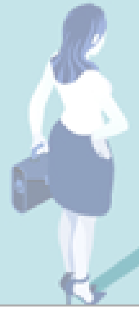
WINTECHCON 2018

Dual-Purpose Hardware Accelerator to implement a High throughput FFT and Sorting Engine

Indu Prathapan, Pankaj Gupta

Texas Instruments



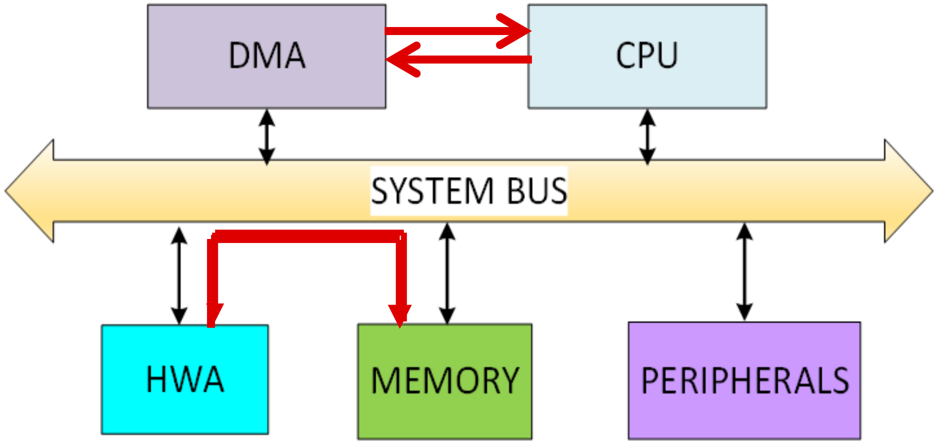
OUTLINE

- ❑Motivation – Need for Hardware Accelerators (HWA)
- ❑Overview of R2SDF FFT hardware architecture
- ❑Bitonic Sorting Network
- ❑Proposed Methods for
  - ❑Dual-purpose FFT and Sorting Hardware Accelerator architecture
  - ❑Improved parallelism for combined HWA to achieve a 4X throughput
- ❑Results
- ❑Conclusion

Need for Hardware Accelerators

- ❑Hardware acceleration is the use of dedicated hardware to perform some functions more efficiently than is possible using software running on a general-purpose CPU
  - High performance, Low power

- ❑Performance of the core processor can also be speeded up many times if computationally-intensive operations and repeated functions are delegated to dedicated hardware accelerators
  - Allows the processor to focus on other general-purpose tasks.
  - Boosts up the effective computational throughput of the processor
  - Improved the system performance



Optimized Hardware Accelerators

❑Ideal candidates for Hardware Acceleration

- ❑ Time critical functions that require high-throughput
- ❑ Computationally-intensive operations
- ❑ Repeated functions
- ❑ Architectures with regular structure : direct implementation in hardware

- ✓ Digital signal processing functions such as FFT, FIR and IIR filters, Matrix multiplications
- ✓ Data processing operations such as Sorting , Searching ,Compression..

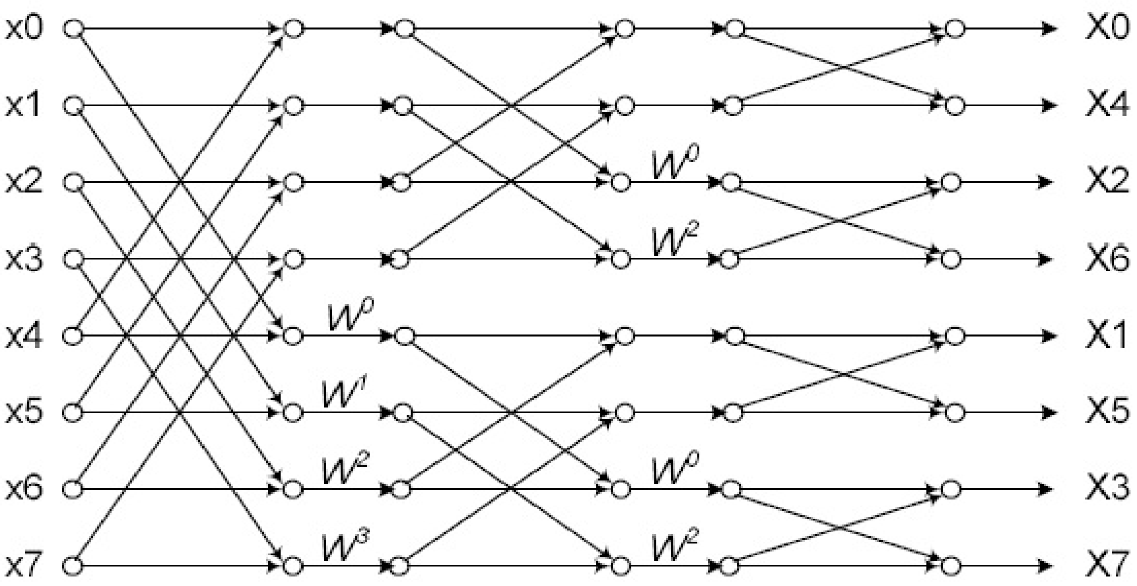
- ⬇ Development Cost for building Custom HWA
- ⬇ Silicon area for extra HWA in addition to on-chip CPU

