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?

- Resolution - Running our initial 256x144p design on the board caused issues with the fitter - we we over-utilizing the logic (ALMs) resources the board has. Minute changes to the design
    - Risks Managed: We only instantiate half of the total processing resources (registers and ALU, etc) in our design and then process half a frame per round in stage1 instead of one frame per round (by round, I mean each run through the energy map pipeline - so only half a frame's columns will be parallelized). We do this by loading in 129 rows to have that last row part of the 3 column partial sum calculation.
    - Contingency: 128x128p video resolution
- Display onto monitor - We plan to use a C++ program to post-process the video for our three output options and use the Linux desktop GUI to open and display the output videos. Our issue is that the Quartus project that drives the VGA for the Linux GUI (default on the board when it is turned on) is overwritten by the new project that holds our algorithm and the VGA output for the desktop GUI no longer shows
    - Risks Managed: Once the algorithm runs and the final output videos are stored on the SoC file system, we can turn the board on and off. This will remove our Quartus project and leave the default project that drives the VGA with the desktop GUI output - this way we will be able to interact with the GUI and use it to show our output videos
    - Contingency - We can use our laptops and scp onto the board to get the output videos (we would have to work around the issue of finding a shared network)
- Each time we reset the board (and on the first use) we are getting inconsistent seam values. This could be because embedded memory is not connected to the reset and we never explicitly clear it before using it.
    - Risks Managed: We are going through each module and using Signal Tap to test and verify outputs and correctness, in case the issue is beyond the embedded memory.
    - Contingency: We will just do a key reset and use the second time. We will also add an extra stage that writes 0 to all the addresses in each embedded memory bank.
- Correctness - see above point (using Signal Tap)

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?

- Resolution - see above
- Monitor display - see above
- We only remove one seam per algorithm iteration - for the next seam, we must re-input video data. Using HPS this should not be difficult to manage.
- We use the HPS-FPGA FIFOs to transfer video pixel data

Provide an updated schedule if changes have occurred.