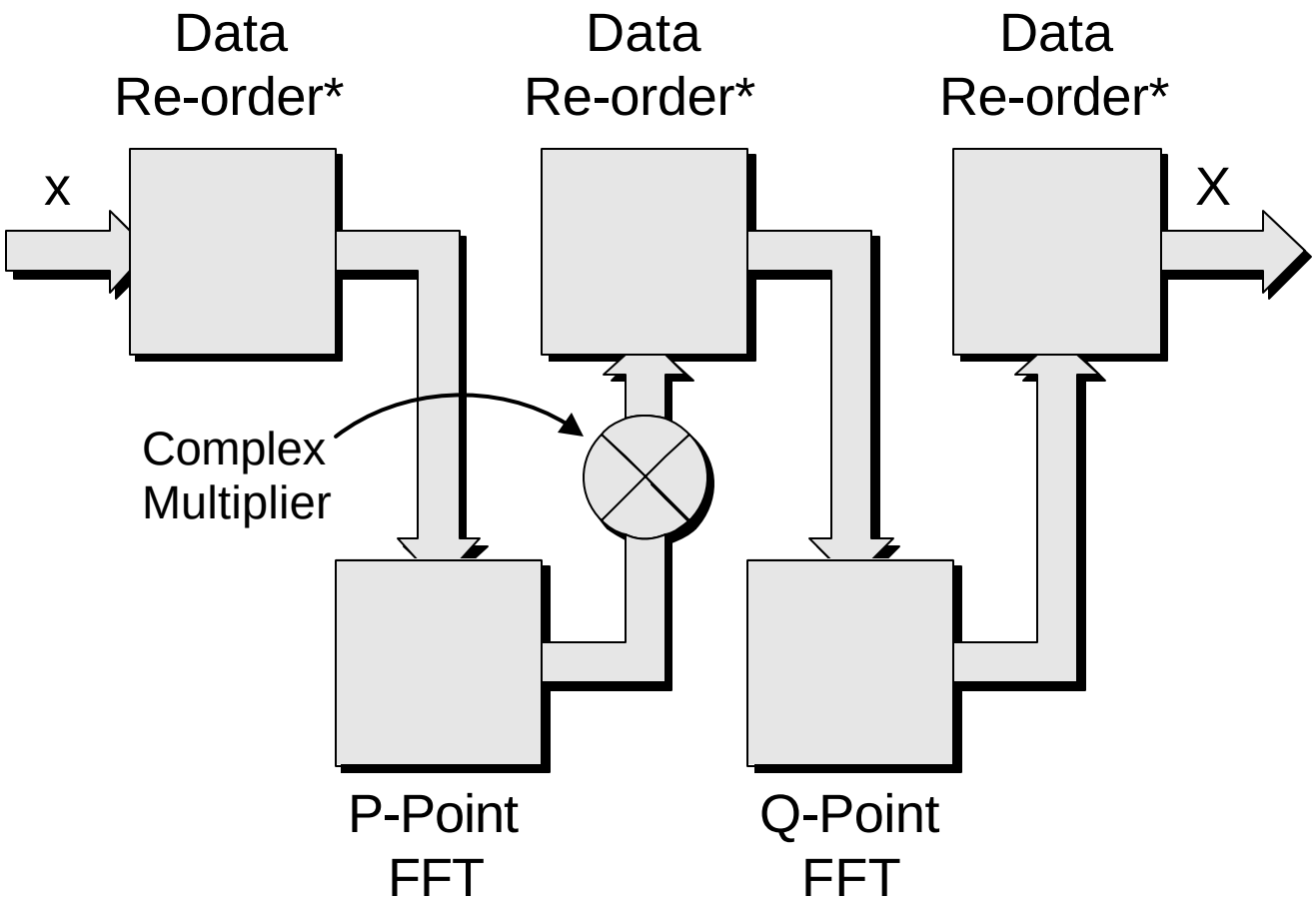
Implementing Efficient Split-Radix FFTs in FPGAs

- Radix-2 and Radix-4 FFTs are common
- Many applications benefit from other lengths
- OFDM Transceiver, Digital Video Broadcasts, and software defined radios often require FFTs that aren't a radix-2 or radix-4 length
- Split-radix simplifies the logic for these applications
- Common Split-radix algorithms are Cooley-Tukey, Kolbe-Parks, and Good-Thomas

