OpenACC

Lecture Objectives:

Yet another way to program parallel machines . . .

Where we came from…

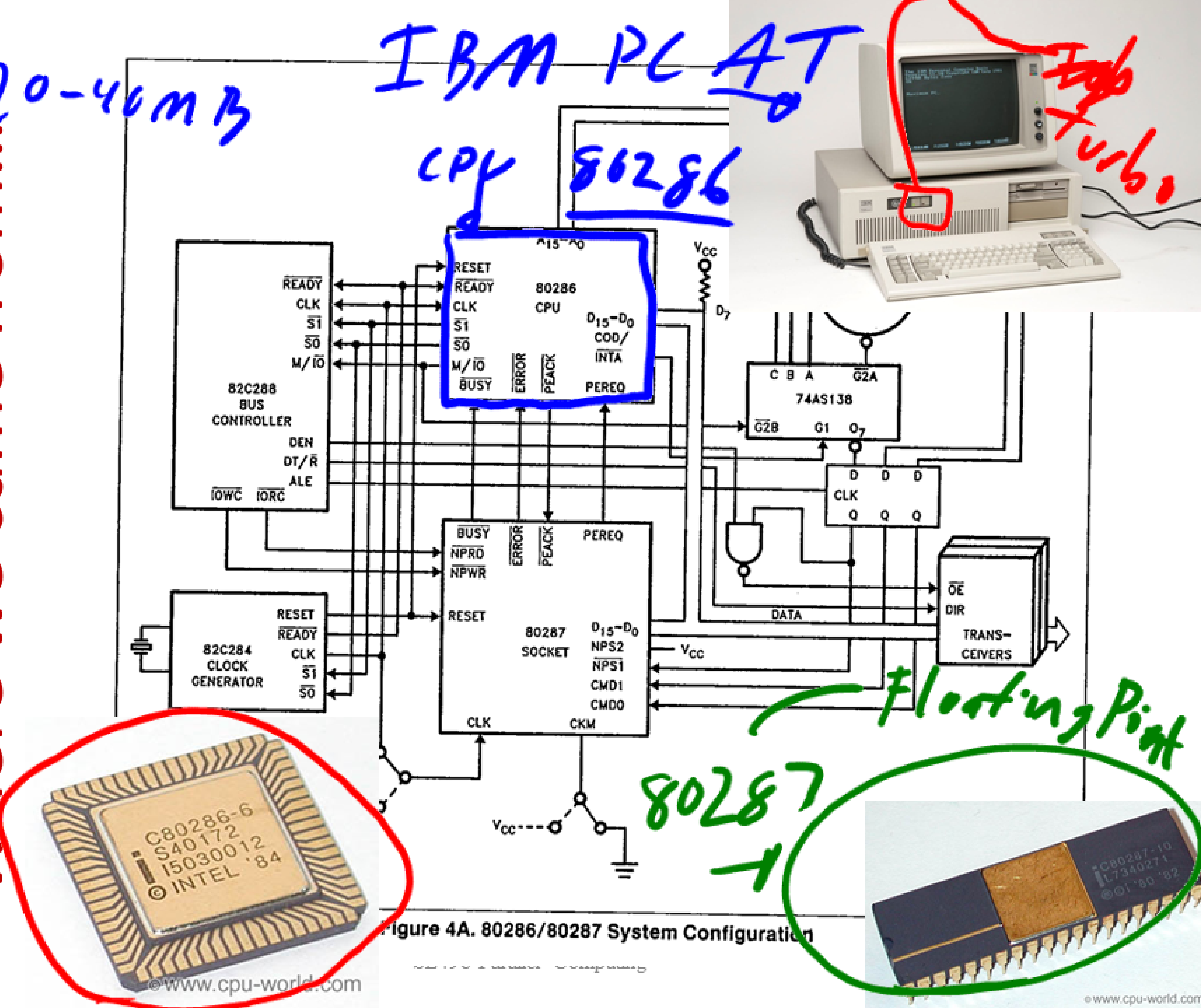
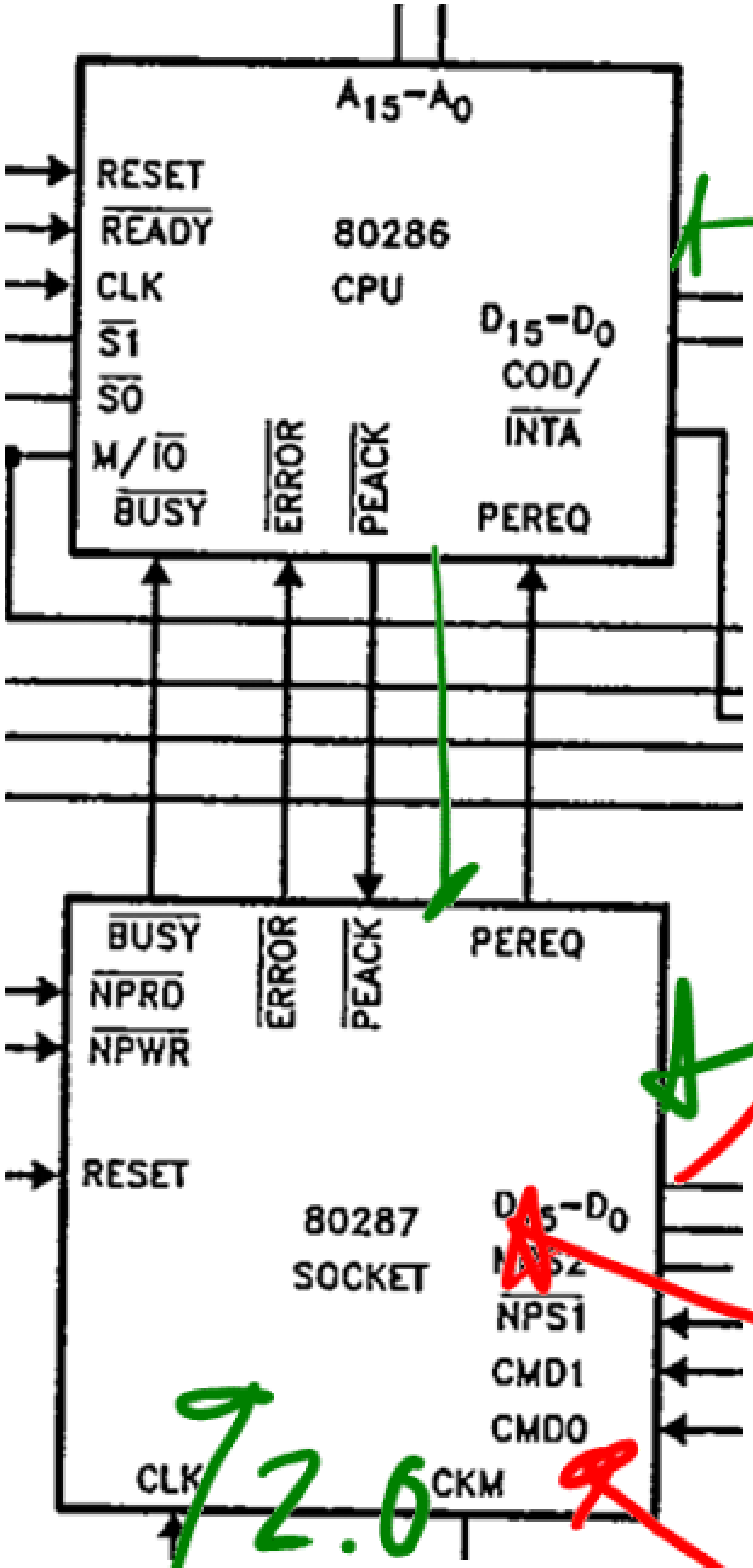


