1. create the correct test binaries for UPC verification

① create upc_uart.bin (run on ARM formater core of upc)

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-ld -entry=0x0 -Ttext 0x0 -EL -Tdata **0x0401E000** -Tbss **0x0401E000** -o **upc_uart.elf** preboot_fm.o upc_uart.o dbg_printf.o

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-objcopy -O binary upc_uart.elf **upc_uart.bin**

② create upc_wait.bin (run on ARM io core of upc)

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-ld -entry=0x0 -Ttext 0x0 -EL -Tdata **0x0401EA00** -Tbss **0x0401EA00** -o **upc_wait.elf** preboot_io.o upc_wait.o dbg_printf.o

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-objcopy -O binary upc_wait.elf **upc_wait.bin**

③ create upctalk0.bin (run on ARM formater core of upc)

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-ld -entry=0x0 -Ttext 0x0 -EL -Tdata **0x0401E000** -Tbss **0x0401E000** -o **upctalk0.elf** preboot_fm.o upctalk0.o dbg_printf.o

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-objcopy -O binary upctalk0.elf **upctalk0.bin**

④ create upctalk1.bin (run on ARM io core of upc)

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-

vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-ld -entry=0x0 -Ttext 0x0 -EL -Tdata **0x0401EA00** -Tbss **0x0401EA00** -o **upctalk1.elf** preboot_io.o upctalk1.o dbg_printf.o

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-objcopy -O binary upctalk1.elf **upctalk1.bin**

由于formater core and io core share the same data LCM，所以在compile在不同core上运行的elf时要让它们的data segment地址错开，否则它们的变量地址会冲突。目前把在formater core的elf的data 设置在offset为0x041e000,而在io core上运行的elf的data设置的offset为0x0401ea00. 两者相差0xa00，由于测试code都很小，应该够了。