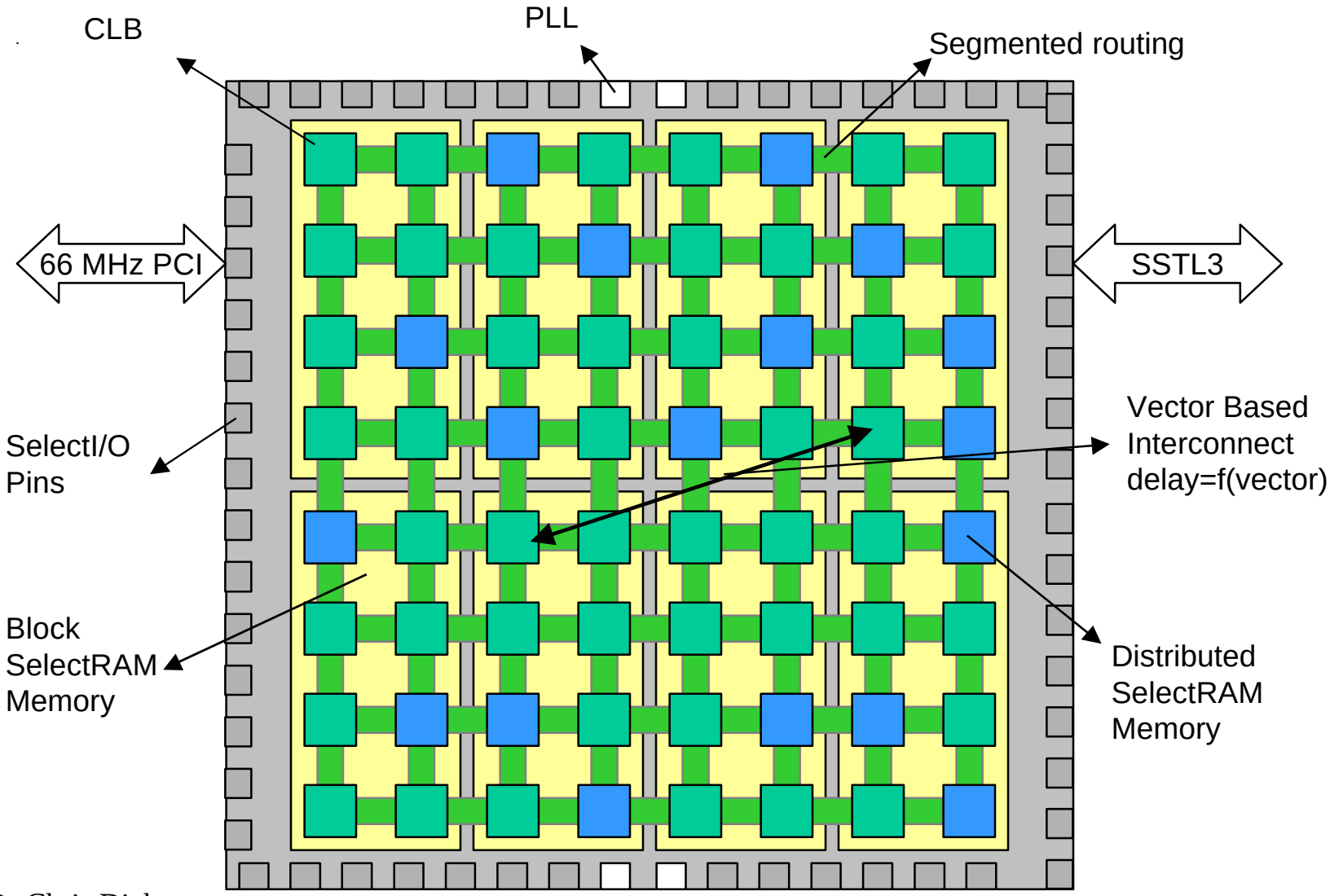


Segmented = Lower Power, Faster Operation

Power Dissipation Advantage
Often the Limiting Factor In DSP

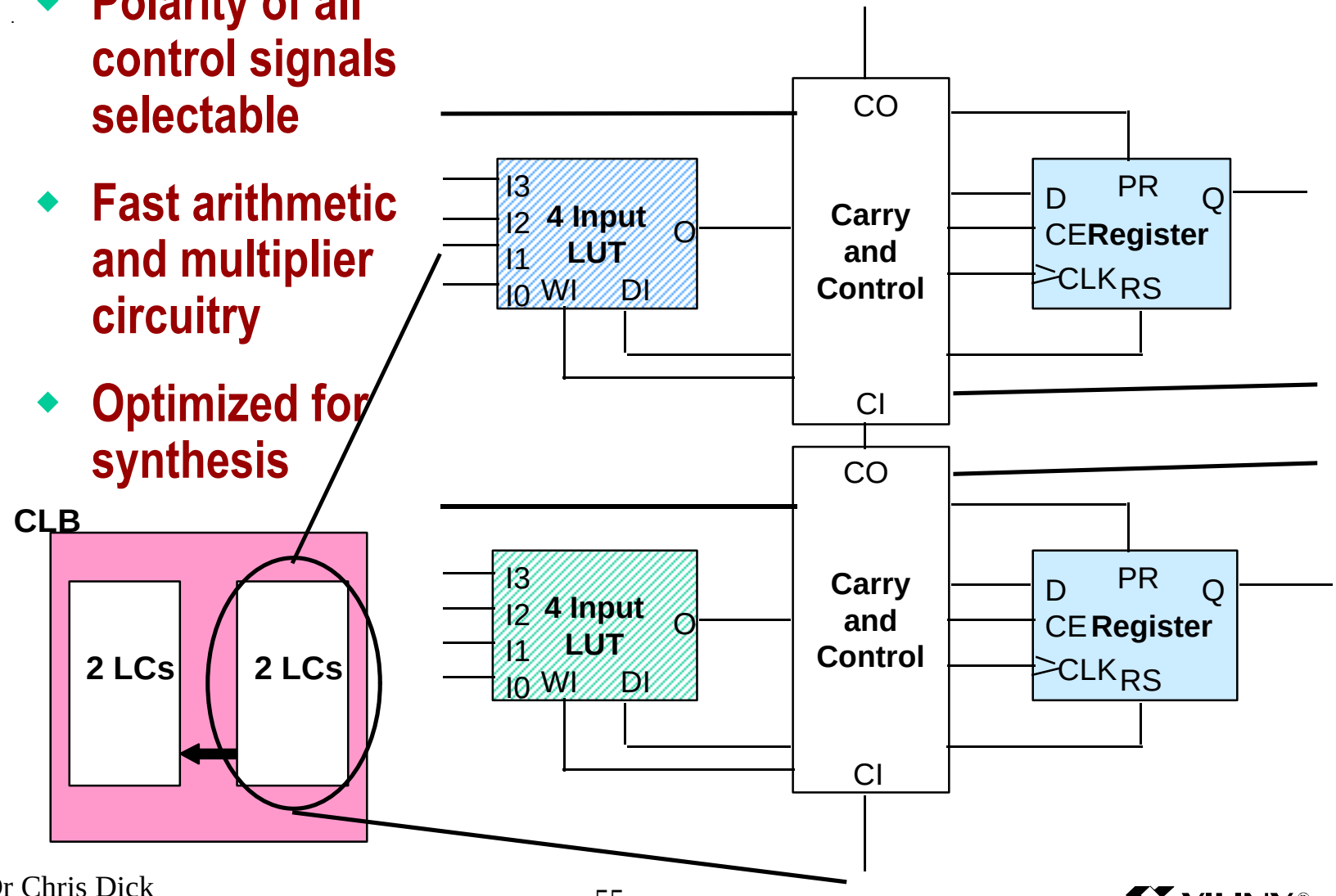
- Xilinx Advantage over competitive FPGAs
  - Segmented routing is essential in DSP applications
  - 3:1 advantage over non-segmented architecture
- Xilinx advantage over DSP processors:
  - Half the power of popular DSPs
    - Independent study by Stanford