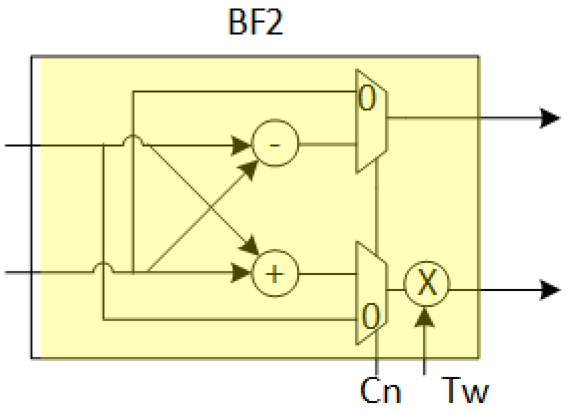
❑ Dual-Purpose Hardware Accelerator to implement FFT and Sorting

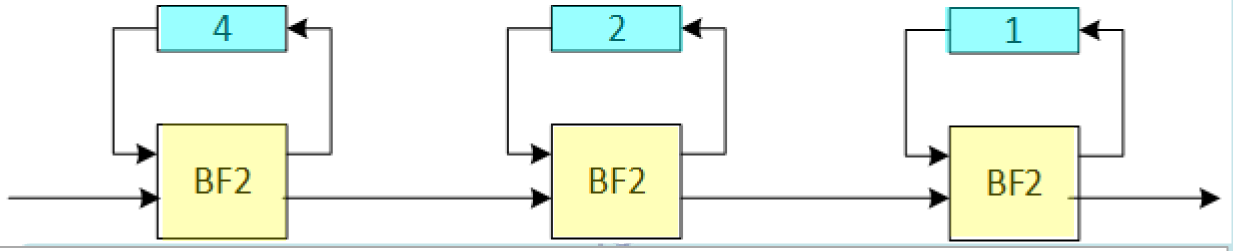
Pipelined FFT Hardware Architecture (R2SDF) - I

RADIX-2 SINGLE DELAY FEEDBACK

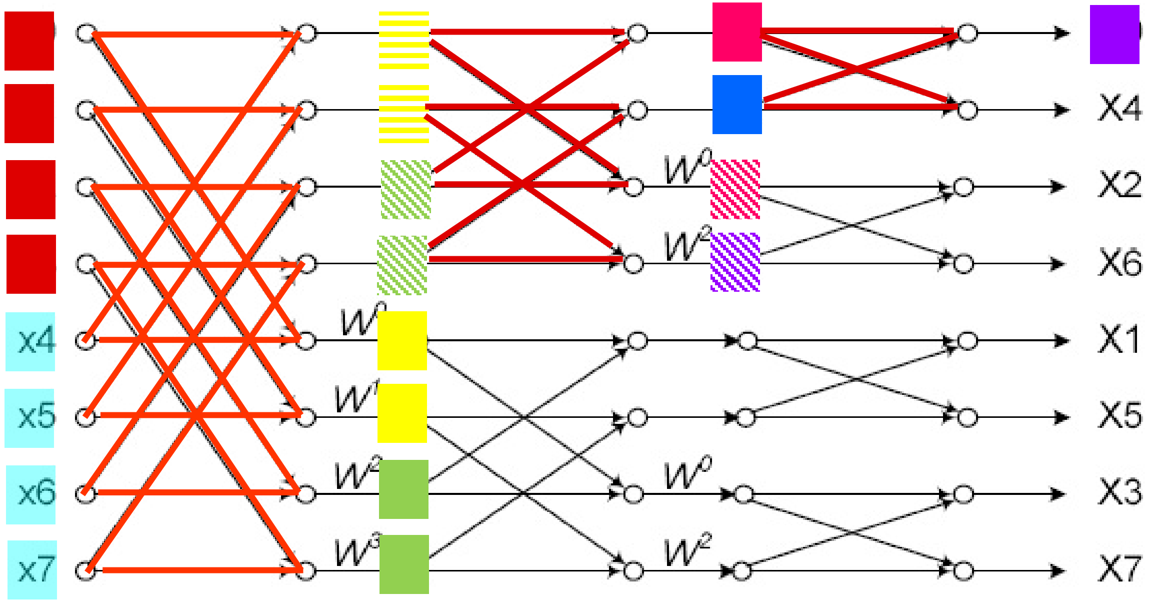
- ❑ log$_2$N pipelined butterfly stages
  - – N is the number of FFT points
- ❑ N-1 delay elements using memory or shift registers in the feedback path across all stages.

