

JASPER and the SKARAB

Wesley New
2017 CASPER workshop

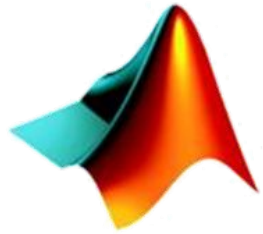


science and technology

Department:
Science and Technology
REPUBLIC OF SOUTH AFRICA

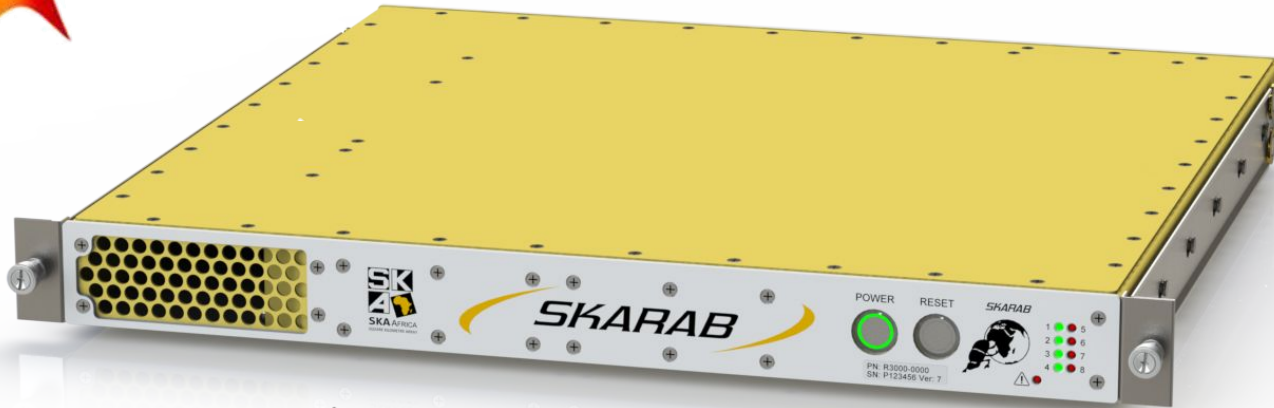


Hardware



MATLAB®

VIVADO™



python™

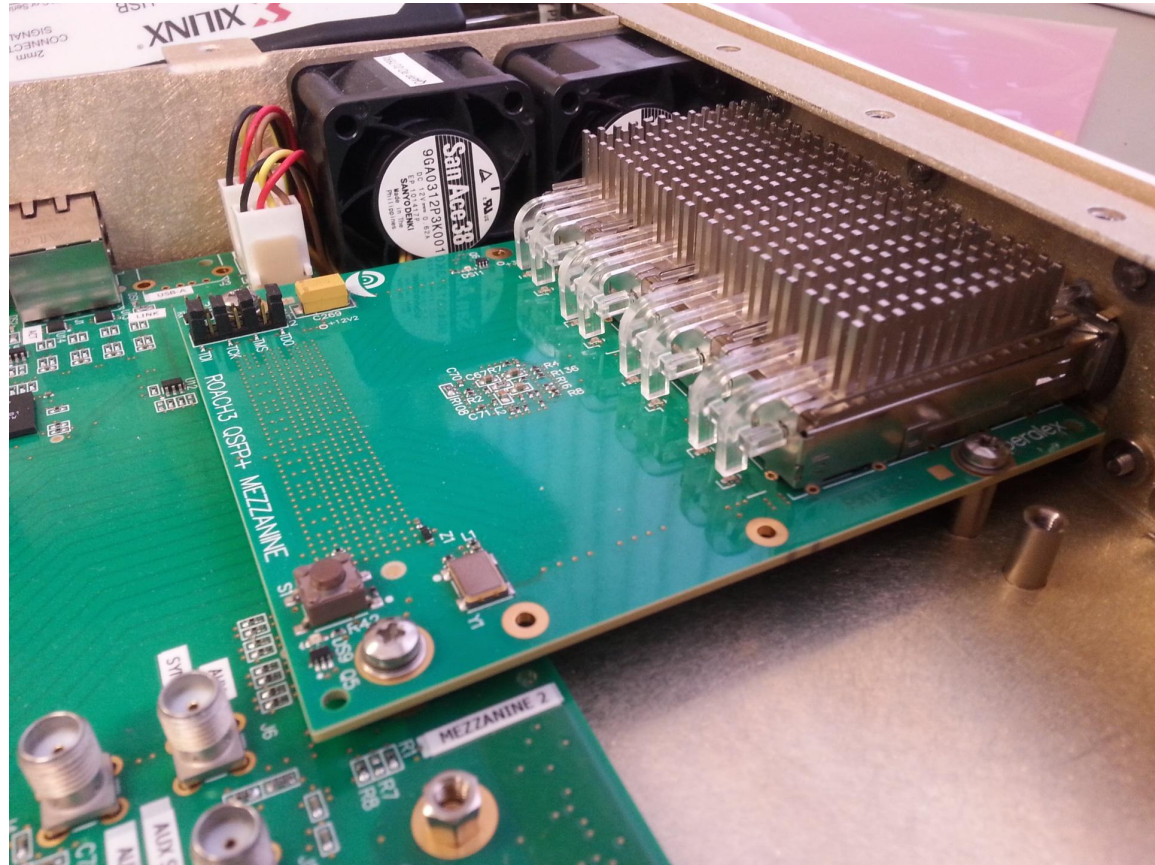
