GRIDS

Grids

- A **grid** is simply a set of blocks arranged in X & a Y axis, in a 2D mapping.
- The final Y mapping gives you Y * X * T possibilities for a thread index.
- The no. of threads in a block be a multiple of the **warp size**, which is currently defined as 32.
- you can schedule a full warp on the hardware, if you don’t do this, then the remaining part of the warp goes unused.
- you have to introduce a condition to ensure you don’t process elements off the end of the X axis.

Grids

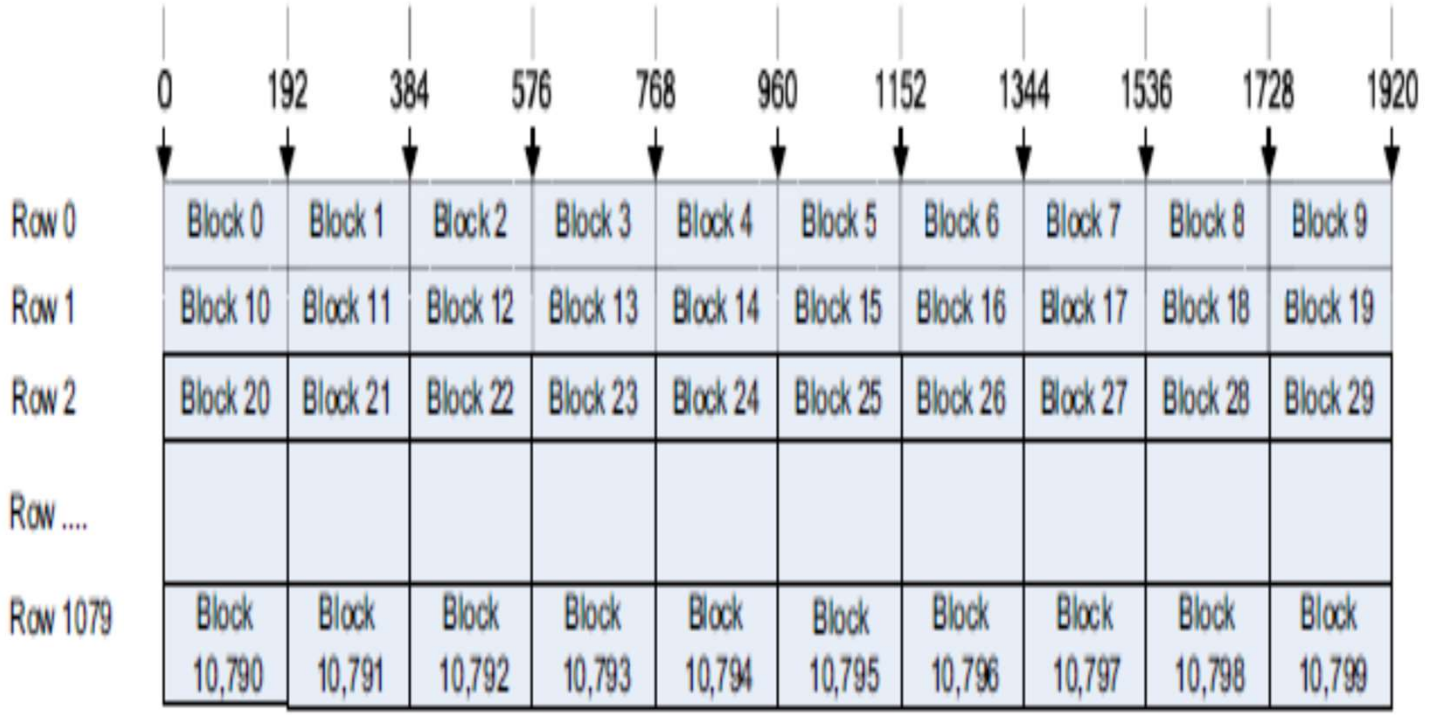


Fig: Block allocation to Rows

Grids

- A thread size that is a multiple of the X axis and the warp size makes it easier.
- Along the top on the X-axis, you have the thread index.
- The row index forms the Y-axis.
- The height of the row is exactly one pixel.
- You have 1080 rows of 10 blocks, you have 1080x10=10,800 blocks.

Grids

Stride and offset

- The **width of the array** is referred to as the **stride** of the memory access.
- The **offset** is the column value being accessed, starting at the left, which is always element 0.
- CUDA is designed to allow for data decomposition into parallel threads and blocks.
- It allows you to define 1D, 2D, or 3D indexes (Y x X xT) when referring to the parallel structure of the program.