Figure 4A. 80286/80287 System Configuration

Where we came from…



"Accelerated" Data Floating Point.

Result

Floating

2.0/5.0

Divide

12.0

5.0

Definitions

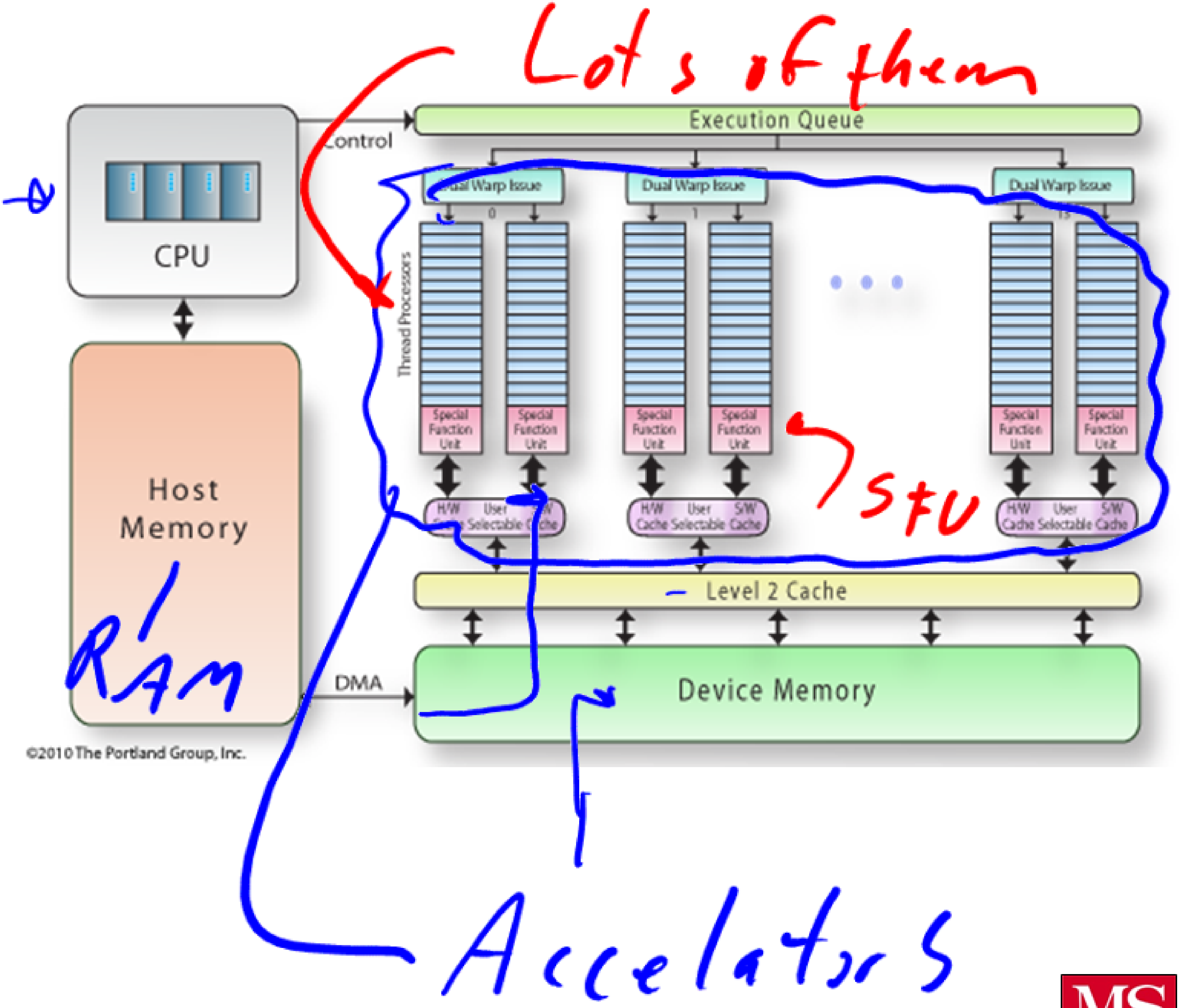
- Offloading
  - Offloading is the ability to send code to a coprocessor or GPU to run using many coprocessor cores.

Graphics Processing Unit

- Accelerator
  - A coprocessor installed on a computer for the purpose of improving system performance.

