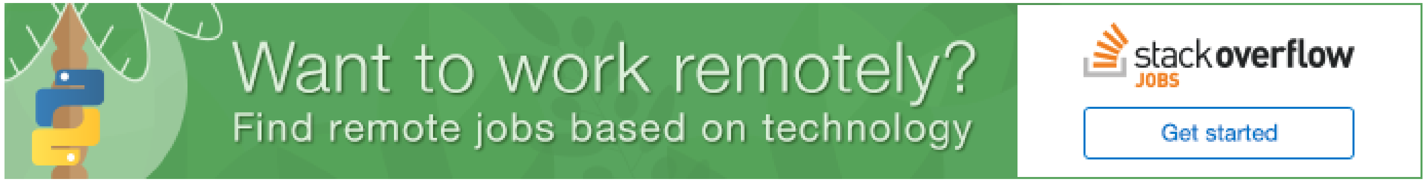
OpenCL: Store pointer to global memory in local memory?



any solutions?

Is that even possible?

```
__global *float abc; // pointer to global memory stored in private memory
```

I want abc to be stored **in local memory** instead of private memory.

c opencl gpgpu

asked Aug 15 '12 at 22:05

user562529
575 1 6 18

1 abc is an address of some storage in global memory. Typically its stored in a register. Do you what to hold this address in a local? Why? Or do you want to copy the data at address abc to local storage? Please clarify. – Tim Child Aug 17 '12 at 3:20

Yes, you got it right: I want to hold this address (abc) in local memory. Why? Maybe to be able to access it from other threads?! (That's what local memory is for, isn't it) – user562529 Sep 21 '12 at 20:54

3 Answers

I think this is clarified here List 5.2:

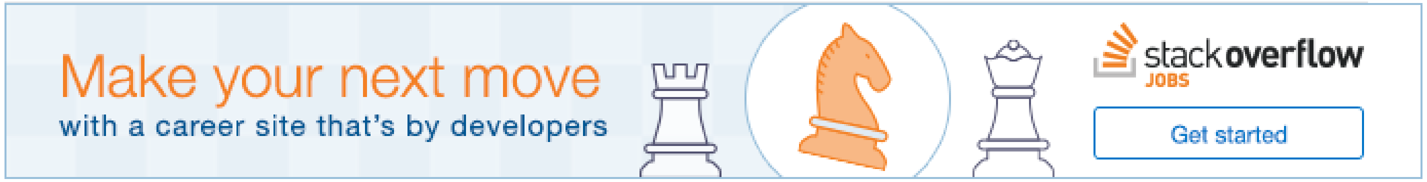
```
__global int global_data[128];  // 128 integers allocated on global memory
__local float *lf;  // pointer placed on the private memory, which points to a
single-precision float located on the local memory
__global char * __local lgc[8];  // 8 pointers stored on the local memory that
points to a char located on the global memory
```

As I understand for pointers: [where they point] type * [where to store] name;

answered Mar 31 '14 at 9:06

kvik
56 3

Haven't verified this but thanks for the reference! This is now the accepted answer. – user562529 Sep 9 '14 at 21:39



Well, I have also encountered a similar problem as you before, as I explored, I think it works in such a way: in OpenCL kernel, variables inside a kernel without an address qualifier (__local, __global, etc) will be automatically considered to be stored in __private memory space. For pointers which are not declared with space qualifier, it will also be considered to point to __private space. Here is an example code snippet:

```
__kernel void foo(__global uint *ptr)
{
    uint tid = get_global_id(0);
    uint * localPtr = ptr;
    uint   var= *(localPtr + tid);
    ... ...
}
```

In the above code snippet, the localPtr is an uint type pointer which has no memory space qualifier, so it will be automatically considered to point to _*private memory space. In OpenCL, a pointer to address space A (*_global, for example) can only be assigned to a pointer to the same address space (__global). So in the above example, a pointer to __private address space cannot be assigned with a pointer (ptr) which points to __global address. (For this point, you can also refer to OpenCL Spec in the section "Address Space Qualifiers"). Hope this can be helpful!

answered Sep 19 '12 at 8:18

acekiller

108 11

Sorry, your answer is not answering my question, but thanks anyway. :) (Your description is correct, though) – user562529 Sep 21 '12 at 20:59

This is legal:

```
global uint *globalPtr=&ptr[tid]
```

answered Mar 19 '13 at 7:36

user2185463

11 1