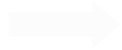




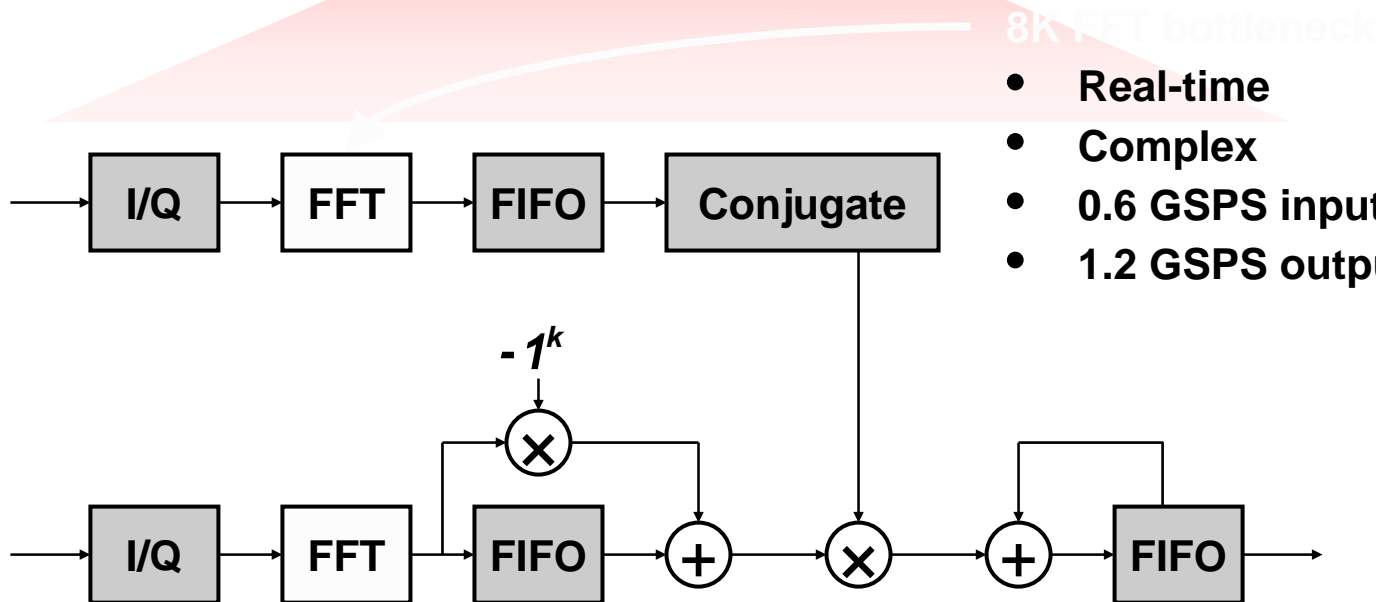
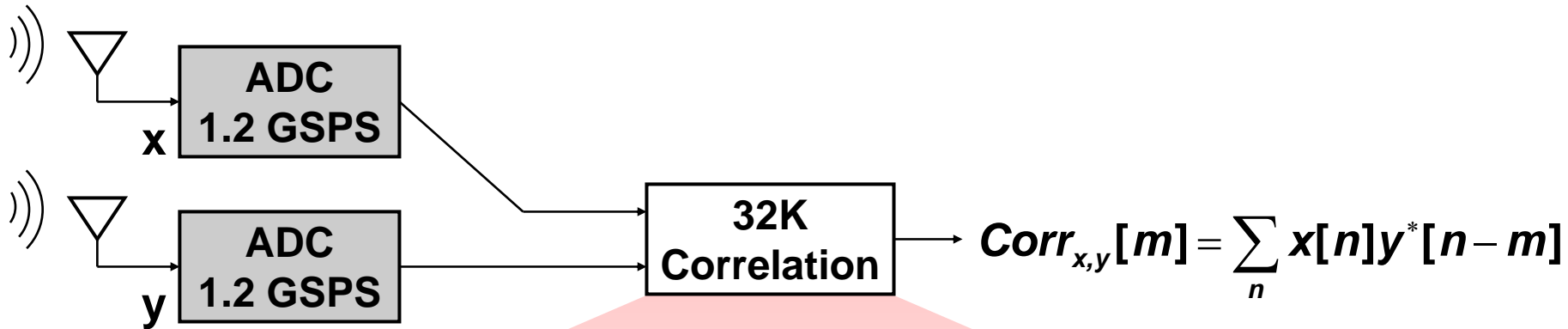
Outline



- Introduction
 - Motivation
 - Evaluation metrics
- Parallel architecture
- Systolic architecture
- Performance summary
- Conclusions



Radar Processing Application



- Real-time
- Complex
- 0.6 GSPS input (16-bits)
- 1.2 GSPS output (12-bits)



Evaluation Scorecard

- The design changes will be scored based on the following metrics:

	Size	16	8192	Δ
Length of FFT				
IO pins	Pins	?	?	?
Butterflies	Fly	?	?	?
Multipliers	Mult	?	?	?
Adder/subtractors	Add	?	?	?
Shift registers	Shift	?	?	?

