**Pizzabox FPGA programming workflow:**

1) Change custom core (veriog/RTL), build in ISE. \enc\pquest\_1.srcs\sources\_1 \edk\microblaze\pcores\*core\_name\_ver*\devl\projnav\

2) In XPS: import periphery again. remove module wrapper \*.ngc files from implementation folder (\enc\pquest\_1.srcs\sources\_1\edk\microblaze\implementation\ and \enc\pquest\_1.srcs\sources\_1\edk\microblaze\implementation\cache\) and put OPTION ARCH\_SUPPORT\_MAP = (others = DEVELOPMENT) to /data/\*.mpd file, then Generate netlist. (Q: do we need to re-import if no ports/sub-modules are not changed?)

3) Open Planahead, add sources from PlatformStudio/implementaion, add whole dir (\enc\pquest\_1.srcs\sources\_1\edk\microblaze\implementation\) with netlists (.ngc). Import constraints if ucf file was changed. Implement design, generate bitstream. Go to File ->Export bitstream to hw folder of workspace of SDK. (\enc\pquest\_1.sdk\SDK \SDK\_Export\pquest\system.bit or \enc\pquest\_1.srcs\sources\_1\edk\microblaze\SDK\SDK\_Export \hw\system\_cclktemp.bit?)

4) If new modules were added or address space allocation was changed: export .bit to SDK export directory of XPS. In XPS click on Export design, uncheck include bit and bmm. This will update .xml of hw platform in SDK export dir (\enc\pquest\_1.srcs\sources\_1\edk\microblaze\SDK\ SDK\_Export\hw\system.xml).

Copy .xml to hw platform in EDK workspace (which directory?). NOT NEEDED.

5) (We also need to do this even programing FPGA only?) If mcs will be generated: in SDK: open \*\_bd.bmm file and manually edit ramb36e1 locations. Open implemented design in PlanAhead, search for instance \*ramb36e1\*, edit all sites placed. (Q: How to decide the sites?)

6) In SDK open XMD console, cd to hw platform directory (\enc\pquest\_1.sdk\SDK\SDK\_Export\ pquest\), execute

data2mem -bm "system\_bd.bmm" -p xc6vlx240tff1156-1 -bt "system.bit" -bd "boot.elf" -o b pquest.bit

Alternatively make boot active project and download fpga. This will create pquest.bit file that is merged boot.elf and system.bit.

Make sure boot was build with heap and stack in BRAM, use standalone\_bsp with main project in DDR3 and uses xil\_kernel\_bsp. (Not clear about this operation)

7) If mcs will be generated: In ImPACT start page select create a PROM, BPI -> virtex6 -> 128Mb -> 16bit. Save and download .mcs. Use 28AF00P30 flash, RS pins 26-25. Uncheck verification after programming to save time. Program result mms (mcs?) file into flash.

8) Cycle the unit, check LED indication for proper firmware load.

9) In SDK, program flash with application elf. (Q: Why need to download elf separately? It should have been included in pquest.bit generated in 6.)

10) Cycle the unit, check the serial console of the unit.