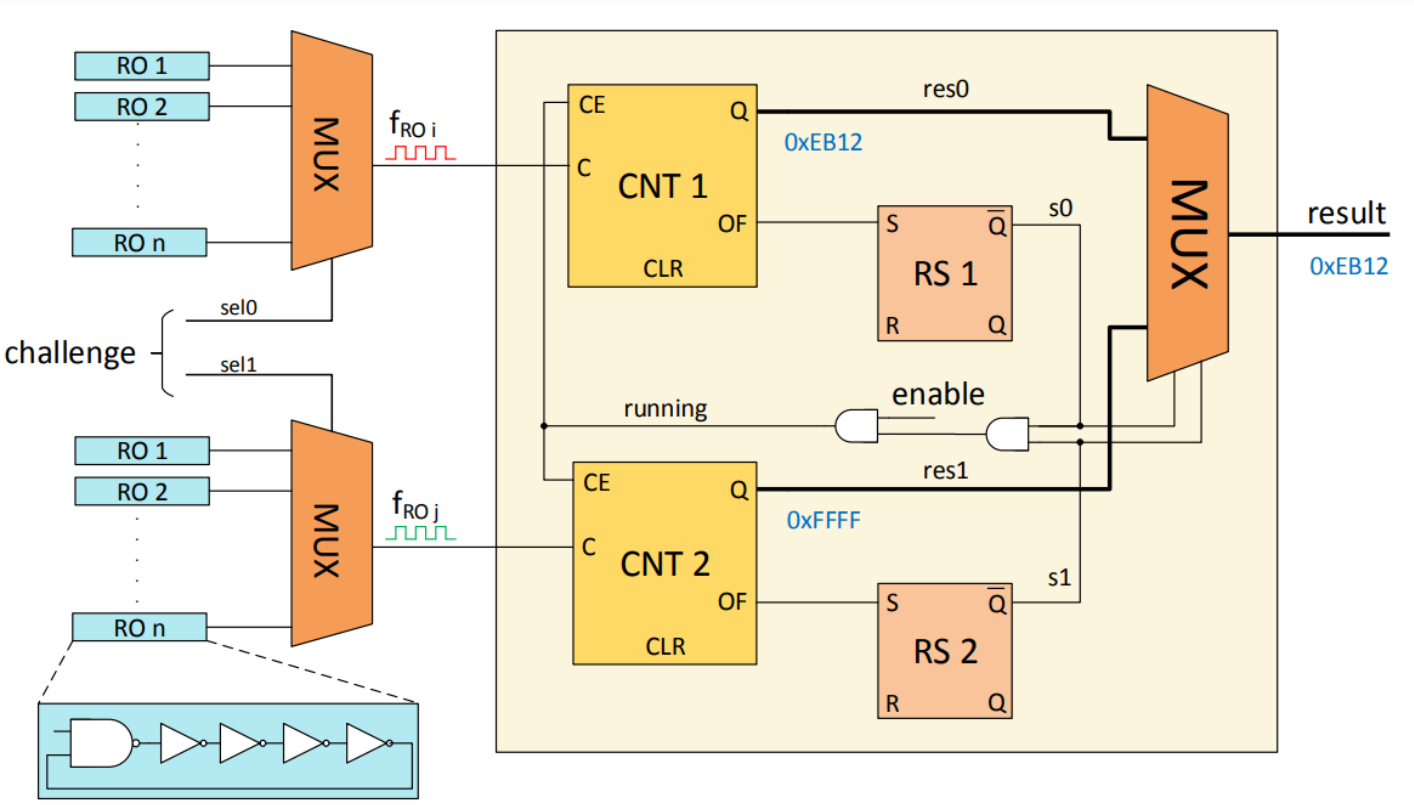
10. TRNG attack on FPGA

Máme TRNG relizovaný v FPGA, konkrétně na desce Digilent CMOD S7. Víme, že tento generátor se používá ke generování 128bitového tajného čísla k použití v dalších kryptografických opercích. Naším úkolem tedy je zaútočit na tento TRNG pomocí neinvazivní metody a odhalit generované tajemství. Máme přímý přístup k FPGA, kde běží daná implementace TRNG, a můžeme použít osciloskop k měření, které nám poslouží k extrakci tajné informace. Všechny potřebné soubory jsou v archivu TRNG_attack.zip (../files/TRNG_attack.zip)

TRNG, na který útočíme, je záměrně zjednodušený a upravený tak, aby byl útok co nejsnadnější (abychom všechno stihli na hodině). Vytvořené náhodné číslo nám pošle na konci měření. (Abychom ho mohli porovnat s výsledkem našeho útoku.) Kruhové oscilátory, které se interně používají pro generování náhody, jsou připojeny také na externí piny (Pmod konektor), aby se nám snadno měřily.