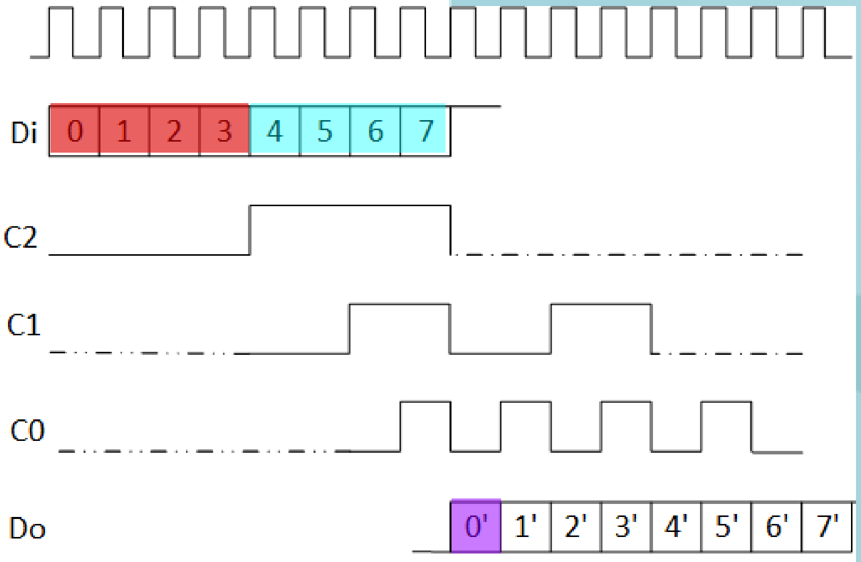


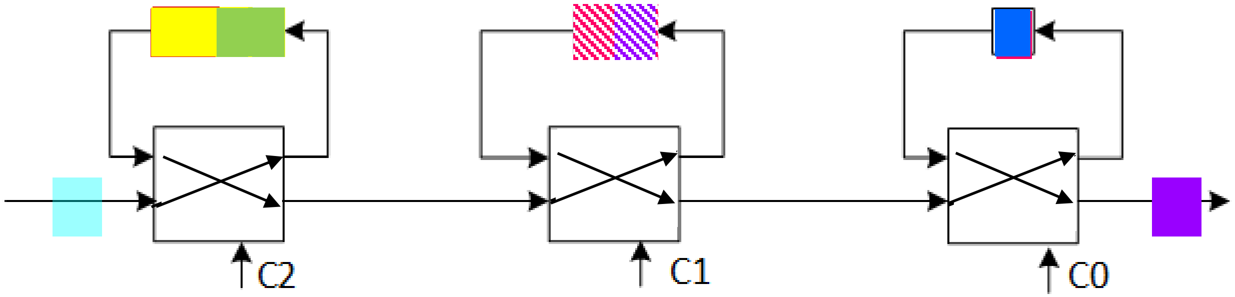




Pipelined FFT Hardware Architecture (R2SDF)- II







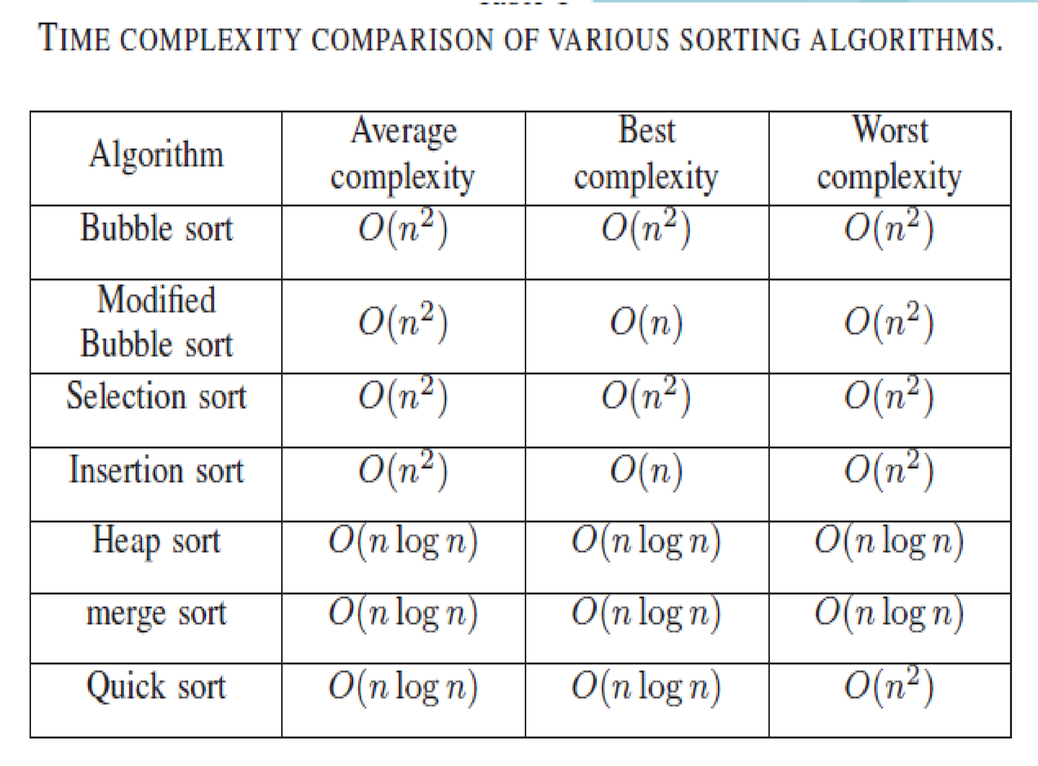
- □ Latency : Θ(N) clock cycles
- □ Throughput : 1 Sample/clocks

Sorting Algorithms and its Complexity

❑ Sorting is one of the key functions performed by computer programs as an internal step for many data and signal processing and computer graphics applications

❑ The lower bound on any comparison-based sort of N numbers is Θ(N log N) on a serial computer.

❑ Optimal (theoretical) complexity that can be achieved with a parallel sorting algorithm using N processors is O(logN)

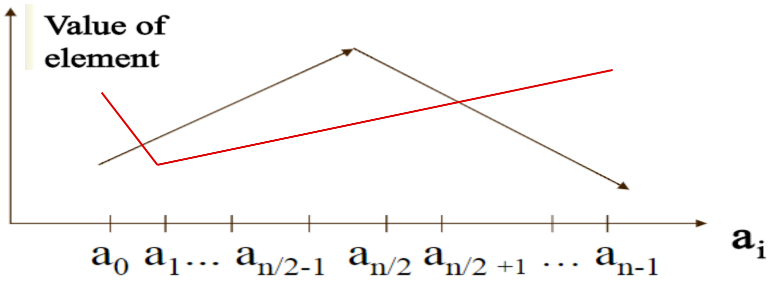


TIME COMPLEXITY COMPARISON OF VARIOUS SORTING ALGORITHMS.