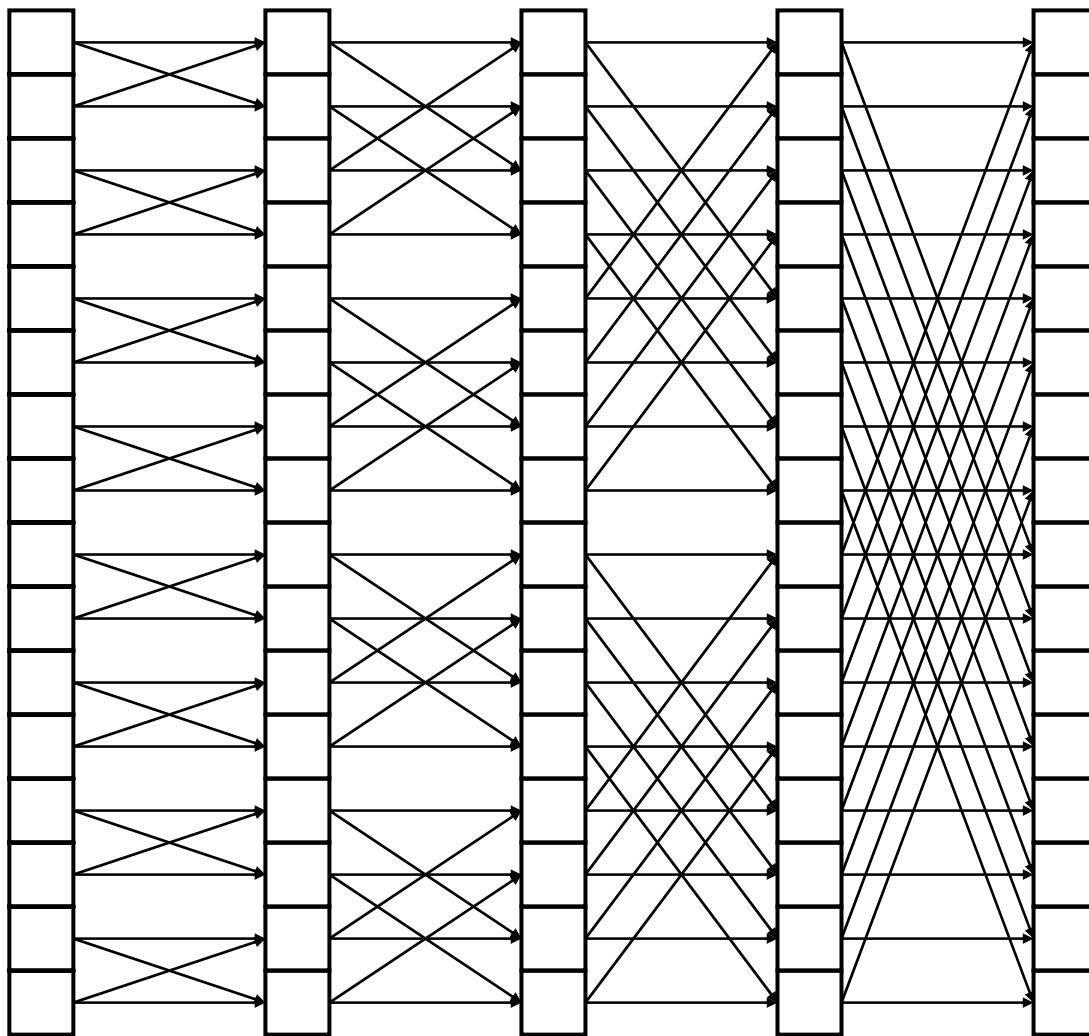


Outline

- **Introduction**
- **Parallel architecture**
 - **Data flow graph**
 - **Effects of serial input**
- **Systolic architecture**
- **Performance summary**
- **Conclusions**