Grids

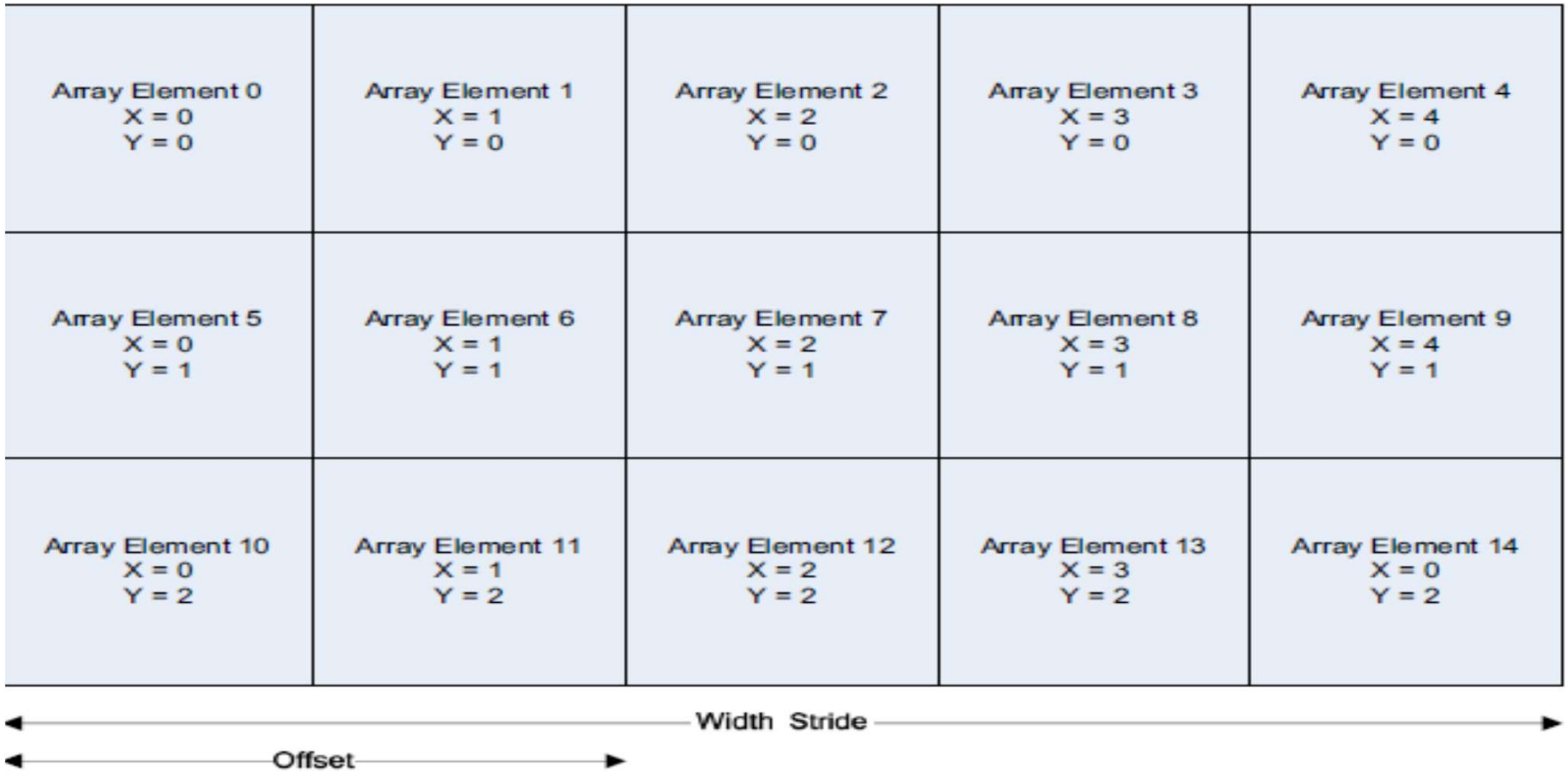


Fig : Array mapping to elements

Grids

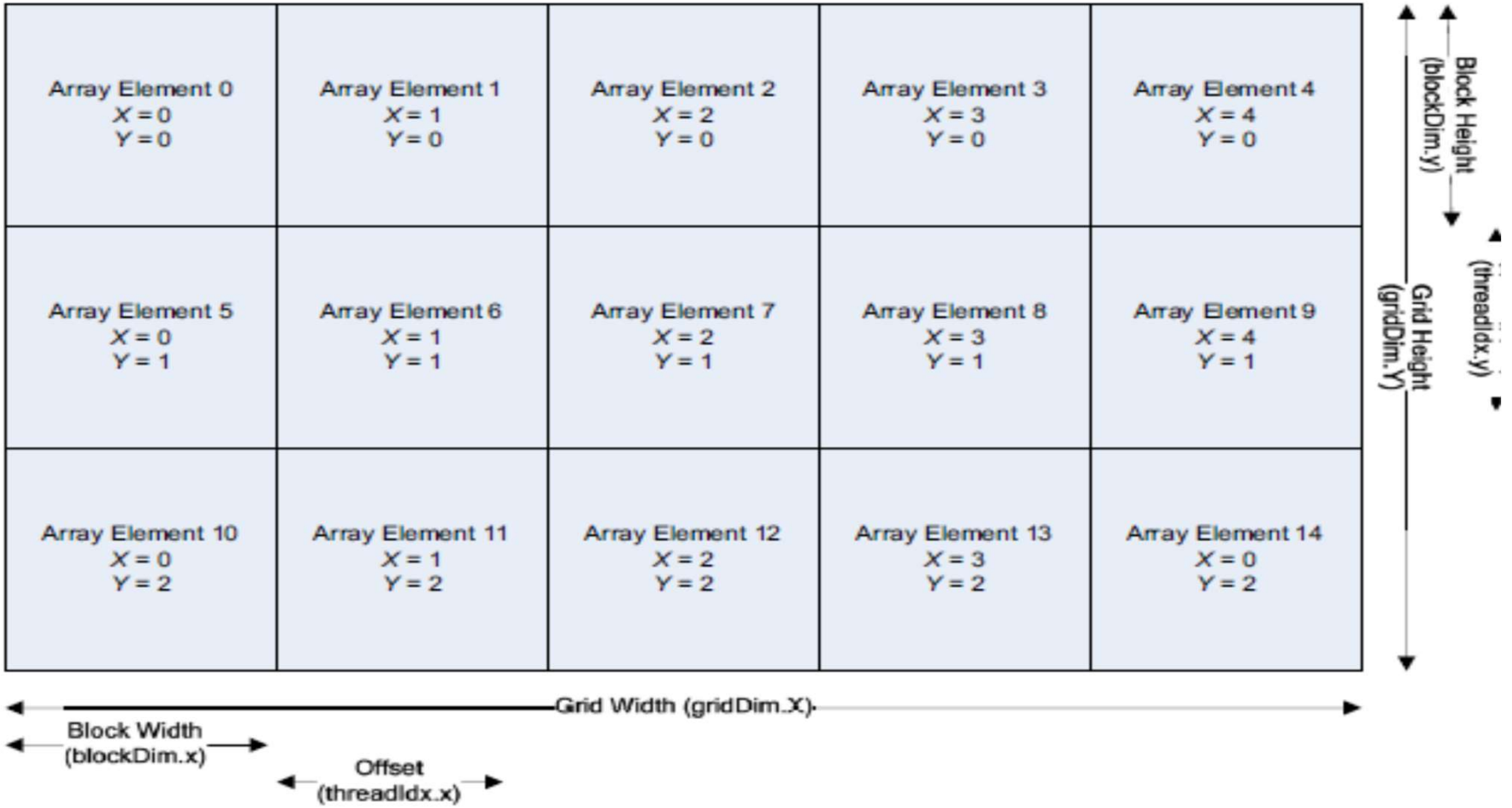
X and Y thread indexes

- A 2D array in terms of blocks means you get two thread indexes.
- you will be accessing the data in a 2D way:

```
const unsigned int idx = (blockIdx.x * blockDim.x) + threadIdx.x;
const unsigned int idy = (blockIdx.y * blockDim.y) + threadIdx.y;
some_array[idy][idx] += 1.0;
```

- blockDim.x and blockDim.y, specifies the dimension on the X and Y axis.
- You can schedule them as stripes across the array, or as squares within the array fig below

Grids



- Fig : Grid, block, and thread dimensions.

Grids

- You can see a number of new parameters, which are:

  gridDim.x–The size in blocks of the X dimension of the grid.

  gridDim.y–The size in blocks of the Y dimension of the grid.

  blockDim.x–The size in threads of the X dimension of a single block.

  blockDim.y–The size in threads of the Y dimension of a single block.

  threadIdx.x–The offset within a block of the X thread index.

  threadIdx.y–The offset within a block of the Y thread index.