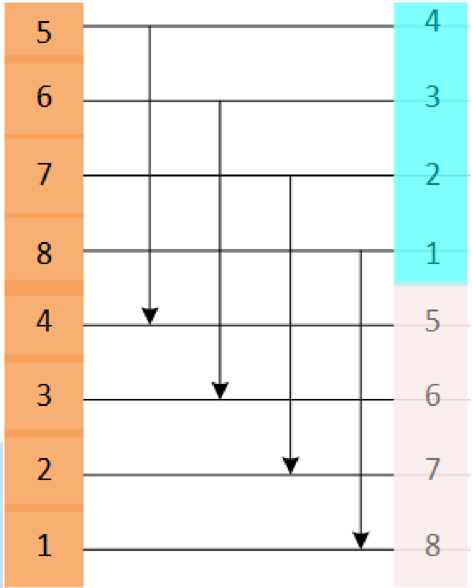
Algorithm	Average complexity	Best complexity	Worst complexity
Bubble sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Modified Bubble sort	$O(n^2)$	$O(n)$	$O(n^2)$
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Insertion sort	$O(n^2)$	$O(n)$	$O(n^2)$
Heap sort	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$
merge sort	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$
Quick sort	$O(n\log n)$	$O(n\log n)$	$O(n^2)$

Bitonic Sequence

- ❑ A bitonic sequence has two tones – increasing and decreasing, or vice versa.
  - Any cyclic rotation of such networks is also considered bitonic.



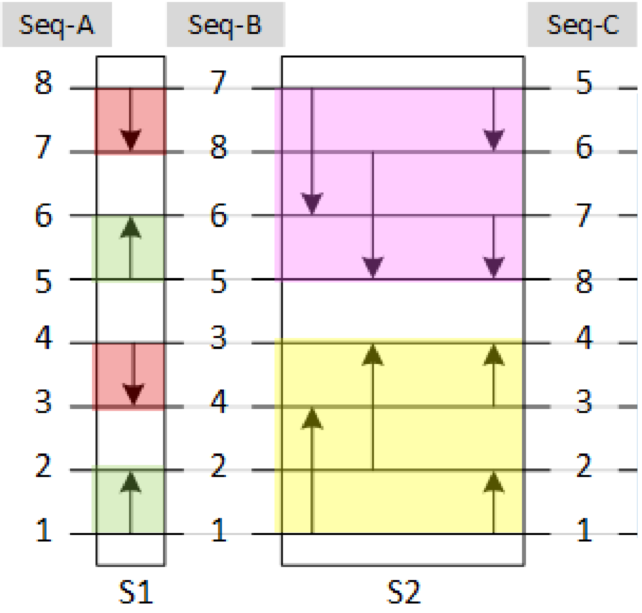
- ❑ If we do a compare-and-exchange operation with elements a_i and $a_{i+N/2}$, in a sequence of size N, we obtain two bitonic sequences in which all the values in one sequence are smaller then the values of the other
  - ➢ The compare-and-exchange operation moves smaller values (tail of arrow) to the top and greater values to the bottom (head of arrow)



- ❑ Given a bitonic sequence, if we apply recursively these operations we get a sorted sequence.

