Let's imagine a system similar to L1Calo (not identical)

- Your firmware is going to receive input data from 7296 towers, 57 in eta and 64 in phi (see data.txt, 9 events)
- Each row of the input has:
  - A tower ID (just a number between 0 and 7295)
  - Whether the tower is an EM tower (if not it is a hadronic tower)
  - Phi for the tower
  - Eta for the tower
  - Energy for the tower in MeV
- Note that the input energies for hadronic towers have been found to be low, and all need to be scaled by a x1.25!
- Your goal: Output MET and MET phi for each event
  - Naively you want to find MET=$\sqrt{E_x^2 + E_y^2}$ where E_x and E_y are $\sum_i E_{xi}$ and $\sum_i E_{yi}$ but with hadronic towers scaled to the right energy and also $\phi_{MET} = \tan^{-1}(E_y/E_x)$

Some thoughts on your project

- You will need a bunch of trig functions and even exponentials to calculate Ex and Ey. Do you want to calculate this per event or have some LUT (lookup table)?
- How can you pipeline this? You do know the size of each event
- Maybe you can assume the inputs do not come in one tower at a time. We leave it to you decide how they arrive, as you are a firmware engineer but also a system engineer now

Project complication

- Uh oh! Some of the towers are noisy and bad and can't be trusted! To first order you can just mask and ignore them.
  - The list of bad towers is in noisy_towers.txt. There are 7 different example sets of noisy towers (not all the same size!). Assume these do not change per-event so you can load them as a LUT. See how this changes your MET calculation/answers but also your resource utilization and timing
  - FYI: of the 9 events in your input file, 3 of them should have no real MET, the others have varying amounts of MET
- Think about how to be more clever about dealing with noisy towers
  - Can you subtract an "average" per-tower quantity per event? Note that this "average" is not the same in each event! As before, how does this change your output but also your resource utilization and timing?

Project extension

- You will need a bunch of trig functions and even exponentials to calculate Ex and Ey. Do you want to calculate this per event or have some lookup?
- How can you pipeline this? You do know the size of each event

HW assignment!

In the git repo there is a folder with data.txt. Each line is a new set of data. The format is:

x,y,sigma_y,last

Where sigma_y is the estimated uncertainty on that y value and "last" tells you if this was the last input for this event (there should be 6 events in the file).

Write HLS code to do linear fits to data in each event. Note that you don't know in advance how big each event is, but you can assume a maximum of 250 events. Anything beyond that can be ignored. Try and optimize the code as much as you can.

For each event, return a,b, the uncertainty on those two parameters, the chi2/ndf and if you had to skip any events. Note that some of the events should fit very well to the model. Others might not!