Hardware: SKARAB Motherboard



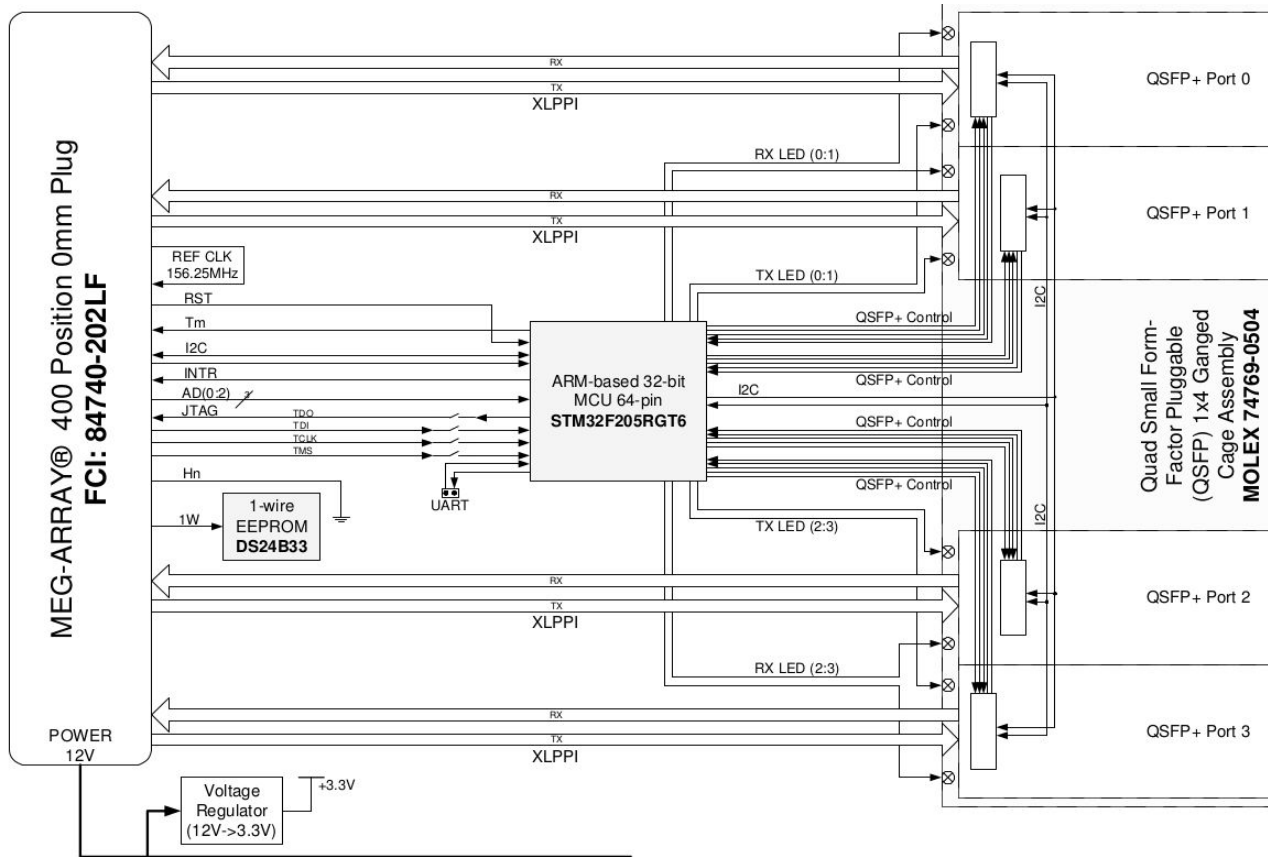
- Peralex in conjunction with SKA-SA have designed the SKARAB.
- Based on the Virtex 7, 690T FPGA
 - 53Mb BRAM
 - 3600 DPS Slices
 - 693k Logic Cells
- 4 Mezzanine sites per SKARAB
 - 40GbE
 - HMC
 - ADC