TRNG design

Návrh TRNG použitý pro tuto úlohu je zobrazen na následujícím obrázku:



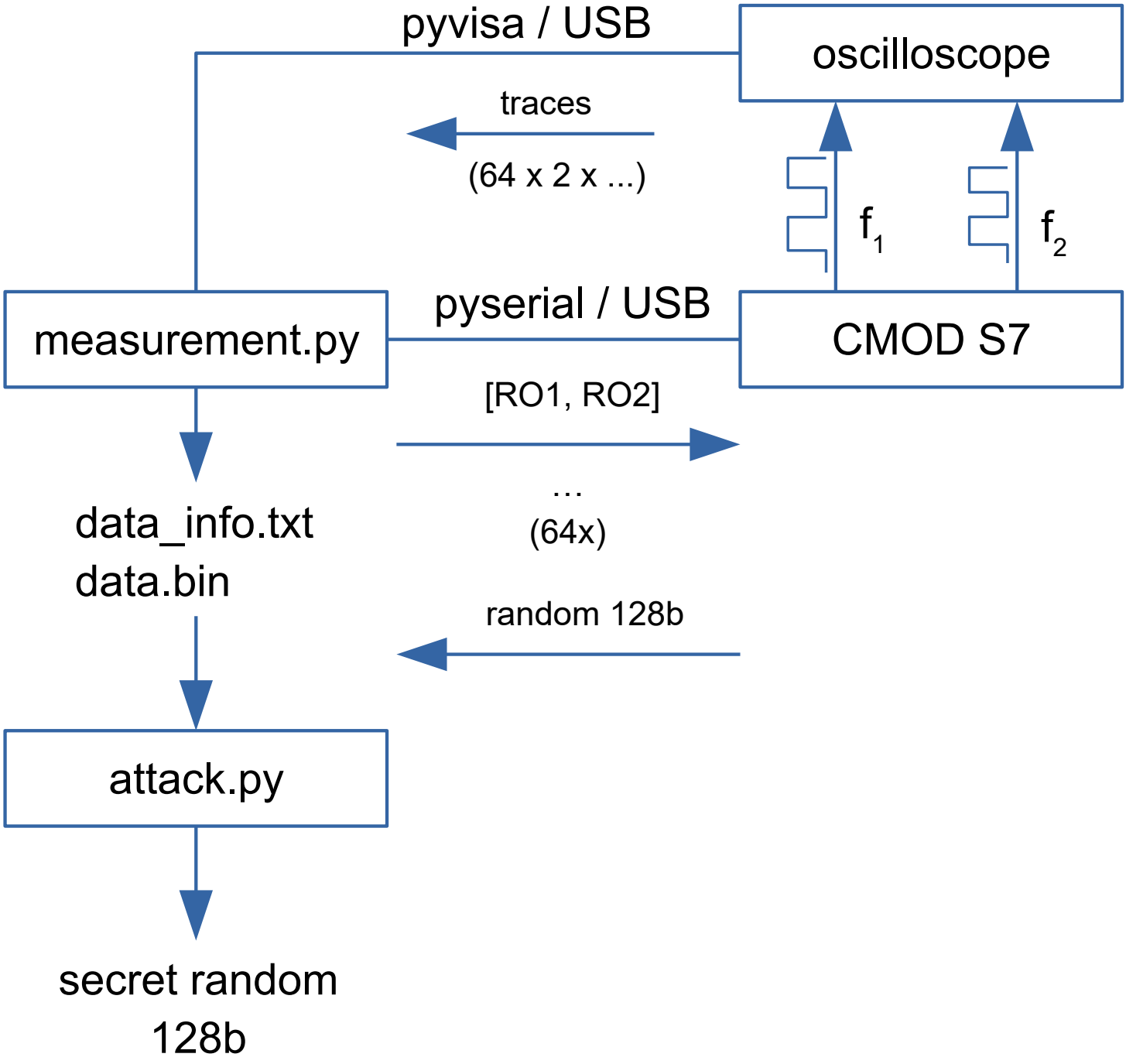
Tento TRNG je založená na kruhových oscilátorech (Ring Oscillator, RO). Obsahuje dvě sady RO po 64 RO v každé. Pro generování náhodných bitů se vždy vybere dvojice RO (z každé sady jeden), a jejich výstupy jsou připojeny na hodinové vstupy dvou binárních čítačů. Každý z těchto dvou čítačů je inkrementován při náběžných hranách na výstupu odpovídajícího RO. Jakmile jeden z čítačů přeteče (16bitovou hodnotu), jsou oba zastaveny, a druhý z čítačů (který nepřetekl) je použit jako syrová výstupní hodnota. Tato hodnota má na svých nejnižších bitech nejvyšší entropii, proto jsou dva nejnižší bity vybrány jako součást generované náhodné posloupnosti.