Split-Radix FFTs

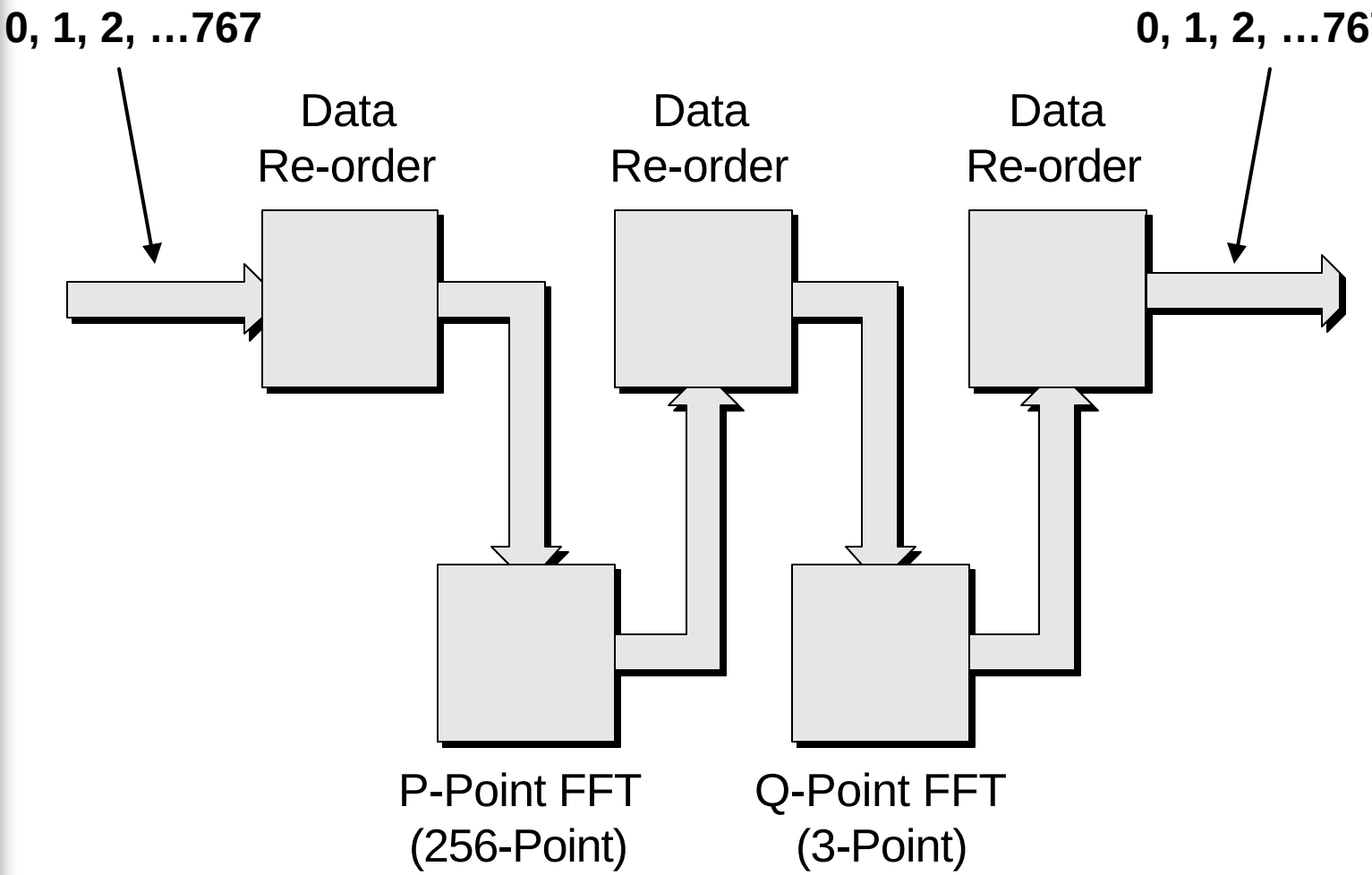
- Split-radix refers to combinations of two (or more) FFT engines
- Split-radix FFTs have a similar structure to 2D FFTs
- Split-radix FFTs provide bin spacing that produce better results for many applications
- Two split-radix approaches employed by DE:
  - Serial (traditional) – lower performance, higher memory requirements by using serial versions of both FFTs
  - Parallel – higher performance by placing larger radix FFTs in parallel and using a parallel version of the smaller radix FFT
- Parallel version mainly used when combining a larger FFT with a 3 or 5 point FFT, since it is feasible to use 3 or 5 large FFTs in a single device