Hardware: QSFP 40G mezzanine card

- Quad 40G QSFP Ethernet card
- The PHY is located on the FPGA.
- The mezzanine has an Arm microcontroller to configure the QSFP modules, via I2C.
- JASPER still requires support for multiple ports and multiple cards.

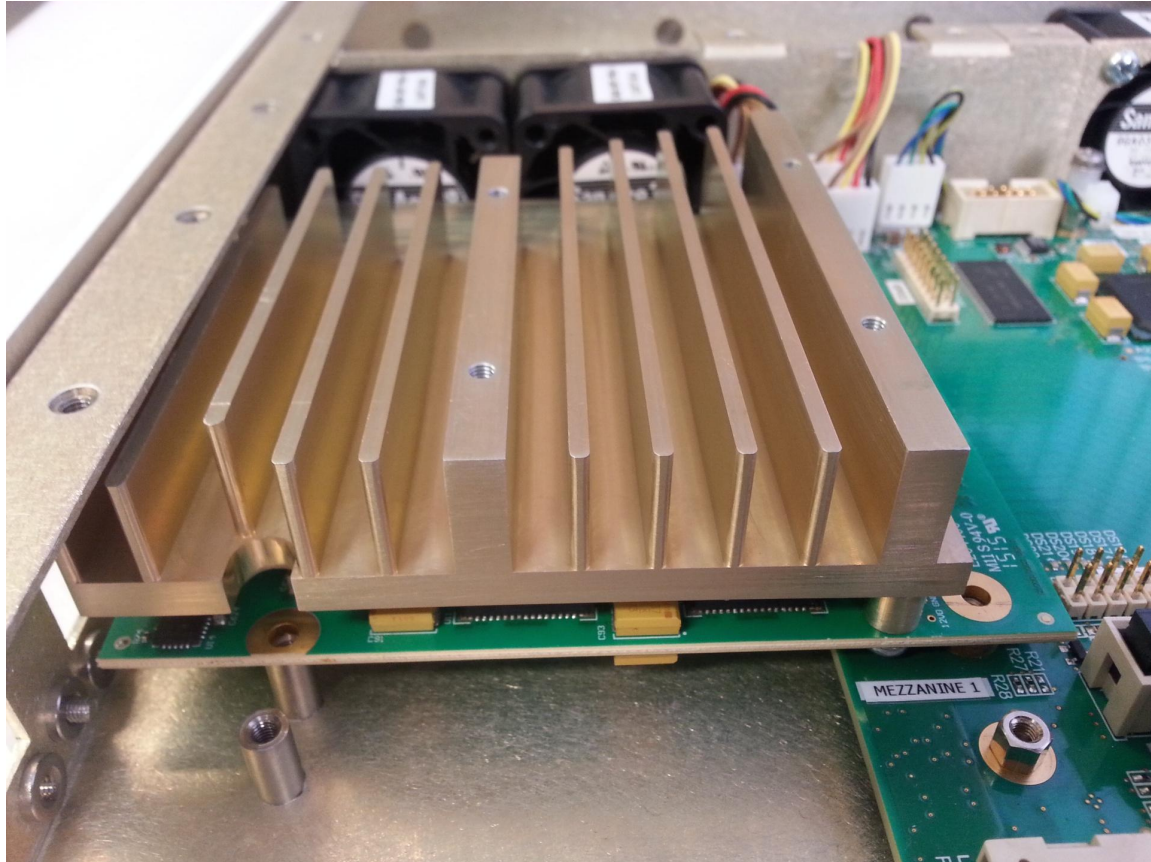


Hardware: QSFP 40G mezzanine card

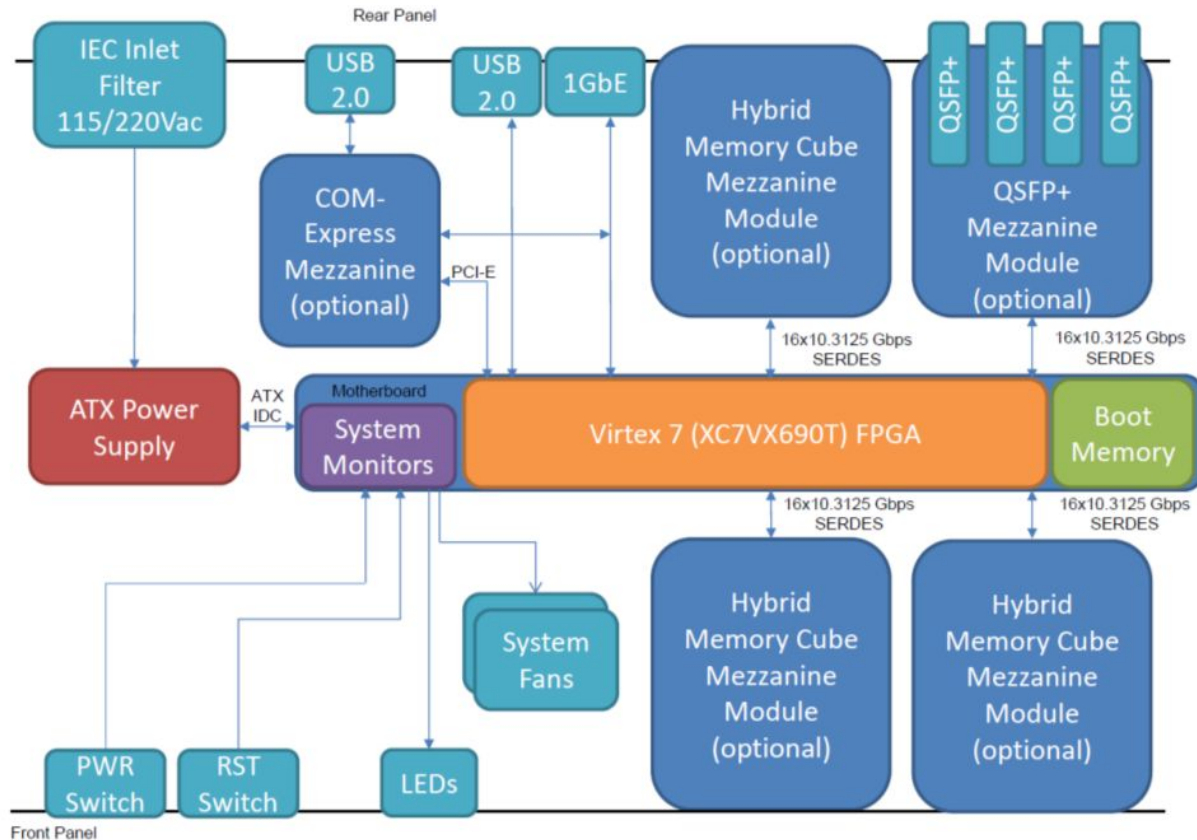


Hardware: HMC Mezzanine

- Single HMC device per card
- HMC is 2GiB or 4GiB
- 3D Stacked DRAM
- Through Silicon Vias
- Bi-directional.
- Up to 160Gbps throughput per card
- Jason will share more about this in another talk...
- There is a plan to move to HBM on the next version of SKARAB



Hardware: HMC Mezzanine



Softcore: Microblaze



- SKARAB uses a softcore Microblaze
- Replaces the PowerPC of the ROACH1 and 2