New Devices for FPGA DSP - *Virtex*



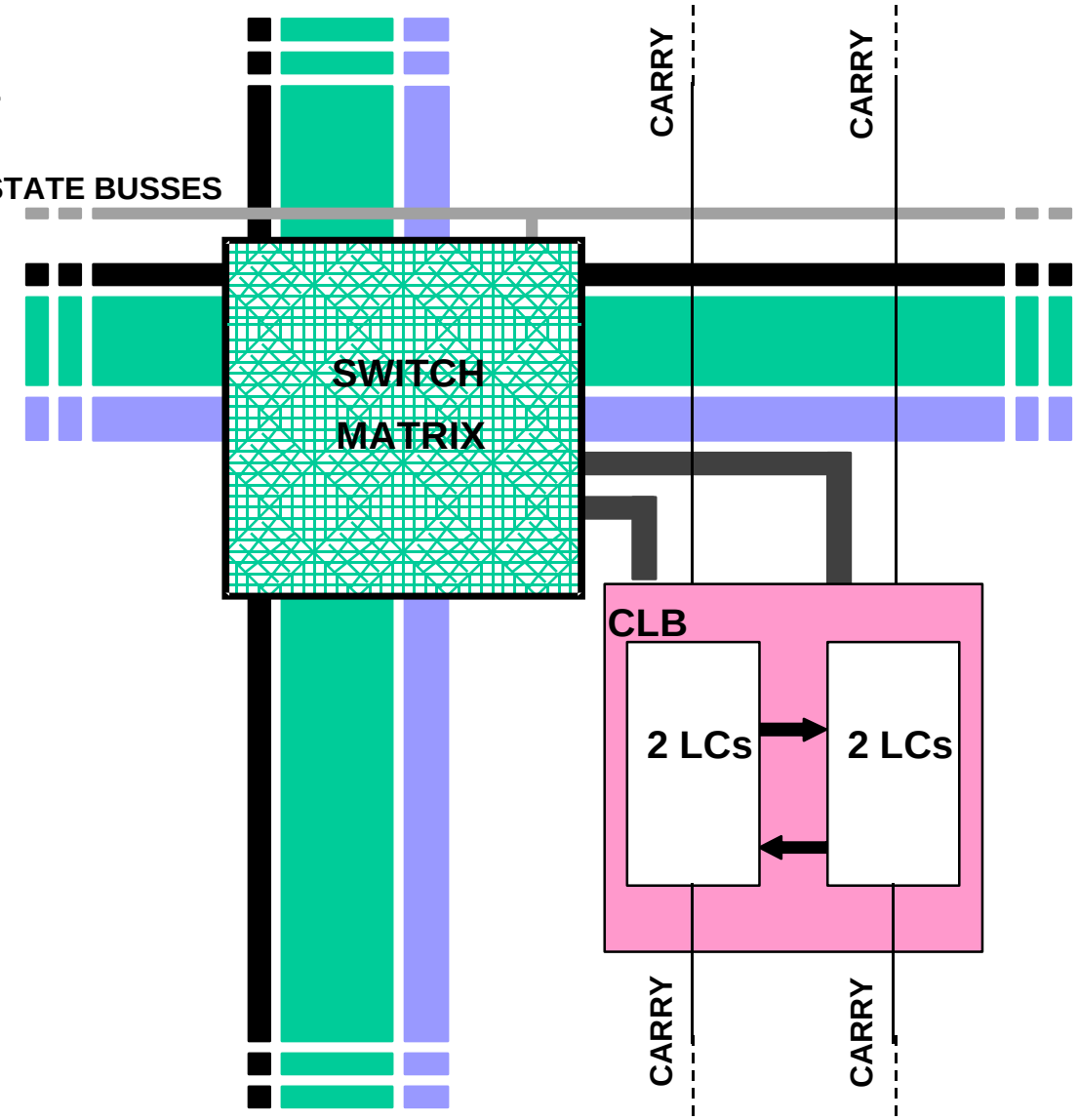
Virtex Configurable Logic Block

- Polarity of all control signals selectable
- Fast arithmetic and multiplier circuitry
- Optimized for synthesis



