# Logical detector geometry

The purpose of the logical detector geometry specification is to unambiguously convert wire and grid position values from the readout system into detector voxel ids for instruments based on multigrid detectors.

Several groups in different locations reliey on the correct processing of this information, so it is necessary to agree on how this data is interpreted. This should facilitate interoperation and minimize incompatibilitites of the developed software.

This document is not concerned with physical dimensions and positions. Instead it only works with discrete position indexes. Hence the name logical geometry.

The conceptual multigrid detector geometry is illustrated in Figure 1: Each detector (top left) consist of a number P of panels (right). Each panel is made up of M modules each containing G grids (bottom left). Each grid consists of a number of cells in the X and Z direction, these are named Nx and Nz.

Thus a ***single panel*** contains G\*M grids and (Nx \* Nz) \* M wires, and a *whole detector* has a total of P \* M columns (of grids). Figure 2 lists the numbers of panels, modules, grids and cells, etc. for possible multigrid detector based instrument configurations.

|  |  |
| --- | --- |
| MGB-IN5-V4-001-Ensemble 12 Modules.JPGIMG_0002.JPG | MGB-IN5-V4-002-Ensemble Module 8 tubes.JPG |

Figure 1: Detector geometry. Top left: The full detector assembly consists of a number panels. Right: A panel consists of modules. Moules are a stack of grids.



Figure 2: Parameters for various instrument and test configurations. The shaded cells are calculcated values, whereas the white cells are design parameters.

## Values start at 1

As a convention all ids, whether they are readout values, positions or voxels start at 1.

## Voxel definitions

Looking at the top-left picture in Figure 1, the x-direction is defined from left to right, the y-direction is from top to bottom, and the z-direction is from front to back. In integer units an (x,y,z) position corresponds to a unique voxel id. Given the specific detector geometry one can convert unambiguously between coordinate indices and voxel ids.

Given the above geometry definitions the x, y and z coordinates have the following ranges

The voxel ids are calculated from these positions as

It is also possible to convert from Voxel id to x,y and z values.

where *mod* is the modulus function and the edged braces are the floor function (possibly integer division).

|  |
| --- |
| #!/usr/bin/python  class MultiGridGeometry:  def \_\_init\_\_(self, panels, modules, grids, nx, nz):  self.P = panels  self.M = modules  self.G = grids  self.NX = nx  self.NZ = nz  def Voxel(self, x,y,z):  return (x - 1) \* self.G \* self.NZ + (y - 1) \* self.NZ + z  def XYZ(self, voxel):  x = (voxel - 1) / (self.G \* self.NZ) + 1  y = (voxel - 1) / self.NZ % self.G + 1  z = (voxel - 1) % 16 + 1  return [x, y, z]  def xyz2voxel(mg, x, y, z):  print "Coordinates (%3d, %3d, %3d) -> Voxel %4d" % (x,y,z, mg.Voxel(x,y,z))  def voxel2xyz(mg, voxel):  print "Voxel %4d -> coordinates (%3d, %3d, %3d)" % (voxel, mg.XYZ(voxel)[0], mg.XYZ(voxel)[1], mg.XYZ(voxel)[2])  MG = MultiGridGeometry(1, 2, 48, 4, 16)  xyz2voxel(MG, 1, 1, 1)  xyz2voxel(MG, 1, 1, 16)  xyz2voxel(MG, 1, 2, 1)  xyz2voxel(MG, 8, 48, 16)  voxel2xyz(MG, 1)  voxel2xyz(MG, 16)  voxel2xyz(MG, 17)  voxel2xyz(MG, 6144) |

Figure 4: Python implementation of conversions between Voxel and x,y,z values.

## Readout definitions

The formulae in the preceeding section enables us to calculate voxel ids once the x,y and z coordinates are known. However we first need to calculate x,y, and z from the readout data. This also requires some geometric conventions.

Referring to Figure 3 we get the following conventions for a single panel.