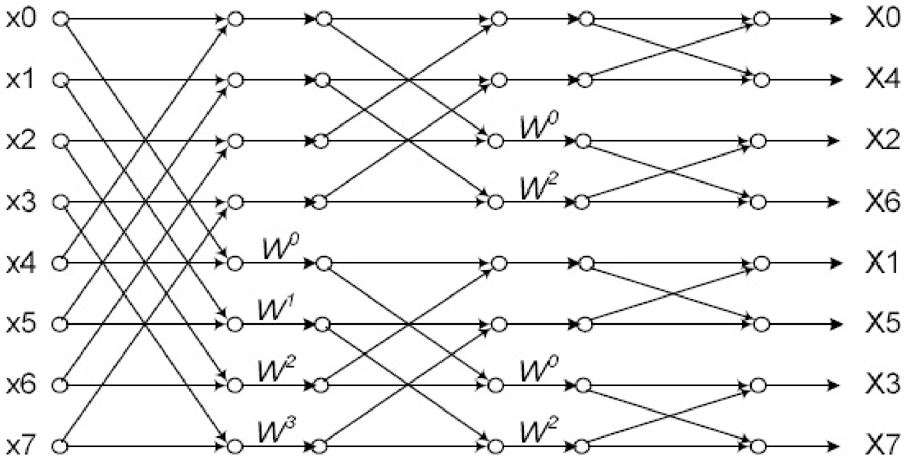
Bitonic Network

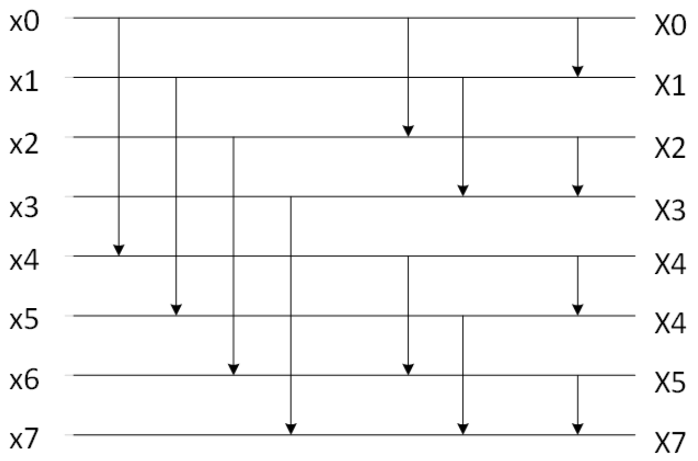
- ❑ A bitonic sorting network consists of the following two operations:
  1. Rearrangement of an unsorted data sequence (Seq-A) into a bitonic sequence (Seq-C). This is performed in the first $\log_2 N$ - 1 stages.
  2. Rearrangement of the Bitonic sequence(Seq-C) into a sorted sequence (Seq-D) is performed in the last stage.

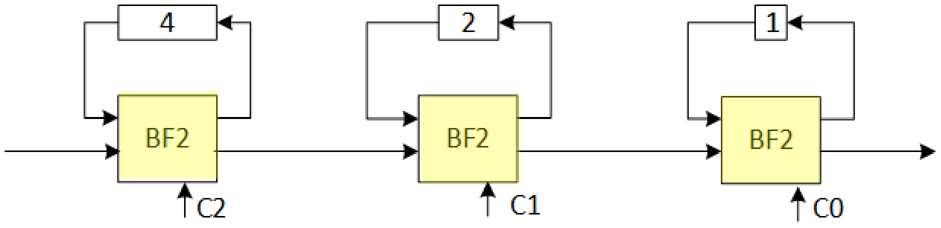


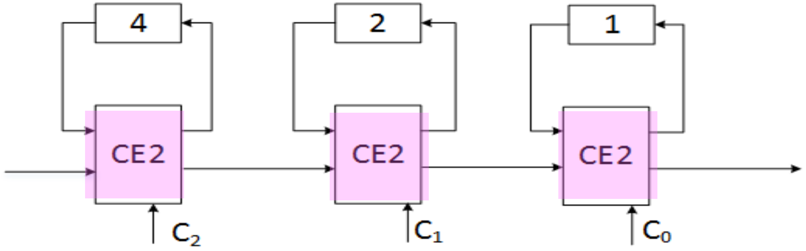
Proposed : R2SDF Implementation of a Bitonic Sorter

- ❑ Spatial regularity of the bitonic sorter and its similarity to FFT’s signal flow graph (SFG) is exploited to map its comparator stages to the butterfly stages of the R2SDF FFT hardware