- SNAP also uses a flavour of the Microblaze, but implements the communication over KATCP.
- The SKARAB uses a set of opcodes which are sent via UDP packets over either the 1GbE or 40GbE
- Used to manage the control communications to/from the SKARAB
- Facilitates programming and flashing images to the SAKARAB
- Manages the Fabric Ethernet ports, ARP, DHCP, IGMP, ICMP, LLDP, Hostnames etc

40GbE Core



The 40GbE core on SKARAB is exposed to the user in much the same way that the 10GbE and 1GbE were in CASPER.

- Exceptions of the data widths, which are now 256bits and a word size of 64 bits (Although this is transparent to the user)
- The multicast support has been migrated from the 10GbE to the 40GbE.
- The rx packet ip and port are now provided to the user, this is particularly for multicast environments so that the packet destination can be determined.
- The memory map of the core info is much the same as the 1 and 10GbE cores. This is managed and populated by the microblaze after the DHCP transactions have taken place.
- Currently only one of the 40GbE ports is supported in JASPER, but support for the remaining will be added in the future.

HMC Core



Where the ROACHs used QDR the SKARABs now use HMC.

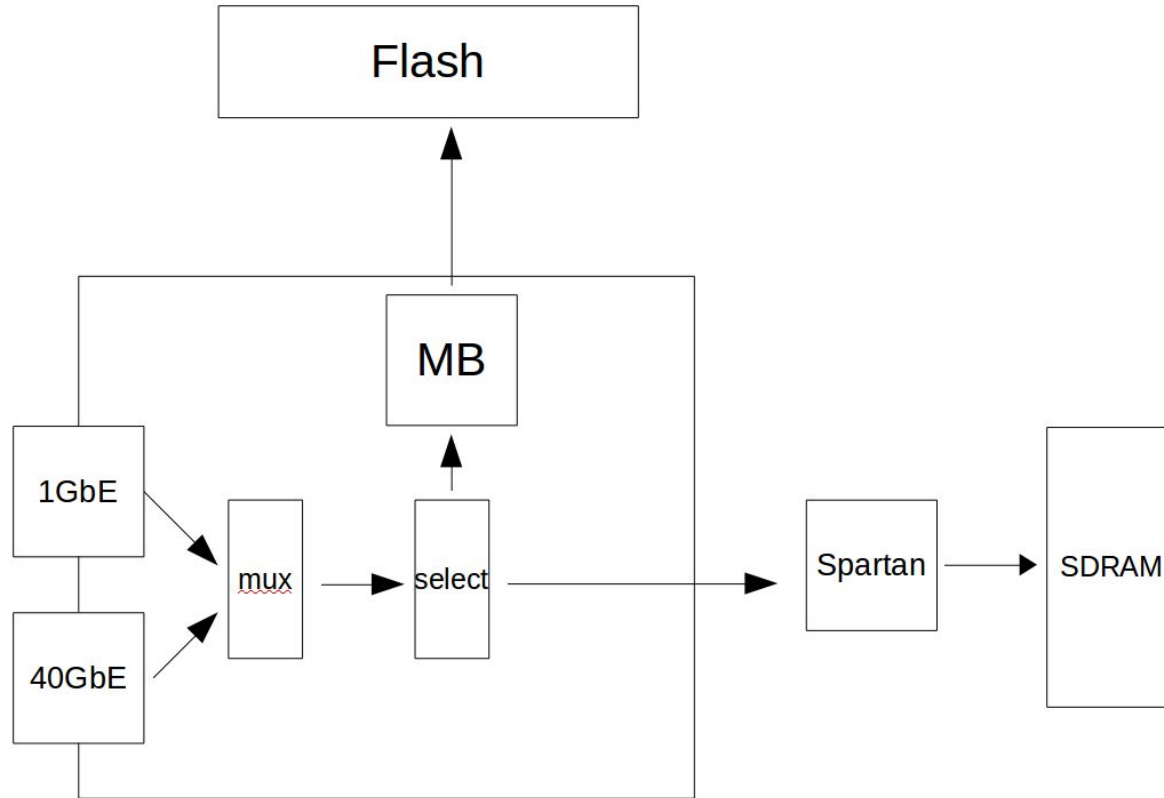
- Adam has done some great work to integrate the HMC into JASPER.
- Using the Open HMC code and incorporating it into JASPER as a yellow block.
- We have tried to make the interface as similar to the QDR as possible.
- Unfortunately it is a complex interface and this has come with its challenges.
- But Jason will be sharing more on the HMC in another talk...