Baseline Parallel Architecture



Size	16	8192	Δ
Pins			
Fly			
Mult			
Add			
Shift	0	0	

Parallel FFT

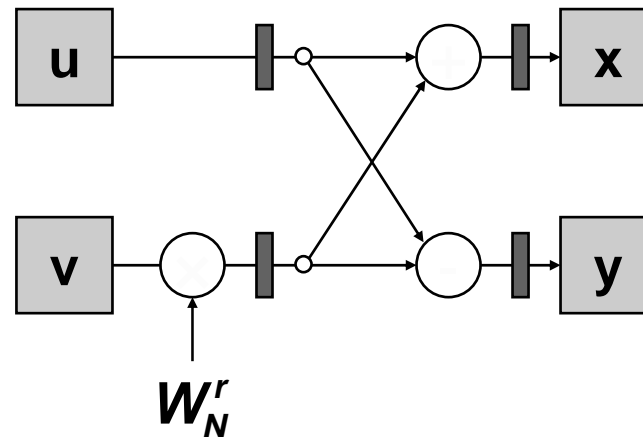
- Butterfly structure
- Removes redundant calculation



Complex Butterfly

- **Butterfly contains**
 - 1 complex addition
 - 1 complex subtraction
 - 1 complex, constant multiply

Size	16	8192	Δ
Pins	448	229K	
Fly	32	53K	
Mult			
Add			
Shift	0	0	



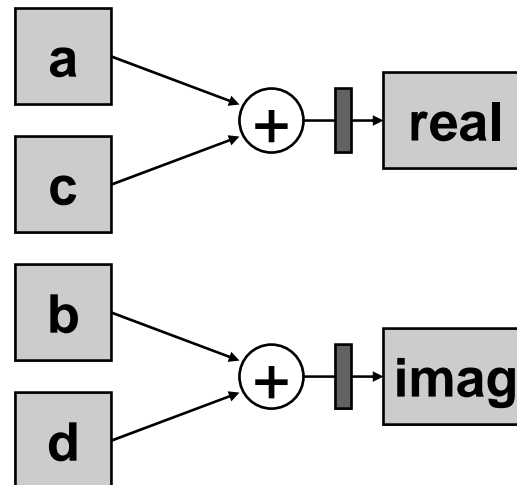


Complex Addition

- Complex addition adds the real and imaginary parts separately:

$$(a + jb) + (c + jd) = (a + c) + j(b + d)$$

Size	16	8192	Δ
Pins	448	229K	
Fly	32	53K	
Mult			
Add	128	213K	
Shift	0	0	



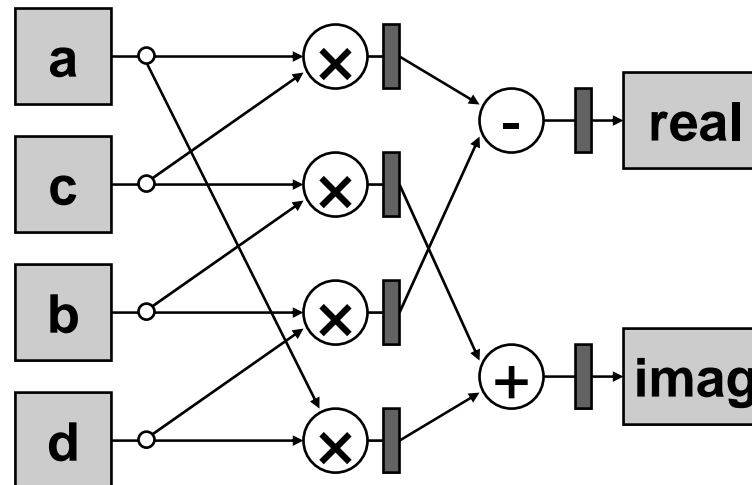


Complex Multiply

- The FOIL method of multiplying complex numbers:

$$(a + jb)(c + jd) = (ac - bd) + j(ad + bc)$$

4 multiplies and 2 adds



Size	16	8192	Δ
Pins	448	229K	
Fly	32	53K	
Mult	128	213K	
Add	192	320K	
Shift	0	0	



Efficient Complex Multiply

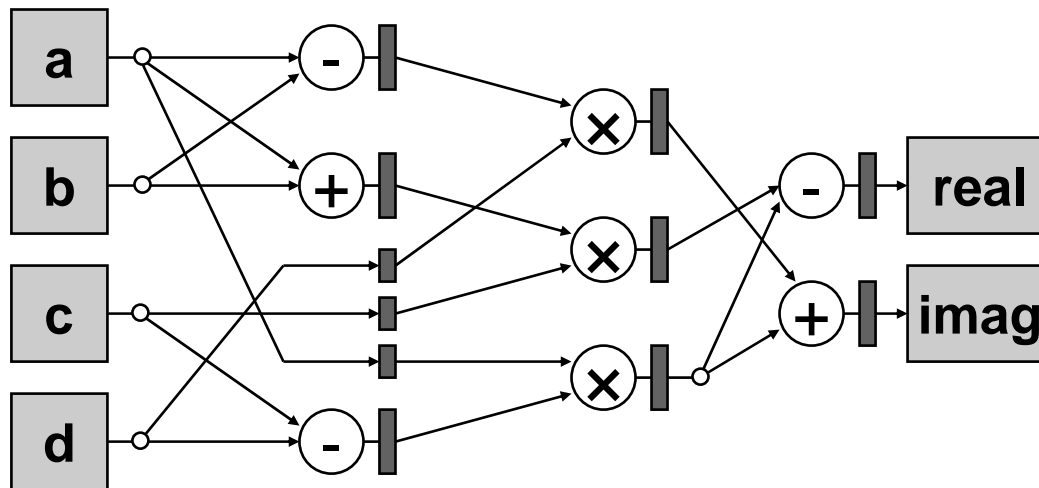
Size	16	8192	Δ
Pins	448	229K	
Fly	32	53K	
Mult	96	159K	75%
Add	200	400K	150%
Shift	0	0	

- Another approach requires fewer multiplies:

$$(ad + bc) = c(a + b) - a(c - d)$$

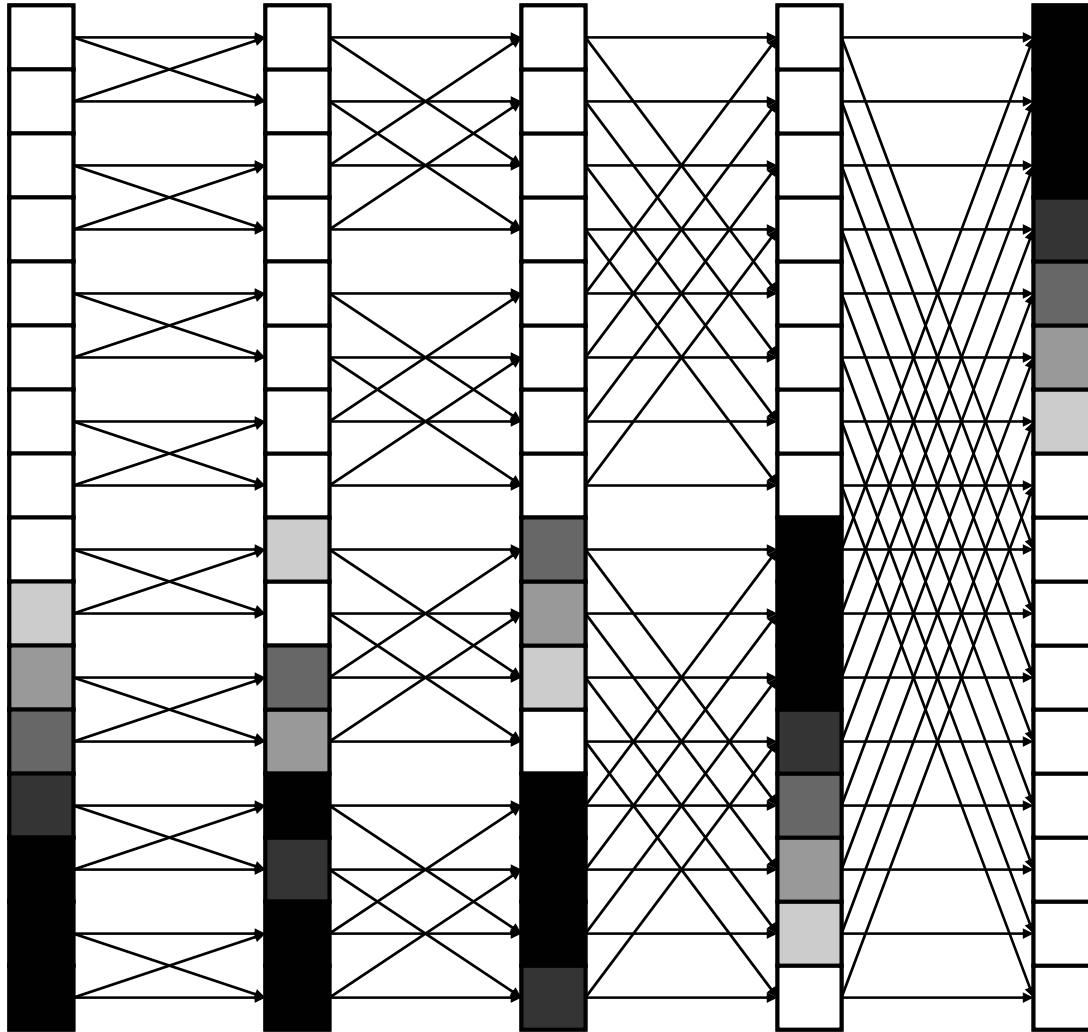
$$(ac - bd) = d(a - b) + a(c - d)$$

3 multiplies and 5 adds





Parallel-Pipelined Architecture



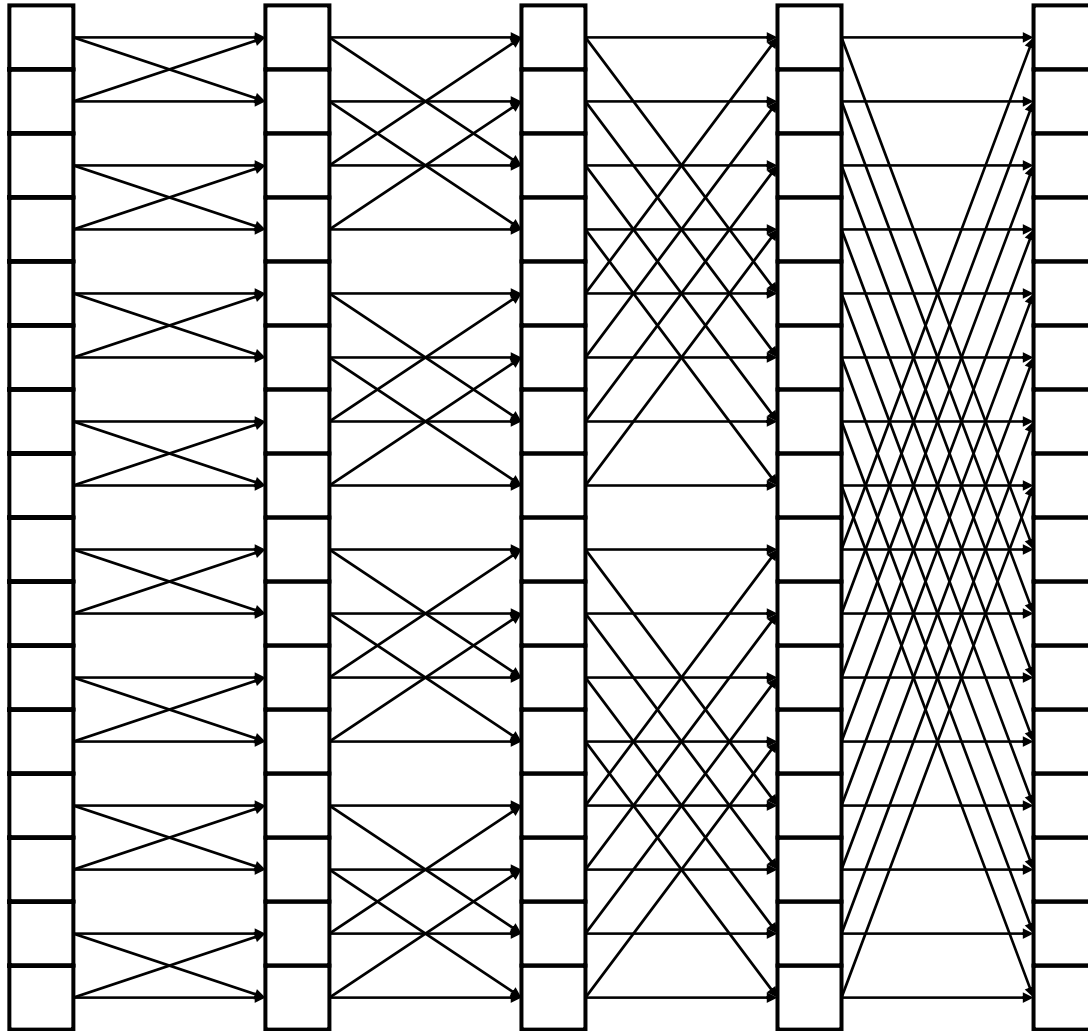
Size	16	8192	Δ
Pins	448	229K	
Fly	32	53K	
Mult	96	159K	
Add	288	480K	
Shift	0	0	