Traditional Serial Split-Radix Approach

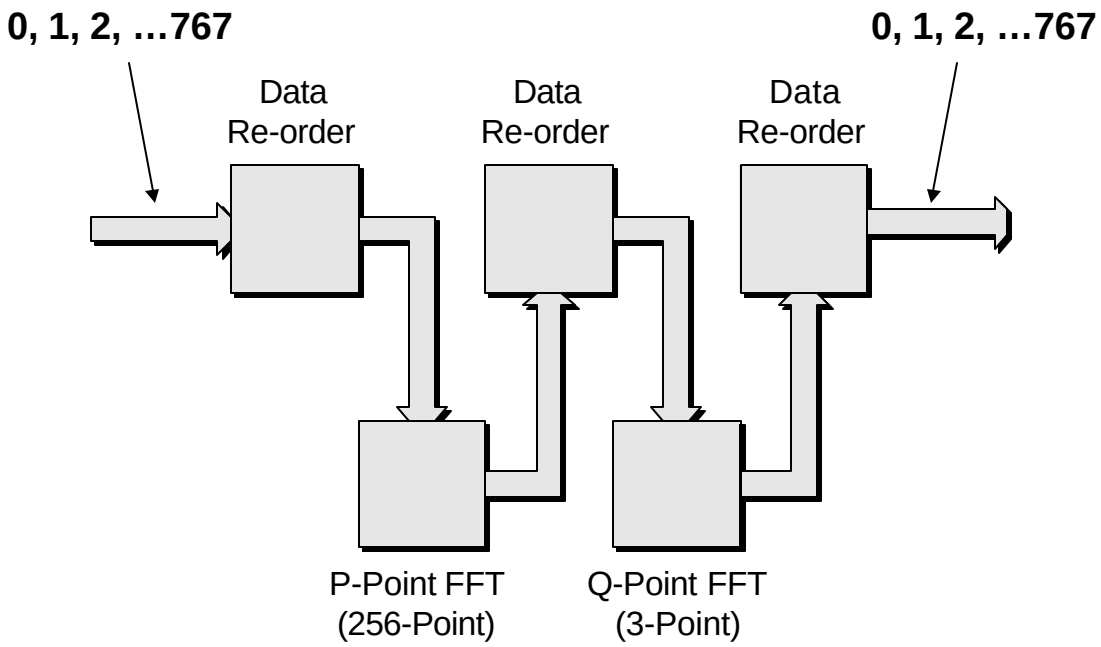


***Continuous data FFTs require enough memory to store two full copies of the data for each re-order stage**

Serial 768-Point Split-Radix FFT

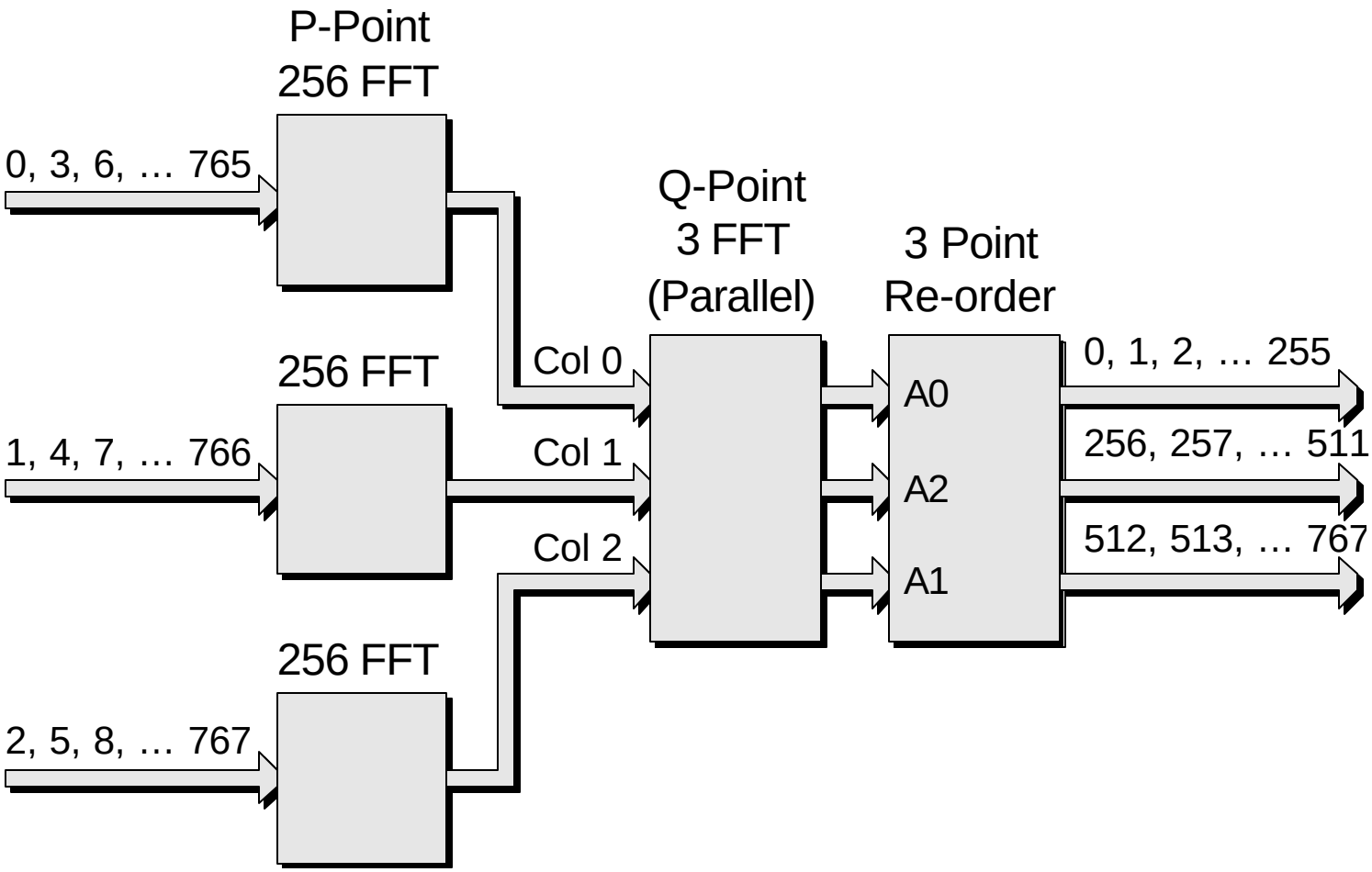


Serial 768-Point Split-Radix FFT (cont.)

