

# Brook+ FAQ

## 1. How do I control multiple GPUs?

See the Multi-GPU Support section of the *Brook+ Programming Guide*.

## 2. What is the difference between: `float a<10,10>;` and `float a<>[10][10];` ?

The first term, `a<X,Y>`, is a 2D stream of size `10x10`; the second is an array of streams (and is not permitted in the current implementation).

## 3. What kind of performance penalty is there if we use a single HD3870 for both the normal 2D display and for running stream computing programs on it (as opposed to having a separate stream-computing-only card)?

The 2D display does not consume substantial GPU resources since the kernel execution uses separated hardware resources. Thus, the performance impact is minimal.

## 4. Does using a FOR loop in a kernel cause the GPU to parallelize computations?

By definition, each kernel executes on a single stream processing pipeline; thus, any for loops are executed in sequential order by the stream processing element. The developer must invoke multiple kernel instances by using a suitable domain of execution to leverage GPU parallelism.

## 5. Are global variables available inside a kernel? If not, do I need to use local within the kernel?

No. Global variables are not available inside kernels. The best way to think about kernels is in terms of the off-load model. If you want CPU-side information to be available to a kernel, it must be supplied as a stream or a scalar argument.

## 6. How do I work around conditional compilation in Brook+?

Conditional compilations are now supported by the `brcc` preprocessor. See the `brcc` Preprocessor section in the *Brook+ Programming Guide*.

## 7. How does the Brook+ runtime handle 2D reductions when the reduction multiple is different for each dimension? For example: `float a <12,12>` and `b <4,3>`.

The runtime handles reduction cases like the one given by using multiple passes. In this case, `a<12,12>` reduces to `t<4,12>` then `t<4,12>` reduces to `b<4,3>`.

## 8. Can I reduce just in the y component?

If the x dimension is the same, then yes. Then, `a<12,12>` to `b<12,3>` is y only.

## 9. How does reduction happen on `float2`, `float4`? How does reduction happen on `float4 <x,y>`?

Short-vectors are handled in parallel.

## 10. Can I also cast float to double, or vice versa, when doing the reduction?

No. The accumulated value must be the same type as the input values; otherwise, the commutativity

requirement is violated.

**11. I am trying to measure the performance of Brook+ samples shipped with the SDK, but it seems that setting BRT\_RUNTIME to cal or cpu does not affect the timing information.**

When using Windows and debugging with Visual Studio, you must restart Visual Studio for the environment variable to take effect. A better way is to specify the environment variable under Project Settings > Debugging options; this avoids having to restart Visual Studio for every change.

**12. I have everything installed, but I cannot do a makefile build (on Windows).**

Visual Studio does not set up the environment variables its tools need. Either open a command shell from inside Visual Studio, or manually add the correct values of INCLUDE and LIB for your system to your environment.

**13. Why do I get corruption of stream elements that are not written to by my kernel?**

A kernel generates a result for all elements in its output stream (or domain of a stream), whether the code specifies a value for that result or not.

Consider this kernel:

```
kernel void greaterThanFour( out float a<10> )
{
    int x = indexof(a);
    if (x > 4)
    {
        a = 1.0;
    }
}
```

Invoking greaterThanFour() modifies all elements of a, not just those with indices above 4. Elements 5 to 9 are set to 1.0, and elements 0 to 4 have an undefined value written to them.

**14. I only need to inspect half of the input data to generate a result, but the half I need is mixed with the half I do not need. Should I copy all the data into a stream and ignore the pieces I don't need, or should I just copy the pieces I need?**

For this kind of ratio, it is better to upload everything. For very sparse data, it may be worth doing a pre-filtering pass on the CPU. (The exact threshold depends on your application and can be determined only by experiment.)

**15. Is there a stream size threshold below which streams are allocated in CPU memory?**

No, when using the GPU runtime, all streams are allocated on the GPU.

**16. I am running lots of kernels and seeing high CPU usage, even though there is no real work for the CPU to do. Why is this?**

The CPU thread is polling the GPU continually to try and launch the next kernel as quickly as possible. If you can tolerate a tiny increase in latency and there is work for the CPU to do in another thread, inserting sleep() operations between kernel invocations can free up additional CPU cycles for other tasks.

**17. How many kernels can be running at one time?**

One kernel per GPU.

**18. Should a boundary check be performed within the kernel code?**

When using the CAL backend in Brook+, all reads that go out of bounds are clamped by the hardware to the boundary pixels. Using the CPU backend causes a buffer overrun, and the results are undefined.

**19. If I try to compile or run my application on Windows, I get one of the following errors.**

**The application failed to initialize properly (0xc0000135).**

**This application has failed to start because the application configuration is incorrect.**

**Reinstalling application may fix this problem.**

**The system cannot execute the specified program.**

The required Visual C++ libraries are not available on your system. The Visual C++ Redistributable Package (VCRedist\_x86.exe, VCRedist\_x64.exe) must be executed on the target system.

Go to the following MSDN links, and install the redistributable package.

For 32-bit SDK: <http://www.microsoft.com/downloads/details.aspx?familyid=200b2fd9-ae1a4a14-984d-389c36f85647&displaylang=en>

For 64-bit SDK: <http://www.microsoft.com/downloads/details.aspx?familyid=EB4EBE2D33C0-4A47-9DD4-B9A6D7BD44DA&displaylang=en>

**20. I get linker errors about missing Brook+ definitions.**

Ensure your application is compiled using a supported C++ compiler (VS8.0, VS8.0SP1, VS9.0, g++4.1.2). Different compilers use different calling conventions. An unsupported compiler may use incompatible calling conventions.