A pipelined version

- **IO Bound**



Serial Input



Size	16	8192	Δ
Pins	28	28	.01%
Fly	32	53K	
Mult	96	159K	
Add	288	480K	
Shift	0	0	

A serial version

- **IO-rate matches A/D**

• **25% Efficient**



Outline

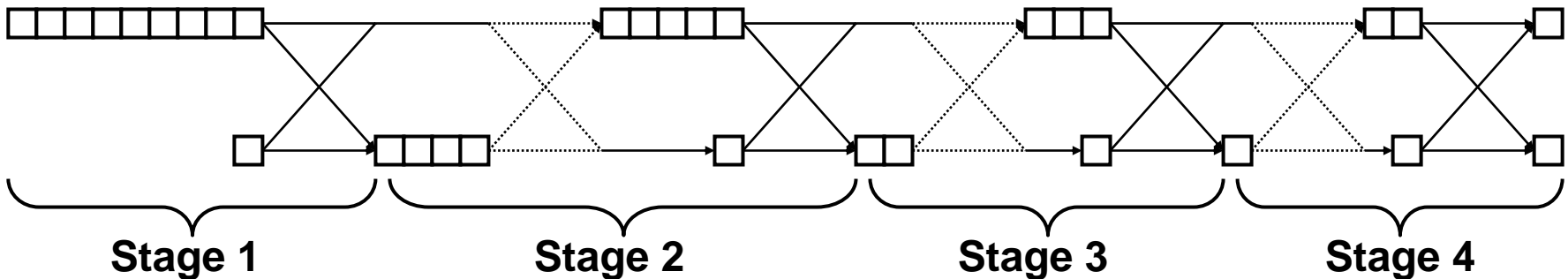
- **Introduction**
- **Parallel architecture**
- **Systolic architecture**
 - **Serial implementation**
 - **Application specific optimizations**
- **Performance summary**
- **Conclusions**



Serial Architecture

- **The parallel architecture can be collapsed**
 - **One butterfly per stage**
 - **Consumes 1 sample per cycle**
 - **Same latency and throughput**
 - **More efficient design**

Size	16	8192	Δ
Pins	28	28	
Fly	4	13	.03%
Mult	12	39	.03%
Add	36	117	.03%
Shift	22	128	



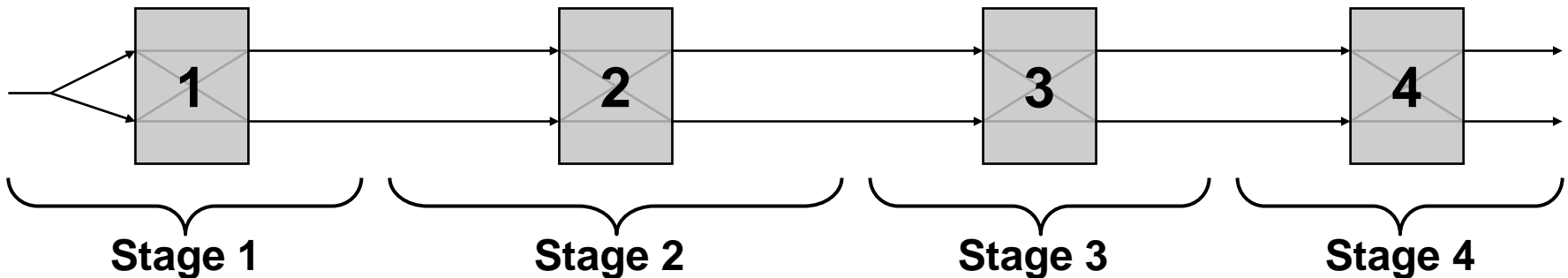
50% Efficiency



High Level View

- Replace complex structure with an abstract cell which contains:
 - FIFOs
 - Butterfly
 - Switch network

Size	16	8192	Δ
Pins	28	28	
Fly	4	13	
Mult	12	39	
Add	36	117	
Shift	22	12K	

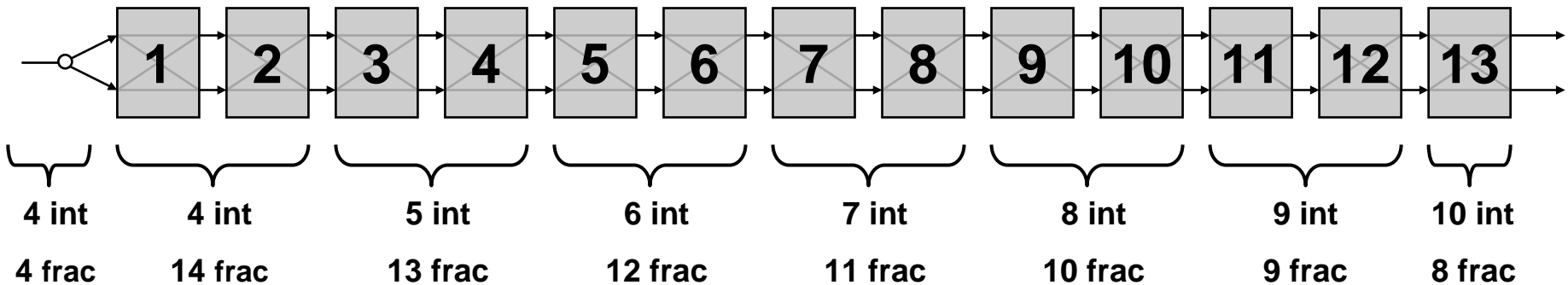




8192-Point Architecture

Size	16	8192	Δ
Pins	28	28	
Fly	4	13	
Mult	12	39	
Add	36	117	
Shift	22	12K	

- Requires 13 stages
- Fixed point arithmetic
- Varies the dynamic range to increase accuracy
- Overflow replaced with saturated value