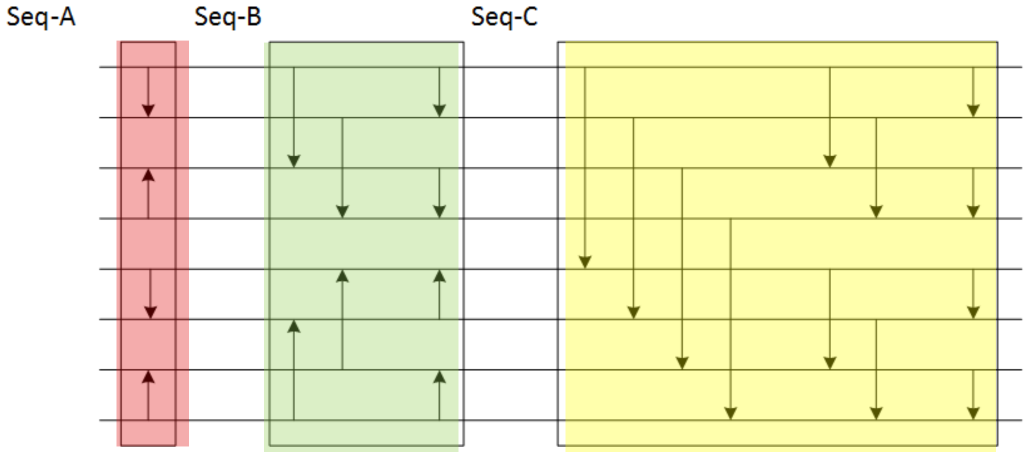


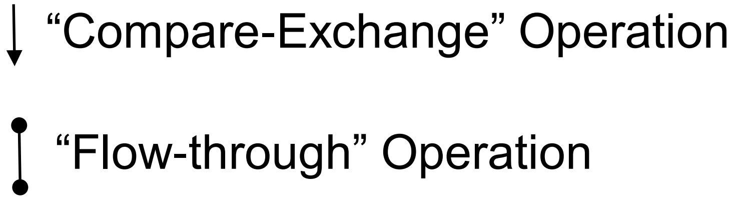


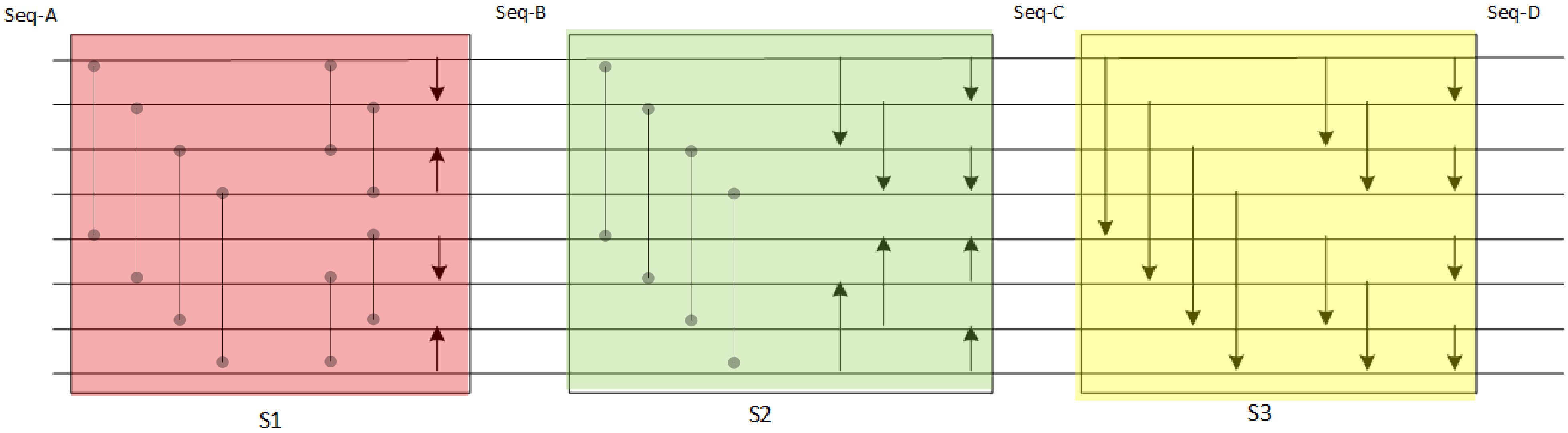




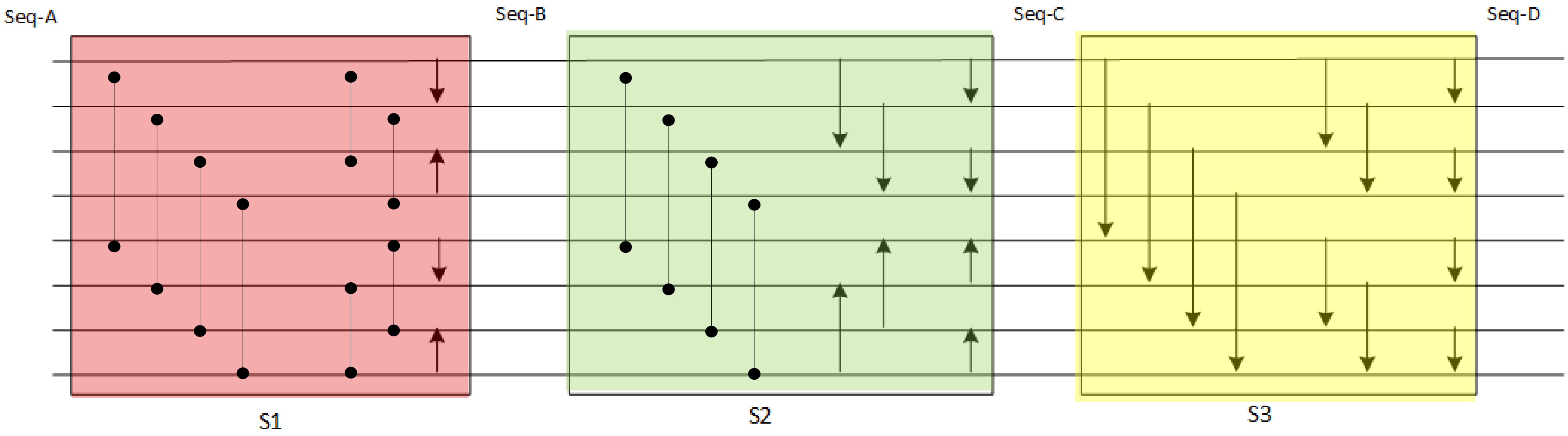
Proposed : Mapping a Bitonic Network to a R2SDF -I