Řízení tohoto generátoru obstarává konečný automat, který vždy očekává na sériové lince dva bajty (identifikující použité RO z každé sady), pak spustí oscilátory a čítače, po naplnění čítače si zapamatuje dva nejnižší bity, a toto se opakuje celkem 64krát. Poté vyšle po sériové lince celé 128bitové číslo, které vzniklo spojením 64krát 2 bity z každého opakování.

Přehled útoku

Pro provedení útoku vykonáme následující kroky:

1. Připojíme FPGA k PC a naprogramujeme do něj bitstream s TRNG designem.
2. Zapneme osciloskop a připojímeho k PC (pokud ještě není).
3. Připojíme sondy k osciloskopu a pomocí měřicího adaptéru k Pmod konektoru FPGA desky.
4. Spustíme měřicí program.
5. Zpracujeme měřená data a odhalíme generovanou hodnout. Ověříme, že se tato hodnota shoduje s tím, co poslalo FPGA.



Konfigurace FPGA

Nahrajeme do FPGA bitstream s konfigurací, která obsahuje náš TRNG pomocí programu **Adept**. Nejprve musíme připojit FPGA desku pomocí USB kabelu. Pak spustíme progam **Adept** a najdeme naši FPGA desku v menu **Connect** (ale nejspíš už bude nalezena automaticky). Pak klikeme na **Browse** a zvolíme soubor `TRNG_impl.bit`, pak stiskneme **Program**.

Provedení měření

Je připraven python script `measurement.py`, který použijeme pro provedení vlastního měření. Už je hotový, ale možná ho bude potřeba mírně upravit, takže se nebojte si s ním trochu pohrát. Skript provádí kroky:

- Vypsání připojených osciloskopů
- Vypsání COM portů (asi budeme muset změnit index v `ports[0].name` podle toho, který COM port je přidělen USB/serial převodníku na FPGA desce)
- Inicializace a konfigurace osciloskopu (`scope.load_conf('scope_setup.conf')`)
- (volitelné) Nekonečný běh s různými páry RO (pro prohlížení na osciloskopu). Můžete jej přerušit pomocí CTRL-C.
- Provedení měření (`trng_read(scope, s)`)

Funkce `trng_read()` zapíše výsledky měření do 2 souborů. `data_info.txt` obsahuje délku průběhu (počet vzorků), vzorkovací frekvenci a generovanou náhodnou hodnotu. Druhý soubor, `data.bin`, obsahuje syrová data měřených průběhů (vzorek = uint8)

Pro provedení měření stačí spustit `measurement.py` (po drobných úpravách). Můžete také určit, které RO se mají použít, stačí nahradit `i` v `fpga_comm.write(bytes([i,i]))` za cokoliv uznáte za vhodné. Můžete nechat pár RO i stále stejný.

Odhalení náhodné hodnoty

Skript `attack_student.py` obsahuje téměř kompletní kód k odhalení tajemství z naměřených průběhů. Vaším úkolem je dokončit tento skript a ověřit generovanou TRNG hodnotu.