# BROOK+ SAMPLE

# Bitonic Sort

## 1 Introduction

Sorting is a fundamental problem in computing. We illustrate parallel bitonic sort on the stream processor using Brook+. Bitonic sort is a data-independent sorting algorithm, where the order of comparison operations does not depend on the input. This makes it a candidate for acceleration by data-parallel implementation.

A *bitonic sequence* is a juxtaposition of two monotonic sequences: one ascending, the other descending. It remains bitonic if it is split anywhere and the two parts are interchanged[1]. Alternatively, a sequence of numbers is bitonic if it has at most one local maximum or one local minimum.

A one-dimensional comparator network, $\mathbf{B}_n$, for a list of $n$ elements, where $n \in N$, can be defined as a sequence of comparison operations. For example, define $\mathbf{B}_n$ as the following sequence of comparisons: $\mathbf{B}_n = [0 : n/2]\ [1 : n/2+1]\ \ldots\ [n/2-1 : n-1]$. Figure 1 shows a diagram of $\mathbf{B}_n$ for $n = 8$.

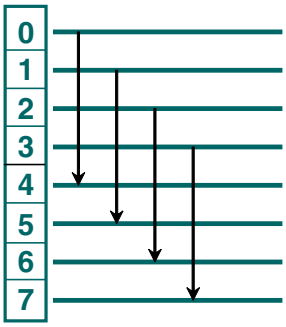


**Figure 1 Comparator Network $B_n$ for $n = 8$**

If $a = a_0, a_1, \ldots, a_n$ is a bitonic sequence, then the application of $B_n$ to $a$ produces two subsequences: $b = b_0, b_1, \ldots, b_{n/2-1}$, and $c = c_0, c_1, \ldots, c_{n/2-1}$ so that all $b_i \leq$ all $c_i$ and both $b$ and $c$ are bitonic sequences. This forms the basis of an iterative algorithm for bitonic sort.

## 2 Bitonic Sorting with Brook+

The bitonic sorting network used for the Brook+ implementation is made up of **log(n)** stages, where **n** is the length of the input array. Figure 2 shows the process diagrammatically.

1. Batcher, K.E.: “Sorting Networks and their Applications”. *Proc. AFIPS Spring Joint Comput. Conf.*, Vol. 32, 307-314 (1968).

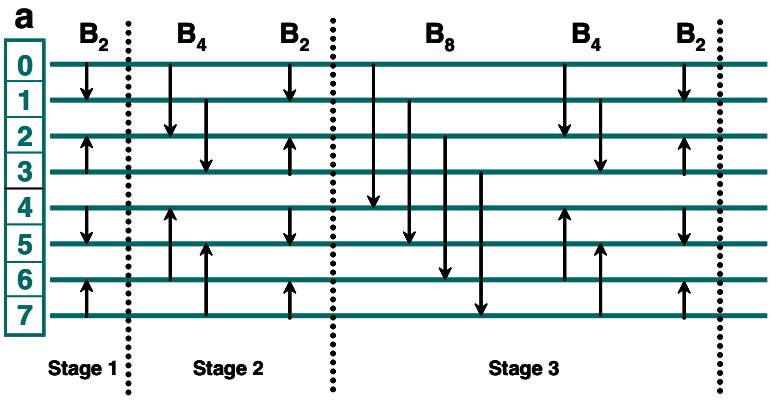


**Figure 2 Bitonic Sorting Based on the Bn Comparator Networks (Input Sequence a has Length 8 and Requires log(8) = 3 Stages for Sorting)**

In Figure 2, the span of each arrow shows the elements that are compared; the direction indicates whether to sort in ascending or descending order. The ***i*th** stage is composed of ***i*** steps, and each step is an application of the comparator network, **Bn**. At the end of stage ***i***, every sub-sequence of length $\mathbf{2}^{i}$ is sorted. During the first stage, sub-sequences of length 2 are sorted alternately in ascending and descending order. At the end of stage 1, **a(0-1)** and **a(4-5)** are sorted in ascending order, whereas **a(2-3)** and **a(6-7)** are sorted in descending order. Note that this results in two sub-sequences of length 4 each (**a(0-3)** and **a(4-7)**) both of which are bitonic. These bitonic sub-sequences are the input to the next stage.

**Contact**

**Advanced Micro Devices, Inc.**
**One AMD Place**
**P.O. Box 3453**
**Sunnyvale, CA, 94088-3453**
**Phone: +1.408.749.4000**

**For Stream Computing:**
**URL: http://ati.amd.com/technology/streamcomputing**
**Questions: streamcomputing@amd.com**
**Developing: streamdeveloper@amd.com**
**Forum: http://forums.amd.com/devforum/categories.cfm?catid=38**