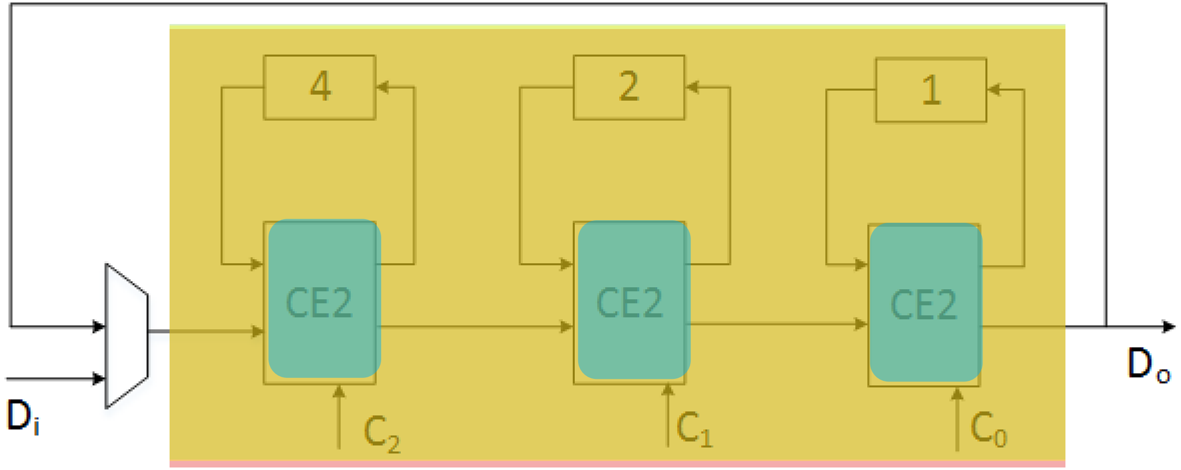


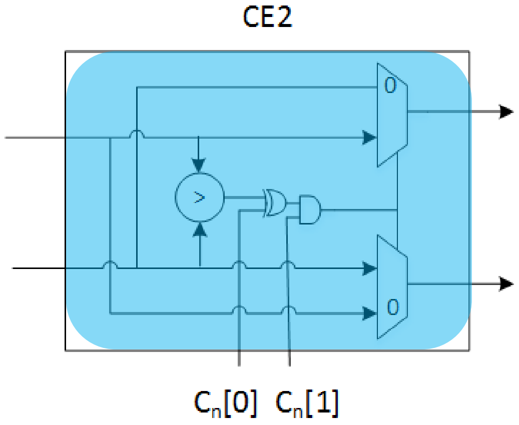




Proposed : Mapping a Bitonic Network to a R2SDF II







Proposed Dual-Purpose HWA : Key features

❑ A R2SDF FFT Engine of size N is converted to a N-sample Bitonic Sorter(BS) by replacing the Radix-2 butterfly (BF2) units with a 2-input compare-exchange units (CE2)

❑ The N-sample BS is used iteratively $\log_2 N$ times to implement an N-sample Bitonic Network.

❑ CE2 unit will have a 2-bit control to additionally specify the “direction” of compare-exchange as described below:

  ❑ 0: Bypass compare-exchange corresponding to "flow-through" operation in SFG. Store the data from the previous stage into its memory. Send the oldest data from its memory to the next stage

  ❑ 2: Compare data from previous stage with the oldest data in its memory. Store the larger data to its memory and pass the smaller data to the next stage.

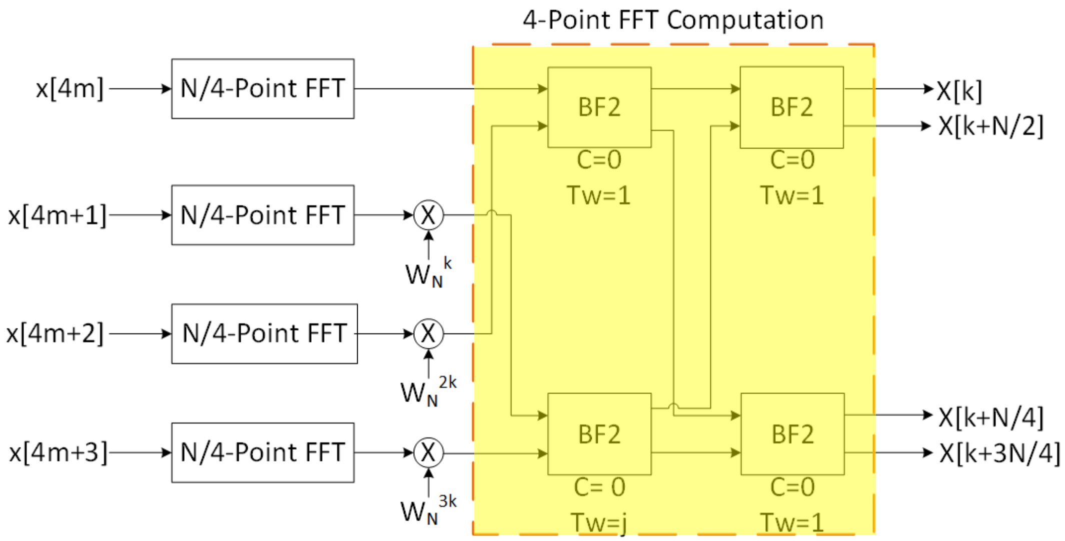
  ❑ 3: Compare data from previous stage to the oldest data in its memory. Store the smaller data into its memory and pass the bigger data to the next stage.

❑ Latency of the sorting engine using the R2SDF architecture is equal to $\Theta(N*\log_2 N)$ clock cycles.

❑ Throughput of the R2SDF sorting accelerator is $\Theta(\log_2 N)$ clock cycles per sample