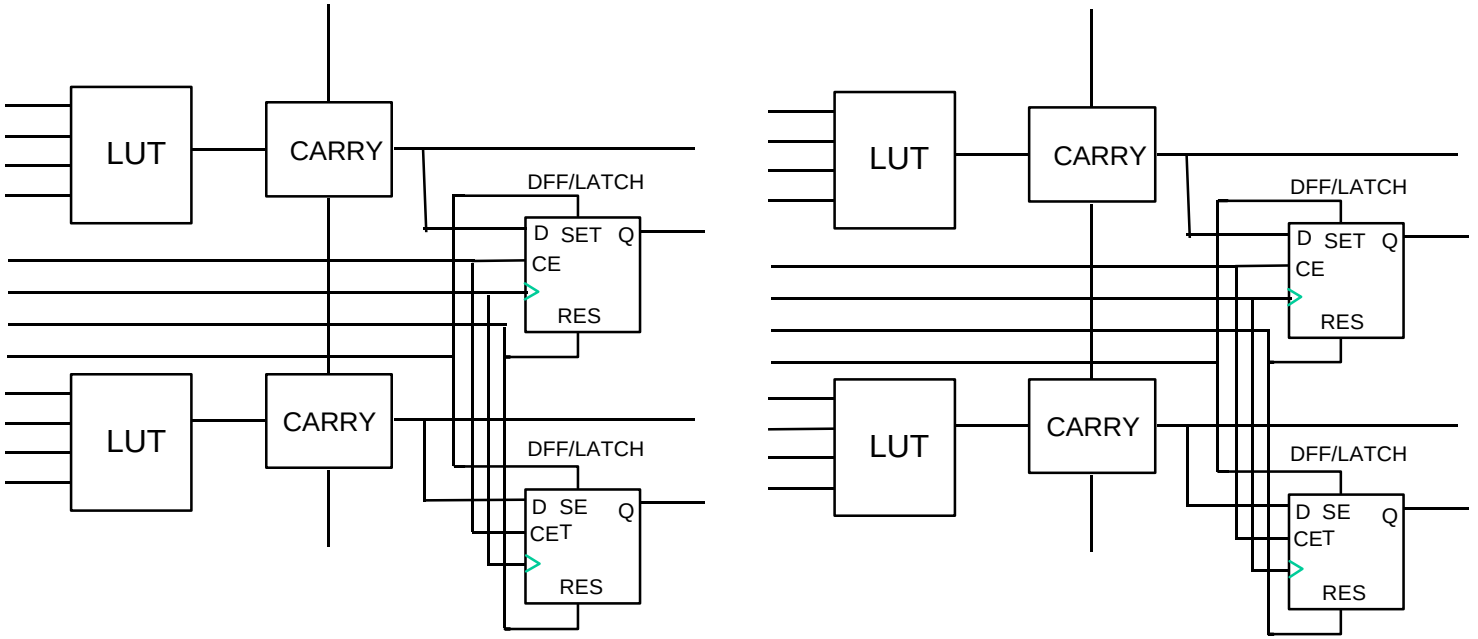
Segmented Routing Interconnect

- Fast local routing within CLBs
- General purpose routing between CLBs
- Fast Interconnect
  - 8ns across 250,000 system gates
- Predictable for early design analysis
- Optimized for five layer metal process



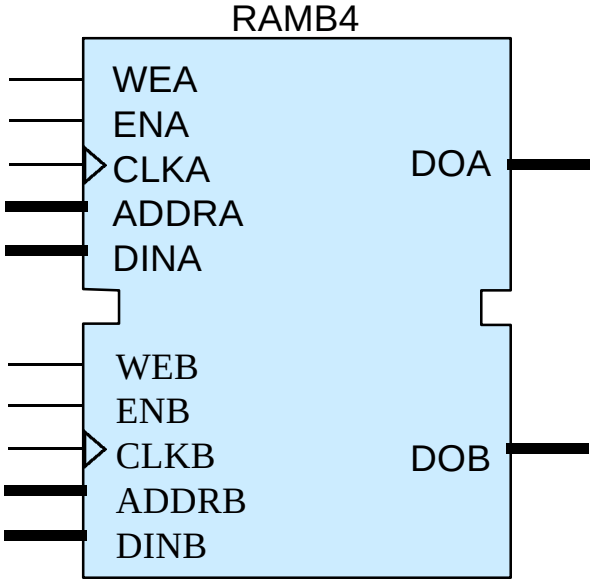
Simplified CLB

- 4 Logic Cells per CLB
- Carry logic (2 independent chains)
- 4 FFs/Latches, 2 BUFTs per CLB
- 16 bits of SelectRAM per LUT (single / dual-port)



Block RAM

- Configure as: 4096 bits with variable aspect ratio
- 8-32 blocks per device
- True dual-port, fully synchronous operation
  - Cycle time <10 ns
- Flexible block RAM configuration
  - 5 blocks: 2K x 10 video line buffer
  - 1 block: 512 x 8 ATM buffer (9 frames)
  - 4 blocks: 2K x 8 FIFO
  - 9 blocks: 4K x 9 FIFO with parity



Conclusion

- FPGAs
  - maintain flexibility with high-performance
  - track standards evolution
- Performance through parallelism

  $\Rightarrow$high sample rates

  $\Rightarrow$high BW

  $\Rightarrow$concurrent I/O
- FPGA based design opens new-dimensions in the design space
  - use bit-width appropriate for your design and for

    for different modules in same design

Conclusion (cont'd)

- CORE generator facilitates DSP datapath implementation
- System level tools
  - Elanix System View
  - others under development

Conclusion (cont'd)

- New architectures - *Virtex*
- Virtex 1000
  - 0.25 micron
  - 75 million transistors
  - 100 MHz
- DSP support
- Partially reconfigurable for hardware sharing