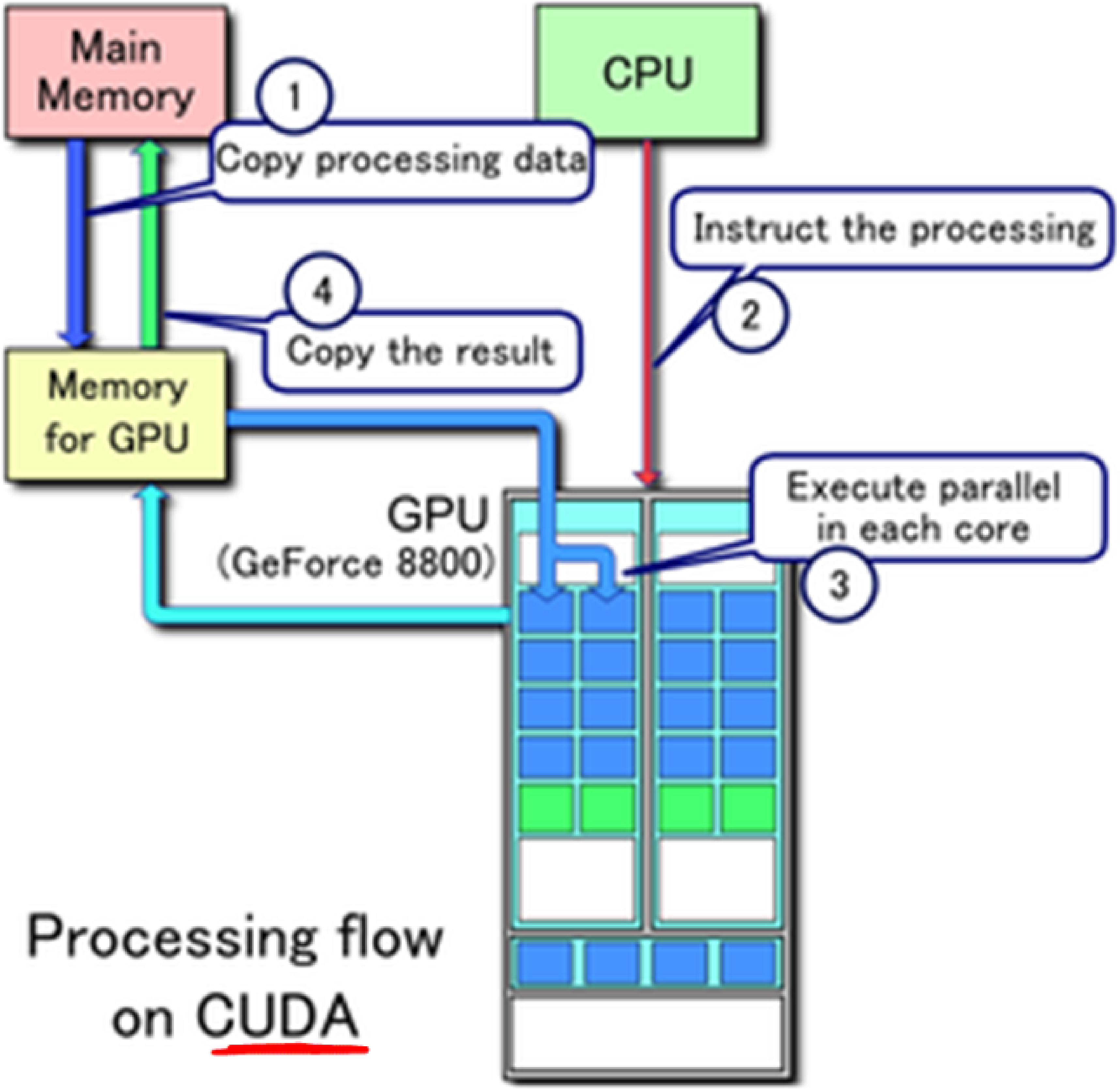
Specialized Processor.

Processor architecture to use accelerators



©2010 The Portland Group, Inc.