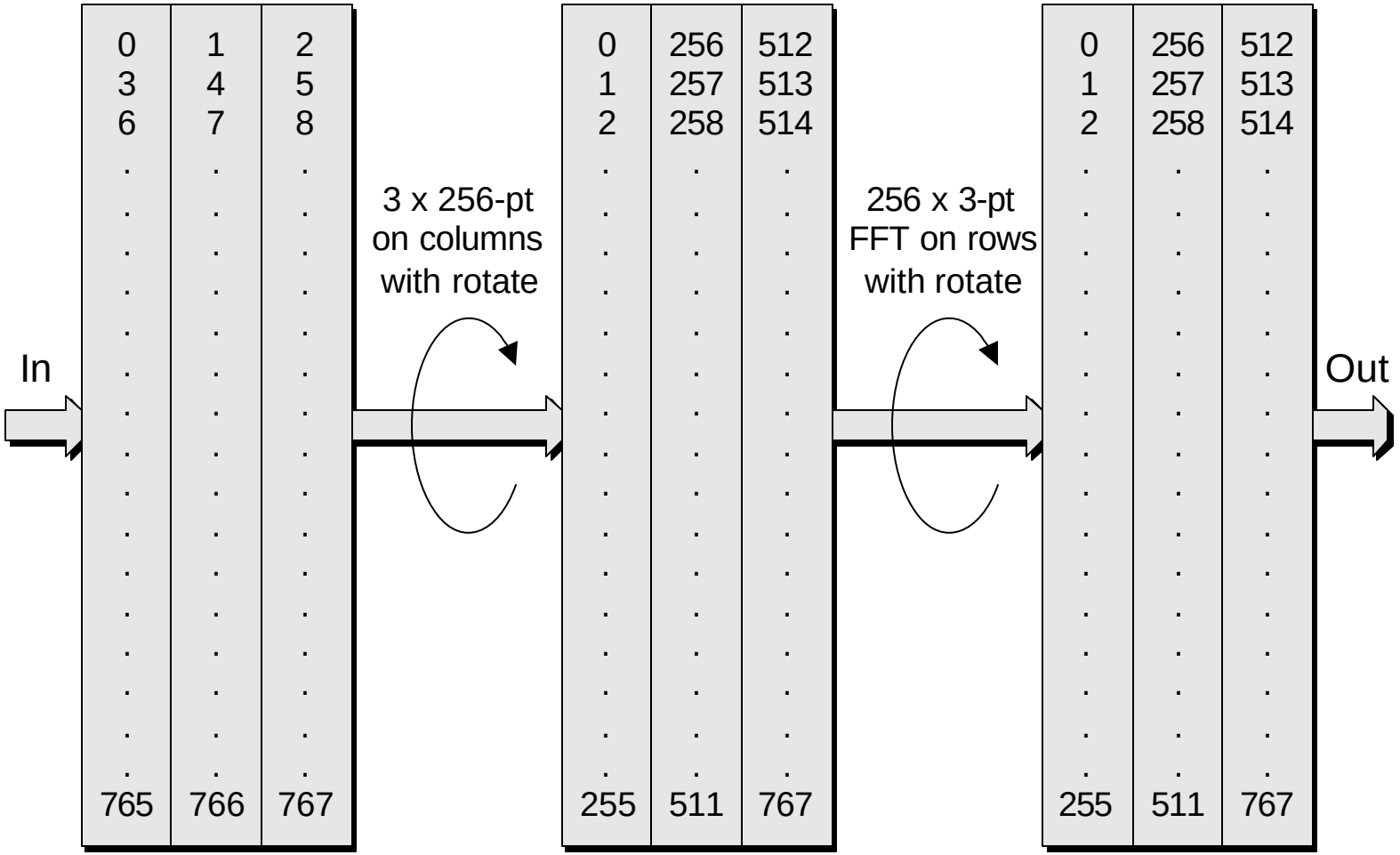


- Single engine of each radix (256-point FFT followed by 3-point FFT)
- Lower device utilization, with performance suitable for most applications
- High memory requirements for data re-ordering
- Speeds up to continuous data, slower data rates require less logic
- Same structure (with external memory) used for ultra-long FFTs

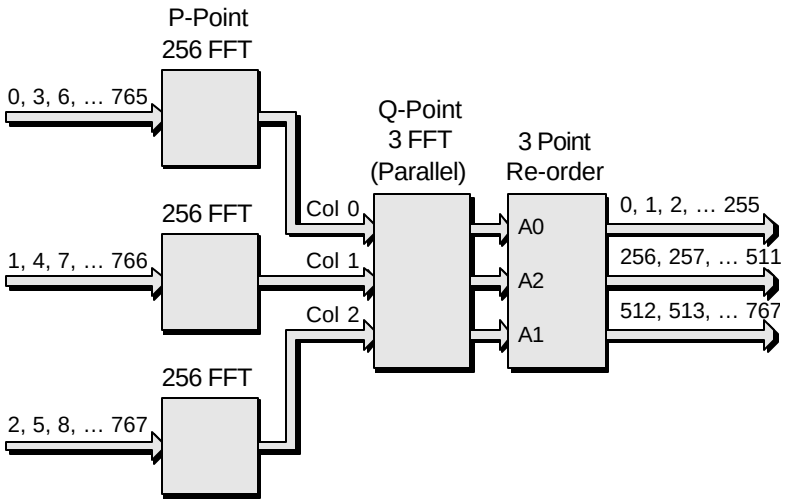
Parallel 768-Point Split-Radix FFT



Parallel 768-Point Split-Radix FFT Data Flow



Parallel 768-Point Split-Radix FFT (cont.)



- Combines 256-point FFT with 3-point FFT
  - 3 x 256-point FFT executions
  - 256 x 3-point FFT executions
- Eliminates the need for intermediate memory
- Higher resource (logic) usage as more computations are performed in parallel
- Very high performance – perform a new 768-point FFT every 256 clock cycles (1.7uS @ 150 MHz)

Virtex II Performance

Type	Virtex II Performance @ 150 MHz (18-bit Complex I/O)							
	Number of Butterflies	Latency (uS)	FFT Rate (uS)	Sizes	Block RAM	Multipliers	Power (mW)	Cost ($)
Serial	1	20.67	20.42	3,562	12	4	756	290
Serial	4	15.67	5.12	5,224	18	16	969	500
Parallel	3	6.96	6.83	4,180	24	12	974	350
Parallel	6	3.54	3.41	6,260	45	24	1,331	700
Parallel	12	1.84	1.70	11,123	75	48	1,840	1,400

- **Latency: Time from last point in to first out**
- **FFT rate: Rate to input FFT data sets**
- **Power: Estimate via Xilinx XPower**
- **Cost: Based on single piece XC2V3000-6 from Partminer.com**

Other Dillon Engineering Resources

- ParaCore Architect (parameterized core builder)
- DSP Algorithms
  - Ultra-long FFTs (2k x 2k = 4M points)
  - 2D FFTs for image processing
  - Fixed or floating-point FFTs
  - Floating point math library
- System level DSP
  - OFDM Transceivers
  - Radar Processing on single FPGA
  - Image Compression/Processing
- FPGA-based DSP development platforms
- Hareware/Software SOC
  - High speed Ethernet Appliances
  - Linux Based SOC in FPGA
  - MicroBlaze application