Title: TEAM STATUS

Team B2 - Eshani Mishra (emishra), Kimberly Lim (klimjinx), Shruti Narayan (shrutin1)

What are the most significant risks that could jeopardize the success of the project? How are these risks being managed? What contingency plans are ready?

- We are not sure how to save 5sec of the live video feed into SDRAM - there are issues making sure the right data is saved into different addresses in SDRAM. We have spent a lot of time on this and have not made promising progress…
    - Contingency - We will record a video separately and use the HPS to pre-process it to meet requirements and pass it into SDRAM

Were any changes made to the existing design of the system (requirements, block diagram, system spec, etc)? Why was this change necessary, what costs does the change incur, and how will these costs be mitigated going forward?

- For stage one and the initial blocking of the video frame, we changed the format of how the pixels will be stored into embedded memory and then processed. Instead of parallelizing the computation over the rows, we switched to parallelizing over columns. This change occurred because the initial input of the video from the SDRAM will be serial, and the stream will be received row by row. If we parallelize over the columns and compute partial sums for the energy map one row at a time, we can start this process as soon as one row is received from SDRAM rather than wait for the whole frame to be loaded in. This involved recalculating the partial sums over rows and organizing the embedded memory
- Instead of storing the entire energy map (and all such matrices the size of one frame) in a single embedded memory module of 64 blocks, we can store each individual column (and potential copies) in a single block of its own and use 256 total blocks for each frame. We only need to work with max 2 frame-sized embedded memory sections at a time in this way (eg Stage 1 will have the image storage and the energy map storage, Stage 2 will have the energy map storage and the accumulation paths matrix storage), so we will be under the ~550 blocks total cap for this particular board. This will allow simultaneous reads/writes to all columns at once, to allow for parallelization (otherwise, a single 64-block section

Provide an updated schedule if changes have occurred.