|  |
| --- |
| **MEETING MINUTES – Sponsor & Team Meeting** |
| DATE: October 2, 2014 |
| POSTING TIME: 10/03/14 12:39:24 PM EDT |
| OWNER: Julia Kim |

**Last Edited: Sunday, October 5, 2014 4:24:37 PM EDT**

Present: Matt Cammuse, Patrick Delallana, Joshua Cushion, Julia Kim, Malcolm Harmon, Mark Poindexter

Time 5:00 p.m. – 6:15 p.m.

We asked for his comments on our Needs Analysis and Requirements Specification report and he said that the only thing that he wanted to comment on was that the antenna arrays were outdated. He said that we would only be working on the linear array for now as it is more straightforward, since the sunflower array is too complicated to deal with right now.

Joshua asked about the ADS simulation with the components. Pete asked what software we have for that, and he mentioned that for now, it doesn’t seem like we have any of that software available but would look into it with Dr. Foo in order to verify. It’s not a viable solution to buy the license for the software that Joshua looked into because it is very expensive and would go against our budget. Pete said that we could do this on Excel in order to have a better understanding on what’s going on. For the ADS, we would have the components there that would be pulled into a schematic and cascade them and give them certain characteristics, such as small signal gain, compression point, radio transmit. Then a test bench is set up to measure the overall performance. Joshua will coordinate with Dr. Foo and go over the different software that we have and see if there’s something that could do this. The ADS simulation isn’t a circuit simulation/design, but instead writing equations representing components characteristics.

For the range equations, we would just have to plug in the equation in Excel and also have the variables as well. That’s how we would do our signal to noise; when we have a certain amount of transmit power, a certain antenna gain, the target is at a certain range and has a certain scattering, that’ll go into the equation to determine how much signal we’ll get back at the receive port. We need to get enough signal back to measure. There should be plenty of examples that show how to use the equations.

He asked if we looked into the array factor and pattern on horn. Flashlight analogy: each horn is like a flashlight that’s putting out light, and as you angularly go off of the peak of the light beam, and where the peak (power) drops 3dB, which is the beam width on either side. So if the power drops off about 3dB on either side, then we would have a 2 degree beam width. So if we’re 20 ft. away and have a 2 degree beam width, we can figure out what extent it spans, and that basically becomes your scene. Every horn has to be able to put energy on that scene extent and that’s defined by the beam width, which is proportional to the gain of the antenna. There’s a basic equation for apertures, 26000/(beam width squared) = directivity gain of the antenna, which is a magnitude. As the gain goes up, the beam width gets narrow. So we would need to pick an antenna gain that gives us a 3dB beam width that would cover the scene extent of 1 person, of about 30 inches or so.

The other variable is how close the horns go to each other. If the horns are too far apart from each other and go out to different angles, we would get a different beam come into the antennas. Each horn is picking up a different phase and when they combine, they would be out of phase. If we take the 8 horns, separate them by a rod, and go off by a certain angle, we would create a path line difference that is pretty significant between each horn. So what’ll happen at a not too far off angle, the horns will sum up the energy and shoot it again, creating grating lobes. The farther apart the horns are, the smaller the angle. We would be synthesizing the pattern from all 8 horns together. If each horn is illuminating 30 inches at 20ft, when we take the data from all the horns and combine them, we get the beam width of an antenna that’s 8 horns wide. That makes it go to like 3 inches, a fraction of the scene extent, and that allows you to locate where the scatter is in that scene. If the antennas are too far apart, when you combine all the data from the antennas, it will also pick up energy that’s outside the scene and bring it into your scene, which is called ambiguity, where you get the same scattering result at other angles and the idea is to keep that angle out of the scene extent.

If you have two horns, where one’s transmit and the other’s receive and they’re separated at a distance d. If you transmit out of one horn and receive through the other, from a performance standpoint, it’s equivalent to having one horn midway in between that has a receive/transmit behind it. It reduces the hardware behind the horns because instead of having a transmit amp and a receive amp behind each horn, we would have one transmit and all receives connected to a switch that we receive into at different times.  So we create the same phase centers over time, so we transmit out one horn and receive through another, which creates a phase center that’s equivalent to having one horn midway between that transmit/receive, and then we transmit out the same horn and switch the switch over to receive through the next horn over, which creates another virtual phase center, and so on. So essentially we’re creating these transmit/receive apertures without any hardware and sacrificing time, as we cannot do it at the same time.

On the antenna aperture drawing, each one of the open ended triangles represents a physical horn, so there are 10. Configure the horns in the arrangement drawn and point them to the scene. Ideally, we would want another set going vertically. Figure out a way to get the face of the antenna pointing at the scene and erect it in a manner that it stays put instead of wobble. Orientation of the horn should be in a manner so that the wider beam is in elevation.

We’re using 10 antennas, 8 receives and 2 transmits on the outside, in each line. So we’re using 20 antennas in total.

For the testing room, nothing can be in the way of the path from the equipment to the target. Also put absorbers at 5ft on each side on the path and there shouldn’t be anything in the way, and the room could also be approximately 8ft high.

For the programming, Pete wants the code to be tested in order to verify whether the 20ns wide pulse is generated successfully from the FPGA board. The pulse width can be changed with the slider switches, and the space in between the pulses should also be realized in some manner. There are multiple pulses that come out to do various tasks, such as a switch that switches between the different receive horns. There has to be some switch that can take 3 bits. For the single pole 8 way switch: there are 8 positions and need 3 bits to switch that and need some kind of driver that will take the 3 signals and depending on the level will correspond to one of the switch positions. That switch can be slow and can be mechanical too. There are 3 outputs needed and the switch doesn’t have to be 20 ns. A timing diagram should be made on all the things that need to be controlled and map that out with the pulses relative to each other. You would set the switch position for the receive and that would stay static, then send out the transmit pulse, and then switch the switch to receive. So the timing should be set out in an appropriate manner.

The clock rate should be at least 25MHz for the 20 ns pulse and a period of 40 ns. So the FPGA should be at least 100 MHz; the higher the better because you get better quality. If you have a pulse that goes down the transmission line and is not well matched, then it’s a mess. Research a D to A that can be plugged into the FPGA and can talk to and send a digital wave.