- Multipliers limit design to 18-bits and 150 MHz
- Achieves 70 dB of accuracy

$$\underbrace{0110.0101}_{6} + \frac{5}{16}$$

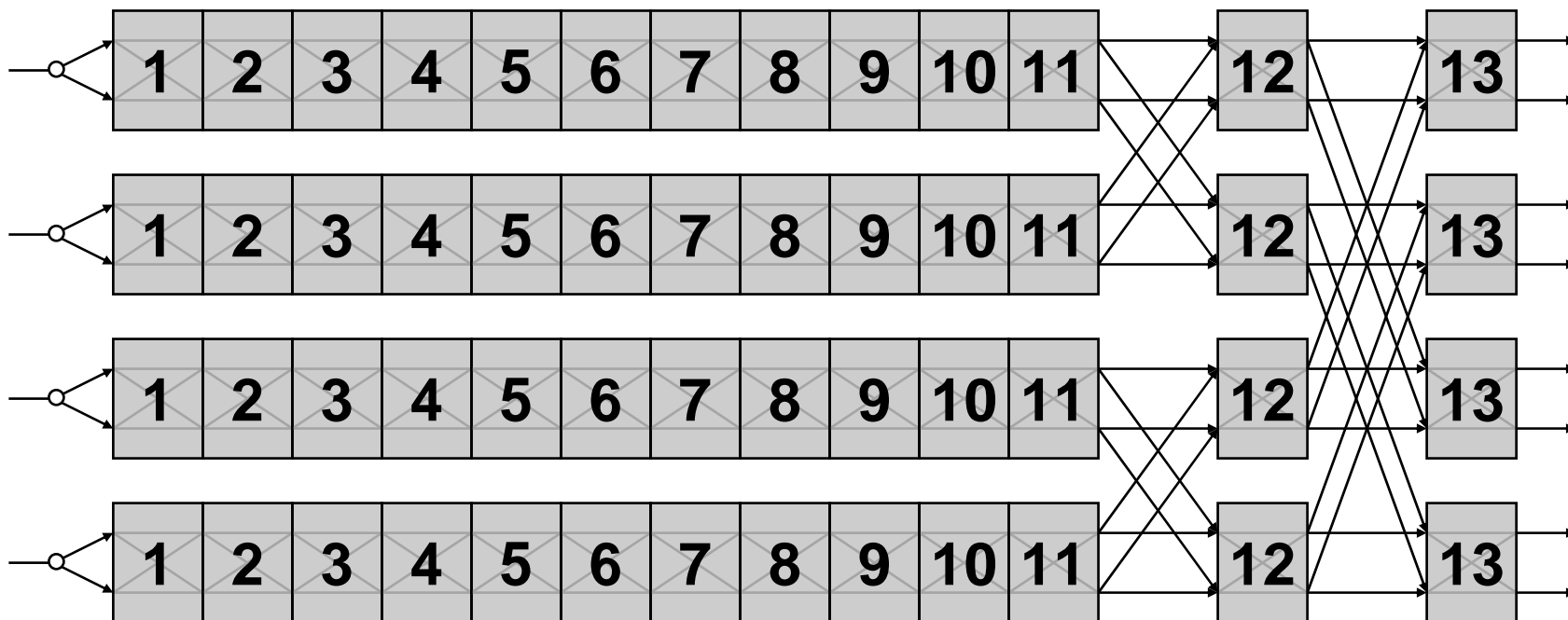


Increase Parallelism

Add more pipelines

- Design limited to 150 MHz by multipliers
- I/Q module generate 600 MSPS
- Meets real-time requirement through parallelism

Size	16	8192	Δ
Pins	112	112	100%
Fly	150	150	100%
Mult	150	150	100%
Add	150	150	100%
Shift	16	12K	100%