Improved Parallelism for Dual purpose HWA

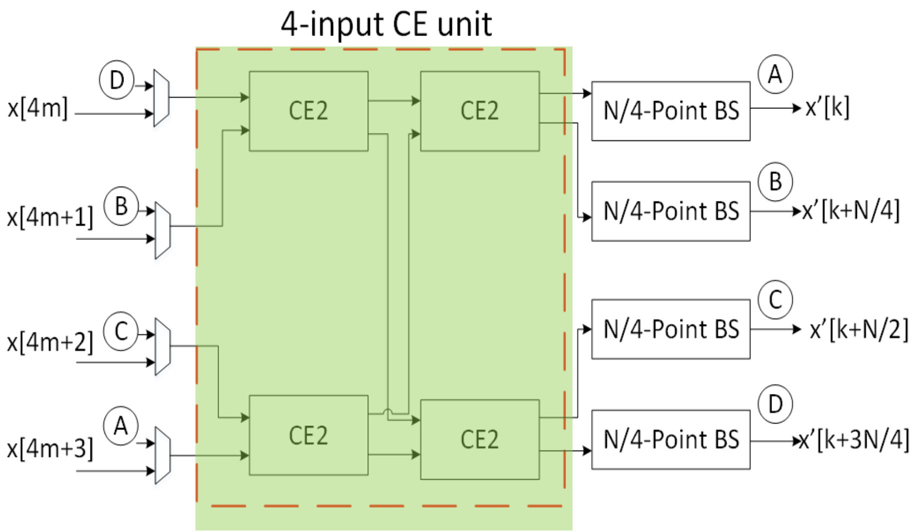
4X R2SDF FFT Engine



Latency : N/4 clock cycles

Throughput : 4 samples / clock

4X R2SDF Sorting Engine



Latency : Θ((N*logN)/4) clock cycles

Throughput : 4 sample /(logN) clock cycles

Experimental Results

Accelerator Throughput Mode	Processing Engine	HWA Area (Sq.mm)	Cycles@400Mhz (N = 4096)
1x	FFT Engine	0.44	4108
	Sorting Engine		49296
4x	FFT Engine	0.98	1034
	Sorting Engine		12408

** 45nm CMOS technology

❑ FFT with complex data bit-width of 24bits and a Sorting engine with real-data bit width of 48bits

❑ The area numbers for the standard cell logic are largely dominated by the complex multipliers in butterfly units for FFT.

❑ The delay elements are implemented as single-port RAMs.

❑ Additional area overhead for implementing a sorting engine is only around 5% of the total accelerator area.

Conclusions

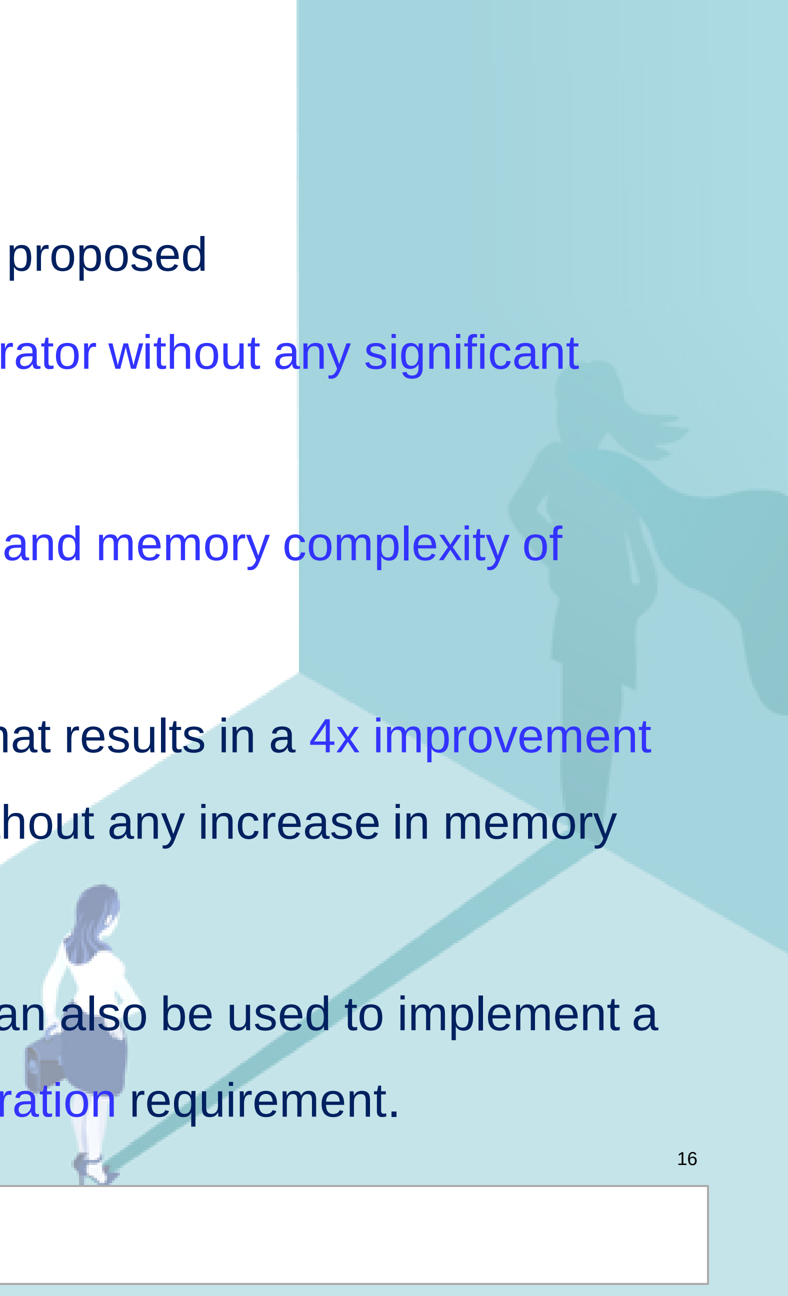
❑ An architecture for dual purpose FFT and Sorting accelerator is proposed

❑ A standard R2SDF FFT engine is re-used as a sorting accelerator without any significant area penalty.

❑ A serial sorting implementation using R2SDF structure has time and memory complexity of Θ(N*log$_2$N) and Θ(N) for sorting an array of size N.

❑ The dual-purpose HWA with a 4X parallelism is also proposed that results in a 4x improvement in the throughput of both the FFT and the sorting accelerators without any increase in memory complexity.

❑ The proposed dual-purpose hardware accelerator architecture can also be used to implement a standalone FFT or sorting accelerator based on the system integration requirement.



Thank You

Q&A ?