SKARAB Programming Architecture



- The SKARAB can be programmed over either the 1GbE or the 40GbE interfaces.
- This is enabled by the microblaze which ascertains which interface the programming commands are coming over and configures the things accordingly.
- When programming the software sends a command to the microblaze to ready it for programming.
- The microblaze then selects the correct interface to drive the SDRAM interface. Well, not exactly, it is a Nor-flash interface to a Spartan FPGA which translates to an SDRAM interface.
- Now any data that comes in on a specific port will be directed to the SDRAM.
- The SKARAB also has flash memory which house the Board Support Packages. One of which is loaded into the FPGA on boot. This provides the interface to upload a bitstream over the FGPA fabric.

SKARAB Programming Architecture





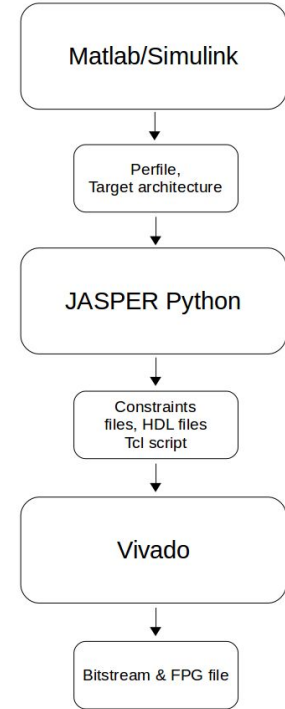
SKARAB uses the new JASPER flow rather than CASPER

- ISE has been replaced with Vivado, currently 2016.2 SKARAB 2016.4 SNAP
- Same Matlab Simulink frontend, all your ROACH designs should be compatible. Besides swapping out peripherals (GbE, Memory ADCs)
- The backend is no longer Matlab code but Python
- The idea is to separate the frontend and the backend so that it is easy to add support for other frontends or even backends (in the future possibly Altera)

JASPER Architecture



- The JASPER flow ties together the Matlab/Simulink design with the Vivado backend
- JASPER Matlab script runs sysgen and generates perfile
- The Python takes the perfile and creates the tcl script to run in Vivado
- The Tcl script generates the Vivado project and pulls together all the required IP and cores
- The design is then synthesized and implemented
- The bitstream is generated and the FPG metadata appended to the head of the Output bitstream



How to use JASPER



- The old xsg_config block has been replaced with a single block per platform.
- These blocks can be found in the platforms sub-library
- To run the toolflow type 'jasper' in the matlab command window
- This will kick off a compilation and with any luck you will get a bitstream out the otherside.
- You can also compile your design in two stages. You can run "jasper_frontend" which just runs matlab and system generator And generates the perfile
- Then you can run the python by executing the string that is returned from "jasper_frontend", this runs the python and generates the necessary files for vivado and then kicks off vivado.
- The python function is "exec_flow.py" and can be run with a range of flags.