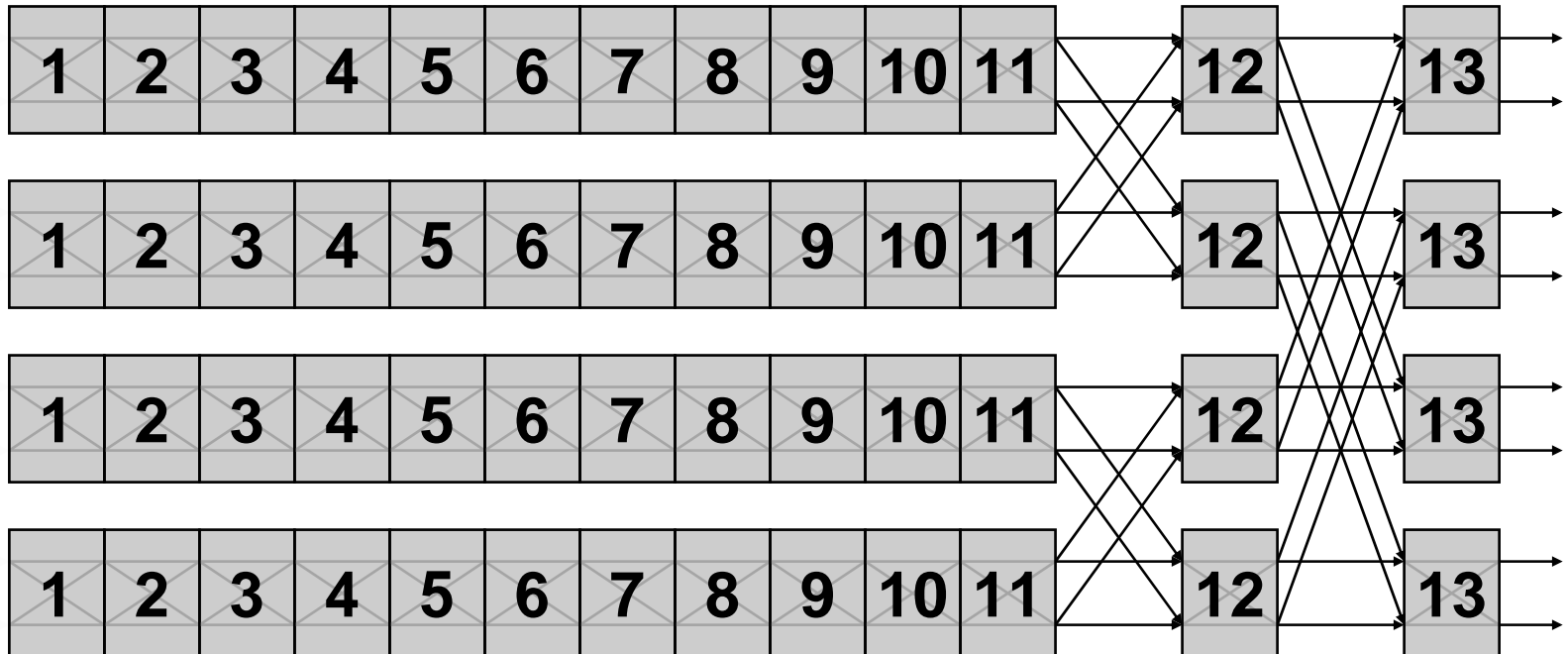


Simplification

Target application allows a specific simplification

- Pads a 4096-point sequence with 4096 zeros
- Removes 1st stage multipliers and adders
- Achieves 100% efficiency in steady state

Size	16	8192	Δ
Pins	160	160	0%
Fly	16	52	
Mult	36	144	92%
Add	108	432	92%
Shift	4	8K	67%





Outline

- **Introduction**
- **Parallel architecture**
- **Systolic architecture**
- • **Performance summary**
 - **Power, operations per second**
 - **FPGA resources, frequency**
 - **Latency, throughput**
- **Conclusions**



Results

The current implementation has been placed on a Virtex II 8000 and verified at 150 MHz

- **Power: 22 Watts @ 65 C**
- **GOPS: 86 total @ 3.9 GOPS/Watt**

- **FPGA resources (XC2V8000)**
 - **Multipliers: 144 (85%)**
 - **LUTs and SRLs: 39,453 (42%)**
 - **BlockRAM: 56 (33%)**
 - **Filp flops: 35,861 (38%)**

- **Frequency: 150 MHz**
- **Latency: 1127 cycles**
- **Throughput: 1.2 GSPS**



Outline

- **Introduction**
- **Parallel architecture**
- **Systolic architecture**
- **Performance summary**
- **Conclusions**
 - **Applicability to other platforms**
 - **Future work**



Conclusions

- **Created a high performance, real-time FFT core**
 - Low power (3.9 GOPS/Watt)
 - High throughput (1.2 GSPS), low latency (7.6 μ sec/sample)
 - Fixed-point (18-bits), high accuracy (70 dB)
- **General architecture**
 - Extendable to a generic FPGA core
 - Retargetable to ASIC technology
- **Future work**
 - Develop a parameterizable IP core generator