**Wire and grid orientations**

Within a *panel*, wire ids increase from front to back and from left to right. Wire ids range from 1 to NxNzM

Within a *panel* grids ids are increasing from bottom up and left to right. Grid ids range from 1 to G\*M

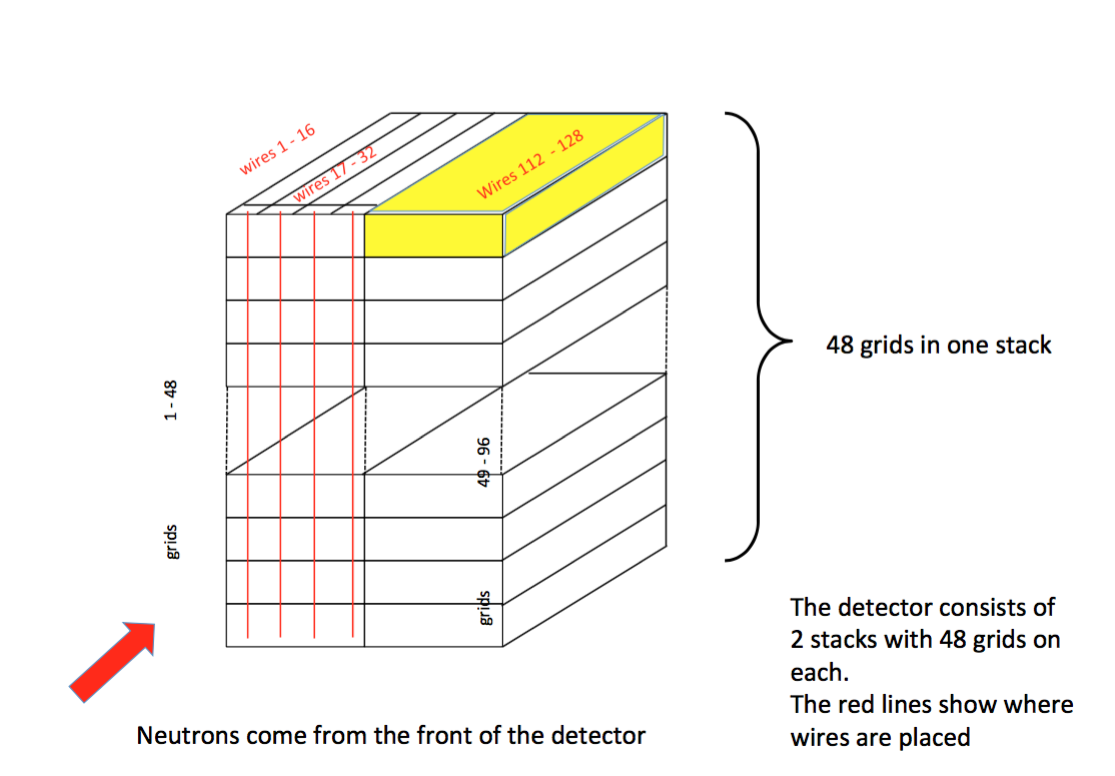


Figure 4: Geometry readout interpretation: Within each Panel, Grid IDs are numbered from bottom-up and from left to right. Wire IDs are numbered from front to back and from left to right. Voxels are numbered from front to back, top to bottom, left to right.

*Neither wire positions nor grid positions are globally unique, but can be used to calculate Y and Z. However in order to calculate X we need additional information about which panel the readout belongs to.*

*How to obtain the panel id from the readout system is not yet determined, but this is not material for the current discussion.*

We denote p, w, g as the panel, wire and grid ids from the readout system. Then x,y and z can be calculated as follows (again using modulus and floor functions).

For a complete example showing the connection between voxels, positions and wire/grid ids see the figure in the next chapter. The figure shows the (P=1, M=2, G=48, NX=4, NZ=16) configuration which has been manufactured and has already been subjected to several tests at the CNCS instrument. The figure shows the top and bottom grids of both modules. All corners of the grids have been annontated.

# Geometry Drawing

|  |  |
| --- | --- |
|  | Demo detector. This figure illustrates the conventions applied to the Demo detector (last configuration in Figure 2). Only the top and bottom grids are shown for the two modules. |