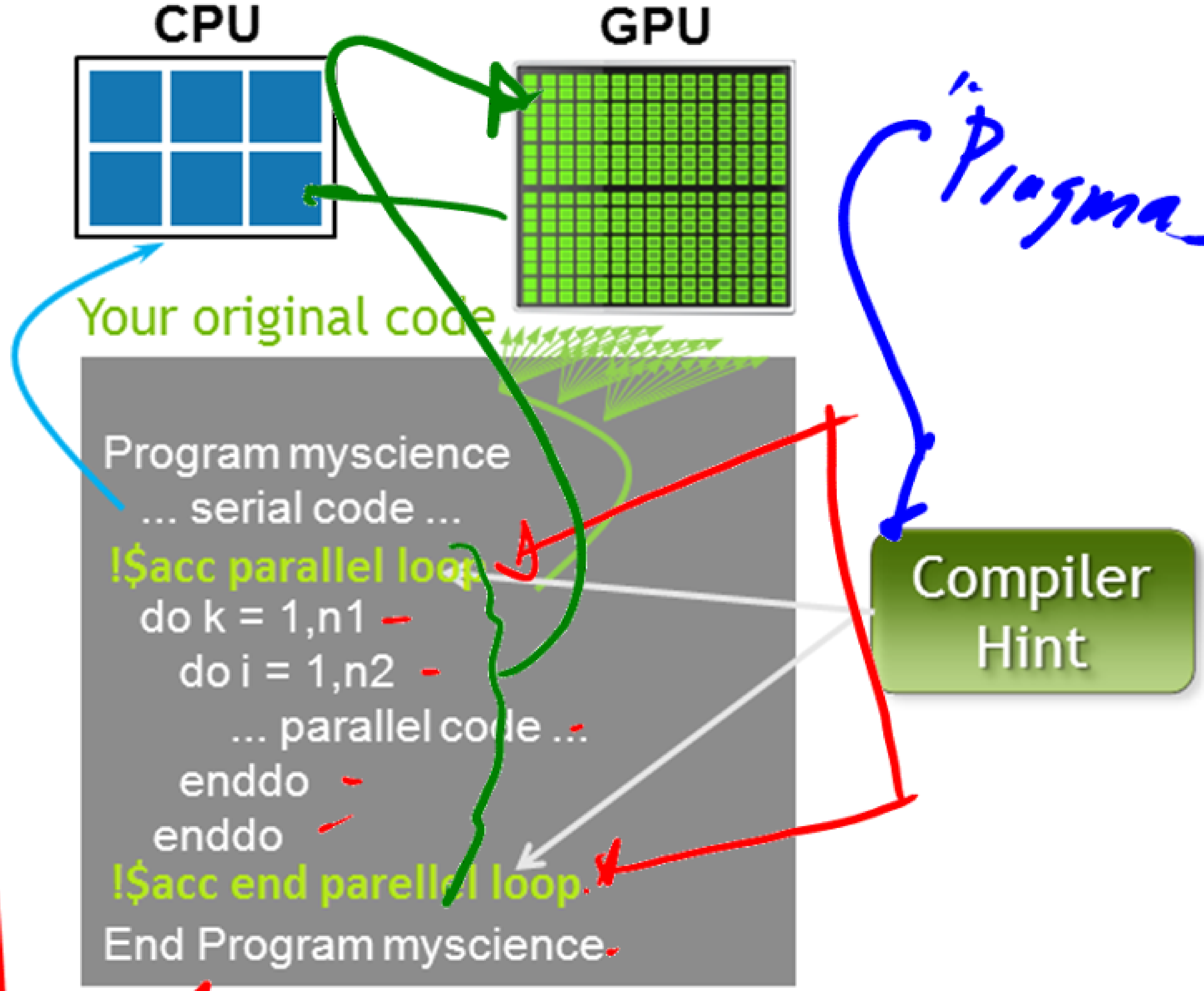
Using an accelerator



Processing flow on CUDA

OpenACC Program Structure

http://www.epcc.ed.ac.uk/blog/2013/04/19/opening-openacc



```
Program myscience
 ... serial code ...
!$acc parallel loop
 do k = 1,n1
   do i = 1,n2
      ... parallel code ...
   enddo
 enddo
!$acc end parellel loop
End Program myscience
```

"Fortran"

MSOE

Lets look at some code

- it will send a vector of floats to the GPU, double it, and bring the results back.

1.0 2.0 3.0 4.0 5.0 6.0

↓

2.0 4.0 6.0 8.0 10.0 12.0

```
/*******************************************************************************************
 * This is the main program for the doubler program.  It will send a vector of floats to the
GPU,
 * double it, and bring the results back.
 *
 * @param  [0] - The name of the program.
 * @param  [1] - This is the size of the array.  If not present a default value is used.
 * @return There is no return from the main method.

*******************************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

int main( int argc, char* argv[] )
{
    int n;          /* size of the vector */
    float *a;  /* the vector */
    float *restrict r;   /* the results */
    float *e;  /* expected results */
    int i;
    if( argc > 1 )
        n = atoi( argv[1] );
    else
        n = 100000;
    if( n <= 0 ) n = 100000;
```

How big is ?

100,000

n

Results

myprogram.exe 1,000.000

```
    a = (float*)malloc(n*sizeof(float));
    r = (float*)malloc(n*sizeof(float));
    e = (float*)malloc(n*sizeof(float));

          /* initialize */
    for( i = 0; i < n; ++i )
          {
                    a[i] = (float)(i+1);
          }
#pragma acc kernels loop
    for( i = 0; i < n; ++i )
          {
            r[i] = a[i]*2.0f;
          }
    /* compute on the host to compare */
    for( i = 0; i < n; ++i )
          {
            e[i] = a[i]*2.0f;
          }

          /* check the results */
    for( i = 0; i < n; ++i )
          {
        assert( r[i] == e[i] );
          }
    printf( "%d iterations completed\n", n );
    return 0;
}
```

Allocating memory.

Source values

Done on the accelerator.

Check results

Restrict keyword

- Declaration of intent given by the programmer to the compiler.
- Indicates that for the lifetime of the pointer, only it or a value directly derived from it (such as pointer + 1) will be used to access the object to which it points.
- Limits the effects of pointer aliasing
- Aids caching optimizations.

Scope of a pointer.

Lets look at another demo

- This multiplies two matrices together.