FPG File Structure



The FPG has replaced the bof file. It contains the same data as the bof file and more design information.

Design Metadata	Memory Map	Git Info	Bitstream
-----------------	------------	----------	-----------

CASPERFPGA



So now that we have an FPG file how do we program the SKARAB?

- The same way that ROACH was programmed, via CASPERFPG
- The SKARABs don't support katcp, but have their own set of opcodes
- This has been encapsulated into CASPERFPGA, which exposes the same functionality as for the ROACHs
- Now you can create a casperfpga object and pass it the IP of your hardware, it will automatically detect the platform (ROACH, SKARAB, SNAP) and use the appropriate transport layer, which is abstracted away from the user.
- This allows us to write software that is independent of the hardware
-

CASPERFPGA



So now that we have an FPG file how do we program the SKARAB?

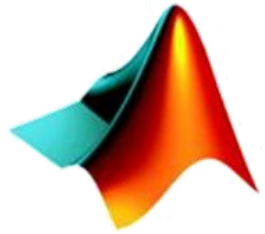
- The same way that ROACH was programmed, via CASPERFPG
- The SKARABs don't support katcp, but have their own set of opcodes
- This has been encapsulated into CASPERFPGA, which exposes the same functionality as for the ROACHs
- Now you can create a casperfpga object and pass it the IP of your hardware, it will automatically detect the platform (ROACH, SKARAB, SNAP) and use the appropriate transport layer, which is abstracted away from the user.
- This allows us to write software that is independent of the hardware

MeerKAT

- Set of 12 SKARABs on which the MeerKAT Backend is being designed
- Ultimately we will have over 200 SKARABs Out in the Karoo desert in South Africa.

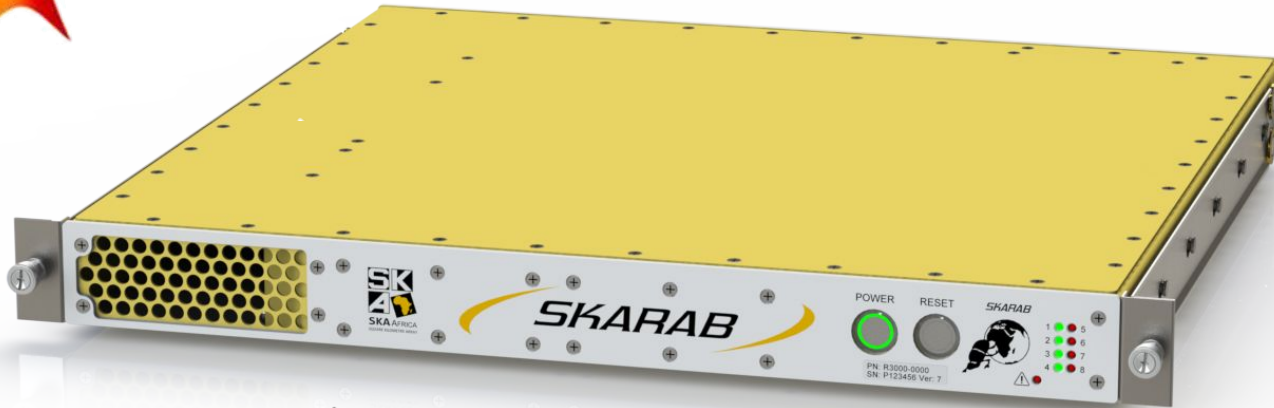


Hardware



MATLAB®

VIVADO™



python™

Any questions?